ระบบตรวจสอบการใช้น้ำและไฟฟ้าแบบไร้สาย
WATER AND ELECTRICITY CONSUMPTION MONITORING SYSTEM VIA WIRELESS

โดย

นายพศวีร์ ศรีพัฒน์

นายพีรณัฐ คำจำนรรจ์

นางสาวยุวดี ศรีสถาน

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

ภาควิชาวิศวกรรมโทรคมนาคม

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2560

ระบบตรวจสอบการใช้น้ำและไฟฟ้าแบบไร้สาย

WATER AND ELECTRICITY CONSUMPTION MONITORING SYSTEM VIA WIRELESS

โดย

นายพศวีร์	ศรีพัฒน์	57010858
นายพีรณัฐ	คำจำนรรจ์	57010913
นางสาวยุวดี	ศรีสถาน	57011039

อาจารย์ที่ปรึกษา

ผศ.ดร. ธเนศ พัฒนธาดาพงษ์

ดร. สมปอง วิเศษพานิชกิจ

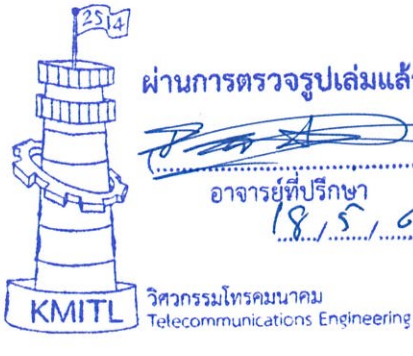
ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

ภาควิชาวิศวกรรมโทรคมนาคม

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2560



ปริญญานิพนธ์ปีการศึกษา 2560

ภาควิชาวิศวกรรมโทรคมนาคม

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง ระบบตรวจสอบการใช้น้ำและไฟฟ้าแบบไร้สาย

Water and electricity consumption monitoring system via wireless

ผู้จัดทำ

1.	นายพศวีร์	ศรีพัฒน์	57010858
2.	นายพีรณัฐ	คำจำนรรจ์	57010913
3.	นางสาวยุวดี	ศรีสถาน	57011039

..... อาจารย์ที่ปรึกษา

(ผศ.ดร. ธเนศ พัฒนธาดาพงษ์)

..... อาจารย์ที่ปรึกษาร่วม

(ดร. สมปอง วิเศษพานิชกิจ

กิตติกรรมประกาศ

ปริญญานิพนธ์ฉบับนี้สำเร็จไปได้ด้วยดีด้วยความกรุณาและความช่วยเหลือจากอาจารย์ที่ปรึกษา ผศ.ดร.ธเนศ พัฒนธาดาพงษ์ และอาจารย์ที่ปรึกษาร่วม ดร.สมปอง วิเศษพานิชกิจ ที่ให้คำแนะนำ คำปรึกษา เสนอแนวคิดทำให้จุดประกายความคิดใหม่ๆและคอยตรวจสอบแก้ไขข้อบกพร่องให้ตลอดการทำปริญญานิพนธ์นี้ และขอขอบพระคุณ ผศ.ดร.นภัทร สระเอี่ยม ที่คอยชี้แนะแนวทาง ให้คำแนะนำ คำปรึกษาและให้ข้อมูลเพิ่มเติม ตลอดจนปรับปรุงแก้ไขข้อบกพร่องต่างๆด้วยความเอาใจใส่อย่างดียิ่ง ทำให้ปริญญานิพนธ์ฉบับนี้มีความสมบูรณ์มากยิ่งขึ้น

ขอขอบพระคุณอาจารย์ประจำภาควิชาวิศวกรรมโทรคมนาคม คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบังทุกท่านที่ให้ความรู้และอบรมสั่งสอนแก่ผู้จัดทำ

ขอขอบพระคุณบิดา มารดา ที่คอยดูแลอุปการะอบรมเลี้ยงดู ตลอดจนส่งเสริมการศึกษาและให้กำลังใจเป็นอย่างดี สุดท้ายนี้ผู้จัดทำหวังเป็นอย่างยิ่งว่าปริญญานิพนธ์ฉบับนี้จะเป็นแนวทางในการประยุกต์ใช้ให้เกิดประโยชน์แก่ผู้ที่สนใจ และสามารถนำไปเป็นแนวทางในการพัฒนาให้เกิดประโยชน์ต่อไปในอนาคต หากมีข้อบกพร่องประการใดที่อาจจะเกิดขึ้นนั้น ทางคณะผู้จัดทำต้องขออภัยไว้ ณ ที่นี้ด้วย

นายพศวีร์ ศรีพัฒน์
นายพีรณัฐ คำจำนรรจ์
นางสาวยุวดี ศรีสถาน

ผู้จัดทำ

ระบบตรวจสอบการใช้น้ำและไฟฟ้าแบบไร้สาย
Water and electricity consumption monitoring system via wireless

โดย นายพศวีร์ ศรีพัฒน์ 57010858
นายพีรณัฐ คำจำนรรจ์ 57010913
นางสาวยุวดี ศรีสถาน 57011039

อาจารย์ที่ปรึกษา ผศ.ดร. ธเนศ พัฒนธาดาพงษ์
อาจารย์ที่ปรึกษาร่วม ดร. สมปอง วิเศษพานิชกิจ

บทคัดย่อ

ปริญญานิพนธ์ฉบับนี้เป็นการออกแบบและสร้างระบบตรวจสอบการใช้น้ำและไฟฟ้าแบบไร้สาย เพื่อที่จะนำไปคำนวณค่าน้ำประปาและค่าไฟฟ้า และแสดงผลผ่านหน้าต่างแอปพลิเคชั่นในรูปแบบของตารางและกราฟ เริ่มจากใช้ NodeMCU ในการอ่านข้อมูลจาก Power meter module และ Water flow sensor เพื่อประมวลผลและส่งข้อมูลผ่านโครงข่ายไร้สาย ด้วยเทคโนโลยี Wi-Fi บนมาตรฐาน IEEE802.11 ไปยังฐานข้อมูลที่ติดตั้งใน Raspberry Pi 3 และใช้ phpMyAdmin เป็นตัวจัดการฐานข้อมูล ในขณะเดียวกัน NodeMCU แต่ละตัวสามารถสื่อสารและแลกเปลี่ยนข้อมูลซึ่งกันและกันได้โดยใช้โปรโตคอล MQTT เพื่อช่วยแก้ปัญหาเมื่อ NodeMCU ตัวใดตัวหนึ่งไม่สามารถติดต่อกับฐานข้อมูลได้ NodeMCU ตัวอื่นจะส่งข้อมูลแทน

ABSTRACT

This thesis design and create wireless system used to measure and monitor the amount of tap water and electricity consumption. It is used to calculate water bills and electricity bills through application by receiving data from database to display in the form of tables and graphs using the NodeMCU to read data from Power meter and Water flow sensor to transmit readable data to database is installed in Raspberry Pi 3 by wireless network with Wi-Fi technology on IEEE802.11 standards and uses phpMyAdmin as the database manager. Each NodeMCU can communicate with each other using the MQTT protocol, which enables each NodeMCU to communicate and exchange data with each other to help resolve the

problem, when one NodeMCU fails to contact the database, another NodeMCU sends the data.

สารบัญ

		หน้า
กิตติกรรมประกาศ		I
บทคัดย่อ		II
สารบัญ		III
สารบัญรูป		V
สารบัญตาราง		VI
บทที่ 1	**บทนำ**	**1**
	1.1 ความเป็นมาและความสำคัญของปัญหา	1
	1.2 วัตถุประสงค์	1
	1.3 ขอบเขตของปริญญานิพนธ์	2
บทที่ 2	**ทฤษฎีและหลักการที่เกี่ยวข้อง**	**3**
	2.1 Power meter module: PZEM-004T AC multifunction meter	3
	2.1.1 รายละเอียดอุปกรณ์	4
	2.1.2 การคำนวณหน่วยไฟฟ้า	4
	2.1.3 การคำนวณไฟฟ้าเป็นจำนวนเงิน (บาท)	5
	2.2 Water flow sensor: G3/4 Water flow sensor	5
	2.2.1 รายละเอียดอุปกรณ์	6
	2.2.2 การคำนวณอัตราการไหลของน้ำ	6
	2.2.3 การคำนวณน้ำประปาเป็นจำนวนเงิน (บาท)	7
	2.3 NodeMCU: NodeMCU Lua ESP8266 CP2102	8
	2.3.1 รายละเอียดอุปกรณ์	9

สารบัญ(ต่อ)

		หน้า
	2.4 Raspberry Pi: Raspberry Pi 3	9
	2.4.1 รายละเอียดอุปกรณ์	10
	2.5 แบตเตอรี่สำรอง	10
	2.6 การใช้งานและเข้าถึงฐานข้อมูล	11
	2.6.1 Apache	11
	2.6.2 MySQL	11
	2.6.3 phpMyAdmin	12
	2.7 โปรโตคอล MQTT	12
บทที่ 3	**การออกแบบและการจัดทำปริญญานิพนธ์**	**14**
	3.1 การออกแบบ	14
	3.1.1 บล็อกไดอะแกรม	14
	3.1.2 แผนผังการทำงานของระบบการแสดงผลและการควบคุม	15
	3.1.3 การตั้งค่า Raspberry Pi 3	19
	3.1.4 วัดอัตราการไหลของน้ำด้วย Water flow sensor	27
	3.1.5 วัดอัตราการไหลของกระแสไฟฟ้าด้วย Power meter module	28
	3.1.6 เขียน PHP เพื่อรับค่าจาก NodeMCU ไปเก็บยังฐานข้อมูล	29
	3.1.7 การทำให้ผู้ใช้สามารถเข้ามาใช้งานระบบจากเครือข่ายภายนอกได้	31
	3.1.8 การสร้างการสื่อสารระหว่าง NodeMCU	32
	3.1.9 การรวมระบบการรับส่งข้อมูล MQTT	33

สารบัญ(ต่อ)

		หน้า
	3.2 เครื่องมือที่ใช้ในการทดลอง	36
	3.3 การจัดเก็บผลการทดลอง	36
บทที่ 4	**ผลการทดลอง**	**37**
	4.1 การทดสอบความถูกต้องของอุปกรณ์	37
	4.1.1 การทดสอบความถูกต้องของ Water flow sensor	37
	4.1.2 การทดสอบความถูกต้องของ Power meter module	42
	4.2 การคำนวณค่ายูนิต และส่งไปยังฐานข้อมูล	45
	4.2.1 แสดงค่าที่คำนวณได้บน serial monitor	45
	4.2.2 เขียนโปรแกรมเพื่อนำข้อมูลที่รับมาจาก NodeMCU ไปเก็บไว้ในฐานข้อมูล	46
	4.3 การนำข้อมูลจากฐานข้อมูลมาแสดงบนหน้าต่างเว็บเบราว์เซอร์	47
	4.3.1 ฐานข้อมูลของยูนิตของน้ำประปาและไฟฟ้า	47
	4.3.2 เขียนโปรแกรมเพื่ออัพเดทตารางฐานข้อมูล	48
	4.3.3 เขียนโปรแกรมเพื่อดึงข้อมูลจากฐานข้อมูลมาพล็อตกราฟและตาราง	49
	4.3.4 การสร้างการแจ้งเตือนบนเว็บเบราว์เซอร์	49
	4.3.5 การแสดงผลบนเว็บเบราว์เซอร์	50
	4.4 การสร้าง Mobile Application เว็บเบราว์เซอร์	56
บทที่ 5	**สรุปผลและข้อเสนอแนะ**	**58**
	5.1 สรุปผล	58
	5.2 ข้อเสนอแนะ	59

สารบัญ(ต่อ)

		หน้า
บรรณานุกรม		**60**
ภาคผนวก ก.	โปรแกรม NodeMCU ในส่วนระบบการส่งค่ายูนิตน้ำประปา	62
ภาคผนวก ข.	โปรแกรม PHP ในส่วนการส่งค่ายูนิตน้ำประปาไปยังฐานข้อมูล	71
ภาคผนวก ค.	โปรแกรม NodeMCU ในส่วนระบบการส่งค่ายูนิตไฟฟ้า	73
ภาคผนวก ง.	โปรแกรม PHP ในส่วนการส่งค่ายูนิตไฟฟ้าไปยังฐานข้อมูล	82
ภาคผนวก จ.	โปรแกรม PHP ในส่วนการแสดงผลบนหน้าต่างเว็บเบราว์เซอร์	84
ภาคผนวก ฉ.	โปรแกรมสร้าง Mobile Application เว็บเบราว์เซอร์	102

สารบัญรูป

รูปที่		หน้า
1.1	บล็อกไดอะแกรมภาพรวมของปริญญานิพนธ์	2
2.1	PZEM-004T	3
2.2	G3/4 Water flow sensor	5
2.3	โครงสร้างขาของ NodeMCU	8
2.4	ESP8266 CP2102	8
2.5	Raspberry Pi 3	9
2.6	ลักษณะการทำงานร่วมกันของแต่ละส่วน	13
3.1	บล็อกไดอะแกรมแสดงการทำงานของอุปกรณ์ต่างๆ	14
3.2	แผนผังการทำงานโดยรวมของระบบ	15
3.3	แผนผังการทำงานของระบบอ่านและส่งยูนิตน้ำประปา	16
3.4	แผนผังการทำงานของระบบอ่านและส่งยูนิตไฟฟ้า	17
3.5	แผนผังการทำงานของระบบการส่งข้อมูลเข้าฐานข้อมูล	18
3.6	แผนผังการแสดงผล	18
3.7	ติดตั้งระบบปฏิบัติการ Raspbian	19
3.8	การ Update โปรแกรม	19
3.9	การ Upgrade โปรแกรม	20
3.10	การดู IP Address ของเครื่อง User ปัจจุบัน	20
3.11	การดู IP Address ของเราเตอร์	20
3.12	การเปิดไฟล์ /etc/dhcpcd.conf	21
3.13	การกำหนด IP Address ให้เป็นแบบ Static	21
3.14	การดู IP Address	22

สารบัญรูป(ต่อ)

รูปที่		หน้า
3.15	การติดตั้ง apache2	22
3.16	การติดตั้ง mysql-server	23
3.17	การติดตั้ง PHP 5	23
3.18	หน้าต่างเบราว์เซอร์	24
3.19	การติดตั้ง phpMyAdmin	24
3.20	เลือก apache2 เป็น Web server	24
3.21	หน้าต่างเว็บเบราว์เซอร์ของ phpMyAdmin	25
3.22	การเปิดการทำงานของ ssh	25
3.23	ทำการเชื่อมต่อไปยัง Raspberry Pi	26
3.24	เป็นการเชื่อมต่อระหว่างฝั่ง User และฝั่ง Raspberry Pi	26
3.25	การต่อขาวงจรของ Water flow sensor เข้ากับ NodeMCU	27
3.26	เขียนโปรแกรมแสดงการอ่านและส่งค่าอัตราการไหลของน้ำ	27
3.27	การต่อโมดูล PZEM-004T กับ NodeMCU	28
3.28	เขียนโปรแกรมที่ใช้ในการอ่าน และส่งค่าอัตราการใช้ไฟฟ้า	29
3.29	เป็นการเพิ่มข้อมูลเข้าฐานข้อมูลโดยใช้คำสั่ง GET รอรับข้อมูลจาก	30
3.30	การใช้ phpMyAdmin สร้างตารางฐานข้อมูล เพื่อจัดการและจัดเก็บฐานข้อมูล	30
3.31	การ Set Forwarding Port	31
3.32	การจับคู่ Host Name กับ IP Address	31
3.33	ทำ NodeMCU เป็น Publisher	32
3.34	ทำ NodeMCU เป็น Subscriber	33
3.35	หน้าต่าง serial monitor ของ NodeMCU ที่ใช้ส่งค่ายูนิตไฟฟ้า	34
3.36	หน้าต่าง serial monitor ของ NodeMCU ที่ใช้ส่งค่ายูนิตน้ำประปา	34

สารบัญรูป(ต่อ)

รูปที่		หน้า
3.37	การรับส่งข้อมูล MQTT บนหน้าต่าง serial monitor	35
3.38	ปัญหาที่เกิดขึ้น	35
3.39	การแก้ไขปัญหาเมื่อ NodeMCU ตัวใดตัวหนึ่งไม่สามารถติดต่อกับฐานข้อมูลได้	36
4.1	มิเตอร์น้ำประปา	38
4.2	ค่ายูนิตจาก Water flow sensor ในแต่ละวัน	39
4.3	การแสดงผลการหมุนของใบพัดผ่าน serial monitor	40
4.4	สัญญาณพัลส์ของ Water flow sensor	40
4.5	การแสดงผลการหมุนของใบพัดผ่าน serial monitor เมื่อใบพัดหมุนไป 5 รอบ	41
4.6	สัญญาณพัลส์ของ Water flow sensor เมื่อใบพัดหมุนไป 5 รอบ	41
4.7	Power meter ยี่ห้อ Intertek	42
4.8	ผลที่ได้จาก Power meter module ผ่าน serial monitor	42
4.9	ผลที่ได้จากการวัดกำลังไฟฟ้าด้วย Power meter	43
4.10	ผลที่ได้จากการวัดกระแสไฟฟ้าด้วย Power meter	43
4.11	ผลที่ได้จากการวัดแรงดันไฟฟ้าด้วย Power meter	44
4.12	ค่ายูนิตของน้ำประปา และไฟฟ้าบน serial monitor	45
4.13	NodeMCU ส่งข้อมูลยูนิตน้ำแทน NodeMCU ที่ติดต่อกับฐานข้อมูลไม่ได้	45
4.14	NodeMCU ส่งข้อมูลยูนิตไฟฟ้าแทน NodeMCU ที่ติดต่อกับฐานข้อมูลไม่ได้	46
4.15	เขียนโปรแกรมเพื่อเก็บข้อมูลในฐานข้อมูล	46
4.16	ข้อมูลที่ถูกบันทึกลงในฐานข้อมูล	47
4.17	ฐานข้อมูลของยูนิตการใช้น้ำและไฟฟ้าที่ได้รับจาก NodeMCU	47
4.18	เขียนโปรแกรมเพื่ออัพเดทตารางฐานข้อมูล	48
4.19	ตารางฐานข้อมูลที่ทำการอัพเดท	48

สารบัญรูป(ต่อ)

รูปที่		หน้า
4.20	เขียนโปรแกรมเพื่อดึงข้อมูลจากฐานข้อมูลมาสร้างกราฟและตาราง	49
4.21	เขียนโปรแกรมการแจ้งเตือน	49
4.22	หน้าต่างแจ้งเตือนบนเว็บเบราว์เซอร์	50
4.23	หน้า Login	50
4.24	ส่วนของผู้ดูแลระบบ	51
4.25	ส่วนของผู้ใช้งาน	51
4.26	รายละเอียดข้อมูลการใช้น้ำประปาและไฟฟ้าเป็นรายปี	52
4.27	รายละเอียดข้อมูลการใช้น้ำประปาและไฟฟ้าเป็นรายเดือน	53
4.28	รายละเอียดข้อมูลการใช้น้ำประปาและไฟฟ้าเป็นรายวัน	53
4.29	หน้าต่างการแจ้งเตือน	54
4.30	ตัวอย่างการแจ้งเตือน	54
4.31	รายละเอียดข้อมูลการใช้น้ำประปาและไฟฟ้าเดือนปัจจุบัน	55
4.32	รายละเอียดข้อมูลการใช้น้ำประปาและไฟฟ้าแบบสะสมเป็นรายเดือน	55
4.33	การสร้าง Mobile Application ด้วย Android Studio	56
4.34	Mobile Application	57

สารบัญตาราง

ตารางที่		หน้า
2.1	อัตราค่าไฟฟ้าปกติที่เรียกเก็บ	5
2.2	อัตราค่าน้ำประปาที่เรียกเก็บ	7
4.1	ค่ายูนิตน้ำจากมิเตอร์น้ำประปาที่ใช้ในแต่ละวัน	38
4.2	ค่าความคลาดเคลื่อน	39
4.3	ผลการวัดจากอุปกรณ์ทั้งสองแบบ	44
4.4	ผลการคำนวณความคลาดเคลื่อน	44

บทที่ 1

บทนำ

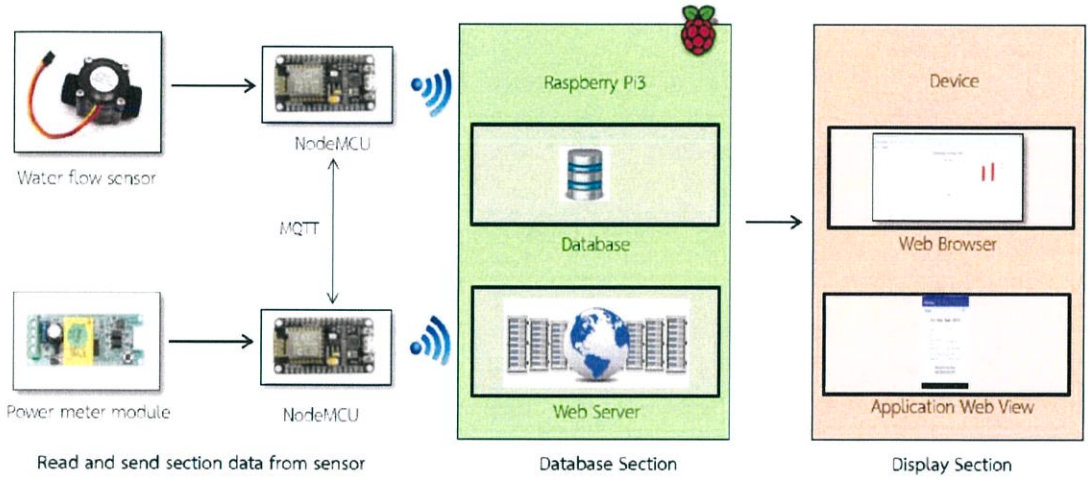
1.1 ความเป็นมาและความสำคัญของปัญหา

น้ำประปาและไฟฟ้าถือเป็นปัจจัยที่สำคัญในการใช้ชีวิตประจำวันของมนุษย์ ทั้งการใช้ในภาคครัวเรือน ภาคอุตสาหกรรมและในส่วนอื่นๆ ซึ่งหากต้องการทราบปริมาณการใช้งานน้ำประปาและไฟฟ้าระหว่างเดือนนั้นอาจจะไม่สะดวก เนื่องจากผู้ใช้น้ำประปาและไฟฟ้าจะทราบปริมาณการใช้ได้ก็ต่อเมื่อได้รับใบแจ้งค่าน้ำประปาและค่าไฟฟ้า หรือต้องไปดูที่มิเตอร์ตามบ้านเอง ซึ่งทำให้เสียเวลา และมีความยากในการบริหารจัดการ เนื่องจากต้องการข้อมูลในอดีตและปัจจุบันจึงจำเป็นต้องบันทึกข้อมูลอยู่ตลอดเพื่อที่จะทราบปริมาณการใช้งาน ณ เวลานั้น ทางผู้จัดทำได้เล็งเห็นถึงปัญหานี้ จึงได้นำระบบตรวจสอบการใช้น้ำประปาและไฟฟ้าแบบไร้สายมาพัฒนาเพื่อที่จะทำให้สามารถตรวจสอบและทราบปริมาณการใช้น้ำประปาและไฟฟ้าได้ในแต่ละเดือน แต่ละปี ซึ่งเป็นการประมาณจำนวนหน่วยที่ใช้ จำนวนเงินที่ต้องจ่าย และสามารถดูผ่านหน้าต่างแอปพลิเคชันได้ตลอดเวลา ทำให้ผู้ใช้น้ำประปาและไฟฟ้าเกิดความตระหนักถึงการใช้น้ำประปาและไฟฟ้า ซึ่งทำให้สามารถปรับเปลี่ยนพฤติกรรมการใช้น้ำประปาและไฟฟ้าเพื่อให้เกิดประโยชน์แก่ผู้ใช้

1.2 วัตถุประสงค์

1) เพื่อใช้วัดปริมาณการใช้น้ำประปาและไฟฟ้า แสดงผลผ่านหน้าต่างแอปพลิเคชัน
2) เพื่อศึกษาการทำงานของเซ็นเซอร์ชนิดต่างๆ
3) เพื่อศึกษาการทำงานของ Raspberry Pi เชื่อมต่อกับฐานข้อมูล
4) เพื่อศึกษาการทำงานของ NodeMCU
5) เพื่อศึกษาการทำงานของโปรโตคอล MQTT

1.3 ขอบเขตของปริญญานิพนธ์



รูปที่ 1.1 บล็อกไดอะแกรมภาพรวมของปริญญานิพนธ์

จากรูปที่ 1.1 เป็นบล็อกไดอะแกรมภาพรวมของปริญญานิพนธ์ แสดงขอบเขตการทำงานของแต่ละอุปกรณ์ที่ใช้ในปริญญานิพนธ์ดังนี้

1. ระบบสามารถแสดงค่าน้ำประปาและค่าไฟฟ้าผ่านหน้าต่างแอปพลิเคชันได้
2. ระบบใช้ Raspberry Pi เป็นฐานข้อมูล และใช้ phpMyAdmin เป็นตัวจัดการฐานข้อมูล
3. หน้าต่างแอปพลิเคชันแสดงผลปริมาณการใช้น้ำประปาและไฟฟ้า ในรูปของตารางและกราฟ โดยข้อมูลที่ถูกแสดงจะถูกดึงมาจากเว็บเซิร์ฟเวอร์
4. ระบบสามารถแจ้งเตือนการใช้น้ำประปาและไฟฟ้าผ่านแอปพลิเคชัน
5. ระบบสามารถให้ผู้ใช้สามารถเข้ามาใช้งานระบบจากเครือข่ายภายนอกได้
6. ระบบทำงานโดยใช้โปรโตคอล MQTT โดยทำให้ NodeMCU เป็น Broker ทำให้ NodeMCU แต่ละตัวสามารถสื่อสารและแลกเปลี่ยนข้อมูลซึ่งกันและกันได้ เพื่อช่วยแก้ปัญหาเมื่อ NodeMCU ตัวใดตัวหนึ่งไม่สามารถติดต่อกับฐานข้อมูลได้ จึงให้ NodeMCU ตัวอื่นส่งข้อมูลแทน

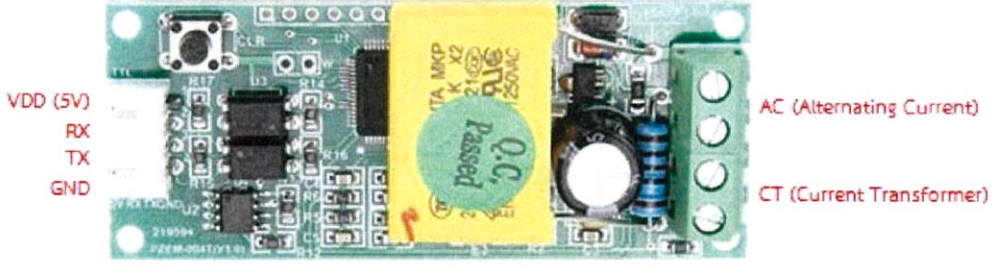
บทที่ 2

ทฤษฎีและหลักการที่เกี่ยวข้อง

ระบบตรวจสอบการใช้น้ำและไฟฟ้าแบบไร้สาย มีวัตถุประสงค์เพื่อใช้วัดปริมาณการใช้น้ำประปาและไฟฟ้า ซึ่งจะแสดงผลผ่านหน้าต่างแอปพลิเคชันในรูปของตารางและกราฟ โดยใช้ NodeMCU ESP8266 ในการอ่านข้อมูลจาก Power meter module และใช้ NodeMCU ESP8266 ในการอ่านข้อมูลจาก Water flow sensor เพื่อส่งข้อมูลที่อ่านได้ผ่านเทคโนโลยี Wi-Fi ไปยัง Raspberry Pi ซึ่งทำหน้าที่เป็นฐานข้อมูล ซึ่งใช้ phpMyAdmin เป็นตัวจัดการฐานข้อมูล สำหรับดึงข้อมูลไปแสดงบนเว็บเซิร์ฟเวอร์และ NodeMCU ทำตัวเป็น Broker โดยใช้โปรโตคอล MQTT ซึ่งสามารถสื่อสารและแลกเปลี่ยนข้อมูลซึ่งกันและกันได้ ดังนั้นจึงต้องศึกษาเกี่ยวกับทฤษฎีพื้นฐานต่างๆให้เข้าใจ เพื่อที่จะสามารถนำเอาข้อมูลและความรู้ที่ศึกษานั้นมาใช้ในการออกแบบ โดยอุปกรณ์ที่ใช้ในปริญญานิพนธ์ฉบับนี้ประกอบไปด้วย Power meter module, Water flow sensor, NodeMCU, Raspberry Pi 3 และใช้แบตเตอรี่สำรองเป็นแหล่งพลังงานสำรอง ซึ่งมีรายละเอียดดังนี้

2.1 Power meter module: PZEM-004T AC multifunction meter

PZEM-004T เป็นโมดูลที่ใช้วัดแรงดันไฟฟ้าของไฟฟ้าที่ใช้ทั่วไป วัดกระแสไฟฟ้าของอุปกรณ์ไฟฟ้าที่ใช้และวัดค่ากำลังไฟฟ้า ซึ่งสามารถนำค่าเหล่านี้ไปใช้คำนวณการใช้ไฟฟ้าได้หรือวัดปริมาณการใช้พลังงานของเครื่องใช้ไฟฟ้าแต่ละชิ้น



รูปที่ 2.1 PZEM-004T

จากรูปที่ 2.1 แสดง Power meter module ซึ่งในปริญญานิพนธ์ฉบับนี้ Power meter module เป็นอุปกรณ์ที่ใช้วัดแรงดันไฟฟ้า กระแสไฟฟ้าและวัดค่ากำลังไฟฟ้า โดยใช้

NodeMCU ทำการรับและอ่านค่าข้อมูลจาก Power meter module เพื่อทำการส่งข้อมูลไปยัง Raspberry Pi ต่อไป

2.1.1 รายละเอียดอุปกรณ์

1) Voltage: Test Range 80 – 260 VAC
2) Current: Test Range 0 – 100 A
3) Power Test Range 0 – 22 kW
4) Energy: Test Range 0 - 9999kWh
5) Operating frequency 45 - 65Hz
6) Communication with TTL Serial Communication or UART
7) Measurement accuracy 1.0 grade
8) Dimensions 7.3cm x 3cm

2.1.2 การคำนวณหน่วยไฟฟ้า

การคำนวณหน่วยไฟฟ้า(Unit) มีนิยามว่า การใช้ไฟฟ้า 1หน่วย คือเครื่องใช้ไฟฟ้าขนาด 1,000 วัตต์ที่ถูกใช้งานใน 1 ชั่วโมง คิดเป็นสูตรคำนวณได้ดังนี้

$$\text{Unit (unit/day)} = (\text{Power (watt)} / 1000) * n \text{ (hour/day)} \quad (2.1)$$

โดยให้ Power แทนปริมาณการใช้ไฟฟ้า (watt) และ n แทนจำนวนชั่วโมงที่ใช้งานใน 1 วัน จากนั้นนำมาแปลงสมการเพื่อหาค่า Unit (unit/min) ได้ดังนี้

$$\text{Unit (unit/min)} = \text{Sumwatt (watt/min)} / 3600 / 1000 \quad (2.2)$$

โดยให้ Sumwatt เป็นปริมาณการใช้ไฟฟ้า (watt) ในแต่ละนาที

2.1.3 การคำนวณไฟฟ้าเป็นจำนวนเงิน (บาท)

ตารางที่ 2.1 อัตราค่าไฟฟ้าปกติที่เรียกเก็บ

การใช้ไฟฟ้า	อัตราค่าใช้ไฟฟ้าที่เรียกเก็บ	
1.ใช้พลังงานไฟฟ้าไม่เกิน 150 หน่วยต่อเดือน	ค่าพลังงานไฟฟ้า (บาท/หน่วย)	ค่าบริการ (บาท/เดือน)
ค่าบริการรายเดือน		8.19
หน่วยที่ 0-15	2.3488	
หน่วยที่ 16-25	2.9882	
หน่วยที่ 26-35	3.2405	
หน่วยที่ 36-100	3.6237	
หน่วยที่ 101-150	3.7171	
หน่วยที่ 151-400	4.2218	
หน่วยที่ 401 เป็นต้นไป	4.4217	

*อัตราค่าไฟฟ้านี้ เริ่มใช้ตั้งแต่ค่าไฟฟ้าประจำเดือนพฤศจิกายน 2558 เป็นต้นไป [1]

จากตารางที่ 2.1 อัตราค่าไฟฟ้าเป็นอัตราปกติที่เรียกเก็บสำหรับการใช้ไฟฟ้าในที่อยู่อาศัยทั่วไป โดยอ้างอิงจากอัตราค่าไฟฟ้าของการไฟฟ้านครหลวงและการไฟฟ้าส่วนภูมิภาค ซึ่งขอบเขตของปริญญานิพนธ์ฉบับนี้กำหนดให้ใช้อัตราปกตินี้ในการคำนวณค่าไฟฟ้า

2.2 Water flow sensor: G3/4 Water flow sensor

Water flow sensor เป็นเซ็นเซอร์สำหรับวัดอัตราการไหลของน้ำสามารถต่อกับท่อ PVC ขนาด 3/4 นิ้วได้ โดยมีการส่งสัญญาณพัลส์ (Pulse Signal) ออกมา

รูปที่ 2.2 G3/4 Water flow sensor

จากรูป 2.2 คือ G3/4 Water flow sensor ซึ่งในปริญญานิพนธ์ฉบับนี้ G3/4 Water flow sensor เป็นอุปกรณ์ตรวจวัดอัตราการไหลของน้ำ โดยใช้ NodeMCU ทำการรับและอ่าน

ข้อมูลจาก G3/4 Water flow sensor เพื่อทำการส่งข้อมูลไปยัง Raspberry Pi ต่อไป ส่วนการต่อวงจรนั้นจะป้อนไฟเลี้ยงให้กับเซ็นเซอร์ที่แรงดัน 5V สายสีดำเป็นกราวด์ และสายสีเหลืองเป็นสายสัญญาณพัลส์ขาออก

2.2.1 รายละเอียดอุปกรณ์

1) Working voltage : 5V-24V
2) Mini. working voltage : DC 4.5V
3) Max. working current : 15mA (DC 5V)
4) External diameters : 26mm
5) Flow rate range : 1~60 L/min
6) Operating temperature : 0°C ~80°C
7) Liquid temperature : <120°C
8) Operating humidity : 35%~90%RH
9) Operating pressure : under 1.75Mpa
10) Store temperature : -25°C~+80°C

2.2.2 การคำนวณอัตราการไหลของน้ำ

ข้อมูลที่ได้จากสัญญาณจะอยู่ในรูปของดิจิทัลพัลส์ (Digital Pulse) ความถี่สัญญาณที่ออกมาจะแปรเปลี่ยนตามความเร็วการไหลของน้ำ โดยตัวเซ็นเซอร์ปล่อยสัญญาณออกมา 1 พัลส์ เมื่อใบพัดภายในตัวมิเตอร์หมุนไป 1 รอบ ดังนั้นหลักการวัดอัตราการไหลน้ำจึงใช้การนับจำนวนที่ถูกส่งออกมาใน 1 หน่วยเวลา สำหรับรุ่น G3/4 Water flow sensor ที่อัตราการไหล 1 L/min สัญญาณที่ออกมามีความถี่ 5.5 Hz ซึ่งสูตรในการคำนวณอัตราการไหลเฉลี่ยเป็นดังนี้

$$Q = \frac{freq}{coeff} \tag{2.3}$$

โดยให้ Q แทนอัตราการไหลในหน่วย (L/min)

freq แทนจำนวนพัลส์ที่นับได้ใน 1 วินาที

coeff แทนค่าคงที่ของความถี่ที่ปล่อยออกมาเทียบกับอัตราการไหล (ในที่นี้คือ 5.5 Hz per L/min)

ดังนั้น จาก 1 Unit = 1000 L เพราะฉะนั้นปริมาณน้ำในหน่วย unit / min จะเท่ากับ

$$\text{Unit (unit / min)} = \frac{Q\ (L / min)}{1000} \tag{2.4}$$

2.2.3 การคำนวณน้ำประปาเป็นจำนวนเงิน (บาท)

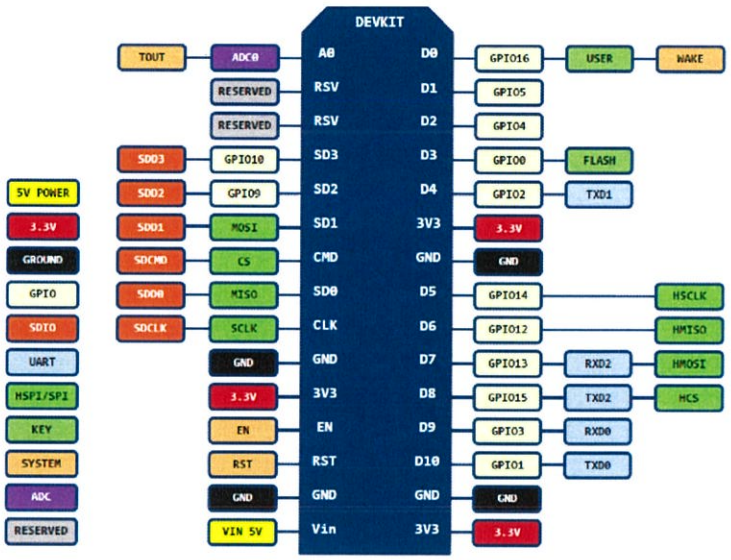
ตารางที่ 2.2 อัตราค่าน้ำประปาที่เรียกเก็บ

ปริมาณน้ำประปาใช้ (ลูกบาศก์เมตร)	ราคาค่าน้ำประปา (บาท/ลูกบาศก์เมตร)
1-30	8.50
31-40	10.03
41-50	10.35
51-60	10.68
61-70	11.00
71-80	11.33
81-90	12.50
91-100	12.82
101-120	13.15
121-160	13.47
161-200	13.80
มากกว่า 200	14.45

*อัตราค่าน้ำประปานี้ เริ่มใช้ตั้งแต่ ค่าน้ำประปาประจำเดือนธันวาคม 2542 เป็นต้นไป [2]

จากตารางที่ 2.2 อัตราค่าน้ำประปาในอัตราปกติที่เรียกเก็บ สำหรับการใช้น้ำประปาในบ้านเรือนที่อยู่อาศัย ซึ่งขอบเขตของปริญญานิพนธ์ฉบับนี้กำหนดให้ใช้อัตราปกตินี้ในการคำนวณค่าน้ำประปา

2.3 NodeMCU: NodeMCU Lua ESP8266 CP2102



รูปที่ 2.3 โครงสร้างขาของ NodeMCU

NodeMCU เป็นบอร์ดที่ใช้ ESP8266 เป็น CPU สำหรับประมวลผลโปรแกรมต่างๆ มีข้อดีกว่า Arduino คือมีขนาดเล็กกว่า มีหน่วยความจำสำหรับเขียนโปรแกรมมากกว่า และสามารถใช้เทคโนโลยี Wi-Fi ได้ มีพื้นที่หน่วยความจำรอมสูงถึง 4MB เพียงพอสำหรับการเขียนโปรแกรมขนาดใหญ่ ใช้ความถี่สูงถึง 40MHz ทำให้สามารถประมวลผลโปรแกรมได้อย่างรวดเร็ว

รูปที่ 2.4 ESP8266 CP2102

จากรูปที่ 2.4 แสดงรูปของ NodeMCU ซึ่งในปริญญานิพนธ์ฉบับนี้ใช้ NodeMCU เป็นอุปกรณ์ที่ใช้อ่านข้อมูลจาก Power meter module หรือ Water flow sensor จากนั้นทำการส่งข้อมูลไปยัง Raspberry Pi ผ่านเทคโนโลยี Wi-Fi ซึ่งควบคุมการทำงานทั้งหมดผ่านโปรแกรม Arduino IDE และ NodeMCU ทำตัวเป็น Broker โดยใช้โปรโตคอล MQTT ซึ่งสามารถสื่อสารและแลกเปลี่ยนข้อมูลซึ่งกันและกันได้

2.3.1 รายละเอียดอุปกรณ์

1) USB – micro USB port for power, programming and debugging
2) Headers – 2x 2.54mm 15-pin header with access to GPIOs, SPI, UART, ADC, and power pins
3) Misc – Reset and Flash buttons
4) Input Voltage Range = 4.0 - 5.25 V
5) Output Voltage Range = 3.0 – 3.6 V
6) Maximum Total current through VDD and GND = 500 mA
7) Maximum output current sunk by RST or any I/O pin = 100 mA
8) PCB antenna
9) Temperature Range: –40 to +85 °C
10) Dimensions 49 x 24.5 x 13mm

2.4 Raspberry Pi: Raspberry Pi 3

Raspberry Pi 3 คือบอร์ดคอมพิวเตอร์ขนาดเล็ก มีคุณสมบัติเด่นคือ ติดต่อสื่อสารและควบคุมอุปกรณ์อิเล็กทรอนิกส์ได้ เป็นคอมพิวเตอร์ที่มีความสามารถในการใช้งานทั่วไปเช่น ใช้เพื่อทำงานเอกสาร ใช้เพื่อการคำนวณต่างๆ สั่งงานให้เครื่องใช้ไฟฟ้าทำงาน หรือหยุดทำงาน นอกจากนี้ยังสามารถทำเป็นเว็บเซิร์ฟเวอร์ได้อีกด้วย

รูปที่ 2.5 Raspberry Pi 3

จากรูป 2.5 แสดงรูปของ Raspberry Pi 3 ซึ่งในปริญญานิพนธ์ฉบับนี้ Raspberry Pi 3 เป็นอุปกรณ์ที่ทำหน้าที่เป็นทั้งเซิร์ฟเวอร์และที่เก็บฐานข้อมูลที่จะสามารถเชื่อมต่อกับ NodeMCU ผ่านเทคโนโลยี Wi-Fi เพื่อรับข้อมูลไปเก็บไว้ในฐานข้อมูล สำหรับดึงข้อมูลไปแสดงบนแอปพลิเคชัน

2.4.1 รายละเอียดอุปกรณ์

1) Broadcom BCM2387 chipset
2) 802.11 bgn Wireless LAN and Bluetooth 4.1 (Bluetooth Classic)
3) 1GB RAM
4) 4 USB ports
5) 10/100BaseT Ethernet socket to quickly connect the Raspberry Pi to the Internet
6) DSI display port for connecting the Raspberry Pi touch screen display
7) Micro SD port for loading your operating system and storing data
8) Micro USB power source
9) 1.2GHz 64-bit quad-core ARM Cortex-A53 CPU (~ 10x the performance of Raspberry Pi 1)
10) Integrated 802.11n wireless LAN and Bluetooth 4.1
11) Complete compatibility with Raspberry Pi 1 and 2

2.5 แบตเตอรี่สำรอง

เนื่องจากอุปกรณ์ที่ใช้ในปริญญานิพนธ์ฉบับนี้ต้องการพลังงานสำรองเพื่อใช้ในช่วงที่แหล่งพลังงานหลักมีปัญหาเช่น ไฟฟ้าดับ เป็นต้น เพราะฉะนั้นจึงต้องเลือกขนาดของแบตเตอรี่สำรองให้มีขนาดและความจุที่เหมาะสม

ภายในแบตเตอรี่พกพาจะมีแบตเตอรี่ซึ่งในปัจจุบันจะเป็น Lithium-Ion และหลักการในการป้อนกระแสไฟฟ้าเข้าไปคือ การแปลงพลังงานไฟฟ้าไปเป็นพลังงานทางเคมี และเมื่อมีการคายพลังงานออก จะต้องสูญเสียค่าในการแปลงพลังงาน และจ่ายกระแสไฟฟ้าออกเช่นกัน โดยปกติค่าที่สูญเสียไปของแบตเตอรี่จะอยู่เฉลี่ยราวๆ 30% ซึ่งขึ้นอยู่กับปัจจัยแวดล้อมเช่น อุณหภูมิเป็นต้น

$$T = \frac{\text{ความจุของแบตเตอรี่} \times (70/100)}{\text{อัตราการใช้กระแสต่อชั่วโมง}} \qquad (2.5)$$

โดยให้ T แทนระยะเวลาที่อุปกรณ์สามารถทำงานได้ในหน่วยชั่วโมง

2.6 การใช้งานและเข้าถึงฐานข้อมูล

2.6.1 Apache

Apache คือ เว็บเซิร์ฟเวอร์พัฒนามาจาก HTTPD Web Server โดย Apache นี้จะทำหน้าที่ในการจัดเก็บ Webpage และส่ง Webpage ไปยังเบราว์เซอร์ที่มีการเรียกเข้า ซึ่งปัจจุบันจัดได้ว่าเป็นเว็บเซิร์ฟเวอร์ที่มีความน่าเชื่อถือมากเนื่องจากเป็นที่นิยมใช้กันทั่วโลก อีกทั้ง Apache ยังเป็นซอฟต์แวร์แบบ Open Source ที่เปิดให้บุคคลทั่วไปสามารถเข้ามาร่วมพัฒนาส่วนต่างๆของ Apache ได้ซึ่งทำให้เกิดเป็นโมดูลที่เกิดประโยชน์มากมายและทำงานร่วมกับภาษาอื่นได้ แทนที่จะเป็นเพียงเซิร์ฟเวอร์ที่ให้บริการเพียงแค่ HTML อย่างเดียว โดยสามารถหาดาวน์โหลดได้จากเว็บไซต์ www.apache.org นอกจากนี้ Apache เองยังมีความสามารถอื่นๆด้วยเช่น เพิ่มความปลอดภัยในการสื่อสารผ่านโปรโตคอล HTTPS และยังมีโมดูลอื่นๆที่ได้รับความนิยมใช้อีกมากมาย

2.6.2 MySQL

MySQL คือ โปรแกรมระบบจัดการฐานข้อมูลที่พัฒนาโดยบริษัท MySQL AB มีหน้าที่เก็บข้อมูลอย่างเป็นระบบ รองรับคำสั่ง SQL เป็นเครื่องมือสำหรับเก็บข้อมูล ที่ต้องใช้ร่วมกับเครื่องมือหรือโปรแกรมอื่นอย่างบูรณาการ เพื่อให้ได้ระบบงานที่รองรับความต้องการของผู้ใช้เช่น ทำงานร่วมกับเว็บเซิร์ฟเวอร์เพื่อให้บริการแก่ภาษาสคริปต์ที่ทำงานฝั่ง Server-Side Script หรือทำงานร่วมกับโปรแกรมประยุกต์ (Application Program) โปรแกรมถูกออกแบบให้สามารถทำงานได้บนระบบปฏิบัติการที่หลากหลายและเป็นระบบฐานข้อมูล Open Source ที่ถูกนำไปใช้งานมากที่สุด MySQL จัดเป็นระบบจัดการฐานข้อมูลเชิงสัมพันธ์ (RDBMS : Relational Database Management System) ซึ่งเป็นที่นิยมใช้กันมากในปัจจุบัน โดยความสามารถและการทำงานของโปรแกรม MySQL ได้แก่

1) MySQL ถือเป็นระบบจัดการฐานข้อมูล DataBase Management System (DBMS) ฐานข้อมูลมีลักษณะเป็นโครงสร้างของการเก็บรวบรวมข้อมูล การที่จะเข้าถึงหรือประมวลผลข้อมูลที่เก็บในฐานข้อมูลจำเป็นจะต้องอาศัยระบบจัดการฐานข้อมูล ซึ่งจะทำหน้าที่เป็นตัวกลางในการจัดการกับข้อมูลในฐานข้อมูลทั้งสำหรับการใช้งานเฉพาะ และรองรับการทำงานของแอปพลิเคชันอื่นๆ ที่ต้องการใช้งานข้อมูลในฐานข้อมูล เพื่อให้ได้รับความสะดวกในการจัดการกับข้อมูลจำนวนมาก MySQL ทำหน้าที่เป็นทั้งตัวฐานข้อมูลและระบบจัดการฐานข้อมูล

2) MySQL เป็นระบบจัดการฐานข้อมูลแบบ Relational โดยจะทำการเก็บข้อมูลทั้งหมดในรูปแบบของตารางแทนการเก็บข้อมูลทั้งหมดลงในไฟล์ เพียงไฟล์เดียว ทำให้ทำงานได้รวดเร็วและมีความยืดหยุ่น นอกจากนั้นแต่ละตารางที่เก็บข้อมูลสามารถเชื่อมโยงเข้าหากันทำให้

สามารถรวมหรือจัดกลุ่มข้อมูลได้ตามต้องการ โดยอาศัยภาษา SQL ที่เป็นส่วนหนึ่งของโปรแกรม MySQL ซึ่งเป็นภาษามาตรฐานในการเข้าถึงฐานข้อมูล

3) MySQL แจกจ่ายให้ใช้งานแบบ Open Source ผู้ใช้งาน MySQL ทุกคน สามารถใช้งานและปรับแต่งการทำงานได้ตามต้องการ ซึ่งสามารถดาวน์โหลดโปรแกรม MySQL ได้จากอินเทอร์เน็ตและนำมาใช้งานโดยไม่มีค่าใช้จ่ายใดๆ

2.6.3 phpMyAdmin

phpMyAdmin คือโปรแกรมที่ถูกพัฒนาโดยใช้ภาษา PHP เพื่อใช้ในการบริหารจัดการฐานข้อมูล MySQL แทนการคีย์คำสั่ง เนื่องจากถ้าจะใช้ฐานข้อมูลที่เป็น MySQL บางครั้งจะมีความลำบากและยุ่งยากในการใช้งาน ดังนั้นจึงมีเครื่องมือในการจัดการฐานข้อมูล MySQL ขึ้นมาเพื่อให้สามารถจัดการDBMSที่เป็น MySQL ได้ง่ายและสะดวกยิ่งขึ้น

phpMyAdmin เป็นส่วนต่อประสานที่สร้างโดยภาษา PHP ซึ่งใช้จัดการฐานข้อมูล MySQL ผ่านเว็บเบราว์เซอร์ โดยสามารถที่จะทำการสร้างฐานข้อมูลใหม่หรือทำการสร้าง TABLE ใหม่ๆ และยังมีฟังก์ชันที่ใช้สำหรับการทดสอบการ Query ข้อมูลด้วยภาษา SQL อีกทั้งยังสามารถทำการ insert, delete, update หรือแม้กระทั่งใช้คำสั่งต่างๆเหมือนกันกับการใช้ภาษา SQL ในการสร้างตารางข้อมูล phpMyAdmin เป็นโปรแกรมประเภท MySQL Client ใช้ในการจัดการข้อมูล MySQL ผ่านเว็บเบราว์เซอร์ได้โดยตรง โดย phpMyAdmin นี้จะทำงานบนเว็บเซิร์ฟเวอร์เป็น PHP Application ที่ใช้ควบคุมจัดการ MySQL Server ความสามารถของ phpMyAdmin ได้แก่ สร้างและลบฐานข้อมูล, สร้างและจัดการตาราง และโหลด Text files เข้าไปเก็บเป็นข้อมูลในตารางได้ และหาผลสรุปด้วยคำสั่ง SQL

2.7 โปรโตคอล MQTT

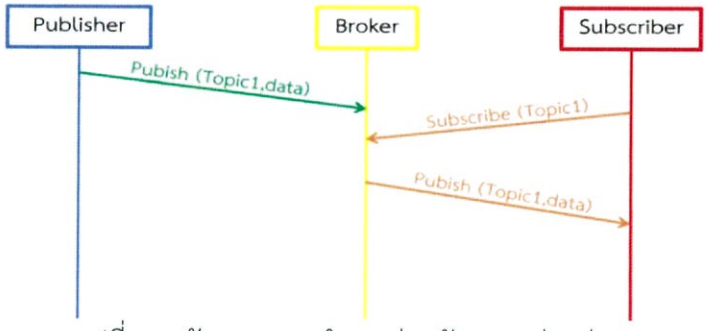
Message Queuing Telemetry Transport หรือ MQTT เป็นโปรโตคอลที่ออกแบบมาเพื่อการเชื่อมต่อแบบ M2M (Machine to Machine) คือการเชื่อมต่อระหว่างอุปกรณ์กับอุปกรณ์ เพื่อใช้สนับสนุนเทคโนโลยี IoT (Internet of Things) ซึ่งเป็นเทคโนโลยีที่เชื่อมต่อกับอุปกรณ์ต่างๆ เข้ากับอินเทอร์เน็ตทำให้สามารถเชื่อมโยงและสื่อสารกับอุปกรณ์อื่นๆ ซึ่งจะทำให้มนุษย์สามารถควบคุมอุปกรณ์ต่างๆจากที่อื่นได้โดยผ่านอินเทอร์เน็ต โดยมีการสร้างการเชื่อมต่อแบบ TCP ข้อดีของโปรโตคอลชนิดนี้คือมีขนาดเล็ก ออกแบบมาเพื่อใช้งานกับอุปกรณ์อิเล็กทรอนิกส์ขนาดเล็ก หรือการรับส่งข้อมูลในเครือข่ายที่มีขนาดเล็ก แบนด์วิดท์ต่ำ โดยหลักการสื่อสารของโปรโตคอลชนิดนี้ใช้หลักการแบบ Publisher/Subscriber ซึ่งมีความคล้ายคลึงกับหลักการที่ใช้ใน

Web Service ที่ต้องใช้เว็บเซิร์ฟเวอร์เป็นตัวกลางระหว่างคอมพิวเตอร์ของผู้ใช้ แต่ MQTT จะใช้ตัวกลางที่เรียกว่า Broker เพื่อทำหน้าที่จัดการคิวการส่งข้อมูล และทำหน้าที่รับส่งข้อมูลระหว่างอุปกรณ์ โดยข้อมูลที่รับส่งกันนั้นจะใช้ Topic เป็นตัวอ้างอิงหลัก โดยข้อมูลที่รับส่งกันจะต้องมีชื่อ Topic กำหนดไว้เสมอ องค์ประกอบของการสื่อสารโดยใช้โปรโตคอล MQTT ซึ่งแต่ละส่วนทำหน้าที่ต่างกันได้แก่

1) Broker ทำหน้าที่เป็นตัวกลางคอยจัดการกับข้อมูลโดยอ้างอิงจาก Topic

2) Publisher ทำหน้าที่คอยส่งข้อมูลไปยัง Topic นั้นๆ

3) Subscriber ทำหน้าที่คอยดูการเปลี่ยนแปลงของข้อมูลที่อ้างอิงด้วย Topic โดยถ้ามีการเปลี่ยนแปลงก็จะทำการดึงข้อมูลนั้นๆมาใช้งาน



รูปที่ 2.6 ลักษณะการทำงานร่วมกันของแต่ละส่วน

จากรูปที่ 2.6 แสดงให้เห็นถึงลักษณะการทำงานของโปรโตคอล MQTT ซึ่งจะเห็นได้ว่า Broker จะทำหน้าที่เป็นตัวกลางคอยจัดการรับและส่งข้อมูลตามที่ Publisher และ Subscriber ต้องการ และในการสื่อสารแต่ละครั้งจะมีการระบุ Topic ที่ต้องการสื่อสารเสมอ

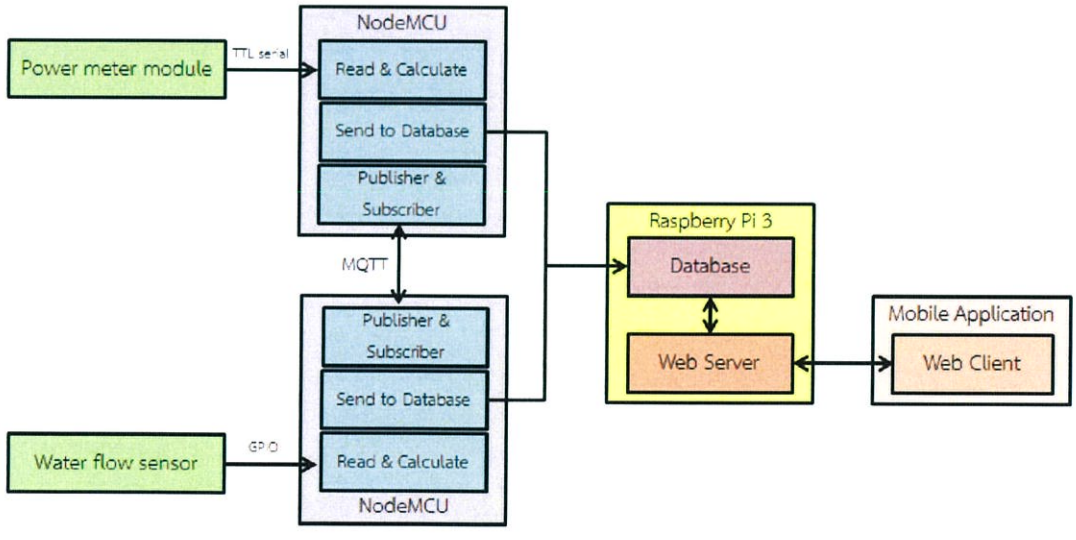
บทที่ 3

การออกแบบและการจัดทำปริญญานิพนธ์

3.1 การออกแบบ

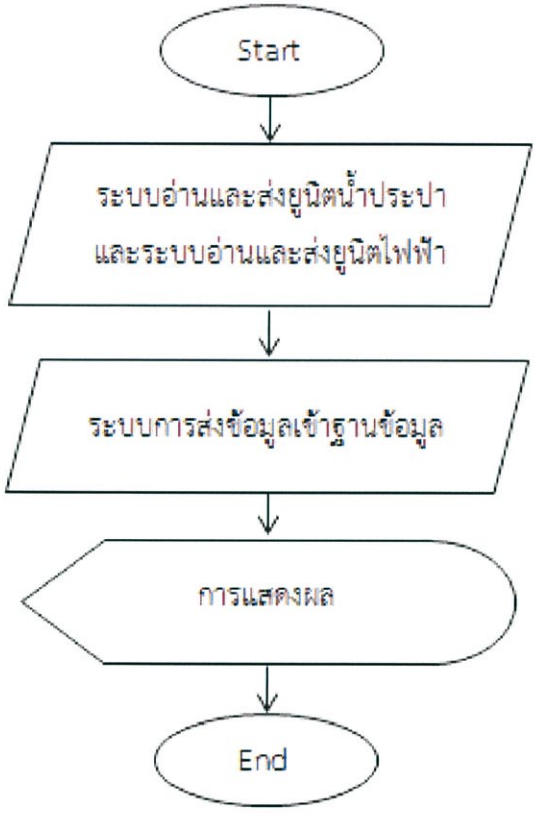
จากภาพรวมของปริญญานิพนธ์ในบทที่ 1 รูปที่ 1.1 แสดงภาพรวมการทำงานของระบบซึ่งประกอบด้วย ส่วนอ่านและส่งข้อมูลจากเซ็นเซอร์ ส่วนของฐานข้อมูล และส่วนแสดงผล โดยมีรายละเอียดดังนี้ 1) ส่วนอ่านและส่งข้อมูลจากเซ็นเซอร์ประกอบด้วย NodeMCU (ESP8266) อ่านข้อมูลจาก Water flow sensor และ Power meter module แล้วทำการคำนวณหาค่ายูนิตน้ำประปาและไฟฟ้า และส่งข้อมูลไปยังฐานข้อมูลผ่านเทคโนโลยีการสื่อสารไร้สาย(Wi-Fi) 2) ส่วนฐานข้อมูลใช้ Raspberry Pi 3 เป็นฐานข้อมูลและใช้ phpMyAdmin เป็นตัวจัดการฐานข้อมูลโดยรับข้อมูลมาจาก NodeMCU มาเก็บในฐานข้อมูล จากนั้นทำการรวมยูนิตน้ำประปา และยูนิตไฟฟ้าแล้วนำมาคำนวณค่าน้ำประปาและค่าไฟฟ้า จากนั้นส่งไปยังส่วนแสดงผล 3) ส่วนแสดงผล จะแสดงผลผ่านหน้าต่างแอปพลิเคชันในรูปแบบตารางและกราฟ ซึ่งข้อมูลนั้นได้รับมาจากฐานข้อมูล

3.1.1 บล็อกไดอะแกรม

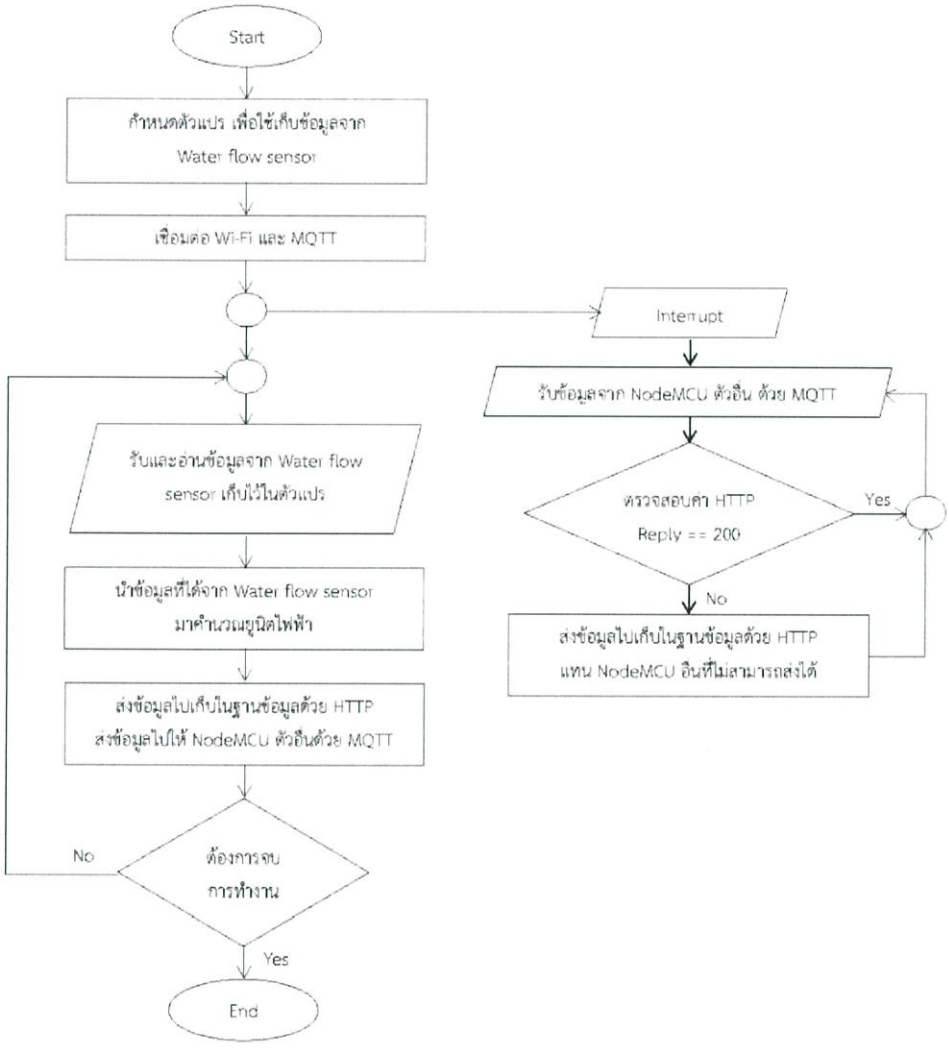


รูปที่ 3.1 บล็อกไดอะแกรมแสดงการทำงานของอุปกรณ์ต่างๆ

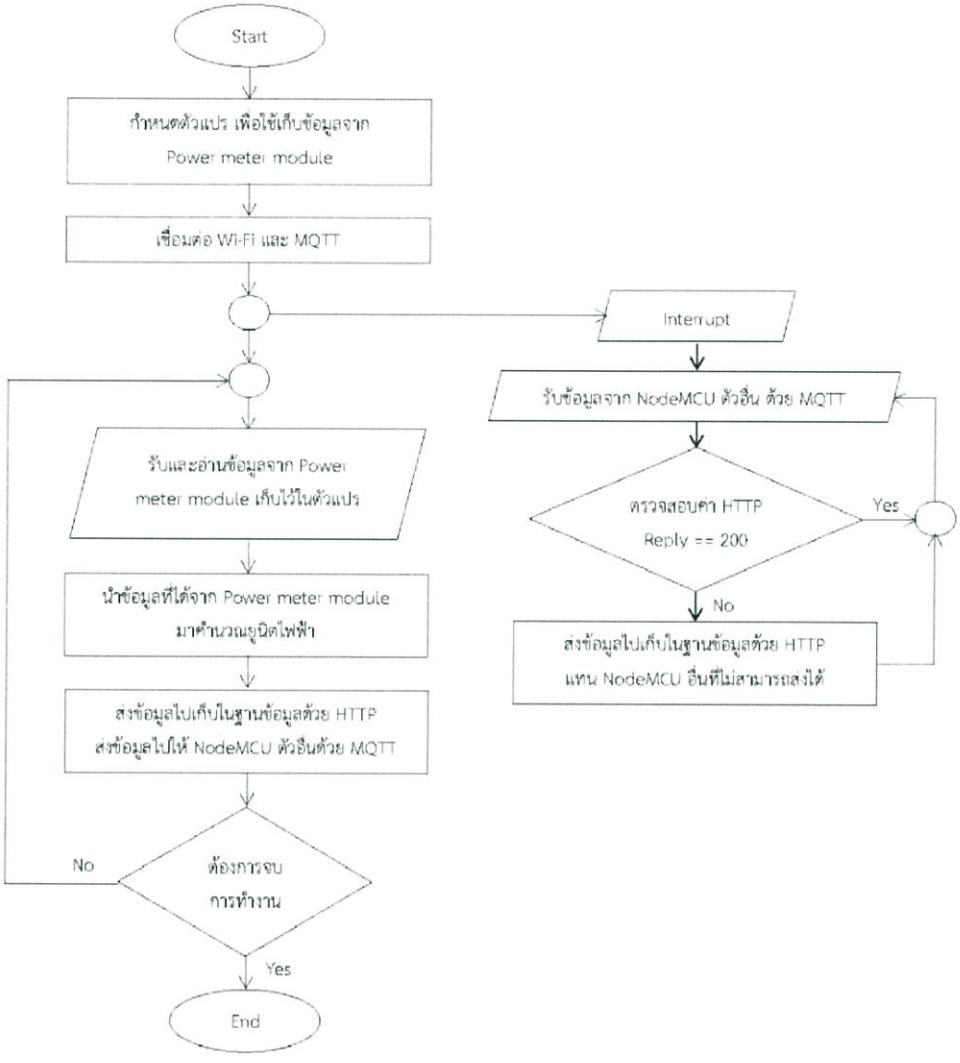
3.1.2 แผนผังการทำงานของระบบการแสดงผลและการควบคุม



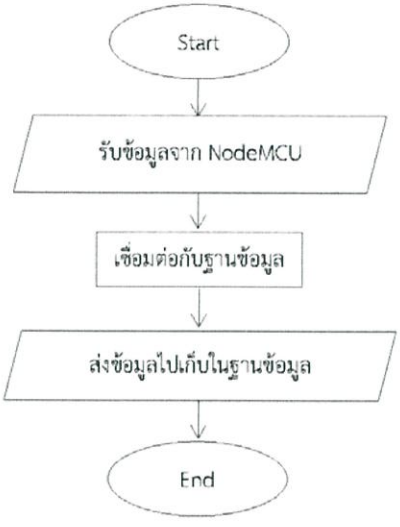
รูปที่ 3.2 แผนผังการทำงานโดยรวมของระบบ



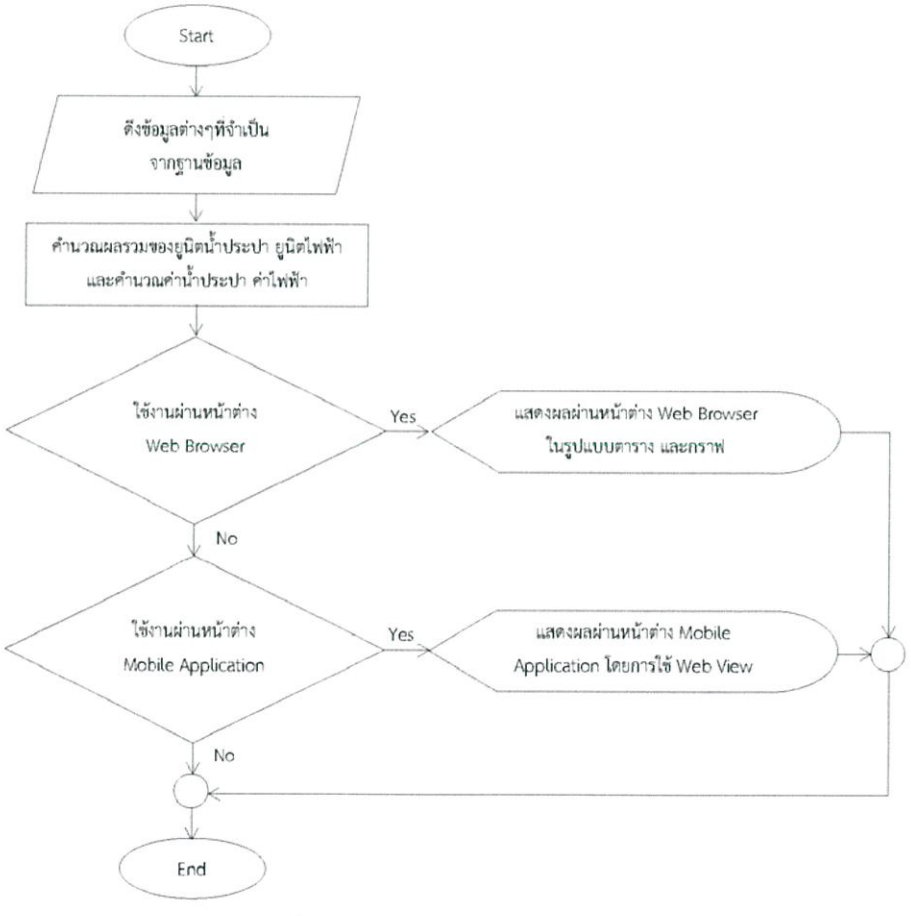
รูปที่ 3.3 แผนผังการทำงานของระบบอ่านและส่งยูนิตน้ำประปา



รูปที่ 3.4 แผนผังการทำงานของระบบอ่านและส่งยูนิตไฟฟ้า



รูปที่ 3.5 แผนผังการทำงานของระบบการส่งข้อมูลเข้าฐานข้อมูล



รูปที่ 3.6 แผนผังการแสดงผล

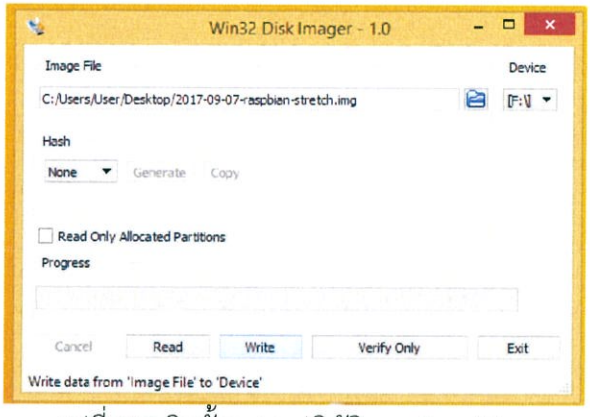
3.1.3 การตั้งค่า Raspberry Pi 3

3.1.3.1 ติดตั้งระบบปฏิบัติการ Raspbian ลงบน Micro SD card

1) ดาวน์โหลด OS Raspbian และ Win32DiskImager

2) แตกไฟล์ของ Raspbian แล้ว Run Win32DiskImager

3) เลือก Device เป็น Port Micro SD Card และเลือก Image File เป็น Raspbian



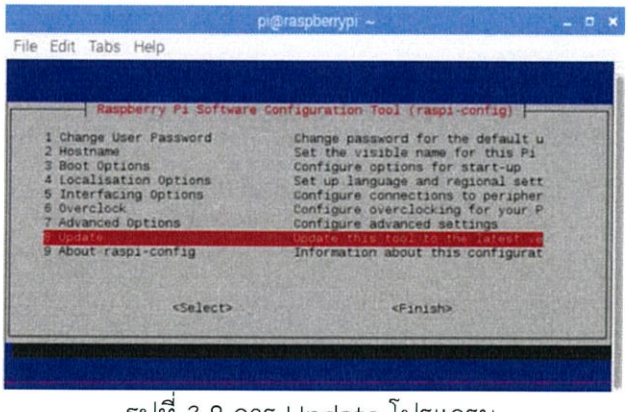
รูปที่ 3.7 ติดตั้งระบบปฏิบัติการ Raspbian

4) กด Write รอจนเสร็จ แล้วกด YES เป็นการจบกระบวนการ

3.1.3.2 Update และ Upgrade โปรแกรม

1) เปิด Terminal และพิมพ์คำสั่ง sudo raspi-config

2) เลือก Update แล้วกด Select



รูปที่ 3.8 การ Update โปรแกรม

3) รอจนเสร็จ สิ้นสุดขั้นตอนการ Update

4) พิมพ์คำสั่ง sudo apt-get upgrade

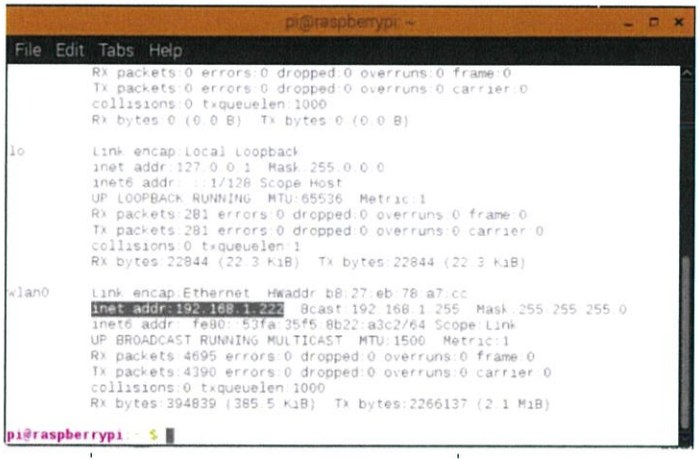
รูปที่ 3.9 การ Upgrade โปรแกรม

5) จะขึ้นคำถามว่า Do you want to continue? แล้วกด y และ Enter เพื่อยืนยัน

6) รอจนเสร็จ สิ้นสุดขั้นตอนการการ Upgrade

3.1.3.3 การตั้งค่า Static IP Address ของ Raspberry Pi

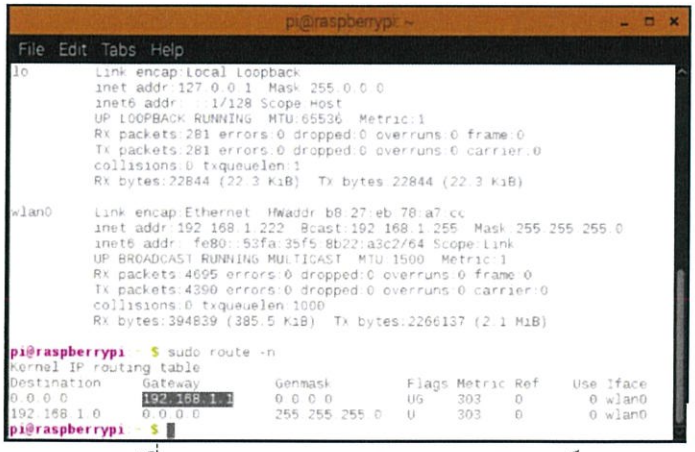
1) ใช้ Terminal พิมพ์คำสั่ง : sudo ifconfig



รูปที่ 3.10 การดู IP Address ของเครื่อง User ปัจจุบัน

จากรูปที่ 3.10 เป็นการใช้คำสั่ง sudo ifconfig เพื่อดูว่า IP Address ของเครื่อง User ปัจจุบันเป็น IP Address อะไร

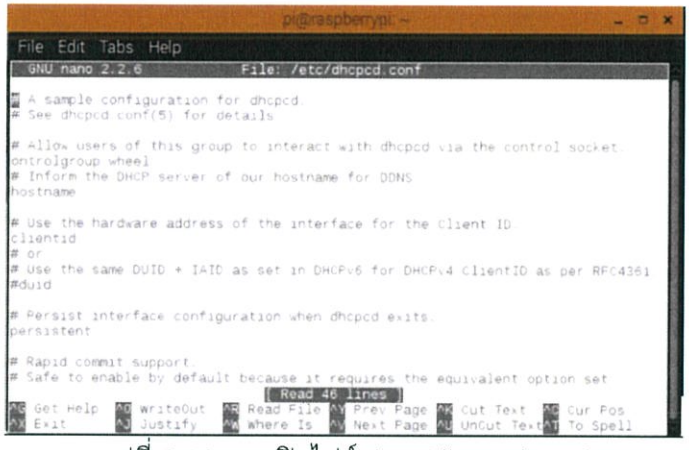
2) พิมพ์คำสั่ง : sudo route –n



รูปที่ 3.11 การดู IP Address ของเราเตอร์

จากรูปที่ 3.11 เป็นการใช้คำสั่ง sudo route –n เพื่อดูว่า IP Address ของเราเตอร์เป็น IP Address อะไร เพื่อจะใช้กำหนดใน static routers และ static domain_name_server

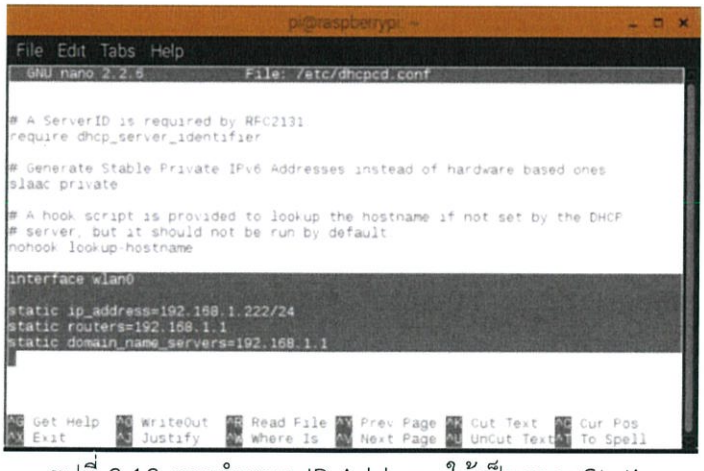
3) พิมพ์คำสั่ง : sudo nano /etc/dhcpcd.conf



รูปที่ 3.12 การเปิดไฟล์ /etc/dhcpcd.conf

จากรูปที่ 3.12 เป็นการใช้คำสั่ง sudo nano /etc/dhcpcd.conf เพื่อเปิดไฟล์ /etc/dhcpcd.conf ขึ้นมา เพื่อที่จะทำการกำหนด IP Address ให้เป็นแบบ Static ตามที่ต้องการ

4) เลื่อนลงมาที่บรรทัดสุดท้ายสุดของการเขียนโปรแกรม แล้วเพิ่มบรรทัดใหม่เข้าไปดังรูป พิมพ์เสร็จกด ctrl+x แล้ว Save โดยพิมพ์ตัว y แล้วกด Enter

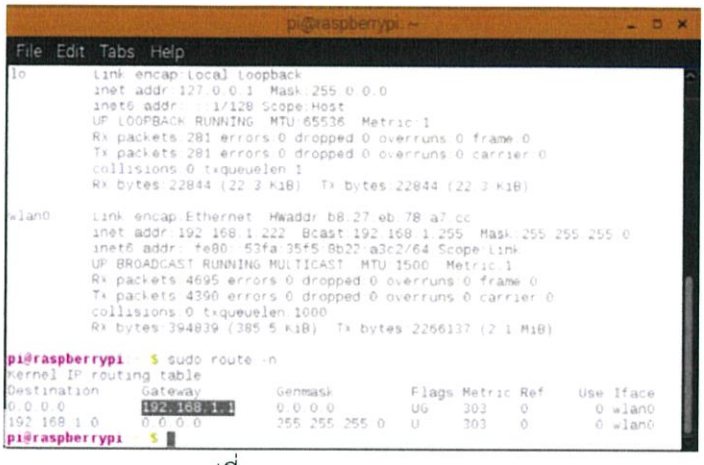


รูปที่ 3.13 การกำหนด IP Address ให้เป็นแบบ Static

จากรูปที่ 3.13 เป็นการเพิ่มบรรทัดใหม่เข้าไปดังรูป เพื่อทำการกำหนด IP Address เป็น Static ตามที่ต้องการโดยกำหนด เป็น 192.168.1.222

5) พิมพ์คำสั่ง : sudo reboot เพื่อทำการรีสตาร์ท

6) เมื่อ reboot เสร็จเปิด Terminal ขึ้นมาแล้วพิมพ์คำสั่ง : ifconfig

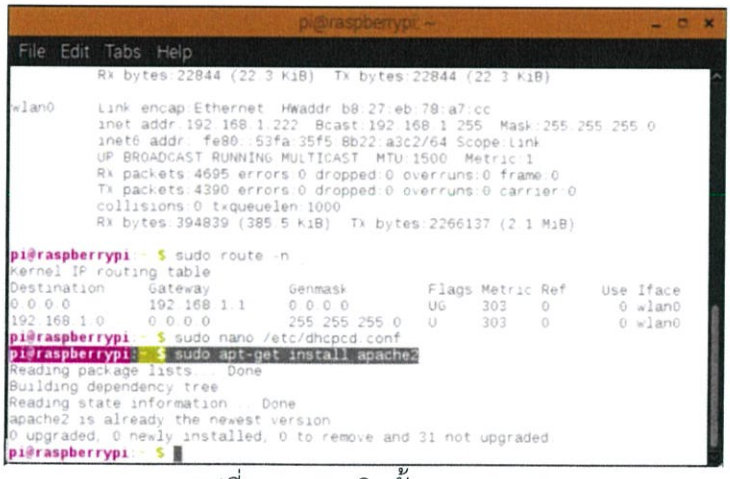


รูปที่ 3.14 การดู IP Address

จากรูปที่ 3.14 เป็นการใช้คำสั่ง ifconfig เพื่อตรวจสอบ IP Address ว่าเป็นไปตามที่กำหนดแล้วหรือไม่ โดยในรูป เป็น 192.168.1.222 ซึ่งเป็นไปตามที่กำหนดไว้

3.1.3.4 การติดตั้งโปรแกรมบน Raspberry Pi 3 เพื่อให้ทำงานเป็นเว็บเซิร์ฟเวอร์

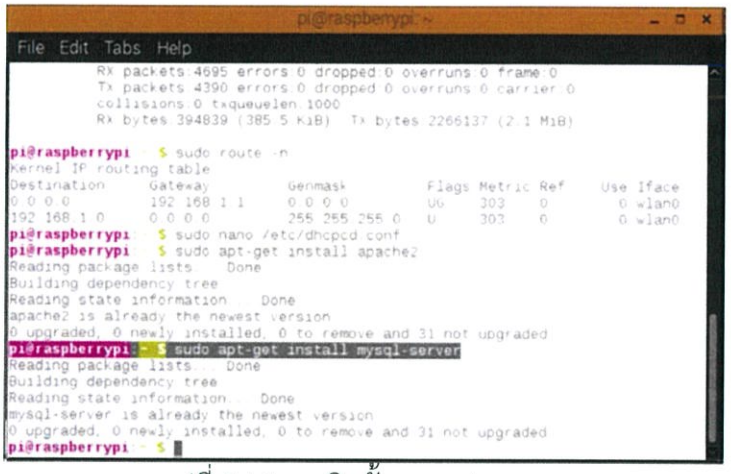
1) ใช้ Terminal พิมพ์คำสั่ง : sudo apt-get install apache2



รูปที่ 3.15 การติดตั้ง apache2

จากรูปที่ 3.15 เป็นการใช้คำสั่ง sudo apt-get install apache2 เพื่อติดตั้งโปรแกรม Apache2 ซึ่งทำหน้าที่เป็นเว็บเซิร์ฟเวอร์

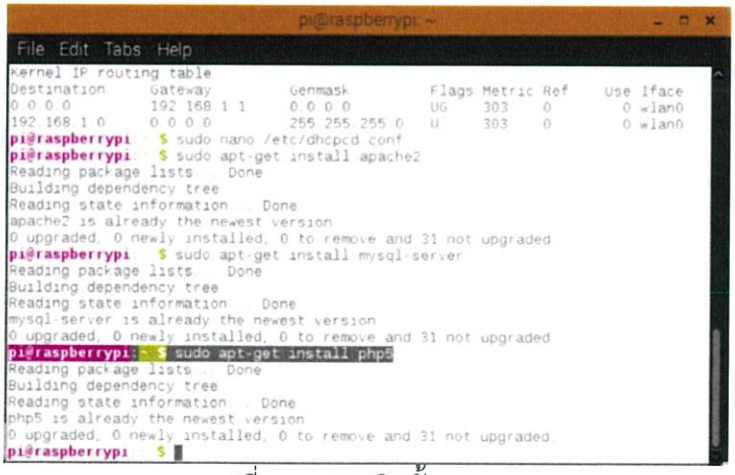
2) ใช้ Terminal พิมพ์คำสั่ง : sudo apt-get install mysql-server



รูปที่ 3.16 การติดตั้ง mysql-server

จากรูปที่ 3.16 เป็นการใช้คำสั่ง sudo apt-get install mysql-server เพื่อติดตั้งโปรแกรม MySQL ซึ่งทำหน้าที่เป็นตัวจัดการฐานข้อมูลทำงานร่วมกับเว็บเซิร์ฟเวอร์

3) พิมพ์คำสั่ง : sudo apt-get install php5



รูปที่ 3.17 การติดตั้ง PHP 5

จากรูปที่ 3.17 เป็นการใช้คำสั่ง sudo apt-get install php5 เพื่อติดตั้ง PHP verson5 ซึ่งเป็นภาษาคอมพิวเตอร์ชนิดหนึ่ง เพื่อใช้ติดต่อกับโปรแกรมจัดการฐานข้อมูล เพื่อทดสอบการติดตั้ง PHP โดยการเขียน script ของ index.php เข้าไปแทนที่ index.html ใน Directory ที่ใช้เป็นเว็บเซิร์ฟเวอร์ โดยจะอยู่ที่ /var/www/html โดยพิมพ์คำสั่งดังนี้

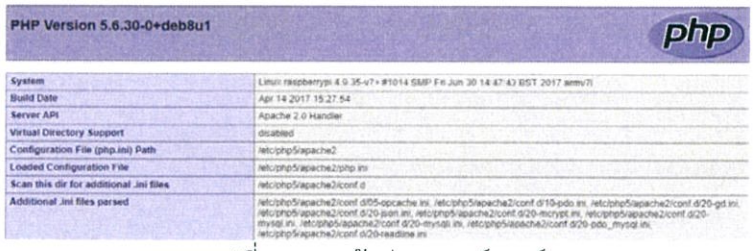
1) cd /var/www/html (เปลี่ยน Directory)

2) sudo rm index.html (ลบไฟล์ index.html)

3) sudo nano index.php (สร้างไฟล์ index.phpและเข้าไปเขียน script)

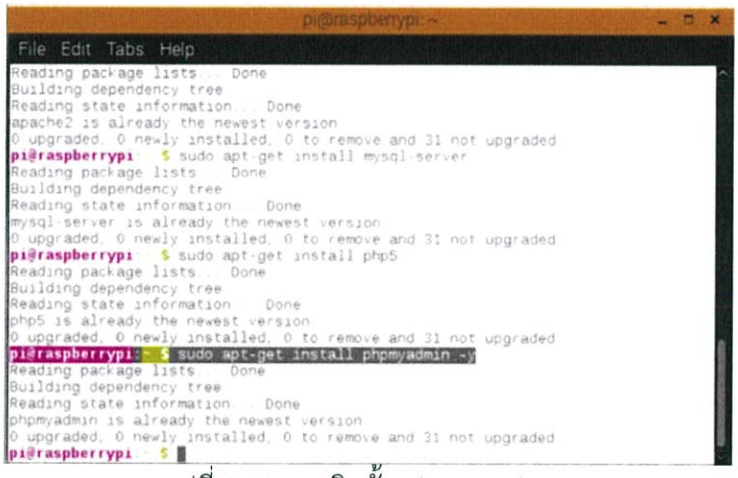
4) เขียน script ในไฟล์ index.php เป็น <?php phpinfo(); ?>

5) เปิดเบราว์เซอร์ พิมพ์ IP Address ของ Raspberry Pi



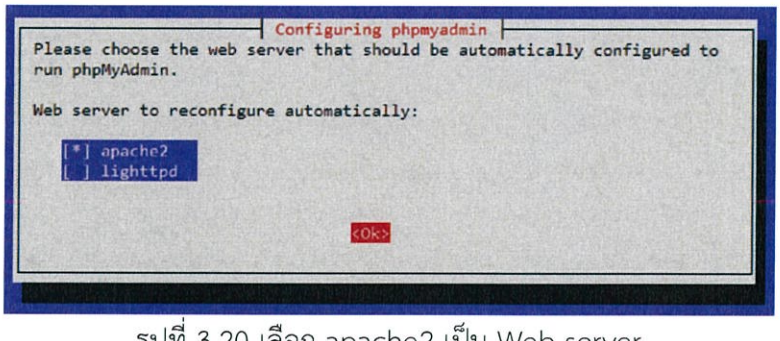
รูปที่ 3.18 หน้าต่างเบราว์เซอร์

6) พิมพ์คำสั่ง : sudo apt-get install phpmyadmin



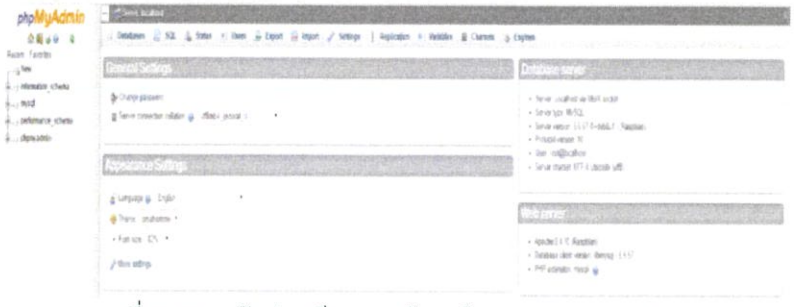
รูปที่ 3.19 การติดตั้ง phpMyAdmin

จากรูปที่ 3.19 เป็นการใช้คำสั่ง sudo apt-get install phpmyadmin เพื่อติดตั้งโปรแกรม phpMyAdmin ซึ่งใช้ในการบริหารจัดการฐานข้อมูล MySQL โดยลดความซับซ้อนลง



รูปที่ 3.20 เลือก apache2 เป็น Web server

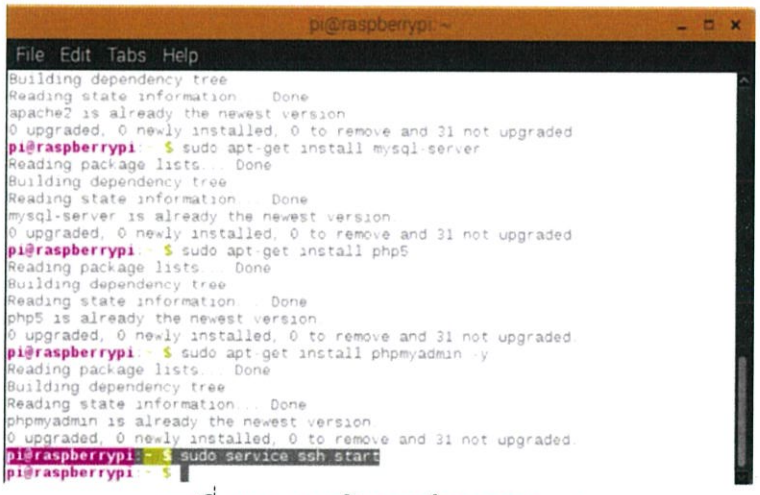
หลังจากติดตั้งสำเร็จสามารถตรวจสอบได้โดยเปิดเบราว์เซอร์ พิมพ์ IP Address ของ Raspberry Pi ตามด้วย /phpmyadmin จากนั้นทำการ login จะได้



รูปที่ 3.21 หน้าต่างเว็บเบราว์เซอร์ของ phpMyAdmin

3.1.3.5 การเปิดใช้งาน Service ssh

1) ใช้ Terminal พิมพ์คำสั่ง : sudo service ssh start

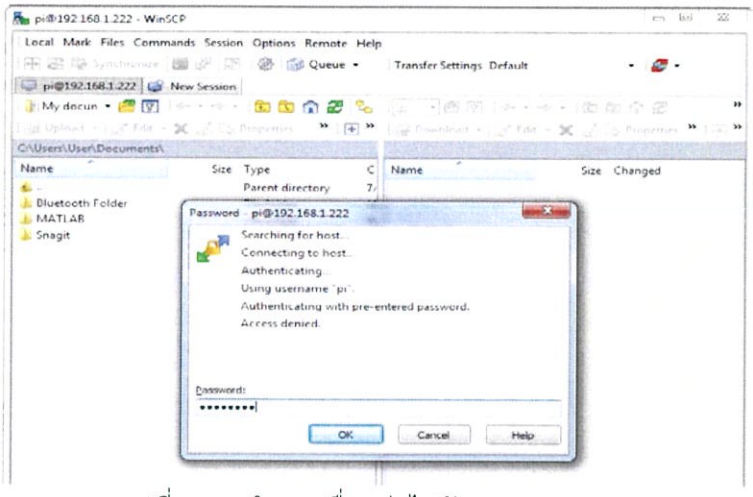


รูปที่ 3.22 การเปิดการทำงานของ ssh

จากรูปที่ 3.22 เป็นการใช้คำสั่ง sudo service ssh start เพื่อเป็นการเข้าถึงเครื่องเซิร์ฟเวอร์ โดยจะจัดการและควบคุมการดำเนินงานต่าง ๆ

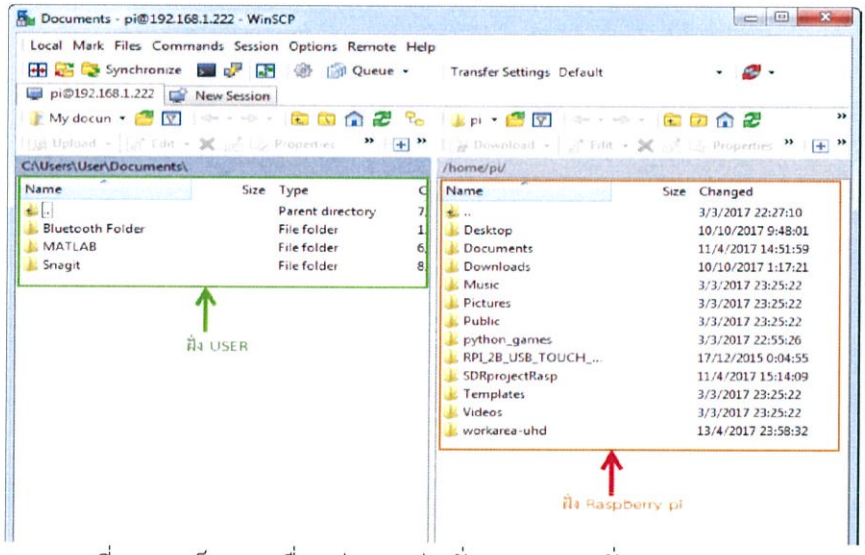
2) ทำการใส่ Host name ที่เป็นของเครื่อง Raspberry Pi ซึ่ง Port ที่จะทำการติดต่อเป็น Portที่ 22 และทำการใส่ User name, Password ของ Raspberry Pi

3) ทำการใส่รหัสของ Raspberry Pi และระบบก็ทำการเชื่อมต่อไปยัง Raspberry Pi



รูปที่ 3.23 ทำการเชื่อมต่อไปยัง Raspberry Pi

4) ทำการรับส่งข้อมูลระหว่าง User และ Raspberry Pi ได้

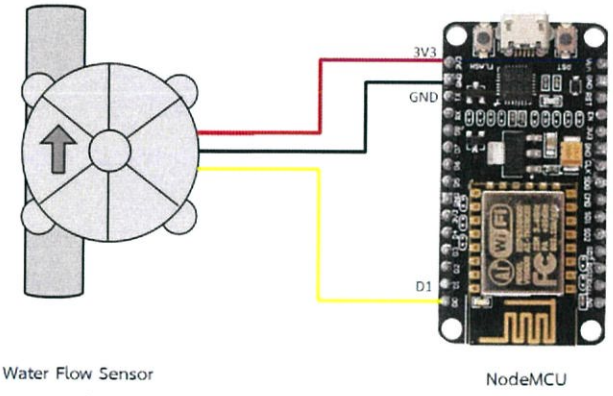


รูปที่ 3.24 เป็นการเชื่อมต่อระหว่างฝั่ง User และฝั่ง Raspberry Pi

SSH ย่อมาจาก Secure Shell คือ Network Protocol ที่สามารถแลกเปลี่ยนข้อมูลโดยช่องทางที่ปลอดภัย (Secure Channel) ระหว่างอุปกรณ์เครือข่ายสองตัว ซึ่ง SSH ได้รับการออกแบบให้มาแทนการ Telnet, Rlogin, RSH (The remote shell) ด้วยเหตุผลทางด้านความปลอดภัย การส่งข้อมูลจะอยู่ในรูปแบบตัวอักษรที่มีการเข้ารหัสข้อมูล(Encryption) เพื่อให้ข้อมูลเป็นความลับและให้สามารถส่งข้อมูลผ่านเครือข่ายอินเทอร์เน็ตได้อย่างสมบูรณ์ สามารถใช้งาน SSH ได้บนระบบปฏิบัติการ UNIX, Microsoft Windows, Apple Mac และ Linux

3.1.4 วัดอัตราการไหลของน้ำด้วย Water flow sensor

3.1.4.1 การต่อขาวงจรของ Water flow sensor เข้ากับ NodeMCU



รูปที่ 3.25 การต่อขาวงจรของ Water flow sensor เข้ากับ NodeMCU

จากรูปที่ 3.25 เป็นการต่อขาวงจรของ Water flow sensor เข้ากับ NodeMCU การต่อวงจรนั้น ซึ่งจะป้อนไฟเลี้ยงให้กับเซ็นเซอร์ผ่านสาย +5V ที่แรงดัน 5V สาย GND เป็นกราวน์และสาย O/P เป็นสัญญาณพัลส์ขาออก ซึ่งข้อมูลที่ได้จากสัญญาณ จะอยู่ในรูปของดิจิทัลพัลส์ (Digital Pulse) ซึ่งในที่นี้กำหนดให้ใช้ขา D1 ของ NodeMCU เพื่อติดต่อกับ Water flow sensor

3.1.4.2 เขียนโปรแกรมการอ่านและส่งค่ายูนิตน้ำประปาจาก Water flow sensor ไปฐานข้อมูล

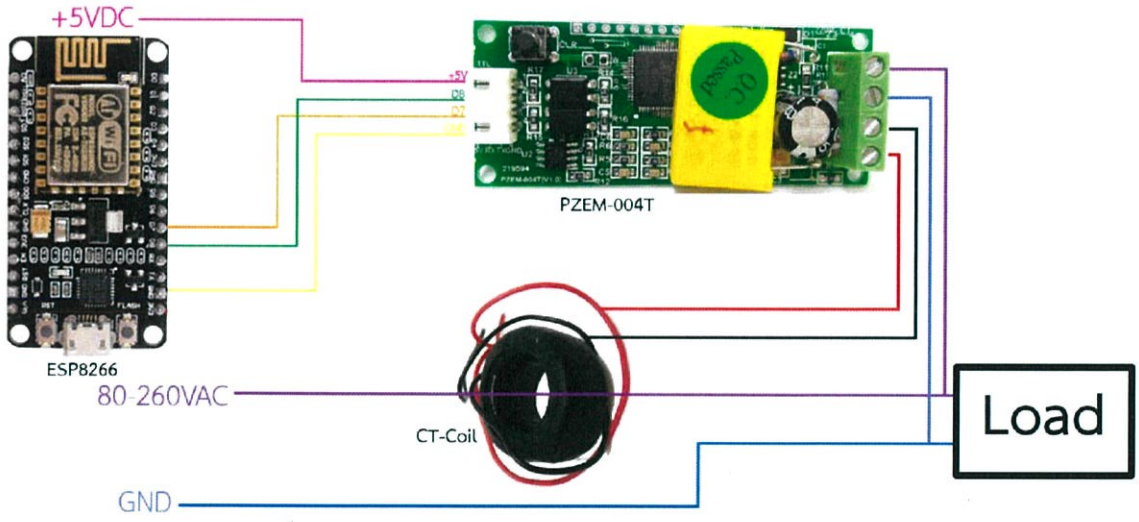
```
void loop()
{
  calc ();
  Serial.println("");
  if ((WiFiMulti.run() == WL_CONNECTED)) {
    HTTPClient http;
    Cal = ((Fan / 5.5) / 1000) / 60; //คำนวณยูนิตน้ำประปา
    Serial.print("Water unit = ");
    Serial.println(Cal, 6);
    url = "http://192.168.1.222/add2.php?home=1&waterunit=" + String(Cal, 6);
    http.begin(url); //HTTP
    httpCode = http.GET();
    if (httpCode > 0) {
      Serial.printf("[HTTP] GET... code: %d\n", httpCode);
    } else {
      Serial.printf("[HTTP] GET... failed, error: %s\n", http.errorToString(httpCode).c_str());
    }
    http.end();
  }
}
```

รูปที่ 3.26 เขียนโปรแกรมแสดงการอ่านและส่งค่าอัตราการไหลของน้ำ

จากรูปที่ 3.26 เป็นการเขียนโปรแกรมการอ่านและส่งค่ายูนิตของน้ำประปาไปเก็บยังฐานข้อมูล ซึ่งจะเปลี่ยนสัญญาณที่อยู่ในรูปอัตราของของไหล ให้มาอยู่ในรูปแบบสัญญาณของดิจิทัลพัลส์ (Digital Pulse)เพื่อที่จะส่งสัญญาณเหล่านี้ไปยัง NodeMCU ต่อไป ซึ่งความถี่สัญญาณที่ออกมาจะเปลี่ยนตามความเร็วการไหลของน้ำ โดยเซ็นเซอร์ปล่อยสัญญาณออกมา 1 พัลส์ เมื่อใบพัดภายในตัวมิเตอร์หมุนไป 1 รอบ ดังนั้นหลักการวัดอัตราการไหลน้ำจึงใช้การนับจำนวนพัลส์ที่ถูกส่งออกมาใน 1 หน่วยเวลา สำหรับรุ่น G3/4 Water flow sensor ที่อัตราการไหล 1 L/min สัญญาณที่ออกมามีความถี่ 5.5 Hz ซึ่งสูตรในการคำนวณอัตราการไหลเฉลี่ยเป็นดังสมการที่ 2.3 จากนั้นนำสมการมาใช้ในการคำนวณยูนิตน้ำประปาแล้วส่งไปยังฐานข้อมูล

3.1.5 วัดอัตราการไหลของกระแสไฟฟ้าด้วย Power meter module

3.1.5.1 การต่อขาวงจรของ PZEM-004T เข้ากับ NodeMCU



รูปที่ 3.27 การต่อโมดูล PZEM-004T กับ NodeMCU

จากรูปที่ 3.27 เป็นการต่อโมดูล PZEM-004T กับ ESP8266 NodeMCU โดยจะพิจารณาเป็น 2 ด้านคือด้านการรับส่งข้อมูลกับ NodeMCU และด้านการวัดพลังงานไฟฟ้า โดยจะพิจารณาที่ด้านการรับส่งข้อมูลกับ NodeMCU ก่อน เริ่มจากทำการป้อนไฟเลี้ยง 5 VDC ในขา VDD ของโมดูล PZEM-004T ต่อขา RX ของโมดูล PZEM-004T เข้ากับขา และต่อขา TX ของโมดูล PZEM-004T เข้ากับขา D7 เพื่อรับส่งข้อมูล และต่อกราวด์ของโมดูล PZEM-004T เข้ากับขากราวด์ของ NodeMCU ต่อมาพิจารณาในส่วนของการวัดพลังงานไฟฟ้าจะวัดแรงดันไฟฟ้าผ่านขา AC ซึ่งเป็นการต่อคร่อมโหลด และจะวัดกระแส CT coil โดยใช้ขา CT

3.1.5.2 เขียนโปรแกรมเพื่อการอ่าน และส่งค่ายูนิตไฟฟ้าจาก PZEM-004T ไปยังฐานข้อมูล

```
void loop() {
calc ();
  Serial.println("");
  if ((WiFiMulti.run() == WL_CONNECTED)) {
    HTTPClient http;
    unit = ((Sumwatt) * 2.2) / 3600000; //(sent every 1 min) : unit/min;
    Serial.print("Power unit = ");
    Serial.println(unit, 6);
    url = "http://192.168.1.222/add.php?home=1&powerunit=" + String(unit, 6);
    http.begin(url); //HTTP
    httpCode = http.GET();
    if (httpCode > 0) {
      Serial.printf("[HTTP] GET... code: %d\n", httpCode);
    } else {
      Serial.printf("[HTTP] GET... failed, error: %s\n", http.errorToString(httpCode).c_str());
    }
    http.end();
  }
  Sumwatt = 0.0;
  k = 0;
  Serial.println("");
  }
```

รูปที่ 3.28 เขียนโปรแกรมที่ใช้ในการอ่าน และส่งค่าอัตราการใช้ไฟฟ้า

จากรูปที่ 3.28 เป็นการเขียนโปรแกรมเพื่อใช้ในการอ่านและส่งค่าอัตราการใช้ไฟฟ้าไปเก็บยังฐานข้อมูล โดยจะอ่านและส่งข้อมูลไปยังฐานข้อมูลทุกๆ 5 วินาที ซึ่งจะส่งค่าแรงดันไฟฟ้า กระแสไฟฟ้าที่ใช้กำลังไฟฟ้า และจำนวน Unit ที่ได้จากการคำนวณดังนี้

$$\text{Unit} = \text{Power} \times \text{Time} \qquad (3.2)$$

โดยให้ Unit คือ หน่วยของการคิดค่าพลังงานไฟฟ้า มีหน่วยเป็น kW-Hour

Power คือ กำลังไฟฟ้า มีหน่วยเป็น kW

Time คือ เวลาที่ใช้ไฟฟ้า หน่วยเป็น ชั่วโมง

3.1.6 เขียน PHP เพื่อรับค่าจาก NodeMCU ไปเก็บยังฐานข้อมูล

เขียนโปรแกรมของ PHP และ MySQL เพื่อเป็นตัวกลาง สำหรับรับข้อมูลจาก ESP8266 ซึ่งเป็น NodeMCU ที่ใช้ส่งข้อมูลเพื่อนำมาบันทึกลงในฐานข้อมูล

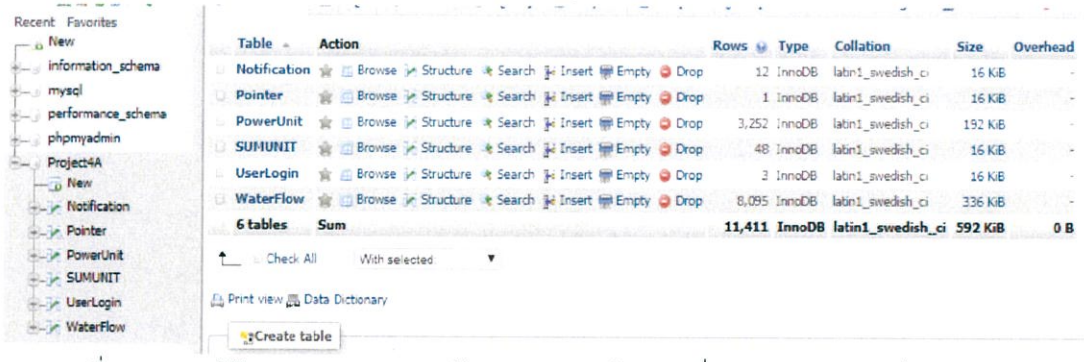
3.1.6.1 เขียนโปรแกรมด้วยภาษา PHP เพื่อเพิ่มข้อมูลลงบนฐานข้อมูล

```
<?php
$home = $_GET['home'];
$waterunit = $_GET['waterunit'];

 mysql_connect("localhost", "root", "12345671") or die(mysql_error());
 mysql_select_db("Project4A") or die(mysql_error());

 $data = mysql_query("INSERT INTO WaterFlow(home,unit) VALUES ($home,$waterunit)") or die(mysql_error());
?>
```

รูปที่ 3.29 เขียนโปรแกรมโดยใช้ภาษา PHP เพื่อเพิ่มข้อมูลเข้าสู่ฐานข้อมูล

จากรูปที่ 3.29 เป็นการเพิ่มข้อมูลเข้าฐานข้อมูลโดยใช้คำสั่ง GET รอรับข้อมูลจาก NodeMCU และใช้คำสั่ง insert ของ MySQL ในการเพิ่มข้อมูลนั้นลงในตารางฐานข้อมูล

3.1.6.2 สร้างตารางฐานข้อมูล



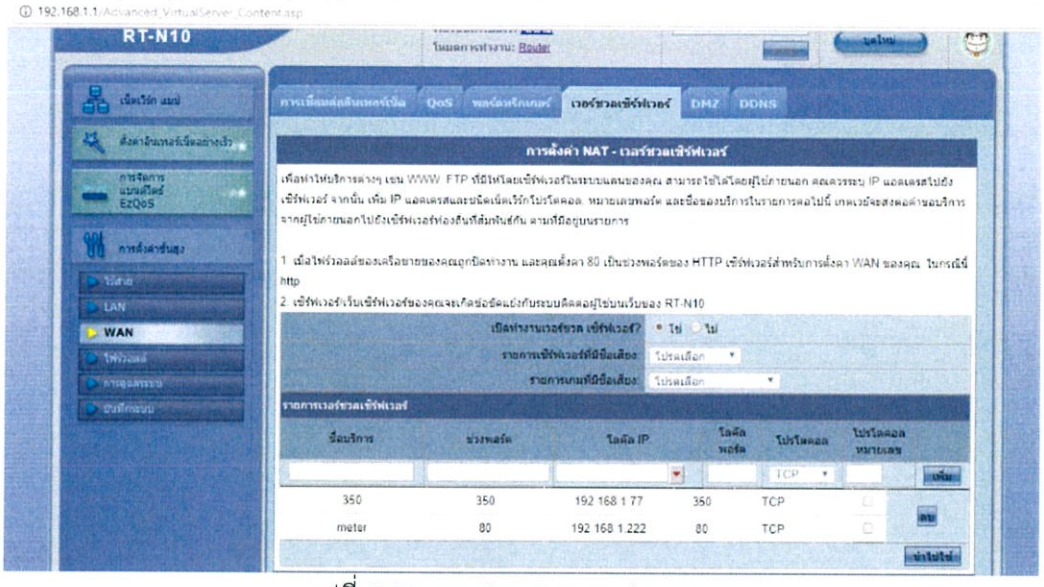
รูปที่ 3.30 การใช้ phpMyAdmin สร้างตารางฐานข้อมูล เพื่อจัดการและจัดเก็บฐานข้อมูล

จากรูปที่ 3.30 เป็นการสร้างตารางฐานข้อมูล เพื่อให้ง่ายในการจัดการและจัดเก็บฐานข้อมูล ทำให้ผู้ดูแลสามารถนำข้อมูลจากฐานข้อมูลไปใช้ได้อย่างถูกต้อง โดยออกแบบให้มีตารางดังนี้

1) UserLogin เพื่อเก็บข้อมูล username password level ของผู้ใช้งาน
2) Pointer เพื่อตรวจสอบผู้ใช้
3) WaterFlow เพื่อเก็บข้อมูลยูนิตของน้ำประปาและเวลา
4) PowerUnit เพื่อเก็บข้อมูลยูนิตของไฟฟ้า และเวลา
5) SUMUNIT เพื่อเก็บข้อมูลผลรวมของยูนิตในแต่ละเดือน
6) Notification เพื่อเก็บข้อมูลการแจ้งเตือน

3.1.7 การทำให้ผู้ใช้สามารถเข้ามาใช้งานระบบจากเครือข่ายภายนอกได้

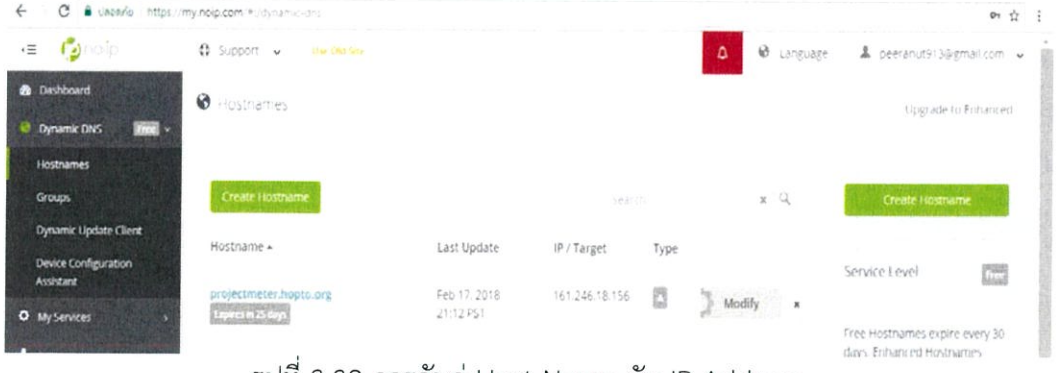
1) การ Set Forwarding Port



รูปที่ 3.31 การ Set Forwarding Port

จากรูปที่ 3.31 เป็นการ Set Forwarding Port ของเราเตอร์ยี่ห้อ ASUS RT-N10 เพื่อเรียกใช้บริการเว็บเซิร์ฟเวอร์ที่อยู่ในระบบ LAN ของจากเครือข่ายภายนอก โดยกำหนด Port ที่ใช้งานอยู่ที่ Port 80 ซึ่งใช้เป็นโปรโตคอลของ HTTP และกำหนด Local IP ของ Raspberry Pi ที่ทำหน้าที่เป็นเว็บเซิร์ฟเวอร์เป็น 192.168.1.222 ซึ่งจะทำให้สามารถเข้าถึงเว็บเซิร์ฟเวอร์ได้ โดยไม่ต้องอยู่ในระบบ LAN เดียวกัน สามารถเข้าถึงได้โดย Public IP ของเราเตอร์ที่ Port 80

1) การจับคู่ Host Name กับ IP Address



รูปที่ 3.32 การจับคู่ Host Name กับ IP Address

จากรูปที่ 3.32 เป็นการจับคู่ Host Name กับ IP Address ของเราเตอร์โดยใช้บริการ DDNS (Dynamic Domain Name Service) เพื่อให้ผู้ใช้สามารถเข้าถึงระบบได้โดยใช้ชื่อ Host Name ซึ่งใช้ชื่อว่าProjectmeter.hopto.org เข้ากับ IP Public ของเราเตอร์

3.1.8 การสร้างการสื่อสารระหว่าง NodeMCU

ในปริญญานิพนธ์ฉบับนี้ใช้โปรโตคอล MQTT ในการสื่อสารกันระหว่าง NodeMCU ด้วยกัน ทดลองโดยใช้ NodeMCU 2 ตัว โดย NodeMCU ทั้งสองตัวอยู่ในเครือข่ายเดียวกัน กำหนดให้อุปกรณ์ตัวหนึ่งเป็น Publisher เพื่อส่งข้อมูลโดยกำหนด Topic ที่ต้องการและ Broker ทำหน้าที่เป็นตัวกลางคอยจัดการกับข้อความโดยอ้างอิงจาก Topic อุปกรณ์อีกตัวหนึ่งเป็น Subscriber ซึ่งจะคอยดูการเปลี่ยนแปลงของข้อความที่อ้างอิงด้วย Topic

```
  /*
     Register the callback
  */
  MQTT_server_onData(data_callback);

  /*
     Start the broker
  */
  Serial.println("Starting MQTT broker");
  MQTT_server_start(mqttPort, max_subscriptions, max_retained_topics);

  /*
     Subscribe to anything
  */
  MQTT_local_subscribe((unsigned char *)"#", 0);
}

int counter = 0;

void loop()
{
  String myData(counter++);

  /*
     Publish the counter value as String
  */
  MQTT_local_publish((unsigned char *)"/MyBroker/count", (unsigned char *)
myData.c_str(), myData.length(), 0, 0);

  // wait a second
  delay(1000);
}
```

รูปที่ 3.33 ทำ NodeMCU เป็น Publisher

จากรูปที่ 3.33 เป็นการเขียนโปรแกรมผ่าน Arduino IDE โดยใช้โปรโตคอล MQTT ทำให้ NodeMCU เป็น Publisher เพื่อส่งข้อมูลไปยัง Topic ที่กำหนดไว้

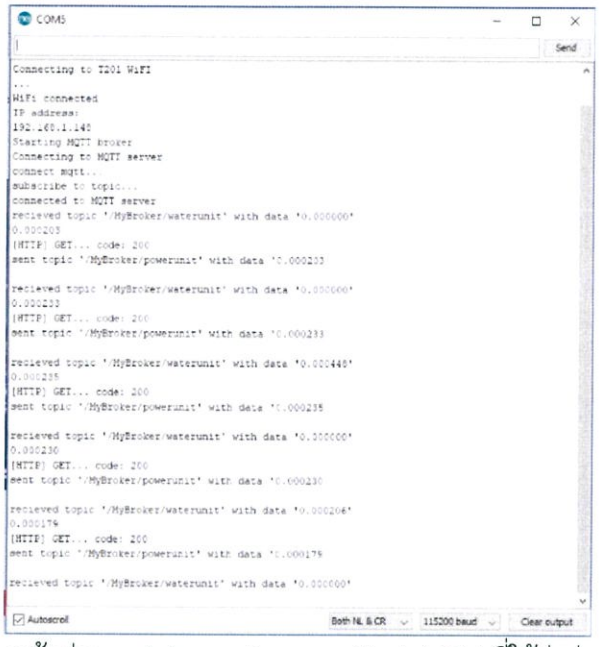
```
void loop() {
}

void myConnectedCb()
{
  Serial.println("connected to MQTT server");
}

void myDisconnectedCb()
{
  Serial.println("disconnected. try to reconnect...");
  delay(500);
  myMqtt.connect();
}

void myPublishedCb()
{
  //Serial.println("published.");
}

void myDataCb(String& topic, String& data)
{

  Serial.print("recieved topic '");
  Serial.print(topic);
  Serial.print("' with data '");
  Serial.println(data);
}
```

รูปที่ 3.34 ทำ NodeMCU เป็น Subscriber

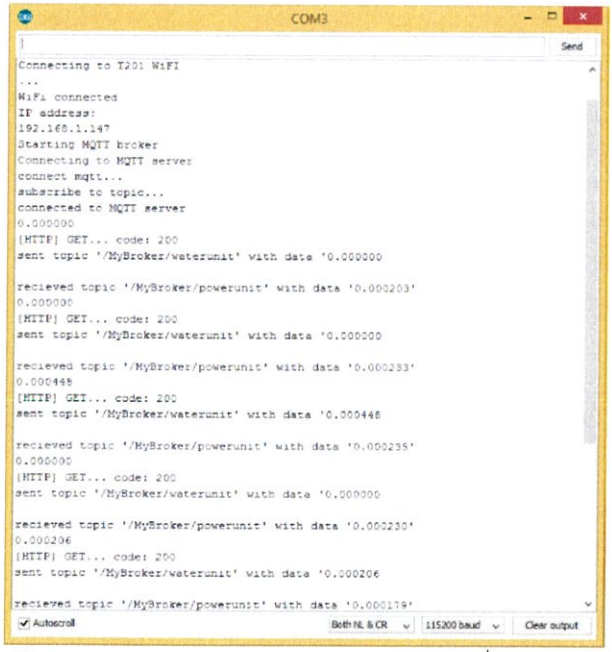
จากรูปที่ 3.34 เป็นการเขียนโปรแกรมผ่าน Arduino IDE โดยใช้โปรโตคอล MQTT ทำให้ NodeMCU เป็น Subscriber เพื่อรับข้อมูลไปจาก Topic ที่กำหนดไว้

3.1.9 การรวมระบบการรับส่งข้อมูล MQTT

เมื่อสามารถทำให้ NodeMCU เป็น Publisher หรือ Subscriber ได้แล้วในลำดับต่อไปได้ทำการเขียนโปรแกรมเพื่อรวมระบบการรับส่งข้อมูลด้วยโปรโตคอล MQTT เข้าด้วยกัน ทำให้ NodeMCU แต่ละตัวเป็นทั้ง Publisher และ Subscriber ในตัวเดียวกัน และทำงานร่วมกับระบบเดิมที่มีการสื่อสารระหว่าง NodeMCU และ Raspberry Pi ในรูปแบบ Client Server ได้โดยแสดงการรับส่งข้อมูลผ่าน serial monitor

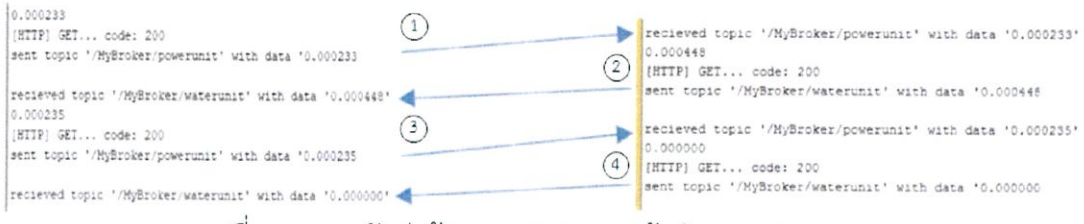


รูปที่ 3.35 หน้าต่าง serial monitor ของ NodeMCU ที่ใช้ส่งค่ายูนิตไฟฟ้า



รูปที่ 3.36 หน้าต่าง serial monitor ของ NodeMCU ที่ใช้ส่งค่ายูนิตน้ำประปา

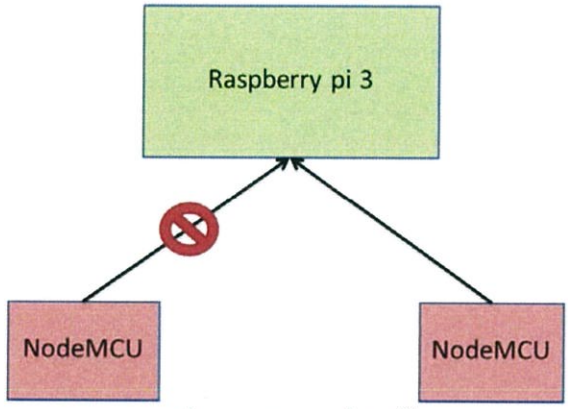
จากรูป 3.35 และ 3.36 แสดงหน้าต่าง serial monitor ของ NodeMCU แต่ละตัว ที่แสดงให้เห็นถึงการส่งข้อมูลยูนิตน้ำและยูนิตไฟฟ้าไปยัง Raspberry Pi ที่ทำหน้าที่เป็นฐานข้อมูล พร้อมทั้งส่งข้อมูลนั้นผ่านโปรโตคอล MQTT ด้วย โดยการส่งนั้นจะส่งแบบเป็น topic แสดงดังรูปที่ 3.37



รูปที่ 3.37 การรับส่งข้อมูล MQTT บนหน้าต่าง serial monitor

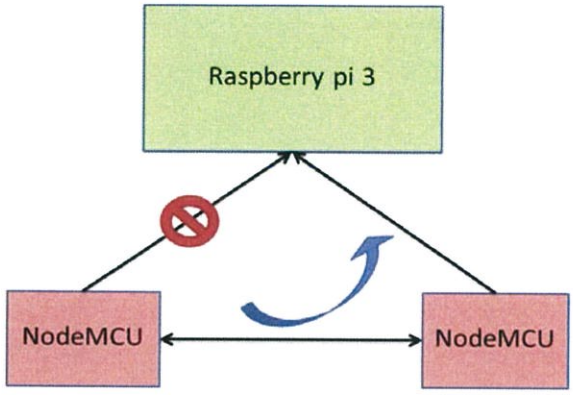
จากรูป 3.37 เป็นการแสดงการสื่อสารด้วยโปรโตคอล MQTT โดยเริ่มจากส่งข้อมูลค่ายูนิตไฟฟ้าที่คำนวณได้เท่ากับ 0.000233 ไปยัง topic '/MyBroker/powerunit' ดังลูกศรหมายเลข 1 จากนั้น NodeMCU อีกตัวจะได้รับค่ายูนิตไฟฟ้าจาก topic เดียวกันนั้น และทำการส่งข้อมูลยูนิตน้ำที่คำนวณได้เท่ากับ 0.000448 ไปยัง topic '/MyBroker/waterunit' จากนั้น NodeMCU อีกตัวจะได้รับค่ายูนิตไฟฟ้าจาก topic เดียวกันดังลูกศรหมายเลข2 และจะทำการรับส่งค่ายูนิตไฟฟ้าและค่ายูนิตน้ำระหว่างกันไปเรื่อยๆดังเช่นลูกศรหมายเลข 3 และ 4

3.1.10 การใช้ประโยชน์จากโปรโตคอล MQTT



รูปที่ 3.38 ปัญหาที่เกิดขึ้น

จากรูปที่ 3.38 แสดงถึงปัญหาที่เกิดขึ้นเมื่อ NodeMCU ไม่สามารถติดต่อกับฐานข้อมูลได้ เกิดขึ้นได้ในหลายกรณี และมีการตอบกลับจากโปรโตคอล HTTP เรียกว่า HTTP Reply Status Code ซึ่งทำให้ทราบถึงปัญหาที่เกิดขึ้นเช่น 4xx: Client Error แสดงถึงคำขอไม่ถูกต้องหรือไม่สามารถปฏิบัติตามได้ 5xx: Server Error แสดงถึงเซิร์ฟเวอร์ไม่สามารถตอบสนองคำขอที่ถูกต้องได้ ทำให้ข้อมูลเกิดความไม่ต่อเนื่องและขาดความน่าเชื่อถือ ดังนั้นสามารถใช้การสื่อสารด้วยโปรโตคอล MQTT ระหว่าง NodeMCU เข้ามาแก้ไขปัญหาได้ดังรูปที่ 3.39



รูปที่ 3.39 การแก้ไขปัญหาเมื่อ NodeMCU ตัวใดตัวหนึ่งไม่สามารถติดต่อกับฐานข้อมูลได้

จากรูปที่ 3.39 แสดงถึงการใช้ MQTT สื่อสารกันระหว่าง NodeMCU ทำให้สามารถใช้ NodeMCU ตัวอื่นส่งข้อมูลแทน NodeMCU ที่ไม่สามารถส่งข้อมูลไปยังฐานข้อมูลได้ โดยใช้การเขียนโปรแกรมลงบน NodeMCU ซึ่งทำให้สามารถตรวจสอบการส่งข้อมูลได้โดยอาศัย HTTP Reply Status Code นี้ในการตัดสินใจ เมื่อส่งข้อมูลด้วย HTTP สำเร็จ จะตอบรับเป็นเลข 200 ออกมา ดังนั้นถ้าตอบกลับไม่ใช่เลข 200 แสดงว่าไม่สามารถส่งข้อมูลไปยังฐานข้อมูลได้ จึงต้องใช้ NodeMCU ตัวอื่นในการส่งข้อมูลแทน โดยมีหลักการคือหลังจากที่ส่งข้อมูลไปยังฐานข้อมูล ฝั่ง Publisher จะส่งข้อมูลผ่านโปรโตคอล MQTT และแนบ HTTP Reply Status Code ไปด้วย เมื่อฝั่ง Subscriber ได้รับจะสามารถตัดสินใจได้ว่าจะส่งข้อมูลนั้นแทนหรือไม่

3.2 เครื่องมือที่ใช้ในการทดลอง

1) Power meter module : PZEM-004T AC digital display Multifunction Meter

2) Water flow sensor : G3/4 Water flow sensor

3) NodeMCU : ESP8266 CP2102 module

4) Raspberry Pi : Raspberry Pi 3

5) แบตเตอรี่สำรอง

3.3 การจัดเก็บผลการทดลอง

1) เก็บข้อมูลโดยใช้ NodeMCU ส่งข้อมูลไปเก็บใน Raspberry Pi โดยใช้ HTTP GET

2) เก็บข้อมูลโดยใช้ Raspberry เป็นฐานข้อมูล โดยมี phpMyAdmin เป็นตัวจัดการฐานข้อมูล

บทที่ 4

ผลการทดลอง

ปริญญานิพนธ์ฉบับนี้มีวัตถุประสงค์เพื่อใช้วัดปริมาณการใช้น้ำประปาและไฟฟ้าแบบไร้สาย ซึ่งจะนำไปแสดงผลบนหน้าต่างแอปพลิเคชัน เพื่อให้ผู้ใช้น้ำประปาและไฟฟ้าสามารถตรวจสอบและทราบถึงประมาณการใช้น้ำประปาและไฟฟ้าได้ตลอดเวลา ซึ่งเป็นการประมาณจำนวนหน่วยที่ใช้และจำนวนเงินที่ต้องจ่าย เพื่อให้ผู้ใช้เกิดความตระหนักถึงการใช้น้ำประปาและไฟฟ้า ซึ่งทำให้สามารถปรับเปลี่ยนพฤติกรรมการใช้น้ำประปาและไฟฟ้าเพื่อให้เกิดประโยชน์สูงสุดแก่ผู้ใช้

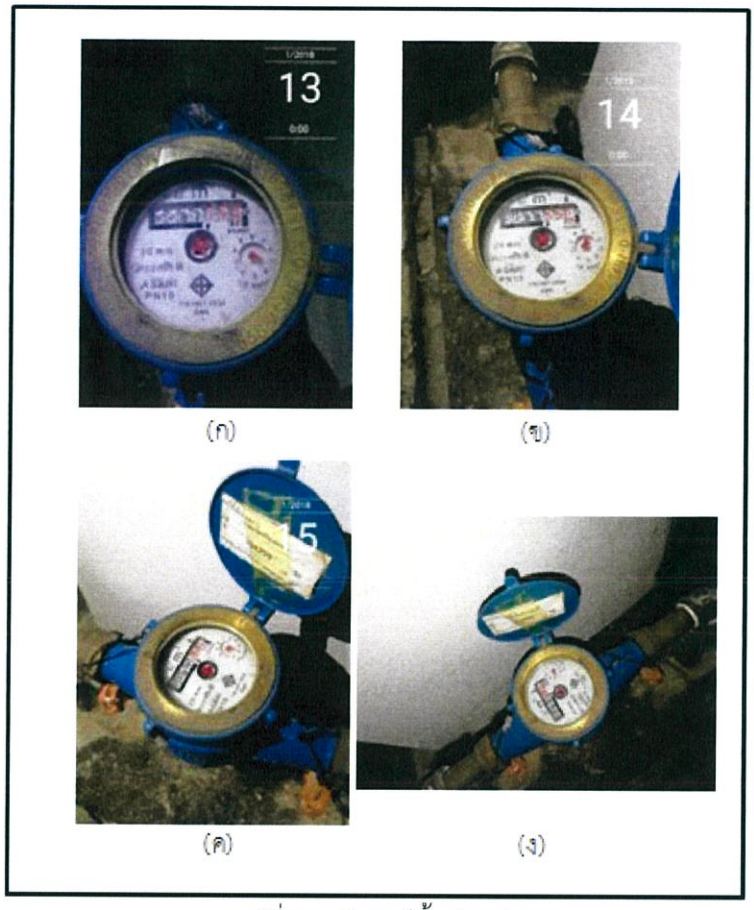
ในบทที่ 4 นี้จะเป็นการแสดงผลในขั้นตอนต่างๆของการดำเนินงานประกอบด้วย การทดสอบความถูกต้องของอุปกรณ์ การคำนวณค่ายูนิตของน้ำประปาและส่งไปยังฐานข้อมูล การนำข้อมูลจากฐานข้อมูลมาแสดงบนหน้าต่างเว็บเบราว์เซอร์ และการสร้าง Mobile Application ครอบหน้าต่างเว็บเบราว์เซอร์

4.1 การทดสอบความถูกต้องของอุปกรณ์

4.1.1 การทดสอบความถูกต้องของ Water flow sensor

เป็นการทดสอบความถูกต้องของค่ายูนิตน้ำประปาที่วัดได้จาก Water flow sensor เปรียบเทียบกับปริมาณน้ำที่ใช้จริงของมิเตอร์การประปานครหลวง เพื่อหาค่าความคลาดเคลื่อนของ Water flow sensor

1) เปรียบเทียบปริมาตรน้ำประปาที่วัดได้ภายใน 1 วัน ในหน่วยยูนิตของมิเตอร์น้ำการประปานครหลวง กับ Water flow sensor โดยทำการทดลอง 3 วัน



รูปที่ 4.1 มิเตอร์น้ำประปา

จากรูปที่ 4.1 แสดงมิเตอร์น้ำประปาตั้งแต่วันที่ 13/01/2018 ถึง 16/01/2018 โดยที่ (ก) แสดงมิเตอร์น้ำประปาในวันที่ 13/01/2018 (ข) แสดงมิเตอร์น้ำประปาในวันที่ 14/01/2018 (ค) แสดงมิเตอร์น้ำประปาในวันที่ 15/01/2018 (ง) แสดงมิเตอร์น้ำประปาในวันที่ 16/01/2018

2) ทำการบันทึกค่าที่อ่านได้จากมิเตอร์และหาค่าความคลาดเคลื่อน

ตารางที่ 4.1 ค่ายูนิตน้ำจากมิเตอร์น้ำประปาที่ใช้ในแต่ละวัน

วันที่	ค่ามิเตอร์ที่อ่านได้	ค่ายูนิตน้ำที่ใช้ในแต่ละวัน (ยูนิต)
13/01/2018	0387,797	0388,551 - 0387,797 = 0.754
14/01/2018	0388,551	0389,297 - 0388,551 = 0.746
15/01/2018	0389,297	0389,806 - 0389,297 = 0.509
16/01/2018	0389,806	0

จากตารางที่ 4.1 แสดงค่ายูนิตน้ำจากมิเตอร์น้ำประปาที่อ่านได้ในเวลาเที่ยงคืนของแต่ละวัน และนำมาหาผลต่างจะได้ค่ายูนิตน้ำประปาที่ใช้ในแต่ละวัน

y	m	d	sumunit
2018	1	13	0.7375260023009105
2018	1	14	0.7576660003214784
2018	1	15	0.5271100001828017

รูปที่ 4.2 ค่ายูนิตจาก Water flow sensor ในแต่ละวัน

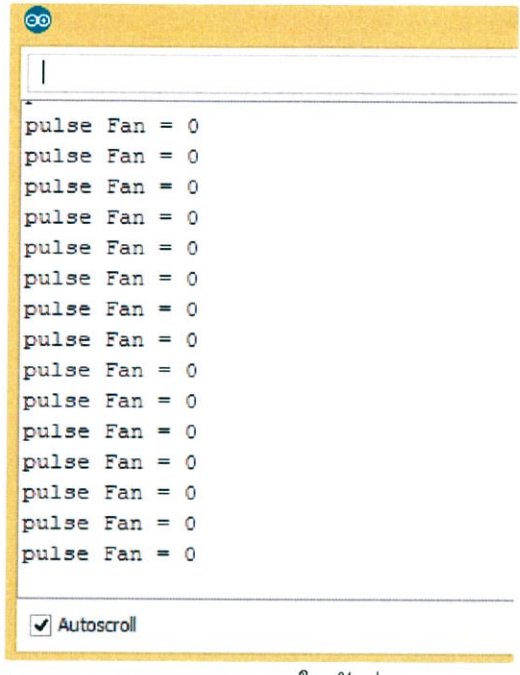
จากรูปที่ 4.2 แสดงค่าผลรวมของยูนิตน้ำในแต่ละวัน จาก Water flow sensor นำมาเปรียบเทียบกับค่ายูนิตน้ำจากมิเตอร์น้ำประปาที่ใช้ในแต่ละวัน เพื่อหาค่าความคลาดเคลื่อนแสดงดังตารางที่ 4.2

ตารางที่ 4.2 ค่าความคลาดเคลื่อน

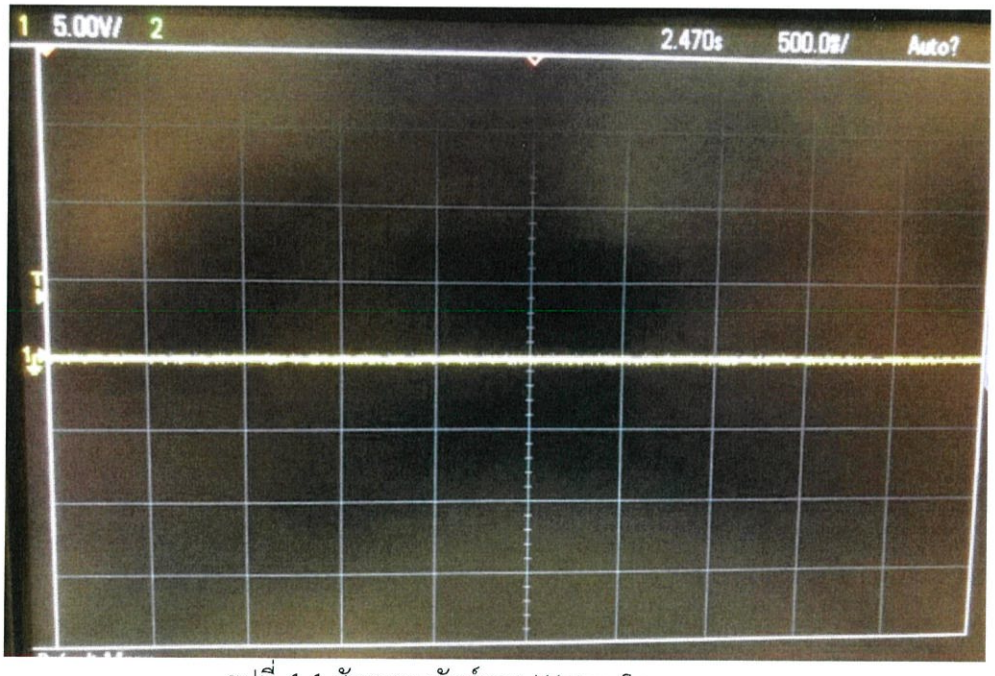
วันที่	ค่าความคลาดเคลื่อน (%)
13/01/2018	2.1220
14/01/2018	1.5684
15/01/2018	3.5363

จากตารางที่ 4.2 แสดงค่าความคลาดเคลื่อนที่คำนวณได้ในแต่ละวัน จากนั้นทำการหาค่าความคลาดเคลื่อนรวม 3 วัน โดยหาผลรวมของยูนิตน้ำที่ได้จากมิเตอร์ได้เท่ากับ 2.009 ยูนิต หาผลรวมของยูนิตน้ำจาก Water flow sensor ได้เท่ากับ 2.022302 หาค่าความคลาดเคลื่อนรวมได้ ((2.022302 – 2.009) * 100) / 2.009 เท่ากับ 0.662120 เปอร์เซ็นต์

3) วัดสัญญาณพัลส์ของ Water flow sensor ผ่าน Oscilloscope



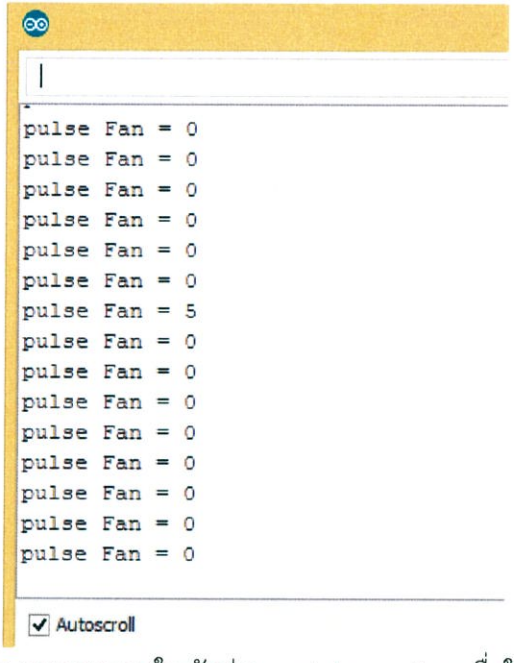
รูปที่ 4.3 การแสดงผลการหมุนของใบพัดผ่าน serial monitor



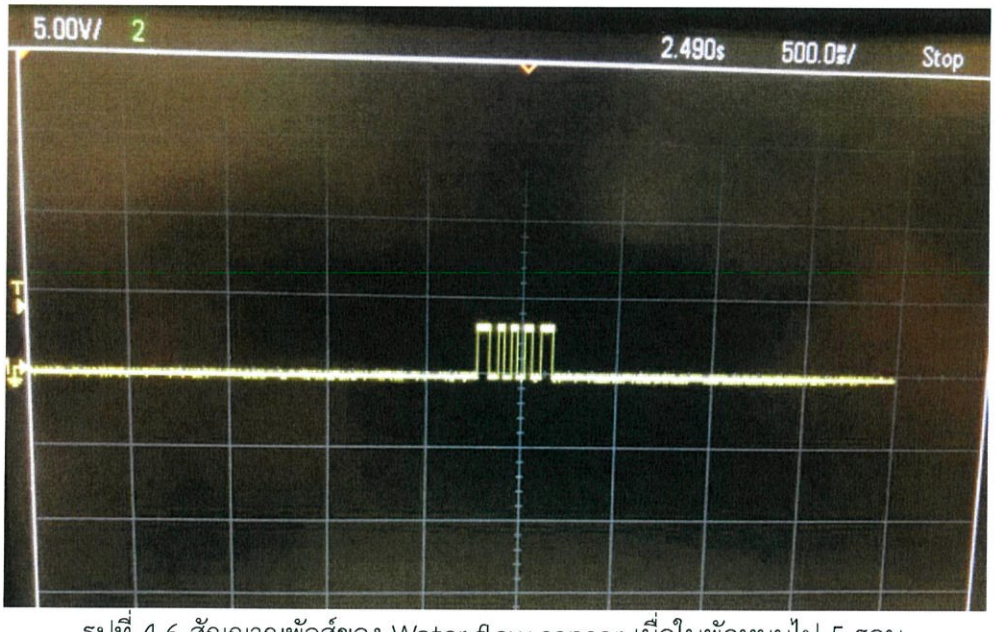
รูปที่ 4.4 สัญญาณพัลส์ของ Water flow sensor

จากรูปที่ 4.3 และ 4.4 แสดงหน้าต่าง serial monitor ที่แสดงจำนวนใบพัดของ Water flow sensor และแสดงสัญญาณพัลส์ของ Water flow sensor ผ่าน oscilloscope เมื่อ

ใบพัดของ Water flow sensor หมุนจะทำให้เกิดพัลส์ขึ้นมาหมุน n รอบ จะเกิดสัญญาณ n พัลส์ และ NodeMCU จะสามารถอ่านค่าออกมาได้ในจำนวนที่เท่ากัน



รูปที่ 4.5 การแสดงผลการหมุนของใบพัดผ่าน serial monitor เมื่อใบพัดหมุนไป 5 รอบ



รูปที่ 4.6 สัญญาณพัลส์ของ Water flow sensor เมื่อใบพัดหมุนไป 5 รอบ

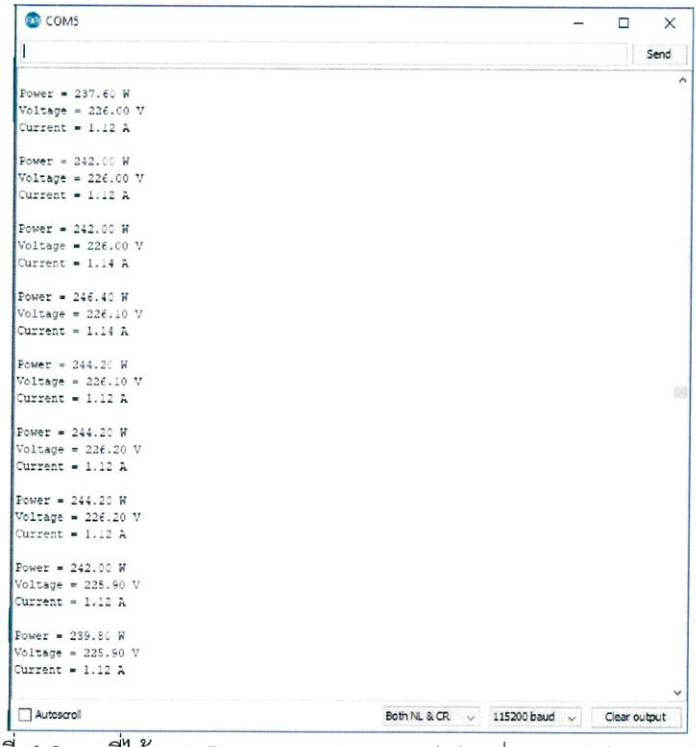
จากรูปที่ 4.5 และ 4.6 แสดงความถี่ที่ได้จากสัญญาณพัลส์ของ Water flow sensor ซึ่งเป็นตัวแปรที่ใช้ในการคำนวณหาค่าอัตราไหลของน้ำดังสมการที่ 2.3

4.1.2 การทดสอบความถูกต้องของ Power meter module

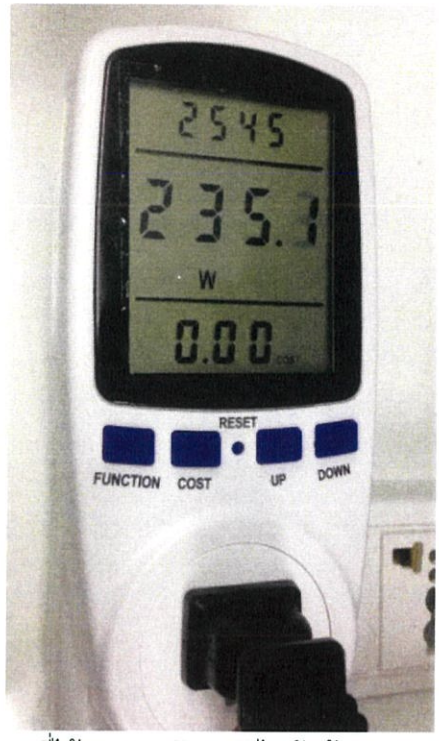
เป็นการทดสอบความถูกต้องของค่าปริมาณการใช้ไฟฟ้าที่วัดได้จาก Power meter module เปรียบเทียบกับ Power meter ยี่ห้อ Intertek เพื่อหาค่าความคลาดเคลื่อนของ Power meter module

1) ใช้ NodeMCU อ่านค่าปริมาณการใช้ไฟฟ้าจาก Power meter module และแสดงผลผ่าน serial monitor นำค่าที่แสดงไปเปรียบเทียบกับค่าที่อ่านได้จาก Power meter ยี่ห้อ Intertek

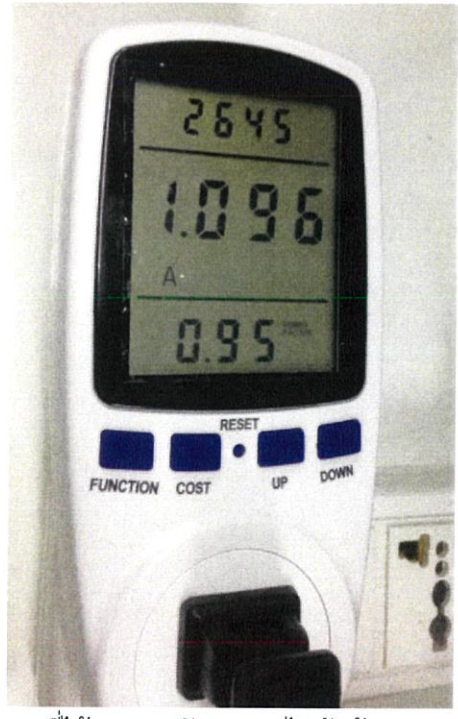
รูปที่ 4.7 Power meter ยี่ห้อ Intertek



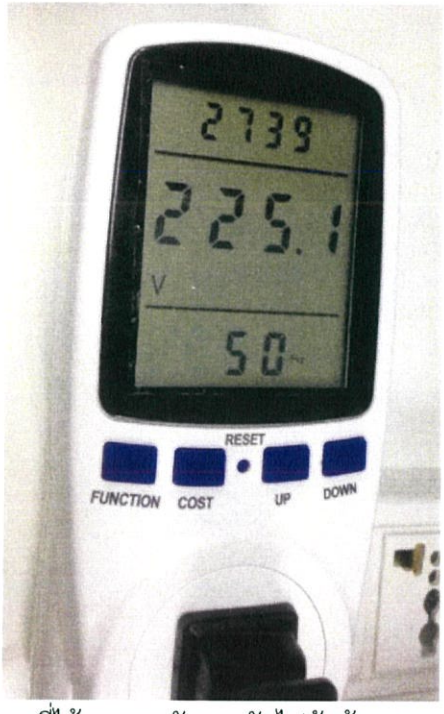
รูปที่ 4.8 ผลที่ได้จาก Power meter module ผ่าน serial monitor



รูปที่ 4.9 ผลที่ได้จากการวัดกำลังไฟฟ้าด้วย Power meter



รูปที่ 4.10 ผลที่ได้จากการวัดกระแสไฟฟ้าด้วย Power meter



รูปที่ 4.11 ผลที่ได้จากการวัดแรงดันไฟฟ้าด้วย Power meter

2) การคำนวณความคลาดเคลื่อนของอุปกรณ์

เนื่องจากทั้ง Power meter module และ Power meter มีค่าที่ไม่คงที่ จึงเลือกค่าที่สูงที่สุดและต่ำที่สุดของการวัดทั้งสองแบบมาเปรียบเทียบกันเพื่อคำนวณความคลาดเคลื่อน โดยผลลัพธ์ที่ได้จากการวัดของอุปกรณ์ทั้งสองมีดังนี้

ตารางที่ 4.3 ผลการวัดจากอุปกรณ์ทั้งสองแบบ

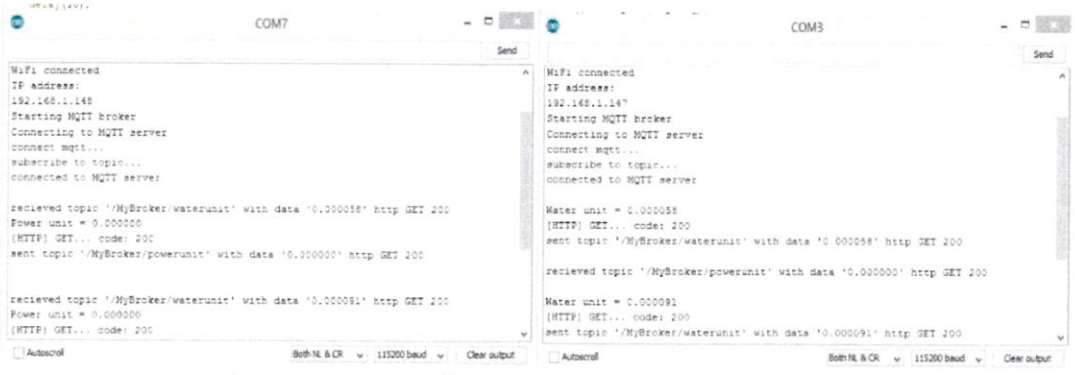
	กำลังไฟฟ้า		แรงดันไฟฟ้า		กระแสไฟฟ้า	
	ค่าสูงสุด	ค่าต่ำสุด	ค่าสูงสุด	ค่าต่ำสุด	ค่าสูงสุด	ค่าต่ำสุด
Power meter module	246.4	237.6	226.2	225.9	1.14	1.12
Power meter	239.8	234.1	225.2	224.9	1.110	1.091

ตารางที่ 4.4 ผลการคำนวณความคลาดเคลื่อน

	กำลังไฟฟ้า		แรงดันไฟฟ้า		กระแสไฟฟ้า	
	ค่าสูงสุด	ค่าต่ำสุด	ค่าสูงสุด	ค่าต่ำสุด	ค่าสูงสุด	ค่าต่ำสุด
ความคลาดเคลื่อน	2.75	1.50	0.44	0.44	2.70	2.66

4.2 การคำนวณค่ายูนิต และส่งไปยังฐานข้อมูล

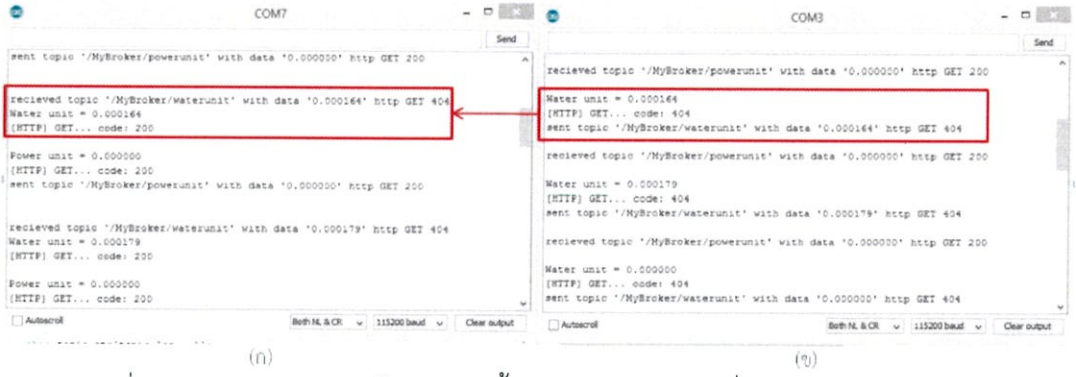
4.2.1 แสดงค่าที่คำนวณได้บน serial monitor



รูปที่ 4.12 ค่ายูนิตของน้ำประปา และไฟฟ้าบน serial monitor

จากรูปที่ 4.12 ด้านขวาคือการแสดงค่ายูนิตของน้ำประปา ซึ่งได้จากการคำนวณค่าความถี่พัลส์ที่อ่านได้จาก Water flow sensor ส่วนด้านซ้ายแสดงค่ายูนิตของไฟฟ้า ซึ่งได้จากการคำนวณค่ากำลังไฟฟ้าที่อ่านได้จาก Power meter module

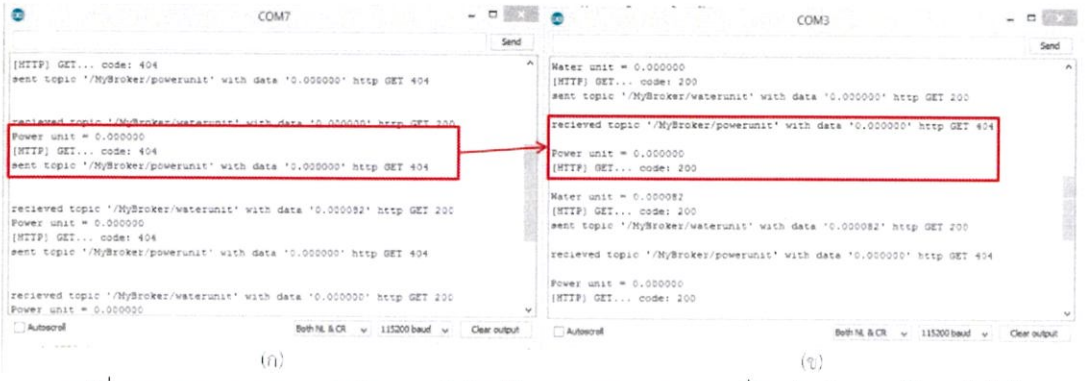
โดยทั้งหมดถูกเขียนด้วยโปรแกรม Arduino IDE และแสดงผลบน serial monitor ซึ่งจะแสดงทุกๆ 1 นาที จากนั้นค่ายูนิตจะถูกส่งไปเก็บยังฐานข้อมูล และส่งข้อมูลไปให้ NodeMCU ตัวอื่น ด้วยโปรโตคอล MQTT



(ก) (ข)

รูปที่ 4.13 NodeMCU ส่งข้อมูลยูนิตน้ำแทน NodeMCU ที่ติดต่อกับฐานข้อมูลไม่ได้

จากรูปที่ 4.13 เป็นการจำลองสถานการณ์ในกรณีที่ (ก) COM3 เป็น NodeMCU ที่ทำหน้าที่ส่งข้อมูลยูนิตน้ำไม่สามารถติดต่อกับฐานข้อมูลได้ โดยมี HTTP Reply Status Code เท่ากับ 404 แสดงถึงคำขอไม่ถูกต้องหรือไม่สามารถปฏิบัติตามได้ เมื่อ (ข) COM7 NodeMCU ตัวอื่นได้รับข้อมูลและตรวจสอบพบว่า HTTP Reply Status Code เท่ากับ 404 NodeMCU จะตัดสินใจส่งข้อมูลยูนิตน้ำแทน NodeMCU ที่ติดต่อกับฐานข้อมูลไม่ได้



รูปที่ 4.14 NodeMCU ส่งข้อมูลยูนิตไฟฟ้าแทน NodeMCU ที่ติดต่อกับฐานข้อมูลไม่ได้

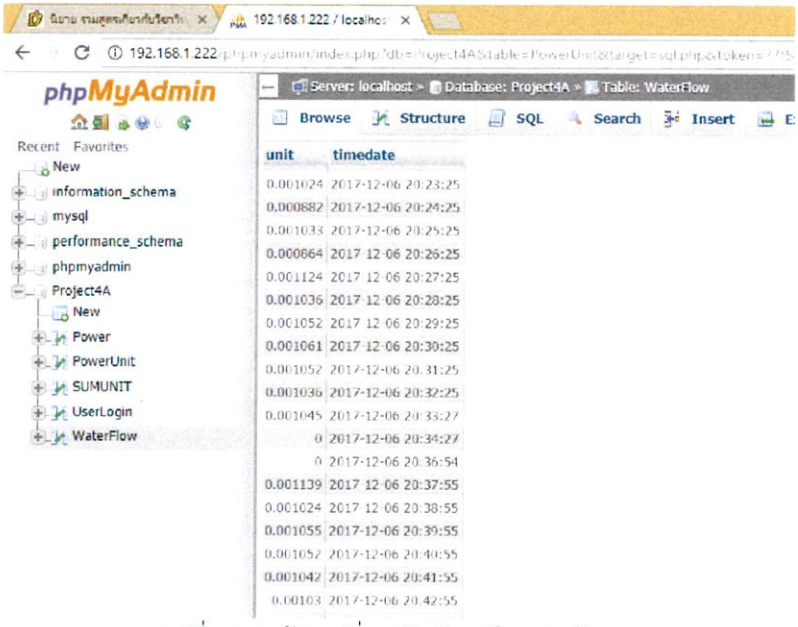
จากรูปที่ 4.14 เป็นการจำลองสถานการณ์ในกรณีที่ (ก) COM7 เป็น NodeMCU ที่ทำหน้าที่ส่งข้อมูลยูนิตไฟฟ้าไม่สามารถติดต่อกับฐานข้อมูลได้ โดยมี HTTP Reply Status Code เท่ากับ 404 แสดงถึงคำขอไม่ถูกต้องหรือไม่สามารถปฏิบัติตามได้ เมื่อ (ข) COM3 NodeMCU ตัวอื่นได้รับข้อมูลและตรวจสอบพบว่า HTTP Reply Status Code เท่ากับ 404 NodeMCU จะตัดสินใจส่งข้อมูลยูนิตไฟฟ้าแทน NodeMCU ที่ติดต่อกับฐานข้อมูลไม่ได้

4.2.2 เขียนโปรแกรมเพื่อนำข้อมูลที่รับมาจาก NodeMCU ไปเก็บไว้ในฐานข้อมูล

```php
<?php
$home = $_GET['home'];
$waterunit = $_GET['waterunit'];

 mysql_connect("localhost", "root", "12345671") or die(mysql_error());
 mysql_select_db("Project4A") or die(mysql_error());

 $data = mysql_query("INSERT INTO WaterFlow(home,unit) VALUES ($home,$waterunit)") or die(mysql_error());
?>
```

รูปที่ 4.15 เขียนโปรแกรมเพื่อเก็บข้อมูลในฐานข้อมูล

จากรูปที่ 4.15 แสดงการเขียนโปรแกรมโดยใช้ภาษา PHP เชื่อมต่อกับฐานข้อมูล และเพิ่มข้อมูลโดยใช้คำสั่ง insert ของ MySQL ไปยังตารางฐานข้อมูล

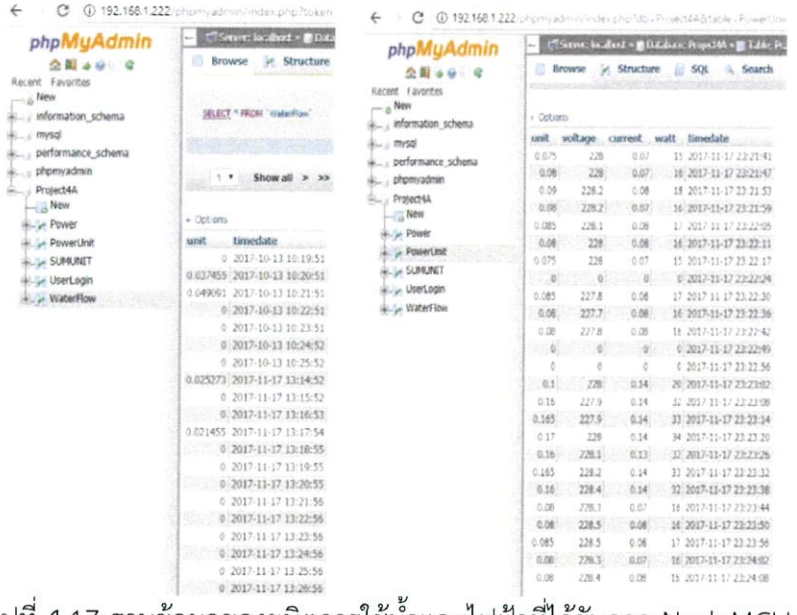


รูปที่ 4.16 ข้อมูลที่ถูกบันทึกลงในฐานข้อมูล

จากรูปที่ 4.16 แสดงข้อมูลยูนิตที่ถูกเก็บไว้ในตารางฐานข้อมูล ซึ่งใช้ phpMyAdmin เป็นตัวจัดการฐานข้อมูล

4.3 การนำข้อมูลจากฐานข้อมูลมาแสดงบนหน้าต่างเว็บเบราว์เซอร์

4.3.1 ฐานข้อมูลของยูนิตของน้ำประปาและไฟฟ้า



รูปที่ 4.17 ฐานข้อมูลของยูนิตการใช้น้ำและไฟฟ้าที่ได้รับจาก NodeMCU

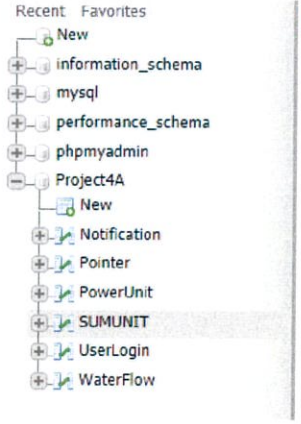
จากรูปที่ 4.17 แสดงตารางฐานข้อมูลของยูนิตของน้ำประปาและไฟฟ้าที่วัดได้จาก Water flow sensor กับ Power meter module และส่งผ่านเทคโนโลยีสื่อสารไร้สาย(Wi-Fi)โดยใช้ NodeMCU นำข้อมูลที่ได้รับไปเก็บไว้ยังฐานข้อมูลที่ถูกสร้างขึ้นโดย phpMyAdmin

4.3.2 เขียนโปรแกรมเพื่ออัพเดทตารางฐานข้อมูล

```
$set = mysql_query("UPDATE SUMUNIT SET sumpowerunit = 0 WHERE home=$homepointer") or die(mysql_error());
$set = mysql_query("UPDATE SUMUNIT SET powerbill = 0 WHERE home=$homepointer") or die(mysql_error());
$set = mysql_query("UPDATE SUMUNIT SET sumwaterflow = 0 WHERE home=$homepointer") or die(mysql_error());
$set = mysql_query("UPDATE SUMUNIT SET waterbill = 0 WHERE home=$homepointer") or die(mysql_error());

$data = mysql_query("SELECT year(timedate) as y, month(timedate) as m, sum(unit) as sumunit from PowerUnit where home=$homepointer g
  while($info = mysql_fetch_array($data))
  {
    $sumunit = $info['sumunit'];
    if ($sumunit <= 15)
    {
      $price = ($sumunit*2.3488) + 8.19 ;
    }
    elseif ($sumunit <= 25)
    {
      $price = (($sumunit-15) * 2.9882) + 8.19 +35.232;
    }
    elseif ($sumunit <= 35)
    {
      $price = (($sumunit-25) * 3.2405) + 8.19 +65.114;
    }
    elseif ($sumunit <= 100)
    {
      $price = (($sumunit-35) * 3.6237) + 8.19 +97.519;
    }
    elseif ($sumunit <= 150)
    {
      $price = (($sumunit-100) * 3.7171) + 8.19 +333.0595;
    }
    elseif ($sumunit <= 400)
```

รูปที่ 4.18 เขียนโปรแกรมเพื่ออัพเดทตารางฐานข้อมูล

จากรูปที่ 4.18 แสดงการเขียนโปรแกรมเพื่ออัพเดทตารางฐานข้อมูลโดยดึงข้อมูลการใช้น้ำประปาและไฟฟ้า นำมารวมกันโดยจับกลุ่มเป็นรายเดือนและรายบ้าน อีกทั้งคำนวณค่าน้ำประปาและไฟฟ้า อัพเดทลงในตารางฐานข้อมูล SUMUNIT ดังรูปที่ 4.19



home	y	Month	m	sumpowerunit	powerbill	sumwaterflow	waterbill
1	2017	Jan	1	0	0	0	0
1	2017	Feb	2	0	0	0	0
1	2017	Mar	3	0	0	0	0
1	2017	Apr	4	0	0	0	0
1	2017	May	5	0	0	0	0
1	2017	June	6	0	0	0	0
1	2017	July	7	0	0	0	0
1	2017	Aug	8	0	0	0	0
1	2017	Sept	9	0	0	0	0
1	2017	Oct	10	0	0	0	0
1	2017	Nov	11	3.51	17.5847	0.105221	0.956985
1	2017	Dec	12	2.35	14.6694	0.072878	0.662825
1	2018	Jan	1	0.257187	9.40967	2.11009	19.1912
1	2018	Feb	2	0	8.7633	0	0

รูปที่ 4.19 ตารางฐานข้อมูลที่ทำการอัพเดท

4.3.3 เขียนโปรแกรมเพื่อดึงข้อมูลจากฐานข้อมูลมาพล็อตกราฟและตาราง

```
function drawChart() {
  var data = google.visualization.arrayToDataTable([
    <?php
    $data = mysql_query("SELECT * from SUMUNIT where home=$homepointer and y='$sY1'") or die(mysql_error());
    print "['Month', 'Water Bill (baht)' , 'Power Bill (baht)'],";
    while($info = mysql_fetch_array($data))
    {
      print "['" . $info['Month'] . "'," . $info['waterbill'] . ",". $info['powerbill'] .'],";
    }
    ?>
  ]);

  var options = {
    vAxis: {
format: '####.######' // show axis values to 3 decimal places
    },
    hAxis: {title: 'Month',

      viewWindow: {
                min: 0,
                max: 12
                },
    },
    bars: 'vertical' // Required for Material Bar Charts.
```

รูปที่ 4.20 เขียนโปรแกรมเพื่อดึงข้อมูลจากฐานข้อมูลมาสร้างกราฟและตาราง

จากรูปที่ 4.20 เป็นการเขียนโปรแกรมเพื่อเชื่อมต่อกับฐานข้อมูล หลังจากการอัพเดทตารางฐานข้อมูลแล้ว ทำการ select ข้อมูลยูนิตน้ำประปาและไฟฟ้าที่ได้รับมาจาก NodeMCU ผลรวมของยูนิตของน้ำประปา ไฟฟ้า ค่าน้ำประปา และค่าไฟฟ้าในแต่ละเดือน แล้วนำไปสร้างกราฟและตารางเพื่อนำไปแสดงบนหน้าต่างของเว็บเบราว์เซอร์

4.3.4 การสร้างการแจ้งเตือนบนเว็บเบราว์เซอร์

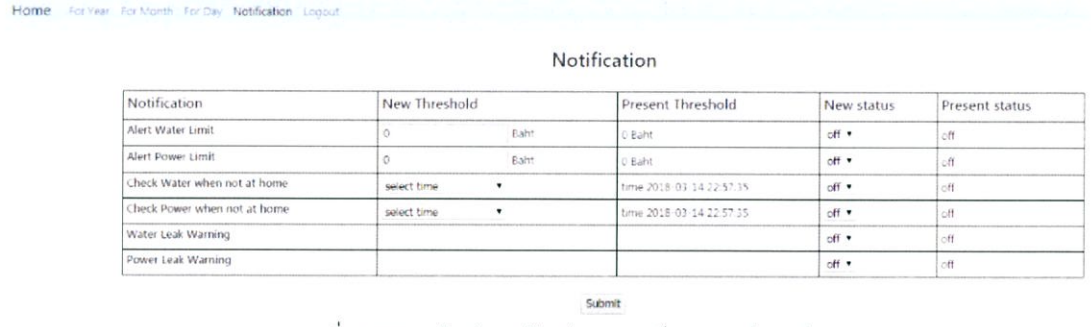
1) เขียนโปรแกรมการแจ้งเตือนบนเว็บเบราว์เซอร์

```
$data = mysql_query("SELECT * from Notification WHERE home=$homepointer and NotificationName = 'WarningWater'") or die(mysql_error())
while($info = mysql_fetch_array($data))
    {
      $warningwater = $info['Status'];
    }
$data = mysql_query("SELECT * from Notification WHERE home=$homepointer and NotificationName = 'WarningPower'") or die(mysql_error())
while($info = mysql_fetch_array($data))
    {
      $warningpower = $info['Status'];
    }

if ($_SERVER["REQUEST_METHOD"] == "POST") {
  $waterlimit = test_input($_POST["waterlimit"]);
  $tswaterlimit = test_input($_POST["tswaterlimit"]);
  $powerlimit = test_input($_POST["powerlimit"]);
  $tspowerlimit = test_input($_POST["tspowerlimit"]);
  $checkwater = test_input($_POST["checkwater"]);
  $tscheckwater = test_input($_POST["tscheckwater"]);
  $checkpower = test_input($_POST["checkpower"]);
  $tscheckpower = test_input($_POST["tscheckpower"]);
  $warningwater = test_input($_POST["warningwater"]);
  $warningpower = test_input($_POST["warningpower"]);
}

function test_input($data) {
  $data = trim($data);
  $data = stripslashes($data);
  $data = htmlspecialchars($data);
  return $data;
```

รูปที่ 4.21 เขียนโปรแกรมการแจ้งเตือน

จากรูปที่ 4.21 เป็นการเขียนโปรแกรมการแจ้งเตือนบนเว็บเบราว์เซอร์ โดยใช้ภาษา HTML และ PHP แสดงดังรูปที่ 4.22

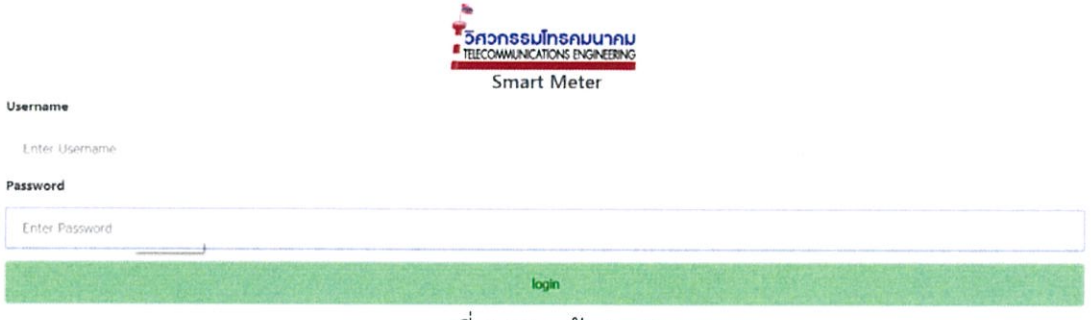
2) หน้าต่างแจ้งเตือนบนเว็บเบราว์เซอร์



รูปที่ 4.22 หน้าต่างแจ้งเตือนบนเว็บเบราว์เซอร์

4.3.5 การแสดงผลบนเว็บเบราว์เซอร์

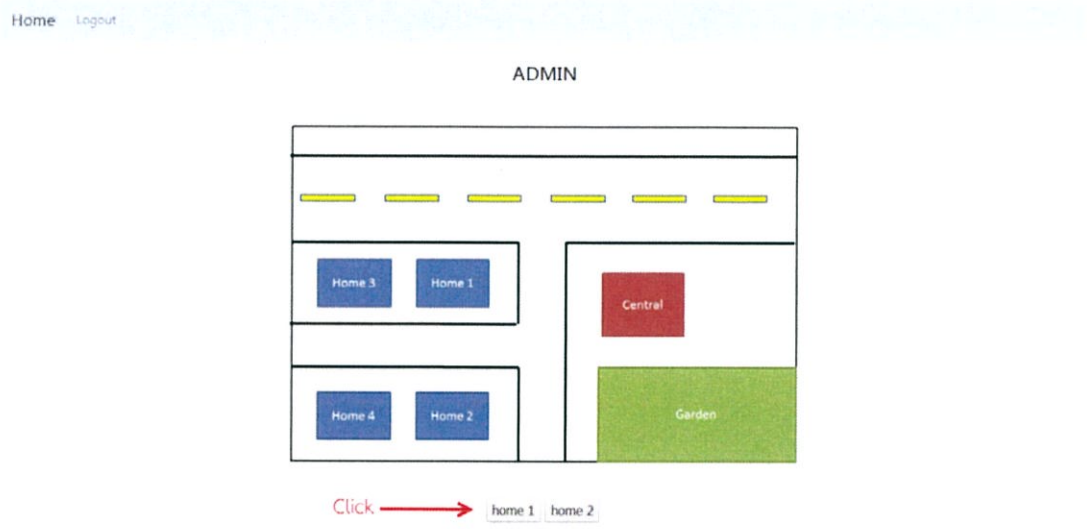
4.5.3.1) หน้า Login



รูปที่ 4.23 หน้า Login

จากรูปที่ 4.23 เป็นการแสดงหน้าต่าง Login ซึ่งจะแบ่งความสำคัญออกเป็น 2 ส่วนคือ 1) ส่วนของผู้ดูแลระบบ สามารถดูข้อมูลการใช้น้ำประปาและไฟฟ้าของบ้านแต่ละหลังในหมู่บ้านได้ 2) ส่วนของผู้ใช้งาน สามารถดูข้อมูลการใช้น้ำประปาและไฟฟ้าภายในบ้านของตัวเองเท่านั้น

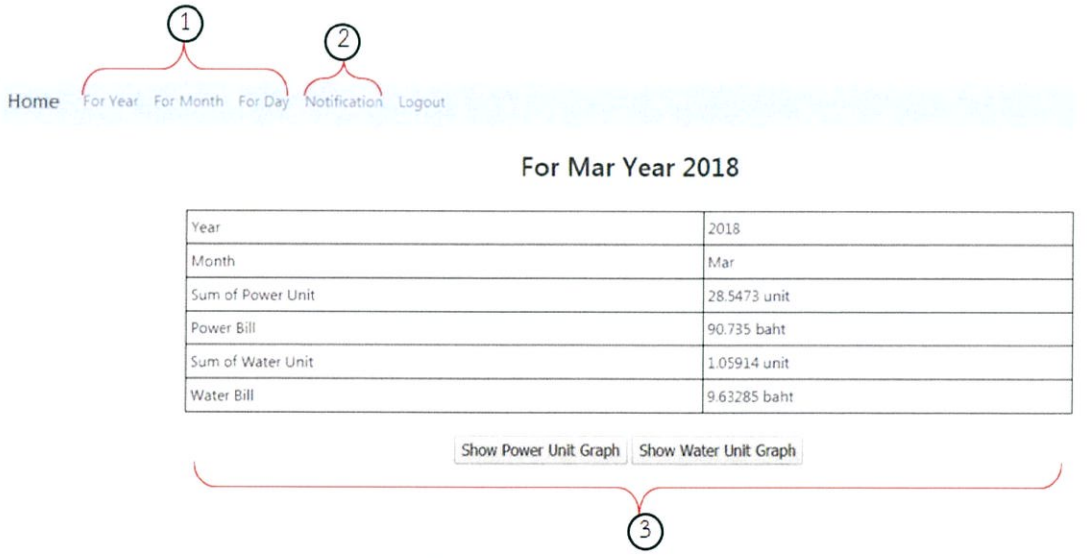
4.3.5.2) ส่วนของผู้ดูแลระบบ



รูปที่ 4.24 ส่วนของผู้ดูแลระบบ

จากรูปที่ 4.24 เป็นการแสดงส่วนของผู้ดูแลระบบ เมื่อผู้ดูแลระบบทำการ Login เข้ามาจะพบกับหน้าต่างผู้ดูแลระบบ ซึ่งจากรูปจะพบว่าผู้ดูแลระบบสามารถดูข้อมูลการใช้น้ำประปาและไฟฟ้าของบ้านแต่ละหลังได้

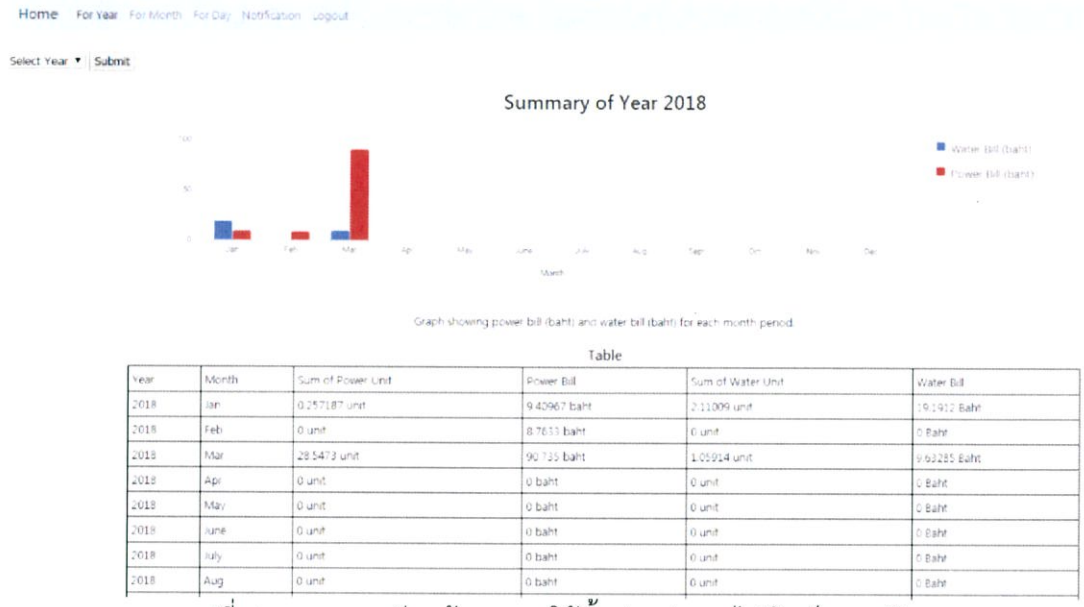
4.3.5.3) ส่วนของผู้ใช้งาน



รูปที่ 4.25 ส่วนของผู้ใช้งาน

จากรูปที่ 4.25 เป็นการแสดงส่วนของผู้ใช้งาน เมื่อผู้ใช้งานทำการ Login เข้ามาจะพบกับหน้าต่างผู้ใช้งาน ซึ่งจากรูปจะพบว่า ผู้ใช้งานสามารถดูข้อมูลการใช้น้ำประปาและไฟฟ้าได้ 3 ส่วน คือ

1) การดูรายละเอียดการใช้น้ำประปาและไฟฟ้าเป็นรายปี รายเดือน รายวัน

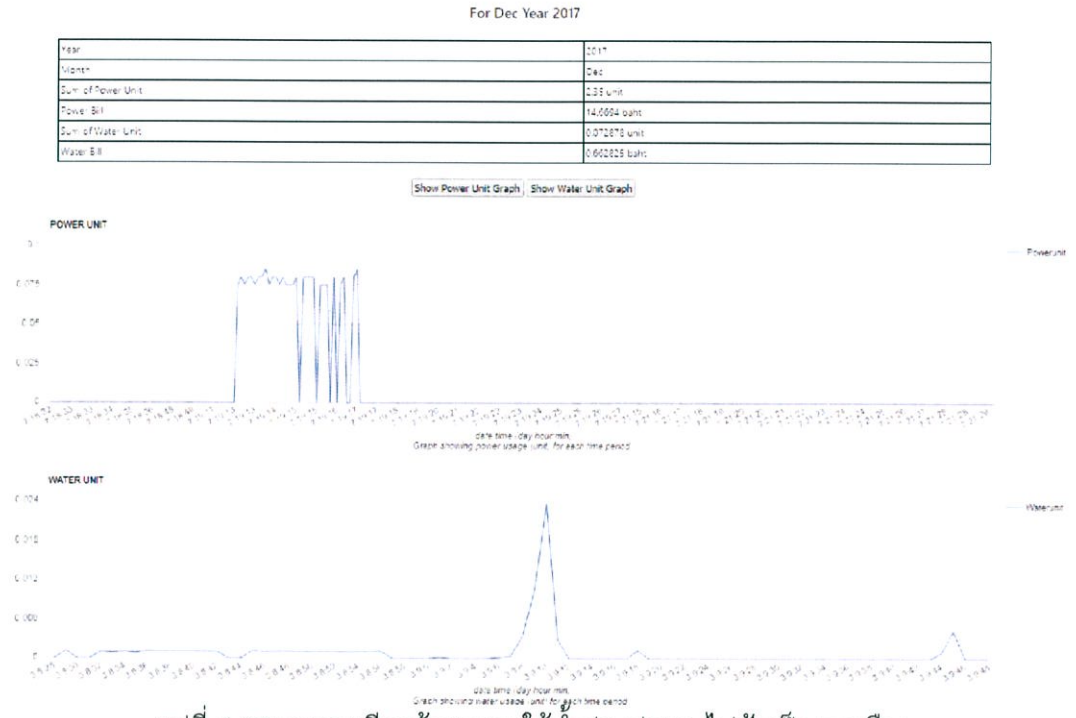


Table

Year	Month	Sum of Power Unit	Power Bill	Sum of Water Unit	Water Bill
2018	Jan	0.257187 unit	9.40967 baht	2.11009 unit	19.1912 Baht
2018	Feb	0 unit	8.7633 baht	0 unit	0 Baht
2018	Mar	28.5473 unit	90.735 baht	1.05914 unit	9.62285 Baht
2018	Apr	0 unit	0 baht	0 unit	0 Baht
2018	May	0 unit	0 baht	0 unit	0 Baht
2018	June	0 unit	0 baht	0 unit	0 Baht
2018	July	0 unit	0 baht	0 unit	0 Baht
2018	Aug	0 unit	0 baht	0 unit	0 Baht

รูปที่ 4.26 รายละเอียดข้อมูลการใช้น้ำประปาและไฟฟ้าเป็นรายปี

จากรูปที่ 4.26 จะแสดงรายละเอียดข้อมูลการใช้น้ำประปาและไฟฟ้าเป็นรายปี ซึ่งถูกแสดงผ่านรูปแบบของกราฟและตาราง โดยชนิดของกราฟที่ใช้แสดงเป็นกราฟแท่ง แสดงผลรวมค่าน้ำประปาและไฟฟ้าที่ใช้ในแต่ละเดือนในปีที่เลือก โดยแกนตั้งคือจำนวนเงิน แกนนอนคือเดือน ในส่วนตารางจะแสดงผลรวมยูนิตของน้ำประปา ไฟฟ้า ค่าน้ำประปาและค่าไฟฟ้าในแต่ละเดือนของในปีที่เลือก



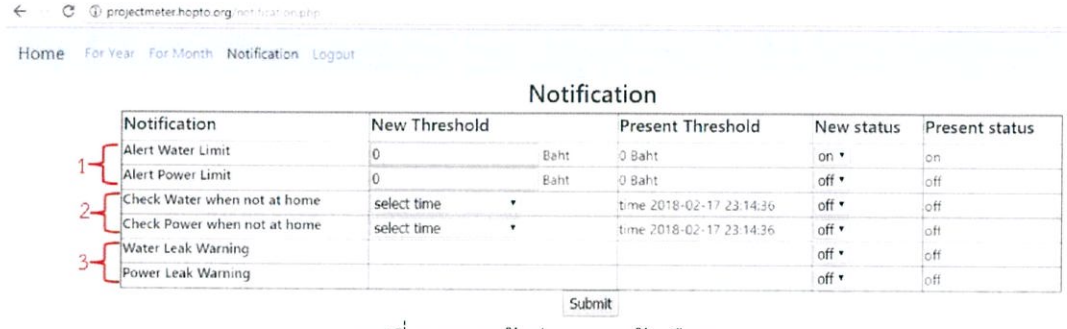
รูปที่ 4.27 รายละเอียดข้อมูลการใช้น้ำประปาและไฟฟ้าเป็นรายเดือน

จากรูปที่ 4.27 จะแสดงรายละเอียดข้อมูลการใช้น้ำประปาและไฟฟ้าเป็นรายเดือน ซึ่งถูกแสดงผ่านรูปแบบของกราฟและตาราง โดยกราฟจะเป็นกราฟเส้นแสดงยูนิตน้ำประปา และไฟฟ้าที่ใช้ในเดือนที่เลือก โดยแกนตั้งคือปริมาณยูนิตที่ใช้ต่อนาที แกนนอนคือเวลา ในส่วนตารางจะแสดงผลรวมยูนิตของน้ำประปา ไฟฟ้า ค่าน้ำประปาและค่าไฟฟ้าในเดือนที่เลือก

รูปที่ 4.28 รายละเอียดข้อมูลการใช้น้ำประปาและไฟฟ้าเป็นรายวัน

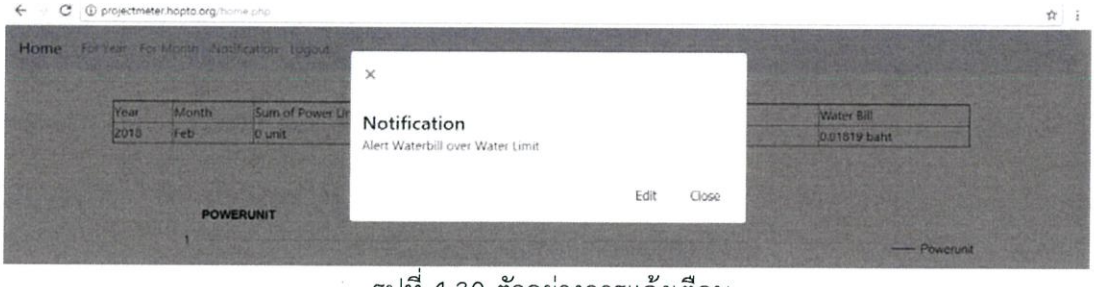
จากรูปที่ 4.28 จะแสดงรายละเอียดข้อมูลการใช้น้ำประปา และไฟฟ้าเป็นรายวัน ซึ่งถูกแสดงผ่านรูปแบบของกราฟและตาราง โดยชนิดของกราฟที่ใช้แสดงผลเป็นกราฟเส้น แสดงยูนิตน้ำประปาและไฟฟ้าที่ใช้ในวันที่เลือก โดยแกนตั้งคือปริมาณยูนิตที่ใช้ต่อนาที แกนนอนคือเวลา ในส่วนตารางจะแสดงผลรวมยูนิตของน้ำประปาและไฟฟ้าในวันที่เลือก

2) หน้าต่างการแจ้งเตือน



รูปที่ 4.29 หน้าต่างการแจ้งเตือน

จากรูปที่ 4.29 เป็นการแสดงหน้าต่างการแจ้งเตือนบนเว็บเบราว์เซอร์ โดยแบ่งออกเป็น 3 ส่วนคือ 1) เป็นการแจ้งเตือนค่าน้ำประปาและค่าไฟฟ้าเกินจากปริมาณที่ผู้ใช้กำหนด 2) เป็นการแจ้งเตือนเมื่อมีการใช้น้ำประปาและไฟฟ้าขณะไม่อยู่บ้าน 3) เป็นการเช็คการรั่วไหลของน้ำประปาและไฟฟ้า ซึ่งทั้ง 3 ส่วนนี้ผู้ใช้งานสามารถกำหนดเองได้ว่า จะให้มีการแจ้งเตือนที่ส่วนไหนบ้างและเมื่อเป็นไปตามเงื่อนไขที่กำหนดจะมีการแจ้งเตือนในรูปแบบหน้าต่างแจ้งเตือนปรากฏขึ้นมาบนเว็บเบราว์เซอร์ ดังรูปที่ 4.30



รูปที่ 4.30 ตัวอย่างการแจ้งเตือน

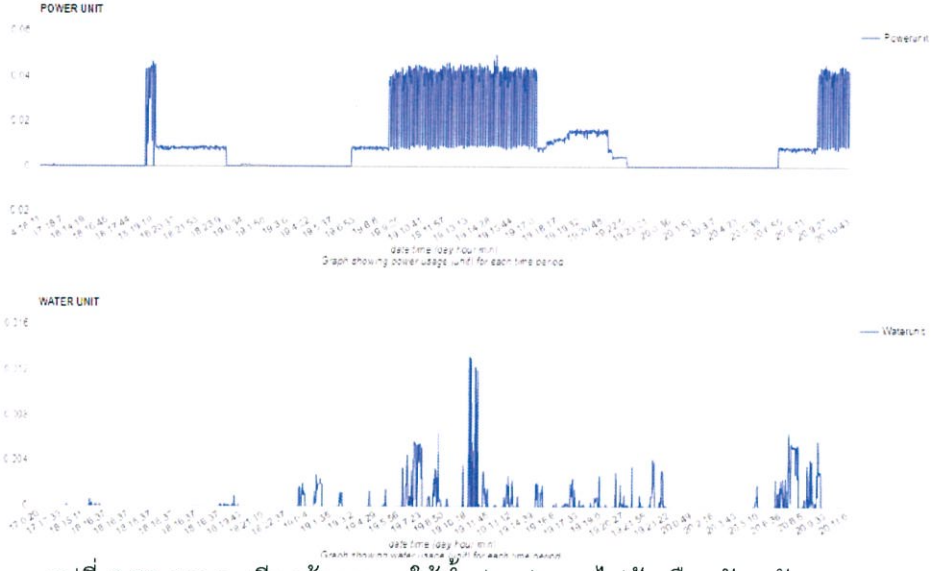
จากรูปที่ 4.30 เป็นตัวอย่างในส่วนที่ 1 ของการแจ้งเตือนเมื่อมีค่าน้ำประปาเกินจากราคาที่ผู้ใช้กำหนด ซึ่งสามารถกด Edit กลับไปแก้ไขข้อมูลที่หน้าต่างการแจ้งเตือนได้

3) ข้อมูลปัจจุบัน

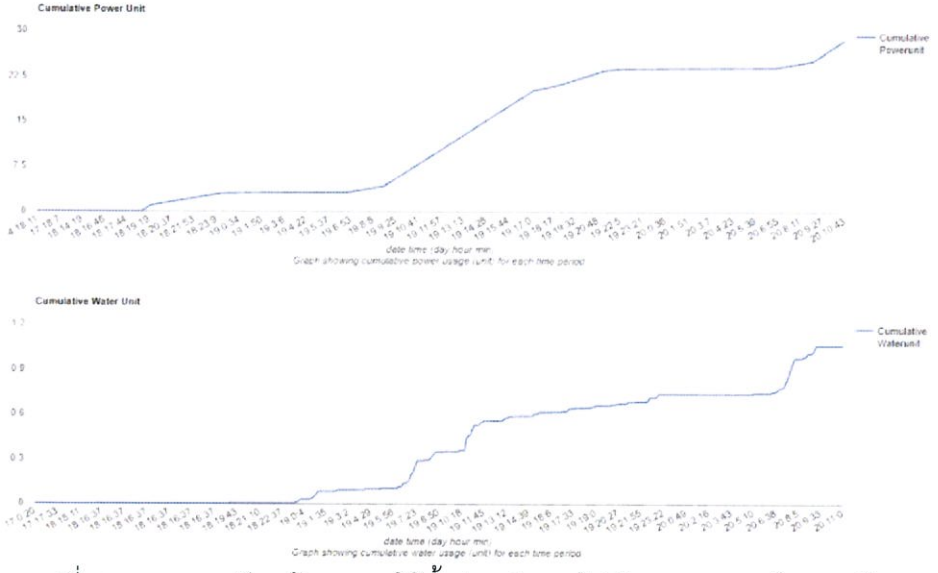
For Mar Year 2018

Year	2018
Month	Mar
Sum of Power Unit	28.5473 unit
Power Bill	90.735 baht
Sum of Water Unit	1.05914 unit
Water Bill	9.63265 baht

Show Power Unit Graph | Show Water Unit Graph | Show Cumulative Power Unit Graph | Show Cumulative Water Unit Graph



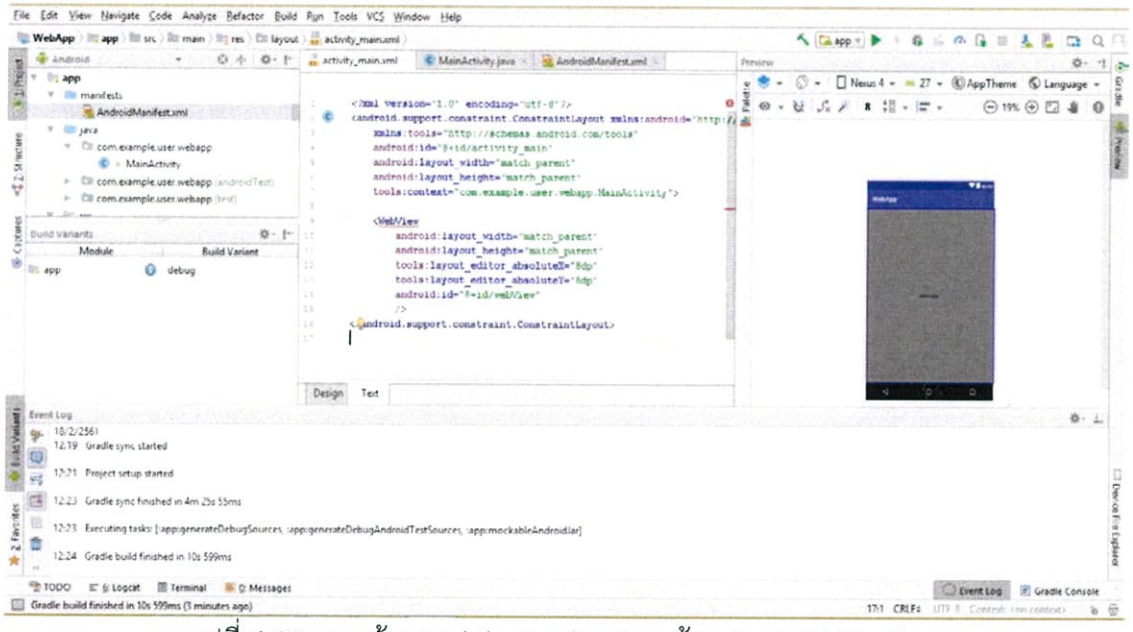
รูปที่ 4.31 รายละเอียดข้อมูลการใช้น้ำประปาและไฟฟ้าเดือนปัจจุบัน



รูปที่ 4.32 รายละเอียดข้อมูลการใช้น้ำประปาและไฟฟ้าแบบสะสมเป็นรายเดือน

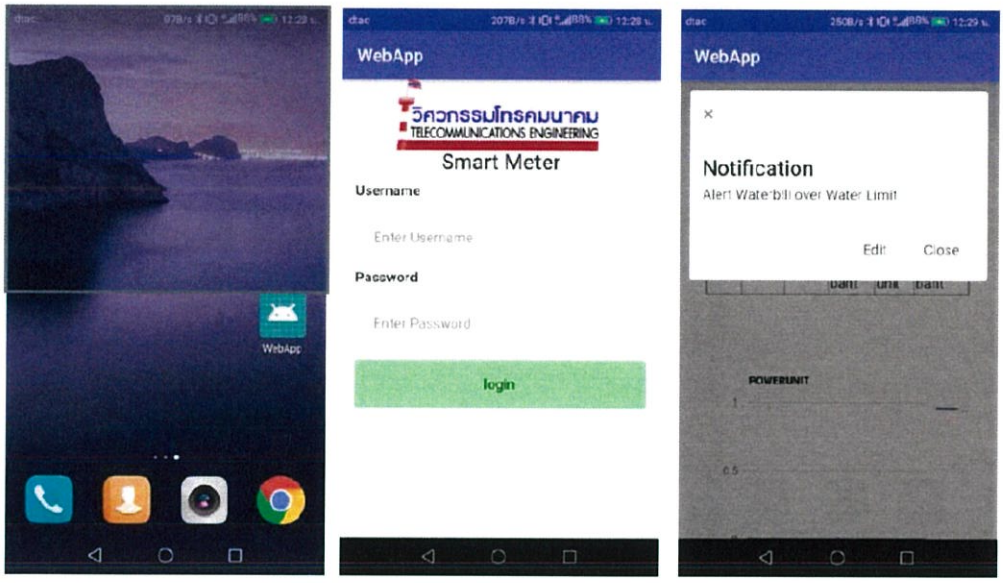
จากรูปที่ 4.31 และ 4.32 จะแสดงรายละเอียดข้อมูลการใช้น้ำประปาและไฟฟ้าในเดือนปัจจุบัน ซึ่งถูกแสดงผ่านรูปแบบของกราฟและตาราง โดยชนิดของกราฟที่ใช้เป็นกราฟเส้นแสดงยูนิตน้ำประปาและไฟฟ้าที่ใช้ในเดือนปัจจุบันทั้งแบบปริมาณยูนิตในแต่ละช่วงเวลาและปริมาณยูนิตสะสม โดยแกนตั้งคือปริมาณยูนิตที่ใช้ แกนนอนคือเวลา ในส่วนตารางจะแสดงผลรวม ยูนิตของน้ำประปา ไฟฟ้า ค่าน้ำประปาและค่าไฟฟ้าในเดือนปัจจุบัน

4.4 การสร้าง Mobile Application เว็บเบราว์เซอร์



รูปที่ 4.33 การสร้าง Mobile Application ด้วย Android Studio

จากรูปที่ 4.33 เป็นการสร้าง Mobile Application ด้วย Android Studio โดยให้ Mobile Application มี Activity เป็น Web View โดยมี URL เป็น URL ของเว็บเบราว์เซอร์ของระบบ และสร้างเป็นไฟล์ .APK เพื่อไปติดตั้งในมือถือ ดังรูปที่ 4.34



รูปที่ 4.34 Mobile Application

จากรูปที่ 4.34 เมื่อติดตั้งไฟล์ .APK สำเร็จจะได้ Mobile Application ที่สร้างขึ้นมา โดยมีหน้าตาและฟังก์ชันเหมือนกับดูผ่านเว็บเบราว์เซอร์

บทที่ 5

สรุปผลและข้อเสนอแนะ

5.1 สรุปผล

ระบบตรวจสอบการใช้น้ำและไฟฟ้าแบบไร้สาย (Water and electricity consumption monitoring system via wireless) เป็นระบบที่ใช้วัดปริมาณการใช้น้ำประปาและไฟฟ้า ซึ่งจะแสดงผลผ่านแอปพลิเคชัน ในรูปของตารางและกราฟ โดยอาศัยหลักการทำงานของ NodeMCU ESP8266, Power meter module, Water flow sensor และ Raspberry Pi รวมกันเป็นระบบซึ่งสามารถสรุปผลการทดลองได้ดังนี้

1) ระบบสามารถส่งข้อมูลยูนิตน้ำประปาและไฟฟ้า โดยใช้ NodeMCU ทำการส่งข้อมูลทุกๆ 1 นาที

2) ระบบสามารถรวมค่ายูนิตน้ำประปาและไฟฟ้า แล้วนำมาคำนวณค่าน้ำประปา และไฟฟ้าได้ในแต่ละเดือน

3) ระบบสามารถแสดงค่าน้ำประปาและค่าไฟฟ้าผ่านเว็บเบราว์เซอร์

4) ระบบสามารถแสดงผลผ่านเว็บเบราว์เซอร์ ในรูปของตารางและกราฟ ซึ่งในหน้าอินเตอร์เฟสจะสามารถเลือกกลุ่มข้อมูลได้ว่าจะดูข้อมูลแบบปี, เดือนหรือวัน

5) ระบบสามารถแจ้งเตือนการใช้น้ำประปาและไฟฟ้าผ่านเว็บเบราว์เซอร์

6) ระบบสามารถให้ผู้ใช้สามารถเข้ามาใช้งานระบบจากเครือข่ายภายนอกได้

7) ระบบสามารถแสดงผลผ่าน Mobile Application ในรูปแบบ Web View

8) ระบบทำงานโดยใช้โปรโตคอล MQTT โดยทำให้ NodeMCU เป็น Broker ซึ่งสามารถรับ-ส่งข้อมูลหากันได้ และส่งข้อมูลแทน NodeMCU ตัวอื่นที่ไม่สามารถส่งไปฐานข้อมูลได้

ระบบตรวจสอบการใช้น้ำและไฟฟ้าแบบไร้สายเป็นระบบที่ออกแบบมาให้มีความยืดหยุ่น ปรับเปลี่ยนได้ให้ตรงกับความต้องการของผู้ใช้งาน และยังสามารถประยุกต์ใช้จริงในชีวิตประจำวัน รวมถึงการพัฒนาต่อยอดในอนาคตได้

5.2 ข้อเสนอแนะ

1) ระบบตรวจสอบการใช้น้ำและไฟฟ้าแบบไร้สายเป็นเพียงการประมาณค่าใช้จ่ายการใช้น้ำประปาและไฟฟ้า ซึ่งอาจจะมีความคลาดเคลื่อนเล็กน้อยเมื่อเทียบกับค่าใช้จ่ายจริงของทางการประปานครหลวงและการไฟฟ้านครหลวง

2) ระบบตรวจสอบการใช้น้ำและไฟฟ้าแบบไร้สายเป็นการเชื่อมต่อในเครือข่าย LAN เดียวกัน ถ้าใช้ในพื้นที่ที่มีขนาดใหญ่ขึ้น จำเป็นจะต้องใช้ Access Point เพื่อขยายสัญญาณให้ครอบคลุมทั่วทั้งพื้นที่

บรรณานุกรม

[1] การไฟฟ้านครหลวง. “อัตราค่าไฟฟ้า”
www.mea.or.th/upload/download/file_4432fed72763bdc538c0a8ef0208df35.pdf

[2] การประปานครหลวง. “อัตราค่าน้ำประปา”
https://www.mwa.co.th/ewt_news.php?nid=303

[3] ฐิติมา มโนหมั่นศรัทธา, และสมชาย พงษ์เกษม. PHP&MySQL Make it easy ง้ายง่าย. กรุงเทพฯ : บริษัทโอเอวันจำกัด, 2550.

[4] Arduitronics. “G3/4 Water Flow Sensor 1-60L/min แรงดันต่ำกว่า 1.75 MPa.”
https://www.arduitronics.com/product/910/g3-4-water-flow-sensor-1-60l-min-แรงดันต่ำกว่า-1-75-mpa-เซนเซอร์วัดอัตราการไหลของน้ำ.

[5] Circuitspecialists. “Product Type: PZEM-004(V3.0)”
https://www.circuitspecialists.com/content/189799/ac004.pdf

[6] Codebender_cc. “How to Use Water Flow Sensor - Arduino Tutorial.”
http://www.instructables.com/id/How-to-Use-Water-Flow-Sensor-Arduino-Tutorial.

[7] Ioxshop. “PZEM-004T AC Digital Power Energy Meter Module.”
https://www.ioxhop.com/product/594/pzem-004t-ac-digital-power-energy-metermodule.

[8] Olehs. “Arduino communication library for Peacefair PZEM-004T Energy monitor.”
https://github.com/olehs/PZEM004T.

[9] RASPBERRY PI. “RASPBERRY PI 3 MODEL B”
https://www.raspberrypi.org/products/raspberry-pi-3-model-b

[10] Silicon Laboratories Inc. “CP2102”
https://www.sparkfun.com/datasheets/IC/cp2102.pdf

[11] ThaiEasyElec. “การใช้งานเริ่มต้น ESP8266 NodeMCU และการใช้งาน Application ต่างๆ.” http://thaieasyelec.com/article-wiki/embedded-electronics-application/gettingstarted-with-esp8266-nodemcu-ch3.html.

[12] The Raspberry Pi Foundation. "Downloading and installing applications on your Raspberry Pi." https://www.raspberrypi.org/learning/software-guide/install-apps.

[13] Network Utilities." portforward" https://portforward.com/asus/rt-n10

[14] Saixiii. "DDNS" https://saixiii.com/what-is-ddns

[15] Tanakorn Piamsin. "ติดตั้ง MQTT SERVER" https://medium.com/@tanakornpiamsin/ติดตั้ง-mqtt-server-d31bcae85d0d

[16] mindphp. "MQTT (Message Queuing Telemetry Transport) คืออะไร" http://www.mindphp.com/บทความ/31-ความรู้ทั่วไป/3343-mqtt.html

[17] physiclab. "percentage error" https://physiclab08.wordpress.com/lab01

ภาคผนวก ก.

โปรแกรม NodeMCU ในส่วนระบบการส่งค่ายูนิตน้ำประปา

1. ส่วนกำหนดตัวแปรและฟังก์ชันที่ใช้ในการอ่านค่าจาก Water flow sensor

```
#include <ESP8266WiFi.h>
#include <ESP8266WiFiMulti.h>
#include <ESP8266HTTPClient.h>
ESP8266WiFiMulti WiFiMulti;
#include <uMQTTBroker.h>
#include <MQTT.h>
char ssid[] = "chenchan";    // your network SSID (name)
char pass[] = "150203LAN"; // your network password
int httpCode;
String url;
float SumFan;                          //กำหนดตัวแปรไว้รวมค่า Pulse frequency
volatile int Fan;                      //กำหนดตัวแปรที่ใช้ในการรับ Pulse frequency
float Cal = 0.0;
float unit = 0.0;
const int Wsensor = D1;                //กำหนดตำแหน่งขาของ Sensor
void runn ()                           //ฟังก์ชันรับค่า Pulse frequency ในการ interrupts
{
  Fan++;
}
void calc ()                           //ฟังก์ชัน เปิด-ปิด การใช้ interrupts
{
  Fan = 0;                             //set ค่าเริ่มต้นเป็น 0
  sei();                               //Enables interrupts
  delay (60000);                       //Wait 1 minute
  cli();                               //Disable interrupts
}
void setup()
{
```

```
  pinMode(Wsensor, INPUT);                    //กำหนดให้ขา D1 เป็น input
  Serial.begin(115200);
  Serial.println();
  Serial.println();
  attachInterrupt(Wsensor, runn, RISING);     //ใช้งาน interrupts
  Serial.print("Connecting to ");
  Serial.println(ssid);
  WiFi.begin(ssid, pass);                     //เลือก ssid , password
  while (WiFi.status() != WL_CONNECTED) {
  delay(500);
  Serial.print(".");
  }
  Serial.println("");
  IPAddress local_ip = {192, 168, 1, 147};    //ตั้งค่า IP
  IPAddress gateway = {192, 168, 1, 1};       //ตั้งค่า IP Gateway
  IPAddress subnet = {255, 255, 255, 0};      //ตั้งค่า Subnet
  WiFi.config(local_ip, gateway, subnet);     //setค่าไปยังโมดูล
  Serial.println("WiFi connected");
  Serial.println("IP address: ");
  Serial.println(WiFi.localIP());
}
```

2. ส่วนคำนวณค่ายูนิตน้ำประปาและส่งไปยังฐานข้อมูล

```
void loop()
{
  calc ();
  Serial.println("");
  if ((WiFiMulti.run() == WL_CONNECTED)) {
    HTTPClient http;
    Cal = ((Fan / 5.5) / 1000) / 60; //คำนวณยูนิตน้ำประปา
    Serial.print("Water unit = ");
    Serial.println(Cal, 6);
    url = "http://192.168.1.222/add2.php?home=1&waterunit=" + String(Cal, 6);
    http.begin(url); //HTTP
    httpCode = http.GET();
    if (httpCode > 0) {
      Serial.printf("[HTTP] GET... code: %d\n", httpCode);
    } else {
      Serial.printf("[HTTP] GET... failed, error: %s\n",
http.errorToString(httpCode).c_str());
    }
    http.end();
  }
}
```

3. ส่วนส่งค่ายูนิตน้ำประปาด้วยโปรโตคอล MQTT

```
unsigned int mqttPort = 1883;                    // the standard MQTT broker port
unsigned int max_subscriptions = 30;
unsigned int max_retained_topics = 30;
void data_callback(uint32_t *client /* we can ignore this */, const char* topic,
uint32_t topic_len, const char *data, uint32_t lengh) {
 char topic_str[topic_len + 1];
 os_memcpy(topic_str, topic, topic_len);
 topic_str[topic_len] = '\0';
 char data_str[lengh + 1];
 os_memcpy(data_str, data, lengh);
 data_str[lengh] = '\0';
 String Val = data_str;
 float firstVal;
 String secondVal;
 for (int i = 0; i < Val.length(); i++)
 {
  if (Val.substring(i, i + 1) == ",") {
   firstVal = Val.substring(0, i).toFloat();
   secondVal = Val.substring(i + 1).toInt();
   break;
  }
 }
 Serial.print("sent topic '");
 Serial.print(topic_str);
 Serial.print("' with data '");
 Serial.print(firstVal, 6);
 Serial.print("'");
 Serial.print(" http GET ");
```

```
  Serial.println(secondVal);
}
void setup()
{
  MQTT_server_onData(data_callback);                    // Register the callback
  Serial.println("Starting MQTT broker");               // Start the broker
  MQTT_server_start(mqttPort, max_subscriptions, max_retained_topics);
  MQTT_local_subscribe((unsigned char *)"#", 0);        // Subscribe to anything
}
void loop()
{
  String B(Cal, 6);
  String C(B + "," + httpCode);
  String myData(C);
  MQTT_local_publish((unsigned char *)"/MyBroker/waterunit", (unsigned char
*)myData.c_str(), myData.length(), 0, 0);          // Publish the counter value as String
  Serial.println();
}
```

4. ส่วนรับค่ายูนิตของ NodeMCU อื่นด้วยโปรโตคอล MQTT

```
void myDataCb(String& topic, String& data);
void myPublishedCb();
void myDisconnectedCb();
void myConnectedCb();
#define CLIENT_ID "client1"
#define TOPIC "/MyBroker/powerunit"
MQTT myMqtt(CLIENT_ID, "192.168.1.148", 1884);
void setup()
{
  Serial.println("Connecting to MQTT server");
  // setup callbacks
  myMqtt.onConnected(myConnectedCb);
  myMqtt.onDisconnected(myDisconnectedCb);
  myMqtt.onPublished(myPublishedCb);
  myMqtt.onData(myDataCb);
  Serial.println("connect mqtt...");
  myMqtt.connect();
  Serial.println("subscribe to topic...");
  myMqtt.subscribe(TOPIC);
  delay(10);
}
void loop()
{
}
void myConnectedCb()
{
  Serial.println("connected to MQTT server");
}
```

```
void myDisconnectedCb()
{
  Serial.println("disconnected. try to reconnect...");
  delay(500);
  myMqtt.connect();
}
void myPublishedCb()
{
}
void myDataCb(String& topic, String& data)
{
  String Vall = data;
  float firstVall;
  String secondVall;
  for (int i = 0; i < Vall.length(); i++)
  {
    if (Vall.substring(i, i + 1) == ",") {
      firstVall = Vall.substring(0, i).toFloat();
      secondVall = Vall.substring(i + 1).toInt();
      break;
    }
  }
  Serial.print("recieved topic '");
  Serial.print(topic);
  Serial.print("' with data '");
  Serial.print(firstVall, 6);
  Serial.print("'");
  Serial.print(" http GET ");
  Serial.println(secondVall);
  if (secondVall != "200")
```

```
  {
    unit = firstVall;
    if ((WiFiMulti.run() == WL_CONNECTED)) {
      HTTPClient http;
      url = "http://192.168.1.222/add.php?home=1&powerunit=" + String(unit, 6);
      http.begin(url);
      httpCode = http.GET();
      http.end();
    }
  }
}
```

ภาคผนวก ข.

โปรแกรม PHP ในส่วนการส่งค่ายูนิตน้ำประปาไปยังฐานข้อมูล

1. ส่วนรับข้อมูลจาก NodeMCU ส่งไปยังฐานข้อมูลด้วย sql

```
<?php
 $home = $_GET['home'];                    //รับค่าเพื่อแสดงว่าเป็ยบ้านหลังใด
 $waterunit = $_GET['waterunit'];          //รับค่ายูนิตน้ำประปาเก็บในตัวแปร
 mysql_connect("localhost", "root", "12345671") or die(mysql_error());
 mysql_select_db("Project4A") or die(mysql_error());
//เชื่อมต่อฐานข้อมูล
 $data = mysql_query("INSERT INTO WaterFlow(home,unit) VALUES
($home,$waterunit)") or die(mysql_error());   //ส่งข้อมูลไปยังฐานข้อมูล
?>
```

ภาคผนวก ค.

โปรแกรม NodeMCU ในส่วนระบบการส่งค่ายูนิตไฟฟ้า

1. ส่วนกำหนดตัวแปรและฟังก์ชันที่ใช้ในการอ่านค่าจาก Power meter module

```
#include <SoftwareSerial.h>
#include <PZEM004T.h>
#include <ESP8266WiFi.h>
#include <ESP8266WiFiMulti.h>
#include <ESP8266HTTPClient.h>
ESP8266WiFiMulti WiFiMulti;
#include <MQTT.h>
#include <uMQTTBroker.h>
PZEM004T pzem(13, 15);      //กำหนด (RX,TX) เพื่อเชื่อมต่อกับ TX,RX ของ Power meter
IPAddress ip(192, 168, 1, 1);
char ssid[] = "chenchan";      //your network SSID (name)
char pass[] = "150203LAN";     //your network password
int httpCode;
String url;
float p;
float Cal = 0.0;
float unit = 0.0;
float Sumwatt = 0.0;
int k=0;
void calc ()
{
  sei();                       //Enables interrupts
  for (k; k <= 59; k++)
  {
    p = pzem.power(ip);        //power every sec
    if (p < 0)
    {
      p = 0.0;
```

```
    }
    Sumwatt = Sumwatt + p;
    delay(1000);
  }
  cli();                          //Disable interrupts
}
void setup() {
  Serial.begin(115200);
  Serial.println();
  Serial.println();
  Serial.print("Connecting to ");
  Serial.println(ssid);
  WiFi.begin(ssid, pass);
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }
  Serial.println("");
  IPAddress local_ip = {192, 168, 1, 148};     //ตั้งค่า IP
  IPAddress gateway = {192, 168, 1, 1};        //ตั้งค่า IP Gateway
  IPAddress subnet = {255, 255, 255, 0};       //ตั้งค่า Subnet
  WiFi.config(local_ip, gateway, subnet);      //setค่าไปยังโมดูล
  Serial.println("WiFi connected");
  Serial.println("IP address: ");
  Serial.println(WiFi.localIP());
}
```

2. ส่วนคำนวณค่ายูนิตไฟฟ้าและส่งไปยังฐานข้อมูล

```
void loop() {
calc ();
  Serial.println("");
  if ((WiFiMulti.run() == WL_CONNECTED)) {
    HTTPClient http;
    unit = ((Sumwatt) * 2.2) / 3600000; //(sent every 1 min) : unit/min;
    Serial.print("Power unit = ");
    Serial.println(unit, 6);
    url = "http://192.168.1.222/add.php?home=1&powerunit=" + String(unit, 6);
    http.begin(url); //HTTP
    httpCode = http.GET();
    if (httpCode > 0) {
      Serial.printf("[HTTP] GET... code: %d\n", httpCode);
    } else {
      Serial.printf("[HTTP] GET... failed, error: %s\n",
http.errorToString(httpCode).c_str());
    }
    http.end();
  }
  Sumwatt = 0.0;
  k = 0;
  Serial.println("");
  }
```

3. ส่วนส่งค่ายูนิตไฟฟ้าด้วยโปรโตคอล MQTT

```
unsigned int mqttPort = 1884;                    // the standard MQTT broker port
unsigned int max_subscriptions = 30;
unsigned int max_retained_topics = 30;
void data_callback(uint32_t *client /* we can ignore this */, const char* topic,
uint32_t topic_len, const char *data, uint32_t lengh) {
  char topic_str[topic_len + 1];
  os_memcpy(topic_str, topic, topic_len);
  topic_str[topic_len] = '\0';
  char data_str[lengh + 1];
  os_memcpy(data_str, data, lengh);
  data_str[lengh] = '\0';
  String Val = data_str;
  float firstVal;
  String secondVal;
  for (int i = 0; i < Val.length(); i++)
  {
    if (Val.substring(i, i + 1) == ",") {
      firstVal = Val.substring(0, i).toFloat();
      secondVal = Val.substring(i + 1).toInt();
      break;
    }
  }
  Serial.print("sent topic '");
  Serial.print(topic_str);
  Serial.print("' with data '");
  Serial.print(firstVal, 6);
  Serial.print("'");
  Serial.print(" http GET ");
```

```
  Serial.println(secondVal);
}
void setup()
{
  MQTT_server_onData(data_callback);                    // Register the callback
  Serial.println("Starting MQTT broker");               // Start the broker
  MQTT_server_start(mqttPort, max_subscriptions, max_retained_topics);
  MQTT_local_subscribe((unsigned char *)"#", 0);        // Subscribe to anything
}
void loop()
{
  String B(unit, 6);
  String C(B + "," + httpCode);
  String myData(C);
  MQTT_local_publish((unsigned char *)"/MyBroker/waterunit", (unsigned char
*)myData.c_str(), myData.length(), 0, 0);          // Publish the counter value as String
  Serial.println();
}
```

4. ส่วนรับค่ายูนิตของ NodeMCU อื่นด้วยโปรโตคอล MQTT

```
void myDataCb(String& topic, String& data);
void myPublishedCb();
void myDisconnectedCb();
void myConnectedCb();
#define CLIENT_ID "client1"
#define TOPIC "/MyBroker/waterunit"
MQTT myMqtt(CLIENT_ID, "192.168.1.148", 1883);
void setup()
{
  Serial.println("Connecting to MQTT server");
  // setup callbacks
  myMqtt.onConnected(myConnectedCb);
  myMqtt.onDisconnected(myDisconnectedCb);
  myMqtt.onPublished(myPublishedCb);
  myMqtt.onData(myDataCb);
  Serial.println("connect mqtt...");
  myMqtt.connect();
  Serial.println("subscribe to topic...");
  myMqtt.subscribe(TOPIC);
  delay(10);
}
void loop()
{
}
void myConnectedCb()
{
  Serial.println("connected to MQTT server");
}
```

```
void myDisconnectedCb()
{
  Serial.println("disconnected. try to reconnect...");
  delay(500);
  myMqtt.connect();
}
void myPublishedCb()
{
}
void myDataCb(String& topic, String& data)
{
  String Vall = data;
  float firstVall;
  String secondVall;
  for (int i = 0; i < Vall.length(); i++)
  {
    if (Vall.substring(i, i + 1) == ",") {
      firstVall = Vall.substring(0, i).toFloat();
      secondVall = Vall.substring(i + 1).toInt();
      break;
    }
  }
  Serial.print("recieved topic '");
  Serial.print(topic);
  Serial.print("' with data '");
  Serial.print(firstVall, 6);
  Serial.print("'");
  Serial.print(" http GET ");
  Serial.println(secondVall);
  if (secondVall != "200")
```

```
  {
    Cal = firstVall;
    if ((WiFiMulti.run() == WL_CONNECTED)) {
      HTTPClient http;
      url = "http://192.168.1.222/add.php?home=1&waterunit=" + String(Cal, 6);
      http.begin(url);
      httpCode = http.GET();
      http.end();
    }
  }
}
```

ภาคผนวก ง.

โปรแกรม PHP ในส่วนการส่งค่ายูนิตไฟฟ้าไปยังฐานข้อมูล

1. ส่วนรับข้อมูลจาก NodeMCU ส่งไปยังฐานข้อมูลด้วย sql

```
<?php
$home = $_GET['home'];                          //รับค่าระบุว่าเป็นบ้านหลังใด
$powerunit = $_GET['powerunit'];                //รับค่ายูนิดไฟฟ้าเก็บในตัวแปร
mysql_connect("localhost", "root", "12345671") or die(mysql_error());
 mysql_select_db("Project4A") or die(mysql_error());
//เชื่อมต่อฐานข้อมูล
 $data = mysql_query("INSERT INTO PowerUnit(home,unit) VALUES
($home,$powerunit)") or die(mysql_error());  //ส่งข้อมูลไปยังฐานข้อมูล
?>
```

ภาคผนวก จ.

โปรแกรม PHP ในส่วนการแสดงผลบนหน้าต่างเว็บเบราว์เซอร์

1. ส่วน Update ข้อมูลและคำนวณค่าน้ำประปะและค่าไฟฟ้า

```
<?php
mysql_connect("localhost", "root", "12345671") or die(mysql_error());
mysql_select_db("Project4A") or die(mysql_error());
$homepointer= 1;
  $set = mysql_query("UPDATE SUMUNIT SET sumpowerunit = 0 WHERE
home=$homepointer") or die(mysql_error());
  $set = mysql_query("UPDATE SUMUNIT SET powerbill = 0 WHERE
home=$homepointer") or die(mysql_error());
  $set = mysql_query("UPDATE SUMUNIT SET sumwaterflow = 0 WHERE
home=$homepointer") or die(mysql_error());
  $set = mysql_query("UPDATE SUMUNIT SET waterbill = 0 WHERE
home=$homepointer") or die(mysql_error());
  $data = mysql_query("SELECT year(timedate) as y, month(timedate) as m,
sum(unit) as sumunit from PowerUnit where home=$homepointer group by
year(timedate),month(timedate)") or die(mysql_error());
    while($info = mysql_fetch_array($data))
   { $sumunit = $info['sumunit'];
     if ($sumunit <= 15)
     {$price = ($sumunit*2.3488) + 8.19 ;}
     elseif ($sumunit <= 25)
     {$price = (($sumunit-15) * 2.9882) + 8.19 +35.232;}
     elseif ($sumunit <= 35)
     {$price = (($sumunit-25) * 3.2405) + 8.19 +65.114;}
     elseif ($sumunit <= 100)
     {$price = (($sumunit-35) * 3.6237) + 8.19 +97.519;}
     elseif ($sumunit <= 150)
     {$price = (($sumunit-100) * 3.7171) + 8.19 +333.0595;}
     elseif ($sumunit <= 400)
```

```
{$price = (($sumunit-150) * 4.2218) + 8.19 +518.9145;}
elseif ($sumunit > 400)
{$price = (($sumunit-400) * 4.4217) + 8.19 +1574.3645;}
$pricee=$price + ($price*0.07);
$update = mysql_query("UPDATE SUMUNIT SET sumpowerunit = ".$sumunit."
WHERE home=$homepointer y= ".$info['y']." and m= ".$info['m']) or
die(mysql_error());
$updatee = mysql_query("UPDATE SUMUNIT SET powerbill = ".$pricee." WHERE
home=$homepointer y= ".$info['y']." and m= ".$info['m']) or die(mysql_error());}
$dataa = mysql_query("SELECT year(timedate) as y, month(timedate) as m,
sum(unit) as sumunit from WaterFlow where home=$homepointer group by
year(timedate),month(timedate)") or die(mysql_error());
while($info = mysql_fetch_array($dataa))
{ $sumunit = $info['sumunit'];
if ($sumunit <= 30)
{$price = ($sumunit*8.50) ;}
elseif ($sumunit <= 40)
{$price = (($sumunit-30) * 10.03) + 255;}
elseif ($sumunit <= 50)
{$price = (($sumunit-40) * 10.35) + 355.3;}
elseif ($sumunit <= 60)
{$price = (($sumunit-50) * 10.68) + 462.1;}
elseif ($sumunit <= 70)
{$price = (($sumunit-60) * 11.00) + 568.9;}
elseif ($sumunit <= 80)
{$price = (($sumunit-70) * 11.33) + 678.9;}
elseif ($sumunit <= 90)
{$price = (($sumunit-80) * 12.50) + 792.2;}
elseif ($sumunit <= 100)
{$price = (($sumunit-90) * 12.82) + 917.2;}
```

```
    elseif ($sumunit <= 120)
    {$price = (($sumunit-100) * 13.15) + 1045.4;}
    elseif ($sumunit <= 160)
    {$price = (($sumunit-120) * 13.47) + 1308.4;}
    elseif ($sumunit <= 200)
    {$price = (($sumunit-160) * 13.80) + 1847.2;}
    elseif ($sumunit > 200)
    {$price = (($sumunit-200) * 14.45) + 2399.2;}
    $pricee=$price + ($price*0.07);
  $update = mysql_query("UPDATE SUMUNIT SET sumwaterflow = ".$sumunit."
WHERE home=$homepointer y= ".$info['y']." and m= ".$info['m']) or
die(mysql_error());
  $updatee = mysql_query("UPDATE SUMUNIT SET waterbill = ".$pricee." WHERE
home=$homepointer y= ".$info['y']." and m= ".$info['m']) or die(mysql_error());}
?>
```

2. ส่วนแสดงผลตารางและกราฟการใช้ไฟฟ้าและน้ำประปาในเดือนปัจจุบัน

```
<?php
include("new.php");
include("noti.php");
include("notify.php");
?>
<html><head><title>HOME</title>
<meta http-equiv="refresh" content="90">
</head><body>
  <nav class="navbar navbar-toggleable-md navbar-light bg-faded"
style="background-color: #e3f2fd;">
  <button class="navbar-toggler navbar-toggler-right" type="button" data-
toggle="collapse" data-target="#navbarNavAltMarkup" aria-
controls="navbarNavAltMarkup" aria-expanded="false" aria-label="Toggle navigation">
  <span class="navbar-toggler-icon"></span></button>
  <a class="navbar-brand" href="home.php">Home</a>
  <div class="collapse navbar-collapse" id="navbarNavAltMarkup">
    <div class="navbar-nav">
      <a class="nav-item nav-link" href="show.php">For Year</a>
      <a class="nav-item nav-link" href="show2.php">For Month</a>
      <a class="nav-item nav-link" href="show3.php">For Day</a>
      <a class="nav-item nav-link" href="notification.php">Notification</a>
      <?php
      $data = mysql_query("SELECT * from Pointer WHERE 1 ") or die(mysql_error());
      while($info = mysql_fetch_array($data))
          { $HH = $info['home'];}
          if($HH==0){
      print "<a class='nav-item nav-link' href='../intro.php'>Back to Admin</a>"; }
      ?>
```

```
      <a class="nav-item nav-link" href="../index.php">Logout</a>
    </div></div></nav>
<?php
$homepointer= 1;
$updatetime =date('Y-m-d H:i:s');
$YY =date('Y');
$MM =date('M');
$data = mysql_query("SELECT * from SUMUNIT WHERE y=year('$updatetime') and
m=month('$updatetime') and home=$homepointer") or die(mysql_error());
while($info = mysql_fetch_array($data))
{
print "<br><center><h3>For ".$MM." Year ".$YY."</h3><br>";
print "<table width=70% border=1 >";
printb"<tr><td>Year</td><td>".$info['y']."</td></tr><tr><td>Month</td><td>".$info[
'Month']."</td></tr><tr><td>Sum of Power Unit</td><td>".$info['sumpowerunit']."
unit</td></tr><tr><td>Power Bill</td><td>".$info['powerbill']."
baht</td></tr><tr><td>Sum of Water Unit</td><td>".$info['sumwaterflow']."
unit</td></tr><tr><td>Water Bill</td><td>".$info['waterbill']." baht</td></tr>";
}
print "</table></center>";
?><br><center>
<button onclick="myFunction()">Show Power Unit Graph</button>
<button onclick="myFunction2()">Show Water Unit Graph</button>
<div id="chart_div" style="width: 90%; height: 500px;"></div>
<div id="chart_div2" style="width: 90%; height: 500px;"></div></center>
<script type="text/javascript"
src="https://www.gstatic.com/charts/loader.js"></script>
    <script type="text/javascript">
      function myFunction() {
      google.charts.load('current', {'packages':['corechart', 'line']});
```

```
      google.charts.setOnLoadCallback(drawChart);
      function drawChart() {
        var data = google.visualization.arrayToDataTable([
          <?php
          $data = mysql_query("SELECT * , day(timedate) as d , hour(timedate) as h ,
minute(timedate) as m FROM PowerUnit WHERE year(timedate)=year('$updatetime')
and month(timedate)=month('$updatetime') and home=$homepointer") or
die(mysql_error());
          print "['Time', 'Powerunit'],";
          while($info = mysql_fetch_array($data))
          {
            print "['" . $info['d'] ." ". $info['h'] . ":" . $info['m'] . "'," . $info['unit'] . "],";
          }
          ?>
        ]);
        var options = {title: 'POWER UNIT',
          hAxis: {title: 'date time (day hour:min) \n Graph showing power usage (unit)
for each time period.',  titleTextStyle: {color: '#333'} , format: '####.######'},
          vAxis: {minValue: 0,format: '####.######'}
        };
        var chart = new
google.visualization.LineChart(document.getElementById('chart_div'));
        chart.draw(data, options);
      }
    }
    </script>
    <script type="text/javascript">
      function myFunction2() {
      google.charts.load('current', {'packages':['corechart', 'line']});
      google.charts.setOnLoadCallback(drawChart);
```

```
      function drawChart() {
        var data = google.visualization.arrayToDataTable([
          <?php
          $data = mysql_query("SELECT * , day(timedate) as d , hour(timedate) as h ,
minute(timedate) as m FROM WaterFlow WHERE year(timedate)=year('$updatetime')
and month(timedate)=month('$updatetime') and home=$homepointer") or
die(mysql_error());
          print "['Time', 'Waterunit'],";
          while($info = mysql_fetch_array($data))
          {
            print "['" . $info['d'] ." ". $info['h'] . ":" . $info['m'] . "'," . $info['unit'] . "],";
          }
          ?>
        ]);
        var options = {title: 'WATER UNIT',
          hAxis: {title: 'date time (day hour:min) \n Graph showing water usage (unit) for
each time period.',  titleTextStyle: {color: '#333'} ,format: '####.######'},
          vAxis: {minValue: 0 ,format: '####.######'}
        };
        var chart = new
google.visualization.LineChart(document.getElementById('chart_div2'));
        chart.draw(data, options);
      }
    }
</script></body></html>
```

3. ส่วนแสดงผลตารางและกราฟการใช้ไฟฟ้าและน้ำประปารายปี

```
<?php
include("new.php");
include("noti.php");
include("notify.php");
?>
<html>
<head>
<title>For Year</title>
</head>
<body>
<nav class="navbar navbar-toggleable-md navbar-light bg-faded" style="background-
color: #e3f2fd;">
  <button class="navbar-toggler navbar-toggler-right" type="button" data-
toggle="collapse" data-target="#navbarNavAltMarkup" aria-
controls="navbarNavAltMarkup" aria-expanded="false" aria-label="Toggle navigation">
    <span class="navbar-toggler-icon"></span>
  </button>
  <a class="navbar-brand" href="home.php">Home</a>
  <div class="collapse navbar-collapse" id="navbarNavAltMarkup">
    <div class="navbar-nav">
      <a class="nav-item nav-link active" href="show.php">For Year</a>
      <a class="nav-item nav-link" href="show2.php">For Month</a>
      <a class="nav-item nav-link" href="show3.php">For Day</a>
      <a class="nav-item nav-link" href="notification.php">Notification</a>
      <?php
      $data = mysql_query("SELECT * from Pointer WHERE 1 ") or die(mysql_error());
      while($info = mysql_fetch_array($data))
          {
```

```
            $HH = $info['home'];
          }
          if($HH==0){
      print "<a class='nav-item nav-link' href='../intro.php'>Back to Admin</a>"; }
      ?>
      <a class="nav-item nav-link" href="../index.php" >Logout</a>
    </div>
  </div>
</nav>
<?php
$homepointer= 1;
$sY1 = date('Y');
if ($_SERVER["REQUEST_METHOD"] == "POST") {
  $sY1 = test_input($_POST["sY1"]);
}
function test_input($data) {
  $data = trim($data);
  $data = stripslashes($data);
  $data = htmlspecialchars($data);
  return $data;
}
?><br>
<form method="post" action="<?php echo
htmlspecialchars($_SERVER["PHP_SELF"]);?>">
<select name='sY1' id='sY1'>
<?php
$xYear=date('Y');
echo '<option value="'.$xYear.'">Select Year</option>';
echo '<option value="'.$xYear.'">'.$xYear.'</option>'; // ปีปัจจุบัน
  for($i=1;$i<=10;$i++){
```

```
    echo '<option value="'.($xYear-$i).'">'.($xYear-$i).'</option>';
  }
?>
</select>
  <input type="submit" name="submit" value="Submit">
</form><br>
    <center><h3>Summary of Year <?php echo $sY1; ?></h3><br>
    <div id="barchart" style="width: 90%;"></div></center><br><br>
    <?php
    $data = mysql_query("SELECT * from SUMUNIT where home=$homepointer and
y=$sY1") or die(mysql_error());
    print "<center><h7>Graph showing power bill (baht) and water bill (baht) for
each month period.</h7><br><br><h5>Table</h5>";
print "<table width=80% border=1 >";
 print "<tr><td>Year</td><td>Month</td><td>Sum of Power Unit</td><td>Power
Bill</td><td>Sum of Water Unit</td><td>Water Bill</td></tr>";
 while($info = mysql_fetch_array($data))
 {
 print
"<tr><td>".$info['y']."</td><td>".$info['Month']."</td><td>".$info['sumpowerunit']."
unit</td><td>".$info['powerbill']." baht</td><td>".$info['sumwaterflow']."
unit</td><td>".$info['waterbill']." Baht</td></tr>";
}
print "</table><br><br><br></center>";
 ?>
    <script type="text/javascript"
src="https://www.gstatic.com/charts/loader.js"></script>
   <script type="text/javascript">
     google.charts.load('current', {'packages':['corechart','bar']});
     google.charts.setOnLoadCallback(drawChart);
```

```
    function drawChart() {
      var data = google.visualization.arrayToDataTable([
        <?php
        $data = mysql_query("SELECT * from SUMUNIT where home=$homepointer
and y='$sY1'") or die(mysql_error());
        print "['Month', 'Water Bill (baht)' , 'Power Bill (baht)'],";
        while($info = mysql_fetch_array($data))
        {
          print "['" . $info['Month'] . "'," . $info['waterbill'] . ",". $info['powerbill'] .'],';
        }
        ?>
      ]);
      var options = {
        vAxis: {format: '####.######' // show axis values to 3 decimal places
      },
        hAxis: {title: 'Month',
          viewWindow: {min: 0,max: 12},
        },
        bars: 'vertical' // Required for Material Bar Charts.
      };
      var chart = new google.charts.Bar(document.getElementById('barchart'));
      chart.draw(data, google.charts.Bar.convertOptions(options));
    }
  </script>
</body>
</html>
```

4. ส่วนแสดงผลตารางและกราฟการใช้ไฟฟ้าและน้ำประปารายวัน

```
<?php
include("new.php");
include("noti.php");
include("notify.php");
?>
<html><head><title>For Day</title></head> <body>
<nav class="navbar navbar-toggleable-md navbar-light bg-faded" style="background-
color: #e3f2fd;">
  <button class="navbar-toggler navbar-toggler-right" type="button" data-
toggle="collapse" data-target="#navbarNavAltMarkup" aria-
controls="navbarNavAltMarkup" aria-expanded="false" aria-label="Toggle navigation">
    <span class="navbar-toggler-icon"></span>
  </button>
  <a class="navbar-brand" href="home.php">Home</a>
  <div class="collapse navbar-collapse" id="navbarNavAltMarkup">
    <div class="navbar-nav">
      <a class="nav-item nav-link" href="show.php">For Year</a>
      <a class="nav-item nav-link" href="show2.php">For Month</a>
      <a class="nav-item nav-link active" href="show3.php">For Day</a>
      <a class="nav-item nav-link" href="notification.php">Notification</a>
      <?php
      $data = mysql_query("SELECT * from Pointer WHERE 1 ") or die(mysql_error());
      while($info = mysql_fetch_array($data))
          { $HH = $info['home'];}
          if($HH==0){
      print "<a class='nav-item nav-link' href='../intro.php'>Back to Admin</a>"; }
      ?>
      <a class="nav-item nav-link" href="../index.php" >Logout</a>
```

```
    </div>
  </div>
</nav>
<?php
$homepointer= 1;
$sY1 = date('Y');
$sM1 = date('m');
$sD1 = date('d');
if ($_SERVER["REQUEST_METHOD"] == "POST") {
  $sY1 = test_input($_POST["sY1"]);
  $sM1 = test_input($_POST["sM1"]);
  $sD1 = test_input($_POST["sD1"]);
}
function test_input($data) {
  $data = trim($data);
  $data = stripslashes($data);
  $data = htmlspecialchars($data);
  return $data;
}
?><br>
<form method="post" action="<?php echo
htmlspecialchars($_SERVER["PHP_SELF"]);?>">
<select name='sD1' id='sD1'>
<?php
echo '<option value="'.$sD1.'">Select Day</option>';
for($d=1;$d<=31;$d++){
  $d2=sprintf("%02d",$d);
  echo '<option value="'.$d2.'">'.$d2.'</option>';
 }?>
</select>
```

```
<select name='sM1' id='sM1'>
<?php
echo '<option value="'.$sM1.'">Select Month</option>';
for($i=1;$i<=12;$i++){
  $i2=sprintf("%02d",$i);
  echo '<option value="'.$i2.'">'.$i2.'</option>';
 }?>
</select>
<select name='sY1' id='sY1'>
<?php
$xYear=date('Y');
echo '<option value="'.$sY1.'">Select Year</option>';
echo '<option value="'.$xYear.'">'.$xYear.'</option>';
  for($i=1;$i<=10;$i++){
    echo '<option value="'.($xYear-$i).'">'.($xYear-$i).'</option>'; }
?>
</select>
<input type="submit" name="submit" value="Submit">
</form><br>
<?php
print "<center><h3>For ".$sD1." Mouth ".$sM1." Year ".$sY1."</h3>";
$data1 = mysql_query("SELECT sum(unit) as p from PowerUnit WHERE
year(timedate) =$sY1 and month(timedate) =$sM1 and day(timedate) =$sD1 and
home=$homepointer") or die(mysql_error());
while($info = mysql_fetch_array($data1))
{
  if($info['p'] == ""){$p = 0;}else{$p = $info['p'];}
}
```

```
$data2 = mysql_query("SELECT sum(unit) as w from WaterFlow WHERE
year(timedate) =$sY1 and month(timedate) =$sM1 and day(timedate) =$sD1 and
home=$homepointer") or die(mysql_error());
while($info = mysql_fetch_array($data2))
{
  if($info['w'] == ""){$w = 0;}else{$w = $info['w'];}
}
print "<br><table width=70% border=1 >";
print "<tr><td>Sum of Power Unit</td><td>".$p." unit</td></tr><tr><td>Sum of
Water Unit</td><td>".$w." unit</td></tr>";
print "</table></center><br>";
?>
<center>
<button onclick="myFunction()">Show Power Unit Graph</button>
<button onclick="myFunction2()">Show Water Unit Graph</button>
<div id="chart_div" style="width: 90%; height: 500px;"></div>
<div id="chart_div2" style="width: 90%; height: 500px;"></div>
</center>
<script type="text/javascript"
src="https://www.gstatic.com/charts/loader.js"></script>
    <script type="text/javascript">
      function myFunction() {
      google.charts.load('current', {'packages':['corechart', 'line']});
      google.charts.setOnLoadCallback(drawChart);
      function drawChart() {
        var data = google.visualization.arrayToDataTable([
          <?php
          $data = mysql_query("SELECT * , day(timedate) as d , hour(timedate) as h ,
minute(timedate) as m FROM PowerUnit WHERE year(timedate)=$sY1 and
```

```
month(timedate)=$sM1 and day(timedate)=$sD1 and home=$homepointer") or
die(mysql_error());
          print "['Time', 'Powerunit'],";
          while($info = mysql_fetch_array($data))
          { print "['" . $info['d'] ." ". $info['h'] . ":" . $info['m'] . "'," . $info['unit'] . "],";}
          ?>
        ]);
        var options = {title: 'POWER UNIT',
          hAxis: {title: 'date time (day hour:min) \n Graph showing power usage (unit)
for each time period.',  titleTextStyle: {color: '#333'} , format: '####.######'},
          vAxis: {minValue: 0,format: '####.######'}
        };
        var chart = new
google.visualization.LineChart(document.getElementById('chart_div'));
        chart.draw(data, options);
      }
      }
    </script>
    <script type="text/javascript">
      function myFunction2() {
      google.charts.load('current', {'packages':['corechart', 'line']});
      google.charts.setOnLoadCallback(drawChart);
      function drawChart() {
        var data = google.visualization.arrayToDataTable([
          <?php
          $data = mysql_query("SELECT * , day(timedate) as d , hour(timedate) as h ,
minute(timedate) as m FROM WaterFlow WHERE year(timedate)=$sY1 and
month(timedate)=$sM1 and day(timedate)=$sD1 and home=$homepointer") or
die(mysql_error());
          print "['Time', 'Waterunit'],";
```

```
          while($info = mysql_fetch_array($data))
          { print "['" . $info['d'] ." ". $info['h'] . ":" . $info['m'] . "'," . $info['unit'] . "],";}
          ?>
        ]);
        var options = {title: 'WATER UNIT',
          hAxis: {title: 'date time (day hour:min) \n Graph showing water usage (unit) for
each time period.',  titleTextStyle: {color: '#333'} ,format: '####.######'},
          vAxis: {minValue: 0 ,format: '####.######'}
        };
        var chart = new
google.visualization.LineChart(document.getElementById('chart_div2'));
        chart.draw(data, options);
      }
    }
</script></body></html>
```

ภาคผนวก ฉ.

โปรแกรมสร้าง Mobile Application เว็บเบราว์เซอร์

1. ส่วน activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/activity_main"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context="com.example.user.webapp.MainActivity">
    <WebView
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        tools:layout_editor_absoluteX="8dp"
        tools:layout_editor_absoluteY="8dp"
        android:id="@+id/webView"
        />
</android.support.constraint.ConstraintLayout>
```

2. ส่วน MainActivity.java

```
package com.example.user.webapp;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.webkit.WebSettings;
import android.webkit.WebView;
import android.webkit.WebViewClient;
public class MainActivity extends AppCompatActivity {
    WebView webview;
    @Override
    public void onBackPressed (){
        if(webview.canGoBack()){
            webview.goBack();
        }else{
            super.onBackPressed();
        }
    }
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        webview = (WebView) findViewById(R.id.webView);
        WebSettings webSettings = webview.getSettings();
        webSettings.setJavaScriptEnabled(true);
        webview.setWebViewClient(new WebViewClient());
        webview.loadUrl("http://projectmeter.hopto.org");
    }
}
```