

ระบบระบุตำแหน่งโดยใช้เครือข่าย Zigbee
Real Time Location System Using Zigbee Network

จักริน กองแก้ว
ธนกร พุทธิรักษา
ปฏิภาณ รัตนพงศ์

ปฏิญานีพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิชาวิศวกรรมเครื่องกล
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2556

ระบบระบุตำแหน่งโดยใช้เครือข่าย Zigbee

Real Time Location System Using Zigbee Network

จักริน	กองแก้ว
ฉนกร	พุทธรักษา
ปฏิภาณ	รัตนพงศ์

ปริญญาานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมเครื่องกล

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2556

Real Time Location System Using Zigbee Network

JAKKARIN KONGKAEW
TANAKORN PUTTARUGSA
PATIPAN RATTANAPONG

A THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENT FOR THE DEGREE OF
BACHELOR OF ENGINEERING IN MECHANICAL ENGINEERING
FACULTY OF ENGINEERING
KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG
2013

ปริญญาานิพนธ์ปีการศึกษา 2556

สาขาวิชาวิศวกรรมเครื่องกล คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง ระบบระบุตำแหน่งโดยใช้เครือข่าย Zigbee

Real Time Location System Using Zigbee Network

ผู้จัดทำ

1. นาย จักริน กองแก้ว รหัสประจำตัว 53010190
2. นาย ธนกร พุทธรักษา รหัสประจำตัว 53010613
3. นาย ปฏิภาณ รัตนพงศ์ รหัสประจำตัว 53010902



อาจารย์ที่ปรึกษา

(อาจารย์ภูดิท ชัยติลภพัฒน์กุล)

ระบบระบุตำแหน่งโดยใช้เครือข่าย Zigbee

นายจักริน	กองแก้ว	53010190
นายธนกร	พุทธรักษา	53010613
นายปฏิภาณ	รัตนพงศ์	53010902

อาจารย์ภูดิท ชัยติลกพัฒนกุล อาจารย์ที่ปรึกษา
ปีการศึกษา 2556

บทคัดย่อ

ในปัจจุบันมีความต้องการที่จะระบุตำแหน่งของมีค่าต่างๆภายในอาคารซึ่งสัญญาณ GPS เข้าไม่ถึงและไม่มีความแม่นยำเพียงพอ จึงได้มีการประยุกต์ใช้สัญญาณต่างๆมาระบุตำแหน่งสิ่งของภายในอาคารเช่น WiFi, RFID, Zigbee งานวิจัยนี้ได้ใช้ Zigbee มาประยุกต์เพื่อใช้ระบุตำแหน่งในอาคารโดยการทำ fingerprint จากความเข้มของสัญญาณระหว่าง Sensor จากการทดลองเราสามารถสรุปได้ว่าสามารถใช้ Zigbee ในการระบุตำแหน่งได้อย่างแม่นยำ แต่มีข้อควรระวังเรื่องสัญญาณรบกวนที่ทำให้การระบุตำแหน่งผิดพลาด

Real Time Location System Using Zigbee Network

JAKKARIN	KONGKAEW	53010190
TANAKORN	PUTTARUGSA	53010613
PATIPAN	RATTANAPONG	53010902
Mr.Phudit	Chaidilokpattanakul	Advisor

Year 2556

ABSTRACT

There is a need to locate valuable objects in the building where GPS signal is weakened and unreliable. Few technologies exist and can be applied to use for this purpose including WiFi, RFID, Zigbee. This research implements Zigbee for Real Time Location System (RTLS) by fingerprinting Relative Signal Strength Indicator (RSSI). From our experiment, we can conclude that Zigbee can be used for RTLS. However, noise from other sources such as WiFi can adversely affect results.

กิตติกรรมประกาศ

ปริญญาบัตรฉบับนี้สำเร็จลุล่วงได้ด้วยดี เพราะด้วยความเอาใจใส่ แนะนำ และช่วยเหลือในด้านต่าง ๆ เสมอมาจาก อาจารย์ภูดิท ชัยติลกพัฒนกุล อาจารย์ที่ปรึกษา อีกทั้งยังเอื้อเฟื้ออุปกรณ์และเครื่องมือ ผู้ทำโครงการรู้สึกซาบซึ้งและขอกราบขอบพระคุณเป็นอย่างสูง

ขอขอบคุณสาขาวิชาวิศวกรรมเครื่องกลที่เอื้อเฟื้อให้ยืมเครื่องมือและอุปกรณ์ที่ใช้ในโครงการนี้ และขอบคุณเพื่อน ๆ นักศึกษาทุกคนที่ให้คำแนะนำอันเป็นประโยชน์ต่อการทำปริญญาบัตรฉบับนี้

และต้องขอขอบพระคุณบุคคลสำคัญที่สุดที่ทำให้ข้าพเจ้ามีวันนี้ ก็คือ บิดา มารดา อันเป็นที่เคารพรักยิ่ง ซึ่งได้เลี้ยงดูผู้เขียนมาเป็นอย่างดี พร้อมทั้งให้โอกาสในการศึกษาอย่างเต็มที่ และยังให้กำลังใจ เอาใจใส่เสมอมาในทุกๆ ด้านอันหาที่เปรียบมิได้ ข้าพเจ้าขอระลึกในพระคุณอันสุดประมาณ และขอกราบขอบพระคุณมา ณ ที่นี้

นายจักริน	กองแก้ว
นายธนกร	พุทธรักษา
นายปฏิภาณ	รัตนพงศ์

สารบัญ

	หน้า
บทคัดย่อภาษาไทย	i
บทคัดย่อภาษาอังกฤษ	ii
กิตติกรรมประกาศ	iii
สารบัญ	iv
สารบัญตาราง	viii
สารบัญรูป	ix
บทที่ 1 บทนำ	1
1.1 ความสำคัญของปริญญาโท	1
1.2 วัตถุประสงค์ของปริญญาโท	1
1.3 ขอบเขตของปริญญาโท	2
1.4 ขั้นตอนการศึกษา	2
1.5 ประโยชน์ที่คาดว่าจะได้รับ	2
บทที่ 2 หลักการและทฤษฎีที่เกี่ยวข้อง	3
2.1 โครงข่ายไร้สายส่วนบุคคล (Wireless Personal Area Network: WPAN)	3
2.2 ZigBee	5
2.3 XBee	7
2.3.1 คุณสมบัติพื้นฐานของ XBee	8
2.3.2 การเลือกใช้ XBee เพื่อความเหมาะสมกับงาน	10
2.3.3 รูปแบบโครงข่ายไร้สายของ Xbee	13
2.4 ไมโครคอนโทรลเลอร์ (Microcontroller)	17
2.5 XBee USB Dongle	20

สารบัญ (ต่อ)

	หน้า
บทที่ 3 ทฤษฎีและหลักการของการหาตำแหน่งภายในอาคาร	21
3.1 กล่าวนำ	21
3.2 การหาตำแหน่งโดยวิธีการเรขาคณิตรูปสามเหลี่ยม (Triangulation)	21
3.2.1 วิธีการหาตำแหน่งด้วยระยะทางแบบ Time-of-Arrival (TOA)	22
3.2.2 วิธีการหาตำแหน่งด้วยมุมแบบ Angle of Arrival (AOA)	25
3.3 วิธีการหาตำแหน่งด้วยการวัดค่าความใกล้เคียงกับจุดอ้างอิง (Proximity)	25
3.4 วิธีการหาตำแหน่งโดยการวิเคราะห์บริเวณเฉพาะ(Scene Analysis/Fingerprint)	25
3.4.1 Offline phase หรือ Training Phase	25
3.4.2 Online phase หรือ Localization Phase	26
3.5 ความผิดพลาดที่อาจเกิดขึ้น	26
3.6 การเขียนภาษา C# สำหรับ Microsoft Visual Studio 2010 เบื้องต้น	27
3.6.1 การสร้างแอปพลิเคชันด้วย MS Visual C#	29
3.6.2 ภาษา C# และขั้นตอนการพัฒนา	29
3.6.3 เครื่องมือสำหรับพัฒนาโปรแกรมด้วย C#	29
3.6.4 โครงสร้างโปรแกรมภาษา C#	31
3.6.5 กฎการตั้งชื่อตัวระบุในภาษา C#	33
3.6.6 ข้อมูลและตัวดำเนินการ	33
3.6.6.1 ชนิดข้อมูล (Data Types) ใน C#	33
3.6.6.2 ตัวแปร (variable)	33
3.6.6.3 ค่าคงที่ (Constants)	34
3.6.6.4 นิพจน์ทางคณิตศาสตร์ (Arithmetic Expressions)	35
3.6.6.5 คำสั่งที่ใช้ในการแสดงผล	36
3.6.6.6 คำสั่งเงื่อนไขและการทำซ้ำ	40

สารบัญ (ต่อ)

	หน้า
บทที่ 4 วิธีการและขั้นตอนการทดลองโครงการ	43
4.1 กล่าวนำ	43
4.2 อุปกรณ์ที่ใช้ในการทดลองโครงการ	43
4.3 ขั้นตอนและรูปแบบวิธีการทดลองโครงการในการหาตำแหน่ง	50
4.3.1 ทำการกำหนดค่าพารามิเตอร์ของ Xbee	50
4.3.2 ทำการศึกษาและเขียนโปรแกรมอินเทอร์เฟซ	51
4.3.3 จัดเตรียมการทดลองและสภาพแวดล้อมในการทดลอง	52
4.3.4 รูปแบบในการทดลองหาตำแหน่ง	55
4.3.5 ขั้นตอนในการทดลองหาตำแหน่ง	55
4.3.5.1 Offline Phase หรือ Training Phase	56
4.3.5.2 Online Phase หรือ Localization Phase	56
4.4 สรุป	66
บทที่ 5 ผลการทดลองแล้วผลการวิเคราะห์การหาตำแหน่งของเป้าหมายภายในอาคาร	67
5.1 กล่าวนำ	67
5.2 ผลการทดลองในพื้นที่ไม่มีสัญญาณ wifi ครอบคลุม	67
5.3 การทดลองในพื้นที่ที่มีสัญญาณwifiครอบคลุม	71
5.4 สรุป	74
บทที่ 6 บทสรุปและข้อเสนอแนะ	75
6.1 กล่าวนำ	75
6.2 บทสรุป	75
6.3 ปัญหาที่พบในการทดลอง	76
6.4 ข้อเสนอแนะ	77
เอกสารอ้างอิง	77
ภาคผนวก	78

สารบัญตาราง

ตารางที่	หน้า
2.1 มาตรฐาน 802.15 ของ WPAN	4
2.2 แสดงการเปรียบเทียบเทคโนโลยีไร้สายในแบบต่างๆ	6
2.3 การทำงานแต่ละขาของ XBee	9
2.3 (ต่อ) การทำงานแต่ละขาของ XBee	10
2.4 แสดงการเปรียบเทียบคุณสมบัติของ Xbee รุ่นต่างๆ	12
3.1 ตัวดำเนินการทางคณิตศาสตร์	36
3.2 ตัวดำเนินการทางตรรกศาสตร์	38
3.3 ตัวดำเนินการทางคณิตศาสตร์	39
3.4 ตัวดำเนินการเปรียบเทียบ	39
4.1 รายละเอียดค่าต่างๆที่ใช้ในการทดลองหาค่าแห่ง	53
5.1 ตำแหน่งที่ได้จากการทดลองในข้อ 1)	67
5.1 (ต่อ) ตำแหน่งที่ได้จากการทดลองในข้อ 1)	68
5.1 (ต่อ) ตำแหน่งที่ได้จากการทดลองในข้อ 1)	69
5.1 (ต่อ) ตำแหน่งที่ได้จากการทดลองในข้อ 1)	70
5.2 พิกัดที่คำนวณได้จากการทดลองโดยใช้	71
5.2 (ต่อ) พิกัดที่คำนวณได้จากการทดลองโดยใช้	72
5.2 (ต่อ) พิกัดที่คำนวณได้จากการทดลองโดยใช้	73
5.2 (ต่อ) พิกัดที่คำนวณได้จากการทดลองโดยใช้	74

สารบัญรูป

รูปที่	หน้า
2.1 ย่านมาตรฐานของ ZigBee	5
2.2 Xbee Stack	6
2.3 ลักษณะการทำงานของ ZigBee	7
2.4 การนับตำแหน่งขาของ Xbee	8
2.5 Xbee แบบ Chip Antenna	11
2.6 Xbee แบบ Wire Antenna	11
2.7 Xbee แบบ UFL Antenna	11
2.8 Xbee แบบ SMA Antenna	12
2.9 กรรเชื่อมต่อแบบ Star (Broadcast)	13
2.10 กรรเชื่อมต่อแบบ Cluster Tree	14
2.11 กรรเชื่อมต่อแบบ Mesh	15
2.12 XBee USB Dongle	20
3.1 แสดงการหาตำแหน่งด้วยการวัดแบบ TOA	22
3.2 แสดงวิธีการหาตำแหน่งของวัตถุ	23
3.3 แสดงวิธีการหาตำแหน่งของวัตถุโดยใช้ความสัมพันธ์ของสามเหลี่ยม	23
3.4 แสดงวิธีการหามุมของวัตถุ.	24
3.5 (a) แสดงวิธีการคำนวณหามุมโดยการใช้ความแตกต่างของ TOA ของเซนเซอร์ แต่ละตัวในโนดหนึ่งๆ : เซนเซอร์รูปร่าง Y (b) AOA สามารถคำนวณได้จาก RSS ratio ระหว่าง Directional antennas	27
3.6 รายการเมนูสำหรับสร้างโปรเจกต์ใหม่	28
3.7 ไดอะล็อกซ์ในสร้างโปรเจกต์ใหม่และการตั้งค่าสำหรับคอนโซลแอปพลิเคชัน	28
3.8 หน้าจอโดยรวมของ MS Visual C# หลังกระบวนการสร้างโปรเจกต์	30
3.9 ตัวอย่างคอนโซลแอปพลิเคชัน (Console Application)	30

สารบัญรูป (ต่อ)

รูปที่	หน้า
3.10 ตัวอย่างวินโดวส์แอปพลิเคชัน (Windows Application)	31
3.11 ส่วนของโปรแกรมหลัก	32
3.12 โปรแกรมแสดงข้อความ	32
3.13 โปรแกรมแสดงข้อความแบบไม่ระบุเนมสเปซ	34
3.14 การระบุชนิดข้อมูลที่จะใช้กับตัวแปร	34
3.15 การกำหนดค่าเริ่มต้นให้กับตัวแปร	34
3.16 การกำหนดชนิดข้อมูลเป็น uint	34
3.17 การประกาศตัวแปรชื่อ salary ที่มีชนิดข้อมูลเป็น long และค่าเริ่มต้นเป็น 30000	35
3.18 การประกาศค่าคงที่	35
3.19 การประกาศข้อมูลเป็นแบบ double และใช้แทนค่า 2.71828	37
3.20 การเรียกใช้งานคำสั่ง Write และ WriteLine โดยระบุเนมสเปซเต็มรูปแบบ	37
3.21 การเรียกใช้งานคำสั่ง Write และ WriteLine โดยใช้คำสั่ง using	38
3.22 การเรียกใช้งานคำสั่ง Write และ WriteLine โดยใช้สตรีนกำหนดรูปแบบ	44
4.1 โปรแกรมที่ใช้ในการลงเฟิร์มแวร์กับ Xbee	44
4.2 โปรแกรมที่ใช้ในการระบุตำแหน่งของ Xbee	45
4.3 อุปกรณ์แปลงพอร์ตยูเอสบีมินิบีเป็นยูเอสบีแบบเอ	45
4.4 อุปกรณ์ยูเอสบีฮับ 4 พอร์ต OKER H-409	46
4.5 ยูเอสบีแบบแจ๊คเอไปเป็นปลั๊กเอ	47
4.6 มินิ Xbee ยูเอสบีตองเกิ้ลวี 2	47
4.7 Xbee Series 1	47
4.9 การกำหนดค่าพารามิเตอร์ส่วนที่ 1	50
4.10 การกำหนดค่าพารามิเตอร์ส่วนที่ 2	51
4.11 หน้าต่างแสดงผลที่สร้างจาก Visual Studio 2010	51

สารบัญรูป (ต่อ)

รูปที่	หน้า
4.12 การแบ่งพื้นที่เพื่อศึกษาบริเวณที่ไม่มีสัญญาณรบกวน	52
4.13 การแบ่งพื้นที่เพื่อศึกษาบริเวณที่มีสัญญาณรบกวน	53
4.14 การแบ่งพื้นที่เพื่อศึกษาบริเวณที่มีสัญญาณรบกวน	54
4.15 แสดงการติดตั้ง Coordinator Xbee จำนวน 4 ตัว	54
4.16 แสดงการเก็บค่าความเข้มของสัญญาณ	55
4.17 แสดงจุดตำแหน่งในการทำ Fingerprint	56
4.18 จำลองการทดลองในพื้นที่ไม่มีสัญญาณรบกวน	57
4.19 การติดตั้งอุปกรณ์	58
4.20 พื้นที่ทดลองขนาด5x5ตารางเมตร	58
4.21 มองจากด้านบน	59
4.22 มองจากด้านข้าง	59
4.23 ตำแหน่งที่สู่มทดลอง	60
4.24 รูปการเก็บค่าพื้นฐานและการทดลอง	61
4.25 หน้าจอแสดงผลการเก็บค่าสัญญาณพื้นฐาน	61
4.26 จำลองการทดลองโดยใช้ 4 Reference node	62
4.27 ติดตั้งอุปกรณ์	63
4.28 พื้นที่ทดลองขนาด 5x5 ตารางเมตร	63
4.29 มองด้านบน	64
4.30 มองด้านข้าง	64
4.31 การเก็บค่าพื้นฐานแต่ละตาราง	65
4.32 การทดลองหาตำแหน่งเป้าหมาย	65
5.1 เปรียบเทียบตำแหน่งจริงกับตำแหน่งที่คำนวณได้	70

บทที่ 1

บทนำ

1.1 ความสำคัญของปริญญานิพนธ์

ในปัจจุบันการค้นหาวัตถุที่มีค่าหรือไม่มีค่าก็ตามเมื่อเกิดมีการสูญหายนั้นเป็นไปได้ยาก จึงทำให้มีการคิดค้นวิธีการระบุตำแหน่งของวัตถุขึ้นมา เนื่องจากระบบระบุตำแหน่งของวัตถุมีความสำคัญเป็นอย่างมากไม่ว่าจะเป็นภาคอุตสาหกรรมหรืออื่นๆ เพื่อที่จะให้เป็นที่ไปตามผู้สนใจต้องการทราบตำแหน่งที่ตั้งของวัตถุ ณ เวลาและที่ตั้งจริงในขณะนั้นซึ่งถ้าหากการระบุตำแหน่งผิดพลาดหรือมีความคลาดเคลื่อนไปอาจจะทำให้เกิดความสูญเสียต่อทรัพยากรที่มีค่า

เทคโนโลยี Zigbee เป็นการสื่อสารที่ออกแบบขึ้นเพื่อใช้ในเครือข่ายเซ็นเซอร์แบบไร้สาย (Wireless Sensor Network) โดยมีการกำหนดมาตรฐานการรับ-ส่งข้อมูลแบบ IEEE 802.15.4 ซึ่งเน้นการสื่อสารแบบประหยัดพลังงาน ความเร็วการรับส่งข้อมูลค่าและมีราคาถูก ช่วยทำให้การสื่อสารข้อมูลเป็นไปด้วยความสะดวกและการติดตั้งระบบสามารถทำได้ง่ายโดยมี โครงการนี้ได้นำความรู้ทางเทคโนโลยี Zigbee มาประยุกต์ใช้ในการระบุตำแหน่งของอุปกรณ์คืออุปกรณ์ End Device Xbee โดยปราศจากการใช้ไมโครคอนโทรลเลอร์ (Microcontroller) ซึ่งทางผู้ศึกษาได้สังเกตเห็นความสำคัญของการศึกษากระบวนการควบคุมจำนวนของ Coordinator Xbee ที่มีจำนวน 3 จุดและ 4 จุด อีกทั้งหาความแตกต่างของค่าความเข้มสัญญาณระหว่างการทดลองในที่ที่มีสัญญาณรบกวนและไม่มีสัญญาณรบกวน โดยควบคุมการแสดงผลออกผ่านทางหน้าจอคอมพิวเตอร์ แล้วค่าข้อมูลดังกล่าวจะแสดงผลออกมาในรูปของจุดแสดงตำแหน่งที่ตั้งของ Object Node

1.2 วัตถุประสงค์ของปริญญานิพนธ์

1. เพื่อศึกษาและพัฒนาระบบการค้นหตำแหน่งของ End Device Xbee และทำการส่งข้อมูลไปยังคอมพิวเตอร์โดยใช้ Zigbee
2. เพื่อทำการศึกษาและทดลองนำทฤษฎี Fingerprint มาประยุกต์ใช้และนำการวัดค่าความเข้มของคู่สัญญาณมาใช้ในการระบุตำแหน่ง
3. เพื่อทำการเปรียบเทียบความแม่นยำในการใช้ Coordinator Xbee จำนวน 3 จุดและ 4 จุด

1.3 ขอบเขตของปริญญาโท

1. ทำการสร้างและติดตั้งระบบของ Xbee เพื่อให้เป็น Star Topology
2. เขียนโปรแกรมอินเตอร์เฟซเพื่อแสดงตำแหน่งของ End Device Xbee ออกมาบนหน้าจอคอมพิวเตอร์โดยใช้ Microsoft Visual Basic 2010
3. คำนวณตำแหน่งจากทฤษฎี Fingerprint กับการบันทึกค่าความเข้มของสัญญาณ ณ จุดต่างๆ

1.4 ขั้นตอนการศึกษา

1. วางแผนงานการทำโครงการ โดยกำหนดระยะเวลาในการดำเนินงาน
2. รวบรวมข้อมูล ทำการศึกษาทฤษฎีและงานวิจัยที่เกี่ยวข้องกับการใช้ Xbee และระบบระบุตำแหน่งแบบไร้สาย
3. จัดทำชุดระบบระบุตำแหน่งโดยใช้ค่าความเข้มของสัญญาณเป็นตัวระบุตำแหน่ง
4. กำหนดรูปแบบการทดลอง เลือกประเภทของซอฟต์แวร์ที่จะใช้เขียนโปรแกรมเพื่อแสดงผล
5. ทำการทดลองตามรูปแบบที่กำหนดไว้ และทำการคำนวณตำแหน่งจากทฤษฎีที่ได้ศึกษามา
6. วิเคราะห์ผลที่ได้จากการทดลองว่าได้ผลมีความถูกต้องแม่นยำหรือมีความคลาดเคลื่อนเท่าใด
7. สรุปผลการทดลองตามวัตถุประสงค์พร้อมข้อเสนอแนะในการทำโครงการต่อไป

1.5 ประโยชน์ที่คาดว่าจะได้รับ

1. มีความรู้ความเข้าใจเกี่ยวกับอุปกรณ์ Xbee
2. สามารถเขียนโปรแกรมเพื่อแสดงตำแหน่งของ End Device Xbee
3. สามารถออกแบบวิธีการทดลองเพื่อทำการเก็บค่าความเข้มของสัญญาณได้
4. ได้อุปกรณ์ระบุตำแหน่งของวัตถุและส่งข้อมูลไปยังคอมพิวเตอร์เพื่อแสดงผล
5. ติดตั้งอุปกรณ์และใช้งานได้จริง

บทที่ 2

หลักการ และทฤษฎีที่เกี่ยวข้อง

2.1 โครงข่ายไร้สายส่วนบุคคล (Wireless Personal Area Network: WPAN)

เทคโนโลยี WPAN ถูกกำหนดให้อยู่ในมาตรฐาน IEEE 802.15 และมีมาตรฐานย่อย IEEE 802.15.1 ที่เป็นมาตรฐานเฉพาะของเทคโนโลยีบลูทูธ (Bluetooth) ซึ่งในขั้นต้นได้เสนอมาตรฐานนี้โดยการร่วมกันของผู้ผลิตจากหลายบริษัท โดยการใช้งานในปัจจุบันนั้นการส่งข้อมูลด้วย Bluetooth สามารถใช้งานแทนการส่งผ่านสาย (Cable) และมีระยะการส่งถึง 20 เมตร ต่อมา มีการเสนอมาตรฐาน IEEE 802.15.3a เป็นมาตรฐานสำหรับระบบ Ultra Wide Band (UWB) ซึ่งเป็นระบบเครือข่ายที่มีอัตราการรับส่งสูงมาก (11Mbps 55Mbps) และมีระยะการรับส่งข้อมูลไกล (ไม่เกิน 20 เมตร) ซึ่งมาตรฐาน IEEE 802.15.3a ยังไม่มีความสมบูรณ์และยังมีการพัฒนาอย่างต่อเนื่อง ส่วนมาตรฐาน IEEE 802.15.4 เป็นมาตรฐานระบบ ZigBee ซึ่งเป็นเทคโนโลยีที่มีความซับซ้อนน้อย ออกแบบมาเพื่อเป็นโครงข่ายไร้สายสำหรับอุปกรณ์ควบคุม ในโรงงานอุตสาหกรรม

เทคโนโลยี WPAN (Wireless Personal Area Network) เริ่มได้รับความสนใจมากขึ้นหลังจากที่เทคโนโลยีก่อนหน้านี้คือ WLAN (Wireless Local Area Network) ได้รับการยอมรับอย่างกว้างขวาง และมีผลิตภัณฑ์ออกมาให้ใช้งานอย่างแพร่หลาย WPAN เป็นเทคโนโลยีในขั้นสุดท้ายของเครือข่ายไร้สายที่จะรองรับผู้ใช้งานโดยตรง และถ้าต้องการที่จะให้ WPAN ได้รับการยอมรับอย่างกว้างขวาง จะต้องสอดคล้องกับความต้องการพื้นฐาน คือ มีราคาถูก ใช้กำลังไฟฟ้าสามารถติดต่อสื่อสารหรือทำงานร่วมกับอุปกรณ์จากผู้ผลิตรายอื่นได้ และมี อัตราเร็วการส่งข้อมูลที่สูง

มาตรฐาน WPAN ขึ้นมา ที่เรียกว่า IEEE 802.15 ซึ่งมาตรฐาน IEEE 802.15 นี้จะกำหนดให้ WPAN เป็นเครือข่ายไร้สายสำหรับใช้งานในระยะสั้นและมีราคาถูก โดยจะทำให้อุปกรณ์ อย่างเช่น เครื่องคอมพิวเตอร์ PC, เครื่องคอมพิวเตอร์ laptop, เครื่องพิมพ์ และเครื่อง PDA สามารถติดต่อสื่อสารกันอย่างไร้สายในช่วงระยะทางสั้นๆ ได้ เป้าหมายของมาตรฐาน 802.15 ก็คือ ต้องการที่จะสร้างกรอบการทำงานของการพัฒนาอุปกรณ์ที่มีราคาถูกและใช้กำลังไฟฟ้าต่ำ สำหรับเชื่อมโยงการสื่อสารแบบไร้สายในระยะสั้น โดยมีจุดมุ่งหมายให้ WPAN เป็นเครือข่ายขนาดเล็กสำหรับใช้งานในบ้านหรือสำนักงาน และมี

จำนวนโหนดไม่เกิน 8-16 โหนด มาตรฐาน WPAN นี้จะเป็นมาตรฐานเครือข่ายไร้สายของอุปกรณ์ประเภทที่เคลื่อนย้ายได้ (portable) และอุปกรณ์ประเภทโมบายล์ (mobile) ได้ถูกจัดแบ่งออกเป็นหลายกลุ่มด้วยกัน คือ

- 802.15.1 (Standard) ทำการพัฒนาข้อกำหนดมาตรฐาน Bluetooth สำหรับ WPAN
- 802.15.2 (Recommended Practice) เสนอแนะประเด็นสำคัญที่จะต้องพิจารณาเมื่อมีการนำ WLAN (802.11) และ WPAN (802.15) มาใช้งานในที่เดียวกัน
- 802.15.3 (Standard) ทำการพัฒนามาตรฐานของ High-Rate (HR) WPAN ซึ่งจะมีอัตราการเร็วการส่งข้อมูลมากกว่า 20 Mbps
- 802.15.4 (Standard) ทำการพัฒนามาตรฐานของ Low-Rate (LR) WPAN ซึ่งจะมีอัตราการเร็วการส่งข้อมูลระหว่าง 2 - 250 kbps

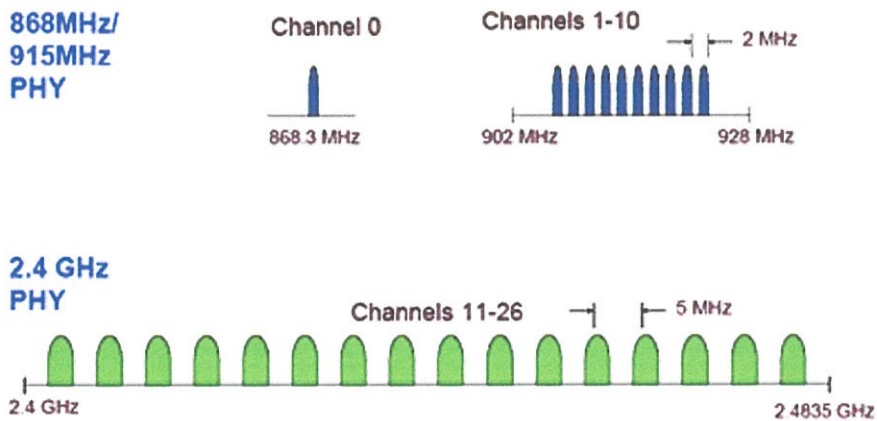
ตารางที่ 2.1 มาตรฐาน 802.15 ของ WPAN

	Data Rate	Range	Configuration	Other Features
802.15.1 (Bluetooth)	1 Mbps	10M(Class 3) 100M(Class 1)	8 active device Piconet/Scatternet	Authentication Encryption, Voice
802.15.3 High Rate	22,33,44,55 Mbps	10M	256 active device Piconet/Scatternet	FCC part 15.249 Qos, Fast Join Multimedia
802.15.4 Low Rate (ZigBee)	Up to 250 kbps	10M nomial 1M-100M based on settings	Master/Slave (256 Devices or more) Peer to Peer	Battery Life: Multi-month to infinite
802.15.2 Coexistence	Develop a Coexistence Model and Mechanisms Document as a Recommended Practice			

2.2 ZigBee

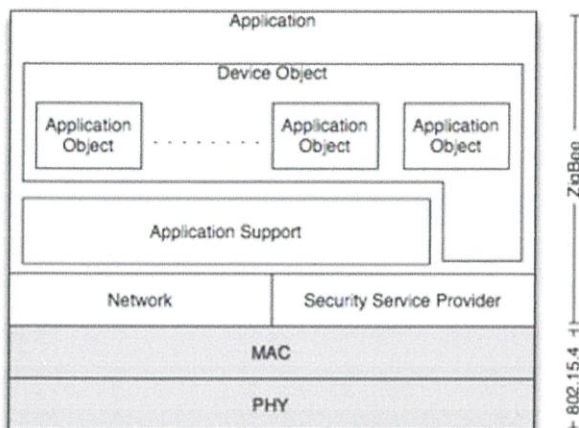
ZigBee มาตรฐานสากล กำหนดโดย ZigBee Alliance เป็น การสื่อสารแบบไร้สาย ที่มีอัตรา การรับส่งข้อมูลต่ำ ใช้พลังงานต่ำ ราคาถูก จุดประสงค์ก็เพื่อให้สามารถสร้างระบบที่เรียกว่า Wireless Sensor Network ได้ ซึ่งระบบนี้ จะสามารถทำงาน ในร่ม กลางแจ้ง หนแดด หนฝน และอยู่ได้ด้วย แบตเตอรี่ก้อนเล็ก (เช่นถ่าน AA 2 ก้อน) นานเป็นเดือนเป็นปีเหมาะสมใช้งานกับพวก monitoring ต่างๆ ZiBee กำหนด ย่านความถี่ใช้งานตามมาตรฐานไว้ 3 ย่านความถี่คือ

- 1.ย่านความถี่ 2.4 GHz มี 16 ช่องสัญญาณ อัตรารับส่งข้อมูล 250 Kbps
- 2.ย่านความถี่ 915 MHz มี 10 ช่องสัญญาณ อัตรารับส่งข้อมูล 40 Kbps
- 3.ย่านความถี่ 868 MHz มี 1 ช่องสัญญาณ อัตรารับส่งข้อมูล 20 Kbps



รูปที่ 2.1 ย่านมาตรฐานของ ZigBee

ระหว่าง Zigbee กับ Wifi นั้น ย่านความถี่เหมือนกัน จะสามารถสื่อสารกันได้หรือไม่ คำตอบคือ สื่อสารกันไม่ได้ เพราะทางกายภาพ ถึงแม้จะเป็นย่านความถี่เดียวกัน แต่ Protocol ที่ใช้สื่อสารกันนั้นไม่เหมือนกัน ZigBee นำ Physical Layer และ MAC Layer ของ IEEE 802.15.4 ซึ่งเป็นมาตรฐานการ กำหนดการสื่อสารไร้สายแบบ WPAN (Wireless Personal Area Network) มาทำงานใน Layer ที่ต่ำกว่า (2 Layer ล่างสุด) เช่น เรื่องของ ระดับกำลังสัญญาณ, Link Quality, Access control, Security ฯลฯ แต่ใน Layer ถัดไปจะเป็นรูปแบบของ Zigbee



รูปที่ 2.2 Xbee Stack

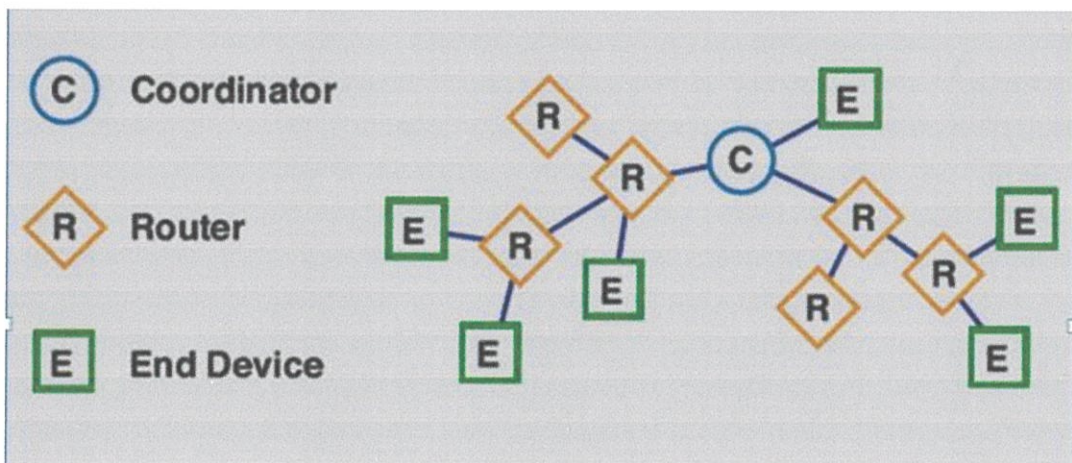
จากที่กล่าวมา ZigBee จะสามารถสร้างเป็นเครือข่ายได้เพราะอิงมาตรฐานตาม IEEE 802.15.4 และมีการจัดการในแบบของ Zigbee ใน Layer ถัดไป ทั้งนี้ IEEE 802.15.4 แบ่งชนิดอุปกรณ์ในเครือข่ายออกเป็น 2 ประเภท คือ FFD (Full Function Device) ซึ่งหมายถึงอุปกรณ์ที่สามารถทำงานได้ทุกอย่างในเครือข่าย และ RFD (Reduce Function Device) ซึ่งหมายถึงอุปกรณ์ที่ลดความสามารถการทำงาน

ตารางที่ 2.2 แสดงการเปรียบเทียบเทคโนโลยีไร้สายในแบบต่างๆ

standard	Zigbee (802.15.4)	Wi-Fi (802.11b)	Bluetooth(802.15.1)
Transmission Range (meters)	1 - 100	1 - 100	1 - 10
Battery Life (days)	100 - 1,000	0.5-5.0	
Network Size (# of nodes)	>64000	32	7
Application	Monitoring&Control	Web,email	Cable Replacement
Stack Size (KB)	4 - 32	1000	250
Throughput (kb/s)	20 - 250	11,000	720

ZigBee ได้แบ่งตามลักษณะการทำงาน 3 แบบ คือ

1. Coordinator มีหน้าที่สร้างการสื่อสาร เชื่อมโยงเครือข่าย ระหว่าง End Device กับ Router หรือ Coordinator กับ Coordinator ด้วยกัน หรือ Coordinator กับ Router กำหนด address ให้กับ device ที่อยู่ใว้ในวงเครือข่าย ไม่ให้ซ้ำกัน จัดการเรื่องการ Routing เส้นทาง ซึ่งเทียบได้กับ FFD
2. End Device เป็นอุปกรณ์ปลายทางสุด ซึ่งจะรับสัญญาณจาก Sensor ที่ปลายทาง โดยใช้พลังงานต่ำในการทำงาน เทียบได้กับ RFD หรือ FFD บางกรณี ขึ้นอยู่กับ sensor ที่ใช้
3. Router มีหน้าที่ รับส่งข้อมูล ในเส้นทางต่าง ๆ ของเครือข่าย ซึ่งเทียบได้กับ FFD



รูปที่ 2.3 ลักษณะการทำงานของ ZigBee

2.3 XBee

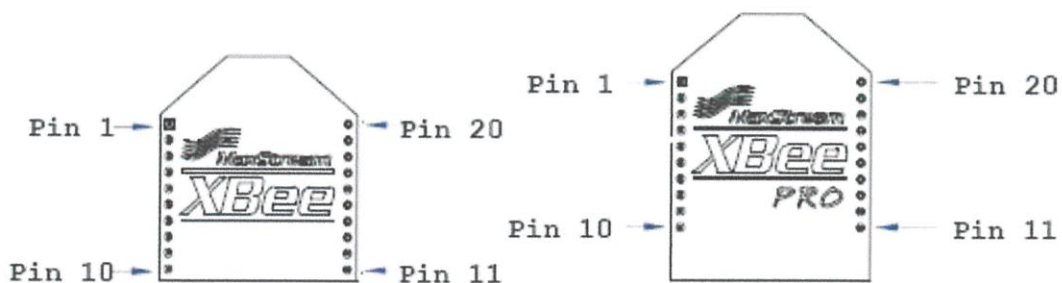
XBee เป็นอุปกรณ์ที่มี Microcontroller และ RF IC อยู่ภายใน (อ้างอิงได้จาก Schematic) ทำหน้าที่เป็น อุปกรณ์ transceiver (อุปกรณ์รับ-ส่งสัญญาณ) แบบ แบบ Half Duplex ย่านความถี่ 2.4 GHz มีการจัดการโดยใช้พลังงานต่ำ ใช้งานง่าย มี interface ที่ใช้รับและส่งข้อมูลกับ Xbee เป็น UART (TTL) ซึ่งสำหรับทางด้านไมโครคอนโทรลเลอร์ เรานำขาที่ใช้ติดต่อสื่อสาร UART ของ Xbee ต่อเข้ากับ UARTของไมโครคอนโทรลเลอร์ได้เลย XBee สามารถใช้งานตามมาตรฐาน ZigBee ได้ โดยที่ไม่ต้องเขียนโปรแกรมสร้าง เครือข่าย ZigBee เลย เพราะว่าทางผู้ผลิตได้จัดทำ firmware ที่จะโหลดเข้าไปในตัว XBee ให้เราสามารถ set parameter ผ่าน software interface (X-CTU หรือโปรแกรมที่เขียนขึ้นเอง) ,

ผ่านทาง At command (เหมือนกับการควบคุม GSM Module) โดยใช้ Hyper terminal หรือ ผ่านทาง การรับส่งข้อมูลด้วยไมโครคอนโทรลเลอร์ ได้อย่างง่ายดาย โดยเมื่อ set XBee ให้ทำงานเป็นอุปกรณ์ในเครือข่าย Zigbee แล้ว เราจะเรียก XBee แต่ละตัวว่าเป็น Node

Firmware ที่ใช้กับ XBee จะใช้โหลดผ่านโปรแกรม X-CTU ซึ่งจะกล่าวถึงวิธีการใช้งานในบทความถัดไป (การใช้งาน XBee เบื้องต้น) ทั้งนี้ XBee แต่ละรุ่น จะสามารถ Setting Function การใช้งานได้มากมาย ทำให้ Firmware ที่จะต้องโหลดเข้าไบนั้น มีมากมายหลายแบบ เราต้องเลือกให้เหมาะสมกับการใช้งานเอง

2.3.1 คุณสมบัติพื้นฐานของ XBee

1. ทำงานในช่วงความถี่ 2.4 GHz
2. มีสายอากาศให้เลือกใช้หลายแบบ คือ แบบ Chip Ant, Whip Ant, UFL con, RPSMA con โดย 2 แบบหลัง เราต้องไปหาเสาอากาศย่าน 2.4 GHz ที่เป็น connector แบบ UFL หรือ SMA
3. ไฟเลี้ยงอยู่ที่ 2.8-3.4 V
4. กระแสไฟตกร้อยกว่า 10 uA
5. มี อัตราการรับส่งข้อมูล อยู่ที่ 250 Kbps (เป็นส่วนของ สัญญาณที่ส่งผ่านอากาศ)
6. มีอัตราการส่งข้อมูลอินเตอร์เฟสแบบอนุกรมอยู่ระหว่าง 1200 – 115200 Bps (เป็นส่วนที่ติดต่อกับไมโครคอนโทรลเลอร์)
7. เป็นการกระจายคลื่นชนิด DSSS (Direct Sequence Spread Spectrum)
8. การกำหนดที่อยู่ มีลำดับลักษณะคือ กำหนด PAN ID สำหรับเครือข่ายหนึ่งๆ , กำหนด =jv ช่อง และ กำหนดที่อยู่ของแต่ละตัว



รูปที่ 2.4 การนับตำแหน่งขาของ Xbee

ตารางที่ 2.3 การทำงานแต่ละขาของ XBee

ขาที่	ชื่อขา	การทำงาน
1	Vcc	ขาต่อไฟเลี้ยง +3.3V
2	DOUT	ขาเอาต์พุตส่งข้อมูลอนุกรม
3	DIN	ขาอินพุตรับข้อมูลอนุกรม
4	DO8	ขาเอาต์พุตดิจิตอลช่อง 8
5	RESET	ขารีเซ็ตหลัก (แอกทีฟ 0)
6	PWM0/RSSI	ขาเอาต์พุต PWM ช่อง 0 และ ขาเอาต์พุตแสดงความแรงของการรับสัญญาณ
7	PWMI	ขาเอาต์พุต PWM ช่อง 1
8	NC	ไม่ใช้งาน
9	DTR/SLEEP_RQ/DI8	ขาอินพุตรับสัญญาณให้หยุดทำงานเข้าสู่โหมดสลีปหรือ เป็นขาอินพุตดิจิตอลช่อง 8
10	GND	ขาต่อกราวด์
11	AD4/DIO4	ขาอินพุตอนาล็อก 4 หรือ ขาเอาต์พุตดิจิตอล 4
12	CTS/DIO7	อินพุตรับสัญญาณแจ้งการส่งข้อมูลของโฮสต์ (Clear-To-Send) ใช้ในการควบคุมจังหวะการรับส่งข้อมูล หรือเป็นขาอินพุต เอาต์พุตดิจิตอล 7
13	ON/SLEEP	ขาแสดงสถานะการทำงาน 1 : อยู่ในโหมดทำงานปกติ 2 : อยู่ในโหมดสลีป
14	VRE	ขาต่อแรงดันอ้างอิงสำหรับโมดูลแปลงสัญญาณอนาล็อก เป็นดิจิตอล
15	Associated/AD5/DIO5	ขาแสดงสถานะการเชื่อมต่อหรือขาอินพุตอนาล็อก 5 หรือขาอินพุตเอาต์พุตดิจิตอล 5
16	RTS/AD6/DIO6	ขาเอาต์พุตแจ้งความพร้อมในการส่งข้อมูล (Ready-To-Send) ใช้ควบคุมจังหวะการรับส่งข้อมูลหรือเป็นขา อินพุตอนาล็อก 6 หรือเป็นขาอินพุตเอาต์พุตดิจิตอล 6
17	AD3/DIO3	ขาอินพุตอนาล็อก 3 หรือขาอินพุตเอาต์พุตดิจิตอล 3
18	AD2/DIO2	ขาอินพุตอนาล็อก 2 หรือขาอินพุตเอาต์พุตดิจิตอล 2

ตารางที่ 2.3 (ต่อ) การทำงานแต่ละขาของ XBee

19	AD1/DIO1	ขาอินพุตนาฬิกา 1 หรือขาอินพุตเอาต์พุตดิจิทัล 1
20	AD0/DIO0	ขาอินพุตนาฬิกา 0 หรือขาอินพุตเอาต์พุตดิจิทัล 0

2.3.2 การเลือกใช้ XBee เพื่อความเหมาะสมกับงาน

1. การเลือก Series

ข้อแตกต่างระหว่าง Series1 และ Series2 คือ Series 1 ทำ Mesh Network ไม่ได้ แต่ Series2 ทำ Mesh Network ดังนั้นการจะเลือกซื้อในขั้นแรก ให้พิจารณาลักษณะงานที่ได้ออกแบบไว้ยกตัวอย่างเช่น ระบบ Wireless Sensor Network ในเกษตรกรรม ในพื้นที่กว้าง ๆ และมีการวัดค่า Sensor หลายจุด และมีความซับซ้อนของการรับส่งข้อมูล ควรพิจารณาว่าจะใช้ Mesh Network อาจจะเหมาะกว่า Star Network ระบบ Wireless Sensor ในโรงงาน 3 จุดไม่จำเป็นต้องใช้ Mesh Network ก็ได้ สามารถรับส่งค่าด้วย Star Network

2.การเลือกระยะการรับส่งสัญญาณ

XBee จะแยกรุ่นสำหรับเรื่อง ระยะการรับส่งอย่างชัดเจน ด้วยคำว่า "PRO" โดยรุ่นระยะสั้น กำลังการส่ง 1-2 mW จะมีระยะรับส่งประมาณ 100-120 เมตร ส่วนรุ่นระยะไกล "PRO" กำลังส่งจะอยู่ในช่วง 50-60 mW โดยจะมีระยะประมาณ 1500 เมตร ทั้งนี้ขึ้นอยู่กับ สภาพแวดล้อมของระบบ และ สายอากาศที่ใช้ ครับ เนื่องจากเป็นความถี่ 2.4 Ghz ซึ่งเป็นย่านความถี่สูง อัตราการลดทอนสัญญาณจะสูง และ สิ่งกีดขวางจะมีผลอย่างมาก

3.เลือกแบบของสายอากาศ

XBee ที่วางจำหน่าย จะมี 4 แบบ คือ Wire, Chip, UFL และ SMA Chip Antenna เหมาะกับการใช้งานใน โครงการที่ต้องการขนาดเล็ก เพราะการใช้สายอากาศแบบนี้ สายอากาศไม่เกะกะ นำไปใส่กล่องได้ แต่ได้เฉพาะกล่องพลาสติก ไม่สามารถใส่กล่องเหล็กได้ เนื่องจากใส่กล่องเหล็กสัญญาณจะไม่สามารถส่งออกมานอกกล่องเหล็กได้ หากต้องใช้กล่องเหล็ก ควรเลือกใช้สายอากาศที่ต่อออกมานอกกล่องเหล็ก



รูปที่ 2.5 Xbee แบบ Chip Antenna

Wire Antenna ระยะและความเสถียรจะได้ตาม Spec และด้วยสายอากาศที่ยื่นออกมาลักษณะนี้บางที่ผู้ใช้อาจจะรู้สึกเกะกะ ทำให้ใส่กล่องที่ออกแบบมาไม่ได้แต่ถ้าใช้งานแบบทั่วไปควรเลือกใช้สายอากาศแบบนี้



รูปที่ 2.6 Xbee แบบ Wire Antenna

UFL Antenna ระยะและความเสถียร จะได้ตาม Spec เหมาะกับ งานที่ออกแบบใส่ในกล่อง และต้องการให้สายอากาศยื่นออกมานอกกล่อง และเนื่องจากการที่ต้องต่อสาย UFL to SMA ออกมาเพิ่มเติม ตรงจุดนี้จะทำให้เกิดการลดทอนสัญญาณบ้าง แต่ก็จะมีการขยายสัญญาณที่สายอากาศอีกที จึงต้องไปพิจารณาอัตราขยายที่สายอากาศต่อด้วยครับ (อัตราขยาย เรียกว่า Gain มีหน่วยเป็น dB หรือ dBi)



รูปที่ 2.7 Xbee แบบ UFL Antenna

SMA Antenna ระยะและความเสถียร จะได้ตามสเปคดีที่สุด ต่อใช้งานร่วมกับสายอากาศ จะมีการขยายสัญญาณที่สายอากาศอีกที ในการต่อใช้งานจริง การออกแบบใส่กล่อง จะต้องออกแบบให้มีตำแหน่งของ Xbee ให้ใกล้กับรูเจาะกล่องครับ



รูปที่ 2.8 Xbee แบบ SMA Antenna

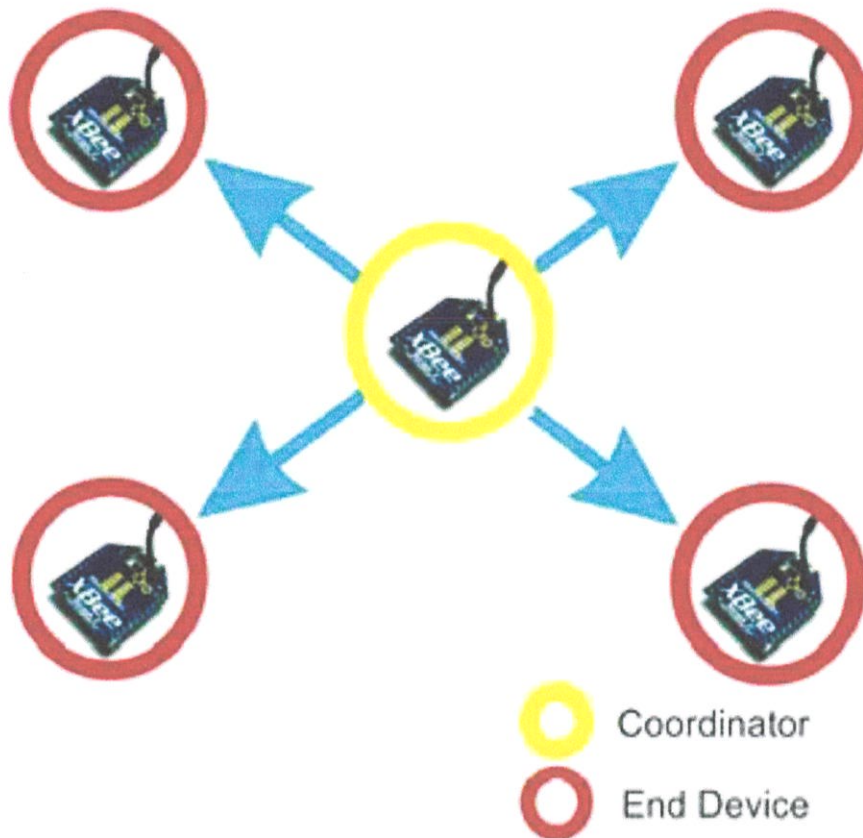
ตารางที่ 2.4 แสดงการเปรียบเทียบคุณสมบัติของ Xbee รุ่นต่างๆ

Feature	Series1	Series2	Series1 Pro	Series 2 Pro
Power Input	3.3V @ 50mA	3.3V @ 40mA	3.3V @ 215mA	3.3V @ 295mA
Max data rate(Air)	250 kbps	250 kbps	250 kbps	250 kbps
Power Output	1mW output (+0dBm)	2mW output (+3dBm)	60mW output (+18dBm)	50mW output (+17dBm)
Distance	300ft(100m)range	400ft(120m)range	1mile(1500m)range	1mile(1600m)range
Antenna	Wire,chip,UFL, SMA	Wire,chip,UFL,SMA	Wire,chip,UFL,SMA	Wire,chip,UFL, SMA
Peripheral	6 10-bit ADC input pins 8 digital IO pins	6 10-bit ADC input pins 8 digital IO pins	6 10-bit ADC input pins 8 digital IO pins	6 10-bit ADC input pins 8 digital IO pins
Upgrade Firmware	Local	over-air configuration(ZB)	Local	over-air configuration (ZB)
Network	Point to point and multi-point networks	Point to point / multi-point /Mesh networks	Point to point and multi-point networks	Point to point / multi-point /Mesh networks

2.3.3 รูปแบบโครงข่ายไร้สายของ Xbee

1. Star (Broadcast)

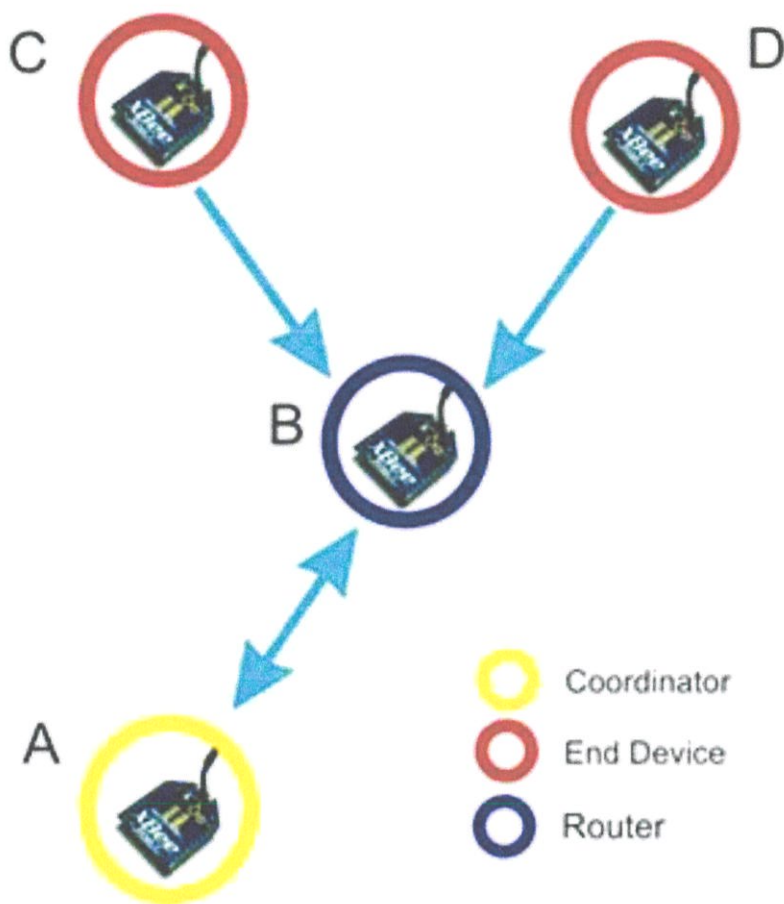
การเชื่อมต่อแบบ Star หรือแบบ Broadcast เป็นการรับส่งข้อมูลแบบ ไม่เฉพาะเจาะจงจุดหมายปลายทาง หรือ XBee ทุกตัวที่อยู่ในระบบเครือข่ายเดียวกันสามารถ รับข้อมูลทุกข้อมูลได้ทุกตัว



รูปที่ 2.9 การเชื่อมต่อแบบ Star (Broadcast)

2.ClusterTree

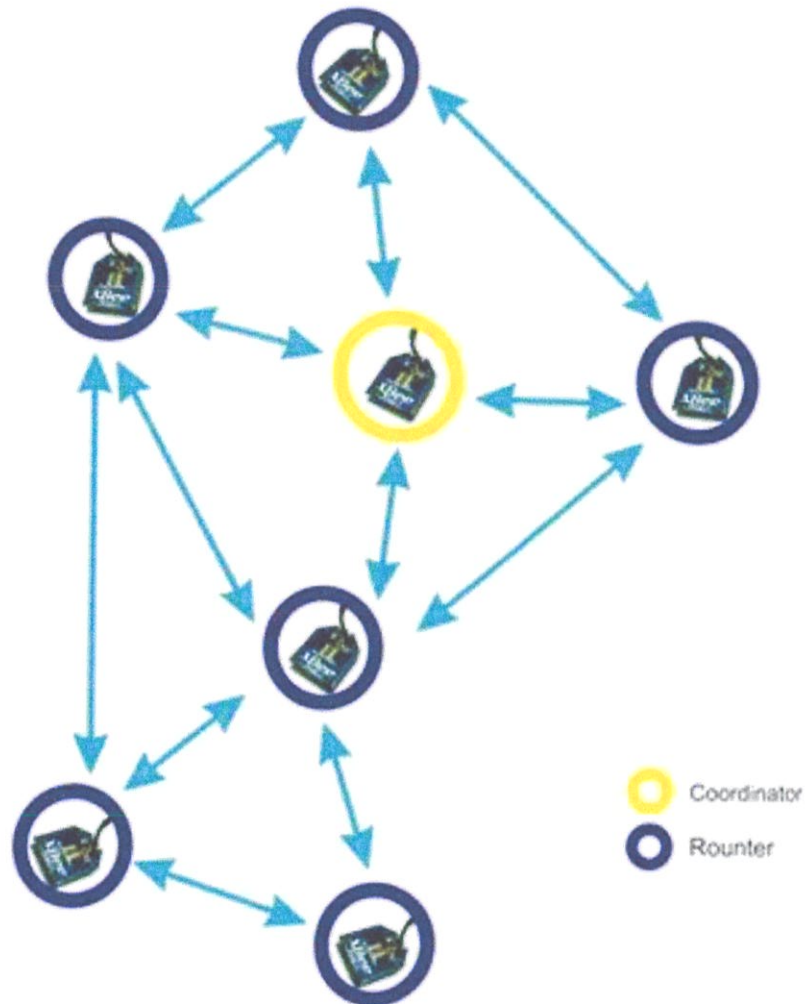
เป็นการ รับส่งข้อมูล แบบ ส่งผ่าน เช่น A ต้องการติดต่อ กับ C แต่ C อยู่ไกลจาก A จน A ไม่สามารถ ติดต่อกับ C ได้ แต่พอดีมี B ที่อยู่ระหว่าง A กับ C ดังนั้น Cluster Tree จะใช้ B เป็นเหมือนตัวกลาง เชื่อมการติดต่อ (Repeater) ระหว่าง A กับ C



รูปที่ 2.10 การเชื่อมต่อแบบ Cluster Tree

3.Mesh

การเชื่อมต่อเครือข่ายแบบ Mesh เป็นโครงข่ายที่มีประสิทธิภาพสูงเนื่องจากข้อมูลสามารถส่งไปถึงเป้าหมายได้หลายทาง ทำให้ ระบบนี้สามารถรับส่งข้อมูลไปยังจุดหมายปลายทางได้ แม้จะเกิดความเสียหายของระบบในบางส่วนก็ตาม (ขึ้นอยู่กับการออกแบบระบบของผู้ใช้ด้วย) ระบบนี้จึงเป็นระบบที่ได้รับความนิยมเป็นอย่างมาก



รูปที่ 2.11 การเชื่อมต่อแบบ Mesh

Xbee Association ในเครือข่าย Zigbee ต้องมีการทำงานในโหมดประหยัดพลังงาน ในช่วงเวลาที่ไม่มีการทำงานและรับส่งข้อมูลดังนั้นตัว Xbee จึงมี Parameter ที่จะกำหนดการทำงานสำหรับ Sleep mode อยู่ (Parameter A1,A2,SP,ST)

Xbee Addressing ตัว Xbee จะสามารถกำหนดค่าประจำตัวอ้างอิงของมัน (Address) 2 แบบ คือ แบบ 16 bit address และ 64 bit address ปกติแล้ว Xbee ทุกตัวจะถูกกำหนดค่ามาจากโรงงาน เป็น Address 64 bit อยู่แล้ว ซึ่งจะสามารถอ่านค่าได้จาก parameter SH+SL การใช้งาน Address 64 bit สามารถทำได้โดยกำหนด parameter MY ให้มีค่า 0xFFFF หรือ 0xFFFE ส่วนการกำหนด 16 bit address นั้นทำได้โดย กำหนด parameter MY ให้มีค่าน้อยกว่า 0xFFFE โดยจะเรียกเป็น mode การทำงาน 2 ประเภทคือ

1. Unicast Mode คือ การรับส่งข้อมูล โดยอาศัยหลักการ Acknowledgement คือหากทางด้านส่งนั้น ส่งข้อมูลไป แต่ไม่ได้รับการ Ack ตอบกลับจากตัวรับ ก็จะทำให้การส่งข้อมูลใหม่

2. Broadcast Mode คือการส่งข้อมูลไปยังปลายทางให้ได้รับข้อมูลทุกตัว

Xbee Operation Mode

Xbee จะสามารถแบ่งช่วงการทำงานได้เป็น 5 แบบ คือ

1. Idle Mode โหมดนี้ จะเป็นโหมดที่ไม่ได้รับส่งข้อมูล ตัว Xbee เตรียมที่จะทำงานในโหมดอื่น ๆ ต่อไปทันที หากมีเงื่อนไขบางอย่าง

2. Transmit Mode คือช่วงที่ Xbee ส่งข้อมูล โดยจะแบ่งลักษณะการทำงานย่อยออกเป็น direct กับแบบ Indirect

3. Receive Mode คือช่วงที่ Xbee รับข้อมูล การกำหนด Address ต้นทางและปลายทาง, clear Channel Assessment และ การตอบรับ Acknowledgement

4. Sleep Mode คือ ช่วงที่ Xbee อยู่ในสถานการณ์ทำงานพลังงานต่ำที่สุด เมื่อไม่มีการใช้งาน

5. Command Mode คือ เป็นส่วนการปรับ parameter ของ Xbee ซึ่งจะมีการกำหนด 2 แบบคือ แบบ AT command กับแบบ API Command

2.4 ไมโครคอนโทรลเลอร์ (Microcontroller)

ไมโครคอนโทรลเลอร์ เป็นอุปกรณ์ไอซี (IC: Integrated Circuit) ที่สามารถโปรแกรมการทำงานได้ซับซ้อน สามารถรับข้อมูลในรูปสัญญาณดิจิทัลเข้าไปทำการประมวลผลแล้วส่งผลลัพธ์ข้อมูลดิจิทัลออกมาเพื่อนำไปใช้งานตามที่ต้องการได้ ไมโครคอนโทรลเลอร์ภายในชิพจะมีหน่วยความจำ, Port อยู่ในชิพเพียงตัวเดียวซึ่งอาจจะเรียกได้ว่าเป็นคอมพิวเตอร์ชิพเดี่ยว ตัวของไมโครคอนโทรลเลอร์เป็นไมโครโปรเซสเซอร์ชนิดหนึ่ง เช่นเดียวกับหน่วยประมวลผลกลาง (CPU: Central Processing Unit) ที่ใช้ในคอมพิวเตอร์ แต่ได้รับการพัฒนาแยกออกมาภายหลังเพื่อนำไปใช้ในวงจรทางด้านงานควบคุม คือ แทนที่ในการใช้งานจะต้องต่อวงจรภายนอกต่างๆ เพิ่มเติมเช่นเดียวกับไมโครโปรเซสเซอร์ ก็จะทำการรวมวงจรที่จำเป็น เช่น หน่วยความจำ, ส่วนอินพุท/เอาต์พุท บางส่วนเข้าไปในตัว ไอซีเดียวกัน และเพิ่มวงจรบางอย่างเข้าไปด้วยเพื่อให้มีความสามารถเหมาะสมกับการใช้งานควบคุม เช่น วงจรตั้งเวลา, วงจรการสื่อสารอนุกรม วงจรแปลงสัญญาณอนาลอกเป็นดิจิทัล เป็นต้น

โครงสร้างโดยทั่วไปของไมโครคอนโทรลเลอร์นั้นสามารถแบ่งออกมาได้เป็น 5 ส่วนใหญ่ๆดังต่อไปนี้

1. หน่วยประมวลผลกลางหรือซีพียู (CPU: Central Processing Unit) ทำหน้าที่เป็นศูนย์กลางควบคุมการทำงานของระบบคอมพิวเตอร์ทั้งหมดโดยนำข้อมูลจากอุปกรณ์รับข้อมูลมาทำงาน ประมวลผลข้อมูลตามคำสั่งของโปรแกรม และส่งผลลัพธ์ออกไปหน่วยแสดงผล

2. หน่วยความจำ (Memory) สามารถแบ่งออกเป็น ส่วน คือ หน่วยความจำที่มีไว้สำหรับเก็บโปรแกรมหลัก (Program Memory) เปรียบเสมือนฮาร์ดดิสก์ของ เครื่องคอมพิวเตอร์ตั้งโต๊ะ คือข้อมูลใดๆ ที่ถูกเก็บไว้ในนี้จะไม่สูญหายไปแม้ไม่มีไฟเลี้ยง อีกส่วนหนึ่งคือหน่วยความจำข้อมูล (Data Memory) ใช้เป็นเหมือนกระดานขดในการคำนวณของซีพียู และเป็นที่พักข้อมูลชั่วคราวขณะทำงาน แต่หากไม่มีไฟเลี้ยง ข้อมูลก็จะหายไปคล้ายกับหน่วยความจำแรม (RAM) ในเครื่องคอมพิวเตอร์ทั่วๆ ไป แต่สำหรับไมโครคอนโทรลเลอร์สมัยใหม่ หน่วยความจำข้อมูลจะมีทั้งที่เป็นหน่วยความจำแรม ซึ่งข้อมูลจะหายไปเมื่อไม่มีไฟเลี้ยง และเป็นอีอีพรอม (EEPROM : Erasable Electrically Read-Only Memory) ซึ่งสามารถเก็บข้อมูลได้แม้ไม่มีไฟเลี้ยง

3. ส่วนติดต่อกับอุปกรณ์ภายนอก หรือพอร์ต (Port) มี 2 ลักษณะคือ พอร์ตอินพุต (Input Port) และพอร์ตส่งสัญญาณหรือพอร์ตเอาต์พุต (Output Port) ส่วนนี้จะใช้ในการเชื่อมต่อกับอุปกรณ์ภายนอก

ถือว่าเป็นส่วนที่สำคัญมาก ใช้ร่วมกันระหว่างพอร์ตอินพุต เพื่อรับสัญญาณ อาจจะใช้การกดสวิตช์ เพื่อนำไปประมวลผลและส่งไปพอร์ตเอาต์พุต เพื่อแสดงผลเช่น การติดสว่างของหลอดไฟ เป็นต้น

4. ช่องทางเดินของสัญญาณ หรือบัส (BUS) คือเส้นทางการแลกเปลี่ยนสัญญาณข้อมูลระหว่างซีพียูหน่วยความจำและพอร์ตเป็นลักษณะของสายสัญญาณจำนวนมากอยู่ในตัวไมโครคอนโทรลเลอร์ โดยแบ่งเป็นบัสข้อมูล (Data Bus) , บัสแอดเดรส (Address Bus) และบัสควบคุม (Control Bus)

5. วงจรกำเนิดสัญญาณนาฬิกา นับเป็นส่วนประกอบที่สำคัญมากอีกส่วนหนึ่งเนื่องจากการทำงานที่เกิดขึ้นในตัวไมโครคอนโทรลเลอร์จะขึ้นอยู่กับข้อกำหนดจังหวะหากสัญญาณนาฬิกาที่มีความถี่สูง จังหวะการทำงานก็จะสามารถทำได้ถี่ขึ้นส่งผลให้ไมโครคอนโทรลเลอร์ดังนั้นมีความเร็วในการประมวลผลสูงตามไปด้วย

นอกจากนี้ยังมีส่วนพิเศษอื่นๆจะขึ้นอยู่กับกระบวนการผลิตของแต่ละบริษัทที่จะผลิตขึ้นมาใส่คุณสมบัติพิเศษลงไปเช่น

- ADC (Analog to Digital) ส่วนภาครับสัญญาณอนาล็อกแปลงไปเป็นสัญญาณดิจิทัล
- DAC (Digital to Analog) ส่วนภาคส่งสัญญาณดิจิทัลแปลงไปเป็นสัญญาณอนาล็อก
- I2C (Inter Integrate Circuit Bus) เป็นการสื่อสารอนุกรม แบบซิงโครนัส (Synchronous) เพื่อใช้ติดต่อสื่อสาร ระหว่าง ไมโครคอนโทรลเลอร์ (MCU) กับอุปกรณ์ภายนอก ซึ่งถูกพัฒนาขึ้นโดยบริษัท Philips Semiconductors โดยใช้สายสัญญาณเพียง 2 เส้นเท่านั้น คือ serial data (SDA) และสาย serial clock (SCL) ซึ่งสามารถ เชื่อมต่ออุปกรณ์ จำนวนหลายๆ ตัว เข้าด้วยกันได้ ทำให้ MCU ใช้พอร์ตเพียง 2 พอร์ตเท่านั้น
- SPI (Serial Peripheral Interface) เป็นการเชื่อมต่อกับอุปกรณ์เพื่อรับส่งข้อมูลแบบซิงโครนัส (Synchronize) มีสัญญาณนาฬิกาเข้ามาเกี่ยวข้องกับระหว่างไมโครคอนโทรลเลอร์ (Microcontroller) หรือจะเป็นอุปกรณ์ภายนอกที่มีการรับส่งข้อมูลแบบ SPI อุปกรณ์ที่ทำหน้าที่เป็นมาสเตอร์ (Master) โดยปกติแล้วจะเป็นไมโครคอนโทรลเลอร์ หรืออาจกล่าวได้ว่าอุปกรณ์ Master จะต้องควบคุมอุปกรณ์ Slave ได้ โดยปกติตัว Slave มักจะเป็นไอซี (IC) หน้าที่พิเศษต่างๆ เช่น ไอซีอุณหภูมิ, ไอซีฐานเวลานาฬิกาจริง (Real-Time Clock) หรืออาจเป็นไมโครคอนโทรลเลอร์ที่ทำหน้าที่ในโหมด Slave ก็ได้เช่นกัน
- PWM (Pulse Width Modulation) การสร้างสัญญาณพัลส์แบบสแควร์เวฟ ที่สามารถปรับเปลี่ยนความถี่และ Duty Cycle ได้เพื่อนำไปควบคุมอุปกรณ์ต่างๆเช่น มอเตอร์

- UART (Universal Asynchronous Receiver Transmitter) ทำหน้าที่รับส่งข้อมูลแบบอะซิงโครนัส สำหรับมาตรฐานการรับส่งข้อมูลแบบ RS-232

ไมโครคอนโทรลเลอร์ มีกี่ประเภทอะไรบ้าง?

ไมโครคอนโทรลเลอร์ มีด้วยกันหลายประเภทแบ่งตามสถาปัตยกรรม

(การผลิตและกระบวนการทำงานระบบการประมวลผล) ที่มีใช้ในปัจจุบันยกตัวอย่างดังนี้

1. ไมโคร คอนโทรลเลอร์ตระกูล PIC (บริษัทผู้ผลิต Microchip ไมโครชิป)
2. ไมโคร คอนโทรลเลอร์ตระกูล MCS51 (บริษัทผู้ผลิต Atmel, Phillips)
3. ไมโคร คอนโทรลเลอร์ตระกูล AVR (บริษัทผู้ผลิต Atmel)
4. ไมโคร คอนโทรลเลอร์ตระกูล ARM7, ARM9 (บริษัทผู้ผลิต Atmel, Phillips, Analog Device, Sumsung, STMicroelectronics)
5. ไมโคร คอนโทรลเลอร์ตระกูล Basic Stamp (บริษัทผู้ผลิต Parallax)
6. ไมโคร คอนโทรลเลอร์ตระกูล PSOC (บริษัทผู้ผลิต CYPRESS)
7. ไมโคร คอนโทรลเลอร์ตระกูล MSP (บริษัทผู้ผลิต Texas Instruments)
8. ไมโคร คอนโทรลเลอร์ตระกูล 68HC (บริษัทผู้ผลิต MOTOROLA)
9. ไมโคร คอนโทรลเลอร์ตระกูล H8 (บริษัทผู้ผลิต Renesas)
10. ไมโคร คอนโทรลเลอร์ตระกูล RABBIT (บริษัทผู้ผลิต RABBIT SEMICONDUCTOR)
11. ไมโคร คอนโทรลเลอร์ตระกูล Z80 (บริษัทผู้ผลิต Zilog)

ภาษาที่ใช้เขียน โปรแกรมควบคุมไมโครคอนโทรลเลอร์ มีอะไรบ้าง?

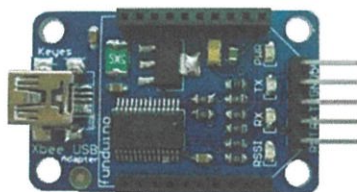
1. ภาษา Assembly
2. ภาษา Basic
3. ภาษา C
4. ภาษา Pascal
5. ภาษา Java

และอีกหลากหลายภาษาซึ่งได้รับการพัฒนามาอย่างต่อเนื่อง

ภาษา ดัง กล่าวที่กล่าวในเบื้องต้น ไมโครคอนโทรลเลอร์บางตระกูล จะใช้ได้ครบทุกภาษา แต่บางตระกูล จะใช้ได้บางภาษา ขึ้นอยู่กับบริษัทผู้ผลิต Software (โดยทั่วไปจะเรียกว่า Editor and Compiler) ที่ใช้ เขียนภาษาไมโครคอนโทรลเลอร์นั้นจะผลิตออกมาให้ Support หรือไม่

ผมขอสรุปในเบื้องต้นแบบง่ายๆ ไมโครคอนโทรลเลอร์เปรียบเทียบรถยนต์ทั่วไปรถยนต์ มีหลาย บริษัทผู้ผลิต, ในแต่ละบริษัทก็มีอยู่หลายรุ่นหลายแบบ รถยนต์มีระบบทุกอย่างพร้อมขึ้นอยู่กับเราจะขับ หรือควบคุมการใช้งานนั่นเองไมโคร คอนโทรลเลอร์ก็เช่นกัน มีหลายบริษัทผู้ผลิต,ในแต่ละบริษัทผู้ผลิต ก็ จะมีหลายเบอร์ให้เลือกใช้งาน,ไมโครคอนโทรลเลอร์ก็มีระบบต่างๆพร้อมสำหรับการใช้งานขึ้นอยู่กับ การเขียนโปรแกรมควบคุมให้มันทำงานตามที่ต้องการ

2.5 XBee USB Dongle



รูปที่ 2.12 XBee USB Dongle

XBee USB Dongle บอร์ดเชื่อมต่อ XBee กับคอมพิวเตอร์ผ่าน USB เพื่อแก้ไขค่าต่างๆ หรือ อัปเดต Firmware ด้วยโปรแกรม X-CTU เพียงแค่ติดตั้งโมดูล XBee เข้ากับบอร์ดก็สามารถเชื่อมต่อกับ คอมพิวเตอร์ด้วยสาย USB ได้เลย และสามารถจะเชื่อมต่อกับไมโครคอนโทรลเลอร์ผ่าน UART Serial Port ได้เช่นกัน

คุณสมบัติ

- ใช้ไฟ 5 V.
- มีพอร์ต Mini USB และ 5 pin สำหรับเชื่อมต่อแบบ UART
- ใช้ได้กับ XBee ทุกรุ่น (ที่ไม่ใช่แบบ Surface Mount)

บทที่ 3

ทฤษฎีและหลักการของการหาตำแหน่งภายในอาคาร

3.1 กล่าวนำ

การหาตำแหน่งภายในอาคารเป็นที่นิยมมากในปัจจุบันมีการประยุกต์ใช้งานการหาตำแหน่งภายในอาคารมากมาย อาทิ เช่น การหาตำแหน่งสินค้าในคลังสินค้า การหาตำแหน่งคนไข้ หรือ บุคลากรภายในโรงพยาบาล การหาตำแหน่งของพนักงานดับเพลิงในอาคารที่เกิดเพลิงไหม้ การหาตำแหน่งของสุนัขตำรวจที่ถูกฝึกให้ค้นหาวัตถุระเบิดภายในอาคาร การหาป้ายของเครื่องมือบำรุงรักษาหรืออุปกรณ์ที่กระจัดกระจายในโรงงาน ในบทนี้ได้อธิบายถึงทฤษฎีและหลักการในการหาตำแหน่งภายในอาคาร โดยในหัวข้อนี้จะอธิบายพร้อมทั้งจำแนกประเภทและเทคนิคในการหาตำแหน่งภายในอาคารซึ่งวิธีการคำนวณหาตำแหน่งโดยทั่วไปมีด้วยกัน 3 วิธี คือ การคำนวณหาตำแหน่งโดยวิธีเรขาคณิตรูปสามเหลี่ยม (Triangulation) การคำนวณหาตำแหน่งโดยการวัดค่าใกล้เคียงกับจุดอ้างอิง (Proximity) และการคำนวณหาตำแหน่งโดยการวิเคราะห์บริเวณเฉพาะ (Scene Analysis) ซึ่งบางครั้งเรียกว่า Fingerprint ทั้งนี้วิทยานิพนธ์ฉบับนี้จะมุ่งเน้นในการพัฒนาเทคนิคในการหาตำแหน่งภายในอาคารแบบ Fingerprint

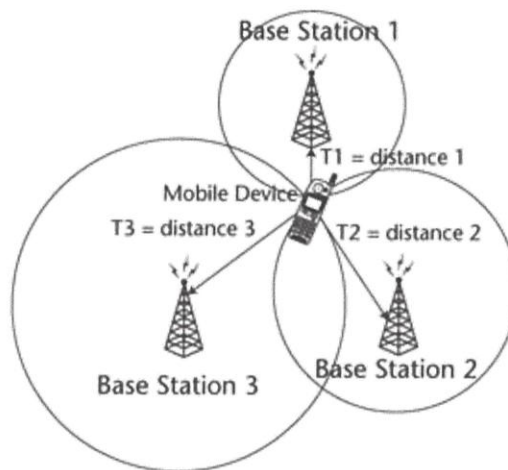
3.2 การหาตำแหน่งโดยวิธีการเรขาคณิตรูปสามเหลี่ยม (Triangulation)

การหาตำแหน่งโดยวิธีการเรขาคณิตรูปสามเหลี่ยม (Triangulation) โดยวิธีนี้สามารถจำแนกได้ ออกเป็นวิธีย่อย 2 วิธีด้วยกัน คือ วิธีการหาตำแหน่งด้วยระยะทาง (Lateration) และวิธีการหาตำแหน่งด้วยมุม (Angulation)

3.2.1 วิธีการหาตำแหน่งด้วยระยะทางแบบ Time-of-Arrival (TOA)

วิธีการคำนวณเวลาอาศัยหลักการบันทึกช่วงเวลาของสัญญาณของเครื่องส่งไปยัง เครื่องรับ แนวทางแรกของวิธีนี้คือ ให้เครื่องรับและเครื่องส่งมีการประสานเวลากัน (Synchronized) ดังนั้นสัญญาณที่มาถึงเครื่องรับ จึงสามารถเปรียบเทียบค่าความต่างเวลาได้ เช่น ในระบบ GPS ดาวเทียมจะส่งสัญญาณมายังเครื่องรับ โดยแต่ละเครื่องรับจะสร้างสัญญาณ เทียมขึ้นมาเอง (replica signal) เมื่อดาวเทียมส่งสัญญาณมาถึงเครื่องรับเครื่องรับจะทำการ เปรียบเทียบสัญญาณที่ได้กับสัญญาณที่สร้างขึ้นเองเพื่อคำนวณความต่างของเวลา Tof แต่ใน ทางปฏิบัติการประสานเวลาของสัญญาณนาฬิกาดาวเทียม

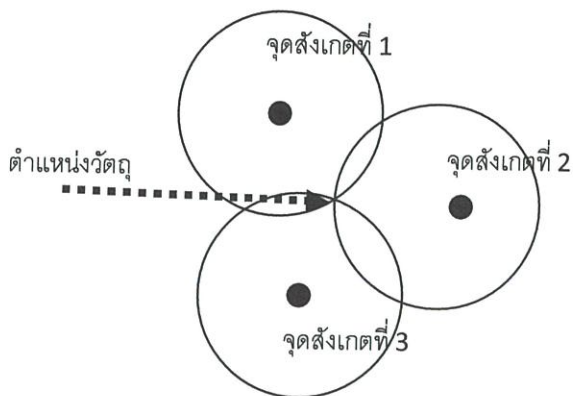
และเครื่องรับอาจไม่แม่นยำ 100% จึงต้องมีการอ่านสัญญาณดาวเทียมเพื่อคำนวณความคลาดเคลื่อนของเวลาด้วยแนวทางอื่นของวิธีนี้ คือการใช้ 2 สัญญาณที่มีการแพร่ของสัญญาณต่างกัน ค่าความต่างของสัญญาณที่ไปถึง ปลายทางของทั้งสองสัญญาณจะนำไปใช้ในการคำนวณระยะทาง



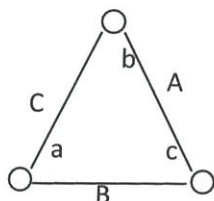
รูปที่ 3.1 แสดงการหาตำแหน่งด้วยการวัดแบบ TOA

3.2.2 วิธีการหาตำแหน่งด้วยมุมแบบ Angle of Arrival (AOA)

วิธีการหาทิศทางใช้การคำนวณมุมของสัญญาณที่มาถึง (Angle of arrival) หรือทิศทาง ของสัญญาณที่มาถึง (Direction of arrival) สำหรับการคำนวณหาตำแหน่ง ตำแหน่งของวัตถุ สามารถคำนวณได้ถ้าค่าที่ได้จากการวัดสามารถอ้างอิงไปยังจุดที่สามารถสังเกตได้ การวัดระยะทางต้องมีอย่างน้อย 3 สามารถระยะจากจุดสังเกตไปยังจุดที่ต้องการรู้ที่ตั้งสำหรับประมาณตำแหน่ง โหนดในระนาบ 2 มิติ (หากเป็นระบบ 3 มิติ ต้องการ 4 ระยะทางจาก 4 จุดสังเกต) ตำแหน่งของวัตถุจะอยู่ที่จุดตัด (Intersection) ของวงกลม 3 วงที่แต่ละวงมีรัศมีเท่ากับวัตถุไปยังจุดสังเกต พิจารณาตำแหน่งของเซนเซอร์โดยใช้การตัดกันของวงกลมซึ่งเป็น วงกลมที่มีรัศมีเท่ากับระยะทางที่เซนเซอร์ที่อยู่ห่างจากสถานีฐาน เราสามารถหาระยะทางโดยการคำนวณจากเวลาที่สัญญาณ จากเซนเซอร์นั้นเดินทางมาถึงสถานีฐาน ด้วยวิธีการนี้จะต้องใช้วงกลมอย่างน้อย 3 วงมาตัดกัน จึงจะได้ตำแหน่งวัตถุตั้งรูปที่ 1 หรือสามารถคำนวณได้โดยใช้ความสัมพันธ์ของสามเหลี่ยมดังรูปที่ 3.2



รูปที่ 3.2 แสดงวิธีการหาตำแหน่งของวัตถุ



Sines rule
$$\frac{A}{\sin a} = \frac{B}{\sin b} = \frac{C}{\sin c}$$

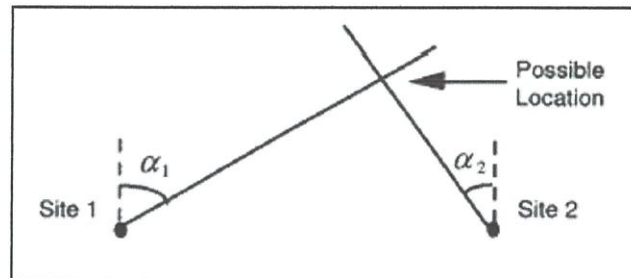
Cosine rule
$$\begin{aligned} C^2 &= A^2 + B^2 + 2AB \cos(c) \\ B^2 &= A^2 + C^2 - 2BC \cos(b) \\ A^2 &= B^2 + C^2 - 2BC \cos(a) \end{aligned}$$

รูปที่ 3.3 แสดงวิธีการหาตำแหน่งของวัตถุโดยใช้ความสัมพันธ์ของสามเหลี่ยม

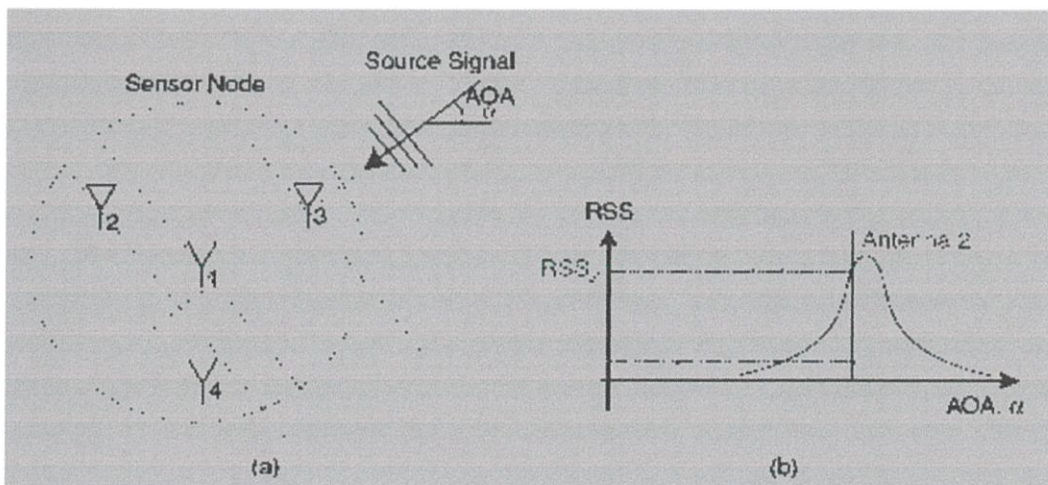
การวัดมุมตำแหน่งสามารถพิจารณาได้โดยใช้จุดอ้างอิงเพียง 2 จุด แต่ปัญหาสำหรับวิธีนี้คือ ใช้ได้กับกรณีไม่มีสัญญาณรบกวน (ideal noiseless) เท่านั้น ซึ่งในชีวิตจริงจะต้องมีสัญญาณรบกวนแน่นอน โดยอาจเกิดจากเครื่องมือวัดหรือปัจจัยภายนอกอื่นๆอีกมาก ดังนั้นค่าความคลาดเคลื่อน (error) จึงควรนำมาพิจารณาประกอบกับการคำนวณตำแหน่งวัตถุด้วย

AOA เป็นเทคนิคในการหาทิศทางของสัญญาณซึ่งใช้ Antenna array โดยการวัด Time Difference of Arrival (TDOA) ที่ Array หนึ่งๆ โดยปกติการวัด TDOA ทำได้โดยวัดความแตกต่าง ระหว่างเฟสของสัญญาณที่รับมา ณ element หนึ่งๆ ใน antenna array หรืออาจจะคิดว่าเป็นการ reverse การ Beamforming สัญญาณแต่ละ Element ถูกทำให้ Delay โดยน้ำหนักของการนำทางสู่ Antenna array ใน AOA Delay ของการมาถึงของแต่ละ Element จะถูกวัดและแปลงเป็นการวัดแบบ AOA

พิจารณาตำแหน่งของเซนเซอร์ โดยพิจารณาจากมุมที่เซนเซอร์ทำอยู่กับสถานีฐาน ด้วยวิธีการนี้จึงจำเป็นที่จะต้องใช้การพิจารณาจากสถานีฐานอย่างน้อย 2 แห่งด้วยกัน ตำแหน่งที่ได้นั้นเป็นจุดตัดของเส้นที่ลากทำมุมตามที่วัดได้ของสถานีฐานแห่งหนึ่งกับเส้นที่ลากทำมุมของอีก สถานีฐานหนึ่ง ด้วยวิธีการนี้ทำให้ต้องมีการติดตั้งอุปกรณ์เพิ่มขึ้นจากเดิมคือ Directional antenna เป็นดังรูปที่ 3 และ 4



รูปที่ 3.4 แสดงวิธีการหามุมของวัตถุ



รูปที่ 3.5 (a) แสดงวิธีการคำนวณหามุมโดยการใช้ความแตกต่างของ TOA ของเซนเซอร์แต่ละตัวในโนดหนึ่งๆ : เซนเซอร์รูปร่าง Y (b) AOA สามารถคำนวณได้จาก RSS ratio ระหว่าง Directional antennas

3.3 วิธีการหาตำแหน่งด้วยการวัดค่าความใกล้เคียงกับจุดอ้างอิง (Proximity)

วิธีการหาตำแหน่งด้วยการวัดค่าความใกล้เคียงกับจุดอ้างอิง (Proximity) จะบอกตำแหน่งที่สัมพันธ์กับสัญลักษณ์ (Symbolic Relative Location) โดยปกติจะอาศัยความหนาแน่นของกริดของสายอากาศ โดยเมื่อเป้าหมายถูกตรวจจับได้ด้วยสายอากาศเดี่ยว แต่ถ้ามีหลายๆ สายอากาศ ที่สามารถตรวจจับเป้าหมายได้ จะทำเลือกพิจารณาสัญญาณที่รับได้ที่มีความเข้มสัญญาณที่สุด ระบบที่ใช้เช่นระบบจะใช้คลื่นอินฟราเรด (IR) และคลื่นวิทยุ ตัวอย่างอื่นๆ เช่น วิธีพิสูจน์ว่าเป็นเซลล์เดียวกัน (Cell Identification (Cell-ID)) หรือ Cell of Origin (COO) วิธีนี้อาศัยเครือข่ายเซลลูลาร์ (Mobile Cellular Network) ในการระบุตำแหน่งอย่างคร่าวๆของอุปกรณ์เคลื่อนที่ ข้อดีของ Cell-ID คือถูกใช้งานในปัจจุบันและสามารถใช้ได้กับทุกอุปกรณ์เคลื่อนที่

3.4 วิธีการหาตำแหน่งโดยการวิเคราะห์บริเวณเฉพาะ (Scene Analysis/Fingerprint)

วิธีการหาตำแหน่งโดยการวิเคราะห์บริเวณเฉพาะ (Scene Analysis) ในบางครั้งเราจะเรียก Fingerprint วิธีนี้จะต้องมีการเก็บค่าของสัญญาณก่อนครั้งหนึ่งแล้วจึงค่อยมีการนำสัญญาณที่ได้มาทำการเปรียบเทียบกับสัญญาณที่ต้องการค้นหาตำแหน่งโดยใช้อัลกอริทึมในการเปรียบเทียบซึ่งวิธีนี้สามารถแบ่งการทำงานได้เป็น 2 ขั้นตอน

3.4.1 Offline phase หรือ Training Phase

ในการทำงานของช่วงนี้จะเป็นช่วงของการเก็บข้อมูล โดยจะทำการเก็บข้อมูล ที่ทำการตรวจเจอในแต่ละตำแหน่ง ซึ่งจะต้องทำการกำหนดตำแหน่ง (Grid Point) ที่ต้องการเก็บข้อมูลให้ครอบคลุมบริเวณที่หาตำแหน่ง โดยข้อมูลที่เก็บได้ในแต่ละตำแหน่งจะถูกเก็บไว้ในฐานข้อมูล เพื่อนำไปคำนวณค่าทางสถิติ ซึ่งเรียกว่าเป็นการทำ Radio map และข้อมูลที่เก็บได้ในแต่ละตำแหน่งที่กำหนดจะเรียกว่า Fingerprint

3.4.2 Online phase หรือ Localization Phase

ในการทำงานช่วงนี้จะเป็นช่วงของการหาตำแหน่ง โดยจะทำการหาตำแหน่งเครื่องอ่าน แล้วนำค่าที่ตรวจพบ ไปทำการเปรียบเทียบกับค่าที่ได้เก็บไว้ในฐานข้อมูลในช่วงแรก (Fingerprint) ด้วยอัลกอริทึมต่างๆ ต่อจากนั้นเมื่อเปรียบเทียบเสร็จแล้วก็จะแสดงตำแหน่งเครื่องอ่านออกมา โดยส่วนมากจะแสดงออกมาเป็นตัวเลขพิกัด ซึ่งประสิทธิภาพของการหาตำแหน่งขึ้นอยู่กับ Radio map ที่ทำไว้ในช่วงแรกและอัลกอริทึมที่นำมาใช้ในการเปรียบเทียบ

3.5 ความผิดพลาดที่อาจเกิดขึ้น

การประมาณค่าตำแหน่งจากการวัดที่มีสัญญาณรบกวนสามารถปรับปรุงให้ดีขึ้นได้หากเราเข้าใจถึงพฤติกรรมของ Error ที่เกิดขึ้น ชนิดของ Error ขึ้นกับชนิดของสัญญาณและสิ่งแวดล้อมของระบบ

1. Multipath Fading and shadowing

ในการวัดความแรงของสัญญาณคลื่นวิทยุปัญหา Multipath fading และ shadowing เป็นสาเหตุให้เกิดความแปรปรวนได้ถึง 30-40 dB ในช่วงครึ่งความยาวคลื่น การกระจัดกระจายใกล้เครื่องรับจะทำให้เกิดการแปรผันทางมุม (Angle of arrival) ทำให้ค่าการวัดคลาดเคลื่อนไป ถ้าใช้วิธี Time-of-Arrival เมื่อเกิด Multipath fading จะทำให้เกิดค่าหน่วงเวลา (Delay) การเปรียบเทียบค่าของเวลาคลาดเคลื่อนจากความจริง

2. Nonline-of-Sight (NLOS)

สำหรับวิธีวัดมุม (AoA) เมื่อใช้กับระยะทางไกลๆจะทำให้เกิดความคลาดเคลื่อนสูง สำหรับวิธีวัดความต่างของเวลา (Tof) ถ้าเส้นทางไปยังเครื่องรับที่ใกล้ที่สุดถูกกีดขวางจะทำให้ผลของการวัดระยะทางไกลกว่าที่เป็นจริง

3. Multiple-Access Interference

ปัญหานี้มักเกิดกับระบบ CDMA โดยเครื่องที่มีกำลังสูงไปรบกวนเครื่องที่มีกำลังต่ำกว่า สามารถเกิดกับระบบคลื่นเสียงและระบบคลื่นอัลตราโซนิก (Ultrasonic)

Transducer Calibration Issues ในระบบที่ใช้การวัดความแรงของสัญญาณที่ได้ (received signal strength indicator (RSSI)) จากคลื่นวิทยุราคาถูก เมื่อระบบไม่ใช่อุปกรณ์ที่มีความแม่นยำสูง ค่าความแปรปรวนจะเกิดจากความคลาดเคลื่อนเมื่อใช้เครื่องแปลงกำลัง

4. Fluctuations in Signal Propagation Speeds

เกิดกับคลื่นเสียงที่การแพร่ถูกรบกวนจากปัจจัยภายนอกเช่น แรงลมหรืออุณหภูมิ และค่าความชื้นในอากาศ ปัจจัยเหล่านี้จะทำให้ค่าความคลาดเคลื่อนสูงขึ้นเมื่อระยะทางมากขึ้น

3.6 การเขียนภาษา C# สำหรับ Microsoft Visual Studio 2010 เบื้องต้น

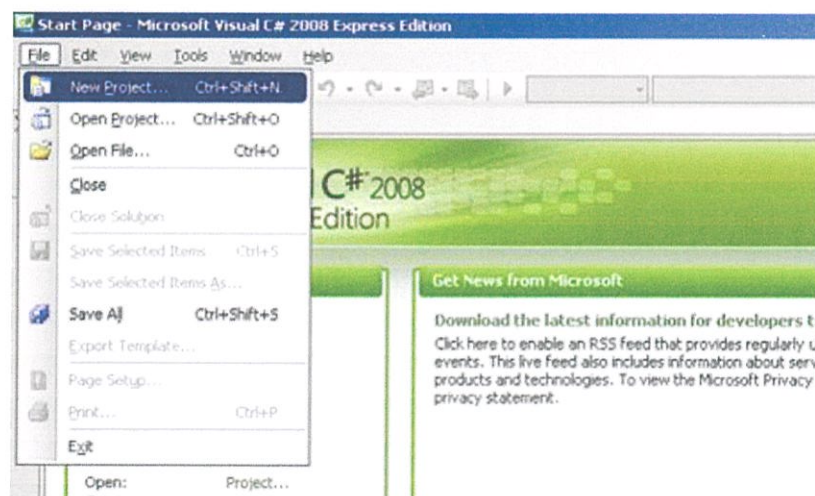
ภาษา C# (C# Programming Language) จะเขียนว่า C Sharp (ซี-ชาร์ป) ก็ได้ไม่ผิดบางที่เราอาจจะเคยได้ยินภาษาอื่นๆที่คล้าย C# เช่น ภาษา C, Java, C++ ซึ่งภาษาเหล่านี้เป็นที่มาของ C# นั่นเอง (ตัว C เป็นตัวบ่งบอกให้รู้ว่า C# มีต้นกำเนิดมาจาก C นั่นเอง) เครื่องหมาย # คือ เป็นสิ่งที่แสดงถึงความก้าวหน้ากว่า C++ ไปอีกระดับหนึ่ง (มอง # ให้เป็นเครื่องหมาย + ซ้อนกันสี่มุม อาจมองแบบ

C++++ ก็ได้ไม่ผิด) ถ้าจะให้พูดกันเข้าใจง่าย ๆ อีกก็คือ C# ได้รวบรวมข้อดีของภาษาต่างๆ เช่น Java, Delphi, C++ เข้าไว้ด้วยกัน อีกทั้งยังมีความเรียบง่ายกว่า ใครเคยใช้ Java จะรู้ว่ามีความคล้ายกับ C# มากที่สุด อีกทั้งยังมีเครื่องมือดีๆ อย่าง Visual C# 2012 ของทางไมโครซอฟท์ซึ่งได้พัฒนาเป็น Version ปัจจุบันในขณะนี้ จึงลดความยุ่งยากในการเขียนโปรแกรมได้มาก สามารถพัฒนาโปรแกรมระดับสูงได้ด้วย ภาษา C# เป็นเครื่องมือพัฒนาโปรแกรมที่อยู่ภายใต้โปรแกรม Microsoft Visual Studio ที่ทางบริษัท Microsoft ได้พัฒนาและออก Version ต่างๆ ดังนั้น ถ้าหากต้องการพัฒนาโปรแกรมด้วยภาษานี้ จึงต้องทำการดาวน์โหลดโปรแกรม Microsoft Visual Studio มาติดตั้งในเครื่อง ซึ่งโปรแกรม Microsoft Visual Studio นั้นก็มีอยู่หลาย Version ให้เลือกใช้ และ Version ล่าสุดในปัจจุบันนี้เป็น Microsoft Visual Studio 2012

3.6.1 การสร้างแอปพลิเคชันด้วย MS Visual C#

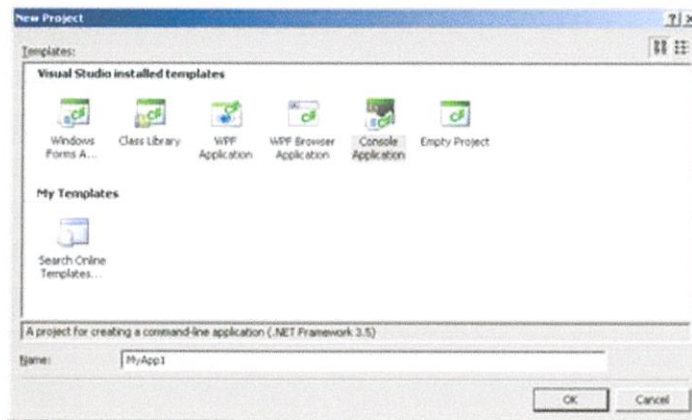
ในสภาพแวดล้อมของ MS Visual C# การพัฒนาแอปพลิเคชันขึ้นมาหนึ่งชิ้นเรียกว่าการสร้างโซลูชัน (solution) ซึ่งประกอบไปด้วยโปรเจกต์ (project) ตั้งแต่หนึ่งหรือมากกว่า โดยแต่ละโปรเจกต์เป็นส่วนหนึ่งของซอฟต์แวร์ที่อาจเป็นส่วนหนึ่งของโปรแกรมหลัก หรือส่วนไลบรารี (library) ที่ถูกเรียกใช้โดยโปรเจกต์อื่นๆ ดังนั้นแอปพลิเคชันแต่ละตัวจะประกอบไปด้วยโปรเจกต์อย่างน้อยหนึ่งโปรเจกต์เสมอ ในความรู้นี้จะอธิบายถึงกระบวนการสร้างโปรเจกต์ใน MS Visual C# สำหรับพัฒนาแอปพลิเคชันแบบคอนโซล โดยมีขั้นตอนต่อไปนี้

1. เลือกรายการเมนู New Project จากเมนู File (แสดงดังรูปที่ 3.6)



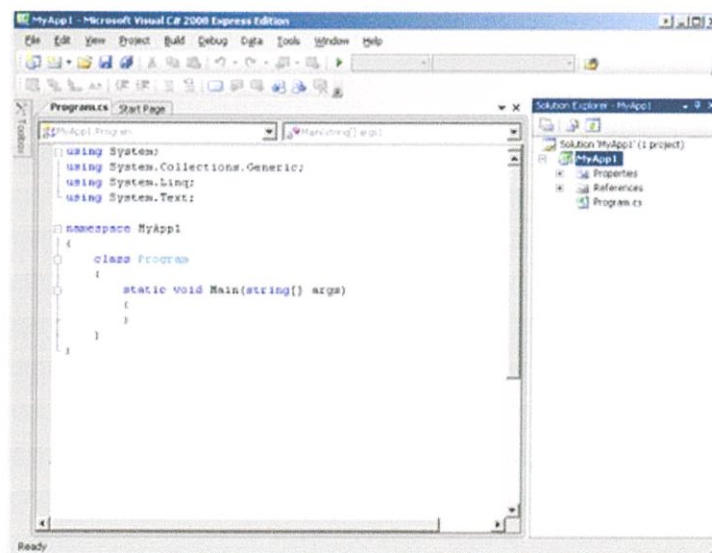
รูปที่ 3.6 รายการเมนูสำหรับสร้างโปรเจกต์ใหม่

2. ในไดอะล็อกซ์ New Project เลือกชนิดของโปรเจกต์เป็น Console Application และตั้งชื่อให้กับ โปรเจกต์ในกล่องข้อความด้านล่าง (แสดงดังรูปที่ 3.2) จากนั้นกดปุ่ม OK



รูปที่ 3.7 ไดอะล็อกซ์ในสร้างโปรเจกต์ใหม่และการตั้งค่าสำหรับคอนโซลแอปพลิเคชัน

3. โปรเจกต์ใหม่จะถูกสร้างขึ้นมาพร้อมทั้งไฟล์ชื่อ Program.cs ซึ่งมีโปรแกรมภาษา C# มาให้บางส่วนเพื่อให้ง่ายต่อการเริ่มต้น (แสดงดังรูปที่ 3.3) ในที่นี้เราอาจลบโปรแกรมที่ให้มาทั้งหมดทิ้งไป แล้วเริ่มเขียนโปรแกรมของเราเองก็ได้ เมื่อต้องการทดสอบการทำงานของโปรแกรมให้คลิก ที่ปุ่ม Run () เพื่อสั่งให้ MS Visual C# คอมไพล์โปรแกรมและให้โปรแกรมเริ่มทำงาน



รูปที่ 3.8 หน้าจอโดยรวมของ MS Visual C# หลังกระบวนการสร้างโปรเจกต์

3.6.2 ภาษา C# และขั้นตอนการพัฒนา

ภาษา C# เป็นภาษาโปรแกรมเชิงวัตถุ (object-oriented programming language) ที่ถูกพัฒนาขึ้นมาโดยบริษัทไมโครซอฟต์ การพัฒนาโปรแกรมคอมพิวเตอร์ด้วยภาษา C# นั้นจะประกอบด้วยขั้นตอนดังนี้

- วิเคราะห์ปัญหาและความต้องการในการพัฒนาโปรแกรม เช่น โปรแกรมจะติดต่อกับผู้ใช้อย่างไร ข้อมูลที่ผู้ใช้จะป้อนให้กับโปรแกรมเป็นอย่างไร และผลลัพธ์จะถูกแสดงผลอย่างไร
 - ออกแบบขั้นตอนวิธี โดยแสดงการทำงานของโปรแกรมในภาพรวมออกมาเป็นลำดับขั้นตอน แต่ละขั้นตอนมีความชัดเจนและสามารถเปลี่ยนให้อยู่ในรูปคำสั่งภาษา C# ได้โดยง่าย
 - นำขั้นตอนวิธีที่ออกแบบไว้มาสร้างเป็นไฟล์โปรแกรมรหัสต้นฉบับ (source code) ที่ถูกต้องตามโครงสร้างและไวยากรณ์ของตัวภาษา C# ทั้งนี้ไฟล์รหัสต้นฉบับต้องมีนามสกุล .cs เสมอ เช่น prog1.cs
 - แปลงรหัสต้นฉบับให้อยู่ในรูปรหัสภาษาเครื่องที่คอมพิวเตอร์เข้าใจและทำงานตามคำสั่งได้ ขั้นตอนนี้ต้องใช้โปรแกรมที่เรียกว่า คอมไพเลอร์ (compiler) ไฟล์รหัสภาษาเครื่องที่ถูกสร้างขึ้น จากคอมไพเลอร์จะมีนามสกุล .exe ซึ่งย่อมาจาก executable หมายถึงไฟล์ที่ถูกเรียกทำงานได้
 - ทดสอบการทำงานของโปรแกรม หากพบข้อผิดพลาดให้ตรวจสอบความถูกต้องในขั้นตอนที่ผ่านมา ซึ่งอาจหมายถึงการแก้ไขโปรแกรม ขั้นตอนวิธี หรือแม้กระทั่งวิเคราะห์ปัญหาและความต้องการใหม่
- ขั้นตอนการพัฒนาโปรแกรมหากดำเนินการดังกล่าวนี้นอกจากจะสามารถใช้กับภาษา C# แล้วยังสามารถนำไปประยุกต์ใช้กับภาษาอื่น ๆ ได้ด้วย

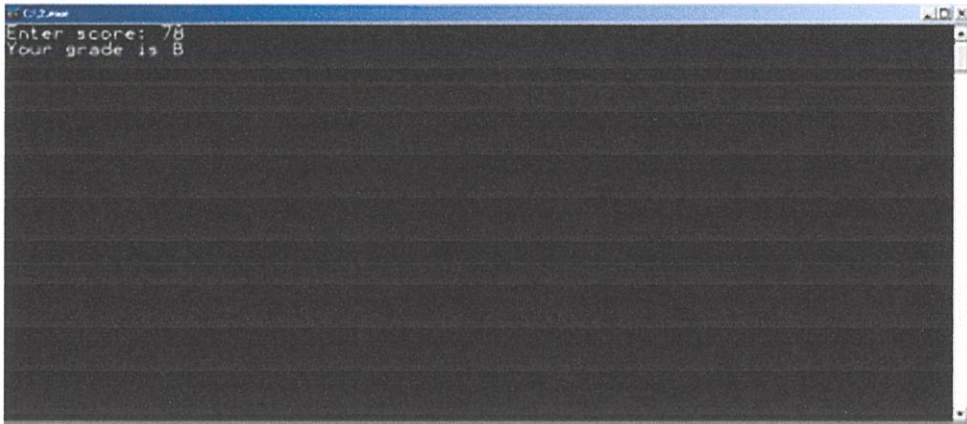
3.6.3 เครื่องมือสำหรับพัฒนาโปรแกรมด้วย C#

ขั้นตอนในการพัฒนาโปรแกรมที่กล่าวไปข้างต้นอาจดูซับซ้อนสำหรับผู้ที่ยังไม่มีประสบการณ์ อย่างไรก็ตาม ในปัจจุบันได้มีซอฟต์แวร์สำหรับช่วยพัฒนาโปรแกรมภาษา C# อยู่มากมาย ให้เลือกใช้ซึ่งเพิ่มความสะดวกและลดข้อผิดพลาดลงได้เป็นอย่างมาก ซอฟต์แวร์หลายตัวถูกแจกจ่ายให้นำไปใช้งานได้โดยไม่ต้องเสียค่าใช้จ่าย หนึ่งในซอฟต์แวร์เหล่านั้นคือ Microsoft Visual C# 2008 Express Edition (ในที่นี้ขอเรียกย่อ ๆ ว่า MS Visual C#)

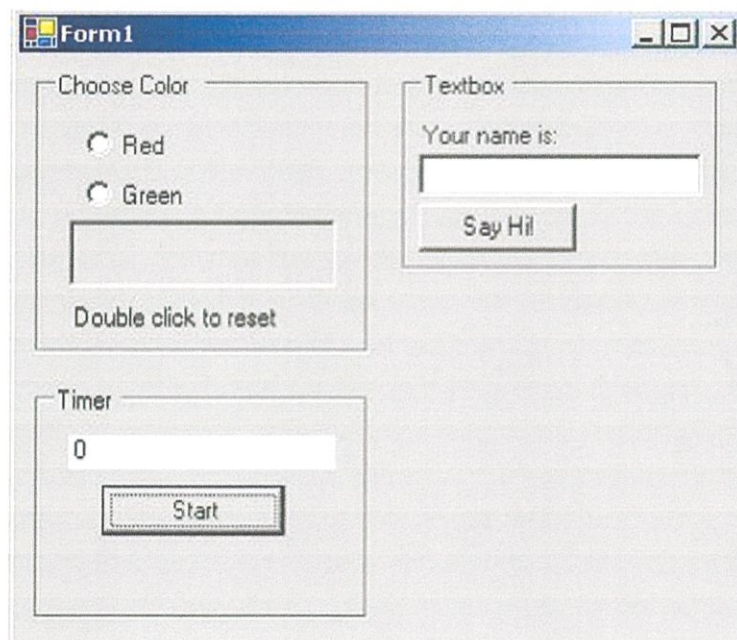
MS Visual C# มีคุณสมบัติคร่าว ๆ ดังต่อไปนี้

- สร้าง/แก้ไข/บันทึก โปรแกรมที่เขียนด้วยภาษา C# ได้
- คอมไพล์และทดสอบโปรแกรมที่เขียนขึ้นได้ทันที
- เพิ่มความง่ายในการเขียนโปรแกรมด้วยคุณสมบัติการเติมเต็มคำสั่ง (Code Completion)
- สนับสนุนการพัฒนาโปรแกรมทั้งประเภทที่ติดต่อกับผู้ใช้ผ่านคอนโซล (Console Application) ซึ่ง รับข้อมูลผ่านแป้นพิมพ์และแสดงผลข้อมูลในรูปตัวอักษรเพียงอย่างเดียว (แสดงดังรูปที่ 3.4) และ

ประเภทที่ใช้คุณสมบัติของวินโดวส์เต็มรูปแบบ (Windows Application) ซึ่งอนุญาตให้ผู้ใช้งาน ควบคุมโปรแกรมโดยใช้เมาส์และแสดงผลในแบบกราฟิกได้ (แสดงดังรูปที่ 3.5)



รูปที่ 3.9 ตัวอย่างคอนโซลแอปพลิเคชัน (Console Application)



รูปที่ 3.10 ตัวอย่างวินโดวส์แอปพลิเคชัน (Windows Application)

3.6.4 โครงสร้างโปรแกรมภาษา C#

การโปรแกรมภาษา C# ชั้นพื้นฐานที่มีเฉพาะส่วนของโปรแกรมหลักและไม่มีโปรแกรมน้อย (subroutine) จะมีส่วนประกอบดังนี้

```
namespace ___(A)___
{
  class ___(B)___
  {
    static void Main()
    {
      ___(C)___
    }
  }
}
```

รูปที่ 3.11 ส่วนของโปรแกรมหลัก

ตามโครงสร้างข้างต้น ณ ตำแหน่ง (A), (B), and (C) มีความหมายดังต่อไปนี้

- ตำแหน่ง (A) ระบุชื่อของเนมสเปซ (namespace) ซึ่งใช้ในการกำหนดขอบเขตให้กับคลาสต่างๆ รวมถึงใช้ในการจัดโครงสร้างของโปรแกรมขนาดใหญ่ให้เป็นสัดส่วนอีกด้วย โดยเฉพาะอย่างยิ่ง ในการพัฒนาแอปพลิเคชันที่ซับซ้อนโดยผู้พัฒนาหลายคน การกำหนดเนมสเปซของตนเองสามารถ ป้องกันปัญหาการตั้งชื่อคลาสหรือค่าคงที่อื่น ๆ ซ้ำกันได้ ส่วนไลบรารีของภาษา C# ที่มีให้เรา เรียกใช้งานได้ ก็ถูกเตรียมเอาไว้ในเนมสเปซชื่อต่าง ๆ เช่นเดียวกัน

- ตำแหน่ง (B) ระบุชื่อของคลาส (class)

- ตำแหน่ง (C) เป็นพื้นที่สำหรับคำสั่งต่าง ๆ ที่ผู้เขียนโปรแกรมต้องการให้คอมพิวเตอร์ปฏิบัติตาม นอกจากนี้โปรแกรมที่ไม่ซับซ้อนมากยังสามารถละส่วนที่ระบุเนมสเปซทิ้งไปได้ คลาสที่ถูกสร้าง ขึ้นมาโดยไม่อยู่ในขอบเขตของเนมสเปซใดจะถือว่าอยู่ในเนมสเปซกลาง (global namespace)

โปรแกรมภาษา C# ซึ่งแสดงข้อความ Hello World! ออกทางจอภาพ จากนั้นรอนจนกว่าผู้ใช้จะกด Enter และจบการทำงาน โปรแกรมนี้อยู่ในเนมสเปซชื่อ HelloApp และคลาสชื่อ HelloClass (แสดงดังรูป 4.7)

```
namespace HelloApp
{
class HelloClass
{
    static void Main()
    {
        System.Console.WriteLine("Hello World!");
        System.Console.ReadLine();
    }
}
}
```

รูปที่ 3.12 โปรแกรมแสดงข้อความ

```
class HelloClass
{
    static void Main()
    {
        System.Console.WriteLine("Hello World!");
        System.Console.ReadLine();
    }
}
```

รูปที่ 3.13 โปรแกรมแสดงข้อความแบบไม่ระบุเนมสเปซ

3.6.5 กฎการตั้งชื่อตัวระบุในภาษา C#

โปรแกรมภาษา C# ที่นำไปใช้งานจริงมักจะมีการใช้งานตัวระบุ (identifier) อยู่ทั่วไปภายในโปรแกรม เช่นชื่อของเนมสเปซและคลาสที่ได้กล่าวไปข้างต้น ภาษา C# ได้จำกัดกฎเกณฑ์การตั้งชื่อให้ตัวระบุเอาไว้ดังต่อไปนี้

- ชื่อตัวระบุต้องประกอบด้วยตัวอักษรภาษาอังกฤษ (A-Z,a-z) ตัวเลข (0-9) หรือเครื่องหมายขีดเส้นใต้ (_) เท่านั้น

- ตัวอักษรตัวแรกของชื่อต้องเป็นตัวอักษรภาษาอังกฤษ หรือตัวขีดเส้นใต้

- ชื่อตัวระบุจะมีความยาวได้ไม่เกิน 63 ตัวอักษร

- ชื่อตัวระบุต้องไม่ซ้ำกับคำสงวน (reserved word) เช่น class, namespace, int, void, static

3.6.6 ข้อมูลและตัวดำเนินการ

3.6.6.1 ชนิดข้อมูล (Data Types) ใน C#

C# กำหนดชนิดของข้อมูลไว้หลากหลายชนิดเพื่อรองรับการจัดเก็บข้อมูลหลาย ๆ ประเภทตารางที่ 1 แสดงชนิดข้อมูลหลัก ๆ ที่พบบ่อยในโปรแกรมทั่วไป

char	อักขระเดี่ยว เช่น a
bool	ค่าความจริง เป็นไปได้สองค่าคือ true หรือ false
byte	จำนวนเต็มไม่มีเครื่องหมาย ตั้งแต่ 0 ถึง 255
int	จำนวนเต็มมีเครื่องหมาย ตั้งแต่ - 2,147,483,648 ถึง 2,147,483,647
uint	จำนวนเต็มไม่มีเครื่องหมาย ตั้งแต่ 0 ถึง 4,294,967,295
long	จำนวนเต็มมีเครื่องหมาย ตั้งแต่ - 9,223,372,036,854,775,808 ถึง 9,223,372,036,854,775,807
ulong	จำนวนเต็มไม่มีเครื่องหมาย ตั้งแต่ 0 ถึง 18,446,744,073,709,551,615
float	จำนวนจริง (มีทศนิยมได้) เช่น 3.14159
double	จำนวนจริงที่เก็บความละเอียดมากเป็นสองเท่า
string	ข้อความ (สายอักขระ) เช่น Hello

3.6.6.2 ตัวแปร (variable)

เป็นตัวระบุประเภทหนึ่งที่นำมาใช้ในการอ้างถึงข้อมูล โดยค่าของมันสามารถ ถูกเปลี่ยนแปลงได้ตลอดเวลาที่โปรแกรมกำลังทำงานอยู่ ในภาษา C# ตัวแปรทุกตัวต้องถูกประกาศ ก่อนที่จะถูกนำมาใช้งาน โดยมีการระบุชนิดข้อมูลที่จะใช้กับตัวแปรนั้น ๆ ไว้ด้วย (แสดงดังรูป 3.9)

```
DataType variableName;
```

รูปที่ 3.14 การระบุชนิดข้อมูลที่จะใช้กับตัวแปร

ในที่นี้ *variableName* คือชื่อของตัวแปร และ *DataType* คือชื่อของชนิดข้อมูลที่ตัวแปรนี้ เก็บค่าได้ (ตามตัวอย่างในตารางที่ 1) นอกจากการประกาศตัวแปรตามแบบข้างต้นแล้ว เรายังสามารถกำหนดค่าเริ่มต้นให้กับตัวแปรนั้น ๆ ได้อีกด้วย (แสดงดังรูป 3.10)

```
DataType variableName = initialValue;
```

รูปที่ 3.15 การกำหนดค่าเริ่มต้นให้กับตัวแปร

คำสั่งด้านล่างประกาศตัวแปรชื่อ *distance* ที่มีชนิดข้อมูลเป็น *uint* และไม่มีกำหนดค่าเริ่มต้น (แสดงดังรูปที่ 3.11)

```
uint distance;
```

รูปที่ 3.16 การกำหนดชนิดข้อมูลเป็น *uint*

คำสั่งด้านล่างประกาศตัวแปรชื่อ *salary* ที่มีชนิดข้อมูลเป็น *long* และค่าเริ่มต้นเป็น 30000

```
long salary = 30000;
```

รูปที่ 3.17 การประกาศตัวแปรชื่อ *salary* ที่มีชนิดข้อมูลเป็น *long* และค่าเริ่มต้นเป็น 30000

3.6.6.3 ค่าคงที่ (Constants)

ค่าคงที่เป็นตัวระบุอีกประเภทหนึ่งที่น่ามาใช้ในการอ้างถึงข้อมูลเช่นเดียวกับตัวแปร สิ่งที่แตกต่างกันจากตัวแปรก็คือค่าของมันไม่สามารถเปลี่ยนแปลงได้อีกหลังจากการประกาศ ในภาษา C# ค่าคงที่ต้องถูก

ประกาศโดยระบุชนิดข้อมูลและค่าตั้งต้นก่อนถูกนำมาใช้งานเสมอ การประกาศค่าคงที่จะคล้ายคลึงกับการประกาศตัวแปร แตกต่างกันตรงที่ต้องมีการระบุด้วยคีย์เวิร์ด `const` (แสดงดังรูปที่ 3.13)

```
const DataType constantName = value;
```

รูปที่ 3.18 การประกาศค่าคงที่

ในที่นี้ `constantName` คือชื่อของค่าคงที่ ส่วน `DataType` คือชื่อชนิดข้อมูลที่ค่าคงที่นี้ใช้ อ้างถึง และ `value` ระบุค่าที่ค่าคงที่นี้ถูกใช้เป็นตัวแทน คำสั่งด้านล่างประกาศค่าคงที่ชื่อ `myconst` โดยมีชนิดข้อมูลเป็นแบบ `double` และใช้แทนค่า `2.71828` (แสดงดังรูปที่ 3.14)

```
const double myconst = 2.71828;
```

รูปที่ 3.19 การประกาศข้อมูลเป็นแบบ `double` และใช้แทนค่า `2.71828`

การใช้งานค่าคงที่มีประโยชน์มากในกรณีที่เราต้องอ้างอิงถึงค่าใดๆ ซ้ำกันอยู่บ่อยครั้งภายใน โปรแกรม

3.6.6.4 นิพจน์ทางคณิตศาสตร์ (Arithmetic Expressions)

นิพจน์ (expression) ภายในโปรแกรมภาษา C# หมายถึงส่วนของโปรแกรมที่สามารถถูกตีความเป็นค่าต่าง ๆ ได้ โดยนิพจน์อาจประกอบด้วยเทอมเพียงเทอมเดียว หรือเกิดจากการผสมกันของนิพจน์อื่นที่เล็กกว่า ด้านล่างเป็นตัวอย่างของนิพจน์ที่ประกอบด้วยเทอมเพียงเทอมเดียว

- ตัวเลขโดด เช่น `3000`, `1.414`
- ข้อความ เช่น `"Hello, World"`
- ค่าความจริง ได้แก่ `true` และ `false`
- ตัวแปรหรือค่าคงที่เดี่ยว ๆ ที่ผ่านการกำหนดค่าให้แล้ว เช่น `myName`, `salary` ในที่นี้จะเน้น

การศึกษาเกี่ยวกับนิพจน์ที่ให้ค่าเป็นตัวเลข เรียกว่า นิพจน์ทางคณิตศาสตร์ (arithmetic expression) โดยเราสามารถผสมนิพจน์ต่าง ๆ ให้เป็นนิพจน์ที่ซับซ้อนขึ้นโดยอาศัยตัวดำเนินการทางคณิตศาสตร์ที่มีให้ในภาษา C# ดังตารางที่ 1

ตารางที่ 3.1 ตัวดำเนินการทางคณิตศาสตร์

ตัวดำเนินการ	สัญลักษณ์	จำนวนนิพจน์ ที่นำมาผสม	ตัวอย่าง	ค่าของนิพจน์เมื่อ $x = 20$
บวก (add)	+	2	$x+4$	24
ลบ (subtract)	-	2	$32-x$	12
คูณ (multiply)	*	2	$x*2$	40
หาร (divide)	/	2	$x/2$	10
หารเอาเศษ(modulo)	%	2	$x\%6$	2
กลับเครื่องหมาย (negate)	-	1	$-x$	-20
จัดกลุ่ม	()	N/A	$(x+2)*3$	66

หากนิพจน์ที่สร้างขึ้นใหม่ประกอบด้วยตัวดำเนินการมากกว่าหนึ่งตัว C# จะคำนวณค่าเรียงตามลำดับก่อนหลังดังนี้

- ()
- *, / และ %
- + และ -
- หากตัวดำเนินการมีลำดับเท่าเทียมกัน คำนวณจากซ้ายไปขวา

3.6.6.5 คำสั่งที่ใช้ในการแสดงผล

คำสั่งหลักที่ใช้ในการแสดงข้อความออกทางจอภาพได้แก่คำสั่ง Write และ WriteLine ซึ่งถูกนิยามไว้ในคลาสชื่อ Console และเนมสเปซชื่อ System สองคำสั่งนี้มีการใช้งานในลักษณะเดียวกัน แตกต่างกันที่คำสั่ง WriteLine จะพิมพ์ข้อความตามด้วยการขึ้นบรรทัดใหม่หลังจากพิมพ์เสร็จ ลองพิจารณาการใช้งานจากตัวอย่างต่อไปนี้

```

class Hello
{
    static void Main()
    {
        System.Console.Write("Hello, ");
        System.Console.WriteLine("everybody");
    }
}

```

รูปที่ 3.20 การเรียกใช้งานคำสั่ง Write และ WriteLine โดยระบุเนมสเปสเต็มรูปแบบ

สังเกตว่าการเรียกใช้คำสั่ง Write และ WriteLine นั้นต้องระบุถึงคลาส Console ซึ่งเป็น คลาสที่นิยาม คำสั่งนี้เอาไว้ และเนื่องจากคลาส Console เป็นคลาสที่อยู่ในเนมสเปส System จึงต้อง ระบุชื่อเนมสเปส ไว้ด้วยตามตัวอย่าง อย่างไรก็ตามเราสามารถทำโปรแกรมให้สั้นลงได้โดยใช้คำสั่ง using ซึ่งเป็นการสั่งให้ คอมไพเลอร์ค้นหาคลาสที่เรียกใช้งานในเนมสเปสที่ระบุ ดังตัวอย่างต่อไปนี้

```

using System;

class Hello
{
    static void Main()
    {
        Console.Write("Hello, ");
        Console.WriteLine("everybody");
    }
}

```

รูปที่ 3.21 การเรียกใช้งานคำสั่ง Write และ WriteLine โดยใช้คำสั่ง using คำสั่ง Write และ WriteLine นั้นยังสามารถเรียกใช้งานในรูปแบบที่ซับซ้อนกว่านี้อีกมาก ดังแสดง ใน ตัวอย่างต่อไปนี้

```

using System;
class Hello
{
    static void Main()
    {
        int width = 80, height = 30;
        Console.WriteLine("Area of {0}x{1} rectangle = {2}",
            width, height, width*height);
    }
}

```

รูปที่ 3.22 การเรียกใช้งานคำสั่ง Write และ WriteLine โดยใช้สตริงกำหนดรูปแบบ (formatting string)

จะเห็นว่าสตริงที่เป็นพารามิเตอร์ตัวแรกของคำสั่ง WriteLine นั้นเป็นสตริงที่ใช้สำหรับ กำหนดรูปแบบในการแสดงผลของนิพจน์ที่ตามมาอีกสามตัว สัญลักษณ์ {0} {1} และ {2} คือตำแหน่งที่ C# จะนำค่าของนิพจน์ width, height และ width*height มาแทนที่

ตารางที่ 3.2 ตัวดำเนินการทางตรรกศาสตร์

ตัวดำเนินการ	คำอธิบาย	ตัวอย่าง
!	นิเสธ (Negation)	!a ผลลัพธ์คือนิเสธของ a (มีค่าความจริงตรงข้ามกับ a)
&&	และ (And)	a && b ผลลัพธ์จะเป็นจริงเมื่อ a และ b ต่างเป็นจริง
	หรือ (Or)	a b ผลลัพธ์จะเป็นจริงเมื่อ a หรือ b เป็นจริง

ตารางที่ 3.3 ตัวดำเนินการทางคณิตศาสตร์

ตัวดำเนินการ	คำอธิบาย	ตัวอย่าง
+	บวก	$a+b$ ผลลัพธ์คือ a บวก b
-	ลบ	$a-b$ ผลลัพธ์คือ a ลบ b
*	คูณ	$a*b$ ผลลัพธ์คือ a คูณ b
/	หาร	a/b ผลลัพธ์คือ a หาร b
%	หาเศษจากการหาร	$a\%b$ ผลลัพธ์คือเศษจาก a หาร b เช่น $(5\%3$ เท่ากับ 2)
++	การเพิ่มค่า	$a++$ ผลลัพธ์คือ $a=a+1$
--	การลดค่า	$a--$ ผลลัพธ์คือ $a=a-1$

ตารางที่ 3.4 ตัวดำเนินการเปรียบเทียบ

ตัวดำเนินการ	คำอธิบาย	ตัวอย่าง
==	เท่ากับ	$a==b$ จะเป็นจริงเมื่อ a เท่ากับ b
!=	ไม่เท่ากับ	$a!=b$ จะเป็นจริงเมื่อ a ไม่เท่ากับ b
<	น้อยกว่า	$a < b$ จะเป็นจริงเมื่อ a น้อยกว่า b
<=	น้อยกว่าหรือเท่ากับ	$a <= b$ จะเป็นจริงเมื่อ a น้อยกว่าหรือเท่ากับ b
>	มากกว่า	$a > b$ จะเป็นจริงเมื่อ a มากกว่า b
>=	มากกว่าหรือเท่ากับ	$a >= b$ จะเป็นจริงเมื่อ a มากกว่าหรือเท่ากับ b

3.6.6 คำสั่งเงื่อนไขและการทำซ้ำ

การเขียนโปรแกรมภาษา c# ในบางกรณีจำเป็นต้องให้ระบบตัดสินใจดำเนินการอย่างใดอย่างหนึ่ง โดยการกำหนดเป็นเงื่อนไข เช่น ถ้าเงื่อนไขเป็น " จริง " ให้ทำการดำเนินการตามชุดคำสั่ง A แต่ถ้าเงื่อนไขเป็น " เท็จ " ให้ดำเนินการตามชุดคำสั่ง B เป็นต้น คำสั่งกำหนดเงื่อนไขได้แก่ คำสั่ง if ใช้ควบคุมการทำงานของโปรแกรมตามเงื่อนไข

```
if( เงื่อนไข )
{
    คำสั่ง
}
```

คำสั่ง if-else ใช้ควบคุมการทำงานของโปรแกรมสองส่วน

```
if( เงื่อนไข )
{
    คำสั่ง
}
else
{
    คำสั่ง
}
```

คำสั่ง while เป็นการทำซ้ำโดยการตรวจสอบเงื่อนไขก่อน ถ้าเงื่อนไขเป็นจริงจะกระทำซ้ำคำสั่ง ที่อยู่ใน {...} และหากเงื่อนไขเป็นเท็จจะกระทำคำสั่งหลัง }.... (หลุดออกจากลูป)

```
Unsigned int i;
i = 0;
while( i < 100 )
{
    คำสั่ง
i++;
}
คำสั่ง
```

คำสั่ง do - while เป็นการทำซ้ำโดยจะกระทำซ้ำคำสั่งที่อยู่ใน {...} ก่อนแล้วตรวจสอบเงื่อนไขทีหลัง จนกระทั่ง เงื่อนไขเป็นเท็จจะกระทำคำสั่ง หลัง } (หลุดจากลูป)

```
Unsigned int i;
```

```
i = 0;
```

```
do
```

```
{
```

```
    คำสั่ง
```

```
    i++;
```

```
}
```

```
while ( i < 100 );
```

```
คำสั่ง
```

คำสั่ง Switch - Case

```
switch ( เงื่อนไข )
```

```
{
```

```
    case 1 :
```

```
        คำสั่ง;
```

```
        Break;
```

```
    case 2 :
```

```
        คำสั่ง;
```

```
        Break;
```

```
    case n :
```

```
        คำสั่ง;
```

```
        Break;
```

```
}
```

คำสั่ง for เป็นการซ้ำที่ผู้เขียนโปรแกรมสามารถกำหนดจำนวนครั้งให้กระทำคำสั่งได้ โดยการกำหนดเริ่มต้นให้แก่ตัวแปร Counter และจำนวนรอบที่ต้องการทำซ้ำ เมื่อกระทำซ้ำคำสั่งใน {...} จนครบตามจำนวนรอบก็จะไปทำคำสั่งหลัง } (ออกจากลูป)

```
Unsigned int i;
```

```
for ( i = 0; i < 100; i++)
```

```
{
```

```
    คำสั่ง
```

```
}
```

บทที่ 4

วิธีการและขั้นตอนการทดลองโครงการ

4.1 กล่าวนำ

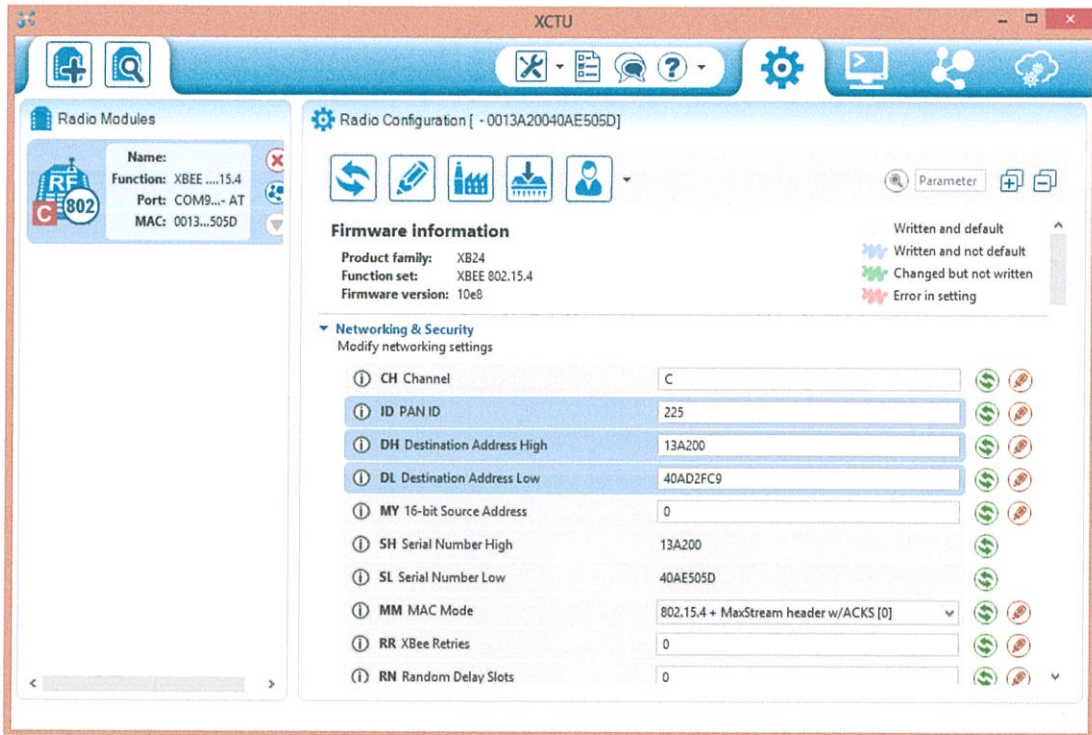
ในบทนี้จะกล่าวถึงรายละเอียดของวิธีการและขั้นตอนในการทดลองหาตำแหน่งซึ่งในการทดลองโครงการในปริภูมิกอณูนิพนธ์ฉบับนี้จะใช้เทคนิคการวิเคราะห์บริเวณเฉพาะ (Scene Analysis) บางครั้งเรียกว่า Fingerprint มาทำการหาตำแหน่งโดยการเปรียบเทียบรูปแบบของสัญญาณที่ได้รับมาจากสถานีฐานหรือเป้าหมายกับสัญญาณที่ได้ทำการสำรวจเก็บข้อมูลเอาไว้แล้วในฐานข้อมูล โดยใช้อัลกอริทึมในการเปรียบเทียบรูปแบบของสัญญาณ โดยในปริภูมิกอณูนิพนธ์ฉบับนี้ได้ใช้ Xbee ที่ทำงานในย่านความถี่ 2.4 กิกะเฮิรตซ์ ในการทดลองโครงการนี้จะนำ Referent Node ติดไว้บนเพดานซึ่งทราบพิกัดตำแหน่งที่ได้ติดตั้งไว้แล้วและนำ Object Node ไปติดไว้กับวัตถุที่ต้องการทราบตำแหน่ง ชั้นแรกจะทำการเก็บข้อมูลไว้ในฐานข้อมูลก่อน (ข้อมูลที่เก็บคือค่าความเข้มของสัญญาณของ Object Node ที่อยู่ ณ ตำแหน่งต่างๆ) ต่อมาเมื่อ Referent Node ทำการอ่านค่าความเข้มของสัญญาณ Object Node ก็ให้นำข้อมูลที่อ่านได้มาทำการเปรียบเทียบกับข้อมูลที่ทำการเก็บไว้ในฐานข้อมูลและทำการคำนวณหาพิกัดของตำแหน่งของวัตถุ รายละเอียดของวิธีการและขั้นตอนในการทดลองนั้นจะแสดงในหัวข้อต่อไป

4.2 อุปกรณ์ที่ใช้ในการทดลองโครงการ

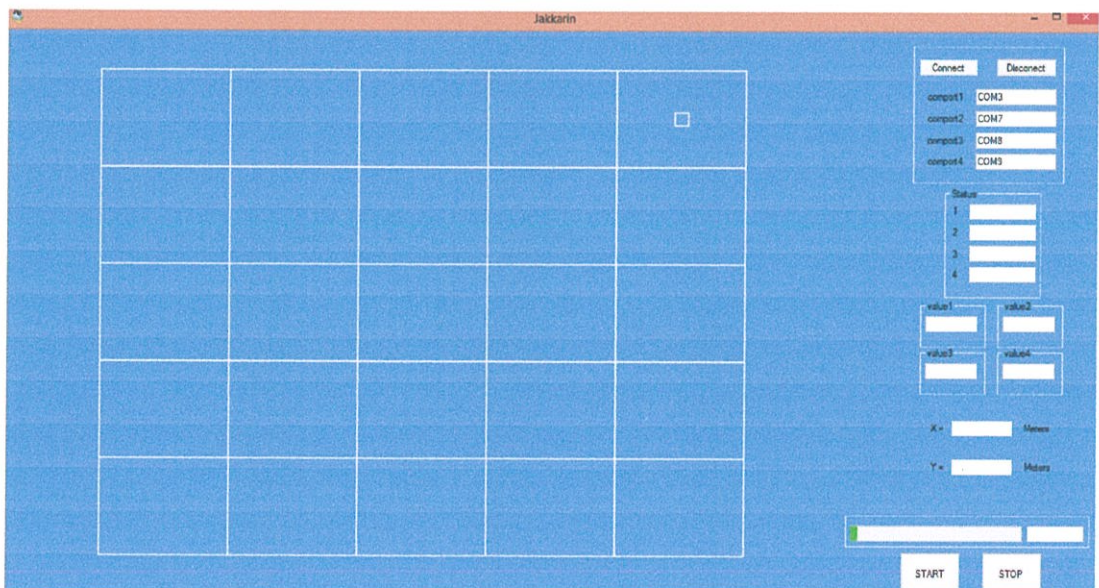
ในการทดลองหาตำแหน่งซึ่งในการทดลองโครงการในปริภูมิกอณูนิพนธ์ฉบับนี้ได้ทำการทดลองในบริเวณห้องน้ำตึก 12 ชั้น คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง เพื่อศึกษาในที่ที่ไม่มีสัญญาณไวไฟรบกวนและได้ทำการทดลองในบริเวณชั้น 4 ตึกภาคเครื่องกลเพื่อศึกษาในที่ที่มีสัญญาณไวไฟรบกวน โดยอุปกรณ์ที่นำมาใช้ในการทดลองมีดังต่อไปนี้

- เครื่องคอมพิวเตอร์พกพา (Notebook Computer)

เครื่องคอมพิวเตอร์พกพาพร้อมทำการติดตั้งลงโปรแกรมติดต่อ Xbee โดยใช้โปรแกรม X-CTU (แสดงดังรูปที่ 4.1) และโปรแกรม Microsoft Visual Studio 2010 ในการติดต่อสื่อสารกันระหว่างคอมพิวเตอร์พกพาและ Xbee (แสดงดังรูปที่ 4.2) จะใช้พอร์ตยูเอสบีมินิบี (USB Mini b) ในการติดต่อกับคอมพิวเตอร์พกพาแต่ว่าเครื่องคอมพิวเตอร์พกพาที่นำมาใช้ในการทดลองนั้นมีแต่พอร์ตยูเอสบี จึงต้องมีอุปกรณ์เพิ่มเติมก็คืออุปกรณ์แปลงพอร์ตยูเอสบีมินิบีเป็นยูเอสบีแบบเอ (แสดงดังรูปที่ 4.3)



รูปที่ 4.1 โปรแกรมที่ใช้ในการลงเฟิร์มแวร์กับ Xbee



รูปที่ 4.2 โปรแกรมที่ใช้ในการระบุตำแหน่งของ Xbee



รูปที่ 4.3 อุปกรณ์แปลงพอร์ตยูเอสบีมินิเป็นยูเอสบีแบบเอ

- ยูเอสบีฮับ 4 พอร์ต OKER H-409

ยูเอสบีฮับ 4 พอร์ต คุณสมบัติเพิ่ม 4 พอร์ทการใช้งานยูเอสบีความเร็วสูง 2.0 สนับสนุนการเชื่อมต่อแบบยูเอสบี 2.0 สนับสนุนการใช้งานอุปกรณ์ที่มีความเร็วสูงได้ถึง 127 อุปกรณ์ สามารถเชื่อมต่อกับยูเอสบี 1.1 ได้สนับสนุนอัตราการถ่ายโอนข้อมูลที่ 1.5/12/480 เมกะบิตต่อวินาที จ่ายไฟได้ที่ 500 มิลลิแอมป์ (แสดงดังรูปที่ 4.4)



รูปที่ 4.4 อุปกรณ์ยูเอสบีฮับ 4 พอร์ต OKER H-409

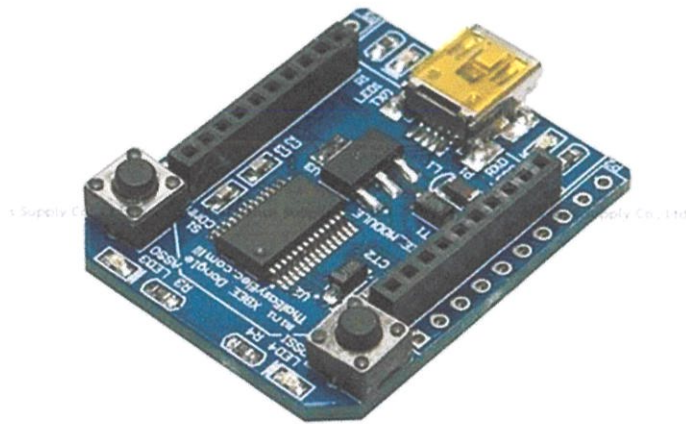
- ยูเอสบีแบบแจ็คเอไปเป็นปลั๊กเอ
- ยูเอสบีแบบแจ็คเอไปเป็นปลั๊กเอความยาว 10 เมตร จำนวน 4 เส้น (แสดงดังรูปที่ 4.5)



รูปที่ 4.5 ยูเอสบีแบบแจ็คเอไปเป็นปลั๊กเอ

- มินิ Xbee ยูเอสบีตองเกิ้ลวี 2

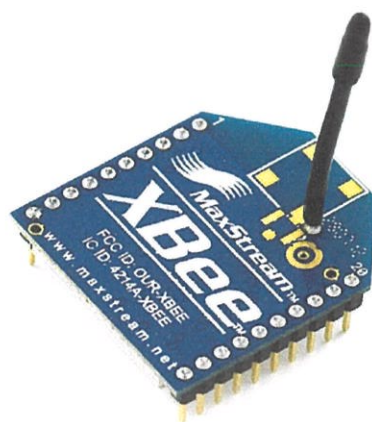
มินิ Xbee ยูเอสบีตองเกิ้ลวี 2 เป็นอุปกรณ์ที่ใช้สำหรับทำงานร่วมกับ สำหรับ อัปเดตเฟิร์มแวร์ หรือ การกำหนดค่าพารามิเตอร์ หรือ อัปเดตเฟิร์มแวร์ผ่านทางคอมพิวเตอร์ ร่วมกับโปรแกรม X-CTU สามารถต่อ Xbee ร่วมกับ มินิ Xbee ยูเอสบีตองเกิ้ลวี 2 ผ่านยูเอสบีพอร์ตได้โดยตรง ซึ่งคอมพิวเตอร์จะมองเป็นคอมพอร์ท (Serial UART) เนื่องจากใช้ IC FT232RL ซึ่งเป็นยูเอสบีถึงอนุกรมในบอร์ดเดียว ไม่ต้องต่ออุปกรณ์ใด ๆ เพิ่มเติม สามารถใช้ร่วมกับ Xbee ได้ทุกรุ่น อัปเดตเป็น V2 เพื่อให้สามารถใช้ได้กับ Xbee และ Xbee Pro ได้ทุกรุ่น ใช้จำนวน 4 ตัว (แสดงดังรูปที่ 4.6)



รูปที่ 4.6 มินิ Xbee ยูเอสบีตองเกิ้ลวี 2

- Xbee

ในการทดลองนี้ได้ใช้ Xbee Series 1 ในการทดลองเพราะว่าราคาค่อนข้างถูกกว่าราคาของ Xbee Series 2 และ Xbee Pro อีกทั้งระยะในการถ่ายโอนข้อมูลอยู่ในระยะที่เหมาะสมกับการทดลองโครงการนี้ โดย Xbee ที่ใช้เป็นยี่ห้อ Digi เป็นโมดูล รับส่งสัญญาณไร้สาย ย่านความถี่ 2.4 GHz ซึ่งเป็นย่านความถี่วิทยุ สมัครงเล่น ตามมาตรฐานโปรโตคอล 802.15.4 ใช้พลังงานต่ำ สายอากาศแบบไวร์ ซึ่งในการทดลองนี้ได้ใช้ Xbee ทั้งหมดจำนวน 5 ตัว ในการทดลอง (แสดงดังรูปที่ 4.7)

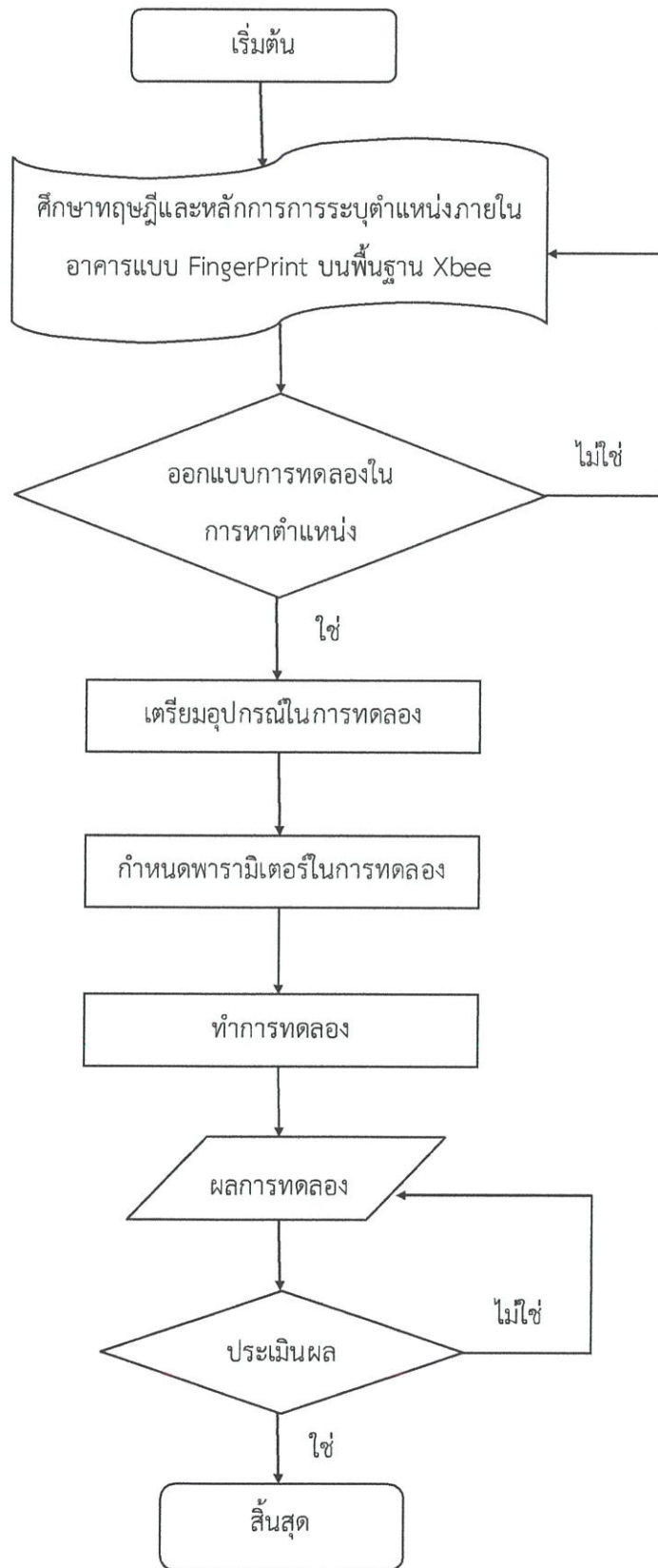


รูปที่ 4.7 Xbee Series 1

เมื่อได้ดูตารางเปรียบเทียบระหว่าง Xbee Series 1, Xbee Series 2, Xbee Series 1 Pro และ Xbee Series 2 Pro แล้วจะพบว่า Xbee Series 1 ค่อนข้างเหมาะสมกับการทดลองโครงการนี้เพราะว่า Xbee Series 1 มีราคาถูกกว่าและมีระยะเวลาการถ่ายโอนข้อมูลก็เหมาะสมกับการทดลองโครงการนี้ ในการทดลองของเราจึงทำการเลือก Xbee Series 1 มาทำการทดลอง

- เชือกฟาง เทปกาวและกรรไกร

ในการทดลองนี้ได้นำเชือกฟาง เทปกาวและกรรไกรมาเพื่อใช้อำนวยความสะดวกในการทดลอง ซึ่งจะกล่าวในหัวข้อต่อไป



รูปที่ 4.8 โฟลว์ชาร์ต

4.3 ขั้นตอนและรูปแบบวิธีการทดลองโครงการในการหาตำแหน่ง

ในหัวข้อนี้จะเสนอขั้นตอนและรูปแบบวิธีการทดลองโครงการในการหาตำแหน่ง โดยโฟลว์ชาร์ตของขั้นตอนการทำงานทดลองวิจัย (แสดงดังรูปที่ 4.8) และมีวิธีการดังต่อไปนี้

4.3.1 ทำการกำหนดค่าพารามิเตอร์ของ Xbee

ในการกำหนดค่าพารามิเตอร์ของ Xbee จะต้องใช้โปรแกรม X-CTU ในการกำหนดโดยกำหนดค่าพารามิเตอร์ส่วนที่ 1 (แสดงดังรูปที่ 4.9) แต่ก่อนที่จะทำการกำหนดค่าพารามิเตอร์ต้องทำการเลือก Xbee จำนวน 1 ตัว จาก 5 ตัวมาทำหน้าที่เป็น End Device Xbee อีก 4 ตัวที่เหลือจะทำหน้าที่เป็น Coordinator Xbee

▼ Networking & Security
Modify networking settings

① CH Channel	C	1
① ID PAN ID	225	
① DH Destination Address High	13A200	2
① DL Destination Address Low	40AD2FC9	
① MY 16-bit Source Address	0	
① SH Serial Number High	13A200	3
① SL Serial Number Low	40AE505D	
① CE Coordinator Enable	Coordinator [1]	4

รูปที่ 4.9 การกำหนดค่าพารามิเตอร์ส่วนที่ 1

1. กำหนดค่า Channel เป็นตัวอักษรใดก็ได้ตามที่ต้องการและกำหนดค่า PAN ID เป็นตัวเลขใดก็ได้ตามที่ต้องการ (Xbee ทุกตัวต้องมีค่าเดียวกัน)

2. กำหนดค่า Destination Address High ของ Coordinator Xbee ทั้ง 4 ตัวไปยังค่า Serial Number High และค่า Destination Address Low ไปยังค่า Serial Number Low ของ End Device Xbee

3. แสดงค่า Serial Number High และค่า Serial Number Low ของ Xbee แต่ละตัวหรือสามารถดูได้จากใต้ห้องของ Xbee

4. กำหนดค่า Coordinate Enable ของ Coordinator Xbee เป็น Coordinator ทั้ง 4 ตัว และกำหนดให้เป็น End Device จำนวน 1 ตัว (ตัวที่ได้ทำการเลือกไว้แล้วตอนแรก)

Serial Interfacing

Modify modem interfacing options

BD Interface Data Rate	9600 [3]	5
NB Parity	No Parity [0]	
RO Packetization Timeout	3 x character times	
AP API Enable	API disabled [0]	6
PR Pull-up Resistor Enable	FF	

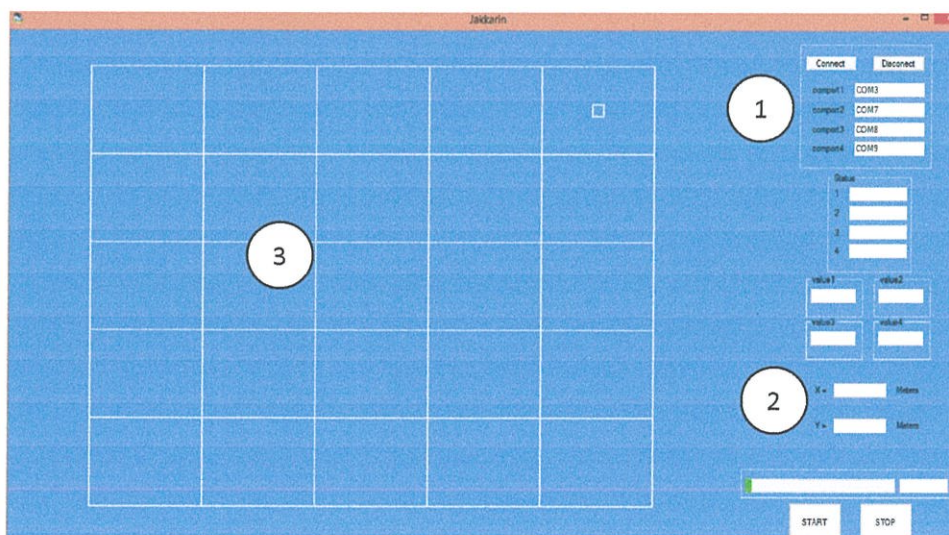
รูปที่ 4.10 การกำหนดค่าพารามิเตอร์ส่วนที่ 2

5. กำหนดค่า Interface Data Rate เป็น 9600 ทุกตัว

6. กำหนดค่า API Enable ให้เป็น API disabled ทุกตัว

4.3.2 ทำการศึกษาและเขียนโปรแกรมอินเตอร์เฟซ

ในส่วนนี้เป็นส่วนที่สร้างขึ้นจากโปรแกรม Visual Studio 2010 เพื่อใช้ในการรับและแสดงผลที่ได้จากกระบวนการ โดยจะสื่อสารกับ Xbee โดยผ่านทาง Serial Port โดยในส่วนแสดงผลนี้จะปรากฏหน้าต่างต่าง (แสดงดังรูป 4.11)



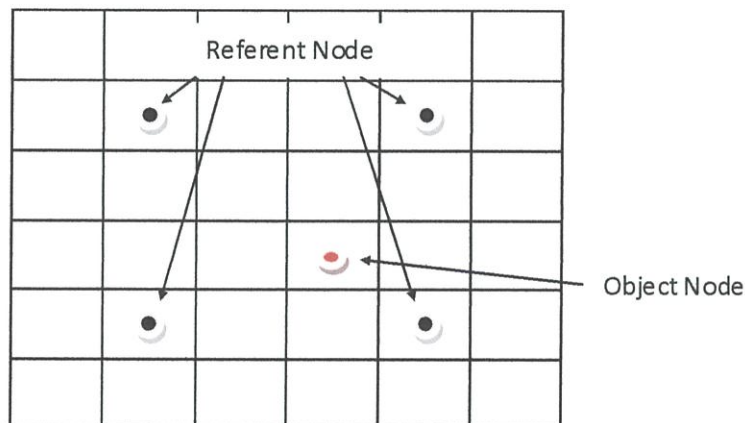
รูปที่ 4.11 หน้าต่างแสดงผลที่สร้างจาก Visual Studio 2010

จากรูปที่ 4.11 เป็นหน้าต่างแสดงผลที่สร้างขึ้นจากโปรแกรม Visual Studio 2010 ซึ่งสามารถอธิบายเป็นส่วนๆ ตามภาพได้ดังนี้

1. ส่วนการตั้งค่า จะใช้ในการตั้งค่าการใช้งานในการติดต่อกับส่วนกระบวนการผ่าน Serial Port
2. ส่วนแสดงผลแบบตัวอักษร จะแสดงผลในรูปแบบของตัวเลขที่ได้จากกระบวนการ โดยส่วนนี้จะแสดงค่าของ ระยะในแกน X และแกน Y
3. ส่วนที่แสดงผลด้วยพิกัดแบบจุดและจะทำการแสดงผลที่ได้รับมาจากกระบวนการในรูปของจุด เพื่อให้ทราบถึงตำแหน่งที่ตั้งของ Object Node และเพื่อทำให้มองเห็นภาพชัดเจนยิ่งขึ้น

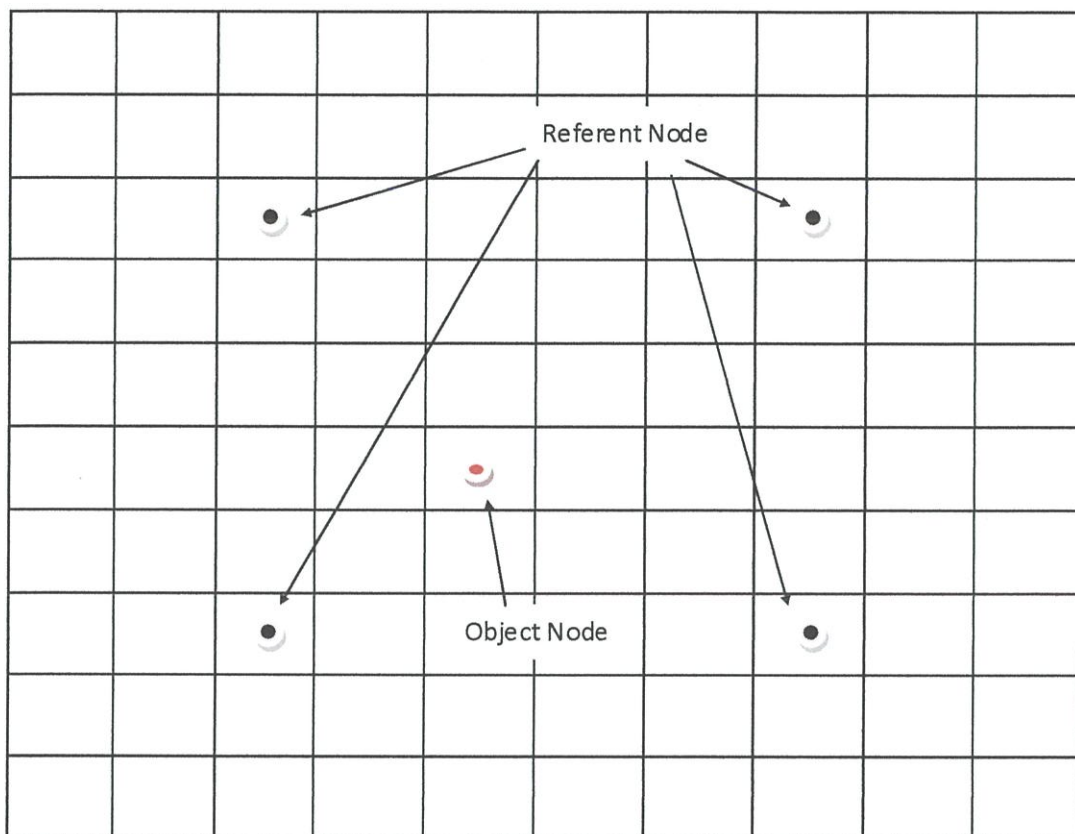
4.3.3 จัดเตรียมการทดลองและสภาพแวดล้อมในการทดลอง

ในการทดลองนี้ได้มีการจัดเตรียมการทดลองและสภาพแวดล้อมในการทดลอง โดยขั้นแรกต้องทำการแบ่งพื้นที่เป็นตารางช่องละ 0.5×0.5 ตารางเมตร จำนวน 36 ช่องโดยทำในบริเวณที่ไม่มีสัญญาณไวไฟรบกวน (แสดงดังรูปที่ 4.12)



รูปที่ 4.12 การแบ่งพื้นที่เพื่อศึกษาบริเวณที่ไม่มีสัญญาณรบกวน

เมื่อทำการติดตั้ง Coordinator Xbee ไว้ทั้ง 4 มุมเสร็จเรียบร้อยแล้วก็นำยูเอสบีซีที่ออกมาจากดองเกล์ต่อเข้ากับยูเอสบีซี แล้วนำฮับต่อเข้ากับคอมพิวเตอร์พกพา จากนั้นทำการกดปุ่มสตาร์ทโปรแกรมและทำการเก็บค่าความเข้มของสัญญาณในแต่ละพื้นที่แล้วทำการบันทึกผลว่าพื้นที่ที่ไม่มีสัญญาณไวไฟรบกวนค่าความเข้มของสัญญาณมีการเปลี่ยนแปลงมากน้อยเพียงใด ต่อไปเราก็จะไปทำการศึกษาในที่ที่มีสัญญาณไวไฟรบกวนเป็นอย่างมาก อีกทั้งจะทำการเปรียบเทียบว่าถ้ามีการใช้ Coordinator Xbee จำนวน 3 ตัว และ 4 ตัว มีความแม่นยำแตกต่างกันมากน้อยเพียงใด โดยจะทำการจัดเตรียมการทดลองและสภาพแวดล้อมในการทดลอง (แสดงดังรูปที่ 4.13)



รูปที่ 4.13 การแบ่งพื้นที่เพื่อศึกษาบริเวณที่มีสัญญาณรบกวน

ในกรณีที่จะทำการศึกษาบริเวณที่มีสัญญาณไวไฟรบกวนผู้ทำโครงการจะทำการแบ่งพื้นที่เป็นช่องละ 0.5x0.5 ตารางเมตร จำนวน 100 ช่องเมื่อเราทำการแบ่งพื้นที่เสร็จเรียบร้อยแล้ว (แสดงดังรูปที่ 4.14)

ตารางที่ 4.1 รายละเอียดค่าต่างๆที่ใช้ในการทดลองหาตำแหน่ง

รายละเอียด	ค่า
จำนวนช่องที่ทำการแบ่ง	100 ช่อง
ความสูง Referent Node จากพื้น	2.5 เมตร
ความสูง Object Node จากพื้น	0.6 เมตร
พื้นที่ในการหาตำแหน่ง	5x5 เมตร
ย่านความถี่ของ Xbee	2.4 กิกะเฮิรตซ์



รูปที่ 4.14 การแบ่งพื้นที่เพื่อศึกษาบริเวณที่มีสัญญาณรบกวน



รูปที่ 4.15 แสดงการติดตั้ง Coordinator Xbee จำนวน 4 ตัว

4.3.4 รูปแบบในการทดลองหาค่าตำแหน่ง

จากที่ได้กล่าวไปแล้วในบทที่ 3 ถึงทฤษฎีการหาค่าตำแหน่งแบบการวิเคราะห์บริเวณเฉพาะ (Scene Analysis) บางครั้งเรียกว่า Fingerprint ซึ่งวิธีนี้จะต้องมีการเก็บค่าของสัญญาณก่อนครั้งหนึ่ง แล้วจึงค่อยมีการนำสัญญาณที่ได้มาทำการเปรียบเทียบกับสัญญาณที่ต้องการค้นหาตำแหน่ง โดยใช้ อัลกอริทึมในการเปรียบเทียบ รูปแบบในการทดลองวิจัยนี้จะเป็นดังรูปที่ 4.16 โดยขั้นตอนแรกจะทำการเก็บข้อมูลของค่าความเข้มของสัญญาณที่อ่านค่าได้ในแต่ละตำแหน่ง โดยข้อมูลทั้งหมดนี้จะทำการเก็บไว้ในฐานข้อมูลในช่วงแรกด้วยอัลกอริทึมต่างๆ เมื่อเปรียบเทียบเสร็จแล้วก็จะแสดงตำแหน่งของวัตถุออกมา โดยข้อมูลจะแสดงออกมาเป็นค่าของพิกัดและจุดบนโปรแกรม



รูปที่ 4.16 แสดงการเก็บค่าความเข้มของสัญญาณ

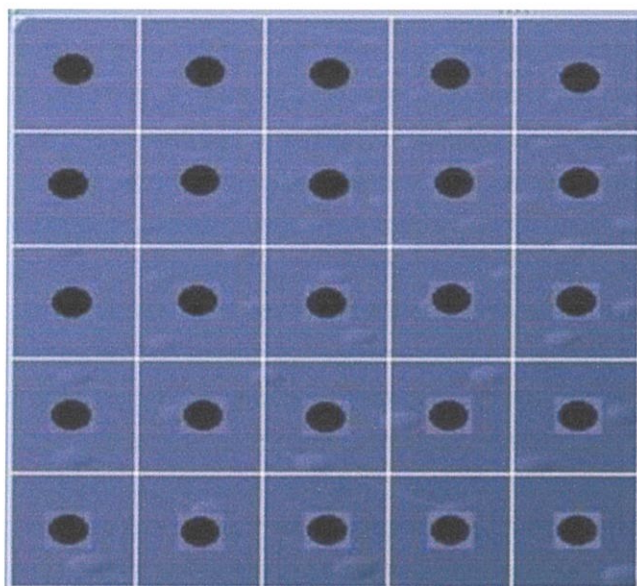
4.3.5 ขั้นตอนในการทดลองหาค่าตำแหน่ง

ในการทดลองหาค่าตำแหน่งโดยการใช้เทคนิคการหาค่าตำแหน่งแบบการวิเคราะห์บริเวณเฉพาะ (Scene Analysis) บางครั้งเรียกว่า Fingerprint ซึ่งวิธีนี้จะสามารถแบ่งการทำงานได้เป็น 2 ขั้นตอนดังต่อไปนี้

4.3.5.1 Offline Phase หรือ Training Phase

ในการทำงานของช่วงนี้จะเป็นช่วงของการเก็บข้อมูลความเข้มของสัญญาณ โดยจะทำการเก็บข้อมูลในแต่ละตำแหน่ง โดยมีขั้นตอนดังต่อไปนี้

- กำหนดตำแหน่งของ Fingerprint ที่จะทำการสำรวจเก็บไว้ในฐานข้อมูล โดยการทดลองโครงการนี้ได้กำหนดตำแหน่งของ Fingerprint โดย (แสดงดังรูปที่ 4.17)
- นำ Referent Node มาวาง ณ ตำแหน่ง Fingerprint แล้วบันทึกค่าความเข้มของสัญญาณที่อ่านได้ในทุกๆตำแหน่งของ Fingerprint ซึ่งจะทำการทดลองซ้ำทั้งหมด 5, 7, 10 รอบแล้วจะนำค่ามาเฉลี่ย



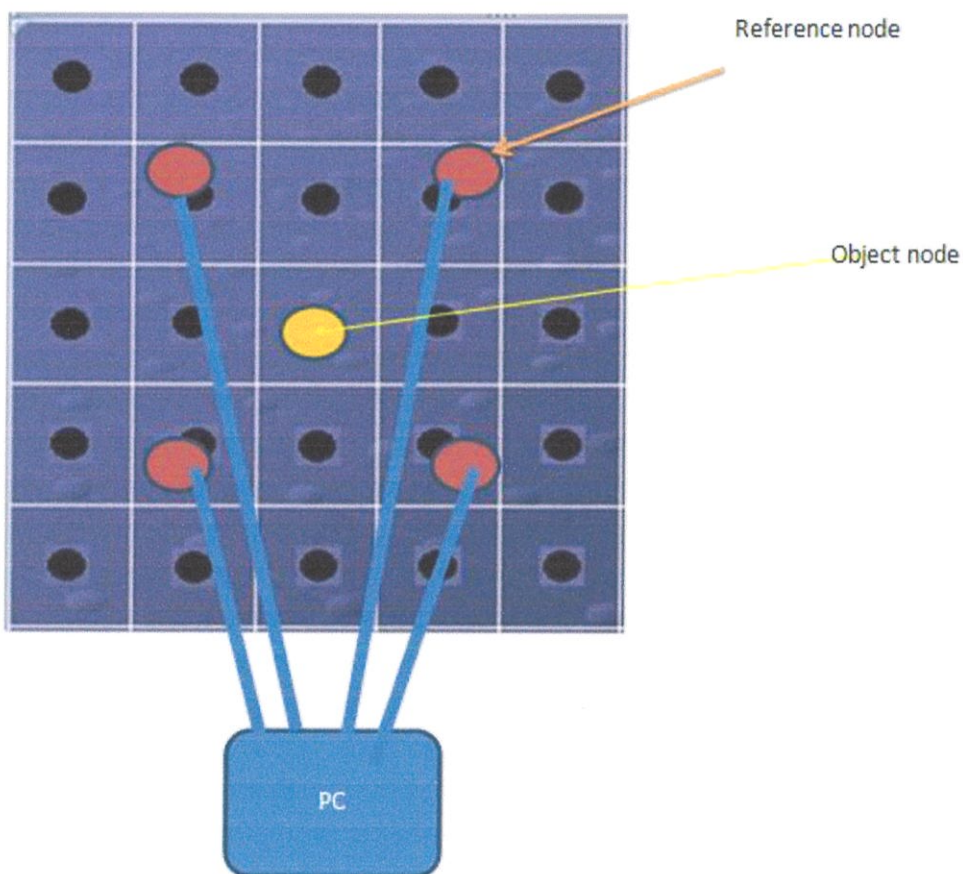
รูปที่ 4.17 แสดงจุดตำแหน่งในการทำ Fingerprint

4.3.5.2 Online Phase หรือ Localization Phase

ในการทำงานช่วงนี้จะเป็นช่วงของการหาตำแหน่ง โดยจะทำการหาตำแหน่งของ Object Node ด้วยการอ่านข้อมูลค่าความเข้มของสัญญาณที่ส่งมา ณ ตำแหน่งที่ Object Node ตั้งอยู่แล้วระบบจะนำค่าที่ได้ไปทำการเปรียบเทียบกับค่าที่ได้เก็บไว้ในฐานข้อมูลในช่วงแรก ด้วยอัลกอริทึมต่างๆ จากนั้นเมื่อเปรียบเทียบเสร็จแล้วก็แสดงตำแหน่งของ Object Node ออกมา โดยจะแสดงผลออกมาเป็นพิกัดตัวเลขและพิกัดจุด โดยมีขั้นตอนการทดลองโครงการดังนี้

การทดลองในโครงการนี้แบ่งเป็นสองส่วนการทดลองคือ 1) การทดลองในพื้นที่ไม่มีสัญญาณ wifi ครอบคลุม
2) การทดลองในพื้นที่ที่มีสัญญาณ wifi ครอบคลุม

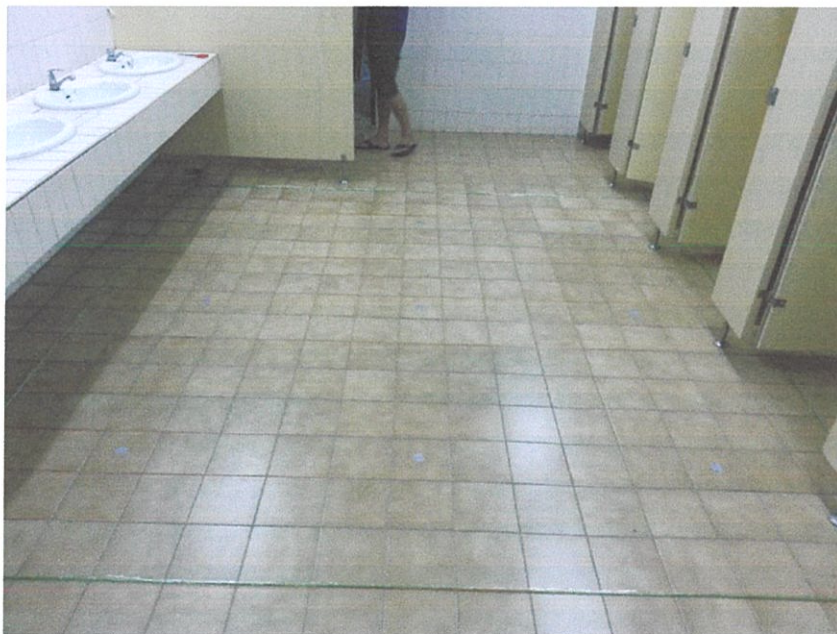
1) การทดลองในพื้นที่ไม่มีสัญญาณ wifi ทำการทดลองที่ ห้องน้ำชาย ชั้น 8 อาคารเรียน 12 ชั้น คณะวิศวกรรมศาสตร์ ทำการทดลองโดยใช้พื้นที่การทดลองขนาด 5x5 ตารางเมตร แบ่งพื้นที่ออกเป็น 25 ช่อง ช่องละ 1 ตารางเมตร และใช้ 4 Reference node ในการทดลอง ติดตั้งให้ Reference node ห่างจากแต่ละมุมของพื้นที่ทดลองในแนวราบ $x = 1$ เมตร และ $y = 1$ เมตร และให้ Reference node อยู่สูงจากพื้นในแนวตั้ง 2.5 เมตร และให้ Object node อยู่สูงจากพื้น 0.5 เมตร



รูปที่ 4.18 จำลองการทดลองในพื้นที่ไม่มีสัญญาณ wifi ครอบคลุม



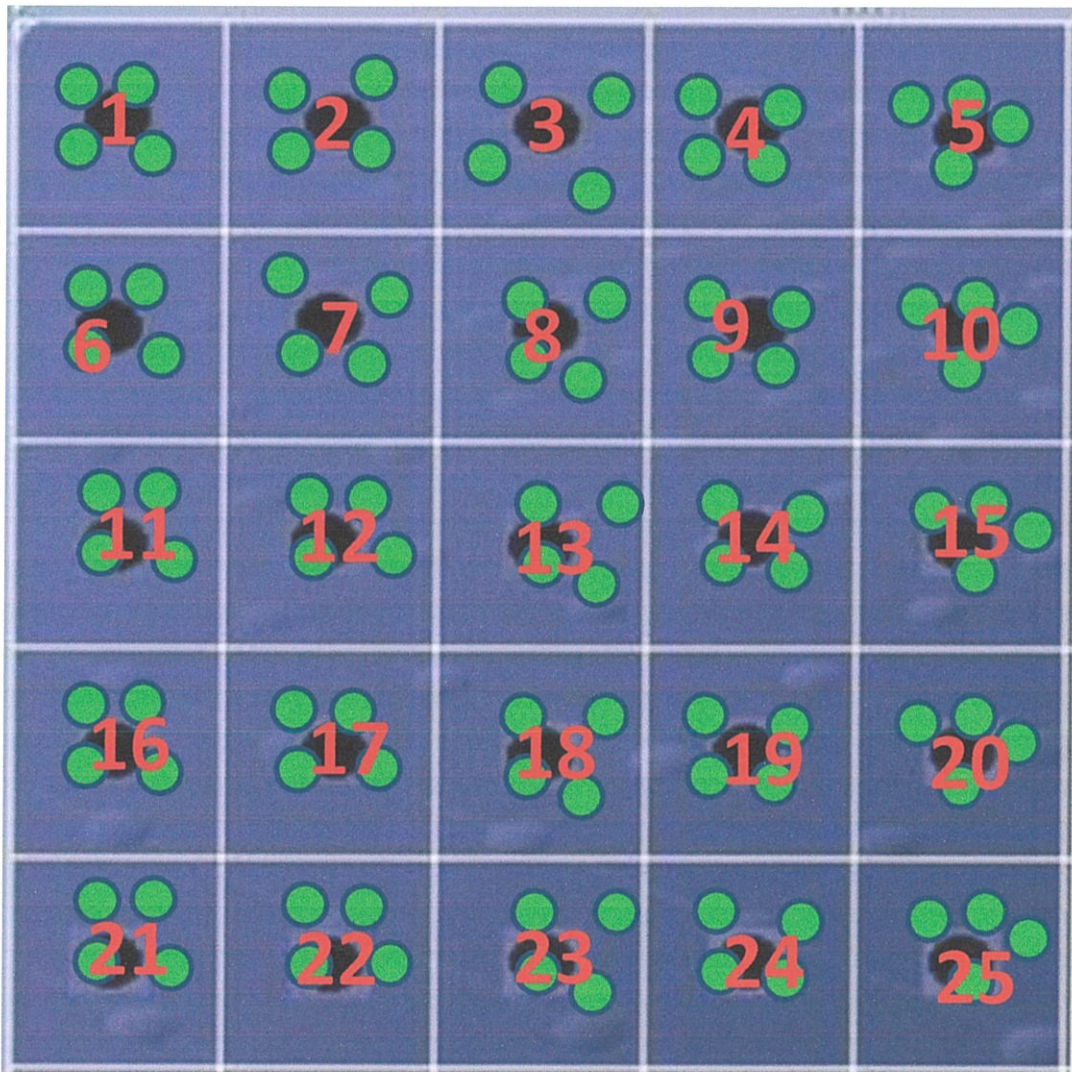
รูปที่ 4.19 การติดตั้งอุปกรณ์



รูปที่ 4.20 พื้นที่ทตลองขนาด5x5ตารางเมตร

จากการคำนวณในแนวตั้งจะได้ระยะห่างมากที่สุดจาก Reference node ถึง Object node 5.7 เมตร ระยะน้อยกว่า 6 เมตร แสดงว่ารับสัญญาณได้ดี เหมาะสมแก่การทดลอง

วิธีการทดลองหาตำแหน่งใช้เทคนิค Fingerprint ตั้งที่กล่าวมาข้างต้นโดยเก็บค่าแต่ละช่องช่องละ 10 ครั้งแล้วนำมาเฉลี่ยกันเป็นค่าพื้นฐานข้อมูลของแต่ละช่อง ทำการทดสอบระบุตำแหน่งโดยการสุ่มทดสอบ 100 ตำแหน่ง

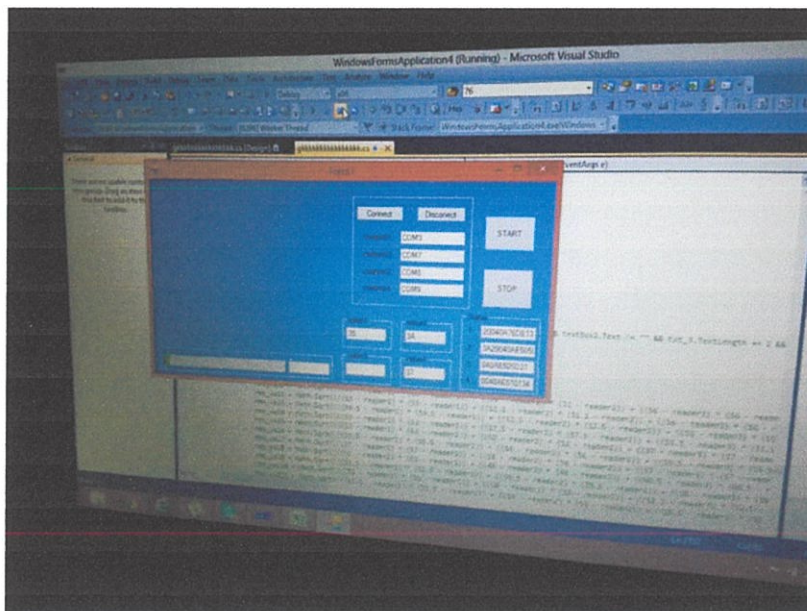


รูปที่ 4.23 ตำแหน่งที่สุ่มทดลอง

สัญลักษณ์ วงกลมสีเขียวคือตำแหน่งปุ่มทดลอง 100 ตำแหน่ง และตัวเลขสีแดงเป็นเลขบอกตำแหน่งช่องในการทดลอง

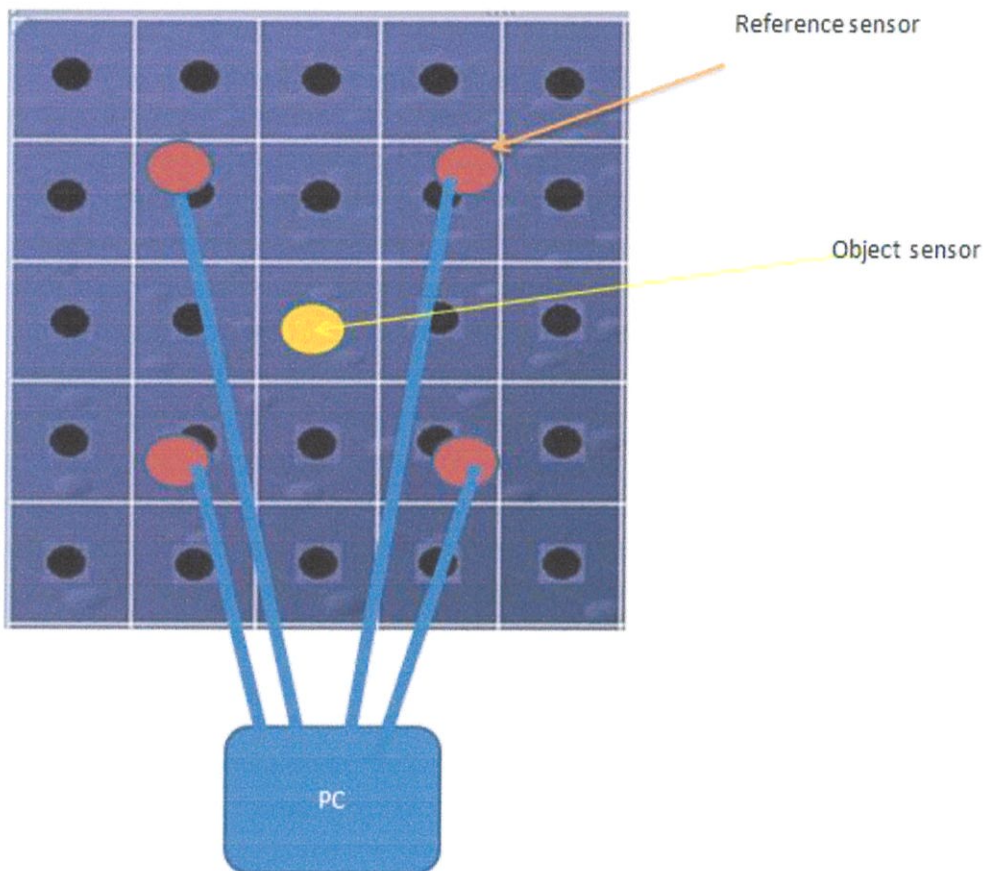


รูปที่ 4.24 รูปการเก็บค่าพื้นฐานและการทดลอง



รูปที่ 4.25 หน้าจอแสดงผลการเก็บค่าสัญญาณพื้นฐาน

- 2) การทดลองในพื้นที่ที่มีสัญญาณ wifi ครอบคลุม ทำการทดลองที่ ลานหน้าลิฟต์ชั้น 4 ตึกวิชาภาค เครื่องกล คณะวิศวกรรมศาสตร์ ทำการทดลองโดยใช้พื้นที่การทดลองขนาด 5x5 ตารางเมตร แบ่งพื้นที่ออกเป็น 25 ช่อง ช่องละ 1 ตารางเมตร และใช้ 4 Reference node ในการทดลอง ติดตั้ง Reference node ให้ห่างจากแต่ละมุมของพื้นที่ทดลองในแนวราบ $x = 1$ เมตร และ $y = 1$ เมตร และให้ Reference node อยู่สูงจากพื้นในแนวตั้ง 2.5 เมตร และให้ Object node อยู่สูงจากพื้น 0.5 เมตร



รูปที่ 4.26 จำลองการทดลองโดยใช้ 4 Reference node

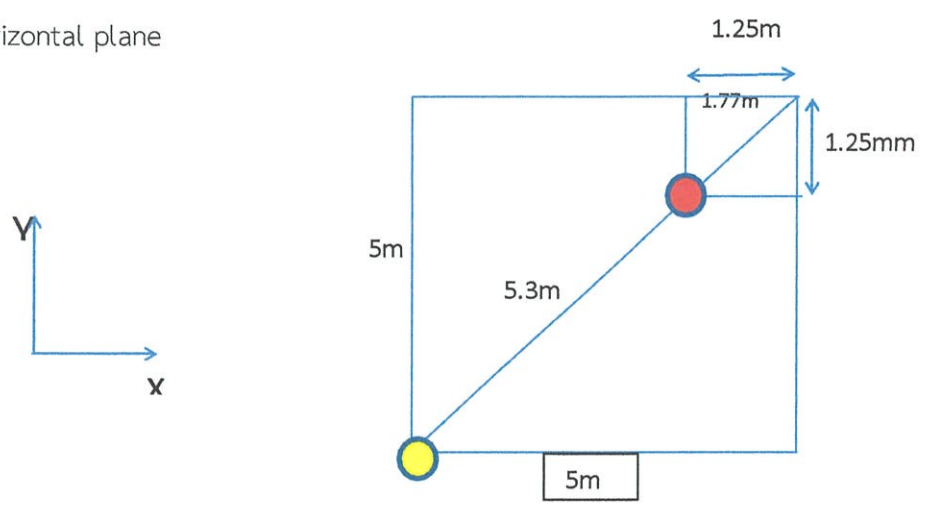


รูปที่ 4.27 ติดตั้งอุปกรณ์



รูปที่ 4.28 พื้นที่ทดลองขนาด 5x5 ตารางเมตร

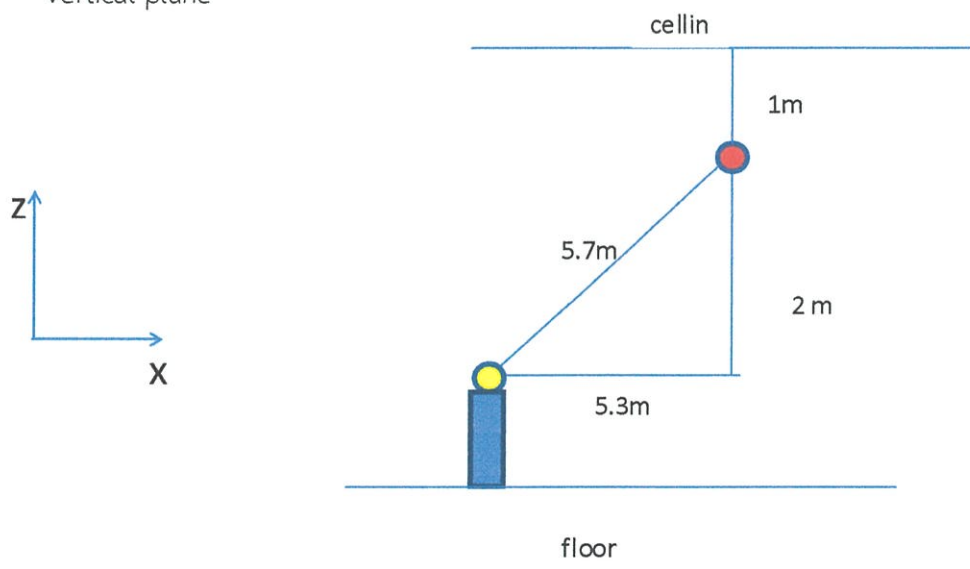
Horizontal plane



รูปที่ 4.29 มองด้านบน

จากการคำนวณในแนวราบจะได้ระยะห่างมากที่สุดจาก Reference node ถึง Object node 5.3 เมตร ระยะน้อยกว่า 6 เมตร แสดงว่ารับสัญญาณได้ดี เหมาะสมแก่การทดลอง

Vertical plane



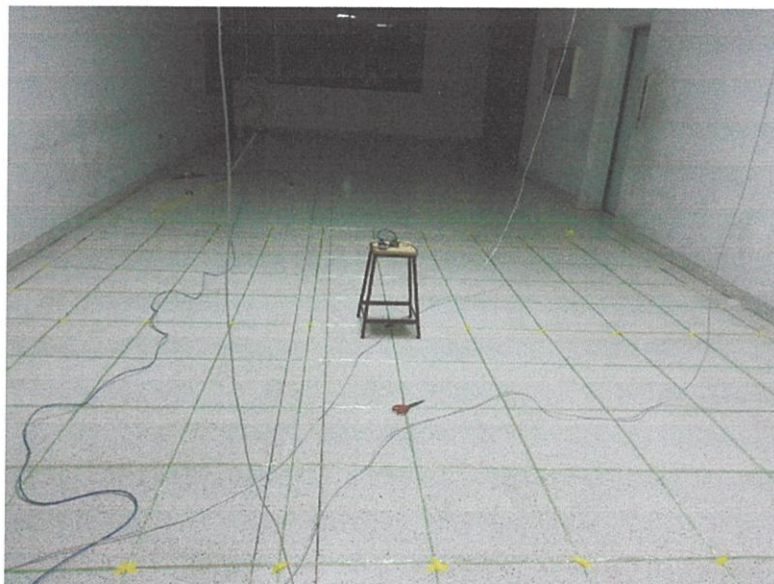
รูปที่ 4.30 มองด้านข้าง

จากการคำนวณในแนวตั้งจะได้ระยะห่างมากที่สุดจาก Reference node ถึง Object node 5.7 m ระยะน้อยกว่า 6 m แสดงว่ารับสัญญาณได้ดี เหมาะสมแก่การทดลอง

วิธีการทดลองหาตำแหน่งใช้เทคนิค Fingerprint ตั้งที่กล่าวมาข้างต้นโดยเก็บค่าแต่ละช่องช่องละ 10 ครั้งแล้วนำมาเฉลี่ยกันเป็นค่าพื้นฐานของแต่ละช่อง ทำการทดสอบระบุตำแหน่งโดยการสุ่มทดสอบ 100 ตำแหน่ง เหมือนกับการทดลองในข้อ 1)



รูปที่ 4.31 การเก็บค่าพื้นฐานแต่ละตาราง



รูปที่ 4.32 การทดลองหาตำแหน่งเป้าหมาย

4.4สรุป

บทนี้จะกล่าวถึงรายละเอียดของวิธีการทดลองเพื่อหาดำแหน่งเป้าหมายในการทดลองโครงการฉบับนี้จะใช้เทคนิค Fingerprint มาทำการระบุตำแหน่งโดยการเปรียบเทียบสัญญาณที่ได้รับมาจาก Reference node กับสัญญาณที่ได้เก็บค่าเป็นฐานข้อมูลในแต่ละช่องโดยทำการทดลอง 2 แบบคือ 1) การทดลองในพื้นที่ไม่มีสัญญาณ wifi รบกวน 2) การทดลองในพื้นที่ที่มีสัญญาณ wifi รบกวนแผนการทดลองและขั้นตอนการทดลองโครงการต้องคำนึงถึงปัจจัยต่างๆ ได้แก่ สถานที่ทำการทดลอง ระยะอ่านสัญญาณของ Xbee ความสูงของ Object node และ Reference node พื้นที่ในการหาดำแหน่งเพื่อให้สอดคล้องกับระยะที่ Xbee สามารถที่จะอ่านสัญญาณได้และเหมาะสมกับการใช้งานจริง ซึ่งผลการทดลองและผลการวิเคราะห์จะกล่าวถึงในบทต่อไป

บทที่ 5

ผลการทดลองแล้วผลการวิเคราะห์การหาตำแหน่ง ของเป้าหมายภายในอาคาร

5.1 กล่าวนำ

ในบทนี้จะกล่าวถึงผลการทดลองและผลการวิเคราะห์ในการหาตำแหน่งเป้าหมายที่ทำการสุ่มทดลองโดยใช้เทคนิค Fingerprint บนพื้นฐานเทคโนโลยี Zigbee ซึ่งในการทดลองนี้ได้ทำการเปรียบเทียบ 1) การระบุตำแหน่งในพื้นที่ไม่มีสัญญาณ wifi ครอบคลุมและ 2) พื้นที่ที่มีสัญญาณ wifi ครอบคลุมทำการทดลองโดยใช้ 4 reference node โดยนำเปอร์เซ็นต์ความถูกต้องที่ได้จากการทดลองในแต่ละข้อมาเปรียบเทียบกัน

5.2 ผลการทดลองในพื้นที่ไม่มีสัญญาณ wifi ครอบคลุม

จากการทดลองที่ได้ทำการสุ่มทดลอง 100 ตำแหน่งและคำนวณหาผลตำแหน่งจากการทดลองโดยการเปรียบเทียบใช้เทคนิค fingerprint จะได้ตำแหน่งที่สามารถคำนวณออกมาได้โดยระบุตำแหน่งออกมาเป็นช่อง ดังตาราง 5.1

ตารางที่ 5.1 ตำแหน่งที่ได้จากการทดลองในข้อ 1)

จำนวนครั้งที่ทดลอง	ช่องสุ่มทดลอง (ช่องที่)	ช่องที่โปรแกรมแสดง (ช่องที่)	ความถูกต้อง
1	6	6	T
2	15	15	T
3	6	6	T
4	8	8	T
5	16	16	T
6	1	1	T
7	3	3	T
8	24	24	T
9	24	24	T
10	20	20	T

ตารางที่ 5.1 (ต่อ) ตำแหน่งที่ได้จากการทดลองในข้อ 1)

จำนวนครั้งที่ทดลอง	ช่องสุ่มทดลอง (ช่องที่)	ช่องที่โปรแกรมแสดง (ช่องที่)	ความถูกต้อง
11	12	12	T
12	25	25	T
13	24	24	T
14	10	10	T
15	8	8	T
16	7	7	T
17	11	12	F
18	22	22	T
19	14	14	T
20	1	1	T
21	7	7	T
22	13	13	T
23	23	23	T
24	18	18	T
25	8	8	T
26	6	6	T
27	15	15	T
28	15	15	T
29	19	19	T
30	13	12	F
31	1	1	T
32	22	22	T
33	7	7	T
34	9	9	T
35	22	22	T
36	5	5	T
37	6	6	T
38	21	21	T
39	14	14	T
40	13	13	T
41	11	11	T
42	7	7	T
43	4	9	F
44	11	11	T
45	4	4	T
46	4	4	T
47	20	20	T
48	9	9	T
49	5	5	T
50	10	10	T
51	24	24	T

ตารางที่ 5.1 (ต่อ) ตำแหน่งที่ได้จากการทดลองในข้อ 1)

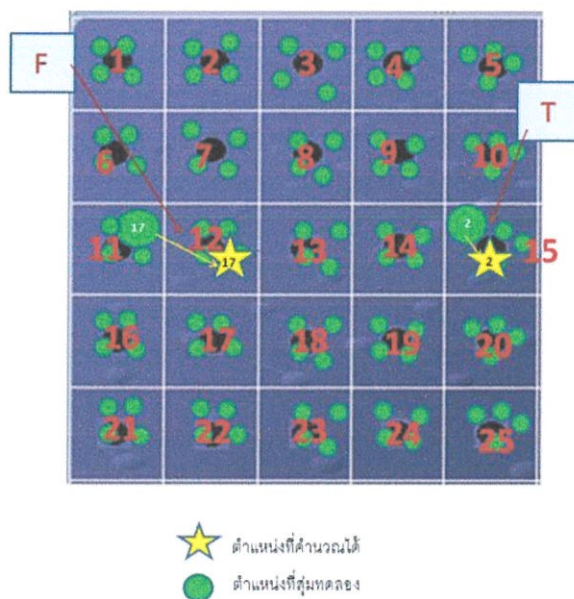
จำนวนครั้งที่ทดลอง	ช่องสุ่มทดลอง (ช่องที่)	ช่องที่โปรแกรมแสดง (ช่องที่)	ความถูกต้อง
52	18	18	T
53	9	9	T
54	9	9	T
55	17	17	T
56	25	25	T
57	19	19	T
58	23	23	T
59	10	10	T
60	21	21	T
61	21	21	T
62	16	16	T
63	2	3	F
64	2	2	T
65	14	14	T
66	3	3	T
67	8	8	T
68	16	16	T
69	25	25	T
70	15	15	T
71	19	19	T
72	20	20	T
73	3	3	T
74	1	1	T
75	18	18	T
76	5	5	T
77	21	21	T
78	25	25	T
79	2	7	F
80	19	18	F
81	17	17	T
82	23	23	T
83	17	17	T
84	12	12	T
85	5	5	T
86	20	20	T
87	12	11	F
88	11	11	T
89	22	22	T
90	17	17	T
91	3	3	T
92	4	4	T

ตารางที่ 5.1 (ต่อ) ตำแหน่งที่ได้จากการทดลองในข้อ 1)

จำนวนครั้งที่ทดลอง	ช่องสุ่มทดลอง (ช่องที่)	ช่องที่โปรแกรมแสดง (ช่องที่)	ความถูกต้อง
93	10	10	T
94	12	12	T
95	16	16	T
96	14	14	T
97	2	2	T
98	13	13	T
99	18	18	T
100	23	23	T

ผลการทดลองที่ได้ จากการสุ่มทดลอง 100 ครั้ง จะมี 93 ครั้งที่ทำการระบุตำแหน่งได้ถูกต้อง เนื่องจากสัญญาณของ Zigbee ที่อ่านได้ในที่ไม่มีสัญญาณ wifi มีความแม่นยำและค่อนข้างคงที่ และมี 7 ครั้งที่ทำการระบุตำแหน่งไม่ถูกต้อง อาจเป็นเพราะว่ามีสัญญาณรบกวนจากแหล่งอื่นๆผ่านเข้ามาในพื้นที่ทดลองชั่วคราว เช่น คลื่นวิทยุ คลื่นโทรศัพท์ และคลื่นสัญญาณจากเครื่องบินบินผ่าน เป็นต้น

จากตารางที่ 5.1 เมื่อนำตำแหน่งสุ่มที่ 2 และ 17 ในช่องที่ 15 กับ 11 มาพล็อตแผนผังเปรียบเทียบตำแหน่งระหว่างตำแหน่งจริงที่สุ่มทดลองและตำแหน่งจากการทดลองแสดงผลออกมาเป็นช่องได้ดังรูป 5.1



รูปที่ 5.1 เปรียบเทียบตำแหน่งจริงกับตำแหน่งที่คำนวณได้

สัญลักษณ์วงกลมสีเขียว คือ ตำแหน่งจริงที่สุ่มทดลอง ส่วนดาวสีเหลือง คือ ตำแหน่งที่ได้จากการทดลอง เส้นที่เชื่อมต่อระหว่างวงกลมกับดาว เป็นเส้นแสดงความสัมพันธ์ระหว่างตำแหน่งจริงและตำแหน่งที่ได้จากการทดลอง จากแผนผังการสุ่มทดลองครั้งที่ 2 วางเป้าหมายหรือ object node ในช่องที่ 15 ผลการระบุตำแหน่งแสดงผลออกมาในช่อง 15 แสดงว่าการระบุตำแหน่งถูกต้อง (T) และการสุ่มทดลองครั้งที่ 17 วางเป้าหมายในช่องที่11ผลการระบุตำแหน่งแสดงผลออกมาในช่องที่12แสดงว่าผลการระบุตำแหน่งไม่ถูกต้อง (F) ที่จะหาเปอร์เซ็นต์ความถูกต้องโดยใช้สมการ

$$\text{เปอร์เซ็นต์ความถูกต้อง} = \left(\frac{\text{จำนวนครั้งที่ถูกต้อง}}{\text{จำนวนครั้งที่ทดลองทั้งหมด}} \right) \times 100$$

$$\text{เปอร์เซ็นต์ความถูกต้อง} = 93\%$$

5.3 การทดลองในที่ที่มีสัญญาณwifirบกวน

จากการทดลองที่ได้ทำการสุ่มทดลอง 100 ตำแหน่งและแล้วคำนวณหาตำแหน่งโดยการเปรียบเทียบใช้เทคนิค Fingerprint จะได้ตำแหน่งที่สามารถคำนวณออกมาเป็นช่อง ดังตาราง 5.2 ตารางที่5.2 พิกัดที่คำนวณได้จากการทดลองโดยใช้

จำนวนครั้งที่ทดลอง	ช่องที่สุ่มทดลอง (ช่องที่)	ช่องที่โปรแกรมแสดง (ช่องที่)	เลือกตำแหน่งที่ดีที่สุด (ช่องที่)	ความถูกต้อง
1	6	5, 12, 25	12	F
2	15	23, 17, 13	13	F
3	6	2, 11, 25	11	F
4	8	1, 13, 7	13	F
5	16	25, 11, 15	11	F
6	1	8, 6, 17	6	F
7	3	10, 9, 21	9	F
8	24	18, 19, 14	19	F
9	24	13, 19, 10	19	F
10	20	19, 15, 10	15	F
11	12	9, 18, 2	18	F
12	25	10, 19, 5	19	F
13	24	7, 19, 4	19	F

ตารางที่ 5.2 (ต่อ) พิกัดที่คำนวณได้จากการทดลองโดยใช้

จำนวนครั้งที่ทดลอง	ช่องที่สุ่มทดลอง (ช่องที่)	ช่องที่โปรแกรมแสดง (ช่องที่)	เลือกตำแหน่งที่ดีที่สุด (ช่องที่)	ความถูกต้อง
14	10	8, 5, 25	5	F
15	8	4, 2, 11	2	F
16	7	19, 7, 21	7	T
17	11	1, 11, 15	11	T
18	22	7, 17, 21	17	F
19	14	16, 9, 25	9	F
20	1	25, 6, 21	6	F
21	7	16, 12, 25	12	F
22	13	13, 22, 1	13	T
23	23	1, 14, 18	18	F
24	18	1, 12, 25	12	F
25	8	5, 14, 21	14	F
26	6	5, 11, 19	11	F
27	15	7, 20, 21	20	F
28	15	6, 10, 25	10	F
29	19	19, 24	19	T
30	13	1, 8, 21	8	F
31	1	7, 19, 21	7	F
32	22	18, 24, 15	18	F
33	7	1, 7, 19	7	T
34	9	1, 14, 24	14	F
35	22	6, 17, 24	17	F
36	5	1, 10, 19	10	F
37	6	1, 12, 21	12	F
38	21	2, 21, 25	21	T
39	14	7, 20, 21	20	F
40	13	1, 7, 24	7	F
41	11	16, 19, 24	16	F
42	7	1, 7, 19	7	T
43	4	2, 9, 20	9	F
44	11	5, 21, 19	5	F
45	4	2, 1, 9	9	F
46	4	10, 25, 17	10	F
47	20	10, 14, 24	14	F
48	9	3, 21, 25	3	F
49	5	3, 11, 25	11	F
50	10	3, 10, 17	10	T

ตารางที่ 5.2 (ต่อ) พิกัดที่คำนวณได้จากการทดลองโดยใช้

จำนวนครั้งที่ทดลอง	ช่องที่สุ่มทดลอง (ช่องที่)	ช่องที่โปรแกรมแสดง (ช่องที่)	เลือกตำแหน่งที่ดีที่สุด (ช่องที่)	ความถูกต้อง
51	24	4, 18, 17	18	F
52	18	1, 18, 19	18	T
53	9	3, 19, 20	3	F
54	9	1, 4, 24	4	F
55	17	1, 11, 19	11	F
56	25	7, 20, 21	20	F
57	19	13, 24	13	F
58	23	17, 19, 21	17	F
59	10	1, 10, 15	10	T
60	21	11, 15, 19	15	F
61	21	1, 21, 25	21	T
62	16	11, 21, 25	21	F
63	2	7, 21, 22	7	F
64	2	8, 19, 21	8	F
65	14	7, 14, 21	14	T
66	3	1, 8, 24	8	F
67	8	5, 2, 22	2	F
68	16	10, 19, 24	21	F
69	25	1, 15, 25	25	T
70	15	5, 20, 25	20	F
71	19	13, 21, 25	13	F
72	20	5, 14, 24	14	F
73	3	1, 8, 18	8	F
74	1	6, 19, 20	6	F
75	18	5, 12, 21	12	F
76	5	5, 18, 24	5	T
77	21	5, 16, 25	16	F
78	25	3, 18, 25	25	T
79	2	5, 8, 24	8	F
80	19	5, 1, 13	13	F
81	17	7, 22, 19	22	F
82	23	8, 18, 22	18	F
83	17	13, 17, 18	17	T
84	12	14, 17, 25	17	F
85	5	1, 10, 18	10	F
86	20	7, 14, 25	14	F
87	12	6, 17, 21	6	F
88	11	5, 19, 21	5	F
99	22	4, 17, 25	17	F
90	17	5, 17, 22	17	T

ตารางที่ 5.2 (ต่อ) พิกัดที่คำนวณได้จากการทดลองโดยใช้

จำนวนครั้งที่ทดลอง	ช่องที่สุ่มทดลอง (ช่องที่)	ช่องที่โปรแกรมแสดง (ช่องที่)	เลือกตำแหน่งที่ดีที่สุด (ช่องที่)	ความถูกต้อง
91	3	8, 19, 13	8	F
92	4	1, 9, 25	9	F
93	10	2, 10	10	T
94	12	1, 6, 23	6	F
95	16	9, 10, 24	10	F
96	14	3, 14, 17	14	T
97	2	8, 9, 24	8	F
98	13	1, 19, 22	19	F
99	18	9, 23, 24	23	F
100	23	10, 18, 25	18	F

ผลการทดลองจะระบุตำแหน่งออกมาหลายช่อง เนื่องจากสัญญาณของ Zigbee มีการเปลี่ยนแปลงตลอดเวลาประกอบกับมีสัญญาณรบกวนเช่น wifi และสัญญาณจากแหล่งอื่นๆผ่านเข้ามาในพื้นที่ทดลอง จึงทำให้ช่องที่ระบุตำแหน่งเปลี่ยนไปตามสัญญาณของ Zigbee ที่เปลี่ยนไป จึงนำช่องที่มีความใกล้เคียงกับเป้าหมายที่สุดมีวิเคราะห์ผล

ผลการทดลองที่ได้ จากการสุ่มทดลอง 100 ครั้ง จะมี 19 ครั้งที่ทำการระบุตำแหน่งได้ถูกต้อง อาจเพราะว่าสัญญาณ wifi และสัญญาณรบกวนอื่น ๆ มีผลกระทบต่อสัญญาณของ zigbee ในขณะนั้นค่อนข้างน้อย และมี 81 ครั้งที่ทำการระบุตำแหน่งไม่ถูกต้อง เพราะว่ามีสัญญาณรบกวนจาก wifi และแหล่งอื่นๆรบกวนสัญญาณ zigbee ความคลาดเคลื่อนจากเป้าหมายจะมากหรือน้อยขึ้นอยู่กับสัญญาณรบกวน จากการทดลองในพื้นที่ที่มีสัญญาณ wifi จำนวนเปอร์เซ็นต์ความถูกต้องออกมาได้ 19%

5.4 สรุป

ในบทนี้ได้กล่าวถึงผลการทดลองและผลการวิเคราะห์ในการหาตำแหน่งเป้าหมายที่ทำการสุ่มทดลองภายในอาคารโดยใช้เทคโนโลยี zigbee โดยผลการทดลองที่ได้จากการคำนวณโดยเทคนิค Fingerprint ในแต่ละการทดลองจะนำมาคำนวณหาเปอร์เซ็นต์ความถูกต้องซึ่งบทสรุปและข้อเสนอแนะของโครงการนี้จะกล่าวถึงในบทต่อไป

บทที่ 6

บทสรุปและข้อเสนอแนะ

6.1 กล่าวนำ

ในปัจจุบันได้มีการนำเทคโนโลยี Zigbee มาใช้กันอย่างแพร่หลายทั้งในอาคารและนอกอาคาร และยังเป็นเป็นเทคโนโลยีที่สามารถช่วยเพิ่มความสะดวกและความรวดเร็วแก่ภาคธุรกิจและอุตสาหกรรม ในโครงการฉบับนี้ได้มีการนำเทคโนโลยี Zigbee มาประยุกต์ใช้งานในการระบุตำแหน่งของเป้าหมาย ภายในอาคารโดยปกติแล้วการหาตำแหน่งภาคในอาคารมีการใช้เทคโนโลยีการหาตำแหน่งได้หลาย ประเภทอาทิเช่น Wireless local area network, Wifi network, rfid แต่ในที่นี้ได้เลือกใช้เทคโนโลยี Zigbee Series 1 ความถี่ 2.4 Ghz มาทดลองด้วยเหตุผลที่ว่า ราคาต่ำ ขนาดเล็ก และมีระยะส่งสัญญาณ ที่ไกล โดยให้ Object node ติดกับวัตถุเป้าหมายและให้ Reference node เป็นตัวอ้างอิงในการเก็บค่า สัญญาณและใช้เทคนิค Fingerprint ในการระบุตำแหน่งซึ่งในโครงการนี้เสนอการทดลองสองแบบคือ 1) การทดลองในที่ที่ไม่มีสัญญาณwifiรบกวนและ 2) การทดลองในที่ที่มีสัญญาณwifiรบกวน ผลสรุปของของ โครงการจะกล่าวถึงในหัวข้อถัดไป

6.2 บทสรุป

จากการศึกษาวิธีการที่ใช้ในระบบการหาตำแหน่งเป้าหมายภายในอาคารบนพื้นฐานเทคโนโลยี Zigbee โดยใช้เทคนิคการหาตำแหน่งแบบ Fingerprint ในการทดลองนี้จะทำการหาตำแหน่งของ Object none เป็น Zigbee node ที่ติดกับวัตถุที่ต้องการหาตำแหน่งในขณะที่เคลื่อนที่หรืออยู่นิ่งก็ได้ใน กรณีที่เคลื่อนที่จะประมวลผลตำแหน่งช้ากว่าในขณะที่วัตถุอยู่นิ่งและมี Reference node เป็นตัวอ้างอิง ในการรับสัญญาณ ซึ่งในโครงการนี้ได้ทำการทดลอง 2 แบบในแบบแรกจะทดลองในที่ที่ไม่มีสัญญาณ wifi รบกวน ขนาดพื้นที่ทดลอง 5x5 ตารางเมตรและแบบที่สองจะทดลองในที่ที่มีสัญญาณ wifi รบกวน ขนาด พื้นที่ทดลอง 5x5 ตารางเมตร จากผลที่ได้ทำการทดลองในพื้นที่ที่ไม่มีสัญญาณ wifi รบกวน มี 6 ครั้งจาก การสุ่มทดลอง 100 ครั้ง มีไม่ถูกต้องในการระบุตำแหน่ง เนื่องจากการทดลองในครั้งนั้นมีสัญญาณรบกวน จากแหล่งอื่นๆผ่านเข้ามาในพื้นที่ชั่วคราวขณะที่ สามารถคำนวณเปอร์เซ็นต์ความถูกต้องออกมาได้ 93% จากการ

ทดลองในพื้นที่ที่มีสัญญาณ wifi รบกวน ผลการระบุตำแหน่งจะได้ออกมาหลายช่องเนื่องจากสัญญาณมีการเปลี่ยนแปลงตลอดเวลาประกอบกับมีสัญญาณจากแหล่งอื่นๆเข้ามารบกวนสัญญาณ Zigbee จึงเลือกตำแหน่งที่ใกล้เคียงที่สุดมาวิเคราะห์จะเห็นได้ว่ามี 19 ครั้งจากการสุ่มทดลอง 100 ครั้งสามารถระบุตำแหน่งออกมาได้ถูกต้อง อาจเพราะในขณะที่ทดลองในครั้งนั้นๆสัญญาณรบกวนมีผลกระทบต่อสัญญาณ Zigbee น้อยกว่าครั้งอื่นๆ สามารถคำนวณเปอร์เซ็นต์ความถูกต้องออกมาได้ 19% จะเห็นได้ว่าการทดลองในพื้นที่ที่ไม่มีสัญญาณรบกวนอาทิเช่น wifi และสัญญาณจากแหล่งอื่นๆการระบุตำแหน่งจะมีความถูกต้องของตำแหน่งมาก แต่ความถูกต้องจะน้อยลงเมื่ออยู่ในพื้นที่ที่มี wifi และสัญญาณอื่นๆ

6.3 ปัญหาที่พบในการทดลอง

ปัญหาที่พบในการทดลองอย่างแรกคือ ระยะอ่านสัญญาณ Zigbee ใกล้กว่าที่ระบุไว้ในข้อมูลอุปกรณ์ จึงได้มีการกำหนดขนาดพื้นที่ทดลองให้สอดคล้องกับระยะอ่านสัญญาณที่ทดลองได้ Zigbee รบกวนสัญญาณได้ง่ายมาก หากสถานที่ทดลองในส่วนพื้นที่ไร้สัญญาณรบกวนได้ยาก การทดลองในพื้นที่มีสัญญาณรบกวนได้ผลการทดลองหลายตำแหน่งไม่คงที่ เพราะสัญญาณมีการเปลี่ยนแปลงตลอดเวลา และยังทำให้ตำแหน่งที่ได้คลาดเคลื่อนไปจากความเป็นจริง วิเคราะห์ผลโดยการนำตำแหน่งที่ใกล้เคียงที่สุดมาพิจารณา

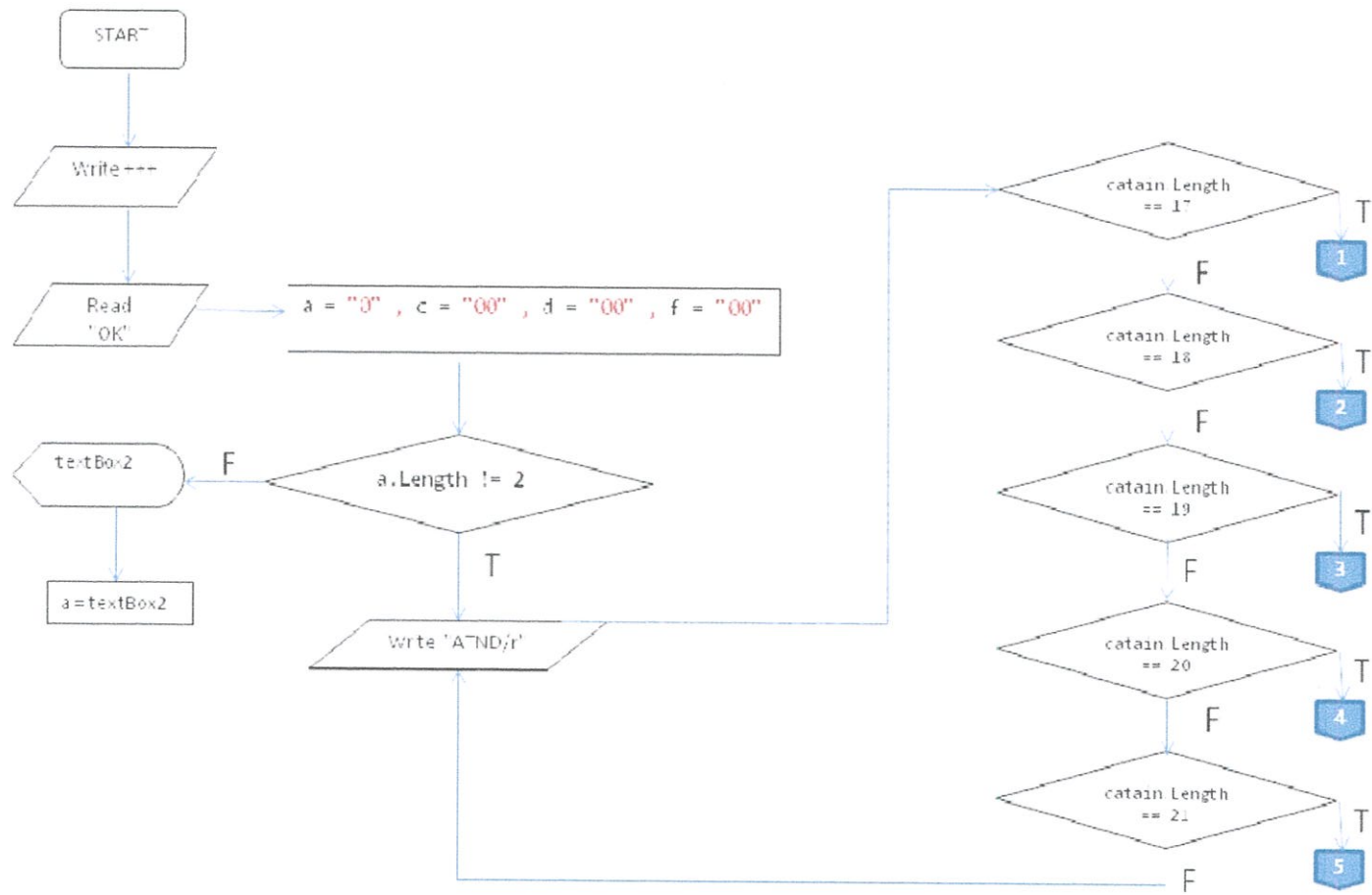
6.4 ข้อเสนอแนะ

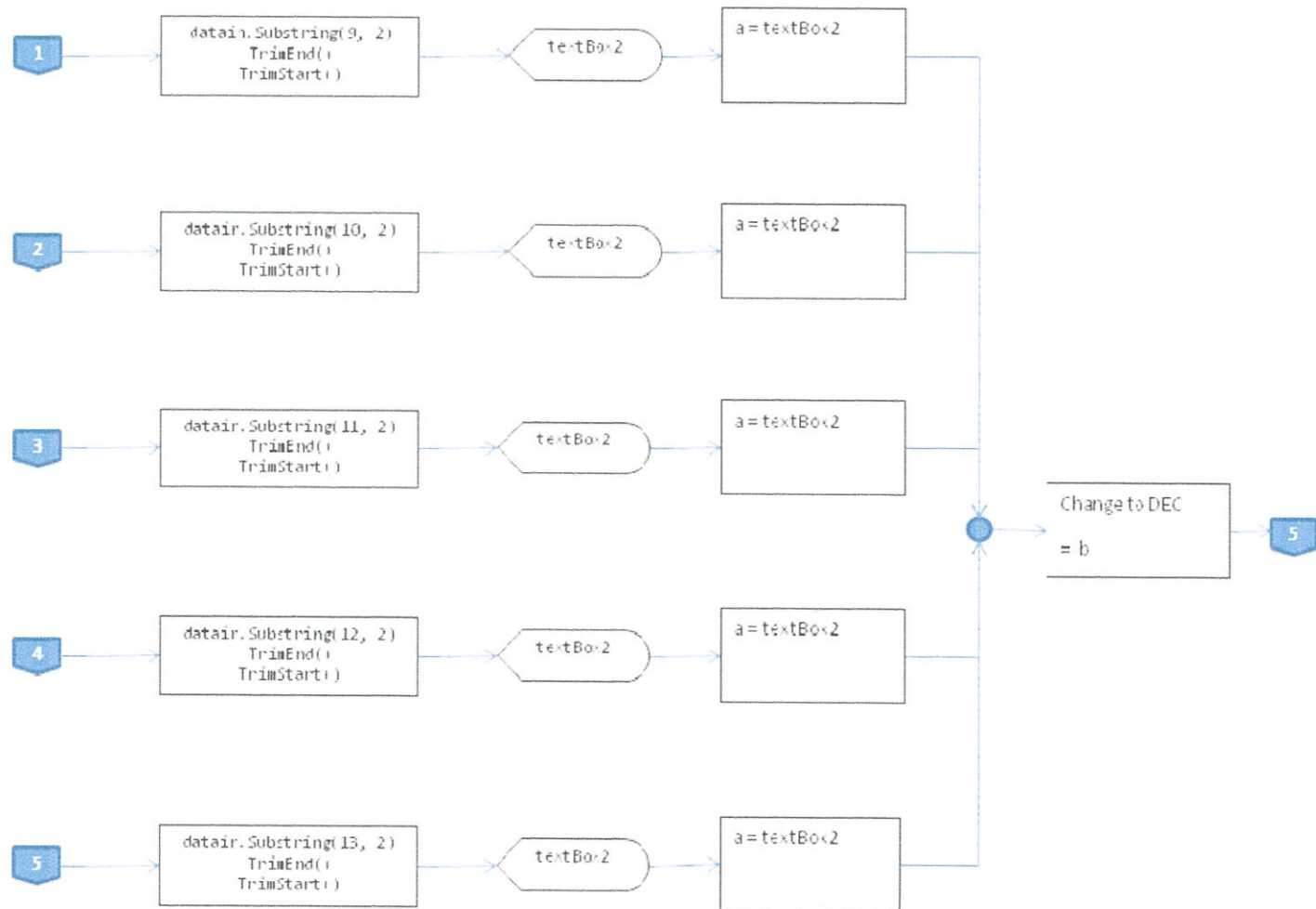
ในการทดลองโครงการนี้มีการใช้ USB ในการเชื่อมต่อระหว่าง Reference node กับคอมพิวเตอร์และระยะรับสัญญาณของ Xbee series 1 สั้นในการพัฒนาโครงการควรจะเป็นการเชื่อมต่อแบบไร้สายจะมีความสะดวกในการติดตั้งและใช้งานมากขึ้น โดยทำการเปลี่ยนจาก Xbee เป็น Xbee Pro จะได้ระยะรับสัญญาณที่ไกลมากขึ้น แต่ทั้งนี้ขึ้นอยู่กับงบประมาณของผู้ทำโครงการด้วย สุดท้ายนี้ผู้จัดทำหวังว่าโครงการฉบับนี้จะมีประโยชน์และสามารถที่จะนำเทคโนโลยี Xbee มาประยุกต์หาตำแหน่งของเป้าหมายในอาคารได้ต่อไปในอนาคต

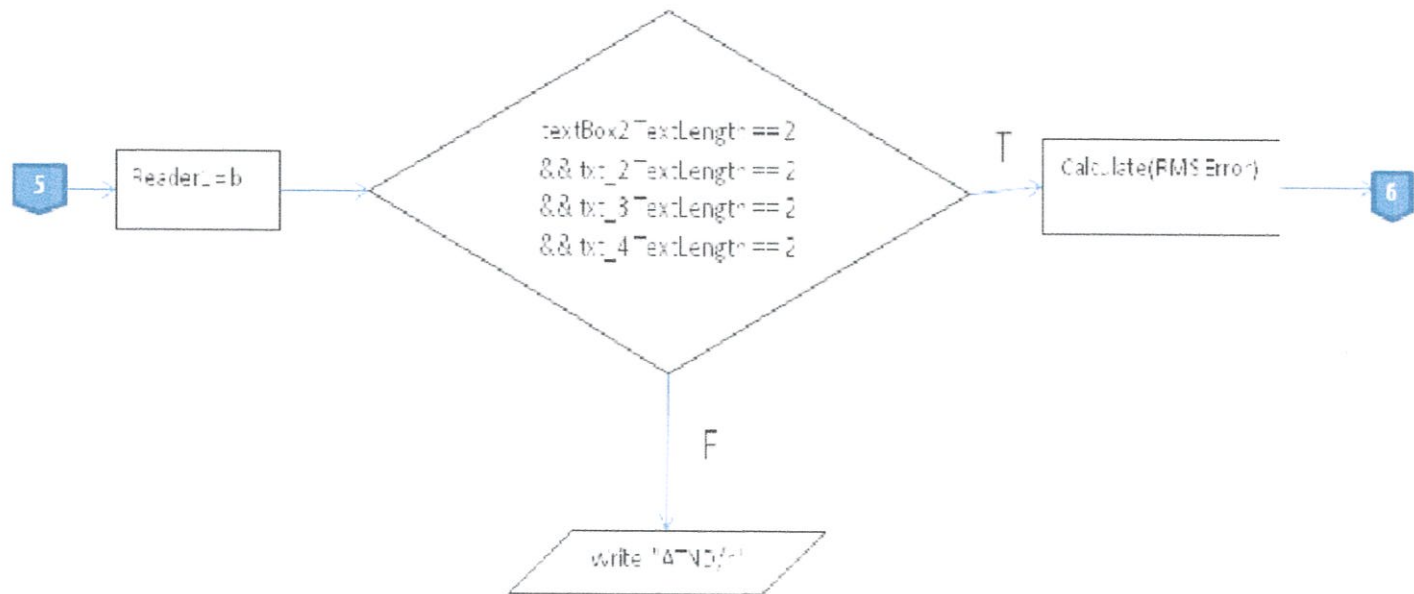
เอกสารอ้างอิง

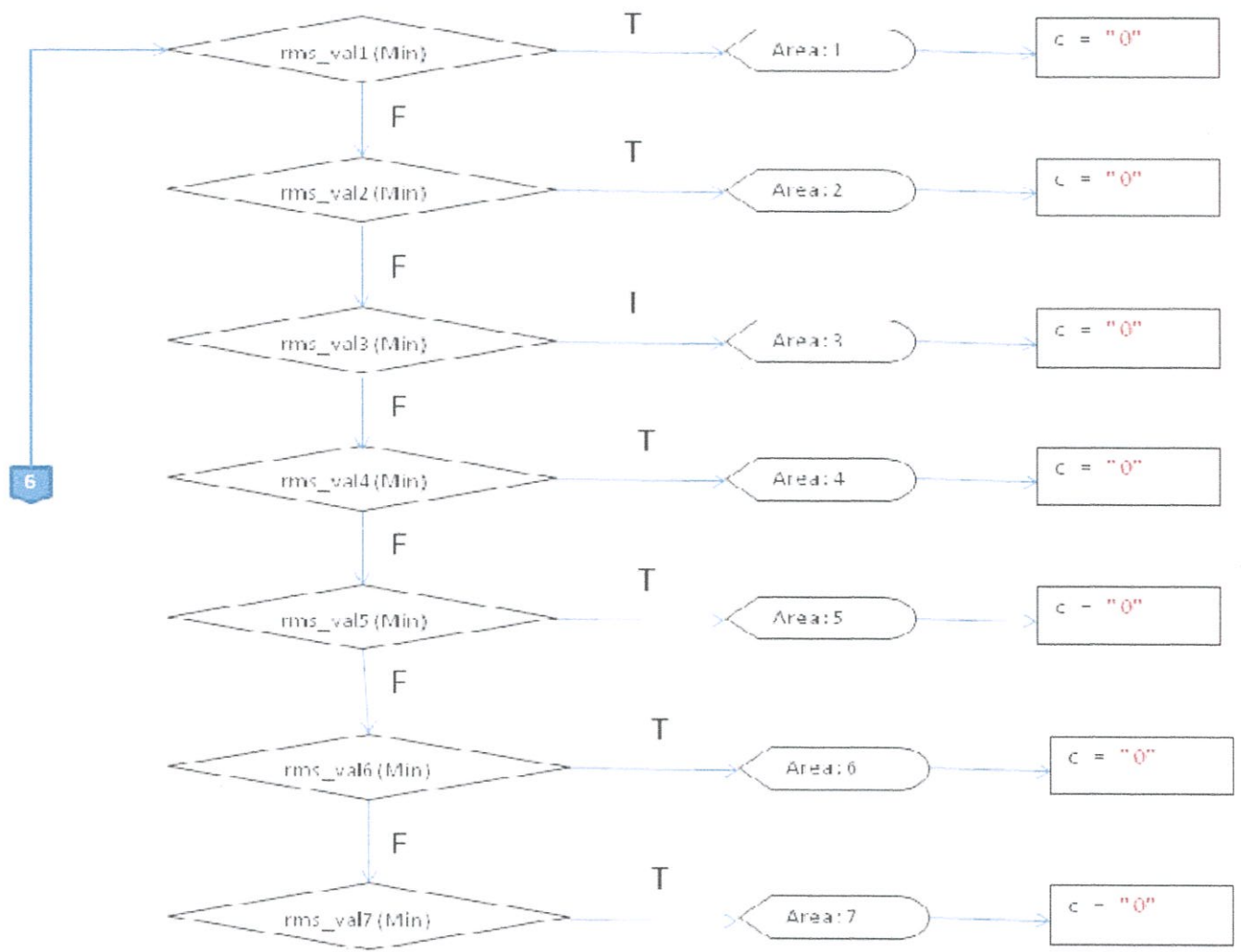
- [1] ภาคย สธนเสาวภาคย, ชัชชัย คุณบัว มหาวิทยาลัยขอนแก่น
ระบบระบุตำแหน่งวัตถุภายในอาคารโดยใช้มาตรฐาน IEEE 802.15.4
- [2] A.M.Ladd, K.E.Bekris, G.Marceau, A.Rudys, L.E.Kavraki, D.S.Wallach. 2002.
"Robotics-based location sensing using wireless Ethernet," ACM
International Conference on Mobile Computing and Networking(MOBICOM'02).
- [3] Abdalkarim Awad, Thorsten Frunzke, Falko
Dressler. 2007. "Adaptive Distance Estimation and Localization in WSN using
RSSI Measures," Proceedings of 10th EUROMICRO Conference on Digital System
Design Architectures .
- [4] Widyawan, Martin Klepal, Dirk Pesch. 2007.
" Influence of Predicted and Measured Fingerprint on the Accuracy of RSSI-
based Indoor Location Systems ", WPNC.
- [5] [http://www.thaieasyelec.com/Embedded-Electronics-Application/Xbee-Basic-
Configuration-in-Network-Application.html](http://www.thaieasyelec.com/Embedded-Electronics-Application/Xbee-Basic-Configuration-in-Network-Application.html)
access on 14/08/2013
- [6] http://www.ecti-thailand.org/assets/papers/4_pub_1.pdf
access on 12/01/2014
- [7] <https://dlnmh9ip6v2uc.cloudfront.net/learn/materials/29/22AT%20Commands.pdf>
access on 14/01/2014
- [8] <http://alselectro.wordpress.com/category/xbee-radios/page/2/>
access on 16/01/2014

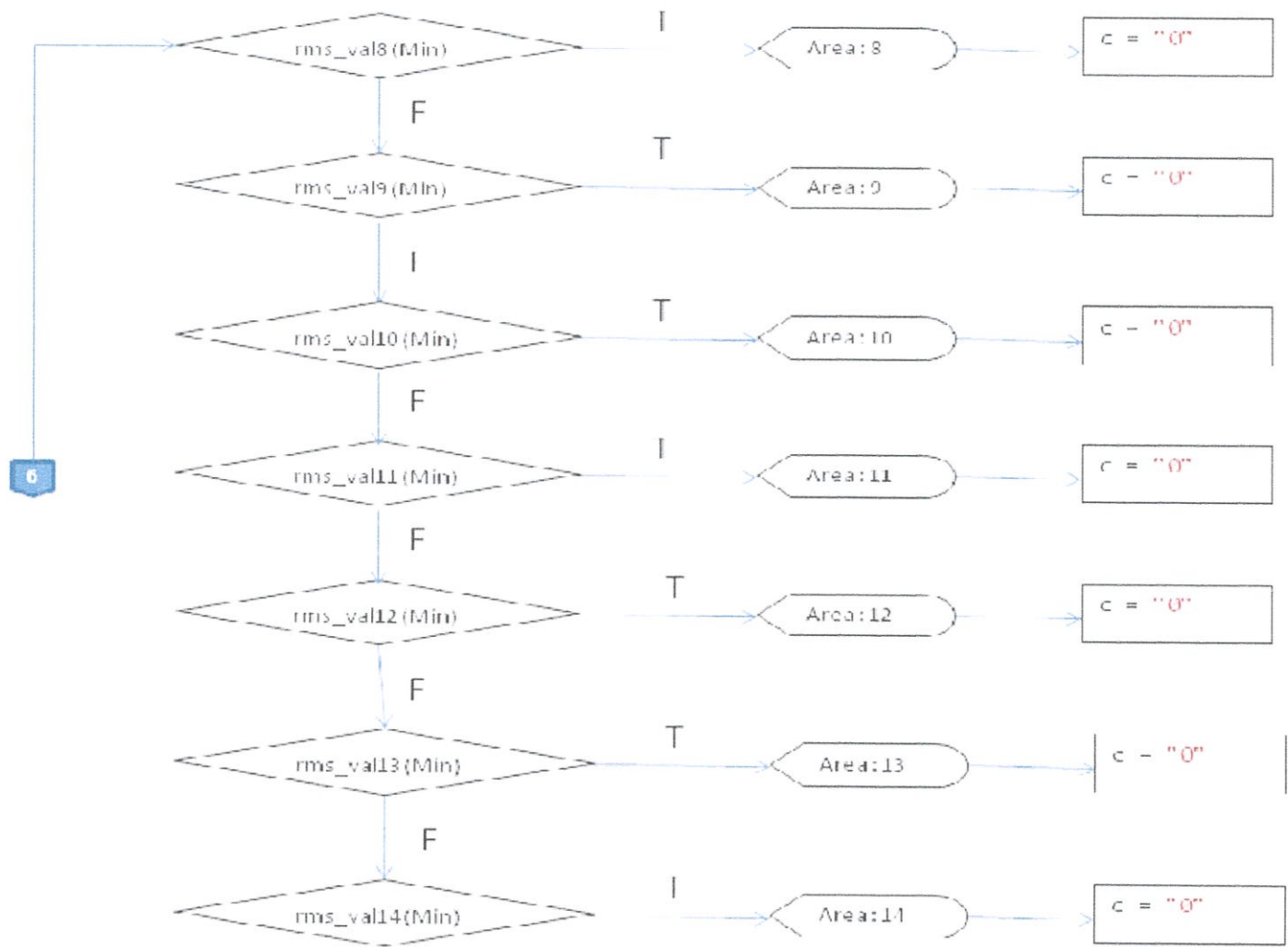
ภาคผนวก
Flow Chart

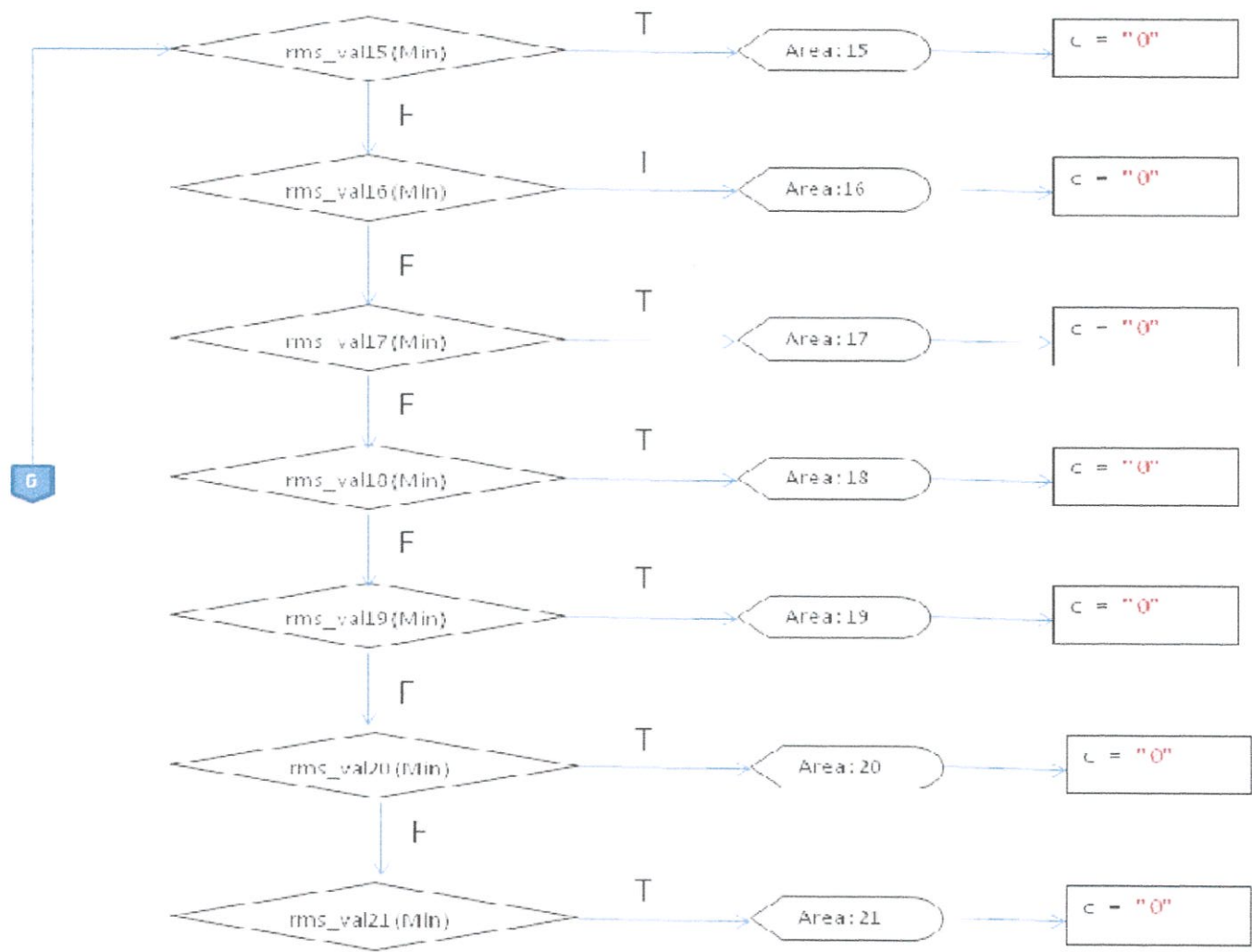


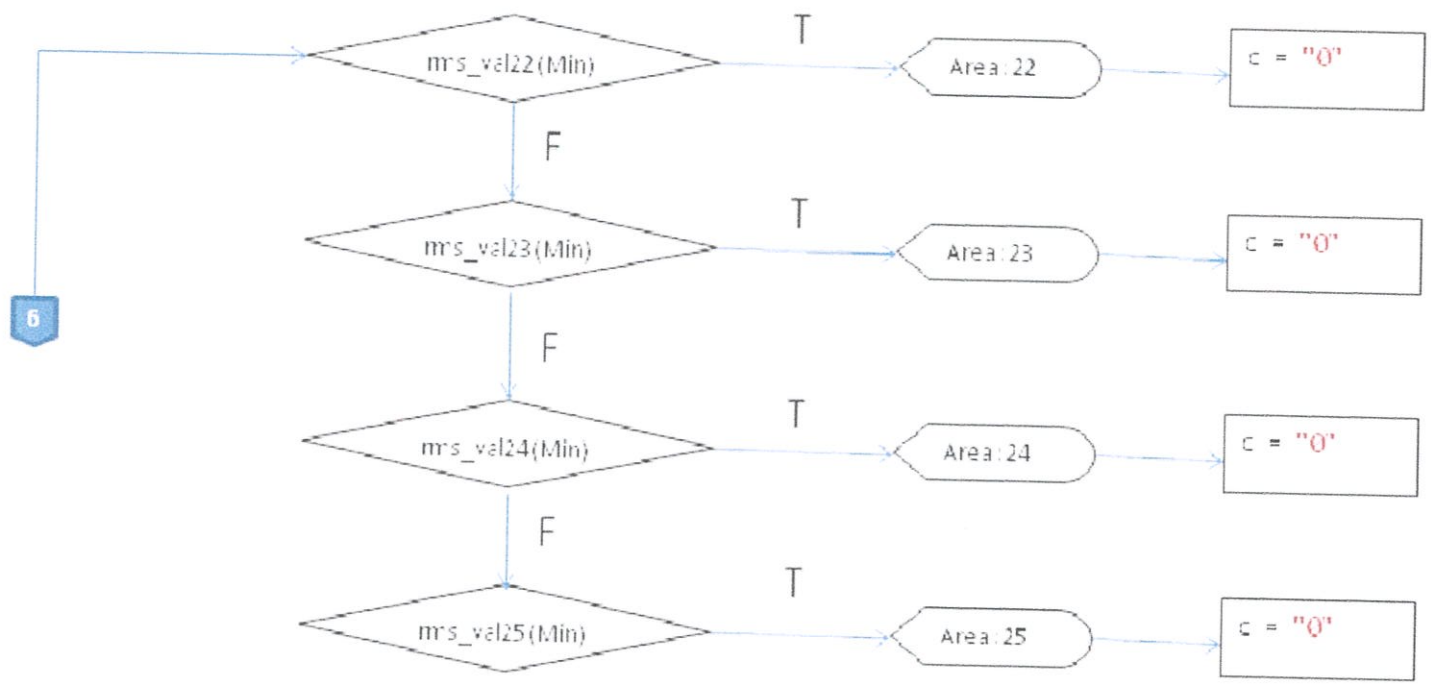












1. XBee®/XBee-PRO® RF Modules

The XBee and XBee-PRO RF Modules were engineered to meet IEEE 802.15.4 standards and support the unique needs of low-cost, low-power wireless sensor networks. The modules require minimal power and provide reliable delivery of data between devices.

The modules operate within the ISM 2.4 GHz frequency band and are pin-for-pin compatible with each other.



Key Features

Long Range Data Integrity

XBee

- Indoor/Urban: up to 100' (30 m)
- Outdoor line-of-sight: up to 300' (90 m)
- Transmit Power: 1 mW (0 dBm)
- Receiver Sensitivity: -92 dBm

XBee-PRO

- Indoor/Urban: up to 300' (90 m), 200' (60 m) for International variant
- Outdoor line-of-sight: up to 1 mile (1600 m), 2500' (750 m) for International variant
- Transmit Power: 63mW (18dBm), 10mW (10dBm) for International variant
- Receiver Sensitivity: -100 dBm

RF Data Rate: 250,000 bps

Advanced Networking & Security

Retries and Acknowledgements
DSSS (Direct Sequence Spread Spectrum)
Each direct sequence channels has over 65,000 unique network addresses available
Source/Destination Addressing
Unicast & Broadcast Communications
Point-to-point, point-to-multipoint and peer-to-peer topologies supported

Low Power

XBee

- TX Peak Current: 45 mA (@3.3 V)
- RX Current: 50 mA (@3.3 V)
- Power-down Current: < 10 μ A

XBee-PRO

- TX Peak Current: 250mA (150mA for international variant)
- TX Peak Current (RPSMA module only): 340mA (180mA for international variant)
- RX Current: 55 mA (@3.3 V)
- Power-down Current: < 10 μ A

ADC and I/O line support

Analog-to-digital conversion, Digital I/O
I/O Line Passing

Easy-to-Use

No configuration necessary for out-of box RF communications
Free X-CTU Software (Testing and configuration software)
AT and API Command Modes for configuring module parameters
Extensive command set
Small form factor

Worldwide Acceptance

FCC Approval (USA) Refer to Appendix A [p64] for FCC Requirements. Systems that contain XBee®/XBee-PRO® RF Modules inherit Digi Certifications.

ISM (Industrial, Scientific & Medical) **2.4 GHz frequency band**

Manufactured under **ISO 9001:2000** registered standards

XBee®/XBee-PRO® RF Modules are optimized for use in the United States, Canada, Australia, Japan, and Europe. Contact Digi for complete list of government agency approvals.



Specifications

Table 1-01. Specifications of the XBee®/XBee-PRO® RF Modules

Specification	XBee	XBee-PRO
Performance		
Indoor/Urban Range	Up to 100 ft (30 m)	Up to 300 ft. (90 m), up to 200 ft (60 m) International variant
Outdoor RF line-of-sight Range	Up to 300 ft (90 m)	Up to 1 mile (1600 m), up to 2500 ft (750 m) international variant
Transmit Power Output (software selectable)	1mW (0 dBm)	63mW (18dBm)* 10mW (10 dBm) for International variant
RF Data Rate	250,000 bps	250,000 bps
Serial Interface Data Rate (software selectable)	1200 bps - 250 kbps (non-standard baud rates also supported)	1200 bps - 250 kbps (non-standard baud rates also supported)
Receiver Sensitivity	-92 dBm (1% packet error rate)	-100 dBm (1% packet error rate)
Power Requirements		
Supply Voltage	2.8 – 3.4 V	2.8 – 3.4 V
Transmit Current (typical)	45mA (@ 3.3 V)	250mA (@3.3 V) (150mA for international variant) RPSMA module only: 340mA (@3.3 V) (180mA for international variant)
Idle / Receive Current (typical)	50mA (@ 3.3 V)	55mA (@ 3.3 V)
Power-down Current	< 10 µA	< 10 µA
General		
Operating Frequency	ISM 2.4 GHz	ISM 2.4 GHz
Dimensions	0.960" x 1.087" (2.438cm x 2.761cm)	0.960" x 1.297" (2.438cm x 3.294cm)
Operating Temperature	-40 to 85° C (industrial)	-40 to 85° C (industrial)
Antenna Options	Integrated Whip, Chip or U.FL Connector, RPSMA Connector	Integrated Whip, Chip or U.FL Connector, RPSMA Connector
Networking & Security		
Supported Network Topologies	Point-to-point, Point-to-multipoint & Peer-to-peer	
Number of Channels (software selectable)	16 Direct Sequence Channels	12 Direct Sequence Channels
Addressing Options	PAN ID, Channel and Addresses	PAN ID, Channel and Addresses
Agency Approvals		
United States (FCC Part 15.247)	OUR-XBEE	OUR-XBEEPRO
Industry Canada (IC)	4214A XBEE	4214A XBEEPRO
Europe (CE)	ETSI	ETSI (Max. 10 dBm transmit power output)*
Japan	R201WW07215214	R201WW08215111 (Max. 10 dBm transmit power output)*
Australia	C-Tick	C-Tick

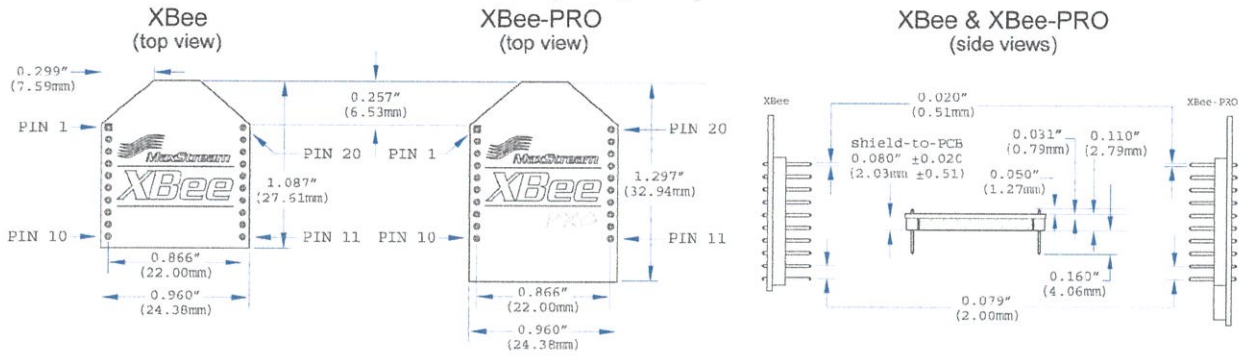
* See Appendix A for region-specific certification requirements.

Antenna Options: The ranges specified are typical when using the integrated Whip (1.5 dBi) and Dipole (2.1 dBi) antennas. The Chip antenna option provides advantages in its form factor; however, it typically yields shorter range than the Whip and Dipole antenna options when transmitting outdoors. For more information, refer to the "XBee Antennas" Knowledgebase Article located on Digi's Support Web site

Mechanical Drawings

Figure 1-01. Mechanical drawings of the XBee®/XBee-PRO® RF Modules (antenna options not shown)

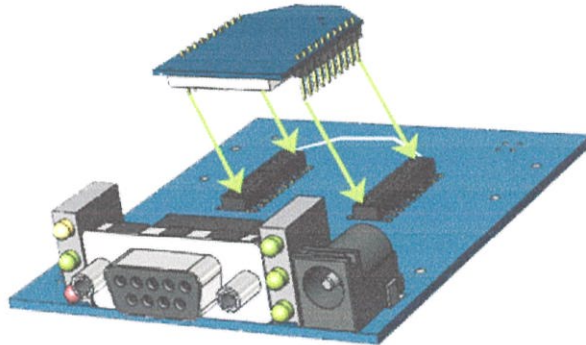
The XBee and XBee-PRO RF Modules are pin-for-pin compatible.



Mounting Considerations

The XBee®/XBee-PRO® RF Module was designed to mount into a receptacle (socket) and therefore does not require any soldering when mounting it to a board. The XBee Development Kits contain RS-232 and USB interface boards which use two 20-pin receptacles to receive modules.

Figure 1-02. XBee Module Mounting to an RS-232 Interface Board.



The receptacles used on Digi development boards are manufactured by Century Interconnect. Several other manufacturers provide comparable mounting solutions; however, Digi currently uses the following receptacles:

- Through-hole single-row receptacles - Samtec P/N: MMS-110-01-L-SV (or equivalent)
- Surface-mount double-row receptacles - Century Interconnect P/N: CPRMSL20-D-0-1 (or equivalent)
- Surface-mount single-row receptacles - Samtec P/N: SMM-110-02-SM-S

Digi also recommends printing an outline of the module on the board to indicate the orientation the module should be mounted.

Pin Signals

Figure 1-03. XBee®/XBee-PRO® RF Module Pin Numbers

(top sides shown - shields on bottom)



Table 1-02. Pin Assignments for the XBee and XBee-PRO Modules

(Low-asserted signals are distinguished with a horizontal line above signal name.)

Pin #	Name	Direction	Description
1	VCC	-	Power supply
2	DOUT	Output	UART Data Out
3	DIN / CONFIG	Input	UART Data In
4	DO8*	Output	Digital Output 8
5	RESET	Input	Module Reset (reset pulse must be at least 200 ns)
6	PWM0 / RSSI	Output	PWM Output 0 / RX Signal Strength Indicator
7	PWM1	Output	PWM Output 1
8	[reserved]	-	Do not connect
9	DTR / SLEEP_RQ / DI8	Input	Pin Sleep Control Line or Digital Input 8
10	GND	-	Ground
11	AD4 / DIO4	Either	Analog Input 4 or Digital I/O 4
12	CTS / DIO7	Either	Clear-to-Send Flow Control or Digital I/O 7
13	ON / SLEEP	Output	Module Status Indicator
14	VREF	Input	Voltage Reference for A/D Inputs
15	Associate / AD5 / DIO5	Either	Associated Indicator, Analog Input 5 or Digital I/O 5
16	RTS / AD6 / DIO6	Either	Request-to-Send Flow Control, Analog Input 6 or Digital I/O 6
17	AD3 / DIO3	Either	Analog Input 3 or Digital I/O 3
18	AD2 / DIO2	Either	Analog Input 2 or Digital I/O 2
19	AD1 / DIO1	Either	Analog Input 1 or Digital I/O 1
20	AD0 / DIO0	Either	Analog Input 0 or Digital I/O 0

* Function is not supported at the time of this release

Design Notes:

- Minimum connections: VCC, GND, DOUT & DIN
- Minimum connections for updating firmware: VCC, GND, DIN, DOUT, RTS & DTR
- Signal Direction is specified with respect to the module
- Module includes a 50k Ω pull-up resistor attached to RESET
- Several of the input pull-ups can be configured using the PR command
- Unused pins should be left disconnected

Electrical Characteristics

Table 1-03. DC Characteristics (VCC = 2.8 - 3.4 VDC)

Symbol	Characteristic	Condition	Min	Typical	Max	Unit
V _{IL}	Input Low Voltage	All Digital Inputs	-	-	0.35 * VCC	V
V _{IH}	Input High Voltage	All Digital Inputs	0.7 * VCC	-	-	V
V _{OL}	Output Low Voltage	I _{OL} = 2 mA, VCC >= 2.7 V	-	-	0.5	V
V _{OH}	Output High Voltage	I _{OH} = -2 mA, VCC >= 2.7 V	VCC - 0.5	-	-	V
I _{IN}	Input Leakage Current	V _{IN} = VCC or GND, all inputs, per pin	-	0.025	1	μA
I _{OZ}	High Impedance Leakage Current	V _{IN} = VCC or GND, all I/O High-Z, per pin	-	0.025	1	μA
TX	Transmit Current	VCC = 3.3 V	-	45 (XBee) 215, 140 (PRO, Int)	-	mA
RX	Receive Current	VCC = 3.3 V	-	50 (XBee) 55 (PRO)	-	mA
PWR-DWN	Power-down Current	SM parameter = 1	-	< 10	-	μA

Table 1-04. ADC Characteristics (Operating)

Symbol	Characteristic	Condition	Min	Typical	Max	Unit
V _{REFH}	VREF - Analog-to-Digital converter reference range		2.08	-	V _{DDAD} *	V
I _{REF}	VREF - Reference Supply Current	Enabled	-	200	-	μA
		Disabled or Sleep Mode	-	< 0.01	0.02	μA
V _{INDC}	Analog Input Voltage ¹		V _{SSAD} - 0.3	-	V _{DDAD} + 0.3	V

1. Maximum electrical operating range, not valid conversion range.

* V_{DDAD} is connected to VCC.

Table 1-05. ADC Timing/Performance Characteristics¹

Symbol	Characteristic	Condition	Min	Typical	Max	Unit
R _{AS}	Source Impedance at Input ²		-	-	10	kΩ
V _{AIN}	Analog Input Voltage ³		V _{REFL}		V _{REFH}	V
RES	Ideal Resolution (1 LSB) ⁴	2.08V ≤ V _{DDAD} ≤ 3.6V	2.031	-	3.516	mV
DNL	Differential Non-linearity ⁵		-	±0.5	±1.0	LSB
INL	Integral Non-linearity ⁶		-	±0.5	±1.0	LSB
E _{ZS}	Zero-scale Error ⁷		-	±0.4	±1.0	LSB
F _{FS}	Full-scale Error ⁸		-	±0.4	±1.0	LSB
E _{IL}	Input Leakage Error ⁹		-	±0.05	±5.0	LSB
E _{TU}	Total Unadjusted Error ¹⁰		-	±1.1	±2.5	LSB

1. All ACCURACY numbers are based on processor and system being in WAIT state (very little activity and no IO switching) and that adequate low-pass filtering is present on analog input pins (filter with 0.01 μF to 0.1 μF capacitor between analog input and VREFL). Failure to observe these guidelines may result in system or microcontroller noise causing accuracy errors which will vary based on board layout and the type and magnitude of the activity.

Data transmission and reception during data conversion may cause some degradation of these specifications, depending on the number and timing of packets. It is advisable to test the ADCs in your installation if best accuracy is required.

2. R_{AS} is the real portion of the impedance of the network driving the analog input pin. Values greater than this amount may not fully charge the input circuitry of the ATD resulting in accuracy error.

3. Analog input must be between V_{REFL} and V_{REFH} for valid conversion. Values greater than V_{REFH} will convert to \$3FF.

4. The resolution is the ideal step size or 1LSB = (V_{REFH} - V_{REFL})/1024

5. Differential non-linearity is the difference between the current code width and the ideal code width (1LSB). The current code width is the difference in the transition voltages to and from the current code.

6. Integral non-linearity is the difference between the transition voltage to the current code and the adjusted ideal transition voltage for the current code. The adjusted ideal transition voltage is (Current Code - 1/2) * (1 / ((V_{REFH} + E_{FS}) - (V_{REFL} + E_{ZS}))).

7. Zero-scale error is the difference between the transition to the first valid code and the ideal transition to that code. The Ideal transition voltage to a given code is (Code - 1/2) * (1 / (V_{REFH} - V_{REFL})).

8. Full-scale error is the difference between the transition to the last valid code and the ideal transition to that code. The ideal transition voltage to a given code is (Code - 1/2) * (1 / (V_{REFH} - V_{REFL})).

9. Input leakage error is error due to input leakage across the real portion of the impedance of the network driving the analog pin. Reducing the impedance of the network reduces this error.

10. Total unadjusted error is the difference between the transition voltage to the current code and the ideal straight-line transfer function. This measure of error includes inherent quantization error (1/2LSB) and circuit error (differential, integral, zero-scale, and full-scale) error. The specified value of E_{TJ} assumes zero E_{IL} (no leakage or zero real source impedance).

2. RF Module Operation

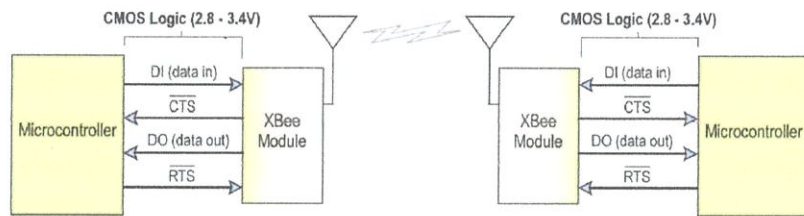
Serial Communications

The XBee®/XBee-PRO® RF Modules interface to a host device through a logic-level asynchronous serial port. Through its serial port, the module can communicate with any logic and voltage compatible UART; or through a level translator to any serial device (For example: Through a Digi proprietary RS-232 or USB interface board).

UART Data Flow

Devices that have a UART interface can connect directly to the pins of the RF module as shown in the figure below.

Figure 2-01. System Data Flow Diagram in a UART-interfaced environment
(Low-asserted signals distinguished with horizontal line over signal name.)

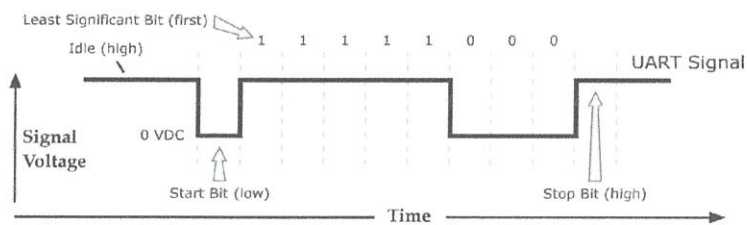


Serial Data

Data enters the module UART through the DI pin (pin 3) as an asynchronous serial signal. The signal should idle high when no data is being transmitted.

Each data byte consists of a start bit (low), 8 data bits (least significant bit first) and a stop bit (high). The following figure illustrates the serial bit pattern of data passing through the module.

Figure 2-02. UART data packet 0x1F (decimal number "31") as transmitted through the RF module
Example Data Format is 8-N-1 (bits - parity - # of stop bits)



Serial communications depend on the two UARTs (the microcontroller's and the RF module's) to be configured with compatible settings (baud rate, parity, start bits, stop bits, data bits).

The UART baud rate and parity settings on the XBee module can be configured with the BD and SB commands, respectively. See the command table in Chapter 3 for details.

Transparent Operation

By default, XBee®/XBee-PRO® RF Modules operate in Transparent Mode. When operating in this mode, the modules act as a serial line replacement - all UART data received through the DI pin is queued up for RF transmission. When RF data is received, the data is sent out the DO pin.

Serial-to-RF Packetization

Data is buffered in the DI buffer until one of the following causes the data to be packetized and transmitted:

1. No serial characters are received for the amount of time determined by the RO (Packetization Timeout) parameter. If RO = 0, packetization begins when a character is received.
2. The maximum number of characters that will fit in an RF packet (100) is received.
3. The Command Mode Sequence (GT + CC + GT) is received. Any character buffered in the DI buffer before the sequence is transmitted.

If the module cannot immediately transmit (for instance, if it is already receiving RF data), the serial data is stored in the DI Buffer. The data is packetized and sent at any RO timeout or when 100 bytes (maximum packet size) are received.

If the DI buffer becomes full, hardware or software flow control must be implemented in order to prevent overflow (loss of data between the host and module).

API Operation

API (Application Programming Interface) Operation is an alternative to the default Transparent Operation. The frame-based API extends the level to which a host application can interact with the networking capabilities of the module.

When in API mode, all data entering and leaving the module is contained in frames that define operations or events within the module.

Transmit Data Frames (received through the DI pin (pin 3)) include:

- RF Transmit Data Frame
- Command Frame (equivalent to AT commands)

Receive Data Frames (sent out the DO pin (pin 2)) include:

- RF-received data frame
- Command response
- Event notifications such as reset, associate, disassociate, etc.

The API provides alternative means of configuring modules and routing data at the host application layer. A host application can send data frames to the module that contain address and payload information instead of using command mode to modify addresses. The module will send data frames to the application containing status packets; as well as source, RSSI and payload information from received data packets.

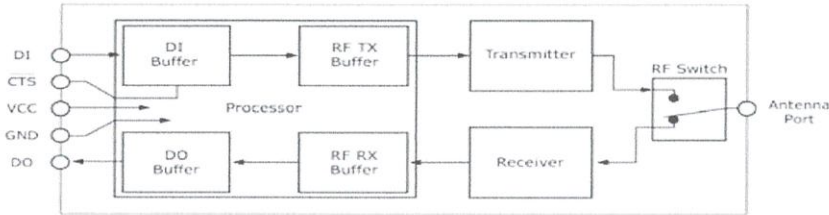
The API operation option facilitates many operations such as the examples cited below:

- > Transmitting data to multiple destinations without entering Command Mode
- > Receive success/failure status of each transmitted RF packet
- > Identify the source address of each received packet

To implement API operations, refer to API sections [p57].

Flow Control

Figure 2-03. Internal Data Flow Diagram



DI (Data In) Buffer

When serial data enters the RF module through the DI pin (pin 3), the data is stored in the DI Buffer until it can be processed.

Hardware Flow Control ($\overline{\text{CTS}}$). When the DI buffer is 17 bytes away from being full; by default, the module de-asserts $\overline{\text{CTS}}$ (high) to signal to the host device to stop sending data [refer to D7 (DIO7 Configuration) parameter]. $\overline{\text{CTS}}$ is re-asserted after the DI Buffer has 34 bytes of memory available.

How to eliminate the need for flow control:

1. Send messages that are smaller than the DI buffer size (202 bytes).
2. Interface at a lower baud rate [BD (Interface Data Rate) parameter] than the throughput data rate.

Case in which the DI Buffer may become full and possibly overflow:

If the module is receiving a continuous stream of RF data, any serial data that arrives on the DI pin is placed in the DI Buffer. The data in the DI buffer will be transmitted over-the-air when the module is no longer receiving RF data in the network.

Refer to the RO (Packetization Timeout), BD (Interface Data Rate) and D7 (DIO7 Configuration) command descriptions for more information.

DO (Data Out) Buffer

When RF data is received, the data enters the DO buffer and is sent out the serial port to a host device. Once the DO Buffer reaches capacity, any additional incoming RF data is lost.

Hardware Flow Control ($\overline{\text{RTS}}$). If $\overline{\text{RTS}}$ is enabled for flow control (D6 (DIO6 Configuration) Parameter = 1), data will not be sent out the DO Buffer as long as $\overline{\text{RTS}}$ (pin 16) is de-asserted.

Two cases in which the DO Buffer may become full and possibly overflow:

1. If the RF data rate is set higher than the interface data rate of the module, the module will receive data from the transmitting module faster than it can send the data to the host.
2. If the host does not allow the module to transmit data out from the DO buffer because of being held off by hardware or software flow control.

Refer to the D6 (DIO6 Configuration) command description for more information.

ADC and Digital I/O Line Support

The XBee®/XBee-PRO® RF Modules support ADC (Analog-to-digital conversion) and digital I/O line passing. The following pins support multiple functions:

Table 2-01. Pin functions and their associated pin numbers and commands

AD = Analog-to-Digital Converter, DIO = Digital Input/Output
Pin functions not applicable to this section are denoted within (parenthesis).

Pin Function	Pin#	AT Command
AD0 / DIO0	20	D0
AD1 / DIO1	19	D1
AD2 / DIO2	18	D2
AD3 / DIO3 / (COORD_SEL)	17	D3
AD4 / DIO4	11	D4
AD5 / DIO5 / (ASSOCIATE)	15	D5
DIO6 / (RTS)	16	D6
DIO7 / (CTS)	12	D7
D18 / (DTR) / (Sleep_RQ)	9	D8

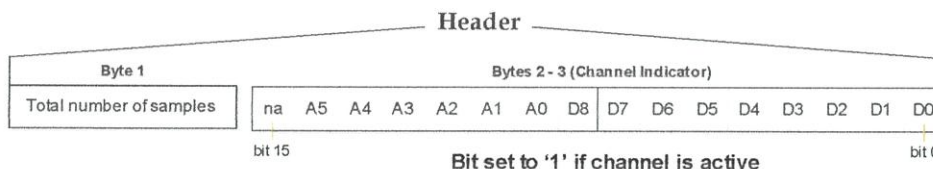
To enable ADC and DIO pin functions:

For ADC Support:	Set ATDn = 2
For Digital Input support:	Set ATDn = 3
For Digital Output Low support:	Set ATDn = 4
For Digital Output High support:	Set ATDn = 5

I/O Data Format

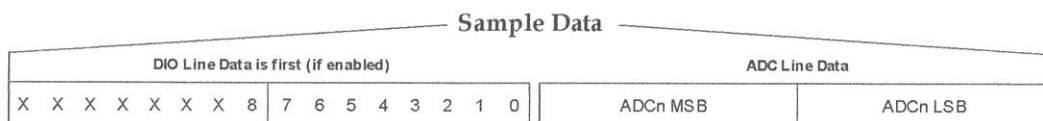
I/O data begins with a header. The first byte of the header defines the number of samples forthcoming. The last 2 bytes of the header (Channel Indicator) define which inputs are active. Each bit represents either a DIO line or ADC channel.

Figure 2-04. Header



Sample data follows the header and the channel indicator frame is used to determine how to read the sample data. If any of the DIO lines are enabled, the first 2 bytes are the DIO sample. The ADC data follows. ADC channel data is represented as an unsigned 10-bit value right-justified on a 16-bit boundary.

Figure 2-05. Sample Data



API Support

I/O data is sent out the UART using an API frame. All other data can be sent and received using Transparent Operation [refer to p11] or API framing if API mode is enabled (AP > 0).

API Operations support two RX (Receive) frame identifiers for I/O data (set 16-bit address to 0xFFFE and the module will do 64-bit addressing):

- 0x82 for RX (Receive) Packet: 64-bit address I/O
- 0x83 for RX (Receive) Packet: 16-bit address I/O

The API command header is the same as shown in the "RX (Receive) Packet: 64-bit Address" and "RX (Receive) Packet: 16-bit Address" API types [refer to p63]. RX data follows the format described in the I/O Data Format section [p13].

Applicable Commands: AP (API Enable)

Sleep Support

Automatic wakeup sampling can be suppressed by setting SO bit 1. When an RF module wakes, it will always do a sample based on any active ADC or DIO lines. This allows sampling based on the sleep cycle whether it be Cyclic Sleep (SM parameter = 4 or 5) or Pin Sleep (SM = 1 or 2). To gather more samples when awake, set the IR (Sample Rate) parameter.

For Cyclic Sleep modes: If the IR parameter is set, the module will stay awake until the IT (Samples before TX) parameter is met. The module will stay awake for ST (Time before Sleep) time.

Applicable Commands: IR (Sample Rate), IT (Samples before TX), SM (Sleep Mode), IC (DIO Change Detect), SO (Sleep Options)

DIO Pin Change Detect

When "DIO Change Detect" is enabled (using the IC command), DIO lines 0-7 are monitored. When a change is detected on a DIO line, the following will occur:

1. An RF packet is sent with the updated DIO pin levels. This packet will not contain any ADC samples.
2. Any queued samples are transmitted before the change detect data. This may result in receiving a packet with less than IT (Samples before TX) samples.

Note: Change detect will not affect Pin Sleep wake-up. The D8 pin (DTR/Sleep_RQ/DI8) is the only line that will wake a module from Pin Sleep. If not all samples are collected, the module will still enter Sleep Mode after a change detect packet is sent.

Applicable Commands: IC (DIO Change Detect), IT (Samples before TX)

NOTE: Change detect is only supported when the Dx (DIOx Configuration) parameter equals 3,4 or 5.

Sample Rate (Interval)

The Sample Rate (Interval) feature allows enabled ADC and DIO pins to be read periodically on modules that are not configured to operate in Sleep Mode. When one of the Sleep Modes is enabled and the IR (Sample Rate) parameter is set, the module will stay awake until IT (Samples before TX) samples have been collected.

Once a particular pin is enabled, the appropriate sample rate must be chosen. The maximum sample rate that can be achieved while using one A/D line is 1 sample/ms or 1 KHz (Note that the modem will not be able to keep up with transmission when IR & IT are equal to "1" and that configuring the modem to sample at rates greater than once every 20ms is not recommended).

Applicable Commands: IR (Sample Rate), IT (Samples before TX), SM (Sleep Mode)

I/O Line Passing

Virtual wires can be set up between XBee®/XBee-PRO® Modules. When an RF data packet is received that contains I/O data, the receiving module can be setup to update any enabled outputs (PWM and DIO) based on the data it receives.

Note that I/O lines are mapped in pairs. For example: AD0 can only update PWM0 and DI5 can only update DO5. The default setup is for outputs not to be updated, which results in the I/O data being sent out the UART (refer to the IU (Enable I/O Output) command). To enable the outputs to be updated, the IA (I/O Input Address) parameter must be setup with the address of the module that has the appropriate inputs enabled. This effectively binds the outputs to a particular module's input. This does not affect the ability of the module to receive I/O line data from other modules - only its ability to update enabled outputs. The IA parameter can also be setup to accept I/O data for output changes from any module by setting the IA parameter to 0xFFFF.

When outputs are changed from their non-active state, the module can be setup to return the output level to its non-active state. The timers are set using the Tn (Dn Output Timer) and PT (PWM Output Timeout) commands. The timers are reset every time a valid I/O packet (passed IA check) is received. The IC (Change Detect) and IR (Sample Rate) parameters can be setup to keep the output set to their active output if the system needs more time than the timers can handle.

Note: DI8 cannot be used for I/O line passing.

Applicable Commands: IA (I/O Input Address), Tn (Dn Output Timeout), P0 (PWM0 Configuration), P1 (PWM1 Configuration), M0 (PWM0 Output Level), M1 (PWM1 Output Level), PT (PWM Output Timeout), RP (RSSSI PWM Timer)

Configuration Example

As an example for a simple A/D link, a pair of RF modules could be set as follows:

Remote Configuration	Base Configuration
DL = 0x1234	DL = 0x5678
MY = 0x5678	MY = 0x1234
D0 = 2	P0 = 2
D1 = 2	P1 = 2
IR = 0x14	IU = 1
IT = 5	IA = 0x5678 (or 0xFFFF)

These settings configure the remote module to sample AD0 and AD1 once each every 20 ms. It then buffers 5 samples each before sending them back to the base module. The base should then receive a 32-Byte transmission (20 Bytes data and 12 Bytes framing) every 100 ms.