สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

การพัฒนาโปรแกรมสำหรับการปรับตำแหน่งจานดาวเทียมติดรถยนต์

THE DEVELOPMENT OF POSITION CONTROL PROGRAM FOR MOBILE SATELLITE DISH

T119450

นางสาวณัฐชยา แย้มนาก
นายธัญกร ศรีสิงห์
นางสาวเสาวลักษณ์ โกศลนิรัติวงษ์

เลขหมู่ 119450
เลขทะเบียน
วัน,เดือน,ปี - 7 S.A. 2554

b
i

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิชาวิศวกรรมการวัดคุม
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2553

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

THE DEVELOPMENT OF POSITION CONTROL PROGRAM FOR MOBILE SATELLITE DISH

NATCHAYA YAMNAK

THUNYAKORN SRISING

SAOWALAK KOSOLNIRATTIWONG

A THESIS SUBMITTED IN PARTIAL FULFILLMENT

OF THE REQUIREMENT FOR THE DEGREE OF

BACHELOR OF ENGINEERING IN INSTRUMENTATION ENGINEERING

FACULTY OF ENGINEERING

KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

ACADEMIC YEAR 2010

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สาขาวิชาวิศวกรรมการวัดคุม

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ใบรับรองปริญญานิพนธ์

.....

หัวข้อปริญญานิพนธ์ การพัฒนาโปรแกรมสำหรับการปรับตำแหน่งจานดาวเทียมติดรถยนต์

THE DEVELOPMENT OF POSITION CONTROL PROGRAM FOR MOBILE SATELLITE DISH

นักศึกษาผู้จัดทำ นางสาวณัฐชยา แย้มนาก รหัสนักศึกษา 50010444

นายธัญกร ศรีสิงห์ รหัสนักศึกษา 50010671

นางสาวเสาวลักษณ์ โกศลนิรัติวงษ์ รหัสนักศึกษา 50011794

ปริญญา วิศวกรรมศาสตรบัณฑิต

สาขาวิชา วิศวกรรมการวัดคุม

ปีการศึกษา 2553

อาจารย์ผู้ควบคุมปริญญานิพนธ์	ลายมือชื่อ
รศ.วิศรุต ศรีรัตนะ อาจารย์สุธรรม สัทธรรมสกุล	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อปริญญานิพนธ์ การพัฒนาโปรแกรมสำหรับการปรับตำแหน่งจานดาวเทียมติดรถยนต์
THE DEVELOPMENT OF POSITION CONTROL PROGRAM FOR MOBILE SATELLITE DISH

นักศึกษาผู้จัดทำ นางสาวณัฐชยา แย้มนาก รหัสนักศึกษา 50010444
นายธัญกร ศรีสิงห์ รหัสนักศึกษา 50010671
นางสาวเสาวลักษณ์ โกศลนิรัติวงษ์ รหัสนักศึกษา 50011794

อาจารย์ที่ปรึกษา รศ.วิศรุต ศรีรัตนะ
อาจารย์ สุธรรม สัทธรรมสกุล

ปีการศึกษา 2553

บทคัดย่อ

โครงงานนี้เป็นการนำเสนอการให้บริการอินเตอร์เน็ตผ่านดาวเทียม (IPSTAR) ในลักษณะของรถ MVV (Mobile VSAT Vehicle) ซึ่งเป็นรถที่ได้รับการติดตั้งชุดอุปกรณ์ดาวเทียม IPSTAR แบบเต็มรูปในลักษณะพร้อมใช้งานได้ทันที สามารถที่จะเคลื่อนย้ายไปตามสถานที่ต่างๆ ได้ตามต้องการ ซึ่งเหมาะกับงานด้านการถ่ายทอดสด หรืองานภาคสนามต่างๆ และสามารถใช้เป็นตัวเชื่อมต่อสื่อการเรียนการสอน และการเรียนรู้ สำหรับชุมชนที่อยู่ในชนบทหรือพื้นที่ที่อยู่ห่างไกลทั่วประเทศ โดยจุดประสงค์หลักของโครงงานนี้ จะเป็นการพัฒนาในส่วนของโปรแกรมสำหรับการติดต่อกับผู้ใช้งานด้วยภาษา Visual C# เพื่อให้ง่ายและสะดวกต่อการใช้งานมากขึ้น และจะทำการแก้ไขโปรแกรมควบคุมการหมุนของแกน Polarization จากเดิมที่องศาในการหมุนไม่ถูกต้อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Thesis Title	The development of position control program for mobile satellite dish	
Authors	Miss Natchaya	Yamnak
	Mr. Thunyakorn	Srising
	Miss Saowalak	Kosolnirattiwong
Thesis Advisor	Assoc.Prof. Witsarut	Srira tana
	Mr. Sutham	Satthamsakul
Year	2010	

ABSTRACT

This project has been brought the technology of internet services via a satellite (IPSTAR) system with MVV car (Mobile VSAT Vehicle). The car is installed IPSTAR satellite kit, and it can move to different suitable locations for broadcasting. This system can also use for long distance learning. So the country people can improve their life. The objective of this project is developed a new program portion by visual C#. This new program is conveniently and friendly to the user. Also, it can be precisely controlled the rotation polarize angle.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

ปริญญานิพนธ์ฉบับนี้สำเร็จลุล่วงได้ด้วยดี ซึ่งได้รับความกรุณาจากอาจารย์วิศรุต ศรีรัตนะ และอาจารย์สุธรรม สัทธรรมสกุล อาจารย์ที่ปรึกษา ที่ได้ให้ความกรุณาให้ความอนุเคราะห์อุปกรณ์ที่ใช้สำหรับทดลองในปริญญานิพนธ์ และให้คำแนะนำ คำปรึกษา แนวคิดต่างๆในการปรับปรุงแก้ไขงาน และการทดลอง ตลอดจนอาจารย์ทุกท่าน ที่ได้อบรมให้ความรู้ ทำให้สามารถนำความรู้ที่ได้มาใช้ในการทำปริญญานิพนธ์ฉบับนี้ ซึ่งผู้จัดทำขอกราบขอบพระคุณเป็นอย่างสูงมา ณ โอกาสนี้ด้วย

คณะผู้จัดทำ

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

หน้า

บทคัดย่อภาษาไทย.....I

บทคัดย่อภาษาอังกฤษ.....II

กิตติกรรมประกาศ.....III

สารบัญ.....IV

สารบัญตาราง.....VIII

สารบัญภาพ.....IX

บทที่ 1 บทนำ.....1

1.1 ความสำคัญของปริญญานิพนธ์.....1

1.2 วัตถุประสงค์ของปริญญานิพนธ์.....2

1.3 ขอบเขตของปริญญานิพนธ์.....2

1.4 ขั้นตอนการศึกษา.....2

1.5 ประโยชน์ที่คาดว่าจะได้รับ.....3

บทที่ 2 ทฤษฎีและหลักการที่เกี่ยวข้อง.....4

2.1 องค์ประกอบของดาวเทียมไอพีสตาร์(IPSTAR).....4

2.1.1 ระบบดาวเทียม.....4

2.1.2 ระบบภาคพื้นดินของไอพีสตาร์.....5

2.1.2.1 อุปกรณ์ปลายทางและกระบวนการส่งสัญญาณ.....5

2.1.2.2 ระบบควบคุมเครือข่ายหรือเกตเวย์.....6

2.1.2.3 การให้บริการของดาวเทียมไอพีสตาร์.....6

2.1.2.4 การนำไอพีสตาร์มาใช้กับบรอดแบนด์อินเตอร์เน็ต.....7

2.2 ระบบพิกัดบนพื้นโลกจีพีเอส.....7

2.2.1 ส่วนประกอบของระบบจีพีเอส.....8

2.2.1.1 ส่วนอวกาศ (Space Segment).....9

2.2.1.2 ส่วนสถานีควบคุม (Control Station Segment).....10

2.2.1.3 ส่วนของผู้ใช้งาน (User Segment).....11

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

หน้า

2.2.2 หลักการทำงานของจีพีเอส.....13

2.2.2.1 การให้บริการจีพีเอส (GPS Service).....13

2.2.2.2 การทำงานของเครื่องรับจีพีเอส.....13

2.2.2.3 การรับสัญญาณดาวเทียม (Satellite Signal Acquisition) 14

2.2.3 ย่านความถี่ในการส่งสัญญาณ.....15

2.2.4 ระบบการส่งสัญญาณ.....15

2.2.4.1 การส่งสัญญาณแบบ C-BAND.....15

2.2.4.2 การส่งสัญญาณแบบ KU-BAND.....16

2.2.5 ตำแหน่งดาวเทียมและพื้นที่ติดตั้ง.....16

2.2.5.1 ตำแหน่งของดาวเทียม.....16

2.2.5.2 การกำหนดพื้นที่ในการติดตั้ง.....17

2.2.5.3 การคำนวณมุมก้มเงยและมุมกวาด.....17

2.3 ไมโครคอนโทรลเลอร์ PIC 16F873A.....18

2.3.1 โครงสร้างภายในของไมโครคอนโทรลเลอร์.....18

2.3.1.1 คุณสมบัติหลัก.....18

2.3.1.2 คุณสมบัติพิเศษเพิ่มเติม.....19

2.3.2 โครงสร้างภายนอกของไมโครคอนโทรลเลอร์.....21

2.3.2.1 หน้าที่และการใช้งานของสัญญาณต่างๆ.....21

2.3.2.2 การจัดการหน่วยความจำของ PIC 16F873A.....23

2.4 มอเตอร์กระแสตรง (DC MOTOR).....25

2.4.1 ส่วนประกอบของมอเตอร์ไฟฟ้ากระแสตรง.....26

2.4.1.1 ส่วนประกอบหลักของมอเตอร์ไฟฟ้ากระแสตรง.....26

2.4.2 การทำงานของมอเตอร์ไฟฟ้า.....27

2.4.3 การขับและกลับทิศทางของมอเตอร์กระแสตรง.....28

2.4.4 การควบคุมความเร็วของมอเตอร์กระแสตรง.....30

2.4.4.1 วิธีการมอดูเลชั่นทางความกว้างของพัลส์ (PWM).....30

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

หน้า

บทที่ 3 วิธีการดำเนินงาน.....32

3.1 โครงสร้าง Mobile VSAT Vehicle (MVV).....32

3.1.1 บล็อกไดอะแกรมแสดงโครงสร้างการเชื่อมต่ออุปกรณ์.....32

3.1.2 องค์ประกอบของ MVV.....33

3.1.2.1 ส่วนประกอบของจานดาวเทียม.....33

3.1.2.2 ส่วนประกอบของ Azimuth Part.....34

3.1.2.3 ส่วนประกอบของ Elevation Part.....34

3.1.2.4 ส่วนประกอบของ Polarization Part.....34

3.1.2.5 ส่วนประกอบของ Controller.....35

3.2 การศึกษาโปรแกรมสำหรับการปรับตำแหน่งจานดาวเทียมติดรถยนต์ที่เขียนด้วยวิชวลเบสิก 6 (Visual Basic 6).....35

3.2.1 ลักษณะการสร้างโปรแกรมใช้งานด้วยวิชวลเบสิก.....36

3.2.2 ชนิดข้อมูลของโปรแกรมวิชวลเบสิก.....36

3.2.3 ส่วนที่ติดต่อกับผู้ใช้งาน.....38

3.2.3.1 ส่วนของ Signal Status.....38

3.2.3.2 ส่วนของ Antenna Details.....39

3.2.3.3 ส่วนของ Dish Position.....39

3.3 การพัฒนาโปรแกรมสำหรับการปรับตำแหน่งจานดาวเทียมติดรถยนต์ด้วย Visual C# 40

3.3.1 การเขียนโปรแกรมแบบ Event Driven Programming.....40

3.3.2 ออบเจ็กต์ (Object).....40

3.3.3 พร็อพเพอร์ตี้ (Property).....41

3.3.4 คอนโทรลกับคอมโพเนนต์.....41

3.3.5 การกำหนดค่าใน Property Window.....41

3.3.5.1 การกำหนดค่าอย่างอิสระ.....41

3.3.5.2 การกำหนดค่าจากรายการตัวเลือก.....41

3.3.5.3 การกำหนดค่าจากไดอะล็อกบ็อกซ์.....41

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

หน้า

3.3.6 การออกแบบส่วนติดต่อกับผู้ใช้งาน.....42

3.3.6.1 Basic Mode.....42

3.3.6.2 Auto Mode.....43

3.3.6.3 Manual Mode.....44

3.3.6.4 Communication Mode.....45

3.3.6.5 Parameter.....45

3.4 การทำงานของไมโครคอนโทรลเลอร์ PIC16F873A.....46

3.4.1 การตรวจสอบคำสั่ง.....47

3.4.2 การทำงานของกระบวนการหลัก.....48

บทที่ 4 ผลการทดลอง.....49

4.1 การเคลื่อนที่แนวแกนของมุมก้มเงย (Elevation).....49

4.1.1 ลำดับขั้นตอนในการทำงาน.....49

4.1.2 ผลการทดลองก่อนการแก้ไข.....50

4.1.3 ผลการทดลองหลังการแก้ไข.....51

4.2 การเคลื่อนที่แนวแกนของการหมุนขั้วคลื่น (Polarization).....53

4.2.1 ลำดับขั้นตอนในการทำงาน.....53

4.2.2 ผลการทดลองก่อนการแก้ไข.....54

4.2.3 ผลการทดลองหลังการแก้ไข.....55

บทที่ 5 สรุปผลการวิจัยและข้อเสนอแนะ.....57

5.1 สรุปผล.....57

5.2 ปัญหาและแนวทางแก้ไข.....57

5.3 ข้อเสนอแนะ.....57

บรรณานุกรม.....59

ภาคผนวก.....60

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

ตารางที่ **หน้า**

2.1 รายละเอียดดาวเทียมไทยคม 4 (ไอพีสตาร์).....5
2.2 หน้าที่อื่นๆ ของพอร์ต A.....22
2.3 หน้าที่อื่นๆ ของพอร์ต B และ C.....23
3.1 ชนิดข้อมูลในวิชวลเบสิก.....37
4.1 แสดงค่าเฉลี่ยการหมุนแนวแกน Elevation ก่อนการแก้ไขโดยเพิ่ม/ลด ทีละ 10 องศา.....49
4.2 แสดงค่าเฉลี่ยการหมุนแนวแกน Elevation ก่อนการแก้ไขโดยเพิ่ม/ลด ทีละ 20 องศา 49
4.3 แสดงค่าเฉลี่ยการหมุนแนวแกน Elevation ก่อนการแก้ไขโดยเพิ่ม/ลด ทีละ 30 องศา.....50
4.4 แสดงค่าเฉลี่ยการหมุนแนวแกน Elevation หลังการแก้ไขโดยเพิ่ม/ลด ทีละ 10 องศา.....50
4.5 แสดงค่าเฉลี่ยการหมุนแนวแกน Elevation หลังการแก้ไขโดยเพิ่ม/ลด ทีละ 20 องศา 51
4.6 แสดงค่าเฉลี่ยการหมุนแนวแกน Elevation หลังการแก้ไขโดยเพิ่ม/ลด ทีละ 30 องศา.....51
4.7 แสดงค่าเฉลี่ยการหมุนแนวแกน Polarization ก่อนการแก้ไขโดยเพิ่ม/ลด ทีละ 10 องศา.....53
4.8 แสดงค่าเฉลี่ยการหมุนแนวแกน Polarization ก่อนการแก้ไขโดยเพิ่ม/ลด ทีละ 20 องศา.....53
4.9 แสดงค่าเฉลี่ยการหมุนแนวแกน Polarization ก่อนการแก้ไขโดยเพิ่ม/ลด ทีละ 30 องศา.....54
4.10 แสดงค่าเฉลี่ยการหมุนแนวแกน Polarization หลังการแก้ไขโดยเพิ่ม/ลด ทีละ 10 องศา.....54
4.11 แสดงค่าเฉลี่ยการหมุนแนวแกน Polarization หลังการแก้ไขโดยเพิ่ม/ลด ทีละ 20 องศา.....55
4.12 แสดงค่าเฉลี่ยการหมุนแนวแกน Polarization หลังการแก้ไขโดยเพิ่ม/ลด ทีละ 30 องศา.....55

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญภาพ

ภาพที่ **หน้า**

2.1 ดาวเทียมไอพีสตาร์.....4

2.2 การให้บริการของดาวเทียมไอพีสตาร์.....6

2.3 แสดงองค์ประกอบของระบบดาวเทียม GPS.....9

2.4 แสดงการโคจรของดาวเทียม GPS รอบโลก.....10

2.5 แสดงที่ตั้งของสถานีควบคุมดาวเทียม GPS 5 แห่ง.....11

2.6 แสดงBlock Diagram อย่างง่ายของเครื่องรับจีพีเอส.....14

2.7 การส่งและรับสัญญาณดาวเทียม.....15

2.8 แสดงพื้นที่การให้บริการของดาวเทียม.....16

2.9 แสดงตำแหน่งของประเทศต่างๆ ตามแนวเส้นรุ้งเส้นแวง.....17

2.10 โครงสร้างภายในของไมโครคอนโทรลเลอร์ PIC16F873A.....20

2.11 แสดงขาสัญญาณของไมโครคอนโทรลเลอร์ PIC16F873A.....21

2.12 การจัดพื้นที่หน่วยความจำใน PIC16F873A.....24

2.13 แสดงโครงสร้างทั่วไปของมอเตอร์กระแสตรง.....25

2.14 แสดงวงแหวนคอมมิวเตเตอร์และแปรงถ่าน.....27

2.15 แสดงการทำงานของมอเตอร์.....27

2.16 แสดงการทำงานของมอเตอร์.....28

2.17 ทิศทางการเคลื่อนที่ของอาร์เมเจอร์ (โรเตอร์).....28

2.18 แสดงการกลับทิศทางของมอเตอร์กระแสตรงโดยใช้รีเลย์.....29

2.19 แสดงการใช้ทรานซิสเตอร์เพื่อขับรีเลย์ให้ทำงาน.....29

2.20 แสดงการใช้ทรานซิสเตอร์เป็นวงจรขับและกำหนดทิศทางของมอเตอร์กระแสตรง.....30

2.21 แสดงความกว้างของพัลท์ขนาดต่างๆ และค่าดิวตี้ไซเคิลของช่วงพัลท์ที่มีความถี่คงที่.....31

3.1 บล็อกไดอะแกรมแสดงโครงสร้าง.....32

3.2 MVV Vehicle (Toyota Fortuner).....33

3.3 Antenna.....33

3.4 Antenna Platform.....33

3.5 Antenna Dish / Feed Support.....33

3.6 Opto Switch.....34

3.7 Limit Switch.....34

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญภาพ (ต่อ)

ภาพที่ **หน้า**

3.8 Elevation Motor.....34

3.9 Elevation Gear.....34

3.10 Limit Switch.....34

3.11 Polarization Motor.....34

3.12 controller.....35

3.13 Enterprise Series.....35

3.14 ส่วนติดต่อกับผู้ใช้งานเดิม Visual Basic.....38

3.15 Basic Mode.....42

3.16 Auto Mode.....43

3.17 Manual Mode.....44

3.18 Communication Port.....44

3.19 Opto Position Parameter.....45

3.20 Flow Chart แสดงการทำงานของไมโครคอนโทรลเลอร์.....45

3.21 Flow Chart แสดงการตรวจสอบของคำสั่งที่ได้รับ.....46

3.22 Flow Chart แสดงการทำงานของกระบวนการหลัก.....47

4.1 แสดงการเคลื่อนที่ของแกน Elevation ไปที่ 30 องศา.....48

4.1 แสดงการเคลื่อนที่ของแกน Elevation ไปที่ 90 องศา.....48

4.3 แสดงการเคลื่อนที่ของแกน Polarization ไปที่ 30 องศา.....52

4.4 แสดงการเคลื่อนที่ของแกน Polarization ไปที่ 60 องศา.....52

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

1.1 ความสำคัญของปริญญานิพนธ์

ในยุคปัจจุบันอาจกล่าวได้ว่าเป็นยุคแห่งเทคโนโลยี ซึ่งมีการพัฒนาในส่วนของระบบสื่อสารโทรคมนาคม และเทคโนโลยีต่างๆขึ้นมามากมายหลายรูปแบบ รวมถึงเทคโนโลยีการสื่อสารผ่านดาวเทียมได้เข้ามามีบทบาทสำคัญต่อการพัฒนาประเทศในหลายๆ ด้านทั้งทางตรง และทางอ้อม อาทิเช่น การสื่อสารข้อมูลผ่านดาวเทียม การศึกษาทางไกลผ่านดาวเทียมเพื่อยกระดับมาตรฐานการศึกษาของชาติ การสำรวจทรัพยากรธรรมชาติและพยากรณ์ภูมิอากาศจากภาพถ่ายดาวเทียม การการพัฒนาทางด้านการแพทย์ การพัฒนาด้านเศรษฐกิจ การจัดการด้านสังคมและชุมชน รวมถึงการทดลองและทดสอบทางด้านวิทยาศาสตร์โดยนำอุปกรณ์การวัดติดตั้งบนตัวดาวเทียม เป็นต้น สำหรับ Mobile VSAT Vehicle (MVV) นั้นเป็นนวัตกรรมใหม่สำหรับเทคโนโลยีการใช้บริการอินเตอร์เน็ตผ่านดาวเทียม (IPSTAR) ในลักษณะของรถ ซึ่งเป็นรถที่ได้รับการติดตั้งชุดอุปกรณ์ดาวเทียม IPSTAR เต็มรูปแบบพร้อมใช้งาน โดยสามารถเคลื่อนย้ายไปตามสถานที่ต่างๆ ได้ตามต้องการเหมาะกับงานด้านถ่ายทอดสด หรืองานภาคสนามต่างๆ และใช้เป็นสื่อการสอนการเรียนรู้ให้บริการได้ครอบคลุมทุกพื้นที่ในประเทศไทยไม่มีข้อจำกัดด้านพื้นที่เนื่องจาก IPSTAR สามารถให้บริการอินเทอร์เน็ตความเร็วสูงได้ทั้งในย่านธุรกิจ ในย่านชุมชนเมือง หรือแม้แต่พื้นที่ห่างไกลทุรกันดาร ในขณะที่อินเทอร์เน็ตความเร็วสูงระบบอื่นๆ เช่น ADSL และ Cable Modem สามารถให้บริการได้เฉพาะพื้นที่เล็กๆ ในย่านธุรกิจสำคัญของกรุงเทพฯเท่านั้น IPSTAR ให้บริการด้านบรอดแบนด์อินเทอร์เน็ต (broadband internet) ให้กับผู้ใช้ต่างๆ ไม่ว่าจะเป็นผู้ใช้ประเภทบ้านเรือน (home users) หรืออาคารห้องชุดที่พักอาศัย (apartments) ผู้ใช้ประเภทธุรกิจทั้งกลุ่มบริษัทขนาดใหญ่ ขนาดกลาง และขนาดเล็ก (SME: small and medium enterprise) ผู้ใช้ประเภทสาธารณะ (public users) นอกจากบรอดแบนด์อินเทอร์เน็ตแล้วไอพีสตาร์ยังให้บริการด้านการถ่ายทอดสัญญาณภาพจากสถานีห้องส่งไปยังผู้ใช้ (broadcast/multicast) ซึ่งอาจจะเป็นการถ่ายทอดสัญญาณโทรทัศน์ การถ่ายทอดการอบรมสัมมนาของหน่วยงานภาครัฐ หรือเอกชน การถ่ายทอดการเรียนการสอนของการศึกษาทางไกลผ่านดาวเทียม ในลักษณะ E-Learning อีกด้วย

จากความน่าสนใจของอุปกรณ์ Mobile VSAT Vehicle (MVV) เป็นที่มาของการพัฒนาในส่วนของโปรแกรมสำหรับการติดต่อกับผู้ใช้งานขึ้นใหม่ด้วยภาษา Visual C# เพื่อให้ง่ายสะดวกต่อการใช้งานมากขึ้น และทำการแก้ไขโปรแกรมควบคุมการหมุนของแกน Polarization จากเดิมที่องศาในการหมุนไม่ถูกต้อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.2 วัตถุประสงค์ของปริญญานิพนธ์

1. เพื่อศึกษาระบบการสื่อสารสัญญาณผ่านดาวเทียมเบื้องต้น
2. เพื่อศึกษาลักษณะโครงสร้างของจานรับส่งสัญญาณดาวเทียม
3. เพื่อศึกษาระบบควบคุมการหมุนหาตำแหน่งเพื่อรับส่งสัญญาณของจานดาวเทียม ตามองศาและการเคลื่อนที่
4. เพื่อศึกษาและพัฒนาโปรแกรมสำหรับแสดงผล และควบคุมตำแหน่งของจานดาวเทียมผ่านทางเครื่องคอมพิวเตอร์ที่มีความสะดวกคล่องตัว และง่ายต่อการใช้งานของผู้ใช้
5. เพื่อศึกษาและพัฒนาโปรแกรมสำหรับควบคุมการปรับตำแหน่งจานดาวเทียม
6. เพื่อศึกษาหลักการขับเคลื่อนมอเตอร์ไฟฟ้ากระแสตรงแบบ PWM สำหรับใช้ในการกำหนดองศาการเคลื่อนที่ของจานดาวเทียม

1.3 ขอบเขตของปริญญานิพนธ์

1. สามารถใช้โปรแกรมสั่งการ และควบคุมการหมุนหาตำแหน่ง เพื่อรับส่งสัญญาณของจานดาวเทียม ได้ตามองศา ทั้ง 3 แกน คือแกนของการกวาด (Azimuth), แกนของมุมเงย (Elevation) และแกนของการหมุนขั้วคลื่น (Polarization)
2. สามารถพัฒนาโปรแกรมการแสดงผล และควบคุมจานดาวเทียมผ่านทางเครื่องคอมพิวเตอร์ ที่มีความสะดวกคล่องตัว และง่ายต่อการใช้งานของผู้ใช้ โดยใช้โปรแกรม Visual C#
3. สามารถพัฒนาโปรแกรมควบคุมตำแหน่งองศา และการหมุนของจานดาวเทียมที่มีความเที่ยงตรง

1.4 ขั้นตอนการศึกษา

1. ศึกษาโครงสร้างการทำงานของ Mobile VSAT Vehicle (MVV)
2. ศึกษาหลักการและองค์ประกอบของ IPSTAR
3. ศึกษาหลักการและองค์ประกอบของ GPS
4. ศึกษาการทำงานของมอเตอร์กระแสตรง (DC Motor)
5. ศึกษาการเขียนโปรแกรมด้วยภาษา Visual Basic 6
6. ศึกษาโปรแกรมสำหรับติดต่อกับผู้ใช้งานที่เขียนจากภาษา Visual Basic 6
7. ศึกษาการเขียนโปรแกรมด้วยภาษา Visual C#
8. เขียนโปรแกรมสำหรับติดต่อกับผู้ใช้งานด้วยภาษา Visual C#
7. ศึกษาโปรแกรมควบคุมการหมุนของมอเตอร์บนไมโครคอนโทรลเลอร์
8. แก้ไขและปรับปรุงการหมุนของแกนการหมุนขั้วคลื่น (Polarization)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

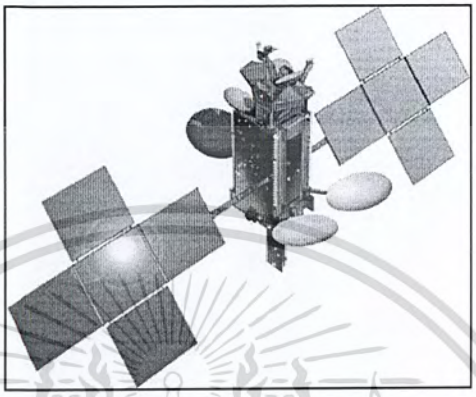
1.5 ประโยชน์ที่คาดว่าจะได้รับ

ทำให้เกิดทักษะจากการศึกษาหลักการทำงานของอุปกรณ์ต่างๆ รวมถึงสัญญาณที่ได้หรือสัญญาณที่ใช้ในการควบคุม และเมื่อพบว่ามีข้อบกพร่องเกิดขึ้นหรือระบบการทำงานได้ไม่สมบูรณ์ก็สามารถทำความเข้าใจ และวิเคราะห์หาสาเหตุของความผิดพลาดที่เกิดขึ้นตลอดจนสามารถแก้ไขปัญหาจุดบกพร่อง และปรับปรุงการปรับตำแหน่งจานดาวเทียมให้มีประสิทธิภาพมากขึ้น ในกรณีที่หมุนหาตำแหน่งเพื่อรับสัญญาณของจานดาวเทียมได้ตามองศาทั้ง 3 แกน คือ แกนของการกวาด (Azimuth), แกนของมุมเงย (Elevation) และแกนของการหมุนขั้วคลื่น (Polarization)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2
ทฤษฎีและหลักการที่เกี่ยวข้อง

2.1 องค์ประกอบของดาวเทียมไอพีสตาร์ (IPSTAR)



ภาพที่ 2.1 ดาวเทียมไอพีสตาร์

เทคโนโลยีของไอพีสตาร์ประกอบด้วย 2 ส่วนหลักที่สำคัญ คือ ระบบดาวเทียม ได้แก่ ดาวเทียมไทยคม 4 (“ไอพีสตาร์”) และระบบภาคพื้นดินซึ่งประกอบด้วยระบบควบคุมเครือข่ายหรือเกตเวย์ และอุปกรณ์ผู้ใช้ปลายทาง

2.1.1 ระบบดาวเทียม

ดาวเทียมไทยคม 4 (“ไอพีสตาร์”) ใช้เทคโนโลยีการกระจายคลื่นแบบรังผึ้งเหมือนกับที่ใช้ในระบบโทรศัพท์เคลื่อนที่ ผนวกกับระบบจานสายอากาศดาวเทียมแบบใหม่ทำให้ดาวเทียมไทยคม 4 (“ไอพีสตาร์”) สามารถนำความถี่กลับมาใช้งานใหม่ได้อย่างมีประสิทธิภาพ ซึ่งทำให้การรับส่งสัญญาณมีประสิทธิภาพที่เพิ่มขึ้นมาก และนอกจากนี้ยังใช้ระบบบริหารการส่งตามความต้องการของผู้ใช้งานเพื่อทำให้การส่งสัญญาณบนดาวเทียมมีประสิทธิภาพสูงสุดดาวเทียมไทยคม 4 (“ไอพีสตาร์”) สามารถรับส่งข้อมูลได้ถึง 45 กิกะบิตต่อวินาที (Gbps) ซึ่งสูงกว่าดาวเทียมปกติถึง 20 เท่า ทำให้สามารถรองรับความต้องการใช้งานอินเทอร์เน็ตความเร็วสูงจำนวนมากนับล้านคนทั้งระดับผู้บริโภครายย่อยจนถึงระดับองค์กรทั้งผู้ให้บริการอินเทอร์เน็ตและโทรคมนาคม

ดาวเทียมไทยคม 4 (“ไอพีสตาร์”) มีช่องสัญญาณทั้งสิ้น 94 บีมแบ่งเป็น เคยู-สปอตบีม (Spot beams) จำนวน 84 บีม เคยู-เชพบีม (Shapedbeams) จำนวน 3 บีม และ เคยู-บรอดคาสต์บีม (Broadcast beams) จำนวน7 บีม โดยดาวเทียมไทยคม 4 (“ไอพีสตาร์”) มีขอบเขตการให้บริการครอบคลุมภาคพื้นเอเชียแปซิฟิก ออสเตรเลีย และนิวซีแลนด์ เป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.1 รายละเอียดดาวเทียมไทยคม 4 ("ไอพีสตาร์")

การออกแบบ	ดาวเทียมรุ่น LS-1300 SX
บริษัทผู้ผลิตดาวเทียม	บริษัท สเปซ ซิสเต็มส์ ลอเรล ประเทศสหรัฐอเมริกา
อายุการใช้งาน	อย่างน้อย 12 ปี
ตำแหน่งวงโคจร	119.5 องศาตะวันออก
จำนวนช่องสัญญาณ	Ku-Spot Beam จำนวน 84 บีม Ku-Shape Beam จำนวน 3 บีม Ku-Broadcast Beam จำนวน 7 บีม
น้ำหนักก่อนส่งขึ้นสู่วงโคจรในอวกาศ	6,505 กิโลกรัม
กำลังไฟฟ้า	ประมาณ 14,400 วัตต์ เมื่อสิ้นปีที่ 12
การส่งดาวเทียมสู่วงโคจร	จรวดเอเรียน 5G ของบริษัท แอเรียนสเปซ ประเทศฝรั่งเศส
วันที่ส่งดาวเทียมขึ้นสู่วงโคจร	11 สิงหาคม 2548

2.1.2 ระบบภาคพื้นดินของไอพีสตาร์

2.1.2.1 อุปกรณ์ปลายทาง และกระบวนการส่งสัญญาณ

- ใช้เทคโนโลยีด้านการเข้ารหัส และการปรับแต่งสัญญาณแบบใหม่เพื่อส่งผลทำให้เพิ่มความสามารถในการรับส่งสัญญาณโดยใช้กำลังส่งที่ต่ำลง ลดขนาดจานสายอากาศ และลดค่าใช้จ่ายในส่วนอุปกรณ์ปลายทาง

- สามารถจัดสรรแบนด์วิดท์ให้ถูกใช้งานได้อย่างมีประสิทธิภาพอย่างสูงสุด โดยผู้ใช้สามารถใช้แบนด์วิดท์ร่วมกันได้ ซึ่งเหมาะสำหรับการใช้บริการอินเทอร์เน็ตผ่านดาวเทียมแบบ 2 ทางที่การรับส่งข้อมูลมีปริมาณไม่เท่ากัน นอกจากนั้นความสามารถในการจัดสรรแบนด์วิดท์ของไอพีสตาร์ยังเหมาะกับการใช้งานแบนด์วิดท์ที่สูงขึ้นเป็นบางช่วง (Bursty) เช่น การเรียกหรือดาวน์โหลดหน้าเว็บไซต์ต่างๆ ซึ่งระบบของไอพีสตาร์จะมีการใช้แบนด์วิดท์เฉพาะช่วงที่มีการเรียกหรือดาวน์โหลดเท่านั้นจึงช่วยให้ประหยัดด้านการใช้แบนด์วิดท์ และลดค่าใช้จ่ายในการใช้บริการอินเทอร์เน็ต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

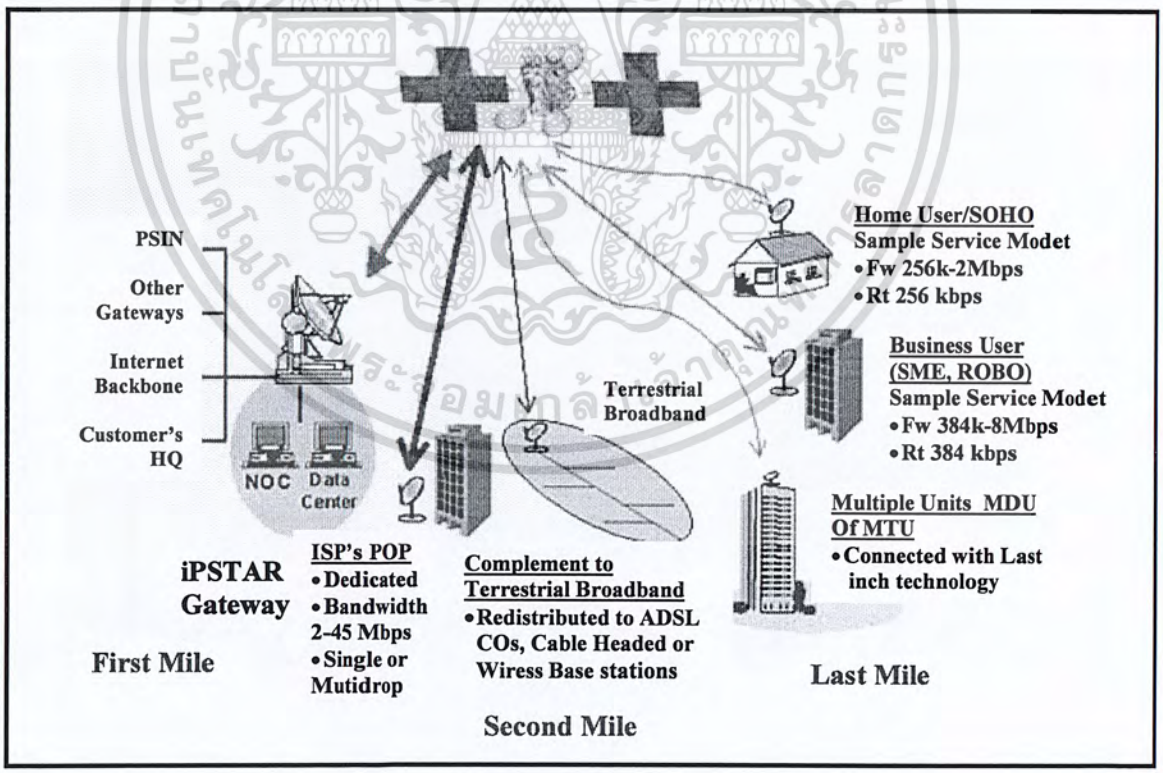
2.1.2.2 ระบบควบคุมเครือข่าย หรือ เกตเวย์

ระบบควบคุมเครือข่ายหรือเกตเวย์ คือ สถานีควบคุมระบบเครือข่ายภาคพื้นดิน ทำหน้าที่ในการเชื่อมต่อสัญญาณจากทุกๆ อุปกรณ์ปลายทางของผู้ใช้เข้ากับเครือข่ายอินเทอร์เน็ต ชุมสายโทรศัพท์ เครือข่ายของบริษัทและองค์กร หรือสถานีควบคุมระบบเครือข่ายภาคพื้นดินของไอพีสตาร์ในประเทศอื่นๆ ด้วยเทคโนโลยีที่ล้ำหน้าของไอพีสตาร์ช่วยให้อุปกรณ์ภาคพื้นดินของไอพีสตาร์มีต้นทุนที่ต่ำกว่าอุปกรณ์สื่อสารดาวเทียมภาคพื้นดินระบบต่างๆ ใช้อยู่ในปัจจุบัน

รูปแบบคลื่นวิทยุของภาคส่งจะใช้เทคโนโลยีแบบ OFDM/TDM (Orthogonal Frequency Division Multiplex/ Time Division Multiplex) เทคโนโลยีดังกล่าวจะช่วยเพิ่มประสิทธิภาพสูงสุดให้การส่งข้อมูลที่มีความเร็วสูงขึ้น

2.1.2.3 การให้บริการของดาวเทียมไอพีสตาร์

ไอพีสตาร์ให้บริการด้านบรอดแบนด์อินเทอร์เน็ต (Broadband internet) ให้กับผู้ใช้ต่างๆ ไม่ว่าจะเป็นผู้ใช้ประเภทบ้านเรือน (home users) หรืออาคารห้องชุดที่พักอาศัย (apartments) ผู้ใช้ประเภทธุรกิจทั้งกลุ่มบริษัทขนาดใหญ่ ธุรกิจขนาดกลาง และธุรกิจเล็ก (SME) ผู้ใช้ประเภทสาธารณะ (public users) อาทิเช่น อินเทอร์เน็ตคาเฟ่



ภาพที่ 2.2 การให้บริการของดาวเทียมไอพีสตาร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

นอกจากบรอดแบนด์อินเทอร์เน็ตแล้ว ไอพีสตาร์ยังให้บริการในด้านการถ่ายทอดสัญญาณภาพจากสถานีห้องส่งไปยังผู้ใช้ (broadcast/multicast) ซึ่งอาจจะเป็นการถ่ายทอดสัญญาณโทรทัศน์ หรือการถ่ายทอดการอบรมสัมมนาของหน่วยงานภาครัฐหรือเอกชน หรือการถ่ายทอดการเรียนการสอนของการศึกษาทางไกลผ่านดาวเทียมหรือการเรียนการสอนในลักษณะ E-Learning

นอกจากการบริการข้างต้นแล้วไอพีสตาร์ยังให้บริการด้านอินเทอร์เน็ต ภาพวิดีโอและเสียงสำหรับองค์กรต่างๆ ระหว่างสำนักงานใหญ่ และสำนักงานสาขารวมไปถึงการประชุมผ่านวิดีโอคอนเฟอเรน

2.1.2.4 การนำไอพีสตาร์มาใช้กับบรอดแบนด์อินเทอร์เน็ต

เนื่องจากเทคโนโลยีไอพีสตาร์ได้ถูกออกแบบขึ้นเพื่อการรองรับการใช้งานในด้านบรอดแบนด์อินเทอร์เน็ตโดยตรง โดยดาวเทียมไทยคม 4 (“ไอพีสตาร์”) ถูกสร้างมาเพื่อให้บริการแก่ผู้ใช้เป็นจำนวนมากนั่นคือมีช่องสัญญาณรวม 45 กิกะบิตต่อวินาที สามารถรองรับผู้ใช้งานจำนวนมากนับล้านคน และยังเป็นดาวเทียมที่มีช่องสัญญาณราคาถูกที่สุดเมื่อเปรียบเทียบกับระบบการสื่อสารดาวเทียมในปัจจุบันประกอบกับอุปกรณ์ผู้ใช้ปลายทาง ซึ่งมีแนวโน้มว่าราคาจะลดลงอีกหากมีจำนวนผู้ใช้งานมากขึ้นในอนาคตจะส่งผลให้ผู้ใช้งานในระดับบุคคลทั่วไป องค์กรขนาดเล็กและขนาดกลางสามารถใช้บรอดแบนด์อินเทอร์เน็ตผ่านดาวเทียมเพื่อกิจกรรมทางธุรกิจ และความบันเทิงได้อย่างไม่มีขีดจำกัด

อุปกรณ์ผู้ใช้ปลายทางของไอพีสตาร์นั้นยังสามารถใช้งานได้กับดาวเทียมทั่วไป (Conventional Satellite) ทำให้ผู้ใช้งานอินเทอร์เน็ตสามารถใช้ระบบของไอพีสตาร์ได้เพียงแค่ติดตั้งชุดอุปกรณ์ไอพีสตาร์และจานดาวเทียมขนาดเล็กที่มีเส้นผ่านศูนย์กลาง 120 ซม. ก็สามารถจะรับส่งข้อมูลขนาดใหญ่ผ่านทางอินเทอร์เน็ตได้ ปัจจุบันผู้ใช้งานสามารถใช้บริการบรอดแบนด์อินเทอร์เน็ตผ่านดาวเทียมไทยคม 4 (“ไอพีสตาร์”) ได้อย่างมีประสิทธิภาพยิ่งขึ้นโดยมีสัญญาณขารับขนาด 4 เม็กกะบิตต่อวินาที (Mbps) และสัญญาณขาส่งขนาด 2 เม็กกะบิตต่อวินาที (Mbps)

2.2 ระบบระบุพิกัดบนพื้นโลกจีพีเอส (Global Positioning System : GPS)

จีพีเอส (GPS) ย่อมาจาก Global Positioning System หมายถึง ระบบบอกตำแหน่งบนพื้นผิวโลกโดยอาศัยการคำนวณพิกัด (เส้นรุ้ง-เส้นแวง) จากสัญญาณนาฬิกาที่ส่งมาจากดาวเทียมที่โคจรอยู่รอบโลกซึ่งมีตำแหน่งที่แน่นอน ระบบนี้สามารถบอกตำแหน่ง ณ จุดที่สามารถรับสัญญาณได้ทั่วโลก โดยเครื่องรับสัญญาณจีพีเอสรุ่นใหม่ๆ จะสามารถคำนวณความเร็วและทิศทางนำมาใช้ร่วมกับโปรแกรมแผนที่ เพื่อใช้ในการนำทางได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แนวคิดในการพัฒนาระบบจีพีเอสเริ่มต้นตั้งแต่ปี ค.ศ. 1957 เมื่อนักวิทยาศาสตร์ของสหรัฐอเมริกานำโดย Dr. Richard B. Kershner ได้ติดตามการส่งดาวเทียมสปุตนิกของโซเวียตและพบปรากฏการณ์ดอปเปอร์ของคลื่นวิทยุที่ส่งมาจากดาวเทียม พวกเขาพบว่าหากทราบตำแหน่งที่แน่นอนบนพื้นผิวโลกก็สามารถระบุตำแหน่งของดาวเทียมได้จากการตรวจวัดดอปเปอร์ และในทางกลับกันหากทราบตำแหน่งที่แน่นอนของดาวเทียม ก็สามารถระบุตำแหน่งบนพื้นโลกได้

กองทัพเรือสหรัฐได้ทดลองระบบนำทางด้วยดาวเทียมชื่อ TRANSIT เป็นครั้งแรกเมื่อ ค.ศ. 1960 ประกอบด้วยดาวเทียมจำนวน 5 ดวง ส่วนดาวเทียมที่ใช้ในระบบจีพีเอส (GPS Block-I) ส่งขึ้นทดลองเป็นครั้งแรกเมื่อ ค.ศ. 1978 เพื่อใช้ในทางการทหาร

เมื่อ ค.ศ. 1983 หลังจากเกิดเหตุการณ์โคเรียนแอร์ไลน์เที่ยวบินที่ 007 ของเกาหลีใต้บินพลัดหลงเข้าไปในน่านฟ้าของสหภาพโซเวียตและถูกยิงตก ผู้โดยสาร 269 คนเสียชีวิตทั้งหมด ประธานาธิบดีโรนัลด์ เรแกน จึงได้ประกาศว่าเมื่อพัฒนาระบบจีพีเอสแล้วเสร็จจะอนุญาตให้ประชาชนทั่วไปใช้งานได้

ดาวเทียมที่ใช้กับระบบจีพีเอสเป็นดาวเทียมที่มีวงโคจรต่ำที่ระดับความสูง 11,000 ไมล์จากพื้นโลก ใช้การยืนยันตำแหน่งโดยอาศัยพิกัดจากดาวเทียม 3 หรือ 4 ดวง ดาวเทียมจะโคจรรอบโลกเป็นเวลา 12 ชั่วโมงต่อหนึ่งรอบ ที่ความเร็ว 4 กิโลเมตร/วินาที การโคจรแต่ละรอบนั้นสามารถได้เป็น 6 ระนาบๆ ละ 4 ดวง ทำมุม 55 องศา โดยทั้งระบบจะต้องมีดาวเทียม 24 ดวง หรือมากกว่าเพื่อให้สามารถยืนยันตำแหน่งได้ครอบคลุมทุกจุดบนผิวโลก ปัจจุบัน เป็นดาวเทียม GPS Block-II

ระบบจีพีเอส ในปัจจุบัน มีหลายระบบ ได้แก่

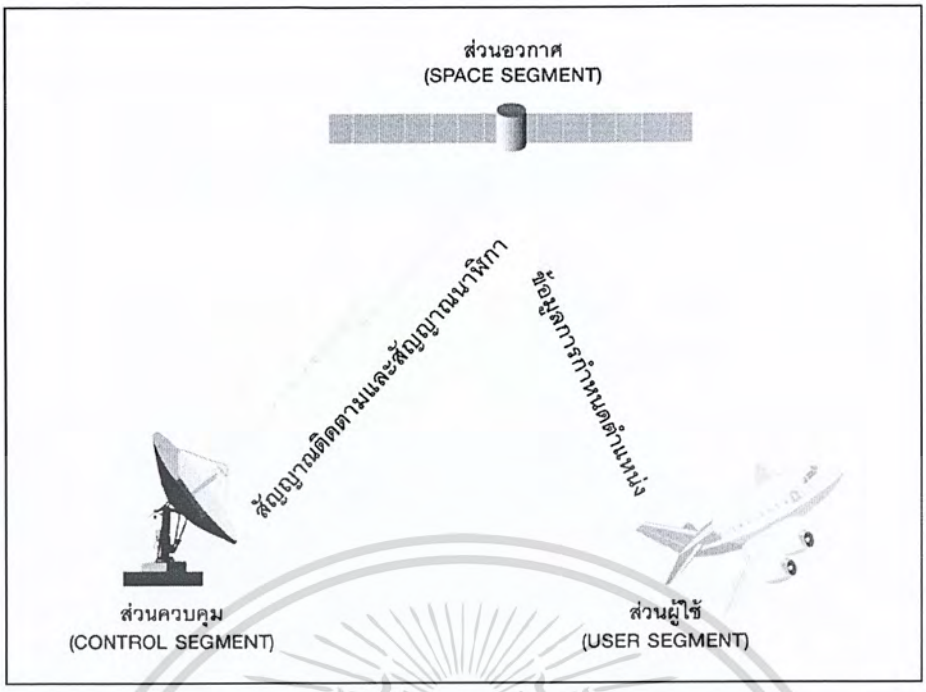
- NAVSTAR GPS (Navigation Satellite Timing And Ranging Global Positioning System) พัฒนาโดยกระทรวงกลาโหมของสหรัฐอเมริกา
- GLONASS (Global Navigation Satellite System) เป็นระบบของรัสเซีย ที่พัฒนาเพื่อแข่งขันกับสหรัฐอเมริกา แต่ระบบนี้ยังใช้งานได้ไม่สมบูรณ์ใช้งานได้ดีเฉพาะในรัสเซีย ยุโรป และแคนาดา
- GALILEO เป็นระบบที่กำลังพัฒนาโดยสหภาพยุโรปร่วมกับจีน ยูเครน อิสราเอล อินเดีย โมร็อกโก ซาอุดิอาระเบีย และเกาหลีใต้ จะแล้วเสร็จในปี พ.ศ. 2553
- BEIDOU เป็นระบบที่กำลังพัฒนาโดยประเทศจีน

2.2.1 ส่วนประกอบของระบบจีพีเอส

ระบบจีพีเอสประกอบไปด้วย 3 ส่วน ดังนี้คือ

- ส่วนอวกาศ
- ส่วนสถานีควบคุม
- ส่วนผู้ใช้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



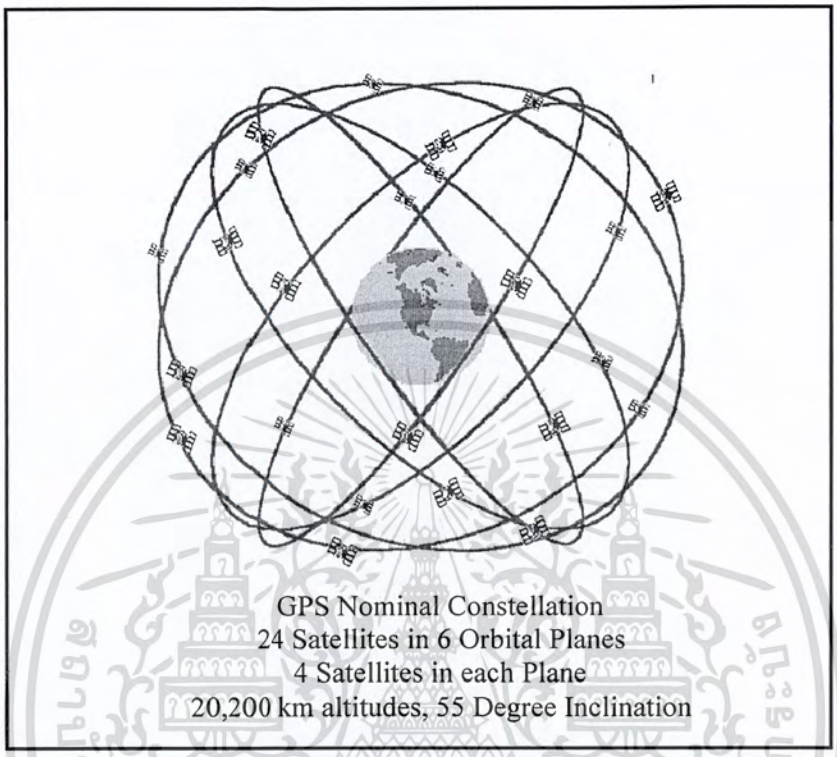
ภาพ ที่ 2.3 แสดงองค์ประกอบของระบบดาวเทียม GPS

2.2.1.1 ส่วนอวกาศ (SPACE SEGMENT)

ประกอบด้วยดาวเทียมทั้งหมด 24 ดวง โคจรรอบโลกที่ระยะ 11,000 ไมล์อากาศจากพื้นโลกใช้เวลา 12 ช.ม. ในการโคจรรอบโลกหนึ่งรอบ คือ คาบของการโคจรเป็น 12 ชั่วโมง/รอบ ความถี่ที่ใช้ในการบอกตำแหน่งค่าพิกัดของดาวเทียมแต่ละดวงมี 2 ความถี่ คือ ความถี่ L1:1,575.42 MHz และความถี่ L2:1,27.60 MHz โดยดาวเทียมจำนวน 21 ดวงจะใช้ในการบอกค่าพิกัด ส่วนที่เหลือ 3 ดวงจะสำรองเอาไว้ดาวเทียมทั้ง 24 ดวงนี้จะมีวงโคจรอยู่ 6 วงโคจรด้วยกัน โดยแบ่งจำนวนดาวเทียมวงโคจรละ 4 ดวง และมีรัศมีวงโคจรสูงจากพื้นโลกประมาณ 20,200 กิโลเมตร (12,600 ไมล์) วงโคจรทั้ง 6 จะเอียงทำมุมกับเส้นศูนย์สูตร (Equator) เป็นมุม 55 องศา การวางวงโคจรเช่นนี้ทำให้เราสามารถรับสัญญาณจากดาวเทียมได้คราวละถึง 6 ดวง ดาวเทียมติดตั้งนาฬิกาที่เที่ยงตรงมากๆ ถึง 3 nanoseconds (ความเที่ยงตรง 0.000000003 ของวินาที หรือ 3e-9) ความเที่ยงตรงมีความสำคัญมากสำหรับเครื่องรับเพราะเครื่องรับจำเป็นต้องทราบเวลาที่เที่ยงตรง แน่นอนว่าระยะเวลาเท่าไรที่สัญญาณคลื่นจากดาวเทียมเดินทางถึงเครื่องรับ ดาวเทียมแต่ละดวงมีเชื้อเพลิง และเครื่องยนต์ขนาดเล็กซึ่งสามารถที่จะปรับแต่งดาวเทียมให้อยู่ในตำแหน่งที่ถูกต้องในวงโคจร ถ้าดาวเทียมเกิดเคลื่อนออกจากตำแหน่งที่กำหนดดาวเทียมแต่ละดวงมีนาฬิกา atomic clocks 4 อัน นาฬิกานี้มีความเที่ยงตรงถึงหนึ่งในหนึ่งพันล้านของวินาที หรือ nanosecond ดาวเทียมแต่ละดวงจะส่งคลื่นสัญญาณออกมาสองคลื่นสัญญาณ หนึ่งคลื่นสำหรับการทหาร และอีกคลื่นหนึ่งสำหรับพลเรือน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตำแหน่งของดาวเทียมจะถูกคำนวณให้เครื่องรับหาตำแหน่งของผู้ใช้ที่สามารถรับข้อมูลได้ 50 bps ต่อเนื่องกัน วงโคจรของแต่ละดวงต่อระยะเวลา 1 ชั่วโมง โดยการตั้ง Element การโคจรที่ 15 keplerian พร้อมทั้งค่าสัมประสิทธ์ฮาร์โมนิคเพิ่มขึ้นจากการรบกวนและแก้ไขทุกๆ 4 ชั่วโมง



ภาพที่ 2.4 แสดงการโคจรของดาวเทียม GPS รอบโลก

2.2.1.2 ส่วนสถานีควบคุม (Control Station Segment)

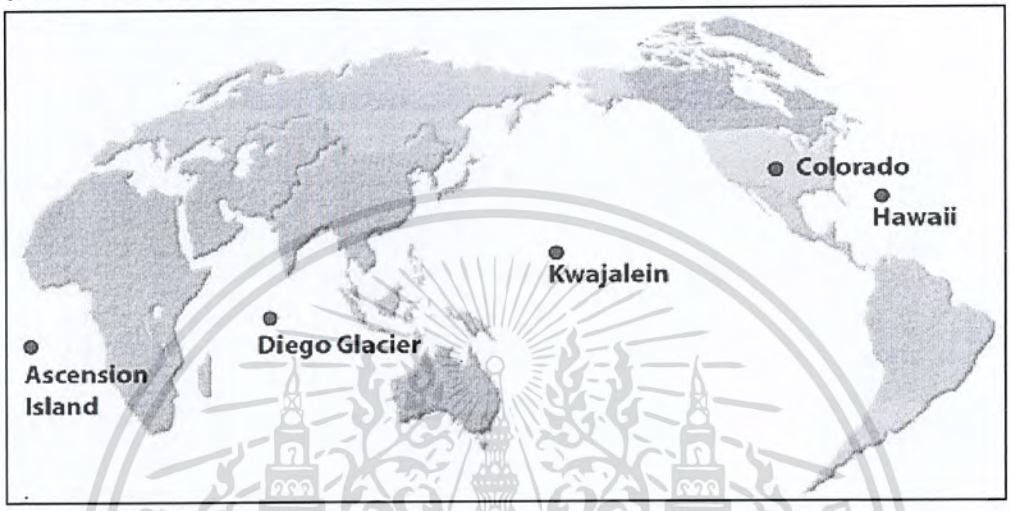
ส่วนสถานีควบคุมประกอบด้วย 5 สถานีย่อยกระจายอยู่ทั่วโลกทำหน้าที่คอยติดต่อสื่อสาร (Tracking) กับดาวเทียมและทำการคำนวณผล (Computation) เพื่อบอกตำแหน่งของดาวเทียมแต่ละดวง และส่งข้อมูลที่ได้ไปยังดาวเทียมอยู่ตลอดเวลาทำให้ข้อมูลที่ได้เป็นข้อมูลที่ทันสมัยอยู่เสมอ สถานีควบคุมดาวเทียมประกอบด้วย

- สถานีควบคุมแม่ข่าย (Master Control Station) มีอยู่ 1 สถานี ทำหน้าที่รับผิดชอบในการจัดการทั่วไป และบริการสถานีลูกข่ายเป็นศูนย์กลางที่ให้การสนับสนุนการทำงานเครื่องแม่ข่ายจะคำนวณตำแหน่ง และนาฬิกาดูความคลาดเคลื่อนของดาวเทียมแต่ละดวงจากสถานีลูกข่ายภาคพื้นและสั่งคำสั่งแก้ไขกลับไปยังสถานีลูกข่ายเพื่อส่งไปยังดาวเทียมดวงนั้นๆ

- สถานีควบคุมลูกข่าย (Monitor Stations) มีอยู่ 4 สถานี ซึ่งจะทำการตรวจสอบความสูง, ตำแหน่ง, ความเร็ว และวงจรทั่วไปของดาวเทียม สถานีควบคุมนี้จะตรวจสอบดาวเทียมได้ครั้งละ 11 ดวง การตรวจสอบนี้แต่ละสถานีกระทำวันละ 2 ครั้ง เมื่อดาวเทียมโคจรรอบโลก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- สายอากาศภาคพื้นดิน (Ground Uplink Antenna Facility) ส่วนสถานีสายอากาศภาคพื้นดินจะทำการส่งคำสั่งข้อมูลการนำร่อง และข้อมูลอื่นๆ ที่เรียกว่า ทีทีแอนซี (Telemetry, Tracking and Command: TT&C) ซึ่งเตรียมพร้อมโดยสถานีควบคุมหลัก สำหรับดาวเทียมแต่ละดวง ข้อมูลเหล่านี้จะถูกส่งไปยังสถานีสายอากาศภาคพื้นดิน และเก็บไว้จนกว่าดาวเทียมดวงที่ต้องการส่งข้อมูลจะผ่านมาโดยจะส่งคลื่นความถี่ S-Band โดยสถานีสายอากาศจะตั้งอยู่กับสถานีสังเกตการณ์



ภาพที่ 2.5 แสดงที่ตั้งของสถานีควบคุมดาวเทียม GPS 5 แห่ง

2.2.1.3 ส่วนผู้ใช้ (User Segment)

ในส่วนนี้เป็นส่วนประกอบที่สำคัญ คือ เครื่องรับสัญญาณจีพีเอส โดยจะรับสัญญาณ L-Band ที่ถูกส่งมาจากดาวเทียมและนำมาคำนวณเพื่อหาตำแหน่ง ความเร็วและเวลา ของเครื่องรับ จากนั้นจะนำค่ามาประยุกต์ใช้งานตามลักษณะการใช้งานซึ่งเครื่องรับจีพีเอสสามารถแบ่งออกเป็น 2 กลุ่ม คือ เครื่องประเภทที่สามารถรับดาวเทียมได้ 4 ดวง หรือมากกว่าได้พร้อมกันทีเดียวกับเครื่องที่มีการรับดาวเทียมโดยการเรียงลำดับ และแต่ละกลุ่มยังแบ่งย่อยได้อีก คือ

- เครื่องรับแบบเรียงลำดับสัญญาณดาวเทียม

เครื่องรับช่องเดียวแบบจำกัดพลังงาน (Starved - Power Single Receivers) เครื่องแบบนี้ออกแบบให้พกพาได้และสามารถทำงานได้ด้วยถ่านไฟฉายขนาดเล็ก การจำกัดการใช้กระแสไฟโดยให้ปิดการทำงานตัวเองโดยอัตโนมัติเมื่อแสดงตำแหน่งครั้งสองครั้งใน 1 นาที ซึ่งจะเหมาะสำหรับใช้งานบอกตำแหน่งส่วนตัว ข้อเสีย คือ ความถูกต้องของ GPS ไม่ดี, ต่อเชื่อมกับอุปกรณ์อื่นไม่ได้ และไม่สามารถใช้วัดหาความเร็วได้

เครื่องรับช่องเดียว (Single Channel Receivers) เป็นเครื่องรับสัญญาณห้องเดียวใช้ทำงานหาระยะจากดาวเทียมทุกดวง แต่ที่ไม่เหมือนคือเครื่องรับช่องเดียวแบบมาตรฐานไม่จำกัดที่กำลังไฟ ดังนั้นจึงทำการรับต่อเนื่องได้มีผลทำให้ความถูกต้องสูงกว่า และใช้วัดหาความเร็วได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เครื่องรับช่องเดียวแบบมัลติเพล็กเร็ว (Fast - Multiplexing Single Receivers) เครื่องรับนี้สามารถเปลี่ยนดาวเทียมได้เร็วกว่ามาก ข้อดี คือ สามารถทำการวัดได้ในขณะที่กำลังรับข้อมูลจากดาวเทียม ดังนั้นเครื่องทำงานได้อย่างต่อเนื่อง และการที่มีนาฬิกาไม่เที่ยงตรงจึงมีผลต่อเครื่องประเภทนี้น้อย

เครื่องรับสัญญาณสองช่องตามลำดับ (Two-Channel Sequencing Receivers) การเพิ่มช่องรับสัญญาณขึ้นอีกหนึ่งช่อง ช่วยให้เครื่องเพิ่มขีดความสามารถขึ้นอย่างเห็นได้ชัด

- **เครื่องรับแบบต่อเนื่อง (Continuous Receivers)**

เครื่องรับที่สามารถรับสัญญาณดาวเทียมพร้อมกันได้ตั้งแต่ 4 ดวงขึ้นไป และสามารถแสดงผลค่าตำแหน่งและความเร็วได้ทันทีการรับดาวเทียมได้ทั้ง 4 ดวง พร้อมกับที่มีค่าในการวัดหาในขณะที่มีการเปลี่ยนตำแหน่งรวดเร็วหรือต้องการความถูกต้องสูง ดังนั้นเครื่องแบบนี้จึงนำมาใช้ในงานรังวัดและทางด้านวิทยุซึ่งจะพบว่าจะมีช่องรับสัญญาณทั้ง 4, 5, 8, 10 และ 12 ช่อง นอกจากข้อดีที่ใช้วัดตำแหน่งอย่างต่อเนื่องได้แล้ว เครื่องรับ GPS แบบหลายช่องสามารถช่วยขจัดปัญหา GDOP ได้อีกด้วย คือ แทนที่จะรับดาวเทียม 4 ดวงใดก็ได้จะคำนวณหาค่า GDOP ดาวเทียม 4 ดวงของกลุ่มดาวเทียมที่ขึ้นอยู่ และทำการวัดจากดาวกลุ่มที่มีค่า GDOP ต่ำสุด เครื่องรับ 4 ช่องสัญญาณ สามารถให้ค่าสัญญาณต่อสัญญาณรบกวน (Signal to Noise Ratio) เพิ่มขึ้นเป็นสองเท่าของเครื่อง 2 ช่อง และเป็นสี่เท่าของเครื่องรับแบบช่องเดียว การเปรียบเทียบค่าการรับของแต่ละช่องเครื่องสามารถปรับตั้งค่าพิกัดเทียมระหว่างช่องรับสัญญาณ ซึ่งช่วยทำให้การวัดมีความถูกต้องแม่นยำมากขึ้น นอกจากข้อดีข้อเสียที่กล่าวมาแล้วนั้นยังมีข้ออื่นมาพิจารณาอีกคือมีเครื่องแบบใหม่ที่สามารถได้ค่าความถูกต้องสูงมาก โดยการใช้ทั้งรหัส Pseudo Random ที่กล่าวมาแล้ว และใช้ความถี่ของคลื่นพาห์ (Carrier Frequency) ซึ่งทำให้เครื่องรับทำงานมีความเที่ยงสูงที่รหัส Pseudo Random ไม่สามารถให้ได้ และใช้ในการวัดหาเวลาได้แม่นยำมากขึ้น ซึ่งช่วยในการบอกตำแหน่งได้ดียิ่งขึ้นด้วยและมีบางเครื่องที่ไม่ต้องใส่ค่าประมาณตำแหน่งและเวลา โดยประมาณให้เครื่องก่อนทำการวัดเครื่องรับแบบนี้จะใช้ตัวเองในการใส่ค่าเริ่มตำแหน่งได้โดยตัวมันเอง ข้อที่ควรพิจารณา คือ การต่อเชื่อมกับอุปกรณ์อื่น และความสะดวกบางเครื่องแสดงได้เฉพาะพิกัดภูมิศาสตร์ บางเครื่องไม่สามารถต่อเข้ากับเครื่องมืออื่นหรือคอมพิวเตอร์ขนาดเล็ก (PC) ได้ และข้อใหญ่ที่ต้องพิจารณาความแข็งแรงทนทานถ้าต้องใช้เครื่องทำงานที่พื้นที่ทะเล หรือที่พื้นที่ป่าเขา การใช้ไฟและความร้อนที่เกิดขึ้นเป็นตัวชี้สำคัญที่จะต้องเอาใจใส่ ในทางสถิติแสดงให้เห็นว่าอัตราของค่าความผิดพลาดจะเพิ่มเป็นสองเท่าของความร้อนในเครื่องเพิ่มทุกๆ 7 องศาฟาเรนไฮต์ เครื่องรับรุ่นใหม่ปัจจุบันได้เพิ่มคุณค่าให้แก่เครื่องรับ GPS อีกหลายประการ เช่น ใช้การประมวลผลที่ซับซ้อนแสดงผลด้วยจอภาพรายละเอียด เครื่องรับ GPS อาจแสดงจุดตำแหน่งบนแผนที่ที่ได้วาดไว้แล้วให้เห็นทันที

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.2 หลักการทำงานของ GPS

2.2.2.1 การให้บริการจีพีเอส (GPS Service)

การให้บริการจีพีเอสมีอยู่ 2 ระดับ คือ บริการบอกตำแหน่งแบบสมบูรณ์ หรือพีพีเอส (PPS : Precise Positioning Service) และบริการบอกตำแหน่งแบบมาตรฐานหรือเอสพีเอส (SPS : Standard Positioning Service)

- บริการบอกตำแหน่งแบบสมบูรณ์ (Precise Positioning Service : PPS)

พีพีเอสเป็นบริการที่ให้ตำแหน่ง ความเร็ว และเวลาที่ถูกต้องเฉพาะผู้ที่ได้รับอนุญาตให้ใช้งานเท่านั้น พีพีเอสจะเน้นไปในส่วนทางการทหาร การได้รับอนุญาตให้ใช้พีพีเอสจะพิจารณาโดย U.S.Military User , NATO Military User และกองทัพอื่นๆ หรือพลเรือน เช่น Australian Defense Forces , U.S. Defense Mapping Agency พีพีเอสจะให้ค่าความคลาดเคลื่อนประมาณ 16 เมตร และความผิดพลาดทางด้านเวลาไม่เกิน 100 นาโนเซค ส่วนผู้ใช้งานที่ไม่ได้รับอนุญาตจะให้ค่าที่แตกต่าง คือ ประมาณ 37 เมตร และ 197 นาโนเซค ภายใต้เงื่อนไขการทำงานโดยปกติของระบบเครื่องรับแบบพีพีเอส สามารถให้ค่าถูกต้องของความเร็วมีค่าความคาดเคลื่อนไม่เกิน 0.2 เมตรต่อวินาที แต่ก็ขึ้นกับการออกแบบเครื่องรับด้วย

- บริการบอกตำแหน่งแบบมาตรฐาน (Standard Positioning Service : SPS)

เอสพีเอสเป็นบริการที่ให้ความถูกต้องน้อยกว่าพีพีเอสแต่จะใช้กับผู้ใช้งานทั้งหมดในสภาวะปกติ ระดับของเอสเอจะถูกควบคุมเพื่อให้ค่าความถูกต้องในแนวเส้นขอบฟ้า 100 เมตร มีค่าประมาณ 156 เมตร เครื่องรับเอสพีเอสมีความสามารถด้านความถูกต้องของเวลาประมาณ 337 นาโนเซค การลดความถูกต้องของระบบสามารถเพิ่มขึ้นได้ถ้ามีความจำเป็น เช่น ในเวลาศึกสงครามโดยประธานาธิบดีสหรัฐอเมริกาเท่านั้นที่มีอำนาจสั่งการผ่าน U.S National Command Authority เพื่อเปลี่ยนระดับของเอสเอเป็นระดับอื่น

2.2.2.2 การทำงานของเครื่องรับ จีพีเอส

- การเลือกดาวเทียม (Satellite Selection)

กระบวนการแทรคกิ้งจะเริ่มต้น โดยเครื่องรับจีพีเอสจะสำรวจค้นหาดาวเทียมดวงที่เป็นไปได้ในการแทรค ถ้าเครื่องรับสามารถตัดสินการมองเห็นดาวเทียมได้ทันทีจะเล็งดาวเทียมเป้าหมายเพื่อจะทำการแทรคและเริ่มกระบวนการรับสัญญาณการมองเห็น (Satellite Visibility) ซึ่งจะตัดสินจากข้อมูลอัลมาแนค (GPS Satellite Almanac) และค่าการประมาณ (หรือ user point) เริ่มต้นของเวลาและตำแหน่งของเครื่องรับ หากเครื่องรับไม่มีค่าเหล่านี้เก็บไว้จะเริ่มทำการสำรวจท้องฟ้า (Search the Sky) ซึ่งจะค้นหาซูโดแรนดอมน้อยส์ซึ่งก็คือรหัสซีเอโค้ด จนล็อคได้จากดาวเทียมดวงหนึ่งที่อยู่ในวิสัย เมื่อดาวเทียมถูกแทรคแล้วเครื่องรับจะสามารถดีมอดูเลตข้อมูลการนำร่องและได้รับค่าปัจจุบันของข้อมูลอัลมาแนค เช่นเดียวกับสถานะของสภาพของดาวเทียมที่

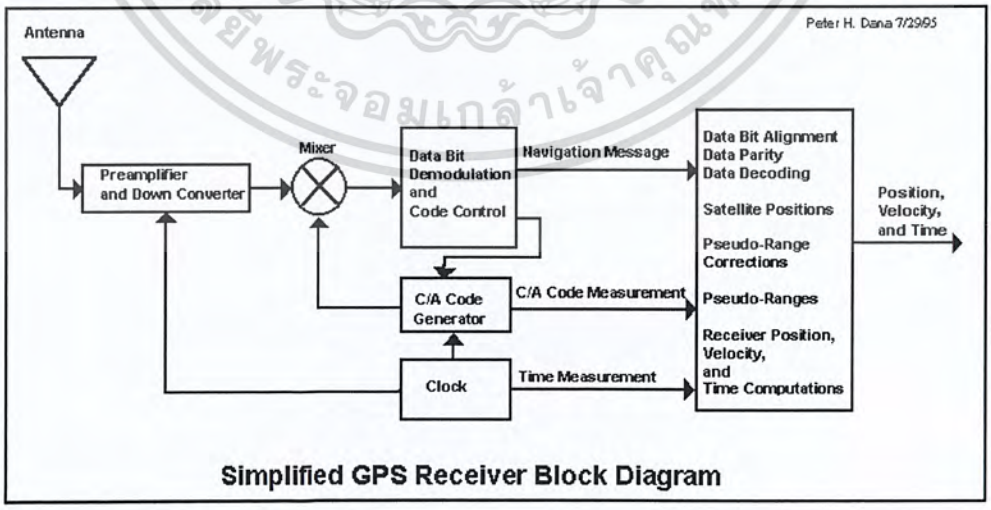
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เหลือทั้งหมดในกลุ่ม การเลือกดาวเทียมนั้นขึ้นกับสถาปัตยกรรมของเครื่องรับอาจจะเลือกกลุ่มที่ดีที่สุดในดาวเทียมที่มองเห็นหรือใช้ดาวเทียมที่มีสภาพดีทั้งหมดเพื่อใช้พิจารณาหาตำแหน่ง ความเร็ว และเวลา ผลจากการคำนวณมักจะมีความถูกต้องมากกว่าการใช้ดาวเทียม 4 ดวง ถึงแม้จะต้องการความซับซ้อนของเครื่องรับและการประมวลผลมากกว่า

เครื่องรับส่วนใหญ่จะแทรคดาวเทียมได้มากกว่า 4 ดวง แต่น้อยกว่าที่เห็นในวิสัยทั้งหมด ซึ่งเนื่องจากการประนีประนอมระหว่างความซับซ้อน ความถูกต้อง และความแข็งแรงของเครื่องรับที่ใช้

2.2.2.3 การรับสัญญาณดาวเทียม (Satellite Signal Acquisition)

กำลังของสัญญาณดาวเทียมที่พื้นผิวโลก จะมีค่าต่ำกว่าระดับสัญญาณของเสียงรบกวนเนื่องจากการมอดูเลตสัญญาณ โดยวิธีการสเปกตรัมความสูงของวงโคจรและกำลังส่งเทียม เพื่อที่จะนำสัญญาณกลับคืนมา เครื่องรับจะใช้เทคนิคโค้ดคีรีเลชั่นโดยจะสร้างสัญญาณเลียนแบบสัญญาณที่รับเข้ามาและนำมาจัดให้ตรงกับสัญญาณที่ได้รับ เครื่องรับจะเลื่อนสัญญาณเลียนแบบให้ตรงกับสัญญาณดาวเทียม เมื่อโค้ดเกิดการตรงกันสัญญาณก็จะถูกคอมเพรสสัญญาณกลับมาเป็นความถี่พาห์ ต้นกำเนิดความล่าช้าของสัญญาณในรหัสของเครื่องรับคือ เวลาที่ใช้ในการเดินทางของสัญญาณระหว่างดาวเทียมกับเครื่องรับ ทำให้ได้ระยะทางออกมาเรียกว่า Pseudo range เพราะว่ายังไม่ใช่ระยะทางที่แท้จริง เนื่องจากยังไม่ได้ลบค่าไบอัสของสัญญาณนาฬิกา (Clock Bias) ของเครื่องรับออกไป ปกติเครื่องรับจะใช้วิธีการเฟสลอคลูปเพื่อซิงโครไนซ์สัญญาณที่เครื่องรับสร้างขึ้นภายในกับสัญญาณที่ได้รับจากดาวเทียม โค้ดแทรคกิ้งลูปจะใช้คลื่นพาห์แทรคกิ้งลูปทั้งสองจะช่วยกันเพื่อที่จะได้รับและแทรคสัญญาณดาวเทียมดังภาพที่ 2.6

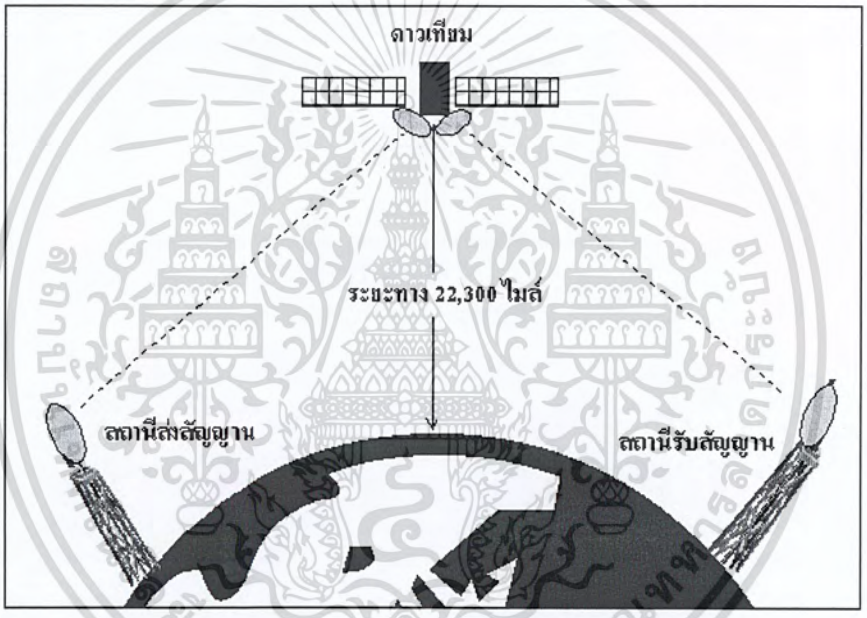


ภาพที่ 2.6 แสดง Block Diagram อย่างง่ายของเครื่องรับจีพีเอส

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.3 ย่านความถี่ในการส่งสัญญาณ

ดาวเทียมที่ใช้ในการสื่อสารจะมีอุปกรณ์การรับ-ส่งคลื่นวิทยุภายในตัวดาวเทียม และทำหน้าที่ถ่ายทอดทวนสัญญาณ (REPEATER) ไปยังสถานีภาคพื้นดินที่ส่งสัญญาณ และรับสัญญาณการส่งสัญญาณคลื่นความถี่ของคลื่นไมโครเวฟจากสถานีภาคพื้นดินที่ส่งสัญญาณขาขึ้น เรียกว่า "การเชื่อมโยงขาขึ้น" (Up-link) ซึ่งจานรับสัญญาณบนตัวดาวเทียมจะรับคลื่นสัญญาณข้อมูลภาพ, ข้อมูลเสียง และคอมพิวเตอร์ไว้ แล้วขยายให้มีความแรงของสัญญาณมากขึ้นจึงส่งคลื่นสัญญาณลงมายังสถานีภาคพื้นดินตามต้องการซึ่งการส่งสัญญาณจากดาวเทียมลงมายังสถานีภาคพื้นดิน เรียกว่า "การเชื่อมโยงขาลง" (Down – link) โดยความถี่คลื่นไมโครเวฟขาขึ้นจะแตกต่างกับความถี่ขาลง ซึ่งเป็นไปตามหลักของระบบการถ่ายทอดความถี่ และการหลีกเลี่ยงการรบกวนของสัญญาณ โดยความถี่ขาขึ้นจะสูงกว่าความถี่ขาลงจาก



ภาพที่ 2.7 การส่งและรับสัญญาณดาวเทียม

2.2.4 ระบบการส่งสัญญาณ

ระบบการส่งสัญญาณมี 2 แบบ ดังนี้

2.2.4.1 การส่งสัญญาณแบบ C - BAND

การส่งสัญญาณแบบ C-BAND จะส่งคลื่นความถี่กลับมายังโลก อยู่ในช่วงความถี่ 3.4 – 4.2 GHz แบบนี้จะมีฟุตปริ้นท์กว้างสามารถส่งสัญญาณครอบคลุมพื้นที่ได้หลายประเทศ

ข้อดี : การใช้ดาวเทียมประเภทนี้ เหมาะที่จะใช้ในประเทศที่ใหญ่ ๆ เพราะส่งดาวเทียมดวงเดียว ก็สามารถครอบคลุมพื้นที่ได้ทั่วประเทศ เช่น สหรัฐอเมริกา, รัสเซีย, จีน และอินโดนีเซีย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ข้อเสีย : เนื่องจากการส่งสัญญาณจะครอบคลุมพื้นที่กว้าง ความเข้มของสัญญาณจะต่ำจึงต้องใช้จาน 4 - 10 ฟุต ขนาดใหญ่รับสัญญาณภาพจึงจะชัด

2.2.4.2 การส่งสัญญาณแบบ KU - BAND

การส่งสัญญาณแบบ KU - BAND ส่งความถี่ในช่วง 10 - 12 GHz ซึ่งสูงกว่าความถี่แบบ C – BAND ส่งสัญญาณที่ครอบคลุมพื้นที่ได้น้อยจึงเหมาะสำหรับการส่งสัญญาณเฉพาะภายในประเทศ และเป็นที่นิยมใช้ในประเทศสหรัฐอเมริกาในยุโรป และประเทศญี่ปุ่น

ข้อดี : ความเข้มของสัญญาณสูงมากใช้จานขนาดเล็กๆ 80 – 120 เซนติเมตร เหมาะสำหรับส่งสัญญาณภายในประเทศ เช่น สัญญาณ CABLE TV ผ่านดาวเทียม

ข้อเสีย : ฟุตปริ้นท์ระบบ KU - BAND จะแคบส่งเฉพาะจุดที่ต้องการครอบคลุมพื้นที่ได้น้อยทำให้เสียค่าใช้จ่ายสูง

2.2.5 ตำแหน่งดาวเทียมและพื้นที่ติดตั้ง

2.2.5.1 ตำแหน่งของดาวเทียม

ภาพที่ 2.8 แสดงพื้นที่การให้บริการของดาวเทียม

การติดตั้งจานดาวเทียมจะต้องรู้ตำแหน่งอ้างอิงของดาวเทียม ดาวเทียมทุกดวงจะแขวนอยู่ที่เส้นศูนย์สูตร (เส้นแบ่งระหว่างซีกโลกเหนือและใต้) ซึ่งตำแหน่งดาวเทียมแต่ละดวงต้องได้รับอนุญาตจากองค์การดาวเทียมระหว่างประเทศหรือ IUT (Internation Telecommunication Union) และดาวเทียมแต่ละดวงจะมีตำแหน่งที่แตกต่างกันโดยใช้เส้นแวง (LONGTITUDE) เป็นตัวกำหนดตำแหน่ง ดังนั้นชื่อของดาวเทียมจะมีตัวเลขต่อท้ายเสมอ เช่น THAICOM 78.5 °E หมายถึง เส้นแวงที่ 78.5° และ E หมายถึง ซีกโลกด้านตะวันออก

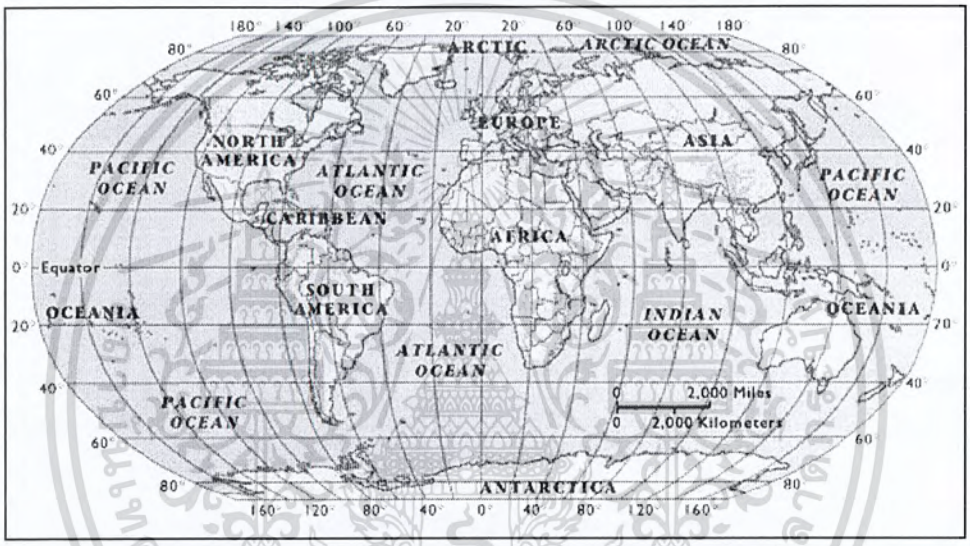
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

2.2.5.2. การกำหนดพื้นที่ในการติดตั้ง

เมื่อรู้ว่าดาวเทียมทุกดวงอยู่ที่เส้นศูนย์สูตร และประเทศไทยอยู่ที่ซีกโลกทางด้านเหนือ การติดตั้งจานจะต้องหันหน้าจานไปทางทิศใต้ส่วนจะหันหน้าจานไปทางทิศตะวันออกหรือตะวันตกนั้นขึ้นอยู่กับตำแหน่งที่ติดตั้งกับตำแหน่งของดาวเทียมกับตำแหน่งของดาวเทียม

การติดตั้งจานที่จังหวัดนราธิวาสและการติดตั้งจานที่จังหวัดเชียงรายจะมีมุมก้มเงยที่แตกต่างกันเพราะจังหวัดนราธิวาสอยู่ใกล้เส้นศูนย์สูตรจานจะเงยมากกว่า ส่วนจังหวัดเชียงรายอยู่ห่างจากเส้นศูนย์สูตรจานจะก้มมากกว่ามุมก้มเงย หรือกวาด ในแต่ละตำแหน่งจะขึ้นอยู่กับดาวเทียม ดังนั้นการติดตั้งจานดาวเทียมจะต้องทราบค่ามุมก้มเงย และมุมกวาดของแต่ละพื้นที่เพื่อรับสัญญาณดาวเทียม



ภาพที่ 2.9 แสดงตำแหน่งของประเทศต่างๆ ตามแนวเส้นรุ้งเส้นแวง

2.2.5.3 การคำนวณมุมก้มเงย (Elevation) และ มุมกวาด (Azimuth)

- การคำนวณมุมก้มเงย

$$EL = \tan^{-1}\left[\frac{\cos\theta\cos\emptyset - \frac{R}{R+H}}{\sqrt{1-cos^2\theta\, cos^2\emptyset}}\right] \quad (2.1)$$

- การคำนวณมุมกวาด

$$AZ = \tan^{-1}\left[\frac{\tan\emptyset}{\sin\theta}\right] \quad (2.2)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
119450

EL = ELEVATION มุมก้มเงย

AZ = AZIMUTH มุมส่าย

0 = LATITUDE เส้นรุ้งของสถานี

Ø = LONGTITUDE เส้นแวงของดาวเทียมลบด้วยเส้นแวงของสถานี

R = รัศมีของโลก 6,370 ก.ม.

H = ระยะความสูงของดาวเทียม 35,680 ก.ม.

2.3 ไมโครคอนโทรลเลอร์ PIC16F873A

ไมโครคอนโทรลเลอร์ในตระกูล PIC16F87xA เป็นไมโครคอนโทรลเลอร์ขนาด 8 บิต ที่มีอุปกรณ์สนับสนุนภายในหลายอย่าง ได้แก่ หน่วยความจำสำหรับเก็บข้อมูล หน่วยความจำสำหรับเก็บโปรแกรม ตัวตั้งเวลา/ตัวนับ อุปกรณ์สำหรับส่งข้อมูลแบบอนุกรม และเป็นชิปรุ่นแรกที่หน่วยความจำโปรแกรมเป็นแบบแฟลช และมีหน่วยความจำข้อมูลแบบ EEPROM ทำให้สามารถพัฒนาโปรแกรมได้ง่าย และสนับสนุนการทำงานแบบอินเซอร์กิตดีบักเกอร์ (In Circuit Debugger) ทำให้ไม่ต้องซื้ออีพรอมอีมูเลเตอร์ (EPROM Emulator) มีคำสั่งภาษาแอสเซมบลี 35 คำสั่ง และมีวงจรแปลงสัญญาณอะนาล็อกเป็นสัญญาณดิจิตอล (A/D) ขนาด 10 บิตอยู่ภายในด้วยเนื่องจากโครงสร้างของไมโครคอนโทรลเลอร์มีอุปกรณ์สนับสนุนประกอบอยู่ภายใน ซึ่งจะทำให้ใช้งานง่ายและมีประสิทธิภาพมากขึ้น โดยไม่ต้องทำการการเชื่อมต่ออุปกรณ์ภายนอกเพิ่มเติมมากเหมือนกับไมโครโปรเซสเซอร์ทั่วไป นอกจากนี้เรายังสามารถนำอุปกรณ์เสริมอื่นๆ มาต่อเพิ่มเติมเข้ากับไมโครคอนโทรลเลอร์ได้

2.3.1 โครงสร้างภายในของไมโครคอนโทรลเลอร์ PIC16F873A

โครงสร้างภายในพื้นฐานของไมโครคอนโทรลเลอร์ PIC16F873A แสดงในภาพที่ 2.10 ซึ่งมีคุณสมบัติดังนี้

2.3.1.1 คุณสมบัติหลัก

- มีคำสั่งที่เป็นภาษาแฮสแซมบลี 35 คำสั่ง
- ใน 1 คำสั่งใช้เวลาทำงาน 1 ถึง 2 ไซเคิล
- ทำงานได้สูงสุดที่สัญญาณนาฬิกาตั้งแต่ DC- 20 MHz
- ทำงานแบบ Pipe-line สามารถทำงาน 2 อย่างในเวลาเดียวกันได้
- หน่วยความจำโปรแกรมเป็นแบบ Flash มีขนาด 8 K Word (1 word=14 bit)
- มีหน่วยความจำข้อมูล (Data Memory RAM) ขนาด 192 ไบท์
- มีหน่วยความจำข้อมูลแบบ EEPROM ขนาด 128 ไบท์
- ตอบสนองแหล่งกำเนิดอินเทอร์รัปต์ได้ทั้งหมด 14 แหล่ง

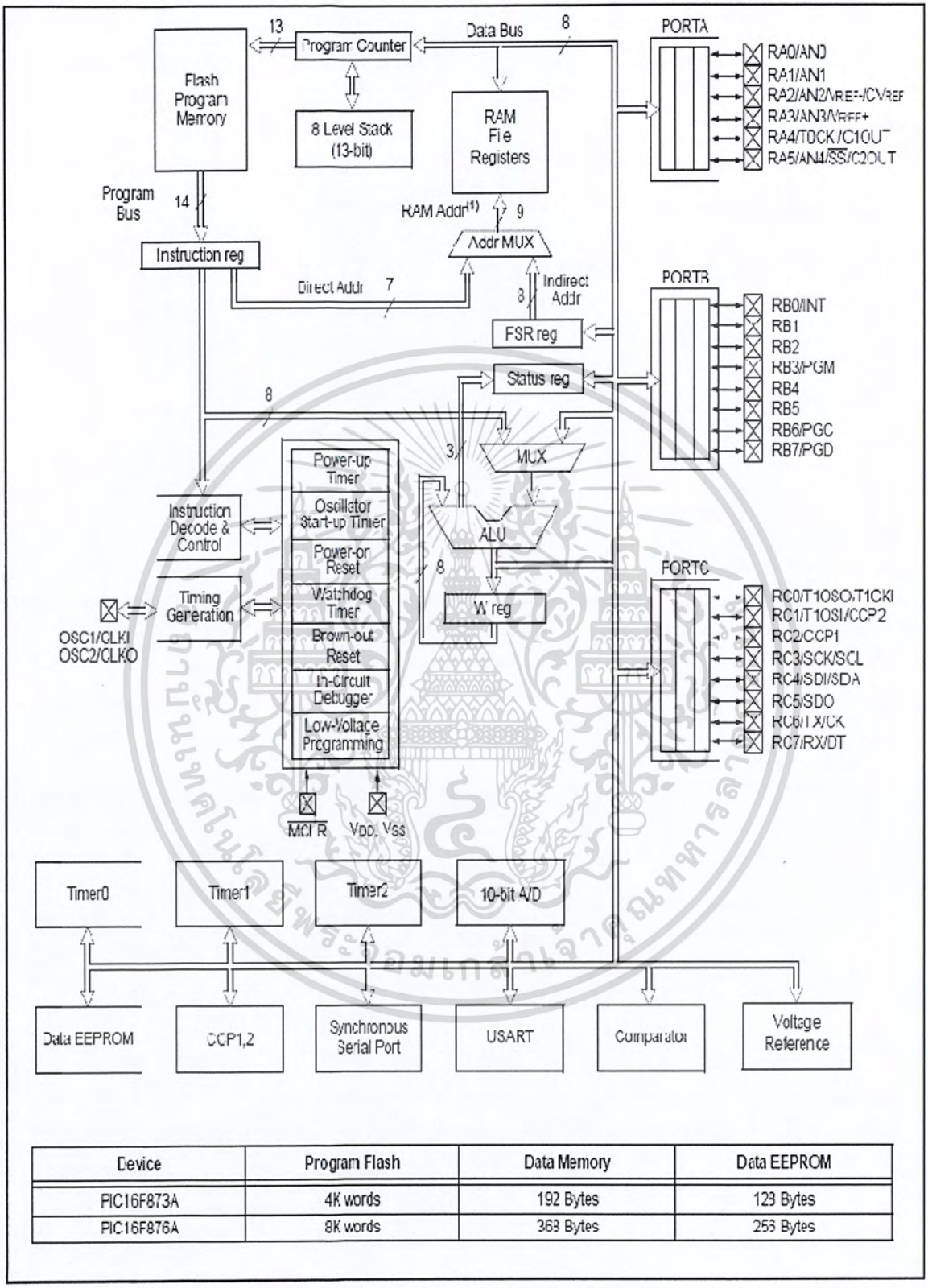
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- มี stack ให้ใช้งานได้สูงสุด 8 ระดับ
- มีวงจรเพาเวอร์ออนรีเซต (POR)
- มีเพาเวอร์อัปไทเมอร์ (PWRT) และออสซิเลเตอร์สตาร์อัปไทเมอร์ (OST)
- มีวงจรวอตช์ดือกไทเมอร์ (WDT) ที่มีวงจรออสซิเลเตอร์ในตัวซึ่งทำให้มีความเชื่อถือในการทำงานสูง
- มีระบบ Code Protection กันการคัดลอก
- มีโหมดประหยัดพลังงาน (Sleep Mode)
- สามารถโปรแกรมด้วยไฟ +5 VDC ได้
- ใช้การโปรแกรมแบบ In-Circuit Serial Programming
- ทำงานที่ไฟเลี้ยง 2VDC ถึง 5.5 VDC
- มี Current Sink และ Current Source อยู่ที่ 25 mA
- มี I/O พอร์ตทั้งหมด 5 พอร์ต แต่ละพอร์ตมีจำนวนบิตไม่เท่ากัน
- สามารถแก้ไขข้อมูลในหน่วยความจำโปรแกรมด้วยกระบวนการ ไอซีดี (ICD) ผ่านพอร์ตเพียง 2 ขา
- ซีพียูสามารถอ่านและเขียนหน่วยความจำโปรแกรมได้

2.3.1.2 คุณสมบัติพิเศษเพิ่มเติม

- มี Timer/Counter 3 ตัว คือ Timer 0 ขนาด 8 บิต, Timer 1 ขนาด 16 บิต และ Timer 2 ขนาด 8 บิต และรีจิสเตอร์คาบเวลา (period register) ขนาด 8 บิตในตัว
- มีส่วนเปรียบเทียบสัญญาณ (Compare) มีขนาด 16 บิต ความละเอียดสูงสุด 200 นาโนวินาที
- ส่วนตรวจจับสัญญาณหรือแคปเจอร์ (Capture) มีขนาด 16 บิต ความละเอียดสูงสุด 12.5 นาโนวินาที
- มีวงจร PWM (Pulse Width Modulation) ขนาด 10 บิต
- มีวงจรแปลงสัญญาณ Analog เป็น Digital (A/D Converter) ขนาด 10 บิต
- มีวงจรตรวจจับระดับแรงดันไฟเลี้ยงเพื่อรีเซตซีพียู หรือเรียกว่า บราวเอาต์รีเซต (Brown-out Reset: BOR)
- มีวงจรสื่อสารข้อมูลอนุกรม (USART) พร้อมการตรวจจับแอดเดรส 9 บิต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



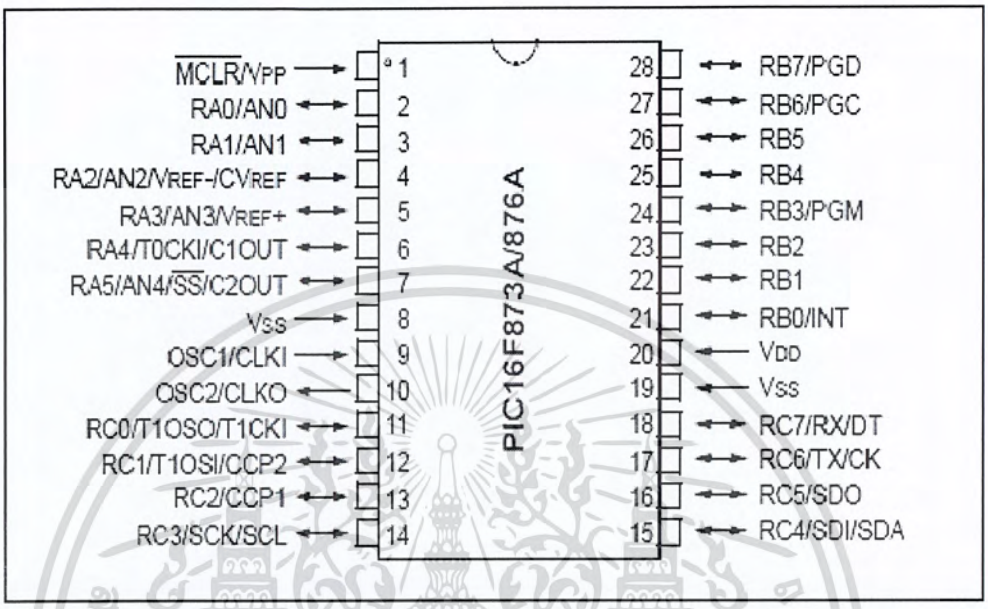
Device	Program Flash	Data Memory	Data EEPROM
PIC16F873A	4K words	192 Bytes	128 Bytes
PIC16F876A	8K words	368 Bytes	256 Bytes

ภาพที่ 2.10 โครงสร้างภายในของไมโครคอนโทรลเลอร์ PIC16F873A

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3.2 โครงสร้างภายนอกของไมโครคอนโทรลเลอร์ PIC16F873A

ไมโครคอนโทรลเลอร์ตระกูล PIC เบอร์ PIC16F873A เป็นไมโครคอนโทรลเลอร์ขนาด 28 ขา แสดงดังภาพที่ 2.11



ภาพที่ 2.11 แสดงขาสัญญาณของไมโครคอนโทรลเลอร์ PIC16F873A

2.3.2.1 หน้าที่และการใช้งานของสัญญาณต่างๆ

- MCLR/Vpp : Master Clear (Reset) Input/Programming Voltage Input ทำหน้าที่เป็นขาสัญญาณรีเซต เมื่อขานี้ได้รับลอจิก 0 ไมโครคอนโทรลเลอร์จะถูกรีเซต และทำหน้าที่เป็นขาสัญญาณรับแรงดันขณะทำการบันทึกโปรแกรมลงหน่วยความจำของไมโครคอนโทรลเลอร์
- VDD : Positive Supply (+2.00 V ถึง +5.5 V) ทำหน้าที่เป็นขาไฟเลี้ยงของไมโครคอนโทรลเลอร์
- VSS : Ground ทำหน้าที่เป็นขากราวด์
- OSCI/CLKIN : Oscillator Crystal Output/External Clock Source Input ทำหน้าที่เป็นขาสัญญาณอินพุตสำหรับต่อคริสตอล ในกรณีที่อยู่ในโหมดการใช้สัญญาณนาฬิกาจากภายนอก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

• RA0-RA5 : พอร์ต A มีจำนวน 6 ขา เป็นพอร์ตแบบสองทิศทาง (Bi-directional I/O Port) คือ เป็นได้ทั้งพอร์ตอินพุตและพอร์ตเอาต์พุตใช้ในการส่งและรับข้อมูล นอกจากนี้ยังทำหน้าที่อื่นๆ แสดงดังตารางที่ 2.2

ตารางที่ 2.2 หน้าที่อื่นๆ ของพอร์ต A

Pin Name	PDIP, SOIC, SSOP Pin#	QFN Pin#	I/O/P Type	Buffer Type	Description
					PORTA is a bidirectional I/O port.
RA0/AN0	2	27		TTL	
RA0			I/O		Digital I/O.
AN0			I		Analog input 0.
RA1/AN1	3	28		TTL	
RA1			I/O		Digital I/O.
AN1			I		Analog input 1.
RA2/AN2/VREF-/ CVREF	4	1		TTL	
			I/O		Digital I/O.
RA2			I		Analog input 2.
AN2			I		A/D reference voltage (Low) input.
VREF-			O		Comparator VREF output.
CVREF					
RA3/AN3/VREF+	5	2		TTL	
RA3			I/O		Digital I/O.
AN3			I		Analog input 3.
VREF+			I		A/D reference voltage (High) input.
RA4/T0CKI/C1OUT	6	3		ST	
RA4			I/O		Digital I/O – Open-drain when configured as output.
T0CKI			I		Timer0 external clock input.
C1OUT			O		Comparator 1 output.
RA5/AN4/$\overline{SS}$/C2OUT	7	4		TTL	
RA5			I/O		Digital I/O.
AN4			I		Analog input 4.
$\overline{SS}$			I		SPI slave select input.
C2OUT			O		Comparator 2 output.

• RB0-RB7 : พอร์ต B มีจำนวน 8 ขา ขนาด 8 บิต เป็นพอร์ตแบบสองทิศทาง ใช้ในการส่งและรับข้อมูล นอกจากนี้บางขายังทำหน้าที่รับสัญญาณอินพุตจากการอินเทอร์รัปต์ (Interrupt) ภายนอกด้วย แสดงดังตารางที่ 2.3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

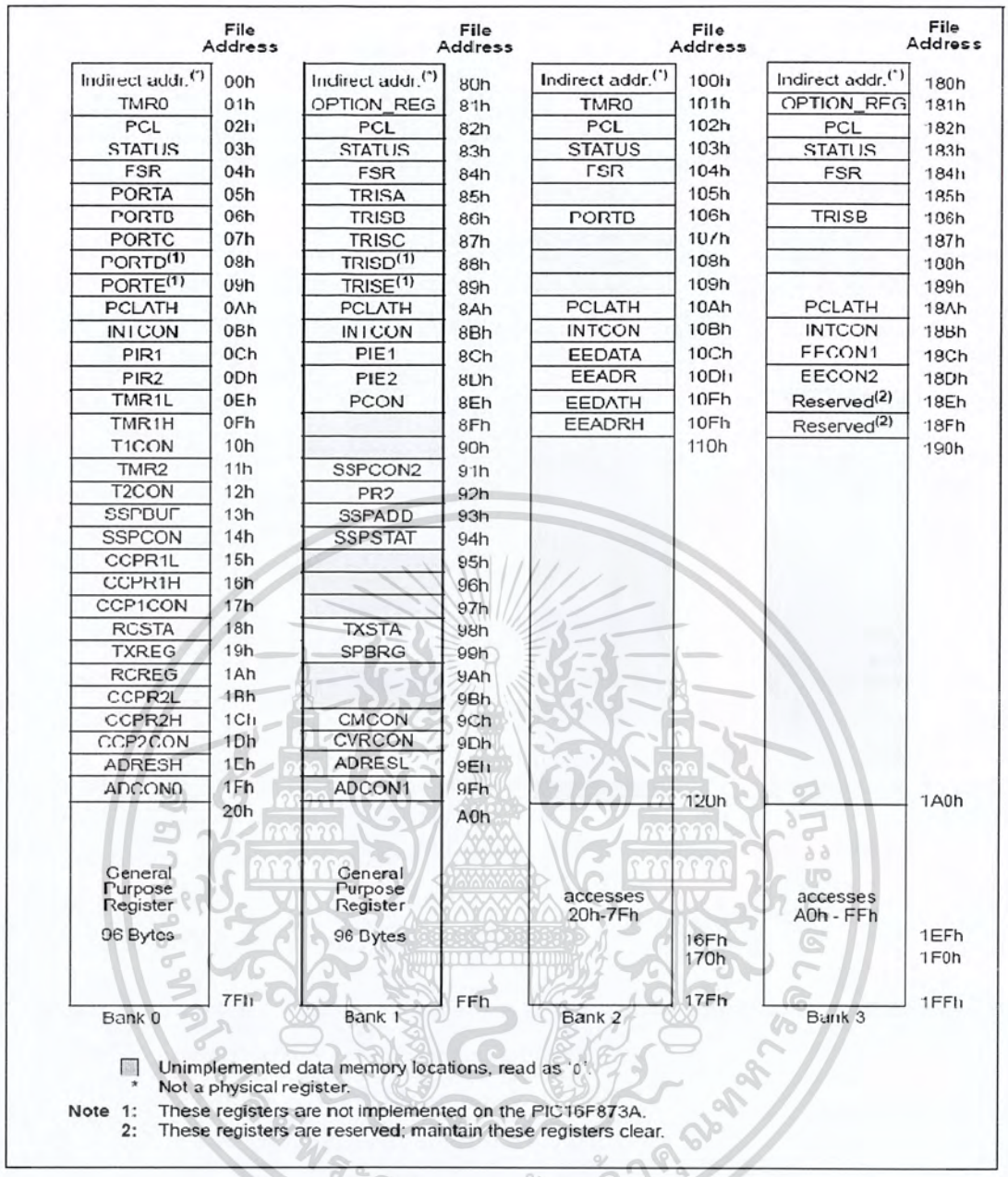
ตารางที่ 2.3 หน้าที่อื่นๆ ของพอร์ต B และ C

Pin Name	PDIP, SOIC, SSOP Pin#	QFN Pin#	I/O/P Type	Buffer Type	Description
					PORTB is a bidirectional I/O port. PORTB can be software programmed for internal weak pull-ups on all inputs.
RB0/INT	21	18		TTL/ST(1)	
RB0			I/O		Digital I/O.
INT			I		External interrupt.
RB1	22	19	I/O	TTL	Digital I/O.
RB2	23	20	I/O	TTL	Digital I/O.
RB3/PGM	24	21		TTL	
RB3			I/O		Digital I/O.
PGM			I		Low-voltage (single-supply) ICSP programming enable pin.
RB4	25	22	I/O	TTL	Digital I/O.
RB5	26	23	I/O	TTL	Digital I/O.
RB6/PGC	27	24		TTL/ST(2)	
RB6			I/O		Digital I/O.
PGC			I		In-circuit debugger and ICSP programming clock.
RB7/PGD	28	25		TTL/ST(2)	
RB7			I/O		Digital I/O.
PGD			I/O		In-circuit debugger and ICSP programming data.
					PORTC is a bidirectional I/O port.
RC0/T1OSO/T1CKI	11	8		ST	
RC0			I/O		Digital I/O.
T1OSO			O		Timer1 oscillator output.
T1CKI			I		Timer1 external clock input.
RC1/T1OSI/CCP2	12	9		ST	
RC1			I/O		Digital I/O.
T1OSI			I		Timer1 oscillator input.
CCP2			I/O		Capture2 input, Compare2 output, PWM2 output.
RC2/CCP1	13	10		ST	
RC2			I/O		Digital I/O.
CCP1			I/O		Capture1 input, Compare1 output, PWM1 output.
RC3/SCK/SCL	14	11		ST	
RC3			I/O		Digital I/O.
SCK			I/O		Synchronous serial clock input/output for SPI mode.
SCL			I/O		Synchronous serial clock input/output for I^2C mode.
RC4/SDI/SDA	15	12		ST	
RC4			I/O		Digital I/O.
SDI			I		SPI data in.
SDA			I/O		I^2C data I/O.
RC5/SDO	16	13		ST	
RC5			I/O		Digital I/O.
SDO			O		SPI data out.
RC6/TX/CK	17	14		ST	
RC6			I/O		Digital I/O.
TX			O		USART asynchronous transmit.
CK			I/O		USART1 synchronous clock.
RC7/RX/DT	18	15		ST	
RC7			I/O		Digital I/O.
RX			I		USART asynchronous receive.
DT			I/O		USART synchronous data.

2.3.2.2 การจัดการหน่วยความจำของ PIC16F873A

ไมโครคอนโทรลเลอร์ PIC16F873A มีหน่วยความจำโปรแกรมแบบแฟลช ซึ่งใช้สำหรับเก็บโปรแกรมควบคุมการทำงานของ PIC มีหน่วยความจำข้อมูล (Data Memory RAM) ขนาด 192 ไบต์ หน่วยความจำข้อมูลแบบอีอีพรอม (Data Memory EEPROM) ขนาด 128 ไบต์ และได้มีการจัดการหน่วยความจำของ PIC ออกเป็นแบงก์ต่างๆ ได้ทั้งหมด 4 แบงค์ คือ Bank0-Bank3 และแต่ละแบงก์มีขนาด 128 ไบต์ ดังภาพที่ 2.15

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 2.12 การจัดพื้นที่หน่วยความจำใน PIC16F873A

Bank 0-Bank 1 จะแบ่งออกเป็น 2 ส่วนเหมือนกันคือ ส่วนที่ 1 ขนาด 32 ไบต์ เป็นพื้นที่ของรีจิสเตอร์ไฟล์ที่เรียกว่า SFR (Special Function Register) ซึ่งใช้สำหรับกำหนดเงื่อนไขการทำงานและบันทึกสถานการณ์ทำงานของไมโครคอนโทรเลอร์ ส่วนที่ 2 ขนาด 96 ไบต์เป็นพื้นที่หน่วยความจำสำหรับใช้งานทั่วไป ซึ่งใช้สำหรับเก็บผลลัพธ์และเงื่อนไขต่างๆ ของโปรแกรม

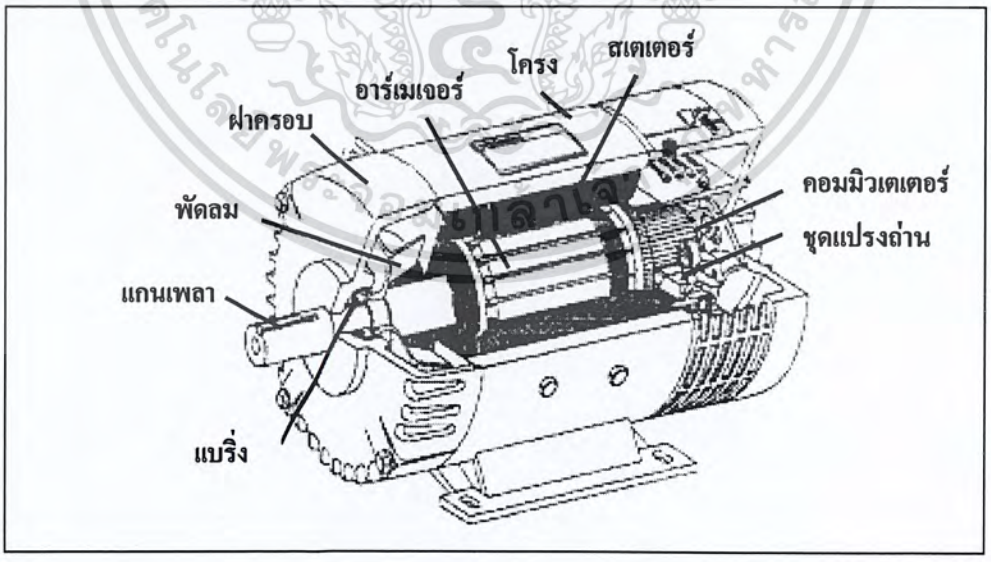
Bank 2-Bank 3 จะแบ่งออกเป็นสองส่วนแต่มีขนาดแตกต่างกับ Bank 0-Bamk 1 คือ ส่วนที่ 1 ขนาด 16 ไบต์ เป็นพื้นที่ของรีจิสเตอร์ไฟล์ ส่วนที่ 2 ขนาด 112 ไบต์ เป็นพื้นที่หน่วยความจำสำหรับใช้งานทั่วไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4 มอเตอร์กระแสตรง (DC MOTOR)

โครงสร้างมอเตอร์ไฟฟ้ากระแสตรง หรือดีซีมอเตอร์ จะประกอบด้วยคอมมิวเตเตอร์ (COMMUTATOR) และแปรงถ่าน (BRUSH) ซึ่งในการใช้งานของมอเตอร์ชนิดนี้ จะมีการสึกหรอเกิดขึ้นระหว่างหน้าสัมผัสของซี่คอมมิวเตเตอร์กับแปรงถ่าน จึงจำเป็นที่ต้องมีการทำความสะอาด ซ่อมแซมและบำรุงรักษาอยู่เสมอ ปกติแล้วมอเตอร์ชนิดนี้เหมาะสำหรับการใช้งานในลักษณะงานที่มีระบบกาควบคุมและตรวจสอบแบบย้อนกลับ (FEED BACK CONTROL) หรือในลักษณะของเซอร์โวมอเตอร์ มอเตอร์กระแสตรงจะมีหลักการทำงานโดยวิธีการผ่านกระแสให้กับขดลวดในสนามแม่เหล็ก ซึ่งจะทำให้เกิดแรงแม่เหล็กโดยส่วนของแรงนี้จะขึ้นอยู่กับกระแส และกำลังของสนามแม่เหล็ก

หลักการทำงานของมอเตอร์ทางเดินของฟลั๊กซ์แม่เหล็ก และสนามแม่เหล็กจะเกิดจากแท่งแม่เหล็กเฟอร์ไรต์ 2 ชิ้นที่ขึ้นรูปเป็นแบบโค้งยึดติดกับตัวถังได้พอดี เพื่อที่จะให้เส้นแรงแม่เหล็กวิ่งเข้าสู่กลางมอเตอร์ได้ ดังนั้นความเข้มของแม่เหล็กจะขึ้นอยู่กับขนาดความหนาของแม่เหล็กซึ่งจะส่งผลให้ฟลั๊กซ์แม่เหล็กวิ่งไปบนตัวถังโลหะ กระแสไฟฟ้าในขดลวดที่พันกับทุ่นโรเตอร์จะทำให้เกิดสนามแม่เหล็กไฟฟ้า และต้านกับสนามแม่เหล็กถาวรซึ่งจะส่งผลให้เกิดเป็นแรงบิดเพื่อที่จะหมุนทุ่นโรเตอร์ให้ไปในทิศทางเดียวกันกับทิศทางของสนามแม่เหล็กที่มีแรงมากกว่า กระแสก็จะไหลผ่านไปยังทุ่นโรเตอร์โดยผ่านแปรงถ่านซึ่งจะสัมผัสกับแหวนตัวนำในทุ่นโรเตอร์ และแหวนคอมมิวเตเตอร์ ซึ่งจะถูกแบ่งออกเป็น 3 เซกเมนต์เพื่อที่จะทำหน้าที่นำกระแสเข้าขดลวด



ภาพที่ 2.13 โครงสร้างของมอเตอร์ไฟฟ้ากระแสตรง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4.1 ส่วนประกอบของมอเตอร์ไฟฟ้ากระแสตรง

มอเตอร์ไฟฟ้ากระแสตรงมีโครงสร้าง และส่วนประกอบคล้ายกับเครื่องกำเนิดไฟฟ้ากระแสตรง มีส่วนประกอบที่สำคัญเหมือนกันมีรูปร่างลักษณะภายนอกคล้ายกันแตกต่างกันตรงที่การนำไปใช้งาน โดยเครื่องกำเนิดไฟฟ้ากระแสตรงนั้นจะทำให้เกิดไฟฟ้าในรูปของแรงดันไฟตรงออกมา ส่วนมอเตอร์ไฟฟ้ากระแสตรงเมื่อจ่ายแรงดันไฟตรงให้มอเตอร์ทำให้มอเตอร์หมุนเกิดพลังกลขึ้นมา

2.4.1.1 ส่วนประกอบหลักของมอเตอร์ไฟฟ้ากระแสตรง

- **ขดลวดสนามแม่เหล็ก (Field Coil)** คือ ขดลวดที่ถูกพันอยู่กับขั้วแม่เหล็กที่ยึดติดกับโครงมอเตอร์ ทำหน้าที่กำเนิดขั้วแม่เหล็กขั้วเหนือ (N) และขั้วใต้ (S) แทนแม่เหล็กถาวร ขดลวดที่ใช้เป็นขดลวดอาบน้ำยาฉนวน สนามแม่เหล็กจะเกิดขึ้นเมื่อจ่ายแรงดันไฟตรงให้มอเตอร์

- **ขั้วแม่เหล็ก (Pole Pieces)** คือแกนสำหรับรองรับขดลวดสนามแม่เหล็กถูกยึดติดกับโครงมอเตอร์ด้านใน ขั้วแม่เหล็กทำมาจากแผ่นเหล็กอ่อนบางๆ อัดซ้อนกัน (Lamination Sheet) เพื่อลดการเกิดกระแสไหลวน (Edgy Current) ที่จะทำให้ความเข้มของสนามแม่เหล็กลดลง ขั้วแม่เหล็กจะทำหน้าที่ให้กำเนิดขั้วสนามแม่เหล็กมีความเข้มสูงสุดเพื่อแทนขั้วสนามแม่เหล็กถาวร ผิวด้านหน้าของขั้วแม่เหล็กทำให้โค้งรับกับอาร์เมเจอร์พอดี

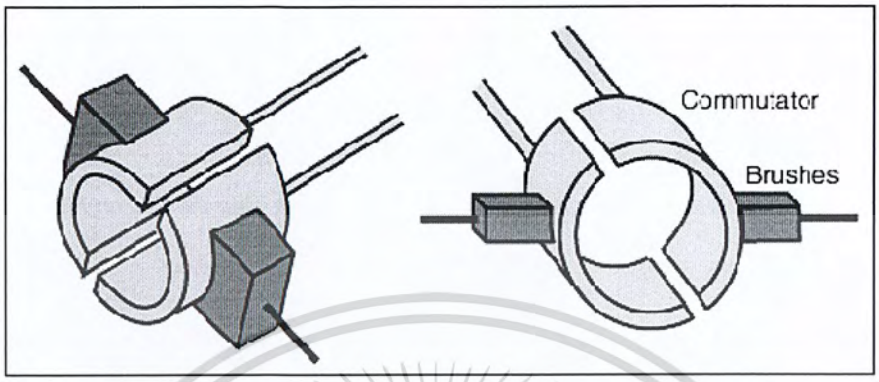
- **โครงมอเตอร์ (Motor Frame)** คือ ส่วนเปลือกหุ้มภายนอกของมอเตอร์ และยึดส่วนอยู่กับที่ (Stator) ของมอเตอร์ไว้ภายในร่วมกับฝาปิดหัวท้ายของมอเตอร์ โครงมอเตอร์ทำหน้าที่เป็นทางเดินของเส้นแรงแม่เหล็กระหว่างขั้วแม่เหล็กให้เกิดสนามแม่เหล็กครบวงจร

- **อาร์เมเจอร์ (Armature)** คือ ส่วนเคลื่อนที่ (Rotor) ที่ถูกยึดติดกับเพลา (Shaft) และรองรับการหมุนด้วยที่รองรับการหมุน (Bearing) ตัวอาร์เมเจอร์ทำจากเหล็กแผ่นบางๆ อัดซ้อนกันถูกเซาะร่องออกเป็นส่วนๆ เพื่อไว้พันขดลวดอาร์เมเจอร์ (Armature Winding) ขดลวดอาร์เมเจอร์เป็นขดลวดอาบน้ำยาฉนวน ส่วนของร่องขดลวดอาร์เมเจอร์จะมีขดลวดพันอยู่ และมีลิ่มไฟเบอร์อัดแน่นยึดขดลวดอาร์เมเจอร์ไว้ ในส่วนปลายขดลวดอาร์เมเจอร์ต่อไว้กับคอมมิวเตเตอร์ อาร์เมอเจอร์จะผลักดันของสนามแม่เหล็กทั้งสอง ทำให้อาร์เมเจอร์หมุนเคลื่อนที่

- **คอมมิวเตเตอร์ (Commutator)** คือ ส่วนเคลื่อนที่อีกส่วนหนึ่งถูกยึดติดเข้ากับอาร์เมเจอร์และเพลาร่วมกัน คอมมิวเตเตอร์ทำมาจากแท่งทองแดงแข็งประกอบเข้าด้วยกันเป็นรูปทรงกระบอก แต่ละแท่งทองแดงของคอมมิวเตเตอร์ถูกแยกออกจากกันด้วยฉนวนไมก้าอาร์เมเจอร์ คอมมิวเตเตอร์ทำหน้าที่เป็นขั้วรับแรงดันไฟตรงที่จ่ายมาจากแปรงถ่านเพื่อส่งไปให้ขดลวดอาร์เมอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- **แปรงถ่าน (Brush)** คือ ตัวสัมผัสกับคอมมิวเตเตอร์ ทำเป็นแท่งสี่เหลี่ยมผลิตจากคาร์บอนหรือแกรไฟต์ผสมผงทองแดง เพื่อให้แข็งและนำไฟฟ้าได้ดีมีสายตัวนำต่อร่วมกับแปรงถ่าน เพื่อไปรับแรงดันไฟตรงที่จ่ายเข้ามา แปรงถ่านทำหน้าที่รับแรงดันไฟตรงจากแหล่งจ่าย จ่ายผ่านไปให้คอมมิวเตเตอร์

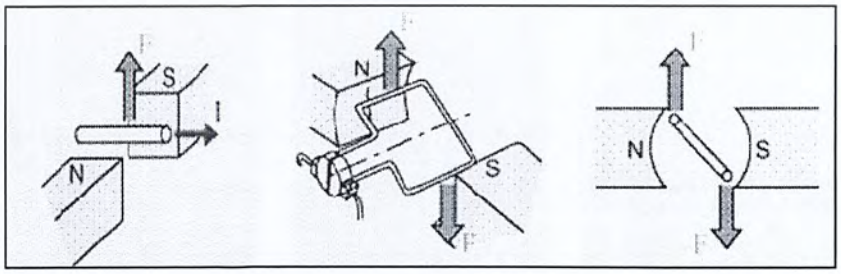


ภาพที่ 2.14 แสดงวงแหวนคอมมิวเตเตอร์ และ แปรงถ่าน

2.4.2 การทำงานของมอเตอร์ไฟฟ้า

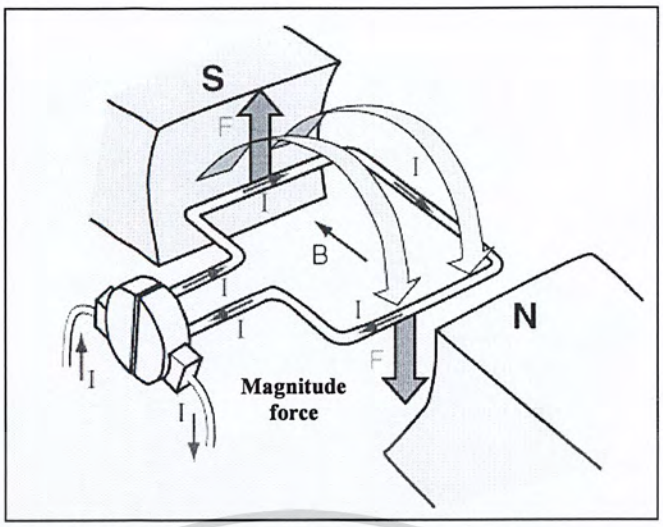
มอเตอร์ไฟฟ้าที่ถูกผลิตขึ้นมาใช้งานแบ่งออกได้เป็น 2 ประเภท คือ มอเตอร์ไฟฟ้ากระแสตรง (DC Motor) เป็นมอเตอร์ที่ต้องใช้กับแหล่งจ่ายไฟฟ้ากระแสตรง (DC Source) เป็นมอเตอร์แบบเบื้องต้นที่ถูกผลิตมาใช้งาน และมอเตอร์ไฟฟ้ากระแสสลับ (AC Motor) เป็นมอเตอร์ที่ต้องใช้กับแหล่งจ่ายไฟฟ้ากระแสสลับ (AC Source) มอเตอร์ชนิดนี้ถูกพัฒนามาจากมอเตอร์กระแสตรง เพื่อให้สามารถใช้งานได้อย่างกว้างขวางมากขึ้น

มอเตอร์ไฟฟ้ากระแสตรงประกอบด้วยแม่เหล็กถาวร 2 ขั้ววางอยู่ระหว่างขดลวดตัวนำ ขดลวดตำนำจะได้รับแรงดันไฟตรงป้อนให้ในการทำงานทำให้เกิดอำนาจแม่เหล็ก 2 ชุด ซึ่งจะมีขั้วแม่เหล็กเหมือนกันวางใกล้กันเกิดแรงผลักดันทำให้ขดลวดตัวนำหมุนเคลื่อนที่ได้ การทำงานเบื้องต้นของมอเตอร์ไฟฟ้ากระแสตรง แสดงดังภาพที่ 2.15



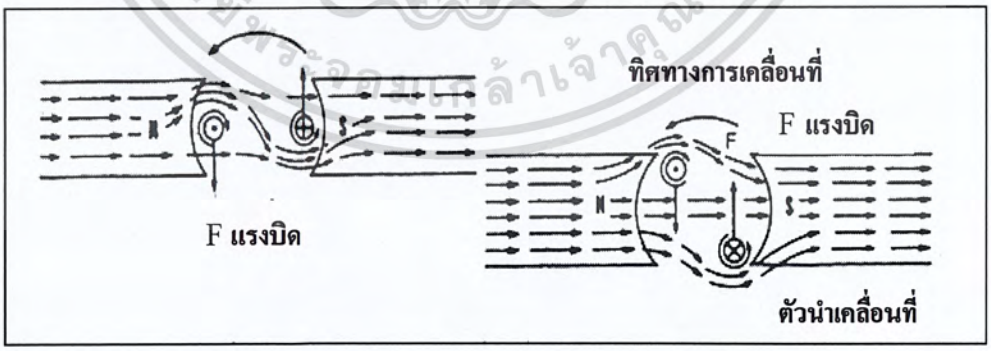
ภาพที่ 2.15 แสดงการทำงานของมอเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 2.16 แสดงการทำงานของมอเตอร์

จากภาพที่ 2.15 และภาพที่ 2.16 เป็นการทำงานเบื้องต้นของมอเตอร์ไฟฟ้ากระแสตรง มีแรงดันไฟตรงจ่ายผ่านแปรงถ่านไปคอมมิวเตเตอร์ และผ่านไปให้ขดลวดตัวนำที่อาร์เมเจอร์ซึ่งส่งผลให้ขดลวดอาร์เมเจอร์เกิดสนามแม่เหล็กไฟฟ้าขึ้นมาทางด้านซ้ายมือเป็นขั้วเหนือ (N) และด้านขวาเป็นขั้วใต้ (S) เหมือนกับขั้วแม่เหล็กถาวรที่วางอยู่ใกล้ๆ เกิดอำนาจแม่เหล็กผลักดันกันอาร์เมเจอร์หมุนไปในทิศทางตามเข็มนาฬิกาพร้อมกับคอมมิวเตเตอร์หมุนตามไปด้วย แปรงถ่านสัมผัสกับส่วนของคอมมิวเตเตอร์เปลี่ยนไปในอีกปลายหนึ่งของขดลวด มีผลทำให้เกิดขั้วแม่เหล็กที่อาร์เมเจอร์เหมือนกับขั้วแม่เหล็กถาวรที่อยู่ใกล้ๆ อีกครั้งทำให้อาร์เมเจอร์ยังคงถูกผลักให้หมุนไปในทิศทางตามเข็มนาฬิกาตลอดเวลาเกิดการหมุนของอาร์เมเจอร์คือมอเตอร์ไฟฟ้าทำงาน



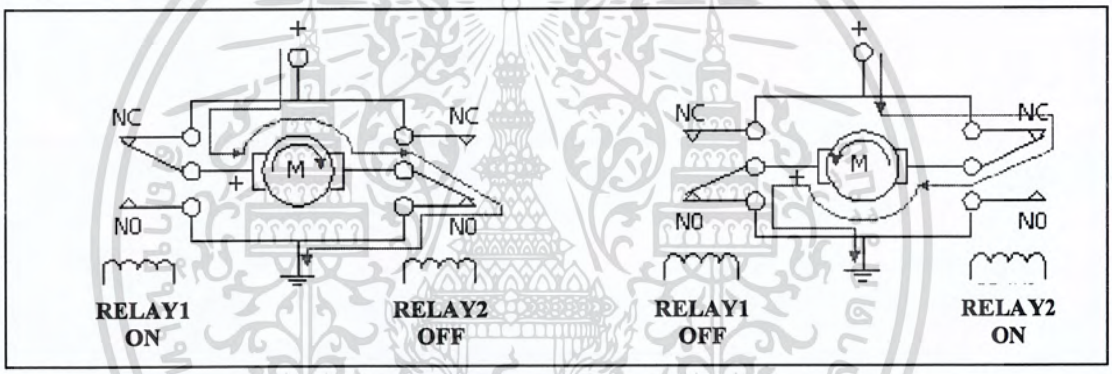
ภาพที่ 2.17 ทิศทางการเคลื่อนที่ของอาร์เมเจอร์ (โรเตอร์)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

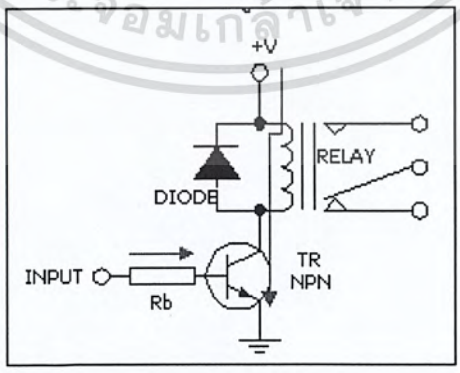
2.4.3 การขับและกลับทิศทางของมอเตอร์กระแสตรง (DC MOTOR)

ในการใช้ไอซีไมโครคอนโทรเลอร์เป็นตัวควบคุมการหมุน และทิศทางของมอเตอร์กระแสตรงนั้นจะต้องมีส่วนของวงจรที่เรียกว่าวงจรขับมอเตอร์ (Driver) ในส่วนของวงจรกลับทิศทางของมอเตอร์นั้น สามารถที่จะใช้รีเลย์ต่อวงจรสวิตช์เพื่อกลับทิศทางของขั้วไฟกระแสตรง หรืออาจใช้อุปกรณ์สารกึ่งตัวนำที่เป็นวงจรขับกำลัง เช่น ทรานซิสเตอร์ มอสเฟต แล้วแต่วิธีที่เราจะเลือกใช้งาน

จากภาพเป็นการใช้รีเลย์ควบคุมการเปลี่ยนทิศทางการหมุนของมอเตอร์ โดยการควบคุมการปิด - เปิดที่รีเลย์ 2 ตัว ซึ่งจะทำหน้าที่กลับทิศทางของขั้วไฟที่ป้อนให้กับมอเตอร์โดยการสลับการทำงานของรีเลย์ เช่น ให้รีเลย์ตัวที่ 1 ทำงาน (ON) และรีเลย์ตัวที่ 2 หยุดทำงาน (OFF) จะทำให้มอเตอร์หมุนไปทางซ้าย และในทำนองเดียวกันถ้าหากรีเลย์ตัวที่ 1 หยุดทำงาน (OFF) และรีเลย์ตัวที่ 2 ทำงาน (ON) ก็จะทำให้มอเตอร์หมุนไปทางขวา



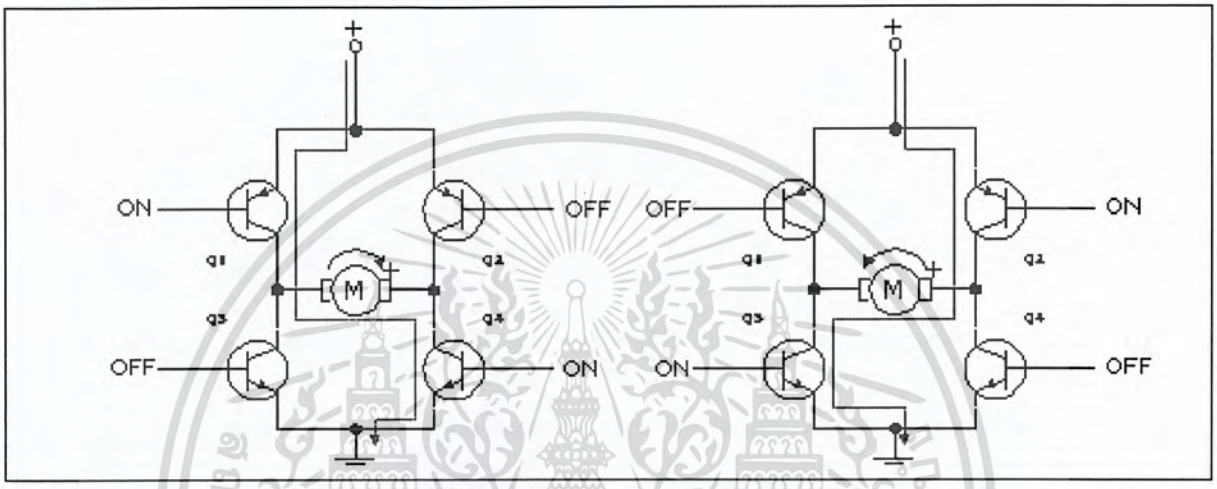
ภาพที่ 2.18 แสดงการกลับทิศทางของมอเตอร์กระแสตรงโดยใช้รีเลย์



ภาพที่ 2.19 แสดงการใช้ทรานซิสเตอร์เพื่อขับรีเลย์ให้ทำงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากภาพเป็นวงจรขับรีเลย์โดยใช้ทรานซิสเตอร์ทำหน้าที่ขยายกระแส เพราะไม่สามารถจะใช้ขาเอาต์พุตของไมโครคอนโทรลเลอร์ป้อนกระแสไฟที่ขดลวดของรีเลย์โดยตรงได้ เนื่องจากว่ากระแสที่จ่ายออกมาจากขาเอาต์พุตของไมโครคอนโทรลเลอร์มีค่าน้อยเกินไป ดังนั้นจึงต้องมีส่วนของวงจรทรานซิสเตอร์ เพื่อที่จะทำการขยายกระแสให้เพียงพอในการป้อนให้กับขดลวดของรีเลย์ ส่วนไดโอดนำมาต่อไว้สำหรับป้องกันแรงดันย้อนกลับที่เกิดจากการเหนี่ยวนำของสนามแม่เหล็กในขณะเกิดการยุบตัว ซึ่งอาจจะทำให้ทรานซิสเตอร์เสียหาย



ภาพที่ 2.20 แสดงการใช้ทรานซิสเตอร์เป็นวงจรขับและกำหนดทิศทางของมอเตอร์กระแสตรง

จากภาพเป็นวงจรลิเนียร์บริดจ์แอมป์ ซึ่งจะประกอบไปด้วยทรานซิสเตอร์กำลัง 4 ตัวที่ทำหน้าที่ขับ และควบคุมทิศทางการหมุนของมอเตอร์ หากกำหนดให้ทรานซิสเตอร์ Q1 และ Q4 อยู่ในสภาวะทำงาน (Active) กระแสไฟฟ้าจะไหลผ่านทรานซิสเตอร์จากซ้ายไปขวา โดยผ่านมอเตอร์ กระแสตรงทำให้มอเตอร์หมุนไปทางขวา ในทำนองเดียวกันถ้าหากเราทำให้ทรานซิสเตอร์ Q2 และ Q3 อยู่ในสภาวะทำงาน (Active) กระแสไฟฟ้าก็จะไหลจากทางขวาไปทางซ้ายซึ่งจะส่งผลให้มอเตอร์กลับทิศทางการหมุนจากทางขวาไปทางซ้าย

2.4.4 การควบคุมความเร็วของมอเตอร์กระแสตรง

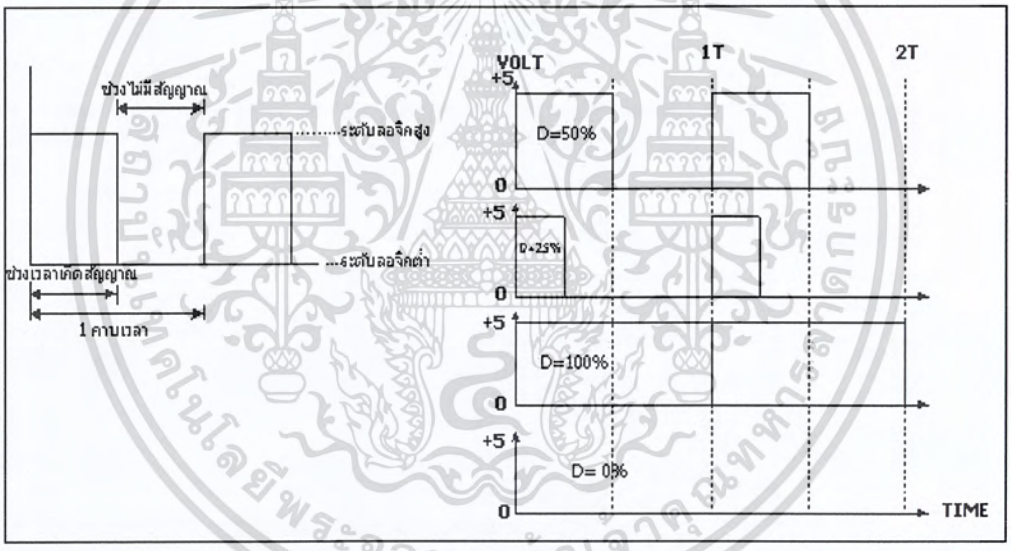
การควบคุมความเร็วของมอเตอร์กระแสตรงมีหลายวิธีด้วยกัน ซึ่งอาจจะใช้วิธีการควบคุมแบบพื้นฐานทั่วไป เช่น การควบคุมด้วยวิธีการใช้ตัวต้านทานปรับค่าโดยต่ออนุกรมกับมอเตอร์ หรือใช้วิธีการการควบคุมโดยการเปลี่ยนค่าของระดับแรงดันที่ป้อนให้กับมอเตอร์ แต่การควบคุมในวิธีดังกล่าวถึงแม้ว่าจะควบคุมความเร็วมอเตอร์ให้คงที่ได้ แต่ที่ความเร็วต่ำจะส่งผลให้แรงบิดต่ำไปด้วย ดังนั้นจึงเลือกใช้วิธีการควบคุมโดยการจ่ายกระแสไฟให้กับมอเตอร์เป็นช่วงๆ โดยอาศัยกระแสไฟที่ป้อนให้กับมอเตอร์ให้เป็นค่าเฉลี่ยที่เกิดขึ้นในแต่ละช่วง ซึ่งเราเรียกว่าวิธีการของการมอดูเลชั่นทางความกว้างของพัลส์ PWM (Pulse Width Modulation)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4.4.1 วิธีการมอดูเลชั่นทางความกว้างของพัลส์ (PWM)

การมอดูเลชั่นทางความกว้างของพัลส์ PWM (Pulse Width Modulation) เป็นการปรับเปลี่ยนที่สัดส่วนและความกว้างของสัญญาณพัลส์ โดยความถี่ของสัญญาณพัลส์จะไม่มีการเปลี่ยนแปลง หรือเป็นการเปลี่ยนแปลงที่ค่าของดิวตี้ไซเกิล (duty cycle) ซึ่งค่าของดิวตี้ไซเคิล คือ ช่วงความกว้างของพัลส์ที่มีสถานะลอจิกสูง โดยคิดสัดส่วนเป็นเปอร์เซ็นต์จากความกว้างของพัลส์ทั้งหมด ยกตัวอย่างเช่น ถ้าหากค่าดิวตี้ไซเคิลมีค่าเท่ากับเท่ากับ 50% ก็หมายถึง ใน 1 รูปสัญญาณพัลส์จะมีช่วงของสัญญาณที่เป็นสถานะลอจิกสูงอยู่ครึ่งหนึ่ง และสถานะลอจิกต่ำอยู่อีกครึ่งหนึ่ง และในทำนองเดียวกันถ้าหากค่าดิวตี้ไซเคิลมีค่ามาก หมายความว่า ความกว้างของพัลส์ที่เป็นสถานะลอจิกสูงจะมีความกว้างมากขึ้น หากค่าดิวตี้ไซเคิลมีค่าเท่ากับ 100% ก็หมายความว่าจะไม่มีสถานะลอจิกต่ำเลย ซึ่งค่าดิวตี้ไซเคิลสามารถ จะหาได้จากค่าความสัมพันธ์ดังนี้

ค่าดิวตี้ไซเคิล = (ช่วงของสัญญาณพัลส์/คาบเวลาทั้งหมดของสัญญาณ) X100%



ภาพที่ 2.21 แสดงความกว้างของพัลท์ขนาด

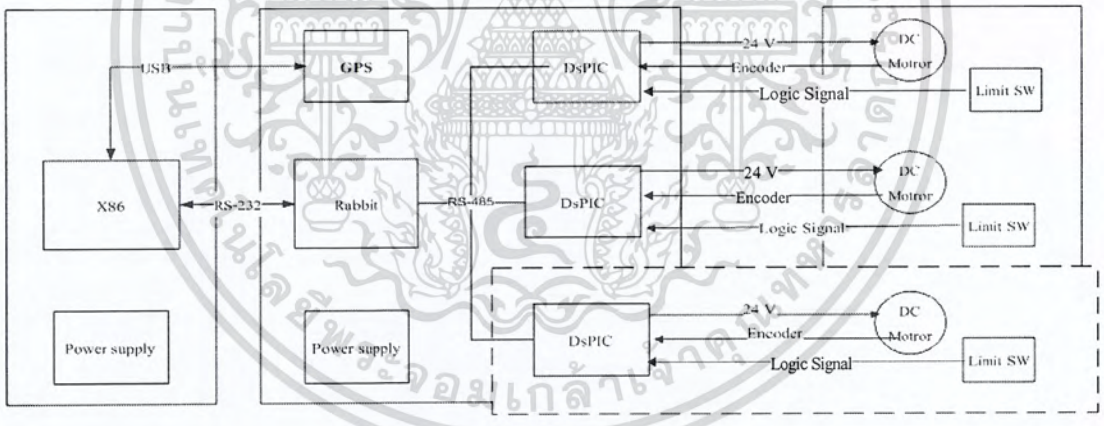
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3
วิธีการดำเนินงาน

3.1 โครงสร้าง Mobile VSAT Vehicle (MVV)

" IPSTAR MVV " คือ IPSTAR Mobile VSAT Vehicle ซึ่งหมายถึง ยานพาหนะที่มีระบบและอุปกรณ์ ซึ่งสามารถใช้งานกับดาวเทียม IPSTAR และ Application ของ IPSTAR ได้ ซึ่งรถ MVV ได้ถูกออกแบบมาเพื่อให้สามารถใช้งานได้อย่างรวดเร็ว สมบุกสมบัน สอดคล้องกับสถานการณ์ที่ต้องอาศัยความรวดเร็วในการปฏิบัติงาน เช่น เหตุการณ์ภัยพิบัติ เหตุภัยธรรมชาติ การรายงานข่าวสด และจะเน้นใช้ในด้านสื่อการเรียนการสอน หรือในพื้นที่ห่างไกล และทุรกันดาร สามารถใช้ Application ต่างๆ ที่ใช้งานในระบบ IPSTAR ได้เช่น Internet, WiFi, VOIP, Video Conference, Digital SNG, Nano BTS สามารถใช้แหล่งจ่ายไฟ จากตัวรถเอง จาก DC-AC Inverter , จาก AC Generator หรือจากไฟฟ้าตามบ้านเรือนทั่วไป ซึ่งสามารถเปิดโลกกว้างแห่งการสื่อสารได้ทุกที่ทุกเวลา

3.1.1 บล็อกไดอะแกรมแสดงโครงสร้างการเชื่อมต่อ Hardware



ภาพที่ 3.1 บล็อกไดอะแกรมแสดงโครงสร้าง

X86 คือ คอมพิวเตอร์ที่จะทำการรับข้อมูลตำแหน่งปัจจุบันของตัวจานดาวเทียมจาก GPS และจะทำหน้าที่รับคำสั่งจากผู้ใช้งานแล้วทำการส่งข้อมูลต่างๆ เหล่านั้นผ่านทางสาย RS232 ไปยังคอนโทรเลอร์ทั้ง 3 ตัวเพื่อใช้ประกอบในการหาตำแหน่งการเคลื่อนที่ที่ถูกต้อง ซึ่งจะประกอบไปด้วยคอนโทรเลอร์จำนวน 3 ตัว คือ Controller ควบคุมแกน Azimuth Controller ควบคุมแกน Elevation และ controller ควบคุมแกน Polarization เพื่อควบคุมมอเตอร์ให้ขับเคลื่อนไปตามตำแหน่งที่ต้องการได้อย่างแม่นยำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.2 องค์ประกอบของ MVV

ภาพที่ 3.2 MVV Vehicle (Toyota Fortuner)

3.1.2.1 ส่วนประกอบของจานดาวเทียม

- Antenna Structure

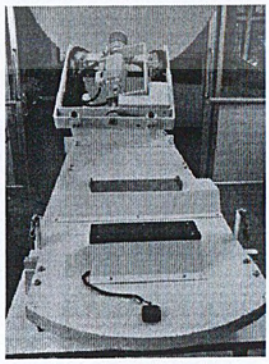
ภาพที่ 3.3 Antenna

Antenna Specification

Antenna Type : One-piece precision offset Thermoset-molded reflector

Reflector Material : Glass Fiber ReinforcedPolyester

- Antenna Component



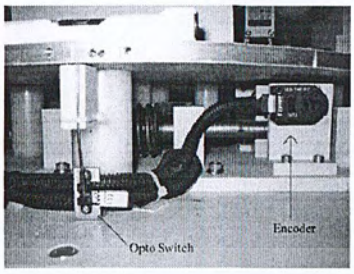
ภาพที่ 3.4 Antenna Platform

ภาพที่ 3.5 Antenna Dish / Feed Support

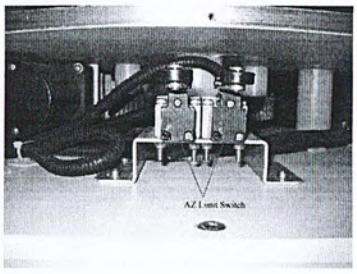
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.2.2 ส่วนประกอบของ Azimuth Part

Azimuth Part ประกอบด้วย Motor, Limit Switch, Opto Sensor และ Azimuth Gear



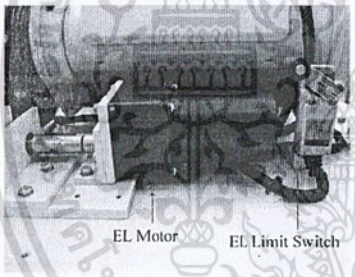
ภาพที่ 3.6 Opto Switch



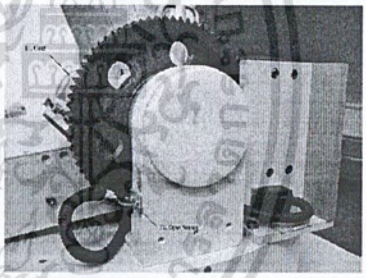
ภาพที่ 3.7 Limit Switch

3.1.2.3 ส่วนประกอบของ Elevation Part

Elevation Part ประกอบด้วย Motor, Limit Switch, Opto Sensor และ Elevation Gear



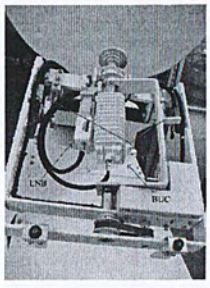
ภาพที่ 3.8 Elevation Motor



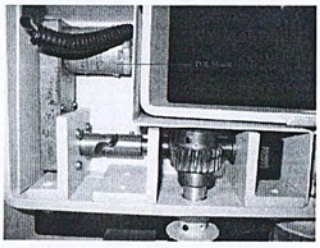
ภาพที่ 3.9 Elevation Gear

3.1.1.4 ส่วนประกอบของ Polarization Part

Polarization Part ประกอบด้วย Motor, Limit Switch และ Polarization Gear



ภาพที่ 3.10 Limit Switch

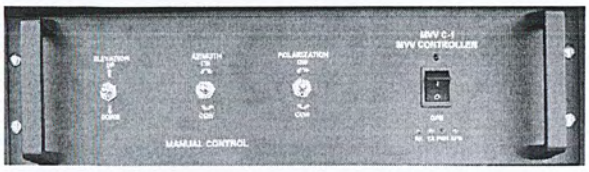


ภาพที่ 3.11 Polarization Motor

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.2.5 ส่วนประกอบของ Controller

Controller Box ประกอบด้วย Software Control, Manual Control และ GPS



ภาพที่ 3.12 controller

GPS Part ประกอบด้วย GPS Antenna และ GPS Receiver

- IPSTAR IDU

ภาพที่ 3.13 Enterprise Series

3.2 การศึกษาโปรแกรมสำหรับการปรับตำแหน่งจานดาวเทียมติดรถยนต์ที่เขียนด้วยวิชวลเบสิก 6 (Visual Basic 6)

วิชวลเบสิกเป็นภาษาคอมพิวเตอร์เป็นที่พัฒนาโดยบริษัท ไมโครซอฟท์ ซึ่งเป็นบริษัทที่สร้างระบบปฏิบัติการวินโดว์ (Windows) โดยตัวภาษามีรากฐานมาจากภาษาเบสิก (Basic) ซึ่งย่อมาจาก Beginners All Purpose Symbolic Instruction แปลความหมายก็คือ ชุดคำสั่งหรือภาษาคอมพิวเตอร์สำหรับผู้เริ่มต้น ภาษาเบสิกมีจุดเด่น คือ เหมาะสำหรับผู้ที่ไม่มีพื้นฐานการเขียนโปรแกรม ซึ่งสามารถเรียนรู้และนำไปใช้งานได้อย่างง่ายและรวดเร็วเมื่อเทียบกับการเรียนรู้ภาษาคอมพิวเตอร์อื่น ไมโครซอฟท์ได้พัฒนาโปรแกรมภาษาเบสิกมาตั้งแต่ภาษา Microsoft Basic, BASIC Advanced GWBASIC และ QuickBasic ซึ่งติดตั้งมาพร้อมกับระบบปฏิบัติการ NS DOS ในที่สุดได้ชื่อว่า QBASIC โดยแต่ละเวอร์ชั่นออกมานั้นได้มีการพัฒนาได้เพิ่มคำสั่งต่างๆเข้าไปตลอด

ในอดีตโปรแกรมภาษาเหล่านี้ล้วนทำงานใน Text Mode คือเป็นตัวอักษรล้วนๆ ไม่มีตัวเลือกใช้งานที่เป็นภาพกราฟฟิกสวยงามแบบระบบปฏิบัติการ Window ในปัจจุบันจนกระทั่งเมื่อระบบปฏิบัติการวินโดว์ (Window) ได้รับความนิยมมากขึ้น ไมโครซอฟเห็นว่าโปรแกรมภาษาใน Text Mode นั้นจะได้รับความนิยมในการใช้งานที่น้อยลง จึงได้พัฒนาปรับปรุงโปรแกรมภาษา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เบสิกออกแบบมารูปแบบใหม่ เพื่อรองรับการทำงานในระบบปฏิบัติการ Windows เป็นที่มาของการพัฒนาโปรแกรม Visual Basic นั่นเอง

วิชวลเบสิกเวอร์ชั่นแรกคือ เวอร์ชั่น 1.0 เปิดตัวอย่างเป็นทางการตั้งแต่ปี 1991 โดยในช่วงแรกไม่มีความสามารถต่างจากภาษา QBASIC มากนัก แต่จะเน้นเรื่องเครื่องมือที่ช่วยในการเขียนโปรแกรมบนวินโดว์ ซึ่งปรากฏว่าวิชวลเบสิกได้รับความนิยมและประสบความสำเร็จเป็นอย่างดี ไมโครซอฟท์ได้พัฒนาวิชวลเบสิกให้ดีเพิ่มมากขึ้น ทั้งในด้านประสิทธิภาพ ความสามารถ และเครื่องมือต่างๆ เช่น เครื่องมือตรวจสอบแก้ไขโปรแกรม (Debugger) สภาพแวดล้อมการพัฒนาโปรแกรม การเขียนโปรแกรมหลายวินโดว์ย่อย และเมื่อเป็นวิชวลเบสิกซึ่งใช้ลักษณะของการมองเห็นได้ ที่เป็นการติดต่อกับผู้ใช้งานด้วยกราฟิกหรือรูปภาพ (Graphical User Interface: GUI) จึงทำให้การพัฒนาโปรแกรมใช้งานได้สะดวกและรวดเร็วขึ้น ถึงแม้จะใช้งานง่าย แต่ก็มีความสามารถสูงเหมาะสำหรับการพัฒนาโปรแกรมใช้งานได้หลายด้าน

3.2.1 ลักษณะการสร้างโปรแกรมใช้งานด้วยวิชวลเบสิก

การสร้างโปรแกรมใช้งานด้วยวิชวลเบสิกเป็นการเขียนโปรแกรมเชิงวัตถุ โดยสร้างโครงงานประกอบด้วย ฟอร์มสำหรับจัดวางวัตถุชนิดต่างๆ ซึ่งเรียกว่า ตัวควบคุม ที่เป็นรูปแบบเพื่อให้ผู้ใช้โปรแกรมมองเห็นง่ายและสามารถเลือกกระทำกับตัวควบคุมตัวใดตัวหนึ่งได้ การเขียนคำสั่ง หรือโปรแกรมเก็บไว้ที่ตัวควบคุมแต่ละตัวเพื่อให้ตัวควบคุมทำงานตามลักษณะการกระทำหรือเหตุการณ์ที่เกิดขึ้นกับตัวควบคุมนั้น ดังนั้นจึงต้องเขียนโปรแกรมเก็บไว้ที่ปุ่มให้มีความหมายว่า ถ้าคลิกปุ่มนี้แล้วจะให้ทำอะไรต่อไปบ้าง

3.2.2 ชนิดข้อมูลของโปรแกรมวิชวลเบสิก

ข้อมูลที่ใช้ในการเขียนกับวิชวลเบสิก นั้นมีหลายประเภทไม่ว่าจะเป็นข้อมูลในรูปแบบของตัวเลข ข้อมูลที่เป็นตัวอักษร หรือข้อมูลที่ประกอบกันขึ้นเป็นโครงสร้างพิเศษ เช่น ข้อมูลในลักษณะเป็นชุด หรือข้อมูลที่มีลักษณะเฉพาะตัว เช่น ออบเจ็กต์ อาจแบ่งข้อมูลต่างๆ ที่ใช้ในการเขียนโปรแกรมกับวิชวลเบสิก ออกเป็นกลุ่มได้ดังนี้

- กลุ่มข้อมูลชนิดตัวเลข เป็นกลุ่มข้อมูลที่เก็บตัวเลขชนิดต่างๆ ทั้งจำนวนเต็ม และเลขทศนิยม
- กลุ่มข้อมูลชนิดข้อความ เป็นกลุ่มข้อมูลที่เก็บตัวอักษรทั้งอักษรตัวเดียวและข้อความ
- กลุ่มข้อมูลตรรกศาสตร์ ข้อมูลชนิด Boolean ใช้สำหรับเก็บข้อมูลด้านตรรกศาสตร์ ซึ่งก็คือ ค่าจริง (True) และเท็จ (False)
- ข้อมูลชนิดออบเจ็กต์ เป็นชนิดข้อมูลที่สามารถเก็บข้อมูลชนิดอื่นๆ ไว้ภายในและมีความสามารถในการทำงานอยู่ภายในด้วย

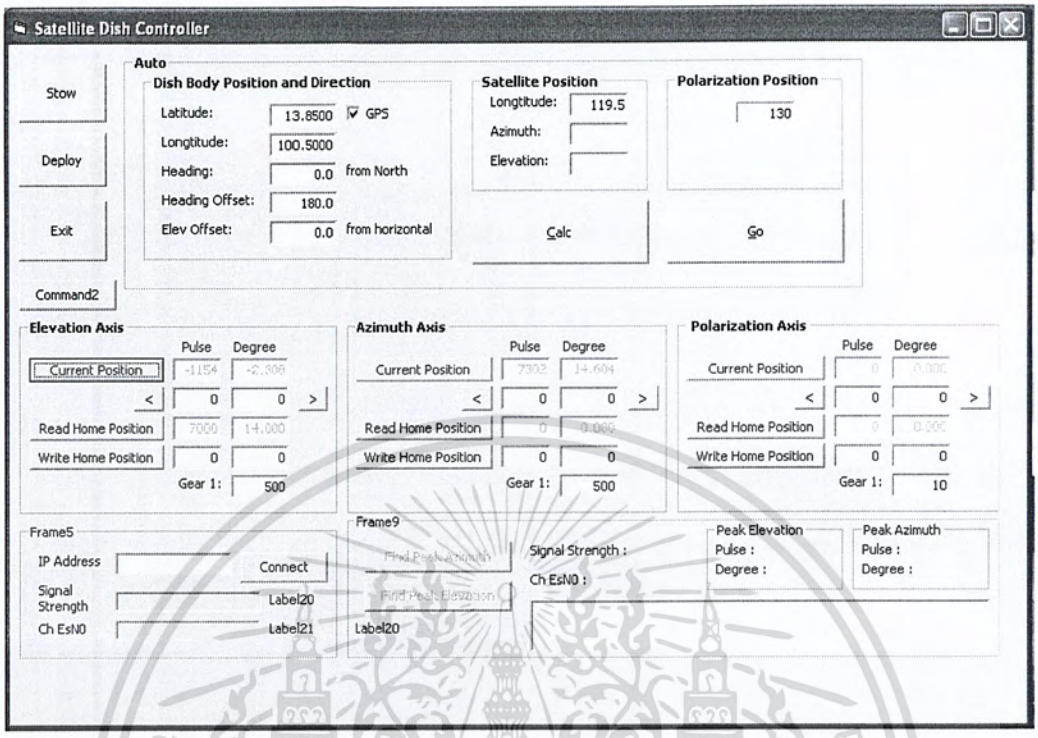
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3.1 ชนิดข้อมูลในวิชวลเบสิก

ชนิดข้อมูล	ขนาดหน่วยความจำ (ไบต์)	ค่าที่เก็บในตัวแปร
Boolean	2	มี 2 ค่าคือ True กับ False
Byte	1	0 ถึง 255 (หรือเรียกว่า Unsigned Byte)
SByte	1	-128 ถึง 127 (หรือเรียกว่า Signed Byte)
Char	2	เป็นตัวอักษรหนึ่งตัว ซึ่งจะเก็บแบบ Unicode (มีตัวอักษรแทบทุกภาษาในโลก หรือเรียกว่า Unsigned Char)
Short	2	-32, 768 ถึง -32, 767
UShort	2	0 ถึง 65,535
Integer	4	-2,147,483,648 ถึง 2,147,647
UInteger	4	0 ถึง 65,535 (หรือเรียกว่า Unsigned Integer)
Long	8	-9,223,372,036,854,775,808 ถึง 9,223,372,036,854,775,807
ULong	8	0 ถึง18,446,744,073,709,551,615 (หรือเรียกว่า Unsigned Long)
Decimal	16	เป็นเลขจำนวนเต็มและทศนิยม ซึ่งมีความละเอียดแล้วแต่จะกำหนด
Single	4	เป็นเลขทศนิยมระหว่าง -3.402823E38 ถึง -1.401298E-45 สำหรับค่าลบและ 1.401298E-45 ถึง 3.402823E38 สำหรับค่าบวก
Double	8	เป็นเลขทศนิยมระหว่าง -1.79769313486232E308 ถึง 4.94065645841247 E-324 สำหรับค่าลบ และ 4.940656455841247E-324 ถึง 1.79769313486232E308 สำหรับค่าบวก
String	แปรผันตามความยาวของข้อมูล	สตริงคือ ข้อความ ซึ่งมีความยาวตั้งแต่ 0 ถึง 2 พันล้านตัวอักษร (ตัวอักษรนี้เป็นชนิด Unicode Character)
Date	8	วันเวลาตั้งแต่วันที่ 1 มกราคม ค.ศ 0000 เวลา 00:00:00 ถึง 31 ธันวาคม ค.ศ. 9999 เวลา 23:59:59
Object	4	เป็นข้อมูลชนิดออบเจ็กต์ ซึ่งก็มีออบเจ็กต์อยู่หลายรูปแบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.3 ส่วนที่ติดต่อกับผู้ใช้งาน



ภาพที่ 3.14 ส่วนติดต่อกับผู้ใช้งานเดิม Visual Basic

3.2.3.1 ส่วนของ Signal Status

Signal Status ประกอบด้วย

- **Deploy** หมายถึง การตั้งตำแหน่งของจานดาวเทียมไว้ที่ตำแหน่ง Deploy เตรียมพร้อมที่จะทำการค้นหาตำแหน่งของดาวเทียม
- **Stow** หมายถึง การเก็บจานดาวเทียบเข้าที่ตำแหน่ง Home เมื่อจบการใช้งาน และพร้อมเดินทางต่อ
- **Signal Strength** หมายถึง ส่วนแสดงผลความแรงของสัญญาณดาวเทียมที่สามารถรับได้ในขณะนั้น
- **Channel EsNo** หมายถึง ส่วนแสดงผลของสัญญาณดาวเทียมที่สามารถรับได้ในขณะนั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.3.2 ส่วนของ Antenna Details

Antenna Details คือส่วนที่แสดงถึงรายละเอียดของส่วนต่างๆ ของจานดาวเทียม ประกอบด้วย

• **Dish Body Position and Direction** ประกอบด้วย ค่าต่าง ๆ ที่สามารถรับได้จากดาวเทียม GPS เพื่อระบุตำแหน่งของรถ MVV ได้แก่

- **GPS** ถ้ามีการถูกเลือก (check box) ด้านหลัง หมายถึงขณะนี้ใช้สัญญาณจากดาวเทียม GPS เพื่อใช้ในการคำนวณหาตำแหน่งของรถ MVV
- **Latitude** แสดงตำแหน่งของรถในด้าน Latitude
- **Longitude** แสดงตำแหน่งของรถในด้าน Longitude
- **Heading** แสดงทิศทางของรถ MVV เทียบกับ True North

• **Satellite Position** ประกอบด้วยส่วนต่าง ๆ ดังนี้

- **Longitude** แสดงตำแหน่งของดาวเทียมดวงที่จะทำการค้นหา
- **Elevation** แสดงผลจากการคำนวณตำแหน่งของดาวเทียมและตำแหน่งที่จานรับสัญญาณดาวเทียมจะเคลื่อนที่ทางด้าน EL
- **Azimuth** แสดงผลจากการคำนวณตำแหน่งของดาวเทียม และตำแหน่งที่จานรับสัญญาณดาวเทียมจะเคลื่อนที่ทางด้าน AZ

• **Polarization Position** แสดงผลจากการคำนวณตำแหน่งของดาวเทียมและตำแหน่งที่จานรับสัญญาณดาวเทียมจะเคลื่อนที่ทางด้าน POL

3.2.3.3 ส่วนของ Dish Position

Dish Positionหมายถึง ตำแหน่งของจานดาวเทียมในขณะนั้น ในด้านต่าง ๆได้แก่

• **Elevation Axis** แสดงตำแหน่งของจานดาวเทียมในขณะนั้น ว่าอยู่ที่ตำแหน่งกี่องศา และจำนวน Pulse ที่จ่ายให้แก่มอเตอร์ด้าน EL

• **Azimuth Axis** แสดงตำแหน่งของจานดาวเทียมในขณะนั้น ว่าอยู่ที่ตำแหน่งกี่องศา และจำนวน Pulse ที่จ่ายให้แก่มอเตอร์ด้าน AZ

• **Polarization Axis** แสดงตำแหน่งของจานดาวเทียมในขณะนั้น ว่าอยู่ที่ตำแหน่งกี่องศา และจำนวน Pulse ที่จ่ายให้แก่มอเตอร์ด้าน POL

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3 การพัฒนาโปรแกรมสำหรับการปรับตำแหน่งจานดาวเทียมติดรถยนต์ด้วย Visual C#

เนื่องจาก Visual C# เป็นภาษาคอมพิวเตอร์ที่ออกแบบมาเพื่อรองรับการทำงานในยุคใหม่ เป็นภาษาโปรแกรมที่ถูกสร้างขึ้นมาสำหรับการพัฒนาซอฟต์แวร์ภายใต้เทคโนโลยี Microsoft.NET โดยเป็นภาษาระดับสูงที่ใช้รูปแบบการเขียนโปรแกรมแบบเชิงวัตถุสมัยใหม่ (Modern Object Oriented Programming) จุดเด่นสำคัญของภาษา Visual C# คือ การรวมเอาความสามารถของภาษา C++ มารวมกับความใช้งานง่ายของภาษา Visual Basic ทำให้ภาษา Visual C# เป็นภาษาคอมพิวเตอร์ที่ความน่าเชื่อถือสูง สามารถนำมาใช้พัฒนาเป็น Desktop Application บนระบบปฏิบัติการในตระกูล Windows ได้ ในขณะเดียวกันใช้งานง่าย ทำให้การพัฒนาโปรแกรมทำได้อย่างมีประสิทธิภาพ สะดวก ง่ายดาย รวดเร็ว ทำให้สะดวกต่อการศึกษาและพัฒนาต่อไปในอนาคต

ทั้งนี้ในการพัฒนาโปรแกรมไปเป็น Visual C# นั้นก็เพื่อให้แกนการหมุนของจานดาวเทียมสามารถหมุนได้พร้อมกันทั้งสามแนวแกน เนื่องจากส่วนของโปรแกรม Visual Basic 6 นั้นแกนการหมุนของจานดาวเทียมจะหมุนได้ทีละแกนเท่านั้น

3.3.1 การเขียนโปรแกรมแบบ Event Driven Programming

ในการเขียนโปรแกรมเพื่อสร้างแอพพลิเคชันเพื่อทำงานบน Windows นั้นจะมีวิธีคิดในการเขียนโปรแกรมที่แตกต่างจากรูปแบบเดิม หากต้องเขียนโปรแกรมด้วยภาษา C หรือ Pascal นั้นมีความจำเป็นที่จะต้องเข้าใจขั้นตอนการทำงานทั้งหมดเสียก่อน จึงจะสามารถเขียนโปรแกรมให้ครอบคลุมการทำงานเหล่านั้นได้ ซึ่งถือว่าสามารถใช้ได้ดีหากโปรแกรมนั้นมีขนาดเล็กหรือไม่ซับซ้อนมากนัก เรียกวิธีคิดนี้ว่า การเขียนโปรแกรมแบบ Structure Programming

แต่หากโปรแกรมนั้นมีขนาดใหญ่ขึ้นและซับซ้อน หรือเป็นระบบงานที่มีความหลากหลาย หากใช้วิธีคิดแบบ Structure Programming แล้วจะทำให้การเขียนโปรแกรมยุ่งยากแก้ไขยาก โดยเฉพาะอย่างยิ่งเมื่อโปรแกรมนั้นต้องทำงานบน Windows ดังนั้นไมโครซอฟต์จึงมีวิธีคิดที่เรียกว่า Event Driven Programming ขึ้นมาสำหรับการเขียนโปรแกรมบน Windows โดย Event Driven Programming นั้นจะประเมินว่าหากเกิดเหตุการณ์ใดขึ้นจะจัดการกับมันอย่างไรโดยไม่สนใจผลกระทบที่จะเกิดขึ้นกับโปรแกรม

3.3.2 ออบเจ็กต์ (Object)

สิ่งที่ต้องเรียนรู้เมื่อต้องเขียนโปรแกรม หรือพัฒนาแอพพลิเคชันเพื่อใช้งานบน Windows ก็คือ เรื่องของออบเจ็กต์ (Object) ซึ่งจะสอดคล้องกับแนวคิดของ Event Driven Programming ซึ่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ที่ผ่านมาจะเห็นว่าแอพพลิเคชันนั้นจะประกอบมาจากสิ่งต่างๆ เช่น ปุ่มกด, วินโดว์ ซึ่งเรียกว่าแต่ละสิ่งที่ประกอบเป็นแอพพลิเคชันว่า ออบเจ็กต์ (Object)

3.3.3 พร็อพเพอร์ตี้ (Property) : คุณสมบัติของออบเจ็กต์

ออบเจ็กต์แต่ละชนิดจะมีคุณสมบัติประจำตัว เช่น ออบเจ็กต์รถยนต์ก็จะมีคุณสมบัติ เช่น ยี่ห้อ, รุ่น, ขนาดของเครื่องยนต์ และสี เป็นต้น ซึ่งเราเรียกคุณสมบัตินี้ว่าพร็อพเพอร์ตี้

ในการใช้งาน Visual C# นั้นการกำหนดพร็อพเพอร์ตี้ให้กับออบเจ็กต์ชนิดต่างๆ จะสามารถทำได้ทั้งในช่วงของการออกแบบ และในขณะที่แอพพลิเคชันนั้นทำงานจะใช้การเขียนโปรแกรมกำกับการทำงานเอาไว้

3.3.4 คอนโทรลกับคอมโพเนนต์

ในการสร้างแอพพลิเคชันด้วย Visual C# นั้นมีออบเจ็กต์ให้เลือกใช้งานได้หลายตัว ออบเจ็กต์ที่สามารถมองเห็นได้เราจะเรียกว่า คอนโทรล (Control) เช่น คอนโทรล Button และคอนโทรล Textbox เป็นต้น และยังมีออบเจ็กต์อีกประเภทหนึ่งที่มองไม่เห็นเวลาที่แอพลิเคชันทำงาน (ทำงานอยู่ฉาก) จะเรียกว่า คอมโพเนนต์ (Component)

3.3.5 การกำหนดค่าใน Property Window

ในการกำหนดค่าให้กับพร็อพเพอร์ตี้ต่างๆ จะกำหนดผ่านหน้าต่าง Property Window ซึ่งจะสามารถกำหนดได้ 3 รูปแบบ

3.3.5.1 การกำหนดค่าอย่างอิสระ

สามารถป้อนค่าใน Property Window ได้อย่างอิสระ เช่น พร็อพเพอร์ตี้ Text และพร็อพเพอร์ตี้ Name เป็นต้น

3.3.5.2 การกำหนดค่าจากรายการตัวเลือก

เป็นการกำหนดค่าโดยเลือกจากตัวเลือกในรายการที่มีให้ของ Property Window เช่น Border Style และ Back Color เป็นต้น

3.3.5.3 การกำหนดค่าจากไดอะล็อกบ็อกซ์

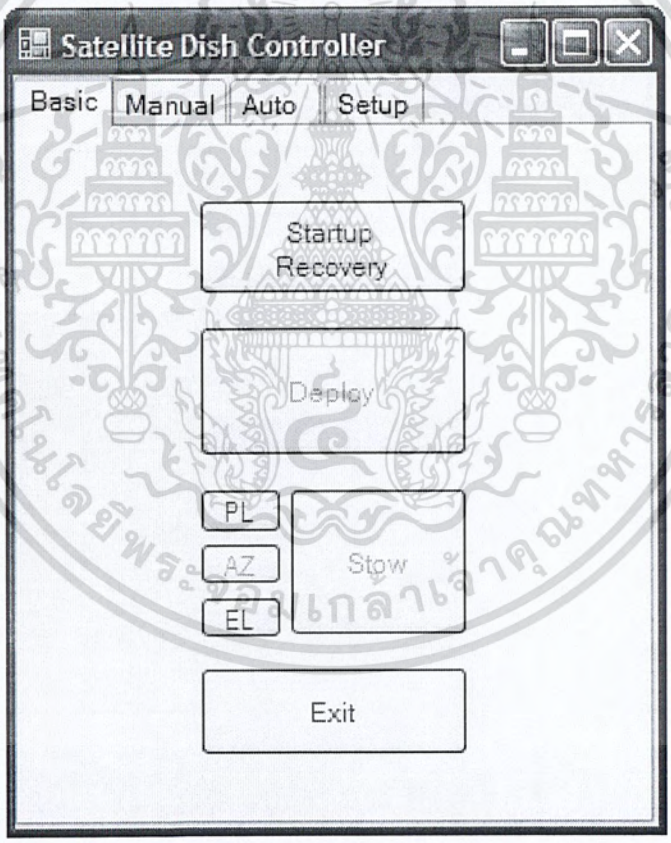
เป็นการกำหนดค่าโดยเลือกจากตัวเลือกที่มีให้ในไดอะล็อกบ็อกซ์ เช่น Font, Picture File เป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3.6 การออกแบบส่วนติดต่อกับผู้ใช้งาน

3.3.6.1 Basic Mode

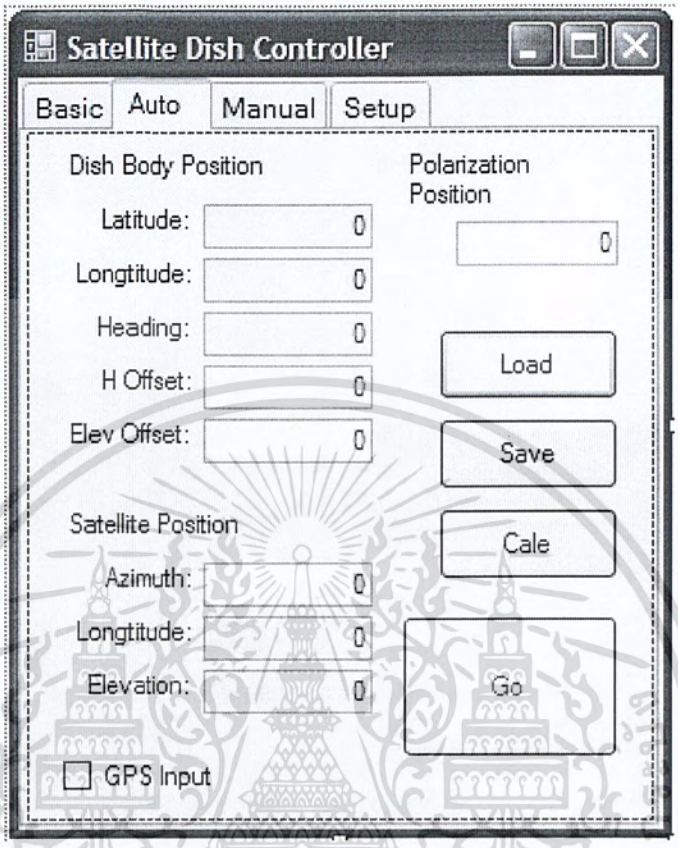
- **Startup Recovery** หมายถึง การเริ่มต้นค้นหาตำแหน่งของดาวเทียม
- **Deploy** หมายถึง การตั้งตำแหน่งของจานดาวเทียมไว้ที่ตำแหน่ง Deploy เตรียมพร้อมที่จะทำการค้นหาตำแหน่งของดาวเทียม
- **Stow** หมายถึง การเก็บจานดาวเทียบเข้าที่ตำแหน่ง Home เมื่อจบการใช้งาน และพร้อมเดินทางต่อ
- **PL AZ EL** หมายถึง การสามารถเก็บจานดาวเทียมเข้าที่ตำแหน่ง Home เมื่อจบการใช้งานที่ละแกนได้
- **Exit** หมายถึง การจบการทำงาน



ภาพที่ 3.15 Basic Mode

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3.6.2 Auto Mode



ภาพที่ 3.16 Auto Mode

- **Dish Body Position and Direction** ประกอบด้วย ค่าต่าง ๆ ที่สามารถรับได้จากดาวเทียม GPS เพื่อระบุตำแหน่งของรถ MVV ได้แก่
  - **GPS Input** ถ้ามีการถูกเลือก (check box) ด้านหลัง หมายถึง ขณะนี้ใช้สัญญาณจากดาวเทียม GPS เพื่อใช้ในการคำนวณหาตำแหน่งของรถ MVV
  - **Latitude** แสดงตำแหน่งของรถในด้าน Latitude
  - **Longitude** แสดงตำแหน่งของรถในด้าน Longitude
  - **Heading** แสดงทิศทางของรถ MVV เทียบกับ True North
- **Satellite Position** ประกอบด้วยส่วนต่าง ๆ ดังนี้
  - **Longitude** แสดงตำแหน่งของดาวเทียมดวงที่จะทำการค้นหา

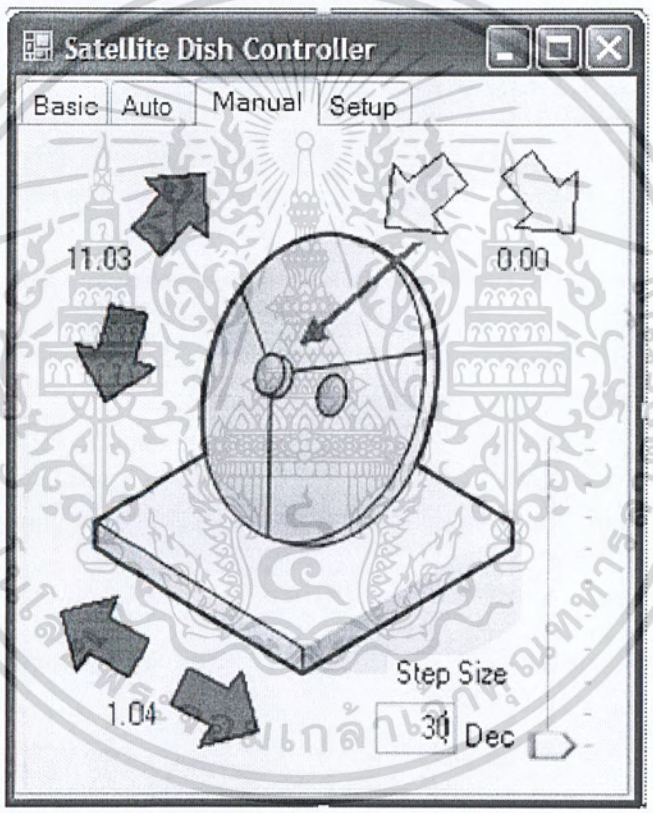
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- **Elevation** แสดงผลจากการคำนวณตำแหน่งของดาวเทียมและตำแหน่งที่จานรับสัญญาณดาวเทียมจะเคลื่อนที่ทางด้าน EL

- **Azimuth** แสดงผลจากการคำนวณตำแหน่งของดาวเทียมและตำแหน่งที่จานรับสัญญาณดาวเทียมจะเคลื่อนที่ทางด้าน AZ

• **Polarization Position** แสดงผลจากการคำนวณตำแหน่งของดาวเทียมและตำแหน่งที่จานรับสัญญาณดาวเทียมจะเคลื่อนที่ทางด้าน POL

3.3.6.3 Manual Mode

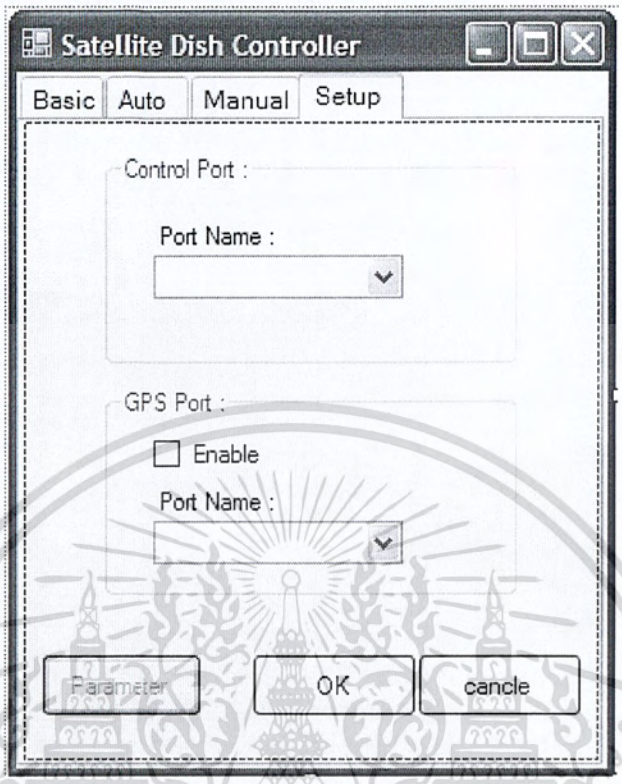


ภาพที่ 3.17 Manual Mode

• **Step Size** หมายถึง องศาที่สั่งให้แกนหมุนในการคลิกเมาส์ 1 ครั้ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

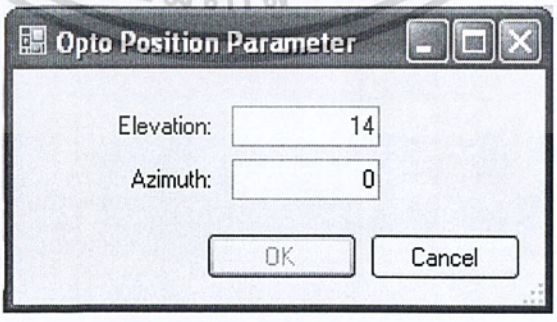
3.3.6.4 Communication Mode



ภาพที่ 3.18 Communication Port

- **Com Port Mode** เป็นโหมดใน การ set ค่า Com Port และ GPS Ports

3.3.6.5 Parameter

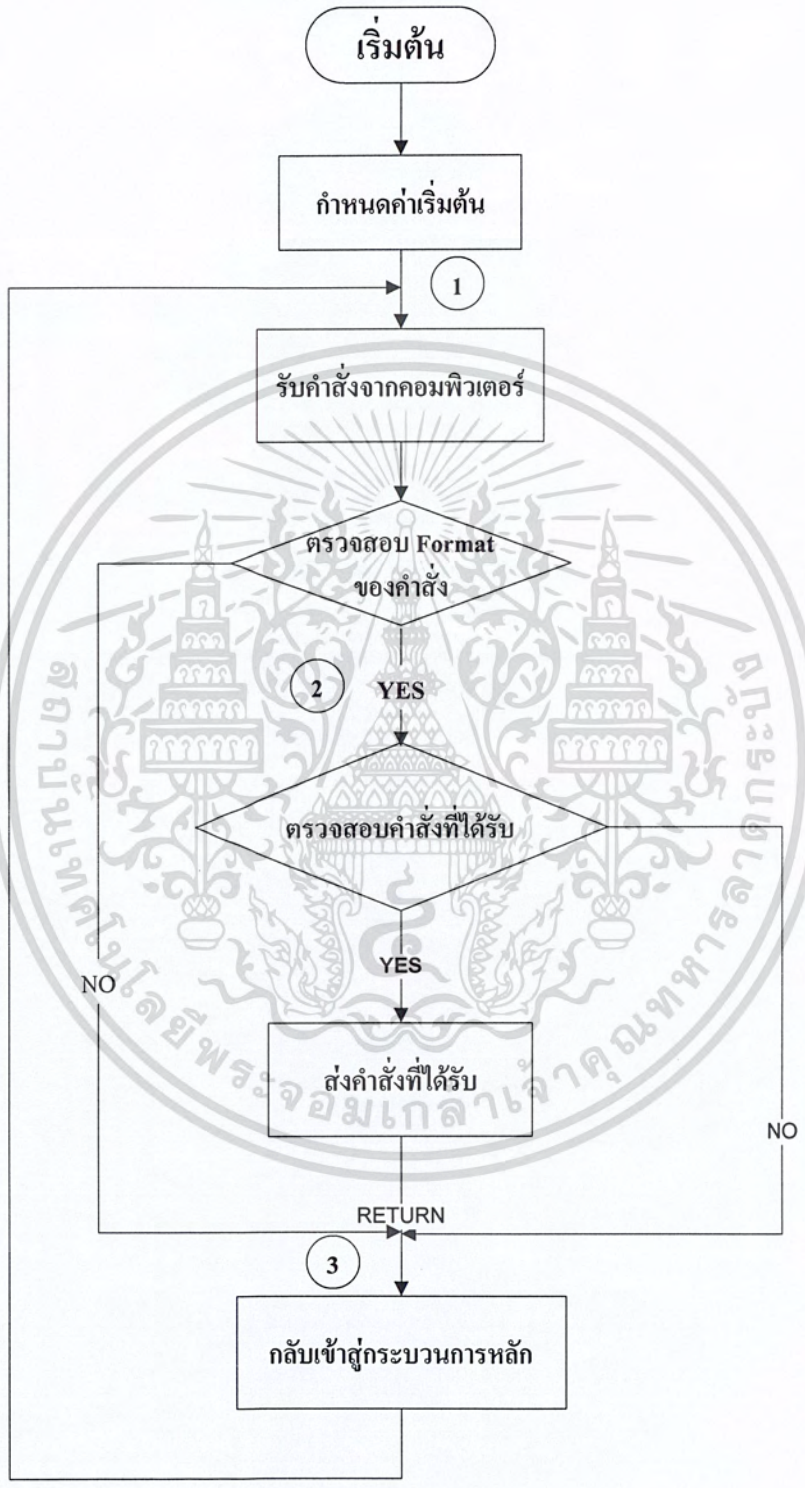


ภาพที่ 3.19 Opto Position Parameter

- **Opto Position Parameter** คือ การตั้งค่า Opto Sensor ของแกน Elevation และ Azimuth

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4 การทำงานของไมโครคอนโทรลเลอร์ PIC16F873A

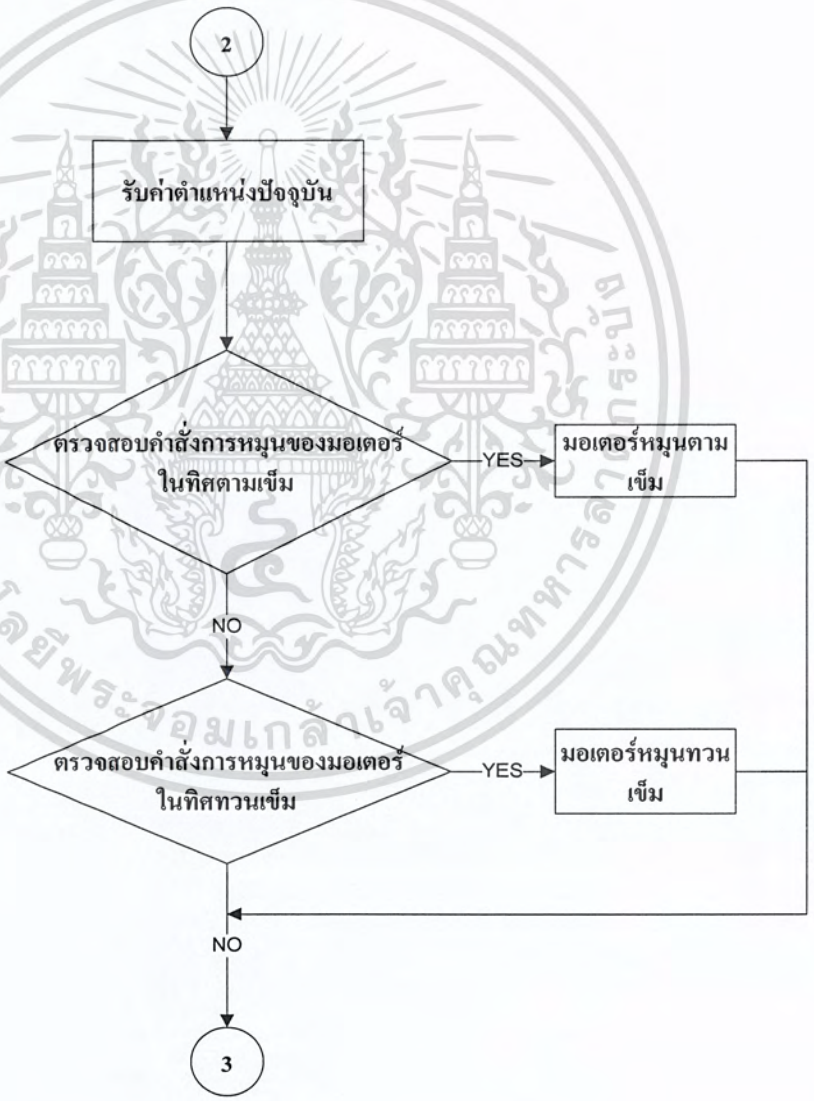


ภาพที่ 3.20 แผนภูมิแสดงการทำงานของไมโครคอนโทรลเลอร์ PIC16F873A

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อเริ่มต้นการทำงาน โปรแกรมจะทำการเคลียร์ค่าที่ค้างอยู่ เพื่อพร้อมรับค่าใหม่เข้ามา จากนั้นส่วนรับข้อมูล (Get Command From Serial Port) จะรับคำสั่งจากผู้ใช้มาทำการตรวจข้อมูล Format เบื้องต้นว่าตรงตามรูปแบบหรือไม่ เมื่อข้อมูลที่รับเข้ามาถูกต้องตรงตามรูปแบบ ก็จะส่งไปยังส่วนตรวจสอบคำสั่งเพื่อตีความคำสั่ง ในที่นี้คือคำสั่งการขับมอเตอร์ในทิศทางทวนเข็มนาฬิกา หรือตามเข็มนาฬิกา จากนั้นคำสั่งนี้จะส่งสัญญาณไปทำการขับมอเตอร์ให้หมุนไปในทิศทางที่ต้องการเพื่อขับเคลื่อนจานดาวเทียม โดยการเคลื่อนที่ของจานดาวเทียมจะถูกจำกัดให้อยู่ในช่วงที่เหมาะสมโดยลิมิตสวิทช์ ถ้าหากจานดาวเทียมเคลื่อนที่จนชนลิมิตสวิทช์จะมีการส่งสัญญาณเข้าสู่กระบวนการหลักเพื่อทำการหยุดการหมุนของมอเตอร์

3.4.1 การตรวจสอบคำสั่ง

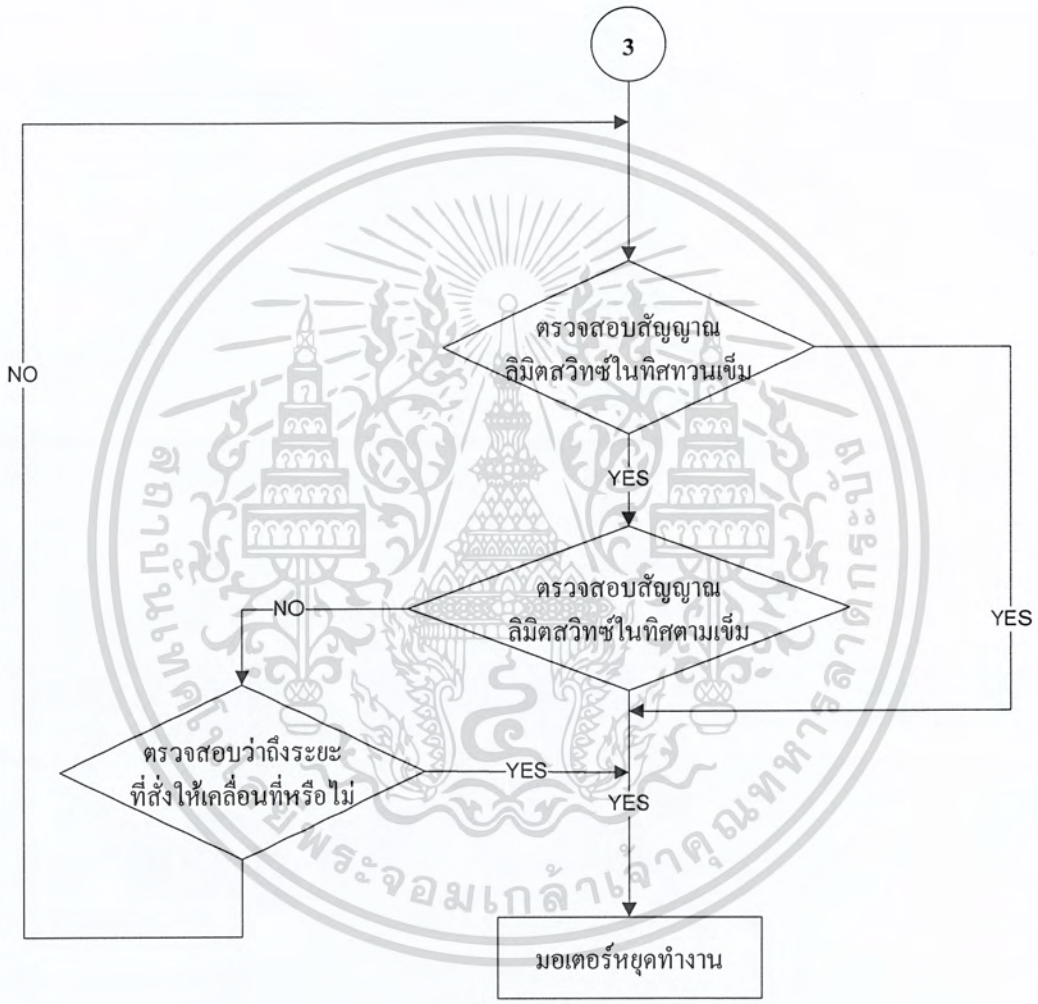


ภาพที่ 3.21 แผนภูมิแสดงการตรวจสอบของคำสั่งที่ได้รับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยจะมีการตรวจสอบค่าตำแหน่งปัจจุบันก่อน เมื่อทราบค่าของตำแหน่งแล้วจะทำการส่งคำสั่งที่ได้รับมาไปตรวจสอบ ว่าเป็นคำสั่งหมุนมอเตอร์ตามเข็มนาฬิกาหรือไม่ ถ้าใช่คำสั่งนี้ก็จะถูกนำไปปฏิบัติ แต่ถ้าไม่ใช่ก็จะตรวจสอบคำสั่งถัดไปว่าใช่คำสั่งหมุนมอเตอร์ทวนเข็มนาฬิกาหรือไม่ ถ้าใช่คำสั่งนี้ก็จะถูกนำไปปฏิบัติ แต่ถ้าไม่ใช่ก็จะถูกส่งกลับเข้าสู่กระบวนการหลัก

3.4.2 การทำงานของกระบวนการหลัก



ภาพที่ 3.22 แผนภูมิแสดงการทำงานของกระบวนการหลัก

หลังจากที่ตรวจสอบคำสั่งและนำไปปฏิบัตินั้น การหมุนของมอเตอร์จะต้องมีขอบเขตที่จำกัดเพื่อไม่ให้จานรับสัญญาณเคลื่อนที่มากเกินไปจนก่อให้เกิดความชำรุดเสียหาย ดังนั้นในกระบวนการหลักนี้จึงมีหน้าที่ตรวจสอบ และหยุดเมื่อถึงระยะที่จำกัดการเคลื่อนที่ โดยมีหลักการดังนี้ ตรวจสอบว่ามีสัญญาณจาก Limit Switch เข้ามาหรือยังหากมีก็จะดูว่าใช่สัญญาณจาก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Limit Switch ทางด้านตามเข็มนาฬิกาหรือไม่ ถ้าใช่มอเตอร์ก็จะหยุดหมุน ส่วนในกรณีที่ไม่มีสัญญาณจาก Limit Switch ทางด้านตามเข็มนาฬิกา ให้ไปตรวจสอบสัญญาณจาก Limit Switch ทางด้านทวนเข็มนาฬิกา ถ้ามีสัญญาณมอเตอร์ก็จะหยุดหมุนแต่ถ้าไม่มีสัญญาณจาก Limit Switch มอเตอร์ก็ยังคงทำงานตามคำสั่งต่อไปจนกว่าจะถึงระยะที่กำหนดให้เคลื่อนที่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

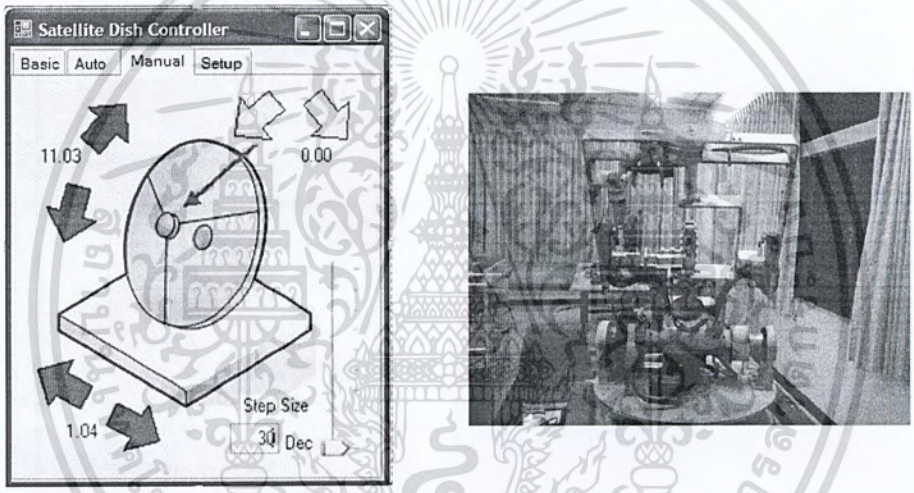
ผลการทดลอง

4.1 การเคลื่อนที่ในแนวแกนของมุมก้มเงย (Elevation)

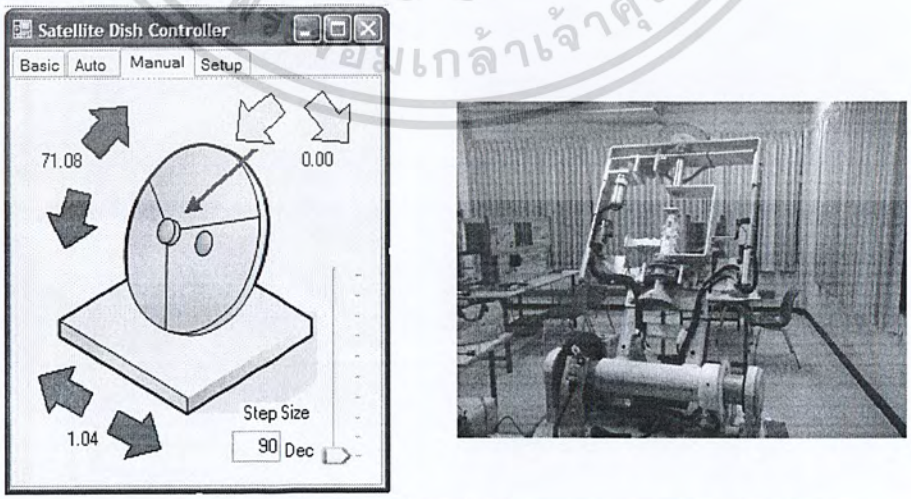
4.1.1 ลำดับขั้นตอนในการทำงาน

- การทดลองทดสอบความถูกต้องการเคลื่อนที่ของแกน Elevation

ในการทดลองจะทำการเคลื่อนที่ครั้งละ 10 องศา, 20 องศา และ 30 องศา ตามลำดับ โดยทำการทดลององศาละ 3 ครั้ง เพื่อหาค่าเฉลี่ยซึ่งในการทดลองแต่ละครั้งนั้นจะทำการบันทึกค่าองศาที่กำหนด, องศาที่เคลื่อนที่จริงจากอุปกรณ์วัดมุม และค่าองศาที่แสดงในโปรแกรม



ภาพที่ 4.1 แสดงการเคลื่อนที่ของแกน Elevation ไปที่ 30 องศา



ภาพที่ 4.2 แสดงการเคลื่อนที่ของแกน Elevation ไปที่ 90 องศา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.1.2 ผลการทดลองก่อนการแก้ไข

ตารางที่ 4.1 แสดงค่าเฉลี่ยการหมุนแนวแกน Elevation ก่อนการแก้ไขโดยเพิ่ม/ลด ครั้งละ 10 องศา

องศา	Forward				Reward			
	ครั้งที่ 1	ครั้งที่ 2	ครั้งที่ 3	เฉลี่ย	ครั้งที่ 1	ครั้งที่ 2	ครั้งที่ 3	เฉลี่ย
0	0	0	0	0.0	0	0	0	0.0
10	10	10	10	10.0	4	4	4	4.0
20	20	20	20	20.0	14	14	14	14.0
30	30	30	30	30.0	25	24	24	24.3
40	40	40	40	40.0	35	34	34	34.3
50	51	50	50	50.3	50	44	44	46.0
60	61	60	60	60.3	55	55	55	55.0
70	71	70	70	70.3	66	65	65	65.3
80	81	80	80	80.3	76	75	75	75.3
90	92	90	90	90.7	86	85	85	85.3
100	102	101	101	101.3	96	100	95	97.0
110	112	111	111	111.3	106	105	105	105.3
120	122	121	121	121.3	116	115	115	115.3
130	132	131	131	131.3	126	125	125	125.3
140	136	135	135	135.3	136	135	135	135.3

ตารางที่ 4.2 แสดงค่าเฉลี่ยการหมุนแนวแกน Elevation ก่อนการแก้ไขโดยเพิ่ม/ลด ครั้งละ 20 องศา

องศา	Forward				Reward			
	ครั้งที่ 1	ครั้งที่ 2	ครั้งที่ 3	เฉลี่ย	ครั้งที่ 1	ครั้งที่ 2	ครั้งที่ 3	เฉลี่ย
0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
20	20.0	20.0	20.0	20.0	15.0	15.0	15.0	15.0
40	40.0	40.0	40.0	40.0	35.0	36.0	36.0	35.7
60	60.0	60.0	60.0	60.0	55.0	56.0	56.0	55.7
80	80.0	80.0	80.0	80.0	76.0	76.0	76.0	76.0
100	101.0	100.0	100.0	100.3	96.0	96.0	96.0	96.0
120	121.0	120.0	120.0	120.3	116.0	116.0	116.0	116.0
140	135.5	136.0	136.0	135.8	135.5	136.0	136.0	135.8

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.3 แสดงค่าเฉลี่ยการหมุนแนวแกน Elevation ก่อนการแก้ไขโดยเพิ่ม/ลด ครั้งละ 30 องศา

องศา	Forward				Reward			
	ครั้งที่ 1	ครั้งที่ 2	ครั้งที่ 3	เฉลี่ย	ครั้งที่ 1	ครั้งที่ 2	ครั้งที่ 3	เฉลี่ย
0	0.0	0.0	0.0	0.0	0	0	0	0.0
30	30.0	30.0	30.0	30.0	30	30	30	30.0
60	60.0	60.0	60.0	60.0	60	60	60	60.0
90	90.0	90.0	90.0	90.0	90	90	90	90.0
120	120.0	120.0	120.0	120.0	120.0	120.0	120.0	120.0

4.1.3 ผลการทดลองหลังการแก้ไข

ตารางที่ 4.4 แสดงค่าเฉลี่ยการหมุนแนวแกน Elevation หลังการแก้ไขโดยเพิ่ม/ลด ครั้งละ 10 องศา

องศา	Forward				Reward			
	ครั้งที่ 1	ครั้งที่ 2	ครั้งที่ 3	เฉลี่ย	ครั้งที่ 1	ครั้งที่ 2	ครั้งที่ 3	เฉลี่ย
0	0	0	0	0.0	0	0	0	0.0
10	10	10	10	10.0	4	4	4	4.0
20	20	20	20	20.0	14	14	14	14.0
30	30	30	30	30.0	25	24	24	24.3
40	40	40	40	40.0	35	34	34	34.3
50	51	50	50	50.3	50	44	44	46.0
60	61	60	60	60.3	55	55	55	55.0
70	71	70	70	70.3	66	65	65	65.3
80	81	80	80	80.3	76	75	75	75.3
90	92	90	90	90.7	86	85	85	85.3
100	102	101	101	101.3	96	100	95	97.0
110	112	111	111	111.3	106	105	105	105.3
120	122	121	121	121.3	116	115	115	115.3
130	132	131	131	131.3	126	125	125	125.3
140	136	135	135	135.3	136	135	135	135.3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.5 แสดงค่าเฉลี่ยการหมุนแนวแกน Elevation หลังการแก้ไขโดยเพิ่ม/ลด ครั้งละ 20 องศา

องศา	Forward				Reward			
	ครั้งที่ 1	ครั้งที่ 2	ครั้งที่ 3	เฉลี่ย	ครั้งที่ 1	ครั้งที่ 2	ครั้งที่ 3	เฉลี่ย
0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
20	20.0	20.0	20.0	20.0	15.0	15.0	15.0	15.0
40	40.0	40.0	40.0	40.0	35.0	36.0	36.0	35.7
60	60.0	60.0	60.0	60.0	55.0	56.0	56.0	55.7
80	80.0	80.0	80.0	80.0	76.0	76.0	76.0	76.0
100	101.0	100.0	100.0	100.3	96.0	96.0	96.0	96.0
120	121.0	120.0	120.0	120.3	116.0	116.0	116.0	116.0
140	135.5	136.0	136.0	135.8	135.5	136.0	136.0	135.8

ตารางที่ 4.6 แสดงค่าเฉลี่ยการหมุนแนวแกน Elevation หลังการแก้ไขโดยเพิ่ม/ลด ครั้งละ 30 องศา

องศา	Forward				Reward			
	ครั้งที่ 1	ครั้งที่ 2	ครั้งที่ 3	เฉลี่ย	ครั้งที่ 1	ครั้งที่ 2	ครั้งที่ 3	เฉลี่ย
0	0.0	0.0	0.0	0.0	0	0	0	0.0
30	30.0	30.0	30.0	30.0	30	30	30	30.0
60	60.0	60.0	60.0	60.0	60	60	60	60.0
90	90.0	90.0	90.0	90.0	90	90	90	90.0
120	120.0	120.0	120.0	120.0	120.0	120.0	120.0	120.0

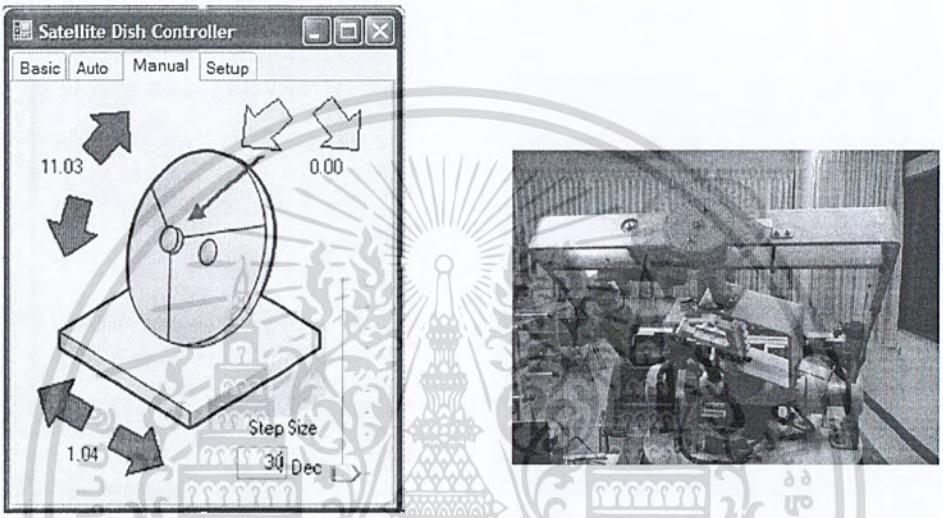
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2 การเคลื่อนที่ในแนวแกนของการหมุนขั้วคลื่น (Polarization)

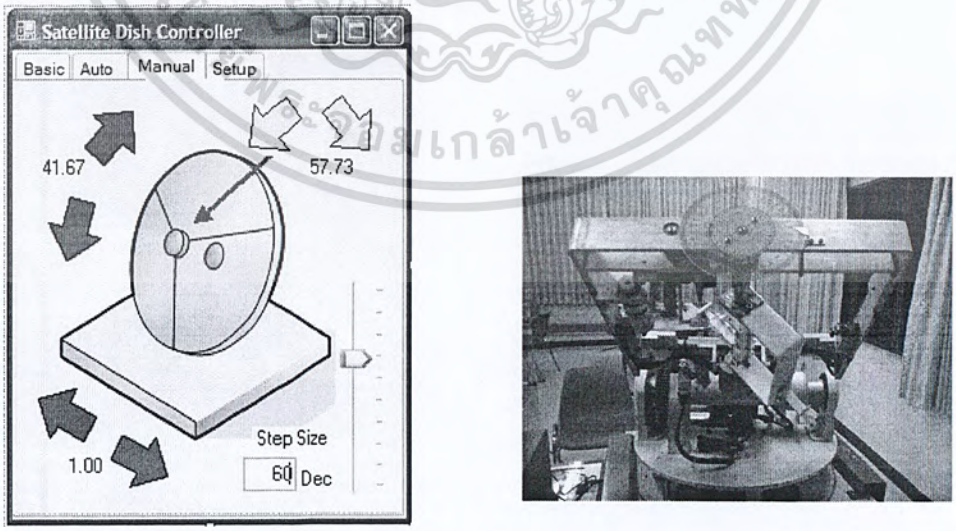
4.2.1 ลำดับขั้นตอนในการทำงาน

- ทดสอบความถูกต้องการเคลื่อนที่ของแกน Polarization

ในการทดลองจะทำการเคลื่อนที่ครั้งละ 10 องศา, 20 องศา และ 30 องศา ตามลำดับ โดยทำการทดลององศาละ 3 ครั้ง เพื่อหาค่าเฉลี่ยซึ่งในการทดลองแต่ละครั้งนั้นจะทำการบันทึกค่าองศาที่กำหนด, องศาที่เคลื่อนที่จริงจากอุปกรณ์วัดมุม และค่าองศาที่แสดงในโปรแกรม



ภาพที่ 4.3 แสดงการเคลื่อนที่ของแกน Polarization ไปที่ 30 องศา



ภาพที่ 4.4 แสดงการเคลื่อนที่ของแกน Polarization ไปที่ 60 องศา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2.2 ผลการทดลองก่อนการแก้ไข

ตารางที่ 4.7 แสดงค่าเฉลี่ยการหมุนแนวแกน Polarization ก่อนการแก้ไขโดยเพิ่ม/ลด ครั้งละ 10 องศา

องศา	Forward				Reward			
	ครั้งที่ 1	ครั้งที่ 2	ครั้งที่ 3	เฉลี่ย	ครั้งที่ 1	ครั้งที่ 2	ครั้งที่ 3	เฉลี่ย
0	0.0	0.0	0.0	0.0				
10	4.0	4.0	6.0	4.7				
20	12.0	12.0	13.0	12.3	0	0	0	0.0
30	19.0	18.0	21.0	19.3	3	3	3	3.0
40	26.0	25.0	28.0	26.3	12	12	12	12.0
50	33.0	31.0	36.0	33.3	20	18	21	19.7
60	39.0	37.0	42.0	39.3	29	27	29	28.3
70	46.0	43.0	50.0	46.3	38	36	38	37.3
80	53.0	49.0	57.0	53.0	46	44	45	45.0
90	59.0	55.0	64.0	59.3	54	52	54	53.3
100	65.0	59.0	71.0	65.0	62	60	62	61.3
110	73.0	70.0	77.0	73.3	70	69	70	69.7
120	79.0	77.0	83.0	79.7	78	76	78	77.3
130	87.0	84.0	90.0	87.0	86	84	86	85.3
140	95.0	92.0	96.0	94.3	94	92	94	93.3

ตารางที่ 4.8 แสดงค่าเฉลี่ยการหมุนแนวแกน Polarization ก่อนการแก้ไขโดยเพิ่ม/ลด ครั้งละ 20 องศา

องศา	Forward				Reward			
	ครั้งที่ 1	ครั้งที่ 2	ครั้งที่ 3	เฉลี่ย	ครั้งที่ 1	ครั้งที่ 2	ครั้งที่ 3	เฉลี่ย
0	0.0	0.0	0.0	0.0				
20	12.0	10.0	12.0	11.3	0.0	0.0	0.0	0.0
40	25.0	24.0	25.0	24.7	7.0	7.0	9.0	7.7
60	39.0	37.0	40.0	38.7	24.0	23.0	26.0	24.3
80	52.0	49.0	54.0	51.7	40.0	39.0	43.0	40.7
100	64.0	61.0	68.0	64.3	55.0	53.0	59.0	55.7
120	76.0	73.0	82.0	77.0	72.0	70.0	77.0	73.0
140	88.0	84.0	96.0	89.3	87.0	85.0	93.0	88.3
160	102.0	99.0	109.0	103.3	102.0	99.0	109.0	103.3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.9 แสดงค่าเฉลี่ยการหมุนแนวแกน Polarization ก่อนการแก้ไขโดยเพิ่ม/ลด ครั้งละ 30 องศา

องศา	Forward				Reward			
	ครั้งที่ 1	ครั้งที่ 2	ครั้งที่ 3	เฉลี่ย	ครั้งที่ 1	ครั้งที่ 2	ครั้งที่ 3	เฉลี่ย
0	0.0	0.0	0.0	0.0				
30	19.0	21.0	21.0	20.3	0.0	0.0	0.0	0.0
60	41.0	42.0	41.0	41.3	22.0	24.0	25.0	23.7
90	59.0	62.0	61.0	60.7	48.0	51.0	52.0	50.3
120	79.0	82.0	81.0	80.7	73.0	77.0	78.0	76
150	98.0	102.0	104.0	101.3	98.0	102.0	104.0	101.3

4.2.3 ผลการทดลองหลังการแก้ไข

ตารางที่ 4.10 แสดงค่าเฉลี่ยการหมุนแนวแกน Polarization หลังการแก้ไขโดยเพิ่ม/ลด ครั้งละ 10 องศา

องศา	Forward				Reward			
	ครั้งที่ 1	ครั้งที่ 2	ครั้งที่ 3	เฉลี่ย	ครั้งที่ 1	ครั้งที่ 2	ครั้งที่ 3	เฉลี่ย
0	0.0	0.0	0.0	0.0				
10	9.5	10.0	10.0	9.83				
20	19.0	20.5	20.0	19.83	0.0	0.0	0.0	0.0
30	28.0	30.5	29.0	29.17	8.0	10.0	10.0	9.3
40	38.0	40.0	40.0	39.33	20.0	22.0	21.0	21.0
50	48.0	50.0	50.0	49.33	32.0	34.0	32.0	32.7
60	58.0	61.0	59.0	59.33	44.0	47.0	45.0	45.3
70	68.0	71.0	59.5	66.17	55.5	57.0	56.0	56.2
80	78.0	80.0	80.0	79.33	67.0	69.0	69.0	68.3
90	88.0	91.0	90.5	89.83	79.0	80.0	80.0	79.7
100	98.5	102	99.5	100.00	89.0	93.0	92.0	91.3
110	108.0	111.0	109.5	109.50	102.0	103.0	102.0	102.3
120	119.0	121.0	119.0	119.67	114.0	117.0	115.0	115.3
130	129.0	132.0	131.0	130.67	126.5	129.0	128.0	127.8
140	139.5	141.0	140.0	140.17	138.0	141.0	140.0	139.7

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.11 แสดงค่าเฉลี่ยการหมุนแนวแกน Polarization หลังการแก้ไขโดยเพิ่ม/ลด ครั้งละ 20 องศา

องศา	Forward				Reward			
	ครั้งที่ 1	ครั้งที่ 2	ครั้งที่ 3	เฉลี่ย	ครั้งที่ 1	ครั้งที่ 2	ครั้งที่ 3	เฉลี่ย
0	0.0	0.0	0.0	0.0				
20	20.0	20.0	21.0	20.3	0.0	0.0	0.0	0.0
40	40.0	41.0	40.0	40.3	22.0	22.5	23.0	22.5
60	60.5	60.0	61.0	60.5	46.0	47.0	48.0	47.0
80	79.0	78.0	79.0	78.7	70.0	71.0	73.0	71.3
100	99.0	99.5	99.0	99.2	94.0	95.0	96.0	95.0
120	121.0	120.0	119.0	120.0	117.0	118.0	119.0	118.0
140	140.0	141.0	140.0	140.3	141.0	142.0	141.0	141.3
160	162.0	163.0	161.0	162.0	162.0	163.0	161.0	162.0

ตารางที่ 4.12 แสดงค่าเฉลี่ยการหมุนแนวแกน Polarization หลังการแก้ไขโดยเพิ่ม/ลด ครั้งละ 30 องศา

องศา	Forward				Reward			
	ครั้งที่ 1	ครั้งที่ 2	ครั้งที่ 3	เฉลี่ย	ครั้งที่ 1	ครั้งที่ 2	ครั้งที่ 3	เฉลี่ย
0	0.0	0.0	0.0	0.0				
30	30.5	30.0	30.0	30.2	0.0	0.0	0.0	0.0
60	61.0	59.0	60.0	60.0	35.0	34.0	35.0	34.7
90	91.0	89.0	89.5	89.8	74.0	73.0	75.0	74.0
120	121.0	120.0	120.5	120.5	112.0	114.0	114.0	113.3
150	152.0	151.0	151.0	151.3	152.0	151.0	151.0	151.3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5
สรุปผลการวิจัยและข้อเสนอแนะ

5.1 สรุปผล

จากการทดลองการเคลื่อนที่ของแกน Elevation ตามองศาที่กำหนดพบว่าผลการตอบสนองนั้นมีค่าใกล้เคียงกับค่าในส่วนของโปรแกรมเดิม ซึ่งมีค่าความคาดเคลื่อนเพียงเล็กน้อยอยู่ในระดับที่สามารถยอมรับได้ แต่ในส่วนของแกน Polarization นั้นพบว่าการเคลื่อนที่มีความคาดเคลื่อนจากความเป็นจริงค่อนข้างมาก ซึ่งหากนำไปใช้งานจริงย่อมส่งผลกระทบต่อประสิทธิภาพอย่างแน่นอน ซึ่งทางคณะผู้จัดทำได้ทำการศึกษาเพื่อหาสาเหตุของความคาดเคลื่อนนั้น และทำการแก้ไขแกน Polarization ให้สามารถเคลื่อนที่ได้ใกล้เคียงกับค่าจริงตามที่แสดงในผลการทดลองในบทที่ 4

5.2 ปัญหาและแนวทางการแก้ไข

การเคลื่อนที่ของแกน Polarization มีความคาดเคลื่อนจากค่าที่ถูกต้องค่อนข้างมากซึ่งมีสาเหตุมาจากจำนวนพัลส์ที่นำไปใช้ในการควบคุมการหมุนของมอเตอร์ ที่ต้องมีการคำนวณอัตราทดเฟืองในการถ่ายทอดกำลังจากมอเตอร์ไปยังตัวอุปกรณ์ มีการใช้ค่าที่ไม่ถูกต้อง ในการแก้ปัญหาจะต้องแก้ไขที่โปรแกรมติดต่อกับผู้ใช้ เนื่องจากโปรแกรมควบคุมจะรับค่าที่เป็นจำนวนพัลส์จากโปรแกรมติดต่อกับผู้ใช้งาน ดังนั้นจึงไม่มีความจำเป็นในการแก้ไขโปรแกรมสำหรับควบคุมการหมุนของมอเตอร์ที่ตัวไมโครคอนโทรลเลอร์

5.3 ข้อเสนอแนะ

โครงงานนี้เป็นโครงงานที่จะมีการนำไปใช้งานจริงกับ Mobile VSAT Vehicle (MVV) ซึ่งโครงงานนี้จะทำในส่วนของการปรับแก้ไขในเรื่องของความแม่นยำในการเคลื่อนที่แกนต่างๆ ของจานรับสัญญาณดาวเทียมให้มีความถูกต้อง เพื่อประสิทธิภาพในการนำไปใช้งานจริงแต่ยังเป็นการทำงานในแบบควบคุมการสั่งงานเองยังไม่สามารถสั่งงานให้ทำงานโดยอัตโนมัติได้ เนื่องจากติดปัญหาเกี่ยวกับการเขียนโปรแกรม เพื่อติดต่อกับอุปกรณ์ระบุตำแหน่งพิกัดบนพื้นโลก (GPS) ซึ่งมีความซับซ้อนมาก แต่ได้มีการออกแบบหน้าต่างในการทำงานไว้รองรับแล้ว ทางคณะผู้จัดทำหวังเป็นอย่างว่าการออกแบบหน้าต่างสำหรับการใช้งาน และเขียนโปรแกรมสำหรับสั่งงานรวมถึงการศึกษาเกี่ยวกับลำดับการทำงานของไมโครคอนโทรลเลอร์สำหรับควบคุมการขับมอเตอร์จะเป็น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ประโยชน์ในการนำไปพัฒนาต่อเพื่อให้มีประสิทธิภาพมากขึ้น โดยอาศัยพื้นฐานจากศึกษาการทำงานและโปรแกรมที่ได้พัฒนาขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

กิตติ ภักดีวัฒนะกุล. 2546. *Visual Basic6 ฉบับ ฐานข้อมูล*. กรุงเทพมหานคร: เคทีพี คอมพ์ แอนด์ คอนซัลท์ จำกัด.

ฉัททวุฒิ พืชผล, พิชิต สันติกุลานนท์และพร้อมเลิศ หล่อวิจิตร. 2547. *คู่มือเรียน Visual Basic 6.* พิมพ์ครั้งที่ 11. กรุงเทพมหานคร : โปรวิชั่น.

สมยศ จุณณะปิยะ. 2543. *การประยุกต์ใช้งานไมโครคอนโทรลเลอร์ตระกูล MCS 51.* พิมพ์ครั้งที่ 3. กรุงเทพมหานคร: สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง.

สมศักดิ์ ศรีขจรเกียรติ. 2541. *เรียน Visual Basic 6 ด้วยตัวเอง.* กรุงเทพมหานคร : บิบลิโอไฟล์ พับลิชชิ่ง.

สัจจะ จรัสรุ่งรวีวร. 2544. *คู่มือการเขียนโปรแกรมและใช้งาน Visual Basic 6.* กรุงเทพมหานคร : อินโฟเพรส.

สุธี พงศาสกุลชัยและ หทัยชนก งามอินทร์. 2549. *คัมภีร์ Visual C# 2005.* กรุงเทพมหานคร : เคทีพี คอมพ์ แอนด์ คอนซัลท์.

ศุภชัย สมพานิช . 2544. *สร้างระบบงานฐานข้อมูลด้วย Visual Basic ฉบับโปรแกรมเมอร์.* กรุงเทพมหานคร : อินโฟเพรส

ศุภชัย สมพานิช. 2546. *คู่มือการเขียนโปรแกรมและใช้งาน Visual C# .NET ฉบับสมบูรณ์.* กรุงเทพมหานคร : อินโฟเพรส.

วิกิพีเดีย. 2553. *ไมโครคอนโทลเลอร์.* [Online]. Available : http://th.wikipedia.org/wiki/ไมโครคอนโทลเลอร์

สคูลดอทเน็ท. 2553. *Microcontroller.* [Online]. Available : http://www.school.net/library/ web contest 2003/100team/ Microcontroller

Ayarafun. 2553. *เริ่มต้นเรียนรู้กับ ไมโครคอลโทลเลอร์.* [Online]. Available : http://www.logicthai.net /mode/14

Engineer001. 2553. *ไมโครคอนโทลเลอร์คืออะไร.* [Online]. Available : http:// www.engineer001.com /index.php?mo=3&art=507518

Global5. 2553. *ความรู้เรื่อง GPS.* [Online]. Available : http://www.global5thailand.com/thai/gps.htm

GPSdeedee. 2553. *GPS คืออะไร.* [Online]. Available : http://www.gpsdeedee.com/index.php?lay=show&ac=article&Ntype=1&Id=538768601

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Inex. 2553. *what is Microcontroller.* [Online]. Available : http://www.inex.co.th/microcontroller/ what is Microcontroller

Ipstar. 2553. *Thai Operator AIS Deploys Femtocell Service on IPSTAR Platform for Emergency Communications.* [Online]. Available : http://www.ipstar.com/en/case_cs_aisfemto.html

Thaimcu. 2553. *รูปแบบการพัฒนาภาษาโปรแกรมของ Microcontroller.* [Online]. Available: http://www.themcu.com/article1.htm

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
using System;
using System.Collections.Generic;
using System.Text;
using System.IO.Ports;
using System.Data;
using System.Runtime.InteropServices;
using System.Windows.Forms;
using Microsoft.Win32;


namespace DishControl
{
    /// <summary>
    /// Axis name enum data type
    /// </summary>
    public enum AxisName
    {
        EL, AZ, PL
    }
    /// <summary>
    /// Control command data type
    /// </summary>
    public enum CmdName
    {
        Move, GetCur, SetCur, GetHome, SetHome
    }
    /// <summary>
    /// Axis data status
    /// </summary>
    public struct AxisStatus
    {
        public bool CmdComplete;
        public bool AxisBusy;
        public double HomePosition;
        public double CurrentPosition;
        public double TargetPosition;
        public int Scaling;
        public double Data;
    }
    /// <summary>
    /// Axis Prooerties
    /// </summary>
    public struct AxisInfo
    {
        public double posSafetyMin;
        public double posSafetyMax;
        public double posMin;
        public double posMax;
        public double posHome;
    }
    /// <summary>
    /// Interface control logic for All Axis
    /// </summary>
    public interface IAxisLogic
    {
        IBaseControl Control
        {
            get;
            set;
        }
        AxisName Name
        {
            get;
            set;
        }
        AxisInfo Info
        {
            get;
            set;
        }
        int Scaling
        {
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            get;
            set;
        }
        bool Busy
        {
            get;
        }
        bool moveAbs(double Dec);
        bool moveRel(double Dec);
        bool stow(int state);
        bool depoly(int state);
        bool startup(int state);
        bool getPosition(ref double Dec);
        bool setPosition(ref double Dec);
        bool getHome(ref double Dec);
        bool setHome(ref double Dec);
    }
    /// <summary>
    /// Interface control logic for Hardware control.
    /// </summary>
    public interface IBaseControl
    {
        string PortName
        {
            get;
            set;
        }
        bool Open();
        void Close();
        bool DishCmd(CmdName cmdName, AxisName axisName, ref double
dataValue);
        AxisStatus getStatus(AxisName axisName);
        void setScaling(AxisName axisName, int scaling);
        int getScaling(AxisName axisName);
        bool isBusy(AxisName axisName);

    }

        /// <summary>
        /// Store the value of Hardware Control port
        /// </summary>
        /// <value>COM1 is the default</value>
        public string BasePort
        {
            get
            {
                RegistryKey Reg = Registry.CurrentUser.CreateSubKey
(rootKey);
                return (string)Reg.GetValue("BasePort", "COM1");
            }
            set
            {
                RegistryKey Reg = Registry.CurrentUser.CreateSubKey
(rootKey);
                Reg.SetValue("BasePort", value);

            }
        }

        /// <summary>
        /// The Opto position of EL Axis
        /// </summary>
        public double OptoElPos
        {
            get
            {
                RegistryKey Reg = Registry.CurrentUser.CreateSubKey
(rootKey);
                return Convert.ToDouble(Reg.GetValue("OptoElPos", "0.0"));
            }
            set
            {
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                RegistryKey Reg = Registry.CurrentUser.CreateSubKey
(rootKey);
                Reg.SetValue("OptoElPos", value);
            }
        }
        /// <summary>
        /// The Optp position of AZ Axis
        /// </summary>
        public double OptoAzPos
        {
            get
            {
                RegistryKey Reg = Registry.CurrentUser.CreateSubKey
(rootKey);
                return Convert.ToDouble(Reg.GetValue("OptoAzPos", "0.0"));
            }
            set
            {
                RegistryKey Reg = Registry.CurrentUser.CreateSubKey
(rootKey);
                Reg.SetValue("OptoAzPos", value);
            }
        }
    }

    /// <summary>
    /// Get/Set the value form File.
    /// </summary>
    public class ProgramConfig
    {
        /// <summary>
        /// Buffer DataSet
        /// </summary>
        public DataSet proDataSet;
        /// <summary>
        ///Data Storage file name
        /// </summary>
        private string fileName;
        public string FileName
        {
            get
            {
                return fileName;
            }
            set
            {
                fileName = value;
            }
        }
        /// <summary>
        /// Class constraction for initial Data.
        /// </summary>
        public ProgramConfig()
        {
            // Get data to DataSet
            proDataSet = new DataSet("PositionDB");
            DataTable dtData = new DataTable("position");
            dtData.Columns.Add("name", typeof(string));
            dtData.Columns.Add("dishLatitude", typeof(double));
            dtData.Columns.Add("dishLongtitude", typeof(double));
            dtData.Columns.Add("dishHeading", typeof(double));
            dtData.Columns.Add("dishHeadingOffset", typeof(double));
            dtData.Columns.Add("dishElevationOffset", typeof(double));
            dtData.Columns.Add("satLongtitude", typeof(double));
            dtData.Columns.Add("satAzimuth", typeof(double));
            dtData.Columns.Add("satElvation", typeof(double));
            dtData.Columns.Add("polPosition", typeof(double));
            proDataSet.Tables.Add(dtData);

            // Get Config to DatatSet
            DataTable dtConfig = new DataTable("env");
            dtConfig.Columns.Add("Name", typeof(string));
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            dtConfig.Columns.Add("satAzimuth", typeof(double));
            dtConfig.Columns.Add("satElvation", typeof(double));
            dtConfig.Columns.Add("polPosition", typeof(double));
            proDataSet.Tables.Add(dtConfig);
        }
        /// <summary>
        /// Initial DataSet using the default value
        /// </summary>
        /// <returns>"true" if the data can write to file else return
"false"</returns>
        public bool InitFile()
        {
            if (proDataSet.Tables["position"].Rows.Count == 0)
            {
                DataRow drData = proDataSet.Tables["position"].NewRow();
                drData["name"] = "New";
                drData["dishLatitude"] = 0;
                drData["dishLongtitude"] = 0;
                drData["dishHeading"] = 0;
                drData["dishHeadingOffset"] = 0;
                drData["dishElevationOffset"] = 0;
                drData["satLongtitude"] = 0;
                drData["satAzimuth"] = 0;
                drData["satElvation"] = 0;
                drData["polPosition"] = 0;
                proDataSet.Tables["position"].Rows.Add(drData);
            }
            return Save();
        }
        /// <summary>
        /// Load Data from  file
        /// </summary>
        /// <returns>"true" if success else return "false"</returns>
        public bool Load()
        {

            try
            {
                proDataSet.ReadXml(fileName);

            }
            catch (Exception )
            {
                return false;
            }
            if (proDataSet.Tables["position"].Rows.Count > 1)
                proDataSet.Tables["position"].Rows.RemoveAt(0);
            return true;
        }
        /// <summary>
        /// Save Data to file
        /// </summary>
        /// <returns>"true" if success to write else return
"false"</returns>
        public bool Save()
        {
            try
            {
                proDataSet.WriteXml(fileName);
            }
            catch (Exception)
            {
                return false;
            }
            return true;
        }
        /// <summary>
        /// Calculate data value of Axis position
        /// </summary>
        /// <returns>"true" if no error to calculate else return
"false"</returns>
        public bool CalData()
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        {
            double Az = 0, El = 0;
            DataRow drData = proDataSet.Tables["position"].Rows[0];
            try
            {
                CalSatelliteDishAngle(ref Az, ref El, (double)drData
["dishLatitude"], (double)drData["dishLongtitude"], (double)drData
["satLongtitude"]);

                Az = Az - (double)drData["dishHeading"] - (double)drData
["dishHeadingOffset"];
                if (Az < -180) Az = Az + 360;
                if (Az > 180) Az = Az - 360;
                drData["satAzimuth"] = Math.Round(Az,3);

                El = Math.Abs(El);
                El = El - (double)drData["dishElevationOffset"];
                drData["satElvation"] = Math.Round(El, 3);

            }
            catch (Exception)
            {
                return false;
            }
            proDataSet.Tables["position"].Rows[0].AcceptChanges();
            return true;
        }
        /// <summary>
        /// Calculate data value of Axis position
        /// </summary>
        /// <param name="Az">AZ position value output</param>
        /// <param name="El">EL Position Value output</param>
        /// <param name="dishLat">Dish Lat</param>
        /// <param name="dishLng">Dish Long</param>
        /// <param name="satLng">satellite Long</param>
        private void CalSatelliteDishAngle(ref double Az, ref double El,
double dishLat, double dishLng, double satLng)
        {
            double beta, alpha;
            beta = dishLng - satLng;
            //beta = satLng - dishLng;
            alpha = dishLat;

            beta = beta / 57.29578;
            alpha = alpha / 57.29578;

            Az = (Math.PI + (Math.Atan((Math.Tan(beta) / Math.Sin
(alpha)))));
            El = Math.Atan((Math.Cos(beta) * Math.Cos(alpha) - 0.178) /
(Math.Sin(beta) / Math.Sin(Az)));

            //El = Math.Atan((a * b - 0.1512) / Math.Sqrt(1 - (a * a) * (b *
b)));

            //Az = CDbl(180) + (Atn(Tan(Beta) / Sin(Alpha)) * CDbl(180) /
(CDbl(22) / CDbl(7)))
            //El = Atn((Cos(Beta) * Cos(Alpha) - 0.178) / (Sin(Beta) / Sin
(Az * ((CDbl(22) / CDbl(7)) / CDbl(180)))))
            //El = El * (CDbl(180) / (CDbl(22) / CDbl(7)))

            Az = Az * 57.29578;
            El = El * 57.29578;
        }

    }
    /// <summary>
    /// Control motion hardware.
    /// </summary>
    public class BaseControl : IBaseControl
    {

        private int Delay = 50;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        /// <summary>
        /// Communication serial port name.
        /// </summary>
        private SerialPort serialPort;
        /// <summary>
        /// Data store for All Axis status.
        /// </summary>
        public AxisStatus[] Axis = new AxisStatus[(int)AxisName.PL + 1];
        /// <summary>
        /// Sermaform to lock transection
        /// </summary>
        private Object thisLock = new Object();
        /// <summary>
        /// Constraction Hardware control class.
        /// </summary>
        public BaseControl()
        {
            //Comport stting
            serialPort = new SerialPort();
            serialPort.PortName = "COM1";
            serialPort.BaudRate = 9600;
            serialPort.DataBits = 8;
            serialPort.StopBits = StopBits.One;
            serialPort.Parity = Parity.None;
            serialPort.ReadTimeout = 1000;
            //serialPort.DataReceived += new SerialDataReceivedEventHandler
(serialPort_DataReceived);

            //Scaling Axis Data
            Axis[(int)AxisName.EL].Scaling = 500;
            Axis[(int)AxisName.AZ].Scaling = 500;
            Axis[(int)AxisName.PL].Scaling = 10;
            Axis[(int)AxisName.EL].AxisBusy = false;
            Axis[(int)AxisName.AZ].AxisBusy = false;
            Axis[(int)AxisName.PL].AxisBusy = false;

        }

//        void serialPort_DataReceived(object sender,
SerialDataReceivedEventArgs e)
        /// <summary>
        /// Communication Package decoder
        /// </summary>
        /// <param name="ReceivedData">return data insize the package
</param>
        /// <returns>sucress or not</returns>
        private bool DataReceived(ref int ReceivedData)
        {
            byte[] buf = new byte[8];
            byte[] next = new byte[1];
            AxisName axisName;
            CmdName cmdName;
            while (serialPort.BytesToRead >= 8)
            {
                if (ReadData(ref buf) && ErrorCheck(buf))
                {
                    byte[] tmp = new byte[4];
                    tmp[0] = buf[2];
                    tmp[1] = buf[3];
                    tmp[2] = buf[4];
                    tmp[3] = buf[5];
                    ReceivedData = ByteToInt(tmp);

                    switch (buf[0])
                    {
                        case 0x01:
                            axisName = AxisName.EL;
                            break;
                        case 0x02:
                            axisName = AxisName.AZ;
                            break;
                        case 0x03:
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                        axisName = AxisName.PL;
                        break;
                    default:
                        return false;
                }

                Axis[(int)axisName].CmdComplete = true;

                switch (buf[1])
                {
                    case 0x11:
                    case 0x12:
                        cmdName = CmdName.Move;
                        if (Axis[(int)axisName].CurrentPosition ==
ReceivedData)
                            Axis[(int)axisName].AxisBusy = false;
                        Axis[(int)axisName].TargetPosition =
ReceivedData;
                        break;
                    case 0x10:
                        cmdName = CmdName.GetCur;
                        if (Axis[(int)axisName].CurrentPosition ==
ReceivedData)
                            Axis[(int)axisName].AxisBusy = false;
                        Axis[(int)axisName].CurrentPosition =
ReceivedData;
                        break;
                    case 0x30:
                        cmdName = CmdName.GetHome;
                        Axis[(int)axisName].HomePosition = ReceivedData;
                        break;
                    case 0x13:
                        cmdName = CmdName.SetCur;
                        break;
                    case 0x31:
                        cmdName = CmdName.SetHome;
                        break;
                    default:
                        return false;
                }
            }
            else
            {
                ReadData(ref next);
                BufferShift(ref buf,next);
            }
        }
        return true;
    }
    /// <summary>
    /// Deconstraction class for clase Communication port
    /// </summary>
    ~BaseControl()
    {

        if (serialPort.IsOpen) serialPort.Close();
    }
    /// <summary>
    /// Communication port
    /// </summary>
    public string PortName
    {
        get
        {
            return serialPort.PortName;
            //throw new System.NotImplementedException();
        }
        set
        {
            serialPort.PortName = value;
        }
    }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        /// <summary>
        /// Get Status of Axis data.
        /// </summary>
        /// <param name="axisName">Axis Name to geting status</param>
        /// <returns>Status of Axis</returns>
        public AxisStatus getStatus(AxisName axisName)
        {
            return Axis[(int)axisName];
        }
        /// <summary>
        /// Set Scaling value of Axis
        /// </summary>
        /// <param name="axisName">Axis name to seting scaling value</param>
        /// <param name="scaling">scaling value</param>
        public void setScaling(AxisName axisName, int scaling)
        {
            Axis[(int)axisName].Scaling = scaling;
        }
        /// <summary>
        /// Get Scaling value of Axis
        /// </summary>
        /// <param name="axisName">Axis name to geting scaling value</param>
        /// <returns>scaling value</returns>
        public int getScaling(AxisName axisName)
        {
            return Axis[(int)axisName].Scaling;
        }
        /// <summary>
        /// Chacking Axis Busy state.
        /// </summary>
        /// <param name="axisName">Axis name to checking</param>
        /// <returns>Is Busy?</returns>
        public bool isBusy(AxisName axisName)
        {
            return Axis[(int)axisName].AxisBusy;
        }
        /// <summary>
        /// Open Communication serial port
        /// </summary>
        /// <returns>Can Open COM port</returns>
        public bool Open()
        {
            if
                (serialPort.IsOpen) serialPort.Close();

            try
            {
                serialPort.Open();
            }
            catch (Exception)
            {
                return false;
            }
            return true;

        }
        /// <summary>
        /// Close Communication serial port
        /// </summary>
        public void Close()
        {
            if (serialPort.IsOpen) serialPort.Close();
        }
        /// <summary>
        /// Calculate Scaling value for send to hardware.
        /// </summary>
        /// <param name="axisName">Axis name for select scaling value
</param>
        /// <param name="dec">Data was convert</param>
        private void Scaling(AxisName axisName, ref double dec)
        {
            double scale;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            scale = Axis[(int)axisName].Scaling;
            dec = scale * dec;
        }
        /// <summary>
        /// Convert Hardware axis counter to axis position with scaling
value
        /// </summary>
        /// <param name="axisName">Axis name for select descaling value
</param>
        /// <param name="dec">Data was convert</param>
        private void DeScaling(AxisName axisName, ref double dec)
        {
            double scale;
            scale = Axis[(int)axisName].Scaling;
            dec = dec / scale;
        }
        /// <summary>
        /// Data sending to hardware
        /// </summary>
        /// <param name="buffer">data buffer</param>
        /// <returns>Sucress process?</returns>
        private bool SendData(byte[] buffer)
        {
            try
            {
                serialPort.Write(buffer, 0, buffer.Length);

            }
            catch (Exception)
            {
                return false;
            }
            return true;
        }
        /// <summary>
        /// Reading data from Hardware
        /// </summary>
        /// <param name="buffer">Data Buffer</param>
        /// <returns>Sucress?</returns>
        private bool ReadData(ref byte[] buffer)
        {
            try
            {
                serialPort.Read(buffer, 0, buffer.Length);
            }
            catch (Exception )
            {
                return false;
            }
            return true;
        }
        /// <summary>
        /// Data Converter Integer to Byte
        /// </summary>
        /// <param name="dec">Interger Data</param>
        /// <returns>Buffer Byte</returns>
        private byte[] IntToByte(int dec)
        {
            byte[] buf = new byte[4];
            buf[3] = (byte)(dec % 0x100);
            dec = dec >> 8;
            buf[2] = (byte)(dec % 0x100);
            dec = dec >> 8;
            buf[1] = (byte)(dec % 0x100);
            dec = dec >> 8;
            buf[0] = (byte)(dec % 0x100);
            return buf;

        }
        /// <summary>
        /// Data Convert Byte to Int
        /// </summary>
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        /// <param name="buf">Buffer Byte data</param>
        /// <returns>Interger output</returns>
        private int ByteToInt(byte[] buf)
        {
            int dec;
            dec = buf[0];
            dec = dec << 8;
            dec = dec + buf[1];
            dec = dec << 8;
            dec = dec + buf[2];
            dec = dec << 8;
            dec = dec + buf[3];
            return dec;

        }
        /// <summary>
        /// Check sum data package genalator.
        /// </summary>
        /// <param name="buf">Data package that you want to genalate error
value</param>
        /// <returns>Byte Error check sum</returns>
        private byte ErrorCheckGen(byte[] buf)
        {
            byte sum = 0;
            for (int i = 0; i < buf.Length; i++)
            {
                sum = (byte)(sum ^ buf[i]);
            }
            return sum;
        }
        /// <summary>
        /// Package check Error
        /// </summary>
        /// <param name="buf">Package Data block</param>
        /// <returns>Check sum of the data package input</returns>
        private bool ErrorCheck(byte[] buf)
        {
            byte sum = 0;
            bool zeroAll = true;
            for (int i = 0; i < buf.Length -1; i++)
            {
                sum = (byte)(sum ^ buf[i]);
                if (buf[i] != 0) zeroAll = false;
            }
            if (zeroAll)
                return false;
            else
                return (sum==buf[buf.Length-1]);
        }
        /// <summary>
        /// Shift Buffer data
        /// </summary>
        /// <param name="buf">Input/Output buffer to shift</param>
        /// <param name="data">data to shift into buffer</param>
        private void BufferShift(ref byte[] buf, byte[] data)
        {
            int bufL = buf.Length;
            int dataL = data.Length;
            if (bufL > dataL)
            {
                int delta = bufL - dataL;
                for (int i = 0; i < bufL; i++)
                {
                    if (i < delta) buf[i] = buf[i + dataL];
                    else buf[i] = data[i - delta];

                }
            }
        }
        /// <summary>
        /// Create data package for sending to Hardware controller.
        /// </summary>
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        /// <param name="cmdName">Command name type</param>
        /// <param name="axisName">Axis Name</param>
        /// <param name="input">Value parameter</param>
        /// <returns>success?</returns>
        public bool DishCmd(CmdName cmdName, AxisName axisName, ref double
input)
        {
            double dataValue = input;
            //double data = dataValue;
            byte[] buf = new byte[7];
            int ReceivedData = new int();

            switch (axisName)
            {
                case AxisName.EL:
                    buf[0] = 0x01;
                    break;
                case AxisName.AZ:
                    buf[0] = 0x02;
                    break;
                case AxisName.PL:
                    buf[0] = 0x03;
                    break;
                default:
                    break;
            }

            switch (cmdName)
            {
                case CmdName.Move:
                    if (dataValue < 0) buf[1] = 0x12; else buf[1] = 0x11;
                    Scaling(axisName, ref dataValue);
                    if (dataValue < 0) dataValue = -dataValue;
                    Axis[(int)axisName].AxisBusy = true;
                    break;
                case CmdName.GetCur:
                    buf[1] = 0x10;
                    dataValue = 0;
                    break;
                case CmdName.GetHome:
                    buf[1] = 0x30;
                    dataValue = 0;
                    break;
                case CmdName.SetCur:
                    buf[1] = 0x13;
                    break;
                case CmdName.SetHome:
                    Scaling(axisName, ref dataValue);
                    buf[1] = 0x31;
                    break;
            }
            int data = Convert.ToInt32(dataValue);
            byte[] bufData = IntToByte(data);
            buf[2] = bufData[0];
            buf[3] = bufData[1];
            buf[4] = bufData[2];
            buf[5] = bufData[3];

            buf[6] = ErrorCheckGen(buf);

            Axis[(int)axisName].CmdComplete = false;
                SendData(buf);
                int i = Environment.TickCount;
                do
                {
                    if ((Environment.TickCount - i) > Delay)
                        break;
                } while (!Axis[(int)axisName].CmdComplete);
                DataReceived(ref ReceivedData);
                input = ReceivedData;
            DeScaling(axisName, ref input);
            return true;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        }

    }
    /// <summary>
    /// Control logic of Axis
    /// </summary>
    public abstract class AxisLogic : IAxisLogic
    {
        /// <summary>
        /// Hardware subprogram for Control axis movement
        /// </summary>
        private IBaseControl baseControl;
        private AxisName axisName;
        private AxisInfo axisInfo;
        private int delayCommand = 200;
        /// <summary>
        /// Delay time between two control command
        /// </summary>
        public void delay()
        {
            int i = Environment.TickCount;
            do
            {
                if ((Environment.TickCount - i) > delayCommand)
                    break;
            } while (true);

        }
        /// <summary>
        /// Construction class
        /// </summary>
        /// <param name="BC">Base contr</param>
        /// <param name="Na">Axis name</param>
        /// <param name="Scaling">Scaling value</param>
        /// <param name="Info">Status of Axis</param>
        public AxisLogic(IBaseControl BC, AxisName Na,int Scaling,AxisInfo
Info)
        {
            baseControl = BC;
            axisName = Na;
            axisInfo = Info;
            baseControl.setScaling(axisName, Scaling);
        }

        public IBaseControl Control
        {
            get
            {
                return baseControl;
            }
            set
            {
                baseControl = value;
            }
        }
        public AxisName Name
        {
            get
            {
                return axisName;
            }
            set
            {
                axisName = value;
            }
        }
        public AxisInfo Info
        {
            get
            {
                return axisInfo;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        }
        set
        {
            axisInfo = value;
        }
    }

    public int Scaling
    {
        get
        {
            return baseControl.getScaling(axisName) ;
        }
        set
        {
            baseControl.setScaling(axisName, value);
        }
    }
    /// <summary>
    /// Check the hardware ready to control?
    /// </summary>
    public bool Busy
    {
        get
        {
            double Dec1 = 0;
            double Dec2 = 0;
            delay();
            baseControl.DishCmd(CmdName.GetCur, axisName, ref Dec1);
            delay();
            baseControl.DishCmd(CmdName.GetCur, axisName, ref Dec2);

            return (Dec1 != Dec2);
        }
    }
    /// <summary>
    /// abstract stow subprogram
    /// </summary>
    /// <param name="state">Parameter for control stow state</param>
    /// <returns>success or not?</returns>
    public abstract bool stow(int state);
    /// <summary>
    /// abstract depoly subprogram
    /// </summary>
    /// <param name="state">Parameter fot control depoly state</param>
    /// <returns></returns>
    public abstract bool depoly(int state);
    /// <summary>
    /// abstract startup subprogram
    /// </summary>
    /// <param name="state">Parameter fot control startup state</param>
    /// <returns></returns>
    public abstract bool startup(int state);

    /// <summary>
    /// Moving posiotion in Abs mode
    /// </summary>
    /// <param name="Dec">target position</param>
    /// <returns>"true" if success process.</returns>
    public bool moveAbs(double Dec)
    {
        double Data = 0;
        delay();
        if (baseControl.DishCmd(CmdName.GetCur, axisName, ref Data))
        {
            Data = Dec - Data;
            delay();
            if (!baseControl.isBusy(axisName))
            {
                delay();
                if (baseControl.DishCmd(CmdName.Move, axisName, ref
Data))
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                    return true;
                }
            }
            return false;
        }
        /// <summary>
        /// Moving position in relative mode
        /// </summary>
        /// <param name="Dec">Relative value</param>
        /// <returns>"true" if success process.</returns>
        public bool moveRel(double Dec)
        {
            delay();
            if (baseControl.DishCmd(CmdName.Move, axisName, ref Dec))
                return true;
            else
                return false;
        }
        /// <summary>
        /// Get the current position.
        /// </summary>
        /// <param name="Dec">output current position</param>
        /// <returns>if system can get data from hardware then return
"true" else "false"</returns>
        public bool getPosition(ref double Dec)
        {
            delay();
            if (baseControl.DishCmd(CmdName.GetCur, axisName, ref Dec))
                return true;
            else
                return false;
        }
        /// <summary>
        /// Set the current position.
        /// </summary>
        /// <param name="Dec">Input/Output position</param>
        /// <returns>if system can set data from hardware then return
"true" else "false"</returns>
        public bool setPosition(ref double Dec)
        {
            delay();
            if (baseControl.DishCmd(CmdName.SetCur, axisName, ref Dec))
                return true;
            else
                return false;
        }
        /// <summary>
        /// Get Axis Home position
        /// </summary>
        /// <param name="Dec">return Home Axis position</param>
        /// <returns>if can get return "true"</returns>
        public bool getHome(ref double Dec)
        {
            delay();
            if (baseControl.DishCmd(CmdName.GetHome, axisName, ref Dec))
                return true;
            else
                return false;
        }
        /// <summary>
        /// Set Axis Home position
        /// </summary>
        /// <param name="Dec">Home Axis position setting</param>
        /// <returns>if can set return "true"</returns>
        public bool setHome(ref double Dec)
        {
            delay();
            if (baseControl.DishCmd(CmdName.SetHome, axisName, ref Dec))
                return true;
            else
                return false;
        }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    }
    /// <summary>
    /// The axis hardware for EL axis
    /// </summary>
    public class AxisLogicSensorHome : AxisLogic //EL
    {
        /// <summary>
        /// Construction class and Initial base clase
        /// </summary>
        /// <param name="BC">Base Control</param>
        /// <param name="Name">Axis name</param>
        /// <param name="Scaling">axis scaling value</param>
        /// <param name="Info">axis status</param>
        public AxisLogicSensorHome(IBaseControl BC,AxisName Name,int
Scaling,AxisInfo Info)
            : base(BC,Name,Scaling,Info)
        {
        }
        /// <summary>
        /// stow control state
        /// </summary>
        /// <param name="state">state Action</param>
        /// <returns>success</returns>
        public override bool stow(int state)
        {
            switch (state)
            {
                case -1:
                    moveAbs(Info.posSafetyMin);
                    break;
                case 0:
                    moveRel(-360);
                    break;
            }
            return true;
        }
        /// <summary>
        /// depoly control state
        /// </summary>
        /// <param name="state">state action</param>
        /// <returns>success</returns>
        public override bool depoly(int state)
        {
            switch (state)
            {
                case 0:
                    moveAbs(Info.posHome - 5);
                    break;
            }
            return true;
        }
        /// <summary>
        /// Initial and recovery dish state
        /// </summary>
        /// <param name="state">state action</param>
        /// <returns>success</returns>
        public override bool startup(int state)
        {
            switch (state)
            {
                case 0:
                    moveRel(100);
                    break;
                case 1:
                    moveAbs(Info.posHome - 5);
                    break;
            }
            return true;
        }
    }
    /// <summary>
    /// The axis hardware for AZ axis
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    /// </summary>
    public class AxisLogicOptoHome : AxisLogic //AZ
    {
        /// <summary>
        /// Construction class and Initial base clase
        /// </summary>
        /// <param name="BC">Base Control</param>
        /// <param name="Name">Axis name</param>
        /// <param name="Scaling">axis scaling value</param>
        /// <param name="Info">axis status</param>
        public AxisLogicOptoHome(IBaseControl BC, AxisName Name, int
Scaling, AxisInfo Info)
            : base(BC, Name, Scaling, Info)
        {
        }
        /// <summary>
        /// stow control state
        /// </summary>
        /// <param name="state">state Action</param>
        /// <returns>success</returns>
        public override bool stow(int state)
        {
            double Adj = 1;
            switch (state)
            {
                case -1:
                    moveAbs(Info.posHome - 5);
                    break;
                case 0:
                    moveAbs(Info.posHome + Adj);
                    break;
            }
            return true;
        }
        /// <summary>
        /// depoly control state
        /// </summary>
        /// <param name="state">state action</param>
        /// <returns>success</returns>
        public override bool depoly(int state)
        {
            double Adj = 1;
            switch (state)
            {
                case 0:
                    moveAbs(Info.posHome - 5);
                    break;
                case 1:
                    moveAbs(Info.posHome + Adj);
                    break;
            }
            return true;
        }
        /// <summary>
        /// Initial and recovery dish state
        /// </summary>
        /// <param name="state">state action</param>
        /// <returns>success</returns>
        public override bool startup(int state)
        {
            switch (state)
            {
                case 0:
                    moveRel(-10);
                    break;
                case 1:
                    moveRel(20);
                    break;
                case 2:
                    moveAbs(Info.posHome);
                    break;
            }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            return true;
        }
    }
    /// <summary>
    /// The axis hardware for PL axis
    /// </summary>
    public class AxisLogicLessHome : AxisLogic //PL
    {
        /// <summary>
        /// Construction class and Initial base clase
        /// </summary>
        /// <param name="BC">Base Control</param>
        /// <param name="Name">Axis name</param>
        /// <param name="Scaling">axis scaling value</param>
        /// <param name="Info">axis status</param>
        public AxisLogicLessHome(IBaseControl BC, AxisName Name, int
Scaling, AxisInfo Info)
            : base(BC, Name, Scaling, Info)
        {
        }
        /// <summary>
        /// stow control state
        /// </summary>
        /// <param name="state">state Action</param>
        /// <returns>success</returns>
        public override bool stow(int state)
        {
            switch (state)
            {
                case 0:
                    moveRel(-360);
                    break;
            }
            return true;
        }
        /// <summary>
        /// depoly control state
        /// </summary>
        /// <param name="state">state action</param>
        /// <returns>success</returns>
        public override bool depoly(int state)
        {
            return true;
        }
        /// <summary>
        /// Initial and recovery dish state
        /// </summary>
        /// <param name="state">state action</param>
        /// <returns>success</returns>
        public override bool startup(int state)
        {
            switch (state)
            {
                case -1:                 // if no zero
                    moveRel(-100);
                    break;
                case 0:                 //fast forward
                    moveRel(10);
                    break;
                case 1:                 //Jog mode
                    moveRel(1);
                    break;
            }

            return true;
        }
    }
    /// <summary>
    /// Position logic control class
    /// </summary>
    public class PositionLogic
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
{
    private ProgramRegisty programRegisty;
    private IBaseControl baseControl;
    public IGPSControl gpsControl;
    public IAxisLogic EL;
    public IAxisLogic AZ;
    public IAxisLogic PL;

    /// <summary>
    /// Construction class and initial value
    /// </summary>
    public PositionLogic()
    {
        /// File configulation
        programRegisty = new ProgramRegisty();
        /// Hardware movment control
        baseControl = new BaseControl();
        baseControl.PortName = programRegisty.BasePort;
        /// GPS control
        gpsControl = new GPSControl();
        gpsControl.PortName = programRegisty.GPSPort;

        /// initial Axis default value
        AxisInfo Info = new AxisInfo();
        Info.posMin = -97;      //-70;
        Info.posSafetyMin = 17;   //15;
        Info.posSafetyMax = 180;
        Info.posMax = 180;
        Info.posHome = programRegisty.OptoElPos;
        EL = new AxisLogicSensorHome(baseControl,AxisName.EL,500,Info);
        Info.posMin = -180;
        Info.posSafetyMin = -180;
        Info.posSafetyMax = 180;
        Info.posMax = 180;
        Info.posHome = programRegisty.OptoAzPos;
        AZ = new AxisLogicOptoHome(baseControl, AxisName.AZ, 500, Info);
        Info.posMin = -20;
        Info.posSafetyMin = -20;
        Info.posSafetyMax = 175;
        Info.posMax = 175;
        PL = new AxisLogicLessHome(baseControl,AxisName.PL,11,Info);
    }

    public string BasePort
    {
        get
        {
            return programRegisty.BasePort;
        }
        set
        {
            programRegisty.BasePort = value;
        }
    }
        public double OptoElPos
    {
        get
        {
            return programRegisty.OptoElPos;
        }
        set
        {
            programRegisty.OptoElPos = value;
        }
    }
    public double OptoAzPos
    {
        get
        {
            return programRegisty.OptoAzPos;
        }
        set
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            {
                programRegisty.OptoAzPos = value;
            }
        }

        /// <summary>
        /// Show Stow form to input from actor
        /// </summary>
        /// <returns>submetion Dialog?</returns>
        public bool Stow()
        {
            Stow form = new Stow(this);
            if (form.ShowDialog() == DialogResult.OK) return true;
            return false;
        }
        /// <summary>
        /// Show Depoly form to input from actor
        /// </summary>
        /// <returns>submetion Dialog?</returns>
        public bool Depoly()
        {
            Deploy form = new Deploy(this);
            if(form.ShowDialog() == DialogResult.OK) return true;
            return false;
        }
        /// <summary>
        /// Show Startup and Recovery form to input from actor
        /// </summary>
        /// <returns>submetion Dialog?</returns>
        public bool Startup()
        {
            Startup form = new Startup(this);
            if (form.ShowDialog() == DialogResult.OK) return true;
            return false;

        }

    }

}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Agilent Technologies

Quick Assembly Two and Three Channel Optical Encoders

Technical Data

HEDM-550x/560x
HEDS-550x/554x
HEDS-560x/564x

Features

- **Two Channel Quadrature Output with Optional Index Pulse**
- **Quick and Easy Assembly**
- **No Signal Adjustment Required**
- **External Mounting Ears Available**
- **Low Cost**
- **Resolutions Up to 1024 Counts Per Revolution**
- **Small Size**
- **-40°C to 100°C Operating Temperature**
- **TTL Compatible**
- **Single 5 V Supply**

Description

The HEDS-5500/5540, HEDS-5600/5640, and HEDM-5500/5600 are high performance, low cost, two and three channel optical incremental encoders. These encoders emphasize high reliability, high resolution, and easy assembly.

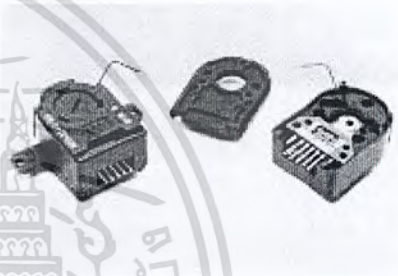
Each encoder contains a lensed LED source, an integrated circuit with detectors and output circuitry, and a codewheel which rotates between the emitter and detector IC. The outputs of the HEDS-5500/5600 and HEDM-5500/5600 are two square waves in quadrature. The HEDS-5540 and 5640 also have a third channel index output in addition to the two channel quadrature. This index output is a 90 electrical degree, high true index pulse which is generated once for each full rotation of the codewheel.

The HEDS series utilizes metal codewheels, while the HEDM series utilizes a film codewheel allowing for resolutions to 1024 CPR. The HEDM series is nont available with a third channel index.

These encoders may be quickly and easily mounted to a motor. For larger diameter motors, the HEDM-5600, and HEDS-5600/5640 feature external mounting ears.

The quadrature signals and the index pulse are accessed through five 0.025 inch square pins located on 0.1 inch centers.

Standard resolutions between 96 and 1024 counts per revolution are presently available. Consult local Agilent sales representatives for other resolutions.



Applications

The HEDS-5500, 5540, 5600, 5640, and the HEDM-5500, 5600 provide motion detection at a low cost, making them ideal for high volume applications. Typical applications include printers, plotters, tape drives, positioning tables, and automatic handlers.

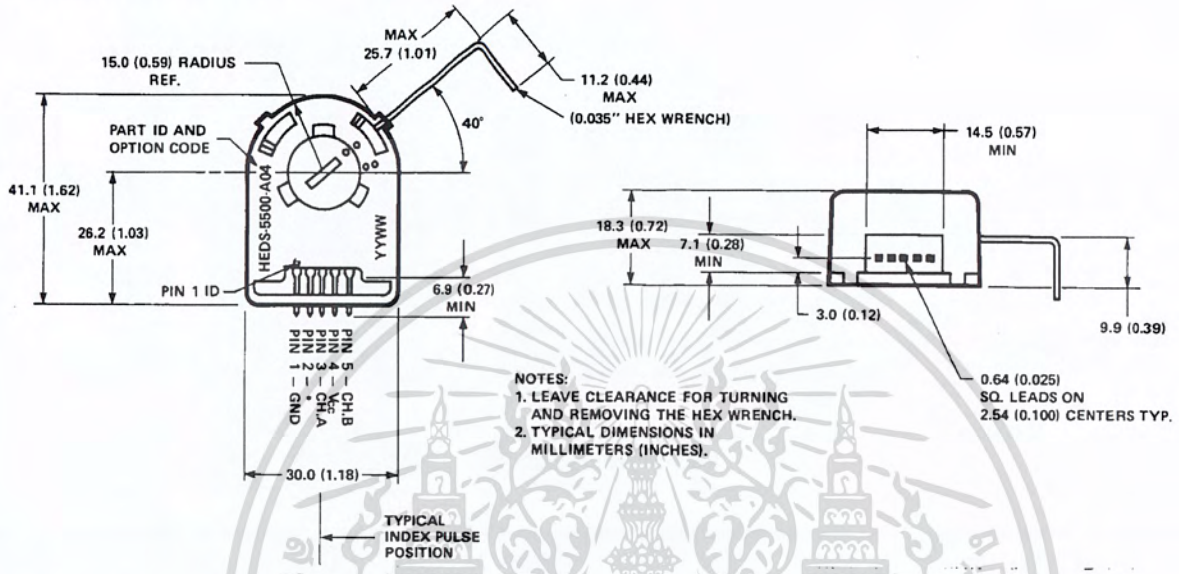
Note: Agilent Technologies encoders are not recommended for use in safety critical applications. Eg. ABS braking systems, power steering, life support systems and critical care medical equipment. Please contact sales representative if more clarification is needed.

ESD WARNING: NORMAL HANDLING PRECAUTIONS SHOULD BE TAKEN TO AVOID STATIC DISCHARGE.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

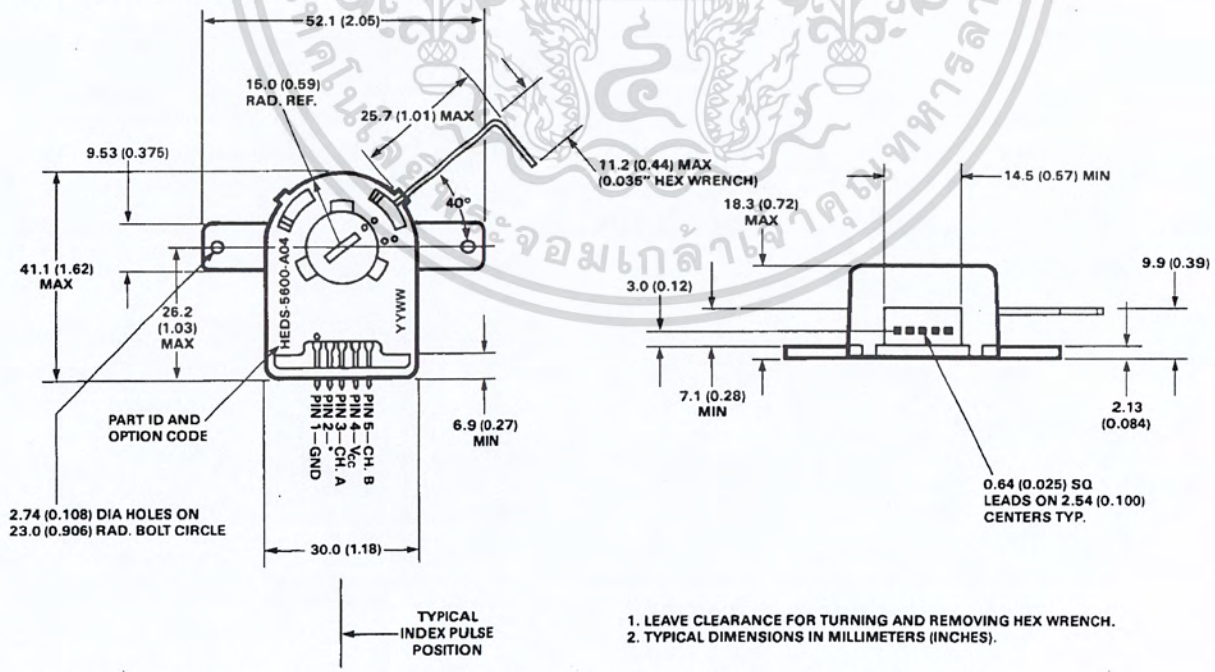
Package Dimensions

HEDS-5500/5540, HEDM-5500



***Note:** For the HEDS-5500 and HEDM-5500, Pin #2 is a No Connect. For the HEDS-5540, Pin #2 is CH. I, the index output.

HEDS-5600/5640, HEDM-5600



***Note:** For the HEDS-5600 and HEDM-5600, Pin #2 is a No Connect. For the HEDS-5640, Pin #2 is CH. I, the index output.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Theory of Operation

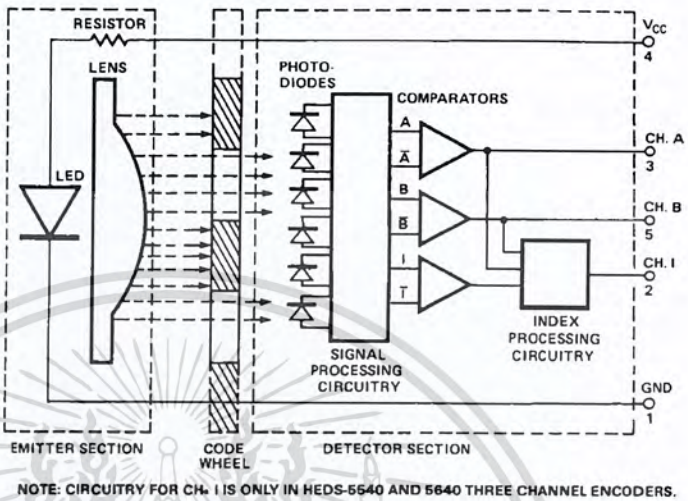
The HEDS-5500, 5540, 5600, 5640, and HEDM-5500, 5600 translate the rotary motion of a shaft into either a two- or a three-channel digital output.

As seen in the block diagram, these encoders contain a single Light Emitting Diode (LED) as its light source. The light is collimated into a parallel beam by means of a single polycarbonate lens located directly over the LED. Opposite the emitter is the integrated detector circuit. This IC consists of multiple sets of photodetectors and the signal processing circuitry necessary to produce the digital waveforms.

The codewheel rotates between the emitter and detector, causing the light beam to be interrupted by the pattern of spaces and bars on the codewheel. The photodiodes which detect these interruptions are arranged in a pattern that corresponds to the radius and design of the codewheel. These detectors are also spaced such that a light period on one pair of detectors corresponds to a dark period on the adjacent pair of detectors. The photodiode outputs are then fed through the signal processing circuitry resulting in A, $\overline{A}$, B and $\overline{B}$ (also I and $\overline{I}$ in the HEDS-5540 and 5640). Comparators receive these signals and produce the final outputs for channels A and B. Due to this integrated phasing technique, the digital output of channel A is in quadrature with that of channel B (90 degrees out of phase).

In the HEDS-5540 and 5640, the output of the comparator for I and $\overline{I}$ is sent to the index processing circuitry along with the outputs of channels A and B.

Block Diagram



NOTE: CIRCUITRY FOR CH. I IS ONLY IN HEDS-5540 AND 5640 THREE CHANNEL ENCODERS.

The final output of channel I is an index pulse P_O which is generated once for each full rotation of the codewheel. This output P_O is a one state width (nominally 90 electrical degrees), high true index pulse which is coincident with the low states of channels A and B.

Definitions

Count (N): The number of bar and window pairs or counts per revolution (CPR) of the codewheel.

One Cycle (C): 360 electrical degrees (°e), 1 bar and window pair.

One Shaft Rotation: 360 mechanical degrees, N cycles.

Position Error ($\Delta\Theta$): The normalized angular difference between the actual shaft position and the position indicated by the encoder cycle count.

Cycle Error (ΔC): An indication of cycle uniformity. The difference between an observed shaft angle which gives rise to one electrical cycle, and the nominal angular increment of 1/N of a revolution.

Pulse Width (P): The number of electrical degrees that an output is high during 1 cycle. This value is nominally 180°e or 1/2 cycle.

Pulse Width Error (ΔP): The deviation, in electrical degrees, of the pulse width from its ideal value of 180°e.

State Width (S): The number of electrical degrees between a transition in the output of channel A and the neighboring transition in the output of channel B. There are 4 states per cycle, each nominally 90°e.

State Width Error (ΔS): The deviation, in electrical degrees, of each state width from its ideal value of 90°e.

Phase (ϕ): The number of electrical degrees between the center of the high state of channel A and the center of the high state of channel B. This value is nominally 90°e for quadrature output.

Phase Error ($\Delta\phi$): The deviation of the phase from its ideal value of 90°e.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

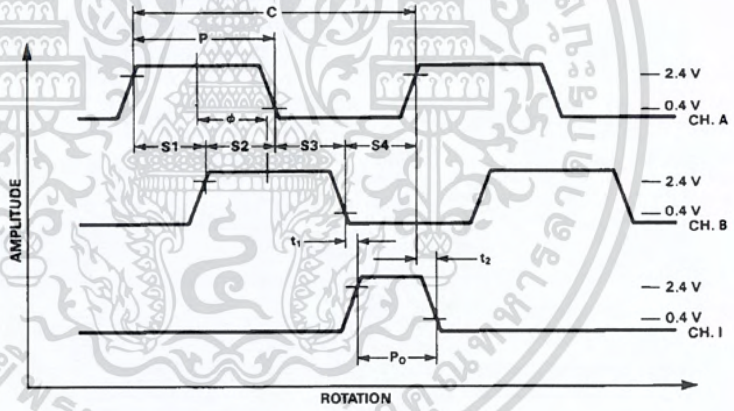
Absolute Maximum Ratings

Parameter	HEDS-55XX/56XX	HEDM-550X/560X
Storage Temperature, T_S	-40°C to 100°C	-40°C to +70°C
Operating Temperature, T_A	-40°C to 100°C	-40°C to +70°C
Supply Voltage, V_{CC}	-0.5 V to 7 V	-0.5 V to 7 V
Output Voltage, V_O	-0.5 V to V_{CC}	-0.5 V to V_{CC}
Output Current per Channel, I_{OUT}	-1.0 mA to 5 mA	-1.0 mA to 5 mA
Vibration	20 g, 5 to 1000 Hz	20 g, 5 to 1000 Hz
Shaft Axial Play	± 0.25 mm (± 0.010 in.)	± 0.175 mm (± 0.007 in.)
Shaft Eccentricity Plus Radial Play	0.1 mm (0.004 in.) TIR	0.04 mm (0.0015 in.) TIR
Velocity	30,000 RPM	30,000 RPM
Acceleration	250,000 rad/sec^2	250,000 rad/sec^2

Direction of Rotation: When the codewheel rotates in the counter-clockwise direction (as viewed from the encoder end of the motor), channel A will lead channel B. If the codewheel rotates in the clockwise direction, channel B will lead channel A.

Index Pulse Width (P_O): The number of electrical degrees that an index output is high during one full shaft rotation. This value is nominally 90°e or 1/4 cycle.

Output Waveforms



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Recommended Operating Conditions

Parameter	Symbol	Min.	Typ.	Max.	Units	Notes
Temperature HEDS Series	T_A	-40		100	°C	
Temperature HEDM Series	T_A	-40		70	°C	non-condensing atmosphere
Supply Voltage	V_{CC}	4.5	5.0	5.5	Volts	Ripple < 100 mV_{p-p}
Load Capacitance	C_L			100	pF	2.7 kΩ pull-up
Count Frequency	f			100	kHz	Velocity (rpm) x N/60
Shaft Perpendicularity Plus Axial Play (HEDS Series)				± 0.25 (± 0.010)	mm (in.)	6.9 mm (0.27 in.) from mounting surface
Shaft Eccentricity Plus Radial Play (HEDS Series)				0.04 (0.0015)	mm (in.) TIR	6.9 mm (0.27 in.) from mounting surface
Shaft Perpendicularity Plus Axial Play (HEDM Series)				± 0.175 (± 0.007)	mm (in.)	6.9 mm (0.27 in.) from mounting surface
Shaft Eccentricity Plus Radial Play(HEDM Series)				0.04 (0.0015)	mm (in.) TIR	6.9 mm (0.27 in.) from mounting surface

Note: The module performance is guaranteed to 100 kHz but can operate at higher frequencies. 2.7 kΩ pull-up resistors required for HEDS-5540 and 5640.

Encoding Characteristics

Encoding Characteristics over Recommended Operating Range and Recommended Mounting Tolerances unless otherwise specified. Values are for the worst error over the full rotation.

Part No.	Description		Sym.	Min.	Typ.*	Max.	Units
HEDS-5500 HEDS-5600 (Two Channel)	Pulse Width Error		ΔP		7	45	°e
	Logic State Width Error		ΔS		5	45	°e
	Phase Error		Δϕ		2	20	°e
	Position Error		ΔΘ		10	40	min. of arc
	Cycle Error		ΔC		3	5.5	°e
HEDM-5500 HEDM-5600 (Two Channel)	Pulse Width Error		ΔP		10	45	°e
	Logic State Width Error		ΔS		10	45	°e
	Phase Error		Δϕ		2	15	°e
	Position Error		ΔΘ		10	40	min. of arc
	Cycle Error		ΔC		3	7.5	°e
HEDS-5540 HEDS-5640 (Three Channel)	Pulse Width Error		ΔP		5	35	°e
	Logic State Width Error		ΔS		5	35	°e
	Phase Error		Δϕ		2	15	°e
	Position Error		ΔΘ		10	40	min. of arc
	Cycle Error		ΔC		3	5.5	°e
	Index Pulse Width		P_O	55	90	125	°e
	CH. I rise after CH. A or CH. B fall	-40°C to +100°C	t_2	-300	100	250	ns
	CH. I fall after CH. B or CH. A rise	-40°C to +100°C	t_2	70	150	1000	ns

Note: See Mechanical Characteristics for mounting tolerances.
*Typical values specified at V_{CC} = 5.0 V and 25°C.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Electrical Characteristics

Electrical Characteristics over Recommended Operating Range.

Part No.	Parameter	Sym.	Min.	Typ.*	Max.	Units	Notes
HEDS-5500 HEDS-5600	Supply Current High Level Output Voltage Low Level Output Voltage	I_{CC} V_{OH} V_{OL}	 2.4 	17 	40 0.4	mA V V	 I_{OH} = -40 µA max. I_{OL} = 3.2 mA
	Rise Time Fall Time	t_r t_f	 	200 50	 	ns ns	C_L = 25 pF R_L = 11 kΩ pull-up
HEDS-5540 HEDS-5640 HEDM-5500 HEDM-5600	Supply Current High Level Output Voltage Low Level Output Voltage	I_{CC} V_{OH} V_{OL}	30 2.4 	57 	85 0.4	mA V V	 I_{OH} = -200 µA max. I_{OL} = 3.86 mA
	Rise Time Fall Time	t_r t_f	 	180 40	 	ns ns	C_L = 25 pF R_L = 2.7 kΩ pull-up
HEDM-5500 HEDM-5600	Supply Current High Level Output Voltage Low Level Output Voltage	I_{CC} V_{OH} V_{OL}	30 2.4 	57 	85 0.4	mA V V	 I_{OH} = -40 µA max. I_{OL} = 3.86 mA
	Rise Time Fall Time	t_r t_f	 	180 40	 	ns ns	C_L = 25 pF R_L = 3.2 kΩ pull-up

*Typical values specified at V_{CC} = 5.0 V and 25°C.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Mechanical Characteristics

Parameter	Symbol	Dimension	Tolerance[1]	Units
Codewheel Fits These Standard Shaft Diameters		2 3 4 5 6 8	+0.000 -0.015	mm
		5/32 1/8 3/16 1/4	+0.0000 -0.0007	in
Moment of Inertia	J	0.6 (8.0 x 10^{-6})		g-cm^2 (oz-in-s^2)
Required Shaft Length[2]		14.0 (0.55)	± 0.5 (± 0.02)	mm (in.)
Bolt Circle[3]	2 screw mounting	19.05 (0.750)	± 0.13 (± 0.005)	mm (in.)
	3 screw mounting	20.90 (0.823)	± 0.13 (± 0.005)	mm (in.)
	external mounting ears	46.0 (1.811)	± 0.13 (± 0.005)	mm (in.)
Mounting Screw Size[4]	2 screw mounting	M 2.5 or (2-56)		mm (in.)
	3 screw mounting	M 1.6 or (0-80)		mm (in.)
	external mounting ears	M 2.5 or (2-56)		mm (in.)
Encoder Base Plate Thickness		0.33 (0.130)		mm (in.)
Hub Set Screw		(2-56)		(in.)

Notes:
1. These are tolerances required of the user.
2. The HEDS-55X5 and 56X5, HEDM-5505, 5605 provide an 8.9 mm (0.35 inch) diameter hole through the housing for longer motor shafts. See Ordering Information.
3. The HEDS-5540 and 5640 must be aligned using the aligning pins as specified in Figure 3, or using the alignment tool as shown in "Encoder Mounting and Assembly". See also "Mounting Considerations."
4. The recommended mounting screw torque for 2 screw and external ear mounting is 1.0 kg-cm (0.88 in-lbs). The recommended mounting screw torque for 3 screw mounting is 0.50 kg-cm (0.43 in-lbs).

Electrical Interface

To insure reliable encoding performance, the HEDS-5540 and 5640 three channel encoders require 2.7 kΩ (± 10%) pull-up resistors on output pins 2, 3, and 5 (Channels I, A, and B) as shown in Figure 1. These pull-up resistors should be located as close to the encoder as possible (within 4 feet). Each of the three encoder outputs can drive a single TTL load in this configuration.

The HEDS-5500, 5600, and HEDM-5500, 5600 two channel encoders do not normally require pull-up resistors. However, 3.2 kΩ pull-up resistors on output pins 3 and 5 (Channels A and B) are recommended to improve rise times, especially when operating above 100 kHz frequencies.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

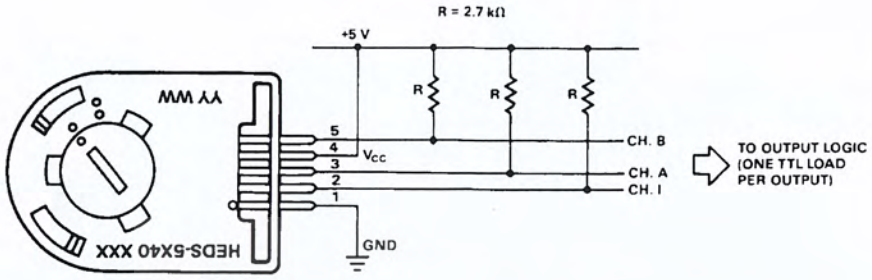


Figure 1. Pull-up Resistors on HEDS-5X40 Encoder Outputs.

Mounting Considerations

The HEDS-5540 and 5640 three channel encoders and the HEDM Series high resolution encoders must be aligned using the aligning pins as specified in Figure 3, or using the HEDS-8910 Alignment Tool as shown in Encoder Mounting and Assembly.

The use of aligning pins or alignment tool is recommended but not required to mount the HEDS-5500 and 5600. If these two channel encoders are attached to a motor with the screw sizes and mounting tolerances specified in the mechanical characteristics section without any additional mounting bosses, the encoder output errors will be within the maximums specified in the encoding characteristics section.

The HEDS-5500 and 5540 can be mounted to a motor using either the two screw or three screw mounting option as shown in Figure 2. The optional aligning pins shown in Figure 3 can be used with either mounting option.

The HEDS-5600, 5640, and HEDM-5600 have external mounting ears which may be used for mounting to larger motor base plates. Figure 4 shows the necessary mounting holes with optional aligning pins and motor boss.

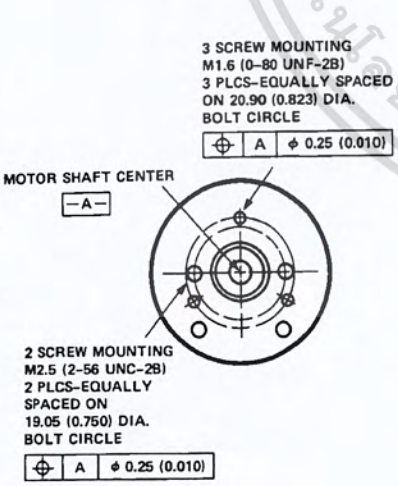


Figure 2. Mounting Holes.

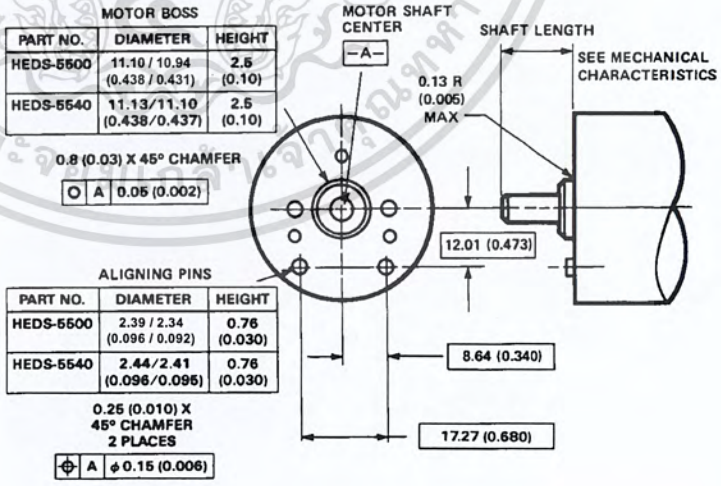


Figure 3. Optional Mounting Aids.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

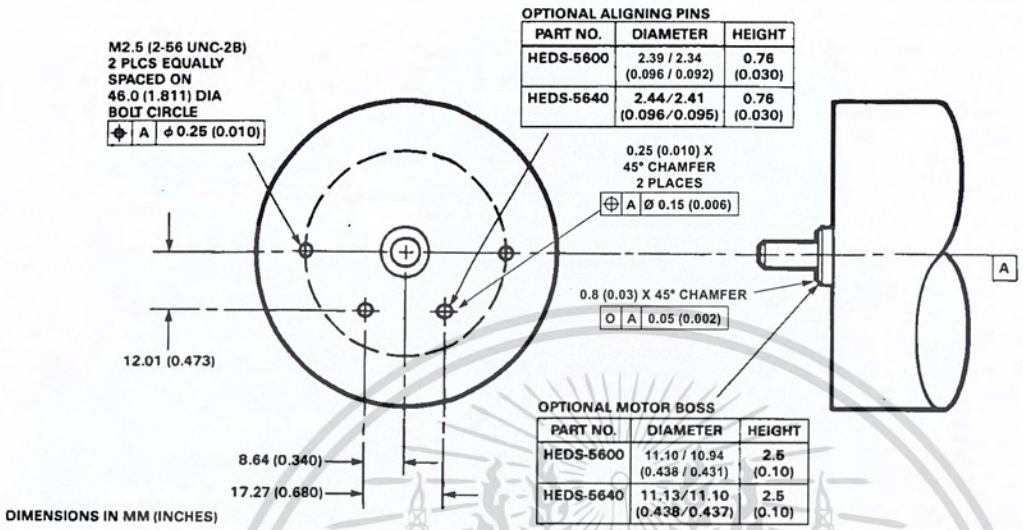
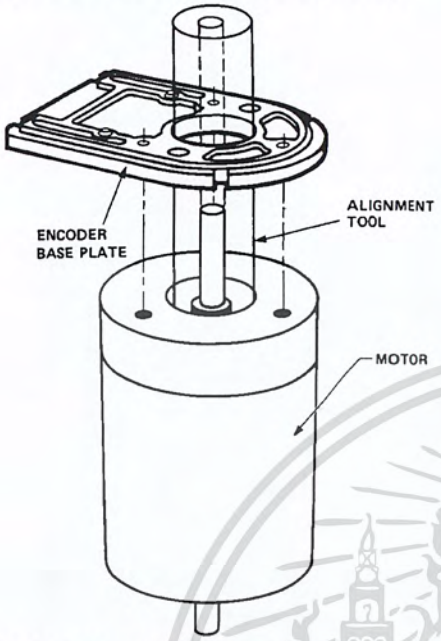
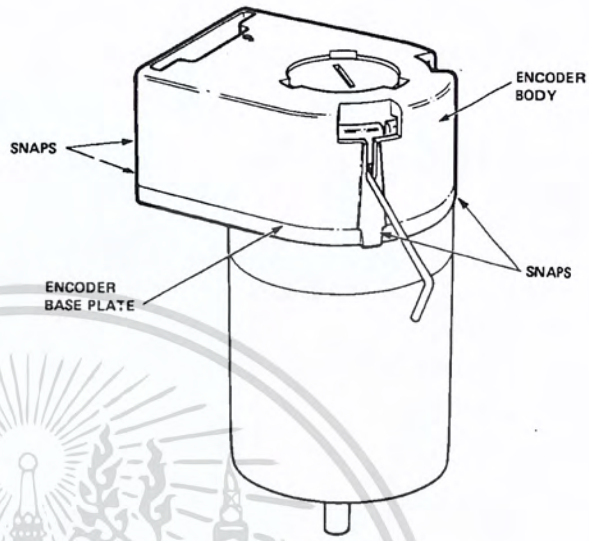


Figure 4. Mounting with External Ears.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Encoder Mounting and Assembly

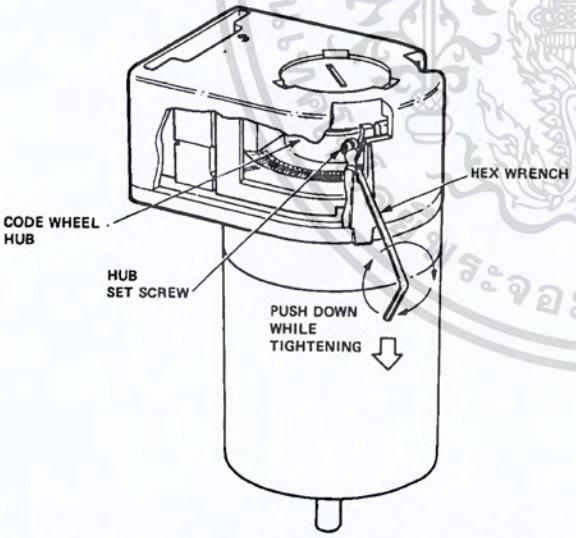


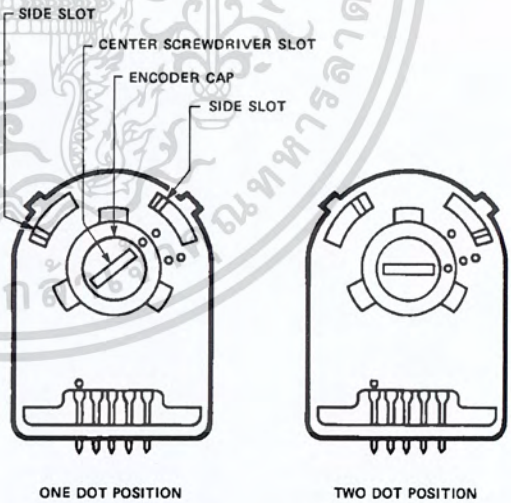


1. For HEDS-5500 and 5600: Mount encoder base plate onto motor. Tighten screws. Go on to step 2.

1a. For HEDS-5540, 5640 and HEDM-5500, 5600: Slip alignment tool onto motor shaft. With alignment tool in place, mount encoder baseplate onto motor as shown above. Tighten screws. Remove alignment tool.

2. Snap encoder body onto base plate locking all 4 snaps.





3a. Push the hex wrench into the body of the encoder to ensure that it is properly seated into the code wheel hub set screws. Then apply a downward force on the end of the hex wrench. This sets the code wheel gap by levering the code wheel hub to its upper position.

3b. While continuing to apply a downward force, rotate the hex wrench in the clockwise direction until the hub set screw is tight against the motor shaft. The hub set screw attaches the code wheel to the motor's shaft.

3c. Remove the hex wrench by pulling it straight out of the encoder body.

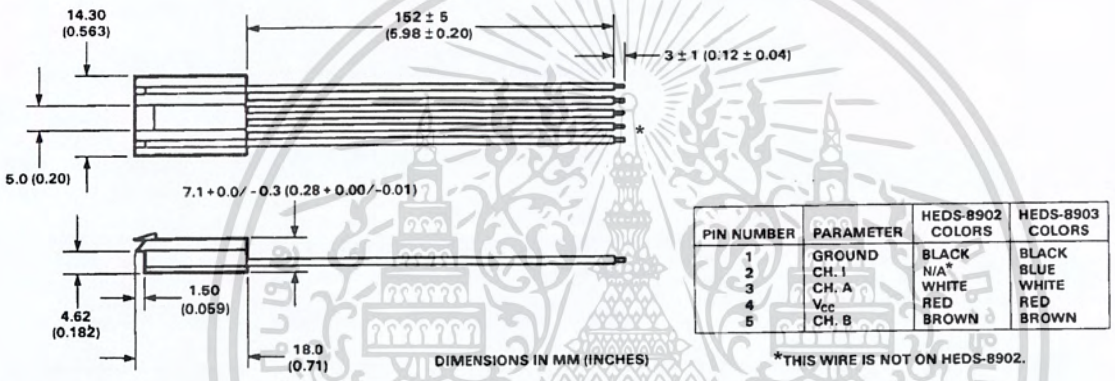
4. Use the center screwdriver slot, or either of the two side slots, to rotate the encoder cap dot clockwise from the one dot position to the two dot position. Do not rotate the encoder cap counterclockwise beyond the one dot position.

The encoder is ready for use!

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Connectors

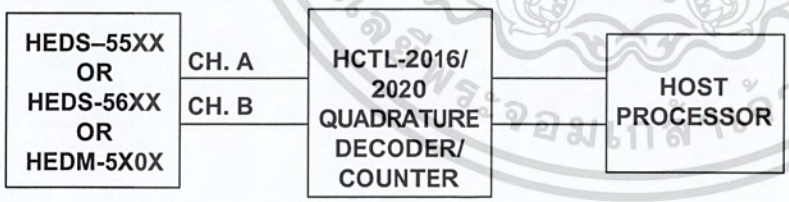
Manufacturer	Part Number
AMP	103686-4 640442-5
Dupont/Berg	65039-032 with 4825X-000 term.
Agilent (designed to mechanically lock into the HEDS-5XXX, HEDM-5X0X Series)	HEDS-8902 (2 ch.) with 4-wire leads
	HEDS-8903 (3 ch.) with 5-wire leads
Molex	2695 series with 2759 series term.

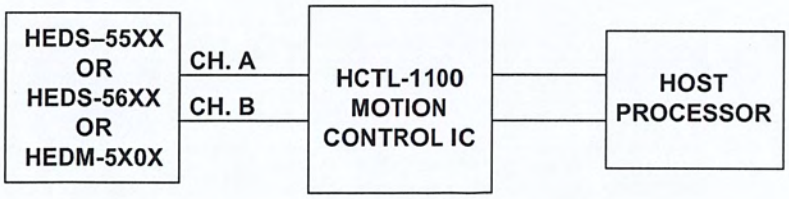


PIN NUMBER	PARAMETER	HEDS-8902 COLORS	HEDS-8903 COLORS
1	GROUND	BLACK	BLACK
2	CH. I	N/A*	BLUE
3	CH. A	WHITE	WHITE
4	V_{cc}	RED	RED
5	CH. B	BROWN	BROWN

Figure 5. HEDS-8902 and 8903 Connectors.

Typical Interfaces





เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Ordering Information

Encoders with Film Codewheels

HEDM-5 □ 0 □ Option □ □

Mounting Type	Outputs	Through Hole	Resolution (Cycles/Rev)	Shaft Diameter
5 - Standard 6 - External Mounting Ears	0 - 2 Channel	0 - None 5 - 8.9 mm (0.35 in.)	B - 1000 CPR J - 1024 CPR	01 - 2 mm 02 - 3 mm 03 - 1/8 in. 04 - 5/32 in. 05 - 3/16 in. 06 - 1/4 in. 11 - 4 mm 14 - 5 mm 12 - 6 mm 13 - 8 mm

HEDS-8910 0 □ □ Alignment Tool

(Included with each order of HEDM-550X/560X two channel encoders)

Encoders with Metal Codewheels

HEDS-5 □ □ □ Option □ □

Mounting Type	Outputs	Through Hole	Resolution (Cycles/Rev)	Shaft Diameter
5 - Standard 6 - External Mounting Ears	0 - 2 Channel 4 - 3 Channel	0 - None 5 - 8.9 mm (0.35 in.)	(HEDS-550X, 560X 2 Channel) S - 50 CPR K - 96 CPR C - 100 CPR D - 192 CPR E - 200 CPR F - 256 CPR G - 360 CPR H - 400 CPR A - 500 CPR I - 512 CPR (HEDS-554X, 564X 3 Channel) S - 50 CPR K - 96 CPR C - 100 CPR E - 200 CPR F - 256 CPR G - 360 CPR H - 400 CPR A - 500 CPR I - 512 CPR	01 - 2 mm 02 - 3 mm 03 - 1/8 in. 04 - 5/32 in. 05 - 3/16 in. 06 - 1/4 in. 11 - 4 mm 14 - 5 mm 12 - 6 mm 13 - 8 mm

HEDS-8910 0 □ □ Alignment Tool

(Included with each order of HEDS-554X/564X three channel encoders)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

		01	02	03	04	05	06	11	12	13	14
HEDM-5500	**B**	*	*				*	*	*	*	*
	J		*				*		*	*	*
HEDM-5505	**B**				*						
	J			*			*			*	
HEDM-5600	**B**						*			*	
	J						*				
HEDM-5605	**B**						*				
	J						*				
HEDS-5500	**A**	*	*	*	*	*	*	*	*	*	*
	C	*	*	*	*	*	*	*	*	*	*
	E		*		*	*	*	*	*		*
	F	*	*		*	*	*	*	*		*
	G		*			*	*		*		*
	H					*	*		*		*
	I	*	*	*	*	*	*	*	*	*	*
	K				*	*	*	*			*
	S									*	
HEDS-5505	**A**				*		*			*	*
	C				*		*		*		*
	E				*		*				*
	F				*		*				*
	G				*		*				*
	H						*				*
	I				*		*			*	
	K				*						
HEDS-5540	**A**	*	*	*	*	*	*	*	*	*	*
	C	*	*				*	*	*	*	*
	E						*	*			
	F	*					*	*			*
	G						*				
	H						*				*
	I	*	*				*	*	*	*	*
HEDS-5545	**A**								*		*
	C								*		
	H						*				*
	I						*				
HEDS-5600	**A**						*		*	*	*
	C						*		*		*
	E						*				
	G						*			*	
	H						*		*		
	I	*						*			

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

		01	02	03	40	05	06	11	12	13	14
HEDS-5605	**A** **C** **E** **F** **G** **H** **I**						* * * * * *			* * 	 *
HEDS-5640	**A** **E** **F** **H**						* * * *		*	*	
HEDS-5645	**A** **C** **E** **G** **H** **I**						* *		* *	* * * * * *	 *

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Agilent Technologies

www.agilent.com/semiconductors
For product information and a complete list of distributors, please go to our web site.

For technical assistance call:

Americas/Canada: +1 (800) 235-0312 or (408) 654-8675

Europe: +49 (0) 6441 92460

China: 10800 650 0017

Hong Kong: (+65) 271 2451

India, Australia, New Zealand: (+65) 271 2394

Japan: (+81 3) 3335-8152(Domestic/International), or 0120-61-1280(Domestic Only)

Korea: (+65) 271 2194

Malaysia, Singapore: (+65) 271 2054

Taiwan: (+65) 271 2654

Data subject to change.
Copyright © 2002 Agilent Technologies, Inc.
Obsoletes 5988-2579EN
January 17, 2002
5988-3996EN

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้