สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

ระบบขอความช่วยเหลือโดยการระบุตำแหน่งด้วย GPS

Rescue Positioning System By GPS

โดย

นาย ฉัฐยศ นรเศรษฐกมล

นาย พลานนท์ จิตต์สำรวย

นางสาว ศิวาพร คมสัน

วพ.
ร 2915
2549

เลขหมู่.....
เลขทะเบียน 72971
วัน,เดือน,ปี 26 ส.ย. 2550

b 11424949
i.....

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมโทรคมนาคม

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2549

ภาควิชา
วิศวกรรมโทรคมนาคม

ผ่านการตรวจชั้นงานแล้ว
(ลงชื่อ).....ผู้ตรวจ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระบบขอความช่วยเหลือโดยการระบุตำแหน่งด้วย GPS

Rescue Positioning System By GPS

โดย

นาย ฉัฐยศ นรเศรษฐกมล 46012006

นาย พลานนท์ จิตต์สำรวย 46012023

นางสาว ศิวาพร คมสัน 46012030

อาจารย์ที่ปรึกษา

รศ.ดร. ปราโมทย์ วาดเขียน

ผศ.ดร. จีรสุดา โกษียาภรณ์

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมโทรคมนาคม

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2549

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์ปีการศึกษา 2549

ภาควิชาวิศวกรรมโทรคมนาคม

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง **ระบบขอความช่วยเหลือโดยการระบุตำแหน่งด้วย GPS**

Rescue Positioning System By GPS

ผู้จัดทำ

1. นาย ฉัฐยศ นรเศรษฐกมล 46012006

2. นาย พฉานนท์ จิตต์สำรวย 46012023

3. นางสาว ศิวาพร คมสัน 46012030

อาจารย์ที่ปรึกษา

(รศ.ดร.ปราโมทย์ วาดเขียน)

อาจารย์ที่ปรึกษา

(ผศ.ดร.จีรสุดา โกษียาภรณ์)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระบบขอความช่วยเหลือโดยการระบุตำแหน่งด้วย GPS

Rescue Positioning System By GPS

โดย นาย ฉัฐยศ นรเศรษฐกมล 46012006
นาย พลานนท์ จิตต์สำรวย 46012023
นางสาว ศิวาพร คมสัน 46012030

อาจารย์ที่ปรึกษา รศ.ดร.ปราโมทย์ วาดเขียน
ผศ.ดร.จีรสุดา โกษียาภรณ์

บทคัดย่อ

โครงงานนี้นำเสนอระบบขอความช่วยเหลือโดยการระบุตำแหน่งด้วย GPS (Global Positioning System) ซึ่งเป็นการนำระบบ GPS มาประยุกต์ใช้ซึ่งจะทำงานโดยเมื่อผู้ป่วยกดสวิตซ์ขอความช่วยเหลือ ระบบ GPS จะนำค่าตำแหน่งที่ได้จากเครื่องรับสัญญาณ GPS เข้าสู่โทรศัพท์เคลื่อนที่ เพื่อนำค่าตำแหน่งและข้อมูลเบื้องต้นของผู้ป่วยส่งผ่านทางเครือข่าย GPRS ไปยังเครื่องคอมพิวเตอร์ในโรงพยาบาลหรือสถานรักษาของผู้ป่วยนั้น เพื่อแสดงตำแหน่งของผู้ร้องขอ

ABSTRACT

This project presents rescue position system by applying GPS (Global Positioning System).This system is operated when the patient presses the rescue button. The GPS system will send the position received from the GPS receiver to mobile phone. This information will be automatically transmitted via GPRS network to computer at hospital for displaying position of requested patient.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

คณะผู้จัดทำงานโครงการใคร่ขอแสดงความขอบคุณต่อ รศ.ดร. ปราโมทย์ วาดเขียน และ ดร. จีรสุดา โกษียาภรณ์ อาจารย์ที่ปรึกษาโครงการ ผู้ให้คำปรึกษาแนวทางแก้ไขปัญหา และข้อเสนอแนะต่างๆ ในการจัดทำโครงการนี้ให้สำเร็จลุล่วงด้วยดี พร้อมกันนี้ ใคร่ขอขอบคุณต่อ

คุณชนะรัตร์ สุมาลัย ที่ได้ให้คำปรึกษาในด้านการเขียนโปรแกรมไมโครคอนโทรลเลอร์

คุณปัญญา จิรจราชีพ ที่ได้ให้คำปรึกษาในการใช้เครื่องมือ และอุปกรณ์

ขอขอบพระคุณผู้ที่สนใจโครงการนี้ทุกท่าน ซึ่งเป็นแรงผลักดันอย่างดียิ่ง ที่ทำให้มีกำลังใจที่จะพัฒนาโครงการนี้ให้ดียิ่งๆ ขึ้นไป และพร้อมกันนี้ ขอขอบคุณต่อผู้ที่มีส่วนเกี่ยวข้องทุกๆ ท่านที่ช่วยให้โครงการนี้ สำเร็จลุล่วงไปด้วยดี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

	หน้า
บทที่ 1 บทนำ	1
บทที่ 2 ทฤษฎีและหลักการ	2
2.1 ระบบกำหนดตำแหน่งบนพื้นโลก (Global Positioning System: GPS)	2
2.2 องค์ประกอบหลักของ GPS	3
2.2.1 ส่วนอวกาศ (Space segment)	3
2.2.2 ส่วนสถานีควบคุม (Control segment)	6
2.2.3 ส่วนผู้ใช้ (User segment)	7
2.3 หลักการสำรวจหาตำแหน่งบนโลกด้วยดาวเทียม	8
2.3.1 หน้าที่สำคัญ ของดาวเทียม GPS	9
2.3.2 การคำนวณตำแหน่งพิกัด	9
2.3.3 วิธีการหาพิกัดตำแหน่งโดยใช้ระบบกำหนดตำแหน่งบนโลก	9
2.4 ความถูกต้อง (ACCURACY)	10
2.4.1 ความคลาดเคลื่อนในงานรังวัดดาวเทียม ระบบกำหนดตำแหน่งบนโลก	11
2.5 แนวโน้มการพัฒนาระบบกำหนดตำแหน่งบนโลกด้วยดาวเทียม	12
2.6 โครงสร้างของข้อมูลที่ส่งจากดาวเทียม	14
2.7 หลักการของระบบ DGPS (Differential GPS)	15
2.8 มาตรฐาน NMEA 0183	16
2.9 การอ้างพิกัดแบบ UTM	18
2.10 TCP/IP Protocol	25
2.10.1 Encapsulation/Demultiplexing	25
2.11 มาตรฐานการเชื่อมต่อทางคลื่นวิทยุของเครือข่าย GPRS	32
2.12 กระบวนการแสดงตัวต่อเครือข่าย GPRS	37
2.13 เครื่องลูกข่าย GPRS	43
2.13.1 เครื่องลูกข่าย	41
2.13.2 เครือข่ายสถานีฐาน	42
2.14 ทฤษฎีเกี่ยวกับไมโครคอนโทรลเลอร์และการประยุกต์ใช้งาน	43
2.14.1 คุณสมบัติของไมโครคอนโทรลเลอร์ตระกูล MCS-51	43
2.14.2 การจัดเรียงขาและอธิบายรายระเอียดของขาต่างๆของ ไมโครคอนโทรลเลอร์ตระกูล MCS – 51	44
2.15 หน่วยความจำภายใน Internal RAM	48

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ(ต่อ)

	หน้า
2.16 เว็บเซอร์วิส (Web service)	51
2.16.1 พื้นฐานของ Web Service	52
2.17 ความเป็นมาของ XML (Extensive Markup Language)	53
บทที่ 3 การออกแบบและการสร้าง	56
3.1 โครงสร้างการทำงานของระบบที่ออกแบบ	55
3.2. เครื่องรับระบบพิกัดตำแหน่ง (GPS Receiver)	57
3.3 การต่อใช้งานโมดูล VT002p	58
3.4 วงจร TCP/IP Stack	60
3.4.1 AT-COMMAND	61
3.4.2 AT command ที่เกี่ยวข้องกับการเชื่อมต่อ ของชิป TNS010i	62
3.5 วงจรไมโครคอนโทรลเลอร์	63
3.6 วงจร Regulator	62
3.7 การสร้างและการออกแบบโปรแกรม	64
3.8 การออกแบบโปรแกรม Test latitude longitude	66
3.9 การออกแบบ Form โปรแกรม Test latitude longitude	67
3.10 การออกแบบโปรแกรม GPSTrackingMobile เพื่อใช้ทดสอบการส่งพิกัดละติจูด ลองติจูด จากโทรศัพท์มือถือผ่านเครือข่าย GPRS	68
3.11 การติดตั้งโปรแกรม Position Tracking	69
บทที่ 4 ผลการทดลอง	72
4.1 ทดสอบการรับค่าจากโมดูลเครื่องรับ GPS	72
4.2 การเชื่อมต่อโทรศัพท์มือถือกับ Computer	73
4.3 ทดสอบการทำงานของวงจร TCP/IP Stack	75
4.4 การจำลองการทำงานโปรแกรมไมโครคอนโทลเลอร์	76
4.5 การทดลองการพล็อตค่าตำแหน่งละติจูด ลองติจูด โดยเป็นการส่งข้อมูลระหว่าง Computer กับ Computer	77
4.6 ผลการส่งค่าพิกัดละติจูด ลองติจูด ทางโทรศัพท์มือถือผ่านเครือข่าย GPRS	79
บทที่ 5 สรุป	82
ภาคผนวก ก.	
กิตติกรรมประกาศ	
หนังสืออ้างอิง	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

	หน้า
ตารางที่ 2.1 ความหมายของข้อมูลที่อยู่ในประโยค GGA	16
ตารางที่ 2.2 ความหมายของข้อมูลที่อยู่ในประโยค GLL	17
ตารางที่ 2.3 ความหมายของข้อมูลที่อยู่ในประโยค GSA	18
ตารางที่ 2.4 ความหมายของข้อมูลที่อยู่ในประโยค GSV	19
ตารางที่ 2.5 ความหมายของข้อมูลที่อยู่ในประโยค RMC	20
ตารางที่ 2.6 ความหมายของข้อมูลที่อยู่ในประโยค VTC	21
ตารางที่ 2.7 สรุปคุณสมบัติของ 6 ประโยคหลักใน NMEA message	22
ตารางที่ 2.8 รายละเอียดของการจัดกลุ่มเครื่องลูกข่าย GPRS ออกตามคลาส การใช้งานหลายช่องเวลา	40
ตารางที่ 3.1 ขาต่อใช้งาน VT002p	57

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป

	หน้า
รูปที่ 2.1 แสดงเส้นทางการโคจรของดาวเทียม GPS รอบโลก	3
รูปที่ 2.2 แสดงเส้นทางการโคจรของดาวเทียม GPS เมื่อพิจารณาจากตำแหน่งใดๆบนพื้นโลก	4
รูปที่ 2.3 สัญญาณข้อมูลจากระบบ Global Positioning System	5
รูปที่ 2.4 แสดงการส่งสัญญาณจากสถานีภาคพื้นดินไปควบคุมดาวเทียม GPS	5
รูปที่ 2.5 แสดงตำแหน่งสถานีภาคพื้นดินที่ทำการตรวจดูและควบคุมดาวเทียม GPS	6
รูปที่ 2.6 แสดงตัวอย่างการใช้งาน GPS ในยานพาหนะต่างๆ	7
รูปที่ 2.7 โครงสร้างของข้อมูลที่ส่งมากจากดาวเทียม GPS	14
รูปที่ 2.8 รายระเอียดการแบ่งเฟรม	14
รูปที่ 2.9 แผนผังแสดงองค์ประกอบของระบบ DGPS	15
รูปที่ 2.10 การแบ่งพื้นที่ออกเป็นโซน (Zone) สำหรับการอ้างพิกัดแบบ UTM	23
รูปที่ 2.11 ขั้นตอนการ Encapsulation และ Demultiplexing	25
รูปที่ 2.12โครงสร้าง TCP/IP	26
รูปที่ 2.13 IP Header	27
รูปที่ 2.14 ICMP Header	28
รูปที่ 2.15 UDP Header	29
รูปที่ 2.16 TCP Header	30
รูปที่ 2.17 รายละเอียดของจุดเชื่อมต่อและโปรโตคอลที่เกี่ยวข้อง ระหว่างอุปกรณ์ภายในเครือข่ายสถานีฐานของมาตรฐานโทรศัพท์เคลื่อนที่GPRS	33
รูปที่ 2.18 การจัดสรรและบริหารจัดการช่องเวลาตามมาตรฐานโทรศัพท์เคลื่อนที่ GSM ให้รองรับการสื่อสารด้วยมาตรฐาน GSM โดยเป็นไปในลักษณะการร่วมใช้ความถี่เดียวกัน	34
รูปที่ 2.19 การรับส่งข้อมูลผ่านเครือข่าย GPRS สู่เครือข่ายคอมพิวเตอร์ภายนอก	38
รูปที่ 2.20 โครงสร้างการจัดโปรโตคอลภายในเครือข่าย GPRS	42
รูปที่ 2.21 แสดงตำแหน่งขาต่างๆของชิปไมโครคอนโทรลเลอร์ตระกูล MCS – 51	48
รูปที่ 2.22 การใช้งานขา Reset	45
รูปที่ 2.23 แสดงการใช้งานขา Reset	46
รูปที่ 2.24 แสดงการแบ่งประเภทหน่วยความจำภายใน	48
รูปที่ 3.1 โครงสร้างการทำงานของระบบ	56
รูปที่ 3.2 เครื่องรับ GPS ที่เลือกใช้รุ่น VT002p	57
รูปที่ 3.3วงจรจ่ายไฟโมดูล GPS	58
รูปที่ 3.4 ผังการทำงานในส่วนของโปรแกรมทั้งหมด	59
รูปที่ 3.5 แสดงโครงสร้างของชิป TNS 010i	60

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป(ต่อ)

	หน้า
รูปที่ 3.6 โครงสร้างการเชื่อมต่อของวงจร	60
รูปที่ 3.7 การต่อใช้งานของวงจร TCP/IP Stack	61
รูปที่ 3.8 วงจรไมโครคอลโทรลเลอร์	62
รูปที่ 3.9 การต่อใช้งานไมโครคอนโทรลเลอร์	63
รูปที่ 3.10 วงจร Regulator	63
รูปที่ 3.11 แสดงการทำงานของโปรแกรม Position tracking	64
รูปที่ 3.12 แสดงให้เห็นหน้าต่าง Form โปรแกรมที่มี Tool ต่างๆประกอบอยู่	65
รูปที่ 3.13 แสดงการทำงานของโปรแกรม Test latitude longitude	66
รูปที่ 3.13 แสดงการทำงานของโปรแกรม Test latitude longitude	67
รูปที่ 3.15 แสดงการทำงานของโปรแกรม Test latitude longitude	68
รูปที่ 3.16 แสดงหน้าต่าง Web Server ที่เราทำการเรียก	69
รูปที่ 3.17 แสดงโครงสร้าง Path ตั้งต้นของ IIS	70
รูปที่ 3.18 แสดงหน้าต่าง Default Web Site Properties เพื่อทำการเซทค่าต่างๆของ IIS	71
รูปที่ 4.1 ผลการรับข้อมูลจากเครื่องรับ GPS	63
รูปที่ 4.2 ผลการรับข้อมูลจากเครื่องรับ GPS ผ่านทาง HyperTerminal	73
รูปที่ 4.3 โทรศัพท์มือถือรุ่น Siemens S45	73
รูปที่ 4.4 ผลการทดสอบคำสั่งพื้นฐานโทรศัพท์มือถือ	74
รูปที่ 4.5 แสดงผลการทำงานของวงจร TCP/IP ขณะเชื่อมต่อโทรศัพท์เคลื่อนที่	75
รูปที่ 4.6 ผลการจำลองการทำงานของโปรแกรม	76
รูปที่ 4.7 โปรแกรม Test เมื่อยังไม่มีการใส่ค่าพิกัด	77
รูปที่ 4.8 ค่าเริ่มต้นเมื่อยังไม่มีการส่งค่าตำแหน่งจากเครื่อง Client เข้ามายังเครื่อง Server	77
รูปที่ 4.9 เมื่อใส่ตำแหน่งพิกัดละติจูด ลองติจูด	78
รูปที่ 4.10 การแสดงตำแหน่งพิกัดที่เครื่อง Server ได้รับพิกัดมาจากเครื่อง Client	78
รูปที่ 4.11 แสดง Application ในการส่งพิกัดละติจูด ลองติจูด	79
รูปที่ 4.12 ตำแหน่งเริ่มต้นเมื่อยังไม่มีการส่งพิกัดตำแหน่ง	79
รูปที่ 4.13 แสดงการใส่ค่าพิกัดละติจูด ลองติจูด แล้วทำการส่งตำแหน่งพิกัดผ่านเครือข่ายGPRS	80
รูปที่ 4.14 แสดงหน้าเว็บเมื่อมีการส่งพิกัดตำแหน่ง	81
รูปที่ 4.15 แสดงตำแหน่งพิกัดที่รับมาจากโทรศัพท์มือถือผ่านเครือข่ายGPRS	81

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

GPS เป็นระบบที่สามารถระบุตำแหน่งบนพื้นผิวโลกได้อย่างแม่นยำที่สำคัญ GPS เป็นระบบเปิดที่สามารถนำไปสร้างหรือนำไปประยุกต์ใช้งานได้ตามความต้องการได้ ข้อมูลที่ได้รับจากระบบนั้นเป็นข้อมูลพิกัดตำแหน่งปัจจุบันที่อยู่ขณะนั้น ข้อมูลดังกล่าวจึงสามารถนำมาใช้ประโยชน์ได้หลากหลาย เช่น ระบบนำร่อง การสำรวจพื้นที่ และการทำแผนที่

1.1วัตถุประสงค์

1.1.1 เพื่อต้องการนำเทคโนโลยีของดาวเทียมบอกพิกัด GPS (Global Positioning Systems) มาประยุกต์ใช้ในการระบุตำแหน่งปัจจุบัน

1.1.2 เพื่อนำเทคโนโลยีของดาวเทียมบอกพิกัด GPS มาพัฒนาร่วมกับการส่งข้อมูลผ่านเครือข่าย Internet ผ่านระบบ GPRS โดยโทรศัพท์เคลื่อนที่

1.1.3 เพื่อระบุตำแหน่งของผู้ร้องขอความช่วยเหลือหรือผู้ป่วยได้อย่างถูกต้อง

1.1.4 เพื่อสามารถช่วยเหลือผู้ร้องขอความช่วยเหลือหรือผู้ป่วยได้อย่างรวดเร็ว

1.1.5 เพื่อช่วยเพิ่มความสะดวกในการหาเส้นทางเพื่อหลีกเลี่ยงการหลงทาง การไม่ทราบเส้นทางในการเดินทาง ให้โรงพยาบาลหรือสถานรักษาของผู้ป่วย สามารถเดินทางไปช่วยผู้ร้องขอได้อย่างถูกต้องและใช้เวลาน้อยที่สุด

1.2 ขอบเขตของโครงงาน

1.2.1 โครงงานสามารถแสดงตำแหน่งปัจจุบันของผู้ร้องขอความช่วยเหลือได้อย่างถูกต้อง

1.2.2 สามารถระบุจุดหมายปลายทางเพื่อทำการค้นหาเส้นทางบนแผนที่จำลองที่อยู่ในรูปแบบของแผนที่ ของโรงพยาบาล

1.3 องค์ประกอบของระบบ

ส่วนประกอบของระบบ ประกอบด้วย 5 ส่วนหลักคือ GPS Module, MCS-51, TCP/IP Chip, GPRS, Sever ซึ่งแต่ละส่วนมีหน้าที่ดังนี้

1.3.1 โมดูล GPS (GPS Module) เป็นอุปกรณ์ที่รับข้อมูลพิกัดตำแหน่งจากดาวเทียม GPS อยู่ตลอดเวลา เพื่อระบุตำแหน่งปัจจุบัน

1.3.2 วงจร MCS-51 เป็นอุปกรณ์ควบคุม สั่งงาน และเก็บค่าพิกัด เพื่อเตรียมส่งข้อมูลต่อไป

1.3.3 วงจร TCP/IP stack เป็นวงจรช่วยสนับสนุนโปรโตคอลหลายชนิด เช่น telnet, ftp, talk, rsh, และอื่น ๆ อีกมาก ไฟล์หลักสำหรับกำหนดค่าต่าง ๆ

1.3.4 ระบบ GPRS (โทรศัพท์เคลื่อนที่) เป็นอุปกรณ์ที่ใช้สำหรับส่งข้อมูลผ่านเครือข่าย Internet เข้าสู่ระบบของโรงพยาบาล

1.3.5 หน่วยประมวลผลและแสดงตำแหน่ง เป็นอุปกรณ์ที่ใช้ในการประมวลผลข้อมูลต่างๆ ที่ได้รับมาและนำไปแสดงผลบนแผนที่จำลองของโรงพยาบาล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2
ทฤษฎีและหลักการ

2.1 ระบบกำหนดตำแหน่งบนพื้นโลก (Global Positioning System: GPS)

ระบบกำหนดตำแหน่งบนโลก หรือ Global Positioning System ระบบนี้ได้พัฒนาขึ้นโดยกระทรวงกลาโหม ประเทศสหรัฐอเมริกาซึ่งจัดทำโครงการ Global Positioning System มาตั้งแต่ปี พ.ศ. 2521 โดยอาศัยดาวเทียมและระบบคลื่นวิทยุนำร่องและรหัสที่ส่งมาจากดาวเทียม NAVSTAR จำนวน 24 ดวง โดยแบ่งเป็นชุด ชุดละ 4 ดวงโดยทำการโคจรอยู่รอบโลกวันละ 2 รอบ และมีตำแหน่งอยู่เหนือพื้นโลกที่ความสูง 20,200 กิโลเมตร

ระบบ GPS ถูกควบคุมโดย กองทัพอากาศ สหรัฐอเมริกา จากสถานีควบคุมหลัก ในรัฐโคโลราโด ซึ่งจะคอยตรวจสอบดาวเทียมทุกดวงในระบบ ป้อนคำสั่งควบคุม และป้อนข้อมูล รวมทั้งให้ข่าวสารในการนำร่อง สถานีตรวจสอบภาคพื้นดิน ใช้สายอากาศภาคพื้นดิน ในการควบคุม ดาวเทียม GPS และส่งต่อข้อมูลให้แก่สถานี Master Control เพื่อกำหนดตำแหน่งพิกัดที่แน่นอน ของดาวเทียมแต่ละดวง และปรับปรุงความถูกต้อง ของข้อมูลอยู่ตลอดเวลา ถ้าดาวเทียมดวงใดเกิดความผิดปกติขึ้น สถานีควบคุมภาคพื้นดิน ก็จะทำการกำหนดสุขภาพ ดาวเทียมดวงนั้นเป็น "Un- healthy" เพื่อให้ GPS Receiver ทราบว่าไม่ควรใช้ข้อมูล จากดาวเทียมดวงนี้ ซึ่งเครื่องรับ ก็จะทำการตรวจสอบได้ จากการตรวจสอบสถานะของดาวเทียม และเครื่องก็จะไม่ทำการ รับข้อมูล จากดาวเทียมดวงดังกล่าว แล้วใช้ดาวเทียมดวงอื่น ที่มีความเหมาะสมในการคำนวณตำแหน่งพิกัดแทนในบางครั้งดาวเทียมอาจถูกปิดใช้งานเพื่อทำการบำรุงรักษา หรืออาจจะถูกปิดเพื่อเปลี่ยนวงโคจรตามความเหมาะสม มีสถานีควบคุมภาคพื้นดินเพื่อให้ระบบ GPS สามารถที่จะทำงานได้ทุกสภาวะและตลอด 24 ชั่วโมง ลักษณะการทำงานในการกำหนดค่าพิกัดของระบบ GPS ทำได้ด้วยการนำเครื่องรับไปยังตำแหน่งที่ต้องการจะทราบค่าพิกัดจากนั้นเครื่องรับจะรอสัญญาณจากดาวเทียม เมื่อเครื่องรับได้สัญญาณจากจำนวนดาวเทียมที่เพียงพอ ก็จะประมวลผลสัญญาณข้อมูลที่ได้จากดาวเทียมและแสดงผลออกมาเป็นค่าพิกัดของตำแหน่งเครื่องรับ (Receiver) ซึ่งจะเสร็จสิ้นในระยะเวลาที่รวดเร็วมากเมื่อเทียบกับการรังวัดในแบบเดิม ความถูกต้องของค่าพิกัดที่ได้จากระบบ GPS จะขึ้นอยู่กับความสามารถของอุปกรณ์เครื่องรับซึ่งอาจจะมีความถูกต้องได้ตั้งแต่ 1 เซนติเมตรไปจนถึง 300 เมตรทีเดียว โดยที่กระทรวงกลาโหม สหรัฐอเมริกายังสามารถที่จะลดค่าความถูกต้องของเครื่องระบุได้อีกด้วยการส่งค่า Selective Availability (SA) ออกมาเพื่อทำให้การคำนวณค่าพิกัดคลาดเคลื่อน อย่างไรก็ตามก็ยังคงมีวิธีการที่จะแก้ไขปัญหาดังกล่าวได้ด้วยการใช้วิธี Differential Correction ซึ่งทำให้ความถูกต้องของค่าพิกัดที่ได้อยู่ในช่วง 1-5 เมตรเท่านั้น เนื่องจากการที่ระบบ GPS สามารถจัดเก็บค่าพิกัดได้ด้วยความรวดเร็ว มีความถูกต้องสูง และทำงานได้ตลอด 24 ชั่วโมงนี้เองจึงทำให้มีการนำระบบ GPS ไปใช้ในงานด้านต่างๆ อย่างกว้างขวางโดยเฉพาะอย่างยิ่งสำหรับงานทางด้านแผนที่และงานทางด้านการสำรวจและทำให้ระบบ GPS มีความสำคัญมากขึ้น ดาวเทียมที่ใช้ในการสำรวจด้วยระบบ GPS จะมีด้วยกันทั้งหมดในปัจจุบัน 24 ดวงครบตามที่กระทรวงกลาโหม สหรัฐอเมริกากำหนดไว้ในโครงการ GPS โดยที่ดาวเทียมทั้งหมดจะโคจรครอบคลุมทั่วทั้งโลก การที่เครื่องรับสัญญาณจะสามารถที่จะกำหนดค่าพิกัด (X,Y) ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

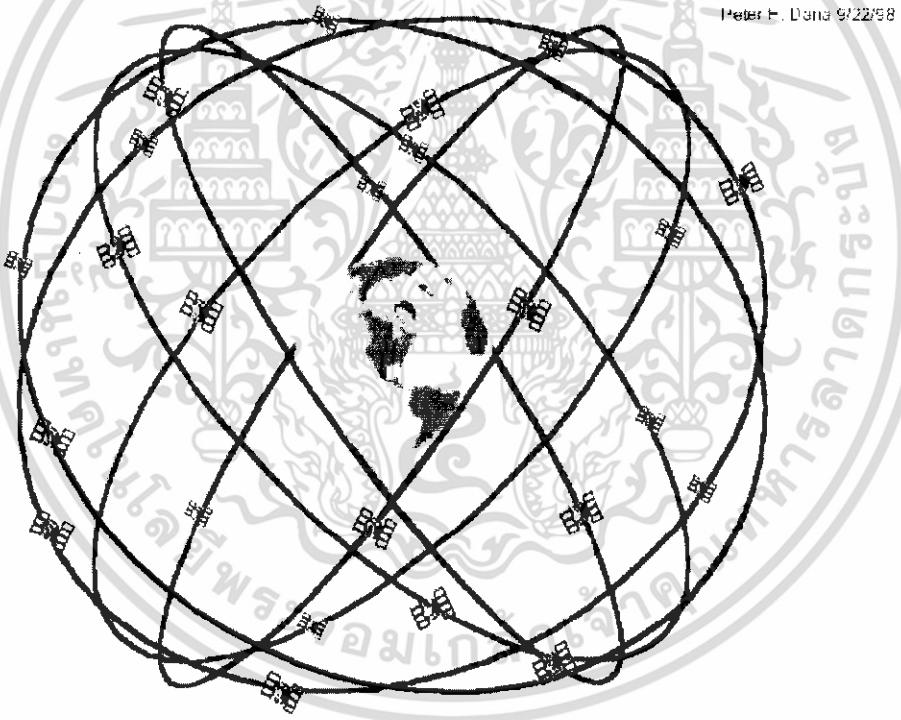
จะต้องรับสัญญาณดาวเทียมได้อย่างน้อย 3 ดวงขึ้นไป แต่ถ้ารับได้ 4 ดวงก็จะสามารถกำหนดค่าพิกัด (X,Y) พร้อมทั้งค่าความสูง (Z) ของตำแหน่งนั้นได้ด้วย

2.2 องค์ประกอบหลักของ GPS

ระบบกำหนดตำแหน่งบนโลก ประกอบด้วย 3 ส่วนหลัก คือ ส่วนอวกาศ (Space segment) ส่วนสถานีควบคุม (Control segment) และส่วนผู้ใช้ (User segment)

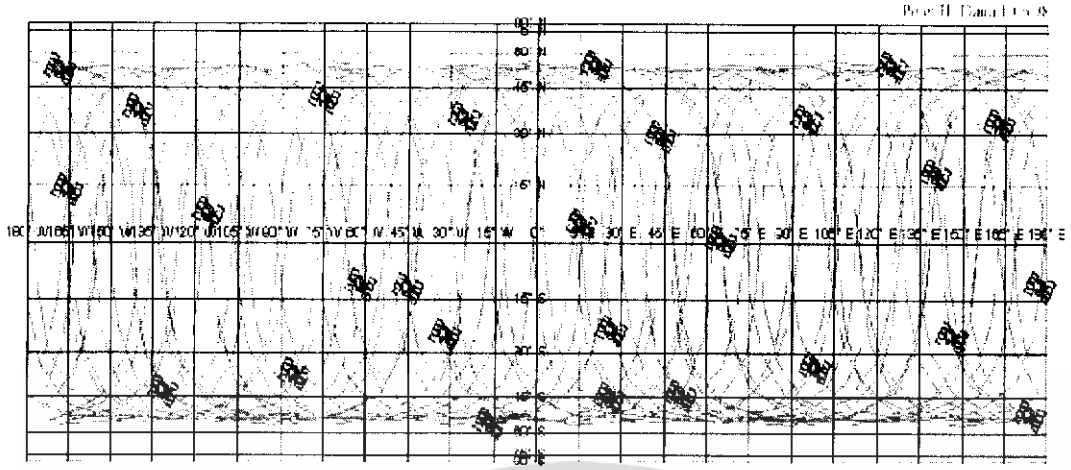
2.2.1 ส่วนอวกาศ (Space segment)

เป็นส่วนที่อยู่บนอวกาศ ประกอบด้วยดาวเทียม 24 ดวง โดยมี 21 ดวง ทำหน้าที่ส่งสัญญาณคลื่นวิทยุจากอวกาศ (Space Vehicles, SVs) ส่วนอีก 3 ดวง สำหรับเป็นดาวเทียมปฏิบัติการเสริมวงโคจร ของดาวเทียมแต่ละดวงจะใช้เวลาโคจร 12 ชั่วโมง ต่อ 1 รอบ โดยมีทั้งหมด 6 วงโคจร แต่ละวงโคจรมีดาวเทียม 4 ดวง วงโคจรมีมุมเอียง 55 องศา กับระนาบศูนย์สูตร และห่างกัน 60 องศา วงโคจรในลักษณะดังกล่าวทำให้มีดาวเทียมอย่างน้อย 4 ดวง อยู่บนท้องฟ้าทุก ๆ จุดบนพื้นผิวโลกตลอดเวลา 24 ชั่วโมง



รูปที่ 2.1 แสดงเส้นทางการโคจรของดาวเทียม GPS รอบโลก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



Global Positioning System Satellites and Orbits
for 27 Operational Satellites on September 29, 1998
Satellite Positions at 00:00:00 9/29/98 with 24 Hours (2 orbits) of Ground Tracks to 00:00:00 9/30/98

รูปที่ 2.2 แสดงเส้นทางการโคจรของดาวเทียม GPS เมื่อพิจารณาจากตำแหน่งใดๆบนพื้นโลก

ดาวเทียมชุดแรกเรียก GPS Block 1 มีทั้งหมด 10 ดวง ดาวเทียม NAVSTAR (Navigation Satellite Timing and Ranging) เป็นดวงแรกที่ส่งขึ้นใน พ.ศ. 2521 อีก 9 ดวง ต่อมาส่งใน พ.ศ. 2531 ดาวเทียมส่วนที่เหลือได้ถูกส่งขึ้นไปในช่วง พ.ศ. 2532 – 2536 จนครบ 24 ดวง ใน พ.ศ. 2537 ดาวเทียมแต่ละดวงจะมีนาฬิกาที่มีความแม่นยำสูง ซึ่งจะช่วยในการคำนวณระยะทางระหว่างดาวเทียมกับเครื่องรับสัญญาณ เพื่อที่จะคำนวณค่าพิกัดตำแหน่งนาฬิกา ชุดนี้ยังใช้ในการควบคุมความถี่ของคลื่นส่ง และรหัสที่ใช้ในระบบดาวเทียม GPS ส่งออกมาเป็นคลื่นในช่วงที่เรียกว่า L band มี 2 ความถี่ คือ

- คลื่น L1 ที่ความถี่ 1575.42 MHz เป็น 154 เท่าของความถี่พื้นฐาน ความยาวคลื่นเป็น 19 ซม.
- คลื่น L2 ที่ความถี่ 1227.60 MHz เป็น 120 เท่าของความถี่พื้นฐาน ความยาวคลื่นเป็น 24 ซม.

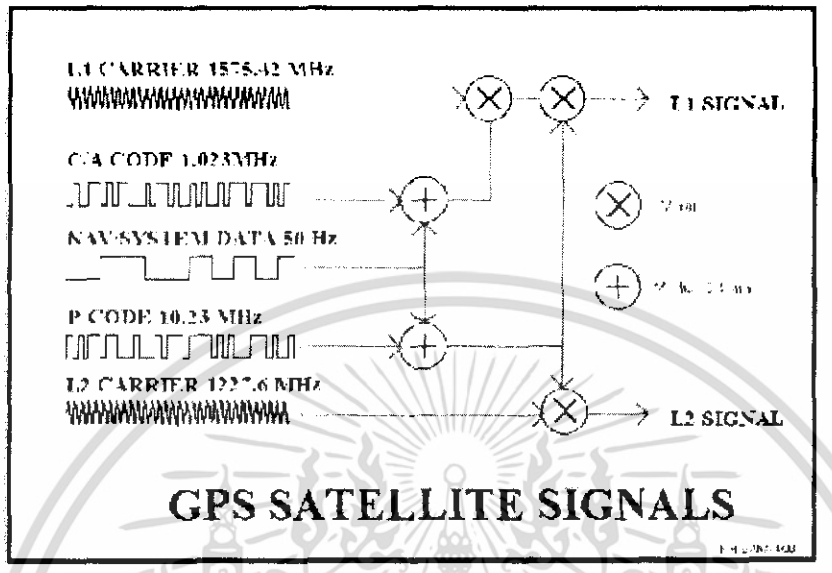
สัญญาณรหัสแฝงประกอบด้วย

- รหัส C/A มีความถี่เป็น 1/10 ของความถี่พื้นฐาน คือ 1.023 MHz ความยาวคลื่นเป็น 300 เมตร มีคาบเป็น 1 ใน 1,000 วินาที นั่นคือในช่วงเวลา 1 วินาที จะสร้างรหัส C/A ที่มีรูปแบบเหมือนกันซ้ำถึง 1,000 ครั้ง การตรวจสอบรูปแบบของรหัส C/A จึงทำได้ง่าย และรวดเร็วมาก รหัส C/A จึงทำได้ง่าย และรวดเร็วมาก รหัส C/A เปิดให้ทุกคนใช้ได้อย่างอิสระ

- รหัส P มีความถี่เท่ากับความถี่พื้นฐานคือ 10.23 MHz ความยาวคลื่นเป็น 30 เมตร และมีคาบเป็น 267 วัน นั่นคือ ในช่วง 267 วัน รหัส P ที่ส่งออกมาจะมีรูปแบบที่ไม่ซ้ำเลย จึงเป็นการยากที่จะตรวจสอบว่ารหัส P ที่ดาวเทียมใช้ในแต่ละวันเป็นส่วนไหนของรหัส ผู้ที่ไม่มีข้อมูลเกี่ยวกับการสร้างรหัส P ของดาวเทียม จึงไม่อาจใช้ประโยชน์จากรหัส P เพื่อหาตำแหน่งได้ รหัส P จะถูกสงวนไว้ใช้เฉพาะวงการทหาร และบางหน่วยงานของรัฐบาลสหรัฐอเมริกา รวมทั้งพันธมิตรทางทหารของสหรัฐอเมริกาเท่านั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

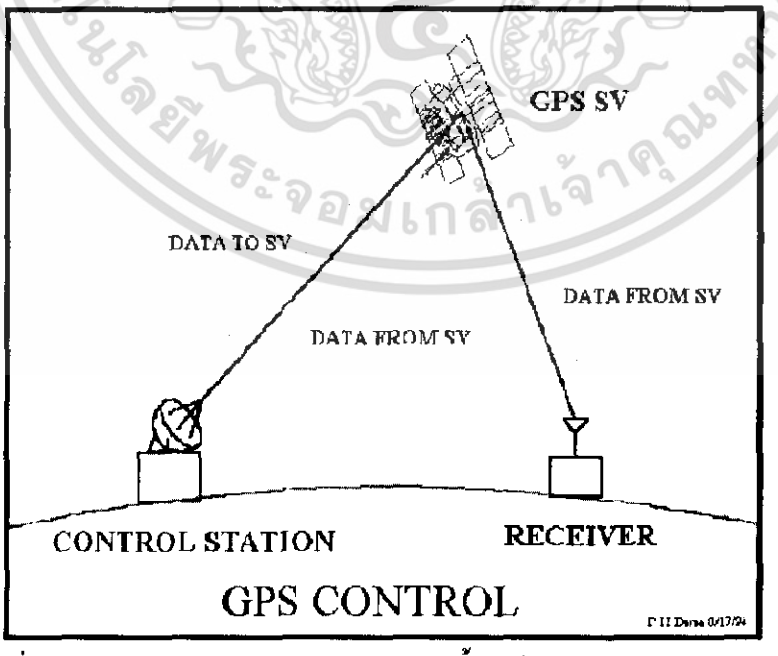
- สัญญาณ Navigation จะมอดูเลตร่วมกับสัญญาณ L1 และ C/A โดยมีความถี่ 50 Hz สัญญาณนี้จะประกอบด้วยข้อมูลสำหรับอธิบายถึงวงโคจรของดาวเทียม การแก้ไขความคลาดเคลื่อนของสัญญาณ Clock และข้อมูลที่จำเป็นอื่น ๆ ในระบบ GPS



รูปที่ 2.3 สัญญาณข้อมูลจากระบบ Global Positioning System

2.2.2 ส่วนสถานีควบคุม (Control segment)

เพื่อให้การทำงานของระบบและค่าที่ได้รับจากอุปกรณ์ GPS เป็นไปอย่างถูกต้องจึงจำเป็นจะต้องมีสถานีควบคุมการทำงานของดาวเทียมที่ใช้งาน โดยส่วนควบคุมนี้ประกอบด้วยสถานีติดตามภาคพื้นดินที่กระจายตัวอยู่ตามจุดต่างๆของพื้นโลก



รูปที่ 2.4 แสดงการส่งสัญญาณจากสถานีภาคพื้นดินไปควบคุมดาวเทียม GPS

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ประกอบไปด้วยสถานีภาคพื้นดินที่ควบคุมระบบ (Operational Control System: OCS) ที่กระจายอยู่ตามส่วนต่าง ๆ ของโลก โดยแบ่งออกเป็น

- สถานีควบคุมหลัก ตั้งอยู่ที่ฐานทัพอากาศในเมืองโคโลราโดปสริงส์ (Colorado Springs) มลรัฐโคโรลาโดของสหรัฐอเมริกา
- สถานีติดตามดาวเทียม 5 แห่ง ทำการรังวัดติดตามดาวเทียมตลอดเวลา โดยตั้งอยู่ที่
  - หมู่เกาะฮาวาย (Hawaii) ในมหาสมุทรแปซิฟิก
  - หมู่เกาะอัสเซนชัน (Ascension) มหาสมุทรแอตแลนติก
  - หมู่เกาะดิเอโกการ์เซีย (Diego Garcia) มหาสมุทรอินเดีย
  - หมู่เกาะควาจาเลียน (Kwajalein) ประเทศฟิลิปปินส์
  - เมืองโคโลราโดสปริงส์, สหรัฐอเมริกา
- สถานีรับส่งสัญญาณ 3 แห่ง
  - หมู่เกาะควาจาเลียน
  - หมู่เกาะดิเอโกการ์เซีย
  - หมู่เกาะอัสเซนชัน

รูปที่ 2.5 แสดงตำแหน่งสถานีภาคพื้นดินที่ทำการตรวจดูและควบคุมดาวเทียม GPS

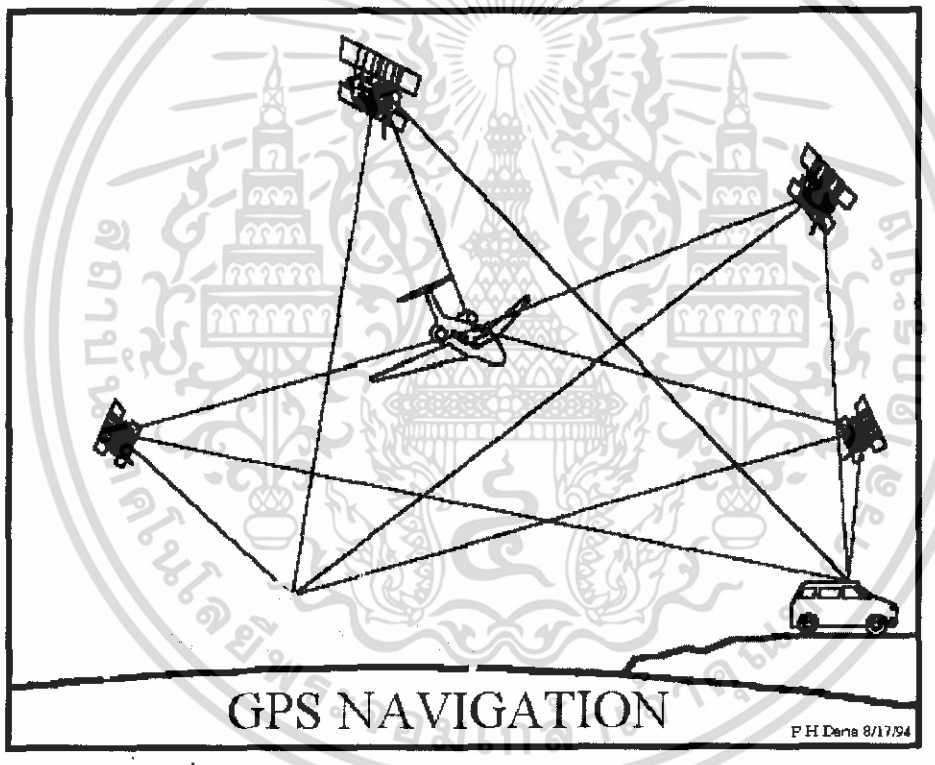
สถานีภาคพื้นดินที่ควบคุมระบบจะเฝ้าระวังติดตามดาวเทียม และจากข้อมูลที่ได้จากการเฝ้าระวังติดตามดาวเทียมสามารถที่จะบอกวงโคจรล่วงหน้าได้อย่างถูกต้องแม่นยำและจะส่งสัญญาณข้อมูลวงโคจรจากสถานีสู่ดาวเทียมวันละ 3 ครั้ง และกระจายข้อมูลวงโคจรเหล่านั้น จากดาวเทียมสู่เครื่องรับสัญญาณ GPS ได้ การรับสัญญาณนี้เรียกว่า "broadcast ephemeris" ซึ่งจะทำให้เครื่องรับสัญญาณ GPS คำนวณตำแหน่งในเวลาจริงได้ สำหรับการสำรวจในทาง geodetic อาจจะไม่ละเอียดเพียงพอ อย่างไรก็ตามก็ไม่มีความจำเป็นที่จะต้องคำนวณตำแหน่งในเวลาจริง แต่อาจสามารถทำได้ด้วยวิธีที่เรียกว่า precise ephemeris ซึ่งจะให้ข้อมูลเกี่ยวกับวงโคจรภายหลัง ที่ละเอียดกว่า การติดตามเฝ้าระวังไม่เพียงแต่ใช้สถานี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

OCS ทั้ง 5 สถานี แต่ใช้สถานีเครือข่ายอื่น ๆ Cooperative International GPS Network: (CIGNET) ซึ่งมีอยู่ทั้งสิ้น 29 สถานีกระจายทั่วโลก

2.2.3 ส่วนผู้ใช้ (User segment)

ประกอบด้วยเครื่องรับสัญญาณ หรือเครื่องระบบกำหนดตำแหน่งบนโลก ที่ใช้อยู่มีหลายขนาด สามารถพกพาไปได้ หรือจะติดไว้ในรถ เรือ หรือเครื่องบิน เครื่องระบบกำหนดตำแหน่งบนโลกจะทำหน้าที่ในการเปลี่ยนสัญญาณจาก SVs เป็นตำแหน่ง ความเร็ว และเวลาโดยประมาณ ถ้าหากต้องการทราบค่า X Y Z (position) และเวลาต้องใช้ดาวเทียมอย่างน้อย 4 ดวง ความถูกต้องของตำแหน่งขึ้นอยู่กับนาฬิกาและเครื่องระบบกำหนดตำแหน่งบนโลก ซึ่งอาจจะหาตำแหน่งบนโลก ซึ่งอาจจะหาตำแหน่งที่มีความผิดพลาดได้น้อยกว่า 3 ฟุต นาฬิกาที่ใช้จะมีความถูกต้องสามารถวัดได้ในเวลา 0.000000003 วินาที เวลาที่ใช้ในการอ้างอิงสำหรับดาวเทียมระบบกำหนดตำแหน่งบนโลก เรียกว่า GPS



รูปที่ 2.6 แสดงตัวอย่างการใช้งาน GPS ในยานพาหนะต่างๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3 หลักการสำรวจหาตำแหน่งบนโลกด้วยดาวเทียม

การใช้ดาวเทียม NAVSTAR เป็นตำแหน่งอ้างอิงจากการที่ถูกควบคุมและติดตามจากสถานีภาคพื้นดิน และมีวงโคจรที่แน่นอน ทำให้สามารถทราบว่าดาวเทียมแต่ละดวง ปัจจุบันอยู่ที่ตำแหน่งใด ซึ่งตำแหน่งของดาวเทียมจะพยากรณ์ไว้ล่วงหน้า เรียกฟังก์ชันการคำนวณเวลานี้ว่า อีฟีเมอริสดาวเทียม (satellite ephemeris) ซึ่งได้จากการรังวัดไปยังดาวเทียมของสถานีติดตามดาวเทียม ในการหาตำแหน่งที่ต้องการความถูกต้องสูงต้องใช้อีฟีเมอริสดาวเทียมที่ได้จากวงโคจรจริง ๆ ซึ่งจะได้ข้อมูล หลังจากที่ได้รังวัดหาตำแหน่งจากระบบกำหนดตำแหน่งบนโลกแล้ว

การหาระยะทางตำแหน่งห่างจากดาวเทียม ได้จากการวัดเวลาที่สัญญาณคลื่นวิทยุเดินทางจากดาวเทียมมาถึงเครื่องรับ GPS คูณด้วยความเร็วของแสง (300,000 กิโลเมตร/วินาที) ซึ่งจะทราบระยะเวลาคลื่นวิทยุจากดาวเทียมเดินทางมาถึงเครื่องรับโดยเปรียบเทียบเวลารหัสดาวเทียมกับรหัสเครื่องรับ

การทราบระยะทางจากดาวเทียมอย่างน้อย 3 ดวง สามารถคำนวณหาตำแหน่งได้ แต่หากต้องการทราบตำแหน่งได้ แต่หากต้องการทราบตำแหน่งที่แน่นอนต้องทราบระยะทางจากดาวเทียม 4 ดวงขึ้นไป ก็จะได้ตำแหน่งที่ถูกต้องมากขึ้น โดยจะทำให้ทราบตำแหน่งแบบ 3 มิติ คือทราบค่า X, Y, Z และทราบเวลา (T)

เครื่องรับระบบกำหนดตำแหน่งบนโลก จะรับสัญญาณคลื่นวิทยุจากดาวเทียม และสร้างรหัส C/A ขึ้นมาเปรียบเทียบกับรหัสที่ถอดได้จากสัญญาณ เมื่อเปรียบเทียบได้รหัสที่ตรงกันจะทำให้รู้เวลาที่คลื่นวิทยุใช้ในการเดินทางจากดาวเทียมมายังเครื่องรับ เมื่อนำความเร็วของคลื่นวิทยุคูณด้วยเวลาจะได้ระยะทางระหว่างดาวเทียม ซึ่งในเครื่องรับแบบนำหน เรียกว่า ซูโดเรนจ์ (pseudo range) เครื่องรับแบบรังวัดรับข้อมูลแบบ Carrier Phase

2.3.1 หน้าที่สำคัญ ของดาวเทียม GPS

1. รับข้อมูลวงโคจรที่ถูกต้องของดาวเทียม (Ephemeris Data) ที่ส่งมาจาก สถานีควบคุมดาวเทียมหลัก (Master Control Station) เพื่อส่งกระจายสัญญาณข้อมูลนี้ ลงไปยังพื้นโลก สำหรับ GPS Receiver ใช้ในการคำนวณ ระยะห่าง (Range) ระหว่างดาวเทียมดวงนั้น กับ ตัวเครื่อง GPS Receiver และตำแหน่งของดาวเทียมบนท้องฟ้า เพื่อใช้คำนวณหา ตำแหน่งพิกัด ของตัวเครื่อง GPS Receiver เอง

2. ส่งรหัส (Code) และข้อมูล (Carrier Phase) ไปกับคลื่นวิทยุ ลงไปยังพื้นโลก สำหรับ GPS Receiver ใช้ในการคำนวณ ระยะห่าง (Range) ระหว่างดาวเทียมดวงนั้น กับ ตัวเครื่อง GPS Receiver

3. ส่งข้อมูลตำแหน่งโดยประมาณของดาวเทียมทั้งหมด (Almanac Information) และข้อมูลสุขภาพ ของดาวเทียม ลงไปยังพื้นโลก สำหรับ GPS Receiver ใช้ในการกำหนดดาวเทียม ที่จะสามารถรับสัญญาณได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3.2 การคำนวณตำแหน่งพิกัด

ดาวเทียม GPS แต่ละดวงจะส่งกระจายสัญญาณ 2 ชนิดอย่างต่อเนื่องได้แก่ สัญญาณ Standard Positioning Service (SPS) ซึ่งใช้สำหรับบุคคลทั่วไป และสัญญาณ Precise Positioning Service (PPS) ซึ่งใช้สำหรับทางทหาร สัญญาณ SPS เป็นสัญญาณแบบ Spread-Spectrum ที่กระจายสัญญาณด้วยความถี่ 1575.42 MHz สภาพแวดล้อม หรือสัญญาณรบกวนที่เกิดจากอุปกรณ์ไฟฟ้า บนพื้นโลก มีผลกระทบค่อนข้างน้อย ต่อสัญญาณดังกล่าว สัญญาณ SPS ประกอบด้วยข้อมูลเกี่ยวกับวงโคจร ของดาวเทียม 2 ชนิด คือ ข้อมูล Almanac และข้อมูล Ephemeris ข้อมูล Almanac เป็นข้อมูลที่บอกถึงสภาพของดาวเทียม และตำแหน่งวงโคจรของดาวเทียมทุกดวงในระบบอย่างคร่าวๆ เครื่องรับ GPS จะรับข้อมูล Almanac จากดาวเทียมดวงใดๆที่สามารถรับสัญญาณได้ แล้วใช้ข้อมูลดังกล่าวเพื่อการเลือกรับดาวเทียม ที่สามารถจะใช้ได้ ในการคำนวณตำแหน่งพิกัด ส่วนข้อมูล Ephemeris ประกอบด้วยข้อมูล ที่แม่นยำ โดยละเอียดของวงโคจรของดาวเทียม แต่ละดวง ที่ทำการรับสัญญาณได้ สัญญาณ SPS จะส่งรหัส (Code) ลงมาด้วย โดยรหัสดังกล่าว จะทำให้ GPS Receiver สามารถคำนวณ เวลาที่สัญญาณ เดินทางจากดาวเทียม มาถึงตัวเครื่อง GPS Receiver ได้ เมื่อเครื่องทราบเวลาที่เดินทาง และตำแหน่งดาวเทียม (Ephemeris) ก็จะสามารถคำนวณหา ระยะ (Pseudo range) ระหว่างดาวเทียม แต่ละดวง กับ GPS Receiver ได้ เครื่องรับจะทำการรับสัญญาณจากดาวเทียมอย่างน้อย 3 ถึง 4 ดวงในเวลาเดียวกัน เครื่องจะใช้ดาวเทียม 3 ดวง ในการคำนวณหาตำแหน่งพิกัดเพียงอย่างเดียว โดยเมื่อทราบระยะทาง จาก GPS Receiver ถึงดาวเทียม 3 ดวง เครื่องจะสามารถ คำนวณจุดตำแหน่งพิกัด ของตนเองได้ เมื่อกำหนดให้ความสูงคงที่ (ผู้ใช้ต้องป้อนค่าความสูง ที่ทราบ ให้กับเครื่อง) และถ้ารับสัญญาณ จากดาวเทียมได้ 4 ดวง เครื่องจะใช้ดาวเทียม 4 ดวงในการคำนวณ ตำแหน่งพิกัด และความสูงได้ โดยไม่จำเป็น ต้องป้อนค่าความสูงให้กับเครื่อง

2.3.3 วิธีการหาพิกัดตำแหน่งโดยใช้ระบบกำหนดตำแหน่งบนโลก

1. เครื่องรับแบบนำหน ใช้วิธีการวัด ซูโดเรนจ์ไปยัง ดาวเทียม 4 ดวง ซึ่งทำให้สามารถนำมาคำนวณตำแหน่งของเครื่องรับได้ในทันทีเรียกว่าการหาตำแหน่งแบบสมบูรณ์ แต่มีความคลาดคลื่อนทางตำแหน่งค่อนข้างมากประมาณ +- 10 ถึง 25 เมตร ซึ่งสามารถทำการรังวัดแบบสัมพันธ์ได้โดยการนำเครื่องรับแบบนำหนมากกว่า 2 เครื่องขึ้นไป นำเครื่องรับไปวางที่หมุดตั้งแต่ 2 หมุดขึ้นไป แล้วทำการวัดไปยังดาวเทียมกลุ่มเดียวกันในเวลาเดียวดัน ตำแหน่งสัมพันธ์จะมีความคลาดเคลื่อนน้อยกว่าตำแหน่งที่ได้จากการรังวัดด้วยเครื่องรับสัญญาณเดียว โดยมีความคลาดเคลื่อน ประมาณ 2 ถึง 5 เมตร

2. เครื่องรังแบบรังวัด เป็นการวัด เวลาของเคลื่อนส่งที่ต้องนำมาประมวลผลเพื่อหาตำแหน่งสัมพันธ์เท่านั้น วิธีการทำงาน คือ นำเครื่องรับแบบรังวัดไปวางที่หมุดที่ต้องการหาตำแหน่งเปรียบเทียบกันเป็นเวลาตั้งแต่ 30 นาทีขึ้นไป จากนั้นนำข้อมูลที่ได้จากการรับสัญญาณมาประมวลผลได้เป็นสัณฐานระหว่างหมุดที่วางเครื่องรับนั้นซึ่งสามารถนำข้อมูลมาประมวลผลร่วมกับข้อมูลที่ได้จากตำแหน่งที่ต้องการทราบค่าพิกัดได้ การทำงานสามารถทำได้ 5 วิธี คือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- การรังวัดแบบสถิติ (Static survey)
- การรังวัดแบบจลน์ (kinematics survey)
- การรังวัดแบบกึ่งสถิติ (pseudo static survey)
- การรังวัดแบบสถิติอย่างเร็ว (rapid static survey)
- การรังวัดแบบจลน์ในทันที (real time kinematics survey)

2.4 ความถูกต้อง (ACCURACY)

ความถูกต้องและความแม่นยำของตำแหน่งพิกัดที่คำนวณได้โดยทั่วไปแล้วเครื่องรับ GPS ที่ทำงานโดยอาศัยสัญญาณ SPS สามารถคำนวณ ค่าตำแหน่งพิกัด ที่มีความถูกต้อง อยู่ในระยะ 25 เมตร และค่าความถูกต้องของความเร็วอยู่ในระยะ 5 เมตรต่อ วินาที (เครื่อง GPS ของ Magellan สามารถคำนวณ ค่าตำแหน่งพิกัด ที่มีความถูกต้องอยู่ในระยะ 15 เมตร) เนื่องจากค่าความถูกต้องที่ได้นี้ จะขึ้นอยู่กับ นโยบาย ของรัฐบาลสหรัฐอเมริกา ที่เรียกว่า Selective Availability (SA) เพื่อรักษาความมั่นคงทางทหาร สัญญาณ SA นี้จะทำให้เกิด ค่าความผิดพลาด ขึ้นกับข้อมูล Ephemeris ที่ส่งกระจายมาจากดาวเทียม ส่งผลให้ค่าความผิดพลาด ของค่าตำแหน่งพิกัดที่ได้ มีค่าเพิ่มขึ้นเป็นระยะ 100 เมตร ในการใช้งานทั่วไปแล้ว ค่าความผิดพลาดในระยะ 100 ก็ดีเพียงพอ สำหรับการใช้งาน ที่ต้องการความถูกต้อง ที่มากกว่านี้ สามารถทำได้โดยใช้เทคนิค Differential เพื่อกำจัดผลของ SA ซึ่งทำให้ค่าที่ได้มีความถูกต้องมากขึ้น (ขณะนี้ รัฐบาลสหรัฐอเมริกา ปิด S/A แล้ว)

นอกจากนี้ ความถูกต้องของตำแหน่งพิกัด ยังขึ้นกับ ชุดของค่าคงที่ ที่เรียกว่า Map Datum ซึ่งค่าเหล่านี้ มีความแตกต่างกัน สำหรับพื้นที่ ในแต่ละพื้นที่ โดยทั่วไป แต่ละประเทศ จะใช้ Map Datum ที่แตกต่างกัน ในการสร้างแผนที่ ของพื้นที่ในประเทศ ตำแหน่งเดียวกัน บนแผนที่ 2 ฉบับ ที่ใช้ Map Datum ต่างกันในการสร้างแผนที่ จะให้ตำแหน่งพิกัด ที่แตกต่างกัน ดังนั้น การเทียบตำแหน่งพิกัด ที่ได้จาก GPS Receiver กับตำแหน่งพิกัดจริง ที่ได้จากแผนที่ จึงต้องใช้ Map Datum เดียวกัน โดยที่ GPS Receiver ส่วนมาก จะสามารถ เปลี่ยน Map Datum ของเครื่องได้หลายแบบ เพื่อให้สามารถนำเครื่องไปใช้ บอกตำแหน่ง เทียบกับแผนที่ ในพื้นที่แต่ละประเทศได้ (เครื่อง GPS ของ Magellan โดยมาก จะมี Map Datum 72 แบบ ให้เลือกใช้ ตามประเทศ โดยรวมถึง Map Datum Thai-Viet ซึ่งใช้ได้กับพื้นที่ ประเทศไทย เวียดนาม และบริเวณ อินโดจีน หลายประเทศ ดังนั้น ถ้าท่านใช้งานเครื่อง ในประเทศ และใกล้เคียง จึงสามารถตั้ง Map Datum ของเครื่อง เป็นแบบ Thai-Viet และไม่จำเป็นต้องใช้ Map Datum อื่นแต่อย่างใด สำหรับ Map Datum อื่นๆ ที่มีในเครื่อง จะครอบคลุม การใช้งานในประเทศต่างๆ ทั่วโลก เว้นใน บางบริเวณ หรือประเทศ ที่ไม่มีความสำคัญ มากนัก) สำหรับ การใช้เครื่อง ในการเดินเรือ ในทะเล และมหาสมุทร จะต้องเลือกใช้ Map Datum WGS-84 ซึ่งเป็น ชุดของค่าคงที่ สำหรับบริเวณ ทะเล มหาสมุทร และชายฝั่ง ที่ใช้ได้เกือบทุกพื้นที่ ทั่วโลก การตั้ง Map Datum ที่ไม่ถูกต้อง ให้กับเครื่อง GPS อาจทำให้ตำแหน่งพิกัดที่อ่านได้จากเครื่อง ไม่ตรงกับ ตำแหน่งพิกัด ที่ได้จากแผนที่ ความแตกต่างอาจเป็นได้ ตั้งแต่ไม่กี่เมตรจนมากถึงหลายร้อยเมตร โดยทั่วไป ถ้าไม่ทราบ ว่าแผนที่ที่ใช้อ้างอิง ทำโดยใช้ Map Datum ใด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ให้เลือกตั้ง Map Datum ของเครื่องเป็น WGS-84 แต่ถ้าทราบ Map Datum ของแผนที่ที่ใช้เปรียบเทียบ ก็ให้ตั้ง Map Datum ของเครื่อง GPS เป็นแบบเดียวกัน

สำหรับประเทศไทย ถ้าตั้ง WGS-84 ให้กับเครื่อง GPS จะทำให้ตำแหน่งพิกัด ที่อ่านได้จากเครื่องเทียบกับแผนที่ ประเทศไทย ที่อ้างอิงกับ Map Datum แบบ Thai-Viet มีความแตกต่าง ในแนวราบ ที่ประมาณ 413 เมตร ซึ่งค่อนข้างสูงมาก ดังนั้น ก่อนการใช้เครื่อง GPS ควรตั้งค่า Map Datum ให้ตรงกับแผนที่ ที่จะใช้เปรียบเทียบ ทุกครั้ง

2.4.1 ความคลาดเคลื่อนในงานรังวัดดาวเทียม ระบบกำหนดตำแหน่งบนโลก

ในงานรังวัดดาวเทียม ความคลาดเคลื่อนที่เกิดขึ้นอาจพิจารณาแยกได้เป็น 3 กลุ่ม คือ

- กลุ่มเกี่ยวข้องกับดาวเทียม ได้แก่ ความคลาดเคลื่อนวงโคจร และความคลาดเคลื่อนนาฬิกาดาวเทียม
- กลุ่มเกี่ยวข้องกับการแพร่กระจายของสัญญาณดาวเทียม ได้แก่ ความคลาดเคลื่อนของการหักเหในชั้นบรรยากาศ และการเกิดคลื่นสะท้อน
- กลุ่มสุดท้ายเกี่ยวข้องกับเครื่องรับสัญญาณ คือ นาฬิกาเครื่องรับ

1) ความคลาดเคลื่อนวงโคจรดาวเทียม วงโคจรดาวเทียมที่มาจากข้อมูลดาวเทียมในสัญญาณที่รับได้นั้นเป็นวงโคจรที่ได้จากการคำนวณล่วงหน้าโดยอาศัยรูปจำลองของแรงต่าง ๆ ที่กระทำต่อดาวเทียม รูปจำลองที่ใช้อาจจะไม่ถูกต้อง หรือไม่ละเอียดเพียงพอเมื่อเทียบกับแรงจริง ๆ ที่กระทำต่อดาวเทียมในขณะที่ทำการวัด ดังนั้นตำแหน่งดาวเทียมจาก อิฟีเมอริส ดาวเทียมที่ส่งกระจายลงมาพร้อมสัญญาณดาวเทียมจึงไม่ถูกต้อง ซึ่งมีผลต่อการหาตำแหน่งบนโลก การหาตำแหน่งแบบสัมพัทธ์จะทำให้ความคลาดเคลื่อนวงโคจรลดลงได้

2) ความคลาดเคลื่อนนาฬิกาดาวเทียม และนาฬิกาเครื่องรับเวลาดาวเทียม หมายถึง เวลาที่อ่านได้จากนาฬิกาของดาวเทียมแต่ละดวง มีนาฬิกาที่มีมาตรฐานสูง 4 เครื่อง สถานีควบคุมหลักสามารถควบคุมเวลาดาวเทียมให้เวลา GPS ต่างกันไม่เกินกว่า 1 ใน 1,000 วินาที (1 ms) และควบคุมความถี่ให้มีความถูกต้องถึง 10^{9} เวลาที่ถูกต้องของเครื่องรับ สามารถหาได้จากสัญญาณดาวเทียมที่มีเวลาจากนาฬิกาดาวเทียมอยู่ เมื่อเครื่องรับถอดรหัสได้ก็จะรู้เวลา GPS เครื่องรับจึงไม่จำเป็นต้องมีนาฬิกาที่มีมาตรฐานสูง

3) ความคลาดเคลื่อนเนื่องจากการหักเหในชั้นบรรยากาศชั้นไอโอโนสเฟียร์ และชั้นโทรโพสเฟียร์มีผลต่อการหักเหคลื่นแม่เหล็กไฟฟ้า ทำให้เส้นทางการเคลื่อนที่ของคลื่นเบี่ยงเบนไป

4) คลื่นสะท้อน (multipart) หมายถึงการแพร่กระจายของคลื่นที่มีการสะท้อนตั้งแต่หนึ่งครั้งขึ้นไป พื้นผิวที่สะท้อนอาจจะอยู่ในแนวดิ่ง ราบ หรือเอียงก็ได้ เช่น ผนังตึก ถนน ผิวน้ำ หรือยานพาหนะ คลื่นสะท้อนมีผลกับทั้งรหัส และคลื่นผลของคลื่นสะท้อนเป็นสาเหตุหนึ่งที่ทำให้เกิดคลื่นหลุด (cycle slip) เพราะช่องรับสัญญาณไม่สามารถจับสัญญาณดาวเทียมได้ การรังวัดเป็นเวลานานจะทำให้ผลของคลื่นสะท้อนลดลง แต่เทคนิคของการรังวัดบางวิธีไม่สามารถวางเสาอากาศทิ้งไว้เป็นเวลานาน ๆ ได้ ดังนั้นจึงต้องหลีกเลี่ยงการรับสัญญาณที่มีคลื่นสะท้อนโดยเลือกจุดวางเครื่องรับที่ไม่มีพื้นผิว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สะท้อนอยู่ใกล้เคียง เลือกเสาอากาศที่ออกแบบเฉพาะ เช่น เสาอากาศที่มีแผ่นกราวนด์ หรือใช้วัสดุที่ดูดซับคลื่นวางรอบเสาอากาศ

5) ความคลาดเคลื่อนของเครื่องรับ มีสาเหตุมาจากหลายส่วน เช่น Noise ในการวัดของเครื่องรับ bias ระหว่างช่องรับสัญญาณ การประวิงของเฟส ความไม่เสถียรของออสซิลเลเตอร์ และจุดศูนย์กลางเฟสของเสาอากาศ เป็นต้น

2.5 แนวโน้มการพัฒนาระบบกำหนดตำแหน่งบนโลกด้วยดาวเทียม

การยกเลิกการใช้ SA (Selective Availability) เมื่อวันที่ 2 พฤษภาคม พ.ศ. 2543 ทำให้ความถูกต้องของการหาตำแหน่งบนโลกมีความถูกต้องมากขึ้นอยู่ในระดับไม่เกิน 10 เมตร และเมื่อต้นปี 2542 สหรัฐอเมริกาได้ประกาศว่าจะกล้ำสัญญาณ รหัส C/A ในคลื่นส่ง L2 ซึ่งปัจจุบันวัดระนะจาก C/A ในคลื่นส่ง L1 ได้เพียงความถี่เดียว โดยมีแผนจะดำเนินการได้ใน พ.ศ. 2546 ซึ่งจะส่งผลให้คำนวณความคลาดเคลื่อนของการวัดระยะทางที่เกิดขึ้นในชั้นไอโอสเฟียร์ได้และจะทำให้การคำนวณตำแหน่งของเครื่องรับแบบมือถือมีความถูกต้องยิ่งขึ้น

ในพ.ศ. 2548 สหรัฐอเมริกามีแผนจะส่งดาวเทียมระบบกำหนดตำแหน่งบนโลก รุ่น 2F ขึ้นสู่วงโคจรโดยส่งสัญญาณในคลื่นความถี่ที่สามหรือ L3 ซึ่งจะทำให้การหาตำแหน่งมีความถูกต้องของตำแหน่งที่ดีขึ้น และระยะเวลาที่ใช้เก็บข้อมูลเพื่อการคำนวณตำแหน่งจะลดน้อยลง

ในอนาคตคาดว่าการคำนวณวงโคจรซึ่งเป็นข้อมูลสำคัญในการคำนวณ หาตำแหน่งของเครื่องรับนั้นจะมีเทคนิคที่รวดเร็วขึ้นและมีความถูกต้องมากยิ่งขึ้น ซึ่งส่งผลถึงการหาตำแหน่งของเครื่องรับมีความถูกต้องมากขึ้น

การขยายตัวด้านการสำรวจวัดด้วยดาวเทียมทำให้เกิดเทคโนโลยี GNSS (The Global Navigation Satellite System) ซึ่งเป็นการปรับปรุงระบบกำหนดตำแหน่งบนโลกเดิมให้มีประสิทธิภาพยิ่งขึ้น รวมทั้งนำระบบอื่นมาช่วยเพิ่มประสิทธิภาพของระบบกำหนดตำแหน่งบนโลก ซึ่งการผสมผสานระบบกำหนดตำแหน่งบนโลกกับระบบอื่น ได้แก่

1) นำระบบกำหนดตำแหน่งบนโลกทำงานร่วมกับดาวเทียมระบบอื่น โดยประเทศรัสเซียได้สร้างดาวเทียมของตนเองเพื่อแข่งขันกับระบบดาวเทียม GPS โดยมีชื่อว่า ระบบดาวเทียม GLONASS ซึ่งปัจจุบันใช้งานได้ 7 ดวง กลุ่มประเทศ สหภาพยุโรป (The European Union :EU) ได้ร่วมกับองค์การอวกาศของยุโรป (The European Space Agency : ESA) มีแนวคิดที่จะสร้างระบบดาวเทียมของตนเองโดยมีชื่อว่า GALILEO คาดว่าจะมีดาวเทียมจำนวน 30 ดวง สามารถใช้งานได้ใน พ.ศ. 2551

2) การนำร่องระบบกำหนดตำแหน่งบนโลก ใช้ร่วมกับ Pseudo life (PL) ซึ่งเป็นเครื่องส่งสัญญาณภาคพื้นดิน ทำให้ผู้ใช้มีจำนวนดาวเทียมเพิ่มมากขึ้น และสามารถประยุกต์ใช้ได้ในบริเวณที่มีสิ่งกีดขวางสัญญาณดาวเทียม

3) การนำระบบกำหนดตำแหน่งบนโลกใช้ร่วมกับ INS (Initial Navigator System) ซึ่งเป็นระบบการคำนวณทิศทางและระยะทาง สามารถให้ข้อมูลเชิงตำแหน่ง ความเร็วและการวางตัว ซึ่งสามารถใช้ทดแทนการนำทางในกรณีที่เกิดการกีดขวางของสัญญาณดาวเทียมทำให้ขาดการรับสัญญาณที่ต่อเนื่อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4) การประยุกต์ใช้ระบบกำหนดตำแหน่งบนโลกกับระบบอื่นๆ เช่น การผสมผสานระหว่าง GPS+GLONASS+INS; GPS+PL+INS, GPS+GLONASS+INS+PL เพื่อตอบสนองความต้องการของผู้ใช้มากยิ่งขึ้น

ระบบ GNSS จะนำมาประยุกต์ใช้ในการคำนวณหาความถูกต้องทางตำแหน่งสูง เช่น งานรังวัดเพื่อควบคุมและทำแผนที่ การตรวจวัดการเคลื่อนตัวของเปลือกโลกนอกจากนี้งานที่ไม่ต้องการค่าความถูกต้องทางตำแหน่งสูงมาก ก็สามารถประยุกต์ใช้ GNSS ได้เช่นเดียวกัน เช่น งานติดตามรถยนต์ เครื่องบิน การหาตำแหน่งของผู้ใช้มือถือ (mobile positioning)

ในด้านการพัฒนาเครื่องรับสัญญาณดาวเทียมจะต่ำลง ขนาดของเครื่องรับจะเล็กลง และมีขีดความสามารถที่สูงมากขึ้น และมีแนวโน้มที่จะพัฒนาการผลิตเครื่องรับสัญญาณที่สามาถนำไปติดตั้งไว้ในอุปกรณ์อื่น ๆ เช่น นาฬิกาข้อมือ Personal Digital Assistant (PDA) โทรศัพท์มือถือ รวมทั้งการพัฒนาเทคโนโลยี เครื่องรับสัญญาณดาวเทียมระบบกำหนดตำแหน่งบนโลกภายในอาคาร (indoor GPS) ให้ใช้งานได้ภายในอาคารหรือบริเวณที่มีการปิดกั้นของสัญญาณดาวเทียมได้

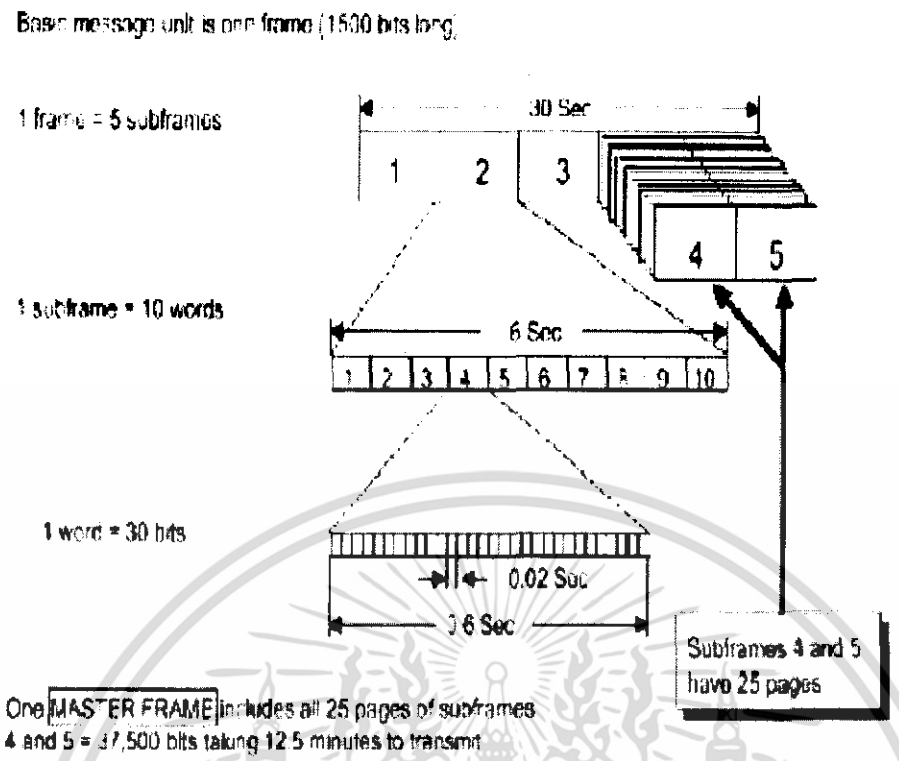
2.6 โครงสร้างของข้อมูลที่ส่งจากดาวเทียม

ข้อมูลที่ส่งจากดาวเทียม GPS ประกอบด้วยข้อมูลซึ่งถูกแบ่งออกเป็นเฟรมย่อย ๆ เรียกว่า ซับเฟรม (Sub frame) แต่ละซับเฟรมจะแทรกค่าเวลาที่ซับเฟรมนั้น ๆ ถูกส่งมาจากดาวเทียม GPS ไว้เพื่อใช้ร่วมในการคำนวณหาค่าพิกัดตำแหน่ง ข้อมูลแต่ละเฟรมมีขนาด 1,500 บิต ถูกแบ่งในรูปซับเฟรมขนาด 300 บิต จำนวน 5 ซับเฟรม ข้อมูลหนึ่งเฟรมจะถูกส่งมาจากดาวเทียม ทุก ๆ 30 วินาที ซับเฟรมขนาด 6 วินาที (300 บิต) จะบรรจุไว้ด้วยข้อมูลเส้นทางโคจรและของข้อมูลนาฬิกาโดยข้อมูลในแต่ละเฟรมประกอบด้วยส่วนปลีกย่อยดังนี้

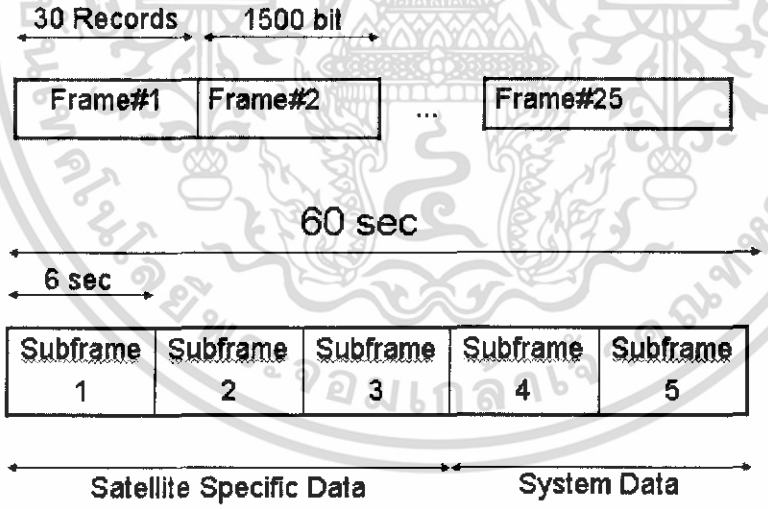
- ซับเฟรมที่ 1 เป็นข้อมูลในการแก้ไขเวลาของดาวเทียม GPS
- ซับเฟรมที่ 2 และ 3 เป็นข้อมูลเส้นทางโคจรของดาวเทียม GPS
- ซับเฟรมที่ 4 และ 5 เป็นข้อมูลอื่น ที่เกี่ยวข้องกับระบบ

ข้อมูลจากดาวเทียมซึ่งบรรจุไว้ด้วยข้อมูลในการนำร่อง (Navigation Message) ที่ครบสมบูรณ์จะประกอบด้วยเฟรมข้อมูลจำนวน 25 เฟรม (125 ซับเฟรม) โดยข้อมูลดังกล่าวจะถูกส่งจากดาวเทียมทุก ๆ 12.5 นาที เป็นอย่างน้อย โดยทั่วไปเครื่องรับสัญญาณจะได้รับข้อมูลของตำแหน่งล่าสุดของดาวเทียมทุกชั่วโมง เพื่อใช้ร่วมกับอัลกอริทึมในการคำนวณพิกัดตำแหน่ง และข้อมูลการโคจรของดาวเทียมแต่ละดวง เพื่อให้เครื่องรับสัญญาณทราบตำแหน่งของดาวเทียมแต่ละดวงรวมทั้งปรับชดเชยความผิดพลาดของสัญญาณพาหะจาก ปรากฏการณ์ดอปเปอร์ของความถี่พาหะ (Carrier Doppler frequency) ซึ่งเกิดจากการที่ความถี่พาหะมีการเบนค่าไปเนื่องจากการเคลื่อนที่ของดาวเทียม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.7 โครงสร้างของข้อมูลที่ส่งมาจากดาวเทียม GPS



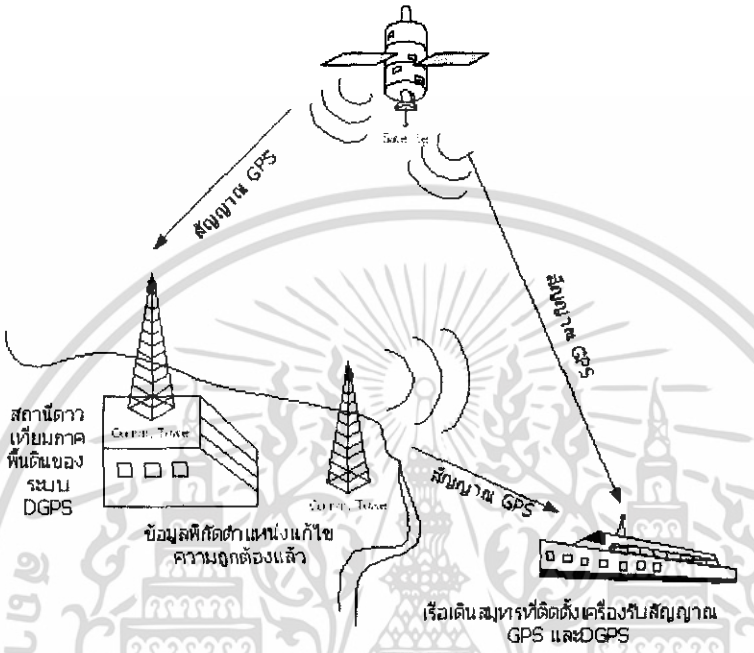
รูปที่ 2.8 รายละเอียดการแบ่งเฟรม

นอกจากนั้นชุดข้อมูลดาวเทียม GPS โดยสมบูรณ์ยังจะประกอบด้วยข้อมูลแบบจำลองของชั้นบรรยากาศไอโอโนสเฟียร์อยู่ด้วย เพื่อให้เครื่องรับสัญญาณสามารถประเมินค่าในการหน่วงเฟสของสัญญาณ (Phase delay) ที่เกิดจากสภาพชั้นบรรยากาศไอโอโนสเฟียร์ แต่ส่วนที่เครื่องรับสัญญาณจำเป็นต้องใช้ก็คือข้อมูลใน 3 ซับเฟรมแรก หากสามารถที่ได้รับข้อมูลดังกล่าวจากดาวเทียมตั้งแต่สามดวงขึ้นไป จึงจะสามารถคำนวณหาพิกัดตำแหน่งปัจจุบันของเครื่องรับสัญญาณได้ เวลาที่เครื่องรับ GPS สามารถรับและแสดงผลได้นั้นขึ้นอยู่กับตัวเครื่องรับ

เอกสารนี้ใช้เพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.7 หลักการของระบบ DGPS (Differential GPS)

หลักการ DGPS คือ การแก้ไขความคลาดเคลื่อนที่เกิดขึ้นโดยอาศัยการติดตั้งเครื่องรับสัญญาณ GPS บนสถานีภาคพื้นดินในจุดที่ทราบพิกัดที่แน่นอนเพื่อเป็นตำแหน่งอ้างอิง โดยที่สถานีภาคพื้นดินในจุดที่ทราบพิกัดที่แน่นอนเพื่อเป็นตำแหน่งอ้างอิง โดยที่สถานีภาคพื้นดินจะทำหน้าที่แก้ไขแล้วส่งไปยังเครื่องรับสัญญาณที่อยู่ในรัศมีทำการของระบบ



รูปที่ 2.9 แผนผังแสดงองค์ประกอบของระบบ DGPS

ระบบ DGPS ใช้คลื่นวิทยุในย่านต่าง ๆ เช่น MF, HF, UHF/VHF เป็นต้น เพื่อสื่อสารกับเครื่องลูกข่ายในระบบ โดยการเลือกใช้คลื่นวิทยุจะเป็นย่านใดนั้น ขึ้นอยู่กับประเภทของงานที่จะนำระบบไปประยุกต์ด้วย อย่างไรก็ตามระบบ DGPS ก็ยังมีอุปสรรคหลายอย่างในการใช้งาน เช่น รัศมีทำการของระบบที่มีขอบเขตจำกัด ข้อมูลที่ส่งออกไปสถานีภาคพื้นดินสามารถถูกแทรกแซงจากความแปรปรวนของสภาพอากาศรวมทั้งจะต้องติดตั้งเสาอากาศ และเครื่องรับสัญญาณเพิ่มเพื่อใช้งานระบบ แม้ว่าการติดตั้งและใช้งานระบบ DGPS จะมีอุปสรรคและความยุ่งยากอยู่บ้าง แต่ระบบก็สามารถลดความคลาดเคลื่อนที่เกิดขึ้นได้จากหลายร้อยฟุต

2.8 มาตรฐาน NMEA 0183

NMEA Message คือข้อมูลซึ่งส่งออกมาจากโมดูลรับสัญญาณ GPS ข้อมูลใน NMEA Message สามารถแบ่งได้เป็นประโยค (Sentence) โดยในแต่ละประโยคจะประกอบด้วย 2 ส่วนคือ อักขระแอสกี (ASCII) ซึ่งเป็นข้อมูล และเครื่องหมายสำหรับขั้นข้อมูล ซึ่งมีความยาวรวมไม่เกิน 80 ตัวอักษร สามารถอ่านข้อมูล NMEA Message ได้โดยการใช้ซอฟต์แวร์สื่อสาร เช่น Hyper Terminal โดยแต่ละประโยคจะมีรายละเอียดที่ต่างกันแล้วแต่จะเลือกใช้ ประโยคที่ใช้กันเป็นหลักใน NMEA Message จะมีอยู่หลายประโยครายละเอียดภายในประโยคต่างๆ ของ NMEA Message มีดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

GGA (Global Positioning System Fixed Data)

$GPGGA, 161229.487, 3723.245, N, 12158.3416, W, 1, 07, 1.0, 9.0, M,,,, 0000*18<CR><LF>

หมายเลขฟิลด์	ชื่อ	ตัวอย่าง	คำบรรยาย
	Message ID	$GPGGA	ส่วนหัวโปรโตคอลของ GGA
1	UTC Position	161229.487	hhmmss.sss เวลามาตรฐานกลาง
2	Latitude	3723.2475	Ddmm.mmmm ตำแหน่งละติจูด
3	N/S Indicator	N	N = north or S = south
4	Longitude	12158.3416	Dddmm.mmmm ตำแหน่งลองจิจูด
5	E/W Indicator	W	E = east or W = west
6	Position Fix Indicator	1	เป็นตัวบอกลักษณะของ GPS (0 = not fix , 1 = GPS fix, 2 = DGPS fix)
7	Satellites Used	07	จำนวนดาวเทียมที่ใช้คำนวณพิกัดอยู่ในช่วง 0 ถึง 12
8	HDOP	1.0	Horizontal Dilution of Precision
9	MSL Altitude	9.0	ความสูงของสายอากาศเหนือระดับน้ำทะเล (เมตร)
10	Units	M	หน่วยของความสูงของเสาอากาศ (เมตร)
11	Geoid Separation		ความต่างระหว่างระบบ WGS-84 กับระดับน้ำทะเล (เมตร)
12	Units	M	หน่วยของความต่างของจีออยด์ (ระดับน้ำทะเล) (เมตร)
13	Age of Diff. Corr.		จะไม่มีฟิลด์นี้เมื่อไม่ใช้ DGPS (วินาที)
14	Diff. Ref. Station ID	0000	หมายเลขประจำสถานีอ้างอิง (DGPS)
	Checksum	*18	เช็คซัม
	<CR><LF>		สิ้นสุดประโยค

ตารางที่ 2.1 ความหมายของข้อมูลที่อยู่ในประโยค GGA

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

GLL (Geographic Position Latitude/Longitude)

$GPGLL, 3723.245, N, 12158.3416, W, 161229.487, A*2C<CR><LF>

หมายเลขฟิลด์	ชื่อ	ตัวอย่าง	คำบรรยาย
	Message ID	$GPGLL	ส่วนหัวโปรโตคอลของ GLL
1	Latitude	3723.2475	ddmm.mmmm ตำแหน่งละติจูด
2	N/S Indicator	N	N = north or S = south
3	Longitude	12158.3416	dddmm.mmmm ตำแหน่งลองจิจูด
4	E/W Indicator	W	E=east or W= west
5	UTC Position	161229.487	hhmmss.ss เวลามาตรฐานกลาง
6	Status	A	A = data valid or V = data not valid
	Checksum	*2C	เช็คซัม
	<CR><LF>		สิ้นสุดประโยค

ตารางที่ 2.2 ความหมายของข้อมูลที่อยู่ในประโยค GLL

72971

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

GSA (GNSS DOP and Active Satellites)

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

หมายเลขฟิลด์	ชื่อ	ตัวอย่าง	คำบรรยาย
	Message ID	$GPGSA	ส่วนหัวโปรโตคอลของ GSA
1	Mode 1	A	M = Manual, A = Automatic
2	Mode 2	3	1 = ไม่ระบุค่า, 2 = 2 มิติ, 3 = 3 มิติ
3-14	Satellite Used	07, 02, 26,27, 09, 04, 15,,,,,,	RPNs ของดาวเทียมใช้ในการแก้ปัญหา (เป็น null สำหรับฟิลด์ที่ไม่ได้ใช้)
15	PDOP	1.8	Position Dilution of Precision
16	HDOP	1.0	Horizontal Dilution of Precision
17	VDOP	1.5	Vertical Dilution of Precision
	Checksum	*33	เช็คซัม
	<CR><LF>		สิ้นสุดประโยค

ตารางที่ 2.3 ความหมายของข้อมูลที่อยู่ในประโยค GSA

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

GSV (GNSS Satellites in View)

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

หมายเลขฟิลด์	ชื่อ	ตัวอย่าง	คำบรรยาย
	Message ID	$GPGSV	ส่วนหัวโปรโตคอลของ GSV
1	Number of Messages	2	จำนวนรวมทั้งหมดของ Messages (1-3)
2	Messages Number	1	หมายเลข Message (1-3)
3	Satellites in View	07	จำนวนรวมทั้งหมดของดาวเทียมในการมองเห็น
4	Satellites ID	07	Ch.1 (อยู่ในช่วง 1-32)
5	Elevation	79	Ch.1 (จำนวนสูงสุดคือ 90) หน่ายคือ degrees
6	Azimuth	048	Ch.1 (True, อยู่ในช่วง 0-359) หน่วยเป็น degrees
7	SNR (C/No)	42	Range 0 to 99, null when not tracking
8-11	...	02, 51, 062, 43	Ch.2 ของ Satellites ID, Elevation, Azimuth, SNR
12-15	...	26, 36, 256, 42	Ch.3ของ Satellites ID, Elevation, Azimuth, SNR
16-19	...	27, 27, 138, 42	Ch.4 ของ Satellites ID, Elevation, Azimuth, SNR
	Checksum	*71	เช็คซัม
	<CR><LF>		สิ้นสุดประโยค

ตารางที่ 2.4 ความหมายของข้อมูลที่อยู่ในประโยค GSV

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

RMC (Recommended Minimum Specific GNSS Data)

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

หมายเลขฟิลด์	ชื่อ	ตัวอย่าง	คำบรรยาย
	Message ID	$GPRMC	ส่วนหัวโปรโตคอลของ RMC
1	UTC Position	161229.487	hhmmss.sss เวลามาตรฐานกลาง
2	Status	A	A = data valid or V = data not valid
3	Latitude	3723.2475	ddmm.mmmm ตำแหน่งละติจูด
4	N/S Indicator	N	N = north or S = south
5	Longitude	12158.3416	dddmm.mmmm ตำแหน่งลองจิจูด
6	E/W Indicator	W	E = east or W = west
7	Speed Over Ground	0.13	ความเร็ว (Knots)
8	Course Over Ground	309.62	True (degrees)
9	Date	120598	Ddmmyy
10	Magnetic Variation		Degrees
11	Magnetic variation (Ref)		E = east or W = west (degrees)
	Checksum	*10	เช็คซัม
	<CR><LF>		สิ้นสุดประโยค

ตารางที่ 2.5 ความหมายของข้อมูลที่อยู่ในประโยค RMC

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

VTG (Course Over Ground and Ground Speed)

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

หมายเลขฟิลด์	ชื่อ	ตัวอย่าง	คำบรรยาย
	Message ID	$GPVTG	ส่วนหัวโปรโตคอลของ VTC
1	Course	309.62	Measured heading (degrees)
2	Reference	T	True
3	Course		Measured heading (degrees)
4	Reference	M	Magnetic
5	Speed	0.13	ได้จากการวัดความเร็วด้านแนวนอน (knots)
6	Units	N	Knots
7	Speed	0.2	ได้จากการวัดความเร็วด้านแนวนอน (Km/hr)
8	Units	K	กิโลเมตรต่อชั่วโมง
หมายเลขฟิลด์	ชื่อ	ตัวอย่าง	คำบรรยาย
	Checksum	*6E	เช็คซัม
	<CR><LF>		สิ้นสุดประโยค

ตารางที่ 2.6 ความหมายของข้อมูลที่อยู่ในประโยค VTC

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ข้อมูลรายระเอียดดังกล่าวที่ได้กล่าวมาแล้วนี้สรุปคร่าว ๆ ได้ตามการใช้งานดังนี้

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

ตารางที่ 2.7 สรุปคุณสมบัติของ 6 ประโยคหลักใน NMEA message

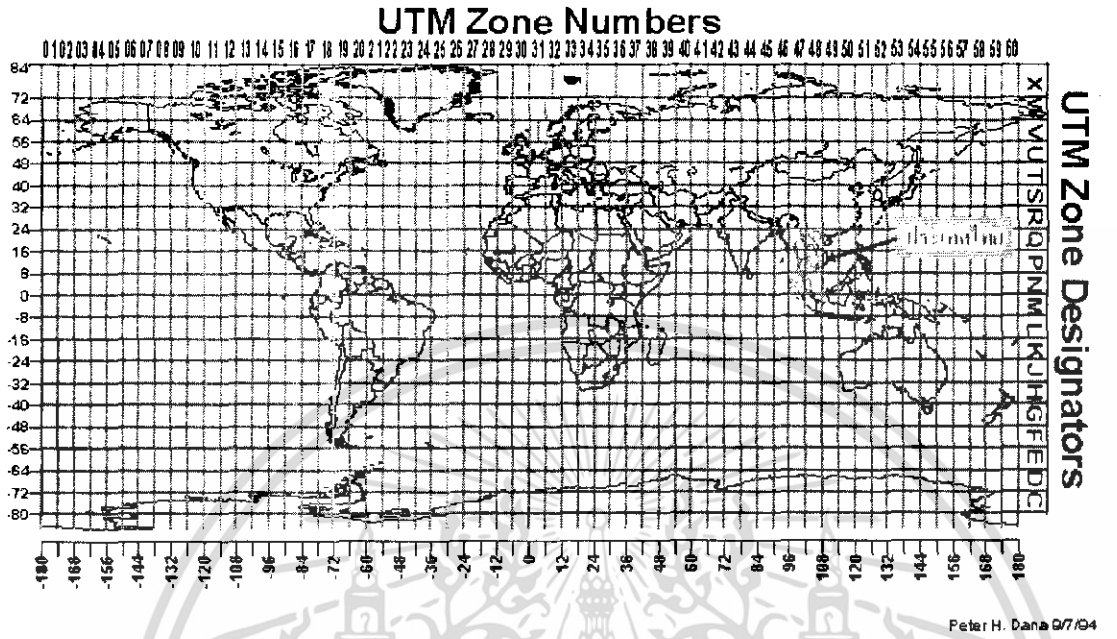
2.9 การอ้างพิกัดแบบ UTM

การอ้างพิกัดตำแหน่งแบบ UTM (Universal Transverse Mercator) เป็นระบบกริดตาราง (Grid) ที่พัฒนาขึ้นโดยกระทรวงกลาโหมของสหรัฐอเมริกาตั้งแต่ปี ค.ศ.1974 สำหรับใช้ชี้ตำแหน่งที่ตั้งบนโลกอย่างรวดเร็วและแม่นยำในการทหาร วิธีนี้สามารถทำได้ง่ายและได้ผลลัพธ์ที่ดี โดยเฉพาะอย่างยิ่งสำหรับการส่งข้อมูลจากแผนที่หนึ่งไปยังอีกแผนที่หนึ่ง ตัวอย่างที่ดีตัวอย่างหนึ่งก็คือ การส่งข้อมูลของพิกัดตำแหน่งจากแผนที่ส่งหนึ่งไปให้ผู้รับ โดยผู้รับสามารถเอาข้อมูลของพิกัดตำแหน่งดังกล่าวไปใช้เพื่อจะได้ทราบที่ตั้งจริงบนแผนที่ได้

UTM เป็นระบบตารางกริดที่ใช้ช่วยในการกำหนดตำแหน่งและใช้อ้างอิง ในการบอกตำแหน่งที่นิยมใช้กับแผนที่ในกิจการทหารของประเทศต่างๆ เกือบทั่วโลกในปัจจุบัน เพราะเป็นระบบตารางกริดที่มีขนาดรูปร่างเท่ากันทุกตาราง และมีวิธีการกำหนดบอกค่าพิกัดที่ง่ายและถูกต้องเป็นระบบกริดที่นำเอาเส้นโครงแผนที่แบบ Universal Transverse Mercator Projection ของ Gauss Krugger มาใช้ดัดแปลงให้สามารถถ่ายทอดรายละเอียดของพื้นผิวโลก ให้อยู่ในรูปทรงกระบอก (แกนของรูปทรงกระบอกจะทับกับแนวเส้นอิเควเตอร์ และตั้งฉากกับแนวแกนของขั้วโลก) ประเทศไทยเราได้นำเอาเส้นโครงแผนที่แบบ UTM นี้มาใช้กับการทำแผนที่กิจการทหารภายในประเทศจากรูปถ่ายทางอากาศในปี 1953 ร่วมกับสหรัฐอเมริกาเป็นแผนที่มาตราส่วน 1:50,000 ชุด 708 และปรับปรุงใหม่เป็นชุด L 7017 ที่ใช้ในปัจจุบัน UTM เป็นระบบตาราง กริดที่ใช้ช่วยในการกำหนดตำแหน่งและใช้อ้างอิง ในการบอกตำแหน่งที่นิยม ใช้กับแผนที่ในกิจการทหารของประเทศต่างๆ เกือบทั่วโลกในปัจจุบัน เพราะเป็นระบบตารางกริดที่มีขนาดรูปร่างเท่ากันทุกตาราง และมีวิธีการกำหนดบอกค่าพิกัดที่ง่ายและถูกต้องเป็นระบบกริดที่นำเอาเส้นโครงแผนที่แบบ Universal Transverse Mercator Projection ของ Gauss Krugger

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

มาใช้ดัดแปลงการถ่ายทอดรายละเอียดของพื้นผิวโลกให้รูปทรงกระบอก Mercator Projection อยู่ในตำแหน่ง Mercator Projection (แกนของรูปทรงกระบอกจะทับกับแนวเส้นอิเควเตอร์ และตั้งฉากกับแนวแกนของขั้วโลก)



รูปที่ 2.10 การแบ่งพื้นที่ออกเป็นโซน (Zone) สำหรับการอ้างพิกัดแบบ UTM

แผนที่ระบบพิกัดกริดที่ใช้เส้นโครงแผนที่แบบ UTM เป็นระบบเส้นโครงชนิดหนึ่งที่ใช้ผิวรูปทรงกระบอกเป็นผิวแสดงเส้นเมริเดียน (หรือเส้นลองติจูด) และเส้นละติจูดของโลก โดยใช้ทรงกระบอกตัดโลกระหว่างละติจูด 84 องศาเหนือ และ 80 องศาใต้ในลักษณะแกนรูปทรงกระบอกทำมุมกับแกนโลก 90 องศารอบโลก แบ่งออกเป็น 60 โซน ๆ ละ 6 องศา โซนที่ 1 อยู่ระหว่าง 180 องศากับ 174 องศาตะวันตก และมีลองติจูด 177 องศาตะวันตก เป็นเมริเดียนย่านกลาง (Central Meridian) มีเลขกำกับแต่ละโซนจาก 1 ถึง 60 โดยนับจากซ้ายไปทางขวาระหว่างละติจูด 84 องศาเหนือ 80 องศาใต้แบ่งออกเป็น 2 ช่องๆ ละ 8 องศายกเว้นช่องสุดท้ายเป็น 12 องศาโดยเริ่มนับตั้งแต่ละติจูด 80 องศาใต้ขึ้นไปทางเหนือให้ช่องแรกเป็นอักษร C และช่องสุดท้ายเป็นอักษร X (ยกเว้น I และ O) จากการแบ่งตามที่กล่าวแล้วจะเห็นพื้นที่ในเขตลองติจูด 180 องศาตะวันตกถึง 180 องศาตะวันออก และละติจูด 80 องศาใต้ถึง 84 องศาเหนือ จะถูกแบ่งออกเป็นรูปสี่เหลี่ยมผืนผ้า 1,200 รูปแต่ละรูปมีขนาดกว้างยาว $6° \times 8°$ จำนวน 1,140 รูป และกว้างยาว $6° \times 12°$ จำนวน 60 รูป รูปสี่เหลี่ยมนี้เรียกว่า Grid Zone Designation (GZD) การเรียกชื่อ Grid Zone Designation ประเทศไทยมีพื้นที่อยู่ ระหว่างละติจูด 5 องศา 30 ลิปดาเหนือ ถึง 20 องศา 30 ลิปดาเหนือ และลองติจูดประมาณ 97 องศา 30 ลิปดาตะวันออก ถึง 105 องศา 30 ลิปดาตะวันออก ดังนั้นประเทศไทยจึงตกอยู่ใน GZD 47N 47P 47Q 48N 48P และ 48Q การอ่านค่าพิกัดกริดเพื่อให้พิกัดค่ากริดในโซนหนึ่งๆ มีค่าเป็นบวกเสมอ จึงกำหนดให้มีเส้นศูนย์สูตรสมมุติขึ้น 2 แห่ง ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ในบริเวณที่อยู่เหนือเส้นศูนย์สูตร: เส้นศูนย์สูตรมีระยะห่างจากศูนย์สมมุติเท่ากับ 0 เมตร และเส้นเมริเดียนย่านกลางห่างจากศูนย์สมมุติ 500,000 เมตร ทางตะวันออก

- ในบริเวณที่อยู่ใต้เส้นศูนย์สูตร: เส้นศูนย์สูตรมีระยะห่างจากศูนย์สมมุติไปทางเหนือ 10,000,000 เมตร และเมริเดียนย่านกลางห่างจากศูนย์สมมุติ 500,000 เมตรทางตะวันออก

2.10 TCP/IP Protocol

TCP/IP (Transmission Control Protocol/Internet Protocol) เป็นชุดของโปรโตคอลที่ถูกใช้ในการสื่อสารผ่านเครือข่ายอินเทอร์เน็ต โดยมีวัตถุประสงค์เพื่อให้สามารถใช้สื่อสารจากต้นทางข้ามเครือข่ายไปยังปลายทางได้ และสามารถหาเส้นทางที่จะส่งข้อมูลไปได้เองโดยอัตโนมัติ ถึงแม้ว่าในระหว่างทางอาจจะผ่านเครือข่ายที่มีปัญหา โปรโตคอลก็ยังคงหาเส้นทางอื่นในการส่งผ่านข้อมูลไปให้ถึงปลายทางได้

ชุดโปรโตคอลนี้ได้รับการพัฒนามาตั้งแต่ปี 1960 ซึ่งถูกใช้เป็นครั้งแรกในเครือข่าย ARPANET ซึ่งต่อมาได้ขยายการเชื่อมต่อไปทั่วโลกเป็นเครือข่ายอินเตอร์เน็ต ทำให้ TCP/IP เป็นที่ยอมรับอย่างกว้างขวางจนถึงปัจจุบัน

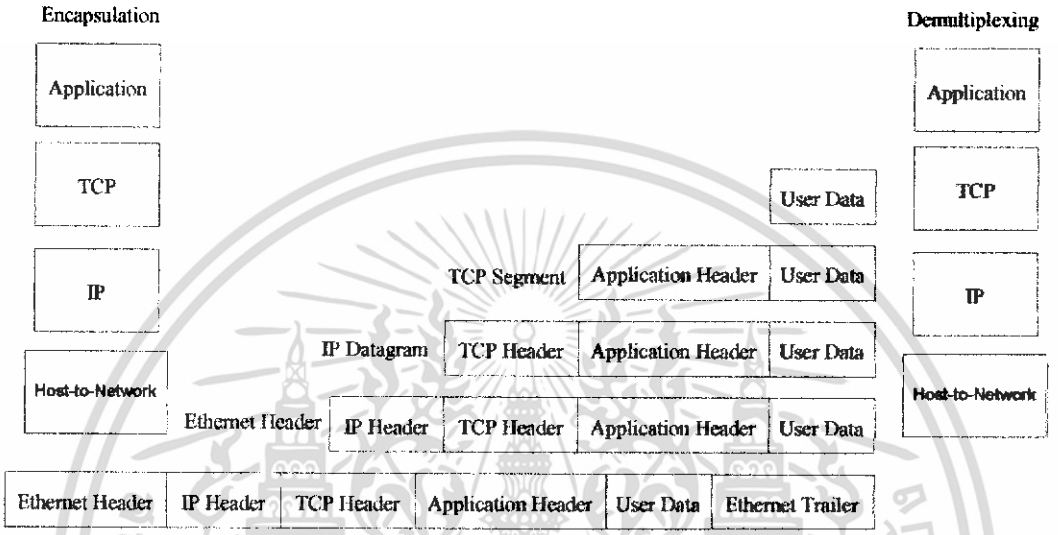
TCP/IP มีจุดประสงค์ของการสื่อสารตามมาตรฐาน คือ

1. เพื่อใช้ติดต่อสื่อสารระหว่างระบบที่มีความแตกต่างกัน
2. ความสามารถในการแก้ไขปัญหาที่เกิดขึ้นในระบบเครือข่าย เช่นในกรณีที่ผู้ส่งและผู้รับยังคงมีการติดต่อกันอยู่ แต่โหนดกลางที่ใช้เป็นผู้ช่วยรับ-ส่งเกิดเสียหายใช้การไม่ได้ หรือสายสื่อสารบางช่วงถูกตัดขาด กฎการสื่อสารนี้จะต้องสามารถจัดหาทางเลือกอื่นเพื่อทำให้การสื่อสารดำเนินต่อไปได้โดยอัตโนมัติ
3. มีความคล่องตัวต่อการสื่อสารข้อมูลได้หลายชนิดทั้งแบบที่ไม่มีความเร่งด่วน เช่น การจัดส่งแฟ้มข้อมูล และแบบที่ต้องการรับประกันความเร่งด่วนของข้อมูล เช่น การสื่อสารแบบ real-time และทั้งการสื่อสารแบบเสียง (Voice) และข้อมูล (data)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.10.1 Encapsulation/Demultiplexing

การส่งข้อมูลผ่านในแต่ละเลเยอร์ แต่ละเลเยอร์จะทำการประกอบข้อมูลที่ได้รับมา กับข้อมูลส่วนควบคุมซึ่งถูกนำมาไว้ในส่วนหัวของข้อมูลเรียกว่า Header ภายใน Header จะบรรจุข้อมูลที่สำคัญของโปรโตคอลที่ทำการ Encapsulate เมื่อผู้รับได้รับข้อมูล ก็จะเกิดกระบวนการทำงานย้อนกลับคือโปรโตคอลเดียวกัน ทางฝั่งผู้รับก็จะได้รับข้อมูลส่วนที่เป็น Header ก่อนและนำไปประมวลและทราบว่าข้อมูลที่ตามมามีลักษณะอย่างไร ซึ่งกระบวนการย้อนกลับนี้เรียกว่า Demultiplexing



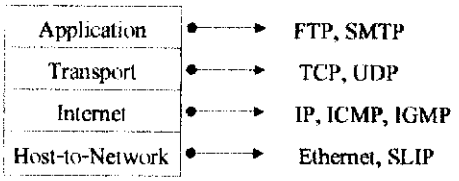
รูปที่ 2.11 ขั้นตอนการ Encapsulation และ Demultiplexing

ข้อมูลที่ผ่านการ Encapsulate ในแต่ละเลเยอร์มีชื่อเรียกแตกต่างกัน ดังนี้

- ข้อมูลที่มาจาก User หรือก็คือข้อมูลที่ User เป็นผู้ป้อนให้กับ Application เรียกว่า User Data
- เมื่อแอพพลิเคชั่นได้รับข้อมูลจาก user ก็จะนำมาประกอบกับส่วนหัวของแอพพลิเคชั่น เรียกว่า Application Data และส่งต่อไปยังโปรโตคอล TCP
- เมื่อโปรโตคอล TCP ได้รับ Application Data ก็จะนำมารวมกับ Header ของ โปรโตคอล TCP เรียกว่า TCP Segment และส่งต่อไปยังโปรโตคอล IP
- เมื่อโปรโตคอล IP ได้รับ TCP Segment ก็จะนำมารวมกับ Header ของ โปรโตคอล IP เรียกว่า IP Datagram และส่งต่อไปยังเลเยอร์ Host-to-Network Layer
- ในระดับ Host-to-Network จะนำ IP Datagram มาเพิ่มส่วน Error Correction และ flag เรียกว่า Ethernet Frame ก่อนจะแปลงข้อมูลเป็นสัญญาณไฟฟ้า ส่งผ่านสายสัญญาณที่เชื่อมโยงอยู่ต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในแต่ละเลเยอร์ของโครงสร้าง TCP/IP สามารถอธิบายได้ดังนี้



รูปที่ 2.12 โครงสร้าง TCP/IP

1. ชั้นโฮสต์-เครือข่าย (Host-to-Network Layer)

โปรโตคอลสำหรับการควบคุมการสื่อสารในชั้นนี้เป็นสิ่งที่ไม่มีการกำหนดรายละเอียดอย่างเป็นทางการ หน้าที่หลักคือการรับข้อมูลจากชั้นสื่อสาร IP มาแล้วส่งไปยังโหนดที่ระบุไว้ในเส้นทางเดินข้อมูล ทางด้านผู้รับก็จะทำงานในทางกลับกัน คือรับข้อมูลจากสายสื่อสารแล้วนำส่งให้กับโปรแกรมในชั้นสื่อสาร

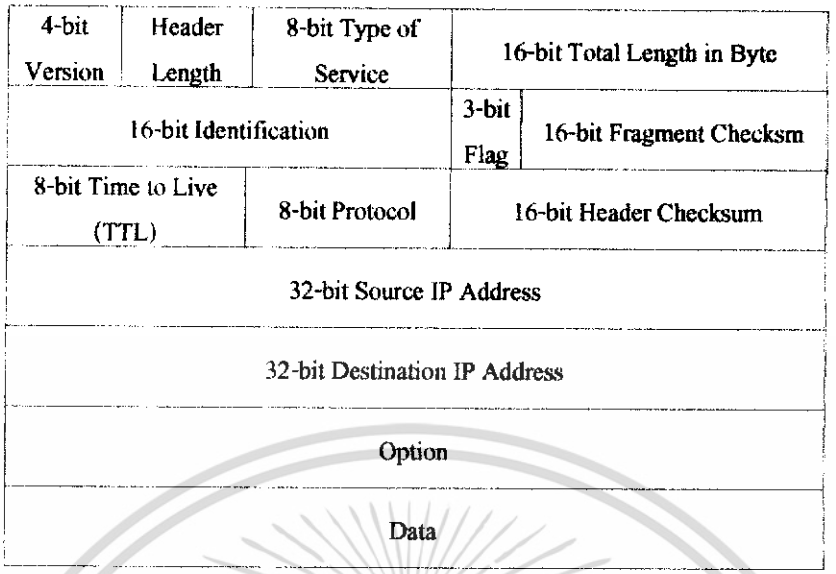
2. ชั้นสื่อสารอินเทอร์เน็ต (The Internet Layer)

ใช้ประเภทของระบบการสื่อสารที่เรียกว่า ระบบเครือข่ายแบบสลับช่องสื่อสารระดับแพ็กเก็ต (packet-switching network) ซึ่งเป็นการติดต่อแบบไม่ต่อเนื่อง (Connectionless) หลักการทำงานคือการปล่อยให้ข้อมูลขนาดเล็กที่เรียกว่า แพ็กเก็ต (Packet) สามารถไหลจากโหนดผู้ส่งไปตามโหนดต่างๆ ในระบบจนถึงจุดหมายปลายทางได้โดยอิสระ หากว่ามีการส่งแพ็กเก็ตออกมาเป็นชุดโดยมีจุดหมายปลายทางเดียวกันในระหว่างการเดินทางในเครือข่าย แพ็กเก็ตแต่ละตัวในชุดนี้ก็จะเป็นอิสระแก่กันและกัน ดังนั้น แพ็กเก็ตที่ส่งไปถึงปลายทางอาจจะไม่เป็นไปตามลำดับก็ได้

a. IP (Internet Protocol) IP เป็นโปรโตคอลในระดับเน็ตเวิร์คเลเยอร์ ทำหน้าที่จัดการเกี่ยวกับแอดเดรสและข้อมูล และควบคุมการส่งข้อมูลบางอย่างที่ใช้ในการหาเส้นทางของแพ็กเก็ต ซึ่งกลไกในการหาเส้นทางของ IP จะมีความสามารถในการหาเส้นทางที่ดีที่สุด และสามารถเปลี่ยนแปลงเส้นทางได้ในระหว่างการส่งข้อมูล และมีระบบการแยกและประกอบดาต้าแกรม (datagram) เพื่อรองรับการส่งข้อมูลระดับ data link ที่มีขนาด MTU (Maximum Transmission Unit) ที่แตกต่างกัน ทำให้สามารถนำ IP ไปใช้บนโปรโตคอลอื่นได้หลากหลาย เช่น Ethernet ,Token Ring หรือ Apple Talk

การเชื่อมต่อของ IP เพื่อทำการส่งข้อมูล จะเป็นแบบ connectionless หรือเกิดเส้นทางการเชื่อมต่อในทุกๆครั้งของการส่งข้อมูล 1 ดาต้าแกรม โดยจะไม่ทราบถึงข้อมูลดาต้าแกรมที่ส่งก่อนหน้าหรือส่งตามมา แต่การส่งข้อมูลใน 1 ดาต้าแกรม อาจจะเกิดการส่งได้หลายครั้งในกรณีที่มีการแบ่งข้อมูลออกเป็นส่วนย่อยๆ (fragmentation) และถูกนำไปรวมเป็นดาต้าแกรมเดิมเมื่อถึงปลายทาง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.13 IP Header

เฮดเดอร์ของ IP โดยปกติจะมีขนาด 20 bytes ยกเว้นในกรณีที่มีการเพิ่ม option บางอย่าง ฟิลด์ของเฮดเดอร์ IP จะมีความหมายดังนี้

a. **Version** : หมายเลขเวอร์ชันของโปรโตคอล ที่ใช้งานในปัจจุบันคือ เวอร์ชัน 4 (IPv4)
b. **Header Length** : ความยาวของเฮดเดอร์ โดยทั่วไปถ้าไม่มีส่วน option จะมีค่าเป็น 5 (5*32 bit)
c. **Type of Service (TOS)** : ใช้เป็นข้อมูลสำหรับเราเตอร์ในการตัดสินใจเลือกการเราท์ข้อมูลในแต่ละดาต้าแกรม แต่ในปัจจุบันไม่ได้มีการนำไปใช้งานแล้ว
d. **Length** : ความยาวทั้งหมดเป็นจำนวนไบต์ของดาต้าแกรม ซึ่งด้วยขนาด 16 บิตของฟิลด์ จะหมายถึงความยาวสูงสุดของดาต้าแกรม คือ 65535 byte (64k) แต่ในการส่งข้อมูลจริง ข้อมูลจะถูกแยกเป็นส่วนๆตามขนาดของ MTU ที่กำหนดในลิงค์เลเยอร์ และนำมารวมกันอีกครั้งเมื่อส่งถึงปลายทางแอพพลิเคชั่นส่วนใหญ่จะมีขนาดของดาต้าแกรมไม่เกิน 512 byte
e. **Identification** : เป็นหมายเลขของดาต้าแกรมในกรณีที่มีการแยกดาต้าแกรมเมื่อข้อมูลส่งถึงปลายทางจะนำข้อมูลที่มี identification เดียวกันมารวมกัน
f. **Flag** : ใช้ในกรณีที่มีการแยกดาต้าแกรม
g. **Fragment offset** : ใช้ในการกำหนดตำแหน่งข้อมูลในดาต้าแกรมที่มีการแยกส่วน เพื่อให้สามารถนำกลับมาเรียงต่อกันได้อย่างถูกต้อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

h. **Time to live (TTL)** : กำหนดจำนวนครั้งที่มากที่สุดที่ดาต้าแกรมจะถูกส่งระหว่าง hop (การส่งผ่านข้อมูลระหว่างเน็ตเวิร์ค) เพื่อป้องกันไม่ให้เกิดการส่งข้อมูลโดยไม่สิ้นสุด โดยเมื่อข้อมูลถูกส่งไป 1 hop จะทำการลดค่า TTL ลง 1 เมื่อค่าของ TTL เป็น 0 และข้อมูลยังไม่ถึงปลายทาง ข้อมูลนั้นจะถูกยกเลิก และเราเตอร์สุดท้ายจะส่งข้อมูล ICMP แจ้งกลับมายังต้นทางว่าเกิด time out ในระหว่างการส่งข้อมูล

i. **Protocol** : ระบุโปรโตคอลที่ส่งในดาต้าแกรม เช่น TCP ,UDP หรือ ICMP

j. **Header checksum** : ใช้ในการตรวจสอบความถูกต้องของข้อมูลในเฮดเดอร์

k. **Source IP address** : หมายเลข IP ของผู้ส่งข้อมูล

l. **Destination IP address** : หมายเลข IP ของผู้รับข้อมูล

m. **Data** : ข้อมูลจากโปรโตคอลระดับบน

b. ICMP (Internet Control Message Protocol)

ICMP เป็นโปรโตคอลที่ใช้ในการตรวจสอบและรายงานสถานภาพของดาต้าแกรม (Datagram) ในกรณีที่เกิดปัญหากับดาต้าแกรม เช่น เราเตอร์ไม่สามารถส่งดาต้าแกรมไปถึงปลายทางได้ ICMP จะถูกส่งออกไปยังโฮสต้นทางเพื่อรายงานข้อผิดพลาด ที่เกิดขึ้น อย่างไรก็ดี ไม่มีอะไรรับประกันได้ว่า ICMP Message ที่ส่งไปจะถึงผู้รับจริงหรือไม่ หากมีการส่งดาต้าแกรมออกไปแล้วไม่มี ICMP Message ฟ้อง Error กลับมา ก็แปลความหมายได้สองกรณีคือ ข้อมูลถูกส่งไปถึงปลายทางอย่างเรียบร้อย หรืออาจจะมีปัญหา ในการสื่อสารทั้งการส่งดาต้าแกรม และ ICMP Message ที่ส่งกลับมาก็มีปัญหาระหว่างทางก็ได้ ICMP จึงเป็นโปรโตคอลที่ไม่มีความน่าเชื่อถือ (unreliable) ซึ่งจะเป็นหน้าที่ของ โปรโตคอลในระดับสูงกว่า Network Layer ในการจัดการให้การสื่อสารนั้นๆ มีความน่าเชื่อถือ ในส่วนของ ICMP Message จะประกอบด้วย Type ขนาด 8 บิต Checksum ขนาด 16 บิต และส่วนของ Content ซึ่งจะมีขนาดแตกต่างกันไปตาม Type และ Code ดังรูป

8-bit Type	8-bit Code	16-bit Checksum
ICMP Content		

รูปที่ 2.14 ICMP Header

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. ชั้นสื่อสารนำส่งข้อมูล (Transport Layer)

แบ่งเป็นโปรโตคอล 2 ชนิดตามลักษณะ ลักษณะแรกเรียกว่า Transmission Control Protocol (TCP) เป็นแบบที่มีการกำหนดช่วงการสื่อสารตลอดระยะเวลาการสื่อสาร (connection-oriented) ซึ่งจะยอมให้มีการส่งข้อมูลเป็นแบบ Byte stream ที่ไว้ใจได้โดยไม่มีข้อผิดพลาด ข้อมูลที่มีปริมาณมากจะถูกแบ่งออกเป็นส่วนเล็กๆ เรียกว่า message ซึ่งจะถูกส่งไปยังผู้รับผ่านทางชั้นสื่อสารของอินเทอร์เน็ต ทางฝ่ายผู้รับจะนำ message มาเรียงต่อกันตามลำดับเป็นข้อมูลตัวเดิม TCP ยังมีความสามารถในการควบคุมการไหลของข้อมูลเพื่อป้องกันไม่ให้ผู้ส่ง ส่งข้อมูลเร็วเกินกว่าที่ผู้รับจะทำงานได้ทันอีกด้วย

โปรโตคอลการนำส่งข้อมูลแบบที่สองเรียกว่า UDP (User Datagram Protocol) เป็นการติดต่อแบบไม่ต่อเนื่อง (connectionless) มีการตรวจสอบความถูกต้องของข้อมูลแต่จะไม่มีการแจ้งกลับไปยังผู้ส่ง จึงถือได้ว่าไม่มีการตรวจสอบความถูกต้องของข้อมูล อย่างไรก็ตาม วิธีการนี้มีข้อดีในด้านความรวดเร็วในการส่งข้อมูล จึงนิยมใช้ในระบบผู้ให้และผู้ใช้บริการ (client/server system) ซึ่งมีการสื่อสารแบบ ถามตอบ (request/reply) นอกจากนั้นยังใช้ในการส่งข้อมูลประเภทภาพเคลื่อนไหวหรือการส่งเสียง (voice) ทางอินเทอร์เน็ต

a. UDP : (User Datagram Protocol)

เป็นโปรโตคอลที่อยู่ใน Transport Layer เมื่อเทียบกับโมเดล OSI โดยการส่งข้อมูลของ UDP นั้นจะเป็นการส่งครั้งละ 1 ชุดข้อมูล เรียกว่า UDP datagram ซึ่งจะไม่มีความสัมพันธ์กันระหว่างดาต้าแกรมและจะไม่มีกลไกการตรวจสอบความสำเร็จในการรับส่งข้อมูล

กลไกการตรวจสอบโดย checksum ของ UDP นั้นเพื่อเป็นการป้องกันข้อมูลที่อาจจะถูกแก้ไขหรือมีความผิดพลาดระหว่างการส่ง และหากเกิดเหตุการณ์ดังกล่าว ปลายทางจะได้รู้ว่ามีข้อผิดพลาดเกิดขึ้น แต่มันจะเป็นการตรวจสอบเพียงฝ่ายเดียวเท่านั้น โดยในข้อกำหนดของ UDP หากพบว่า Checksum Error ก็ให้ผู้รับปลายทางทำการทิ้งข้อมูลนั้น แต่จะไม่มีการแจ้งกลับไปยังผู้ส่งแต่อย่างใด การรับส่งข้อมูลแต่ละครั้งหากเกิดข้อผิดพลาดในระดับ IP เช่น ส่งไม่ถึง, หมดเวลา ผู้ส่งจะได้รับ Error Message จากระดับ IP เป็น ICMP Error Message แต่เมื่อข้อมูลส่งถึงปลายทางถูกต้อง แต่เกิดข้อผิดพลาดในส่วนของ UDP เอง จะไม่มีการยืนยัน หรือแจ้งให้ผู้ส่งทราบแต่อย่างใด

16-bit Source Port	16-bit Destination Port
Length	Checksum
Data	

รูปที่ 2.15 UDP Header

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

มีรายละเอียด ดังนี้

- **Source Port Number** : หมายเลขพอร์ตต้นทางที่ส่งดาต้าแกรมนี้
- **Destination Port Number** : หมายเลขพอร์ตปลายทางที่จะเป็นผู้รับดาต้าแกรม
- **UDP Length** : ความยาวของดาต้าแกรม ทั้งส่วน Header และ data นั่นหมายความว่าค่าที่น้อยที่สุดในฟิลด์นี้คือ 8 ซึ่งเป็นขนาดของ Header
- **Checksum** : เป็นตัวตรวจสอบความถูกต้องของ UDP datagram และจะนำข้อมูลบางส่วนใน IP Header มาคำนวณด้วย

b. TCP : (Transmission Control Protocol)

อยู่ใน Transport Layer เช่นเดียวกับ UDP ทำหน้าที่จัดการและควบคุมการรับส่งข้อมูล ซึ่งมีความสามารถและรายละเอียดมากกว่า UDP โดยดาต้าแกรมของ TCP จะมีความสัมพันธ์ต่อเนื่องกัน และมีกลไกควบคุมการรับส่งข้อมูลให้มีความถูกต้อง (reliable) และมีการสื่อสารอย่างเป็นกระบวนการ (connection-oriented)

<table>
<tr><td colspan="8">16-bit Source Port number</td><td colspan="8">16-bit Source Destination Port</td></tr>
<tr><td colspan="16">32-bit Sequence number</td></tr>
<tr><td>Header Length</td><td colspan="2">6-bit Reserved</td><td>URG</td><td>ACK</td><td>PUSH</td><td>Reset</td><td>SYN</td><td>FIN</td><td colspan="8">16-bit Window size</td></tr>
<tr><td colspan="8">16-bit TCP Checksum</td><td colspan="8">16-bit Urgent Pointer</td></tr>
<tr><td colspan="16">TCP Option</td></tr>
<tr><td colspan="16">Data</td></tr>
</table>

รูปที่ 2.16 TCP Header

มีรายละเอียด ดังนี้

- Source Port Number : หมายเลขพอร์ตต้นทางที่ส่งดาต้าแกรมนี้
- Destination Port Number : หมายเลขพอร์ตปลายทางที่จะเป็นผู้รับดาต้าแกรม
- Sequence Number : ฟิลด์ที่ระบุหมายเลขลำดับอ้างอิงในการสื่อสารข้อมูลแต่ละครั้ง เพื่อใช้ในการแยกแยะว่าเป็นข้อมูลของชุดใด และนำมาจัดลำดับได้ถูกต้อง
- Acknowledgment Number : ทำหน้าที่เช่นเดียวกับ Sequence Number แต่จะใช้ในการตอบรับ
- Header Length : โดยปกติความยาวของเฮดเดอร์ TCP จะมีความยาว 20 ไบต์ แต่อาจจะมากกว่านั้น ถ้ามีข้อมูลในฟิลด์ option แต่ต้องไม่เกิน 60 ไบต์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- Flag : เป็นข้อมูลระดับบิตที่อยู่ในเฮดเดอร์ TCP โดยใช้เป็นตัวบอกคุณสมบัติของแพ็กเก็ต TCP ขณะนั้นๆ และใช้เป็นตัวควบคุมจังหวะการรับส่งข้อมูลด้วย ซึ่ง Flag มีอยู่ทั้งหมด 6 บิต แบ่งได้ดังนี้

Type	Descreiption
URG	ใช้บอกความหมายเป็นข้อมูลด่วน และมีข้อมูลพิเศษมาด้วย(อยู่ใน Urgent pointer
ACK	แสดงว่าข้อมูลในฟิลด์ Acknowledge Number นำมาใช้งานได้
DSH	เป็นการแจ้งให้ผู้รับข้อมูลทราบว่าควรจะส่งข้อมูล Segment นี้ไปยัง Application ที่รออยู่โดยเร็ว
RST	ยกเลิกการติดต่อ (reset) เนื่องจากในกรณีที่เกิดการสับสนขึ้นด้วยเหตผลต่างๆเช่น โฮสต์มีปัญหา ให้เริ่มต้นสื่อสารกันใหม่
SYN	ใช้ในการเริ่มต้นติดต่อกับปลายทาง
FIN	ใช้ส่งเพื่อแจ้งให้ปลายทางทราบว่าหยุดการติดต่อ

Flag ในเฮดเดอร์ของ TCP มีความสำคัญในการกำหนดการทำงานของ TCP segment เนื่องจากข้อมูลในเฮดเดอร์ของ TCP จะมีข้อมูลครบถ้วนทั้งการรับและการส่งข้อมูล ซึ่งในการสทำงานแต่ละอย่างจะมีการใช้งานฟิลด์ไม่เหมือนกัน flag จะเป็นตัวกำหนดว่าให้ใช้งานฟิลด์ไหน เช่น ฟิลด์ Acknowledgment number จะไม่ถูกใช้ในขั้นตอนการเริ่มต้นการเชื่อมต่อ แต่จะมีข้อมูลในฟิลด์ ซึ่งเป็นข้อมูลที่ไม่มีความหมายใดๆ ซึ่งถ้าไม่มี flag เป็นตัวกำหนดก้ออาจจะมีการนำข้อมูลมาใช้ และก่อให้เกิดความผิดพลาดได้

4. ชั้นสื่อสารการประยุกต์ (Application Layer)

มีโปรโตคอลสำหรับสร้างจอเทอร์มินัลเสมือน เรียกว่า TELNET โปรโตคอลสำหรับการจัดการแฟ้มข้อมูล เรียกว่า FTP และโปรโตคอลสำหรับการให้บริการจดหมายอิเล็กทรอนิกส์ เรียกว่า SMTP โดยโปรโตคอลสำหรับสร้างจอเทอร์มินัลเสมือนช่วยให้ผู้ใช้สามารถติดต่อกับเครื่องโฮสต์ที่อยู่ไกลออกไปโดยผ่านอินเทอร์เน็ต และสามารถทำงานได้เสมือนกับว่ากำลังนั่งทำงานอยู่ที่เครื่องโฮสต์นั้น โปรโตคอลสำหรับการจัดการแฟ้มข้อมูลช่วยในการคัดลอกแฟ้มข้อมูลมาจากเครื่องอื่นที่อยู่ในระบบเครือข่ายหรือส่งสำเนาแฟ้มข้อมูลไปยังเครื่องใดๆก็ได้ โปรโตคอลสำหรับให้บริการจดหมายอิเล็กทรอนิกส์ช่วยในการจัดส่งข้อความไปยังผู้ใช้ในระบบ หรือรับข้อความที่มีผู้ส่งเข้ามา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.11 มาตรฐานการเชื่อมต่อทางคลื่นวิทยุของเครือข่าย GPRS

เนื่องจาก GPRS ถือเป็นมาตรฐานการสื่อสารข้อมูลประเภทหนึ่ง ดังนั้นจึงสามารถแสดงภาพการจัดแบ่งหน้าที่การทำงานของวงจรทางฮาร์ดแวร์และซอฟท์แวร์ โดยอ้างอิงตามแบบจำลอง OSI (Open System Integration) ซึ่งจะเห็นได้ว่า หน้าที่การทำงานของสถานีฐานหรืออุปกรณ์ BTS ในการรองรับเทคโนโลยี GPRS นั้นเป็นเพียงการส่งผ่านข้อมูลทั้งที่เป็นสัญญาณควบคุม (Control Signaling) และสัญญาณข่าวสารข้อมูล (Data Signaling) ระหว่างเครื่องลูกข่าย GPRS กับอุปกรณ์ BSC และ TRAU เท่านั้น โดยข้อมูลที่ถูกรับส่งผ่านจุดเชื่อมต่อ Um และจุดเชื่อมต่อ Abis หรือ Agprs ซึ่งล้วนแล้วเกี่ยวข้องกับสถานีฐานจะถูกปรับเปลี่ยนรูปแบบของสัญญาณจากคลื่นความถี่วิทยุไปเป็นสัญญาณทางไฟฟ้าเพื่อส่งต่อไปให้กับอุปกรณ์ BSC เท่านั้น สถานีฐานจะไม่ทำหน้าที่ใดๆในการอ่านหรือแก้ไขเนื้อหาข้อมูลที่เป็นการรับส่งตามมาตรฐาน GPRS ซึ่งจะถูกห่อหุ้มโดยบล็อกข้อมูลประเภท MAC (Medium Access Control) และ RLC (Radio Link Control) ซึ่งจะถูกนำไปประมวลผลในเครื่องลูกข่ายโทรศัพท์เคลื่อนที่และอุปกรณ์ BSC/TRAU เท่านั้น

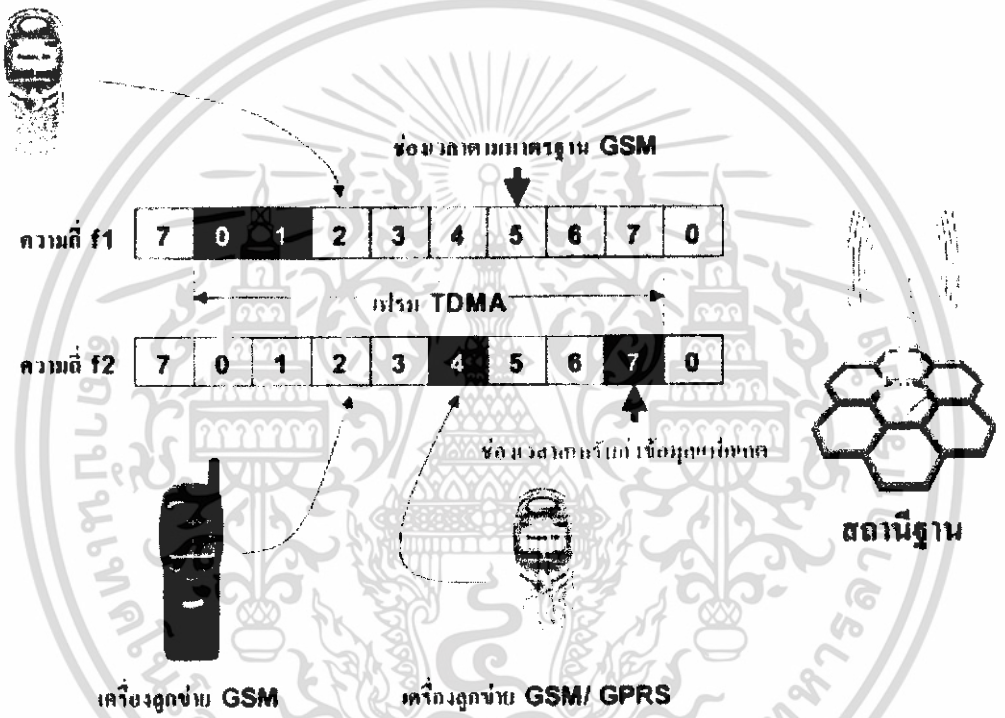
การสร้างและการจัดการช่องสื่อสารทางตรรก (Logical Channel) ซึ่งจะได้กล่าวถึงต่อไป เป็นการดำเนินการร่วมกันระหว่างอุปกรณ์ BSC กับเครื่องลูกข่ายโทรศัพท์เคลื่อนที่ GPRS โดยการกำหนดค่าบิตข้อมูลในระดับของโปรโตคอล RLC โดยมีการใช้โปรโตคอล MAC ในการควบคุมการส่งผ่านข้อมูลทุกๆ บิตที่เป็นการสื่อสารแบบ GPRS ให้มีความถูกต้อง ไม่เกิดความผิดพลาดของข้อมูล การจัดการกำหนดรายละเอียดของการใช้ช่องเวลา (Timeslot) ในแต่ละความถี่ของสถานีฐานให้เป็นช่องสื่อสารสำหรับโทรศัพท์เคลื่อนที่ GSM หรือ GPRS ก็เป็นอีกภาระหน้าที่หนึ่งของอุปกรณ์ เช่นเดียวกัน สำหรับอุปกรณ์ซึ่งติดตั้ง ไว้ภายในจะทำหน้าที่ปรับรูปแบบข้อมูลเพื่อทำการติดต่อสื่อสารแบบแพ็กเกต ซึ่งได้แก่อุปกรณ์ SGSN และ GGSN โดยผ่านโปรโตคอลชนิดใหม่ที่มีชื่อเรียกว่า ในกรณีของการเชื่อมต่อวงจรแบบสวิตช์วงจรระหว่างอุปกรณ์ กับชุมสายโทรศัพท์เคลื่อนที่มาตรฐาน

อุปกรณ์จะทำการติดต่อสื่อสารกับเครื่องลูกข่ายโทรศัพท์เคลื่อนที่ โดยตรวจสอบและควบคุมตำแหน่งที่อยู่ของเครื่องลูกข่าย และบริหารการเชื่อมต่อวงจรแบบสวิตช์แพ็กเกต เพื่อรับส่งข้อมูลไปยังโทรศัพท์เคลื่อนที่ เครื่องอื่น หรือไปยังเครือข่ายสื่อสารข้อมูลภายนอก เช่น เครือข่ายอินเตอร์เน็ต ทั้งนี้การติดต่อสื่อสารดังกล่าวจะอยู่ในรูปแบบของบล็อกข้อมูลที่ถูกควบคุมโดยโปรโตคอล ที่ถูกส่งมาจากอุปกรณ์ รวมถึงการเข้ารหัสข้อมูลเพื่อป้องกันการดักจับโดยบุคคลที่สามสำหรับการส่งสัญญาณควบคุมเพื่อดำเนินการต่างๆเกี่ยวกับการติดต่อสื่อสารระหว่างอุปกรณ์ กับเครือลูกข่าย นั้นจะเป็นหน้าที่ของโปรโตคอลที่มีชื่อว่า SNDCP (Subnet work-Dependent Convergence Protocol)

ข้อมูลที่ถูกถอดออกจากโปรโตคอล SDNCP แล้วจะเป็นข้อมูลที่อยู่ในบล็อกข้อมูลมาตรฐาน IP (Internet Protocol) หรือ X.25 ซึ่งเป็นมาตรฐานการสื่อสารที่สามารถใช้เชื่อมต่อกับเครือข่ายสื่อสารข้อมูลประเภทต่างๆได้ทั่วโลก เนื้อหาภายในบล็อกข้อมูลแบบ IP หรือ X.25 นั้นก็จะเป็นแอปพลิเคชั่นใช้งานต่างๆตามมาตรฐานการสื่อสารทั่วไป ไม่ว่าจะเป็น HTTP, FTP, Telnet ฯลฯ ข้อมูลเหล่านี้ซึ่งถือว่าอยู่ในระดับชั้นบนสุดของแบบจำลอง OSI ที่มีการรับส่งจากเครื่องลูกข่าย GPRS จะถูกเชื่อมต่อเพื่อการสื่อสารกับเครื่องลูกข่าย GPRS เครื่องอื่น หรือกับเครือข่ายภายนอก ตัวอย่างเช่น เครือข่ายอินเตอร์เน็ต เป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เนื่องจาก GPRS เป็นเทคโนโลยีที่ได้รับการออกแบบขึ้นบนมาตรฐานโทรศัพท์เคลื่อนที่GSM ดังนั้นจึงไม่มีการเปลี่ยนแปลงรายละเอียดของมาตรฐานการเชื่อมต่อทางวิทยุในระดับชั้นกายภาพแต่ประการใด แสดงโครงสร้างของการเชื่อมต่อทางคลื่นวิทยุระหว่างสถานีฐานและเครื่องลูกข่ายโทรศัพท์เคลื่อนที่ทั้งชนิด GSM และ GPRS โดยที่จะต้องมีการกำหนดความถี่ใช้งานอย่างน้อย 1 คู่สำหรับการติดต่อสื่อสารซึ่งความถี่ค่าหนึ่งมีค่าความถี่ต่ำกว่าจะถูกใช้ในการส่งสัญญาณจากเครื่องลูกข่ายไปยังสถานีฐาน เรียกว่าความถี่ขาขึ้น หรือ Uplink Frequency และความถี่ที่มีค่าสูงกว่าจะใช้ส่งสัญญาณในทิศทางกลับกัน เรียกว่า ความถี่ขาลง หรือ Downlink Frequency มาตรฐานโทรศัพท์เคลื่อนที่ในตระกูล GSM ทั้งย่านความถี่ 900 MHz 1800 MHz และ 1900 MHz ต่างก็มีการกำหนดค่าความแตกต่างของคู่ความถี่ไว้ กล่าวคือ GSM 900 กำหนดให้ค่าความถี่ขาขึ้นและขาลงแต่ละคู่อยู่ห่างกัน 45 MHz



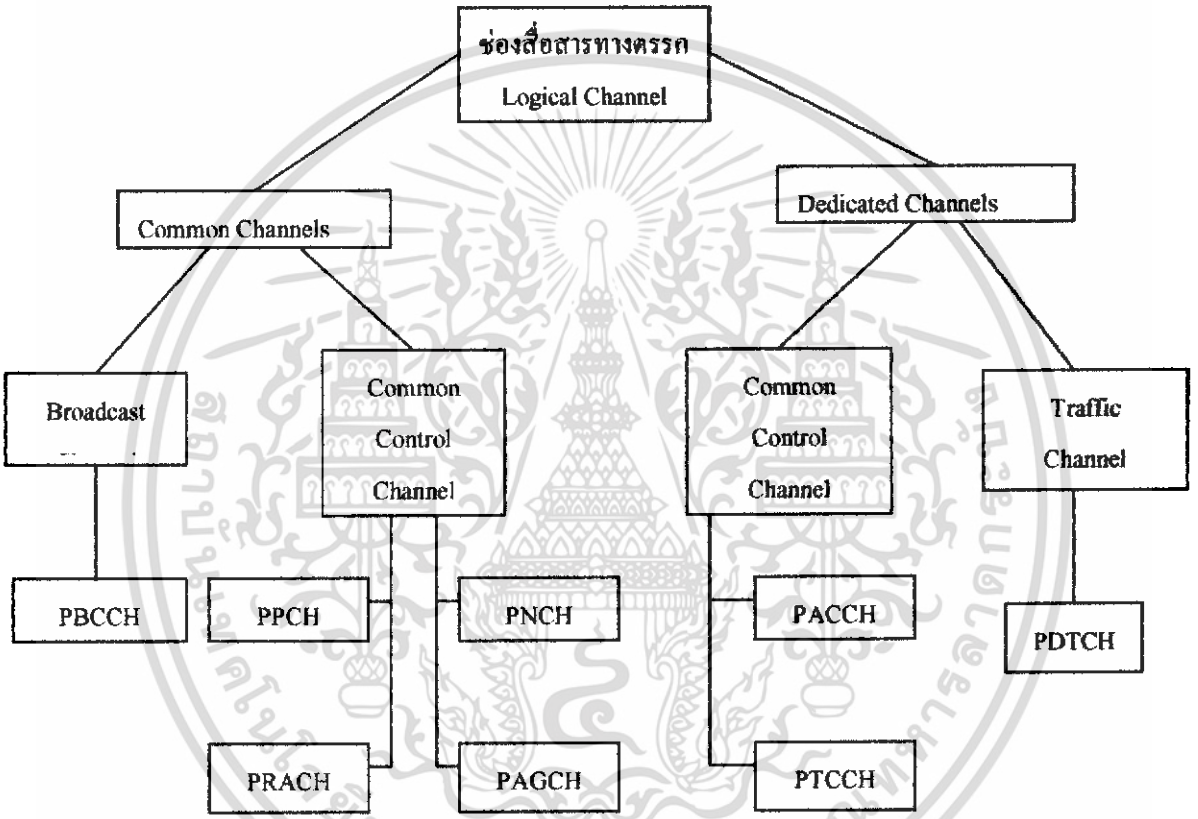
รูปที่ 2.17 การจัดสรรและบริหารจัดการช่องเวลาตามมาตรฐานโทรศัพท์เคลื่อนที่ GSM ให้รองรับการสื่อสารด้วยมาตรฐาน GSM โดยเป็นไปในลักษณะการร่วมใช้ความถี่เดียวกัน

สถานีฐานมีการใช้งานความถี่ 2 คู่ คือ F1 และ F2 สำหรับการรับส่งข้อมูลกับเครื่องลูกข่ายโทรศัพท์เคลื่อนที่ในพื้นที่แห่งหนึ่ง แต่ละความถี่จะมีการกำหนดโครงสร้างโดยแบ่งออกเป็นช่วงเวลาย่อยๆ เรียกว่า ช่องเวลา สำหรับให้เครื่องลูกข่ายแต่ละเครื่องสลับกันรับส่งข้อมูลเพื่อติดต่อสื่อสารกับสถานีฐาน โดยกำหนดให้มีโครงสร้างที่เรียกว่า เฟรม แบบ TDMA (Time Division Multiple Access) ซึ่งแต่ละเฟรมประกอบไปด้วยช่องเวลาจำนวน 8 ช่อง ในกรณีของสถานีฐานสนับสนุนเทคโนโลยี GPRS

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

นั้นทั้งเครื่องลูกข่ายที่เป็นแบบ GSM และเครื่องลูกข่ายแบบ GPRS ต่างก็สามารถติดต่อสื่อสารกับเครือข่ายโทรศัพท์เคลื่อนที่ผ่านทางช่องเวลาได้เท่าเทียมกัน

ในกรณีของมาตรฐานโทรศัพท์เคลื่อนที่ GSM จะมีการกำหนดนิยามช่องสื่อสารทางตรรกขึ้นซ้อนบนช่องเวลา โดยที่ช่องสื่อสารทางตรรกแต่ละประเภทจะมีรูปแบบการใช้งานที่แตกต่างกันสำหรับมาตรฐาน GPRS ก็มีการกำหนดช่องสื่อสารทางตรรกสำหรับการติดต่อสื่อสารแบบแพ็กเกต ซึ่งเป็นไปในลักษณะเดียวกันกับในมาตรฐาน GSM อย่างไรก็ตามรายละเอียดของการจัดการช่องสื่อสารทางตรรกและรูปแบบการใช้งานในมาตรฐาน GPRS ซึ่งสามารถจัดกลุ่มในการอธิบายได้ 3 ประเภทด้วยกันดังนี้



รูปที่ 2.18 การแบ่งกลุ่มของช่องสื่อสารทางตรรกตามมาตรฐานโทรศัพท์เคลื่อนที่ GPRS

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กลุ่มที่ 1: ช่องสื่อสาร Packet Common Control Channel หรือ PCCCH มีหน้าที่การทำงานโดยทั่วไปจะคล้ายคลึงกับกลุ่มช่องสื่อสารแบบ Common Control Channel (PCCCH) ของระบบโทรศัพท์เคลื่อนที่ GSM ในกลุ่มนี้สามารถแบ่งช่องสื่อสารทางตรรกออกได้เป็น 4 ชนิดดังนี้

1. Packet Random Access channel หรือ PRACH ใช้สำหรับการส่งข้อมูลจากเครื่องลูกข่าย GPRS ไปยังสถานีฐานในทิศทางขาขึ้น สำหรับเริ่มต้นการร้องขอสร้างวงจรเชื่อมต่อกับเครือข่ายโทรศัพท์เคลื่อนที่ GSM ในกรณีของการรับส่งข้อมูลแบบแพ็กเกต ช่องสื่อสารชนิดนี้มีความคล้ายคลึงกับช่องสื่อสารแบบ RACH ในกรณีของมาตรฐาน GSM
2. Packet Paging Channel หรือ PPCH ใช้สำหรับการส่งข้อมูลจากสถานีฐานไปยังเครื่องลูกข่าย GPRS ในกรณีที่มีการขอเชื่อมต่อผ่านเครือข่ายโทรศัพท์เคลื่อนที่ไปยังเครื่องลูกข่าย GPRS เครื่องใดเครื่องใดเครื่องหนึ่งภายในเครือข่าย สิ่งที่ควรทราบก็คือเครือข่าย GPRS จะใช้ช่องสื่อสาร PPCH นี้สำหรับการส่งสัญญาณเรียกติดต่อเฉพาะกรณีของเครื่องลูกข่าย GPRS คลาส A และ B เท่านั้น
3. Packet Access Grant Channel หรือ PACGH ใช้ในทิศทางการสื่อสารขาลง โดยสถานีฐานจะใช้ในการแจ้งยืนยันไปยังเครื่องลูกข่าย GPRS ว่าสามารถเริ่มต้นการสร้างวงจรเชื่อมต่อเพื่อทำการสื่อสารข้อมูลได้ ช่องสื่อสารนี้มีรูปแบบการใช้งานคล้ายคลึงกับช่องสื่อสารควบคุม AGCH ในมาตรฐาน GSM
4. Packet Notification Channel หรือ PNCH เป็นช่องสื่อสารชนิดใหม่สำหรับการสื่อสารแบบกระจายระหว่างสถานีฐานไปยังเครื่องลูกข่าย GRRS หลายๆเครื่องพร้อมๆกัน สำหรับการส่งข่าวสารแจ้งรายละเอียดการจัดสรรช่องสื่อสารให้กับโทรศัพท์เคลื่อนที่ GRRS แต่ละเครื่อง

กลุ่มที่ 2: ช่องสื่อสาร Packet Broadcast Control Channel หรือ PBCCH ในกลุ่มนี้มีช่องสื่อสารควบคุมเพียงชนิดเดียว PBCCH เป็นช่องสื่อสารในทิศทางขาลง ซึ่งสถานีฐานจะทำการส่งขอมูลที่สำคัญเกี่ยวกับรายละเอียดการเชื่อต่อและติดต่อสื่อสารกับเครือข่ายให้กับเครื่องลูกข่าย GPRS ทั้งนี้เครื่องลูกข่าย GRRS จะทราบว่าสามารถรออ่านค่าข้อมูลที่อยู่ในช่องสื่อสารควบคุม PBCCH ในช่วงเวลาใดๆ ได้จากการส่งข่าวจากช่องสื่อสารควบคุม BCCH ซึ่งอยู่ในช่องเวลามาตรฐานของการสื่อสารแบบ GSM ผู้ให้บริการระบบเครือข่ายสามารถเลือกกำหนดได้ว่าจะรวมให้มีการกระจายข่าวสารที่จำเป็นในการติดต่อสื่อสารแบบแพ็กเกต (GPRS) และสวิตช์วงจร (GSM) โดยรวมส่ง BCCH ก็ได้

กลุ่มที่ 3: ช่องสื่อสาร Packet Traffic Channel หรือ PTCH เป็นกลุ่มของช่องสื่อสารที่ทำหน้าที่เกี่ยวข้องกับการรับส่งข้อมูลของผู้ใช้บริการซึ่งมีการรับส่งในรูปแบบของแพ็คเกต ประกอบไปด้วยช่องสื่อสารควบคุม 3 ชนิด ดังนี้

1. Packet Data Traffic Channel (PDTCH) เป็นช่องสื่อสารที่มีอยู่ทั้งในทิศทางการสื่อสารขาขึ้นและขาลง โดยทำหน้าที่เป็นช่องทางในการรับส่งข้อมูลของผู้ใช้บริการระหว่างเครื่องลูกข่าย GPRS และสถานีฐาน ตำแหน่งการจัดสรรช่องสื่อสาร PDTCH ในช่องเวลาบนมาตรฐานเฟรมแบบ TDMA นั้นจะเป็นไปตามรูปแบบซึ่งจะได้กล่าวถึงต่อไป แต่หากจะกล่าวโดยง่ายแล้วก็คือช่องสื่อสาร PDTCH หนึ่งช่อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จะถูกจัดสรรบนช่องเวลาหนึ่งช่อง เครื่องลูกข่าย GPRS 1 เครื่อง จะสามารถร้องขอจากระบบเครือข่ายเพื่อทำการรับส่งข้อมูล เพื่อทำการรับส่งข้อมูลผ่านช่องสื่อสาร PTDCH เพียงหนึ่งช่องหรือมากกว่าหนึ่งช่องได้ในเวลาเดียวกัน นอกจากนั้นมาตรฐาน GPRS ยังสามารถกำหดให้เครื่องลูกข่าย GPRS หลายๆ เครื่อง แบ่งกันใช้ช่องสื่อสาร PDTCH ได้มากกว่า 1 ช่องพร้อมๆกัน โดยให้อยู่ภายใต้การควบคุมจากระบบเครือข่ายผ่านทางช่องสื่อสารควบคุม PNCH ถือเป็นช่องสื่อสารที่มีความคล้ายคลึงกันกับช่องสื่อสาร TCH ในกรณีของมาตรฐาน GSM

2. Packet Associated Control Channel (PACCH) เป็นช่องสื่อสารควบคุมที่ถูกจัดให้ฝังตัวอยู่ร่วมกับช่องสื่อสารรับส่งข้อมูล ซึ่งมีรายละเอียดการจัดวางช่องสื่อสารดังแสดงในรูปที่ 2.13 โดยจะได้กล่าวต่อไปมีอยู่ทั้งในทิศทางการสื่อสารขาขึ้นและขาลง โดยทั้งเครื่องลูกข่าย GPRS และสถานีฐานจะใช้งานช่องสื่อสารควบคุมดังกล่าว ในกรณีของการควบคุมกำลัง (Power Control) และการติดต่อสื่อสารในขณะที่เครื่องลูกข่าย GPRS เครื่องนั้นๆอยู่ในช่วงของการทำงานรับส่งข้อมูลแบบแพ็กเกตกับสถานีฐาน

3. Packet Timing-Advance Control Channel (PTCCH) เป็นช่องสื่อสารควบคุมชนิดใหม่ที่ถูกกำหนดให้มีการใช้งานในการปรับแก้ไขฐานเวลาภายในเครื่องลูกข่าย GPRS โดยมีวัตถุประสงค์เพื่อป้องกันปัญหาการชนกันของข้อมูล(Collision)ที่ถูกส่งจากเครื่องลูกข่าย GPRS แต่ละเครื่องมายังภาครับของสถานีฐานโดยเครื่องลูกข่าย GPRS แต่ละเครื่องจะทำการส่งข้อมูลประมาณค่าความแตกต่างของช่วงเวลา(Timing advance หรือ TA) ที่คำนวณได้กลับมายังสถานีฐาน เรื่องของTA นั้นเป็นมาตรฐานดั้งเดิมที่ถูกกำหนดขึ้นในระบบโทรศัพท์เคลื่อนที่ GSM ใช้ในการแสดงระยะห่างระหว่างเครื่องลูกข่ายแต่ละเครื่องกับสถานีฐาน

เครื่องลูกข่ายที่อยู่ห่างจากสถานีฐานย่อมมีโอกาสส่งข้อมูลกลับมายังฐานเวลาในลักษณะเวลาที่เหลื่อมกับตำแหน่งช่องเวลาที่ควรจะเป็นได้มากกว่าเครื่องลูกข่ายที่อยู่ใกล้สถานีฐานมากกว่า ทั้งนี้ในการป้องกันการเกิดปัญหาของข้อมูลเหลื่อม ซึ่งทำให้เกิดการปะทะกันของข้อมูลจากเครื่องลูกข่ายที่มีการรับส่งข้อมูลในช่องเวลาที่ติดกันนั้น สำหรับมาตรฐาน GPRS สถานีฐานจะทำการคำนวณฐานเวลาที่เหมาะสมให้กับเครื่องลูกข่ายแต่ละเครื่อง พร้อมทั้งทำการส่งสัญญาณแจ้งเพื่อขอปรับเปลี่ยนฐานเวลาไปยังเครื่องลูกข่ายแต่ละเครื่องโดยผ่านทางช่องสื่อสาร PTCCH เครื่องลูกข่ายที่อยู่ห่างจากสถานีฐานมาก ซึ่งมีโอกาสจะส่งข้อมูลมาถึงสถานีฐานล่าช้าก็จะได้รับการปรับฐานเวลาภายในตัวเครื่องให้ทำงานเร็วขึ้นกว่าเครื่องลูกข่ายที่อยู่ใกล้สถานีฐานมากกว่า ซึ่งเป็นการปรับปัญหาเวลาของเครื่องลูกข่าย GPRS ที่มีการรับส่งข้อมูลในช่องเวลาที่ 7 ให้ส่งข้อมูลกลับมายังสถานีฐานในเวลาที่ตรงกับช่องเวลาของตนพอดี

ตามมาตรฐาน GSM กำหนดไว้ว่าสถานีฐานแต่ละแห่งจะมีรัศมีครอบคลุมได้ไกลที่สุดไม่เกิน 35 กิโลเมตร ทั้งนี้ข้อกำหนดดังกล่าวเกี่ยวข้องกับการออกแบบรูปแบบของ Burst แบบ Random Access ก็คือปัจจัยในเรื่องของการกำหนดขอบเขตของเซลสถานีฐานก็นับเป็นอีกหนึ่งปัจจัยต่อค่า TA ของเครื่องลูกข่าย กลไกในการปรับฐานเวลาของโทศัพท์เคลื่อนที่ GSM และ GPRS จะทำงานได้ดีหรือไม่ จึงขึ้นอยู่กับแนวทางในการออกแบบวางเครือข่ายของผู้ให้บริการเป็นสำคัญ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำหรับแนวทางในการใช้งานช่องสื่อสารทางตรรกที่มีอยู่เป็นจำนวนมากนี้ มาตรฐานระบบโทรศัพท์เคลื่อนที่ GPRS ได้แบ่งแนวทางในการจัดสรรใช้งานออกเป็น 2 ระยะ คือ ระยะที่ 1 ซึ่งอยู่ในช่วงปัจจุบันที่มาตรฐาน GPRS ถือว่าอยู่ในช่วงเริ่มต้น กำหนดให้มาการใช้งานช่องสื่อสารในกลุ่ม CCCH ตามมาตรฐาน GSM สำหรับทำหน้าที่แทนกลุ่มช่องสื่อสาร BCCH ตามมาตรฐาน GSM แทนช่องสื่อสาร PBCCH จนกว่าการพัฒนาเครื่องลูกข่ายรุ่นใหม่ๆ และสถานีฐาน GSM จะมีความก้าวหน้ามากขึ้น จึงจะประกาศใช้งานช่องสื่อสารควบคุมตามมาตรฐาน GPRS นั้นหมายความว่า ETSI ซึ่งเป็นผู้ว่ามาตรฐาน GPRS ได้มีการพัฒนาข้อกำหนดของช่องสื่อสารตามมาตรฐาน GSM ให้มีศักยภาพมากขึ้นเพื่อรองรับการทำงานตามมาตรฐาน GPRS ตัวอย่างเช่น ในช่องสื่อสารควบคุมแบบ BCCH นั้น ก็ได้มีการเพิ่มเติมประเภทของการส่งข้อมูลแจ้งสถานะและรายละเอียดการใช้งานชนิดใหม่ ที่มีชื่อว่า System Information Type13 หรือ SI13ขึ้นสำหรับใช้แจ้งข่าวสารไปยังเครื่องลูกข่าย GPRS โดยเฉพาะ

2.12 กระบวนการแสดงตัวต่อเครือข่าย GPRS

เมื่อใดก็ตามที่เครื่องลูกข่ายโทรศัพท์เคลื่อนที่ GPRS ต้องการมีใช้บริการสื่อสารข้อมูลผ่านเครือข่ายโทรศัพท์เคลื่อนที่ GPRS สิ่งที่ต้องเกิดขึ้นเป็นอันดับแรกก็คือ เครื่องลูกข่ายจะทำการแสดงตัวต่ออุปกรณ์ SGSN โดยที่อุปกรณ์ SGSN จะทำการตรวจสอบเงื่อนไขสำคัญ 3 ประการจากเครื่องลูกข่ายเพื่อเป็นการยืนยันสิทธิในการใช้บริการของเครื่องลูกข่าย พร้อมๆกับจัดเตรียมระดับคุณภาพของการให้บริการไปในเวลาเดียวกัน เงื่อนไขทั้ง 3 ประการได้แก่

1 Authorization เป็นการตรวจสอบว่าเลขหมายโทรศัพท์เคลื่อนที่ซึ่งแสดงตัวด้วยแผ่นซิมภายในตัวเครื่องลูกข่าย ได้รับอนุญาตให้ใช้บริการ GPRS หรือไม่ ผู้ใช้บริการที่ยังไม่ได้เปิดใบริการ GPRS จะไม่ได้รับสิทธิในการใช้บริการดังกล่าว โดยการตรวจสอบตั้งแต่ขั้นตอนนี้

2. Authentication เป็นการตรวจสอบความถูกต้องของเลขหมายผู้ใช้บริการ ว่าเป็นผู้ใช้บริการที่ลงทะเบียนภายในเครือข่ายโทรศัพท์เคลื่อนที่ GSM ถูกต้องหรือไม่ ในการนี้อุปกรณ์ SGSN จะทำหน้าที่ตรวจสอบความถูกต้องของแผ่นซิมโดยมีการทำงานร่วมกับอุปกรณ์ HLR/AuC ภายในเครือข่าย GSM ที่เก็บข้อมูลเกี่ยวกับผู้ใช้บริการ (เลขหมาย IMSI รหัส KI) เพื่อใช้กลไกการตรวจยืนยันผู้ใช้บริการตามขั้นตอนของเครือข่าย GSM ตามปกติ

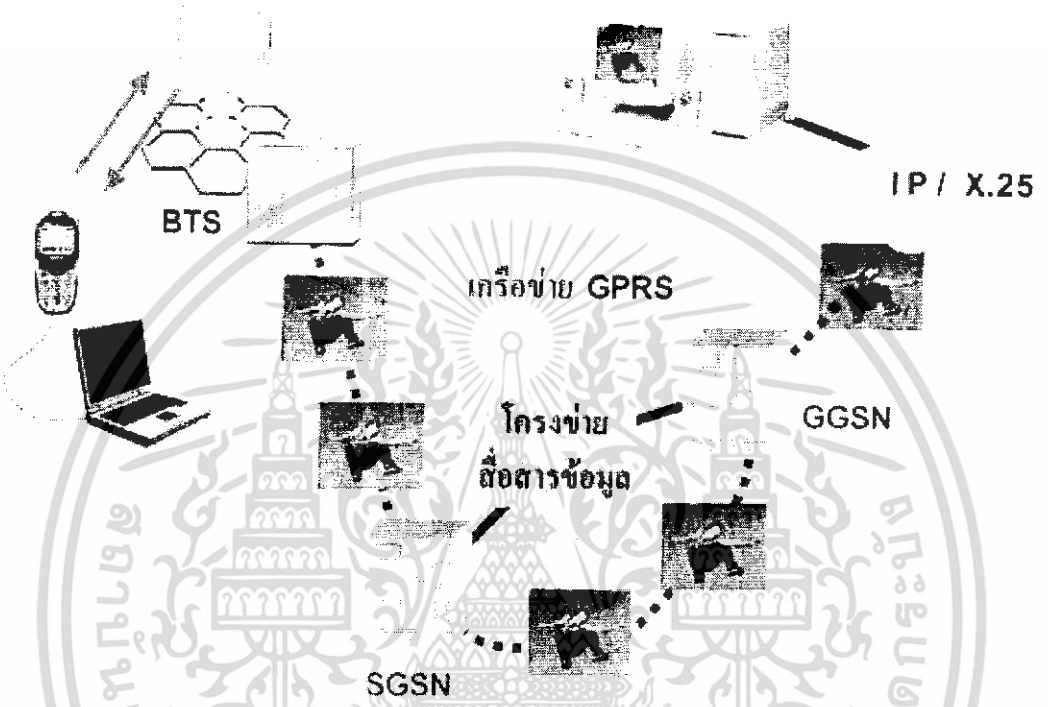
3. Quality of Service เพื่อตรวจสอบว่าผู้ใช้บริการที่ขอใช้บริการ GPRS แต่ละรายนั้นมีความจำนงขอใช้บริการรับส่งข้อมูลภายใต้การควบคุมคุณภาพในระดับใดเครือข่ายสามารถให้บริการภายใต้ระดับคุณภาพที่ต้องการนั้นได้หรือไม่ และหากสามารถให้บริการได้แล้ว จะทำให้คุณภาพในการสื่อสารข้อมูลของผู้ใช้บริการรายอื่นภายในเซลนั้นๆ ลดต่ำลงหรือไม่

หากอุปกรณ์ SGSN ตัดสินใจยอมรับเงื่อนไขข้างต้นทั้ง 3 ประการ และเปิดโอกาสให้เครื่องลูกข่ายสามารถใช้บริการ GPRS ได้ อุปกรณ์ SGSN จะทำการติดตามเครื่องลูกข่ายนั้นอยู่ตลอดเวลาตราบเท่าที่เครื่องลูกข่ายดังกล่าวยังเปิดเครื่องใช้งานอยู่ภายในพื้นที่ภายใต้การควบคุมของอุปกรณ์ SGSN นั้น ซึ่งอาจประกอบด้วยพื้นที่ RA จำนวนหลายๆ ชุด ทั้งนี้ก็เพื่อจะสามารถทราบได้ว่าเครื่องลูกข่ายดังกล่าวต้องการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ทุกครั้งที่ต้องการรับส่งข้อมูลเครื่องลูกข่ายดังกล่าวจะตองำการส่งข้อมูล PDP เพื่อแสดงรายละเอียดต่างๆ เกี่ยวกับการเชื่อให้กับอุปกรณ์ SGSN เป็นอีกครั้งหนึ่ง

นั่นหมายความว่า ทุกครั้งที่ผู้ให้บริการนำเครื่องลูกข่ายโทรศัพท์เคลื่อนที่ GPRS มาเลือกรายการเปิดใช้บริการ GPRS นั้น เครื่องลูกข่ายจำการแสดงตัวต่อเครือข่าย ซึ่งก็หมายถึงอุปกรณ์ SGSN ที่ควบคุม ดูแลพื้นที่ที่ใช้งานนั้น เมื่อใดที่เครื่องลูกข่ายมีการเคลื่อนที่ย้ายจากพื้นที่ใช้งานภายใน SGSN หนึ่งไปสู่อีก SGSN หนึ่ง ก็จะมีการโอนย้ายข้อมูลผู้ใช้บริการที่เก็บไว้ใน SGSN ต้นทางไปยัง SGSN ปลายทาง



รูปที่ 2.19 การรับส่งข้อมูลผ่านเครือข่าย GPRS สู่เครือข่ายคอมพิวเตอร์ภายนอก

สำหรับการสื่อสารระหว่างอุปกรณ์ GGSN และ SGSN นั้นมาตรฐาน GPRS ได้มีการกำหนดเทคนิคการสื่อสารชนิดพิเศษขึ้นเรียกว่า การส่งผ่านอุโมงค์ หรือ tunneling เพื่อช่วยให้ผู้ให้บริการเครือข่าย GSM/GPRS สามารถใช้ประโยชน์ของโครงข่ายสื่อสารข้อมูลที่มีอยู่แล้ว หรือที่ต้องดำเนินการเช่าใช้งานจากผู้อื่นในการเชื่อมต่ออุปกรณ์ GSN ต่างๆ ภายในเครือข่าย GPRS เข้าด้วยกัน ตัวอย่างเช่น เช่าโครงข่าย X.25 จากบริษัท A เพื่อใช้เชื่อมต่อกับอุปกรณ์ GGSN ที่อยู่ที่กรุงเทพฯกับอุปกรณ์ SGSN 4 ชุด ที่ติดตั้งอยู่ที่ กรุงเทพฯ,เชียงใหม่,ภูเก็ต,ขอนแก่น และชลบุรี ซึ่งเมื่อพิจารณาถึงการทำงานของโครงข่ายภายนอกที่ต้องมีการจัดรูปแบบแพ็คเกตข้อมูลตามมาตรฐานปกติแล้ว จะทำให้เกิดความสับสนต่อการรับส่งข้อมูลแพ็คเกต GPRS ระหว่างอุปกรณ์ GSN ด้วยกันได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.13 .เครื่องลูกข่าย GPRS

มาตรฐาน GPRS กำหนดให้ทั้งเครือข่ายสถานีฐานและเครื่องลูกข่าย สามารถติดต่อสื่อสารเพื่อรับส่งข้อมูล โดยมีการใช้ช่องเวลา (Timeslot) บนเฟรมมาตรฐาน ได้หลากหลายรูปแบบ โดยมีกำหนดคลาสการใช้งานหลายช่องเวลา (Multislot Class) ไว้ทั้งสิ้น 29 คลาส เครื่องลูกข่ายโทรศัพท์เคลื่อนที่ GPRS แต่ละรุ่นที่ได้รับการออกแบบมาสำหรับใช้งานในเชิงพาณิชย์ จะได้รับระบุว่าสนับสนุนคลาสการใช้งานหลายช่องเวลาที่ระดับใด ตารางที่ 2.8 เป็นการแสดงรายละเอียดทางเทคนิคของเครื่องลูกข่ายทั้ง 29 คลาส โดยระบุจำนวนช่องเวลาสูงสุดที่เครื่องลูกข่ายสามารถใช้ส่งข้อมูลไปยังสถานีฐาน (Tx) โดยในทางปฏิบัติเครื่องลูกข่ายสามารถใช้ส่งข้อมูลไปยังสถานีฐาน (Rx)โดยในทางปฏิบัติเครื่องลูกข่ายสามารถใช้ช่องเวลาได้ตั้งแต่ 0 ช่องไปจนถึง Tx และ Rx จะต้องไม่เกินค่าผลรวมในตารางช่องที่ 3

ตัวอย่างเช่นเครื่องลูกข่าย Siemens รุ่น 45 เป็นเครื่องลูกข่าย GPRS คลาส 8 หมายความว่า เครื่อง S45 มีความสามารถใช้ช่องเวลาบนเฟรม TDMA ขาขึ้น (Uplink) สำหรับการส่งข้อมูลไปยังเครือข่าย GPRS ได้พร้อมกันสูงสุด 1 ช่องเวลา และรับข้อมูลจากเครือข่ายได้โดยใช้ช่องเวลาบนเฟรม TDMA ขาลงได้สูงสุดถึง 4 ช่อง แต่อย่างไรก็ตามในทางปฏิบัติผลรวมของค่า Tx และ Rx จะต้องมีค่าไม่เกินกว่า 5 ช่องเวลา นั่นเอง เครื่องลูกข่ายดังกล่าวอาจมีจำนวนช่องเวลาในเฟรมข้อมูลขาขึ้นและขาลงในสัดส่วนเท่ากัน 0 ต่อ 4 ,0 ต่อ 3, 0 ต่อ 2 , 0 ต่อ 1 , 1 ต่อ 4 , 1 ต่อ 3, 1 ต่อ 2 , 1 ต่อ 1 , หรือ 1 ต่อ 0 ในช่วงเวลาใดเวลาหนึ่งก็ได้ อย่างไรก็ตามผู้ให้บริการเครือข่ายจะมีกำหนดจำนวนช่องได้มากตามขีดความสามารถของเครื่องลูกข่าย ทั้งนี้ก็เพื่อให้เกิดความมั่นใจว่ายังมีช่องเวลาเหลือพอสำหรับให้บริการเชื่อมต่อวงจรเพื่อการสนทนาบนเครือข่าย GSM ได้เพียงพอ ป้องกันการเกิดปัญหาวงจรเชื่อมต่อเต็มจนเกิดผลกระทบต่อความพอใจในการใช้บริการของผู้บริโภค

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คลาส Multislot	จำนวนช่องเวลาสูงสุด			ประเภท
	Rx	Tx	รวม	
1	1	1	2	1
2	2	1	3	1
3	2	2	3	1
4	3	1	4	1
5	2	2	4	1
6	3	2	4	1
7	3	3	4	1
8	4	1	5	1
9	3	2	5	1
10	4	2	5	1
11	4	3	ไม่ระบุ	1
12	4	4	ไม่ระบุ	1
13	3	3	ไม่ระบุ	2
14	6	4	ไม่ระบุ	2
15	7	5	ไม่ระบุ	2
16	8	6	ไม่ระบุ	2
17	6	7	ไม่ระบุ	2
18	6	8	ไม่ระบุ	2
19	6	2	ไม่ระบุ	1
20	6	3	ไม่ระบุ	1
21	6	4	ไม่ระบุ	1
22	8	4	ไม่ระบุ	1
23	8	6	ไม่ระบุ	1
24	8	2	ไม่ระบุ	1
25	8	3	ไม่ระบุ	1
26	8	4	ไม่ระบุ	1
27	8	4	ไม่ระบุ	1
28	8	6	ไม่ระบุ	1
29	8	8	ไม่ระบุ	1

ตารางที่ 2.8 รายละเอียดของการจัดกลุ่มเครื่องลูกข่าย GPRS ออกตามคลาสการใช้งานหลายช่องเวลา

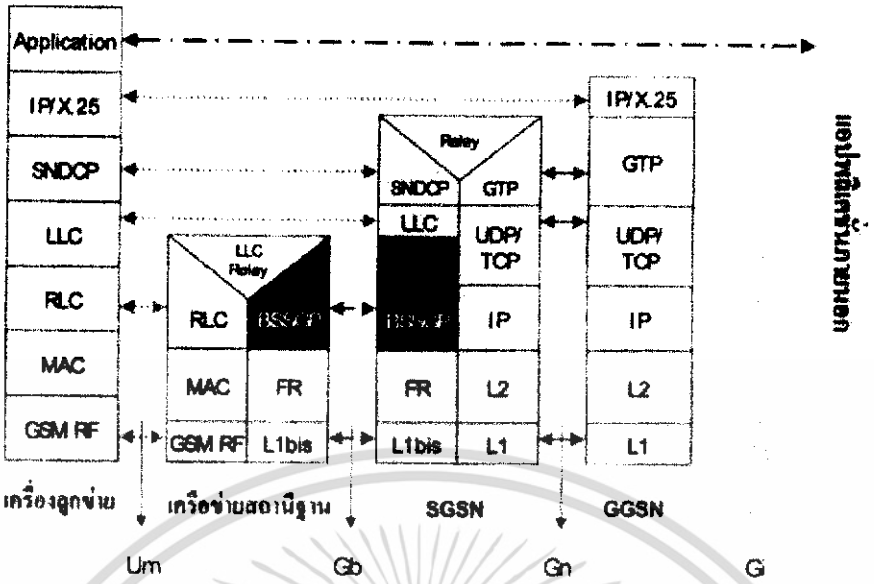
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำหรับประเภทของเครื่องลูกข่าย GPRS ที่มีการระบุไว้ใน ตารางที่ 2.8 โดยแบ่งออกเป็นกลุ่มที่ 1 และกลุ่มที่ 2 นั้น เครื่องลูกข่ายที่อยู่ในคลาสซึ่งตรงกับกลุ่มที่ 1 ไม่จำเป็นจะต้องมีการรับและส่งข้อมูลพร้อมๆกัน ในเวลาเดียวกัน แต่สำหรับเครื่องลูกข่ายในกลุ่มที่ 2 จำเป็นที่จะต้องมีการรับและส่งข้อมูลระหว่างเครื่องลูกข่ายและสถานีฐานพร้อมๆกันเวลาเดียวกัน อนึ่งผู้อ่านควรแยกความแตกต่างระหว่างเครื่องของคลาสการใช้งานหลายช่องเวลาสำหรับเครื่องลูกข่ายและละเครื่อง ออกจากการกำหนดคลาสของเครื่องลูกข่าย ซึ่งได้กล่าวถึงไว้แล้วว่ามี 3 คลาส คือ คลาส A , คลาส B และ คลาส C ซึ่งแบ่งตามพฤติกรรมการรับส่งข่อมูลพร้อมๆกันระหว่างเครือข่าย GPRS และ GSM

2.13.1 เครื่องลูกข่าย

GPRS เชื่อมต่อกับสถานีฐานโดยผ่านทางคลื่นความถี่วิทยุ ข้อมูลหน่วยย่อยที่สุดจะอยู่ในรูปของ Burst ซึ่งเป็นไปตามข้อกำหนดมาตรฐานของเครือข่าย GSM โดยเฟรมข้อมูลที่ผ่านการเข้ารหัสช่องสื่อสารความยาว 456 บิต จะถูกแบ่งออกเป็น 4 ส่วนย่อยๆส่วนละ 57 บิตเพื่อนำมาผ่านกระบวนการอินเตอร์ลีฟ (Interleaving) การแยกส่วนและการรวมกลับคืนข้อมูลทั้ง 456 บิตเข้าด้วยกันถือเป็นการใช้กระบวนการปกติของเครือข่าย GSM ที่เรียกว่า GSM RF สำหรับเนื้อหาภายในเฟรมข้อมูลแต่ละชุดนั้นจะมีโปรโตคอล MAC และ RLC ฝังอยู่เพื่อทำหน้าที่ปรับแต่งแบบการเข้ารหัสตามขบวนการที่ได้กล่าวไปแล้ว เมื่อหักข้อมูล MAC/RLC ออกแล้ว ข้อมูลส่วนที่เหลือจะมีส่วนของโปรโตคอล LLC (Logical Link Control) และ SNDCP) ซึ่งไม่ถูกนำไปใช้งานโดยเครือข่ายสถานีฐาน แต่จะเป็นการสื่อสารโดยตรงระหว่างเครื่องลูกข่ายกับอุปกรณ์ SGSN โปรโตคอล LLC และ SNDCP ถือเป็นหัวใจสำคัญของการตรวจสอบตำแหน่งของเครื่องลูกข่ายโดยอุปกรณ์ SGSN และ GGSN รวมถึงการส่งสัญญาณควบคุมอื่นๆอีกหลายชนิดระหว่างเครือข่ายกับเครื่องลูกข่าย GPRS การออกแบบโปรโตคอล LLC และ SNDCP ขึ้นทำให้สามารถนำเทคโนโลยี GPRS ไปใช้ได้บนเครือข่ายโทรศัพท์เคลื่อนที่ชนิดอื่นๆ เช่น เครือข่าย CDMA หรือการพัฒนาเทคโนโลยี GPRS ไปสู่เครือข่ายโทรศัพท์เคลื่อนที่ยุคที่ 3 อันได้แก่ มาตรฐาน UMTS ข้อมูลส่วนที่เหลือจะมีส่วนของโปรโตคอล IP หรือ X.25 ซึ่งเครื่องลูกข่ายใช้ในการติดต่อสื่อสารเพื่อดำเนินการจัดการโปรไฟล์ PDP กับอุปกรณ์ GGSN การเลือกว่าจะใช้โปรโตคอล IP หรือ X.25 นั้นก็ขึ้นอยู่กับการออกแบบทางด้านวิศวกรรมเครือข่ายของผู้ให้บริการแต่ละราย ข้อมูลหลังจากโปรโตคอล IP/X.25 ก็จะเป็นเนื้อหาข้อมูลจริงของผู้ใช้บริการ หรือ แอบพลิเคชั่นที่ต้องการให้มีการรับส่งระหว่างเครื่องลูกข่าย GPRS กับแหล่งข้อมูลทั้งที่อยู่ภายในและภายนอกเครือข่าย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.20 โครงสร้างการจัดโปรโตคอลภายในเครือข่าย GPRS

2.13.2 เครือข่ายสถานีฐาน

ทำหน้าที่เป็นทั้งตัวทวนสัญญาณ (Relay Function) สำหรับข้อมูลที่เครื่องลูกข่ายต้องการสื่อสารกับอุปกรณ์ SGSN ขึ้นไปโดยตรง โดยส่งผ่านข้อมูลที่ถูกห่อหุ้มโดยโปรโตคอล LLC ต่อไปยังอุปกรณ์ SGSN และยังเป็นอุปกรณ์ปลายทางสำหรับควบคุมการทำงานเกี่ยวกับการจัดสรรช่องสื่อสารทางด้านความถี่วิทยุให้กับเครื่องลูกข่าย GPRS โดยผ่านการทำงานของโปรโตคอล GSM RF,MAC และ RLC ในขณะเดียวกันเครือข่ายสถานีฐาน (ซึ่งประกอบด้วยสถานีฐานและอุปกรณ์ BSC) ยังต้องมีการส่งสัญญาณควบคุมเพื่อติดต่อสื่อสารโดยตรงระหว่างเครือข่ายสถานีฐานเองกับอุปกรณ์ SGSN ซึ่งจำเป็นต้องมีการออกแบบโปรโตคอลอีกชุดหนึ่งเพื่อรองรับการติดต่อสื่อสารดังกล่าว รวมถึงเป็นรากฐานในการรับส่งข้อมูล LLC ที่ถูกส่งผ่านไปยังอุปกรณ์ SGSN ไปพร้อมๆกัน โปรโตคอลในจุดเชื่อมต่อ Gb ดังกล่าวประกอบไปด้วย ชั้นล่างสุดคือ L1bis ซึ่งสนับสนุนการรับส่งข้อมูลผ่านเครือข่าย สื่อสารสารพัดรูปแบบไม่ว่าจะเป็นวงจรสื่อสารแบบ TDM เช่น E1 ผ่านเครือข่าย ATM หรือผ่านเครือข่ายเฟรมรีเลย์ เป็นต้น โดยในระดับชั้นที่สองเป็นโปรโตคอลแบบ Frame Relay (FR) ทำหน้าที่แก้ไขความผิดพลาดของข้อมูลที่อาจเกิดขึ้นบนจุดเชื่อมต่อ Gb สำหรับเนื้อหาของสัญญาณควบคุมระหว่างเครือข่ายสถานีฐานกับอุปกรณ์ SGSN จะถูกบรรจุอยู่ในโปรโตคอล BSSGP (Base Station Subsystem GPRS Protocol) ซึ่งเป็นช่องทางในการบรรจุข้อมูล LLC ระหว่างเครื่องลูกข่าย GPRS กับอุปกรณ์ SGSN ด้วยในเวลาเดียวกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.14 ทฤษฎีเกี่ยวกับไมโครคอนโทรลเลอร์และการประยุกต์ใช้งาน

ไมโครคอนโทรลเลอร์ (Microcontrollers) คือชิปที่ประกอบไปด้วยไมโครคอนโทรลเลอร์ หน่วยความจำข้อมูล, หน่วยความจำโปรแกรมและพอร์ตอินพุท เอาท์พุท ซึ่งทำให้สามารถนำไมโครคอนโทรลเลอร์ไปประยุกต์ใช้งานต่างๆได้โดยไม่ต้องพึ่งอุปกรณ์ภายนอกมากนัก

ไมโครคอนโทรลเลอร์ที่จะกล่าวถึงที่นี้นั้นคือไมโครคอนโทรลเลอร์ตระกูล MCS-51 ของบริษัท อินเทล ซึ่งเบอร์ที่จัดว่าเป็นพื้นฐานในตระกูลนี้ได้แก่เบอร์ 8051, 8052 และ 8031 โดยในแต่ละเบอร์จะมีความแตกต่างกันในเรื่องของหน่วยความจำและโปรแกรมย่อยภายใน

2.14.1 คุณสมบัติของไมโครคอนโทรลเลอร์ตระกูล MCS-51

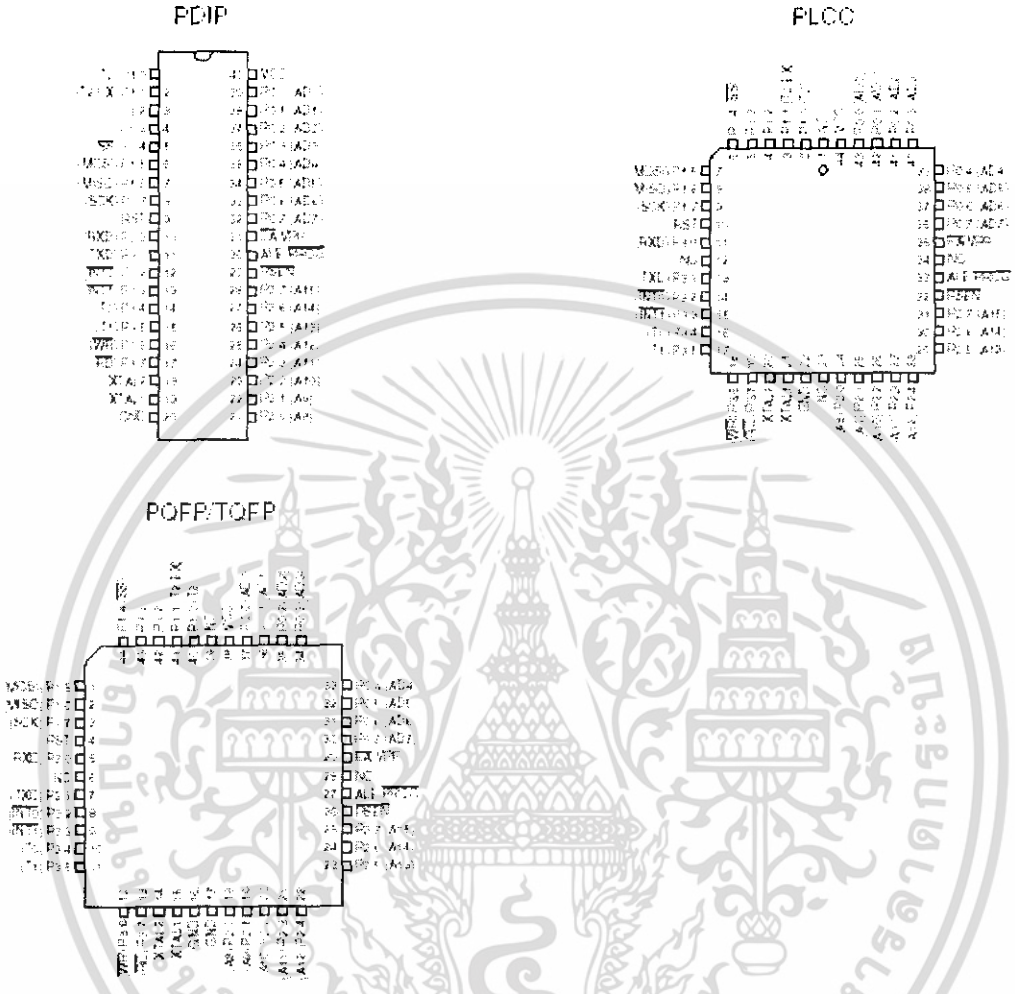
คุณสมบัติสำคัญ ได้แก่

- ใช้แหล่งจ่ายไฟ 1ชุด ขนาด 5โวลต์
- มีหน่วยความจำโปรแกรมภายในขนาด 4 กิโลไบต์ (สำหรับเบอร์ 8051) และขนาด 8 กิโลไบต์ (สำหรับเบอร์ 8052)
- มีหน่วยความจำข้อมูลภายในขนาด 128 ไบต์ (สำหรับเบอร์ 8051) และขนาด 256ไบต์ (สำหรับเบอร์ 8052)
- สามารถใช้หน่วยความจำโปรแกรมและข้อมูลภายนอกได้อย่างละ 64 กิโลไบต์ แยกจากกัน
- ใช้สัญญาณนาฬิกา ความถี่ 12 MHz ที่ผลิตสัญญาณจากคลิสตอล
- มีพอร์ตอินพุท พอร์ตเอาท์พุท ที่สามารถรับส่งข้อมูลได้สองทิศทาง จำนวน 4 พอร์ตๆละ 8 บิต คือมีพอร์ตใช้งานได้ 32 พอร์ต
- มีพอร์ตที่ใช้สำหรับรับส่งข้อมูลอนุกรม ที่สามารถกำหนดอัตราเร็วในการส่งผ่านข้อมูลได้
- จัดลำดับความสำคัญของสัญญาณอินเตอร์รัปได้ 2 ระดับ
- มีรีจิสเตอร์ที่สามารถใช้งานเป็นไทเมอร์และเคาน์เตอร์ขนาด 16 บิต จำนวน 2 ตัว
- หน่วยความจำข้อมูลภายในบางส่วนสามารถเข้าถึงข้อมูลได้ทั้งระดับไบต์และระดับบิต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.14.2 การจัดเรียงขาและอธิบายรายระเอียดของขาต่างๆของไมโครคอนโทรลเลอร์ตระกูล MCS – 51

ไมโครคอนโทรลเลอร์ตระกูล MCS – 51 เบอร์ 8051, 8052 จะมี 40 ขา ดังรูป



รูปที่ 2.21 แสดงตำแหน่งขาต่างๆของชิปไมโครคอนโทรลเลอร์ตระกูล MCS – 51

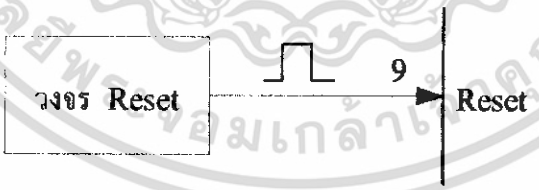
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หน้าที่ของแต่ละขา

P0.0 - P0.7	(ขา 39 - 32) เป็น Port ใช้งานทั่วไป และพอร์ท Data และ พอร์ท Address ไบท์ต่ำ
P1.0 - P1.7	(ขา 1 - 8) เป็น Port ใช้งานทั่วไป
P2.0 - P2.7	(ขา 21- 28) เป็น Port ใช้งานทั่วไป และพอร์ท Address ไบท์สูง
P3.0 -3.7	(ขา 10 -17) เป็นพอร์ตใช้งานทั่วไป นอกจากนั้นยังสามารถทำหน้าที่อื่น ๆ ได้ดังนี้
P3.0, RXD	สามารถเป็นขารับสัญญาณ Serial Port
P3.1, TXD	สามารถเป็นขาส่งสัญญาณ Serial Port
P3.2, INT0	สามารถเป็นขารับสัญญาณ Interrupt หมายเลข 0
P3.3, INT1	สามารถเป็นขารับสัญญาณ Interrupt หมายเลข 1
P3.4, T0	สามารถเป็นขาที่รับสัญญาณพัลส์ หมายเลข 0 เพื่อเข้าวงจร Counter
P3.5, T1	สามารถเป็นขาที่รับสัญญาณพัลส์ หมายเลข 1 เพื่อเข้าวงจร Counter
P3.6, WR	สามารถเป็นขาสัญญาณเขียนข้อมูลไปยังอุปกรณ์ภายนอก
P3.7, RD	สามารถเป็นขาสัญญาณอ่านข้อมูลจากอุปกรณ์ภายนอก

Reset (ขา 9)

ขานี้มีไว้สำหรับรีเซตการทำงานของ MCS-51 ให้เริ่มต้นทำงานใหม่ ขานี้ทำงานที่ลอจิก "1" สังเกตว่าไม่มีเครื่องหมาย บนชื่อของขา ถ้ามีแสดงว่าทำงานที่ลอจิก "0" สิ่งที่เราต้องออกแบบวงจรก็คือต้องสร้างสัญญาณพัลส์บวกที่ขา RST ในช่วงเวลาที่เราจ่ายไฟให้กับ MCS-51 ตอนเปิดเครื่อง เพราะฉะนั้นเมื่อเราเปิดเครื่อง ไมโครคอนโทรลเลอร์จะรีเซตตัวเองโดยอัตโนมัติ วงจรนี้เราสามารถสร้างโดยใช้ R และ C เพื่อสร้างสัญญาณพัลส์บวก ได้ดังวงจรข้างล่างนี้



รูปที่ 2.22 การใช้งานขา Reset

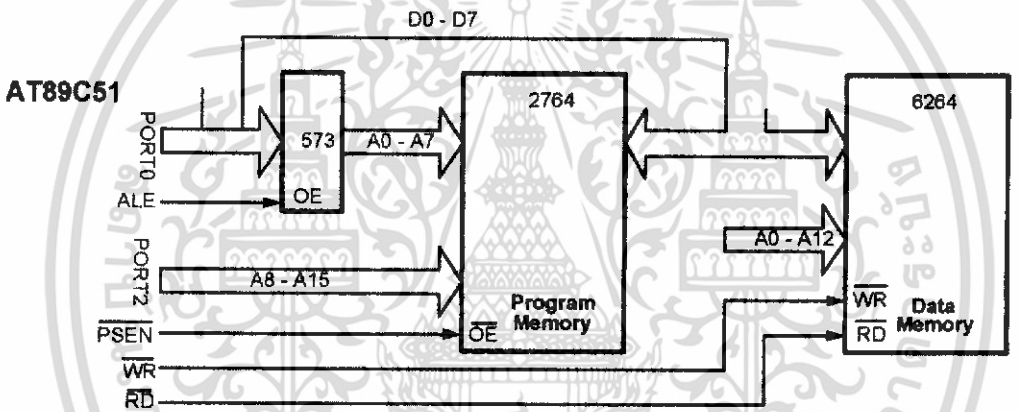
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

XTAL2,1 (ขา 18,19)

ไว้สำหรับต่อ X – TAL เพื่อสร้างสัญญาณนาฬิกาให้กับ MCS – 51 เพื่อกำหนดจังหวะการทำงานของไมโครคอนโทรลเลอร์ วงจรสร้างสัญญาณนาฬิกานั้นจึงต้องการ X-Tal เพื่อสร้างสัญญาณ ซึ่งค่าความถี่ X – Tal ที่เราใช้คือ 11.0592 MHz การที่เราใช้ค่านี้ เพราะใน MCS - 51 นั้นมีวงจร Serial Port ซึ่งทำหน้าที่ในการส่งข้อมูลอนุกรม ซึ่งจะต้องมีอัตราการส่งข้อมูลหรือความถี่ในการส่งข้อมูลเป็นไปตามมาตรฐานของ Serial Port ซึ่ง Serial Port ความถี่ในการส่งข้อมูลนั้น จะได้จากการหารความถี่จาก X-Tal ซึ่งค่าความถี่ X-Tal ค่านี้เมื่อหารความถี่แล้ว เราจะได้ความถี่ที่ตรงตามความถี่มาตรฐานพอดี

PSEN (Program Status Enable) (ขา 29)

ขานี้จะทำงานที่ลอจิก "0" (Active low) ขานี้ Active เมื่อ MCS – 51 ต้องการติดต่อกับ Program Memory ซึ่ง Program Memory จะเป็นหน่วยความจำที่ทำหน้าที่ เก็บคำสั่ง หน่วยความจำส่วนนี้ส่วนใหญ่จะเป็น EPROM ที่ต่ออยู่ภายนอกหรือ FLASH ROM ที่อยู่ภายในเบอร์ ดังนั้นขานี้จะต่อเข้ากับขา OE ของ EPROM ในกรณีที่เราไม่ได้ต่อ EPROM ภายนอก จะทำการปล่อยลอยๆไว้ อย่างนั้น



รูปที่ 2.23 แสดงการใช้งานขา Reset

ALE (Address Latch Enable) (ขา30)

เนื่องจาก MCS-51 มีความสามารถในอ้างหน่วยความจำและอุปกรณ์ภายนอก จึงใช้ พอร์ท 2 เป็น Address ไบท์สูง (A8 - A15) และให้พอร์ท 0 เป็น Data Bus และ Address Bus ไบท์ต่ำ เพื่อเป็นการประหยัดพอร์ท ดังนั้นเมื่อพอร์ท 0 มีถึงสองหน้าที่ จึงต้องมีสัญญาณมา บอกให้รู้ว่าตอนนี้ข้อมูลที่อยู่บนพอร์ท 0 เป็น Address หรือ Data จึงใช้ขา ALE เป็นตัวบอก ว่าที่พอร์ท 0 เป็น Address ขา ALE จะเป็น "1" ด้วยเหตุนี้ที่พอร์ท 0 จึงมีไอซีที่ทำหน้าที่ เก็บข้อมูล (Latch) เช่นเบอร์ 74HC373, 74HC573 ในกรณีที่ต่อหน่วยความจำภายนอก โดย ขา ALE จะต่อเข้ากับขา OE เพื่อส่งสัญญาณให้กับไอซี เอาข้อมูลที่พอร์ท 0 ซึ่งเป็น Address ไบท์ต่ำ (A0 - A7) ออกไปที่เอาท์พุทของ ไอซีตัวนี้ ดังนั้นเอาท์พุทของไอซี Latch จึงเป็น Address (A0-A7)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โครงสร้างภายในของไมโครคอนโทรเลอร์ตระกูล MCS -51 มี 2 ส่วนดังนี้ คือ

- หน่วยความจำโปรแกรม (Program Memory)
- หน่วยความจำข้อมูล (Data Memory)

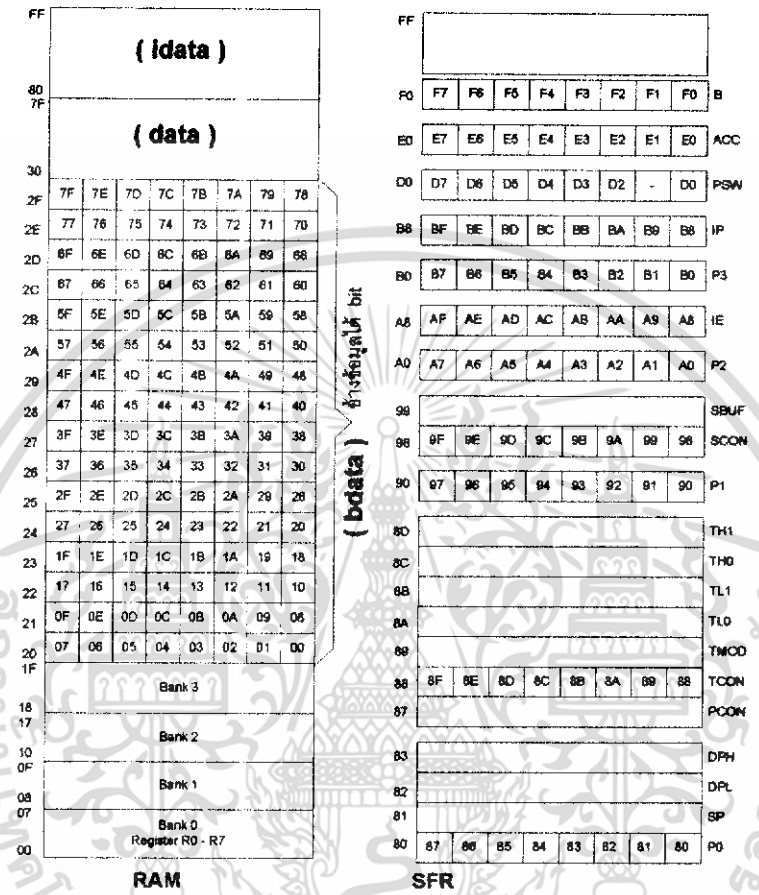
1. หน่วยความจำโปรแกรมจะแบ่งเป็น 2 ส่วน คือ หน่วยความจำโปรแกรมภายใน (Internal Program Memory) มีขนาด 4 กิโลไบต์ (สำหรับ 8051) และหน่วยความจำภายนอก (External Program Memory) มีได้มากสุดถึง 64 กิโลไบต์ หน่วยความจำโปรแกรมภายนอก (Program Memory) สามารถอ้างข้อมูลได้ 64 Kbytes เป็นส่วนที่ทำหน้าที่เก็บคำสั่งที่เราเขียนส่วนใหญ่ถูกเก็บไว้ที่ EPROM ภายนอก แต่ปัจจุบันนี้ ได้มีการพัฒนาจนสามารถนำเอา Program Memory อีกแบบหนึ่งซึ่งเรียกว่า Flash Rom เข้าไปในไมโครคอนโทรลเลอร์ได้แล้ว Flash ROM นั้นมีข้อดีกว่า EPROM เพราะสามารถลบได้ด้วยไฟฟ้าและเขียนได้เร็วกว่า ด้วยเหตุนี้จึงต้องมีขา EA เป็นตัวเลือกว่าจะใช้ Program Memory ภายนอกหรือภายใน ซึ่งในที่นี้เราใช้เบอร์ AT89C51 ซึ่งมี Program Memory ภายในที่เป็น Flash ROM เราจึงให้ขา EA ต่อ VCC ถ้าเราใช้ Program Memory ภายนอกขา EA จะต่อลงกราวด์

2. หน่วยความจำข้อมูลจะแบ่งเป็น 2 ส่วน คือ หน่วยความจำข้อมูลภายใน (Internal RAM) มีขนาด 256 ไบต์ (สำหรับ 8051) และหน่วยความจำภายนอก (External Data Memory) มีได้มากสุดถึง 64 กิโลไบต์ หน่วยความจำชนิดนี้ทำหน้าที่ เก็บข้อมูลต่างๆ ที่เราต้องการซึ่งก็คือ RAM ที่ต่ออยู่ ภายนอก สามารถเข้าถึงข้อมูลโดยใช้ขา RD, WR ในการเขียนและอ่าน Data จาก RAM MCS-51 สามารถอ้างหน่วยความจำข้อมูลภายนอก ได้ 64 Kbytes เช่นกัน ส่วนใหญ่แล้วหน่วยความจำนี้จะถูกจัด Address ให้ถัดจาก Program Memory (EPROM) RAM ที่ใช้ส่วนใหญ่จะเป็น Static RAM เบอร์จะขึ้นด้วย หมายเลข 61, 62, 64, 68 แล้วตามด้วยขนาดของหน่วยความจำเช่น 6264 หมายเลข 64 คือขนาดของหน่วยความจำ โดยเอาค่า 64 / 8 = 8 Kbytes เราก็จะได้ขนาดของ RAM คำสั่งที่ใช้ในการอ้างข้อมูลของ Data Memory คือ x data

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.15 หน่วยความจำภายใน Internal RAM

เป็นหน่วยความจำภายใน ซึ่งเป็น RAM ในหน่วยความจำนั้น ก็จะประกอบด้วย Register ซึ่งจะมี Address ประจำตัวทำหน้าที่ใช้งานแตกต่างกันไปตามการใช้งาน รูปข้างล่างนี้แสดงรีจิสเตอร์ของหน่วยความจำภายใน



รูปที่ 2.24 แสดงการแบ่งประเภทหน่วยความจำภายใน

ภายในหน่วยความจำข้อมูลทั่วไป มีวิธีการเข้าถึงแบบโดยตรงและทางอ้อม แบ่งเป็นกลุ่มดังนี้

- รีจิสเตอร์ที่ใช้งานทั่วไป (Register Bank) :ประกอบด้วย 4 กลุ่มๆ ละ 8 ไบต์ จำนวนทั้งหมด 32 ไบต์ ใช้เก็บ ข้อมูลทั่วไป ระดับของการอ้างอิงเป็นระดับไบต์
- รีจิสเตอร์ที่เข้าถึงข้อมูลระดับบิต จำนวน 16ไบต์ ทั้งหมด 128 บิต
- รีจิสเตอร์ที่ใช้งานเก็บข้อมูลทั่วไป จำนวน 86 ไบต์ ระดับของการอ้างอิงข้อมูลเป็นระดับไบต์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โครงสร้างพอร์ตข้อมูลของไมโครคอนโทรลเลอร์ตระกูล MCS- 51 แบ่งได้ดังนี้

- พอร์ตข้อมูลแบบขนาน มีขนาดพอร์ตละ 8 บิต จำนวน 4 พอร์ต (P0, P1, P2, P3) แต่ละพอร์ตสามารถใช้เป็นพอร์ตอินพุทหรือเอาท์พุทได้อย่างอิสระ
- พอร์ตข้อมูลแบบอนุกรม ใช้รับส่งข้อมูลแบบอนุกรม สามารถกำหนดอัตราเร็วในการส่งข้อมูล(Baud Rate) ได้ (ตั้งแต่ 110 bps ถึง 375 kbps)

ไทม์เมอร์(Timer) และเคาน์เตอร์ (counter)

- ไทเมอร์นั้นแต่ละตัวจะมีให้ใช้ไม่เท่ากันโดยอย่างเบอร์ 8052 นั้น จะมีไทม์เมอร์ อยู่ 2ตัว คือ ไทม์เมอร์เบอร์0 และ 1 แต่ตัวอื่นๆที่ดีกว่าอาจมีมากกว่า 2 ตัว
- เคาน์เตอร์ ใช้สำหรับการนับเปลี่ยนแปลงสถานะของสัญญาณภายนอกซึ่งขาใช้งานจะเป็นขาเดียวกับไทม์เมอร์โดยการเลือกใช้ต้องใช้คำสั่งในการเลือกใช้ เช่นเดียวกับไทม์เมอร์

การเข้าถึงข้อมูลต่างในหน่วยความจำและกลุ่มคำสั่งของไมโครคอนโทรลเลอร์ตระกูล MCS -51

วิธีการเข้าถึงข้อมูลแบ่งเป็นดังนี้

1. การเข้าถึงข้อมูลโดยตรง (Direct Addressing) เป็นการระบุค่าตำแหน่งของหน่วยความจำในคำสั่งใช้กับหน่วยความนำข้อมูลทั่วไป (ภายใน) รีจิสเตอร์ที่มีหน้าที่เฉพาะ (SFR) และหน่วยความจำข้อมูลภายนอก
2. การเข้าถึงข้อมูลภายนอก (Indirect Addressing) เป็นการระบุค่าตำแหน่งของหน่วยความจำ โดยระบุค่าผ่านค่าที่เก็บในรีจิสเตอร์ ใช้กับหน่วยความจำข้อมูลทั่วไป (ภายใน) และหน่วยความจำข้อมูลภายนอก รีจิสเตอร์ที่ใช้เป็นตัวชี้ตำแหน่งได้แก่ รีจิสเตอร์แบงค์ (R0 –R7ในแต่ละกลุ่ม), สแตกพอยต์เตอร์และดาต้าพอยต์เตอร์ (DPTR)
3. การเข้าถึงขอมูลในรีจิสเตอร์ที่ใช้งานทั่วไป (Register Addressing) เป็นการเข้าถึงข้อมูลในรีจิสเตอร์ทั่วไป (Register Bank, Ro-R7 ในแต่ละกลุ่ม)
4. การเข้าถึงข้อมูลในรีจิสเตอร์เฉพาะ (Register – Specific Addressing) เป็นการเข้าถึงข้อมูลในรีจิสเตอร์เฉพาะเท่านั้นเช่น Accumulator ,DPTR ,SP เป็นต้น
5. การเข้าถึงข้อมูลที่กำหนดเองโดยตรง (Immediate Addressing) เป็นการเข้าถึงข้อมูลแบบกำหนดข้าข้อมูลโดยตรง
6. การเข้าถึงข้อมูลโดยใช้ตัวอ้างอิง (Indexed Addressing) ใช้ระบบตำแหน่งของหน่วยความจำโปรแกรมเท่านั้น โดยจะนำค่าในรีจิสเตอร์เฉพาะได้แก่ DPTR, PC มารวมกับค่า รีจิสเตอร์ A เพื่อระบุตำแหน่งของหน่วยความจำโปรแกรมต่อไป

คำสั่งในการใช้ไมโครคอนโทรลเลอร์ตระกูล MCS -51แบ่งเป็นกลุ่มๆได้ดังนี้

1. กลุ่มคำสั่งทางคณิตศาสตร์ (Arithmetic Instructions) ได้แก่คำสั่งที่ใช้ในการประมวลผลการคำนวนทางคณิตศาตร์ เช่น การ บวก ลบ คูณ หารการเพิ่มค่า การลดค่าเป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. กลุ่มคำสั่งทางตรรกศาสตร์ (Logical Instructions) ได้แก่ กลุ่มคำสั่งที่ใช้ในการประมวลผลทางตรรกศาสตร์ เช่น AND, OR, Exclusive-OR, complement, การเลื่อนสลับบิต เป็นต้น
3. กลุ่มคำสั่งในการเคลื่อนย้ายข้อมูล (Data Transfer) เป็นกลุ่มคำสั่งที่ใช้ในการควบคุมการเคลื่อนย้ายข้อมูลในหน่วยความจำหรือรีจิสเตอร์แบ่งเป็น
    - กลุ่มคำสั่งในการเคลื่อนย้ายข้อมูลในหน่วยความจำภายใน
    - กลุ่มคำสั่งในการเคลื่อนย้ายข้อมูลในหน่วยความจำภายนอก
    - กลุ่มคำสั่งในการเคลื่อนย้ายข้อมูลในหน่วยความจำโปรแกรมทั้งภายในและภายนอก
4. กลุ่มคำสั่งในการควบคุมลำดับการทำงานของโปรแกรม (Program Control Instructions) ใช้ในการควบคุมลำดับการทำงานของโปรแกรมแบ่งเป็นแบบมีเงื่อนไขกับไม่มีเงื่อนไข
5. กลุ่มคำสั่งในการประมวลผลแบบบูลีน (Boolean Instructions) เป็นคำสั่งในการประมวลผลแบบพิเศษ โดยจะประมวลผลได้ในระดับบิต ประกอบด้วย คำสั่งในการเซตบิต, เคลียร์บิต, การกระทำคำสั่งทางตรรกศาสตร์ระหว่างหน่วยความจำกับบิต Carry Flag, การตรวจสอบสถานะบิตแล้วข้ามไปทำงานส่วนอื่นของโปรแกรมขึ้นอยู่กับสถานะของบิตและการเคลื่อนย้ายข้อมูลระหว่าง Carry Flag กับหน่วยความจำเป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.16 เว็บเซอร์วิส (Web service)

คือระบบซอฟต์แวร์ที่ออกแบบมา เพื่อสนับสนุนการแลกเปลี่ยนข้อมูลกัน ระหว่างเครื่องคอมพิวเตอร์ผ่านระบบเครือข่าย โดยที่ภาษาที่ใช้ในการติดต่อสื่อสารระหว่างเครื่องคอมพิวเตอร์ คือภาษาเอ็กซ์เอ็มแอล XML เว็บเซอร์วิสมีอินเตอร์เฟส ที่ใช้อธิบายรูปแบบข้อมูลที่เครื่องคอมพิวเตอร์ประมวลผลได้ เช่น WSDL ระบบคอมพิวเตอร์ใช้งานสื่อสารโต้ตอบกับเว็บเซอร์วิสตามรูปแบบที่ได้กำหนดไว้แล้ว โดยการส่งสารตามอินเตอร์เฟสของเว็บเซอร์วิสนั้น โดยที่สารดังกล่าวอาจแนบไว้ในซอง SOAP หรือส่งตามอินเตอร์เฟสในแนวทางของ REST ข้อมูลเหล่านี้ปกติแล้วถูกส่งโดยอาศัย HTTP และใช้ XML ร่วมกับมาตรฐานเกี่ยวกับเว็บอื่นๆ โปรแกรมประยุกต์ที่เขียนโดยภาษาต่างๆ และทำงานบนแพลตฟอร์มต่างๆกันสามารถใช้เว็บเซอร์วิสเพื่อแลกเปลี่ยนข้อมูลผ่านทางเครือข่ายคอมพิวเตอร์ เช่น อินเทอร์เน็ต ในลักษณะเดียวกับการสื่อสารระหว่างโปรเซส (Inter-process communication) บนเครื่องเดียวกัน ความสามารถในการแลกเปลี่ยนข้อมูลระหว่างระบบที่ต่างกันนี้ (เช่น การแลกเปลี่ยนข้อมูลระหว่างโปรแกรมที่เขียนโดยภาษาจาวา และโปรแกรมที่เขียนโดยภาษาไพทอน หรือการแลกเปลี่ยนข้อมูลระหว่างโปรแกรมประยุกต์ที่ทำงานบนไมโครซอฟท์วินโดวส์และโปรแกรมประยุกต์ที่ทำงานบนลีนุกซ์) เกิดขึ้นได้เนื่องจากการใช้มาตรฐานเปิด โดย OASIS และ W3C เป็นคณะกรรมการหลักในการรับผิดชอบมาตรฐานและสถาปัตยกรรมของเว็บเซอร์วิส

คำว่า Service ไม่ได้หมายถึงอะไรที่เด่นชัดอย่าง Promool.com Pantip.com แต่หมายถึงส่วนประกอบที่คนอื่นๆนำไปใช้ในการทำบริการที่กว้างกว่านี้ด้วย ตัวอย่างเช่น Microsoft Passport ที่ให้บริการตรวจสอบความเป็นตัวตนจริง (Authentication) ผ่านเว็บ ทำให้การบริการข่าวของ Bangkok Post ไม่ต้องตรวจสอบการเข้าสู่ระบบเอง แต่ยกให้ Passport เป็นตัวจัดการแทน หรืออย่าง Dynamic services whitepaper ของ Oracle ก็มีส่วนที่ให้บริการ แปลงค่าเงิน แปลภาษา การส่งของ เป็นต้น ส่วนความหมายอย่างเป็นทางการของ Web Service ก็คงเป็นของ IBM ที่กล่าวว่าเว็บเซอร์วิส คือ Web Application ยุคใหม่ ที่ประกอบด้วยส่วนย่อยๆมีความสมบูรณ์ในตัวเอง สามารถติดตั้ง ค้นหา เริ่มทำงานได้ผ่านเว็บ Web Service สามารถทำอะไรก็ได้ตั้งแต่งานง่ายๆ เช่นดึงข้อมูล จนถึงกระบวนการทางธุรกิจที่ซับซ้อน เมื่อ Web Service ตัวใดตัวหนึ่งเริ่มทำงาน Web Service ตัวอื่นก็สามารถรับรู้และเริ่มทำงานได้อีก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.16.1 พื้นฐานของ Web Service

พื้นฐานของ Web Service ก็คือ XML กับ HTTP ซึ่งจะพบว่า HTTP ก็เป็นที่รู้จักกันดี และไปได้ทั่วทุกแห่งที่มี internet ส่วน XML คือภาษาสากลที่คุณสามารถปรับแต่งได้ตามใจชอบ เพื่อให้เกิดกิจกรรมระว่าง client และบริการ หรือระหว่างส่วนประกอบต่างๆ เบื้องหลัง Web server ก็คือ ข้อความ XML จะถูกแปลงให้การขอบริการจาก Middle ware และผลที่ได้ก็จะแปลงกลับมาในรูป XML

แต่การเข้าถึงและการสั่งงานนั้นยังเป็นเพียงโครงสร้างพื้นฐาน แต่ในความเป็นจริงยังมีอะไรมากกว่านั้น เช่น การค้นหา การทำธุรกรรม ความปลอดภัย การพิสูจน์ตัวตน และอื่นๆ อันเป็นบริการที่ทำให้เป็นบริการพื้นฐานจริงๆ ระบบเพิ่มเติมที่ต้องมีและต้องรักษาความสะดวกและใช้งานง่ายไว้ด้วย พื้นฐานของ Web Service เต็มรูปแบบคือ XML + HTTP + SOAP + WSDL + UDDI หรือในระดับสูงกว่านั้น แต่ไม่ได้ถือเป็นสิ่งจำเป็นเสมอไปคือต้องเพิ่มเทคโนโลยี XAML, XLANG, XKMS, XFS เป็นต้น

SOAP

Web service เป็นลักษณะในรูปแบบของการออกแบบโมเดลสื่อสาร ในลักษณะของการกระจายการติดต่อสื่อสารที่เป็นตัวกลาง โดยโปรโตคอลที่ใช้ในการสื่อสารคือ SOAP (Simple Object Access Group) เป็นโปรโตคอลในการติดต่อแลกเปลี่ยนข้อมูลระหว่างกันเป็นโปรโตคอลการสื่อสารในระดับ Application Layer หรือในระดับ แอปพลิเคชันโดยอาศัยผ่านอินเทอร์เน็ตโปรโตคอล อย่างเช่น HTTP, SMTP โปรโตคอลพัฒนารากฐานมาจาก XML

WSDL

ย่อมาจาก Web Services Description Language เป็นภาษาที่อธิบายเว็บเซอร์วิส ซึ่งทำให้ผู้เรียกใช้เว็บเซอร์วิสหรือโปรแกรมที่ต้องการเรียกใช้เว็บเซอร์วิสทราบว่ามีโอเปอเรชั่นอะไรให้บริการ และในแต่ละโอเปอเรชั่นจะต้องส่งข้อมูลอะไรไปและจะได้รับข้อมูลแบบใดกลับมา ตลอดจนทราบอินเทอร์เน็ตโปรโตคอลที่จะต้องใช้ในการติดต่อเว็บเซอร์วิสและที่อยู่ของเว็บเซอร์วิส

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.17 ความเป็นมาของ XML (Extensive Markup Language)

โปรโตคอลอินเตอร์เนต (Internet Protocol-IP), Hyper Text Markup Language และ Hyper Text Transport Protocol (HTTP) ได้เป็นการปฏิวัติและสร้างมิติใหม่ในการกระจายข้อมูล การนำเสนอ ตลอดจนการค้นหา โดยให้ผู้ใช้สามารถหาข้อมูลที่ต้องการได้ง่ายด้วยเบราวเซอร์ และมี search engine หรือเครื่องมือในการช่วยค้นหา นอกจากนั้นยังมีการประยุกต์ไปใช้กับเครือข่ายในสำนักงานหรืออินเตอร์เน็ต และใช้สำหรับการบริการข้อมูลสำหรับลูกค้าและคู่ค้าให้สามารถตอบสนองทางด้านข้อมูลได้อย่างมีประสิทธิภาพมากขึ้น

สำหรับ Extensive Markup Language จะให้ประโยชน์อย่างเต็มที่เมื่อทำงานร่วมกับ HTML ด้วยเหตุที่ว่า XML ได้มีความพร้อมในรายละเอียด และการนำข้อมูลตลอดจนโครงสร้างข้อมูลมาแสดงได้ในรูปแบบ Text ผ่านทาง HTTP ที่เปิดให้ข้อมูลขึ้นใหม่และมีความสามารถในการจัดข้อมูลได้อีกด้วย ในการเขียนเว็บเพจเมื่อใช้ HTML ผู้พัฒนาสามารถกำหนดได้ว่าส่วนไหนจะเป็นตัวหนา ตัวเอียง หรือตัวอักษรเป็นแบบไหน ส่วน XML นั้นจะเป็นการเตรียมส่วนของข้อมูลที่จะนำไปใส่ในช่องที่กำหนดตามการเขียนของ HTML ยกตัวอย่างเช่น ข้อมูลด้านราคาหรือราคาที่ตั้งสำหรับการจัดรายการส่งเสริมการขาย อัตราภาษี ค่าขนส่ง เป็นต้น

XML ถือได้ว่าเป็นส่วนหนึ่งของ Standard Generalized Language Markup Language (SGML) ที่เป็นข้อกำหนดในการสร้างหรือจัดทำเอกสารในรูปแบบอิเล็คทรอนิกส์ที่กำหนดโดย W3C หรือ World Wide Web Consortium สามารถดูรายละเอียดเพิ่มเติมได้จาก http://www.w3.org/TR/REC-xml ที่มีโครงสร้างและรูปแบบที่เปิดให้แอพพลิเคชันต่างๆ สามารถเรียกไปใช้งานได้ เช่น บนเว็บไซต์ต่างๆ เป็นต้น และทางไมโครซอฟท์ได้มีการทำงานร่วมกับ W3C เพื่อพัฒนามาตรฐานข้อมูลบนเว็บที่ให้ HTML สามารถแสดงข้อมูลที่ XML ได้เตรียมไว้ และทางไมโครซอฟท์เองได้มีการเปิดตัว เบราว์เซอร์ตั้งแต่ IE 4.0 เป็นต้นไป ที่สามารถเรียกดูและประมวลผลข้อมูลได้ และเป็นข้อกำหนดให้ เบราว์เซอร์เวอร์ชันใหม่ของค่ายไมโครซอฟท์สนับสนุน XML

สำหรับประโยชน์ในการใช้งานนั้น เราจะสามารถนำมาใช้สำหรับการเข้าถึงระบบข้อมูลขนาดใหญ่ใช้กับระบบเครือข่ายในองค์กร หรืออินเตอร์เน็ตเพื่อดูข้อมูลหรือเรียกใช้ข้อมูลที่ให้การแสดงผลทางหน้าจอที่รวดเร็วและง่ายในการจัดการ

Extensive Markup Language เป็นฟอร์แมตที่อธิบายถึงรายละเอียดของโครงสร้างและแบบของข้อมูลเป็นภาษาหรือชุดคำสั่งเกี่ยวกับข้อมูลบนเว็บ ที่ให้การพัฒนาและมีศักยภาพในส่วนของโครงสร้างข้อมูลจากหลากหลายแอพพลิเคชันมานำเสนอบนเครื่องเดสก์ทอป ด้วย XML จะทำให้การจัดการข้อมูลหรือเรียกใช้ข้อมูลจากแอพพลิเคชันต่างๆ จะเข้าสู่มาตรฐานเดียวกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

XML จะเป็นส่วนหนึ่งของ HTML ซึ่ง XML จะให้รายละเอียดเกี่ยวกับข้อมูล เช่น ชื่อเมือง อุณหภูมิ ความกดอากาศ ส่วน HTML เป็นการกำหนด tag ต่างๆ ที่จะทำให้ข้อมูลแสดงออกมาในรูปแบบไหน ซึ่งข้อมูลจะสามารถแสดงออกมาได้หลายรูปแบบ ไม่ว่าจะเป็นตารางหรือ text ธรรมดา ขึ้นอยู่กับการกำหนดของ HTML และในปัจจุบันนี้ ด้วย XML จะมีการให้รายละเอียดของเนื้อหาเอกสารที่เรียกว่า Document Type Definition (DTD) ที่ให้รายละเอียดเกี่ยวกับตัวเอกสารว่าจะแสดงหรือซ่อนส่วนไหนของเอกสารบ้าง ซึ่ง DTD จะเป็นส่วนที่เพิ่มเติมสำหรับ XML ถ้าหากมีการส่งข้อมูลในรูปแบบ DTD ก็จะรู้กันว่าเป็น XML มีความหมายหลายๆ คำที่ อธิบายลักษณะของ XML

Richard Baldwin นิยามความหมายของ XML ไว้ดังนี้

XML ทำให้ผู้ใช้สามารถสร้างและดูแล structured documents (เอกสารที่มีโครงสร้าง) ที่บรรจุ plain text (ตัวอักษร) โดยทำให้สามารถ rendered หรือปรับเปลี่ยนการแสดง ผลในรูปแบบที่หลากหลาย จุดประสงค์หลักของ XML คือการแยกส่วน ข้อมูลเพื่อประโยชน์ในการแสดงผลตามข้อความด้านบน สามารถอธิบายความหมายของคำศัพท์ได้ดังนี้

Structured document

พิจารณาจากตัวอย่างโครงสร้างของ หนังสือ ต่อไปนี้คู

- หนังสือหนึ่งเล่มประกอบจากเนื้อหาแต่ละบท (chapter)
- ในแต่ละบทประกอบด้วยหัวข้อย่อย (section)
- ในแต่ละหัวข้อย่อย อาจจะถูกอธิบายหรือมีตารางข้อมูล (table) บรรจุอยู่
- ตารางข้อมูลถูกสร้างขึ้นมาจากแถว (row) และคอลัมน์ (column)

Plain text

ตัวอักษร (Character) ที่เห็นในจดหมายหรือบนกระดาษทั่วไป เป็นลักษณะตัวอักษรประเภทที่พิมพ์ได้ (alphabet and punctuation) เช่นที่เราใช้พิมพ์ในเอกสารในเวิร์ดหรือ โปรแกรมพิมพ์รายงานทั่วไป ซึ่งรวมถึงตัวเลขด้วย ถือว่าเป็น plain text ทั้งนี้และทั้งนั้น ผู้อ่านอาจจะสงสัยว่า แล้วตัวอักษรอะไรพิมพ์ไม่ได้หรือลองมองดูที่คีย์บอร์ดของท่าน แล้วดูที่แป้น Shift, Ctrl, Alt, Insert, Delete เหล่านี้เป็นต้น

ASCII คือที่มาของรหัสตัวอักษร รูปแบบที่ใช้แทนตัวอักษรที่นิยมใช้กันมากที่สุด เรียกว่า (ASCII) โดยจะแทนตัวอักษรด้วยรหัสหรือค่าตัวเลข เช่น ตัวอักษร "A" ถูกแทนด้วยค่า 65 ในขณะที่ตัวอักษร "B" แทนด้วยค่า 66 และ "C" คือ 67 ตามลำดับ รหัส ASCII นี้เอง ซึ่งถูกนำมาใช้งานในการแสดงข้อมูล รูปแบบเอกสาร ที่สามารถพิมพ์ให้ผู้ใช้ได้เห็นในขณะที่การบันทึกบนระบบคอมพิวเตอร์ จะใช้ในการเก็บเป็นตัวเลขแทน ASCII เป็นคำย่อที่มาจาก American Standards Committee on Information Interchange โดยเป็นองค์กรที่จะตั้งขึ้นมาดูแล รหัสคำสั่งต่างๆเหล่านี้ แต่ในความเป็นจริงนอกจาก ASCII จะเก็บบันทึกตัวอักษรแล้ว ยังมีรหัสที่แทนตัวอักษร ต่อไปนี้เป็นความหมายที่ทำให้เข้าใจความเกี่ยวข้อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระหว่างรหัส ASCII กับ Plain text มาตรฐานที่เกิดจาก American Standards Committee on Information Interchange คือแนวทางที่ถูกใช้ในการกำหนดลักษณะของ plain text กล่าวคือตัวอักษรที่สามารถอ่านได้โดยไม่ต้องใช้ความสามารถ ของโปรแกรมใดๆมาช่วยในการอ่านตัวอักษรเหล่านั้น ข้อพิเศษของ ASCII อีกอย่างหนึ่งคือ เป็นข้อมูลที่สามารถ นำไปแสดงได้ในทุกๆแพลตฟอร์ม แต่ก็มีข้อเสียที่ เป็นรูปแบบที่น่าเบื่อ (ไม่สวยงาม) ไม่สามารถแสดงในรูปแบบตัวหนา ตัวเอียง หรือขีดเส้นใต้ และไม่มีรูปแบบที่ดูเป็นกราฟฟิก แต่อย่างไรก็ตาม ถือว่าเป็นแบบในการแสดงตัวอักษรที่ง่ายที่สุด และเข้าใจได้ในทุกๆ ระบบ

ข้อมูลดิบของเอกสาร XML (Raw-XML)

เอกสาร XML ที่ถือเป็นข้อมูลดิบ คือข้อความที่เกิดจากการรวมกันของตัวอักษร เพื่อทำให้เกิดเป็นเอกสาร ก่อนที่จะนำเอกสารที่เกิดจากข้อมูลดิบ มาใช้แสดงผลลัพธ์ตามวัตถุประสงค์ของการใช้งานอีกที ซึ่งเป็นรูปแบบที่น่าสนใจมากกว่า plain text

ความหมายของการปรับการแสดงผล (Rendering)

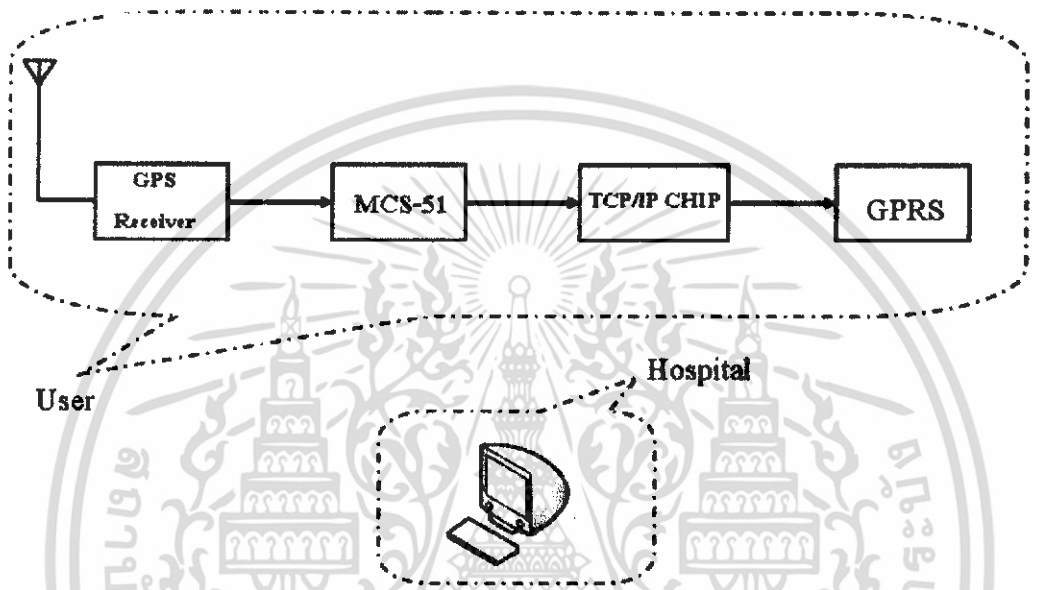
เป็นศัพท์สำหรับคอมพิวเตอร์สมัยใหม่ Rendering คือรูปแบบของการนำข้อมูลดิบ หรือ plain text มาปรับเปลี่ยนใหม่ ให้ดูน่าสนใจยิ่งขึ้น สำหรับการมองของมนุษย์ rendering เป็นการวาดภาพ (Drawing) ตัวอย่างของการวาดภาพในระบบคอมพิวเตอร์ เป็นการนำเอาสมการ ทางคณิตศาสตร์ มาใช้ร่วมกับตัวเลขต่างๆ เช่น สมการวาดวงกลม กับค่าตัวเลขหนึ่งให้วงกลมขนาดหนึ่ง กับอีกตัวเลข ให้วงกลมในอีกหนึ่งขนาด ซึ่งตัวเลขและสมการเหล่านี้ เป็นสิ่งที่มนุษย์ หรือผู้ใช้งานไม่อาจได้เห็นเลย แต่จะเห็นเป็นรูปร่างที่เกิดขึ้นแทน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3
การออกแบบและการสร้าง

3.1 โครงสร้างการทำงานของระบบที่ออกแบบ

ส่วนโครงสร้างของระบบนี้จะถูกติดอยู่กับผู้ป่วยหรือผู้ต้องการความช่วยเหลือและที่โรงพยาบาลของผู้ป่วย โดยจะมีส่วนของการประมวลผลทั้งหมดและจัดการข้อมูลต่างๆ โดยจะมีโครงสร้างโดยรวมแสดงดังรูปที่ 3.1

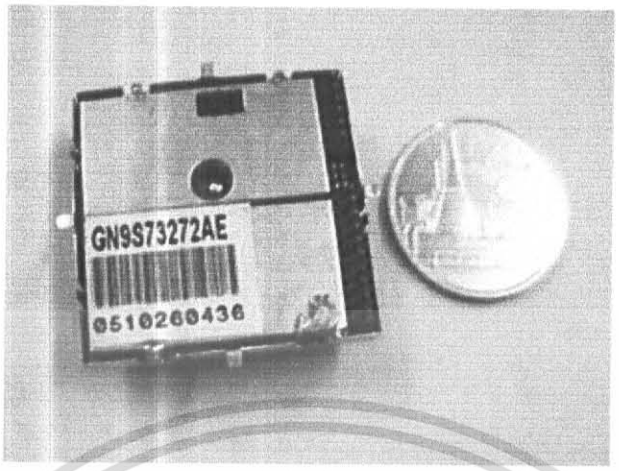


รูปที่ 3.1 โครงสร้างการทำงานของระบบ

หลักการทำงานของระบบแสดงตำแหน่งของผู้ร้องขอและหาเส้นทางในการไปพบผู้ร้องขอเป็นการนำเอาเทคโนโลยีของ ระบบ GPS มาประยุกต์ใช้ในการประมวลผลข้อมูลต่าง ๆ เพื่อที่จะสามารถแสดงตำแหน่งปัจจุบันของผู้ร้องขอ ณ ขณะนั้น ว่าอยู่ ณ ตำแหน่งใด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2 เครื่องรับระบบพิกัดตำแหน่ง (GPS Receiver)



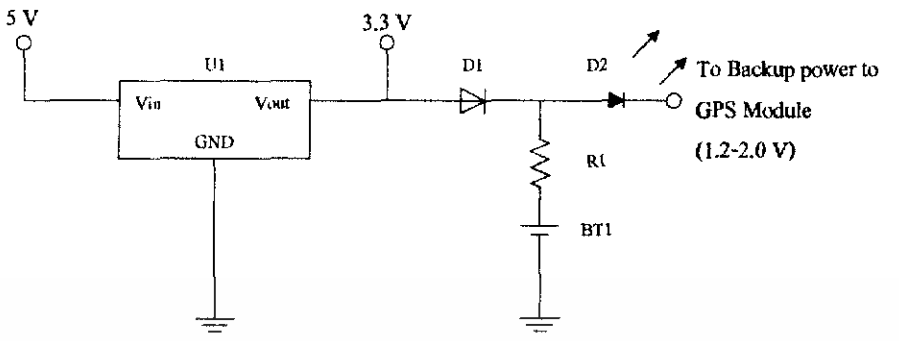
รูปที่ 3.2 เครื่องรับ GPS ที่เลือกใช้รุ่น VT002P

ขา	ชื่อ	ประเภท	หน้าที่
1	GPIO[4]	I/O	บอกสถานะ
2	NC (Mode)	NU	ไม่มีหน้าที่
3	NMEA TX	I/O	ด้านส่ง ให้เอาต์พุตเป็นข้อมูลและสัญญาณนำร่อง (0)
4	NMEA RX	I/O	ด้านรับ จะรับสัญญาณจากซอร์ฟแวร์ (0)
5	NC(WAKEUP)	NU	ไม่มีหน้าที่
6	Reset	I	ต่อปุ่มรีเซต (Active Low)
7	VBAT	I	แบตเตอร์รี่สำรองใช้ไฟ DC 2.5-3.3V
8	GND	PWR	กราวด์ (Ground)
9	VDD	PWR	ไฟ DC +5V
10	GPIO[0]	I/O	กราวด์ (Ground)

ตารางที่ 3.1 ขาต่อใช้งาน VT002P

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3 การต่อใช้งานโมดูล VT002P

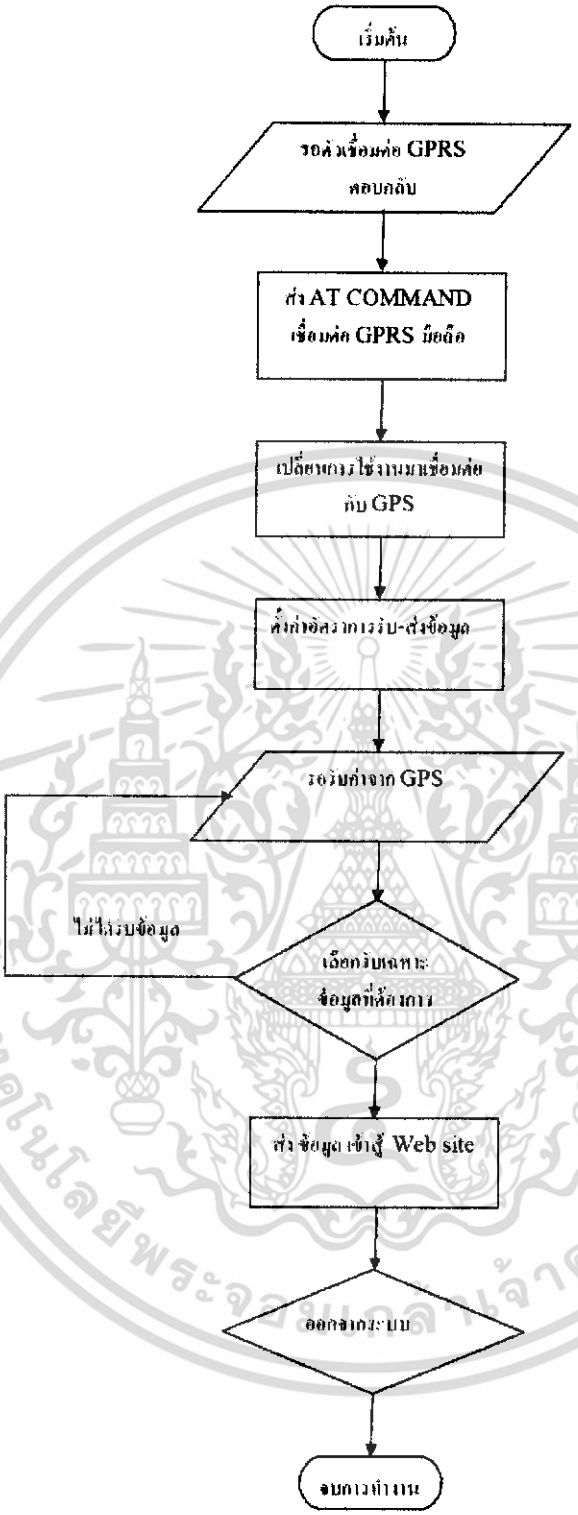


รูปที่ 3.3 วงจรจ่ายไฟโมดูล GPS

โดยวงจรประกอบด้วย

U1	:	RT9167-33CB (RICHTEK)
D1	:	BAT54 (FAIRCHIAL)
R1	:	100 Ohm
D2	:	LED
BT1	:	Battery ML4145 (Panasonic)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

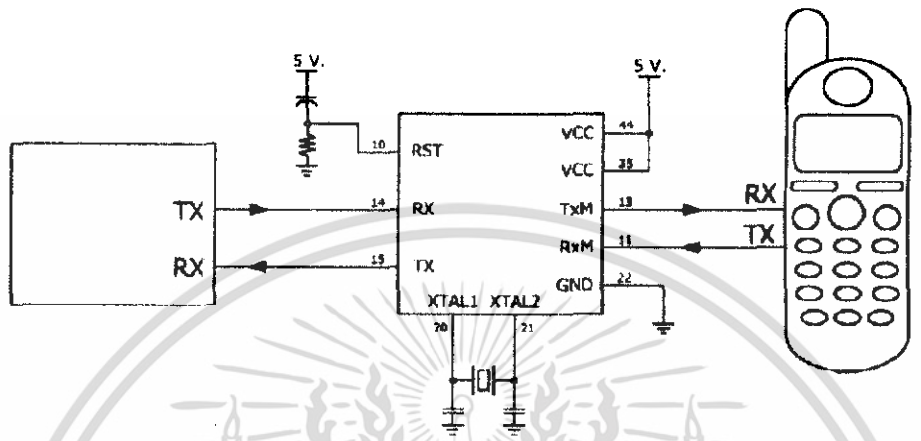


รูปที่ 3.4 ผังการทำงานในส่วนของโปรแกรมทั้งหมด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

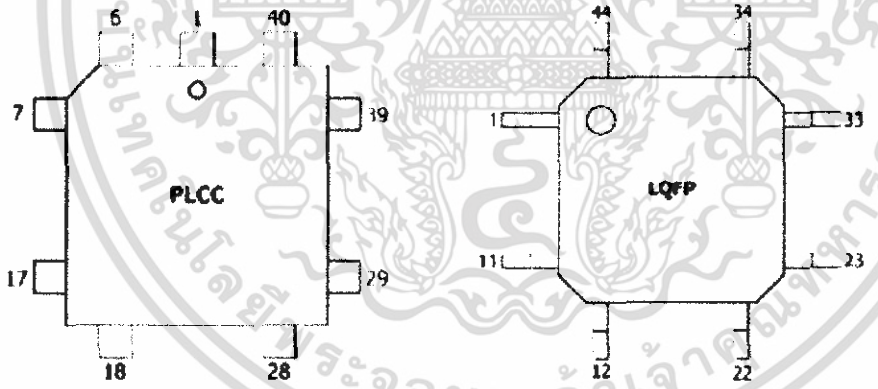
3.4 วงจร TCP/IP Stack

ในโครงงานนี้จะใช้ชิพ TCP/IP Stack เบอร์ TNS010i ซึ่งมีส่วนของโปรแกรมที่ทำหน้าที่กำหนดลำดับข้อมูลและแปลความข้อมูล รูปแพ็กเกจและโปรโตคอลแบบ TCP และ ให้ IP เป็นลักษณะการกำหนดจุดตัวเชื่อมเน็ตเวิร์ค ซึ่งมีข้อดีคือสามารถสนับสนุนโปโตคอล พื้นฐาน เช่น TCP/IP, PPP, DNS, HTTP, SMTP, POP3 และ TCP/IP socket



รูปที่ 3.5 โครงสร้างการเชื่อมต่อของวงจร TNS010i

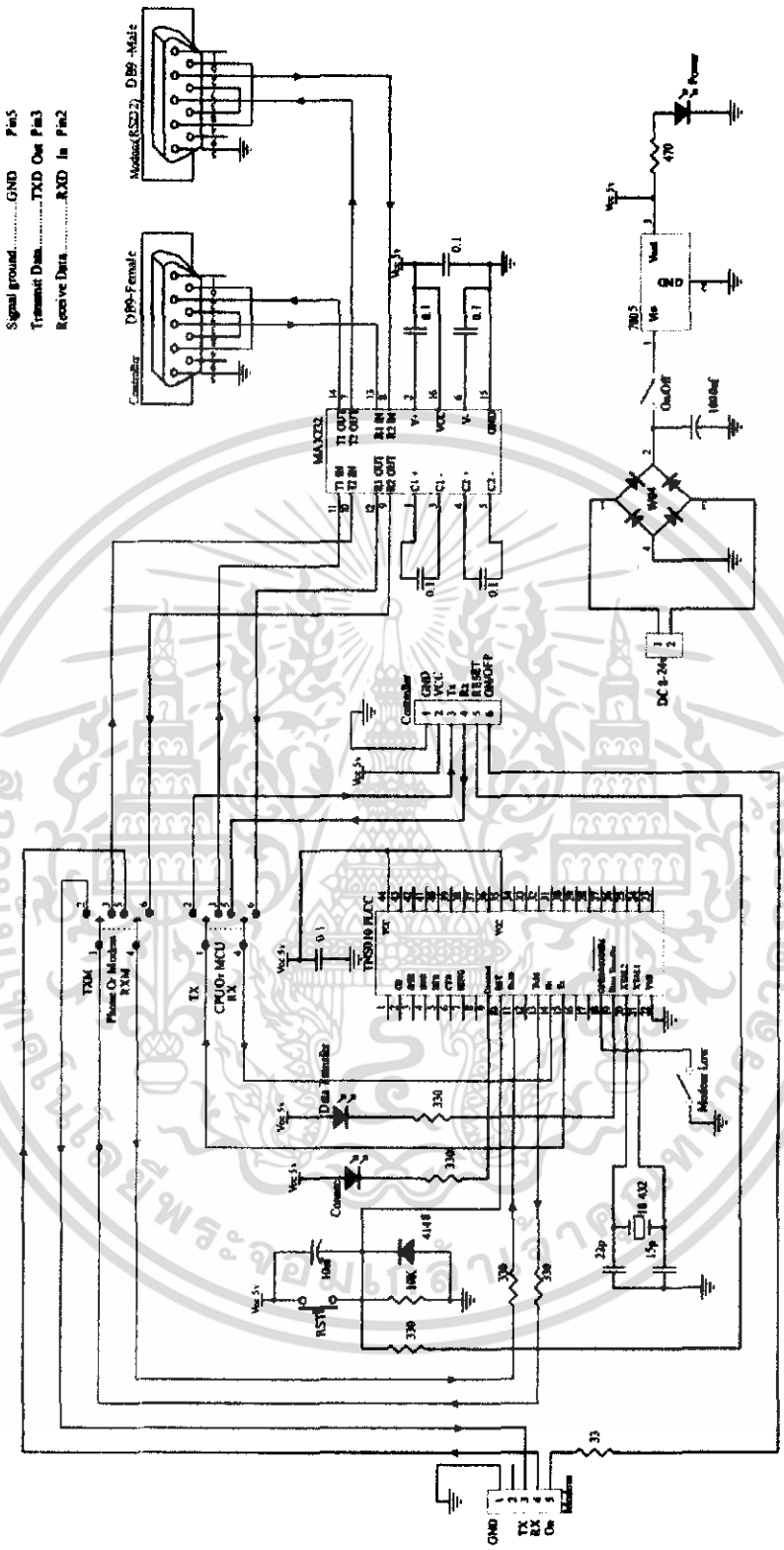
โครงสร้างของชิพมีลักษณะดังรูป



รูปที่ 3.6 แสดงโครงสร้างของชิป TNS 010i

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

มีรูปวงจรดังต่อไปนี้



รูปที่ 3.7 การต่อใช้งานของวงจร TCP/IP Stack

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4.1 AT COMMAND

AT COMMAND คือชุดคำสั่งมาตรฐาน ที่สามารถใช้ติดต่อสื่อสารกับอุปกรณ์สื่อสารต่างๆ เช่น โมเด็ม หรือ อุปกรณ์ DTE (Data Terminal Equipment) เพื่อโต้ตอบตั้งค่าหรือสั่งงานอุปกรณ์เหล่านั้น ให้ทำงานตามที่ต้องการ และสำหรับการติดต่อกับโทรศัพท์มือถือ จะใช้ชุดคำสั่งที่เรียกว่า GSM AT COMMAND

ตัวอย่างคำสั่งที่เป็น BASIC AT COMMAND

AT	ตรวจสอบความพร้อมของอุปกรณ์ ถ้าสามารถติดต่อกันได้ อุปกรณ์จะตอบกลับมาว่า OK
ATD phone number;	โทรไปยังเลขหมายปลายทาง (phone number)
ATH	วางสาย
ATA	รับสาย

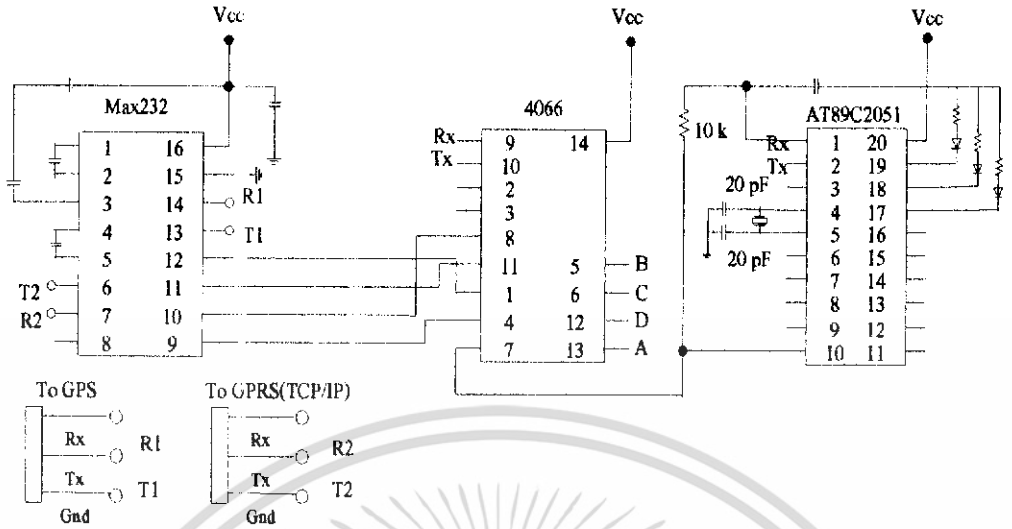
3.4.2 AT COMMAND ที่เกี่ยวข้องกับการเชื่อมต่อ ของชิป TNS010i

ชุดคำสั่ง AT COMMAND ที่ใช้กับโทรศัพท์มือถือได้มีอยู่มากมาย ทั้งการอ่านรุ่นโทรศัพท์ ตรวจสอบระดับแบตเตอร์รี่ ตรวจสอบระดับสัญญาณ แต่ในที่นี้จะกล่าวเพียงคำสั่งที่เกี่ยวกับการรับ-ส่งข้อมูลโดยผ่าน ชิป TNS010i เท่านั้น

AT+I	ตรวจสอบความพร้อมของชิป
AT+IOPEN	เปิดการเชื่อมต่อ PPP
AT+ISETD=XXX	ใส่เบอร์ติดต่อกับ ISP
AT+ISETDNS=<IP>	ตั้งค่า DNS server's IP address (ในกรณีที่ ISP ไม่กำหนดไว้เป็นอัตโนมัติ)
AT+ISETUSER=XXX	ตั้งค่า PPP user name
AT+ISETPASS=XXX	ตั้งค่า PPP password

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

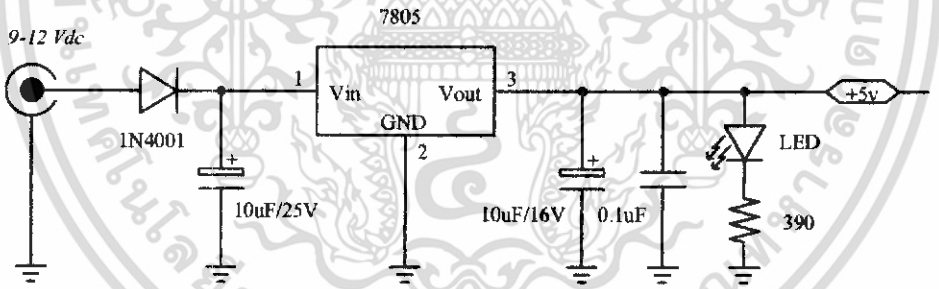
3.5 วงจรไมโครคอนโทรลเลอร์



รูปที่ 3.9 การต่อใช้งานไมโครคอนโทรลเลอร์

3.6 วงจรภาคจ่ายไฟ

เป็นวงจรที่จำกัดแรงดันให้เหลือเพียง 5 V เพื่อที่จะเป็นแหล่งจ่ายไฟให้กับวงจรไมโครคอนโทรลเลอร์ของเรา ซึ่งตัวที่ทำหน้าที่นี้คือ ไอซี 7805

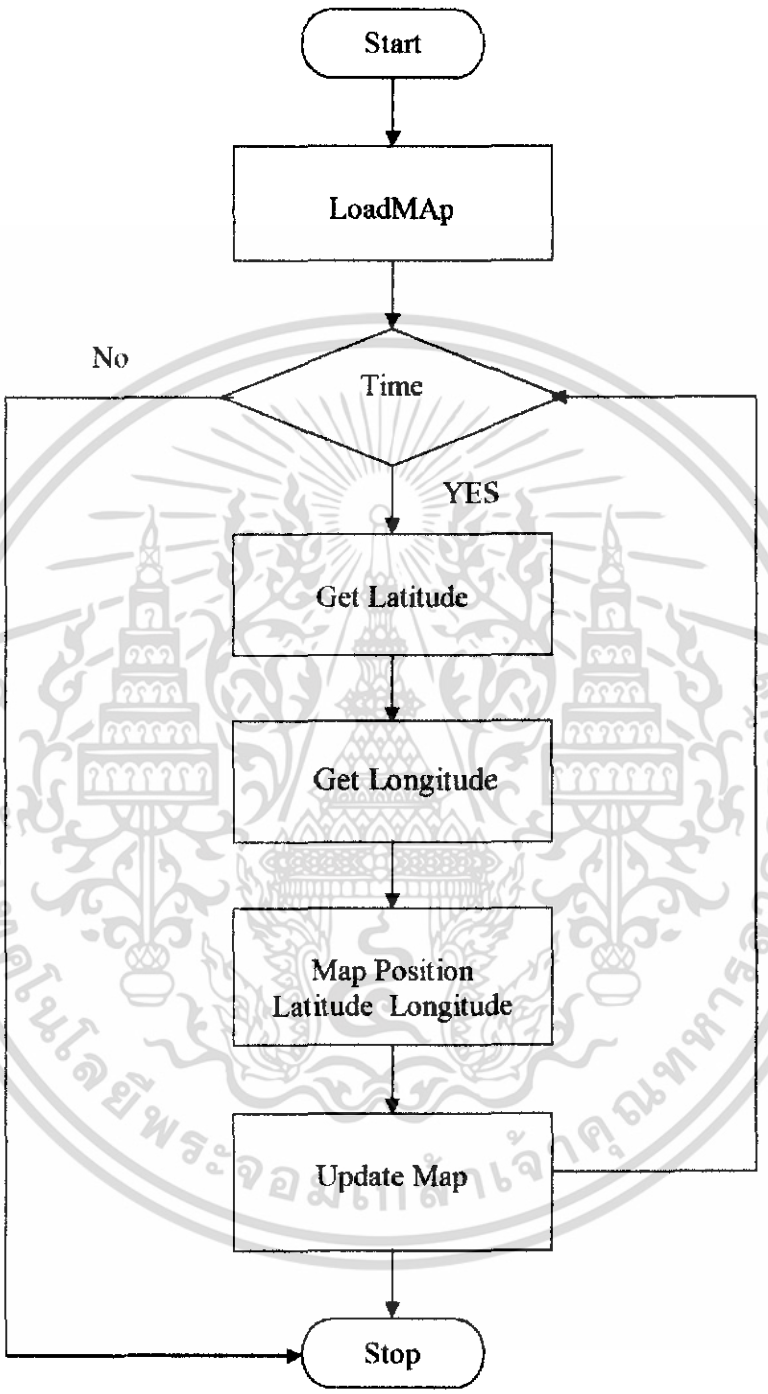


รูปที่ 3.10 วงจรภาคจ่ายไฟ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.7 การสร้างและการออกแบบโปรแกรม

ขั้นตอนการเขียนโปรแกรม Position tracking สามารถอธิบายโครงสร้างด้วย Flow Chart ได้ดังนี้

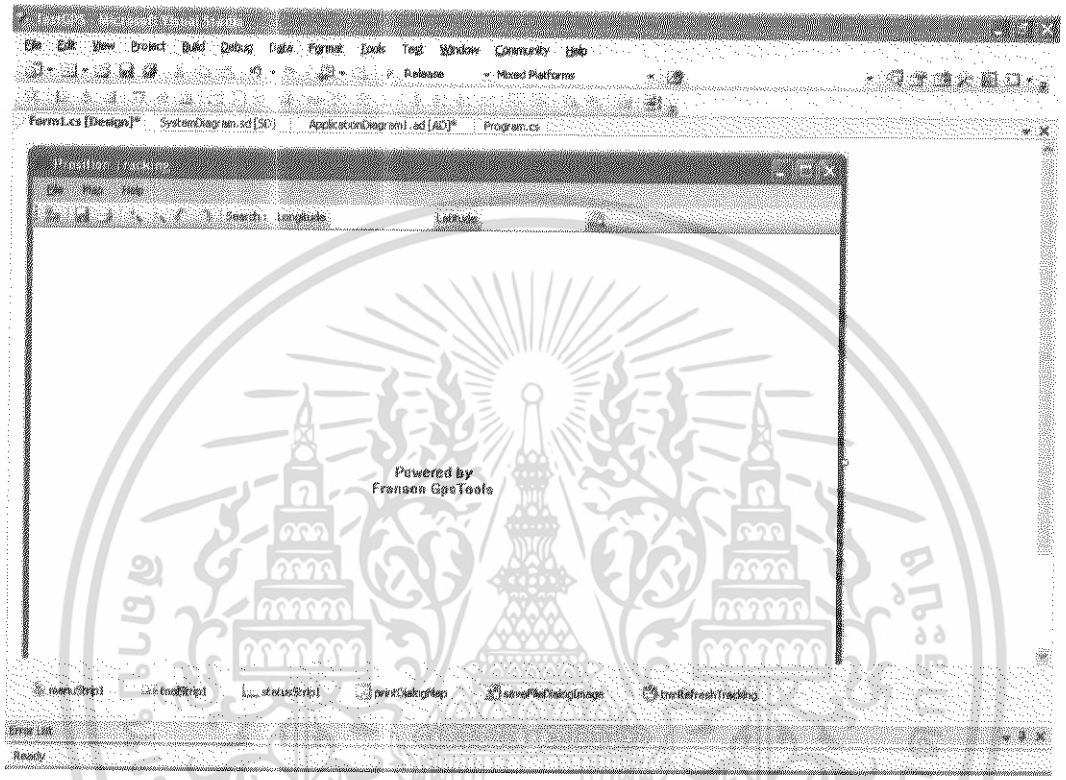


รูปที่ 3.11 แสดงการทำงานของโปรแกรม Position tracking

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยโปรแกรม Position tracking ถูกเขียนขึ้นมาโดยใช้โปรแกรม Microsoft visual studio 2005 ซึ่งจะมีโปรแกรมหลายโปรแกรมบรรจุอยู่ โดยเราจะใช้ในส่วนของ Microsoft visual C# เป็นเครื่องมือในการเขียนและยังเป็นภาษาหนึ่งที่มีรูปแบบของภาษาเป็น Object Oriented Program

เริ่มออกแบบ Form โปรแกรมโดยการเพิ่ม Tool ต่างๆใน Form แล้วเขียนคำสั่งเพื่อควบคุม Tool นั้นๆตามหน้าที่ของมัน ซึ่งแสดงหน้าต่าง Form ดังรูปที่ 3.12



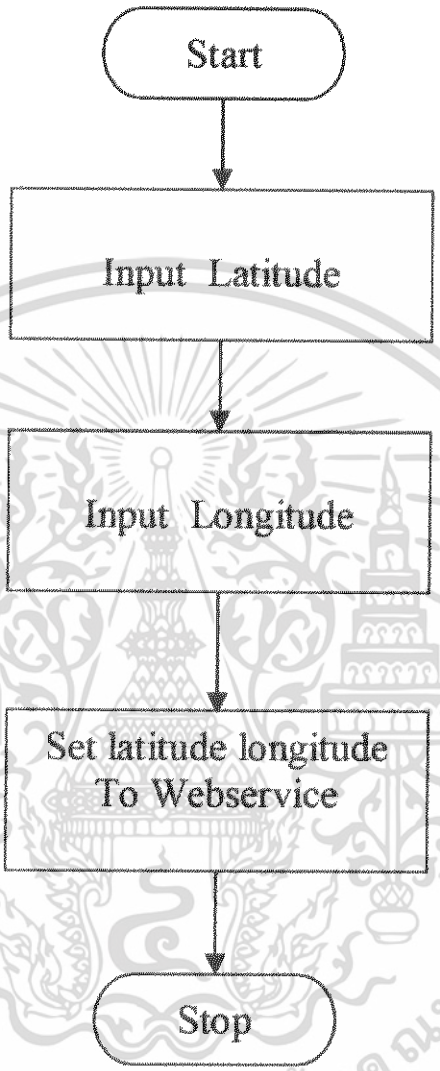
รูปที่ 3.12 แสดงให้เห็นหน้าต่าง Form โปรแกรมที่มี Tool ต่างๆ ประกอบอยู่

ในการคำนวณการสร้าง Grid นั้นได้สร้างเก็บไว้เป็น file.dll เพื่อฝังคำสั่งนี้ไว้ในโปรแกรม ซึ่ง Form จะไปดึงคำสั่งนี้มาใช้งานร่วมกับแผนที่ โดยแผนที่ที่นำมาใช้นั้น ได้ทำการจับภาพมาจากแผนที่ของโปรแกรม Smartmap GPS เมื่อทำการจับภาพเสร็จแล้ว จำเป็นที่จะต้อง Mark ตำแหน่งอ้างอิงไว้ประมาณ 10 จุดเพื่อให้พิกัดของแผนที่ของโปรแกรม Position Tracking แม่นยำที่สุดเท่าเทียมกับพิกัดในแผนที่ของโปรแกรม Smartmap GPS

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.8 การออกแบบโปรแกรม Test latitude longitude

แสดงได้ตาม Flow Chart ดังนี้

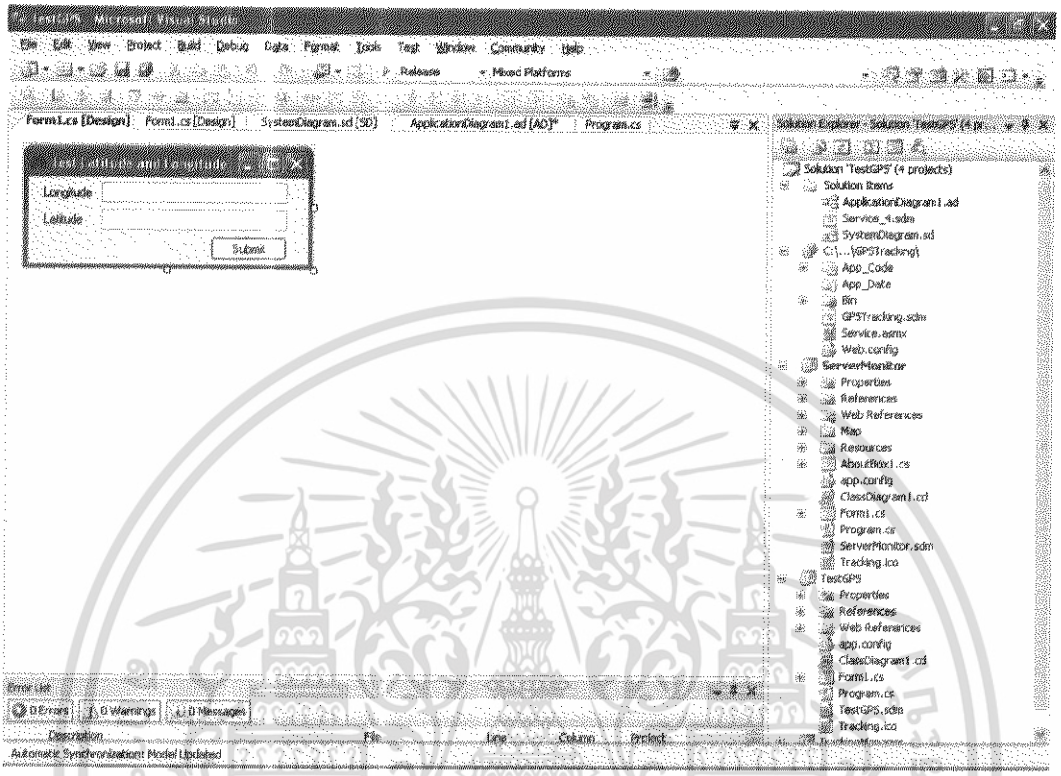


รูปที่ 3.13 แสดงการทำงานของโปรแกรม Test latitude longitude

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.9 การออกแบบ Form โปรแกรม Test latitude longitude

โปรแกรม Test latitude longitude ใช้ Microsoft Visual C# เป็นเครื่องมือในการเขียนแสดงได้ดังรูป 3.14



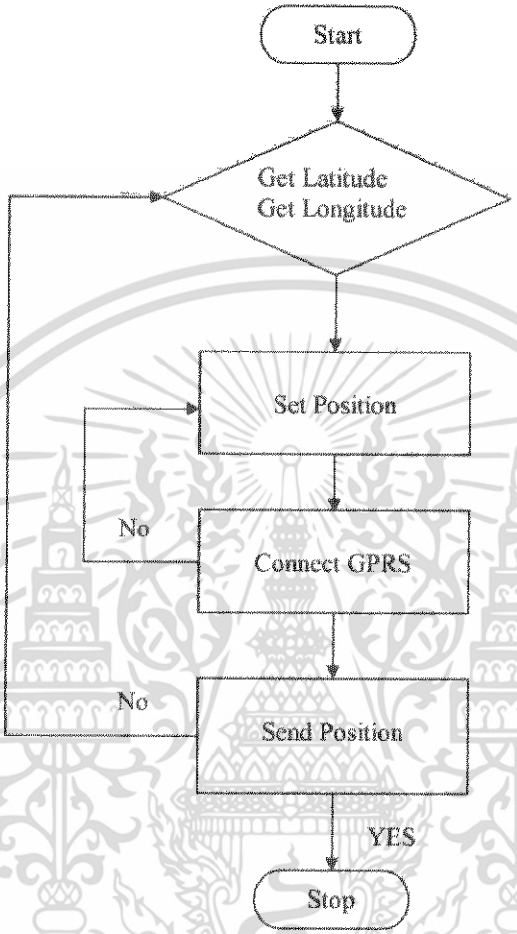
รูปที่ 3.14 แสดงการออกแบบ Form ของโปรแกรม Test latitude longitude

ซึ่งโปรแกรม Test latitude longitude สามารถใช้งานได้เลยโดยไม่จำเป็นต้องติดตั้ง โดยสามารถทำการทดสอบได้จากเครื่องเดียวกันกับตัวเครื่อง Server หรือคนละเครื่องกันก็ได้ (Client) โดยการตั้งค่า IP ใน XML Configuration File ของเครื่องที่ทดสอบ (Client) ให้ตรงกับ IP ของเครื่อง Server

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.10 การออกแบบโปรแกรม GPSTrackingMobile เพื่อใช้ทดสอบการส่งพิกัดละติจูด ลองติจูด จากโทรศัพท์มือถือผ่านเครือข่าย GPRS

การออกแบบโปรแกรม GPSTrackingMobile ถูกออกแบบและเขียนด้วยภาษา JAVA ตาม FlowChart ดังรูปที่ 3.15



รูปที่ 3.15 แสดงขั้นตอนการทำงานของ โปรแกรม GPSTrackingMobile

การติดตั้งโปรแกรม GPSTrackingMobile ทำได้โดยการใช้โปรแกรม Nokia PC Suite ช่วยในการติดตั้งตัว Application นี้ มีขั้นตอนดังนี้

1. เชื่อมต่อคอมพิวเตอร์กับโทรศัพท์มือถือ แล้วเปิดโปรแกรม Nokia PC Suite โปรแกรมจะทำการค้นหาโทรศัพท์มือถือโดยอัตโนมัติ

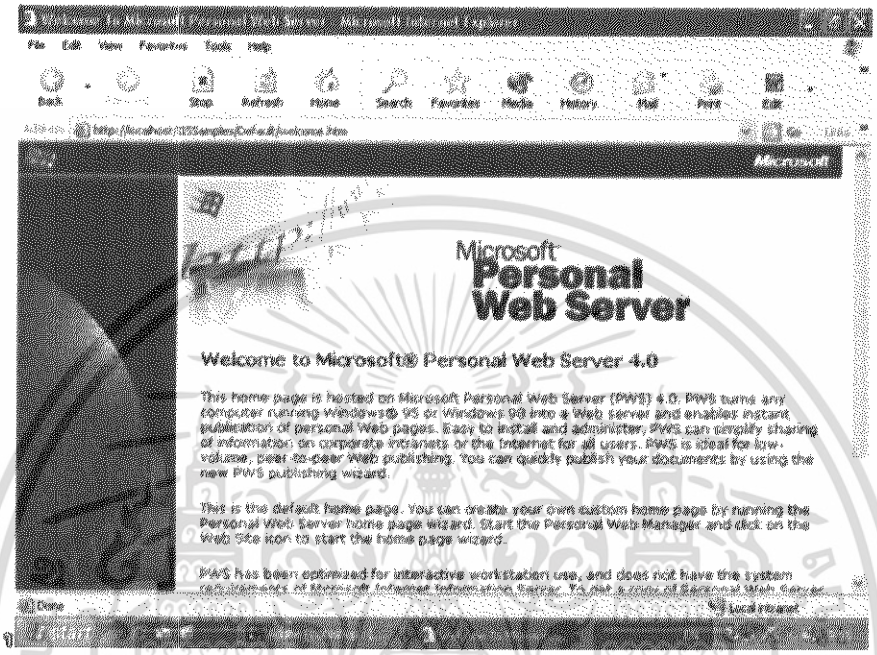
2. เลือก Icon การติดตั้ง Application จากนั้นเลือกไฟล์ที่จะติดตั้งและเลือกแหล่งที่เก็บโปรแกรม (เก็บไว้ในโฟลเดอร์ส่วนตัว)

3. เมื่อทำการติดตั้งสำเร็จ ก็สามารถเข้าไปใช้โปรแกรม GPSTrackingMobile ในไฟล์โทรศัพท์มือถือที่เก็บเอาไว้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.11 การติดตั้งโปรแกรม Position Tracking

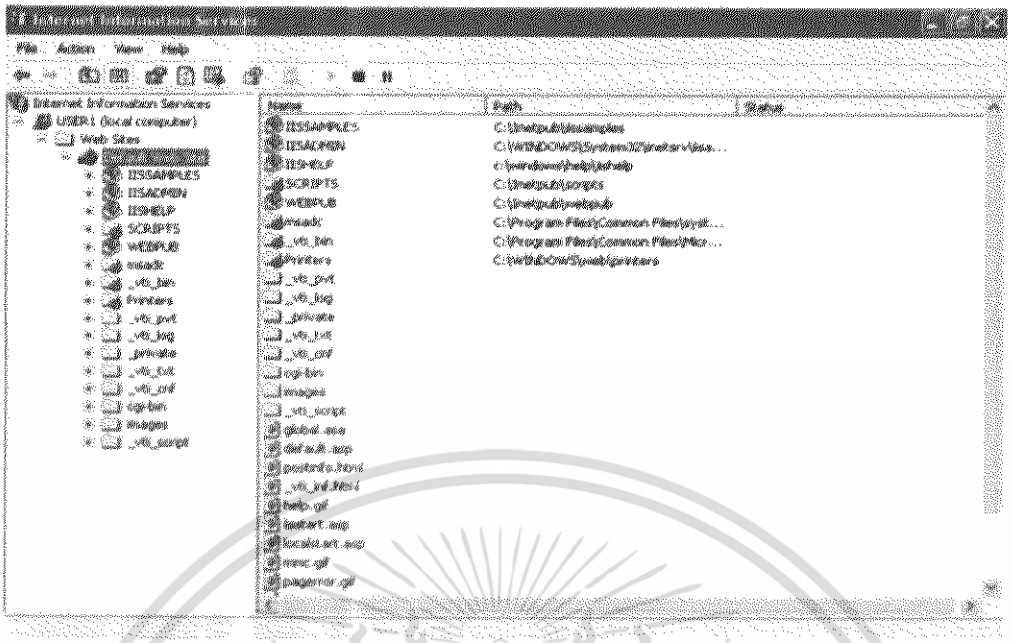
ทำการลงโปรแกรม.NET FRAMWORK 2.0 เพื่อให้โปรแกรม Position tracking ทำงานได้ เมื่อทำการติดตั้งสำเร็จแล้วก็ทำการติดตั้ง IIS ให้กับเครื่องคอมพิวเตอร์เพื่อทำหน้าที่เป็น Server เมื่อลง IIS แล้ว มาตั้งค่าการใช้งาน หลังจากที่เราลง IIS เมื่อเปิด Browser ขึ้นมาและพิมพ์ http://localhost ซึ่งจะแสดงได้ดังรูปที่ 3.16



รูปที่ 3.16 แสดงหน้าต่าง Web Server ที่เราทำการเรียก

ซึ่งโดยปกติแล้ว Path ตั้งต้นของ IIS จะถูก Set ไว้ที่ c:\Intepub\wwwroot เพราะฉะนั้นเมื่อสร้างไฟล์ของโปรแกรม Position tracking ก็ให้เก็บไว้ที่ Path นี้ แต่สามารถที่จะเซทให้ IIS เรียกไปที่ Path อื่นได้ หรือเซท Path ตั้งต้นที่ไม่ใช่ C:\Intepub\wwwroot ก็ได้โดยให้เข้าไปที่ Control Panel ดับเบิ้ลคลิกที่ Administrative Tools จะได้หน้าจอของ Administrative Tools และให้ดับเบิ้ลคลิกที่ Internet Information Services จะได้หน้าจอของ Internet Information Services ตามรูปที่ 3.17

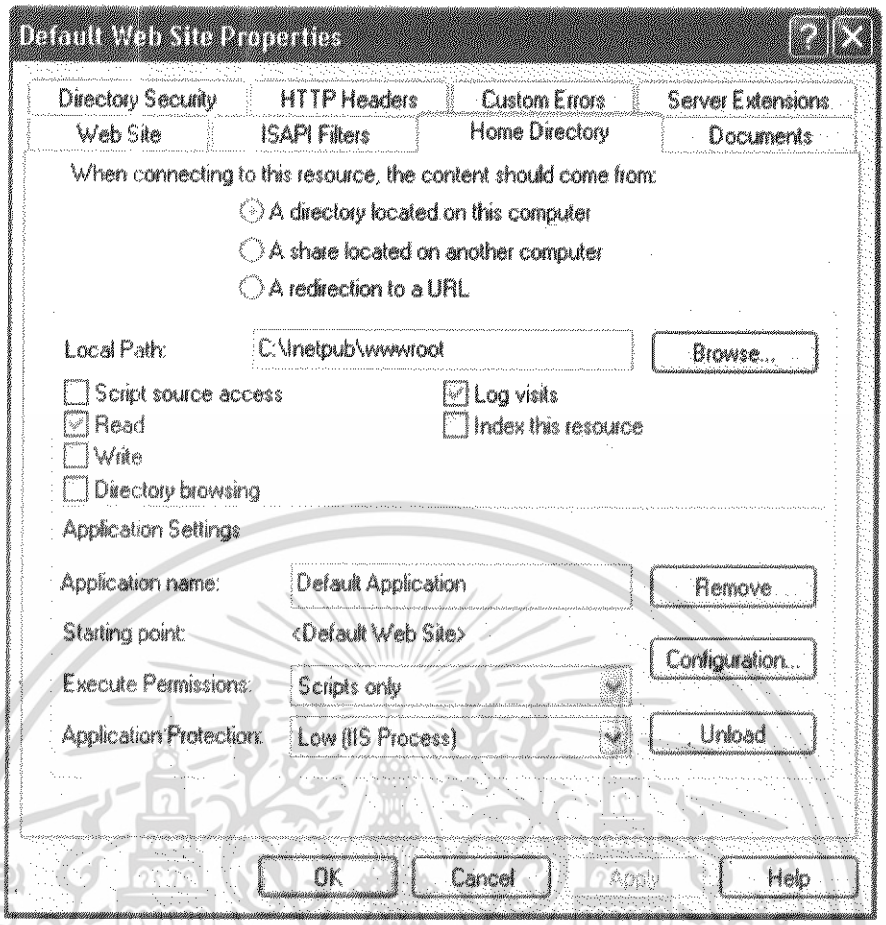
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.17 แสดงโครงสร้าง Path ตั้งต้นของ IIS

ถ้าต้องการที่จะเปลี่ยน Path ตั้งต้นของ IIS ให้คลิกขวาที่ Default Web Site แล้วเลือกไปที่ Properties ก็จะได้หน้าจอของ Default Web Site Properties จากนั้นให้คลิก Browse เพื่อเลือก Path ที่เราต้องการ เวลาเรียก http://localhost ตามด้วยชื่อไฟล์ IIS ก็จะชี้ไปที่ Path ที่ Set แทนที่จะเป็น C:\Intepub\wwwroot เหมือนอย่างเดิม ซึ่งโปรแกรมเราจะเก็บไฟล์ไว้ที่C:\Inetpub\wwwroot

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.18 แสดงหน้าต่าง Default Web Site Properties เพื่อทำการเซทค่าต่างๆของ IIS

เมื่อทำการเซทค่า IIS ได้แล้วก็จะสามารถใช้งานโปรแกรมในเครื่อง Serverให้สามารถเชื่อมต่อกับเครื่อง Client ได้ ซึ่งในการใช้งานจริงเราจะใช้ URL ของตัว Web Service เพื่อให้เครื่อง Client เครื่องอื่นสามารถเข้ามาใช้บริการโปรแกรม Position tracking ได้ เพื่อให้มีความหลากหลายในการใช้งานมากขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

ผลการทดลอง

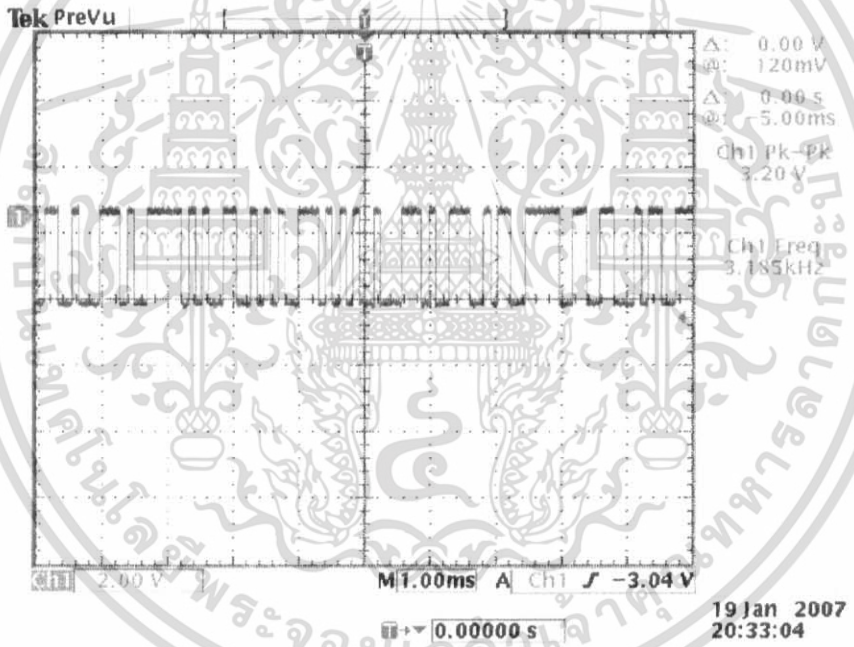
ในบทนี้จะกล่าวถึงการดำเนินงาน โดยประมวลความรู้ที่ศึกษามาทั้งหมด มาประกอบกัน และประยุกต์ใช้ ออกมาเป็นผลงานที่สมบูรณ์ แบ่งเป็นขั้นตอนการดำเนินงานดังนี้

4.1 ทดสอบการรับค่าจากโมดูลเครื่องรับ GPS

วัตถุประสงค์ของการทดลอง

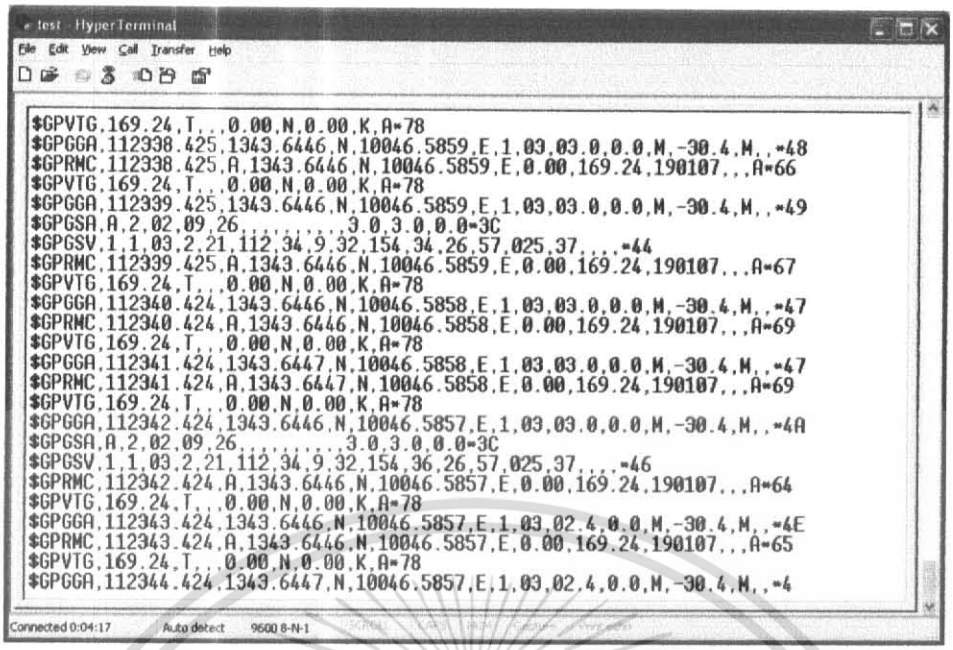
1. ทดสอบการทำงานของโมดูลเครื่องรับ GPS
2. ตรวจสอบผลของข้อมูลที่รับได้ว่าถูกต้องตามความต้องการหรือไม่
3. สามารถนำข้อมูลที่รับไปใช้ได้งานได้ถูกต้อง

จากรูปที่ 4.1 เป็นการทดลองการรับข้อมูลจากเครื่องรับสัญญาณ GPS โดยต่อผ่านวงจร MAX232 พิกัดเป็นตัวอย่างการทดสอบรับสัญญาณ GPS บนทางด่วนบางโคล่ -แข้งวัฒนะ



รูปที่ 4.1 ผลการรับข้อมูลจากเครื่องรับ GPS

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.2 ผลการรับข้อมูลจากเครื่องรับ GPS ผ่านทาง HyperTerminal

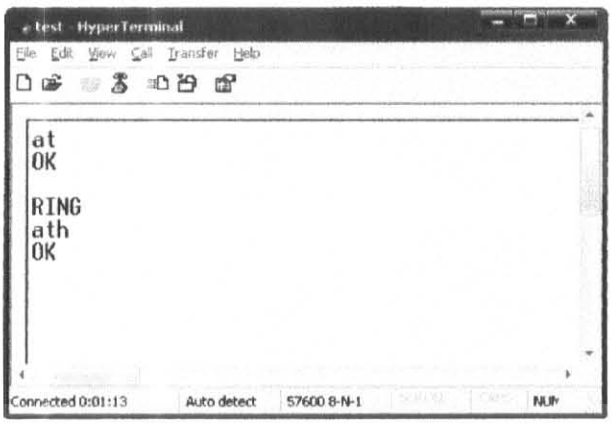
4.2 การเชื่อมต่อโทรศัพท์มือถือกับ Computer

ในการทำการติดต่อกับโทรศัพท์เคลื่อนที่จากคอมพิวเตอร์ โดยใช้โปรแกรม Hyper terminal ส่งชุดคำสั่ง AT COMMAND ผ่านสาย Data Link แล้วดูผลตอบสนองจากโทรศัพท์มือถือ

รูปที่ 4.3 โทรศัพท์เคลื่อนที่รุ่น Siemens S45

เมื่อเซตอัตราการรับส่งข้อมูลระหว่างโทรศัพท์มือถือกับ PC ให้ตรงกัน สำหรับมือถืออัตราการรับส่งข้อมูลสามารถเซตได้จากเมนูในตัวเครื่องโทรศัพท์เคลื่อนที่เอง ซึ่งโทรศัพท์เคลื่อนที่รุ่นที่ใช้เซตไว้ให้อยู่ที่อัตรา 19200 bps และจะทำการทดสอบโดยใช้คำสั่ง at <enter> หากระบบเชื่อมต่อได้จะส่งตอบกลับมาว่า OK จากนั้นจึงทดสอบคำสั่งพื้นฐานต่างๆ เพื่อสั่งการโทรศัพท์เคลื่อนที่ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.4 ผลการทดสอบคำสั่งพื้นฐานโทรศัพท์มือถือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3 ทดสอบการทำงานของวงจร TCP/IP Stack

การทดสอบสอบการทำงานของวงจร TCP/IP Stack เมื่อเชื่อมต่อกับโทรศัพท์เคลื่อนที่นั้น สามารถทำได้ดังนี้

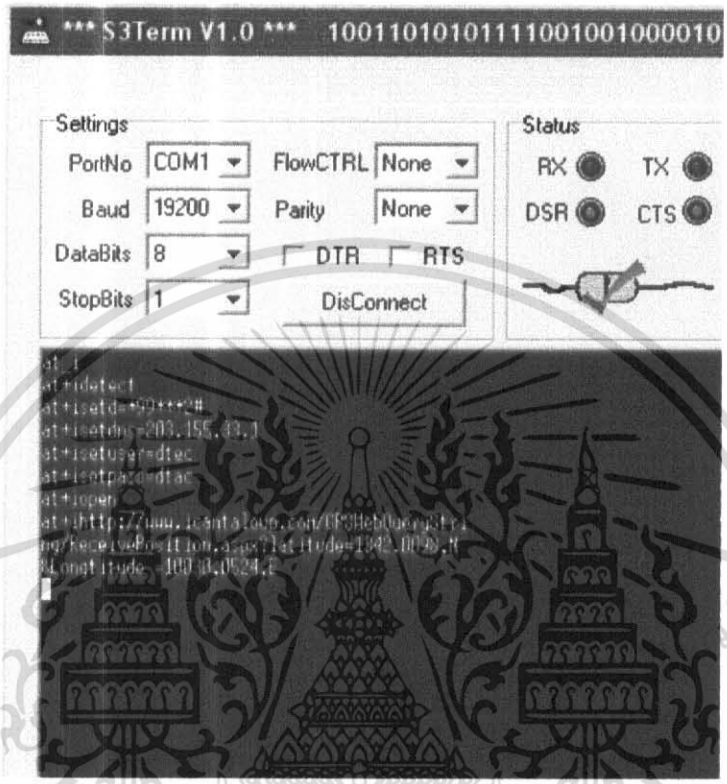
- เชื่อมต่อวงจร กับโทรศัพท์ และคอมพิวเตอร์ทางพอร์ตคอม 1
- ใช้โปรแกรม HyperTerminal และตั้งค่าบอดเรทเท่ากับ 19200 BPS
- หาวงจรใช้งานได้จะขึ้น +I_READY ดังรูป
- เซตค่าต่างเพื่อเชื่อมต่อกับเครือข่าย GPRS

รูปที่ 4.5 แสดงผลการทำงานของวงจร TCP/IP ขณะเชื่อมต่อโทรศัพท์เคลื่อนที่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.4 การจำลองการทำงานโปรแกรมไมโครคอนโทรเลอร์

จากการเขียนโปรแกรมภาษาแอสซิมบลีเพื่อทำการตัดค่าพิกัดและส่งคำสั่งในการเชื่อมต่อ กับวงจร TCP/IP Stack จะจำลองการทำงานโดยใช้โปรแกรม S3term. เพื่อทำการรับค่าคำสั่งและพิกัด ผลการจำลองการทำงานของโปรแกรมจะเป็นดังรูป 4.6



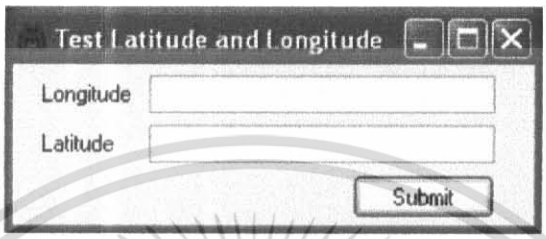
รูปที่ 4.6 ผลการจำลองการทำงานของโปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

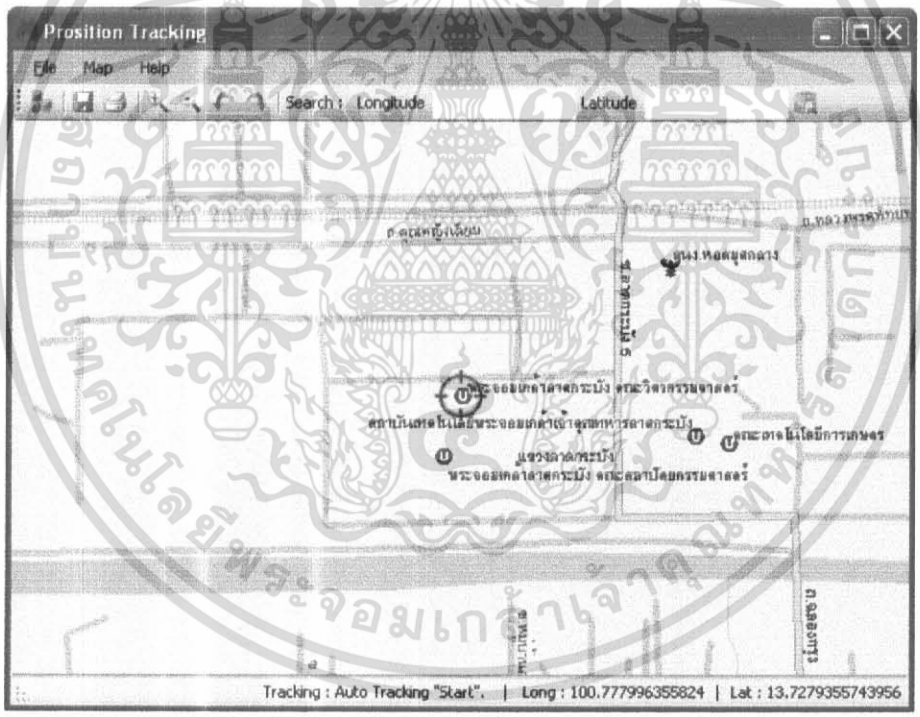
4.5 การทดลองการพล็อตค่าตำแหน่งละติจุด ลองติจุด โดยเป็นการส่งข้อมูลระหว่าง Computer กับ Computer

ผลการพล็อตตำแหน่งพิกัดในแผนที่

เริ่มต้นโดยการเปิดโปรแกรม Test ตำแหน่งพิกัดละติจุด ลองติจุด ที่อยู่ในส่วน Client เพื่อทำการทดสอบการทำงานของโปรแกรม Position Tracking ที่อยู่ในฝั่ง Server และเมื่อทำการเปิดโปรแกรม Position Tracking จะแสดงตำแหน่งพิกัดเริ่มต้นดังรูปข้างล่าง



รูปที่ 4.7 โปรแกรม Test เมื่อยังไม่มีการค่าพิกัด



รูปที่ 4.8 ค่าเริ่มต้นเมื่อยังไม่มีการส่งค่าตำแหน่งจากเครื่อง Client เข้ามายังเครื่อง Server

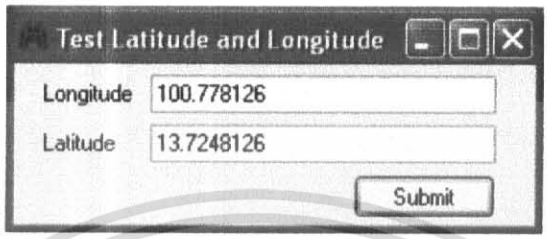
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากนั้นทำการใส่ตำแหน่งพิกัดละติจุด ลองติจุดลงในช่อง Longitude, Latitude ดังนี้

Longitude ที่ 100.778126

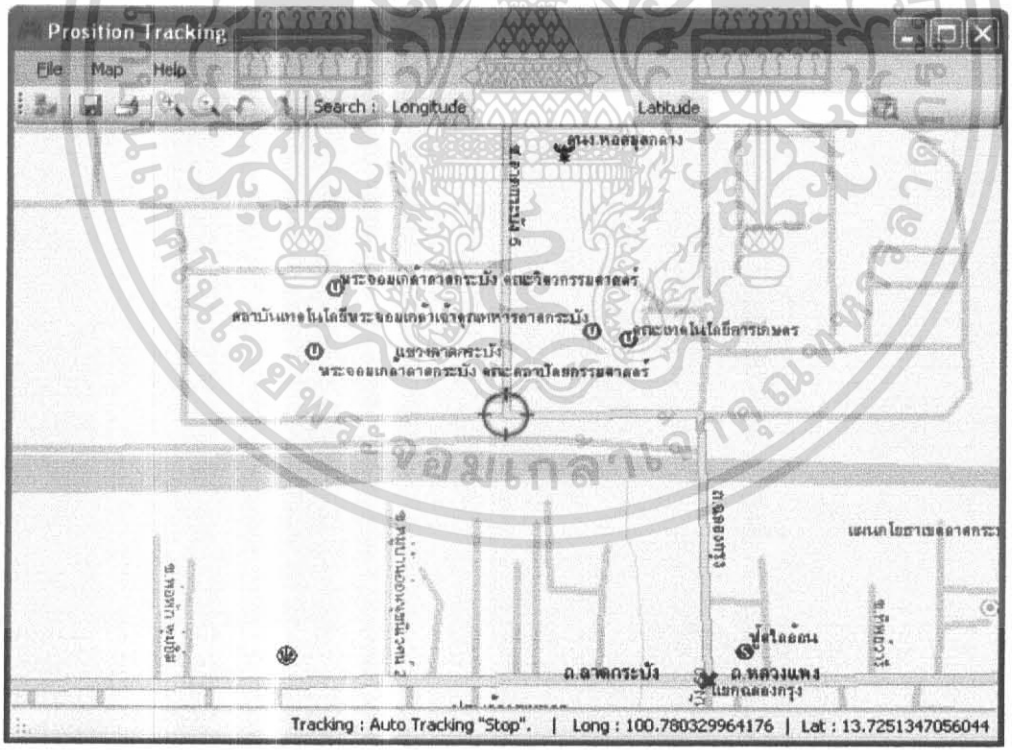
Latitude ที่ 13.7248126

ซึ่งเป็นตำแหน่งหัวมุมถนนหน้าคณะสถาปัตยกรรมศาสตร์ แล้วทำการ Submit เพื่อส่งพิกัดละติจุด ลองติจุด มายังเครื่อง Server



รูปที่ 4.9 เมื่อใส่ตำแหน่งพิกัดละติจุด ลองติจุด

เมื่อตัวเครื่อง Server ได้รับตำแหน่งพิกัดละติจุด ลองติจุด จากเครื่อง Client โปรแกรม Position Tracking ก็จะแสดงตำแหน่งพิกัดนั้นออกมา จะเห็นว่าเป็นตำแหน่งหัวมุมถนนหน้าคณะสถาปัตยกรรมศาสตร์ ซึ่งจะตรงกับตำแหน่งพิกัดที่ส่งมาจากโปรแกรม Test ของเครื่อง Client ดังรูปที่ 4.10

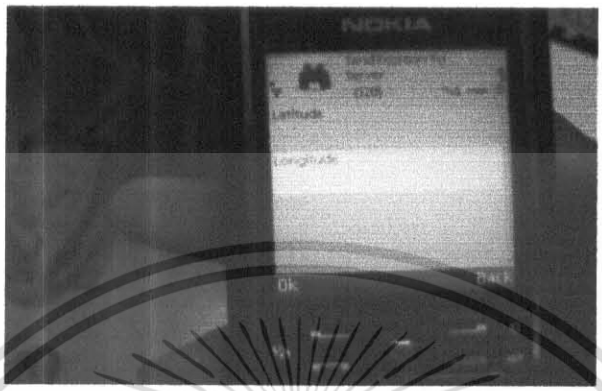


รูปที่ 4.10 การแสดงตำแหน่งพิกัดที่เครื่อง Server ได้รับพิกัดมาจากเครื่อง Client

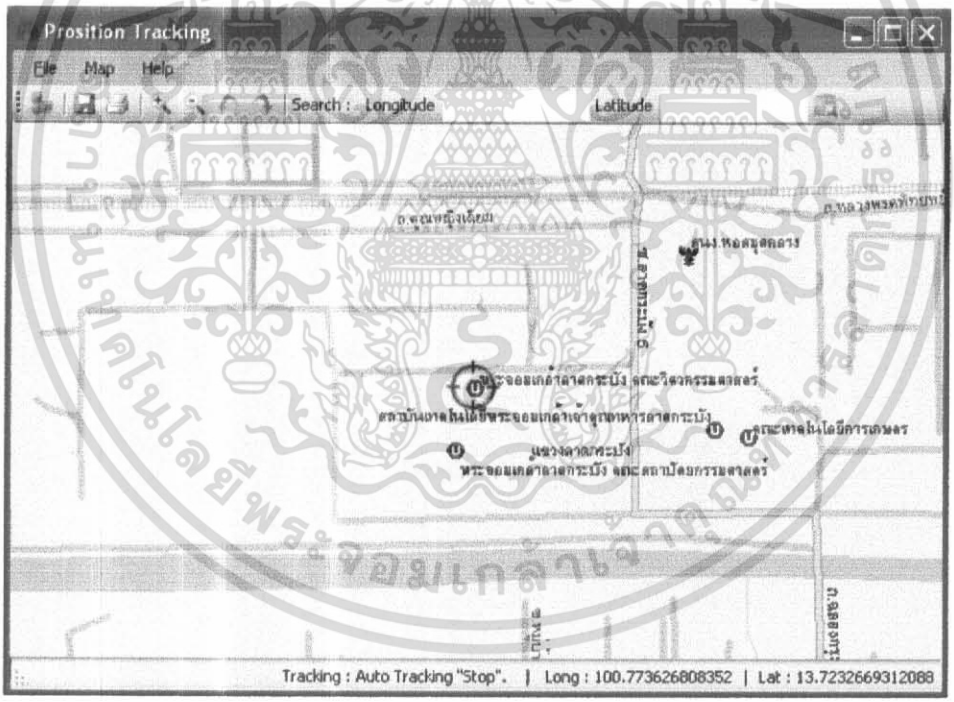
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.6 ผลการส่งค่าพิกัดละติจุด ลองติจุด ทางโทรศัพท์มือถือผ่านเครือข่าย GPRS

เมื่อทำการติดตั้งโปรแกรม GPSTrackingMobile เรียบร้อยแล้ว จากนั้นให้เปิดโปรแกรม GPSTrackingMobile จะปรากฏหน้าจอดังรูปที่ 4.11 เปิดโปรแกรม Position Tracking เพื่อสังเกตตำแหน่งพิกัดในแผนที่ จะเห็นได้ว่าตำแหน่งพิกัดอยู่ในตำแหน่งเริ่มต้น คือคณะวิศวกรรมศาสตร์ ดังรูปที่ 4.12



รูปที่ 4.11 แสดง Application ในการส่งพิกัดละติจุด ลองติจุด



รูปที่ 4.12 ตำแหน่งเริ่มต้นเมื่อยังไม่มีการส่งพิกัดตำแหน่ง

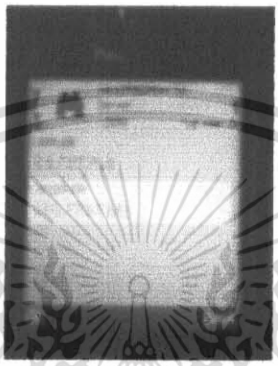
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากนั้นใส่ค่าตำแหน่งพิกัดละติจุด ลองติจุด แล้วกดปุ่ม OK ระบบจะทำการเชื่อมต่อเข้ากับเครือข่าย GPRS จากนั้นก็จะสามารถส่งพิกัดไปยังเครื่อง Server โดยจะแสดงข้อความ Send Position to Server Complete เมื่อทำการส่งสำเร็จ แต่ถ้าการส่งพิกัดไม่สำเร็จก็จะแสดงข้อความ Sending Position failure และพิกัดในการส่งนี้คือ

Latitude ที่ 13.727750

Longitude ที่ 100.778738

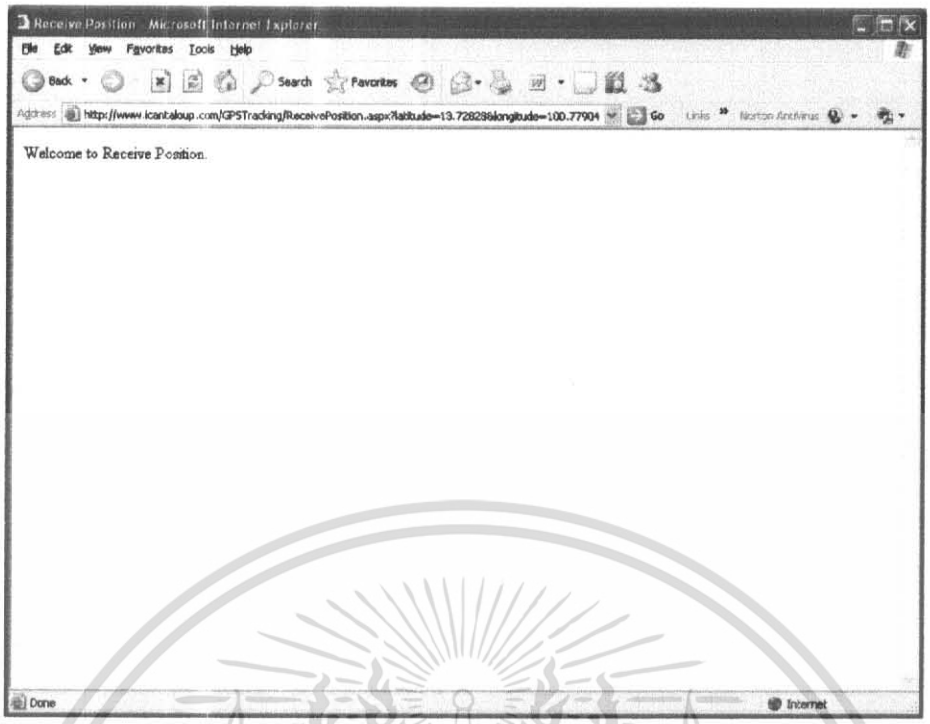
โดยพิกัดนี้คือสำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง



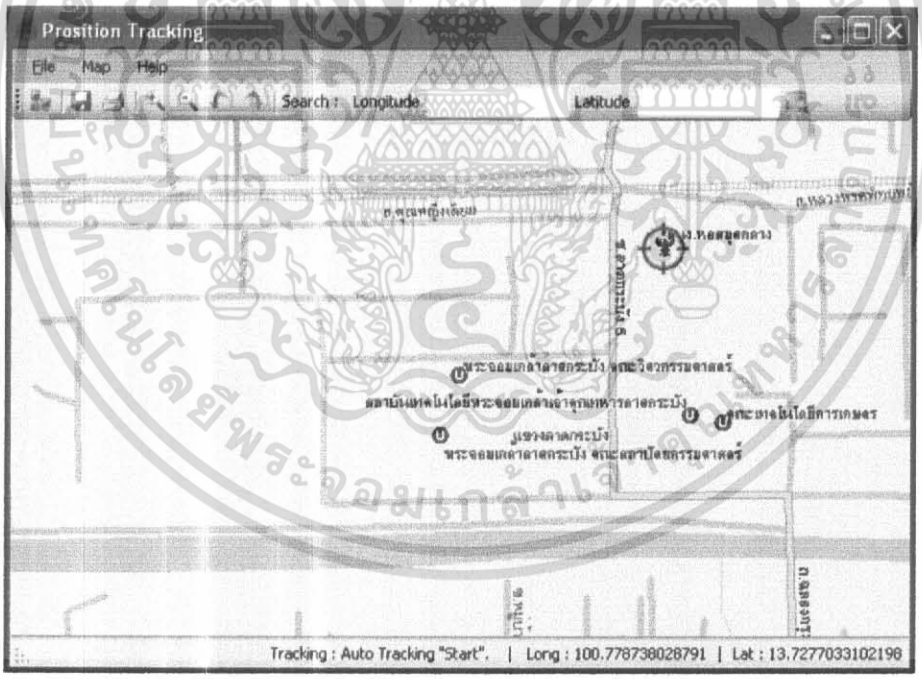
รูปที่ 4.13 แสดงการใส่ค่าพิกัดละติจุด ลองติจุด แล้วทำการส่งตำแหน่งพิกัดผ่านเครือข่าย GPRS

เมื่อมีการส่งพิกัดดังกล่าวแล้วนั้น เครื่อง Server ก็จะรับค่าพิกัดที่ส่งมาจากโทรศัพท์มือถือผ่านเครือข่าย GPRS โดยเครื่อง Server นั้นจะมีการแลกเปลี่ยนข้อมูลกันกับตัว Web Service เพื่อมาอัพเดตตำแหน่งพิกัดในโปรแกรม Position Tracking ของเครื่อง Server ดังแสดงดังรูปที่ 4.14

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.14 แสดงหน้าเว็บเมื่อมีการส่งพิกัดตำแหน่ง



รูปที่ 4.15 แสดงตำแหน่งพิกัดที่รับมาจากโทรศัพท์มือถือผ่านเครือข่าย GPRS

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5
สรุป

5.1 สรุปผลการพัฒนาโครงงาน

โครงงานนี้ได้สำเร็จตามจุดประสงค์และขอบเขตที่ได้ตั้งไว้คือ สามารถรับค่าที่ได้จากเครื่องรับ GPS เพื่อนำตำแหน่งที่รับได้นั้น รวมทั้งสามารถทำการหาเส้นทางได้ตามที่ผู้ใช้ต้องการได้

จากการทำโครงงานนี้ เป็นต้องศึกษาความรู้หลายด้าน คือ ระบบ GPS, การเขียนโปรแกรมบนระบบฝังตัว ซึ่งความรู้ที่ได้จากการศึกษาและการทำโครงงานนี้ ทำให้เข้าใจระบบวิศวกรรมสารสนเทศทางด้านสื่อสารมากขึ้น เพราะเป็นการนำความรู้ด้าน การสื่อสาร, เทคโนโลยีใหม่ และอิเล็กทรอนิกส์ มาใช้รวมเข้าด้วยกัน

5.2 ปัญหาที่เกิดขึ้น

5.2.1 ข้อจำกัดของระบบ GPS

สัญญาณที่ส่งมายังเครื่องรับ GPS ไม่สามารถทะลุผ่านของแข็ง เช่น อาคาร หรือใต้ทางด่วนได้ และสถานที่ต้องไม่แคบเกินไปเพราะเครื่องรับ GPS จะต้องรับสัญญาณจากดาวเทียม GPS อย่างน้อยถึง 3 ดวง ข้อจำกัดอีกอย่างหนึ่งก็คือในเรื่องความคลาดเคลื่อนของข้อมูลตำแหน่งพิกัด ซึ่งเครื่องรับ GPS ทั่วไปจะรับสัญญาณได้ตำแหน่งที่ไม่เที่ยงตรงมากเท่ากับของทหาร ส่วนความแม่นยำและความไวของเครื่องรับ GPS นั้นขึ้นอยู่กับการที่เลือกใช้เครื่องรับ GPS

5.2.2 ข้อมูลแผนที่

ข้อมูลแผนที่ดิจิตอล เป็นการเก็บข้อมูลตำแหน่งทีละจุดจากสถานที่จริง ตามจุดที่ต้องการทั้งหมด โดยถ้ายิ่งความละเอียดสูงก็จะทำให้ได้ข้อมูลที่ละเอียดมาก การเก็บข้อมูลดังกล่าวจึงทำให้ต้องมีค่าใช้จ่ายเกี่ยวกับฐานข้อมูลที่มีราคาสูงไปด้วย

5.3 ข้อเสนอแนะ

ในการนำไปใช้งานทางด้านธุรกิจ นอกจากระบบต้องมีประสิทธิภาพที่ดีแล้วค่าใช้จ่ายก็เป็นส่วนสำคัญ ในการใช้งานจริงต้องติดต่อหาแผนที่ดิจิตอลให้ได้ ซึ่งมีที่ราคาที่สูง และส่วนที่ใช้เชื่อมต่อกับโทรศัพท์มือถือรุ่นต่างๆควรจะเพิ่มพอร์ตให้มากและหลากหลายขึ้นเพื่อให้ใช้ได้กับมือถือรุ่นอื่นๆด้วย

ส่วนของการแสดงผล สามารถพัฒนาให้การแสดงภาพของแผนที่มีรายละเอียดข้อมูลมากยิ่งขึ้น เพราะการแสดงชื่อของถนนและการแสดงผลเมื่อทำการหาเส้นทางยังไม่สมบูรณ์เท่าที่ควร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หนังสืออ้างอิง

1. วรายุทธ คัมภีราวัฒน์ และ สุชาดา ไชยสุข, "ระบบควบคุมอุปกรณ์ไฟฟ้าด้วย SMS", ปริญญานิพนธ์ คณะวิศวกรรมศาสตร์ วิศวกรรมไฟฟ้า มหาวิทยาลัยขอนแก่น, 2547.
2. รศ.ดร.สุวิพล สิทธิชีวภาค, "เทคโนโลยีของโลคัลแอเรียเนตเวอร์ค", สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง, 2539.
3. รศ.ดร. สมยศ จุณปิยะ, "การประยุกต์ใช้งานไมโครคอนโทรลเลอร์ ตระกูล MCS-51", สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง, 2543.
4. พิชญ จุลศิริ, "ครบเครื่องเรื่อง GPS", บริษัท ซีเอ็ดยูเคชัน จำกัด, พิมพ์ครั้งที่ 1, 2549.
5. อนันต์ คำภีระ, "ความรู้เกี่ยวกับแผนที่เบื้องต้น", [Online].เข้าถึงได้จาก: http://www.physics.sci.rit.ac.th/sciencefac/artic/map/map.htm, 2004.
6. ศุภฤกษ์ ชัยชนะ, "แผนที่ตัวเลข", [Online] เข้าถึงได้จาก http://www.geocites.com/sup2822911 , 2004.
7. โปรแกรม SmartMap Viewer 3.0, บริษัท MappointAsia (Thailand) Co, Ltd

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ก.
โปรแกรมที่ใช้ในโครงงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Source code MCS-51

```
ORG     20H     ;SET DATA AREA In Internal RAM
DAT_STR:    DS    1
DAT_AREA:   DS    26
STACK:              DS          1
GPS                 BIT     P3.3
GPRS                BIT     P3.4
BUZZER              BIT     P3.7
IO                  BIT     3.5     ;LED
SW_PORT             EQU         P1
LED                 EQU         P0    ; TEST to P0 ,Real - P1
;-------------------------SET  MAIN  PROGRAM -------------------------
ORG   0
LJMP        MAIN
;---------------------------------------------------------------------
ORG         23H
LJMP        RXINTR      ;Serial Port Interrupt Routine
;**************** START  MAIN  PROGRAM *************************
MAIN:
            CLR         EA
            MOV         SP,#STACK       ;inti. Parameter
            MOV         PSW,#0H         ;inti. Parameter

;*********************************************************************
SP1:        MOV   R5,#1
            LCALL BAUD192
            LCALL SETGPRS
DSP2:       LCALL BAUD96
            LCALL RD_GPS      ;Read GPS Data '$GPRMC  data
            LCALL BAUD192
;           LCALL TX2TEL
            LCALL SEND2TEL    ;send Data Lat/long to Moblie gprs
            LCALL DLY15m
;           DJNZ  R5,DSP2
            SJMP  $
***************** END  MAIN  PROGRAM ****************************
BAUD192:
        CLR     EA
        MOV     SCON,#01010000B     ;set Mode 1  8 bit Asynchronus
ren = 1 Enable Serial Port
        MOV  TMOD,#00100000B        ;set mode (timer)  ;mode 2 Auto
Reload 8 bit)
        MOV  TCON,#01000000B        ;Timer0,1 internal count
        MOV  TH1,#0FBH              ;19200 BPS  at x'tal 18.432MHz
        MOV  TL1,#0FBH              ;Change to #0FD when use x'tal
at 11.092 MHz
        MOV     LED,#0FEH           ;Set LED off
        MOV      A,#80H             ;SET   PCON.7 or SMOD = 1
        ORL      PCON,A             ; TO 19200 bps  baud rate
        MOV     SW_PORT,#0B6H ;SET LED 2 on/ Switch to GPS
D=0;C=1;B=1;A=0
        MOV     R6,#0FFh            ;Wait for TCP/IP Module Respond
        DJNZ    R6,$
        MOV     LED,#07FH
        SETB    EA
        LCALL   HBEEP
        RET
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
BAUD96:    CLR      EA
           MOV      SCON,#01010000B      ;set Mode 1  8 bit Asynchronus
ren = 1 Enable Serial Port
           MOV  TMOD,#00100000B          ;set mode (timer)  ;mode 2 Auto
Reload 8 bit)
           MOV      TCON,#01000000B      ;Timer0,1 internal count
                    MOV      TH1,#0FBH             ;19200 BPS  at x'tal
18.432MHz  >>
                    MOV      TL1,#0FBH    ;Change to #0FD when use x'tal
at 11.092 MHz
                    MOV      A,#0EFH      ;SET   PCON.7 or SMOD = 0
                    MOV      PCON,#0      ; TO 9600 bps  baud rate
                    MOV   SW_PORT,#0E9H ;SET LED 1 on / Switch to
      GPSD=1;C=0;B=0;A=1
                    MOV   LED,#10111111B
                    MOV   R6,#0FFh       ;Wait for GPS Module Respond
                    DJNZ  R6,$
                    SETB        EA
                    LCALL   HBEEP
                    RET
;-------------------------------------------------------------------------
             MOV   LED,#11111011b       ;Test State 6 sequence of
program
             SJMP  $
;-------------------------------------------------------------------------
RD_GPS:          CLR      EA
                 LCALL    RX_BYTE       ;RECIVED '$' First Data Format
                 CJNE     A,#'$',RD_GPS
                 MOV      LED,#11011111B          ;Test State 1
sequence of program
;---------------------- Test  String ' $GPRMC ' from GPS -------------
                 LCALL    RX_BYTE
                 CJNE     A,#'G',RD_GPS
                 LCALL    RX_BYTE
                 CJNE     A,#'P',RD_GPS
                 LCALL    RX_BYTE
                 CJNE     A,#'R',RD_GPS
                 LCALL    RX_BYTE
                 CJNE     A,#'M',RD_GPS
                 LCALL    RX_BYTE
                 CJNE     A,#'C',RD_GPS
      MOV   LED,#11101111B           ;Test State 2 sequence of program
;---------------- Skip Data 14 Byte of date/time from GPS-------------
      MOV     R0,#14
      LCALL   RX_BYTE                ; Read Skip data seq. 14 byte
      DJNZ  R0,Skipdat
      MOV   LED,#11110111B           ;Test State 3 sequence of program
;--------------- Read Data 24-26 Byte lat/long end at 'E' ------------
      MOV   R0,#24
      MOV   R1,#DAT_AREA
GPR_RD:     LCALL RX_BYTE
            CJNE    A,#'E',WR_DAT
            MOV   @R1,A
            SJMP    OUT1
WR_DAT:     MOV   DAT_STR,A
            MOV   LED,#11111011b     ;Test State 4 sequence of program
            MOV   @R1,DAT_STR
            INC   R1
            DJNZ  R0,GPR_RD
OUT1:       RET
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
;---------------------- SEND Data TO Mobile Phone -------------------
TX2TEL:          NOP
ECHO_2:          ;SETB   TR1
                 ; CLR    ES
                 MOV    R1,#DAT_AREA
                 MOV    R0,#24
LOOP2:           MOV    A,@R1
                 LCALL      TX_BYTE
                 INC    R1
                 DJNZ   R0,LOOP2
                 MOV    LED,#11110111b              ;Test State 5
sequence of program
                 RET
;-------------------------------------------------------------------------
;RXRET:     JNB     RI,$
           CLR     RI
           POP     ACC
           SETB    EA
           RETI
;-------------------------------------------------------------------------
;TXRET:     MOV     SBUF,A
           JNB     TI,$
           CLR     TI
           POP     ACC
           SETB    EA
           RETI
;-------------------------------------------------------------------------
RX_BYTE:   JNB     RI,$
           MOV     A,SBUF
           CLR     RI
           RET
;-------------------------------------------------------------------------
TX_BYTE:   MOV     SBUF,A
           JNB     TI,$
           CLR     TI
           RET
;-------------------------------------------------------------------------
CRLF:      MOV     A,#0DH
           LCALL   TX_BYTE
           MOV     A,#0AH
           LCALL   TX_BYTE
           RET
;------------------------- TEST RESPOND of Device --------------------
I_OK:
      LCALL    RX_BYTE
               CJNE    A,#'+',I_OK
      LCALL    RX_BYTE
               CJNE    A,#'I',I_OK
               LCALL   RX_BYTE
               CJNE    A,#'_',I_OK
               LCALL   RX_BYTE
               CJNE    A,#'O',I_OK
               LCALL   RX_BYTE
               CJNE    A,#'K',I_OK
      MOV   A,#99H
      RET
;-------------------------------------------------------------------------
NotRDY:    LCALL LBEEP
           MOV   A,#33H
           MOV   LED,#10111101b
           RET
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
;----------------- SET GPRS Baudrate at 19200 and Dial ---------------
SETGPRS:    MOV   DPTR,#AT_I
            LCALL       SEND_STR
            MOV   DPTR,#IDETECT
            LCALL       SEND_STR
            MOV   DPTR,#DIAL
            LCALL       SEND_STR
            MOV   DPTR,#DNS
            LCALL       SEND_STR
            MOV   DPTR,#SETUSER
            LCALL       SEND_STR
            MOV   DPTR,#SETPASS
            LCALL       SEND_STR
            MOV   DPTR,#I_OPEN
            LCALL       SEND_STR
            RET
SEND2TEL:
            MOV   DPTR,#I_HTTP
            LCALL       SENDSTR2
            LCALL CRLF
            LCALL I_OK
            RET

;----------------- SEND String Routine Baudrate19200 -----------------
SEND_STR:   MOV   A,#00
LOOP9:      MOVC  A,@A+DPTR
            CJNE  A,#0AH,LOOP8
            LCALL TX_BYTE
            LCALL I_OK
            RET
LOOP8:      LCALL TX_BYTE
            INC   DPTR
            MOV   A,#0
            SJMP  LOOP9
;- SEND Latitude+Data and Longtitude+Data to web Site Baudrate19200 -
-----------
SENDSTR2:   MOV   A,#00
LOOP5:      MOVC  A,@A+DPTR
            CJNE  A,#'=',LOOP4
            LCALL TX_BYTE
            SJMP  SENDLAT
LOOP4:      LCALL TX_BYTE
            INC   DPTR
            MOV   A,#0
            SJMP  LOOP5
;-------------------------------------------------------------------
SENDLAT:
            MOV   LED,#11111101b
            MOV   R1,#DAT_AREA
            MOV   R0,#11
LOOPP2:     MOV   A,@R1
            LCALL       TX_BYTE
            INC   R1
            DJNZ  R0,LOOPP2
;---------------------- Send String longtitude ----------------------
            MOV   DPTR,#Longtitude
            MOV   A,#00
LOOPP9:     MOVC  A,@A+DPTR
            CJNE  A,#'=',LOOPP8
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            LCALL TX_BYTE
            INC   R1
;---------------- Send Number of longtitude From GPS -----------------
SENDLONG:   MOV   R0,#12
LOOPP4:     MOV   A,@R1
            LCALL  TX_BYTE
            INC   R1
            DJNZ  R0,LOOPP4
            RET

LOOPP8:     LCALL TX_BYTE
            INC   DPTR
            MOV   A,#0
            SJMP  LOOPP9

;-------------------- Delay 15 minute for Read next Position and send
DLY15m:                 ;UnderconStruction Command
            NOP
            RET
GPRMC:      DB 'GPRMC'
GPRSB:      DB 'AT+IPR=57600',0Dh,0AH
EEP_WR:     DB 'AT&W',0Dh,0AH
AT_I:       DB 'at+i',0Dh,0AH
GPRSTB:     DB 'at+isetbaud=192',0Dh,0AH
DIAL:       DB 'at+isetd=*99***2#',0Dh,0AH
DNS:        DB 'at+isetdns=203.155.33.1',0Dh,0AH
SETUSER:    DB 'at+isetuser=dtec',0Dh,0AH
SETPASS:    DB 'at+isetpass=dtac',0Dh,0AH
I_OPEN:     DB 'at+iopen',0Dh,0AH
IDETECT:    DB 'at+idetect',0Dh,0AH
            DB 'AT+iOPEN=0',0Dh,0AH
            DB 'I_READY',0Dh,0AH
I_HTTP:     DB
     'at+ihttp://www.icantaloup.com/GPSTracking/ReceivePosition.aspx
?latitude='
Longtitude: DB ' &Longtitude ='
$INCLUDE        "voice.ASM"
END
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Source code Server Monitor เขียนโดยใช้ภาษาC#

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Text;
using System.Windows.Forms;
using System.Collections;
using System.Drawing.Printing;
namespace ServerMonitor
{
    public partial class Form1 : Form
    {
        GpsToolsNET.Position gpsPosition;
        bool startTracking;
   public Form1()
        {
            InitializeComponent();
            InitializeGPS();
            startTracking = true;
            ManageTimer();
        }
        /// <summary>
        /// This function use for initial config gps
        /// </summary>
        public void InitializeGPS()
        {
         try
         {
           GpsToolsNET.License gpsLicense;
       gpsLicense = new GpsToolsNET.License();
       gpsPosition = new GpsToolsNET.Position();

       //
       // Input serial number for GPS Tools SDK

gpsLicense.LicenseKey="pCn7g92vpPojTKIjyxAngvmbq4paZf6ToBq3";
       //
       // Set default position
         UpdatePosition();

       //
       // Set location and size for show map in main programs
                trackingMap.Location = this.Location;
                trackingMap.Size = this.Size;
                trackingMap.Open("Map/bkkMap.MapLib");
                trackingMap.Position = this.gpsPosition;
                trackingMap.Update();
GpsViewNET.Icon objIcon = trackingMap.NewIcon("positionPoint2");
                objIcon.Position = this.gpsPosition;
                trackingMap.Update();
            }
            catch (Exception ex)
            {
                MessageBox.Show("initializeGPS : " + ex.Message);
            }
        }
  private void trackingMap_SizeChanged(object sender, EventArgs e)
        {
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            trackingMap.Update();
        }
        private void UpdatePosition()
        {
            try
            {
TrackingService.Service objGPSTracking = new
ServerMonitor.TrackingService.Service();
this.gpsPosition.Latitude = objGPSTracking.GetLatitude();
this.gpsPosition.Longitude = objGPSTracking.GetLongitude();
            }
            catch (Exception)
            {
 this.gpsPosition.Longitude = 100.776322;
this.gpsPosition.Latitude = 13.726211;
//MessageBox.Show(ex.Message, "Server Monitor", MessageBoxButtons.OK,
MessageBoxIcon.Error);
            }
trackingMap.Update();
        }
private void trackingMap_OnMouseMove(MouseButtons Button, short Shift,
GpsToolsNET.Position objPosition, GpsViewNET.Point objPoint)
        {
tsLblLongitude.Text = "Long : " +
Convert.ToString(objPosition.Longitude);
tsLblLatitude.Text = "Lat : " +
Convert.ToString(objPosition.Latitude);
        }
        private void trackingMap_OnMouseDown(MouseButtons Button,
short Shift, GpsToolsNET.Position objPosition, GpsViewNET.Point
objPoint)
        {
            trackingMap.Position = objPosition;
            trackingMap.Update();
        }
        #region " Tool Bar Events "
        private void tsBtnPrint_Click(object sender, EventArgs e)
        {
            printMap();
        }
        private void tsBtnExportToImage_Click(object sender,
EventArgs e)
        {
            exportImage();
        }
         private void tsBtnZoomIn_Click(object sender, EventArgs e)
        {
            ZoomIn();
        }
         private void tsBtnZoomOut_Click(object sender, EventArgs e)
        {
            ZoomOut();
        }
        private void tsBtnRotateLeft_Click(object sender, EventArgs e)
        {
            RotateLeft();
        }
       private void tsBtnRotateRight_Click(object sender, EventArgs e)
        {
            RotateRight();
        }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        #endregion
        #region "Menu Bar Events "
        private void exportToImaToolStripMenuItem_Click(object
sender, EventArgs e)
        {
            exportImage();
        }
       private void printToolStripMenuItem_Click(object sender,
EventArgs e)
        {
            printMap();
        }
       private void zoomInToolStripMenuItem_Click(object sender,
EventArgs e)
        {
            ZoomIn();
        }
       private void zoomOutToolStripMenuItem_Click(object sender,
EventArgs e)
        {
            ZoomOut();
        }
       private void rotateRightToolStripMenuItem_Click(object sender,
EventArgs e)
        {
            RotateRight();
        }
       private void rotateLeftToolStripMenuItem_Click(object sender,
EventArgs e)
        {
            RotateLeft();
        }
       private void exitToolStripMenuItem_Click(object sender,
EventArgs e)
        {
            Application.Exit();
        }
        #endregion
        #region " Save Map To Image "
        public void exportImage()
        {
            try
            {
                saveFileDialogImage.Filter = "Jpeg
(*.jpeg)|*.jpeg|Bitmap (*.bmp)|*.bmp|Gif (*.gif)|*.gif|PNG
(*.png)|*.png";
                saveFileDialogImage.ShowDialog();
       if (saveFileDialogImage.FileName != "")
                {
                  switch (saveFileDialogImage.FilterIndex)
                    {
                        case 1:
                            trackingMap.ImageFormat =
System.Drawing.Imaging.ImageFormat.Jpeg;
                            break;

                        case 2:
                            trackingMap.ImageFormat =
System.Drawing.Imaging.ImageFormat.Bmp;
                            break;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                        case 3:
trackingMap.ImageFormat = System.Drawing.Imaging.ImageFormat.Gif;
                            break;

                        case 4:
trackingMap.ImageFormat = System.Drawing.Imaging.ImageFormat.Png;
                            break;
                    }
System.IO.MemoryStream memStream = trackingMap.GetImageStream();
// Create a file to write the stream into it.
System.IO.FileStream fs =
(System.IO.FileStream)saveFileDialogImage.OpenFile();
// Write the stream into the created file.
memStream.WriteTo(fs);
                    fs.Close();
                }
            }
            catch (Exception ex)
            {
                MessageBox.Show("exportImage", ex.Message);
            }
        }
        #endregion
        #region " Print Map "
        public void printMap()
        {
            try
            {
                PrintDocument doc = new PrintDocument();
                // Add event handler that handles printing
                doc.PrintPage += new
PrintPageEventHandler(doc_PrintPage);

                printDialogMap.ShowDialog();
                doc.PrinterSettings = printDialogMap.PrinterSettings;

                // Print! Will call doc_PrintPage()
                doc.Print();
            }
            catch (Exception ex)
            {
                MessageBox.Show("printMap", ex.Message);
            }
        }
        private void doc_PrintPage(object sender, PrintPageEventArgs
e)
        {
            try
            {
// BMP, GIF, PNG or JPEG (PNG is default)
trackingMap.ImageFormat = System.Drawing.Imaging.ImageFormat.Jpeg;
System.IO.MemoryStream memStream = trackingMap.GetImageStream();
// Make an image of map
                Image img = Image.FromStream(memStream);

// Draw image on print Graphics
                e.Graphics.DrawImage(img, new Point(100, 100));
// No more pages to print
                e.HasMorePages = false;
            }
            catch (Exception ex)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            {
                MessageBox.Show("doc_PrintPage", ex.Message);
            }
        }
        #endregion
        #region " Zoom "
        public void ZoomIn()
        {
            this.UseWaitCursor = true;
            try
            {
                trackingMap.Zoom += 1;
                trackingMap.Update();
            }
            catch (Exception)
            {
            }
            this.UseWaitCursor = false;
        }
        public void ZoomOut()
        {
            this.UseWaitCursor = true;
            try
            {
                trackingMap.Zoom -= 1;
                trackingMap.Update();
            }
            catch (Exception)
            {
            }
            this.UseWaitCursor = false;
        }
        #endregion
        #region " Rotate "
        public void RotateLeft()
        {
            this.UseWaitCursor = true;
            try
            {
                trackingMap.Rotation += 5;
                trackingMap.Update();
            }
            catch (Exception)
            {
            }

            this.UseWaitCursor = false;
        }
    public void RotateRight()
        {
            this.UseWaitCursor = true;
            try
            {
                trackingMap.Rotation -= 5;
                trackingMap.Update();
            }
            catch (Exception)
            {
            }
            this.UseWaitCursor = false;
        }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        #endregion
        private void tsBtnSearchByLatLong_Click(object sender,
EventArgs e)
        {
            try
            {
                if (tsTXTLongitude.Text != "" &&
tsTXTLatitude.Text != "")
                {
                    gpsPosition.Longitude =
Convert.ToDouble(tsTXTLongitude.Text);
                    gpsPosition.Latitude =
Convert.ToDouble(tsTXTLatitude.Text);
                    trackingMap.Update();
                }
                else
                {
                    MessageBox.Show("Please input Longitude and
Latitude", "Server Monitor", MessageBoxButtons.OK,
MessageBoxIcon.Warning);
                }
            }
            catch (Exception)
            {
                MessageBox.Show("Please input numberic only", "Server
Monitor", MessageBoxButtons.OK, MessageBoxIcon.Error);
            }
        }
        private void tmrRefreshTracking_Tick(object sender, EventArgs
e)
        {
            UpdatePosition();
        }
        private void aboutToolStripMenuItem_Click(object sender,
EventArgs e)
        {
            AboutBox1 about = new AboutBox1();
            about.ShowDialog();
        }
        private void stopTrackingToolStripMenuItem_Click(object
sender, EventArgs e)
        {
            ManageTimer();
        }
        private void tsBtnStartStopTracking_Click(object sender,
EventArgs e)
        {
            ManageTimer();
        }
        private void ManageTimer()
        {
            if (startTracking)
            {
                tmrRefreshTracking.Start();
                stopTrackingToolStripMenuItem.Text = "Stop Tracking";
                tsBtnStartStopTracking.ToolTipText = "Stop Tracking";
                tsLBLTrackingMode.Text = "Tracking : Auto Tracking
\"Start\".    | ";
                this.stopTrackingToolStripMenuItem.Image =
global::ServerMonitor.Properties.Resources.Config_Server_off;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                this.tsBtnStartStopTracking.Image =
global::ServerMonitor.Properties.Resources.Config_Server_off;
                startTracking = false;
            }
            else
            {
                tmrRefreshTracking.Stop();
                stopTrackingToolStripMenuItem.Text = "Start Tracking";
                tsBtnStartStopTracking.ToolTipText = "Start Tracking";
                tsLBLTrackingMode.Text = "Tracking : Auto Tracking
\"Stop\".    | ";
                this.stopTrackingToolStripMenuItem.Image =
global::ServerMonitor.Properties.Resources.Config_Server;
                this.tsBtnStartStopTracking.Image =
global::ServerMonitor.Properties.Resources.Config_Server;
                startTracking = true;
            }
        }
    }
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Source code ตัวtestส่วนแสดงผลลัพธ์ของพิกัด

```
using System.Data;
using System.Drawing;
using System.Text;
using System.Windows.Forms;
using System.Collections;
namespace TestGPS
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }
        private void btnSubmit_Click(object sender, EventArgs e)
        {
            TrackingService.Service objGPSTracking = new
TestGPS.TrackingService.Service();
            objGPSTracking.SetPosition(txtLatitude.Text,
txtLongitude.Text);
        }
    }
}
namespace TestGPS
{
    partial class Form1
    {
        /// <summary>
        /// Required designer variable.
        /// </summary>
        private System.ComponentModel.IContainer components = null;

        /// <summary>
        /// Clean up any resources being used.
        /// </summary>
        /// <param name="disposing">true if managed resources should
be disposed; otherwise, false.</param>
        protected override void Dispose(bool disposing)
        {
            if (disposing && (components != null))
            {
                components.Dispose();
            }
            base.Dispose(disposing);
        }

        #region Windows Form Designer generated code

        /// <summary>
        /// Required method for Designer support - do not modify
        /// the contents of this method with the code editor.
        /// </summary>
        private void InitializeComponent()
        {
            System.ComponentModel.ComponentResourceManager resources
= new System.ComponentModel.ComponentResourceManager(typeof(Form1));
            this.btnSubmit = new System.Windows.Forms.Button();
            this.txtLongitude = new System.Windows.Forms.TextBox();
            this.txtLatitude = new System.Windows.Forms.TextBox();
            this.label5 = new System.Windows.Forms.Label();
            this.label6 = new System.Windows.Forms.Label();
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            this.SuspendLayout();
            //
            // btnSubmit
            //
            this.btnSubmit.Location = new System.Drawing.Point(179,
58);
            this.btnSubmit.Name = "btnSubmit";
            this.btnSubmit.Size = new System.Drawing.Size(75, 23);
            this.btnSubmit.TabIndex = 8;
            this.btnSubmit.Text = "Submit";
            this.btnSubmit.UseVisualStyleBackColor = true;
            this.btnSubmit.Click += new
System.EventHandler(this.btnSubmit_Click);
            //
            // txtLongitude
            //
            this.txtLongitude.Location = new System.Drawing.Point(72,
6);
            this.txtLongitude.Name = "txtLongitude";
            this.txtLongitude.Size = new System.Drawing.Size(182, 20);
            this.txtLongitude.TabIndex = 0;
            //
            // txtLatitude
            //
            this.txtLatitude.Location = new System.Drawing.Point(72,
32);
            this.txtLatitude.Name = "txtLatitude";
            this.txtLatitude.Size = new System.Drawing.Size(182, 20);
            this.txtLatitude.TabIndex = 4;
            //
            // label5
            //
            this.label5.AutoSize = true;
            this.label5.Location = new System.Drawing.Point(12, 35);
            this.label5.Name = "label5";
            this.label5.Size = new System.Drawing.Size(45, 13);
            this.label5.TabIndex = 13;
            this.label5.Text = "Latitude";
            //
            // label6
            //
            this.label6.AutoSize = true;
            this.label6.Location = new System.Drawing.Point(12, 9);
            this.label6.Name = "label6";
            this.label6.Size = new System.Drawing.Size(54, 13);
            this.label6.TabIndex = 14;
            this.label6.Text = "Longitude";
            //
            // Form1
            //
            this.AutoScaleMode =
System.Windows.Forms.AutoScaleMode.None;
            this.ClientSize = new System.Drawing.Size(275, 85);
            this.Controls.Add(this.label6);
            this.Controls.Add(this.label5);
            this.Controls.Add(this.txtLatitude);
            this.Controls.Add(this.txtLongitude);
            this.Controls.Add(this.btnSubmit);
            this.Icon =
((System.Drawing.Icon)(resources.GetObject("$this.Icon")));
            this.Name = "Form1";
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            this.Text = "Test Latitude and Longitude";
            this.ResumeLayout(false);
            this.PerformLayout();

        }

        #endregion

        private System.Windows.Forms.Button btnSubmit;
        private System.Windows.Forms.TextBox txtLongitude;
        private System.Windows.Forms.TextBox txtLatitude;
        private System.Windows.Forms.Label label5;
        private System.Windows.Forms.Label label6;
    }
}
```


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Application.config

```
<?xml version="1.0" encoding="utf-8" ?>
<configuration>
    <configSections>
        <sectionGroup name="applicationSettings"
type="System.Configuration.ApplicationSettingsGroup, System,
Version=2.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089" >
            <section name="TestGPS.Properties.Settings"
type="System.Configuration.ClientSettingsSection, System,
Version=2.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
requirePermission="false" />
        </sectionGroup>
    </configSections>
    <applicationSettings>
        <TestGPS.Properties.Settings>
            <setting name="TestGPS_TrackingService_Service"
serializeAs="String">

<value>http://www.Icantaloup.com/GPSTracking/Service.asmx</value>
            </setting>
        </TestGPS.Properties.Settings>
    </applicationSettings>
</configuration>
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Tracking Manager

การสร้างFile xml

```
using System;
using System.Collections.Generic;
using System.Text;
using System.Xml;
namespace TrackingManager
{
    public class CreateXML
    {
        XmlTextWriter writer;
      public void CreateXmlDocument(string xmlPathFileName, string
alLatitude, string alLongitude)
        {
            //string xmlFilename = "Test.xml";
            writer = new XmlTextWriter(xmlPathFileName,
System.Text.Encoding.UTF8);
            // Indent the XML document for readability
            writer.Formatting = System.Xml.Formatting.Indented;
            // Call WriteStartDocument to write xml declaration
            // should be the first call before writing elements
            writer.WriteStartDocument();
            // write a comment
            writer.WriteComment("This is position from mobile");
            // write root element - should atleast contain a root
      element
            //writer.WriteStartElement("Article");
            writer.WriteStartElement("Tracking");
            WritePosition(alLongitude, alLatitude);
            // Write End of root element
            writer.WriteEndElement();
            // Write End of document
            writer.WriteEndDocument();

            writer.Close();

    private void WritePosition(string alLongitude, string alLatitude)
        {
            // Write Element <Header>
            writer.WriteStartElement("Track");
            // Write the Title Attribute
            writer.WriteAttributeString("Position", "Keep Latitude");
            writer.WriteElementString("Longitude", alLongitude);
            writer.WriteElementString("Latitude", alLatitude);
            // Write End Element </Header>
            writer.WriteEndElement();
        }
    }
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Source code serviceSoap_stub.java

```
This class was generated by 172 StubGenerator.
// Contents subject to change without notice.
// @generated
package service;
import javax.xml.rpc.JAXRPCException;
import javax.xml.namespace.QName;
import javax.microedition.xml.rpc.Operation;
import javax.microedition.xml.rpc.Type;
import javax.microedition.xml.rpc.ComplexType;
import javax.microedition.xml.rpc.Element;
public class ServiceSoap_Stub implements service.ServiceSoap,
javax.xml.rpc.Stub {
      private String[] _propertyNames;
      private Object[] _propertyValues;

      public ServiceSoap_Stub() {
            _propertyNames = new String[] {ENDPOINT_ADDRESS_PROPERTY};
             _propertyValues = new Object[]
{"http://www.icantaloup.com/GPSTracking/service.asmx"};
      }
      public void _setProperty(String name, Object value) {
            int size = _propertyNames.length;
            for (int i = 0; i < size; ++i) {
                  if (_propertyNames[i].equals(name)) {
                        _propertyValues[i] = value;
                        return;
                  }
            }
            // Need to expand our array for a new property
            String[] newPropNames = new String[size + 1];
            System.arraycopy(_propertyNames, 0, newPropNames, 0,
size);
             _propertyNames = newPropNames;
            Object[] newPropValues = new Object[size + 1];
            System.arraycopy(_propertyValues, 0, newPropValues, 0,
size);
            _propertyValues = newPropValues;
            _propertyNames[size] = name;
            _propertyValues[size] = value;
      }
      public Object _getProperty(String name) {
            for (int i = 0; i < _propertyNames.length; ++i) {
                  if (_propertyNames[i].equals(name)) {
                        return _propertyValues[i];
                  }
            }
            if (ENDPOINT_ADDRESS_PROPERTY.equals(name) ||
USERNAME_PROPERTY.equals(name) || PASSWORD_PROPERTY.equals(name)) {
                  return null;
            }
            if (SESSION_MAINTAIN_PROPERTY.equals(name)) {
                  return new java.lang.Boolean(false);
            }
            throw new JAXRPCException("Stub does not recognize
property: "+name);
      }
      protected void _prepOperation(Operation op) {
            for (int i = 0; i < _propertyNames.length; ++i) {
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                    op.setProperty(_propertyNames[i],
_propertyValues[i].toString());
                }
        }
        //
        //  Begin user methods
        //
        public service.SetPositionResponse setPosition(java.lang.String
alLatitude, java.lang.String alLongitude) throws
java.rmi.RemoteException {
                // Copy the incoming values into an Object array if
needed.
                Object[] inputObject = new Object[2];
                inputObject[0] = alLatitude;
                inputObject[1] = alLongitude;
                Operation op = Operation.newInstance(_qname_SetPosition,
        _type_SetPosition, _type_SetPositionResponse);
                _prepOperation(op);
                op.setProperty(Operation.SOAPACTION_URI_PROPERTY,
"http://tempuri.org/SetPosition");
                Object resultObj;
                try {
                        resultObj = op.invoke(inputObject);
                } catch (JAXRPCException e) {
                        Throwable cause = e.getLinkedCause();
                        if (cause instanceof java.rmi.RemoteException) {
                                throw (java.rmi.RemoteException) cause;
                        }
                        throw e;
                }
                service.SetPositionResponse result;
                // Convert the result into the right Java type.
                if (resultObj == null) {
                        result = null;
                } else {
                        result = new service.SetPositionResponse();
                }
                return result;
        }
        public double getLatitude() throws java.rmi.RemoteException {
                // Copy the incoming values into an Object array if
needed.
                Object[] inputObject = new Object[0];

                Operation op = Operation.newInstance(_qname_GetLatitude,
_type_GetLatitude, _type_GetLatitudeResponse);
                _prepOperation(op);
                op.setProperty(Operation.SOAPACTION_URI_PROPERTY,
"http://tempuri.org/GetLatitude");
                Object resultObj;
                try {
                        resultObj = op.invoke(inputObject);
                } catch (JAXRPCException e) {
                        Throwable cause = e.getLinkedCause();
                        if (cause instanceof java.rmi.RemoteException) {
                                throw (java.rmi.RemoteException) cause;
                        }
                        throw e;
                }
                double result;
                // Convert the result into the right Java type.
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            // Unwrapped return value
            Object getLatitudeResultObj = ((Object[])resultObj)[0];
            result =
((java.lang.Double)getLatitudeResultObj).doubleValue();
            return result;
      }
            public double getLongitude() throws
      java.rmi.RemoteException {
            // Copy the incoming values into an Object array if
needed.
            Object[] inputObject = new Object[0];

            Operation op = Operation.newInstance(_qname_GetLongitude,
_type_GetLongitude, _type_GetLongitudeResponse);
            _prepOperation(op);
            op.setProperty(Operation.SOAPACTION_URI_PROPERTY,
"http://tempuri.org/GetLongitude");
            Object resultObj;
            try {
                  resultObj = op.invoke(inputObject);
            } catch (JAXRPCException e) {
                  Throwable cause = e.getLinkedCause();
                  if (cause instanceof java.rmi.RemoteException) {
                        throw (java.rmi.RemoteException) cause;
                  }
                  throw e;
            }
            double result;
            // Convert the result into the right Java type.
            // Unwrapped return value
            Object getLongitudeResultObj = ((Object[])resultObj)[0];
            result =
((java.lang.Double)getLongitudeResultObj).doubleValue();
            return result;
      }
      //
      //  End user methods
      //

protected static final QName _qname_GetLatitude = new
QName("http://tempuri.org/", "GetLatitude");
protected static final QName _qname_GetLatitudeResponse = new
QName("http://tempuri.org/", "GetLatitudeResponse");
protected static final QName _qname_GetLatitudeResult = new
QName("http://tempuri.org/", "GetLatitudeResult");
protected static final QName _qname_GetLongitude = new
QName("http://tempuri.org/", "GetLongitude");
protected static final QName _qname_GetLongitudeResponse = new
QName("http://tempuri.org/", "GetLongitudeResponse");
protected static final QName _qname_GetLongitudeResult = new
QName("http://tempuri.org/", "GetLongitudeResult");
protected static final QName _qname_SetPosition = new
QName("http://tempuri.org/", "SetPosition");
protected static final QName _qname_SetPositionResponse = new
QName("http://tempuri.org/", "SetPositionResponse");
protected static final QName _qname_alLatitude = new
QName("http://tempuri.org/", "alLatitude");
protected static final QName _qname_alLongitude = new
QName("http://tempuri.org/", "alLongitude");
protected static final Element _type_SetPosition;
protected static final Element _type_SetPositionResponse;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
protected static final Element _type_GetLatitude;
protected static final Element _type_GetLatitudeResponse;
protected static final Element _type_GetLongitude;
protected static final Element _type_GetLongitudeResponse;
      static {
            // Create all of the Type's that this stub uses, once.
            Element _type_alLatitude;
            _type_alLatitude = new Element(_qname_alLatitude,
Type.STRING, 0, 1, false);
            Element _type_alLongitude;
            _type_alLongitude = new Element(_qname_alLongitude,
Type.STRING, 0, 1, false);
            ComplexType _complexType_setPosition;
            _complexType_setPosition = new ComplexType();
            _complexType_setPosition.elements = new Element[2];
            _complexType_setPosition.elements[0] = _type_alLatitude;
            _complexType_setPosition.elements[1] = _type_alLongitude;
            _type_SetPosition = new Element(_qname_SetPosition,
_complexType_setPosition);
            ComplexType _complexType_setPositionResponse;
            _complexType_setPositionResponse = new ComplexType();
            _complexType_setPositionResponse.elements = new
Element[0];
            _type_SetPositionResponse = new
Element(_qname_SetPositionResponse, _complexType_setPositionResponse);
            _type_GetLatitude = new Element(_qname_GetLatitude,
_complexType_setPositionResponse);
            Element _type_GetLatitudeResult;
            _type_GetLatitudeResult = new
Element(_qname_GetLatitudeResult, Type.DOUBLE);
            ComplexType _complexType_getLatitudeResponse;
            _complexType_getLatitudeResponse = new ComplexType();
            _complexType_getLatitudeResponse.elements = new
Element[1];
            _complexType_getLatitudeResponse.elements[0] =
_type_GetLatitudeResult;
            _type_GetLatitudeResponse = new
Element(_qname_GetLatitudeResponse, _complexType_getLatitudeResponse);
            _type_GetLongitude = new Element(_qname_GetLongitude,
_complexType_setPositionResponse);
            Element _type_GetLongitudeResult;
            _type_GetLongitudeResult = new
Element(_qname_GetLongitudeResult, Type.DOUBLE);
            ComplexType _complexType_getLongitudeResponse;
            _complexType_getLongitudeResponse = new ComplexType();
            _complexType_getLongitudeResponse.elements = new
Element[1];
            _complexType_getLongitudeResponse.elements[0] =
_type_GetLongitudeResult;
            _type_GetLongitudeResponse = new
Element(_qname_GetLongitudeResponse,
_complexType_getLongitudeResponse);
      }

}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Service.asmx.wsdl

```
<?xml version="1.0" encoding="utf-8"?>
<wsdl:definitions xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
xmlns:tm="http://microsoft.com/wsdl/mime/textMatching/"
xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:mime="http://schemas.xmlsoap.org/wsdl/mime/"
xmlns:tns="http://tempuri.org/"
xmlns:s="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://schemas.xmlsoap.org/wsdl/soap12/"
xmlns:http="http://schemas.xmlsoap.org/wsdl/http/"
targetNamespace="http://tempuri.org/"
xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
<wsdl:types>
<s:schema elementFormDefault="qualified"
targetNamespace="http://tempuri.org/">
<s:element name="SetPosition">
<s:complexType>
<s:sequence>
<s:element minOccurs="0" maxOccurs="1" name="alLatitude"
type="s:string" />
<s:element minOccurs="0" maxOccurs="1" name="alLongitude"
type="s:string" />
</s:sequence>
</s:complexType>
</s:element>
<s:element name="SetPositionResponse">
<s:complexType />
</s:element>
<s:element name="GetLatitude">
<s:complexType />
</s:element>
<s:element name="GetLatitudeResponse">
<s:complexType>
<s:sequence>
<s:element minOccurs="1" maxOccurs="1" name="GetLatitudeResult"
type="s:double" />
</s:sequence>
</s:complexType>
</s:element>
<s:element name="GetLongitude">
<s:complexType />
</s:element>
<s:element name="GetLongitudeResponse">
<s:complexType>
<s:sequence>
<s:element minOccurs="1" maxOccurs="1" name="GetLongitudeResult"
type="s:double" />
</s:sequence>
</s:complexType>
</s:element>
</s:schema>
</wsdl:types>
<wsdl:message name="SetPositionSoapIn">
<wsdl:part name="parameters" element="tns:SetPosition" />
</wsdl:message>
<wsdl:message name="SetPositionSoapOut">
<wsdl:part name="parameters" element="tns:SetPositionResponse" />
</wsdl:message>
<wsdl:message name="GetLatitudeSoapIn">
<wsdl:part name="parameters" element="tns:GetLatitude" />
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
</wsdl:message>
<wsdl:message name="GetLatitudeSoapOut">
<wsdl:part name="parameters" element="tns:GetLatitudeResponse" />
</wsdl:message>
<wsdl:message name="GetLongitudeSoapIn">
<wsdl:part name="parameters" element="tns:GetLongitude" />
</wsdl:message>
<wsdl:message name="GetLongitudeSoapOut">
<wsdl:part name="parameters" element="tns:GetLongitudeResponse" />
</wsdl:message>
<wsdl:portType name="ServiceSoap">
<wsdl:operation name="SetPosition">
<wsdl:input message="tns:SetPositionSoapIn" />
<wsdl:output message="tns:SetPositionSoapOut" />
</wsdl:operation>
<wsdl:operation name="GetLatitude">
<wsdl:input message="tns:GetLatitudeSoapIn" />
<wsdl:output message="tns:GetLatitudeSoapOut" />
</wsdl:operation>
<wsdl:operation name="GetLongitude">
<wsdl:input message="tns:GetLongitudeSoapIn" />
<wsdl:output message="tns:GetLongitudeSoapOut" />
</wsdl:operation>
</wsdl:portType>
<wsdl:binding name="ServiceSoap" type="tns:ServiceSoap">
<soap:binding transport="http://schemas.xmlsoap.org/soap/http" />
<wsdl:operation name="SetPosition">
<soap:operation soapAction="http://tempuri.org/SetPosition"
style="document" />
<wsdl:input>
<soap:body use="literal" />
</wsdl:input>
<wsdl:output>
<soap:body use="literal" />
</wsdl:output>
</wsdl:operation>
<wsdl:operation name="GetLatitude">
<soap:operation soapAction="http://tempuri.org/GetLatitude"
style="document" />
<wsdl:input>
<soap:body use="literal" />
</wsdl:input>
<wsdl:output>
<soap:body use="literal" />
</wsdl:output>
</wsdl:operation>
<wsdl:operation name="GetLongitude">
<soap:operation soapAction="http://tempuri.org/GetLongitude"
style="document" />
<wsdl:input>
<soap:body use="literal" />
</wsdl:input>
<wsdl:output>
<soap:body use="literal" />
</wsdl:output>
</wsdl:operation>
</wsdl:binding>
<wsdl:binding name="ServiceSoap12" type="tns:ServiceSoap">
<soap12:binding transport="http://schemas.xmlsoap.org/soap/http" />
<wsdl:operation name="SetPosition">
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
<soap12:operation soapAction="http://tempuri.org/SetPosition"
style="document" />
<wsdl:input>
<soap12:body use="literal" />
</wsdl:input>
<wsdl:output>
<soap12:body use="literal" />
</wsdl:output>
</wsdl:operation>
<wsdl:operation name="GetLatitude">
<soap12:operation soapAction="http://tempuri.org/GetLatitude"
style="document" />
<wsdl:input>
<soap12:body use="literal" />
</wsdl:input>
<wsdl:output>
<soap12:body use="literal" />
</wsdl:output>
</wsdl:operation>
<wsdl:operation name="GetLongitude">
<soap12:operation soapAction="http://tempuri.org/GetLongitude"
style="document" />
<wsdl:input>
<soap12:body use="literal" />
</wsdl:input>
<wsdl:output>
<soap12:body use="literal" />
</wsdl:output>
</wsdl:operation>
</wsdl:binding>
<wsdl:service name="Service">
<wsdl:port name="ServiceSoap" binding="tns:ServiceSoap">
<soap:address
location="http://www.icantaloup.com/GPSTracking/service.asmx" />
</wsdl:port>
<wsdl:port name="ServiceSoap12" binding="tns:ServiceSoap12">
<soap12:address
location="http://www.icantaloup.com/GPSTracking/service.asmx" />
</wsdl:port>
</wsdl:service>
</wsdl:definitions>
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้