

การพัฒนากระบวนการหาตำแหน่งวัตถุด้วยระบบเครือข่ายเซ็นเซอร์ไร้สายโดยใช้โปรโตคอลซิกบี

นางสาวจุฑาทิพย์ เจริญชาศรี

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรมหาบัณฑิต

สาขาวิชาวิทยาศาสตร์คอมพิวเตอร์ ภาควิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย

ปีการศึกษา 2555

ลิขสิทธิ์ของจุฬาลงกรณ์มหาวิทยาลัย

บทคัดย่อและแฟ้มข้อมูลฉบับเต็มของวิทยานิพนธ์ตั้งแต่ปีการศึกษา 2554 ที่ให้บริการในคลังปัญญาจุฬาฯ (CUIR)

เป็นแฟ้มข้อมูลของนิสิตเจ้าของวิทยานิพนธ์ที่ส่งผ่านทางบัณฑิตวิทยาลัย

The abstract and full text of theses from the academic year 2011 in Chulalongkorn University Intellectual Repository (CUIR)

are the thesis authors' files submitted through the Graduate School.

DEVELOPMENT OF OBJECT LOCALIZATION SYSTEM BY WIRELESS SENSOR  
NETWORK USING ZIGBEE PROTOCOL

Miss Jutathip Jaroenchasri

A Thesis Submitted in Partial Fulfillment of the Requirements  
for the Degree of Master of Science Program in Computer Science

Department of Computer Engineering

Faculty of Engineering

Chulalongkorn University

Academic Year 2012

Copyright of Chulalongkorn University

หัวข้อวิทยานิพนธ์	การพัฒนาระบบการหาตำแหน่งวัตถุด้วยระบบเครือข่าย เซ็นเซอร์ไร้สายโดยใช้โปรโตคอลซิกบี
โดย	นางสาวจุฑาทิพย์ เจริญชาติศรี
สาขาวิชา	วิทยาศาสตร์คอมพิวเตอร์
อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก	ผู้ช่วยศาสตราจารย์ ดร.เศรษฐา ปานงาม
อาจารย์ที่ปรึกษาวิทยานิพนธ์ร่วม	ผู้ช่วยศาสตราจารย์ ดร.มานิช โลหเตปานนท์

คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย อนุมัติให้หัวข้อวิทยานิพนธ์ฉบับนี้เป็นส่วน  
หนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรบัณฑิต

..... คณบดีคณะวิศวกรรมศาสตร์  
(รองศาสตราจารย์ ดร.บุญสม เลิศธีรวัฒน์)

คณะกรรมการสอบวิทยานิพนธ์

..... ประธานกรรมการ  
(ศาสตราจารย์ ดร.ประภาส จงสถิตย์วัฒนา)

..... อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก  
(ผู้ช่วยศาสตราจารย์ ดร.เศรษฐา ปานงาม)

..... อาจารย์ที่ปรึกษาวิทยานิพนธ์ร่วม  
(ผู้ช่วยศาสตราจารย์ ดร.มานิช โลหเตปานนท์)

..... กรรมการภายนอกมหาวิทยาลัย  
(ดร.ราชพร เขียนประสิทธิ์)

จุฑาทิพย์ เจริญชาติศรี : การพัฒนาระบบการหาตำแหน่งวัตถุด้วยระบบเครือข่ายเซ็นเซอร์ไร้สายโดยใช้โปรโตคอลซิกบี. (DEVELOPMENT OF OBJECT LOCALIZATION SYSTEM BY WIRELESS SENSOR NETWORK USING ZIGBEE PROTOCOL)

อ.ที่ปรึกษาวิทยานิพนธ์หลัก : ผศ.ดร. เศรษฐา ปานงาม,

อ.ที่ปรึกษาวิทยานิพนธ์ร่วม : ผศ.ดร. มาโนช โลหเตปานนท์ , 162 หน้า.

วิทยานิพนธ์นี้พัฒนาเครื่องมือสำหรับการค้นหาตำแหน่งวัตถุแบบภายในอาคารซึ่งระบบมีประโยชน์สำหรับผู้ใช้งานในแง่ของการค้นหาและตอบตำแหน่งวัตถุแบบพิกัดและแสดงผลออกทางแผนภาพจำลองได้อย่างทันการณ์ โดยหลักการคือ ผู้วิจัยเก็บค่าความแรงสัญญาณ(RSSI)ที่วัดได้จากการสื่อสารของอุปกรณ์ซิกบีที่เกิดขึ้นระหว่างโหนดตัวส่งกับโหนดตัวรับผ่านคลื่นสัญญาณวิทยุ และนำค่าสัญญาณมาคำนวณหาตำแหน่งของวัตถุด้วยอัลกอริทึม Maximum Likelihood และ Min-Max เพื่อให้ได้พิกัดตำแหน่งแกน X และแกน Y จากนั้นจะทำการเปรียบเทียบความแม่นยำระหว่างผลการหาตำแหน่งทั้งอัลกอริทึม Maximum Likelihood และ Min-Max กับอัลกอริทึมของบริษัทผู้ผลิตชุดพัฒนา ซึ่งเป็นโปรแกรมสำเร็จรูป ผู้วิจัยทำการทดลองแบบภายในและภายนอกอาคารด้วยการวางโหนดตัวส่งให้อยู่ในระยะทำการสูงสุดของอุปกรณ์คือ ระยะที่ไม่เกิน 10 เมตร ผลที่ได้คือ แบบภายนอกอาคารอัลกอริทึม Maximum Likelihood มีความแม่นยำในการตอบตำแหน่งมากที่สุด พิจารณาจากค่าความผิดพลาดเฉลี่ยน้อยสุดเท่ากับ 0.81 เมตรและแบบภายในอาคารอัลกอริทึม Min-Max มีค่าความผิดพลาดเฉลี่ยเท่ากับ 1.53 เมตร การค้นหาแบบภายนอกอาคารมีความแม่นยำกว่าแบบภายในอาคาร เนื่องมาจากแบบภายในมีสิ่งรบกวนต่อสัญญาณมากกว่าแบบภายนอกอาคารจึงเป็นเหตุให้ค่าสัญญาณที่ได้รับคลาดเคลื่อนไปจากค่าสัญญาณจริง เมื่อนำสัญญาณมาคำนวณหาตำแหน่งย่อมส่งผลต่อความคลาดเคลื่อนในการตอบพิกัดของวัตถุด้วย

ภาควิชา.....วิศวกรรมคอมพิวเตอร์.....ลายมือชื่อนิสิต.....  
 สาขาวิชา.....วิทยาศาสตร์คอมพิวเตอร์.....ลายมือชื่อ อ.ที่ปรึกษาวิทยานิพนธ์หลัก.....  
 ปีการศึกษา.....2555.....ลายมือชื่อ อ.ที่ปรึกษาวิทยานิพนธ์ร่วม.....

## 5271410921 : MAJOR COMPUTER SCIENCE

KEYWORDS : RECEIVED SIGNAL STRENGTH INDICATOR / WIRELESS SENSOR NETWORK ZIGBEE PROTOCOL / MAXIMUM LIKELIHOOD ALGORITHM / MINMAX ALGORITHM / EUCLIDIANCE DISTANCE

JUTATHIP JAROENCHASRI : DEVELOPMENT OF OBJECT LOCALIZATION SYSTEM BY WIRELESS SENSOR NETWORK USING ZIGBEE PROTOCOL.

ADVISOR : ASST. PROF. SETHA PAN-NGUM, Ph.D.,

CO-ADVISOR : ASST. PROF. MANOJ LOHATEPANONT, Ph.D., 162 pp.

This thesis is to develop a tool to finding indoor objects which could be useful for users in locating object by identifying its X and Y position and displaying it on a map in real time. We collected Received Signal Strength Indicator (RSSI) from communication between reference and blind nodes in Zigbee Wireless Sensor Network .Object position was calculated using Maximum Likelihood Algorithm and Min-Max Algorithm which will give us the X and Y axis position. Localization accuracy was compared between the 2 algorithms and the built-in algorithm by the device manufacturer .We set an experiments indoor and outdoor with the reference nodes positioned within 10 meters from the blind node, which is operating range limit. For outdoor result, the most accurate algorithm is Maximum Likelihood with error of 0.81 meter. For indoor result, the most accurate algorithm is Min-Max with an error around 1.53 meter. Outdoor result has smaller error than Indoor because there are more physical factors inside building which affect signal transmission. The effect causes inaccuracy in position calculation.

Department : ..Computer.Engineering..... Student's Signature .....

Field of Study : Computer.Science..... Advisor's Signature .....

Academic Year : .....2012..... Co-advisor's Signature.....

## กิตติกรรมประกาศ

ข้าพเจ้าขอขอบพระคุณ ผู้ช่วยศาสตราจารย์ ดร.เศรษฐา ปานงาม อาจารย์ที่  
 ปรึกษาวิทยานิพนธ์ ที่เสียสละเวลาช่วยเหลือให้คำปรึกษา คำแนะนำและข้อคิดเห็นที่มีประโยชน์  
 ทำให้การจัดทำวิทยานิพนธ์นี้สำเร็จลุล่วงไปด้วยดี ขอขอบพระคุณคณาจารย์ทุกท่านที่แนะนำ  
 สั่งสอน และให้ความรู้แก่ข้าพเจ้าตลอดระยะเวลาการศึกษา

ขอขอบพระคุณ ศาสตราจารย์ ดร.ประภาส จงสถิตยวิวัฒนา ประธานกรรมการสอบ  
 วิทยานิพนธ์ ที่เสียสละเวลาช่วยเหลือให้คำปรึกษา และให้ความรู้ในแง่ของการใช้ความคิด สอนให้  
 รู้จักวิเคราะห์ รวมทั้งคำแนะนำและข้อคิดเห็นที่มีประโยชน์ พร้อมทั้งให้ทุนสนับสนุนในการจัดหา  
 และพัฒนาโครงการได้อย่างสำเร็จเสร็จสิ้นและสมบูรณ์

ขอขอบพระคุณ ผู้ช่วยศาสตราจารย์ ดร.มานิช โฉมเตพานนท์ อาจารย์ที่ปรึกษา  
 วิทยานิพนธ์ร่วม ที่เสียสละเวลาช่วยเหลือให้คำปรึกษา คำแนะนำและข้อคิดเห็นที่มีประโยชน์ และ  
 ให้ทุนสนับสนุนในการจัดหาอุปกรณ์เพื่อมาพัฒนาระบบให้สำเร็จลุล่วงไปด้วยดี และให้ความรู้แก่  
 ข้าพเจ้าตลอดระยะเวลาการศึกษา

ขอขอบพระคุณ ดร.ราชพร เขียนประสิทธิ์กรรมการสอบวิทยานิพนธ์จาก ศูนย์  
 เทคโนโลยีอิเล็กทรอนิกส์และคอมพิวเตอร์แห่งชาติ (เนคเทค) ที่กรุณาให้ คำแนะนำต่างๆ ทำให้  
 วิทยานิพนธ์นี้มีความถูกต้องและสมบูรณ์มากขึ้น

ขอขอบพระคุณ คุณพ่อ คุณแม่ และน้องชาย น้องสาวที่ให้ความรัก ความห่วงใย  
 คอยให้กำลังใจ พร้อมทั้งให้ความช่วยเหลือ และสนับสนุนในด้านค่าใช้จ่ายในการศึกษาจนสำเร็จ  
 ลุล่วงได้

ขอขอบพระคุณ หัวหน้าฝ่ายเทคโนโลยีสารสนเทศ ธนาคารเพื่อการส่งออกและ  
 นำเข้าแห่งประเทศไทย และพี่ ๆ เพื่อน ๆ ทุกคน สำหรับคำปรึกษาที่ดีและกำลังใจรวมทั้งความ  
 ช่วยเหลือทุกด้านที่มอบให้ข้าพเจ้ามาโดยตลอด

งานวิทยานิพนธ์นี้ ข้าพเจ้าหวังเป็นอย่างยิ่งว่าจะเป็นประโยชน์ต่อผู้ศึกษา ค้นคว้า  
 และสนใจ หากผิดพลาดประการใด ข้าพเจ้าขออภัยไว้เพื่อนำไปใช้ในการปรับปรุงให้  
 วิทยานิพนธ์นี้มีความสมบูรณ์ยิ่งขึ้น สำหรับความดีที่ได้รับจากวิทยานิพนธ์นี้ข้าพเจ้าขอมอบให้แก่  
 ผู้มีพระคุณทุกท่าน

## สารบัญ

	หน้า
บทคัดย่อภาษาไทย.....	ง
บทคัดย่อภาษาอังกฤษ.....	จ
กิตติกรรมประกาศ.....	ฉ
สารบัญ.....	ช
สารบัญตาราง.....	ฎ
สารบัญภาพ.....	ฐ
บทที่	
<b>1 บทนำ.....</b>	<b>1</b>
1.1 ความเป็นมาและความสำคัญของปัญหา.....	1
1.2 วัตถุประสงค์ของการวิจัย.....	2
1.3 ขอบเขตของการวิจัย.....	2
1.4 วิธีดำเนินการวิจัย.....	2
1.5 ประโยชน์ที่คาดว่าจะได้รับ.....	4
1.6 ผลงานตีพิมพ์.....	4
<b>2 ทฤษฎีและงานวิจัยที่เกี่ยวข้อง.....</b>	<b>5</b>
2.1 ทฤษฎีที่เกี่ยวข้องกับงานวิจัย.....	5
2.1.1 เครือข่ายเซ็นเซอร์ไร้สายซิกบี(Zigbee) .....	5
2.1.2 ซิกบีและมาตรฐาน IEEE 802.15.4 ของระดับชั้นเน็ตเวิร์ค.....	7
2.1.3 คุณลักษณะมาตรฐาน IEEE 802.15.4 ของระดับชั้น PHY.....	8
2.1.4 ช่องทางการสื่อสาร (Channel Assignment) .....	8
2.1.5 ตัวตรวจจับพลังงาน (Energy Detection) .....	9
2.1.6 การควบคุมการรับส่งข้อมูล (Carrier Sense) .....	9
2.1.7 ตัวชี้วัดคุณภาพของการเชื่อมโยง (Link Quality Indicator) .....	9
2.1.8 ตรวจสอบพร้อมทั้งเตรียมช่องทาง (Clear Channel Assessment) .....	9
2.1.9 ระดับชั้นโครงข่ายไร้สาย (The Zigbee Network Layer) .....	9
2.1.10 Broadcast .....	9

บทที่	หน้า
2.1.11 Multicasting.....	10
2.1.12 Unicast.....	11
2.1.13 Many –to-one Communication.....	11
2.1.14 การสื่อสารทิศทางตรงหรือ Line of Sight (LOS) .....	12
2.1.15 การสื่อสารทิศทางอ้อมหรือ Non Line of Sight (NLOS) .....	12
2.2 อุปกรณ์สำหรับระบบตัวตน.....	13
2.2.1 อินฟราเรด (IR).....	13
2.2.2 ความถี่วิทยุ (RF) .....	14
2.2.2.1 ระบบการชี้เฉพาะด้วยคลื่นความถี่วิทยุ (RFID) .....	14
2.2.2.2 อัลตราไวด์แบนด์ (UWB) .....	14
2.2.2.3 วิทยุฟาย(Wi-Fi) .....	14
2.2.2.4 Zigbee.....	15
2.3 หลักการหาตำแหน่งวัตถุ.....	16
2.3.1 หลักการที่ใช้พื้นฐานของการคาดคะเน.....	16
2.3.2 หลักการที่ใช้พื้นฐานของการหาระยะทาง.....	19
2.4 การวัดค่าความแรงของสัญญาณ (RSSI) .....	21
2.5 การวัดค่าความแรงสัญญาณด้วยวิธีสามเหลี่ยม .....	21
2.5.1 อัลกอริทึมหาตำแหน่งด้วยมุม (AOA).....	23
2.5.2 อัลกอริทึมหาตำแหน่งด้วยเวลาที่มาถึง (TOA).....	25
2.5.3 อัลกอริทึมหาตำแหน่งด้วยเวลาที่มาถึง (TDOA).....	27
2.6 อัลกอริทึมในการระบุตำแหน่งด้วยวิธีการแปลงค่าสัญญาณเป็นระยะทาง.....	30
2.6.1 อัลกอริทึม Maximum Likelihood.....	30
2.6.2 อัลกอริทึม Min-Max .....	32
2.7 ข้อผิดพลาดที่เกิดจากการหาตำแหน่ง.....	34
2.7.1 อัลกอริทึมการหาข้อผิดพลาด.....	35
2.7.2 วิธีการหาข้อผิดพลาดน้อยที่สุด (Nearest Neighbor Methods).....	35
2.7.3 วิธีการหาข้อผิดพลาดน้อยที่สุด 3 ค่า (3- Nearest Neighbor Methods)...	36
2.7.4 วิธีการหาข้อผิดพลาดด้วยสมการยูคลิดีเนียน (Euclidian Distance).....	36
<b>3 การวิเคราะห์และออกแบบระบบ.....</b>	<b>38</b>



บทที่	หน้า
3.1 การวิเคราะห์ระบบ.....	38
3.2 แนวคิดในการพัฒนา.....	39
3.3 การออกแบบระบบ.....	40
3.3.1 ภาพรวมของระบบ.....	40
3.3.2 การออกแบบและปรับเปลี่ยนโปรโตคอลสื่อสารบน Hardware.....	41
3.4 การออกแบบการทำงานของโปรแกรม.....	46
3.5 สถาปัตยกรรมระบบ.....	47
3.6 การปรับเปลี่ยนค่าตัวแปร A , N ให้เหมาะสม.....	48
3.7 การประเมินประสิทธิภาพของการหาตำแหน่งวัตถุ.....	48
<b>4 การออกแบบและพัฒนาเครื่องมือการค้นหตำแหน่งวัตถุ.....</b>	<b>49</b>
4.1 การออกแบบองค์ประกอบของโปรแกรม.....	49
4.2 สภาพแวดล้อมที่ใช้ในการพัฒนาเครื่องมือ.....	50
4.3 การเชื่อมต่ออุปกรณ์.....	51
4.4 หมวดโปรแกรมการทำงานของโปรโตคอล.....	51
4.2.3 หมวดโปรแกรม Configurations .....	52
4.2.4 หมวดแปลงค่า LQI มาเป็น RSSI .....	53
4.2.5 หมวดแปลงค่า RSSI เป็นระยะทาง (Distance) .....	53
4.2.6 การออกแบบโปรแกรมในส่วนแสดงผล .....	56
4.2.6.1 การแสดงผล Log Messages .....	56
4.2.6.2 การแสดงตำแหน่งวัตถุแบบเป็นพิกัด.....	56
4.2.6.3 การแสดงผลบนแผนที่จำลอง (Graphic User Interface).....	56
4.2.7 การปรับแก้ขนาดแผนที่.....	57
<b>5 การทดลอง.....</b>	<b>58</b>
5.1 การทดลองการหาค่าตัวแปรสมการเพื่อใช้ในการหาระยะทางของตำแหน่งวัตถุ.....	59
5.1.1 หาค่า A แบบภายในอาคาร .....	60
5.1.2 หาค่า A แบบภายนอกอาคาร .....	60
5.1.3 หาค่า N แบบภายในอาคาร .....	62
5.1.4 หาค่า N แบบภายนอกอาคาร .....	64

บทที่	ญ หน้า
5.2 การทดลองหาตำแหน่งของวัตถุด้วยค่าความเข้มสัญญาณ (Received Signal Strength Indicator) .....	66
5.2.1 กลุ่มแรกคือ ชนิดที่ระยะห่างระหว่างตัวส่งและตัวรับถูกกำหนดให้อยู่ในระยะไม่เกิน 10 เมตร ( Valid Length ) .....	68
5.2.2 กลุ่มสองคือ ชนิดที่ระยะห่างระหว่างตำแหน่งตัวส่งและตัวรับถูกกำหนดให้อยู่ในระยะเกิน 10 เมตร ( Invalid Length).....	70
5.3 การทดลองหาตำแหน่งของวัตถุด้วยค่าความเข้มสัญญาณ (Received Signal Strength Indicator) .....	71
5.3.1 กลุ่มแรกคือ ชนิดที่ระยะห่างระหว่างตัวส่งและตัวรับถูกกำหนดให้อยู่ในระยะไม่เกิน 10 เมตร ( Valid Length ).....	72
5.3.2 กลุ่มสองคือ ชนิดที่ระยะห่างระหว่างตัวส่งห่างจากตัวรับเพียง 1 ตัวที่ระยะเกิน 10 เมตร ( Invalid Length ) .....	74
5.3.3 กลุ่มสามคือ ชนิดที่ระยะห่างระหว่างตัวส่งห่างจากตัวรับ 2 ตัวที่ระยะเกิน 10 เมตร ( Invalid Length ) .....	75
5.4 ความแม่นยำของการวัดระยะโดยใช้ค่าความแรงสัญญาณ (Received Signal Strength Indicator : RSSI ).....	76
<b>6 สรุปผลการวิจัย อภิปรายผล และข้อเสนอแนะ.....</b>	<b>81</b>
6.1 สรุปผลการวิจัย.....	81
6.2 ผลการเปรียบเทียบความแม่นยำของอัลกอริทึม Maximum Likelihood, Min-Max และ TI .....	82
6.3 ผลการเปรียบเทียบความคงที่ของอัลกอริทึม Maximum Likelihood , Min-Max และ TI .....	83
6.4 สรุปความแม่นยำของการหาตำแหน่งวัตถุแบบภายในและภายนอกอาคาร .....	83
6.5 ผลการเปรียบเทียบประสิทธิภาพในการหาตำแหน่งวัตถุของ อัลกอริทึม Maximum Likelihood และ อัลกอริทึม Min-Max ของระบบกับงานวิจัยอื่น.....	85
6.6 แนวทางการนำไปใช้งาน.....	87
6.7 อภิปรายผล.....	87
6.8 ข้อจำกัด ปัญหาและอุปสรรค.....	89
6.8.1 ข้อจำกัด.....	89

บทที่	หน้า
6.8.2 ปัญหาและอุปสรรค.....	89
6.8.3 ข้อเสนอแนะ.....	89
6.8.4 แนวทางการวิจัยต่อ.....	89
รายการอ้างอิง.....	92
ภาคผนวก .....	96
ภาคผนวก ก ตัวอย่างผลการทดลอง การหาตำแหน่งวัตถุแบบแสดงผลบน GUI Map และพิกัดตำแหน่ง .....	97
ภาคผนวก ข แสดงคำสั่งการเขียนโปรแกรมคอมพิวเตอร์แบ่งเป็น 2 ส่วน	
1.ส่วนที่เป็น Source Code Embedded.....	102
2.ส่วนที่เป็น Source Code .Net.....	134
ภาคผนวก ค คู่มือการใช้งานระบบเบื้องต้น .....	156
ประวัติผู้เขียนวิทยานิพนธ์.....	162

## สารบัญตาราง

ตารางที่		หน้า
2.1	การแบ่งช่วงค่าความถี่ของช่องทาง.....	8
4.1	ตัวอย่างหมวดการแปลงค่า LQI มาเป็น RSSI .....	53
5.1	ค่าเฉลี่ยรวมของตัวแปร A สำหรับใช้หาวัตถุสำหรับภายในอาคาร.....	60
5.2	ค่าเฉลี่ยรวมของตัวแปร A สำหรับใช้หาวัตถุสำหรับภายนอกอาคาร.....	61
5.3	หาค่า N ในระยะทาง 10 เมตรแบบภายในอาคาร.....	62
5.4	หาค่า N ในระยะทาง 10 เมตรแบบภายนอกอาคาร.....	65
5.5	ผลการทดลองหาตำแหน่งวัตถุแบบภายนอกอาคารในระยะต่ำกว่า 10 เมตร....	69
5.6	ผลการทดลองหาตำแหน่งวัตถุแบบภายนอกอาคารในระยะเกิน 10 เมตร.....	70
5.7	ผลการทดลองการหาตำแหน่งวัตถุภายในอาคารในระยะต่ำกว่า 10 เมตร.....	73
5.8	ผลการทดลองการหาตำแหน่งวัตถุภายในอาคารระยะเกิน 10 เมตร (Invalid Length = 1 Reference node).....	74
5.9	ผลการทดลองการหาตำแหน่งวัตถุภายในอาคารระยะเกิน 10 เมตร (Invalid Length = 2 Reference node).....	75
6.1	แสดงการเปรียบเทียบการหาค่าความแม่นยำของทั้ง 3 อัลกอริทึมแบบภายในอาคาร.....	83
6.2	แสดงการเปรียบเทียบการหาค่าความแม่นยำของทั้ง 3 อัลกอริทึมแบบภายนอกอาคาร.....	84
6.3	แสดงการเปรียบเทียบการหาค่าความแม่นยำของทั้ง 3 อัลกอริทึมแบบภายในอาคารแบบ Invalid Length .....	84
6.4	แสดงการเปรียบเทียบการหาค่าความแม่นยำของทั้ง 3 อัลกอริทึมแบบภายนอกอาคารแบบ Invalid Length .....	85
6.5	แสดงการเปรียบเทียบผลการหาตำแหน่งวัตถุระหว่างระบบค้นหาที่ระบบอื่นแบบภายในอาคาร.....	86

## สารบัญภาพ

ภาพที่		หน้า
2.1	เครือข่ายแบบดาว(Star).....	6
2.2	เครือข่ายแบบเพียร์ทูเพียร์(Peer-to-Peer).....	6
2.3	โครงสร้างของซูเปอร์เฟรม.....	7
2.4	ชุดโพรโตคอลของเครือข่ายซีทีบี.....	8
2.5	Broadcast Communication.....	10
2.6	Multicasting Communication.....	10
2.7	Unicast Communication.....	11
2.8	การสื่อสารแบบ N:1.....	11
2.9	Line of Sight.....	12
2.10	การสื่อสารแบบ Non Line of Sight.....	13
2.11	เทคนิคที่ใช้พื้นฐานของการคาดคะเน.....	18
2.12	(a) การวัดระยะทางด้วยเทคนิคสามเหลี่ยมระยะ (Trilateration).....	22
	(b) การประมาณจุดคลาดเคลื่อนของโหนด.....	22
2.13	แสดงการหาตำแหน่งด้วยการวัดแบบ AOA.....	24
2.14	อัลกอริทึมหาตำแหน่งด้วยมุม (AOA).....	24
2.15	แสดงการหาตำแหน่งด้วยการวัดแบบ TOA .....	25
2.16	การคิดระยะทางจากอัลกอริทึมเวลาที่มาถึง TOA .....	27
2.17	แสดงการหาตำแหน่งด้วยการวัดแบบ TDOA.....	29
2.18	แสดงผลต่างของ d บนพื้นฐานของหลักการ TDOA.....	30
2.19	แสดงการหาระยะทาง (d) ระหว่างโหนดอ้างอิง.....	31
2.20	แสดงการหาค่า Min-Max ระหว่างโหนดอ้างอิง.....	33

ภาพที่	หน้า
2.21	แสดงการหาตำแหน่งของวัตถุโดยใช้ Min-Max..... 34
3.1	ภาพรวมของระบบ..... 40
3.2	แสดงการรับ-ส่ง Protocol Messages ของการสื่อสารระหว่างอุปกรณ์ Gateway, Blind node และ Reference node..... 42
3.3	แสดงอธิบายขั้นตอนการรับและส่ง Protocol Messages..... 43
3.4	แสดงอธิบายขั้นตอนการรับและส่ง Protocol Messages ระหว่าง Blind node และ Reference node กับ Gateway..... 44
3.5	รูปแบบโปรโตคอลของ Blind node response messages (0x014)..... 45
3.6	รูปแบบโปรโตคอลของ Reference NODE Response Messages (0x012)..... 45
3.7	รูปแบบโครงสร้างการทำงานของโปรแกรม..... 46
3.8	แสดงการออกแบบสถาปัตยกรรมระบบ..... 47
4.1	แสดงส่วนการเชื่อมต่อของ Hardware กับส่วนของโปรแกรม..... 52
4.2	แสดงขั้นตอนการนำค่า RSSI มาแปลงเป็น Distance หาตำแหน่งด้วย อัลกอริทึม Maximum Likelihood..... 54
4.3	แสดงขั้นตอนการนำค่า RSSI มาแปลงเป็น Distance หาตำแหน่งด้วย อัลกอริทึม Min-Max..... 55
4.4	แสดงค่าสัญญาณ RSSI และ Node หลังจากเชื่อมต่อเข้ามาในระบบได้..... 56
4.5	แสดงตำแหน่งวัตถุแบบแนวพิกัด X, Y..... 56
4.6	แสดงตำแหน่งวัตถุแบบหน้าจอ GUI Map..... 57
4.7	แสดงส่วนการแก้ไขขนาดของแผนที่..... 57
5.1	แสดงการจำลองการจัดวางโหนดตัวรับและตัวส่งเพื่อหาค่าตัวแปร A..... 60
5.2	แสดงการจำลองการจัดวางโหนดตัวรับและตัวส่งเพื่อหาค่าตัวแปร A..... 61
5.3	แสดงภาพการจำลองการวางโหนดตัวส่งห่างจากตัวรับในระยะ 1-10 เมตร..... 62
5.4	แสดงภาพการจำลองการวางโหนดตัวส่งห่างจากตัวรับในระยะ 1-10 เมตร..... 64
5.5	ภาพจำลองการจัดพื้นที่แบบตารางกริด..... 67

ภาพที่	หน้า
5.6	แสดงการทดลองหาตำแหน่งของวัตถุแบบภายนอกอาคาร..... 68
5.7	แสดงผลการเปรียบเทียบค่าความผิดพลาดเฉลี่ยของทั้ง 3 อัลกอริทึมคือ Maximum Likelihood , Min-Max และ TI แบบภายนอกอาคาร..... 71
5.8	แสดงการทดลองหาตำแหน่งของวัตถุแบบภายในอาคาร..... 72
5.9	SmartboardRF และ Chipcon CC2430 กับ CC2431..... 77
5.10	ภาพจำลองการทดลองการจัดวางตำแหน่งของ Blind node และ Reference node..... 77
5.11	กราฟแสดงแนวโน้มค่า RSSI ของโหนดตัวที่ 1 ทุกช่วง 1 เมตรในระยะทาง 10 เมตรภายในอาคาร..... 78
5.12	กราฟแสดงแนวโน้มค่า RSSI ของโหนดตัวที่ 2 ทุกช่วง 1 เมตรในระยะทาง 10 เมตรภายในอาคาร..... 78
5.13	กราฟแสดงแนวโน้มค่า RSSI ของโหนดตัวที่ 3 ทุกช่วง 1 เมตรในระยะทาง 10 เมตรภายในอาคาร..... 79
5.14	กราฟแสดงแนวโน้มค่า RSSI ของโหนดตัวที่ 4 ทุกช่วง 1 เมตรในระยะทาง 10 เมตรภายในอาคาร..... 79
6.1	แสดงการเปรียบเทียบค่าความผิดพลาดเฉลี่ยของผลการหาตำแหน่งวัตถุ ระหว่างระบบของผู้วิจัยเองกับงานวิจัยอื่นแบบภายในอาคาร..... 87

# บทที่ 1

## บทนำ

### 1.1 ความเป็นมาและความสำคัญของปัญหา

ในปัจจุบันจำนวนอุปกรณ์อิเล็กทรอนิกส์ที่ใช้อยู่ในองค์กรมีหลากหลายประเภท บางประเภทมีจำนวนจำกัดหากเกิดกรณีเสียหายชำรุดอย่างฉับพลัน อาทิ อุปกรณ์อิเล็กทรอนิกส์ประเภทเน็ตเวิร์ค จึงมีความจำเป็นอย่างยิ่งที่ต้องเร่งดำเนินการหาอุปกรณ์ตัวใหม่มาทดแทนและทำการกู้คืนระบบเครือข่ายขององค์กรให้สามารถกลับคืนสู่สภาพปกติอย่างรวดเร็วที่สุด หากกรณีอุปกรณ์อิเล็กทรอนิกส์บางประเภทมีชิ้นส่วนประกอบอยู่มากมายและถูกจัดเก็บอย่างกระจัดกระจาย จึงอาจก่อให้เกิดความยากลำบากในการค้นหาและติดตาม รวมทั้งการค้นหาเอกสารประเภทพวกสัญญาค่าทำนุบำรุง (Maintenance / License Agreements) ซึ่งมีความสำคัญมาก เนื่องจากเอกสารดังกล่าวมีอายุสัญญามากกว่า 5 ปีขึ้นไป เมื่อวันและเวลาผ่านไปเอกสารอาจถูกย้ายไปจัดเก็บในคลังเอกสารใหม่ โดยเรียงตามหมวดอักษรแล้ว แต่พบว่ายังอาจเกิดการสูญหายของเอกสารต้นฉบับอยู่ และการค้นหาเป็นไปด้วยความยากลำบากและช้าเนื่องจากเอกสารมีจำนวนมาก ในปัจจุบันนี้มีการนำเทคโนโลยีเข้ามาช่วยในการค้นหาอุปกรณ์และเอกสารสำคัญ เพื่อช่วยอำนวยความสะดวกมากยิ่งขึ้น เทคโนโลยีเครือข่ายไร้สายเป็นที่นิยมนำมาใช้ในการติดตามและค้นหาซึ่งประกอบด้วย เทคโนโลยีการชี้เฉพาะด้วยคลื่นความถี่วิทยุ (RFID) มาช่วยระบุเอกลักษณ์ของวัตถุด้วยคลื่นวิทยุ เทคโนโลยีสื่อสารไร้สายส่วนบุคคลระยะสั้น (UWB) คลื่นอินฟราเรด (IR) คลื่นความถี่วิทยุ (RF) เครือข่ายไร้สายวายฟาย (Wi-Fi) และเครือข่ายเซ็นเซอร์ไร้สายซิกบี (Zigbee) ด้วยคุณสมบัติเฉพาะที่ใช้กำลังไฟต่ำประหยัดพลังงานจึงเหมาะสมกับงานเฝ้าระวัง (Monitoring) ที่ใช้เวลานาน ทำให้สามารถใช้งานได้ยาวนานกว่าเทคโนโลยีชนิดอื่นและมีการเชื่อมต่ออย่างซับซ้อนเพื่อรองรับระบบการเชื่อมต่อสำหรับเครือข่ายขนาดใหญ่ และเหมาะสมกับงานการสื่อสารระยะใกล้ 10 – 100 เมตร ทางผู้วิจัยจึงเกิดแนวความคิดที่จะนำเทคโนโลยีนี้มาพัฒนาให้เกิดกระบวนการวิเคราะห์หาตำแหน่งเป็นไปอย่างแม่นยำมากยิ่งขึ้น รวมทั้งสามารถแสดงผลได้แบบทันที



## 1.2 วัตถุประสงค์ของการวิจัย

พัฒนาระบบค้นหาตำแหน่งของวัตถุภายในอาคารด้วยระบบเครือข่ายเซ็นเซอร์ไร้สายโดยใช้โปรโตคอลซิกบี

## 1.3 ขอบเขตการวิจัย

1. ระบบทำงานได้ในพื้นที่ไม่เกิน 100 ตารางเมตร(10x10 ตารางเมตร)เมื่ออยู่ใน พื้นที่โล่งและใช้โหนดตัวรับ (Reference Node) วางเป็นรูปสี่เหลี่ยมจัตุรัสห่างกันด้านละ 10 เมตร โดยวางที่มุมทั้ง 4 ของรูปสี่เหลี่ยมจัตุรัส
2. ระยะระหว่างโหนดอ้างอิงและโหนดเป้าหมายต้องไม่เกิน 10 เมตร
3. สามารถบอกตำแหน่งวัตถุเป็นพิกัดได้ในระดับ 2 มิติ คือ (X,Y) ในทางตรงข้ามไม่สามารถบอกตำแหน่งวัตถุที่อยู่ต่างระดับกันได้ (กรณีมีการติดตั้งตำแหน่งข้ามชั้น)
4. เปรียบเทียบผลการทดลองและนำผลลัพธ์มาประเมินประสิทธิภาพในการระบุตำแหน่งได้จากสมการวัดค่าความคลาดเคลื่อน
5. สามารถนำผลลัพธ์มาแสดงตำแหน่งของวัตถุบน Graphic User Interface แบบทันสมัยได้

## 1.4 วิธีดำเนินการวิจัย

1. ศึกษาทฤษฎีและอัลกอริทึม Maximum Likelihood และ Min-max ในการค้นหาตำแหน่งวัตถุ และทฤษฎีสมการหาค่าความคลาดเคลื่อน (Euclidian Distance)
2. ศึกษาความสัมพันธ์ตัวแปรสำคัญ (ค่า A , N ) ระยะทาง(Distance) และค่าความเข้มของสัญญาณ(Received Signal Strength Indicator)
3. ศึกษาหลักการกระบวนการทำงานของอุปกรณ์และชุดโปรแกรมสำเร็จรูปของบริษัท Texas Instrument ที่เกี่ยวข้องที่ใช้ในการค้นหา ประกอบด้วย ชุดพัฒนา (Development Kit) Smartboard RF04eb, Chipboard , Antenna , Chipcon CC2430 และ Chipcon CC2431 ชุดเครื่องมือพัฒนาสำหรับฝังคำสั่งบนชิป คือ SmartRF Studio

7 และ IAR Embedded Workbench โปรแกรมสำเร็จรูปสำหรับจับสัญญาณ คือ LEMonitor , Packet Sniffer โปรแกรมสำเร็จรูปในการระบุพิกัดตำแหน่ง คือ Z-Location Engine

4. ออกแบบโปรแกรมสำหรับดักจับชุดโปรโตคอลจากการสื่อสารระหว่าง โหนดตัวส่งสัญญาณและโหนดตัวรับสัญญาณจาก Smartboard RF04eb และคัดกรองเลือกเฉพาะโปรโตคอลที่ต้องตัดมาเฉพาะ 1 ตัวแปรคือ ค่าสัญญาณ (Link Quality Indicator : LQI )

5. ออกแบบโมเดลจำลองและโปรแกรมที่มีคุณสมบัติในการแปลงค่าสัญญาณดิบ (LQI) มาเป็นค่าสัญญาณ (RSSI) ที่สามารถคำนวณหาตำแหน่งวัตถุด้วย อัลกอริทึม Maximum Likelihood และอัลกอริทึม Min-Max

6. ออกแบบโครงสร้างโปรแกรมและพัฒนาระบบที่มีคุณสมบัติรับค่าสัญญาณเพื่อแปลงเป็นระยะทางมาคำนวณหาพิกัดตำแหน่งด้วยอัลกอริทึม Maximum Likelihood และ อัลกอริทึม Min-Max

7. จำลองสถานการณ์การทดลองแบบภายในอาคารและภายนอกอาคาร เพื่อใช้ในการหาตำแหน่ง

8. สร้างแหล่งพักข้อมูลและทำการเก็บผลในแต่ละตำแหน่งตามสถานการณ์ที่กำหนด

9. ทดสอบและเปรียบเทียบผลระหว่างพิกัดตำแหน่งของอัลกอริทึม Maximum Likelihood , Min-Max และโปรแกรมสำเร็จรูปของ TI ด้วยสมการ Euclidian Distance

10. สรุปผล วิเคราะห์ผล

11. จัดทำรูปเล่มวิทยานิพนธ์

### 1.5 ประโยชน์ที่คาดว่าจะได้รับ

สามารถพัฒนาระบบการค้นหาและติดตามอุปกรณ์ภายในอาคารได้และให้ผลลัพธ์ที่แม่นยำในการระบุตำแหน่งของอุปกรณ์ รวมทั้งสามารถออกแบบมาแล้วสามารถนำไปใช้ประโยชน์ได้จริงหรือสามารถนำไปประยุกต์ใช้กับงานจำพวก การติดตามวัตถุภายนอกอาคาร

### 1.6 ผลงานตีพิมพ์

ส่วนหนึ่งของวิทยานิพนธ์นี้ได้นำเสนอในการประชุมวิชาการ ดังนี้

1. The 37<sup>th</sup> Congress on Science and Technology of Thailand (STT37 2011), 10 – 12 October 2011, Centara Grand & Bangkok Convention Centre at CentralWorld, Bangkok, Thailand ในบทความเรื่อง Simulation of object Localization System By Wireless Sensor Network Using Zigbee Protocol โดยผู้แต่งคือ Jutathip Jaroenchasri , Setha Pan-ngum และ Manoj Lohatepanont

## บทที่ 2

### ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

#### 2.1 ทฤษฎีที่เกี่ยวข้องกับงานวิจัย

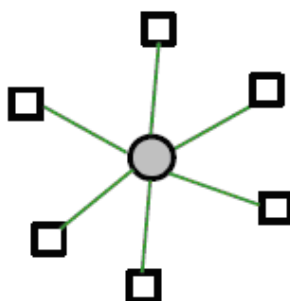
งานวิจัยนี้เกี่ยวข้องกับการนำหลักการทางทฤษฎีที่เกี่ยวข้องอยู่สองสาขาหลัก คือ เครือข่ายเซ็นเซอร์ไร้สายซิกบี (Zigbee) และอัลกอริทึมการหาตำแหน่งของวัตถุ (Location Estimation) ซึ่งมีรายละเอียดดังต่อไปนี้

##### 2.1.1 เครือข่ายเซ็นเซอร์ไร้สายซิกบี (Zigbee)

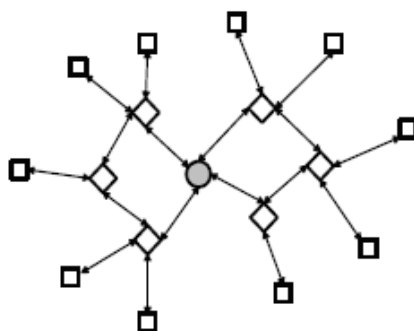
ซิกบี เป็นมาตรฐานที่ใช้กำหนดชุดการสื่อสารของโปรโตคอลสำหรับข้อมูลอัตราต่ำด้วยสัญญาณไร้สายระยะสั้น [1]. อุปกรณ์สัญญาณไร้สายซิกบีจะมีคลื่นความถี่ 868 MHz, 915 MHz และ 2.4 GHz อัตราการส่งข้อมูลสูงสุดคือ 250 บิตต่อวินาที ซิกบีมีเป้าหมายหลักคือการใช้งานแบตเตอรี่ที่ขับเคลื่อนด้วยอัตราการส่งข้อมูลต่ำ จึงทำให้แบตเตอรี่มีอายุการใช้งานที่ยาวนานโดยคุณลักษณะพิเศษคือ Sleep mode ช่วยยืดอายุการใช้งานออกไปได้อีก ซิกบีเป็นเทคโนโลยีไร้สายถูกพัฒนาขึ้นจากกลุ่มสหพันธ์ซิกบี (Zigbee Alliance) ด้วยมาตรฐาน IEEE 802.15.4 ซึ่งทางสหพันธ์ได้ทำการกำหนดมาตรฐานของระดับชั้น Physical Layer (PHY) และ Media Access Control (MAC)[2] ดังนั้นเป็นมาตรฐานสำหรับเครือข่ายไร้สายระยะใกล้ความเร็วต่ำหรือ Low Rate Wireless Personal Area Network (LR-WPAN) [3] ชั้น PHY ที่ถูกกำหนดไว้ในมาตรฐานนี้มีการใช้ 3 ย่านความถี่ คือ 2.4-2.4835 GHz โดยอัตราความเร็วของข้อมูล (bit rate or BR) 250 kbps มีอยู่ 16 ช่องสัญญาณคือ ช่องสัญญาณที่ 11-26, ความถี่ 902 – 928 MHz ที่ BR = 40 kbps มีอยู่ 10 ช่องสัญญาณ คือ ช่องสัญญาณที่ 1-10 และความถี่ 868-868.6 MHz ที่ BR = 20 kbps มีอยู่ 1 ช่องสัญญาณ คือช่องสัญญาณที่ 0 โดยความถี่ 2.4 GHz จัดเป็นช่วงความถี่ที่นิยมใช้กันมากที่สุด

ในเครือข่ายเซ็นเซอร์ไร้สายระยะใกล้ความเร็วต่ำได้แบ่งชนิดอุปกรณ์ของซิกบีออกเป็น 2 ประเภท ดังนี้ Full Function Device (FFD) เป็นเราเตอร์ที่เป็นสื่อกลางในการส่งข้อมูลจากอุปกรณ์อื่น ๆ ใช้พลังงานจาก Power Lineทำงานได้ทุกโทโพโลยีและสามารถทำเป็นจุดเชื่อมต่อกันได้ โดย FFD สามารถที่จะทำงานได้สามโหมดคือ PAN coordinator, Coordinator หรือ Device ซึ่งใน Zigbee Network ต้องมี FFD อย่างน้อยหนึ่งตัวซึ่งจะทำหน้าที่เป็น PAN Coordinator โดย FFD สามารถที่จะติดต่อได้ทั้ง FFD และ RFD แต่ RFD จะสามารถติดต่อได้

เพียง FFD เท่านั้น และ Reduce Function Device (RFD) เหมาะแก่การเชื่อมต่อภายในเครือข่าย ในพลังงานจากแบตเตอรี่ ไม่สามารถถ่ายทอดข้อมูลจากอุปกรณ์อื่น ๆ ได้ ทำได้ง่ายในเครือข่ายที่เป็นแบบสตาร์ (Star) แบบ Logical Device มี 3 ประเภทคือ Zigbee Coordinators เป็นจุดที่ประสานเชื่อมต่อ ทำหน้าที่ในการจัดเก็บข้อมูลในเครือข่าย Zigbee Routers ทำหน้าที่จัดการเส้นทางของข้อความที่ส่งผ่านภายในโครงข่ายระหว่างคู่ของโหนดใด ๆ และซิกบีและอุปกรณ์เป็นโหนดที่อยู่ในส่วนของผู้ใช้งานโดยสามารถเป็นได้ทั้งแบบ RFD และ FFD ซึ่ง FFD สามารถสื่อสารได้กับทุกอุปกรณ์สามารถทำหน้าที่เป็น PAN Coordinator , เจ้าเตอร์ หรืออุปกรณ์ปลายทาง (End Device) ได้ สำหรับ RFD จะถูกใช้ในอุปกรณ์ปลายทางโดยการส่งผ่านของข้อมูลจาก RED หนึ่งไปยังอีก RED หนึ่งซึ่งจะส่งผ่าน FED นอกจากนี้มาตรฐาน IEEE 802.15.4 สามารถรองรับโทโปโลยีแบบต่าง ๆ ได้แก่ การเชื่อมต่อแบบสตาร์ , แบบเพียร์ทูเพียร์ และแบบเมช ดังแสดงในภาพที่ 2.1 และ 2.2 สำหรับโครงสร้างแบบสตาร์ การเชื่อมต่อภายในเครือข่ายจะถูกควบคุมด้วย PAN Coordinator เพียงอันเดียว โดย PAN Coordinator นี้จะเป็นเสมือนผู้ดูแลเครือข่าย ทำหน้าที่จัดการกับโหนดภายในเครือข่าย และเก็บข้อมูลของโหนดต่าง ๆ ในส่วนของเครือข่ายแบบเพียร์ทูเพียร์จะประกอบด้วย PAN Coordinate เช่นเดียวกับแบบสตาร์ การเชื่อมโยงแบบเพียร์ทูเพียร์ทุกอุปกรณ์สามารถสื่อสารกับอุปกรณ์ข้างเคียงได้โดยตรง



ภาพที่ 2.1 เครือข่ายแบบดาว (Star)[4]



ภาพที่ 2.2 เครือข่ายแบบเพียร์ทูเพียร์ (Peer-to-Peer)[4]

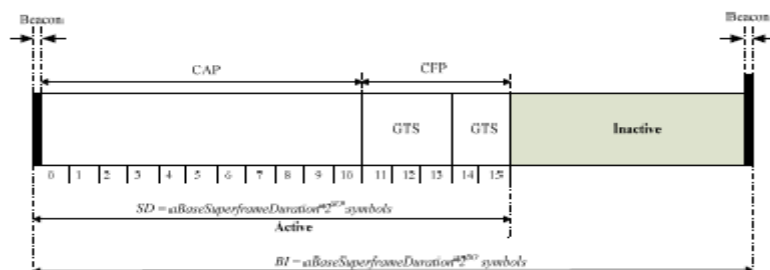
ประกอบด้วยรายละเอียดดังนี้

○ คือ Coordinator

◇ คือ Router

□ คือ End Device

ในการใช้พลังงานให้บรรลุผลและมีประสิทธิภาพที่ดี ชิกปีสามารถทำงานในเครือข่ายแบบ beacon-enabled โดยโครงสร้างของซูเปอร์เฟรม (Superframe) แสดงดังภาพที่ 2.3 (Jelena et al., 2008)

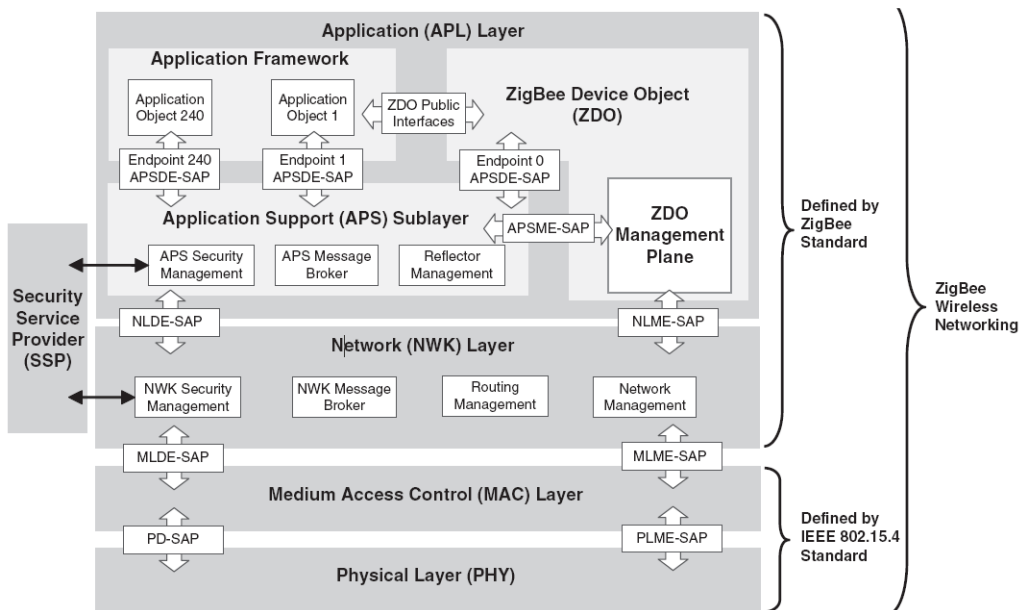


ภาพที่ 2.3 โครงสร้างของซูเปอร์เฟรม[1]

ซูเปอร์เฟรมใช้สำหรับควบคุมการเข้าถึงช่องสัญญาณ ซึ่งถูกสร้างขึ้นโดย Network Coordinator ในการส่งสัญญาณจังหวะ (Beacon) ตามช่วงเวลาที่กำหนดเพื่อใช้ในการเข้าถึงช่องสัญญาณในแต่ละช่วงเวลา (Time Slot) แต่ละซูเปอร์เฟรมจะประกอบด้วยสองส่วนคือ Active และ Inactive ความยาวของ Beacon Interval (BI) และส่วนของ Active ซึ่งอ้างอิงโดย Superframe Duration (SD) ที่ถูกกำหนดโดยสองค่าพารามิเตอร์คือ Beacon Order (BO) และ Superframe Order (SO) ตามลำดับ ในส่วน Active ถูกแบ่งออกเป็น 16 slot ซึ่งประกอบด้วย 3 ส่วนคือ Beacon , Contention Access Period (CAP) และ Contention Free Period (CFP)

### 2.1.2 ชิกปีและมาตรฐาน IEEE 802.15.4 ของระดับชั้นเน็ตเวิร์ค [4]

โพรโตคอลชิกปีของเครือข่ายไร้สายจะขึ้นอยู่กับองค์ประกอบระหว่างประเทศที่มีมาตรฐานได้ดำเนินการเปิดระบบการเชื่อมต่อแบบจำลองอ้างอิง (OSI) ตามแบบพื้นฐาน [1] ซึ่งมีอยู่ด้วยกันทั้งหมด 7 ชั้น ดังภาพที่ 2.4



ภาพที่ 2.4 : ชุดโพรโตคอลของเครือข่ายซิกบี[4]

### 2.1.3 คุณลักษณะมาตรฐาน IEEE 802.15.4 ของระดับชั้น PHY [4]

ซิกบีหรือมาตรฐาน IEEE 802.15.4 กำหนดฟังก์ชันโพรโตคอล PHY Layer และติดต่อกับชั้นระดับ MAC Layer และกำหนดความต้องการของอุปกรณ์ฮาร์ดแวร์ เช่น กำลังรับและกำลังส่งของอุปกรณ์ได้

### 2.1.4 ช่องทางการสื่อสาร (Channel Assignment ) [4]

ค่าความถี่ของแต่ละช่องทางจะถูกกำหนดและแบ่งด้วยความถี่ดังต่อไปนี้

Channel Page	Channel Number	Description
	0	868 MHz band (BPSK)
0	1-10	915 MHz band (BPSK)
	11-26	2.4GHz band (O-QPSK)
	0	868 MHz band (ASK)
1	1-10	915 MHz band (ASK)
	11-26	Reserved
	0	868 MHz band (ASK)
	1-10	915 MHz band (ASK)
2	11-26	Reserved
3-31	Reserved	Reserved

ตารางที่ 2.1 การแบ่งช่วงค่าความถี่ของช่องทาง[4]

### 2.1.5 ตัวตรวจจับพลังงาน (Energy Detection ) [4]

Energy Detection (ED) การตรวจจับพลังงาน คือการที่มีสัญญาณวิ่งเข้ามาในตัว ED จะทำหน้าที่ในการตรวจจับและประเมินสัญญาณว่าสัญญาณความถี่ที่เข้ามาเป็นไปตามมาตรฐานของ IEEE 802.15.4 หรือไม่

### 2.1.6 การควบคุมการรับส่งข้อมูล (Carrier Sense) [4]

มีความเสมือนกับกระบวนการ Energy Detection คือการตรวจสอบคลื่นความถี่ของสัญญาณที่จะถูกส่งไปยังปลายทางนั้น อุปกรณ์มาสถานะอย่างไรและสะดวกที่จะรับสัญญาณเมื่อใด

### 2.1.7 ตัวชี้วัดคุณภาพของการเชื่อมโยง (Link Quality Indicator) [4]

ตัวชี้วัดคุณภาพของการเชื่อมโยง (Link quality indicator : LQI) ตัวช่วยในการหา ระยะทางตามมาตรฐาน IEEE 802.15.4 โดยได้ระบุว่า LQI เป็นตัวบ่งชี้คุณสมบัติ หรือ ตรวจสอบคุณภาพของสัญญาณเมื่อมีการส่งผ่านระหว่างแพ็คเก็ต ได้เกิดปรากฏการณ์ผิดพลาด (Error) หรือไม่อย่างไร

### 2.1.8 ตรวจสอบพร้อมทั้งเตรียมช่องทาง (Clear Channel Assessment) [4]

พบในกรณีในระดับชั้นของ MAC Layer ร้องขอช่องทางไปยัง PHY Layer ก่อนที่จะ ดำเนินการส่งสัญญาณ จำเป็นต้องมีการตรวจสอบช่องทางให้สะดวก และเพื่อเป็นการแน่ใจว่า ช่องทางนี้ไม่มีอุปกรณ์อื่นใดที่กำลังถูกใช้งาน

### 2.1.9 ระดับชั้นโครงข่ายไร้สาย (The Zigbee Network Layer (NWK Layer)) [4]

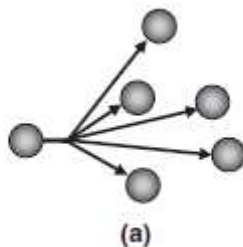
The NWK Layer ประกอบด้วยเซอริวิส 2 ส่วนคือส่วนที่เป็นข้อมูลและส่วนที่เกี่ยวข้องกับการจัดการ ซึ่งมีหน้าที่รับผิดชอบในการส่งข้อมูล พร้อมทั้งถูกออกแบบมาเพื่อช่วยให้ ช่วงของเครือข่ายแผ่กว้างและต้องสามารถรองรับโหนดจำนวนมากโดยใช้พลังงานและเวลาที่ต่ำ โดยจำแนกประเภทของการสื่อสาร 3 ภาพแบบ ดังนี้

#### 2.1.10 Broadcast [4][5]

ข้อความถูกกระจายไปทุก ๆ อุปกรณ์แบบทั่วถึงกันโดยมีการเชื่อมต่ออยู่ในเครือข่าย เดียวกันอุปกรณ์ที่ได้รับสัญญาณจะไม่สามารถควบคุมการใช้งานได้ เพราะทุกข้อความจะถูก



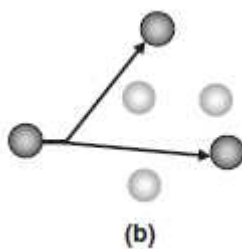
ควบคุมอยู่ที่ต้นทาง และมีเทคนิค 2 ทางคือ Broadcast – Unicast เป็นลักษณะการทำงานที่ทำบน URL เป็นหลัก และ Broadcast-Multicast เป็นลักษณะการทำงานที่ใช้ IP เป็นหลัก ดังนั้นสามารถใช้งานในลักษณะที่เป็น Low-Bandwidth



ภาพที่ 2.5 Broadcast Communication [4]

#### 2.1.11 Multicasting[5]

เป็นการส่งข้อมูลแบบ 1 ต่อ N (N คือ จำนวนผู้รับ) หรือเรียกว่า IP Multicast เป็นการออกแบบเพื่อช่วยให้การทำงานเร็วขึ้น ใช้ทรัพยากรในส่วนต่าง ๆ น้อยลงรวมถึงการใช้ Bandwidth น้อยลง โดยการส่งข้อมูลผู้ส่งจะเจาะจงกลุ่มผู้รับจากนั้นจะดำเนินการส่งข้อมูลเพียงครั้งเดียวให้กับผู้รับจำนวนหลายคนในเวลาเดียวกัน โดยข้อมูลที่ถูกส่งให้กับกลุ่ม Multicast Host ส่วนใหญ่จะเป็นข้อมูลประเภท Streaming และวิธีการนี้เป็นการใช้ Bandwidth อย่างมีประสิทธิภาพ



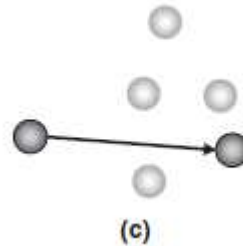
ภาพที่ 2.6 Multicasting Communication [4]

แบ่งออกเป็น 2 ประเภท คือ

- IP Multicast
- Application Layer Multicast (ALM)

### 2.1.12 Unicast [5]

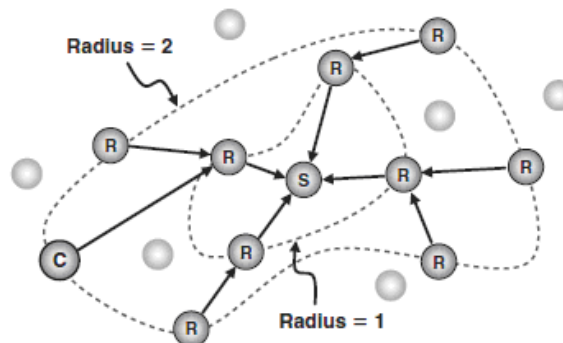
เป็นการส่งข้อมูลแบบ (1: 1) one to one ผู้ใช้สามารถควบคุมได้และส่งซ้ำได้ คือมีผู้ส่ง 1 คนและผู้รับ 1 คนและเมื่อจำนวนผู้รับมีมากกว่า 1 จำเป็นต้องส่งข้อมูลซ้ำตามจำนวนผู้รับ ซึ่งวิธีนี้ค่อนข้างเปลืองเนื้อที่และแบนด์วิธ



ภาพที่ 2.7 Unicast Communication [4]

### 2.1.13 Many –to-one Communication [4]

การสื่อสารแบบหนึ่งอุปกรณ์รับข้อมูลจากหลายแหล่งภายในเครือข่ายเดียวกัน โดยอุปกรณ์ที่ทำหน้าที่รับข้อความคือ Sink



ภาพที่ 2.8 การสื่อสารแบบ N:1 [4]

โดยมีส่วนประกอบดังต่อไปนี้

- ⊙ C คือ Zigbee Coordinator
- ⊙ R คือ Zigbee Router
- ⊙ S คือ Sink Device
- คือ Zigbee End Device

#### 2.1.14 การสื่อสารทิศทางตรงหรือ Line of Sight (LOS) [6]

คือการสื่อสารที่ใช้คลื่นความถี่วิทยุจะติดต่อสื่อสารระหว่างสถานีฐานกับสถานีลูกข่าย จะมาจากทางตรง (Direct Path) ซึ่งอุปกรณ์เครื่องส่งและเครื่องรับจะติดต่อถึงกันได้ จำเป็นต้องมีการติดตั้งให้อยู่ในแนวการสื่อสารที่ตรงกันเท่านั้น โดยปราศจากสิ่งกีดขวางใดๆ สิ่งกีดขวางการแพร่กระจายของคลื่นความถี่วิทยุ จะทำให้เกิดการลดทอนของสัญญาณจนไม่สามารถสื่อสารกันได้ เพื่อความสะดวกในการสำรวจเส้นทางการแพร่กระจายและลดปัญหาที่เกิดจากสิ่งกีดขวางต่างๆ จึงมีการนิยามพื้นที่ซึ่งมีชื่อเรียกว่า Fresnel Zone ซึ่งเป็นพื้นที่ที่มีการแพร่กระจายพลังงานส่วนใหญ่จากคลื่นความถี่วิทยุตลอดเส้นทางไปสู่เครื่องรับปลายทาง โดยขอบเขตของพื้นที่ Fresnel Zone จะกว้างใหญ่เพียงใดขึ้นอยู่กับความถี่ของคลื่นวิทยุที่ใช้รับส่ง และระยะห่างระหว่างเครื่องส่งและเครื่องรับ

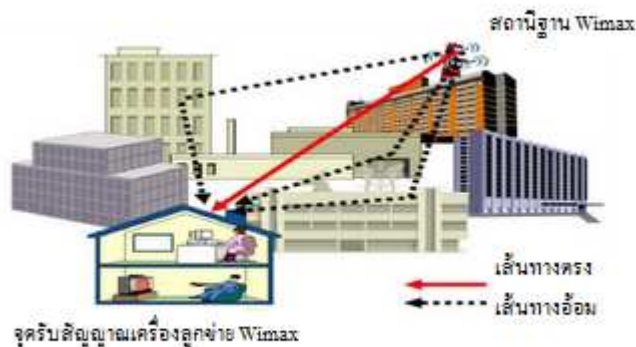


ภาพที่ 2.9 Line of Sight[6]

#### 2.1.15 การสื่อสารทิศทางอ้อมหรือ Non Line of Sight (NLOS) [6]

สถานีฐาน กับสถานีลูกข่าย สามารถสื่อสารกันได้ถึงแม้ว่าจะมีสิ่งกีดขวางสัญญาณที่รับได้มาจากหลายทิศทางคือ สัญญาณที่รอดผ่านสิ่งกีดขวาง (Absorption) สัญญาณอ้อมผ่านสิ่งกีดขวาง (Diffraction) และสัญญาณสะท้อน (Reflection) สัญญาณจากคนละทิศทางจะมีคุณสมบัติทางกายภาพแตกต่างกัน ทั้งความแรงของสัญญาณ (Signal Strength) เฟสของสัญญาณ Polarization และ Delay ของสัญญาณ ซึ่งต้องใช้เทคโนโลยีที่มีความซับซ้อนที่ด้านอุปกรณ์ หรือ Receiver จึงจะสามารถรับและแยกแยะสัญญาณที่มีมาจากคนละทิศทางนี้ได้ คุณสมบัติ Line of Sight และ Non Line of Sight มีความเกี่ยวข้องกับความถี่ที่ใช้งาน ในกรณีที่ความถี่สูงมากกว่า 11 GHz การสื่อสารระหว่างสถานีฐานและสถานีลูกข่ายต้องการการ Line of Sight เท่านั้น ส่วนความถี่ที่

ต่ำกว่า 11 GHz โดยเฉพาะย่านความถี่ที่ต่ำกว่า 6 GHz สถานีฐานและสถานีลูกข่ายสามารถติดต่อแบบ Non Line of Sight ได้คุณสมบัติ Non Line of Sight ทำให้เกิดความยืดหยุ่นในการติดตั้งและใช้งานอุปกรณ์ลูกข่าย (CPE)



ภาพที่ 2.10 การสื่อสารแบบ Non Line of Sight[6]

## 2.2 อุปกรณ์สำหรับระบุตัวตน[4]

### 2.2.1 อินฟราเรด (IR) [9]

คือ คลื่นแม่เหล็กไฟฟ้าที่มีความยาวคลื่นอยู่ระหว่างคลื่นวิทยุและแสงมีความถี่ในช่วง 10<sup>11</sup> – 10<sup>14</sup> เฮิรตซ์ มีความถี่ในช่วงเดียวกับไมโครเวฟ มีความยาวคลื่นอยู่ระหว่างแสงสีแดงกับคลื่นวิทยุสื่อสารทุกชนิดที่มีอุณหภูมิอยู่ระหว่าง -200 องศาเซลเซียสถึง 4,000 องศาเซลเซียส จะปล่อยรังสีอินฟราเรดออกมา คุณสมบัติเฉพาะตัวของรังสีอินฟราเรด ไม่เบี่ยงเบนในสนามแม่เหล็กไฟฟ้า ข้อดีคือสามารถเคลื่อนย้ายอุปกรณ์ได้ง่ายและไม่ต้องติดตั้งสัญญาณใหม่ แต่ข้อจำกัดคือต้องไม่มีสิ่งใดมาขัดขวางเส้นทางของเครื่องรับและเครื่องส่ง พร้อมทั้งระยะทางที่ใช้ในการส่งข้อมูลต้องเป็นแค่ระยะสั้นเท่านั้น (Active Badges 1992) ใช้สัญญาณอินฟราเรดมาช่วยติดตามพนักงานภายในอาคารด้วยแท็กจะปล่อยสัญญาณออกมาทุก 15 นาทีโดยจะถูกตรวจจับด้วยเซนเซอร์ที่ติดตั้งไว้ตามจุดต่าง ๆ ภายในอาคาร และใช้เทคนิคของการวัดระยะทาง (Range-based) ในการหาตำแหน่งเป้าหมาย โดยใช้วิธีการคำนวณระยะทางระหว่างเซนเซอร์กับตราประจำตัวของพนักงาน ค่าความแม่นยำขึ้นอยู่กับขนาดของห้องทำงาน โดยอัตราส่วน 1 กลุ่มตราประจำตัว ต่อ 1 ห้องทำงานโดย 1 กลุ่มตราประจำตัวต้องมี 10 ขึ้น

## 2.2.2 ความถี่วิทยุ (RF)

### 2.2.2.1 ระบบการชี้เฉพาะด้วยคลื่นความถี่วิทยุ (RFID)[7]

คือ ระบบระบุเอกลักษณ์ของวัตถุด้วยคลื่นความถี่วิทยุ เพื่อใช้ในการแสดงตำแหน่ง หรือ แสดงตัวตน ระบบ RFID จะมีองค์ประกอบหลักอยู่ สองส่วน โดยส่วนแรกคือ ทรานสปอนเดอร์หรือแท็ก (Transponder) สำหรับใช้ติดกับวัตถุต่างๆที่ต้องการ โดยแท็กจะบันทึกข้อมูลเกี่ยวกับวัตถุชิ้นนั้นๆไว้ ส่วนที่สอง คือ เครื่องสำหรับอ่านหรือเขียนข้อมูลภายในแท็กด้วยคลื่นความถี่วิทยุ (Interrogator) โดยการทำงานของเครื่องอ่านจะทำหน้าที่จ่ายกำลังงานในภาพคลื่นความถี่วิทยุให้กับตัวแท็กซึ่งส่งผลให้วงจรถออิเล็กทรอนิกส์ภายในสามารถส่งข้อมูลจำเพาะที่แสดงถึง "Identity" กลับมาประมวลผลที่ตัวอ่านได้

### 2.2.2.2 อัลตราไวด์แบนด์ (UWB) [8]

คือ เทคโนโลยีสื่อสารไร้สายส่วนบุคคลระยะสั้นสำหรับเครือข่าย Wireless Personal Area Network (WPAN) Ultrawideband (UWB) จะใช้สัญญาณพัลส์วิทยุที่มีช่วงความกว้างของพัลส์แคบมากในการส่งและรับสัญญาณ ซึ่งทำให้สัญญาณที่ส่งมีความกว้างของแถบสัญญาณที่กว้างมาก ส่งผลให้มีความสามารถถ่ายโอนข้อมูลจำนวนมากภายในระยะเวลาสั้นได้ การสื่อสารจะทำด้วยคลื่นแม่เหล็กไฟฟ้า ซึ่งมีการส่งสัญญาณด้วยคลื่นแบบแถบความถี่หรือสเปกตรัมที่กว้าง ข้อจำกัดคือเทคโนโลยีนี้ใช้กำลังไฟมาก จึงไม่เหมาะสมกับงานที่ต้องทำการ Monitoring ตลอดเวลา แต่ข้อดีคือ เมื่อนำเทคโนโลยีไปประยุกต์ใช้ในการหาวัตถุ ความสามารถในการอ่านตำแหน่งของวัตถุจะให้ค่าความแม่นยำในระดับเซนติเมตร ซึ่งถือว่าสูงกว่าเทคโนโลยี GPS ที่ให้ค่าความแม่นยำแค่หน่วยเมตร [Ubisense2004] นำเทคโนโลยี UWB มาเป็นตัวส่งสัญญาณวิทยุด้วยคลื่นความถี่ 5.8 – 7.2 GHz ออกไปโดยมีตัวรับสัญญาณเป็นแท็ก(Ubitag) ที่ติดตั้งเซ็นเซอร์เมื่อแท็กได้สัญญาณ มันจะส่งสัญญาณกลับไปพร้อมกับตำแหน่งเพื่อใช้ระบุตัวตน โดยมีการใช้เทคนิค TDoA และ AOA มาทำการคำนวณตำแหน่งโดยวัดระยะทางและมุมของสัญญาณ ค่าความแม่นยำอยู่ที่ 95 เปอร์เซ็นต์ เมื่อเซ็นเซอร์แท็กแต่ละตัวถูกติดตั้งระยะห่าง 50 เมตร ระยะอ่าน 15 เซนติเมตร

### 2.2.2.3 วิทยุฟาย(Wi-Fi) [8]

คือ เป็นเทคโนโลยีเครือข่ายไร้สายแบบ WPAN (Wireless Personal Area Network) บนมาตรฐาน IEEE 802.11 เป็นมาตรฐานที่ใช้ในการรับส่งข้อมูลที่ความเร็วสูงได้ถึง 100 เมกะบิตต่อวินาทีและมีการใช้งานในย่านความถี่ 2.4 กิกะเฮิรตซ์ (GHz) ข้อจำกัด Wi-Fi เป็นโครงข่ายที่ใหญ่ดังนั้นจึงใช้พลังงานไฟฟ้ามากกว่าเทคโนโลยีอื่น ๆ ประกอบกับยังไม่มีระบบรักษาความปลอดภัยที่มีความเป็นเสถียรภาพ Wi-Fi เป็นคลื่นวิทยุต้องอาศัยเสาอากาศ

ในการติดต่อกับสัญญาณ หรือต้องมีสภาพเครือข่ายที่เปิดอยู่เสมอ เมื่อสัญญาณถูกปิดกั้นด้วยเสา โลหะ หรือน้ำย่อมส่งผลให้ประสิทธิภาพของสัญญาณต่ำลง (Radar 2000) เลื่อนความถี่วิทยุมาเป็นสัญญาณที่ให้ตัวรับและตัวส่งและเก็บค่าความแรงของสัญญาณมาทำการคำนวณหาตำแหน่งด้วยสามเหลี่ยมอัลกอริทึม ความแม่นยำที่ 50 เมตรเซ็นติในระยะ 3-4.3 เมตร

#### 2.2.2.4 Zigbee [8]

คือ การสื่อสารในเครือข่ายเซ็นเซอร์แบบไร้สาย (Wireless Sensor Network) โดยเริ่มจากการกำหนดมาตรฐานการรับและส่งข้อมูลแบบ IEEE 802.15.4 มีคลื่นความถี่ 3 ความถี่ คือ

1. คลื่นวิทยุความถี่ที่ 2.4 GHz มี 16 ช่องสัญญาณ อัตราการรับส่งข้อมูล 250 Kbps
2. คลื่นวิทยุความถี่ที่ 915 GHz มี 10 ช่องสัญญาณ อัตราการรับส่งข้อมูล 40 Kbps
3. คลื่นวิทยุความถี่ที่ 868 GHz มี 1 ช่องสัญญาณ อัตราการรับส่งข้อมูล 20 Kbps

การทำงานของ Zigbee จะมีการเชื่อมต่อเป็นโครงข่ายของเครือข่ายเซ็นเซอร์ไร้สาย มีอยู่ 2 ภาพแบบได้แก่ ดาว (Star) และแบบระดับเดียว (Peer-to-Peer) ซึ่งมีการเชื่อมต่อแบบ Star เหมือนการเชื่อมต่อแบบโครงข่ายจิว (Piconet) โดยภายในโครงข่ายจะสื่อสารข้อมูลผ่านเซ็นเซอร์ขนาดจิ๋วหลาย ๆ ตัวโดยมีอัตราการรับส่งข้อมูลต่ำ ใช้พลังงานต่ำ เชื่อมโยงเครือข่ายระหว่าง End Device กับ Router หรือ Coordinator เข้าด้วยกัน

เทคโนโลยี Zigbee จะเน้นการใช้กำลังไฟฟ้าต่ำเพื่อให้อายุการใช้งานของแบตเตอรี่ยาวนานเหมาะสมงาน Monitoring ที่ใช้เวลานาน แต่มีข้อจำกัดคือ เรื่องความเร็วในการรับส่งข้อมูลที่น้อยจึงทำให้การส่งผ่านข้อมูลจำนวนมากเป็นไปได้ยาก (Zigbee 2010) นำเทคโนโลยีเครือข่ายไร้สาย Zigbee และ GPS มาช่วยในการคำนวณหาตำแหน่งของตัวอ้างอิง โดยใช้วิธีการวัดความแรงของสัญญาณ (RSSI) จากโหนดอ้างอิงมายังตำแหน่งโหนดเป้าหมายและคำนวณระยะทางด้วยอัลกอริทึมสามเหลี่ยมระยะ (Trilateration) ใช้ สมการ Haversine formula และกฎโคไซน์ทางคณิตศาสตร์ มาช่วยคำนวณระยะห่างระหว่าง GPS ในแนวละติจูดและลองจิจูด พบว่าวิธีดังกล่าวมีค่าความคลาดเคลื่อนจากตำแหน่งจริง เนื่องจากคุณสมบัติ ความชื้น ของสถานที่ทำการทดลอง เขาจึงเสนอแนะเป็นแนวทางการพัฒนาต่อเพื่อมาปรับปรุงผลด้วยการเลือกวิธีของการคาดคะเน (Location Fingerprint Technique) เป็น การหาตำแหน่ง จากการเปรียบเทียบพารามิเตอร์ของสัญญาณของเป้าหมายกับพารามิเตอร์

ของโหนดใกล้เคียง และเพิ่มจำนวนตัวอ้างอิงให้มากขึ้น เพื่อผลที่มีความแม่นยำและลดความคลาดเคลื่อนให้น้อยลง

ทางผู้วิจัย เลือกรูปกรณ์ประเภท Zigbee เข้ามาพัฒนาระบบเนื่องจาก ด้วยคุณสมบัติเฉพาะ ที่ใช้กำลังไฟต่ำประหยัดพลังงาน ทำให้สามารถใช้งานได้ยาวนานกว่าเทคโนโลยีชนิดอื่นและมีการเชื่อมต่ออย่างซับซ้อนเพื่อรองรับระบบการเชื่อมต่อสำหรับเครือข่ายขนาดใหญ่ และเหมาะสมกับงานการสื่อสารระยะใกล้ 10 – 100 เมตร

## 2.3 หลักการหาตำแหน่งวัตถุ

### 2.3.1 หลักการที่ใช้พื้นฐานของการคาดคะเน (Location Estimation Based on Location Fingerprinting ) [4][16]

ในการระบุตำแหน่งสามารถแบ่ง 2 เทคนิคหลักคือ เทคนิคที่ใช้พื้นฐานของการคาดคะเน (Proximity-based) และเทคนิคที่ใช้พื้นฐานของการหาระยะทาง (Range-based)[16]

เทคนิคที่ใช้พื้นฐานของการคาดคะเน (Location Fingerprint Technique)[16] เป็นการหาตำแหน่ง จากการเปรียบเทียบพารามิเตอร์ของสัญญาณของเป้าหมายกับพารามิเตอร์ของโหนดใกล้เคียง ตำแหน่งเป้าหมายสามารถประมาณค่าจากโหนดอ้างอิงที่มีค่าพารามิเตอร์ใกล้เคียงกับโหนดเป้าหมายมากที่สุด เรียกได้อีกแบบว่า Proximity-based จะมีความแม่นยำมาก แต่มีกระบวนการในการวิเคราะห์และประมวลผลที่ซับซ้อนกว่าและใช้เวลามากกว่า

เป็นการระบุตำแหน่งแบบการคาดคะเนจากค่า RSSI ที่ระบบได้ทำการเรียนรู้ไว้ เรียกว่า Fingerprint โดยทำการเก็บค่าทุก ๆ 1 เมตร ตลอดพื้นที่การทดลอง ดังสมการต่อไปนี้

$$LF = \begin{bmatrix} LF_1 \\ \vdots \\ LF_n \end{bmatrix} = \begin{bmatrix} RSSI_{11} & RSSI_{12} & RSSI_{13} & RSSI_{14} \\ \vdots & \vdots & \vdots & \vdots \\ RSSI_{n1} & RSSI_{n2} & RSSI_{n3} & RSSI_{n4} \end{bmatrix} \quad (2.1)$$

และอยู่ในตำแหน่งดังต่อไปนี้

$$X(LF) = \begin{bmatrix} X(LF_1) \\ \vdots \\ X(LF_n) \end{bmatrix} = \begin{bmatrix} x_1, y_1 \\ \vdots \\ x_n, y_n \end{bmatrix} \quad (2.2)$$

เมื่อทำการระบุตำแหน่งจะทำการดึงค่า RSSI ระหว่างโหนดอ้างอิงต่าง ๆ กับโหนดไม่ทราบค่าออกมา

$$OB = [RSSI_1 \text{ } RSSI_2 \text{ } RSSI_3 \text{ } RSSI_4] \quad (2.3)$$

หลังจากนั้นจะนำค่า RSSI ระหว่างโหนดอ้างอิงต่าง ๆ กับโหนดเป้าหมายมาเปรียบเทียบ ค่าที่ได้จะเก็บไว้เพื่อหาตำแหน่ง

$$E = \begin{bmatrix} E_1 \\ \vdots \\ E_n \end{bmatrix} \quad (2.4)$$

$$\frac{\sqrt{(RSSI_{11} - RSSI_1)^2 + (RSSI_{12} - RSSI_2)^2 + (RSSI_{13} - RSSI_3)^2 + (RSSI_{14} - RSSI_4)^2}}{\sqrt{(RSSI_{n1} - RSSI_1)^2 + (RSSI_{n2} - RSSI_2)^2 + (RSSI_{n3} - RSSI_3)^2 + (RSSI_{n4} - RSSI_4)^2}} \quad (2.5)$$

หลักการนี้จะถูกแบ่งออกเป็นสอง ระยะ ดังนี้ [4]

ระยะแรก Site Survey phase (Offline Training) : จะทำการสร้างฐานข้อมูลมาเพื่อเก็บค่า RSSI และตำแหน่งของโหนดอ้างอิง (anchor node)

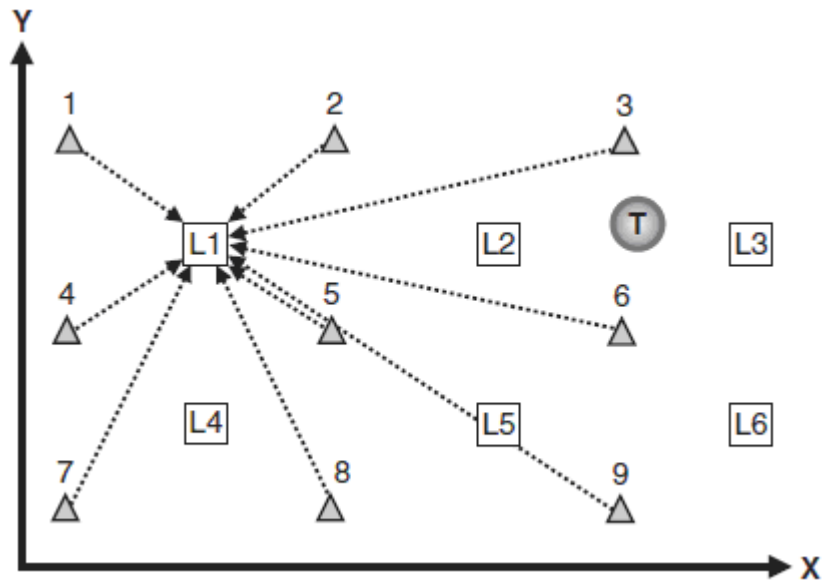
ระยะสอง Real Time phase : โหนดแต่ละตัวที่ถูกติดตาม คือจะระบุตำแหน่งของมันได้ด้วยการเปรียบเทียบค่าการวัดของสัญญาณ RSSI จากโหนดอ้างอิง และการตอบรับจากฐานข้อมูล [1]

ในตำแหน่ง จะประกอบด้วยค่าของอาเรย์ ดังนี้

$$SS_{L_1} = [SS_{L_{11}} \text{ } SS_{L_{12}} \text{ } \dots \text{ } SS_{L_{19}}] \quad (2.6)$$

เมื่อค่า  $SS_{L_i}$  คือค่าสัญญาณที่ได้มาจากโหนดอ้างอิง (Anchor node) ของตัวที่  $i$  ณ ตำแหน่งที่  $L_1$  โดยฐานข้อมูลจะประกอบด้วยค่าความแรงของสัญญาณที่มีความสัมพันธ์กับค่า  $L_1$  ถึง  $L_6$  จะใช้ในการอ้างอิงแผนที่สัญญาณ (Radio map)





ภาพที่ 2.11 เทคนิคที่ใช้พื้นฐานของการคาดคะเน. [4]

- △ คือ โหนดอ้างอิง (anchor node)
- L1 คือ ตัววัดช่วงเวลา Offline Training
- T คือ โหนดเป้าหมายที่ต้องการติดตาม (Blind node)
- > คือ ทิศทางสัญญาณ

หลังจากที่การหาค่าของตำแหน่งถูกนำไปปฏิบัติ , ผลลัพธ์ที่ได้สามารถนำไปปรับปรุงได้ตลอดเวลาด้วยการสร้างฐานข้อมูลกลุ่มใหม่ในขั้นระยะของ Training phase ต่อมาทำการเปิดโหมดเพื่อรับค่าสัญญาณจากโหนดแต่ละตัวที่ถูกประจำที่

$$SS_{current} = [SS_{current1} SS_{current2} \dots SS_{current9}] \tag{2.7}$$

เมื่อ  $SS_{current1}$  คือค่าสัญญาณที่รับมาจากโหนดอ้างอิง (Anchor node)

ใช้ The Euclidian distance ทำการประเมินระยะทางระหว่าง Current Signal ที่ได้มาจากขั้นตอน Real -Time phase และ ค่าสัญญาณที่เกี่ยวข้องกับโหนดทุกโหนด เช่น ระหว่าง  $SS_{L1}$  และ  $SS_{current}$  จะถูกนำมาคำนวณได้โดยใส่สมการของ Euclidian distance ดังนี้

$$d_{(SS_{current}, SS_{L1})} = \sqrt{(SS_{current1} - SS_{L11})^2 + (SS_{current2} - SS_{L12})^2 + \dots + (SS_{current9} - SS_{L19})^2} \tag{2.8}$$

$$d_{(SS_{current}, SS_{L1})} = \sqrt{\sum_{i=1}^9 (SS_{currenti} - SS_{L1i})^2} \tag{2.9}$$

### 2.3.2 หลักการที่ใช้พื้นฐานของการหาระยะทาง (Range-based) [4][16]

เทคนิคที่ใช้พื้นฐานของการหาระยะทาง (Range-based) จะเป็นการหาตำแหน่งเป้าหมาย โดยใช้วิธีการคำนวณระยะทางระหว่างโหนดอ้างอิงกับเป้าหมาย สามารถหาได้หลายวิธี โดยจะเกี่ยวข้องกับ การวัดค่าความแรงสัญญาณระหว่างโหนดเป้าหมายกับโหนดอ้างอิงในทุก ๆ ตำแหน่งจากนั้น นำมาทำการแปลงค่าสัญญาณให้เป็นระยะทาง เมื่อระยะทางใกล้ค่าสัญญาณก็จะติดลบน้อยหรือมีค่าความแรงสัญญาณเข้มมากแต่ในทางตรงข้ามหากโหนดถูกติดตั้งให้อยู่ในระยะที่ไกลกันออกไป ส่งผลให้ค่าสัญญาณอ่อนลงหรือมีค่าติดลบมาก (-dBm) ค่าสัญญาณจะเป็นตัวแปรหลักที่สำคัญของหลักการนี้ เนื่องจากจะเป็นการรับค่าสัญญาณจากอุปกรณ์ (Hardware Chipcon CC2430 และ CC2431) คือ ค่า LQI หรือเรียกว่า ตัวชี้วัดคุณภาพของการเชื่อมโยง (Link Quality Indicator (LQI))[4] เนื่องจากมีปัจจัยทางสภาพแวดล้อมหลายอย่างที่ทำให้ประสิทธิภาพในการทำงานของสัญญาณลดทอนลงไป บริษัทผู้ผลิต [Texas Instruments] จึงได้มีการนำเสนอวิธีการอื่นเข้ามาช่วยเสริมในการวัดระยะทาง นั่นคือ ตัวชี้วัดคุณภาพของการเชื่อมโยง (Link Quality Indicator : LQI) มาช่วยในการหาระยะทางตามมาตรฐาน IEEE 802.15.4 โดยได้ระบุว่า LQI เป็นตัวบ่งชี้คุณสมบัติ หรือคุณภาพของชุดข้อมูลที่ได้รับและเป็นสัดส่วนกับค่าสัญญาณ โดยค่า LQI จะมีค่าในช่วงระหว่าง 0 – 255 [22]

ทางผู้วิจัยเลือกหลักการการระบุตำแหน่งวัตถุ ด้วยวิธีการหาระยะทาง (Range-based) เนื่องจากเป็น หลักการที่ไม่มีขั้นตอนที่ซับซ้อน และมีความสอดคล้องกับชุดอุปกรณ์ที่ใช้ในการพัฒนางานวิจัยในครั้งนี้

### 2.4 การวัดค่าความแรงของสัญญาณ (Received Signal Strength Indication (RSSI))[4]

การประยุกต์ใช้งานโครงข่ายเซ็นเซอร์แบบไร้สายในการระบุตำแหน่ง จะนิยมใช้การวัดระดับความแรงของสัญญาณที่รับได้ (Received Signal Strength Indicator : RSSI) ในการค้นหาโหนดเป้าหมาย (Blind node) และมีการกำหนดตำแหน่งที่แน่นอนของกลุ่มโหนดอ้างอิง (Reference node) และใช้งานได้โดยการวัดความแรงของระดับสัญญาณ ระหว่างโหนดอ้างอิงกับโหนดเป้าหมาย

RSSI ย่อมาจาก Receive Signal Strength Indicator เป็นค่าที่ใช้บอกความแรงของสัญญาณวิทยุที่ได้รับในเทอมของพลังงานมีหน่วยเป็น -dBm (Decibels Milliwatt) โดยค่า RSSI จะแปรผันตรงกับความแรงของสัญญาณ นั่นคือ ถ้าค่า RSSI มีค่ามากแสดงว่าสัญญาณที่ได้รับมีความแรงสูง นั่นคือตัวส่งและตัวรับอยู่ใกล้กันและในทางกลับกันหากค่า RSSI มีค่าน้อยแสดงว่า

สัญญาณที่ได้รับมีความแรงต่ำ ตัวส่งและตัวรับอยู่ไกลกัน ตัวอย่างงานวิจัย RADAR [25] ที่นำเทคโนโลยีคลื่นความถี่วิทยุ (Radio) กับมาตรฐานเครือข่ายไร้สาย IEEE 802.11 ในการติดตามและค้นหาบุคคลภายในอาคาร และมีการคำนวณตำแหน่งและระยะทางเพื่อช่วยเสริมสร้างความแม่นยำด้วยการนำเอาเทคนิคของ Triangulation มาประยุกต์ใช้ ข้อดีของวิธีดังกล่าว ง่ายต่อการติดตั้งค่า configure และอุปกรณ์ต่าง ๆ แต่ข้อจำกัด คือ ต้องหาอุปกรณ์การ์ดแลนเฉพาะรุ่นรวมทั้งระยะทางต้องอยู่ในช่วงของ 3 – 4.3 เมตร และงานวิจัยอีกฉบับหนึ่ง SpotOn [13] ที่ใช้เทคโนโลยีคลื่นความถี่วิทยุ (Radio) กับค่าวัดความแรงสัญญาณ (RSSI) โดยวิธีการวัดระยะทางใช้เทคนิค Adhoc lateration ในการวัดระยะทางจากแท็ก ข้อดีคือ วิธีนี้ให้ค่าความแม่นยำได้ดีกว่าเทคนิค TOA ข้อจำกัดคือ การจัดกลุ่มของแท็ก (Cluster size) ต้องมากกว่าถึง 2 เท่าและปริมาณความหนาแน่น (Density) ในการวางของแต่ละตำแหน่ง สำหรับการจะทำให้ค่า RSSI มีความน่าเชื่อถือมากขึ้นจำเป็นต้องเก็บข้อมูลไว้หลาย ๆ ครั้งเพื่อนำมาวิเคราะห์ภายหลัง ส่งผลให้เกิดการผิดพลาดลดน้อยลง

#### สาเหตุที่ทำให้เกิดข้อผิดพลาด[16]

- Non Line of Sight (NLOS) คือการสื่อสารแบบไม่เป็นเส้นตรง ทำให้การวัดมุมในระยะทางที่ไกลก่อให้เกิดความผิดพลาดขึ้นได้
- Multiple Access Interference คือปัญหาที่พบในระบบ CDMA เกิดจากเครื่องส่งที่มีกำลังส่งสูงกว่าส่งสัญญาณไปรบกวนเครื่องส่งที่มีกำลังส่งต่ำกว่า

#### การวัดค่าหาความแรงสัญญาณ [4][16]

เป็นวิธีการคำนวณหาค่าความแรงของสัญญาณวิทยุระหว่างเซนเซอร์ไร้สายตัวหนึ่งกับเซนเซอร์ไร้สายอีกตัวหนึ่ง โดยมีความสัมพันธ์กันตามสมการดังนี้

$$RSSI = -10n \log_{10}(d) + A \quad (2.10)$$

เมื่อ RSSI คือค่าความแรงของสัญญาณวิทยุ มีหน่วยเป็น เดซิเบลล์มิลลิวัตต์ (-dBm)

n คือค่าคงที่การกระจายสัญญาณของสถานที่นั้น ๆ

d คือระยะห่างระหว่างเซนเซอร์ไร้สาย (เมตร)

A คือค่าความแรงของสัญญาณที่ได้รับใน 1 เมตร

จากสมการข้างต้น ถ้าค่าระยะทาง (d) เพิ่มมากขึ้นก็จะส่งผลให้ค่า RSSI ลดลง ซึ่งบอกค่าความแรงของสัญญาณ สามารถนำผลนี้ไปใช้เพื่อหาระยะทางและปริมาณตำแหน่งวัตถุ

## การวัดความแรงสัญญาณบนพื้นฐานของอัลกอริทึมการหาตำแหน่ง [4][16]

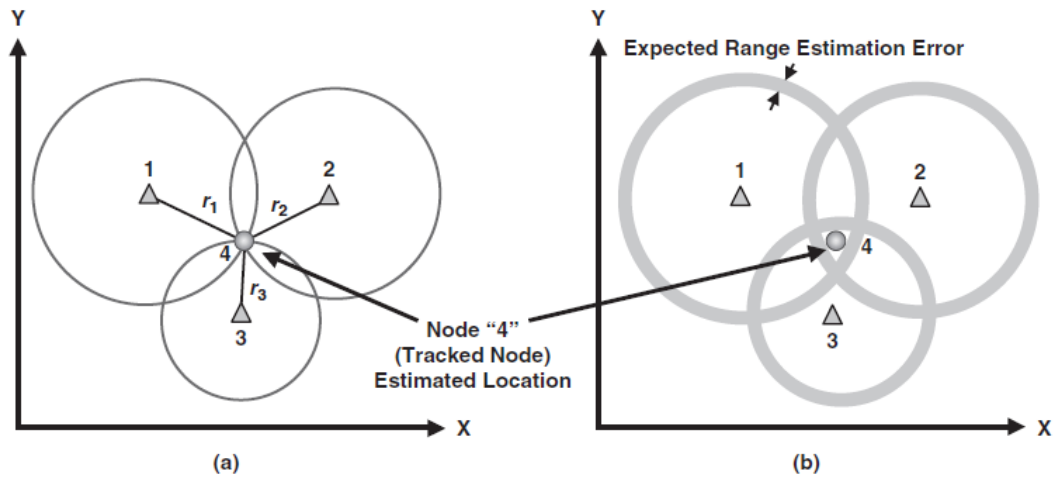
ความแรงของสัญญาณ (RSSI) และเวลา (time stamp) ที่ได้รับมาจะถูกนำมาวิเคราะห์เพื่อหาตำแหน่งของวัตถุ โดยมีพารามิเตอร์เกี่ยวข้อง 4 ตัว คือ ช่วงคงที่ (Dynamic Range) , ความแม่นยำ (Accuracy) , เส้นตรง (Linearity) , ช่วงเวลาเฉลี่ย (Averaging Period)

- ช่วงคงที่ (RSSI Dynamic range) เป็นการบอกค่าเป็นช่วงระหว่างค่ามากที่สุดและค่าน้อยสุด เช่น 92 dB ถึง +4dB
- ความแม่นยำ (RSSI Accuracy) เป็นการวัดความผิดพลาดของค่าเฉลี่ยที่มีความสัมพันธ์กับสัญญาณ RSSI จะมีอัลกอริทึม Nearest และ 3-Nearest สำหรับคำนวณค่าความคลาดเคลื่อน
- เส้นตรง (RSSI Linearity) เป็นการวัดค่าแนวเส้นตรงโดยพิจารณาจากค่ามากที่สุดต่ออัตราส่วนของค่าตามจริงของ RS Signal โดยจะคิดจากค่า RSSI กับ Input ดังสมการนี้ (2.10)
- ช่วงเวลาเฉลี่ย (Averaging Period) เป็นการนำค่าของเวลามาทำการเฉลี่ยกันเพื่อหา RSSI ใช้อัลกอริทึม TDOA , AOA และ TOA

### 2.5 การวัดค่าความแรงสัญญาณด้วยวิธีสามเหลี่ยม [4]

บนพื้นฐานทฤษฎีสามเหลี่ยมระยะจะใช้วิธีการประมาณค่าของตำแหน่งจากโหนด 3 โหนดซึ่งจะถูกติดอยู่ในแต่ละจุดที่ทราบตำแหน่งอย่างชัดเจน และโหนดที่ 4 สามารถเคลื่อนที่ได้โดยมีวัตถุประสงค์ที่จะประเมินค่าตำแหน่งจากโหนดตัวที่ 4 การจัดวางตำแหน่งของโหนดและมาทำการวิเคราะห์เพื่อหาตำแหน่งจะทำได้เป็น 2 มิติ (2 Dimensional) คือมีแกน X และ Y โดยจะเริ่มจาก โหนดที่ 4 ส่งสัญญาณกับแรงกำลังส่ง โดยมีการกระจายสัญญาณออกไปทุกทิศทางตามแกนขนาน X และ Y หรือแนวระนาบ แต่ละโหนดคือ โหนดที่ 1 , 2 และ 3 จะถูกนำมาประมาณหาระยะทางของที่อยู่และ โหนดตัวที่ 4 ใช้สมการดังกล่าวเพื่ออ้างอิงหาตำแหน่ง

$$P_R = P_T - 10 \times n \times \log_{10}(f) - 10 \times n \times \log_{10}(r) + 30 \times n - 32.44 \text{ (dBm)} \quad (2.11)$$



ภาพที่ 2.12 (a)การวัดระยะทางด้วยเทคนิคสามเหลี่ยมระยะ (Trilateration) [4]

(b)การประมาณจุดคลาดเคลื่อนของโหนด [4]

เมื่อ  $P_T$  คือกำลังส่ง (หน่วย dBm) ณ. ที่นี้คือโหนดที่ 4

$P_R$  คือโหนดที่ถูกติดอยู่ประจำจุดต่าง ๆ

$F$  คือ สัญญาณความถี่ที่ใช้ในการส่ง (หน่วย MHz)

$n$  คือ ระยะทางที่สูญเสียไป

$r$  คือ ระยะทาง (หน่วย meters)

โหนดที่ 1 : สามารถประเมินระยะทาง ( $r_1$ ) ระหว่างของตัวเองกับของโหนดที่ 4 โดยการใช้ ค่าความแรงของสัญญาณวิทยุ มีหน่วยเป็น เดซิเบลมิลลิวัตต์ (-dBm) , โหนดที่ 4 จะอยู่ภายใต้รัศมีวงกลมของโหนดที่ 1 นำมาคิดเป็นค่าสมการตามทฤษฎี Euclidian distance [4]

$$(X_1 - X_4)^2 + (Y_1 - Y_4)^2 = r_1^2$$

$$(X_1 - X_4)^2 + (Y_1 - Y_4)^2 - r_1^2 = 0$$

หรือ

$$(X_1 - X_4)^2 + (Y_1 - Y_4)^2 - r_1^2 = 0 \quad (2.12)$$

เมื่อ  $(X_1, Y_1)$  และ  $(X_4, Y_4)$  เป็นโหนดร่วมกันระหว่างโหนดที่หนึ่ง และโหนดที่สี่ จะถูกแบ่งออกเป็นสองกลุ่มคือโหนดที่สอง  $(X_2, Y_2)$  และโหนดที่สาม  $(X_3, Y_3)$

ดังนั้น ต้องการหาตำแหน่งโหนดที่สี่เราจำเป็นต้องหา  $(X_4, Y_4)$  ซึ่งถูกแสดงในภาพสมการดังนี้

$$\begin{bmatrix} (X_1 - X_4)^2 + (Y_1 - Y_4)^2 \\ (X_2 - X_4)^2 + (Y_2 - Y_4)^2 \\ (X_3 - X_4)^2 + (Y_3 - Y_4)^2 \end{bmatrix} - \begin{bmatrix} r_1^2 \\ r_2^2 \\ r_3^2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \quad (2.13)$$

วิธีนี้เป็นวิธีการหาค่าความสัมพันธ์ของโหนดด้วยการใช้เรขาคณิตสามเหลี่ยม โดยกำหนดให้ค่า = 0 อย่างไรก็ตาม วิธีนี้ยังไม่ช่วยตรวจสอบข้อผิดพลาดเคลื่อน (Error) ได้ ดังนั้นจึงเข้าสู่สมการนี้

$$abs \left( \begin{bmatrix} (X_1 - X_4)^2 + (Y_1 - Y_4)^2 \\ (X_2 - X_4)^2 + (Y_2 - Y_4)^2 \\ (X_3 - X_4)^2 + (Y_3 - Y_4)^2 \end{bmatrix} - \begin{bmatrix} r_1^2 \\ r_2^2 \\ r_3^2 \end{bmatrix} \right) = \begin{bmatrix} e_1^2 \\ e_2^2 \\ e_3^2 \end{bmatrix} = E \quad (2.14)$$

เมื่อเกิดข้อผิดพลาดเคลื่อน(Error) จะถูกแทนด้วยสมการนี้

$$\text{Square Error} = e_1^2 + e_2^2 + e_3^2 \quad (2.15)$$

### 2.5.1 อัลกอริทึมหาตำแหน่งด้วยมุม (Angle-of-Arrival Based (AOA)) [4]

เป็นวิธีการหาตำแหน่งด้วยมุม (Angle of Arrival : AOA) โดยใช้ค่ามุมของสัญญาณที่ส่งมาจากวัตถุ สามารถค้นหาตำแหน่งของวัตถุจากค่ามุมที่ได้รับที่โหนดจำนวน 2 มุมในแต่ละโหนดและนำข้อมูลที่ได้รับมาสร้างสมการเส้นตรงหาตำแหน่งของวัตถุได้ดังสมการ เมื่อโหนดอ้างอิง (anchor node) ทำการกระจายสัญญาณออกไปทุกทิศทาง ตัวที่เป็นโหนดเป้าหมายนั้นก็จจะรับค่าสัญญาณจาก โหนดอ้างอิง (anchor node) และทำการประมาณหาตำแหน่งตัวเองโดยผลลัพธ์นั้นจะให้ค่าเป็นองศาประเมินและแบ่งเกณฑ์ตามทิศ เช่นทิศเหนือ

$$(y - y_i) = \tan(\theta_i)(x - x_i)$$

$$(y - y_i) = \tan(\theta_i)(x - x_j)$$

โดยที่

$$\theta_i = \tan^{-1} \left( \frac{|y_i - y|}{|x_i - x|} \right)$$

และ

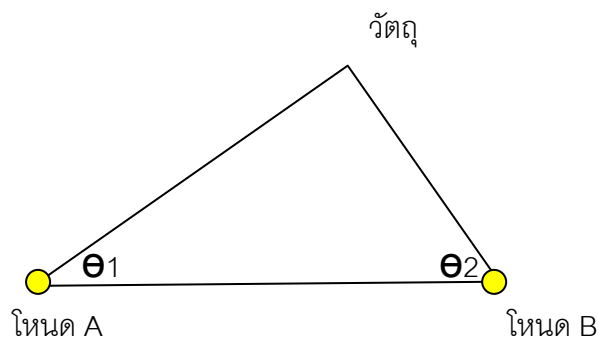
$$\theta_i = \tan^{-1} \left( \frac{|y_j - y_i|}{|x_j - x_i|} \right)$$

(2.16)

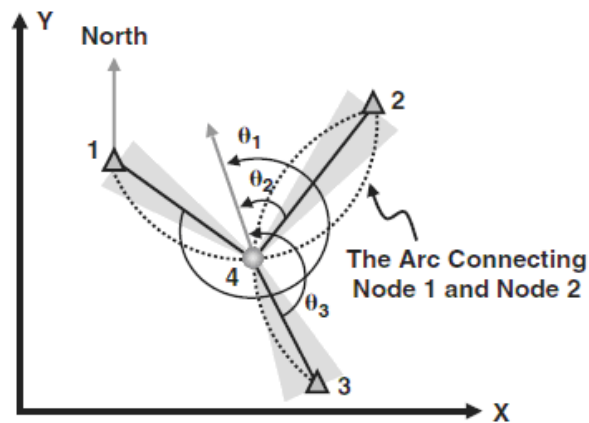
$(x, y)$  เป็นพิกัดที่ต้องการทราบของวัตถุที่ต้องการทราบตำแหน่ง

$(x_i, y_i)$  เป็นพิกัดที่ต้องการทราบของโหนด ตามแนวแกน x และ y จุดที่ i

$(x_j, y_j)$  เป็นพิกัดที่ต้องการทราบของโหนด ตามแนวแกน x และ y จุดที่ j



ภาพที่ 2.13 แสดงการหาตำแหน่งด้วยการวัดแบบ AOA



ภาพที่ 2.14 อัลกอริทึมหาตำแหน่งด้วยมุม (AOA)

- △ คือโหนดอ้างอิง (Anchor node) กับเสา Antenna 2 ทิศทาง
- คือ โหนดเป้าหมาย (Blind node) ที่ทำคำนวณมุมของโหนด

โหนดอ้างอิง (Anchor node) รับและส่งสัญญาณออกไปและโหนดเป้าหมายรับสัญญาณและทำการประมวลผลว่าตำแหน่งของโหนดอ้างอิงใดที่มีค่าสัญญาณแรงที่สุดแสดงว่าตัวนั้นอยู่ใกล้ตัวเป้าหมายมากที่สุด จากนั้นมาทำการคำนวณเพื่อหาทิศทางของโหนด ที่ 1 , 4 และ 2

$$142 = 360^\circ - (\theta_1 - \theta_2) \quad (2.17)$$

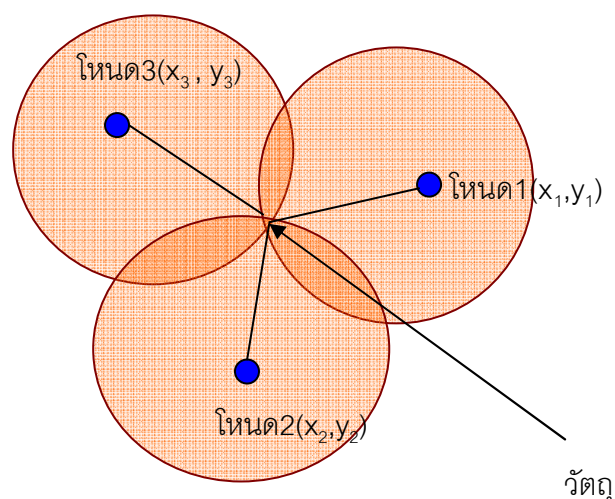
และโหนดที่ 2 , 4 และ 3

$$243 = (\theta_3 - \theta_2) \quad (2.18)$$

โดยการอ้างอิงจากความแตกต่างของเวลาของสัญญาณที่มาถึง จะต้องมีจำนวนโหนดอย่างน้อยจำนวนสาม โหนดขึ้นไป ข้อจำกัดของวิธีการนี้คือ การเกิดเส้นทางของสัญญาณหรือการขาดสัญญาณหรือเส้นทางในแนวสายตา (Line of sight) จะส่งผลให้เกิดการวัดมุมที่ผิดพลาดมากยิ่งขึ้นและการเพิ่มค่าใช้จ่ายในการปรับแต่งเสาอากาศเพื่อให้วัดค่าได้อย่างถูกต้องสม่ำเสมอ

### 2.5.2 อัลกอริทึมหาตำแหน่งด้วยเวลาที่มาถึง (Time of Arrival (TOA)) [4]

การหาตำแหน่งโดยการอ้างอิงจากเวลาของการมาถึง (TOA) ในกรณีที่เป็นตำแหน่งสอง มิติจะต้องพิจารณาสัญญาณอย่างน้อย 3 จุดตามภาพที่ (2.15) และจับเวลาในการเคลื่อนที่ของสัญญาณทางเดียว (One way) เพื่อคำนวณระยะทางระหว่างหน่วยวัดกับเครื่องส่งสัญญาณ (Signal Transmitter)



ภาพที่ 2.15 แสดงการหาตำแหน่งด้วยการวัดแบบ TOA[4]



สำหรับวิธีการคำนวณตำแหน่งแบบวิธีการอ้างอิงจากเวลาของการมาถึงคือ การหาระยะห่างระหว่างวัตถุและโหนด หลักการคือ หาระยะห่าง  $R_i$  ซึ่งเป็นระยะทางระหว่างวัตถุกับโหนดที่  $i$  ได้จากสมการ  $R_i$  เป็นวิธีการประมาณระยะทางของโหนดกับสัญญาณ วิธีการ TOA จะร้องขอเวลาของการรับและส่ง และระยะทางจากโหนดตัวที่ต้องการติดตามกับ Anchor node ( $d_i$ ) ซึ่งจะได้มาจาก ( $t_i$ ) และ Speed of light ในทางตรงข้ามวิธี TDOA จะขอเวลาที่รับสัญญาณมาเท่านั้น Anchor node จะส่งสัญญาณไปโหนดที่ติดตาม และจับเวลาเมื่อสัญญาณมาถึงจากนั้นเอามาคำนวณหาค่าความต่างระหว่าง Anchor node ไปยังตำแหน่งของโหนดติดตาม ด้วยสมการนี้

$$R_i = ct_i \quad (2.19)$$

เมื่อ  $c$  คือค่าความเร็วแสงมีค่าเท่ากับ  $3 \times 10^8$  เมตรต่อวินาที

และ  $t_i$  คือระยะเวลาในการเดินทางของสัญญาณระหว่างวัตถุไปที่โหนด  $i$

วิธีการนี้ต้องมีวงจรในการสร้างความสอดคล้องของสัญญาณนาฬิการะหว่างวัตถุกับโหนด จึงจะทำให้สามารถวัดค่า  $t_i$  ได้ พิจารณาจากสมการมาตรฐานของวงกลมที่รัศมี  $R$  คือ

$$R_i = \sqrt{(X_i - x)^2 + (Y_i - y)^2 + (Z_i - z)^2} \text{ โดย } i = 1, 2, 3, \dots, N \quad (2.20)$$

เมื่อ  $(X_i, Y_i, Z_i)$  เป็นพิกัดของโหนด  $i$

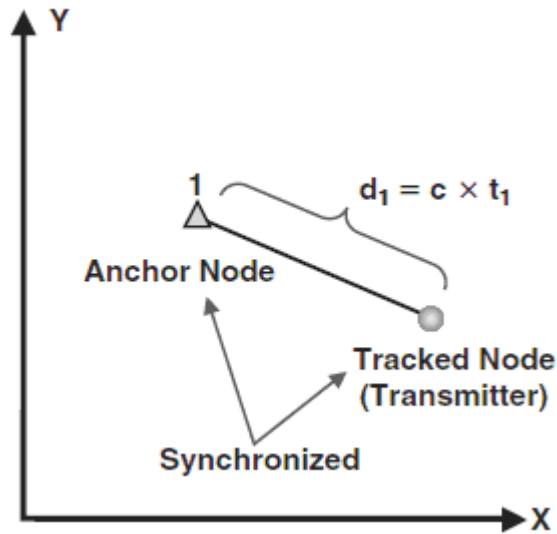
และ  $(x, y, z)$  เป็นพิกัดของตำแหน่งของวัตถุที่ต้องการทราบตำแหน่ง จากภาพที่ (2.15) ทั้งโหนดอ้างอิง (Anchor node) และโหนดเป้าหมาย (Tracked node) ต้องการรับและส่งสัญญาณเป็นจังหวะเดียวกัน (Synchronize) โดยกำหนดค่า

$d_1$  คือระยะทางระหว่างโหนดอ้างอิงกับโหนดเป้าหมาย

$t_1$  คือระยะเวลาในการเดินทางของสัญญาณระหว่างโหนดอ้างอิงไปที่โหนดเป้าหมาย

$c$  คือค่าความเร็วแสงมีค่าเท่ากับ  $3 \times 10^8$  เมตรต่อวินาที

ข้อจำกัดของวิธีนี้ คือ ต้องการความสอดคล้องกันในเรื่องช่วงเวลาของสัญญาณนาฬิการะหว่างโหนดเป้าหมายและโหนดอ้างอิง หากนับระยะเวลาผิดพลาดไป 1 ไมโครวินาทีส่งผลให้เกิดการผิดพลาดของระยะทางไป 300 เมตร [21]



ภาพที่ 2.16 การคิดระยะทางจากอัลกอริทึมเวลาที่มาถึง (TOA)[4]

ข้อจำกัดของเทคนิคนี้คือ ต้องจัดหาอุปกรณ์ฮาร์ดแวร์มาเสริมและ ต้องการความสอดคล้องกันในเรื่องของเวลาของสัญญาณนาฬิการะหว่างโหนดเป้าหมายและโหนดอ้างอิง

### 2.5.3 อัลกอริทึมหาตำแหน่งด้วยเวลาที่มาถึง (Time-Difference of Arrival (TDOA))[4]

การหาตำแหน่งโดยอ้างอิงจากค่าความแตกต่างของเวลาของสัญญาณที่มาถึง (TDOA) แนวคิดนี้เพื่อช่วยแก้ไขเวลาที่ต้องมีความเป็นจังหวะเดียวกันของตัวรับและตัวส่ง (Synchronize) ของวิธี TOA โดยเนื่องจากสัญญาณเคลื่อนที่ด้วยความเร็วแสงทำให้พบว่าในกรณีที่วัดเวลาผิดพลาดไป 1 ไมโครวินาทีทำให้เกิดข้อผิดพลาดในการวัดระยะถึง 300 เมตร ดังนั้นเพื่อแก้ไขปัญหาเรื่องการสอดคล้องของสัญญาณนาฬิกาทำให้เกิดแนวคิดแทนที่จะทำการนับเวลาที่มาถึงโหนด มานับเวลาที่แตกต่างกันที่รับได้จากโหนดจำนวน 2 คู่เพื่อนำข้อมูลมาสร้างสมการไฮเพอร์โบลิก เพื่อในการหาตำแหน่งของวัตถุโดยพิจารณาการหาค่าตำแหน่งของโหนดได้ โดยใช้หลักการของการนับเวลา เวลาที่แตกต่างกันที่นับได้จากคู่ของสัญญาณจากวัตถุไปที่โหนดตำแหน่งที่  $i$  และตำแหน่งที่  $j$  เพื่อสร้างสมการไฮเพอร์โบลิกพิจารณาความสัมพันธ์ของผลต่างของระยะทางระหว่างโหนดตำแหน่งที่  $i$  และตำแหน่งที่  $j$  ดังนี้

$$R_{ij} = ct_{ij} = R_i - R_j \quad (2.21)$$

โดยที่  $R_{ij}$  เป็นผลต่างของระยะทางระหว่างโหนด ตำแหน่งที่  $i$  และตำแหน่งที่  $j$  และ  $t_{ij}$  เป็นความแตกต่างของเวลาของสัญญาณจากวัตถุไปที่โหนดตำแหน่งที่  $i$  และตำแหน่งที่  $j$  และ  $c$  คือค่าความเร็วแสงมีค่าเท่ากับ  $3 \times 10^8$  เมตรต่อวินาทีและ  $R_i$  คือระยะห่างระหว่างโหนดที่ตำแหน่งที่ พิกัด  $(x, y)$  กับวัตถุที่มีพิกัดเป็น  $(X_i, Y_i)$  ดังสมการดังนี้

$$R_i = \sqrt{(X_i - x)^2 + (Y_i - y)^2 + (Z_i - z)^2} = \sqrt{X_i^2 + Y_i^2 - 2X_i x - 2Y_i y + x^2 + y^2} \quad (2.22)$$

และสามารถหาความสัมพันธ์ของผลต่างของระยะทางได้จากวัตถุไปที่ไหนต  $i, j$  โดยมี ความสัมพันธ์ตามสมการดังนี้

$$R_{i,j} = \sqrt{(X_i - x)^2 + (Y_i - y)^2} = \sqrt{(X_j - x)^2 + (Y_j - y)^2} \quad (2.23)$$

โดยที่ค่า  $R_{i,j}$  เป็นผลต่างของระยะทางระหว่างวัตถุตำแหน่งที่  $i$  และตำแหน่งที่  $j$  และ  $(X_i, Y_i)$  และ  $(X_j, Y_j)$  เป็นพิกัดของไหนตที่  $i$  และ  $j$  และ  $(x, y)$  เป็นตำแหน่งของวัตถุที่ต้องการทราบตำแหน่ง เราสามารถหาคำตอบของสมการได้เมื่อจำนวนสมการเท่ากับจำนวนตัวแปรที่ไม่ทราบค่าเนื่องจาก สมการที่ 5 ทำการแก้สมการค่อนข้างยากทำให้ต้องจัดภาพแบบของสมการใหม่ดังนี้

$$R_i^2 = (R_{i,j} + R_1)^2 \quad (2.24)$$

ทำให้จัดภาพแบบสมการใหม่ได้เป็น

$$R_{i,1}^2 + 2R_{i,1}R_1 + R_1^2 = X_i^2 + Y_i^2 - 2X_i x - 2Y_i y + x^2 + y^2 \quad (2.25)$$

ทำการลบค่าของสมการ  $R_{i,1}^2$  ที่  $i=1$  ออกจากสมการข้างต้น จะได้เป็น

$$R_{i,1}^2 + 2R_{i,1}R_1 = X_i^2 + Y_i^2 - 2X_{i,1}x - 2Y_{i,1}y \quad (2.26)$$

เมื่อ  $X_{i,1}$  และ  $Y_{i,1}$  มีค่าเท่ากับ  $(X_i - X_1)$  และ  $(Y_i - Y_1)$  ตามลำดับ ซึ่งสามารถหาคำตอบของ สมการได้ดังนี้

$$\begin{pmatrix} x \\ y \end{pmatrix} = - \begin{pmatrix} X_{2,1} & Y_{2,1} \\ X_{3,1} & Y_{3,1} \end{pmatrix}^{-1} * \begin{pmatrix} R_{2,1} \\ R_{3,1} \end{pmatrix} R_1 + \frac{1}{2} \left\{ \begin{pmatrix} R_{2,1}^2 - K_2 + K_1 \\ R_{3,1}^2 - K_3 + K_1 \end{pmatrix} \right\} \quad (2.27)$$

เมื่อ

$$K_1 = X_1^2 + Y_1^2 \quad (2.28)$$

$$K_2 = X_2^2 + Y_2^2$$

$$K_3 = X_3^2 + Y_3^2$$

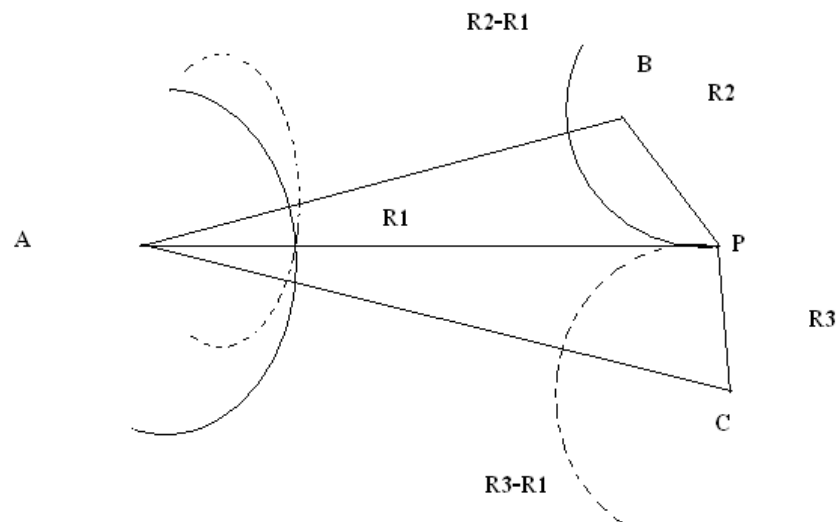
สามารถทำการหาค่า  $x$  และ  $y$  ได้ในเทอมของ  $R_1$  ได้โดยการแทนค่าของ  $x$  และ  $y$  ในสมการ ไป แทนค่าในสมการข้างต้นและจัดภาพแบบสมการในเทอมของ  $R_1$  ในภาพแบบของสมการ Quadratic ดังนี้

$$aR_1^2 + bR_1 + c = 0 \quad (2.29)$$

สามารถหาคำตอบของสมการสูตรได้เป็น

$$R_1 = \frac{-b + \sqrt{b^2 - 4ac}}{2a} \quad (2.30)$$

เลือกใช้รากของสมการเพียงค่าเดียวเนื่องจาก  $R_1$  เป็นระยะห่างระหว่างวัตถุและโหนดที่ 1 ทำให้รากของสมการมีค่าเป็นบวกเพียงค่าเดียวแล้วนำค่าที่ได้ไปแทนค่าในสมการ  $[x, y]$



ภาพที่ 2.17 แสดงการหาตำแหน่งด้วยการวัดแบบ TDOA [4]

วิธี TDOA จะใช้เฉพาะเวลารับของตัวรับ โดยเริ่มต้นจากโหนดอ้างอิง (Anchor node) รับสัญญาณจากโหนดเป้าหมาย (Tracked node) และแทนด้วยตัวแปร ดังนี้

$\Delta d$  คือ ค่าผลต่างของเวลาที่โหนดอ้างอิง (Anchor node) ทั้ง 2 โหนดมาถึงโหนดเป้าหมาย (Tracked node)

$d_1, d_2$  คือ ค่าผลต่างของระยะทาง

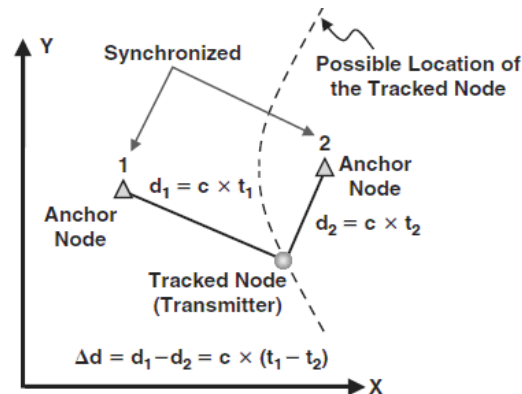
จากภาพที่ 2.17 จะได้สมการออกมาในภาพนี้

$$d_1 = \sqrt{(X_1 - X_E)^2 + (Y_1 - Y_E)^2} \quad (2.31)$$

$$d_2 = \sqrt{(X_2 - X_E)^2 + (Y_2 - Y_E)^2} \quad (2.32)$$

กรณีมี 2 โหนดอ้างอิง (Anchor node) สรุปได้จากเส้นประโค้งไฮเพอร์โบลิก (ภาพที่ 2.17) โหนดเป้าหมาย (tracked node) เขียนสมการดังนี้

$$\sqrt{(X_1 - X_E)^2 + (Y_1 - Y_E)^2} - \sqrt{(X_2 - X_E)^2 + (Y_2 - Y_E)^2} = \Delta d \quad (2.33)$$



ภาพที่ 2.18 แสดงผลต่างของ  $\Delta d$  บนพื้นฐานของ TDOA [4]

ข้อจำกัดคือ ความถี่ของสัญญาณจะลดน้อยลงเมื่อทำการตรวจจับวัตถุที่เคลื่อนที่ผ่านอย่างรวดเร็วผ่านจุดที่ทำมุมตั้งฉากกับแกนของเซ็นเซอร์

## 2.6 อัลกอริทึมในการระบุตำแหน่งด้วยวิธีการแปลงค่าสัญญาณเป็นระยะทาง

### 2.6.1 อัลกอริทึม Maximum Likelihood [4][16]

เป็นวิธีการคำนวณหาตำแหน่ง บนพื้นฐานการคำนวณของระยะทาง ซึ่งสามารถแสดงสมการได้ดังนี้

$$RSSI = -10n \log_{10}(d) + A \quad (2.10)$$

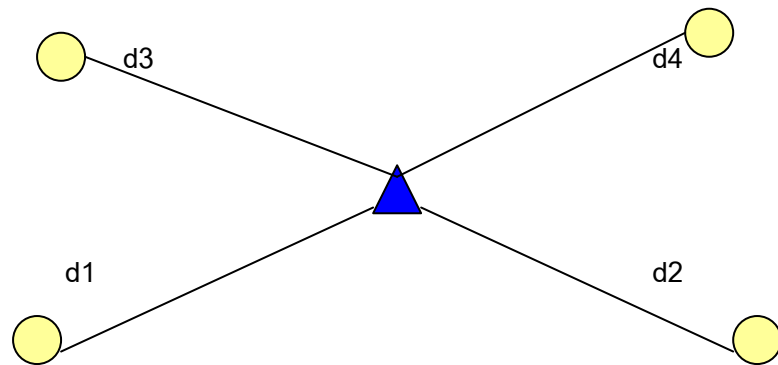
เมื่อ RSSI คือค่าความแรงของสัญญาณวิทยุ มีหน่วยเป็นเดซิเบลมิลลิวัตต์ (-dBm)

$n$  คือค่าคงที่การกระจายสัญญาณของสถานที่นั้น ๆ

$d$  คือระยะห่างระหว่างเซ็นเซอร์ไร้สาย (เมตร)

$A$  คือค่าความแรงของสัญญาณที่ได้รับใน 1 เมตร

จากสมการข้างต้น ถ้าค่าระยะทาง ( $d$ ) เพิ่มมากขึ้นก็จะส่งผลให้ค่า RSSI ลดลงซึ่งบอกค่าความแรงของสัญญาณสามารถนำผลนี้ไปใช้เพื่อหาระยะทางและตำแหน่งของวัตถุ



ภาพที่ 2.19 แสดงการหาระยะทาง (d) ระหว่างโหนดข้างอิง[16]

เมื่อทำการคำนวณหาค่า d แล้วจะได้นำค่าไปใส่ในสมการดังนี้

$$\begin{aligned}(x_1 - x)^2 + (y_1 - y)^2 &= d_1^2 \\(x_2 - x)^2 + (y_2 - y)^2 &= d_2^2 \\(x_n - x)^2 + (y_n - y)^2 &= d_n^2\end{aligned}\quad (2.34)$$

สามารถจัดภาพแบบได้เป็น

$$\begin{aligned}x^2 - x_n^2 - 2(x_1 - x_n)x + y_1^2 \\- y_n^2 - 2(y_1 - y_n)y &= d_1^2 - d_n^2 \\x_{n-1}^2 - x_n^2 - 2(x_{n-1} - x_n)x + y_{n-1}^2 \\- y_n^2 - 2(y_{n-1} - y_n)y &= d_{n-1}^2 - d_n^2\end{aligned}\quad (2.35)$$

โดยที่  $x_n, y_n$  เป็นตำแหน่งในแกน x และแกน y ของโหนดข้างอิงใด ๆ และ  $y$  เป็นตำแหน่งในแกน x และ  $y$  ของโหนดไม่ทราบค่าและทำการจัดภาพแบบให้อยู่ในเทอมของ

$$Xb = a \quad (2.36)$$

เมื่อ

$$X = \begin{pmatrix} 2(x_1 - x_n) & 2(y_1 - y_n) \\ 2(x_{n-1} - x_n) & 2(y_{n-1} - y_n) \end{pmatrix} \quad (2.37)$$

$$a = \begin{pmatrix} x_1^2 - x_n^2 + y_1^2 - y_n^2 - d_1^2 + d_n^2 \\ x_{n-1}^2 - x_n^2 + y_{n-1}^2 - y_n^2 - d_{n-1}^2 + d_n^2 \end{pmatrix} \quad (2.38)$$

$$b = \begin{pmatrix} x \\ y \end{pmatrix} \quad (2.39)$$

โดยสามารถหาค่า  $b$  ได้จาก  $b = (X^T X)^{-1} X^T a$

### 2.6.2 อัลกอริทึม Min-Max [4][16]

เป็นวิธีที่ใช้คำนวณหาตำแหน่ง บนพื้นฐานของการคำนวณระยะทางซึ่งสามารถแสดงสมการได้ดังนี้

$$RSSI = -10n \log_{10}(d) + A \quad (2.10)$$

เมื่อ RSSI คือค่าความแรงของสัญญาณวิทยุ มีหน่วยเป็นเดซิเบลล์มิลลิวัตต์ (-dBm)

$n$  คือค่าคงที่การกระจายสัญญาณของสถานที่นั้น ๆ

$d$  คือระยะห่างระหว่างเซนเซอร์ไร้สาย (เมตร)

$A$  คือค่าความแรงของสัญญาณที่ได้รับใน 1 เมตร

จากสมการข้างต้น ถ้าค่าระยะทาง ( $d$ ) เพิ่มมากขึ้นก็จะส่งผลให้ค่า RSSI ลดลงซึ่งบอกค่าความแรงของสัญญาณสามารถนำผลนี้ไปใช้เพื่อหาระยะทางและตำแหน่งของวัตถุ

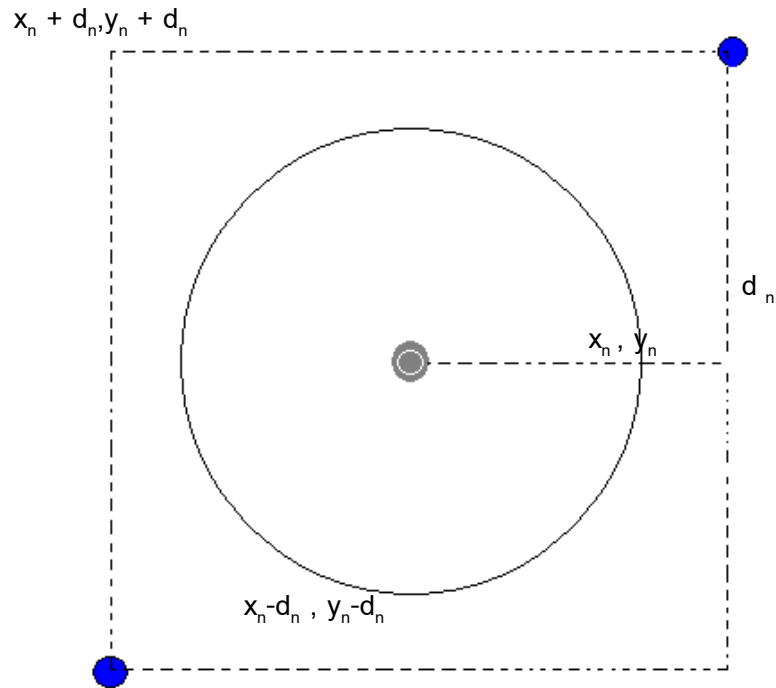
เมื่อคำนวณค่า  $d$  แล้วนำค่าที่ได้ไปหาค่ามากที่สุดและน้อยที่สุดของทุกโหนดอ้างอิงดังสมการ

$$x_{\max} = \begin{cases} x_{1\max} = x_1 + d_1 \\ x_{2\max} = x_2 + d_2 \\ x_{n\max} = x_n + d_n \end{cases} \quad (2.40)$$

$$y_{\max} = \begin{cases} y_{1\max} = y_1 + d_1 \\ y_{2\max} = y_2 + d_2 \\ y_{n\max} = y_n + d_n \end{cases}$$

$$x_{\min} = \begin{cases} x_{1\min} = x_1 + d_1 \\ x_{2\min} = x_2 + d_2 \\ x_{n\min} = x_n + d_n \end{cases}$$

$$y_{\min} = \begin{cases} y_{1\min} = y_1 + d_1 \\ y_{2\min} = y_2 + d_2 \\ y_{n\min} = y_n + d_n \end{cases}$$



ภาพที่ 2.20 แสดงการหาค่า Min – Max ระหว่างโหนดอ้างอิง[16]

ต่อมาจะนำค่ามากที่สุดและน้อยที่สุดไปหาค่าที่น้อยที่สุดของค่ามากที่สุดของพิกัด และหาค่าที่มากที่สุดของค่าน้อยที่สุดของพิกัด ดังสมการ

$$\begin{aligned}
 x_{\min-\max} &= \min(x_{\max}) \\
 x_{\max-\min} &= \max(x_{\min}) \\
 y_{\min-\max} &= \min(y_{\max}) \\
 y_{\max-\min} &= \max(y_{\min})
 \end{aligned}
 \tag{2.41}$$

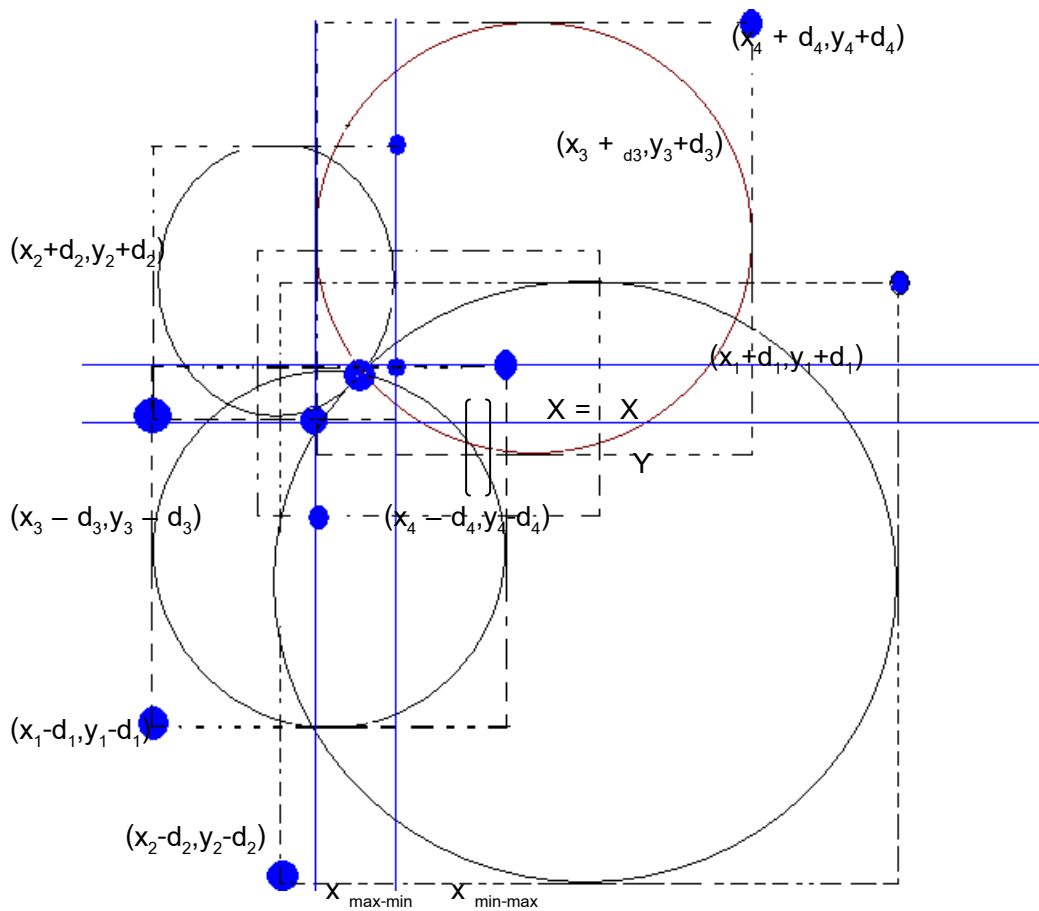
สามารถประมาณพิกัดได้เป็น

$$X = \begin{pmatrix} x \\ y \end{pmatrix}
 \tag{2.42}$$

โดยที่

$$\begin{aligned}
 x &= (x_{\min-\max} + x_{\max-\min}) / 2 \\
 y &= (y_{\min-\max} + y_{\max-\min}) / 2
 \end{aligned}
 \tag{2.43}$$





ภาพที่ 2.21 แสดงการหาตำแหน่งของวัตถุโดยใช้ Min – Max[16]

## 2.7 ข้อผิดพลาดที่เกิดจากการหาตำแหน่ง [4]

การเกิดข้อผิดพลาด (Error) จะถูกแบ่งออกเป็นสาม ประเภทหลัก คือ ฮาร์ดแวร์มี ปัญหา (Hardware related error) อาทิ เสาส่งสัญญาณ (Antenna) ข้อจำกัดของอัลกอริทึมการหาที่ตั้งของตำแหน่งตนเอง (The Limitations of the location-estimation) ซึ่งอาจก่อให้เกิดข้อผิดพลาด (Error) จากการประมาณการณค่าต่ำสุดของระยะ (Range) จากส่วนที่เป็นทั้ง Local และ Global minimum ซึ่งถ้าเลือกให้เป็นแบบ Global จะส่งผลให้ได้ค่าต่ำสุดในทุก ๆ ค่าที่เป็นไปได้ซึ่งจะมีตัวแปรที่ครอบคลุมกว่าแบบที่เป็น Local minimum และ สิ่งกระทบจากภายนอก (Effect of environment) เมื่อเกิดการเปลี่ยนแปลงที่อยู่ของโหนดเคลื่อนที่ และเสาทำหน้าที่กระจายสัญญาณในทุกทิศทาง เมื่อสิ่งแวดล้อมเปลี่ยนแปลงไปก็ย่อมส่งผลต่อค่าของสัญญาณได้ (RSSI)

### 2.7.1 อัลกอริทึมการหาข้อผิดพลาด [4]

เป็นอัลกอริทึมที่ทำงานร่วมกับเทคนิคที่ใช้พื้นฐานในการคาดคะเน (Location Fingerprint Technique) ในวิธีการระบุตำแหน่งแบบการคาดคะเนจากค่า RSSI ที่ระบบได้ทำการเรียนรู้ไว้แล้ว นั้นจะเรียกวิธีการนี้ว่า Fingerprint โดยจะทำการเก็บค่าไว้ที่ทุก ๆ 1 เมตรตลอดพื้นที่ที่ทำการทดลอง ดังสมการ

$$LF = \begin{bmatrix} LF_1 \\ \vdots \\ LF_n \end{bmatrix} = \begin{bmatrix} RSSI_{11} & RSSI_{12} & RSSI_{13} & RSSI_{14} \\ \vdots & \vdots & \vdots & \vdots \\ RSSI_{n1} & RSSI_{n2} & RSSI_{n3} & RSSI_{n4} \end{bmatrix} \quad (2.44)$$

และอยู่ในตำแหน่งดังต่อไปนี้

$$X(LF) = \begin{bmatrix} X(LF_1) \\ \vdots \\ X(LF_n) \end{bmatrix} = \begin{bmatrix} X_1, Y_1 \\ \vdots \\ X_n, Y_n \end{bmatrix} \quad (2.45)$$

มีวิธีการหาสองวิธีด้วยกัน คือ

### 2.7.2 วิธีการหาข้อผิดพลาดน้อยที่สุด (Nearest Neighbor Methods) [19]

วิธีการ Nearest Neighbor Method เป็น Deterministic Algorithms เนื่องจากต้องการเซตของค่าคงที่ของ Location Fingerprint ซึ่งก็คือค่า Mean Vectors และ Standard Deviation Vectors ของ RSS ในการหาตำแหน่งภาพแบบของ Discriminate Function คือการใช้การจำแนกตัวอย่างร่วมกันของ RSS Fingerprint ที่ตำแหน่งหรือ Case-Based Methods [20] เพราะว่าการจำแนกข้อมูลแต่ละตำแหน่งขึ้นอยู่กับ Case หรือ Class โดยชื่อของอัลกอริทึมที่ใช้วิธีการนี้ที่นิยมนำมาใช้ได้แก่ Euclidean Distance [20] และ Manhattan Distance [21] โดยวิธีการดังกล่าวจะเน้นกระบวนการในการหาตำแหน่งจากค่าความผิดพลาดน้อยที่สุด ระหว่างตำแหน่งที่ได้เรียนรู้กับโหนดไม่ทราบค่า

เมื่อทำการระบุตำแหน่งจะทำการตั้งค่า RSSI ระหว่างโหนดอ้างอิงต่าง ๆ กับโหนดไม่ทราบค่าออกมา

$$OB = [RSSI_1 \quad RSSI_2 \quad RSSI_3 \quad RSSI_4] \quad (2.46)$$

หลังจากนั้นจะนำค่า RSSI ระหว่างโหนดอ้างอิงต่าง ๆ กับโหนดเป้าหมายมาเปรียบเทียบ ค่าที่ได้จะเก็บไว้เพื่อหาตำแหน่ง

$$E = \begin{bmatrix} E_1 \\ \vdots \\ E_n \end{bmatrix} \quad (2.47)$$

$$= \sqrt{(RSSI_{11} - RSSI_1)^2 + (RSSI_{12} - RSSI_2)^2 + (RSSI_{13} - RSSI_3)^2 + (RSSI_{14} - RSSI_4)^2}$$

$$\sqrt{(RSSI_{n1} - RSSI_1)^2 + (RSSI_{n2} - RSSI_2)^2 + (RSSI_{n3} - RSSI_3)^2 + (RSSI_{n4} - RSSI_4)^2} \quad (2.48)$$

ซึ่งตำแหน่งที่ได้จะเป็นตำแหน่งของ LF ที่มีค่าความผิดพลาดน้อยที่สุด

$$X(OB) = X(\min(E)) \quad (2.49)$$

ข้อดี คือความเร็วในการหาตำแหน่งและไม่จำเป็นต้องปรับแต่งอัลกอริทึม ข้อจำกัด อัลกอริทึมนี้ไม่เหมาะสมที่นำไปใช้วัดบริเวณที่มีพื้นที่กว้าง เนื่องจากจะทำให้การคำนวณมีความซับซ้อนมากขึ้น เดิมอัลกอริทึมดังกล่าวใช้วิธีการคำนวณอย่างง่ายหรือใช้เพียงแค่จับคู่ข้อมูลกันเท่านั้น

### 2.7.3 วิธีการหาข้อผิดพลาดน้อยที่สุด 3 ค่า (3- Nearest Neighbor Methods) [19]

ลักษณะคล้ายกันกับข้อ 2.7.2 แต่วิธีนี้จะใช้ค่าความผิดพลาดที่น้อยที่สุด 3 ค่ามาหาค่าตำแหน่งเฉลี่ย

$$X(OB) = (X(\min_1(E)) + X(\min_2(E)) + X(\min_3(E)))/3 \quad (2.50)$$

### 2.7.4 วิธีการหาข้อผิดพลาดด้วยสมการยูคลิดีเนียน (Euclidian Distance)[18]

เป็นวิธีการคำนวณหาภาพแบบที่ใกล้เคียงกับตำแหน่งจริงของวัตถุ โดยระยะห่างของค่า E น้อยที่สุดจะถือว่าภาพแบบดังกล่าวมีความใกล้เคียงกับตำแหน่งจริงมากที่สุด โดยสูตรระยะห่างยูคลิดีเนียน สามารถแสดงได้ดังนี้

$$E = \sqrt{(x_0 - x_n)^2 + (y_0 - y_n)^2} \quad (2.51)$$

โดยที่ E คือค่าความผิดพลาดของตำแหน่งที่อ้างอิง

$x_0$  คือ ตำแหน่ง X จริง ณ. จุดที่วัตถุอยู่ (Actual Position)

$y_0$  คือ ตำแหน่ง Y จริง ณ. จุดที่วัดอยู่ (Actual Position)

$x_n$  คือ ตำแหน่ง X ที่หามาได้จากการคำนวณ

$y_n$  คือ ตำแหน่ง Y ที่หามาได้จากการคำนวณ

เมื่อค่า e (Error) น้อยสรุปได้ว่ามีความผิดพลาดน้อยหรือค่าคลาดเคลื่อนจากตำแหน่งจริงเพียงเล็กน้อย ทางผู้วิจัยเลือก วิธีการวัดค่าผิดพลาดด้วยสมการยูคลิเดียน เนื่องจากวิธีการนี้มีวิธีการคำนวณที่ไม่ซับซ้อนและสามารถนำมาช่วยประเมินค่าความแม่นยำของอัลกอริทึมหาตำแหน่งได้

## บทที่ 3

### การวิเคราะห์และออกแบบระบบ

#### 3.1 การวิเคราะห์ระบบ

ในระบบการค้นหาวัตถุจำเป็นต้องใช้ตัวแปร และอัลกอริทึมมาช่วยในกระบวนการค้นหาเพื่อให้ได้มาซึ่งพิกัดตำแหน่งที่มีความถูกต้อง และแม่นยำ กรณีข้อมูลตัวแปรนำเข้ามีความผิดพลาดเมื่อนำมาเข้าสู่กระบวนการคำนวณตำแหน่งผลลัพธ์ที่ได้เกิดความคลาดเคลื่อนด้วยเช่นกัน

ตัวแปรดังกล่าวคือ ค่าความแรงสัญญาณที่ดักจับมาได้จากการสื่อสารเป็นข้อความโปรโตคอลของอุปกรณ์ 3 ตัวคือ Gateway , Blind node และ Reference node ซึ่งทั้งสามประเภทนี้จะสื่อสารกันด้วยข้อความโปรโตคอล Requester (ClusterID:0x0011) และโปรโตคอล Responder(ClusterID:0x0012) จึงจำเป็นต้องคัดกรองเอาเฉพาะชุดโปรโตคอลที่สื่อสารระหว่าง Blind node กับ Reference node เท่านั้น และทำการตัดเฉพาะใบที่ที่ต้องการคือ ใบที่ร้องสุดท้ายมาทำการแปลงให้เป็นตัวแปรที่จะส่งออกไปคำนวณหาตำแหน่งได้อย่างถูกต้อง ซึ่งในกระบวนการค้นหาตำแหน่งของวัตถุ โดยทางทฤษฎีจะเกี่ยวข้องกับการนำสัญญาณมาใช้ในการคำนวณโดยทำให้สามารถประเมินระยะทางระหว่างโหนดตัวส่งและโหนดตัวรับ กรณีค่าระยะทางอยู่ไกลกันมากก็จะส่งผลให้ค่าสัญญาณลดน้อยลงด้วยเช่นกันซึ่งค่าความแรงของสัญญาณสามารถนำไปใช้เพื่อหาระยะทางและตำแหน่งของวัตถุ ในปัจจุบันมีเทคนิคมากมายที่นำมาใช้ในการหาตำแหน่ง แต่งานวิจัยนี้ ผู้วิจัยมีความสนใจและนำอัลกอริทึมที่มีพื้นฐานการคำนวณคล้ายกับอัลกอริทึมสามเหลี่ยมระยะแต่วิธีการได้ถูกปรับปรุงให้มีประสิทธิภาพกว่า คืออัลกอริทึม Maximum Likelihood เนื่องจากมีแนวคิดที่คล้ายกันแต่ต่างกันตรงที่จุดรับต้องมากกว่า 4 จุดและนำวิธีเชิงเมทริกซ์เชิงคณิตศาสตร์มาใช้ในการประเมินตำแหน่งวัตถุจึงทำให้วิธีนี้สามารถให้คำตอบที่เป็นพิกัด X,Y ได้มากกว่าอัลกอริทึมของสามเหลี่ยมระยะ[27] อัลกอริทึม Min-Max เป็นอัลกอริทึมที่มีวิธีการคำนวณไม่ยุ่งยากและซับซ้อน แต่ให้ผลลัพธ์ที่มีค่าแม่นยำและมีความน่าเชื่อถือ ซึ่งจะใช้โหนดรับตั้งแต่ 3 จุดขึ้นไป หลักการหาค่าแบบเฉลี่ยเชิงคณิตศาสตร์ของระยะกรณีตามแนวแกน X จะเลือกพิกัดตามแนวแกน Xmax และ Xmin โดยเลือกให้ค่าอยู่ภายใต้กรอบสี่เหลี่ยมและแนวแกน

Y ทำเช่นเดียวกันทั้ง  $Y_{max}$  และ  $Y_{min}$  คำตอบพิกัดจะได้มาจากการหาค่าเฉลี่ยของทั้ง 2 แกนจะ  
ได้มาเป็นตำแหน่งพิกัดที่ดีที่สุด คือ X และ Y ของตำแหน่งวัตถุนั้น[26]

### 3.2 แนวคิดในการพัฒนา

จากการศึกษาเรื่อง การระบุพิกัดโดยการประมาณค่า (Proximity localization) ทาง  
ผู้วิจัยสนใจที่จะใช้เครือข่ายไร้สายในมาตรฐาน Zigbee มาใช้ในการพัฒนาเพื่อทำการระบุ  
ตำแหน่ง โดยจะนำเอาพารามิเตอร์ที่อยู่ภายในโปรโตคอลที่ถูกต้องโดยผ่านกระบวนการการคัด  
กรองแล้วมาทำการแปลงสัญญาณให้สามารถนำมาใช้ในการคำนวณและประเมินตำแหน่งได้  
อย่างแม่นยำและมีความใกล้เคียงกับตำแหน่งจริงมากที่สุด

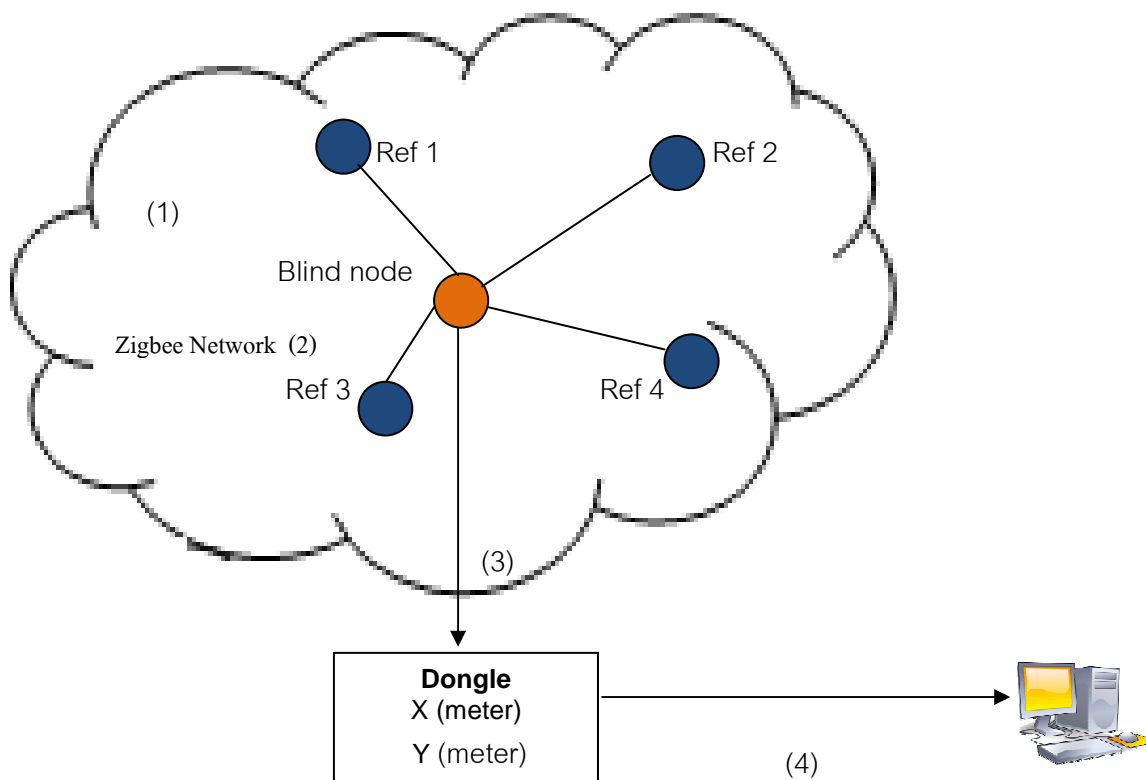
ระบบระบุตำแหน่งของงานวิจัยนี้แบ่งออกเป็น 2 ส่วนคือ ส่วนของชุดอุปกรณ์เซ็นเซอร์ไร้  
สายและส่วนของโปรแกรมประยุกต์ที่ใช้แสดงผล ชุดอุปกรณ์เซ็นเซอร์ไร้สายที่ใช้ในการวิจัย ได้แก่  
ชุดพัฒนา Zigbee TI CC2431 ของบริษัท Texas Instruments จำนวน 1 ชุดโดยแบ่งออกเป็น  
บอร์ด SmartRF@04EB จำนวน 2 บอร์ด ซึ่งเป็นบอร์ดที่ทำหน้าที่ในการเชื่อมต่อและส่งข้อมูลเข้า  
กับเครื่องคอมพิวเตอร์เพื่อสำหรับใช้แสดงผล เสาอากาศสำหรับใช้ทำการติดต่อสื่อสารผ่านอากาศ  
(2.4GHz antennas) 12 ชุด และแบตเตอรี่ (SoC\_BB) 10 ชุด และ Evaluation modules  
CC2431EM 10 ชุด คือโหนดอ้างอิง (Reference node) และ CC2430EM 2 ตัวสำหรับโหนด  
เป้าหมาย (Blind node) และในส่วนของชุดพัฒนาซอฟต์แวร์ คือ เครื่องมือสำหรับพัฒนาโปรแกรม  
บนชิป CC2430 และ CC2431 คือ IAR Embedded Workbench , โปรแกรมสำหรับฝัง source  
embedded คือ SmartRF Studio 7 , โปรแกรมสำเร็จรูปใช้ในการระบุตำแหน่งของ TI คือ Z-  
Location Engine

**ตัวรับส่งสัญญาณวิทยุ (Transceiver)** ได้เลือกใช้โมดูล TIC2430-31 ของบริษัท Texas  
Instruments ซึ่งมีจุดเด่นอยู่ที่ มีข้อมูลรีจิสเตอร์ ในมาตรฐานของ Zigbee และสามารถสร้าง  
Routing Topology ได้หลายแบบและง่ายต่อการใช้งานและกำหนดค่า โดยทำการเชื่อมต่อกับ  
ไมโครคอนโทรลเลอร์ผ่านโมดูล Evaluation และสามารถกำหนดค่าได้โดยโปรแกรมของทาง  
บริษัทผู้ผลิต

### 3.3 การออกแบบระบบ

#### 3.3.1 ภาพรวมของระบบ

จากโหนดอ้างอิง (Reference node) ถูกแปะติดไว้ตามจุดต่าง ๆ ที่ทราบตำแหน่งแน่นอนภายในอาคาร และโหนดเป้าหมาย (Blind node) ติดอยู่กับตัวอุปกรณ์ที่เราต้องการทราบตำแหน่ง โหนดเป้าหมายสื่อสารกับโหนดอ้างอิงโดยผ่านโปรโตคอลซิกบี และทำการคำนวณตำแหน่งตนเองและส่งข้อมูลนี้ไปที่จุด Dongle/Gateway ซึ่งติดอยู่กับคอมพิวเตอร์



ภาพที่ 3.1 ภาพรวมของระบบ

#### อธิบายขั้นตอนภาพรวมของระบบ

จากภาพรวมของระบบ อธิบายขั้นตอนได้ดังต่อไปนี้

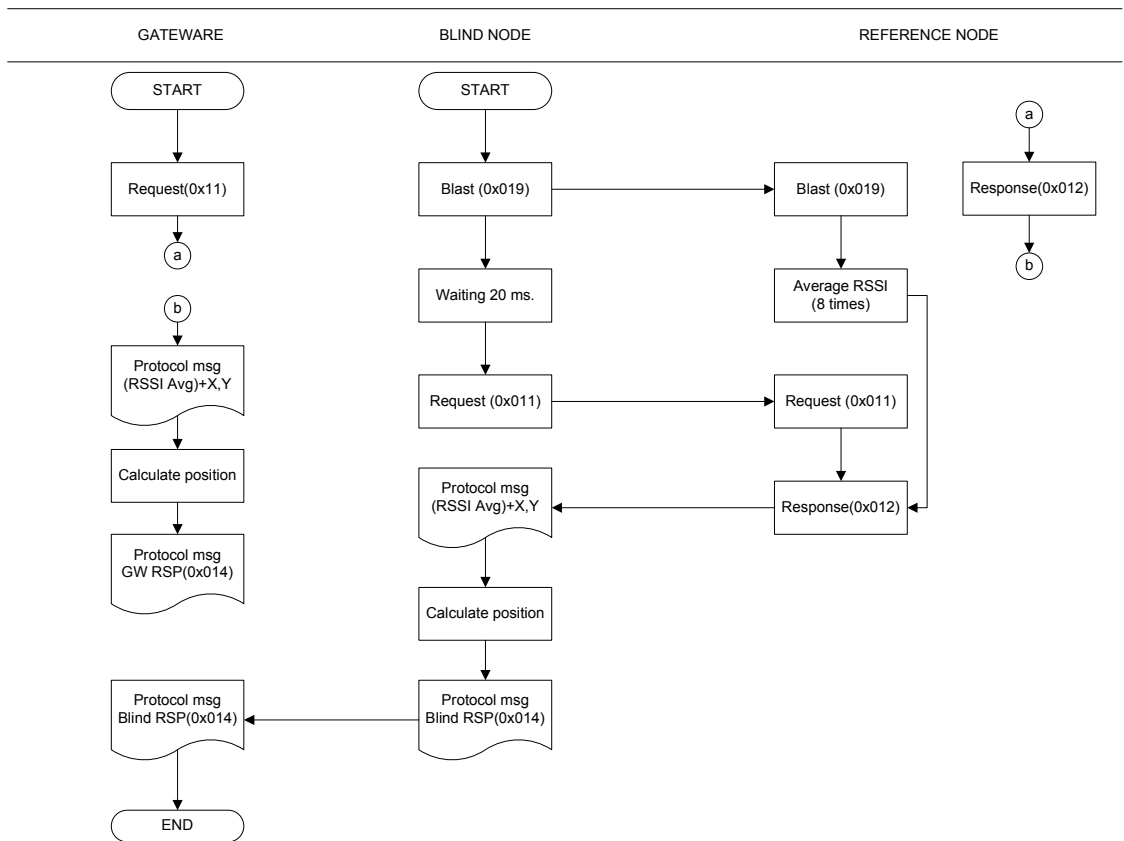
- (1) ใช้เครือข่ายไร้สายในมาตรฐาน Zigbee มาใช้ในการทดลองเพื่อทำการระบุตำแหน่งด้วยการติดตั้งแท็ก 2 ประเภท
  - โหนดอ้างอิง (Reference node) หรือ CC2431EM คือ โหนดวางประจำไว้ในแต่ละตำแหน่งและเก็บค่าพิกัดของตัวเองไว้ในรูป X, Y 2 มิติ จากภาพใช้ตัวย่อว่า Ref1 , Ref2 , Ref3 และ Ref4

- โหนดเป้าหมาย (Blind node) คือบอร์ดตัวที่ 1 SmartRF04EB (1) โหนดเป้าหมายจะได้รับค่าตำแหน่ง X,Y และค่าความแรงสัญญาณ (RSSI) หน่วยเป็น -dBm ค่า Index จากโหนดอ้างอิงทุกตัว และโหนดเป้าหมาย (Blind node) จะทำการคำนวณโดยใช้ระยะทาง (Range based) กับค่าความแรงของสัญญาณจากโหนดอ้างอิง เพื่อหาตำแหน่งเป็นพิกัดของตำแหน่งของโหนดเป้าหมายได้ในหน่วยเมตร
- (2) หลังจากคำนวณเรียบร้อยแล้วต่อไปคือ การส่งผลลัพธ์ (output) ออกมาโดยผ่านสัญญาณโปรโตคอลซิกบี มีค่า X,Y ที่บอกพิกัดมาในรูปแบบของเมตร
- (3) ข้อมูลดังกล่าวถูกส่งต่อมาที่บอร์ดตัวที่ 2 SmartRF04EB (2) หรือ Dongle ที่ทำหน้าที่เสมือน Gateway และส่งผลลัพธ์ออกไปที่คอมพิวเตอร์
- (4) รับค่าผลลัพธ์ (output) ส่งแสดงผลออกทางจอภาพ คอมพิวเตอร์มีซอฟต์แวร์ Z-Location Engine Screenshot ซึ่งนำข้อมูลนี้มาแสดงตำแหน่งของโหนดเป้าหมายเป็นภาพ

### 3.3.2 การออกแบบและปรับเปลี่ยนโปรโตคอลสื่อสารบน Hardware

กระบวนการสื่อสารของอุปกรณ์ระหว่าง Reference node , Blind node และ Gateway จะสื่อสารกันด้วย Protocol messages ที่เกิดขึ้นได้จากทั้ง 3 ประเภท ดังภาพ 3.2



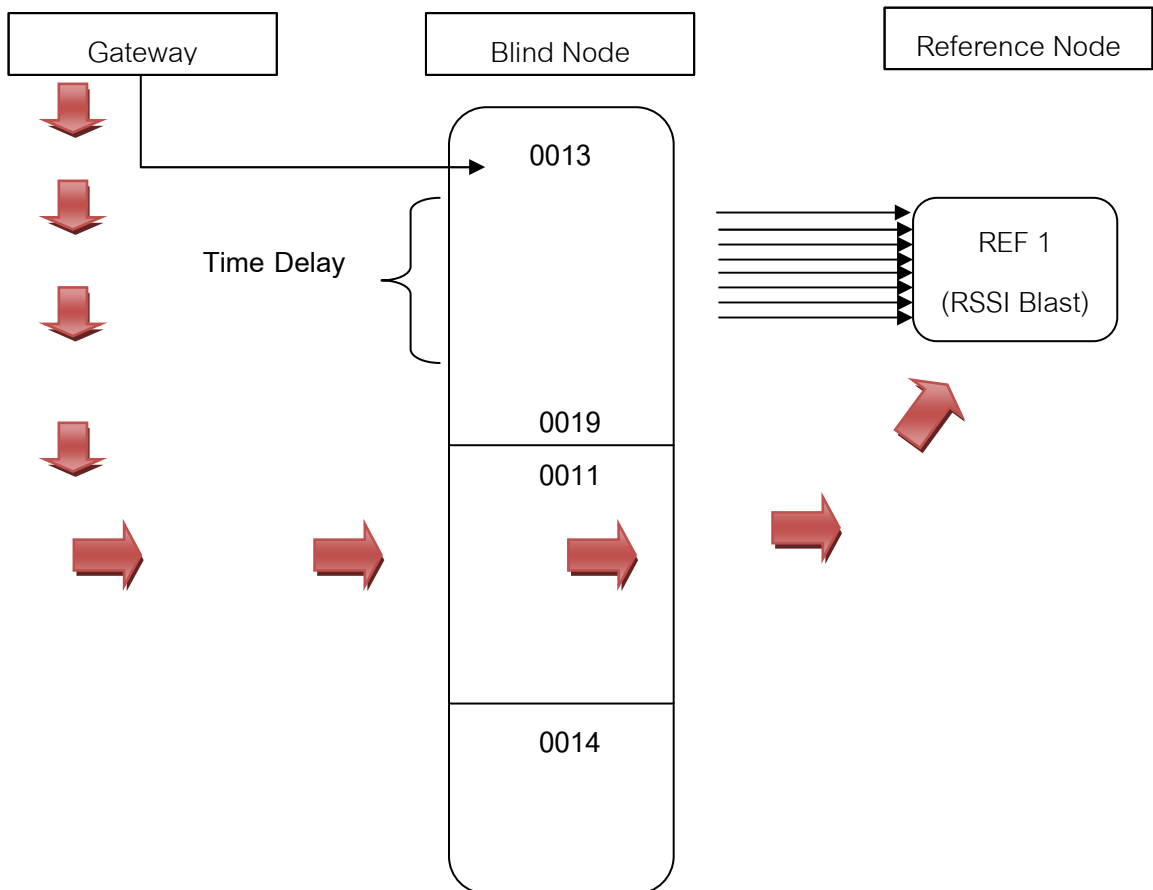


ภาพที่ 3.2 แสดงการรับ-ส่ง Protocol Messages ของการสื่อสารระหว่างอุปกรณ์ Gateway , Blind Node และ Reference Node

สามารถอธิบายขั้นตอนการรับและส่งโปรโตคอลได้ ดังต่อไปนี้

เริ่มจาก Blind node ทำการส่ง Protocol Blast Messages 8 ครั้งกระจายออกไปยัง Reference node ในทุกจุดพื้นที่ที่ได้ทำการติดตั้งประจำไว้ตามแต่ละพื้นที่ทุกตัวที่สามารถรับสัญญาณได้ ซึ่งเมื่อ Reference node ได้ยินจึงทำการเฉลี่ยค่าความสัญญาณ 8 ครั้งเก็บไว้ที่ตัว Reference node แบบชุด Package ขณะที่ Blind node รอให้ Reference node ทำการเฉลี่ย (20 วินาที) ส่ง Request messages (0x011) เพื่อไปขอชุดโปรโตคอลกับตำแหน่ง Reference node แต่ละตัว เมื่อ Reference node ได้รับ Request messages (0x011) มันก็จะเตรียมส่ง Protocol Message (0x012) มาเป็นชุดโปรโตคอลพร้อมค่า X,Y ของ Reference node ในแต่ละตัวกลับไป จากนั้น Blind node คำนวณหาตำแหน่ง ตอบเป็น Protocol Message (0x014) ไปยัง Gateway เพื่อแสดงผล ในขณะเดียวกัน Gateway สามารถส่ง Request Messages และรอรับ Response Message ได้เช่นเดียวกันกับ Blind node ด้วยเหตุนี้จึงเป็นเหตุให้ Protocol Messages ได้รับตอบกลับมา 2 แหล่งคือ Gateway และ Blind Node เพื่อความถูกต้องของการนำเข้าสู่ชุดข้อมูล

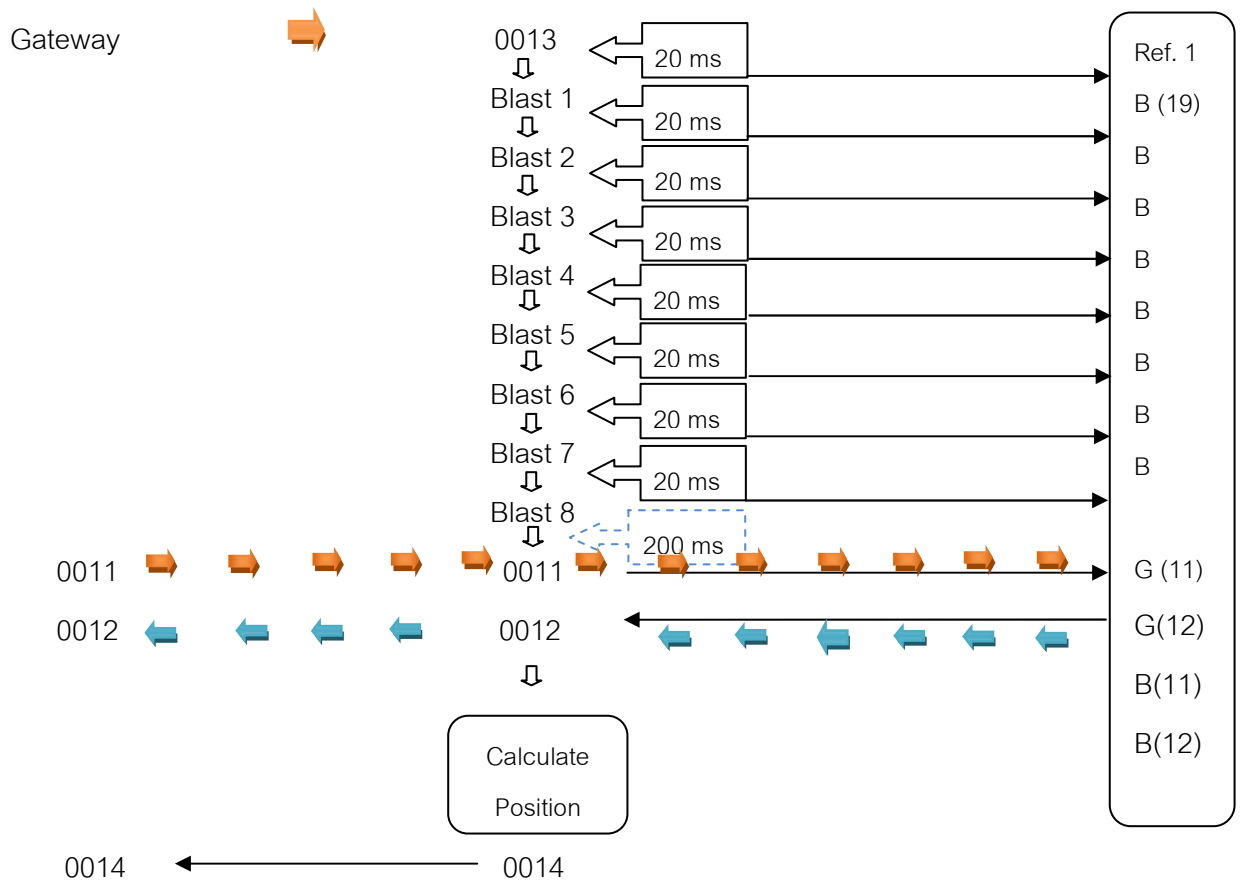
โปรโตคอล ที่ต้องการเฉพาะชุดข้อมูลจาก Blind node Response เท่านั้นจึงต้องมีการแก้ไขใน ส่วนของ Source code Embedded และทำการปิดช่องทางในบางคำสั่งบน Method โปรแกรม



ภาพที่ 3.3 แสดงอธิบายขั้นตอนการรับและส่ง Protocol Messages

#### กระบวนการทำงานของ Smartboard และ Chipcon CC2430 กับ CC2431

Gateway จะส่ง 0013 ไปยัง Blind node เมื่อ Blind node ได้รับก็จะทำการ Blast ค่า RSSI 8 ครั้งมายัง Reference node จากนั้น Reference node จะทำการหาค่าเฉลี่ย คำนวณใช้เวลาอยู่ประมาณ 200 Ms (Time Delay) เพื่อให้ Reference node ทำการคำนวณหาค่า RSSI Blast ที่ Reference node ได้ยินมาจาก Blind ทั้งหมด 8 ครั้งเก็บไว้ใน memory ของตัว Reference เอง จากนั้น Blind node จะทำการ Request 0011 ไปหา Reference node ซึ่ง Reference node จะทำการคิดคำนวณค่า RSSI (AVG) = RSSI Blast + RSSI 0011 แล้วส่งต่อกลับไปให้ Blind node เมื่อ Blind node ได้รับทำการคำนวณหาตำแหน่งเสร็จตอบ 0014 กลับไปยัง Gateway พร้อมทั้งแสดงตำแหน่งบน Map



ภาพที่ 3.4 อธิบายขั้นตอนการรับและส่ง Protocol Messages ระหว่าง Blind node และ Reference node กับ Gateway

**การทำงานของ Message Protocol**

- 1) เริ่มต้นจาก Gateway ส่ง 0013 ไปให้ Blind node เมื่อได้รับมันจะทำ 0019 กระจายออกไป 8 ครั้งต่อ 1 Reference node
- 2) เมื่อ Blind ทำการส่ง Blast ออกไปหา Reference node เสร็จเรียบร้อยแล้ว Blind ก็จะส่ง message 0011 ออกไป
- 3) เมื่อ Reference node ได้รับ Blast มาครบ 8 ครั้งมันจะคำนวณหาค่าเฉลี่ยภายในตัวมันเอง จากนั้นมันก็จะขอ message request 0011 เมื่อมันได้ message request 0011 มา มันจะทำการคำนวณโดยเอา Blast ที่ Reference เก็บไว้ใน memory + message request 0011 จะส่ง 0012 กลับไป

### รูปแบบ Protocol messages

#### สำหรับ Blind node

- Protocol จะต้องเริ่มด้วย msg เริ่มต้นดังนี้ : 0x02 0x10 0x18 0x14 0xCB
- วิธีการคือ นับจำนวน Length ของ byte ไม่เกิน 25 หลัก เริ่มที่ 0xFF

2	10	18	14	CB	5F	3	D3	14	0	0D	0	21	0	7	0	4	2	0	28	0	0	0	34	7D
---	----	----	----	----	----	---	----	----	---	----	---	----	---	---	---	---	---	---	----	---	---	---	----	----

ภาพที่ 3.5 รูปแบบโปรโตคอลของ Blind node response messages (0x014)

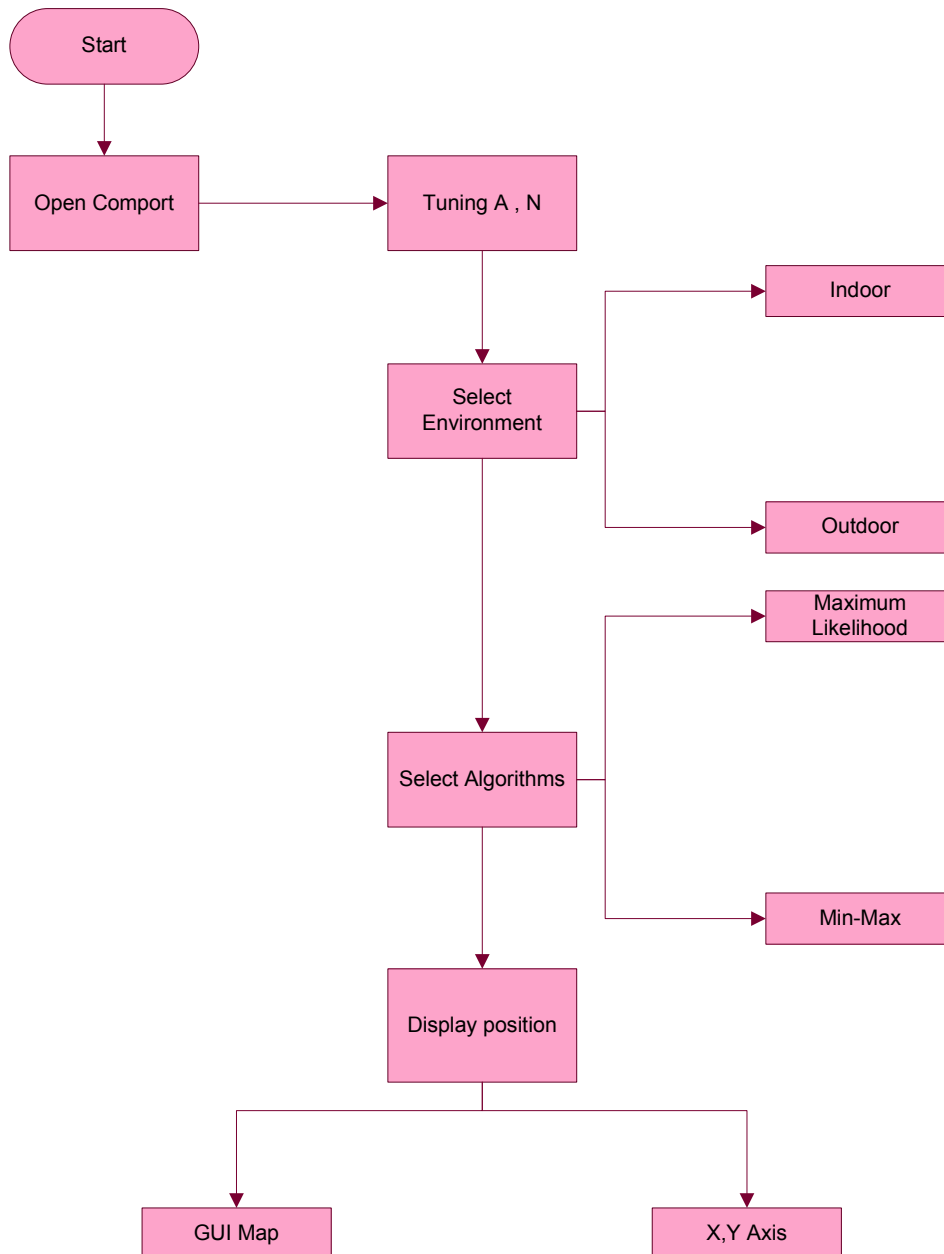
#### สำหรับ Reference node

- Protocol จะต้องเริ่มด้วย msg เริ่มต้นดังนี้ : 0x02 0x10 0x18 0x0C 0xCB 0x73
- วิธีการคือ นับจำนวน Length ของ byte ไม่เกิน 17 หลัก เริ่มที่ 0xFF

2	10	18	0C	CB	1	0	D2	12	0	5	0	0	0	0	43	48
---	----	----	----	----	---	---	----	----	---	---	---	---	---	---	----	----

ภาพที่ 3.6 รูปแบบโปรโตคอลของ Reference node response messages (0x012)

### 3.4 การออกแบบการทำงานของโปรแกรม



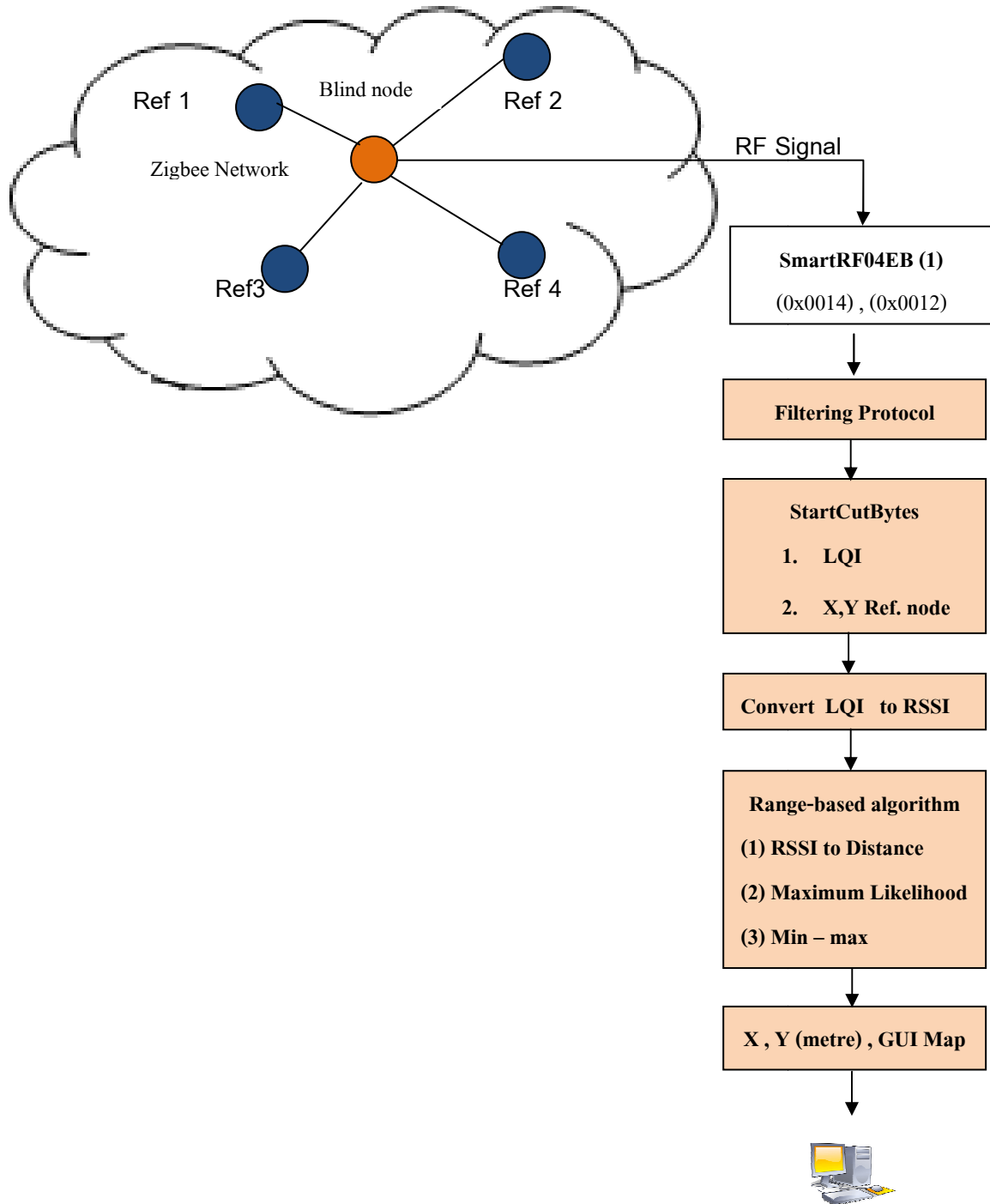
ภาพที่ 3.7 รูปแบบโครงสร้างการทำงานของโปรแกรม

สามารถอธิบายโครงสร้างได้ ดังต่อไปนี้

เริ่มต้นจากการ เชื่อมต่อระหว่างชุดอุปกรณ์กับโปรแกรมผ่าน Com Port เลือกเมนูสภาพแวดล้อม ในขณะนั้นเป็นประเภท Indoor หรือ Outdoor ค่า A และ N ระบบจะ Set ค่าอัตโนมัติแต่กรณี ต้องการเปลี่ยนแปลงสามารถปรับแก้ได้ เลือกอัลกอริทึมที่จะใช้หาตำแหน่ง ผลลัพธ์แสดงเป็น 2 ส่วนคือ พิกัด X,Y หรือแบบแสดงออกทางแผนที่

### 3.5 สถาปัตยกรรมระบบ

จากภาพรวมของระบบ จะนำมาทำการออกแบบเป็นสถาปัตยกรรมระบบดังภาพที่ 3.8 โดยจะประกอบด้วยส่วนการจัดการข้อมูลและส่วนแสดงผล



ภาพที่ 3.8 แสดงการออกแบบสถาปัตยกรรมระบบ

### สถาปัตยกรรมระบบประกอบด้วย

- (1) เริ่มต้นจาก Blind node ส่ง Protocol Blast Messages (0x019) ไปยัง Reference node ทุกตัว หลังจากนั้นเมื่อ Reference node ทำการเฉลี่ยค่า RSSI และรอจังหวะ Request Message (0x011) จาก Blind node ซึ่ง Blind node กำหนดไว้ที่ 20 วินาทีเมื่อครบแล้วก็จะส่ง Messages ดังกล่าวออกไปจากนั้น เมื่อ Reference node ได้รับ Message Request ดังกล่าวก็จะนำชุดข้อมูลส่งเป็นแบบ Package Protocol พร้อมกับตำแหน่งกลับไปยัง Blind node
- (2) Messages Protocol ถูกส่งมาเก็บไว้ที่ Smartboard RF04eb
- (3) โปรแกรม Location Engine ทำการคัดกรองโปรโตคอลที่มีความสมบูรณ์และถูกต้องตามมาตรฐาน
- (4) โปรแกรมนำชุดโปรโตคอลที่ถูกต้องมาทำการตัดเอาเฉพาะไบท์ที่ต้องการคือ รองไบท์สุดท้าย คือไบท์ที่เก็บค่า LQI
- (5) นำ LQI มาทำการแปลงเป็นค่า RSSI
- (6) นำไปคำนวณแปลง RSSI เป็น Distance และส่งเข้าไปคำนวณต่อในสมการทางคณิตศาสตร์ Maximum Likelihood และ Min-Max
- (7) ได้พิกัดออกมาเป็นคำตอบ X,Y และนำผลไปแสดงต่อหน้าจอแผนที่ (GUI Map)

### 3.6 การปรับเปลี่ยนค่าตัวแปร A , N ให้เหมาะสม

ผู้ใช้งานสามารถปรับเปลี่ยนค่าตัวแปร A และ N ให้สอดคล้องกับสิ่งแวดล้อมในขณะนั้นได้เนื่องจากการใส่เมนูการปรับเปลี่ยนหรือสามารถเลื่อน Scroll bar ตามปกติระบบจะมีค่า Default ให้แล้วตามสภาพแวดล้อม

### 3.7 การประเมินประสิทธิภาพของการหาตำแหน่งวัตถุ

ผู้วิจัยจะทำการประเมินผลลัพธ์หลังจากคำนวณหาตำแหน่งพิกัด X,Y ด้วยอัลกอริทึม Maximum Likelihood , Min- Max และ TI ด้วยสมการวัดค่าความผิดพลาด (Euclidian Distance Error) ดังสมการบทที่ 2 ข้อ ( 2.51 )

## บทที่ 4

### การออกแบบและพัฒนาเครื่องมือการค้นหาตำแหน่งวัตถุ

ในบทนี้จะกล่าวถึงรายละเอียดการพัฒนาเครื่องมือเพื่อการค้นหาตำแหน่งวัตถุทั้งแบบภายในอาคารและภายนอกอาคาร โดยจะกล่าวถึงโครงสร้างของเครื่องมือ ส่วนประกอบของเครื่องมือ การแก้ส่วนของซอฟต์แวร์บนชิปครอน CC2430 และ CC2431 เพื่อให้อุปกรณ์เปลี่ยนทิศทางในการสื่อสาร โปรโตคอลสื่อสารระหว่างโหนดหลักการร้องขอและการตอบรับ การตรวจสอบและคัดกรองโปรโตคอลรวมทั้งเครื่องมือที่นำมาใช้ในการพัฒนาซึ่งมีรายละเอียดดังนี้

#### 4.1 การออกแบบองค์ประกอบของโปรแกรม

ประกอบด้วยโมดูลสำคัญ ดังต่อไปนี้

1. มีโมดูลสำหรับคัดกรองชุดโปรโตคอล ในกระบวนการสื่อสารระหว่าง Reference node และ Blind node จะมีโปรโตคอลหลากหลายรูปแบบ อาทิ Protocol Blast Messages (0x019) , Protocol Request Messages (0x011) , Protocol Response Messages (0x012) และ Protocol Blind Configure Request Messages (0x013) จะเลือกมาเฉพาะโปรโตคอลที่ถูกต้องและมีความสมบูรณ์ครบถ้วน
2. มีโมดูลการแปลงค่าสัญญาณ LQI (0-255) เป็นค่าความเข้มสัญญาณ RSSI (40 - 95) dBm .
3. มีโมดูลการนำค่าข้อมูลนำเข้า (Input Parameter) มาคำนวณและแปลงค่าเป็นระยะทาง (Distance)
4. มีโมดูลการรับค่าตัวแปรระยะทาง (Distance) มาคำนวณหาตำแหน่งด้วยอัลกอริทึม Maximum Likelihood และอัลกอริทึม Min-Max
5. มีโมดูลการแสดงผลตำแหน่งพิกัด (X,Y) และแสดงผลลัพธ์ออกมาเป็นแผนที่ (GUI Map)



## 4.2 สภาพแวดล้อมที่ใช้ในการพัฒนาเครื่องมือ

สภาพแวดล้อมที่ใช้ในการพัฒนาเครื่องมือ ผู้วิจัยได้ใช้ฮาร์ดแวร์ และซอฟต์แวร์ ซึ่งมีรายละเอียดดังนี้

### 4.2.1 ฮาร์ดแวร์

ฮาร์ดแวร์สำหรับพัฒนาเครื่องมือเพื่อใช้สำหรับค้นหาตำแหน่งวัตถุ มีรายละเอียดดังนี้

1. เครื่องคอมพิวเตอร์โน้ตบุ๊ก Asus K40INseries หน่วยประมวลผล Intel Core 2 Duo T6500 2.10 กิกะเฮิร์ตซ์
2. หน่วยความจำ (Memory) 4.0 กิกะไบต์
3. จานบันทึกแบบแข็ง (Hard disk) ความจุ 250 กิกะไบต์

### 4.2.2 ซอฟต์แวร์

ซอฟต์แวร์สำหรับพัฒนาเครื่องมือเพื่อใช้สำหรับค้นหาตำแหน่งวัตถุ มีรายละเอียดดังนี้

1. ระบบปฏิบัติการไมโครซอฟท์วินโดวส์ 7 อัลติเมต (Microsoft Windows 7 Ultimate)
2. Microsoft Visual Studio Express 2010 เครื่องมือที่ใช้พัฒนาโปรแกรมภาษา C#
3. Photoshop CS สำหรับสร้างแผนที่บนกริด GUI map
4. ภาษาซีชาร์ป (C# Language) รุ่น 5.2.6 สำหรับพัฒนาแอปพลิเคชัน
5. โปรแกรม Docklight Scripting 1.9 สำหรับดักจับ Messages Protocol
6. โปรแกรม Virtual Serial Port Emulator สำหรับการควบคุมการทำงานของ การเชื่อมต่อ Com Port ที่มีการเชื่อมต่อกันมากกว่า 1 คู่สาย
7. โปรแกรม IAR Embedded Workbench สำหรับพัฒนาต่อเติม source code บนอุปกรณ์ชิปคอน
8. โปรแกรม SmartRF studio 7 สำหรับใช้ Burn โปรแกรมลงบนชิปคอน
9. โปรแกรม LE Monitor สำหรับตรวจสอบการเคลื่อนไหวของโปรโตคอลเป็นแบบ Messages
10. โปรแกรม Package Sniffer สำหรับตรวจสอบการเคลื่อนไหวของโปรโตคอลเป็นแบบ Packages
11. โปรแกรม Z-Location Engine สำหรับนำค่า X,Y มาเปรียบเทียบกับผลการทดลอง

12. โปรแกรม SciLAB 5.3.3 สำหรับแปลงสมการของอัลกอริทึมที่ใช้ในการหาตำแหน่ง
13. โปรแกรม MathType สำหรับการพิมพ์สมการ

#### 4.3 การเชื่อมต่ออุปกรณ์

อุปกรณ์ทุกตัวอาทิ SmartRF04eb , CC2431 และ CC2430 จะสามารถสื่อสารกันได้ จำเป็นต้องทำการลงโปรแกรม Firmware โดยกำหนดดังนี้

- Blind node ใช้ Chipcon CC2430 และลงโปรแกรมโหมด End Device BB
- Reference node ใช้ Chipcon CC24301 และลงโปรแกรมโหมด Router BB
- SmartRF04eb ใช้ Chipcon CC2430 และลงโปรแกรมโหมด Coordinator EB

#### 4.4 หมวดโปรแกรมการทำงานของโปรโตคอล

โปรโตคอลที่ใช้สื่อสารระหว่างกันของอุปกรณ์ระหว่างตัวส่ง (Blind node) และตัวรับ (Reference node) ประกอบด้วยชุดโปรโตคอลดังนี้

- Cluster ID 0x0019 : สำหรับ Blind node ส่ง Blast Messages ไปหา Reference node ทุกตัวที่ถูกติดตั้งอยู่ตามจุดต่าง ๆ
- Cluster ID 0x0011 : สำหรับ Blind node หรือ Gateway ส่งไปร้องขอข้อมูลโปรโตคอลจาก Reference node
- Cluster ID 0x0012 : สำหรับ Reference node ใช้ตอบกลับ Requester
- Cluster ID 0x0013 : สำหรับ Blind node สำหรับแก้ไขค่า Configuration ตัวเอง
- Cluster ID 0x0014 : สำหรับ Blind node ตอบตำแหน่งที่ตัวเองคำนวณเสร็จเรียบร้อยแล้วกลับไปยัง Gateway [23]

ผู้วิจัยได้ปรับแก้ 2 ส่วน

ส่วนแรก : ส่วน Firmware สำหรับฝังลงบน Chipcon

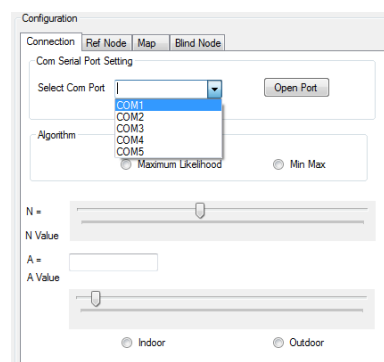
ได้แก้ไขส่วนของ Gateway ไม่ให้ส่ง Request Message (0x0011) ไปยัง Reference node และสร้าง Hex File เพื่อเป็น Firmware สำหรับติดตั้งบนตัวชิป (ภาคผนวก ข.)

ส่วนสอง : ส่วนการพัฒนากระบวนการค้นหาวัตถุ ด้วยการเพิ่ม StartCutByte Method มาตรวจสอบโดย โปรโตคอลต้องเรียงลำดับ ดังนี้

1. รับ Blind Response Cluster ID 0x0014  
รูปแบบ Response Message เริ่มต้นที่ 0x02 0x10 0x18 0x14 0xCB ไม่เกิน 25  
หลัก เริ่มที่ 0xFF
2. ส่ง Message Cluster ID 0x0019 (Trigger message)
3. ส่ง Message Cluster ID 0x0011 (Request Message) ประกอบด้วย 5 ไบท์  
ไบท์ 0 & 1 : ตำแหน่ง X ของ Reference node  
ไบท์ 2 & 3 : ตำแหน่ง Y ของ Reference node  
ไบท์ 4 : ค่าเฉลี่ย RSSI (RSSI Blast 8 ครั้ง)
4. รับ Message Cluster ID 0x0012 (Response Message)  
รูปแบบ Response Messgae เริ่มต้นที่ 0x02 0x10 0x18 0x0C 0xCB 0x73 และ  
ไบท์ไม่เกิน 17 หลักและต้องเริ่มที่ 0xFF

#### 4.2.3 หมวดโปรแกรม Configurations

ประกอบด้วยส่วนที่ทำการเชื่อมต่อกับ อุปกรณ์ Hardware ค่า Parameter A , N และการเลือกสภาพแวดล้อมแบบ 2 สถานการณ์คือ (Indoor , Outdoor) และการเลือก Algorithms เพื่อที่จะแสดงผล ระบบจะเชื่อมต่อระหว่างอุปกรณ์ที่เรียกว่า Gateway ที่ต่อพ่วงระหว่าง Smartboard RF04eb กับเครื่องคอมพิวเตอร์



รูปที่ 4.1 แสดงส่วนการเชื่อมต่อของ Hardware กับส่วนของโปรแกรม

#### 4.2.4 หมวดแปลงค่า LQI มาเป็น RSSI

นำชุดโปรโตคอล 0x0012 มาตัดเอาเฉพาะ Message รองไบท์สุดท้ายคือ LQI จะมีค่าอยู่ในช่วงระหว่าง 0-255 นำมาเข้าสู่ตรรกกรรมเพื่อแปลงเป็นค่า RSSI มีค่าอยู่ในช่วงระหว่าง -40 ถึง -95 (dBm) กรณีค่าเท่ากับ -81 คือค่าสัญญาณไม่สามารถติดต่อสื่อสารระหว่างกันได้  
ดั่งสมการด้านล่าง

$$RSSI = ((81 - LQI * 91) / 255)) \quad [15] \quad (4.1)$$

การแปลงค่า LQI เป็น RSSI			
Node	LQI	RSSI ตักมาจากโปรแกรม	RSSI เกิดจากการคำนวณ
1	0	81	81
2	0	81	81
3	123	38	37.10588235
4	145	30	29.25490196

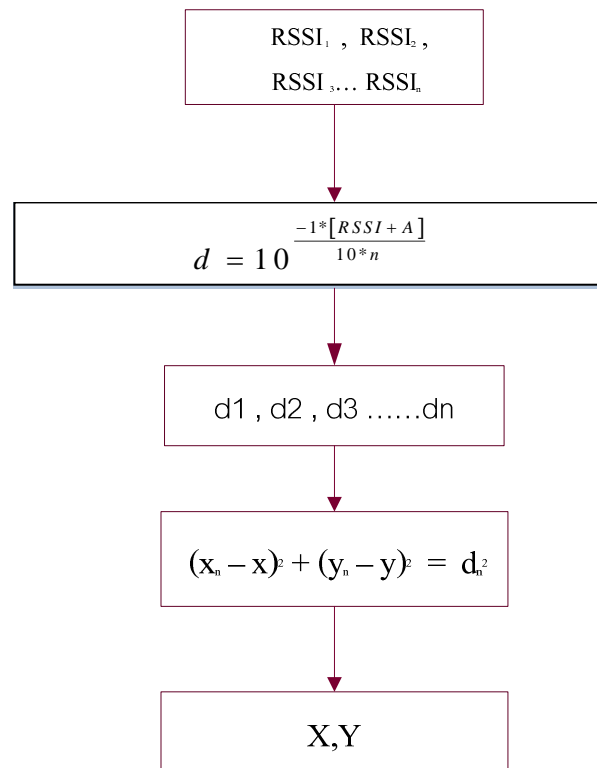
รูปที่ 4.1 ตัวอย่างหมวดการแปลงค่า LQI มาเป็น RSSI

#### 4.2.5 หมวดแปลงค่า RSSI เป็นระยะทาง (Distance) และคำนวณตำแหน่งด้วยอัลกอริทึม Maximum Likelihood และ Min-Max

หลังจากเก็บค่า RSSI จากหลายตำแหน่งได้หลายค่าตามแนวพื้นที่การทดลอง เอาค่า RSSI มาคำนวณหาค่า  $d$  ด้วยสมการด้านล่าง

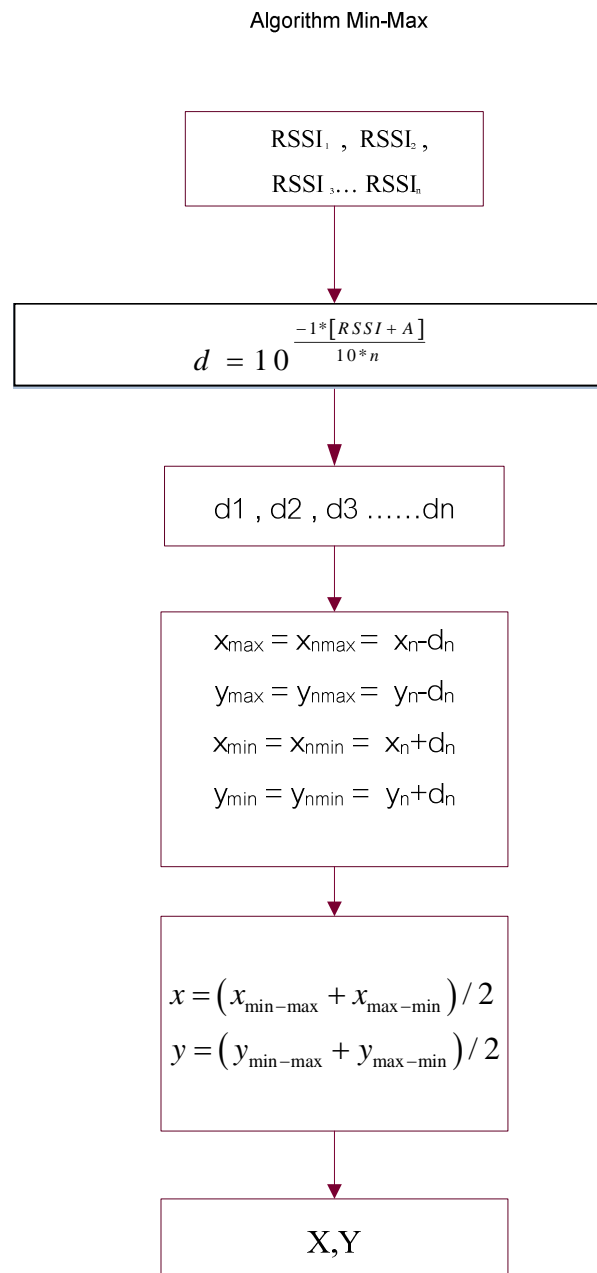
$$RSSI = -10n \log_{10}(d) + A \quad (4.2)$$

## Algorithm Maximum Likelihood



รูปที่ 4.2 แสดงขั้นตอนการนำค่า RSSI มาแปลงเป็น Distance หาตำแหน่งด้วยอัลกอริทึม

Maximum Likelihood



รูปที่ 4.3 แสดงขั้นตอนการนำค่า RSSI มาแปลงเป็น Distance หาตำแหน่งด้วยอัลกอริทึม Min-Max

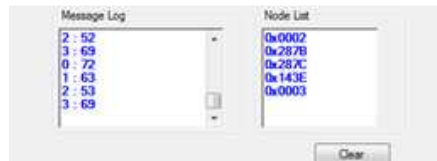
หลังจากคำนวณพิกัดวัตถุได้ทั้ง 2 อัลกอริทึมจะมาทำการส่งต่อไปยังส่วนแสดงผลเพื่อให้ผู้ใช้สามารถหาอุปกรณ์จากแผนที่จำลองได้

#### 4.2.6 การออกแบบโปรแกรมในส่วนแสดงผล

ระบบแบ่งโมดูลการแสดงผลออกเป็น 3 โมดูลคือ

##### 4.2.6.1 การแสดงผล Log Messages

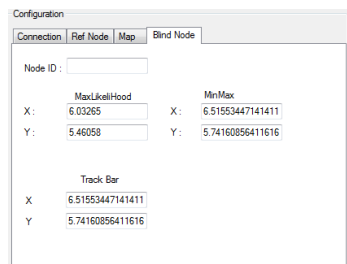
สำหรับแสดงถึงการเชื่อมต่อระหว่าง อุปกรณ์และโปรแกรมสามารถทำได้สมบูรณ์ จะมีค่าสัญญาณ RSSI ที่ดักจับมาได้ระหว่างอุปกรณ์ที่เป็น Blind node และ Reference node สื่อสารกัน



รูปที่ 4.4 แสดงค่าสัญญาณ RSSI และ Node หลังจากเชื่อมต่อเข้ามาในระบบได้

##### 4.2.6.2 การแสดงตำแหน่งวัตถุแบบเป็นพิกัด

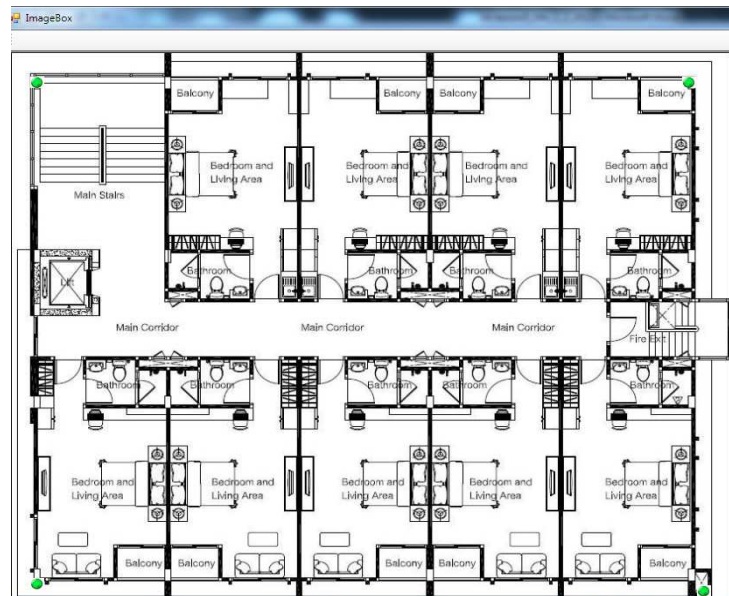
สำหรับแสดงถึงตำแหน่งวัตถุที่บอกพิกัด 2 มิติตามแกนตั้งและแกนนอน



รูปที่ 4.5 แสดงตำแหน่งวัตถุแบบแนวพิกัด X, Y

##### 4.2.6.3 การแสดงผลบนแผนที่จำลอง (Graphic User Interface)

หลังจากระบบสามารถคำนวณหาตำแหน่งวัตถุได้เรียบร้อยแล้ว ก็จะนำพิกัด X,Y มาแสดงผลบนแผนที่

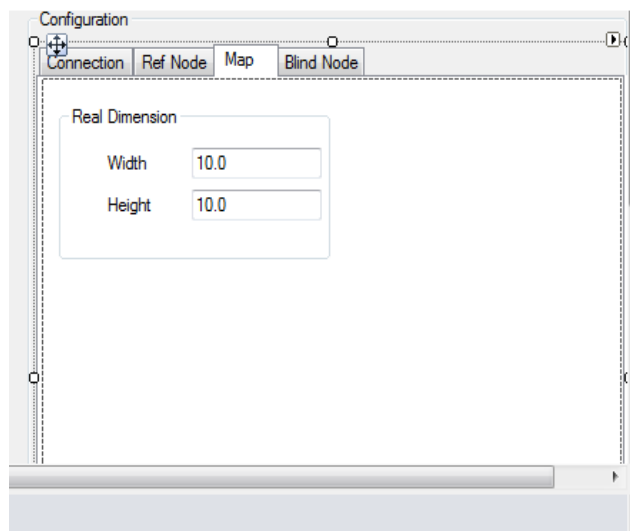


รูปที่ 4.6 แสดงตำแหน่งวัตถุแบบหน้าจอ GUI Map

#### 4.2.7 การปรับแก้ขนาดแผนที่

ผู้ใช้สามารถแก้ขนาดแผนที่ในการ Upload ได้โดยการใส่ค่า

- ความกว้างของรูปภาพ (Width) : pixels
- ความสูงของรูปภาพ (Height) : pixels



รูปที่ 4.7 แสดงส่วนการแก้ไขขนาดของแผนที่



## บทที่ 5

### การทดลอง

ผู้วิจัยได้ทำการทดสอบระบบการค้นหาตำแหน่งวัตถุแบบภายในอาคารและภายนอกอาคาร โดยระบบจะทำงานร่วมกับชุดอุปกรณ์ Zigbee ที่มีอุปกรณ์ที่เป็นโหนดตัวส่งสัญญาณ (Blind node) และโหนดตัวรับสัญญาณ (Reference node) หลักการคำนวณหาตำแหน่งวัตถุจะนำอัลกอริทึม Maximum Likelihood และ Min-Max มาช่วยในการหาค่าตำแหน่ง เนื่องจากสภาพแวดล้อมที่ทำการทดลองมีทั้งแบบภายในอาคารและภายนอกอาคารสถานะของแต่ละแห่งย่อมแตกต่างกันดังนั้นระบบจะต้องใช้ค่า A ซึ่งเป็นตัวแปรชนิดหนึ่ง เป็นตัวแทนของค่าเฉลี่ยความแรงของสัญญาณ (RSSI Average) ของสภาพแวดล้อมในขณะนั้นในระยะ 1 เมตรเท่ากัน[22] โดยค่า A ที่จะถูกนำมาใช้แทนในสมการ (5.1) เพื่อสำหรับใช้หารระยะทางของวัตถุ อีกหนึ่งตัวแปรคือ ตัวแปร N จะใช้แสดงถึงค่าคงที่การกระจายสัญญาณของสถานที่นั้น ซึ่งค่า N จะเป็นตัวแทนของค่าคงที่ของการกระจายสัญญาณแบบภายในอาคารโดยจะนำค่ามาตรฐานของ TI [22] ตั้งแต่ 3.25 – 4.50 และจะเลือกค่า N ที่เหมาะสมเพื่อมาแทนในสมการ (5.1) เพื่อใช้ในการหารระยะให้กับวัตถุ หลังจากคำนวณระยะวัตถุได้เรียบร้อยแล้ว จะนำระยะมาคำนวณหาตำแหน่งด้วยอัลกอริทึม Maximum Likelihood และ Min-Max จากนั้นจะทำการเปรียบเทียบค่าความผิดพลาดของพิกัดระหว่างตำแหน่งจริงกับตำแหน่งที่ได้มาจากระบบการคำนวณ 2 อัลกอริทึม อาทิ Maximum Likelihood , Min Max และ TI เป็นโปรแกรมสำเร็จรูปของบริษัทจัดทำอุปกรณ์ [22] เพื่อใช้ในการประเมินความแม่นยำในพิกัดของตำแหน่งวัตถุ ซึ่งรายละเอียดการทดลองแบ่งตามหัวข้อดังต่อไปนี้

### 5.1 การทดลองการหาค่าตัวแปรสมการเพื่อใช้ในการหาระยะทางของตำแหน่งวัตถุ วัตถุประสงค์ เพื่อหาค่าตัวแปร A และ n ภายในอาคารและภายนอกอาคาร จากสมการ

$$RSSI = -10n \log_{10}(d) + A \quad (5.1)$$

เมื่อ RSSI คือค่าความแรงสัญญาณวิทยุ มีหน่วยเป็น เดซิเบลมิลลิวัตต์ (-dBm)

n คือค่าคงที่การกระจายสัญญาณของสถานที่นั้น ๆ

d คือระยะห่างระหว่างเซนเซอร์ไร้สาย (เมตร)

A คือค่าความแรงสัญญาณที่ได้รับใน 1 เมตร

ผู้วิจัยได้แบ่งการทดลองออกเป็น 2 สภาพแวดล้อมทั้งแบบภายในอาคารและภายนอกอาคาร ดังต่อไปนี้

**สมมติฐาน** ตัวแปร A จะช่วยทำให้ระยะทางที่ได้จากการคำนวณใกล้เคียงหรือเท่ากับ ระยะทางจริง

#### อุปกรณ์ที่ใช้

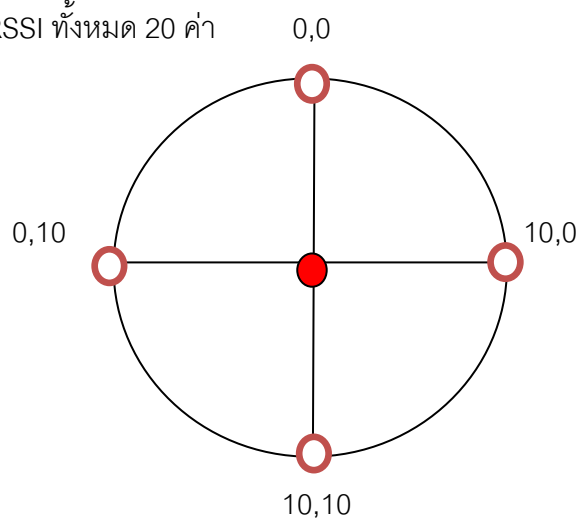
- SmartboardRF04eb 1 ตัว
- Chipcon CC2430 1 ตัว (โหนดตัวส่ง)
- Chipcon CC2431 4 ตัว (โหนดตัวรับ)
- สาย USB 1 เส้น
- สาย Serial Port 1 เส้น
- คอมพิวเตอร์ Notebook 1 ตัว
- กล่องกระดาษ 5 กล่อง
- โปรแกรม Location Engine สำหรับดึงค่าความเข้มของสัญญาณ (RSSI) และ คำนวณหาตำแหน่งวัตถุ

### 5.1.1 หาค่า A แบบภายในอาคาร

วัตถุประสงค์ ทดลองเพื่อหาค่า A สำหรับใช้กับภายในอาคาร

วิธีการทดลอง

- เริ่มจากการวางโหนดตัวส่งไว้ตรงกลางวงกลมให้มีรัศมีระยะห่างเป็น 1 เมตรจากโหนดตัวรับ เก็บค่า RSSI ทั้งหมด 20 ค่า



ภาพที่ 5.1 แสดงการจำลองการจัดวางโหนดตัวรับและตัวส่งเพื่อหาค่าตัวแปร A

- ทำเช่นนี้จนครบตัวรับทั้งหมด 4 ตัวจากนั้นนำค่า RSSI จากโหนดตัวรับทั้ง 4 ตัวมาทำการหาค่าเฉลี่ย
- นำค่าทั้งหมดมาเฉลี่ยรวมจะได้ค่า A สำหรับภายในอาคาร

สภาพแวดล้อม : ใต้อาคารเจริญวิศวกรรม 18 คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย

โหนดที่	ค่าเฉลี่ยของ A ในแต่ละ Node
1	50.0875
2	42.7
3	47.3
4	51.3125
ค่าเฉลี่ยรวม	47.85

ตารางที่ 5.1 ค่าเฉลี่ยรวมของตัวแปร A สำหรับใช้หาวัตถุประสงค์สำหรับภายในอาคาร

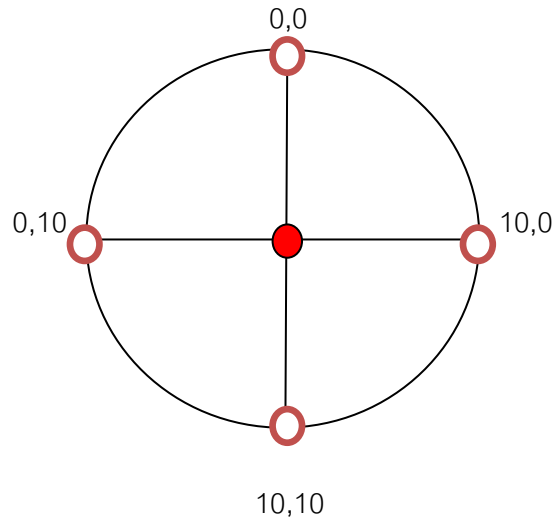
ผลสรุป ค่า A ที่จะถูกนำมาใช้ในการหาระยะทางแบบภายในอาคารเท่ากับ 47.85

### 5.1.2 หาค่า A แบบภายนอกอาคาร

วัตถุประสงค์ ทดลองเพื่อหาค่า A สำหรับใช้กับภายนอกอาคาร

### วิธีการทดลอง

- เริ่มจากการวางโหนดตัวส่งไว้ตรงกลางวงกลมให้มีรัศมีระยะห่างเป็น 1 เมตรจากโหนดตัวรับ เก็บค่า RSSI ทั้งหมด 20 ค่า



ภาพที่ 5.2 แสดงการจำลองการจัดวางโหนดตัวรับและตัวส่งเพื่อหาค่าตัวแปร A

- ทำเช่นนี้จนครบตัวรับทั้งหมด 4 ตัวจากนั้นนำค่า RSSI จากโหนดตัวรับทั้ง 4 ตัวมาทำการหาค่าเฉลี่ย
- นำค่าทั้งหมดมาเฉลี่ยรวมจะได้ค่า A สำหรับภายนอกอาคาร

สภาพแวดล้อม : สนามลานเกียร์ คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย

โหนดที่	ค่าเฉลี่ยของ A ในแต่ละ Node
1	42.91
2	48.01
3	56.53
4	44
ค่าเฉลี่ยรวม	47.865625

ตารางที่ 5.2 ค่าเฉลี่ยรวมของตัวแปร A สำหรับใช้หาวัตถุสำหรับภายนอกอาคาร

**ผลสรุป** ค่า A ที่จะถูกนำมาใช้ในการหาระยะทางแบบภายนอกอาคารเท่ากับ 47.865

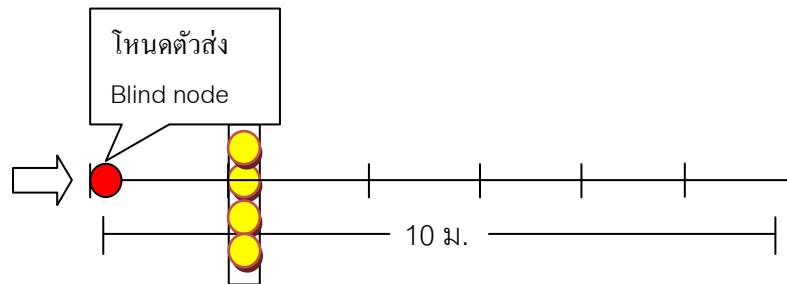
### 5.1.3 หาค่า N แบบภายในอาคาร

วัตถุประสงค์ ทดลองเพื่อหาค่า N ที่ให้ค่า Error น้อยที่สุด

วิธีการทดลอง

- เริ่มจากการวางโหนดตัวรับ(สีเหลือง) ทั้ง 4 ตัวให้อยู่ห่างจากโหนดตัวส่ง (สีแดง) ดังรูป

5.3 เป็นระยะ 1 เมตร



ภาพที่ 5.3 แสดงภาพการจำลองการวางโหนดตัวส่งห่างจากตัวรับในระยะ 1-10 เมตร

- ค่อย ๆ เคลื่อนย้ายระยะเพิ่มไปเรื่อย ๆ ทีละ 1 เมตรจนถึงระยะ 10 เมตรตามลำดับ
- ทำเช่นนี้จนครบค่า n เริ่มต้นที่ 3.25 ,3.375 ,3.5 ,3.625 จนถึง 4.5 จากนั้นนำค่า RSSI ที่วัดได้จากโหนดแต่ละตัวหาค่าเฉลี่ยจากแต่ละ n ตามลำดับ
- หาค่า Error จาก n ทุกตัว สรุปได้ค่า N ของ Indoor ที่ให้ค่า Error น้อยสุดคือ 3.25 ดังตาราง 5.3

Distance /Node	N1	N2	N3	N4	N5	N6	N7	N8	N9	N10	N11
	3.25	3.375	3.5	3.625	3.750	3.875	4.000	4.125	4.250	4.375	4.500
1,1	99.49	94.43	89.84	85.66	81.85	78.36	75.14	72.18	69.44	66.89	64.52
1,2	121.71	115.27	109.45	104.18	99.38	94.99	90.96	87.25	83.83	80.66	77.71
1,3	-99.92	-99.90	-99.87	-99.83	-99.80	-99.75	-99.70	-99.64	-99.58	-99.50	-99.43
1,4	-99.96	-99.95	-99.93	-99.91	-99.89	-99.86	-99.83	-99.79	-99.75	-99.71	-99.65
2,1	45.60	39.95	34.90	30.36	26.26	22.55	19.16	16.06	13.22	10.61	8.20
2,2	-6.78	-8.90	-10.84	-12.60	-14.21	-15.70	-17.06	-18.32	-19.50	-20.58	-21.60
2,3	-99.97	-99.97	-99.96	-99.94	-99.93	-99.91	-99.89	-99.87	-99.85	-99.82	-99.79



Distance /Node	N1	N2	N3	N4	N5	N6	N7	N8	N9	N10	N11
	3.25	3.375	3.5	3.625	3.750	3.875	4.000	4.125	4.250	4.375	4.500
10,1	4.60	-4.11	-11.55	-17.95	-23.51	-28.37	-32.64	-36.43	-39.79	-42.80	-45.51
10,2	4.60	-4.11	-11.55	-17.95	-23.51	-28.37	-32.64	-36.43	-39.79	-42.80	-45.51
10,3	-100.00	-99.99	-100.00	-100.00	-100.00	-100.00	-99.99	-99.99	-99.99	-99.99	-99.99
10,4	-100.00	-100.00	-100.00	-100.00	-100.00	-100.00	-99.99	-99.99	-99.99	-99.99	-99.99
ค่าเฉลี่ย	-37.02	-40.80	-44.07	-46.93	-49.44	-51.67	-53.65	-55.42	-57.02	-58.46	-59.77

ตารางที่ 5.3 หาค่า N ในระยะทาง 10 เมตรแบบภายในอาคาร

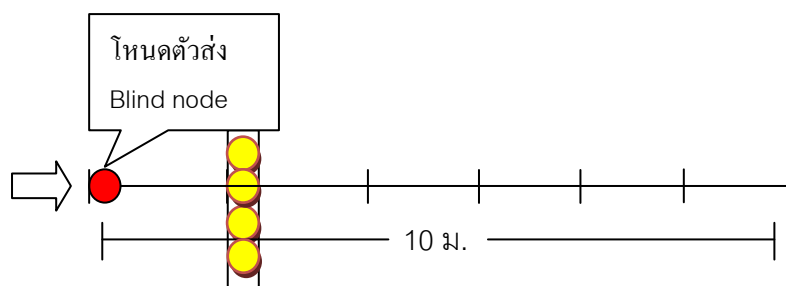
**ผลสรุป** ค่า N ที่มีค่าความผิดพลาดน้อยที่สุดคือ 3.25 จะถูกนำไปใช้ในการคำนวณหาตำแหน่งวัตถุแบบภายในอาคาร

#### 5.1.4 หาค่า N แบบภายนอกอาคาร

**วัตถุประสงค์** ทดลองเพื่อหาค่า N ที่ให้ค่า Error น้อยที่สุด

**วิธีการทดลอง**

- เริ่มจากการวางโหนดตัวรับ (สีเหลือง) ทั้ง 4 ตัวให้อยู่ห่างจากโหนดตัวส่ง (สีแดง) ดังรูป 5.4 เป็นระยะ 1 เมตร และค่อย ๆ เคลื่อนย้ายระยะเพิ่มไปเรื่อย ๆ ทีละ 1 เมตรจนถึงระยะ 10 เมตรตามลำดับ



ภาพที่ 5.4 แสดงภาพการจำลองการวางโหนดตัวส่งห่างจากตัวรับในระยะ 1-10 เมตร

- ทำเช่นนี้จนครบค่า n เริ่มต้นที่ 3.25, 3.375, 3.5, 3.625 จนถึง 4.5 จากนั้นนำค่า RSSI ที่วัดได้จากโหนดแต่ละตัวหาค่าเฉลี่ยจากแต่ละ n ตามลำดับ
- หาค่า Error จาก n ทุกตัว สรุปได้ค่า N ของ Indoor ที่ให้ค่า Error น้อยที่สุดคือ 3.25 ดังตาราง 5.4

Distance (Metre)/Node	N1	N2	N3	N4	N5	N6	N7	N8	N9	N10	N11
	3.25	3.375	3.5	3.625	3.750	3.875	4.000	4.125	4.250	4.375	4.500
1,1	99.49	94.43	89.84	85.66	81.85	78.36	75.14	72.18	69.44	66.89	64.52
1,2	121.71	115.27	109.45	104.18	99.38	94.99	90.96	87.25	83.83	80.66	77.71
1,3	-99.92	-99.90	-99.87	-99.83	-99.80	-99.75	-99.70	-99.64	-99.58	-99.50	-99.43
1,4	-99.96	-99.95	-99.93	-99.91	-99.89	-99.86	-99.83	-99.79	-99.75	-99.71	-99.65
2,1	39.95	45.60	34.90	30.36	26.26	22.55	19.16	16.06	13.22	10.61	8.20
2,2	-8.90	-6.78	-10.84	-12.60	-14.21	-15.70	-17.06	-18.32	-19.50	-20.58	-21.60
2,3	-99.97	-99.97	-99.96	-99.94	-99.93	-99.91	-99.89	-99.87	-99.85	-99.82	-99.79
2,4	-99.98	-99.99	-99.98	-99.97	-99.96	-99.95	-99.94	-99.93	-99.91	-99.89	-99.87
3,1	29.31	22.98	17.38	12.39	7.93	3.92	0.29	-3.00	-6.00	-8.75	-11.26
3,2	-7.24	-10.69	-13.78	-16.56	-19.08	-21.36	-23.44	-25.35	-27.10	-28.71	-30.20
3,3	-99.98	-99.98	-99.97	-99.97	-99.96	-99.95	-99.93	-99.92	-99.91	-99.89	-99.87
3,4	-99.99	-99.99	-99.99	-99.99	-99.98	-99.98	-99.97	-99.96	-99.96	-99.95	-99.94
4,1	18.34	11.72	5.90	0.75	-3.82	-7.91	-11.59	-14.91	-17.92	-20.67	-23.17
4,2	-27.45	-30.26	-32.77	-35.02	-37.06	-38.90	-40.59	-42.13	-43.54	-44.84	-46.04
4,3	-99.99	-99.99	-99.98	-99.98	-99.98	-99.97	-99.96	-99.95	-99.94	-99.93	-99.92
4,4	-100.00	-100.00	-99.99	-99.99	-99.99	-99.99	-99.98	-99.98	-99.97	-99.97	-99.96
5,1	40.85	31.02	22.51	15.08	8.56	2.79	-2.34	-6.92	-11.04	-14.75	-18.12
5,2	-26.66	-30.10	-33.16	-35.88	-38.33	-40.53	-42.52	-44.33	-45.98	-47.49	-48.88
5,3	-99.99	-99.99	-99.99	-99.99	-99.98	-99.98	-99.98	-99.97	-99.96	-99.96	-99.95
5,4	-100.00	-100.00	-100.00	-99.99	-99.99	-99.99	-99.99	-99.98	-99.98	-99.98	-99.97
6,1	74.34	59.82	47.42	36.75	27.48	19.38	12.26	5.96	0.35	-4.67	-9.18
6,2	74.34	59.82	47.42	36.75	27.48	19.38	12.26	5.96	0.35	-4.67	-9.18
6,3	-100.00	-100.00	-99.99	-99.99	-99.99	-99.99	-99.98	-99.98	-99.98	-99.97	-99.97
6,4	-100.00	-100.00	-100.00	-100.00	-99.99	-99.99	-99.99	-99.99	-99.98	-99.98	-99.98
7,1	12.55	4.27	-2.88	-9.09	-14.53	-19.32	-23.57	-27.36	-30.75	-33.80	-36.56
7,2	-15.51	-20.89	-25.59	-29.70	-33.34	-36.57	-39.46	-42.05	-44.39	-46.51	-48.43
7,3	-100.00	-100.00	-99.99	-99.99	-99.99	-99.99	-99.98	-99.98	-99.98	-99.97	-99.96
7,4	-100.00	-100.00	-100.00	-100.00	-99.99	-99.99	-99.99	-99.99	-99.99	-99.98	-99.98
8,1	19.86	30.75	10.57	2.56	-4.39	-10.46	-15.81	-20.53	-24.74	-28.50	-31.88



Distance (Metre)/Node	N1	N2	N3	N4	N5	N6	N7	N8	N9	N10	N11
	3.25	3.375	3.5	3.625	3.750	3.875	4.000	4.125	4.250	4.375	4.500
8,2	4.58	13.48	-3.06	-9.68	-15.44	-20.50	-24.96	-28.93	-32.47	-35.65	-38.51
8,3	-100.00	-100.00	-100.00	-100.00	-99.99	-99.99	-99.99	-99.99	-99.99	-99.98	-99.98
8,4	-100.00	-100.00	-100.00	-100.00	-100.00	-99.99	-99.99	-99.99	-99.99	-99.99	-99.98
9,1	16.22	6.55	-1.72	-8.84	-15.01	-20.41	-25.16	-29.36	-33.10	-36.45	-39.45
9,2	16.22	6.55	-1.72	-8.84	-15.01	-20.41	-25.16	-29.36	-33.10	-36.45	-39.45
9,3	-100.00	-100.00	-100.00	-100.00	-100.00	-99.99	-99.99	-99.99	-99.99	-99.99	-99.98
9,4	-100.00	-100.00	-100.00	-100.00	-100.00	-99.99	-99.99	-99.99	-99.99	-99.99	-99.98
10,1	4.60	-4.11	-11.55	-17.95	-23.51	-28.37	-32.64	-36.43	-39.79	-42.80	-45.51
10,2	4.60	-4.11	-11.55	-17.95	-23.51	-28.37	-32.64	-36.43	-39.79	-42.80	-45.51
10,3	-100.00	-99.99	-100.00	-100.00	-100.00	-100.00	-99.99	-99.99	-99.99	-99.99	-99.99
10,4	-100.00	-100.00	-100.00	-100.00	-100.00	-100.00	-99.99	-99.99	-99.99	-99.99	-99.99
ค่าเฉลี่ย	-37.71	-40.11	-44.07	-46.93	-49.44	-51.67	-53.65	-55.42	-57.02	-58.46	-59.77

ตารางที่ 5.4 หาค่า N ในระยะทาง 10 เมตรแบบภายนอกอาคาร

**ผลสรุป** ค่า N ที่มีค่าความผิดพลาดน้อยที่สุดคือ 3.25 จะถูกนำไปใช้ในการคำนวณหาตำแหน่งวัตถุแบบภายนอกอาคาร

## 5.2 การทดลองหาตำแหน่งของวัตถุด้วยค่าความเข้มสัญญาณ (Received Signal Strength Indicator)

**วัตถุประสงค์** : เพื่อหาค่าความแม่นยำของการระบุตำแหน่งแบบภายนอกอาคาร

**สมมติฐาน** : การหาตำแหน่งมีความแม่นยำในระดับที่ใช้งานได้จริง

### อุปกรณ์

(เหมือนการทดลองที่ 5.1)

### วิธีการทดลอง

- จัดเตรียมพื้นที่การทดลองที่มีขนาดเท่ากับตารางกริด 10 x 10 เมตรดังภาพที่ 5.5 โดยโหนดตัวรับถูกติดตั้งแต่ละมุมของพื้นที่แสดงดังภาพที่ 5.6

- นำโหนดตัวส่งไปไว้ตามจุดต่าง ๆ ทั้งสิ้น 14 จุดดังนี้ (2,5) , (5,2) , (7,3) , (5,5) , (8,4) , (7,5) , (2,8) , (7,8) , (8,7) , (2,2),(7,1) , (10,5) , (0,5) และ (8,3) เมตร เพื่อให้โหนดตัวรับและตัวส่งทำการส่งคำสั่งสัญญาณเข้าโปรแกรมจากนั้นคำนวณผลและทำการบันทึกผล
- ทำการวัดผลตำแหน่งด้วยโปรแกรมของ TI ตามจุดต่าง ๆ ตามด้านบนและบันทึกผล
- ทำจนวนครบทุกตำแหน่งโดยเลือกมาเฉพาะ 20 ค่าแรกและทำการหาค่าเฉลี่ยรวมในทุกตำแหน่ง
- เปรียบเทียบประสิทธิภาพด้วยสมการวัดความผิดพลาด (Euclidian Distance Error)[7] ระหว่างอัลกอริทึม Maximum Likelihood , Min-Max และ TI

0,0	1,0	2,0	3,0	4,0	5,0	6,0	7,0	8,0	9,0	10,0
0,1	1,1	2,1	3,1	4,1	5,1	6,1	7,1	8,1	9,1	10,1
0,2	1,2	2,2	3,2	4,2	5,2	6,2	7,2	8,2	9,2	10,2
0,3	1,3	2,3	3,3	4,3	5,3	6,3	7,3	8,3	9,3	10,3
0,4	1,4	2,4	3,4	4,4	5,4	6,4	7,4	8,4	9,4	10,4
0,5	1,5	2,5	3,5	4,5	5,5	6,5	7,5	8,5	9,5	10,5
0,6	1,6	2,6	3,6	4,6	5,6	6,6	7,6	8,6	9,6	10,6
0,7	1,7	2,7	3,7	4,7	5,7	6,7	7,7	8,7	9,7	10,7
0,8	1,8	2,8	3,8	4,8	5,8	6,8	7,8	8,8	9,8	10,8
0,9	1,9	2,9	3,9	4,9	5,9	6,9	7,9	8,9	9,9	10,9
0,10	1,10	2,10	3,10	4,10	5,10	6,10	7,10	8,10	9,10	10,10

ภาพที่ 5.5 ภาพจำลองการจัดพื้นที่แบบตารางกริด

#### สภาพแวดล้อม

- ภายนอกอาคาร ลานเกียร์ คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย



ภาพที่ 5.6 แสดงการทดลองหาตำแหน่งของวัตถุแบบภายนอกอาคาร

#### ผลการทดลอง

ผู้วิจัยทำการจำแนกผลออกเป็น 2 กลุ่มหลัก ดังนี้

##### 5.2.1 กลุ่มแรกคือ ชนิดที่ระยะห่างระหว่างตัวส่งและตัวรับถูกกำหนดให้อยู่ในระยะต่ำกว่า 10 เมตร ( Valid Length )

การทดลองทำการจัดกลุ่มตำแหน่งของโหนดตัวรับและตัวส่งของทั้ง 4 ตัวรับและ 1 ตัวส่งให้อยู่ในระยะต่ำกว่า 10 เมตร รวมทั้งสิ้น 6 จุด ประกอบด้วย (7,3) , (5,5) , (7,5) , (2,5) , (5,2) และ (8,4)

Algorithm		Maximum Likelihood		Min-Max		TI	
จุดที่	ตำแหน่ง	ค่าเฉลี่ย	ค่าเบี่ยงเบนฯ	ค่าเฉลี่ย	ค่าเบี่ยงเบนฯ	ค่าเฉลี่ย	ค่าเบี่ยงเบนฯ
1	7,3	1.01	0.04	2.35	0.04	3.02	0.00
2	5,5	0.37	0.03	1.21	0.06	2.47	0.00
3	7,5	1.06	0.07	0.94	0.07	1.49	0.03
4	2,5	-	-	2.07	0.34	1.26	0.83
5	5,2	-	-	0.84	0.15	2.41	0.26
6	8,4	-	-	-	-	-	-
	<b>รวม</b>	0.81	0.02	1.48	0.12	2.13	0.36

ตารางที่ 5.5 ผลการทดลองหาตำแหน่งวัตถุแบบภายนอกอาคารในระยะต่ำกว่า 10 เมตร

### สรุปผลการทดลอง

จากการทดลองพบว่าการค้นหาตำแหน่งวัตถุแบบภายนอกอาคาร อัลกอริทึม Maximum Likelihood ให้ค่าความแม่นยำในการหาตำแหน่งวัตถุแบบภายนอกอาคารดีที่สุด ซึ่งค่าความผิดพลาดเฉลี่ยเท่ากับ 0.81 เมตร รองมาคือ Min-Max เท่ากับ 1.48 เมตร และอัลกอริทึม TI เท่ากับ 2.13 เมตร

อัลกอริทึม Maximum Likelihood มีค่าเบี่ยงเบนมาตรฐานเฉลี่ย เท่ากับ 0.02 เมตร รองมาคืออัลกอริทึม Min-Max เท่ากับ 0.12 เมตรและอัลกอริทึม TI เท่ากับ 0.36 เมตร สรุปได้ว่าอัลกอริทึม Maximum Likelihood ตอบตำแหน่งได้คงที่มากที่สุดเมื่อเปรียบเทียบกับอัลกอริทึม Min-Max และ TI

**หมายเหตุ :** อัลกอริทึมที่ใช้ในการหาตำแหน่งคือ อัลกอริทึม Maximum Likelihood ซึ่งอัลกอริทึมนี้จะตอบตำแหน่งได้ต้องได้รับการตอบรับสัญญาณจาก Reference node อย่างน้อย 4 โหนด[24][25] ส่วนอีกอัลกอริทึมที่ใช้ในการหาตำแหน่งคืออัลกอริทึม Min-Max ซึ่งอัลกอริทึมนี้จะตอบตำแหน่งได้จะต้องได้รับการตอบกลับของสัญญาณอย่างน้อย 3 โหนด [26]

ด้วยเหตุนี้ จากตาราง 5.5 กรณีผลตำแหน่งแกน X, Y ตอบมาเป็น ( - ) หมายถึงอัลกอริทึมดังกล่าวได้รับข้อมูล Reference node ไม่ครบและไม่เป็นไปตามข้อกำหนดของกฎแต่จะอัลกอริทึมจึงทำให้คำตอบพิกัดตำแหน่งจึงเป็น Invalid

ทั้งนี้ทั้งนั้น เมื่อระยะไกลออกไปจนเกินอาณาเขตที่สัญญาณจะรับส่งกัน ทำให้โหนดตัวรับไม่สามารถรับสัญญาณได้ โหนดตัวส่งจะตั้งค่าสัญญาณ Default มาใช้แทนคือ -81 dBm

และเมื่อนำมาเข้าสมการการแปลงสัญญาณเป็นระยะทาง จะได้ค่าระยะทาง Default เท่ากับ 10.46 เมตรซึ่งมีความผิดพลาดไปจากตำแหน่งจริงกับค่าที่ควรจะได้รับ

## 5.2.2 กลุ่มสองคือ ชนิดที่ระยะห่างระหว่างตำแหน่งตัวส่งและตัวรับถูกกำหนดให้อยู่ใน ระยะเกิน 10 เมตร ( Invalid Length )

การทดลองจัดกลุ่มตำแหน่งของโหนดตัวรับและตัวส่งของทั้ง 4 ตัวรับและ 1 ตัวส่งให้อยู่ใน ระยะเกินกว่า 10 เมตร โดยตำแหน่งที่ติดตั้งรวมทั้งสิ้น 8 จุด ประกอบด้วย (2,8) , (7,8) , (8,7) , (2,2) , (7,1) , (10,5) , (0,5) และ (8,3)

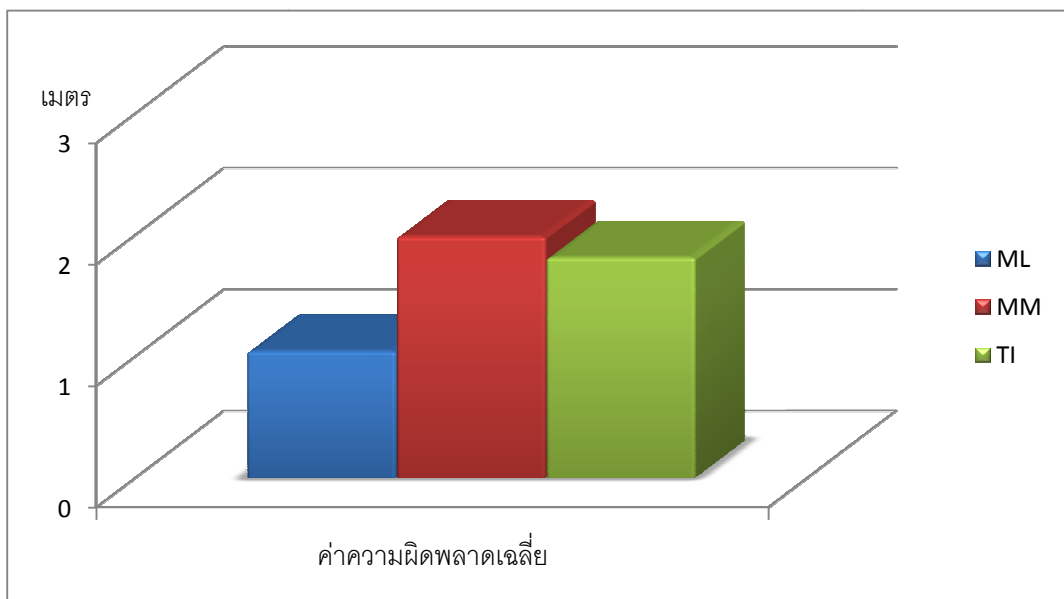
Algorithm		Maximum Likelihood		Min-Max		TI	
จุดที่	ตำแหน่ง	ค่าเฉลี่ย	ค่าเบี่ยงเบนฯ	ค่าเฉลี่ย	ค่าเบี่ยงเบนฯ	ค่าเฉลี่ย	ค่าเบี่ยงเบนฯ
1	2,8	-	-	1.46	0.35	1.96	1.27
2	7,8	-	-	3.57	0.06	2.39	0.13
3	8,7	1.86	0.05	3.73	0.03	2.12	0.00
4	2,2	-	-	1.35	0.38	3.84	0.24
5	7,1	-	-	0.56	0.07	0.79	0.17
6	10,5	4.38	0.17	4.79	0.03	8.05	0.00
7	0,5	-	-	4.53	0.03	3.40	0.00
8	8,3	3.37	0.27	1.96	0.01	1.77	0.00
	รวม	3.20	0.11	2.74	0.15	3.04	0.43

ตาราง 5.6 ผลการทดลองหาตำแหน่งวัตถุแบบภายนอกอาคารในระยะเกิน 10 เมตร

### สรุปผลการทดลอง

จากการทดลองพบว่าอัลกอริทึม Min-Max ให้ประสิทธิภาพในการหาตำแหน่งวัตถุแบบ ภายนอกอาคารได้มีความแม่นยำมากที่สุด โดยค่าความผิดพลาดเฉลี่ย เท่ากับ 2.74 เมตร รองมา คือ TI เท่ากับ 3.04 เมตร และอัลกอริทึม Maximum Likelihood เท่ากับ 3.20 เมตร

อัลกอริทึม Maximum Likelihood มีค่าส่วนเบี่ยงเบนมาตรฐานเฉลี่ยน้อยที่สุด เท่ากับ 0.11 เมตร รองมาคืออัลกอริทึม Min-Max เท่ากับ 0.15 เมตรและอัลกอริทึม TI เท่ากับ 0.43 เมตร



ภาพที่ 5.7 แสดงผลการเปรียบเทียบค่าความผิดพลาดเฉลี่ยของทั้ง 3 อัลกอริทึมคือ Maximum Likelihood , Min-Max และ TI แบบภายนอกอาคาร

### 5.3 การทดลองหาตำแหน่งของวัตถุด้วยค่าความเข้มสัญญาณ (Received Signal Strength Indicator)

**วัตถุประสงค์ :** เพื่อหาค่าความแม่นยำของการระบุตำแหน่งแบบภายในอาคาร

**สมมติฐาน :** การหาตำแหน่งมีความแม่นยำในระดับที่ใช้งานได้จริง

**อุปกรณ์**

(เหมือนการทดลองที่ 5.1)

**วิธีการทดลอง**

- จัดเตรียมพื้นที่การทดลองที่มีขนาด 10 x 10 เมตรโดยโหนดตัวรับถูกติดตั้งแต่ละมุมของพื้นที่แสดงดังภาพที่ 5.8
- นำโหนดตัวส่งไปไว้ตามจุดต่าง ๆ ทั้งสิ้น 29 จุดดังนี้ (2,5) , (5,2) , (7,3) , (5,5) , (8,4) , (7,5) , (5,3) , (3,3) , (2,4) , (4,4) , (6,4) , (3,5) , (2,6) , (4,6) , (6,6) , (8,6) , (7,7) , (5,7) , (3,7) , (5,8) , (8,5) , (2,2) , (2,8) , (8,2) , (8,8) ,(1,5) , (5,1) ,(5,9) และ (9,5) เมตร เพื่อให้โหนดตัวรับและตัวส่งทำการส่งค่าสัญญาณเข้าโปรแกรมจากนั้นคำนวณผลและทำการบันทึกผล

- ทำการวัดผลตำแหน่งด้วยโปรแกรมของ TI ตามจุดต่าง ๆ ด้านบน
- ทำเช่นนี้จนครบ 29 จุดในการทดลองทำทั้งสิ้นจำนวน 4 ครั้ง เก็บค่าเฉลี่ย 10 ค่าแรกโดยจับเวลาตำแหน่งละ 10 นาที จากนั้นนำมาทำการหาค่าเฉลี่ยรวมทั้งหมดในทุกตำแหน่ง
- เปรียบเทียบความแม่นยำการหาตำแหน่งของ Location Engine ระหว่างอัลกอริทึม Maximum Likelihood , Min-Max และ TI

### สภาพแวดล้อม

- ภายในอาคาร สำนักงานฝ่ายเทคโนโลยีสารสนเทศ ชั้น 16 ธนาคารเพื่อการส่งออกและนำเข้าแห่งประเทศไทย (EXIM BANK)



ภาพที่ 5.8 แสดงการทดลองหาตำแหน่งของวัตถุแบบภายในอาคาร

### ผลการทดลอง

ผู้วิจัยทำการจำแนกผลออกเป็น 3 กลุ่มหลัก ดังนี้

#### 5.3.1 กลุ่มแรกคือ ชนิดที่ระยะห่างระหว่างตัวส่งและตัวรับถูกกำหนดให้อยู่ในระยะต่ำกว่า 10 เมตร ( Valid Length )

การทดลองจึงมีการจัดกลุ่มตำแหน่งของโหนดตัวรับและตัวส่งของทั้ง 4 ตัวรับและ 1 ตัวส่งให้อยู่ในระยะต่ำกว่า 10 เมตร รวมทั้งสิ้น 21 จุด ประกอบด้วย (2,5) , (5,2) , (7,3) , (5,5) , (8,4) , (7,5) , (5,3) , (3,3) , (2,4) , (4,4) , (6,4) , (3,5) , (2,6) , (4,6) , (6,6) , (8,6) , (7,7) , (5,7) , (3,7) , (5,8) และ (8,5)

Algorithm		Maximum Likelihood		Min-Max		TI	
จุดที่	ตำแหน่ง	ค่าเฉลี่ย	ค่าเบี่ยงเบน	ค่าเฉลี่ย	ค่าเบี่ยงเบน	ค่าเฉลี่ย	ค่าเบี่ยงเบน
1	2,5	1.17	0.27	1.19	0.45	1.17	0.14
2	5,2	1.97	0.00	1.68	0.02	2.19	0.98
3	7,3	3.10	0.01	3.11	0.01	2.71	0.40
4	5,5	2.19	0.21	1.55	0.20	1.88	0.14
5	8,4	2.93	0.07	2.46	0.07	3.02	0.27
6	7,5	1.00	0.32	1.15	0.14	3.19	1.38
7	5,3	0.52	0.09	1.57	0.01	2.39	0.56
8	3,3	-	-	0.61	0.58	2.14	3.28
9	2,4	1.38	0.04	1.18	0.40	1.04	0.52
10	4,4	0.99	0.10	0.97	0.19	3.41	2.09
11	6,4	1.50	0.16	0.72	0.13	2.11	0.87
12	3,5	0.72	0.17	0.85	0.06	2.07	1.17
13	2,6	-	-	1.49	0.15	2.42	1.02
14	4,6	1.03	0.03	0.32	0.09	1.87	0.80
15	6,6	1.59	0.59	1.09	0.08	1.87	0.80
16	8,6	-	-	2.15	0.74	5.25	3.52
17	7,7	1.43	0.35	2.74	0.03	3.78	0.65
18	5,7	1.36	0.33	1.38	0.10	3.16	1.96
19	3,7	2.09	0.23	2.47	0.04	3.65	1.91
20	5,8	2.30	0.65	2.29	0.45	4.50	1.70
21	8,5	1.56	0.08	1.11	0.53	3.78	2.29
รวม		1.60	0.19	1.53	0.22	2.74	0.96

ตารางที่ 5.7 ผลการทดลองการหาตำแหน่งวัตถุภายในอาคารในระยะต่ำกว่า 10 เมตร



### สรุปผลการทดลอง

จากการทดลองพบว่าอัลกอริทึม Min-Max มีความแม่นยำในการหาตำแหน่งวัตถุแบบภายในอาคารดีที่สุด โดยค่าความผิดพลาดเฉลี่ยเท่ากับ 1.53 เมตร รองลงมาคือ อัลกอริทึม Maximum Likelihood เท่ากับ 1.60 เมตรและ อัลกอริทึม TI เท่ากับ 2.74 เมตร

อัลกอริทึม Maximum Likelihood มีค่าส่วนเบี่ยงเบนมาตรฐานต่ำสุดเท่ากับ 0.19 รองมาคือ Min-Max เท่ากับ 0.22 และ TI เท่ากับ 0.96 พบว่า อัลกอริทึม Maximum Likelihood ตอบค่าตำแหน่งพิกัดได้มีค่าคงที่มากที่สุดเมื่อเปรียบเทียบกับอัลกอริทึม Min-Max และ TI

**หมายเหตุ :** จากตาราง 5.7 สัญลักษณ์ (-) ได้กล่าวไว้แล้วในการทดลองที่ 5.2 ด้วยเหตุนี้ จากตาราง 5.7 กรณีผลตำแหน่งแกน X, Y ตอบมาเป็น (-) หมายถึงอัลกอริทึมดังกล่าวได้รับข้อมูล Reference node ไม่ครบ ค่าตอบพิกัดตำแหน่งจึงเป็น Invalid แต่เนื่องด้วยจุดตำแหน่ง 3,3 และ 2,6 เมื่อตำแหน่งจัดอยู่ในระยะ 10 เมตรจริงแต่ได้ชนิดเป็น Invalid เนื่องจากจุดดังกล่าวติดตั้งอยู่ที่บริเวณอับสัญญาณจึงทำให้ไม่สามารถสื่อสารกลับมายังโหนดตัวส่ง (Blind node) ได้

#### 5.3.2 กลุ่มสองคือ ชนิดที่ระยะห่างระหว่างตัวส่งห่างจากตัวรับเพียง 1 ตัวที่ระยะเกิน 10 เมตร ( Invalid Length )

การทดลองจึงมีการจัดกลุ่มตำแหน่งของโหนดตัวรับและตัวส่งของทั้ง 4 ตัวรับและ 1 ตัวส่งให้อยู่เกิน 10 เมตรทั้งสิ้น 4 จุด ประกอบด้วย (2,2) , (2,8) , (8,2) และ (8,8)

Algorithm		Maximum Likelihood		Min-Max		TI	
ลำดับที่	ตำแหน่ง	ค่าเฉลี่ย	ค่าเบี่ยงเบน	ค่าเฉลี่ย	ค่าเบี่ยงเบน	ค่าเฉลี่ย	ค่าเบี่ยงเบน
1	2,2	-	-	2.95	0.02	2.20	0.62
2	2,8	-	-	1.33	0.09	0.86	0.17
3	8,2	2.95	0.41	1.62	1.08	3.08	2.37
4	8,8	4.20	0.12	2.73	1.45	4.60	3.01
	รวม	3.58	0.20	2.16	0.72	2.68	1.36

ตารางที่ 5.8 ผลการทดลองการหาตำแหน่งวัตถุภายในอาคารระยะเกิน 10 เมตร

(Invalid Length = 1 Reference node)

### สรุปผลการทดลอง

จากการทดลองพบว่าอัลกอริทึม Min-Max มีความแม่นยำในการหาตำแหน่งวัตถุ อยู่ภายในอาคารดีที่สุด ค่าความผิดพลาดเฉลี่ย เท่ากับ 2.16 เมตร รองมาคืออัลกอริทึม TI เท่ากับ 2.68 เมตรและ อัลกอริทึม Maximum Likelihood เท่ากับ 3.58 เมตร อัลกอริทึม Maximum Likelihood ตอบพิกัดตำแหน่งได้คงที่มากที่สุด ค่าเบี่ยงเบนมาตรฐานเท่ากับ 0.20 เมตรรองมาคือ Min-Max เท่ากับ 0.72 เมตรและ TI เท่ากับ 1.36 เมตร

**หมายเหตุ :** จากตาราง 5.8 สัญลักษณ์ (-) ได้กล่าวไว้แล้วในการทดลองที่ 5.2 ด้วยเหตุนี้ จาก ตาราง 5.8 กรณีผลตำแหน่งแกน X, Y ตอบมาเป็น (-) หมายถึงอัลกอริทึมดังกล่าวได้รับข้อมูล Reference node ไม่ครบ คำตอบพิกัดตำแหน่งจึงเป็น Invalid แต่มีบางส่วนที่เป็น Valid เนื่องจากจากชุดคำตอบนี้เกิดจากการนำผลลัพธ์ของตำแหน่งมาเฉลี่ยกัน 4 ครั้งรวมทั้งหมดตำแหน่ง ละ 10 ค่า บางครั้งทำการทดลองสัญญาณสามารถรับได้อาจจะเกิดจากการสะท้อนหรือหักเห ของสัญญาณ

#### 5.3.3 กลุ่มสามคือ ชนิดที่ระยะห่างระหว่างตัวส่งห่างจากตัวรับ 2 ตัวที่ระยะเกิน 10

##### เมตร ( Invalid Length )

การทดลองนี้จัดกลุ่มตำแหน่งของโหนดตัวรับและตัวส่งของทั้ง 4 ตัวรับและ 1 ตัวส่ง โดยให้ โหนดตัวรับ 2 ตัวอยู่เกินระยะ 10 เมตร เพื่อที่จะทดสอบและหาค่าตำแหน่งวัตถุพร้อมทั้ง เปรียบเทียบข้อผิดพลาดเฉลี่ยทั้งสิ้น 4 จุด ประกอบด้วย (1,5) , (5,9) , (9,5) และ (5,1)

Algorithm		Maximum Likelihood		Min-Max		TI	
ลำดับที่	ตำแหน่ง	ค่าเฉลี่ย	ค่าเบี่ยงเบน	ค่าเฉลี่ย	ค่าเบี่ยงเบน	ค่าเฉลี่ย	ค่าเบี่ยงเบน
1	1,5	3.23	1.92	3.28	1.63	2.47	1.16
2	5,9	4.19	0.04	3.97	1.30	5.30	2.10
3	9,5	5.63	4.02	5.99	3.68	6.22	3.53
4	5,1	1.88	0.14	1.73	0.04	1.56	1.39
	รวม	3.73	1.87	3.74	1.51	3.89	1.07

ตารางที่ 5.9 ผลการทดลองการหาตำแหน่งวัตถุภายในอาคารระยะเกิน 10 เมตร

(Invalid Length = 2 Reference node)

### สรุปผลการทดลอง

จากการทดลองพบว่า อัลกอริทึม Maximum Likelihood มีความแม่นยำมากที่สุดในการหาตำแหน่งวัตถุภายในอาคารในระยะเกิน 10 เมตร ค่าความผิดพลาดเฉลี่ย เท่ากับ 3.73 เมตร รองมาคืออัลกอริทึม Min-Max เท่ากับ 3.74 เมตรและ อัลกอริทึม TI เท่ากับ 3.89 เมตร อัลกอริทึม TI ตอบค่าพิกัดตำแหน่งคงที่มากที่สุด ค่าส่วนเบี่ยงเบนมาตรฐานเท่ากับ 1.07 เมตร รองมาคือ Min-Max เท่ากับ 1.51 เมตรและ Maximum Likelihood เท่ากับ 3.73 เมตร

**หมายเหตุ :** จากตาราง 5.9 สัญลักษณ์ ( - ) ได้กล่าวไว้แล้วในการทดลองที่ 5.2 ด้วยเหตุนี้ จากตาราง 5.9 กรณีผลตำแหน่งแกน X, Y ควรจะตอบมาเป็นชนิดของ Invalid แต่กลับเป็นชนิดตรงข้าม เนื่องจากชุดการทดลองนี้ถูกการสะท้อนของสัญญาณกลับในช่วงเวลาทดลองในขณะนั้นผลที่ได้จึงก่อให้เกิดการรับค่าสัญญาณและได้ผลเป็นตำแหน่ง X และ Y ที่มีค่าผิดพลาดค่อนข้างสูง

### สรุปภาพรวม

การหาตำแหน่งวัตถุแบบภายในอาคาร อัลกอริทึม Min-Max มีประสิทธิภาพมากที่สุด และ อัลกอริทึม Maximum Likelihood จะตอบพิกัดตำแหน่งได้คงที่มากที่สุด

### 5.4 ความแม่นยำของการวัดระยะโดยใช้ค่าความแรงสัญญาณ (Received Signal Strength Indicator : RSSI )

**วัตถุประสงค์ :** เพื่อดูแนวโน้มการเปลี่ยนแปลงของค่า RSSI ในทุกระยะ ๆ ที่เพิ่มขึ้นทีละ 1 เมตร จนถึง 10 เมตร แบบภายในอาคาร

**สมมติฐาน :** ผลการเปลี่ยนแปลงของค่า RSSI จะลดลงเมื่อระยะทางเพิ่มขึ้น

### อุปกรณ์

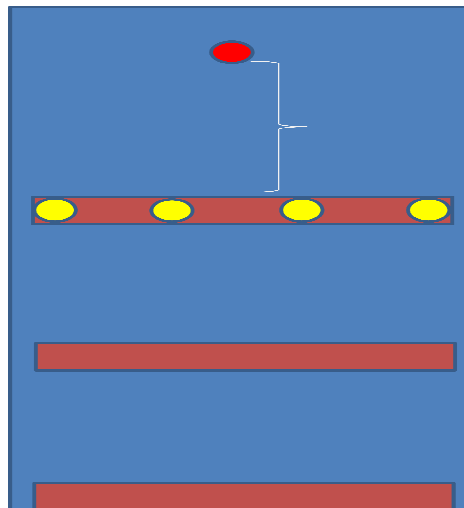
(เหมือนการทดลองที่ 5.1)



ภาพที่ 5.9 SmartboardRF และ Chipcon CC2430 กับ CC2431

### วิธีการทดลอง

- ทำการจัดวางโหนดตัวส่ง ( Blind node ) ไว้ตำแหน่งจุดเริ่มต้น และวางโหนดตัวรับ (Reference node ) ไว้ห่างระยะ 1 เมตรดังภาพที่ 5.10
- หาค่า RSSI จากระบบการค้นหาตำแหน่งวัตถุ เก็บค่าทั้งหมดระยะ 20 ค่าและทำการหาค่าเฉลี่ยและบันทึกผลค่าที่ได้
- ทำเช่นนี้จนครบ 10 เมตรและบันทึกผล

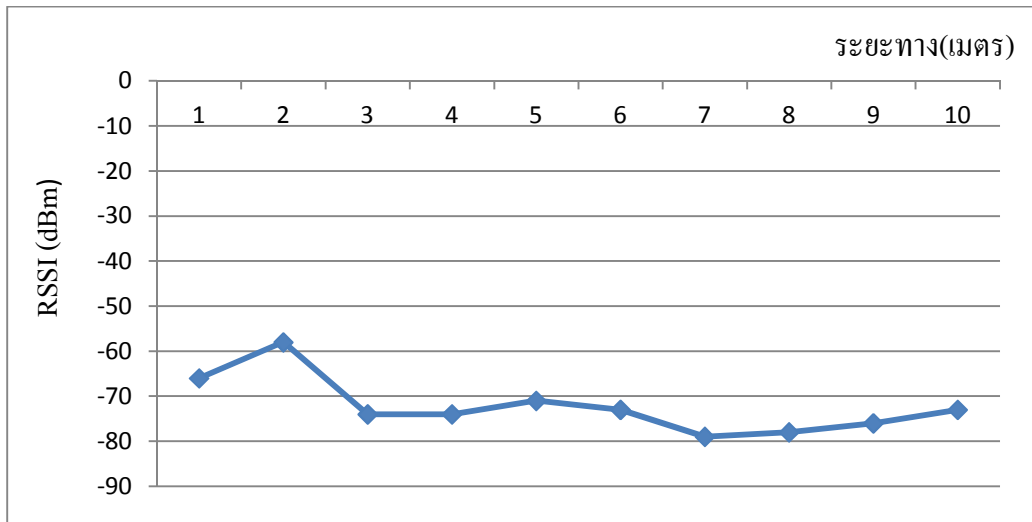


ภาพที่ 5.10 ภาพจำลองการทดลองการจัดวางตำแหน่งของ Blind node และ Reference node

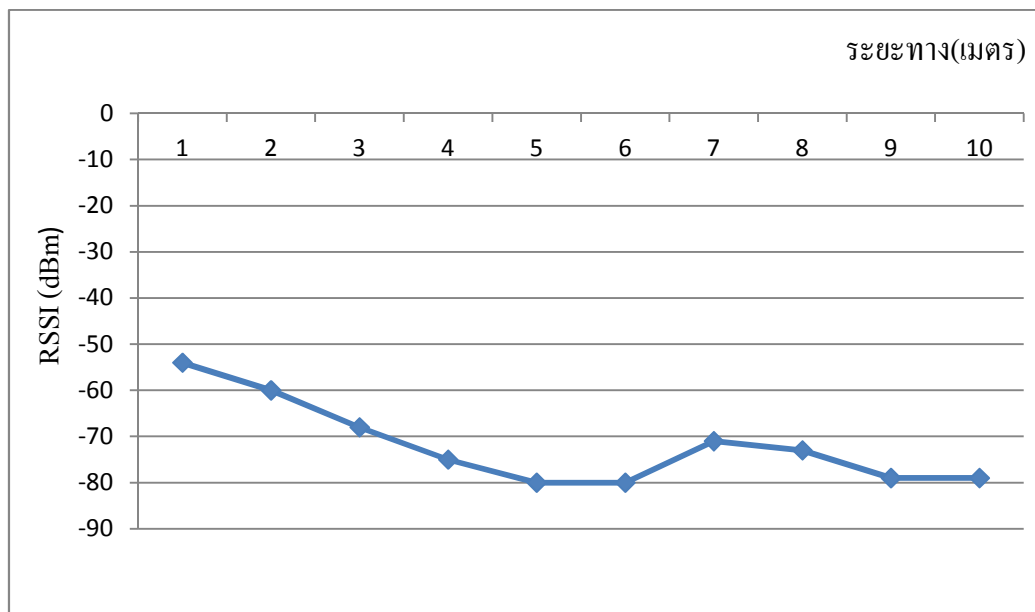
### สภาพแวดล้อม

- ภายในอาคารหอพัก ช่องทางเดินชั้น 2 อาคาร M-Service Apartment

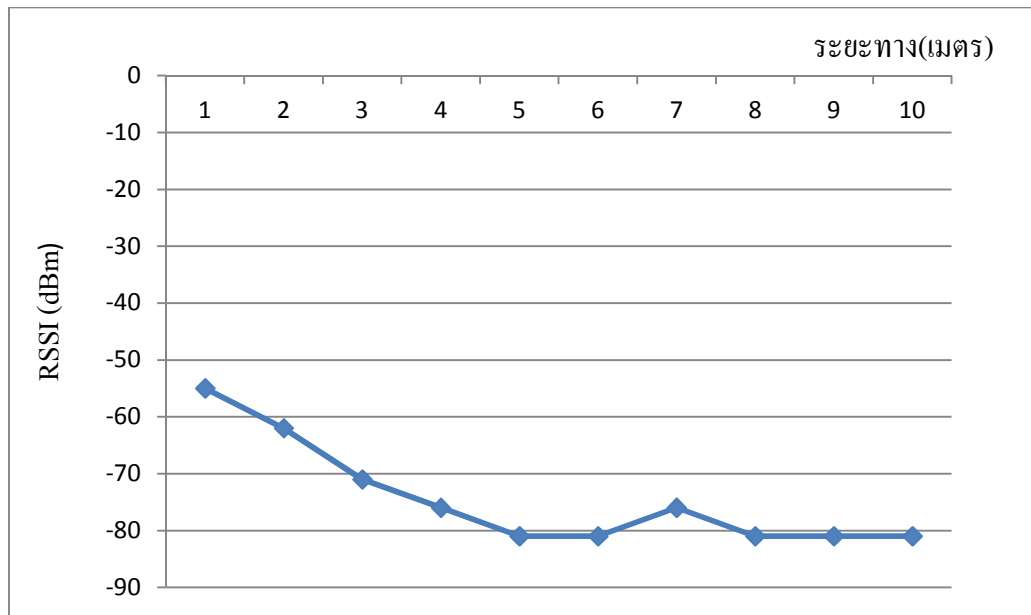
### ผลการทดลอง



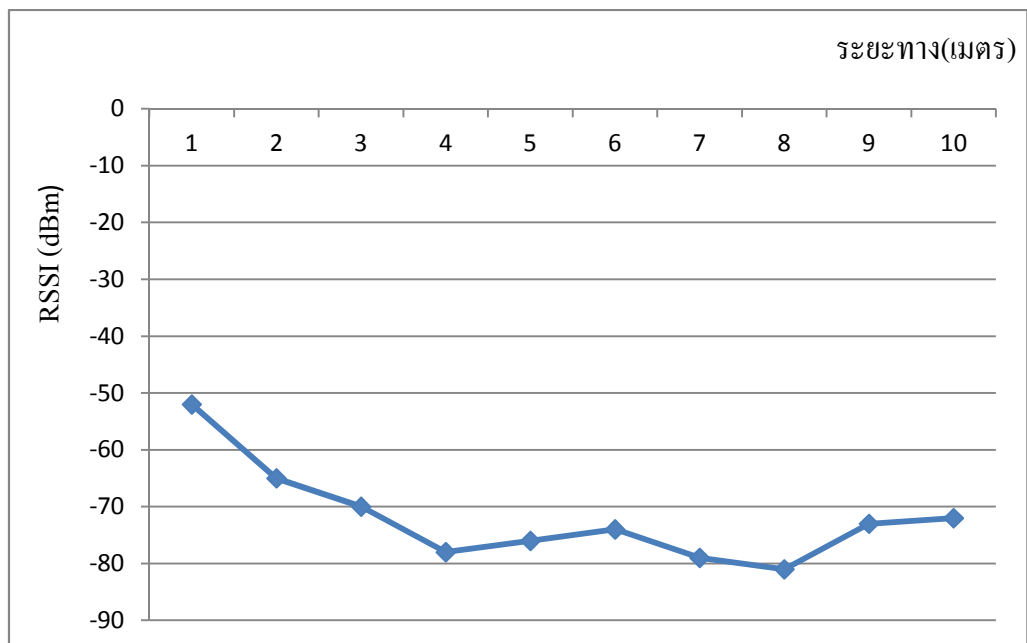
ภาพที่ 5.11 กราฟแสดงแนวโน้มค่า RSSI ของโหนดตัวที่ 1 ทุกช่วง 1 เมตรในระยะทาง 10 เมตร ภายในอาคาร



ภาพที่ 5.12 กราฟแสดงแนวโน้มค่า RSSI ของโหนดตัวที่ 2 ทุกช่วง 1 เมตรในระยะทาง 10 เมตร ภายในอาคาร



ภาพที่ 5.13 กราฟแสดงแนวโน้มค่า RSSI ของโหนดตัวที่ 3 ทุกช่วง 1 เมตรในระยะทาง 10 เมตร  
ภายในอาคาร



ภาพที่ 5.14 กราฟแสดงแนวโน้มค่า RSSI ของโหนดตัวที่ 4 ทุกช่วง 1 เมตรในระยะทาง 10 เมตร  
ภายในอาคาร

### สรุปผลการทดลอง

ค่าความแรงสัญญาณ (RSSI) ที่วัดได้ภายในอาคารมีค่าสัญญาณขึ้นและลงไม่แน่นอน โดยอ้างอิงจากผลกราฟ 5.11 – 5.14 แสดงแนวโน้มค่า RSSI ระหว่างโหนดตัวส่งและโหนดตัวรับ ตั้งแต่ระยะ 1 – 10 เมตร พบว่า สัญญาณไม่มีความคงที่และมีการเปลี่ยนแปลงอยู่ตลอดเวลา เนื่องจากมีปัจจัยสิ่งแวดล้อมรอบข้างเข้ามารบกวน จึงส่งผลให้เกิดการสะท้อนกัน หรือการหักเหของสัญญาณ ทำให้ค่าสัญญาณที่เก็บมาได้จากจุดการทดลองของแต่ละระยะให้ผลค่า RSSI ที่ไม่คงที่ โดยช่วงค่าสัญญาณอยู่ในช่วงระหว่าง -45 ถึง -95 dBm โดยภาพรวมแล้วพบว่าการทดลองมีความสอดคล้องตรงตามสมมติฐานที่กำหนดไว้ในตอนต้น เมื่อระยะระหว่างโหนดตัวส่ง (Blind node) ห่างจากโหนดตัวรับ (Reference node) มากเท่าไรค่าสัญญาณ RSSI จะยิ่งลดน้อยลงไปด้วย จากกราฟที่แสดงผลชัดเจนที่สุดคือ 5.13 และ 5.14 เมื่อผู้วิจัยได้ทำการติดตั้งตำแหน่งของโหนดตัวรับ (Reference node) ตัวที่ 3 และตัวที่ 4 ให้มีระยะห่างจากโหนดตัวส่ง (Blind node) เท่ากับ 1 เมตรพบว่าค่า RSSI มีค่าอยู่ในช่วงระหว่าง -50 ถึง -55 dBm แต่เมื่อทดลองเปลี่ยนโดยการเพิ่มระยะขึ้นเป็น 2 เมตรค่าสัญญาณก็จะมาอยู่ในช่วงระหว่าง -60 ถึง -65 dBm จากนั้นลองเปลี่ยนโดยการเพิ่มระยะไปเรื่อย ๆ จนถึงระยะ 10 เมตรพบว่าค่า RSSI ลดลงเหลือเท่ากับ -80 dBm จึงสรุปได้ว่าระยะทางมีผลให้ค่า RSSI เปลี่ยนแปลง เมื่อระยะทางระหว่างโหนดตัวส่งและโหนดตัวรับใกล้กันค่า RSSI ก็จะมีค่ามากขึ้น แต่ถ้าระยะทางไกลกันออกไปจะส่งผลให้ค่า RSSI ลดน้อยลงตามไปด้วยเช่นกัน

## บทที่ 6

### สรุปผลการวิจัย อภิปรายผล และข้อเสนอแนะ

#### 6.1 สรุปผลการวิจัย

ในทางปฏิบัติระบบสามารถค้นหาตำแหน่งวัตถุด้วยการแปลงสัญญาณมาเป็นระยะทางและนำมาเข้าอัลกอริทึมเพื่อสืบหาตำแหน่งวัตถุที่แท้จริงโดยงานวิจัยนี้ เลือกอัลกอริทึม Maximum Likelihood [27] เนื่องจากเป็นเทคนิคที่นำหลักการของการคิดสามเหลี่ยมระยะ (Trilateration) มาประยุกต์ใช้ทำให้สามารถตอบตำแหน่งวัตถุได้ดีกว่าและมีข้อผิดพลาดน้อยกว่าวิธีของสามเหลี่ยม เนื่องจากจากวิธีคำนวณของสามเหลี่ยมจะใช้จำนวนโหนด 3 ตัวรับส่งสัญญาณเพื่อให้เกิดเป็นรัศมีวงกลมที่เกิดจุดตัด 3 จุดแบบพอดีจึงจะทำให้ได้คำตอบที่เป็นพิกัดวัตถุ แต่ถ้าหากวงกลมนั้นไม่สามารถตัดกันหรือตัดกันแล้วแต่มีจุดตัดมากกว่า 1 จุดอัลกอริทึมสามเหลี่ยมก็จะไม่สามารถตอบพิกัดวัตถุ แต่จะตอบเป็นบริเวณที่วัตถุนั้นแทน จึงทำให้ไม่ได้คำตอบของวัตถุที่เป็นพิกัด X และ Y ในขณะที่วิธีของอัลกอริทึม Maximum Likelihood จะใช้จำนวนโหนด 4 ตัวขึ้นไป ใช้หลักของทางคณิตศาสตร์เชิงสถิติ ที่มีหลักการคิดหาตำแหน่งพิกัดวัตถุจากการถอดสมการเมทริกซ์ ก็จะทำให้ได้พิกัดของวัตถุ

อัลกอริทึมหนึ่งที่ผู้วิจัยเลือกใช้เพื่อมาหาตำแหน่งของวัตถุคือ อัลกอริทึม Min-Max [26] เป็นอัลกอริทึมที่ใช้หลักการวาดกรอบสี่เหลี่ยมล้อมรอบโหนดตัวรับแต่ละตัวโดยมีรัศมีเป็น 2 เท่าจากโหนดตัวส่ง (Blind node) และใช้หลักการการเลือกค่ามากที่สุดแกน Xmax จากค่าน้อยสุดบนแกน X และเลือกค่าน้อยสุด Xmin จากค่ามากที่สุดบนแกน X ส่วนแกน Y เช่นเดียวกันเพื่อวาดให้สี่เหลี่ยมเกิดจุดตัดกันภายใน สุดท้ายการตอบพิกัดวัตถุจะใช้วิธีการคือ หาค่าเฉลี่ยทั้งของ Xmax และ Xmin กับ Ymax และ Ymin เป็นอัลกอริทึมที่มีวิธีการคิดที่ไม่ยุ่งยากและซ้ำซ้อนสุดท้ายอัลกอริทึมของ บริษัทผู้ผลิตฮาร์ดแวร์ (TI) [22] ทางผู้วิจัยจะนำผลการทดลองมาเปรียบเทียบด้วย

จากผลการทดลองทั้งสองสภาพแวดล้อมพบว่าค่าผิดพลาดเฉลี่ยเท่ากับ 1- 4 เมตร และ ผลการทดสอบระบบการค้นหาตำแหน่งวัตถุแบบแบบภายในอาคารของผู้วิจัย โดยแบ่งออกเป็นแบบระยะต่ำกว่า 10 เมตรมีค่าผิดพลาดเฉลี่ยไม่เกิน 1 – 2.7 เมตรและแบบเกินระยะ 10 เมตรค่าความผิดพลาดเฉลี่ยเท่ากับ 2 – 3.58 เมตร ซึ่งคาดว่าสิ่งรบกวนที่อยู่ภายในอาคาร อาทิ เช่น ผนังฉากัน (Partition) สัญญาณคลื่นความถี่เดียวกับอุปกรณ์ อาทิ Wi-Fi กระดาษกาว ก่องลัง กระดาษ เครื่องใช้สำนักงานอื่น ๆ ซึ่งปัจจัยเหล่านี้ล้วนแล้วแต่ส่งผลกระทบต่อการรับส่งสัญญาณ



ระหว่างโหนดตัวส่ง (Blind node) และโหนดตัวรับ (Reference node) จึงส่งผลให้ค่าความผิดพลาดดังกล่าว

โดยในทางปฏิบัติระบบการค้นหาวัดถุแบบภายในอาคารสามารถค้นหาได้จริงเมื่อวัดถุถูกเคลื่อนที่เปลี่ยนตำแหน่งไประบบยังคงสามารถตอบตำแหน่งได้แบบทันการณ์ (Real Time) แม้ว่าผลการค้นหาจะมีข้อผิดพลาดของตำแหน่ง แต่ทางผู้วิจัยได้ทำการวิเคราะห์เพิ่มเพื่อดูค่าแนวโน้มหรือทิศทางของตำแหน่งวัดถุ พบว่ากรณีผิดพลาดไป 1-2 เมตรผู้ใช้งานยังคงสามารถหาสิ่งของได้อยู่แต่กรณีผิดพลาด 3-4 เมตรระบบยังตอบเป็นทิศทางให้ทราบและพอที่จะคาดเดาได้ว่าวัดถุอยู่ทางทิศใดของแผนที่เนื่องจากการตอบตำแหน่งวัดถุ อัลกอริทึม Maximum Likelihood จะตอบได้มีค่าคงที่ของค่าตอบมากที่สุด รองมาคือ Min-Max สามารถดูแนวโน้มตัวอย่างการตอบพิกัดจาก(ภาคผนวก ก.) และระบบยังสามารถหาวัดถุแบบภายนอกอาคารได้โดยค่าความผิดพลาดเฉลี่ยอยู่ในช่วงระหว่าง 0.8 – 1.4 เมตรเท่านั้น ผลการตอบพิกัดจะดีกว่าแบบภายในอาคารเนื่องมาจากภายนอกมีสิ่งรบกวนสัญญาณที่น้อยกว่าแบบภายในอาคาร

สรุปได้ว่า ระบบสามารถค้นหาได้ตรงตามวัดถุประสงค์และตอบตำแหน่งพิกัดได้อย่างทันการณ์ (Real Time) สามารถค้นหาตำแหน่งแบบภายนอกอาคารได้ด้วยแต่ ผู้ใช้งานต้องเลือกโหมดให้ถูกต้องต่อการใช้งานในสภาพแวดล้อมนั้น ดูจากวิธีการสอน (ภาคผนวก ค ) โดยค่าผิดพลาดเกิดขึ้นได้เนื่องจากความไม่แน่นอนของสัญญาณ แต่ผู้ใช้สามารถคาดคะเนตำแหน่งได้ถูกโดยพิจารณาจะพิกัดหรือทิศทางของตำแหน่งบนแผนที่จำลอง (GUI map) ได้

## 6.2 ผลการเปรียบเทียบความแม่นยำของอัลกอริทึม Maximum Likelihood, Min-Max และ TI

การหาตำแหน่งวัดถุแบบภายนอกอาคาร อัลกอริทึม Maximum Likelihood มีความแม่นยำในการตอบพิกัดตำแหน่งมากที่สุด จากค่าความผิดพลาดเฉลี่ยเท่ากับ 0.81 เมตร รองมาคือ Min-Max เท่ากับ 1.48 เมตร และอัลกอริทึม TI เท่ากับ 2.13 เมตร

การหาตำแหน่งวัดถุแบบภายในอาคาร อัลกอริทึม Min-Max มีความแม่นยำในการตอบพิกัดตำแหน่งมากที่สุด โดยค่าผิดพลาดเฉลี่ยเท่ากับ 1.53 เมตร รองลงมาคือ อัลกอริทึม Maximum Likelihood เท่ากับ 1.60 เมตรและ อัลกอริทึม TI เท่ากับ 2.74 เมตร

สรุปการเปรียบเทียบผลการทดลอง จาก 2 สภาพแวดล้อมทั้งแบบภายในและแบบภายนอกอาคาร ในระยะเท่ากับ 10 เมตรเท่ากัน พบว่า กรณีการค้นหาตำแหน่งวัดถุแบบพื้นที่โล่งแจ้งอัลกอริทึม Maximum Likelihood จะมีความแม่นยำในการตอบตำแหน่งวัดถุดีกว่าอัลกอริทึม Min-Max

เนื่องจาก ไม่มีการสะท้อนสัญญาณ หรือหักเห ดังนั้นเมื่อมีการรับสัญญาณครบถ้วนจากโหนดตัวรับ การคำนวณผลแบบวิธีเชิงเมทริกซ์แบบ Maximum Likelihood ดีกว่าวิธีการคำนวณแบบประมาณการด้วยการหาค่าเฉลี่ยแบบ Min-Max

### 6.3 ผลการเปรียบเทียบความคงที่ของอัลกอริทึม Maximum Likelihood , Min-Max และ TI

การหาตำแหน่งวัตถุแบบภายนอกอาคารและแบบภายในอาคาร อัลกอริทึมที่มีความคงที่ในการตอบพิกัดตำแหน่งมากที่สุดคือแบบภายนอกอาคาร คือ อัลกอริทึม Maximum Likelihood ค่าเบี่ยงเบนมาตรฐาน(Standard Deviation) เท่ากับ 0.02 เมตร และแบบภายในอาคาร เท่ากับ 0.19 เมตร รองมาคืออัลกอริทึม Min-Max แบบภายนอกอาคารเท่ากับ 0.12 เมตรและแบบภายในอาคาร 0.22 เมตร และ TI แบบภายนอกอาคารเท่ากับ 0.36 เมตรและแบบภายในอาคาร 0.96 เมตร

สรุปผลจากการเปรียบเทียบค่าความคงที่ในการตอบพิกัดของอัลกอริทึมทั้งสองสภาพแวดล้อมพบว่าทั้งแบบภายในอาคารที่มีสิ่งรบกวนมากมายกับแบบภายนอกอาคารที่มีสิ่งรบกวน พบว่าอัลกอริทึม Maximum Likelihood ให้ค่าคำตอบที่คงที่มากที่สุดโดยแนวโน้มของคำตอบจะเป็นไปทางทิศเดียวกันจึงสะดวกต่อผู้ใช้งานที่จะสืบหาวัตถุที่สามารถประมาณหรือคาดการณ์ได้อย่างชัดเจนว่า วัตถุอยู่น้อยตำแหน่งหรือทิศทางใดของพื้นที่จริงบนแผนที่จำลอง

### 6.4 สรุปความแม่นยำของการหาตำแหน่งวัตถุแบบภายในและภายนอกอาคาร

สรุปผลการทดลองแบ่งเป็น จุดที่ Valid ระหว่างสภาพแวดล้อมแบบภายในอาคารและภายนอกอาคาร

#### แบบภายในอาคาร

	Maximum Likelihood	Min-Max	TI
Average	1.60	1.53	2.74
SD.	0.19	0.22	0.96

ตารางที่ 6.1 แสดงการเปรียบเทียบการหาตำแหน่งความแม่นยำของทั้ง 3 อัลกอริทึมแบบภายในอาคาร

### แบบภายนอกอาคาร

	Maximum Likelihood	Min-Max	TI
Average	1.00	1.41	1.81
SD.	0.02	0.13	0.33

ตารางที่ 6.2 แสดงการเปรียบเทียบการหาค่าความแม่นยำของทั้ง 3 อัลกอริทึมแบบภายนอกอาคาร

จากผลสรุปได้ว่า ระบบจะสามารถหาวัตถุแบบภายนอกอาคารได้แม่นยำกว่าแบบภายในอาคาร โดย อัลกอริทึม Maximum Likelihood จะตอบค่าตำแหน่งได้แม่นยำที่สุดและมีค่าความผิดพลาดเฉลี่ยเท่ากับ 1 เมตรรองมา คือ Min-Max เท่ากับ 1.41 เมตรและ TI เท่ากับ 1.81 เมตร ค่าเฉลี่ยความผิดพลาดของการตอบตำแหน่งของระบบ จะอยู่ในช่วงระหว่าง 1 – 2 เมตร

ขณะที่การหาตำแหน่งวัตถุแบบภายใน ที่มีสิ่งรบกวนมากกว่า ระบบสามารถหาตำแหน่งวัตถุได้ด้วยอัลกอริทึม Min-Max ที่ตอบตำแหน่งวัตถุและให้ค่าความผิดพลาดเฉลี่ยเท่ากับ 1.53 เมตรรองมาคือ อัลกอริทึม Maximum Likelihood เท่ากับ 1.60 เมตรและ TI เท่ากับ 2.74 เมตร ค่าเฉลี่ยความผิดพลาดของการตอบตำแหน่งของระบบ จะอยู่ในช่วงระหว่าง 1.5 – 3 เมตร

กรณีวัดที่ค่าเฉลี่ยความเบี่ยงเบนมาตรฐานในการตอบตำแหน่งพบว่าทั้ง 2 สภาพแวดล้อม อัลกอริทึม Maximum Likelihood ตอบตำแหน่งได้อย่างคงที่มากที่สุด คือแบบภายในเท่ากับ 0.19 และแบบภายนอก 0.02

สรุปผลการทดลองแบ่งเป็น จุดที่ Invalid ระหว่างสภาพแวดล้อมแบบภายในอาคารและภายนอกอาคาร

### แบบภายในอาคาร

	Maximum Likelihood	Min-Max	TI
Average	3.65	2.95	3.29
SD.	1.17	0.56	0.20

ตารางที่ 6.3 แสดงการเปรียบเทียบการหาค่าความแม่นยำของทั้ง 3 อัลกอริทึมแบบภายในอาคารแบบ Invalid Length

### แบบภายนอกอาคาร

	Maximum Likelihood	Min-Max	TI
Average	3.20	2.40	3.04
SD.	0.07	0.16	1.57

ตารางที่ 6.4 แสดงการเปรียบเทียบการหาค่าความแม่นยำของทั้ง 3 อัลกอริทึมแบบภายนอก

#### อาคารแบบ Invalid Length

จากผลสรุปได้ว่า ระบบจะสามารถหาวัตถุแบบภายนอกอาคารได้แม่นยำกว่าแบบภายในอาคาร โดย อัลกอริทึม Min-Max จะตอบค่าตำแหน่งได้ดีที่สุดและมีค่าความผิดพลาดเฉลี่ยเท่ากับ 2.40 เมตร รองมา คือ TI เท่ากับ 3.04 เมตร และ Maximum Likelihood เท่ากับ 3.20 เมตร ค่าเฉลี่ยความผิดพลาดของการตอบตำแหน่งของระบบ จะอยู่ในช่วงระหว่าง 2.4 – 3.2 เมตร

ขณะที่การหาตำแหน่งวัตถุแบบภายใน ที่มีสิ่งรบกวนมากกว่า ระบบสามารถหาตำแหน่งวัตถุได้ด้วยอัลกอริทึม Min-Max ที่ตอบตำแหน่งวัตถุและให้ค่าความผิดพลาดเฉลี่ยเท่ากับ 2.95 เมตร รองมาคือ อัลกอริทึม TI เท่ากับ 3.29 เมตร และ Maximum Likelihood เท่ากับ 3.65 เมตร ค่าเฉลี่ยความผิดพลาดของการตอบตำแหน่งของระบบ จะอยู่ในช่วงระหว่าง 2.9 - 4 เมตร

กรณีวัดที่ค่าเฉลี่ยความเบี่ยงเบนมาตรฐานในการตอบตำแหน่งพบว่าแบบภายใน อัลกอริทึม TI จะตอบตำแหน่งได้อย่างคงที่มากที่สุด เท่ากับ 0.20 รองมาคือ Min-Max เท่ากับ 0.56 และ Maximum Likelihood เท่ากับ 1.17

กรณีวัดที่ค่าเฉลี่ยความเบี่ยงเบนมาตรฐานในการตอบตำแหน่งพบว่าแบบภายนอก อัลกอริทึม Maximum Likelihood จะตอบตำแหน่งได้อย่างคงที่มากที่สุด เท่ากับ 0.07 รองมาคือ Min-Max เท่ากับ 0.16 และ TI เท่ากับ 1.57

### 6.5 ผลการเปรียบเทียบประสิทธิภาพในการหาตำแหน่งวัตถุของ อัลกอริทึม Maximum Likelihood และ อัลกอริทึม Min-Max ของระบบกับงานวิจัยอื่น

จากบทความ “Comparative evaluation of Received Signal-Strength Index(RSSI) based indoor localization techniques for construction jobsites” ของผู้เขียน Xiaowei Luo , William J. O'Brien และ Christine L. Julien

งานวิจัยนี้ได้จัดทำทดลองเปรียบเทียบค่าการหาความแม่นยำตำแหน่งวัตถุแบบภายในอาคาร ด้วยอัลกอริทึมหลัก 3 อัลกอริทึม คือ อัลกอริทึม Min-Max , Maximum Likelihood

,Ring Overlapping Circle RSSI และใช้อัลกอริทึม K-Nearest Neighbor มาหาข้อผิดพลาดจากตำแหน่งที่คำนวณมาได้กับตำแหน่งจริงของวัตถุ

โดยการทดลองจะใช้การดักจับของสัญญาณ (RSSI) จากวัตถุด้วยอุปกรณ์คือ Crossbow Technology ที่มีตัวส่ง (Blind node) และตัวรับ (Reference node) และจะมีตัวบอร์ด (MIB520) ที่จะมาต่อผลออกสู่จอคอมพิวเตอร์

การทดลอง จะทำการเปรียบเทียบ 2 พื้นที่ที่แตกต่างกัน

- สถานที่ ห้องเรียนที่มีขนาด 7.0 X 6.4 เมตร
- สถานที่ ใต้อาคารที่มีขนาดกว้างเป็น 2 เท่าของ

โดยจะติดตั้งโหนดตัวรับ 4 โหนดตามมุมห้องเรียนและโหนดตัวส่งอยู่ภายในตารางกริดบนซากห้อง เพื่อให้อยู่สูงจากพื้นเล็กน้อยเพื่อกำจัดสิ่งรบกวนต่อสัญญาณให้ลดน้อยลง พบว่าได้ผลการเปรียบเทียบเป็นดังนี้

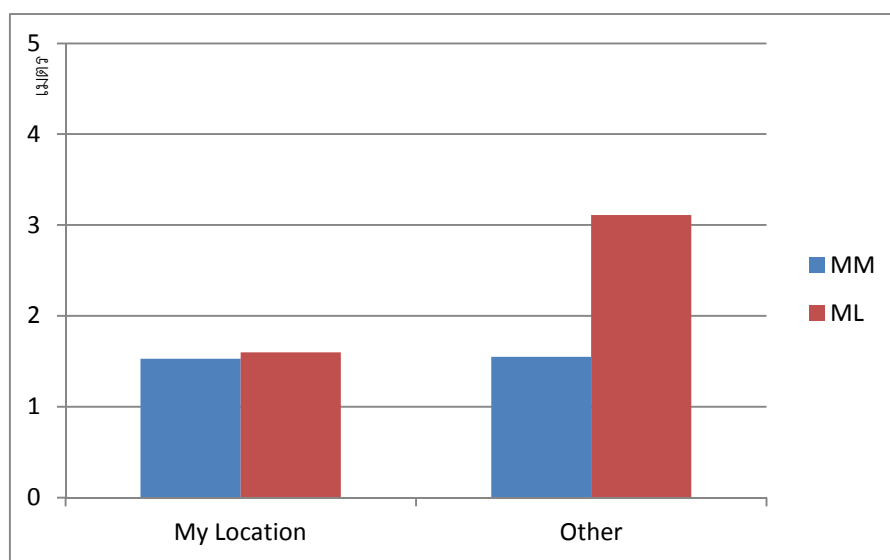
Indoor	Min-Max		Maximum Likelihood		Node
	Avg.	SD	Avg.	SD	
My Localization	1.53	0.22	1.60	0.19	4
Other Localization	1.55	0.73	3.11	1.56	4

ตารางที่ 6.5 แสดงการเปรียบเทียบผลการหาตำแหน่งวัตถุระหว่างระบบค้นหาที่ระบบอื่นแบบภายในอาคาร

### บทสรุปแบบภายในอาคาร

จากผลการทดลองของการหาตำแหน่งวัตถุแบบภายในอาคาร พบว่าอัลกอริทึม Min-Max มีความแม่นยำในการหาตำแหน่งวัตถุที่ดีที่สุด เมื่อเทียบระหว่างผลการทดลองของระบบกับงานวิจัยอื่น [27] ดังตาราง 6.5 พบว่าอัลกอริทึม Min-Max ของผู้วิจัยตอบตำแหน่งที่มีค่าความผิดพลาดเฉลี่ยเท่ากับ 1.53 เมตรขณะที่งานวิจัยอื่น [27] เท่ากับ 1.55 เมตร และอัลกอริทึม Maximum Likelihood มีค่าความผิดพลาดเฉลี่ยเท่ากับ 1.60 เมตรขณะที่งานวิจัยอื่นเท่ากับ 3.11 เมตร แต่ค่าการตอบตำแหน่งมีความคงที่ของพิกัดวัตถุ พบว่าค่าส่วนเบี่ยงเบนมาตรฐานเฉลี่ยระบบผู้วิจัยสามารถตอบได้คงที่ที่สุดด้วยอัลกอริทึม Maximum Likelihood เท่ากับ 0.19 ขณะที่อัลกอริทึม Min-Max เท่ากับ 0.22 และในงานวิจัยอื่นอัลกอริทึมที่ตอบตำแหน่งได้ค่าคงที่มากที่สุดคือ Min-Max เท่ากับ 0.73 และ Maximum Likelihood เท่ากับ 1.56

สรุปผลการทดลองและเปรียบเทียบผลของผู้วิจัยและงานวิจัยอื่น พบว่า อัลกอริทึม Min-Max มีค่าความแม่นยำมากที่สุดในการตอบตำแหน่ง แคตอัลกอริทึมของการตอบพิกัดได้คงที่มากที่สุด คือ อัลกอริทึม Maximum Likelihood



ภาพที่ 6.1 แสดงการเปรียบเทียบค่าความผิดพลาดเฉลี่ยของผลการหาตำแหน่งวัตถุระหว่างระบบของผู้วิจัยเองกับงานวิจัยอื่นแบบภายในอาคาร

## 6.6 แนวทางการนำไปใช้งาน

สามารถนำระบบไปใช้งานได้จริงทั้งแบบการหาตำแหน่งวัตถุแบบภายในอาคารและแบบภายนอกอาคาร ค่าผิดพลาดที่เกิดขึ้นอยู่ในช่วงระยะ 0.8 – 3.8 เมตร ซึ่งคาดว่าสามารถระบุห้องหรือตำแหน่งที่วัตถุอยู่ได้แม่นยำเพียงพอที่ผู้ใช้งานสามารถหาวัตถุได้ โดยสามารถพิจารณาตำแหน่งพิกัดร่วมกับแผนที่ (GUI map) เนื่องจากคำตอบที่ได้รับตำแหน่งของวัตถุจะตอบไปในทิศทางเดียวกันจึงช่วยให้ง่ายและสะดวกต่อการคาดเดาตำแหน่งยิ่งขึ้น

## 6.7 อภิปรายผล

ในการหาตำแหน่งวัตถุแบบภายนอกอาคารและแบบภายในอาคาร จากผลการทดลองพบว่าระบบสามารถตอบตำแหน่งแบบภายนอกอาคารได้แม่นยำกว่าแบบภายในอาคาร ซึ่งในทางปฏิบัติทั้งสองสภาพแวดล้อมล้วนแล้วมีสิ่งรบกวนต่อสัญญาณวิทยุที่ใช้ในการรับส่งกันระหว่างโหนดตัวส่ง (Blind node) และโหนดตัวรับ (Reference node) ซึ่งจากผลการทดลองพบว่าการค้นหาตำแหน่งของวัตถุแบบภายในอาคารจะให้ค่าความผิดพลาดมากกว่าแบบ

ภายนอกอาคาร เนื่องจากสิ่งรบกวนที่มีอยู่ภายในอาคารที่มากกว่า อาทิเช่น ผนังกันห้อง (Partition) , กระจกเงา , เครื่องถ่ายเอกสาร , สัญญาณไร้สาย (Wi-Fi) , กล้องวงจรปิด , อุปกรณ์เครื่องใช้สำนักงาน รวมทั้งพนักงาน ในขณะที่ภายนอกอาคาร อาทิเช่นคลื่นสัญญาณ การเดินทางของผู้คน ซึ่งสิ่งรบกวนต่าง ๆ ที่ได้กล่าวมาในตอนต้นมีผลกระทบต่อคลื่นสัญญาณวิทยุทำให้สัญญาณขาดหายไปในช่วงหนึ่งเนื่องจากการบดบังสัญญาณจากเสารับส่ง ( Antenna ) หรือการสะท้อน การหักเหของสัญญาณเมื่อนำวัตถุไปติดตั้งบริเวณติดชิดผนังห้อง หรือชิดกระจกมากเกินไป ก็ส่งผลให้การรับและส่งสัญญาณมีปัญหาได้ เมื่อนำค่าสัญญาณมาคำนวณเพื่อแปลงเป็นระยะทางก็จะได้ค่าที่ผิดพลาดด้วยเช่นกัน

ในการค้นหาตำแหน่งวัตถุทั้งแบบภายในอาคารและแบบภายนอกอาคารระยะทางมีความสำคัญ เมื่อระยะเพิ่มมากขึ้นย่อมส่งผลกระทบต่อสัญญาณที่รับได้น้อยลงด้วย(สมการ 5.1) ขึ้นอยู่กับสิ่งรบกวนในช่วงขณะนั้น การหาตำแหน่งแบบภายนอกอาคาร อัลกอริทึมที่มีความแม่นยำมากที่สุดคือ Maximum Likelihood และการค้นหาแบบภายในอาคารอัลกอริทึมที่มีความแม่นยำมากที่สุดคือ Min-Max และเมื่อได้ทำการเปรียบเทียบกับงานวิจัยอื่น (การทดลอง 6.5) พบว่าอัลกอริทึม Min-Max เช่นเดียวกันที่ค่าความแม่นยำกว่า Maximum Likelihood สาเหตุเนื่องมาจากข้อจำกัดของอัลกอริทึม Maximum Likelihood จะต้องรับโหนดอย่างน้อย 4 ตัวขึ้นไปเพื่อมาคำนวณแต่จากการทดลองพบว่า มีโหนดในบางตัวไม่สามารถรับสัญญาณได้จึงทำให้คงเหลือโหนด 3 ตัวในการคำนวณตำแหน่ง แต่ขณะเดียวกันอัลกอริทึม Min-Max ยังคงสามารถคำนวณได้เพราะสามารถรับโหนดได้ 3 ตัวเป็นอย่างน้อยจึงทำให้สามารถคำนวณได้ผลที่ยังคงแม่นยำอยู่ แต่ทั้งแบบภายในและภายนอกอาคารอัลกอริทึมที่ตอบตำแหน่งได้ค่าคงที่มากที่สุดคืออัลกอริทึม Maximum Likelihood

## 6.8 ข้อจำกัด ปัญหาและอุปสรรค

### 6.8.1 ข้อจำกัด

- อุปกรณ์จะสามารถรับและส่งสัญญาณกันได้ในเสาอากาศ (Antenna) ต้องอยู่ในที่ไม่ถูกบดบังสัญญาณ หากกรณีวางอุปกรณ์ไว้ในตู้เก็บเอกสาร ทำให้สัญญาณไม่สามารถรับได้ ค่าสัญญาณ RSSI จะถูกกำหนดว่าเท่า  $-81\text{dBm}$  ระยะทางเท่ากับ 10.46 เมตร

### 6.8.2 ปัญหาและอุปสรรค

- ไมโครชิปรุ่น CC2431/CC2430 เกิดอาการค้างแก้ไขโดยการลงโปรแกรม Firmware Embedded บนตัวชิปใหม่
- เสาอากาศที่เชื่อมต่อกับทุกอุปกรณ์มีความเปราะบางมากเกินไป ง่ายต่อการแตกหัก

### 6.8.3 ข้อเสนอแนะ

- ควรติดตั้งอุปกรณ์ให้อยู่ในระดับที่สูงเพื่อหลีกเลี่ยงการถูกรบกวนสัญญาณจากปัจจัยภายใน อาทิเช่น เครื่องถ่ายเอกสาร , ฉากกั้น (Partition) ความถี่ของการเดินผ่านของพนักงาน
- การติดตั้งอุปกรณ์ไม่ควรให้อยู่ในมุมอับสัญญาณ อาทิเช่น ตู้ , ช่องแคบ หรือหลังเครื่องคอมพิวเตอร์
- การติดตั้งโหนดตัวรับ (Reference node) ควรติดตั้งให้เป็นกริดที่มีพื้นที่สี่เหลี่ยมที่มีความกว้างและยาวไม่เกิน  $7 \times 7$  เมตร เนื่องจากเป็นระยะทำการสูงสุดของโหนดระหว่างตัวส่ง (Blind node) กับโหนดอ้างอิง (Reference node)

### 6.8.4 แนวทางการวิจัยต่อ

ทางผู้วิจัยประสงค์ที่จะปรับปรุงให้ระบบสามารถค้นหาตำแหน่งวัตถุเพื่อให้มีความแม่นยำมากขึ้น โดยคำนึงถึงปัจจัยข้อมูลนำเข้าที่มีความไม่แน่นอนให้เกิดความน่าเชื่อถือมากยิ่งขึ้น คือ

ตัวแปรค่าความแรงของสัญญาณ (RSSI) มีความสำคัญและส่งผลกระทบต่อค่าความคลาดเคลื่อนในการตอบตำแหน่งวัตถุเป็นอย่างมาก เนื่องจากเป็นตัวแปรที่มีความไม่แน่นอน สัญญาณจะแรงขึ้นหรือต่ำลงขึ้นอยู่กับสภาวะแวดล้อมและสิ่งรบกวนในขณะนั้น ด้วยเหตุนี้ทาง



ผู้วิจัยจึงเสนอแนวทางที่จะทำให้ค่าสัญญาณมีความน่าเชื่อถือมากยิ่งขึ้นด้วยการหาค่าเฉลี่ยของค่าสัญญาณ (RSSI) ด้วยวิธีการหาจำนวนครั้งที่จะเก็บสัญญาณเป็นสิ่งสำคัญที่ต้องพิจารณาว่าควรเก็บกี่ครั้งถึงจะทำให้เฉลี่ยค่าออกมาแล้วได้ผลที่ดีที่สุดและมีความน่าเชื่อถือมากขึ้น ซึ่งสามารถคำนวณได้จากสมการ (5.1) และแปลงเป็นระยะทางได้ดังสมการที่ 6.1

$$d_t = 10^{-\left[\frac{(RSSI_t + A)}{10^n}\right]} \quad (6.1)$$

เมื่อ  $d_t$  คือระยะทางระหว่างโหนดอ้างอิงหรือโหนดตัวรับ (Reference node) ณ.เวลาหนึ่ง RSSI คือ ค่าความแรงสัญญาณ ณ.เวลาที่ทำให้เกิด  $d_t$  ค่า RSSI จะเป็นเลขจำนวนเต็ม 8 บิตซึ่งมีความกว้างข้อมูลอยู่ในช่วงระหว่าง 0 - 255 [20] และทำการหาผลต่างเมื่ออุปกรณ์ถูกติดตั้งให้อยู่จุดเดิมแต่ค่าสัญญาณ RSSI เปลี่ยนทำให้ระยะทางเปลี่ยนไปด้วย (d) ดังสมการ

$$d_{t+1} = 10^{-\left[\frac{(RSSI_{t+1} + A)}{10^n}\right]} \quad (6.2)$$

เอาสมการ 6.1 – 6.2

$$\Delta d = 10^{\left\{\frac{\Delta RSSI}{10 \times n}\right\}}, n \in [1,8] \quad (6.3)$$

ต้องการทำให้  $\Delta d$  มีค่าน้อยมาก ๆ คือมีค่าเข้าใกล้ 0

$$d\left(\frac{\Delta d}{d_t}\right) = 0 \quad (6.4)$$

พจน์ที่จะทำให้  $\Delta d$  น้อยที่สุดที่เป็นไปได้ เมื่อ  $n$  มีค่ามากที่สุดที่เป็นไปได้ คือ 8.000

$$\Delta d_{1,8} = 10^{\left\{\frac{1+\Delta f}{10 \times n}\right\}} \quad (6.5)$$

$$\Delta d_{1,8} = 10^{\left\{\frac{1+\Delta f}{80}\right\}} \quad (6.6)$$

ถ้าหาก  $\Delta f$  มีค่าเป็น 0 จะได้

$$\Delta d_{1,8} = 0.012 \quad (6.7)$$

แสดงว่า ค่า RSSI ที่เป็นเลขจำนวนเต็มสามารถแสดงค่าความแม่นยำสูงสุดได้ 0.0125 เมตร หากต้องการทำให้ระยะ d ละเอียดมากกว่านี้สองเท่าต้องทำให้

$\frac{\Delta f}{80} \rightarrow 1$  แสดงว่า  $\Delta f$  ต้องมีค่า 80 นั่นคือการหาค่าเฉลี่ยอย่างน้อย 80 ครั้งซึ่งคือ

$N \times 10$  ครั้ง ดังนั้นจำนวนครั้งของการเฉลี่ย จะต้องเป็น 10 เท่าของ  $n$  ที่เลือกมาใช้งาน

หมายความว่า ทางผู้วิจัยต้องทำการเก็บค่าความแรงสัญญาณ (RSSI) ทั้งหมดจากโหนดอ้างอิง 4 ค่าด้วยจำนวน  $10 \times n$  ครั้งต่อหนึ่งโหนดอ้างอิง ณ. ตำแหน่งเดียวกัน และ นำค่า RSSI ที่เก็บได้ทั้งหมด  $10 \times n$  ครั้งมาหาค่าเฉลี่ย โดยที่ RSSI ที่ได้จากแต่ละโหนดในหนึ่งคาบเวลา จากสมการ

$$\overline{RSSI} = \frac{1}{n} \sum_{i=0}^n RSSI_i \quad (6.8)$$

จากนั้นนำค่าสัญญาณที่ผ่านการหาค่าเฉลี่ยเรียบร้อยแล้ว ( $\overline{RSSI}$ ) มาทำการคำนวณหาตำแหน่งวัตถุ ซึ่งระบบต้องใช้เวลาในการรอคอยค่าสัญญาณโดยประมาณ 16 วินาที (Time delay  $\cong 16$  s.) ก่อนที่จะนำมาเข้ากระบวนการการหาตำแหน่งวัตถุด้วยอัลกอริทึม Maximum Likelihood และ Min-Max เพื่อจะตอบตำแหน่งวัตถุเป็นแบบพิกัดต่อไป

## รายการอ้างอิง

- [1] IEEE Standard for Information Technology, **Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (WPANs)**, Journal of Telecommunications and Information Exchange Between Systems [Online]. 2006. Available from : [http://www.it-expo.org/docs/9\\_2\\_en.pdf](http://www.it-expo.org/docs/9_2_en.pdf) [2011, April 19]
- [2] Grebennikov, A. **RF and Microwave Power Amplifier Design**. SG : McGraw-Hill Publications, 2005.
- [3] Texas Instruments, **CC2430 Software Example User's Guide**, Texas Instrument, Journal of Chipcon Products [Online]. 2006. Available from : <http://www.ti.com.cn/cn/lit/ug/swru178b/swru178b.pdf> [2011, May 25].
- [4] Farahani, S. **ZIGBEE Wireless Networks and Transceivers**. UK : British Library Publications, 2008.
- [5] สถาบันชีววิทยาศาสตร์โมเลกุล,มหาวิทยาลัยมหิดล. **ทำความเข้าใจกับ On-Demand, Broadcast, Unicast และ Multicast**. [ออนไลน์]. 2553. แหล่งที่มา : [http://www.cc.mahidol.ac.th/newsletter/Old/Vol2\\_2/content\\_5.html](http://www.cc.mahidol.ac.th/newsletter/Old/Vol2_2/content_5.html) [3 มีนาคม 2553]
- [6] ศูนย์เทคโนโลยีอิเล็กทรอนิกส์และคอมพิวเตอร์แห่งชาติ (เนคเทค). **WIMAX รูปแบบการใช้งานในส่วนต่าง ๆ**. [ออนไลน์]. 2553. แหล่งที่มา : <http://wimax-msit51.blogspot.com/2008/09/1433-mesh-topology.html> [15 มีนาคม 2553]
- [7] วัชรกร หนูทอง และอนุกุล น้อยไม้. **RFID หนึ่งในเทคโนโลยีที่น่าจับตามอง**. [ออนไลน์]. 2553. แหล่งที่มา : <http://www.fit.sru.ac.th/thanawit/files/RFID / final2.pdf/> [15 มีนาคม 2553]
- [8] National Electronics and Computer Technology Center(NECTEC), **Ultrawideband (UWB)Technology**, Journal of Business and Industry Development [Online]. 2011. Available from : [http://www.nectec.or.th/bid/mkt\\_info\\_tech\\_ultrawideband.html](http://www.nectec.or.th/bid/mkt_info_tech_ultrawideband.html) [2011, March 19]

### รายการอ้างอิง (ต่อ)

- [9] Want, R., Hopper, A., Falcao, V. and Gibbons, J. **The Active Badge Location System**. Journal of Transactions on Information Systems (TOIS) (January 1992): 91 - 102.
- [10] Wiriaporn Pattarakom. **Ranging in Wireless Sensor Network Systems**. Master's Thesis, Development of Computer Engineering, Faculty of Computer Engineering, Prince of Songkla University, 2008.
- [11] Bahl, P. and Padmanabhan, V. **RADAR : An In-Building RF-Based User Location And Tracking System**. In Proceedings of the 19th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM), 2000, 775-784.
- [12] Sontaya Wilaijit. **An Accuracy Improvement of Indoor Local Positioning System**. Master's Thesis, Department of Computer, Faculty of Engineering, Kasetsart University, 2010.
- [13] Hightower, J., Borriello, G. and Want, R. **SpotON : An Indoor 3d Location Sensing Technology Based on RF Signal Strength**. Journal of UW-CSE 2000-02-02[Online]. 2000. Available from : <http://seattle.intel-research.net/hightower2000indoor.pdf> [2008, April 30]
- [14] **Maximum Likelihood Algorithm and TDoA**. [Online]. Available from : [http://wikipedia.org/wiki/Maximum\\_Likelihood](http://wikipedia.org/wiki/Maximum_Likelihood) [2011, March]
- [15] Priyantha, N.B., Chakraborty, A. and Balakrishnan, H. **The Cricket Location-Support System**, In Proceedings of 6th International Conference on Mobile Computing and Networking (MOBICOM 2000), 2000, 71-78.
- [16] Kanokwan Jomprom, Nuttapon Nakarach and Thanasate Bokam. **Localization and Tracking in Wireless Sensor Network**. Bachelor Thesis, Department of Information Engineering, Faculty of Engineering, King Mongkut's Institute of Technology Ladkrabang, 2010.

### รายการอ้างอิง (ต่อ)

- [17] Lionel, M., Yunhao Liu, Yiu Cho Lau and Abhishek, P. **LANDMARC: Indoor Location Sensing Using Active RFID**, In Proceedings of 1st National Conference on the IEEE Computer and Communications Societies(PerCom 2003), 2003, 407-415.
- [18] ภาคย์ สอนเสาวภาคย์ และชชชชชช คูนบัว. **ระบบระบุตำแหน่งวัตถุภายในอาคารโดยใช้มาตรฐาน IEEE 802.15.4**. ใน สมาคมวิชาการไฟฟ้าอิเล็กทรอนิกส์คอมพิวเตอร์ โทรคมนาคม และสารสนเทศ, หน้า 149 – 154. 4 – 6 พฤษภาคม 2552 ณ.มหาวิทยาลัยขอนแก่น , 2009.
- [19] Jeffrey, H. and Gaetano, B. **Location Sensing in Ubiquitous Computing a comparative review of four seminal works**. Journal of Computer Science (April 2006): 57-66.
- [20] Bulusu, N., Heidemann, J. and Estrin, D. **GPS-Less Low Cost Outdoor Localization for Very Small Devices**. Journal of Computer Science (October 2000): 28-34 .
- [21] Xiaowei Luo, William J.O'Brien and Christine L.Julien. **Comparative evaluation of Received Signal-Strength Index (RSSI) based indoor localization techniques for construction Jobsites**, Journal of Advanced Engineering Informatics.(May 2010): 1-9.
- [22] Aamodt. **CC2431 Application Note AN042 (Rev. 1.0)**, Texas Instrument, Journal of Chipcon Products(2006): 1-20.
- [23] Texas Instruments Chipcon. **CC2431DK Development Kit User manual(Rev.1.5)**, Texas Instrument, Journal of Chipcon Products(2007): 1-33.
- [24] Hongbo Wang, Ke Yu and Hongnian Yu. **Mobile Robot Positioning Based on Zigbee Wireless Sensor Networks and Vision Sensor**, Journal of Modelling(2010) : 184-193.

**รายการอ้างอิง (ต่อ)**

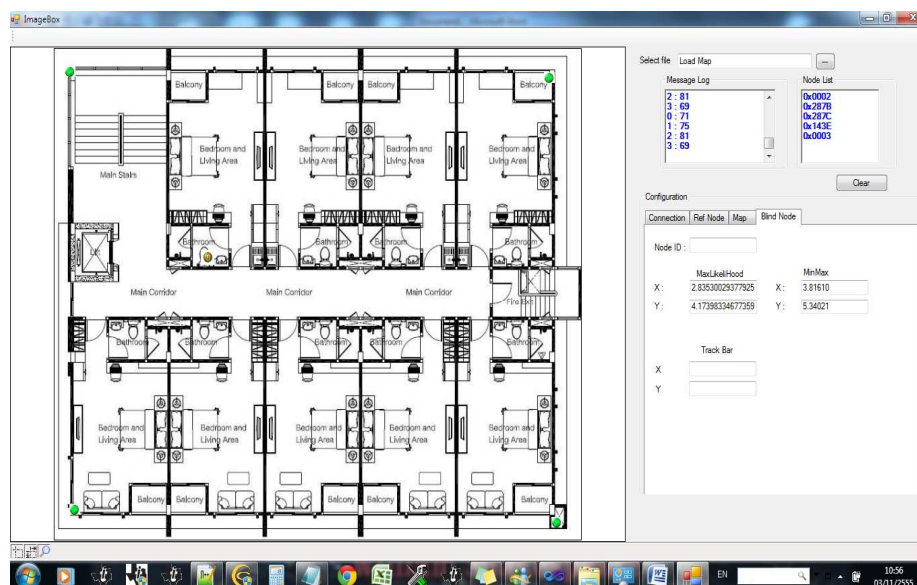
- [25] Chengpei TANG and Jiao .A. Localization Algorithm of Weighted Maximum Likelihood Estimation for Wireless Sensor Network, Journal of Information & Computational Science 8 (December 2011): 4293-4300.
- [26] Riccardo Masiero. RSSI Based Tracking Algorithms for Wireless Sensor Networks : theoretical aspects and performance evaluation. Master's Thesis, Department of Information Engineering, PADOVA University, (2006-2007).
- [27] Sen Wang and Huosheng Hu. Three-Dimensional Localisation using Cricket System. Master's Thesis, School of Computer Science and Electronic Engineering, Essex University, 2011.
- [28] Min-Max Algorithm. [Online]. Available from : <http://en.wikipedia.org/wiki/Minimax> [2011, April].

ภาคผนวก

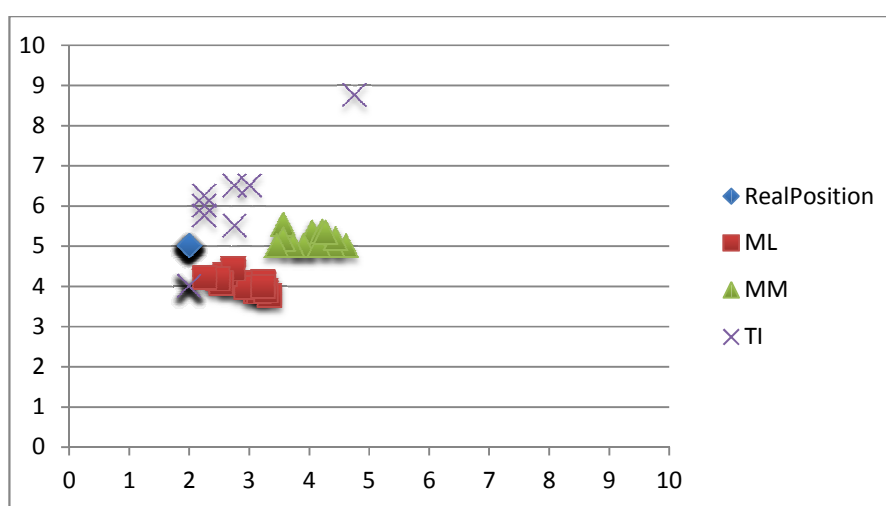
## ภาคผนวก ก

ตัวอย่างผลการทดลอง การหาตำแหน่งวัตถุแบบแสดงผลบน GUI Map และพิกัดตำแหน่ง  
ผลการทดลอง

## 1. ตำแหน่ง 2,5



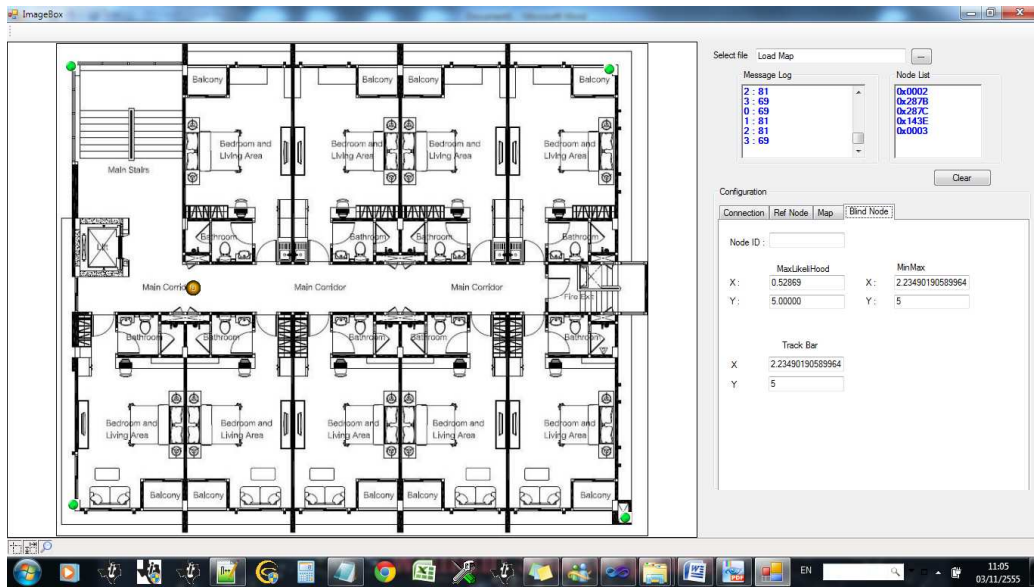
รูปที่ ก-1 แสดงตำแหน่งวัตถุ 2,5 แบบพิกัดและตำแหน่งบนแผนที่จำลอง



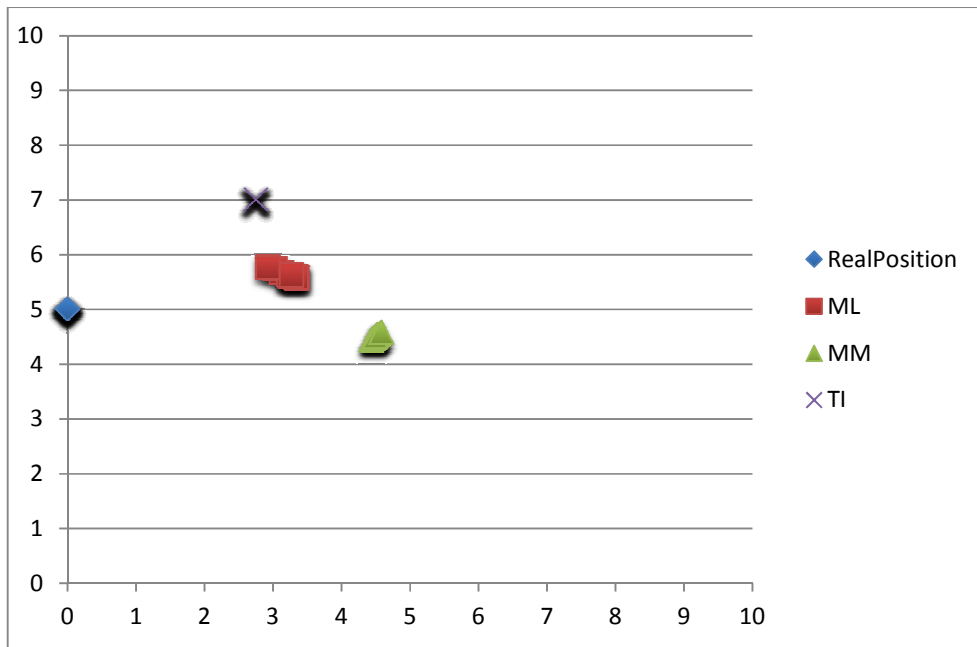
รูปที่ ก-2 แสดงตำแหน่งวัตถุ 2,5 แบบการกระจายของตำแหน่งด้วยค่าเบี่ยงเบนมาตรฐาน



2. ตำแหน่ง 0,5

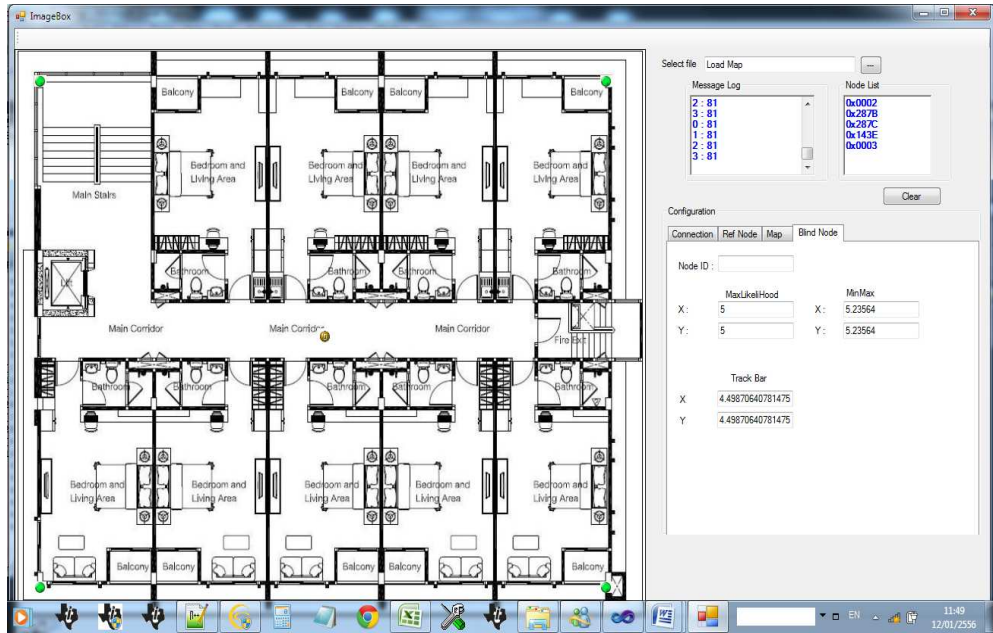


รูปที่ ก-3 แสดงตำแหน่งวัตถุ 0,5 แบบพิกัดและตำแหน่งบนแผนที่จำลอง

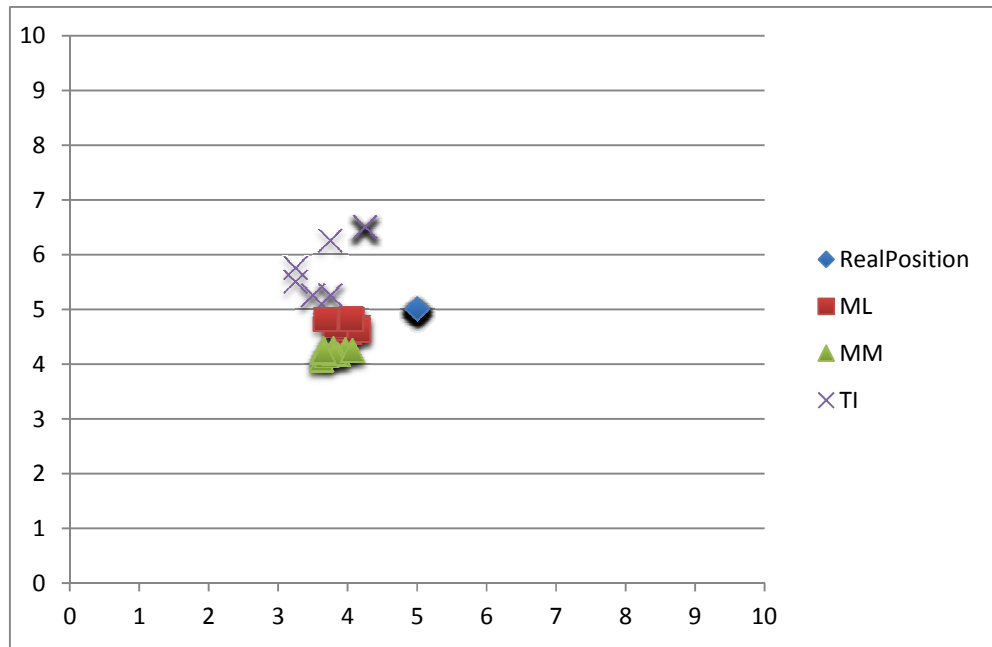


รูปที่ ก-4 แสดงตำแหน่งวัตถุ 0,5 แบบการกระจายของตำแหน่งด้วยค่าเบี่ยงเบนมาตรฐาน

3. ตำแหน่ง 5,5

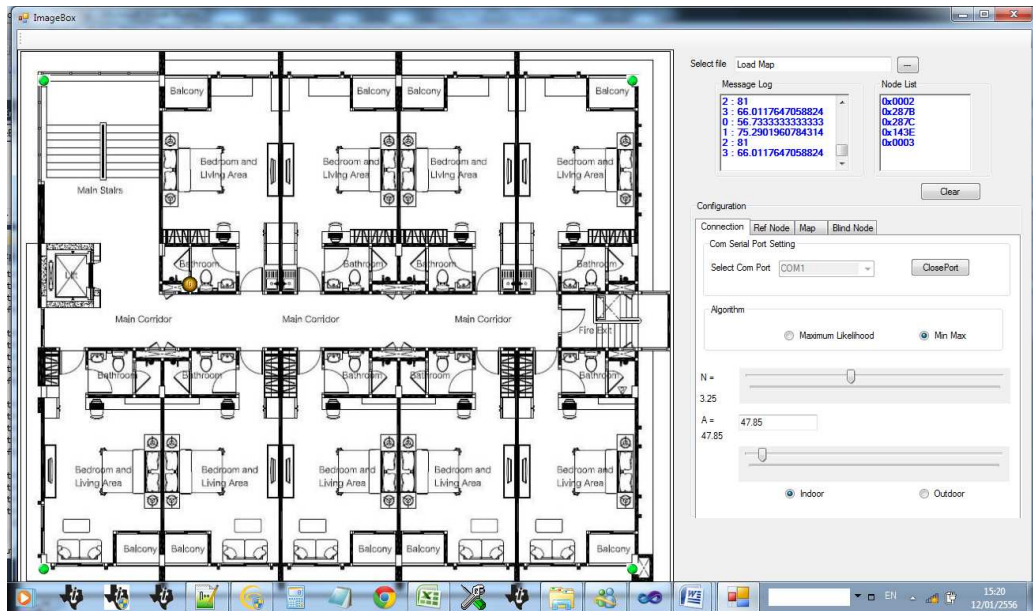


รูปที่ ก-5 แสดงตำแหน่งวัตถุ 5,5 แบบพิกัดและตำแหน่งบนแผนที่จำลอง

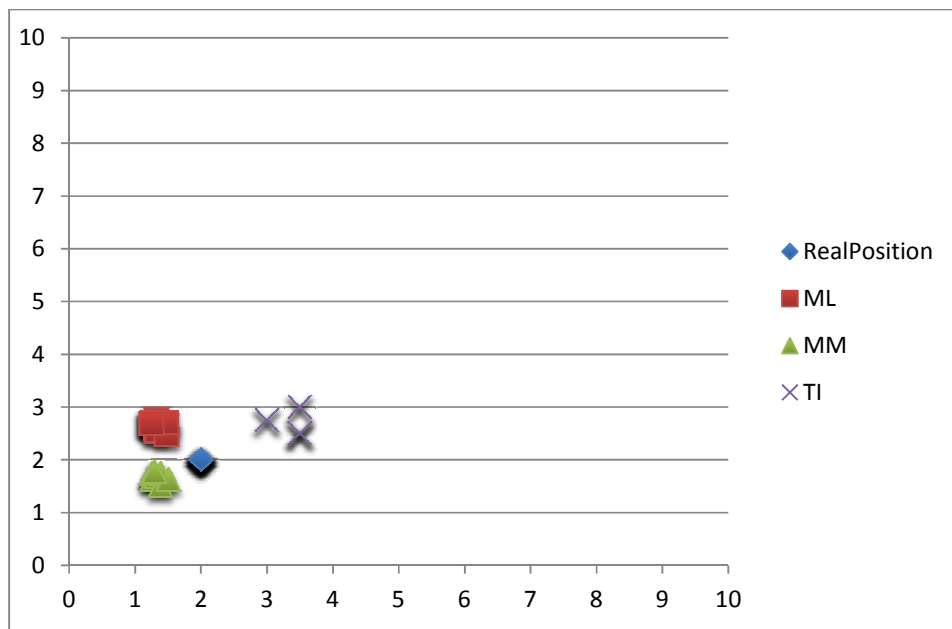


รูปที่ ก-6 แสดงตำแหน่งวัตถุ 5,5 แบบการกระจายของตำแหน่งด้วยค่าเบี่ยงเบนมาตรฐาน

4. ตำแหน่ง 2,2

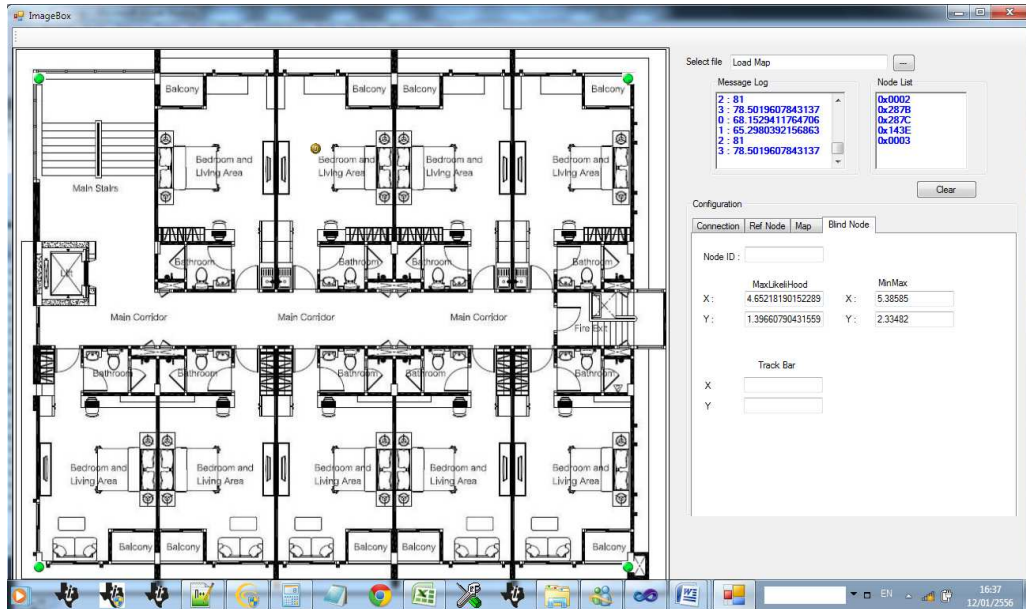


รูปที่ ก-7 แสดงตำแหน่งวัตถุ 2,2 แบบพิกัดและตำแหน่งบนแผนที่จำลอง

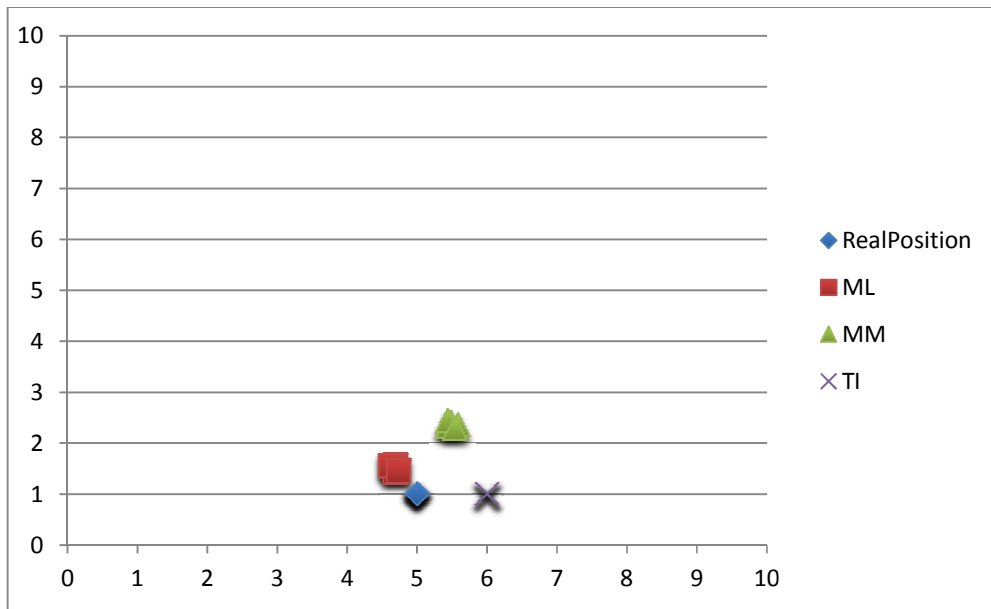


รูปที่ ก-8 แสดงตำแหน่งวัตถุ 2,2 แบบการกระจายของตำแหน่งด้วยค่าเบี่ยงเบนมาตรฐาน

5. ตำแหน่ง 5,1



รูปที่ ก-9 แสดงตำแหน่งวัตถุ 5,1 แบบพิกัดและตำแหน่งบนแผนที่จำลอง



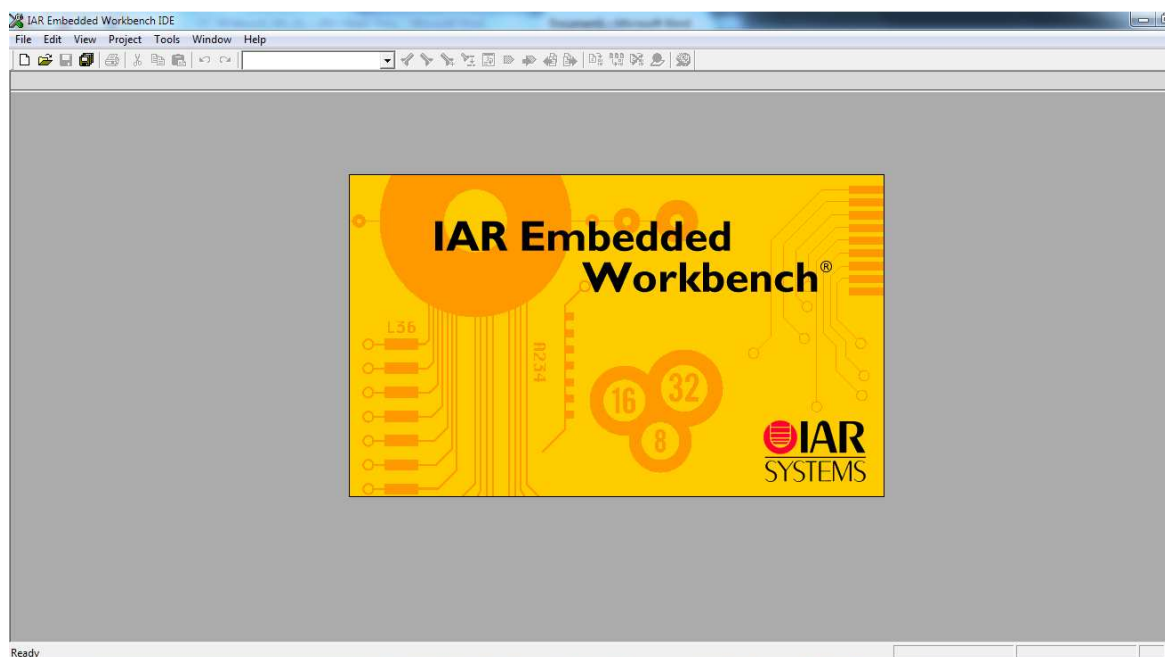
รูปที่ ก-10 แสดงตำแหน่งวัตถุ 2,2 แบบการกระจายของตำแหน่งด้วยค่าเบี่ยงเบนมาตรฐาน

## ภาคผนวก ข

แสดงคำสั่งการเขียนโปรแกรมคอมพิวเตอร์แบ่งเป็น 2 ส่วน คือ

1. ส่วนที่เป็น Source Code Embedded

- เข้า IAR Workbench Embedded



รูปที่ ข-1 แสดงการเข้าใช้โปรแกรม IAR Embedded Workbench

- แก้ไข Source code C++ ใหม่ของ RSSI Response ของส่วนที่เป็น Coordinator EB คือ Gateway

```
/******
```

```
* @fn    rssiRsp
```

```
*
```

```
* @brief Respond to requester with average of their RSSI blasts.
```

```
*
```

```
* @param uint16 - Network address of the Blind Node requesting.
```

```
*      byte  - Endpoint of the Blind Node requesting.
*      byte  - RSSI of the blast message received.
*
* @return none
*/
static void rssiRsp( afIncomingMSGPacket_t *pkt )
{
    blastAcc_t *ptr = blastPtr;
    blastAcc_t *prev = NULL;
    byte options, radius;

    while ( ptr )
    {
        if ( ptr->addr == pkt->srcAddr.addr.shortAddr )
        {
            break;
        }
        prev = ptr;
        ptr = ptr->next;
    }

    if ( ptr )
    {
        rspMsg[LOCATION_XY_RSSI_RSSI_IDX] =
            (ptr->acc + pkt->LinkQuality) / (ptr->cnt + 1);
    }
}
```

```

if ( prev )
{
    prev->next = ptr->next;
}
else
{
    blastPtr = ptr->next;
}
osal_mem_free( ptr );

options = AF_SKIP_ROUTING;
radius = 1;
}
else
{
// Comment
//  rspMsg[LOCATION_XY_RSSI_RSSI_IDX] = pkt->LinkQuality;
//  options = AF_TX_OPTIONS_NONE;
//  radius = AF_DEFAULT_RADIUS;

}

pkt->srcAddr.addrMode = afAddr16Bit;
(void)AF_DataRequest(      &pkt->srcAddr,      (endPointDesc_t
*)&epDesc,
                        LOCATION_XY_RSSI_RESPONSE,
LOCATION_XY_RSSI_LEN,

```

```
rspMsg, &transId, options, radius );
```

```
}
```

```
/******
```

```
******/
```

- หลังจากแก้ไขเสร็จเรียบร้อยแล้วเป็นขั้นตอนการสร้าง Hex file เพื่อเป็น Firmware ให้กับอุปกรณ์โดยแบ่งประเภทเป็น ดังนี้
  - ❖ Blind node คือ End Device BB
  - ❖ Reference node คือ Router BB
  - ❖ Gateway คือ Coordinator EB

โดยมีขั้นตอนดังนี้

1) โหมด Coordinator EB หรือ Gateway จะใช้ Chip CC2430

ให้เข้าไปแก้ไข ตามด้านล่าง

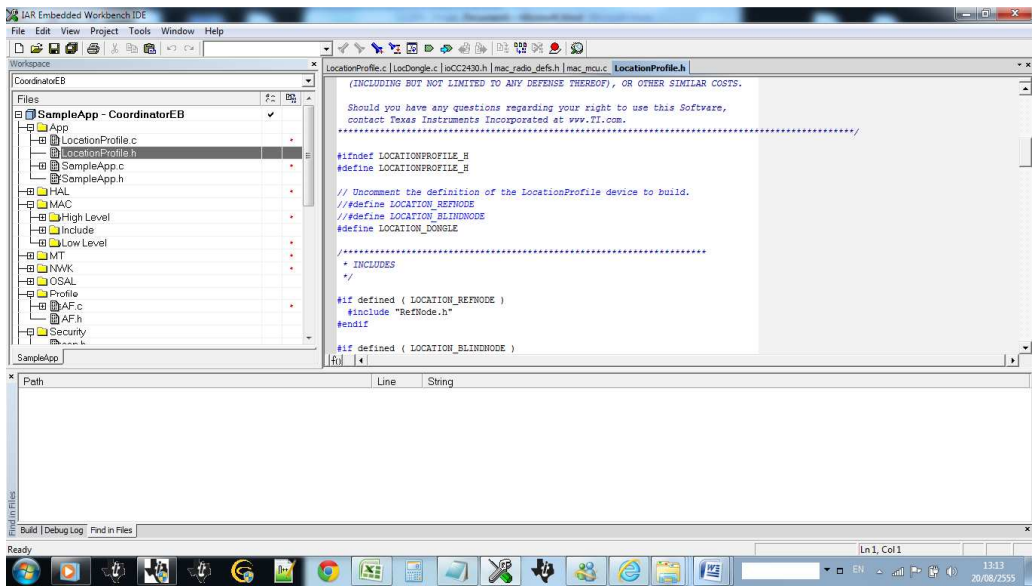
- Comment
 

```
//#define LOCATION_REFNODE
```

```
//#define LOCATION_BLINDNODE
```
- Uncomment
 

```
#define LOCATION_DONGLE
```

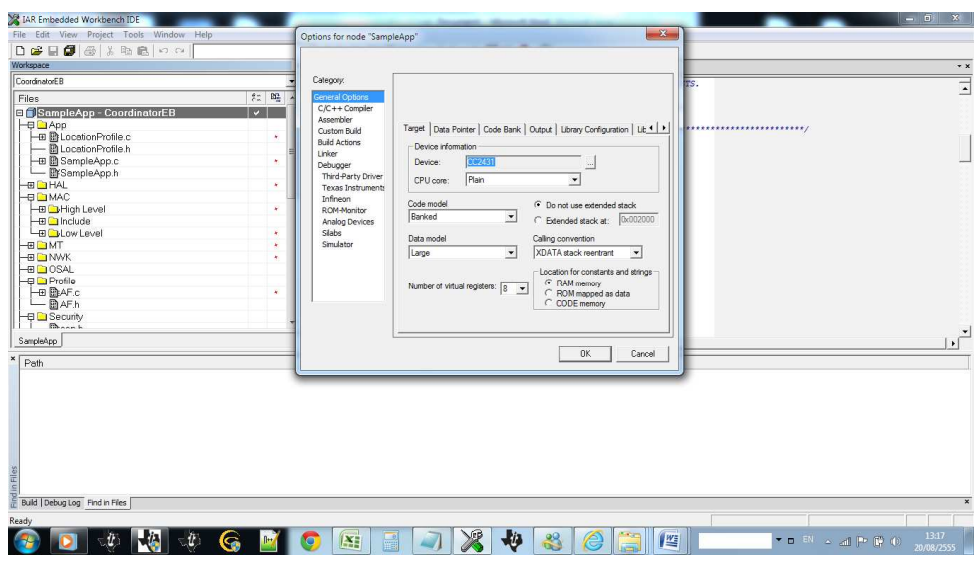




รูปที่ ข-2 แสดงการแก้ไขไฟล์ Configure ของ LocationProfile.h

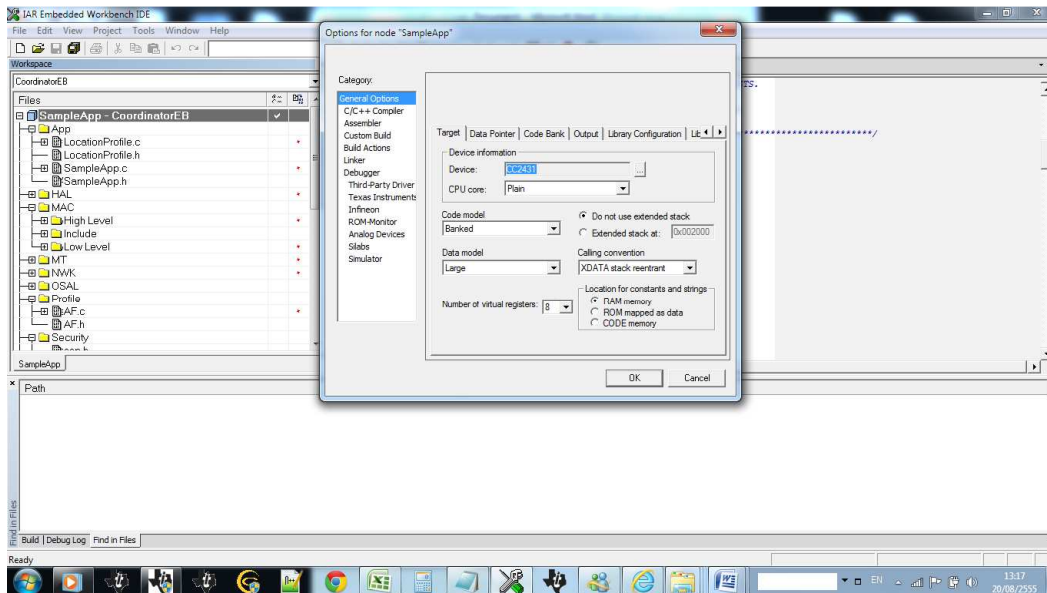
ต่อมาแก้ไข Configure ดังนี้

- คลิกขวา > SampleApp-CoordinatorEB เลือก Option
- ไปที่ Tab General Option เลือก Device = CC2430 และรายละเอียดตามภาพ



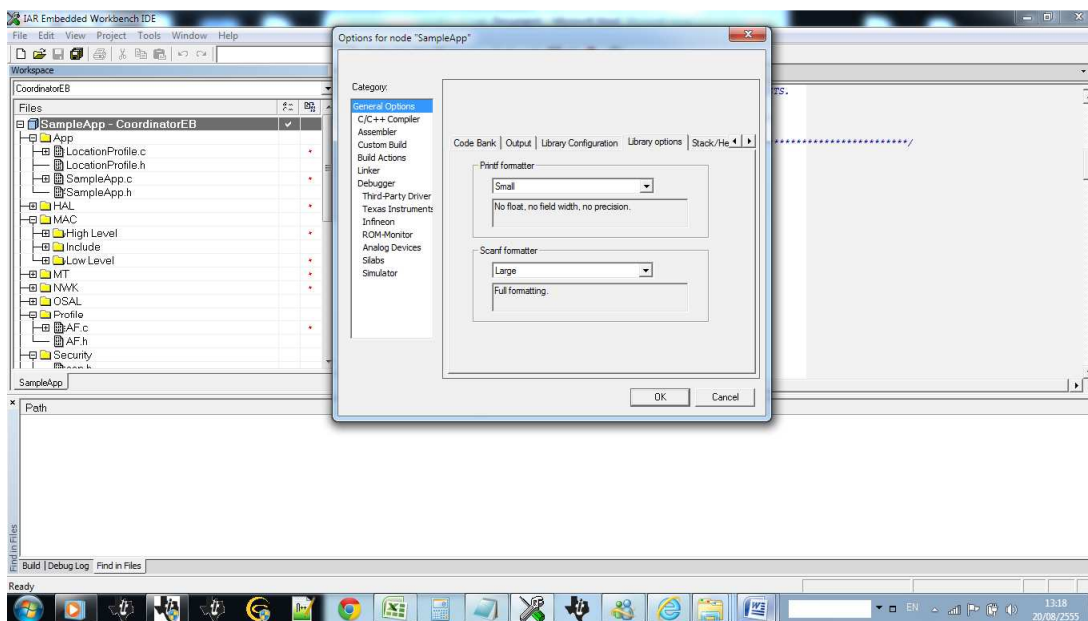
รูปที่ ข-3 แสดงการแก้ไขไฟล์ Configure ของ SampleApp-CoordinatorEB Part :Code model

- ไปที่ Tab Library : C onfiguration



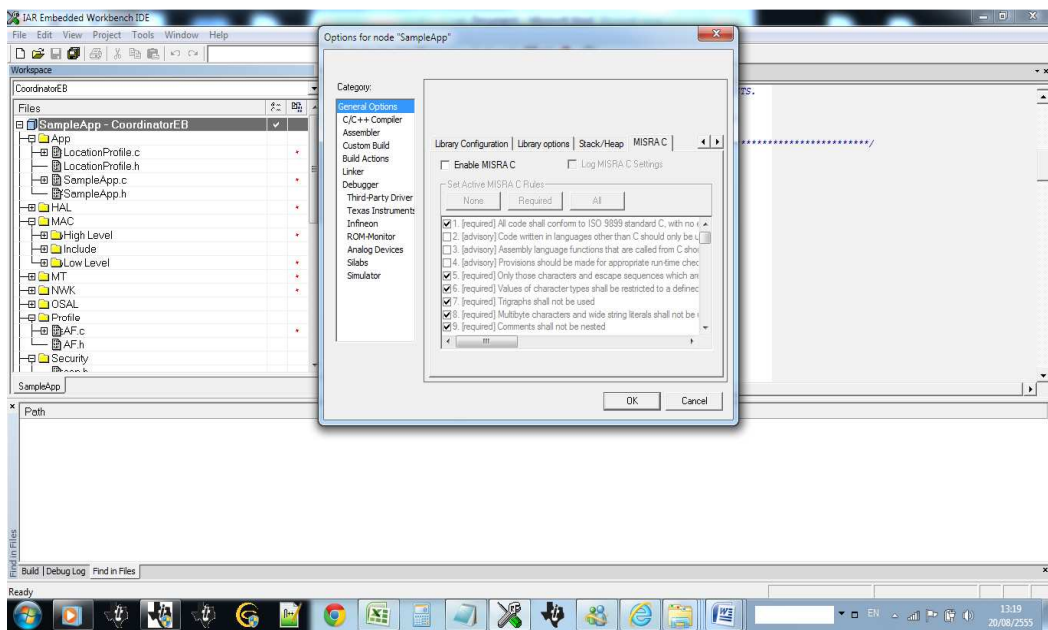
รูปที่ ข-4 แสดงการแก้ไขไฟล์ Configure ของ SampleApp-CoordinatorEB Part:Data Model

- เลือก Tab Library Option



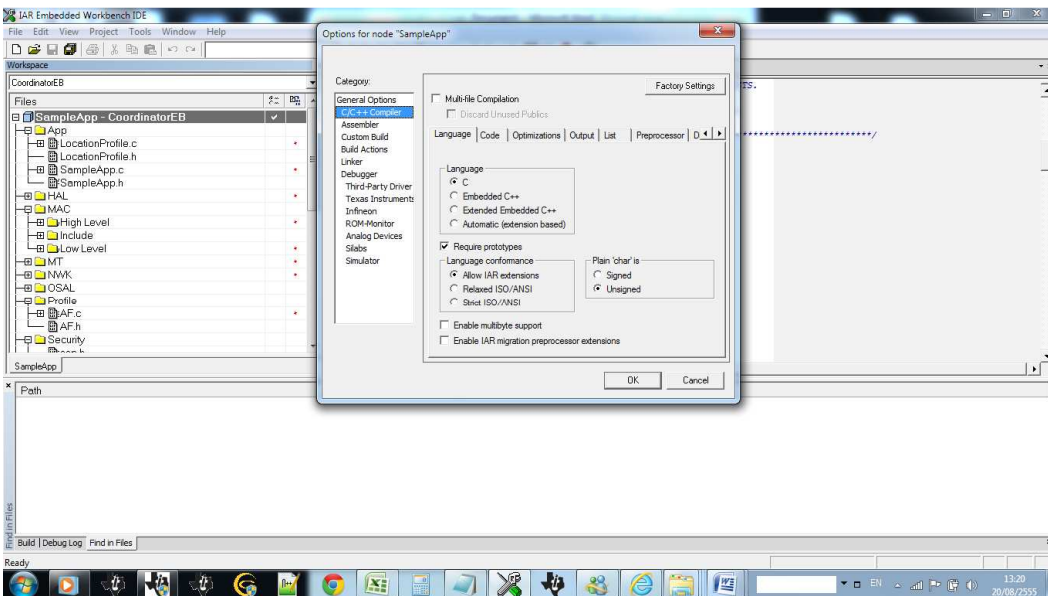
รูปที่ ข-5 แสดงการแก้ไขไฟล์ Configure ของ SampleApp-CoordinatorEB เลือก Format

- ไปที่ Tab MISARAC



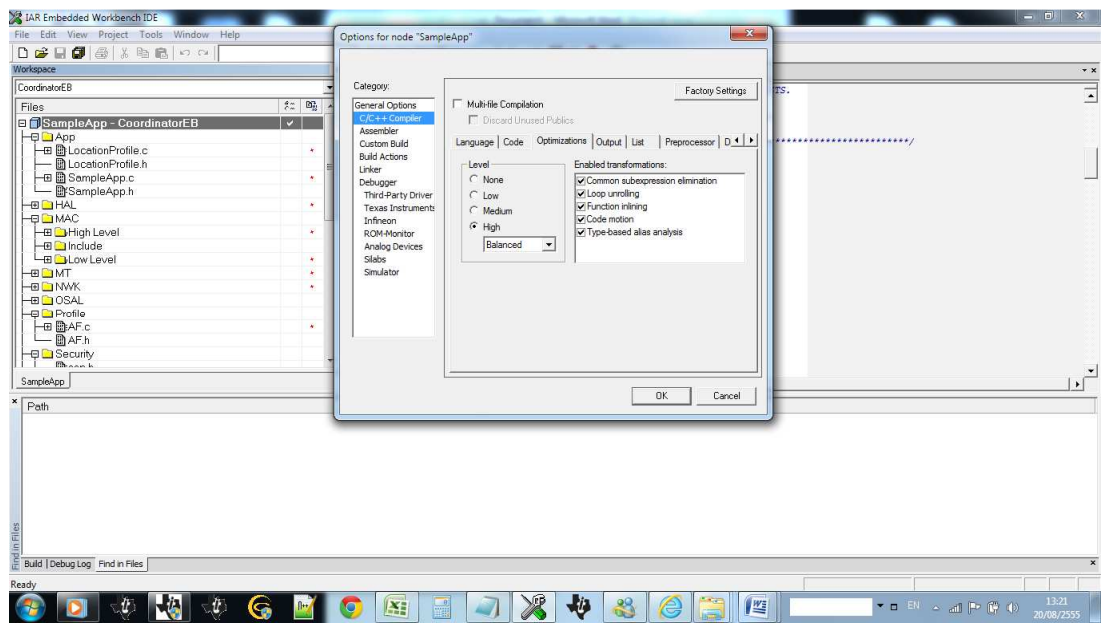
รูปที่ ข-6 แสดงการแก้ไขไฟล์หมวด MISARAC

- เลือกชนิด Category : C/C++ Compiler ตามรุ่น



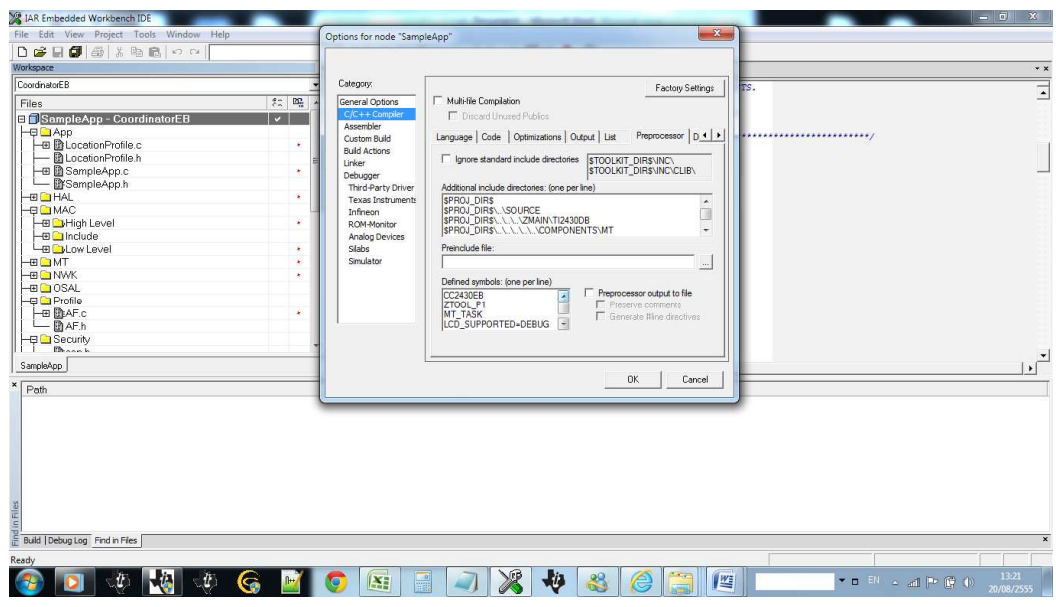
รูปที่ ข-7 แสดงการแก้ไขไฟล์หมวด Language

• ไปที่ Tab Optimization



รูปที่ ข-8 แสดงการแก้ไขไฟล์หมวด C/C++ Compiler

• ไปที่ Tab Preprocessors



รูปที่ ข-9 แสดงการ Add DIR ใน Additional include directories

- ให้ทำการ Add directory ดังนี้

\$PROJ\_DIR\$

\$PROJ\_DIR\$\..\SOURCE

\$PROJ\_DIR\$\..\..\ZMAIN\TI2430DB

\$PROJ\_DIR\$\..\..\..\COMPONENTS\MT

\$PROJ\_DIR\$\..\..\..\COMPONENTS\HAL\INCLUDE

\$PROJ\_DIR\$\..\..\..\COMPONENTS\HAL\TARGET\CC2430EB

\$PROJ\_DIR\$\..\..\..\COMPONENTS\OSAL\MCU\CCSOC

\$PROJ\_DIR\$\..\..\..\COMPONENTS\OSAL\INCLUDE

\$PROJ\_DIR\$\..\..\..\COMPONENTS\STACK\AF

\$PROJ\_DIR\$\..\..\..\COMPONENTS\STACK\NWK

\$PROJ\_DIR\$\..\..\..\COMPONENTS\STACK\SEC

\$PROJ\_DIR\$\..\..\..\COMPONENTS\STACK\SYS

\$PROJ\_DIR\$\..\..\..\COMPONENTS\STACK\ZDO

\$PROJ\_DIR\$\..\..\..\COMPONENTS\ZMAC\F8W

\$PROJ\_DIR\$\..\..\..\COMPONENTS\ZMAC

\$PROJ\_DIR\$\..\..\..\COMPONENTS\SERVICES\SADDR

\$PROJ\_DIR\$\..\..\..\COMPONENTS\SERVICES\SDATA

\$PROJ\_DIR\$\..\..\..\COMPONENTS\MAC\INCLUDE

\$PROJ\_DIR\$\..\..\..\COMPONENTS\MAC\HIGH\_LEVEL

\$PROJ\_DIR\$\..\..\..\COMPONENTS\MAC\LOW\_LEVEL\SRF03

\$PROJ\_DIR\$\..\..\..\COMPONENTS\MAC\LOW\_LEVEL\SRF03\SINGLE\_CHIP

Defined Symbol :

CC2430EB

ZTOOL\_P1

MT\_TASK

LCD\_SUPPORTED=DEBUG

NV\_RESTORE

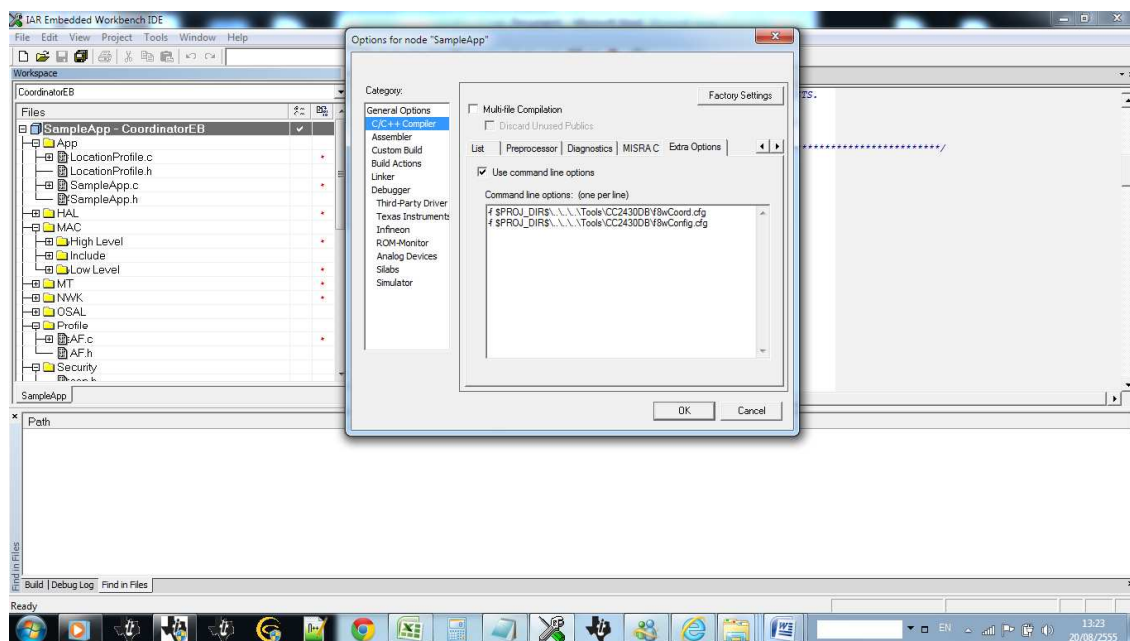
INT\_HEAP\_LEN=4608

HAL\_UART\_BIG\_TX\_BUF=1

SPI\_MGR\_DEFAULT\_MAX\_TX\_BUFF=1024

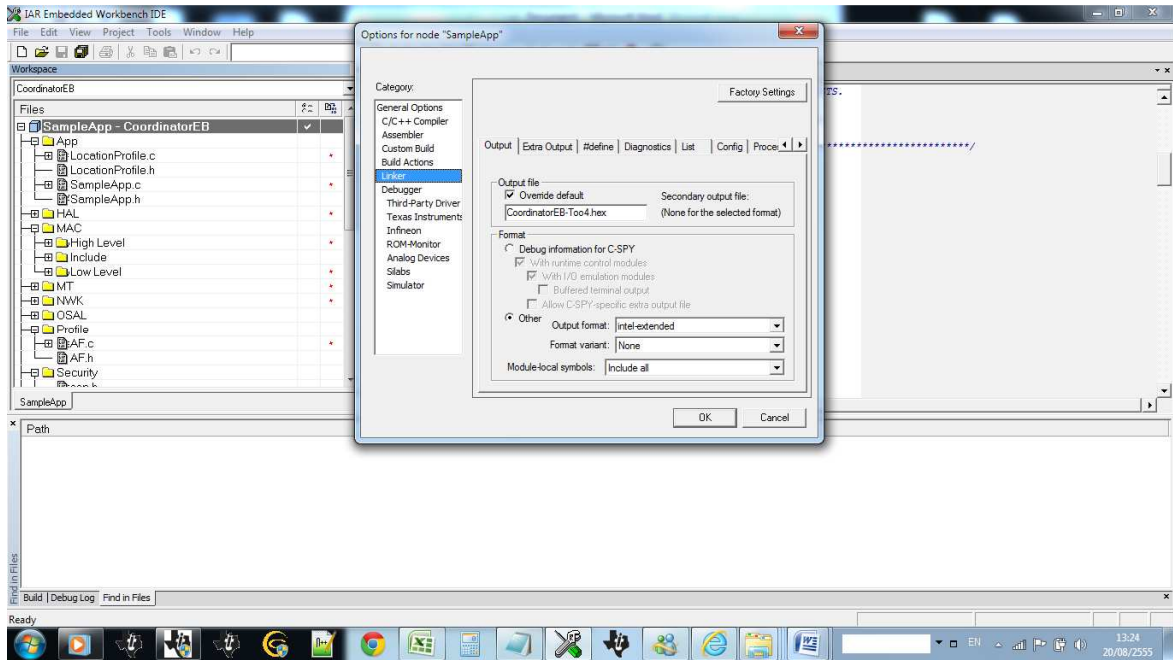
SPI\_MGR\_DEFAULT\_OVERFLOW=1

- ไปที่ Tab Extra Option



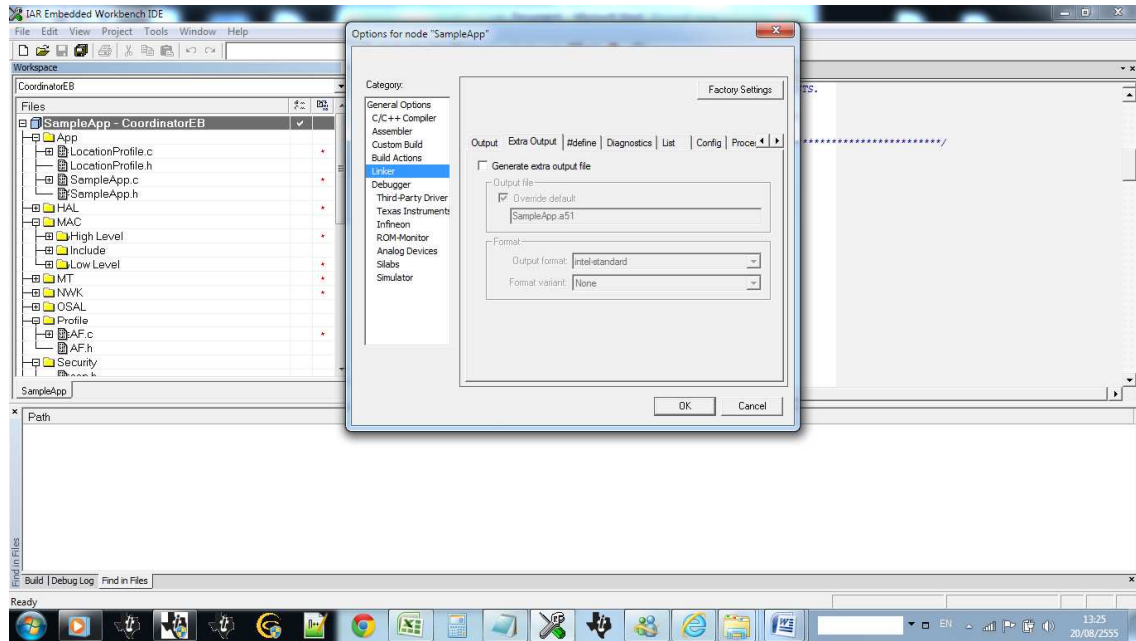
รูปที่ ข-10 แสดงการแก้ไขไฟล์หมวด Language

- ไปที่ Tab Linker



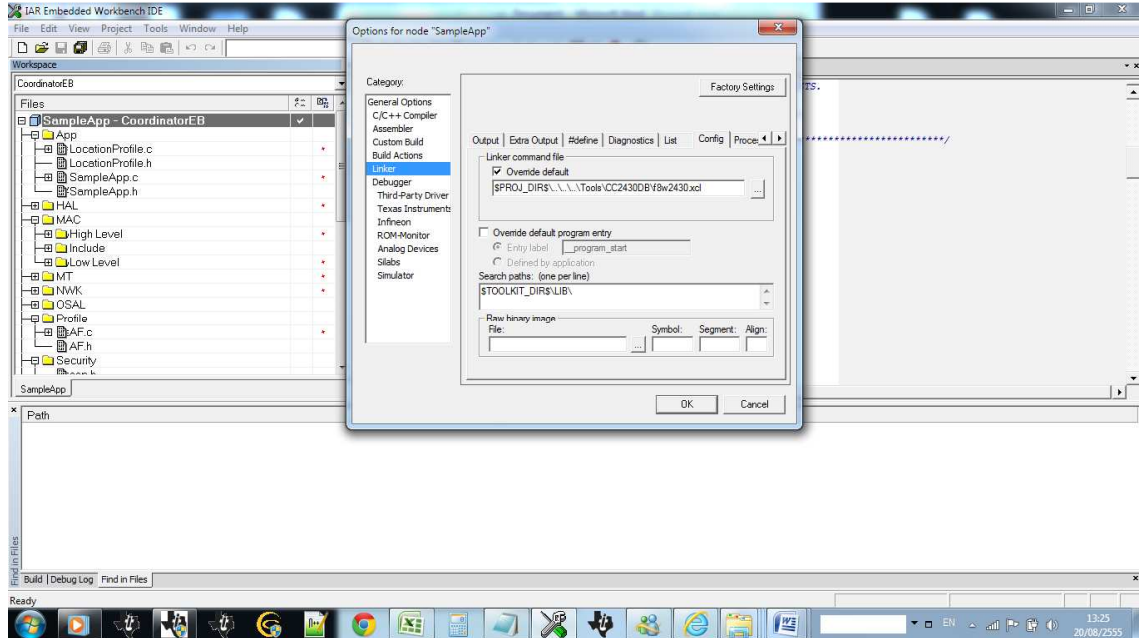
รูปที่ ๑-11 แสดง TYPE output file

- ไปที่ Tab Extra Output



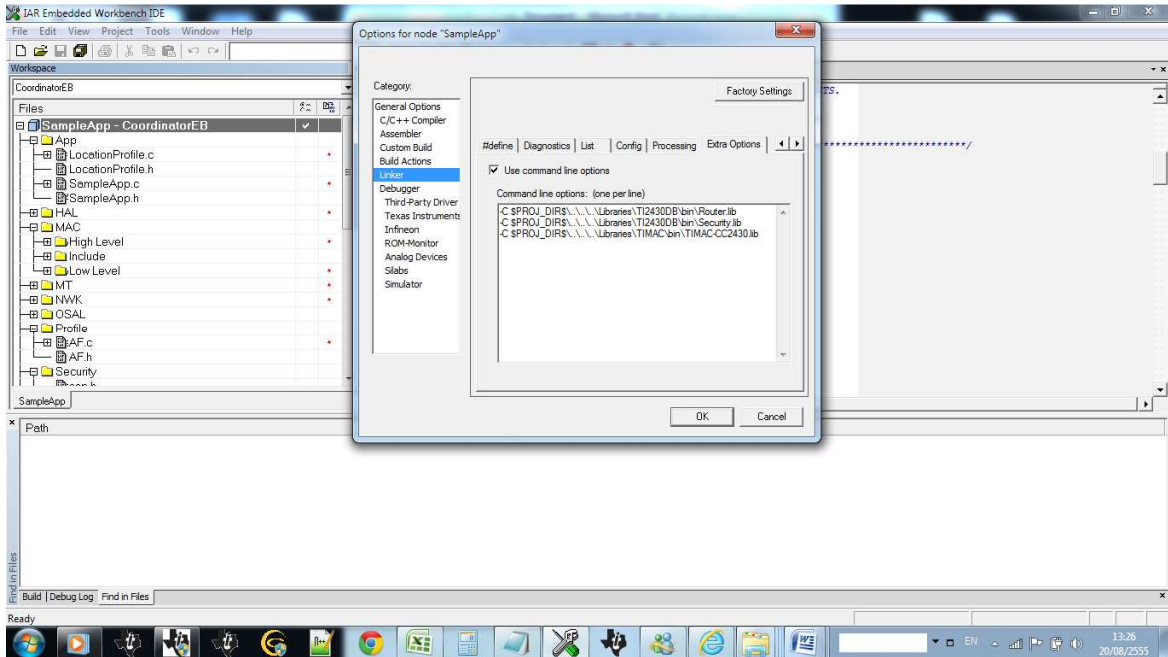
รูปที่ ๑-12 แสดงการ Add Linker

- ไปที่ Tab Config



รูปที่ ข-13 แสดงการ Add Linker command file และ Search file

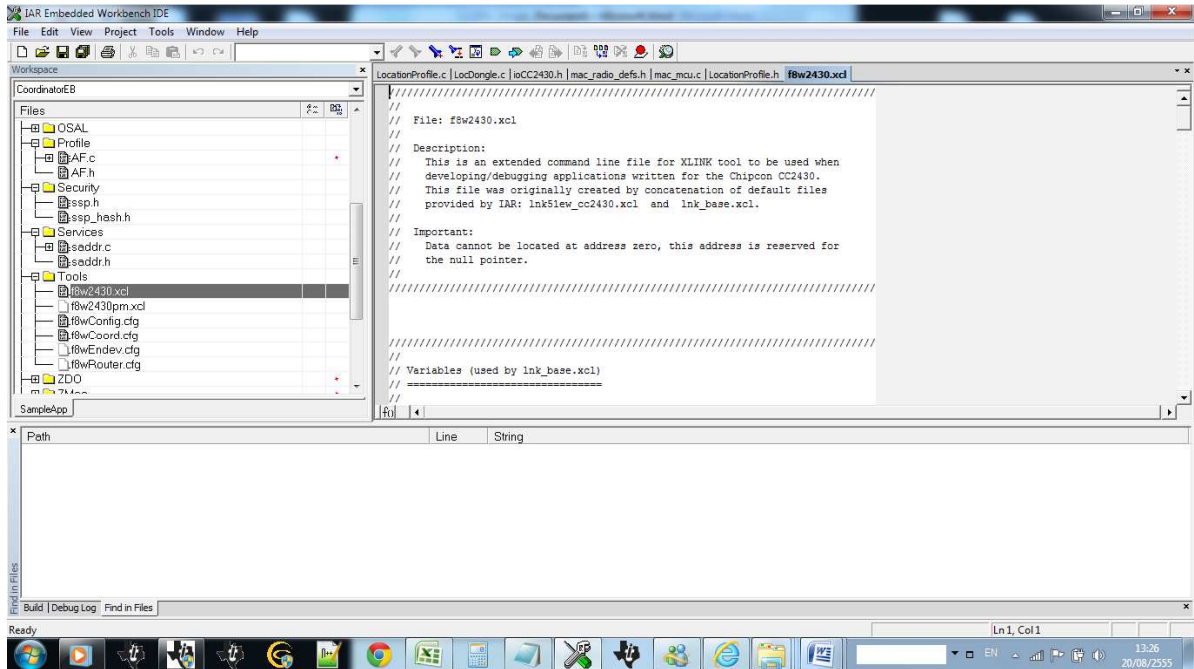
- ไปที่ Tab Extra Option



รูปที่ ข-14 แสดงการ Add Linker ใน Command Line



- กลับไปที่ เมนูซ้ายมือเลือก f8w2430.xcl

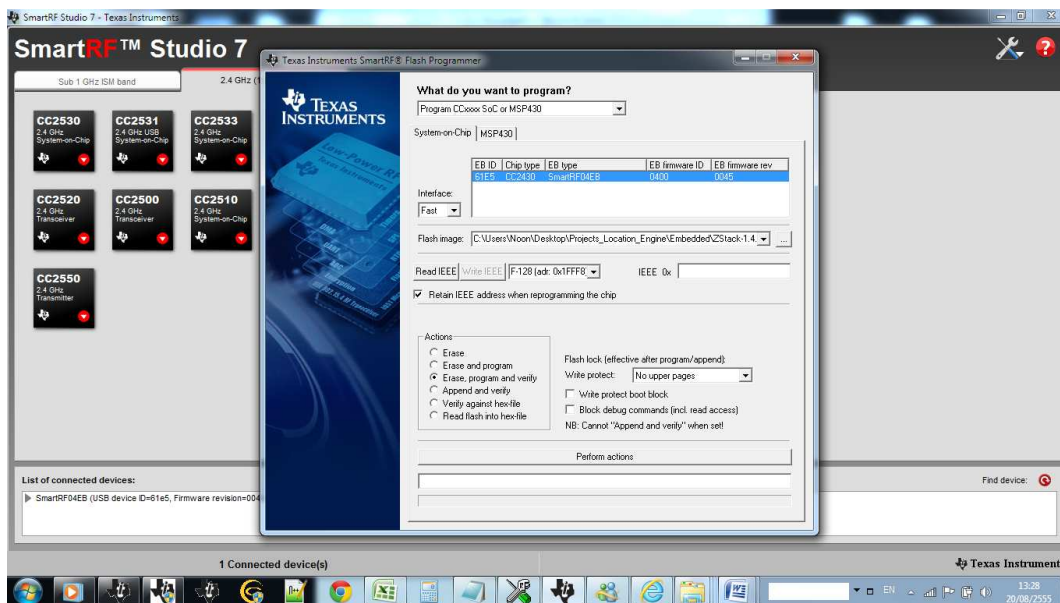


รูปที่ ๗-15 แสดงการ Add Tools : f8W2430.xcl

-M ( เอา uncoment ออก)

การ Burn Firmware ลง Chipcon

- ไปที่ Application Smart RF Studio 7



รูปที่ ข-16 แสดงโหมดโปรแกรม Smart RF Studio 7

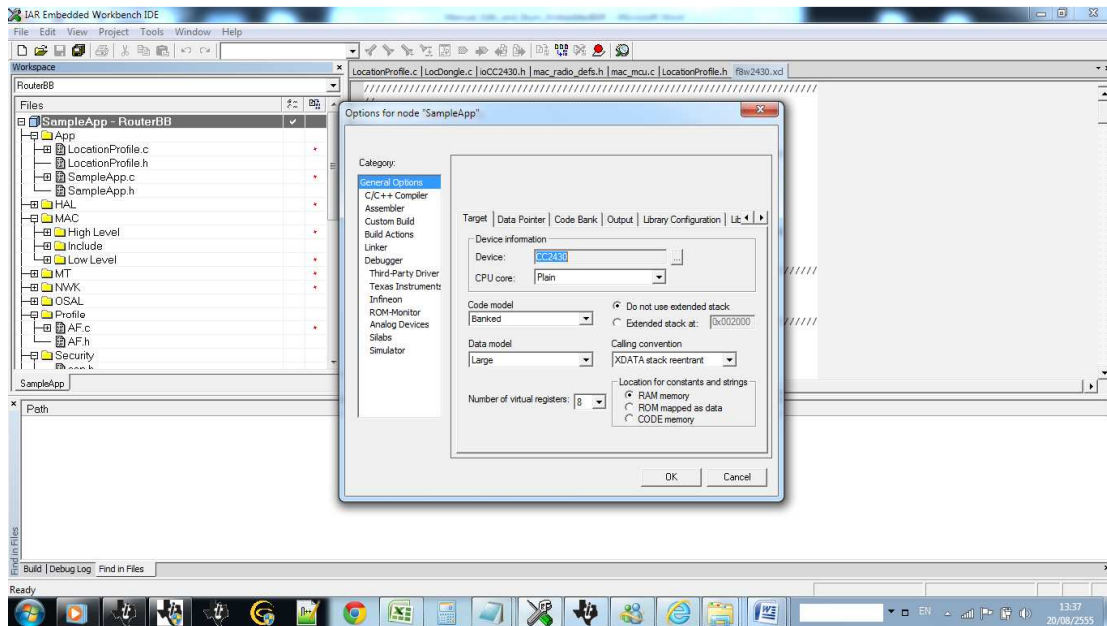
- เลือก Retain IIEEE ออก
- กด Read IIEEE
- Browse :  
C:\Users\Noon\Desktop\Projects\_Location\_Engine\Embedded\ZStack-1.4.3-1.2.1\Projects\zstack\Samples\Location\CC2430DB\CoordinatorEB\Exe\CoordinatorEB-Too3.hex
- กด Perform Action
- กด Write IIEEE
- Success Output

2. โหมด RouterBB หรือ Reference node

- Comment  
// #define LOCATION\_DONGLE

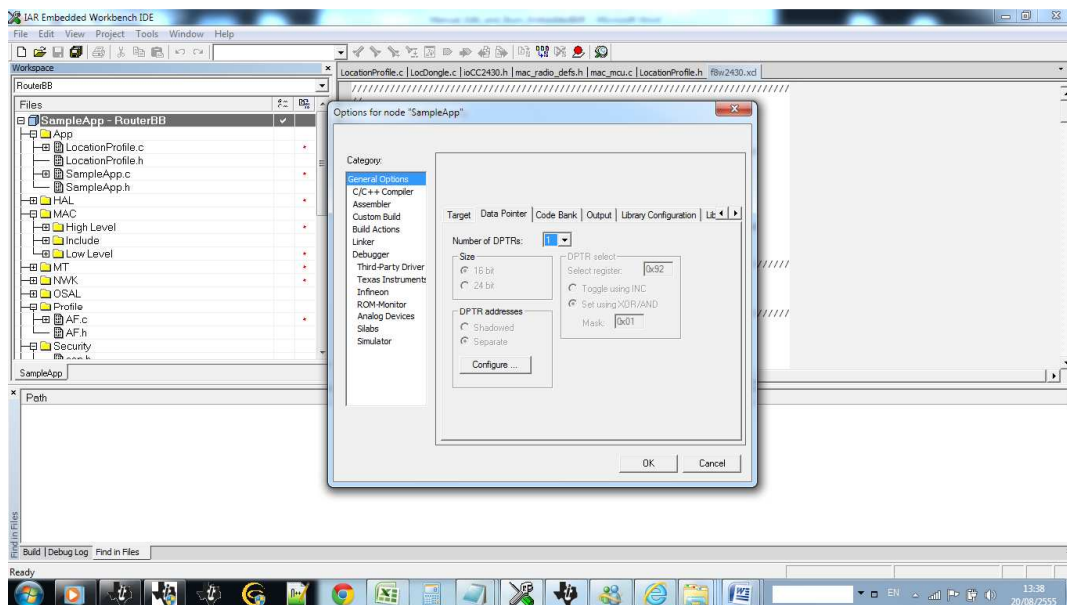
```
//#define LOCATION_BLINDNODE
```

- Uncomment : #define LOCATION\_REFNODE
- ไปที่ Tab General Options



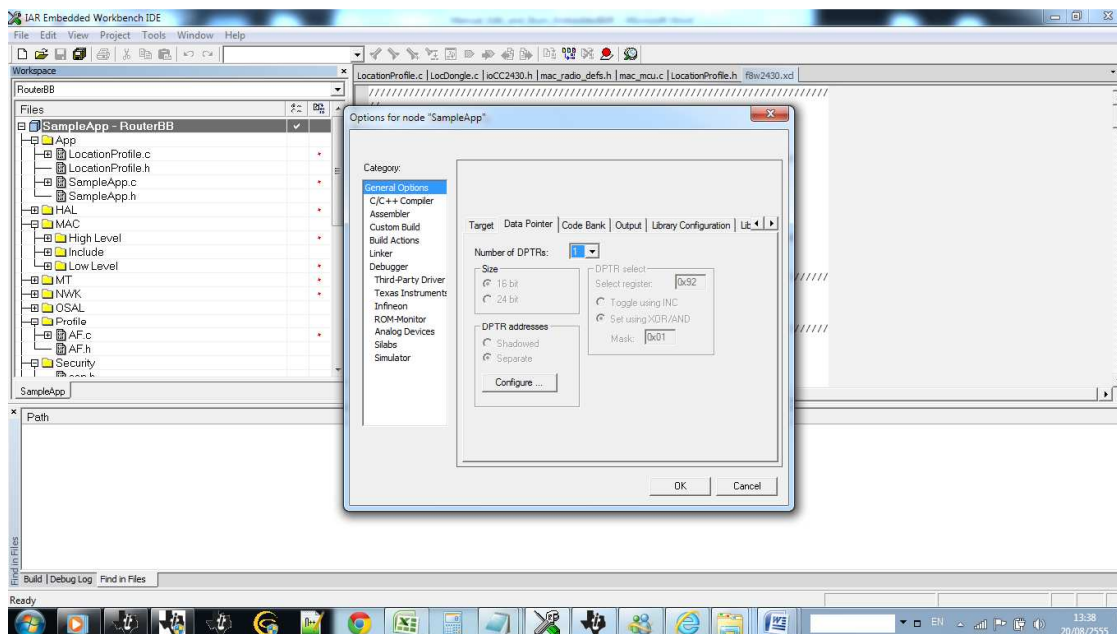
รูปที่ ๗-17 แสดงการตั้งค่า Tab General

- ไปที่ Tab Data Pointer



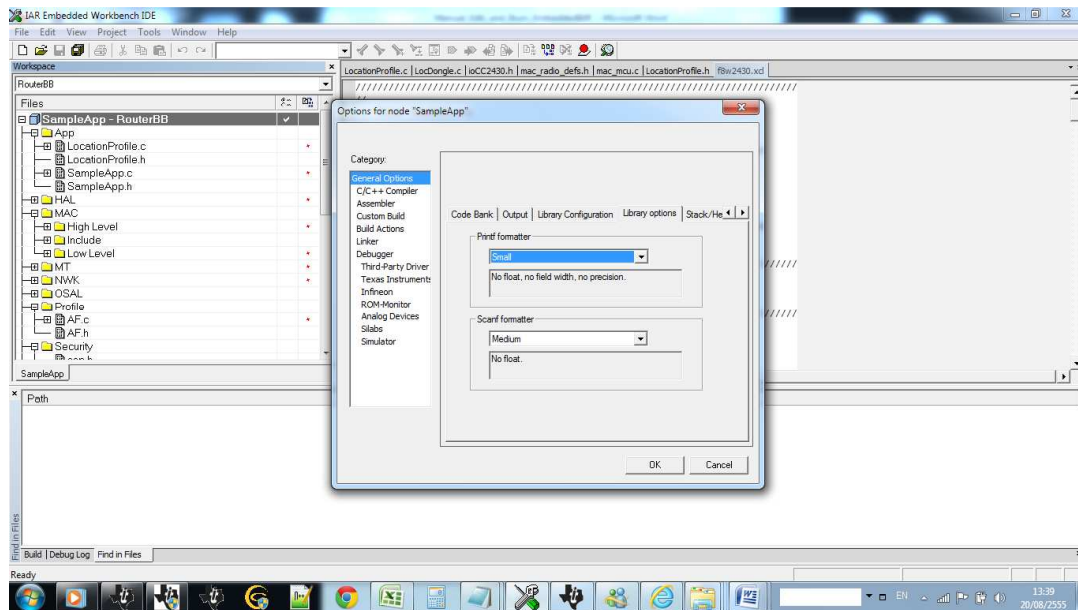
รูปที่ ข-18 แสดงการตั้งค่า Tab Data Pointer

- ไปแก้ไขใน Library Configuration



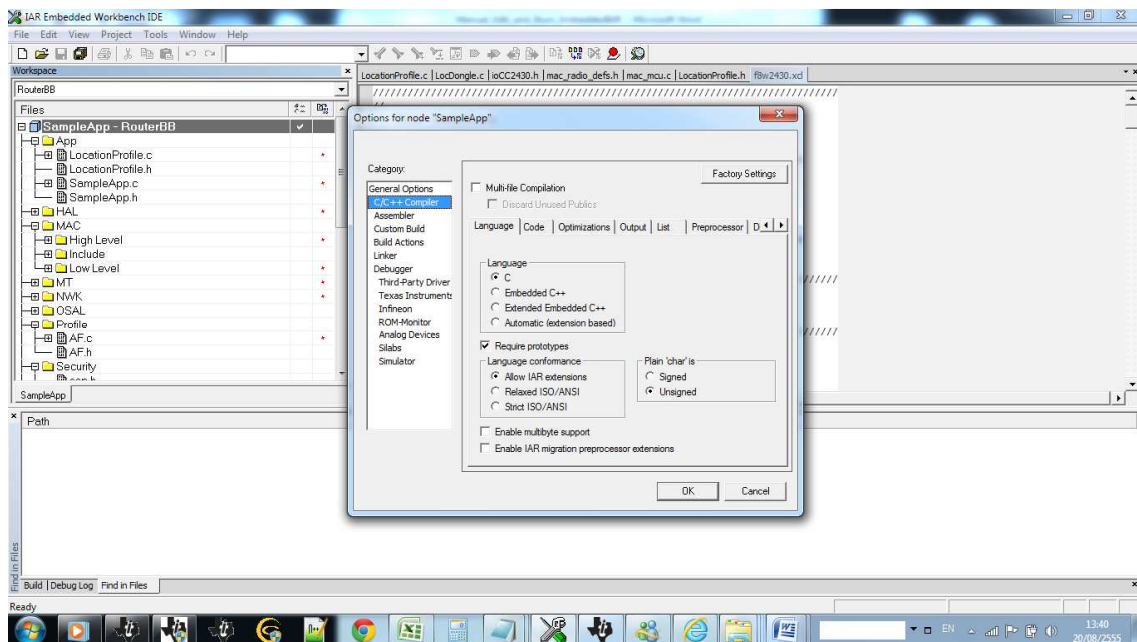
รูปที่ ข-19 แสดงการตั้งค่า Tab Library Configuration

- ไปที่ Library Option



รูปที่ ข-20 แสดงการตั้งค่า Tab Library Option

- ไปที่ Tab C/C++ Compiler



รูปที่ ข-21 แสดงการตั้งค่า Tab C/C++ Compiler



\$PROJ\_DIR\$

\$PROJ\_DIR\$\..\SOURCE

\$PROJ\_DIR\$\..\..\ZMAIN\TI2430DB

\$PROJ\_DIR\$\..\..\..\COMPONENTS\MT

\$PROJ\_DIR\$\..\..\..\COMPONENTS\HAL\INCLUDE

\$PROJ\_DIR\$\..\..\..\COMPONENTS\HAL\TARGET\CC2430BB

\$PROJ\_DIR\$\..\..\..\COMPONENTS\OSAL\MCU\CCSOC

\$PROJ\_DIR\$\..\..\..\COMPONENTS\OSAL\INCLUDE

\$PROJ\_DIR\$\..\..\..\COMPONENTS\STACK\AF

\$PROJ\_DIR\$\..\..\..\COMPONENTS\STACK\NWK

\$PROJ\_DIR\$\..\..\..\COMPONENTS\STACK\SEC

\$PROJ\_DIR\$\..\..\..\COMPONENTS\STACK\SYS

\$PROJ\_DIR\$\..\..\..\COMPONENTS\STACK\ZDO

\$PROJ\_DIR\$\..\..\..\COMPONENTS\ZMAC\F8W

\$PROJ\_DIR\$\..\..\..\COMPONENTS\ZMAC

\$PROJ\_DIR\$\..\..\..\COMPONENTS\SERVICES\SADDR

\$PROJ\_DIR\$\..\..\..\COMPONENTS\SERVICES\SDATA

\$PROJ\_DIR\$\..\..\..\..\COMPONENTS\MAC\INCLUDE

\$PROJ\_DIR\$\..\..\..\..\COMPONENTS\MAC\HIGH\_LEVEL

\$PROJ\_DIR\$\..\..\..\..\COMPONENTS\MAC\LOW\_LEVEL\SRF03

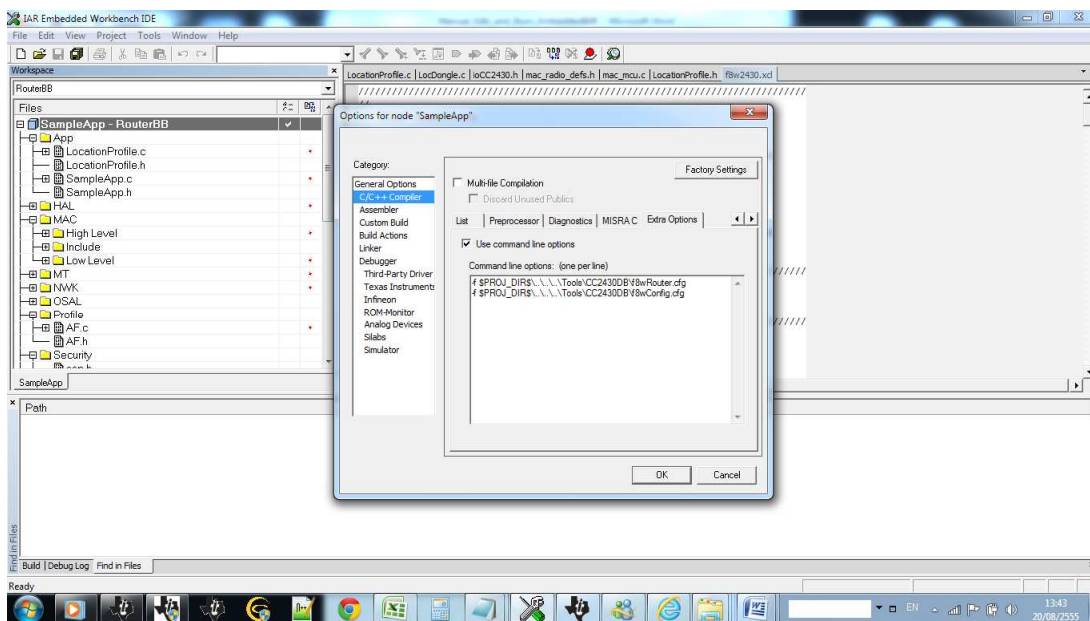
\$PROJ\_DIR\$\..\..\..\..\COMPONENTS\MAC\LOW\_LEVEL\SRF03\SINGLE\_CHIP

Defined Symbols :

CC2430BB

NV\_RESTORE

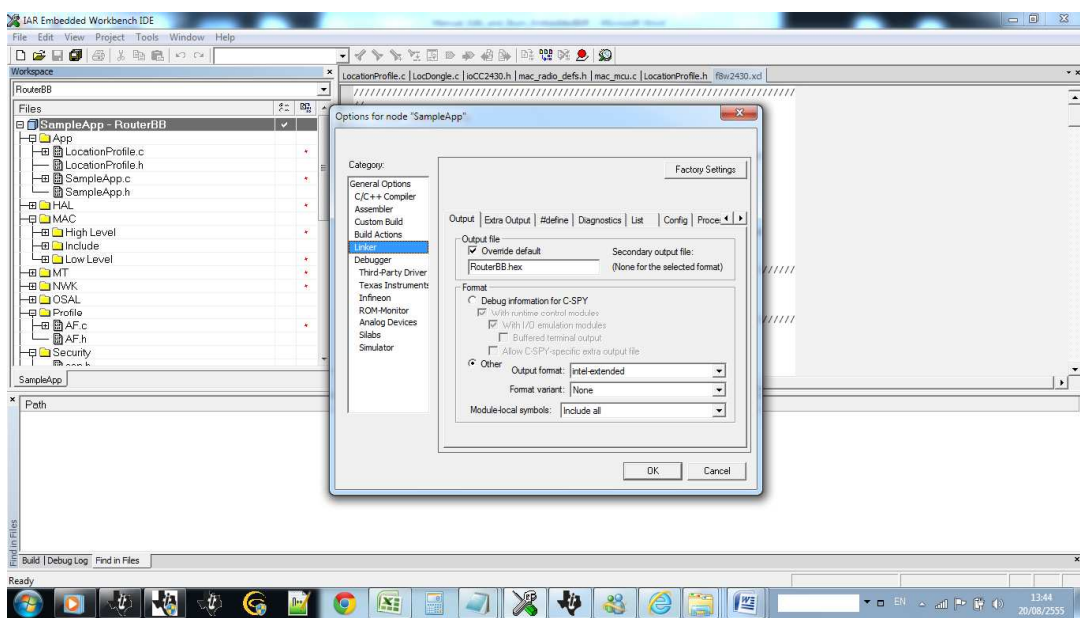
- ไปที่ Extra Option



รูปที่ ข-24 แสดงการตั้งค่า Tab Extra Option

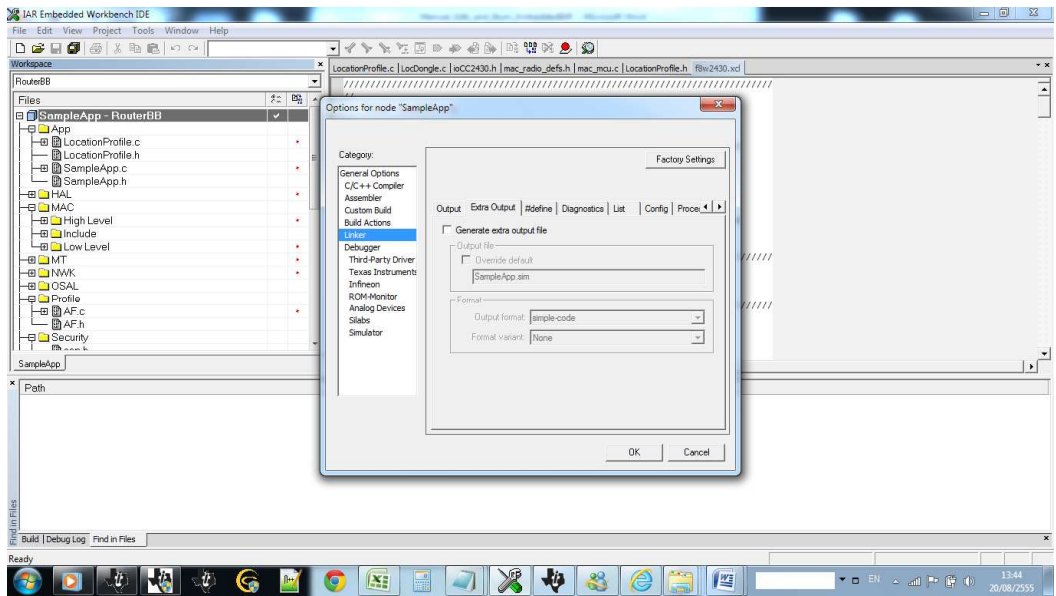


- ไปที่ Tab Linker



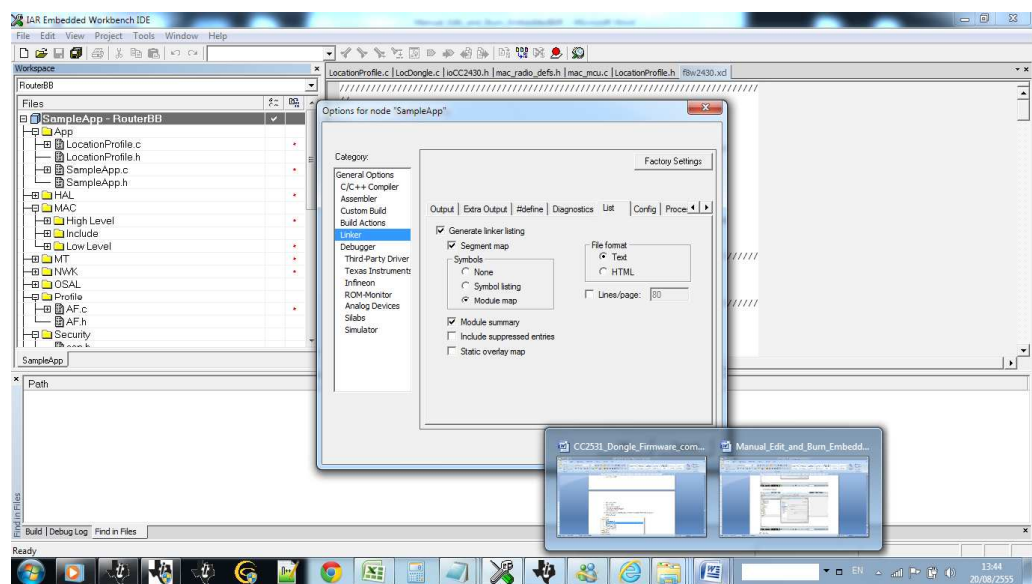
รูปที่ ข-25 แสดงการตั้งค่า Tab Linker

- ไปที่ Tab Extra Output



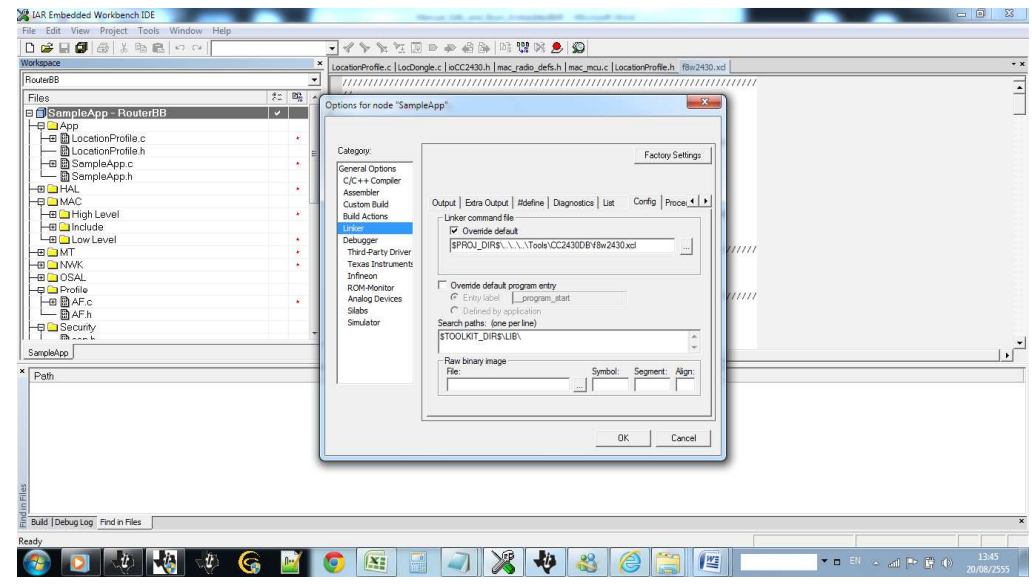
รูปที่ ข-26 แสดงการตั้งค่า Extra Output

- ไปที่ Tab List



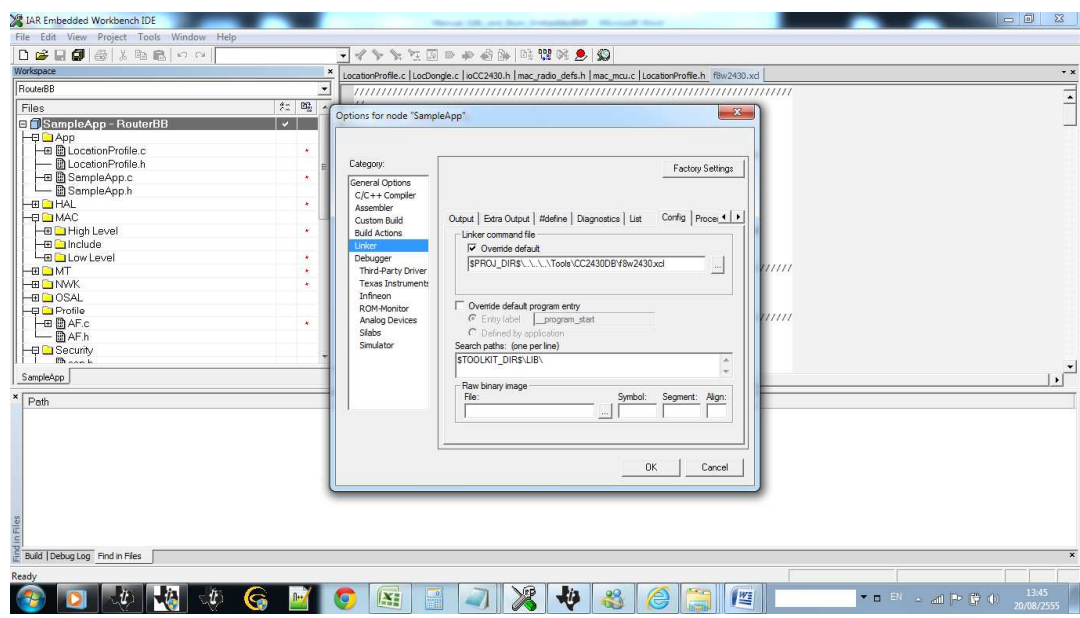
รูปที่ ข-27 แสดงการตั้งค่า Tab List

- ไปที่ Tab Config



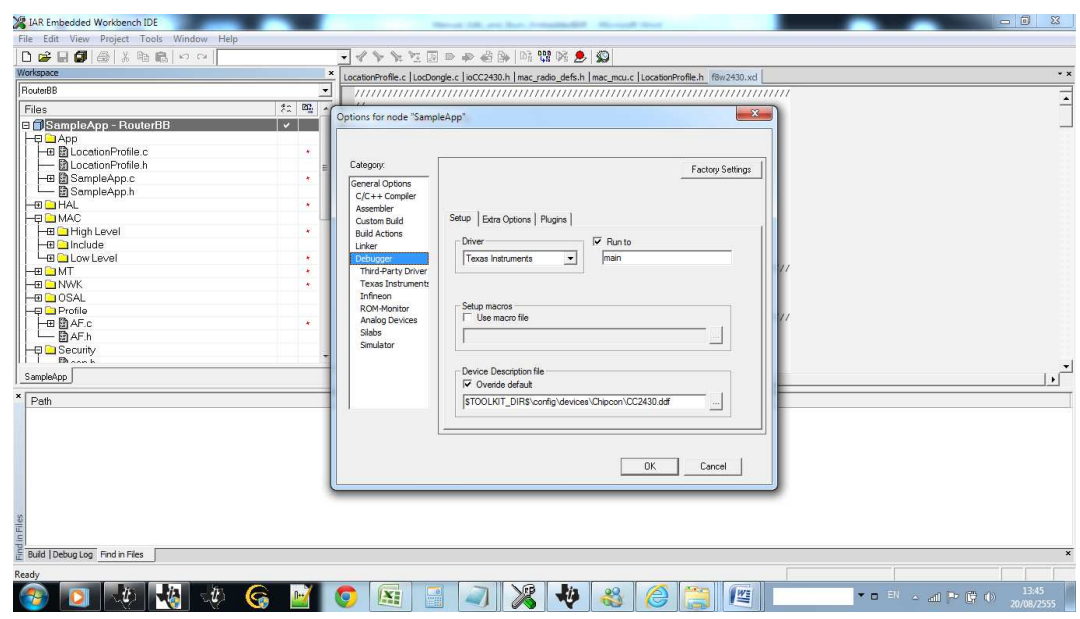
รูปที่ ข-28 แสดงการตั้งค่า Tab Config

- ไปที่ Tab Extra Options



รูปที่ ข-29 แสดงการตั้งค่า Tab Extra Option

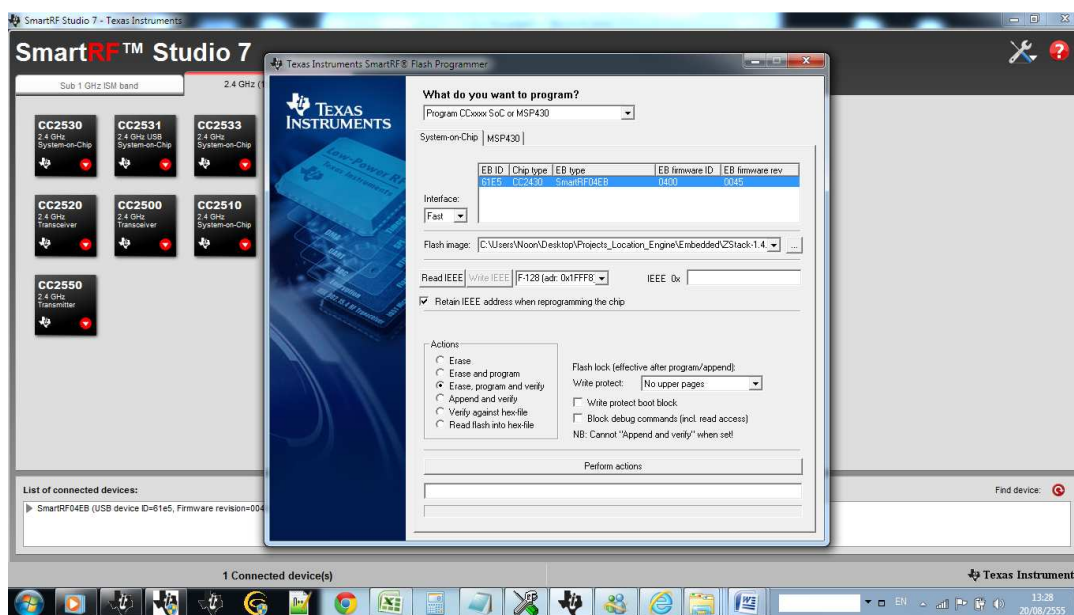
- ไปที่ Tab Debugger



รูปที่ ข-30 แสดงการตั้งค่า Tab Debugger

การ Burn Firmware ลง Chipcon

- ไปที่ Application Smart RF Studio 7

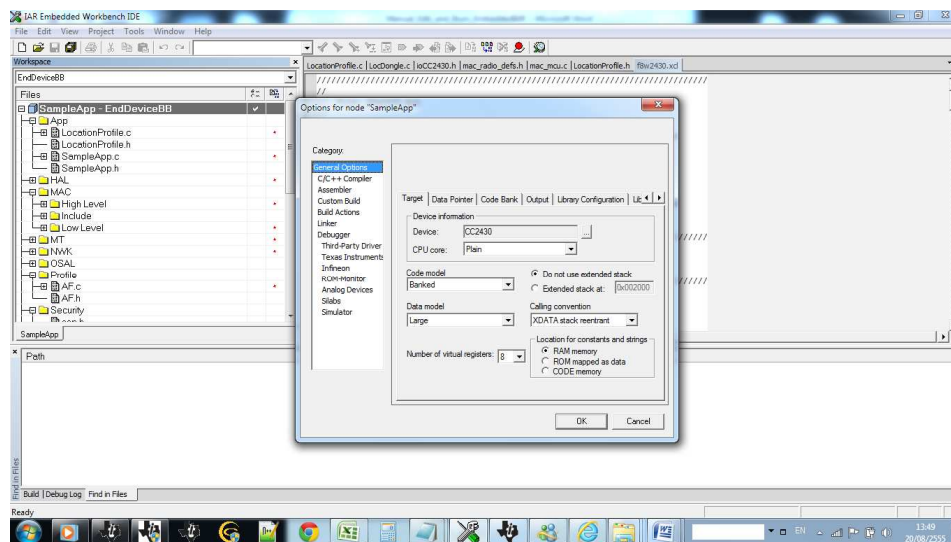


รูปที่ ข-31 แสดงโปรแกรม Smart RF Studio 7

- เลือก Retain IEEE ออก
- กด Read IEEE
- Browser :  
C:\Users\Noon\Desktop\Projects\_Location\_Engine\Embedded\ZStack-1.4.3-1.2.1\Projects\zstack\Samples\Location\CC2430DB\RouterBB\Exe\RouterBB.hex
- กด Perform Action
- กด Write IEEE
- Success Output

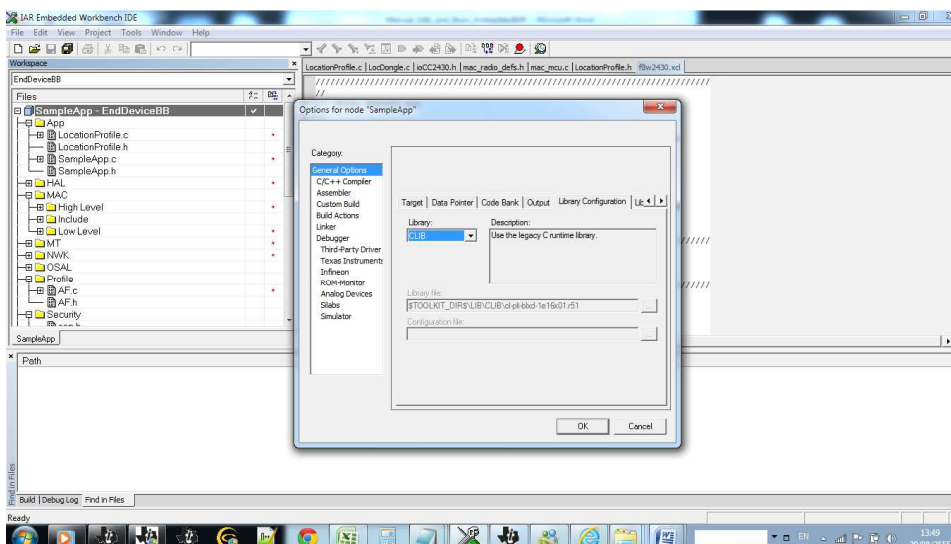
### 3. โหมด EndDevice หรือ Blind node

- ไปที่ Tab General Options



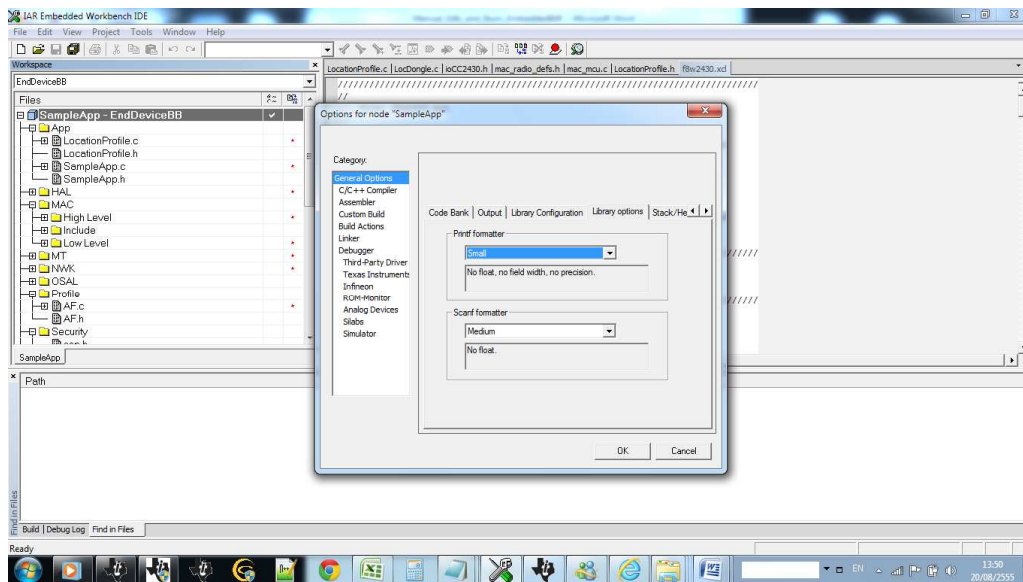
รูปที่ ข-32 แสดงการเข้าไปแก้ไขค่า Configure ใน Tab General Option

- ไปที่ Tab Configuration



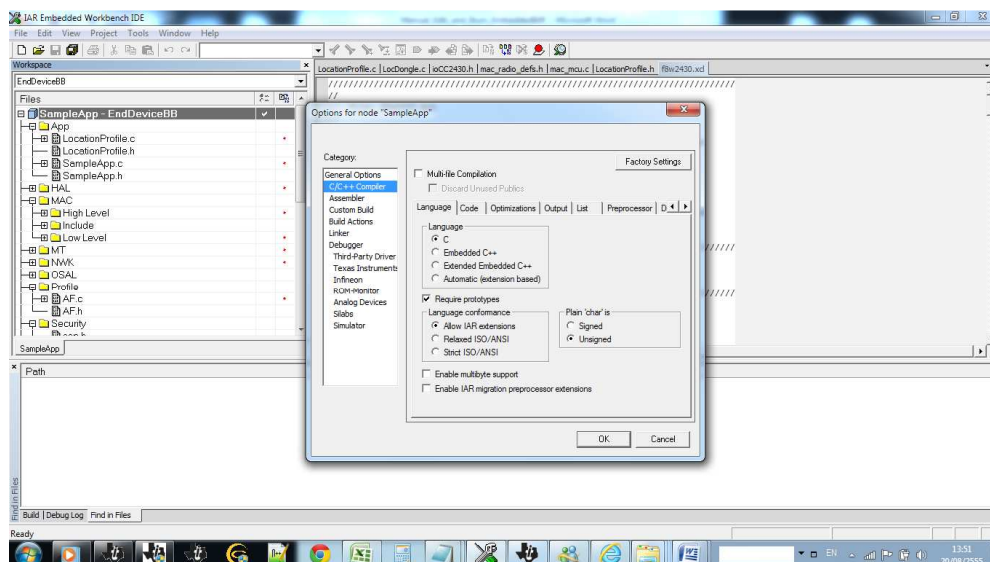
รูปที่ ข-33 แสดงการเข้าไปแก้ไขค่า Configure ใน Tab Configuration

- ไปที่ Tab Library Option



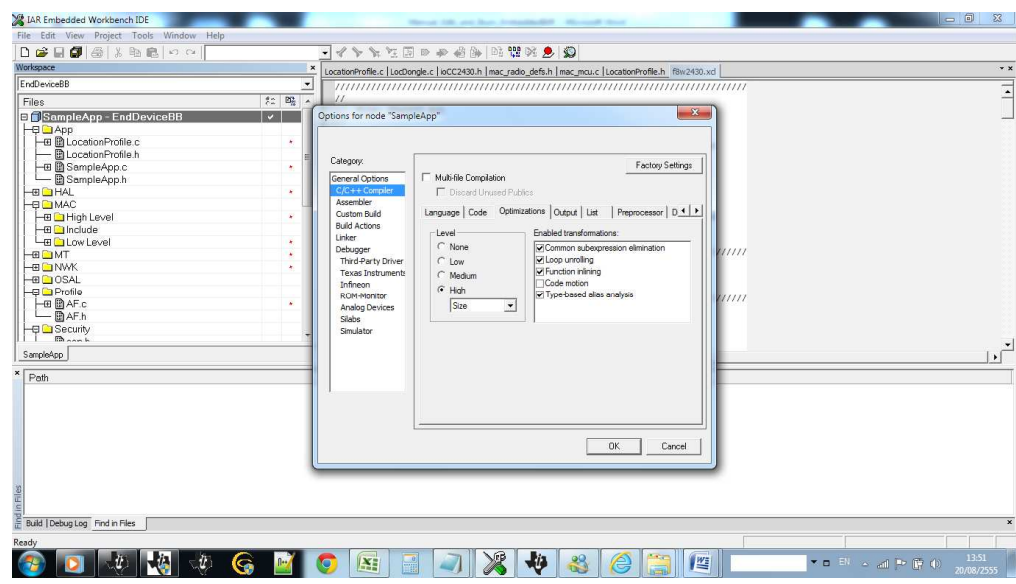
รูปที่ ข-34 แสดงการเข้าไปแก้ไขค่า Configure ใน Tab Library Option

- ไปที่ Tab C/C++ Compiler



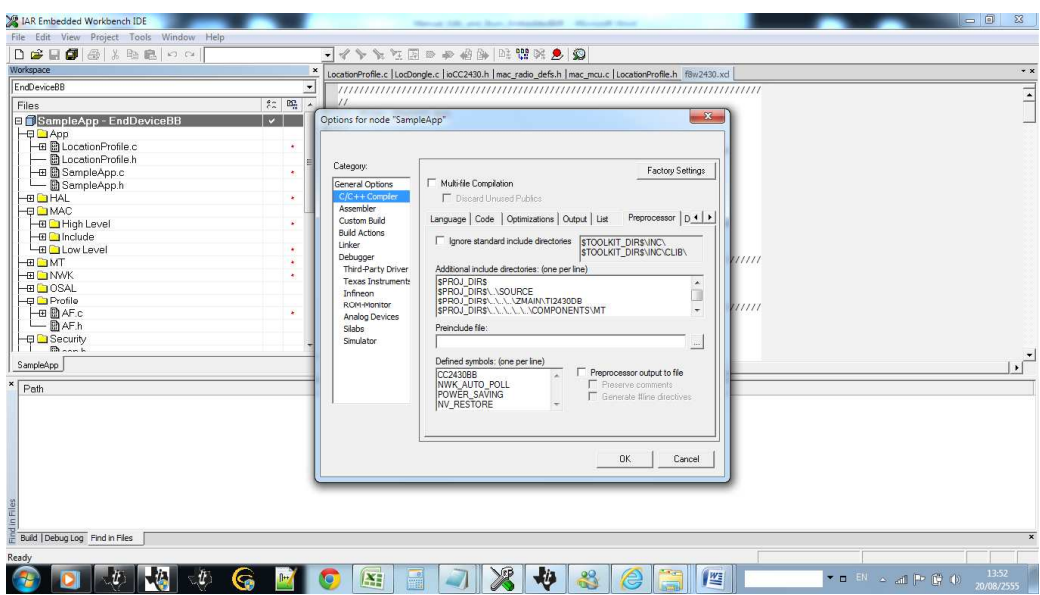
รูปที่ ข-35 แสดงการเข้าไปแก้ไขค่า Configure ใน Tab C/C++ Compiler

- ไปที่ Tab Optimizations



รูปที่ ข-36 แสดงการเข้าไปแก้ไขค่า Configure ใน Tab Optimization

- ไปที่ Tab Preprocessor



รูปที่ ข-37 แสดงการเข้าไปแก้ไขค่า Configure ใน Tab Preprocessor

- Copy Dir เหล่านี้ไปไว้ใน Additional include directories :

\$PROJ\_DIR\$

\$PROJ\_DIR\$\..\SOURCE

\$PROJ\_DIR\$\..\..\ZMAIN\TI2430DB

\$PROJ\_DIR\$\..\..\..\COMPONENTS\MT

\$PROJ\_DIR\$\..\..\..\COMPONENTS\HAL\INCLUDE

\$PROJ\_DIR\$\..\..\..\COMPONENTS\HAL\TARGET\CC2430BB

\$PROJ\_DIR\$\..\..\..\COMPONENTS\OSAL\MCU\CCSOC

\$PROJ\_DIR\$\..\..\..\COMPONENTS\OSAL\INCLUDE

\$PROJ\_DIR\$\..\..\..\COMPONENTS\STACK\AF

\$PROJ\_DIR\$\..\..\..\COMPONENTS\STACK\NWK

\$PROJ\_DIR\$\..\..\..\COMPONENTS\STACK\SEC

\$PROJ\_DIR\$\..\..\..\COMPONENTS\STACK\SYS

\$PROJ\_DIR\$\..\..\..\COMPONENTS\STACK\ZDO

\$PROJ\_DIR\$\..\..\..\COMPONENTS\ZMAC\F8W

\$PROJ\_DIR\$\..\..\..\COMPONENTS\ZMAC

\$PROJ\_DIR\$\..\..\..\COMPONENTS\SERVICES\SADDR



\$PROJ\_DIR\$\\.\.\.\.\.\COMPONENTS\SERVICES\SDATA

\$PROJ\_DIR\$\\.\.\.\.\.\COMPONENTS\MAC\INCLUDE

\$PROJ\_DIR\$\\.\.\.\.\.\COMPONENTS\MAC\HIGH\_LEVEL

\$PROJ\_DIR\$\\.\.\.\.\.\COMPONENTS\MAC\LOW\_LEVEL\SRF03

\$PROJ\_DIR\$\\.\.\.\.\.\COMPONENTS\MAC\LOW\_LEVEL\SRF03\SINGLE\_CHIP

Defined Symbol :

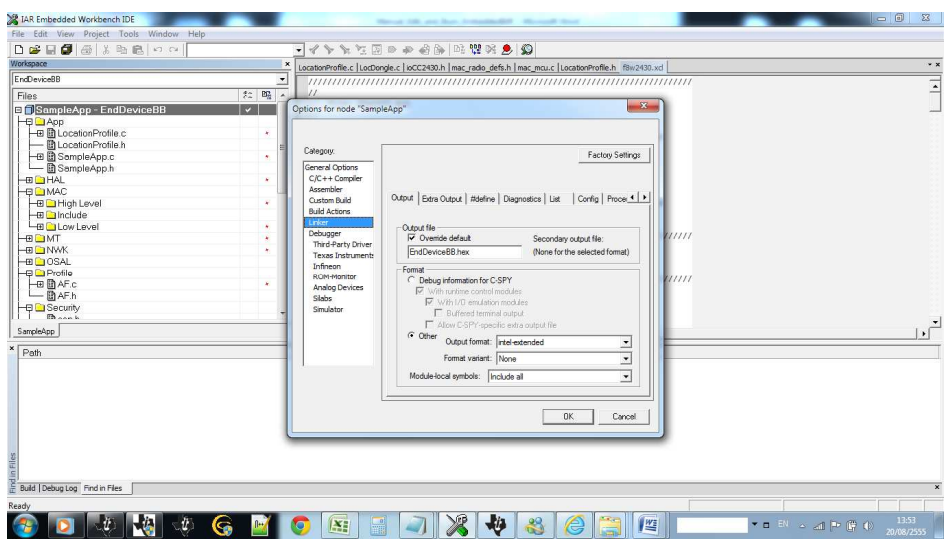
CC2430BB

NWK\_AUTO\_POLL

POWER\_SAVING

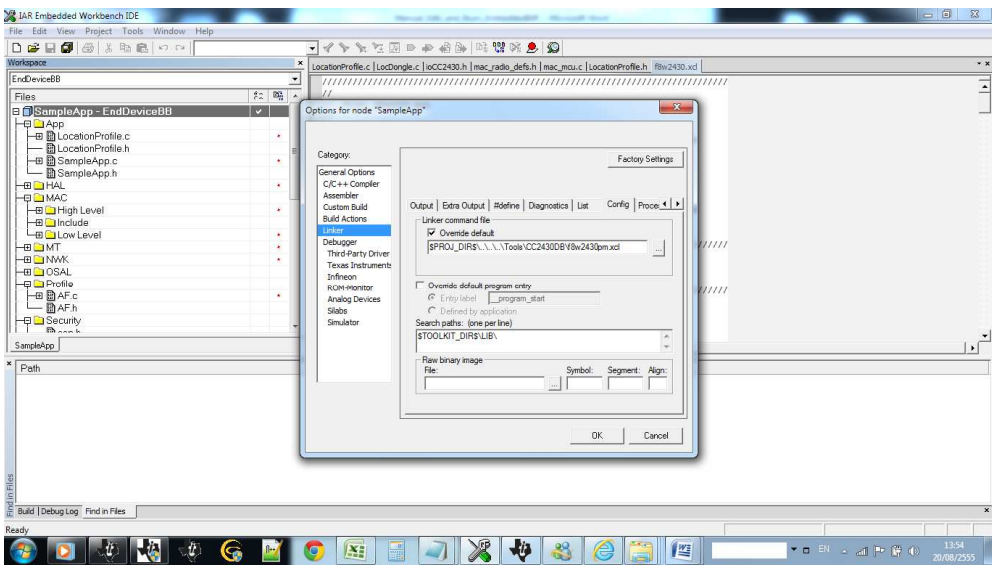
NV\_RESTORE

- ไปที่ Tab Linker



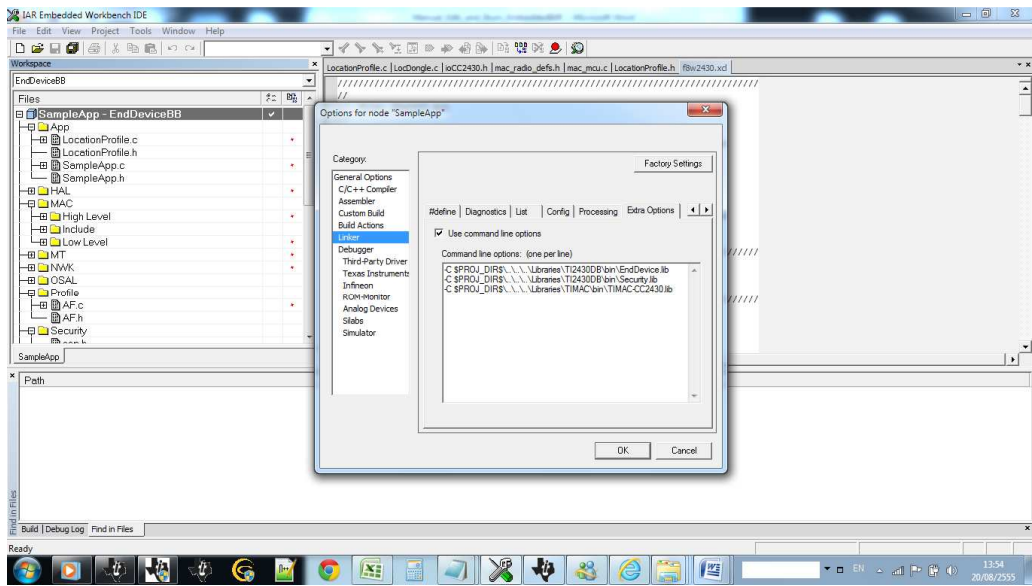
รูปที่ ข-38 แสดงการเข้าไปแก้ไขค่า Configure ใน Tab Linker

- ไปที่ Tab Config



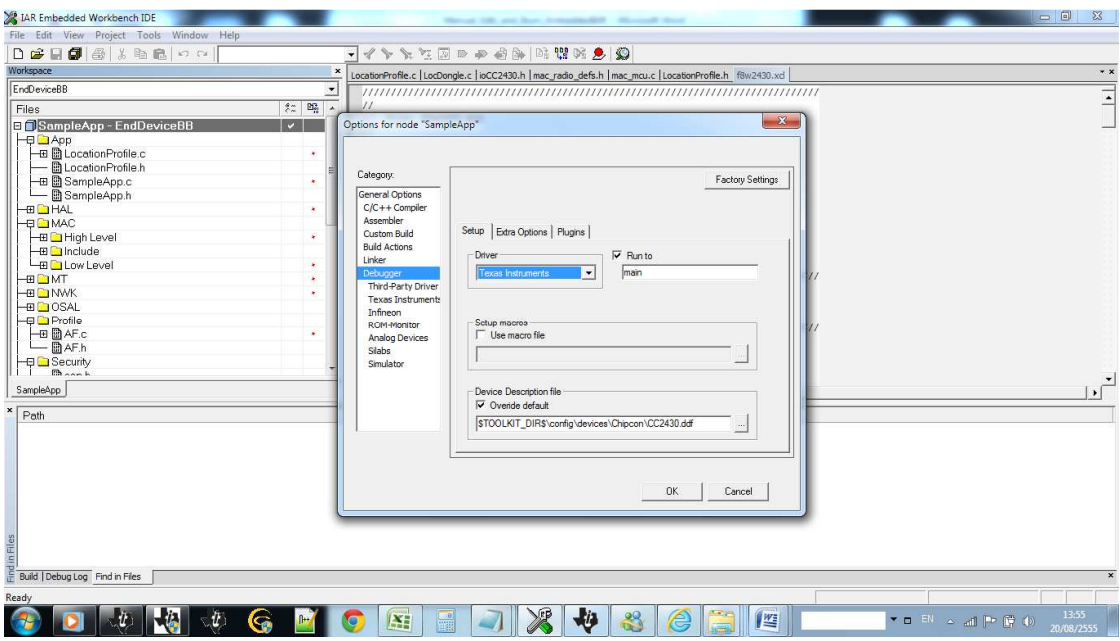
รูปที่ ข-39 แสดงการเข้าไปแก้ไขค่า Configure ใน Tab Config

- ไปที่ Tab Extra Option



รูปที่ ข-40 แสดงการเข้าไปแก้ไขค่า Configure ใน Tab Extra Option

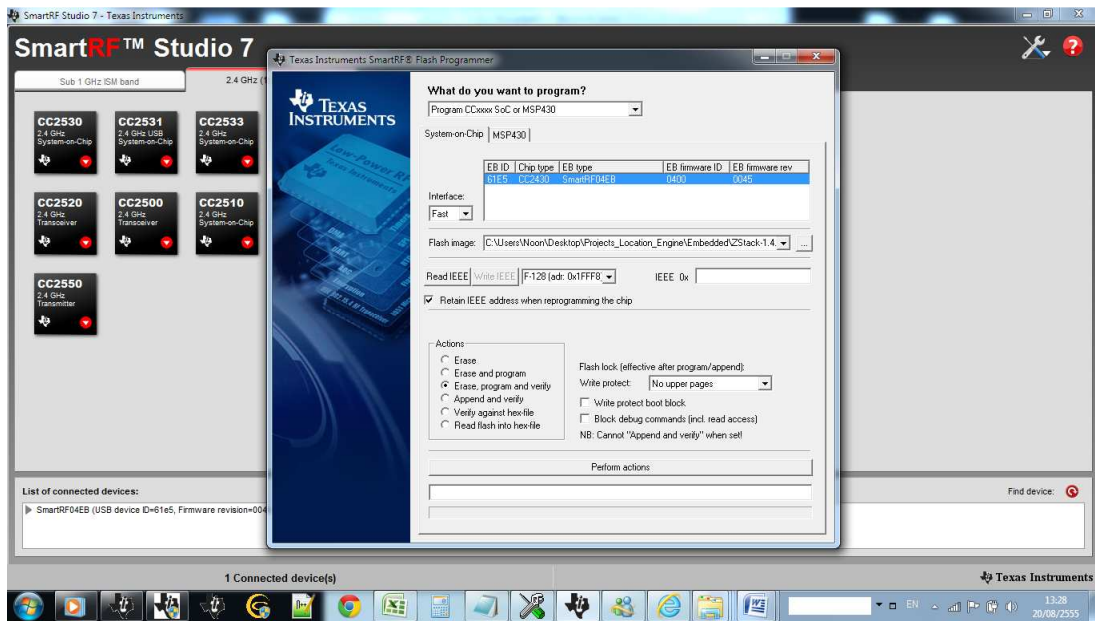
- ไปที่ Tab : Debugger > Setup



รูปที่ ข-41 แสดงการเข้าไปแก้ไขค่า Configure ใน Tab Setup

## การ Burn Firmware ลง Chipcon

- ไปที่ Application Smart RF Studio 7



รูปที่ ข-42 แสดงโปรแกรม Smart Studio RF7

- เลือก Retain IEEE ออก
- กด Read IEEE
- EndDeviceBB =>

C:\Users\Noon\Desktop\Projects\_Location\_Engine\Embedded\ZStack-1.4.3-1.2.1\Projects\zstack\Samples\Location\CC2430DB\EndDeviceBB\Exe\EndDeviceBB.hex

- กด Perform Action
- กด Write IEEE
- Success Output

#### 4. ส่วนที่เป็น Source Code .Net

##### 4.1 การดักและตรวจสอบ Protocol Messages

```
5.     public delegate void SerialPortDelegate();
6.     private void OnSerialDataReceive(object sender,
    SerialDataReceivedEventArgs e)
7.     {
8.         Thread.Sleep(1000);
9.         SerialPortDelegate d = new SerialPortDelegate(update);
10.        this.BeginInvoke(d, new object[] { });
11.        #if Debug
12.            Console.WriteLine("Data Received");
13.        #endif
14.    }
15.
16.    static int StartOfProtocolPosition = (0);
17.    static int ProtocolLength = (1); // 16
18.    static int ProtocolIdentify3 = (2); // 0
19.    static int ID_HI_BYTE = (6);
20.    static int ID_LO_BYTE = (5);
21.    static int X_Position_HI_BYTE = (11);
22.    static int X_Position_LO_BYTE = (12);
23.    static int Y_Position_HI_BYTE = (13);
24.    static int Y_Position_LO_BYTE = (14);
25.    static int RSSI_POSITION = (15);
26.    static int ProtocolCRCPosition = (16); // sum
27.    static double[,] NodeInfo = new double[6, 17];
28.    static double[,] TempNodeInfo = new double[6, 17];
```

```
29.     static int TempNodeCounter = 0;
30.
31.     static double TempXRefNodePosition;
32.     static double TempYRefNodePosition;
33.     static int TempXRefNodePosition1;
34.
35.
36.
37.
38.
39.
40.
41. //     static int SummationFactor = 2;
42.
43.     static int SummationFactor = 10; //จำนวนครั้งการเฉลี่ย RSSI
44.
45.
46.     public void update()
47.     {
48.         byte[] BLIND_NODE_RESP = { 0x02, 0x10, 0x18, 0x14, 0xCB };
49.         byte[] REF_NODE_RESP1 = { 0x02, 0x10, 0x18, 0x0C, 0xCB, 0x73, 0x43 };
50.         byte[] REF_NODE_RESP2 = { 0x02, 0x10, 0x18, 0x0C, 0xCB, 0x19, 0x4E };
51.         byte[] REF_NODE_RESP3 = { 0x02, 0x10, 0x18, 0x0C, 0xCB, 0x8A, 0x4D };
52.         byte[] REF_NODE_RESP4 = { 0x02, 0x10, 0x18, 0x0C, 0xCB, 0x18, 0x4E };
53.         int numberOfBytesToRead;
54.         int StartCutByte = 0;
55.
```

```
56.
57.     tmrProcessRSSI.Enabled = false;
58.
59.     numberOfBytesToRead = serialPort1.BytesToRead;
60.     Byte[] newReceivedData = new Byte[numberOfBytesToRead];
61.     serialPort1.Read(newReceivedData, 0, numberOfBytesToRead);
62.
63.     // tmrProcessRSSI.Enabled = false;
64.
65.
66.
67.     StartCutByte = IsExist(newReceivedData, BLIND_NODE_RESP);
68.     if (StartCutByte >= 0 && numberOfBytesToRead >= 25)
69.     {
70.         newReceivedData[StartCutByte] = 0xFF;
71.         Thread.Sleep(200);
72.
73.         SendXY_RSSI_REQ();
74.
75.
76.     }
77.
78.     StartCutByte = IsExist(newReceivedData, REF_NODE_RESP1);
79.     if (StartCutByte >= 0 && numberOfBytesToRead >= 17)
80.     {
81.         for (int i = 0; i < 17; i++)
82.         {
```

```
83.         TempNodeInfo[0, i] = newReceivedData[i + StartCutByte];
84.
85.     }
86.
87.     newReceivedData[ StartCutByte] = 0xFF;
88. }
89.
90. StartCutByte = IsExist(newReceivedData, REF_NODE_RESP2);
91. if (StartCutByte >= 0 && numberOfBytesToRead >= 17)
92. {
93.     for (int i = 0; i < 17; i++)
94.     {
95.         TempNodeInfo[1, i] = newReceivedData[i + StartCutByte];
96.
97.     }
98.     newReceivedData[StartCutByte] = 0xFF;
99. }
100.     StartCutByte = IsExist(newReceivedData, REF_NODE_RESP3);
101.     if (StartCutByte >= 0 && numberOfBytesToRead >= 17)
102.     {
103.         for (int i = 0; i < 17; i++)
104.         {
105.             TempNodeInfo[2, i] = newReceivedData[i + StartCutByte];
106.
107.         }
108.         newReceivedData[StartCutByte] = 0xFF;
109.     }
```



```
110.         StartCutByte = IsExist(newReceivedData, REF_NODE_RESP4);
111.         if (StartCutByte >= 0 && numberOfBytesToRead >= 17)
112.         {
113.             for (int i = 0; i < 17; i++)
114.             {
115.                 TempNodeInfo[3, i] = newReceivedData[i + StartCutByte];
116.             }
117.             newReceivedData[StartCutByte] = 0xFF;
118.         }
119.
120.         for (int j = 0; j < 4; j++)
121.         {
122.             for (int i = 0; i < 17; i++)
123.             {
124.
125.                 NodeInfo[j, i] = TempNodeInfo[j, i];
126.
127.             }
128.
129.             Console.WriteLine("LQI"+j.ToString()+"="+NodeInfo[j,RSSI_POSITION]);
130.             tmrProcessRSSI.Enabled = true;
131.
132.
133.
134.         }
135.
```

```
136.     public static int IsExist(byte[] Comparer,byte[]dummy)
137.     {
138.         int DataLength = Comparer.Length;
139.         int i=0;
140.         int j=0;
141.         while(DataLength > 0)
142.         {
143.             if(Comparer[j] != dummy[i])
144.             {
145.                 i=0;
146.             }
147.             else
148.             {
149.                 i = i + 1;
150.                 if(i == dummy.Length)
151.                 {
152.                     return j+1-i;
153.                 }
154.             }
155.             j = j+1;
156.             DataLength = DataLength - 1;
157.         }
158.         return -1;
159.     }
160.     void SendXY_RSSI_REQ()
161.     {
162.         LocationEngine.MaxLikelihood ML = new MaxLikelihood();
```

```

163.         byte[]RSSI_REQ_CONST = new byte[]
           {002,000,024,007,203,255,255,202,017,000,000,015};
164.         if(serialPort1.IsOpen == true)
165.         {
166.             serialPort1.Write(RSSI_REQ_CONST,0,12);
167.         }
168.     }
169.     public static bool XORCheckSumOK(int[] data)
170.     {
171.         int xorchecksum = 0;
172.         for (int i = 1; i < 16; i++)
173.         {
174.             xorchecksum = xorchecksum ^ data[i];
175.         }
176.
177.         if (xorchecksum == data[16])
178.         {
179.             return true;
180.         }
181.         else
182.         {
183.             return false;
184.         }
185.     }

```

#### 185.1 การแปลงสัญญาณจาก LQI เป็น RSSI

```

186.     private void tmrProcessRSSI_Tick(object sender, EventArgs e)

```

```
187.     {
188.
189.         double iBlindPosition_X, iBlindPosition_Y;
190.         MatrixLibrary.Matrix DataMatrix = new MatrixLibrary.Matrix(4, 3);
191.         MatrixLibrary.Matrix Result = new MatrixLibrary.Matrix(2, 1);
192.
193.         for (int i = 0; i < 4; i++)
194.         {
195.
196.
197.
198.             TempXRefNodePosition = (81 - (NodeInfo[i, RSSI_POSITION] *
199.             91) / 255);
200.             NodeInfo[i, RSSI_POSITION] = TempXRefNodePosition;
201.
202.
203.             DisplayData(LogMsgType.Incoming, i.ToString() + ": " +
                NodeInfo[i, RSSI_POSITION].ToString() + "\n");
204.             Console.WriteLine("NodeInfo [RSSI_Position]" + NodeInfo[i,
                RSSI_POSITION].ToString() + "\n");
205.
206.
207.             text_n_value.Text = (trackbarN.Value * 0.025).ToString();
208.             // text_a_value.Text = (trackbarA.Value / 4.000).ToString();
209.
210.
211.             if (rbOutdoor.Checked)
```

```
212.         {
213.
214.             txtAValue.Text = "47.865625";
215.             text_a_value.Text = txtAValue.Text;
216.
217.             DataMatrix[0, 0] = toDistance(NodeInfo[0, RSSI_POSITION],
                Convert.ToDouble(trackbarN.Value) * 0.025, Convert.ToDouble(txtAValue.Text));
218.             Console.WriteLine("DataMatrix [0,0]" + DataMatrix[0, 0] + "N = "
                + trackbarN.Value * 0.025 + "A = " + txtAValue.Text);
219.             DataMatrix[1, 0] = toDistance(NodeInfo[1, RSSI_POSITION],
                Convert.ToDouble(trackbarN.Value) * 0.025, Convert.ToDouble(txtAValue.Text));
220.             Console.WriteLine("DataMatrix [1,0]" + DataMatrix[1, 0] + "N = "
                + trackbarN.Value * 0.025 + "A = " + txtAValue.Text);
221.             DataMatrix[2, 0] = toDistance(NodeInfo[2, RSSI_POSITION],
                Convert.ToDouble(trackbarN.Value) * 0.025, Convert.ToDouble(txtAValue.Text));
222.             Console.WriteLine("DataMatrix [2,0]" + DataMatrix[2, 0] + "N = "
                + trackbarN.Value * 0.025 + "A = " + txtAValue.Text);
223.             DataMatrix[3, 0] = toDistance(NodeInfo[3, RSSI_POSITION],
                Convert.ToDouble(trackbarN.Value) * 0.025, Convert.ToDouble(txtAValue.Text));
224.             Console.WriteLine("DataMatrix [3,0]" + DataMatrix[3, 0] + "N = "
                + trackbarN.Value * 0.025 + "A = " + txtAValue.Text);
225.
226.         }
227.         else if (rbIndoor.Checked)
228.         {
229.             txtAValue.Text = "47.85";
230.             text_a_value.Text = txtAValue.Text;
```

```
231.             DataMatrix[0, 0] = toDistance(NodeInfo[0, RSSI_POSITION],
                Convert.ToDouble(trackbarN.Value) * 0.025, Convert.ToDouble(txtAValue.Text));
232.             Console.WriteLine("DataMatrix [0,0]" + DataMatrix[0, 0] + "N = "
                + trackbarN.Value * 0.025 + "A = " + txtAValue.Text);
233.             DataMatrix[1, 0] = toDistance(NodeInfo[1, RSSI_POSITION],
                Convert.ToDouble(trackbarN.Value) * 0.025, Convert.ToDouble(txtAValue.Text));
234.             Console.WriteLine("DataMatrix [1,0]" + DataMatrix[1, 0] + "N = "
                + trackbarN.Value * 0.025 + "A = " + txtAValue.Text);
235.             DataMatrix[2, 0] = toDistance(NodeInfo[2, RSSI_POSITION],
                Convert.ToDouble(trackbarN.Value) * 0.025, Convert.ToDouble(txtAValue.Text));
236.             Console.WriteLine("DataMatrix [2,0]" + DataMatrix[2, 0] + "N = "
                + trackbarN.Value * 0.025 + "A = " + txtAValue.Text);
237.             DataMatrix[3, 0] = toDistance(NodeInfo[3, RSSI_POSITION],
                Convert.ToDouble(trackbarN.Value) * 0.025, Convert.ToDouble(txtAValue.Text));
238.             Console.WriteLine("DataMatrix [3,0]" + DataMatrix[3, 0] + "N = "
                + trackbarN.Value * 0.025 + "A = " + txtAValue.Text);
239.
240.         }
241.     else
242.     {
243.
244.
245.         text_a_value.Text = (trackbarA.Value / 4.000).ToString();
246.
247.         DataMatrix[0, 0] = toDistance(NodeInfo[0, RSSI_POSITION],
                Convert.ToDouble(trackbarN.Value) * 0.025, Convert.ToDouble(trackbarA.Value /
                4.000));
```

```
248.         Console.WriteLine("DataMatrix [0,0]" + DataMatrix[0, 0] + "N = "
    + trackbarN.Value * 0.025 + "A = " + trackbarA.Value / 4.0);
249.
250.         DataMatrix[1, 0] = toDistance(NodeInfo[1, RSSI_POSITION],
    Convert.ToDouble(trackbarN.Value) * 0.025, Convert.ToDouble(trackbarA.Value /
    4.000));
251.         Console.WriteLine("DataMatrix [1,0]" + DataMatrix[1, 0] + "N = "
    + trackbarN.Value * 0.025 + "A = " + trackbarA.Value / 4.0);
252.
253.         DataMatrix[2, 0] = toDistance(NodeInfo[2, RSSI_POSITION],
    Convert.ToDouble(trackbarN.Value) * 0.025, Convert.ToDouble(trackbarA.Value /
    4.000));
254.
255.         Console.WriteLine("DataMatrix [2,0]" + DataMatrix[2, 0] + "N = "
    + trackbarN.Value * 0.025 + "A = " + Convert.ToDouble(trackbarA.Value / 4.000));
256.         DataMatrix[3, 0] = toDistance(NodeInfo[3, RSSI_POSITION],
    Convert.ToDouble(trackbarN.Value) * 0.025, Convert.ToDouble(trackbarA.Value /
    4.000));
257.
258.         Console.WriteLine("DataMatrix [3,0]" + DataMatrix[3, 0] + "N = "
    + trackbarN.Value * 0.025 + "A = " + Convert.ToDouble(trackbarA.Value / 4.000));
259.
260.     }
261. }
262.
263.
264.     DataMatrix[0, 1] = 0;     // คือ X = 0
```

```
265.         DataMatrix[0, 2] = 0;           // y = 0
266.
267.         DataMatrix[1, 1] = Convert.ToDouble(tbMapWidth.Text); // x= 10
268.         DataMatrix[1, 2] = 0;           // y = 0
269.
270.         DataMatrix[2, 1] = Convert.ToDouble(tbMapWidth.Text); // x= 10
271.         DataMatrix[2, 2] = Convert.ToDouble(tbMapHeight.Text); // y= 10
272.
273.         DataMatrix[3, 1] = 0;           // x=0
274.         DataMatrix[3, 2] = Convert.ToDouble(tbMapHeight.Text); // y=10
275.
276.         //Check if Matrix is not Zero.
277.         BlindX.Text = Convert.ToDouble(MaxLikelihood.getXY(DataMatrix)[0,
           0]).ToString("0.00000");
278.         BlindY.Text = Convert.ToDouble(MaxLikelihood.getXY(DataMatrix)[1,
           0]).ToString("0.00000");
279.
280.
281.         tbBlindX_MM.Text = Convert.ToDouble(MinMax.getXY(DataMatrix)[0,
           0]).ToString("0.00000");
282.         tbBlindY_MM.Text = Convert.ToDouble(MinMax.getXY(DataMatrix)[1,
           0]).ToString("0.00000");
283.
284.         if (rbSelectML.Checked)
285.         {   Result = MaxLikelihood.getXY(DataMatrix);
286.
287.         iBlindPosition_X = Convert.ToDouble((Result[0, 0]));
```



```
288.         iBlindPosition_Y = Convert.ToDouble((Result[1, 0]));
289.
290.
291.         BlindX.Text = iBlindPosition_X.ToString();
292.         BlindY.Text = iBlindPosition_Y.ToString();
293.         Console.WriteLine("BlindX Pass Calculate already"+BlindX.Text);
294.         Console.WriteLine("BlindY Pass Calculate already" + BlindY.Text);
295.
296.
297.         UpdateBlindNodePosition_ML();
298.         imageBlindNode.Visible = true;
299.
300.     }
301.     else if (rbSelectMM.Checked)
302.     {
303.         Result = MinMax.getXY(DataMatrix);
304.         iBlindPosition_X = Convert.ToDouble(Result[0, 0]);
305.         iBlindPosition_Y = Convert.ToDouble(Result[1, 0]);
306.
307.
308.         tbTI_X.Text = iBlindPosition_X.ToString();
309.         tbTI_Y.Text = iBlindPosition_Y.ToString();
310.
311.
312.         tbBlindX_MM.Text = iBlindPosition_X.ToString();
313.         tbBlindY_MM.Text = iBlindPosition_Y.ToString();
314.
```

```
315.             UpdateBlindNodePosition_MM();
316.             imageBlindNode.Visible = true;
317.
318.         }
319.
320.             tmrProcessRSSI.Enabled = false;
321.             if (rbIndoor.Checked)
322.             {
323.
324.                 tw.WriteLine("[A]=" + Convert.ToDouble("47.85").ToString("F4")
+ ",[N]=" + Convert.ToDouble(trackbarN.Value * 0.025).ToString("F4") +
325.                     "RSSI=" + NodeInfo[0, RSSI_POSITION] + "," +
NodeInfo[1, RSSI_POSITION] + "," + NodeInfo[2, RSSI_POSITION] + "," +
NodeInfo[3, RSSI_POSITION] + "," +
326.                     "ML," + BlindX.Text.ToString() + "," +
BlindY.Text.ToString() + "," +
327.                     "MM," + tbBlindX_MM.Text.ToString() + "," +
tbBlindY_MM.Text.ToString()
328.                 );
329.                 tw.Flush();
330.
331.             }
332.             else if (rbOutdoor.Checked)
333.             {
334.
```

```

335.             tw.WriteLine("[A]=" +
                Convert.ToDouble("47.865625").ToString("F4") + ",[N]=" +
                Convert.ToDouble(trackbarN.Value * 0.025).ToString("F4") +
336.             "RSSI=" + NodeInfo[0, RSSI_POSITION] + ";" +
                NodeInfo[1, RSSI_POSITION] + ";" + NodeInfo[2, RSSI_POSITION] + ";" +
                NodeInfo[3, RSSI_POSITION] + ";" +
337.             "ML," + BlindX.Text.ToString() + ";" +
                BlindY.Text.ToString() + ";" +
338.             "MM," + tbBlindX_MM.Text.ToString() + ";" +
                tbBlindY_MM.Text.ToString()
339.             );
340.             tw.Flush();
341.
342.         }
343.     else
344.     {
345.
346.
347.             tw.WriteLine("[A]=" + Convert.ToDouble(trackbarA.Value /
                4.000).ToString("F4") + ",[N]=" + Convert.ToDouble(trackbarN.Value *
                0.025).ToString("F4") +
348.             "RSSI=" + NodeInfo[0, RSSI_POSITION] + ";" +
                NodeInfo[1, RSSI_POSITION] + ";" + NodeInfo[2, RSSI_POSITION] + ";" +
                NodeInfo[3, RSSI_POSITION] + ";" +
349.             "ML," + BlindX.Text.ToString() + ";" +
                BlindY.Text.ToString() + ";" +

```

```

350.         "MM," + tbBlindX_MM.Text.ToString() + "," +
           tbBlindY_MM.Text.ToString()
351.         );
352.         tw.Flush();
353.
354.
355.
356.
357.
358.     }
359. }

```

### 359.1 การแปลงสัญญาณเป็นระยะทาง

```

360.
361.     public double toDistance(double RSSI, double n, double A)
362.     {
363.         return Math.Pow(10, ((RSSI - A) / (10 * n)));
364.
365.     }
366.

```

### 366.1 อัลกอริทึม Maximum Likelihood

```

367.     public class MaxLikelihood
368.     {
369.         //Least Mean Squire Methodor MMSE Minimum Mean Squire use in
           Triratalation
370.

```

```

371.         public static Matrix getXY(Matrix distance,Matrix Xi,Matrix Yi)
372.         {
373.
374.             Matrix Result = new Matrix(2, 1);
375.             Matrix b = new Matrix(distance.NoRows, 1);
376.             Matrix A = new Matrix(distance.NoRows, 2);
377.
378.             if(distance.NoRows != Xi.NoRows && distance.NoRows !=
                Yi.NoRows && Xi.NoRows != Yi.NoRows)
379.                 {
380.                     throw new NotSupportedException();
381.                 }
382.             for(int i = 1;i < distance.NoRows;i++) {
383.                 b[i - 1,0]=Math.Pow(distance[0,0],2)-
384.                     Math.Pow(distance[i,0],2) +
385.                 (
386.                     Math.Pow(Xi[i,0],2)+
387.                     Math.Pow(Yi[i,0],2)-
388.                     Math.Pow(Xi[0,0],2)-
389.                     Math.Pow(Yi[0,0],2)
390.                 );
391.                 A[i-1,0]= 2 * Xi[i,0] - Xi[0,0];
392.                 A[i-1,0]= 2 * Yi[i,0] - Yi[0,0];
393.
394.             }
395.             Console.WriteLine("Node 0 Distance=" + distance[0, 0] + " Ref Pos X
                = " + distance[0, 1] + "Ref Pos Y = " + distance[0, 2]);

```

```

396.         Console.WriteLine("Node 1 Distance=" + distance[1, 0] + " Ref Pos X
           = " + distance[1, 1] + "Ref Pos Y = " + distance[1, 2]);
397.         Console.WriteLine("Node 2 Distance=" + distance[2, 0] + " Ref Pos X
           = " + distance[2, 1] + "Ref Pos Y = " + distance[2, 2]);
398.         Console.WriteLine("Node 3 Distance=" + distance[3, 0] + " Ref Pos X
           = " + distance[3, 1] + "Ref Pos Y = " + distance[3, 2]);
399.
400.
401.         Result = Matrix.Inverse((Matrix.Transpose(A) * A) *
           (Matrix.Transpose(A) * b));
402.         return Result;
403.
404.
405.     }
406.
407.
408.

```

#### 408.1 อัลกอริทึม Min-Max

```

409.     public class MinMax
410.     {
411.         public static double findminvalue(Matrix Input, int ColumnNumber)
412.         { // class หาค่า min
413.
414.             int i = 0;
415.             double minValue = 65536.0;
416.

```

```
417.         for (i = 0; i < Input.NoRows; i++)
418.     {
419.         if (Input[i, ColumnNumber] < minValue)
420.     {
421.         minValue = Input[i, ColumnNumber];
422.     }
423.     }
424.     return minValue;
425.
426.     }
427.     public static double findmaxvalue(Matrix DataMatrix, int
    ColumnNumber)
428.     { // class หาค่า max
429.         int i = 0;
430.         double maxValue = 0.0;
431.         for (i = 0; i < DataMatrix.NoRows; i++)
432.     {
433.         if (DataMatrix[i, ColumnNumber] > maxValue)
434.     {
435.         maxValue = DataMatrix[i, ColumnNumber];
436.     }
437.     }
438.     return maxValue;
439.
440.     }
441.     public static Matrix getXY(Matrix DataMatrix)
442.     {
```

```

443.         // InputMatrix
444.         //           /-----Reference-----\
445.         // | Distance | Xi | Yi |
446.
447.         Matrix Result = new Matrix(2, 1);
448.         // Reference Node
449.         // |X-left-up |Y-left-up|X-right-down|Y-right-down| <==> | Xlo | Ylo | |
           X Hi | YHi |
450.         Matrix ReferenceNode = new Matrix(DataMatrix.NoRows,4);
451.         //#if DEBUG
452.         Console.WriteLine("Node 0 Distance=" + DataMatrix[0, 0] + " Ref Pos
           X = " + DataMatrix[0, 1] + "Ref Pos Y = " + DataMatrix[0, 2]);
453.         Console.WriteLine("Node 1 Distance=" + DataMatrix[1, 0] + " Ref Pos
           X = " + DataMatrix[1, 1] + "Ref Pos Y = " + DataMatrix[1, 2]);
454.         Console.WriteLine("Node 2 Distance=" + DataMatrix[2, 0] + " Ref Pos
           X = " + DataMatrix[2, 1] + "Ref Pos Y = " + DataMatrix[2, 2]);
455.         Console.WriteLine("Node 3 Distance=" + DataMatrix[3, 0] + " Ref Pos
           X = " + DataMatrix[3, 1] + "Ref Pos Y = " + DataMatrix[3, 2]);
456.         //#endif
457.         if (DataMatrix.NoCols != 3)
458.         {
459.             throw new NotSupportedException();
460.         }
461.         else
462.         {
463.             for(int i=1;i<=DataMatrix.NoRows;i++)
464.             {

```



```

465.          //
466.          Console.WriteLine("Column[" + i+"] ");
467.          ReferenceNode[i - 1, 0] = DataMatrix[i - 1, 1] - DataMatrix[i - 1,
           0]; //Xlo
468.          ReferenceNode[i - 1, 1] = DataMatrix[i - 1, 2] - DataMatrix[i - 1,
           0]; //Ylo
469.          ReferenceNode[i - 1, 2] = DataMatrix[i - 1, 1] + DataMatrix[i - 1,
           0]; //XHi
470.          ReferenceNode[i - 1, 3] = DataMatrix[i - 1, 2] + DataMatrix[i - 1,
           0]; //YHi
471.
472.
473.          Console.WriteLine(ReferenceNode[i - 1, 0] + "," + ReferenceNode[i
           - 1, 1] + "," + ReferenceNode[i - 1, 2] + "," + ReferenceNode[i - 1, 3]);
474.          }
475.
476.          }
477.
478.          //Select (Min of Max) and (Max of Min)
479.
480.          Console.WriteLine("Max of Min X " + findmaxvalue(ReferenceNode,
           0));
481.          Console.WriteLine("Max of Min Y " + findmaxvalue(ReferenceNode,
           1));
482.          Console.WriteLine("Min of Max X " + findminvalue(ReferenceNode,
           2));

```

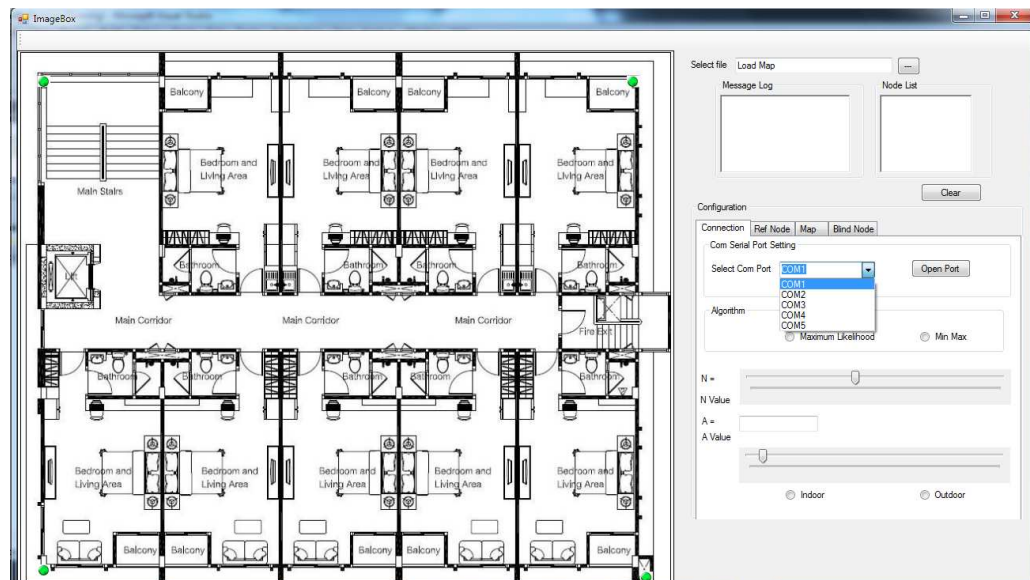
```
483.         Console.WriteLine("Min of Max Y " + findminvalue(ReferenceNode,
3));
484.
485.
486.         Result[0, 0] = (findminvalue(ReferenceNode, 2) +
findmaxvalue(ReferenceNode, 0))/2.000 ;
487.         Result[1, 0] = (findminvalue(ReferenceNode, 3) +
findmaxvalue(ReferenceNode, 1))/2.000 ;
488.
489.
490.
491.         return Result;
492.
493.     }
494. }
495.
496.
497.
498. }
```

## ภาคผนวก ค

คู่มือการใช้งานโปรแกรมการค้นหาตำแหน่งวัตถุ  
การใช้งานโปรแกรมมีรายละเอียดดังต่อไปนี้

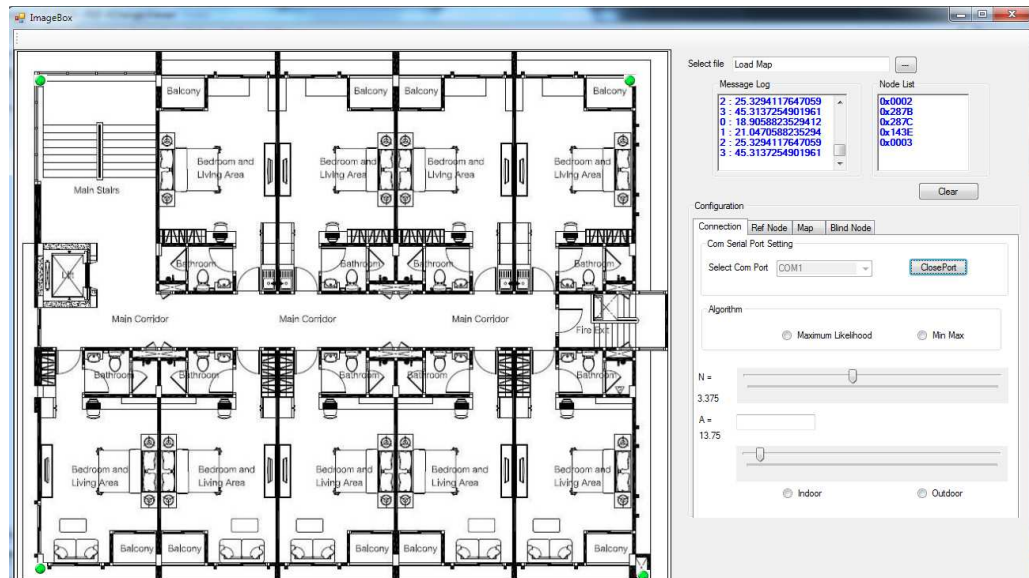
1. ส่วนติดต่อกับอุปกรณ์ผู้ใช้งานคลิกเข้ามาหน้าแรก ให้ทำการเลือก Select Com Port เป็น COM1

กด Open Port



รูปที่ ค-1 หน้าแรกแสดงการเชื่อมต่อกับอุปกรณ์ชิคปี

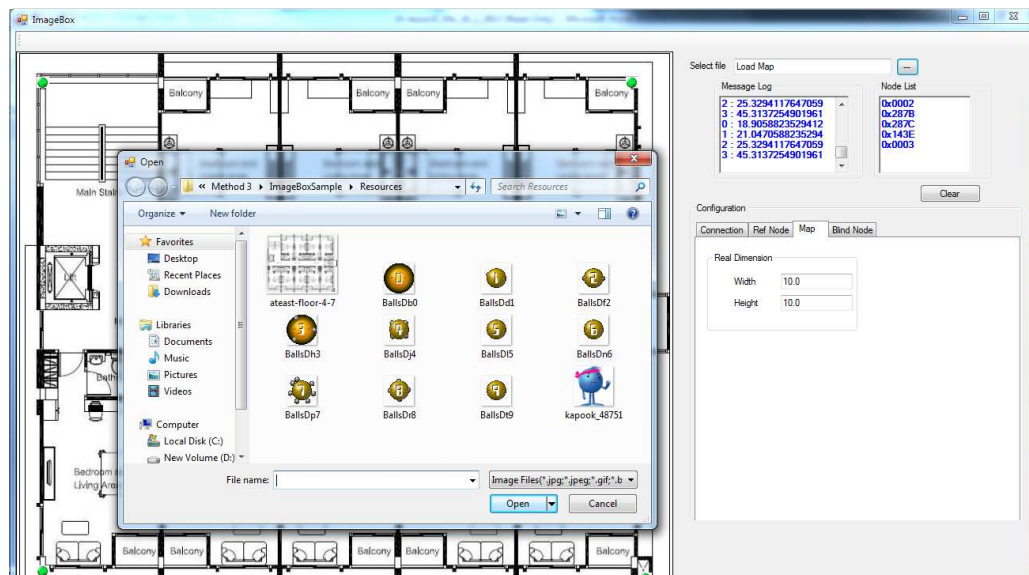
- หลังจากได้ทำการเชื่อมต่อกับอุปกรณ์ซิกนัลที่เรียบร้อยแล้วจะปรากฏค่า RSSI และ Node ID ดังภาพ



รูปที่ ค-2 แสดงหลังการเชื่อมต่อกับอุปกรณ์ซิกนัลที่ได้สำเร็จ

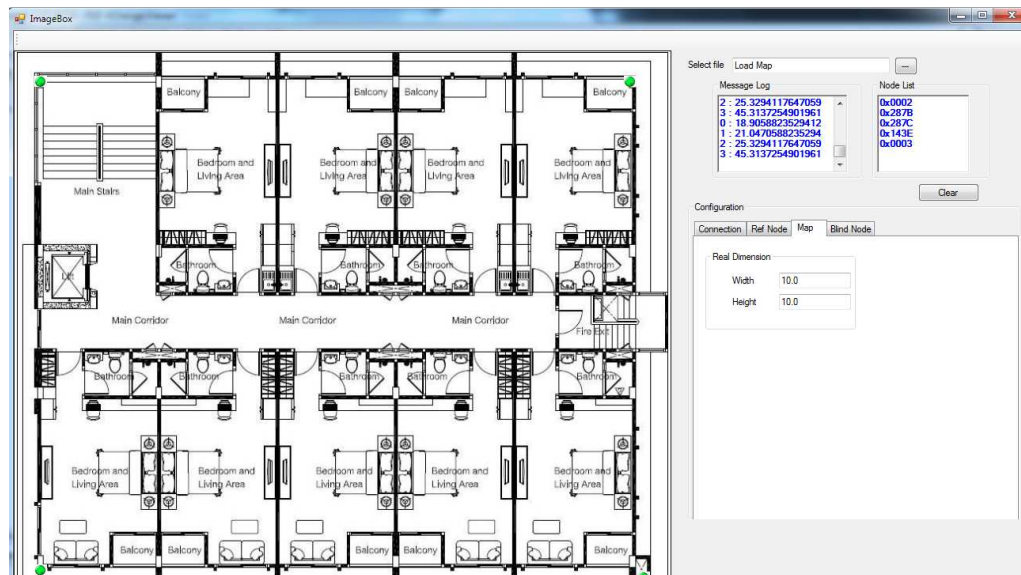
- การ Upload แผนที่ฉบับใหม่เข้าสู่ระบบ

- กดปุ่ม Load Map



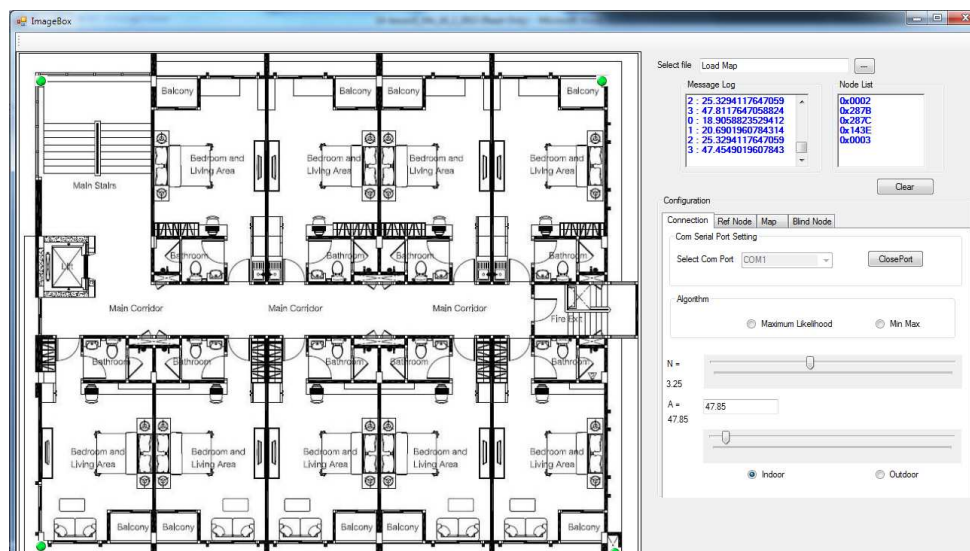
รูปที่ ค-3 แสดงการ Upload แผนที่ (GUI Map)

4. การปรับค่าขนาดแผนที่



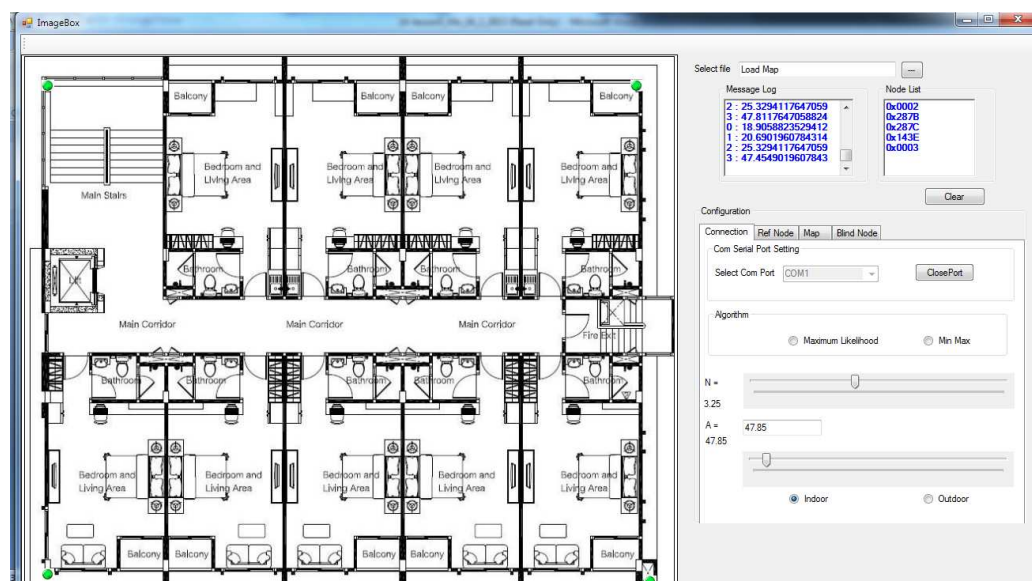
รูปที่ ค-4 แสดงโหมดการปรับค่าแผนที่

5. เลือกสภาพแวดล้อมให้เหมาะสมแบ่งเป็น 2 ประเภทคือแบบภายในอาคาร ให้เลือก “Indoor” และกรณีแบบภายนอกอาคารให้เลือกแบบ “Outdoor”



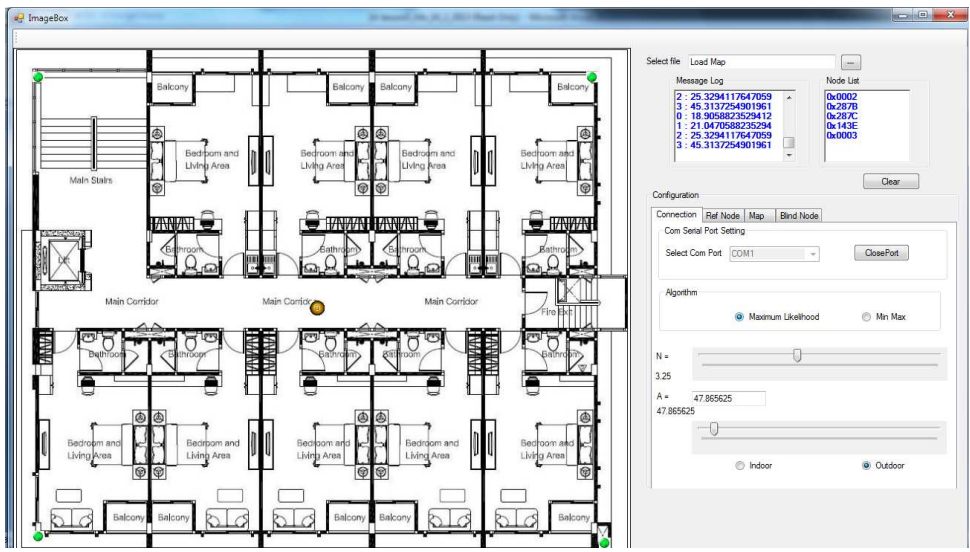
รูปที่ ค-5 แสดงโหมดการเลือก Type : Indoor และ Outdoor

6. เลือก Tab Scroll Bar ปรับค่า A และ N ได้ตามความเหมาะสมแต่ในทางปฏิบัติหลังจากที่ผู้ใช้เลือกสภาพแวดล้อมแบบ Indoor หรือ Outdoor ระบบจะกำหนดค่าที่เหมาะสมให้อย่างอัตโนมัติแล้ว



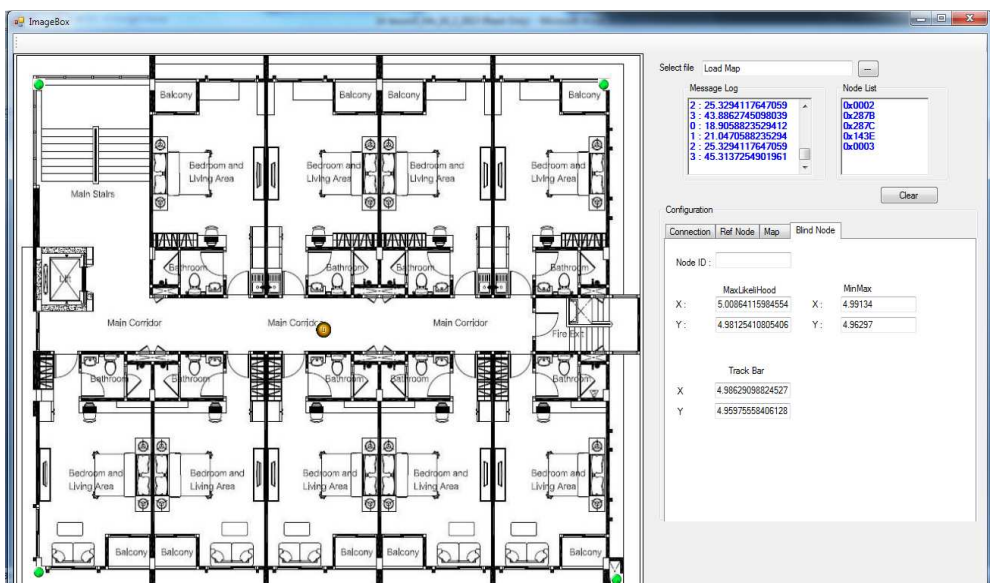
รูปที่ ค-6 แสดงโหมดการปรับค่าตัวแปร A และค่าคงที่ N

7. เลือกการแสดงผลแบ่งออกเป็น
- 7.1 แบบ GUI Map เลือก Algorithm ว่าจะใช้วิธีการแสดงผลแบบใดระหว่าง Maximum Likelihood หรือแบบวิธี Min-Max



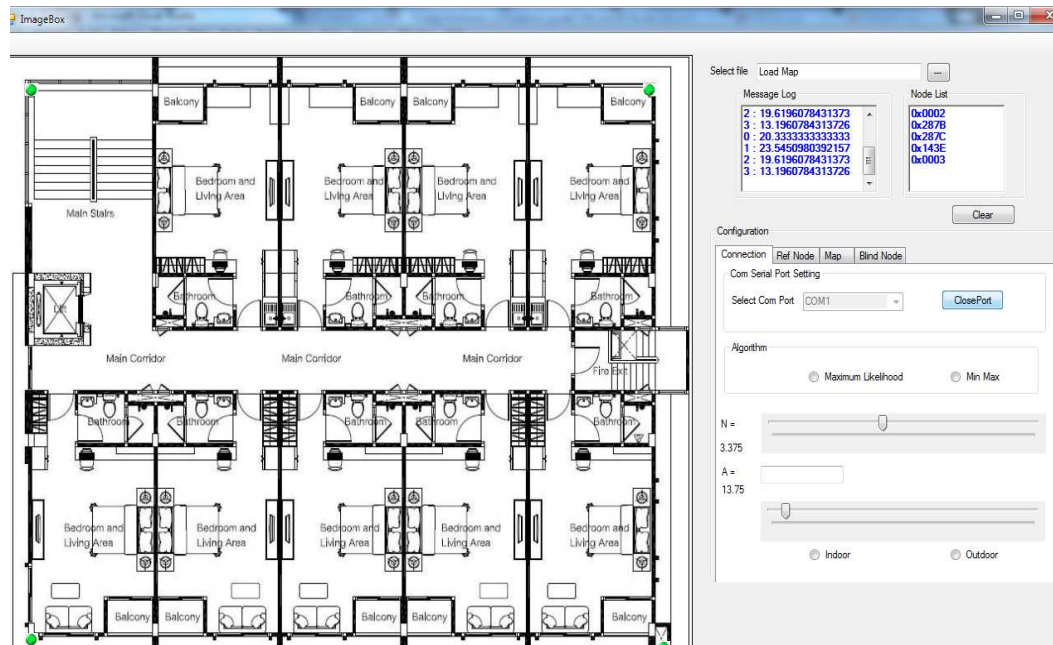
รูปที่ ค-7 แสดงโหมดการแสดงผลแบบ Maximum Likelihood และ Min-Max

7.2 เลือกแสดงตำแหน่งแบบพิกัดให้คลิก Tab : Blind Node



รูปที่ ค-8 แสดงโหมดการตอบของพิกัดของ Maximum Likelihood และ Min-Max

8. หลังจากเลิกใช้งานแล้วให้ปิด Comport ด้วยการเลือกปุ่ม Close Port



รูปที่ ค-9 แสดงโหมดการเลิกใช้ระบบ



## ประวัติผู้เขียนวิทยานิพนธ์

นางสาวจุฑาทิพย์ เจริญชาติศรี เกิดเมื่อวันที่ 28 ธันวาคม พ.ศ. 2526 ที่จังหวัดสมุทรสาคร สำเร็จการศึกษาระดับปริญญาบัณฑิต หลักสูตรบริหารธุรกิจบัณฑิต สาขาวิทยาการจัดการคอมพิวเตอร์ จาก มหาวิทยาลัยราชภัฏสวนสุนันทา เมื่อ พ.ศ. 2545 และเข้าศึกษาต่อในหลักสูตร วิทยาศาสตร์มหาบัณฑิต สาขาวิทยาศาสตร์คอมพิวเตอร์ ณ คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์ มหาวิทยาลัย เมื่อปี พ.ศ. 2552