สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

ระบบระบุตำแหน่งวัตถุโดยใช้ ZIGBEE

(ZIGBEE-BASED LOCALIZATION SYSTEM)

T117480

โดย

นางสาวรฉัตร ระเบียบ

นางสาวริสา เถรว่อง

นายวีระพล คนคง

เลขหมู่.....
เลขทะเบียน 117480
วัน,เดือน,ปี - 5 ส.ค. 2554

.b.....
.i.....

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมโทรคมนาคม

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2553

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระบบระบุตำแหน่งวัตถุโดยใช้ ZIGBEE

(ZIGBEE-BASED LOCALIZATION SYSTEM)

โดย

นางสาวรฉัตร	ระเบียบ	50011261
นางสาวริสา	เถรว่อง	50011306
นายวีระพล	คนคง	50011510

อาจารย์ที่ปรึกษา

รศ.ดร. ยุทธพงษ์ รังสรรค์เสรี

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมโทรคมนาคม

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2553

ผ่านการตรวจรูปเล่มแล้ว

(ลงชื่อ).....ผู้ตรวจ

ผ่านการตรวจชิ้นงานแล้ว

(ลงชื่อ).....ผู้ตรว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์ปีการศึกษา 2553

สาขาวิชาวิศวกรรมโทรคมนาคม

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง ระบบระบุตำแหน่งวัตถุโดยใช้ ZIGBEE

ZIGBEE-BASED LOCALIZATION SYSTEM

ผู้จัดทำ

1. นางสาวรฉัตร ระเบียบ 50011261
2. นางสาวริสา เถรว่อง 50011306
3. นายวีระพล คนคง 50011510

..... อาจารย์ที่ปรึกษา

(รศ.ดร.ยุทธพงษ์ รังสรรค์เสรี)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

ผู้จัดทำขอขอบพระคุณอาจารย์ที่ปรึกษาปริญญานิพนธ์ รศ.ดร.ยุทธพงษ์ รังสรรค์เสรีและ ผศ.พิชญ สุพรรณกูล สำหรับแนวทางดำเนินการ คำแนะนำเพิ่มเติมต่างๆ และขอขอบคุณเพื่อนๆ ภาควิชาวิศวกรรมโทรคมสำหรับคำแนะนำ และความช่วยเหลือ และขอขอบคุณผลงานปริญญานิพนธ์ของรุ่นพี่ในสถาบันและสถาบันอื่นที่เป็นแนวทางในการทำงานได้สำเร็จ และขอขอบคุณสถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบังที่ให้ทุนสนับสนุนเพื่อทำปริญญานิพนธ์สำหรับนักศึกษามา ณ ที่นี้ด้วย

นางสาวรฉัตร ระเบียบ
นางสาวริสา เถรว่อง
นายวีระพล คนคง

ผู้จัดทำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระบบระบุตำแหน่งวัตถุโดยใช้ ZIGBEE

ZIGBEE-BASED LOCALIZATION SYSTEM

โดย	นางสาว รณัตร	ระเบียบ	50011261
	นางสาว ริสา	เถรว่อง	50011306
	นาย วีระพล	คนคง	50011510

อาจารย์ที่ปรึกษา รศ.ดร. ยุทธพงษ์ รังสรรค์เสรี

บทคัดย่อ

ปริญญานิพนธ์นี้นำเสนอระบบระบุตำแหน่งของวัตถุโดยใช้อุปกรณ์สื่อสารไร้สาย ZigBee ติดตั้งไว้ที่จุดอ้างอิง 3 จุด แล้วนำระยะทางที่ได้มาคำนวณหาตำแหน่งวัตถุ โดยอาศัยหลักการการแปลงความแรงสัญญาณ RSSI (received signal strength indication) และแสดงผลที่คอมพิวเตอร์

ABSTRACT

This thesis presents localization system. The ZigBee wireless devices are installed at three reference nodes. The system automatically estimates the distance between nodes by measuring the RSSI (received signal strength indicator) and displays on the computer.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

		หน้า
กิตติกรรมประกาศ		I
บทคัดย่อ		II
สารบัญ		III
สารบัญรูป		VII
สารบัญตาราง		X
บทที่ 1	**บทนำ**	**1**
	1.1 ความเป็นมาและความสำคัญของปัญหา	1
	1.2 วัตถุประสงค์	2
	1.3 ขอบเขตของปริญญานิพนธ์	2
บทที่ 2	**ทฤษฎีและหลักการที่เกี่ยวข้อง**	**3**
	2.1 ระบบระบุตำแหน่งภายในอาคาร	3
	2.1.1 ประเภทของระบบระบุตำแหน่งภายในอาคาร	3
	2.1.1.1 อินฟราเรด	3
	2.1.1.2 ความถี่วิทยุ	3
	2.1.1.3 อัลตร้าซาวด์	4
	2.1.2 เทคนิคการวัดที่นำมาใช้ในระบบหาตำแหน่ง	4
	2.2 Received Signal Strength Indication (RSSI)	6
	2.2.1 Channel Model	8
	2.2.2 Linear Regression Model	9
	2.3 คณิตศาสตร์พื้นฐานสำหรับการแก้ปัญหา lateration	12
	2.4 หลักการทำงานของการระบุตำแหน่งวัตถุในระบบเครือข่ายไร้สาย	14
	2.5 สิ่งที่ทำให้เกิดข้อผิดพลาดของสัญญาณ	15
	2.5.1 Multipath Fading and shadowing	15
	2.5.2 Non line-of-Sight (NLOS)	15
	2.5.3 Multiple-Access Interference	15

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

	หน้า
2.5.4 Fluctuations in Signal Propagation Speeds	16
2.6 ZigBee	16
2.6.1 ชนิดอุปกรณ์ของ ZigBee	17
2.6.2 การประยุกต์ใช้งาน ZigBee	18
2.6.3 โครงสร้างของโพรโทคอล ZigBee	18
2.6.3.1 Application layer	19
2.6.3.2 Network layer	20
2.6.4 ขั้นตอนการทำงานของโพรโทคอล ZigBee	21
2.6.5 การส่งข้อมูลของ ZigBee	22
2.6.6 การติดต่อสื่อสารของ ZigBee ผ่านทางพอร์ตอนุกรม	22
2.6.7 ลักษณะทางกายภาพของโมดูล XBee	23
2.7 ไมโครคอนโทรลเลอร์ (Microcontroller)	23
2.7.1 PIC (Peripheral Interface Controller)	25
2.7.2 รูปแบบการเขียนโปรแกรม Microcontroller	27
2.7.3 ชนิดของ PIC	29
2.7.4 PIC เบอร์ต่าง	30
2.8 การส่งข้อมูลในรูปแบบของ API	33
2.8.1 รูปแบบของ API Frame	33
2.8.2 ประเภทของ API	34
2.8.2.1 AT Command	35
2.8.2.2 AT Command Queue Parameter Value	35
2.8.2.3 AT Command Response	36
2.8.2.4 Tx (Transmit) Request : 64-bit address	36
2.8.2.5 Tx (Transmit) Request : 16-bit address	37
2.8.2.6 Tx (Transmit) Status	38
2.8.2.7 RX (Receive) Packet 64-bit address	38

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

		หน้า
	2.8.2.8 RX (Receive) Packet 16-bit address	39
	2.8.3 ตัวอย่างการใช้ API frame packet	40
	2.8.3.1 การส่งและรับคำสั่ง ATND จาก ZigBee	40
	2.8.3.2 การส่งข้อมูลค่าความแรงไปยัง ZigBee ที่Monitor Node	41
บทที่ 3	**การออกแบบและการจัดทำปริญญานิพนธ์**	**42**
	3.1 การออกแบบ	42
	3.1.1 โครงสร้างของระบบ	42
	3.1.2 การทำงานของระบบ	42
	3.1.2.1 วงจร Reference Node	42
	3.1.2.2 วงจร Object Node	43
	3.1.2.3 วงจร Monitor Node	43
	3.1.3 การทำงานของไมโครคอนโทรลเลอร์	48
	3.1.3.1 ขั้นตอนการทำงานที่ Object Node	48
	3.1.3.2 ขั้นตอนการทำงานที่ Monitor Node	50
	3.1.4 โปรแกรมแสดงผล	50
	3.1.4.1 ขั้นตอนการทำงานของโปรแกรมแสดงผล	50
	3.1.4.2 การใช้หน้าต่างแสดงพิกัดและตำแหน่ง	52
	3.2 เครื่องมือที่ใช้ในการทดลอง	53
	3.3 การจัดเก็บผลการทดลอง	54
บทที่ 4	**ผลการทดลอง**	**55**
	4.1 การตั้งค่าพารามิเตอร์สำหรับการทดลอง	55
	4.1.1 การตั้งค่าพารามิเตอร์ในโปรแกรม X-CTU	55
	4.2 ผลการทดลองวัดค่าแรงดันจากออสซิลโลสโคป	60
	4.3 ผลการทดลองการ Calibrate ระยะทางระหว่าง Object Node และ Monitor Node	63

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

		หน้า
	4.4 ผลการทดลองการระบุพิกัดของ Object Node ผ่านโปรแกรมแสดงผล	58
บทที่ 5	**สรุปผลและข้อเสนอแนะ**	**73**
	5.1 สรุปผล	73
	5.2 ปัญหาที่พบ	73
	5.3 ข้อเสนอแนะ	74
บรรณานุกรม		**75**
ภาคผนวก ก	โค้ด PIC 18F2525 ที่ Object Node	76
ภาคผนวก ข	โค้ด PIC 18F2525 ที่ Monitor Node	80
ภาคผนวก ค	โค้ดโปรแกรมแสดงผลโดยใช้โปรแกรม Delphi	89

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป

หน้า

รูปที่

2.1 การจัดจำแนกประเภทของระบบระบุตำแหน่ง 6

2.2 กราฟค่ากำลังงานเฉลี่ยที่แต่ละระยะทาง 9

2.3 กราฟค่าความผิดพลาดของข้อมูลที่ i=1 10

2.4 การระบุตำแหน่งวัตถุด้วยเทคนิค trilateration 12

2.5 Function block diagram ของระบบค้นหาตำแหน่งในเครือข่ายไร้สาย 14

2.6 ปริมาณการใช้ Throughput ของมาตรฐานการสื่อสารไร้สายแบบต่างๆ 17

2.7 เครือข่าย Zigbee แบบ Star , Cluster , Mesh 19

2.8 IEEE 802.15.4 / ZigBee Stack 21

2.9 การต่อไมโครคอนโทรลเลอร์กับซิกบี 23

2.10 ลักษณะทางกายภาพของโมดูล XBee 25

2.11 โครงสร้างภายในไมโครคอนโทรลเลอร์ 26

2.12 รูปแบบการโปรแกรมไมโครคอนโทรลเลอร์ด้วย Assembly ไฟล์เดียว 27

2.13 รูปแบบการโปรแกรมไมโครคอนโทรลเลอร์ด้วย Assembly หลายไฟล์ 28

2.14 รูปแบบโปรแกรมไมโครคอนโทรลเลอร์ด้วยภาษาขั้นสูง 29

2.15 ตำแหน่งขา (PIN Diagram) ของ PIC 18F2525 33

2.16 รูปแบบของ API Packet frame Mode 1 34

2.17 รูปแบบของ API Packet frame 34

2.18 รูปแบบของ API Packet frame ประเภท AT Command 35

2.19 รูปแบบของ API Packet frame ประเภท AT Command Queue Parameter Value 36

2.20 รูปแบบของ API Packet frame ประเภท AT Command Response 36

2.21 รูปแบบของ API Packet frame ประเภท Tx (Transmit) Request : 64-bit address 37

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป (ต่อ)

หน้า

รูปที่

2.22 รูปแบบของ API Packet frame ประเภท Tx (Transmit) Request : 16-bit address 38

2.23 รูปแบบของ API Packet frame ประเภท Tx (Transmit) 38

2.24 รูปแบบของ API Packet frame ประเภท RX (Receive) Packet 64-bit address 39

2.25 รูปแบบของ API Packet frame ประเภท RX (Receive) Packet 16-bit address 39

3.1 วงจร Reference Node 44

3.2 วงจร Object Node 45

3.3 วงจร MAX 232 ติดต่อกับไมโครคอนโทรเลอร์ที่ Monitor Node 46

3.4 วงจร MAX 232 ติดต่อกับ Zigbee ที่ Monitor Node 47

3.5 ความสัมพันธ์ส่วนต่างๆ ของระบบ 48

3.6 ขั้นตอนการทำงานของฝั่ง Object Node 49

3.7 ขั้นตอนการทำงานของไมโครคอนโทรลเลอร์ฝั่ง Monitor Node 51

3.8 หน้าต่างแสดงพิกัดและตำแหน่ง 52

3.9 ขั้นตอนการออกแบบของโปรแกรมแสดงผล 53

4.1 PC Setting Tab 55

4.2 หน้าต่าง Com test/Query Modem 56

4.3 หน้าต่าง com test/ Query Modem ที่ไม่สามารถติดต่อกับ Xbee 56

4.4 Modem Configuration Tab 57

4.5 กด Write เพื่อเขียนค่าพารามิเตอร์ลงใน Xbee 59

4.6 การรับ – ส่งข้อมูลระหว่าง Xbee 59

4.7 Tab Range Test 60

4.8 วัดขา Tx Object Node ซึ่งส่งอักขระ “ + + + “ 61

4.9 วัดขา Tx Object Node ซึ่งส่งอักษร “ A ” 61

4.10 วัดขา Tx Object Node ซึ่งส่งอักษร “ T ” 62

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป (ต่อ)

หน้า

รูปที่

4.11 วัดขา Tx Object Node ซึ่งส่งอักษร “ N ” 62

4.12 วัดขา Tx Object Node ซึ่งส่งอักษร “ D ” 62

4.13 วัดขา Tx Object Node ซึ่งส่งอักษร “ 1 ” 63

4.14 วัดขา Rx Monitor Node ซึ่งรับอักษร “ 1 ” 63

4.15 ค่าความแรงของสัญญาณที่รับได้ผ่านโปรแกรม HyperTerminal 64

4.16 ความสัมพันธ์ระหว่างระยะทางกับค่าความแรงสัญญาณ 67

4.17 หน้าต่างแสดงพิกัดและตำแหน่ง 69

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

หน้า

ตารางที่

2.1 การเปรียบเทียบเทคโนโลยีไร้สายในแบบต่างๆ 16

2.2 ขาสัญญาณโมดูล XBee และ XBee Pro 24

2.3 แสดงคุณสมบัติของ PIC 18F2525 32

2.4 คำสั่งของพารามิเตอร์ API Mode 40

2.5 API frame packet ที่ได้รับทั้งแต่บิตที่ 0 – 10 40

2.6 API frame packet ที่ได้รับทั้งแต่บิตที่ 11 – 21 41

2.7 ACK ของคำสั่ง ATND จำนวน 9 Byte 41

2.8 API frame ที่ได้ทำการส่งไปให้ Monitor Node 41

4.1 การตั้งค่าพารามิเตอร์ ZigBee ผ่านทาง AT Command 58

4.2 ค่าความแรงของสัญญาณ RSSI (ในหน่วย dBm) ที่ระยะทางต่างๆ 65

4.3 ตำแหน่งของวัตถุ เทียบกับตำแหน่งที่คำนวณได้ และค่าความคลาดเคลื่อน 69

4.4 ค่าความผิดพลาดของการทดลอง และค่าเฉลี่ยทั้งหมดของความผิดพลาด 72

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1

บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

ระบบระบุตำแหน่งในปัจจุบันที่เป็นที่รู้จักคือ จีพีเอส (Global Positioning System) เทคโนโลยีนี้นำมาใช้เพื่อระบุตำแหน่งวัตถุต่างๆบนพื้นผิวโลก แต่จีพีเอสยังคงมีข้อจำกัดเมื่อนำมาใช้ในอาคาร เนื่องจากจีพีเอสใช้การคำนวณจากความถี่สัญญาณนาฬิกาที่ส่งมาจากดาวเทียม เมื่อใช้ในอาคารหรือที่ที่มีการกีดขวางสัญญาณจากดาวเทียมจะทำให้ไม่สามารถคำนวณตำแหน่งได้ ด้วยเหตุนี้ระบบระบุตำแหน่งวัตถุภายในอาคารจึงได้ถูกนำมาพิจารณาประกอบกับในปัจจุบันเทคโนโลยีการสื่อสารแบบไร้สาย ได้มีการพัฒนาอุปกรณ์ให้มีคุณภาพสูงขึ้นและมีราคาถูก สามารถประยุกต์ใช้งานด้านต่างๆในชีวิตประจำวันได้อย่างแพร่หลายการระบุตำแหน่งวัตถุภายในอาคารนั้นสามารถนำเครือข่ายเซ็นเซอร์ไร้สาย (Wireless Sensor Network) มาประยุกต์ใช้งานได้ เนื่องจากเซ็นเซอร์มีขนาดเล็กสามารถพกพาได้สะดวกใช้พลังงานต่ำ สามารถสร้างเครือข่ายได้ และการติดตั้งสามารถทำได้อย่างง่ายดาย ในการประยุกต์ใช้ในชีวิตประจำวันสามารถทำได้หลายรูปแบบเช่น การระบุตำแหน่งผู้ป่วยที่ต้องดูแลเป็นพิเศษในโรงพยาบาล, การระบุตำแหน่งสินค้าที่มีราคาสูงภายในโรงงาน, การระบุตำแหน่งเด็กเมื่อพลัดหลงในห้างสรรพสินค้า เป็นต้น

ZigBee เป็นอุปกรณ์สื่อสารแบบไร้สายระยะใกล้ Wireless Personal Area Networks (WPANs) โดยใช้เทคโนโลยีมาตรฐาน IEEE 802.15.4 ซึ่งแม้จะเป็นประเภทความเร็วในการส่งข้อมูลต่ำ แต่สามารถนำมาประยุกต์ใช้งานได้ในชีวิตประจำวันมากกว่าอุปกรณ์ไร้สายความเร็วสูง เนื่องจากมีขนาดเล็ก กินไฟน้อยและมีราคาถูก

ดังนั้น ผู้จัดทำจึงมีแนวคิดในการสร้างระบบระบุตำแหน่งวัตถุขึ้นจากเทคโนโลยี ZigBee โดยจะศึกษาถึงโมดูลที่ใช้และอัลกอริทึมต่างๆที่ใช้ในการระบุตำแหน่งวัตถุ รวมถึงปัจจัยอื่นๆที่มีผลต่อการทำงานของระบบ เพื่อนำมาปรับใช้ให้เหมาะสมกับสถานที่นั้นๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.2 วัตถุประสงค์

1) เพื่อสามารถใช้อุปกรณ์รับส่งสัญญาณ ZigBee และเชื่อมต่อกับ Microcontroller เพื่อประมวลผล
2) เพื่อสร้างอุปกรณ์ที่สามารถระบุตำแหน่งวัตถุในพื้นที่ที่ติดตั้งอุปกรณ์ได้
3) เพื่อนำไปประยุกต์ใช้งานควบคุมระบบฝังตัวไร้สายราคาถูก

1.3 ขอบเขตของปริญญานิพนธ์

1) ศึกษาวิธีการทำงานของอุปกรณ์ ZigBee
2) รับและส่งข้อมูล ZigBee โดยใช้ไมโครคอนโทรลเลอร์ PIC
3) รับข้อมูลจากคอมพอร์ต (Comport) บนคอมพิวเตอร์ และระบุพิกัดที่ถูกต้องบนโปรแกรมแสดงผล

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2

ทฤษฎีและหลักการที่เกี่ยวข้อง

2.1 ระบบระบุตำแหน่ง (Localization System)

เมื่อกล่าวถึงระบบการระบุตำแหน่งในปัจจุบันนี้นับว่ามีการนำมาใช้งานกันอย่างแพร่หลายและมีการนำเอาเทคโนโลยีใหม่ๆมาใช้ร่วมกับการระบุตำแหน่งมากมาย โดยตัวอย่างของการนำเอาระบบระบุตำแหน่งมาใช้ในปัจจุบันนี้ตัวอย่างเช่น การระบุตำแหน่งของผู้ป่วยภายในโรงพยาบาล หรือ การระบุตำแหน่งของนักดับเพลิงที่เข้าไปช่วยเหลือผู้ประสพภัย เป็นต้น

2.1.1 ประเภทของระบบระบุตำแหน่ง

ระบบระบุตำแหน่งสามารถแบ่งออกเป็น 3 ประเภทใหญ่ๆ ตามเทคโนโลยีที่ใช้ในการตรวจจับ คือ อินฟราเรด ความถี่วิทยุ และอัลตร้าซาวด์ โดยแต่ละประเภทจะมีรายละเอียดดังนี้

2.1.1.1 อินฟราเรด (Infrared)

สัญญาณอินฟราเรดมีคุณสมบัติเหมือนกับแสงทั่วไปคือไม่สามารถเดินทางผ่านกำแพงหรือสิ่งกีดขวางได้ ดังนั้นจึงค่อนข้างมีข้อจำกัดของระยะการใช้งานในสภาพแวดล้อม แต่อย่างไรก็ตามความเร็วในการแพร่สัญญาณสูง ประมาณ 3×10^{8} m/s ด้วยเหตุนี้จึงทำให้ระบบนี้ต้องการอุปกรณ์ที่มากกว่าระบบอัลตร้าซาวด์ และการรบกวนของแสงจะมีผลต่อความถูกต้องของการตรวจจับสัญญาณ โดยทั่วไปแล้วอินฟราเรดจะมีระยะการใช้งานอยู่ที่ประมาณ 5 เมตร และโดยปกติอุปกรณ์อินฟราเรดจะมีขนาดเล็กกว่าอุปกรณ์อัลตร้าซาวด์

2.1.1.2 ความถี่วิทยุ (Radio Frequency)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สัญญาณความถี่วิทยุ (RF) สามารถเดินทางผ่านวัสดุได้มาก ดังนั้นระบบนี้จึงมีระยะการใช้งานที่ดีภายในสภาพแวดล้อมของอาคารซึ่งความเร็วในการแพร่สัญญาณก็สูงเหมือนกันคือประมาณ 3×10^8 m/s และเป็นระบบที่ใช้ความถี่สาธารณะ โดยระบบนี้มีระยะการใช้งานที่กว้างกว่าระบบที่ใช้อินฟราเรดและอัลตร้าซาวด์

2.1.1.3 อัลตร้าซาวด์ (Ultrasound)

ถึงแม้ว่าอัลตร้าซาวด์จะทำงานที่ย่านความถี่ต่ำ (40 kHz) เมื่อเปรียบเทียบกับสองระบบก่อนหน้านี้ โดยระบบนี้มีความแม่นยำที่ดีสำหรับตำแหน่งที่ตรวจจับที่มีความเร็วของการแพร่สัญญาณต่ำ (343 m/s) ส่วนข้อดีของระบบนี้คือ ง่ายและราคาไม่แพง แต่อย่างไรก็ตามอัลตร้าซาวด์ ไม่สามารถเดินทางผ่านกำแพงได้แต่จะสะท้อนกับสิ่งกีดขวางมาก โดยระบบนี้มีระยะการใช้งานประมาณ 3 เมตรถึง 10 เมตร และมีความละเอียด 1 เซนติเมตรของการวัดระยะ ซึ่งอุณหภูมิการใช้งานมีผลต่อประสิทธิภาพของอัลตร้าซาวด์

2.1.2 เทคนิคการวัดที่นำมาใช้ในระบบหาตำแหน่ง

ปัจจุบันได้มีการกล่าวถึงเทคนิคการวัดที่นำมาใช้ในระบบหาตำแหน่งอยู่หลายเทคนิคโดยในงานวิจัยนี้จะขอกล่าวเพียงเทคนิคพื้นฐานที่นำมาใช้ในระบุตำแหน่งซึ่งประกอบด้วยการวัดระยะทาง (Distance), มุม (Angle) และความเป็นเอกลักษณ์ของตำแหน่ง (Location Pattern or Fingerprint)

วิธีการวัดระยะทาง โดยปกติเรียกว่า Lateration ส่วนวิธีการวัดมุม โดยปกติเรียกว่า Angulations ซึ่งทั้งสองวิธีนี้แยกมาจาก Triangulation ซึ่งใช้ประโยชน์จากรูปแบบสามเหลี่ยมในระบุตำแหน่ง โดยชื่อของวิธีการวัดที่รู้จักกันคือ Angle of Arrival (AOA), Time of Arrival (TOA) และ Fingerprinting ซึ่งสองวิธีการแรกจะเป็นวิธีการที่นิยมใช้ในระบบหาตำแหน่งภายนอกอาคารเนื่องจากต้องการพื้นที่โล่งที่ไม่มีสิ่งกีดขวาง (Line-of-Sight) และทั้งสองวิธีนี้มีปัญหาหรือต้องการการคำนวณที่ซับซ้อนในช่องสัญญาณความถี่ถ้ามี Noise การรบกวนและ Multipath

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

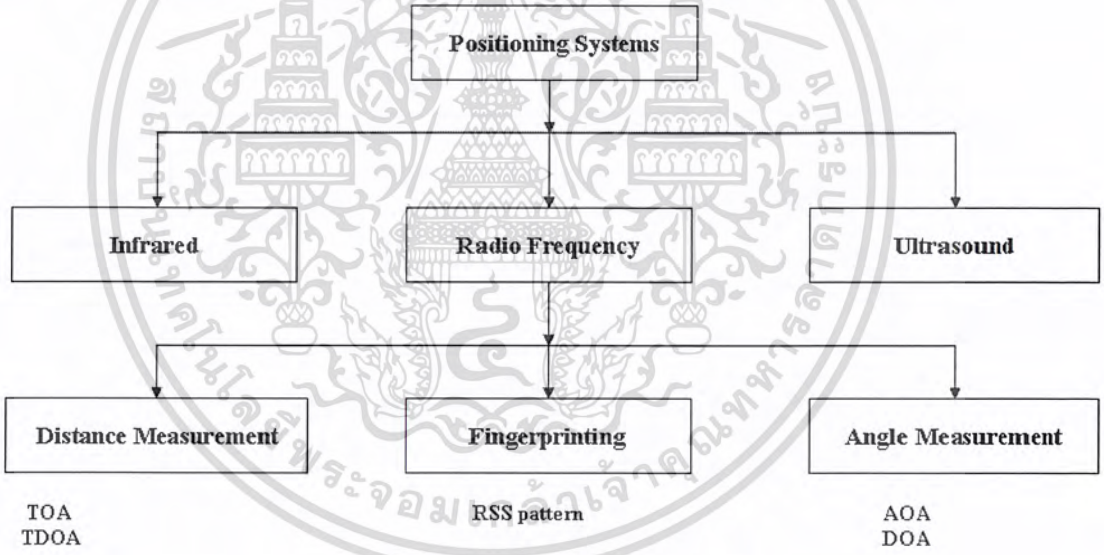
ในสภาพแวดล้อม Mobile Station (MS) หรือ ผู้ใช้งาน จะอยู่รอบๆ Base Station (BS) หรือ Access Point (AP) แบบกระจาย ทำให้เกิดมุมขึ้นมาหลายมุมในการรับสัญญาณหรือระยะห่างระหว่างเครื่องส่งและเครื่องรับสั้นกว่าเวลาที่จะสามารถวัดได้ด้วยระบบ ดังนั้น AOA และ TOA จึงเป็นวิธีการที่ไม่สามารถนำมาใช้ได้ดีในสภาพแวดล้อม ส่วนวิธี Fingerprinting Technique จะนิยมนำมาใช้งานในการระบุตำแหน่งมากกว่าสองวิธีแรก เนื่องจากง่ายและไม่ต้องติดตั้งอุปกรณ์พิเศษเพิ่มเติม ซึ่งรายละเอียดของแต่ละวิธีการวัดมีดังนี้

การวัดระยะทางบนพื้นฐานเวลาที่ดีเลย์ Time of Arrival (TOA) และ Time Difference of Arrival (TDOA) Techniques โดยความแม่นยำของเทคนิคนี้จะอาศัยการประสานเวลาให้ตรงกัน (Synchronized) ระหว่างเครื่องส่งสัญญาณและเครื่องรับสัญญาณ โดยใช้ความต่างของเวลา Time of Flight (TOF) ที่คำนวณระยะทางระหว่างเครื่องส่งสัญญาณและเครื่องรับสัญญาณ ดังนั้นความแม่นยำของการประสานเวลากันจึงเป็นเรื่องที่สำคัญมากในระบบนี้ โดยการวัดระยะทางจะต้องมีอย่างน้อย สามระยะทางจาก BS ไปยังจุดที่ต้องการรู้ตำแหน่งของผู้ใช้งาน (MS) ในกรณีระนาบ 2มิติ (หากเป็นระบบ 3 มิติ ต้องการ 4 ระยะทางจาก 4 BS) ซึ่งตำแหน่งของ MS หรือวัตถุจะอยู่ที่จุดตัด (Intersection) ของวงกลม 3 วงที่แต่ละวงมีรัศมีเท่ากับวัตถุไปยัง BS หรือสามารถคำนวณได้โดยใช้ความสัมพันธ์ของสามเหลี่ยมในการระบุตำแหน่งของ MS ซึ่งวิธีการทั้งสองนี้จะต้องการความแม่นยำของสัญญาณนาฬิกาสูงในการติดต่อกับระบบ ซึ่งในการใช้งานโดยมากจะนิยมใช้ TDOA ส่วนตัวอย่างของระบบที่ใช้เทคนิคนี้ในการระบุตำแหน่ง เช่น GPS, Active Bats และ Cricket

การวัดมุม Angle of Arrival (AOA) และ Direction of Arrival (DOA) Techniques เป็นการระบุตำแหน่งที่ใช้การวัดมุมตกกระทบของสัญญาณ โดยการระบุตำแหน่งคำนวณมาจากการตัดกันของมุมที่ MS ทำอยู่กับ BS ด้วยวิธีการนี้จึงจำเป็นที่จะต้องใช้ BS อย่างน้อย 2 ตัว ตำแหน่งของ MSที่ได้นั้น จะเป็นจุดตัดของเส้นที่ลากทำมุมตามที่วัดได้ของ BS ตัวหนึ่งกับเส้นที่ลากทำมุมของ BSตัวที่สอง ซึ่งวิธีการนี้จำเป็นที่จะต้องมีการติดตั้งอุปกรณ์เพิ่มคือ Directional Antenna และ Antenna Arrays เพื่อใช้ในการวัดมุมตกกระทบ ดังนั้นจึงเป็นเรื่องที่ยุ่งยากในการวัดมุมที่ MS

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ความเป็นเอกลักษณ์ของตำแหน่ง (Location Pattern or Fingerprinting) วิธีการนี้โดยทั่วไปต้องการการวัดความแรงของสัญญาณที่รับได้ (Received Signal Strength (RSS)) เพียงอย่างเดียว เท่านั้น โดยการระบุตำแหน่งของ MS ระบบจะต้องทำการวัด RSS ที่ตำแหน่งของ MS อยู่ขณะนั้น เพื่อนำค่า RSS ที่วัดได้ไปเปรียบเทียบความเป็นเอกลักษณ์ของแต่ละตำแหน่งที่ได้เก็บค่า RSS ไว้ก่อนหน้านี้ในฐานข้อมูล แล้วทำการแสดงตำแหน่งที่เปรียบเทียบแล้วใกล้เคียงที่สุด โดยวิธีการนี้ไม่จำเป็นต้องใช้ BS หรือ AP 3 ตัวในการระบุตำแหน่ง แต่จะต้องมี BS หรือ AP อย่างน้อย 1 ตัว ส่วนข้อด้อยของการใช้วิธีนี้ในการระบุตำแหน่งคือต้องใช้เวลามากในการเก็บสัญญาณ RSS ไว้ในฐานข้อมูล กรณีที่บริเวณที่ต้องการระบุตำแหน่งมีขนาดใหญ่ ดังนั้นวิธีการนี้จึงนิยมนำมาใช้ในระบบหาตำแหน่งมากกว่าระบบหาตำแหน่งภายนอกอาคาร ซึ่งการจำแนกประเภทของระบบระบุตำแหน่ง แสดงดังรูปที่ 2.1



รูปที่ 2.1 การจัดจำแนกประเภทของระบบระบุตำแหน่ง

2.2 Received Signal Strength Indication (RSSI)

RSSI เป็นค่าที่ใช้บอกความแรงของสัญญาณวิทยุที่ได้รับในเทอมของพลังงานมีหน่วยเป็น dBm (Decibels Mill watt) โดยค่า RSSI จะแปรผันตรงกับความแรงของสัญญาณ นั่น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คือ ถ้าค่า RSSI มีค่ามาก แสดงว่าสัญญาณที่ได้รับมีความแรงสูง นั่นคือ ตัวส่งและตัวรับอยู่ใกล้กัน และในทางกลับกันหากค่า RSSI มีค่าน้อยแสดงว่าสัญญาณที่ได้รับมีความแรงต่ำ ตัวส่งและตัวรับอยู่ไกลกัน

ข้อดีของการใช้การวัดแบบนี้คือ ไม่ต้องการฮาร์ดแวร์ที่มีราคาสูงและง่ายต่อการติดตั้ง เหมาะสำหรับการวัดในเมืองหรือที่ร่ม ใน Cellular และ WLAN Network มีอุปกรณ์สำหรับวัดค่า RSS อยู่แล้วโดยไม่ต้องเพิ่มอุปกรณ์ใดๆ สำหรับการวัดแบบ Direct Measurement ไม่มีความน่าเชื่อถือเพราะว่ามีStandard Deviation ของ Error มากซึ่งได้รับผลกระทบจาก Shadow Fading ค่อนข้างมาก การทำให้ RSSI น่าเชื่อถือมากขึ้นจะต้องสร้างระบบในการจดจำพื้นที่นั้นๆ โดยการวัดค่าและเก็บข้อมูลไว้หลายๆครั้ง เมื่อวิเคราะห์ภายหลังก็จะมีความผิดพลาดน้อยลง

วิธีนี้ไม่นิยมใช้ในงานที่ต้องการความแม่นยำสูงเนื่องจากค่าความแปรปรวนของสัญญาณจากสภาพแวดล้อมต่างๆกัน โดยเฉพาะอย่างยิ่งเมื่อสัญญาณมาจากหลายทิศทางและเกิดเงาสะท้อนของสัญญาณ (Shadowing Effect) อย่างไรก็ตามวิธีการวัดความแรงของสัญญาณ (Signal-Strength-Based) มักนำไปใช้ในการประมาณค่าตำแหน่งวัตถุหรือนำไปใช้ร่วมกับวิธี อื่นในการพิจารณา

นอกจากนี้เราสามารถเพิ่มความแม่นยำของการระบุตำแหน่งวัตถุได้มากขึ้น โดยใช้หลักการทางคณิตศาสตร์โดยโหนดที่ใช้วัดวัดระยะทางต้องมีอย่างน้อย 3 โหนดในระนาบ 2 มิติ (หากเป็น ระบบ 3 มิติต้องการ 4 ระยะทางจาก 4 จุดสังเกต) ตำแหน่งของวัตถุจะอยู่ที่จุดตัด (Intersection) ของวงกลม 3 วงที่แต่ละวงมีรัศมีเท่ากับวัตถุไปยังจุดสังเกต ซึ่งจะอธิบายในหัวข้อถัดไป

หนึ่งความเป็นไปได้ที่จะใช้ในการวัดระยะทางคือความแรงของสัญญาณที่ได้รับของสัญญาณวิทยุที่เข้ามา แนวคิดเบื้องหลัง RSSI คือกำลังที่เครื่องส่ง (P_t) มีผลกระทบโดยตรงต่อเครื่องรับ (P_r) ตามสมการ Friis (Friis's Transmission Equation)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$P_r = P_t G_t G_r \left(\frac{\lambda}{4\pi}\right)^2 \left(\frac{1}{d}\right)^n \quad (2.1)$$

เมื่อ P_t คือ กำลังส่ง (Transmitting Power)

P_r คือ กำลังรับ (Receiving Power)

G_t คือ อัตราขยายของสายอากาศส่ง (Transmitting Antenna Gain)

G_t คือ อัตราขยายของสายอากาศรับ (Receiving Antenna Gain)

λ คือ ความยาวคลื่น (Wavelength)

n คือ Path Loss Exponent

สำหรับสภาพแวดล้อมใดๆ

$$P_r(d) \alpha \frac{1}{d^n}$$

2.2.1 Channel Model

$$P_r(d) = \frac{K}{d^n} + x$$

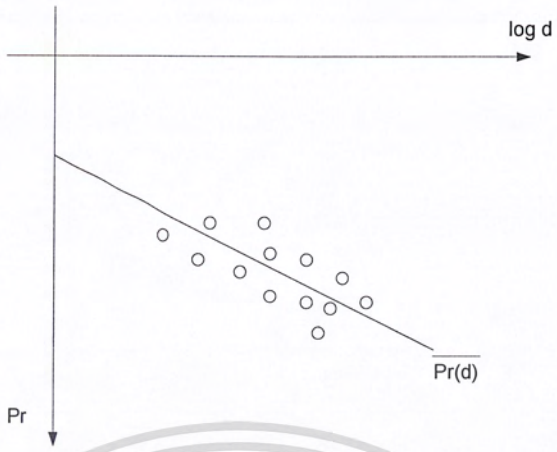
เมื่อ K คือ ค่าคงที่

x คือ ตัวแปรสุ่มของ Fading

$$P_r(d)[dB] = -10n \log d + K[dB] + x[dB]$$
$$\overline{P_r(d)}[dB] = -10n \log d + K[dB] \quad (2.2)$$

ค่า $\overline{P_r(d)}$ คือ ค่ากำลังงานที่รับได้เฉลี่ย ดังรูปที่ 2.2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.2 กราฟค่ากำลังงานเฉลี่ยที่แต่ละระยะทาง

2.2.2 Linear Regression Model

กำหนด d_0 เป็น Reference Distance

$$\overline{P_r(d_0)}[dB] = -10n \log d_0 + K[dB] \tag{2.3}$$

สมการ (2.2) - (2.3) จะได้

$$\overline{P_r(d)}[dB] = -10n \log\left(\frac{d}{d_0}\right) + \overline{P_r(d_0)}[dB]$$

กำหนดให้ $d_0 = 1$ เมตร

$$\overline{P_r(d)}[dB] = -10n \log d + \overline{P_r(1m)}[dB]$$

กำหนดให้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

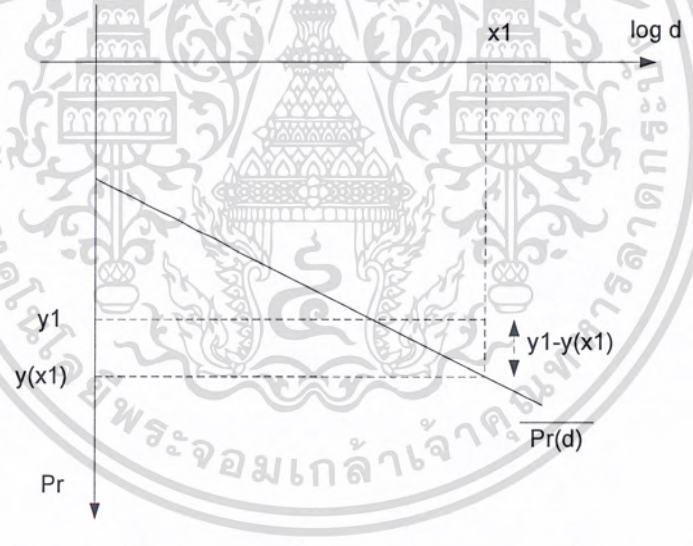
$$y = \overline{P_r(d)}[dB], A = -10n$$
$$x = \log d, B = \overline{P_r(1m)}[dB]$$

จะได้ y = Ax + B

กำหนดให้มีข้อมูลที่ได้จากการวัดจำนวน N ข้อมูล

$$x_i = \log d_i, y_i = P_r(d_i)[dB] \text{ ; i = 1,2,3,...,N}$$

จากสมการข้างต้น สามารถคำมาหาค่า error ของ $y_i - y(x_i)$ จาก $\overline{P_r(d)}$ ดังรูปที่ 2.3 เราหาค่า error ที่ตำแหน่ง i=1



รูปที่ 2.3 กราฟค่าความผิดพลาดของข้อมูลที่ i=1

จะได้ Error ของข้อมูลที่ i เป็น

$$E = y - y(x_i)$$
$$E = y_i - Ax_i - B$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จะได้ Total Square Error

$$\sum_{i=1}^{N} E_i^{\,2} = \sum_{i=1}^{N} (y_i - Ax_i - B)^2$$

ทำเงื่อนไข Least Square Error

$$\frac{\partial \sum_{i=1}^{N} E_i^{\,2}}{\partial A} = -2\sum_{i=1}^{N} (y_i - Ax_i - B) = 0$$

$$\frac{\partial \sum_{i=1}^{N} E_i^{\,2}}{\partial B} = -2\sum_{i=1}^{N} (y_i - Ax_i - B) = 0$$

จะได้

$$\sum_{i=1}^{N} x_i y_i - A\sum_{i=1}^{N} x_i^{\,2} - B\sum_{i=1}^{N} x_i = 0$$

$$\sum_{i=1}^{N} y_i - A\sum_{i=1}^{N} x_i - BN = 0$$

จัดรูปใหม่

$$A\sum_{i=1}^{N} x_i^{\,2} + B\sum_{i=1}^{N} x_i = \sum_{i=1}^{N} x_i y_i \qquad (2.4)$$

$$A\sum_{i=1}^{N} x_i + BN = \sum_{i=1}^{N} y_i \qquad (2.5)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3 คณิตศาสตร์พื้นฐานสำหรับการแก้ปัญหา Lateration

Multilateration เป็นเทคนิคหนึ่งที่นิยมใช้ในการประยุกต์การหาตำแหน่งวัตถุในระบบเครือข่ายเซนเซอร์ไร้สาย ใช้ความรู้ด้านคณิตศาสตร์ในการช่วยแก้ปัญหา สมมติให้มี 3 โหนดที่รู้ตำแหน่ง (x_i, y_i) โดยที่ i=1, 2, 3 จากทฤษฎีปีทากอรัส (Pythagoras Theorem) ดังรูปที่ 2.4

รูปที่ 2.4 การระบุตำแหน่งวัตถุด้วยเทคนิค Trilateration

และจะได้สมการ 3 สมการดังต่อไปนี้

$$(x_i - x_u)^2 + (y_i - y_u)^2 = R_i^2 \text{ เมื่อ i = 1, 2, 3} \qquad (2.6)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เพื่อแก้ปัญหาสมการข้างบน จะสร้างเซตของสมการเส้นตรงในรูปของ x_uและ y_u เพื่อที่จะทำให้เทอมกำลังสองของ x_u^2และ y_u^2 โดยการนำสมการ i=3 ลบออกจากสมการ i=1, 2 ตามลำดับ จะได้

$$(x_1 - x_u)^2 - (x_3 - x_u)^2 + (y_1 - y_u)^2 - (y_3 - y_u)^2 = R_1^2 - R_3^2$$
$$(x_2 - x_u)^2 - (x_3 - x_u)^2 + (y_2 - y_u)^2 - (y_3 - y_u)^2 = R_2^2 - R_3^2$$

ทำการจัดเทอม จะได้

$$2(x_3 - x_1)x_u + 2(y_3 - y_1)y_u = (R_1^2 - R_3^2) - (x_1^2 - x_3^2) - (y_1^2 - y_3^2)$$
$$2(x_3 - x_2)x_u + 2(y_3 - y_2)y_u = (R_2^2 - R_3^2) - (x_2^2 - x_3^2) - (y_2^2 - y_3^2)$$

เขียนให้อยู่ในรูปสมการเมตริกซ์

$$2\begin{bmatrix} x_3 - x_1 & y_3 - y_1 \\ x_3 - x_2 & y_3 - y_2 \end{bmatrix}\begin{bmatrix} x_u \\ y_u \end{bmatrix} = \begin{bmatrix} (R_1^2 - R_3^2) - (x_1^2 - x_3^2) - (y_1^2 - y_3^2) \\ (R_2^2 - R_3^2) - (x_2^2 - x_3^2) - (y_2^2 - y_3^2) \end{bmatrix} \quad (2.7)$$

จากนั้นแก้สมการหาค่า x_u และ y_u

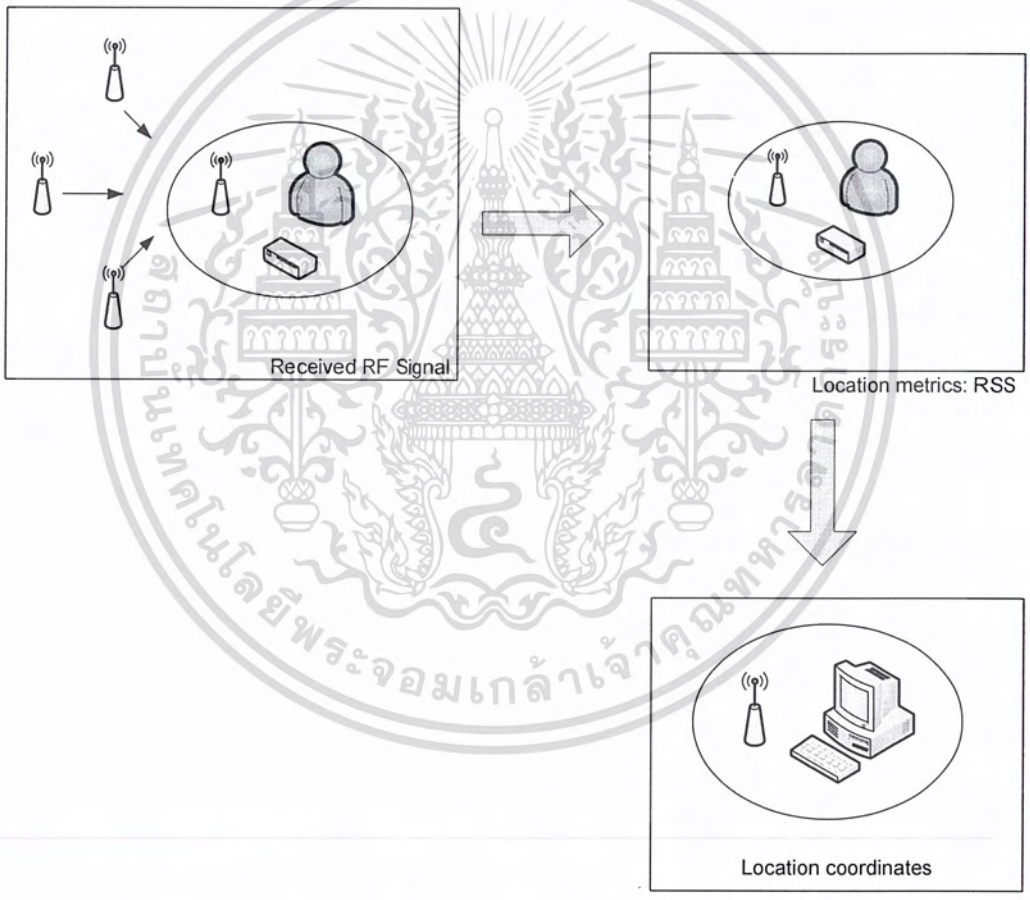
เมตริกซ์ในรูปทั่วไปที่ใช้หาตำแหน่งโดยวิธี Multilateration

$$2\begin{bmatrix} x_n - x_1 & y_n - y_1 \\ \vdots & \vdots \\ x_1 - x_{n-1} & y_1 - y_{n-1} \end{bmatrix}\begin{bmatrix} x_u \\ y_u \end{bmatrix} = \begin{bmatrix} (R_1^2 - R_n^2) - (x_1^2 - x_n^2) - (y_1^2 - y_n^2) \\ \vdots \\ (R_{n-1}^2 - R_n^2) - (x_{n-1}^2 - x_n^2) - (y_{n-1}^2 - y_n^2) \end{bmatrix}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4 หลักการทำงานของการระบุตำแหน่งวัตถุในระบบเครือข่ายไร้สาย

การระบุตำแหน่งวัตถุในระบบเครือข่ายไร้สายมีส่วนประกอบหลักๆ 3 ส่วน คือ อุปกรณ์สำหรับระบุตำแหน่ง, อัลกอริทึมสำหรับระบุตำแหน่ง และอุปกรณ์แสดงผล โดยอุปกรณ์สำหรับ ระบุตำแหน่งวัตถุจะรับสัญญาณจากวัตถุ และนำสัญญาณนั้นมาคำนวณโดยใช้อัลกอริทึมต่างๆในการค้นหาตำแหน่ง เช่น TOA, AOA และ RSSI หลังจากได้ตำแหน่งของวัตถุแล้วจึงนำไปแสดงผล ดังรูปที่ 2.5



รูปที่ 2.5 Function Block Diagram ของระบบระบุตำแหน่งในเครือข่ายไร้สาย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.5 สิ่งที่ทำให้เกิดข้อผิดพลาดของสัญญาณ

การประมาณค่าตำแหน่งจากการวัดที่มีสัญญาณรบกวนสามารถปรับปรุงให้ดีขึ้นได้ หากเราเข้าใจถึงพฤติกรรมของข้อผิดพลาดที่เกิดขึ้น ชนิดของข้อผิดพลาดที่เกิดขึ้นกับชนิดของสัญญาณและสิ่งแวดล้อมของระบบ

2.5.1 Multipath Fading and Shadowing

ในการวัดความแรงของสัญญาณคลื่นวิทยุปัญหา Multipath Fading และShadowing เป็นสาเหตุให้เกิดความแปรปรวนได้ถึง 30-40 dB ในช่วงครึ่งความยาวคลื่น การกระจัดกระจายใกล้เครื่องรับจะทำให้เกิดการแปรผันทางมุม (Angle of Arrival) ทำให้ค่าการวัดคลาดเคลื่อนไป ถ้าใช้วิธี Time-of-Arrival เมื่อเกิด Multipath Fading จะทำให้เกิดค่าหน่วงเวลา (Delay) การเปรียบเทียบค่าเวลาคลาดเคลื่อนจากความเป็นจริง

2.5.2 Non line-of-Sight (NLOS)

สำหรับวิธีวัดมุม (AoA) เมื่อใช้กับระยะทางไกลๆจะทำให้เกิดความคลาดเคลื่อนสูง สำหรับวิธีวัดความต่างของเวลา (Tof) ถ้าเส้นทางไปยังเครื่องรับที่ใกล้ที่สุดถูกกีดขวางจะทำให้ผลของการวัดระยะทางไกลกว่าที่เป็นจริง

2.5.3 Multiple-Access Interference

ปัญหานี้มักเกิดกับระบบ CDMA โดยเครื่องที่มีกำลังสูงไปรบกวนเครื่องที่มีกำลังต่ำกว่า สามารถเกิดกับระบบคลื่นเสียงและระบบคลื่นอัลตราโซนิค (Ultrasonic) ในระบบที่ใช้การวัดความแรงของสัญญาณที่ได้ (Received Signal Strength Indicator (RSSI)) จากคลื่นวิทยุราคาถูก เมื่อระบบไม่ใช้อุปกรณ์ที่มีความแม่นยำสูง ค่าความแปรปรวนจะเกิดจากความคลาดเคลื่อนเมื่อใช้เครื่องแปลงกำลัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.5.4 Fluctuations in Signal Propagation Speeds

เกิดกับคลื่นเสียงที่การแพร่ถูกรบกวนจากปัจจัยภายนอก เช่น แรงลมหรืออุณหภูมิ และค่าความชื้นในอากาศ ปัจจัยเหล่านี้จะทำให้ค่าความคลาดเคลื่อนสูงขึ้นเมื่อระยะทางมากขึ้น

2.6 ZigBee

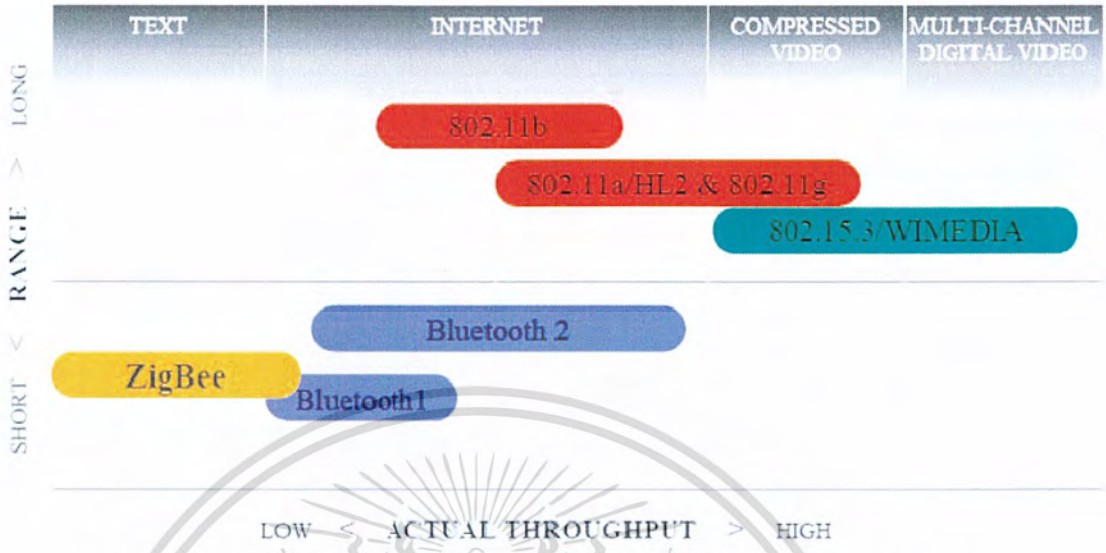
ZigBee เป็นเทคโนโลยีการสื่อสารแบบไร้สายที่ถูกพัฒนาให้มีจุดเด่นกว่าเทคโนโลยีไร้สายแบบอื่นๆ เช่น ราคาต่ำ ใช้พลังงานน้อยสามารถติดตั้งไว้ได้นาน และสามารถสร้างเครือข่ายได้ซึ่งเหมาะกับการใช้งานด้านเซนเซอร์ไร้สาย ตรวจสอบตำแหน่งของวัตถุ ตรวจสอบสภาพแวดล้อม ดังตารางที่ 2.1 และรูปที่ 2.6

ตารางที่ 2.1 การเปรียบเทียบเทคโนโลยีไร้สายในแบบต่างๆ

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	1-7
Network Size (# of nodes)	>64,000	32	7
Application	Monitoring & Control	Web,Email, Video	Cable Replacement
Stack Size (KB)	4-32	1,000	250
Throughput kb/s)	20-250	11,000	720

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง



รูปที่ 2.6 ปริมาณการใช้ Throughput ของมาตรฐานการสื่อสารไร้สายแบบต่างๆ

2.6.1 ชนิดอุปกรณ์ของ ZigBee

แบบ Physical Device มี 2 ประเภท คือ

1) Full Function Device : FFD เป็นเราท์เตอร์ที่เป็นสื่อกลางในการส่งข้อมูลจากอุปกรณ์อื่นๆ ใช้พลังงานจาก Power Line ทำงานได้ในทุกทอพอโลยี (Topology) และสามารถทำเป็นจุดเชื่อมต่อกันได้
2) Reduced Function Device : RFD เหมาะแก่การเชื่อมต่อภายในเครือข่าย ใช้พลังงานจากแบตเตอรี่ ไม่สามารถถ่ายทอดข้อมูลจากอุปกรณ์อื่นๆได้ ทำได้ง่ายในเครือข่ายแบบสตาร์ (Star)

แบบ Logical Device มี 3 ประเภท คือ

1) ZigBee Coordinators เป็นจุดที่ประสานเชื่อมต่อกัน ทำหน้าที่เริ่มจัดการเชื่อมโครงข่าย จัดการโหนดในโครงข่ายและเก็บข่าวสารของโหนดในโครงข่าย
2) ZigBee Routers ทำหน้าที่จัดการเส้นทางของข้อความที่ส่งผ่านภายในโครงข่ายโหนดใดๆ

117480
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3) ZigBee End Devices เป็นโหนดที่อยู่ในส่วนของผู้ใช้งาน โดยสามารถเป็นได้ทั้งแบบ RFD และ FF

2.6.2 การประยุกต์ใช้งาน ZigBee

แบ่งแยกตามประเภทของข้อมูลข่าวสาร ที่มีอยู่ 3 แบบคือ

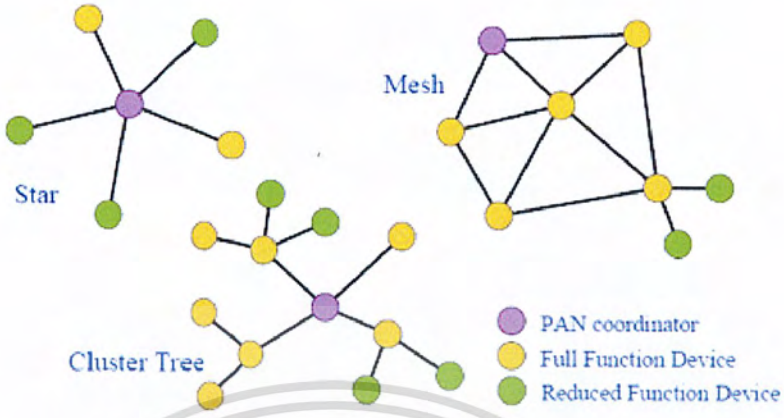
1) ข้อมูลแบบ Periodic ข้อมูลเป็นช่วงเวลา โปรแกรมสามารถควบคุมอัตราการส่งและตัวตรวจจับสัญญาณกระตุ้น เช็คข้อมูลและทำให้ข้อมูลไม่เคลื่อนไหว ใช้สำหรับ เซนเซอร์ และ มิเตอร์
2) ข้อมูลแบบ Intermittent เป็นลักษณะที่มีการส่งผ่านข้อมูลเมื่อมีการใช้งาน เช่น สวิตซ์ไฟ
3) ข้อมูลแบบ Repetitive low latency ใช้ในงานที่ต้องการ Latency น้อย ๆ โดยการสื่อสารจะใช้วิธีจัดสรรช่องเวลา และสามารถใช้กลไกแบบ GTS เพื่อรับประกันคุณภาพของการบริการ นำไปใช้ในงาน เช่น เมาส์ไร้สาย

2.6.3 โครงสร้างของโพรโทคอล ZigBee

โพรโทคอล ZigBee ถูกออกแบบมาเฉพาะในส่วนของ Application Layer, Application Support Layer และ Network Layer เท่านั้น แต่ใช้ MAC Layer และ Physical Layer ตามมาตรฐาน IEEE 802.15.4 ชั้น Network นี้สามารถรองรับ Topologies คือแบบ Star, Mesh และ Cluster-Tree ซึ่งโครงสร้างแสดงดังรูปที่ 2.7

ในการเชื่อมต่อแบบ Star นั้น จะใช้งานกับอุปกรณ์พื้นฐาน เช่น เซนเซอร์ ที่มีช่วงการทำงานได้นาน ส่วนการเชื่อมต่อแบบ Mesh หรือ Peer-to-Peer นั้นเป็นการส่งผ่านข้อมูลภายในโครงข่ายของอุปกรณ์ จะส่งออกไปในหลายเส้นทาง และมีระดับของความเชื่อถือและแบ่งแยกได้สูง ส่วนแบบ Cluster-Tree จะเป็นการรวมกันแบบ Star กับ Mesh เพื่อที่จะได้ประโยชน์จากทั้งสองแบบ คือมีระดับของความน่าเชื่อถือได้และรองรับโหนดที่สามารถใช้พลังงานจากแบตเตอรี่ได้ยาวนาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.7 เครือข่าย ZigBee แบบ Star , Cluster , Mesh

มาตรฐานของ ZigBee มีการแบ่งเป็น Layer ซึ่ง Layer เหล่านี้จะทำให้การใช้งานได้มีประสิทธิภาพมากขึ้น ราคาถูก ติดตั้งง่าย การส่งข้อมูลที่น่าเชื่อถือ ใช้พลังงานน้อยแบ่งเป็น Layer ต่างๆ มีหน้าที่ต่างกัน ดังนี้

2.6.3.1 Application Layer

Application Layer ของ ZigBee ประกอบด้วย ZigBee Device Object (ZDO), User-Defined Application Profile(s) and the Application Support (APS) Sub-Layer

Application Support Layer (APS) ทำหน้าที่ดังนี้

1) Discovery สามารถค้นหาและระบุได้ว่าอุปกรณ์ตัวใดติดต่ออยู่กับอุปกรณ์ตัวใดอยู่
2) Binding สามารถจับคู่อุปกรณ์ไว้ด้วยกันได้โดยใช้ Table of Binding และ Forward Messages ระหว่างอุปกรณ์

ZigBee Device Object (ZDO) ทำหน้าที่ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1) กำหนดหน้าที่ของอุปกรณ์แต่ละตัวภายใน Network เช่น ZigBee Coordinator or End Device

2) สร้างหรือตอบสนอง Binding Requests

3) สร้างระบบรักษาความปลอดภัยระหว่าง Network Device โดยเลือกจาก ZigBee's Security Methods เช่น public key, symmetric key

User-Defined Application หมายถึง End Device ซึ่งเป็นไปตามมาตรฐาน ZigBee

2.6.3.2 Network Layer

Network Layer ถูกออกแบบมาเพื่อทำให้การส่งข้อมูลใน Network ใช้พลังงานไม่มาก สามารถจัดการกับ Network ที่มีจำนวนโนดมากๆ โดยใช้ Latency ทำหน้าที่ดังนี้

1) Starting a network สามารถสร้าง Network ขึ้นใหม่ได้

2) Joining and leaving a network สามารถเข้าร่วมและออกจาก Network ได้

3) Configuring a new device สามารถกำหนดค่าของ Stack ได้

4) Addressing ZigBee coordinator กำหนด Address ให้กับอุปกรณ์แต่ละตัวได้

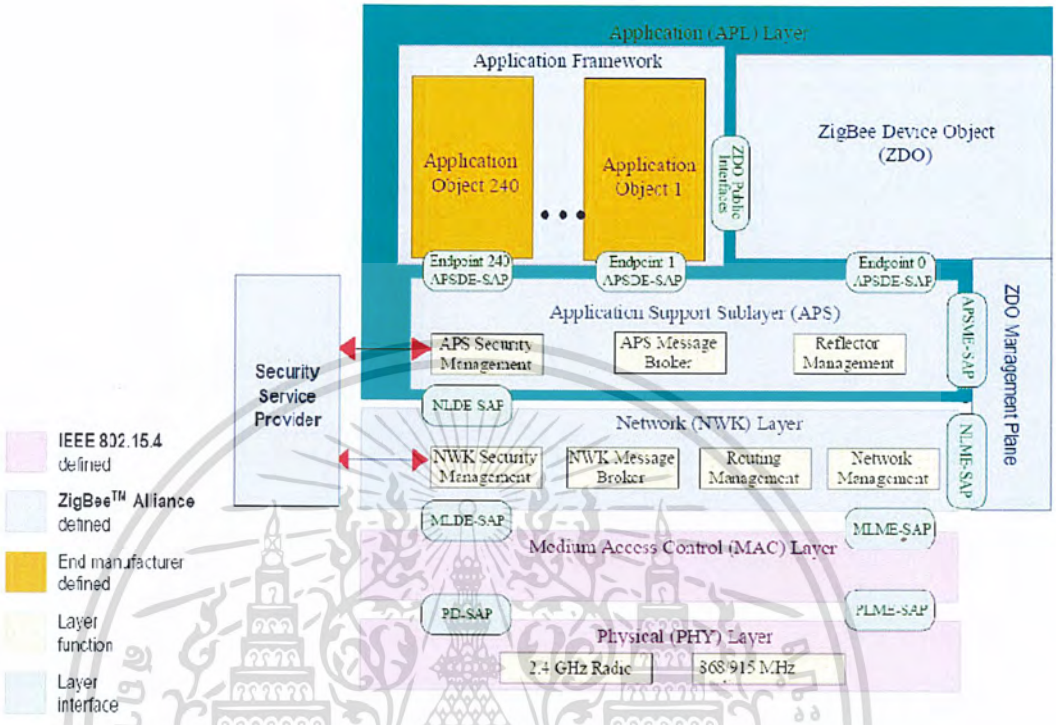
5) Synchronization within a network สามารถติดต่อกับอุปกรณ์อื่นๆแบบ Synchronization ได้

6) Security ทำให้เฟรมรับส่งมีความปลอดภัย

7) Routing จัดหาเส้นทางของเฟรมปลายทาง

โครงสร้างและการทำงานของ IEEE 802.15.4 / ZigBee Stack แสดงดังรูปที่ 2.8

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.8 IEEE 802.15.4 / ZigBee Stack

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

2.6.4 ขั้นตอนการทำงานของโพรโทคอล ZigBee

ขั้นตอนการทำงานของ ZigBee Coordinator จะเริ่มต้นเครือข่าย โดยการตรวจสอบการใช้ช่องสัญญาณวิทยุภายในบริเวณรอบๆ ถ้ามีช่องสัญญาณที่ไม่ถูกใช้โดย Coordinator ตัวอื่นก็สามารถเริ่มต้นเครือข่ายได้ หลังจากนั้น Cordinator ก็จะทำหน้าที่เป็นศูนย์กลางของเครือข่าย รองรับการเข้าร่วมของเครือข่ายของ ZigBee End-Device และรองรับการร้องขออื่นๆตามมาตรฐานด้วยเช่นกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขั้นตอนการทำงานของ ZigBee End-Device จะเริ่มต้นการทำงานโดยการร้องขอการเข้าร่วมเครือข่ายไปยัง Coordinator ประจำเครือข่ายนั้นๆ โดยการตรวจสอบผ่านช่องสัญญาณต่างๆว่า Coordinator ใช้ช่องสัญญาณใดอยู่เมื่อเข้าร่วมเครือข่ายแล้ว End-Device จึงสามารถทำการร้องขอคำสั่งอื่นๆผ่านทาง Coordinator ได้ เช่น การส่งข้อความทั่วไป (Message), การร้องขอ Binging (Binding Request), การขอออกจากเครือข่าย

2.6.5 การส่งข้อมูลของ ZigBee

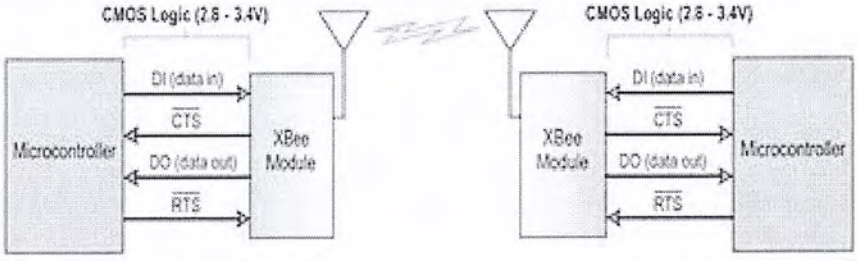
การส่งข้อมูลแบบ RF ของแต่ละแพ็กเกจในส่วนของ header จะประกอบไปด้วย Source Address และDestination Address โดยที่ IEEE 802.15.4 จะมีโครงสร้าง 2 แบบ นั่นคือ แบบ short 16-bit addresses และแบบ long 64-bit addresses ซึ่ง64-bit จะสามารถอ่านคำสั่ง SL (Serial Number Low) และ SH (Serial Number High) และการส่งข้อมูลแบบ RF จะส่งได้ 2 โหมด คือ Unicast Mode และ Broadcast Mode

1) การส่งแพ็คเกจโดยใช้โครงสร้าง 16-bit addressing ให้ตั้งค่า ตัวแปร DL (Destination Address Low) ให้เท่ากับ ตัวแปร MY และตั้งค่าตัวแปร DH (Destination Address High) เป็น '0'
2) การส่งแพ็คเกจโดยใช้โครงสร้าง 64-bit addressing ให้ตั้งค่า Destination Address (DL + DH) ให้เข้ากับ Source Address (SL + SH) ของปลายทางที่เราจะส่งแพ็คเกจไป

2.6.6 การติดต่อสื่อสารของ ZigBee ผ่านทางพอร์ตอนุกรม

ZigBee อินเตอร์เฟสกับอุปกรณ์ผู้ใช้ (Host) ผ่านทางพอร์ตอนุกรม อะซิงโครนัสระดับตรรกะ ส่งผ่านโมดูลทางพอร์ตอนุกรม โมดูลสามารถติดต่อสื่อสารกับตรรกะต่างๆเข้ากับ UART หรือระดับตัวแปรกับอุปกรณ์อนุกรมต่างๆ UART การไหลของข้อมูล อุปกรณ์มีอินเตอร์เฟส UART สามารถติดต่อโดยตรงกับพินของ RF โมดูล ดังรูปที่ 2.9

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูป 2.9 การต่อไมโครคอนโทรลเลอร์กับซิกบี

ข้อมูลส่งขาเข้า UART โมดูล ผ่านขา Data In (ขา 3) ด้วยสัญญาณอนุกรมอะซิงโครนัสสัญญาณจะว่างเมื่อไม่มีข้อมูลถูกส่ง แต่ละข้อมูลไบต์ประกอบด้วย บิตเริ่มต้น (บิตต่ำ) 8 ข้อมูลบิต และบิตหยุด (บิตสูง) ตามรูปแบบงานโมดูล UART ดังนั้นเวลาและพาริตี้ตรวจสอบต้องการข้อมูลติดต่อสื่อสาร การสื่อสารอนุกรมขึ้นอยู่กับ 2 UARTS ถูกจัดโครงสร้างกับการจัดตั้งอัตราบอร์ด (baud rate), พาริตี้ (parity), บิตเริ่มต้น (start bit), บิตหยุด (stop bit) และบิตข้อมูล (data bit)

2.6.7 ลักษณะทางกายภาพของโมดูล XBee

ลักษณะทางกายภาพเปรียบเทียบระหว่าง XBee กับ XBee-Pro แสดงดังรูปที่ 2.10 และตารางแสดงขาสัญญาณโมดูล XBee และ XBee Pro ดังตารางที่ 2.2

2.7 ไมโครคอนโทรลเลอร์ (Microcontroller)

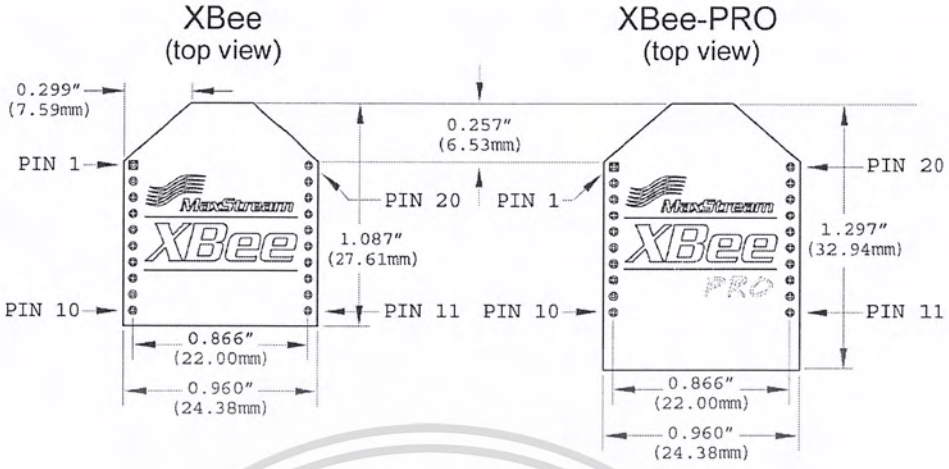
ไมโครคอนโทรลเลอร์ คือ อุปกรณ์อิเล็กทรอนิกส์อย่างหนึ่งซึ่งภายในประกอบด้วยวงจรอื่น ๆ หลายวงจรทำงานร่วมกัน เช่น หน่วยประมวลผลกลาง (Central Processing Unit: CPU) หน่วยคำนวณทางคณิตศาสตร์และลอจิก (Arithmetic Logic Unit: ALU) วงจรออสซิลเลเตอร์ (Oscillator) หน่วยความจำ (Memory: ROM, RAM) วงจรรับสัญญาณอินพุทและขับสัญญาณเอาท์พุท (I/O Port) ซึ่งโครงสร้างของไมโครคอนโทรลเลอร์ แสดงดังรูปที่ 2.11

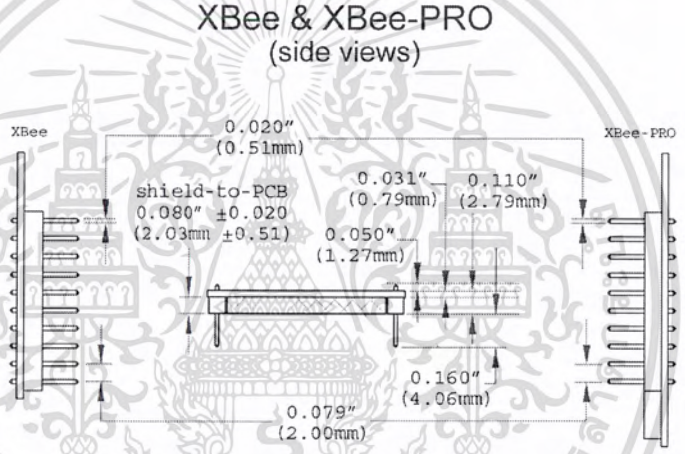
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.2 ขาสัญญาณโมดูล XBee และ XBee Pro

Pin #	Name	Direction	Description
1	VCC	-	Power supply
2	DOUT	Output	UART Data Out
3	DIN/CONFIG	Input	UART Data In
4	DO8*	Either	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	PWM	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	ADP/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/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	AD3/DIO3	Either	Analog Input 2 or Digital I/O 2
19	AD3/DIO3	Either	Analog Input 1 or Digital I/O 1
20	AD3/DIO3	Either	Analog Input 0 or Digital I/O 0

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



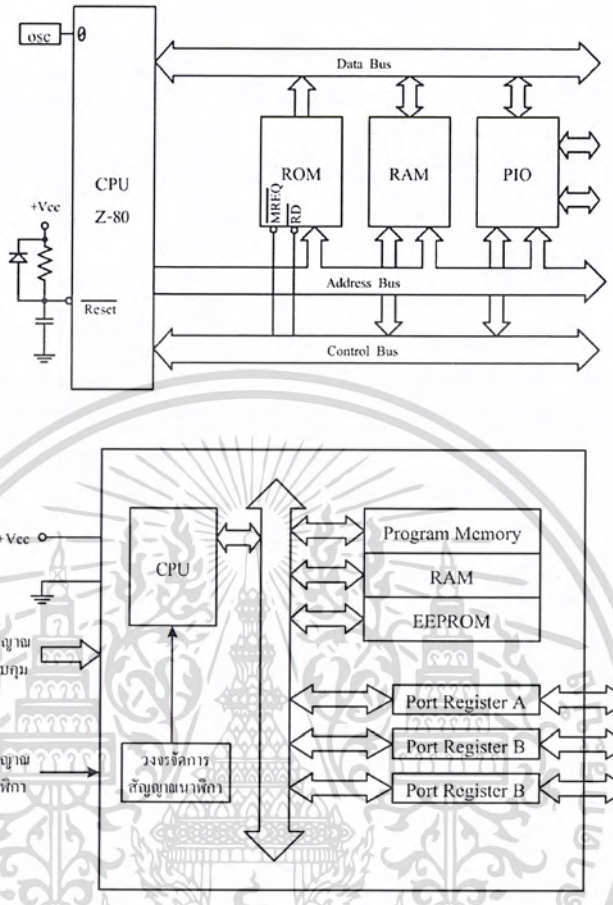


รูปที่ 2.10 ลักษณะทางกายภาพของโมดูล XBee

2.7.1 PIC (Peripheral Interface Controller)

PIC คือ Microcontroller อีกตระกูลหนึ่ง ซึ่ง Concept ของMicrocontroller ตระกูลนี้ก็คือ พยายามรวมเอาทุกอย่างเอาไว้ในตัวของมันไม่ว่าจะเป็น PROGRAM MEMROY, RAM, EEPROM, SERIAL, I2C, PWM, A/D ฯลฯ โดยไม่จำเป็นต้องต่ออุปกรณ์เสริมจากภายนอก ในตัวของ PIC จะมีฟังก์ชันที่ใช้ในการประมวลผล รวมทั้งหน่วยความจำ ซึ่งทำให้มันเหมือนกัน CPU ตัวหนึ่งเลยทีเดียว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.11 โครงสร้างภายในไมโครคอนโทรลเลอร์

ภาษาที่ใช้สำหรับการเขียนโปรแกรมบน Microcontroller แบ่งได้เช่นเดียวกับการเขียนโปรแกรมบนคอมพิวเตอร์คือ ภาษาระดับสูง และภาษาระดับต่ำ

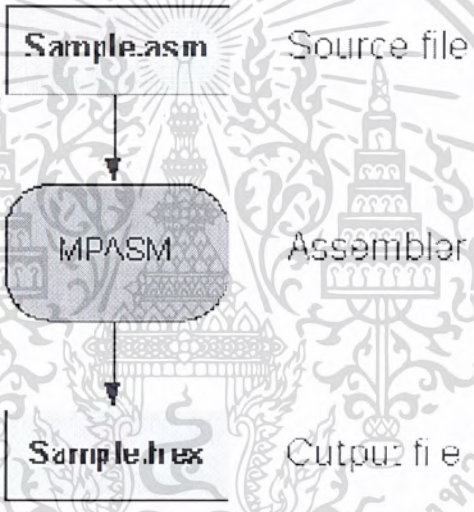
1) ภาษาระดับสูงเช่น C, Basic ข้อดีคือเขียนง่าย, แก้ไขเปลี่ยนแปลง หรือเพิ่มเติมได้ง่าย ส่วนข้อเสียก็คือการทำงานจะช้า ขนาดโปรแกรมที่เขียนมีขนาดใหญ่

2) ภาษาระดับต่ำ ซึ่งก็คือ ภาษา Assembly ข้อดีคือ ตัว Compiler แจกฟรี ขนาดโปรแกรมหลังจาก Compiled แล้วมีขนาดเล็ก โปรแกรมมีความเร็ว แต่ข้อเสียก็คือเขียนยาก เพราะลักษณะภาษาไม่ค่อยสื่อความหมาย แก้ไขเปลี่ยนแปลงยาก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.7.2 รูปแบบการเขียนโปรแกรม Microcontroller

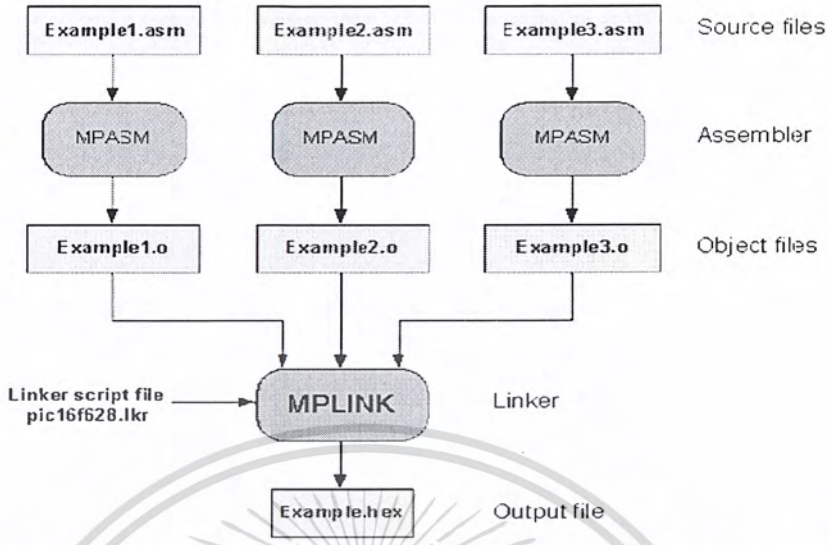
เขียนด้วยภาษา Assembly แบบ ไฟล์เดียว หลังจากนั้นก็ทำการ Compile ด้วย Assembler ของ MCU ตัวนั้น ซึ่งส่วนในผู้ผลิต Chip MCU จะแจกจ่ายให้ฟรี สำหรับ Assembler ของ Microchip ก็คือ MPASM โดยไฟล์ที่ได้มามีได้หลายชนิดแต่ส่วนใหญ่จะอยู่ในรูปของ Hex file ซึ่งรูปแบบของโปรแกรมไมโครคอนโทรลเลอร์ด้วย Assemblyไฟล์เดียว แสดงเป็นขั้นตอนลำดับ ตามรูปที่ 2.12



รูปที่ 2.12 รูปแบบการโปรแกรมไมโครคอนโทรลเลอร์ด้วย Assemblyไฟล์เดียว

ใช้ภาษา Assembly เช่นกัน แต่แบ่งเป็นหลายๆ ไฟล์ หลังจากนั้นก็จะ Compile แต่ละไฟล์ให้ออกมาเป็น Object files และทำการรวมกันด้วย Linker ในขณะทำการ Link ก็จะมี Script File ของ MCU เบอร์นั้นๆ ประกอบ หลังจากทำการ Link แล้วก็จะได้ Hex file ออกมา ซึ่งรูปแบบของโปรแกรมไมโครคอนโทรลเลอร์ด้วย Assemblyหลายไฟล์ แสดงเป็นขั้นตอนลำดับ ตามรูปที่ 2.13

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

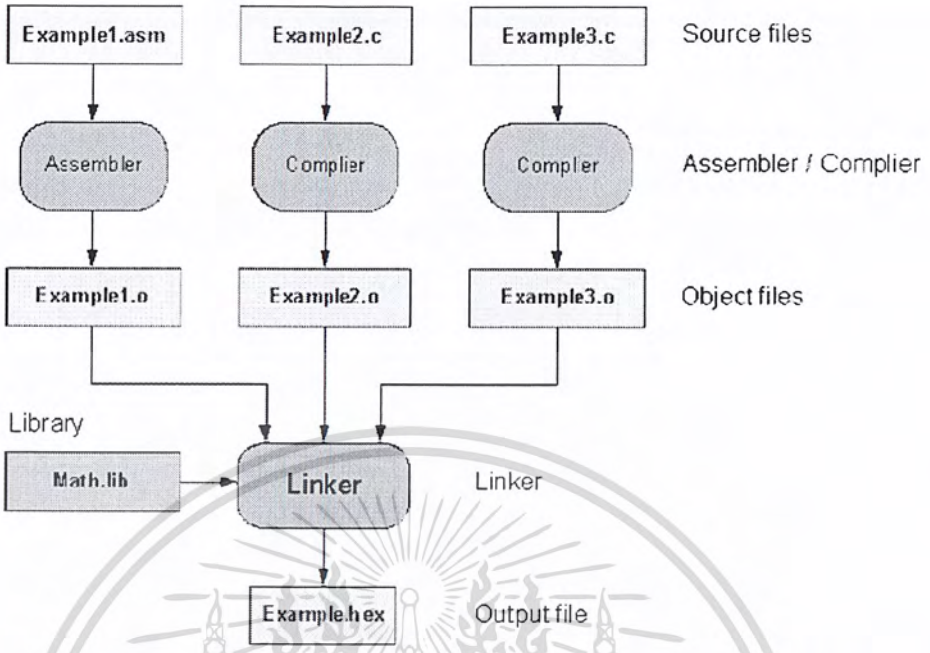


รูปที่ 2.13 รูปแบบการโปรแกรมไมโครคอนโทรลเลอร์ด้วย Assembly หลายไฟล์

การเขียนด้วยภาษาสูง อาจจะเป็น C, Basic ฯลฯ ซึ่งอาจจะเขียนร่วมกับ ภาษา Assembly โดยไฟล์ที่เขียนจะถูกทำให้กลายเป็น Object files โดย Assembler สำหรับภาษา Assembly และ Compiled โดย Compiler สำหรับภาษาสูง ทำการ Link เข้าด้วยกันด้วย Linker ซึ่งขณะทำการ Link ก็จะมีการรวมเอา Library ที่ถูกเรียกใช้ในโปรแกรมเข้าไปรวมด้วยกัน สุดท้ายก็จะออกมาเป็น Hex file ซึ่งรูปแบบของโปรแกรมไมโครคอนโทรลเลอร์ด้วยภาษาขั้นสูงแสดงเป็นขั้นตอนลำดับ ตามรูปที่ 2.14

หลังจากได้ Hex file มาแล้ว เราก็จะทำการอัดโปรแกรมเข้าสู่ Chip ด้วยตัวโปรแกรมเมอร์ ส่วนใหญ่จะมีรูปแบบคือ มี Software บนคอมพิวเตอร์ สำหรับใช้ในการควบคุมการอ่าน เขียน หรือ ลบ โดยส่วนใหญ่จะเชื่อมต่อไปยัง Programmer ด้วย Serial, Parallel มีราคาให้เลือกตั้งแต่หลักร้อยไปจนถึงหลักหมื่น เมื่ออัดโปรแกรมเข้า Chip ได้แล้วเราก็พร้อมจะนำไปทดสอบการทำงานต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.14 รูปแบบโปรแกรมไมโครคอนโทรลเลอร์ด้วยภาษาขั้นสูง

2.7.3 ชนิดของ PIC

MCU ในตระกูล PIC ถ้าแบ่งออกตามชนิดของ PROGRAM MEMORY แบ่งได้เป็น 3 แบบคือ

1) OTP (One Time Programmable) เป็น Chip ที่มีราคาถูกที่สุด ในสามประเภท สาเหตุก็มาจากว่า Chip แบบ OTP จะสามารถทำการโปรแกรมได้แค่ครั้งเดียวเท่านั้น หลักจาก Chip ได้ถูกโปรแกรมไปแล้วจะไม่สามารถโปรแกรมเข้าไปใหม่ได้อีก ดังนั้น Chip ประเภทนี้จะนิยมใช้หลังจากได้พัฒนาโปรแกรมจนกระทั่งจุดบกพร่องต่างๆ ในโปรแกรม ไม่มีอีกแล้ว เพราะจะมีต้นทุนต่ำเมื่อเทียบตัว Memory ประเภทอื่น จะมีตัวอักษร C แสดงบนตัว Chip เช่น 16C84, 16C74
2) EPROM (Erasable Programmable ROM) เป็น Chip ที่มี Program Memory ที่เมื่อเขียนโปรแกรมเข้าไปแล้วสามารถโปรแกรมใหม่ด้วยการลบโปรแกรมเดิมโดย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ให้แสง UV (Ultra Violet) ส่องผ่านเข้าไปยัง Chip ประมาณ 5-10 นาที ดังนั้นที่ด้านบนของ Chip จะมีกรอบกระจกเพื่อให้แสดง UV สามารถส่องผ่านเข้าไปในตัว Chip ได้ แต่ก็มีจำนวนครั้งในการลบโปรแกรมเช่นกัน เมื่อลบโปรแกรมด้วยแสง UV มากๆ เข้าก็จะเกิดอาการด้าน คือโปรแกรมไม่เข้านั่นเอง จะมีตัวอักษร JW หรือว่าดูเอาว่ามีกรอบกระจกอยู่บน Chip หรือไม่

3) EEPROM / Flash (Electronically Erasable Programmable ROM) เป็น Chip ที่ออกมาไม่กี่ปีนี้เอง ส่วนของ Program Memory สามารถอ่านหรือเขียนด้วยสัญญาณทางไฟฟ้า ใช้เวลาในการ ลบข้อมูลไม่กี่วินาที และสามารถลบ และเขียนใหม่ได้หลายพันครั้ง ทำให้เป็นที่นิยมที่สุดใน 3 ประเภท มีตัวอักษร F เป็นตัวบอก เช่น 16F84, 16F877

2.7.4 PIC เบอร์ต่างๆ

ปัจจุบัน MCU ของ PIC มีหลากหลายเบอร์ แต่ในความเป็นจริงแล้ว การออกเบอร์ใหม่ๆ ออกมาได้มีการกำหนดทิศทางที่แน่นอน ซึ่งเราสามารถ จะแบ่งออกเป็น 6 กลุ่ม ได้แก่

1) PIC12CXXX, PIC12FXXX (FLASH MCUs) มี 8 ขา มีคำสั่งในภาษา Assembly ทั้งหมด 33 หรือ 35 คำสั่ง มี I/O, Timer, Watch dog บางเบอร์มี Internal EEPROM จะมี 4MHz Oscillator อยู่ภายในตัว Chip เลย ทำให้เราไม่ต้องมี Oscillator ภายนอก (แต่หากต้องการความแม่นยำของเวลาก็จำเป็นต้องใช้ Crystal Oscillator ภายนอก)

2) PIC16C5X มีคำสั่งในภาษา Assembly 33 คำสั่ง มี I/O, Timer, Watch dog มี Feature ค่อนข้างน้อย จะไม่มี I2C หรือ SERIAL เพราะเป็น Chip ที่ออกมาในยุคแรกๆ ของ Microchip เนื่องจากเป็นเบอร์ที่ออกมานานมากแล้ว จึงมีแต่แบบ OTP และ EPROM Feature ก็มีน้อย I2C หรือ USART นั้นไม่มีเลย Microchip คงเห็นจุดด้อยตรงนี้เลยเข็นตระกูลใหม่เข้ามาคือ PIC16CXXX

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3) PIC16CXXX มีคำสั่งในภาษา Assembly 35 คำสั่ง มี I/O, มี Timer มากกว่า 1 ตัว, Watch dog, I2C, USART, SPI, PWM เริ่มมี Feature ใส่เข้าไปมากขึ้นทำให้การ Interface กับอุปกรณ์ภายนอกทำได้ง่ายขึ้น
4) PIC17CXXX จะออกมาพร้อมๆกับ PIC16CXXX แต่ต่างกันที่ว่า PIC17CXXX ถือว่าเป็นเบอร์ที่ใหญ่กว่า มีคำสั่งใน Assembly มากกว่า รวมทั้งขนาดของ PROGRAM Memory จะมีขนาดมากกว่าในตระกูล PIC16CXXX และสามารถต่อ Memory ภายนอกก็ได้ มีคำสั่งในภาษา Assembly 58 คำสั่ง มีคำสั่งการ คูณ หาร สามารถต่อกับ Program Memory ภายนอก โดยอ้างได้ถึง 64K x 16Program Memory
5) PIC16F8XXX (FLASH MCUs) เป็น PIC ที่ได้รับความนิยมมากเพราะเป็น Chip ยุคแรกๆ ที่มี PROGRAM MEMORY เป็นแบบ FLASH และมี DATA MEMORY ที่เป็น EEPROM เพิ่มเติมเข้ามา รวมทั้งตัว PROGRAM CHIP สามารถสร้างได้ง่าย อีกทั้งยังมี In Circuit Debugger อยู่ภายในทำให้ ไม่จำเป็นต้องพึ่ง Emulator ที่มีราคาแพงอีกต่อไป
6) PIC18CXXX, PIC18FXXX (FLASH MCUs) จุดด้อย คือ Program Memory มีลักษณะเป็น Page จึงได้พัฒนาออกมาเป็น PIC18CXXX ซึ่ง PROGRAM MEMORY ไม่ถูกแบ่งเป็น Page อีกต่อไป และเพิ่มคำสั่ง ASSEMBLY เป็น 77 คำสั่ง รวมทั้งออกแบบให้ PROGRAM MEMORY มีขนาดใหญ่ขึ้นเพื่อรองรับการเขียนโปรแกรมภาษา C หลังจากนั้นจึงได้ออก PIC18FXXX ตามมา พร้อมกับ ICD2

มีคำสั่งในภาษา Assembly 77 คำสั่ง มี I/O, มี Timer มากกว่า 1 ตัว, Watch dog, I2C, USART, SPI, PWM, CAN มี A/D ขนาด 10 bits สามารถต่อกับ Program Memory ภายนอก โดยอ้างได้ถึง 64K x 16 มี Program Memory เป็นแบบ Flash ในตระกูล 18FXXX มี EEPROM ภายในในตระกูล 18FXXX สนับสนุน In Circuit Debugging (ICD)

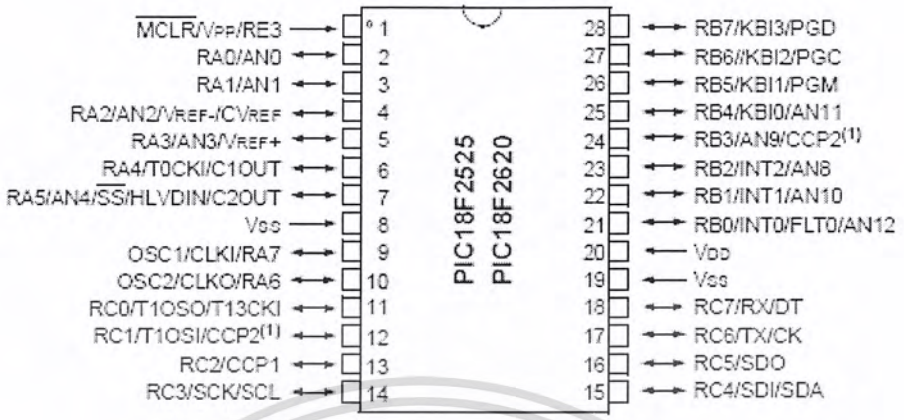
PIC ที่ใช้ในการทำโครงงานนี้คือ PIC 18F2525 มีตำแหน่งขาและคุณสมบัติดังรูปที่ 2.15 และตารางที่ 2.3 ตามลำดับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.3 คุณสมบัติของ PIC 18F2525

คุณสมบัติ	PIC18F2525
Operating Frequency	DC – 40 MHz
Program Memory (Bytes)	49152
Program Memory (Instructions)	24576
Data Memory (Bytes)	3968
Data EEPROM Memory (Bytes)	1024
Interrupt Sources	19
I/O Ports	Ports A,B,C,(E)
Timers	4
Capture/Compare/PWM Modules	2
Enhanced Capture/Compare/PWM Modules	0
Serial Communications	MSSP, Enhanced USART
Parallel Communications (PSP)	No
10-Bit Analog-to-Digital Module	10 Input Channels
Resets (and Delays)	POR, BOR, RESET Instruction, Stack Full, Stack Underflow(PWRT, OST), MCLR (optional), WDT
Programmable Low-Voltage Detect	Yes
Programmable Brown-out Reset	Yes
Instruction Set	75 Instructions; 83 with Extended Instruction Set Enabled
Packages	28-Pin SPDIP 28-Pin SOIC

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.15 ตำแหน่งขา (PIN Diagram) ของ PIC 18F2525

2.8 การส่งข้อมูลในรูปแบบของ API

การส่งข้อมูลในรูปแบบ API (Application Programming Interface) คือการจัดเก็บข้อมูลที่ต้องการจะติดต่อสื่อสารกับโมดูลอื่นๆในรูปแบบโครงสร้างที่แน่นอน โดยจัดเก็บเป็น Frame Packet ซึ่งสามารถบอกได้ว่า Packet ที่ส่งไปนั้นมีคำสั่งอะไร การส่งข้อมูลถูกต้องหรือไม่ และมีสถานะของการส่งและการรับของ Packet นั้นๆด้วย ซึ่งข้อมูลทีส่งไปจะอยู่ในรูปแบบของเลขฐานสิบหก

2.8.1 รูปแบบของ API Frame

เราสามารถส่งข้อมูลในรูปแบบของ API Mode ได้โดยการตั้งค่าพารามิเตอร์ของคำสั่ง AP (API Enable) ดังนี้

AP = 0 เป็นการตั้งค่าการส่งข้อมูลให้อยู่ในรูปแบบของ UART แบบปกติ API Mode จะถูกยกเลิก

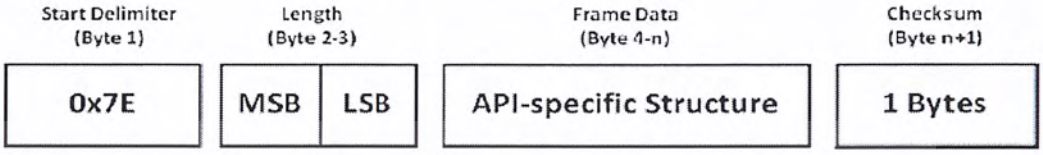
AP = 1 เป็นการตั้งค่าการส่งข้อมูลให้อยู่ในรูปแบบของ API Mode

AP = 2 เป็นการตั้งค่าการส่งข้อมูลให้อยู่ในรูปแบบของ API Mode (Escaped Character)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในโครงงานเราใช้การส่งข้อมูลในรูปแบบของ API Mode (AP = 1)

API Mode (AP = 1) เมื่อพารามิเตอร์ AP ถูกตั้งค่าเป็น 1 โครงสร้างของข้อมูลที่ส่งจะมีลักษณะดังรูปที่ 2.16



รูปที่ 2.16 รูปแบบของ API Packet frame Mode 1

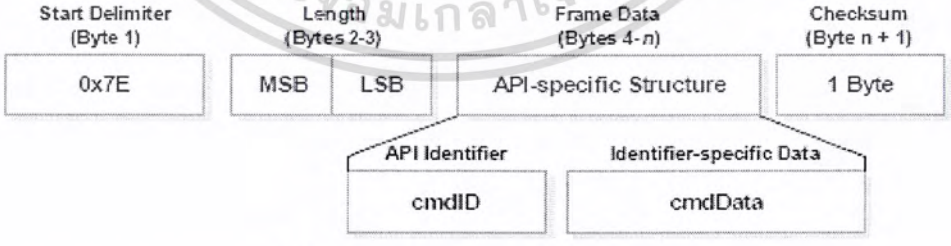
MSB = Most Significant Byte

LSB = Last Significant Byte

Checksum คือ ไบต์ที่ตรวจสอบความถูกต้องของข้อมูลที่ได้รับหรือส่ง สามารถคำนวณหาค่า Checksum ได้โดยบวก Frame Data ทุกไบต์ในลักษณะของเลขฐานสิบหก หลังจากนั้นนำไบต์สุดท้ายไปลบออกจาก 0xFF ก็จะได้ค่า Checksum ที่ถูกต้อง

2.8.2 ประเภทของ API

รูปแบบของการส่งข้อมูลมีโครงสร้างแพ็คเกต ดังรูปที่ 2.17



รูปที่ 2.17 รูปแบบของ API Packet frame

cmdID (API Identifier) คือ ไบต์ที่เป็นตัวกำหนดรูปแบบของ API ว่าอยู่ในประเภทใด เช่น อยู่ในรูปแบบของ AT Command การรับข้อมูล หรือการส่งข้อมูล เป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

cmdData (Identifier-specific Data) คือ รูปแบบของข้อมูลที่ต่างๆ ที่ถูกกำหนดโดย cmdID

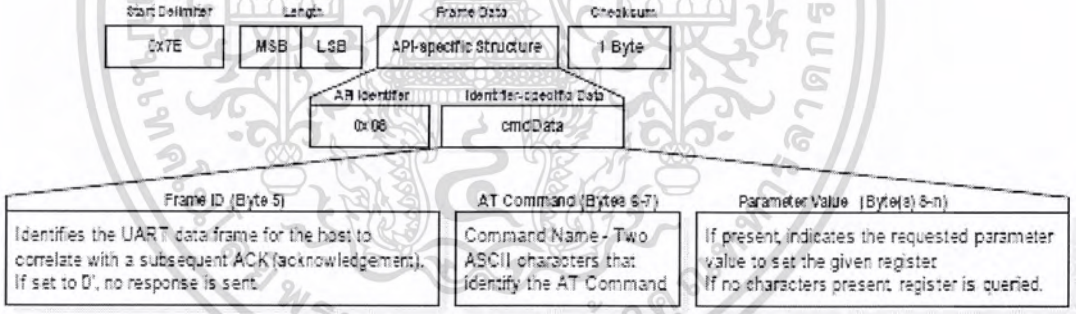
2.8.2.1 AT Command

API frame ประเภท AT Command เป็นแพ็คเกตที่สามารถใช้เขียนหรืออ่านค่าพารามิเตอร์ของโมดูล ZigBee ได้ ซึ่งใช้คำสั่ง AT Command ในรูปแบบนี้ไม่จำเป็นต้องอยู่ในโหมด AT Command ค่าต่างๆสามารถอ่านหรือแก้ไขได้ทันที โดยจะมีค่า API Identifier เป็น 0x08 โครงสร้างของ AT Command frame ดังรูปที่ 2.18

Frame ID คือ ค่า ID ของ frame ที่ส่งไป ถ้าถูกตั้งค่าเป็น 0 ACK จะไม่ถูกส่งกลับมา

AT Command คือ ชื่อของ AT Command ประกอบด้วยตัวอักษร 2 ตัว

Parameter Value คือ ค่าพารามิเตอร์ของ AT Command นั้นๆที่ต้องการตั้งค่า



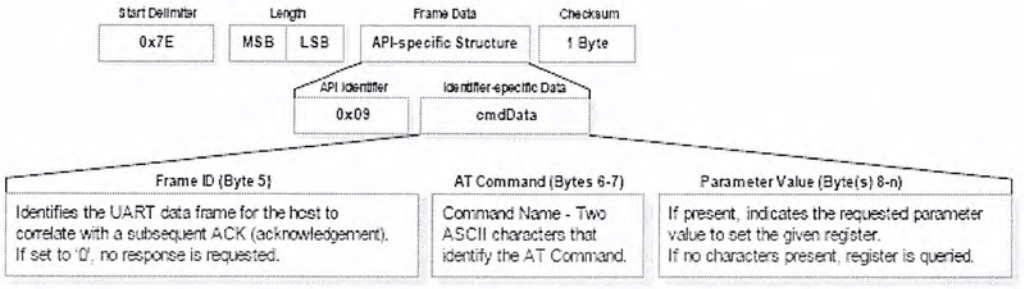
รูปที่ 2.18 รูปแบบของ API Packet frame ประเภท AT Command

2.8.2.2 AT Command Queue Parameter Value

API frame ประเภทนี้จะเหมือนกับ API frame AT Command ซึ่งสามารถใช้เขียนหรืออ่านค่าพารามิเตอร์ของโมดูล ZigBee ได้โดยใช้ AT Command แต่ API frame ประเภทนี้จะทำงานก็ต่อเมื่อ AT Command (API Identifier = 0x08) ทำงานเสร็จ ซึ่ง AT Command Queue

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

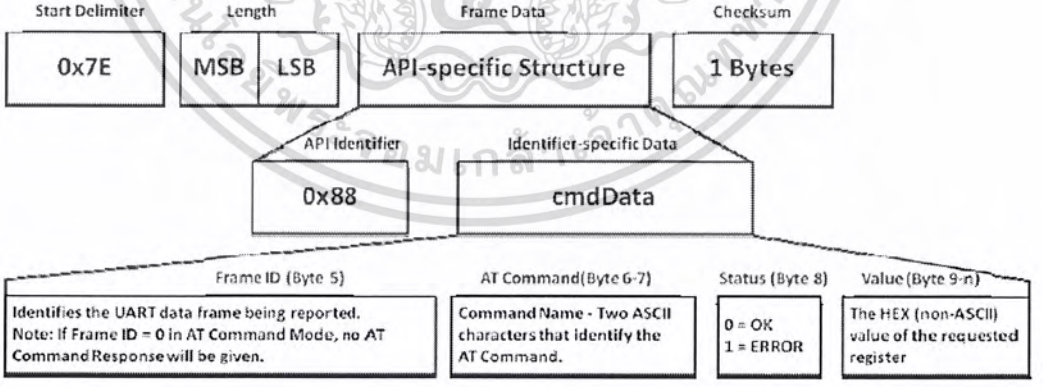
Parameter Value มีค่า API Identifier เป็น 0x09 โครงสร้างของ API frame ประเภทนี้เป็นดังรูปที่ 2.19



รูปที่ 2.19 รูปแบบของ API Packet frame ประเภท AT Command Queue Parameter Value

2.8.2.3 AT Command Response

API frame ประเภทนี้เป็นผลจากการใช้ API frame AT Command ซึ่งจะแสดงผลลัพธ์ของข้อความ AT Command ที่ส่งไป บางคำสั่งเช่น ND (Node Discovery) จะส่งผลลัพธ์กลับมาหลายๆframe ซึ่ง frame สุดท้ายจะบอกสถานะของการส่ง API frame AT Command Response มีค่า API Identifier เป็น 0x88 โครงสร้างของ API frame ประเภทนี้ซึ่งรูปแบบการทำงานของ API Packet Frame แสดงดังรูปที่ 2.20 เป็นดังนี้

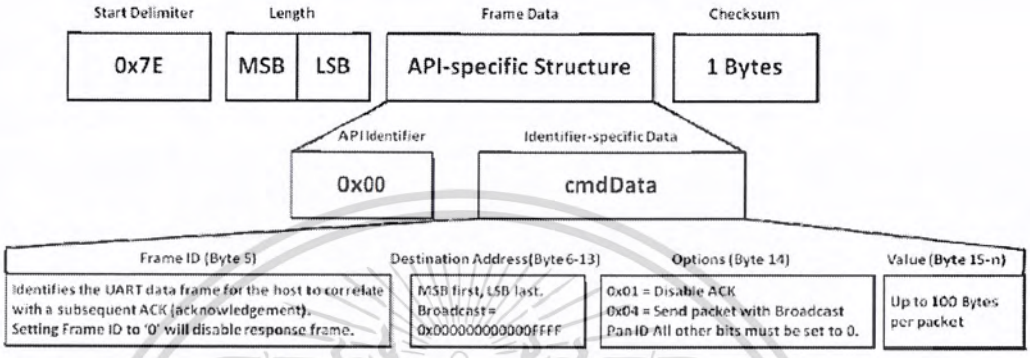


รูปที่ 2.20 รูปแบบของ API Packet frame ประเภท AT Command Response

2.8.2.4 Tx (Transmit) Request: 64-bit address

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

API frame ประเภทนี้จะส่งข้อมูลในรูปแบบของ packet ไปยังโมดูลปลายทาง โดยใช้ที่อยู่แบบ 64 bit และมี API Identifier เป็น 0x00 โครงสร้างของ API frame ประเภทนี้ แสดงโครงสร้างดังรูปที่ 2.21 เป็นดังนี้



รูปที่ 2.21 รูปแบบของ API Packet frame ประเภท Tx (Transmit) Request : 64-bit address

Destination Address คือ ที่อยู่ของโมดูลปลายทาง เมื่อถูกตั้งค่าเป็น 0x000000000000FFFF ข้อมูลจะถูกส่งแบบ broadcast

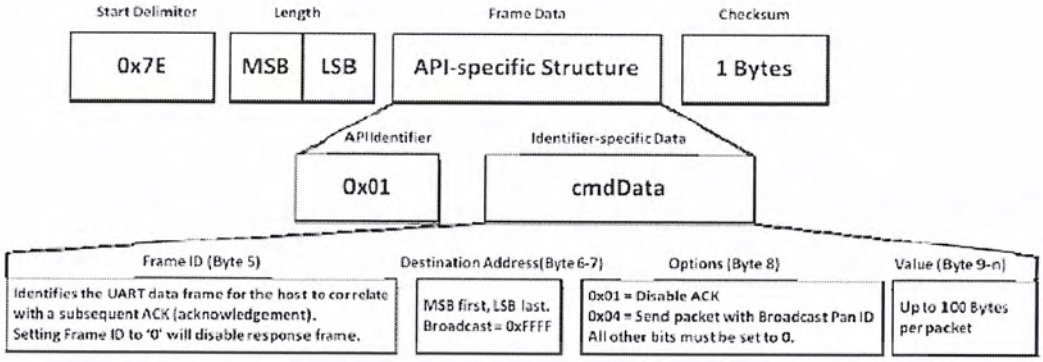
Option Byte คือ ตัวเลือกของการส่ง เมื่อถูกตั้งค่าเป็น 0x01 จะยกเลิกการส่ง ACK เมื่อตั้งค่าเป็น 0x04 จะส่ง packet ไปยังทุกโมดูลที่อยู่ Pan ID เดียวกัน

RF Data คือ ข้อมูลที่ต้องการส่ง

2.8.2.5 Tx (Transmit) Request : 16-bit address

API frame ประเภทนี้จะส่งข้อมูลในรูปแบบของ packet ไปยังโมดูลปลายทาง โดยใช้ที่อยู่แบบ 16 bit และมี API Identifier เป็น 0x01 โครงสร้างของ API frame ประเภทนี้มีรูปแบบของ Packet ตามรูปที่ 2.22 เป็นดังนี้

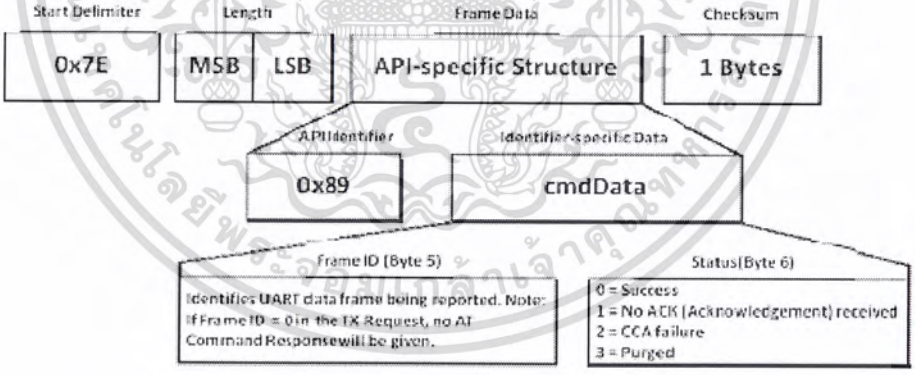
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.22 รูปแบบของ API Packet frame ประเภท Tx (Transmit) Request : 16-bit address

2.8.2.6 Tx (Transmit) Status

เมื่อการส่งข้อมูลเสร็จสมบูรณ์ โมดูลที่ได้รับข้อมูลจะส่งสถานะของการส่งกลับมายังโมดูลที่ส่งข้อมูล API frame ประเภทนี้เป็นสถานะของการส่งข้อมูลว่าสำเร็จหรือไม่ มี API Identifier เป็น0x89โครงสร้างของ API frame ประเภทนี้ เป็นดังรูปที่ 2.23

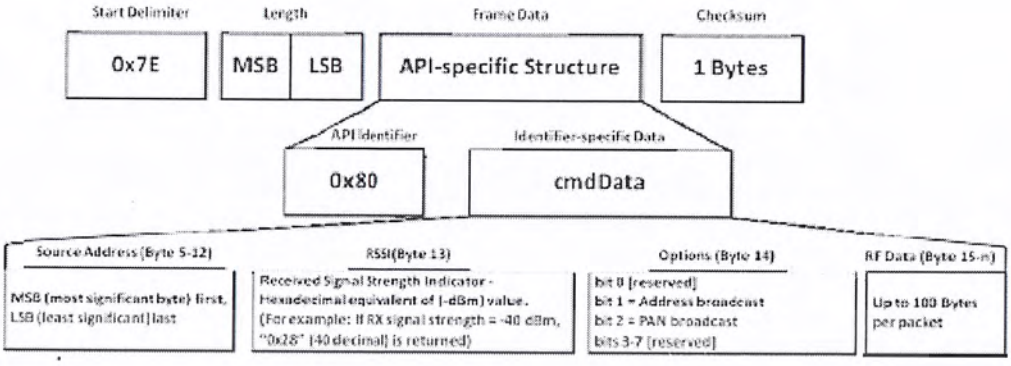


รูปที่ 2.23 รูปแบบของ API Packet frame ประเภท Tx (Transmit)

2.8.2.7 RX (Receive) Packet 64-bit address

API frame ประเภทนี้มี API Identifier เป็น 0x80 เมื่อโมดูลได้รับข้อมูลจากการส่งแบบที่อยู่ 64 bitแล้ว frame packet ที่ได้รับจะมีโครงสร้างดังรูปที่ 2.24

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.24 รูปแบบของ API Packet frame ประเภท RX (Receive) Packet 64-bit address

Source Address คือ ที่อยู่ของโมดูลที่ส่งข้อมูลมายังตัวรับ

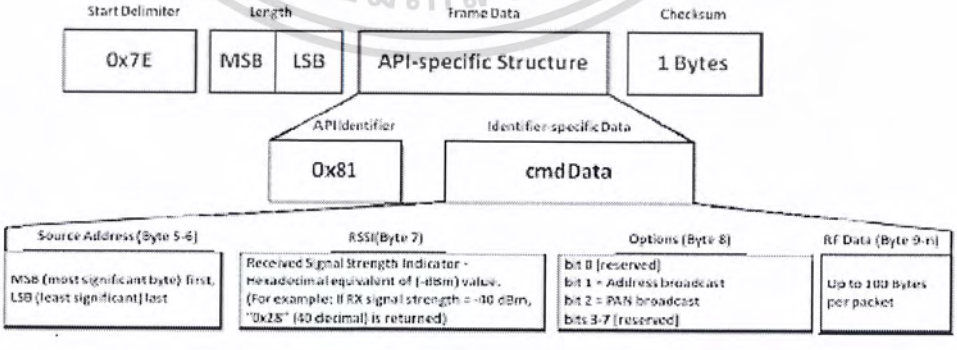
RSSI คือ ความแรงของสัญญาณ

Option Byte คือ รูปแบบของการส่ง ถ้าเป็น 0x01 จะเป็นการส่งแบบหนึ่งต่อหนึ่ง ถ้าเป็น 0x02 จะเป็นการส่งแบบ PAN broadcast

RF Data คือ ข้อมูลที่ได้รับ

2.8.2.8 RX (Receive) Packet 16-bit address

API frame ประเภทนี้มี API Identifier เป็น 0x81 เมื่อโมดูลได้รับข้อมูลจากการส่งแบบที่อยู่ 16 bit แล้ว frame packet ที่ได้รับจะมีโครงสร้างดังรูปที่ 2.25



รูปที่ 2.25 รูปแบบของ API Packet frame ประเภท RX (Receive) Packet 16-bit address

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.8.3 ตัวอย่างการใช้ API frame packet

การใช้งานในโหมดนี้จะต้องตั้งค่าพารามิเตอร์ AP ให้เป็น 1 ก่อนจึงจะสามารถใช้งานโหมดนี้ได้ และข้อมูลที่อยู่ใน API frame packet จะอยู่ในรูปแบบของเลขฐานสิบหก

2.8.3.1 การส่งและรับคำสั่ง ATND จาก ZigBee

เป็นคำสั่งที่ใช้สำรวจและแสดงรายละเอียดของทุกๆโนดที่อยู่ในระยะการส่งข้อมูล โดยสิ่งที่เราต้องการจากคำสั่ง ATND คือค่าความแรงสัญญาณของแต่ละโนดที่อยู่ในรัศมี ในการส่งคำสั่ง ATND ไปยัง Zigbee เราใช้ API Modeในการส่งคำสั่งซึ่งมีลักษณะดังตารางที่ 2.4

ตาราง 2.4 คำสั่งของพารามิเตอร์ API Mode

Byte	0	1	2	3	4	5	6	7
Value	7E	00	04	08	52	4E	44	13

เมื่อ API frame packet ดังกล่าวถูกส่งไปยัง ZigBee แล้ว ZigBee จะตอบกลับมาในรูปแบบของAPI frame packet ดังตารางที่ 2.5 และ 2.6

Byte ที่ 9-10 คือ MY (Source Address)

Byte ที่ 11-14 คือ SH (Serial Number High)

Byte ที่ 15-18 คือ SL (Serial Number Low)

Byte ที่ 19 คือ DB (Receive Signal Strength)

Byte ที่ 20 ขึ้นไปคือ NI (Node Identifier)

ตาราง 2.5 API frame packet ที่ได้รับทั้งแต่บิตที่ 0 – 10

Byte	0	1	2	3	4	5	6	7	8	9	10
Value	7E	00	12	88	52	4E	44	00	FF	FF	00

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตาราง 2.6 API frame packet ที่ได้รับทั้งแต่บิตที่ 11 – 21

Byte	11	12	13	14	15	16	17	18	19	20	21
Value	13	A2	00	40	01	3B	80	29	20	00	09B

และ ACK ของคำสั่ง ATND อีก 9 Byte ตารางที่ 2.7

ตาราง 2.7 ACK ของคำสั่ง ATND จำนวน 9 Byte

Byte	0	1	2	3	4	5	6	7	8
Value	7E	00	05	88	52	4E	44	00	93

2.8.3.2 การส่งข้อมูลค่าความแรงไปยัง ZigBee ที่ Monitor Node

เมื่อได้ค่าความแรงจากการใช้คำสั่ง ATND แล้ว หลังจากนั้นไมโครคอนโทรลเลอร์จะเก็บค่าความแรงไว้ในตัวแปร และสร้าง API frame packet กำหนดที่อยู่ปลายทางเป็น ZigBee ที่ Monitor Node ไมโครคอนโทรลเลอร์ดังกล่าวจะส่ง API frame Packet ที่ได้สร้างไว้ให้กับ ZigBee ตัวที่ติดอยู่กับวัตถุเพื่อส่งข้อมูลไปยัง ZigBee ที่ Monitor โหนด ตัวอย่างของ API frame packet ที่ส่งเป็นดังตารางที่ 2.8

Byte ที่ 6-13 คือ ที่อยู่ปลายทาง 64 bit

Byte ที่ 15-22 คือ ที่อยู่ของโนดที่ได้จากคำสั่ง ATND

Byte ที่ 23 คือ ค่าความแรงสัญญาณ

ตารางที่ 2.8 API frame ที่ได้ทำการส่งไปให้ Monitor Node

Byte	0	1	2	3	4	5	6	7	8
Value	7E	00	12	00	00	00	13	A2	00
Byte	0	1	2	3	4	5	6	7	8
value	01	00	13	A2	00	40	01	3B	80

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3

การออกแบบและการจัดทำปริญญานิพนธ์

3.1 การออกแบบ

3.1.1 โครงสร้างของระบบ

ระบบระบุตำแหน่งวัตถุที่ได้ออกแบบไว้มี 3 ส่วนใหญ่ๆ ดังนี้

1) Reference Node คือ โหนดที่ใช้เป็นจุดอ้างอิงถึงความแรงของสัญญาณ โดยโหนดนี้ต้องส่งค่าความแรงไปยัง Object Node ซึ่งต้องติดตั้งตามจุดต่างๆ 3 จุด
2) Object Node คือ โหนดที่อยู่ติดกับวัตถุ โหนดนี้ทำหน้าที่ร้องขอค่าความแรงของสัญญาณจาก Reference Node และส่งค่าความแรงดังกล่าวไปยัง Monitor Node ที่ติดกับคอมพิวเตอร์
3) Monitor Node คือโหนดที่ใช้ในการคำนวณค่าความแรงของสัญญาณเป็นระยะทางและแสดงผลตำแหน่งของวัตถุโดยโปรแกรมแสดงผลผ่านทางคอมพิวเตอร์

3.1.2 การทำงานของระบบ

3.1.2.1 วงจร Reference Node

วงจรนี้ทำหน้าที่เป็นจุดอ้างอิงถึงความแรงของสัญญาณกับวัตถุ ซึ่งจะต้องติดตั้งตามจุดต่างๆเป็นจำนวน 3 จุด เมื่อ Object Node มีการใช้คำสั่ง AT Command เพื่อร้องขอค่าความแรงของสัญญาณ จากนั้น Reference Node จะส่งสัญญาณ Beacon กลับไปให้กับ Object Node วงจรนี้ประกอบไปด้วย ตัว Zigbee Pro และต้องป้อนแรงดันกระแสตรงขนาด 5 โวลต์เข้าทางขา 3 ของ IC LM1117T3.3 เพื่อแปลงแรงดันจาก 5 โวลต์เป็น 3.3 โวลต์ออกที่ขา 2 แล้วจ่ายไฟต่อให้ ZigBee

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ทำงาน หลอดไฟ LED สีเขียวจะติดสว่างเมื่อมีแรงดัน 3.3 โวลต์เข้ามา หลอดไฟ LED สีแดงเป็นการบอกสถานะ RSS ของ ZigBee จะกระพริบทุกๆ 1 วินาที เมื่อ ZigBee ทำงาน ดังรูปที่ 3.1

3.1.2.2 วงจร Object Node

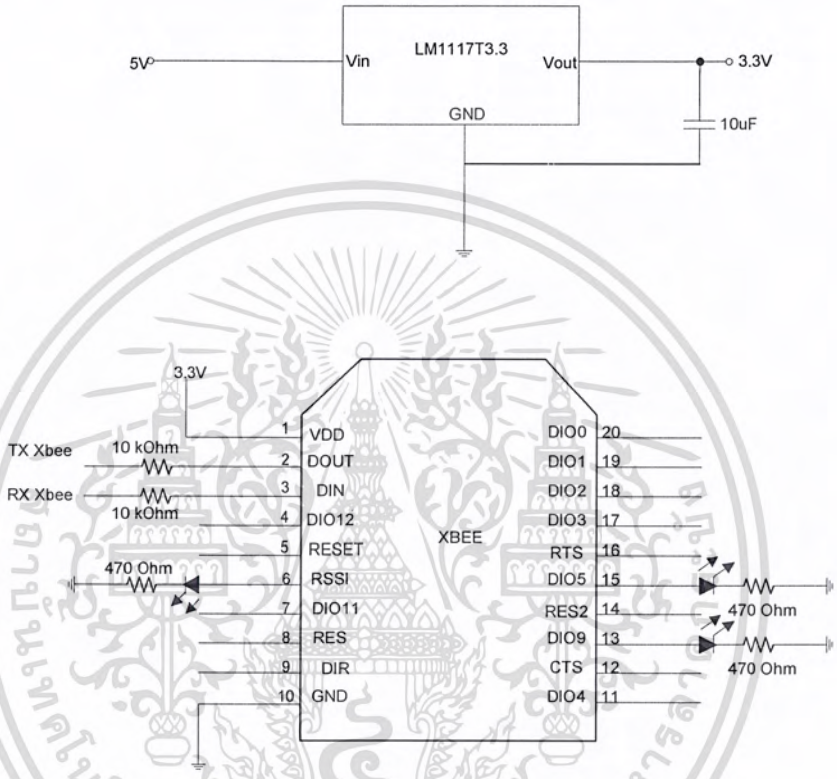
วงจรนี้ประกอบไปด้วยไมโครคอนโทรลเลอร์และ Zigbee โดยไมโครคอนโทรลเลอร์มีหน้าที่ควบคุมการทำงานของ ZigBee ที่ติดอยู่กับวัตถุ โดยหาค่าความแรงของสัญญาณเทียบกับแต่ละ Reference Node และเก็บค่าความแรงดังกล่าวไว้ ก่อนที่จะส่งค่าความแรงที่บันทึกไว้ไปยัง Monitor Node ที่ติดอยู่กับคอมพิวเตอร์ วงจรนี้ต้องป้อนแรงดันกระแสตรงขนาด 9 โวลต์ จากนั้น IC LM7805 จะลดขนาดแรงดันลงเหลือ 5 โวลต์ เพื่อให้สามารถป้อนให้กับไมโครคอนโทรลเลอร์ได้ และ IC LM1117T3.3 จะลดแรงดันจาก 5โวลต์ เหลือ 3.3 โวลต์ เพื่อให้สามารถป้อนไฟเลี้ยงต่อให้ ZigBee ได้ ขา output ของไมโครคอนโทรลเลอร์จะต่อกับขา Input ของ ZigBee และขา output ของ ZigBee จะต่อกับขา Input ของไมโครคอนโทรลเลอร์ โดยหลักการทำงานนั้นตัวไมโครคอนโทรลเลอร์จะทำการส่งคำสั่งให้ ZigBee ที่ติดกับตัว Object Node นั้นส่งคำสั่ง AT Command (ATND) เพื่อทำการร้องขอค่าพารามิเตอร์ต่างๆของ Reference Node ที่อยู่โดยรอบ และทำการปิดคำสั่งโดย AT Command (ATCN) เมื่อได้ค่าพารามิเตอร์ที่ต้องการแล้ว จะทำการเก็บที่ได้ไว้ในหน่วยความจำ ก่อนที่จะส่งค่าที่บันทึกไว้ไปยัง Monitor Node ดังรูปที่ 3.2

3.1.2.3 วงจร Monitor Node

วงจรนี้ประกอบไปด้วยไมโครคอนโทรลเลอร์และ Zigbee โดยการทำงานของวงจรมีหน้าที่รับค่าความแรงที่ส่งมาจาก Object Node แล้วทำการนำค่าความแรงของสัญญาณที่ได้นั้นมาทำการแปลงให้เป็นระยะทาง จากนั้นทำการส่งค่าระยะทางที่ได้ทำการแปลงข้อมูลแล้วเข้าคอมพิวเตอร์ผ่านทางพอร์ตอนุกรม เพื่อมาคำนวณหาตำแหน่งของวัตถุผ่านโปรแกรมที่ได้พัฒนาการติดต่อระหว่าง ZigBee กับคอมพิวเตอร์ผ่านทางพอร์ตอนุกรม RS-232 โดยจะต้องมี IC MAX232 เพื่อปรับแรงดันให้เป็นแบบ TTL ได้ ขาส่งข้อมูลของ ZigBee จะต่อกับขารับข้อมูลของคอมพิวเตอร์ และขารับข้อมูลของ ZigBee จะต่อกับขาส่งข้อมูลของคอมพิวเตอร์ วงจรนี้ต้องการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

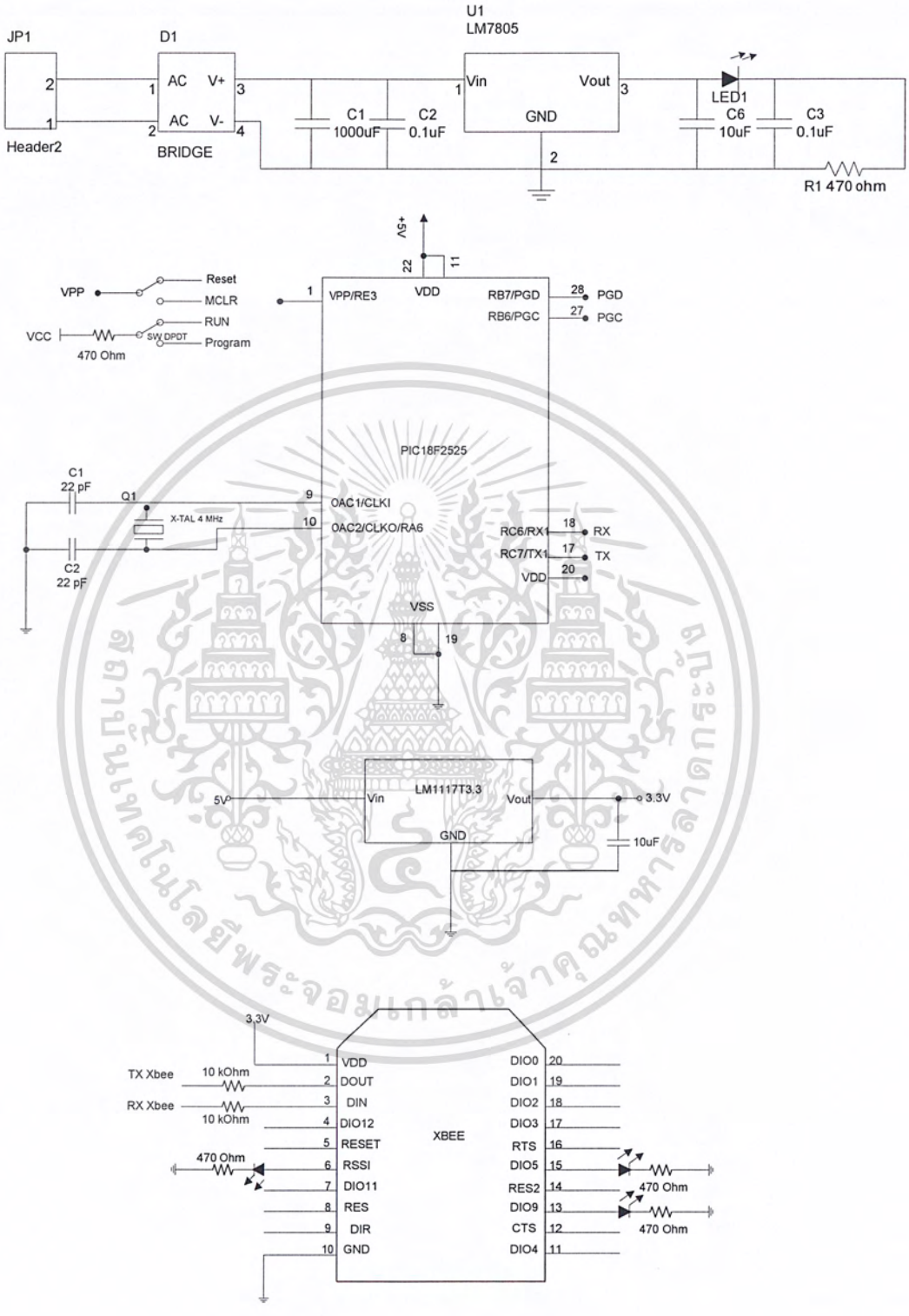
แรงดัน 3.3 โวลต์ป้อนเข้าที่ขา VCC และ GND ของ ZigBee ไฟ LED สีเขียวจะติดสว่างเมื่อแรงดันเข้ามา 3.3 โวลต์ ไฟ LED สีแดงเป็นการบอกสถานะ RSS ของ ZigBee จะกระพริบทุกๆ 1 วินาทีเมื่อ ZigBee ทำงาน ดังรูปที่ 3.3 และ 3.4



รูปที่ 3.1 วงจร Reference Node

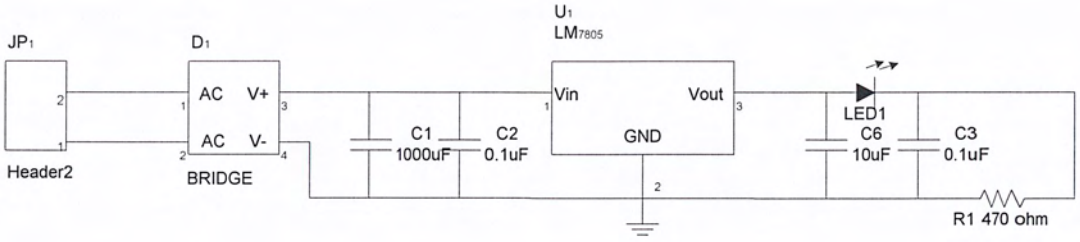
สรุปการทำงานของวงจรทั้งสามวงจร เริ่มจากวงจร Object Node ร้องขอค่าความแรงของสัญญาณจากวงจร Reference Node ทั้งสามโหนดของ Reference Node จะส่งสัญญาณกลับไปทำให้ Object Node ได้ค่าความแรงของสัญญาณ ไมโครคอนโทรลเลอร์เก็บค่าความแรงดังกล่าวในรูปแบบ API Frame Packet เพื่อส่งค่าความแรงไปยัง Monitor Node แล้ว Monitor Node จะทำการแปลงค่าความแรงของสัญญาณเป็นค่าระยะทาง และส่งค่าที่ได้นำไปคำนวณหาตำแหน่งของวัตถุผ่านทางโปรแกรมแสดงผลต่อไป ดังรูปที่ 3.5

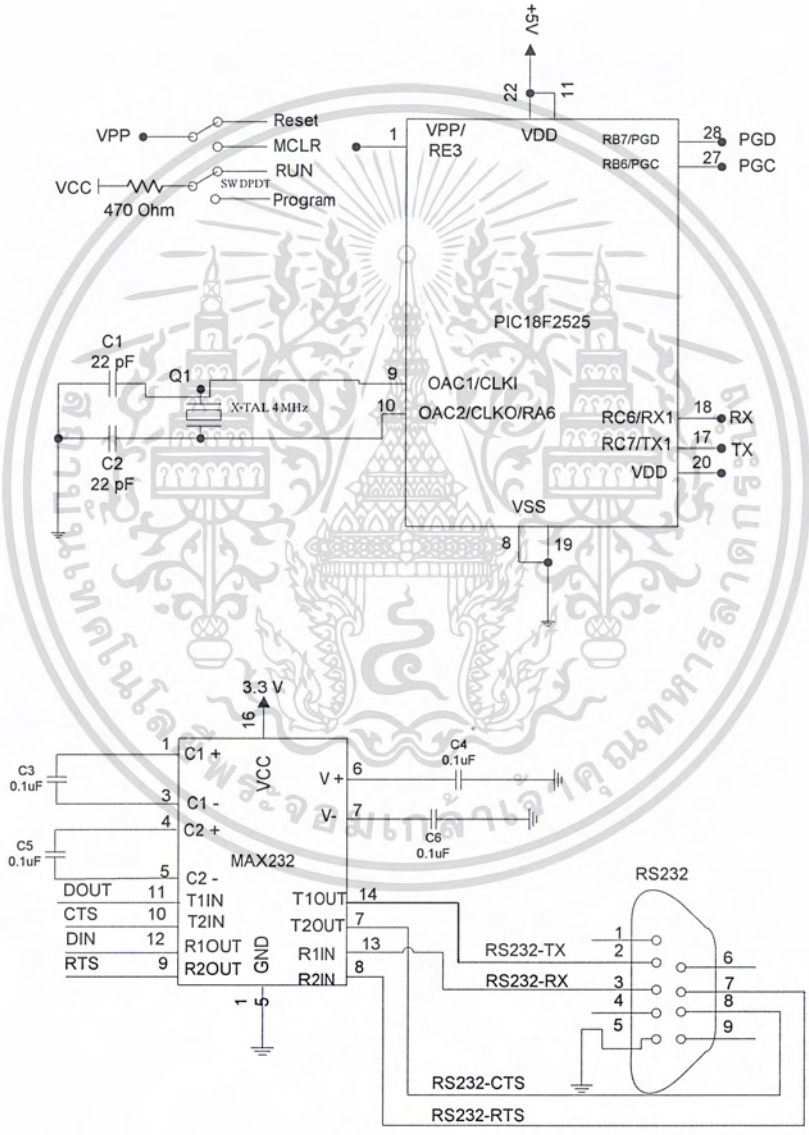
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.2 วงจร Object Node

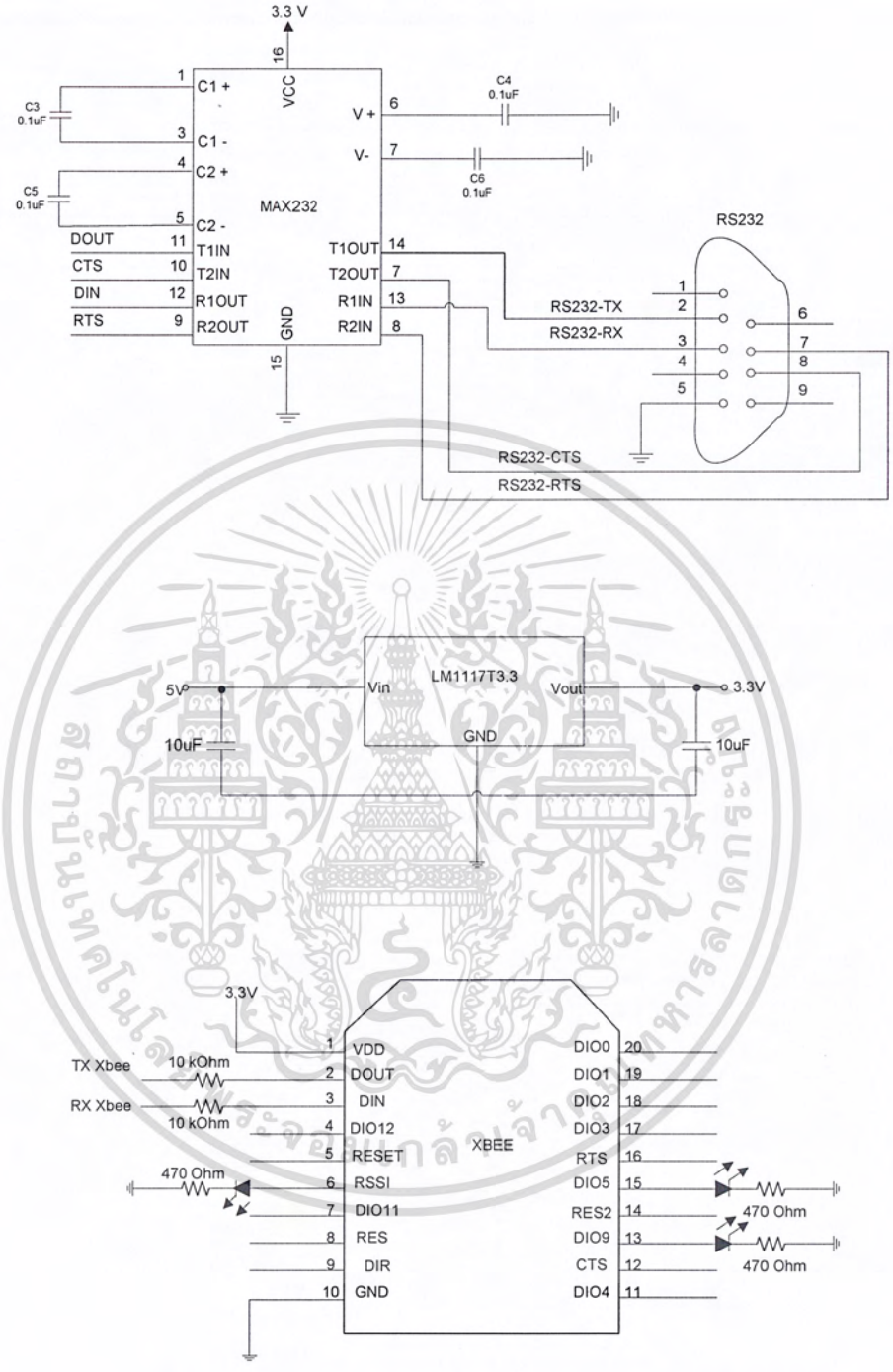
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้





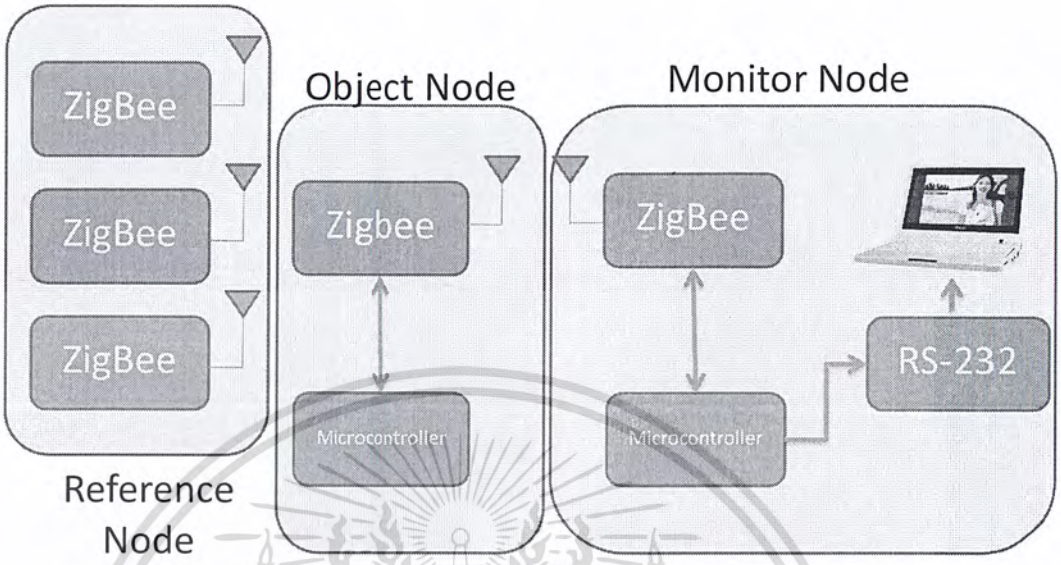
รูปที่ 3.3 วงจร MAX 232 ติดต่อกับไมโครคอนโทรเลอร์ที่ Monitor Node

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.4 วงจร MAX 232 ติดต่อกับ Zigbee ที่ Monitor Node

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.5 ความสัมพันธ์ส่วนต่างๆ ของระบบ

3.1.3 การทำงานของไมโครคอนโทรลเลอร์

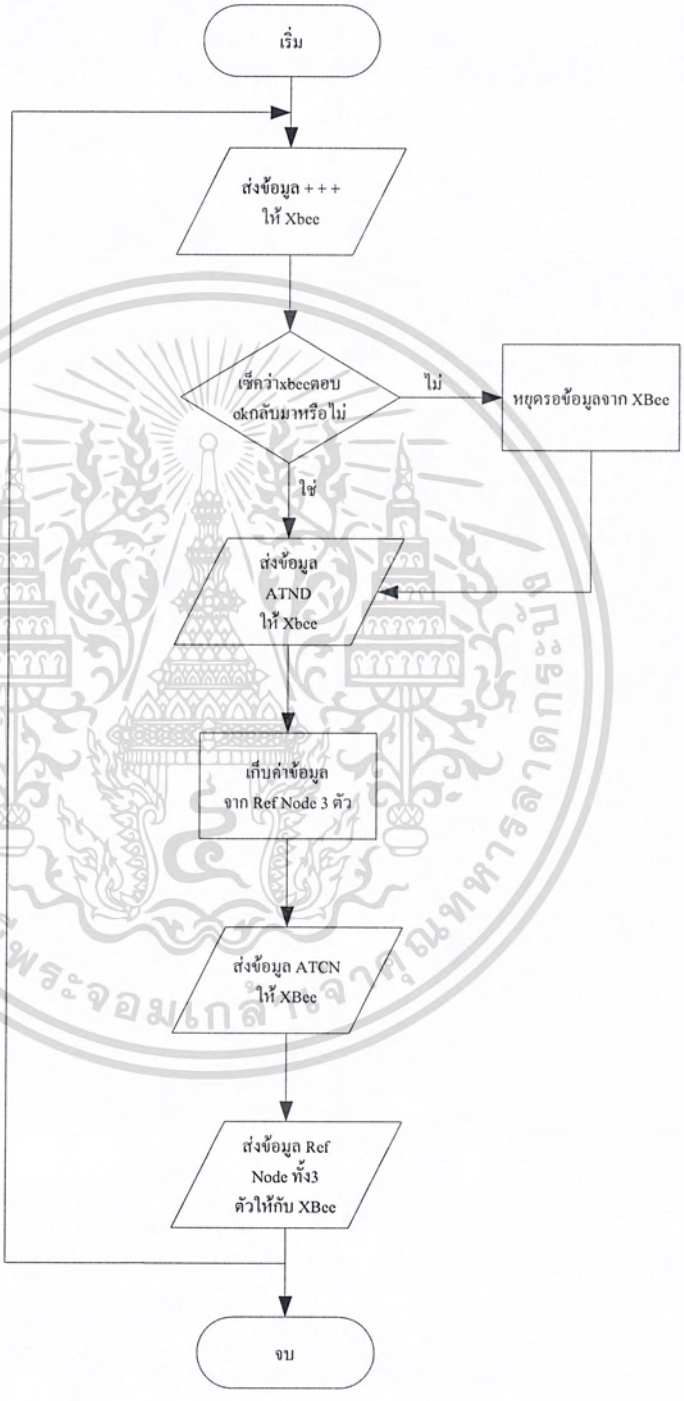
วัตถุที่ต้องการระบุตำแหน่งจะต้องมีอุปกรณ์ ZigBee และไมโครคอนโทรลเลอร์เพื่อใช้ในการรับค่าความแรงของสัญญาณของโหนดอ้างอิงต่างๆ ไมโครคอนโทรลเลอร์มีหน้าที่ควบคุมการทำงานของโมดูล ZigBee ตัวที่ติดอยู่กับวัตถุเพื่อร้องขอค่าความแรงของสัญญาณและส่งค่าความแรงดังกล่าวไปยัง Monitor Node ที่ติดกับคอมพิวเตอร์เพื่อคำนวณหาตำแหน่งของวัตถุ ไมโครคอนโทรลเลอร์ที่ใช้คือ PIC18F2525 และซอฟต์แวร์ที่ใช้ในการโหลดโปรแกรมลงบนไมโครคอนโทรลเลอร์คือ PICkit v2.0

3.1.3.1ขั้นตอนการทำงานที่ Object Node เป็นดังนี้

การทำงานของ Object Node จะเริ่มที่ตัวไมโครคอนโทรลเลอร์ทำการส่ง คำสั่ง " +++ " เพื่อทำการเช็คว่าตัว Xbee ฝั่ง Object Node นั้นพร้อมรับคำสั่งหรือไม่ ถ้าพร้อมจะทำการส่งคำว่า " OK " กลับมา หลังจากนั้นทำการส่งคำสั่ง " ATND " เพื่อทำการร้องขอค่าพารามิเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ต่างๆ จาก Reference Node แล้วทำการเก็บข้อมูลที่ได้ว่าในตัวไมโครคอนโทรลเลอร์ เพื่อรอทำการส่งไปยัง Monitor Node ต่อไป ดังแสดงในรูปที่ 3.6



รูปที่ 3.6 ขั้นตอนการทำงานของฝั่ง Object Node

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.3.2 ขั้นตอนการทำงานที่ Monitor Node เป็นดังนี้

การทำงานในฝั่งของ Monitor Node ตัวไมโครคอนโทรลเลอร์จะทำการเปิดอินเตอร์รัปไว้รับค่าจาก Xbee ที่ฝั่ง Monitor Node ตลอดเวลา แล้วทำการเช็คว่าข้อมูลที่ได้มาแต่ละตัวนั้นเป็น Reference Node ตัวที่ 1,2 และ 3 ใช่หรือไม่ ถ้าใช่ จะทำการเก็บค่าความแรงของสัญญาณ ซึ่งในขณะที่ส่งมาจะเป็นเลขฐาน 16 ซึ่ง ไมโครคอนโทรลเลอร์จะทำการแปลงค่าที่ได้ให้อยู่ในรูปฐานสิบ แล้วทำการแปลงค่าความแรงของสัญญาณเป็นค่าระยะทาง ก่อนทำการส่งค่าของระยะทางนั้น ผ่านทางพอร์ตอนุกรมเพื่อนำข้อมูลไปทำการประมวลผลต่อบนคอมพิวเตอร์ ดังแสดงในรูปที่ 3.7

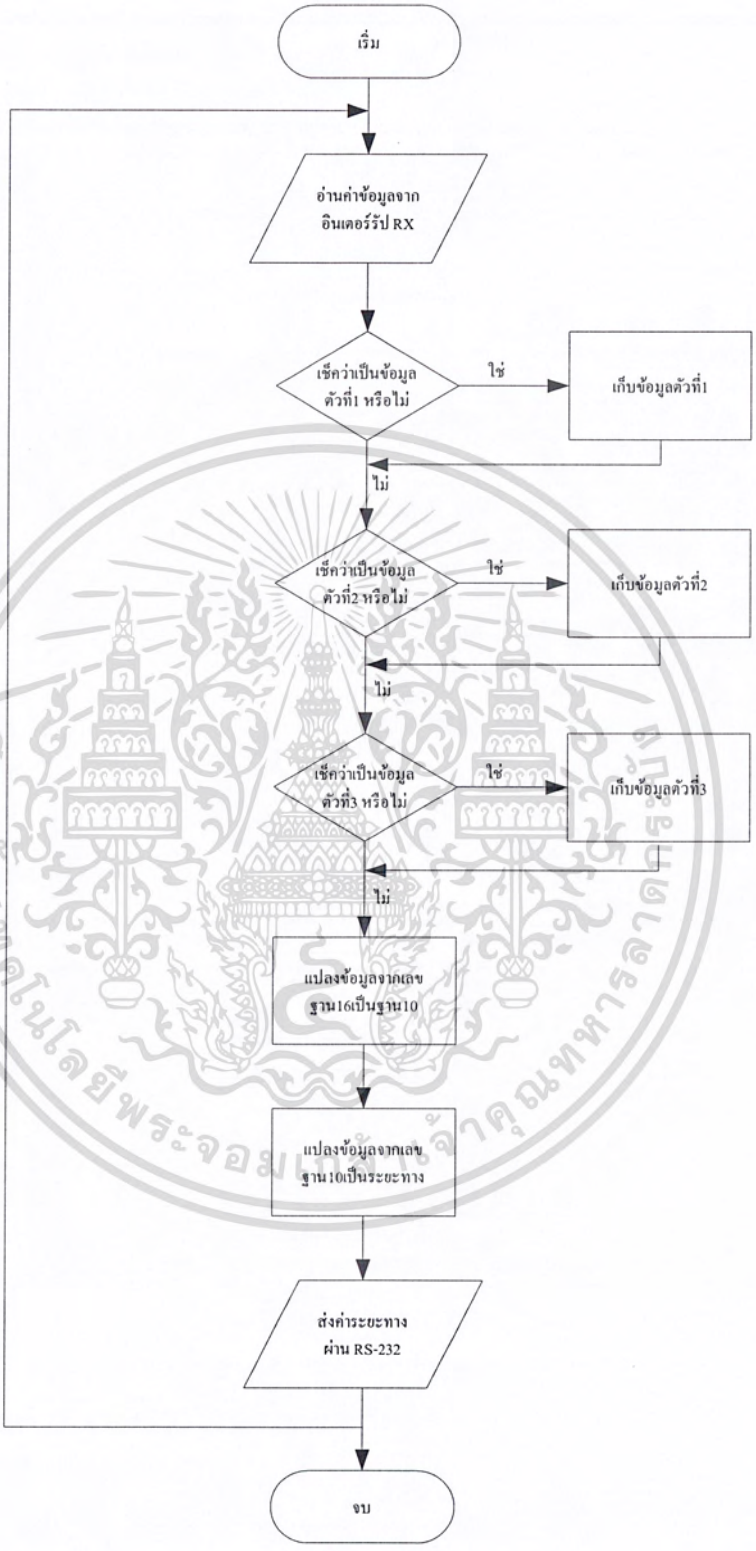
3.1.4 โปรแกรมแสดงผล

พัฒนาโปรแกรมโดยใช้ภาษาปาสคาล (Pascal) ติดต่อระหว่างคอมพิวเตอร์กับ ZigBee ผ่าน Serial Port โดยใช้โปรแกรม Delphi7 โดยโปรแกรมจะสามารถทำการรับค่าความแรงของสัญญาณจาก Monitor Node ตลอดเพื่อทำการคำนวณพิกัดของ Object Node หลังจากได้ค่าตำแหน่งวัตถุแล้วก็นำผลลัพธ์ที่ได้มาแสดงที่หน้าจอเพื่อให้ผู้ใช้ทราบตำแหน่งวัตถุเทียบกับจุดอ้างอิงได้ โดยจะแสดงในรูปพิกัด (x,y) ดังรูปที่ 3.8

3.1.4.1 ขั้นตอนการทำงานของโปรแกรมแสดงผล เป็นดังนี้

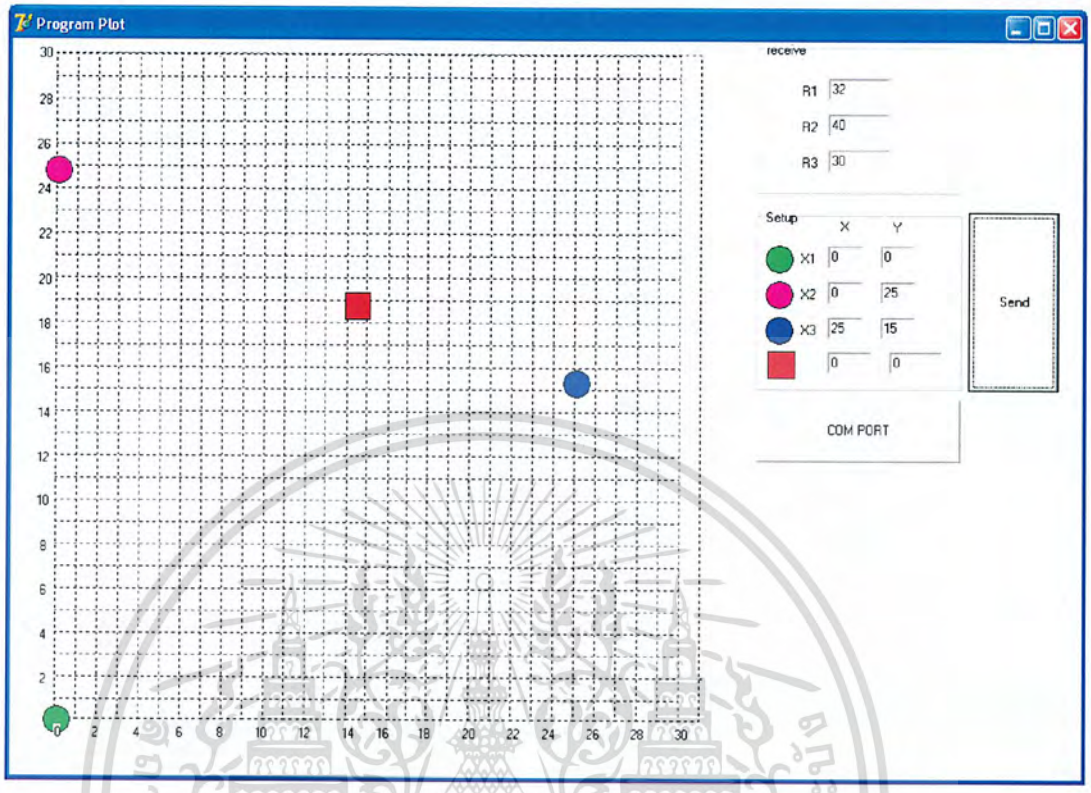
ผู้ใช้งานกรอกค่าพิกัดของ Reference Node ทั้ง 3 โหนด จากนั้นโปรแกรมก็จะทำการพล็อตค่าพิกัดของ Reference Node ทั้ง 3 โหนด และทำการเปิดพอร์ตการสื่อสาร เพื่อกำหนดค่าต่างๆ เช่น Baud Rate, Comport , Parity จากนั้นเช็คค่าที่ได้รับมาจาก RS-232 ว่าค่ามาครบทั้ง 3 โหนดหรือไม่ ค่าที่ได้นั้นเป็นระยะห่างระหว่าง Reference Node กับ Object Node แต่ละตัว ต่อไปก็จะสามารถคำนวณหาค่าพิกัด (x,y) ของวัตถุที่ต้องการระบุตำแหน่งได้แล้ว จากนั้นนำค่าพิกัดที่ได้ พล็อตลงบนโปรแกรมแสดงผล ดังรูปที่ 3.9

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.7 ขั้นตอนการทำงานของไมโครคอนโทรลเลอร์ฝั่ง Monitor Node

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

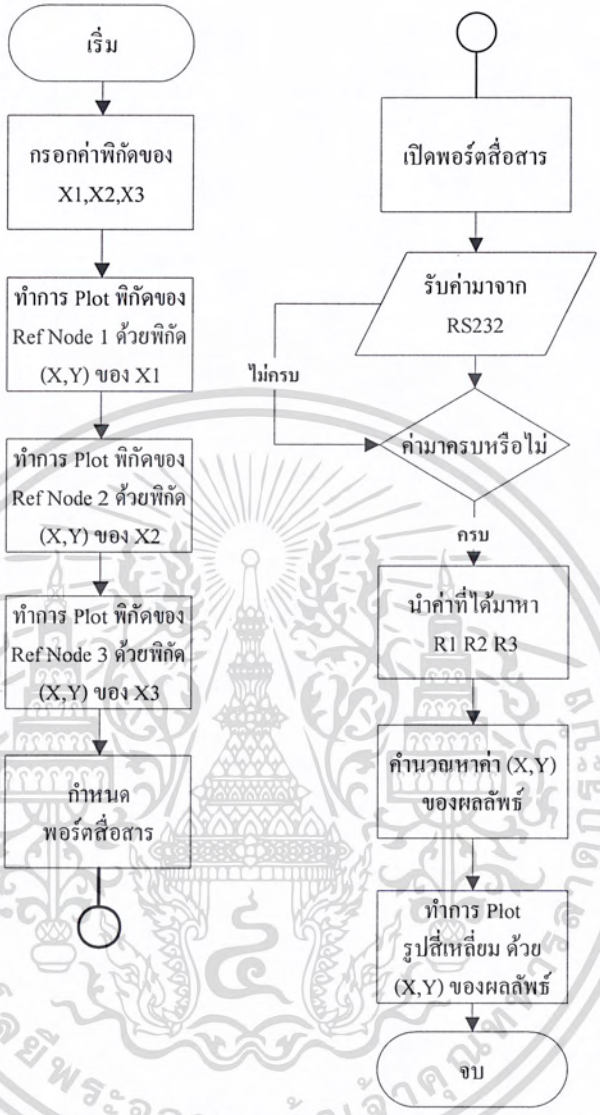


รูปที่ 3.8 หน้าต่างแสดงพิกัดและตำแหน่ง

3.4.1.2 การใช้งานหน้าต่างแสดงพิกัดและตำแหน่ง

1) ส่วนที่ใช้ในการแสดงผลว่าตัววัตถุอยู่ที่พิกัดใดในพื้นที่ที่ทำการทดลองโดย Reference Node จะแสดงเป็นจุดวงกลมสีต่างๆ ตามที่วางไว้ในพื้นที่
2) ส่วน Receive นั้นจะเป็นส่วนที่แสดงผลของการรับค่าระยะทางของ Reference Node กับ Object Node ที่ทำการรับมาจากพอร์ตอนุกรม
3) ส่วน Setup จะเป็นการปรับค่าพิกัดของ Reference Node ตามพิกัดต่างๆที่เราได้วางไว้ในพื้นที่ที่ใช้ทดลอง และในสองช่องล่างสุดของส่วน Setup จะเป็นค่าพิกัดของวัตถุที่ได้ทำการคำนวณเรียบร้อยแล้ว
4) Comport คือ ส่วนที่ได้ใช้ในการตั้งค่าพารามิเตอร์ของพอร์ตอนุกรมเพื่อใช้ในการรับค่าระยะทางจาก Monitor Node

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.9 ขั้นตอนการออกแบบของโปรแกรมแสดงผล

3.2 เครื่องมือที่ใช้ในการทดลอง

1) XBee 1mW Wire Antenna Series 1 จำนวน 2 ตัว
2) XBee Pro 60mW Wire Antenna Series 1 จำนวน 3 ตัว
3) ET-PIC STAMP 18F2525 จำนวน 2 ตัว
4) ET-PGM PIC USB จำนวน 1 ตัว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5) Oscilloscope จำนวน 1 เครื่อง

6) ตลับเมตร จำนวน 1 ตัว

3.3 การจัดเก็บผลการทดลอง

1) ก่อนจะทำการทดลองนั้นต้องทำการ Calibrate ความแรงของสัญญาณจากตัว Reference Node มาที่ตัว Object Node เพื่อทำการหาค่าเฉลี่ยก่อนที่จะทำการทดลอง เพื่อให้ได้ค่าความแรงเฉลี่ย ที่ระยะต่างๆ ก่อนที่จะนำค่าความแรงที่ได้ทำการบันทึกไปทำการหาค่า Path Loss Exponential

2) เมื่อทำการหาค่าเฉลี่ยของความแรงและค่า Path Loss Exponential นำค่าที่ได้นั้นไปคำนวณหาค่ากำลังของสัญญาณจากสมการ (2.3)

3) เมื่อได้ค่ากำลังของสัญญาณแล้วนำค่ากำลังของสัญญาณนั้น ไปคำนวณค่าระยะทางหรือค่ารัศมีของตัว Reference Node

4) นำค่าที่ได้นั้น ไปทำการเขียนคำสั่งลงบนไมโครคอนโทรลเลอร์ เพื่อทำการแปลงค่าความแรงของสัญญาณไปเป็นค่าของระยะทาง โดยไมโครคอนโทรลเลอร์จะแปลงค่าความแรงที่เป็นเลขฐาน 16 ให้เป็นเลขฐาน 10 ก่อน จากนั้นจะทำการแปลงเป็นค่าระยะทาง แล้วทำการการส่งค่าระยะทางของแต่ละตัว Reference Node เข้าให้กับคอมพิวเตอร์ผ่านทางพอร์ตอนุกรมบนฝั่งของคอมพิวเตอร์จะมีการเขียนโปรแกรมเพื่อรับค่าของระยะทางที่ได้จากไมโครคอนโทรลเลอร์ โดยโปรแกรมต้องทำการพล็อตพิกัดของ Reference Node ก่อนที่จะทำการคำนวณหาค่าพิกัด แล้วเมื่อทำการเปิดพอร์ตอนุกรมโปรแกรมจะทำการคำนวณหาค่าพิกัดของตัว Object Node ให้อัตโนมัติ และค่าระยะทางของตัว Reference Node จะทำการส่งค่ามาทุกๆ 5 วินาที

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

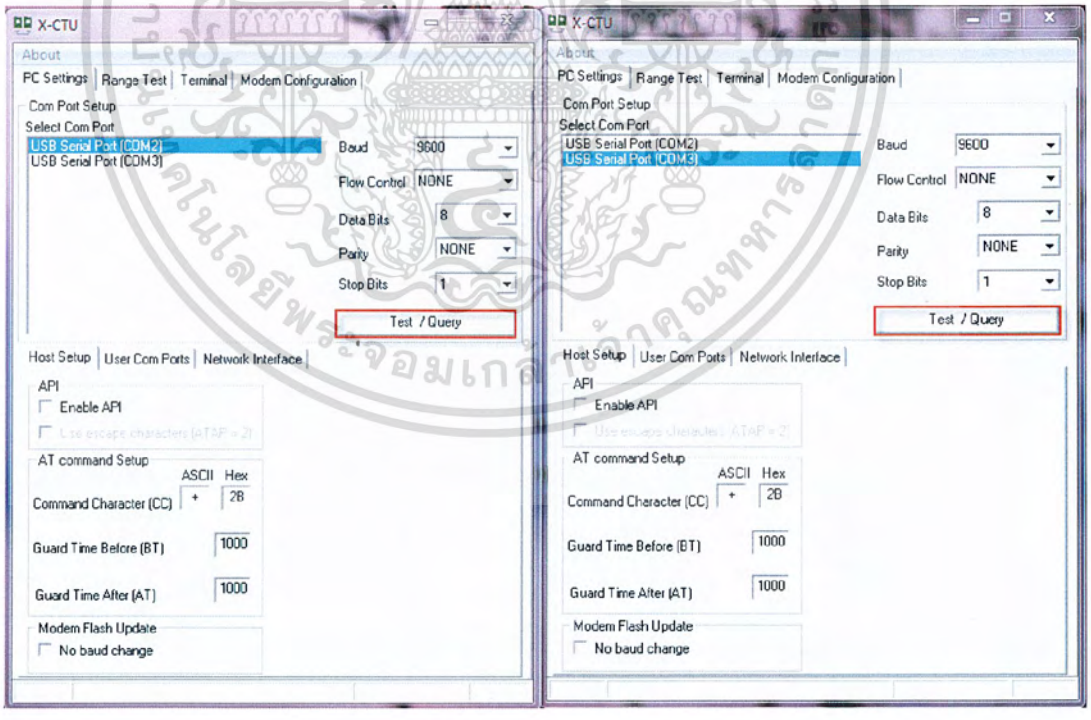
บทที่ 4

ผลการทดลอง

4.1 การตั้งค่าพารามิเตอร์สำหรับการทดลอง

4.1.1 การตั้งค่าพารามิเตอร์ในโปรแกรม X-CTU

ต่อ XBee ทั้งสองตัวเข้ากับคอมพิวเตอร์ผ่านทางSerial Port แล้วทำการเปิดโปรแกรม X-CTU ขึ้นมา 2 หน้าต่าง ทำการเลือกพอร์ตของ XBee แต่และตัว จากนั้นกดปุ่ม Test / Query เพื่อทดสอบว่าคอมพิวเตอร์นั้นสามารถติดต่อกับบอร์ดคอนฟิคของ XBee ได้ ดังรูปที่ 4.1



รูปที่ 4.1 PC Setting Tab

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อกด Test / Query จะขึ้นหน้าต่างที่แสดงสถานะว่าสามารถติดต่อกับ Xbee ได้ถูกต้อง ดังรูปที่ 4.2

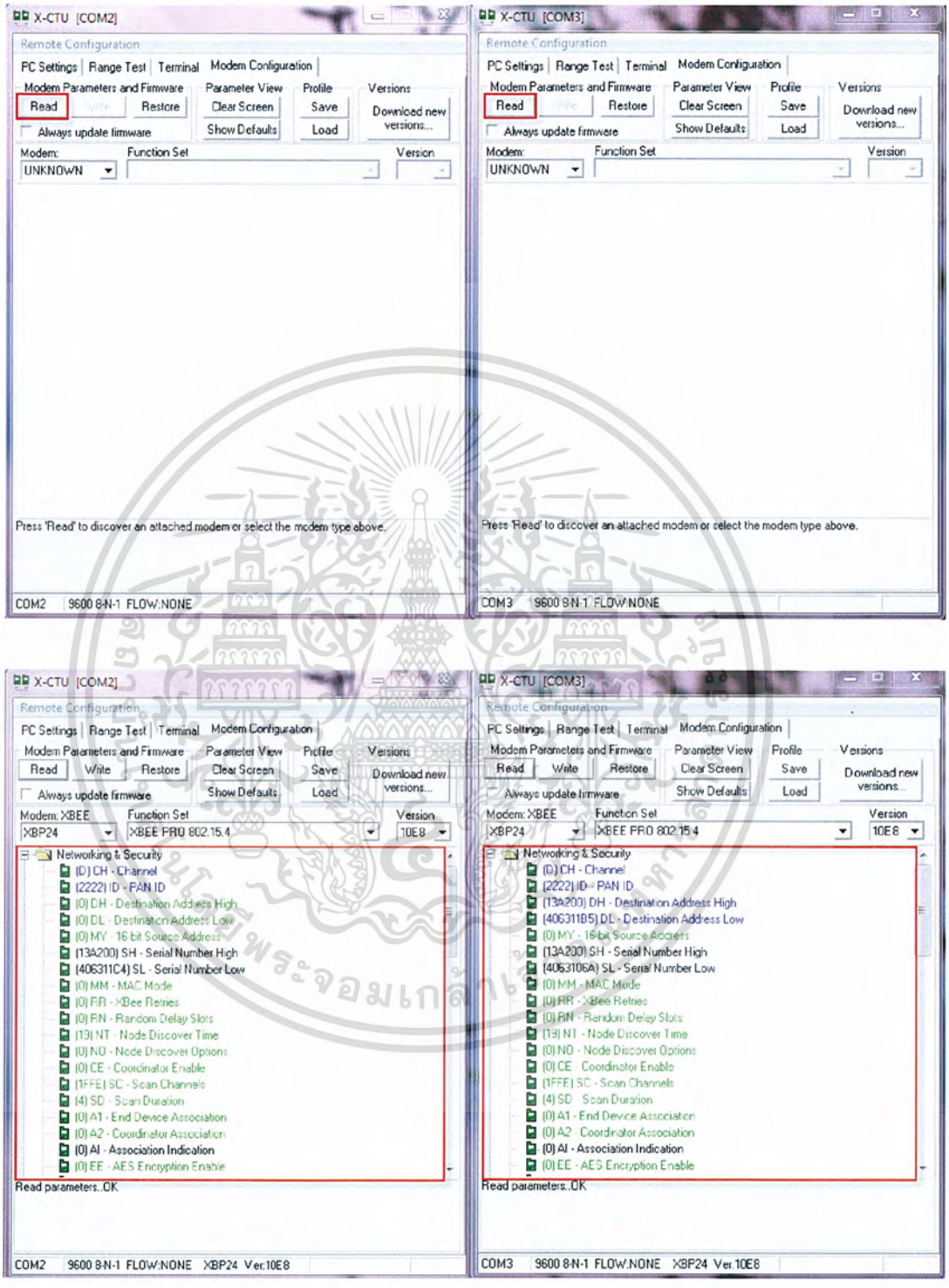
รูปที่ 4.2 หน้าต่าง Com test/Query Modem

แต่ถ้าคอมพิวเตอร์ไม่สามารถติดต่อกับ Xbee ได้จะขึ้นหน้าต่างดังรูปที่ 4.3

รูปที่ 4.3 หน้าต่าง Com test/ Query Modem ที่ไม่สามารถติดต่อกับ Xbee

ไปที่ Tab Modem Configuration แล้วกดปุ่ม Read เพื่อทำการอ่านค่าพารามิเตอร์ของ Xbee ที่จะทำการตั้งค่า ดังรูปที่ 4.4

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.4 Modem Configuration Tab

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ให้ทำการเซตค่าพารามิเตอร์ที่จะใช้ในการทดลอง การตั้งค่าพารามิเตอร์ให้แต่ละโหนดอยู่ในเครือข่ายเดียวกัน พารามิเตอร์ต่างๆ กำหนดเป็นดังนี้

CH (Chanel) คือ หมายเลขของช่องสัญญาณที่ใช้ส่งและรับระหว่าง RF Module ต้องตั้งค่าให้เป็นค่าเดียวกัน

ID (PAN ID) คือ Personal Area Network ID ต้องเป็นค่าเดียวกัน

AP (API Enable) ตั้งค่าเป็น 1 เพื่อเปิด API Mode

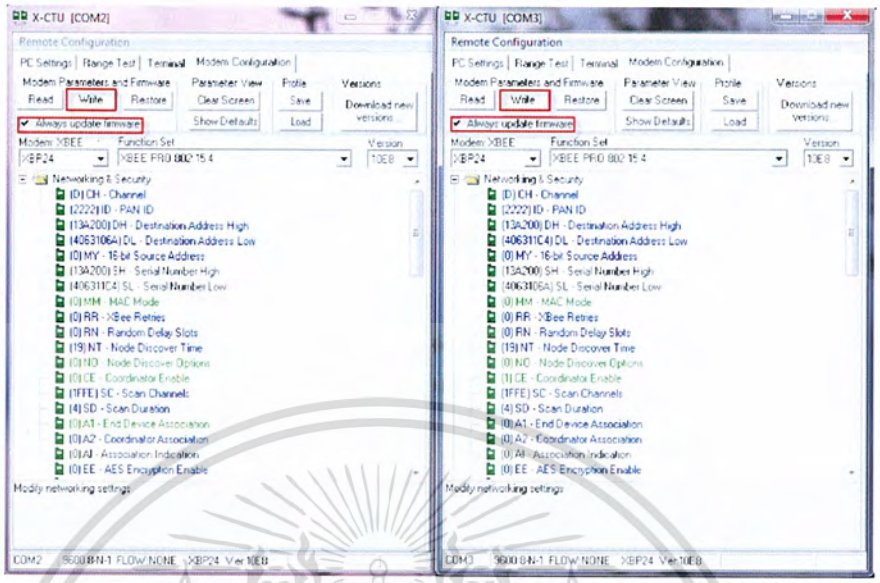
MY (16-bit source address) ตั้งค่าเป็น 0xFFFE เพื่อเปิดการใช้งานแบบที่อยู่ 64 bit

เมื่อทำการตั้งค่าเสร็จแล้ว ให้กด Write เพื่อเขียนค่าพารามิเตอร์ลงในตัว Xbee โดยค่าต่างๆ ที่ใส่เป็นไปตามรูปที่ 4.5 จากนั้นทำการกด Write เพื่อเขียนค่าพารามิเตอร์ต่างๆ ในตารางที่ 4.1 ลงในตัว XBee

ตารางที่ 4.1 การตั้งค่าพารามิเตอร์ ZigBee ผ่านทาง AT Command

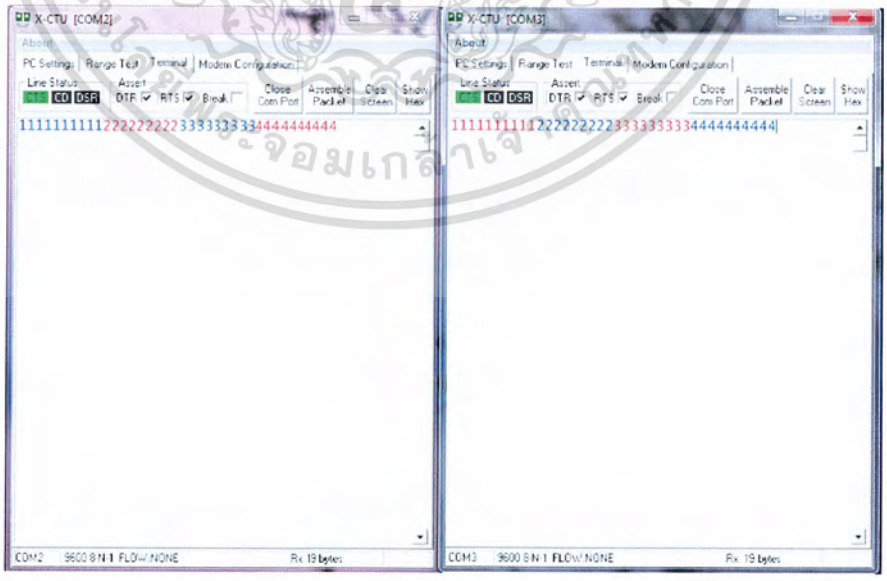
พารามิเตอร์	Reference Node1	Reference Node2	Reference Node3	Object Node	Monitor Node
CH (Chanel)	D	D	D	D	D
PAN ID (ID)	2222	2222	2222	2222	2222
AP (API Enable)	0	0	0	1	1
MY (16-bit source address)	0	0	0	FFFE	0

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.5 กด Write เพื่อเขียนค่าพารามิเตอร์ลงใน Xbee

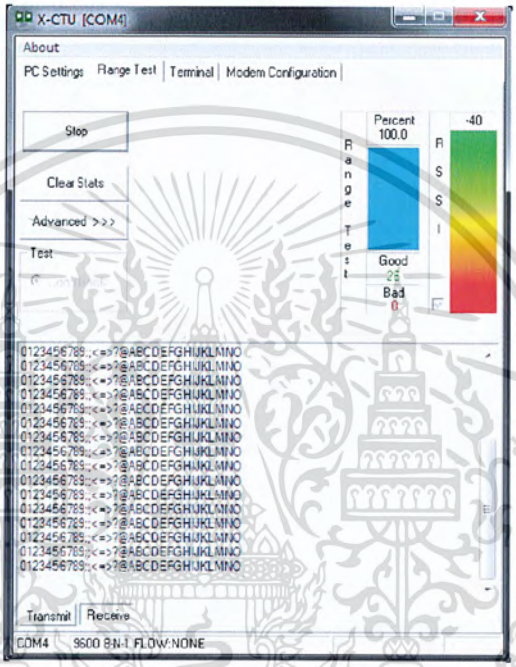
ทำการทดลองรับ – ส่งข้อมูลโดยไปที่ Tab Terminal แล้วทำการทำลองส่งข้อมูล โดยข้อมูลที่ทำการส่งคือหมายเลข 1 ซึ่งจะปรากฏบนหน้าโปรแกรม X-CTU ฝั่งส่ง และในขณะเดียวกัน ข้อมูลดังกล่าวก็แสดงบนหน้าโปรแกรม X-CTU ฝั่งรับ เป็น 1 เช่นเดียวกัน ดังแสดงในรูปที่ 4.6



รูปที่ 4.6 การรับ – ส่งข้อมูลระหว่าง Xbee

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ทำการทดสอบความถูกต้องในการรับค่า โดยตรวจสอบค่าความผิดพลาด โดยไปที่ Tab Range Test วิธีการทดสอบ ทำโดยการถอดสายพอร์ตอนุกรมทางฝั่งรับออก โดยทำการเชื่อมต่อระหว่างขา Rx และ Tx เข้าด้วยกัน และทำเครื่องหมายตรง RSSI แล้วทำการเลื่อนระยะของตัวฝั่งรับออกไป เพื่อหาระยะในการส่งที่ไกลที่สุดที่สามารถทำการติดต่อได้ แสดงดังรูปที่ 4.7

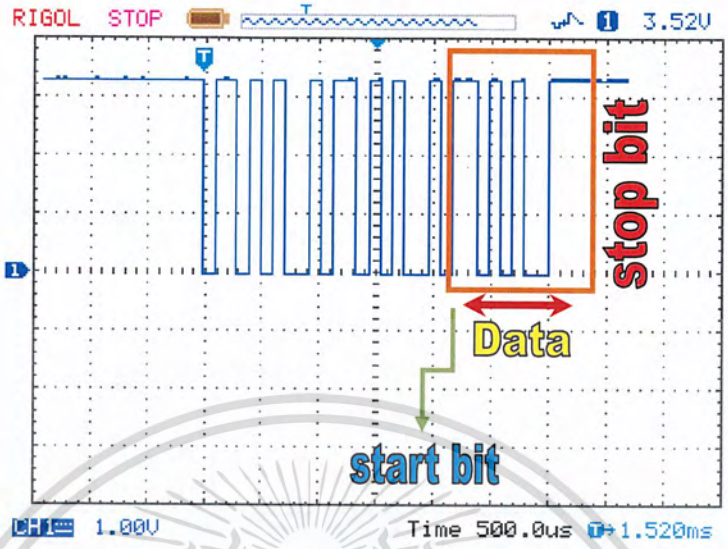


รูปที่ 4.7 Tab Range Test

4.2 ผลการทดลองวัดค่าแรงดันจากออสซิลโลสโคป

1) วัดค่าแรงดันที่ขา Tx ของตัว ZigBee ที่ตัว Object Node ซึ่งเราได้ทำการส่งอักขระ “+++” เพื่อเริ่มต้นเข้าสู่สถานะพร้อมรับคำสั่ง โดยผลที่ได้ออกมาจะอยู่ในรูปแบบของขบวนพัลส์ ซึ่งสามารถอ่านค่าได้โดยเปิดตารางแอสกี้โค๊ด รูปแบบของขบวนพัลส์จาก Oscillator ในการส่งอักขระ “+++” แสดงดังรูปที่ 4.8

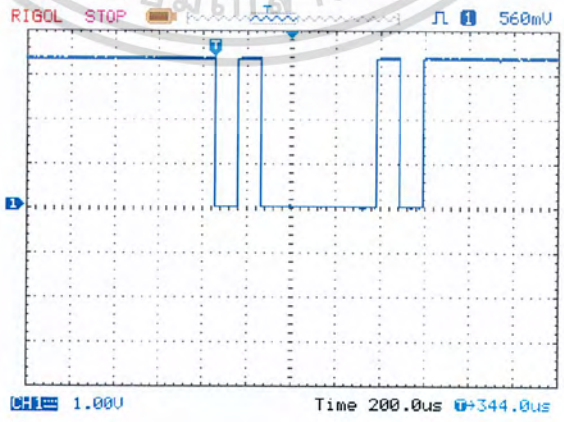
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.8 วัดขา Tx Object Node ซึ่งส่งอักขระ “ + + + “

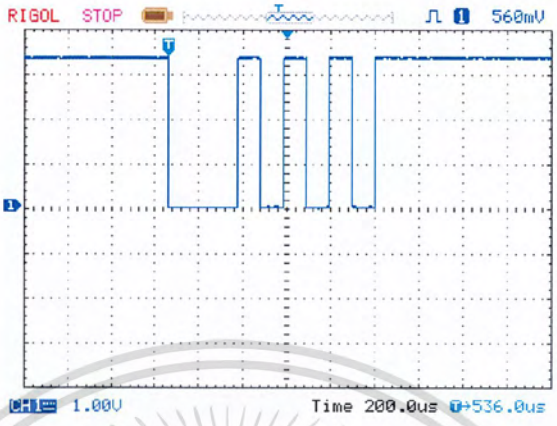
ซึ่งค่าจากกราฟจะอ่านจากข้างหลังไปหน้าได้เป็น 1 0010 1011 0 โดยบิตแรกเป็น stop bit และ 8 บิตตรงกลางเป็นข้อมูลที่ส่งไป คือ อักขระ “ + ” และบิตสุดท้ายคือ start bit

2) ทำการส่งค่า “ ATND ” จากตัว Object Node แล้วนำ Oscillator ไปวัดที่ขา Tx โดยทำการจับภาพหน้าจอทีละ 1 แล้วนำไปตรวจผลที่ได้จากตารางแอสกี้โค๊ด และทำการส่งข้อมูลตัวอักษร A, T, Nและ D และวัดสัญญาณออกมา แสดงดังรูปที่ 4.9, 4.10,4.11 และ 4.12 ตามลำดับ

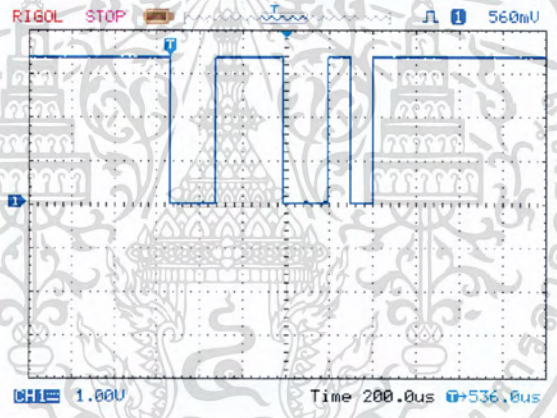


รูปที่ 4.9 วัดขา Tx Object Node ซึ่งส่งอักษร “A”

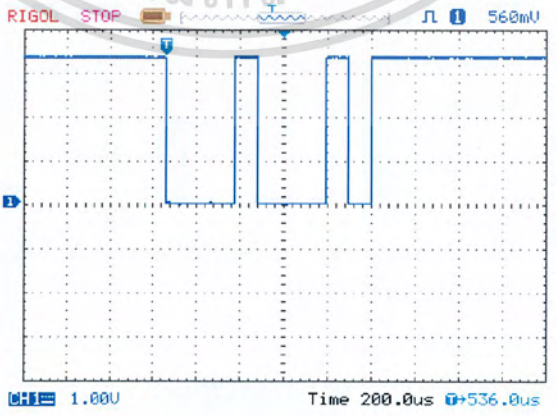
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.10 วัดขา Tx Object Node ซึ่งส่งอักษร “T”



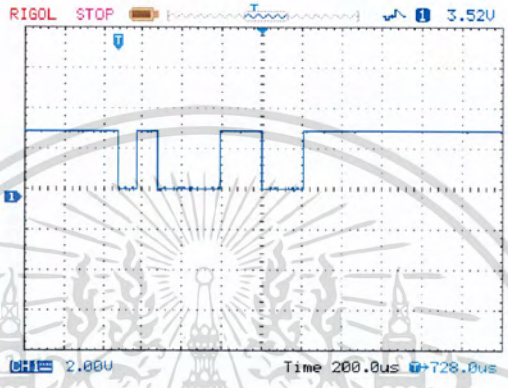
รูปที่ 4.11 วัดขา Tx Object Node ซึ่งส่งอักษร “N”



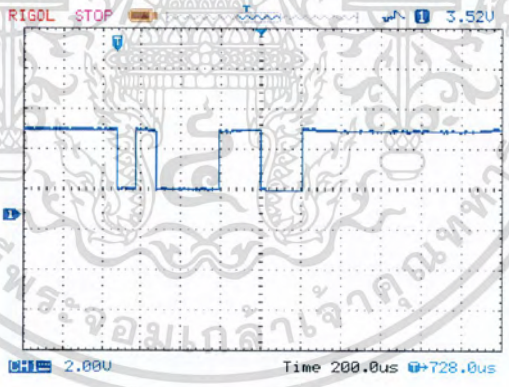
รูปที่ 4.12 วัดขา Tx Object Node ซึ่งส่งอักษร “D”

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3) เปรียบเทียบการรับ – ส่ง ระหว่าง ZigBee 2 ตัว โดยการทดลองนี้ ให้ส่งอักขระ “1” แล้วทำกรวัดค่าเปรียบเทียบกันระหว่าง ขา Tx ของ Object Node กับ ขา Rx ของ Monitor Node จากรูปที่ 4.13 เป็นสัญญาณที่วัดอักษร “1” จากฝั่งส่ง เทียบกับรูปที่ 4.14 เป็นสัญญาณที่วัดจากฝั่งรับ



รูปที่ 4.13 วัดขา Tx Object Node ซึ่งส่งอักษร “1”



รูปที่ 4.14 วัดขา Rx Monitor Node ซึ่งรับอักษร “1”

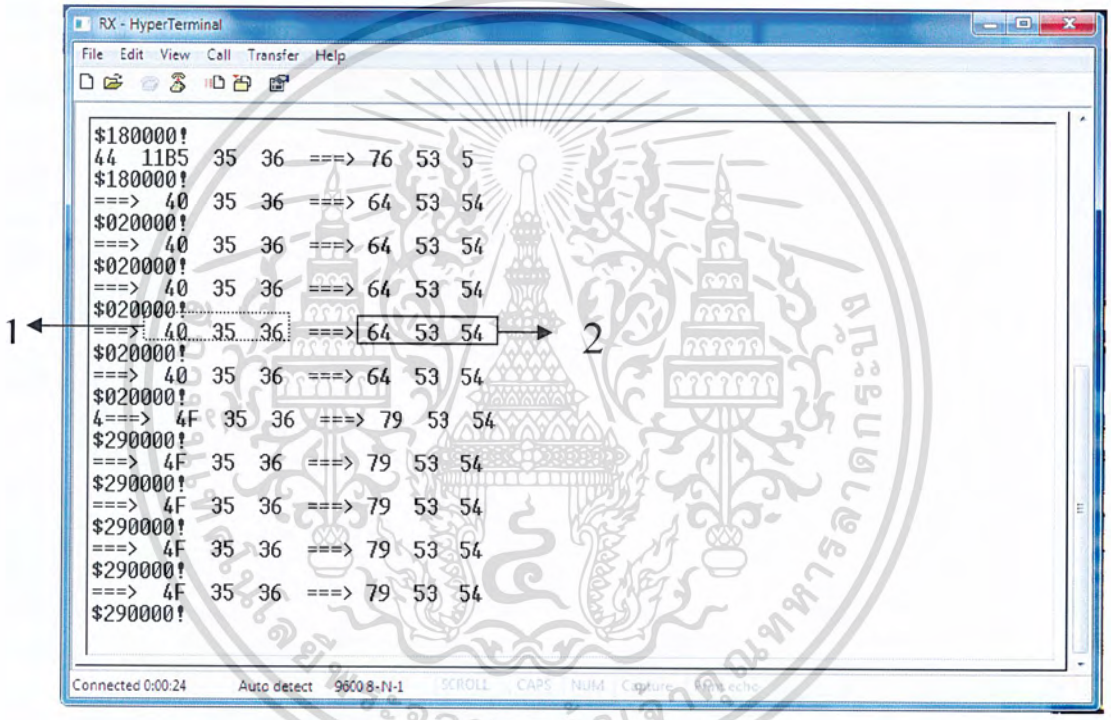
4.3 ผลการทดลองการ Calibrate ระยะทางระหว่าง Object Node และ Monitor Node

ไมโครคอนโทรลเลอร์ที่ Object Node ทำหน้าที่ ร้องขอค่าจาก Monitor Node และแสดงผลค่าที่ได้ผ่านโปรแกรม Hyper Terminal เพื่อใช้เป็นระยะทางอ้างอิงในการหาระบุตำแหน่งวัตถุโดยทำการวัดความแรงสัญญาณระหว่าง ZigBee 2 โหนดที่ระยะทางต่างๆทุกๆ 1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมตร เมตรละ 10 ครั้ง ค่าที่ได้แสดงดังตารางที่ 4.2 แล้วนำค่าที่ได้มาหาค่าเฉลี่ยทางสถิติเพื่อใช้ในการแปลงความแรงสัญญาณเป็นระยะทางต่อไป

การ Calibrate ครั้งนี้ได้ทำการทดลองที่สนามรักบี้ ในบริเวณสนามกีฬาของสถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง เพื่อทดสอบหาค่าความแรงของสัญญาณผ่านโปรแกรม Hyper Terminal ดังรูปที่ 4.15



รูปที่ 4.15 ค่าความแรงของสัญญาณที่รับได้ผ่านโปรแกรม HyperTerminal

หมายเลข 1 คือ ค่าความแรงของสัญญาณในรูปเลขฐาน 16

หมายเลข 2 คือ ค่าความแรงของสัญญาณที่ถูกแปลงมาอยู่ในรูปเลขฐาน 10

จะเห็นว่ามี 3 ค่า ซึ่งแต่ละค่าจะเป็นค่าเฉพาะของ ZigBee แต่ละตัว โดย

ตัวที่หนึ่ง คือ Reference Node ที่มีค่า ID คือ 40633DA9

ตัวที่สอง คือ Reference Node ที่มีค่า ID คือ 406311B5

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวที่สาม คือ Reference Node ที่มีค่า ID คือ 406311C4

ตารางที่ 4.2 ค่าความแรงของสัญญาณ RSSI (ในหน่วย dBm) ที่ระยะทางต่างๆ

ครั้งที่/เมตร	1	2	3	4	5	6	7	8	9	10
0	-36	-40	-40	-36	-40	-36	-40	-36	-36	-36
1	-50	-50	-50	-50	-51	-51	-50	-51	-52	-51
2	-59	-59	-59	-59	-59	-59	-61	-61	-64	-61
3	-68	-66	-69	-69	-69	-72	-72	-72	-72	-70
4	-64	-65	-65	-66	-64	-65	-65	-65	-65	-72
5	-67	-69	-71	-69	-69	-69	-69	-70	-67	-69
6	-70	-71	-73	-73	-74	-71	-71	-72	-72	-71
7	-72	-71	-72	-73	-74	-71	-71	-72	-71	-73
8	-79	-76	-76	-78	-82	-80	-75	-77	-76	-76
9	-79	-76	-76	-78	-82	-80	-75	-77	-76	-76
10	-69	-69	-70	-71	-69	-69	-69	-69	-69	-69
11	-73	-75	-75	-75	-76	-75	-73	-73	-73	-74
12	-74	-83	-83	-81	-75	-74	-74	-75	-76	-73
13	-69	-71	-69	-69	-69	-69	-69	-69	-69	-69
14	-76	-76	-78	-77	-77	-76	-76	-75	-76	-74
15	-71	-72	-72	-70	-72	-72	-72	-72	-70	-73
16	-83	-86	-87	-80	-80	-84	-83	-90	-84	-80
17	-71	-72	-72	-71	-72	-72	-72	-72	-71	-72
18	-80	-75	-78	-75	-77	-81	-77	-74	-79	-76
19	-75	-92	-80	-79	-76	-76	-80	-78	-78	-76
20	-72	-72	-72	-73	-73	-71	-73	-72	-72	-73
21	-75	-73	-76	-77	-76	-76	-75	-75	-75	-75

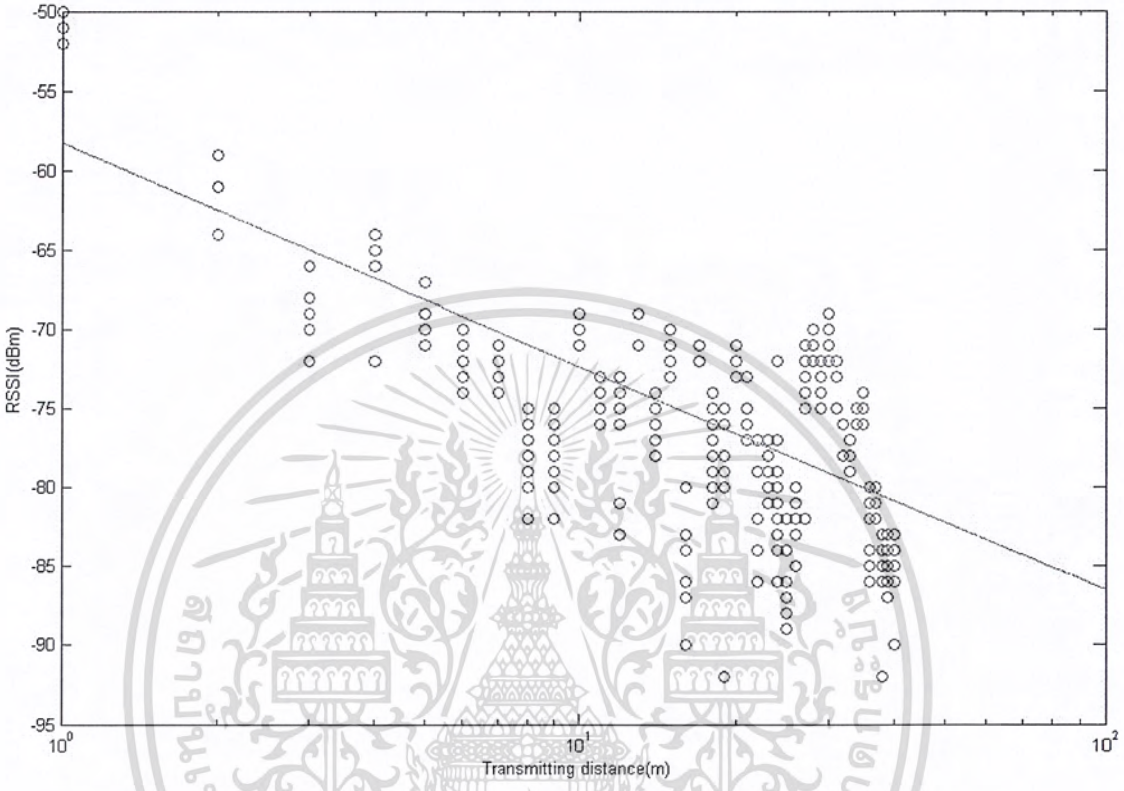
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.2 ค่าความแรงของสัญญาณ RSSI (ในหน่วย dBm) ที่ระยะทางต่างๆ (ต่อ)

ครั้งที่/เมตร	1	2	3	4	5	6	7	8	9	10
22	-79	-77	-79	-79	-77	-77	82	-79	-86	-84
23	-77	-79	-80	-80	-78	-80	-81	-81	-77	-80
24	-86	-72	-80	-81	-83	-84	-86	-82	-79	-77
25	-82	-86	-84	-89	-87	-86	-87	-86	-84	-88
26	-82	-85	-82	-80	-81	-83	-83	-81	-81	-83
27	-82	-75	-72	-71	-74	-72	-73	-74	-72	-74
28	-71	-71	-71	-70	-70	-70	-71	-71	-72	-72
29	-73	-72	-73	-74	-73	-72	-74	-73	-75	-75
30	-72	-69	-72	-70	-72	-71	-72	-72	-72	-72
31	-73	-73	-73	-72	-73	-72	-72	-73	-72	-75
32	-76	-76	-76	-78	-76	-76	-76	-76	-76	-76
33	-79	-78	-77	-78	-77	-77	-77	-77	-77	-78
34	-76	-76	-76	-76	-76	-75	-76	-76	-76	-76
35	-76	-75	-75	-75	-75	-74	-75	-75	-75	-75
36	-84	-81	-86	-85	-82	-84	-80	-82	-82	-56
37	-82	-80	-82	-81	-82	-81	-82	-81	-81	-81
38	-83	-84	-84	-84	-85	-84	-84	-86	-85	-92
39	-86	-86	-84	-86	-87	-85	-84	-85	-86	-83
40	-84	-84	-84	-84	-86	-90	-84	-83	-83	-85

และเมื่อทำการนำค่าที่ทำการทดลองที่ระยะต่างไปทำการคำนวณด้วยสมการ Linear Regression Model จะได้กราฟดังรูปที่ 4.16

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.16 กราฟแสดงความสัมพันธ์ระหว่างระยะทางกับค่าความแรงสัญญาณ

กำหนดให้

$$y = \overline{P_r(d)}[dB], A = -10n$$
$$x = \log d, B = \overline{P_r(1m)}[dB]$$

จากสมการ (2.4) และ (2.5)

$$A\sum_{i=1}^{N} x_i^2 + B\sum_{i=1}^{N} x_i = \sum_{i=1}^{N} x_i y_i$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$A\sum_{i=1}^{N} x_i + BN = \sum_{i=1}^{N} y_i$$

แก้สมการ จะได้ n=1.4092 , $\overline{P_r(1m)}[dB] = -58.2457$

แทนค่า n และ $\overline{P_r(1m)}[dB]$ ในสมการ

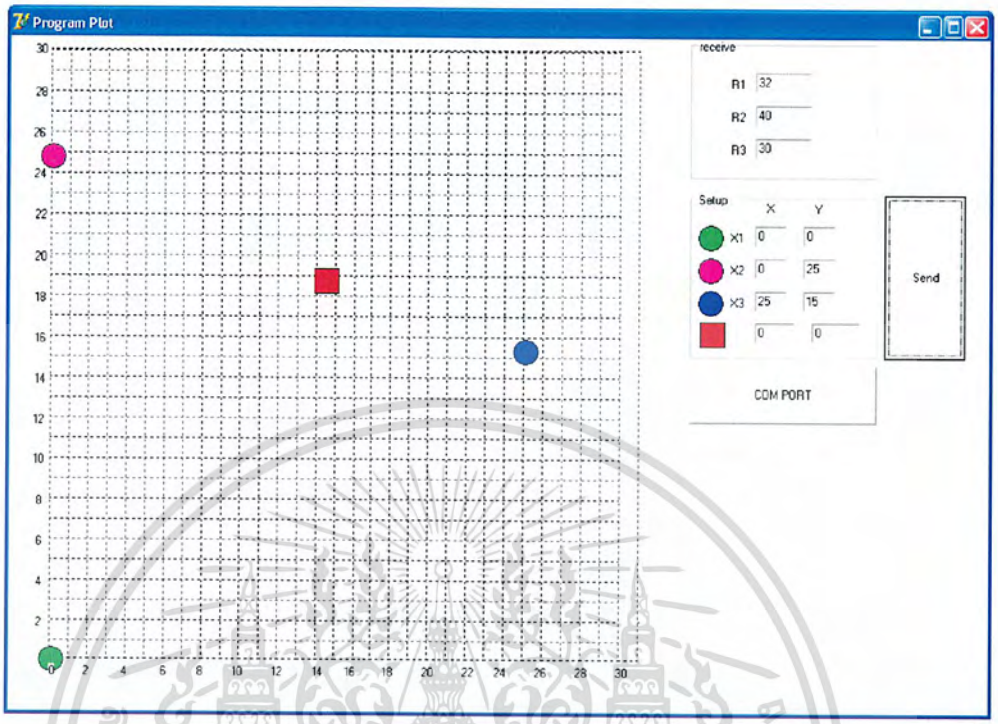
$$\overline{P_r(d)}[dB] = -10n\log d + \overline{P_r(1m)}[dB]$$

จะได้ ค่า Power ที่แต่ละระยะทาง นั่นก็คือค่า RSSI นั่นเอง

4.4 ผลการทดลองการระบุพิกัดของ Object Node ผ่านโปรแกรมแสดงผล

นำอุปกรณ์ของแต่ละ Reference Node ไปวางยังพิกัดที่ต้องการ จากนั้นมากรอกค่าพิกัดต่างๆในช่อง Setup X1,X2 และ X3 ลงในโปรแกรม แล้วกด Send เพื่อส่งค่าพิกัดไปให้โปรแกรม และกด Comport เพื่อกำหนดคุณสมบัติ Comport ที่เราต้องการ จากนั้นโปรแกรมจะประมวลผลเพื่อให้ได้ค่าพิกัดของวัตถุที่ถูกต้อง ดังรูปที่ 4.17 โดยเราจะทำการวัดค่าพิกัดของวัตถุในตำแหน่งต่างๆ เป็นจำนวน 5 ครั้ง เนื่องจากค่าที่ได้มีความคลาดเคลื่อน ไม่แน่นอน แล้วบันทึกค่าไว้ในตารางที่ 4.3 จากนั้นคำนวณหาค่าระยะทางที่ผิดพลาดไปในแต่ละครั้ง พร้อมทั้งค่าเฉลี่ย ดังตารางที่ 4.4

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.17 หน้าต่างแสดงพิกัดและตำแหน่ง

ตารางที่ 4.3 ตำแหน่งของวัตถุ เทียบกับตำแหน่งที่คำนวณได้ และค่าความคลาดเคลื่อน

พิกัดวัตถุจริง (เมตร)		พิกัดวัตถุที่ทำการทดลอง (เมตร)									
		ครั้งที่ 1		ครั้งที่ 2		ครั้งที่ 3		ครั้งที่ 4		ครั้งที่ 5	
x	y	x	y	x	y	x	y	x	y	x	Y
1	1	2	2	1	3	1	1	2	1	2	2
1	5	2	3	1	4	5	2	1	5	3	5
1	10	1	8	2	10	1	10	3	8	1	9
1	15	1	12	2	14	1	9	1	15	3	9
1	20	2	16	3	15	1	14	1	20	5	17
5	1	4	2	4	3	4	1	5	1	5	2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.3 ตำแหน่งของวัตถุ เทียบกับตำแหน่งที่คำนวณได้ และค่าความคลาดเคลื่อน (ต่อ)

พิกัดวัตถุจริง (เมตร)		พิกัดวัตถุที่ทำการทดลอง (เมตร)									
		ครั้งที่ 1		ครั้งที่ 2		ครั้งที่ 3		ครั้งที่ 4		ครั้งที่ 5	
5	5	5	6	5	5	6	5	4	5	5	5
5	10	4	8	5	9	4	10	5	10	5	12
5	15	4	9	5	12	3	15	5	15	6	15
5	20	4	20	5	20	5	15	3	20	6	20
10	1	8	2	10	2	10	1	9	4	5	2
10	5	10	2	10	5	8	5	9	5	10	2
10	10	9	9	8	10	10	10	10	12	11	13
10	15	11	14	12	15	10	15	9	14	13	15
10	20	9	18	10	19	10	20	13	18	15	20
15	1	14	0	13	1	15	1	16	3	15	6
15	5	15	6	15	5	13	5	15	7	14	2
15	10	15	9	15	8	12	7	14	10	17	8
15	15	14	13	15	17	15	15	12	17	15	14
15	20	14	18	11	15	15	20	15	21	17	20
20	1	21	2	18	1	20	1	21	4	19	1
20	5	18	3	19	2	20	5	21	4	20	5
20	10	21	7	19	10	20	10	18	10	22	8

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.3 ตำแหน่งของวัตถุ เทียบกับตำแหน่งที่คำนวณได้ และค่าความคลาดเคลื่อน (ต่อ)

พิกัดวัตถุจริง (เมตร)		พิกัดวัตถุที่ทำการทดลอง (เมตร)									
		ครั้งที่ 1		ครั้งที่ 2		ครั้งที่ 3		ครั้งที่ 4		ครั้งที่ 5	
20	15	17	16	19	15	20	15	21	14	20	15
20	20	22	19	21	20	20	20	19	15	21	20

ตารางที่ 4.4 ค่าความผิดพลาดของการทดลอง และค่าเฉลี่ยของความผิดพลาด

ครั้งที่ 1 (เมตร)	ครั้งที่ 2 (เมตร)	ครั้งที่ 3 (เมตร)	ครั้งที่ 4 (เมตร)	ครั้งที่ 5 (เมตร)	ค่าเฉลี่ย (เมตร)
1.41	2.00	0.00	1.00	1.41	1.17
2.24	1.00	5.00	0.00	2.00	2.05
2.00	1.00	0.00	2.83	1.00	1.37
3.00	1.41	6.00	0.00	6.32	3.35
4.12	5.39	6.00	0.00	5.00	4.10
1.41	2.24	1.00	0.00	1.00	1.13
1.00	0.00	1.00	1.00	0.00	0.60
2.24	1.00	1.00	0.00	2.00	1.25
6.08	3.00	2.00	0.00	1.00	2.42
1.00	0.00	5.00	2.00	1.00	1.80
2.24	1.00	0.00	3.16	5.10	2.30
3.00	0.00	2.00	1.00	3.00	1.80
1.41	2.00	0.00	2.00	3.16	1.72
1.41	2.00	0.00	1.41	3.00	1.57
2.24	1.00	0.00	3.61	5.00	2.37
1.41	2.00	0.00	2.24	5.00	2.13

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.4 ค่าความผิดพลาดของการทดลอง และค่าเฉลี่ยของความผิดพลาด (ต่อ)

ครั้งที่ 1 (เมตร)	ครั้งที่ 2 (เมตร)	ครั้งที่ 3 (เมตร)	ครั้งที่ 4 (เมตร)	ครั้งที่ 5 (เมตร)	ค่าเฉลี่ย (เมตร)
1.00	0.00	2.00	2.00	3.16	1.63
1.00	2.00	4.24	1.00	2.83	2.21
2.24	2.00	0.00	3.61	1.00	1.77
2.24	6.40	0.00	1.00	2.00	2.33
1.41	2.00	0.00	3.16	1.00	1.52
2.83	3.16	0.00	1.41	0.00	1.48
3.16	1.00	0.00	2.00	2.83	1.80
3.16	1.00	0.00	1.41	0.00	1.12
2.24	1.00	0.00	5.10	1.00	1.87
ค่าเฉลี่ยความผิดพลาดทั้งหมด (เมตร)					1.87

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

สรุปผลและข้อเสนอแนะ

ปริญญานิพนธ์นี้เสนอวิธีการคำนวณตำแหน่งวัตถุโดยใช้ความแรงของสัญญาณจากจุดอ้างอิงรอบพื้นที่ที่ใช้ในการทดลอง มาทำการคำนวณด้วยวิธี Trilaterion ซึ่งค่าความแรงของสัญญาณนี้ใช้หาระยะทางจากสมการ Linear Regression Model วิธีการเหล่านี้สามารถที่จะนำมาคำนวณและระบุตำแหน่งของวัตถุภายในอาคารได้ แต่ยังมีข้อจำกัดเรื่องความถูกต้องของตำแหน่งที่ได้

5.1 สรุปผล

1) Zigbee ที่ Object Node สามารถส่งคำร้องขอค่าความแรงของสัญญาณ และรับค่าความแรงที่ได้จาก Reference Node ทั้ง 3 จุด มาทำการบันทึกค่าลงบนไมโครคอนโทรลเลอร์ ก่อนที่จะทำการส่งไปให้แก่ Zigbee ที่ Monitor Node
2) Zigbee ที่ Monitor Node สามารถแปลงค่าความแรงของสัญญาณที่ได้เป็นค่าระยะทางโดยใช้สมการ Linear Regression Model แล้วทำการส่งค่าระยะทางที่ทำการแปลงแล้วไปยังคอมพิวเตอร์เพื่อทำการระบุตำแหน่ง พร้อมแสดงผล
3) โปรแกรมแสดงผลบนคอมพิวเตอร์ทำการคำนวณหาพิกัดโดยใช้วิธี Trilaterion แล้วทำการแสดงพิกัดของวัตถุลงบนโปรแกรมแสดงผล
4) ค่าความผิดพลาดของการระบุตำแหน่งเฉลี่ยคิดเป็น1.87 เมตร

5.2 ปัญหาที่พบ

1) ค่าความแรงของสัญญาณ ณ ระยะทางเดียวกันไม่ค่อยคงที่ เนื่องจากอุปกรณ์ที่ใช้ถูกรบกวนจากสภาพแวดล้อม เช่น สัญญาณไร้สาย สิ่งกีดขวาง
2) อุปกรณ์มีโอกาสเสียหายได้ง่าย ถ้าหากไม่ระมัดระวังในระหว่างการใช้งาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.3 ข้อเสนอแนะ

สถานที่ที่ใช้ในการทดลองจริง ไม่ควรมีสิ่งที่รบกวนต่อการวัดค่าความแรงของสัญญาณ เพราะจะทำให้ค่าความแรงของสัญญาณที่วัดได้มีค่าไม่คงที่และผิดพลาด หากต้องการความแม่นยำสูงก็สามารถเพิ่มจำนวนโนดอ้างอิงเพื่อช่วยในการประมาณตำแหน่งวัตถุและสามารถใช้วิธีการตรวจลายพิมพ์ (FingerPrint) เพื่อบันทึกค่าตามตำแหน่งต่างๆสำหรับการเปรียบเทียบจะทำให้ค่าที่ได้มีความถูกต้องน่าเชื่อถือมากขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

[1] Shashank Tadamadia. "*Indoor Local Positioning System For ZigBee,Base On RSSI*". M.Sc.Thesis report, Electrical Engineering D, The Department of Information Technology and Media,Mid Sweden University, 2006.

[2] ประจิน พลังสันติกุลล. *PIC Works Examples and C Source Code*.กรุงเทพฯ : บริษัท แอพซอฟต์เทค, 2537.

[3] ประจิน พลังสันติกุลล. *All About CCS C (Programming with CCS C Compiler)*.กรุงเทพ: บริษัท แอพซอฟต์เทค, 2551.

[4] ดอนสัน ปงผาม, ทิพวัลย์ คำน้ำนอง. *ไมโครคอนโทรลเลอร์ PIC และการประยุกต์ใช้งาน*. ครั้งที่ 3. กรุงเทพฯ : สำนักพิมพ์ ส.ส.ท. , 2552.

[5] สัจจะ จรัสรุ่งรวีวร, จักรพงษ์ สุขประเสริฐ.*เริ่มต้นอย่างมืออาชีพด้วย Delphi 7 ฉบับสมบูรณ์*. กรุงเทพฯ : สำนักพิมพ์ Dev Book, 2550.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ก

โค้ด PIC 18F2525 ที่ Object Node

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
#include <18F2525.h>
#define TX1   PIN_C6
#define RX1   PIN_C7
#include   <string.h>
#include   <stdlib.h>
#include   <stdio.h>

#fuses   HS,NOLVP,NOWDT,NOPROTECT,NOSTVREN,NOPUT
#use     delay (clock = 4000000)
#use     rs232(baud = 9600,parity=N, xmit = TX1, rcv = RX1)
#use fast_io(A)
#use fast_io(B)

char data = 0;
int16 data_xbee,data_xbee1;
char data_buff[150];
int32 i,j,count;

#int_RDA
void RxD_IRS(void)
{
  data = getc();
  if((data == 'K'))
  {
     data_xbee = 1;
     count=0;
  }
   if ((data_xbee1 == 1)&&(data != 'K')&&(data != 'O'))
   {
     count++;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
      data_buff[count] = data;
    }
}

void main (void)
{
   enable_interrupts(GLOBAL);
   enable_interrupts(INT_RDA);
  data_xbee1 = 1;
   data_xbee = 0;

  while(1)
  {
      delay_ms(2000);
      data_xbee = 0;
      putchar('+');
       delay_ms(10);
       putchar('+');
       delay_ms(10);
       putchar('+');
      while(data_xbee == 0)
      {
         data_xbee1 = 1;
       }

       putchar('A');
       delay_ms(10);
       putchar('T');
       delay_ms(10);
       putchar('N');
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        delay_ms(10);
        putchar('D');
        delay_ms(10);
        putchar(0x0D);
        data_xbee = 0;
        delay_ms(3000);

        putchar('A');
        delay_ms(10);
        putchar('T');
        delay_ms(10);
        putchar('C');
        delay_ms(10);
        putchar('N');
        delay_ms(10);
        putchar(0x0D);

        while(data_xbee == 0)
       {
       data_xbee1 = 0;
        };
      delay_ms(100);
       if(data_xbee1 == 0)
       {
       for(i=1;i<=100;i++)
          putchar(data_buff[i]);
       }
   }
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ข

โค้ด PIC 18F2525 ที่ Monitor Node

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
#include <18F2525.h>
#include <stdlib.h>
#define TX1   PIN_C6
#define RX1   PIN_C7
#include   <string.h>
#include   <stdlib.h>
#include   <stdio.h>
#include   <math.h>

#fuses   HS,NOLVP,NOWDT,NOPROTECT,NOSTVREN,NOPUT
#use     delay (clock = 4000000)
#use     rs232(baud = 9600,parity=N, xmit = TX1, rcv = RX1)
#use fast_io(A)
#use fast_io(B)

char ch = 0;
char data1[2];
char data2[2];
char data3[2];
char strat_Buff1 = 0,strat_Buff2 = 0,strat_Buff3 = 0;
char strat_Buff1_1 = 0,strat_Buff2_1 = 0,strat_Buff3_1 = 0;
int index_data1,index_data2,index_data3;
int data_Save1_,data_Save2_,data_Save3_;
int32 Conversion,Distance;
int32 Distance1,Distance2,Distance3;

char data_Save1[2];
char data_Save2[2];
char data_Save3[2];
int j,k;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
#int_RDA
void RxD_IRS(void)
{
    ch = getc();
  putchar(ch);
////////////////////////////////////////////// data 1 ////////////////////////////////////////////////////
  if((strat_Buff1_1 == 1)&&(ch != '\r')&&(ch != '\n'))
  {
    data1[index_data1] = ch;
    index_data1++;
    if(index_data1 >= 2)
    {
      strat_Buff1_1 = 0;
      data_Save1[0] = data1[0];
      data_Save1[1] = data1[1];
    }
  }
 if((strat_Buff1 == 1)&&(ch=='5'))
 {
    for(j=0;j<=2;j++)  data1[j] = 0;
   index_data1 = 0;
   strat_Buff1_1 = 1;

 }else strat_Buff1 = 0;
   if(ch=='B')
 {
  strat_Buff1 = 1;
 }

////////////////////////////////////////////// data 2 ////////////////////////////////////////////////////
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  if((strat_Buff2_1 == 1)&&(ch != '\r')&&(ch != '\n'))
  {
    data2[index_data2] = ch;
    index_data2++;
    if(index_data2 >= 2)
    {
      strat_Buff2_1 = 0;
      data_Save2[0] = data2[0];
      data_Save2[1] = data2[1];
    }
  }
  if((strat_Buff2 == 1)&&(ch=='A'))
  {
    for(j=0;j<=2;j++)  data2[j] = 0;
    index_data2 = 0;
    strat_Buff2_1 = 1;
  }else strat_Buff2 = 0;
    if(ch=='6')
  {
    strat_Buff2 = 1;
  }
////////////////////////////////////////////// data 3 ////////////////////////////////////////////////////////
  if((strat_Buff3_1 == 1)&&(ch != '\r')&&(ch != '\n'))
  {
    data3[index_data3] = ch;
    index_data3++;
    if(index_data3 >= 2)
    {
      strat_Buff3_1 = 0;
      data_Save3[0] = data3[0];
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        data_Save3[1] = data3[1];
      }
    }
    if((strat_Buff3 == 1)&&(ch=='4'))
    {
      for(j=0;j<=2;j++)  data3[j] = 0;
      index_data3 = 0;
      strat_Buff3_1 = 1;
    }else strat_Buff3 = 0;
      if(ch=='C')
    {
      strat_Buff3 = 1;
    }
}
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////

int32 convert (char Conversion1,char Conversion2)
{
  if((Conversion1 == '2')&&(Conversion2 == '0'))  Conversion = 32;
   else if((Conversion1 == '2')&&(Conversion2 == '1'))  Conversion = 33;
   else if((Conversion1 == '2')&&(Conversion2 == '2'))  Conversion = 34;
   else if((Conversion1 == '2')&&(Conversion2 == '3'))  Conversion = 35;
   else if((Conversion1 == '2')&&(Conversion2 == '4'))  Conversion = 36;
   else if((Conversion1 == '2')&&(Conversion2 == '5'))  Conversion = 37;
   else if((Conversion1 == '2')&&(Conversion2 == '6'))  Conversion = 38;
   else if((Conversion1 == '2')&&(Conversion2 == '7'))  Conversion = 39;
   else if((Conversion1 == '2')&&(Conversion2 == '8'))  Conversion = 40;
   else if((Conversion1 == '2')&&(Conversion2 == '9'))  Conversion = 41;
   else if((Conversion1 == '2')&&(Conversion2 == 'A'))  Conversion = 42;
   else if((Conversion1 == '2')&&(Conversion2 == 'B'))  Conversion = 43;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
else if((Conversion1 == '2')&&(Conversion2 == 'C'))  Conversion = 44;
else if((Conversion1 == '2')&&(Conversion2 == 'D'))  Conversion = 45;
else if((Conversion1 == '2')&&(Conversion2 == 'E'))  Conversion = 46;
else if((Conversion1 == '2')&&(Conversion2 == 'F'))  Conversion = 47;
else if((Conversion1 == '3')&&(Conversion2 == '0'))  Conversion = 48;
else if((Conversion1 == '3')&&(Conversion2 == '1'))  Conversion = 49;
else if((Conversion1 == '3')&&(Conversion2 == '2'))  Conversion = 50;
else if((Conversion1 == '3')&&(Conversion2 == '3'))  Conversion = 51;
else if((Conversion1 == '3')&&(Conversion2 == '4'))  Conversion = 52;
else if((Conversion1 == '3')&&(Conversion2 == '5'))  Conversion = 53;
else if((Conversion1 == '3')&&(Conversion2 == '6'))  Conversion = 54;
else if((Conversion1 == '3')&&(Conversion2 == '7'))  Conversion = 55;
else if((Conversion1 == '3')&&(Conversion2 == '8'))  Conversion = 56;
else if((Conversion1 == '3')&&(Conversion2 == '9'))  Conversion = 57;
else if((Conversion1 == '3')&&(Conversion2 == 'A'))  Conversion = 58;
else if((Conversion1 == '3')&&(Conversion2 == 'B'))  Conversion = 59;
else if((Conversion1 == '3')&&(Conversion2 == 'C'))  Conversion = 60;
else if((Conversion1 == '3')&&(Conversion2 == 'D'))  Conversion = 61;
else if((Conversion1 == '3')&&(Conversion2 == 'E'))  Conversion = 62;
else if((Conversion1 == '3')&&(Conversion2 == 'F'))  Conversion = 63;
else if((Conversion1 == '4')&&(Conversion2 == '0'))  Conversion = 64;
else if((Conversion1 == '4')&&(Conversion2 == '1'))  Conversion = 65;
else if((Conversion1 == '4')&&(Conversion2 == '2'))  Conversion = 66;
else if((Conversion1 == '4')&&(Conversion2 == '3'))  Conversion = 67;
else if((Conversion1 == '4')&&(Conversion2 == '4'))  Conversion = 68;
else if((Conversion1 == '4')&&(Conversion2 == '5'))  Conversion = 69;
else if((Conversion1 == '4')&&(Conversion2 == '6'))  Conversion = 70;
else if((Conversion1 == '4')&&(Conversion2 == '7'))  Conversion = 71;
else if((Conversion1 == '4')&&(Conversion2 == '8'))  Conversion = 72;
else if((Conversion1 == '4')&&(Conversion2 == '9'))  Conversion = 73;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
else if((Conversion1 == '4')&&(Conversion2 == 'A'))  Conversion = 74;
else if((Conversion1 == '4')&&(Conversion2 == 'B'))  Conversion = 75;
else if((Conversion1 == '4')&&(Conversion2 == 'C'))  Conversion = 76;
else if((Conversion1 == '4')&&(Conversion2 == 'D'))  Conversion = 77;
else if((Conversion1 == '4')&&(Conversion2 == 'E'))  Conversion = 78;
else if((Conversion1 == '4')&&(Conversion2 == 'F'))  Conversion = 79;
else if((Conversion1 == '5')&&(Conversion2 == '0'))  Conversion = 80;
else if((Conversion1 == '5')&&(Conversion2 == '1'))  Conversion = 81;
else if((Conversion1 == '5')&&(Conversion2 == '2'))  Conversion = 82;
else if((Conversion1 == '5')&&(Conversion2 == '3'))  Conversion = 83;
else if((Conversion1 == '5')&&(Conversion2 == '4'))  Conversion = 84;
else if((Conversion1 == '5')&&(Conversion2 == '5'))  Conversion = 85;
else if((Conversion1 == '5')&&(Conversion2 == '6'))  Conversion = 86;
else if((Conversion1 == '5')&&(Conversion2 == '7'))  Conversion = 87;
else if((Conversion1 == '5')&&(Conversion2 == '8'))  Conversion = 88;
else if((Conversion1 == '5')&&(Conversion2 == '9'))  Conversion = 89;
else if((Conversion1 == '5')&&(Conversion2 == 'A'))  Conversion = 90;
else if((Conversion1 == '5')&&(Conversion2 == 'B'))  Conversion = 91;
else if((Conversion1 == '5')&&(Conversion2 == 'C'))  Conversion = 92;
else if((Conversion1 == '5')&&(Conversion2 == 'D'))  Conversion = 93;
else if((Conversion1 == '5')&&(Conversion2 == 'E'))  Conversion = 94;
else if((Conversion1 == '5')&&(Conversion2 == 'F'))  Conversion = 95;
else if((Conversion1 == '6')&&(Conversion2 == '0'))  Conversion = 96;
else if((Conversion1 == '6')&&(Conversion2 == '1'))  Conversion = 97;
else if((Conversion1 == '6')&&(Conversion2 == '2'))  Conversion = 98;
else if((Conversion1 == '6')&&(Conversion2 == '3'))  Conversion = 99;
else if((Conversion1 == '6')&&(Conversion2 == '4'))  Conversion = 100;
else if((Conversion1 == '6')&&(Conversion2 == '5'))  Conversion = 101;
else if((Conversion1 == '6')&&(Conversion2 == '6'))  Conversion = 102;
else if((Conversion1 == '6')&&(Conversion2 == '7'))  Conversion = 103;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    else if((Conversion1 == '6')&&(Conversion2 == '8'))  Conversion = 104;
    else if((Conversion1 == '6')&&(Conversion2 == '9'))  Conversion = 105;
    else if((Conversion1 == '6')&&(Conversion2 == 'A'))  Conversion = 106;
    else if((Conversion1 == '6')&&(Conversion2 == 'B'))  Conversion = 107;
    else if((Conversion1 == '6')&&(Conversion2 == 'C'))  Conversion = 108;
    else if((Conversion1 == '6')&&(Conversion2 == 'D'))  Conversion = 109;
    else if((Conversion1 == '6')&&(Conversion2 == 'E'))  Conversion = 110;
    return(Conversion);
}

int32 calculator (int32 cal)
{
   Distance =  pow(10,(((float)(cal*-1)+58.2457)/-14.092));
   return(Distance);
}

void main (void)
{
   enable_interrupts(GLOBAL);
   enable_interrupts(INT_RDA);
  while(1)
  {
     data_Save1_ = convert(data_Save1[0],data_Save1[1]);
     data_Save2_ = convert(data_Save2[0],data_Save2[1]);
     data_Save3_ = convert(data_Save3[0],data_Save3[1]);
    printf("===> %c%c %c%c %c%c ===> %d %d %d
\r\n",data_Save1[0],data_Save1[1],data_Save2[0],data_Save2[1],data_Save3[0],data_Save3[1],dat
a_Save1_,data_Save2_,data_Save3_);
     Distance1 = calculator(data_Save1_);
     Distance2 = calculator(data_Save2_);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        Distance3 = calculator(data_Save3_);
        printf("$%02lu%02lu%02lu!\r\n",Distance1,Distance2,Distance3);
       delay_ms(1000);
    }
}
```


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ค

โค้ดโปรแกรมแสดงผลโดยใช้โปรแกรม Delphi

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
unit Unit1;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, StdCtrls, ExtCtrls, CPortCtl, CPort;

type
  TForm1 = class(TForm)
  Shape1: TShape;
  Button1: TButton;
  GroupBox1: TGroupBox;
  Edit7: TEdit;
  Edit8: TEdit;
  Edit9: TEdit;
  Button2: TButton;
  Label5: TLabel;
  Label6: TLabel;
  Label7: TLabel;
  ComPort1: TComPort;
  ComTerminal1: TComTerminal;
  Button3: TButton;
  Button4: TButton;
  GroupBox2: TGroupBox;
  Label8: TLabel;
  Label9: TLabel;
  Label10: TLabel;
  Edit1: TEdit;
  Edit2: TEdit;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Label11: TLabel;

Label12: TLabel;

Edit3: TEdit;

Edit4: TEdit;

Edit5: TEdit;

Edit6: TEdit;

GroupBox3: TGroupBox;

Label3: TLabel;

Label4: TLabel;

Edit10: TEdit;

Edit11: TEdit;

Edit12: TEdit;

Edit13: TEdit;

Shape7: TShape;

Shape8: TShape;

Shape9: TShape;

Shape10: TShape;

Shape11: TShape;

Shape12: TShape;

Shape13: TShape;

Shape6: TShape;

Shape14: TShape;

Shape15: TShape;

Shape16: TShape;

Shape17: TShape;

Shape18: TShape;

Shape19: TShape;

Shape20: TShape;

Shape21: TShape;

Shape22: TShape;

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Shape23: TShape;

Shape24: TShape;

Shape25: TShape;

Shape26: TShape;

Shape27: TShape;

Shape28: TShape;

Shape29: TShape;

Shape30: TShape;

Shape31: TShape;

Shape32: TShape;

Shape33: TShape;

Shape34: TShape;

Shape35: TShape;

Shape38: TShape;

Shape40: TShape;

Shape41: TShape;

Shape42: TShape;

Shape43: TShape;

Shape44: TShape;

Shape45: TShape;

Shape46: TShape;

Shape47: TShape;

Shape48: TShape;

Shape49: TShape;

Shape50: TShape;

Shape51: TShape;

Shape52: TShape;

Shape53: TShape;

Shape54: TShape;

Shape55: TShape;

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Shape56: TShape;

Shape57: TShape;

Shape58: TShape;

Shape59: TShape;

Shape60: TShape;

Shape61: TShape;

Shape62: TShape;

Shape63: TShape;

Shape64: TShape;

Shape65: TShape;

Shape66: TShape;

Shape67: TShape;

Shape68: TShape;

Shape2: TShape;

Shape5: TShape;

Shape4: TShape;

Shape3: TShape;

Shape36: TShape;

Label1: TLabel;

Label2: TLabel;

Label13: TLabel;

Label14: TLabel;

Label15: TLabel;

Label16: TLabel;

Label17: TLabel;

Label18: TLabel;

Label19: TLabel;

Label20: TLabel;

Label21: TLabel;

Label22: TLabel;

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Label23: TLabel;
Label24: TLabel;
Label25: TLabel;
Label26: TLabel;
Label27: TLabel;
Label28: TLabel;
Label29: TLabel;
Label30: TLabel;
Label31: TLabel;
Label32: TLabel;
Label33: TLabel;
Label34: TLabel;
Label35: TLabel;
Label36: TLabel;
Label37: TLabel;
Label38: TLabel;
Label39: TLabel;
Label40: TLabel;
Label41: TLabel;
Label42: TLabel;
Shape37: TShape;
Shape39: TShape;
Shape69: TShape;
Shape70: TShape;
procedure Button1Click(Sender: TObject);
procedure FormCreate(Sender: TObject);
procedure Button3Click(Sender: TObject);
procedure ComTerminal1Char(Sender: TObject; Ch: Char);
procedure Button4Click(Sender: TObject);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  private
    { Private declarations }
  public
    { Public declarations }
    LengX , LengY : integer;
    start_b_x , start_b_y,Ready : integer;
    temp_st : string;

    detA,detA1,detA2 : real;

    X3X1   : real;
    X3X2   : real;
    Y3Y1   : real;
    Y3Y2   : real;

    X2X1   : real;

    XU,YU  : real;
    R1,R2,R3 : real;
    SUMR1,SUMR2 : real;
    X1,X2,X3,Y1,Y2,Y3 :real;
    XXX1,YYY1 : real;
    XXX2,YYY2 : real;
    XXX3,YYY3 : real;

    XX,YY ,mux: integer;

  end;
var
  Form1: TForm1;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
implementation

{$R *.dfm}
////////////////////////////////ปุ่ม send//////////////////////////////////////
procedure TForm1.Button1Click(Sender: TObject);
begin

    Shape2.Left:= start_b_x + LengX-  ( LengX -  strtoint(edit1.Text)*mux+10);
    Shape2.top:=  start_b_y + LengY-( LengY +  strtoint(edit2.Text)*20-15);

    Shape3.Left:= start_b_x + LengX-( LengX -  strtoint(edit3.Text)*mux+10);
    Shape3.top:=  start_b_y + LengY-( LengY +  strtoint(edit4.Text)*20-15);

    Shape4.Left:= start_b_x + LengX-( LengX -  strtoint(edit5.Text)*mux+10);
    Shape4.top:=  start_b_y + LengY-( LengY +  strtoint(edit6.Text)*20-15);

    X1:=strtofloat(edit1.Text);
    Y1:=strtofloat(edit2.Text);
    X2:=strtofloat(edit3.Text);
    Y2:=strtofloat(edit4.Text);
    X3:=strtofloat(edit5.Text);
    Y3:=strtofloat(edit6.Text);

end;

////////////////////////////////form ใหญ่//////////////////////////////////////
procedure TForm1.FormCreate(Sender: TObject);
begin
mux:=19;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
start_b_x :=40;
start_b_y :=585;
Shape1.Left:= 40;
Shape1.top:= 12;

LengX :=  585;
LengY :=  585;
Shape1.Width:=LengX;
Shape1.Height:=LengY;

start_b_x :=40;
start_b_y :=585;

Button1.Click;
end;

////////////////////////////ปุ่ม COM PORT (hidden)//////////////////////////////////////
procedure TForm1.Button3Click(Sender: TObject);
begin
   ComPort1.ShowSetupDialog;
   ComPort1.Connected:=true;
end;

////////////////////////////HyperTerminal Test(DOS)//////////////////////////////////////
procedure TForm1.ComTerminal1Char(Sender: TObject; Ch: Char);
begin
   if Ch='$' then    begin temp_st:=''; Ready := 1; end
   else  if Ch='!' then
   begin
       if (Ready =1) and (Length(temp_st)>=6) then
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        begin
          edit7.Text :=   temp_st[1]+temp_st[2];
          edit8.Text :=   temp_st[3]+temp_st[4];
          edit9.Text :=   temp_st[5]+temp_st[6];
          Ready:=0;
           R1 := strtoint(trim(edit7.Text));
           R2 := strtoint(trim(edit8.Text));
           R3 := strtoint(trim(edit9.Text));

           Button4.Click;
        end;

   end
   else
   begin
      if (Ch >= '0') and (Ch<='9') then temp_st:=temp_st+Ch;
   end;
end;

////////////////////////////////ปุ่ม calculator (hidden)////////////////////////////////////
procedure TForm1.Button4Click(Sender: TObject);
begin

   //==================== X3 X1  X3 X2 ================
 {
   R1 := strtofloat(edit7.Text);
   R2 := strtofloat(edit8.Text);
   R3 := strtofloat(edit9.Text);
   }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
try

X3X1 := X3-X1;
X3X2 := X3-X2;
Y3Y1 := Y3-Y1;
Y3Y2:= Y3-Y2;
SUMR1 := (R1*R1-R3*R3) - (X1*X1-X3*X3) - (Y1*Y1-Y3*Y3);
SUMR2 := (R2*R2-R3*R3) - (X2*X2-X3*X3) - (Y2*Y2-Y3*Y3);
detA := (X3X1*Y3Y2) - (X3X2*Y3Y1);
detA1 := (SUMR1*Y3Y2) - (SUMR2*Y3Y1);
detA2 := (X3X1 * SUMR2) - (X3X2 * SUMR1);

if detA <> 0 then
begin
    XXX1 := (detA1/detA)/2;
    YYY1 := (detA2/detA)/2;
end
else
begin
    XXX1:=0;
    YYY1:=0;
end;

//====================== X2 X1 X3 X1 ====================
{
R1 := strtofloat(edit7.Text);
R2 := strtofloat(edit8.Text);
R3 := strtofloat(edit9.Text); }
X3X1 := X2-X1;
X3X2 := X3-X1;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Y3Y1 := Y2-Y1;
Y3Y2:= Y3-Y1;
SUMR1 := (R1*R1-R2*R2) - (X1*X1-X2*X2) - (Y1*Y1-Y2*Y2);
SUMR2 := (R1*R1-R3*R3) - (X1*X1-X3*X3) - (Y1*Y1-Y3*Y3);

detA := (X3X1*Y3Y2) - (X3X2*Y3Y1);
detA1 := (SUMR1*Y3Y2) - (SUMR2*Y3Y1);
detA2 := (X3X1 * SUMR2) - (X3X2 * SUMR1);

if detA <> 0 then
begin
    XXX2 := (detA1/detA)/2;
    YYY2 := (detA2/detA)/2;
end
 else
begin
    XXX2:=0;
    YYY2:=0;
end;

//========================== X2 X1 X3 X2 ======================
{
R1 := strtofloat(edit7.Text);
R2 := strtofloat(edit8.Text);
R3 := strtofloat(edit9.Text);   }

X3X1 := X2-X1;
X3X2 := X3-X2;
Y3Y1 := Y2-Y1;
Y3Y2:= Y3-Y2;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
SUMR1 :=  (R1*R1-R2*R2) - (X1*X1-X2*X2) - (Y1*Y1-Y2*Y2);
SUMR2 :=  (R2*R2-R3*R3) - (X2*X2-X3*X3) - (Y2*Y2-Y3*Y3);

detA  :=    (X3X1*Y3Y2) - (X3X2*Y3Y1);
detA1 :=    (SUMR1*Y3Y2) - (SUMR2*Y3Y1);
detA2 :=    (X3X1 * SUMR2) - (X3X2 * SUMR1);

if  (detA <> 0 )then
begin
     XXX3 :=  (detA1/detA)/2;
     YYY3 :=  (detA2/detA)/2;
end
  else
begin
     XXX3:=0;
     YYY3:=0;
end;

XXX1 := (XXX1+XXX2+XXX3)/3;
YYY1 := (YYY1+YYY2+YYY3)/3;

edit12.Text := formatfloat('###',(XXX1));
edit13.Text := formatfloat('###',(YYY1));

if strtoint(trim(edit12.Text)) < 10 then
begin
      if  edit12.Text[1] <> '-' then   XX :=  strtoint(edit12.Text[1]);
end
else   if strtoint(trim(edit12.Text)) < 100 then
begin
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        XX := strtoint(edit12.Text[1])*10   + strtoint(edit12.Text[2]);


    end
    else  if strtoint(trim(edit12.Text)) < 1000 then
    begin
        XX :=    strtoint(edit12.Text[1])*100 + strtoint(edit12.Text[2])*10 +
strtoint(edit12.Text[3]);
    end;

    if strtoint(trim(edit13.Text)) < 0 then
        edit13.Text:='0';
        YY:=0;

   if strtoint(trim(edit13.Text)) < 10 then
    begin
        YY := strtoint(edit13.Text[1]);
    end
    else   if strtoint(trim(edit13.Text)) < 100 then
    begin
        YY := strtoint(edit13.Text[1])*10 +  strtoint(edit13.Text[2]);
    end
    else  if strtoint(edit13.Text) < 1000 then
    begin
        YY :=    strtoint(edit13.Text[1])*100 +strtoint(edit13.Text[2])*10 +
strtoint(edit13.Text[3]);
    end;


    Shape5.Left:= start_b_x + LengX-( LengX -  XX*mux+10);
    Shape5.top:=  start_b_y + LengY-( LengY +  YY*mux);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    finally

    end;
end;
end.
```


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้