สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

เครื่องบันทึกการเดินทาง
GPS DATA LOGGER

โดย
นายวัฒณสิทธิ์ พิมเพา

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เลขหมู่.....
เลขทะเบียน 61876
วัน,เดือน,ปี 24 ก.ค. 2549

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

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิชาอิเล็กทรอนิกส์
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2547

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เครื่องบันทึกการเดินทาง
GPS DATA LOGGER

โดย

นายวัฒณสิทธิ์ พิมเพา 45015251

อาจารย์ที่ปรึกษา

รศ.ดร. มนัส สังวรศิลป์

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาอิเล็กทรอนิกส์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2547

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รายงาน ปีการศึกษา 2547

ภาควิชา อิเล็กทรอนิกส์

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง เครื่องบันทึกการเดินทาง

ผู้จัดทำ

1. นายวัฒณสิทธิ์ พิมเพา

.....อาจารย์ที่ปรึกษา

(รศ.ดร.มนัส สังวรศิลป์)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เครื่องบันทึกการเดินทาง

นายวัฒณสิทธิ์ พิมเพา

รศ.ดร.มนัส สังวรศิลป์

ปีการศึกษา 2547

บทคัดย่อ

เครื่องบันทึกการเดินทางเป็นการประยุกต์ใช้ระบบจีพีเอสมาใช้ในงานคมนาคม โดยเครื่องบันทึกการเดินทางจะบันทึกตำแหน่งพิกัดของรถยนต์ในเวลาต่างๆ และยังสามารถเตือนผู้ขับขี่เกี่ยวกับความเร็วที่เกินกำหนด ซึ่งทำให้สินเปลืองพลังงาน ข้อมูลพิกัดดังกล่าวสามารถที่จะนำมาวิเคราะห์ด้วยซอฟท์แวร์คอมพิวเตอร์ เกี่ยวกับเส้นทางที่ผ่านมา ความเร็ว และเหตุการณ์ต่างๆ เพื่อให้ผู้ขับขี่สามารถศึกษาและคำนวณเส้นการเดินทาง หรือตรวจตราคนขับรถในการปฏิบัติหน้าที่เพื่อการบริหารการคมนาคม ขนส่งที่มีประสิทธิภาพ

เครื่องบันทึกการเดินทางที่จัดทำขึ้นสามารถที่จะบันทึกการเดินทางของรถยนต์เกี่ยวกับเส้นรุ้ง เส้นแวง วันเวลาที่เดินทาง และสามารถต่อกับคอมพิวเตอร์เพื่อตั้งค่าต่างๆ เช่น ตั้งวันที่ เวลา หรือเป็นการนำข้อมูลการเดินทางที่เก็บไว้มาแสดงผลบนหน้าจอคอมพิวเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

GPS DATA LOGGER

Wattansit Pimpao

Assoc. Prof. Dr. Manas Sangworasil

2004

Abstract

GPS DATA LOGGER is GPS application for logistic. It can save position of a vehicle and warn about speed to save the energy. The positions can be analyzed by a computer when and where the vehicle you gone. It provides to manage the best way logistic.

GPS DATA LOGGER can record latitude, longitude, date and time. It can connect to computer by serial port in order to setting parameter such as date time and download data for analysis.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

หน้า

บทคัดย่อ

Abstract

สารบัญ

สารบัญรูปภาพ

สารบัญตาราง

บทที่ 1 บทนำ 1

บทที่ 2 ทฤษฎีและหลักการ 2

2.1 จีพีเอส (GPS Global Positioning System) 2

2.2 ประวัติและพัฒนาการระบบจีพีเอส 2

2.3 ประวัติและพัฒนาการของดาวเทียมจีพีเอส 3

2.4 ส่วนประกอบของระบบ GPS 4

2.5 ส่วนประกอบของเครื่องรับสัญญาณดาวเทียมจีพีเอส 7

2.6 การประยุกต์ใช้งาน 9

2.7 ไมโครคอนโทรลเลอร์ตระกูล MCS-51 10

2.8 การทำงานของ 8051 11

2.9 โครงสร้างหน่วยความจำ 12

2.10 หน่วยความจำใช้งานทั่วไป 12

2.11 รีจิสเตอร์ฟังก์ชันพิเศษ (Speacial Function Register) 13

2.12 Program Status World 13

2.13 รีจิสเตอร์ B (B Register) 14

2.14 ตัวชี้สแตค (Stack Pointer) 14

2.15 รีจิสเตอร์ Data Pointer (DPTR) 15

2.16 รีจิสเตอร์พอร์ต (Port Registers) 15

2.17 รีจิสเตอร์เวลา (Timer Registers) 15

2.18 รีจิสเตอร์พอร์ตอนุกรม (Serial Port Registers) 15

2.19 รีจิสเตอร์อินเทอร์รัพท์ (Interrupt Port Registers) 16

2.20 Power Control Register (PCON) 16

2.21 หน่วยความจำภายนอก (External Memory) 16

2.22 การติดต่อกับหน่วยความจำโปรแกรมภายนอก 16

2.23 การติดต่อกับหน่วยความจำข้อมูลภายนอก 17

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.24 Reset Operation 17
2.25 TIMER 17
2.26 โมดูลรับสัญญาณ GPS และมาตรฐาน NMEA 18
2.27 NMEA Message 18
บทที่ 3 การออกแบบและการสร้าง **25**
3.1 การออกแบบวงจร 25
3.2 วงจรแหล่งจ่ายไฟ 25
3.3 วงจรรีเซตไมโครคอนโทลเลอร์(EconoReset) 26
3.4 วงจรไมโครคอนโทลเลอร์ 27
3.5 วงจรนาฬิกา(Real Time Clock) 28
3.6 วงจรหน่วยความจำภายนอก (EEPROM) 28
3.7 วงจรเชื่อมต่อกับคอมพิวเตอร์(Computer Interface) 29
3.8 วงจรสลับการเชื่อมต่อกับ (Serial Exchange) 29
3.9 วงจรหลอดแสดงผล(LED Display) 30
3.10 วงจรย่อยต่างๆ 30
3.11 คอนเนคเตอร์(Connector) 31
3.12 การออกลายทองแดง 32
3.13 การประกอบวงจร 35
3.14 ทดสอบวงจรกับบอร์ดทดลอง 36
3.15 ทดสอบวงจรกับแผ่นปริ้นส์เอนกประสงค์ 36
3.16 ทดสอบวงจรกับแผ่นปริ้นส์ลายทองแดง 37
3.17 การประกอบวงจรลงกล่อง 39
บทที่ 4 การทดลองและผลการทดลอง **42**
4.1 การทดลองและผลการทดลองของการโปรแกรมไมโครคอนโทลเลอร์ 42
4.2 การวิเคราะห์สัญญาณจีพีเอส 46
4.3 การทดลองและผลการทดลองต่อจีพีเอสกับคอมพิวเตอร์ด้วยซอฟท์แวร์ทั่วไป 53
4.4 การทดลองและผลการทดลองระบบคำสั่งของเครื่องบันทึกการเดินทาง 56
4.5 การทดสอบฮาร์ดแวร์ 61
4.6 การทดลองและผลการทดลองของเครื่องบันทึกการเดินทางกับซอฟท์แวร์คอมพิวเตอร์61
4.7 การทดลองและผลการทดลองของเครื่องบันทึกการเดินทางกับซอฟท์แวร์คอมพิวเตอร์ที่มีขายตามท้องตลาด 64
บทที่ 5 สรุปและวิจารณ์ **65**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก

กิตติกรรมประกาศ

บรรณานุกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูปภาพ

	หน้า
รูปที่ 2.4.1 แสดงการโคจรของดาวเทียม GPS รอบโลก	5
รูปที่ 2.4.2 แสดงสถานีควบคุมระบบดาวเทียม GPS 5 แห่ง	6
รูปที่ 2.5.1 แสดงส่วนประกอบของระบบดาวเทียม GPS	7
รูปที่ 2.6.1 หลักการทำงานของจีพีเอส	8
รูปที่ 2.6.2 การใช้ดาวเทียม 3 ดวงขึ้นไปในการหาพิกัด	8
รูปที่ 3.1 ซอฟท์แวร์โพรเทล99	25
รูปที่ 3.2 วงจรแหล่งจ่ายไฟ	25
รูปที่ 3.3 วงจรรีเซตไมโครคอนโทลเลอร์	26
รูปที่ 3.4 วงจรไมโครคอนโทลเลอร์	27
รูปที่ 3.5 วงจรนาฬิกา	28
รูปที่ 3.6 วงจรหน่วยความจำภายนอก	28
รูปที่ 3.7 วงจรเชื่อมต่อกับคอมพิวเตอร์	29
รูปที่ 3.8 วงจรสลับการเชื่อมต่อ	29
รูปที่ 3.9 วงจรหลอดแสดงผล	30
รูปที่ 3.10 วงจรคริสตัล	30
รูปที่ 3.11 วงจรป้อนกับสัญญาณรบกวน	31
รูปที่ 3.11.1 จีพีเอสคอนเนคเตอร์	31
รูปที่ 3.11.2 DB9 คอนเนคเตอร์	31
รูปที่ 3.11.3 คอนเนคเตอร์ลำโพงบัซเซอร์	32
รูปที่ 3.12.1 เอกสารวงจรที่สมบรูณ์	32
รูปที่ 3.12.2 อุปกรณ์ในเอกสารลายทองแดง	33
รูปที่ 3.12.3 ขณะสร้างลายทองแดง	33
รูปที่ 3.12.4 ลายทองแดงสมบูรณ์	34
รูปที่ 3.12.5 ลายทองแดงและอุปกรณ์แบบ 3 มิติ	34
รูปที่ 3.13.1อุปกรณ์อิเล็กทรอนิกส์	35
รูปที่ 3.13.2 อุปกรณ์รับสัญญาณจีพีเอส	35
รูปที่ 3.13.3 เครื่องมือ	35
รูปที่ 3.14.1 การต่อวงจรกับบอร์ดทดลอง	36
รูปที่ 3.15.1 การต่อวงจรกับแผ่นปริ้นส์เอนกประสงค์ด้านหน้า	36
รูปที่ 3.15.2 การต่อวงจรกับแผ่นปริ้นส์เอนกประสงค์ด้านหลัง	37

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 3.16.1 แผ่นคัดลอกลายทองแดงจากเอกสารลายทองแดง 37
รูปที่ 3.16.2 การต่ออุปกรณ์ลงแผ่นลายทองแดงด้านหน้า 38
รูปที่ 3.16.3 การต่ออุปกรณ์ลงแผ่นลายทองแดงด้านหลัง 38
รูปที่ 3.17.1 กล่องที่เตรียมไว้ 39
รูปที่ 3.17.2 การเจาะรูที่เหมาะสม 39
รูปที่ 3.17.3 สายอากาศจีพีเอส 39
รูปที่ 3.17.4 สายเชื่อมต่อกับคอมพิวเตอร์ 40
รูปที่ 3.17.5 หลอดแสดงสถานะ 40
รูปที่ 3.17.6 ลำโพงบัซเซอร์ 40
รูปที่ 3.17.7 ปลั๊กไฟในรถยนต์ 41
รูปที่ 3.17.8 การเชื่อมต่ออุปกรณ์ภายนอกเข้ากับวงจร 41
รูปที่ 3.17.9 การประกอบอุปกรณ์ลงกล่อง 41
รูปที่ 4.1.1 การต่ออุปกรณ์เชื่อมต่อกับคอมพิวเตอร์ 42
รูปที่ 4.1.2 สาย RS-232 ที่ทีสวิตซ์ต่อขา PSEN เพื่อเพิ่มความสะดวก 42
รูปที่ 4.1.3 ซอฟท์แวร์แฟลชเมจิก 43
รูปที่ 4.1.4 การลบข้อมูลเดิม 44
รูปที่ 4.1.5 การโปรแกรมข้อมูลใหม่เข้าไป 45
รูปที่ 4.1.6 การสิ้นสุดการโปรแกรม 46
รูปที่ 4.2.1 การเลือกที่มาของข้อมูล 47
รูปที่ 4.2.2 ไม่สามารสหาข้อมูลพิกัดได้ 47
รูปที่ 4.2.3 จำนวนดาวเทียมที่หาได้มี 2 ดวง 48
รูปที่ 4.2.4 เปิดรับสัญญาณได้แค่ 2 ช่องสัญญาณ 48
รูปที่ 4.2.5 ข้อมูลไม่พอที่จะหาพิกัดได้ 49
รูปที่ 4.2.6 ข้อมูลพิกัด 49
รูปที่ 4.2.7 จำนวนดาวเทียมที่รับได้ 5 ดวง 50
รูปที่ 4.2.8 จำนวนช่องที่เปิดใช้งานมากขึ้น 50
รูปที่ 4.2.9 การนำข้อมูลที่เพียงพอมาคำนวณหาพิกัด 51
รูปที่ 4.2.10 คำสั่งการตั้งค่า 51
รูปที่ 4.2.11 การตั้งค่าให้รับเฉพาะ GRMC 52
รูปที่ 4.2.12 ข้อมูล GRMC 52
รูปที่ 4.3.1 ซอฟท์แวร์ไฮเปอร์เทอร์มินอล 53
รูปที่ 4.3.2 การตั้งค่าซอฟท์แวร์ไฮเปอร์เทอร์มินอล 54

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 4.3.3 ข้อมูลที่ปรากฏ 54
รูปที่ 4.3.4 การตั้งค่าซอฟท์แวร์เทอร์มินอล 55
รูปที่ 4.3.5 การใช้ซอฟท์แวร์เทอร์มินอลรับค่าจากจีพีเอส 55
รูปที่ 4.4.1 การเข้าสู่โหมดคำสั่ง 56
รูปที่ 4.4.2 คำสั่งหมายเลขเครื่อง 56
รูปที่ 4.4.3 คำสั่งเสียง 57
รูปที่ 4.4.4 คำสั่งวัน 57
รูปที่ 4.4.5 คำสั่งเวลา 58
รูปที่ 4.4.6 คำสั่งตั้งเวลาเก็บข้อมูล 58
รูปที่ 4.4.7 คำสั่งสถานะหน่วยความจำ 59
รูปที่ 4.4.8 การใช้คำสั่งโหลดข้อมูลในโหมดคำสั่ง 59
รูปที่ 4.4.9 ข้อมูลที่โหลดมาได้ 59
รูปที่ 4.4.10 การใช้คำสั่งลบข้อมูลในโหมดคำสั่ง 60
รูปที่ 4.4.11 ข้อมูลขณะทำการลบ 60
รูปที่ 4.4.12 เมื่อทำการลบเสร็จสิ้น 60
รูปที่ 4.5.1 ข้อความต้องรับที่ส่งไปยัง RS-232 ขณะเปิดเครื่อง 61
รูปที่ 4.6.1 แสดงเส้นทางการเดินของรถระยะครึ่งทาง 62
รูปที่ 4.6.2 แสดงเส้นทางการเดินของรถที่ปลายทาง 62
รูปที่ 4.7.1 แสดงตัวอย่างซอฟท์แวร์แผนที่ที่มีขายตามท้องตลาด 63
รูปที่ 4.7.2 แสดงเส้นทางการเดินของรถ 64

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

หน้า

ตารางที่ 2.13.1 แสดงบิตและหน้าที่ต่างๆใน PSW 13
ตารางที่ 2.28.1เรคอร์ดหลักๆใน NMEA message ที่มักถูกนำมาใช้งาน 18
ตารางที่ 2.28.2 ความหมายของข้อมูลที่อยู่ในเรคอร์ด GGA 19
ตารางที่ 2.28.3 ความหมายของข้อมูลที่อยู่ในเรคอร์ด GLL 20
ตารางที่ 2.28.4 ความหมายของข้อมูลที่อยู่ในเรคอร์ด GSA 21
ตารางที่ 2.28.5 ความหมายของข้อมูลที่อยู่ในเรคอร์ด GSV 22
ตารางที่ 2.28.6 ความหมายของข้อมูลที่อยู่ในเรคอร์ด RMC 23
ตารางที่ 2.28.7 ความหมายของข้อมูลที่อยู่ในเรคอร์ด VTG 24
ตารางที่ 2.28.8 ตารางสรุปคุณสมบัติของ 6 เรคอร์ดหลักใน NMEA message 24

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

1.1 ความเป็นมาของปัญหา

ในปัจจุบันในการเดินทางไปยังสถานที่ต่างๆ ในเรื่องของการคมนาคม การสัญจรของผู้ขับขี่รถยนต์ไปยังที่ต่างๆ บางครั้งผู้ขับขี่ไม่ทราบเส้นทางหรือในกรณีที่ผู้ขับขี่ไม่ชำนาญเส้นทางที่ต้องการไปอาจทำให้หลงทาง เกิดความลังเล ทำให้ขับขี่ช้าลงและเสียเวลาในการเดินทาง ก่อให้เกิดปัญหาการจราจร โดยเฉพาะในกรุงเทพมหานคร เป็นสาเหตุให้การดำเนินธุรกิจล่าช้า ความเสียหายย่อมเกิดกับหน่วยงานหรือประเทศชาติ ไม่ว่าจะเป็นการสูญเสียน้ำมัน ค่าใช้จ่ายในการขนส่ง ฯลฯ ส่วนหนึ่งในด้านการเดินทางช่วยให้ผู้ใช้มีความมั่นใจในการเดินทาง โดยเฉพาะเส้นทางที่จะไป รวมถึงบันทึกเส้นทางที่ผ่านมาได้ จึงได้มีการนำเทคโนโลยีการนำร่องด้วยดาวเทียมจีพีเอส มาประยุกต์ใช้ในการบอกพิกัดตำแหน่งมาแสดงผลในแผนที่ของเครื่องคอมพิวเตอร์ ระบบสามารถที่จะทำการวิเคราะห์เส้นทางที่จะไปยังปลายทางเส้นทาง เพื่อให้ผู้ใช้สามารถประหยัดค่าใช้จ่ายในการเดินทาง ตรวจสอบการใช้งานยานพาหนะให้อยู่ในขอบเขตภารกิจที่กำหนด ให้ข้อมูลการเดินทาง แสดงเส้นทางต่างๆบนแผนที่ แสดงข้อมูลตำแหน่งของรถยนต์ซึ่งเป็นการเดินทางที่ผ่านมา แสดงกราฟความเร็วของยานพาหนะ ตั้งแต่จุดเริ่มต้นจนเสร็จสิ้นภารกิจ ผู้ใช้สามารถนำข้อมูลมาบริหาร วิเคราะห์ ตรวจสอบ หรืออ้างอิง สามารถนำเสนอข้อมูลได้หลายรูปหรือจัดพิมพ์รายงาน โดยข้อมูลต่างๆจะถูกเก็บไว้ในคอมพิวเตอร์และสามารถค้นหาข้อมูลได้ตามต้องการ

1.2 วัตถุประสงค์ของการจัดทำโครงงาน

1. เพื่อให้มี ความรู้ ความเข้าใจ การประยุกต์ใช้งาน ระบบบอกตำแหน่งด้วยดาวเทียม
2. ระบบฮาร์ดแวร์ (Hardware) สามารถจัดเก็บตำแหน่ง รายละเอียดของการเดินทางได้
3. ระบบซอฟท์แวร์ (Software) สามารถรายงานรายละเอียดเส้นทางการเดินทางได้ เพื่อ
4. บริหารจัดการการเดินทางและการปฏิบัติงานของพนักงานขับรถให้เหมาะสม

1.3 ขอบเขตของโครงงาน

1 สามารถทำงานได้ในเขตพื้นที่กรุงเทพมหานคร

2 สามารถใช้งานร่วมกับ คอมพิวเตอร์ ผ่านพอร์ตอนุกรม(Serial Port)

3 สามารถบันทึกเส้นทางและวันเวลาที่เดินทาง

4 สามารถแสดงเส้นทางและรายละเอียดทั้งหมดที่สามารถเดินทางได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2
ทฤษฎีและหลักการ

2.1 จีพีเอส (GPS Global Positioning System)

จีพีเอสมีรากศัพท์มาจาก Global Positioning System แปลว่าเครื่องบอกพิกัดรอบโลก จีพีเอส เป็นระบบหาพิกัดบนพื้นโลกโดยการอ้างอิงจากดาวเทียมที่มีความแม่นยำสูง ใช้หาพิกัดใดๆ บนพื้นโลกได้ ทุกเวลา ทุกสภาพอากาศ มีการนำจีพีเอสไปใช้ในกิจกรรมต่างๆ เช่น การทหาร ระบบการเดินเรือ การสำรวจแผนที่ ระบบติดตามตำแหน่งบนพื้นผิวโลกช่วยวางแผนเดินทางด้วยแผนที่ในรถยนต์ เป็นต้น

2.2 ประวัติและพัฒนาการระบบจีพีเอส

ตั้งแต่ในอดีตมนุษย์เราก็มีความพยายามที่จะสร้างเครื่องไม้เครื่องมือเพื่อบอกให้ได้ว่าเรากำลังอยู่ที่ใด เพื่อป้องกันการหลงทางและสามารถกลับไปยังจุดเดิมได้อย่างถูกต้อง ซึ่งในการเดินเรือสมัยแรก ๆ ก็มีการใช้ดวงดาวเป็นการบอกตำแหน่งและทิศทาง ต่อมาเมื่อเทคโนโลยีทันสมัยมากขึ้น ก็ได้มีการคิดค้นประดิษฐ์เข็มทิศและเครื่องวัดระยะทางหาเส้นรุ้งและเส้นแวง (Sextant) ขึ้นมา โดยเข็มทิศจะชี้ไปทางเหนือเสมอ ฉะนั้นไม่ว่าเราจะไม่รู้ตำแหน่งของเราแต่เราจะยังสามารถรู้ทิศทางที่กำลังเดินทางไปได้ ส่วนเครื่องวัดระยะทางหาเส้นรุ้งและเส้นแวง นั้นจะช่วยในการวัดมุมระหว่างดวงดาวกับพื้นดิน ในยุคแรกๆนั้นเครื่องมือนี้จะใช้ในการเดินเรือและสามารถบอกได้แต่เส้นรุ้งเท่านั้น ไม่สามารถบอกเส้นแวงได้

ต่อมาในศตวรรษที่ 17 ประเทศอังกฤษก็ได้ตั้งกลุ่มนักวิทยาศาสตร์เพื่อทำการสร้างเครื่องมือเพื่อหาเส้นแวงให้ได้ ซึ่งกลุ่มที่ตั้งขึ้นมาถูกเรียกว่า แบนด์ออฟลองกิจูด (Board of Longitude) โดยมีรางวัลให้กับผู้ที่สามารถสร้างเครื่องมือที่ใช้หาเส้นแวงได้ ซึ่งในปี ค.ศ.1761 จอห์น แฮร์ริสัน (John Harrison) ได้พัฒนาเครื่องมือที่สามารถใช้หาเส้นแวงได้ซึ่งเรียกว่า Chronometer ซึ่งต่อมาก็มีการใช้เครื่องหาเส้นรุ้งและเครื่องหาเส้นแวงร่วมกันในการเดินทางอย่างแพร่หลาย

ในช่วงต้นศตวรรษที่ 20 ได้มีการพัฒนาระบบการส่งสัญญาณวิทยุมาใช้งานกันมากขึ้นจนกระทั่งได้มีการนำมาใช้กันอย่างแพร่หลายในสมัยสงครามโลกครั้งที่ 2 โดยทั้งเรือ และเครื่องบินจะใช้ระบบการรับ-ส่งสัญญาณวิทยุจากสถานีภาคพื้นดินเป็นตัวนำทาง การส่งสัญญาณวิทยุนั้นจะสามารถส่งได้ทั้งแบบความถี่สูงและความถี่ต่ำ แต่ข้อเสียก็คือหากส่งสัญญาณในช่วงความถี่สูงจะสามารถรับ-ส่งข้อมูลได้อย่างถูกต้องแต่ครอบคลุมได้เพียงพื้นที่จำกัด ส่วนการรับ-ส่งสัญญาณในช่วงความถี่ต่ำสามารถครอบคลุมพื้นที่ได้กว้างไกลกว่าแต่ความถูกต้องต่ำกว่า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในศตวรรษที่ 20 ดาวเทียมสปุตนิก (Sputnik) ของประเทศรัสเซียได้ถูกส่งออกสู่อวกาศเมื่อวันที่ 4 ตุลาคม ค.ศ. 1957 และทำให้เราเริ่มตระหนักกันว่าเราสามารถใช้ดาวเทียมในการนำทางได้เช่นเดียวกับดวงดาวบนท้องฟ้า โดยนักวิจัยจากสถาบัน MIT ได้ติดตามวิถีการโคจรของดาวเทียมสปุตนิกและได้สังเกตเห็นว่าสัญญาณวิทยุจากดาวเทียมสปุตนิกจะสูงขึ้นเมื่อดาวเทียมโคจรเข้ามาใกล้และต่ำลงเมื่อดาวเทียมโคจรห่างออกไป จากข้อเท็จจริงดังกล่าวที่ว่าเราสามารถจะติดตามตำแหน่งของดาวเทียมในขณะโคจรรอบโลกได้จากภาคพื้นดินนั้น จึงเป็นที่มาของสมมุติฐานที่ว่า ในทางกลับกัน เราก็น่าจะสามารถติดตามหรือระบุตำแหน่งของวัตถุใดๆ บนพื้นโลกโดยการใช้สัญญาณวิทยุจากดาวเทียมได้เช่นกัน

ต่อมาทางประเทศสหรัฐอเมริกาก็ได้มีการพัฒนาดาวเทียมนำร่องออกสู่อวกาศเช่นกัน โดยทางอเมริกาเรียกระบบนี้ว่า ทรานสิท (Transit) ซึ่งประกอบไปด้วยดาวเทียม 6 ดวงโคจรรอบโลกผ่านขั้วโลก ที่ความสูงประมาณ 1,100 กิโลเมตร โดยใช้สำหรับหาตำแหน่งของเรือเดินสมุทร และเครื่องบิน โดยระบบนี้รัฐบาลอเมริกาอนุญาตให้เอกชนบางรายใช้ในงานสำรวจเท่านั้น โดยยังไม่เปิดให้บุคคลทั่วไปใช้งาน แต่ระบบนี้ก็ใช้งานกันได้ไม่นานนักเนื่องจากการส่งสัญญาณช้าและมีความถูกต้องต่ำ จึงได้เริ่มมีการพัฒนาระบบจีพีเอสเพื่อให้มีการบอกตำแหน่งได้อย่างแม่นยำมากขึ้น โดยได้เริ่มมีการพัฒนาอย่างต่อเนื่องจากหน่วยงานที่เกี่ยวข้องส่งผลทำให้ระบบจีพีเอสที่สมบูรณ์ได้ถูกใช้งานเต็มรูปแบบจากดาวเทียม 24 ดวงในกลางปี 1990

2.3 ประวัติและพัฒนาการของดาวเทียมจีพีเอส

ในศตวรรษที่ 20 ในการพัฒนาเครื่องส่งวิทยุทำให้เครื่องช่วยการเดินทางได้พัฒนาไปอีกขั้นเรียกว่า Radio beacons รวมทั้ง Loran และ Omega ในที่สุดเทคโนโลยีของดาวเทียมทำให้เครื่องช่วยการเดินทางและการหาตำแหน่งจะพิจารณาจากเส้นที่สัญญาณเดินทางผ่านด้วยการวัดของ Doppler ที่เคลื่อนที่ไป

ระบบทรานสิท เป็นระบบเครื่องช่วยการเดินเรือโดยอาศัยดาวเทียม ได้รับการคิดค้นสำเร็จในปี ค.ศ.1950 และใช้งานอยู่ 33ปี จึงได้ปลดประจำการไป ระบบทรานสิทได้พัฒนามาให้ข้อมูลการหาตำแหน่งที่แน่นอนให้กับเรือดำน้ำที่มีจรวดนำวิถี หลักการคือ การคาดการณ์โดยใช้ความถี่ Doppler ที่เปลี่ยนแปลงตำแหน่งไปจากดาวเทียมสปุตนิก สัญญาณเปลี่ยนของ Doppler สามารถพิจารณาการโคจรของดาวเทียมใช้ข้อมูลที่จดเอาไว้ที่สถานีหนึ่งเมื่อดาวเทียมโคจรผ่านไป ระบบทรานสิท ประกอบด้วย ดาวเทียม 6 ดวงที่เกือบเป็นวงกลม การโคจรผ่านขั้วโลกที่ความสูง 1,075 กิโลเมตร ระยะเวลาของการหมุน 107 นาที การโคจรของดาวเทียมทรานสิทจะแน่นอนกว่าโดยการติดตามจากสถานีพื้นโลกที่กำหนดไว้ ด้วยสภาพที่น่าพอใจความเร็วที่แน่นอนเป็น 35 ถึง 100 เมตรรอบต่อนาที ปัญหาของระบบทรานสิท คือการครอบคลุมพื้นที่มีช่องว่างระหว่างกันมาก ผู้ใช้ต้องคำนวณโดยการ interpolate ตำแหน่งของตนเองระหว่างที่ดาวเทียมโคจรผ่านไป ความสำเร็จของ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระบบทรานสิทเป็นการกระตุ้นให้ทั้งกองทัพเรือและกองทัพอากาศของสหรัฐฯ พิจารณาระบบช่วยการเดินทางที่ก้าวหน้ากว่าเดิมและมีประสิทธิภาพยิ่งขึ้น ทางกระทรวงกลาโหมของสหรัฐฯ ได้ผลิตระบบการหาตำแหน่งทั่วโลกแนฟสตาร์ (Navstar ย่อมาจาก The Navstar global Positioning system - GPS) ซึ่งจะเอาไว้ในการระบุตำแหน่งการนำวิถีของจรวดทั้งทางบกและทางอากาศและยังสามารถบอกได้ว่ากองกำลังทหารอยู่ ณ. ที่ใดของสนามรบและนั่นก็เป็นจุดเริ่มต้นของการผลิตคิดค้นระบบวิธีการระบุตำแหน่งบนพื้นโลก ซึ่งระบบจีพีเอสจะขัดแย้งกับระบบทรานสิท คือระบบจีพีเอสให้สัญญาณครอบคลุมพื้นที่ต่อเนื่องและให้ความถูกต้องและแม่นยำกว่าระบบเดิม ซึ่งได้ผลิตให้ดาวเทียมมีความทันสมัยและเหมาะสมในการนำไปใช้งานต่างๆจนถึงปัจจุบันดาวเทียม จีพีเอสได้ถูกสร้างขึ้นมาแล้วถึง 4 รุ่น คือ Block I, Block II/IIA, Block IIR , Block IIF

2.4 ส่วนประกอบของระบบ GPS

1 ส่วนอวกาศ (Space Segment) ประกอบไปด้วยดาวเทียมจำนวน 24 ดวง ซึ่งบินโคจรรอบโลก ดาวเทียมนี้ผลิตโดยบริษัท รอคเวลล์อินเตอร์เนชั่นแนล และถูกปล่อยสู่อวกาศจากแหลมฟลอริด้า ประเทศสหรัฐอเมริกา ขนาดของดาวเทียมจะประมาณเท่ากับขนาดรถยนต์ และมีน้ำหนักประมาณ 19,000 ปอนด์ วงโคจรของดาวเทียมจะอยู่สูงจากพื้นโลกประมาณ 12,660 ไมล์ ทำมุมกับพื้นโลก 55 องศา มีวงโคจรทั้งหมด 6 เส้นทาง ในแต่ละเส้นทางจะมีดาวเทียมโคจรอยู่ 4 ดวง โดยดาวเทียมหนึ่งดวงจะสามารถโคจรรอบโลกได้ 1 รอบใน 12 ชั่วโมง (ประมาณ 1.8 ไมล์ต่อวินาที) ในระหว่างการโคจรรอบโลกนั้นดาวเทียมจะมีการส่งสัญญาณสู่พื้นโลกผ่านเสาส่งสัญญาณที่ติดตั้งจากดาวเทียมมายังโลก และมีการนำพลังงานแสงอาทิตย์มาใช้ในการขับเคลื่อน โครงสร้างของวงโคจรในลักษณะนี้ทำให้มีดาวเทียมจำนวน 5-8 ดวง ที่เครื่องรับ จีพีเอสสามารถรับสัญญาณได้ ณ. ตำแหน่งหนึ่งตำแหน่งใดได้ตลอดเวลาและดาวเทียมจีพีเอส จะมีปีกเป็นแผงเซลพลังงานแสงอาทิตย์ (solar cell panels) โดยปกติจะพยายามหมุนตัวให้สามารถรับพลังงานแสงอาทิตย์ได้มากที่สุด ดังนั้นตัวดาวเทียมจะมีการหมุนปรับตัวตลอดเวลาโดยให้ปีกเซลพลังงานแสงอาทิตย์ตั้งฉากกับตำแหน่งของดวงอาทิตย์ในตัวดาวเทียมยังบรรจุแบตเตอรี่สำหรับให้พลังงานเมื่อดาวเทียมจีพีเอสเคลื่อนตัวอยู่ภายในเงาของโลก ตำแหน่งของดาวเทียมตลอดเวลาจะถูกคำนวณให้เครื่องรับหาตำแหน่งของผู้ใช้ที่สามารถรับข้อมูลได้ 50 bps ต่อเนื่องกัน วงโคจรของแต่ละดวงต่อระยะเวลา 1 ชั่วโมง โดยการตั้ง element การโคจรที่ 15 keplerian พร้อมทั้งค่าสัมประสิทธิฮาร์โมนิคเพิ่มขึ้นจากการรบกวนและแก้ไขทุกๆ 4 ชั่วโมง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 2.4.1 แสดงการโคจรของดาวเทียม GPS รอบโลก

2 ส่วนควบคุม (Control Segment) ส่วนควบคุมจะประกอบไปด้วยสถานีซึ่งคอยตรวจสอบดูแลการทำงานของดาวเทียมโดยใช้เรดาร์ส่งสัญญาณไปยังดาวเทียม เพื่อให้ดาวเทียมอยู่ในวงโคจรในความสูง ความเร็ว และตำแหน่งที่ถูกต้อง และในทางกลับกัน สถานทีเหล่านี้ยังทำหน้าที่รับสัญญาณจากดาวเทียมและส่งข้อมูลไปยังเครื่องลูกข่ายจีพีเอส เพื่อบอกตำแหน่งและข้อมูลของเครื่องลูกข่ายนั้น ๆ อย่างถูกต้องด้วย สถานีที่ทำการควบคุมดาวเทียมจะมีอยู่ 5 แห่ง คือ สถานีหลักที่ Colorado สถานีบนเกาะ Ascension, สถานี Diego Garcia (มหาสมุทรอินเดีย), Kwajalein และ Hawaii ส่วนควบคุม เป็นศูนย์ควบคุมระบบและบัญชาการการทำงานของระบบจีพีเอสรวมไปถึงการตรวจตราดูความเรียบร้อยของระบบ ตั้งอยู่ที่ฐานทัพอากาศเมือง Colorado Spring สหรัฐอเมริกา ซึ่งศูนย์ควบคุมกลางประกอบด้วย สถานีสังเกตการณ์ (Monitor Station) จำนวน 5 แห่ง กระจายอยู่ตามจุดต่างๆ ของโลก ได้แก่ Hawaii , Kwajalein , Ascension Island , Diego Garcia และ Colorado Spring จานส่งสัญญาณภาคพื้นดิน (Ground Antennas) ซึ่งมีอยู่ด้วยกัน 3 จุด ได้แก่ Ascension Island , Diego Garcia , Kwajalein ศูนย์บัญชาการ (Master Control Station(MCS)) ตั้งอยู่ฐานทัพอากาศสหรัฐฯ Schriever AFB รัฐ Colorado เมื่อสถานีรับสัญญาณจากดาวเทียมมา เพื่อปรับแก้ไขข้อมูลวงโคจรและข้อมูลเวลา(Clock Correction) ของดาวเทียมแต่ละดวงแล้วจะทำการส่งข้อมูลวงโคจรและข้อมูลเวลา กลับไปยังดาวเทียม แล้วดาวเทียมก็จะทำการส่งข้อมูลที่ได้รับการแก้ไขแล้วมาพร้อมกับคลื่นวิทยุมายังเครื่องรับ GPS

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Global Positioning System (GPS) Master Control and Monitor Station Network

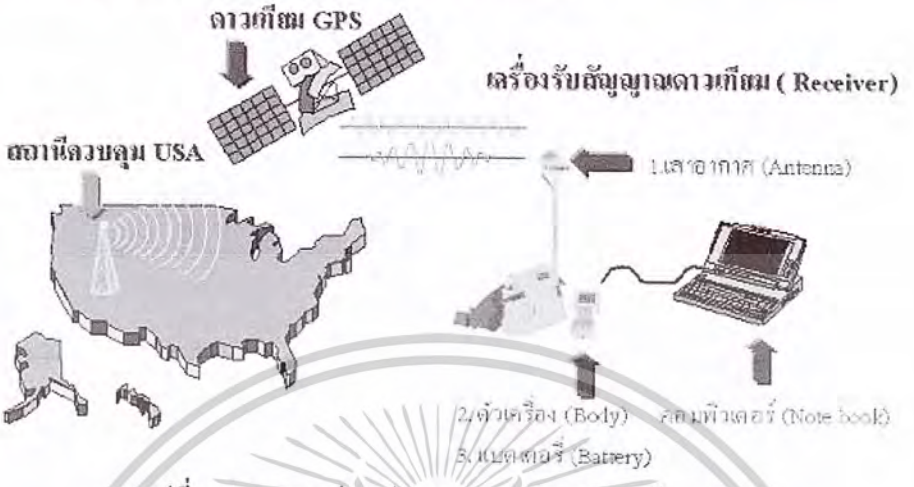
รูปที่ 2.4.2 แสดงสถานีควบคุมระบบดาวเทียม GPS 5 แห่ง

3 ส่วนผู้ใช้งาน (User Segment) ประกอบด้วย 2 ส่วนใหญ่ๆ คือ ส่วนที่เกี่ยวข้องกับทางทหาร (Military) และทางพลเรือน (Civilian) ซึ่งทางพลเรือนจะได้รับสัญญาณฟรีแต่ผู้ใช้ต้องรับผิดชอบหาซื้อจานรับ (Antenna) และเครื่องรับ (Receiver) ด้วยตนเอง นโยบายการให้บริการข้อมูล GPS ของรัฐบาลสหรัฐฯ มีดังนี้ Precise Positioning Services: PPS ใช้ในการทางทหารเป็นหลัก และ Standard Positioning Services: SPS ใช้ในกิจการพลเรือนเป็นหลัก ส่วนผู้ใช้งานประกอบด้วยเครื่องรับสัญญาณ หรือเครื่องจีพีเอสแบบมือถือที่มีใช้กันอยู่ทั่วไปนั่นเอง โดยในเครื่องจีพีเอสนั้นจะมีโปรแกรมคอมพิวเตอร์อยู่ในตัวเครื่องเพื่อให้เครื่องทราบว่าดาวเทียมอยู่ในตำแหน่งใด ในเวลานั้น ๆ โดยเครื่องจีพีเอสจะทำการคำนวณตรวจสอบ และถอดรหัสสัญญาณที่ได้จากดาวเทียมเพื่อให้ได้ข้อมูลมาซึ่งข้อมูลที่ได้โดยปรกติก็มักจะถูกประมวลผลโดยโปรแกรมและส่งข้อมูลออกมาทางหน้าจอของเครื่องจีพีเอสนั้น ๆ เพื่อให้ผู้ใช้ได้ทราบข้อมูล โดยการแสดงผลก็จะต่างกันขึ้นกับโปรแกรมในเครื่องจีพีเอสแต่ละรุ่นและแต่ละยี่ห้อ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.5 ส่วนประกอบของเครื่องรับสัญญาณดาวเทียมจีพีเอส

โดยทั่วไปเครื่องรับสัญญาณดาวเทียมจีพีเอส ประกอบด้วย 3 ส่วนคือ ตัวเครื่อง (Body), ส่วนให้พลังงาน (Power Supply), ส่วนเสาอากาศ (Antenna)



รูปที่ 2.5.1 แสดงส่วนประกอบของระบบดาวเทียม GPS

2.6 การทำงานของจีพีเอส

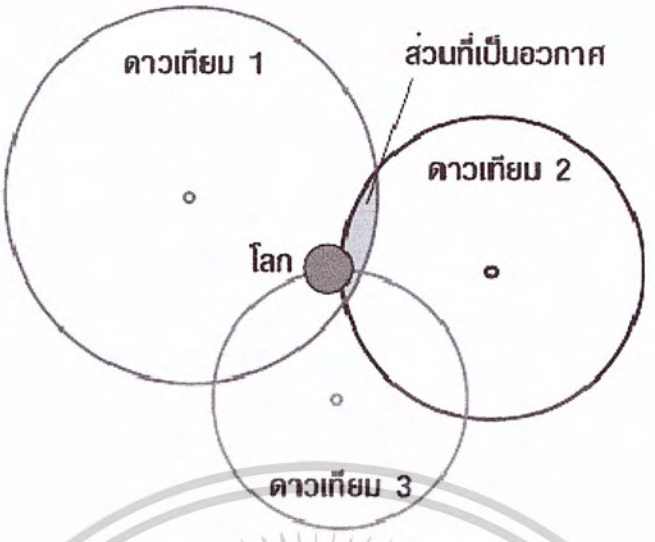
หลักการคำนวณระยะทางระหว่างดาวเทียมกับเครื่องรับจีพีเอส ซึ่งจะต้องใช้ระยะทางจากดาวเทียมอย่างต่ำ 3 ดวง เพื่อให้ได้ตำแหน่งที่แน่นอน ซึ่งเมื่อเครื่องรับจีพีเอสสามารถรับสัญญาณจากดาวเทียมได้ 3 ดวงขึ้นไปแล้ว จะมีคำนวณระยะทางระหว่างดาวเทียมถึงเครื่อง GPS โดยจากสูตรคำนวณทางฟิสิกส์คือ ความเร็ว X เวลา = ระยะทาง โดยดาวเทียมทั้ง 3 ดวงจะส่งสัญญาณที่เหมือนกันมายังเครื่องรับจีพีเอสโดยความเร็วแสง 186,000 ไมล์ต่อวินาที แต่ระยะเวลาในการรับสัญญาณได้จากดาวเทียมแต่ละดวงนั้นจะไม่เท่ากัน เนื่องจากระยะทางไม่เท่ากัน เช่น

ดาวเทียม 1 ระยะเวลาในการส่งสัญญาณจากดาวเทียมดวงแรกถึงเครื่องรับจีพีเอส คือ 0.10 วินาที ระยะทางระหว่างดาวเทียมกับเครื่องรับจีพีเอส คือ 18,600 ไมล์ (186,000 ไมล์ต่อวินาที X 0.10 วินาที = 18,600 ไมล์) ฉะนั้นตำแหน่งปัจจุบันก็จะสามารถเป็นจุดใดก็ได้ในวงกลมที่มีรัศมี 18,600 ไมล์ ซึ่งจะเห็นว่าดาวเทียมเพียงดวงเดียวยังไม่สามารถบอกตำแหน่งที่แน่นอนได้

ดาวเทียม 2 ระยะเวลาในการส่งสัญญาณจากดาวเทียมดวงแรกถึงเครื่องรับจีพีเอสคือ 0.08 วินาที ระยะทางระหว่างดาวเทียมกับเครื่องรับจีพีเอส คือ 13,200 ไมล์ (186,000 ไมล์ต่อวินาที X 0.08 วินาที = 13,200 ไมล์) ฉะนั้นตำแหน่งปัจจุบันก็จะสามารถเป็นจุดใดก็ได้ในจุดระหว่างวงกลมจากดาวเทียมดวงแรกกับดาวเทียมดวงที่ 2

ดาวเทียม 3 ระยะเวลาในการส่งสัญญาณจากดาวเทียมดวงแรกถึงเครื่องรับจีพีเอส คือ 0.06 วินาที ระยะทางระหว่างดาวเทียมกับเครื่องรับจีพีเอส คือ 11,160 ไมล์ (186,000 ไมล์ต่อวินาที X 0.06 วินาที = 11,160 ไมล์) ฉะนั้นตำแหน่งปัจจุบันก็จะสามารถเป็นจุดใดก็ได้ในจุดระหว่างวงกลมจากดาวเทียมทั้ง 3 ดวง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.6.1 หลักการทำงานของจีพีเอส

จะเห็นได้ว่าจะเหลือตำแหน่งอยู่ 2 จุดที่บริเวณวงกลมทั้ง 3 ตัดกันคือตำแหน่งที่อยู่ในอวกาศ ซึ่งเป็นจุดที่อยู่ในอวกาศตำแหน่งนี้จะถูกตัดทิ้งอัตโนมัติโดยเครื่องรับจีพีเอส อีกตำแหน่งคือตำแหน่งบนพื้นโลกซึ่งเป็นตำแหน่งที่เรายืนถือเครื่องรับจีพีเอสอยู่นั้นเอง ซึ่งความถูกต้องแม่นยำของตำแหน่งก็ขึ้นกับจำนวนดาวเทียมที่สามารถรับสัญญาณได้ในขณะนั้นหากมีมากกว่า 3 ดวงก็จะละเอียดมากขึ้น และก็ขึ้นกับเครื่องรับจีพีเอสด้วย หากเป็นเครื่องที่มีราคาแพงที่ใช้เฉพาะงาน ก็จะมีความถูกต้องแม่นยำมากขึ้น

รูปที่ 2.6.2 การใช้ดาวเทียม 3 ดวงขึ้นไปในการหาพิกัด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.7 การประยุกต์ใช้งาน

1 ระบบนำร่อง (Navigation System)

2 ระบบติดตามยานพาหนะ (Automatic Vehicle Location)

3 การสำรวจพื้นที่ (Survey)

4 การทำแผนที่ (Mapping)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.8 ไมโครคอนโทรลเลอร์ตระกูล MCS-51

ไมโครคอนโทรลเลอร์ตระกูล MCS-51 มีสมาชิกในตระกูลหลายเบอร์ด้วยกัน แต่ละเบอร์จะมีคุณสมบัติบางอย่างแตกต่าง เช่น มีหน่วยความจำภายในสำหรับเก็บโปรแกรมและข้อมูลภายในชิปเพิ่มขึ้น มีวงจรเปลี่ยนค่าสัญญาณอนาล็อกเป็นดิจิตอลในตัว สามารถรับสัญญาณอินเตอร์รัพต์ได้หลายชนิด ทำกระบวนการ DMA (Direct Memory Access) ได้ในตัว มีรีจิสเตอร์สำหรับใช้เป็นไทม์เมอร์หรือเคาน์เตอร์เพิ่มขึ้นคุณสมบัติพิเศษที่แตกต่างกันของไมโครคอนโทรลเลอร์แต่ละเบอร์ในตระกูลนี้ดังแสดงในตาราง

ไมโครคอนโทรลเลอร์เบอร์ที่นับได้ว่าเป็นเบอร์พื้นฐานสำหรับตระกูล MCS-51 นี้ ได้แก่ เบอร์ 8051,8031,8751 โดยเบอร์ 8051 จัดเป็นสมาชิกตัวแรกในตระกูล ซึ่งมีหน่วยความจำสำหรับเก็บโปรแกรมภายในชิป เป็น ROM ขนาด 4 กิโลไบต์ และหน่วยความจำสำหรับเก็บข้อมูลทั่วไปภายใน MCS-51 (RAM)เองจำนวน 128 ไบต์ มีพอร์ตขนาด 8 บิต 4พอร์ต มีรีจิสเตอร์สำหรับใช้เป็นไทม์เมอร์หรือเคาน์เตอร์ขนาด 16 บิต รวม 2 ตัว รับสัญญาณอินเตอร์รัพต์จากภายนอกได้ 2 ชนิด สามารถรับและส่งข้อมูลแบบอนุกรมผ่านทางพอร์ตสื่อสารข้อมูลแบบอนุกรม มีวงจรออสซิลเลเตอร์เพื่อสร้างสัญญาณนาฬิกาควบคุมการทำงานในตัวเอง ส่วนเบอร์ 8751 จะมีคุณสมบัติเหมือน เบอร์ 8051 ทุกอย่าง ต่างกันเพียงชนิดของหน่วยความจำสำหรับเก็บโปรแกรมภายในชิป คือ เบอร์ 8751 จะเป็น EPROM แทนที่จะเป็น ROM ส่วนเบอร์ 8031 จะเหมือนกับเบอร์ 8051 ต่างกันเพียงในเบอร์ 8031 ไม่มีหน่วยความจำสำหรับเก็บโปรแกรมภายในชิปเท่านั้น

ไมโครคอนโทรลเลอร์ตระกูล MCS-51 ใช้แรงดันไฟเพียง 5 โวลต์ในการทำงานส่วนกระแสไฟฟ้าที่ใช้จะแตกต่างกันไปตามชนิดของเทคโนโลยีที่ใช้ในการผลิตเบอร์ของไมโครคอนโทรลเลอร์ตระกูลนี้ที่มีตัวอักษร C อยู่ตรงกลางเบอร์ เช่น 80C31, 80C51 จะเป็นเบอร์ของชิปที่ผลิตโดยอาศัยเทคโนโลยี CHMOS ซึ่งใช้พลังงานในการทำงานน้อยกว่าและสามารถควบคุมการใช้พลังงานของตัวชิปได้จากโปรแกรมเพื่อการประหยัดในระบบ

MCS-51 เป็นตระกูลของไมโครคอนโทรลเลอร์ที่ถูกพัฒนาขึ้นมาจากตระกูล MCS-48 ดังนั้นจึงมีความสามารถเหนือกว่าหลายอย่าง ข้อดีของ MCS-51 คือสามารถใช้ความถี่ได้ถึง 12 เมกะเฮิรตซ์ หรือ สำหรับบางเบอร์ในตระกูลสามารถใช้ได้ 16 เมกะเฮิรตซ์ ทำให้ช่วงเวลาในการทำงานแต่ละคำสั่งน้อยมาก เมื่อใช้ความถี่ 12 เมกะเฮิรตซ์ คำสั่งที่ใช้เวลาน้อยสุดจะใช้เวลาเพียง 1 ไมโครวินาที ส่วนคำสั่งที่ใช้เวลามากที่สุดจะใช้เวลาเพียง 4 ไมโครวินาทีเท่านั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.9 การทำงานของ MCS-51

คอมพิวเตอร์จะทำงานด้วยวงจรที่เรียกว่าฮาร์ดแวร์ (Hardware) ประกอบขึ้นมาเพียงอย่างเดียวไม่ได้ จะต้องมีโปรแกรมหรือคำสั่งที่จัดเรียงไว้ให้คอมพิวเตอร์ทำงานตามลำดับใน 8051 ก็เช่นกัน ผู้ใช้จะต้องเขียนโปรแกรมเป็นภาษาเครื่อง ซึ่งอยู่ในรูปของเลขฐาน 2 เก็บไว้ในหน่วยความจำประเภท Program Memory แต่ละคำสั่งของ 8051 อาจประกอบด้วย 1 , 2 หรือ 3 ไบต์ แล้วแต่ว่าจะเป็นคำสั่งให้ทำงานอะไร คอมพิวเตอร์ก็จะเหมือนกับคนที่จะต้องทำงานตามคำสั่ง เมื่อรับคำสั่งแล้วก็จะไปทำตามคำสั่งนั้นเสร็จสิ้นแล้วก็กลับมารับคำสั่งต่อไป

เมื่อเริ่มป้อนไฟเลี้ยงให้กับ 8051 ซึ่งมีวงจร Program Counter ซึ่งเป็นวงจรนับ (Counter Circuit) ชนิดหนึ่ง ส่งค่าตำแหน่งหน่วยความจำสำหรับโปรแกรมลงไปบนบัส (Bus) หมายเลข 1 บัสนี้มีขนาด 16 บิต ค่าตำแหน่งหน่วยความจำนี้จะถูกส่งไปเก็บไว้ที่ Program ADDR Register ที่เป็นวงจร Latch ข้อมูลซึ่งเป็นค่าตำแหน่งหน่วยความจำ จะปรากฏที่บนบัส 16 บิต หมายเลข 2 ถ้าเป็นค่าตำแหน่งหน่วยความจำหลังจากรีเซตค่าตำแหน่งหน่วยความจำจะเป็น 0000H หน่วยความจำสำหรับโปรแกรมจะเลือกได้ว่าเป็น ROM ภายในหรือภายนอก 8051 โดยการป้อนสภาวะลอจิกเข้าไปที่ 8051 ทางขา EA ซึ่งต่ออยู่กับส่วน Timing and Control ทำหน้าที่เป็นวงจรถอดรหัส (Decoder) แล้วสร้างสัญญาณควบคุมต่อไปถ้าสัญญาณลอจิก 0 เข้าไปที่ขา EA จะเป็นการเลือกใช้ ROM ภายใน 8051 โดยใช้วงจร Timing and Control จะสร้างสัญญาณไปยัง ROM ภายในให้ส่งข้อมูลที่เป็นคำสั่งจากตำแหน่งที่ส่งมาทางบัสหมายเลข 2 ข้อมูลจาก ROM จะถูกส่งไปยังบัสหมายเลข 3 ที่เรียกว่า Internal Data Bus แล้วนำไปเก็บไว้ที่ Instruction Register (เป็นวงจร Latch) เพื่อส่งต่อไปให้กับวงจร Timing and Control ทำการถอดรหัสแล้วควบคุมการทำงานส่วนอื่นๆต่อไปแล้วแต่ว่าเป็นคำสั่งให้ทำงานอะไรในกรณีที่เลือก ROM ภายนอก 8051 โดยป้อนสัญญาณลอจิก 1 เข้าไปที่ขา EA จะทำให้วงจร Timing and Control ส่งสัญญาณไปยังพอร์ต 0 และพอร์ต 2 เพื่อส่งค่าตำแหน่งหน่วยความจำบนบัสหมายเลข 2 ออกไปชี้หน่วยความจำภายนอก จากนั้นจะอ่านข้อมูลที่เป็นคำสั่งกลับเข้าทางพอร์ต 0 ไปยัง Internal Data Bus แล้วไปเก็บที่ Instruction Register เรียกว่าเป็นช่วงของการ Fetch (Fetch cycle) ช่วงต่อไปจะเป็นช่วงของการทำงานตามคำสั่งเรียกว่า Execute Cycle เช่น ถ้าเป็นคำสั่งให้บวกข้อมูลในรีจิสเตอร์ Accumulator กับข้อมูลจากหน่วยความจำ Data Memory ภายใน RAM ตำแหน่ง 23H วงจร Timing and Control ก็จะส่งสัญญาณให้ Instruction Register ส่งค่าตำแหน่งหน่วยความจำ 23H ลงไปยัง Internal Data Bus แล้วนำข้อมูลไปเก็บไว้ที่ RAM ADDR Register เพื่อใช้ชี้ตำแหน่งหน่วยความจำ RAM จากนั้น Timing and Control จะสั่ง RAM ส่งข้อมูลที่เก็บอยู่ในหน่วยความจำตำแหน่ง 23H ลงมายัง Internal Data Bus แล้วนำข้อมูลไปเก็บไว้ที่ TMP1(เป็นวงจร Latch) ซึ่งจะทำหน้าที่เก็บสถานะ ผลลัพธ์ของการทำงานใน ALU เช่นผลลัพธ์การบวกมีค่าเกิน 8 บิต ก็จะทำให้บิตหนึ่งใน PSW ถูก SET เป็น 1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.10 โครงสร้างหน่วยความจำ

หน่วยความจำสำหรับ MCS-51 จะมี 2 ชนิดคือ หน่วยความจำที่ใช้เก็บโปรแกรม (ROM) กับหน่วยความจำที่ใช้เก็บข้อมูลในการประมวลผล(RAM) MCS-51 บางเบอร์เช่น 8051 , 8052 จะมีหน่วยความจำโปรแกรมภายในชิป และ MCS-51 ทุกเบอร์สามารถอ้างหน่วยความจำโปรแกรมภายนอกได้มากที่สุด 64K และอ้างหน่วยความจำข้อมูลภายนอกได้มากที่สุด 64K สำหรับหน่วยความจำ RAM ภายใน จะประกอบไปด้วยพื้นที่ใช้งานทั่วไป, รีจิสเตอร์แบงค์ , พื้นที่ใช้งานระดับบิต และรีจิสเตอร์ฟังก์ชันพิเศษ เราอาจเขียนไดอะแกรมของหน่วยความจำของ 8031 ได้ดังรูปที่ 2.13 โดยในรูปจะบอกด้วยว่าขาใดจะแอคทีฟ

ใน 8031 จะมีหน่วยความจำภายในตั้งแต่ตำแหน่ง 00H ถึง FFH และสามารถอ้างหน่วยความจำโปรแกรมภายนอกได้ 64K ตำแหน่ง ถ้าอ่านข้อมูลจากหน่วยความจำโปรแกรมขา PSEN จะแอคทีฟ นอกจากนี้ 8031 สามารถอ้างหน่วยความจำข้อมูลภายนอกได้ 64K ตำแหน่ง โดยการติดต่อกับหน่วยความจำนี้ ขา RD และ WR จะแอคทีฟ สำหรับหน่วยความจำข้อมูลภายในนั้นจะแบ่งออกได้ดังนี้

1 ชุดรีจิสเตอร์ 4 ชุด แต่ละชุดเรียกว่ารีจิสเตอร์แบงค์ที่ตำแหน่ง 00K ถึง 1FH โดยแต่ละชุดประกอบด้วยรีจิสเตอร์ R0 ถึง R7
2 หน่วยความจำที่สามารถเข้าถึงขอมูลระดับบิตได้ ตำแหน่ง 20H ถึง 2FH
3 หน่วยความจำใช้งานทั่วไปตำแหน่ง 30H ถึง 7FH
4 รีจิสเตอร์ฟังก์ชันพิเศษ ตำแหน่ง 80H ถึง FFH

2.11 หน่วยความจำใช้งานทั่วไป

จะเห็นได้ว่าใน 8031 จะมีหน่วยความจำ RAM สำหรับใช้งานทั่วไปจำนวน 80 ไบต์ ตั้งแต่ตำแหน่ง 30H ถึง 7FH ตำแหน่งเหล่านี้สามารถอ้างตำแหน่งแบบ Direct Addressing Mode หรือ Indirect Addressing Mode การย้ายข้อมูลจากตำแหน่งที่เก็บโดยตรง (ตำแหน่ง 5FH) เรียกว่าการอ้างตำแหน่งแบบ Direct Addressing Mode นอกจากนี้ยังสามารถอ่านข้อมูลโดยรีจิสเตอร์ R0 หรือ R1 การการตัวชี้ตำแหน่งได้ เรียกว่าการอ้างตำแหน่งแบบ Indirect Addressing Mode Bit-addressable

RAM ใน MCS-51 จะมีหน่วยความจำที่สามารถอ้างข้อมูลในระดับบิตไดตั้งแต่ตำแหน่ง 20H ถึง 2FH รวม16 ไบต์ โดยสามารถ SET, CLEAR, AND, OR ทางลอจิกได้ จำนวนบิตที่ใช้งานได้ทั้งหมดมีจำนวน 128 บิต (8 บิต x 16 ไบต์)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.12 รีจิสเตอร์ฟังก์ชันพิเศษ (Special Function Register)

ใน MCS-51 รีจิสเตอร์จะใช้หน่วยความจำ RAM ภายในชิป โดยส่วนหนึ่งเป็นรีจิสเตอร์พิเศษ (Special Function Register: SFR) ซึ่งมีทั้งหมด 21 ตัว โดยรีจิสเตอร์พิเศษต่างๆจะเริ่มที่หน่วยความจำตั้งแต่ 80H ถึง FFH ซึ่งมีทั้งหมด 128 ตำแหน่ง แต่จะเป็นรีจิสเตอร์ฟังก์ชันพิเศษเพียง 21 ตำแหน่ง แต่ถ้าเป็น 8032/8051 จะใช้ 26 ตำแหน่งหรือมี SFR 26 ตัว

2.13 Program Status World

รีจิสเตอร์ตัวนี้เรียกย่อๆว่า PSW จะอยู่ที่ตำแหน่ง D0H ซึ่งสามารถเข้าถึงข้อมูลระดับบิตได้ โดยรีจิสเตอร์นี้จะเป็นตัวบอกสภาวะต่างๆของไมโครคอนโทรลเลอร์ ความหมายของแต่ละบิตแสดงได้ดังตารางที่ 2.13.1

บิต	ชื่อบิต	ตำแหน่ง	ความหมาย
PSW.7	CY	D7H	Carry Flag
PSW.	AC	D7H	Auxiliary Carry Flag
PSW.	F0	D7H	Flag 0
PSW.	RS1	D7H	บิตสำหรับการเลือก Register Bank 1
PSW.	RS0	D7H	บิตสำหรับการเลือก Register Bank 0
			00 = Bank 0; Address 00H – 07H 01 = Bank 1; Address 08H – 0FH 10 = Bank 2; Address 10H – 17H 11 = Bank 3; Address 18H – 1FH
PSW.	OV	D7H	Overflow Flag
PSW.	-	D7H	Reserved
PSW.	P	D7H	Even Parity Flag

ตารางที่ 2.13.1 แสดงบิตและหน้าที่ต่างๆใน PSW

แฟลกตัวทด Carry Flag (CF) บิตนี้เป็นบิตที่ 7 ของ PSW บิตนี้จะมีความสำคัญหากมีการกระทำทางคณิตศาสตร์ โดยบิตนี้จะ Set เมื่อเกิดการทดของตัวทดบิตที่ 7 ขณะทำการบวกเลขหรือ Set เมื่อเกิดการยืมของบิตที่ 7 เมื่อเกิดการลบเลข

แฟลกตัวช่วยทด Auxiliary Carry Flag เมื่อมีการบวกแบบ Binary-Code-Decimal (BCD) บิต Auxiliary Carry Flag (AC) หรือบิตตัวช่วยทดจะถูก Set เมื่อมีการทดจากบิตที่ 3 ไปบิตที่ 4

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หรือถ้าใน Lower Nibble มีค่าระหว่าง 0AH-0FH เนื่องจากรหัส BCD นี้มีค่าได้มากที่สุดแค่ 9 ถ้าหากการบวกเลขแบบ BCD จะต้องตามด้วยคำสั่ง DAA (Decimal Adjust Accumulator) เพื่อปรับค่าที่มีค่าเกิน 9 โดยบวกเลข 6 เข้าไป จะทำให้เป็นรหัส BCD ที่แทนเลขฐานสิบได้

แฟลกศูนย์ (Flag 0) เป็น Flag ที่ผู้ใช้สามารถใช้งานทั่วไปได้

บิตเลือกรีจิสเตอร์แบงก์ (Register Bank Select Bits) ตามที่เราทราบมาแล้วว่าใน MCS-51 จะมีชุดรีจิสเตอร์อยู่ 4 ชุด ถ้าจะเลือกให้ใช้ชุดใดแอคทีฟจะกำหนดได้ในบิต RS1และ RS2 ของ PSW และจะ Clear ตัวเองเมื่อระบบถูกรีเซต

แฟลกโอเวอร์โฟลว์ Overflow Flag แฟลก OV จะถูก Set หลังจากการกระทำทางคณิตศาสตร์แล้วเกิด Overflow คือ จำนวนที่เกิดจากการบวกหรือการลบ มีค่าเกินกว่าที่จำนวนไบต์จะเป็นไปได้คือ มากกว่า +128 น้อยกว่า -128

บิตพาริตี้ (Parity Bit) พาริตี้บิต (P) ในบิตที่บอกค่าพาริตี้ของรีจิสเตอร์ Accumulator ซึ่งอาจตรวจสอบความถูกต้องของข้อมูลได้ โดยจะเซตหรือเคลียร์ ขึ้นกับผลที่เกิดขึ้นกับ Accumulator

2.14 รีจิสเตอร์ B (B Register)

รีจิสเตอร์ B จะอยู่ตำแหน่ง F0H ของหน่วยความจำข้อมูลภายใน เป็นรีจิสเตอร์ที่สามารถใช้งานทั่วไปได้ โดยทั่วไปรีจิสเตอร์นี้จะใช้คูณหรือหารกับรีจิสเตอร์ Accumulator เช่น การทำคำสั่ง MUL AB ซึ่งเป็นการคูณแบบ 8 บิต โดยผลลัพธ์ที่ได้จะมีขนาด 16 บิต ซึ่งรีจิสเตอร์ A จะเก็บค่า 8 บิตต่ำ รีจิสเตอร์ B จะเก็บค่า 8 บิตสูง สำหรับการหารโดยการทำคำสั่ง DIV AB โดยค่าใน A จะถูกหารด้วย B ผลลัพธ์ที่ได้จะเก็บใน รีจิสเตอร์ AB โดย B จะเก็บค่า 8 บิตต่ำและ A จะเก็บค่า 8 บิตสูง รีจิสเตอร์ B นี้สามารถเข้าถึงข้อมูลระดับบิตได้ โดยตำแหน่งของบิตคือตำแหน่ง F0H ถึง F7H

2.15 ตัวชี้สแตค (Stack Pointer)

Stack Pointer (SP) เป็นรีจิสเตอร์ขนาด 8 บิต อยู่ที่ตำแหน่ง 81H การเขียนค่าเข้าไปในตำแหน่งที่ SP ชี้อยู่นี้ เรียกว่า “Pushing” สำหรับการอ่าน ค่าที่ SP ชี้อยู่ เรียกว่า “Popping” ค่าของ SP จะเพิ่มขึ้นหนึ่งก่อนที่จะเขียนข้อมูลลงไป และจะลดลงหนึ่งเมื่ออ่านข้อมูลออกมาแล้ว หากโปรแกรมทำคำสั่ง CALL จะใช้รีจิสเตอร์สแตคนี้เก็บค่าตำแหน่งเดิมของโปรแกรม (PC) ก่อนที่จะทำโปรแกรมย่อยเมื่อทำโปรแกรมย่อยเสร็จแล้วจะคืนค่าในสแตคให้กับ PC ตามเดิม โดยปกติค่า PC จะกำหนดให้อยู่ใน RAM ภายใน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.16 รีจิสเตอร์ Data Pointer (DPTR)

รีจิสเตอร์นี้ใช้สำหรับชี้ตำแหน่งรหัสโปรแกรมหรือข้อมูลในหน่วยความจำ โดยรีจิสเตอร์ขนาด 16 บิต ซึ่งประกอบด้วยรีจิสเตอร์ 2 ตัว คือ DPL ตำแหน่งที่ 82H โดยจะเก็บเป็น 8 บิตต่ำ และ DPH ตำแหน่ง 83H โดยเก็บเป็นค่า 8 บิตสูง รีจิสเตอร์ทั้งสองตัวนี้จะรวมกันกลายเป็นรีจิสเตอร์ 16 บิต

2.17 รีจิสเตอร์พอร์ต (Port Registers)

ใน MCS-51 ค่าของพอร์ตจะหมายถึงค่าของหน่วยความจำด้วย หากต้องการส่งขอมูลออกไปที่พอร์ต ก็เพียงแต่เขียนข้อมูลไปที่หน่วยความจำตำแหน่งที่พอร์ตนั้นอยู่ และถ้าหากต้องการจะอ่านข้อมูลจากพอร์ต ก็เพียงอ่านค่าจากตำแหน่งที่หน่วยความจำนั้นอยู่ ใน MCS-51 พอร์ต 0 จะอยู่ที่ตำแหน่ง 80H , พอร์ต 1 จะอยู่ที่ตำแหน่ง 90H , พอร์ต 2 จะอยู่ที่ตำแหน่ง A0H และพอร์ต 3 จะอยู่ที่ตำแหน่ง B0H พอร์ต 0,2 และ 3 โดยทั่วไปแล้วจะไม่ใช้หากติดต่อกับหน่วยความจำภายนอกหรือใช้เป็นพอร์ตพิเศษ โดยปกติแล้วจะใช้พอร์ต 1 ในการติดต่อกับอุปกรณ์ภายนอก พอร์ตทุกพอร์ตสามารถอ้างข้อมูลในระดับบิตได้

2.18 รีจิสเตอร์เวลา (Timer Registers)

ใน MCS-51 เบอร์ 8051 จะมีรีจิสเตอร์ที่ใช้นับและจับเวลาขนาด 16 บิต 2 ตัว คือ Timer 0 อยู่ที่ตำแหน่ง 8AH และ 8CH โดยตำแหน่ง 8AH หมายถึง TL0 ซึ่งจะเป็น 8 ไบต์ต่ำ และหมายถึง 8 ไบต์สูง รีจิสเตอร์อีกตัวคือ Timer 1 โดยแบ่งเป็น TL1 อยู่ที่ตำแหน่ง 8BH เป็นไบต์ต่ำ และ TH1 อยู่ที่ตำแหน่ง 8DH เป็นไบต์สูง การใช้ Timer จะต้องกำหนดการทำงานในรีจิสเตอร์ TMOD (Timer/Count Mode Control Register) ซึ่งอยู่ในตำแหน่ง 88H เสียก่อน

2.19 รีจิสเตอร์พอร์ตอนุกรม (Serial Port Registers)

MCS-51 จะมีพอร์ตสื่อสารอนุกรมอยู่ (Serial Port) ภายในชิป ซึ่งสามารถจะรับหรือส่งข้อมูลได้โดยติดต่อผ่านรีจิสเตอร์ SBUF (Serial Data Buffer) ซึ่งอยู่ในตำแหน่ง 99H โดยถ้าต้องการส่งข้อมูลแบบอนุกรมให้เขียนข้อมูลไปที่จิสเตอร์นี้ ตัว Serial Port สามารถโปรแกรมให้ทำงานได้ 4 โหมด โดยโปรแกรมผ่านรีจิสเตอร์ SCON (Serial Port Control Registers) ตำแหน่ง 98H

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.20 รีจิสเตอร์อินเทอร์รัพท์ (Interrupt Port Registers)

MCS-51 สามารถ Interrupt ได้ 5 ตำแหน่งโดยมี 2-Priority ตัว Interrupt นี้จะถูก Disable หลังจากระบบถูกรีเซต และจะ Enabled หลังจากที่เขียนข้อมูลไปที่รีจิสเตอร์ IE หรือตำแหน่ง A8H ลำดับความสำคัญสามารถเซตได้ที่รีจิสเตอร์ IP หรือตำแหน่ง 88H

2.21 Power Control Register (PCON)

รีจิสเตอร์ PCON อยู่ที่ตำแหน่ง 87H ใช้หยุดการทำงานของ MCS-51 โดยจะหยุดจ่ายสัญญาณนาฬิกาให้ระบบ ทำให้ข้อมูลต่างๆภายใน MCS-51 ไม่มีการเปลี่ยนแปลงนอกจากนี้ยังลดพลังงานไฟฟ้าที่จ่ายให้ MCS-51 ลงด้วย

2.22 หน่วยความจำภายนอก (External Memory)

MCS-51 สามารถอ้างหน่วยความจำข้อมูลภายนอกได้ 64K และอ้างหน่วยความจำโปรแกรมภายนอกได้ 64K จะใช้พอร์ต 0 ในการอ้างตำแหน่งหน่วยความจำ 8 บิตล่างและใช้พอร์ต 0 เป็นพอร์ตข้อมูล (DATA) ด้วย โดยขา A LE มาเป็น Latch ข้อมูลพอร์ต 0 และใช้พอร์ต 2 เป็นขาอ้างตำแหน่ง 8 บิตบน(รวมขาอ้างตำแหน่ง 16 เส้น ซึ่งอ้างได้ 64K)เนื่องจากพอร์ต 0 จะใช้งาน 2 หน้าที่ในการติดต่อกับหน่วยความจำ จะใช้วิธี Multiplex ระหว่าง Address กับ Data พิจารณาจากรูป ถ้าต้องการติดต่อกับหน่วยความจำที่เก็บข้อมูล 8 บิต และเก็บได้ 64K จะต้องใช้สายสัญญาณ 24 เส้น คือ เป็นขา Address 16 เส้น และขาข้อมูล 8 เส้น

2.23 การติดต่อกับหน่วยความจำโปรแกรมภายนอก

ในการอ่านข้อมูลจากหน่วยความจำโปรแกรมภายนอก MCS-51 จะส่งค่าตำแหน่งของหน่วยความจำออกไปก่อน ซึ่งค่าตำแหน่งจะเก็บอยู่ใน PC โดยส่งออกไปทางพอร์ต 0 และพอร์ต 2 จากนั้นเวลาต่อมาจะส่งขา ALE ให้เป็นลอจิก “0” เพื่อ Latch ขา Address ของ 8 บิตต่ำ คือ พอร์ต 0 จากนั้นจะส่งสัญญาณทางขา PSEN ให้เป็นลอจิก “0” เพื่ออ่านข้อมูลซึ่งจะได้ Opcode เข้าไปทางขา Data Bus คือ พอร์ต 0

สำหรับการต่อหน่วยความจำกับ MCS-51 โดยขา EA จะต่อเป็น “0” เพื่อบอก MCS-51 ว่าอ่านหน่วยความจำโปรแกรมภายนอก สำหรับการ Multiplex จะใช้ฟลิปฟลอป 8 ตัวเบอร์ 74373 เก็บค่าตำแหน่ง 8 บิตต่ำเอาไว้ เมื่อ MCS-51 ส่งค่าตำแหน่งพอร์ตออกไปเวลาต่อมาจะส่งขา ALE ให้เป็น “0” ซึ่งจะใช้ขานี้ต่อกับ 74373 เพื่อให้ Latch ข้อมูลสำหรับขา PSEN จะต่อกับขา Output Enable (OE) ของหน่วยความจำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

2.24 การติดต่อกับหน่วยความจำข้อมูลภายนอก

หน่วยความจำข้อมูลภายนอก MCS-51 สามารถอ่านและเขียนได้ ในการติดต่อกับหน่วยความจำข้อมูลภายนอก MCS-51 จะส่งขา Address ออกไปทางพอร์ต 0 และพอร์ต 2 จากนั้นจะส่งขา ALE เพื่อไป Latch Address 8 บิตต่ำ โดยการอ่านเขียนข้อมูลนั้นจะใช้ขา RD หรือ P3.7 และขา WR หรือ P3.6 ตามลำดับ

2.25 Reset Operation

การรีเซตหรือเริ่มตนทำงานใหม่ของ MCS-51 จะต้องให้ลอจิก "1" ที่ขา RST เป็นเวลา 2 Machine Cycle (1 Machine Cycle เท่ากับ 12 Clock) จากนั้นให้กลับเป็นลอจิก "0" การรีเซตอาจทำได้โดยใช้สวิตช์กดหรือใช้วิธี Power-up โดยใช้ตัว R-C ต่อเป็นวงจร

เมื่อ MCS-51 ถูกรีเซต ค่ารีจิสเตอร์ต่างๆจะถูกกำหนดค่าตาราง โดย PC จะชี้ไปอยู่ที่ตำแหน่งเริ่มต้น คือ 0000H เมื่อขา RST กลับเป็น "0" MCS-51 จะเริ่มทำโปรแกรมที่ตำแหน่งแรก

2.26 TIMER

ตัว Timer อาจพิจารณาได้ง่ายๆว่าเป็นตัวฟลิปฟลอปมาต่อเรียงกันโดยมี Clock เป็นอินพุต สำหรับเอาต์พุตที่ออกมาจากฟลิปฟลอปแต่ละตัวจะถูกหารด้วย 2 พิจารณาการต่อฟลิปฟลอป ถ้าใส่ Clock เข้าไปในฟลิบฟลอปตัวแรก ความถี่ของ Clock ที่ออกจากเอาต์พุตตัวแรกจะถูกหารด้วย 2 และเอาต์พุตที่จะต่อกับฟลิปฟลอปตัวที่สอง และสัญญาณที่ออกมาจะถูกหารด้วย 2 อีก ดังนั้นถ้ามีฟลิปฟลอปต่ออยู่ n Stage จะหารสัญญาณนาฬิกาได้ 2^n ถ้าให้เอาต์พุต Stage สุดท้ายของ Timer เป็น Overflow Flip-flop หรือ Flag และจะให้เอาต์พุตออกมาเมื่อมีการนับเป็น Overflow เช่น ถ้าเป็นตัวนับแบบ 16 บิต(มีฟลิปฟลอปต่ออยู่ 16 ตัว) วงจรจะนับตั้งแต่ 0000H ถึง FFFFH เมื่อฟลิปฟลอปเปลี่ยนจาก 0000H จะให้บิต Overflow ออกมา ใน MCS-51 จะมีตัวจับเวลาอยู่ภายในชิป ถ้าเป็นเบอร์ 8051 หรือ 8031 จะมี 2 ตัว คือ Timer 0 และ Timer 1 แต่ถ้าเป็นเบอร์ 8052 จะมีเพิ่มอีกหนึ่งตัวคือ Timer 2 รีจิสเตอร์ต่างๆที่เกี่ยวข้องกับการใช้ Timer

61876
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.27 โมดูลรับสัญญาณจีพีเอสมาตรฐาน NMEA

NMEA เป็นโปรโตคอลสำหรับการสื่อสารซึ่งกำหนดขึ้นโดยองค์กรกลางคือ National Marine Electronics Association ในแรกเริ่มนั้น NMEA ถูกพัฒนาขึ้นมาเพื่อใช้ในการแลกเปลี่ยนข้อมูลกับเซนเซอร์อิเล็กทรอนิกส์ที่ใช้ในการเดินเรือเป็นหลัก ต่อมาเมื่อระบบ GPS ถูกนำมาใช้และมีบทบาทในการเดินเรือมากขึ้นตามวันเวลาที่ผ่านไป จึงทำให้ NMEA ถูกพัฒนามาเป็นมาตรฐานกลางสำหรับใช้สื่อสารระหว่างอุปกรณ์รับสัญญาณ GPS และอุปกรณ์ต่อพ่วงอื่นๆ (Terminal equipment) แต่ถึงกระนั้นก็ยังคงมีอุปกรณ์รับสัญญาณ GPS อยู่บ้างบางผู้ผลิตที่มีโปรโตคอลเฉพาะสำหรับใช้งานเอง

สำหรับมาตรฐาน NMEA ที่หมายถึงในที่นี้คือมาตรฐานซึ่งมีชื่อเรียกเต็มๆว่า NMEA 04183 เวอร์ชัน 1.5 หรือ 2.2 ซึ่งเป็นสำหรับมาตรฐานที่ใช้กัน อย่างไรก็ตามมาตรฐาน NMEA 0183 เวอร์ชัน 2.2 ซึ่งเป็นมาตรฐานถูกประกาศใช้มาตั้งแต่วันที่ 1 มกราคม ค.ศ. 1997 เป็นเวอร์ชันซึ่งใหม่กว่าและในปัจจุบันอุปกรณ์รับสัญญาณ GPS ส่วนใหญ่สามารถรองรับได้

2.28 NMEA Message

NMEA message คือข้อมูลซึ่งส่งออกมาจากโมดูลรับสัญญาณ GPS ข้อมูลใน NMEA message สามารถแบ่งได้เป็นเรคอร์ด (record) หรือฟิลด์ (field) ย่อย โดยในแต่ละเรคอร์ดจะประกอบด้วยอักขระแอสกี (ASCII) ซึ่งมีความยาวรวมไม่เกิน 80 ตัวอักษร เราสามารถอ่านดูข้อมูล NMEA message ที่ว่านี้ได้โดยการใช้ซอฟต์แวร์สื่อสาร เช่น HyperTerminal อย่างที่กล่าวไปแล้วข้างต้น เรคอร์ดข้อมูลใน NMEA message แต่ละเวอร์ชันอาจมีอย่มากน้อยแตกต่างกัน แต่เรคอร์ดที่มักใช้กันเป็นหลักใน NMEA message จะมีอยู่ 6 เรคอร์ด ดังตารางที่ 2.28.1

NMEA Record	Description
GGA	Global positioning system fixed data
GLL	Geographic position-latitude/longitude
GSA	GNSS DOP and active satellites
GSV	GNSS satellites in view
RMC	Recommended minimum specific GNSS data
VTG	Course over ground and ground speed

ตารางที่ 2.28.1 เรคอร์ดหลักๆใน NMEA message ที่มักถูกนำมาใช้งาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

NAME	Example	Unit	Description
Message ID	$GPGGA		GGA protocol header
UTC Position	161229.487		hhmmss.sss
Latitude	3723.2475		ddmm.mmmm
N/S Indicator	N		N = north or S = south
Longitude	12158.3416		ddmm.mmmm
E/W Indicator	W		E = east or W = west
Position Fix Indicator	1		
Satellites Used	07		Range 0 to 12
HDOP	1.0		HoriZontal Dilution of Precision
MSL Altitude	9.0	meters	
Units	M	meters	
Geoid Separation		meters	
Units	M	meters	
Age of Diff. Corr.		second	Null fields when DGPS is mot used
Diff. Ref. Station ID	0000		
Checksum	*18		
<CR> <LF>			End of message termination

Value	Description
0	Fix not available or invalid
1	GPS SPS Mode , fix valid
2	Differential GPS , SPS Mode , fix valid
3	GPS PPS Mode , fix valid

ตารางที่ 2.28.2 ความหมายของข้อมูลที่อยู่ในเรคอร์ด GGA

รายละเอียดภายในเรคอร์ดต่างๆของ NMEA Message

GGA (Global Positioning System Fixed Data) เรคอร์ดนี้ประกอบด้วยข้อมูลซึ่งใช้บอกถึงตำแหน่งพิกัด ละติจูด, ลองติจูด, เวลา, จำนวนดาวเทียมที่ใช้คำนวณพิกัด (Satellites used) และความสูงจากระดับน้ำทะเล (MSL Altitude) โดยตัวอย่างของเรคอร์ด GGA ที่โมดูลรับสัญญาณ GPS ส่งออกมา จะมีโครงสร้างเป็นดังนี้ (ดูตารางที่ 2.28.2 ประกอบ)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$GPGGA , 161229.487 , 3723.2475 , N ,12158.3416 , W , 1 ,07 , 1.0 , 9.0 , M ,,,
0000*18<CR> <LF>

GLL (Geographic Position Latitude/Longitude) เรคอร์ดนี้ประกอบด้วยข้อมูลซึ่งใช้บอกถึงตำแหน่งพิกัด ละติจูด, ลองติจูด, เวลา และสถานะในการรับสัญญาณ (Status) โดยตัวอย่างของเรคอร์ด GLL ที่โมดูลรับสัญญาณ GPS ส่งออกมา จะมีโครงสร้างเป็นดังนี้ (ดูตารางที่ 2.28.3 ประกอบ)

$GPGLL, 3723.2475, N, 12158.3416, W, 161229.487, A*2C<CR> <LF>

NAME	Example	Unit	Description
Message ID	$GPGLL		GGA protocol header
Latitude	3723.2475		ddmm.mmmm
N/S Indicator	N		N = north or S = south
Longitude	12158.3416		ddmm.mmmm
E/W Indicator	W		E = east or W = west
UTC Position	161229.487		hhmmss.sss
Status	A		A = data valid or V = data not valid
Checksum	*18		
<CR> <LF>			End of message termination

ตารางที่ 2.28.3 ความหมายของข้อมูลที่อยู่ในเรคอร์ด GLL

GSA (GNSS DOP and Active Satellites) เรคอร์ดนี้ประกอบด้วยข้อมูลซึ่งใช้บอกถึงตำแหน่งพิกัด ละติจูด, ลองติจูด, ทิศทาง, เวลา และสถานะในการรับสัญญาณ (Status) โดยตัวอย่างของเรคอร์ด GSA ที่โมดูลรับสัญญาณ GPS ส่งออกมา จะมีโครงสร้างเป็นดังนี้ (ดูตารางที่ 2.28.4 ประกอบ)

$GPGSA , A , 3 , 07 , 02 , 26 , 27 , 09 , 04 , 15 ,,, 1.8 , 1.0 , 1.5 *33<CR> <LF>

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

NAME	Example	Unit	Description
Message ID	$GPGSA		GGA protocol header
Mode 1	A		
Mode 2	3		
Satellites Used	07		Sv on Channel 1
Satellites Used	02		Sv on Channel 2
...			...
Satellites Used			Sv on Channel 12
PDOP	1.8		Position Dilution of Precision
HDOP	1.0		HoriZontal Dilution of Precision
VDOP	1.5		Vertical Dilution of Precision
Units	M	meters	
Checksum	*33		
<CR> <LF>			End of message termination

Value	Description
1	Fix not available
2	2D
3	3D

Value	Description
M	Manual-force to operate in 2D or 3D mode
A	Automatic-allowed to automatically switch 2D/3D

ตารางที่ 2.28.4 ความหมายของข้อมูลที่อยู่ในเรคอร์ด GSA

GSV (GNSS Satellites in View) เรคอร์ดนี้ประกอบด้วยข้อมูลซึ่งใช้บอกค่าเทคนิคต่างๆที่ได้รับจากดาวเทียม GPS ที่โมดูลรับสัญญาณได้ โดยตัวอย่างของเรคอร์ด GSV ที่โมดูลรับสัญญาณ GPS ส่งออกมา จะมีโครงสร้างเป็นดังนี้ (ดูตารางที่ 2.28.5 ประกอบ)

$GPGSV , 2 , 1, 07 , 07 , 79 , 048 , 42 , 02 , 51 , 062 , 43 , 26 , 36 , 256 , 42 , 27 , 27 , 138 , 42 *71<CR> <LF>

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

NAME	Example	Unit	Description
Message ID	$GPGSV		GSV protocol header
Number of Message	2		Range 1 to 3
Message Number	1		Range 1 to 3
Satellites in View	07		
Satellites ID	07		Channel (Range 1 to 32)
Elevation	79	degree	Channel (Maximum 90)
Azimuh	048	degree	Channel (True, Range 0 to 359)
SNR(C/No)	42	dBHz	Range 0 to 99, null when not tracking
...			...
Satellites ID	27		Channe4(Range 1 to 32)
Elevation	27	degree	Channe4 (Maximum 90)
Azimuh	138	degree	Channe4 (True, Range 0 to 359)
SNR(C/No)	42	dBHz	Range 0 to 99, null when not tracking
Checksum	*71		
<CR> <LF>			End of message termination

ตารางที่ 2.28.5 ความหมายของข้อมูลที่อยู่ในเรคอร์ด GSV

RMC (Recommended Minimum Specific GNSS Data) เรคอร์ดนี้ประกอบด้วยข้อมูลซึ่งใช้บอกถึงค่าวันที่และเวลา, สถานะในการรับสัญญาณตำแหน่งพิกัดละติจูดและลองติจูด, ทิศทาง, ความเร็ว โดยตัวอย่างของเรคอร์ด RMC ที่โมดูลรับสัญญาณ GPS ส่งออกมา จะมีโครงสร้างเป็นดังนี้ (ดูตารางที่ 2.28.6 ประกอบ)

$GPRMC , 161229.487 , A , 3723.2475 , N , 12158.3416 , W , 0.13 , 309.62 , 120598 ,,
*10<CR> <LF>

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

NAME	Example	Unit	Description
Message ID	$GPRMC		GGA protocol header
UTC Position	161229.487		hhmmss.sss
Status	A		A = data valid or V = data not valid
Latitude	3723.2475		ddmm.mmmm
N/S Indicator	N		N = north or S = south
Longitude	12158.3416		ddmm.mmmm
E/W Indicator	W		E = east or W = west
Speed Over Ground	0.13	knot	
Course Over Ground	309.62	degree	True
Date	120598		ddmmyy
Magnetic Variation		degree	E = east or W = west
Checksum	*10		
<CR> <LF>			End of message termination

ตารางที่ 2.28.6 ความหมายของข้อมูลที่อยู่ในเรคอร์ด RMC

VTG (Course over Ground and Ground Speed) เรคอร์ดนี้ประกอบด้วยข้อมูลซึ่งใช้บอกถึงทิศทางและความเร็ว โดยตัวอย่างของเรคอร์ด VTG ที่โมดูลรับสัญญาณ GPS ส่งออกมา จะมีโครงสร้างเป็นดังนี้ (ดูตารางที่ 2.28.7 ประกอบ)

$GPVTG, 309.62, T,, M , 0.13 , N , 0.2 , K*6E<CR> <LF>

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

NAME	Example	Unit	Description
Message ID	$GPVTG		VTG protocol header
Course	309.62	degree	Measured heading
Reference	T		True
Course		degree	Measured heading
Reference	M		Magnetic
Speed	0.13	knot	Measured horizontal speed
Units	N		Knot
Speed	0.2	Km/hr	Measured horizontal speed
Units	K		Kilometer horizontal speed
Checksum	*6E		
<CR> <LF>			End of message termination

ตารางที่ 2.28.7 ความหมายของข้อมูลที่อยู่ในเรคอร์ด VTG

จากรายละเอียดของแต่ละเรคอร์ดภายใน NMEA message ที่กล่าวมาข้างต้น จะเห็นได้ว่าแต่ละเรคอร์ดต่างก็มีประโยชน์ใช้สอยเฉพาะตัวที่แตกต่างกันซึ่งเราสามารถหยิบมาใช้งานได้ตามความเหมาะสม เมื่อต้องการนำข้อมูลใดมาใช้งานก็จะต้องเลือกเรคอร์ดที่เหมาะสมซึ่งข้อมูลนั้นๆอยู่ ยกตัวอย่างเช่น ในกรณีที่ต้องการทราบความเร็วในการเคลื่อนที่ก็จะต้องเลือกอ่านเรคอร์ด RMC หรือ VTG เป็นต้น ในที่นี้ได้ทำการสรุปและจัดหมวดหมู่คุณสมบัติของแต่ละเรคอร์ดไว้ดังตารางที่ 2.28.8 เพื่อเป็นการสรุปความและเพื่อให้สามารถหยิบมาใช้สอยได้โดยง่าย

กลุ่มข้อมูลที่ต้องการ	เรคอร์ดที่เก็บข้อมูลที่ต้องการไว้
การระบุพิกัดตำแหน่ง	$GPGGA,$GPGLL,$GPRMC
ความเร็ว	$GPRM,$GPVTG
วัน , เวลา	$GPRMC,$GPGGA,$GPGLL
ระดับแนวระนาบ , ความสูง	$GPGSA,$GPGGA
ข้อมูลของดาวเทียม	$GPGSV
สถานะของตัวรับ	$GPGGA,$GPGSA
การแก้ไขในเรื่อง DGPS	$GPGGA

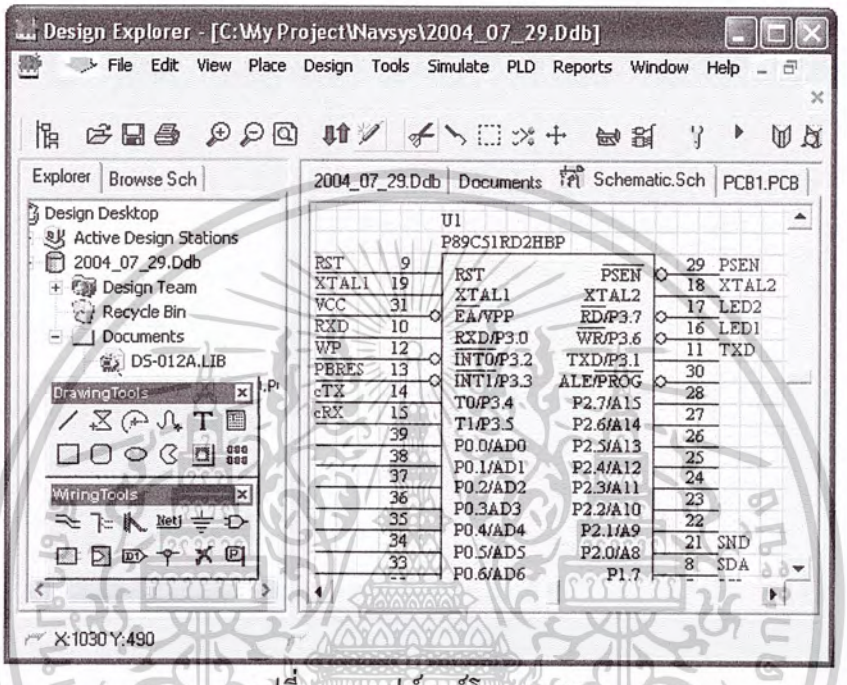
ตารางที่ 2.28.8 ตารางสรุปคุณสมบัติของ 6 เรคอร์ดหลักใน NMEA message

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3
การออกแบบและการสร้าง

3.1 การออกแบบวงจร

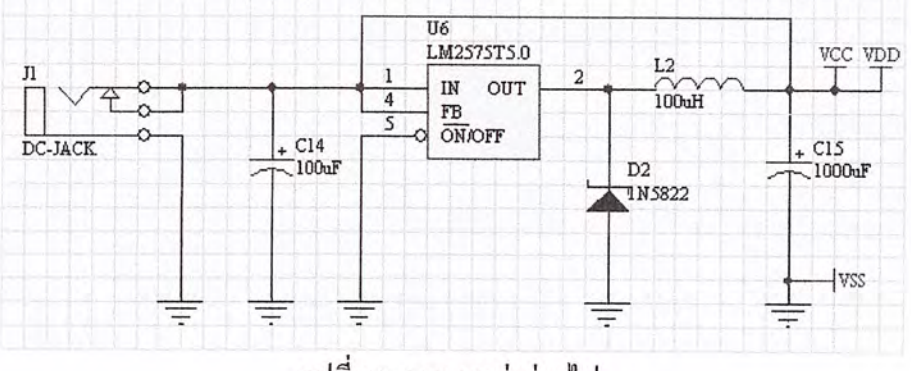
เป็นการออกแบบวงจรอิเล็กทรอนิกส์โดยใช้ซอฟท์แวร์คอมพิวเตอร์ช่วยอำนวยความสะดวกในการออกแบบ คือซอฟท์แวร์โพรเทล99 (Protel99)



รูปที่ 3.1 ซอฟท์แวร์โพรเทล99

ซึ่งเป็นซอฟท์แวร์ที่มีความสมบูรณ์ในเรื่องการออกแบบทางอิเล็กทรอนิกส์ในเรื่องการสร้างวงจร การเดินลายทองแดง(PCB) การจำลองการทำงาน (Simulate) การออกแบบพีแอลดี (PLD)

3.2 วงจรแหล่งจ่ายไฟ

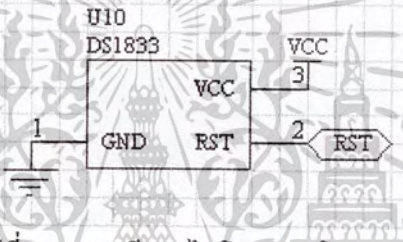


รูปที่ 3.2 วงจรแหล่งจ่ายไฟ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 3.2 เป็นวงจรแหล่งจ่ายไฟแบบสวิตชิ่ง(Switching Power Supply) โดยอาศัยการทำงานของไอซี LM2575T5.0 ซึ่งทำหน้าที่เป็นวงจรรวมของแหล่งจ่ายไฟแบบสวิตชิ่งที่มีแรงดันเอาท์พุทออก 5 โวลต์ สามารถจ่ายกระแสไฟฟ้าได้สูงสุด 1 แอมแปร์ (Ampere) และสามารถรองรับแรงดันอินพุทได้สูงสุด 60 โวลต์ เหตุที่ต้องการความสามารถในการรองรับแรงดันอินพุทที่สูงๆ ก็เพราะว่าเครื่องบันทึกการเดินทางนี้ส่วนใหญ่จะนำไปติดในรถยนต์ ในการทำงานของรถยนต์จะมีไฟกระชากมาเป็นระยะทำให้อุปกรณ์แหล่งจ่ายไฟเสียหาย อาจจะส่งผลไปถึงส่วนต่างของวงจรด้วย วงจรแหล่งจ่ายไฟแบบสวิตชิ่งทำงานที่ความถี่ 52 kHz

3.3 วงจรรีเซตไมโครคอนโทลเลอร์ (EconoReset)

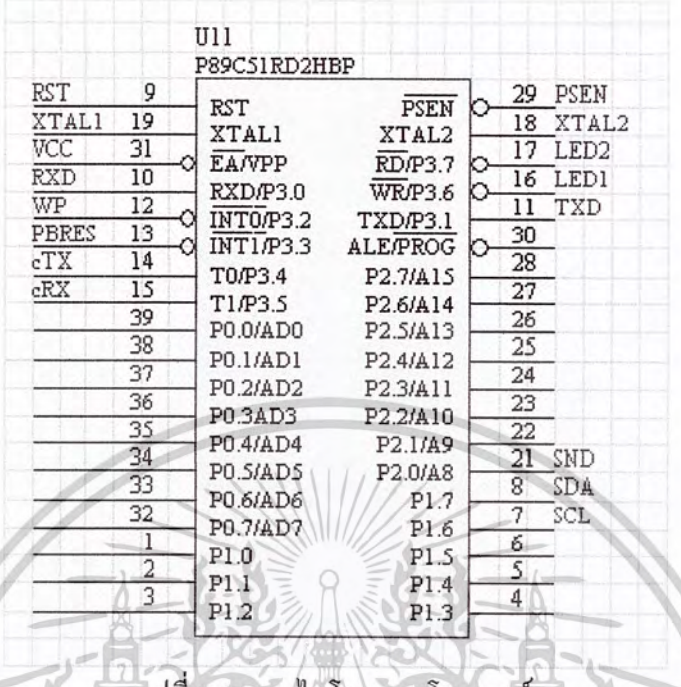


รูปที่ 3.3 วงจรรีเซตไมโครคอนโทลเลอร์

จากรูปที่ 3.3 วงจรรีเซตไมโครคอนโทลเลอร์จากไอซี DS1833 ซึ่งจะทำหน้าที่สั่งให้ไมโครคอนโทลเลอร์รีเซตทุกครั้งที่มีการเปิดเครื่องและทุกครั้งที่แรงดันจากแหล่งจ่ายไฟเปลี่ยนแปลงไป 10% ช่วงเวลาการรีเซทเป็นช่วงเวลาที่แน่นอนและเหมาะสม (350mS) ทำให้ไมโครคอนโทลเลอร์ริเซตได้อย่างมีประสิทธิภาพ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4 วงจรไมโครคอนโทลเลอร์



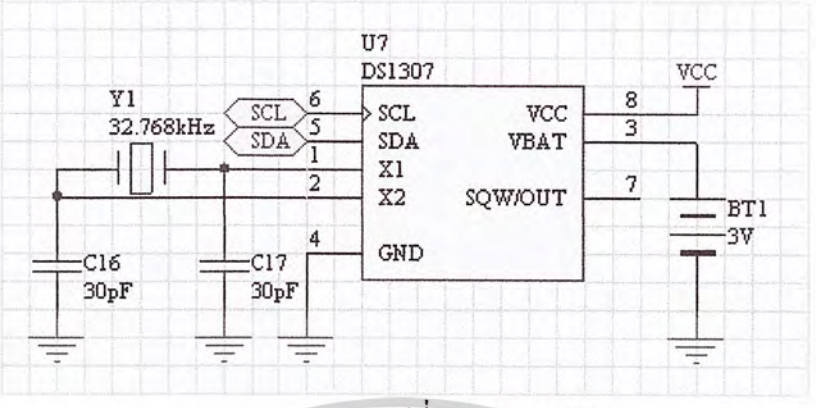
รูปที่ 3.4 วงจรไมโครคอนโทลเลอร์

จากรูปที่ 3.4 วงจรไมโครคอนโทลเลอร์เบอร์ P89C51RD2HBN มีความจุของโปรแกรม (ROM Memory size) ได้ 64 กิโลไบต์ (Kbytes) ซึ่งในการเขียนโปรแกรมนี้มีความซับซ้อนมากจำมีความจำเป็นที่จะต้องใช้ความจุโปรแกรมมากเช่นเดียวกัน และก็มีคุณสมบัติพิเศษอย่างอื่นเช่นกัน ตัวอย่างเช่น แรม 1 กิโลไบต์

ไมโครคอนโทลเลอร์จะทำหน้าที่เชื่อมต่อกับส่วนต่างของวงจรเช่น หลอดแสดงผล(LED) ลำโพงบัซเซอร์ (Buzzer) นาฬิกา (Real Time Clock) หน่วยความจำภายนอก (EEPROM) จีพีเอส และส่วนติดต่อกับคอมพิวเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

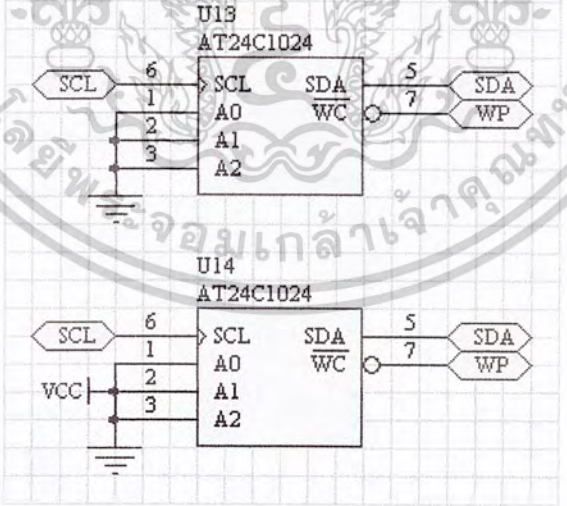
3.5 วงจรนาฬิกา(Real Time Clock)



รูปที่ 3.5 วงจรนาฬิกา

จากรูปที่ 3.5 วงจรนาฬิกาที่ติดอยู่กับเครื่องบันทึกการเดินทางโดยจะมีถ่าน(Battery) ค่อยจ่ายไฟเลี้ยงอยู่ตลอดเวลา เพราะฉะนั้นนาฬิกาก็จะเดินอยู่ตลอดเวลาไม่ว่าจะเปิดหรือปิดเครื่องก็ตาม นาฬิกานี้จะอาศัยการทำงานของไอซี DS1307 ซึ่งจะมีของมูล ปี, เดือน, วันที่, วัน, ชั่งโมง, นาทีและวินาที ซึ่งจัดรูปแบบคล้ายปฏิทินเป็นเวลานาน 10 ปี ข้อมูลดังกล่าวสามารถอ่านหรือเขียนได้โดยผ่านรูปแบบการสื่อสารข้อมูลแบบ I^{2}C

3.6 วงจรหน่วยความจำภายนอก (EEPROM)



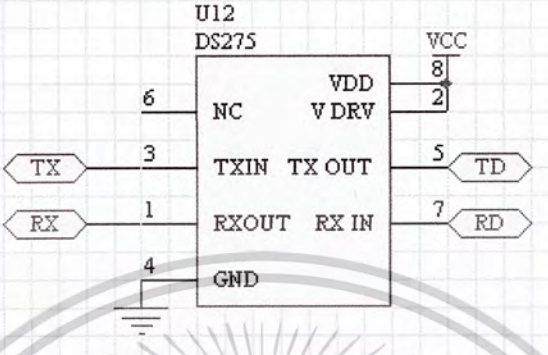
รูปที่ 3.6 วงจรหน่วยความจำภายนอก

เป็นวงจรที่เอาไว้เก็บข้อมูลการเดินทางซึ่งเป็นค่าเส้นรุ้ง เส้นแวง เวลา ที่เป็นข้อมูลพื้นฐาน และข้อมูลต่างที่คาดว่าจะเพิ่มขึ้นในอนาคตไม่ว่าจะเป็นความเร็ว การเข้าโค้ง การหยุด การสตาร์ทรถ เป็นต้น ข้อมูลสามารถเก็บไว้ในไอซี AT24C1024 สามารถเก็บข้อมูลได้ถึง 1 เมกกะไบต์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

(MBytes) โดยสามารถต่อโอซีร่วมกันได้สูงสุด 2 ตัวและสามารถเก็บข้อมูลได้ถึง 2 เมกกะไบต์ ข้อมูลดังกล่าวสามารถอ่านหรือเขียนได้โดยผ่านรูปแบบการสื่อสารข้อมูลแบบ I^2C

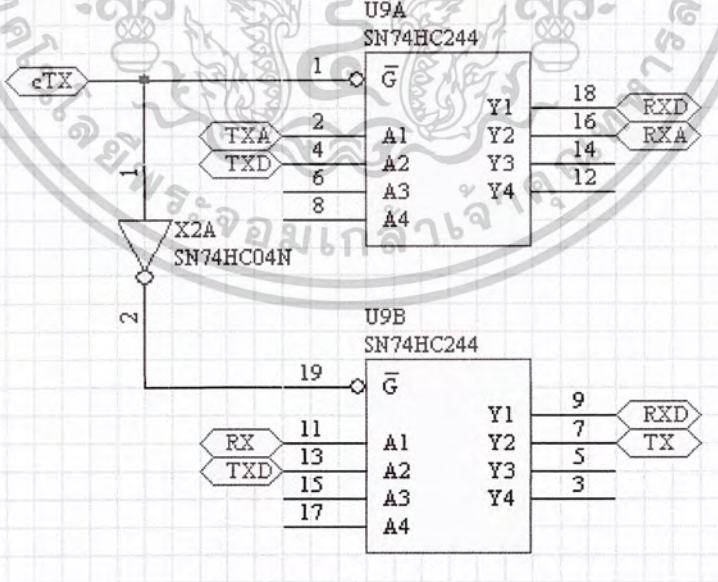
3.7 วงจรเชื่อมต่อกับคอมพิวเตอร์(Computer Interface)



รูปที่ 3.7 วงจรเชื่อมต่อกับคอมพิวเตอร์

รูปที่ 3.7 วงจรเชื่อมต่อกับคอมพิวเตอร์เป็นวงจรที่เปลี่ยนรูปแบบการสื่อสารของมูลแบบอนุกรมของคอมพิวเตอร์(RS-232) เป็นรูปแบบการสื่อสารของมูลแบบอนุกรมของไมโครคอนโทลเลอร์ (UART) โดยไอซี DS275 ที่คอยจะรูปแบบและแรงดันให้เหมาะสม

3.8 วงจรสลับการเชื่อมต่อกับ (Serial Exchange)

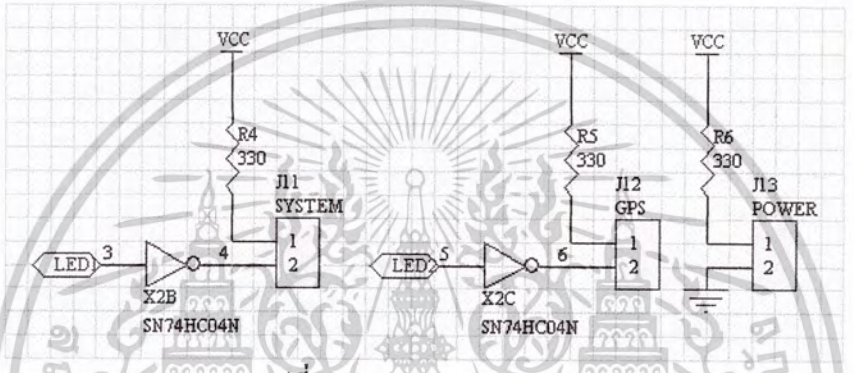


รูปที่ 3.8 วงจรสลับการเชื่อมต่อ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เป็นวงจรสลับการเชื่อมต่อกับระหว่างคอมพิวเตอร์และจีพีเอส เนื่องจากจะ MCS-51 มีการขัดจังหวะ (Interrupt) แบบการสื่อสารแบบอนุกรม (Serial Port) อยู่เพียง 1 ชุด แต่จะจำเป็นต้องคอยสื่อสารกับคอมพิวเตอร์และจีพีเอส ทำให้ ขาอนุกรมไม่พอจึงต้องมีการสลับการทำงานแบบคนละเวลา (Time Multiplex) โดยใช้ไอซี 74244 เพื่อมาช่วยในการสลับทางเดินข้อมูล (Bus Transceiver) โดยจะให้คอมพิวเตอร์มีความสำคัญมากกว่า คือเมื่อเริ่มเปิดเครื่องก็จะสามารถติดต่อกับคอมพิวเตอร์ได้ก่อนเพื่อที่จะทำการ แก้ไขข้อมูล อับเดทโปรแกรมได้

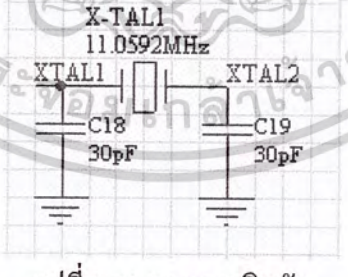
3.9 วงจรหลอดแสดงผล(LED Display)



รูปที่ 3.9 วงจรหลอดแสดงผล

ซึ่งเป็นการแสดงผลของสถานะแต่ๆ ไม่ว่าจะเป็น การทำงานของระบบ การรับสัญญาณจากดาวเทียม และไฟเลี้ยงวงจร

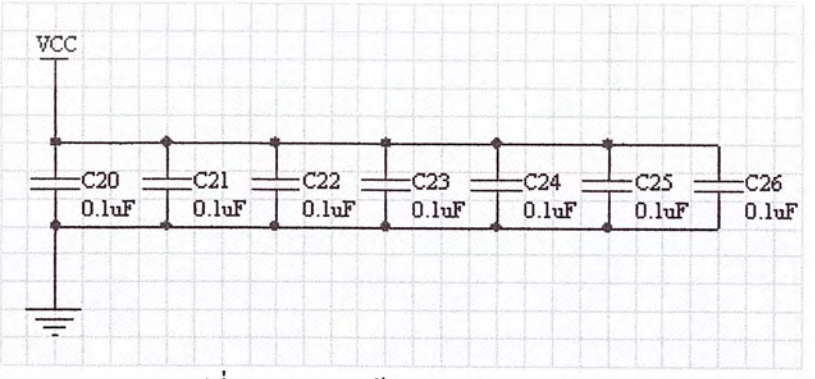
3.10 วงจรย่อยต่างๆ



รูปที่ 3.10.1 วงจรคริสตัล

วงจรคริสตัลเป็นวงจรกำเนิดสัญญาณนาฬิกาให้จังหวะในการทำงานแก่ไมโครคอนโทลเลอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

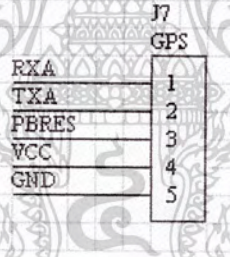


รูปที่ 3.10.2 วงจรป้อนกับสัญญาณรบกวน

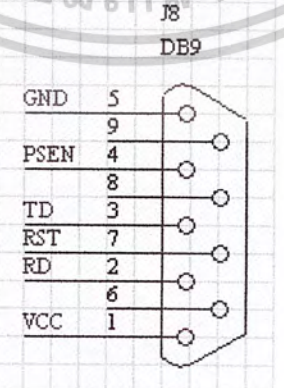
วงจรป้อนกับสัญญาณรบกวนเป็นวงจรที่ตัดสัญญาณรบกวนอันเนื่องมาจากการทำงานของไอซีดิจิตอล ที่จะไปรบกวนไฟของระบบลงกราวนด์ โดยการต่อตัวเก็บประจุ(Capacitor) คร่อมจุด VCC และ GND ของไอซีทุกตัว

3.11 คอนเนคเตอร์(Connector)

เป็นจุดเชื่อมต่อจากวงจรไปสู่อุปกรณ์อื่นๆภายนอก

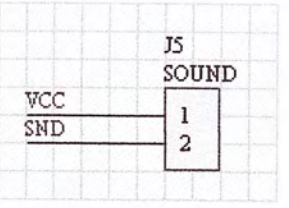


รูปที่ 3.11.1 จีพีเอสคอนเนคเตอร์



รูปที่ 3.11.2 DB9 คอนเนคเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

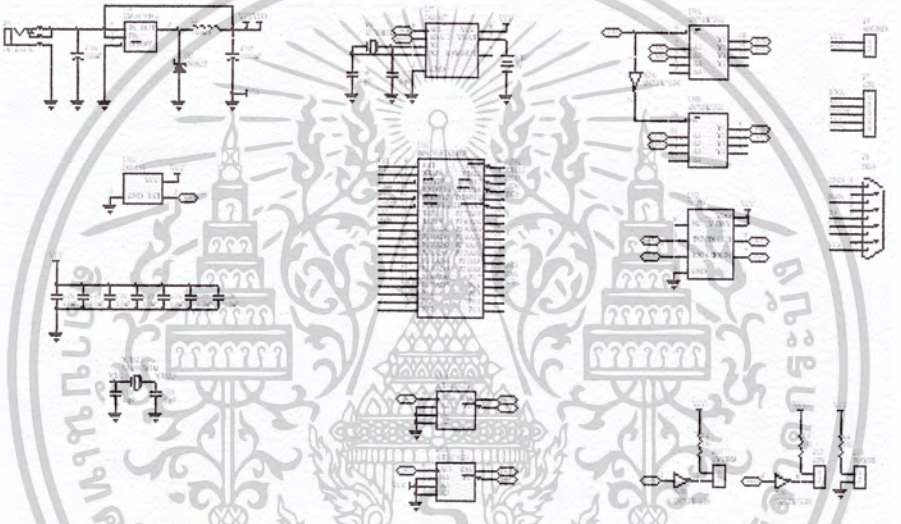


รูปที่ 3.11.3 คอนเนคเตอร์ลำโพงบัซเซอร์

3.12 การออกลายทองแดง

การออกลายทองแดงโดยซอฟท์แวร์โพลเทล99 ทำได้โดย

1 การเตรียมเอกสารวงจรที่สมบรูณ์แล้ว

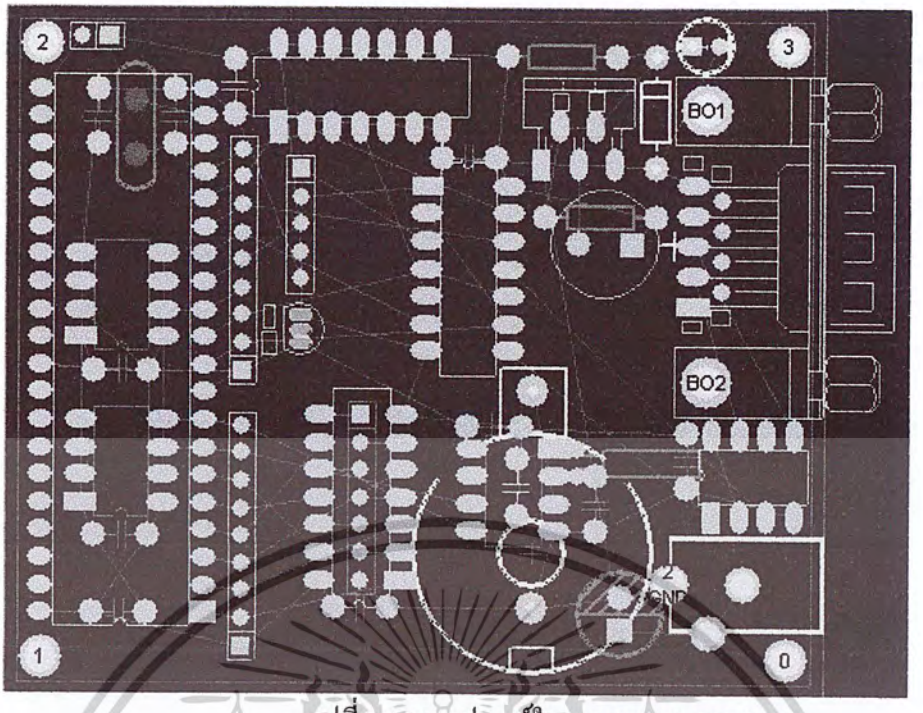


รูปที่ 3.12.1 เอกสารวงจรที่สมบรูณ์

2 การแปลงวงจรเป็นลายทองแดง

การแปลงวงจรเป็นลายทองแดงทำได้โดยใช้คำสั่ง Update PCB ก็จะมีรูปตัวอุปกรณ์ปรากฏที่เอกสารลายทองแดงและผูกโยงกันด้วยเส้นตามวงจร จากนั้นจัดเรียงอุปกรณ์บนแผ่นลายทองแดง

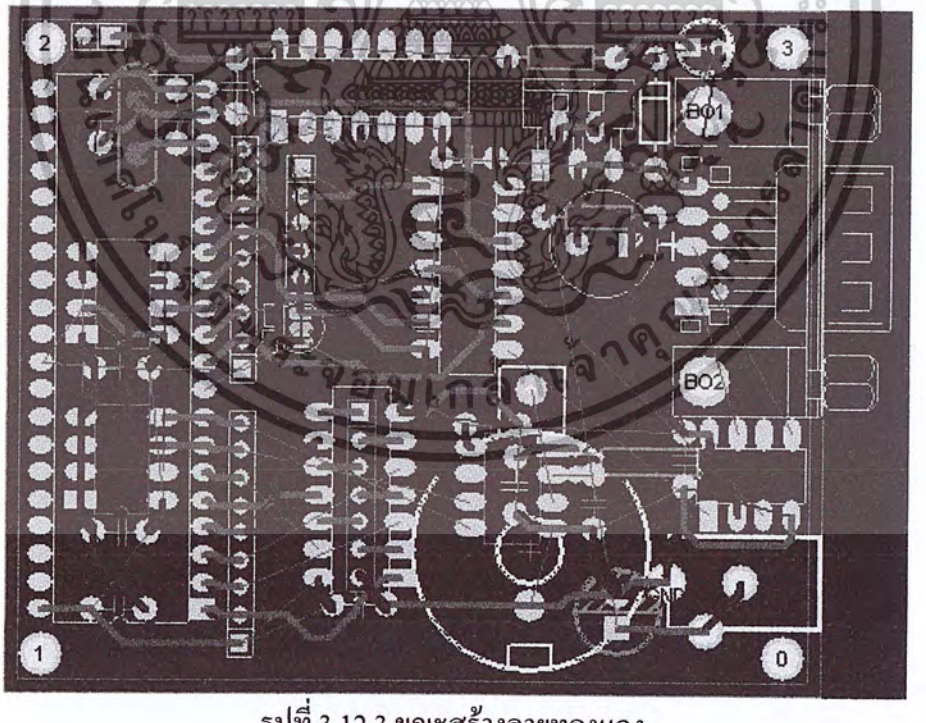
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.12.2 อุปกรณ์ในเอกสารลายทองแดง

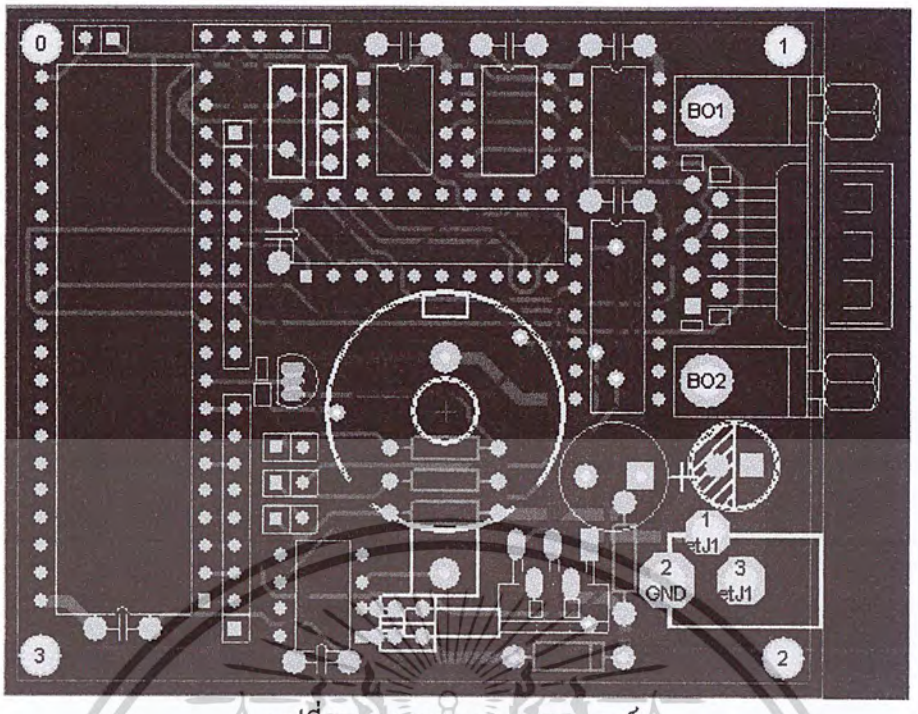
3 สร้างลายทองแดง

สร้างลายทองแดงอัตโนมัติโดยคำสั่ง Auto Route

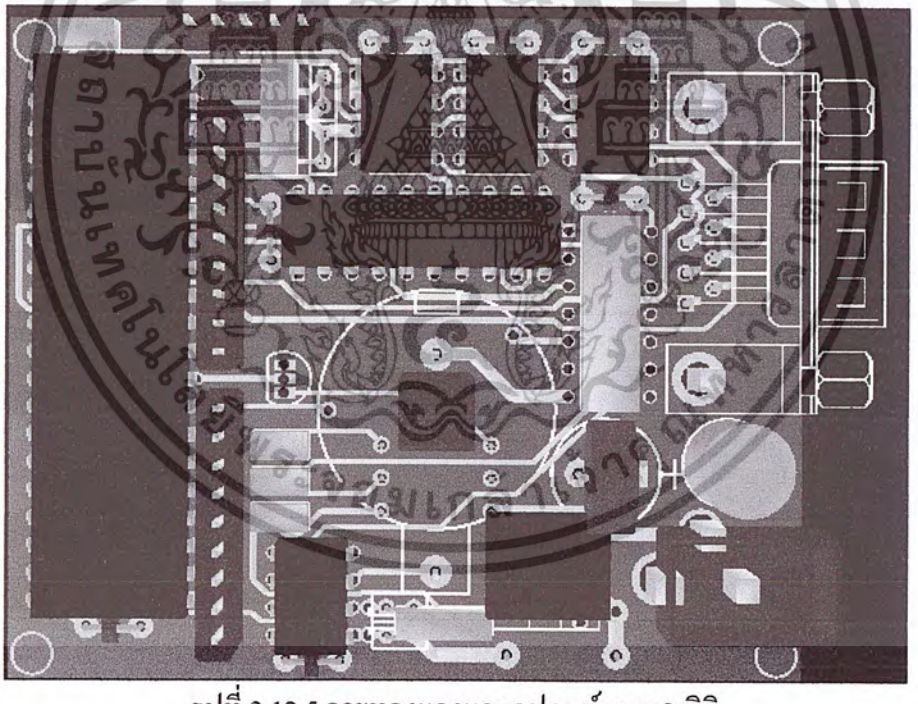


รูปที่ 3.12.3 ขณะสร้างลายทองแดง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.12.4 ลายทองแดงสมบูรณ์

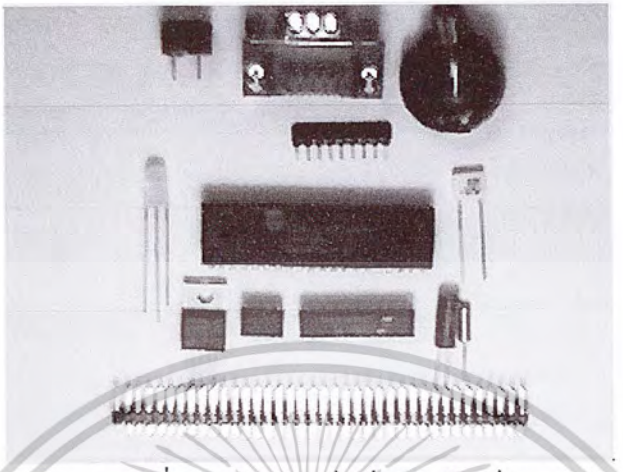


รูปที่ 3.12.5 ลายทองแดงและอุปกรณ์แบบ 3 มิติ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.13 การประกอบวงจร

การจัดเตรียม อุปกรณ์อิเล็กทรอนิกส์ม, อุปกรณ์จีพีเอส, เครื่องมือการสร้าง



รูปที่ 3.13.1อุปกรณ์อิเล็กทรอนิกส์

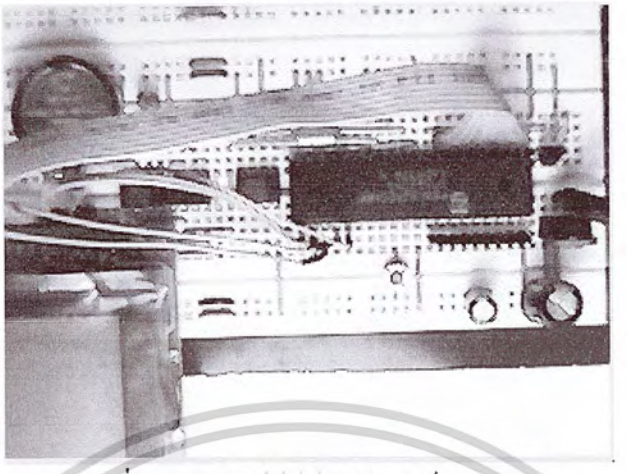
รูปที่ 3.13.2 อุปกรณ์รับสัญญาณจีพีเอส

รูปที่ 3.13.3 เครื่องมือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.14 ทดสอบวงจรกับบอร์ดทดลอง

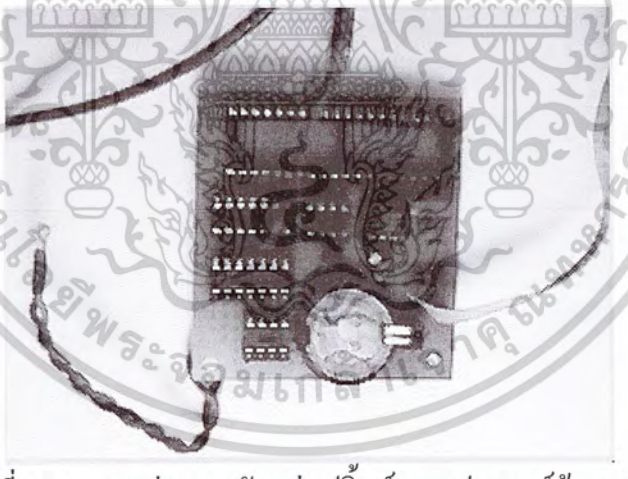
ทำการต่อวงจรกับบอร์ดทดลองเพื่อตรวจสอบวงจร



รูปที่ 3.14.1 การต่อวงจรกับบอร์ดทดลอง

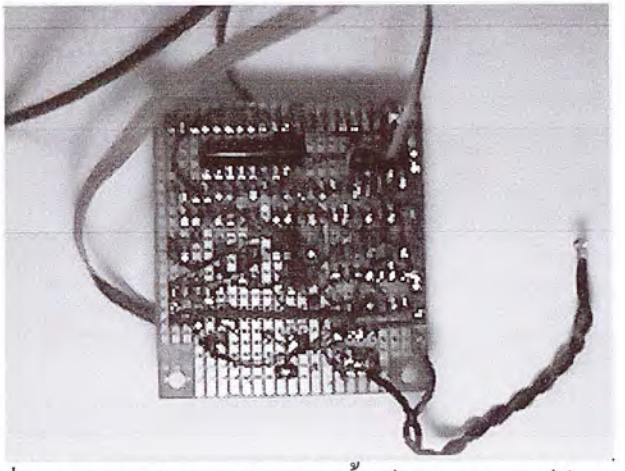
3.15 ทดสอบวงจรกับแผ่นปริ้นส์เอนกประสงค์

ทำการต่อวงจรกับแผ่นปริ้นส์เอนกประสงค์เพื่อให้แน่ใจว่าวงจรสามารถทำงานได้ในลายทองแดงจริง



รูปที่ 3.15.1 การต่อวงจรกับแผ่นปริ้นส์เอนกประสงค์ด้านหน้า

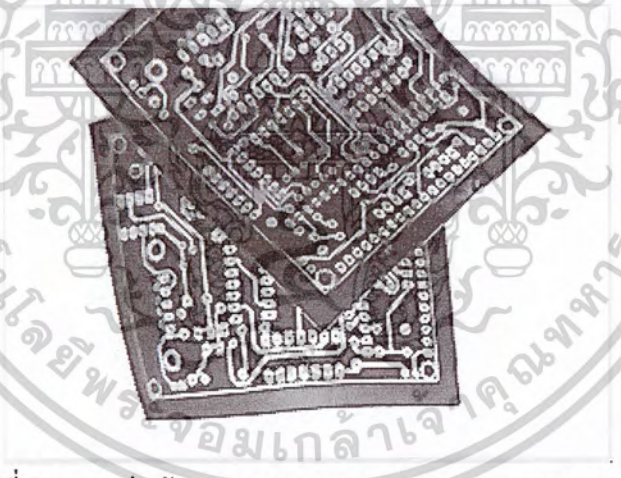
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.15.2 การต่อวงจรกับแผ่นปริ้นส์เอนกประสงค์ด้านหลัง

3.16 ทดสอบวงจรกับแผ่นปริ้นส์ลายทองแดง

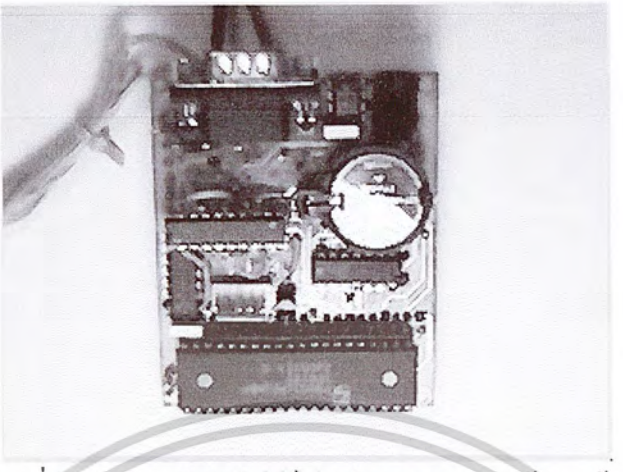
เมื่อทดสอบประสิทธิภาพในบอร์ดทดลอง แผ่นปริ้นส์อเนกประสงค์แล้วก็มาถึงขั้นตอนการทำแผ่นลายทองแดงตามให้ได้ออกแบบมาก่อนหน้านี้แล้ว โดยการนำเอกสารลายทองแดงของซอฟท์แวร์โพลเทล มาสู่ขั้นตอนการผลิตลายทองแดง



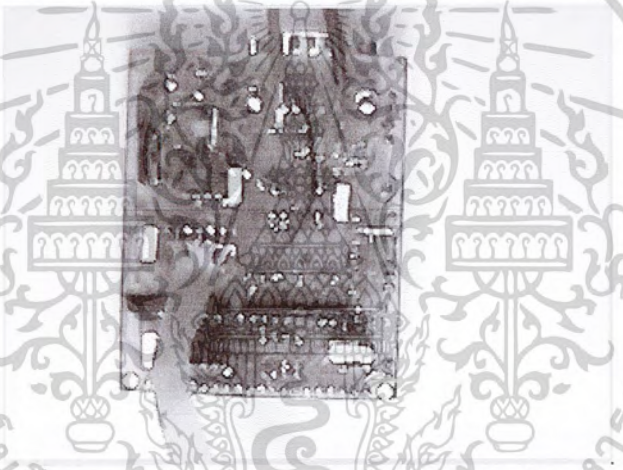
รูปที่ 3.16.1 แผ่นคัดลอกลายทองแดงจากเอกสารลายทองแดง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากนั้นทำการต่ออุปกรณ์ลงแผ่นลายทองแดง แล้วทดสอบการทำงานเบื้องต้น



รูปที่ 3.16.2 การต่ออุปกรณ์ลงแผ่นลายทองแดงด้านหน้า

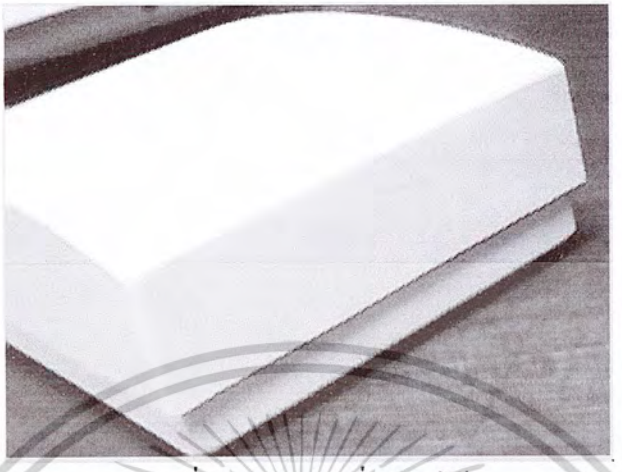


รูปที่ 3.16.3 การต่ออุปกรณ์ลงแผ่นลายทองแดงด้านหลัง

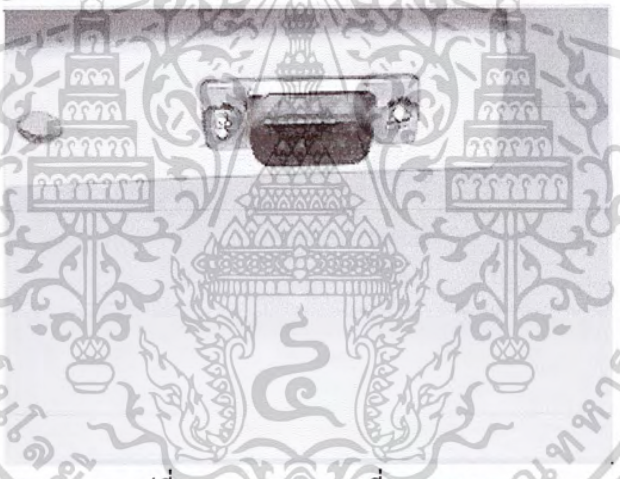
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.17 การประกอบวงจรลงกล่อง

จะเริ่มด้วยการเตรียมอุปกรณ์ที่จำเป็นต่างๆ พร้อมด้วยกล่องที่ทำการเจาะรูให้เหมาะสม จากนั้นทำการประกอบส่วนอุปกรณ์ภายนอกต่อเข้ากับวงจร สุดท้ายบรรจุไว้ในกล่องสวยงาม



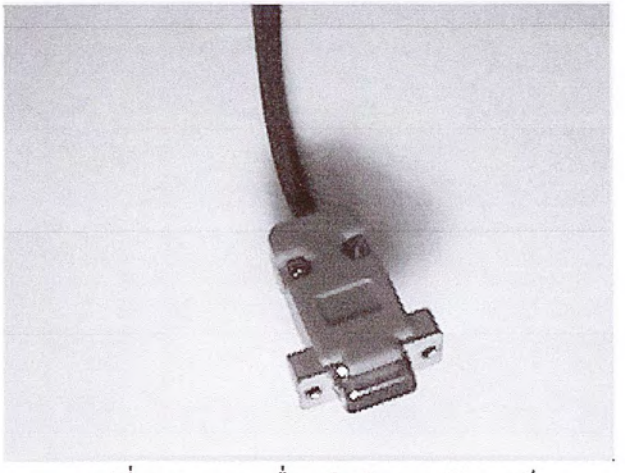
รูปที่ 3.17.1 กล่องที่เตรียมไว้



รูปที่ 3.17.2 การเจาะรูที่เหมาะสม

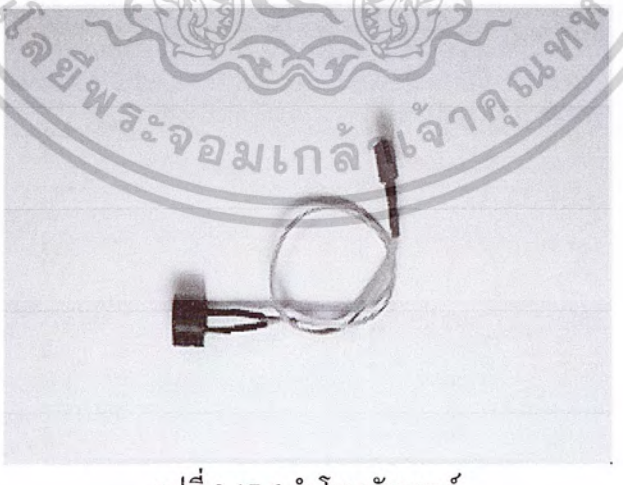
รูปที่ 3.17.3 สายอากาศจีพีเอส

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.17.4 สายเชื่อมต่อกับคอมพิวเตอร์

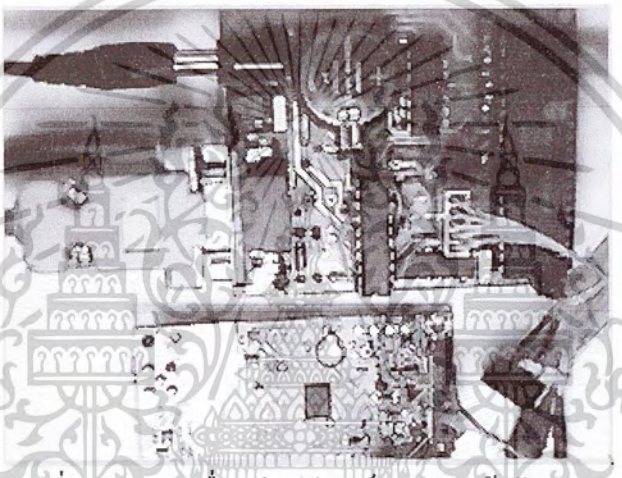
รูปที่ 3.17.5 หลอดแสดงสถานะ



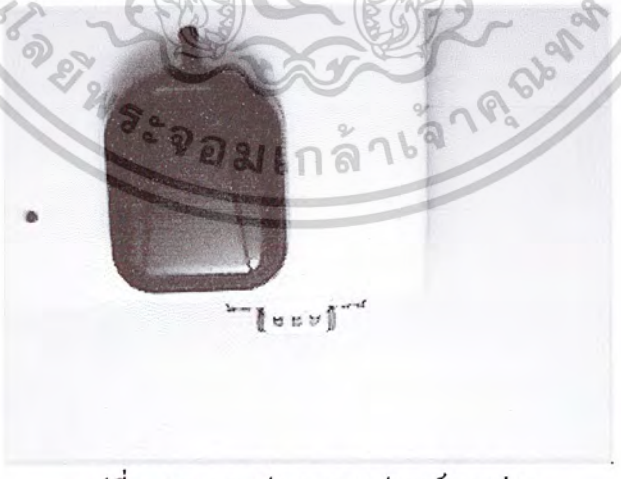
รูปที่ 3.17.6 ลำโพงบัซเซอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 3.17.7 ปลั๊กไฟในรถยนต์



รูปที่ 3.17.8 การเชื่อมต่ออุปกรณ์ภายนอกเข้ากับวงจร



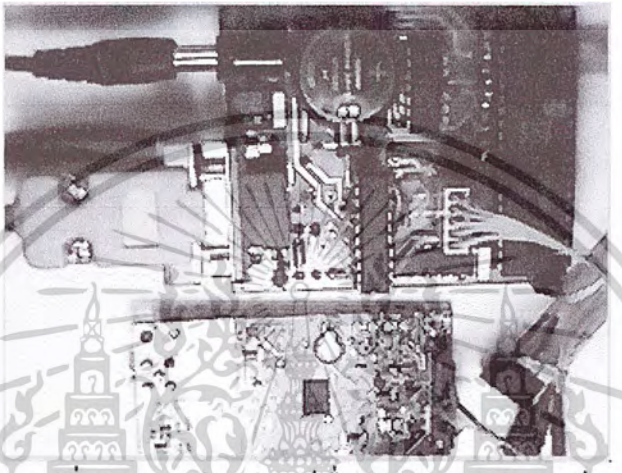
รูปที่ 3.17.9 การประกอบอุปกรณ์ลงกล่อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4
การทดลองและผลการทดลอง

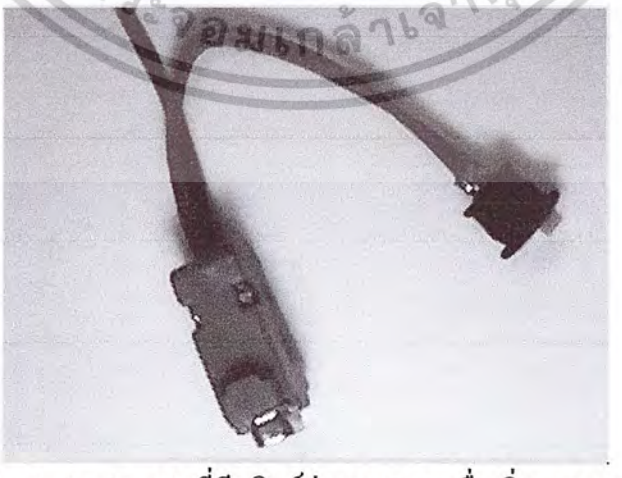
4.1 การทดลองและผลการทดลองของการโปรแกรมไมโครคอนโทลเลอร์

เป็นการทดลองทำการโปรแกรมไมโครคอนโทลเลอร์โดยใช้ซอฟท์แวร์แฟลชเมจิก(Flash Magic) ทำงานร่วมกับไมโครคอนโทลเลอร์เบอร์ P89C51RD2HBP เริ่มแรกต้องต่ออุปกรณ์เชื่อมต่อกับคอมพิวเตอร์



รูปที่ 4.1.1 การต่ออุปกรณ์เชื่อมต่อกับคอมพิวเตอร์

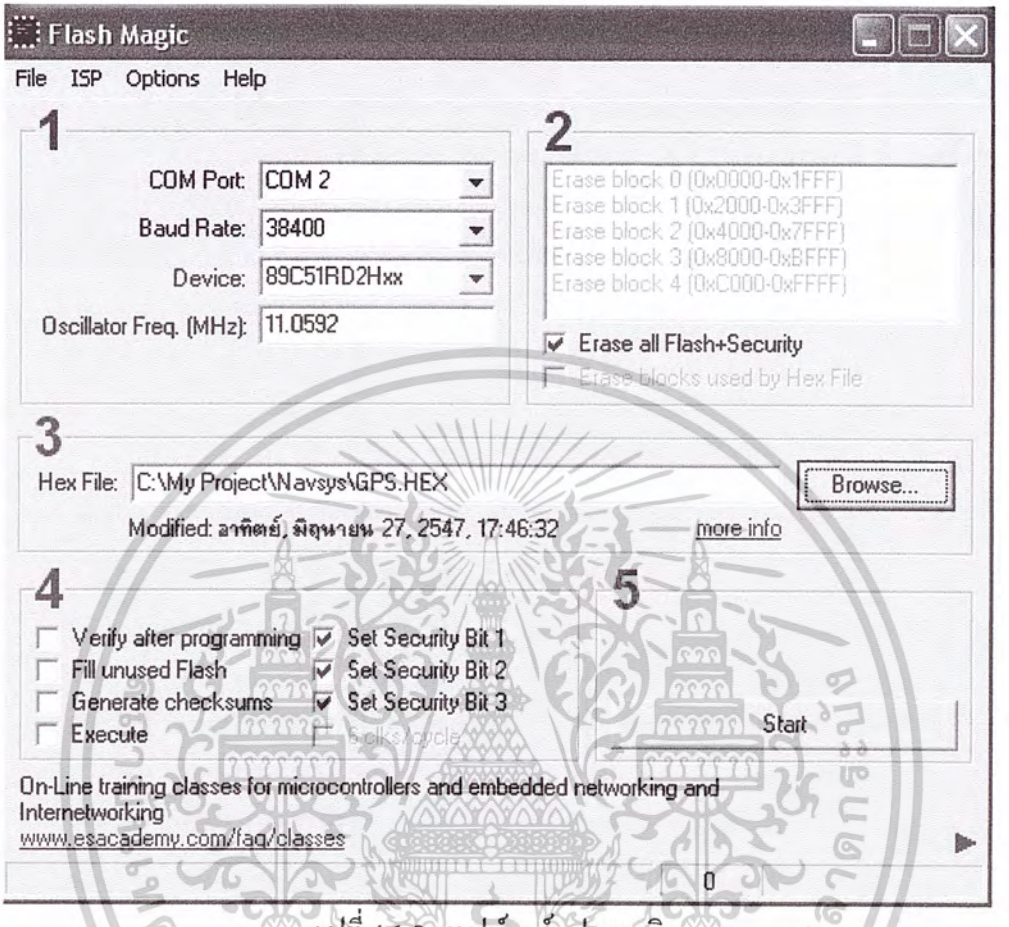
การทำงานของซอฟท์แวร์แฟลชเมจิก(Flash Magic) ทำงานร่วมกับไมโครคอนโทลเลอร์เบอร์ P89C51RD2HBP จะสามารถโปรแกรมได้ทันทีผ่านสาย RS-232 โดยการต่อขา PSEN ของไมโครคอนโทลเลอร์ลงกราวนด์ก่อน จากนั้นต่อไฟเลี้ยงให้กับวงจร ขณะนี้ไมโครคอนโทลเลอร์ได้เข้าสู่โหมดโปรแกรมแล้ว



รูปที่ 4.1.2 สาย RS-232 ที่ทีสวิตซ์ต่อขา PSEN เพื่อเพิ่มความสะดวก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

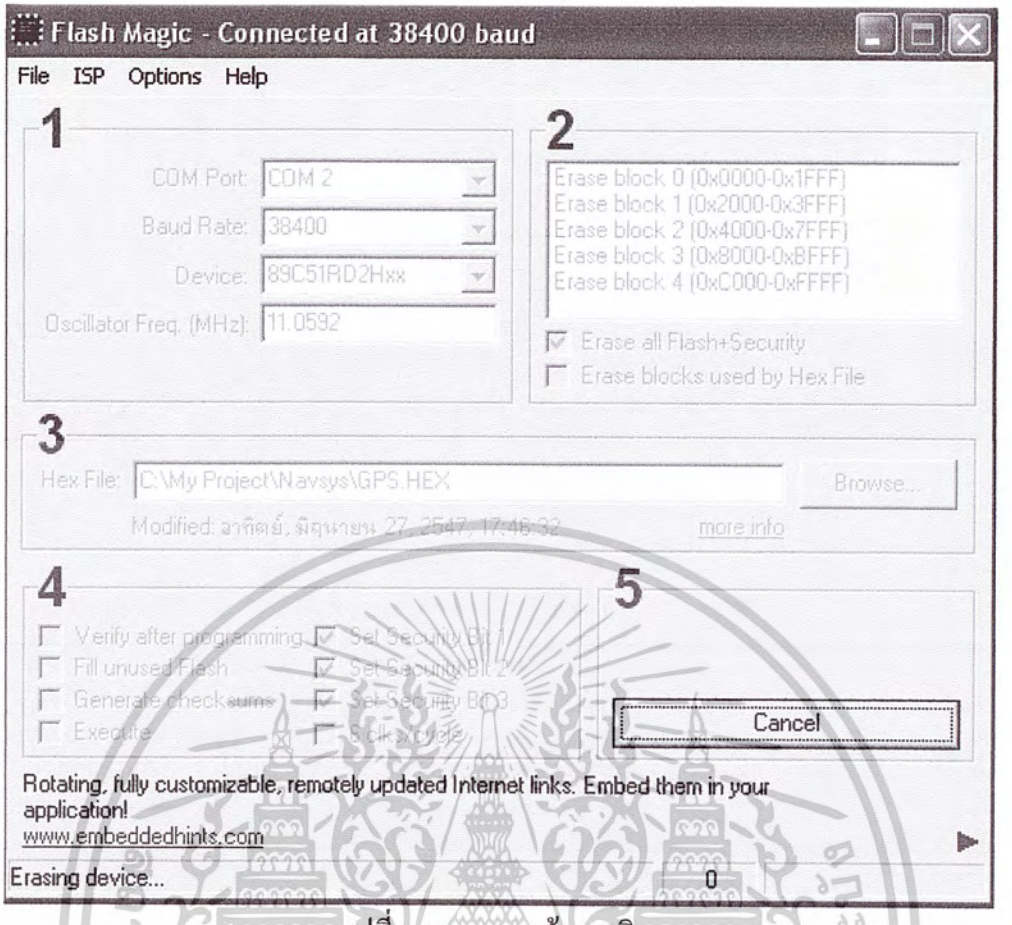
จากนั้นเริ่มใช้ซอฟท์แวร์แฟลชเมจิก จากนั้นทำการเซทค่าต่างไม่ว่าจะเป็นพอร์ต เบอร์ไมโครคอนโทลเลอร์ ความถี่คริสตัล แฟ้มข้อมูลที่ต้องการจะโปรแกรม เป็นต้น



รูปที่ 4.1.3 ซอฟท์แวร์แฟลชเมจิก

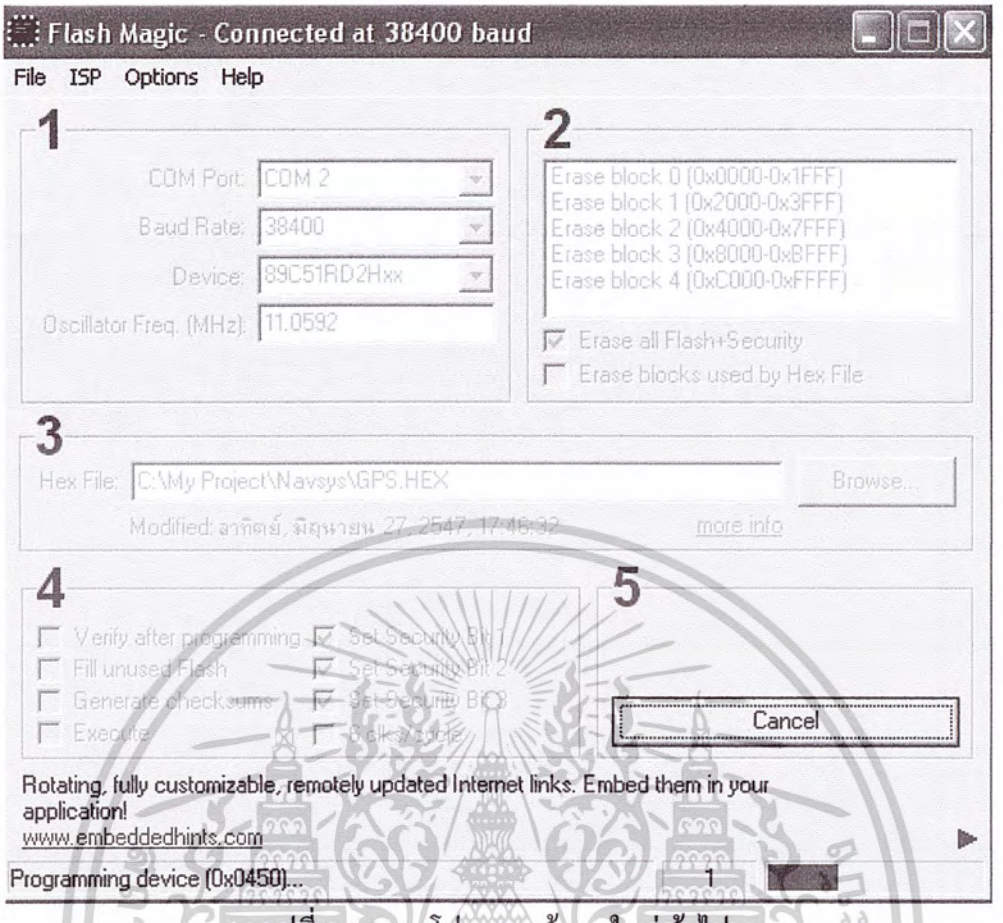
จากนั้นเลือก Start ซอฟท์แวร์แฟลชเมจิกจะทำการลบข้อมูลเดิม ตามด้วยการโปรแกรมข้อมูลใหม่เข้าไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



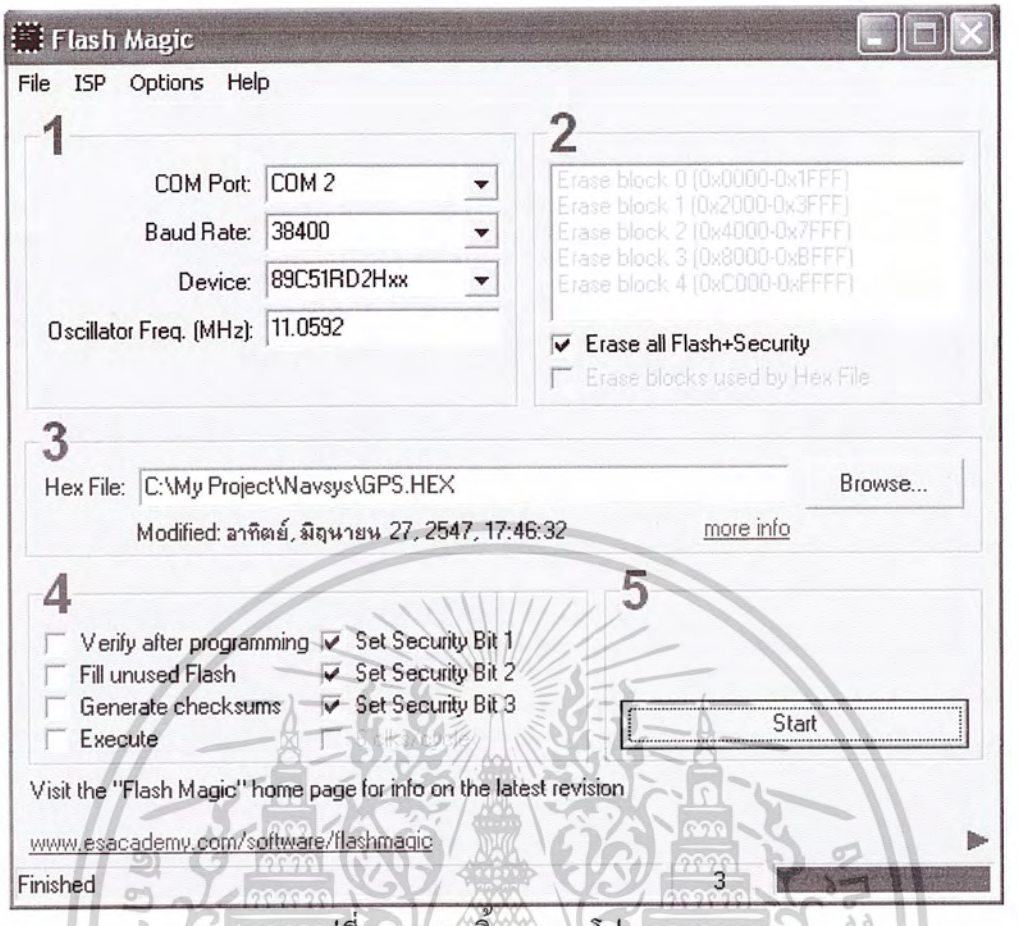
รูปที่ 4.1.4 การลบข้อมูลเดิม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.1.5 การโปรแกรมข้อมูลใหม่เข้าไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

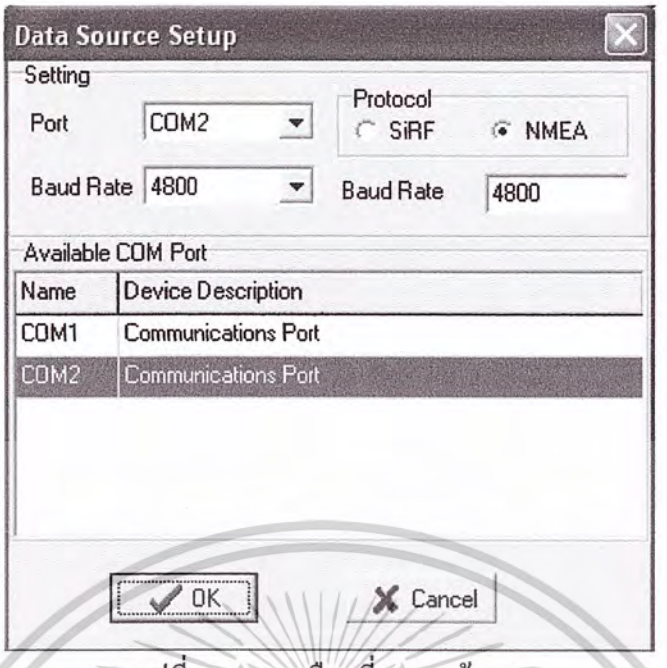


รูปที่ 4.1.6 การสิ้นสุดการโปรแกรม

4.2 การวิเคราะห์สัญญาณจีพีเอส

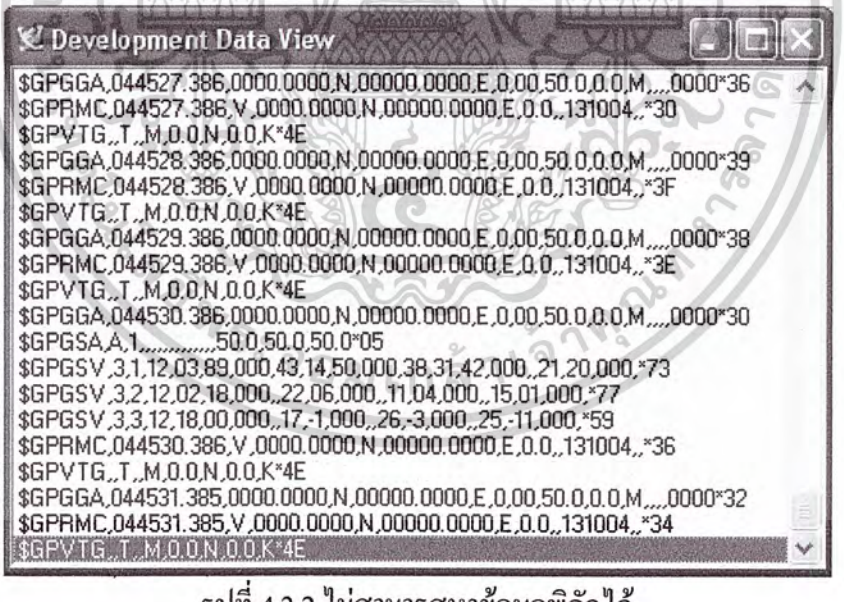
เป็นการวิเคราะห์สัญญาณจีพีเอสด้วยซอฟท์แวร์จีมอนิเตอร์(GMonitor) เริ่มจากการเปิดใช้ซอฟท์แวร์จีมอนิเตอร์จากนั้นต่อจีพีเอสเข้ากับคอมพิวเตอร์ทำการเลือกที่มาของข้อมูล (Data Source) แล้วเลือกค่าต่างๆให้ตรงกับที่ต่อไว้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.2.1 การเลือกที่มาของข้อมูล

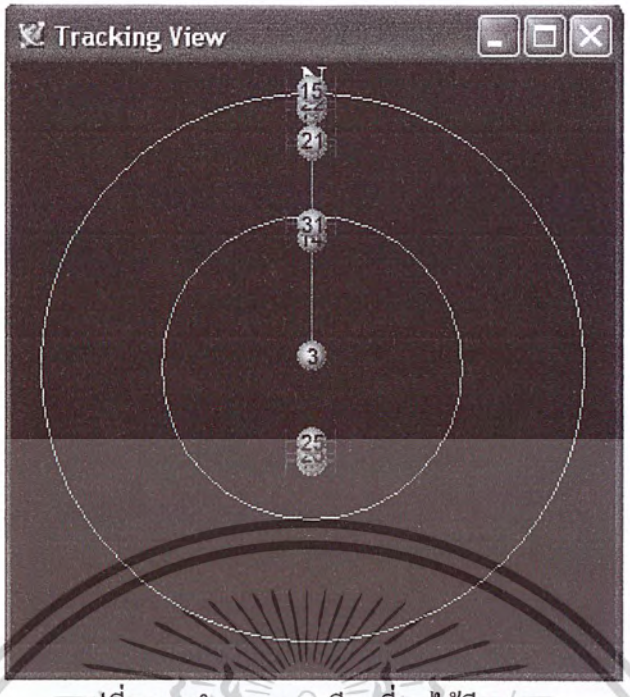
เมื่อเริ่มเปิดเครื่องจีพีเอสยังไม่สามารถตรวจสอบดาวเทียมพบจึงต้องใช้เวลาในการค้นหาดาวเทียม จึงไม่สามารสหาพิกัดได้



รูปที่ 4.2.2 ไม่สามารสหาข้อมูลพิกัดได้

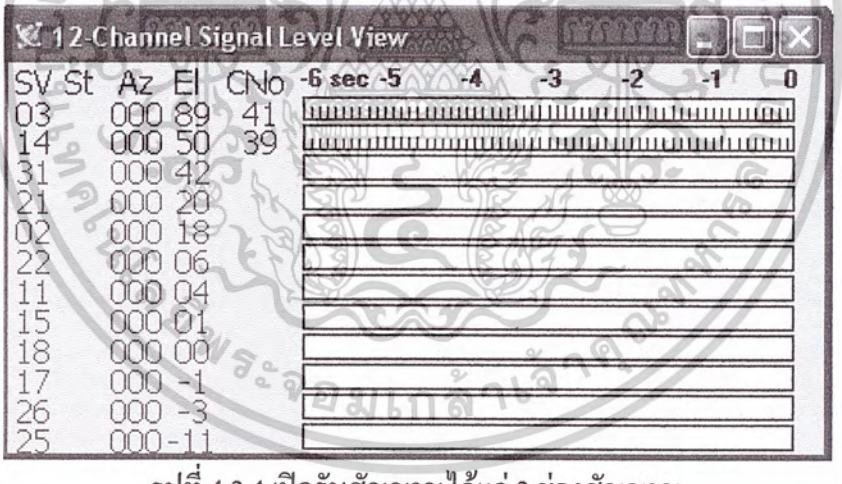
จากรูปที่ 4.2.2 ไม่สามารสหาข้อมูลพิกัดได้ จะเห็นจะเส้นรุ้งและค่าแวงจะมีค่าเท่ากับศูนย์ ขณะเดียวกันจำนวนดาวเทียมที่หาได้มี 2 ดวง คือดวงที่ 14 และ 3 ซึ่งไม่พอที่จะคำนวณหาพิกัดได้ ดังรูปที่ 4.2.3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



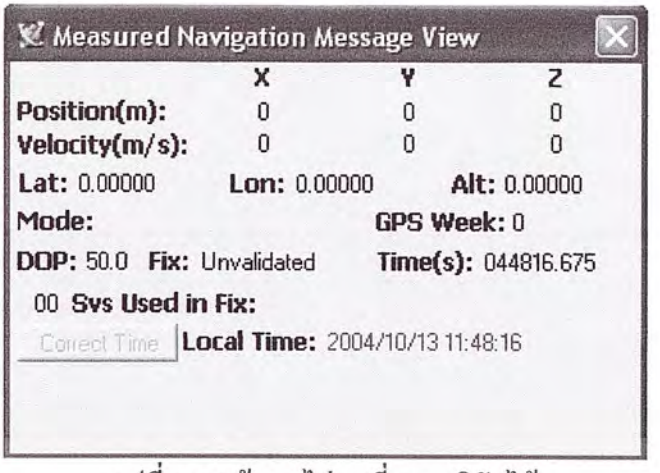
รูปที่ 4.2.3 จำนวนดาวเทียมที่หาได้มี 2 ดวง

ช่องรับสัญญาณดาวเทียมเปิดรับสัญญาณได้แค่ 2 ช่องสัญญาณ ทำให้ข้อมูลไม่พอที่จะหาพิกัดได้



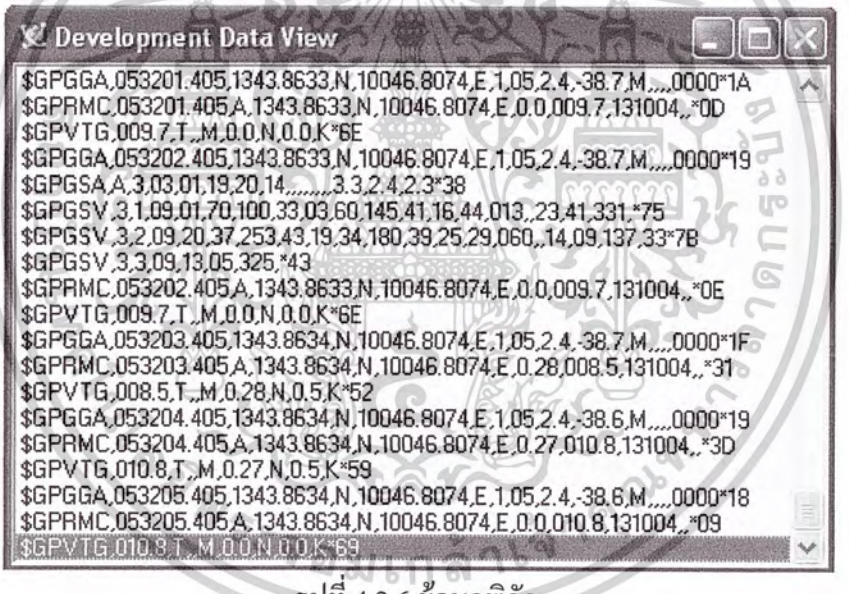
รูปที่ 4.2.4 เปิดรับสัญญาณได้แค่ 2 ช่องสัญญาณ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.2.5 ข้อมูลไม่พอที่จะหาพิกัดได้

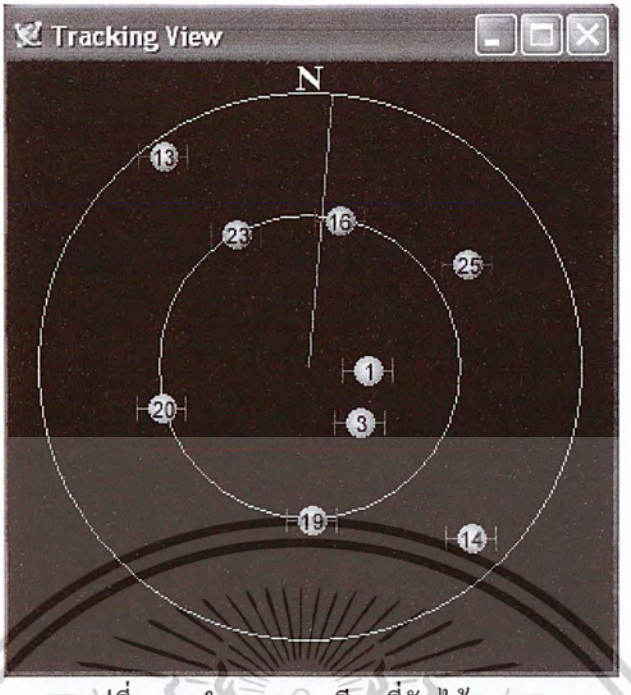
จากนั้นทำการปล่อยทิ้งไว้ประมาณ 30 นาทีเพื่อให้จีพีเอสค้นหาสัญญาณ เมื่อเวลาผ่านไปปรากฏว่าจีพีเอสสามารถ0ค้นหาสัญญาณได้โดยข้อมูลที่เข้ามามีพิกัด



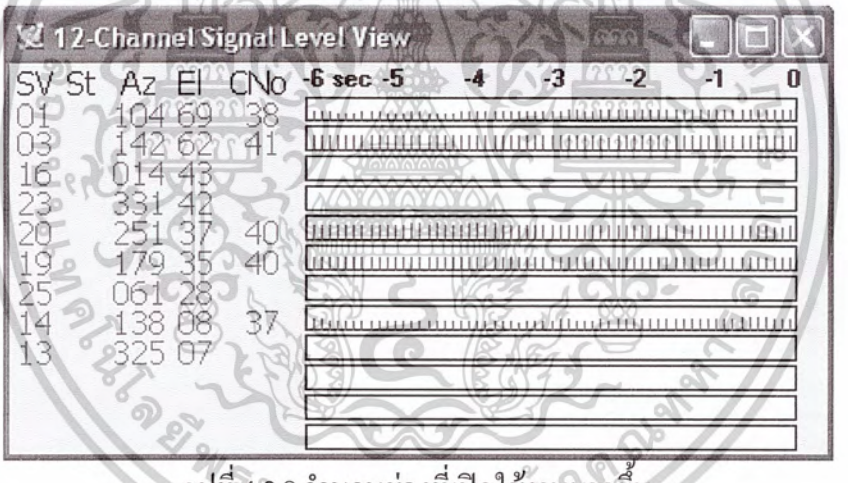
รูปที่ 4.2.6 ข้อมูลพิกัด

และจำนวนดาวเทียมที่หาได้ก็มากกว่าเดิม ทำให้ช่องการรับสัญญาณทำงานเพิ่มขึ้นด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



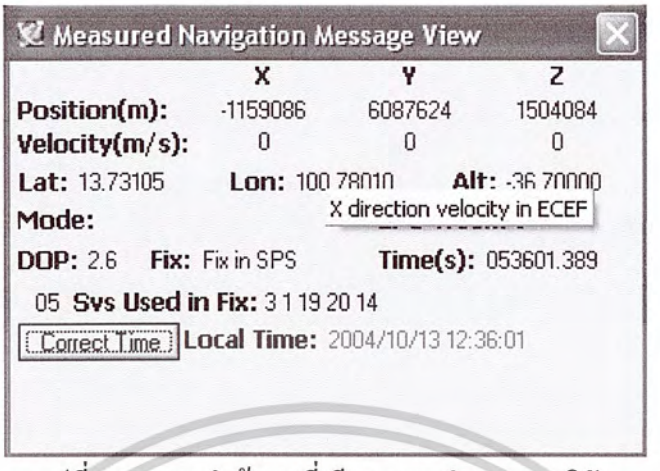
รูปที่ 4.2.7 จำนวนดาวเทียมที่รับได้ 5 ดวง



รูปที่ 4.2.8 จำนวนช่องที่เปิดใช้งานมากขึ้น

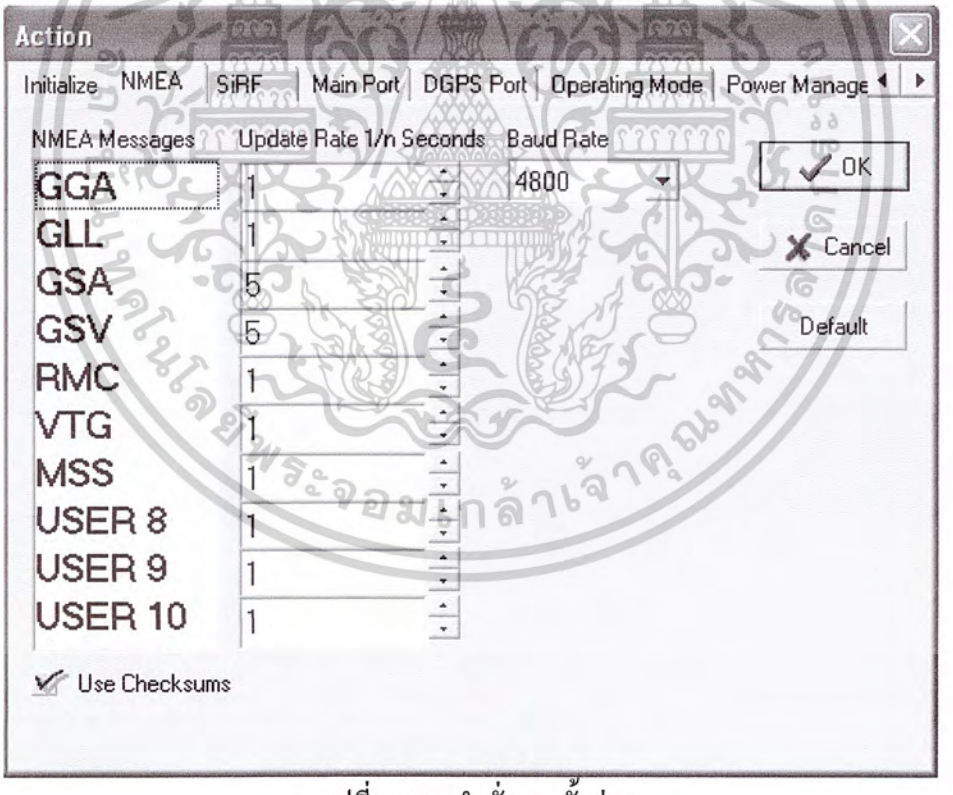
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ทำให้มีข้อมูลเพียงพอที่จะนำมาคำนวณหาพิกัดได้



รูปที่ 4.2.9 การนำข้อมูลที่เพียงพอมาคำนวณหาพิกัด

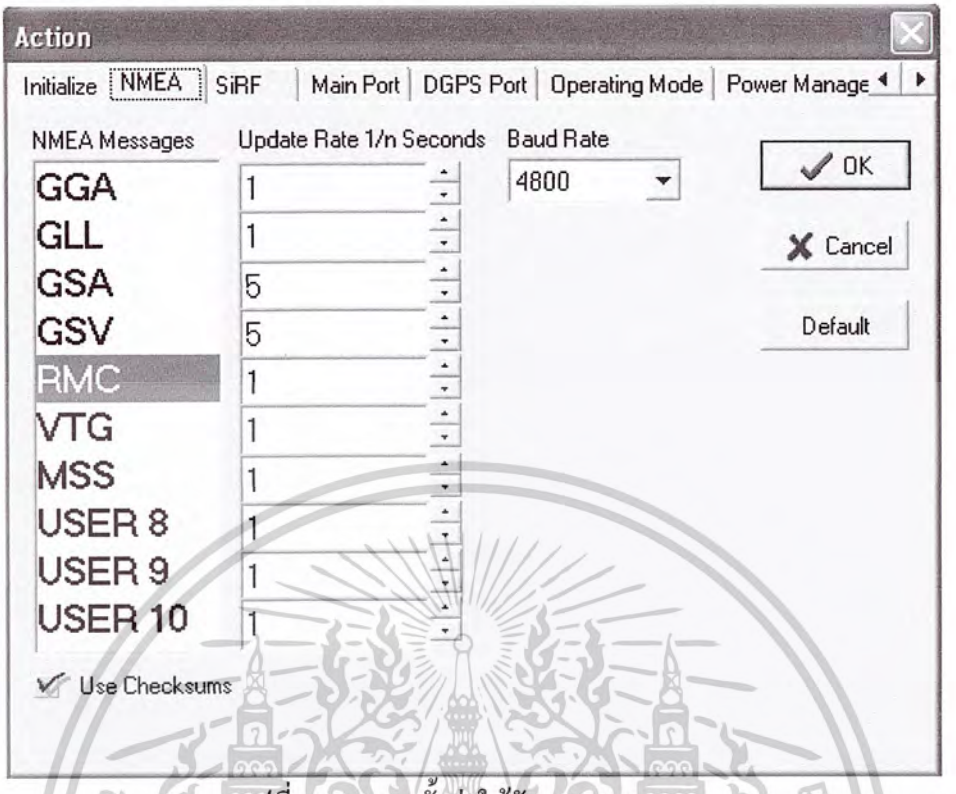
ซอฟท์แวร์จีมอนิเตอร์สามารถตั้งค่าของจีพีเอสได้



รูปที่ 4.2.10 คำสั่งการตั้งค่า

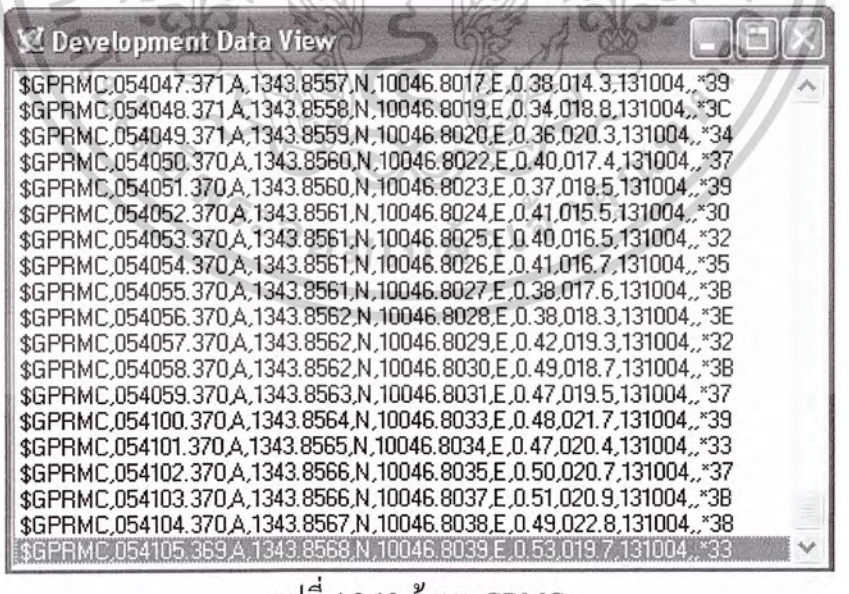
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยการทดลองการตั้งค่าให้รับเฉพาะ GRMC



รูปที่ 4.2.11 การตั้งค่าให้รับเฉพาะ GRMC

ผลที่ได้ก็คือจีพีเอสจะส่งเฉพาะข้อมูล GRMC มาเท่านั้น

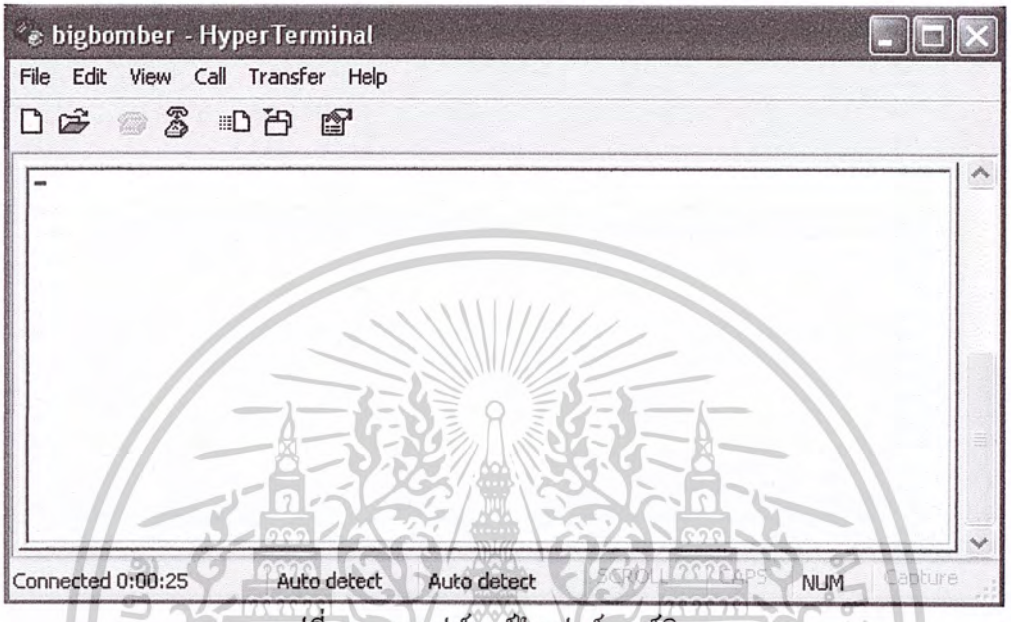


รูปที่ 4.2.12 ข้อมูล GRMC

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3 การทดลองและผลการทดลองต่อจีพีเอสกับคอมพิวเตอร์ด้วยซอฟท์แวร์ทั่วไป

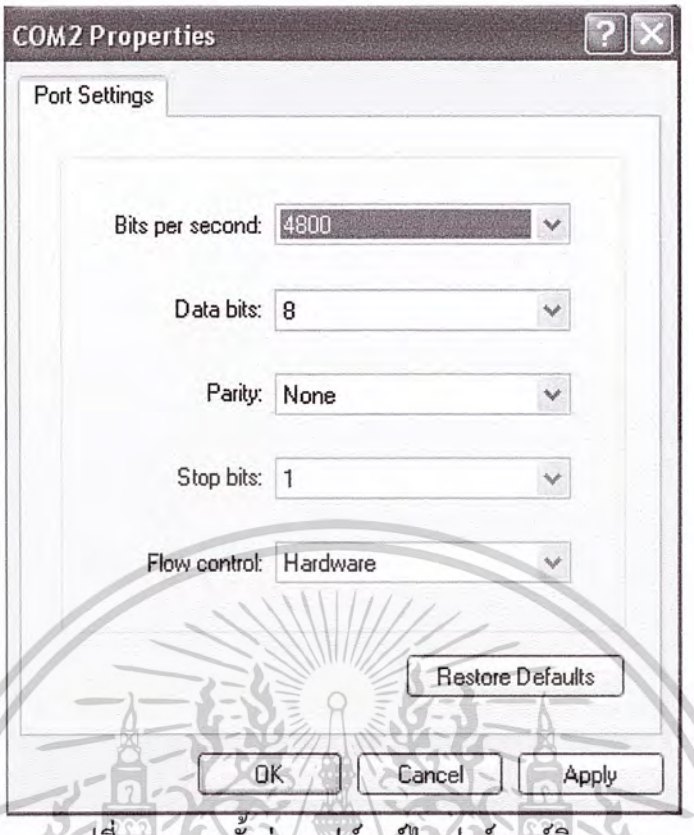
เมื่อทำการต่อจีพีเอสกับซอฟท์แวร์จีมอนิเตอร์ได้แล้ว ลองทำการต่อกับซอฟท์แวร์มาตรฐานของวินโดวส์ คือ ซอฟท์แวร์ไฮเปอร์เทอร์มินอล(Hyper Terminal) โดยการเรียกใช้ซอฟท์แวร์ไฮเปอร์เทอร์มินอล ก็จะปรากฏดังรูป 4.3.1



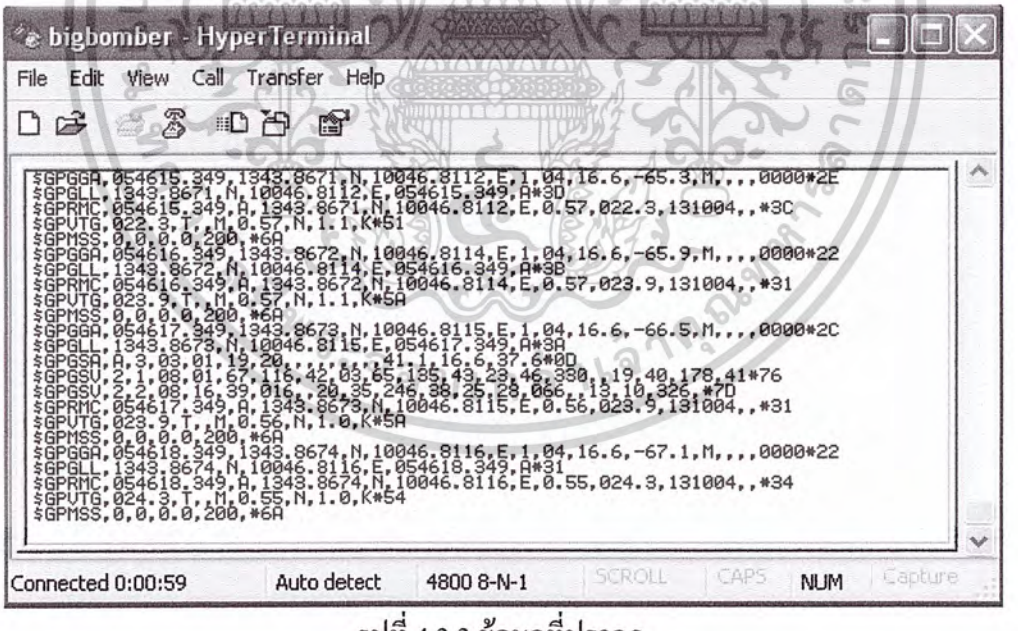
รูปที่ 4.3.1 ซอฟท์แวร์ไฮเปอร์เทอร์มินอล

จากนั้นทำการตั้งค่าตามที่ได้ต่อไว้ แล้วทำการเชื่อมต่อ(Connect) จีพีเอสเข้ากับคอมพิวเตอร์ก็จะปรากฏข้อมูลที่ซอฟท์แวร์ไฮเปอร์เทอร์มินอล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.3.2 การตั้งค่าซอฟท์แวร์ไฮเปอร์เทอร์มินอล

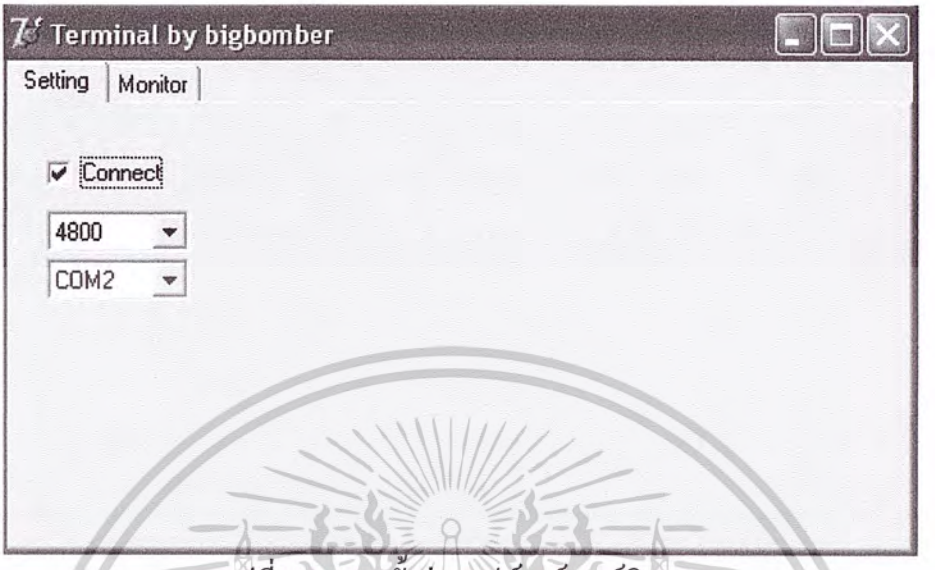


รูปที่ 4.3.3 ข้อมูลที่ปรากฏ

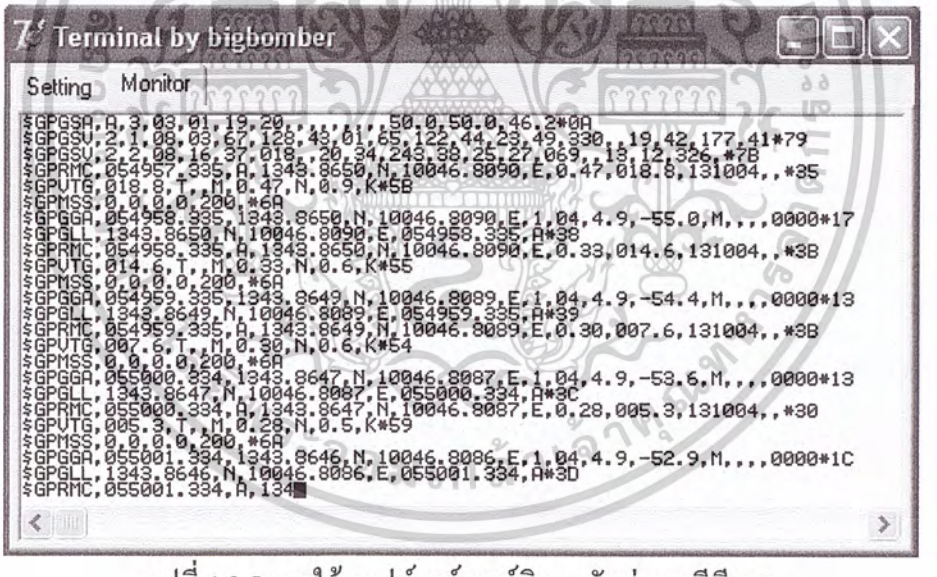
เมื่อจีพีเอสสามารถใช้ต่อร่วมกับซอฟท์แวร์ไฮเปอร์เทอร์มินอลได้แล้วจากนั้นลองเขียนซอฟท์แวร์รับค่าทางพอร์ตอนุกรมเพื่อประโยชน์ในการพัฒนาต่อไป เมื่อได้ซอฟท์แวร์ที่เขียนเองได้แล้วก็ให้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ชื่อว่า ซอฟท์แวร์เทอร์มินอล จากนั้นทำการตั้งค่าตามที่ต่อไว้ แล้วก็ลองรับข้อมูลจากจีพีเอสดู ซึ่งผลก็คือสามารสรับข้อมูลจากจีพีเอสได้



รูปที่ 4.3.4 การตั้งค่าซอฟท์แวร์เทอร์มินอล

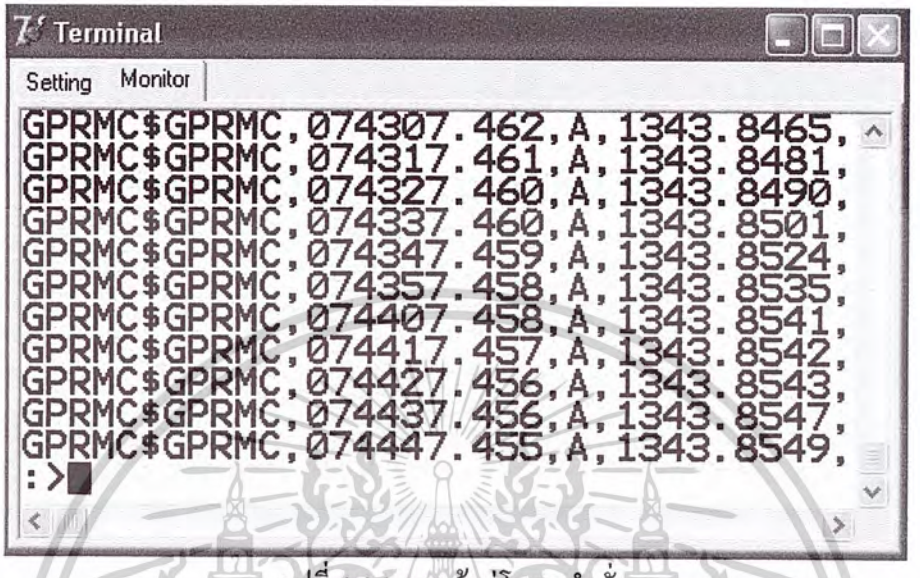


รูปที่ 4.3.5 การใช้ซอฟท์แวร์เทอร์มินอลรับค่าจากจีพีเอส

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

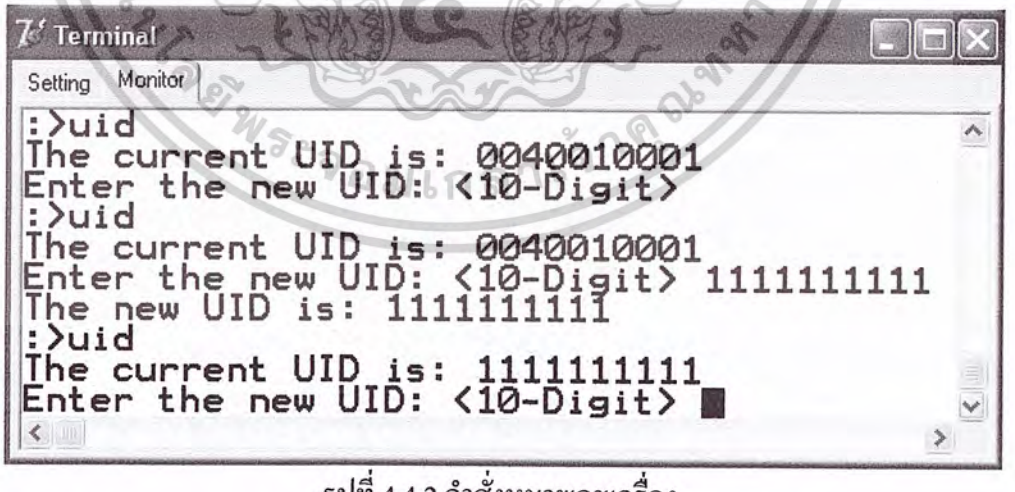
4.4 การทดลองและผลการทดลองระบบคำสั่งของเครื่องบันทึกการเดินทาง

การเข้าสู่โหมดคำสั่งของเครื่องสามารถทำได้โดยการพิมพ์เครื่องหมาย @ เข้าไปเครื่องจะตอบรับกลับมาด้วยข้อความ :> ช่วยเป็นเครื่องหมายแสดงว่าพร้อมรับคำสั่ง



รูปที่ 4.4.1 การเข้าสู่โหมดคำสั่ง

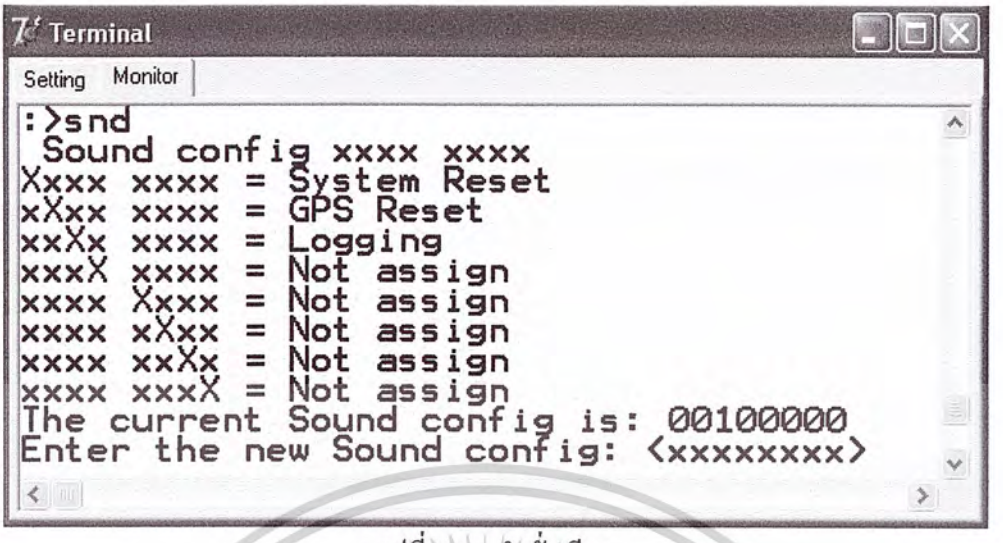
คำสั่งหมายเลขเครื่อง เป็นคำสั่งที่สามารถอ่านและเขียนหมายเลขเครื่องเพื่อประโยชน์ในการจัดการในกรณีที่ต้องใช้จำนวนมาก ทำได้โดยการพิมพ์ UID ไปที่โหมดคำสั่ง เครื่องก็จะบอกหมายเลขเครื่องเดิมมาพร้อมกับให้เปลี่ยนหมายเลขเครื่องใหม่ ถ้าไม่ต้องการเปลี่ยนให้กด Enter



รูปที่ 4.4.2 คำสั่งหมายเลขเครื่อง

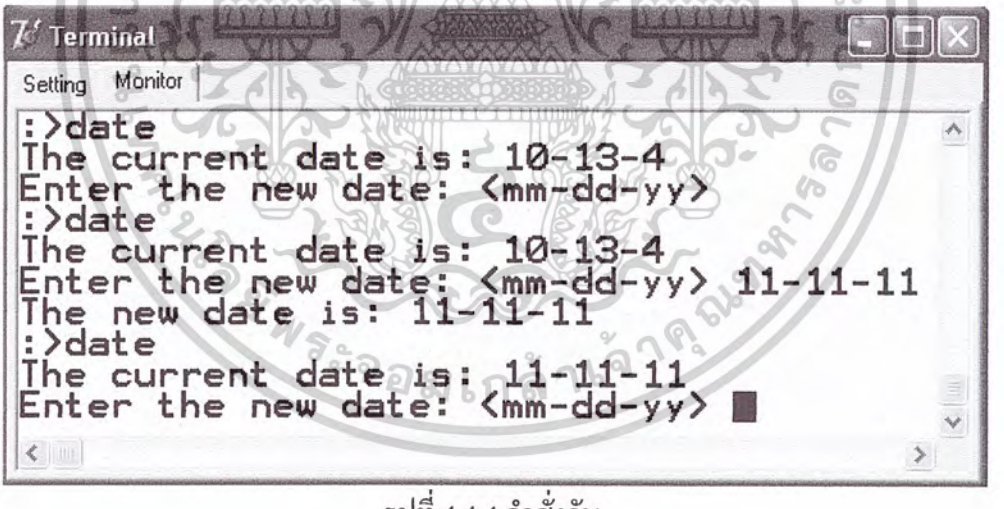
คำสั่งเสียง เป็นคำสั่งให้แสดงเสียงเมื่อมีเหตุการณ์ต่างๆ เช่นเวลาเปิดเครื่อง เวลาเก็บข้อมูล เป็นต้น ให้พิมพ์ SND ที่โหมดคำสั่ง เครื่องจะแสดงค่าเดิมที่ตั้งไว้พร้อมกับให้เปลี่ยนใหม่ ถ้าไม่ต้องการเปลี่ยนให้กด Enter

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.4.3 คำสั่งเสียง

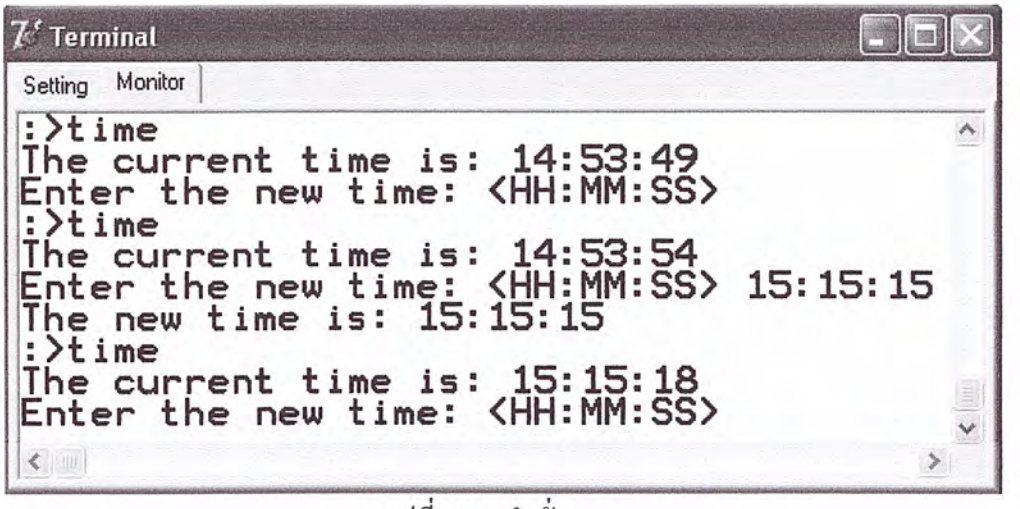
คำสั่งวัน เป็นคำสั่งที่ใช้อ่านหรือเขียนข้อมูล วัน เดือน และปี จากนาฬิกาในเครื่อง ให้พิมพ์ DATE ที่โหมดคำสั่ง เครื่องจะแสดง วัน เดือน และปี เดิมพร้อมกับให้เปลี่ยนข้อมูลใหม่ ถ้าไม่ต้องการเปลี่ยนให้กด Enter



รูปที่ 4.4.4 คำสั่งวัน

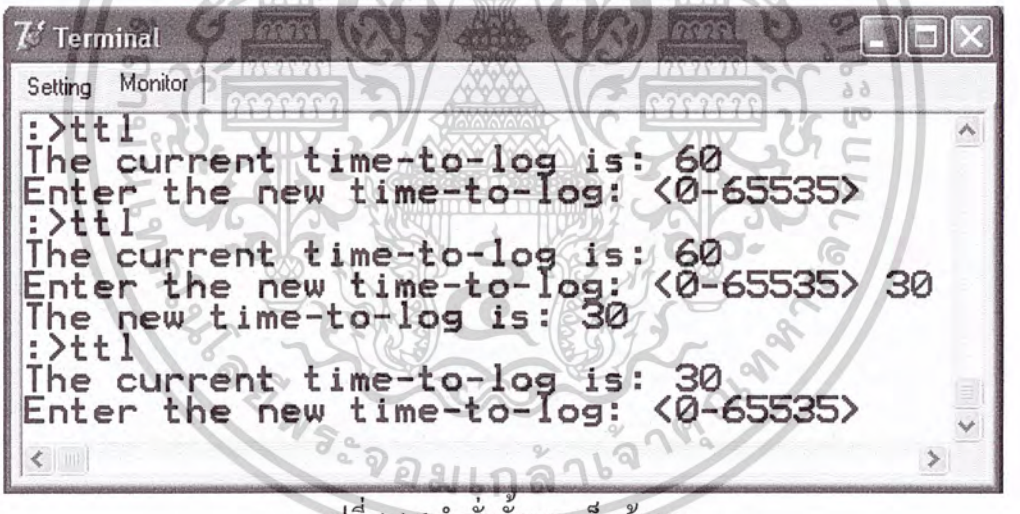
คำสั่งเวลา เป็นคำสั่งที่ใช้อ่านหรือเขียนข้อมูล เวลาเป็น ชั่วโมง นาที และวินาที จากนาฬิกาในเครื่อง ให้พิมพ์ TIME ที่โหมดคำสั่ง เครื่องจะแสดงชั่วโมง นาที และวินาที เดิมพร้อมกับให้เปลี่ยนข้อมูลใหม่ ถ้าไม่ต้องการเปลี่ยนให้กด Enter

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.4.5 คำสั่งเวลา

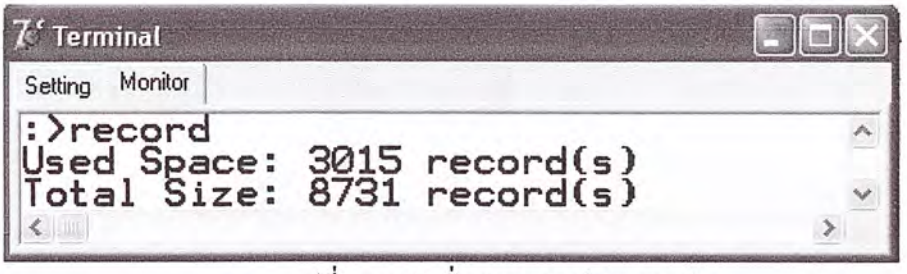
คำสั่งตั้งเวลาเก็บข้อมูล เป็นคำสั่งที่ใช้ตั้งเวลาที่จะเก็บข้อมูล มีหน่วยเป็นวินาที ให้พิมพ์ TTL(Time To Log) ที่โหมดคำสั่ง เครื่องจะแสดงเวลาเดิมพร้อมกับให้เปลี่ยนเวลาใหม่ ถ้าไม่ต้องการเปลี่ยนให้กด Enter



รูปที่ 4.4.6 คำสั่งตั้งเวลาเก็บข้อมูล

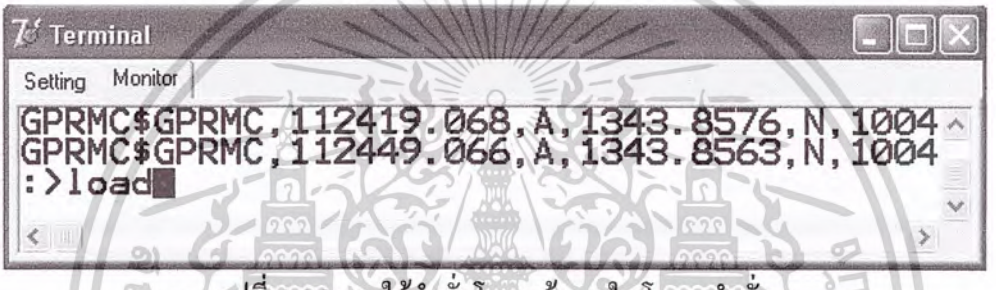
คำสั่งสถานะหน่วยความจำ เป็นคำสั่งที่ใช้ดูสถานะของตัวหน่วยความจำว่าสามารถเก็บข้อมูลได้ทั้งหมดเท่าไหร่ และข้อมูลตอนนี้เก็บไปถึงไหนแล้ว ให้พิมพ์ TTL(Time To Log) ที่โหมดคำสั่ง เครื่องจะแสดงข้อมูลของหน่วยความจำให้ดู

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

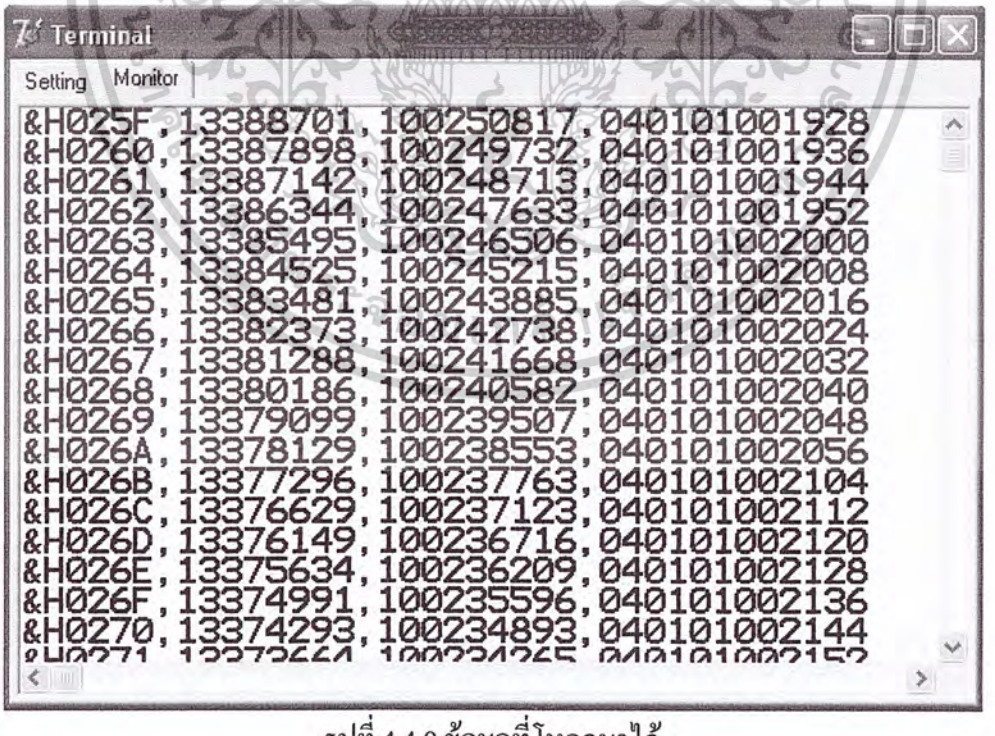


รูปที่ 4.4.7 คำสั่งสถานะหน่วยความจำ

คำสั่งโหลดข้อมูล เป็นการดึงข้อมูลการหน่วยความจำทั้งหมดมาแสดง ซึ่งเราสามารถใช้โปรแกรมดับข้อมูลนี้ไปแสดงการซอฟท์แวร์ทางด้านจีพีเอสได้ โดยการพิมพ์ LOAD ที่โหมดคำสั่ง จากนั้นเครื่องจะทำการโหลดข้อมูลทั้งหมดออกมา



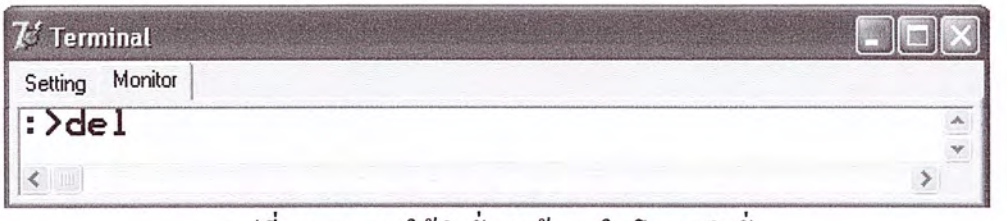
รูปที่ 4.4.8 การใช้คำสั่งโหลดข้อมูลในโหมดคำสั่ง



รูปที่ 4.4.9 ข้อมูลที่โหลดมาได้

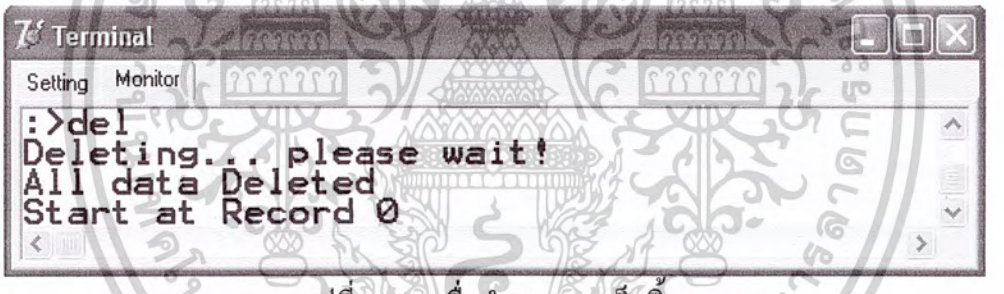
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คำสั่งลบข้อมูล เป็นคำสั่งที่ใช้ลบข้อมูลในหน่วยความจำทั้งหมด โดยการพิมพ์ DEL ในโหมดคำสั่ง



รูปที่ 4.4.10 การใช้คำสั่งลบข้อมูลในโหมดคำสั่ง

รูปที่ 4.4.11 ข้อมูลขณะทำการลบ

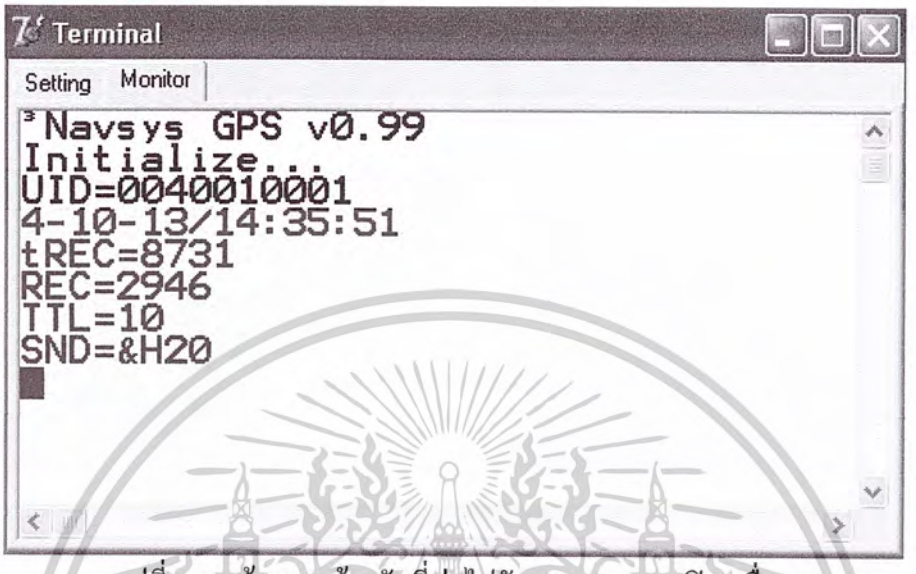


รูปที่ 4.4.12 เมื่อทำการลบเสร็จสิ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.5 การทดสอบฮาร์ดแวร์

ทดสอบ RS-232 โดยการเปิดเครื่องบันทึกการทดลองโดยจะมีข้อความต้องรับและค่าต่างๆ ของเครื่อง ทดที่ได้คือมีข้อความเหล่านั้นจริง



รูปที่ 4.5.1 ข้อความต้องรับที่ส่งไปยัง RS-232 ขณะเปิดเครื่อง

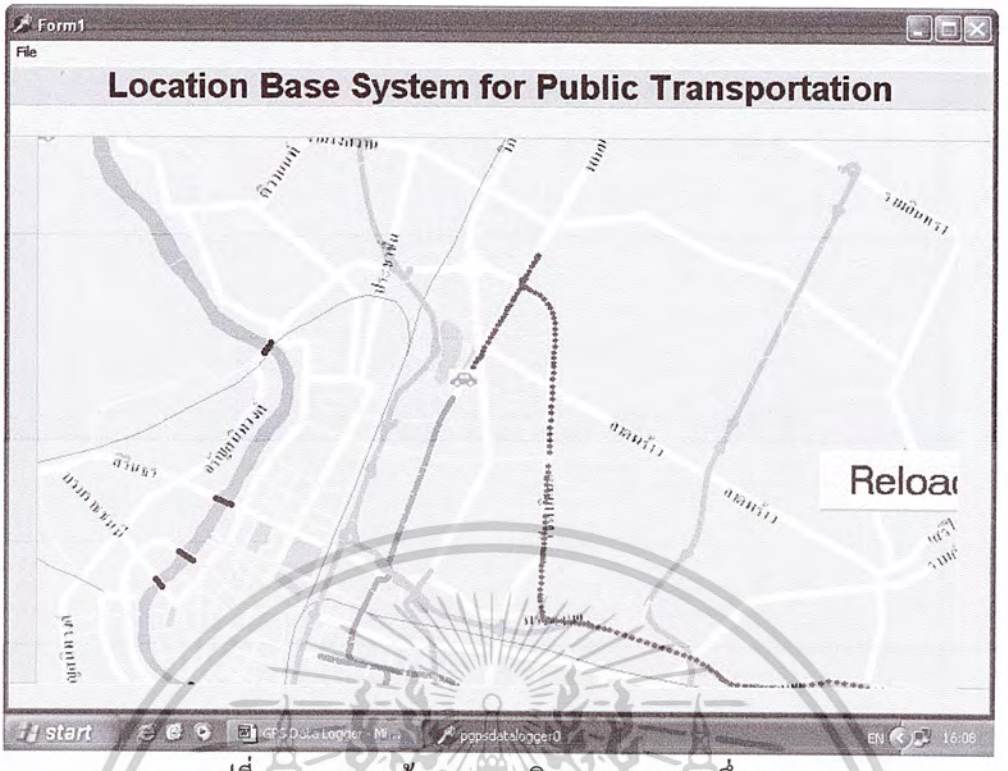
ทดสอบลำโพงบัซเซอร์ทำการตั้งค่าเสียงโดยคำสั่งเสียง SND ให้มีเสียงตอนเปิดเครื่องจะนั้นทำการรีเซตเครื่องใหม่ แล้วฟังเสียงจากลำโพงบัซเซอร์ ปรากฏว่ามีเสียงจริง

ทำการทดสอบหลอดไฟแสดงสถานะโดยการเปิดเครื่องใหม่จากการเขียนโปรแกรมสั่งให้หลอดแสดงเป็นไฟวิ่งก่อน แล้วสังเกตไฟ ปรากฏว่าหลอดแสดงเป็นไฟวิ่งจริง

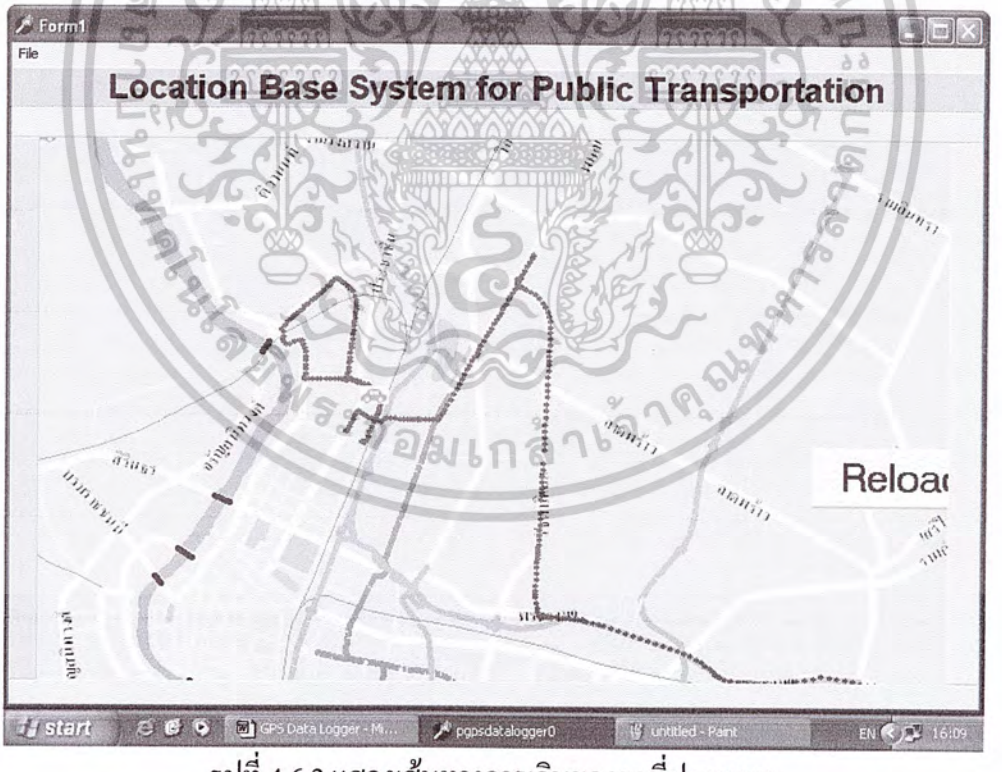
4.6 การทดลองและผลการทดลองของเครื่องบันทึกการเดินทางกับซอฟท์แวร์คอมพิวเตอร์

เป็นการใช้งานเครื่องบันทึกการเดินทางกับซอฟท์แวร์คอมพิวเตอร์ในเบื้องต้นโดยให้ซอฟท์แวร์คอมพิวเตอร์ทำการโหลดข้อมูลการเดินทางจากเครื่องบันทึกการเดินที่ได้ทำการนำไปติดในรถยนต์แล้วให้วิ่งไปตามถนนในกรุงเทพมหานคร ข้อมูลที่โหลดได้ซอฟท์แวร์คอมพิวเตอร์จะคำนวณแล้วนำค่าเส้นรุ้ง เส้นแวงไปแสดงบนแผนที่ในคอมพิวเตอร์ จะรถยนต์คันดังกล่าวได้เดินทางไปไหนมาบ้าง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.6.1 แสดงเส้นทางการเดินของรถระยะครึ่งทาง

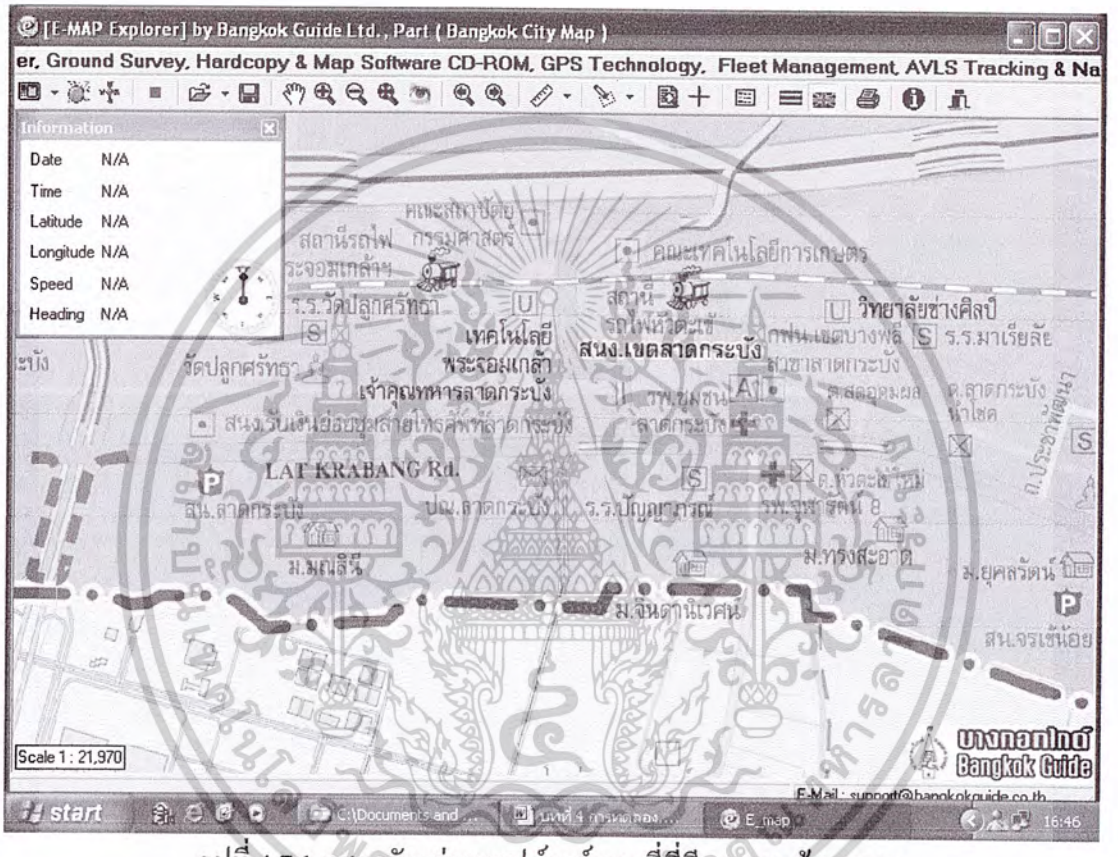


รูปที่ 4.6.2 แสดงเส้นทางการเดินของรถที่ปลายทาง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

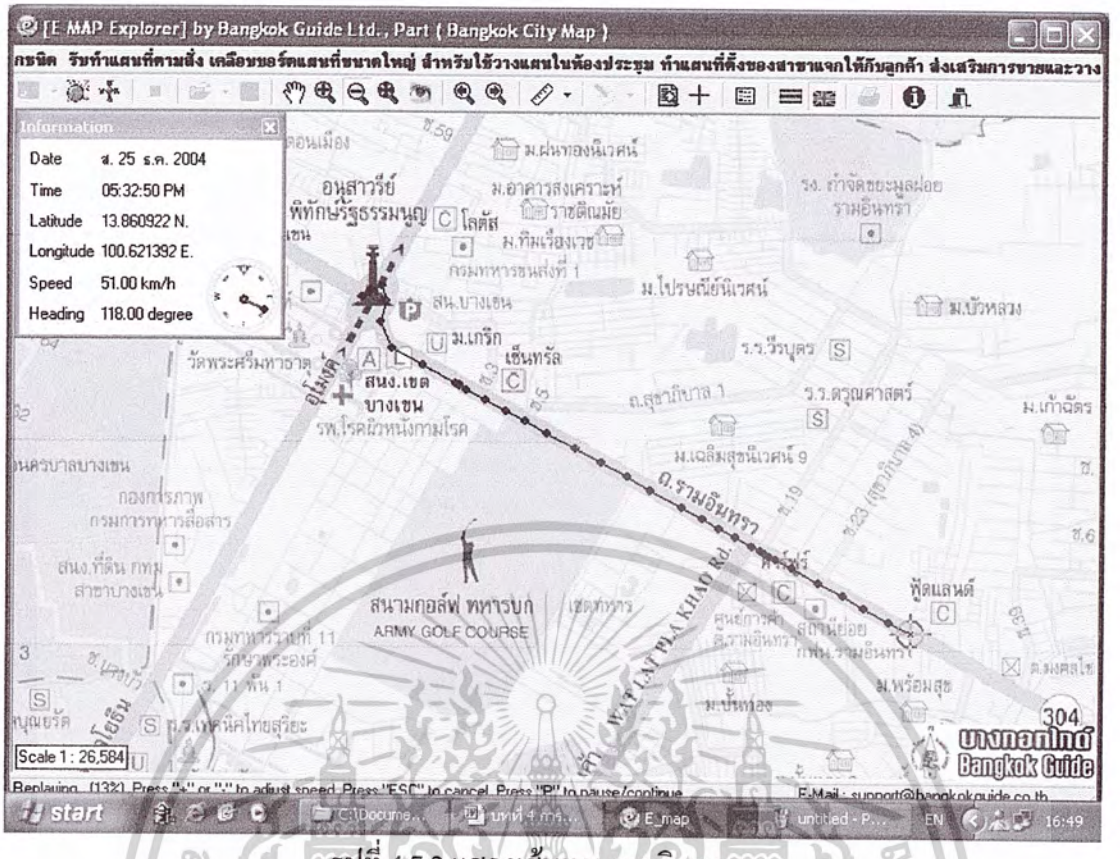
4.7 การทดลองและผลการทดลองของเครื่องบันทึกการเดินทางกับซอฟท์แวร์คอมพิวเตอร์ที่มีขายตามท้องตลาด

เป็นการใช้งานเครื่องบันทึกการเดินทางกับซอฟท์แวร์คอมพิวเตอร์ที่มีขายตามท้องตลาด โดยเลือกซอฟท์แวร์ของ บริษัท บางกอกไกด์ ซึ่งเป็นซอฟท์แวร์ที่ใช้มาตรฐาน NMEA และเลือกใช้ Protocol ของ RMC ซึ่งโปรแกรมสามารถแสดง วัน เวลา ความเร็ว และรายละเอียดอื่นๆของการเดินทาง



รูปที่ 4.7.1 แสดงตัวอย่างซอฟท์แวร์แผนที่ที่มีขายตามท้องตลาด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.7.2 แสดงเส้นทางการเดินของรถ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5
สรุปและวิจารณ์

การนำระบบจีพีเอสมาประยุกต์ใช้ในการบันทึกเส้นทางการเดินทาง โดยได้ศึกษาการทำงานของระบบจีพีเอส และเครื่องรับจีพีเอส จนสามารถนำเครื่องรับจีพีเอสมาทดลองศึกษาการทำงานตลอดจนเขียนโปรแกรมคอมพิวเตอร์ควบคุมได้ และได้นำเครื่องรับจีพีเอสมาประยุกต์ใช้กับไมโครคอนโทลเลอร์เพื่อที่จะสามารถเคลื่อนที่และติดตั้งบนรถยนต์ได้ โดยการออกแบบวงจรและเส้นลายทองแดงด้วยซอฟท์แวร์คอมพิวเตอร์ ซึ่งวงจรที่ได้มี วงจรแหล่งจ่ายไฟเป็นแหล่งจ่ายพลังงานให้แก่วงจร ซึ่งสามารถใช้ในรถยนต์ที่มีไฟกระชากได้เป็นอย่างดี วงจรนาฬิกาเป็นฐานเวลาให้แก่วงจรคอยที่จะบอก วัน เวลา และเป็นจับเวลาเก็บข้อมูล วงหน่วยความจำเป็นวงจรซึ่งเป็นที่เก็บข้อมูลการเดินทาง วงจรเชื่อมต่อเป็นวงจรที่เชื่อมต่อระหว่างไมโครคอนโทลเลอร์กับคอมพิวเตอร์ สลับกับไมโครคอนโทลเลอร์กับเครื่องรับจีพีเอส วงจรแสดงสถานะทำหน้าที่แสดงการทำงานของระบบ การรับสัญญาณจากดาวเทียม โหมดของการทำงาน เสียงที่จะคอยเตือนผู้ขับขี่ โดยวงจรทั้งหมดสามารสที่จะทำงานร่วมกันเป็นเครื่องบันทึกการเดินทางได้อย่างมีประสิทธิภาพ เครื่องนี้สามารถทำงานร่วมกับคอมพิวเตอร์ได้ผ่านพอร์ต RS-232 ซึ่งสามารถที่จะดูหรือเปลี่ยนแปลงค่าต่างๆ เช่น วันเวลา เวลาในการเก็บข้อมูล การตั้งเสียง การดูความจุและจำนวนการใช้งานของหน่วยความจำ และการโหลดข้อมูลการเดินทาง โดยข้อมูลนี้สามารถนำมาวิเคราะห์กับคอมพิวเตอร์เพื่อแสดงภาพเส้นทางการเดินทางได้

จากการสร้างเครื่องบันทึกการเดินทาง มีปัญหาต่างๆ เช่น ระบบไฟฟ้าในรถยนต์มีการกระชากของไฟมากทำให้วงจรแหล่งจ่ายไฟเสียหาย การทำงานของไมโครคอนโทลเลอร์ทำงานช้ากว่าคอมพิวเตอร์มาก เพราะฉะนั้นการวิเคราะห์ข้อมูลจากเครื่องรับจีพีเอสจึงไม่ทันทั้งหมดจึงได้มีการให้วิเคราะห์เฉพาะข้อมูลที่จำเป็นต้องใช้เท่านั้น การรับค่าข้อมูลทางพอร์ตอนุกรมของไมโครคอนโทลเลอร์ในภาษาเบสิกเป็นแบบวนรับทำให้เกิดปัญหาในเรื่องจังหวะการวนรับ โดยจะมีการแก้ไขต่อไปโดยจะเปลี่ยนเป็นรับการขัดจังหวะ(Interrupt) แทน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ซอร์ซโค๊ด(Source Code) ภาษาเบสิก

```
'--------------------------------------------------------------------------------
$regfile = "89c51rd.dat"
$large
$baud = 4800
$crystal = 22118400                    '11059200MHz * 2
$ramstart = 0
$ramsize = 1024
$timeout
'--------------------------------------------------------------------------------
Dim int_BYTE As Byte , int_WORD As Word
Dim PAGE As Byte
Dim Hi_BYTE As Byte , Lo_BYTE As Byte
Dim Addr  As Long
Dim Value As Byte
Dim KBhit As Byte
Dim REC   As Word
Dim cREC  As Word
Dim GPS   As Byte
Dim SND   As Byte
Dim TTL   As Word
Dim TTL_countor As Word
Dim i  As Byte , j As Byte  , k As Byte
Dim S  As Byte , M  As Byte , H  As Byte , S_old As Byte
Dim dd As Byte , mm As Byte , yy As Byte , Day   As Byte , CLK_ctrl As Byte
Dim Line   As Byte , LCD_data As Byte
Dim EEPROM As Byte
Dim Pos_01 As Byte
Dim Pos_02 As Byte
Dim Length As Byte
Dim Speed  As Single
Dim oneCHR As String * 1
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Dim Course As String * 6
Dim latSTR As String * 9
Dim lonSTR As String * 10
Dim subSTR As String * 10
Dim sysSTR As String * 80
'--------------------------------------------------------------------------
Declare Sub TTL
Declare Sub read_TTL
Declare Sub write_TTL
Declare Sub DEL
Declare Sub GPS
Declare Sub SND
Declare Sub Date
Declare Sub Time
Declare Sub GPGGA
Declare Sub GPRMC
Declare Sub info
Declare Sub initial
Declare Sub get_CLK
Declare Sub read_CLK
Declare Sub write_clk
Declare Sub initial_clk
Declare Sub Read_eeprom(addr As Long)
Declare Sub Write_eeprom(addr As Long , Value As Byte)
Declare Sub Save_data(rec As Word)
Declare Sub Load_data
Declare Sub LCD_initial
Declare Sub LCD_ctrl(LCD_data As Byte)
Declare Sub LCD_data(LCD_data As Byte)
Declare Sub LCD_append(sysSTR As String * 80)
Declare Sub LCD_wSTR(sysSTR As String * 80 , Line As Byte)
'--------------------------------------------------------------------------
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
' - Pin Config
' GPS
GPTXA    Alias P3.6
'Gprxa Alias P1.6
PBRES    Alias P3.3
' LCD
RS       Alias P0.6
E        Alias P0.7
' Buzzer
Buzzer   Alias P2.0
' LED

gLED1    Alias P1.0
gLED2    Alias P1.1
gLED3    Alias P1.2
rLED1    Alias P1.5
rLED2    Alias P1.4
rLED3    Alias P1.3

'LED_r    Alias P1.1
'LED_g    Alias P1.2
' EEPROM
Write_Protect Alias P3.2
Config I2Cdelay  = 4
Config SCL       = P1.6
Config SDA       = P1.7
Const  EEPROMsrt  = 100
Const  EEPROMend  = 131072
Const  dataSIZE   = 15
' CLOCK
Const  Clkw      = 208
Const  Clkr      = 209
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
' Analog-to-Digital
Const  ADCw      = 144                              ' 1001 0000 (Write)
Const  ADCr      = 145                              ' 1001 0001 (Read)
Const  ADCc      = &B00000000
Const  UIDsrt    = 1
Const  UIDend    = 10
Const  tREC      = 8731
'-------------------------------------------------------------------------
' - Initialize
Sound  Buzzer , 1000 , 10
Print  "Navsys GPS v0.99"
Print  "Initialize..."
Reset  PBRES
Waitms 100
Set    PBRES
Call   Initial
Reset  GPTXA
'-------------------------------------------------------------------------
P1 =  &HFE
For i = 1 To 6
  Wait 1
  Rotate P1,Left,1
Next
Reset GPTXA
Print
Print "$PSRF103,00,00,00,01*24"
Print "$PSRF103,01,00,00,01*25"
Print "$PSRF103,02,00,00,01*26"
Print "$PSRF103,03,00,00,01*27"
Print "$PSRF103,04,00,01,01*28"
Print "$PSRF103,05,00,00,01*29"
Print "$PSRF103,06,00,00,01*2A"
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Print "$PSRF103,07,00,00,01*2B"
Print "$PSRF103,08,00,00,01*2C"
Print "$PSRF103,08,00,00,01*2D"
Set GPTXA
Reset gLED1
Begin:
  KBhit = Inkey()
  If KBhit = &H40 Then
    Reset  rLED3
    Set  GPTXA
    Input ":>" , sysSTR Timeout = 1000000
    sysSTR = Lcase(sysSTR)
    If    sysSTR = "date"   Then
      Call Date
    Elseif sysSTR = "time"   Then
      Call Time
    Elseif sysSTR = "uid"    Then
      Call UID
    Elseif sysSTR = "ttl"    Then
      Call TTL
    Elseif sysSTR = "del"    Then
      Call DEL
    Elseif sysSTR = "load"   Then
      Call load_DATA
    Elseif sysSTR = "record" Then
      Call Record
    Elseif sysSTR = "gps"    Then
      Call GPS
    Elseif sysSTR = "snd"    Then
      Call SND
    Elseif sysSTR = "info"    Then
      Call info
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    End If
    Set rLED3
  Else
    Call get_CLK
    If S <> S_old Then
      If gLED3 = 1 Then gLED3 = 0 Else gLED3 = 1
      Incr TTL_countor
      If TTL_countor = TTL Then
        TTL_countor = 0
        '+++ Clear Variable +++
        latSTR = ""
        lonSTR = ""
        '++++++++++++++++++++++
        Call GPRMC
        if latSTR = "0000.0000" Then
          Set   gLED2
          Reset rLED2
        else
          Reset gLED2
          Set   rLED2
        endif
        Call save_DATA(REC)
        If SND.5 = 1 Then
          Sound Buzzer , 255 , 11
          Sound Buzzer , 255 , 10
        End If
      End If
      S_old = S
    End If
  End If
Goto Begin
'-------------------------------------------------------------------------
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Sub initial
  sysSTR = ""
  For i  = UIDsrt To UIDend
    Call read_EEPROM(i)
    oneCHR = Str(Value)
    sysSTR = sysSTR + oneCHR
  Next
  Print "UID=" ; sysSTR
  Call  get_CLK
  Print yy ; "-" ; mm ; "-" ; dd ; "/" ; H ; ":" ; M ; ":" ; S
  Print "tREC=" ; tREC
  Call  read_TTL
  Call  read_EEPROM(16) : SND = Value
  Call  auto_RECORD
  Print "REC=" ; REC
  Print "TTL=" ; TTL
  Printhex "SND=&H" ; SND
End Sub
'-------------------------------------------------------------------------
Sub info
  sysSTR = ""
  For i  = UIDsrt To UIDend
    Call read_EEPROM(i)
    oneCHR = Str(Value)
    sysSTR = sysSTR + oneCHR
  Next
  Print "UID=" ; sysSTR
  Call  get_CLK
  Print yy ; "-" ; mm ; "-" ; dd ; "/" ; H ; ":" ; M ; ":" ; S
  Print "tREC=" ; tREC
  Call  read_TTL
  Call  read_EEPROM(16) : SND = Value
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  Print "REC=" ; REC
  Print "TTL=" ; TTL
  Printhex "SND=&H" ; SND
End Sub
'-------------------------------------------------------------------------
' - Read Byte from EEPROM address
Sub read_EEPROM(addr As Long)
  EEPROM = &B10100000                    'Write Mode
  If Addr > 65535 Then                   '-------------------------
    int_WORD = Addr - 65535              'Page 1
    EEPROM.1 = 1                         'if Address more than 65535
  Else                                   '-------------------------
    int_WORD = Addr                      'Page 0
    EEPROM.1 = 0                         'if Address less than 65535
  End If                                 '-------------------------
  MOV {Hi_BYTE} , {int_WORD + 1}         'Hi-Byte
  MOV {Lo_BYTE} , {int_WORD + 0}         'Lo-Byte
  I2cstart                               'Generate start
  I2cwbyte EEPROM                        'Write adsress
  I2cwbyte Hi_BYTE                       'Hi-Byte address
  I2cwbyte Lo_BYTE                       'Hi-Byte address
  Incr EEPROM                            'Change to Read Mode
  I2cstart                               'Repeated start
  I2cwbyte EEPROM                        'Read address
  I2crbyte Value , 9                     'Read byte
  I2cstop                                'Generate stop
End Sub
'-------------------------------------------------------------------------
' - Write Byte to EEPROM address
Sub write_EEPROM(addr As Long , Value As Byte)
  CLR  Write_Protect                     'Allow to Write
  EEPROM = &B10100000                    'Write Mode
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  If Addr > 65535 Then                    '-------------------------
    int_WORD = Addr - 65535               'Page 1
    EEPROM.1 = 1                          'if Address more than 65535
  Else                                    '-------------------------
    int_WORD = Addr                       'Page 0
    EEPROM.1 = 0                          'if Address less than 65535
  End If                                  '-------------------------
  MOV {Hi_BYTE} , {int_WORD + 1}          'Hi-Byte
  MOV {Lo_BYTE} , {int_WORD + 0}          'Lo-Byte
  I2cstart                                'Generate start
  I2cwbyte EEPROM                         'Write adsress
  I2cwbyte Hi_BYTE                        'Hi-Byte address
  I2cwbyte Lo_BYTE                        'Hi-Byte address
  I2cwbyte Value                          'Write Byte
  I2cstop                                 'Generate stop
  Waitms 20                               'Wait for 10 mSec
  SETB  Write_Protect                     'Deny to Write
End Sub
'-------------------------------------------------------------------------
' - View RECORD status
Sub RECORD
  Print "Used Space: " ; REC ; " record(s)"
  Print "Total Size: " ; tREC ; " record(s)"
End Sub
'-------------------------------------------------------------------------
Sub auto_RECORD
  REC = 0
  Addr = REC * dataSIZE
  Addr = Addr + EEPROMsrt
  Incr Addr
  Call read_EEPROM(addr)
  If Value <> &HFF Then
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    Do
      Incr REC
      If REC > tREC Then
        REC = 0
        Exit Do
      End If
      Addr = REC * Datasize
      Addr = Addr + EEPROMsrt
      Incr Addr
      Call read_EEPROM(addr)
    Loop Until Value = &HFF
  End If
End Sub
'-------------------------------------------------------------------
Sub load_DATA
  Addr = EEPROMsrt
  Print REC ; " record(s)"
  For cREC = 1 To REC
    sysSTR = ""
    latSTR = ""
    lonSTR = ""
    subSTR = Hex(cREC)
    sysSTR = "&H" + subSTR
    sysSTR = sysSTR + ","
    ' Latitude
    For I = 1 To 4
      Incr Addr
      Call read_EEPROM(addr)
      Hi_BYTE = Value And &HF0
      Lo_BYTE = Value And &H0F
      Rotate Hi_BYTE , Right , 4
      oneCHR = Str(Hi_BYTE)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    latSTR = latSTR + oneCHR
    oneCHR = Str(Lo_BYTE)
    latSTR = latSTR + oneCHR
  Next
  ' Longitue
  For I = 1 To 4
    Incr Addr
    Call read_EEPROM(addr)
    Hi_BYTE = Value And &HF0
    Lo_BYTE = Value And &H0F
    Rotate Hi_BYTE , Right , 4
    oneCHR = Str(Hi_BYTE)
    lonSTR = lonSTR + oneCHR
    oneCHR = Str(Lo_BYTE)
    lonSTR = lonSTR + oneCHR
  Next
  Incr Addr
  Call read_EEPROM(addr)
  oneCHR = Str(Value)
  lonSTR = lonSTR + oneCHR

  sysSTR = sysSTR + latSTR
  sysSTR = sysSTR + ","
  sysSTR = sysSTR + lonSTR
  sysSTR = sysSTR + ","
  For I = 1 To 6
    Incr Addr
    Call read_EEPROM(addr)
    subSTR = Str(Value)
    Length = Len(subSTR)
    If Length = 1 Then sysSTR = sysSTR + "0"
    sysSTR = sysSTR + subSTR
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    Next
    Print sysSTR
  Next
End Sub
'--------------------------------------------------------------------------
' - Save Data to EEPROM
'    Total 15 Byte
'    4 Byte : Latitude
'    5 Byte : Longitude
'    6 Byte : DateTime
Sub save_DATA(REC As Word)
  ' Total 15 Byte
  addr = REC  * dataSIZE
  addr = addr + EEPROMsrt
  ' Latitude
  ' 4 Byte
  subSTR = latSTR
  latSTR = ""
  For i  = 1 To 9
    oneCHR = Mid(subSTR , i , 1)
    If oneCHR <> "." Then latSTR = latSTR + oneCHR
  Next
  For i  = 1 To 8
    oneCHR   = Mid(latSTR , i , 1)
    Value    = Val(oneCHR)
    Rotate Value , Left , 4
    Incr i
    oneCHR   = Mid(latSTR , i , 1)
    int_BYTE = Val(onechr)
    Value    = Value + int_BYTE
    Incr Addr
    Call write_EEPROM(addr , Value)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Next
' Longitude
' 5 Byte
subSTR = lonSTR
lonSTR = ""
For i  = 1 To 10
  oneCHR = Mid(subSTR , i , 1)
  If oneCHR <> "." Then lonSTR = lonSTR + oneCHR
Next
For i = 1 To 8
  oneCHR   = Mid(lonSTR , i , 1)
  Value    = Val(oneCHR)
  Rotate Value , Left , 4
  Incr I
  oneCHR   = Mid(lonSTR , i , 1)
  int_BYTE = Val(oneCHR)
             Value    = Value + int_BYTE
  Incr Addr
  Call write_EEPROM(addr , Value)
Next
Onechr     = Mid(lonstr , 9 , 1)
Value      = Val(onechr)
Incr Addr
Call write_EEPROM(addr , Value)
' DateTime
' 6 Byte
Incr Addr : Call write_EEPROM(addr , yy)
Incr Addr : Call write_EEPROM(addr , mm)
Incr Addr : Call write_EEPROM(addr , dd)
Incr Addr : Call write_EEPROM(addr , H)
Incr Addr : Call write_EEPROM(addr , M)
Incr Addr : Call write_EEPROM(addr , S)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  '---------- INC & SAVE RECORD ----------
  Incr REC
  If REC > tREC Then REC = 0
  Incr Addr : CALL write_EEPROM(addr,&HFF)
End Sub
'--------------------------------------------------------------------------
' - Time-to-Log
Sub TTL
  Print "The current time-to-log is: " ; Ttl          'Show current TTL
  Input "Enter the new time-to-log: <0-65535> " , Sysstr Timeout = 1000000
  If Sysstr <> "" Then
    TTL = Val(sysstr)                                 'get  TTL
    Call write_TTL                                    'Write TTL to  EEPROM
    Call read_TTL                                     'Read  TTL from EEPROM
    Print "The new time-to-log is: " ; Ttl            'Show new TTL
    TTL_countor = 0                                   'Reset TTL_countor
  End If
End Sub
'--------------------------------------------------------------------------
' - Write TTL to EEPROM
Sub write_TTL
  MOV  {VALUE},{TTL+1}                                'get  Hi-Byte
  Call write_EEPROM(11 , Value)                       'write Hi-Byte
  MOV  {VALUE},{TTL+0}                                'get  Lo-Byte
  Call write_EEPROM(12 , Value)          'write Lo-Byte
End Sub
'--------------------------------------------------------------------------
' - Read TTL from EEPROM
Sub read_TTL
  Call read_EEPROM(11)                                'read  Hi-Byte
  MOV  {TTL+1},{VALUE}                                'get  Hi-Byte
  Call read_EEPROM(12)                                'read  Lo-Byte
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  MOV  {TTL+0},{VALUE}                    'get  Lo-Byte
End Sub
'-------------------------------------------------------------------------------
Sub UID
   ' xxxxxxxxxx : 10 Digit Unit ID
   ' XXXxxxxxxx : Year
   ' xxxXXXxxxx : Lot
   ' xxxxxxXXXX : Number
   sysSTR = ""
   For i = UIDsrt To UIDend
    Call read_EEPROM(i)
    oneCHR = Str(Value)
    sysSTR = sysSTR + oneCHR
   Next
   Print "The current UID is: " ; sysSTR
   Input "Enter the new UID: <10-Digit> " , sysSTR Timeout = 1000000
   If sysSTR <> "" Then
    Length = Len(sysSTR)
    If Length <> 10 Then
      Print "Wrong data!"
      Goto End_UID
    End If
    For i = UIDsrt To UIDend
      oneCHR = Mid(sysSTR , i , 1)
      Value  = Val(oneCHR)
      Call write_EEPROM(i , Value)
    Next
    sysSTR = ""
    For i = UIDsrt To UIDend
      Call read_EEPROM(i)
      oneCHR = Str(Value)
      sysSTR = sysSTR + oneCHR
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    Next
    Print "The new UID is: " ; sysSTR
  End If
End_UID:
End Sub
'-------------------------------------------------------------------------
' - DELETE data all EEPROM
Sub DEL
  Print "Deleting... please wait!"
  'Print "Deleting... block1"
  for REC = 0 to tREC
    addr = REC  * dataSIZE
    addr = addr + EEPROMsrt
    INCR addr
    CALL write_EEPROM(addr,&HFF)
  next
  '(Addr = Eepromsrt                    'Start Address to DELETE
  EEPROM = &B10100000                   'DELETE Block1
  for PAGE = 0 to 255
    CLR Write_Protect                   'Allow to DELETE
    addr = 255 * PAGE
    if addr <> 0 then INCR addr
    MOV {Hi_BYTE} , {addr + 1}          'Hi-Byte
    MOV {Lo_BYTE} , {addr + 0}          'Lo-Byte
    I2Cstart                            'Generate start
    I2Cwbyte EEPROM                     'Write adsress
    I2Cwbyte Hi_BYTE                    'Hi-Byte address
    I2Cwbyte Lo_BYTE                    'Hi-Byte address
    For i = 0 To 255                    'Loop Until End-of-EEPROM
      I2cwbyte 0                        'Write &H00 to EEPROM
    Next
    I2cstop                             'Generate stop
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    Waitms 20                          'Wait for 10 mSec
    SETB Write_Protect                 'Deny to DELETE
    Print "Deleted PAGE"; PAGE
  next
  Print "Block1 Cleared"
  Print "Deleting... block2"
  EEPROM = &B10100010                  'DELETE Block2
  for PAGE = 0 to 255
    CLR Write_Protect                  'Allow to DELETE
    addr = 255 * PAGE
    if addr <> 0 then INCR addr
    MOV {Hi_BYTE} , {addr + 1}           'Hi-Byte
    MOV {Lo_BYTE} , {addr + 0}           'Lo-Byte
    I2Cstart                         'Generate start
    I2Cwbyte EEPROM                    'Write adsress
    I2Cwbyte Hi_BYTE                   'Hi-Byte address
    I2Cwbyte Lo_BYTE                   'Hi-Byte address
    For i = 0 To 255                   'Loop Until End-of-EEPROM
      I2cwbyte 0                       'Write &H00 to EEPROM
    Next
    I2cstop                          'Generate stop
    Waitms 20                          'Wait for 10 mSec
    SETB Write_Protect                 'Deny to DELETE
    Print "Deleted PAGE"; PAGE
  next
  ')
  'Print "Block2 Cleared"
  REC = 0
  Print "All data Deleted"
  Print "Start at Record " ; REC
End Sub
'--------------------------------------------------------------------------
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Sub Date
  Call read_CLK
  Call get_CLK
  Print "The current date is: " ; mm ; "-" ; dd ; "-" ; yy
  Input "Enter the new date: <mm-dd-yy> " , sysSTR Timeout = 1000000
  If sysSTR <> "" Then
    Length = Len(sysSTR)
    If Length <> 8 Then
      Print "Wrong data!"
      Goto End_date
    End If
    '07-20-04
    subSTR = Mid(sysSTR , 1 , 2)
    mm     = Hexval(subSTR)
    subSTR = Mid(sysSTR , 4 , 2)
    dd     = Hexval(subSTR)
    subSTR = Mid(sysSTR , 7 , 2)
    yy     = Hexval(subSTR)
    subSTR = Str(H)
    H      = Hexval(subSTR)
    subSTR = Str(M)
    M      = Hexval(subSTR)
    subSTR = Str(S)
    S      = Hexval(subSTR)
    Call write_CLK
    Call read_CLK
    Call get_CLK
    Print "The new date is: " ; mm ; "-" ; dd ; "-" ; yy
  End If
End_date:
End Sub
'--------------------------------------------------------------------------
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Sub Time
  Call read_CLK
  Call get_CLK
  Print "The current time is: " ; H ; ":" ; M ; ":" ; S
  Input "Enter the new time: <HH:MM:SS> " , Sysstr Timeout = 1000000
  If sysSTR <> "" Then
    Length = Len(sysSTR)
    If Length <> 8 Then
      Print "Wrong data!"
      Goto End_time
    End If
    subSTR = Mid(sysSTR , 1 , 2)
    H      = Hexval(subSTR)
    subSTR = Mid(sysSTR , 4 , 2)
    M      = Hexval(subSTR)
    subSTR = Mid(sysSTR , 7 , 2)
    S      = Hexval(subSTR)
    subSTR = Str(yy)
    yy     = Hexval(subSTR)
    subSTR = Str(mm)
    mm     = Hexval(subSTR)
    subSTR = Str(dd)
    dd     = Hexval(subSTR)
    Call write_CLK
    Call read_CLK
    Call get_CLK
    Print "The new time is: " ; H ; ":" ; M ; ":" ; S
  End If
End_time:
End Sub
'--------------------------------------------------------------------------
Sub get_CLK
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    Call read_CLK : If S = 128 Then Call initial_CLK
  ' - SECOND
    S     = S And &B01111111
    subSTR = Hex(S)
    S     = Val(subSTR)
  ' - MINUTE
    M     = M And &B01111111
    subSTR = Hex(M)
    M     = Val(subSTR)
  ' - HOUR
    H     = H And &B00111111
    subSTR = Hex(H)
    H     = Val(subSTR)
  ' - DAY
    Day   = Day And &B00000111
    subSTR = Hex(Day)
    Day   = Val(subSTR)
  ' - DATE
    dd    = dd And &B00111111
    subSTR = Hex(dd)
  ' - MOUNT
    dd    = Val(subSTR)
    mm    = mm And &B00111111
    subSTR = Hex(mm)
    mm    = Val(subSTR)
  ' - YEAR
    subSTR = Hex(yy)
    yy    = Val(subSTR)
End Sub
'--------------------------------------------------------------------------
Sub read_CLK
  I2cstart
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    I2cwbyte CLKw
    I2cwbyte 0
    I2cstart
    I2cwbyte CLKr
    I2crbyte S , Ack
    I2crbyte M , Ack
    I2crbyte H , Ack
    I2crbyte Day , Ack
    I2crbyte dd , Ack
    I2crbyte mm , Ack
    I2crbyte yy , Ack
    I2crbyte CLK_ctrl , Nack
    I2cstop
End Sub
'-------------------------------------------------------------------
Sub write_CLK
    I2cstart
    I2cwbyte CLKw
    I2cwbyte 0
    I2cwbyte S
    I2cwbyte M
    I2cwbyte H
    I2cwbyte Day
    I2cwbyte dd
    I2cwbyte mm
    I2cwbyte yy
    I2cwbyte 144
    I2cstop
    Waitms 20
End Sub
'-------------------------------------------------------------------
Sub initial_CLK
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    yy = 04 : mm = 01 : dd = 01 : Day = 05 : H = 00 : M = 00 : S = 00
    Call write_CLK
    Call get_CLK
End Sub
'-------------------------------------------------------------------------
Sub GPS
  Print " GPS config xxxx xxxx "                    ' ADDR=17
  Print "Xxxx xxxx = Gps Update"
  Print "xXxx xxxx = Not assign"
  Print "xxXx xxxx = Not assign"
  Print "xxxX xxxx = Not assign"
  Print "xxxx Xxxx = Not assign"
  Print "xxxx xXxx = Not assign"
  Print "xxxx xxXx = Not assign"
  Print "xxxx xxxX = Not assign"
  sysSTR = ""
  Call read_EEPROM(17) : GPS = Value
  For i = 7 Downto 0
    If GPS.i = 1 Then sysSTR = sysSTR + "1" Else sysSTR = sysSTR + "0"
  Next
  Print "The current GPS config is: " ; sysSTR
  Input "Enter the new GPS config: <xxxxxxxx> " , sysSTR Timeout = 1000000
  If sysSTR <> "" Then
    Length = Len(sysSTR)
    If Length <> 8 Then
      Print "Wrong data!"
      Goto End_GPS
    End If
    For i = 1 To 8
      j      = 8 - i
      oneCHR  = Mid(sysSTR , i , 1)
      Value   = Val(oneCHR)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    If Value = 1 Then GPS.j = 1 Else GPS.j = 0
  Next
  Call write_EEPROM(17 , GPS)
  Call read_EEPROM(17) : GPS = Value
  sysSTR = ""
  For i = 7 Downto 0
    If GPS.i = 1 Then sysSTR = sysSTR + "1" Else sysSTR = sysSTR + "0"
  Next
  Print "The new GPS config is: " ; sysSTR
 End If
End_GPS:
End Sub
'--------------------------------------------------------------------
Sub SND
 Print " Sound config xxxx xxxx "                ' ADDR=16
 Print "Xxxx xxxx = System Reset"
 Print "xXxx xxxx = GPS Reset"
 Print "xxXx xxxx = Logging"
 Print "xxxX xxxx = Not assign"
 Print "xxxx Xxxx = Not assign"
 Print "xxxx xXxx = Not assign"
 Print "xxxx xxXx = Not assign"
 Print "xxxx xxxX = Not assign"
 sysSTR = ""
 Call read_EEPROM(16) : SND = Value
 For i = 7 Downto 0
   If SND.i = 1 Then sysSTR = sysSTR + "1" Else sysSTR = sysSTR + "0"
 Next
 Print "The current Sound config is: " ; sysSTR
 Input "Enter the new Sound config: <xxxxxxxx> " , sysSTR Timeout = 1000000
 If sysSTR <> "" Then
   Length = Len(sysSTR)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    If Length <> 8 Then
      Print "Wrong data!"
      Goto End_SND
    End If
    For i = 1 To 8
      j = 8 - i
      oneCHR = Mid(sysSTR , i , 1)
      Value  = Val(oneCHR)
      If Value = 1 Then SND.j = 1 Else SND.j = 0
    Next
    Call write_EEPROM(16 , SND)
    Call read_EEPROM(16) : SND = Value
    sysSTR = ""
    For i = 7 Downto 0
      If SND.i = 1 Then sysSTR = sysSTR + "1" Else sysSTR = sysSTR + "0"
    Next
    Print "The new Sound config is: " ; sysSTR
  End If
End_SND:
End Sub
'-------------------------------------------------------------------
Sub GPGGA
  subSTR = "$GPGGA"
  Input sysSTR Noecho Timeout = 100000
  Pos_01 = Instr(sysSTR , subSTR)
  If Pos_01 <> 0 Then
    Print sysSTR
    ' - GGA     : Global postition system fixed data -
    ' $GPGGA,094226.503,1343.1003,N,10046.6719,E,1,03,14.3,2.6,M,,,,0000*3E
    subSTR = ","
    ' - $GPGGA  : GGA protal header
    Length = Len(sysSTR)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    Pos_01 = Instr(sysSTR , subSTR )
'   Pos_02 = Pos_01 - 1
    Length = Length - Pos_01
'   Substr = Left(sysstr , Pos_02)
    Sysstr = Right(sysSTR , Length)
    '---------------------------
'   Print "GGA : " ; Substr
'   Print Sysstr
    '---------------------------
    ' - UTC Position  : hhmmss.sss
'   Substr = ","
    Length = Len(sysSTR)
    Pos_01 = Instr(sysSTR , subSTR )
'   Pos_02 = Pos_01 - 1
    Length = Length - Pos_01
          '     Substr = Left(sysstr , Pos_02)
    Sysstr = Right(sysSTR , Length)
    '---------------------------
'   Print "UTC Position : " ; Substr
'   Print Sysstr
    '---------------------------
    ' - Latitude      : ddmm.mmmm
'   Substr = ","
    Length = Len(sysSTR)
    Pos_01 = Instr(sysSTR, subSTR )
    Pos_02 = Pos_01 - 1
    Length = Length - Pos_01
    subSTR = Left(sysSTR , Pos_02)
    sysSTR = Right(sysSTR, Length)
    '---------------------------
    latSTR = subSTR
'   Print Sysstr
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    '---------------------------
    ' - N/S Indicator : N=north or south
    subSTR = ","
    Length = Len(sysSTR)
    Pos_01 = Instr(sysSTR, subSTR)
'   Pos_02 = Pos_01 - 1
    Length = Length - Pos_01
'   Substr = Left(sysstr , Pos_02)
    sysSTR = Right(sysSTR , Length)
    '---------------------------
'   Print "N/S : " ; Substr
'   Print Sysstr
    '---------------------------
    ' - Longitude :dddmm.mmmm
'   Substr = ","
    Length = Len(sysSTR)
    Pos_01 = Instr(sysSTR , subSTR )
    Pos_02 = Pos_01 - 1
    Length = Length - Pos_01
    subSTR = Left(sysSTR , Pos_02)
    sysSTR = Right(sysSTR , Length)
    '---------------------------
    Lonstr = subSTR
'   Print Sysstr
    '---------------------------
  End If
End Sub
'-------------------------------------------------------------------------------
Sub GPRMC
  subSTR = "$GPRMC"
  Print "GPRMC:";
  Reset GPTXA
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Input sysSTR Noecho Timeout = 100000
Set GPTXA
If sysSTR = "" Then
  Print "GPS not response"
  CLR  PBRES
  Waitms 20
  Print "GPS Reset"
  SETB PBRES
  If SND.6 = 1 Then Sound Buzzer , 1000 , 50
  Goto Begin
End If
Pos_01 = Instr(sysSTR , subSTR )
If Pos_01 <> 0 Then
  Print sysSTR
  ' - RMC     : Recommended minimum specific GNSS data -
  ' - $GPRMC,080309.408,A,1343.0790,N,10046.6577,E,0.43,164.9,060604,,*3F
  subSTR = ","
  ' - $GPRMC  : RMC protal header
  Length = Len(sysSTR)
  Pos_01 = Instr(sysSTR , subSTR )
'   Pos_02 = Pos_01 - 1
  Length = Length - Pos_01
'   Substr = Left(sysstr , Pos_02)
  sysSTR = Right(sysSTR , Length)
  '---------------------------
'   Print "RMC : " ; Substr
'   Print Sysstr
  '---------------------------
  ' - UTC Position  : hhmmss.sss
'   Substr = ","
  Length = Len(sysSTR)
  Pos_01 = Instr(sysSTR , subSTR )
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
'    Pos_02 = Pos_01 - 1
     Length = Length - Pos_01
'    Substr = Left(sysstr , Pos_02)
     sysSTR = Right(sysSTR , Length)
     '---------------------------
'    Print "UTC Position : " ; Substr
'    Print Sysstr
     '---------------------------
     ' - Status  : A=data valid or V=data not valid
'    Substr = ","
     Length = Len(sysSTR)
     Pos_01 = Instr(sysSTR , subSTR )
'    Pos_02 = Pos_01 - 1
     Length = Length - Pos_01
'    Substr = Left(sysstr , Pos_02)
     sysSTR = Right(sysSTR , Length)
     '---------------------------
'    Print "Status : " ; Substr
'    Print Sysstr
     '---------------------------
     ' - Latitude     : ddmm.mmmm
'    Substr = ","
     Length = Len(sysSTR)
     Pos_01 = Instr(sysSTR , subSTR )
     Pos_02 = Pos_01 - 1
     Length = Length - Pos_01
     subSTR = Left(sysSTR , Pos_02)
     sysSTR = Right(sysSTR , Length)
     '---------------------------
     Latstr = subSTR
'    Print Sysstr
     '---------------------------
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    ' - N/S Indicator : N=north or south
    subSTR = ","
    Length = Len(sysSTR)
    Pos_01 = Instr(sysSTR , subSTR )
'   Pos_02 = Pos_01 - 1
    Length = Length - Pos_01
'   Substr = Left(sysstr , Pos_02)
    sysSTR = Right(sysSTR , Length)
    '---------------------------
'   Print "N/S : " ; Substr
'   Print Sysstr
    '---------------------------
    ' - Longitude :dddmm.mmmm
'   Substr = ","
    Length = Len(sysSTR)
    Pos_01 = Instr(sysSTR , subSTR )
    Pos_02 = Pos_01 - 1
    Length = Length - Pos_01
    subSTR = Left(sysSTR , Pos_02)
    sysSTR = Right(sysSTR , Length)
    '---------------------------
    lonSTR = subSTR
'   Print Sysstr
    '---------------------------
    ' - E/W Indicator : E=east or W=west
    subSTR = ","
    Length = Len(sysSTR)
    Pos_01 = Instr(sysSTR , subSTR )
'   Pos_02 = Pos_01 - 1
    Length = Length - Pos_01
'   Substr = Left(sysstr , Pos_02)
    sysSTR = Right(sysSTR , Length)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    '---------------------------
'   Print "E/W : " ; Substr
'   Print Sysstr
    '---------------------------
    ' - Speed over Ground :knots
'   Substr = ","
    Length = Len(sysSTR)
    Pos_01 = Instr(sysSTR , subSTR )
    Pos_02 = Pos_01 - 1
    Length = Length - Pos_01
    subSTR = Left(sysSTR , Pos_02)
    sysSTR = Right(sysSTR , Length)
    '---------------------------
    Speed = Val(subSTR)
    Speed = Speed * 1.852
'   Print Sysstr
    '---------------------------
    ' - Course over Ground :knots
    subSTR = ","
    Length = Len(sysSTR)
    Pos_01 = Instr(sysSTR , subSTR )
    Pos_02 = Pos_01 - 1
    Length = Length - Pos_01
    subSTR = Left(sysSTR , Pos_02)
    sysSTR = Right(sysSTR , Length)
    '---------------------------
    Course = subSTR
'   Print Sysstr
    '---------------------------
  End If
End Sub
'-------------------------------------------------------------------------
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Sub Lcd_initial
' LCD Initailize
  Call Lcd_ctrl(&B00000001)          'Clear LCD
  Call Lcd_ctrl(&B00000010)          'SET DD RAM : Return to Home
  Call Lcd_ctrl(&B00111000)          'Function Set
  Call Lcd_ctrl(&B00001100)          'Display ON/OFF
  Call Lcd_ctrl(&B00010100)          'Cursor or Display Shift
  Call Lcd_ctrl(&B00000110)          'Entry Mode Set
End Sub
'-------------------------------------------------------------------------
Sub Lcd_ctrl(lcd_data As Byte)
  E = 1
  Rs = 0
  P2 = Lcd_data
  Waitms 1
  E = 0
  Waitms 1
End Sub
'-------------------------------------------------------------------------
Sub Lcd_data(lcd_data As Byte)
  E = 1
  Rs = 1
  P2 = Lcd_data
  Waitms 1
  E = 0
  Waitms 1
End Sub
'-------------------------------------------------------------------------
Sub Lcd_wstr(sysstr As String * 80 , Line As Byte)
  If Line = 2 Then Call Lcd_ctrl(&B11000000) Else Call Lcd_ctrl(&B00000010)
  Length = Len(sysstr)
  For K = 1 To Length
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    Onechr = Mid(sysstr , K , 1)
    Lcd_data = Asc(onechr)
    Call Lcd_data(lcd_data)
  Next
End Sub
'-------------------------------------------------------------------------------
Sub Lcd_append(sysstr As String * 80)
  Length = Len(sysstr)
  For K = 1 To Length
    Onechr = Mid(sysstr , K , 1)
    Lcd_data = Asc(onechr)
    Call Lcd_data(lcd_data)
  Next
End Sub
'-------------------------------------------------------------------------------
End
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

โครงงานนี้ได้รับความช่วยเหลือจาก รศ.ดร.มนัส สังวรศิลป์ ซึ่งเป็นอาจารย์ที่ปรึกษา ขอขอบคุณ คุณกิตติ เปรมพิจิต เอื้อเฟื้อซอร์ฟแวร์วิเคราะห์การเดินทาง ขอขอบคุณ บริษัท รอมบิกซ์ จำกัด ที่อบรม ฝึกทักษะในเรื่องของไมโครคอนโทลเลอร์

(.....)

นายวัฒณสิทธิ์ พิมเพา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

1. อารัมภีย์ จันทร์ไย , " เซมิคอนดักเตอร์อิเล็กทรอนิกส์ " , ซีเอ็ดยูเคชั่น , 2546
2. ผศ.ธีรวัฒน์ ประกอบผล , " การพัฒนาไมโครคอนโทรลเลอร์ด้วยภาษาซี " , สมาคมส่งเสริมเทคโนโลยี (ไทย-ญี่ปุ่น) , 196 หน้า , 2545

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้