การพัฒนาระบบติดตามดาว
ASTRONAVIGATION TRACKING

จัดทำโดย

น.ส. วารุณี ฉันทวิบูลชัย
น.ส. วิมลพรรณ แก้วพรม

ปริญญานิพนธ์นี้ เป็นส่วนหนึ่งของหลักสูตรวิศวกรรมศาสตร์บัณฑิต
สาขาวิชา วิศวกรรมระบบควบคุม
สถาบันเทคโนโลยี พระจอมเกล้า เจ้าคุณทหาร ลาดกระบัง
ปีการศึกษา 2536

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์ ปีการศึกษา 2536

ภาควิชา วิศวกรรมระบบควบคุม

คณะ วิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้า เจ้าคุณทหาร ลาดกระบัง

เรื่อง การพัฒนาระบบติดตามดาว

ASTRONAVIGATION TRACKING

ผู้จัดทำ น.ส. วารุณี ฉันทวิบูลชัย 33100339

น.ส. วิมลพรรณ แก้วพรม 33100350

ดร. วันชัย ริ้วรุจา

อาจารย์ที่ปรึกษา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

033205

การพัฒนาระบบติดตามดาว

(Astronavigation Tracking)

โดย นางสาว วารุณี ฉันทวิบูลชัย
นางสาว วิมลพรรณ แก้วพรม

อาจารย์ที่ปรึกษา ดร. วันชัย ริ้วรุจา

บทคัดย่อ

โครงงานนี้นำเสนอโปรแกรมสนับสนุนการพัฒนาระบบติดตามดาว โปรแกรมเขียนด้วยภาษา C ประกอบด้วย ส่วนคำนวณตำแหน่งของวัตถุท้องฟ้า (ดาวเคราะห์,ดาวฤกษ์,ดวงจันทร์ และดวงอาทิตย์) ส่วนแสดงผลของการคำนวณตำแหน่งออกทางหน้าจอในรูปแบบต่างๆ และส่วนส่งผลที่ได้จากการคำนวณ ออกทางพอร์ตอนุกรมของคอมพิวเตอร์ เพื่อให้นำไปใช้ในการควบคุมการหมุนของกล้องดูดาวต่อไป

ABSTRACT

This project presents a program to support astronavigation tracking development . The program is written in C-language . There are three parts in this project , the calculation of the sky objects' (planet, stars, moon and sun) position , the display of the positon calculation on the monitor in various formats , and the output transfer from the calculation to the serial port of the computer to control the telescope movement later .

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

ขอขอบคุณ อาจารย์ ดร. วันชัย ริ้วรุจา และ อาจารย์ ปรเมษฐ์ ประนยานันทน์ ซึ่งกรุณาสนับสนุน และให้คำแนะนำตลอดมา

ขอขอบคุณ ท้องฟ้าจำลอง ที่กรุณาให้รายละเอียดข้อมูลต่าง ๆ

ขอบคุณ นางสาว อรพินท์ อัสรางชัย ที่ช่วยทำรายงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

หัวข้อเรื่อง		หน้า
บทคัดย่อ		I
Abstract		I
กิตติกรรมประกาศ		II
บทที่ 1	บทนำ	1
บทที่ 2	พื้นฐานทางดาราศาสตร์	2
บทที่ 3	ระบบพิกัด	3
บทที่ 4	ส่วนประกอบของโครงงาน	4
สรุปผลและวิจารณ์		58
หนังสืออ้างอิง		60
ภาคผนวก		61

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

ตารางที่	หน้า
1. ค่า RA ที่ได้จากโปรแกรมเทียบกับ Ephemeris	58
2. ค่า DEC ,RA ของดาวฤกษ์ทั้ง 88 กลุ่มดาว	61

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญภาพ

ภาพที่	หน้า
1. แสดงทรงกลมท้องฟ้า	2
2. Flow Chart แสดงการทำงานของรายการหลัก	5
3. Flow Chart แสดงการทำงานของส่วนคำนวณตำแหน่ง	32
4. Flow Chart แสดงการทำงานของส่วน Track	42
5. รายชื่อ ดาวเคราะห์ ดวงอาทิตย์ และดวงจันทร์	48
6. ผลการแสดงตำแหน่งของ Planet Fix	49
7. รายชื่อดาวฤกษ์	50
8. ผลการแสดงตำแหน่งของดาวฤกษ์	51
9. ปฏิทินดาราศาสตร์	52
10. การแสดงผลของคำสั่ง Track	53
11. การแสดงผลของคำสั่ง Find	53
12. แสดงการแก้ไขข้อมูล ละติจูด ลองจิจูด	54
13. แสดงการแก้ไขข้อมูลเวลา	54
14. แสดงการแก้ไขข้อมูลประเทศ	54
15. แสดง Timing Diagram ของวงจร	55
16. แสดงวงจรแปลง Digital เป็น Analog	57

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

ดาราศาสตร์เป็นสิ่งที่อยู่ใกล้ตัวมนุษย์มาก แต่มีเพียงส่วนน้อยที่รู้จักและเห็นความสำคัญของการศึกษา ทั้งๆ ที่การศึกษาดาราศาสตร์ให้ประโยชน์ในหลายๆ ด้าน ทั้งความรู้และสันทนาการ ดังที่คนโบราณได้เคยสงสัยมาก่อนว่า โลกแบนจริงหรือ ? โลกหมุนรอบดวงอาทิตย์ หรือดวงอาทิตย์หมุนรอบโลก ? เป็นคำตอบที่ได้มาแล้วจากดาราศาตร์ คำตอบเหล่านี้นำมาซึ่งการพัฒนาด้านเทคโนโลยีหลายๆ ด้าน เช่น ด้านอวกาศ ด้านการคมนาคม เป็นต้น แต่ยังมีคำถามอีกมากมายที่ยังไม่ได้รับคำตอบและยังจะต้องค้นคว้ากันต่อไป

ในประเทศไทย การศึกษาทางดาราศาสตร์ยังไม่กว้างขวางนัก เครื่องมือที่ใช้จึงยังไม่ค่อยมีประสิทธิภาพมากนัก เช่น กล้องดูดาวที่ใช้อยู่ทั่วๆ ไป ส่วนใหญ่เป็นแบบที่ต้องปรับด้วยมือซึ่งยุ่งยาก ยิ่งต้องการจะใช้กล้องที่มีกำลังขยายสูงๆ เพื่อศึกษาดาวดวงใดให้ละเอียด ปัญหาที่ประสบกันมาก คือต้องมีการปรับกล้องอยู่ตลอดเวลา เพราะดาวจะเคลื่อนออกจากเลนซ์ของกล้องที่ส่องอยู่อย่างรวดเร็วมาก กล้องที่สามารถติดตามดาวได้ในปัจจุบัน มักจะเป็นกล้องดูดาวที่ติดตั้งอยู่บนหอดูดาว ซึ่งมีขนาดใหญ่และไม่สะดวกในการใช้งานที่ต้องมีการเคลื่อนย้ายบ่อยๆ

ด้วยเหตุนี้ จึงเกิดความคิดที่จะพัฒนาระบบติดตามดาว โดยการประดิษฐ์กล้องที่สามารถหาตำแหน่งของดาวและติดตามดาวได้ เพื่อลดความยุ่งยากในการปรับกล้องลง นอกจากนี้ยังสามารถหาตำแหน่งของดาวได้ เพื่อให้ผู้ที่มีความรู้ด้านดาราศาสตร์ไม่มากนัก สามารถศึกษาด้านดาราศาสตร์ได้ เป็นการขยายวงการดาราศาสตร์ให้กว้างขวางยิ่งขึ้น

โครงงานนี้เป็นการพัฒนาระบบติดตามดาว โดยพัฒนาโปรแกรมที่ใช้ในการคำนวณตำแหน่งของวัตถุบนท้องฟ้า และส่งค่าตำแหน่งของดาวออกมาทางพอร์ตอนุกรม ส่วนประกอบของโปรแกรมมีดังนี้

1.1 รายการหลัก (menu) เป็นส่วนหลักของโปรแกรม จะแสดงเมนูให้ผู้ใช้เลือกว่าต้องการทำอะไร ซึ่งผู้ใช้จะสามารถเลือกแบบของการแสดงผล และกำหนดค่าตัวแปรในการคำนวณได้

1.2 ส่วนคำนวณตำแหน่ง ประกอบด้วย ฟังก์ชันคำนวณตำแหน่งของดาวเคราะห์ ดาวฤกษ์ ดวงจันทร์ และดวงอาทิตย์

1.3 ส่วนแสดงผลการทำงาน ประกอบด้วยฟังก์ชันที่แสดงผลการคำนวณตำแหน่งของดวงดาวในรูปแบบต่างๆ

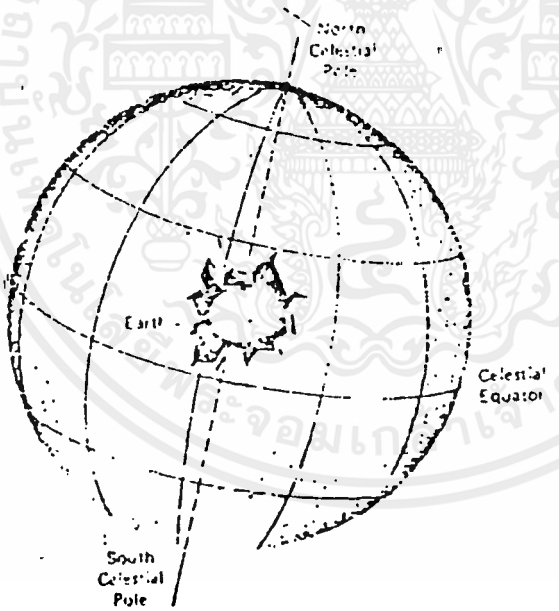
1.4 ส่วนส่งค่าที่ได้จากการคำนวณออกมาทางพอร์ตอนุกรม โดยค่าที่ส่งออกไปเป็นค่าอัลติจูดและอะซิมุท (โคออดิเนตในระบบพิกัดเส้นขอบฟ้า)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2
พื้นฐานทางดาราศาสตร์

ในทางดาราศาสตร์ เราสมมติว่าดาวและวัตถุท้องฟ้าอื่นๆ ติดอยู่บนทรงกลมกลวงใบหนึ่ง เรียกว่า ทรงกลมท้องฟ้า (CELESTRIAL SPHERE) วัตถุท้องฟ้าต่าง ๆ อยู่ห่างจากโลกมาก รัศมีของทรงกลมท้องฟ้าเป็นอนันต์ เมื่อเทียบกับโลกแล้ว โลกจะมีขนาดเป็นจุด เมื่อเราสังเกตอยู่บนโลก เราจะถือว่า โลกเป็นจุดศูนย์กลางของทรงกลมท้องฟ้า โดยมีตำแหน่งคงที่ (ยกเว้น ดวงอาทิตย์ ดวงจันทร์ และดาวเคราะห์ มีตำแหน่งไม่คงที่บนท้องฟ้า)

โลกของเราหมุนรอบตัวเองรอบแกนสมมติอันหนึ่ง เมื่อเราต่อแกนสมมติอันนี้ออกไปยังทรงกลมท้องฟ้า แกนที่ต่อออกจากขั้วโลกเหนือของโลก จะไปพบผิวทรงกลมท้องฟ้าที่ขั้วเหนือ-ท้องฟ้า (NORTH CELESTRIAL POLE ,NCP) และแกนที่ต่อออกจากขั้วโลกใต้จะพบผิวทรงกลมท้องฟ้าที่ขั้วใต้ท้องฟ้า (SOUTH CELESTRIAL POLE ,SCP) ปัจจุบันนี้ตำแหน่งของขั้วเหนือท้องฟ้าอยู่ห่างจากดาวที่มีชื่อว่า POLARIS (อยู่ในหมู่ดาว เหนือ) ประมาณ 1 องศา



รูปที่ 1 แสดงทรงกลมท้องฟ้า

ทรงกลมท้องฟ้าเป็นทรงกลมที่อยู่คงที่ ดังนั้นดาวฤกษ์ต่างๆ จะมีตำแหน่งคงที่อยู่บนทรงกลมท้องฟ้าใบนี้ แต่เนื่องจากโลกของเราหมุนรอบแกนสมมติอันหนึ่งจากทิศตะวันตกไปตะวันออก ใช้เวลา 24 ชั่วโมงต่อรอบโดยประมาณ (23 ชั่วโมง 56 นาที) เมื่อเราสังเกตอยู่บนโลกซึ่งมีขนาดใหญ่มากเมื่อเทียบกับตัวเรา เราจะรู้สึกเหมือนกับว่าโลกหยุดนิ่งแต่ทรงกลมท้องฟ้าหมุนรอบโลกรอบแกนสมมติอันเดิมจากทิศตะวันออกไปตะวันตก สวนทางกับการหมุนของโลกใช้เวลา 24 ชั่วโมงต่อรอบเท่ากัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3
ระบบพิกัด

ระบบพิกัดที่ใช้กันอยู่ในทางดาราศาสตร์มี 4 ระบบคือ

1.ระบบเส้นขอบฟ้า (Horizon system)

2.ระบบเส้นศูนย์สูตร (Equator system)

3.ระบบสุริยะวิถี (Ecliptic system)

4.ระบบแกแลคติค (Galactic system)

ระบบเส้นขอบฟ้า

ระบบพิกัดแบบนี้เป็นระบบที่ง่ายต่อการใช้งาน เพราะจะถือผู้สังเกตเป็นจุดศูนย์กลางมีเส้นขอบฟ้าเป็นวงกลมใหญ่ล้อมรอบตัวผู้สังเกตอยู่ ระบบนี้มีโคออดิเนต 2 ตัวคือ อัลติจูด และอะซิมุท

โดยที่อัลติจูดคือ มุมเงยที่วัดจากเส้นขอบฟ้าที่ 0 องศาขึ้นมาถึงจุดเหนือศีรษะที่ 90 องศา ส่วนอะซิมุทคือ มุมที่วัดไปตามเส้นขอบฟ้า 0 องศา ที่จุดทิศเหนือ , 90 องศา ที่จุดทิศตะวันออก , 180 องศา ที่จุดทิศใต้ และ 270 องศา ที่จุดทิศตะวันตก

ระบบเส้นศูนย์สูตร

ระบบนี้เป็นที่นิยมแพร่หลายในแผนที่ดาวฤกษ์ มีโคออดิเนต 2 ตัว คือ เดคลิเนชัน (declination,DEC) และไรท์-แอสเซนชั่น (right asccension,RA)

ระบบสุริยะวิถี

ระบบนี้ใช้ในการศึกษาถึงการเคลื่อนที่และตำแหน่งของดาวเคราะห์ ดวงจันทร์ เมื่อเทียบกับดวงอาทิตย์ มีโคออดิเนต 2 ตัวคือ ละติจูดท้องฟ้า (celestial latitude) และลองจิจูดท้องฟ้า (celestial longitude)

ระบบแกแลคติค

ระบบแกแลคติค เป็นระบบที่ใช้ศึกษาถึงแกแลคซิคต่างๆ มีโคออดิเนต 2 ตัว คือ ละติจูดแกแลคติค (Galactic latitude) และลองจิจูดแกแลคติค (Galatic longtitude)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

ส่วนประกอบของโครงงาน

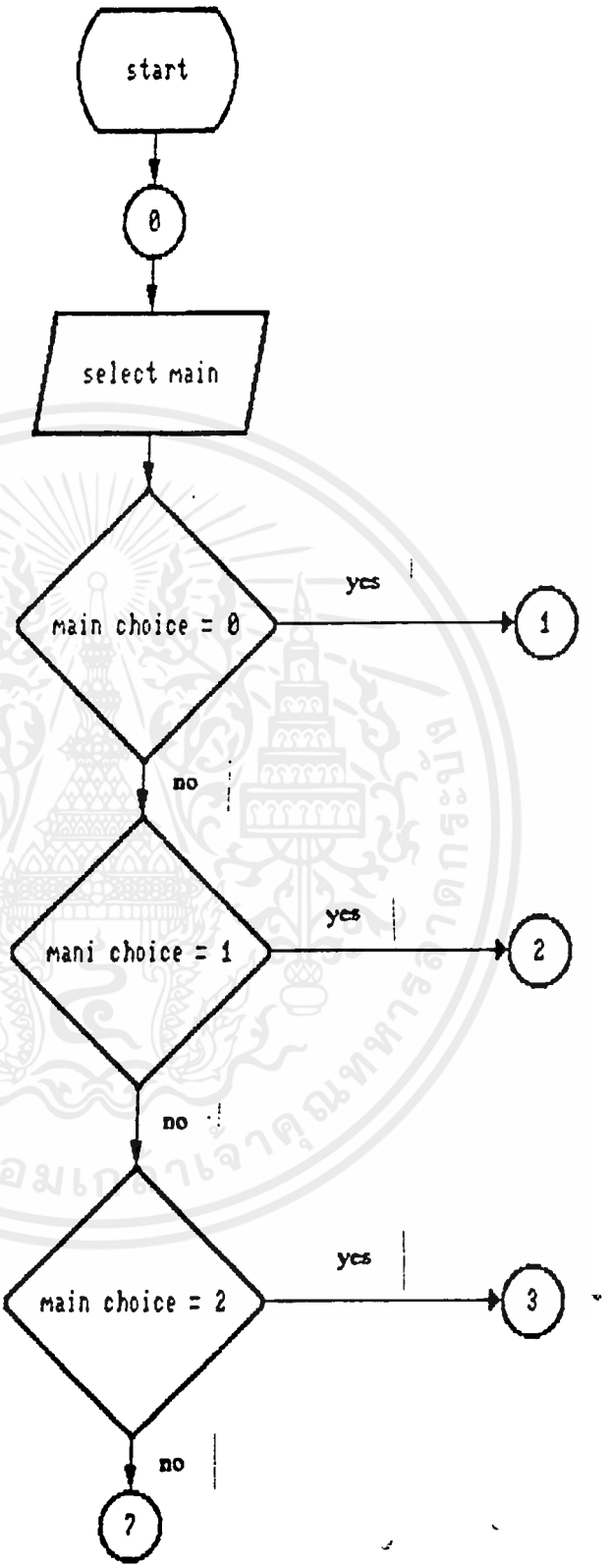
1. โปรแกรม ประกอบด้วย
   1..1 ส่วนของ Menu รายการให้เลือกการทำงาน
   1.2 ส่วนคำนวณตำแหน่งของดวงอาทิตย์ ดวงจันทร์ ดาวเคราะห์
   1.3 ส่วนแสดงผลการทำงาน
   1.4 ส่วนส่งค่าที่คำนวณได้ไปยัง Hardware
2. Hardware ประกอบด้วย
   2.1 ส่วนรับค่าและแปลงสัญญาณจาก Digital เป็น Analog

โปรแกรมระบบติดตามดาว

โปรแกรมระบบติดตามดาวนี้ ประกอบด้วย การคำนวณหาตำแหน่งของดาวและส่งข้อมูลตำแหน่งของดาวที่ต้องการออกมายัง Hardware ที่ควบคุมการทำงานของกล้องดูดาวต่อไป

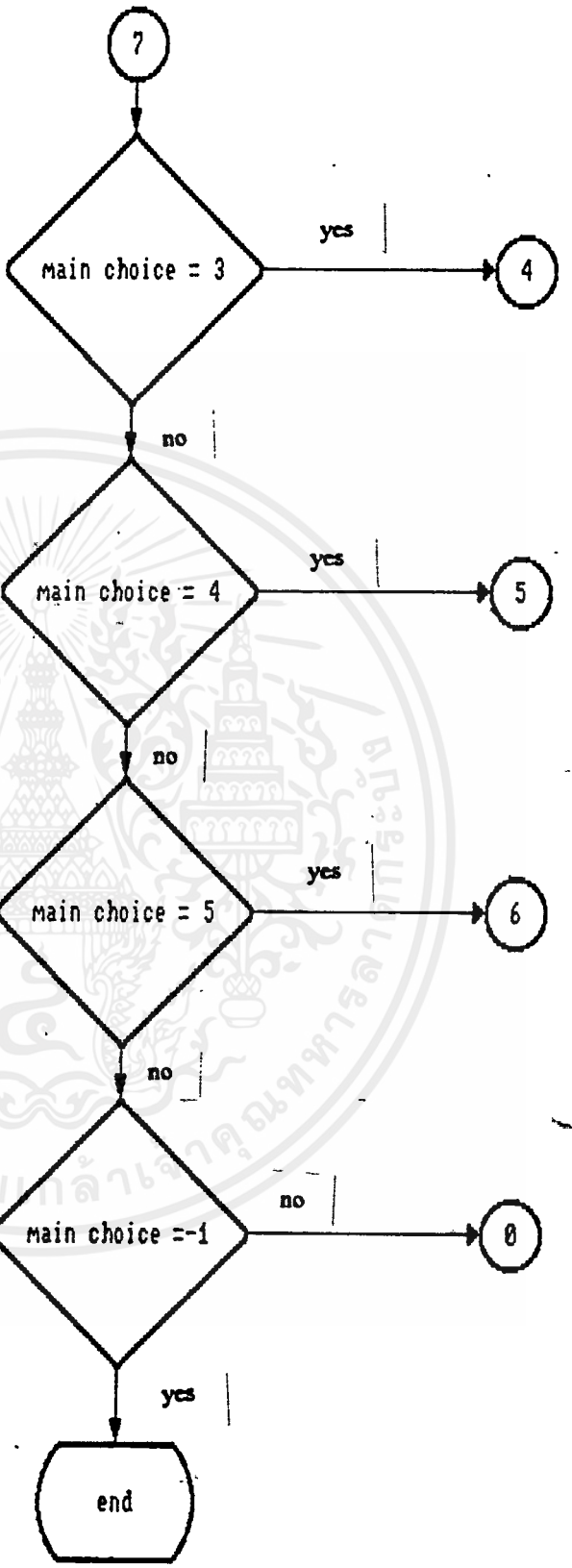
การทำงานของโปรแกรมส่วน Menu แสดงรายการหลัก แสดงได้ด้วย Flowchart ดังรูป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

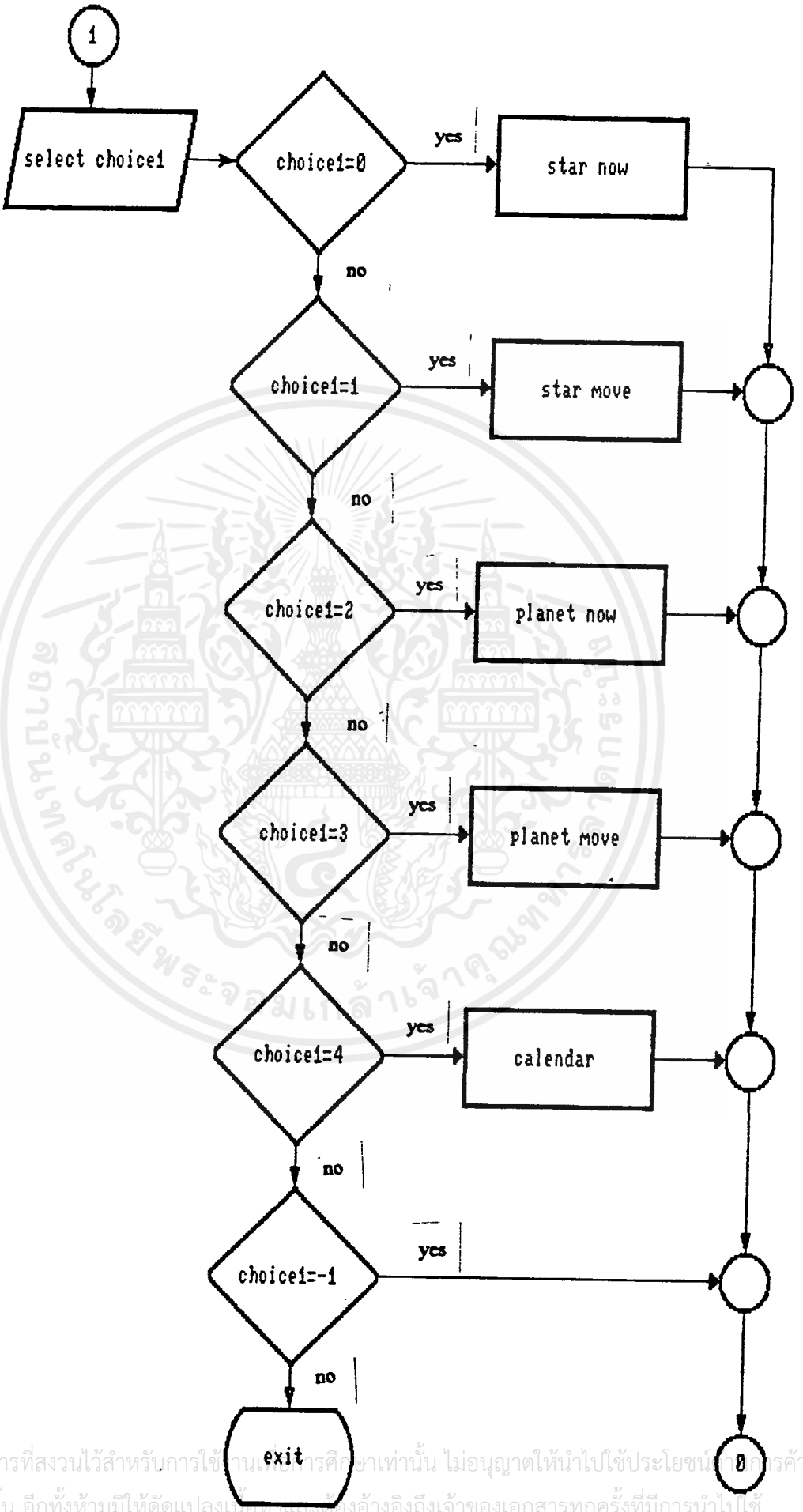


รูปที่ 2 Flow Chart แสดงรายการหลัก

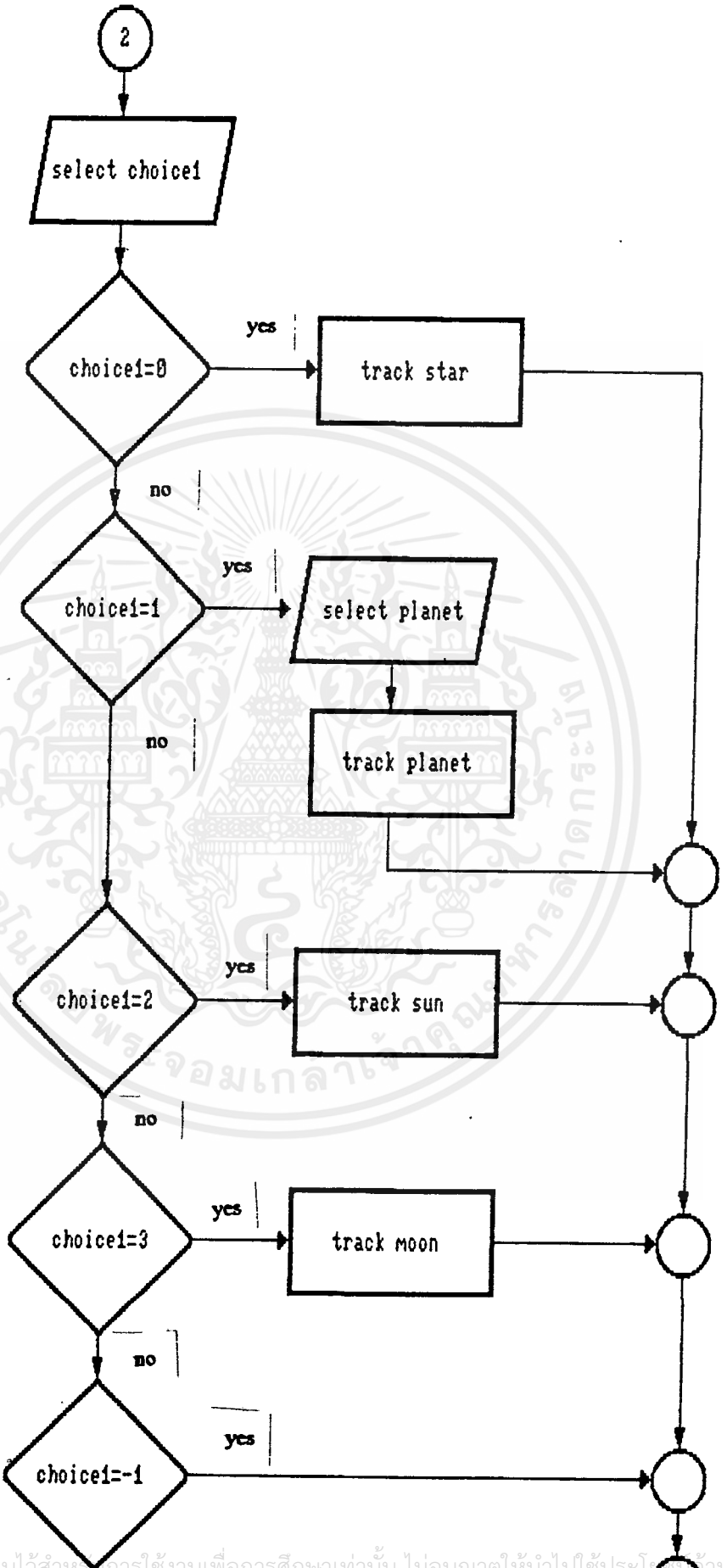
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



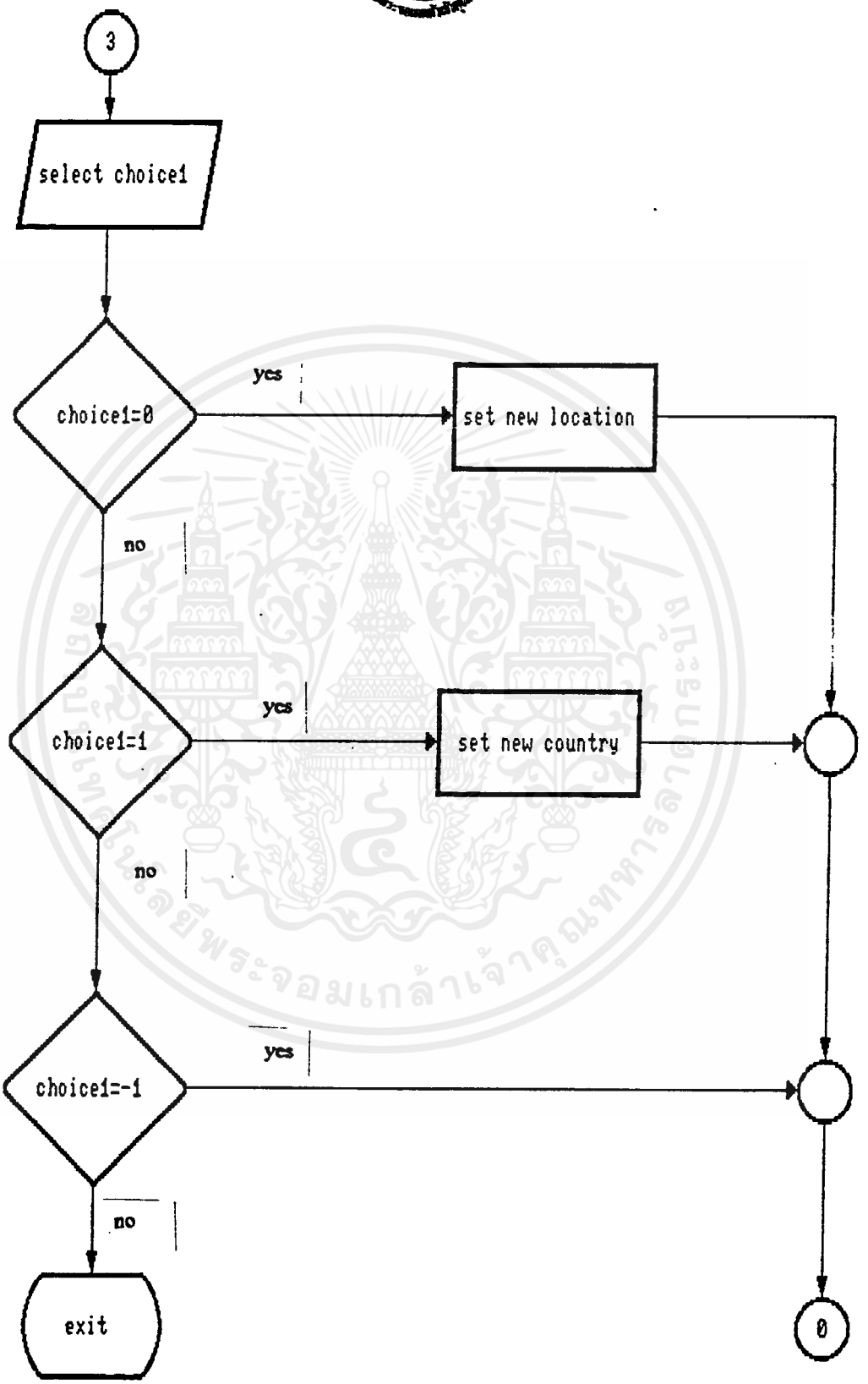
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

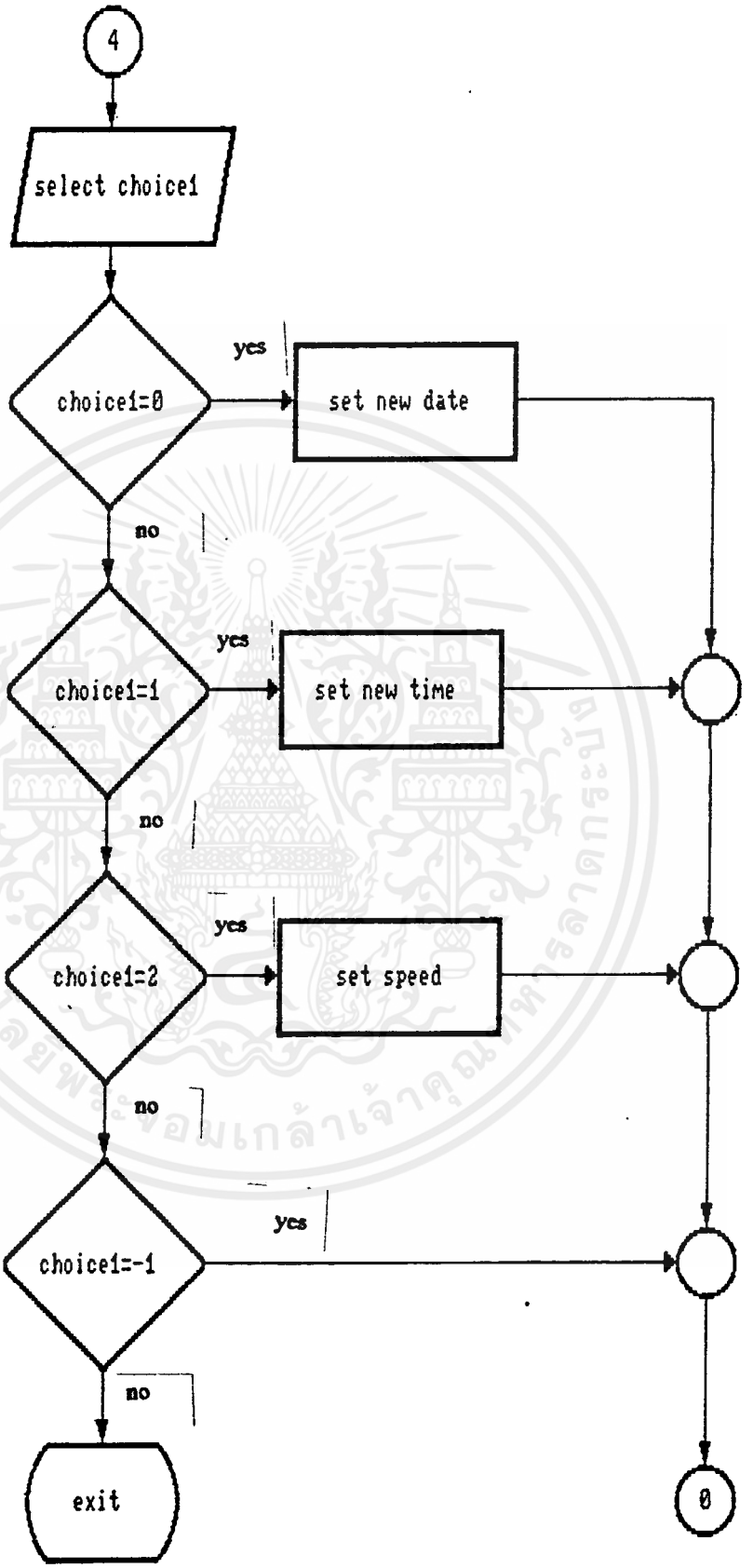


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

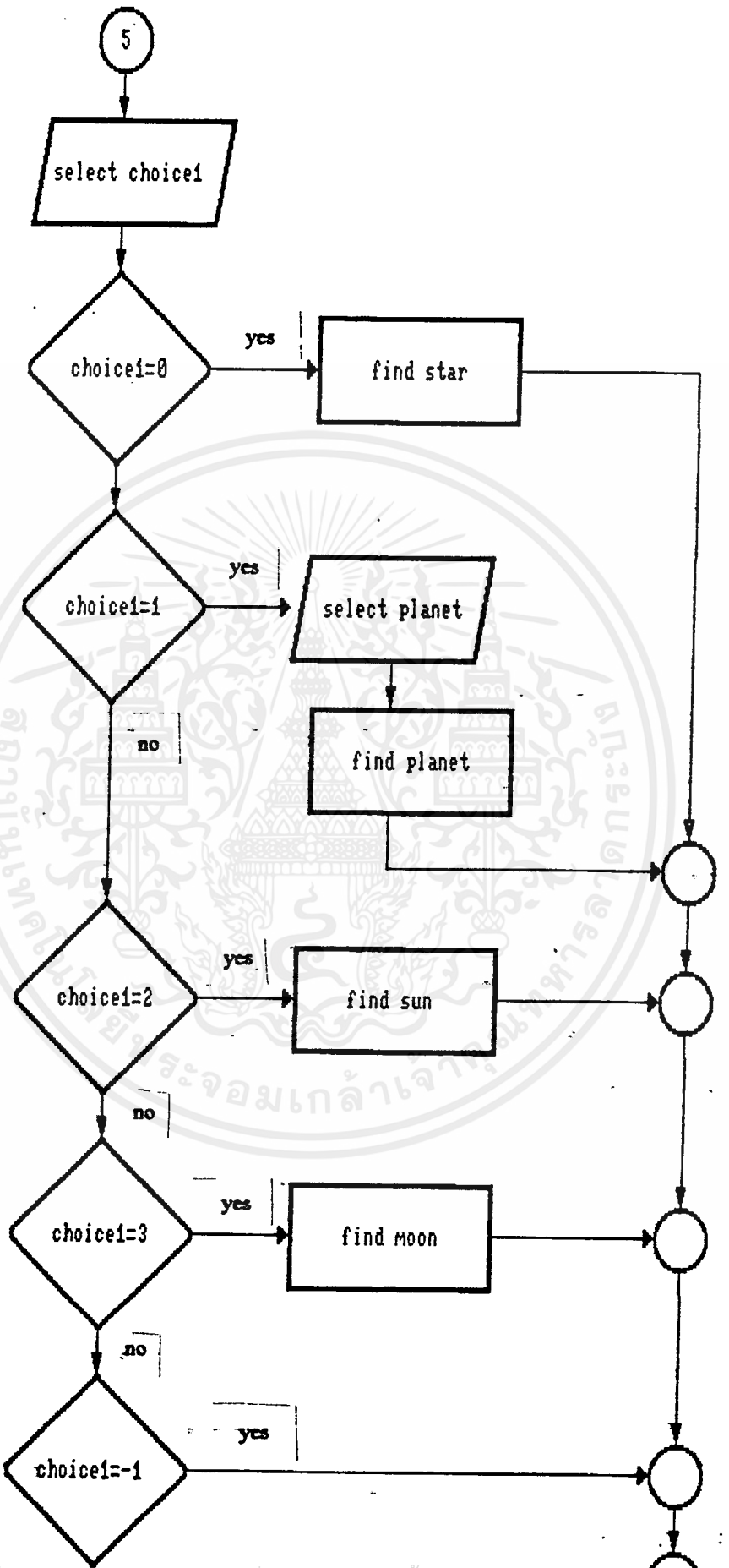


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

033205



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
#include<position.c>
#include<dos.h>
#include<setspeed.c>

coordinate place;
char  countrys[25];
long float speed;
int count1=0;

div_t domi ;
y_m_d day;
h_m_s times;

void setdata()
{
   struct time t ;
   struct date d ;

   gettime(&t) ;
   times.hour = t.ti_hour ;
   times.minute = t.ti_min ;
   times.second = t.ti_sec ;

   getdate(&d) ;
   day.year = d.da_year ;
   day.month = d.da_mon ;
   day.date = d.da_day ;

   place.lattitude = 14.00 ;
   place.longtitude = 100.50 ;
   speed=3600;
   strcpy(countrys,"Thailand") ;
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
#include<response.c>

#define BORDER 1
#define MAX_FRAME 10
#define MAX_MAIN 6
#define REV_VID 0x70
#define NORM_VID 7

void save_video() ;
void restore_video() ;
void pd_driver() ;
void make_main() ;
void make_menu() ;
void write_video() ;
void write_char() ;
void write_string() ;
void write_main() ;
void display_menu() ;
void draw_border() ;

int pulldown() ;
int select_main() ;

struct mainmenu
      {
       char *item ;
       int loc ;
      }main_menu[MAX_MAIN] ;
struct menu_frame
      {
       int startx,starty,endx,endy ;
       unsigned char *p ;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        char **menu,*keys ;
        int border,count ;
        int active ;
        void *ptr ;
      } frame[MAX_FRAME] ;
char *display[] =
    {
     "STAR FIX",
     "STAR MOVE",
     "PLANET FIX",
     "PLANET MOVE",
    "ASTRO CALENDAR",
    } ;
char *location[] =
    {
    "CORDINATE",
    "COUNTRY"
    } ;
char *abtime[] =
    {
     "DATE",
     "TIME",
     "SPEED"
    } ;
char *track[] =
    {
      "STAR",
      "PLANET",
      "SUN",
      "MOON"
    } ;
char *find[] =
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    {
      "STAR",
      "PLANET",
      "SUN",
      "MOON"
    } ;
char *planet[] =
    {
      "MERCURY",
      "VENUS",
      "MAR",
      "JUPITER",
      "SATURN",
      "URANUS",
     "NEPTUNE",
     "PLUTO"
    } ;
char *speeds[] =
    {
     "REAL TIME",
     "DEFINE"
    } ;



main()
{
   int i ;
   int temp ;


   setdata() ;
   initialgraphic() ;
   clearviewport() ;
```

เอกสารนี้มีไว้เพื่อสงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    gotoxy(0,0) ;
    make_menu(0,display,"fmpla",5,2,2,BORDER) ;
    make_menu(1,track,"tpsm",4,20,2,BORDER) ;
    make_menu(2,location,"oc",2,32,2,BORDER) ;
    make_menu(3,abtime,"dts",3,47,2,BORDER) ;
    make_menu(4,find,"tpsm",4,58,2,BORDER) ;
    make_menu(6,speeds,"rd",2,51,4,BORDER) ;
    make_menu(7,planet,"mvrjsunp",8,62,4,BORDER) ;
    make_menu(8,planet,"mvrjsunp",8,24,4,BORDER) ;
    make_main() ;

    pd_driver() ;
    closegraph() ;
    return 0 ;
}

void pd_driver()
{
   int main_choice,choice1,choice2 ;
   double jlday ;
   float gmttime,gsttime,lsttime ;
   coordinate position ;
   char *name ;

  while((main_choice=select_main())!=-1)
  {
     write_main(main_choice,REV_VID) ;
     switch(main_choice)
     {
case 0 : choice1 = pulldown(0) ;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  gsttime = gmttogst(day,gmttime,jlday) ;
  gsttime = correct(gsttime) ;
  lsttime = gsttolst(place,gsttime) ;
  lsttime = correct(lsttime) ;

  while((choice1=pulldown(1))!=-1)
  {
    switch(choice1)
    {
case 0 : starmenu(1) ;
 position = get_star() ;
 position = equatohori(position,lsttime,countrys) ;
 restore_video(1) ;
 tracking(position) ;
 break ;
case 1 : choice2 = pulldown(8) ;
                        restore_video(8) ;
 switch(choice2)
 {
   case 0 : position = planetpos('m',jlday) ;
    break ;
   case 1 : position = planetpos('v',jlday) ;
    break ;
   case 2 : position = planetpos('r',jlday) ;
    break ;
   case 3 : position = planetpos('j',jlday) ;
    break ;
   case 4 : position = planetpos('s',jlday) ;
    break ;
   case 5 : position = planetpos('u',jlday) ;
    break ;
   case 6 : position = planetpos('n',jlday) ;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        break ;
       case 7 : position = planetpos('p',jlday) ;
        break ;
       case -1 : break ;
       default : printf("PD DRIVER ERROR") ;
         exit(1) ;
   }
if (choice2==-1)
      break ;
   else
   {
      position = ecliptoequa(position,jlday) ;
      position = equatohori(position,lsttime,countrys) ;
      restore_video(1) ;
      tracking(position) ;
      break ;
   }
 case 2 : position = sunpos(jlday) ;
   position = ecliptoequa(position,jlday) ;
   position = equatohori(position,lsttime,countrys) ;
   restore_video(1) ;
   tracking(position) ;
   break ;
 case 3 : position = moonpos(jlday) ;
   position = ecliptoequa(position,jlday) ;
   position = equatohori(position,lsttime,countrys) ;
   restore_video(1) ;
   tracking(position) ;
   break ;
      }
   }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
 restore_video(1) ;
  setviewport(0,0,getmaxx(),getmaxy(),0) ;
  break ;
 case 2 : choice1 = pulldown(2) ;
  restore_video(2) ;
  switch(choice1)
  {
     case 0 : place = get_loc(place) ;
      break ;
     case 1 : get_country() ;
      break ;
     case -1 : break ;
     default : printf("PD DRIVER ERROR") ;
       exit(1) ;
  }
  setviewport(0,0,getmaxx(),getmaxy(),0) ;
  break ;
 case 3 : while((choice1=pulldown(3))!=-1)
     switch(choice1)
     {
case 0 : restore_video(3) ;
 day = get_date(day) ;
 break ;
case 1 : restore_video(3) ;
 times = get_time(times) ;
 break ;
case 2 : choice2 = pulldown(6) ;
 restore_video(6) ;
 switch(choice2)
 {
     case 0 : restore_video(3) ;
     speed = 1 ;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        break ;
       case 1 : restore_video(3) ;
        get_speed() ;
        break ;
       case -1 : break ;
       default : printf("PD DRIVER ERROR") ;
         exit(1) ;
    }
      }
    restore_video(3) ;
    setviewport(0,0,getmaxx(),getmaxy(),0) ;
    break ;
   case 4 : jlday = jdcal(day,times) ;

    gmttime = lttogmt(hmstodec(times),countrys,1) ;
    gmttime = correct(gmttime) ;

    gsttime = gmttogst(day,gmttime,jlday) ;
    gsttime = correct(gsttime) ;

    lsttime = gsttolst(place,gsttime) ;
    lsttime = correct(lsttime) ;

    while((choice1=pulldown(4))!=-1)
    {
      switch(choice1)
      {
  case 0 : starmenu(1) ;
   position = get_star() ;
   restore_video(4) ;
   findobj(position,"THE STAR",lsttime) ;
   break ;
  case 1 : choice2 = pulldown(5) ;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น และไม่อนุญาตให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        restore_video(5) ;
    switch(choice2)
     {
       case 0 : position = planetpos('m',jlday) ;
        *name = "MERCURY" ;
        break ;
       case 1 : position = planetpos('v',jlday) ;
        *name = "VENUS" ;
        break ;
       case 2 : position = planetpos('r',jlday) ;
        *name = "MARS" ;
        break ;
       case 3 : position = planetpos('j',jlday) ;
        *name = "JUPITER" ;
        break ;
       case 4 : position = planetpos('s',jlday) ;
        *name = "SATURN" ;
        break ;
       case 5 : position = planetpos('u',jlday) ;
        *name = "URANUS" ;
        break ;
       case 6 : position = planetpos('n',jlday) ;
        *name = "NEPTUNE" ;
        break ;
       case 7 : position = planetpos('p',jlday) ;
        *name = "PLUTO" ;
        break ;
       case -1 : break ;
       default : printf("PD DRIVER ERROR") ;
    }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
int pulldown(num)
int num ;
{
   if(!frame[num].active)
   {
      save_video(num) ;
      frame[num].active = 1 ;
   }
   if(frame[num].border)
     draw_border(num) ;
   display_menu(num) ;
   return get_resp(num) ;
}
void make_main()
{
   int i ;
   int gap ;

   main_menu[0].item = "DISPLAY" ;
   main_menu[1].item = "TRACK" ;
   main_menu[2].item = "LOCATION" ;
   main_menu[3].item = "TIME" ;
   main_menu[4].item = "FIND" ;
   main_menu[5].item = "QUIT" ;
   gap = 0 ;
   for(i=0;i<MAX_MAIN;i++)
     gap += strlen(main_menu[i].item) ;
   gap = gap*8 ;
   if(gap>getmaxx())
   {
     printf("Too many main menu.\n") ;
     printf("Press any key too halt.");
     getch() ;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    exit(1) ;
    }
    else
      gap = (int)((getmaxx()-gap)/(MAX_MAIN+1)) ;
    main_menu[0].loc = gap ;
    for(i=1;i<MAX_MAIN;i++)
      main_menu[i].loc = main_menu[i-1].loc+
  strlen(main_menu[i-1].item)*8+gap ;
}

void make_menu(num,menu,keys,count,x,y,border)
int num ;
char *menu[] ;
char *keys ;
int ocunt ;
int x,y ;
int border ;
{
  register int i,len ;
  int endx,endy,choice,vmode ;
  unsigned char *p ;

  if(num>MAX_FRAME)
  {
    printf("too many menu\n") ;
    return 0 ;
  }

  if((y>24)||(y<0)||(x>79)||(x<0))
  {
    printf("range menu error \n") ;
    return 0 ;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    }
  len = 0 ;
   for(i=0;i<count;i++)
      if(len<strlen(menu[i]))
  len = strlen(menu[i]) ;
   endx = x+len+3 ;
   endy = count+1+y ;
   if((endy>25)||(endx>80))
   {
      printf("menu won't fit\n") ;
      return 0 ;
   }
   p = (unsigned char *)malloc(2*(endx-x+1)*(endy-y+1)) ;
   if(!p)
      exit(1) ;
   frame[num].startx = x*8-10 ;
   frame[num].starty = y*14-10 ;
   frame[num].endx = endx*8+10 ;
   frame[num].endy = endy*14+10 ;
   frame[num].p = p ;
   frame[num].menu = (char**)menu ;
   frame[num].border = border ;
   frame[num].keys = keys ;
   frame[num].count = count ;
   frame[num].active = 0 ;
   return 1 ;
}


void display_menu(num)
int num ;
{
   register int y,i ;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น ทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    char **m ;

  y = frame[num].starty+20 ;
   m = frame[num].menu ;
   for(j=0;j<frame[num].count;j++,y+=14)
     write_string(frame[num].startx+18,y,m[j],NORM_VID,num) ;
}

void draw_border(num)
int num ;
{
   draw_rectan(frame[num].startx,frame[num].starty,
       frame[num].endx,frame[num].endy,
       SOLID_FILL,LIGHTGRAY) ;
   setcolor(0) ;
   rectangle(frame[num].startx+8,frame[num].starty+8,
     frame[num].endx-8,frame[num].endy-8) ;
}

int select_main()
{
   union inkey
 {
    char ch[2] ;
    int i ;
 } c ;
   int i ;
   int arrow_choice = 0 ;

   setcolor(BLACK) ;
   draw_rectan(0,0,getmaxx(),14,SOLID_FILL,LIGHTGRAY) ;
```

เอกสารนี้ ... ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    for(i=0;i<MAX_MAIN;i++)
      outtextxy(main_menu[i].loc,4,main_menu[i].item) ;
  write_main(arrow_choice,REV_VID) ;
    for(;;)
    {
      c.i = bioskey(0) ;
      write_main(arrow_choice,NORM_VID) ;
      if(c.ch[0])
      {
 switch(c.ch[0])
 {
    case 'd' : return 0 ;
    case 'k ': return 1 ;
    case 'l' : return 2 ;
    case 't' : return 3 ;
    case 'f' : return 4 ;
    case 'q' : return 5 ;
    case '\r' : return arrow_choice ;
    case 27  : exit(0) ;
}
     }
     else
     {
switch(c.ch[1])
{
   case 75 : arrow_choice-- ;
     break ;
   case 77 : arrow_choice++ ;
     break ;
   default : break ;
}
     }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        if(arrow_choice==MAX_MAIN)
    arrow_choice = 0 ;
        if(arrow_choice<0)
    arrow_choice = MAX_MAIN-1 ;
        write_main(arrow_choice,REV_VID) ;
     }
}

get_resp(num)
int num ;
{
    int i ;
    union inkey
  {
     char ch[2] ;
     int i ;
  } c ;
    int arrow_choice = 0,key_choice ;
    int x,y ;

    x = frame[num].startx+8 ;
    y = frame[num].starty+10 ;

    write_string(x+10,y+10,frame[num].menu[0],REV_VID,num) ;

    for(;;)
    {
      c.i = bioskey(0) ;
      write_string(x+10,y+arrow_choice*14+10,
    frame[num].menu[arrow_choice],NORM_VID,num) ;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    if(c.ch[0])
        {
    key_choice = is_in(frame[num].keys,tolower(c.ch[0])) ;
    if(key_choice)
       return key_choice-1 ;
    switch(c.ch[0])
    {

       case '\r' : return arrow_choice ;
       case ' '  : arrow_choice++ ;
    break ;
       case 27  : return -1 ;
    }
       }
        else
        {
    switch(c.ch[1])
    {
       case 72 : arrow_choice-- ;
        break ;
       case 80 : arrow_choice++ ;
        break ;
    }
        }
        if(arrow_choice==frame[num].count)
    arrow_choice = 0 ;
        if(arrow_choice<0)
    arrow_choice = frame[num].count-1 ;
        write_string(x+10,y+arrow_choice*14+10,
     frame[num].menu[arrow_choice],REV_VID,num) ;
    }
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

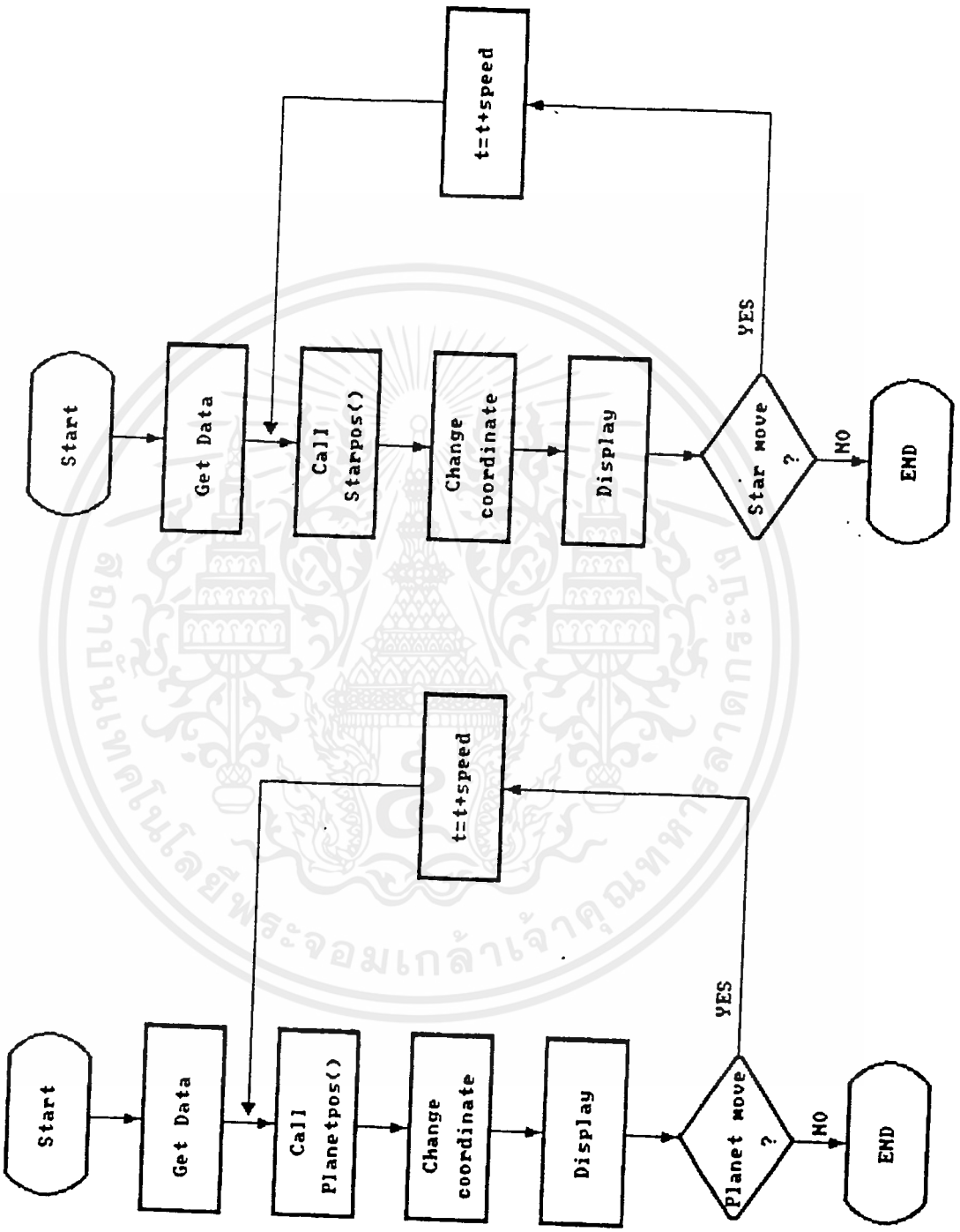
```
void write_main(num,attrib)
int num ;
int attrib ;
{
   if(attrib==NORM_VID)
      draw_rectan(main_menu[num].loc,0,
   main_menu[num].loc+strlen(main_menu[num])*8,14,
   SOLID_FILL,LIGHTGRAY) ;
   else if(attrib==REV_VID)
   draw_rectan(main_menu[num].loc,0,
       main_menu[num].loc+strlen(main_menu[num])*8,14,
       SOLID_FILL,GREEN) ;
   setcolor(BLACK) ;
   outtextxy(main_menu[num].loc,4,main_menu[num].item) ;
}

void write_string(x,y,p,attrib,num)
int x,y ;
int num ;
char *p ;
int attrib ;
{  if(attrib==NORM_VID)
      draw_rectan(frame[num].startx+9,y-3,
  frame[num].endx-9,y+10,
  SOLID_FILL,LIGHTGRAY) ;
   else if(attrib==REV_VID)
   draw_rectan(frame[num].startx+9,y-3,
       frame[num].endx-9,y+10,
       SOLID_FILL,GREEN) ;
   setcolor(BLACK) ;
   outtextxy(x,y,p) ;
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
void restore_video(num)
int num ;
{
   putimage(frame[num].startx,frame[num].starty,
    frame[num].ptr,COPY_PUT) ;
   free(frame[num].ptr) ;
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทำงานของโปรแกรมส่วนคำนวนตำแหน่งของดาว แสดงได้ด้วย Flowchart ดังรูป



รูปที่ 3 แสดงการคำนวณตำแหน่งของดวงอาทิตย์ ดวงจันทร์ ดาวเคราะห์ ดาวฤกษ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
#include<stdio.h>
#include<ctype.h>
#include<alloc.h>

typedef struct star *STARLINK ;
typedef struct star
{
   char starname[50] ;
   float dec,ra ;
   STARLINK left ;
   STARLINK right ;
} star ;

FILE *fp,*in;

STARLINK getstarnode()
{
  return((STARLINK) malloc(sizeof(star))) ;
}

STARLINK searchstar(ptr,key)
STARLINK ptr ;
char key[50] ;
{
  int test ;

  if(ptr!=NULL)
  {
test = strcmp(ptr->starname,key) ;
    if(test==0)
  return(ptr) ;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        if(test<0)
        {
       ptr=ptr->right ;
       searchstar(ptr,key) ;
        }
       else
        {
       ptr=ptr->left ;
       searchstar(ptr,key) ;
        }
     }
     else
       return(NULL);
 }

 void freenode(ptr)
 STARLINK *ptr ;
 {
    if ((*ptr)->left != NULL)
      freenode(&((*ptr)->left));
    else if ((*ptr)->right !=NULL)
    freenode(&((*ptr)->right));
 else
    free(*ptr);
 }

int treeinsertstar(root,newitem)
STARLINK *root ;
STARLINK newitem ;
{
  int test ;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    if(*root==NULL)
    {
       *root=newitem ;
       (*root)->left = NULL ;
       (*root)->right = NULL ;
    }
    else
    {
 test=strcmp(newitem->starname,(*root)->starname) ;
      if(test<0)
 treeinsertstar(&((*root)->left),newitem) ;
      else
 if(test>0)
    treeinsertstar(&((*root)->right),newitem) ;
 else
 {
    fprintf(stderr,"\nduplicate key error") ;
    exit(0) ;
 }
   }
}

void loadstar(root)
STARLINK *root ;
{
   star temp ;
   int count = 0 ;
   STARLINK newnode ;

   if((fp=fopen("star.dat","rb"))==NULL)
   {
      printf("can't open file") ;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        return ;
    }
    *root=NULL ;
    do
    {
        if(fread(&temp,sizeof(star),1,fp)==1)
       {
  if((newnode=getstarnode())==NULL)
  {
    fprintf(stderr,"\ngetstarnode error") ;
    exit(0) ;
  }
  count++ ;
  strcpy(newnode->starname,temp.starname) ;
  newnode->dec  = temp.dec ;
  newnode->ra = temp.ra ;
  treeinsertstar(root,newnode) ;
      }
   }while(!feof(fp)) ;
   freenode(&newnode);
   fclose(fp);
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
#include <coord.c>
#include <starlink.c>

#define Eg 278.833540 /* ecliptic longtitude at epoch */
#define wg 282.596403 /* ecliptic longtitude at perigee */
#define e 0.016718    /* eccentricity */
#define TE 1.00004
#define EE 98.833540
#define wE 102.596403
#define ACCURACY 0.000001
#define  L0    64.975464
#define  I     5.145396
#define  P0    349.383063
#define  N0    151.950429
#define  E     0.054900
#define  ANG0  0.5181
#define  A     384401.00
#define  PI0   0.9507


double keplerequa(meanano)
double meanano ;
{
   double diff,Ec ;
   Ec = meanano ;
    diff = Ec-e*sin(Ec)-meanano ;
  while(diff>ACCURACY)
  {
Ec = Ec-diff/(1-e*cos(Ec)) ;
diff = Ec-e*sin(Ec)-meanano ;
  }
  return(Ec) ;
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
coordinate sunpos(jd)
double jd ;
{
   coordinate suneclip ;
   double meanano,k,N ;
   N = (jd-2444238.5)*360/365.2422 ;
   N = testang(N) ;
   meanano = N+Eg-wg ;
   if(meanano<0)
      meanano = meanano+360 ;
   meanano = meanano*M_PI/180 ;
   suneclip.lattitude = 0 ;
   suneclip.longtitude = 2*atan(pow(((1+e)/(1-e)),0.5)
  *tan(keplerequa(meanano)/2))*180/M_PI+wg ;
   suneclip.longtitude = testang(suneclip.longtitude) ;
   return(suneclip) ;
}

coordinate moonpos(jd)
double jd;
{
   coordinate suneclip,mooneclip;
   double epoch,meanl,meanags,corano,corlong,orblong,n,phase;

   epoch=(jd-2444238.5);
   meanags=testang(epoch*360/365.2422)+Eg-wg;
   if (meanags<0)
  meanags=meanags+360;
   suneclip=sunpos(jd);
   meanl=13.1763966*epoch+L0;
   meanl=testang(meanl);
```

เอกสารนี้จัดทำขึ้นสำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
corano=sin((2*(meanl-suneclip.longtitude)-testang(meanl-0.1114041*epoch
-P0))*M_PI/180)*1.2739+testang(meanl-0.1114041*epoch-P0)-0.1858
*sin(meanags*M_PI/180)-0.37*sin(meanags*M_PI/180);

  corlong=sin((2*(meanl-suneclip.longtitude)-testang(meanl-0.1114041*epoch
-P0))*M_PI/180)*1.2739+meanl+6.2886*sin(corano*M_PI/180)-0.1858
*sin(meanags*M_PI/180)+0.214*sin(2*corano*M_PI/180);

  orblong=corlong+0.6583*sin((2*(corlong-suneclip.longtitude))*M_PI/180);
  n=testang(N0-0.0529539*epoch)-0.16*sin(meanags*M_PI/180);
  mooneclip.longtitude=sparctan(cos((orblong-n)*M_PI/180),sin((orblong-n)
    *M_PI/180)*cos(I*M_PI/180))+n;

  mooneclip.longtitude=testang(mooneclip.longtitude);
  mooneclip.lattitude=asin(sin((orblong-n)*M_PI/180)*sin(I*M_PI/180))*180/M_PI;
  return(mooneclip);
}

coordinate planetpos(planet,jd)
char planet ;
double jd ;

{
  double Tp,LE,w,e1,a,i,o,ang0,Np,NE,Mp,ME,hl,HL,vp,vE,r,R,phy,d[9];
  coordinate plt ;
  FILE *datafile ;
  int j ;

  planet = tolower(planet) ;
  switch (planet)
  {
    case 'm' : datafile = fopen("mercury.dat","r"); break ;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    case 'v' : datafile = fopen("venus.dat","r") ;    break ;
    case 'e' : datafile = fopen("earth.dat","r") ;    break ;
    case 'r' : datafile = fopen("mars.dat","r") ;     break ;
    case 'j' : datafile = fopen("jupiter.dat","r") ;  break ;
    case 's' : datafile = fopen("saturn.dat","r") ;   break ;
    case 'u' : datafile = fopen("uranus.dat","r") ;   break ;
    case 'n' : datafile = fopen("neptune.dat","r");   break ;
    case 'p' : datafile = fopen("pluto.dat","r") ;    break ;
    default  : printf("\nIT IS NOT A PLANET") ;      exit(1) ;
  }

  if(datafile == NULL )
  {
printf("OPEN FILE ERROR") ;
getch() ;
exit(1) ;
  }
  fscanf(datafile,"%lf %lf %lf %lf %lf %lf %lf %lf"
  ,&Tp,&LE,&w,&e1,&a,&i,&o,&ang0);
  fclose(datafile) ;
  Np = testang(360*(jd-2444238.5)/(365.2422*Tp)) ;
  Mp = Np+LE-w ;
  hl = testang(Np+((360/M_PI)*e1*sin(Mp*M_PI/180))+LE) ;
  vp = hl-w ;
  r = a*(1-pow(e1,2))/(1+e1*cos(vp*M_PI/180)) ;
  NE = 360*(jd-2444238.5)/(365.2422*TE) ;
  ME = NE+EE-wE ;
  HL = testang(NE+360*e*sin(ME*M_PI/180)/M_PI+EE) ;
  vE = HL-wE ;
  R = (1-pow(e,2))/(1+e*cos(vE*M_PI/180)) ;
  phy = asin(sin((hl-o)*M_PI/180)*sin(i*M_PI/180)) ;
  hl = sparctan(cos((hl-o)*M_PI/180),sin((hl-o)*M_PI/180)*cos(i*M_PI/180))+o ;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    r = r*cos(phy) ;
     if (planet=='m'||planet=='v')
    plt.longtitude = testang( 180+HL+atan((r*sin((HL-hl)*M_PI/180))
     /(R-r*cos((HL-hl)*M_PI/180)))*180/M_PI ) ;
     else
    plt.longtitude = testang(hl+atan((R*sin((hl-HL)*M_PI/180))
     /(r-R*cos((hl-HL)*M_PI/180)))*180/M_PI ) ;

     plt.lattitude = atan((r*tan(phy)*sin((plt.longtitude-hl)*M_PI/180))
     /(R*sin((hl-HL)*M_PI/180)))*180/M_PI ;

     return(plt) ;

}

coordinate starpos(key)
char key[50] ;
{
   char flag ;
   coordinate equa ;
   STARLINK found,root ;
   int choice,root1 ;

   loadstar(&root) ;
   found = searchstar(root,key) ;
   equa.lattitude =  found->ra  ;
   equa.longtitude = found->dec*15  ;
   return(equa) ;
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทำงานของโปรแกรมส่วน Tracking แสดงได้ด้วย Flowchart ดังรูป

รูปที่ 4 แสดงFlow Chart การทำงานของการ Track

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
void tracking(obj)
int obj ;
{
  unsigned int size ;
  void *ptr ;
  coordinate position ;
  double jlday ;
  float gmttime,gsttime,lsttime ;

  size = imagesize(200,100,getmaxx()-200,200) ;
  ptr = malloc(size) ;
  getimage(200,100,getmaxx()-200,200,ptr) ;
  setviewport(200,100,getmaxx()-200,200,0) ;
  clearviewport() ;

  draw_rectan(0,0,getmaxx()-400,100,SOLID_FILL,BLUE) ;
  setcolor(LIGHTGRAY) ;
  rectangle(8,8,getmaxx()-408,92) ;

  setcolor(BLUE) ;
  draw_rectan(getmaxx()/2-264,8,getmaxx()/2-136,8) ;

  setcolor(YELLOW) ;
  outtextxy(getmaxx()/2-264,5,"OUT DATA TO PORT") ;
  outtextxy(38,32,"send altitude :") ;
  outtextxy(38,63,"send azimuth  :") ;

  jlday = jdcal(day,times) ;

  gmttime = lttogmt(hmstodec(times),countrys,1) ;
  gmttime = correct(gmttime) ;
```

เอกสารนี้สงวนลิขสิทธิ์สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    gsttime = gmttogst(day,gmttime,jlday) ;
    gsttime = correct(gsttime) ;
    lsttime = gsttolst(place,gsttime) ;
    lsttime = correct(lsttime) ;

    switch(obj)
    {
        case 0 : starmenu(1) ;
position = get_star() ;
break ;
        case 1 : position = planetpos('m',jlday) ;
position = ecliptoequa(position,jlday) ;
break ;
        case 2 : position = planetpos('v',jlday) ;
position = ecliptoequa(position,jlday) ;
break ;
        case 3 : position = planetpos('r',jlday) ;
position = ecliptoequa(position,jlday) ;
break ;
        case 4 : position = planetpos('j',jlday) ;
position = ecliptoequa(position,jlday) ;
break ;
        case 5 : position = planetpos('s',jlday) ;
position = ecliptoequa(position,jlday) ;
break ;
        case 6 : position = planetpos('u',jlday) ;
position = ecliptoequa(position,jlday) ;
break ;
        case 7 : position = planetpos('n',jlday) ;
position = ecliptoequa(position,jlday) ;
break ;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
      case 8 : position = planetpos('p',jlday) ;
position = ecliptoequa(position,jlday) ;
break ;
      case 9 : position = sunpos(jlday) ;
position = ecliptoequa(position,jlday) ;
break ;
      case 10 : position = moonpos(jlday) ;
position = ecliptoequa(position,jlday) ;
break ;
 }
 position = equatohori(position,lsttime,countrys) ;

 send(position.lattitude) ;
 send(position.longtitude) ;

 if(position.lattitude<0)
 {
   setcolor(RED) ;
   outtextxy(20,80,"IT IS NOT ON THE SKY NOW!") ;
   getch() ;
   setviewport(0,0,getmaxx(),getmaxy(),0) ;
   putimage(200,100,ptr,COPY_PUT) ;
   free(ptr) ;
   return  ;
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

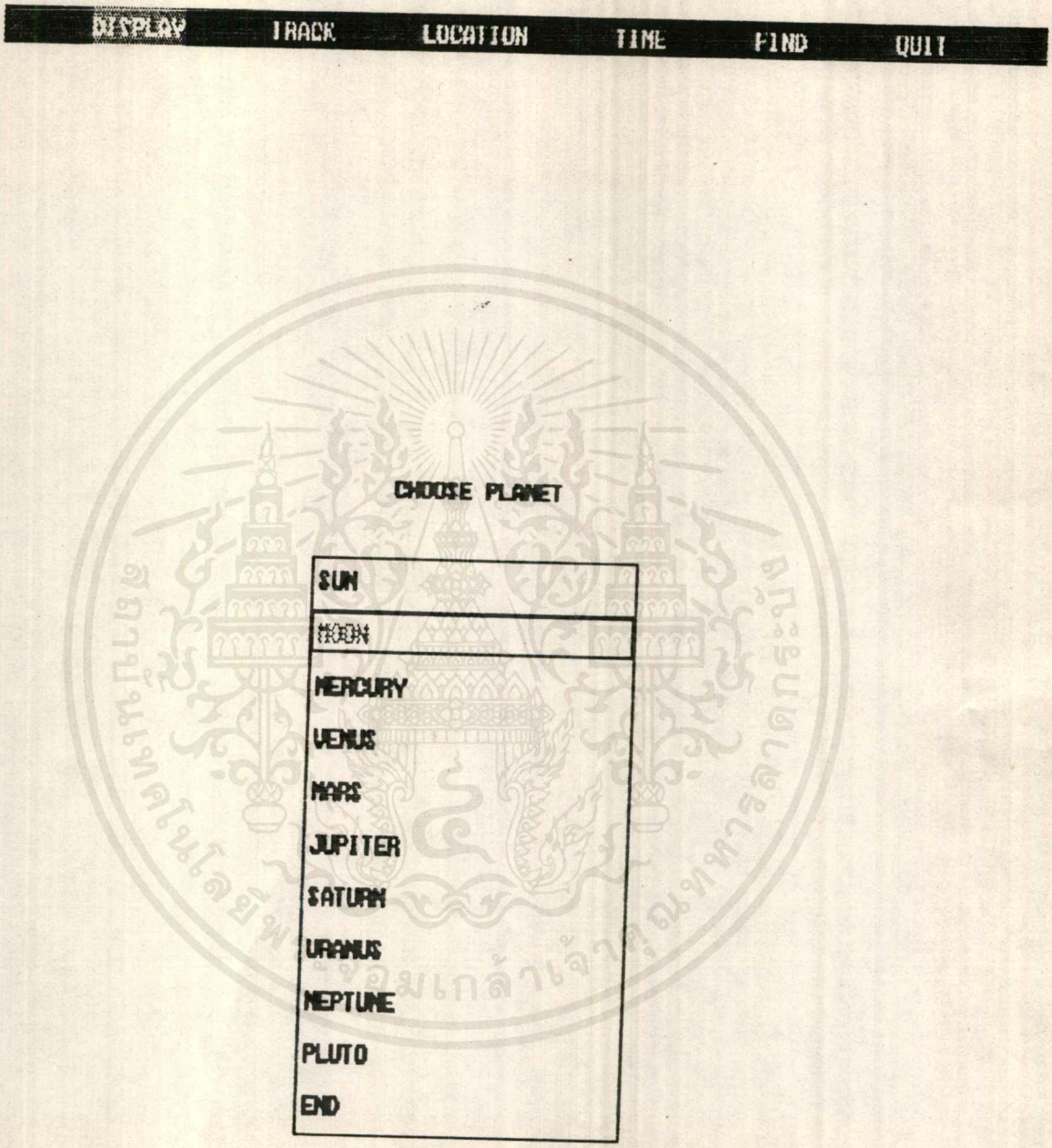
```
#include<dos.h>
#include<stdio.h>

#define PORT 1
#define CODE 219

void send() ;
void send_port() ;

void send(output)
unsigned int ;
{
   union REGS r ;
   unsigned char out1,out2 ;

   /* initialize port */
   r.x.dx = PORT ;
   r.h.ah = 0 ;
   r.h.al = CODE ;
   int86(0x14,&r,&r) ;

   output = (output*4095)/10 ;
   out1 = (unsigned char)(output%128) ;
   out2 = (unsigned char)(output/128) ;

   /*send low byte*/
   send_port(0x80) ;
   send_port(out2) ;

   /*send high byte*/
   send_port(0x80) ;
   send_port(out1) ;
}
```

เอกสารนี้จัดทำขึ้นไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
void send_port(output)
unsigned char output ;
{
   while(!kbhit())
   {
      union REGS r ;
      r.x.dx = PORT ;
      r.h.ah = 1 ;
      r.h.al = output ;
      int86(0x14,&r,&r) ;
      if(r.h.ah&128)
      {
setcolor(RED) ;
outtextxy(40,80,"SEND ERROR DETECTED") ;
setcolor(BLACK) ;
      }
   }
}
```

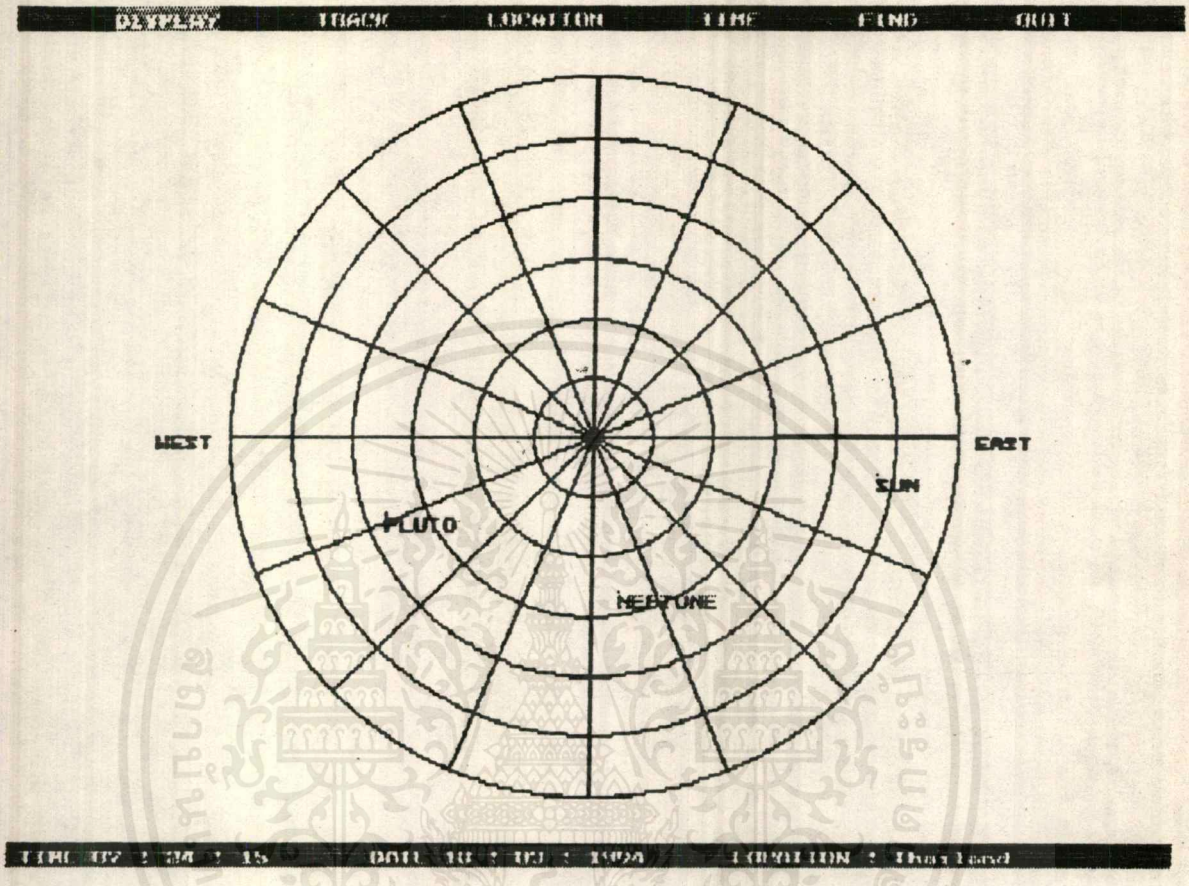
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทำงานของส่วนแสดงผล แสดงได้ดังรูปต่อไปนี้



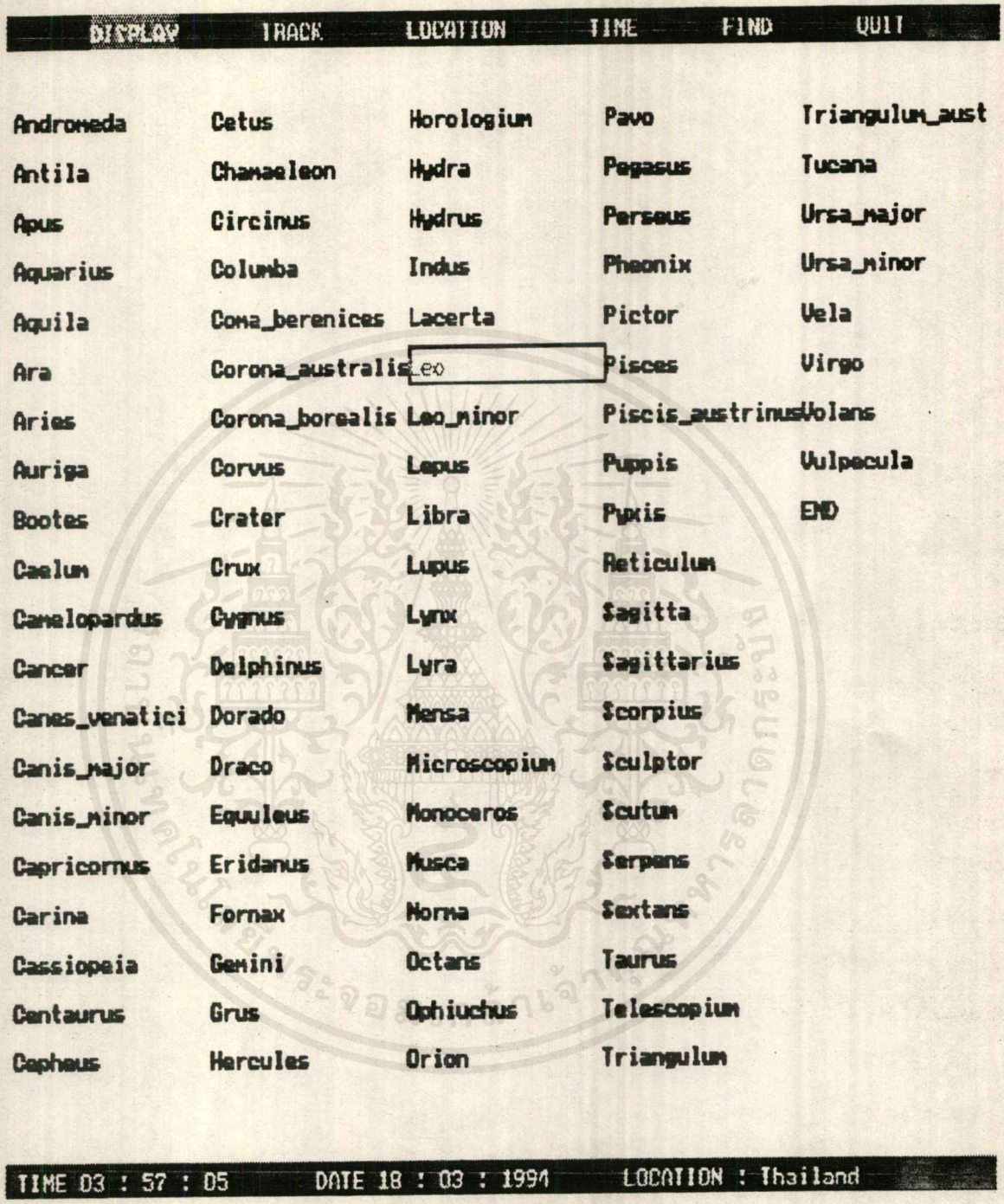
รูปที่ 5 แสดงรายชื่อ ดวงอาทิตย์ ดวงจันทร์ ดาวเคราะห์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



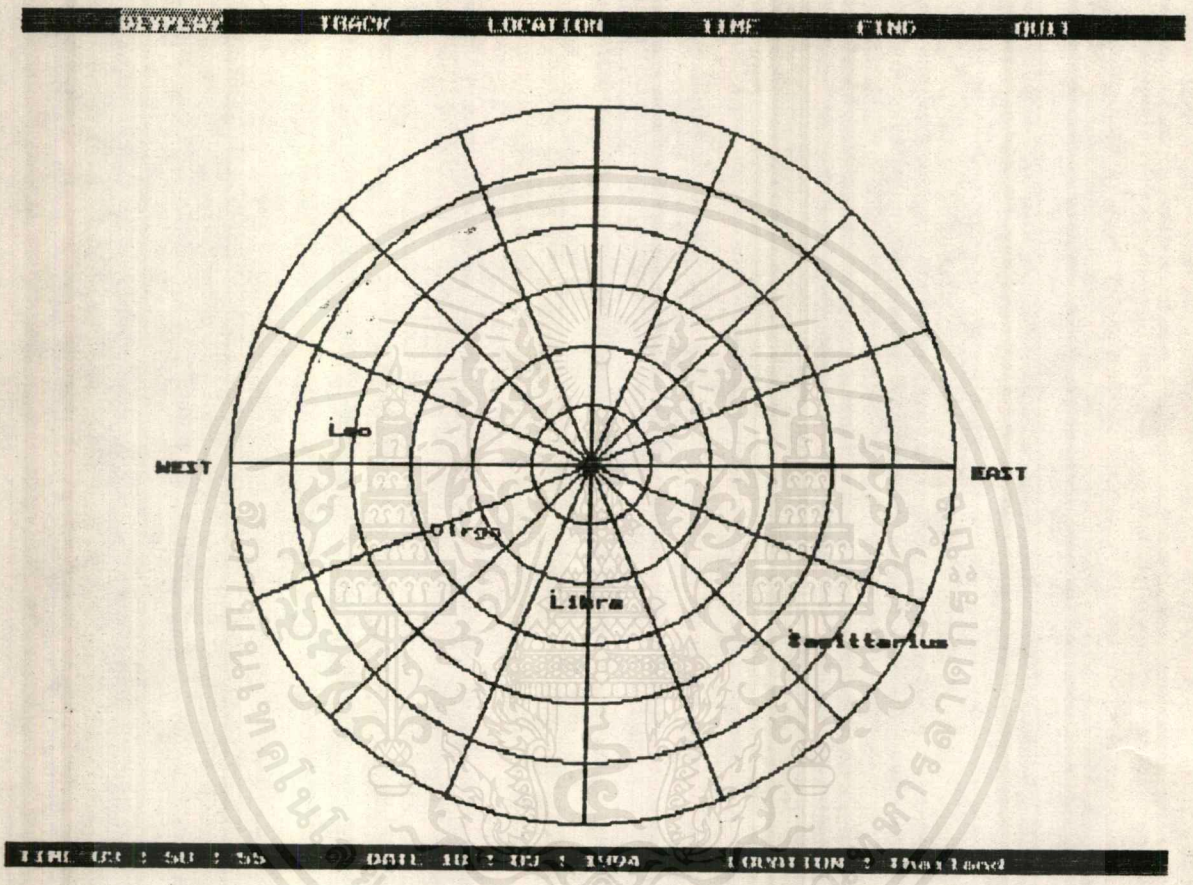
รูปที่ 6 ผลการ Display ของ Planet fix

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



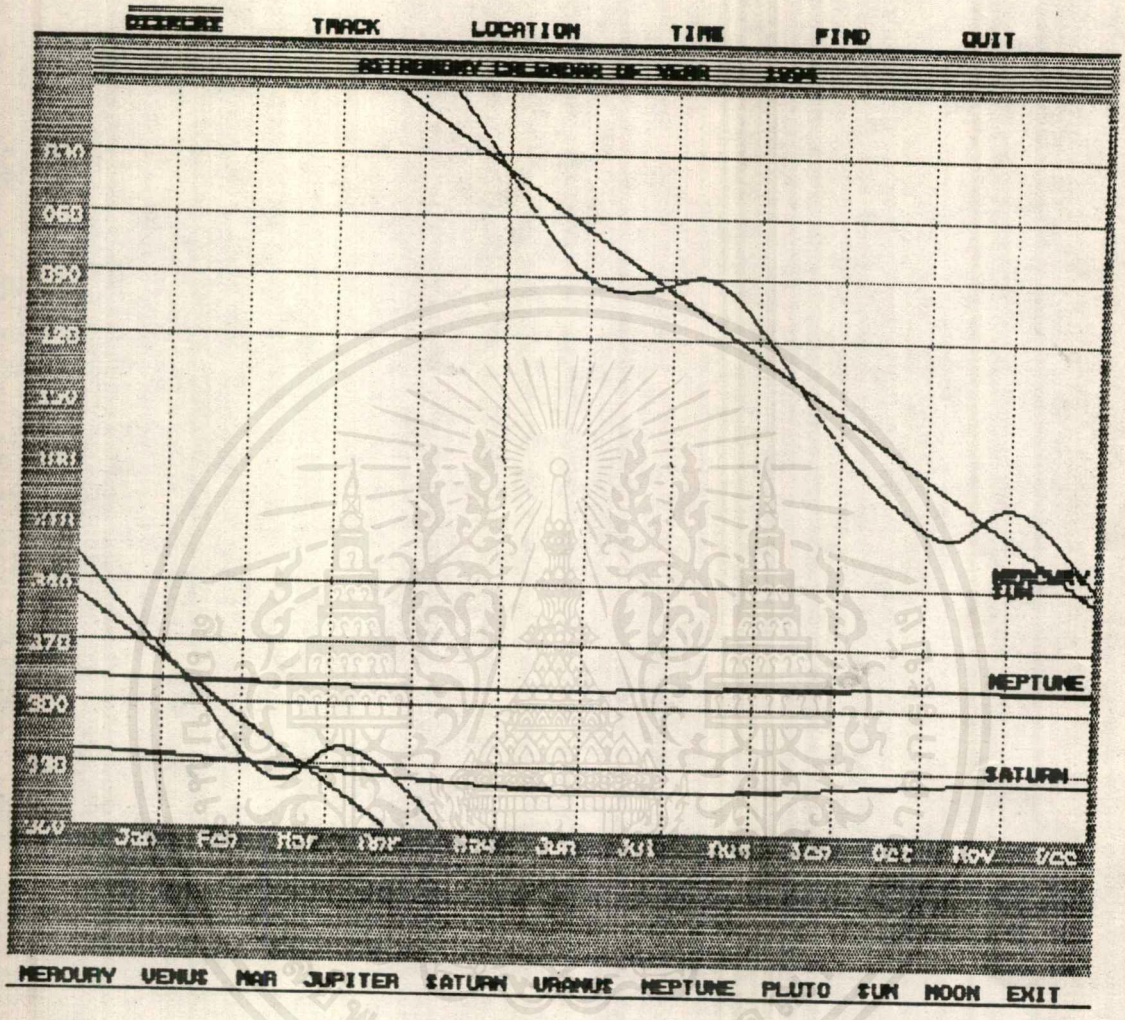
รูปที่ 7 แสดงรายชื่อ ดาวฤกษ์ ทั้ง 88 กลุ่ม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



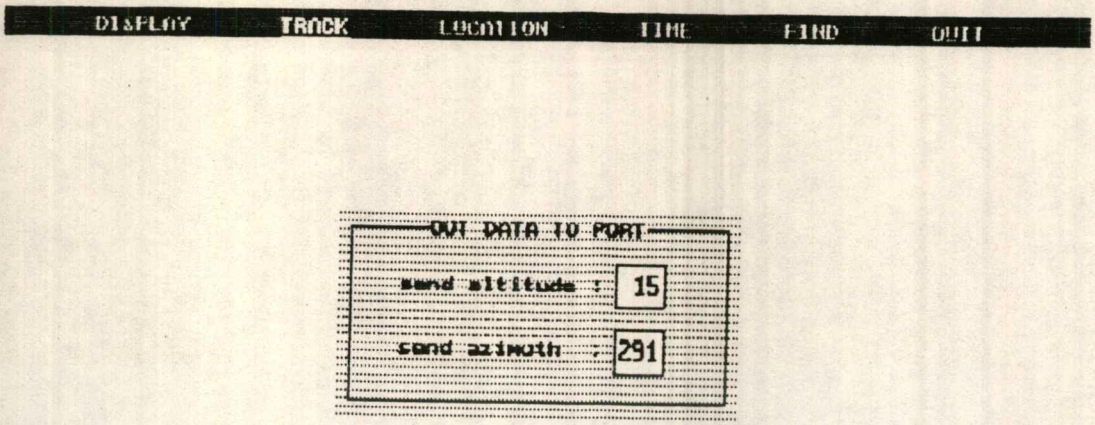
รูปที่ 8 ผลการ Display ของ starfix

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

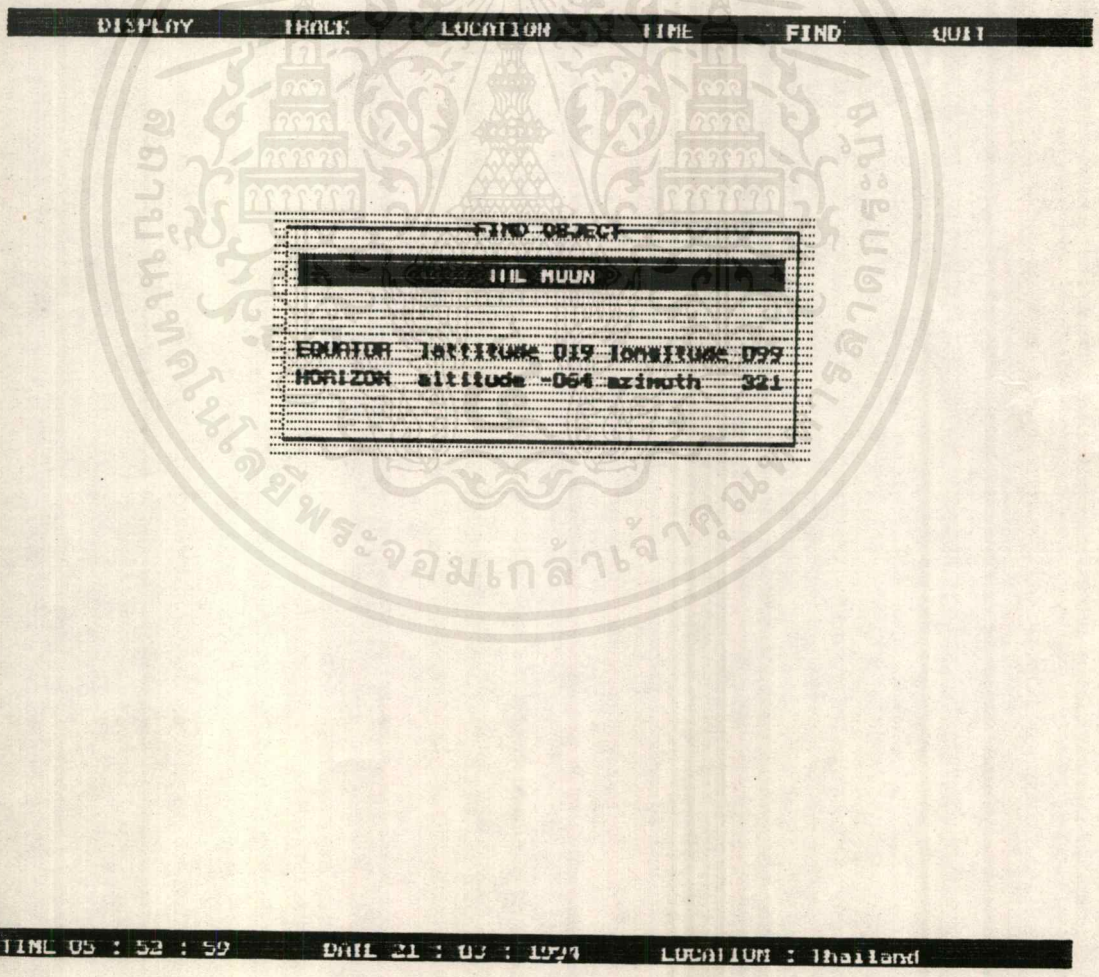


รูปที่ 9 ปฏิทินดาราศาสตร์ ใน ปี 1994

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

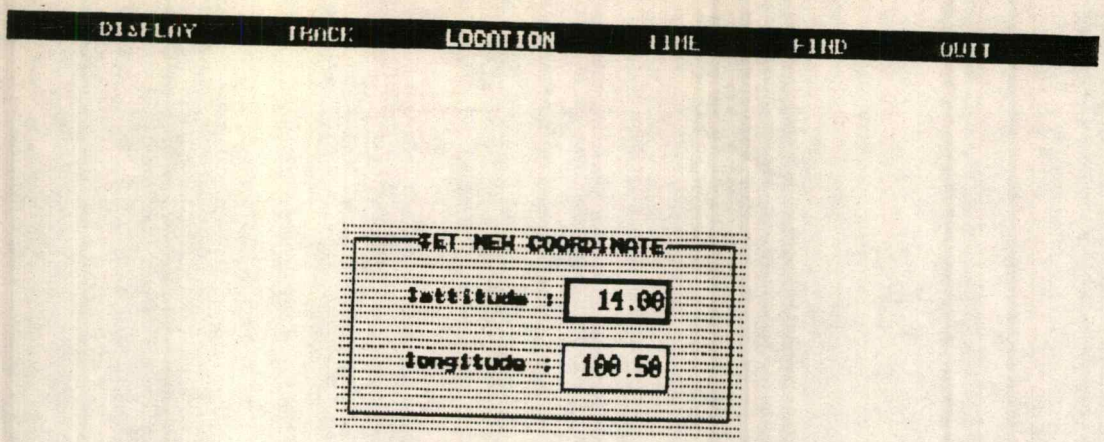


รูปที่ 10 แสดงการทำงานของคำสั่ง Track



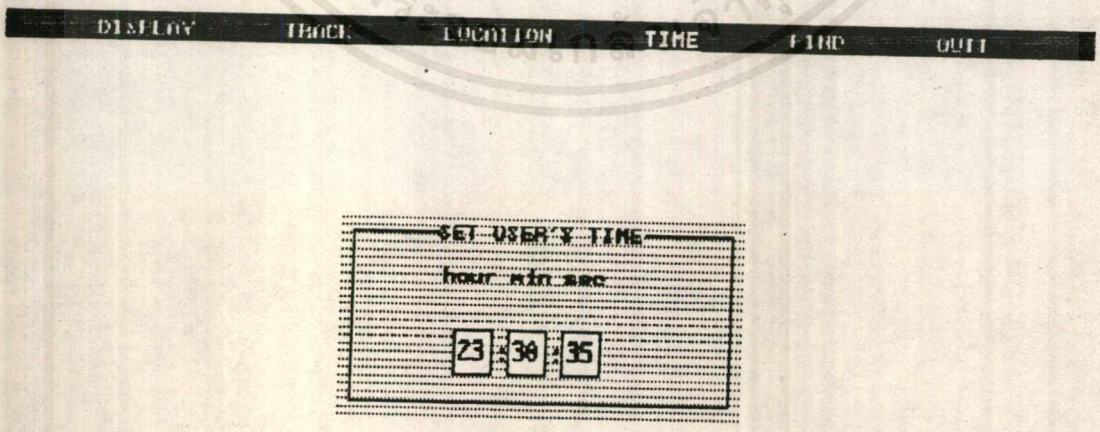
รูปที่ 11 แสดงการทำงานของคำสั่ง Find

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 12 แสดงการแก้ไขข้อมูล ละติจูด ลองติจูด

รูปที่ 13 แสดงการแก้ไขข้อมูลประเทศ



รูปที่ 14 แสดงการแก้ไขข้อมูลเวลา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Hardware ของระบบติดตามดาว

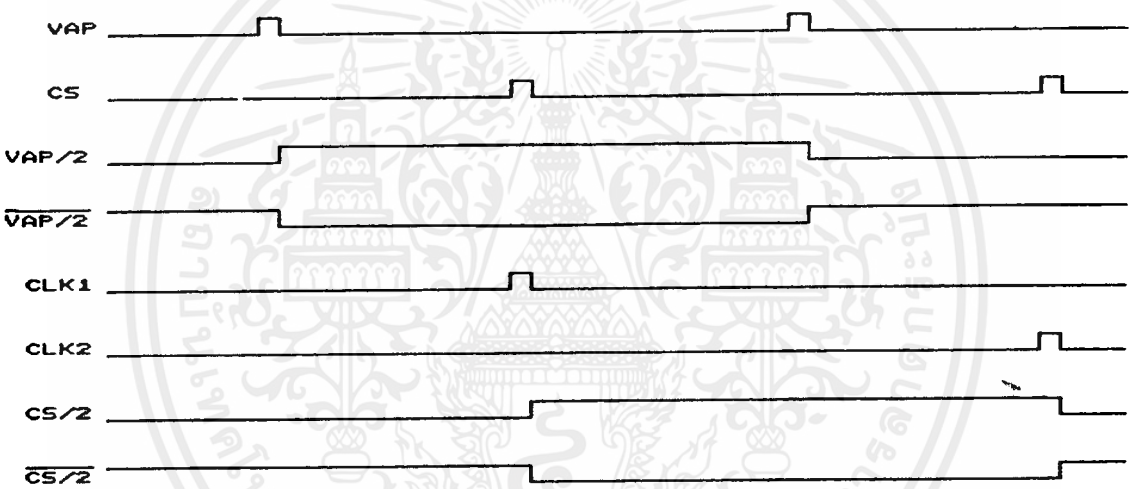
Hardware ของระบบติดตามดาวทั้งหมด จะประกอบไปด้วย

1.ส่วนรับค่าจากคอมพิวเตอร์ และแปลงค่าจาก Digital เป็น Analog

2.ชุดควบคุม Position ของ Motor

3.มอเตอร์ที่ Coupling กับกล้องดูดาว

ส่วนที่ทำไปแล้ว คือ ส่วนรับค่าจากคอมพิวเตอร์ และแปลงค่าจาก Digital เป็น Analog โดยสัญญาณที่ได้คือ ค่าโวลท์เตจค่าต่ำสุด vary ไปจนถึงค่าโวลท์เตจสูงสุด ซึ่งแสดงได้ด้วยวงจรดังรูป และมี Timing diagram ดังรูป



รูปที่ 15 Timing Diagram

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

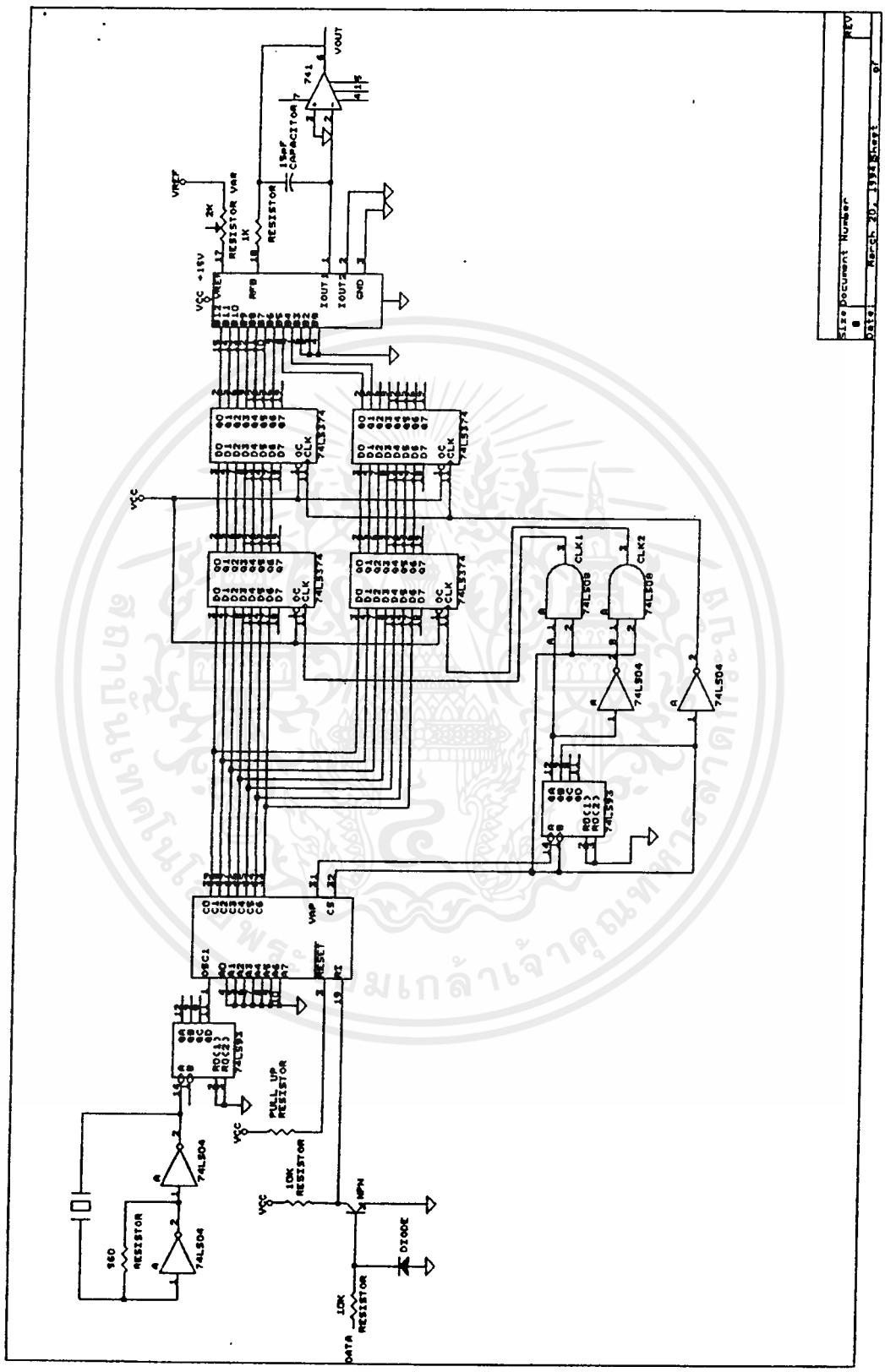
หลักการทำงานของวงจร

เราต้องการส่งข้อมูลที่มีค่า 0 ถึง 360 ดังนั้นเราต้องการ 9 บิท สำหรับจัดการข้อมูลของเรา เราจะส่งข้อมูลออกจากคอมพิวเตอร์ 2 ครั้ง ครั้งละ 7 บิท โดยส่งไบท์ที่มีความสำคัญต่ำกว่าออกไปก่อน แล้วจึงส่งไบท์ที่มีความสำคัญมากกว่าตามมา ข้อมูลจะส่งออกมาทางพอร์ตอนุกรมเข้า IC เบอร์ MC 14469 ซึ่งจะทำหน้าที่เป็น transmitter แปลงข้อมูลจาก อนุกรมเป็นขนาน

ข้อมูลไบท์ที่มีความสำคัญต่ำกว่าจะนำไปเก็บไว้ที่ latch 1 ข้อมูลที่มีความสำคัญสูงกว่าจะเก็บไว้ที่ latch 2 และข้อมูลจะถูกส่งต่อไปยัง latch 3,latch 4 ก่อนที่จะส่งให้กับ DAC ต่อไป การควบคุมการส่งข้อมูลไปให้ latch เป็นไปตาม Timing Diagram ในรูป

ข้อมูลจาก latch 9 บิท ที่มีความสำคัญต่ำกว่าจะส่งไปยัง DAC PM 7451 เพื่อแปลงเป็นสัญญาณ Analog สำหรับนำไปขับมอเตอร์ต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 16 วงจรแปลง Digital เป็น Analog

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สรุปผลและวิจารณ์

ผลการทำงานของโปรแกรมที่ได้มีการผิดพลาดน้อยมาก เมื่อพิจารณาในตารางที่ 1 ซึ่งเป็นการเปรียบเทียบผลการทำงานของโปรแกรม กับ Astronomical Ephemeris

วัน เดือน ปี	ชื่อดาว	ผลจากโปรแกรม	Ephemeris
1 JAN 1993	ดวงอาทิตย์	280.59	281
	ดวงจันทร์	9.04	9
	ดาวพุธ	267.97	268
	ดาวศุกร์	326.49	327
	ดาวอังคาร	111.53	110
	ดาวพฤหัส	193.22	193
	ดาวเสาร์	318.93	316
	ดาวยูเรนัส	288.35	286
	ดาวเนปจูน	287.64	286
	ดาวพลูโต	233.28	235

ตารางที่ 1 แสดงค่าไรท์แอสเซนชั่นที่ได้จากโปรแกรมกับ Ephemeris

แนวทางในการทำโครงงานต่อไป

เป้าหมายของโครงงานที่ตั้งไว้ มี 3 ข้อคือ

1. สามารถคำนวณตำแหน่งของดาวได้อย่างแม่นยำ และส่งค่าที่คำนวณได้ออกมายัง Hardware
2. สามารถควบคุมตำแหน่งของ Motor ที่ต่ออยู่กับกล้อง ให้เลื่อนไปยังตำแหน่งที่ต้องการได้
3. สามารถหาตำแหน่งของดาวเหนือได้โดยอัตโนมัติ

ซึ่งจากเป้าหมายดังกล่าว รวมถึงส่วนของโครงงานที่ได้ทำไปแล้ว ทำให้มีแนวทางในการทำโครงงานต่อไปได้ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การยึดติดแกนกล้องดูดาวเข้ากับมอเตอร์ โดยให้มีแนวหมุนกล้อง 2 แนวตั้งฉากกัน แต่ละแกนจะประกอบด้วยชุดเฟืองหนอน โดยชุดเฟืองหนอนตัวล่าง จะขับชุดเฟืองหนอนตัวบนและตัวกล้องดูดาว ส่วนชุดเฟืองหนอนตัวบนจะขับเฉพาะตัวกล้องเท่านั้น (เหตุที่ต้องใช้เฟืองหนอนในการขับแกน ก็เพราะชุดเฟืองหนอนสามารถรับโหลดได้มาก จึงเหมาะสมในการใช้งาน เนื่องจากตัวกล้องน้ำหนักมาก)

ในการดูดาว เราจะต้องตั้งแกนกล้อง (จุดศูนย์กลางของแกนเฟืองตัวล่าง) ให้ชี้ไปที่ขั้วเหนือท้องฟ้า แล้วจึงเลื่อนตัวกล้องไปจับดาวที่ต้องการศึกษา โดยการควบคุมให้มอเตอร์หมุนเกลียวหนอนทั้ง 2 ชุดไปยังตำแหน่งที่ต้องการ ส่วนในการติดตามดาวอาจจะอาศัยหลักการอีกอย่างหนึ่งที่ว่า ดาวทุกดวงจะเคลื่อนที่ด้วยอัตราเร็วเชิงมุมคงที่ จากทิศตะวันออกไปยังทิศตะวันตก อันเนื่องมาจากการหมุนรอบตัวเองของโลก

ดังนั้นถ้าเราต้องการจะติดตามดาว เราก็ต้องควบคุมให้เฟืองตัวล่างหมุนด้วยความเร็วเท่ากับความเร็วที่ดาวเคลื่อนที่ นอกเหนือจากการคำนวณตำแหน่งจากคอมพิวเตอร์ และส่งข้อมูลออกมาเพื่อให้มอเตอร์หมุนไปยังตำแหน่งที่คำนวณได้ในอัตราเร็วตามที่กำหนด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หนังสืออ้างอิง

1. ดาราศาสตร์เชิงปฏิบัติการกับเครื่องคิดเลขของท่าน , ม.รามคำแหง

2. Nicholas Campion , The Practical Astrologer

3. Herbert Schildt , การประยุกต์ใช้งานภาษาซี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก

ตารางที่ 2 แสดงค่าเดคลิเนชันและไรท์แอสเซนชันของกลุ่มดาว

กลุ่มดาว	เดคลิเนชัน (องศา)	ไรท์แอสเซนชัน (องศา)
Andromeda	15	+40
Antila	150	-35
Apus	240	-75
Aqurius	345	-15
Aquila	300	+5
Ara	255	-55
Aries	45	+20
Auriga	90	+40
Bootes	225	+30
Caelum	75	-40
Camelopardus	90	-70
Cancer	135	+20
Canes Venatici	195	+40
Canis Major	105	-20
Canis Minor	120	+55
Capricornus	315	-20
Carina	135	-60
Cassiopeia	15	+60
Centaurus	195	-50
Cepheus	330	+70
Cetus	30	-10
Chamaeleon	165	-80
Circinus	225	-60

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กลุ่มดาว	เดคลิเนชัน (องศา)	ไรท์แอสเซนชัน (องศา)
Columba	90	-35
Coma Berenices	195	+20
Corona Australis	285	-40
Corona Borealis	240	+30
Corvus	180	-20
Crater	165	-15
Crux	180	-60
Cygnus	315	+40
Delphinus	315	+10
Dorado	75	-65
Draco	255	+65
Equuleus	315	+10
Eridanus	45	-20
Fornax	45	-30
Gemini	105	+20
Grus	330	-45
Hercules	255	+30
Herologium	45	-60
Hydra	150	-20
Hydrus	30	-75
Indus	315	-55
Lacerta	330	+45
Leo	165	+15
Leo Minor	150	+35
Lepus	90	-20

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กลุ่มดาว	เดคลิเนชัน (องศา)	ไรท์แอสเซนชัน (องศา)
Libra	225	-15
Lupus	225	-45
Lynx	120	+45
Lyra	285	+40
Mensa	75	-80
Microscopium	315	-35
Monoceros	105	-5
Musca	180	-70
Norma	240	-50
Octans	330	-85
Ophiuchus	255	0
Orion	75	+5
Pavo	300	-65
Pegasus	330	+20
Perseus	45	+45
Pheonix	15	-50
Pictor	90	-55
Pisces	15	+15
Piscis Austrinus	330	-30
Puppis	120	-40
Pyxis	135	-30
Reticulum	60	-60
Sagitta	300	+10
Sagittarius	285	-25
Scorpius	255	-40
Sculptor	0	-30

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กลุ่มดาว	เดคลิเนชัน (องศา)	ไรท์แอสเซนชัน (องศา)
Scutum	285	-10
Serpens	255	0
Sextans	150	0
Taurus	60	+15
Telescopium	285	-50
Triangulum Australe	30	+30
Tucana	0	-65
Ursa Major	165	-65
Ursa Minor	225	+50
Vela	135	-50
Virgo	195	0
Volans	120	-70
Vulpecula	300	+25

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้