โครงงานที่ 19/2562 (วศบ.อุตสาหการ)

การประยุกต์ใช้อุปกรณ์สัญญาณไร้สาย IOT ในการบันทึกข้อมูลดิจิตอลมิเตอร์ไฟฟ้า

นายภัทรเชษฐ์ สุจิรานนท์ รหัสนักศึกษา 590610319
นายศุภวิชญ์ สถิตวาณิช รหัสนักศึกษา 590610345

โครงงานนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
ภาควิชาวิศวกรรมอุตสาหการ
คณะวิศวกรรมศาสตร์ มหาวิทยาลัยเชียงใหม่
ปีการศึกษา 2562

หัวข้อโครงงาน	การประยุกต์ใช้อุปกรณ์สัญญาณไร้สาย IoT ในการบันทึกข้อมูลดิจิตอลมิเตอร์ไฟฟ้า
โดย	นายภัทรเชษฐ์ สุจิรานนท์ รหัสนักศึกษา 590610319
	นายศุภวิชญ์ สถิตวาณิช รหัสนักศึกษา 590610345
ภาควิชา	วิศวกรรมอุตสาหการ คณะวิศวกรรมศาสตร์ มหาวิทยาลัยเชียงใหม่
อาจารย์ที่ปรึกษา	ผศ.ดร.วรพจน์ เสรีรัฐ
ปีการศึกษา	2562

ภาควิชาวิศวกรรมอุตสาหการ คณะวิศวกรรมศาสตร์ มหาวิทยาลัยเชียงใหม่ อนุมัติให้นับโครงงานนี้ เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

กรรมการสอบโครงงาน

..... ประธานกรรมการ
(ผศ.ดร.วรพจน์ เสรีรัฐ)

..... กรรมการ
(รศ.ดร.วัสสนัย วรรธนัจฉริยา)

..... กรรมการ
(ผศ.ดร.อลงกต แก้วโชติช่วงกูล)

กิตติกรรมประกาศ

การทำโครงงานวิจัยเรื่อง การประยุกต์ใช้อุปกรณ์สัญญาณไร้สาย IoT ในการบันทึกข้อมูลดิจิตอลมิเตอร์ไฟฟ้า สามารถดำเนินไปได้ด้วยดี เนื่องจากได้รับความอนุเคราะห์และสนับสนุนจากหลาย ๆ ฝ่ายซึ่งหากไม่มีบุคคลเหล่านี้โครงงานวิจัยนี้อาจไม่ประสบความสำเร็จ

ขอขอบพระคุณ ผศ.ดร.วรพจน์ เสรีรัฐ ซึ่งเป็นอาจารย์ที่ปรึกษาโครงงานที่ได้ให้คำแนะนำความรู้ เสนอแนวทางในการแก้ไขปรับปรุง ตลอดจนคอยให้คำปรึกษาตลอดมา

ขอขอบพระคุณคณาจารย์ภาควิชาวิศวกรรมอุตสาหการ คณะวิศวกรรมศาสตร์ มหาวิทยาลัยเชียงใหม่ ที่ได้ให้ความรู้แก่ผู้วิจัย ตลอดจนบุคลากรทุกท่านที่คอยให้ความช่วยเหลือในการทำโครงงานวิจัยตลอดมา

ขอขอบพระคุณ นายเกียรติวุฒิ ทองหวาน นักศึกษาภาควิชาวิศวกรรมระบบสารสนเทศและเครือข่าย คณะวิศวกรรมศาสตร์ มหาวิทยาลัยเชียงใหม่ ที่ได้ให้ความรู้เรื่องการเขียนโค้ดบนโปรแกรม Arduino IDE ช่วยในการพัฒนาดิจิตอลมิเตอร์ให้สามารถรับ-ส่งข้อมูลได้

ขอขอบพระคุณ ณ โอกาสนี้ด้วย สุดท้ายนี้ทางผู้จัดทำโครงงานวิจัยหวังเป็นอย่างยิ่งว่าโครงงานวิจัยเล่มนี้จะเป็นประโยชน์ต่อผู้ที่สนใจ หากโครงงานวิจัยเล่มนี้บกพร่องหรือผิดพลาดประการใด ทางผู้จัดทำต้องขออภัยและขอน้อมรับข้อเสนอแนะที่เป็นประโยชน์ทุกประการ

ภัทรเชษฐ์ สุจิรานนท์

ศุภวิชญ์ สถิตวาณิช

หัวข้อโครงงาน การประยุกต์ใช้อุปกรณ์สัญญาณไร้สาย IoT ในการบันทึกข้อมูลดิจิตอลมิเตอร์ไฟฟ้า
โดย นายภัทรเชษฐ์ สุจิรานนท์ รหัสนักศึกษา 590610319
นายศุภวิชญ์ สถิตวาณิช รหัสนักศึกษา 590610345
ภาควิชา วิศวกรรมอุตสาหการ คณะวิศวกรรมศาสตร์ มหาวิทยาลัยเชียงใหม่
อาจารย์ที่ปรึกษา ผศ.ดร.วรพจน์ เสรีรัฐ
ปีการศึกษา 2562

บทคัดย่อ

งานวิจัยนี้มุ่งเน้นในการศึกษาหาวิธีการประยุกต์ใช้อุปกรณ์สัญญาณไร้สาย IoT (Internet of Things) ในการบันทึกข้อมูลดิจิตอลมิเตอร์ไฟฟ้า เพื่อให้เกิดความแม่นยำมากขึ้นในขั้นตอนการจดบันทึกหน่วยไฟฟ้าที่ใช้ไปและช่วยลดเวลาในขั้นตอนการจดบันทึกหน่วยไฟฟ้าที่ใช้ไปในแต่ละจุด

โดยในการศึกษานั้นจะทำการหาข้อมูลและข้อจำกัดของโปรแกรมและอุปกรณ์สัญญาณไร้สายต่างๆ หลังจากนั้นทำการวางแผนและออกแบบรูปแบบการทำงาน โดยจะใช้ดิจิตอลมิเตอร์ไฟฟ้าที่รองรับระบบ RS-485 รุ่น DDS238-4 W ซึ่งเป็นมาตรฐานการสื่อสารข้อมูลดิจิตอลแบบอนุกรมซึ่งสามารถนำไปประยุกต์ใช้กับแผงบอร์ดควบคุม Arduino UNO R3 ได้ โดยต้องใช้บอร์ดรองรับระบบ RS-485 ช่วย ควบคู่กับอุปกรณ์รับ-ส่งสัญญาณ XBee รุ่น Pro S2B และ S2C เพื่อให้อุปกรณ์ทั้งสองชิ้นนี้ทำงานร่วมกันได้ จะต้องใช้บอร์ดขยาย Arduino XBee ในการเชื่อมระหว่างอุปกรณ์ทั้งสองชิ้นและใช้ถ่านขนาด 9 V. ในการจ่ายไฟให้กับบอร์ด Arduino เพื่อให้บอร์ด Arduino สามารถทำงานรับ-ส่งข้อมูลหน่วยไฟฟ้าที่ใช้ไปจากดิจิตอลมิเตอร์ไฟฟ้ามายังคอมพิวเตอร์แบบไร้สายในระยะไกลได้

จากผลการทดลองและทดสอบสรุปได้ว่าการเชื่อมต่อแบบไร้สายระหว่างคอมพิวเตอร์กับดิจิตอลมิเตอร์ไฟฟ้า ผ่านอุปกรณ์รับ-ส่งสัญญาณ XBee ซึ่งนำมาประยุกต์ใช้ร่วมกับแผงบอร์ด Arduino สามารถใช้งานได้จริงและเครือข่ายทั้ง 2 แบบ ได้แก่แบบ Point to Point และ แบบ Cluster Tree มีความแม่นยำ 100 เปอร์เซ็นต์ สามารถทำงานรับ-ส่งข้อมูลได้เป็นทอด ๆ แบบเครือข่ายซึ่งมีประสิทธิภาพ ดังนี้ แบบเครือข่าย Point to Point และแบบ Cluster Tree มีประสิทธิภาพการรับ-ส่งข้อมูล 100 เปอร์เซ็นต์ และ 44.5 เปอร์เซ็นต์ ตามลำดับ และมีระยะเฉลี่ยในการรับ-ส่งข้อมูล แบบ Point to Point และแบบ Cluster Tree อยู่ที่ 258.5 เมตรและ 348.5 เมตร ตามลำดับ

Project Title	Application of IoT Wireless Signal Device to Recording Data of Digital Electrical Meter
Name	Pattarachet Sujiranon Code 590610319
	Supavich Satitvanich Code 590610345
Department	Industrial Engineering, Faculty of Engineering, Chiang Mai University
Project Advisor	Assistant Professor Worapod Sereerat, D.Eng.
Academic Year	2019

Abstract

This research focuses on the study of the method of applying an IOT wireless device to record digital electric meter data in order to be more accurate in taking notes of the electric units used and reduce the time required for recording of electric units used at each point.

In this study, it will find information and limitations of programs and various wireless signal devices. After that, make plans and design work patterns. It will use a digital power meter that supports the RS-485 model DDS238-4 W, which is a serial digital data communication standard, which can be applied to the Arduino UNO R3 control panel. The board supports RS-485. Coupled with the XBee Pro S2B and S2C transceiver devices so that both devices can work together Will need an Arduino XBee expansion board to connect the two devices And use a size of 9 V. to supply power to the Arduino board so that the Arduino board can work to send and receive data of the electrical units used from digital electricity meter to wireless computers at a distance.

From the results of the experiment and testing, it can be concluded that the wireless connection between the computer and the digital electricity meter Through the XBee receiver device which is applied together with the Arduino board, it can be used practically and both types of network such as Point to Point and Cluster Tree are 100 percent accurate. It can work to send and receive data as Network spanning, which has the efficiency as follows: Point to Point network type and Cluster Tree type have the efficiency of data transmission - 100 percent and 44.5 percent respectively, and with the average distance Reception - transmission Point to Point and Cluster Tree at 258.5 meters and 348.5 meters respectively.

สารบัญ

หน้า

สารบัญ (ต่อ)

หน้า

สารบัญตาราง

ตาราง หน้า

สารบัญภาพ

สารบัญภาพ (ต่อ)

ภาพ หน้า

สารบัญภาพ (ต่อ)

ภาพ หน้า

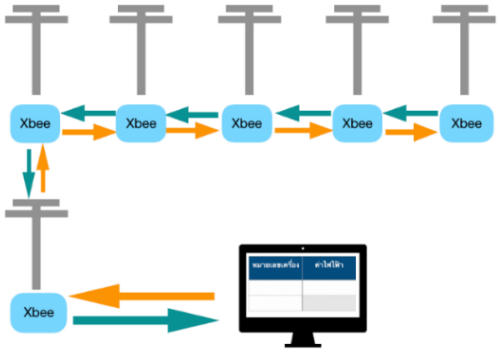
บทที่1

บทนำ

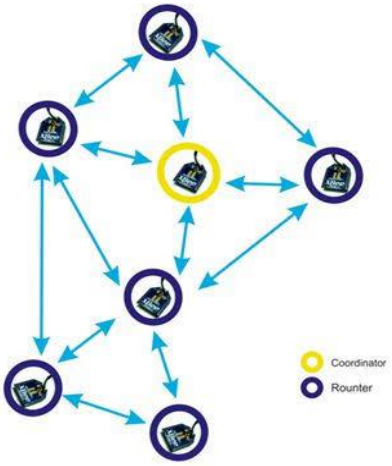
1.1 ความสำคัญ และที่มาของปัญหาที่ทำโครงงาน

ในปัจจุบันนี้การจดบันทึกหน่วยไฟฟ้าที่ใช้ไปนั้นมีความยุ่งยากมากดังภาพ 1.1 โดยผู้ดำเนินการต้องไปที่บริเวณหน้ามิเตอร์เพื่ออ่านค่า ซึ่งในหนึ่งวันนั้นจะสามารถจดบันทึกค่าที่อ่านได้จากมิเตอร์ได้น้อยค่าและ ในกรณีที่เป็นหอพัก คอนโด หรือในบริเวณที่มีมิเตอร์เป็นจำนวนมาก ๆ จะส่งผลให้ใช้เวลาในการจดบันทึกค่าเป็นเวลานานและยังมีโอกาสที่จะเกิดการจดบันทึกค่าผิดพลาดอีกด้วย จึงควรหาแนวทางใหม่ๆ ระบบสื่อสารเครือข่ายแบบไร้สายเป็นระบบที่ได้รับความนิยมมากในปัจจุบันและได้เข้ามาแทนที่ระบบสื่อสารแบบมีสายและได้ถูกนำไปใช้งานอย่างกว้างขวาง ทั้งในหน่วยงานองค์กรต่าง ๆ หรือนำไปใช้ในชีวิตประจำวันเพื่อที่ช่วยอำนวยความสะดวกสบายต่าง ๆ จึงได้มีการพัฒนานวัตกรรมที่เรียกว่า การบันทึกข้อมูลดิจิตอลมิเตอร์แบบไร้สายเพื่อที่จะพัฒนาให้เราสามารถจดบันทึกค่าได้อย่างรวดเร็วและแม่นยำยิ่งขึ้นเพื่อลดขั้นตอนการทำงานของผู้ดำเนินการ เช่น การนำระบบไร้สายมาประยุกต์ใช้กับมิเตอร์ที่เป็นแบบดิจิตอล โดยผ่านตัว XBee ที่เป็นเหมือนตัวรับและตัวส่งสัญญาณเพื่อที่จะเอาค่าที่รับมาแสดงผลบนหน้าจอคอมพิวเตอร์ โดยที่ไม่จำเป็นต้องไปที่หน้ามิเตอร์เพื่ออ่านค่าอีกต่อไปและเมื่อค่ามาถึงที่คอมพิวเตอร์แล้วดังภาพ 1.2 ซึ่ง XBee นั้นสามารถเชื่อมต่อเครือข่ายให้สามารถรับ-ส่งข้อมูลได้หลายรูปแบบดังภาพ 1.3 และ 1.4 ก่อนจะเข้าสู่ขั้นตอนการประมวลผลเพื่อคำนวณหาค่าไฟฟ้าที่ผู้บริโภคต้องชำระ ตามหมายเลขมิเตอร์ของแต่ละที่ แล้วพิมพ์ใบเสร็จส่งไปยังที่อยู่นั้น ๆ ซึ่งจะสามารถช่วยลดข้อจำกัดต่าง ๆ ได้อีกมากมายดังตาราง 1.1

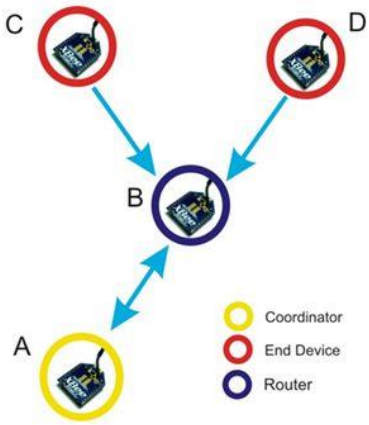
ภาพ 1.1 ผู้จดค่าไฟฟ้าที่หน้าเครื่องมิเตอร์ไฟฟ้า



ภาพ 1.2 การรับส่งสัญญาณของ XBee



ภาพ 1.3 XBee แบบ Mesh Network



ภาพ 1.4 XBee แบบ Cluster Tree

ตาราง 1.1 ข้อดี-ข้อเสีย เมื่อนำระบบไร้สายมาแทนระบบเดิม

ข้อดี	ข้อเสีย
1. ใช้คนในการทำงานน้อยลง	1. เสียค่าจัดส่งใบแจ้งชำระค่าไฟ
2. มีความแม่นยำมากขึ้น	2. มีค่าใช้จ่ายในการเปลี่ยนอุปกรณ์ระบบใหม่
3. ลดปัญหาการอ่านค่ามิเตอร์ในที่สูง	
4. ประหยัดเวลาและค่าใช้จ่ายในกระบวนการทำงาน	

1.2 วัตถุประสงค์

1.2.1 เพื่อให้เกิดความแม่นยำมากขึ้นในขั้นตอนการจดบันทึกหน่วยไฟฟ้าที่ใช้ไป

1.2.2 เพื่อลดเวลาในขั้นตอนการจดบันทึกหน่วยไฟฟ้าที่ใช้ไปในแต่ละจุด

1.3 ขอบเขตการศึกษา

1.3.1 สถานที่ศึกษา ภาควิชาวิศวกรรมอุตสาหการ คณะวิศวกรรมศาสตร์ มหาวิทยาลัยเชียงใหม่

1.4 ประโยชน์ที่คาดว่าจะได้รับ

1.4.1 สามารถลดขั้นตอนในการบันทึกหน่วยไฟฟ้าที่ใช้ไปได้อย่างรวดเร็วมากขึ้นและลดต้นทุนในการดำเนินงานในแต่ละครั้ง

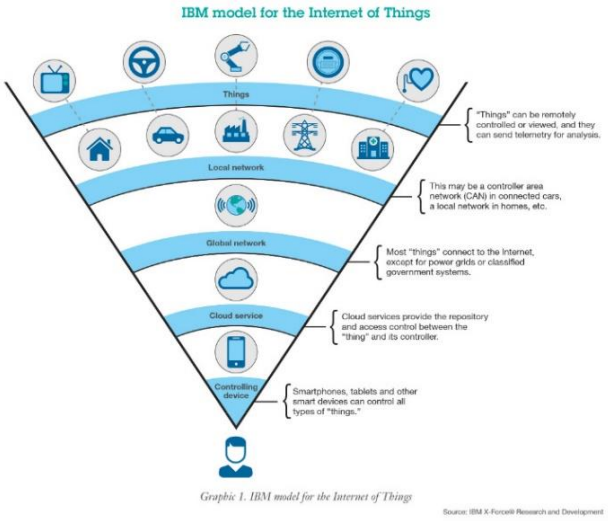
1.4.2 สามารถรับค่าได้ทีละหลาย ๆ เครื่องมิเตอร์พร้อมกันและมีความแม่นยำกว่าระบบเดิมที่ใช้คนในการจดบันทึกค่า

บทที่ 2

หลักการและทฤษฎีที่เกี่ยวข้อง

2.1 ความหมายของ IoT

อินเตอร์เน็ตในทุกสิ่ง (IoT) หมายถึง เทคโนโลยีอินเทอร์เน็ตที่เชื่อมต่ออุปกรณ์และเครื่องมือต่างๆ เข้าไว้ด้วยกัน โดยสามารถเชื่อมโยงและสื่อสารกันได้โดยผ่านระบบอินเทอร์เน็ตสามารถสั่งการการทำงานของโปรแกรมหรือสั่งการควบคุมการใช้งานอุปกรณ์ผ่านทางเครือข่ายอินเทอร์เน็ต ดังภาพ 2.1 ปัจจุบันมีการพัฒนาอุปกรณ์ให้สามารถทำงานบนแนวคิดของอินเตอร์เน็ตในทุกสิ่ง (IoT) เพิ่มมากขึ้น โดยนำมาประยุกต์ใช้งานมากขึ้น เช่น การประยุกต์ใช้งานในเกษตรอัจฉริยะ (Smart Farm) คือ การนำเทคโนโลยีที่มีอยู่มาประยุกต์ใช้งานกับระบบโรงเรือนหรือในแปลงปลูกให้มีผลผลิตและคุณภาพที่ดีขึ้น



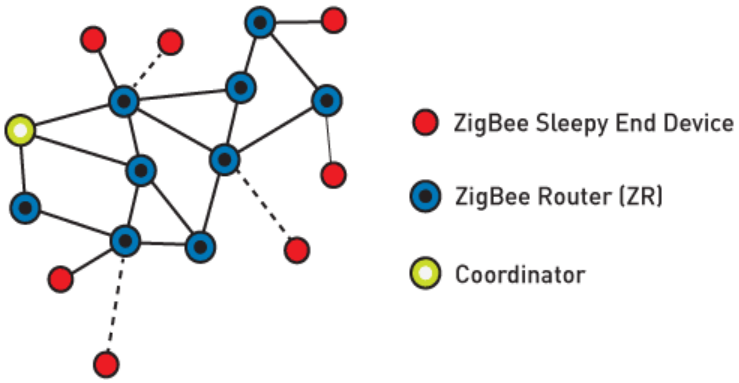
ภาพ 2.1 แสดงถึงความหลากหลายในการประยุกต์ใช้ IoT

2.2 ระบบเครือข่ายเซนเซอร์ไร้สาย (WSN)

มีการใช้งานอย่างแพร่หลายในปัจจุบันมีการนำไปประยุกต์ใช้งานในหลายๆด้าน เช่น ด้านสุขภาพด้านการทหาร ด้านการเกษตรกรรม เครือข่ายเซนเซอร์ไร้สายประกอบด้วยอุปกรณ์ที่ทำหน้าที่เป็นโหนดเซนเซอร์ (Sensor Node) และโหนดสถานีฐาน (Base Station Node) โหนดเซนเซอร์ทำหน้าที่ในการส่งข้อมูลที่วัดจากเซนเซอร์ไปยังโหนดสถานีฐานผ่านทางคลื่นวิทยุ ส่วนโหนดสถานีฐานจะทำหน้าที่ในการติดต่อสื่อสารระหว่างเครือข่ายเซนเซอร์ไร้สายกับ คอมพิวเตอร์ผ่านทางพอร์ตอนุกรม (Serial Protocol) โดยการติดต่อสื่อสารในเครือข่ายไร้สายสามารถติดต่อกันได้ทั้งรูปแบบ Single-Hop และรูปแบบ Multihop

2.3 XBee

เป็นโมดูลสื่อสารไร้สายความถี่ 2.4 เฮิรตซ์ ที่มีความสามารถเป็นไมโครคอนโทรลเลอร์ในตัว สามารถเชื่อมต่อกันเป็นเครือข่ายโดยใช้โปรโตคอลสื่อสาร Zigbee (หรืออาจจะเป็นโปรโตคอลอื่นๆก็ได้) แต่การที่จะใช้งานเป็นเครือข่ายนั้นต้องมีการกำหนดหน้าที่การทำงานของแต่ละโมดูลเพื่อให้ทำงานสอดคล้องกัน โดยการกำหนดหน้าที่การทำงานนั้นจะขึ้นอยู่กับความเหมาะสมของการนำไปใช้ ดังภาพ 2.2



ภาพ 2.2 แสดงให้เห็นหน้าที่ต่างๆของ Zigbee

2.3.1 Coordinator

- เป็นผู้เริ่มต้นและจัดการเครือข่ายภายในหนึ่งเครือข่ายจะมี Coordinator เพียงตัวเดียวเสมอ
- Coordinator ไม่สามารถเข้าสู่โหมด Sleep Mode ได้ จะต้องพร้อมทำงานอยู่เสมอ

- กำหนด Channel และ PAN ID ของเครือข่ายอนุญาตให้ Router และ End Device สามารถเข้ามา Join เครือข่ายได้ อีกทั้งยังเป็นตัวจัดหาเส้นทางถ่ายโอนข้อมูลและสามารถเป็นที่พักข้อมูลที่รับส่งกันภายในเครือข่าย สำหรับกรณีที่มีโหนดลูกข่ายใดๆ อยู่ใน Sleep Mode

2.3.2 Router

- รับ/ส่ง และถ่ายทอดข้อมูลจากที่อยู่ต้นทางไปปลายทางได้
- อนุญาตให้ Router อื่นๆ และ End Device เข้า Join กับตัวเองได้
- สามารถทำหน้าที่พักข้อมูลให้กับโหนดลูกข่ายใดๆที่อยู่ใน Sleep Mode ได้

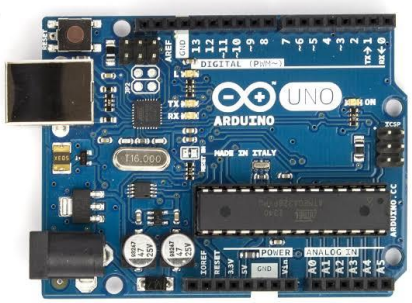
2.3.3 End Device

- ทำหน้าที่เป็นลูกข่ายสามารถเข้าร่วมกับเครือข่ายใดๆ ที่มีอยู่ได้ แต่ไม่อนุญาตให้โมดูลอื่นๆ เข้า มา Join เป็นลูกข่ายของตัวเองได้
- รับ/ส่ง ข้อมูลได้ แต่ไม่สามารถเป็นตัวถ่ายทอดข้อมูลไปยังโหนดอื่นๆได้
- สามารถใช้งาน Sleep Mode เพื่อประหยัดพลังงานได้

2.4 Arduino

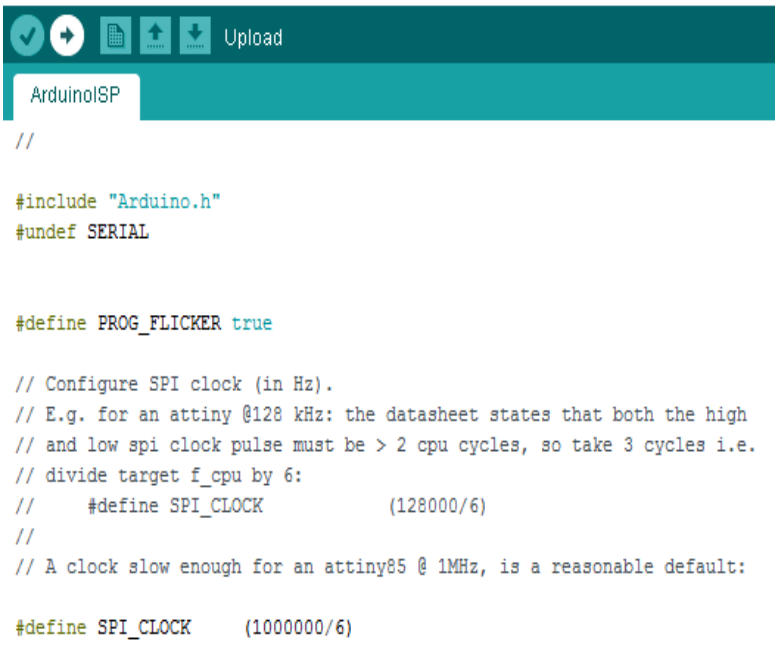
เป็นแพลตฟอร์มต้นแบบด้านอิเล็กทรอนิกส์แบบโอเพ่นซอร์ส ซึ่งใช้ฮาร์ดแวร์และซอฟท์แวร์ที่ยืดหยุ่นและใช้งานง่าย มีไว้สำหรับศิลปินนักออกแบบงานอดิเรกและทุกคนที่สนใจในการสร้างวัตถุเชิงโต้ตอบหรือสภาพแวดล้อม การเขียนโค้ดโปรแกรมควบคุมการทำงานของ Arduino มีความง่ายและยืดหยุ่นสามารถใช้งานในระดับสูงได้อีกด้วย เครื่องมือที่ใช้สำหรับเขียนโค้ดควบคุมมีเวอร์ชั่นที่สามารถรันได้ในทุกระบบปฏิบัติการ เช่น ระบบแมคอินทอช วินโดวส์ ลีนุกซ์ ทำให้ได้รับความนิยมเป็นอย่างสูง แพลตฟอร์ม Arduino ประกอบไปด้วย ฮาร์ดแวร์ และซอฟต์แวร์

- ฮาร์ดแวร์ (Hardware) เป็นบอร์ดอิเล็กทรอนิกส์ขนาดเล็กที่มีไมโครคอนโทรลเลอร์เป็นชิ้นส่วนหลักประกอบร่วมกับอุปกรณ์อิเล็กทรอนิกส์อื่นๆ ดังภาพ 2.3 โดยในแต่ละรุ่นอาจมีความแตกต่างกันในเรื่องของขนาดของบอร์ดหรือคุณสมบัติ เช่น จำนวนของขารับ-ส่งสัญญาณ แรงดันไฟที่ใช้ ประสิทธิภาพของ MCU



ภาพ 2.3 บอร์ด Arduino รุ่น UNO

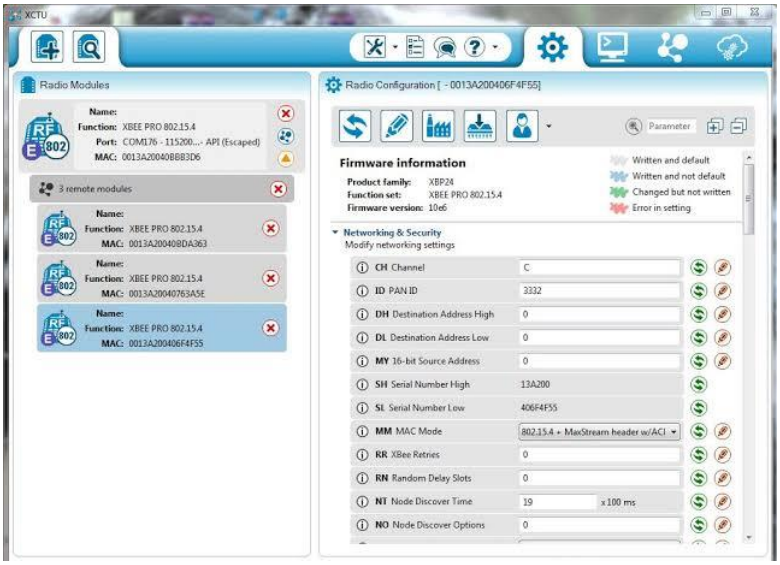
- ซอฟต์แวร์ (Software) ภาษาที่ใช้เขียนโค้ด ควบคุมบอร์ด Arduino เป็นภาษาสำหรับเขียนโปรแกรมควบคุมที่มีไวยากรณ์ แบบเดียวกับภาษา C/C++ และ Arduino IDE เป็นเครื่องมือสำหรับเขียนโค้ดโปรแกรม การคอมไพล์โปรแกรม (การแปลงไฟล์ ภาษาซีให้เป็นภาษาเครื่อง) และอัปโหลดโปรแกรมลงบอร์ดดังภาพ 2.4



ภาพ 2.4 หน้าต่างของโปรแกรม Arduino

2.5 โปรแกรม X-CTU

X-CTU เป็นการติดต่อประสานระหว่างเครื่องคอมพิวเตอร์กับผู้ใช้ (Software Interface) บนคอมพิวเตอร์ที่จะช่วยในการอัพเดท Firmware หรือทดสอบการใช้งานหรือปรับค่าตัวแปรต่าง ๆ กับ XBee ซึ่งมีหน้าต่างโปรแกรม ดังภาพ 2.5

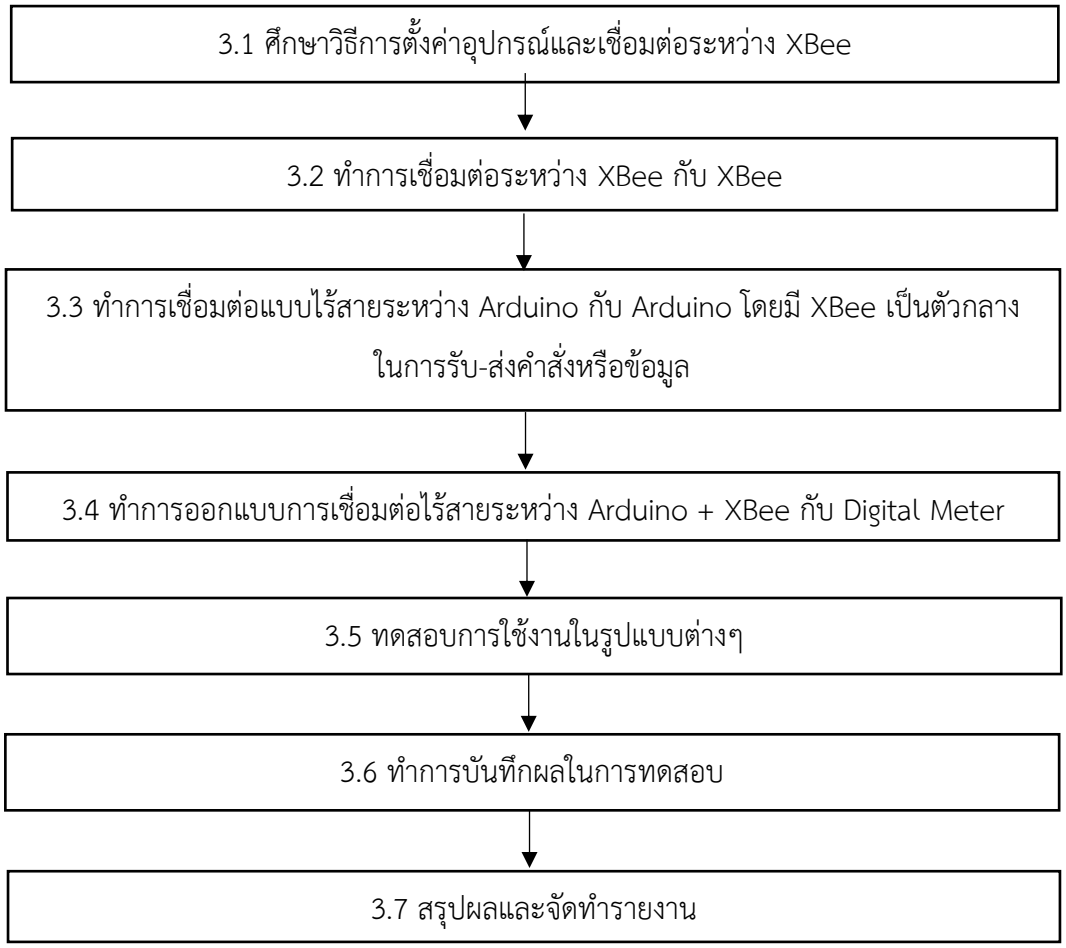


ภาพ 2.5 โปรแกรม X-CTU

บทที่ 3

ระเบียบวิธีการทำวิจัย

ในการศึกษาการประยุกต์ใช้อุปกรณ์สัญญาณไร้สาย IOT ในการบันทึกข้อมูลดิจิตอลมิเตอร์ไฟฟ้า ผู้วิจัยเล็งเห็นว่าวิธีการดังกล่าวมีประโยชน์ในการนำมาประยุกต์กับการจดบันทึกหน่วยไฟฟ้าที่ใช้ไปบนมาตรวัดมิเตอร์ที่ต้องใช้เจ้าหน้าที่ในการดำเนินการเป็นหลัก ซึ่งส่งผลให้มีต้นทุนที่สูงและเกิดการเคลื่อนที่มากจนเกินไป การดำเนินงานวิจัย การประยุกต์ใช้อุปกรณ์สัญญาณไร้สาย IOT ในการบันทึกข้อมูลดิจิตอลมิเตอร์ไฟฟ้า มีขั้นตอนดังภาพ 3.1



ภาพ 3.1 ขั้นตอนการดำเนินการวิจัย

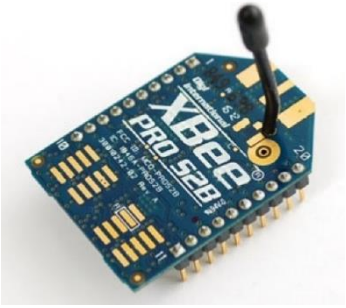
3.1 ศึกษาวิธีการตั้งค่าอุปกรณ์และเชื่อมต่อระหว่าง XBee

3.1.1. การ Set ค่า Firmware ให้ตรงกันกับโปรแกรม X-CTU สำหรับ XBee นั้น ในแต่ละรุ่น จะมี Firmware เฉพาะรุ่น ซึ่งได้โปรแกรมมาแล้วจากทางโรงงาน ซึ่งการเลือก Firmware ให้ถูกต้องจะส่งผลให้อุปกรณ์ของเราทำงานได้อย่างเต็มประสิทธิภาพ ซึ่งสามารถดู Firmware ได้จากด้านหลังของตัวอุปกรณ์ XBee ดังภาพ 3.2 และ 3.3

ภาพ 3.2 Firmware รุ่นต่างๆ ของ XBee

ภาพ 3.3 รหัส Firmware ของ XBee

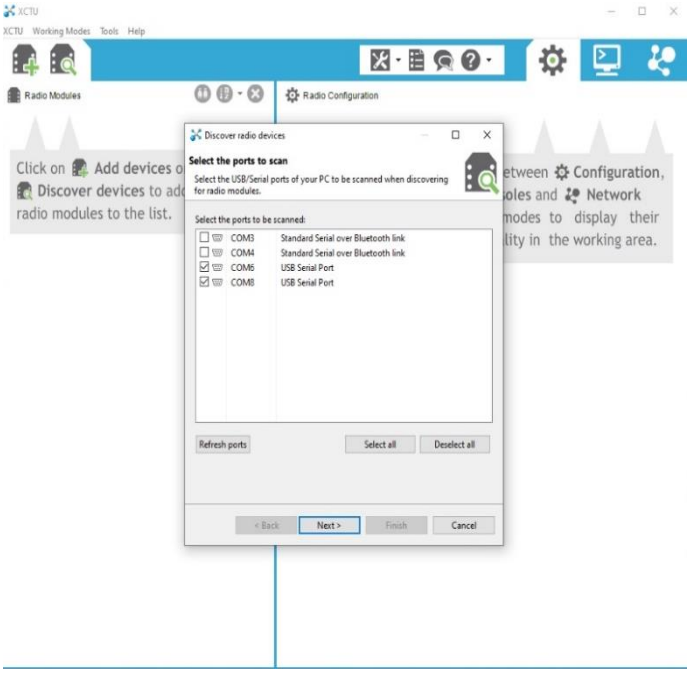
การตั้งค่าจะใช้อุปกรณ์ Mini XBee USB Dongle V2 สำหรับเสียบอุปกรณ์ XBee กับคอมพิวเตอร์ (การตั้งค่าจะต้องใช้อุปกรณ์เป็นคู่ๆ ดังนั้นต้องใช้ XBee Pro S2B และ Mini XBee USB Dongle V2 ทั้งหมด 2 ชุด) ดังภาพ 3.4 และ 3.5 หลังจากนั้นทำการเปิดโปรแกรม X-CTU 2 หน้าต่าง แล้วไปที่หน้า Tab Modem Configuration เพื่อเลือก Firmware ให้ตรงกับรุ่นอุปกรณ์ที่ใช้ แล้วกด Read จะพบว่า X-CTU จะทำการ Load Firmware ของ XBee Pro S2B ออกมาซึ่งจะเป็นค่าที่ Set ไว้แบบ Default



ภาพ 3.4 XBee **ภาพ 3.5 Mini XBee USB Dongle V 2**

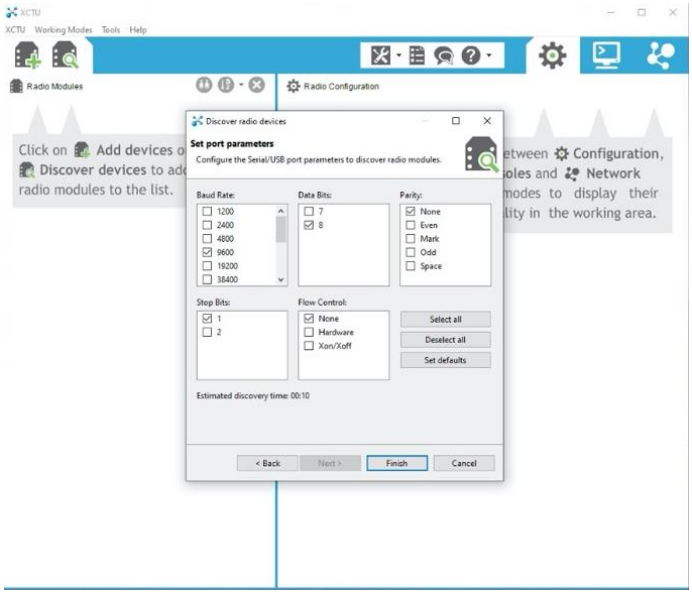
3.1.2 ขั้นตอนการตั้งค่า XBee ด้วย X-CTU มีขั้นตอน 4 ขั้นตอนดังนี้

1. เปิดโปรแกรม X-CTU ใน PC Settings Tab หน้าต่าง Select Com Port จะแสดง Port ต่างๆที่เชื่อมต่ออยู่กับคอมพิวเตอร์ ในที่นี้จะเลือก Port COM 6 และ COM 8 ที่เป็น Port ที่เชื่อมต่ออยู่กับ XBee ดังภาพ 3.6 แล้วกดปุ่ม Next



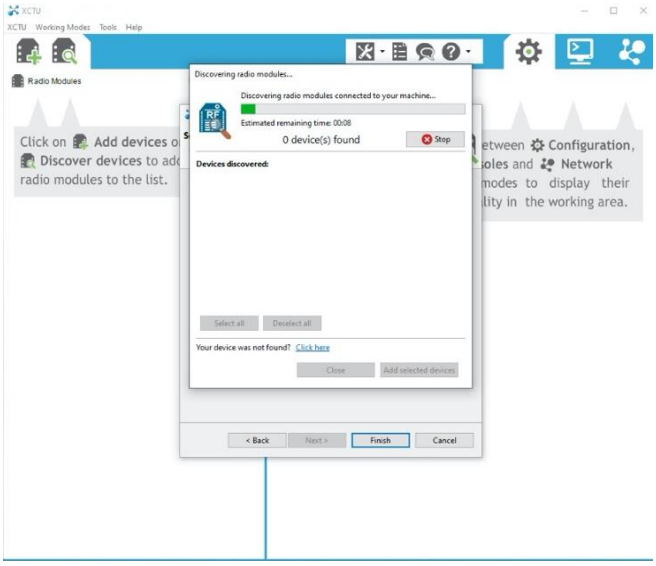
ภาพ 3.6 วิธีสแกนหา XBee (ก)

2. หลังจากนั้นกดปุ่ม Finish ดังภาพ 3.7



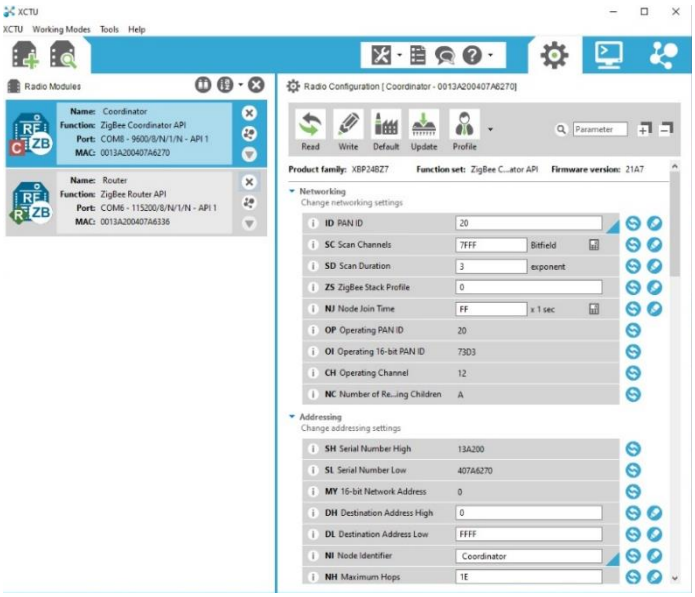
ภาพ 3.7 วิธีสแกนหา XBee (ข)

3. รอโปรแกรม ตรวจหา XBee ดังภาพ 3.8



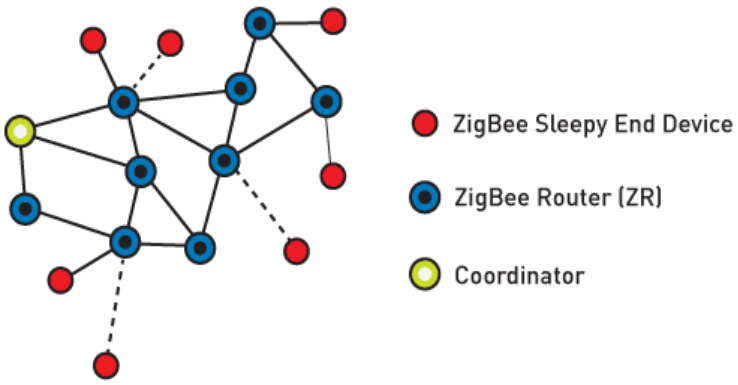
ภาพ 3.8 วิธีสแกนหา XBee (ค)

4. ต่อมาทำการ Read เพื่อบันทึกค่าล่าสุดที่ได้ทำการตั้งค่าและรอจนจบกระบวนการ จะปรากฏค่าต่าง ๆ ที่จะใช้แก้ไข XBee Module ดังภาพ 3.9



ภาพ 3.9 วิธีสแกนหา XBee (ง)

3.1.3 กำหนดหน้าที่การทำงาน XBee ให้เหมาะกับการใช้งาน ดังภาพ 3.10



ภาพ 3.10 หน้าที่การทำงานของ XBee

XBee สามารถทำหน้าที่ได้ 3 แบบ ได้แก่ 1. แบบ Coordinator 2. แบบ Router 3. แบบ End Device ซึ่งควรเลือกหน้าที่การทำงานให้เหมาะกับการนำไปใช้งาน ซึ่งมีรายละเอียดดังนี้

แบบที่ 1 คุณสมบัติของหน้าที่ Coordinator

- เป็นผู้เริ่มต้นและจัดการเครือข่ายภายในหนึ่งเครือข่ายจะมี Coordinator เพียงตัวเดียวเสมอ

- Coordinator ไม่สามารถเข้าสู่โหมด Sleep Mode ได้ จะต้องพร้อมทำงานเสมอ

- กำหนด Channel และ PAN ID ของเครือข่ายอนุญาตให้ Router และ End Device สามารถเข้ามา Join เครือข่ายได้ อีกทั้งยังเป็นตัวจัดหาเส้นทางถ่ายโอนข้อมูลและสามารถเป็นที่พักข้อมูลที่รับส่งกันภายในเครือข่าย สำหรับกรณีที่มีโหนดลูกข่ายใด ๆ อยู่ใน Sleep Mode

แบบที่ 2 คุณสมบัติของหน้าที่ Router

- รับ/ส่ง และถ่ายทอดข้อมูลจากที่อยู่ต้นทางไปปลายทางได้

- อนุญาตให้ Router อื่นๆ และ End Device เข้า Join กับตัวเองได้

- สามารถทำหน้าที่พักข้อมูลให้กับโหนดลูกข่ายใดๆที่อยู่ใน Sleep Mode ได้

แบบที่ 3 คุณสมบัติของหน้าที่ End Device

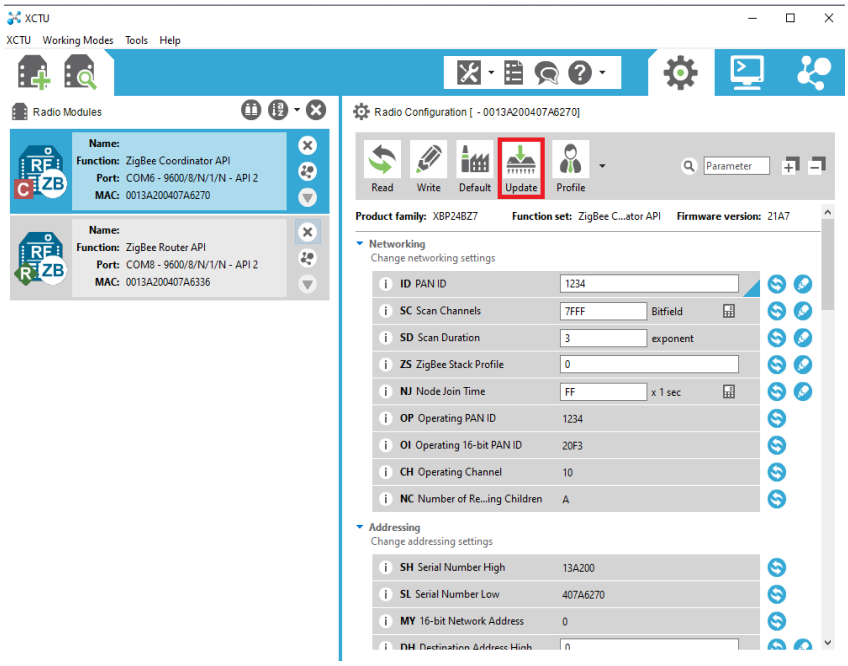
- ทำหน้าที่เป็นลูกข่ายสามารถเข้าร่วมกับเครือข่ายใด ๆ ที่มีอยู่ได้ แต่ไม่อนุญาตให้โมดูลอื่นๆ เข้า มา Join เป็นลูกข่ายของตัวเองได้

- รับ/ส่ง ข้อมูลได้ แต่ไม่สามารถเป็นตัวถ่ายทอดข้อมูลไปยังโหนดอื่น ๆ ได้

- สามารถใช้งาน Sleep Mode เพื่อประหยัดพลังงานได้

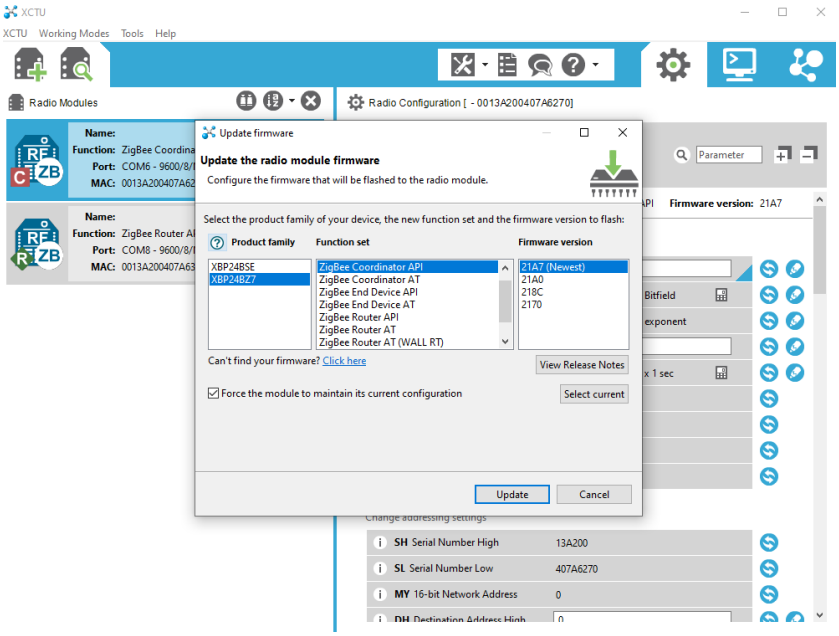
มีขั้นตอนในการตั้งค่าดังนี้

1.เข้าที่ Update เพื่อทำการเลือกหน้าที่ของ XBee แต่ละตัว ดังภาพ 3.11



ภาพ 3.11 วิธีตั้งค่าหน้าที่ของ XBee (ก)

2. เลือกหน้าที่ที่ต้องการให้ XBee ตัวนั้น ๆ และกดปุ่ม Update ดังภาพ 3.12



ภาพ 3.12 วิธีตั้งค่าหน้าที่ของ XBee (ข)

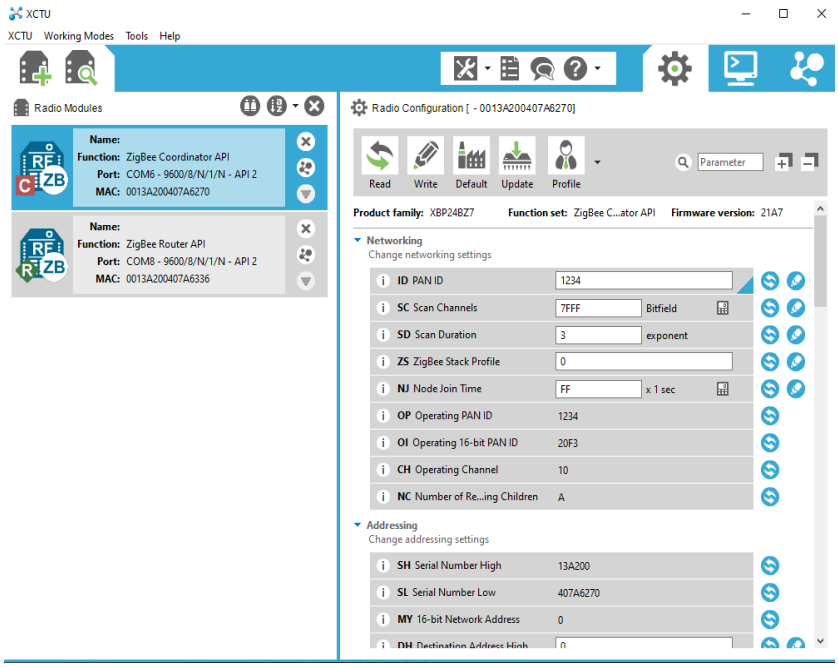
จากภาพ 3.12 จะสังเกตได้ว่ามีให้เลือกอยู่ 2 โหมด คือโหมด API และโหมด AT ซึ่งทั้ง 2 โหมดมีรายละเอียด ดังนี้

1. โหมด AT (Application Transparent Mode) เป็นโหมดการทำงานที่ตัวโมดูลจะทำการส่งผ่านข้อมูลที่ได้รับไปยังที่อยู่ปลายทาง ตัวอย่างเช่น ถ้าหากตัวไมโครคอนโทรลเลอร์ ส่งข้อความผ่าน Serial Interface ไปยังโมดูล XBee โมดูล A ว่า "Hello" ตัวโมดูล A ก็จะรับข้อความนั้น ๆ และทำการส่งต่อไปยังโมดูล B ที่อยู่ระยะห่างออกไป ฝั่งโมดูล B เมื่อได้รับข้อความ ก็จะทำการส่งข้อความนั้น ๆ ผ่านออกไปทาง Serial Interface ไปยังอุปกรณ์ต่อพวงที่ต่ออยู่กับตัวมันเอง พูดอีกอย่างหนึ่งคือ โมดูล XBee ทำหน้าที่เป็นเหมือนสายส่งข้อมูล Serial Interface นั่นเอง

2. โหมด API (Application Programming Interface) เป็นโหมดการสื่อสารที่ข้อมูลที่รับ-ส่งกันจะถูกจัดแจงให้อยู่ในรูปของกลุ่มข้อมูล (Packet) ข้อมูลที่มีการกำหนดค่าต่าง ๆ วิธีนี้จะช่วยทำให้ผู้ใช้งานสามารถรับ-ส่งข้อมูลกันระหว่างโมดูลได้ในรูปแบบที่ซับซ้อนมากยิ่งขึ้น ตัวอย่างเช่น เมื่อฝั่งส่ง A ต้องการส่งข้อมูลไปยังฝั่งรับ B ฝั่งส่งจะทำการแปลงข้อมูลที่ต้องการส่งให้อยู่ในรูปของ Packet ข้อมูล ซึ่งจะประกอบไปด้วย ที่อยู่ผู้รับ ที่อยู่ผู้ส่ง ประเภทของข้อมูลที่ต้องการนำส่ง ข้อความที่ต้องการนำส่ง บิทตรวจสอบความถูกต้อง เป็นต้น เมื่อกลุ่มข้อมูล (Packet) ข้อมูลถูกประกอบขึ้นเรียบร้อย ตัวโมดูล A จะทำการส่งข้อมูลไปยังโมดูล B เมื่อโมดูล B ได้รับ ก็จะส่งข้อมูลไปยังอุปกรณ์ต่อพ่วง ซึ่งอาจจะเป็นคอมพิวเตอร์ หรือบอร์ดไมโครคอนโทรลเลอร์ใด ๆ จากนั้นก็จะเข้าสู่กระบวนการถอด Packet ข้อมูลออก นำข้อความจริง ๆ ที่ต้องการไปใช้งาน

ซึ่งในการทดลองในครั้งนี้จะเลือกใช้โหมด API เนื่องจากเป็นระบบที่รับ-ส่งข้อมูลหนึ่งครั้ง สามารถรับส่งข้อมูลไปยังปลายทางเดียวหรือหลาย ๆ ปลายทางได้ และกลุ่มข้อมูล (Packet) ข้อมูลที่ได้รับสามารถบ่งบอกได้ว่าส่งมาจากใคร และยังตรวจสอบได้ว่าการรับส่งล้มเหลวหรือสมบูรณ์ ซึ่งแบบ AT ทำไม่ได้

3. จะปรากฏหน้าต่างที่ใช้ในการตั้งค่าการเชื่อมต่อแบบเครือข่าย ดังภาพ 3.13

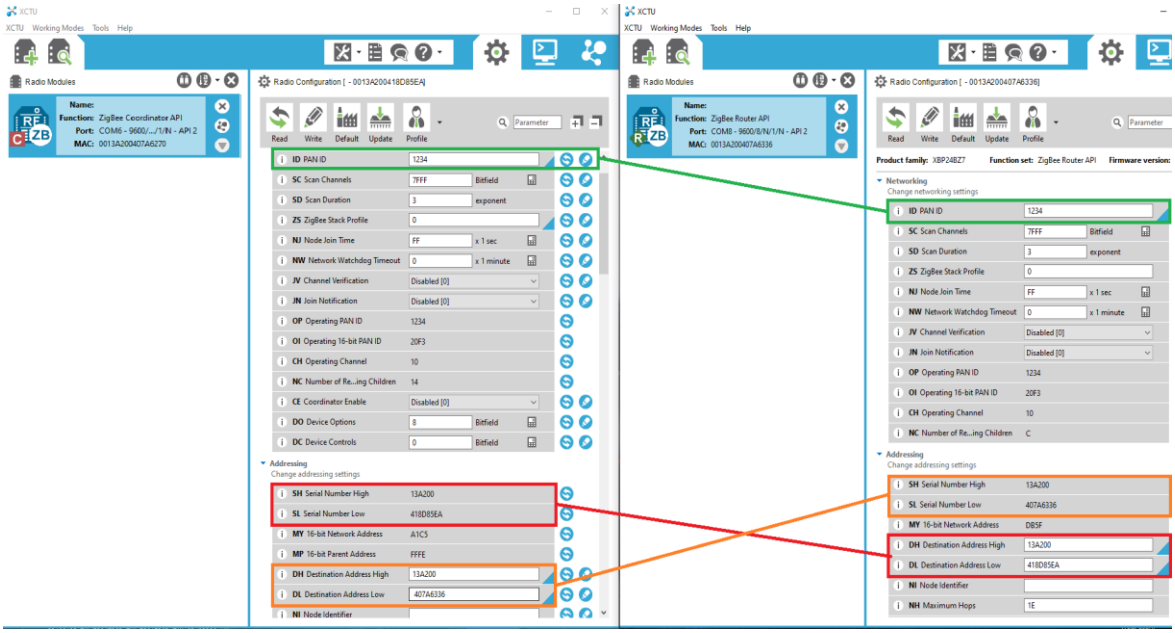


ภาพ 3.13 หน้าต่างที่ใช้ในการตั้งค่าเครือข่าย XBee

ในการเริ่มทำการทดสอบนั้นจะเริ่มจากการทดสอบการรับ-ส่งคำสั่งหรือข้อมูลสำหรับเปิด-ปิดไฟ จะใช้เครือข่ายแบบ Point to Point ซึ่งเป็นการเชื่อมต่อแบบตัวต่อตัว โดยกำหนดให้ตัวแรกเป็น Coordinator ส่วนอีกตัวกำหนดเป็น Router ดังภาพ 3.14

ภาพ 3.14 เครือข่ายแบบ Point to Point

3.1.4 การตั้งค่า Point-to-point (จุด-จุด) โดยค่า ID เหมือนกันถ้าอยู่ในเครือข่ายเดียวกัน ในที่นี้ กำหนดให้ PAN ID มีค่าเท่ากับ 1234 และ ค่า DH DL ใช้กำหนดอุปกรณ์ปลายทางที่ต้องการติดต่อ ให้กำหนดค่า DH DL ของ Node หนึ่ง (ของตัว Router) ให้เหมือนกับ SH SL ของอีก Node หนึ่ง (ของตัว Coordinator) ดังภาพ 3.15

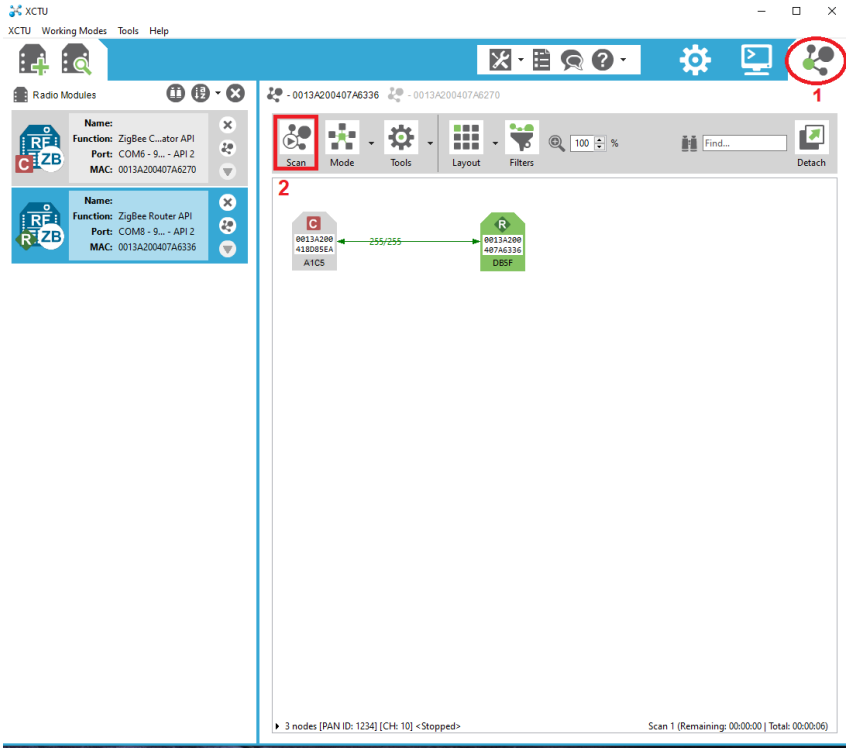


ภาพ 3.15 การกำหนดค่า PAN ID SH SL DH DL

ซึ่งค่า DH DL ที่จะนำไปใช้นั้นสามารถดูได้จากด้านหลังของอุปกรณ์ XBee ดังภาพ 3.16

ภาพ 3.16 ค่า DH DL ด้านหลัง XBee

3.1.5 ทำการตรวจสอบว่า XBee ทั้งสองตัว ตั้งค่าถูกต้องหรือไม่ ถ้าถูกต้องจะสามารถจับคู่กันได้ ซึ่งทราบได้จากการใช้ Network woking mode (หมายเลข 1) และทำการ Scan (หมายเลข 2) ดังภาพ 3.17 แล้วรอซักครู่



ภาพ 3.17 ตรวจสอบการจับคู่ของ XBee

จากภาพ 3.17 จะเห็นได้ว่าเมื่อทำการ Scan แล้ว จะปรากฏ XBee ขึ้นมาทั้ง 2 ตัว ทั้งตัวที่ทำหน้าที่เป็น Coordinator และ ตัวที่ทำหน้าที่เป็น Router และมีลูกศรสองหัวเชื่อมอยู่ระหว่างกลาง ซึ่งหมายถึงการที่ XBee ทั้งสองตัวนี้พร้อมที่จะทำงานร่วมกันแล้ว และพร้อมสำหรับดำเนินการในขั้นตอนต่อไป

3.2 ทำการเชื่อมต่อระหว่าง XBee

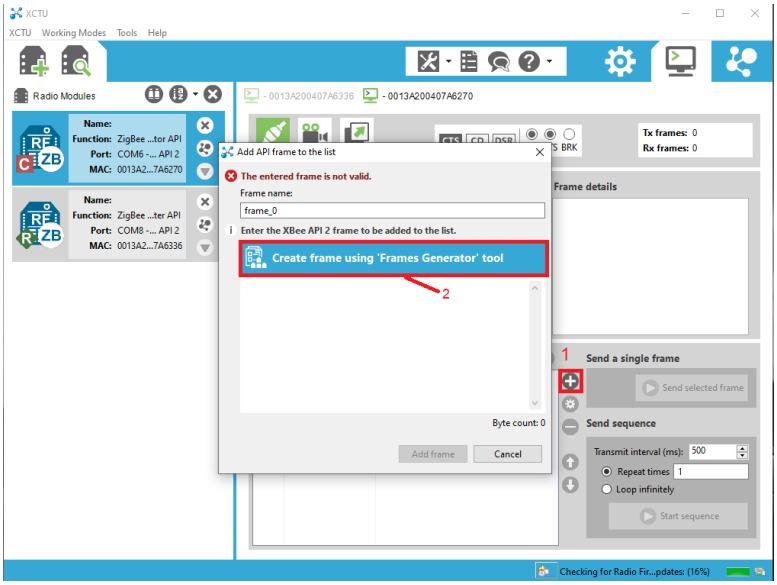
ในขั้นตอนนี้จะใช้โปรแกรม X-CTU โหมด Console Working Mode ในการทดสอบว่า XBee แต่ละตัวสามารถรับ-ส่งข้อมูลกันได้หรือไม่ ซึ่งมีขั้นตอน 5 ขั้นตอนดังนี้

3.2.1 ทำการสแกนหา XBee ทั้ง 2 ตัว

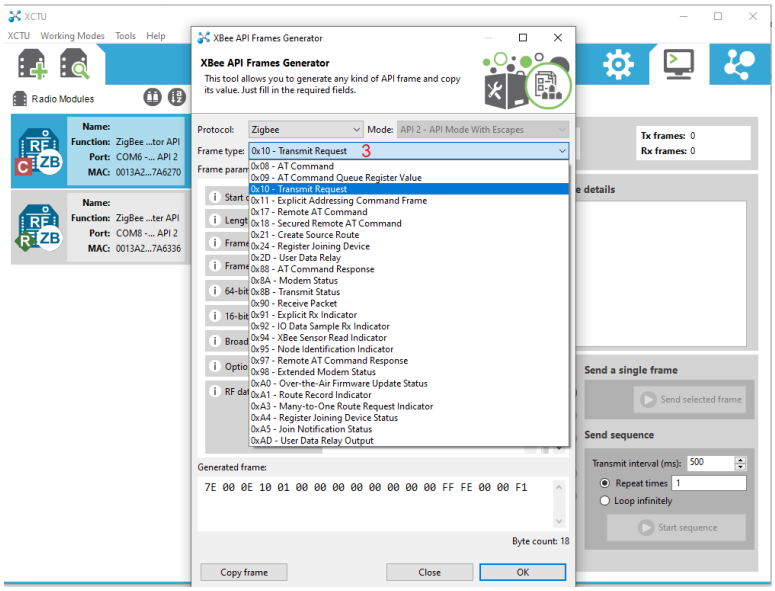
3.2.2 เข้าที่ Console Working Mode 

3.2.3 กดปุ่ม Connect เพื่อให้ XBee ทั้ง 2 ตัว พร้อมทำงาน 

3.2.4 ทำการ Copy MAC Address ของตัวที่ต้องการรับ-ส่งข้อมูล นำไปใส่ใน Address ปลายทางตามขั้นตอน ดังภาพ 3.18 3.19 3.20 และ 3.21

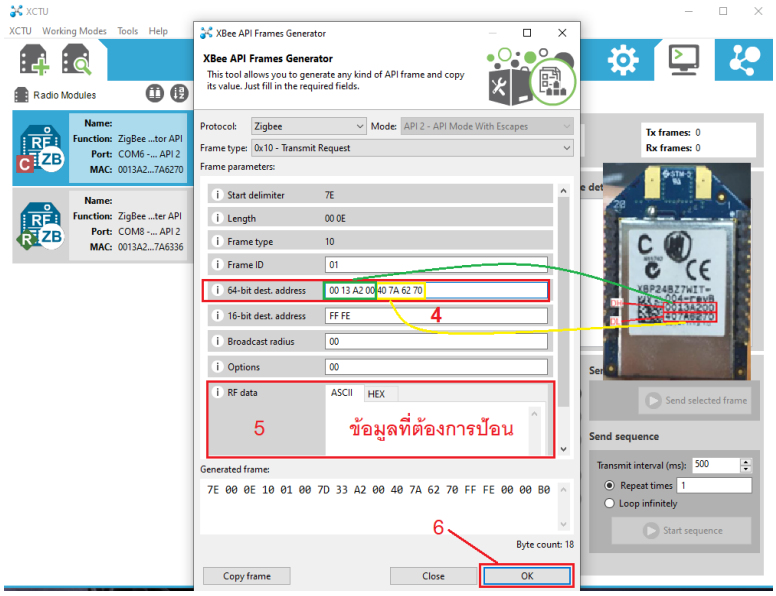


ภาพ 3.18 ทำการใส่ Adress ปลายทาง (ก)



ภาพ 3.19 ทำการใส่ Adress ปลายทาง (ข)

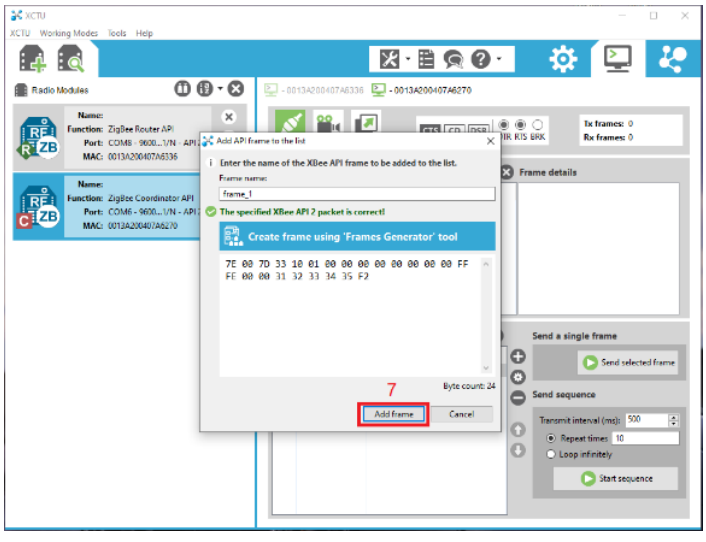
จากภาพ 3.19 จะใช้ Transmit Request เพื่อใช้ในการส่งคำขอข้อมูลจาก XBee ที่ทำหน้าที่เป็น Coordinator ไปยัง XBee ที่ทำหน้าที่เป็น Router และให้ส่งข้อมูลกลับมาที่ XBee ที่ทำหน้าที่เป็น Coordinator ซึ่งเหมือนกันกับตอนที่นำไปใช้งานจริง



ภาพ 3.20 ทำการใส่ Adress ปลายทาง (ค)

จากภาพ 3.20 ในการใส่ Adress นั้นจะต้องใช้ค่าที่อยู่ด้านหลังของ XBee ตัวที่ต้องการจะส่งข้อมูลไป โดยกรอกค่า Adress ลงในช่อง 64-bit dest. Address ซึ่งค่า 8 ตัวแรกนั้นจะเป็นค่า DH และ ค่า 8 ตัวหลังนั้นจะเป็นค่า DL

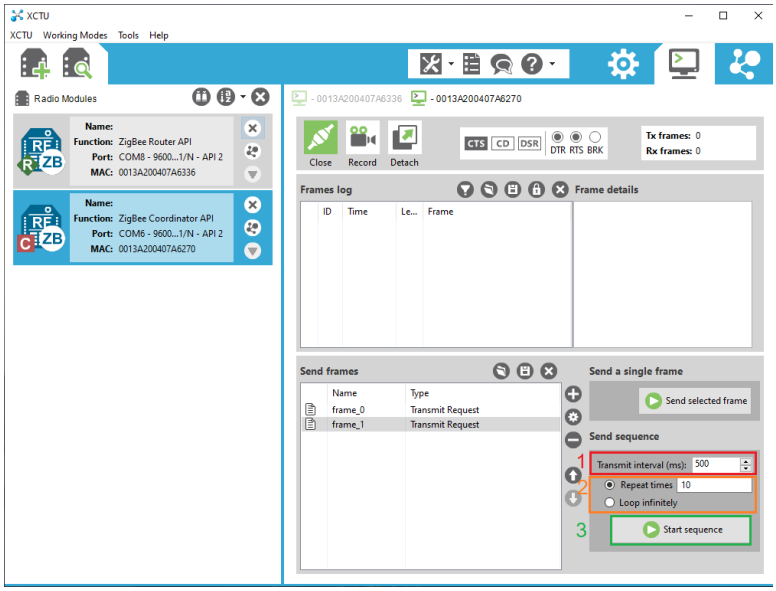
หลังจากนั้นทำการ Add Frame เพื่อบันทึกค่าข้อมูลทั้งหมดที่ได้ทำการตั้งค่าไว้ ดังภาพ 3.21 และเตรียมดำเนินการรับ-ส่งข้อมูลในขั้นตอนต่อไป



ภาพ 3.21 ทำการใส่ Adress ปลายทาง (ง)

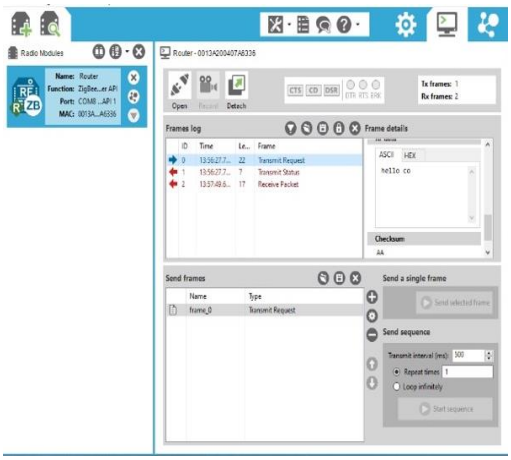
3.2.5 ทำการรับ-ส่งข้อมูล

ในการรับ-ส่งข้อมูลนั้น สามารถตั้งค่าได้ 2 ค่า คือ 1. ค่าระยะห่างเวลาในการส่งข้อมูล (หมายเลข 1) 2. สามารถกำหนดจำนวนครั้งในการส่งข้อมูลได้หรือเลือกที่จะให้ส่งข้อมูลไปเรื่อยๆ จนกว่าผู้ใช้จะสั่งหยุด (หมายเลข 2) และมีปุ่มคำสั่งเริ่มทำการส่ง (หมายเลข 3) ดังภาพ 3.22



ภาพ 3.22 เริ่มทำการรับ-ส่งข้อมูล

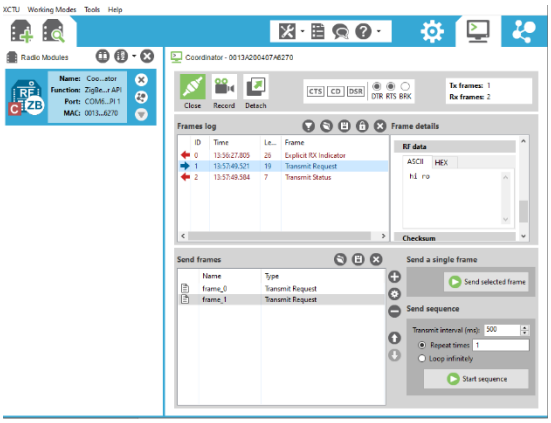
- จาก XBee Router ไปยัง XBee Coordinator จากภาพ 3.23 จะสังเกตได้ว่าตัว XBee Router ทำการส่งข้อความ “Hello co” ไปยัง XBee Coordinator ตอนเวลา 13:56:27 น. จากภาพ 3.24 จะสังเกตได้ว่า ข้อความที่ถูกส่งมาจากตัว XBee Router มาถึง XBee Coordinator ตอนเวลา 13:56:27 น. ซึ่งมีข้อความว่า “Hello co” เหมือนกัน



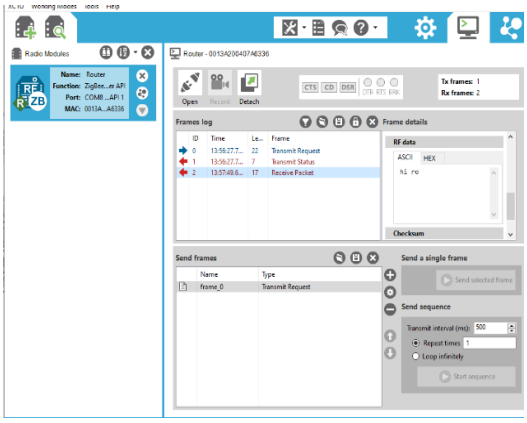
ภาพ 3.23 หน้าต่างของ XBee Router

ภาพ 3.24 หน้าต่างของ XBee Coordinator

- จาก XBee Coordinator ไปยัง XBee Router จากภาพ 3.25 จะสังเกตได้ว่า ตัว XBee Coordinator ได้ทำการส่งข้อความ “hi ro” ไปยัง XBee Router ตอนเวลา 13:57:49.58 น. จากภาพ 3.26 จะสังเกตได้ว่า ข้อความที่ถูกส่งมาจากตัว XBee Coordinator มาถึง XBee Router ตอนเวลา 13:57:49.6 น. ซึ่งมีข้อความว่า “hi ro” เหมือนกัน



ภาพ 3.25 หน้าต่างของ XBee Coordinator



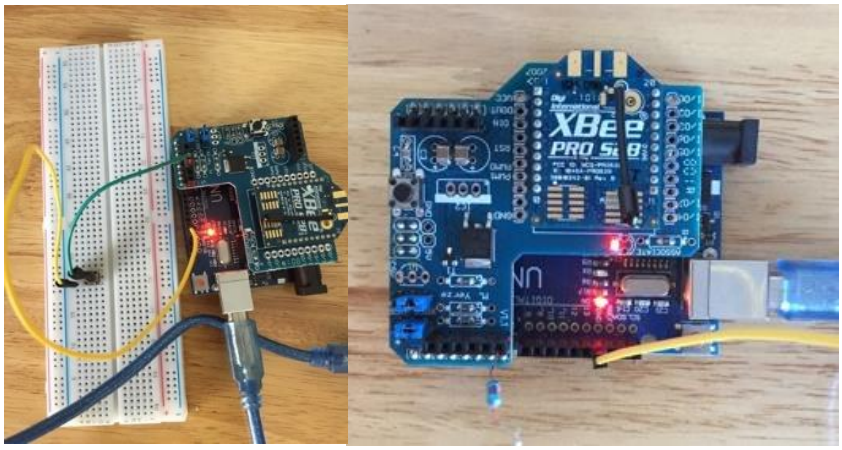
ภาพ 3.26 หน้าต่างของ XBee Router

ดังนั้นจึงสรุปได้ว่า XBee ทั้งสองตัวตั้งค่าได้อย่างถูกต้องและสามารถรับ-ส่งข้อมูลกันได้ หลังจากนั้นจะนำเอา XBee ทั้งสองตัวไปใช้ในขั้นตอนต่อไป

3.3 ทำการเชื่อมต่อแบบไร้สายระหว่าง Arduino กับ Arduino โดยมี XBee เป็นตัวกลางในการรับ-ส่งคำสั่งหรือข้อมูล สำหรับเปิด-ปิดไฟ

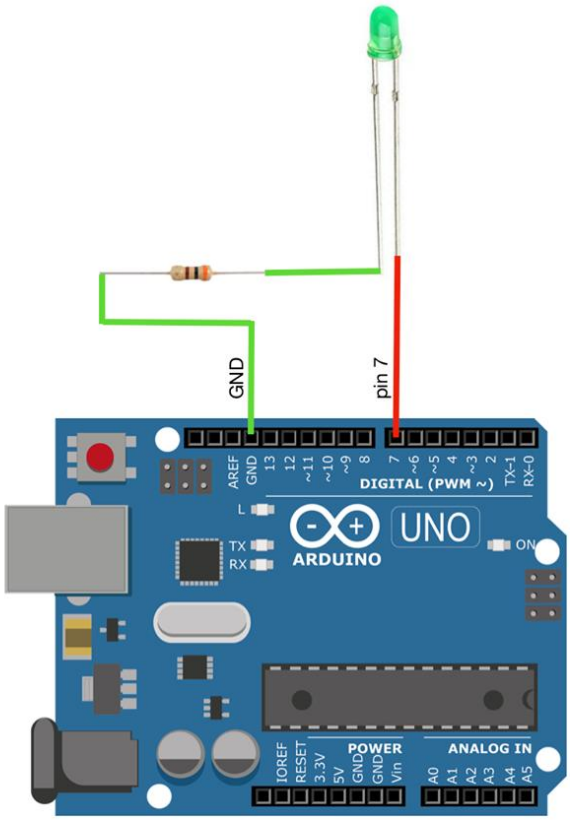
ในขั้นตอนนี้จะทำการประกอบ XBee เข้ากับแผง Arduino ทั้ง 2 ชุด โดยใช้บอร์ดสำหรับ Arduino XBee เป็นตัวกลางในการเชื่อมอุปกรณ์ทั้งสองชิ้นเข้าด้วยกัน ดังภาพ 3.27 และ 3.28 เพื่อทำการส่งคำสั่งจากบอร์ด Arduino + XBee ที่ทำหน้าที่เป็น Coordinator ไปยังบอร์ด Arduino + XBee ที่ทำหน้าที่เป็น Router และรอดูการตอบสนองของ XBee

โดยการทดสอบขั้นนี้จะทำการสั่งปิด-เปิดไฟแบบไร้สาย เพื่อตรวจสอบว่า บอร์ด Arduino และ XBee สามารถทำงานร่วมกันได้หรือไม่และตรวจสอบการทำงานของ XBee อีกครั้งว่ายังสามารถทำการรับ-ส่งคำสั่งหรือข้อมูลได้อยู่หรือไม่



(ก) (ข)

ภาพ 3.27 แสดงแผง Arduino ที่ทำหน้าที่ส่งคำสั่ง (ก) รับคำสั่ง (ข)



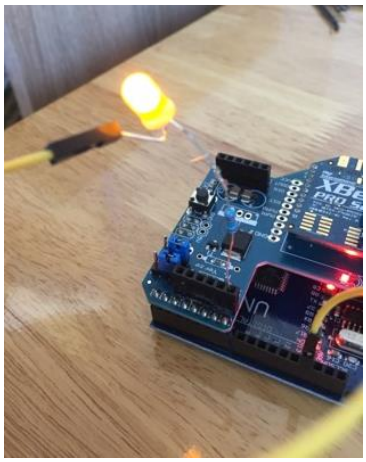
ภาพ 3.28 Wiring Diagram ของบอร์ด Arduino กับหลอดไฟ

หลังจากนั้นทำการเสียบสาย USB ที่ต่อจากบอร์ด Arduino เข้ากับคอมพิวเตอร์ 2 เครื่อง โดยคอมพิวเตอร์ 1 เครื่องจะเชื่อมต่อกับชุด Arduino + XBee เพียง 1 ชุดเท่านั้น และเมื่อทำการเชื่อมต่อเรียบร้อยแล้วจะทำการเข้าโปรแกรม Arduino IDE เพื่อทำการเขียนโค้ด ดังภาพ 3.29 เพื่อให้ XBee กับบอร์ด Arduino สื่อสารกันได้ แล้วจึงทำการทดสอบรับ-ส่งคำสั่งหรือข้อมูล โดยเริ่มจากใช้คอมพิวเตอร์ตัวที่ 1 ส่งข้อมูลไปยังคอมพิวเตอร์ตัวที่ 2

```
co_api_test §
#include <XBee.h>
XBee xbee = XBee();
XBeeResponse response = XBeeResponse();
ZBRxResponse rx = ZBRxResponse();

void setup() {
  Serial.begin(9600);
  xbee.setSerial(Serial);
  pinMode(7,OUTPUT);
}

void loop() {
  uint8_t data[] = "COORDINATOR-N-MIND";;
  XBeeAddress64 addr64 = XBeeAddress64(0x00000000, 0x0000FFFF);
  ZBTxRequest zbTx = ZBTxRequest(addr64, data, sizeof(data));
  xbee.send(zbTx);

////////////////////////////////////////////////////////////////////////////////
  int sample;
  xbee.readPacket();
    if (xbee.getResponse().isAvailable()) {
      Serial.println(xbee.getResponse().getApiId());
     if (xbee.getResponse().getApiId() == ZB_RX_RESPONSE) {
       xbee.getResponse().getZBRxResponse(rx);
      for (int i = 0; i < rx.getDataLength(); i++) {
         sample = (int)rx.getData(i);
       }
       Serial.println(sample);
      // int vrl = map(sample,0,1023,0,255);
      // analogWrite(7,vrl);
       if(sample==0){
   digitalWrite(7,1);
  }else{
    digitalWrite(7,0);
  }

    }
    }else if (xbee.getResponse().isError()) {
      Serial.println("Error reading packet.  Error code: ");
```

ภาพ 3.29 โค้ดสำหรับทำการทดสอบ

ทำการทดสอบสั่งให้หลอดไฟทำงานและหยุดทำงาน ดังภาพ 3.30 และ 3.31



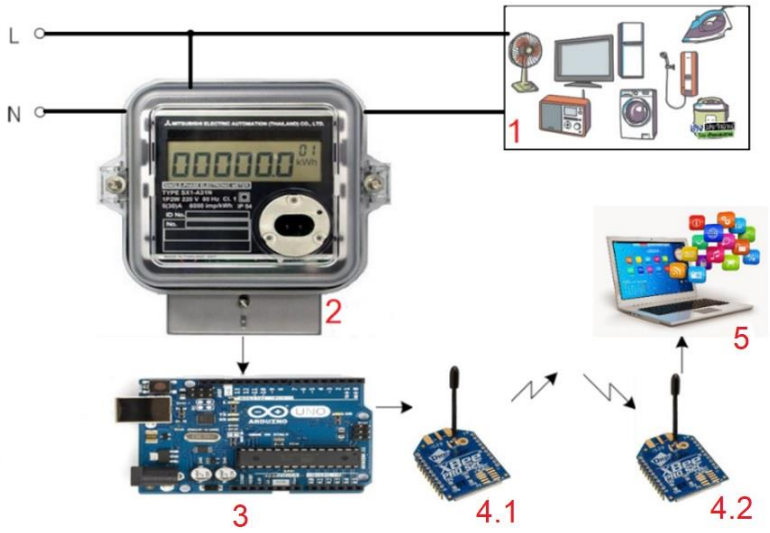
ภาพ 3.30 ทำการสั่งเปิดไฟ

ภาพ 3.31 ทำการสั่งปิดไฟ

ดังนั้นจึงสรุปได้ว่า บอร์ด Arduino กับ XBee สามารถใช้งานร่วมกันได้และสามารถนำไปทดสอบและพัฒนาในขั้นตอนต่อไปได้

3.4 ทำการออกแบบการเชื่อมต่อไร้สายระหว่าง Arduino + XBee กับ Digital Meter

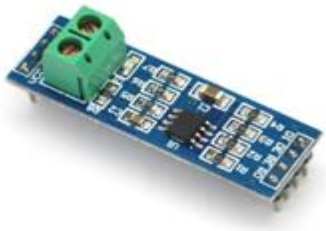
ในขั้นตอนนี้จะทำการเปลี่ยนจากการเชื่อมต่อแผง Arduino + XBee เข้ากับหลอดไฟเป็นเชื่อมต่อแผง Arduino + XBee เข้ากับดิจิตอลมิเตอร์ไฟฟ้า โดยเชื่อมต่อผ่านระบบ RS-485 และใช้บอร์ดขยาย RS-485 และ XBee เป็นตัวกลางในการเชื่อมต่อระหว่างแผง Arduino กับ ดิจิตอลมิเตอร์ไฟฟ้า ดังภาพ 3.32



ภาพ 3.32 แสดงการออกแบบและต่อวงจร

โดยเครื่องใช้ไฟฟ้า (1) เป็นภาระไฟฟ้าเชื่อมต่อกับด้านหลังของเครื่องมิเตอร์ไฟฟ้าดิจิตอลแบบไร้สาย มาตรฐาน RS-485 (2) (End Device) ซึ่งเป็นตัวบันทึกค่าหน่วยของไฟฟ้าที่ใช้ ซึ่งมีหน่วยเป็น กิโลวัตต์ชั่วโมง ซึ่งในการทำงานเพื่อที่จะได้ค่าหน่วยของไฟฟ้าที่ใช้ออกมานั้น จะต้องมีการติดตั้งตัวแผงควบคุม Arduino รุ่น UNO (3) เสริมเข้าไปเพื่อดึงข้อมูลค่าหน่วยของไฟฟ้าที่ใช้ออกมาแล้วส่งต่อไปยังตัวรับค่าที่มีชื่อว่า XBee (4.1) (Router) โดยผ่านคลื่นสัญญาณ Wi-Fi ที่ตัว XBee ปล่อยออกมาและสามารถส่งข้อมูลค่าหน่วยไฟฟ้าที่ใช้ไปยังตัว XBee (4.2) (Router) ที่อยู่ใกล้เพื่อที่จะส่งต่อข้อมูลไปยังคอมพิวเตอร์เพื่อทำการประมวลผลหาค่าไฟฟ้าที่ผู้บริโภคต้องชำระ โดยข้อมูลที่ได้รับมานั้นจะมีทั้งหมายเลขเครื่องแต่ละตัวพร้อมกับค่าหน่วยไฟฟ้าที่ใช้ของมิเตอร์ตัวที่ติดกับ XBee ตัวนั้นๆ ซึ่งสามารถเรียกดูข้อมูลได้ทีละหลาย ๆ เครื่องมิเตอร์ตามที่ผู้ดำเนินการต้องการอุปกรณ์ทั้งหมดในการทำการทดสอบ ดังตาราง 3.1

ตาราง 3.1 รายการอุปกรณ์ทั้งหมด

รายการ	อุปกรณ์	จำนวน
1	 XBee Pro S2B หรือ S2C	4 ตัว
2	 แผงควบคุม Arduino รุ่น UNO R3	4 ชุด
3	 บอร์ดขยาย Arduino XBee	4 ชุด
4	 บอร์ดรองรับระบบ RS-485	2 อัน

ตาราง 3.1 รายการอุปกรณ์ทั้งหมด (ต่อ)

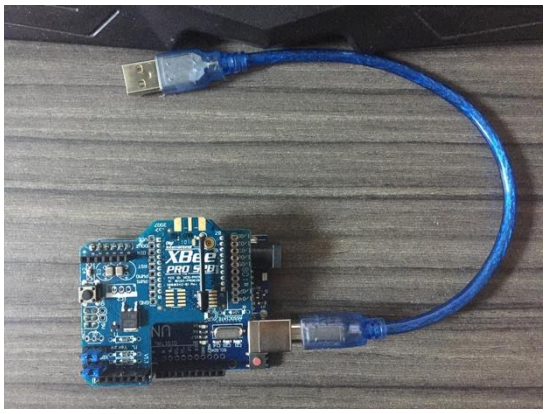
รายการ	อุปกรณ์	จำนวน
5	 ดิจิตอลมิเตอร์ไฟฟ้า รุ่น DDS238-4W	2 ตัว
6	 ถ่าน 9 V.	2 ก้อน
7	 ปลั๊กพ่วง	2 อัน

ในการทำการทดสอบจะแบ่งอุปกรณ์ออกเป็น 3 กลุ่ม

อุปกรณ์กลุ่ม 1 จะเชื่อมต่ออยู่กับคอมพิวเตอร์ ดังภาพ 3.33 3.34 3.35 โดยควบคุมผ่านโปรแกรม Arduino IDE จะทำหน้าที่เป็น Coordinator ซึ่งเป็นตัวเริ่มต้นของเครือข่าย ทำการขอ-รับค่าข้อมูลจากอุปกรณ์กลุ่ม 2 มีอุปกรณ์ดังนี้

1. XBee Pro S2B 1 ตัว
2. แผงควบคุม Arduino UNO 1 แผง
3. บอร์ดขยาย Arduino XBee 1 อัน

ภาพ 3.33 อุปกรณ์ที่ใช้

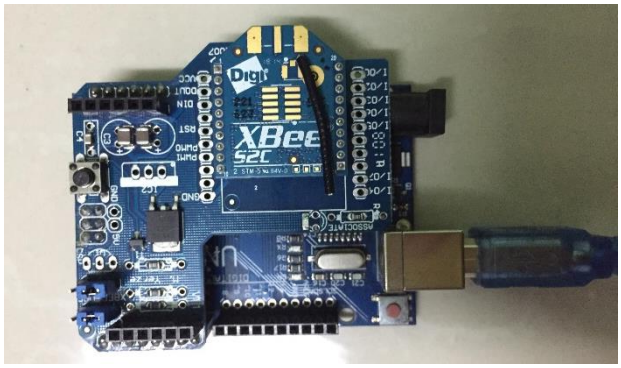


ภาพ 3.34 ประกอบอุปกรณ์ต่าง ๆ เข้าด้วยกัน

ภาพ 3.35 แสดงการเชื่อมต่อของอุปกรณ์กลุ่ม 1

อุปกรณ์กลุ่ม 2 จะเป็นตัวกลางอยู่ระหว่างดิจิตอลมิเตอร์ไฟฟ้ากับคอมพิวเตอร์ ทำหน้าที่เป็นตัวกลางคอยรับ-ส่งข้อมูลระหว่างกลุ่มที่ 1 และกลุ่มที่ 3 ซึ่งจะมีอุปกรณ์ดังภาพ 3.36 ดังนี้

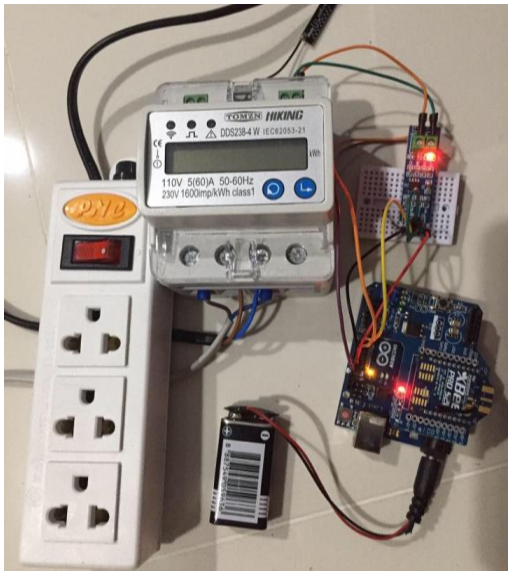
1. XBee Pro S2C 1 ตัว
2. แผงควบคุม Arduino UNO 1 แผง
3. บอร์ดขยาย Arduino XBee 1 อัน



ภาพ 3.36 แสดงการเชื่อมต่อของอุปกรณ์กลุ่ม 2

อุปกรณ์กลุ่ม 3 จะอยู่กับดิจิตอลมิเตอร์ไฟฟ้า ซึ่งเป็นปลายทางของเครือข่าย ดังนั้นเราจึงสามารถตั้งค่าให้ XBee ของกลุ่มที่ 3 ทำหน้าที่เป็นได้ทั้ง 2 แบบ คือ Router หรือ End device ก็ได้ ซึ่งจะเป็นตัวรับ-ส่งคำขอและข้อมูลจากอุปกรณ์กลุ่มที่ 2 กลุ่มนี้จะใช้ดิจิตอลมิเตอร์ไฟฟ้ามีอยู่ทั้งหมด 2 ชุด มีอุปกรณ์ ดังภาพ 3.37 ดังนี้

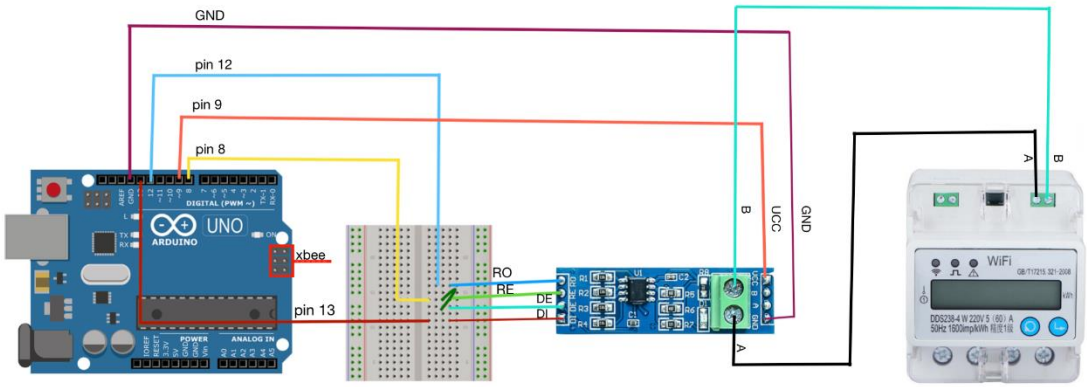
1. XBee Pro S2B ชุดละ 1 ตัว
2. แผงควบคุม Arduino รุ่น UNO ชุดละ 1 แผง
3. ดิจิตอลมิเตอร์ไฟฟ้า รุ่น DDS238-4W ชุดละ 1 ตัว
4. บอร์ดรองรับระบบ RS-485 ชุดละ 1 อัน
5. ถ่าน 9V. ชุดละ 1 ก้อน
6. ปลั๊กพ่วง ชุดละ 1 อัน



ภาพ 3.37 แสดงการเชื่อมต่อของอุปกรณ์กลุ่ม 3

สรุปการทำงาน โดยเริ่มจากอุปกรณ์กลุ่มที่ 1 ทำการส่งคำขอข้อมูลไปยังอุปกรณ์กลุ่มที่ 2 และกลุ่มที่ 2 จะทำการทำซ้ำคำขอข้อมูลจากอุปกรณ์กลุ่มที่ 1 ส่งไปยังอุปกรณ์กลุ่มที่ 3 เมื่อคำขอข้อมูลส่งมาถึงอุปกรณ์กลุ่มที่ 3 ก็จะส่งค่าหน่วยไฟฟ้าที่ใช้ไปกับมาที่อุปกรณ์กลุ่มที่ 2 และอุปกรณ์กลุ่มที่ 2 ก็จะทำการทำซ้ำส่งค่าข้อมูลหน่วยไฟฟ้าไปยังอุปกรณ์กลุ่มที่ 1 และเมื่อข้อมูลมาถึงอุปกรณ์กลุ่มที่ 1 แล้ว ข้อมูลเลขหน่วยไฟฟ้าที่ใช้ไปก็จะมาปรากฏบนหน้าต่างของโปรแกรม Arduino IDE บนคอมพิวเตอร์

ซึ่งการเชื่อมต่อสายไฟสามารถทำได้ดังภาพ 3.38



ภาพ 3.38 Wiring Diagram ของอุปกรณ์กลุ่ม 3

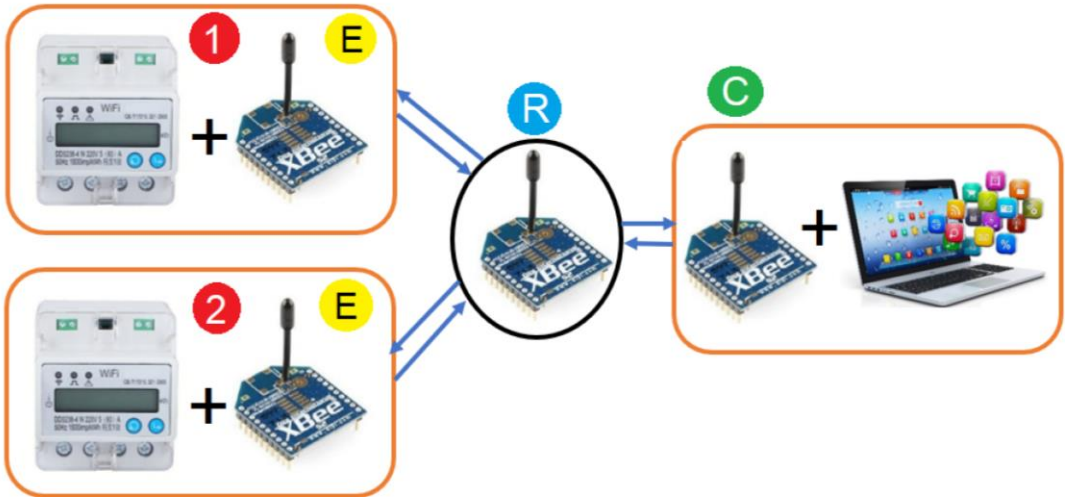
หลังจากนั้นทำการเขียนโค้ดบนโปรแกรม Arduino IDE ดังภาพ 3.39 เพื่อทำการทดลองการรับ-ส่งข้อมูล

```
#include <XBee.h>
#include "StringSplitter.h"
XBee xbee = XBee();
XBeeResponse response = XBeeResponse();
ZBRxResponse rx = ZBRxResponse();
//int num = 99;

void setup() {
  Serial.begin(9600);
  xbee.setSerial(Serial);
}
void loop() {

  const unsigned long fivesecond = 5 * 1000UL;
  static unsigned long lastSampleTime = 0 - fivesecond;  // initialize such that a reading is due the first time through loop()
  unsigned long now = millis();
  if (now - lastSampleTime >= fivesecond)
  {
    lastSampleTime += fivesecond;
    uint8_t data[] = "COORDINATOR-N-MIND";
  XBeeAddress64 addr64 = XBeeAddress64(0x0013A200, 0x407A6336);
```

ภาพ 3.39 แสดงโค้ดที่ใช้ในการสั่งการแผงควบคุม Arduino

3.5 ทดสอบการใช้งานในรูปแบบต่างๆ

ในการทดสอบการใช้งานนำค่าสองค่าที่ได้รับจากมิเตอร์ไฟฟ้ามาเปรียบเทียบ ระหว่าง มาตรวัดมิเตอร์ กับ ผลที่แสดงบนหน้าจอคอมพิวเตอร์ ที่ผ่าน XBee ว่าตรงกันหรือไม่ โดยมีรูปแบบดังภาพ 3.40



ภาพ 3.40 เครือข่ายแบบผ่าน XBee ตัวกลาง

โดยการทดสอบจะใช้โปรแกรม Arduino IDE ในการควบคุมระบบซึ่งในการทดสอบจะมุ่งเน้นความแม่นยำเป็นหลัก โดยจะทำการทดสอบรับ-ส่งข้อมูล เป็นรอบ รอบละ 20 ครั้ง

3.6 ทำการบันทึกผลในการทดสอบ

เพื่อนำข้อมูลที่ได้นั้นนำไปวิเคราะห์เพื่อหาข้อสรุปต่อไป

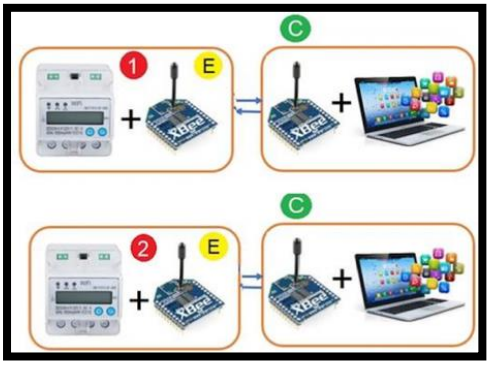
3.7 สรุปผลและจัดทำรายงาน

วิเคราะห์ผลที่ได้จากการจดบันทึกระหว่างบนมาตรวัดบนดิจิตอลมิเตอร์ไฟฟ้ากับค่าที่แสดงบน Arduino IDE ทำการเปรียบผลและสรุปผล

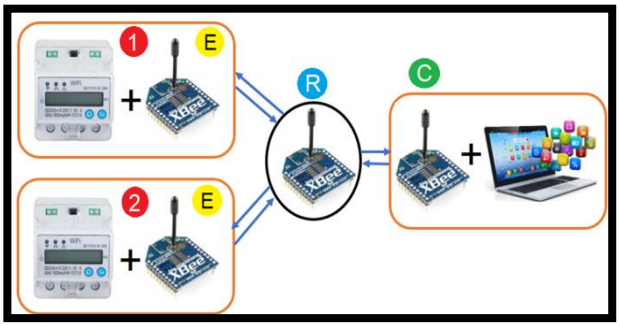
บทที่ 4

ผลการดำเนินงาน

ในบทนี้จะแสดงผลการดำเนินงานที่ได้จากการทดลอง โดยจะแสดงผลความแม่นยำในการรับ-ส่งข้อมูลแบบ Point to Point คือการไม่ผ่าน XBee ตัวกลางและแบบ Cluster Tree คือการผ่าน XBee ตัวกลางด้วยการติดตั้ง XBee กับ Arduino เป็นตัวกลางในการรับ-ส่งข้อมูลทำได้ด้วยการเขียนโค้ดลง Arduino IDE โดยผ่านโปรแกรม Arduino และทำการทดสอบการรับ-ส่งของข้อมูล ดังภาพ 4.1 และ 4.2



ภาพ 4.1 แบบ Point to Point



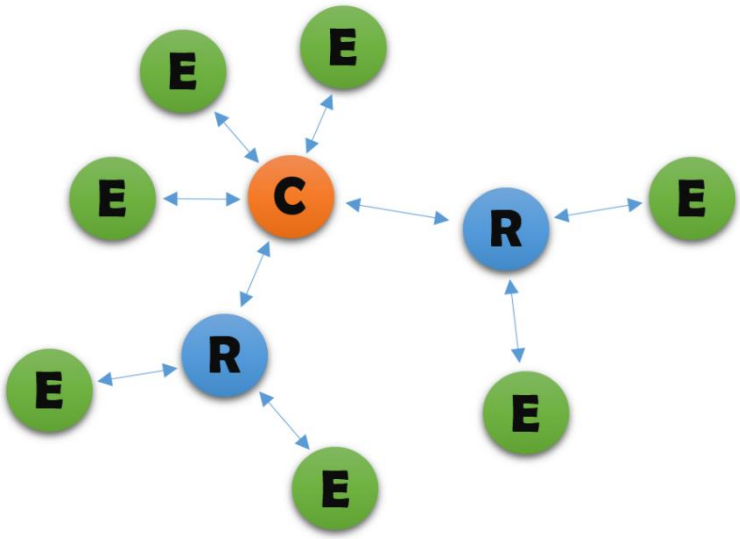
ภาพ 4.2 แบบ Cluster Tree

โดยที่หมายเลข 1 คือดิจิตอลมิเตอร์ไฟฟ้า และ (E) คือบอร์ด Arduino ที่มีการติดตั้ง XBee ที่ทำหน้าที่เป็น End Device ในการรับ-ส่งสัญญาณและเชื่อมต่อกับดิจิตอลมิเตอร์ไฟฟ้า และ (C) คือบอร์ด Arduino ที่มีการติดตั้ง XBee ที่ทำหน้าที่เป็น Coordinator ในการรับ-ส่งสัญญาณและเชื่อมต่อกับคอมพิวเตอร์เพื่อแสดงค่าบนจอภาพ ซึ่งใช้ในการดูค่าที่ส่งมาและเปรียบเทียบค่ากับหน้าจอบนดิจิตอลมิเตอร์ไฟฟ้า ส่วน (R) คือบอร์ด Arduino ที่มีการติดตั้ง XBee ที่ทำหน้าที่เป็น Router ในการรับ-ส่งสัญญาณใช้เป็นตัวกลางในการรับส่งข้อมูลระหว่าง XBee (Coordinator และ End Device) ทั้ง 2 ตัว เพื่อเพิ่มระยะในการรับ-สั่งส่งสัญญาณข้อมูล

การเชื่อมต่อเครือข่ายแบบ Point to Point เป็นการเชื่อมต่อแบบตัวต่อตัว โดยกำหนดให้ตัวหนึ่งเป็น Coordinator และอีกตัวหนึ่งเป็น Router หรือ End Device ก็ได้ ดังภาพ 4.3

ภาพ 4.3 เครือข่ายแบบ Point to Point

การเชื่อมต่อเครือข่ายแบบ Cluster Tree เป็นการรับส่งข้อมูลแบบส่งผ่านหรือส่งต่อ เช่น A ต้องการติดต่อกับ C แต่ C อยู่ไกลจาก A จน A ไม่สามารถติดต่อกับ C ได้โดยตรง แต่เนื่องจากมี B อยู่ระหว่าง A กับ C ดังนั้น Cluster Tree จะใช้ B เป็นเหมือน ตัวกลางเชื่อมการติดต่อ (Repeater) ระหว่าง A กับ C ดังภาพ 4.4



ภาพ 4.4 เครือข่ายแบบ Cluster Tree

ในการทดสอบโดยการเขียนโค้ดผ่านโปรแกรม Arduino IDE อัปโหลดลงบอร์ด Arduino จะมีด้วยกันทั้งหมด 3 โค้ด ซึ่งจะแบ่งเป็นโค้ดของ Coordinator , Router และ End Device ดังภาพ 4.5 4.6 และ 4.7 โดยที่โค้ด Coordinator จะอัปโหลดลงบอร์ด Arduino ที่เชื่อมต่อกับคอมพิวเตอร์ ส่วนโค้ด Router จะอัปโหลดลงบอร์ด Arduino ที่อยู่ระหว่างกลางของดิจิตอลมิเตอร์ไฟฟ้าคอมพิวเตอร์ในการรับ-ส่งข้อมูลไร้สายและโค้ด End Device จะอัปโหลดลงบอร์ด Arduino ที่เชื่อมต่อกับดิจิตอลมิเตอร์ไฟฟ้า

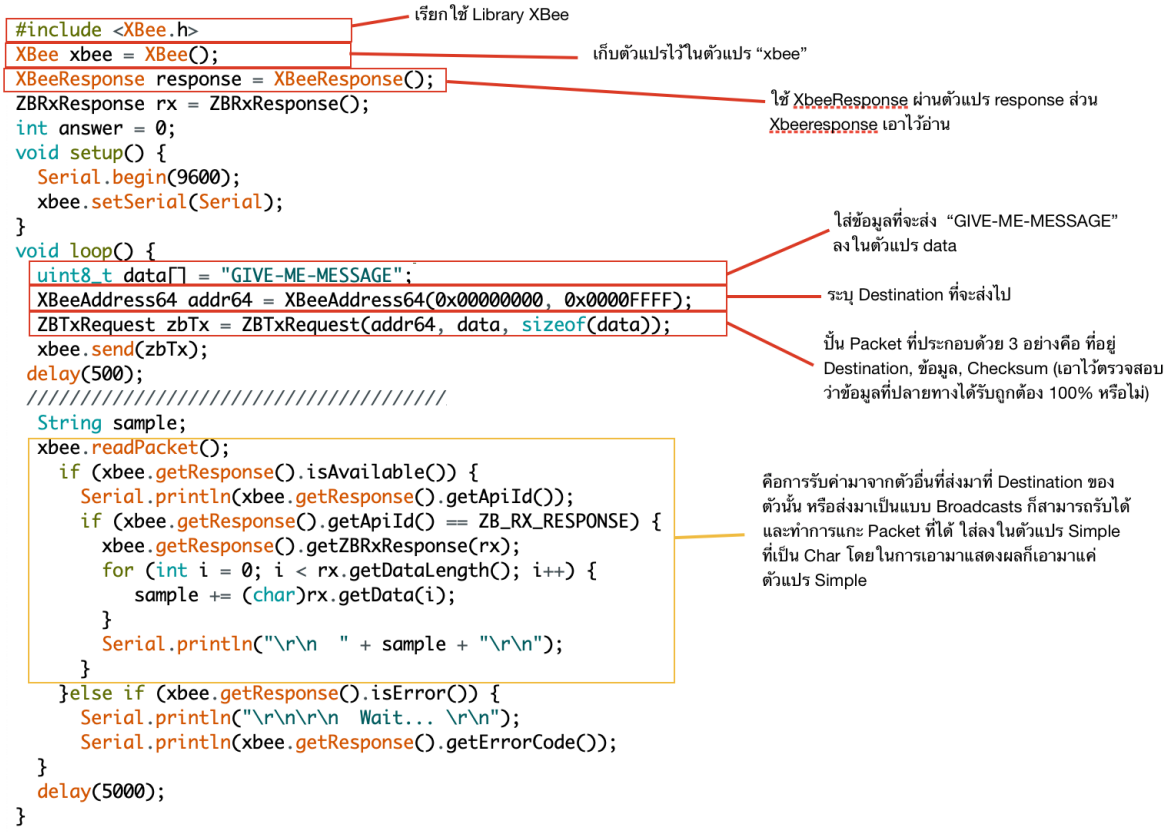
```
#include <XBee.h>
XBee xbee = XBee();
XBeeResponse response = XBeeResponse();
ZBRxResponse rx = ZBRxResponse();
int answer = 0;
void setup() {
  Serial.begin(9600);
  xbee.setSerial(Serial);
}
void loop() {
  uint8_t data[] = "GIVE-ME-MESSAGE";
  XBeeAddress64 addr64 = XBeeAddress64(0x00000000, 0x0000FFFF);
  ZBTxRequest zbTx = ZBTxRequest(addr64, data, sizeof(data));
  xbee.send(zbTx);
 delay(500);
 ////////////////////////////////////
  String sample;
  xbee.readPacket();
    if (xbee.getResponse().isAvailable()) {
      Serial.println(xbee.getResponse().getApiId());
      if (xbee.getResponse().getApiId() == ZB_RX_RESPONSE) {
        xbee.getResponse().getZBRxResponse(rx);
        for (int i = 0; i < rx.getDataLength(); i++) {
           sample += (char)rx.getData(i);
        }
        Serial.println("\r\n  " + sample + "\r\n");
      }
    }else if (xbee.getResponse().isError()) {
      Serial.println("\r\n\r\n  Wait... \r\n");
      Serial.println(xbee.getResponse().getErrorCode());
  }
  delay(5000);
}
```

ใส่ Address ที่จะส่ง (0x00000000, 0x0000FFFF) คือส่งให้ทุกตัวที่สามารถรับได้ โดยสิ่งที่ส่ง “GIVE-ME-MESSAGE”

คือการรับค่ามาจากตัวอื่นที่ส่งมาที่ Destination ของตัวนั้น หรือส่งมาเป็นแบบ Broadcasts ก็สามารถรับได้ และทำการแกะ Packet ที่ได้ ใส่ลงในตัวแปร Simple ที่เป็น Char โดยในการเอามาแสดงผลก็เอามาแค่ตัวแปร Simple

ภาพ 4.5 Code of Coordinator

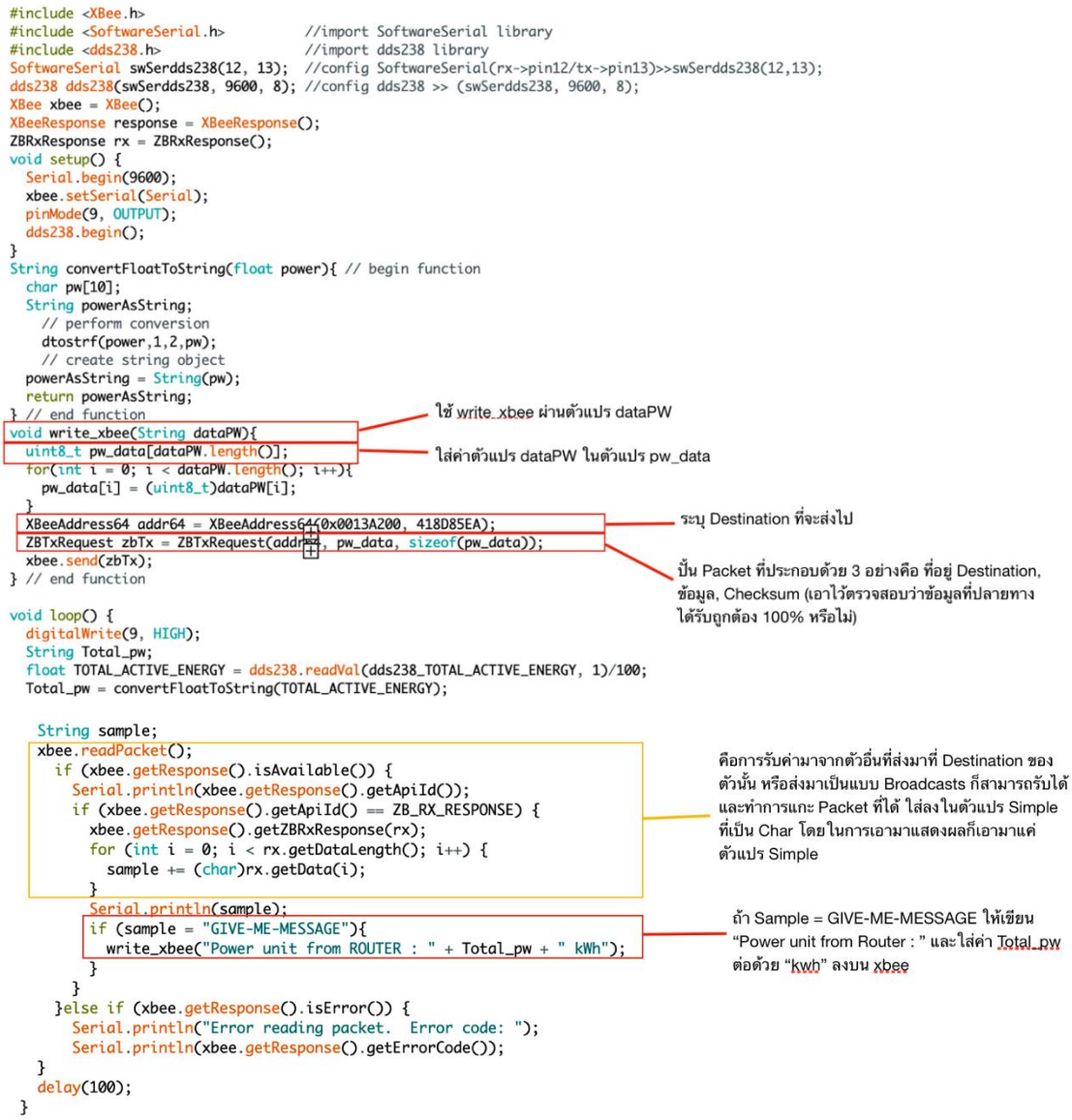
โค้ด Coordinator ทำงานโดยการส่งค่าหรือข้อความว่า “GIVE-ME-MESSAGE” ไปยัง XBee ที่กำหนดไว้เพื่อกระตุ้นให้ XBee ตัวนั้นส่งค่าที่ต้องการกลับมาแสดงผลบนหน้าจอคอมพิวเตอร์ ซึ่งในที่นี้ XBee ตัวนั้นคือ ตัวที่ทำหน้าที่เป็น Router




```
#include <XBee.h>
XBee xbee = XBee();
XBeeResponse response = XBeeResponse();
ZBRxResponse rx = ZBRxResponse();
int answer = 0;
void setup() {
  Serial.begin(9600);
  xbee.setSerial(Serial);
}
void loop() {
  uint8_t data[] = "GIVE-ME-MESSAGE";
  XBeeAddress64 addr64 = XBeeAddress64(0x00000000, 0x0000FFFF);
  ZBTxRequest zbTx = ZBTxRequest(addr64, data, sizeof(data));
  xbee.send(zbTx);
 delay(500);
 ////////////////////////////////////
  String sample;
  xbee.readPacket();
    if (xbee.getResponse().isAvailable()) {
      Serial.println(xbee.getResponse().getApiId());
      if (xbee.getResponse().getApiId() == ZB_RX_RESPONSE) {
        xbee.getResponse().getZBRxResponse(rx);
        for (int i = 0; i < rx.getDataLength(); i++) {
           sample += (char)rx.getData(i);
        }
        Serial.println("\r\n  " + sample + "\r\n");
      }
    }else if (xbee.getResponse().isError()) {
      Serial.println("\r\n\r\n  Wait... \r\n");
      Serial.println(xbee.getResponse().getErrorCode());
  }
  delay(5000);
}
```

ภาพ 4.6 Code of Router

โค้ด Router ทำงานเป็นตัวกลางในการรับ-ส่งข้อมูล โดยการรับค่าจาก End Device แล้วส่งไปยัง Coordinator เพื่อไปแสดงผลบนหน้าจอคอมพิวเตอร์



ภาพ 4.7 Code of End Device

โค้ด End Device จะทำงานโดยการรับค่าจากดิจิตอลมิเตอร์ไฟฟ้าส่งไปยังตัวกลาง XBee ที่ทำหน้าที่เป็น Router หลังจากนั้น ตัวกลาง XBee ที่ทำหน้าที่เป็น Router ก็จะส่งข้อมูลไปยัง XBee Coordinator เพื่อที่จะได้สามารถอ่านค่าบนโปรแกรม Arduino IDE บนหน้าจอคอมพิวเตอร์ได้

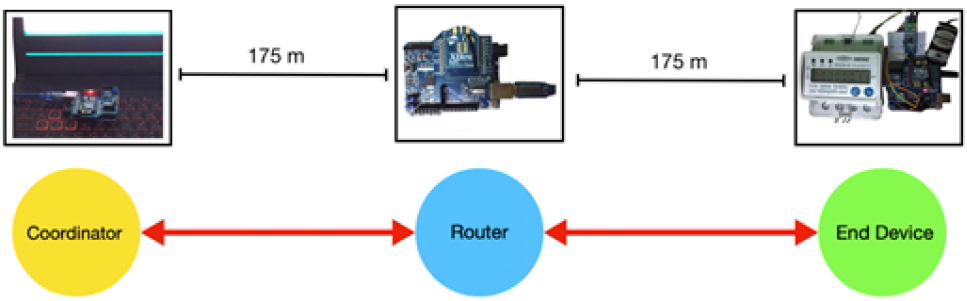
4.1 การทดสอบประสิทธิภาพของการรับ-ส่งข้อมูลไร้สาย

การทดสอบในแต่ละครั้งสำหรับการส่งข้อมูลสามรถรับค่าได้หรือไม่ เพื่อวิเคราะห์ประสิทธิภาพของระบบ โดยจะทดสอบอยู่ 2 แบบคือแบบ Point to Point เป็นการรับ-ส่งของข้อมูลไร้สายแบบที่มีแค่ Coordinator กับ End Device หรือจะเป็น Router ก็ได้ซึ่งจะไม่มีตัวกลางตัวอื่นระหว่างของตัวนี้และแบบ Cluster Tree เป็นการรับ-ส่งข้อมูลโดยมี Router เป็นตัวกลางอยู่ระหว่าง Coordinator กับ End Device โดยในการทดสอบแต่ละรูปแบบในหนึ่งรอบการทดลองจะทำการรับ-ส่งข้อมูล 20 ครั้งแล้วแสดงผลตามตาราง 4.1 และ 4.2

ตาราง 4.1 การทดสอบประสิทธิภาพของการรับ-ส่งข้อมูลแบบ Point to Point

รอบ	จำนวนในการส่งข้อมูล (ครั้ง)	จำนวนการได้รับข้อมูล (ครั้ง)	ประสิทธิภาพ (เปอร์เซ็นต์)
1	20	20	100
2	20	20	100
3	20	20	100
4	20	20	100
5	20	20	100
6	20	20	100
7	20	20	100
8	20	20	100
9	20	20	100
10	20	20	100

แบบจำลองการทดสอบแบบ Cluster Tree ดังภาพ 4.8



ภาพ 4.8 แบบจำลองการทดสอบแบบ Cluster Tree

ตาราง 4.2 การทดสอบประสิทธิภาพของการรับ-ส่งข้อมูลแบบ Cluster Tree

รอบ	จำนวนในการส่งข้อมูล (ครั้ง)	จำนวนการได้รับข้อมูล (ครั้ง)	ประสิทธิภาพ (เปอร์เซ็นต์)
1	20	8	40
2	20	9	45
3	20	9	45
4	20	10	50
5	20	8	40
6	20	9	45
7	20	10	50
8	20	8	40
9	20	9	45
10	20	9	45

จากการวิเคราะห์พบว่าการรับ-ส่งข้อมูลแบบ Point to Point มีประสิทธิภาพในการรับ-ส่งข้อมูล 100 เปอร์เซ็นต์ ซึ่งดีกว่าแบบ Cluster Tree ที่มีประสิทธิภาพในการรับ-ส่งข้อมูลเพียง 44.5 เปอร์เซ็นต์ มีประสิทธิภาพสูงกว่าเนื่องจากไม่ต้องรับ-ส่งข้อมูลผ่านตัวกลางทำให้เป็นความเสถียรมากกว่าในการรับ-ส่งข้อมูล

4.2 การทดสอบความแม่นยำของการรับ-ส่งข้อมูลไร้สาย

การทดสอบความแม่นยำทำโดยการเปรียบเทียบค่าที่รับกับค่าที่ส่งมาตรงกันหรือไม่ เพื่อวิเคราะห์ความแม่นยำของระบบ โดยจะทดสอบอยู่ 2 แบบคือแบบ Point to Point และแบบ Cluster Tree โดยในการทดสอบแต่ละรูปแบบในหนึ่งรอบการทดลองจะทำการรับ-ส่งข้อมูล 20 ครั้ง แล้วแสดงผลตามตาราง 4.3 และ 4.4

ตาราง 4.3 การทดสอบแม่นยำของการรับ-ส่งข้อมูลแบบ Point to Point

รอบ	จำนวนการทดลอง (ครั้ง)	ความแม่นยำ (เปอร์เซ็นต์)
1	20	100
2	20	100
3	20	100
4	20	100
5	20	100

ตาราง 4.4 การทดสอบแม่นยำของการรับ-ส่งข้อมูลแบบ Cluster Tree

รอบ	จำนวนการทดลอง (ครั้ง)	ความแม่นยำ (เปอร์เซ็นต์)
1	20	100
2	20	100
3	20	100
4	20	100
5	20	100

จากการวิเคราะห์พบว่าค่าความแม่นยำในการรับ-ส่งข้อมูลแบบ Point to Point และแบบ Cluster Tree มีความแม่นยำอยู่ที่ 100 เปอร์เซ็นต์

4.3 การทดสอบระยะของการรับ-ส่งข้อมูลไร้สาย

ในการทดสอบระยะสัญญาณการรับ-ส่งข้อมูลเพื่อหาระยะเฉลี่ยในการรับ-ส่งข้อมูล เพื่อวิเคราะห์ความแม่นยำของระบบโดยจะทำการทดสอบทั้งหมด 10 ครั้ง ซึ่งจะแบ่งการทดสอบออกเป็น 2 แบบคือแบบ Point to Point และแบบ Cluster Tree สามารถสรุปผลได้ดังตาราง 4.5 และ 4.6

ตาราง 4.5 การทดสอบระยะของการรับ-ส่งข้อมูลแบบ Point to Point

รอบ	ระยะสัญญาณ (เมตร)
1	250
2	260
3	250
4	270
5	265
6	260
7	260
8	265
9	250
10	255
ระยะสัญญาณเฉลี่ย	**258.5**

ตาราง 4.6 การทดสอบระยะของการรับ-ส่งข้อมูลแบบ Cluster Tree

รอบ	ระยะสัญญาณ (เมตร)
1	350
2	340
3	350
4	355
5	345
6	350
7	355
8	350
9	340
10	350
ระยะสัญญาณเฉลี่ย	**348.5**

จากการวิเคราะห์พบว่าระยะสัญญาณในการรับ-ส่งข้อมูลแบบ Cluster Tree มีระยะไกลมากกว่าแบบ Point to Point เนื่องจากมีตัวกลางทำให้สามารถเพิ่มระยะทางให้ไกลขึ้นเฉลี่ยอยู่ที่ 110 เมตรสำหรับการรับ-ส่งข้อมูล

บทที่ 5

สรุปผลการดำเนินงาน

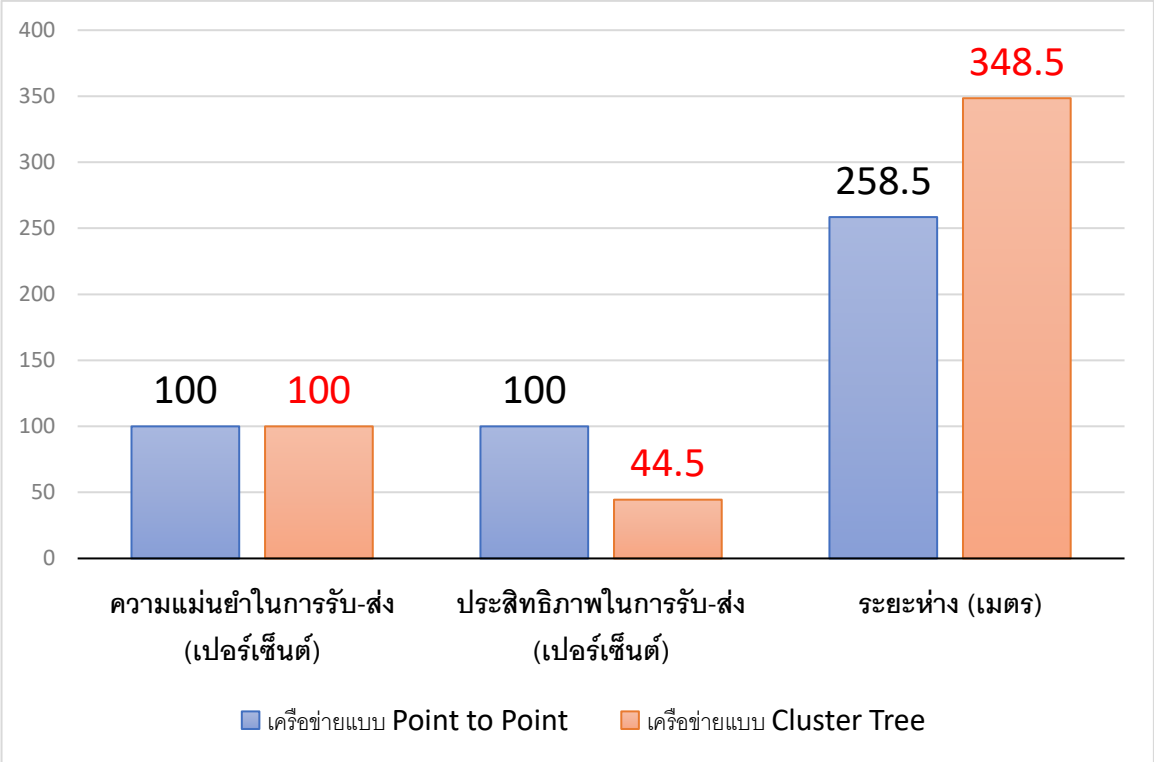
งานวิจัยนี้ได้ศึกษาเกี่ยวกับการประยุกต์ใช้อุปกรณ์สัญญาณไร้สาย IoT ในการบันทึกข้อมูลดิจิตอลมิเตอร์ไฟฟ้า โดยมีจุดมุ่งเน้นในการพัฒนาให้เราสามารถจดบันทึกค่าได้อย่างรวดเร็วและแม่นยำยิ่งขึ้นเพื่อลดขั้นตอนการทำงานของผู้ดำเนินการ เช่น การนำระบบไร้สายมาประยุกต์ใช้กับมิเตอร์ที่เป็นแบบดิจิตอล โดยผ่านตัว XBee ที่เป็นเหมือนตัวรับและตัวส่งสัญญาณเพื่อที่จะเอาค่าที่รับมาแสดงผลบนหน้าจอคอมพิวเตอร์ โดยที่ไม่จำเป็นต้องไปที่หน้ามิเตอร์เพื่ออ่านค่าอีกต่อไป ซึ่งในบทนี้จะกล่าวถึงการสรุปผลที่ได้จากการดำเนินงาน ปัญหาที่พบจากการดำเนินงานและข้อเสนอแนะในการทำโครงงาน

5.1 อภิปรายผลการทดลอง

จากผลการทดลองพบว่า การสร้างระบบสัญญาณไร้สายเพื่อใช้ในการรับ-ส่งข้อมูลจากดิจิตอลมิเตอร์ไฟฟ้าไปยังคอมพิวเตอร์ สามารถนำมาประยุกต์ใช้ให้เป็น IoT (Internet of Things) ได้ โดยการใช้บอร์ดควบคุม Arduino และ XBee ซึ่งใช้งานร่วมกันกับดิจิตอลมิเตอร์ไฟฟ้าได้ตามทฤษฎี โดยการรับ-ส่งข้อมูลนั้นสามารถทำได้ 2 รูปแบบ ดังนี้ 1. แบบ Point to Point 2. แบบ Cluster Tree ซึ่งทั้ง 2 แบบนั้นส่งข้อมูลได้แม่นยำทุกครั้ง โดยวิเคราะห์จากการทดสอบความแม่นยำในการรับ-ส่งข้อมูล และระบบทั้ง 2 แบบ มีการรับ-ส่งข้อมูลแบบไร้สายได้อย่างมีประสิทธิภาพ โดยการทดสอบประสิทธิภาพในการรับ-ส่งข้อมูล อย่างไรก็ตามตัวกลางและระยะห่างของแหล่งรับกับแหล่งส่งข้อมูลมีผลโดยตรงกับประสิทธิภาพในการรับ-ส่งข้อมูลของระบบไร้สายโดยผ่านตัวรับ-ส่งสัญญาณ XBee และสิ่งที่มีผลต่อผลการทดสอบมากที่สุด คือ การที่มีตัวกลางระหว่างแหล่งรับข้อมูลกับแหล่งส่งข้อมูล

5.2 สรุปผลการดำเนินงาน

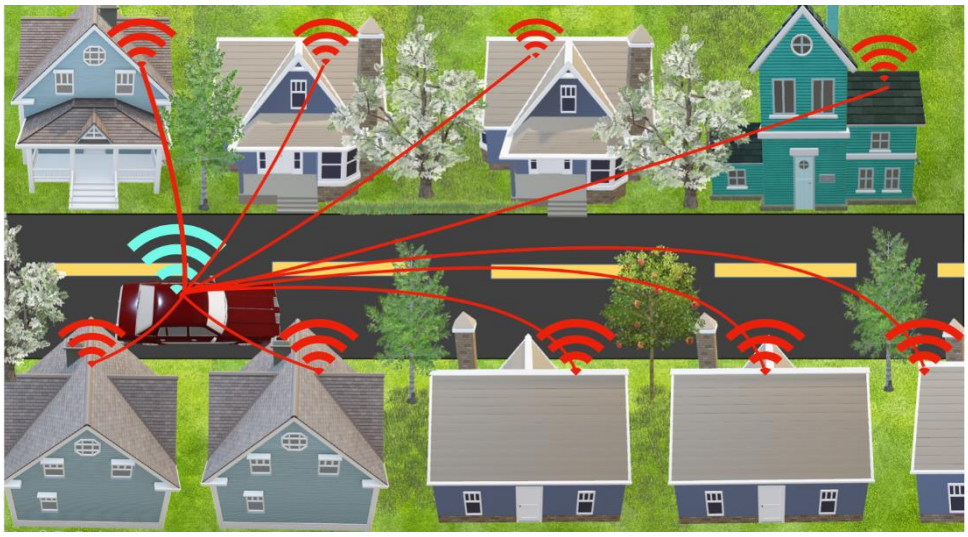
จากผลการทดสอบสรุปได้ว่าการเชื่อมต่อแบบไร้สายระหว่างคอมพิวเตอร์กับดิจิตอลมิเตอร์ไฟฟ้า ผ่านอุปกรณ์รับ-ส่งสัญญาณ XBee ซึ่งนำมาประยุกต์ใช้ร่วมกับแผงบอร์ด Arduino สามารถใช้งานได้จริงและเครือข่ายทั้ง 2 แบบ มีความแม่นยำ 100 เปอร์เซ็นต์ สามารถทำงานรับ-ส่งข้อมูลได้เป็นทอด ๆ แบบเครือข่ายซึ่งมีประสิทธิภาพ ดังนี้ แบบเครือข่าย Point to Point และแบบ Cluster Tree มีประสิทธิภาพการรับ-ส่งข้อมูล 100 เปอร์เซ็นต์ และ 44.5 เปอร์เซ็นต์ ตามลำดับ และมีระยะเฉลี่ยในการรับ-ส่งข้อมูล แบบ Point to Point และแบบ Cluster Tree อยู่ที่ 258.5 เมตรและ 348.5 เมตรตามลำดับ ซึ่งค่าทั้งหมดถูกนำมาเปรียบเทียบดังภาพ 5.1 จึงสรุปได้ว่าบอร์ดและโปรแกรม Arduino XBee มีประโยชน์เป็นอย่างมากในการที่จะรับ-ส่งข้อมูลในระยะไกล เหมาะที่จะนำไปใช้งานในชีวิตจริงและผลจากการทดสอบแสดงให้เห็นว่าข้อมูลที่ได้มานั้นไม่เกิดความคลาดเคลื่อนเลยแม้แต่น้อย จึงเป็นทางเลือกใหม่อีกทางเลือกหนึ่งที่จะหันมาใช้วิธีนี้



ภาพ 5.1 กราฟเปรียบเทียบข้อมูลจากการทดสอบรูปแบบต่าง ๆ ระหว่างเครือข่ายแบบ Point to Point กับแบบ Cluster Tree

ภาพตัวอย่างการจำลองการนำการเชื่อมต่อดิจิตอลมิเตอร์แบบไร้สายไปใช้งานในชีวิตประจำวันสามารถทำได้ 3 แบบ ดังนี้

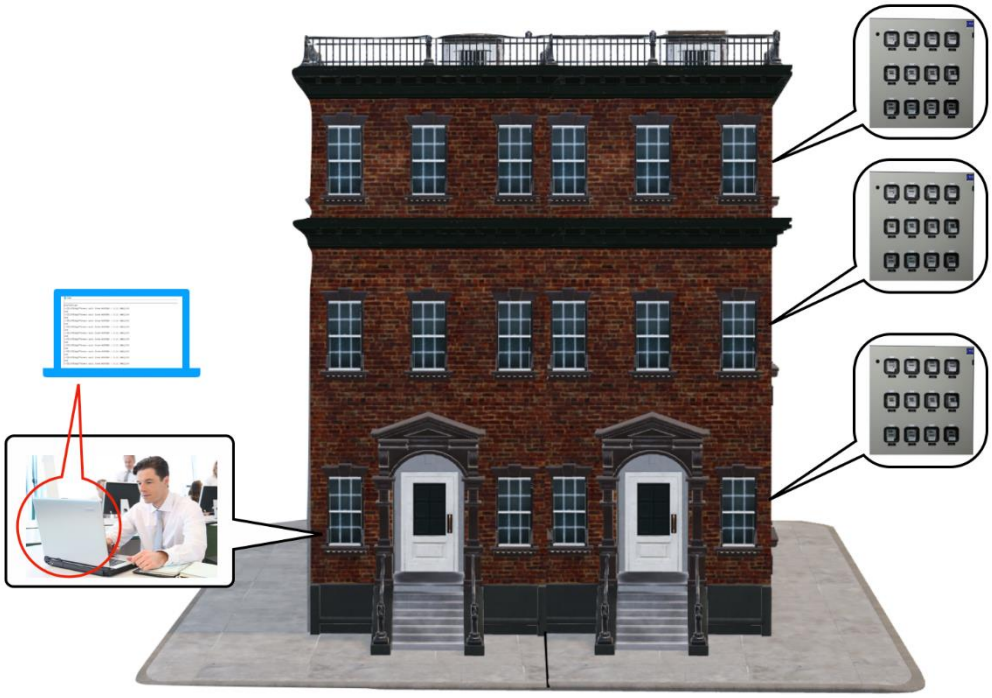
1. แบบ Point to Point เหมาะกับการนำไปใช้ในที่สาธารณะ ดังภาพ 5.2



ภาพ 5.2 ภาพจำลองการนำแบบ Point to Point มาใช้ในชุมชน

จากภาพ 5.2 สามารถอธิบายได้ดังนี้ ยานพาหนะที่ใช้ในการบันทึกหน่วยไฟฟ้าที่ใช้ไปนั้นจะต้องมีคอมพิวเตอร์ที่มีอุปกรณ์กลุ่ม 1 (บอร์ด Arduino และ XBee หน้าที่ Coordinator) เชื่อมต่ออยู่ เพื่อทำการส่งคำขอข้อมูลจากดิจิตอลมิเตอร์ที่อยู่ในบริเวณบ้านแต่ละหลัง ซึ่งดิจิตอลมิเตอร์นั้นจะต้องเชื่อมอยู่กับอุปกรณ์กลุ่ม 3 (บอร์ด Arduino และ XBee หน้าที่ Router / End Device) ซึ่งเป็นตัวที่ทำให้สามารถส่งข้อมูลกลับไปยังยานพาหนะที่ใช้ในการบันทึกหน่วยไฟฟ้าได้ และยานพาหนะที่ใช้ไม่ควรมีความเร็วมากจนเกินไป ควรใช้ความเร็วไม่เกิน 40 กิโลเมตรต่อชั่วโมง หลังจากนั้นจะนำเอาข้อมูลที่ได้ไปคำนวณค่าไฟที่ผู้บริโภคต้องชำระต่อไป

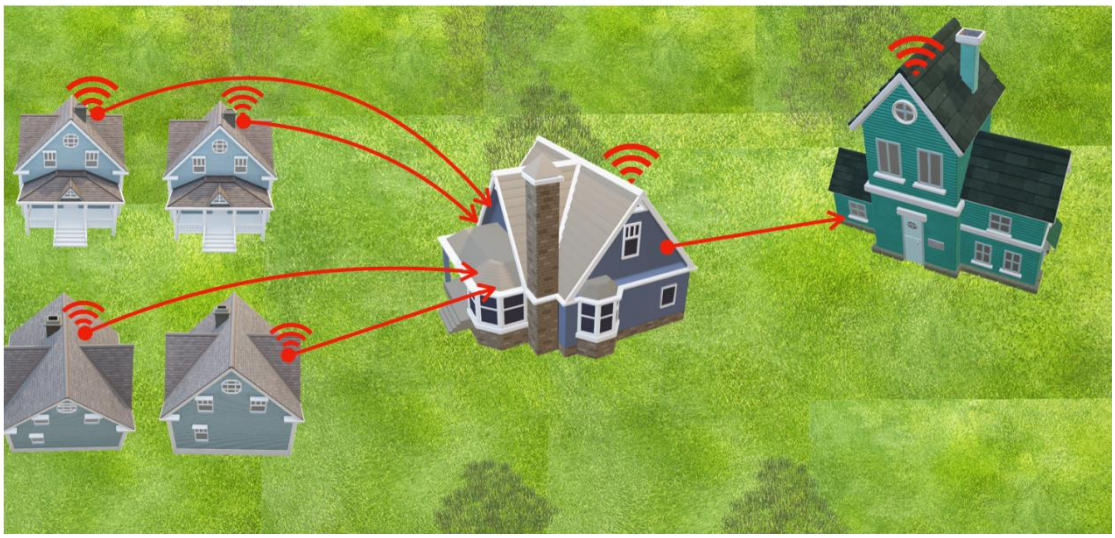
2. แบบ Cluster Tree เหมาะกับการนำไปใช้ในคอนโด อาคาร หรือหอพักดังภาพ 5.3



ภาพ 5.3 ภาพจำลองการนำแบบ Cluster Tree มาใช้ในหอพัก

จากภาพ 5.3 สามารถอธิบายได้ดังนี้ คอนโด อาคาร หรือหอพักต่าง ๆ ที่มีความสูงมากและในแต่ละชั้นนั้นมีห้องอยู่เป็นจำนวนมาก ซึ่งถ้าใช้การเชื่อมต่อแบบ Point to Point นั้น ในกรณีที่มีออฟฟิศอยู่ชั้นล่างสุดแน่นอนว่าข้อมูลจากชั้นบนๆจะไม่สามารถรับ-ส่งข้อมูลได้ เนื่องจากข้อจำกัดทางด้านความไกลของการรับ-ส่งสัญญาณอุปกรณ์ XBee จึงควรมีการเพิ่มตัวกลางในการช่วยขยายสัญญาณให้สามารถรับ-ส่งข้อมูลได้ไกลยิ่งขึ้น โดยการใช้อุปกรณ์กลุ่ม 2 (บอร์ด Arduino และ XBee หน้าที่ Router) โดยจะออกแบบให้ตัวกลางนั้นรวบรวมข้อมูลจากดิจิตอลมิเตอร์ไฟฟ้า เพื่อนำข้อมูลหน่วยไฟฟ้าที่ใช้ไปของทั้งชั้น ส่งข้อมูลไปยังออฟฟิศชั้นล่าง เพื่อคำนวณค่าไฟที่ผู้บริโภคต้องชำระแยกเป็นห้อง ๆ

3. แบบผสมระหว่าง Point to Point กับ Cluster Tree เหมาะกับการนำไปใช้กับคอนโดอาคาร หรือหอพัก และในที่สาธารณะตามความเหมาะสม ดังภาพ 5.4



ภาพ 5.4 ภาพจำลองแบบผสมระหว่าง Point to Point กับ Cluster Tree ในแต่ละหมู่บ้าน

จากภาพ 5.4 สามารถอธิบายได้ดังนี้ บ้านทั้ง 4 หลังเปรียบเสมือนหมู่บ้านเชื่อมต่อกับบ้านศูนย์กลางแบบ Cluster Tree และ บ้านศูนย์กลางเชื่อมต่อกับบ้านหลังสีเขียวแบบ Point to Point ซึ่งเราสามารถทำให้บ้านสีเขียวเชื่อมต่อกับบ้านหลังถัด ๆ ไปได้ตามความเหมาะสมของบริเวณนั้น

ในการบันทึกข้อมูลนั้นควรมีแบบฟอร์มในการบันทึกข้อมูล ดังตาราง 5.1 ดังนี้

ตาราง 5.1 แบบฟอร์มในการจดบันทึกข้อมูลหน่วยไฟฟ้าที่ใช้ไป

วัน-เวลา	ชื่อมิเตอร์	หน่วยไฟฟ้าที่ใช้ไป (KW)	หมายเลขรถ
Ex.			
15/3/2563 14.53 น.	12345678	2550	2
15/3/2563 14.55 น.	12345679	1234	2

จากตาราง 5.1 จะเห็นได้ว่า มีข้อมูลที่สำคัญอยู่ทั้งหมด 4 ข้อมูล ได้แก่ 1. วัน-เวลาที่บันทึก 2. ชื่อมิเตอร์ 3.หน่วยไฟฟ้าที่ใช้ไป 4. หมายเลขรถที่ทำการบันทึกข้อมูล ซึ่งหน่วยไฟฟ้าที่บันทึกมาได้นั้นจะถูกนำไปเปรียบเทียบกับข้อมูลหน่วยไฟฟ้าที่ใช้ไปของเดือนที่ผ่านมา เพื่อคำนวณหาว่าในเดือนปัจจุบันนั้น ผู้บริโภคได้ใช้ไฟฟ้าไปทั้งหมดกี่หน่วย แล้วนำหน่วยไฟฟ้าสุทธิที่ใช้ไปคำนวณค่าไฟฟ้าที่ผู้บริโภคต้องชำระ ซึ่งการที่จะแจ้งให้ผู้บริโภคทราบนั้น จะใช้วิธีการส่งจดหมายไปที่บ้าน หลังจากนั้นให้ผู้บริโภคต้องดำเนินการชำระเงินภายในวันที่กำหนด

5.3 ข้อเสนอแนะ

5.3.1 ในการศึกษาการประยุกต์ใช้อุปกรณ์สัญญาณไร้สาย IOT ในการบันทึกข้อมูลดิจิตอลมิเตอร์ไฟฟ้านั้นอาจใช้อุปกรณ์รับ-ส่งสัญญาณอื่นนอกเหนือจาก โปรโตคอล Zigbee ได้ เช่น Z-Wave ซึ่งมาตรฐานปิดโดย ilicon Labs ซึ่งไม่จำเป็นต้องใช้ Device Hub เข้ามาช่วย (ทำให้อุปกรณ์ต่างมาตรฐาน ยี่ห้อ หรือต่างชนิดสามารถทำงานเชื่อมต่อกันได้) โดยอุปกรณ์ที่ใช้ Z-wave จะต้องได้ตามมาตรฐานที่กำหนดไว้เพื่อหลีกเลี่ยงปัญหาที่เกิดขึ้นกับ อุปกรณ์ใน Zigbee ที่ไม่ยอมคุยกัน

5.3.2 ZigBee จะใช้ย่านความถี่ 2.4 GHz ในขณะที่ Z-Wave จะใช้ย่านความถี่แตกต่างกันไปในแต่ละประเทศซึ่งก็กลายเป็นข้อดีเพราะสัญญาณจะไม่ไปรบกวนกับการใช้งานอื่น ดังนั้นหากเราไม่มีการนำอุปกรณ์เหล่านั้นข้ามประเทศ Z-Wave จะเหมาะสมกว่า

5.3.3 อุปกรณ์ที่ใช้นั้นเป็นอุปกรณ์อิเลกทรอนิกส์ ซึ่งเสียหายได้ง่าย ควรมีอุปกรณ์ป้องกัน เช่น หากล่องมาใส่ เพื่อป้องกันละอองน้ำ ความชื้น ความร้อน และปัจจัยอื่นๆ

5.3.4 ในการทดสอบทั้ง 3 รูปแบบนั้นไม่สามารถทำให้บอร์ด Arduino กับ XBee อยู่ใน Sleep Mode ได้ เพราะฉะนั้น จะเกิดการรับ-ส่งข้อมูลกันอยู่ตลอดเวลาซึ่งเหมาะกับการทำจะนำไปใช้ในกลุ่มหอพักหรืออาคารต่างๆ ที่ต้องการทราบว่าห้องใดใช้ไฟมากน้อยเพียงใด แต่ถ้าเป็น Sleep Mode การรับ-ส่งข้อมูลจะเกิดแค่เพียงตอนที่ผู้ใช้ต้องการที่จะทราบข้อมูลเท่านั้น ซึ่งจะเหมาะกับกลุ่มเจ้าหน้าที่อ่านมาตรวัดการไฟฟ้า ซึ่งต้องการข้อมูลแค่ช่วงสิ้นเดือนเพียงครั้งเดียว

5.3.5 อุปกรณ์บอร์ด Arduino กับ XBee ที่ใช้ในการทดสอบนั้นใช้แหล่งจ่ายไฟเป็นถ่านขนาด 9 V. ซึ่งใช้งานได้ไม่นานเท่าที่ควร ประกอบกับตัวบอร์ดไม่ได้ทำการ Sleep Mode ด้วยจึงส่งผลให้ใช้พลังงานเยอะ จึงควรหาแหล่งจ่ายไฟใหม่ให้กับบอร์ด Arduino เช่น แปลงไฟจากตัวดิจิตอลมิเตอร์ต่อเข้ากับบอร์ด Arduino ซึ่งจะลดการทำงานในการเปลี่ยนถ่านซึ่งใช้งานได้ไม่นานเท่าที่ควร ประกอบกับตัวบอร์ดไม่ได้ทำการ Sleep Mode ด้วยจึงส่งผลให้ใช้พลังงานเยอะ จึงควรหาแหล่งจ่ายไฟใหม่ให้กับบอร์ด Arduino เช่น แปลงไฟจากตัวดิจิตอลมิเตอร์ต่อเข้ากับบอร์ด Arduino ซึ่งจะช่วยลดปัญหาการทำงานมากขึ้นจากในการเปลี่ยนถ่านบ่อยๆดิจิตอลมิเตอร์ต่อเข้ากับบอร์ด Arduino ซึ่งจะลดการทำงานในการเปลี่ยนถ่านดิจิตอลมิเตอร์ต่อเข้ากับบอร์ด Arduino ซึ่งจะลดการทำงานในการเปลี่ยนถ่านดิจิตอลมิเตอร์ต่อเข้ากับบอร์ด Arduino ซึ่งจะลดการทำงานในการเปลี่ยนถ่านดิจิตอลมิเตอร์ต่อเข้ากับบอร์ด Arduino ซึ่งจะลดการทำงานในการเปลี่ยนถ่านดิจิตอลมิเตอร์ต่อเข้ากับบอร์ด Arduino ซึ่งจะลดการทำงานในการเปลี่ยนถ่านดิจิตอลมิเตอร์ต่อเข้ากับบอร์ด Arduino ซึ่งจะลดการทำงานในการเปลี่ยนถ่านดิจิตอลมิเตอร์ต่อเข้ากับบอร์ด Arduino ซึ่งจะลดการทำงานในการเปลี่ยนถ่านดิจิตอลมิเตอร์ต่อเข้ากับบอร์ด Arduino ซึ่งจะลดการทำงานในการเปลี่ยนถ่านดิจิตอลมิเตอร์ต่อเข้ากับบอร์ด Arduino ซึ่งจะลดการทำงานในการเปลี่ยนถ่านดิจิตอลมิเตอร์ต่อเข้ากับบอร์ด Arduino ซึ่งจะลดการทำงานในการเปลี่ยนถ่าน

5.4 ปัญหาและแนวทางการแก้ไข

ปัญหาที่พบและวิธีการดำเนินการแก้ไขสามารถสรุปได้ดังตาราง 5.2

ตาราง 5.2 ปัญหาและแนวทางการแก้ไข

ปัญหาอุปสรรค	แนวทางการแก้ไข
1. XBee Coordinator สามารถส่งข้อมูลจากโปรแกรม Arduino IDE ไปยัง XBee Router ได้ แต่ไม่สามารถรับข้อมูลจาก XBee Router กลับมาได้	ทำการตรวจสอบค่าต่างๆที่สำคัญใน XBee ทั้งสองตัว เช่น ค่า Pan ID , DH, DL ในโปรแกรม X-CTU อีกครั้ง
2. โปรแกรม Arduino IDE ได้รับข้อมูลจากดิจิตอลมิเตอร์ไฟฟ้า ได้เพียงบางครั้งจากการเรียกขอข้อมูลหลาย ๆ ครั้ง	ทำการศึกษาจึงพบว่าตัวขอรับข้อมูลกับตัวส่งข้อมูล มีค่าความถี่เกินไปจึงส่งผลให้ คำขอรับข้อมูลจาก Coordinator ชนกับตัวส่งข้อมูลของ Router
3. XBee ตำแหน่งกลาง(ระหว่างคอมพิวเตอร์กับดิจิตอลมิเตอร์) ที่เพิ่มเข้ามาภายหลัง เพื่อเพิ่มระยะในการรับ-ส่งข้อมูล มีความบกพร่องส่งผลให้ข้อมูลมีค่าที่ผิดเพี้ยนไป	สลับพอร์ทบนบอร์ดทั้งสองพอร์ทจาก USB ให้เป็น XBee
4. ไม่สามารถตั้งค่าให้ บอร์ด Arduino และ XBee อยู่ใน Sleep Mode ได้ เนื่องจากไม่ทราบขั้นตอนในการปลุกให้บอร์ด Arduino และ XBee กับมาทำงานได้ หลังจากที่ทำการ Sleep Mode ไปแล้ว	

บรรณานุกรม

ณัฐกณฑ์ ชมพูพัทธิพงศ์. (2562). ความแตกต่างระหว่างโปรโตคอล ZigBee และ Z-Wave. ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ทวีป ตรีหะจินดารัตน์ ทศพร ปั้นจาด และปวรัชฎ์ คชรินทร์. (2559). เรื่องอินเตอร์เน็ตกับทุกสิ่งของสวนอัจฉริยะ มหาวิทยาลัยศรีนครินทรวิโรฒ

ประภาส สุวรรณเพชร. (2560). เรียนรู้และลองเล่น Arduino เบื้องต้น วิทยาลัยเทคนิคชัยภูมิ “พื้นฐานการสื่อสารด้วยโมดูล Xbee” [ระบบออนไลน์]. แหล่งที่มา http://www.arduino.codemobiles.com/article/13/พื้นฐานการสื่อสารด้วยโมดูล-xbee-part1-เริ่มต้นกับ-xbee-2 (03/09/2562)

ภาณุพงศ์ คงประเสริฐ, พิชามญชุ์ บุญประคอง, ศุภลักษณ์ ศรีสมบัติ, พรเทพ เจ๊กคา. (2556). การออกแบบและสร้างเครื่องมือวัดไฟฟ้าโดยใช้โปรแกรม LabVIEW มหาวิทยาลัยสยาม

ภาคภูมิ มโนยุทธ มัลลิกา อุณหวิวรรธน์ และ วรรณรัช สันติอมรทัต. 2553. ระบบเครือข่ายเซนเซอร์ไร้สายและการต่ออุปกรณ์เสริมเพื่อใช้ในสวนยางพารา. ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ มหาวิทยาลัยสงขลานครินทร์.

สุวิทย์ ภูมิฤทธิกูลและปานวิทย์ ธุวะนุติ.2559. Internet of Thing เพื่อการเฝ้าระวังและเตือนภัยต่อสุขภาพของมนุษย์ และการวิเคราะห์ข้อมูลที่ได้โดยใช้โปรแกรม Hadoop หลักสูตรวิทยาศาสตรบัณฑิต สาขาวิชาเทคโนโลยีสารสนเทศ สถาบันเทคโนโลยีปทุมวัน และคณะเทคโนโลยีสารสนเทศ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง.

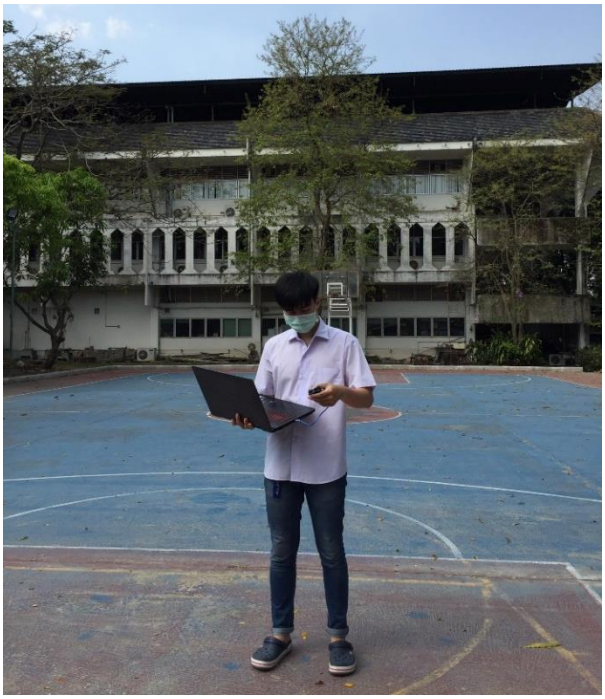
“ Arduino Sleep Modes and How to use them to Save the Power” [ระบบออนไลน์]. แหล่งที่มา https://circuitdigest.com/microcontroller-projects/arduino-sleep-modes (12/01/63)

“Xbee Basic Configuration in Network Application” [ระบบออนไลน์]. แหล่งที่มา https://www.thaieasyelec.com/article-wiki/embedded-electronics-application/xbee-basic-configuration-in-network-application.html (10/12/62)

“XBee-PRO Article (Thai) Chapter 1 ” [ระบบออนไลน์]. แหล่งที่มา https://issuu.com/innovativeexperiment/docs/tpe_xbee-pro/3 (10/12/62)

ภาคผนวก ก

การทดสอบการใช้งานจริง



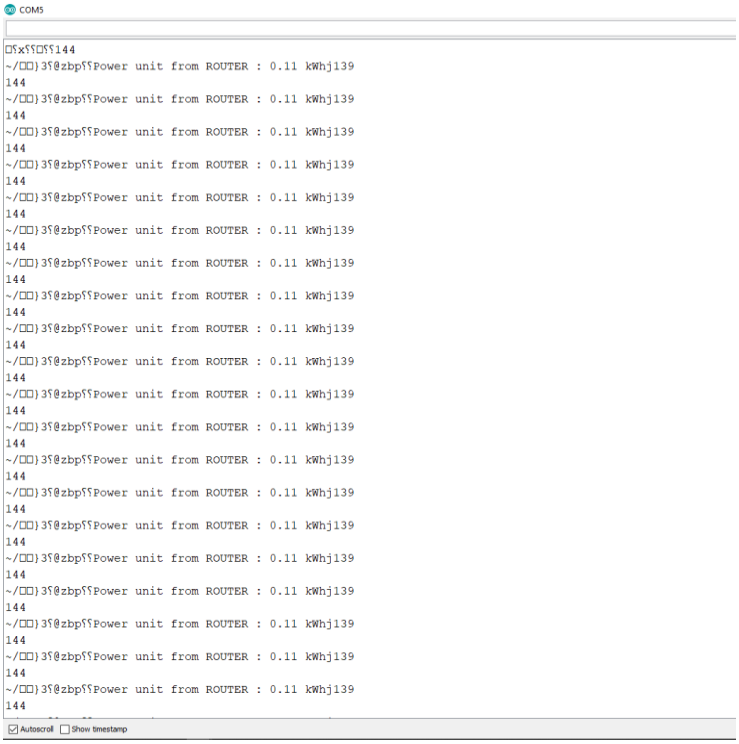
ภาพ ก-1 อ่านค่าที่ถูกส่งจากดิจิตอลมิเตอร์ไฟฟ้ามายังคอมพิวเตอร์

ภาพ ก-2 XBee ตัวกลางถูกนำมาต่อเข้ากับแหล่งจ่ายไฟชั่วคราว (คอมพิวเตอร์)

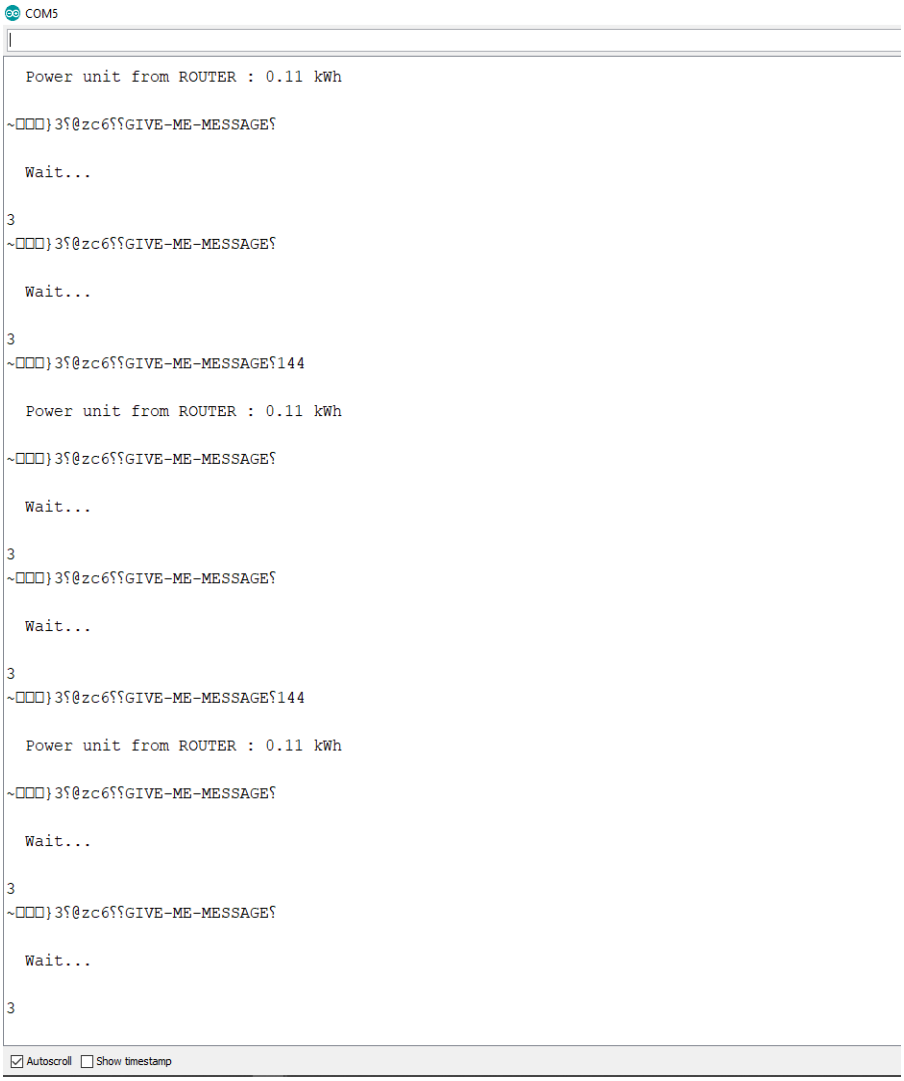
ภาคผนวก ข

ค่าที่อ่านได้บนArduino IDE เทียบกับดิจิตอลมิเตอร์ไฟฟ้า

ข-1 ค่าที่แสดงบนมาตรวัดมิเตอร์ไฟฟ้า



ข-2 ค่าที่แสดงบนโปรแกรม Arduino IDE ของ XBee ตำแหน่งตัวกลาง



ข-3 ค่าที่แสดงบนโปรแกรม Arduino IDE บนคอมพิวเตอร์

ประวัติผู้เขียน

ชื่อสกุล : นายภัทรเชษฐ์ สุจิรานนท์

รหัสนักศึกษา : 590610319

วัน เดือน ปี เกิด : 4 สิงหาคม 2539

ประวัติการศึกษา : กำลังศึกษาระดับอุดมศึกษา คณะวิศวกรรมศาสตร์
สาขาวิศวกรรมอุตสาหการ มหาวิทยาลัยเชียงใหม่
สำเร็จการศึกษามัธยมศึกษาตอนปลาย โรงเรียนจักรคำคณาทร จังหวัดลำพูน
สำเร็จการศึกษามัธยมศึกษาตอนต้น โรงเรียนจักรคำคณาทร จังหวัดลำพูน

ที่อยู่ปัจจุบัน : 162/4 หมู่ 6 ตำบล เหมืองง่า อำเภอ เมือง จังหวัด ลำพูน 51000

อีเมล : pattarachet.s@gmail.com

ชื่อสกุล : นาย ศุภวิชญ์ สถิตวาณิช

รหัสนักศึกษา : 590610345

วัน เดือน ปี เกิด : 24 กรกฎาคม 2540

ประวัติการศึกษา : กำลังศึกษาระดับอุดมศึกษา คณะวิศวกรรมศาสตร์
สาขาวิศวกรรมอุตสาหการ มหาวิทยาลัยเชียงใหม่
สำเร็จการศึกษามัธยมศึกษาตอนปลาย โรงเรียนสามัคคีวิทยาคม จังหวัดเชียงราย
สำเร็จการศึกษามัธยมศึกษาตอนต้น โรงเรียนสามัคคีวิทยาคม จังหวัดเชียงราย

ที่อยู่ปัจจุบัน : 386 หมู่ 8 บ้านห้วยหมอเฒ่า ตำบล เจดีย์หลวง อำเภอ แม่สรวย จังหวัด เชียงราย 57180

อีเมล : supavich_s@cmu.ac.th