สมาร์ทมิเตอร์และบ้านอัจฉริยะ

SMART METERS AND SMART HOME

ปวเรศ มาตร์มงคล

ภาสกร เกตวัฒน์

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2556

สมาร์ทมิเตอร์และบ้านอัจฉริยะ

Smart Meters and Smart Home

ปวเรศ มาตร์มงคล

ภาสกร เกตวัลห์

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2556

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์ปีการศึกษา 2556

สาขาวิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง สมาร์ทมิเตอร์และบ้านอัจฉริยะ

Smart Meters and Smart Home

ผู้จัดทำ

1. นายปวเรศ มาตร์มงคล รหัสนักศึกษา 53010962
2. นายภาสกร เกตวัลห์ รหัสนักศึกษา 53011259

..... อาจารย์ที่ปรึกษา

(ผู้ช่วยศาสตราจารย์ ดร.วิศิษฏ์ หิรัญกิตติ)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สมาร์ทมิเตอร์และบ้านอัจฉริยะ

นาย ปวเรศ	มาตร์มงคล	53010962
นาย ภาสกร	เกตวัลห์	53011259
ผศ.ดร.วิศิษฏ์	หิรัญกิตติ	อาจารย์ที่ปรึกษา

ปีการศึกษา 2556

บทคัดย่อ

ปริญญานิพนธ์ฉบับนี้มีวัตถุประสงค์ในการพัฒนาโมบายล์แอพพลิเคชันที่ใช้อ่านและบันทึกค่าไฟฟ้าจากสมาร์ทมิเตอร์และควบคุมหลอดไฟภายในบ้าน โดยแอพพลิเคชันนี้จะเชื่อมต่อสื่อสารกับสมาร์ทมิเตอร์ ผ่านทางตัวกลางคือ เครื่องคอมพิวเตอร์ด้วย Wifi จากนั้นเครื่องคอมพิวเตอร์จะสื่อสารกับสมาร์ทมิเตอร์ด้วยโปรโทคอล DLMS/COSEM เพื่อนำข้อมูลที่ได้จากสมาร์ทมิเตอร์มาวิเคราะห์เพื่อการประหยัดพลังงานด้วยการควบคุมการทำงานของหลอดไฟภายในบ้าน และการที่โมบายล์แอพพลิเคชันจะสามารถควบคุมหลอดไฟภายในบ้านได้ จะต้องควบคุมผ่าน Bridge โดยจะเชื่อมต่อกับ Bridge ด้วย Wifi จากนั้น Bridge จะเชื่อมต่อกับหลอดไฟภายในบ้านที่มี ZigBee ติดตั้งอยู่ โดยอาศัยการสื่อสารด้วย ZigBee ผ่านโปรโทคอล Light Link โดยโมบายล์แอพพลิเคชันสามารถควบคุมการเปิด/ปิด ปรับความเข้มสี เปลี่ยนสี และ ปรับความสว่างของหลอดไฟได้ตามการสั่งงานของโปรแกรม รวมทั้งมีความสามารถในการกำหนดเงื่อนไขว่า เมื่อมีการใช้ไฟฟ้าภายในบ้านมีค่าพลังงานในช่วงเวลาหนึ่งเกินค่าที่ตั้งไว้ แล้วจะควบคุมให้ปิด หรือ หรี่หลอดไฟ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Smart Meters and Smart Home

Mr. Pawares	Matramongkol	53010962
Mr. Passakorn	Ketvan	53011259
Asst. Prof. Dr. Visit	Hirankitti	Advisor

Academic Year 2013

ABSTRACT

This thesis aims to develop mobile application which can read and record energy usage data from a smart meter and can control light bulbs in the house. This application reads the data from the smart meter wirelessly via a personal computer and the computer then communicates with the smart meter using the DLMS/COSEM protocol. The data received from the smart meter is analyzed in order to save energy by controlling the light bulbs. The mobile application can control light bulbs via the Bridge which communicates with mobile application via Wifi , but it communicates with light bulbs via ZigBee using the Light Link protocol. The mobile application can turn on/off the light, adjust colour density, change colour and adjust brightness of the light bulbs. In additions, the mobile application can set a condition for turning on/off or dimming the light bulbs when the energy consumption read from the smart meter is greater than a certain amount in a short period of time.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

โครงงานสมาร์ทมิเตอร์และบ้านอัจฉริยะฉบับนี้สำเร็จลุล่วงไปได้ด้วยดี เนื่องจากได้รับความช่วยเหลืออย่างดียิ่งจากอาจารย์ วิศิษฏ์ หิรัญกิตติ ซึ่งเป็นอาจารย์ที่ปรึกษาโครงงาน ผู้ซึ่งให้คำแนะนำ คำสั่งสอน ชี้แนะแนวทางการทำงาน ชี้ข้อปรับปรุงแก้ไข และให้คำปรึกษาที่ดีในเรื่องต่าง ๆ อย่างสม่ำเสมอ จึงทำให้โครงงานชิ้นนี้สำเร็จลุล่วงด้วยดี

ขอขอบคุณรุ่นพี่ รวมทั้งเพื่อน ๆ ทุกคนที่พร้อมให้คำปรึกษา แลกเปลี่ยนความคิดเห็น ให้กำลังใจและแนะนำการทำโครงงานในด้านต่าง ๆ จึงทำให้โครงงานนี้สำเร็จลุล่วง

ขอขอบพระคุณบิดามารดาและครอบครัวที่ให้การอบรมสั่งสอน เลี้ยงดู ให้โอกาสทางการศึกษา และให้การสนับสนุนในทุก ๆ ด้านมาโดยตลอด

สุดท้ายนี้ขอขอบพระคุณสถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบังที่ให้ข้าพเจ้าได้เข้ามาศึกษาหาความรู้ที่นี่ ข้าพเจ้ารู้สึกเป็นเกียรติอย่างยิ่ง คุณความดีใด ๆ ที่ปรากฏในโครงงานนี้ ข้าพเจ้าขอมอบแด่ผู้มีพระคุณทุกท่าน

นาย ปวเรศ มาตร์มงคล

นาย ภาสกร เกตวัลห์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

หน้า

บทคัดย่อภาษาไทย I

บทคัดย่อภาษาอังกฤษ II

กิตติกรรมประกาศ III

สารบัญ IV

สารบัญตาราง VII

สารบัญรูป VIII

บทที่ 1 บทนำ 1

1.1 ความสำคัญและที่มาของโครงงาน 1

1.2 วัตถุประสงค์ของโครงงาน 1

1.3 ขอบเขตของโครงงาน 2

1.4 วิธีการดำเนินการ 2

1.5 ประโยชน์ที่คาดว่าจะได้รับ 3

1.6 ส่วนประกอบของปริญญานิพนธ์ 3

บทที่ 2 ทฤษฎีที่เกี่ยวข้อง 4

2.1 สมาร์ทมิเตอร์ 4

2.2 Smart Home และ Home Automation 7

2.3 ZigBee 8

2.3.1 ZigBee Alliance 8

2.3.2 ZigBee Light Link 11

2.3.2.1 หลักการทำงานของ ZigBee Light Link 12

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3.2.2 การเริ่มการทำงานของ ZigBee Light Link 13
2.3.2.3 ZigBee Light Link Network Address 13
2.3.3 ZigBee Smart Energy 13
2.4 เทคโนโลยี Wireless LAN 14
2.5 DLMS/COSEM 15
2.6 การเขียนโปรแกรมเชื่อมต่อกับ Socket 18
2.7 Hue System 20
บทที่ 3 การออกแบบและพัฒนา 23
3.1 ความต้องการของระบบ 23
3.2 การออกแบบระบบ 24
3.2.1 Use case diagram 24
3.2.2 Class Diagram 26
3.2.3 ภาพรวมของระบบ 27
3.2.5 Flow chart ของฟังก์ชันตั้งค่าขีดจำกัดการใช้ไฟฟ้า 28
3.3 การออกแบบฐานข้อมูล 29
3.4 การพัฒนาแอพพลิเคชัน 30
3.4.1 การพัฒนาการเชื่อมต่อด้วย Socket 30
3.4.2 การพัฒนาส่วนการตั้ง Peak load 34
3.4.3 การพัฒนาส่วนควบคุมหลอดไฟ 35
3.5 Library ที่ใช้ในการพัฒนาแอพพลิเคชัน 37
3.5.1 TapKu Library 37
3.5.2 SDKWIzard ของ Philips HUE 37
บทที่ 4 การทดลองและผลการทดลอง 39

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.1 การทดลองการแสดงผลของแอพพลิเคชัน 39
4.1.1 การแสดงผลของหน้าต่างแสดงข้อมูลการใช้ไฟฟ้าค่าไฟฟ้า 39
4.1.2 หน้าแสดงข้อมูลการใช้ไฟฟ้าย้อนหลัง 40
4.1.3 หน้าแสดงข้อมูลการเงิน 41
4.1.4 หน้าแสดงการควบคุมหลอดไฟ 42
4.1.5 หน้าแสดงการตั้งค่า 43
4.2 ทดลองการอ่านค่าจากมิเตอร์ 44
4.3 ทดลองการควบคุมหลอดไฟผ่านแอพพลิเคชัน 48
4.4 การทดลองการตั้งค่าเวลาที่ใช้ในการรับค่าจากมิเตอร์ 55
บทที่ 5 บทสรุปและข้อเสนอแนะ 56
5.1 บทสรุปของโครงงาน 56
5.2 ปัญหาอุปสรรคและแนวทางการแก้ไข 56
5.3 แนวทางในการพัฒนาต่อ 57
บรรณานุกรม 58

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

ตารางที่ **หน้า**

ตารางที่ 3.2 ตาราง Meter 29

ตารางที่ 3.3 ตาราง User 29

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป

รูปที่ **หน้า**

รูปที่ 2.1 แสดงหลักการทำงานของ Smart Meter [2] 5

รูปที่ 2.2 โครงข่ายของ AMI 6

รูปที่ 2.3 สมาร์ทมิเตอร์ Home automation และ In-home display 8

รูปที่ 2.4 ย่านความถี่มาตรฐานของ ZigBee 9

รูปที่ 2.5 แสดง ZigBee Stack 9

รูปที่ 2.6 แสดง ZigBee Nodes 10

รูปที่ 2.7 แสดง ZigBee Mesh Network 11

รูปที่ 2.8 Message format ของ Light Link 11

รูปที่ 2.9 คำสั่งของ Light devices 12

รูปที่ 2.10 คำสั่งของ Controller devices 12

รูปที่ 2.11 Zigbee Smart Energy Profile 14

รูปที่ 2.12 Smart Meters เชื่อมต่อกับ Data collector ด้วยโปรโตคอล DLMS/COSEM 16

รูปที่ 2.13 DLMS/COSEM เมื่อเปรียบเทียบกับ OSI model 17

รูปที่ 2.14 Socket 18

รูปที่ 2.15 แสดง Socket workflow 19

รูปที่ 2.13 ภาพรวมของ Hue System 20

รูปที่ 2.14 โครงสร้างของ Hue System ในส่วนของ Software 22

รูปที่ 3.1 แสดง Use case diagram ของระบบ 24

รูปที่ 3.2 Class Diagram ของระบบ 26

รูปที่ 3.3 ภาพรวมของระบบ 27

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 3.4 Flow chart การตั้งค่าขีดจำกัดการใช้ไฟฟ้า 28
รูปที่ 3.5 การเชื่อมต่อแบบ Socket 30
รูปที่ 3.6 การส่งข้อความ read ไปยัง host เพื่อที่จะอ่านค่า 31
รูปที่ 3.7 ส่วนที่ใช้เช็คว่ามีข้อมูลถูกส่งเข้ามาหรือไม่ 32
รูปที่ 3.8 ฟังก์ชันที่ใช้ปรับเวลาในการรับค่า 33
รูปที่ 3.9 การพัฒนาส่วนการตั้งค่า Peak load 34
รูปที่ 3.10 การสร้าง lightState 35
รูปที่ 3.11 Action เมื่อผู้ใช้งานกดส่ง lightState 36
รูปที่ 4.1 หน้าแสดงข้อมูลการใช้ไฟฟ้า 39
รูปที่ 4.2 หน้าแสดงข้อมูลการใช้ไฟฟ้าย้อนหลัง 40
รูปที่ 4.3 หน้าแสดงข้อมูลการเงิน 41
รูปที่ 4.4 หน้าแสดงการควบคุมหลอดไฟ 42
รูปที่ 4.5 หน้าแสดงการตั้งค่าผู้ใช้งาน 43
รูปที่ 4.6 ดู IP Address ของเครื่องคอมพิวเตอร์ 44
รูปที่ 4.7 ทำการแก้ไข IP Address 45
รูปที่ 4.8 การเก็บข้อมูลลง Core Data 46
รูปที่ 4.9 ผลลัพธ์ที่ได้จากการอ่านค่าจากสมาร์ทมิเตอร์ 47
รูปที่ 4.10 ทำการเชื่อมต่อ Bridge เข้ากับ Router 48
รูปที่ 4.11 การ Connect ระหว่าง Bridge และ Application 49
รูปที่ 4.12 หน้าต่าง Appliance 50
รูปที่ 4.13 หน้าต่าง Overview และ Control 51
รูปที่ 4.14 หน้าต่างรายชื่อหลอดไฟ 52
รูปที่ 4.15 หน้าต่างการควบคุมหลอดไฟ 53

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 4.16 ผลลัพธ์จากการทดลองการใช้งานควบคุมหลอดไฟ 54
รูปที่ 4.17 โค้ดที่ใช้ในการปรับค่าเวลา 55
รูปที่ 4.18 ทำการตั้งค่า Interval = 6 55
รูปที่ 4.19 ผลลัพธ์เมื่อทำการตั้งค่า Interval = 6 55

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

1.1 ความสำคัญและที่มาของโครงงาน

ทรัพยากรธรรมชาติในปัจจุบันนั้นมีการลดลงอย่างรวดเร็ว จากทั้งจำนวนประชากรที่เพิ่มขึ้นแล้ว เทคโนโลยีที่มีความล้ำหน้าทันสมัยมากขึ้น เพื่อที่จะรักษาทรัพยากรธรรมชาติเอาไว้ใช้ในอนาคตภายภาคหน้า จึงต้องมีกระบวนการหรือวิธีที่จะใช้ในการจัดการทรัพยากรอย่างมีประสิทธิภาพ โดยเริ่มจากการประหยัดไฟฟ้า เนื่องจาก ในการผลิตไฟฟ้านั้นต้องอาศัย การเผาผลาญเชื้อเพลิงเป็นจำนวนมาก ซึ่งเป็นสาเหตุหลักในการสูญเสียทรัพยากรธรรมชาติ

สมาร์ทมิเตอร์ จึงได้เข้ามีบทบาทในการจัดการไฟฟ้าได้อย่างมีประสิทธิภาพ ซึ่งตัวสมาร์ทมิเตอร์นี้มีความสามารถต่างๆมากมาย เช่น สามารถแสดงค่าการใช้งานไฟฟ้าต่างๆได้หลายค่า สามารถติดต่อสื่อสารเพื่อส่งข้อมูลการใช้งานไฟฟ้าได้ ทำให้สะดวกรวดเร็วกว่ามิเตอร์สมัยก่อน เนื่องจากไม่จำเป็นต้องใช้คนอ่าน และยังสามารถควบคุมปริมาณไฟได้โดยอัตโนมัติหากมีการใช้งานไฟฟ้าเกินกว่าที่กำหนด จึงเหมาะที่จะนำสมาร์ทมิเตอร์มาบริหารการใช้พลังงานไฟฟ้าภายในบ้านให้มีประสิทธิภาพได้ ในส่วนของโมบายล์แอพพลิเคชันพัฒนาขึ้นมาเพื่อใช้ในการอ่านข้อมูลจากสมาร์ทมิเตอร์เพื่อที่จะนำค่าไฟฟ้าต่างๆเหล่านั้นมาแสดงผล หรือนำมาใช้ในการควบคุมการใช้งานไฟฟ้า โดยค่าต่างๆเหล่านี้จะรับผ่านมาจากเครื่องคอมพิวเตอร์ ซึ่งเป็นตัวกลางในการเชื่อมต่อระหว่างโมบายล์แอพพลิเคชัน กับ สมาร์ทมิเตอร์ และ โมบายล์แอพพลิเคชันยังสามารถนำไปควบคุมอุปกรณ์ทางอิเล็กทรอนิกส์ เช่น หลอดไฟได้อีกด้วย

1.2 วัตถุประสงค์ของโครงงาน

1) เพื่อพัฒนาให้แอพพลิเคชันสามารถอ่านค่าไฟฟ้า ได้แก่ ค่าพลังงาน มีหน่วยเป็น kw ค่าพลังงานตีกลับ มีหน่วยเป็น kVAR ค่ายูนิต มีหน่วยเป็น kWh ค่าความต่างศักย์ไฟฟ้า มีหน่วยเป็น Volt และ ค่ากระแสไฟฟ้า มีหน่วยเป็น Ampere จากสมาร์ทมิเตอร์โดยผ่าน Wifi จากเครื่องคอมพิวเตอร์ที่เชื่อมกับสมาร์ทมิเตอร์ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2) เพื่อพัฒนาให้แอพพลิเคชันสามารถนำข้อมูลการใช้ไฟฟ้าจากสมาร์ทมิเตอร์มาประมวลผลเพื่อใช้ในการบริหารจัดการควบคุมการทำงานของหลอดไฟให้ใช้พลังงานอย่างมีประสิทธิภาพ
3) เพื่อพัฒนาให้แอพพลิเคชันสามารถควบคุมการทำงานของหลอดไฟด้วยการสื่อสารไร้สายแบบ ZigBee

1.3 ขอบเขตของโครงงาน

ขอบเขตของโครงงานนี้ครอบคลุมการพัฒนาโมบายล์แอพพลิเคชัน โดยใช้ภาษา Objective-C เพื่อใช้อ่านข้อมูลค่าไฟฟ้าจากสมาร์ทมิเตอร์ โดยใช้เครื่องคอมพิวเตอร์เป็นตัวกลาง โดยเครื่องคอมพิวเตอร์ จะติดต่อกับสมาร์ทมิเตอร์ผ่านโปรโทคอล DLMS/COSEM และโมบายล์แอพพลิเคชันจะติดต่อกับเครื่องคอมพิวเตอร์ผ่าน Wifi โดยโมบายล์แอพพลิเคชัน จะสามารถควบคุมการทำงานของหลอดไฟภายในบ้านโดยการสื่อสารผ่านเครือข่าย Wifi กับ Bridge โดย Bridge จะติดต่อสื่อสารกับหลอดไฟภายในบ้านผ่านการสื่อสารแบบ Zigbee ซึ่งติดตั้งอยู่ในหลอดไฟ ด้วยโปรโทคอล Light Link

1.4 วิธีการดำเนินการ

1) ศึกษาความรู้เกี่ยวกับ สมาร์มิเตอร์ และ การสื่อสารไร้สายแบบ ZigBee
2) ศึกษาภาษา Objective-C
3) ศึกษาการสร้างแอพพลิเคชันบนไอโอเอส
4) ศึกษาหลักการทำของของ Hue Bulb
5) ศึกษาหลักการ Network Programming แบบ TCP/IP
6) ศึกษาเกี่ยวกับโครงสร้างของไฟล์และฐานข้อมูล
7) พัฒนาแอพพลิเคชัน
8) พัฒนาส่วนติดต่อผู้ใช้งาน
9) ทดสอบผลการดำเนินงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.5 ประโยชน์ที่คาดว่าจะได้รับ

1) ช่วยประหยัดไฟฟ้า ส่งผลให้ช่วยประหยัดทรัพยากรธรรมชาติที่นำมาผลิตไฟฟ้า
2) สามารถควบคุมการใช้ไฟฟ้าได้แบบบอัตโนมัติเมื่อมีการใช้ไฟฟ้ามากเกินไปทำให้ประหยัดไฟฟ้า
3) สามารถดูการใช้งานไฟฟ้าย้อนหลังได้ ทำให้สามารถดูพฤติกรรมการใช้ไฟฟ้าของแต่ละครัวเรือนได้ ซึ่งมีประโยชน์ในการนำข้อมูลที่ได้ไปวิเคราะห์ ปริมาณการใช้ไฟฟ้าในแต่ละพื้นที่ต่อไป
4) ได้ความรู้ ความเข้าใจ ความสามารถ และ ได้ประสบการณ์ในการพัฒนาแอพพลิเคชันบนระบบปฏิบัติการไอโอเอส ซึ่งกำลังเป็นที่นิยมในปัจจุบัน
5) ได้ความรู้เกี่ยวกับ Smart meters และ Home automation ซึ่งมีผู้คนให้ความสนใจอยู่ในปัจจุบัน
6) สามารถนำแอพพลิเคชันนี้ไปต่อยอด เพื่อการนำไปใช้งานจริงในอนาคตภายหน้าได้

1.6 ส่วนประกอบของปริญญานิพนธ์

ปริญญานิพนธ์ฉบับนี้ได้แบ่งเนื้อหาโดยทั่วไปออกเป็น 5 บทด้วยกันคือ

บทที่ 1 บทนำ กล่าวถึงความสำคัญและที่มาของโครงงาน วัตถุประสงค์ของโครงงาน ขอบเขตของโครงงานวิธีการดำเนินการ ประโยชน์ที่คาดว่าจะได้รับ และส่วนประกอบของปริญญานิพนธ์

บทที่ 2 ทฤษฎีที่เกี่ยวข้อง กล่าวถึงพื้นฐานที่ใช้ในโครงงาน ประกอบด้วย สมาร์ทมิเตอร์ ,Home Automation, ZigBee, Wifi, Hue bulb และ โปรโทคอล DLMS/COSEM

บทที่ 3 การออกแบบและพัฒนา จะกล่าวถึงความต้องการของระบบ ภาพรวมของระบบทั้งฮาร์ดแวร์และซอร์ฟแวร์ การออกแบบระบบ ตั้งแต่ส่วนการติดต่อสื่อการ ส่วนการแสดงผล ส่วนวิเคราะห์และแจ้งเตือนผู้ใช้งาน และ ส่วนติดต่อกับผู้ใช้

บทที่ 4 การทดลองและผลการทดลอง กล่าวถึงการติดตั้งใช้งานระบบ ประกอบไปด้วย สมาร์ทมิเตอร์ เครื่องคอมพิวเตอร์ และ แอพพลิเคชัน

บทที่ 5 บทสรุปและข้อเสนอแนะ กล่าวถึงบทสรุป ปัญหาที่พบ วิธีการแก้ไขปัญหา และสุดท้ายแนวทางในการพัฒนาต่อ กล่าวถึงแนวทางในการพัฒนาต่อของแอพพลิเคชัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2
ทฤษฎีที่เกี่ยวข้อง

2.1 สมาร์ทมิเตอร์

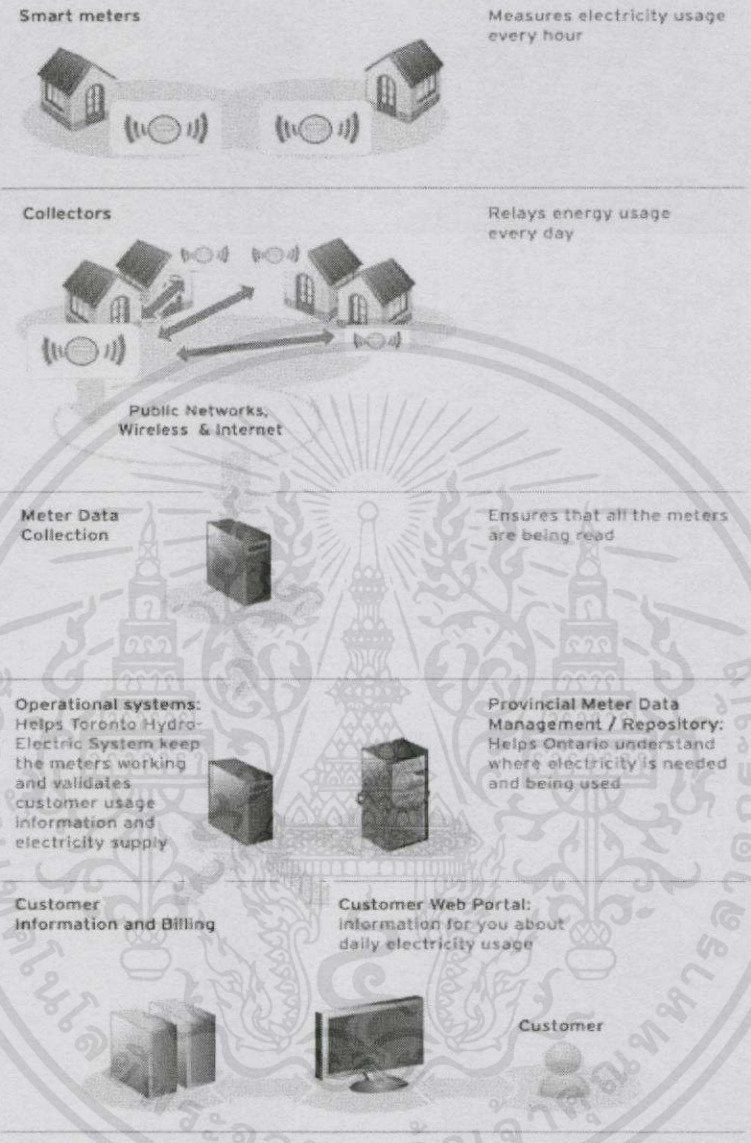
AMI (Advanced Metering Infrastructure) คือ ระบบการวัดค่าไฟฟ้าและการบริหารจัดการพลังงาน ประกอบไปด้วย สมาร์ทมิเตอร์ ส่วนรวบรวมข้อมูล และ ส่วนควบคุมกลาง

สมาร์ทมิเตอร์ คือมิเตอร์วัดพลังงานไฟฟ้าชนิดใหม่ที่ถูกพัฒนามาเพื่อทดแทนมิเตอร์แบบเดิมที่มีอยู่ทั่วไป โดยข้อดีคือ มีความสามารถที่จะแสดงผลข้อมูลการใช้งานได้มากกว่ามิเตอร์ที่มีอยู่ในปัจจุบัน และไม่ต้องใช้คนในการอ่านค่าจากมิเตอร์ สมาร์ทมิเตอร์สามารถสื่อสารได้ 2 ทางคือ สื่อสารกับศูนย์กลางระบบ และ โครงข่ายระบบสื่อสารภายในบ้าน

สมาร์ทมิเตอร์จะทำการอ่านค่าปริมาณการใช้ไฟฟ้าของผู้ใช้บริการ แล้วส่งข้อมูลไปยังศูนย์รวบรวมข้อมูล เพื่อที่จะส่งไปให้ศูนย์กลางระบบประมวลผลต่อไป การประมวลผลก็ตัวอย่างเช่น ผู้ใช้บริการไม่จ่ายค่าไฟฟ้า ส่วนควบคุมกลางก็จะสามารถสั่งการให้สมาร์ทมิเตอร์หยุดการจ่ายไฟได้ นอกจากนี้ส่วนควบคุมกลางยังสามารถที่จะ ปรับคาบเวลาในการวัด และเวลาในการบันทึกข้อมูลของสมาร์ทมิเตอร์ได้อีกด้วย

สมาร์ทมิเตอร์สามารถเชื่อมต่อกับโครงข่ายสื่อสารในบ้าน (HAN: Home Area Network) เพื่อเชื่อมต่อกับระบบ Home Automation โดยสามารถที่จะตรวจวัดปริมาณการใช้ไฟฟ้า และสามารถนำข้อมูลที่ได้นั้นไปประมวลผล เพื่อใช้ในการควบคุมเครื่องใช้ไฟฟ้าอย่างประหยัด และมีประสิทธิภาพต่อไป เช่นการตั้งขีดจำกัดในการใช้พลังงานไฟฟ้า เมื่อค่าพลังงานไฟฟ้าที่อ่านได้จากสมาร์ทมิเตอร์ในขณะนั้นสูงเกินกว่าปริมาณที่กำหนดเอาไว้ สักระยะเวลาหนึ่ง ระบบก็จะสามารถสั่งปิดเครื่องใช้ไฟฟ้าได้ เป็นต้น [1]

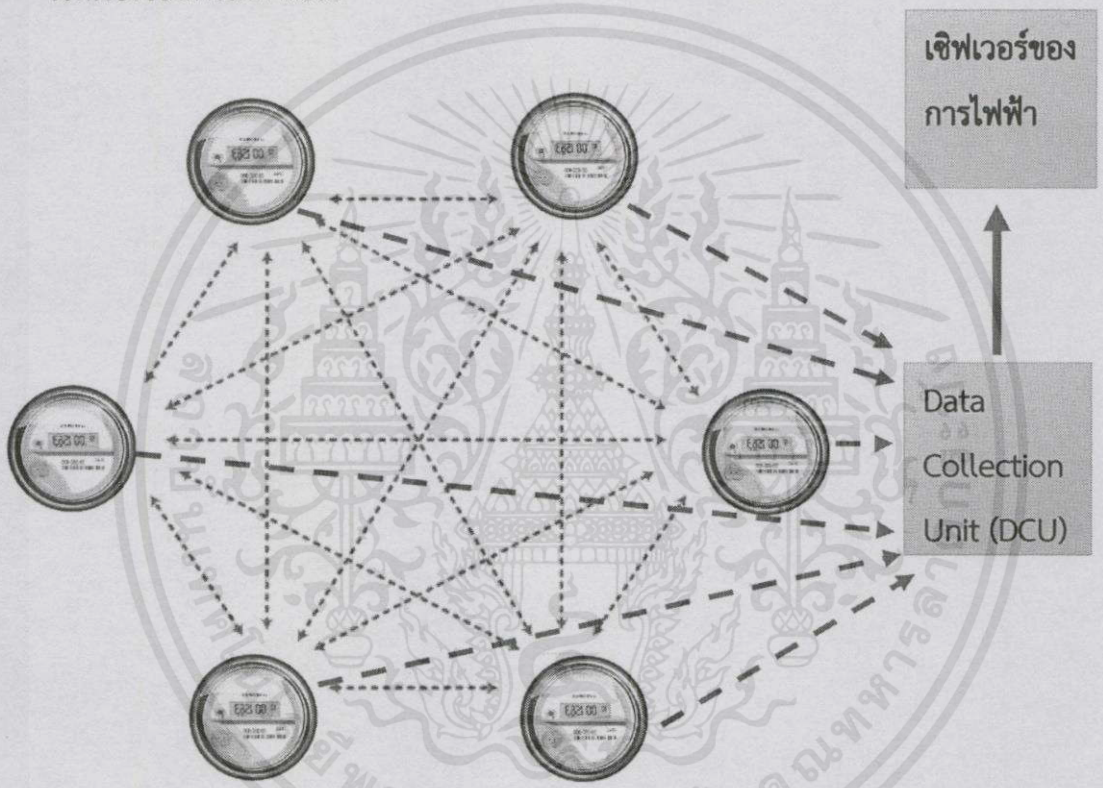
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.1 แสดงหลักการทำงานของ Smart Meter [2]

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สมาร์ทมิเตอร์เมื่อมีจำนวนมาก จะสามารถเชื่อมต่อกันเป็นโครงข่ายไร้สายของสมาร์ทมิเตอร์ขึ้นมาได้ โดยเป็นโครงข่ายที่ใช้เทคโนโลยี Zigbee โครงข่ายที่ใช้เทคโนโลยี Zigbee นั้นจะสามารถเชื่อมต่อกันด้วย topology แบบ Mesh กับสมาร์ทมิเตอร์ตัวอื่นๆ เพื่อสามารถแลกเปลี่ยนข้อมูลกันได้โดยตรง โดยมีความสามารถที่จะปรับ network ใหม่ เมื่อมีการเพิ่ม หรือ ลดจำนวนของสมาร์ทมิเตอร์ ข้อมูลการใช้ไฟฟ้าที่ได้จากสมาร์ทมิเตอร์จะถูกส่งไปที่ Data collection unit เพื่อที่จะส่งไปที่เซิฟเวอร์ของการไฟฟ้าต่อไป



รูปที่ 2.2 โครงข่ายของ AMI

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2 Smart Home และ Home Automation

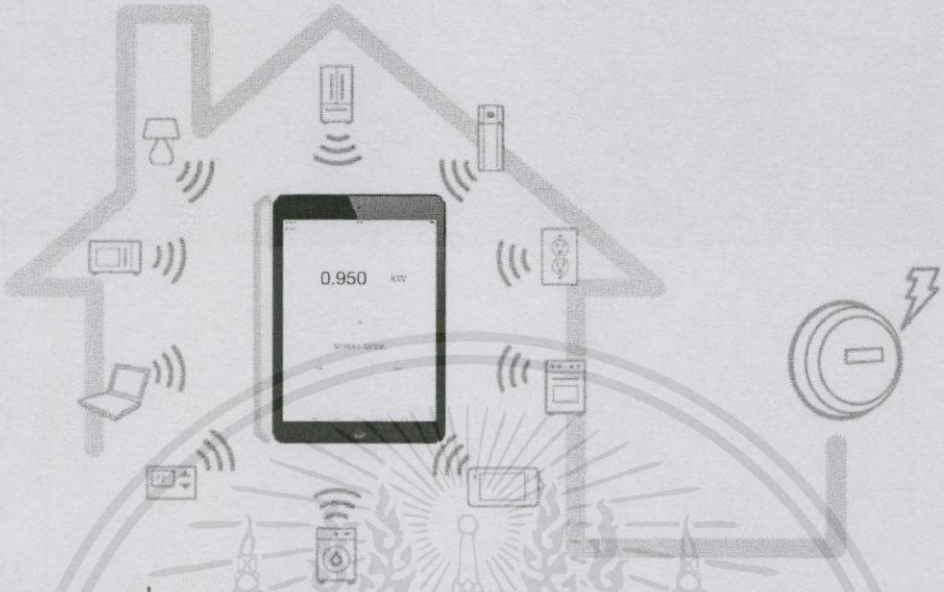
Smart home หรือบ้านอัจฉริยะ คือการใช้เทคโนโลยีที่มีอยู่ในปัจจุบันมาควบคุม สั่งการอุปกรณ์เครื่องใช้ไฟฟ้าที่ติดตั้งภายในบ้านให้ทำงานแบบอัตโนมัติ หรือทำงานตามที่เราสั่ง โดยจะมีความสามารถในการคิด วิเคราะห์สถานการณ์ต่างๆที่เกิดขึ้นจากสิ่งแวดล้อมรอบตัวได้ ทั้งนี้มีจุดประสงค์ในการออกแบบเพื่อให้มีความสะดวกสบายในการใช้งานเครื่องใช้ไฟฟ้า มีความปลอดภัย รวมถึงจุดประสงค์หลักเลยก็คือ เป็นมิตรกับสิ่งแวดล้อม และการประหยัดพลังงาน สมาร์ทมิเตอร์จึงเข้ามามีบทบาทช่วยให้บ้านอัจฉริยะสามารถประหยัดพลังงานได้ โดยการตรวจวัดค่าปริมาณการใช้ไฟฟ้า และนำมาประมวลผลตามเงื่อนไขที่ได้กำหนดไว้ เพื่อทำการควบคุมเครื่องใช้ไฟฟ้าต่อไป

โดยทั่วไปแล้วบ้านอัจฉริยะจะควบคุมด้วยระบบคอมพิวเตอร์ เรียกว่า Home Automation ซึ่งเปรียบเสมือนศูนย์กลางการควบคุมอุปกรณ์ต่างๆ

ตัวอย่างเทคโนโลยีที่นำมาประยุกต์ใช้กับ Home Automation

- ระบบตรวจสอบและควบคุมการใช้ไฟฟ้า เพื่อควบคุมปริมาณการใช้พลังงานและตรวจสอบความต้องการพลังงานไฟฟ้าของแต่ละอุปกรณ์
- ระบบการวัดอุณหภูมิในจุดต่างๆทั้งภายนอกและภายในบ้าน เพื่อปรับอุณหภูมิภายในบ้านให้เหมาะสม
- ระบบการควบคุมการเปิด-ปิดประตู หน้าต่าง และพัดลมระบายอากาศ โดยใช้การเปรียบเทียบระดับเอนทาลปีภายในและภายนอก เพื่อควบคุมการนอากาศจากภายนอกเข้ามาใช้
- ระบบควบคุมการจ่ายลมเย็นตามภาระความเย็นที่เกิดขึ้นในแต่ละบริเวณ
- ระบบควบคุมการเปิด-ปิดไฟแสงสว่างตามช่วงเวลา และการใช้งานในแต่ละพื้นที่
- ระบบควบคุมและสัญญาณเตือนอัตโนมัติ เช่น ระบบรดน้ำต้นไม้อัตโนมัติ ระบบสัญญาณกันขโมย เป็นต้น [3]

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.3 สมาร์ทมิเตอร์ Home automation และ In-home display

2.3 ZigBee

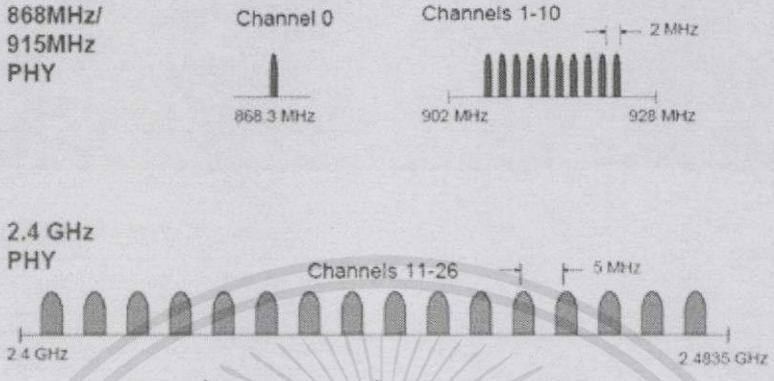
2.3.1 ZigBee Alliance

ZigBee Alliance ก่อตั้งขึ้นเมื่อปี 2002 โดยเป็นสมาคมที่ไม่แสวงหาผลกำไรที่พยายามพัฒนาระบบนิเวศของโลกให้เจริญรุ่งเรือง ทุกๆคนสามารถเข้าร่วมในสมาคมได้ทั้งสิ้น ในปัจจุบัน ZigBee Alliance ได้สร้างกลุ่มของนวัตกรรมที่ เชื่อถือได้ ใช้งานง่าย นั่นก็คือ มาตรฐาน ZigBee โดยจากการพัฒนานั้น ผลลัพธ์ออกมาเป็นดังนี้ โดย ZigBee สามารถ :

- ทำระบบ ตรวจวัด สั่งการ ให้มีการใช้พลังงานที่ต่ำ มีมาตรฐานเครือข่ายไร้สายที่เป็นสากล และ เป็นมิตรต่อสิ่งแวดล้อม
- ช่วยให้อุปกรณ์ใช้ไฟจากแบตเตอรี่ได้เป็นปี ด้วยการเชื่อมต่อแบบไร้สายที่มีพลังงานต่ำ
- สามารถเชื่อมต่ออุปกรณ์ที่แตกต่างกันมาอยู่ในเครือข่ายเดียวกันได้
- ง่ายในการ Set up และบำรุงรักษา

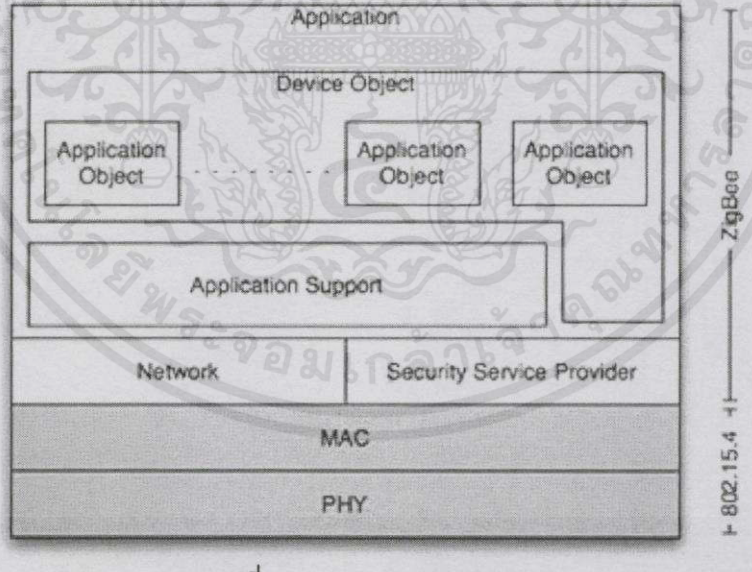
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ZigBee กำหนด ย่านความถี่ใช้งานตามมาตรฐานคือ ย่าน 2.4 Ghz



รูปที่ 2.4 ย่านความถี่มาตรฐานของ ZigBee

ZigBee นำ Physical Layer และ MAC Layer ของ IEEE 802.15.4 ซึ่งเป็นมาตรฐานการกำหนดการสื่อสารไร้สายแบบ WPAN (Wireless Personal Area Network) มาทำงานใน Layer ที่ต่ำกว่า (2 Layer ล่างสุด) เช่น เรื่องของ ระดับกำลังสัญญาณ , Link Quality , Access control , Security ฯลฯ แต่ใน Layer ถัดไปจะเป็นรูปแบบของ ZigBee



รูปที่ 2.5 แสดง ZigBee Stack

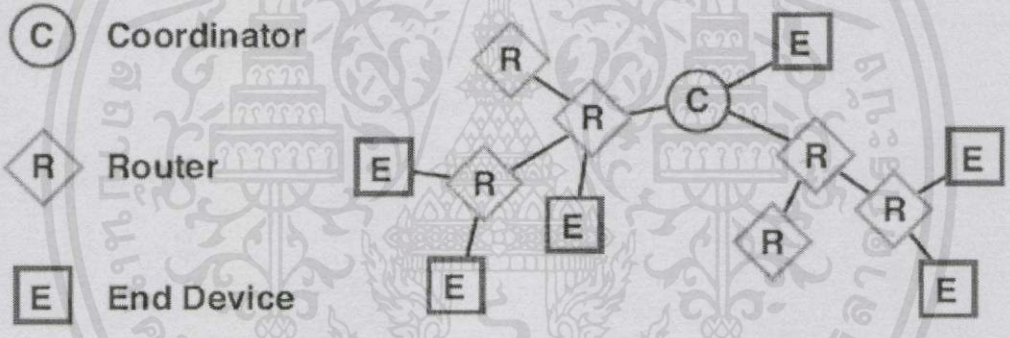
ZigBee จะสามารถสร้างเป็นเครือข่ายได้เพราะอิงมาตรฐานตาม IEEE 802.15.4 และมีการจัดการในแบบของ Zigbee ใน Layer ถัดไป ทั้งนี้ IEEE 802.15.4 แบ่งชนิดอุปกรณ์ในเครือข่ายออกเป็น 2 ประเภท คือ FFD (Full Function Device) ซึ่งหมายถึงอุปกรณ์ที่สามารถทำงานได้ทุก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อย่างในเครือข่าย และ RFD (Reduce Function Device) ซึ่งหมายถึงอุปกรณ์ที่ถูกลดความสามารถการทำงานในเครือข่าย

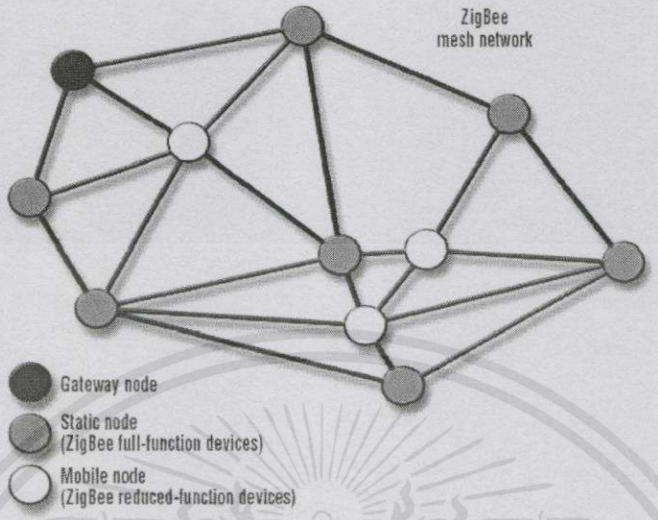
ZigBee ได้แบ่งตามลักษณะการทำงาน 3 แบบ คือ

1) Coordinator มีหน้าที่สร้างการสื่อสาร เชื่อมโยงเครือข่าย ระหว่าง End Device กับ Router หรือ Coordinator กับ Coordinator ด้วยกัน หรือ Coordinator กับ Router กำหนด address ให้กับ device ที่อยู่ในวงเครือข่าย ไม่ให้ซ้ำกัน ดูแลจัดการเรื่องการ Routing เส้นทาง ซึ่งเทียบได้กับ FFD
2) End Device เป็นอุปกรณ์ปลายทางสุด ซึ่งจะใช้รับสัญญาณจาก Sensor ที่ปลายทางโดยที่ใช้พลังงานต่ำในการทำงาน เทียบได้กับ RFD หรือ FFD บางกรณี ขึ้นอยู่กับ sensor ที่ใช้
3) Router มีหน้าที่ รับส่งข้อมูล ในเส้นทางต่าง ๆ ของเครือข่าย ซึ่งเทียบได้กับ FFD [4]



รูปที่ 2.6 แสดง ZigBee Nodes

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.7 แสดง ZigBee Mesh Network

2.3.2 ZigBee Light Link

ZigBee Light Link เป็นข้อกำหนดมาตรฐานและเป็นโปรโทคอลของระบบแสงสว่าง (หลอดไฟ) สำหรับการควบคุมที่ง่าย และควบคุมได้หลากหลายชนิดของหลอดไฟ ทำให้ผู้ใช้สามารถควบคุมระบบไฟแบบไร้สายได้อย่างง่ายดายและครบถ้วนทุกๆชนิดๆ การควบคุมแบบไร้สายนั้นจะทำให้ผู้ใช้สามารถควบคุมระบบแสงสว่างได้อย่างรวดเร็ว สะดวกสบาย สามารถสั่งเปิด-เปิดได้อย่างรวดเร็ว ส่งผลให้เป็นการลดการใช้พลังงานอย่างมีประสิทธิภาพ

อุปกรณ์ที่ใช้มาตรฐาน ZigBee Light Link นี้ จะใช้งานง่ายเหมือนกับเราใช้งาน เปิด-ปิดหลอดไฟทั่วไป มาตรฐานนี้ไม่จำเป็นต้องมีระบบหรืออุปกรณ์อะไรพิเศษมาเกี่ยวข้องกับระบบไฟเลย ทำให้ง่ายต่อการใช้งานในชีวิตประจำวัน โดยตอนนี้ได้มีบริษัทที่ผลิตระบบที่ควบคุมผ่านมาตรฐาน ZigBee Light Link ออกมาแล้ว คือ GE, Greenwave, OSRAM Sylvania และ Philips.[5]

Application End Point	Network Address	Destination End Point	ZCL Cluster ID	Date Len	Address Mode	ZCL Frame Control Field	ZCL Trans Seq No.	ZCL Command ID	ZCL Command Payload
1 Byte	2 Byte	1 Byte	2 Bytes	1 Byte	1 Byte	1 Byte	1 Byte	1 Byte	x Byte

รูปที่ 2.8 Message format ของ Light Link

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3.2.1 หลักการทำงานของ ZigBee Light Link

ZigBee Light Link คือระบบเครือข่ายไร้สายที่ประกอบไปด้วย node 2 node คือ Controller nodes และ Light (controlled) nodes โดย Controller nodes ประกอบไปด้วย สวิตช์ไฟ Occupancy sensors Smart phone หรือ Computing devices เช่น PC หรือ tablet และ Light nodes คือ หลอดไฟต่างๆ ไม่ว่าจะเป็นแบบหลายสี หรือสีเดียว

ZLL Device	Device ID	ZLL Commissioning Cluster		
		Server	Server/Client	Client
On/Off Light	0x0000	✓	✓	x
On/Off Plug-in Unit	0x0010	✓	✓	x
Dimmable Light	0x0100	✓	✓	x
Dimmable Plug-in Unit	0x0110	✓	✓	x
Colour Light	0x0200	✓	✓	x
Extended Colour Light	0x0210	✓	✓	x
Colour Temperature Light	0x0220	✓	✓	x

รูปที่ 2.9 คำสั่งของ Light devices

ZLL Device	Device ID	ZLL Commissioning Cluster		
		Server	Server/Client	Client
Colour Controller	0x0800	✓	✓	✓
Colour Scene Controller	0x0810	✓	✓	✓
Non-Colour Controller	0x0820	✓	✓	✓
Non-Colour Scene Controller	0x0830	✓	✓	✓
Control Bridge	0x0840	✓	✓	✓
On/Off Sensor	0x0850	✓	✓	✓

รูปที่ 2.10 คำสั่งของ Controller devices

ZigBee light link จะใช้ network topology แบบ Mesh เพื่อจุดประสงค์ในการสร้างความยืดหยุ่นในการ Routing และ node ทุกๆ node ควรจะเป็น Zigbee Routers

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3.2.2 การเริ่มการทำงานของ ZigBee Light Link

ZigBee Light Link จะทำการแลกเปลี่ยน command ระหว่าง nodes เพื่อเริ่มติดต่อสื่อสารกัน แบ่งออกเป็น 3 ส่วนดังนี้

1. Discovery – Initiator node จะทำการ scan หา Zigbee Light Link node ในบริเวณใกล้เคียงขึ้นอยู่กับความแรงของสัญญาณ โดยผลลัพธ์ก็คือ list ของ node ที่ประกอบไปด้วย device type และ network capabilities
2. Transfer of network setting – Initiator จะ request เพื่อรอรับ network setting ของ node ที่ต้องการ
3. Request network formation or join – Initiator จะ request ให้ node นั้นสร้างnetwork ใหม่ หรือ join network ที่มีอยู่แล้ว

2.3.2.3 ZigBee Light Link Network Address

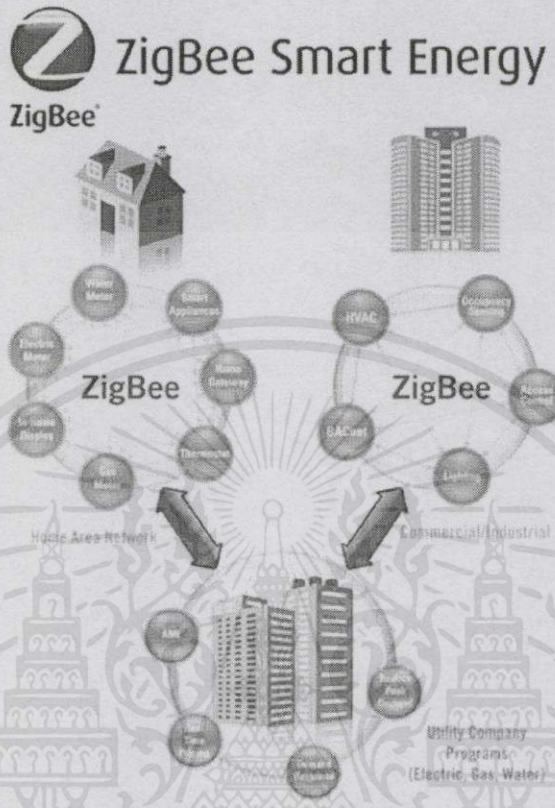
ZigBee Light Link network จะใช้ network address 16 บิท ในการตรวจสอบ node ทั้งหมด โดยจะมีเพียง ZigBee Light Link ที่เป็น Controller device ที่จะสามารถแจก network address ได้ ถ้า Controller device ตัวใหม่ถูกเพิ่มเข้ามาใน network มันจะแบ่งปริมาณ address ที่สามารถแจกจ่ายให้กับ node ที่พึ่งเข้ามาใน network [6]

2.3.3 ZigBee Smart Energy

Zigbee Smart Energy เป็นมาตรฐานของผลิตภัณฑ์ที่ตรวจจับ ควบคุม แจ้งเตือน ระบบพลังงานและน้ำอย่างอัตโนมัติ โดยจะช่วยให้บ้านที่เราอยู่นั้นมีความเป็นมิตรกับสิ่งแวดล้อมมากขึ้น โดยการให้ข้อมูลที่จำเป็นเพื่อการลดการใช้พลังงาน และทำให้ช่วยประหยัดเงินได้อีกด้วย

Zigbee Energy Profile 2 เป็นมาตรฐานของการควบคุมด้วย Internet Protocol ทั้งแบบมีสายและไร้สาย สำหรับการจัดการพลังงาน และ Home Area Networks (HANs) โดย Zigbee Energy Profile 2 นั้น คือการพัฒนาต่อจาก Zigbee Smart Energy และได้เพิ่มความสามารถในการควบคุมรถยนต์ Plug-in Hybrid Vehicles (PHEVs) เช่นการชาร์จไฟ ส่วน Home Area Networks ก็ได้พัฒนาให้รองรับกับ การจัดการพลังงานที่ต้องใช้งานพร้อมกันจำนวนมาก เช่นใน อาคารใหญ่ๆ หรือ อพาร์ทเม้น โดยจะอยู่บนมาตรฐาน IETF IP compliant protocols เช่น ZigBee IP Specification [7]

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.11 Zigbee Smart Energy Profile

2.4 เทคโนโลยี Wireless LAN

Wi-Fi (ย่อมาจาก wireless fidelity) ก็คือองค์กรหนึ่ง ที่ทำหน้าที่ทดสอบผลิตภัณฑ์ Wireless Lan หรือระบบ Network แบบไร้สายภายใต้เทคโนโลยีการสื่อสาร ภายใต้มาตรฐาน IEEE 802.11 ว่าอุปกรณ์ทุกตัวซึ่งต่างยี่ห้อกันนั้นสามารถติดต่อสื่อสารกันได้โดยไม่ มีปัญหา หากว่าอุปกรณ์ตัวนั้นผ่านตามมาตรฐานจะมีสัญลักษณ์ Wi-Fi certified ซึ่งเป็นอันรู้กันว่าอุปกรณ์ สามารถติดต่อสื่อสารกับอุปกรณ์ตัวอื่น ที่มีตรา WiFi certified นี้ได้เช่นกัน

Wireless คือลักษณะของการใช้งานอุปกรณ์ด้านสื่อสารโทรคมนาคม แปลตรงตัวว่าไร้สาย ฉะนั้น อุปกรณ์อะไรก็ตามที่ติดต่อสื่อสารกันโดยไม่ใช้สายสัญญาณถือ ว่าอุปกรณ์นันเป็น Wireless เหมือนกัน [8]

ระบบ เครือข่ายไร้สาย (Wireless LAN : WLAN) หมายถึงเทคโนโลยีที่ช่วยให้การติดต่อสื่อสารระหว่างเครื่องคอมพิวเตอร์ 2 เครื่อง หรือกลุ่มของเครื่องคอมพิวเตอร์สามารถสื่อสารกัน

เอกสารนี้เป็นเอกสาร... ไม่ว่ากรณี... ...ด้านการค้า

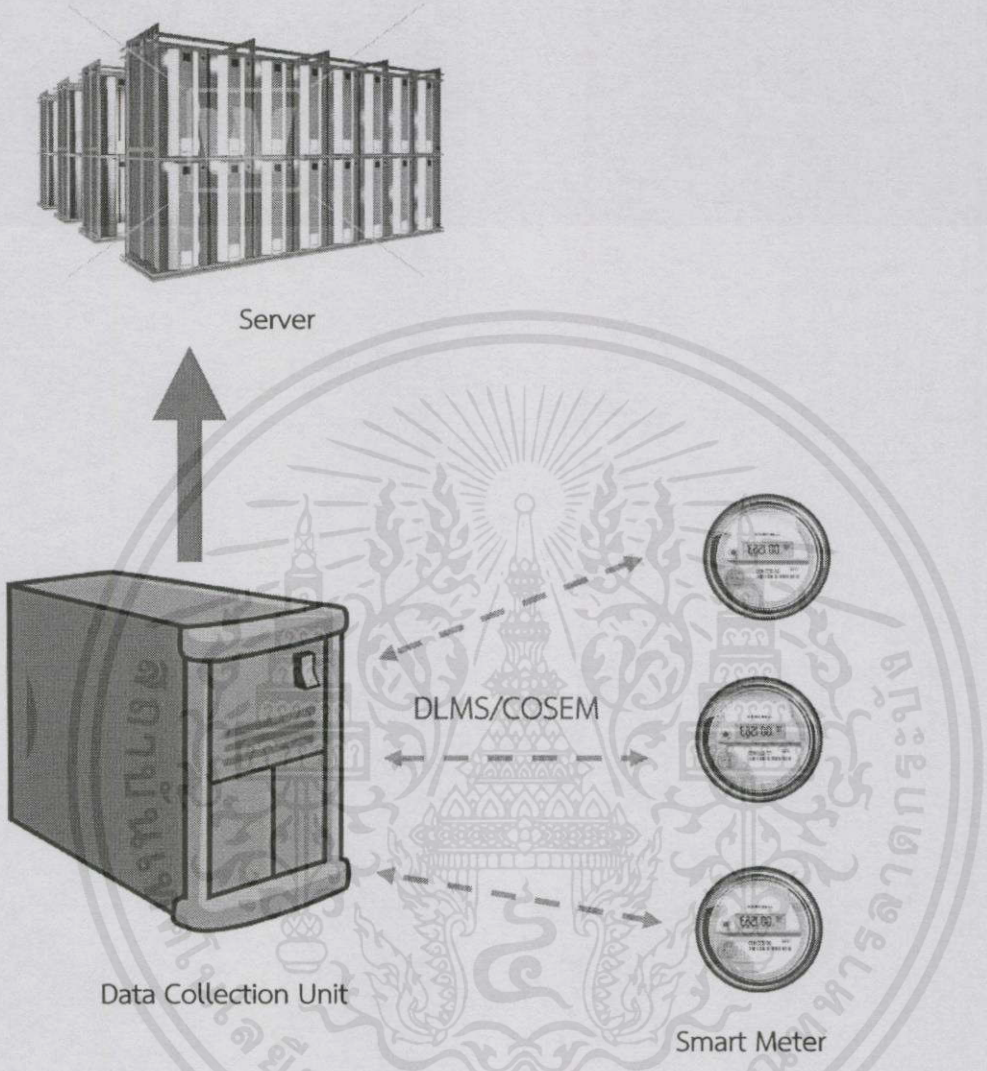
ได้ รวมถึงการติดต่อสื่อสารระหว่างเครื่องคอมพิวเตอร์กับอุปกรณ์เครือข่าย คอมพิวเตอร์ด้วยเช่นกัน โดยปราศจากการใช้สายสัญญาณในการเชื่อมต่อ แต่จะใช้คลื่นวิทยุเป็นช่องทางการสื่อสารแทน การรับส่งข้อมูลระหว่างกันจะผ่านอากาศ ทำให้ไม่ต้องเดินสายสัญญาณ และติดตั้งใช้งานได้สะดวกขึ้น ระบบเครือข่ายไร้สายใช้แม่เหล็กไฟฟ้าผ่าน อากาศ เพื่อรับส่งข้อมูลข่าวสารระหว่างเครื่องคอมพิวเตอร์ และระหว่างเครื่องคอมพิวเตอร์กับอุปกรณ์เครือข่าย โดยคลื่นแม่เหล็กไฟฟ้านี้อาจเป็นคลื่นวิทยุ (Radio) หรืออินฟาเรด (Infrared) ก็ได้

การสื่อสารผ่านเครือข่ายไร้สายมีมาตรฐาน IEEE802.11 เป็นมาตรฐานกำหนดรูปแบบการสื่อสาร ซึ่งมาตรฐานแต่ละตัวจะบอกถึงความเร็วและคลื่นความถี่สัญญาณที่แตกต่างกันใน การสื่อสารข้อมูล เช่น 802.11b และ 802.11g ที่ความเร็ว 11 Mbps และ 54 Mbps ตามลำดับ และขอบเขตของสัญญาณคลอบคลุมพื้นที่ประมาณ 100 เมตรในพื้นที่โล่งและประมาณ 30 เมตรในอาคาร ซึ่งระยะทางของสัญญาณมีผลกระทบจากสิ่งรอบข้างหลายๆ อย่าง เช่น โทรศัพท์มือถือ ความหนาของกำแพง เครื่องใช้ไฟฟ้า อุปกรณ์อิเล็กทรอนิคส์ต่างๆ รวมถึงร่างกายมนุษย์ด้วยเช่นกัน สิ่งเหล่านี้มีผลกระทบต่อการใช้งานเครือข่ายไร้สายทั้งสิ้น

2.5 DLMS/COSEM

DLMS (Device Language Message Specification) และ COSEM (Companion Specification for Energy Metering) เป็นโปรโทคอลในชั้น Application layer ที่พัฒนาขึ้นสำหรับให้เป็นมาตรฐานในการส่งข้อมูลระหว่าง สมาร์ทมิเตอร์ และ หน่วยเก็บข้อมูล (Data collection unit) โดยในการส่งจะมีการ encode ข้อมูลที่ฝั่งส่ง และ ฝั่งรับต้องทำการ decode เพื่อที่จะตีความ message ที่ได้รับ [9]

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.12 Smart Meters เชื่อมต่อกับ Data collector ด้วยโปรโตคอล DLMS/COSEM

สมาร์ทมิเตอร์จะส่งข้อมูลให้กับ Data collector ผ่านโปรโทคอล DLMS/COSEM เพื่อเก็บข้อมูลปริมาณการใช้ไฟฟ้า และ ค่าหน่วยไฟฟ้าต่างๆ จากนั้น Data collector จะส่งข้อมูลต่อไปยังเซิฟเวอร์กลาง เพื่อที่จะนำไปใช้ในการวิเคราะห์ และ ประมวลผลต่อไป ไม่ว่าจะเป็นการคำนวนค่าไฟฟ้า การพิจารณาการจ่ายไฟ การพิจารณาการตัดไฟ เมื่อผู้ใช้บริการไม่จ่ายค่าไฟฟ้า หรือการแจ้งเตือนเมื่อเกิดไฟดับ เป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

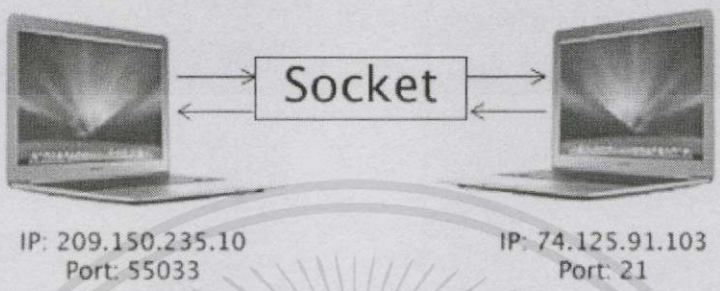
	Data unit	Layer	Function	DLMS/COSEM
		7. Application	Network process to application	Application like GXDLMSDirector
Host layers	Data	6. Presentation	Data representation, encryption and decryption, convert machine dependent data to machine independent data	COSEM
		5. Session	Interhost communication, managing sessions between applications	DLMS
	Segments	4. Transport	End-to-end connections, reliability and flow control	DLMS
	Packet/Datagram	3. Network	Path determination and logical addressing	
Media layers	Frame	2. Data link	Physical addressing	HDLC or IEC 62056-47
	Bit	1. Physical	Media, signal and binary transmission	Gurux Media like Serial, TCP/IP

รูปที่ 2.13 DLMS/COSEM เมื่อเปรียบเทียบกับ OSI model

Object ของสมาร์ทมิเตอร์จะถูกอ่านเข้ามาผ่าน interface ตัวอย่างเช่น clock register และอื่นๆ โดยแต่ละ Object จะมี logical name ติดตัวเสมอ โดย logical name คือ attribute แรกใน object ตัวอย่างเช่น 0.0.1.0.0.266 คือ Clock's Logical Name หรือ OBIS Code โดยถ้าผู้ผลิตใช้ logical name เหมือนกันทั้งหมด Data collector จะสามารถอ่านค่าจาก สมาร์ทมิเตอร์หลายๆอันด้วย application เดียว โดยการเปรียบเทียบจาก logical name

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.6 การเขียนโปรแกรมเชื่อมต่อกับ Socket



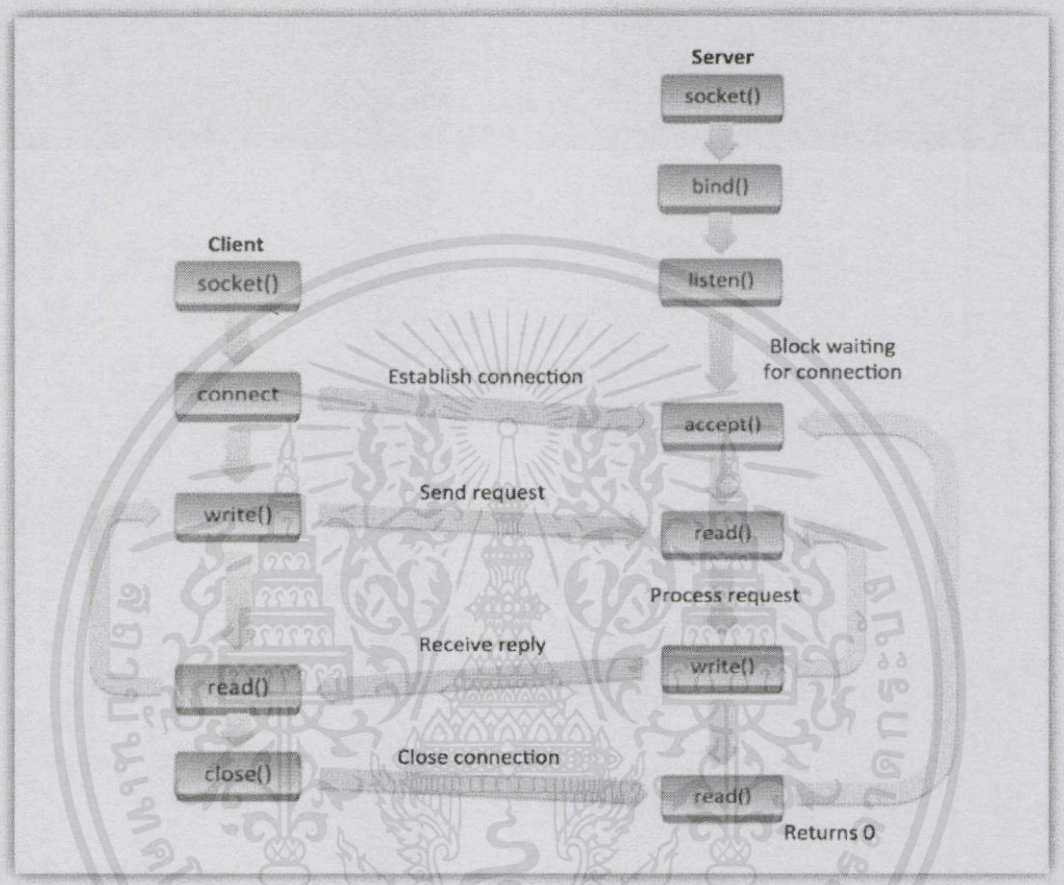
รูปที่ 2.14 Socket

โดยทั่วไปในการติดต่อสื่อสารที่นิยมใช้กันในปัจจุบันจะทำบนพื้นฐานของโปรโทคอล TCP/IP โดยผู้พัฒนาโปรโทคอล TCP/IP ไม่ต้องการให้ TCP/IP ใช้งานได้กับแพลตฟอร์มใดแพลตฟอร์มหนึ่งเป็นการเฉพาะ หรือใช้งานได้เพียงระบบปฏิบัติการระบบใดระบบหนึ่ง โดยได้เปิดให้นักพัฒนาสามารถกำหนดมาตรฐานในการเชื่อมต่อได้เอง ต่อมาในปี 1980 Advanced Research Projects Agency (ARPA) ได้ให้ทุนกับ University of California at Berkeley เพื่อพัฒนา TCP/IP Software และผลที่ได้รับคือ Berkeley Sockets Interface หรือเรียกสั้นๆว่า Sockets Interface

Socket นั้นคือหนึ่งช่องทางในการสื่อสารเชื่อมต่อกันผ่านเครือข่ายแบบ Client/Server ซึ่งหมายถึงการเชื่อมต่อสื่อสารกันระหว่างจุดสองจุด หรือ อาจจะมากกว่าสองจุดก็สามารถเชื่อมต่อได้ ตัวอย่างที่ใช้ Socket ได้แก่ Web browser Application ที่ใช้ส่งข้อความหากัน หรือ การแลกเปลี่ยนไฟล์แบบ Peer to Peer ก็ล้วนใช้ Socket ทั้งสิ้น Socket นั้นมีวิธีในการส่งข้อมูลแบบ bidirectional ซึ่งหมายถึงแต่ละฝั่งของการเชื่อมต่อ (Client หรือ Server) ต่างมีความสามารถในการทั้งส่งและรับข้อมูลในตัวของมันเอง โดยมีชนิดของ Socket ดังนี้

1. Datagram sockets เป็น socket แบบ Connectionless ซึ่งใช้ User Datagram Protocol (UDP)
2. Stream sockets เป็น socket แบบ Connection – oriented ซึ่งใช้ Transmission Control Protocol (TCP) หรือ Stream Control Transmission Protocol (SCTP)
3. Raw sockets หรือ Raw IP sockets เป็น socket ที่สามารถพบได้ตาม routers หรืออุปกรณ์เครือข่ายต่างๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.15 แสดง Socket workflow

เริ่มต้นฝั่ง Server จะทำการสร้าง Socket ของตนขึ้นมาซึ่งประกอบด้วย IP และ port ที่จะใช้ในการรับส่งข้อมูลจากนั้น ฝั่ง server จะเข้าสู้ listening mode และจะรอจนกว่าจะมีการเชื่อมต่อจาก client เข้ามาที่ socket นี้ เมื่อมีการเชื่อมต่อจากฝั่ง client เข้ามาแล้วนั้น ฝั่ง server จะทำการตอบรับ และอ่านคำสั่งที่ ฝั่ง client ส่งเข้ามา และจึงดำเนินการตามกระบวนการตามคำสั่งที่ได้รับเข้ามา จากนั้นจะส่งผลลัพธ์ของคำสั่งไปให้ client การทำงานจะวนไปเรื่อยๆจนยกเลิกการเชื่อมต่อ [10]

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.7 Hue System

รูปที่ 2.13 ภาพรวมของ Hue System

ประกอบไปด้วย 3 ส่วน คือ

1. Apps – มีหน้าที่ในการควบคุมหลอดไฟฟ้าโดยการส่งคำสั่งต่างๆที่ต้องการควบคุมหลอดไฟผ่านไปทาง Bridge เพื่อให้ Bridge ส่งคำสั่งนี้ไปยังหลอดไฟเพื่อควบคุมอีกที โดยสามารถสั่งให้หลอดไฟ เปิด ปิด เปลี่ยนสี ปรับความเข้มแสง และ ปรับความสว่างได้ เป็นต้น
2. Bridge – สำหรับทำการเชื่อมต่อแอพพลิเคชันกับหลอดไฟ เนื่องจาก แอพพลิเคชันไม่สามารถที่จะควบคุมหลอดไฟโดยตรง จึงต้องทำการ เชื่อมต่อผ่าน Bridge ด้วย Wifi และ Bridge จะทำการเชื่อมต่อหลอดไฟผ่าน Zigbee ด้วย โปรโตคอล ZigBee light link
3. Lights – หลอดไฟ ทำหน้าที่รับคำสั่งจากตัวที่ส่งมาจากแอพพลิเคชัน ที่ผ่านทางตัว Bridge โดยหลอดไฟฟ้าทำงานภายใต้มาตรฐาน Zigbee Light Link ซึ่งเกี่ยวกับการความสามารถในการทำงานต่างๆเช่น สามารถเปลี่ยนสี ความสว่าง ความเข้มแสง เป็นต้น

ขั้นตอนในการเชื่อมต่อระหว่างตัวแอพพลิเคชัน กับ ตัว Bridge

การทำงานจะเริ่มจากหน้า PHMainViewController โดยฟังก์ชันที่เริ่มทำงานตัวแรก คือ enableLocalheartbeat ซึ่งถ้าหากมี Bridge ต่อด้วยอยู่จะทำการ readCache มาจาก Bridge ทุกๆ 30 วินาที แต่ถ้าหาก ไม่มี Bridge ต่ออยู่จะต้องทำการค้นหา Bridge โดยใช้ searchForBridgeLocal ซึ่งฟังก์ชันนี้จะทำงานโดยใช้ Upnp โดยเมื่อพบ Bridge จะเปลี่ยนการทำงานไปยังหน้า PHBridgeSelectionViewController เพื่อให้ผู้ใช้งานทำการเลือก Bridge ที่ต้องการเพื่อใช้งาน เมื่อผู้ใช้ทำการเลือก Bridge IP Adress และ MAC Address ของ Bridge ตัวนั้นจะถูกใช้โดยฟังก์ชัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น ไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

setBridgeToUseWithIpAddress: macAddress: เพื่อเป็นการกำหนดว่า Bridge ที่เราใช้จะมี IP Address และ mac Address ตามนี้จากนั้นจึงจะเริ่มทำฟังก์ชัน doAuthentication เพื่อเข้าใช้งานตัว Bridge โดยการทำ authentication นี้แอพพลิเคชันจะแสดงหน้า pushLinkViewController ขึ้นมาเพื่อให้ผู้ใช้งานทำการกดปุ่มที่ตัว Bridge เพื่อเป็นการยืนยันการเข้าใช้งาน ถ้าการ authentication สำเร็จ ฟังก์ชัน pushLinkSuccess จะทำงาน เพื่อเริ่มการทำ enableLocalheartbeart ในรอบต่อไป เป็นอันเสร็จสิ้นกระบวนการในการเชื่อมต่อกับ Bridge

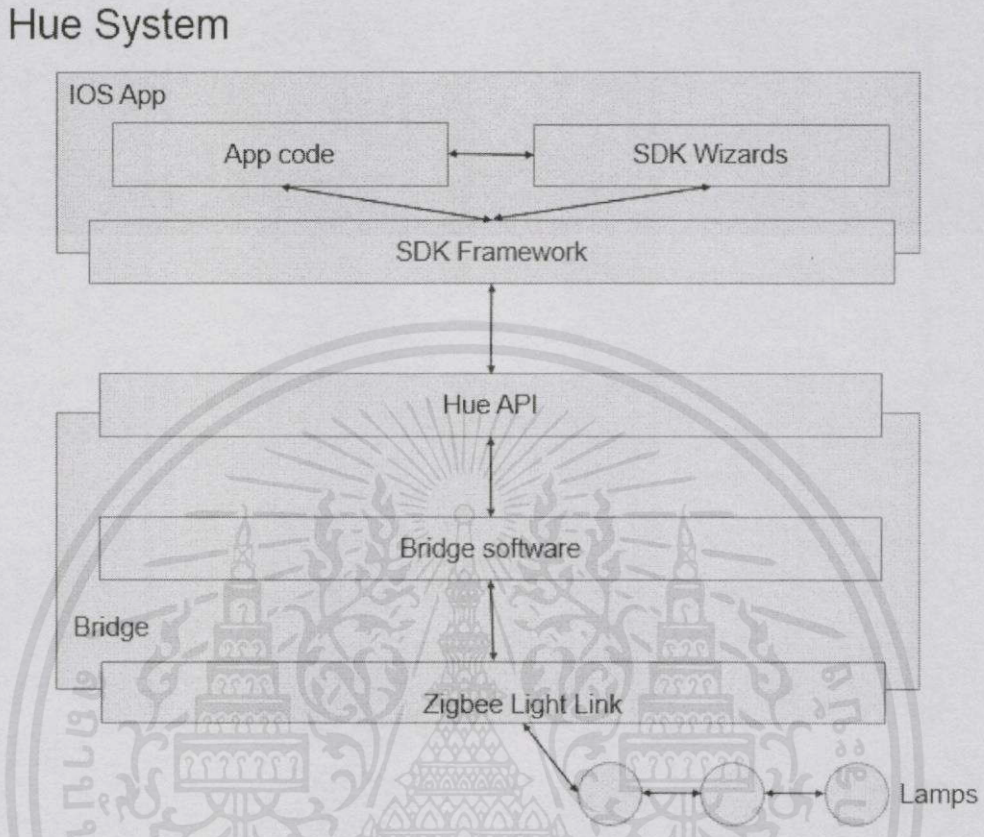
ขั้นตอนในการค้นหาหลอดไฟของ Bridge

จะเริ่มจากฟังก์ชัน startSearch ซึ่งอยู่ใน PHFindLightResultViewController โดยจะใช้ฟังก์ชัน searchForNewLights ซึ่งอยู่ในโปรโทคอล PHBridgeSendAPI ทำการค้นหาหลอดไฟทุก 1 วินาที แต่จะ timeOut เมื่อครบ 60 วินาที โดยถ้าพบหลอดไฟแล้วจะทำฟังก์ชัน updataLightsFound เพื่อทำการเอาข้อมูลของหลอดไฟมาใส่ใน cache ของ Bridge โดยใช้ฟังก์ชัน getNewFoundLight ซึ่งเป็นโปรโทคอล PHBridgeSendAPI เช่นเดียวกัน

ขั้นตอนการควบคุมหลอดไฟ

จะเริ่มที่หน้า PHSelectionViewController ซึ่งเป็นหน้าที่เอาไว้เลือกหลอดไฟที่เราต้องการจะควบคุม โดยหลอดไฟที่เราสามารถเลือกได้จะได้มาจากฟังก์ชัน updateLights ซึ่งฟังก์ชันนี้จะไปอ่าน cache มาจากตัว Bridge อีกที แล้วจะนำข้อมูลของหลอดไฟมาแสดงในรูปแบบของตารางเพื่อให้ผู้ใช้งานเลือกหลอดไฟเพื่อใช้งาน เมื่อผู้ใช้งานเลือกหลอดไฟดวงไหนก็จะเข้าสู่การควบคุมของหลอดไฟดวงนั้นในหน้า PHLightControlViewController ต่อไปโดยการควบคุมจะสามารถควบคุมได้หลายค่าเช่น ควบคุมความสว่าง ควบคุมความอิ่มตัวของสี ควบคุมการเปิด ปิดหลอดไฟ เป็นต้น โดยเมื่อผู้ใช้ทำการตั้งค่าต่างๆเสร็จค่าต่างๆเหล่านี้จะถูกสร้างขึ้นมาเป็น lightState หากผู้ใช้งานต้องการให้หลอดไฟทำงานตามที่ผู้ใช้งานตั้งค่าไว้ ก็จะต้องส่งค่า lightState นี้ไปยัง Bridge โดยใช้ฟังก์ชัน updateLightStateForId: withLightState: เพื่อให้ Bridge ไปควบคุมหลอดไฟให้ทำงานตามที่ผู้ใช้ต้องการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.14 โครงสร้างของ Hue System ในส่วนของ Software

จากภาพจะเป็น IOS Application ประกอบด้วยส่วนของ App Code ซึ่งเป็นส่วนของการควบคุมการทำงานต่างๆของหลอดไฟ เช่น การเปิด ปิดไฟ การปรับแสงสว่าง เป็นต้น ในส่วนของ SDK Wizards เป็นส่วนที่ช่วยในการเชื่อมต่อระหว่าง แอพพลิเคชัน กับ Bridge และใช้ในการค้นหาหลอดไฟ เพื่อเอามาใช้งานร่วมกับตัว Bridge เช่น ฟังก์ชัน searchForBridgeLocal ที่เอาไว้ใช้ในการค้นหา Bridge หรือ ฟังก์ชัน updateLight ที่เอาไว้ใช้ในการอัพเดทข้อมูลของหลอดไฟลงใน cache ของ Bridge เป็นต้น เมื่อส่วน AppCode และส่วน SDK Wizards มารวมตัวกัน จะกลายเป็น SDK Framework ซึ่งจะทำการส่งไปให้ Bridge API ต่อไป ซึ่ง Bridge API นี้จะมี Software ที่ใช้ในการส่งค่าไปยังหลอดไฟเพื่อทำการควบคุมโดยใช้โปรโทคอล Light Link

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3
การออกแบบและพัฒนา

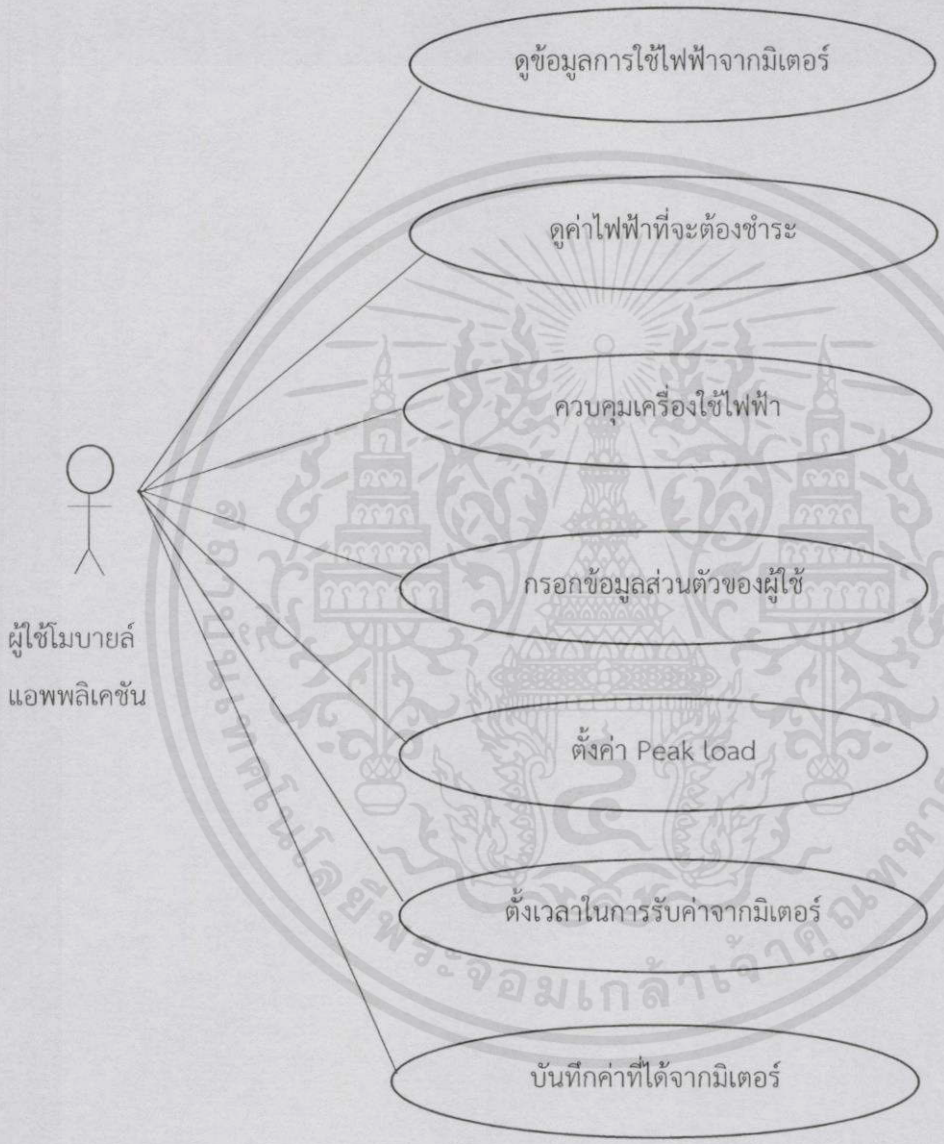
3.1 ความต้องการของระบบ

1. แสดงผลค่าการใช้งานไฟฟ้าต่างๆในช่วงเวลานั้นๆ โดยสามารถเลือกแสดงแบบ Scroll mode เพื่อให้ผู้ใช้งานสามารถเลือกค่าการใช้งานไฟฟ้าที่ต้องการดูได้ หรือเลือกแบบ default mode ได้ในกรณีที่ผู้ใช้งานต้องการให้มีการแสดงค่าต่างๆวนไปเรื่อยๆ
2. มีการเก็บข้อมูลการใช้ไฟฟ้าสำหรับการแสดงข้อมูลย้อนหลังได้
3. เลือกแสดงข้อมูลการใช้ไฟฟ้าย้อนหลังจากปฏิทินได้
4. สามารถแสดงยูนิตไฟฟ้าที่ใช้ภายในแต่ละวันได้โดยการเลือกวันที่บนปฏิทินที่เราต้องการให้แสดง
5. สามารถคำนวนค่าไฟฟ้าจากปริมาณการใช้ไฟฟ้าได้
6. สามารถแสดงค่าไฟฟ้าที่ได้จากการคำนวน และ สามารถเลือกให้แสดงค่าไฟฟ้าย้อนหลังได้
7. สามารถควบคุมหลอดไฟภายในบ้านผ่านเครือข่าย Wifi ได้
8. มีการตั้งค่า Peak load เพื่อควบคุมการใช้ปริมาณไฟฟ้าได้
9. เมื่อมีการใช้ปริมาณไฟฟ้าเกินค่า Peak load ที่ตั้งไว้ จะสามารถลด หรือระงับการใช้งานไฟฟ้าได้โดยอัตโนมัติ
10. สามารถตั้งค่าข้อมูลส่วนตัว และ ข้อมูลการใช้ไฟฟ้าได้
11. ข้อมูลส่วนตัว และ ข้อมูลการใช้ไฟฟ้าสามารถบันทึกลงในตัวแอพพลิเคชันได้
12. ข้อมูลส่วนตัว และ ข้อมูลการใช้ไฟฟ้าต้องถูกโหลดขึ้นมาแสดงทุกครั้งที่ผู้ใช้งานเปิดแอพพลิเคชัน
13. ผู้ใช้งานสามารถปรับเวลาในการรับข้อมูลการใช้งานไฟฟ้าที่ถูกส่งมาจากเครื่องคอมพิวเตอร์ได้
14. ตัวแอพพลิเคชันสามารถติดต่อสื่อสารกับเครื่องคอมพิวเตอร์ผ่านทาง Socket ได้
15. มีฐานข้อมูลที่เอาไว้ใช้ในการเก็บข้อมูลการใช้ไฟฟ้า
16. มีหน้าตาของแอพพลิเคชันที่เรียบง่ายใช้งานง่าย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2 การออกแบบระบบ

3.2.1 Use case diagram



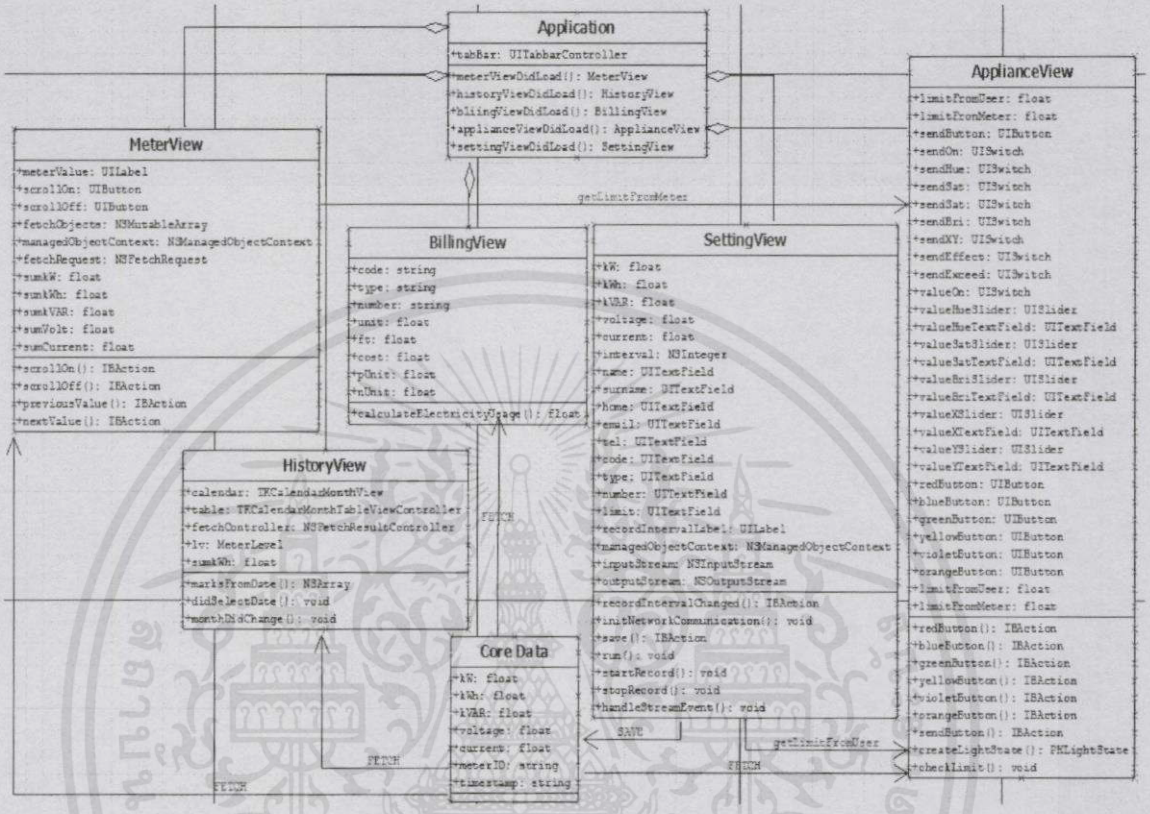
รูปที่ 3.1 แสดง Use case diagram ของระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. ดูข้อมูลการใช้ไฟฟ้าจากมิเตอร์ - ผู้ใช้งานสามารถดูข้อมูลข้อมูลการใช้ไฟฟ้าต่างๆที่มาจากสมาร์ทมิเตอร์ได้ เช่น ค่า kW ค่า kVAR ค่า kWh ค่า Voltage และ ค่า Current โดยสามารถดูข้อมูลต่างๆย้อนหลังได้ตามที่ต้องการ
2. ดูค่าไฟฟ้าที่จะต้องชำระ - ผู้ใช้สามารถดูค่าไฟฟ้าที่ต้องจ่าย รวมถึงรายละเอียดการใช้งานต่างๆในแต่ละเดือนได้ เช่น ยูนิตที่ใช้ในเดือนนี้ ยูนิตที่ใช้เดือนก่อน เป็นต้น โดยค่าไฟฟ้าคำนวณโดยใช้ค่าที่รับเข้ามาจากมิเตอร์ ผู้ใช้สามารถเลือกเดือน เลือกปีที่ต้องการดูค่าไฟฟ้าได้
3. ควบคุมเครื่องไฟฟ้า - ผู้ใช้สามารถควบคุมหลอดไฟได้ เช่น เปิด ปิด หลอดไฟ ปรับความเข้มสี ปรับความสว่าง และ ปรับสี เป็นต้น
4. ตั้งค่า Peak load - ผู้ใช้สามารถตั้งขีดจำกัดการใช้ไฟฟ้า ที่สูงที่สุดที่สามารถรับไหวได้ ถ้ามีการใช้ไฟฟ้าเกินค่า Peak Load นี้เมื่อไหร่ หลอดไฟฟ้าจะถูกสั่งปิด หรือ หรี่ในทันที
5. กรอกข้อมูลส่วนตัวของผู้ใช้ - ผู้ใช้สามารถกรอกข้อมูลต่างๆได้ เช่น ประวัตส่วนตัว เช่น ชื่อ ที่อยู่ เบอร์โทรศัพท์ เป็นต้น หรือ ข้อมูลเกี่ยวกับการใช้ไฟฟ้าต่างๆ เช่น รหัสการใช้ไฟฟ้า ประเภท เป็นต้น
6. ผู้ใช้สามารถปรับเวลาที่ใช้รับค่ามาจากมิเตอร์ได้ - ผู้ใช้สามารถเลือกช่วงเวลาที่ต้องการรับค่าที่มาจากมิเตอร์ได้ โดยสามารถเลือกให้เร็ว หรือ ช้า ก็ได้แล้วแต่ผู้ใช้งาน
7. บันทึกค่าที่ได้จากมิเตอร์ - แอพพลิเคชันสามารถบันทึกค่าที่อ่านได้จากมิเตอร์เก็บลงในฐานข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.2 Class Diagram



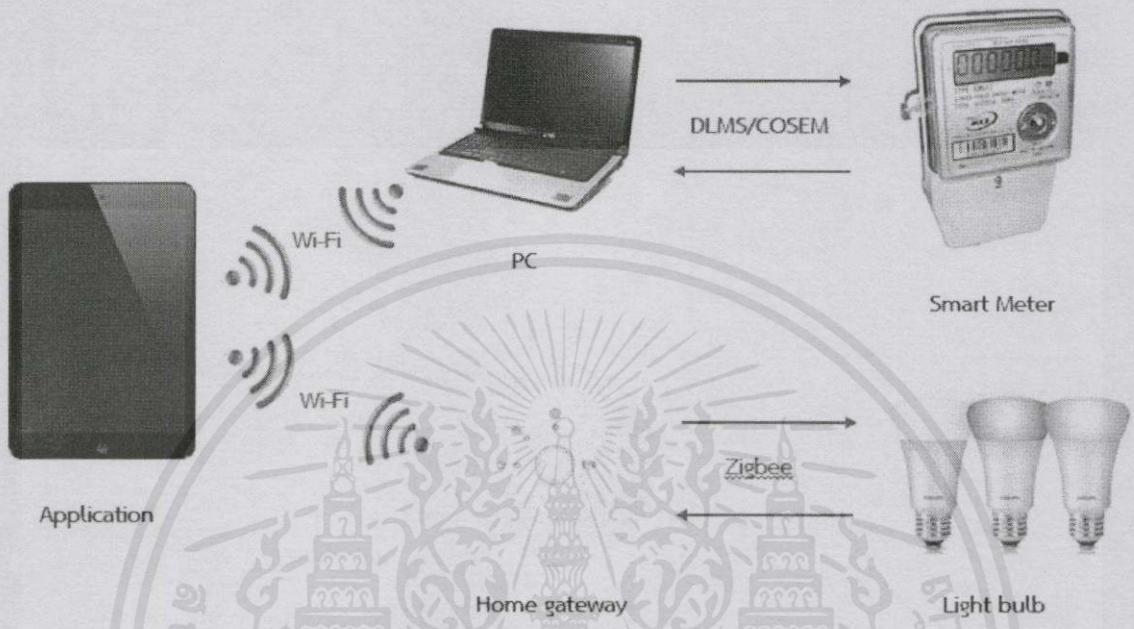
รูปที่ 3.2 Class Diagram ของระบบ

Class Diagram ประกอบด้วย ตัว Application หลัก ซึ่งอยู่ในรูปแบบของ UITabbarController โดยจะประกอบไปด้วยคลาสอื่นๆดังนี้

- MeterView เป็นส่วนที่เกี่ยวกับการแสดงข้อมูลต่างๆที่ได้รับมาจากการ FETCH จาก Core Data
- HistoryView เป็นส่วนที่เกี่ยวกับการแสดงข้อมูลแบบย้อนหลังในรูปแบบของตาราง และมีปฏิทิน
- BillingView เป็นส่วนที่เกี่ยวกับการคำนวณค่าใช้ไฟฟ้าและแสดงรายละเอียดของการใช้ไฟฟ้า
- ApplianceView เป็นส่วนที่เกี่ยวกับการควบคุมหลอดไฟ
- SettingView เป็นส่วนที่เกี่ยวกับการตั้งค่าการใช้งานต่างๆ และ ทำหน้าที่ในการรับและบันทึกค่าที่รับมา
- Core Data เป็นส่วนที่ทำหน้าที่เป็นฐานข้อมูลของตัวแอพพลิเคชัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.3 ภาพรวมของระบบ



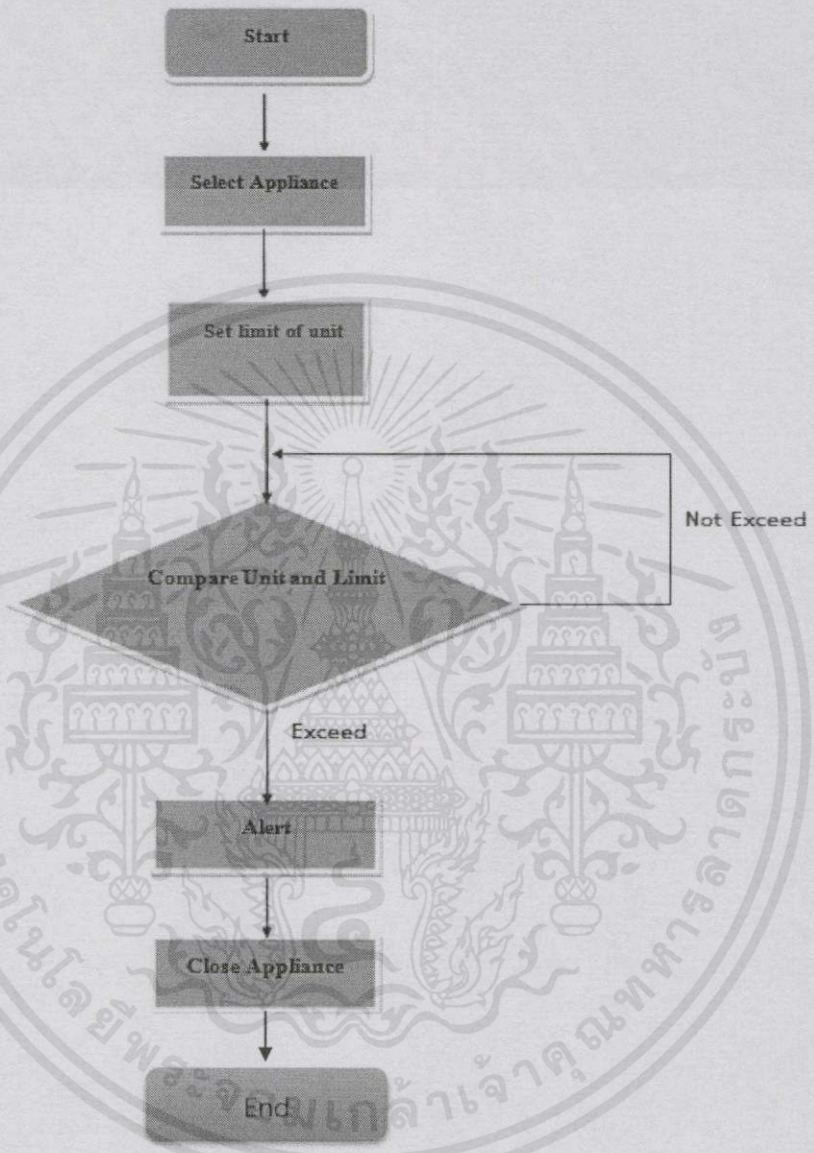
รูปที่ 3.3 ภาพรวมของระบบ

ระบบประกอบด้วย โมบายล์แอพพลิเคชัน เครื่องคอมพิวเตอร์ สมาร์ทมิเตอร์ และ หลอดไฟ โดยระบบจะแบ่งการทำงานออกเป็น 2 ส่วนดังนี้

1) โมบายล์แอพพลิเคชันกับมิเตอร์ – แอพพลิเคชันนี้จะเชื่อมต่อสื่อสารกับสมาร์ทมิเตอร์ ผ่านทางตัวกลางคือ เครื่องคอมพิวเตอร์ด้วย Wifi จากนั้นเครื่องคอมพิวเตอร์จะสื่อสารกับสมาร์ทมิเตอร์ด้วยโปรโทคอล DLMS/COSEM เพื่อนำข้อมูลที่ได้จากสมาร์ทมิเตอร์มาวิเคราะห์เพื่อการประหยัดพลังงานด้วยการควบคุมการทำงานของหลอดไฟภายในบ้าน
2) โมบายล์แอพพลิเคชันกับเครื่องใช้ไฟฟ้า – โมบายล์แอพพลิเคชันจะสามารถควบคุมหลอดไฟภายในบ้านได้ จะต้องควบคุมผ่าน Bridge โดยจะเชื่อมต่อกับBridge ด้วย Wifi จากนั้น Bridge จะเชื่อมต่อกับหลอดไฟภายในบ้านที่มี ZigBee ติดตั้งอยู่ โดยอาศัยการสื่อสารด้วย ZigBee ผ่านโปรโทคอล Light Link โดยโมบายล์แอพพลิเคชันสามารถควบคุมการ เปิด/ปิด ปรับความเข้มสี เปลี่ยนสี และ ปรับความสว่างของหลอดไฟได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.5 Flow chart ของฟังก์ชันตั้งค่าขีดจำกัดการใช้ไฟฟ้า



รูปที่ 3.4 Flow chart การตั้งค่าขีดจำกัดการใช้ไฟฟ้า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3 การออกแบบฐานข้อมูล

ใช้ Core data ใน Xcode ในการออกแบบฐานข้อมูล โดยได้ออกแบบฐานข้อมูลโดยแบ่งออกเป็น 2 ตาราง ดังนี้คือ Meter และ User

ตารางที่ 3.2 ตาราง Meter

Entity	Meter		
Key	Attributes	Data Type	Description
PK	Meter_ID	String	รหัสประจำตัวมิเตอร์
	kW	Float	ค่า kW
	kVAR	Float	ค่า kVAR
	kWh	Float	ค่า kWh
	Voltage	Float	ค่า Voltage
	Current	Float	ค่า Current
	timestamp	Date	วันที่

ตารางที่ 3.3 ตาราง User

Entity	User		
Key	Attributes	Data Type	Description
PK	User_ID	String	รหัสประจำตัวผู้ใช้งานไฟฟ้า
	Firstname	String	ชื่อจริงผู้ใช้งานไฟฟ้า
	Lastname	String	นามสกุลผู้ใช้งานไฟฟ้า
	Address	String	ที่อยู่ผู้ใช้งานไฟฟ้า
	Email	String	อีเมลล์ผู้ใช้งานไฟฟ้า
	Telephone	String	เบอร์โทรศัพท์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4 การพัฒนาแอพพลิเคชัน

3.4.1 การพัฒนาการเชื่อมต่อด้วย Socket

```
-(void)initNetworkCommunication {

    CFReadStreamRef readStream;
    CFWriteStreamRef writeStream;

   // CFStringRef host = CFSTR("161.246.5.83"); // name's ip address

    CFStringRef host = CFSTR("192.168.0.106");

    CFStreamCreatePairWithSocketToHost(NULL,host, 12345, &readStream, &writeStream);

    //(CFStringRef)@"localhost"

    inputStream  = (__bridge NSInputStream *)readStream;
    outputStream = (__bridge NSOutputStream *)writeStream;



    [inputStream setDelegate:self];
    [outputStream setDelegate:self];

    [inputStream scheduleInRunLoop:[NSRunLoop currentRunLoop] forMode:NSDefaultRunLoopMode];
    [outputStream scheduleInRunLoop:[NSRunLoop currentRunLoop] forMode:NSDefaultRunLoopMode];

    [inputStream open];
    [outputStream open];

}
```

รูปที่ 3.5 การเชื่อมต่อแบบ Socket

จากภาพเป็นฟังก์ชัน initNetworkCommunication ซึ่งใช้ในการสร้างการเชื่อมต่อแบบ Socket ระหว่าง ตัวแอพพลิเคชัน และ Host ที่ต้องการเชื่อมต่อ จากภาพ Host ที่ต้องการเชื่อมต่อมี IP Address คือ 192.168.0.106 และ เชื่อมไปที่ Port 12345

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
- (void)run{

    NSString *response  = [NSString stringWithString:@"read"];
    NSData *data = [[NSData alloc] initWithData:[response dataUsingEncoding:NSASCIIStringEncoding]];
    [outputStream write:[data bytes] maxLength:[data length]];

}

-(void) startRecord
{
    // Start record timer.
    interval
    = [[NSUserDefaults standardUserDefaults] integerForKey:kRecordIntervalKey];

    [self.recordTimer invalidate];
    self.recordTimer = [NSTimer timerWithTimeInterval:interval+9
                                               target:self
                                             selector:@selector(run)
                                             userInfo:nil
                                              repeats:YES];

    [[NSRunLoop currentRunLoop] addTimer:self.recordTimer
                                 forMode:NSDefaultRunLoopMode];
    [self.recordTimer fire];
}

-(void) stopRecord
{

    [self.recordTimer invalidate];


}
```

รูปที่ 3.6 การส่งข้อความ read ไปยัง host เพื่อที่จะอ่านค่า

จากภาพคือเราจะทำการส่งข้อความ “read” ไปยัง Host เพื่อที่จะอ่านค่า โดยเราจะทำการอ่านค่าเรื่อยๆโดยใช้ ฟังก์ชัน startRecord ซึ่งมีการสั่งให้ทำฟังก์ชัน run แบบวนซ้ำอยู่ และมีฟังก์ชัน stopRecord เพื่อเอาไว้หยุดการทำงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
while ([inputStream hasBytesAvailable]) {
    len = [inputStream read:buffer maxLength:sizeof(buffer)];
    if (len > 0) {

        NSString *output = [[NSString alloc] initWithBytes:buffer length:len encoding:NSASCIIStringEncoding];
```

1

```
        if (nil != output) {
            NSLog(@"server said: %@", output);

            //result.text = output;

            NSArray * ar = [output componentsSeparatedByString:@","];

            for(int i = 0; i < ar.count;i++){

                if(i == 0){
                    str = [ar objectAtIndex:0 ];
                }
                else if (i == 1){
                    str1 = [ar objectAtIndex:1];
                }
                else if (i == 2){
                    str2 = [ar objectAtIndex:2];
                }
                else if (i == 3){
                    str3 = [ar objectAtIndex:3];
                }
                else
                str4 = [ar objectAtIndex:4];
}
```

2

```
MeterLevel *lv
= [NSEntityDescription insertNewObjectForEntityForName:@"MeterLevel"
                                inManagedObjectContext:self.managedObjectContext];
lv.volt     = str;
lv.amp      =   str1;
lv.kW       =   [NSString stringWithFormat:@"%.2f",([str floatValue] * [str1 floatValue])/1000];
lv.kWh      =   [NSString stringWithFormat:@"%.2f",[lv.kW floatValue] * interval];
lv.kVAR     =   str4;

NSDate *today = [NSDate date];
NSDateFormatter *datef = [[NSDateFormatter alloc] init];
[datef setDateFormat:@"dd/MM/yyyy hh:mm:ss"];
NSString *dates = [datef stringFromDate:today];

lv.timestamp = dates;
```

3

รูปที่ 3.7 ส่วนที่ใช้เช็คว่ามีข้อมูลถูกส่งเข้ามาหรือไม่

1. เป็นส่วนที่ไว้เช็คว่ายังมีข้อมูลถูกส่งเข้ามาหรือเปล่าหากมีให้ใส่ข้อมูลนั้นให้แก่ตัวแปร output
2. เนื่องจากข้อมูลที่ส่งมาเป็นข้อความแบบสตริงยาวติดกัน ทำให้ต้องแยกออกเป็นส่วนๆก่อน โดยใช้ฟังก์ชัน componentSeperatedByString@”,” จากนั้นจึงค่อยนำส่วนของข้อความที่แยกแล้วไปใส่ให้ตัวแปรอื่นๆต่อไป

เอกสารนี้เป็นเอกสาร... ไม่ว่ากรณีใดๆทั้งสิ้น... และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. ทำการบันทึกค่าลงใน Core Data โดยใช้ตัวแปรจากข้อ 2

```
- (IBAction)recordIntervalChanged:(id)sender {

    UIStepper* stepper = sender;

    // Save new value.
   // NSInteger interval = stepper.value;
    interval = stepper.value;
    [[NSUserDefaults standardUserDefaults] setInteger:interval
                                               forKey:kRecordIntervalKey];
    [[NSUserDefaults standardUserDefaults] synchronize];

    // Update UI.
    self.recordIntervalLabel.text = [NSString stringWithFormat:@"%d",interval];

    [self stopRecord];
    [self startRecord];
}
```

รูปที่ 3.8 ฟังก์ชันที่ใช้ปรับเวลาในการรับค่า

เป็นฟังก์ชันที่ไว้ใช้ในการปรับเวลาในการรับค่ามาจาก Host โดยตัวปรับจะอยู่ในรูปแบบของ stepper คือเป็นปุ่มที่มี – และ + ติดกัน เมื่อปรับแล้วค่าที่ปรับจะถูกใส่ให้แก่ interval โดยค่านี้จะถูกนำไปใช้ใน ฟังก์ชัน startRecord เพื่อเริ่มการทำงานต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4.2 การพัฒนาส่วนการตั้ง Peak load

```
- (void)viewDidLoad {
    [super viewDidLoad];

    NSUserDefaults * defaults = [NSUserDefaults standardUserDefaults];
    limitFromUser = [defaults objectForKey:@"limitS"];
    limitFromMeter = [defaults objectForKey:@"limitkW"];

    NSLog(@" %f  ,  %f", [limitFromMeter floatValue],[limitFromUser floatValue]);
    if (([limitFromUser floatValue] < [limitFromMeter floatValue]) && (sendExceed.on == YES)){



        self.valueOn = NO;
        [self sendLightState];

    }

    NSLog(@" %@ , %@" , limitFromMeter , limitFromUser );

    self.title = self.light.name;

    self.redButton.backgroundColor = [UIColor redColor];
    self.blueButton.backgroundColor = [UIColor blueColor];
    self.greenButton.backgroundColor = [UIColor greenColor];
    self.yellowButton.backgroundColor = [UIColor yellowColor];
    self.violetButton.backgroundColor = [UIColor purpleColor];
    self.orangeButton.backgroundColor = [UIColor orangeColor];

    [self.scrollView setContentSize:self.scrollContentView.frame.size];

    [NSTimer scheduledTimerWithTimeInterval:1.0
                                     target:self selector:@selector(viewDidLoad)
                                     userInfo:nil repeats:NO];
}
```

1

2

รูปที่ 3.9 การพัฒนาส่วนการตั้งค่า Peak load

1. เริ่มต้นจะมีการนำข้อมูลค่า kW ที่ได้เซฟไว้ใน MainViewController เข้ามาใช้ในหน้านี้โดยใส่ให้กับค่า limitFromMeter เพื่อที่จะได้นำไปเปรียบเทียบกับค่า limitFromUser ซึ่งเป็นค่าที่ได้จากการตั้งค่า Peak Load ในหน้าของการ Setting นั่นเอง ซึ่งถ้าเป็นไปตามเงื่อนไขข้างต้นสถานการณ์เปิดปิดของไฟจะถูกตั้งค่าเป็นปิดและค่านี้จะถูกส่งผ่านไปให้แก่ Bridge ด้วยคำสั่ง sendLightState ต่อไป
2. จะมีการวนซ้ำเพื่อเช็คค่า limitFromUser กับ limitFromMeter ทุกๆ 1 วินาที

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4.3 การพัฒนาส่วนควบคุมหลอดไฟ

```
- (PHLightState *)createLightState {
    /***************************************************
     The PHLightState class is used as a parameter for the
     Hue SDK. It contains the attribute settings for an individual
     light. This method creates it from the current
     user interface settings for the light
     ***************************************************/

    // Create an empty lightstate
    PHLightState *lightState = [[PHLightState alloc] init];


        if (self.sendOn.on) {
        [lightState setOnBool:self.valueOn.on];
    }

    // Check if hue value should be send
    if (self.sendHue.on) {
        [lightState setHue:[NSNumber numberWithInt:((int)self.valueHueSlider.value)]];
    }

    // Check if saturation value should be send
    if (self.sendSat.on) {
        [lightState setSaturation:[NSNumber numberWithInt:((int)self.valueSatSlider.value)]];
    }

    // Check if brightness value should be send
    if (self.sendBri.on) {

    |
        [lightState setBrightness:[NSNumber numberWithInt:((int)self.valueBriSlider.value)]];

    }

    // Check if xy values should be send
    if (self.sendXY.on) {
        [lightState setX:[NSNumber numberWithFloat:self.valueXSlider.value]];
        [lightState setY:[NSNumber numberWithFloat:self.valueYSlider.value]];
    }
```

รูปที่ 3.10 การสร้าง lightState

จากภาพจะเป็นการสร้าง lightState ขึ้นมาเพื่อเตรียมพร้อมที่จะทำการส่งต่อไป โดยก่อนอื่นต้องมีการตรวจสอบก่อนว่า Switch Enable ของแต่ละค่านั้นมีการเปิดอยู่หรือไม่ ถ้าเปิดอยู่เราจึงจะสามารถปรับค่านั้นๆแล้วส่งไปเป็น lightState ได้ จากตัวอย่างจะมีค่าอยู่ 5 ค่าคือ ค่า Hue , ค่า Saturation , ค่า Brightness , ค่า X และ ค่า Y ซึ่ง เป็นค่าของสีใน CIE Color Space

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
- (IBAction)sendButton:(id)sender {
    // Create a lightstate based on selected options
    PHLightState *lightState = [self createLightState];


    // Create a bridge send api, used for sending commands to bridge locally
    id<PHBridgeSendAPI> bridgeSendAPI = [[[PHOverallFactory alloc] init] bridgeSendAPI];

    // Send lightstate to light
    [bridgeSendAPI updateLightStateForId:self.light.identifier withLighState:lightState completionHandler:^(NSArray *errors)
        if (UIAppDelegate.showResponses || errors != nil) {
            NSString *message = [NSString stringWithFormat:@"%@: %@", NSLocalizedString(@"Errors", @""), errors != nil ?
                                 errors : NSLocalizedString(@"none", @"")];

            UIAlertView *errorAlert = [[UIAlertView alloc] initWithTitle:NSLocalizedString(@"Response", @"")
                                                                 message:message
                                                                delegate:nil
                                                       cancelButtonTitle:nil
                                                       otherButtonTitles:NSLocalizedString(@"Ok", @""), nil];
            [errorAlert show];
        }
    }];
}
```

รูปที่ 3.11 Action เมื่อผู้ใช้งานกดส่ง lightState

จากภาพจะเป็น Action เพื่อให้ผู้ใช้งานกดใช้งานเพื่อส่ง lightState ไปยัง Bridge โดยจะมีการกำหนดหลอดไฟที่จะทำการส่งค่าไปให้ มีการกำหนด lightState ที่ได้ปรับไว้ และมีการจัดการกับ error ในกรณีที่มีปัญหาเกิดขึ้น โดยใช้ block completionHandler

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.5 Library ที่ใช้ในการพัฒนาแอพพลิเคชัน

3.5.1 TapKu Library

เป็น Library ที่เกี่ยวกับการทำปฏิทินโดยคลาสที่ยกมาใช้มีดังนี้

1. TKCalendarMonthView ใช้ในการสร้างตัวปฏิทิน
2. TKCalendarMonthViewController ใช้ในการเรียกตัวปฏิทินพร้อมทั้งปรับตำแหน่งต่าง
3. TKCalendarMonthTableViewController ใช้ในการสร้างตารางใต้ปฏิทินที่เอาไว้ใช้แสดงค่าต่างๆ

3.5.2 SDKWIzard ของ Philips HUE

เป็นตัวช่วยในการเชื่อมต่อระหว่างแอพพลิเคชันกับตัว Bridge โดยที่ใช้มีดังนี้

1. PHBridgeSelectionViewController ใช้ในการเลือก Bridge เพื่อเอามาใช้ในการควบคุมหลอดไฟ

   -(void)bridgeSelectedWithIpAddress:(NSString*)ipAddress andMacAddress:(NSString *)macAddress; เป็นฟังก์ชันที่ใช้ในการกำหนดค่า ipAddress กับ macAddress ให้กับ Bridge ที่จะใช้งาน

2. PHBridgePushLinkViewController ใช้ในการ authentication เพื่อเข้าใช้งาน Bridge

   - (void)startPushLinking เป็นฟังก์ชันที่ใช้ในการเริ่มการ authentication โดยจะมีการเช็คสถานะต่างๆก่อนการทำ Authentication หลังจากนั้นจึงทำการ Authentication โดยใช้ฟังก์ชัน startPushlinkAuthentication

   - (void)authenticationSuccess หากการ Authentication สำเร็จจะเข้ามาทำงานภายในฟังก์ชันนี้ และจะมีการเรียกฟังก์ชัน pushLinkSuccess เพื่อทำการเริ่ม enableLocalheartbeat ในรอบต่อไป

3. PHBridgesConfigurationViewController ใช้ในการปรับค่าต่างๆของตัว Bridge เช่นค่า ipAddress , net mask gateway และ proxyAddress เป็นต้น โดยจะทำการตั้งค่าผ่านฟังก์ชัน updateConfigurationWithConfiguration

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

-(void) updateValues รับค่ามาจาก cache ของ Bridge เพื่อมาทำการ update interface ให้เป็นไปตามค่าที่ตั้งไว้

-(void) save ใช้ในการบันทึกค่าต่างๆที่ได้ตั้งไว้ โดยนำไปใส่ใน cache ของ Bridge

4. PHBridgesConfigurationViewController ใช้ในการปรับค่าต่างๆ เช่น เปลี่ยนชื่อของหลอดไฟ หรือใช้ในการค้นหา Bridge ใหม่ หรือ ใช้ค้นหาหลอดไฟใหม่ได้

   -(void) updateLights ใช้ในการอ่านหลอดไฟมาจาก cache ของ bridge แล้วทำการจัดเรียงโดยดูจาก identifier

   -(void) renameLight ใช้ในการเปลี่ยนชื่อของหลอดไฟ

5. PHLoadingViewController ใช้ในการแสดงหน้า Load View ตอนทำการเชื่อมต่อกับ Bridge
6. PHFindLightStartViewController ใช้ในการเริ่มต้นการค้นหาหลอดไฟโดย Bridge

   -(IBAction) startSearch กดใช้งานเพื่อเริ่มต้นการค้นหาหลอดไฟ

7. PHLightResultViewController ใช้ในการจัดการกับหลอดไฟที่ค้นเจอ เช่น นำไปเพิ่มใน cache ของ Bridge

   -(void) updateLightsFound ทำการอัพเดทหลอดไฟลงไปใน cache ของ Bridge

   -(void) startSearch ทำการค้นหาหลอดไฟ โดยทำการค้นหา ทุกๆ 1 วินาที และจะ timeOut เมื่อผ่านไปครบ 60 วินาที และยังไม่เจอหลอดไป

8. PHFindLightManualEntryViewController ใช้ในการค้นหาหลอดไฟแบบ manual โดยการใส่ค่าต่างๆเข้าไปโดยตรง เช่น serial ของ หลอดไฟ เป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

การทดลองและผลการทดลอง

ในการทดลองนี้จะใช้อุปกรณ์ iPad Generation 4 ในการทำการทดลองอ่านค่าจากสมาร์ทมิเตอร์ และการทดลองการควบคุมไฟ

4.1 การทดลองการแสดงผลของแอพพลิเคชัน

4.1.1 การแสดงผลของหน้าต่างแสดงข้อมูลการใช้ไฟฟ้าค่าไฟฟ้า

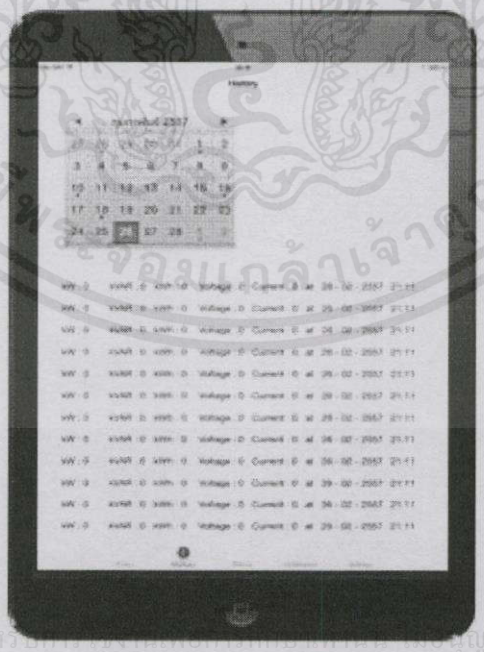
1. มีตัวเลขที่เอาไว้แสดงค่าต่างๆรับเข้ามาจากมิเตอร์ คือ kW kWh kVAR Voltage และ Current โดยใช้ทศนิยม 2 หลักในการแสดงผล
2. มีโหมด Scroll Mode ให้ผู้ใช้งานเลือกว่าจะ ON หรือ OFF ได้ โดยถ้าเลือกเป็น ON จะแสดงลูกศรขึ้นมาทั้งฝั่งซ้ายและฝั่งขวาของค่าที่มิเตอร์แสดงอยู่ ผู้ใช้งานสามารถกดลูกศรซ้าย หรือขวา เพื่อเปลี่ยนค่าที่ผู้ใช้งานต้องการจะดูได้ แต่ในกรณีที่ Scroll Mode เป็น OFF การแสดงผลจะเป็นแบบ Default คือ ค่าที่แสดงจะเปลี่ยนไปเรื่อยๆ

รูปที่ 4.1 หน้าแสดงข้อมูลการใช้ไฟฟ้า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.1.2 หน้าแสดงข้อมูลการใช้ไฟฟ้าย้อนหลัง

1. มีปฏิทินเอาไว้เลือก วัน เดือน และ ปี ที่เราต้องการดูข้อมูลการใช้ไฟฟ้าได้ โดยปฏิทินจะอยู่บนสุดเพื่อความสะดวกในการใช้งาน
2. วันที่ในปฏิทินเปรียบเสมือนปุ่มที่สามารถกดเพื่อดูได้ว่าภายในวันนั้นจำนวนยูนิตรวมที่ใช้งานภายในวันนั้นมีค่าเท่าไหร่
3. มีตารางที่แสดงข้อมูลการใช้ไฟฟ้าต่างๆ โดยในแต่ละเซลล์ของตารางจะประกอบไปด้วยข้อมูลคือ
    1) kW
    2) kWh
    3) kVAR
    4) Voltage
    5) Current
4. วันและเวลาที่ได้รับค่าเข้ามาโดยวันที่และเวลามีฟอร์แมตเป็น “dd/MM/yyyy HH:mm:ss“
5. ตารางสามารถยาวขึ้นได้เรื่อยๆ โดยมีแถว Scroll เพื่อเลื่อนขึ้นเลื่อนลงได้
6. เมื่อมีข้อมูลเข้ามาใหม่จะถูกใส่เข้ามาภายในตารางแบบ Animation Fade

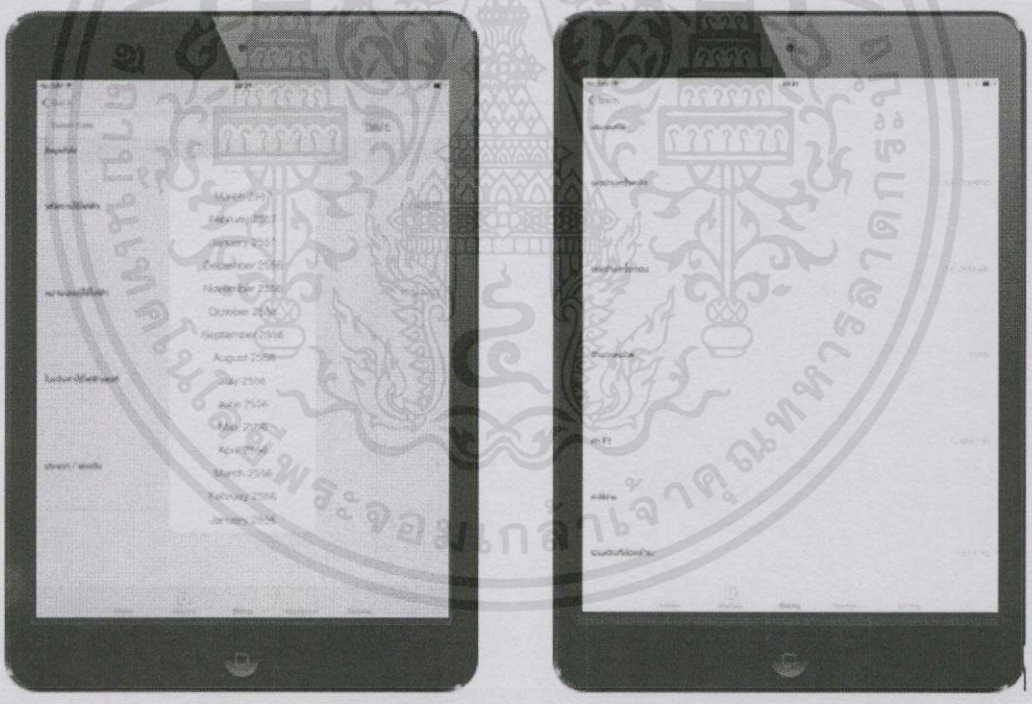


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 4.2 หน้าแสดงข้อมูลการใช้ไฟฟ้าย้อนหลัง

4.1.3 หน้าแสดงข้อมูลการเงิน

1. มีตารางเพื่อแสดงข้อมูลต่างๆ ซึ่งประกอบไปด้วย รหัสการใช้ไฟฟ้า หมายเลขผู้ใช้ไฟฟ้า ใบแจ้งค่าใช้ไฟฟ้าเลขที่ และ ประเภท / แรงดัน ค่ายูนิตครั้งก่อน ค่ายูนิตครั้งหลัง ค่า Ft และค่าใช้จ่ายรวม
2. แบ่ง 3 ส่วน คือ ส่วนข้อมูลการใช้งานไฟฟ้า ส่วนจำนวนไฟฟ้าที่ใช้ และ ส่วนแสดงเงินที่ต้องชำระ
3. ตารางมีรูปแบบคือ หัวข้ออยู่ทางซ้าย เนื้อหาอยู่ทางขวา
4. ด้านบนของตารางมีปุ่มไว้ให้ใช้เลือกเดือนกับปีที่เราต้องการจะดูได้ โดยเมื่อกดปุ่มเลือก จะแสดงหน้าต่างวันเวลาให้เลือกเป็นแบบ Action sheet



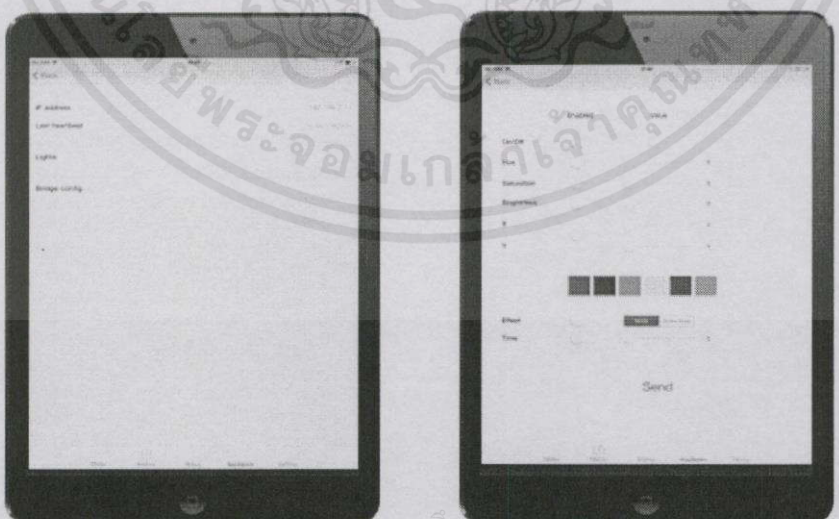
รูปที่ 4.3 หน้าแสดงข้อมูลการเงิน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.1.4 หน้าแสดงการควบคุมหลอดไฟ

มีตารางซึ่งแบ่งการทำงานออกเป็นสองส่วนคือ

1. ส่วนแรกคือส่วนที่นำไว้ใช้เชื่อมต่อกับตัว Bridge โดย จะมี IP Address ของ bridge ขึ้นมาแสดงเมื่อเราทำการเชื่อมต่อแล้ว และมี Last Heartbeat ที่เอาไว้แสดงเวลาสุดท้ายที่แอพพลิเคชันได้เชื่อมต่อกับตัว Bridge
2. ส่วนที่สองคือส่วนที่ทำการควบคุมหลอดไฟฟ้า ในส่วนนี้จะสามารถใช้ได้ก็ต่อเมื่อสามารถเชื่อมต่อกับ Bridge ได้แล้วเท่านั้น โดยจะขึ้นหลอดไฟมาให้เลือก 3 หลอด
3. เมื่อเลือกหลอดไฟหลอดใดหลอดหนึ่งแล้ว จะปรากฏหน้าการควบคุมหลอดไฟขึ้นมา โดยมีค่าต่างๆที่สามารถควบคุมได้ดังนี้
    1) Enable คือ ส่วนที่ต้องเปิดหากต้องการควบคุมหลอดไฟ
    2) Hue หรือ ความเข้มของสี โดยมีค่าอยู่ที่ 0 - 65535
    3) Saturation หรือ ความอิ่มตัวของสี โดยมีค่าอยู่ที่ 0 - 65535
    4) Brightness หรือ ความสว่างของหลอดไฟ โดยมีค่าอยู่ที่ 0 - 254
    5) ค่า X , Y หรือค่าสีตาม CIE Color Space โดยมีค่าอยู่ที่ 0 – 1
    6) โหมด Color Loop คือการทำให้หลอดไฟเปลี่ยนสีไปเรื่อยๆ
4. เมื่อตั้งค่าที่ต้องการได้เรียบร้อยแล้ว ต้องกดปุ่ม Send เพื่อส่งคำสั่งต่างๆไปที่ Bridge เพื่อให้ Bridge ส่งไปให้หลอดไฟเพื่อทำงานต่อไป

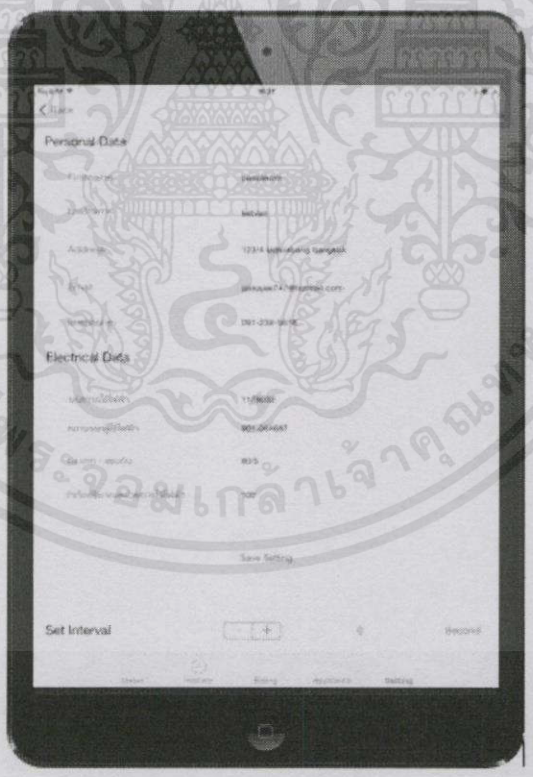


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 4.4 หน้าแสดงการควบคุมหลอดไฟ

4.1.5 หน้าแสดงการตั้งค่า

1. แบ่งเป็นสามส่วนคือ ข้อมูลส่วนตัว ส่วนข้อมูลการใช้งานไฟฟ้า และ การตั้งค่าในการใช้งานตัวแอพพลิเคชัน
2. ข้อมูลส่วนตัว ประกอบไปด้วย ชื่อ นามสกุล ที่อยู่ อีเมล์ และ เบอร์โทรศัพท์ โดยจะมี Text Field ไว้ให้ใส่ข้อมูลลงไป
3. ข้อมูลการใช้งานไฟฟ้า ประกอบไปด้วย รหัสการใช้งานไฟฟ้า เลขที่การใช้ไฟฟ้า ประเภท / แรงดัน และ Peak Load โดยจะมี Text Field ไว้ให้ใส่ข้อมูลลงไป
4. เมื่อทำการปรับค่าทั้งหมดแล้วจะมีปุ่ม Save เพื่อเอาไว้บันทึกข้อมูลลงในตัวโปรแกรม เมื่อปิดโปรแกรมไปข้อมูลก็ยังอยู่
5. การตั้งค่าการใช้งานในแอพพลิเคชัน ประกอบด้วย Set interval คือการ ตั้งค่าช่วงเวลาในการรับค่าที่ส่งมาจากมิเตอร์ โดยมีปุ่ม + และ – ในการกดเพื่อปรับค่า

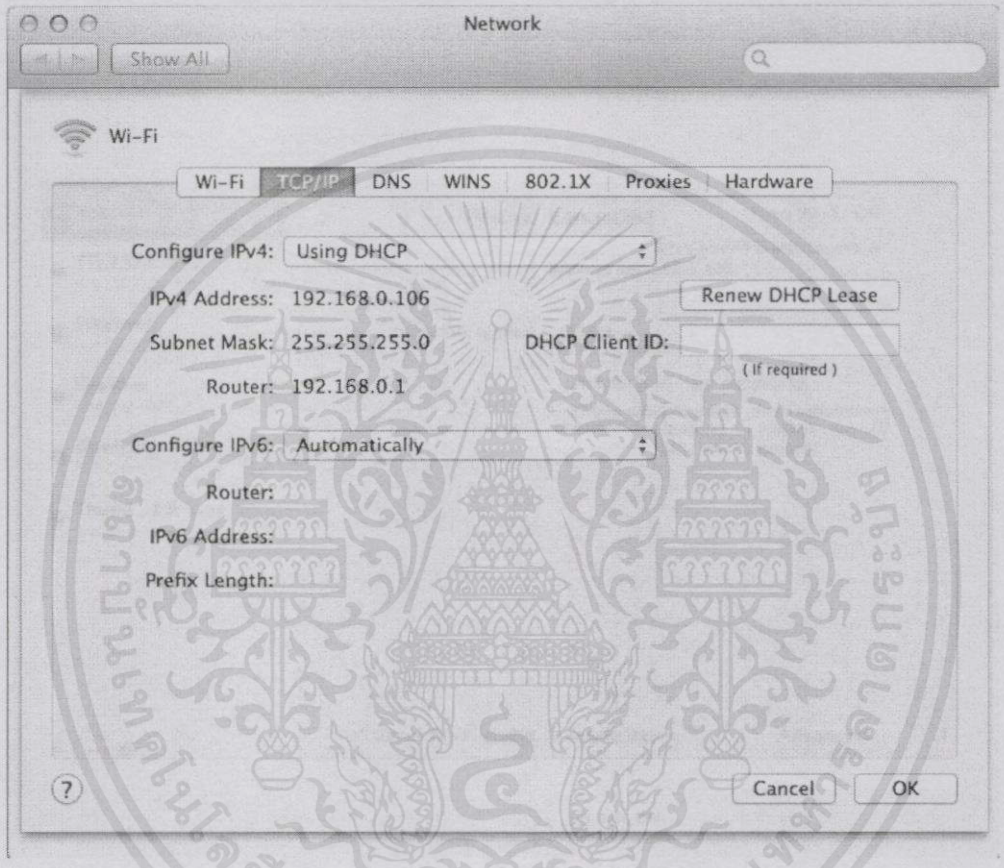


รูปที่ 4.5 หน้าแสดงการตั้งค่าผู้ใช้งาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2 ทดลองการอ่านค่าจากมิเตอร์

1. เริ่มด้วยการดู IP Address ของเครื่อง PC เราจะทำการรับข้อมูลเข้ามาจากรูป IP คือ 192.168.0.106



รูปที่ 4.6 ดู IP Address ของเครื่องคอมพิวเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. ทำการแก้ IP Address ในโค้ด ที่บรรทัด

CFStringRef host = CFSTR("192.168.0.106);

```
-(void)initNetworkCommunication {

    CFReadStreamRef readStream;
    CFWriteStreamRef writeStream;

   // CFStringRef host = CFSTR("161.246.5.83"); // name's ip address

    CFStringRef host = CFSTR("192.168.0.106");

    CFStreamCreatePairWithSocketToHost(NULL,host, 12345, &readStream, &writeStream);

    //(CFStringRef)@"localhost"

    inputStream  = (__bridge NSInputStream *)readStream;
    outputStream = (__bridge NSOutputStream *)writeStream;



    [inputStream setDelegate:self];
    [outputStream setDelegate:self];

    [inputStream scheduleInRunLoop:[NSRunLoop currentRunLoop] forMode:NSDefaultRunLoopMode];
    [outputStream scheduleInRunLoop:[NSRunLoop currentRunLoop] forMode:NSDefaultRunLoopMode];

    [inputStream open];
    [outputStream open];
}
```

รูปที่ 4.7 ทำการแก้ไข IP Address

3. จากนั้นทำการรับข้อมูลแล้วมาเก็บลงใน Core Data เนื่องจากข้อมูลที่ได้รับมาเป็น String ชุดยาว จึงต้องแบ่งออกเป็นส่วนๆด้วยฟังก์ชัน ComponentSeperatedByString:NSString ก่อนจึงจะสามารถนำข้อมูลแต่ละส่วนไปบันทึกได้

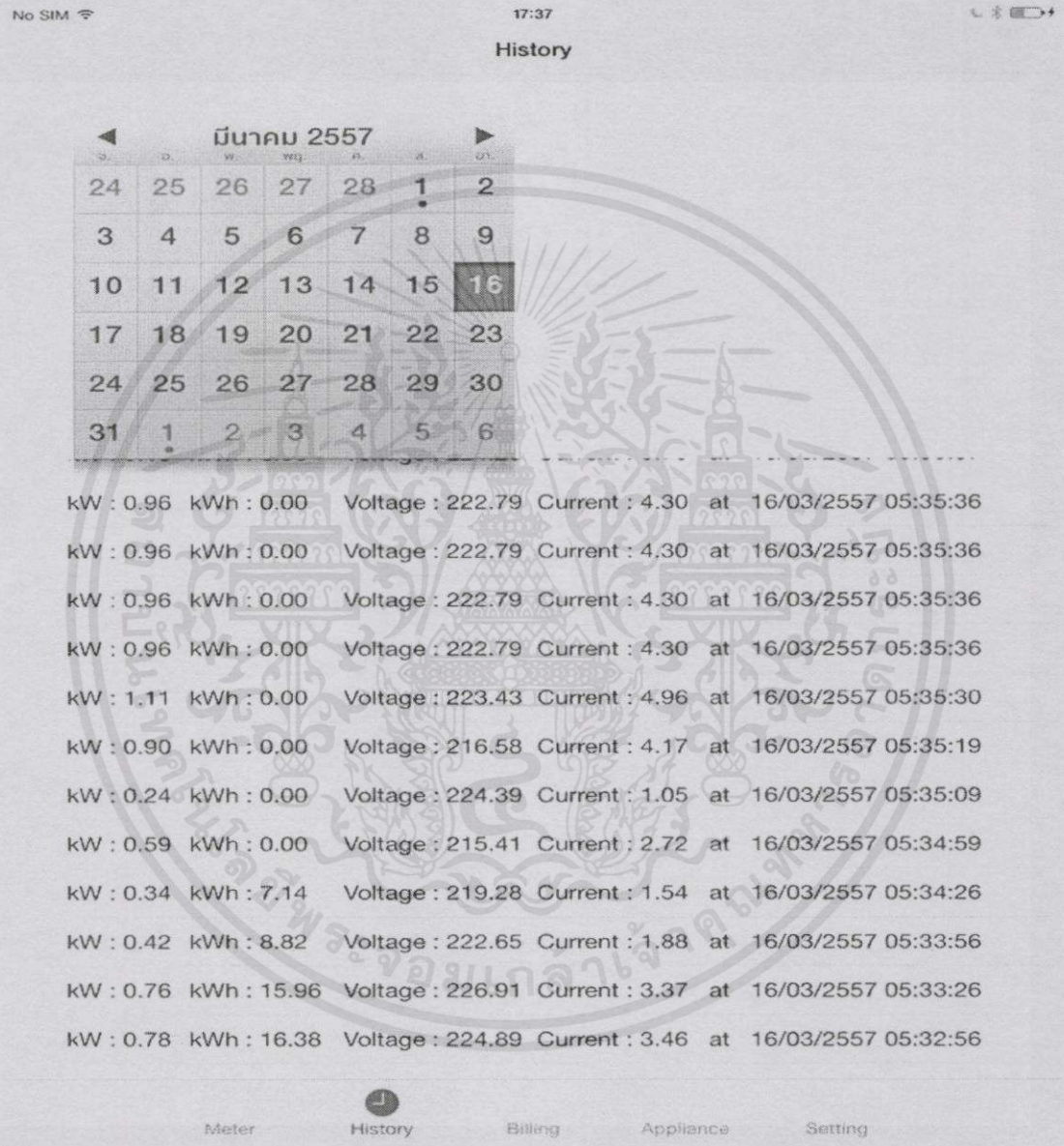
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
while ([inputStream hasBytesAvailable]) {
    len = [inputStream read:buffer maxLength:sizeof(buffer)];
    if (len > 0) {

        NSString *output = [[NSString alloc] initWithBytes:buffer length:len encoding:NSASCIIStringEncoding];

        if (nil != output) {
            NSLog(@"server said: %@", output);

            //result.text = output;

            NSArray * ar = [output componentsSeparatedByString:@","];

            for(int i = 0; i < ar.count;i++){

                if(i == 0){
                    str = [ar objectAtIndex:0 ];
                }
                else if (i == 1){
                    str1 = [ar objectAtIndex:1];
                }
                else if (i == 2){
                    str2 = [ar objectAtIndex:2];
                }
                else if (i == 3){
                    str3 = [ar objectAtIndex:3];
                }
                else
                str4 = [ar objectAtIndex:4];
        }

            MeterLevel *lv
            = [NSEntityDescription insertNewObjectForEntityForName:@"MeterLevel"
                                        inManagedObjectContext:self.managedObjectContext];
            lv.volt    = str;
            lv.amp     =  str1;
            lv.kW      =   [NSString stringWithFormat:@"%.2f",([str floatValue] * [str1 floatValue])/1000];
            lv.kWh     =   [NSString stringWithFormat:@"%.2f",[lv.kW floatValue] * interval];
            lv.kVAR    =   str4;

            NSDate *today = [NSDate date];
            NSDateFormatter *datef = [[NSDateFormatter alloc] init];
            [datef setDateFormat:@"dd/MM/yyyy hh:mm:ss"];
            NSString *dates = [datef stringFromDate:today];

            lv.timestamp = dates;
```

รูปที่ 4.8 การเก็บข้อมูลลง Core Data

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ผลลัพธ์ที่ได้จากการทดลอง



รูปที่ 4.9 ผลลัพธ์ที่ได้จากการอ่านค่าจากสมาร์ทมิเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

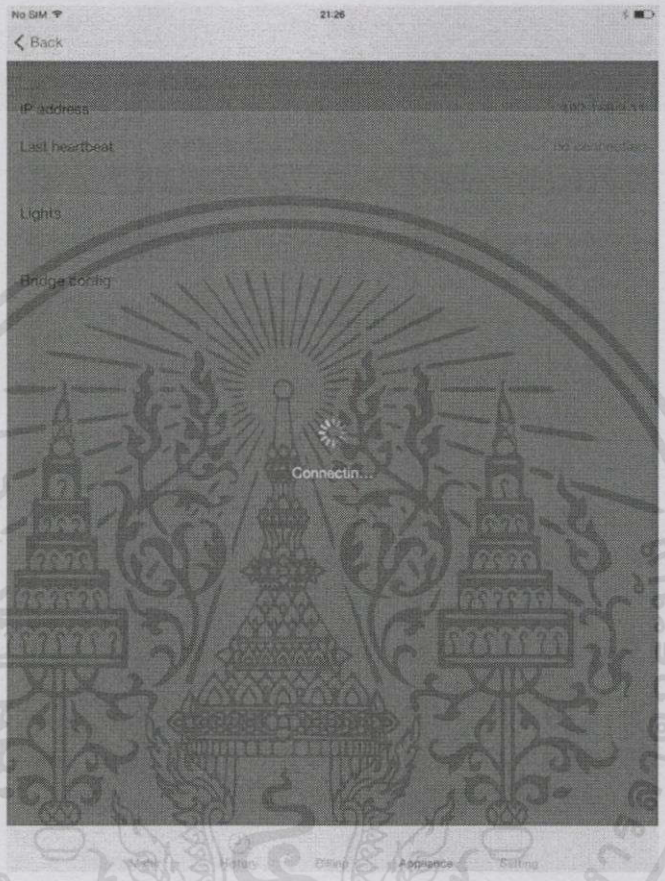
4.3 ทดลองการควบคุมหลอดไฟผ่านแอพพลิเคชัน

ขั้นตอนแรกทำการติดตั้ง Bridge กับ Router ภายในบ้าน

รูปที่ 4.10 ทำการเชื่อมต่อ Bridge เข้ากับ Router

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากนั้นทำการเปิดแอพพลิเคชันไปที่เมนู Appliance แอพพลิเคชันจะทำการเชื่อมต่อกับ Bridge ให้อัตโนมัติ โดยจะมีคำว่า Connecting เพื่อแสดงถึงกำลังทำการเชื่อมต่อกับ Bridge



รูปที่ 4.11 การ Connect ระหว่าง Bridge และ Application

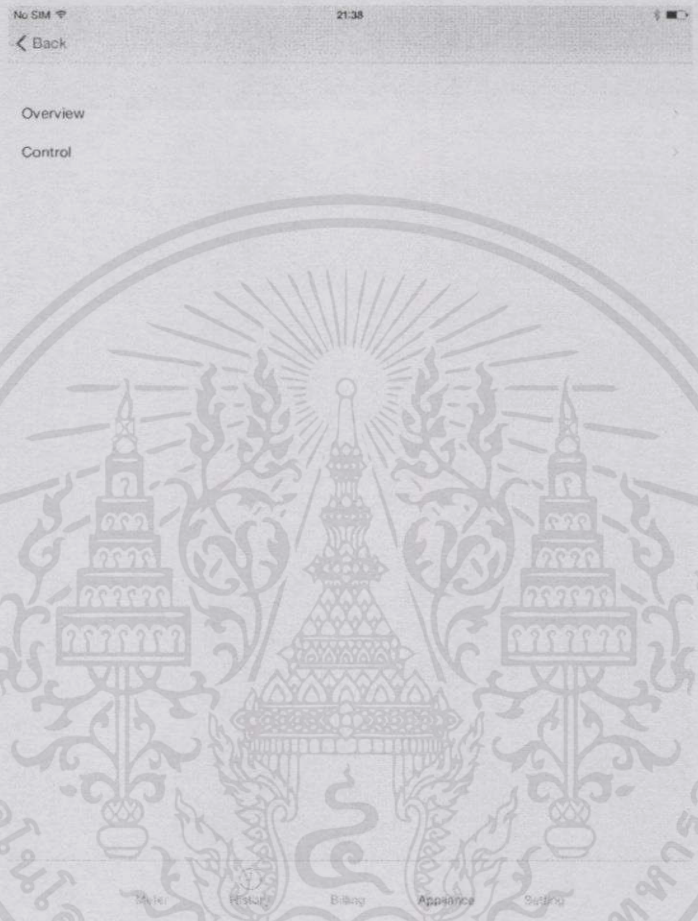
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อทำการเชื่อมต่อเรียบร้อยแล้วจะมีหน้า Appliance แสดงดังนี้ ทำการกดที่ Lights เพื่อเข้าสู่หน้าการควบคุมหลอดไฟต่อไป

รูปที่ 4.12 หน้าต่าง Appliance

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

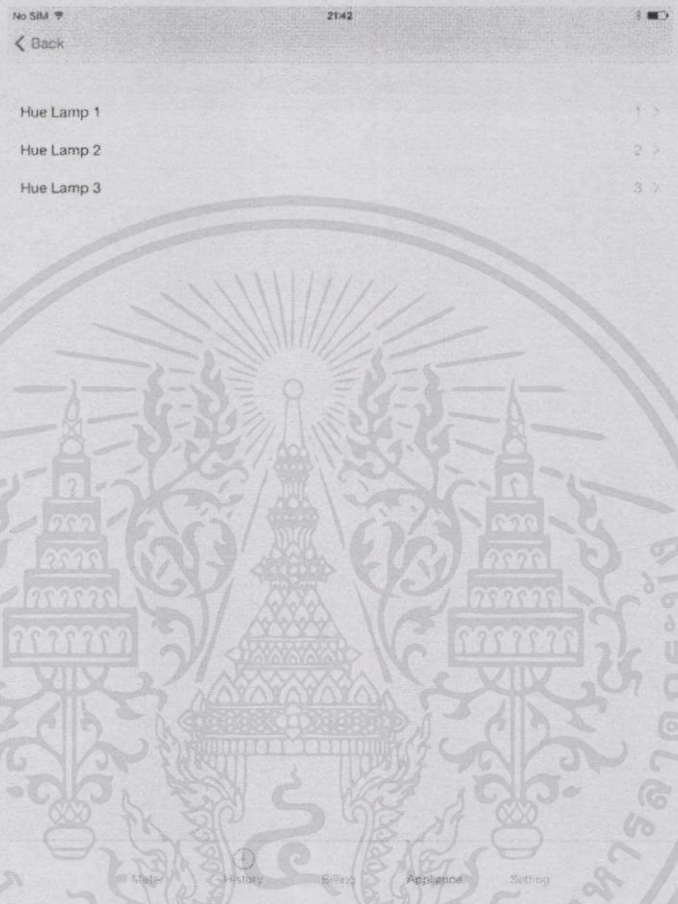
หลังจากกดที่ Lights แล้วจะพบกับ Overview และ Control ให้กดที่ Control เพื่อเข้าสู่หน้าตาการควบคุมหลอดไฟต่อไป



รูปที่ 4.13 หน้าต่าง Overview และ Control

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

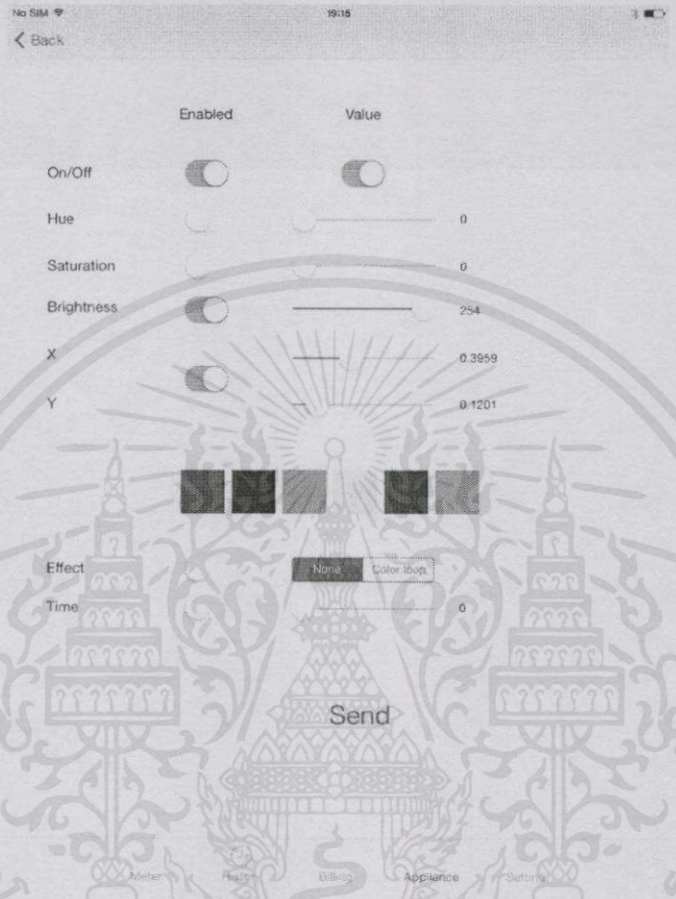
เมื่อกดที่ Control จะพบกับรายชื่อของหลอดไฟทั้งหมด ซึ่งในที่นี้จะประกอบไปด้วย Hue lamp 1 2 และ 3 ให้ทำการเลือกควบคุมหลอดไฟที่ต้องการ ในที่นี้จะเลือกหลอดที่ 1



รูปที่ 4.14 หน้าต่างรายชื่อหลอดไฟ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

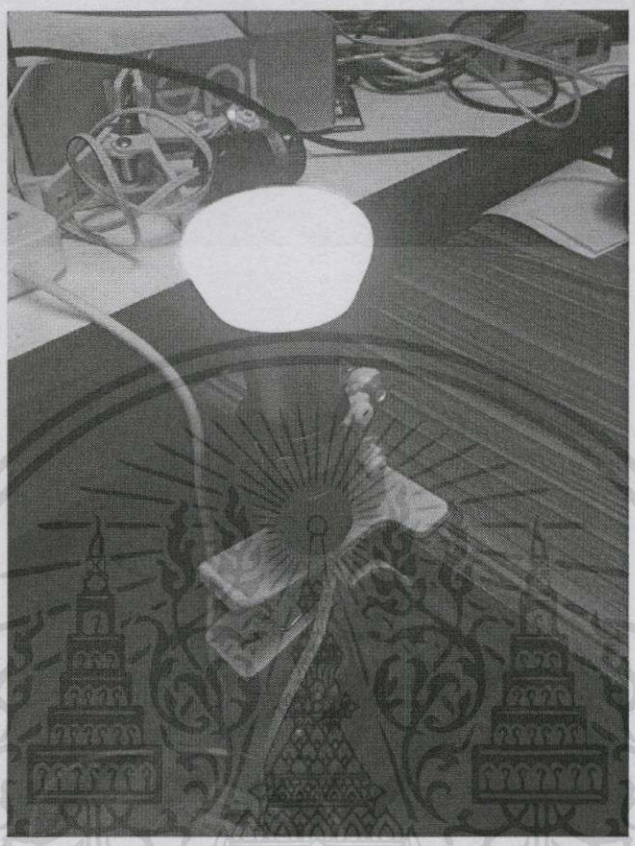
เมื่อกดที่ Hue lamp 1 จะพบกับหน้าต่างการตั้งค่าหลอดไฟ



รูปที่ 4.15 หน้าต่างการควบคุมหลอดไฟ

ได้ทำการทดลองควบคุมหลอดไฟผ่านแอพพลิเคชันโดยทำการเชื่อมต่อกับ Bridge ให้เรียบร้อย จากนั้นทดลองทำการควบคุมในการปรับความเข้มแสง ปรับความสว่าง จากการปรับค่าข้างต้น จากนั้นกดปุ่ม Send เพื่อให้หลอดไฟตอบสนองต่อการปรับแต่งค่าที่เราได้ตั้งไว้ ผลลัพธ์ออกมาเป็นดังนี้ ได้ไฟแสงสีชมพูออกม่วงๆ ดังภาพ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.16 ผลลัพธ์จากการทดลองการใช้งานควบคุมหลอดไฟ

จากนั้นได้ทำการทดลองในส่วนของ effect loop หลอดไฟสามารถเปลี่ยนสีไปเรื่อยๆตามที่ต้องการได้ถูกต้องตามที่ต้องการ

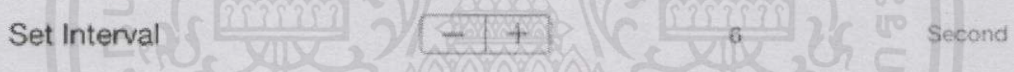
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.4 การทดลองการตั้งค่าเวลาที่ใช้ในการรับค่าจากมิเตอร์

```
- (IBAction)recordIntervalChanged:(id)sender {

    UIStepper* stepper = sender;

    // Save new value.
   // NSInteger interval = stepper.value;
    interval = stepper.value;
    [[NSUserDefaults standardUserDefaults] setInteger:interval
                                               forKey:kRecordIntervalKey];
    [[NSUserDefaults standardUserDefaults] synchronize];

    // Update UI.
    self.recordIntervalLabel.text = [NSString stringWithFormat:@"%d",interval];

    [self stopRecord];
    [self startRecord];
}
```

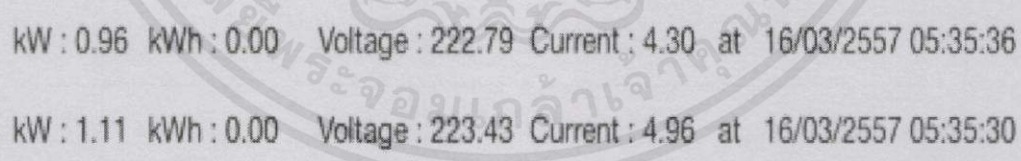
รูปที่ 4.17 โค้ดที่ใช้ในการปรับค่าเวลา

เมื่อทำการตั้งค่า Interval = 6



รูปที่ 4.18 ทำการตั้งค่า Interval = 6

ได้ผลลัพธ์ดังนี้



รูปที่ 4.19 ผลลัพธ์เมื่อทำการตั้งค่า Interval = 6

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5
บทสรุปและข้อเสนอแนะ

5.1 บทสรุปของโครงงาน

Smart Meter and Smart Home เป็นโครงงานพัฒนาโมบายล์แอพพลิเคชันที่ใช้อ่านและบันทึก ค่าไฟฟ้าจากสมาร์ทมิเตอร์และควบคุมหลอดไฟภายในบ้าน จากการพัฒนาโครงงานได้ข้อสรุปดังนี้

โมบายล์แอพพลิเคชันจะเชื่อมต่อสื่อสารกับสมาร์ทมิเตอร์ ผ่านทางตัวกลางคือ เครื่องคอมพิวเตอร์ด้วย Wifi จากนั้นเครื่องคอมพิวเตอร์จะสื่อสารกับสมาร์ทมิเตอร์ด้วยโปรโทคอล DLMS/COSEM เพื่อนำข้อมูลที่ได้จากสมาร์ทมิเตอร์มาวิเคราะห์เพื่อการประหยัดพลังงานด้วยการควบคุมการทำงานของหลอดไฟภายในบ้าน

โมบายล์แอพพลิเคชันจะสามารถควบคุมหลอดไฟภายในบ้านได้ จะต้องควบคุมผ่าน Bridge โดยจะเชื่อมต่อกับBridge ด้วย Wifi จากนั้น Bridge จะเชื่อมต่อกับหลอดไฟภายในบ้านที่มี ZigBee ติดตั้งอยู่ โดยอาศัยการสื่อสารด้วย ZigBee ผ่านโปรโทคอล Light Link

โมบายล์แอพพลิเคชันสามารถควบคุมการ เปิด/ปิด ปรับความเข้มสี เปลี่ยนสี และ ปรับความสว่างของหลอดไฟได้ รวมทั้งมีความสามารถในการกำหนดเงื่อนไขว่า เมื่อมีการใช้ไฟฟ้าภายในบ้านมีค่าพลังงานในช่วงเวลาหนึ่งเกินค่าที่ตั้งไว้ แล้วจะควบคุมให้ปิด หรือ หรี่หลอดไฟ

5.2 ปัญหาอุปสรรคและแนวทางการแก้ไข

1) การควบคุมเครื่องใช้ไฟฟ้าผ่าน Zigbee นั้นไม่สามารถทำได้โดยตรงจากแอพพลิเคชัน เนื่องจากอุปกรณ์ iOS ไม่มี Zigbee อยู่ภายใน จึงต้องควบคุมเครื่องใช้ไฟฟ้าผ่านตัวกลางที่เรียกว่า Bridge ผ่าน wifi จากนั้น Bridge จะคุยกับเครื่องใช้ไฟฟ้าผ่าน Zigbee อีกต่อนึง
2) การที่จะพัฒนาอุปกรณ์ไฟฟ้าให้ครอบคลุมมากกว่าหลอดไฟนั้น มีข้อจำกัด เนื่องจากแต่ละอุปกรณ์ต้องมี profile Zigbee ที่แตกต่างกัน และต้องมีอุปกรณ์เฉพาะทาง ถึงจะสามารถเชื่อมต่อได้
3) DLMS/COSEM เป็นโปรโทคอลที่มีความซับซ้อนมาก ยากต่อการทำความเข้าใจ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.3 แนวทางในการพัฒนาต่อ

แอพพลิเคชัน Smart Meter and Smart Home สามารถนำไปประยุกต์ใช้งาน และสามารถพัฒนาต่อไปได้ดังนี้

1) ศึกษาแนวทางในการพัฒนา ในการรับข้อมูลจากมิเตอร์ ผ่าน Internet ซึ่งจะทำให้ผู้ใช้สามารถดูปริมาณการใช้งานไฟฟ้าภายในบ้านได้ตลอดเวลา ไม่ว่าผู้ใช้จะอยู่ที่ใดก็ตาม ทำให้สามารถจัดการการใช้พลังงานไฟฟ้าได้อย่างมีประสิทธิภาพมากขึ้น
2) ศึกษาแนวทางในการพัฒนา การควบคุมเครื่องใช้ไฟฟ้าภายในบ้าน ผ่าน Internet ซึ่งจะทำให้ผู้ใช้สามารถสั่ง เปิด/ปิด เครื่องใช้ไฟฟ้าภายในบ้านได้ตลอดเวลา ไม่ว่าผู้ใช้จะอยู่ที่ใดก็ตาม ทำให้สามารถควบคุมเครื่องใช้ไฟฟ้าได้ดีขึ้น เช่น สามารถสั่งเปิดไฟหน้าบ้านตอนกลางคืนเมื่อวันที่ไม่ได้กลับบ้าน เป็นต้น
3) ศึกษาแนวทางในการควบคุมเครื่องใช้ไฟฟ้าทุกชนิดภายในบ้าน ด้วยแอพพลิเคชันนี้ เพียงแอพพลิเคชันเดียว ทำให้สามารถควบคุมทุกสิ่งทุกอย่างได้รวดเร็ว เช่น การสั่งปิดเครื่องใช้ไฟฟ้าภายในบ้านทั้งหมดในปุ่มเดียว ทำให้คำว่า Smart Home สมบูรณ์แบบ
4) พัฒนาส่วนติดต่อผู้ใช้ ให้มีความสวยงามมากยิ่งขึ้น แต่ต้องคงความเรียบง่ายใช้งานง่ายไว้เพื่อให้ผู้ใช้อยากที่จะใช้งานแอพพลิเคชัน
5) ปรับปรุงโครงสร้างของแอพพลิเคชันให้มีความเป็นระบบมากขึ้น สเถียรมากขึ้น และมีความปลอดภัยของข้อมูลมากขึ้น
6) เพิ่มการทำงานจากที่สามารถรับค่าจากมิเตอร์เดียว เป็นหลายๆมิเตอร์ ทำให้สามารถดูค่าจากหลายมิเตอร์พร้อมๆกัน หรือเอามารวมกัน เพื่อวิเคราะห์การใช้พลังงานในระดับภูมิภาคหรือประเทศต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

[1] Energy research Institute. **"What is smart meter ?"** 2012

[2] Toronto Hydro. **"HOW SMART METERS WORK "** [Online]. Available : http://www.torontohydro.com/sites/electricsystem/residential/yourmeter/Pages/HowSmartMetersWork.aspx

[3] Meensika Sripan, Xuanxia Lin, Ponchan Petchlorlean and Mahasak Ketcham . **"Research and Thinking of Smart Home Technology "** 2012

[4] ThaiEasyElec. **"Zigbee and Xbee BASIC ตอน Zigbee คืออะไร"** [Online]. Available: http://www.thaieasyelec.com/electronics-in-chapter/what-is-zigbee.html

[5] ZigBee Alliance. **"ZigBee Light Link"** [Online]. Available: http://www.zigbee.org/Standards/ZigBeeLightLink/Overview.aspx

[6] NXP. **" ZigBee Light Link user guide "** 2013

[7] ZigBee Alliance. **"ZigBee Smart Energy Overview"** [Online]. Available: http://www.zigbee.org/Standards/ZigBeeSmartEnergy/Overview.aspx

[8] WIFI. **"Wireless กับ Wi-Fi คืออะไร"** 2013

[9] N. Calamaro, E. Abramowitz and Y. Beck **"A General Review on DLMS/COSEM and IEC 60870-5 Smart Grid Standards "** 2012

[10] VicHargrave **" Socket "** [Online]. Available: http://vichargrave.com/network-programming-design-patterns-in-c/

[11] Hue system **"How hue works"** [Online]. Available: http://developers.meethue.com/howhueworks.html

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้