การติดตามรถไฟฟ้าโดยระบบบอกตำแหน่งพิกัดโลกและการสื่อสารแบบไร้สาย

EV Tracking via GPS and Wireless Communication

นางสาวจุฬาทิพย์ สุพรมมา รหัส 51364712

นายยุทธชัย อยู่บำรุง รหัส 51364910

ห้องสมุดคณะวิศวกรรมศาสตร์
วันที่รับ 19 ต.ค. 2558
เลขทะเบียน 1626979
เลขเรียกหนังสือ ฟร.
มหาวิทยาลัยนเรศวร 9678 ว 2554

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมคอมพิวเตอร์ ภาควิชาวิศวกรรมไฟฟ้าและคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร

ปีการศึกษา 2554

ใบรับรองปริญญานิพนธ์

ชื่อหัวข้อโครงงาน	การติดตามรถไฟฟ้าโดยระบบบอกตำแหน่งพิกัดโลกและการสื่อสารแบบไร้สาย
ผู้ดำเนินโครงงาน	นางสาวจุฬาทิพย์ สุพรมมา รหัส 51364712
	นายยุทธชัย อยู่บำรุง รหัส 51364910
อาจารย์ที่ปรึกษา	อาจารย์เศรษฐา ตั้งค้าวานิช
สาขาวิชา	วิศวกรรมคอมพิวเตอร์
ภาควิชา	วิศวกรรมไฟฟ้าและคอมพิวเตอร์
ปีการศึกษา	2554

.....

คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร อนุมัติให้โครงงานฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิศวกรรมศาสตรบัณฑิต สาขาวิชาวิศวกรรมคอมพิวเตอร์

..... ประธานกรรมการ
(อาจารย์ภาณุพงศ์ สอนคม)

..... กรรมการ
(ผู้ช่วยศาสตราจารย์ ดร.พนมขวัญ ริยะมงคล)

..... กรรมการ
(ดร.สุรเดช จิตประไพกุลศาล)

..... กรรมการ
(อาจารย์เศรษฐา ตั้งค้าวานิช)

หัวข้อโครงงาน	**การติดตามรถไฟฟ้าโดยระบบบอกตำแหน่งพิกัดโลกและการสื่อสารแบบไร้สาย**
ผู้ดำเนินโครงงาน	**นางสาวจุฬาทิพย์ สุพรมมา รหัส 51364712**
	นายยุทธชัย อยู่บำรุง รหัส 51364910
อาจารย์ที่ปรึกษา	**อาจารย์เศรษฐา ตั้งค้าวานิช**
สาขาวิชา	**วิศวกรรมคอมพิวเตอร์**
ภาควิชา	วิศวกรรมไฟฟ้าและคอมพิวเตอร์
ปีการศึกษา	2554

บทคัดย่อ

ปริญญานิพนธ์นี้กล่าวถึงการติดตามรถไฟฟ้าโดยระบบบอกตำแหน่งพิกัดโลกและการสื่อสารแบบไร้สาย โดยการนำเทคโนโลยีของระบบบอกตำแหน่งพิกัดโลก (Global Positioning System) มาใช้ในการระบุพิกัดภูมิศาสตร์ของตำแหน่งรถไฟฟ้า จะใช้ค่าพิกัดที่ประมวลผลมาติดต่อสื่อสารแบบไร้สายระหว่างไมโครคอนโทรลเลอร์ฝั่งแม่ข่ายหรือจุดรอรถไฟฟ้ากับไมโครคอนโทรลเลอร์ฝั่งลูกข่ายหรือฝั่งรถไฟฟ้า เพื่อนำพิกัดที่ได้ไปแสดงบนระบบแสดงผล สำหรับการติดตามรถไฟฟ้านั้นจะช่วยเพิ่มประสิทธิภาพในการขนส่งให้มีความทันสมัยและสะดวกสบายแก่ผู้ใช้บริการรถไฟฟ้ามากขึ้น

Project title	EV Tracking via GPS and Wireless Communication		
Name	Miss.Julathip	Supromma	ID.51364712
	Mr.Yutthachai	Yoobamrung	ID.51364910
Project advitor	Mr.Settha	Tangkawanit	
Major	Computer Engineering		
Department	Electrical and Computer Engineer		
Academic Year	2554		

Abstract

EV tracking via GPS and wireless communication studies about tracking electric vehicle by using global positioning system technology and wireless technology for identify electric vehicle's co-ordinate. Then this project lead co-ordinate that processed by microcontroller for communicate between server microcontroller (at electric vehicle station) and client microcontroller (on electric vehicle) for take the co-ordinate to the monitoring system. In conclusion, EV tracking via GPS and wireless communication has a lot advantage such as increase the efficiency transportation in Naresuan university and user can receive convenient service and high technology service.

กิตติกรรมประกาศ

ปริญญานิพนธ์เรื่อง การติดตามรถไฟฟ้าโดยระบบบอกตำแหน่งพิกัดโลกและการสื่อสารแบบไร้สาย (EV Tracking via GPS and Wireless Communication) นี้ สำเร็จลุล่วงไปด้วยดี เนื่องด้วยความอนุเคราะห์จากอาจารย์ที่ปรึกษาโครงงาน อาจารย์เศรษฐา ตั้งค้าวานิช และคณะกรรมการ ดร.สุรเดช จิตประไพกุลศาล ผู้ช่วยศาสตราจารย์ ดร.พนมขวัญ ริยะมงคล และ อาจารย์ภาณุพงศ์ สอนคม ที่ให้คำปรึกษาแนะนำวิธีการในการทำโครงงาน ตลอดจนถึงการตรวจสอบการทำงานพร้อมทั้งเสนอแนวทางการแก้ไขปัญหา ตอลดระยะเวลาการทำโครงงานจนสำเร็จลุล่วง

ขอขอบพระคุณคณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร ที่ให้แนวคิดและอนุเคราะห์ทุนในการวิจัยโครงงานนี้

สุดท้ายนี้ขอกราบขอบพระคุณอาจารย์ทุกท่าน และบุคคลอื่นๆที่ไม่ได้กล่าวถึง ณ ที่นี้ ที่คอยสนับสนุนในการทำโครงงานครั้งนี้

นางสาวจุฬาทิพย์ สุพรมมา
นายยุทธชัย อยู่บำรุง

สารบัญ

หน้า

บทคัดย่อภาษาไทย ..... ก

บทคัดย่อภาษาอังกฤษ ..... ข

กิตติกรรมประกาศ ..... ค

สารบัญ ..... ง

สารบัญตาราง ..... ช

สารบัญรูป ..... ฌ

บทที่ 1 บทนำ

1.1 ที่มาและความสำคัญ 1

1.2 วัตถุประสงค์ของโครงงาน 2

1.3 ประโยชน์ที่คาดว่าจะได้รับ 2

1.4 ขอบเขตการดำเนินโครงงาน 2

1.5 ขั้นตอนการดำเนินโครงงาน 3

1.6 แผนการดำเนินงานตลอดโครงงาน 3

1.7 งบประมาณของโครงงาน 4

บทที่ 2 หลักการและทฤษฎี

2.1 ระบบบอกตำแหน่งพิกัดโลก (Global Positioning System: GPS) 5

2.1.1 ตำแหน่งดาวเทียม 5

2.1.2 ระยะทางที่ได้ระหว่าง Module GPS กับดาวเทียม 5

2.2 ระบบพิกัด (Coordinate System) 6

2.2.1 เส้นละติจูด 6

2.2.2 เส้นลองจิจูด 7

สารบัญ (ต่อ)

2.3 โครงสร้างข้อมูลแบบ NMEA (National Marine Electronics Association)..... 8

2.3.1 โครงสร้างทั่วไปของ NMEA 8

2.3.2 ความหมายแต่ละตัวของมาตรฐาน NMEA..... 9

2.4 การแปลงหน่อยระบบพิกัด 10

2.4.1 เทคนิควิธีการแปลงค่าหน่อยของฮา ลิปดา ฟิลิปดา..... 10

2.5 RS232 to RF-Wireless รุ่น ET-RF24G V2.0 11

2.5.1 การใช้งานเครื่อง ET-RF24G V2.0..... 12

2.5.2 การกำหนดค่า Configuration 15

2.6 การสื่อสารข้อมูลอนุกรม USART ภายใน STM32..... 16

2.6.1 คุณสมบัติทางเทคนิคที่น่าสนใจของโมดูล USART 16

2.6.2 การทำงานเบื้องต้นของโมดูล 18

2.6.3 รูปแบบข้อมูลอนุกรมที่ใช้ใน USART 18

2.6.4 กระบวนการส่งข้อมูลในโมดูล USART 19

2.6.5 การสื่อสารข้อมูลแบบไบต์เดี่ยว 20

2.6.6 กระบวนการรับข้อมูลในโมดูล USART 20

2.6.7 รีจิสเตอร์ที่สำคัญของโมดูล USART ใน STM32F107VC..... 21

2.7 พอร์ตอนุกรม RS232 17

2.7.1 การจัดขาของคอนเน็กเตอร์อนุกรมแบบ DB9 และหน้าที่การใช้งานต่างๆ..23

2.7.2 การเชื่อมต่ออุปกรณ์อุปกรณ์ภายนอกเข้ากับคอมพิวเตอร์ด้วยสาย DB9.... 24

2.7.3 การทางานของขาสัญญาณ DB9 24

2.7.4 รูปแบบการสื่อสารแบบอนุกรม 25

2.7.5 การรับและส่งข้อมูลแบบอนุกรมตามลักษณะการใช้งาน 26

2.8 Multiple Access Protocols 26

สารบัญ (ต่อ)

2.9 Random Access Protocols 27

2.9.1 Multiple Access (MA) 27

2.9.2 Carrier Sense Multiple Access (CSMA) 28

บทที่ 3 วิธีดำเนินโครงงาน

3.1 ระบบตำแหน่งพิกัดโลก 31

3.2 การพัฒนาโครงสร้างฝั่งแม่ข่าย 33

3.3 การพัฒนาโครงสร้างฝั่งลูกข่าย 36

3.4 Data package 39

3.5 ระบบโดยรวมฝั่งลูกข่าย 40

3.6 ระบบโดยรวมฝั่งแม่ข่าย 41

บทที่ 4 ผลการทดลอง

4.1 การทดสอบการส่งข้อมูลผ่านชุดแปลงสัญญาณระหว่างอาร์เอส232 กับคลื่นความถี่ไร้สาย 42

4.1.1 การทดสอบการส่งข้อมูลจากไมโครคอนโทรลเลอร์ผ่านชุดแปลงสัญญาณไปยังคอมพิวเตอร์ 42

4.1.2 การทดสอบความถูกต้องในการส่งข้อมูลของชุดแปลงสัญญาณ 45

4.2 การทดสอบการรับข้อมูลจากโมดูลจีพีเอส 47

4.2.1 กรณีที่ให้คอมพิวเตอร์รับข้อมูลจากโมดูลจีพีเอส 47

4.2.2 กรณีที่ให้ระบบของฝั่งลูกข่ายรับข้อมูลจากจีพีเอส 48

4.3 ขั้นตอนการทดสอบของระบบ 49

4.3.1 ข้อมูลการติดต่อสื่อสารกันระหว่างแม่ข่ายกับลูกข่าย 1 เครื่อง 51

4.3.2 ข้อมูลการติดต่อสื่อสารกันระหว่างแม่ข่ายกับลูกข่าย 2 เครื่อง 52

บทที่ 5 บทสรุปและข้อเสนอแนะ

5.1 สรุปผลการทดลองของโครงงาน 53

สารบัญ (ต่อ)

5.2 ปัญหาในการดำเนินงานและแนวทางการแก้ไขปัญหา 53
5.3 ข้อเสนอแนะในการดำเนินโครงงาน 54

เอกสารอ้างอิง..... 56
ภาคผนวก ก 58
ประวัติผู้เขียนโครงงาน..... 65

สารบัญตาราง

ตารางที่ หน้า

1.1 แสดงขั้นตอนและระยะเวลาการดำเนินงาน 3

3.1 แสดง Data package 39

4.1 แสดงการส่งข้อมูลของชุดแปลงสัญญาณ ในที่โล่งแจ้ง ถนนหน้าตึกวิศวกรรมศาสตร์ถึงทางออกประตู 5 45

4.2 แสดงการส่งข้อมูลของชุดแปลงสัญญาณ ในทางโค้งที่มีต้นไม้ ถนนหน้าตึกวิศวกรรมศาสตร์ถึงหน้าตึก มน.นิเวศ 46

5.1 แสดงปัญหาในการดำเนินงานและแนวทางการแก้ไขปัญหา 53

สารบัญรูป

รูปที่ หน้า

2.1 รูปเส้นละติจูด 7

2.2 แสดงเส้นลองจิจูด 7

2.3 แสดงสายสัญญาณ RS232 ในโหมด RF Receive Only และ RF Transmit Only 12

2.5 แสดงรูปโปรแกรมที่ใช้สาหรับกาหนดค่า Configuration ของ ET-RF24G V2.0 13

2.6 แสดงส่วนประกอบและไดอะแกรมการทางานของโมดูล USART 17

2.7 รูปแบบของข้อมูลที่สัมพันธ์กับบิต M ในรีจิสเตอร์ควบคุม USART 18

2.8 แสดงการใช้งานพอร์ตอนุกรม RS232 22

2.9 พอร์ตอนุกรมของคอมพิวเตอร์ DB9 ตัวผู้ 22

2.10 พอร์ตอนุกรมของอุปกรณ์ภายนอก DB9 ตัวเมีย 23

2.11 แสดงการจัดขาของคอนเน็กเตอร์อนุกรมแบบ DB9 23

2.12 การเชื่อมต่ออุปกรณ์ภายนอกผ่าน DB9 แบบ Null modem 24

2.13 การต่ออุปกรณ์ภายนอกผ่าน DB9 แบบ 3 เส้น 24

2.14 แสดงการสื่อสารอนุกรมแบบซิงโครนัส 25

2.15 แสดงการสื่อสารอนุกรมแบบอะซิงโครนัส 25

2.16 Multiple Access Protocol 26

3.1 ภาพรวมของโครงงาน 30

3.2 โครงสร้างและระบบโดยรวมของโครงงาน 31

3.3 โมดูลของระบบบอกตำแหน่งพิกัดโลกเป็นโมดูลสำเร็จ รุ่น ET-GPS START KIT 31

3.4 แสดงค่าตำแหน่งพิกัดโลกตามโปรโตคอลมาตรฐานที่พัฒนาโดย NMEA 32

3.5 แสดงแผนภาพในการคัดกรองข้อมูลจากระบบบอกตำแหน่งพิกัดโลก 32

3.6 ไมโครคอนโทลเลอร์ตระกูลอาร์ม รุ่น ET-ARM STAMP STM32 33

3.7 ชุดแปลงสัญญาณระหว่างอาร์เอส232 กับคลื่นความถี่ไร้สาย 33

3.8 บอร์ดแปลงไฟจาก 12 โวลต์ เป็น 5 โวลต์ 34

3.9 หม้อแปลงไฟ 220 โวลต์ เป็น 12 โวลต์ 34

3.10 ลักษณะภายในของโครงสร้างฝั่งแม่ข่าย 35

3.11 ลักษณะภายนอกของโครงสร้างฝั่งแม่ข่าย 35

3.12 ไมโครคอนโทลเลอร์ตระกูลอาร์ม รุ่น ET-STM32F103 36

3.13 ชุดแปลงสัญญาณระหว่างอาร์เอส232 กับคลื่นความถี่ไร้สาย 36

3.14 จอแสดงผลแอลซีดี รุ่น 16216H LED BACKLIGHTS 36

3.15 บอร์ดแปลงไฟจาก 12 โวลต์ เป็น 5 โวลต์ 37

3.16 โมดูลของระบบบอกตำแหน่งพิกัดโลก รุ่น ET-GPS START KIT 37

3.17 แบตเตอรี่ลิเธียมโพลิเมอร์ 2200 mA 37

3.18 ลักษณะภายในของโครงสร้างฝั่งลูกข่าย 38

3.19 ลักษณะภายนอกของโครงสร้างฝั่งลูกข่าย 39

3.20 แผนภาพแสดงระบบโดยรวมฝั่งลูกข่าย 40

3.21 แผนภาพแสดงระบบโดยรวมฝั่งแม่ข่าย 41

4.1 ลูกข่ายส่งชุดข้อมูล (Data Package) ให้กับเครื่องคอมพิวเตอร์ 43

4.2 แสดงการส่งชุดข้อมูล (Data Package) ให้กับเครื่องคอมพิวเตอร์ 44

4.3 ลูกข่ายส่งชุดข้อมูล (Data Package) ให้กับเครื่องคอมพิวเตอร์ 44

4.4 แสดงการส่งค่ารหัสลูกข่ายที่ส่งข้อมูลถูกต้องกลับมายังเครื่องคอมพิวเตอร์ 45

4.5 แสดงการรับข้อมูลจากโมดูลจีพีเอสเข้าคอมพิวเตอร์ในที่โล่งแจ้ง 47

4.6 แสดงการรับข้อมูลจากโมดูลจีพีเอสเข้าคอมพิวเตอร์ในอาคาร 48

4.7 แสดงการรับข้อมูลจากโมดูลจีพีเอสเข้าระบบของฝั่งลูกข่าย 48

4.8 แสดงการรับข้อมูลจากโมดูลจีพีเอสเข้าระบบของฝั่งลูกข่าย 49

4.9 แสดงสถานีจอดรถไฟฟ้าป้ายแรก 59

4.10 แสดงสถานีจอดรถไฟฟ้าป้ายที่สอง 50

4.11 แสดงสถานีจอดรถไฟฟ้าป้ายที่สาม 50

4.12 อุปกรณ์ฝั่งลูกข่ายที่ติดตั้งบนรถไฟฟ้าโดยจาลองรถมอเตอร์ไซด์แทนรถไฟฟ้า 51

4.13 การติดต่อสื่อสารกันระหว่างแม่ข่ายกับลูกข่าย 1 เครื่อง 51

4.14 ข้อมูลการติดต่อสื่อสารกันระหว่างแม่ข่ายกับลูกข่าย 1 เครื่อง 52

4.15 การติดต่อสื่อสารกันระหว่างแม่ข่ายกับลูกข่าย 1 เครื่อง 52

4.16 แสดงข้อมูลการติดต่อสื่อสารกันระหว่างแม่ข่ายกับลูกข่าย 2 เครื่อง 53

รูป ก1 โปรแกรมเพื่อคัดกรองข้อมูลจากโมดูลจีพีเอส 57
รูป ก2 โปรแกรมคัดกรองและรวบรวมข้อมูลที่จะเก็บในคาต้าแพ็คเกต 58
รูป ก3 โปรแกรมแสดงข้อมูลบนหน้าจอแอลซีดี 59
รูป ก4 โปรแกรมคำนวณหาระยะทางระหว่างรถกับป้ายทุกป้าย 59
รูป ก5 โปรแกรมคำนวณหาว่ารถกับป้ายไหนมีระยะทางใกล้กว่ากัน 60
รูป ก6 โปรแกรมรวบรวมคาต้าแพ็คเกตและส่ง 60
รูป ก7 โปรแกรมในส่วนของอินเตอร์รัพท์ 61
รูป ก8 โปรแกรมในส่วนของอินเตอร์รัพท์..... 62
รูป ก9 โปรแกรมในส่วนของอินเตอร์รัพท์ 63

บทที่ 1

บทนำ

1.1 ที่มาและความสำคัญของโครงงาน

เนื่องจากปัจจุบันมหาวิทยาลัยนเรศวรมีการบริการรถไฟฟ้าภายในมหาวิทยาลัย จึงมีผู้ใช้เป็นจำนวนมาก ไม่ว่าจะเป็นอาจารย์ นักศึกษาหรือประชาชนทั่วไป โดยผู้ใช้บริการมีความต้องการให้การบริการรถไฟฟ้าภายในมหาวิทยาลัยนั้น มีประสิทธิภาพมากยิ่งขึ้น เพื่อก่อให้เกิดความสะดวกสบายและรวดเร็วแก่ผู้ใช้บริการ ดังนั้นจึงได้มีการพัฒนาระบบการติดตามรถไฟฟ้าโดยระบบบอกตำแหน่งพิกัดโลกและการสื่อสารแบบไร้สายเกิดขึ้น

ในการติดตามรถไฟฟ้าโดยระบบบอกตำแหน่งพิกัดโลกและการสื่อสารแบบไร้สายนั้นได้นำเทคโนโลยีของระบบที่ใช้ระบุตำแหน่งบนพื้นโลก(Global Positioning System: GPS) มาคำนวณหาค่าละติจูด ลองติจูด โดยอาศัยการคำนวณจากกความถี่ของสัญญาณนาฬิกาที่ส่งมาจากดาวเทียม ทั้ง 24 ดวง ที่โคจรอยู่รอบโลก มาร่วมใช้กับไมโครคอนโทรลเลอร์ ARM Cortex M3 ในการพัฒนาโครงงานการติดตามรถไฟฟ้าโดยระบบบอกตำแหน่งพิกัดโลกและการสื่อสารแบบไร้สาย

การพัฒนาโครงงานนี้ใช้โมดูลระบบบอกตำแหน่งบนพื้นโลก(Module GPS) ของ ETT รุ่น ET-GPS START KIT เพื่อใช้ในการรับค่าตำแหน่งบนพื้นโลกจากโมดูลระบบบอกตำแหน่งบนพื้นโลก แล้วนำข้อมูลที่ได้จากโมดูลนั้น มาคัดกรองข้อมูลเอาเฉพาะค่าละติจูด ลองติจูด โดยใช้ไมโครคอนโทรลเลอร์ในการคัดสตริง เมื่อได้ข้อมูลที่ต้องการแล้ว ก็ส่งต่อให้กับไมโครคอนโทรลเลอร์อีกตัวเพื่อนำค่าที่ได้ไปแสดงผล ณ ป้ายต่าง ๆ ต่อไป สำหรับการพัฒนาโครงงานนี้ได้ทำการทดลองและรันโปรแกรมบน ระบบปฏิบัติการวินโดว์ 7 โดยใช้โปรแกรม IAR embedded work bench IDE ในการพัฒนาโปรแกรมด้วยภาษาซี

สำหรับผลที่คาดว่าจะได้รับจากการทำโครงงานนี้คือ โปรแกรมนี้สามารถใช้งานได้จริง มีความสะดวก สบายแก่ผู้ใช้บริการรถไฟฟ้าและคาดว่าโครงงานนี้จะเป็นประโยชน์อย่างมากสำหรับผู้ที่จะมาพัฒนาโครงงานนี้ต่อไป

1.2 วัตถุประสงค์ของโครงงาน

1. เพื่อสร้างตัวต้นแบบของระบบบอกตำแหน่งพิกัดโลกของรถไฟฟ้า ภายในมหาวิทยาลัยนเรศวร
2. เพื่อพัฒนาระบบขนส่งภายในมหาวิทยาลัย ให้มีประสิทธิภาพมากยิ่งขึ้น
3. เพื่อจัดทำโปรแกรมในการหาค่าตำแหน่งบนพื้นโลก (ละติจูด ลองติจูด) จากโมดูลระบบบอกตำแหน่งบนพื้นโลก

1.3 ประโยชน์ที่คาดว่าจะได้รับ

1. ระบบติดตามรถไฟฟ้าจะช่วยบอกตำแหน่งของรถไฟฟ้าที่ให้บริการอยู่
2. ระบบติดตามรถไฟฟ้าจะช่วยให้ผู้ใช้บริการสามารถคาดการณ์เวลาในการรอใช้บริการได้

1.4 ขอบเขตการดำเนินโครงงาน

1. ระบบนี้สามารถอ่านค่าละติจูด ลองติจูด จากโมดูลระบบบอกตำแหน่งบนพื้นโลกได้
2. ระบบนี้สามารถทำงานได้ในที่โล่งแจ้งและในสภาวะอากาศที่ปลอดโปล่งได้เท่านั้น
3. ระบบนี้สามารถแสดงผลละติจูด ลองติจูด ผ่านจอแอลซีดี (LCD)ได้
4. ระบบนี้สามารถส่งข้อมูลแบบไร้สายไปยังเสารับสัญญาณที่อยู่ใกล้ได้ ในระยะทาง 100 เมตร
5. ระบบนี้สามารถบ่งบอกตัวตนของรถไฟฟ้าที่ติดตั้งระบบอยู่ได้
6. ระบบนี้สามารถรับส่งข้อมูล รหัสรถไฟฟ้า ละติจูดและลองติจูด จากรถไฟฟ้าไปยังป้ายจอดรถไฟฟ้าได้
7. ระบบนี้เป็นระบบต้นแบบ ที่จำลองการทำงานของรถไฟฟ้า 2 คัน ที่ติดต่อกับป้ายแสดงผล 3 ป้าย

1.5 ขั้นตอนการดำเนินโครงงาน

1. ศึกษาและรวบรวมข้อมูลเกี่ยวกับระบบบอกตำแหน่งบนพื้นโลก
2. ศึกษาการใช้ไมโครคอนโทรลเลอร์ ARM Cortex M3 รุ่น ET-STM32F103
3. ศึกษาการส่งข้อมูลแบบไร้สายระหว่างไมโครคอนโทรลเลอร์กับไมโครคอนโทรลเลอร์
4. จัดทำโปรแกรมอ่านค่าละติจูด ลองติจูดจากโมดูลระบบบอกตำแหน่งบนพื้นโลกแล้วส่งข้อมูลแบบไร้สาย

5. ทำการทดลองและแก้ไขปัญหาที่เกิดขึ้น
6. สรุปงานและจัดทำรายงาน

1.6 แผนการดำเนินงานตลอดโครงงาน

ตารางที่ 1.1 แสดงขั้นตอนและระยะเวลาการดำเนินงาน

กิจกรรม	ปี 2554							ปี 2555		
	มิ.ย.	ก.ค.	ส.ค.	ก.ย.	ต.ค.	พ.ย.	ธ.ค.	ม.ค.	ก.พ.	มี.ค.
1. ศึกษาและรวบรวมข้อมูลเกี่ยวกับระบบบอกตำแหน่งบนพื้นโลก	←	→								
2. ศึกษาการใช้ไมโครคอนโทรลเลอร์ ARM Cortex M3 รุ่น ET-STM32F103		←	→							
3. ศึกษาการส่งข้อมูลแบบไร้สายระหว่างไมโครคอนโทรลเลอร์กับไมโครคอนโทรลเลอร์			←	→						
4. จัดทำโปรแกรมอ่านค่าละติจูด ลองติจูดจากโมดูลระบบบอกตำแหน่งบนพื้นโลกแล้วส่งข้อมูลแบบไร้สาย				←	—	→				
5. ทำการทดลองและแก้ไขปัญหาที่เกิดขึ้น						←	—	→		
6. สรุปงานและจัดทำรายงาน									←	— →

1.7 รายละเอียดงบประมาณของโครงงาน

1. ค่าอุปกรณ์อิเล็กทรอนิกส์	1,500	บาท
2. ค่าจัดทำรูปเล่ม	500	บาท
รวมเป็นเงินทั้งสิ้น	2,000	บาท
	(สองพันบาทถ้วน)	

หมายเหตุ ถัวเฉลี่ยทุกรายการ

บทที่ 2
ทฤษฎีและหลักการที่เกี่ยวข้อง

2.1 ระบบบอกตำแหน่งพิกัดโลก(Global Positioning System: GPS) [1]

ระบบบอกตำแหน่งพิกัดโลก คือการติดต่อสื่อสารจากดาวเทียม เพื่อหาตำแหน่งและคำนวณระยะทางระหว่างดาวเทียมและตัว Module GPS โดยการที่จะสามารถทราบถึงตำแหน่งนั้น จะสามารถทำได้เมื่อมีการรับค่าจากดาวเทียมอย่างน้อย 3 ดวงขึ้นไป แล้วหาจุดตัดที่เกิดจากดาวเทียมเพื่อหาตำแหน่งผิวทรงกลม (พื้นโลก) ในทางทฤษฎี สิ่งที่ Module GPS จำเป็นต้องทราบในการคำนวณหาตำแหน่งแต่ละครั้ง คือ

- ตำแหน่ง ดาวเทียม ที่รับค่าได้ อย่างน้อย 3 ดวง (ปัจจุบัน มี 24 ดวง)
- ระยะห่างจาก ดาวเทียม ถึง GPS แต่ละดวงเพื่อที่จะสามารถคำนวณหาจุดตัดได้

2.1.1 ตำแหน่งของดาวเทียม

ตำแหน่งดาวเทียม GPS ในอวกาศ จะต้องมีข้อมูลประกอบ 2 ตัว คือ

1. ข้อมูลวงโคจร จะทำให้ module GPS ทราบว่า เส้นทางการเดินทางของดาวเทียม GPS แต่ละดวงจะอยู่ ณ ตำแหน่งใด เมื่อใด
   - โดยทั่วไป วงโคจรดาวเทียมสูงประมาณ 11,000 ไมล์
   - วงโคจรอาจคลาดเคลื่อน (Ephemeris Errors) เนื่องจากแรงโน้มถ่วงของดวงจันทร์และดวงอาทิตย์
2. เวลาปัจจุบัน ซึ่งเมื่อ module GPS ทราบเวลาปัจจุบันแล้วก็จะใช้เวลาปัจจุบัน ไปคำนวณหาตำแหน่ง ของดาวเทียม GPS จากข้อมูลวงโคจรได้

ดังนั้น เมื่อ GPS ทราบ ข้อมูลวงโคจรดาวเทียมและเวลาปัจจุบัน module GPS ก็จะทราบตำแหน่ง ดาวเทียมในอวกาศได้ ซึ่งข้อมูลทั้งหมดจะได้มาจากสัญญาณดาวเทียมที่อุปกรณ์รับ GPS ตัวนั้นรับได้

2.1.2 ระยะทางที่ได้ระหว่าง Module GPS กับดาวเทียม

คลื่นสัญญาณ GPS นั้น เคลื่อนที่ด้วยความเร็วคงที่ (V) คือ ความเร็วแสง (186,000ไมล์ต่อวินาที) ซึ่งเมื่อเป็นดังนั้น ถ้าอุปกรณ์รับ GPS รู้ระยะเวลา (T) ที่สัญญาณใช้ในการเดินทางจาก ดาวเทียม GPS มายังอุปกรณ์รับ GPS ก็จะสามารถคำนวณระยะทางระหว่าง ดาวเทียม GPS กับ อุปกรณ์รับ GPS ได้ จากสูตร

$$S = VT \text{ คือ ระยะทาง} = \text{ความเร็ว} \times \text{เวลา} \qquad (2.1)$$

เมื่อเราทราบระยะห่างของดาวเทียมกับอุปกรณ์ GPS เราก็จะหาจุดของผิวทรงกลม ทำให้อุปกรณ์ GPS สามารถทราบว่าตัวเองอยู่ณะจุดใดบนพื้นโลกได้ เช่น

ดาวเทียม ลอยอยู่ ณ จุดหนึ่งในอวกาศ ซึ่งเรารู้ตำแหน่ง จากข้อมูลวงโคจร GPS และ เวลาปัจจุบัน ระยะเวลาในการส่งสัญญาณจากดาวเทียมดวงที่ 1 ถึงเครื่องรับ GPS คือ 0.20 วินาที ระยะทางระหว่างดาวเทียมกับเครื่องรับ GPS คือ 37,200 ไมล์ (186,000 ไมล์ต่อวินาที X 0.20 วินาที = 37,200 ไมล์) เนื่องจากการเดินทางของคลื่น GPS นั้นจะเคลื่อนที่ด้วยความเร็วคงที่ คือ ความเร็วแสง 186,000 ไมล์ต่อวินาที

ดังนั้น ตำแหน่งปัจจุบัน ของเครื่องรับ GPS ก็จะสามารถเป็นจุดใดๆ ก็ได้ บนผิวทรงกลมที่มีรัศมี 37,200 ไมล์

จากตัวอย่างดังกล่าว การที่เราจะสามารถทราบจุดที่แน่นอน บนพื้นโลกนั้น เราจะคำนวณจากดาวเทียมอย่างน้อย 3 ดวง แล้ว ทำรัศมีจากการคำนวณที่ได้ มาลากจุดตัดกัน แล้วเราก็จะได้พิกัดบนพื้นโลกที่แน่นอน

2.2 ระบบพิกัด (Coordinate System) [2]

เป็นระบบที่สร้างขึ้นเพื่อใช้ในการอ้างอิง ตำแหน่งบนพื้นโลก โดยสร้างเส้นขึ้นมา 2 แนว คือ แนวตั้งและแนวนอน ค่าพิกัดที่ใช้อ้างอิงในการบอกตำแหน่งต่างๆ จะใช้ค่าของหน่วยที่นับออกจากจุดศูนย์กำเนิดเป็นระยะเชิงมุม (Degree) หรือเป็นระยะทาง (Distance) ไปทางเหนือหรือใต้และตะวันออกหรือตะวันตก ตามตำแหน่งของตำบลที่ต้องการหาค่าพิกัดที่กำหนดตำแหน่งต่างๆ จะถูกเรียกอ้างอิงเป็นตัวเลขในแนวตั้งและแนวนอนตามหน่วยวัดระยะใช้วัดสำหรับระบบพิกัดที่ใช้อ้างอิงกำหนดตำแหน่งบนแผนที่ ที่นิยมใช้กับแผนที่ในปัจจุบัน มีอยู่ด้วยกัน 2 ระบบ คือ

1. ระบบพิกัดภูมิศาสตร์ (Geographic Coordinate System)
2. ระบบพิกัดกริดแบบ UTM (Universal Transverse Mercator co-ordinate System)

สำหรับระบบ ที่จะนำมาใช้ คือ ระบบพิกัดภูมิศาสตร์

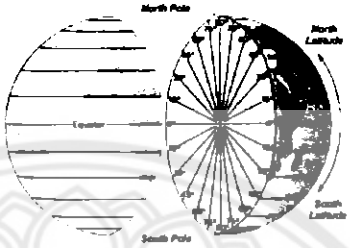
2.2.1 เส้นละติจูด(Latitude)

เส้นละติจูด หรือ เส้นรุ้ง คือ เส้นสมมติที่ลากรอบโลก โดยลากขนานกับเส้นศูนย์สูตร อยู่ระหว่างขั้วโลกเหนือและขั้วโลกใต้ โดยกำหนดว่า

1. เส้นศูนย์สูตร(Equator) คือเส้นละติจูดที่ 0 องศา เป็นเส้นละติจูดที่ยาวที่สุดที่ลากผ่านทิศตะวันออก และทิศตะวันตก

2. ทางซีกโลกเหนือมีค่าละติจูดตั้งแต่ 0-90 องศา เหนือ

3. ทางซีกโลกใต้มีค่าละติจูดตั้งแต่ 0-90 องศา ใต้



รูปที่ 2.1 รูปเส้นละติจูด [3]

2.2.2 เส้นลองจิจูด(Longitude)

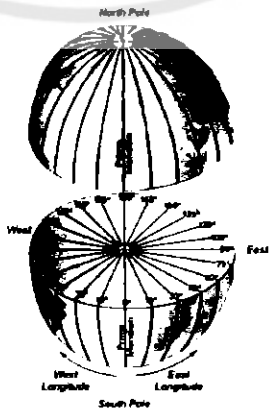
เส้นลองจิจูด หรือ เส้นแวง คือ เส้นสมมติที่ลากตั้งฉากกับเส้นศูนย์สูตรไปผ่านขั้วโลกเหนือและขั้วโลกใต้ โดยกำหนดว่า

1. เส้นลองจิจูดที่ลากผ่านหอดูดาวเมืองกรีนิชประเทศอังกฤษ เป็นเส้นลองจิจูดที่ 0 องศา เรียกว่าเส้นไพร์มเมอริเดียน(Primeridian) ซึ่งเป็นเส้นที่กำหนดเวลามาตรฐานของโลก

2. ค่าลองจิจูดจะบอกเป็นองศาตะวันออกและองศาตะวันตกโดยยึดเส้นไพร์มเมอริเดียนเป็นหลักดังนี้

2.1 เส้นลองจิจูดที่อยู่ทางทิศตะวันออกของเส้นลองจิจูดที่ มีค่า 0-90 องศา ตะวันออก

2.2 เส้นลองจิจูดที่อยู่ทางทิศตะวันตกของเส้นลองจิจูดที่ มีค่า 0-90 องศา ตะวันตก



รูปที่ 2.2 แสดงเส้นลองจิจูด [3]

2.3 โครงสร้างข้อมูลแบบ NMEA (National Marine Electronics Association) [4]

เป็นโครงสร้างข้อมูลที่ใช้สำหรับการโอนถ่ายข้อมูลระหว่างเครื่องรับ GPS กับอุปกรณ์ต่อพ่วงชนิดอื่น โดยเครื่องรับ GPS ส่วนใหญ่จะใช้มาตรฐานข้อมูลแบบ NMEA 0183

ตัวอย่างข้อมูล

$GPGGA,092750.000,5321.6802,N,00630.3372,W,1,8,1.03,61.7,M,55.2,M,,*76

$GPGSA,A,3,10,07,05,02,29,04,08,13,,,,,1.72,1.03,1.38*0A

$GPGSV,3,1,11,10,63,137,17,07,61,098,15,05,59,290,20,08,54,157,30*70

$GPGSV,3,2,11,02,39,223,19,13,28,070,17,26,23,252,,04,14,186,14*79

$GPGSV,3,3,11,29,09,301,24,16,09,020,,36,,,*76

$GPRMC,092750.000,A,5321.6802,N,00630.3372,W,0.02,31.66,280511,,,A*43

$GPGGA,092751.000,5321.6802,N,00630.3371,W,1,8,1.03,61.7,M,55.3,M,,*75

$GPGSA,A,3,10,07,05,02,29,04,08,13,,,,,1.72,1.03,1.38*0A

$GPGSV,3,1,11,10,63,137,17,07,61,098,15,05,59,290,20,08,54,157,30*70

$GPGSV,3,2,11,02,39,223,16,13,28,070,17,26,23,252,,04,14,186,15*77

$GPGSV,3,3,11,29,09,301,24,16,09,020,,36,,,*76

$GPRMC,092751.000,A,5321.6802,N,00630.3371,W,0.06,31.66,280511,,,A*45

2.3.1 โครงสร้างทั่วไปของ NMEA

โครงสร้างข้อมูลทั่วไปของ NMEA จะประกอบไปด้วย

$ ttsss, df1,df2..... **[Carriage Return][Line Feed]**

- แต่ละข้อความเริ่มต้นด้วย $
- ตัวอักษร 5 ตัวที่ตามหลัง $ หมายถึง address field
- ข้อมูลในแต่ละ data field คั่นด้วย commas
- Data field ตัวสุดท้ายเป็น Check sum (optional) สำหรับตรวจสอบความถูกต้อง
  - tt บอกชนิดของเครื่องรับ หากเป็น GPS ใช้ตัวอักษรเริ่มต้นเป็น GP
  - sss ชนิดของประโยค เช่น

    $GPGGA: Global positioning system fixed data

    $GPRMC: Recommended minimum specific GPS data

2.3.2 ความหมายแต่ละตัวของมาตรฐาน NMEA

AAM – สัญญาณเตือนเมื่อถึง Waypoint

ALM – Almanac data

APA – เป็นส่วนที่ใช้นำทางนักบินอัตโนมัติ

APB – Auto Pilot B sentence

BOD – ความสัมพันธ์จากจุดต้นสู่จุดปลาย

BWC – ความสัมพันธ์จากเส้นศูนย์สูตร

DTM – สถิติข้อมูลที่นำไปใช้

GGA – ข้อมูลที่กำหนดไว้

GLL – ข้อมูลละติจูด ลองติจูด

GSA – ข้อมูลดาวเทียมทั้งหมด

GSV – ข้อมูลรายละเอียดของดาวเทียม

MSK – ควบคุมการรับส่งสัญญาณ

MSS – แสดงสถานะของสัญญาณที่รับมา

RMA – แนะนำ Loran data

RMB – แนะนำข้อมูล GPS สำหรับการเดินเรือ

RMC – แนะนำข้อมูลขั้นต่ำของ GPS

RTE – ข้อมูลของเส้นทาง

VTG – เวกเตอร์ติดตามความเร็วบนพื้นดิน

WCV – อัตราความเร็วปิดของ Waypoint(อัตราความเร็วที่ดี)

WPL – ข้อมูลที่ตั้ง Waypoint

XTC – ข้อผิดพลาดของทางที่ตัดกัน

XTE – ข้อผิดพลาดของทางที่ตัดกันที่ได้ตรวจสอบว่าถูกต้องแล้ว

ZTG – เวลาของ zulu และเวลาที่จะไป(ถึงปลายทาง)

ZDA – วันที่และเวลา

HCHDG – ผลที่ได้รับจากเข็มทิศดิจิตอล

PSLIB – การควบคุม DGPS จากระยะไกล

การนำค่าที่รับมาจากโมดูล GPS ไปใช้งานนั้น ก็สามารถเลือกค่าที่ต้องการใช้ได้เลย เช่น

$GPRMC,123519,A,4807.038,N,01131.000,E,022.4,084.4,230394,003.1,W*64

โดยที่:

- RMC คือ การแนะนำข้อมูลขั้นต่ำของ GPS
- 123519 คือ เวลา 12:35:19 UTC
- A คือ สถานะทำงานอยู่ ถ้าเป็น V คือสถานะไม่ทำงาน
- 4807.038,N คือ ละติจูด 48 องศา 07.038 เหนือ
- 01131.000,E คือ ลองติจูด 11 องศา 31.000 ตะวันออก
- 022.4 คือ ความเร็วเหนือพื้นดิน 0.224 Knots
- 084.4 คือ ติดตามในมุมองศาจริง
- 230394 คือ วันเดือนปี
- 003.1 คือ รูปแบบแม่เหล็ก
- W*64 คือ Check sum สำหรับตรวจสอบความถูกต้องของข้อมูล

2.4 การแปลงหน่วยระบบพิกัด [5]

การแสดงพิกัดบนเครื่อง GPS (Global Positioning System) โดยทั่วไป จะนิยมใช้แค่สองระบบเท่านั้น คือ พิกัดภูมิศาสตร์ และพิกัดกริด UTM (Universal Transverse Mercator) การอ่านค่าในระบบพิกัด UTM นั้นไม่ยุ่งยากเท่าไหร่ เพราะอ่านตัวเลขตามค่า East (ค่า X) และ ค่า North (ค่า Y) และหน่วยของ UTM เป็นเมตรอยู่แล้ว แต่การอ่านค่าระบบพิกัดภูมิศาสตร์นั้นค่อนข้างยุ่งยากเล็กน้อย เพราะเครื่อง GPS บางรุ่น บางยี่ห้อแสดงค่าพิกัดภูมิศาสตร์ในหน่วยแบบที่เรียกว่า องศา ลิปดา ฟิลิปดา (DMS : Degree Minute Second) หรือแสดงเป็นหน่วยในระบบพิกัดแบบค่าตัวเลขทศนิยม (DD : Decimal Degree) เพื่อนำไปใช้ในคอมพิวเตอร์ ดังนั้นเมื่อถ้าต้องการใช้งานแบบใดแบบหนึ่ง จึงต้องมีการแปลงค่าหน่วย DMS เป็น DD หรือ DD เป็น DMS

2.4.1 เทคนิควิธีการแปลงค่าหน่วยองศา ลิปดา ฟิลิปดา

ค่าองศา (Degree) 1 องศา มี 60 ลิปดา

ค่าลิปดา (Minute) 1 ลิปดา มี 60 ฟิลิปดา

ฟิลิปดา (Second) 1 ฟิลิปดา มีค่าระยะทางประมาณ 30.48 ม. หรือ 100 ฟุต บริเวณศูนย์สูตร

1. แปลงจาก DMS เป็น DD โดยสูตร

$$DD = Degrees+(Minutes*60+Seconds)/3600 \quad (2.2)$$

2. แปลงจาก DD เป็น DMS

2.1 จากค่าตัวเลขพิกัดในรูปแบบ DD ตัวอย่างเช่น 100.45416 ตัวเลขก่อนหน้าจุดทศนิยม จะเป็นค่า ของหน่วยองศา ในที่นี่คือ 100 องศา

2.2 ให้นำตัวเลขหลังทศนิยมคูณด้วย 60 เช่น .45416 x 60 = 27.2496

2.3 จากค่าที่คำนวณได้27.2496ตัวเลขก่อนหน้าจุดทศนิยมเป็นค่าของหน่วยลิปดาในที่นี่คือ 27 ลิปดา

2.4 ให้นำตัวเลขหลังทศนิยมจากผลคูณในข้อ 2 คูณด้วย 60 เช่น .2496 x 60 = 14.976

2.5 จากค่าที่คำนวณได้ 14.976 ตัวเลขก่อนหน้าจุดทศนิยม จะเป็นค่าของหน่วยฟิลิปดา ในที่นี่ปัดทศนิยมเป็น 15 ฟิลิปดา

2.6 เมื่อนำตัวเลขมาอ่านรวมกันจะได้ 100 องศา 27 ลิปดา 15 ฟิลิปดา

2.5 RS232 to RF-Wireless รุ่น ET-RF24G V2.0 [6]

ET-RF24G V2.0 เป็นชุดแปลงสัญญาณสำหรับใช้แปลงสัญญาณระหว่าง RS232 และ RF-Wireless โดยโหมดการทำงานของการสื่อสารมี 2 โหมดคือโหมดการทำงานแบบการส่งข้อมูล โดยจะทำหน้าที่รอรับข้อมูลจากพอร์ตสื่อสารอนุกรม RS232จากขา RX แล้วแปลงเป็นสัญญาณความถี่(GFSK) ส่งออกไปในอากาศ และโหมดการทำงานแบบรับข้อมูล ก็จะทำหน้าที่คอยตรวจจับข้อมูลที่อยู่ในรูปของสัญญาณความถี่จากด้าน RF เพื่อแปลงกลับเป็นข้อมูลแบบ RS232 ส่งออกไปทางขา TX ได้ด้วย จะเห็นได้ว่า ET-RF24G V2.0 นั้น สามารถนำไปต่อใช้งานร่วมกับพอร์ตสื่อสารอนุกรม แบบ RS232 เพื่อใช้งานในลักษณะของการสื่อสารอนุกรมแบบไร้สายได้โดยตรง

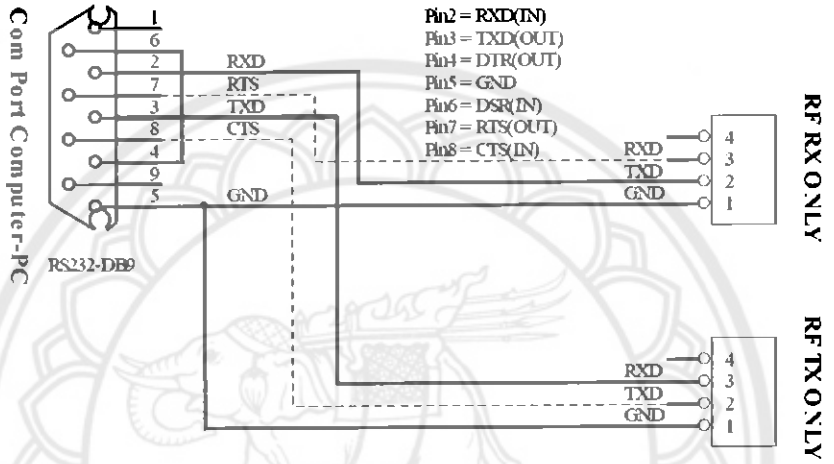
การทำงานใน Run Mode มี 3 แบบ ด้วยกันคือ

- RF Receive Only เป็นการกำหนดให้ ET-RF24G V2.0 ทำหน้าที่เป็นฝ่ายรอรับข้อมูลทางด้าน RF เพื่อเปลี่ยนเป็นข้อมูลแบบ RS232 และส่งออกไปทางด้านขา TX ของ RS232 ตลอดเวลา

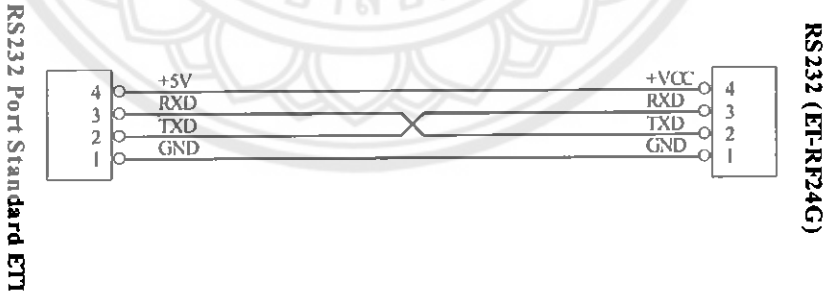
- RF Transmit Only เป็นการกำหนดให้ ET-RF24G V2.0 ทำหน้าที่เป็นฝ่ายรอรับข้อมูล จากขา RX ทางด้านRS232 เพื่อเปลี่ยนเป็นข้อมูลความถี่แบบ GFSK และส่งออกไปทางด้าน RF ตลอดเวลา โดยการใช้งานโหมดนี้ จะต้องนำขา TX (ขาส่งข้อมูลจาก RS232) ของอุปกรณ์ตรงข้ามมาต่อกับ RX ของเครื่อง ET-RF24G V2.0

- RF Auto Direction เป็นการกำหนดโหมดการทำงานแบบ Half Duplex 2 ทิศทาง (ผลัดกันรับ-ส่งข้อมูล) ซึ่งสามารถสลับโหมดการทำงานระหว่างการรับและส่งข้อมูลได้เองโดยอัตโนมัติ โดยในโหมดการทำงานนี้ เครื่อง ET-RF24G V2.0 จะรอตรวจสอบข้อมูลทั้งจากด้าน RS232 และด้าน RF อยู่

ตลอดเวลา โดยถ้าได้รับข้อมูลจากด้าน RS232 ก็จะรับข้อมูลจาก RS232 พร้อมกับเปลี่ยน RF จากรับไปเป็นส่งข้อมูลแทน (เพื่อส่งข้อมูลที่ได้รับจาก RS232 ออกไปทาง RF ในทันที) จากนั้นก็จะกำหนดให้ด้าน RF กลับมาเป็นฝ่ายรอรับข้อมูลตามเดิม และเมื่อได้รับข้อมูลจากด้าน RF ก็จะนำข้อมูลนั้นส่งออกไปทางด้าน RS232 โดยทันที



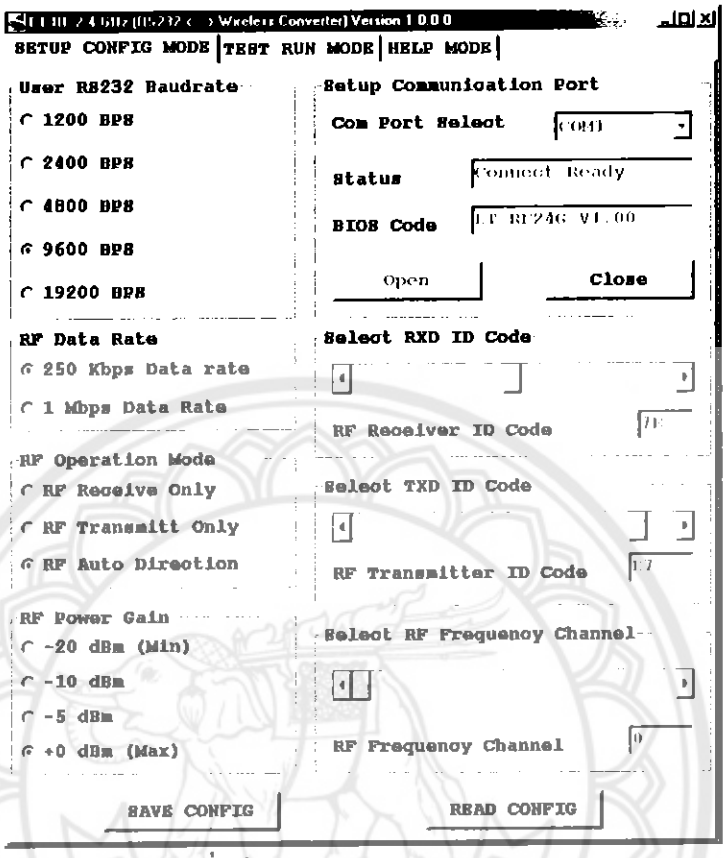
รูปที่ 2.3 แสดงสายสัญญาณ RS232 ในโหมด RF Receive Only และ RF Transmit Only [6]



รูปที่ 2.4 แสดงสายสัญญาณ RS232 ในโหมด Auto Direction [6]

2.5.1 การใช้งานเครื่อง ET-RF24G V2.0 ใน Setup Mode

เป็นโหมดสำหรับใช้กำหนดค่า Configuration ต่างๆ ที่จะใช้ในขณะที่เครื่องทำงานอยู่ใน Run Mode คุณสมบัติของ Configuration ต่างๆนั้นมีดังนี้



รูปที่ 2.5 แสดงรูปโปรแกรมที่ใช้สำหรับกำหนดค่า Configuration ของ ET-RF24G V2.0 [6]

User RS232 Baudrate ใช้สำหรับกำหนดค่าความเร็วในการรับส่งข้อมูลทางด้าน RS232 ของตัวเครื่อง ในขณะที่ทำงานอยู่ใน Run Mode ซึ่งสามารถกำหนดได้ 5 ค่าคือ

- 1200 BPS
- 2400 BPS
- 4800 BPS
- 9600 BPS
- 19200 BPS

RF Data Rate ใช้สำหรับกำหนดความเร็วในการรับส่งข้อมูลทางด้าน RF ของ ET-RF24G V2.0 ซึ่งค่าอัตราความเร็วในการส่งข้อมูลนี้จะมีผลต่อระยะทางการรับส่งข้อมูล ถ้าใช้ความเร็วในการส่งสูง (1Mbps) จะทำให้รัศมีการรับส่งข้อมูลได้ระยะทางสั้นลง แต่ถ้าใช้ความเร็วในการรับส่งข้อมูลที่ช้าลง (250Kbps) จะทำให้ได้รัศมีการรับส่งไกลขึ้น โดยค่า RF Data Rate สามารถกำหนดได้ 2 ค่า คือ

- 250 Kbps

- 1 Mbps

RF Operation Mode ใช้สำหรับกำหนดโหมดการทำงานของ ET-RF24G V2.0 ซึ่งสามารถกำหนดหน้าที่การทำงานได้ 3 แบบ ด้วยกันคือ

- RF Receive Only เป็นการกำหนดให้ ET-RF24G V2.0 ทำหน้าที่เป็นฝ่ายรอรับข้อมูลอย่างเดียว

- RF Transmit Only เป็นการกำหนดให้ ET-RF24G V2.0 ทำหน้าที่เป็นฝ่ายรอรับข้อมูลอย่างเดียว

- RF Auto Direction เป็นการกำหนดโหมดการทำงานแบบ Half Duplex 2 ทิศทาง ซึ่งสามารถสลับโหมดการทำงานระหว่างการรับและส่งข้อมูลได้เองโดยอัตโนมัติ

RF Power Gain เป็นการกำหนดกำลังส่งของวงจร RF Power ที่ใช้ในการส่งข้อมูล โดยค่า +0dBm เป็นค่ากำลังส่งสูงสุด ส่วน –20dBm เป็นค่ากำลังส่งต่ำสุด โดยสามารถกำหนดได้ 4 ระดับคือ

- -20dBm (กำลังส่งต่ำสุด)
- -10dBm
- -5dBm
- +0dBm (กำลังส่งสูงสุด)

RXD ID Code เป็นรหัส ID Code ของเครื่อง ET-RF24G V2.0 ในโหมดของการรับข้อมูลจากRF โดยเมื่อเครื่อง ET-RF24G V2.0 ด้านส่งจะทำการส่งข้อมูลออกไปทาง RF นั้นจะมีการระบุหมายเลข ID Code ของด้านรับรวมไปกับชุดข้อมูลด้วยเสมอ โดยเมื่อเครื่อง ET-RF24G V2.0 ที่อยู่ทางด้านรับทำ การรับข้อมูลจากด้าน RF ได้ อันดับแรกมันจะทำการเปรียบเทียบรหัส ID Code ที่รวมมากับข้อมูลที่ รับมาได้ว่าตรงกับรหัสของ RXD ID Code ที่กำหนดไว้ในตัวมันหรือไม่ ซึ่งถ้าถูกต้องก็จะแยกเอา เฉพาะส่วนของข้อมูลที่รับเข้ามาได้เพื่อเปลี่ยนเป็นข้อมูลแบบ RS232 แล้วส่งออกไปทางด้าน TX ของ RS232 แต่ถ้ารหัส ID Code ที่รับมาได้ไม่ตรงกับรหัส RXD ID Code ที่กำหนดไว้ เครื่อง ET-RF24G V2.0 จะทิ้งข้อมูลชุดนั้นไปทันที โดยค่า RXD ID Code นั้นสามารถกำหนดได้ 256 ค่าในรูปแบบของเลขฐานสิบหก (00H-FFH)

TXD ID Code เป็นรหัส ID Code ปลายทางที่จะส่งข้อมูลไปหาโดยที่เครื่อง ET-RF24G V2.0 ที่ถูกกำหนดให้ทำหน้าที่เป็นฝ่ายส่งข้อมูลนั้น เมื่อมันสามารถรับข้อมูลจาก RS232 ได้แล้ว มันจะทำการนำเอาข้อมูลนั้นไปเข้ารหัสรวมกับ TXD ID Code ที่กำหนดไว้ แล้วส่งออกไปทางด้าน RF โดยรหัสของ TXD ID Code นี้หมายถึง รหัส RXD ID Code ของฝ่ายรับที่ต้องการส่งข้อมูลไปหานั่นเองโดยค่า

TXD ID Code นั้นสามารถกำหนดได้ 256 ค่าในรูปแบบของเลขฐานสิบหก (00H-FFH)

RF Frequency Channel เป็นการกำหนดค่าของช่องความถี่ที่จะใช้ในการรับส่งข้อมูลกัน โดยสามารถเลือกกำหนดช่องความถี่ได้สูงสุดมากถึง 125 ช่อง (0-124)

2.5.2 การกำหนดค่า Configuration

- ความเร็วในการรับส่งข้อมูลด้าน RS232 หรือ User RS232 Baudrate ที่ความเร็ว 19200 Bps นั้นเหมาะกับการใช้งาน ET-RF24G V2.0 แบบ Receive Only หรือ Transmit Only ซึ่งมีการตรวจสอบความพร้อมของสัญญาณในการรับส่งข้อมูลกันด้วย แต่ถ้าต้องการใช้งานเครื่อง ET-RF24G V2.0 ในโหมด Auto Direction นั้น ควรกำหนดค่า User RS232 Baudrate ไว้ที่ความเร็วไม่เกิน 9600 Bps จะดีที่สุด และควรกำหนดค่า Baudrate ของทั้งสองฝ่ายให้มีค่าเท่ากัน

- ค่าความเร็วของการรับส่งข้อมูลด้าน RF หรือ RF Data Rate ที่สามารถรับส่งข้อมูลกันได้ระยะทางไกลมากที่สุด และมีโอกาสผิดพลาดน้อยที่สุด คือ 250Kbps

- ค่า RF Power Gain ที่ดีที่สุดคือ 0dBm ซึ่งเป็นค่ากำลังส่งสูงสุด ซึ่งจะทำให้สามารถส่งข้อมูลได้ระยะทางไกลที่สุด แต่ถ้าระยะการรับส่งข้อมูลไม่ไกลกันมาก และมีการใช้งานเครื่อง ET-RF24G V2.0จำนวนหลายๆกลุ่มในพื้นที่ใกล้เคียงกัน ก็อาจทำการลดกำลังส่งให้ต่ำลงเพื่อลดปัญหาการรบกวนกันหรือกำหนดช่องความถี่ RF Frequency Channel ให้ห่างกันมากๆ

- ในกรณีที่มีการใช้เครื่อง ET-RF24G V2.0 หลายๆกลุ่มในพื้นที่ใกล้เคียงกัน ควรกำหนดช่องความถี่ในการใช้งาน หรือ RF Frequency Channel ให้ห่างกันด้วยเพื่อป้องกันการรบกวนกัน

- การใช้งานเครื่อง ET-RF24G V2.0 แบบ Auto Direction นั้น ถ้ามีการส่งข้อมูลจำนวนมากๆ ควรจัดแบ่งข้อมูลออกเป็นชุดๆ โดยให้มีขนาดข้อมูลชุดละไม่เกิน 64 Byte โดยในการส่งข้อมูลแต่ละชุดนั้นให้ทำการส่งข้อมูลอย่างต่อเนื่องโดยให้ข้อมูลแต่ละ Byte มีระยะเวลาห่างกันไม่เกิน 2.5mS เนื่องจากถ้าข้อมูลขาดหายไปนานกว่านี้ เครื่อง ET-RF24G V2.0 จะทำการเปลี่ยนโหมดของการส่งข้อมูลกลับเป็นโหมดของการรับข้อมูลแทน ซึ่งเมื่อมีการส่งข้อมูล Byte ถัดไปมาอีกก็จะต้องเสียเวลาในการสลับโหมดจากฝ่ายรอรับข้อมูลให้เป็นฝ่ายส่งข้อมูลอีก ซึ่งจะทำให้ประสิทธิภาพในการจัดส่งข้อมูลลดลงเนื่องจากต้องเสียเวลาในการสลับโหมดการทำงานของวงจรภาค RF อยู่ตลอดเวลา โดยที่เมื่อทำการจัดส่งข้อมูลครบ 64 Byte แล้ว ให้ทำการหน่วงเวลาไว้ชั่วขณะหนึ่ง ประมาณ 1mS-2mS แล้วจึงส่งข้อมูลชุดถัดไปอีกอย่างนี้เรื่อยๆ จะทำให้การรับส่งข้อมูลมีประสิทธิภาพสูงสุด

- การใช้งานเครื่อง ET-RF24G V2.0 แบบ Auto Direction นั้น ควรหน่วงเวลาในการสลับโหมดจากฝ่ายของการรอรับข้อมูลเป็นฝ่ายส่งข้อมูล อย่างน้อยที่สุด 3mS – 5mS ซึ่งถ้าส่งข้อมูลย้อนกลับด้วยเวลาที่เร็วกว่านี้อาจทำให้ฝ่ายตรงข้ามไม่สามารถรับข้อมูล Byte แรกได้ทัน

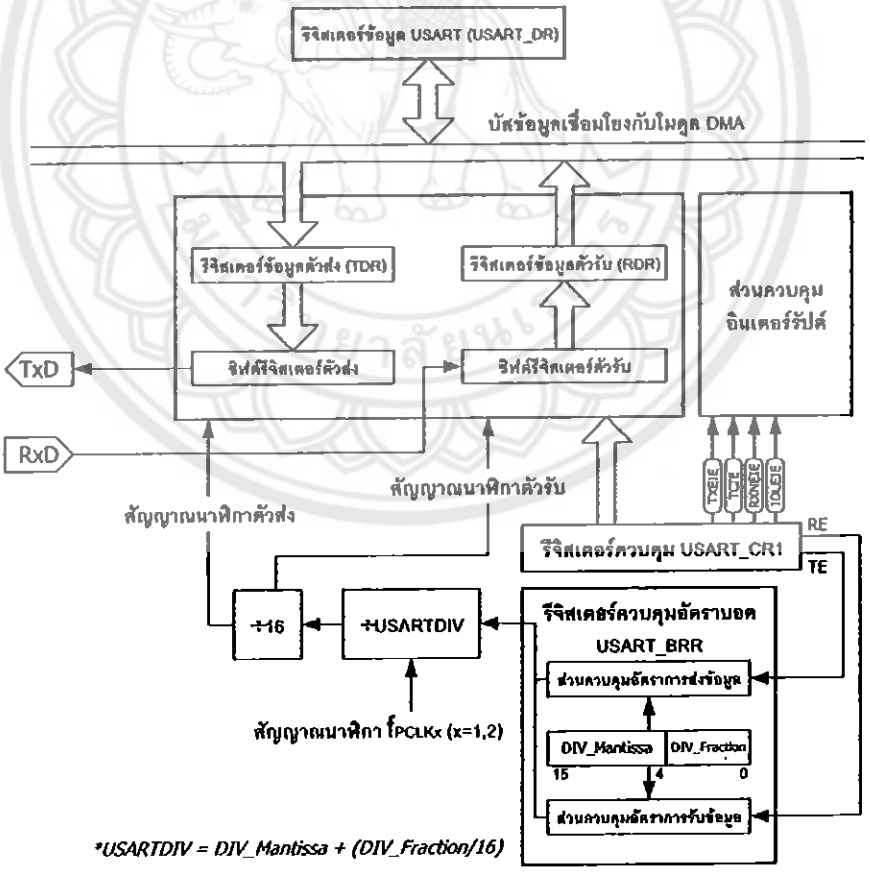
2.6 การสื่อสารข้อมูลอนุกรม USART ภายใน STM32 [7]

โมดูลสื่อสารข้อมูลอนุกรมหรือโมดูล USART (Universal synchronous asynchronous receiver transmitter) ภายในไมโครคอนโทรลเลอร์เบอร์ STM32F107VC มี 3 โมดูลได้แก่ USART1, USART2 และ USART3 ใช้ในการสื่อสารข้อมูลอนุกรมมาตรฐานและสนับสนุนการสื่อสารระบบเครือข่ายเชื่อมโยงในพื้นที่ หรือ LIN (local interconnection network), การสื่อสารกับสมาร์ตการ์ด (Smartcard Protocol), โมดูลรับส่งข้อมูลอนุกรมผ่านแสงอินฟราเรดหรือ IrDA(infrared data association), โมเด็ม (CTS/RTS) และสนับสนุนการทำงานแบบมัลติโปรเซสเซอร์ นอกจากนั้นยังสามารถเชื่อมโยงการทำงานของโมดูล UART กับ DMA ได้ด้วย

2.6.1 คุณสมบัติทางเทคนิคที่น่าสนใจของโมดูล USART

- สามารถสื่อสารข้อมูลแบบไม่เข้าจังหวะ (Asynchronous) ได้สองทิศทาง
- สามารถเลือกให้ทำการสื่อสารข้อมูลแบบทิศทางเดียวโดยใช้สายสัญญาณเพียงเส้นเดียวได้
- กำหนดรูปแบบข้อมูลสื่อสารได้ทั้งความยาวข้อมูล จำนวนบิตหยุด และบิตตรวจสอบ (Parity)
- กำหนดอัตราเร็วในการถ่ายทอดข้อมูล(Baud rate)ได้สูงสุดถึง 4.5 เมกะบิตต่อวินาที
- รองรับการทำงานแบบเครือข่ายเชื่อมโยงในพื้นที่ หรือ LIN
- รองรับการติดต่อกับสมาร์ตการ์ดตามมาตรฐาน ISO 7816-3
- รองรับการติดต่อกับโมดูลสื่อสารข้อมูลอนุกรมผ่านแสงอินฟราเรดหรือ IrDA แบบ 3/16 บิตในโหมดปกติ
- เชื่อมโยงกับโมดูล DMA สามารถส่งผ่านข้อมูลจากเพอริเฟอรัลเพื่อส่งออกไปทางพอร์ตอนุกรมของโมดูล USART ได้โดยไม่ต้องผ่านการประมวลผลจากซีพียู
- แยกบิตควบคุมระหว่างตัวรับและตัวส่งข้อมูลอนุกรม
- มีแฟลกตรวจสอบกระบวนการถ่ายทอดข้อมูล 3 ตัวคือ แฟลกแจ้งบัฟเฟอร์ตัวรับเต็ม, แฟลกแจ้งบัฟเฟอร์ตัวส่งว่าง และแฟลกแจ้งการสิ้นสุดของการถ่ายทอดข้อมูล

- มีแฟลกตรวจสอบความผิดพลาด 4 แบบคือ แฟลกแจ้งความผิดพลาดจากโอเวอร์รัน,แฟลกแจ้งความผิดพลาดจากเฟรมข้อมูล และแฟลกแจ้งความผิดพลาดจากสัญญาณรบกวน และแฟลกแจ้งความผิดพลาดจากพาริตี้
- สามารถกำเนิดสัญญาณอินเตอร์รัปต์ได้ 10 แหล่ง ประกอบด้วย การเปลี่ยนแปลงของสัญญาณ CTS, ตรวจจับการหยุดของ LIN, รีจิสเตอร์ข้อมูลของตัวส่งว่าง, การถ่ายทอดข้อมูลเสร็จสมบูรณ์, รีจิสเตอร์ข้อมูลตัวรับเต็ม, การได้รับภาวะไอเดิลของสายสัญญาณ, ความผิดพลาดจากโอเวอร์รัน, ความผิดพลาดจากเฟรมข้อมูล, ความผิดพลาดจากสัญญาณรบกวน และความผิดพลาดจากพาริตี้
- รองรับการสื่อสารแบบมัลติโปรเซสเซอร์
- รองรับมาตรฐานรูปแบบข้อมูลอนุกรมแบบ NRZ



รูปที่ 2.6 แสดงส่วนประกอบและไดอะแกรมการทำงานของโมดูล USART [7]

2.6.2 การทำงานเบื้องต้นของโมดูล USART

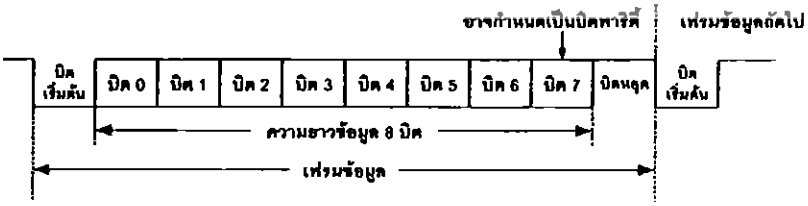
ในรูปที่ 2-14 แสดงส่วนประกอบและไดอะแกรมการทำงานของโมดูล USART ในไมโครคอนโทรลเลอร์ STM32F107VC ขาสัญญาณที่ใช้หลัก ๆ มี 2 ขา คือ Rx ขาข้อมูลอนุกรมเข้า และ Tx ขาข้อมูลอนุกรมออก

กระบวนการทำงานของโมดูล USART ในการรับส่งข้อมูลอนุกรมประกอบด้วย

- การเตรียมสภาวะของสายสัญญาณให้พร้อมสำหรับการส่งหรือรับข้อมูล
- การส่งบิตเริ่มต้น
- การถ่ายทอดข้อมูล (8 หรือ 9 บิต) โดยเริ่มจากบิตในที่สำคัญต่ำสุด หรือ LSB ก่อน
- การส่งบิตสุดท้ายหรือบิตหยุด ซึ่งมีขนาด 0.5, 1, 1.5 หรือ 2 บิต
- การกำหนดอัตราการรับ-ส่งข้อมูล (Baud rate) โดยเขียนข้อมูลไปยังรีจิสเตอร์ USART_BRR
- การตรวจสอบสถานการณ์ทำงานของโมดูล โดยดูได้จากข้อมูลในรีจิสเตอร์ USART_SR
- รีจิสเตอร์ข้อมูลของโมดูล USART ซึ่งก็คือข้อมูลรีจิสเตอร์ USART_DR

2.6.3 รูปแบบข้อมูลอนุกรมที่ใช้ใน USART

(ก) รูปแบบของเฟรมข้อมูลอนุกรม 9 บิต และ 1 บิตหยุด (บิต M ในรีจิสเตอร์ USART_CR1 เป็น "1")



(ข) รูปแบบของเฟรมข้อมูลอนุกรม 8 บิต และ 1 บิตหยุด (บิต M ในรีจิสเตอร์ USART_CR1 เป็น "0")

รูปที่ 2.7 รูปแบบของข้อมูลที่สัมพันธ์กับบิต M ในรีจิสเตอร์ควบคุม USART [7]

โมดูล USART ของ STM32F107VC สามารถรับข้อมูลอนุกรมที่มีความยาว 8 หรือ 9 บิต ดังแสดงรูปแบบของข้อมูลอนุกรมในรูปที่ 2-15 โดยสามารถกำหนดความยาวของข้อมูลที่บิต M ในรีจิสเตอร์ควบคุม USART_CR1

การเริ่มต้นส่งข้อมูลจะเริ่มจากการส่งบิตเริ่มต้น (Start bit) ด้วยการทำให้สายสัญญาณมีลอจิก "0" จากนั้นตามด้วยการส่งข้อมูลซึ่งจะส่งข้อมูลบิต 0 ออกไปก่อน หากข้อมูลมีความยาว 9 บิต บิตสุดท้าย (บิต 8) อาจกำหนดเป็นบิตตรวจสอบ จากนั้นตามด้วยการส่งบิตหยุดหรือบิตปิดท้าย (stop bit) นี่คือ ส่วนประกอบทั้งหมดในหนึ่งเฟรมข้อมูล

ถ้าเลือกใช้ความยาวข้อมูล 8 บิต บิตสุดท้ายหรือบิต 7 สามารถกำหนดให้เป็นบิตตรวจสอบพาริตี้ได้

รูปแบบข้อมูลที่นิยมใช้กันสำหรับการสื่อสารข้อมูลระหว่างไมโครคอนโทรลเลอร์กับพอร์ตอนุกรม RS-232 หรือพอร์ตอนุกรมเสมือนจากการแปลงโดยตัวแปลงสัญญาณพอร์ต USB เป็นพอร์ตอนุกรม RS-232 คือ 8N1 หมายถึง 8 บิตข้อมูล ไม่มีการตรวจสอบและมีบิตหยุด 1 บิต

2.6.4 กระบวนการส่งข้อมูลในโมดูล USART

มีขั้นตอน โดยสรุปดังนี้

1. เปิดการใช้งานโมดูล USART ด้วยการเซตบิต UE ในรีจิสเตอร์ USART_CR1
2. กำหนดความยาวของข้อมูลเป็น 8 หรือ 9 บิต ด้วยการกำหนดค่าที่บิต M ในรีจิสเตอร์ USART_CR1
3. เลือกขนาดของบิตหยุด ด้วยการกำหนดค่าในรีจิสเตอร์ USART_CR2
4. หากต้องการทำงานเชื่อมโยงกับโมดูล DMA เซตบิต DMAT ในรีจิสเตอร์ USART_CR3
5. เซตบิต TE ในรีจิสเตอร์ USART_CR1 เพื่อเริ่มต้นการส่งข้อมูล
6. กำหนดค่าอัตรา(Baud rate)ไปยังรีจิสเตอร์ USART_BRR
7. เขียนข้อมูลที่ต้องการส่งไปยังรีจิสเตอร์ข้อมูล USART_DR และทำการวนเขียนข้อมูลในไบต์ต่อไป ถ้าหากกำหนดการทำงานเป็นแบบบัฟเฟอร์เดี่ยว

อย่างไรก็ตาม จะต้องมีการกำหนดโหมดการทำงานของขาพอร์ตที่ใช้งานให้เป็นเอาต์พุตแบบพุชพูล (Push Pull)ก่อน

การกำหนดขนาดของบิตหยุดสำหรับการสื่อสารข้อมูลอนุกรมปกตินี้จะกำหนดเป็น 1 หรือ 2 บิต ส่วนค่า 0.5 และ 1.5 บิตนั้นจะใช้ในการติดต่อสมาร์ตการ์ด

2.6.5 การสื่อสารข้อมูลแบบไบต์เดี่ยว

เป็นการใช้งานโมดูล USART ขั้นพื้นฐานที่สุด โดยทุกครั้งที่มีการส่งข้อมูลออกจากโมดูล USART จะทำให้บิต TXE ซึ่งเป็นบิตแจ้งสถานะบัฟเฟอร์ตัวส่งว่างเซตเป็น "1" ทำให้ผู้พัฒนาสามารถเขียนข้อมูลไปยังรีจิสเตอร์ข้อมูล USART_DR เพื่อส่งข้อมูลนำไบต์ต่อไปได้ และทุกครั้งที่มีการเขียนข้อมูลไปยังรีจิสเตอร์ USART_DR บิต TXE ก็จะเคลียร์ เพื่อแจ้งให้ทราบว่า บัฟเฟอร์ตัวส่งไม่ว่างแล้ว หากต้องการส่งข้อมูลใหม่จะต้องรอให้มีการส่งข้อมูลออกไปจาโมดูล USART เสียก่อน

การเซตของบิต TXE เป็นการแจ้งให้ทราบถึง สถานการณ์ทำงานของโมดูล USART ดังนี้

- ข้อมูลถูกถ่ายทอดจากรีจิสเตอร์ข้อมูลตัวส่ง (TRD) ไปยังชิฟต์รีจิสเตอร์ อันเป็นการเริ่มต้นของกระบวนการถ่ายทอดข้อมูล
- รีจิสเตอร์ TRD ว่าง
- สามารถเขียนข้อมูลไบต์ต่อไปไปยังรีจิสเตอร์ USART_DR โดยไม่ทับข้อมูลเดิม
- จะเกิดการอินเตอร์รัปต์ขึ้นหากมีการเอ็นเอเบิลที่บิต TXEIE ไว้ (บิต TXEIE เป็น "1")

เมื่อการส่งข้อมูลเสร็จสมบูรณ์ บิต TC (Transmitting Complete) จะถูกเซต จะเกิดการอินเตอร์รัปต์ขึ้นหากมีการเอ็นเอเบิลที่บิต TCIE ไว้ (บิต TCIE เป็น "1") ส่วนการเคลียร์บิต TC สามารถทำได้ด้วยกระบวนการทางซอฟแวร์ตามขั้นตอนดังนี้

1. อ่านข้อมูลสถานะจากรีจิสเตอร์ USART_SR
2. เขียนข้อมูลไปยังรีจิสเตอร์ USART_SR

2.6.6 กระบวนการรับข้อมูลในโมดูล USART

มีขั้นตอน โดยสรุปดังนี้

1. เปิดการใช้งานโมดูล USART ด้วยการเซตบิต UE ในรีจิสเตอร์ USART_CR1
2. กำหนดความยาวของข้อมูลเป็น 8 หรือ 9 บิต ด้วยการกำหนดค่าที่บิต M ในรีจิสเตอร์ USART_CR1
3. เลือกขนาดของบิตหยุด ด้วยการกำหนดค่าในรีจิสเตอร์ USART_CR2
4. หากต้องการทำงานเชื่อมโยงกับโมดูล DMA เซตบิต DMAT ในรีจิสเตอร์ USART_CR3
5. เซตบิต TE ในรีจิสเตอร์ USART_CR1 เพื่อเริ่มต้นการส่งข้อมูล
6. กำหนดค่าอัตราบอดไปยังรีจิสเตอร์ USART_BRR

7. เซตบิต RE ในรีจิสเตอร์ USART_CR1 เพื่อเอ็นเอเบิลการรับข้อมูลให้แก่ตัวรับในโมดูล USART

เมื่อมีการรับข้อมูลเกิดขึ้น จะเกิดการทำงานภายในโมดูล USART ดังนี้

- บิต RXNE ถูกเซต เป็นการแจ้งให้ทราบว่า มีการถ่ายทอดข้อมูลจากชิฟต์รีจิสเตอร์ไปยังรีจิสเตอร์ตัวรับข้อมูล (RDR) เรียบร้อยแล้ว
- เกิดการอินเตอร์รัปต์ขึ้น ถ้าหากมีการเอ็นเอเบิลที่บิต RXNEIE (บิต RXNEIE เป็น "1")
- ถ้าหากมีความผิดพลาดเกิดขึ้น แฟลกแจ้งความผิดพลาดจะถูกเซต
- เคลียร์บิต RXNE ด้วยการอ่านข้อมูลออกจากรีจิสเตอร์ USART_DR และต้องอ่านข้อมูลก่อนที่การรับข้อมูลไบต์ถัดไปจะเสร็จสิ้นลง

2.6.7 รีจิสเตอร์ที่สำคัญของโมดูล USART ใน STM32F107VC ประกอบด้วย

- USART_SR (USART Status Register) รีจิสเตอร์แสดงสถานการณ์ทำงานของโมดูล USART
- USART_DR (USART Data Register) รีจิสเตอร์ข้อมูลของโมดูล USART
- USART_BRR (USART Baud Rate Register) รีจิสเตอร์กำหนดค่าอัตรา Baud rate ของโมดูล USART
- USART_CR1 (USART Control Register1) รีจิสเตอร์ควบคุมโมดูล USART ตัวที่ 1
- USART_CR2 (USART Control Register2) รีจิสเตอร์ควบคุมโมดูล USART ตัวที่ 2
- USART_CR3 (USART Control Register3) รีจิสเตอร์ควบคุมโมดูล USART ตัวที่ 3
- USART_GTPR (USART Guard-Time and Prescaler Register) รีจิสเตอร์กำหนดค่าเวลาป้องกันและปรีสเกลของโมดูล USART

2.7 พอร์ตอนุกรม RS232 [8]

RS-232 เป็นมาตรฐานการเชื่อมต่อข้อมูลแบบอนุกรมที่กำหนดโดย EIA (Electronics Industry Association) หรือสมาคมผู้ประกอบการอุตสาหกรรมอิเล็กทรอนิกส์ของอเมริกา โดยมีจุดเริ่มต้นจาก

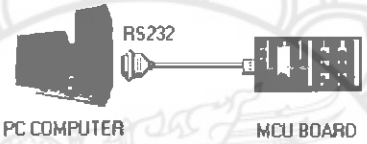
ความต้องการที่จะกำหนดมาตรฐานการเชื่อมต่อระหว่างคอมพิวเตอร์กับโมเด็มในสมัยนั้น ตัวมาตรฐานจะกำหนดสิ่งที่เกี่ยวข้องกับการเชื่อมต่อนี้ด้วยกันทั้งหมด 4 หัวข้อหลักๆ ด้วยกันคือ

1. คุณสมบัติทางไฟฟ้าของสัญญาณ

2. คุณสมบัติทางกลของการเชื่อมต่อ

3. หน้าที่การทำงานของวงจรสำหรับแลกเปลี่ยนข้อมูล

4. มาตรฐานการเชื่อมต่อสำหรับระบบสื่อสารเฉพาะอย่าง

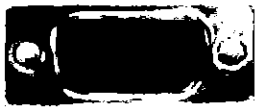


รูปที่ 2.8 แสดงการใช้งานพอร์ตอนุกรม RS232

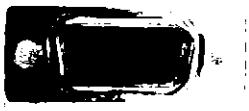
การสื่อสารแบบอนุกรม(Rs-232) มีความสำคัญ ต่อการใช้งานไมโครคอนโทรลเลอร์มาก เพราะสามารถใช้แป้นพิมพ์และจอภาพของคอมพิวเตอร์ เป็นอินพุตและเอาต์พุตในการติดต่อหรือควบคุม ไมโครคอนโทรลเลอร์ ด้วยสัญญาณอย่างน้อย เพียง 3 เส้นเท่านั้น คือ

- สายส่งสัญญาณ TX
- สายรับสัญญาณ RX
- และสาย GND

โดยปกติพอร์ตอนุกรม RS-232C จะสามารถต่อสายได้ยาว 50 ฟุตโดยขึ้นอยู่กับ ชนิดของสายสัญญาณ, ระยะทาง, และปริมาณสัญญาณรบกวน



รูปที่ 2.9 พอร์ตอนุกรมของคอมพิวเตอร์ DB9 ตัวผู้



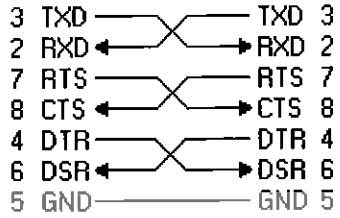
รูปที่ 2.10 พอร์ตอนุกรมของอุปกรณ์ภายนอก DB9 ตัวเมีย

2.7.1 การจัดขาของคอนเน็กเตอร์อนุกรมแบบ DB9 และหน้าที่การใช้งานต่างๆ

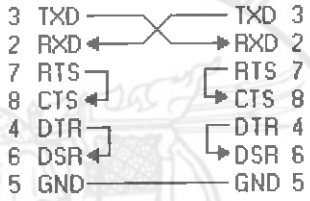
รูปที่ 2.11 แสดงการจัดขาของคอนเน็กเตอร์อนุกรมแบบ DB9

Pin Description	Type
1. Data Carrier Detect (DCD)	Input
2. Received Data (RXD)	Input
3. Transmitted Data (TXD)	Output
4. Data Terminal Ready (DTR)	Output
5. Signal Ground (GND)	Input
6. Data Set Ready (DSR)	Input
7. Request To Send (RTS)	Output
8. Clear to Send (CTS)	Input
9. Ring Indicator (RI)	Input

2.7.2 การเชื่อมต่ออุปกรณ์อุกรณ์ภายนอกเข้ากับคอมพิวเตอร์ด้วยสาย DB9



รูปที่ 2.12 การเชื่อมต่ออุปกรณ์ภายนอกผ่าน DB9 แบบ Null modem[8]



รูปที่ 2.13 การต่ออุปกรณ์ภายนอกผ่าน DB9 แบบ 3 เส้น[8]

2.7.3 การทำงานของขาสัญญาณ DB9

TXD เป็นขาที่ใช้ส่งข้อมูล

RXD เป็นขาที่ใช้รับข้อมูล

DTR แสดงสภาวะพอร์ตว่าเปิดใช้งาน ,DSR ตรวจสอบว่าพอร์ต ที่ติดต่อด้วย เปิดอยู่หรือไม่

- เมื่อเปิดพอร์ตอนุกรม ขา DTR จะ ON เพื่อให้อุปกรณ์ได้รับทราบว่าต้องการติดต่อด้วย
- ในขณะเดียวกันก็จะตรวจสอบขา DSR ว่าอุปกรณ์พร้อมหรือไม่

RTS แสดงสภาวะพอร์ตว่าต้องการส่งข้อมูล ,CTS ตรวจสอบว่าพอร์ตที่ติดต่ออยู่ ต้องการส่งข้อมูลหรือไม่

- เมื่อต้องการส่งข้อมูลขา RTS จะ ON และจะส่งข้อมูลออกที่ขา TXD เมื่อส่งเสร็จก็จะ OFF
- ในขณะเดียวกันก็จะตรวจสอบขา CTS ว่าอุปกรณ์ต้องการที่จะส่งข้อมูลหรือไม่

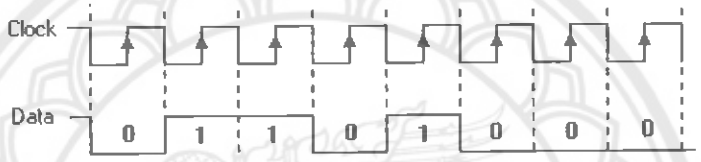
GND ขา ground

2.7.4 รูปแบบการสื่อสารแบบอนุกรม [8]

มีด้วยกันอยู่ 2 แบบ คือแบบซิงโครนัส (Synchronous) และแบบอะซิงโครนัส(Asynchronous)

- **การสื่อสารแบบซิงโครนัส (Synchronous)**

  การรับส่งข้อมูล จะมีสัญญาณนาฬิกา ซึ่งเป็นตัวกำหนด จังหวะเวลา การส่งข้อมูล ร่วมอยู่ด้วย อีกเส้นหนึ่ง ใช้คู่กับสัญญาณข้อมูล ตัวอย่างเช่น การส่งสัญญาณจากคีย์บอร์ด

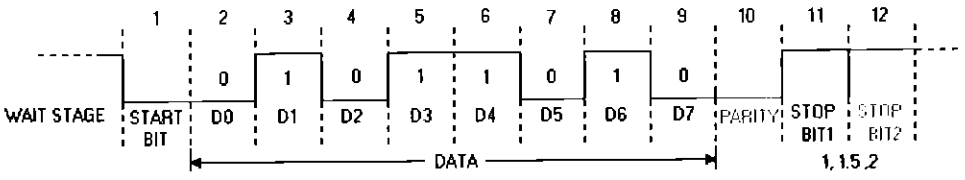


รูปที่ 2.14 แสดง การสื่อสารอนุกรมแบบซิงโครนัส

- **การสื่อสารแบบอะซิงโครนัส (Asynchronous)**

  การรับส่งข้อมูล โดยที่ไม่จำเป็นต้อง มีสัญญาณนาฬิกา ร่วมด้วย แต่จะใช้ให้ตัวส่ง และตัวรับ มีอัตราส่งข้อมูล ที่เท่ากัน รูปแบบข้อมูลแบบอะซิงโครนัส ประกอบด้วย 4 ส่วนคือ

  1. บิตเริ่มต้น (Start bit) มีขนาด1 บิต
  2. บิตข้อมูล (Data) มีขนาด 5,6,7 หรือ 8 บิต
  3. บิตตรวจสอบพาริตี้ (Parity bit) มีขนาด 1 บิตหรือไม่มี
  4. บิตหยุด (Stop bit) มีขนาด 1, 1.5, 2 บิต



รูปที่ 2.15 แสดง การสื่อสารอนุกรมแบบอะซิงโครนัส

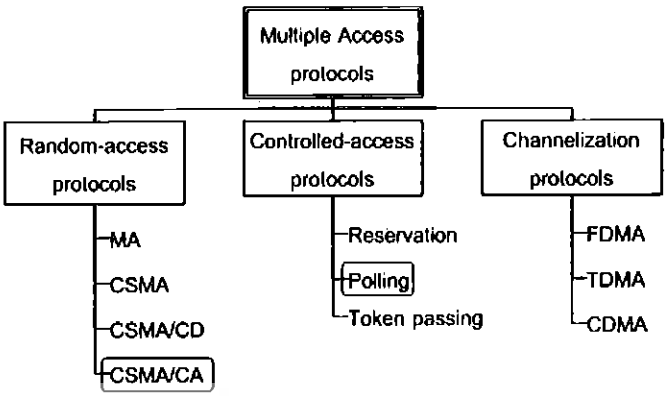
- เมื่อไม่มีการส่งข้อมูล ขา data จะมีสถานะเป็นโลจิก "1" หรือ สถานะหยุดรอ (Waiting stage)
- เมื่อเริ่มต้นส่งข้อมูลจะให้ขา data เป็นโลจิก "0" เป็นจำนวน 1 บิต เรียกว่าบิตเริ่มต้น (Start bit)
- จากนั้นก็จะเริ่มต้นส่งข้อมูล โดยส่งบิตต่ำไปก่อน (LSB)
- แล้วตามด้วยพาริตี้บิต (จะมีหรือไม่ก็ได้ ขึ้นอยู่กับการติดตั้งค่า ของทั้งสองฝ่าย)
- สุดท้ายตามด้วยโลจิก "1" อย่างน้อย 1 บิต (มีขนาด 1, 1.5, หรือ 2 บิต) เพื่อแสดงว่าสิ้นสุดข้อมูล

2.7.5 การรับและส่งข้อมูลแบบอนุกรมยังแบ่งออกเป็นลักษณะการใช้งานได้ 3 แบบคือ

1. แบบซิมเพลกซ์ (Simplex) เป็นการส่ง หรือรับข้อมูล แบบทิศทางเดียว เท่านั้น
2. แบบฮาล์ฟดูเพลกซ์ (Half Duplex) เป็นการส่งและรับข้อมูลแบบสลับกัน คือเมื่อด้านหนึ่งส่ง อีกด้านหนึ่ง เป็นฝ่ายรับ สลับกัน ไม่สามารถรับ-ส่งในเวลาเดียวกันได้
3. แบบฟลูดูเพลกซ์ (Full Duplex) สามารถรับ-ส่งข้อมูลในเวลาเดียวกันได้

2.8 Multiple Access Protocols [9]

Multiple Access Protocol คือ ข้อกำหนดในการควบคุมการเข้าใช้ช่องสัญญาณระหว่างอุปกรณ์สื่อสารหลายชุดที่ใช้ตัวกลางร่วมกัน เพื่อป้องกันไม่ให้สถานีต่างๆ ส่งข้อมูลออกมาพร้อมกัน ซึ่งทำให้เกิดการชนกันของข้อมูลภายในสื่อกลาง เราสามารถแยกกลุ่มของโพรโตคอลที่ใช้ในจัดการการใช้งานสื่อกลางร่วมกัน ออกได้ 3 กลุ่ม ดังรูป 2.16



รูปที่ 2.16 Multiple Access Protocol [9]

จากรูปที่ 2.16 Multiple Access Protocol แบ่งออกเป็น 3 ประเภท ประเภทแรก Random Access Protocols เป็นการกำหนดให้อุปกรณ์สื่อสารทุกตัวเข้าใช้ช่องสัญญาณสื่อสารได้อย่างอิสระ ประเภทที่สอง Controlled Access Protocols ในวิธีการนี้ ทุกๆ สถานีจะต้องหาข้อตกลงกันว่าสถานีใดมีสิทธิเข้าใช้ตัวกลางและประเภทสุดท้ายคือ Channelization Protocols เป็นการจัดสรรแบ่งปัน Bandwidth ของตัวกลางด้วยเวลา (Time) ความถี่ (Frequency) หรือรหัส (Code)

2.9 Random Access Protocols [9]

random access หรือที่รู้จักกันในชื่อ ALOHA จะมีกระบวนการทำงานที่ไม่ซับซ้อน เรียกว่า multiple access (MA) จากนั้นได้มีการพัฒนากระบวนการในการทำงาน เพื่อให้มีประสิทธิภาพมากขึ้น จนกลายมาเป็น carrier sense multiple access (CSMA) ซึ่งเทคนิคนี้ได้แยกออกเป็นอีก 2 ประเภทได้แก่ CSMA/CD และ CSMA/CA

2.9.1 Multiple Access (MA)

เครือข่าย ALOHA จะมีสถานีฐาน (bass station) ที่จะเป็นตัวที่คอยควบคุมการทำงานทั้งหมด การส่งข้อมูลระหว่างสถานีต่างๆ จะต้องกระทำผ่านสถานีฐานก่อนเสมอ โดยสถานีฐานจะส่งต่อข้อมูลนั้นให้กับสถานีปลายทางเอง ในการส่งสัญญาณจากสถานีไปยังสถานีฐานที่ใช้ความถี่ของเครื่องพาห์ที่ 407 MHz ส่วนการส่งสัญญาณจากสถานีฐานไปยังสถานีอื่นๆ จะใช้ความถี่ของคลื่นพาห์ที่ 413 MHz

เครือข่าย ALOHA จะใช้อากาศเป็นสื่อกลางในการนำส่งสัญญาณ ดังนั้นถ้าแต่ละสถานีส่งสัญญาณออกมาพร้อมกัน จะทำให้เกิดการรบกวนกันได้ เพื่อเป็นการแก้ปัญหาที่เกิดขึ้น โพรโตคอล ALOHA จะมีการทำงานง่ายๆ ดังนี้

- สถานีที่จะสามารถส่งข้อมูลได้นั้น จะต้องมีข้อมูลที่ตองการส่งเสียก่อน
- หลังจากส่งข้อมูลแล้ว จะต้องรอการตอบรับ ถ้าไม่มีการตอบรับกลับมาภายในช่วงเวลาที่กำหนดจะทำการสุ่มเวลาของการส่งข้อมูลครั้งต่อไป แล้วทำการส่งข้อมูลใหม่อีกครั้ง

2.9.2 Carrier Sense Multiple Access (CSMA)

CSMA เป็นเทคนิคที่สามารถช่วยลดการชนกันของข้อมูลได้ แต่ไม่สามารถแก้ปัญหาการชนกันได้ทั้งหมด เนื่องจากยังมีบางกรณีที่ข้อมูลอาจจะเกิดการชนกันได้ เช่น ถ้ามีสถานีหนึ่งส่งข้อมูลออกมา ในขณะที่อีกสถานีต้องการที่จะส่ง เมื่อตรวจสอบสื่อในขณะนั้นจะพบว่ายังคงว่างอยู่ เนื่องจากข้อมูลที่ถูกส่งออกมาจากสถานีแรกยังเดินทางมาไม่ถึง สถานีที่สองจึงส่งข้อมูลออกไปด้วย ซึ่งเหตุการณ์แบบนี้จะทำให้เกิดการชนกัน

วิธีส่งเฟรมข้อมูล

ก่อนที่จะทำการส่งเฟรมข้อมูลจะต้องมีการตรวจสอบสื่อก่อนว่าพร้อมที่จะส่งได้หรือไม่ ถ้าสื่อกลางไม่พร้อมที่จะส่งมีวิธีการ 2 วิธีการ ได้แก่ วิธีสุ่มเวลา (nonpersistent) และวิธีสุ่มความน่าจะเป็น (persistent)

วิธีสุ่มเวลา (nonpersistent)

วิธีการแบบนี้ถ้าสื่อว่าง จะสามารถส่งเฟรมข้อมูลออกไปได้ทันที แต่ถ้าสื่อไม่ว่างจะต้องทำการรอก่อน โดยระยะเวลาในการรอนั้นจะใช้วิธีการสุ่มเวลา จากนั้นจึงทำการตรวจสอบสื่ออีกครั้งว่าพร้อมที่จะส่งได้หรือไม่ วิธีการแบบนี้โอกาสที่แต่ละสถานีจะส่งเฟรมข้อมูลออกมาชนกันเป็นครั้งที่สองนั้นจะมีน้อยลง เนื่องจากการสุ่มเวลาของแต่ละสถานีจะเท่ากันนั้นน้อยมาก

วิธีสุ่มความน่าจะเป็น (persistent)

วิธีนี้จะหลักของความน่าจะเป็น เมื่อสื่อกลางไม่พร้อมที่จะส่งได้ แต่ละสถานีจะทำการสุ่มตัวเลขของความน่าจะเป็นขึ้นมา ซึ่งจะมีค่าอยู่ระหว่าง 0 ถึง 100 ถ้าไม่สุ่มมาแล้วปรากฏว่ามีความเป็นไปได้สูง ก็จะส่งเฟรมข้อมูลออกมาทันที แต่ถ้ามีความเป็นไปได้ต่ำจะต้องรอไปก่อน

CSMA/CD

carrier sense multiple access with collision detection (CSMA/CD) จะเพิ่มเติมกระบวนการทำงานหลังจากการชนกันของข้อมูลเกิดขึ้นแล้ว เมื่อสถานีได้ทราบแล้วว่ามีการชนกันของข้อมูลเกิดขึ้นแล้ว เพื่อไม่ให้เกิดการชนกันเป็นครั้งที่สองแต่ละสถานีจะต้องรอก่อน ซึ่งวิธีการที่จะนำเอามาใช้นั้นเรียกว่า backoff

Backoff เป็นวิธีการคำนวณเวลาที่จะต้องส่งเฟรมข้อมูลในครั้งถัดไปหลังจากเกิดการชนกันของข้อมูลแล้ว โดยเวลาที่จะส่งในครั้งถัดไปนั้นจะอยู่ระหว่าง 0 ถึง 2^N เวลาสูงสุดของการเดินทางของข้อมูลจากต้นทางไปยังปลายทาง เมื่อ N หมายถึง จำนวนครั้งของการพยายามที่จะส่งข้อมูล

CSMA/CA

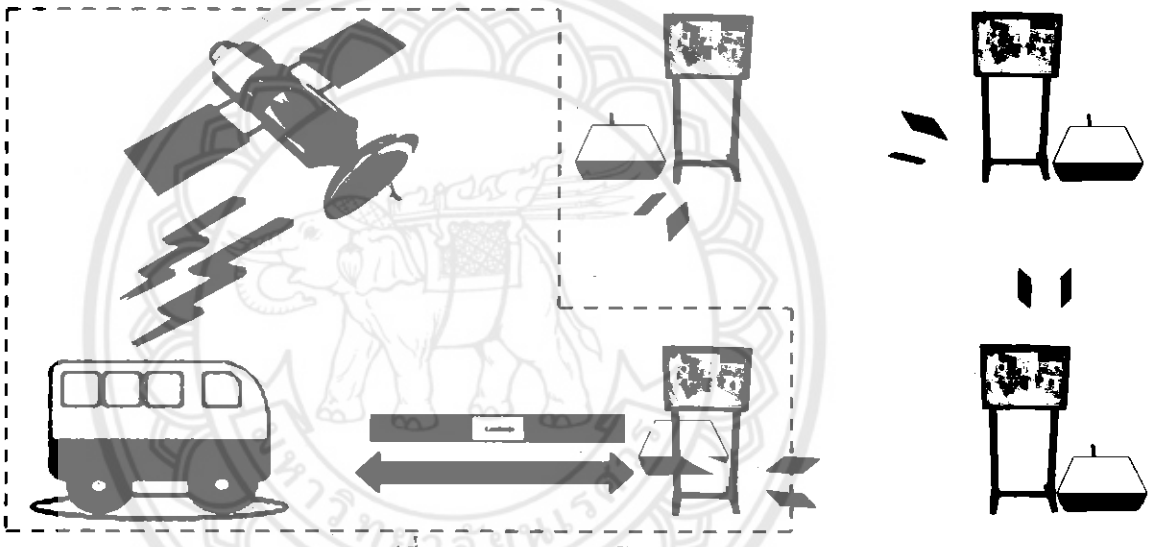
เป็นเทคนิคที่หลีกเลี่ยงการชนกันของข้อมูล ซึ่งสถานีจะเลือกวิธีการสุ่มเวลาหรือความน่าจะเป็นอย่างใดอย่างหนึ่งในการส่งเฟรมข้อมูล เมื่อสายว่างแล้ว สถานีที่ต้องการจะส่งเฟรมข้อมูลจะรอเวลาอยู่ช่วงหนึ่ง ที่เรียกว่า IFC (interframe gap) จากนั้นจะทำการสุ่มเวลาที่จะต้องส่งเฟรมข้อมูล เมื่อถึงกำหนดเวลาที่สุ่มแล้วจึงทำการส่งเฟรมข้อมูลออกไป และทำการกำหนดเวลาในการรอการตอบรับกลับมา ถ้ามีการตอบรับมาภายในเวลาที่กำหนด แสดงว่าสามารถส่งเฟรมข้อมูลไปได้เรียบร้อย แต่ถ้าไม่มีการตอบรับภายในเวลาที่กำหนดแล้ว แสดงว่าการส่งเฟรมข้อมูลครั้งนี้ล้มเหลว สถานีจะต้องทำการเพิ่มค่า backoff แล้วตรวจสอบว่าเกินจากที่กำหนดไว้หรือไม่ ถ้าเกินก็จะเลิกส่งเฟรมข้อมูลนั้น แต่ถ้าไม่เกินรอเวลาตามที่ได้กำหนดไว้ แล้วจึงเริ่มกระบวนการการส่งใหม่อีกครั้ง เทคนิค CSMA/CA จะนิยมใช้กันกับการแลนแบบไร้สาย

บทที่ 3

วิธีดำเนินโครงงาน

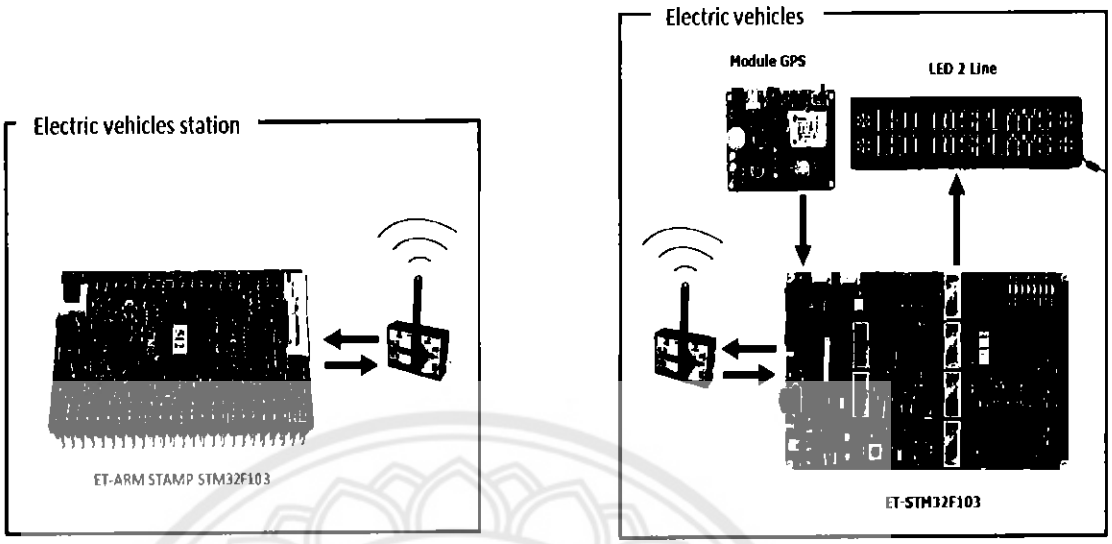
จากบทที่ผ่านมาได้กล่าวถึงทฤษฎีต่าง ๆ ที่จะนำมาพัฒนาระบบบอกตำแหน่งพิกัดโลกและระบบการติดต่อสื่อสารแบบไร้สาย โดยในบทนี้จะกล่าวถึงการวิเคราะห์ ออกแบบและพัฒนาระบบดังนี้

ภาพโดยรวมของโครงงานการติดตามรถไฟฟ้าโดยระบบบอกตำแหน่งพิกัดโลกและการสื่อสารแบบไร้สาย



รูปที่ 3.1 ภาพรวมของโครงงาน

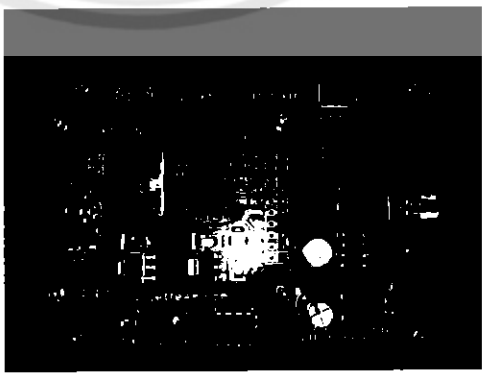
ในระบบนี้ประกอบไปด้วย 2 ส่วนหลัก ๆ คือ 1. ส่วนของระบบบอกตำแหน่งพิกัดโลกและส่วนแสดงผล 2. ส่วนของการติดต่อสื่อสารแบบไร้สายระหว่างไมโครคอนโทรลเลอร์ฝั่งลูกข่ายบนรถไฟฟ้ากับไมโครคอนโทรลเลอร์แม่ข่ายที่ป้ายจอดรถ



รูปที่ 3.2 โครงสร้างและระบบโดยรวมของโครงงาน

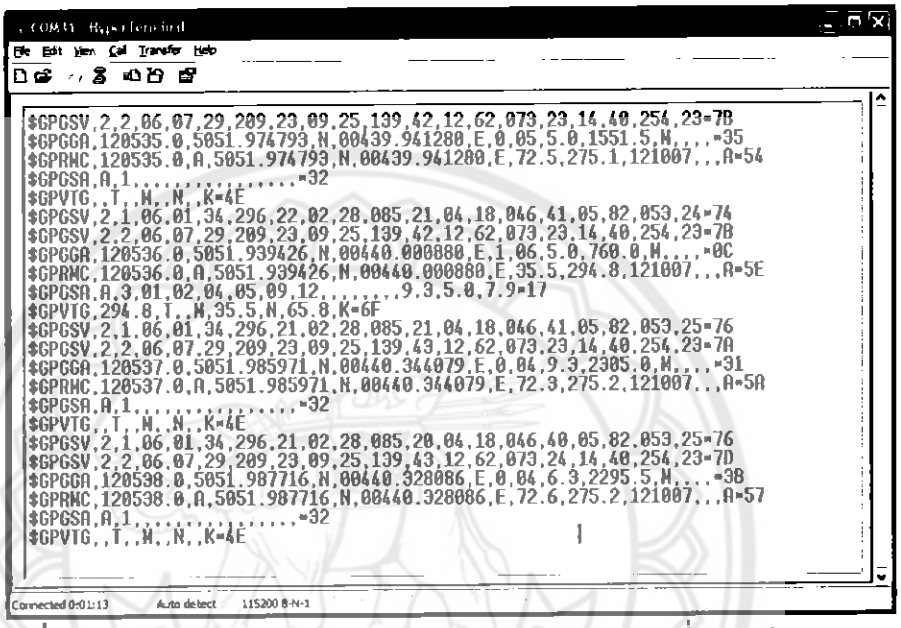
จากรูป 3.2 โครงสร้างและระบบโดยรวมของโครงงาน จะมีหลักการทำงานดังนี้ ส่วนของระบบบอกตำแหน่งพิกัดโลกและส่วนแสดงผลจะอยู่บนรถไฟฟ้า ประกอบไปด้วย โมดูลของระบบบอกตำแหน่งพิกัดโลก (Global Positioning System) โดยโมดูลของระบบบอกตำแหน่งพิกัดโลกนี้ จะมีการรับค่าละติจูดและลองจิจูด ของตำแหน่งของรถไฟฟ้า เมื่อได้ค่าตำแหน่งของรถไฟฟ้าแล้วจะส่งค่าที่ได้นั้นไปแสดงผลบนแอลซีดีบนรถไฟฟ้า แล้วส่งข้อมูลตำแหน่งพิกัดโลกไปยังป้ายจอดรถไฟฟ้าที่ป้ายปัจจุบัน

3.1 ระบบบอกตำแหน่งพิกัดโลก



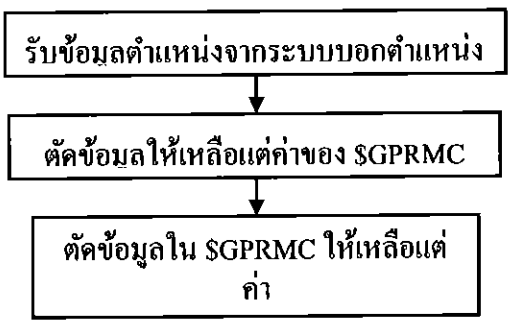
รูปที่ 3.3 โมดูลของระบบบอกตำแหน่งพิกัดโลกเป็นโมดูลสำเร็จ รุ่น ET-GPS START KIT

โมดูลของระบบบอกตำแหน่งพิกัดโลกเป็นโมดูลสำเร็จ รุ่น ET-GPS START KIT ดังรูป 3.2 โมดูลนี้จะนำมาใช้ในการหาตำแหน่งของรถไฟฟ้าบนพิกัดโลก โดยโมดูลของระบบบอกตำแหน่งพิกัดโลกนี้ จะได้ค่าตำแหน่งพิกัดโลกตามโปรโตคอลมาตรฐานที่พัฒนาโดย National Marine Electronics Association (NMEA) ดังรูปที่ 3.4



รูปที่ 3.4 แสดงค่าตำแหน่งพิกัดโลกตามโปรโตคอลมาตรฐานที่พัฒนาโดย NMEA

ในส่วนของการเขียนโปรแกรมของระบบบอกตำแหน่งพิกัดโลกนี้ จะเป็นการออกแบบโปรแกรมเพื่อนำไปใช้ในการคัดกรองข้อมูล ซึ่งในโครงงานนี้จะเลือกใช้ค่า $GPRMC Sโดยจะนำค่าในตำแหน่งที่ 20-43 มาตัดสตริงหาค่าละติจูดและลองจิจูด โดยลักษณะการทำงานของโปรแกรมจะมีการทำงานดังนี้



รูปที่ 3.5 แสดงแผนภาพในการคัดกรองข้อมูลจากระบบบอกตำแหน่งพิกัดโลก

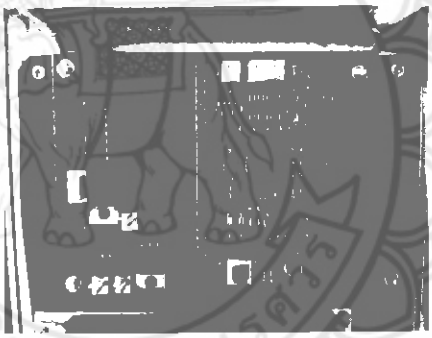
จากรูป 3.5 มีการทำงานดังนี้

1. ไมโครคอนโทรลเลอร์รับข้อมูลจากระบบบอกตำแหน่งพิกัดโลก ผ่านทางพอร์ตอนุกรม
2. นำค่าที่ได้จากจากระบบบอกตำแหน่งพิกัดโลกทั้งหมดมาคัดกรองข้อมูลโดยเลือกเฉพาะค่า $GPRMC
3. เมื่อได้ข้อมูลของ $GPRMC แล้ว ให้นำมาคัดกรองข้อมูลอีกที เพื่อหาค่าละติจูดและลองจิจูด โดยค่าของ $GPRMC ตำแหน่งที่ 20-29 คือค่าละติจูด และค่าของ $GPRMC ตำแหน่งที่ 33-43 คือค่าลองจิจูด

3.2 การพัฒนาโครงสร้างฝั่งแม่ข่าย

การพัฒนาโครงสร้างฝั่งแม่ข่าย ได้มีการใช้อุปกรณ์ดังนี้

- ไมโครคอนโทลเลอร์ตระกูลอาร์ม รุ่น ET-ARM STAMP STM32



รูปที่ 3.6 ไมโครคอนโทลเลอร์ตระกูลอาร์ม รุ่น ET-ARM STAMP STM32

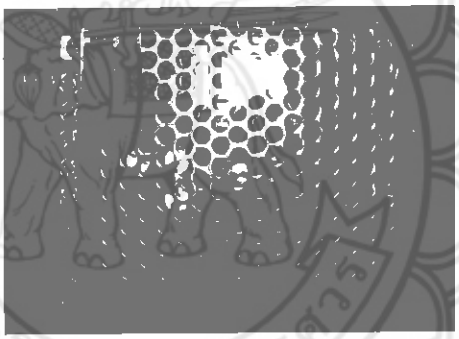
- ชุดแปลงสัญญาณระหว่างอาร์เอส232และคลื่นความถี่ไร้สาย รุ่น ET-RF24G V2.0

รูปที่ 3.7 ชุดแปลงสัญญาณระหว่างอาร์เอส232 กับคลื่นความถี่ไร้สาย

- บอร์ดแปลงไฟจาก 12 โวลต์ เป็น 5 โวลต์

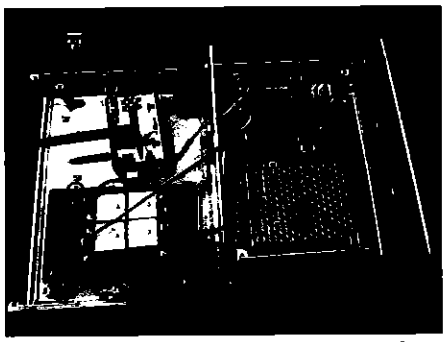
รูปที่ 3.8 บอร์ดแปลงไฟจาก 12 โวลต์ เป็น 5 โวลต์

- หม้อแปลงไฟ 220 โวลต์ เป็น 12 โวลต์



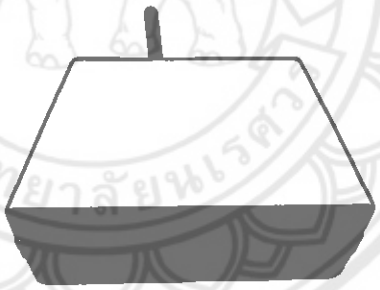
รูปที่ 3.9 หม้อแปลงไฟ 220 โวลต์ เป็น 12 โวลต์

สำหรับการจัดทำโครงสร้างของฝั่งแม่ข่าย จะใช้อะคริลิคมาทำแผ่นรองของกล่องอเนกประสงค์ โดยตัดแผ่นอะคริลิคให้เป็นสี่เหลี่ยมที่สามารถวางลงบนกล่องได้และยึดแผ่นอะคริลิคไว้กลับกล่องอเนกประสงค์ด้วยน็อต จากนั้นจึงนำอุปกรณ์ต่าง ๆ มายึดไว้กลับแผ่นอะคริลิคที่ติดไว้ในกล่อง เมื่อนำอุปกรณ์ต่าง ๆ ประกอบเข้ากับกล่องอเนกประสงค์ดังรูปแล้ว ขั้นตอนต่อไปจะต่อสายไฟดังรูป



รูปที่ 3.10 ลักษณะภายในของโครงสร้างฝั่งแม่ข่าย

1. จ่ายไฟ 5 โวลต์ จากบอร์ดแปลงไฟต่อไฟเลี้ยงให้กับชุดแปลงสัญญาณระหว่างอาร์เอส232 กับคลื่นความถี่ไร้สาย
2. จ่ายไฟ 5 โวลต์จากบอร์ดแปลงไฟให้กับบอร์ดไมโครคอนโทรลเลอร์
3. ต่อสายไฟที่ usart1 บนบอร์ดไมโครคอนโทรลเลอร์กับชุดแปลงสัญญาณระหว่างอาร์เอส232 กับคลื่นความถี่ไร้สาย
4. จ่ายไฟจากหม้อแปลงไฟ 220 โวลต์ เป็น 12 โวลต์ต่อไฟให้กับบอร์ดแปลงไฟ 12 โวลต์ เป็น 5 โวลต์

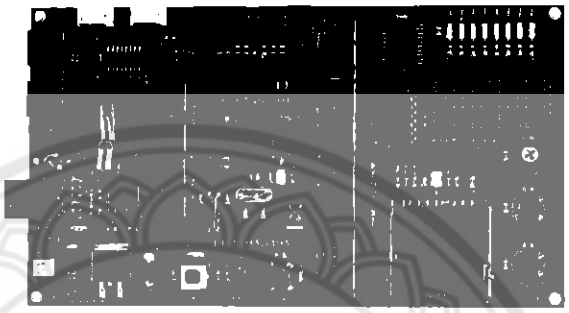


รูปที่ 3.11 ลักษณะภายนอกของโครงสร้างฝั่งแม่ข่าย

3.3 การพัฒนาโครงสร้างฝั่งลูกข่าย

การพัฒนาโครงสร้างฝั่งแม่ข่าย ได้มีการใช้อุปกรณ์ดังนี้

- ไมโครคอนโทลเลอร์ตระกูลอาร์ม รุ่น ET-STM32F103



รูปที่ 3.12 ไมโครคอนโทลเลอร์ตระกูลอาร์ม รุ่น ET-STM32F103

- ชุดแปลงสัญญาณระหว่างอาร์เอส232 กับคลื่นความถี่ไร้สาย รุ่น ET-RF24G V2.0

รูปที่ 3.13 ชุดแปลงสัญญาณระหว่างอาร์เอส232 กับคลื่นความถี่ไร้สาย

- จอแสดงผลแอลซีดี รุ่น 16216H LED BACKLIGHTS

รูปที่ 3.14 จอแสดงผลแอลซีดี รุ่น 16216H LED BACKLIGHTS

- บอร์ดแปลงไฟจาก 12 โวลต์ เป็น 5 โวลต์

รูปที่ 3.15 บอร์ดแปลงไฟจาก 12 โวลต์ เป็น 5 โวลต์

- โมดูลของระบบบอกตำแหน่งพิกัดโลก รุ่น ET-GPS START KIT

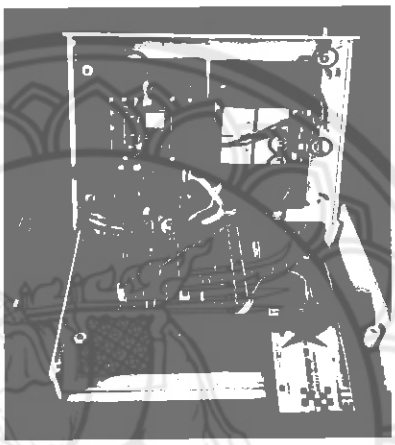
รูปที่ 3.16 โมดูลของระบบบอกตำแหน่งพิกัดโลก รุ่น ET-GPS START KIT

- แบตเตอรี่ลิเธียมโพลิเมอร์ 2200 mA

รูปที่ 3.17 แบตเตอรี่ลิเธียมโพลิเมอร์ 2200 mA

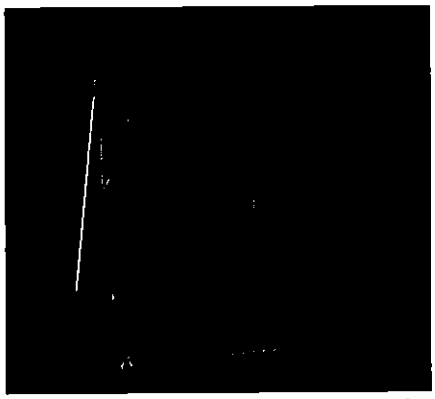
สำหรับการจัดทำโครงสร้างของฝั่งลูกข่าย จะใช้อะคริลิคมาทำแผ่นรองของกล่องอเนกประสงค์ โดยตัดแผ่นอะคริลิคให้เป็นสี่เหลี่ยมที่สามารถวางลงบนกล่องได้และยึดแผ่นอะคริลิคไว้กลับกล่องอเนกประสงค์ จากนั้นจึงนำอุปกรณ์ต่าง ๆ มายึดไว้กลับแผ่นอะคริลิคที่ติดไว้ในกล่อง ดังรูป

เมื่อนำอุปกรณ์ต่าง ๆ ประกอบเข้ากับกล่องอเนกประสงค์ดังรูปแล้ว ขั้นตอนต่อไปจะต่อสายไฟ ดังรูป



รูปที่ 3.18 ลักษณะภายในของโครงสร้างฝั่งลูกข่าย

1. จ่ายไฟ 5 โวลต์ จากบอร์ดแปลงไฟต่อไฟเลี้ยงให้กับชุดแปลงสัญญาณระหว่างอาร์เอส232 กับคลื่นความถี่ไร้สาย
2. จ่ายไฟ 5 โวลต์จากบอร์ดแปลงไฟให้กับบอร์ดไมโครคอนโทรลเลอร์
3. จ่ายไฟ 5 โวลต์จากบอร์ดแปลงไฟให้กับจอแสดงผลแอลซีดี
4. ต่อสายไฟที่พอร์ตอนุกรม1 บนบอร์ดไมโครคอนโทรลเลอร์กับชุดแปลงสัญญาณระหว่างอาร์เอส 232 กับคลื่นความถี่ไร้สาย
5. ต่อสายไฟที่พอร์ตอนุกรม2 บนบอร์ดไมโครคอนโทรลเลอร์กับโมดูลของระบบบอกตำแหน่งพิกัดโลก
6. ต่อแบตเตอรี่เข้ากับบอร์ดแปลงไฟ 12 โวลต์ เป็น 5 โวลต์
7. ต่อสายไฟจากจอแสดงผลแอลซีดีเข้ากับพอร์ตซีแอลซีดีบนไมโครคอนโทรลเลอร์
8. ต่อสายอากาศเข้าที่โมดูลของระบบบอกตำแหน่งพิกัดโลก



รูปที่ 3.19 ลักษณะภายนอกของโครงสร้างฝั่งลูกข่าย

3.4 Data package

คาต้าแพ็กเกจใช้ในการเก็บข้อมูลตำแหน่งของรถไฟฟ้าแต่ละคัน เพื่อที่จะนำค่าที่ได้ไปแสดงตำแหน่งของรถแต่ละคัน โดยคาต้าแพ็กเกจในระบบนี้จะประกอบไปด้วย

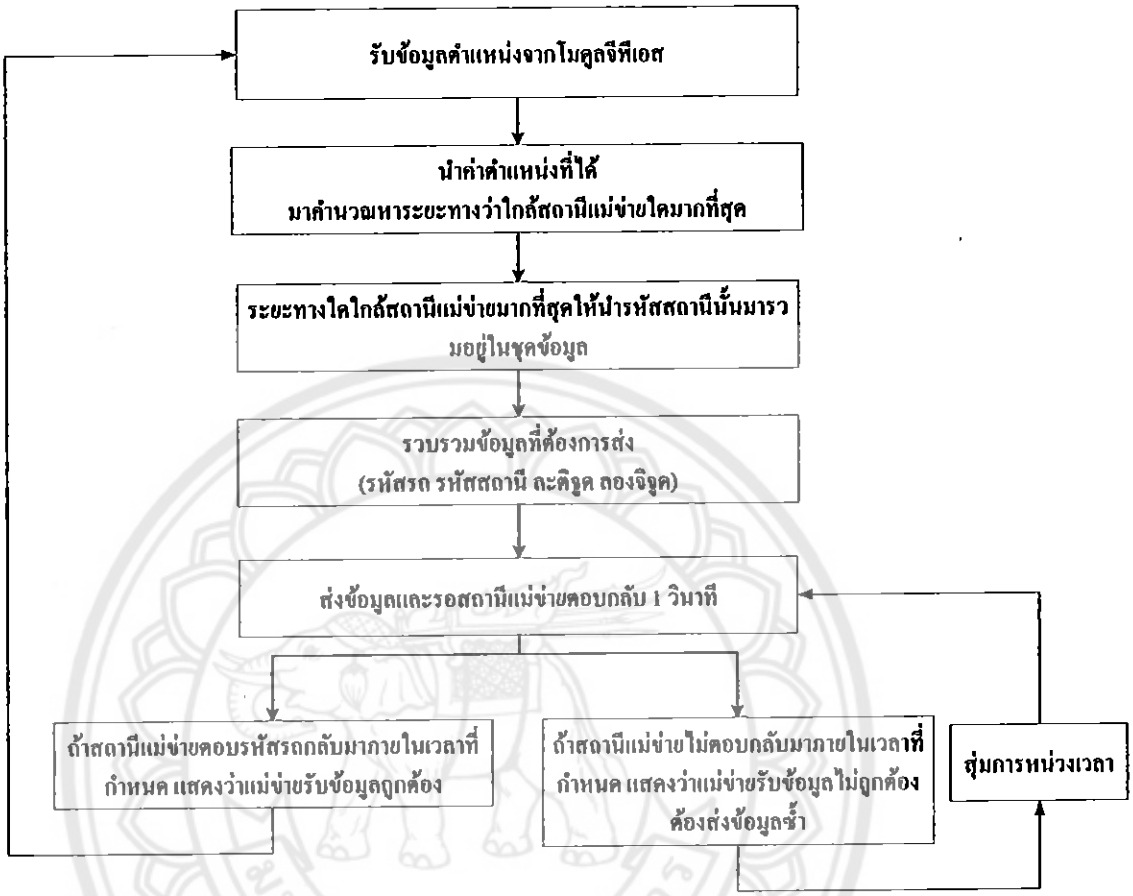
ตารางที่ 3.1 แสดง Data package

Start	Station	EVT_ID	Latitude	Longitude	Checksum	END

- **Start** คือ บิตเริ่มต้น บอกให้ทราบว่าเริ่มมีการส่งข้อมูล
- **Station** คือ ส่วนที่บอกว่ารถไฟฟ้าอยู่ใกล้สถานีใด
- **EVT_ID** คือ รหัสประจำของรถไฟฟ้า
- **Latitude** คือ ค่าละติจูดที่ได้จากโมดูลของระบบบอกตำแหน่งพิกัดโลก
- **Longitude** คือ ค่าลองจิจูดที่ได้จากโมดูลของระบบบอกตำแหน่งพิกัดโลก
- **Longitude** คือ ค่าลองจิจูดที่ได้จากโมดูลของระบบบอกตำแหน่งพิกัดโลก
- **End** เป็นส่วนที่กำหนดว่าการส่งข้อมูลนั้นเสร็จสิ้นแล้ว

3.5 ระบบโดยรวมฝั่งลูกข่าย

ในส่วนของระบบโดยรวมฝั่งลูกข่ายนี้ จะเป็นการออกแบบโปรแกรมที่จะนำค่าละติจูดและลองจิจูดที่ได้จากโมดูลจีพีเอสบนรถไฟฟ้า ส่งไปยังป้ายจอดรถ เพื่อที่จะนำค่าละติจูดและลองจิจูดไปใช้แสดงผลต่อไป



รูปที่ 3.20 แผนภาพแสดงระบบโดยรวมฝั่งลูกข่าย

จากแผนภาพ จะมีการทำงานดังนี้

1. รับข้อมูลละติจูดและลองจิจูดจากโมดูลจีพีเอส ถ้าจีพีเอสรับค่าละติจูดและลองจิจูดไม่ได้ จอแสดงผลแอลซีดีที่ติดอยู่บนฝากล่องฝั่งลูกข่าย จะแสดงข้อความว่าไม่มีสัญญาณจีพีเอส

2. เมื่อได้รับข้อมูลละติจูดและลองจิจูดมาครบแล้ว ก็จะทำการแปลงชนิดข้อมูลของตำแหน่งจากชนิดอาเรย์ข้อความเปลี่ยนเป็นชนิดตัวเลข เพื่อที่จะนำค่าตัวเลขนั้นไปคำนวณหาระยะทาง

3. นำค่าตำแหน่งที่ได้มาคำนวณระยะทาง ระหว่างค่าละติจูดและลองจิจูดของลูกข่ายกับค่าละติจูดและลองจิจูดของแม่ข่ายแต่ละสถานี โดยใช้สูตร

$$\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \qquad (3.1)$$

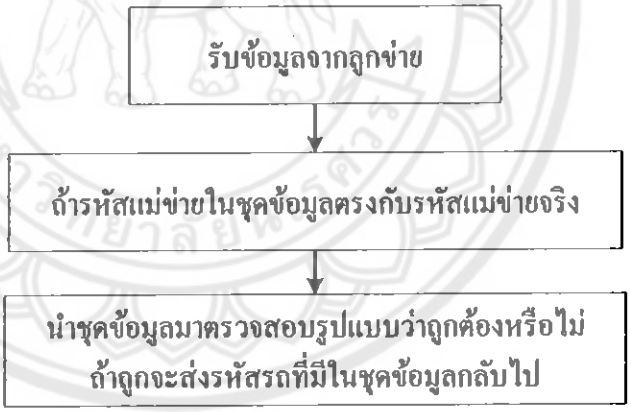
4. ระยะทางจากลูกข่ายกับแม่ข่ายสถานีที่ไหนใกล้สุด ให้นำรหัสของแม่ข่ายนี้ไปรวบรวมลงในชุดข้อมูล

5. รวบรวมข้อมูลทั้งหมดใส่ลงในอาเรย์ของชุดข้อมูลที่จะส่ง โดยข้อมูลที่จะส่งนั้นประกอบไปด้วย รหัสประจำของลูกข่าย รหัสแม่ข่าย ละติจูด ลองจิจูด บิตตรวจสอบข้อมูล

6. ส่งข้อมูลไปยังแม่ข่ายและรอแม่ข่ายที่ใกล้สุดตอบกลับมา 1 วินาที ถ้าภายในเวลา 1 วินาที แม่ข่ายที่ใกล้ที่สุดนั้นยังไม่ตอบกลับมา แสดงว่ามีการชนกันของข้อมูลหรือข้อมูลขาดหายไป ดังนั้นจึงต้องส่งค่าเก่ากลับไปโดยการจะส่งข้อมูลเก่านั้นจะใช้วิธีการสุ่มเวลาในการหน่วงเวลา เพื่อป้องกันไม่ให้เกิดปัญหาการชนกันของข้อมูลและในการส่งข้อมูลเก่านั้นจะไม่ให้เกิน 5 ครั้ง ถ้าแม่ข่ายตอบรหัสลูกข่ายกลับมาภายในเวลาที่กำหนด ระบบก็จะเริ่มรับค่าตำแหน่งจากจีพีเอสและทำกระบวนการดังกล่าวใหม่

3.6 ระบบโดยรวมฝั่งแม่ข่าย

ในส่วนของระบบโดยรวมฝั่งแม่ข่ายนั้น จะเป็นการรับข้อมูลตำแหน่งจากโมดูลจีพีเอสบนรถไฟฟ้า มาเก็บไว้ที่ไมโครคอนโทรลเลอร์ที่ป้ายจอดเพื่อที่จะนำค่าตำแหน่งของรถไฟฟ้าที่ได้ไปแสดงผลต่อไป



รูปที่ 3.21 แผนภาพแสดงระบบโดยรวมฝั่งแม่ข่าย

จากแผนภาพ จะมีการทำงานดังนี้

1. รับข้อมูลจากลูกข่ายมาเก็บไว้ในอาเรย์

2. นำอาเรย์ที่รับได้นั้นมาตรวจสอบรูปแบบว่ารูปแบบของชุดข้อมูลที่รับได้นั้นถูกต้องหรือไม่ โดยดูจาก

อาเรย์ตัวแรกต้องเป็นบิตเริ่มต้น

อาเรย์ตัวที่สองต้องเป็น รหัสของแม่ข่ายเครื่องนี้

อาเรย์ตัวที่หก ตัวที่เก้าและตัวที่สิบเอ็ด ต้องเป็นตัว :

อาเรย์ตัวสุดท้ายต้องเป็นบิตสุดท้ายคือ \n

3. ถ้าชุดข้อมูลถูกต้องตามรูปแบบให้ส่งรหัสลูกข่ายที่ได้จากชุดข้อมูลที่ตรวจสอบนั้นส่งกลับไป

4. ถ้าไม่ถูกต้องตามรูปแบบ แล้วไม่ต้องส่งค่าอะไรไปรอรับชุดข้อมูลใหม่

บทที่ 4

ผลการทดลอง

ในบทนี้จะกล่าวถึงการทดสอบการทำงานของโปรแกรมที่พัฒนาขึ้นมา ซึ่งสามารถแบ่งการทดสอบออกเป็นสามส่วนคือ การทดสอบการส่งข้อมูลผ่านชุดแปลงสัญญาณระหว่างอาร์เอส232 กับคลื่นความถี่ไร้สาย การทดสอบการรับข้อมูลจากโมดูลจีพีเอสและขั้นตอนการทดสอบของระบบ

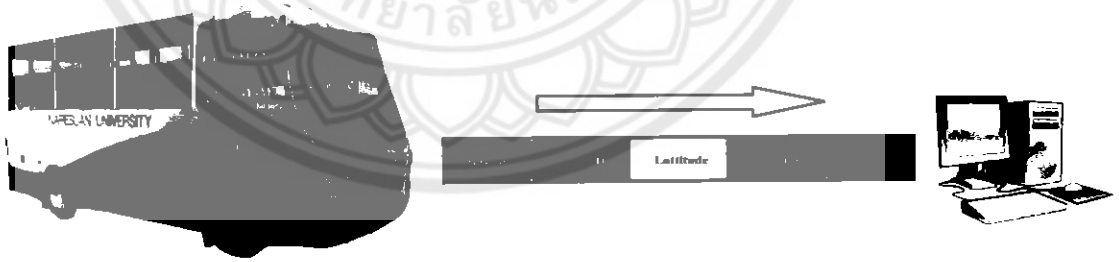
4.1 การทดสอบการส่งข้อมูลผ่านชุดแปลงสัญญาณระหว่างอาร์เอส232 กับคลื่นความถี่ไร้สาย

ในการทดสอบการส่งข้อมูลผ่านชุดแปลงสัญญาณนี้สามารถแบ่งหัวข้อการทดสอบออกเป็นสองส่วนคือ การทดสอบการส่งข้อมูลจากไมโครคอนโทลเลอร์ผ่านชุดแปลงสัญญาณไปยังคอมพิวเตอร์ และการทดสอบความถูกต้องในการส่งข้อมูลของชุดแปลงสัญญาณ

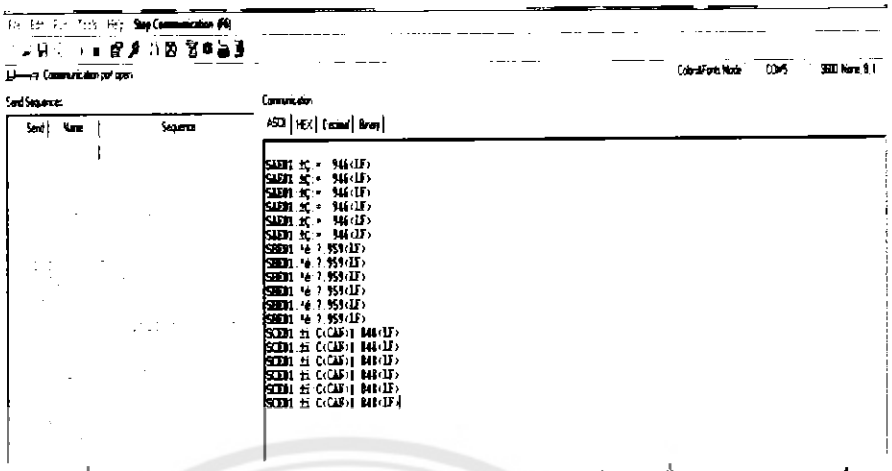
4.1.1 การทดสอบการส่งข้อมูลจากไมโครคอนโทลเลอร์ผ่านชุดแปลงสัญญาณไปยังคอมพิวเตอร์

การทดสอบการส่งข้อมูลแบ่งออกเป็นสองกรณี คือ กรณีที่ระบบฝั่งลูกข่ายส่งชุดข้อมูล (Data Package) ให้กับเครื่องคอมพิวเตอร์ และกรณีที่ระบบฝั่งแม่ข่ายส่งค่ารหัสลูกข่ายที่ส่งข้อมูลถูกต้องกลับมายังเครื่องคอมพิวเตอร์

1. กรณีที่ระบบฝั่งลูกข่ายส่งชุดข้อมูล(Data Package) ให้กับเครื่องคอมพิวเตอร์



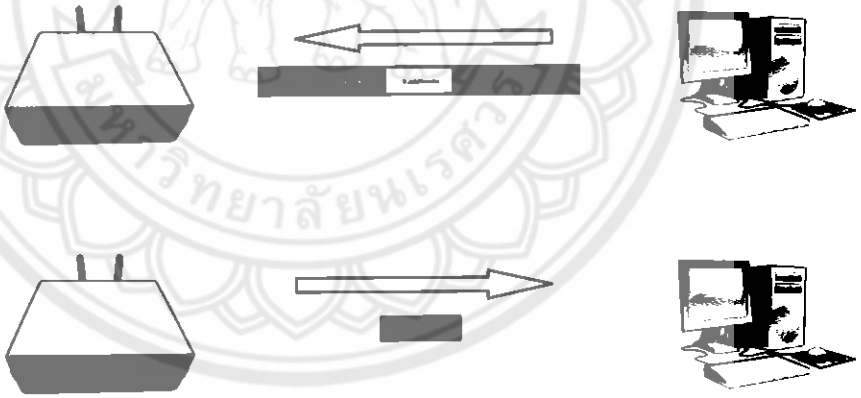
รูปที่ 4.1 ลูกข่ายส่งชุดข้อมูล (Data Package) ให้กับเครื่องคอมพิวเตอร์



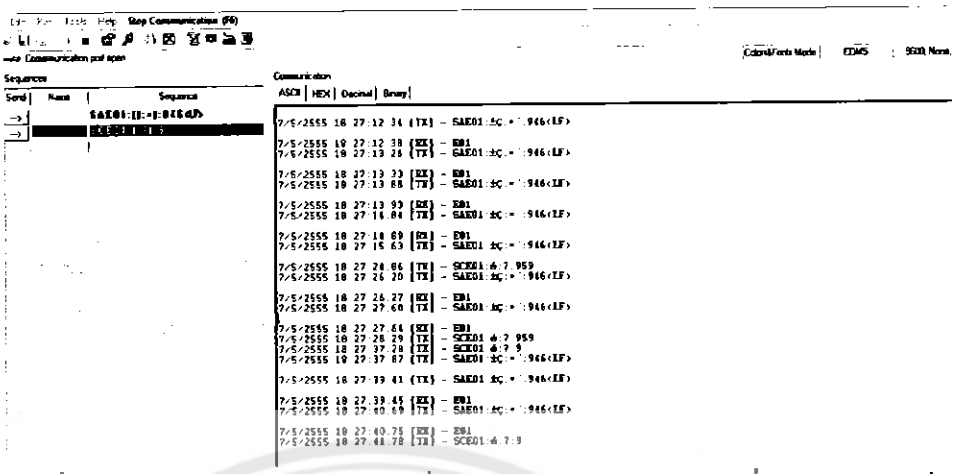
รูปที่ 4.2 แสดงการส่งชุดข้อมูล (Data Package) ให้กับเครื่องคอมพิวเตอร์

จากรูปที่ 4.2 เป็นการส่งชุดข้อมูลไปยังระบบฝั่งแม่ข่ายที่จำลองโดยใช้คอมพิวเตอร์แทน โดยกำหนดให้ข้อมูลในการส่งคือ ไบต์เริ่มต้น รหัสแม่ข่ายที่ลูกข่ายนี้อยู่ใกล้สุด เลขประจำตัวของลูกข่าย ละติจูด ลองจิจูด ไบต์ตรวจสอบความถูกต้องของข้อมูล ไบต์สุดท้าย

2. กรณีที่ระบบฝั่งแม่ข่ายส่งค่ารหัสลูกข่ายที่ส่งข้อมูลถูกต้องกลับมายังเครื่องคอมพิวเตอร์



รูปที่ 4.3 ลูกข่ายส่งชุดข้อมูล (Data Package) ให้กับเครื่องคอมพิวเตอร์



รูปที่ 4.4 แสดงการส่งค่ารหัสลูกข่ายที่ส่งข้อมูลถูกต้องกลับมายังเครื่องคอมพิวเตอร์

จากรูปที่ 4.4 เป็นการส่งค่ารหัสลูกข่ายของเครื่องที่ส่งข้อมูลมาถูกต้องไปยังระบบลูกข่ายที่จำลองโดยใช้คอมพิวเตอร์แทน โดยจะส่งค่ารหัสลูกข่ายของเครื่องที่ส่งข้อมูลมาถูกต้องและจะไม่ส่งข้อมูลใด ๆ เมื่อชุดข้อมูล (Data Package) ไม่ถูกต้อง

4.1.2 การทดสอบความถูกต้องในการส่งข้อมูลของชุดแปลงสัญญาณ

ในโครงงานนี้ใช้ชุดแปลงสัญญาณ รุ่น ET-RF 24G v2.0 กำลังส่ง +20dBm เสาอากาศภายนอก 2.4 กิกะเฮิร์ซ ซึ่งตามดาตาชุดแปลงสัญญาณรุ่นนี้สามารถส่งข้อมูลได้ไกลมากกว่า 100 เมตร แต่ในทางปฏิบัติการส่งสัญญาณจะถูกลดทอนลงโดยสภาพแวดล้อม ซึ่งการทดลองนี้ได้ทดสอบความถูกต้องในการส่งข้อมูลของชุดแปลงสัญญาณในสภาพแวดล้อมต่าง ๆ ซึ่งได้ผลการทดลองดังนี้

ตารางที่ 4.1 แสดงการส่งข้อมูลของชุดแปลงสัญญาณ ในที่โล่งแจ้ง ถนนหน้าตึกวิศวกรรมศาสตร์ถึงทางออกประตู 5

ครั้งที่	1	2	3	4	5	6	7	8	9	10
ระยะ 50 เมตร	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
ระยะ 60 เมตร	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
ระยะ 70 เมตร	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
ระยะ 80 เมตร	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
ระยะ 90 เมตร	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
ระยะ 100 เมตร	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
ระยะ 120 เมตร	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes

ระยะ 140 เมตร	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
ระยะ 160 เมตร	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
ระยะ 180 เมตร	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
ระยะ 200 เมตร	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
ระยะ 250 เมตร	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
ระยะ 300 เมตร	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
ระยะ 350 เมตร	Yes	Yes	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes
ระยะ 400 เมตร	No	Yes	Yes	Yes	No	Yes	Yes	Yes	No	Yes
ระยะ 450 เมตร	No	No	Yes	No	Yes	Yes	No	Yes	Yes	No
ระยะ 500 เมตร	Yes	No	Yes	No	No	No	No	Yes	No	Yes

จากตารางที่ 4.1 สรุปได้ว่า ในการรับส่งข้อมูลในที่โล่งแจ้งนั้น มีระยะทางในการรับส่งได้ไกลและมีความถูกต้อง แต่เมื่อการรับส่งข้อมูลไกลมากขึ้นความถูกต้องจะลดหายไป เนื่องจากเกิดการเบาบางของสัญญาณ

ตารางที่ 4.2 แสดงการส่งข้อมูลของชุดแปลงสัญญาณ ในทางโค้งที่มีต้นไม้ ถนนหน้าตึก วิศวกรรมศาสตร์ถึงหน้าตึก มน.นิเวศ

ครั้งที่	1	2	3	4	5	6	7	8	9	10
ระยะ 50 เมตร	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
ระยะ 60 เมตร	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
ระยะ 70 เมตร	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
ระยะ 80 เมตร	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
ระยะ 90 เมตร	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
ระยะ 100 เมตร	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
ระยะ 120 เมตร	Yes	Yes	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes
ระยะ 140 เมตร	Yes	No	No	Yes	Yes	Yes	No	Yes	No	No
ระยะ 160 เมตร	No	Yes	No	No	Yes	No	No	Yes	No	No
ระยะ 180 เมตร	Yes	No	Yes	Yes	Yes	Yes	No	Yes	Yes	Yes
ระยะ 200 เมตร	Yes	Yes	Yes	Yes	No	Yes	Yes	Yes	Yes	Yes
ระยะ 250 เมตร	Yes	Yes	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes
ระยะ 300 เมตร	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes

ระยะ 350 เมตร	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes

จากตารางที่ 4.2 สรุปได้ว่า ในการรับส่งข้อมูลในที่มีสิ่งกีดขวางนั้น ความถูกต้องในการรับส่งข้อมูลจะขึ้นอยู่กับว่ามีสิ่งกีดขวางขวางช่องทางการสื่อสารหรือไม่ จากตารางจะเห็นได้ว่าช่วงระยะทาง 140-180 จะมีสิ่งกีดขวางอยู่ซึ่งสิ่งกีดขวางในการทดลองนี้คือต้นไม้ตรงทางโค้งพอดี

4.2 การทดสอบการรับข้อมูลจากโมดูลจีพีเอส

การทดสอบการรับข้อมูลตำแหน่งพิกัดโลกจากโมดูลจีพีเอสแบ่งออกเป็นสองกรณี คือ กรณีที่ให้คอมพิวเตอร์รับข้อมูลจากโมดูลจีพีเอส และกรณีที่ให้ระบบของฝั่งลูกข่ายรับข้อมูลจากจีพีเอส

4.2.1 กรณีที่ให้คอมพิวเตอร์รับข้อมูลจากโมดูลจีพีเอส

File Edit Format View Help

```
$GPGSV,3,2,10,14,38,295,41,18,56,039,38,25,26,136,39,12,21,100,29*70
$GPGSV,3,3,10,21,59,186,41,22,38,337,41*74
$GPGLL,1644.78438,N,10011.39584,E,103728.00,A,A*63
$GPZDA,103728.00,09,01,2012,00,00*60
$GPRMC,103729.00,A,1644.77933,N,10011.39918,E,20.688,147.52,090112,,,A*5F
$GPVTG,147.52,T,,M,20.688,N,38.335,K,A*32
$GPGGA,103729.00,1644.77933,N,10011.39918,E,1,08,1.24,43.0,M,-33.0,M,,*42
$GPGSA,A,3,31,09,14,18,25,12,21,22,,,,,2.83,1.24,2.54*0F
$GPGSV,3,1,10,31,20,212,34,09,30,037,29,40,,,38,27,09,036,*4F
$GPGSV,3,2,10,14,38,295,41,18,56,039,42,25,26,136,36,12,21,100,24*7F
$GPGSV,3,3,10,21,59,186,40,22,38,337,42*76
$GPGLL,1644.77933,N,10011.39918,E,103729.00,A,A*62
$GPZDA,103729.00,09,01,2012,00,00*61
$GPRMC,103730.00,A,1644.77422,N,10011.40225,E,20.164,149.38,090112,,,A*56
$GPVTG,149.38,T,,M,20.164,N,37.364,K,A*3E
$GPGGA,103730.00,1644.77422,N,10011.40225,E,1,08,1.24,42.9,M,-33.0,M,,*44
$GPGSA,A,3,31,09,14,18,25,12,21,22,,,,,2.83,1.24,2.54*0F
$GPGSV,3,1,10,31,20,212,37,09,30,037,24,40,,,33,27,09,036,*4A
$GPGSV,3,2,10,14,38,295,39,18,56,039,42,25,26,136,30,12,21,100,21*73
$GPGSV,3,3,10,21,59,186,41,22,38,337,42*77
$GPGLL,1644.77422,N,10011.40225,E,103730.00,A,A*6C
$GPZDA,103730.00,09,01,2012,00,00*69
$GPRMC,103731.00,A,1644.76955,N,10011.40442,E,18.966,150.95,090112,,,A*52
$GPVTG,150.95,T,,M,18.966,N,35.144,K,A*32
$GPGGA,103731.00,1644.76955,N,10011.40442,E,1,08,1.02,43.1,M,-33.0,M,,*43
$GPGSA,A,3,31,09,14,18,25,12,21,22,,,,,2.41,1.02,2.19*0C
$GPGSV,3,1,10,31,20,212,38,09,30,037,33,40,,,40,27,09,036,*47
```

รูปที่ 4.5 แสดงการรับข้อมูลจากโมดูลจีพีเอสเข้าคอมพิวเตอร์ในที่โล่งแจ้ง

```
$GPGGA,000039.042,,,,,0,00,,,M,0.0,M,,0000*5A<CR><LF>
$GPGLL,,,,,000039.042,V,N*76<CR><LF>
$GPGSA,A,1,,,,,,,,,,,,,,,*1E<CR><LF>
$GPGSV,3,1,12,19,00,000,17,01,00,000,,02,00,000,,03,00,000,*76<CR><LF>
$GPGSV,3,2,12,04,00,000,,05,00,000,,06,00,000,,07,00,000,*7B<CR><LF>
$GPGSV,3,3,12,08,00,000,,09,00,000,,10,00,000,,11,00,000,*7A<CR><LF>
$GPRMC,000039.042,V,,,,,,,191210,,,N*4B<CR><LF>
$GPVTG,,T,,M,,N,,K,N*2C<CR><LF>
$GPGGA,000040.042,,,,,0,00,,,M,0.0,M,,0000*54<CR><LF>
$GPGLL,,,,,000040.042,V,N*76<CR><LF>
$GPGSA,A,1,,,,,,,,,,,,,,,*1E<CR><LF>
$GPGSV,3,1,12,19,00,000,17,01,00,000,,02,00,000,,03,00,000,*76<CR><LF>
$GPGSV,3,2,12,04,00,000,,05,00,000,,06,00,000,,07,00,000,*7B<CR><LF>
$GPGSV,3,3,12,08,00,000,,09,00,000,,10,00,000,,11,00,000,*7A<CR><LF>
$GPRMC,000040.042,V,,,,,,,191210,,,N*45<CR><LF>
$GPVTG,,T,,M,,N,,K,N*2C<CR><LF>
$GPGGA,000041.042,,,,,0,00,,,M,0.0,M,,0000*55<CR><LF>
$GPGLL,,,,,000041.042,V,N*79<CR><LF>
$GPGSA,A,1,,,,,,,,,,,,,,,*1E<CR><LF>
$GPGSV,3,1,12,19,00,000,17,01,00,000,,02,00,000,,03,00,000,*76<CR><LF>
$GPGSV,3,2,12,04,00,000,,05,00,000,,06,00,000,,07,00,000,*7B<CR><LF>
$GPGSV,3,3,12,08,00,000,,09,00,000,,10,00,000,,11,00,000,*7A<CR><LF>
$GPRMC,000041.042,V,,,,,,,191210,,,N*44<CR><LF>
$GPVTG,,T,,M,,N,,K,N*2C<CR><LF>
$GPGGA,000042.042,,,,,0,00,,,M,0.0,M,,0000*56<CR><LF>
$GPGLL,,,,,000042.042,V,N*7A<CR><LF>
$GPGSA,A,1,,,,,,,,,,,,,,,*1E<CR><LF>
$GPGSV,3,1,12,19,00,000,16,01,00,000,,02,00,000,,03,00,000,*77<CR><LF>
$GPGSV,3,2,12,04,00,000,,05,00,000,,06,00,000,,07,00,000,*7B<CR><LF>
$GPGSV,3,3,12,08,00,000,,09,00,000,,10,00,000,,11,00,000,*7A<CR><LF>
$GPRMC,000042.042,V,,,,,,,191210,,,N*47<CR><LF>
$GPVTG,,T,,M,,N,,K,N*2C<CR><LF>
$GPGGA,000043.042,,,,,0,00,,,M,0.0,M,,0000*57<CR><LF>
$GPGLL,,,,,000043.042,V,N*7B<CR><LF>
$GPGSA,A,1,,,,,,,,,,,,,,,*1E<CR><LF>
$GPGSV,3,1,12,19,00,000,16,01,00,000,,02,00,000,,03,00,000,*77<CR><LF>
$GPGSV,3,2,12,04,00,000,,05,00,000,,06,00,000,,07,00,000,*7B<CR><LF>
$GPGSV,3,3,12,08,00,000,,09,00,000,,10,00,000,,11,00,000,*7A<CR><LF>
$GPRMC,000043.042,V,,,,,,,191210,,,N*46<CR><LF>
$GPVTG,,T,,M,,N,,K,N*2C<CR><LF>
$GPGGA,000044.047,,,,,0,00,,,M,0.0,M,,0000*55<CR><LF>
$GPGLL,,,,,000044.047,V,N*79<CR><LF>
$GPGSA,A,1,,,,,,,,,,,,,,,*1E<CR><LF>
$GPGSV,3,1,12,19,00,000,16,01,00,000,,02,00,000,,03,00,000,*77<CR><LF>
$GPGSV,3,2,12,04,00,000
```

รูปที่4.6 แสดงการรับข้อมูลจากโมดูลจีพีเอสเข้าคอมพิวเตอร์ในอาคาร

จากรูปที่ 4.5 เป็นการแสดงการรับข้อมูลจากโมดูลจีพีเอสเข้าคอมพิวเตอร์ในที่โล่งแจ้งจะสามารถรับค่าตำแหน่งพิกัดโลกได้โดยดูจากข้อมูล $GPRMC ตัวที่สองของบรรทัดนี้ ถ้าเป็นตัวอักษรเอแสดงว่าสามารถรับค่าตำแหน่งพิกัดโลกได้ แต่ถ้าเป็นตัวอักษรตัววี จะแสดงว่าไม่สามารถรับค่าตำแหน่งพิกัดโลกได้ ดังรูป 4.6

4.2.2 กรณีที่ให้ระบบของฝั่งลูกข่ายรับข้อมูลจากจีพีเอส

รูปที่ 4.7 แสดงการรับข้อมูลจากโมดูลจีพีเอสเข้าระบบของฝั่งลูกข่าย

รูปที่ 4.8 แสดงการรับข้อมูลจากโมดูลจีพีเอสเข้าระบบของฝั่งลูกข่าย

จากรูปที่ 4.7 เป็นการแสดงการรับข้อมูลจากโมดูลจีพีเอสเข้าระบบของฝั่งลูกข่าย โดยจะแสดงค่าตำแหน่งพิกัดโลกผ่านทางจอแอลซีดี ซึ่งถ้าตำแหน่งเปลี่ยนไปค่าตำแหน่งพิกัดโลกที่จอแสดงผลแอลซีดีก็จะเปลี่ยนตามด้วย แต่ถ้าไม่สามารถรับค่าตำแหน่งพิกัดโลกได้ระบบของฝั่งลูกข่ายก็จะแสดงผลว่าไม่สามารถรับสัญญาณจีพีเอสได้ ดังรูปที่ 4.8

4.3 ขั้นตอนการทดสอบของระบบ

1. นำอุปกรณ์ฝั่งแม่ข่ายเครื่องแรก (รหัสแม่ข่ายเอ) มาติดตั้งที่สถานีจอดรถไฟฟ้าป้ายแรก ดังรูป

รูปที่ 4.9 แสดงสถานีจอดรถไฟฟ้าป้ายแรก

*กำหนดให้สถานีจอดรถไฟฟ้าป้ายแรกอยู่ที่หน้าคณะวิทยาศาสตร์สาขาคณิตศาสตร์

2. นำอุปกรณ์ฝั่งแม่ข่ายเครื่องที่สอง (รหัสแม่ข่ายบี) มาติดตั้งที่สถานีจอดรถไฟฟ้าป้ายที่สอง ดังรูป

รูปที่ 4.10 แสดงสถานีจอดรถไฟฟ้าป้ายที่สอง

*กำหนดให้สถานีจอดรถไฟฟ้าป้ายที่สองอยู่ที่หน้าทางเข้าโรงอาหารคณะวิทยาศาสตร์สาขาคณิตศาสตร์

3. นำอุปกรณ์ฝั่งแม่ข่ายเครื่องที่สาม (รหัสแม่ข่ายซี) มาติดตั้งที่สถานีจอดรถไฟฟ้าป้ายที่สาม ดังรูป

รูปที่ 4.11 แสดงสถานีจอดรถไฟฟ้าป้ายที่สาม

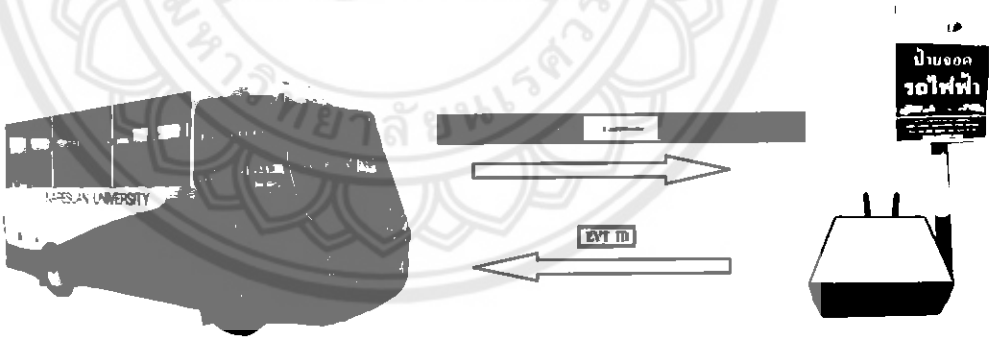
*กำหนดให้สถานีจอดรถไฟฟ้าป้ายที่สามอยู่ที่หน้าทางเข้าคณะวิทยาศาสตร์สาขาชีววิทยา

4. เสียบไฟบ้านให้แม่ข่ายทั้ง 3 เครื่อง

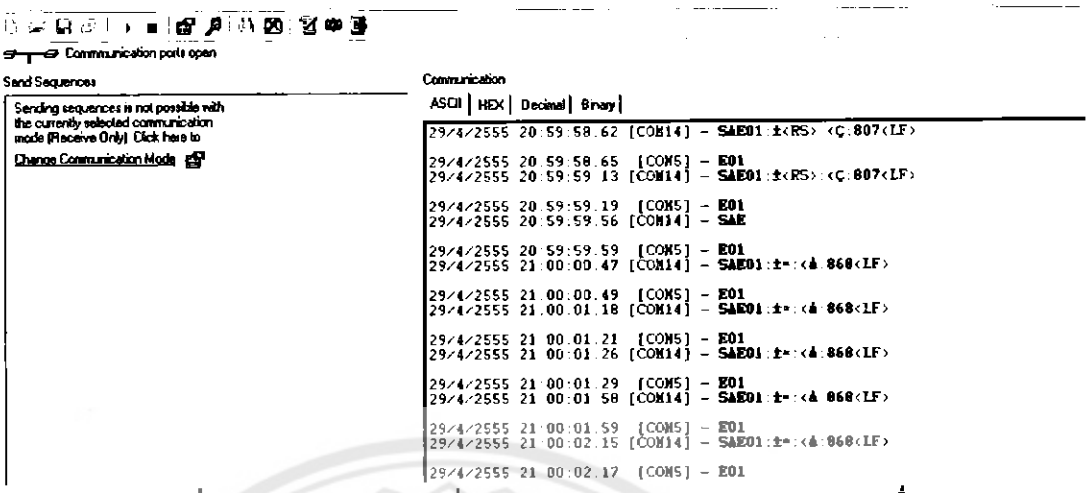
5. นำอุปกรณ์ฝั่งลูกข่ายไปติดตั้งบนรถโดยจำลองรถมอเตอร์ไซด์แทนรถไฟฟ้า จากนั้นเสียบแบตเตอรี่แล้วรอให้โมดูลจีพีเอสรับข้อมูลได้ประมาณ 15-20 วินาที โดยดูจากจอแอลซีดีว่าแสดงค่าละติจูด ลองจิจูดหรือไม่ ถ้าได้รับข้อมูลจากโมดูลจีพีเอสแล้วสามารถพร้อมทดสอบการทำงานได้เลย

รูปที่ 4.12 อุปกรณ์ฝั่งลูกข่ายที่ติดตั้งบนรถไฟฟ้าโดยจำลองรถมอเตอร์ไซด์แทนรถไฟฟ้า

4.3.1 ข้อมูลการติดต่อสื่อสารกันระหว่างแม่ข่ายกับลูกข่าย 1 เครื่อง



รูปที่ 4.13 การติดต่อสื่อสารกันระหว่างแม่ข่ายกับลูกข่าย 1 เครื่อง

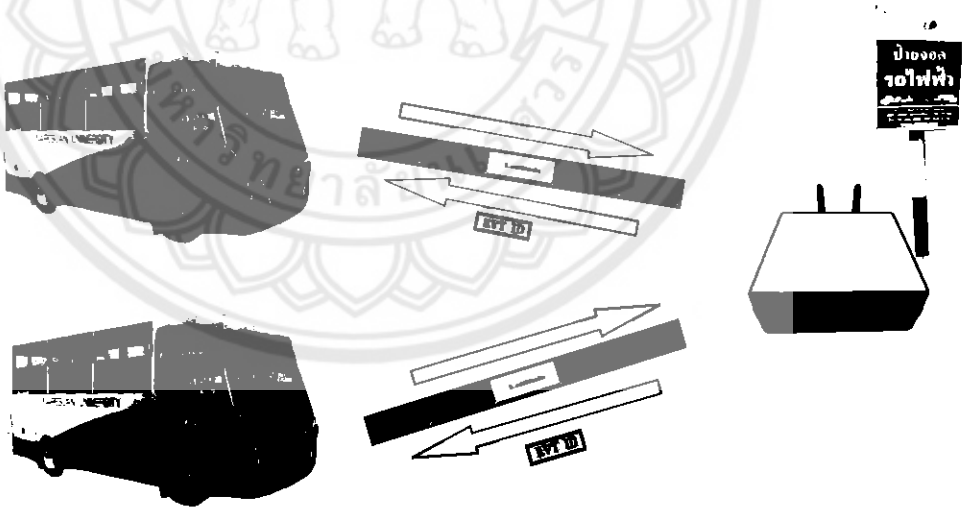


รูปที่ 4.14 ข้อมูลการติดต่อสื่อสารกันระหว่างแม่ข่ายกับลูกข่าย 1 เครื่อง

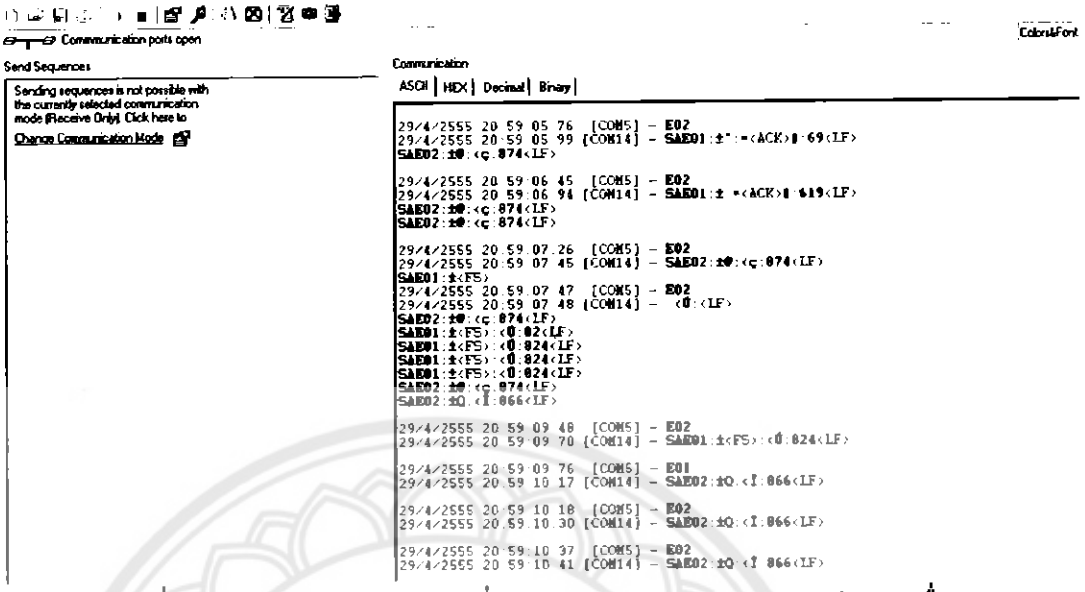
*สีน้ำเงินคือฝั่งแม่ข่ายเอ และสีแดงคือฝั่งลูกข่าย

จากรูป 4.14 จะเห็นว่า ลูกข่าย E01 ส่งชุดข้อมูลมาให้แม่ข่ายเอ และฝั่งแม่ข่ายเอตรวจสอบรูปแบบของชุดข้อมูลว่าถูกต้องหรือไม่ จากรูปนี้จะเห็นว่า ลูกข่าย E01 ส่งชุดข้อมูลมาถูกต้องฝั่งแม่ข่ายเอจึงส่งรหัสของลูกข่ายกลับไปเพื่อแสดงว่าลูกข่าย E01 ได้ส่งข้อมูลมาให้แม่ข่ายเอ ถูกต้องแล้ว

4.3.2 ข้อมูลการติดต่อสื่อสารกันระหว่างแม่ข่ายกับลูกข่าย 2 เครื่อง



รูปที่ 4.15 การติดต่อสื่อสารกันระหว่างแม่ข่ายกับลูกข่าย 1 เครื่อง



รูปที่ 4.16 แสดงข้อมูลการติดต่อสื่อสารกันระหว่างแม่ข่ายกับลูกข่าย **2 เครื่อง**

*สีน้ำเงินคือฝั่งแม่ข่ายเอ และสีแดงคือฝั่งลูกข่าย

จากรูป 4.16 จะเห็นว่า ลูกข่าย E01 และลูกข่าย E02 ส่งชุดข้อมูลมาให้แม่ข่ายเอ และฝั่งแม่ข่ายเอตรวจสอบรูปแบบของชุดข้อมูลว่าถูกต้องหรือไม่ จากรูปนี้จะเห็นว่า ลูกข่าย E02 ส่งชุดข้อมูลมาถูกต้องฝั่งแม่ข่ายเอจึงส่งรหัสของลูกข่ายกลับไปเพื่อแสดงว่าลูกข่าย E02 ได้ส่งข้อมูลมาให้แม่ข่ายเอถูกต้องแล้ว แต่ลูกข่าย E01 ส่งชุดข้อมูลมาผิดแม่ข่ายเอเลยไม่ตอบรหัสลูกข่าย E01 ลูกข่าย E01 เลยต้องใช้หลักการ Random access คือสุ่มการหน่วงเวลา ทั้งหมด 5 ครั้ง โดยครั้งที่ 1 ถึงครั้งที่ 4 หน่วงเวลาแล้วข้อมูลไปชนกับลูกข่าย E02 ทำให้ลูกข่าย E02 ก็ต้องใช้หลักการเดียวกัน โดยลูกข่าย E02 สุ่ม 2 ครั้งแล้วลูกข่าย E01 สุ่มครั้งที่ 5 ทำให้ข้อมูลทั้งสองลูกข่ายไม่ชนกันแล้ว

บทที่ 5
บทสรุปและข้อเสนอแนะ

5.1 สรุปผลการดำเนินโครงการ

โครงงานนี้มีการพัฒนาขึ้นเพื่อสร้างอำนวยความสะดวกสบายแก่ผู้ใช้บริการรถไฟฟ้าภายในมหาวิทยาลัยนเรศวรและช่วยให้ผู้ดูแลระบบขนส่งภายในมหาวิทยาลัยสามารถดูแลระบบได้ง่ายขึ้น เช่น ผู้ดูแลระบบสามารถดูว่ารถไฟฟ้าที่กำลังให้บริการอยู่นั้นอยู่ตำแหน่งใดบ้าง ออกนอกเส้นทางการให้บริการหรือไม่ โดยที่โครงงานนี้มีเป้าหมายเพื่อสร้างตัวต้นแบบการพัฒนาการติดตามรถไฟฟ้าให้กับมหาวิทยาลัยนเรศวร โดยโครงงานนี้ได้จำลองรถไฟฟ้า 2 คันและป้ายจอดรถไฟฟ้าอีก 3 ป้าย ซึ่งจากการทดลองพบว่าระบบการติดตามรถไฟฟ้าโดยระบบบอกตำแหน่งพิกัดโลกและการสื่อสารแบบไร้สาย สามารถรับค่าตำแหน่งของพิกัดโลกของรถไฟฟ้าหลายคันได้และชุดข้อมูลที่แม่ข่ายได้รับไปนั้นสามารถนำไปเป็นพัฒนาต่อเพื่อนำไปแสดงผล ดังนั้นสรุปได้ว่าโครงงานนี้สามารถทำงานได้ตามเป้าหมายที่ได้ตั้งไว้ข้างต้น

5.2 ปัญหาในการดำเนินงานและแนวทางการแก้ไขปัญหา

ตารางที่ 5.1 แสดงปัญหาในการดำเนินงานและแนวทางการแก้ไขปัญหา

ปัญหาในการดำเนินงาน	แนวทางการแก้ไขปัญหา
1. ออกแบบโปรโตคอลยุ่งยากและซับซ้อนเกินไป ทำให้เสียเวลาในการออกแบบและทดลองหลายรอบ	แก้ไขโปรโตคอลใหม่โดยให้ 1. ให้ฝั่งลูกข่ายหรือฝั่งรถไฟฟ้าส่งชุดข้อมูลให้อย่างเดียว 2. ฝั่งแม่ข่ายจะรับข้อมูลที่ฝั่งลูกข่ายส่งมาเพื่อตรวจสอบรูปแบบของชุดข้อมูลว่าถูกต้องหรือไม่ ถ้าถูกฝั่งแม่ข่ายจะตอบกลับว่าถูก 3. ถ้าแม่ข่ายไม่ตอบกลับ ลูกข่ายจะส่งค่าตำแหน่งเดิมไปใหม่ โดยจะส่งค่าซ้ำไม่เกิน 5 ครั้ง

2. เมื่อมีลูกข่าย 2 คัน อยู่ใกล้แม่ข่ายเดียวกัน จะเกิดการชนกันของข้อมูล	แก้ไขโดยใช้หลักการกำหนดการเข้าใช้ช่องสัญญาณหลายชุดที่ใช้ตัวกลางร่วมกัน (Multiple Access Protocols) โดยในโครงงานนี้จะใช้วิธีการสุ่มเวลา (Random Access Protocols) คือ สุ่มการหน่วงเวลาตอนที่เกิดการชนกันของข้อมูล
3. ถ้าถนนโค้งและมีต้นไม้หรือตึกบังการส่งสัญญาณของฝั่งลูกข่ายแล้ว ลูกข่ายจะส่งข้อมูลให้แม่ข่ายไม่ได้	แก้ปัญหาโดยการใช้แม่ข่ายเพิ่ม หรือว่าแม่ข่ายที่เป็นตัวเชื่อมสัญญาณตรงทางโค้ง เพื่อให้ลูกข่ายส่งข้อมูลให้แม่ข่ายที่เป็นตัวเชื่อมสัญญาณแล้วแม่ข่ายที่เป็นตัวเชื่อมสัญญาณนั้นจะส่งข้อมูลให้แม่ข่ายจริงที่รับข้อมูลไม่ได้โดยตรงจากลูกข่าย

5.3 ข้อเสนอแนะในการดำเนินโครงงาน

1. กล่องอุปกรณ์ของโครงงานอาจจะต้องเพิ่มพัดลมเพื่อระบายความร้อนของอุปกรณ์อิเล็กทรอนิกส์ต่าง ๆ ภายในกล่องอุปกรณ์นั้น

2. กล่องอุปกรณ์ของฝั่งลูกข่ายจะต้องมีขนาดใหญ่กว่านี้ โดยให้สามารถนำแบตเตอรี่เข้าไปอยู่ข้างในกล่องอุปกรณ์ด้วยและต้องทำช่องไว้สำหรับชาร์ตแบตเตอรี่ เพื่อความสะดวกในการใช้งาน

3. ควรเพิ่มวงจรเปรียบเทียบแรงดัน เพื่อเช็กว่าแบตเตอรี่ใกล้หมดหรือยัง โดยการแสดงการแจ้งเตือนแบตเตอรี่ใกล้หมดด้วยไฟแอลอีดีบนกล่องอุปกรณ์

4. ในการติดต่อสื่อสารกันด้วยชุดแปลงสัญญาณ ET-RF24G V2 นี้ จะมีปัญหาการชนกันของข้อมูล ถ้าต้องการหลีกเลี่ยงปัญหานี้ อาจเลือกใช้ Zigbee ในการติดต่อสื่อสารแทนชุดแปลงสัญญาณ ET-RF24G V2 ก็ได้

5. ถ้าต้องการให้ระยะทางในการรับส่งข้อมูลไกลมากยิ่งขึ้น ควรใช้การติดต่อสื่อสารแบบเครือข่าย 3G แต่การติดต่อแบบเครือข่าย 3G นี้จะมีค่าใช้จ่ายสูงกว่าเพราะว่าต้องเช่าสัญญาณ

เอกสารอ้างอิง

[1] GPSdeedee.(2010) **หลักการทำงานของ GPS** สืบค้นเมื่อ 10 กรกฎาคม 2544 http://www.gpsdeedee.com/index.php?lay=show&ac=article&Id=538825826

[2] **อ.ธนันต์ ศรีสกุล ,อ.เกียรติชัย บรรลุผลสกุล,รศ.ดร.มนตรี ศิริปรัชญานันท์ ระบบพิกัด** สืบค้นเมื่อ 10 กรกฎาคม 2554 http://wara.com/article-835.html

[3] Social-ave.exteen.com (5 Apr 2012) **รูปแสดงละติจูดลองจิจูดของโลก** สืบค้นเมื่อ 15 กรกฎาคม 2554 http://social-ave.exteen.com/

[4] pkgis2007(9 กุมภาพันธ์ 2009) **NMEA Sentence** สืบค้นเมื่อ 15 กรกฎาคม 2554 http://emap.wordpress.com/2009/02/09/nmea-sentence/

[5] asquare(28 sep 2009) **เทคนิคการแปลงหน่อยในระบบพิกัดแบบ DMS และ DD** สืบค้นเมื่อ 16 กรกฎาคม 2554 http://www.smart-mobile.com/forum/viewtopic.php?f=139&t=174585

[6] etteam **คู่มือการใช้งาน RS232 to RF-Wireless (RF2.4GHz) CONVERTER รุ่น ET-RF24G V2.0** สืบค้นเมื่อ 30 กรกฎาคม 2554 www.etteam.com/product/intf/man-ET-RF24Gv2.pdf

[7] นคร ภักดีชาติ , ชัยวัฒน์ ลิ้มพรจิตรวิไล **โมดูลสื่อสารข้อมูลอนุกรม USART ภายใน STM32** สืบค้นเมื่อ 31 กรกฎาคม 2554 จากหนังสือ ปฏิบัติการไมโครคอนโทรลเลอร์ ARM Cortex-M3 กับ STM32. พิมพ์ครั้งที่ 1. กรุงเทพฯ: บริษัท อินโนเวตีฟ เอ็กเพอริเมนท์ จำกัด, 2553.

[8] โอภาส ศิริครรชิตถาวร **พอร์ตอนุกรม RS232** สืบค้นเมื่อ 15 สิงหาคม 2554 จากหนังสือ เปิดโลก 32 บิตกับ STM32-Discovery. พิมพ์ครั้งที่ 1. กรุงเทพฯ: บริษัท อินโนเวตีฟ เอ็กเพอริเมนท์ จำกัด, 2553.

[9] Behrouz A.Forouzan **Multiple Access Protocols** สืบค้นเมื่อ 15 มีนาคม 2554 จาก Data Communications and networking แปลและเรียบเรียงโดย จักกริช พฤษการ สำนักพิมพ์ท็อป จำกัด

ภาคผนวก ก

ตัวอย่างโค้ดฝั่งลูกข่าย

ตัวอย่างโค้ดโปรแกรมเพื่อคัดกรองข้อมูลค่าตำแหน่งจากโมดูลจีพีเอส ซึ่งในโครงงานนี้จะเลือกใช้ค่า $GPRMC

```
void GPS(void){
 checkGPRMC = 0;
 while(checkGPRMC == 0){
    i = GetKeyUart2();
    if (i == (int)'$'&& j ==0){
      str[0] = i; j = 1; i = GetKeyUart2();
        if (i == (int)'G' && j ==1){
          str[1] = i; j = 2; i = GetKeyUart2();
            if (i == (int)'P'&& j ==2){
              str[2] = i; j = 3; i = GetKeyUart2();
                if (i == (int)'R'&& j ==3){
                  str[3] = i; j = 4; i = GetKeyUart2();
                    if(i == (int)'M'&& j ==4){
                      str[4] = i; j = 5; i = GetKeyUart2();
                      if (i == (int)'C'&& j ==5){
                       str[5] = i; j = 6; i = GetKeyUart2();
                      do{
                       str[j] = i; j = j + 1; i = GetKeyUart2();
                       checkGPRMC = 1;
                      } while (i != (int)'$');
                    }
                    else {j = 0;}
                  }
                  else {j = 0;}
                }
                else {j = 0;}
              }
              else {j = 0;}
            }
          else {j = 0 ;}
          }
    }
}
```

รูป ก1 โปรแกรมเพื่อคัดกรองข้อมูลจากโมดูลจีพีเอส

```
void PackAndSend(void){
  short c = 0, checkcomma = 0;
  while (c < j-2)
     {
        if(str[c] == (int)',')
        {
           checkcomma = checkcomma+1;
           while (checkcomma == 2)
           {
              c = c+1;
              if (checkcomma == 2){
                  while(xx && 1){
                  if (str[c] == (int)'A'){
                   Signal = str[c];
                   GPIO_SetBits(GPIOB, GPIO_Pin_8);
                   data[0] = 'S';
                   data[1] = indexStation;
                   data[2] = 'E';
                   data[3] = '0';
                   data[4] = '2';
                   data[5] = ':';
                   xx = 0;
                  }
                  else if (str[c] == (int)'V') {
                   checkcomma = checkcomma + 10;
                   Signal = str[c];
                   GPIO_ResetBits(GPIOB, GPIO_Pin_8);
                   xx = 0;
                  }
              }
              if(str[c] == (int)','){
                checkcomma = checkcomma+1;
              }
           }
          }
          xx = 1;
          while (checkcomma == 3)
          {
             c=c+1;
             if (checkcomma == 3){
              if(str[c] == (int)',')
              {
                checkcomma = checkcomma+1;
              }
              else if (Signal == (int)'A') {
                iData = iData + 1;
                data[iData] = str[c];
              }
           }
          }
          xx = 1;

          while (checkcomma == 5)
          {
             c = c+1;
             if (checkcomma == 5){
               while(xx && 1){
                  if (Signal == (int)'A') {
                    iData = iData + 1;
                    data[iData] = (int)':';
                    xx = 0;
                  }
                }
                  if(str[c] == (int)',')
                  {
                    checkcomma = checkcomma+1;
                    iData = iData + 1;
                    data[iData] = (int)':';
                  }
                  else if (Signal == 'A') {
                     iData = iData + 1;
                     data[iData] = str[c];
                  }
             }
           }// end checkcomma 5
        }// end ,
     c = c+1;
      }
```

รูป ก2 โปรแกรมคัดกรองและรวบรวมข้อมูลที่จะเก็บในคาต้าแพ็คเกต

```
if (Signal == (int)'V') {
  LCD_PrintString("#c No Signal GPS.");
   }
else{
            LCD_PrintString("#cLat");
            LCD_PrintFloat(LatF);

            LCD_PrintString("#nLong");
            LCD_PrintFloat(LongF);
}
```

รูป ก3 โปรแกรมแสดงข้อมูลบนหน้าจอแอลซีดี

จากโค้ด ก3 จะเห็นว่าถ้ารับสัญญาณจีพีเอสไม่ได้ หน้าจอแอลซีดีจะแสดงคำว่า No Signal GPS ถ้ารับค่าจากจีพีเอสได้จะแสดงค่าละติจูดและลองจิจูด

```
void Distance(void){
  LatF = (LatF*100);
  LongF = (LongF*100);

  for(i = 0;i<3;i=i+1){
    distanceLat[i] = pow((LatStation[i] - LatF),2);
    distanceLong[i] = pow((LongStation[i] - LongF),2);
    distance[i] = sqrt((distanceLat[i]+distanceLong[i]));
  }
}
```

รูป ก4 โปรแกรมคำนวณหาระยะทางระหว่างรถกับป้ายทุกป้าย

```
void CheckDistance(void){
  if((distance[0] < distance[1]) && (distance[0] < distance[2])){
    indexStation = 'A';
    data[1] = indexStation;
    Send();

  }
  else if((distance[1] <= distance[0]) && (distance[1] < distance[2])){
    indexStation = 'B';
    data[1] = indexStation;
    Send();
  }
  else if((distance[2] < distance[0]) && (distance[2] <= distance[1])){
    indexStation = 'C';
    data[1] = indexStation;
    Send();
  }
}
```

รูป ก5 โปรแกรมคำนวณหาว่ารถกับป้ายไหนมีระยะทางใกล้กว่ากัน

```
void Send(void){
GPIO_SetBits(GPIOB, GPIO_Pin_15);

    if (Signal == (int)'A'){
      for(i = 0; i <= 5; i = i + 1){
        data2[i] = data[i];
      }
        data2[6] = hByteLat;
        data2[7] = lByteLat;
        data2[8] = ':';
        data2[9] = hByteLong;
        data2[10] = lByteLong;
        data2[11] = ':';

          for(i = 2; i < 12; i = i + 1){
            sum = sum + data2[i];
          }
            sum = -(~sum);
            checksum[1] = (char)(((short)sum/100)%10) + '0';
            checksum[2] = (char)((((short)sum/10)%100)%10) + '0';
            checksum[3] = (char)((((short)sum%1000)%100)%10) + '0';
            checksum[4] =  10;
              for(i = 1; i < 5; i = i + 1){
              data2[11+i] = checksum[i];
              }

              for(i = 0; i <= 15; i = i + 1){
              SendCharUart1(data2[i]);
              }
  }
}
```

รูป ก6 โปรแกรมรวบรวมค่าต้นแพ็คเกตและส่ง

```
void USART1_IRQHandler(void){
  if(USART_GetITStatus(USART1, USART_IT_RXNE) != RESET){
    //sendDataFlg = 1;
    printed = 0;
    RxMessage = USART_ReceiveData(USART1);

    RxDataBuff = RxMessage;  //move data to data buffer Rx
    RX[countRx] = RxDataBuff;
    countRx = countRx + 1;

    if(countRx >= 2){
      countRx = 0;
      if(RX[0] == 'E' && RX[1] == '0' && RX[2] == '2'){
        checkE01 = 1;
      }
      else{
        checkE01 = 0;
      }
    }

    RxUpdate = 1;
    USART_ClearITPendingBit(USART1, USART_IT_RXNE);

  }
  else if(USART_GetITStatus(USART1, USART_IT_TXE) != RESET){
    printed = 1;
    USART_SendData(USART1, TxBuffer[TxCounter++]);
    USART_ClearITPendingBit(USART1, USART_IT_TXE);
    if(TxBuffer[TxCounter] == 0 || TxCounter > TxBufferSize){
      USART_ITConfig(USART1, USART_IT_TXE, DISABLE);
      TxReady = 1;
    }
  }
}
```

รูป ก7 โปรแกรมในส่วนของอินเตอร์รัพท์

```
  LatStation[0] = 1644.5511;        LongStation[0] = 10011.5784;
  LatStation[1] = 1644.5801;        LongStation[1] = 10011.6272;
  LatStation[2] = 1644.5551;        LongStation[2] = 10011.7176;


////////////////////////////////////////////////////////////////////

while(1){
        CheckRandom = 1;
        GPS();
        PackAndSend();
        Distance();
        CheckDistance();
        DelaymS(700);
        while(CheckRandom){
          GPIO_SetBits(GPIOB, GPIO_Pin_9);
          GPIO_ResetBits(GPIOB, GPIO_Pin_9);
          GPIO_SetBits(GPIOB, GPIO_Pin_13);
          if (checkE01 == 1 && CheckRandom == 1){
            GPIO_SetBits(GPIOB, GPIO_Pin_10);
            CheckRandom = 0;
            countRx = 0;
            checkE01 = 0;
            RxDataBuff = 0;
            n = 0;
            randVal = 0;
            RX[0] = 0;
            RX[1] = 0;
            RX[2] = 0;
          }
          else if(checkE01 == 0 && CheckRandom == 1){
            GPIO_SetBits(GPIOB, GPIO_Pin_11);
            countRx = 0;
            RxDataBuff = 0;
            n = n + 1;
            RandomDelay();
            RX[0] = 0;
            RX[1] = 0;
            RX[2] = 0;
            if(n == 5){ n = 0; CheckRandom = 0;}
          }
          GPIO_ResetBits(GPIOB, GPIO_Pin_10);
        }

        GPIO_ResetBits(GPIOB, GPIO_Pin_11);
        GPIO_ResetBits(GPIOB, GPIO_Pin_13);
        hByteLat = 0;
        lByteLat = 0;
        hByteLong = 0;
        lByteLong = 0;
        LatInt = 0;
        LongInt = 0;

  }
}
```

รูป ก8 โปรแกรมในส่วนของอินเตอร์รัพท์

ตัวอย่างโค้ดฝั่งแม่ข่าย

```
void USART1_IRQHandler(void){
  if(USART_GetITStatus(USART1, USART_IT_RXNE) == SET){

    RxReady = 1;
    RxMessage = USART_ReceiveData(USART1);
    RxDataBuff = RxMessage;  //move data to data buffer Rx

    if(RxReady == 1){
      GPIO_SetBits(GPIOB, GPIO_Pin_11);
      RxReady = 0;
      SysTick_CounterCmd(SysTick_Counter_Disable);
      RxData[nData] = RxDataBuff;    //Pack data to array packageb
      nData = nData + 1;
    }

    if(nData > 16){
      nData = 0;
    }
    else if(nData == 16){
      if(RxData[0] == 'S' && RxData[1] == thisStation && RxData[2] == 'E' && RxData[5] == ':' &&
         RxData[8] == ':' && RxData[11] == ':' && RxData[15] == 10){
           CarLabel1 = ((short)RxData[2] + (short)RxData[3] + (short)RxData[4]) - 166;
           StatusCar[CarLabel1] = RxData[7];
           for(int j = 2; j <= 4; j = j+1){
             SendCharUart1(RxData[j]);        //
           }
         }
      else{
        if(RxData[1] == thisStation){
          nData = 0;
        }
      }
    }
```

รูป ก9 โปรแกรมในส่วนของอินเตอร์รัพท์