



เครื่องปรับมุมสายอากาศตามฐานข้อมูล
ANGLE ADJUSTMENT OF ANTENNA
BY DATABASE

โดย

นายประวิทย์ ลิทธิ เตชากุล

นายพงษ์พันธ์ ธาราพงษ์

นายศักดิ์ชัย วิจิตรบรรจงดี

นายสมชาย บุญณรินทร์

นายสุชิน ประสพฤกษ์

วัน เดือน ปี..... 19 ธ.ค. 2537
เลขทะเบียน..... 034 953
เลขเรียกหนังสือ..... T37 253 24

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมโทรคมนาคม

สถาบัน เทคโนโลยีพระจอมเกล้า เจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2537

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไปว่ากรณีใดที่ผู้รับใช้เอกสารนี้เพื่อวัตถุประสงค์อื่นใดโดยไม่ได้รับอนุญาตจากผู้จัดทำเอกสารหรือผู้ที่เกี่ยวข้อง

เครื่องปรับมุมสายอากาศตามฐานข้อมูล
ANGLE ADJUSTMENT OF ANTENNA BY DATABASE

โดย

นายประวิทย์	สิทธิ เดชากุล	35.103015
นายพงษ์พันธ์	ธाराพงษ์	35.103017
นายศักดิ์ชัย	วิจิตรบรรจงดี	35.103029
นายสมชาย	บุญณรินทร์	35.103031
นายสุชิน	ประสพฤกษ์	35.103034

อาจารย์ที่ปรึกษา

รศ. แรงค์	เหมกรณ์
ผศ. นิภา	ลีลารุจิ

ปริญญานิพนธ์นี้ เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมโทรคมนาคม

คณะวิศวกรรมศาสตร์

สถาบัน เทคโนโลยีพระจอมเกล้า เจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2537

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์ปีการศึกษา 2537

ภาควิชาวิศวกรรมโทรคมนาคม

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
เรื่อง เครื่องปรับมุมสายอากาศตามฐานข้อมูล

ผู้จัดทำ

- 
- | | |
|-----------------|---------------|
| 1. นายประวิทย์ | สิทธิ เตชากุล |
| 2. นายพงษ์พันธ์ | ธाराพงษ์ |
| 3. นายศักดิ์ชัย | วิจิตรบรรจงดี |
| 4. นายสมชาย | บุญณรินทร์ |
| 5. นายสุชิน | ประสพฤกษ์ |

..... อาจารย์ที่ปรึกษา

(รต. แรงค์ เหมกรณ์)

..... อาจารย์ที่ปรึกษา

(ผศ. นิภา ลีลารุจิ)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เครื่องปรับมุมสายอากาศตามฐานข้อมูล

ANGLE ADJUSTMENT OF ANTENNA BY DATABASE

โดย นายประวิทย์ ลีทธิ เตชากุล
นายพงษ์พันธ์ ธาราพงษ์
นายศักดิ์ชัย วิจิตรบรรจงดี
นายสมชาย บุญนิรันดร์
นายสุชิน ประสพฤกษ์

อาจารย์ที่ปรึกษา รศ. ทรงศักดิ์ เหมกรณ์
ผศ. นิภา ลีลารุจิ

บทคัดย่อ

งานปริญาโทฉบับนี้ จะกล่าวถึงรายละเอียดการทำงานของระบบควบคุมสายอากาศติดตามดาวเทียมวงโคจรต่ำ เนื่องจากดาวเทียมวงโคจรต่ำจะมีวงโคจรและตำแหน่งเปลี่ยนแปลงอยู่เสมอ ดังนั้นระบบควบคุมสายอากาศติดตามดาวเทียมวงโคจรต่ำจะใช้ข้อมูลสำรวจโคจรของดาวเทียมป้อนเข้าเครื่องคอมพิวเตอร์ เพื่อทำการประมวลผล และนำข้อมูลที่ได้จากการประมวลผลส่งไปยังซีพียู เบอร์ Z-80 ซึ่งจะใช้ข้อมูลดังกล่าวทำการควบคุมการปรับค่ามุมอาซิมุทและมุมอีเลเวชันของสายอากาศให้ชี้ไปยังตำแหน่งของดาวเทียมดวงที่ต้องการรับสัญญาณ ได้อย่างถูกต้อง

ABSTRACT

THIS THESIS WILL PRESENT ABOUT THE OPERATING OF LOW EARTH ORBIT SATELLITE ANTENNA TRACKING SYSTEM . BECAUSE OF LOW EARTH ORBIT SATELLITE WILL ALWAYS MOVING TARGET.SO THIS SYSTEM WILL USE INFORMATION OF ORBITAL SATELLITE BE INPUT TO COMPUTER . THIS COMPUTER WILL PROCESS THIS INFORMATION AND SEND THE RESULTS FROM COMPUTER TO CPU Z-80 . CPU Z-80 WILL USE THE RESULTS FROM COMPUTER TO CONTROL THE ADJUSTMENT OF AZIMUTH AND ELEVATION ANGLE OF ANTENNA. SO ANTENNA WILL CORRECTLY POINT TO THE SATELLITE.

สารบัญ

หน้า

บทที่ 1	บทนา	1
บทที่ 2	ทฤษฎีหรือหลักการ	2
บทที่ 3	การทดลอง และผลการทดลอง	37
บทที่ 4	บทสรุป	56
หนังสืออ้างอิง		
ภาคผนวก ก		
ภาคผนวก ข		
ภาคผนวก ค		

บทที่ 1

บทนำ

งานการเคลื่อนที่ของดาว เทียมวงโคจรต่ำ ส่วนใหญ่วงโคจรจะมีลักษณะเป็นรูปวงรี โดยอาศัยแรงเหวี่ยงของโลก เพื่อรักษาสภาพสมดุลในขณะที่ดาวเทียมโคจรอยู่ในวงโคจร เวลาที่ใช้ในการโคจรของดาวเทียมแบบนี้ จะเร็วกว่าเวลาที่ใช้ในการหมุนรอบตัวเองของโลก จึงสามารถนำมาใช้ในการติดต่อสื่อสารโดยใช้เทคนิคทางดิจิทัลโดยการรับข้อมูลจากพื้นที่หนึ่ง แล้วส่งไปยังอีกพื้นที่หนึ่ง เมื่อดาวเทียมนั้นโคจรผ่านไป เช่น ดาวเทียมสำรวจทรัพยากร ดาวเทียมตรวจอากาศ ดาวเทียมจารกรรม ฯลฯ ซึ่งแตกต่างจากดาวเทียมวงโคจรอยู่กับที่ เช่น ไทยคม, บาลาป้า ฯลฯ ที่เวลาการโคจรของดาวเทียมจะเท่ากับเวลาในการหมุนรอบตัวเองของโลก จึงทำให้เสมือนว่าดาวเทียมนั้นอยู่นิ่งกับที่ วงโคจรของดาวเทียมวงโคจรต่ำในแต่ละรอบจะมีการเปลี่ยนแปลงของค่าตัวแปรต่าง ๆ อยู่ตลอด เช่น ค่าเวลาที่ดาวเทียมโคจรผ่านพื้นที่ใด ๆ ของโลก ค่าความเร็วในการโคจรของดาวเทียม ค่ามุมของดาวเทียมที่ทาต่อโลก ฯลฯ ลักษณะการส่งสัญญาณของดาวเทียม เป็นแบบรอบตัวจึงทำให้สามารถใช้สายอากาศที่มีมุมรับในการรับสัญญาณกว้างได้โดยไม่มีจำเป็นต้องมีเกนการขยายสูง เช่น สายอากาศแบบครอสยาก็

ในระบบสายอากาศติดตามดาวเทียมวงโคจรต่ำประกอบด้วย เครื่องคอมพิวเตอร์ บอร์ดไมโครโปรเซสเซอร์ Z-80 ชุดขับสแตปปีงมอเตอร์ ระบบเฟืองหมุนสายอากาศ และ ชุดเซ็นเซอร์มุมของสายอากาศ เครื่องคอมพิวเตอร์ จะทำหน้าที่ในการรับข่าวสารการโคจรของดาวเทียมโดยผ่านทางแป้นพิมพ์ทำการประมวลผลข่าวสารดังกล่าว และส่งผลลัพธ์ที่ได้จากการประมวลผล ซึ่งเป็นข้อมูลเกี่ยวกับ เวลาและตำแหน่งของดาวเทียม (มุมอาซิมุทและมุมอีเลเวชัน) ไปยังบอร์ดไมโครโปรเซสเซอร์ Z-80 โดยผ่านทาง RS-232 ของเครื่องคอมพิวเตอร์ และผ่านบอร์ด RS-232 ซึ่งมีการรับส่งข้อมูลแบบอนุกรม ข้อมูลที่ส่งมาจะถูกเก็บไว้ในหน่วยความจำบนบอร์ดไมโครโปรเซสเซอร์ Z-80 จากนั้นซีพียู Z-80 จะนำข้อมูลดังกล่าวไปทำการควบคุมการขับสแตปปีงมอเตอร์ เพื่อทำการปรับค่ามุมอาซิมุท

และมุมอีเลเวชันของสายอากาศของสายอากาศ โดยการขับระบบกลไกเฟือง

เอกสารนี้เป็นเอกสารสงวนไว้สำหรับใช้ภายในเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2

ทฤษฎี

2.1 การติดตามดาวเทียม

ในที่นี้เราจะพิจารณาพื้นฐานการติดตามดาวเทียมว่า หมายถึงอะไร หมายความว่าต้องติดตาม และทว่าอย่างไร สำหรับนักวิทยาศาสตร์การติดตามดาวเทียม หมายถึง ความสามารถในการระบุตำแหน่งของดาวเทียมในอวกาศได้ สำหรับนักวิทยุสมัครเล่น การติดตามดาวเทียมจะเกี่ยวข้องกับการปฏิบัติมากกว่ากล่าวคือ เมื่อไหร่ดาวเทียมจะอยู่ในช่วงที่เราสามารถติดต่อได้ และสายอากาศควรที่จะชี้ไป ในทิศทางใด โดยทั่วไปดาวเทียมจะเป็นวัตถุที่เคลื่อนที่ ดังนั้น สถานีภาคพื้นดิน (GROUND STATION) ที่ใช้สายอากาศแบบมีทิศทาง (DIRECTIONAL ANTENNA) จะต้องชี้ข่าวสารโดยเฉพาะของดาวเทียม ซึ่งเป็นข่าวสารเกี่ยวกับวงโคจรของดาวเทียม ความสามารถในการทำนายเวลาที่สามารถติดต่อกับดาวเทียมได้ (ACCESS TIME) นั้นเป็นสิ่งที่สำคัญมาก เพราะว่าดาวเทียมวงโคจรต่ำส่วนใหญ่จะอยู่ในช่วงเวลาที่แต่ละสถานีภาคพื้นดินสามารถติดต่อได้ เป็นเวลาเพียงบางช่วงของแต่ละวันเท่านั้น ซึ่งแตกต่างจากดาวเทียมแบบวงโคจรอยู่นิ่งกับที่ (GEOSTATIONARY SATELLITE) ซึ่งเป็นดาวเทียมที่รักษาตำแหน่งอย่างคงที่เหนือเส้นศูนย์สูตรในที่นี้จะไม่กล่าวถึง

ดาวเทียมวงโคจรต่ำ (LOW ALTITUDE SATELLITE) เช่น ฟุจิ-ออสการ์ 20 (Fuji-OSCAR20) อาร์เอส-10/11 (RS-10/11) หรือ ไมโครแซท (MicroSat) โดยทั่วไปจะอยู่ในช่วงเวลาที่เราสามารถติดต่อได้น้อยกว่า 25 นาที ในแต่ละครั้งที่ดาวเทียมผ่านเข้ามาใกล้สถานีภาคพื้นดิน ตามปกติเราจะเห็นดาวเทียมผ่านเข้ามาสี่ถึงหกครั้งต่อวันสำหรับดาวเทียมแต่ละดวง ผลดังกล่าวนี้ ทำให้ดาวเทียมที่ทำงานตลอด 24 ชั่วโมงต่อวัน จะสามารถติดต่อกับสถานีภาคพื้นดินแต่ละแห่งได้เพียง 1-2 ชั่วโมง ในแต่ละวันเท่านั้น เวลาโดยเฉลี่ยในการติดต่อกับดาวเทียม สำหรับสถานีภาคพื้นดินแต่ละแห่งนั้นเป็นสิ่งสำคัญมาก ซึ่งเป็นการบอกถึงประโยชน์ของดาวเทียมที่มีต่อสถานีภาคพื้นดินแห่งนั้น

นักวิทยุสมัครเล่นที่ต้องการจะติดตามดาวเทียมวงโคจรต่ำ ต้องรู้ข่าว เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารดังต่อไปนี้

1. ข่าวสารเกี่ยวกับเวลาว่า เมื่อเวลาเท่าใดดาวเทียมจะอยู่ในช่วงที่สถานีภาคพื้นดินจะสามารถติดต่อได้ ข่าวสารดังกล่าวคือ เวลาที่สัญญาณเริ่มรับได้ (ACQUISITION OF SIGNAL:AOS) และเวลาที่สัญญาณเริ่มรับไม่ได้ (LOSS OF SIGNAL:LOS) เมื่อดาวเทียมดวงที่ต้องการติดต่อนั้น ได้โคจรเคลื่อนที่ผ่านเข้ามาใกล้กับสถานีภาคพื้นดินแห่งนั้น

2. ข่าวสารเกี่ยวกับสายอากาศควรรีชีเปนทิศทางใด ข่าวสารดังกล่าว คือ ค่ามุมอาซิมุทและมุมอีเลเวชัน ที่ขณะเวลาใด ๆ ที่สายอากาศควรรีชีเปให้ตรงกับตำแหน่งของดาวเทียม

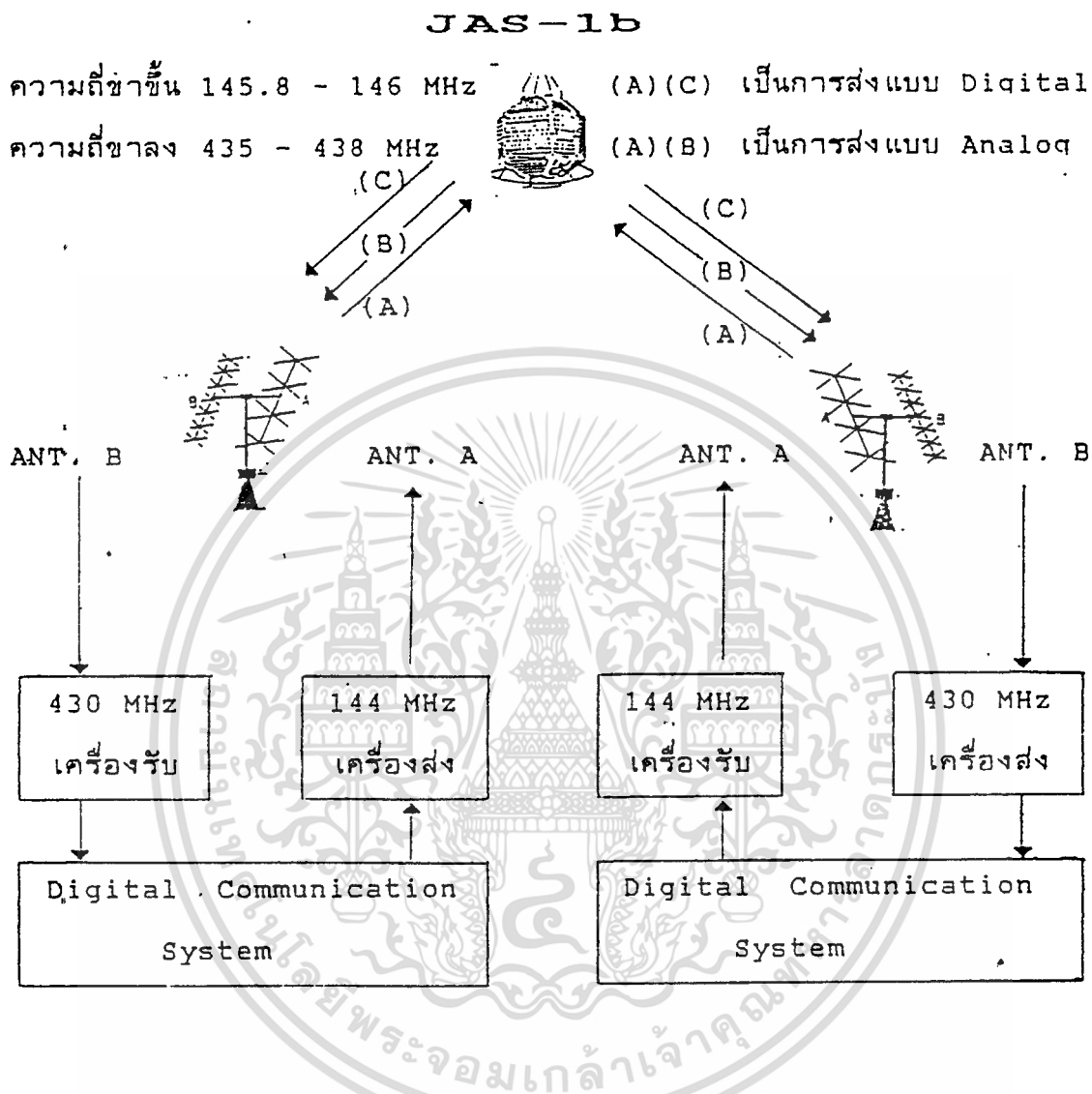
3. ข่าวสารเกี่ยวกับบริเวณส่วนใดส่วนหนึ่งของโลกที่สามารถติดต่อกับดาวเทียมได้ ณขณะเวลานั้น ๆ

ในการติดตามนั้น เราจำเป็นต้องทำการปรับปรุง (UPDATE) ข่าวสารที่บอกถึงวงโคจรของดาวเทียมที่เราสนใจ ข่าวสารดังกล่าวนี้เรียกว่า ข้อมูลอ้างอิงวงโคจร (REFERENCE ORBIT DATA) หรือค่าตัวแปรต่างๆ ที่เกี่ยวข้องกับวงโคจร (ORBITAL ELEMENT) ซึ่งประกอบด้วยชุดของตัวเลข (SET OF NUMBER) ที่เราสามารถป้อนเข้าป้อนอุปกรณ์ติดตาม (TRACKING DEVICE) สำหรับดาวเทียมส่วนใหญ่นั้น จะต้องทำการปรับปรุงข่าวสารวงโคจรของดาวเทียม ทุกๆ 1-2 เดือน อย่างไรก็ตามสำหรับดาวเทียมวงโคจรต่ำ เช่น ชัทเทิล (SHUTTLE) และ เมียร์ (MIR) ต้องการปรับปรุงข่าวสารทุก ๆ สัปดาห์

จุดที่ดาวเทียมโคจรผ่าน (THE SUBSATELLITE POINT:SSP) เป็นจุดบนพื้นผิวโลกที่อยู่ต่ำกว่าดาวเทียมโดยตรง เป็นจุดที่แสดงถึงตำแหน่งของดาวเทียม ณขณะเวลาใด ๆ สำหรับดาวเทียมส่วนใหญ่ยกเว้น ดาวเทียมวงโคจรอยู่กับที่ (GEOSTATIONARY SATELLITE) ตำแหน่งจุดที่ดาวเทียมโคจรผ่านนั้นจะเปลี่ยนแปลงอย่างคงที่ ณขณะที่ดาวเทียมเคลื่อนที่ผ่านอวกาศ ถ้าเราสังเกตจุดที่ดาวเทียมโคจรผ่าน ณขณะที่ดาวเทียมเคลื่อนที่ผ่านในอวกาศ จุดที่ดาวเทียมโคจรผ่านดังกล่าวจะเป็นเสมือนเส้นโค้งบนพื้นผิวของโลก เราเรียกว่า เส้นทางที่เฝ้าติดตามดาวเทียม (GROUND TRACKING) หรือ เส้นทางที่ดาวเทียมโคจรผ่าน (SUBSATELLITE PATH) ดังรูปที่ 2.1.2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวอย่างของดาวเทียมที่มีวงโคจรต่ำแสดงดังรูปที่ 2.1.1



รูปที่ 2.1.1 ดาวเทียมวงโคจรต่ำ (ดาวเทียม JAS-1b)

ดาวเทียมจะอยู่ในช่วงที่สามารถติดต่อได้ เมื่อ จุดที่ดาวเทียมโคจรผ่านนั้น ได้เคลื่อนที่เข้ามาใกล้กับสถานีภาคพื้นดินและจะอยู่ในช่วงที่ไม่สามารถติดต่อได้ เมื่อ จุดที่ดาวเทียมโคจรผ่านนั้นได้เคลื่อนที่ไกลออกไปจากสถานีภาคพื้นดินในการกำหนดว่า จุดที่ดาวเทียมโคจรผ่านนั้นเข้ามาใกล้หรือไกลออกไปจากสถานีภาคพื้นดินนั้นทำได้โดยการคำนวณหาระยะทางที่สถานีภาคพื้นดินสามารถเริ่มรับสัญญาณได้ (CRITICAL ACQUISITION DISTANCE) รูปที่ 2.1.3 เมื่อจุดที่ดาวเทียม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โคจรผ่านนั้น เคลื่อนที่เข้ามาใกล้กับสถานีภาคพื้นดินมากกว่าระยะทางที่สถานีภาคพื้นดินสามารถเริ่มรับสัญญาณได้ ดาวเทียมจะอยู่ในช่วงที่สถานีภาคพื้นดินสามารถติดต่อได้ แต่ถ้าจุดที่ดาวเทียมโคจรผ่านนั้นไกลออกไปจากระยะทางที่สถานีภาคพื้นดินสามารถเริ่มรับสัญญาณได้ สถานีภาคพื้นดินจะไม่สามารถติดต่อกับดาวเทียมได้



รูปที่ 2.1.2 เส้นทางที่ดาวเทียมโคจรผ่าน
(ในรูปนี้เป็นของดาวเทียมออสการ์ 8)

สมมติว่า สถานีภาคพื้นดินของเราตั้งอยู่บนวงขั้วโลกเหนือ ถ้าเส้นทางที่ดาวเทียมโคจรผ่านนั้น ผ่านเข้ามาภายในวงกลมที่สถานีภาคพื้นดินสามารถเริ่มรับสัญญาณได้ สถานีภาคพื้นดินจะสามารถติดต่อกับดาวเทียมได้ในระหว่างที่ดาวเทียมผ่านเข้ามา เวลาที่สัญญาณเริ่มรับได้ เกิดขึ้นเมื่อจุดที่ดาวเทียมโคจรผ่านเคลื่อนที่เข้ามาภายในวงกลมของระยะทางที่สถานีภาคพื้นดินสามารถเริ่มรับสัญญาณได้ส่วน เวลาที่สัญญาณเริ่มรับไม่ได้ เกิดขึ้นเมื่อจุดที่ดาวเทียมโคจรผ่านเคลื่อนที่ออกไปจากวงกลมของระยะทางที่สถานีภาคพื้นดินสามารถเริ่มรับสัญญาณได้ สำหรับเส้นทางที่ดาวเทียมโคจรผ่านนั้น จะผ่านเส้นศูนย์สูตร 2 ครั้ง จุดที่เส้นทางที่ดาวเทียมโคจรผ่าน และเส้นศูนย์สูตรตัดกันนั้น เรียกว่า นอด (NODE) ซึ่งแอส

ขึ้นดิ่งโนด (ASCENDING NODE) เกิดขึ้น เมื่อจุดที่ดาวเทียมโคจรผ่านนั้น เคลื่อนที่ผ่านเส้นศูนย์สูตรไปทางทิศเหนือและดิ่งขึ้นดิ่งโนด (DECENDING NODE) เกิดขึ้น เมื่อจุดที่ดาวเทียมโคจรผ่านนั้น เคลื่อนที่ผ่านเส้นศูนย์สูตรไปทางทิศใต้ วิธีติดตามดาวเทียมบางวิธีใช้ เวลา และ เส้นแวงของ แอส เซ็นดิ่งโนดสำหรับ เป็นข้อมูลอ้างอิงวงโคจรโดยปกติแล้ว แอส เซ็นดิ่งโนดใช้ตัวย่อ อีคิว เอ็กซ์ (EQUATOR CROSSING : EQX)



รูปที่ 2.1.3 วงกลมของระยะทางที่สถานีภาคพื้นดินสามารถ เริ่มรับสัญญาณได้
(ในรูปนี้เป็นดาวเทียมออสการ์ 8 วาดรอบ ๆ วอชิงตันดีซี)

จำนวนของ เวลาที่ดาวเทียมใช้ในการโคจรรอบ 1 รอบ เรียกว่า คาบ เวลาของดาวเทียม ซึ่งแอมแซท อยู่ในช่วงประมาณ 98-120 นาที สถานีอวกาศ เมียร์ ซึ่งต่ำกว่า ยูเอ-11 (UO-11) มีคาบเวลา 90 นาที ถ้าคาบเวลาของดาวเทียมนั้น เราสามารถรู้ได้ เราก็จะสามารถคำนวณหาจำนวนของวงโคจรดาวเทียม ต่อวัน ซึ่งหาได้โดยการหารจำนวนของนาทีก่อนหนึ่งวัน คือ 1440 นาที ด้วยคาบ เวลาของดาวเทียม เป็นนาที จำนวนวงโคจรดาวเทียมต่อวันนั้น เรียกว่า การเคลื่อนที่เฉลี่ย (MEAN MOTION) ดาวเทียมเฟส 2 เช่น ยูเอสซีทีดีล (USCISCTIDIL) เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

(ในรูปนี้เป็นของ ดาวเทียมออสการ์ 8 วาตรอบ ๆ วอชิงตันดีซี)

การใช้เครื่องคอมพิวเตอร์ในการติดตามดาวเทียม เป็นวิธีที่สะดวกที่สุด โดยการใช้โปรแกรมติดตามดาวเทียม (TRACKING SOFTWARE) โปรแกรมติดตามดาวเทียมจะบอกถึง เวลาว่า เมื่อใดดาวเทียมดวงที่ต้องการติดต่อนั้นจะอยู่ในช่วงที่สามารถติดต่อได้ ข้อมูลเกี่ยวกับทิศทางของสายอากาศว่าสายอากาศควรจะต้องชี้ไป ณ ทิศทางใด และข้อมูลอื่น ๆ เช่น ระยะทางระหว่างสถานีภาคพื้นดินและดาวเทียม ความสูงของดาวเทียม ฯลฯ

ในการใช้โปรแกรมติดตามดาวเทียม เราจะต้องป้อนข่าวสารเกี่ยวกับวงโคจรของดาวเทียมแต่ละดวงที่เราต้องการติดต่อ ข่าวสารดังกล่าว เป็นชุดของตัวเลขต่าง ๆ ซึ่งเราเรียกว่า ค่าตัวแปรต่าง ๆ ที่เกี่ยวข้องกับวงโคจร (ORBITAL ELEMENT) เนื่องจากดาวเทียมแต่ละดวง จะมีลักษณะวงโคจร ขนาด มุมของระนาบวงโคจรที่ทำกับระนาบเส้นศูนย์สูตร ฯลฯ ที่แตกต่างกัน ดังนั้นดาวเทียมแต่ละดวงจะมีชุดของค่าตัวแปรต่าง ๆ ที่เกี่ยวข้องกับวงโคจรแตกต่างกัน ค่าของตัวแปรต่าง ๆ ที่เกี่ยวข้องกับวงโคจร แสดงดังรูปที่ 2.1.5

Parameter name	Value	Units
Satellite: ¹	OSCAR 13	
Catalog Number:	19216	
Epoch Year:	1988	
Epoch Time:	258.28144	days
Element Set:	RUH9-88	
Inclination:	57.57	deg
RAAN: ²	239.56	deg
Eccentricity:	0.6563	
Arg of Perigee: ³	190.53	deg
Mean Anomaly:	0.0	deg
Mean Motion:	2.09599369	rev/day
Decay Rate: ⁴	0.	rev/day/day
Epoch Rev: ⁵	193	
Semi-Major axis:	25783	km
Bahn Latitude ⁶	0	deg
Bahn Longitude ⁷	180	deg

Alternate parameter names

¹Object

²Right Ascension of Ascending Node; RA of Node

³Argument of Perigee

⁴Drag factor; Rate of change of mean motion, first derivative of mean motion

⁵Revolution number, orbit number

⁶ALAT, BLAT

⁷ALON, BLON

รูปที่ 2.1.5 ค่าตัวแปรต่าง ๆ ที่เกี่ยวข้องกับวงโคจร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



บรรทัดแรกบอกถึง ชื่อดาวเทียม ๗ ที่นี้ ดาวเทียม ชื่อ ออสการ์ 13 (OSCAR 13)

บรรทัดที่สอง บอกถึงหมายเลขดาวเทียม (CATALOG NUMBER) ซึ่งเป็นหมายเลขที่กำหนดโดยนาซา (NASA)

บรรทัดที่สามและสี่ เป็นเวลาที่ค่าตัวแปรต่าง ๆ ตามตาราง ได้ถูกทำการคำนวณ บรรทัดที่ห้าเป็นอีเลเมนต์เซต (ELEMENTSET) ซึ่งบอกถึงแหล่งของข่าวสารที่ได้ทำการคำนวณ เช่น RUH9-88 หมายถึงข้อมูลจาก JIM MILLER (G3RUH) ๗ เดือนกันยายน ปี 1988

บรรทัดอื่นๆ เป็นค่าของตัวแปรที่เกี่ยวข้องกับวงโคจรของดาวเทียมมีดังนี้คือ

เอพ็อค ๗ทม์ (EPOCH TIME: EPOCH) เป็นเวลาอ้างอิงที่ค่าตัวแปรต่างๆของวงโคจรของดาวเทียมได้ถูกคำนวณขึ้น

อินคลิเนชัน (INCLINATION) เป็นมุมที่บอกถึงการจิดเรียงของระนาบวงโคจรของดาวเทียม เมื่อเทียบกับระนาบเส้นศูนย์สูตรของโลก

ไรท์แอสเซนชัน ออฟ แอสเซนดิง นอด (RIGHT ASCENSION OF - ASCENDING NODE: RAAN) เป็นมุมที่บอกถึงการจิดเรียงของระนาบวงโคจรดาวเทียม เมื่อเทียบกับดวงดาวที่คงที่ (FIXED STAR)

เอ็คเซนทริซิตี (ECCENTRICITY) เป็นพารามิเตอร์ที่สำคัญสำหรับบรรยายถึง ลักษณะวงรี ที่ทำให้เกิดวงโคจรของดาวเทียม

อาร์กิวเมนต์ ออฟ เพริจี (ARGUMENT OF PERIGEE) เป็นมุมที่แสดงถึงตำแหน่งที่ใกล้ที่สุดของดาวเทียม เมื่อค่ามุมอยู่ระหว่าง 0 ถึง 180 องศา ตำแหน่งที่ใกล้ที่สุดของดาวเทียมอยู่ที่ซีกโลกเหนือ เมื่อมุมอยู่ระหว่าง 180 ถึง 360 องศา ตำแหน่งที่ใกล้ที่สุดของดาวเทียมอยู่ที่ซีกโลกใต้

มีน อะนอมอะลี (MEAN ANOMALY) เป็นตัวเลขระหว่าง 0 และ 256 ที่เพิ่มขึ้นอย่างสม่ำเสมอตามเวลา ถูกใช้หาตำแหน่งของดาวเทียมบนวงโคจรวงรี เมื่อ มีน อะนอมอะลี เป็น 0 หรือ 256 ดาวเทียมอยู่ใกล้กับโลกมากที่สุด (PERIGEE) เมื่อ มีน อะนอมอะลี เป็น 128 ดาวเทียมอยู่ไกลจากโลกมากที่สุด (APOGEE) เมื่อมีนอะนอมอะลีอยู่ ระหว่าง 0 และ 128 ดาวเทียมนั้นจะหันหัว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาค้นคว้าเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มี 034953

ขึ้นไปทางจุดไกลจากโลกมากที่สุด เมื่อมีนอระนอมจะลืออยู่ระหว่าง 128 และ 256 ดาว เทียมนั้นจะหันหัวลงไปทางจุดใกล้กับโลกมากที่สุด

มีน มชัน (MEAN MOTION) เป็นจำนวนของการครบรอบหนึ่งรอบ จากเพริจีไปยัง เพริจีในวัน หรือในเวลา 1440 นาที

ดีเคย์ เรท (DECAY RATE) เป็นอัตราการเปลี่ยนแปลงของการเคลื่อนที่เฉลี่ย เป็นพารามิเตอร์ที่บอกถึงว่า ขึ้นบรรยากาศมีผลกระทบต่อการเคลื่อนที่ของดาว เทียมอย่างไร

เอพ็อก รีวอลูชัน (EPOCH REVOLUTION) เป็นจำนวนรอบของดาว เทียมที่โคจร

เซมิ เมเจอร์ เอ็กซิส (SEMI-MAJOR AXIS) เป็นค่าความยาวครึ่งหนึ่งของแกนด้านที่ยาวที่สุดของวงรี ใช้สำหรับบอกลักษณะของวงโคจรวงรี

เมื่อเราเริ่มใช้โปรแกรมติดตามดาว เทียม โปรแกรมดังกล่าวจะให้ผลลัพธ์ของดาว เทียมดวงที่เราต้องการติดต่อรูปที่ 2.1.6 ซึ่งแสดงผลลัพธ์ของโปรแกรมติดตามดาว เทียมของดาว เทียม ออสการ์ 13 ในวันที่ 23 ธันวาคม 2531

จากรูป จะเป็นข้อมูลจากโปรแกรมติดตามดาว เทียม ทุก ๆ 15 นาที (TIME STEP) ในคอลัมน์ที่หนึ่งถึงสามเป็นข้อมูลเกี่ยวกับ เวลา มุมอาซิมุท และ มุมอีเลเวชัน ของดาว เทียม ออสการ์ 13 จะอยู่ในช่วงเวลาที่เราสามารถติดต่อได้ในระหว่าง เวลา 1145 และ 2115 ยูทีซี (UTC) และจะอยู่ในช่วงเวลาที่สามารติดต่อได้ประมาณ 9.5 ชั่วโมง จะสังเกตได้ว่าในระหว่าง 1415 และ 2015 เป็นเวลารวมเป็นเวลา 6 ชั่วโมง ที่มุมอีเลเวชันจะอยู่คงที่ ระหว่าง 45 และ 55 องศา ในช่วงเวลานี้ นักวิทยุสมัครเล่นสามารถที่จะตั้งสายอากาศไปที่มุมอีเลเวชัน 50 องศา โดยไม่ต้องควบคุมมุมอีเลเวชันก็ได้ แต่ถ้าเป็นสายอากาศที่มีบีมวิทแคบ (NARROW BEAMWIDTH ANTENNA) ก็ต้องทำการปรับค่ามุมอีเลเวชัน

ในคอลัมน์ที่สี่ เป็นข้อมูลเกี่ยวกับดอปเปอร์ชิฟ (DOPPLER SHIFT) ที่ เวลา 1200 ยูทีซี สัญญาณที่ผ่านทอมตบี ทรานสพอนเดอร์จะมีความถี่ต่ำกว่าที่ทานายาว 1,867 เฮิรตซ์ เป็นเพราะว่า อัลกอริทึมที่เขียนในคอมพิวเตอร์ ดอปเปอร์ชิฟจึงไม่มีในเวลา 1145 ยูทีซี ซึ่งเป็นเวลาที่ดาว เทียม เริ่มผ่านเข้ามา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการศึกษาเท่านั้น เมื่ออยู่ภายใต้เงื่อนไขข้อควรระวังในการใช้

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ครึ่งแรก

คอแลมที่ห่า เป็นระยะทางระหว่างดาวเทียม และสถานีภาคพื้นดิน

คอแลมที่หก เป็นความสูงของดาวเทียมที่สูงจากพื้นโลก สังเกตจุดที่มี

ความสูงมากที่สุดเรียกว่า อะพอจี (APOGEE) เกิดขึ้นที่เวลา 1600 ยุกีซี คอแลม

ที่เจ็ดและแปด เป็นค่าละติจูด และลองจิจูดของ เส้นทางที่ดาวเทียมโคจรผ่าน

AMSAT-OSCAR 13

Ground Station: Lat = 39°N, Long = 77°W, Ht = 0 km
Minimum Elevation = 0 degrees

Day # 358 - - - Friday, December 23 - - - 1988

UTC HHMM	AZ deg	EL deg	DOPPLER Hz	RANGE km	HEIGHT km	LAT deg	LONG deg	PHASE < 256 >
1145	167	5	—	23325	18353	-29.8	62.7	34
1200	166	11	-1867	25068	20664	-25.2	62.5	40
1215	165	16	-1733	26686	22773	-21.1	63.1	45
1230	166	21	-1596	28176	24694	-17.5	64.2	51
1245	166	26	-1462	29541	26438	-14.3	65.7	57
1300	168	30	-1332	30784	28017	-11.4	67.5	62
1315	170	33	-1206	31910	29438	-8.7	69.5	68
1330	173	37	-1083	32922	30711	-6.2	71.6	73
1345	175	40	-966	33823	31843	-3.9	73.9	79
1400	179	42	-851	34617	32838	-1.6	76.2	85
1415	183	45	-739	35308	33701	0.5	78.7	90
1430	187	47	-630	35896	34437	2.5	81.1	96
1445	192	49	-523	36384	35049	4.5	83.6	101
1500	197	51	-416	36773	35540	6.4	86.2	107
1515	202	52	-311	37063	35911	8.3	88.7	113
1530	208	53	-206	37255	36164	10.1	91.3	118
1545	213	54	-100	37348	36300	11.9	93.8	124
1600	219	55	7	37341	36320	13.7	96.4	129
1615	225	55	116	37233	36224	15.5	98.9	135
1630	231	55	226	37022	36011	17.3	101.4	141
1645	237	54	340	36705	35680	19.2	103.8	146
1700	243	54	456	36279	35231	21.0	106.2	152
1715	248	53	577	35741	34661	22.9	108.5	157
1730	254	53	703	35084	33967	24.8	110.8	163
1745	259	52	835	34305	33148	26.8	112.9	168
1800	264	51	973	33396	32198	28.9	114.9	174
1815	269	51	1119	32351	31114	31.0	116.7	180
1830	273	50	1276	31160	29890	33.3	118.3	185
1845	278	49	1442	29813	28520	35.6	119.6	191
1900	283	49	1621	28300	26996	38.1	120.5	196
1915	287	49	1815	26606	25310	40.8	120.9	202
1930	293	49	2027	24714	23451	43.7	120.6	208
1945	298	50	2258	22606	21407	46.8	119.2	213
2000	305	51	2511	20262	19166	50.0	116.2	219
2015	313	54	2784	17663	16715	53.3	110.6	224
2030	326	57	3066	14801	14045	56.3	100.8	230
2045	348	60	3300	11720	11160	57.5	84.5	236
2100	30	58	3243	8692	8115	53.9	61.2	241
2115	75	30	1933	6887	5126	39.9	36.4	247

รูปที่ 2.1.6 ผลลัพธ์จากโปรแกรมติดตามดาวเทียม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คอลัมน์ที่เก้า เป็นเฟส ซึ่ง เป็นพารามิเตอร์ที่ใช้กับดาวเทียมเฟสสาม เช่น ออสการ์ 13 ซึ่งเป็นดาวเทียมที่มีทรานสปอนเดอร์ที่ไม่สามารถทำงานพร้อมกันได้อยู่หลายอัน การควบคุมจึงต้องดูจากตารางการทำงาน เช่น

off: จาก MA 0 ถึง 49

MODE B: จาก MA 50 ถึง MA 159

MODE JL: จาก MA 160 ถึง MA 199

MODE B: จาก MA 200 ถึง MA 255

2.2 การสื่อสารข้อมูลแบบอนุกรม

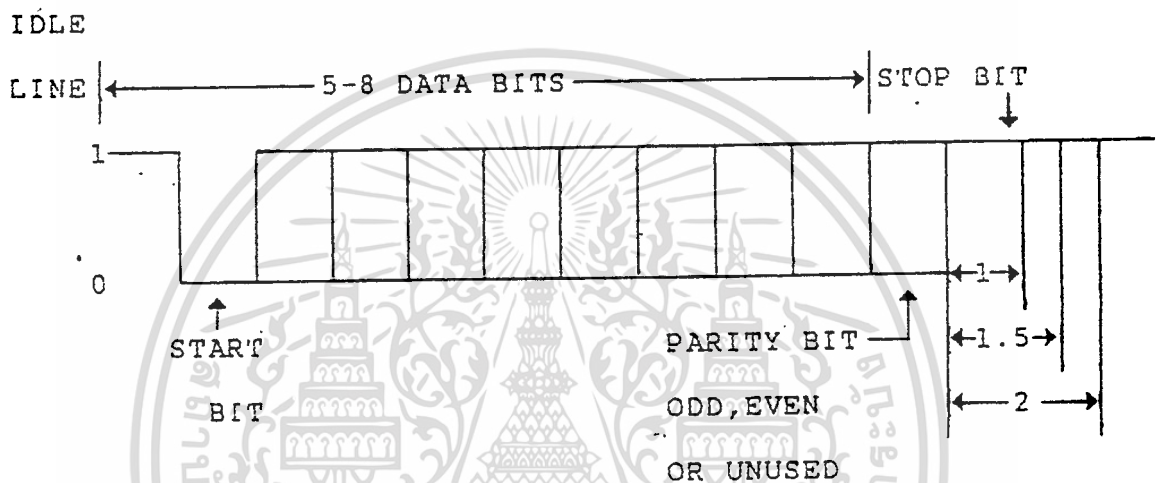
ในการสื่อสารข้อมูลแบบอนุกรมแบบนี้ ประกอบด้วย บิตเริ่มต้น หรือบิตสตาร์ท (Start bit) และบิตสิ้นสุด หรือบิตสต็อป (Stop bit) รูปที่ 2.2.1

ในขณะที่ไม่มีการส่งข้อมูล ในสายสื่อสาร (LINE) จะมีแรงดันตลอดเวลา เพื่อทำให้เกิดความแน่ใจว่าฝ่ายรับยังติดต่อกับฝ่ายส่ง ฝ่ายส่งจะเริ่มส่งข้อมูลบอกจุดเริ่มต้น สัญญาณของอะซิงโครนัสจะเป็น "0" หนึ่งช่วงของสัญญาณนาฬิกา บิตนี้เรียกว่าบิตสตาร์ท ข้อมูลหนึ่งตัวอักษรที่ตามหลังบิตสตาร์ทจะมีขนาดตั้งแต่ 5 บิต จนถึง 8 บิต โดยบิตที่มีความสำคัญน้อยที่สุด (LSB) จะถูกส่งออกมาก่อน และส่งต่อไปจนถึงบิตที่มีความสำคัญมากที่สุด (MSB) จากนั้นส่วนที่ตามหลังข้อมูลก็จะเป็น บิตพาริตี ซึ่งอาจจะใช้ หรือไม่ใช้ก็ได้ บิตพาริตีจะทำหน้าที่เป็นตัวตรวจสอบความถูกต้องของสัญญาณที่ได้รับ บิตพาริตีอาจจะจะเป็น คู่ (Even) หรือคี่ (Odd) ก็ได้ หมายความว่าถ้าหากเป็นพาริตีคู่จำนวนบิตที่เป็นหนึ่งในช่วงบิตข้อมูลกับบิตพาริตีรวมกันแล้วจะต้อง เป็นเลขคู่ ผู้ส่งข้อมูลจะทำการที่ตรวจสอบข้อมูลแล้วใส่บิตพาริตีเอง ฝ่ายรับเมื่อรับสัญญาณแล้ว ก็จะต้องตรวจสอบดูว่าพาริตีเป็นจริงตามสถานะที่ตั้งเอาไว้หรือไม่ หากผิดพลาดก็หมายความว่า สัญญาณที่รับได้นั้นผิดพลาดไปจากสถานะที่ส่งที่ส่งออกมา ทั้งนี้จะต้องผิดเป็นจำนวนคี่ เท่านั้นจึงจะตรวจสอบได้ว่าเกิดความผิดพลาด หลังจากบิตพาริตีแล้วก็ต้องมีบิตสต็อปซึ่งเป็น "1" ความกว้างของบิตสต็อปอาจจะ เป็น 1, 1.5 หรือ 2 พัลส์ของสัญญาณนาฬิกา ซึ่งแล้วแต่ผู้รับ และผู้ส่งจะตกลงใช้กันเอง

การเริ่มใช้พอร์ทอนุกรมจึงจำเป็นต้องตั้งค่าพารามิเตอร์ต่างๆ สำหรับเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การสื่อสารซึ่งมีดังต่อไปนี้ คือ

1. ความเร็วของการส่ง
2. ความยาวของรหัส 1 อักขร
3. บิตตรวจสอบ
4. จำนวนบิตตีกลับ



รูปที่ 2.2.1 รูปแบบของการสื่อสารข้อมูลแบบอะซิงโครนัส

8251 เป็นชิพที่กำหนดหน้าที่เป็นตัวรับ และส่งข้อมูล ระหว่างไมโครโปรเซสเซอร์ กับอุปกรณ์อินพุตและเอาต์พุตแบบอนุกรม ชิปนี้ทำงานได้ทั้งโหมดอะซิงโครนัสและโหมดซิงโครนัส มีโครงสร้างภายในดัง รูปที่ 2.2.2

กลุ่มของสัญญาณที่ติดต่อกับซีพียู

- D0-D7 เป็นบัสข้อมูลที่ใช้ในการติดต่อกับซีพียู
- Reset เป็นอินพุตแอกทีฟ 1 เมื่อถูกรีเซ็ตจะเป็นการรีเซ็ตคำสั่งควบคุมทั้งหมด
- CLK เป็นสัญญาณนาฬิกาที่ใช้ในการควบคุมภายใน จะต้องมีความเร็วเกิน 30 เท่าของความเร็วในการรับส่งข้อมูล

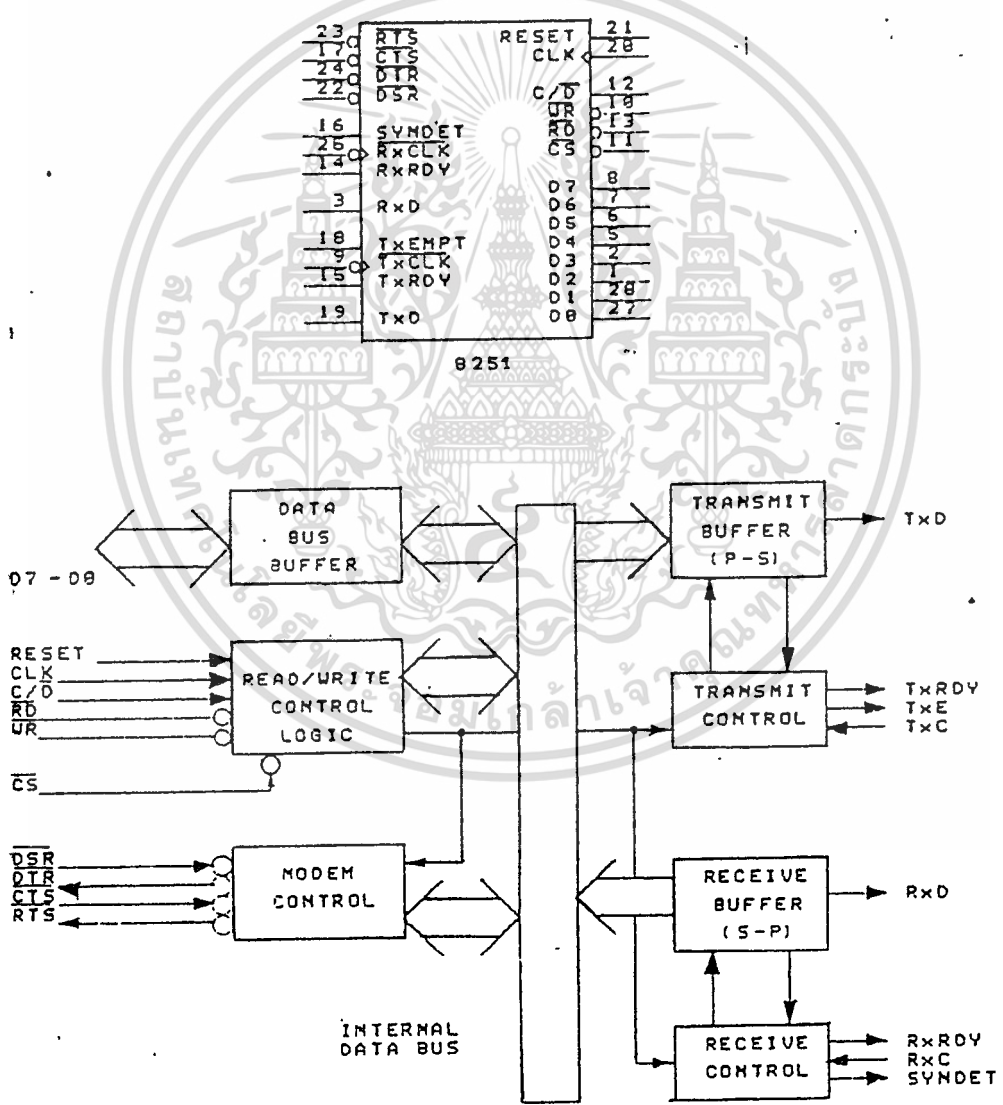
เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้สำหรับใช้ในวงจำกัดเท่านั้น ไม่สามารถเผยแพร่ในที่สาธารณะได้โดยไม่ได้รับอนุญาต
RD เป็นอินพุตแอกทีฟ 0 ต่อไปยังขา RD ของซีพียู จะแอคทีฟเมื่ออ่านค่าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

มูลจาก 8251ไปเก็บ

$\overline{WR}$ เป็นอินพุตแอกทีฟ 0 ต่อไปยังขา $\overline{WR}$ ของชิพๆ จะแอกทีฟ เมื่อเขียนข้อมูลไปยัง 8251

$C/\overline{D}$ เป็นอินพุตเมื่อได้รับลอจิก1 จะอยู่ในช่วงการติดต่อกับ รีจิสเตอร์ควบคุม และถ้าได้รับลอจิก 0 จะอยู่ในช่วงการติดต่อกับรีจิสเตอร์ข้อมูล (DATA REGISTER)

$\overline{CS}$ เป็นอินพุตแอกทีฟ 0 จะอยู่ในช่วงที่ยอมมาให้ชิพ 8251 นี้ติดต่อกับชิพๆได้



รูปที่ 2.2.2 โครงสร้างภายในและการจัดขาของไอซี 8251

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กลุ่มของสัญญาณที่ติดต่อกับแรม เดิม

- RTS** เป็นขาเอาต์พุตแอกทีฟ 0 เมื่อต้องการส่งข้อมูลไปยังแรม เดิม
- CTS** เป็นขาเอาต์พุตแอกทีฟ 0 จะทำให้ 8251 ส่งข้อมูลออกทาง TXDทันที ขานี้ถ้าไม่ได้เป็นลอจิก 0 จะส่งข้อมูลออก
- DTR** เป็นขาเอาต์พุตแอกทีฟ 0 แสดงถึงความพร้อมของ 8251 ที่จะติดต่อกับแรม เดิม สามารถใช้ได้ โดยการโปรแกรมชุดคำสั่ง (Command instruction)
- DSR** เป็นขาอินพุตแอกทีฟ 0 จากแรม เดิมส่งมา เมื่อแรม เดิมพร้อมที่จะติดต่อกับ 8251 เราสามารถทดสอบสัญญาณนี้ โดยการอ่านรีจิสเตอร์สถานะ (STATUS REGISTER)

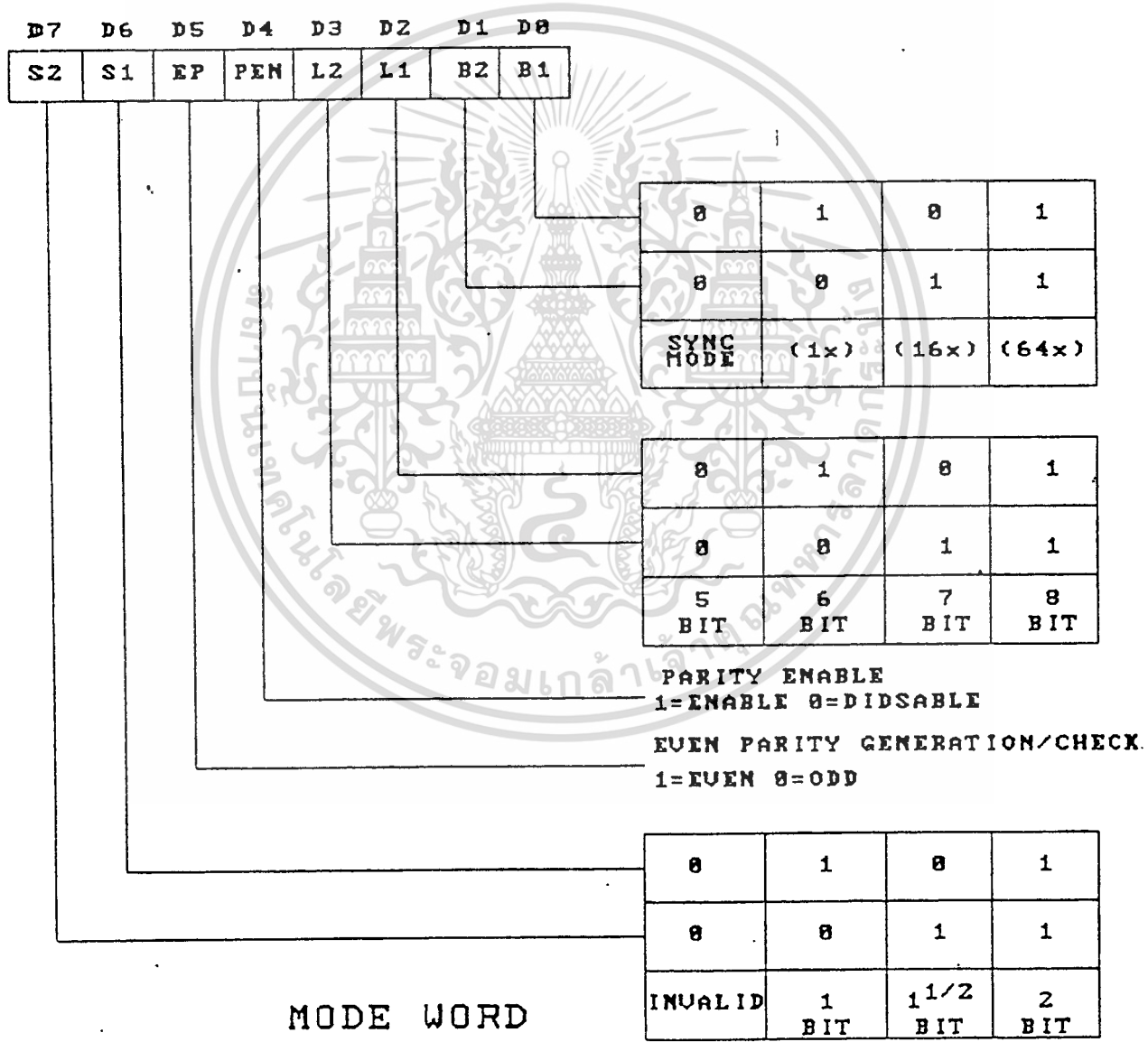
กลุ่มของสัญญาณที่แสดงสถานะ

- TxRDY** เป็นขาเอาต์พุตแอกทีฟ 1 เมื่อ 8251 พร้อมที่จะรับข้อมูลจากซีพียู เพื่อส่งออกที่ขา TXD และจะรีเซ็ตตัวเอง เมื่อขา $\overline{WR}$ เปลี่ยนจาก Low เป็น High ในช่วงที่ซีพียูส่งข้อมูลมา 8251 และสถานะของ TxRDY นี้ยังปรากฏที่รีจิสเตอร์สถานะที่บิต D₀ อีกด้วย ดังนั้นก่อนที่จะส่งข้อมูลใหม่มายังบัฟเฟอร์ส่ง ซีพียูจะต้องตรวจสอบบิต D₀ ของรีจิสเตอร์สถานะนี้ก่อน ถ้าเป็น 0 ให้อ่านค่าเป็น 1 ให้ส่งข้อมูลมาได้
- RxRDY** เป็นขาเอาต์พุตแอกทีฟ 1 เมื่อ 8251 รับข้อมูลเข้าทางขา RXD เสร็จ 1 อักขรแล้วลอจิกนี้ จะปรากฏอยู่ที่รีจิสเตอร์สถานะอีกด้วยคือ ที่บิต D₁ ดังนั้นซีพียูต้องคอยตรวจสอบที่บิตนี้ ถ้าเป็น 1 ให้อ่านข้อมูลจาก 8251 เข้าไปเก็บในซีพียูได้
- TxE** เป็นขาเอาต์พุตแอกทีฟ 1 เมื่อข้อมูลถูกส่งออกจาก 8251 หมดแล้ว และจะเป็นลอจิก 0 เมื่อซีพียู เริ่มส่งข้อมูลมาใหม่

กลุ่มของสัญญาณที่ติดต่อกับแรม เดิม

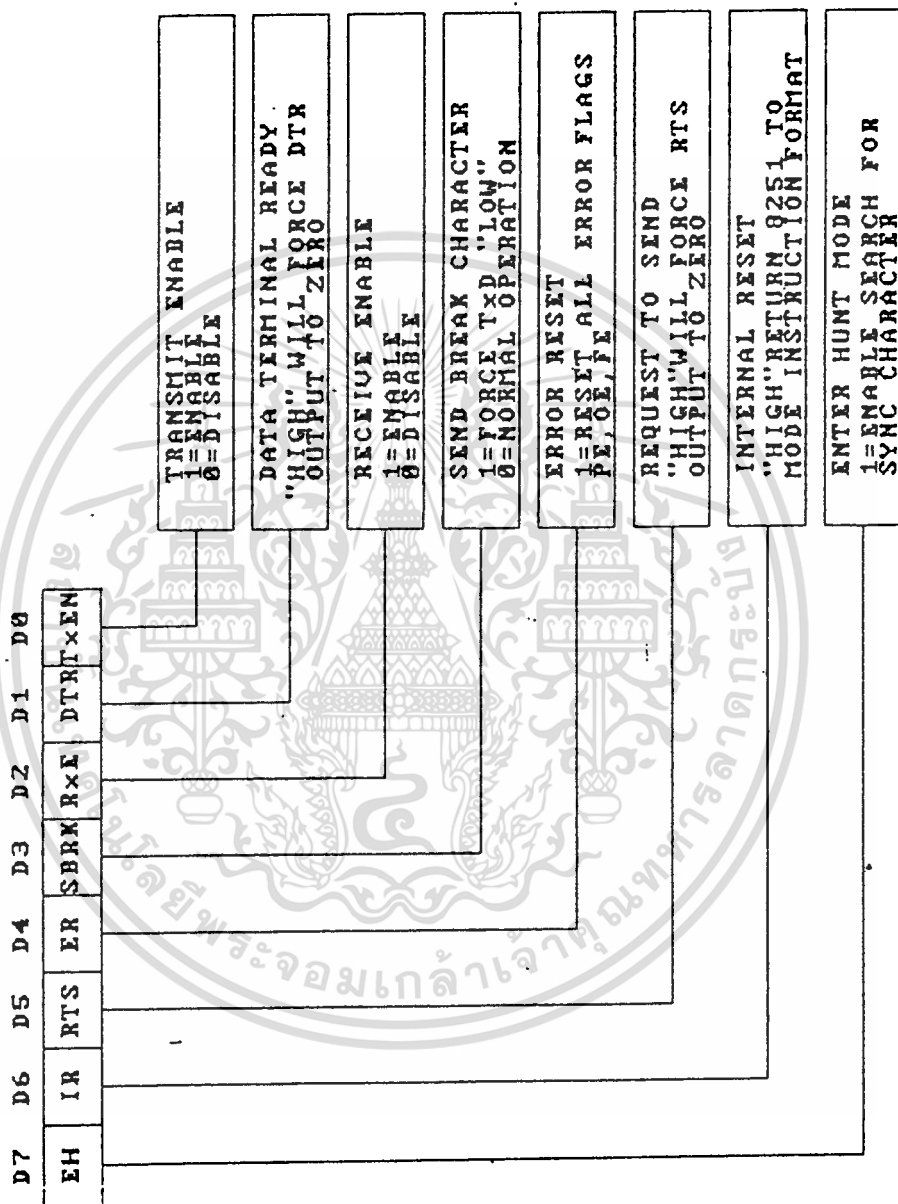
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- $\overline{\text{RTS}}$ เป็นขาเอาต์พุตแอกทีฟ 0 เมื่อต้องการส่งข้อมูลไปยังโมเด็ม
- $\overline{\text{CTS}}$ เป็นขาเอาต์พุตแอกทีฟ 0 จะทำให้ 8251 ส่งข้อมูลออกทาง TXDทันที ขานี้ถ้าไม่ได้เป็นลอจิก 0 จะส่งข้อมูลออก
- $\overline{\text{DTR}}$ เป็นขาเอาต์พุตแอกทีฟ 0 แสดงถึงความพร้อมของ 8251 ที่จะติดต่อกับโมเด็ม สามารถใช้ได้ โดยการโปรแกรมชุดคำสั่ง
- $\overline{\text{DSR}}$ เป็นขาอินพุตแอกทีฟ 0 จากโมเด็มส่งมา เมื่อโมเด็มพร้อมที่จะติดต่อกับ 8251 เราสามารถทดสอบสัญญาณนี้โดยการอ่านรีจิสเตอร์สถานะ



รูปที่ 2.2.3.MODE INSTRUCTION FORMAT, ASYNCHRONOUS MODE

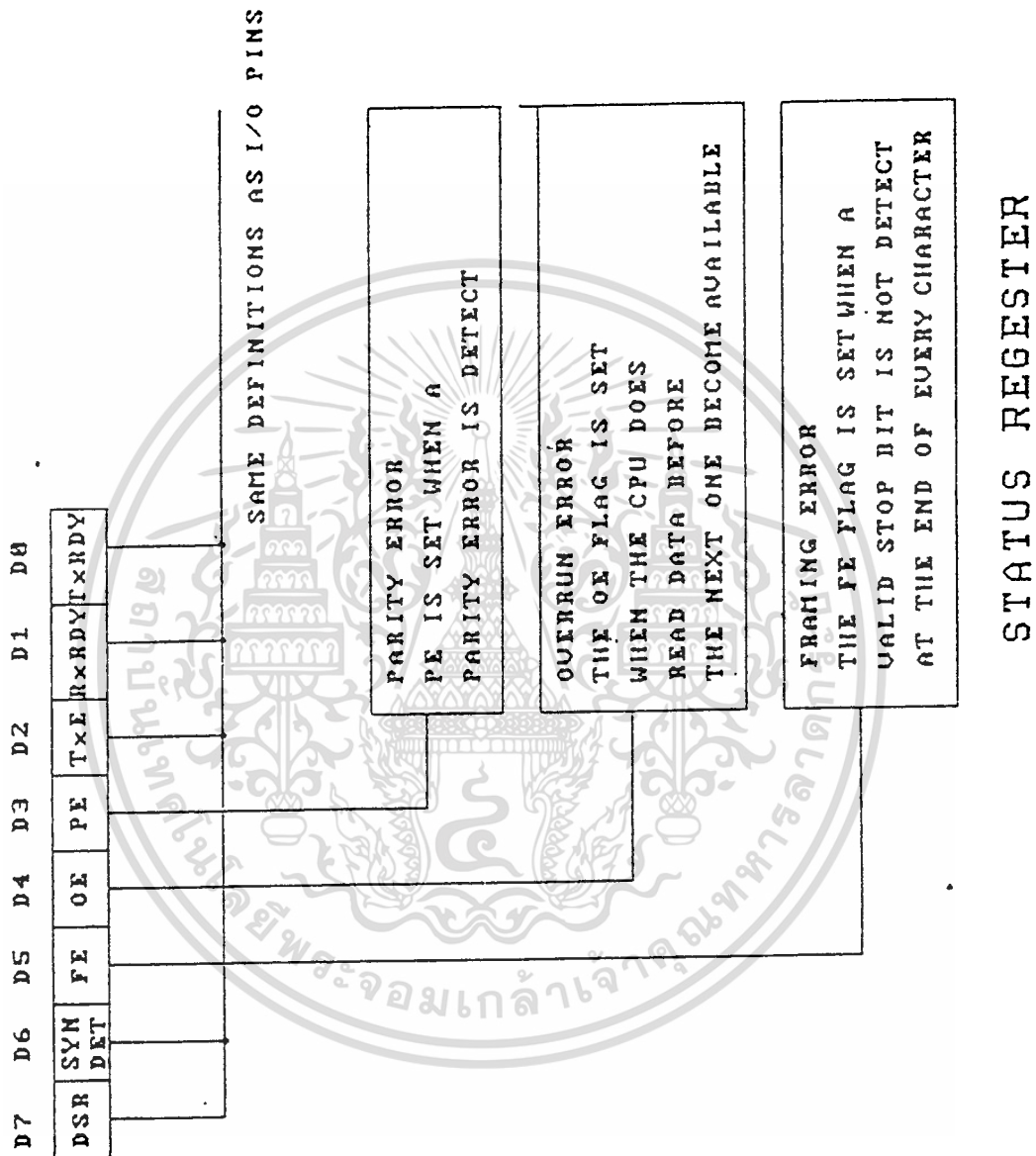
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



COMMAND WORD

รูปที่ 2.2.4 COMMAND INSTRUCTION FORMAT

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



2.3 ดอทแมทริกซ์แอลซีดีโมดูล (DOT MATRIX LCD MODULE)

เป็นอุปกรณ์แสดงผลแบ่งได้เป็น 3 แบบดังนี้

2.3.1 โมดูลอักษร (CHARACTER LCD MODULE)

2.3.2 โมดูลรูปภาพ (GRAPHIC LCD MODULE)

2.3.3 โมดูลตัวเลข (SEGMENT DISPLAY TYPE LCD MODULE)

โดยในแต่ละแบบ จะมีส่วนประกอบใหญ่ๆ แบ่งได้เป็น

ดอทแมทริกซ์แอลซีดี (DOT MATRIX LCD) เป็นตัวแสดงให้เรา
มองเห็นลักษณะ การปิด และ เปิดตัว เองกับแสงก็คือ ส่วนของที่เป็นตัวกระจก
บรรจุผลึก

ไดรเวอร์ (DRIVER) เป็นตัวรับสัญญาณจากตัวควบคุมมาขับดัน
LCD อีกทีหนึ่ง โดยมี เบอร์ที่นิยมมาใช้นแอลซีดีโมดูล (LCD MODULE) เช่น HD44100H,
MSM5259

คอนโทรลเลอร์ (CONTROLLER) เป็นตัวรับข้อมูลจากอุปกรณ์
ภายนอกมา และจัดการควบคุมแอลซีดีโมดูลให้ทำงานแสดงผลต่าง ๆ เช่น การ
ลบจอภาพ การเกิดตัวอักษร เป็นต้น โดยในแบบโมดูลตัวอักษรจะนิยมมาใช้ เบอร์
HD44780 ส่วนในแบบโมดูลรูปภาพจะนิยมมาใช้ เบอร์ HD61830

รีจิสเตอร์ซีเลคชั่น (REGISTER SELECTION:RS) เป็นขาที่ใช้
เลือกรีจิสเตอร์ (REGISTER) ภายในคอนโทรลเลอร์ ซึ่งมี 2 ตัว คือ อินสตรัคชั่น
รีจิสเตอร์ (INSTRUCTION REGISTER :IR) และดาต้ารีจิสเตอร์ (DATA
REGISTER:DR) โดยถ้าเป็น 1 จะเป็นการเลือกข้อมูล (DATA) และถ้าเป็น 0 จะ
เป็นการเลือกคำสั่ง (INSTRUCTION)

รีด/ไรท์ (READ/WRITE:R/W) เป็นตัวเลือกว่าจะเขียน หรือ
จะอ่านข้อมูลจากตัวไอซีโดยอ่านข้อมูล เท่ากับ 1 เขียนข้อมูล เท่ากับ 0

อีน่าเบิล (ENABLE SIGNAL:E) เป็นขากำหนดสภาพการรับ
เขียน และอ่านข้อมูล

ขาต่างๆของ แอลซีดีโมดูล แสดงดัง รูปที่ 2.3.1

DB0-DB7 เป็นขารับส่งข้อมูลจากตัวไอซี

VDD เป็นขาเลี้ยงตัววงจร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

VSS เป็นขา กราวด์
VO, เป็นขา รับ แรงดัน เตจ (VOLTAGE) ในการ ขับ แอล ซี ดี ให้ สว่าง หรือ มี ด

NO.	SIGNAL	NO.	SIGNAL
1	VSS	2	VDD
3	VO	4	RS
5	R/W	6	E
7	DB0	8	DB1
9	DB2	10	DB3
11	DB4	12	DB5
13	DB6	14	DB7

รูปที่ 2.3.1 แสดงขาต่าง ๆ ของ แอล ซี ดี ๓๓๓๓

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4 รีลไทม์เคาน์เตอร์ (REALTIME COUNTER)

เป็นตัวนับ และจัดการเกี่ยวกับเวลา แบ่ง เป็นสองหลัก ๆ ละสี่บิตซึ่งสามารถเข้าถึงได้รอบการอ่าน หรือ เขียน แต่ละหลักจะเป็น เลขบีซีดี (BCD) ซึ่งแสดงได้ดังรูปที่ 2.4.1

Real Time Counter Format											
Counter Addressed		Units				Max BCD Code	Tens				Max BCD Code
		D0	D1	D2	D3		D4	D5	D6	D7	
Milliseconds	(00 _h)	—	—	—	—	0	D4	D5	D6	D7	9
Hundredths and Tenths Sec	(01 _h)	D0	D1	D2	D3	9	D4	D5	D6	D7	9
Seconds	(02 _h)	D0	D1	D2	D3	9	D4	D5	D6	—	5
Minutes	(03 _h)	D0	D1	D2	D3	9	D4	D5	D6	—	5
Hours	(04 _h)	D0	D1	D2	D3	9	D4	D5	—	—	2
Day of the Week	(05 _h)	D0	D1	D2	—	7	—	—	—	—	0
Day of the Month	(06 _h)	D0	D1	D2	D3	9	D4	D5	—	—	3
Month	(07 _h)	D0	D1	D2	D3	9	D4	—	—	—	1

(—) indicates unused bits

รูปที่ 2.4.1 ตารางแสดงรูปแบบรีลไทม์เคาน์เตอร์

บิตที่ไม่ใช่จะเป็นลอจิกศูนย์ในช่วงการอ่าน และจะไม่มีผลต่อการอ่าน การเขียน เช่นในหลักสิบของชั่วโมงจะไม่เกินสอง ดังนั้นจึงมีเพียงสองบิตที่หลักสิบของชั่วโมง ส่วนบิตที่เหลือจะไม่ใช้แรม

งานรีลไทม์คล็อกจะมีแรมขนาด 56 บิตอยู่ภายใน เพื่อใช้เก็บเวลา การเตือน(alarm) ที่ผู้ใช้เปรียบเทียบกับรีลไทม์เคาน์เตอร์

งานหลักหน่วยของมิลลิวินาที (milliseconds) และหลักสิบของวันในสัปดาห์(day of week)จะไม่มีแรมทั้งนี้เพราะไม่ใช้ในรีลไทม์เคาน์เตอร์รูปแบบของแรมจะเหมือนกับรีลไทม์เคาน์เตอร์คือ แต่ละหลักมีสี่บิต และมีทั้งหมดสิบสี่หลัก อย่างไรก็ตามมันจะไม่มีบิตที่ไม่ใช่ ดังนั้นบิตที่ไม่ใช่ในรีลไทม์เคาน์เตอร์จะเปรียบเทียบกับศูนย์ในแรม ซึ่งแอดเดรสแรมแสดงได้ดังรูปที่ 2.4.2

อินเตอร์รัฟ และคอมพาราเตอร์ สามารถให้อินเตอร์รัฟเอาต์พุตได้

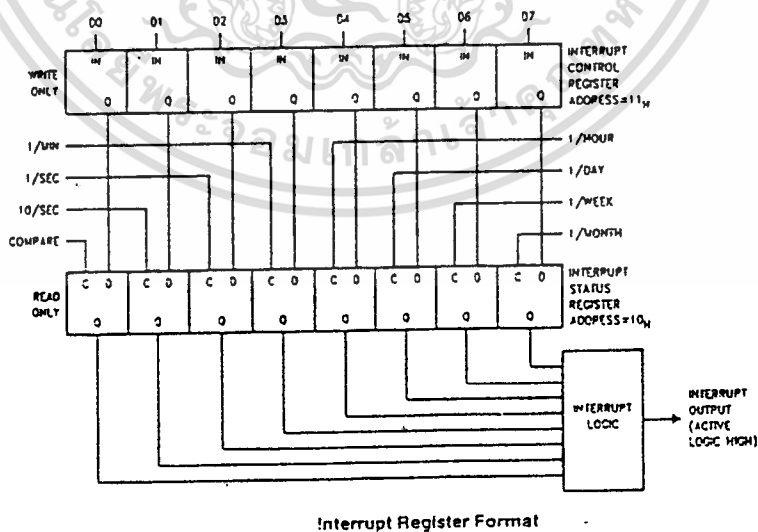
สองแบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Address Codes and Function					Function
A4	A3	A2	A1	A0	
0	0	0	0	0	Counter—Milliseconds
0	0	0	0	1	Counter—Hundredths and Tenths of Seconds
0	0	0	1	0	Counter—Seconds
0	0	0	1	1	Counter—Minutes
0	0	1	0	0	Counter—Hours
0	0	1	0	1	Counter—Day of Week
0	0	1	1	0	Counter—Day of Month
0	0	1	1	1	Counter—Month
0	1	0	0	0	RAM—Milliseconds
0	1	0	0	1	RAM—Hundredths and Tenths of Seconds
0	1	0	1	0	RAM—Seconds
0	1	0	1	1	RAM—Minutes
0	1	1	0	0	RAM—Hours
0	1	1	0	1	RAM—Day of Week
0	1	1	1	0	RAM—Day of Month
0	1	1	1	1	RAM—Months
1	0	0	0	0	Interrupt Status Register
1	0	0	0	1	Interrupt Control Register
1	0	0	1	0	Counters Reset
1	0	0	1	1	RAM Reset
1	0	1	0	0	Status Bit
1	0	1	0	1	GO Command
1	0	1	1	0	STANDBY INTERRUPT
1	1	1	1	1	Test Mode

All others unused

รูปที่ 2.4.2 แอดเดรสแมน



Interrupt Register Format

รูปที่ 2.4.3 รูปแบบอินเตอร์รัพรีจิสเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แบบแรกนั้น อินเตอร์รัฟจะเกิดขึ้นเมื่อเราได้ทำการฮันนา เบิ้ลไว้ และเกิดการ เปรียบเทียบ กับรีลไทม์ เคาน์เตอร์แรม

แบบที่สอง คืออินเตอร์รัฟเอาท์พุทที่สามารถโปรแกรมให้สัญญาณเอาท์พุทแตกต่างจากกัน 8 สัญญาณคือ 10 Hz , 1/sec , 1/min , 1/hour , 1/day , 1/week , 1/month และเมื่อเกิดการเปรียบเทียบกันของแรม และรีลไทม์ เคาน์เตอร์ ในการฮันนา เบิ้ล เอาท์พุททางได้โดยการ เชื่อมกับที่ตรงกับความถี่ เอาท์พุทที่ต้องการ ในอินเตอร์รัฟคอนโทรลลรีจิสเตอร์ดังรูปที่ 2.4.3

2.5 ชุดชิปส เตปปี้งมอ เตอร์

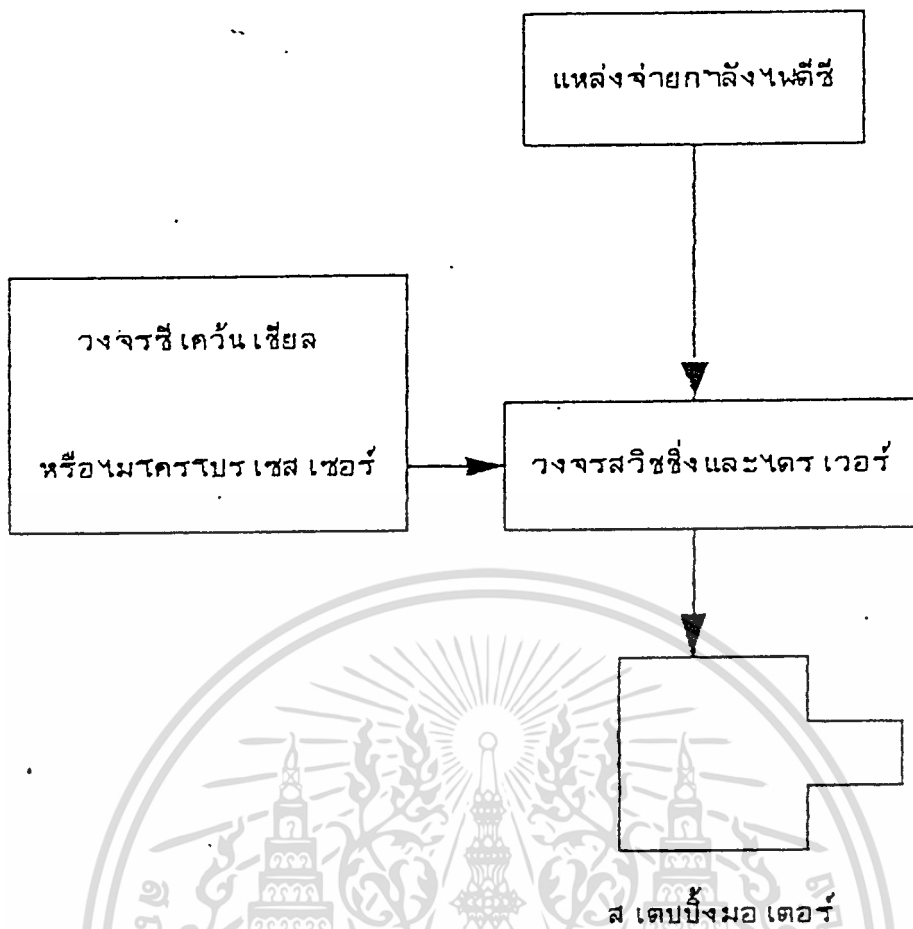
ส เตปปี้งมอ เตอร์ เป็นอุปกรณ์จำพวก เชิงกลทางไฟฟ้าที่มีอินพุทเป็นกลุ่มของไบনারีวอลต์ เดจ และ เอาท์พุทเป็นลักษณะของการ เคลื่อนที่แบบ เชิงมุม หรือ หมุนรอบ เป็นแบบส เตป (แต่ละส เตปอยู่ในช่วง 0.1 ถึง 30 องศา ขึ้นอยู่กับโครงสร้างของส เตปปี้งมอ เตอร์) ตามสัญญาณพัลส์ที่ป้อนให้กับขดลวด เต เตอร์ ซึ่งจะ เกิดแรงผลักดันต่อโรเตอร์ ทำให้โรเตอร์หมุนไป แต่ลักษณะของส เตปปี้งมอ เตอร์จะมีชุดของส เตเตอร์อยู่หลายชุด ซึ่ง เรียกว่า เฟส ฉะนั้นเมื่อป้อนสัญญาณที่เป็นพัลส์ขึ้น ลักษณะซี เควินของ เลขไบনারีโดยผ่านวงจรไดรเวอร์(driver)จะทำให้โรเตอร์ หมุนได้อย่างต่อเนื่องดังบล็อกไดอะแกรมรูปที่ 2.5.1

หลักการกระตุ้นให้กับขดลวดส เตเตอร์ของส เตปปี้งมอ เตอร์ แบ่งออกเป็น 3 แบบ

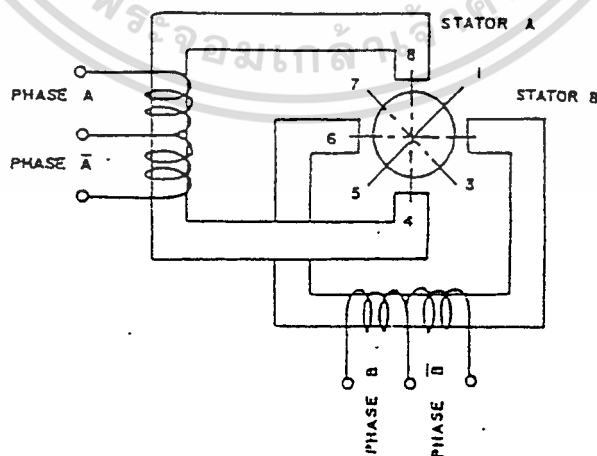
2.5.1 แบบจ่ายกระแสไฟฟ้าให้เฟสเดียว (ONE EXCITATION) วนไปเรื่อย ๆ

2.5.2 แบบจ่ายกระแสไฟฟ้าให้พร้อมกันทีเดียว 2 เฟส (TWO - EXCITATION) วนไปเรื่อย ๆ จะทำให้ส เตปปี้งมอ เตอร์มีแรงบิดที่เกิดขึ้นมากกว่าทุกแบบ

2.5.3 แบบจ่ายกระแสไฟฟ้าให้ทีละ 1 เฟส สลับกับ 2 เฟส (ONE TWO EXCITATION) วนไปเรื่อย ๆ จะทำให้จำนวนส เตปในการหมุนละเอียดมากกว่าสองแบบดังกล่าวข้างต้น



รูปที่ 2.5.1 ระบบควบคุมสแตมป์มอเตอร์

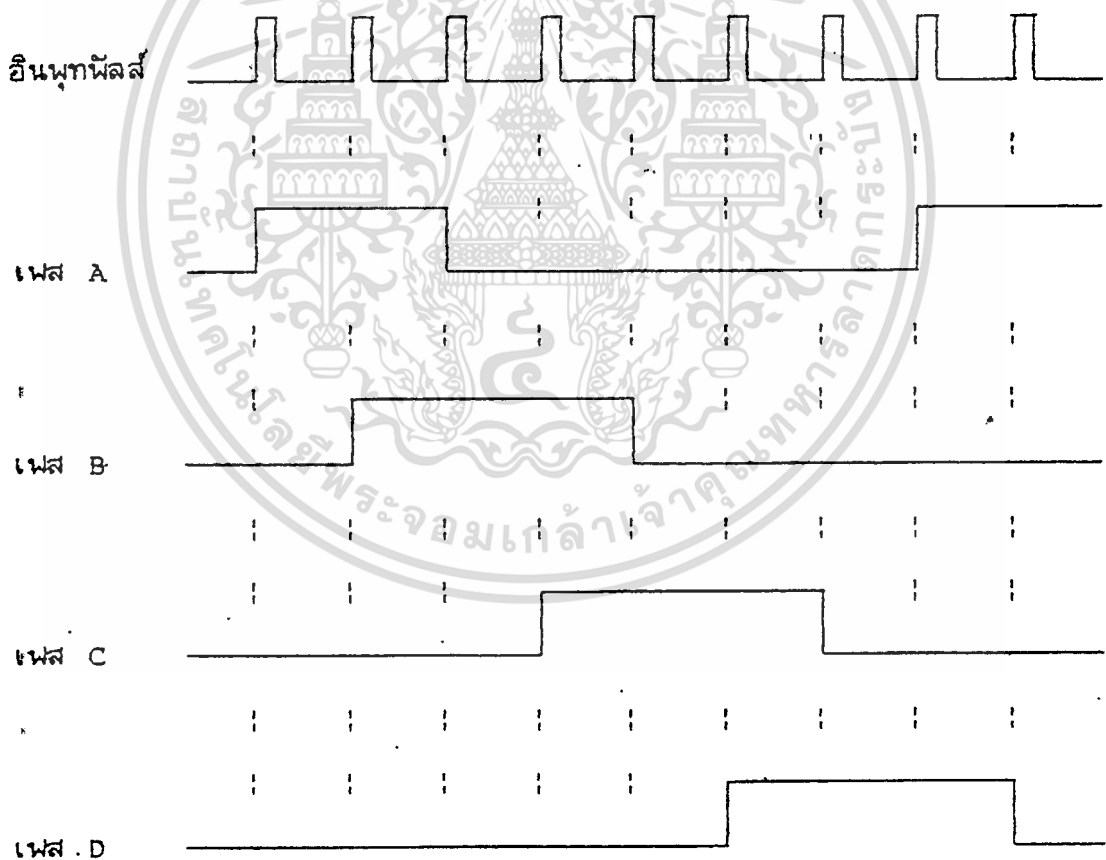


รูปที่ 2.5.2 การกระตุ้นที่ตำแหน่งต่างๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

วงจรไตรเวอร์สำหรับส เตปปีงมอ เตอร์ จะประกอบด้วยวงจรที่สำคัญ 2 ส่วนคือวงจรซีเคิวนลอจิก และวงจรขยายกำลัง วงจรที่ใช้จัดซีเคิวนลอจิกอาจ จะใช้ วงจรนับ หรือไมโครคอมพิวเตอร์ก็ได้ เพื่อจัดให้อินพุตพัลส์ที่เข้ามา เลื่อนลำดับไปทางขวา หรือซ้าย ซึ่งถ้าการจัดลำดับวงจรต์แดงที่จ่ายให้กับขดลวด ส เตเตอร์ของส เตปปีงมอ เตอร์ซึ่ง เรามักเรียกว่า เฟส คือ ถ้าจ่ายให้ลำดับ เฟส จากซ้าย เลื่อนไปทางขวา มอ เตอร์ก็จะหมุนตาม เข็มนาฬิกา แต่ถ้าจ่ายจากขวา เลื่อนไปทางซ้ายมอ เตอร์ ก็จะหมุนทวน เข็มนาฬิกา

วงจรซีเคิวนลอจิกแบบกระตุ้นทีละ 1 เฟส สลับ 2 เฟส สำหรับส เตปปีง มอ เตอร์ 4 เฟส 1 ทมมีดังต่อไปนี้แสดงในรูปแบบที่ 2.5.3

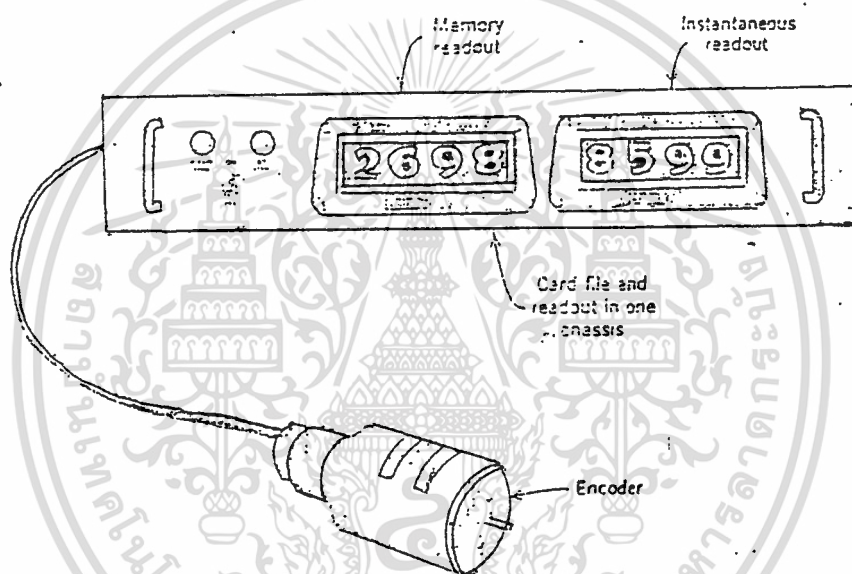


รูปที่ 2.5.3 1 ทมมีดังต่อไปนี้แสดงในรูปแบบที่ 2.5.3

เฟส สลับ 2 เฟส สำหรับส เตปปีงมอ เตอร์ 4 เฟส เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่ออนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.6 ชุด เซ็นเซอร์มุมของสายอากาศ

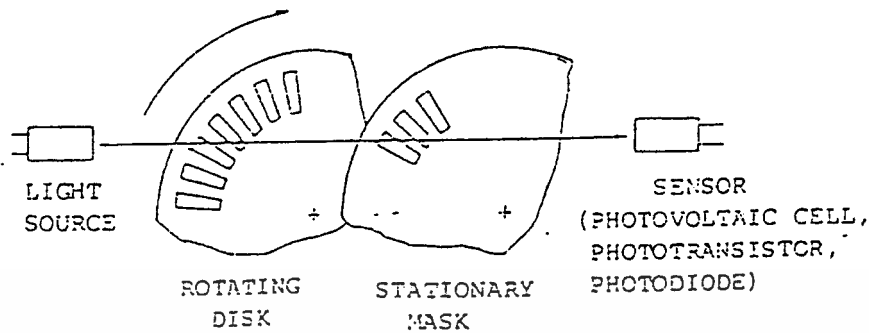
การตรวจสอบตำแหน่งการหมุนของส เตปมิ่งมอเตอร์ ว่าอยู่ที่มุมเท่าใดนั้นสามารถทำได้โดยการใช้อินครีเมนต์เอ็นโค้ดเดอร์ (increment encoder) หรือ เอนโค้ดเดอร์สำหรับรักษาตำแหน่งและสำหรับการสร้างสัญญาณป้อนกลับ โดยตัวเอ็นโค้ดเดอร์ จะสร้างสัญญาณพัลส์ที่แปรผันตรงกับการหมุน รูปที่ 2.6.1 แสดงระบบ เอนโค้ดเดอร์แบบโรตารี



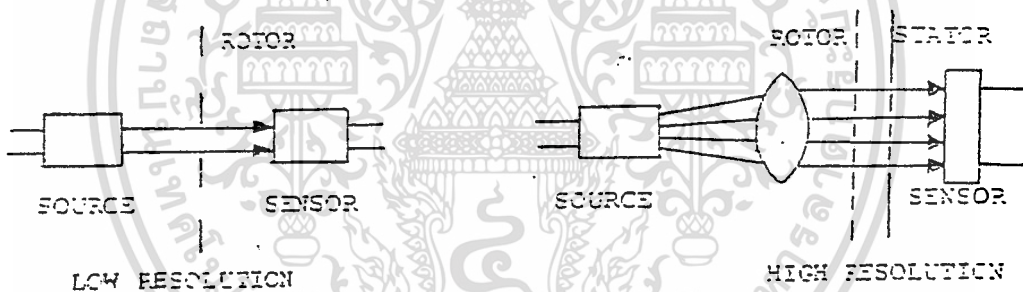
รูปที่ 2.6.1 ระบบ เอนโค้ดเดอร์แบบโรตารี

อินครีเมนต์เอ็นโค้ดเดอร์ประกอบด้วยส่วนที่สำคัญคือ ตัวกำเนิดแสง (LIGHT SOURCE) จานหมุน (ROTARY DISK) จานอยู่กับที่ (STATIONARY MASK) และ ตัวเซ็นเซอร์ (SENSOR) บนแผ่นจานหมุนทำเป็นช่องจัดรอบ ดังแสดงรูปที่ 2.6.2 และบนแผ่นอยู่กับที่จะมีช่องสำหรับให้แสงผ่านตรง เข้าไปยังตัวเซ็นเซอร์ ถ้า เป็นเอ็นโค้ดเดอร์ที่วัดความเร็วต่ำไม่ต้งมีแผ่นอยู่กับที่ก็ได้ ส่วนตัวกำเนิดแสงอาจจะเป็นหลอดไฟหรือ แอลอีดีก็ได้ ดังรูปที่ 2.6.3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



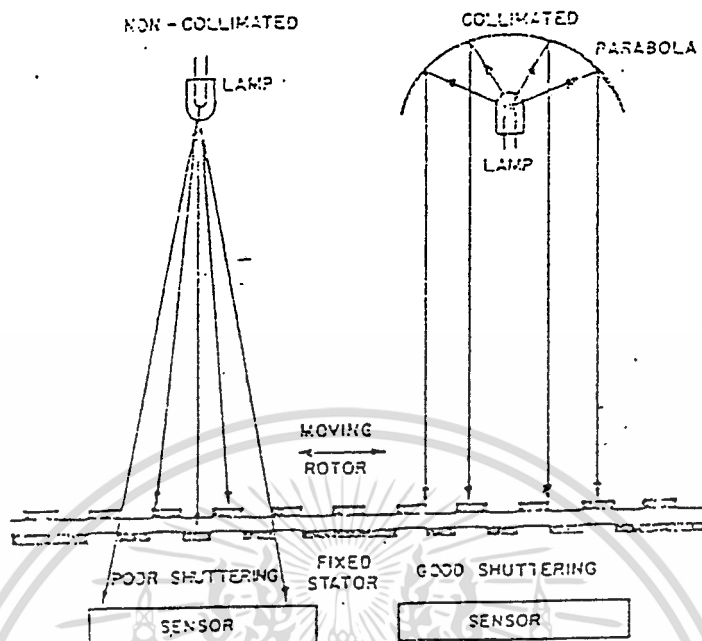
รูปที่ 24 ตัวอย่างกลไกของออปติคัลเอนคริ์



รูปที่ 2.6.3 แสดงตัว เซ็นเซอร์แบบมีช่องปิด เปิดให้แสงผ่านได้ช่อง เดียว และแบบมีหลายช่อง

ความละเอียดของอินคริ เมันท์เอนโค้ดเดอร์ คือจำนวนคาบเวลาของสัญญาณ เอาท์พุตต่อการหมุนของ เฟลา 1 รอบซึ่งเป็นจำนวนพัลส์ต่อรอบ หรือจำนวนพัลส์เคลตต่อ 360 องศา เอนโค้ดเดอร์ที่ใช้โดยทั่วไปมีค่าความละเอียดตั้งแต่ 15 ถึง 10,000 พัลส์ต่อรอบ . ในการปฏิบัติ เนื่องจากแสงที่ออกจากแหล่งกำเนิดแสงเป็นลำแสงเดี่ยว ถ้าเราต้องการให้แสงที่ผ่านไปยัง เอนโค้ดเดอร์ เป็นเส้นตรงพร้อม ๆ กันก็ทำได้โดยใช้เลนส์ หรือ พาราโบリックเพลคเตอร์ดังรูปที่ 2.6.4

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.6.4 แสดงผลของแสงที่เดินแนวเดียวกันและแสงที่แตกกระจาย

จำนวนฟลักซ์ต่อ 1 รอบของสัญญาณที่เอ็นโค้ดเตอร์สร้างออกมาจะเท่ากับจำนวนช่องว่างบนแผ่นจานหมุน และความกว้างของช่องว่างกับความกว้างของแถบที่บระหว่างช่องว่างจะเท่ากัน เพราะฉะนั้นสามารถคำนวณหาความกว้างของช่องว่าง (W) ได้จาก

$$W = \pi D / 2N$$

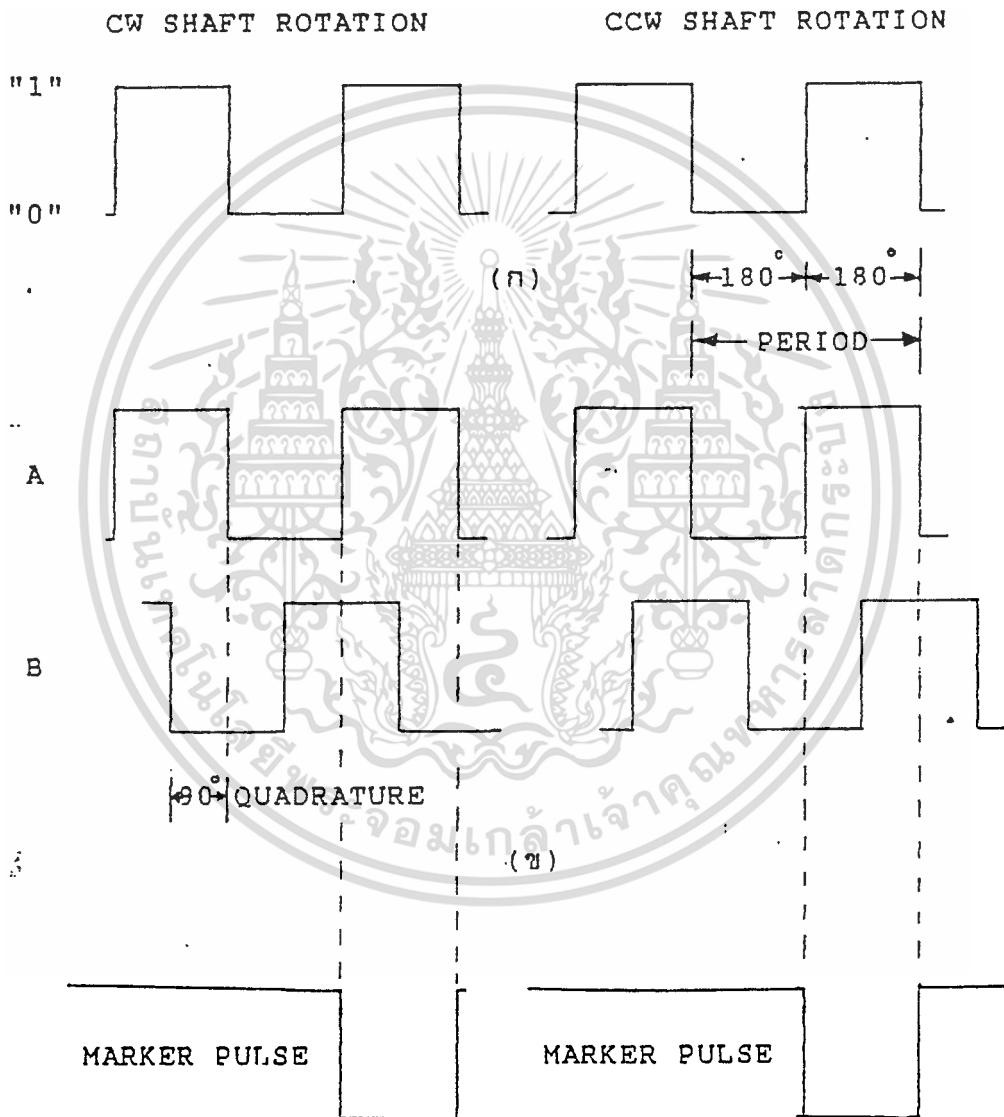
เมื่อ D คือ เส้นผ่าศูนย์กลางเฉลี่ยของ เพ็ก เก็น

N คือ จำนวนของความละเอียด เป็นฟลักซ์ต่อรอบ

W คือ ความกว้างของช่องว่าง

เอาที่พูดของ เอ็นโค้ดเตอร์โดยทั่วๆไปแล้ว สัญญาณเอาที่พูดที่ออกจาก เอ็นโค้ดเตอร์โดยตรงจะมีระดับไม่เพียงพอในการควบคุมหรือสำหรับการประมวลสัญญาณ ดังนั้นจึงต้องมีวงจรขยาย และแปลงรูปร่างลูกคลื่นสัญญาณต่อๆวันตัว เอ็นโค้ดเตอร์ด้วย เสมอ สัญญาณลูกคลื่นที่ได้จากตัว เซ็นเซอร์ปกติแล้วจะ เป็นรูปสัญญาณ เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สาม เหลี่ยม หรือรูปสี่เหลี่ยมผืนผ้าขึ้นอยู่กับความละเอียดที่ต้องการ รูปสี่เหลี่ยมเหล่านี้นี้สามารถทำให้เป็นสัญญาณรูปสี่เหลี่ยมได้ โดยการต่อตัวคอมพาราเตอร์ เข้ากับ ลิเนียร์แอมพลิฟายของ เอนโค้ดเดอร์ ก็จะได้ เอาท์พุทเป็นลูกคลื่นสี่เหลี่ยมตามต้องการ รูปที่ 2.6.5 ก และ รูปที่ 2.6.5 ข



รูปที่ 2.6.5 ก ลูกคลื่นเอาท์พุทสี่เหลี่ยมที่เอนโค้ดเดอร์ชนิด 1 ช่อง

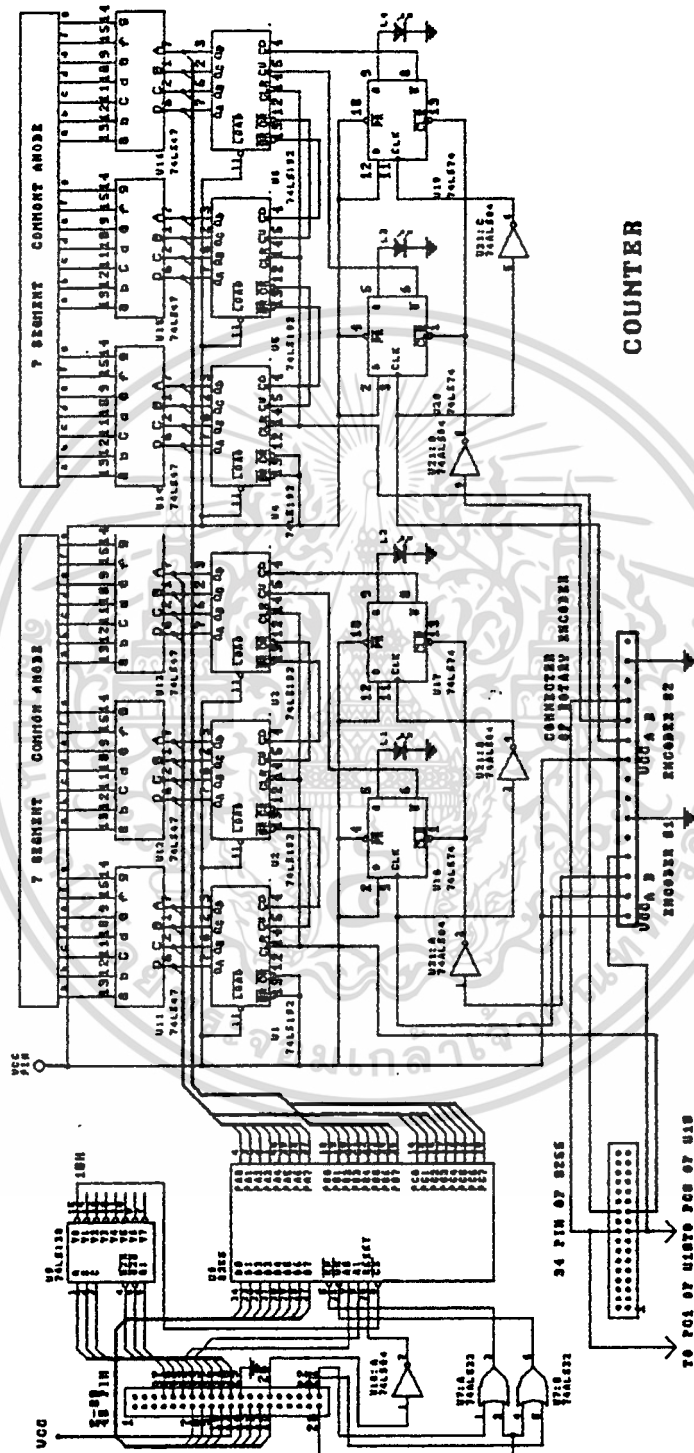
ข สัญญาณ 2 ชุดที่ได้จากเอนโค้ดเดอร์ชนิด 2 ช่อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 2.6.5ก แสดงถึงลูกคลื่น เอาท์พุทสี่เหลี่ยมที่เอนคัตเตอร์ชนิด 1 ช่อง ไม่ว่าเพล่าจะหมุนในทิศทางใด ก็จะได้สัญญาณเหมือนกันจึง เหมาะที่จะใช้กับงานที่ไม่กำหนดทิศทาง เท่านั้น ส่วนในรูปที่ 2.6.5ข แสดงสัญญาณ 2 ชุดที่ได้จากเอนคัตเตอร์ชนิด 2 ช่อง เฟสของสัญญาณ 2 ช่องนี้จะต่างกัน 90 องศาทางไฟฟ้า เราเรียกสัญญาณ 2 ช่องนี้ว่าเป็น ควอดราเจอร์ (quadrature) กัน ซึ่งเหมาะที่จะใช้ในการรับรู้ทิศทางการหมุนของเพล่า หรือใช้ควบคุมระบบที่ซับซ้อนอื่นๆ จากสัญญาณในรูปที่ 2.6.5ข จะเห็นว่าสัญญาณทั้งสองช่องจะเริ่มจาก 0 ถึง 1 และ 1 ถึง 0 ขึ้นอยู่กับทิศทางการหมุนของแผ่นหมุนของเอนคัตเตอร์ อันอินครีเมนต์เอนคัตเตอร์บางชนิด จะมีพัลส์ที่แสดงถึงจำนวนรอบของการหมุนสำหรับให้ เป็นศูนย์ในการอ้างอิงพัลส์ที่ใช้แสดงจำนวนรอบนี้จะเกิดขึ้น 1 พัลส์ต่อ 1 รอบ โดยทั่วไปแล้ว เราจะบอกถึงตำแหน่งเชิงกล หรือใช้เป็นสัญญาณเคลียร์จำนวนที่นับไว้หน่วย เก็บข้อมูล

2.7 วงจรนับ (COUNTER)

เป็นวงจรที่รับสัญญาณอินพุทมาจาก สัญญาณ เอาท์พุทของ เอนคัตเตอร์ ซึ่งเป็นสัญญาณพัลส์ โดยสัญญาณพัลส์ดังกล่าวนี้ไม่สามารถให้ค่า เอาท์พุทออกมาเป็นค่าตัวเลขได้ แต่เราต้องการค่าตัวเลขนี้ไปแสดงค่ามุมของสายอากาศที่หันชี้ไปยังตำแหน่งต่างๆ เราจึงต้องใช้วงจรนับ เพื่อนับพัลส์ที่ออกมาจากเอนคัตเตอร์ ซึ่งวงจรนับแสดงดังรูปที่ 2.7.1



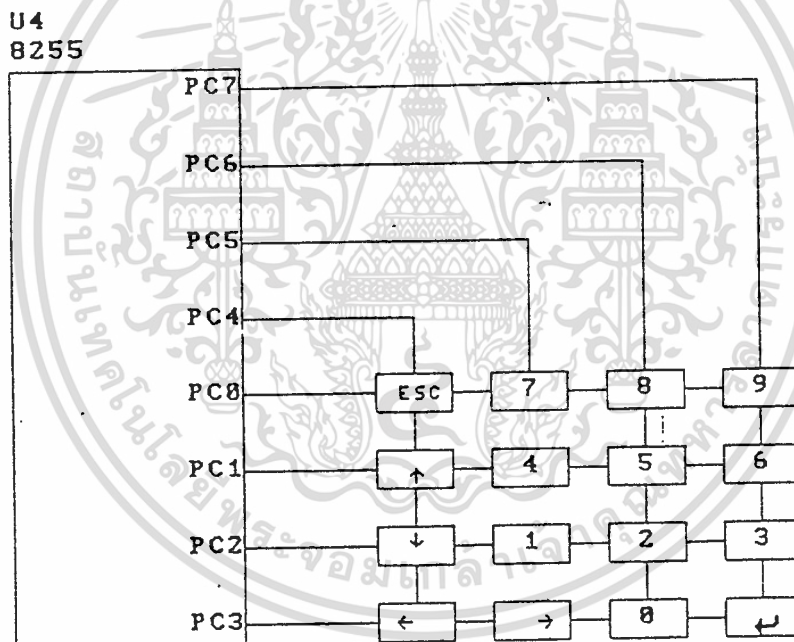
รูปที่ 2.7.1 วงจรนับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.8 แป้นกดแบบเมตริกซ์ 4 x 4

โดยการนำเอาซี 8255 มาใช้งานในโหมดศูนย์ เลือกใช้พอร์ท C บนเป็นอินพุต และพอร์ท C ล่าง เป็นเอาต์พุต เพื่อรับรู้การกดแป้นคีย์ และหาตัวว่ามีคีย์ตำแหน่งใดถูกกดซึ่งสามารถจัดแป้นคีย์ในรูปแบบเมตริกซ์ 4 x 4 หรือ 16 คีย์

หลักการโดยทั่วไปจะทำการสแกนคีย์ที่ตำแหน่งต่างๆโดยการให้แต่ละแถวที่เป็นพอร์ทเอาต์พุตเป็นศูนย์หรือหนึ่งในเวลาต่างกัน เช่น แถวแรกเป็นศูนย์ ส่วนแถวอื่นเป็นหนึ่งหมด แล้วทำการอ่านข้อมูลที่พอร์ทอินพุตดูว่ามีบิตใดบิตหนึ่งทางคอลัมน์เป็นศูนย์ หรือไม่ ถ้ามีก็ทราบได้ว่าคีย์ในตำแหน่งแถวแรกที่เท่าาได้ เป็นคีย์ที่ได้รับการกด การต่อแป้นกด กับพอร์ท8255แสดงดังรูปที่ 2.8.1



รูปที่ 2.8.1 แสดงการต่อแป้นกด 4 x 4

2.9 ระบบ เองซีบสายอากาศ

ระบบ เองซีบสายอากาศประกอบด้วย

2.9.1 งานเสา มีขาตั้ง 4 ขาสามารถยึดติดกับพื้นได้ป้องกันการล้ม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.9.2 ชุดเฟือง แบ่งออกเป็น 2 ส่วนคือ ส่วนของมุมอาซิมุท และมุม
อีเลเวชันของสายอากาศ แต่ละส่วนประกอบด้วย เฟืองตรงทดรอบ 6 ชุด
หลักการคำนวณฟันเฟือง

$$\text{อัตราทด (i)} = n_1/n_2 = z_2/z_1 = t_2/t_1$$

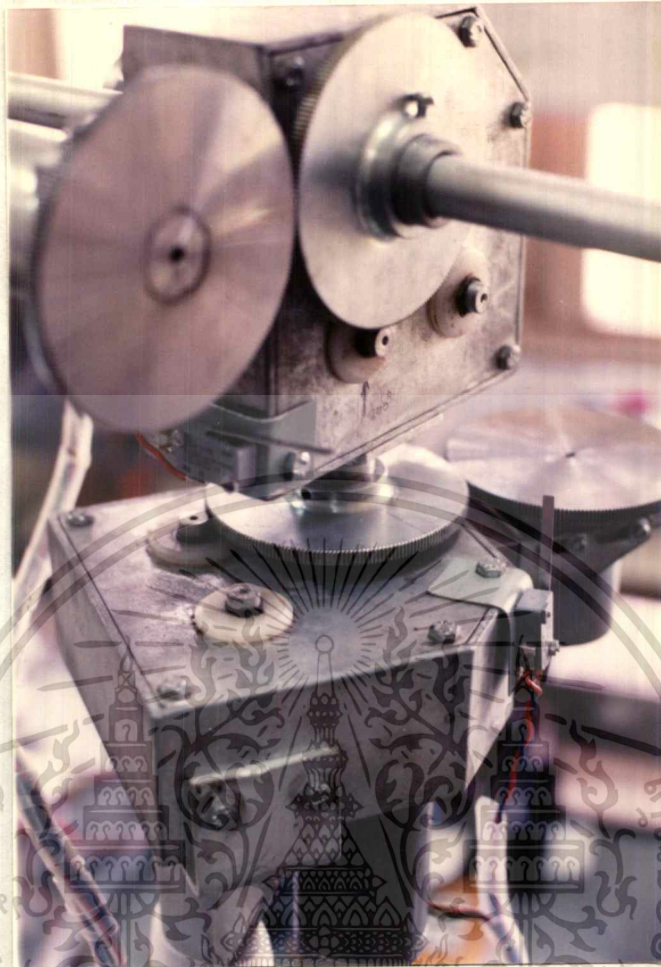
z_1	: จำนวนฟันเฟืองขับตัวที่ 1	=	20	ฟัน ; ที่มอเตอร์
z_2	: จำนวนฟันเฟืองตามตัวที่ 2	=	100	ฟัน
z_3	: จำนวนฟันเฟืองขับตัวที่ 3	=	50	ฟัน
z_4	: จำนวนฟันเฟืองตามตัวที่ 4	=	100	ฟัน
z_5	: จำนวนฟันเฟืองขับตัวที่ 5	=	50	ฟัน
z_6	: จำนวนฟันเฟืองตามตัวที่ 6	=	100	ฟัน ; ที่แกนสายอากาศ
n_1	: ความเร็วรอบของ เฟืองตัวที่ 1	=	20	รอบ
n_2	: ความเร็วรอบของ เฟืองตัวที่ 2	=	4	รอบ
n_3	: ความเร็วรอบของ เฟืองตัวที่ 3	=	4	รอบ
n_4	: ความเร็วรอบของ เฟืองตัวที่ 4	=	2	รอบ
n_5	: ความเร็วรอบของ เฟืองตัวที่ 5	=	2	รอบ
n_6	: ความเร็วรอบของ เฟืองตัวที่ 6	=	1	รอบ
t_1	: แรงบิดของ เฟืองตัวที่ 1			
t_2	: แรงบิดของ เฟืองตัวที่ 2			

จากผลการคำนวณ เมื่อกำหนดให้สเกลบีงมอเตอร์หมุนครบ 20 รอบ
(n_1) เฟืองตัวสุดท้ายที่แกนสายอากาศจะหมุนครบ 1 รอบ (n_6) จากตารางคุณสมบัติ
ของสเกลบีงมอเตอร์ จะเห็นว่า 1 รอบของเฟืองตัวแรกเท่ากับ 200 สเกลของมอ
เตอร์ ดังนั้นสเกลบีงมอเตอร์จะต้องหมุน 4000 สเกล เฟืองตัวสุดท้ายที่แกนสาย
อากาศจะหมุนครบ 1 รอบจาก 20 รอบของสเกลบีงมอเตอร์ ทำให้สายอากาศ
หมุน 360 องศา ดังนั้น 1 รอบของสเกลบีงมอเตอร์ ทำให้สายอากาศหมุน 18
องศา ดังนั้น 1 สเกลของสเกลบีงมอเตอร์ ทำให้สายอากาศหมุน 0.09 องศา

พิจารณาที่ชุดเฟืองของ เอนคัตเตอร์ประกอบด้วย เฟืองที่ชุดสาย
อากาศ 210 ฟัน และเฟืองที่เอนคัตเตอร์ 210 ฟันซึ่งมีอัตราทด 1:1 และดัง

รูปที่ 2.9.1 จำนวนฟิลล์ที่ออกจาก เอนคัตเตอร์ต่อรอบ เท่ากับ 360 ฟิลล์ เมื่อเฟือง

เอกสารนี้เป็นเอกสารลิขสิทธิ์ของกรมการศึกษานอกโรงเรียน
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.9.1 แสดงชุด เฟืองที่เอนโค้ดเตอร์

ที่ชุดสายอากาศหมุนครบ 1 รอบ ทำให้เอนโค้ดเตอร์หมุนครบ 1 รอบ ผลิตพัลส์ 360 พัลส์ส่งฯหัววงจรนับ

สมการ การส่งกำลังจากเฟืองตัวที่ 1 ถึง เฟืองตัวที่ 6 แสดงฯได้ดังนี้

$$t_6 = t_1 \cdot i_1 \cdot i_2 \cdot i_3$$

$$i_1 = z_2/z_1 = 100/20 = 5$$

$$i_2 = z_4/z_3 = 100/50 = 2$$

$$i_3 = z_6/z_5 = 100/50 = 2$$

สมการแรงบิด

$$t = p \cdot 7 / (44 \cdot s)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

t = แรงบิด (นิวตัน-เมตร)

p = กำลังที่ส เตะปิ้งมอ เตอร์ส่งออก (วัตต์)

s = ความเร็วรอบของส เตะปิ้งมอ เตอร์ (รอบ/วินาที)

ตัวอย่างการคำนวณแรงบิด

กำลังที่ส เตะปิ้งมอ เตอร์ส่งออก (p) = 7.88 วัตต์

ความเร็วรอบของส เตะปิ้งมอ เตอร์ (s) = 2/60 รอบ/วินาที

$$t_1 = 7.88 \times 7 / (44 \times 2 / 60)$$

$$= 37.6 \quad \text{นิวตัน-เมตร}$$

$$t_6 = 37.6 \times 5 \times 2 \times 2$$

$$= 752.19 \quad \text{นิวตัน-เมตร}$$



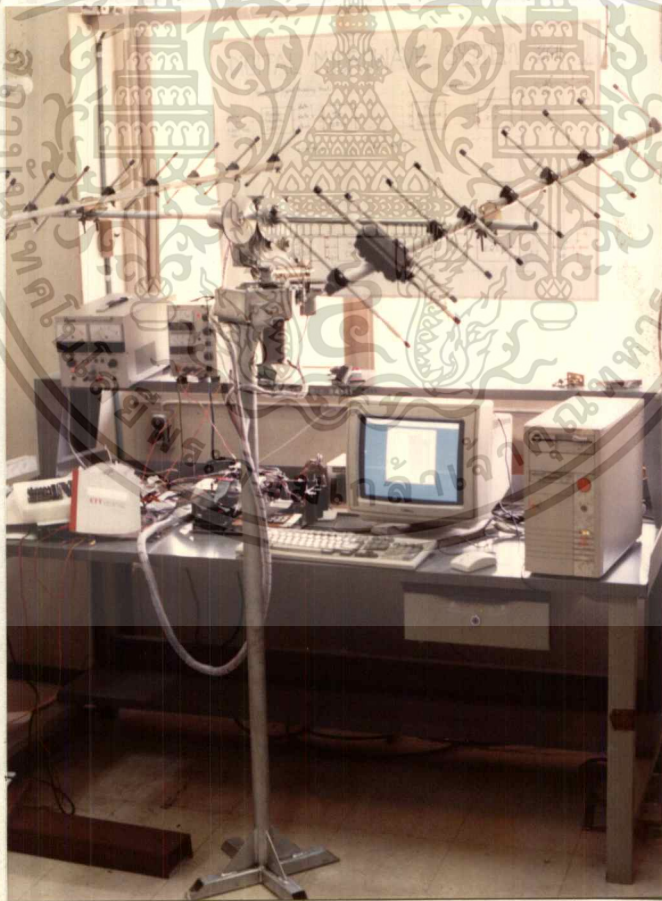
บทที่ 3

การทดลอง และผลการทดลอง

ในการทดลองระบบควบคุมสายอากาศติดตามดาวเทียมวงโคจรต่ำประกอบไปด้วยส่วนใหญ่นั่นคือ

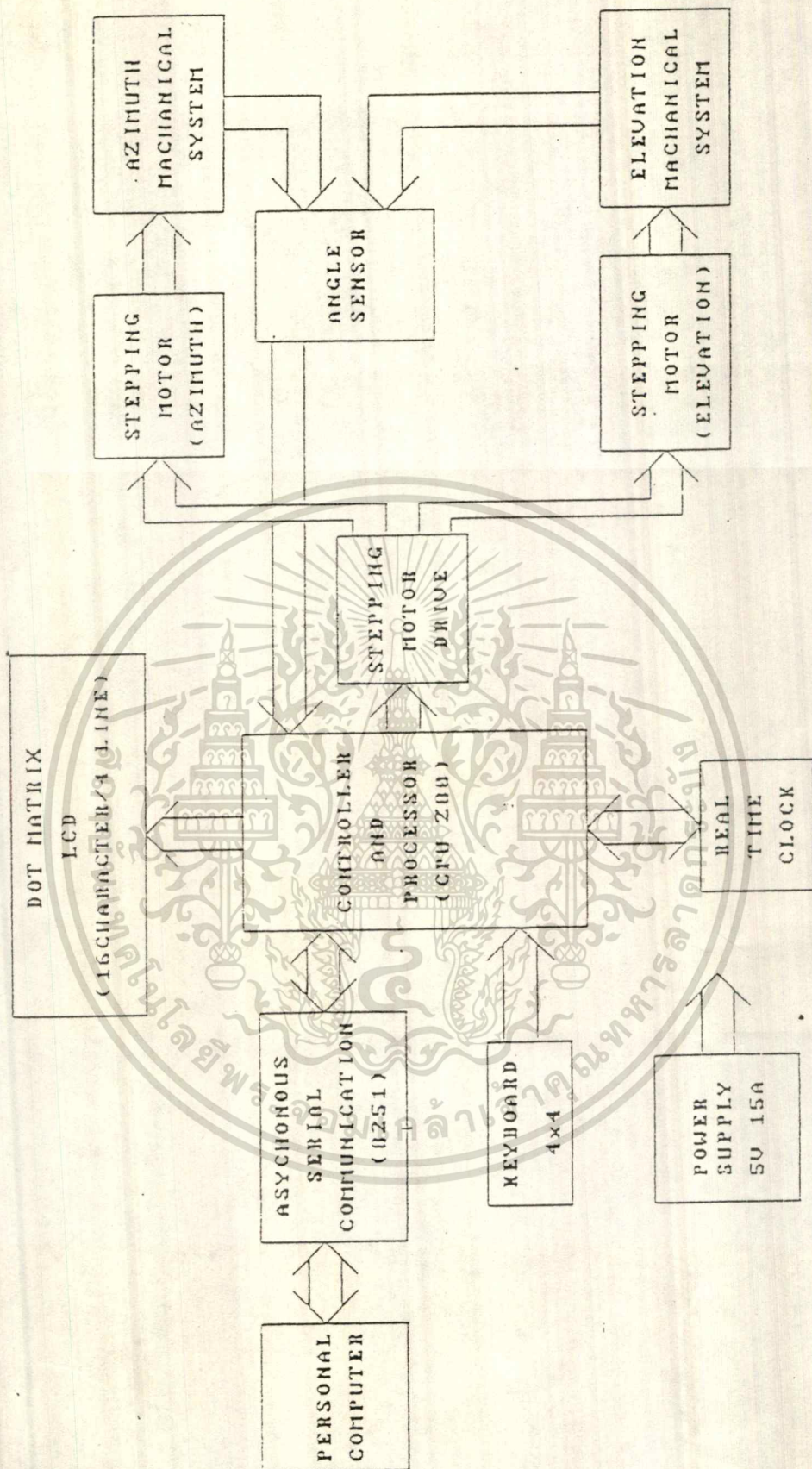
- 3.1 เครื่องคอมพิวเตอร์
- 3.2 ชุดไมโครโปรเซสเซอร์ z-80
- 3.3 ชุดขับสแตปมอเตอร์
- 3.4 ชุดเซ็นเซอร์มุมของสายอากาศ
- 3.5 ระบบเฟืองขับสายอากาศ

โดยส่วนต่าง ๆ สามารถแสดงได้ดังรูปที่ 3.1 และรูปที่ 3.2



รูปที่ 3.1 ภาพถ่ายส่วนต่าง ๆ ของระบบติดตามดาวเทียมวงโคจรต่ำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



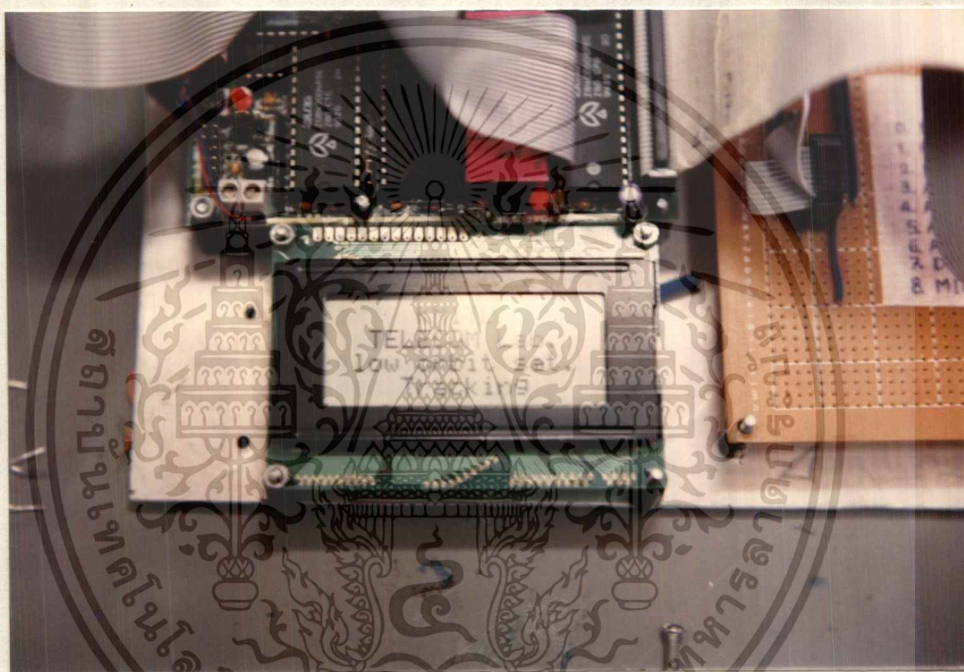
รูปที่ 3.2 บล็อกไดอะแกรมของระบบติดตามดาว เทียมวงโคจรต่ำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากบล็อกเดอะแแกรมมีการใช้บอร์ดรีลไทม์คล็อก ต่อกับบอร์ดไมโครโปรเซสเซอร์ z-80 โดยผ่านทางคอนเน็กเตอร์ (CONNECTER) 40ขา แล้วทำการโปรแกรมให้บอร์ดรีลไทม์คล็อกจัดการเกี่ยวกับเวลา โดยนาฬิกาที่ได้แสดงออกที่จอแอลซีดี

การแสดงการ เริ่มต้นการทำงานของระบบติดตามดาว เทียม แสดงดังรูปที่ 3.3



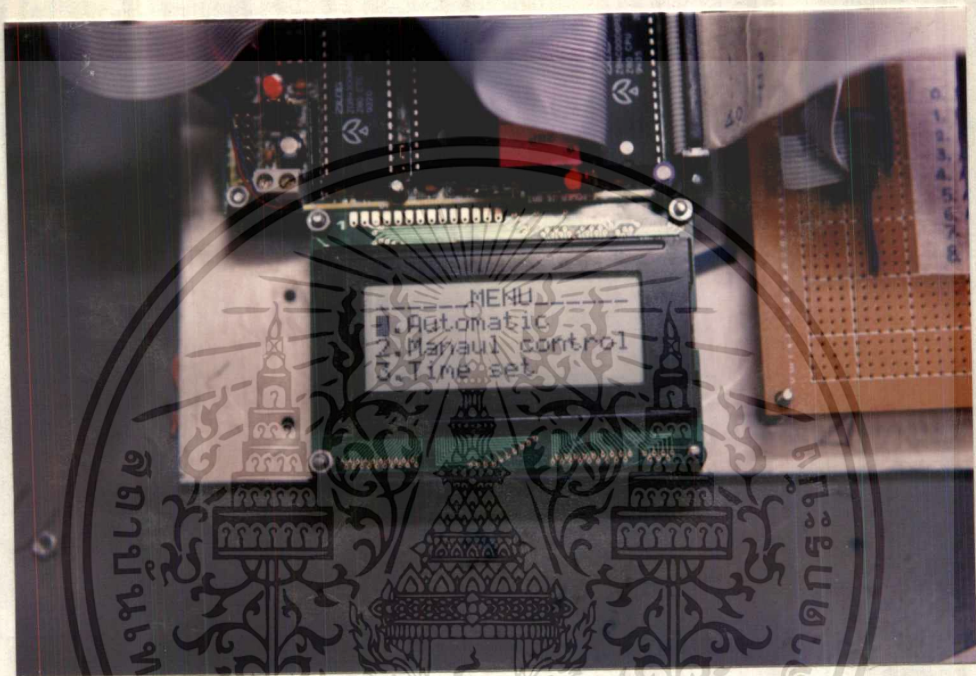
รูปที่ 3.3 แสดงการ เริ่มต้นการทำงานของ

โดยที่บอร์ดไมโครโปรเซสเซอร์ z-80 ยังได้มีโปรแกรมควบคุมการแสดงผลที่จอแอลซีดี เป็นเมนูดังนี้ แสดงดังรูปที่ 3.4

- เมนูการทำงานชื่อ "AUTOMATIC" ใช้ในการควบคุมการปรับมุมของสายอากาศแบบอัตโนมัติ โดยการนำข้อมูลที่เก็บไว้ในหน่วยความจำมาทำการควบคุมการปรับมุมของสายอากาศให้ถูกต้อง ซึ่งข้อมูลดังกล่าว เป็นข้อมูลที่เกี่ยวกับดาว เทียมแต่ละดวง โดยจะมีข้อมูลเกี่ยวกับ ค่ามุมเอ้าซิมุท และมุมอี เล เวชันที่เฉพาะเวลาใด ๆ ที่เราสามารถติดต่อกับดาว เทียมดวงที่ต้องการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- เมนูการทำงานชื่อ "MANUAL" ใช้ในการควบคุมการปรับมุมของสายอากาศโดยควบคุมโดยการกดคีย์
- เมนูการทำงานชื่อ "TIME SET" ใช้ในการตั้ง เวลาของรีเลย์เทมคลิก ล็อก เช่น วัน เดือน ปี ชั่วโมง นาที และวินาที



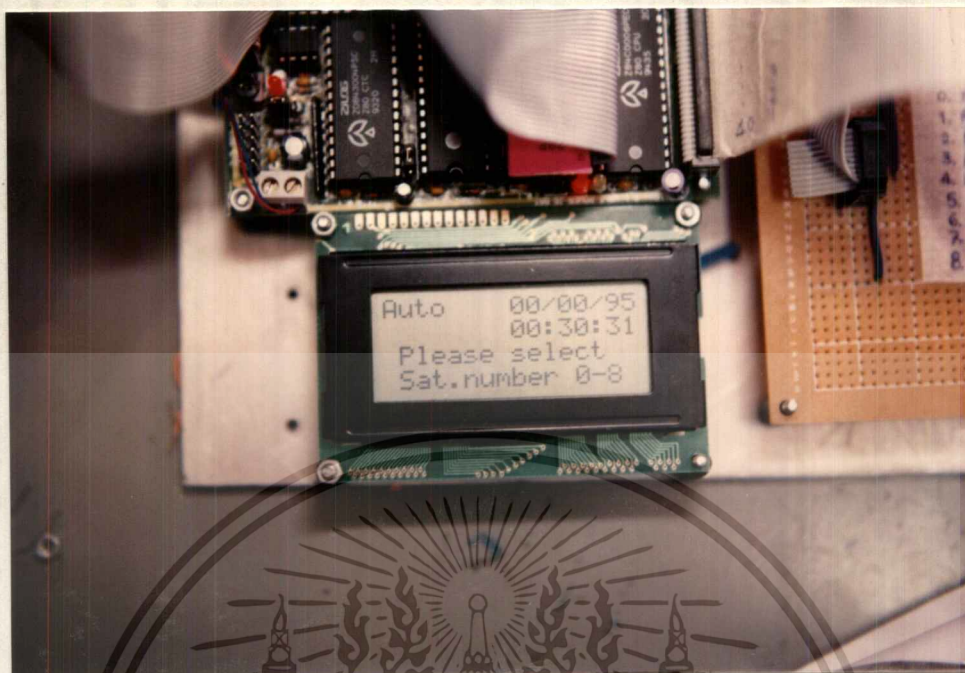
รูปที่ 3.4 แสดง เมนู

นอกจากนี้ที่บอร์ดไมโครโปรเซสเซอร์ Z-80 ยังมีโปรแกรมสแกนคีย์ (SCAN KEY) เพื่อรับรู้การกดคีย์ (KEY) ซึ่งการกดคีย์ต่างๆให้ผลดังนี้

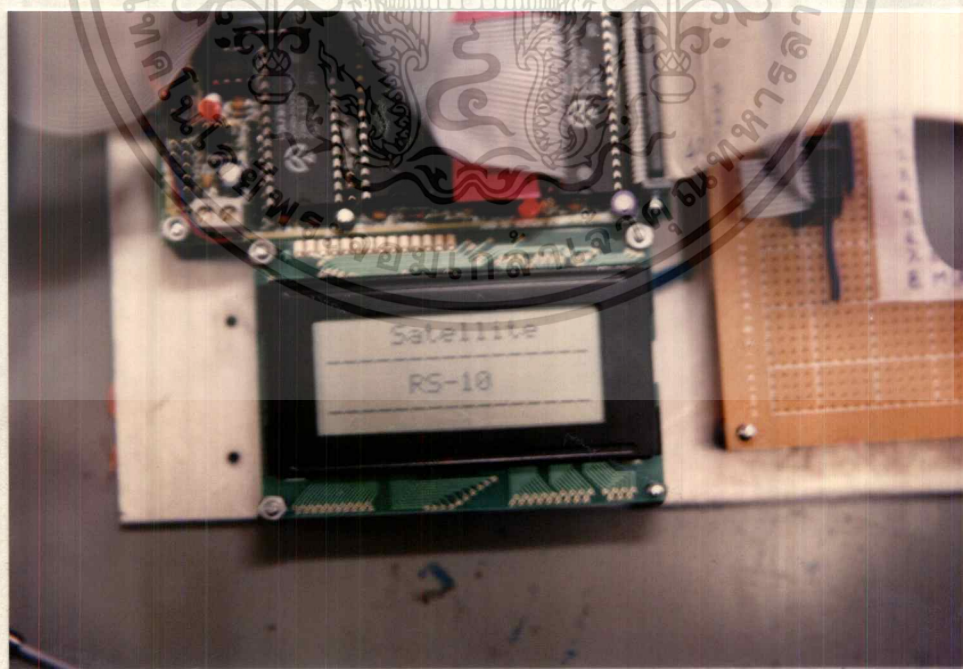
การเลือกเมนูทำได้โดยการเลื่อนคีย์ลูกศร ↑ และ ↓

- ถ้าเลือก 1. ได้ผลดังรูปที่ 3.5

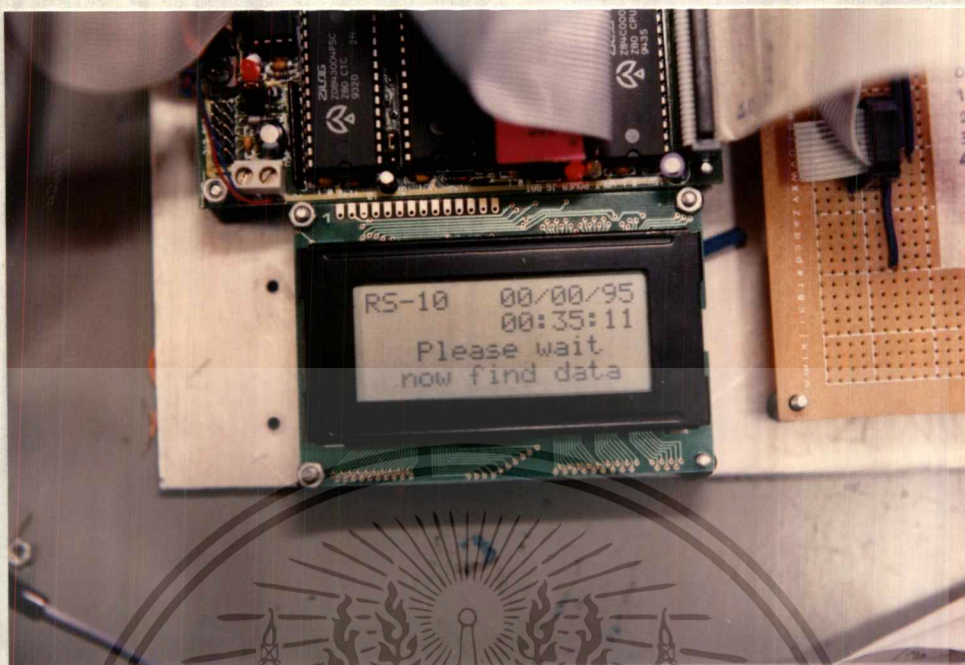
จากรูปที่ 3.5 จะแสดงข้อความถามผู้ช้ว่า ต้องการติดตามดาว เทียมดวงไหน เช่น ถ้าเลือกดาว เทียมชื่อ RS-10/11 ให้กดคีย์ 0 ซึ่งให้ผลดังรูปที่ 3.6 (ก) และ (ข) ซึ่งถ้าข้อมูลของดาว เทียมดวงที่เรา เลือกไม่ได้ผ่านการปรับรูป ให้ เป็นปัจจุบันจะให้ผลดังรูปที่ 3.7



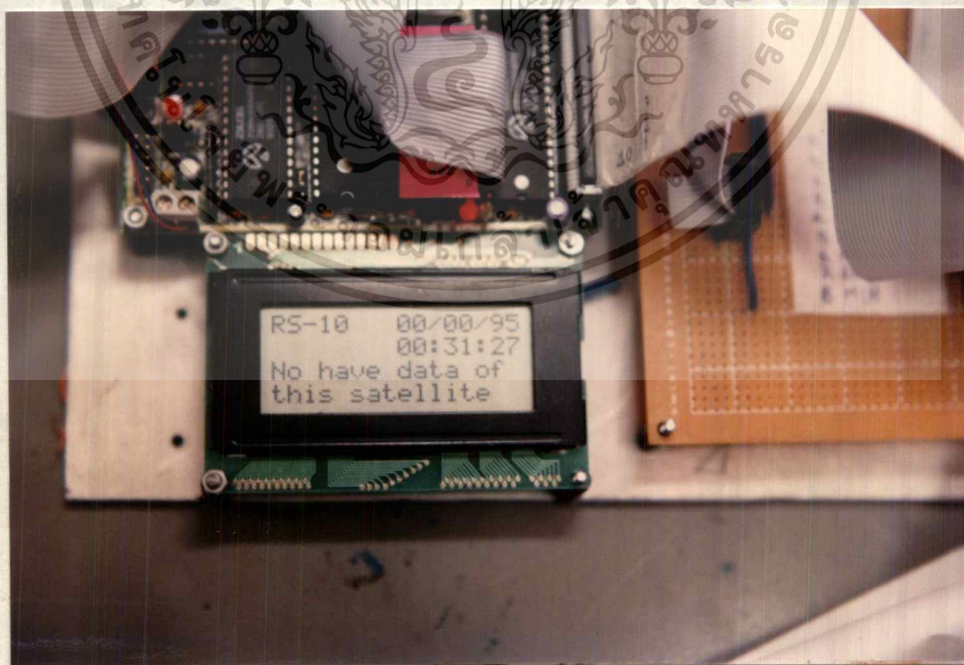
รูปที่ 3.5 แสดงผลภายหลังจากเลือกเมนูแบบ AUTOMATIC



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับรูปที่ 3.6 (ก) แสดงชื่อของดาวเทียม
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

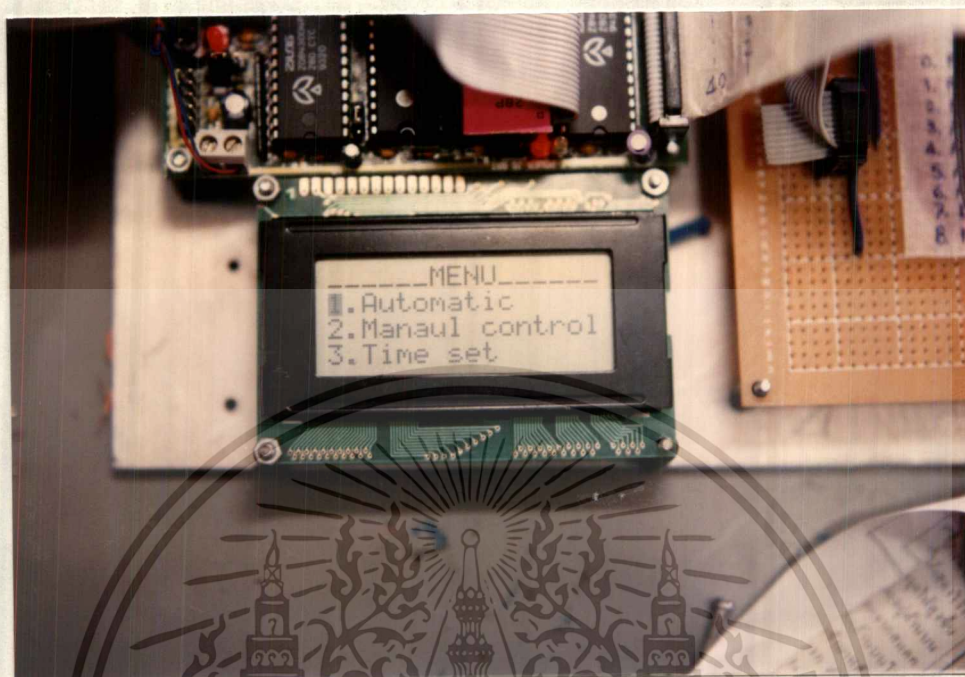


รูปที่ 3.6 (ข) แสดงผลจากระบบกำลังค้นหาข้อมูล



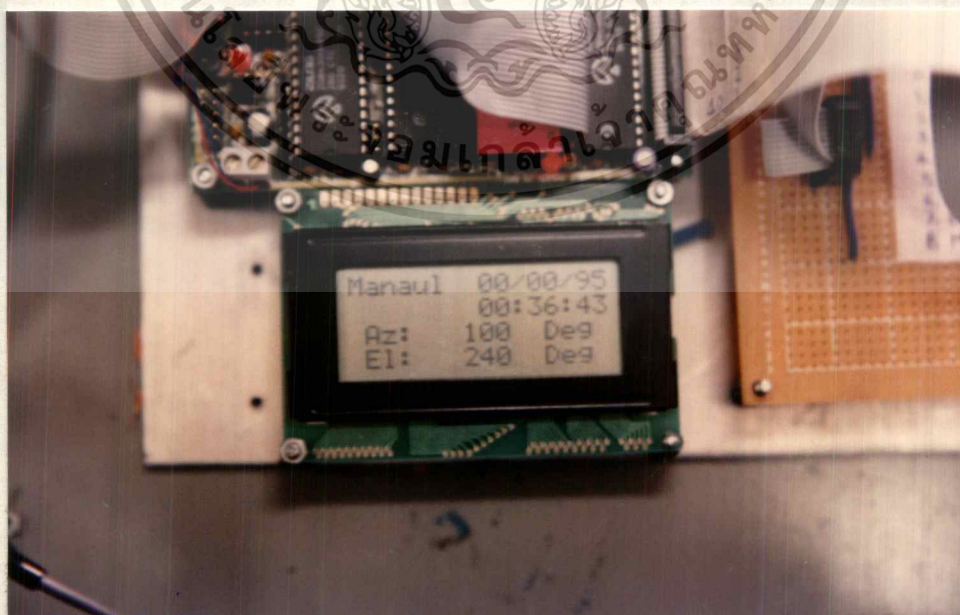
เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ 3.7 แสดงผล เมื่อข้อมูลของตัวดาวเทียมล้าสมัยแล้ว ระยะเวลาในการค้นหาข้อมูลจะเพิ่มขึ้นเรื่อยๆ ไม่สามารถหาข้อมูลได้ทั้งหมด อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อกด ESC จะทำให้กลับไปเมนูเหมือนเดิมดังรูปที่ 3.8



รูปที่ 3.8 แสดง เมนู

- ถ้าเลือก 2. ได้ผลดังรูปที่ 3.9

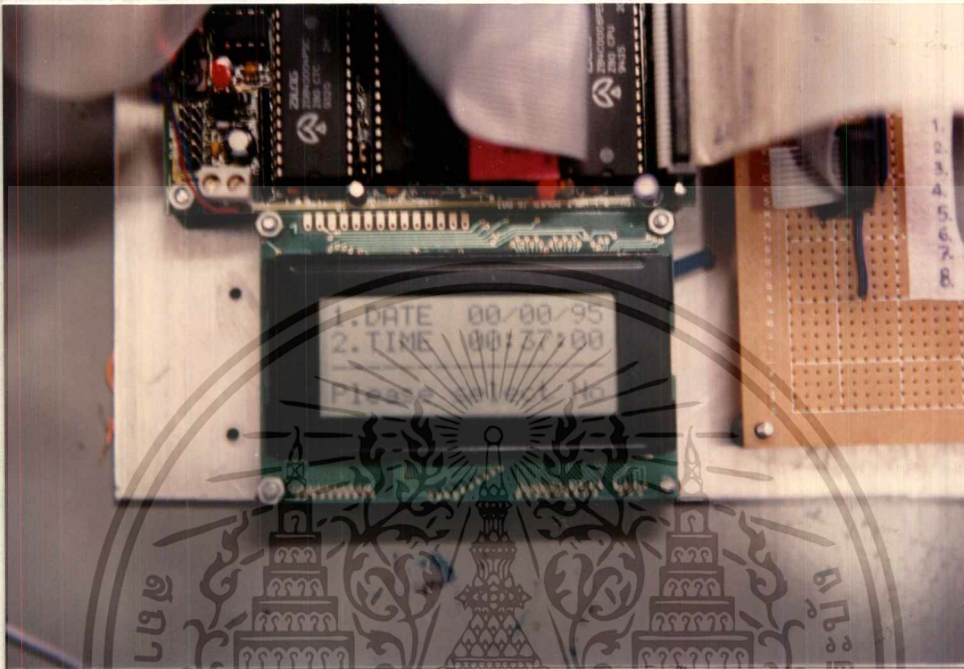


รูปที่ 3.9 แสดงผลหลังจาก เลือก เมนูแบบ MANUAL

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หลังจากนั้นใช้คีย์ลัดควบคุมทิศทางของสายอากาศตามต้องการ

- ถ้าเลือก 3.๙ได้ผลดังรูปที่ 3.10



รูปที่ 3.10 แสดงผลหลังจากเลือกเมนู TIME SET

จากรูป ถ้าเลือก 1. คือ ต้องการตั้ง วัน เดือน ปี

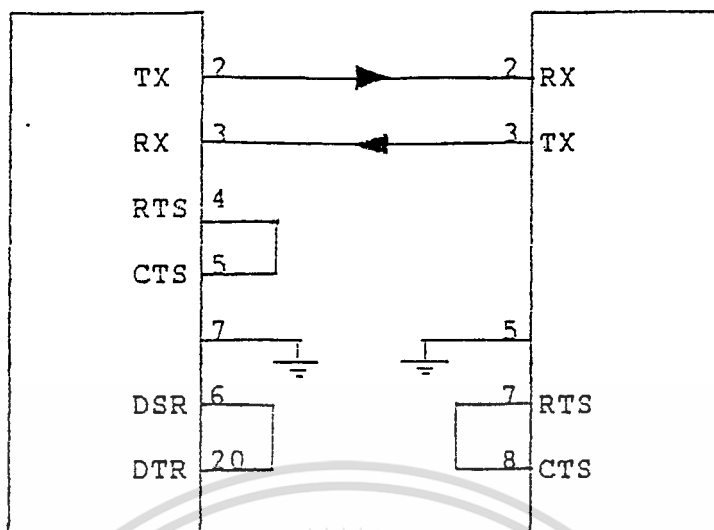
ถ้าเลือก 2. คือ ต้องการตั้ง เวลา

การนำคอมพิวเตอร์ มาต่อกับ บอร์ดควบคุมจะต่อบอร์ด 8251 ใช้ คอนเน็กเตอร์ แบบ DB-25 ที่เครื่องคอมพิวเตอร์ ต่อกับ DB-9 ที่บอร์ดควบคุมการส่งข้อมูลใช้ความเร็ว 2400 bps โดยส่งข้อมูลออกทาง COM2 ลักษณะการต่อสายระหว่าง DB-25 ของเครื่องคอมพิวเตอร์ กับ DB-9 ของบอร์ด 8251 แสดงดังรูปที่ 3.11

ลักษณะการส่งข้อมูล เป็น แบบอนุกรม ไม่มีพริตทรีตี ความยาวของข้อมูลแบบ 8 บิต การรับส่งข้อมูล เป็นแบบ STOP AND WAIT คือ จะส่งข้อมูล 1 byte ซึ่งจะมีการหยุดรอการตอบรับจากฝ่ายรับว่าได้รับถูกต้อง หรือยัง เมื่อฝ่ายรับตอบกลับว่าได้รับข้อมูลถูกต้อง ฝ่ายส่งจะทำการส่งข้อมูลออกไปอีก 1 byte เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

DB-25

DB-9



รูปที่ 3.11 แสดงการต่อสายระหว่าง เครื่องคอมพิวเตอร์กับบอร์ด 8251

และรอการตอบรับจากฝ่ายรับอีก เป็นเช่นนี้เรื่อย ๆ ไป

ในการติดต่อรับส่งข้อมูลระหว่าง เครื่องคอมพิวเตอร์ กับบอร์ดไมโครโปรเซสเซอร์ Z-80 โดยผ่านพอร์ท RS-232 นั้นจะมีการเขียนโปรแกรมภาษาซี โดยผ่าน รอม-ไบออส ซึ่งเป็นการใช้อินเตอร์รัพ 14H

อินเตอร์รัพ 14H (COMMUNICATIONS INPUT/OUTPUT : INT 14H) เป็นการ จัดเตรียมการรับส่งข้อมูลเป็นไบต์ โดยผ่าน RS-232 (RS232: communication port) รีจิสเตอร์ DX (DX REGISTER) ซึ่งจะประกอบด้วย หมายเลขของอแดปเตอร์ (adapter) โดยหมายเลข 0-3 เป็นหมายเลขสำหรับ คอม 1, คอม 2, คอม 3 และคอม 4 (COM1, COM2, COM3, COM4) หมายเลขของอแดปเตอร์จะถูกตั้ง โดยการกำหนดหมายเลข 0-3 เข้าไปในรีจิสเตอร์ AH (AH REGISTER)

บริการ 00 (SERVICE 00 : INITIALIZE COMMUNICATIONS PORT) ใช้สำหรับการตั้งพารามิเตอร์ต่างๆ เช่น บอดเรท พาริตี สติ๊อปบิท และความยาวข้อมูลโดยการกำหนดค่าของรีจิสเตอร์ AL (AL REGISTER) ให้สอดคล้องกับรูปที่ 3.12 ตามที่เราต้องการ

การทำงานในโหมดคั่นสถานะของพอร์ทสื่อสาร (communications) เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น มิได้นับว่าได้อนุญาตให้เผยแพร่หรือใช้โดยไม่ได้รับอนุญาตจากเจ้าของเอกสาร หากมีการนำเอกสารนี้ไปใช้โดยไม่ได้รับอนุญาตจากเจ้าของเอกสาร เจ้าของเอกสารจะไม่รับผิดชอบต่อความเสียหายใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

port) จะถูกคืนโดยผ่าน รีจิสเตอร์ AX (AX REGISTER) ดังรูปที่ 3.13

บริการ 01 (SERVICE 01: TRANSMIT CHARACTER) เป็นการส่งข้อมูลเป็นแบบท โดยจะทำการโหลดรีจิสเตอร์ AL ด้วยตัวอักษรที่ต้องการส่ง และรีจิสเตอร์ DX ด้วยหมายเลขพอร์ท การทำงานจะคืนค่าสถานะพอร์ท (STATUS-PORT) ในรีจิสเตอร์ AH ถ้าการทำงานนั้นไม่สามารถส่งแบบทข้อมูลได้ มันจะเซตบิตที่ 7 ของรีจิสเตอร์ AH เพื่อรายงานถึงการผิดพลาดทางเวลา (time out error)

BAUD RATE	PARITY	STOP BIT	WORD LENGTH
7-5	4-3	2	1-0
000 = 110	00 = NONE	0 = 1	10 = 7 BITS
001 = 150	01 = ODD	1 = 2	11 = 8 BITS
010 = 300	10 = NONE		
011 = 600	11 = EVEN		
100 = 1200			
101 = 2400			
110 = 4800			
111 = 9600			

รูปที่ 3.12 ตารางการตั้งพารามิเตอร์ต่างๆ ของอินเตอร์รับ 14H

บริการ 02 (SERVICE 02: RECEIVE CHARACTER) เป็นการรับข้อมูลเป็นแบบท โดยจะทำการโหลดหมายเลขพอร์ทเข้าไปในรีจิสเตอร์ DX การทำงานนั้นจะรับตัวอักษรจากสายสื่อสาร (COMMUNICATION LINE) เข้าไปในรีจิสเตอร์ AL บริการนี้จะเซตรีจิสเตอร์ AH ด้วยค่าสถานะพอร์ทสำหรับการเกิดข้อผิดพลาดนี้จะแสดงออกที่ บิต 7, 4, 3, 2 และ 1 ดังนั้นค่าในรีจิสเตอร์ AX ไม่เป็นศูนย์จะแสดงถึงการเกิดข้อผิดพลาดทางอินพุต

tions Port) เป็นการคืนค่าสถานะพอร์ตสื่อสาร โดยการไหลดหมาย เลขพอร์ต
เข้าปอร์ตจิสเตอร์ DX การทำงานจะคืนค่าสถานะของสายสื่อสารในรีจิสเตอร์
AH และสถานะโมเด็มในรีจิสเตอร์ AL

AH (LINE STATUS)	AL (MODEM STATUS)
7 TIME OUT	7 RECEIVED LINE SIGNAL DETECT
6 TRAN SHIFT REGISTER EMPTY	6 RING INDICATOR
5 TRANS HOLD REGISTER EMPTY	5 DATA SET READY
AH (LINE STATUS)	AL (MODEM STATUS)
4 BREAK DETECT	4 CLEAR TO SEND
3 FRAMING ERROR	3 DELTA RECEIVE LINE SIGNAL DETECT
2 PARITY ERROR	2 TRAILING EDGE RING DETECTOR
1 OVERRUN ERROR	1 DELTA DATA SET READY
0 DATA READY	0 DELTA CLEAR TO SEND

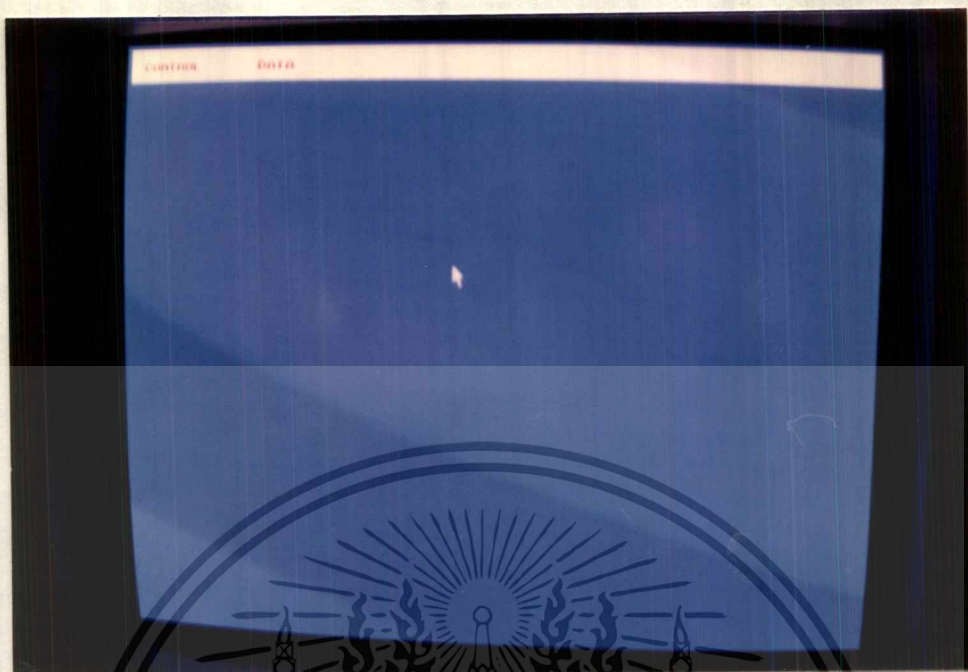
รูปที่ 3.13 ตารางการคืนสถานะของสายสื่อสาร

เครื่องคอมพิวเตอร์ที่ใช้ได้มีการโปรแกรมให้แสดง เมนู และข้อมูลต่าง ๆ
หน้าจอโมนิเตอร์ (MONITOR) เพื่อความสะดวกแก่ผู้ที่ต้องการใช้ระบบติดตามดาว
เทียมนี้ ซึ่งการแสดงผลที่หน้าจอโมนิเตอร์ หลังจากที่โปรแกรม เสร็จแล้วแสดงดัง
รูปที่ 3.14 และโปรแกรม เมนูมีรายละเอียดดังนี้

การทำงานเกี่ยวกับการควบคุมการรับมูมของสายอากาศ เป็น เมนูชื่อ
"CONTROL" แสดงรูปที่ 3.15 มีเมนูย่อยที่มีการทำงานต่าง ๆ ดังนี้

- การควบคุมรับค่ามูมของสายอากาศแบบควบคุมด้วยมือ เป็น เมนูชื่อ
"MANUAL" ใช้ในการควบคุมรับมูมอาชีมูม และมูมอี เล เวชั่นของสายอากาศโดย
ผ่านทาง เมาส์ แสดง รูปที่ 3.16 โดยการปรับ เปลี่ยนมูมนี้จะเป็นไปตามการกด

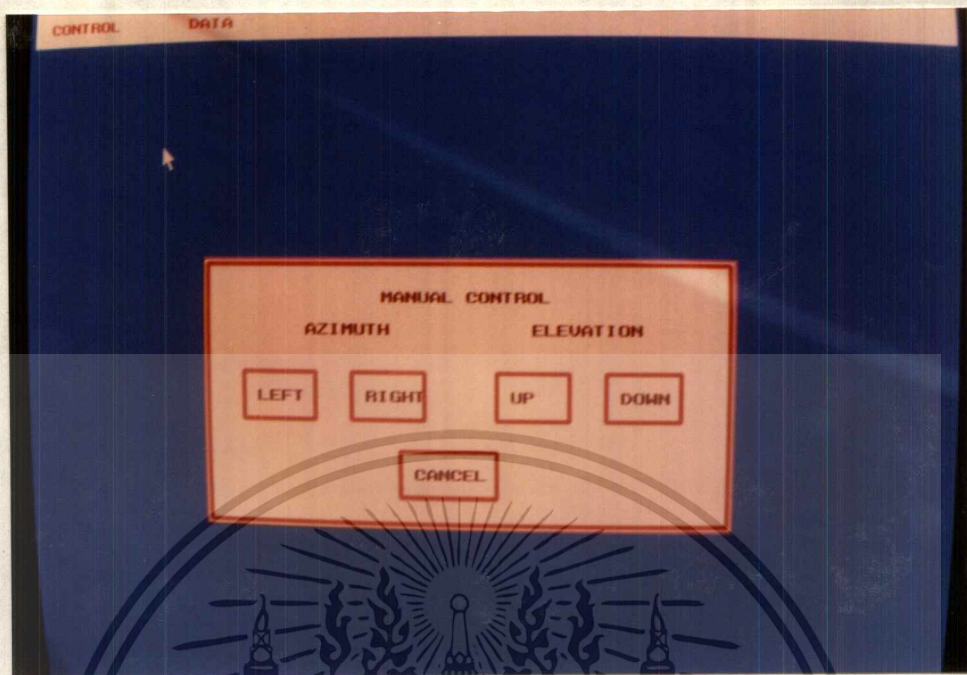
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.14 แสดง เมนูหน้าจอมอนิเตอร์



เอกสารนี้เป็นเอกสารสงวนลิขสิทธิ์ 3.15 แสดง เมนูย่อยหลังจากเลื่อนเมาส์ไปที่ CONTROL โยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



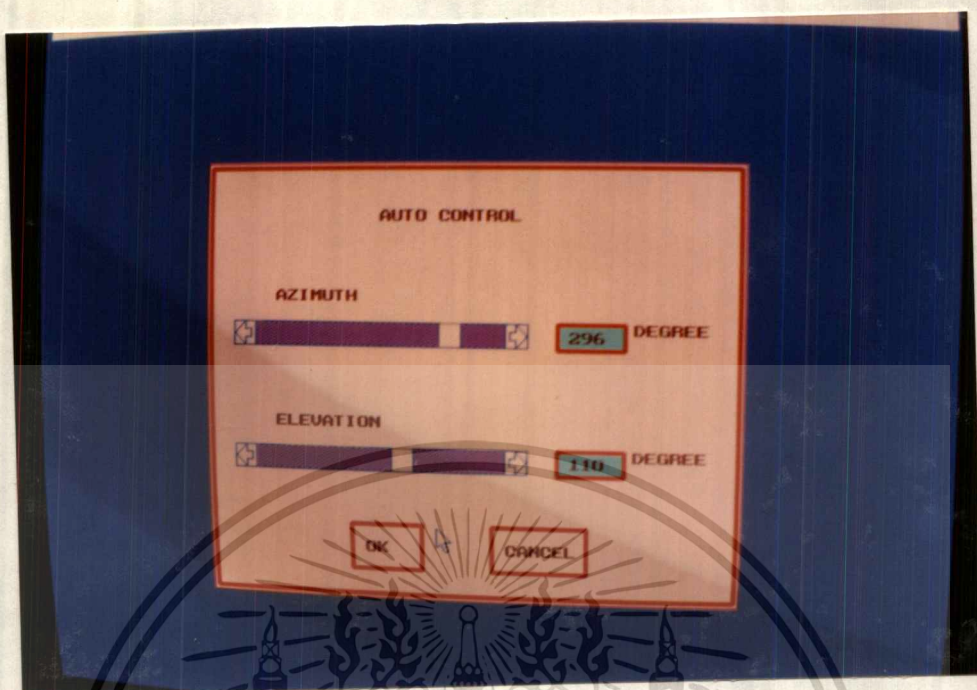
รูปที่ 3.16 แสดงการใช้ระบบ MANUAL

- การควบคุมรับมุมโดยอัตโนมัติ เป็นเมนูชื่อ "AUTOMATIC" ใช้ในการควบคุมปรับมุมอาซิมุท และมุมอีเลเวชันของสายอากาศ โดยการกำหนดตำแหน่งของมุมอาซิมุท และมุมอีเลเวชัน โดยผ่านทางเมาส์ แสดงดังรูป 3.17 จากนั้นก็จะส่งค่ามุมดังกล่าวไปยังควบคุมการปรับมุมของสายอากาศให้ซึ่งไปตามค่ามุมที่กำหนด

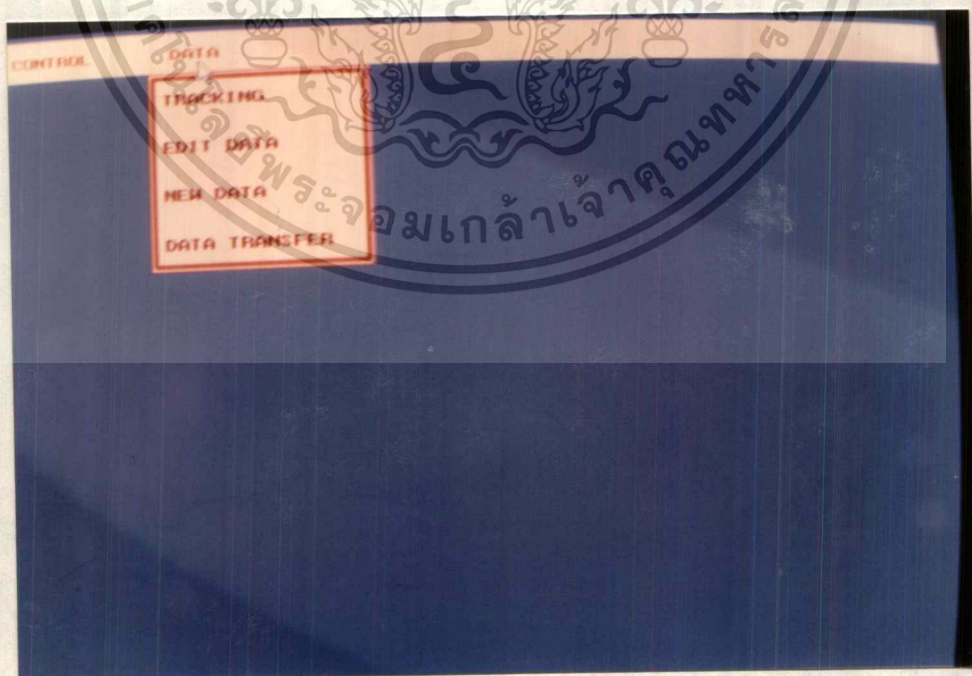
- การออกจากโปรแกรมติดตามดาวเทียม เป็นเมนูชื่อ "QUIT" ใช้เมื่อเรา ต้องการออกจากโปรแกรมติดตามดาวเทียม

การทำงานเกี่ยวกับข้อมูล เป็นเมนูชื่อ "DATA" แสดงดังรูป 3.18 ซึ่งจะมีเมนูย่อยที่มีการทำงานต่าง ๆ ดังนี้

- การแสดงข้อมูลในการติดตามดาวเทียม เป็นเมนูชื่อ "TRACKING" แสดงดังรูป 3.19 ใช้ในการแสดงผลข้อมูลต่างๆ ของดาวเทียมดวงที่เราต้องการติดต่อ โดยจะมีการแสดงข้อมูลเกี่ยวกับ ชื่อของดาวเทียม เวลาที่สามารถติดต่อกับดาวเทียมได้ ซึ่งเป็นข้อมูลเกี่ยวกับวันคริสต์มาส ปี ชั่ววง และวันที่ ที่ดาวเทียมกำลังทำงานอยู่ ไม่ว่าจะเป็นข้อมูลใดก็ตาม หากต้องการแก้ไขข้อมูล และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

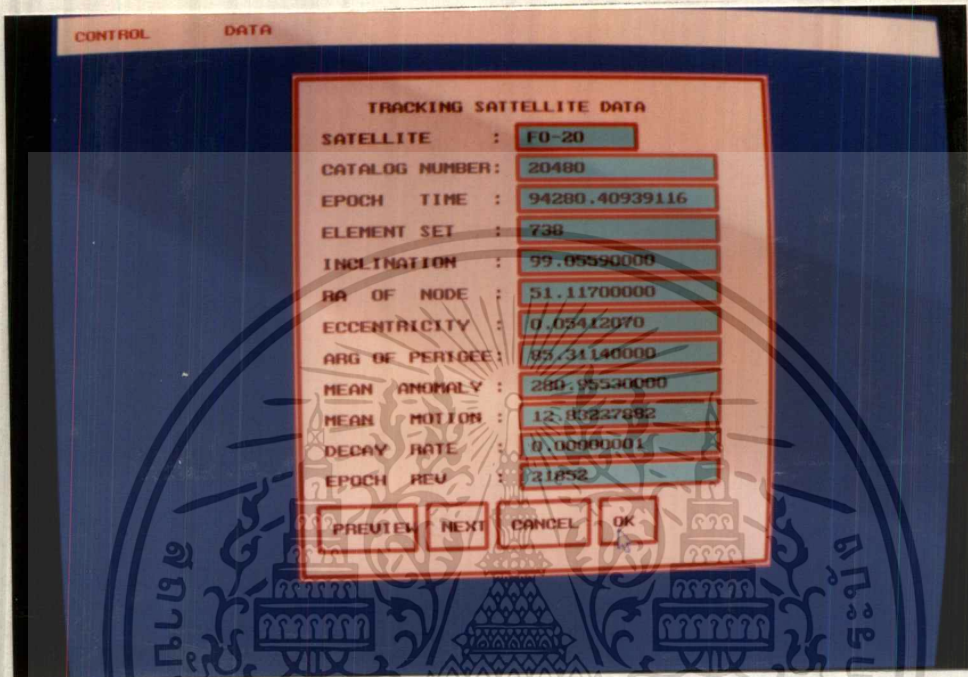


รูปที่ 3.17 แสดงการใช้ระบบ AUTOMATIC



เอกสารนี้เป็นเอกสารรูปที่ 3.18 แสดง เมนูย่อยหลังจากเลือกเมนูที่ DATA ระเบียบด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จะโคจรผ่านเข้ามาใกล้กับสถานีภาคพื้นดิน ค่ามุมอาซิมุท และมุมอีเลเวชันที่ดาวเทียมทำต่อสถานีภาคพื้นดิน ความสูงของดาวเทียม ระยะห่างระหว่างดาวเทียม และสถานีภาคพื้นดิน ฯลฯ



รูปที่ 3.19 แสดงการใช้เมนู TRACKING

- การรับข่าวสารวงโคจรของดาวเทียมจากผู้ใช้เป็นเมนูชื่อ "EDIT DATA" แสดงดังรูป 3.20 ใช้ในการรับข่าวสารวงโคจรของดาวเทียม (ORBITAL DATA ELEMENT) โดยการป้อนผ่านทางแป้นพิมพ์ และเมาส์

- การจัดเตรียมข้อมูลต่างๆ ที่จะส่งไปยังบอร์ดไมโครโปรเซสเซอร์ Z-80 เป็นเมนูชื่อ "NEW DATA" แสดงดังรูป 3.21 ใช้ในการจัดเตรียมข้อมูลที่จะส่งไปยังบอร์ดไมโครโปรเซสเซอร์ Z-80 โดยข้อมูลดังกล่าวจะเป็นข้อมูลเกี่ยวกับ เวลาที่เราจะสามารถติดต่อกับดาวเทียมได้ ค่ามุมอาซิมุท และมุมอีเลเวชันที่ดาวเทียมทำต่อสถานีภาคพื้นดิน

- การส่งข้อมูลที่ได้จากการประมวลผลของโปรแกรมไปยังบอร์ดไมโครโปรเซสเซอร์ Z-80 เป็นเมนูชื่อ "DATA TRANSFER" แสดงดังรูป 3.22 เอกสารประกอบฉบับนี้เป็นการนำเสนอขั้นตอนการทำงานเบื้องต้นในการคำนวณว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

EDIT SATELLITE ORBITAL ELEMENTS

SATELLITE : FO-20

CATALOG NUMBER:

EPOCH TIME :

ELEMENT SET :

INCLINATION :

RA OF NODE :

ECCENTRICITY :

ARG OF PERIGEE:

MEAN ANOMALY :

MEAN MOTION :

DECAY RATE :

EPOCH REV :

☒ CLEAR RECORD ☐ READ OLD DATA

PREV NEXT CANCEL SAVE

รูปที่ 3.20 แสดงการใช้เมนู EDIT DATA

Wed Apr 12 12:46:24 1995
SATELLITE NAME : FO-20
ELEMENT SET : 738
Thursday 13 Apr 95
ORBIT NUMBER : 24258

U.T.C.	AZ	EL	DOPPLER	RANGE	HEIGHT	LAT	LONG
HH:MM	deg	deg	Hz	km	km	deg	deg
04:50	25	3	2597	4280	1483	43	241
04:51	26	7	2559	3965	1504	40	243
04:52	29	10	2505	3656	1524	37	244
04:53	31	15	2428	3357	1544	34	245
04:54	35	19	2323	3071	1563	31	246
04:55	39	25	2180	2802	1582	28	247
04:56	44	30	1986	2558	1599	25	248
04:57	51	36	1727	2345	1616	22	249
04:58	61	43	1389	2174	1632	19	249
04:59	74	48	963	2055	1647	16	250
05:00	92	52	461	1998	1661	13	250
05:01	111	52	-83	2009	1674	10	251
05:02	129	49	-616	2065	1686	8	252
05:03	142	44	-1093	2219	1697	5	253
05:04	152	38	-1485	2402	1707	2	253
05:05	159	32	-1791	2623	1716	-1	254
05:06	165	27	-2021	2872	1723	-4	255
05:07	169	22	-2191	3142	1730	-7	256
05:08	172	18	-2315	3427	1735	-10	257
05:09	174	14	-2405	3723	1740	-13	258
05:10	176	10	-2469	4027	1743	-15	258
05:11	178	7	-2513	4337	1745	-18	259
05:12	179	4	-2542	4650	1745	-21	259
05:13	180	1	-2560	4965	1745	-24	260

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์รูปที่ 3.21 แสดงการใช้เมนู NEW DATA ให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ข้อมูลดังกล่าวคือ ค่าของ เวลา มุมอาซิมุท และมุมอีเลเวชัน ที่สายอากาศควร
จะชี้ไปในขณะเวลานั้น ๆ

AZ= 304 (30 1)	EL= 22 (16 0)
AZ= 292 (24 1)	EL= 26 (1a 0)
AZ= 277 (15 1)	EL= 28 (1c 0)
AZ= 261 (5 1)	EL= 28 (1c 0)
AZ= 246 (f6 0)	EL= 26 (1a 0)
AZ= 234 (ea 0)	EL= 22 (16 0)
AZ= 224 (a0 0)	EL= 18 (12 0)
AZ= 217 (d9 0)	EL= 13 (d 0)
AZ= 211 (d3 0)	EL= 9 (9 0)
AZ= 207 (cf 0)	EL= 6 (6 0)
AZ= 203 (cb 0)	EL= 2 (2 0)
SATELLITE NAME: FO-20	
DAY 12 (c 0)	MONTH 4 (4 0) YEAR 95 (5f 0)
HOUR: 16 (10 0)	MIN: 53 (35 0)
AZ= 157 (9d 0)	EL= 3 (3 0)
AZ= 155 (9b 0)	EL= 6 (6 0)
AZ= 153 (99 0)	EL= 11 (b 0)
AZ= 151 (97 0)	EL= 16 (10 0)
AZ= 147 (93 0)	EL= 23 (17 0)
AZ= 140 (8c 0)	EL= 32 (20 0)
AZ= 127 (7f 0)	EL= 44 (2c 0)
HOUR: 17 (11 0)	MIN: 0 (0 0)
AZ= 99 (63 0)	EL= 55 (37 0)
AZ= 55 (37 0)	EL= 56 (38 0)
AZ= 26 (1a 0)	EL= 44 (2c 0)
AZ= 12 (c 0)	EL= 32 (20 0)
AZ= 5 (5 0)	EL= 23 (17 0)
AZ= 1 (1 0)	EL= 16 (10 0)
AZ= 358 (66 1)	EL= 10 (a 0)
AZ= 357 (65 1)	EL= 6 (6 0)
AZ= 335 (63 1)	EL= 2 (2 0)
HOUR: 18 (12 0)	MIN: 46 (2a 0)

รูปที่ 3.22 แสดงการใช้เมนู DATA TRANSFER

ตารางคุณสมบัติ ส เตปปั๊มมอ เตอร์ และ ชุด เฟือง	
แรงดันไฟตรงอินพุท	3.4 โวลท์
กระแสไฟตรงอินพุท	2.9 แอมป์
กำลังอินพุทสูงสุด	9.86 วัตต์
กำลัง เอาท์พุทสูงสุด	7.88 วัตต์
ประสิทธิภาพ	80 %
แรงบิดที่ 1 รอบ/นาที	1505.89 นิวตัน- เมตร
แรงบิดที่ 2 รอบ/นาที	752.19 นิวตัน- เมตร
องศาต่อส เตปที่มอ เตอร์	1.8 องศา/ส เตป
อัตราทด เฟือง	20:1
องศาต่อส เตปที่สายอากาศ	0.09 องศา/ส เตป
เวลาต่อองศา	1.5 วินาที/องศา
น้ำหนักฐานรวม เฟือง	31 กิโลกรัม

ตารางการใช้กระแสไฟตรง ที่แรงดันไฟตรง 5 โวลท์	
ชุดไมโครโปร เซส เซอร์ Z-80	200 มิลลิแอมป์แปร์
ชุดซิปส เตะปั้งมอ เตอร์	80 มิลลิแอมป์แปร์
ชุด เซ็น เซอร์ มุมของสายอากาศ	560 มิลลิแอมป์แปร์
บอร์ด RS-232	240 มิลลิแอมป์แปร์
บอร์ด RTC	100 มิลลิแอมป์แปร์
รวม	1180 มิลลิแอมป์แปร์

รูปที่ 3.24 ตารางแสดงการใช้กระแสไฟตรงของบอร์ดต่างๆ

บทที่ 4

บทสรุป

งานการติดตามดาวเทียมของระบบควบคุมสายอากาศนี้ จะนำค่าข่าวสารที่เกี่ยวข้องกับการโคจรของดาวเทียม มาป้อนเข้าไปในเครื่องคอมพิวเตอร์ เพื่อทำการประมวลผล แล้วให้ผลข้อมูลซึ่งประกอบไปด้วย ค่ามุมอาซิมุท และค่ามุมอีเลเวชัน ที่เวลาใดๆ จากนั้นก็จะส่งข้อมูลดังกล่าวไปไปยังเครื่องติดตามดาวเทียมที่มีไมโครโปรเซสเซอร์ z-80 เป็นตัวควบคุม โดยใช้การเข้าเมนู "DATA TRANSFER" ของคอมพิวเตอร์ ข้อมูลดังกล่าวจะเป็นของดาวเทียมดวงที่ต้องการจะติดตามซึ่งมีทั้งหมด 9 ดวง โดยในแต่ละดวงจะมี ข้อมูลที่ใช้ในการติดตามประมาณ 2 วัน เมื่อต้องการให้เครื่องติดตามดาวเทียม ติดตามดาวเทียมดวงใด ให้เลือกเลข 1 จากจอแอลซีดี ซึ่งเป็นเมนู "AUTO TRACKING" จากนั้นเครื่องติดตามดาวเทียมจะถามหมายเลขดาวเทียมดวงที่ต้องการจะติดตาม เมื่อใส่หมายเลขดาวเทียมแล้ว เครื่องก็จะทำการติดตามดาวเทียมตามค่าของข้อมูลที่ถูกส่งมาจากเครื่องคอมพิวเตอร์ ถ้าต้องการทำการควบคุมการปรับมุมของสายอากาศ แบบควบคุมด้วยตนเองก็ให้เลือกเลข 2 จากจอแอลซีดี ซึ่งเป็นเมนู "MANUAL CONTROL" การปรับมุมสายอากาศทำได้โดยการกดคีย์ต่างๆ ซึ่งมีลูกศรชี้ทิศทางของการปรับมุม ส่วนการตั้ง เวลาของเครื่องติดตามดาวเทียมทำได้โดยการเลือกเลข 3 จากจอแอลซีดี จากนั้นเครื่องก็จะถามว่าต้องการตั้ง วันที่ หรือ เวลา โดยการกดเลข 1 หมายถึงการตั้งวันที่ และการกดเลข 2 หมายถึงการตั้ง เวลา เมื่อตั้ง เวลา เรียบร้อยแล้วก็ให้กดคีย์ "ENTER"

การควบคุมสายอากาศจากเครื่องคอมพิวเตอร์ทำได้โดย การเลือกเมนู "CONTROL" จากนั้นก็จะมีเมนูย่อยให้เลือกว่า ต้องการควบคุมการปรับมุมสายอากาศด้วยตนเอง หรือแบบอัตโนมัติ โดยแบบควบคุมการปรับมุมของสายอากาศด้วยตนเองจะมีปุ่มให้เลือกว่า ต้องการที่จะปรับมุมไปทางซ้าย หรือทางขวา ปรับขึ้น หรือ ปรับลง ส่วนในแบบอัตโนมัตินั้นจะมีแถบเลื่อนที่สามารถ เปลี่ยนได้โดยการเข้าเมาส์ ซึ่งแถบเลื่อนดังกล่าวมี 2 อัน โดยการเลื่อนแถบในแต่ละ

อันนี้จะเป็นการเลือกค่ามุมอาซิมุท และมุมอีเลเวชัน เมื่อเลือกค่ามุมที่ต้องการแล้ว ไม่ว่าจะวิธีใดทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หนังสืออ้างอิง

- [1] Martin Davidoff " THE SATELLITE EXPERIMENTER'S HANDBOOK"
: The American Radio Relay League 225 Main Street,
Newington,CT 06111
- [2] Herberg Sheldle "TURBO C/C++ COMPLETE REFERENCE"
: MCGRAW-HILL
- [3] Steve Rimer "GRAPHICS USER INTERFACE"
: MCGRAW-HILL
- [4] ปราโมทย์ วาดเขียน และ วิวัฒน์ กิรานนท์ "พื้นฐานการสื่อสารข้อมูล"
กรุงเทพฯ
: สำนักพิมพ์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง 2536
- [5] วิบูลย์ ชื่นแขก "ไมโครโปรเซสเซอร์" กรุงเทพฯ
: สำนักพิมพ์ สถาบันเทคโนโลยีพระจอมเกล้าพระนครเหนือ 2532
- [6] พงษ์วุฒิ สิทธิพล และ ชิตี ธาตรินรานนท์ " ทฤษฎีและงานปฏิบัติส่งกำลัง 2 "
:สำนักพิมพ์ สถาบันเทคโนโลยีราชมงคลวิทยาเขตนนทบุรี



ภาคผนวก ก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

#include <dos.h>
#include <stdlib.h>
#include <graphics.h>
#include <stdio.h>
#include <ctype.h>

#define DROPSHADOW 10
#define ACTIVE      0xffff
#define INACTIVE    0xaa55
#define FONTDEEP 8
#define errorbeep() putchar(7)
#define innothing    0x0000
#define inwindow     0x0001
#define inmenubar    0x0002
#define inbutton     0x0004
#define inbitmap     0x0008
#define intext       0x0010
#define intextfield  0x0020
#define inlist       0x0040
#define invertscroll 0x0080
#define inhorscroll  0x0100
#define inuparrow    0x0200
#define indownarrow  0x0400
#define injumpup     0x0800
#define injumpdown   0x1000
#define inthumb      0x2000
#define incheckbox     0x4000
#define ineditfield  0x8000

```

```

#define pixels2bytes(n) ((n+7)/8)

```

```
#define CHECKMARK 251
#define ACTIVE      0xffff
#define INACTIVE    0xaa55
#define KEYMENUCHAR 4
#define MAXMENUITEM 16
#define MENULINESIZE 16
#define MENULINEDEEP 30
#define MENUCOUNT 2
#define MENUTITLESIZE 12
#define MENUBARDEEP 30
#define SCREENWIDE 640
#define SCREENDEEP 480
#define DEL          83*256
#define HOME         71*256
#define CURSOR_LEFT  75*256
#define CURSOR_RIGHT 77*256
#define END          79*256
#define ALT_Q        0x1000
#define ALT_W        0x1100
#define ALT_E        0x1200
#define ALT_R        0x1300
#define ALT_T        0x1400
#define ALT_Y        0x1500
#define ALT_U        0x1600
#define ALT_I        0x1700
#define ALT_O        0x1800
```

```
#define ALT_P        0x1900
```

```
#define ALT_A        0x1E00
```

```

#define ALT_S 0x1F00
#define ALT_D 0x2000
#define ALT_F 0x2100
#define ALT_G 0x2200
#define ALT_H 0x2300
#define ALT_J 0x2400
#define ALT_K 0x2500
#define ALT_L 0x2600
#define MREC 50
#define MAX 9
typedef struct{
    unsigned int type;
    void * next;
} OBJECTHEAD;
typedef struct{
    int left,top,right,bottom;
} RECT;
typedef struct{
    int x,y;
} POINT;
typedef struct{
    OBJECTHEAD head;
    RECT frame;
    char * back;
} WINDOW;
typedef struct{

```

```

    char *text;

    unsigned int active;

}BUTTON;

typedef struct{
    OBJECTHEAD head;

    int x,y;

    RECT frame;

    unsigned int active;

    char *text;

    char select;

}CHECKBOX;

typedef struct{
    OBJECTHEAD head;

    int x,y;

    int size;

    RECT frame,uparr,dnarr,thumb,jump;

    unsigned int active;

    int min,max,cur;

}SCROLLBAR;

typedef struct{
    char name[MENULINESIZE+1];

    int (*proc)();

}MENUITEM;

typedef struct{
    int count;

    char title[MENUTITLESIZE+1];

    MENUITEM item[MAXMENUITEM];

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
typedef struct {
    OBJECTHEAD head;
    int x,y;
    char *text;
    unsigned int active;
}TEXT;
```

```
typedef struct{
    OBJECTHEAD head;
    int x,y;
    int length;
    RECT frame;
    char *text;
    unsigned int active;
}TEXTFIELD;
```

```
typedef struct{
    OBJECTHEAD head;
    int x,y;
    int length;
    int curpos;
    int (*proc)();
    char select;
    RECT frame;
    char *text;
    unsigned int active;
}EDITFIELD;
```

```
typedef struct{
```

เอกสารนี้เป็นเอกสารลิขสิทธิ์ของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
 ไม่ควรนำเอกสารนี้ไปใช้ในการศึกษาหรือนำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น หากมีข้อผิดพลาดประการใด ขออภัยและต้องอภัยถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    int az,el;

    }REC;

typedef struct{
    int id;
    char *name;
    long int cata;
    double epocht;
    int element;
    double inclination;
    double ranode;
    double eccen;
    double arg;
    double meana;
    double meanm;
    double decay;
    long int epochrev;
}SAT;

SAT sat[MAX];

    char *name[9]={"RS-10/11","RS-12/13","FO-20","AO-10",
    "AO-13","AO-21","AO-27","DO-17","MIR"};

    int cata[9]={18129,21089,20480,14129,19216,
21087,22825,20440,16609};

    char buf[10][15];

    char bufn[15];

REC rec[MREC];

char br[3];

```

```

char btd[3][3];
char baz[3][4];
char bel[3][4];
int inp,n;
BUTTON * findbutton(WINDOW *w,POINT *p);
CHECKBOX *findcheckbox(WINDOW *w,POINT *p);
SCROLLBAR *findscrollbar(WINDOW *w,POINT *p);
char *ega2mono(char *source);
char *mono2ega(char *source);
char scrollbarfill[]="\063\314\063\314\063\314\063\314";
int new=0;
POINT fakemousedown;
int menuindex=0,alive=1,curpos;
MENU *menuarray[MENUCOUNT];
WINDOW screenwindow;
RECT menubarrect;
FILE *fpl;
unsigned int altkeytable[]={
    ALT_Q,ALT_W,ALT_E,ALT_R,
    ALT_T,ALT_Y,ALT_U,ALT_S,
    0xffff
};
int stepcontrol(void);
int autocontrol(void);
int doquit(void);
int readdata(void);
int editdata(void);
int outdata(void);

```

```

int check(void);
int rport(void);
int track(void);
int process(void);
MENU    mol={ 3," CONTROL ",
              " MANUAL ",stepcontrol,
              " AUTOMATIC ",autocontrol,
              " QUIT ",doquit,
};
MENU    mo2={ 4," DATA ",
              " TRACKING ",readdata,
              " EDIT DATA ",editdata,
              " NEW DATA ",process,
              " DATA TRANSFER ",outdata,
};

```

```

main()

```

```

{

```

```

    POINT thepoint;

```

```

    int rt;

```

```

    if(initmouse()){

```

```

        if(initgraphics()){

```

```

            init();

```

```

            mouseon();

```

```

            initwindowmanager();

```

```

            initmenumanager();

```

```

            addmenu(&mol);

```

```

            addmenu(&mo2);

```

```

            drawmenubar();

```

```

while(alive){
    dispatchmenuitem();
    if(mousedown(&thepoint)){
rt=wheremouse(&thepoint,&screenwindow);
if(rt&inmenubar){domenu(&thepoint);drawmenubar();}
    }
}
mouseoff();
deinitgraphics();
    }else puts(" Cannot establish graphics mode");
    }else puts(" Cannot locate mouse driver");
}

/*****
*           WINDOW FUNCTION           *
*****/
initgraphics()
{
    int d,m;
    detectgraph(&d,&m);
    if(d<0)return(0);
    initgraph(&d,&m,"");
    if(graphresult()<0)return (0);
    setbkcolor(LIGHTBLUE);
}

deinitgraphics()

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่าจะ closegraph(); ห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    while(kbhit())getch();
}

openwindow(w,r)

    WINDOW *w;

    RECT *r;

}

int size;

w->frame.left=r->left;
w->frame.top=r->top;
w->frame.right=r->right;
w->frame.bottom=r->bottom;
if((size=bufferize(&w->frame))==-1)return 0;
w->head.next=NULL;
w->head.type=inwindow;
if((w->back=malloc(size))!=NULL){
    mouseoff();

    getimage(w->frame.left,w->frame.top,w->frame.right,
        w->frame.bottom,w->back);

    setwritemode(COPY_PUT);

    setfillstyle(SOLID_FILL,WHITE);

    setlinestyle(SOLID_LINE,0,NORM_WIDTH);

    setcolor(LIGHTRED);

    bar(r->left,r->top,r->right,r->bottom);

    rectangle(r->left,r->top,r->right,r->bottom);

    setlinestyle(SOLID_LINE,0,THICK_WIDTH);

    rectangle(r->left+3,r->top+3,r->right-3,r->bottom-3);

    mouseon();

    setcolor(RED);

```

```

        return 1;
    }else return 0;
}

buffersize(r)
    RECT *r;
{
    return (imagesize(r->left,r->top,r->right,r->bottom));
}

setrect(r,left,top,right,bottom)
    RECT *r;
    int left;
    int top;
    int right;
    int bottom;
{
    r->left=left;
    r->top=top;
    r->right=right;
    r->bottom=bottom;
}

closewindow(w)
    WINDOW *w;
{
    mouseoff();

    if(w->back!=NULL){
        putimage(w->frame.left,w->frame.top,w->back,COPY_PUT);
        free(w->back);
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    mouseon();
}

/*****

*          MOUSE FUNCTION          *
*****/

mouseon()
{
    union REGS r;
    r.x.ax=1;
    int86(0x33,&r,&r);
}

mouseoff()
{
    union REGS r;
    r.x.ax=2;
    int86(0x33,&r,&r);
}

initmouse()
{
    union REGS r;
    r.x.ax=0;
    int86(0x33,&r,&r);
    return r.x.ax;
}

setupmouse(mx,my,x,y)

    int mx,my,x,y;

```

```

union REGS r;

r.x.ax=0x007;

r.x.cx=mx;

r.x.dx=x;

int86(0x33,&r,&r);

r.x.ax=0x008;

r.x.cx=my;

r.x.dx=y;

int86(0x33,&r,&r);
}
arrowcursor()
{
union REGS r;
struct SREGS sr;
static char cursor[]={
    0xff,0x3f,0xff,0x1f,
    0xff,0x0f,0xff,0x07,
    0xff,0x03,0xff,0x01,
    0xff,0x00,0x7f,0x00,
    0x3f,0x00,0x1f,0x00,
    0xff,0x01,0xff,0x10,
    0xff,0x30,0x7f,0xf8,
    0x7f,0xf8,0x7f,0xfc,
    0x00,0x00,0x00,0x40,
    0x00,0x60,0x00,0x70,
    0x00,0x78,0x00,0x7c,
    0x00,0x7e,0x00,0x7f,
    0x80,0x7f,0x00,0x7c,

```

```

        0x00,0x6c,0x00,0x46,
        0x00,0x06,0x00,0x03,
        0x00,0x03,0x00,0x00 };

r.x.ax=0x0009;

r.x.bx=0;

r.x.cx=0;

r.x.dx=FP_OFF(cursor);
sr.es=FP_SEG(cursor);
int86x(0x33,&r,&r,&sr);
}

mousedown(p)
POINT *p;
{
    union REGS r;
    if(fakemousedown.x!=-1&&fakemousedown.y!=-1){
        memcpy((char*)p,(char *)&fakemousedown,sizeof(POINT));
        fakemousedown.x=-1;
        fakemousedown.y=-1;
        return 1;
    }

    r.x.ax=3;

    int86(0x33,&r,&r);

    p->x=r.x.cx;
    p->y=r.x.dx;

    return (r.x.bx&0x01);
}

```

```

{
    union REGS r;

    r.x.ax=3;

    int86(0x33,&r,&r);

    p->x=r.x.cx;

    p->y=r.x.dx;

}

/*****
*
*          BOTTON FUNCTION
*
*****/
addbutton(w,b,tx,active)
    WINDOW *w;
    BUTTON *b;
    char *tx;
    int active;
{
    OBJECTHEAD *oh;
    oh=(OBJECTHEAD *)w;
    while(oh->next!=NULL)oh=oh->next;
    oh->next=(OBJECTHEAD *)b;
    b->text=tx;
    b->active=active;
    b->head.type=inbutton;
    b->head.next=NULL;
    b->frame.left&=0xffff8;
    b->frame.right&=0xffff8;

    drawbutton(b);
}

```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

drawbutton(b)
    BUTTON *b;
{
    mouseoff();
    setwritemode(COPY_PUT);
    setfillstyle(SOLID_FILL,getmaxcolor());
    setlinestyle(SOLID_LINE,0,THICK_WIDTH);
    setcolor(LIGHTRED);
    bar(b->frame.left,b->frame.top,b->frame.right,
        b->frame.bottom);
    rectangle(b->frame.left,b->frame.top,b->frame.right,
        b->frame.bottom);
    setcolor(RED);
    outtextxy(b->frame.left+10,b->frame.top+12,b->text);
    mouseon();
}
BUTTON * findbutton(WINDOW *w,POINT *p)
{
    OBJECTHEAD *oh;
    BUTTON *b;
    oh=(OBJECTHEAD *)w;
    while(oh!=NULL){
        if(oh->type==inbutton){
            b=(BUTTON *)oh;
            if(pointinrect(p,&b->frame)&&b->active==ACTIVE)
                return b;
            oh=oh->next;
        }
    }
    return NULL;
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามแก้ไขตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

}

trackbutton(b)
{
    BUTTON *b;

    {
        POINT p;

        if(b!=NULL&&b->head.type==inbutton&&b->active==ACTIVE){

            invertrect(&b->frame);

            while(mousedown(&p));

            invertrect(&b->frame);

        }else errorbeep();
    }
}

/*****
*          BASIC FUNCTION          *
*****/

invertrect(r)
{
    RECT *r;

    {

        char *p;

        int size;

        if((size=imagesize(r->left+2,r->top+2,r->right-2,
            r->bottom-2))!=-1){

            if((p=malloc(size))!=NULL){

                mquseoff();

                getimage(r->left+2,r->top+2,r->right-2,r->bottom-2,p);

                putimage(r->left+2,r->top+2,p,NOT_PUT);

                mouseon();

                free(p);
            }
        }
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    }

}

wheremouse(p,w)

    POINT *p;

    WINDOW *w;

{
    EDITFIELD *f;
    BUTTON *b;
    CHECKBOX *cb;
    SCROLLBAR *sb;
    TEXTFIELD *tf;
    OBJECTHEAD *oh;
    int r=innothing,i;
    if(pointinrect(p,&menubarrect))return(inmenubar);
    else{
oh=(OBJECTHEAD *)w;
if(!pointinrect(p,&w->frame))return(innothing);
else{
    while(oh!=NULL){
        switch(oh->type){
case inbutton:
            b=(BUTTON *)oh;
            if(pointinrect(p,&b->frame))r!=inbutton;
            break;
case inwindow:
            r=inwindow;
            break;
case intextfield:

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามแก้ไขเปลี่ยนแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        tf=(TEXTFIELD *)oh;

        if(pointinrect(p,&tf->frame))r!=inlist;

        break;

case incheckbox:

        cb=(CHECKBOX *)oh;

        if(pointinrect(p,&cb->frame))

' r!=incheckbox;

        break;

case invertscroll:

        sb=(SCROLLBAR *)oh;

        if(pointinrect(p,&sb->frame)){

r!=invertscroll;

if(pointinrect(p,&sb->uparr))

        r!=inuparrow;

if(pointinrect(p,&sb->dnarr))

        r!=indownarrow;

else if(pointinrect(p,&sb->thumb))

        r!=inthumb;

else{

        if(p->y<sb->thumb.top)r!=injumpup;

        else r!=injumpdown;

}

        }

        break;

case inhorscroll:

        sb=(SCROLLBAR *)oh;

        if(pointinrect(p,&sb->frame)){

```

```

if(pointinrect(p,&sb->uparr))
    r!=inuparrow;
else if(pointinrect(p,&sb->dnarr))
    r!=indownarrow;
else if(pointinrect(p,&sb->thumb))
    r!=inthumb;
else {
    if(p->x<sb->thumb.left)
        r!=injumpup;
    else r!=injumpdown;
}

break;
case ineditfield:
    f=(EDITFIELD *)oh;
    if(f->select){
        f->select=0x00;
        draweditfield(f);
    }

    if(pointinrect(p,&f->frame)){
        r!=ineditfield;
        f->curpos=(p->x-f->frame.left)/8;
        if(f->curpos>(i=strlen(f->text)))
            f->curpos=i;
    }

    break;
}

```

```

    }

    return (r);
}

}

}

keyboard(w)

    WINDOW *w;

{
    OBJECTHEAD *oh;
    BUTTON *bp;
    EDITFIELD *f;
    int c,len,i,bcount=0;
    if(kbhit()){
        c=getkey();
        oh=(OBJECTHEAD *)w;
        while(oh!=NULL){
            if(oh->type==inbutton){
                bp=(BUTTON *)oh;
                ++bcount;

                if(bcount==1&&c==13){
                    fakemousedown.x=bp->frame.left+1;
                    fakemousedown.y=bp->frame.top+1;
                }

                else if(bcount==2&&c==27){
                    fakemousedown.x=bp->frame.left+1;
                    fakemousedown.y=bp->frame.top+1;
                }
            }
        }
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

if(oh->type==ineditfield){
    f=(EDITFIELD *)oh;
    if(f->select){
len=strlen(f->text);
switch(c){
    case DEL:
if(f->curpos<len){
    for(i=f->curpos;i<=len;++i)
        f->text[i]=f->text[i+1];
    }
break;
    case HOME:
f->curpos=0;
break;
    case END:
f->curpos=len;
break;
    case CURSOR_LEFT:
if(f->curpos) --f->curpos;
break;
    case CURSOR_RIGHT:
if(f->curpos<len)++f->curpos;
break;
    case 'X'-0x40:
f->curpos=0;
f->text[0]=0;
break;

```

```

{

    if(f->curpos&&len){

        if(f->curpos<len){

            for(i=f->curpos;i<=len;++i)

                f->text[i-1]=f->text[i];

            --f->curpos;

        }

        else{

            --f->curpos;

            f->text[len-1]=0;

        }

    }

    break;

    default:

        if(f->proc!=NULL)c=(f->proc)(c);

        else{

            if(!isprint(c))c=0;

        }

        if(c&&(len+1)<=f->length){

            for(i=(len+1);i>f->curpos;--i)

                f->text[i]=f->text[i-1];

            f->text[f->curpos]=c;

            ++f->curpos;

        }

        break;

    }

    draweditfield(f);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    oh=oh->next;

    }

}

getkey()
{
    int c;
    c=getch();
    if(!(c&0x00ff))c=getch()<<8;
    return c;
}

pointinrect(p,r)
POINT *p;
RECT *r;
{
    if(p->x>r->left && p->x<r->right && p->y>r->top &&
        p->y<r->bottom) return 1;
    else return 0;
}

/*****

*,          CHECKBOX FUNCTION          *

```

```

*****/

addcheckbox(w,cb,x,y,text,select,active)

WINDOW *w;

```

```

CHECKBOX *cb;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ หากผู้ใดต้องการนำเอกสารนี้ไปใช้โดยไม่ได้รับอนุญาตจากเจ้าของเอกสาร กรุณาแจ้งเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

int y;
char *text;
int select;
int active;
{
    OBJECTHEAD *oh;
    oh=(OBJECTHEAD *)w;
    while(oh->next!=NULL)oh=oh->next;
    oh->next=(OBJECTHEAD *)cb;
    cb->x=x;
    cb->y=y;
    cb->text=text;
    cb->select=select;
    cb->active=active;
    cb->frame.left=x-1;
    cb->frame.top=y-1;
    cb->frame.right=x+10;
    cb->frame.bottom=y+10;
    cb->head.type=incheckbox;
    cb->head.next=NULL;
    drawcheckbox(cb);
}
drawcheckbox(cb)
    CHECKBOX *cb;
{
    mouseoff();

```

```

setlinestyle(SOLID_LINE,0,NORM_WIDTH);
setcolor(12);
bar(cb->frame.left,cb->frame.top,cb->frame.right,
    cb->frame.bottom);
rectangle(cb->frame.left,cb->frame.top,cb->frame.right,
    cb->frame.bottom);
outtextxy(cb->frame.right+16,cb->frame.top+2,cb->text);
if(cb->select){
    line(cb->frame.left,cb->frame.top,cb->frame.right,
        cb->frame.bottom);
    line(cb->frame.left,cb->frame.bottom,
        cb->frame.right,cb->frame.top);
}
mouseon();
}
CHECKBOX *findcheckbox(WINDOW *w,POINT *p)
{
    OBJECTHEAD *oh;
    CHECKBOX *cb;
    oh=(OBJECTHEAD *)w;
    while(oh!=NULL){
        if(oh->type==incheckbox){
            cb=(CHECKBOX *)oh;
            if(pointinrect(p,&cb->frame)&&cb->active==ACTIVE)
                return cb;
        }
        oh=oh->next;
    }
}

```

```

oh=(OBJECTHEAD *)w;
while(oh->next!=NULL)oh=oh->next;
oh->next=(OBJECTHEAD *)sb;
sb->x=x;
sb->y=y;
sb->size=len;
sb->min=min;
sb->max=max;
sb->cur=cur;
sb->active=active;
sb->frame.left=x;
sb->frame.top=y;
sb->frame.right=x+len;
sb->frame.bottom=y+15;
setrect(&sb->uparr,sb->frame.left,sb->frame.top,
        sb->frame.left+15,sb->frame.bottom);
setrect(&sb->dnarr,sb->frame.right-15,sb->frame.top,
        sb->frame.right,sb->frame.bottom);
setrect(&sb->thumb,sb->frame.left+16,sb->frame.top,
        sb->frame.left+32,sb->frame.bottom);
setrect(&sb->jump,sb->uparr.right,sb->frame.top,
        sb->dnarr.left,sb->frame.bottom);
sb->head.type=inhorscroll;
sb->head.next=NULL;
drawhorscroll(sb);

```

}

```

        return NULL;
    }

trackcheckbox(cb)
    CHECKBOX *cb;
{
    POINT p;
    mousetloc(&p);
    if(pointinrect(&p,&cb->frame)){
        invertrect(&cb->frame);
        while(mousedown(&p));
        invertrect(&cb->frame);
    }
}

/*****
 *          SCROLLBAR FUNCTION          *
 *****/
addhorscroll(w,sb,x,y,len,min,max,cur,active)
    WINDOW *w;
    SCROLLBAR *sb;
    int x;
    int y;
    int len;
    int min;
    int max;
    int cur;
    int active;

```

{

```
static char leftarrow[]={ 0x0f,0x00,0x0f,0x00,  
    0x00,0x00,0x7e,0x7e,0x7d,0x7e,0x7b,0x7e,  
    0x77,0x7e,0x6f,0x06,0x5f,0xf6,0x3f,0xf6,  
    0x3f,0xf6,0x5f,0xf6,0x6f,0x06,0x77,0x7e,  
    0x7b,0x7e,0x7d,0x7e,0x7e,0x7e,0x00,0x00 };
```

```
static char rightarrow[]={ 0x0f,0x00,0x0f,0x00,  
    0x00,0x00,0x7e,0x7e,0x7e,0xbe,0x7e,0xde,  
    0x7e,0xee,0x60,0xf6,0x6f,0xfa,0x6f,0xfc,  
    0x6f,0xfc,0x6f,0xfa,0x60,0xf6,0x7e,0xee,  
    0x7e,0xde,0x7e,0xbe,0x7e,0x7e,0x00,0x00 };
```

```
char *p;
```

```
mouseoff();
```

```
if(sb->active==ACTIVE)setfillpattern(scrollbarfill,13);
```

```
else setfillstyle(SOLID_FILL,getmaxcolor());
```

```
setwritemode(COPY_PUT);
```

```
setlinestyle(SOLID_LINE,0,NORM_WIDTH);
```

```
setcolor(BLACK);
```

```
bar(sb->frame.left,sb->frame.top,sb->frame.right,  
    sb->frame.bottom);
```

```
rectangle(sb->frame.left,sb->frame.top,sb->frame.right,  
    sb->frame.bottom);
```

```
if(ismonomode())
```

```
    putimage(sb->uparr.left,sb->uparr.top,leftarrow,  
        COPY_PUT);
```

```
else{
```

```
    if((p=mono2ega(leftarrow))!=NULL){
```

เอกสารนี้เป็นเอกสารลิขสิทธิ์ของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ไม่ว่ากรณีใดๆ不得复制或传播เอกสารนี้ หรือนำไปใช้ประโยชน์ด้านการค้า

```
        putimage(sb->uparr.left,sb->uparr.top,p,COPY_PUT);
```

```

free(p);

    }else errorbeep();

}

if(ismonomode())

    putimage(sb->dnarr.left,sb->dnarr.bottom-15,

        rightarrow,COPY_PUT);

else{

if((p=mono2ega(rightarrow))!=NULL){

    putimage(sb->dnarr.left,sb->dnarr.bottom-15,p,COPY_PUT);

    free(p);

}else errorbeep();

}

drawthumb(sb);

mouseon();

}

erasethumb(sb)

    SCROLLBAR *sb;

{

    mouseoff();

    setfillpattern(scrollbarfill,13);

    if(sb->head.type==invertscroll)

        bar(sb->thumb.left+1,sb->thumb.top,sb->thumb.right-1

            ,sb->thumb.bottom-1);

    else bar(sb->thumb.left,sb->thumb.top+1,

        sb->thumb.right-1,

        sb->thumb.bottom-1);

```

เอกสารนี้เป็น mouseon(); สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

drawthumb(sb)
    SCROLLBAR *sb;
{
    static char thumb2[]={ 0x0f,0x00,0x0f,0x00,
        0x00,0x00,0x7f,0xfe,0x7f,0xfe,0x7f,0xfe,
        0x7f,0xfe,0x7f,0xfe,0x7f,0xfe,0x7f,0xfe,
        0x7f,0xfe,0x7f,0xfe,0x7f,0xfe,0x7f,0xfe,
        0x7f,0xfe,0x7f,0xfe,0x7f,0xfe,0x00,0x00 };
    static char thumb16[]={ 0x0f,0x00,0x0f,0x00,
        0x00,0x00,0x7f,0xfe,0x7f,0xfe,0x7f,0xfe,
        0x7f,0xfe,0x7f,0xfe,0x7f,0xfe,0x7f,0xfe,
        0x7f,0xfe,0x7f,0xfe,0x7f,0xfe,0x7f,0xfe,
        0x7f,0xfe,0x7f,0xfe,0x7f,0xfe,0x7f,0xfe,
        0x7f,0xfe,0x7f,0xfe,0x7f,0xfe,0x7f,0xfe,
        0x7f,0xfe,0x7f,0xfe,0x7f,0xfe,0x7f,0xfe,
        0x7f,0xfe,0x7f,0xfe,0x7f,0xfe,0x7f,0xfe,
        0x7f,0xfe,0x7f,0xfe,0x7f,0xfe,0x7f,0xfe,
        0x7f,0xfe,0x7f,0xfe,0x7f,0xfe,0x7f,0xfe,
        0x7f,0xfe,0x7f,0xfe,0x7f,0xfe,0x7f,0xfe,
        0x7f,0xfe,0x7f,0xfe,0x7f,0xfe,0x7f,0xfe,
        0x7f,0xfe,0x7f,0xfe,0x7f,0xfe,0x7f,0xfe,
        0x7f,0xfe,0x7f,0xfe,0x7f,0xfe,0x7f,0xfe,
        0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00 };

    if(sb->active==ACTIVE){

```

```

        sb->thumb.top,thumb2,COPY_PUT);

    else putimage(sb->thumb.left,sb->thumb.top,thumb16,
        COPY_PUT);

    mouseon();
}

}

SCROLLBAR *findscrollbar(WINDOW *w,POINT *p)
{
    OBJECTHEAD *oh;
    SCROLLBAR *b;
    oh=(OBJECTHEAD *)w;
    while(oh!=NULL){
        if(oh->type==invertscroll||oh->type==inhorscroll){
            b=(SCROLLBAR *)oh;
            if(pointinrect(p,&b->frame)&&b->active==ACTIVE)
                return b;
        }
        oh=oh->next;
    }

    return NULL;
}

trackscrollbar(sb,rt)
    SCROLLBAR *sb;
    unsigned int rt;
{
    POINT p,oldp;
    int i,n;

```

```

sb->head.type==inhorscroll)
    &&sb->active==ACTIVE){
if(rt&inuparrow){
    invertrect(&sb->uparr);
    n=250;
    while(sb->cur>sb->min&&mousedown(&p)&&
        pointinrect(&p,&sb->uparr)){
erasethumb(sb);
adjustthumb(sb,-1);
drawthumb(sb);
delay(n);
n=25;
    }
    invertrect(&sb->uparr);
}
else if(rt&indownarrow){
    invertrect(&sb->dnarr);
    n=250;
    while(sb->cur<(sb->max-1)&&mousedown(&p)&&
        pointinrect(&p,&sb->dnarr)){
erasethumb(sb);
adjustthumb(sb,1);
drawthumb(sb);
delay(n);
n=25;
    }
    invertrect(&sb->dnarr);
}
if(sb->head.type==invertscroll){
    erasethumb(sb);
    adjustthumb(sb,0);
    drawthumb(sb);
    delay(n);
    n=25;
}

```

```

erasethumb(sb);

if (sb->thumb.top==sb->jump.top)
    sb->cur=sb->min;
else if (sb->thumb.bottom==sb->jump.bottom)
    sb->cur=sb->max;
else sb->cur=(int)((float)(sb->max-sb->min)/
    ((float)(sb->size-48)/(sb->thumb.top-sb->jump.top)));
drawthumb(sb);
    }
    else{
erasethumb(sb);
if (sb->thumb.left==sb->jump.left)
    sb->cur=sb->min;
else if (sb->thumb.right==sb->jump.right)
    sb->cur=sb->max;
else sb->cur=(int)((float)(sb->max-sb->min)/
    ((float)(sb->size-48)/(sb->thumb.left-sb->jump.left)));
drawthumb(sb);
    }
}

else if (rt&injumpup){
    erasethumb(sb);
    adjustthumb(sb,-10);
    drawthumb(sb);
}

else if (rt&injumpdown){
    erasethumb(sb);
    adjustthumb(sb,10);
    drawthumb(sb);
}

```

เอกสารนี้เป็นเอกสารเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใด ๆ ขอสงวนสิทธิ์ในนามของอาจารย์ผู้จัดทำ และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

drawthumb(sb);

if(sb->head.type==invertscroll){
    erasethumb(sb);

    if(sb->thumb.top==sb->jump.top)
sb->cur=sb->min;

    else if(sb->thumb.bottom==sb->jump.bottom)
sb->cur=sb->max;

    else sb->cur=(int)(((float)(sb->max-sb->min)/
        ((float)(sb->size-48)/
        (sb->thumb.top-sb->jump.top))));
    drawthumb(sb);
}
else{
    erasethumb(sb);
    if(sb->thumb.left==sb->jump.left)
sb->cur=sb->min;

    else if(sb->thumb.right==sb->jump.right)
sb->cur=sb->max;

    else sb->cur=(int)(((float)(sb->max-sb->min)/
        ((float)(sb->size-48)/
        (sb->thumb.left-sb->jump.left))));
    drawthumb(sb);
}
}

else if(rt&inthumb){
    if(sb->head.type==invertscroll){

```

เอกสารนี้เป็นเอกสาร mouse down(&oldp); การศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น while(mouse down(&p)) { ต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

if(oldp.y!=p.y&&pointinrect(&p,&sb->jump)){
    erasethumb(sb);
    i=p.y-oldp.y;
    if((sb->thumb.top+i)<=sb->jump.top!!
        (sb->thumb.bottom+i)>sb->jump.bottom)
        i=0;
    sb->thumb.top+=i;
    sb->thumb.bottom+=i;
    drawthumb(sb);
    memcpy((char *)&oldp,(char *)&p,sizeof(POINT));
}
}
erasethumb(sb);
if(sb->thumb.top==sb->jump.top)
sb->cur=sb->min;
else if(sb->thumb.bottom==sb->jump.bottom)
sb->cur=sb->max;
else sb->cur=(int)((float)(sb->max-sb->min)/
    ((float)(sb->size-48)/(sb->thumb.top-sb->jump.top)));
drawthumb(sb);
}
else if(sb->head.type==inhorscroll){
    mousedown(&oldp);
    while(mousedown(&p)){
        if(oldp.x!=p.x&&pointinrect(&p,&sb->jump)){
            erasethumb(sb);

```

```

        (sb->thumb.right+i)>sb->jump.right)

        i=0;
        sb->thumb.left+=i;
        sb->thumb.right+=i;
        drawthumb(sb);
        memcpy((char *)&oldp,(char *)&p,sizeof(POINT));
    }
}

erasethumb(sb);
if(sb->thumb.left==sb->jump.left)
    sb->cur=sb->min;
else if(sb->thumb.right==sb->jump.right)
    sb->cur=sb->max;
else sb->cur=(int)((float)(sb->max-sb->min)/
((float)(sb->size-48)/(sb->thumb.left-sb->jump.left)));
drawthumb(sb);
    }
}

while (mousedown(&p));
}
}

```

```
adjustthumb(sb,n)
```

```
    SCROLLBAR *sb;
```

```
    int n;
```

```
{
```

```
    int i,rn,f;
```

```
    i=sb->cur+n;
```

```
    if(i<sb->min)i=sb->min;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น ยกเว้นแต่มีเหตุเปลี่ยนแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

else if(i>=sb->max)i=sb->max-1;
sb->cur=i;
rn=sb->size-48;
if(sb->cur==sb->min)f=0;
else if(sb->cur==(sb->max-1))f=rn;
else f=(int)(((float)rn/((float)(sb->max-sb->min))/
            (float)sb->cur));
if(sb->head.type==invertscroll){
    if(f==rn)sb->thumb.top=sb->dnarr.top-16;
    else sb->thumb.top=sb->uparr.bottom+f+1;
    sb->thumb.bottom=sb->thumb.top+16;
}
else if(sb->head.type==inhorscroll){
if(f==rn)sb->thumb.left=sb->dnarr.left-16;
else sb->thumb.left=sb->uparr.right+f+1;
sb->thumb.right=sb->thumb.left+16;
}
}
char *mono2ega(char *source)
{
    char *p;
    int x,y,j,ls,sz;
    x=1+source[0]+(source[1]<<8);
    y=1+source[2]+(source[3]<<8);
    if((sz=imagesize(0,0,x,y))!=-1){
        if((p=malloc(sz))!=NULL){
            memcpy(p,source,4);

```

เอกสารนี้เป็นเอกสารสงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามเผยแพร่ต่อสาธารณะ และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

for(j=0;j<y;++j){
    memcpy(p+4+((j*4)*ls),source+4+(j*ls),ls);
    memcpy(p+4+ls+((j*4)*ls),source+4+(j*ls),ls);
    memcpy(p+4+(ls*2)+((j*4)*ls),source+4+(j*ls),ls);
    memcpy(p+4+(ls*3)+((j*4)*ls),source+4+(j*ls),ls);
}

return p;
}

}else return NULL;
}

char *ega2mono(char *source)
{
    char *p;
    int x,y,j,ls,sz;
    x=1+source[0]+(source[1]<<8);
    y=1+source[2]+(source[3]<<8);
    ls=pixels2bytes(x);
    sz=(ls*y)+4;
    if((p=malloc(sz))!=NULL){
        memcpy(p,source,4);

        for(j=0;j<y;++j)memcpy(p+4+(j*ls),source+4+
            ((j*4)*ls),ls);

        return p;
    }else return NULL;
}

ismonomode()
{

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 if(getmaxcolor()==1)return 1;
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        else return 0;
    }

    /***** MENU FUNCTION *****/

    initmenumanager()
    {
        int i;
        for(i=0;i<MENUCOUNT;++i) menuarray[i]=NULL;
        setrect(&menubarrect,4,4,SCREENWIDE-5,MENUBARDEEP);
    }

    initwindowmanager()
    {
        screenwindow.head.type=inwindow;
        screenwindow.head.next=NULL;
        setrect(&screenwindow.frame,0,0,SCREENWIDE,SCREENDEEP);
        screenwindow.back=NULL;
        arrowcursor();
    }

    addmenu(m)
        MENU *m;
    {
        if(menuindex<MENUCOUNT){
            menuarray[menuindex++]=m;
            return (1);
        }else return (0);
    }

    drawmenubar()
    {

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

setfillstyle(SOLID_FILL,WHITE);
setcolor(LIGHTRED);
bar(4,4,SCREENWIDE-5,MENUBARDEEP);
for(i=0;i<menuindex;++i)
    outtextxy(i*(9*MENUTITLESIZE)+10,15,menuarray[i]->title);
}

domenu(p)
    POINT *p;
{
    WINDOW w;
    RECT r,lr;
    POINT pn;
    int m,n,i;
    m=p->x/(MENUTITLESIZE<<3);
    if(m>=menuindex)return(0);
    r.left=(m*(MENUTITLESIZE<<3))+15;
    r.top=menubarrect.bottom+1;
    r.right=r.left+(MENULINESIZE<<3)+10;
    r.bottom=r.top+(menuarray[m]->count*MENULINEDEEP)+3;
    if(openwindow(&w,&r)){ setcolor(RED);mouseoff();
        for(i=0;i<menuarray[m]->count;++i){
            if(menuarray[m]->item[i].name[0]==' ')
                outtextxy(r.left+10,r.top+(i*MENULINEDEEP)+14,
                    menuarray[m]->item[i].name+1);
            else if(menuarray[m]->item[i].name[0]==CHECKMARK)
                outtextxy(r.left,r.top+(i*MENULINEDEEP)+6,
                    menuarray[m]->item[i].name);
            else

```

```

outtextxy(r.left+8,r.top+(i*MENULINEDEEP)+6,
menuarray[m]->item[i].name+1);
}
i=-1; mouseon();
while(mousedown(&pn)){
if(pointinrect(&pn,&r)){
n=(pn.y-r.top)/MENULINEDEEP;
if(n!=i){
if(i!=-1){
setrect(&lr,r.left+1,r.top+(i*MENULINEDEEP),
r.right,r.top+(i*MENULINEDEEP)+MENULINEDEEP);
if(menuarray[m]->item[i].name[0]!=' ')
invertrect(&lr);
}
setrect(&lr,r.left+1,r.top+(n*MENULINEDEEP),
r.right,r.top+(n*MENULINEDEEP)+MENULINEDEEP);
if(menuarray[m]->item[n].name[0]!=' ')
invertrect(&lr);
i=n;
}
}
}
closewindow(&w);
if(pointinrect(&pn,&r)&&menuarray[m]->item[m].name[0]
==' '){
n=(pn.y-r.top)/MENULINEDEEP;
(menuarray[m]->item[n].proc)(n);
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        return (1);
    }else return (0);
}
dispatchmenuitem()
{
    char *p;
    unsigned int c,ca=0,i=0,menu,item,rt=0;
    if(kbhit()){
        if(!(bioskey(1)&0x00ff)){
            c=getkey();
            while(altkeytable[i]==0xffff){
                if(altkeytable[i]==c){
                    ca=altkeytable[i+1];
                    break;
                }
                i+=2;
            }
            if(!ca)return (0);
            for(menu=0;menu<menuindex;++menu){
                if(rt)break;
                for(item=0;item<menuarray[menu]->count;++item){
                    if(rt)break;
                    if((p=strchr(menuarray[menu]->item[item].name,
                                KEYMENUCHAR))!=NULL){
                        if(ca==(unsigned int)p[1]){
                            (menuarray[menu]->item[item].proc)(item);
                            rt=1;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    }
}
}
}
}

}

/***** TEXT FUNCTIONS *****/
trackeditfield(f)
    EDITFIELD *f;

{
    POINT p;
    mouseloc(&p);
    if(pointinrect(&p,&f->frame))f->select=0xff;
    else f->select=0x00;
}

EDITFIELD *fineditfield(w,p)
    WINDOW *w;
    POINT *p;

{
    OBJECTHEAD *oh;
    EDITFIELD *f;

    oh=(OBJECTHEAD *)w;
    while(oh!=NULL){
        if(oh->type==ineditfield){
            f=(EDITFIELD *)oh;
            if(pointinrect(p,&f->frame)&&f->active==ACTIVE)
                return(f);
        }
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

t->frame.bottom=y+FONTDDEEP+8;
t->head.type=ineditfield;
t->head.next=NULL;
draweditfield(t);
}
draweditfield(t)
    EDITFIELD *t;
{
    mouseoff();
    setwritemode(COPY_PUT);
    setfillstyle(SOLID_FILL,LIGHTCYAN);
    setlinestyle(SOLID_LINE,0,THICK_WIDTH);
    setcolor(LIGHTRED);
    bar(t->frame.left,t->frame.top,t->frame.right,
        t->frame.bottom);
    rectangle(t->frame.left,t->frame.top,t->frame.right,
        t->frame.bottom);
    setcolor(RED);
    outtextxy(t->x+8,t->y+5,t->text);
    if(t->active==ACTIVE&& t->select){
        setcolor(2);/*GREEN*/
        setwritemode(XOR_PUT);
        setlinestyle(SOLID_LINE,0,THICK_WIDTH);
        line(t->frame.left+(8*t->curpos)+2,t->frame.top+2,
            t->frame.left+(8*t->curpos)+2,t->frame.bottom-2);
    }
    mouseon();
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

/*****/

int doquit()
{
    alive=0;
}

drawgroupe(er,ed,eta,etd,eaz,eel)
, EDITFIELD *er,*ed,*eta,*etd,*eaz,*eel;
,
{
    int i;
    draweditfield(er);
    for(i=0;i<3;i++){
        draweditfield(ed++);
        draweditfield(eta++);
        draweditfield(etd++);
        draweditfield(eaz++);
        draweditfield(eel++);
    }
}

clearbuf()
,
{
    int i;
    for(i=0;i<3;i++){
        sprintf(bd[i],"%s","");
        sprintf(bta[i],"%s","");
        sprintf(btd[i],"%s","");
        sprintf(baz[i],"%s","");
        sprintf(bel[i],"%s","");
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

}
readfile(){
    FILE *fp;

    int i;

    if((fp=fopen("data.mo","r"))!=NULL){

        for(i=0;i<MREC;i++)

        fread(&rec[i],sizeof(REC),1,fp);

        fclose(fp);

    }else errorbeep();
}
int autocontrol()
{
    WINDOW w;
    POINT p;
    BUTTON ok,can,*bp;
    SCROLLBAR saz,sel,*sp;
    EDITFIELD eaz,eel,*ep;
    RECT r;
    char g;

    int rt,a=0xff,i,c;

    char bufaz[4];
    char bufel[4];

    int az,el;

    sprintf(bufaz,"%s","");
    sprintf(bufel,"%s","");

    setrect(&r,140,120,500,400);

    if(openwindow(&w,&r)){

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามแก้ไขเปลี่ยนแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        ,r.bottom-20);

addbutton(&w,&ok,"OK",ACTIVE);

setrect(&can.frame,r.left+200,r.bottom-50,r.left+260
        ,r.bottom-20);

addbutton(&w,&can,"CANCEL",ACTIVE);

mouseoff();

outtextxy(r.left+120,r.top+30,"AUTO CONTROL");
outtextxy(r.left+50,r.top+80,"AZIMUTH");
outtextxy(r.left+50,r.top+160,"ELEVATION");
outtextxy(r.left+290,r.top+100,"DEGREE");
outtextxy(r.left+290,r.top+180,"DEGREE");
mouseon();

addhscroll(&w,&saz,r.left+20,r.top+100,200,
0,360,0,ACTIVE);
addhscroll(&w,&sel,r.left+20,r.top+180,200,
0,180,0,ACTIVE);

addeditfield(&w,&eaz,r.left+240,r.top+100,
3,bufaz,NULL,ACTIVE);
addeditfield(&w,&eel,r.left+240,r.top+180,
3,bufel,NULL,ACTIVE);

do{

keyboard(&w);

if(mousedown(&p)){

rt=wheremouse(&p,&w);

if(rt&inbutton){

bp=findbutton(&w,&p);

trackbutton(bp);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

a=0;

send(0x41);    /* check  Z-80 ready "0x41" */
g=rport();     /* wait for 0x41             */
if(g!=0x41){   /* if any error return      */
    error();
    closewindow(&w);
    return;
}
/* z80 is ready for interface */
c=az;send(c);  /* send AZ  byte 0             */
g=rport();     /* wait for 0x41             */
if(g!=0x41){   /* if any error return */
    error();
    closewindow(&w);
    return;
}

send(c>>8);    /* send AZ byte 1             */
g=rport();     /* wait for 0x41             */
if(g!=0x41){   /* if any error return */
    error();
    closewindow(&w);
    return;
}

c=el;

send(c);        /* send EL byte 0             */
g=rport();      /* wait for 0x41             */
if(g!=0x41){    /* if any error return */
    error();
    closewindow(&w);
    return;
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมีเหตุดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        return;
    }

    send(c>>8);        /* send EL byte 1        */

    }

    else if(bp==&can)a=0;

    }

    else if(rt&inhorscroll){
        sp=findscrollbar(&w,&p);
        trackscrollbar(sp,rt);
        if(sp==&saz){
            az=sp->cur;
            sprintf(bufaz,"%u",az);
            draweditfield(&eaz);
        }else if(sp==&sel){
            el=sp->cur;
            sprintf(bufel,"%u",el);
            draweditfield(&eel);
        }

        } else if(rt&ineditfield){
            ep=findeditfield(&w,&p);
            trackeditfield(ep);
            draweditfield(ep);
        }else errorbeep();
    }

    }

    }while(a);

    closewindow(&w);

    }else errorbeep();

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

int stepcontrol()
{
    WINDOW w;

    POINT p;

    CHECKBOX *cb;

    BUTTON can,azl,azr,elu,eld,*bp;

    RECT r;

    unsigned int rt,a=0xff;

    int i;

    char c;

    setrect(&r,140,170,490,340);

    if(openwindow(&w,&r)){

        outtextxy(r.left+120,r.top+20,"MANUAL CONTROL");

        outtextxy(r.left+70,r.top+40,"AZIMUTH");

        outtextxy(r.left+220,r.top+40,"ELEVATION");


        setrect(&azl.frame,r.left+30,r.bottom-100,r.left+80,
            r.bottom-70);

        addbutton(&w,&azl,"LEFT",ACTIVE);

        setrect(&azr.frame,r.left+100,r.bottom-100,
            r.left+150,r.bottom-70);

        addbutton(&w,&azr,"RIGHT",ACTIVE);

        setrect(&elu.frame,r.left+200,r.bottom-100,
            r.left+250,r.bottom-70);

        addbutton(&w,&elu,"UP",ACTIVE);

        setrect(&eld.frame,r.left+270,r.bottom-100,
            r.left+320,r.bottom-70);
    }
}

```

```

addbutton(&w,&eld,"DOWN",ACTIVE);

setrect(&can.frame,r.left+135,r.bottom-50,
        r.left+200,r.bottom-20);

addbutton(&w,&can,"CANCEL",ACTIVE);

send(0x40); /*send 0x40 to tell z-80 for interface */
for(i=0;i<100;i++); /* delay */

c=rport(); /* read com 1 port */

if(c!=0x41){
error();
closewindow(&w);
return;
} /* z-80 is readdy for interface */
do{
keyboard(&w);
if(mousedown(&p)){
rt=wheremouse(&p,&w);
if(rt&inbutton){
bp=findbutton(&w,&p);
if(bp!=NULL&&bp->head.type==inbutton&&bp->active
==ACTIVE)
{
invertrect(&bp->frame);
delay(1);
invertrect(&bp->frame);
}else errorbeep();
if(bp==&can){

```

```

a=0;

send(0xff);          /* 0xff for cancel */

    }else if(bp==&azl){ /* 01 left az */

send(0x01);          /* 00 right az */

    }else if(bp==&azr){ /* 10 up el */

send(0x00);          /* 11 down el */

    }else if(bp==&elu){

send(0x10);

    }else if(bp==&eld){

send(0x11);

    }

}else errorbeep();

}

    }while(a);

    closewindow(&w);

}else errorbeep();

}

int outdata() /****** transfer data to z-80 *****/

{

    FILE *fp2;

    char str[6],name[10];

    char b[50];

    int i,h=0,m=0,d=0;

    int x,y;

    char c;

    long int j=0;

```

```

    int date,month,year,hour,min,sec,az,el;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    for(i=0;i<100;i++);
    c=rport();      /* read com1 and check that z80 is read */
    if(c!=0x41){
        error();return;
    }
    /* z80 is ready for interface */
    x=100;y=60;
    setcolor(15);
    clearviewport();
    if(!(fp2=fopen("sat2.mo","r"))){return;}
    while(fscanf(fp2,"%s",str)!=EOF){
        if(y>=450){y=60;clearviewport();}
        if(!strcmp("SAT:",str)){
            fscanf(fp2,"%s",name);
            sprintf(b,"SATELLITE NAME: %s",name);
            outtextxy(x,y,b);y+=12;
            send(0x53);    /* send S */
            c=rport();      /* read com1 and check that z80 is read */
            if(c!=0x41){
                error();return;
            }
            send(0x54);    /* send T */
            c=rport();      /* read com1 and check that z80 is read */
            if(c!=0x41){
                error();return;
            }
        }
    }
}

```

เอกสารนี้เป็นเอกสารที่สแกนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        fscanf(fp2,"%d%d%d",&date,&month,&year);

if(!(date==d)){
    d=date;
    i=date;
    sprintf(b,"DAY %d (%x %x)",i,c=i,i>>8);
    send(0x44);/*D*/
    c=rport();        /* read com1 and check that z80 is read */
    if(c!=0x41){
        error();return;
    }
    send(0x44);        /* send code "DAY" to begin new day,month */
    c=rport();        /* read com1 and check that z80 is read */
    if(c!=0x41){
        error();return;
    }
    send(i);            /* date send date byte 0 */
    c=rport();
    if(c!=0x41){
        error();return;
    }
    outtextxy(x,y,b);
    i=month;
    sprintf(b,"MONTH %d (%x %x)",i,c=i,i>>8);
    send(i);            /* send month byte 0 */
    c=rport();
    if(c!=0x41){
        error();return;
    }

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    outtextxy(x+160,y,b);

    i=year;

    sprintf(b,"YEAR %d (%x %x)",i,c=i,i>>8);

    outtextxy(x+300,y,b);

    j+=6; y+=12;
}

    }

    else if(!strcmp("TIME:",str)){
fscanf(fp2,"%d%d%d",&hour,&min,&az,&el);
if(!((hour==h)!!(min==m))){

    send(0x48);    /* send H */

    c=rport();

    if(c!=0x41){
        error();return;
    }

    send(0x48);    /* send H */

    c=rport();

    if(c!=0x41){
        error();return;
    }

    h=hour;m=min;

    i=hour;

    sprintf(b,"HOUR: %d (%x %x)",i,c=i,i>>8);

    send(i);        /* send hour byte0 */

    c=rport();

    if(c!=0x41){

        error();return;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    }

    outtextxy(x,y,b);

    i=min;

    sprintf(b,"MIN: %d (%x %x)",i,c=i,i>>8);

    send(i);          /* send minite byte0 */

    c=rport();

    if(c!=0x41){

        error();return;

    }

    outtextxy(x+200,y,b);y+=12;

    j+=4;

    /* getch(); */

}

send(0x41);          /* send A */

c=rport();

if(c!=0x41){

    error();return;

}

send(0x5a);          /* send Z */

c=rport();

if(c!=0x41){

    error();return;

}

i=az;                /* azimuth angle */

sprintf(b,"AZ= %d (%x %x)",i,c=i,i>>8);

send(i);             /* send AZ byte 0 */

c=rport();

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    error();return;
}
send(i>>8);    /* send azimuth byte1 */
c=rport();
if(c!=0x41){
    error();return;
}
outtextxy(x,y,b);
i=e1;          /* elevation angle */

sprintf(b,"EL= %d (%x %x)",i,c=i,i>>8);
send(i);       /* send EL byte 0 */
c=rport();
if(c!=0x41){
    error();return;
}
send(i>>8);    /* send elevation byte 1 */
c=rport();
if(c!=0x41){
    error();return;
}
outtextxy(x+200,y,b);y+=12;
j+=4;
}
}

fclose(fp2);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามแก้ไขเพิ่มเติมเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    send(0xff);          /* send end code byte 0 */

    c=rport();
    if(c!=0x41){
        error();return;
    }

    send(0xff);          /* send end code byte 1 */
}

init()
{
    union REGS r;
    r.h.ah=0x00;
    r.h.al=0xa3; /* 0xe3 9600 oxa3 2400 */
    r.x.dx=0x01; /* com2 */
    int86(0x14,&r,&r);
}

send(c)
char c;
{
    union REGS r;
    r.h.ah=0x01;
    r.h.al=c;
    r.x.dx=0x01;
    int86(0x14,&r,&r);
    if(r.h.ah&0x80)errorbeep();
    /*printf("error ah bit 7");*/
}

int rport()

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

union REGS r;

int i;

while(!(check()&256)){
    if(kbhit())break;
}

r.x.dx=0x01;
r.h.ah=0x02;

int86(0x14,&r,&r);

if(r.h.ah&0x80)
    errorbeep();

return r.h.al;
}

int check()
{
    union REGS r;
    r.x.dx=0x01;
    r.h.ah=0x03;
    int86(0x14,&r,&r);
    return r.x.ax;
}

error()
{
    RECT r;
    WINDOW w;

    setrect(&r,200,200,400,350);

    if(openwindow(&w,&r)){

```

```

        settextstyle(0,0,2);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามเผยแพร่ต่อสาธารณะโดยไม่ได้รับอนุญาตจากเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    settextstyle(0,0,0);
    getch();
    closewindow(&w);
} else errorbeep();
}

track()
{
    clearviewport();
    tracking();
    clearviewport();
}

int editdata()
{
    WINDOW w;
    RECT r;
    POINT p;
    CHECKBOX cb[2],*cp;
    BUTTON ok,can,prv,nex,*bp;
    EDITFIELD *ep,ed[11],en;

    int a=0xff,i,j;

    SAT *re;

    unsigned int rt;

    char *b;

    for(i=0;i<11;i++){

        sprintf(buf[i],"%s","");

    }

    for(i=0;i<9;i++){

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 sat[i].name=name[i];
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    sat[i].id=0;
}
setrect(&r,170,50,500,400);
if(openwindow(&w,&r)){
    mouseoff();
    outtextxy(r.left+50,r.top+20,
    "EDIT SATELLITE ORBITAL ELEMENTS");
    outtextxy(r.left+20,r.top+40, "SATELLITE      :");
    outtextxy(r.left+150,r.top+40,name[0]);
    outtextxy(r.left+20,r.top+60, "CATALOG NUMBER:");
    outtextxy(r.left+20,r.top+80, "EPOCH   TIME   :");
    outtextxy(r.left+20,r.top+100,"ELEMENT SET   :");
    outtextxy(r.left+20,r.top+120,"INCLINATION   :");
    outtextxy(r.left+20,r.top+140,"RA   OF   NODE :");
    outtextxy(r.left+20,r.top+160,"ECCENTRICITY   :");
    outtextxy(r.left+20,r.top+180,"ARG OF PERIGEE:");
    outtextxy(r.left+20,r.top+200,"MEAN  ANOMALY :");
    outtextxy(r.left+20,r.top+220,"MEAN   MOTION :");
    outtextxy(r.left+20,r.top+240,"DECAY  RATE   :");
    outtextxy(r.left+20,r.top+260,"EPOCH  REV    :");
    mouseon();
    addcheckbox(&w,&cb[0],r.left+20,r.top+280,
    "CLEAR RECORD",0xff,ACTIVE);
    addcheckbox(&w,&cb[1],r.left+180,r.top+280,
    "READ OLD DATA",0x00,ACTIVE);
    setrect(&ok.frame,r.left+210,r.bottom-50,r.left+250,
    r.bottom-20);
    addbutton(&w,&ok, "SAVE",ACTIVE);
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

setrect(&can.frame,r.left+140,r.bottom-50,r.left+200,
        r.bottom-20);
addbutton(&w,&can,"CANCEL",ACTIVE);
setrect(&prv.frame,r.left+20,r.bottom-50,r.left+80,
        r.bottom-20);
addbutton(&w,&prv,"PREVIEW",ACTIVE);
setrect(&nex.frame,r.left+90,r.bottom-50,r.left+130,
        r.bottom-20);
addbutton(&w,&nex,"NEXT",ACTIVE);
sprintf(bufn,"%s",sat[0].name);
addeditfield(&w,&en,r.left+150,r.top+35,8,bufn,NULL,
        INACTIVE);
for(i=0;i<11;i++)
    addeditfield(&w,&ed[i],r.left+150,r.top+55+i*20,15,
    buf[i],NULL,ACTIVE);
n=0;
do{
    keyboard(&w);
    if(mousedown(&p)){
        rt=wheremouse(&p,&w);
        if(rt&inbutton){
            bp=findbutton(&w,&p);
            trackbutton(bp);
            if(bp==&ok){
                a=0;
                for(i=0;i<MAX;i++){popbuf(i);getdata(i);}
                savef();
            }
        }
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

else if(bp==&can){
    a=0;
}

else if(bp==&nex){
if(n<MAX){
    getdata(n);

    if(n==MAX-1)n=0;
    else n++;
    if(sat[n].id==1)popbuf(n);
    else {clear();
sprintf(bufn,"%s",sat[n].name);
    }
    draweditfield(&en);
    for(i=0;i<11;i++)draweditfield(&ed[i]);
    }

    else if(bp==&prv){
if(n<=0)n=MAX-1;
    else n--;
    popbuf(n);
    draweditfield(&en);
    for(i=0;i<11;i++)draweditfield(&ed[i]);
    }
}
else if(rt&ineditfield){
    ep=findeditfield(&w,&p);
    trackeditfield(ep);

    draweditfield(ep);
}
else if(rt&incheckbox){

```

```

cp=findcheckbox(&w,&p);
trackcheckbox(cp);
if(cp==&cb[0]){
    cb[0].select=0xff;
    cb[1].select=0x00;
    drawcheckbox(&cb[0]);
    drawcheckbox(&cb[1]);
    clear();
    for(i=0;i<11;i++)draweditfield(&ed[i]);
}
else if(cp==&cb[1]){
    cb[0].select=0x00;
    cb[1].select=0xff;
    drawcheckbox(&cb[0]);
    drawcheckbox(&cb[1]);
    readf();
    for(i=0;i<MAX;i++){sat[i].id=1;}
    popbuf(n);
    sprintf(bufn,"%s",sat[n].name);
    draweditfield(&en);
    for(i=0;i<11;i++)draweditfield(&ed[i]);
}
}
else errorbeep();
}

while(a);

closewindow(&w);

else errorbeep();

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

int readdata()
{
    WINDOW w;
    RECT r;
    POINT p;
    CHECKBOX cb[2],*cp;
    BUTTON ok,can,prv,nex,*bp;
    EDITFIELD *ep,ed[11],en;
    int a=0xff,i,j;
    unsigned int rt;
    char *b;
    readf();
    for(i=0;i<11;i++){
        sprintf(buf[i],"%s","");
    }
    for(i=0;i<9;i++){sat[i].name=name[i];sat[i].id=0;}
    setrect(&r,170,50,500,380);
    if(openwindow(&w,&r)){
        mouseoff();
        outtextxy(r.left+50,r.top+20,
            "TRACKING SATTELLITE DATA");
        outtextxy(r.left+20,r.top+40, "SATELLITE      :");
        outtextxy(r.left+150,r.top+40,name[0]);
        outtextxy(r.left+20,r.top+60, "CATALOG NUMBER:");
        outtextxy(r.left+20,r.top+80, "EPOCH    TIME  :");
        outtextxy(r.left+20,r.top+100,"ELEMENT SET   :");
        outtextxy(r.left+20,r.top+120,"INCLINATION   :");
        outtextxy(r.left+20,r.top+140,"RA OF NODE    :");
    }
}

```

```

outtextxy(r.left+20,r.top+160,"ECCENTRICITY  :");
outtextxy(r.left+20,r.top+180,"ARG OF PERIGEE:");
outtextxy(r.left+20,r.top+200,"MEAN  ANOMALY  :");
outtextxy(r.left+20,r.top+220,"MEAN  MOTION  :");
outtextxy(r.left+20,r.top+240,"DECAY  RATE   :");
outtextxy(r.left+20,r.top+260,"EPOCH  REV    :");

mouseon();n=0;

popbuf(n);

setrect(&ok.frame,r.left+210,r.bottom-50,r.left+250,
        r.bottom-20);
addbutton(&w,&ok,"OK",ACTIVE);
setrect(&can.frame,r.left+140,r.bottom-50,r.left+200,
        r.bottom-20);
addbutton(&w,&can,"CANCEL",ACTIVE);
setrect(&prv.frame,r.left+20,r.bottom-50,r.left+80,
        r.bottom-20);
addbutton(&w,&prv,"PREVIEW",ACTIVE);
setrect(&nex.frame,r.left+90,r.bottom-50,r.left+130,
        r.bottom-20);
addbutton(&w,&nex,"NEXT",ACTIVE);
addeditfield(&w,&en,r.left+150,r.top+35,8,bufn,NULL,
              INACTIVE);

for(i=0;i<11;i++)
    addeditfield(&w,&ed[i],r.left+150,r.top+55+i*20,15,
                buf[i],NULL,INACTIVE);

do{
    keyboard(&w);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมีเหตุดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

rt=wheremouse(&p,&w);

if(rt&inbutton){
    bp=findbutton(&w,&p);
    trackbutton(bp);
    if(bp==&ok){
a=0;
track();
    }
    else if(bp==&can){
a=0;
    }
    else if(bp==&nex){
if(n<MAX){
    if(n==MAX-1)n=0;
    else n++;
    popbuf(n);
    draweditfield(&en);
    for(i=0;i<11;i++)draweditfield(&ed[i]);
    }
    }
    else if(bp==&prv){
    if(n<=0)n=MAX-1;
    else n--;
    popbuf(n);
    draweditfield(&en);
    for(i=0;i<11;i++)draweditfield(&ed[i]);
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมีเหตุดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    ep=findeditfield(&w,&p);

    trackeditfield(ep);

    draweditfield(ep);

}

}else errorbeep();
}

```

```

    }while(a);

    closewindow(&w);

}else errorbeep();
}

```

```

getdata(n)

```

```

    unsigned int n;

{
    SAT *r;
    r=&sat[n];
    r->id=1;
    r->name=name[n];
    r->cata=atol(buf[0]);
    r->epocht=atof(buf[1]);
    r->element=atoi(buf[2]);
    r->inclination=atof(buf[3]);
    r->ranode=atof(buf[4]);
    r->eccen=atof(buf[5]);
    r->arg=atof(buf[6]);
    r->meana=atof(buf[7]);
    r->meanm=atof(buf[8]);
    r->decay=atof(buf[9]);
    r->epochrev=atol(buf[10]);
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

}

clear()
{
    int i;
    for(i=0;i<11;i++) sprintf(buf[i],"%s","");
}

popbuf(n)
    int n;
{
    int i;
    SAT *r;
    r=&sat[n];
    sprintf(bufn,"%s",r->name);
    sprintf(buf[0],"%ld",r->cata);
    sprintf(buf[1],"%6.8lf",r->epocht);
    sprintf(buf[2],"%d",r->element);
    sprintf(buf[3],"%3.8lf",r->inclination);
    sprintf(buf[4],"%3.8lf",r->ranode);
    sprintf(buf[5],"%1.8lf",r->eccen);
    sprintf(buf[6],"%3.8lf",r->arq);
    sprintf(buf[7],"%3.8lf",r->meana);
    sprintf(buf[8],"%3.8lf",r->meanm);
    sprintf(buf[9],"%2.8lf",r->decay);
    sprintf(buf[10],"%ld",r->epochrev);
}

readf()
{

```

เอกสารนี้เป็น **FILE *fp**; ไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

int i;
SAT *r;
char *name;
long int cata;
double epocht,element,inclination,ranode,eccen,
        arg,meana,meanm,decay;
long int epochrev;
if((fp=fopen("sat.mo","r"))!=NULL){
for(i=0;i<MAX;i++){
fread(&sat[i].cata,sizeof(sat[i].cata),1,fp);
fread(&sat[i].epocht,sizeof(sat[i].epocht),1,fp);
fread(&sat[i].element,sizeof(sat[i].element),1,fp);
fread(&sat[i].inclination,sizeof(sat[i].inclination),1,fp);
fread(&sat[i].ranode,sizeof(sat[i].ranode),1,fp);
fread(&sat[i].eccen,sizeof(sat[i].eccen),1,fp);
fread(&sat[i].arg,sizeof(sat[i].arg),1,fp);
fread(&sat[i].meana,sizeof(sat[i].meana),1,fp);
fread(&sat[i].meanm,sizeof(sat[i].meanm),1,fp);
fread(&sat[i].decay,sizeof(sat[i].decay),1,fp);
fread(&sat[i].epochrev,sizeof(sat[i].epochrev),1,fp);
}
fclose(fp);
}else errorbeep();
}
savef(){
FILE *fp;
int i;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่าการฉีกทึ้งสิ่งอื่น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        if((fp=fopen("sat.mo","w"))!=NULL){
for(i=0;i<MAX;i++){
fwrite(&sat[i].cata,sizeof(sat[i].cata),1,fp);
fwrite(&sat[i].epocht,sizeof(sat[i].epocht),1,fp);
fwrite(&sat[i].element,sizeof(sat[i].element),1,fp);
fwrite(&sat[i].inclination,sizeof(sat[i].inclination),1,
        fp);

fwrite(&sat[i].ranode,sizeof(sat[i].ranode),1,fp);
fwrite(&sat[i].eccen,sizeof(sat[i].eccen),1,fp);
fwrite(&sat[i].arg,sizeof(sat[i].arg),1,fp);
fwrite(&sat[i].meana,sizeof(sat[i].meana),1,fp);
fwrite(&sat[i].meanm,sizeof(sat[i].meanm),1,fp);
fwrite(&sat[i].decay,sizeof(sat[i].decay),1,fp);
fwrite(&sat[i].epochrev,sizeof(sat[i].epochrev),1,fp);
}

        fclose(fp);
    }else errorbeep();
}

int process()
{
    int i;
    readf();
    inp=1;
    for(i=0;i<9;i++){sat[i].name=name[i];sat[i].id=0;}
    if(!(fpl=fopen("sat2.mo","w"))){
        errorbeep();return;
    }

```

1

```
fprintf(fp1,"SAT: %s\n",sat[n].name);  
popbuf(n);  
clearviewport();  
tracking();  
clearviewport();  
}  
inp=0;  
fclose(fp1);  
}
```



1

```

#include <stdio.h>
#include <math.h>
#include <time.h>
#include <graphics.h>
#define PI 3.14159265
#define PI2 (PI*2)
#define DegreesPerRadian (180/PI)
#define RadiansPerDegree (PI/180)
#define MinutesPerDay (24*60.0)
#define SecondsPerDay (60*MinutesPerDay)
#define HalfSecond (0.5/SecondsPerDay)
#define EarthRadius 6378.16 /* Kilometers */
#define EarthFlat (1/298.25) /* Earth Flattening Coeff */
#define SiderealSolar 1.0027379093
#define RefYear 79
#define RefSidereal 0.27518504
#define RefDay 0 /* Day of week */
#define DRAG 1
#define C 2.997925e5 /* Kilometers/Second */
#define ABS(x) ((x) < 0 ? -(x) : (x))
#define SQR(x) ((x)*(x))
#define MaxModes 10
char *MonthNames[] = { "Jan","Feb","Mar","Apr",
                        "May","Jun","Jul","Aug",
                        "Sep","Oct","Nov","Dec" };
int MonthDays[] = {31,28,31,30,31,30,31,31,30,31,30,31};
char *DayNames[] = { "Sunday","Monday","Tuesday",

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        "Friday", "Saturday"};

typedef struct {

    int MinPhase, MaxPhase;

    char ModeStr[20];

} ModeRec;

/* Keplerian Elements and misc. data for the satellite */
double EpochDay; /* time of epoch */
double EpochMeanAnomaly; /* Mean Anomaly at epoch */
long EpochOrbitNum; /* Integer orbit # of epoch */
double EpochRAAN; /* RAAN at epoch */
double epochMeanMotion; /* Revolutions/day */
double OrbitalDecay; /* Revolutions/day^2 */
double EpochArgPerigee; /* argument of perigee at epoch */
double Eccentricity;
double Inclination;
char *SatName="FO-20";
int ElementSet;
double BeaconFreq; /* Mhz, used for doppler calc */
double MaxPhase; /* Phase units in 1 orbit */
double perigeePhase;

int NumModes;

ModeRec Modes[MaxModes];

int PrintApogee;

/* Simulation Parameters */
double StartTime, EndTime, StepTime;

/* In Days, 1 = New Year */
/* of reference year */
/* Site Parameters */

```

```

char *SiteName="BANGKOK";

double SiteLat,SiteLong,SiteAltitude,SiteMinElev;


double Kepler(register double MeanAnomaly,
               register double Eccentricity);

double GetDay(int Year,int Month,double Day);

extern char buf[10][15];
extern char bufn[15];
extern int inp;
extern int n;
extern FILE *fp1;

tracking()
{
    double ReferenceOrbit;
                                /* Floating point orbit # at epoch */
    double CurrentTime; /* In Days */
    double CurrentOrbit;
    double AverageMotion, /* Corrected for drag */
    CurrentMotion;
    double MeanAnomaly,TrueAnomaly;
    double SemiMajorAxis;
    double Radius; /* From geocenter */
    double RAANPrecession,PerigeePrecession;
    double SSPLat,SSPLong;
    long OrbitNum,PrevOrbitNum;

```

เอกสารนี้เป็นเอกสารลิขสิทธิ์ของสำนักงานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่าการ double Azimuth,Elevation,Range; รวมถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

double PrevRange,Velocity,Doppler;

int Phase;

int DidApogee;

char buf[100];

char *bel,*baz;

int i,x=60,y=20;

int new=1;

int j;

double TmpTime,PrevTime;

int PrevVisible; mouseoff();

setcolor(WHITE);

GetSatelliteParams(n);

GetSiteParams();

GetSimulationParams();

outtextxy(x,y,"SATTELLITE NAME :");

sprintf(buf,"%d",ElementSet);

outtextxy(x,y+20,"ELEMEMENT SET :");

outtextxy(x+150,y+20,buf);

outtextxy(x,y+40,"SITE NAME: BANGKOK");

SemiMajorAxis = 331.25 *

    exp(2*log(MinutesPerDay/epochMeanMotion)/3);

GetPrecession(SemiMajorAxis,Eccentricity,Inclination,

    &RAANPrecession,&PerigeePrecession);

ReferenceOrbit = EpochMeanAnomaly/PI2 + EpochOrbitNum;

PrevDay = -10000; PrevOrbitNum = -10000;

PrevTime = StartTime-2*StepTime;

BeaconFreq *= 1E6;

```

เอกสารนี้เป็นเอกสารเพื่อการศึกษา/Convert to Hz ใช้ประโยชน์ด้านการค้า
ไม่ว่าการแก้ไขสิ่งนี้หรือการนำออกให้ต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

/* Start the loop one step early, to init OldRange */
for (CurrentTime = StartTime-StepTime;
    CurrentTime <= EndTime; CurrentTime += StepTime)
{
    #if DRAG
        AverageMotion = epochMeanMotion+
            (CurrentTime-EpochDay)*OrbitalDecay/2;
        CurrentMotion = epochMeanMotion+
            (CurrentTime-EpochDay)*OrbitalDecay;
    #else
        AverageMotion = epochMeanMotion;
        CurrentMotion = epochMeanMotion;
    #endif
    SemiMajorAxis = 331.25 *
        exp(2*log(MinutesPerDay/CurrentMotion)/3);
    CurrentOrbit = ReferenceOrbit +(CurrentTime-EpochDay)
        *AverageMotion;
    OrbitNum = CurrentOrbit;
    MeanAnomaly = (CurrentOrbit-OrbitNum)*PI2;
    TmpTime = CurrentTime;
    if(MeanAnomaly < PI)
        DidApogee = 0;
    if(PrintApogee && !DidApogee && MeanAnomaly > PI)
    {
        /* Calculate Apogee */
        TmpTime -= StepTime; /* So we pick up later
                               where we left off */
    }
}

```

```

        AverageMotion;
    }

    TrueAnomaly = Kepler(MeanAnomaly,Eccentricity);
    Radius = SemiMajorAxis*(1-SQR(Eccentricity))
        /(1+Eccentricity*cos(TrueAnomaly));
    GetSubSatPoint(EpochDay,EpochRAAN,EpochArgPerigee,
        Inclination,RAANPrecession,PerigeePrecession,
        CurrentTime,TrueAnomaly,&SSPLat,&SSPLong);
    GetBearings(SiteLat,SiteLong,SiteAltitude,SSPLat,
        SSPLong,Radius,&Azimuth,&Elevation,&Range);
    Velocity = (Range-PrevRange)/((CurrentTime-PrevTime)*
        SecondsPerDay);
    PrevRange = Range;
    if(Elevation >= SiteMinElev &&
        CurrentTime >= StartTime)
    {
        Day = CurrentTime + HalfSecond;
        if (((double) Day) > CurrentTime+HalfSecond)
        Day -= 1;    /* Correct for truncation of
            negative values */
        if (OrbitNum == PrevOrbitNum && Day ==
            PrevDay && !PrevVisible)
            printf("\n");/* Dipped out of sight;
                print blank */
        if (OrbitNum != PrevOrbitNum || Day != PrevDay||
            (j>30))

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่อผู้จัดทำเห็นว่าไม่เหมาะสมจะขอถอนเอกสารออกโดยไม่另行通知
 ไม่ว่ากรณี i=0,j=0,y=60; ให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    if(inp)new=1;
    if(!new)getch();
    new=0;
    setfillstyle(SOLID_FILL,LIGHTBLUE);
    bar(x,y-10,x+500,y+410);
    PrintDate(x+20,y,Day);
    sprintf(buf,"ORBIT NUMBER : %ld",OrbitNum);
    outtextxy(x,y+20,buf);
    sprintf(buf,"%s",
"U.T.C.    AZ    EL    DOPPLER    RANGE    HEIGHT    LAT    LONG ");
    outtextxy(x,y+40,buf);
    sprintf(buf,"%s",
"HH:MM    deg    deg    Hz    km    km    deg    deg ");
    outtextxy(x,y+60,buf);
    }
    j++;
    PrevOrbitNum = OrbitNum; PrevDay = Day;
    PrintTime(x,y+80+i*10,CurrentTime + HalfSecond);
    Doppler = -BeaconFreq*Velocity/C;
    sprintf(buf," %12.01f %5.01f %8.01f %8.01f
                %8.01f %6.01f %6.01f",
Azimuth*DegreesPerRadian,Elevation*DegreesPerRadian,
Doppler,Range,Radius-EarthRadius,SSPLat*DegreesPerRadian,
SSPLong*DegreesPerRadian);
    outtextxy(x,y+80+i*10,buf);
    i++;

```



```

{
    double RAAN, ArgPerigee;
    int i;
    ArgPerigee = EpochArgPerigee + (Time-EpochTime)*
        PerigeePrecession;
    *Latitude = asin(sin(Inclination)*
        sin(ArgPerigee+TrueAnomaly));
    RAAN = EpochRAAN - (Time-EpochTime)*RAANPrecession;
    *Longitude = -acos(cos(TrueAnomaly+ArgPerigee)/
        cos(*Latitude));
    if ((Inclination > PI/2 && *Latitude < 0)
        || (Inclination < PI/2 && *Latitude > 0))
        *Longitude = -*Longitude;
    *Longitude += RAAN;
    /* Convert from celestial to terrestrial coordinates */
    *Longitude -= PI2*(Time*SiderealSolar + RefSidereal);
    /* Make West be positive */
    *Longitude = -*Longitude;
    /* i = floor(Longitude/2*pi) */
    i = *Longitude/PI2;
    if(i < 0)
        i--;
    *Longitude -= i*PI2;
}

```

```

GetPrecession(SemiMajorAxis, Eccentricity, Inclination,
    RAANPrecession, PerigeePrecession)

```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์หรือการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้เผยแพร่ไปใช้ประโยชน์ด้านการค้า
ไม่ว่าในรูปแบบใดก็ตาม

```

{
    *RAANPrecession=9.95*pow(EarthRadius/SemiMajorAxis,3.5)*
        cos(Inclination)/SQR(1-SQR(Eccentricity)) *
        RadiansPerDegree;
    *PerigeePrecession = 4.97*
        pow(EarthRadius/SemiMajorAxis,3.5)
        *(5*SQR(cos(Inclination))-1)/SQR(1-SQR(Eccentricity)) *
        RadiansPerDegree;
}
GetBearings(SiteLat,SiteLong,SiteAltitude,SatLat,SatLong,
    Radius,Azimuth,Elevation,Range)
double SiteLat,SiteLong,SiteAltitude,
    SatLat,SatLong,Radius;
double *Azimuth,*Elevation,*Range;
{
    double LongDiff;
    double CosBeta,SinBeta;
    LongDiff = SiteLong - SatLong;
    CosBeta = sin(SiteLat)*sin(SatLat)+cos(SiteLat)*
        cos(SatLat)*cos(LongDiff);
    SinBeta = sqrt(1-SQR(CosBeta));
    *Azimuth = acos((sin(SatLat)- sin(SiteLat)*CosBeta)/
        cos(SiteLat)/SinBeta);
    if (LongDiff < -PI)
        LongDiff += PI2;
    if (LongDiff > PI)
        LongDiff -= PI2;
    if (LongDiff < 0)

```

```

    *Azimuth = PI2 - *Azimuth;

/* Convert to geocentric radius */
SiteAltitude += EarthRadius*
    (1-EarthFlat/2+EarthFlat/2*cos(2*SiteLat));

*Elevation = atan((Radius*CosBeta-(SiteAltitude))/
    (Radius*SinBeta));

*Range=sqrt(SQR(Radius)+
    SQR(SiteAltitude)-2*Radius*SiteAltitude*CosBeta);
}

PrintDate(x,y,Day)
{
    int x,y,Day;
    char buf[30];
    int Month,Year,DayOfWeek;
    DayOfWeek = (Day-RefDay) % 7;
    Year = RefYear;
    while(Day <= 0)
    {
        Year -= 1;
        if (Year%4 == 0) Day += 366;
        else Day += 365;
    }
    while((Year%4 == 0 && Day > 366) ||
        (Year%4 != 0 && Day > 365))
    {
        if (Year%4 == 0) Day -= 366;

```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้สำหรับงานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ หากมีข้อผิดพลาดหรือต้องการแก้ไข กรุณาแจ้งให้ทราบเพื่อดำเนินการปรับปรุงแก้ไข

```

    }

    if (Year%4 == 0) MonthDays[11] += 1; /* Leap year */
    Month = 0;
    while (Day > MonthDays[Month])
    {
        Day -= MonthDays[Month];
        Month += 1;
    }
    sprintf(buf,"%s %d %s %d",DayNames[DayOfWeek],Day,
        MonthNames[Month],Year);
    outtextxy(x,y,buf);
    if (inp)fprintf(fp1,"DATE: %d %d %d\n",Day,Month+1,Year);
    if (Year%4 == 0) MonthDays[11] -= 1; /* Leap year */
}

PrintTime(x,y,Time)
    int x,y;
    double Time;
{
    int day,hours,minutes,seconds;
    char buf[30];
    day = Time;
    Time -= day;
    if (Time < 0) Time += 1.0; /* Correct for truncation problems
                                with negatives */

    hours = Time*24;
    Time -= hours/24.0;
    minutes = Time*MinutesPerDay;
    Time -= minutes/MinutesPerDay;

```

เอกสารนี้เป็นเอกสารของสำนักงานคณะกรรมการการอุดมศึกษา
 ไม่ควรนำเอกสารนี้ไปใช้ประโยชน์ในการค้า
 ไม่ควรนำเอกสารนี้ไปใช้เพื่อการค้าหรือเพื่อประโยชน์อื่นใด
 ไม่ควรนำเอกสารนี้ไปใช้เพื่อการค้าหรือเพื่อประโยชน์อื่นใด

```

seconds = Time*SecondsPerDay;
seconds -= seconds/SecondsPerDay;
sprintf(buf,"%02d:%02d",hours,minutes);
outtextxy(x,y,buf);
if(inp)fprintf(fp1,"TIME: %d %d ",hours,minutes);
}

/* Get the Day Number for a given date. January 1 of the
reference year is day 1. Note that the Day Number may
be negative, if the sidereal reference is in the
future. */
double GetDay(Year,Month,Day)
int Year,Month;
double Day;
{
int TmpYear;
TmpYear = Year;
while(TmpYear > RefYear)
{
TmpYear -= 1;
if(TmpYear%4 == 0) Day += 366;
else Day += 365;
}
while (TmpYear < RefYear)
{
if (TmpYear%4 == 0) Day -= 366;
else Day -= 365;
}
}

```

เอกสารนี้เป็นเอกสารลิขสิทธิ์ในสถาบันการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

if (Year%4 == 0) MonthDays[1] += 1; /* Leap year */
while (Month > 1)
{
    Month -= 1;
    Day += MonthDays[Month-1];
}
if(Year%4 == 0) MonthDays[1] -= 1; /* Leap year */
return Day;
}

GetSatelliteParams(n)
int n;
{
    int EpochYear;
    char *name;
    double EpochHour,EpochMinute,EpochSecond;
    BeaconFreq = 146.0; /* Default value */
    name=&bufn;    outtextxy(200,20,name);
    EpochDay=atof(buf[1]);
    EpochYear = EpochDay / 1000.0;
    EpochDay -= EpochYear*1000.0;
    EpochDay = GetDay(EpochYear,1,EpochDay)+(8.0/24.0);
    /*This is Time Thai*/
    ElementSet=atoi(buf[2]);
    Inclination=atof(buf[3]);
    Inclination *= RadiansPerDegree;
    EpochRAAN=atof(buf[4]);
    EpochRAAN *= RadiansPerDegree;
    Eccentricity=atof(buf[5]);

```

```
{  
    break;  
}
```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

;*****
;*          LOW ORBIT SATELLITE TRACKING PROJECT          *
;*          TELECOMMUNICATION ENGINEERING (3T)           *
;*          KING MONGKUT INSTITUTE OF TECHNOLOGY (LADKRABANG) *
;*****

```

CPU"Z80.TBL"

HOF"INT8"

ORG 0000H

```

PA1:      EQU 30H      ; port A of 1st 8255
PB1:      EQU 31H      ; port B of 1st 8255
PC1:      EQU 32H      ; port C of 1st 8255
PCC1:     EQU 33H      ; control port's 1st 8255
PA2:      EQU 40H      ; port A of 2nd 8255
PB2:      EQU 41H      ; port B of 2nd 8255
PC2:      EQU 42H      ; port C of 2nd 8255
PCC2:     EQU 43H      ; control port's 2nd 8255
PA3:      EQU 10H      ; port A of 3rd 8255
PB3:      EQU 11H      ; port B of 3rd 8255
PC3:      EQU 12H      ; port C of 3rd 8255
PCC3:     EQU 13H      ; control port's 3rd 8255
WRINSC:   EQU 60H      ; write instruction code
WRDATA:   EQU 62H      ; write data to LCD
RDBF:     EQU 64H      ; read busy flag from LCD
RDDATA:   EQU 66H      ; read data from LCD
CPORT:    EQU 0A1H     ; status port of 8251
DPORT:    EQU 0A0H     ; data port of 8251
LINE:     EQU 8010H

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 DATA_RIGHT: EQU 23H
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

DATA_LEFT:      EQU 19H
RTC:            EQU 80H
,
HOUR:           EQU RTC+4      ; hour counter
MINUTE:         EQU RTC+3      ; minute counter
SEC:            EQU RTC+2      ; seconde counter
DATE:           EQU RTC+6      ; ,date counter
MONTH:          EQU RTC+7      ; month counter
ESC_KEY:        EQU 10         ; escape key code
UP_ARROW_KEY:   EQU 11         ; up arrow key code
DOWN_ARROW_KEY: EQU 12         ; down arrow key code
L_ARROW_KEY:    EQU 13         ; left arrow key code
R_ARROW_KEY:    EQU 14         ; right arrow key code
;*****
;*             START PROGRAM                                     *
;*****
START:          LD HL,0
ST1:            DEC HL
                LD A,L
                OR H
                JR NZ,ST1
                LD SP,0FFFFH
                CALL INITIAL      ; initial control board
                LD HL,MOTOR_PAT_AZ
                LD (AZ_PAT_ADDR),HL
                LD HL,MOTOR_PAT_EL
                LD (EL_PAT_ADDR),HL
                JP Pre_MAIN

```

```
;*****
;*          JUMP TO NON MARKABLE SURVICE          *
;*****
```

```
ORG 66H
```

```
JP NMI
```

```
;*****
```

```
Pre_MAIN:      IN A,(PC2)
```

```
AND 3
```

```
CP 11
```

```
JP Z,inited
```

```
CP 01B
```

```
JP Z,Init_AZ
```

```
CP 10B
```

```
JP Z,Init_EL
```

```
CP 00B
```

```
JP Z,Init_EL_AZ
```

```
Pre_MAIN_DL:   CALL ROTATE_DELAY
```

```
JP Pre_MAIN
```

```
Init_EL:       CALL ROTATE_D
```

```
JP Pre_MAIN_DL
```

```
Init_AZ:       CALL ROTATE_L
```

```
JP Pre_MAIN_DL
```

```
Init_EL_AZ:    CALL ROTATE_D
```

```
CALL ROTATE_L
```

```
JP Pre_MAIN_DL
```

```
inited:        LD A,0FFH          ; reset counter
```

```
OUT (PC2),A
```

```
LD BC,0
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

LD (AZIMUTE),BC      ; initial azimuth
                        ; anqle = 0

LD (ELEVATION),BC    ; initial elevation
                        ; anqle = 0

CALL DELAY

CALL DELAY

CALL DELAY

XOR A                ; enable counter

OUT (PC2),A          ;

JP MAIN              ; go to main program

;*****
;*                   SPEED OF INITAIL ANTENNA                   *
;*****

ROTATE_DELAY:        PUSH BC
                        LD B,3H
RDL_1:                PUSH BC
                        LD B,0
RDL:                  DJNZ RDL
                        POP BC
                        DJNZ RDL_1
                        POP BC
                        RET

;*****
;*                   MAIN PROGRAM                               *
;*****

MAIN:  IN A,(CPORT)

```

```

BIT 1,A              ; RxREADY = 1 have

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        JP NZ,MAIN_ACKPOLL ; jump to ackpoll
                                ; when RxREADY = 1

        CALL PROG_MENU        ; show main menu

        CP 1                    ; if select 1 goto

        CALL Z,AUTO_CH        ; auto procedure

        CP 2                    ; if select 2 goto

        CALL Z,MANUAL_CH      ; manual procedure

        CP 3                    ; if select 3 goto

        CALL Z,TIME_CH        ; time set procedure

        JP MAIN

MAIN_ACKPOLL: CALL TxRx        ; call transmission
                                ; and receive routine

        JP MAIN

;*****
;*      INITIAL CONTROL BOARD AND PERIFERAL      *
;*****

INITIAL:  PUSH AF

          PUSH BC

          PUSH HL

          CALL INIT_8255      ; initial 8255

          CALL INIT_8251      ; initial 8251

          CALL INIT_LCD      ; initial LCD

          CALL INIT_RTC      ; initial RTC

          LD A,0FFH          ; disable RTC

          OUT (PA2),A

          LD B,0              ; show title

          LD C,0              ; on LCD

          LD HL,TITLE

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

CALL PRINTF

LD B,20

INCDELAY: CALL DELAY

DJNZ INCDELAY

POP HL

POP BC

POP AF

RET

;*****
;*          setup LCD subprogram          *
;*****
INIT_LCD: PUSH AF
          LD A,00111000B ; function set
          OUT (WRINSC),A
          CALL DELAYL
          CALL DELAYL
          LD A,00001100B ; display on/off
                               ; control
          OUT (WRINSC),A
          CALL DELAYL
          LD A,00000110B ; entry mode set
          OUT (WRINSC),A
          CALL DELAYL
          CALL CLS ; clear screen
          POP AF
          RET

```

;*****

```
INIT_8255:      PUSH AF

                LD A,8AH                ; initial 1st 8255

                OUT (PCC1),A

                LD A,81H                ; initial 2nd 8255

                OUT (PCC2),A

                LD A,9BH                ; initial 3rd 8255

                OUT (PCC3),A

                POP AF

                RET
```

;*****

;* INITIAL 8251 *

;*****

```
INIT_8251:      PUSH AF

                LD A,4EH

                OUT (CPORT),A

                LD A,37H

                OUT (CPORT),A

                POP AF

                RET
```

;*****

;* INITIAL RTC *

;*****

```
INIT_RTC:      PUSH AF

                LD A,04H

                OUT (RTC+11H),A

                IN A,(RTC+10H)

                POP AF
```

```

                                RET

;*****
;*                               MAIN MENU                               *
;*                               output : A = number of choice           *
;*****

PROG_MENU:                    PUSH BC

                                PUSH DE

                                PUSH HL

REWRITE_MENU:                LD BC,0                                ; print main menu on
                                LD HL,MENU                        ; LCD screen
                                CALL PRINTF
                                LD DE,0
                                LD HL,9
                                LD B,0
                                LD C,1
                                CALL GOTO_XY
                                LD C,1
                                CALL CURSOR_ONOFF

LOOP_MENU:                    IN A,(CPORT)

                                BIT 1,A

                                JP NZ,MENU_ACKPOLL

                                CALL SCAN_KEY

                                CP 0FFH

                                JP Z,LOOP_MENU

                                CP 0                                ; check key rang 1-3
                                JP Z,LOOP_MENU                    ; to select choice of
                                CP 4                                ; menu
                                JP C,EXIT_MENU                    ;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

CP 11

JP Z,UP_ARROW

CP 12

JP Z,DOWN_ARROW

CP 15

JR Z,ENTER_CHOICE

JP LOOP_MENU

MENU_ACKPOLL: LD C,0

CALL CURSOR_ONOFF

CALL CLS

CALL TxRx

JP REWRITE_MENU

;-----

UP_ARROW: LD A,C

CP 1 ; line 1 ?

JR Z,BOTTOM

DEC C

JR EXIT_ARROW

BOTTOM: LD C,3

JR EXIT_ARROW

;-----

DOWN_ARROW: LD A,C

CP 3 ; line 3 ?

JR Z,UPPER

INC C

JR EXIT_ARROW

UPPER: LD C,1

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

;-----
ENTER_CHOICE:      LD A,B

                  CP 9

                  JR Z,RIGHT_CHOICE

                  LD A,C

                  JR EXIT_MENU

RIGHT_CHOICE:      LD A,3

                  ADD A,C

                  JR EXIT_MENU

;-----

EXIT_ARROW:        CALL GOTO_XY

                  CALL DELAY

                  CALL DELAY

                  JP LOOP_MENU

EXIT_MENU:          POP HL

                  POP DE

                  POP BC

                  RET

;*****

;*                AUTOMATIC SUBPROGRAM                *

;*****

AUTO_CH:           PUSH AF

                  PUSH BC

                  PUSH HL

                  CALL CLSAND_OFFCUR ; clear screen and

                                      ; off cursor

                  LD BC,0

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 LD HL,AUTOPAT ; show automatic
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

CALL PRINTF          ; pattle

XOR A                ; enable RTC

OUT (PA2),A

CALL DELAY

CALL DELAY

LOOP_AUTO:          CALL SCAN_KEY

CALL DELAY

CP 0FFH

JP Z,LOOP_AUTO

CP ESC_KEY

JP Z,EX_AUTO

CP 9

JP C,SELECT_SAT

JP LOOP_AUTO

SELECT_SAT: CALL SAT_NUM

JP AUTO_CH

EX_AUTO: LD A,0FFH

OUT (PA2),A

POP HL

POP BC

POP AF

RET

```

;*****

;* MANUAL SUBPROGRAM *

;*****

```
MANUAL_CH: PUSH AF
```

```
PUSH BC
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

PUSH HL

CALL CLSAND_OFFCUR

LD BC,0

LD HL,MANAUL

CALL PRINTF

XOR A                      ; enable RTC

OUT (PA2),A

CALL DELAY

CALL DELAY

LOOP_MANUAL: CALL SCAN_KEY

CP OFFH

JP Z,LOOP_MANUAL

CP ESC_KEY

JP Z,EX_MANUAL

CP UP_ARROW_KEY

CALL Z,ROTATE_U

CP DOWN_ARROW_KEY

CALL Z,ROTATE_D

CP L_ARROW_KEY

CALL Z,ROTATE_L

CP R_ARROW_KEY

CALL Z,ROTATE_R

CP 0

CALL Z,LOCK_KEY

JP LOOP_MANUAL

EX_MANUAL: LD A,OFFH          ; disable RTC

```

```
OUT (PA2),A
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
POP HL
```

```

        POP BC

        POP AF

        RET

;*****
;*                                *
;*****

LOCK_KEY:    PUSH AF
             PUSH BC
             PUSH HL
             CALL DELAY
             LD HL,LOCK           ; show "LOCK" on screen
             LD B,0
             LD C,1
             CALL PRINTF
WAIT_Function: CALL SCAN_KEY      ; wait key function
             CP 0FFH
             JP Z,WAIT_Function
             CP 0                ; release lock key
             JP Z,EX_LOCK_KEY
             CP UP_ARROW_KEY     ; lock up key
             JP Z,Pre_LOCK_KEY
             CP DOWN_ARROW_KEY   ; lock down key
             JP Z,Pre_LOCK_KEY
             CP L_ARROW_KEY      ; lock left key
             JP Z,Pre_LOCK_KEY
             CP R_ARROW_KEY      ; lock right key
             JP Z,Pre_LOCK_KEY

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Pre_LOCK_KEY:      CALL LOOP_LOCK_KEY

EX_LOCK_KEY:       LD HL,LOCK_CLR

                   LD B,0

                   LD C,1

                   CALL PRINTF

                   CALL DELAY

                   CALL DELAY

                   POP HL

                   POP BC

                   POP AF

                   RET

;*****
;*                SUBPROGRAM OF LOCK ARROW KEY                *
;*****

LOOP_LOCK_KEY:     PUSH AF

                   PUSH BC

                   LD B,A

                   CALL DELAY

Relock_loop:       CALL SCAN_KEY

                   CP 0

                   JP Z,EX_LOOP_U

                   LD A,B

                   CP UP_ARROW_KEY

                   CALL Z,ROTATE_U

                   CP DOWN_ARROW_KEY

                   CALL Z,ROTATE_D

                   CP L_ARROW_KEY

                   CALL Z,ROTATE_L

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        CP R_ARROW_KEY

        CALL Z,ROTATE_R

        JP Relock_loop

EX_LOOP_U:    POP BC

              POP AF

              RET

;*****
;*              TIME SET SUBPROGRAM              *
;*****

TIME_CH:      PUSH AF
              PUSH BC
              PUSH HL
              CALL CLSAND_OFFCUR
              LD BC,0
              LD HL,TIME_MENU
              CALL PRINTF
              XOR A                      ; enable RTC
              OUT (PA2),A

ESC_TIME:     CALL SCAN_KEY

              CALL DELAY

              CP 10

              JP Z,EX_TIME_CH

              CP 2

              CALL Z,SET_T

              CP 1

              CALL Z,SET_D

              JR ESC_TIME

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น หากฝ่าฝืนมีโทษตามกฎหมาย และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

OUT (PA2),A

POP HL

POP BC

POP AF

RET

;* FIND ABSOLUTE SATELLITE NUMBER *

;* input : A= Number sequence of satellite *

;* output : HL= Satellite name *

FIND_SAT_NUM: PUSH AF
PUSH BC
PUSH DE
LD DE,6
LD HL,SATELLITE
CP 0
JP Z,NOT_MUX
LD B,A

MUX_N: ADD HL,DE

DJNZ MUX_N

NOT_MUX: POP DE

POP BC

POP AF

RET

;* FIND DATA OF SELECTED SATELLITE *

SAT_NUM: PUSH AF

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

PUSH BC

PUSH DE

PUSH HL

PUSH IX

PUSH AF

LD A,0FFH          ; disable RTC

OUT (PA2),A

POP AF

CALL CLS

LD B,0

LD C,0

CALL GOTO_XY

LD HL,SAT_PAT

CALL PRINTF

CALL FIND_SAT_NUM ; show satellite name

LD B,5             ; that select on LCD

                     ; display

LD C,2

CALL GOTO_XY

CALL PRINTF

LD B,12

DELAY_SAT_TAB:    CALL DELAY          ; delay show

                  DJNZ DELAY_SAT_TAB ; satellite name on

                                      ; screen

LD BC,0

CALL GOTO_XY

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

CALL PRINTF          ; show "please wait"
                      ; on screen

POP HL

CALL PRINTF          ; show satellite name
                      ; that select on LCD
                      ; display

PUSH AF

XOR A                ; enable RTC

OUT (PA2),A

POP AF

LD IX,RS_10

LD B,A

LD DE,0A00H

CP 0

CALL Z,ACCESS_SAT_BUF

FIND_SAT_BUF: ADD IX,DE

DJNZ FIND_SAT_BUF

CALL ACCESS_SAT_BUF

POP IX

POP HL

POP DE

POP BC

POP AF

RET

;*****

;*

;* FIND DATA IN SATELLITE THAT SELECT

;*. input : IX = start address buffer of selected satellite
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
;*****
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

```
ACCESS_SAT_BUF:  PUSH AF
                  PUSH BC
                  PUSH DE
                  PUSH HL
                  PUSH IX
                  XOR A
                  OUT (PA2),A
```

Refind:

```
                  INC IX
                  INC IX
                  LD A,(IX+0)
                  LD B,A
                  LD A,(IX+1)
                  AND B
                  CP "D"
                  JP Z,CHK_MONTH
                  CP "H"
                  JP Z,CHK_HOUR
                  CP 0FFH
                  JP Z,END_DATA
                  INC IX
                  JP Refind
```

CHK_MONTH: INC IX

```
Recheck_month:  INC IX
                  LD A,(IX+1)
                  LD L,A
```

```
                  CALL HEX2BCD_TEN
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        IN A,(MONTH)          ; current month

        CP D

        JP C,WAIT_M_EQU       ; current month
                                ; < program month
                                ; then goto wait
                                ; until equal

        JP Z,CHK_DATE         ; current month =
                                ; program month
                                ; then goto check date

Refind_DD:    INC IX          ; current month > program m
Find_1st_D:   INC IX          ; then refind DD
                LD A,(IX+0)
                CP "D"
                JP NZ,Find_1st_D
                INC IX
                LD A,(IX+0)
                CP "D"
                JP NZ,Find_1st_D
                JP Recheck_month

WAIT_M_EQU:   CALL SCAN_KEY

                CP ESC_KEY

                JP Z,EX_FIND

                IN A,(MONTH)

                CP D

                JP Z,CHK_DATE

                JP WAIT_M_EQU

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

LD L,A
CALL HEX2BCD_TEN
LD D,E
IN A,(DATE)
CP D
JP C,WAIT_D_EQU
JP Z,Pre_Refind
JP Refind_DD
Pre_Refind: INC IX
JP Refind
WAIT_D_EQU: CALL SCAN_KEY
CP ESC_KEY
JP Z,EX_FIND
IN A,(MONTH)
CP D
JP Z,CHK_HOUR
JP WAIT_D_EQU

```

```

CHK_HOUR: INC IX
INC IX
LD A,(IX+0)
LD L,A
CALL HEX2BCD_TEN
LD D,E
IN A,(HOUR)
CP D
JP C,WAIT_H_EQU
JP Z,CHK_MINUTE

```

```

INC IX

INC IX

JP Refind

WAIT_H_EQU: CALL SCAN_KEY

CP ESC_KEY

JP Z,EX_FIND

IN A,(HOUR)

CP D

JP Z,CHK_MINUTE

JP WAIT_H_EQU

;-----

CHK_MINUTE: LD A,(IX+1)

LD L,A

CALL HEX2BCD_TEN

LD D,E

IN A,(MINUTE)

CP D

JP C,WAIT_MM_EQU

JP Z,Rotate_motor

INC IX

INC IX

JP Refind

WAIT_MM_EQU: CALL SCAN_KEY

CP ESC_KEY

JP Z,EX_FIND

IN A,(MINUTE)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

JP WAIT_MM_EQU

END_DATA: LD B,0

 LD C,2

 LD HL,Found_not_tab

 CALL PRINTF

 CALL DELAY

 JP EX_FIND

Rotate_motor: LD A,0FFH

 OUT (PA2),A

 INC IX

 INC IX

 INC IX

 INC IX

 LD B,0

 LD C,2

 LD HL,AZ_EL

 CALL PRINTF

STEP_TRACK: CALL VECTOR ; calculate absolute

 ; vector

RUN_TRACK: LD L,(IX+0)

 LD H,(IX+1)

 LD (AZIMUTE),HL

 LD L,(IX+2)

 LD H,(IX+3)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

CALL TRACKING

INC IX

INC IX

INC IX

INC IX

LD A,(IX+0)

CP "A"
JP NZ,Refind

INC IX
LD A,(IX+0)
CP "Z"
JP NZ,Refind
XOR A
OUT (PA2),A
IN A,(MINUTE)
LD B,A
WAIT_TRACK: IN A,(MINUTE)
CP B
JP Z,WAIT_TRACK

INC IX

JP STEP_TRACK

EX_FIND: POP IX

POP HL

POP DE

POP BC

POP AF

```

RET

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

;*          FIND ABSOLUTE VECTOR          *
;*    input : IX = current index buffer    *
;*    output : RESULT_AZ = absolute azimuth vector value *
;*          : RESULT_EL = absolute elevation vector value*
;*    if bit 15 of absolute value is set that reverse *
;*    direction else forward direction        *
;*****

```

```

VECTOR:      PUSH DE
              PUSH HL
              CALL put_angle
              LD L,(IX+0)
              LD H,(IX+1)
              LD DE,(AZIMUTE)
              CALL SUB_TWO_DIGIT
              JP C,REV_AZ
              LD (RESULT_AZ),HL ; save absolute
                                ; azimuth in
                                ; BUFFER AZ
              JP FORW_EL

```

```

REV_AZ:      LD E,(IX+0)
              LD D,(IX+1)
              LD HL,(AZIMUTE)
              CALL SUB_TWO_DIGIT
              DFB 0CBH,0FCH ; SET 7,H
              LD (RESULT_AZ),HL

```

```

FORW_EL:     LD L,(IX+2)
              LD H,(IX+3)

```

```

              LD DE,(ELEVATION)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

CALL SUB_TWO_DIGIT

JP C,REV_EL

LD (RESULT_EL),HL ; save absolute
                    ; elevation in
                    ; BUFFER EL0

JP EX_VECTOR

REV_EL:             LD E,(IX+2)
                    LD D,(IX+3)
                    LD HL,(ELEVATION)
                    CALL SUB_TWO_DIGIT
                    DFB 0CBH,0FCH ; SET 7,H
                    LD (RESULT_EL),HL

EX_VECTOR:          POP HL
                    POP DE
                    RET

;*****
;*                  AUTOMATIC TRACKING                  *
;* input: (RESULT_AZ) = absolute azimuth angle to track *
;*          (RESULT_EL) = absolute elevation angle to track*
;*****

TRACKING:            PUSH AF
                    PUSH BC
                    PUSH DE
                    PUSH HL

RE_TRACK:            LD HL,(RESULT_AZ)

                    RES 7,H

                    LD DE,0

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

; azimuth angle=0?
JP Z,ONLY_EL ; if = 0 then rotate
; only elevation

LD HL,(RESULT_EL)
RES 7,H
LD DE,0
CALL SUB_TWO_DIGIT ; compare absolute
; elevation angle=0?
JP Z,ONLY_AZ ; if = 0 then rotate
; only azimuth
JP ROTATE_AZ_EL
;+++++
ONLY_EL: LD HL,(RESULT_EL)
;
BIT 7,H
JP NZ,RO_REV_EL
LD DE,0
CALL SUB_TWO_DIGIT ; compare absolute
; elevation angle=0?
JP Z,NEXT_CYCLE
LD DE,1
LOOP_EL: CALL ONE_U
SCF
INC HL
SBC HL,DE
JP NZ,LOOP_EL
JP NEXT_CYCLE

```

```
RO_REV_EL: LD DE,0
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

CALL SUB_TWO_DIGIT
JP Z,NEXT_CYCLE
LD DE,1
LOOP_REV_EL: CALL ONE_D
SCF
INC HL
SBC HL,DE
JP NZ,LOOP_REV_EL
JP NEXT_CYCLE
;+++++
ONLY_AZ: LD HL,(RESULT_AZ)
BIT 7,H
JP NZ,RO_REV_AZ
LD DE,0
CALL SUB_TWO_DIGIT ; compare absolute
; azimuth angle= 0 ?
JP Z,NEXT_CYCLE
LD DE,1
LOOP_AZ: CALL ONE_R
SCF
INC HL
SBC HL,DE
JP NZ,LOOP_AZ
JP NEXT_CYCLE
RO_REV_AZ: LD DE,0
RES 7,H

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

LD DE,1
LOOP_REV_AZ: CALL ONE_L
SCF
INC HL
SBC HL,DE
JP NZ,LOOP_REV_AZ
JP NEXT_CYCLE
;+++++
ROTATE_AZ_EL: LD HL,(RESULT_AZ)
LD BC,(RESULT_EL)
LD A,H
AND 10000000B
LD H,A
LD A,B
AND 10000000B
SRL A
OR H
CP 0
JP Z,PRE_RU
CP 40H
JP Z,PRE_RD
CP 80H
JP Z,PRE_LU
CP 0C0H
JP Z,PRE_LD
JP NEXT_CYCLE

```

```
PRE_RU: LD B,22
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

```
LEFT_UP: CALL ROTATE_R
```

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

CALL ROTATE_U
CALL TRACK_DL
DJNZ LEFT_UP
LD HL,(RESULT_AZ)
DEC HL
LD (RESULT_AZ),HL
LD HL,(RESULT_EL)
DEC HL
LD (RESULT_EL),HL
JP RE_TRACK
PRE_RD: LD B,22
LEFT_DOWN: CALL ROTATE_R
CALL ROTATE_D
CALL TRACK_DL
DJNZ LEFT_DOWN
LD HL,(RESULT_AZ)
DEC HL
LD (RESULT_AZ),HL
LD HL,(RESULT_EL)
DEC HL
LD (RESULT_EL),HL
JP RE_TRACK
PRE_LU: LD B,22
RIGHT_UP: CALL ROTATE_L
CALL ROTATE_U
CALL TRACK_DL

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องส่งคืนถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

DEC HL

LD (RESULT_AZ),HL

LD HL,(RESULT_EL)

DEC HL

LD (RESULT_EL),HL

JP RE_TRACK

PRE_LD:      LD B,22

RIGHT_DOWN:  CALL ROTATE_L
              CALL ROTATE_D
              CALL TRACK_DL
              DJNZ RIGHT_DOW N
              LD HL,(RESULT_AZ)
              DEC HL
              LD (RESULT_AZ),HL
              LD HL,(RESULT_EL)
              DEC HL
              LD (RESULT_EL),HL
              JP RE_TRACK

NEXT_CYCLE:  POP HL

              POP DE

              POP BC

              POP AF

              RET

```

```

;*****

```

```

;*          SUB TWO VALUE          *

```

```

;*          input : HL = stander    *

```

```

;*          : DE = subtractor       *

```

```

;*          output : HL = result    *

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

;*          : Carry flag = 1  ; negative result  *
;*          = 0   ; positive result  *
;*          : Zero flag = 1  ; zero result      *
;*****

```

```

SUB_TWO_DIGIT:  PUSH DE

```

```

                SCF                ; set carry flag

```

```

                INC HL

```

```

                SBC HL,DE           ; HL=HL-DE-carry flag

```

```

                POP DE

```

```

                RET

```

```

;*****

```

```

;*          PUT AZEMUTE & ELEVATION ANGLE ON SCREEN  *

```

```

;*****

```

```

put_angle:

```

```

                PUSH AF

```

```

                PUSH BC

```

```

                PUSH HL

```

```

                LD A,(IX+0)

```

```

                LD L,A

```

```

                LD A,(IX+1)

```

```

                LD H,A

```

```

                LD B,3

```

```

                LD C,3

```

```

                CALL PUT_BCD

```

```

                LD A,(IX+2)

```

```

                LD L,A

```

```

                LD A,(IX+3)

```

```

                LD H,A

```

```

                LD B,11

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

LD C,3

CALL PUT_BCD

POP HL

POP BC

POP AF

RET

;*****
;*          ROTATE ELEVATION UP ONE DEGREE          *
;*****
ONE_U:      PUSH BC
            LD B,22
RO_U:      CALL ROTATE_U
            CALL TRACK_DL
            DJNZ RO_U
            POP BC
            RET

;*****
;*          ROTATE ELEVATION DOWN ONE DEGREE          *
;*****
ONE_D:      PUSH BC
            LD B,22
RO_D:      CALL ROTATE_D
            CALL TRACK_DL
            DJNZ RO_D
            POP BC
            RET

```

```

;*****
ONE_L:          PUSH BC

                LD B,22

RO_L:           CALL ROTATE_L

                CALL TRACK_DL

                DJNZ RO_L

                POP BC

                RET
;*****
;*              ROTATE AZIMUTE RIGHT ONE DEGREE              *
;*****
ONE_R:          PUSH BC

                LD B,22

RO_R:           CALL ROTATE_R

                CALL TRACK_DL

                DJNZ RO_R

                POP BC

                RET
;*****
;*              TRACK DELAY                                    *
;*****
TRACK_DL:       PUSH BC

                LD B,15

TDL_2:          PUSH BC

                LD B,0

TDL_1:          DJNZ TDL_1

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        POP BC

        RET

;*****

;*          SET TIME SUBPROGRAM          *

;*****

SET_T:      PUSH AF

            PUSH BC

            PUSH DE

            LD A,0FFH          ; disable RTC

            OUT (PA2),A

            LD B,8

            LD C,1

            CALL GOTO_XY

            CALL CURSOR_ONOFF

W_SET_T:    CALL SCAN_KEY

            CALL DELAY

            CP ESC_KEY

            JP Z,EX_SET_T

            CP L_ARROW_KEY

            CALL Z,SHF_CUR_L

            CP R_ARROW_KEY

            CALL Z,SHF_CUR_R

            CP 10

            CALL C,FILL_NUM

            CP 15

            JP Z,ENTER_T

            JP W_SET_T

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น ยกเว้นแต่มีเหตุเปลี่ยนแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

ENTER_T:      LD B,8

               CALL SHIFTL_DATA

               LD L,D

               INC B

               CALL GOTO_XY

               CALL GETCHAR

               LD A,D

               SUB 30H

               ADD A,L

               OUT (HOUR),A

               LD B,11

               CALL SHIFTL_DATA

               LD L,D

               INC B

               CALL GOTO_XY

               CALL GETCHAR

               LD A,D

               SUB 30H

               ADD A,L

               OUT (MINUTE),A

               LD B,14

               CALL SHIFTL_DATA

               LD L,D

               INC B

               CALL GOTO_XY

               CALL GETCHAR

               LD A,D

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

CALL DELAY
CP ESC_KEY
JP Z,EX_SET_D
CP L_ARROW_KEY
CALL Z,SHF_CUR_L
CP R_ARROW_KEY
CALL Z,SHF_CUR_R
CP 10
CALL C,FILL_NUM
CP 15
JP Z,ENTER_D
JP W_SET_D
;+++++
ENTER_D: LD B,8
CALL SHIFTL_DATA
LD L,D
INC B
CALL GOTO_XY
CALL GETCHAR
LD A,D
SUB 30H
ADD A,L
OUT (DATE),A
LD B,11
CALL SHIFTL_DATA
LD L,D
INC B
CALL GOTO_XY

```

```
JP NZ,CUR_SHR
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับคนไข้ก่อนเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
CUR_SHR:
```

```

CP 9
JP Z,JMP_DOTR
CP 12
JP Z,JMP_DOTR

```

```

JP NO_DOTR

JMP_DOTR:      INC B

NO_DOTR:       INC B

C_SHR:        CALL GOTO_XY

              POP AF

              RET

```

```

;*****

```

```

FILL_NUM:     PUSH BC

              ADD A,30H

              LD D,A

              CALL PUTCHAR

              POP BC

              CALL GOTO_XY

              RET

```

```

;*****

```

```

;*           input : B = colume           *
;*           : C = line                   *
;*           output: D = data shifted     *

```

```

;*****

```

```

SHIFTL_DATA:  PUSH BC

              CALL GOTO_XY

              CALL GETCHAR

              LD B,4

SHL_DATA:     SLA D

              DJNZ SHL_DATA

              POP BC

              RET

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

; *                                CLEAR SCREEN & OFF CURSOR                                *
; ****

CLSAND_OFFCUR:    PUSH BC

                  CALL CLS

                  LD C,0

                  CALL CURSOR_ONOFF

                  POP BC

                  RET

; ****

DELAYL:           PUSH AF

                  PUSH BC

                  LD B,0

DELAYL1:          DJNZ DELAYL1

                  POP BC

                  POP AF

                  RET

; ****

; *                                CHECK BUSY FLAG LCD                                *
; ****

CHKBF:            PUSH AF

CHKBF1:           IN A,(RDBF)

                  BIT 7,A

                  JR NZ,CHKBF1

                  POP AF

                  RET

; ****

```

เอกสารนี้ * เอกสารที่สงวนไว้สำหรับ utility subprogram of LCD ให้นำไปใช้ประโยชน์ด้านการทำ
ไม่ว่ากรณีใดๆ ****

```

LINE0:          LD A,0

                ADD A,B

                JR GOTO

LINE1:          LD A,40H

                ADD A,B

                JR GOTO

LINE2:          LD A,10H

                ADD A,B

                JR GOTO

LINE3:          LD A,50H

                ADD A,B

GOTO:           DFB 0CBH,0FFH

                OUT (WRINSC),A

                CALL CHKBF

                POP BC

                POP AF

                RET

```

```

;*****
;*          PUT CHARACTER ON CURSOR POSITION          *
;*          input :  D=data                          *
;*****

```

```

PUTCHAR:        PUSH AF

                LD A,D

                OUT (WRDATA),A

                CALL CHKBF

                POP AF

                RET

```

```

;*          GET CHARACTER ON CURSOR POSITION          *
;*          output      D=data                      *
;*****
GETCHAR:      PUSH AF
              IN A,(RDDATA)
              CALL CHKBF
              LD D,A
              POP AF
              RET
;*****
;*          SET CURSOR ON-OFF                      *
;*          input      C=1 :cursor on              *
;*                   =0 :cursor off                *
;*****
CURSOR_ONOFF: PUSH AF
              PUSH BC
              LD A,C
              CP 0
              JR Z,SET_OFF
              CP 1
              JR Z,SET_ON
              JR EX_SETCUR
SET_OFF:      LD A,00001100B      ; set off cursor
              OUT (WRINSC),A
              CALL DELAYL
              JR EX_SETCUR
SET_ON:       LD A,00001101B      ; set on cursor
              OUT (WRINSC),A

```

```

CALL DELAYL
EX_SETCUR:    POP BC
              POP AF
              RET

;*****
;*              PRINT STRING              *
;*              input:    B = X          *
;*              C = Y          *
;*              HL = STRING          *
;*****
PRINTF:       PUSH AF
              PUSH BC
              PUSH DE
              PUSH HL
NEXT_STRING:  CALL GOTO_XY
              LD A,(HL)
              CP 0FFH
              JR Z,EFTX
              CP 13
              JR Z,CR              ; Carriage return
              LD D,A
              CALL PUTCHAR
              INC HL
              INC B
              JR NEXT_STRING
CR:           LD B,0
              INC C
              INC HL

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

JR NEXT_STRING

EOFTX:      POP HL

            POP DE

            POP BC

            POP AF

            RET

;+++++
TxRx:      PUSH AF

            PUSH BC

            PUSH DE

            PUSH HL

            IN A,(DPORT)

            CP 40H

            JP Z,FUNC0

            CP 41H

            JP Z,FUNC1

            CP 42H

            JP Z,FUNC2

            JP END_TRANSFER

```

```

-----

FUNC0:      CALL ACK

            LD B,0

            LD C,1

            LD HL,PCL

            CALL PRINTF

```

```

WAIT_DATA:  CALL CHECK

            IN A,(DPORT)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

CALL Z,ROTATE_R
CP 01
CALL Z,ROTATE_L
CP 10H
CALL Z,ROTATE_U
CP 11H
CALL Z,ROTATE_D
CP 0FFH
JP Z,END_TRANSFER
JP WAIT_DATA
*****
FUNC1: CALL ACK
CALL CHK_READ
LD L,A
CALL ACK
CALL CHK_READ
LD H,A
LD (PRG_AZ),HL
CALL ACK
CALL CHK_READ
LD L,A
CALL ACK
CALL CHK_READ
LD H,A
LD (PRG_EL),HL
LD HL,PRG_ANGLE

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

CALL put_angle
CALL VECTOR
CALL TRACKING
LD HL,(PRG_AZ)
LD (AZIMUTE),HL
LD HL,(PRG_EL)
LD (ELEVATION),HL
JP END_TRANSFER

;+++++
FUNC2:      LD HL,D_TRANSFER
            LD B,0
            LD C,0
            CALL GOTO_XY
            CALL CLS
            CALL PRINTF
            LD HL,RS_10
            CALL FIND_ST
SAVE_ST:    LD (SAT_BUF_ADDR),HL
            LD A,"S"
            LD (HL),A
            INC HL
            LD A,"T"
            LD (HL),A
Re_find_FF: CALL ACK
            CALL CHK_READ      ; read data from 8251
            CP 0FFH            ; check it is an end
                                ; of data ?
            JP NZ,CHK_S        ; if not ,then check

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

CALL CHK_READ

CP "T"                ; 2nd code

JP Z,EX_FIND_ST

JP RE_FIND

EX_FIND_ST:          POP AF

RET

;*****

;*          WAIT DATA AND READ DATA FROM PORT          *

;*          input:  A = data                               *

;*****

CHK_READ:            CALL CHECK

                     IN A,(DPORT)

                     RET

;*****

;*          SHOW 2 DIGIT ON SCREEN                          *

;*          input  :  B = colume                            *

;*                  C = line                                *

;*          A = value to show                               *

;*****

SHW_2_DIGIT:         PUSH AF

                     PUSH HL

                     LD L,A

                     CALL HEX2BCD_TEN

                     LD L,E

                     CALL PUT_2BCD

                     POP HL

                     POP AF

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

CALL put_angle
CALL VECTOR
CALL TRACKING
LD HL,(PRG_AZ)
LD (AZIMUTE),HL
LD HL,(PRG_EL)
LD (ELEVATION),HL
JP END_TRANSFER
;+++++
FUNC2: LD HL,D_TRANSFER
LD B,0
LD C,0
CALL GOTO_XY
CALL CLS
CALL PRINTF
LD HL,RS_10
CALL FIND_ST
SAVE_ST: LD (SAT_BUF_ADDR),HL
LD A,"S"
LD (HL),A
INC HL
LD A,"T"
LD (HL),A
Re_find_FF: CALL ACK
CALL CHK_READ ; read data from 8251
CP OFFH ; check it is an end
; of data ?
JP NZ,CHK_S ; if not ,then check

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

INC HL

LD (HL),A

LD HL,(SAT_BUF_ADDR)

LD DE,0A00H

ADD HL,DE

JP SAVE_ST

SAVE_ONE_BYTE:    INC HL

LD (HL),A

JP Re_find_FF

SAVE_TWO_BYTE:    INC HL

LD (HL),B

INC HL

LD (HL),A

JP RE_FIND_FF

SAVE_END_CODE:    INC HL

LD A,0FFH

LD (HL),A

INC HL

LD (HL),A

LD B,0

LD C,1

CALL GOTO_XY

LD HL,EXPCL

CALL PRINTF

JP END_TRANSFER

```

END_TRANSFER: CALL CLS

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

LD C,0

LD HL,EXPCL

CALL PRINTF

LD B,10H

DLYY:      CALL KEY_DELAY

          DJNZ DLYY

          CALL CLS

          POP HL

          POP DE

          POP BC

          POP AF

          RET

;*****
;*          FIND "ST" CODE FROM PC          *
;*****

FIND_ST:   PUSH AF

RE_FIND:   CALL ACK

          CALL CHECK          ; check status data
                                ; register of 8251

          IN A,(DPORT)        ; if have data read
                                ; to port A

          CP "S"              ; 1st code of initial
                                ; data transfer

          JP Z,FIND_SAT_CODE   ; if found 1st code
                                ; then find 2nd code

          JP RE_FIND           ; but not found then
                                ; goto refind.

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 FIND_SAT_CODE: CALL ACK
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

CALL CHK_READ

CP "T"                ; 2nd code

JP Z,EX_FIND_ST

JP RE_FIND

EX_FIND_ST:          POP AF

RET

;*****

;*                WAIT DATA AND READ DATA FROM PORT                *

;*                input:  A = data                                     *

;*****

CHK_READ:            CALL CHECK

                     IN A,(DPORT)

                     RET

;*****

;*                SHOW 2 DIGIT ON SCREEN                               *

;*                input  :  B = colume                                *

;*                C = line                                             *

;*                A = value to show                                    *

;*****

SHW_2_DIGIT:         PUSH AF

                     PUSH HL

                     LD L,A

                     CALL HEX2BCD_TEN

                     LD L,E

                     CALL PUT_2BCD

                     POP HL

                     POP AF

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

;*****
;*          PUT SATELLITE NAME ON SCREEN          *
;*****
PUT_SAT_NAME:  PUSH BC

                LD B,10

                LD C,1

                CALL GOTO_XY

                CALL PRINTF

                POP BC

                RET

;*****
;*          CHECK RxREADY STATUSOF 8251          *
;*****
CHECK:          PUSH AF
CHECK1:         IN A,(CPORT)
                BIT 1,A
                JR Z,CHECK1
                POP AF
                RET

;*****
;*          SEND ACKNOWLEDGE TO PC              *
;*****
ACK:            PUSH AF
ACK1:           IN A,(CPORT)
                BIT 0,A
                JR Z,ACK1
                LD A,41H
                OUT (DPORT),A

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        POP AF

        RET

;*****
;*                SCAN KEY SUBPROGRAM                *
;*                output:  A = keycode                *
;*****

SCAN_KEY:    PUSH BC
              PUSH DE
              PUSH HL
              LD HL,LINE
              LD (HL),0FEH
              LD C,0
CALL_SCAN:    CALL SCAN
INPORT:       IN A,(PC1)
              AND 0F0H
              CP 0F0H
              JR NZ,PRESS      ; keypress and goto
                                   ; encode keycode
              INC C

              LD A,04
              CP C
              JR Z,NOPRESS
              JR CALL_SCAN

PRESS:        CALL KEYCODE
              JR EXSCKEY

NOPRESS:      LD A,0FFH

EXSCKEY:      CALL KEY_DELAY
              POP HL

```

```

POP DE

POP BC

RET

+++++

KEYCODE:      PUSH BC

                XOR C

                LD C,0

                LD HL,KEYTAB

NEXCODE:      LD B,(HL)

                CP B

                JR Z,KEYEQU

                INC C

                INC HL

                JR NEXCODE

KEYEQU:      LD A,C

                POP BC

                RET

;+++++

SCAN:      PUSH AF

            PUSH HL

            LD HL,LINE

            LD A,(HL)

            RLCA

            CP 0FEH

            JR NZ,OUTPORT

            LD A,0FEH

OUTPORT:      LD (HL),A

```

```

        POP HL

        POP AF

        RET

;*****
;*
;*          KEY DELAY ROUTINE
;*
;*****

KEY_DELAY:    PUSH BC

               CALL DELAY

               LD C,1FH

               LD B,00

LOOP1:        DJNZ LOOP1

               DEC C

               JR NZ,LOOP1

               POP BC

               RET

;*****
;*
;*          CONTROL STEPPING MOTOR
;*
;*
;*          input:    D = direction
;*
;*                      00  RIGHT
;*
;*                      01  LEFT
;*
;*                      10  UP
;*
;*                      11  DOWN
;*
;*****

DRV:          PUSH AF

               PUSH BC

               PUSH DE

               PUSH HL

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

LD A,D
CP 00
JP Z,RIGHT
CP 01
JP Z,LEFT
CP 10
JP Z,UP
CP 11
JP Z,DOWN
JP EXIT_DRV
RIGHT: LD HL,(AZ_PAT_ADDR)
INC HL
LD A,(HL)
CP 0FH
JP Z,RR
JP MOVE_EXIT_AZ
RR: LD HL,MOTOR_PAT_AZ
INC HL
LD A,(HL)
JP MOVE_EXIT_AZ

```

```

LEFT: LD HL,(AZ_PAT_ADDR)
LD A,(HL)
CP 0
JP Z,RL
CP 1

```

```
JP Z,RL
```

```
DEC HL
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

LD A, (HL)
JP MOVE_EXIT_AZ
RL: LD HL, MOTOR_PAT_AZ+8
LD A, (HL)
JP MOVE_EXIT_AZ

```

```

;-----
UP: LD HL, (EL_PAT_ADDR)
INC HL
LD A, (HL)
CP 0FH
JP Z, RUP
JP MOVE_EXIT_EL
RUP: LD HL, MOTOR_PAT_EL
INC HL
LD A, (HL)
JP MOVE_EXIT_EL

```

```

;-----
DOWN: LD HL, (EL_PAT_ADDR)
LD A, (HL)
CP 0
JP Z, RD
CP 1
JP Z, RD
DEC HL
LD A, (HL)
JP MOVE_EXIT_EL

```

```

RD: LD HL, MOTOR_PAT_EL+8
LD A, (HL)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

MOVE_EXIT_EL:      LD B,A

                   LD (EL_PAT_ADDR),HL

                   LD HL,(AZ_PAT_ADDR)

                   LD A,(HL)

                   AND 0FH

                   OR B

                   OUT (PA1),A

                   JP EXIT_DRV

MOVE_EXIT_AZ:      LD B,A

                   LD (AZ_PAT_ADDR),HL

                   LD HL,(EL_PAT_ADDR)

                   LD A,(HL)

                   AND 0FH

                   OR B

                   OUT (PA1),A

EXIT_DRV:          POP HL

                   POP DE

                   POP BC

                   POP AF

                   RET

```

```

;*****

```

```

;*                ROTATE LEFT ONE STEP                *

```

```

;*****

```

```

ROTATE_L:          PUSH DE

                   LD D,1

                   CALL DRV

```

```

                   POP DE

```

```

                   RET

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

;*****

;* ROTATE RIGHT ONE STEP *

;*****

ROTATE_R: PUSH DE

LD D,0

CALL DRV

POP DE

RET

;*****

;* ROTATE UP ONE STEP *

;*****

ROTATE_U: PUSH DE

LD D,10

CALL DRV

POP DE

RET

;*****

;* ROTATE DOWN ONE STEP *

;*****

ROTATE_D: PUSH DE

LD D,11

CALL DRV

POP DE

RET

;*****

;* DELAY SUB PROGRAM *

;*****

DELAY: PUSH BC

```

LD B,0BFH

LOOP_DELAY0:  PUSH BC

LD B,0FFH

LOOP_DELAY:   DJNZ LOOP_DELAY

POP BC

DJNZ LOOP_DELAY0

POP BC

RET

;*****

;*          INPUT : BC = X-Y coordinate          *

;*          D = MARK                             *

;*****

TWO MARK:     PUSH BC

              PUSH DE

              CALL GOTO_XY

              CALL PUTCHAR

              LD A,B

              ADD A,3

              LD B,A

              CALL GOTO_XY

              CALL PUTCHAR

              POP DE

              POP BC

              RET

;*****

;*          input   D = data                      *

;*          output  D = hdata                     *

;*          E = ldata                             *

```

;*****

BCD2ASCII: PUSH AF

 PUSH BC

 LD A,D

 AND 00001111B

 ADD A,30H

 LD E,A

 LD A,D

 LD B,4

SHLDATA: SRL A

 DJNZ SHLDATA

 ADD A,30H

 LD D,A

 POP BC

 POP AF

 RET

;*****

;* CONVERT HEX TO BCD *

;* input : HL = hex number to convert *

;* output : DE = BCD number is converted *

;*****

HEX2BCD: PUSH AF

 PUSH BC

 PUSH HL

 LD D,0

 LD BC,1000

THOUNSAND: SBC HL,BC

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

JP C,F_HUNDRES
INC D
JP THOUNSAND
F_HUNDRES: ADD HL,BC
LD B,4
SHL_THOUSAND: SLA D
DJNZ SHL_THOUSAND
LD A,D
LD E,0
LD BC,100
HUNDREST: SBC HL,BC
JP C,F_TEN
INC E
JP HUNDREST
F_TEN: ADD HL,BC
OR E
LD D,A
LD A,L
LD E,0
TEN: SUB 10
JP C,F_UNIT
INC E
JP TEN
F_UNIT: ADD A,10
LD B,4
SH_TO_TEN: SLA E
DJNZ SH_TO_TEN

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

LD E,A

POP HL

POP BC

POP AF

RET

;*****
;*          CONVERT HEX TO BCD          *
;*          input : L = hex number to convert          *
;*          output : E = BCD number is converted          *
;*****
HEX2BCD_TEN:    PUSH AF
                PUSH BC
                PUSH HL
                LD A,L
                LD E,0
DIGIT2_TEN:     SUB 10
                JP C,DIGIT2_F_UNIT
                INC E
                JP DIGIT2_TEN
DIGIT2_F_UNIT:  ADD A,10
                LD B,4
SHF_TO_TEN:     SLA E
                DJNZ SHF_TO_TEN
                OR E
                LD E,A
                POP HL
                POP BC

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

RET

;*****
;*          PRINT BCD ON LCD          *
;*          input:  HL = HEX number to print BCD      *
;*          B = colume                  *
;*          C = line                    *
;*****

PUT_BCD:      PUSH BC
              PUSH DE
              PUSH HL
              CALL HEX2BCD              ; BCD number in RegDE
              LD L,E                    ; save low diqit in
                                      ; Reg L
              CALL BCD2ASCII
              CALL GOTO_XY
              CALL PUTCHAR
              INC B
              CALL GOTO_XY
              LD D,E
              CALL PUTCHAR
              LD D,L
              CALL BCD2ASCII
              INC B
              CALL GOTO_XY
              CALL PUTCHAR
              INC B
              CALL GOTO_XY
              LD D,E

```

CALL PUTCHAR

POP HL

POP DE

POP BC

RET

;* PRINT BCD ON LCD *

;* input: L = number to print BCD *

;* B = colume *

;* C = line *

PUT_2BCD: PUSH BC

PUSH DE

PUSH HL

LD D,L

CALL BCD2ASCII

CALL GOTO_XY

CALL PUTCHAR

INC B

CALL GOTO_XY

LD D,E

CALL PUTCHAR

POP HL

POP DE

POP BC

RET

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับกรใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่าวิธีใดทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

;*                C = line                *
;*                A = number to put        *
;*****

```

```

PUT2NUM:          PUSH AF
                  PUSH BC
                  PUSH DE
                  LD D,A
                  CALL BCD2ASCII
                  DEC B
                  CALL GOTO_XY
                  CALL PUTCHAR
                  LD D,E
                  CALL PUTCHAR
                  INC B
                  CALL GOTO_XY
                  POP DE
                  POP BC
                  POP AF
                  RET

```

```

;*****

```

```

;*                B = COLUME NUMBER        *

```

```

;*                C = LINE NUMBER          *

```

```

;*****

```

```

TIME:            PUSH AF
                  PUSH BC
                  PUSH DE

```

```

                  INC B

```

```

                  LD D,":"

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

CALL TWOMARK
IN A,(HOUR)
DEC B
CALL PUT2NUM
IN A,(MINUTE)
INC B
INC B
INC B
CALL PUT2NUM
IN A,(SEC)
INC B
INC B
INC B
CALL PUT2NUM
POP DE
POP BC
POP AF
RET

```

```

;*****

```

```

;*          B = COLUME NUMBER          *

```

```

;*          C = LINE NUMBER             *

```

```

;*****

```

```

DAY_MM:      PUSH AF
              PUSH BC
              PUSH DE
              INC B

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้ภายในเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

IN A,(DATE)

DEC B

CALL PUT2NUM

IN A,(MONTH)

INC B

INC B

INC B

CALL PUT2NUM

POP DE

POP BC

POP AF

RET

;*****
;*      non markable interrupt survice subroutine      *
;*      for real time of system                        *
;*****
NMI:      PUSH AF
          PUSH BC
          LD B,9
          LD C,0
          CALL DAY_MM
          LD C,1
          CALL TIME
          IN A,(RTC+10H)      ; enable RTC to
                              ; interrupt

          POP BC

          POP AF

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

;*****

```

      ORG 6000H
KEYTAB:      DFB 0B2H,0D1H,0B1H,071H  ; 10   7   8   9
              DFB 0D0H,0B0H,070H,0D3H  ; 11   4   5   6
              DFB 0B3H,073H,0E3H,0E0H  ; 12   1   2   3
              DFB 0E1H,0E2H,0D2H,072H  ; 13  14   0  15
                                          ; E3 D3 B3 73
                                          ; E0 D0 B0 70
                                          ; E1 D1 B1 71
                                          ; E2 D2 B2 72
PCL:         DFB "   PC REMOTE   ",13
              DFB " Please wait ",0FFH
D_TRANSFER:  DFB " DATA TRANSFER ",13
              DFB "   from PC   ",13
              DFB " -----",13
              DFB " Please wait ",0FFH
EXPCL:       DFB " Data transfer ",13
              DFB " compleate   ",13
              DFB "               ",13
              DFB "   THANK YOU PC ",0FFH
CLEAR:       DFB "               ",0FFH
CLR_CLK:     DFB "               ",0FFH
LINK_ERROR:  DFB "DATA LINK ERROR ",13
              DFB "   SYSTEM HALT  ",0FFH
TITLE:       DFB "   KMITL       ",13
              DFB " TELECOM Lab.   ",13
              DFB " low orbit sat. ",13
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

AUTOPAT: DFB "Auto /95",13
 DFB " ",13
 DFB " Please select ",13
 DFB " Sat.number 0-8 "0FFH
 MANUAL: DFB "Manaul /95",13
 DFB " ",13
 DFB "Press arrow key ",13
 DFB "to control ant. ",0FFH
 SAT_TAB: DFB " /95",13
 DFB " ",13
 DFB " Please wait ",13
 DFB " now find data ",0FFH
 MENU: DFB " MENU ",13
 DFB "1. Automatic ",13
 DFB "2.Manaul control",13
 DFB "3.Time set ",0FFH
 TIME_MENU: DFB "1.DATE 30/03/95",13
 DFB "2.TIME 16:05:15",13
 DFB "-----",13
 DFB "Please select No",0FFH
 SAT_PAT: DFB " ~ Satellite ",13
 DFB "-----",13
 DFB " ",13
 DFB "-----",0FFH
 SATELLITE: DFB "RS-10",0FFH
 DFB "RS-12",0FFH
 DFB "FO-20",0FFH
 DFB "AO-10",0FFH

```

DFB "AO-13",0FFH
DFB "AO-21",0FFH
DFB "AO-27",0FFH
DFB "DO-17",0FFH
DFB " MIR ",0FFH
MOTOR_PAT_AZ:  DFB 00,01H,03H,02H,06H
                DFB 04H,0CH,08H,09H,0FH
MOTOR_PAT_EL:  DFB 00,10H,30H,20H,60H
                DFB 40H,0C0H,80H,90H,0FH
Find_comple_tab: DFB " FIND COMPLETE ",0FFH
Found_not_tab:  DFB "No have data of",13
                DFB "this satellite ",0FFH
LOCK:           DFB "LOCK",0FFH
LOCK_CLR:       DFB " ",0FFH
PRG_ANGLE:      DFB " Angle from PC ",13
                DFB " ",13
AZ_EL:          DFB "-----",13
                DFB "AZ:  ",0DFH,"EL:  ",0DFH,0FFH

```

```

;*****
;*                TIME OF SATELLITE BUFFER                *
;*****

```

```

                ORG 8100H
RS_10:          DFS 2560
RS_12:          DFS 2560
FO_20:          DFS 2560

```

```

AO_10:          DFS 2560

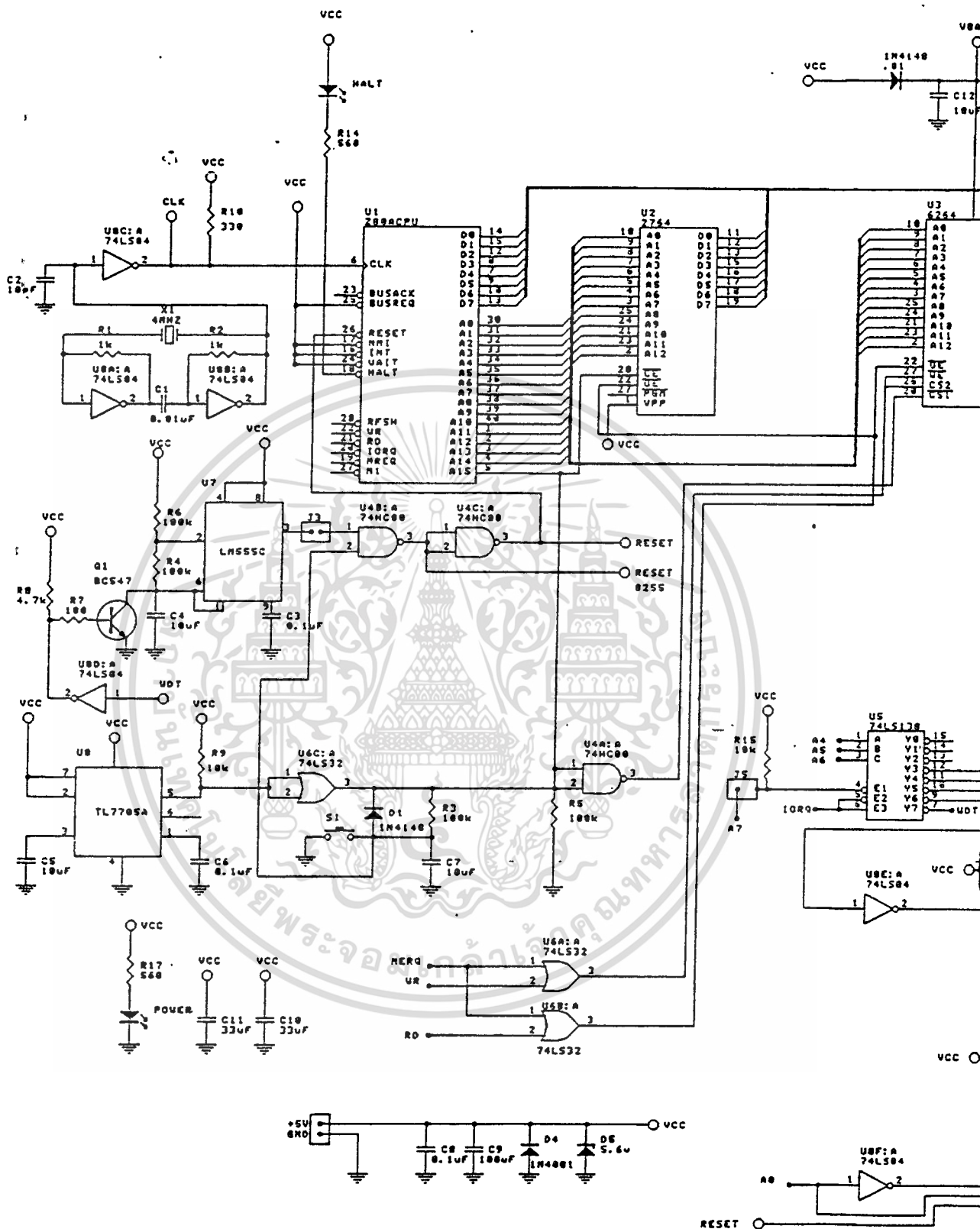
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้เพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่า AO_13: ลีน อีกทั้งห้ามมิให้ดัดแปลง และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

AO_21: DFS 2560
AO_27: DFS 2560
DO_17: DFS 2560
MIR: DFS 2560
SAT_BUF_ADDR: DFS 2
AZ_PAT_ADDR: DFS 2
EL_PAT_ADDR: DFS 2
DUMMY_DATE: DFS 1
DUMMY_HOUR: DFS 1
AZIMUTE: DFS 2
ELEVATION: DFS 2
RESULT_AZ: DFS 2
RESULT_EL: DFS 2
PRG_AZ: DFS 2
PRG_EL: DFS 2
;*****

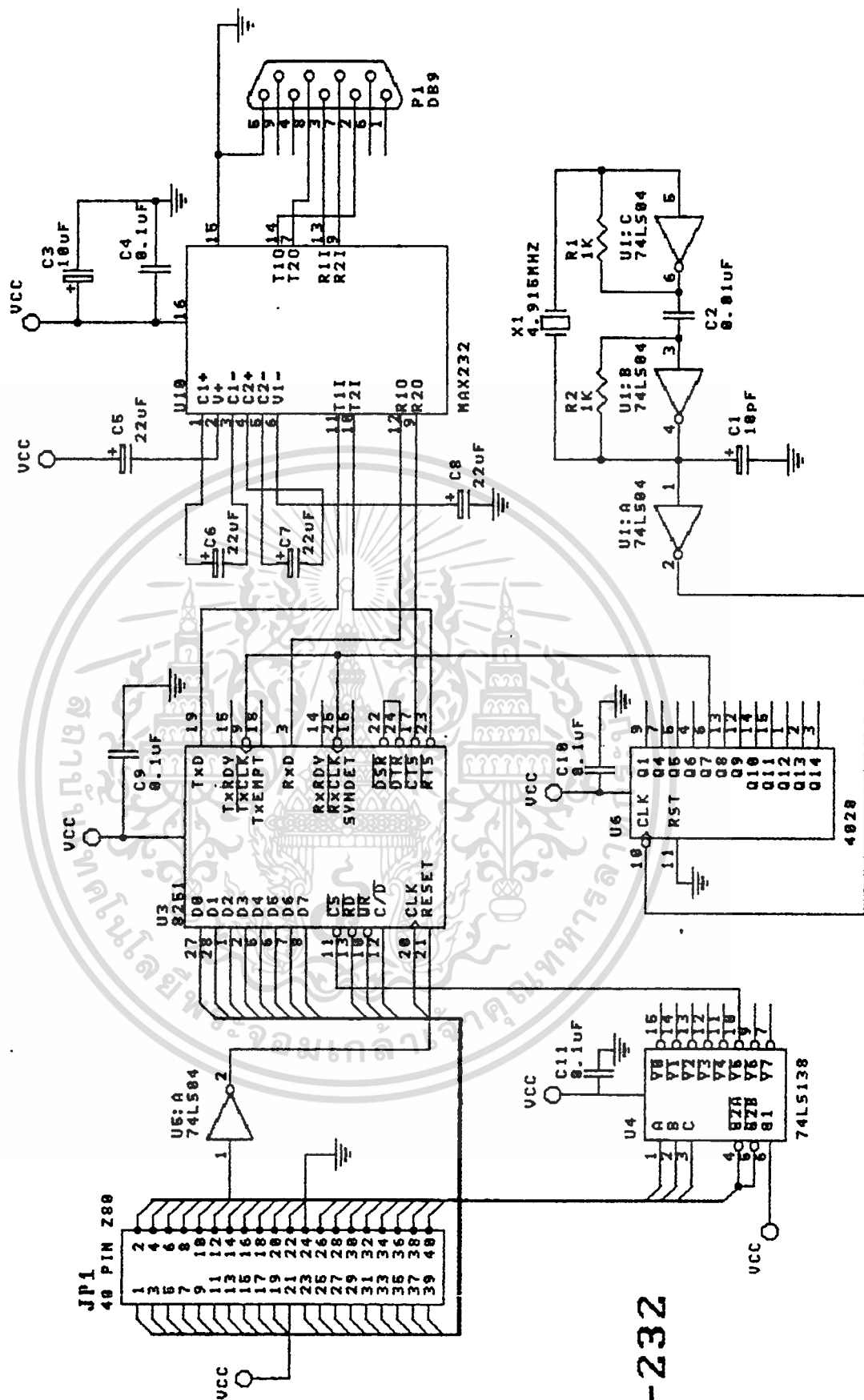


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



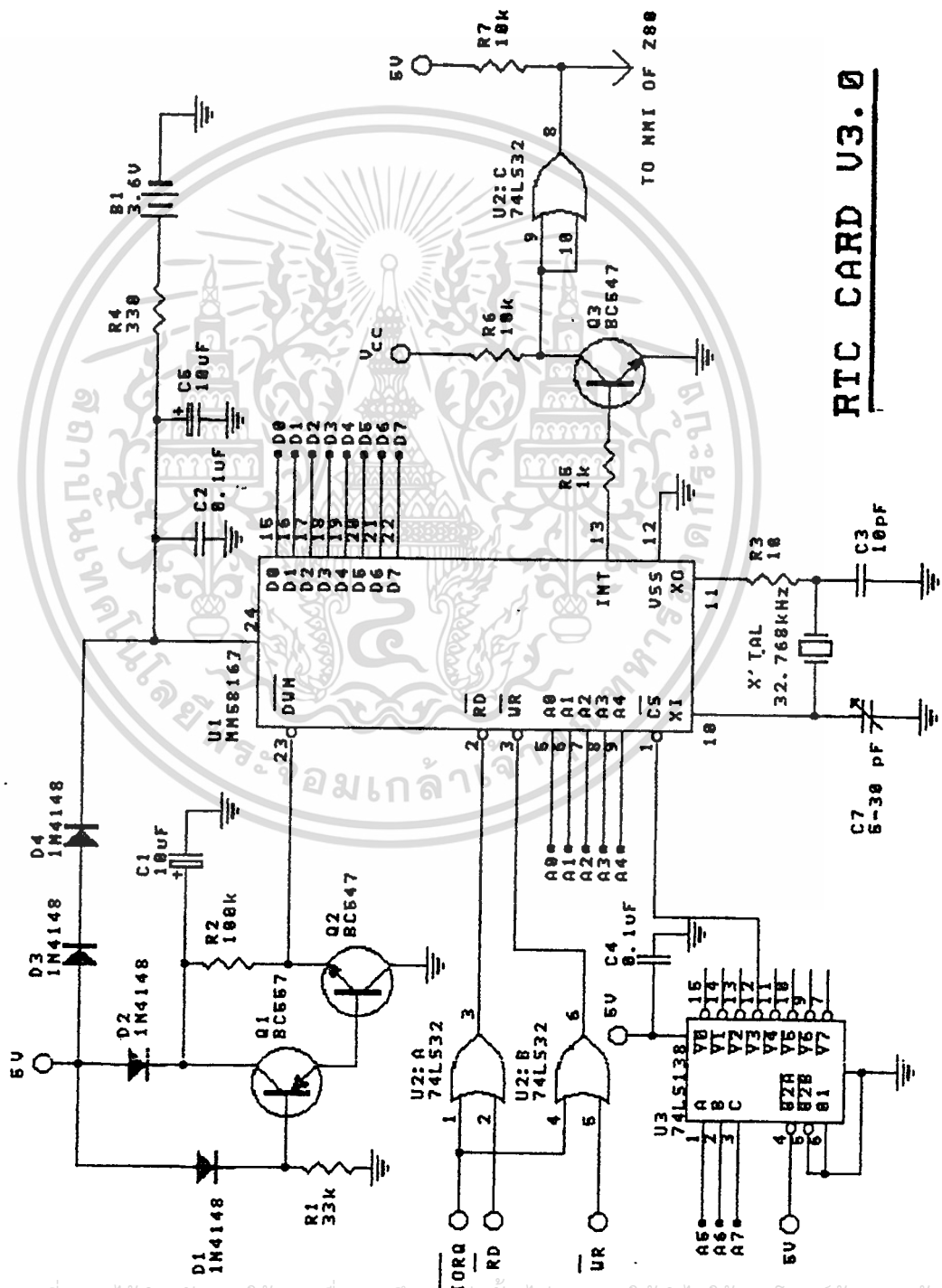
CP-Z80V2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



RS-232

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



RTC CARD V3.0

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

SPECIFICATIONS

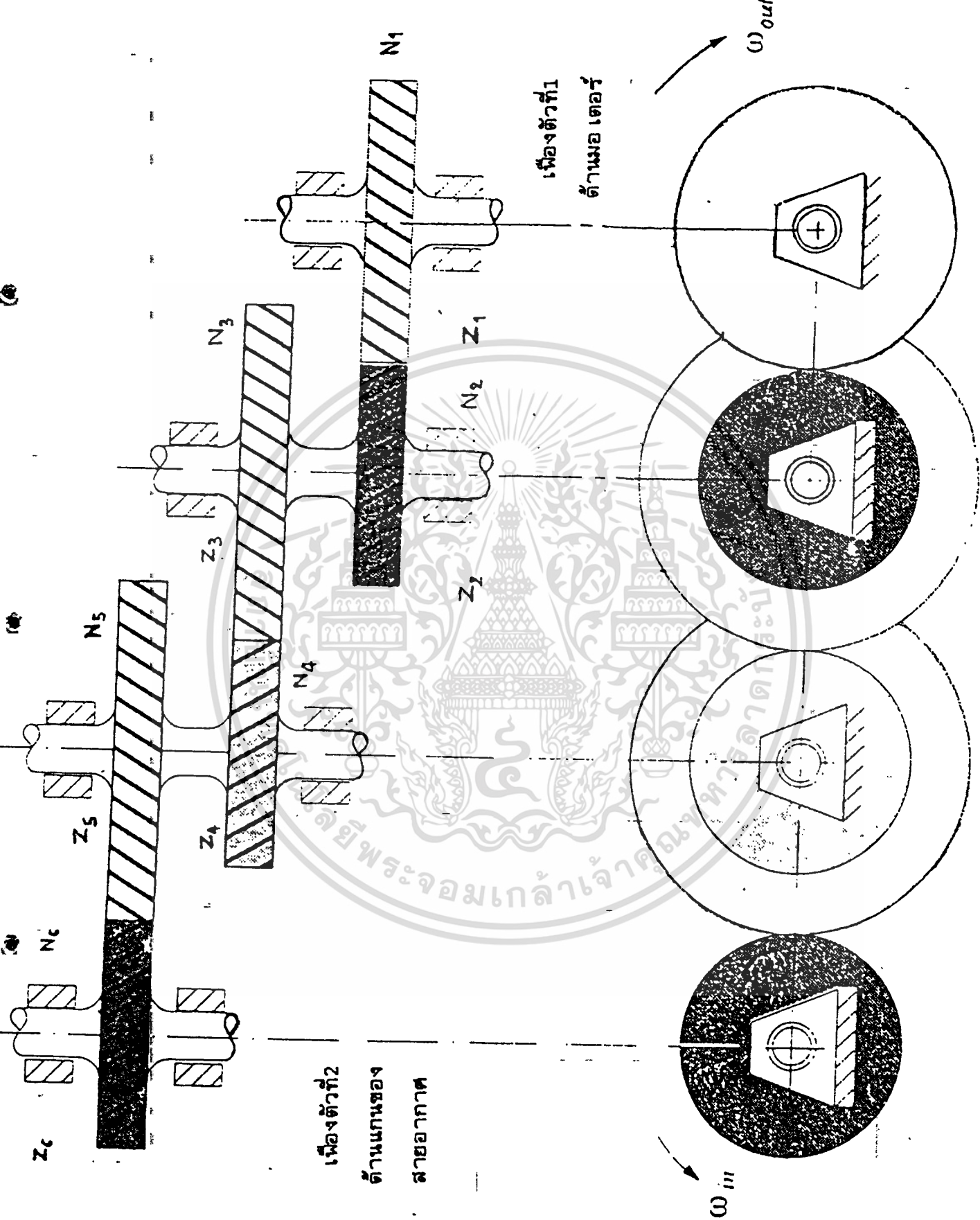
OUTPUT STEPPING MOTOR

CHANNELS : 4 OUTPUT 2 CHANNELS
PHOTO-COUPLED : 4 N 33
CHANNEL STATUS : LED DOT DISPLAY
ISOLATION VOLTAGE : 2500V (PEAK)
OUTPUT DRIVING : 6 A 50 V

INPUT PHOTO-COUPLED INPUT

CHANNELS : 4 CHANNELS
PHOTO-COUPLED : 4 N 33
CHANNEL STATUS : NOT
ISOLATION VOLTAGE : 2500V (PEAK)
HIGH SIGNAL IMMUNITY : 3-12V Photo-Coupling ON
0-1V Photo-Coupling OFF
(R INPUT 1K OHM 1/4W)

CONNECTOR : 1 40 PIN EXPANSION HEADER-STRIP
(8255 BUS , ET-V3.0 , V3.5)
: 2 34 PIN EXPANSION HEADER-STRIP
(7210Z80 BUS)
: 2 6 PIN EXPANSION (STEPPING MOTOR)
: 4 2 PIN EXPANSION (INPUT PHOTO-COUPLED INPUT)
: 1 2 PIN EXPANSION POWER
POWER SUPPLY : CONSUMPTION 5VDC & MAIN INPUT 7VDC-12VDC POWER
SUPPLY CURRENT 250 mA
PCB SIZE : 13 CM X 16 CM



JAS Series

SPACECRAFT NAME: Fuji-OSCAR 20

General

1.1 Identification

International designation: 1990-013C
NASA Catalog number: 20480
Prelaunch designation: JAS-1B
Also known as: FO-20, Fuji-2
Call sign: 8J1JBS

1.2 Launch

Date: 7 February, 1990 (01:33 UTC)
Vehicle: H1
Agency: NASDA (Japanese National Space Agency)
Site: Tanegashima Space Center, Japan

1.3 Orbital Parameters (April 1990)

General designation: low-altitude, elliptical, non-sun-synchronous
Period: 112.23 minutes
Apogee: 1745 km
Perigee: 912 km
Eccentricity: 0.0541
Inclination: 99.05°
Longitude increment: 28.1°W/orbit
Maximum access distance: 4257 km

1.4 Ground Track Data

See Appendix C

1.5 Operations

Coordinating Group: JAMSAT (Japanese AMSAT)

1.6 Design/Construction Credits

JARL (Japanese Amateur Radio League)
NASDA (Japanese National space Agency)
NEC (Nippon Electric Company): space frame, power supply, etc
JAMSAT (Japanese AMSAT): transponders, telemetry/command, IHU, ground-support systems.

1.7 Primary References

Note: Except for its orbit, Fuji-OSCAR 20 is very similar to Fuji-OSCAR 12 (aka JAS-1 and Fuji-1). Although much of the detailed information currently available focuses on FO-12, it is valid for FO-20. All known differences are specified in this section and reference 3.

1) *JAS-1 Handbook*, JARL, August 1985. English translation available from Project OSCAR, PO Box 1136, Los Altos, CA 94023.

2) V. Riportella, "Introducing Japanese Amateur Satellite Number One (JAS-1)," *QST*, Jun 1986, pp 71-72; "Operating the Flying Mailbox: FO-12 Mode JD," *QST*, Nov 1986, pp 66-67.

3) "Introduction of JAS-1b," by JARL, *QEX*, Sep 1989, pp 8-11.

Spacecraft Description

2.1 Physical Structure

Shape: Polyhedron of 26 faces, 25 covered with solar cells.
Overall size: 470 (height) × 440 × 440 mm
Mass: 50 kg
Thermal control: Passive using solar cell cover glass and coatings.
See Fig 1 (FO20)

2.2 Subsystem Organization

See Fig 2 (FO20)

Subsystem Description

3.1 Beacons

Frequency: 435.795 MHz
Modulation: CW (primary) or PSK (secondary)
Power: 60 mW
Maximum Doppler: 10.1 kHz
Frequency: 435.910 MHz (digital transponder downlink)
Modulation: PSK
Power: 1.0 W
Maximum Doppler: 10.1 kHz

3.2 Telemetry

The telemetry system sends two types of information: analog data relating to voltages, currents and temperatures; and status data from two-state (on/off or other) devices. The spacecraft has sensors placed at 29 analog test points and 33 status points. There's a considerable amount of

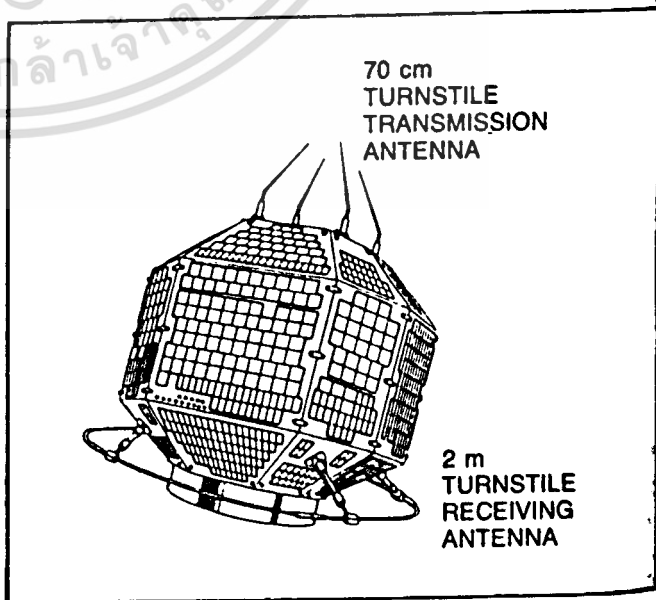


Fig 1 (FO20)—Drawing of Fuji-OSCAR 20 (from OSCAR News, Dec 1989, p 36).

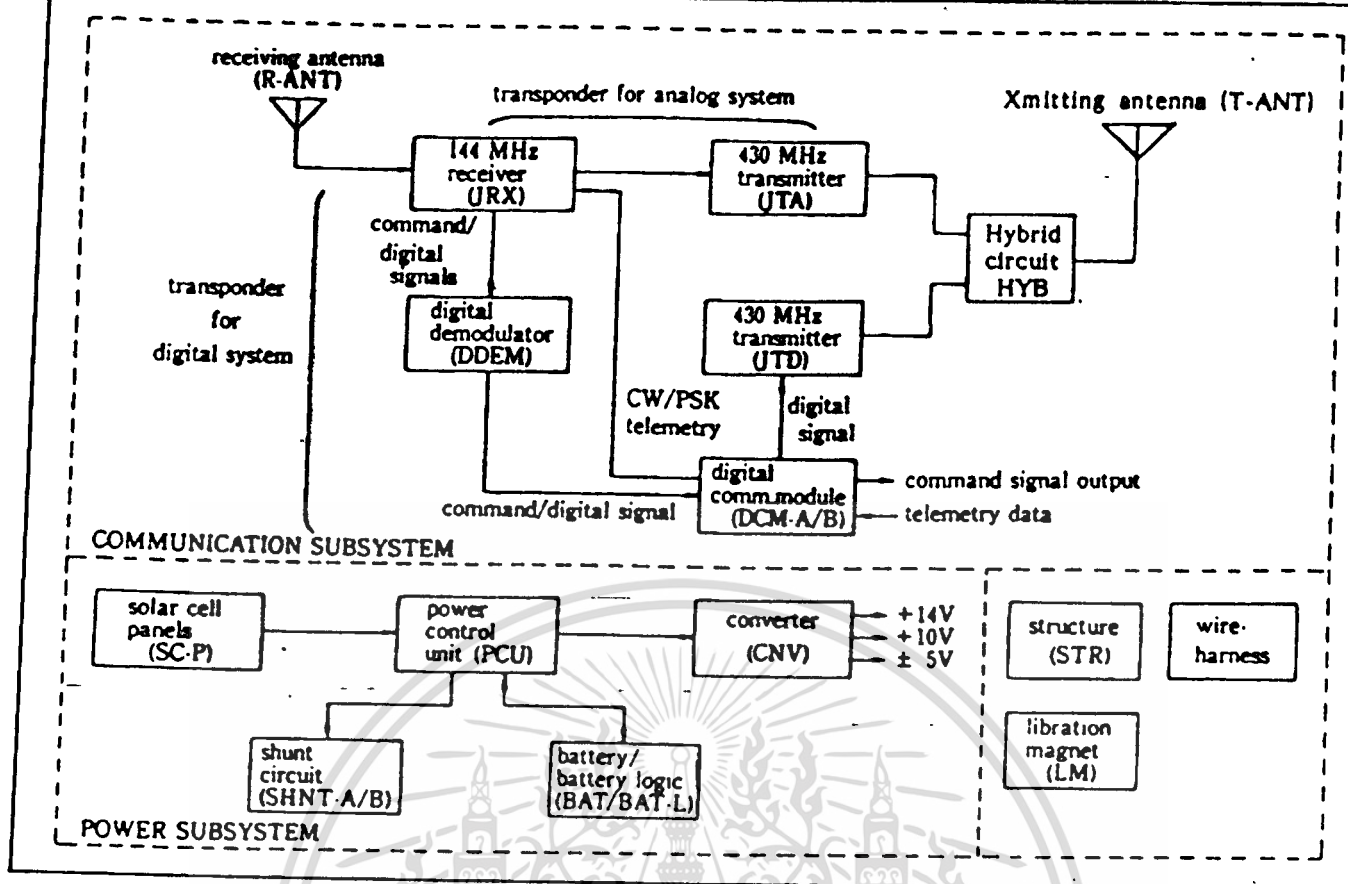


Fig 2 (FO20)—Subsystem block diagram (from OSCAR News, Dec 1989, p 38).

flexibility in how this data is sent, but there are two primary formats: PSK packets and CW.

CW telemetry is sent on the 435.795-MHz beacon at 20 WPM. A frame contains 20 channels, which may be thought of as forming a 5-row by 4-column array. The frame marker is the standard HI. Each channel consists of three digits. The first digit is a row marker. The first 12 channels contain analog data, per Table 1 (FO20). The remaining two digits per channel contain the raw data. The last eight channels contain encoded status data. (See Chapter 15 for additional information on CW telemetry decoding.)

PSK telemetry is generally sent on 435.910 MHz in the form of AX.25 packets. Three types of frames are used: real-time TTY (ASCII or binary), stored TTY (ASCII or binary) and messages (numbered 0 to 9). The real-time and stored ASCII frames contain information on all 29 analog data points and 33 status points. Decoding information is contained in Table 2 (FO20).

3.3 Telecommand System

The telecommand system recognizes five operating modes. Numbers in parentheses refer to operating power required.

Mode D. All loads off except command receiver. (2.5 W)

Mode JA on. (5.3 W)

Mode JD on with 1 to 4 256-k memory banks

activated. (JD-1 [6.2 W], JD-2 [6.7 W], JD-3 [7.0 W], JD-4 [7.3 W])

Mode DI. All loads off except command receiver, CPU and memory.

Mode JAD. Both JA and JD on. (10.2 W)

3.4 Transponders

Transponder 1: Mode JA

Type: linear, inverting

Uplink passband: 145.900-146.000 MHz

Downlink passband: 435.800-435.900 MHz

Translation equation:

Downlink freq [MHz] =

$581.800 - \text{uplink freq. [MHz]} \pm \text{Doppler}$

Power output: 2 W PEP

Recommended uplink EIRP: 100 W

Bandwidth: 100 kHz

Maximum Doppler: 6.7 kHz

Transponder 2: Mode JD

Type: digital

Uplink channels: #1 145.850 MHz

#2 145.870 MHz

#3 145.890 MHz

#4 145.910 MHz

Downlink channel: 435.910 MHz

Power output: 1 W RMS

Recommended uplink EIRP: 100 W

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปเผยแพร่โดยไม่ได้รับอนุญาต

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Table 1 (FO20)

Fuji-OSCAR 20 CW Telemetry

(Reference: JR1NVU, "The Telemetry Formats of JAS-1B/Fuji-OSCAR 20," *The AMSAT Journal*, Sep 1990, pp 20-21.

Channel identification					Channel contents					Decoding Information for status channels (4A-5D con't)				
HI	1A	1B	1C	1C	HI	1nn	1nn	1nn	1nn	CH	BIT	DESCRIPTION	STATE	
	2A	2B	2C	2D		2nn	2nn	2nn	2nn	4C	0	PCU	bit 1	(LSB)
	3A	3B	3C	3D		3nn	3nn	3nn	3nn	4C	1	PCU	bit 2	(LSB)
	4A	4B	4C	4C		4jj	4jj	4jj	4jj	4C	2	PCU	manual	auto
	5A	5B	5C	5D	HI	5jj	5jj	5jj	5jj	4C	3	Eng. data #3	—	—
Decoding equations for analog channels (1A-3D)										4C	4	Eng. data #4	—	—
1A	Total solar panel current				I	= 19(nn + 0.4) mA				4D	0	Memory bank #0	ON	OFF
1B	Batter charge/discharge				I	= -38(nn - 50) mA				4D	1	Memory bank #1	ON	OFF
1C	Battery terminal voltage				V	= 0.22(nn + 4) V				4D	2	Memory bank #2	ON	OFF
1D	Battery center tap				V	= 0.1(nn + 4) V				4D	3	Memory bank #3	ON	OFF
2A	Bus voltage				V	= 0.20(nn + 4) V				4D	4	Computer power	ON	OFF
2B	+5 V regulator output				V	= 0.062(nn + 4) V				5A	0	Memory select	bit 1	(LSB)
2C	Mode JA power output				P	= 2.0(nn + 4) ^{1.618} mW				5A	1	Memory select	bit 2	(MSB)
2D	Calibration voltage #1				V	= (nn + 4)/50 V				5A	2	Eng. data #5	—	—
3A	Battery temperature				T	= 1.4(67 - nn) °C				5A	3	Eng. data #6	—	—
3B	Structure temperature #1				T	= 1.4(67 - nn) °C				5A	4	Eng. data #7	—	—
3C	Structure temperature #2				T	= 1.4(67 - nn) °C				5B	0	Solar panel #1	lit.	dark
3D	Structure temperature #3				T	= 1.4(67 - nn) °C				5B	1	Solar panel #2	lit	dark
Decoding Information for status channels (4A-5D)										5B	2	Solar panel #3	lit	dark
CH	BIT	DESCRIPTION		STATE						5B	3	Solar panel #4	lit	dark
				1	0					5B	4	Solar panel #5	lit	dark
4A	0	JTA Power		ON	OFF					5C	0	JTA CW beacon	CPU	TLM
4A	1	JTD Power		ON	OFF					5C	1	Eng. data #8	—	—
4A	2	Eng. data #1		—	—					5C	2	Eng. data #9	—	—
4A	3	Eng. data #3		—	—					5C	3	Eng. data #10	—	—
4A	4	Beacon		PSK	CW					5C	4	Eng. data #11	—	—
4B	0	UVC		ON	OFF					5D	0	Eng. data #12	—	—
4B	1	UVC level		1	2					5D	1	Eng. data #13	—	—
4B	2	Battery		tric	full					5D	2	Eng. data #14	—	—
4B	3	Battery logic		tric	full					5D	3	Eng. data #15	—	—
4B	4	Main relay		ON	OFF					5D	4	Eng. data #16	—	—

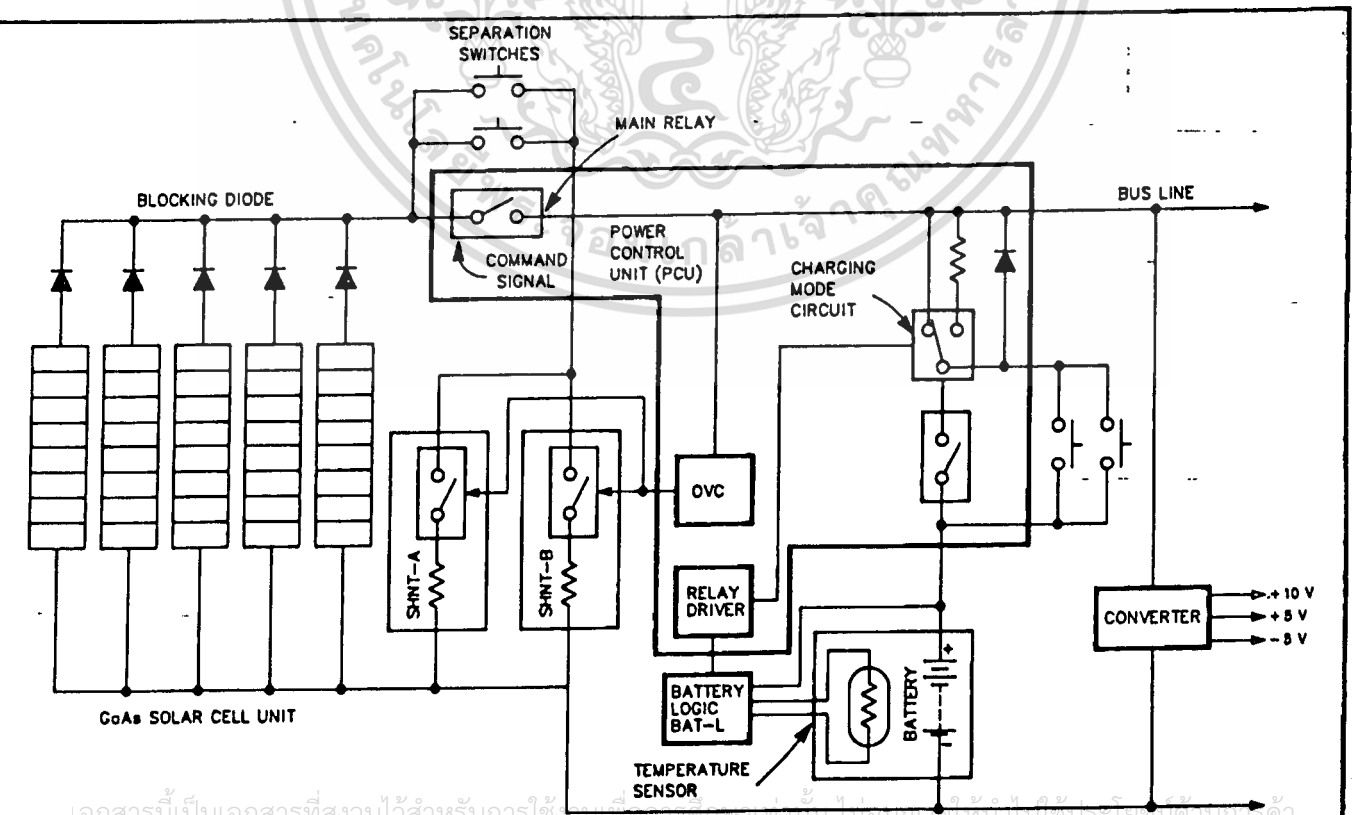


Fig 3 (FO20)—Fuji-OSCAR 20 power subsystem (from OSCAR News, June 1989, p 11).

Table 2 (FO20)
Fuji-OSCAR 20 PSK Telemetry

Channel Identification

... Header ...
00 01 02 03 04 05 06 07 08 09
10 11 12 13 14 15 16 17 18 19
20 21 22 23 24 25 26 27 28 29
30 31 32 33 34 35 36 37 38 39

Sample TTY Frame (as received from FO-20)

8J1JBS>BEACON:
JAS1b RA 90/03/08 11:02:00
596 375 692 698 750 837 849 831 001 686
618 001 507 510 532 527 530 532 655 001
662 654 666 677 999 647 879 960 199 000
010 111 000 000 111 100 001 110 111 000

Frame contents

JAS1b FF YY/MM/DD HH:MM:SS
xxx xxx xxx, xxx xxx xxx xxx xxx xxx
xxx xxx xxx, xxx xxx xxx xxx xxx xxx
xxx xxx xxx, xxx xxx xxx xxx hhh hhh hhh
||| ||| ||| ||| ||| ||| ||| ||| |||

FF = Frame Identifier
RA: Real-time TTY (ASCII)
RB: Real-time TTY (binary)
SA: Stored TTY (ASCII)
SB: Stored TTY (binary)
Mn: Message (n = 0 to 9)

YY/MM/DD HH:MM:SS = Date & Time (UTC)

Following valid only for RA and SA frames

xxx = 3-digit decimal number occurring in channels 00 to 26.
Represented by N in calibration equations.
hhh = series of three individual hex numbers
||| = series of three individual binary numbers

Decoding equations for analog channels (00-26)

CH	DESCRIPTION	CALIBRATION
#00	total solar array current	$1.91 \times (N - 4) \text{mA}$
#01	battery charge/discharge	$-3.81 \times (N - 508) \text{mA}$
#02	battery voltage	$N \times 0.022 \text{ V}$
#03	battery center voltage	$N \times 0.009961 \text{V}$
#04	bus voltage	$N \times 0.02021 \text{V}$
#05	+5 V regulator voltage	$N \times 0.00620 \text{ V}$
#06	-5 V regulator voltage	$-N \times 0.00620 \text{ V}$
#07	+10 V regulator voltage	$N \times 0.0126 \text{ V}$
#08	JTA output power	$5.1 \times (N - 158) \text{mW}$
#09	JTD output power	$5.4 \times (N - 116) \text{mW}$
#10	calibration voltage #2	N/500 V
#11	offset voltage #1	N/500 V
#12	battery temperature	$0.139 \times (669 - N) \text{deg. C}$
#13	JTD temperature	$0.139 \times (669 - N) \text{deg. C}$
#14	Temperature #1	$0.139 \times (669 - N) \text{deg. C}$
#15	Baseplate Temperature #2	$0.139 \times (669 - N) \text{deg. C}$
#16	Baseplate Temperature #3	$0.139 \times (669 - N) \text{deg. C}$

Decoding equations for analog channels (00-26, con't)

CH	DESCRIPTION	CALIBRATION
#17	Baseplate Temperature #4	$0.139 \times (669 - N) \text{deg. C}$
#18	temperature calibration #1	N/500 V
#19	offset voltage #2	N/500 V
#20	Solar Cell Panel Temp #1	$0.38 \times (N - 685) \text{deg. C}$
#21	Solar Cell Panel Temp #2	$0.38 \times (N - 643)$
#22	Solar Cell Panel Temp #3	$0.38 \times (N - 646)$
#23	Solar Cell Panel Temp #4	$0.38 \times (N - 647)$
#24		
#25	temperature calibration #2	N/500 V
#26	temperature calibration #3	N/500 V

Decoding information for hex status bytes (27-29)

CH	DESCRIPTION
#27a	Spare (TBD)
#27b	Spare (TBD)
#27c	Spare (TBD)
#28a	Spare (TBD)
#28b	Spare (TBD)
#28c	error count of memory unit #0
#29a	error count of memory unit #1
#29b	error count of memory unit #2
#29c	error count of memory unit #3

Decoding information for binary status bytes (30-39)

CH	DESCRIPTION	STATE
#30a	JTA power	on off
#30b	JTD power	on off
#30c	JTA beacon	PSK CW
#31a	UVC status	on off
#31b	UVC level	1 2
#31c	main relay	on off
#32a	eng. data #1	—
#32b	battery status	tric full
#32c	battery logic	tric full
#33a	eng. data #2	—
#33b	PCU status	bit 1 (LSB)
#33c	PCU status	bit 2 (MSB)
#34a	memory unit #0	on off
#34b	memory unit #1	on off
#34c	memory unit #2	on off
#35a	memory unit	on off
#35b	memory select	bit 1 (LSB)
#35c	memory select	bit 2 (MSB)
#36a	eng. data #3	—
#36b	eng. data #4	—
#36c	computer power	on off
#37a	eng. data #5	—
#37b	solar panel #1	lit dark
#37c	solar panel #2	lit dark
#38a	solar panel #3	lit dark
#38b	solar panel #4	lit dark
#38c	solar panel #5	lit dark
#39a	eng. data #6	—
#39b	CW beacon source	CPU TLM
#39c	eng. data #7	—

Comments: Uplink must be Manchester-coded FM. The protocol is AX.25, Level 2, Version 2. Data rate is 1200 bauds. The downlink is 1200-baud BPSK. For information on modems by TAPR and G3RUH, see Chapters 9 (hardware), 10 (operating notes) and 15 (additional information).

Note: When the power budget permits, the JA and JD transponders may be operated simultaneously. When this occurs, some spurious signals may be observed in the JA passband, especially when stations employing too much power are accessing the mode JA transponder.

3.5 Attitude Stabilization

Two 1 TAm² permanent magnets mounted parallel to spacecraft Z axis align along earth's local magnetic field.

3.6 Antennas

2-meter receive: Turnstile mounted on bottom of spacecraft. (This antenna replaces the 2-meter ¼-wavelength monopole mounted on top of the Fuji-1 spacecraft.)

70-cm transmit (Mode JA and JD): Canted turnstile on top of spacecraft. A circulator and phase shifter added to Fuji-2 enables a single antenna to be used for both links.

3.7 Energy Supply and Power Conditioning

Power Subsystem: See Fig 3 (FO20). Battery

control logic monitors voltage, current and temperature of storage battery. Software will attempt to utilize spacecraft systems to minimize activation of overvoltage control (OVC) and under-voltage control (UVC) functions.

Solar cells: Gallium arsenide. Spacecraft contains approximately 1530 cells, 900 measuring 1 × 2 cm and 630 measuring 2 × 2 cm. The total cell surface area is about 4320 cm². When first launched, these cells generated about 13 W (optimum satellite orientation) and about 11 W average when spacecraft is in sunlight. Compared to Fuji-1, these figures represent an increase of 9% in cell area and 70% in available power.

Bus voltage: +11 to 18 (14 V average)

Battery: 11 NiCd cells providing about 14 V at 6 Ah.

Instrument Switching Regulators are used to provide +10 V, +5 V and -5 V for spacecraft systems. Efficiency >70%.

3.9 IHU

CPU: NSC-800 (Controls spacecraft and digital transponder)

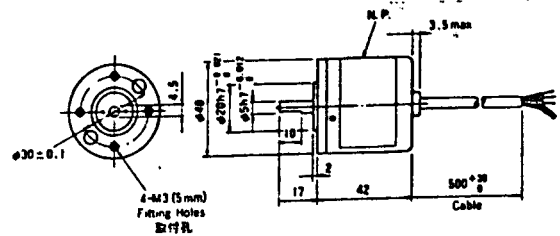
RAM: 48 256-kbit NMOS DRAMs. These are configured using a hardware-based error-detection-and-correction scheme to provide 1 Mbyte of EDAC 8-bit memory. The operating system resides in approximately 32 k; the rest is used for message storage.

Product Number 12□Series

品番 12□シリーズ

◇External Dimensions (mm)

外形寸法



Electrical Connections

Color of Lead Wire	Description
Red	Power source
Black	OV common
Green or Blue	Signal A
White	Signal B
Yellow	Signal M
Shielding Braid	No inside connection

結線

色	接続
赤	電源
黒	OV コモン
緑又は青	信号 A
白	信号 B
黄	信号 M
シールド線	内部無接続

Types 型式	Power Source L 電源		H
	Signal A, B, M (信号)		
	R2 † #121	C #122	C #123
100 P/R	OES-01-2M	OES-01-2MC	OES-01-2MHC
200 //	OES-02-2M	OES-02-2MC	OES-02-2MHC
250 //	OES-025-2M	OES-025-2MC	OES-025-2MHC
256 //	OES-0256-2M	OES-0256-2MC	OES-0256-2MHC
300 //	OES-03-2M	OES-03-2MC	OES-03-2MHC
360 //	OES-036-2M	OES-036-2MC	OES-036-2MHC
400 //	OES-04-2M	OES-04-2MC	OES-04-2MHC
500 //	OES-05-2M	OES-05-2MC	OES-05-2MHC
512 //	OES-0512-2M	OES-0512-2MC	OES-0512-2MHC
600 //	OES-06-2M	OES-06-2MC	OES-06-2MHC
1000 //	OES-10-2M	OES-10-2MC	OES-10-2MHC

L and R are not indicated on type names. L と R は無表示

Specifications 仕様

◇Electrical

電気仕様

Power Source 使用電源	1) L: +4.5V -- +13.2V DC 2) H: +10.8V -- +26.4V DC		
Output Voltage 出力電圧	'1'	Power Source - 1V, with no load 電源電圧 - 1V, 無負荷)	
	'0'	0.5V or less at 20mA sinking (0.5V以下, 20mA引き時)	
Signal Mode 出力信号	A,B (90° phase difference each other), and M signals (90°位相差2信号とM信号)		
Output Mode 出力形態	R2 = With pull-up resistor and a diode (プルアップ抵抗付き) C = Open collector (オープンコレクタ)		
Response Frequency 応答周波数	50kHz, (100-600 P/R) 60kHz, (1,000 P/R)		
Impedance インピーダンス	2kΩ (R2 type)		
Current Consumption 消費電流		100-600 P/R	1,000 P/R
	R2 Type	60mA	80mA
	C Type	40mA	70mA
Signal Rising & Falling Time 信号立上り・立下り時間	1μs or less, for each (各々1μs以下)		
Withstanding Voltage of Output Tr. 出力回路耐圧	-0.4V ~ +50V		

◇Mechanical

機械仕様

Starting Torque 起動トルク	10g · cm or less
Shaft Loading 軸荷重	Radial (ラジアル): 1kg Thrust (スラスト): 0.5kg
Angular Acceleration 回転角加速度	1 × 10 ⁴ rad/sec ²
Inertia of Shaft 軸イナーシャ	3g · cm ²
Bearing Life ベアリング寿命	(1 × 10 ⁶ /rpm) hrs.
Max. Slewing Speed 最大回転速度	6,000rpm max.
Net Weight 正重量	100g max.

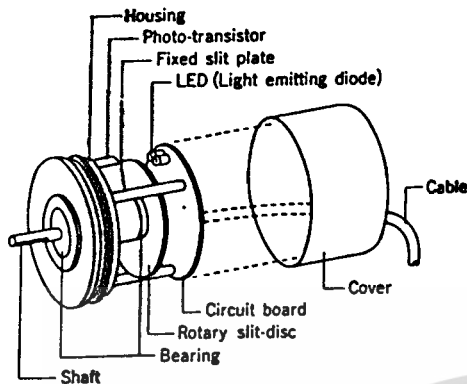
◇Environmental

環境条件

Operating Temperature 動作温度	-10° ~ +70°C
Storage Temperature 保存温度	-30° ~ +80°C
Humidity 湿度	35-85% RH No condensation (結露不可)
Vibration 耐振動	50Hz · 10G · 1hr
Shock 耐衝撃	30G 11ms

General Specifications

Basic Construction (Incremental Encoder)

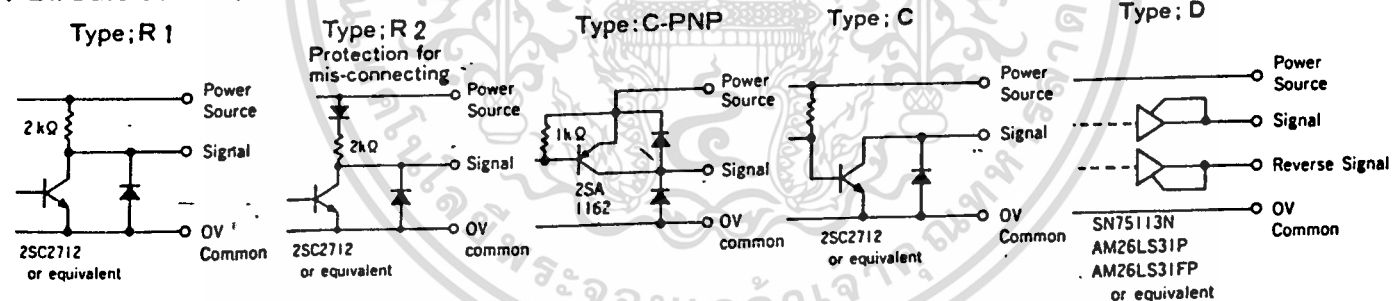


A slit disc is fixed to a rotary shaft held by a pair of bearings. As the disc rotates in accordance with the rotation angle of the shaft, the light from an LED to a photo transistor via a fixed slitplate is intermittently interrupted by the slits on the disc. Then, the phototransistor develops a sine wave signal. This signal whose peaks equal the number of slits on the disc is then shaped to a pulse signal by a circuit on the circuit board.

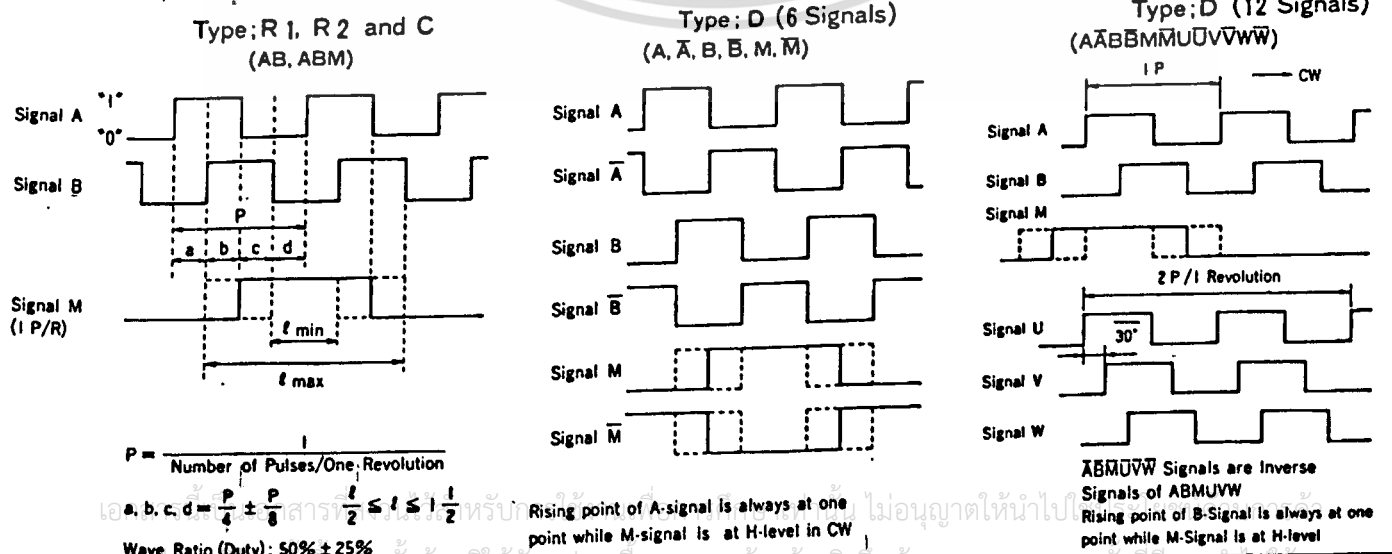
Type Designation RE;OES-05-2MHC

Type	Number of Pulses	Signals	Power Source	Output Mode
OSS	01 : 100 P/R	1: Signal A	L: 4.5-13.2V DC	R1: With Z pull-up resistor
OES	036 : 360 P/R	2: Signals A & B	H: 10.8-26.4V DC	R2: With a pull-up resistor and a diode.
OEL	05 : 500 P/R	2M: Signals A, B, & Marking signal (index)	L is not indicated on type names.	C: Open-collector
NOE	10 : 1000 P/R			D: Line-driver
	1024 : 1024 P/R			R is not indicated on type names.
	20 : 2000 P/R			
	(OMS, OME, DEK, ABSOLUTE ENCODER ARE NOT APPLIED)			

Circuit of Output Signal at Terminals



Output Signals

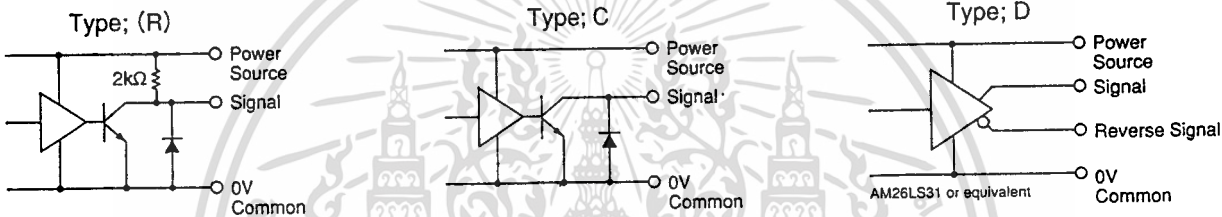


ROTARY ENCODER INSTRUCTION MANUAL

1. Model Name Ex. OEW-10-2MHC

Model	Number of Resolution	Signals	Power Source	Output Mode
OEZ	01 ; 100 P/R	1; Signals A	(L)* 4.5 ~ 13.2V DC	(R) Volatage Output With a pull-up resistor (R) is not indicated.
OEW	036 ; 360 P/R	2; Signals A & B	(L)* is not indicated.	
NOE	05 ; 500 P/R	2M; Signals A, B; & Marking signal (index)	H 10.8 ~ 26.4V DC	
OPN	10 ; 1000 P/R			C Open-collector Output
	1024 ; 1024 P/R			D Line-driver Output
	20 ; 2000 P/R			
	25 ; 2500 P/R			

2. Circuit of Output Signal



3. Precautions for Installation

- 1) Do not give the encoder an excessive shock or vibration because some of the encoders are equipped with a glass disc.
- 2) The shaft of the encoder is hold with a pair of bearings. An excessive load to the shaft may deteriorate the life and/or accuracy of the encoder. Avoid an excessive load to the shaft.
- 3) The power supply to be used should correspond in current capacity. Unsatisfactory power supply may result in an abnormal output waveform.
- 4) For encoder output connection, use a shielded cable. To avoid AC interface pickup, the cable should be run apart from heavy duty AC carrying lines. The maximum length of the cable is 10m for (R)-type output, and 50m for C-type output, but (D)-type output is preferable if there is interference of noise. As some optical components are used in the encoder, the influence of dust, oil, water, etc. should be minimized when installing.

<Handling>

● Connect with a flexible coupling such as a bellows coupling between the encoder and the revolutionary shaft.

<Power Source>

※ Don't bind tightly slipping down their center of the shafts over 0.1mm like both side of drawings above.

4. Electrical Connections

(R) : C Type		D-Type	
Color of Lead Wire	Descriptions	Color of Lead Wire	Descriptions
Red	Power Source	Red	Power Source
Black	0V common	Black	0V common
Blue or Green	Signal A	Green	Signal A
White	Signal B	Blue	Signal A
*Yellow	Signal Z	White	Signal B
		Gray	Signal B
		Yellow	Signal Z
		Orange	Signal Z

TYPE: OEZ; OSS; OEW; FES; OEL; and OPP.

※ For connection of OPN; OEK; NE; SBS and SBG type, please refer to the catalog.

NEMICON CORPORATION

Address: 1-11-15 Higashi Gotanda, Shinagawa-ku, Tokyo, Japan 146

Tel: 03-3442-0346

Telefax: 03-3442-2159

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้เพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า



MM58167B

Microprocessor Real Time Clock

General Description

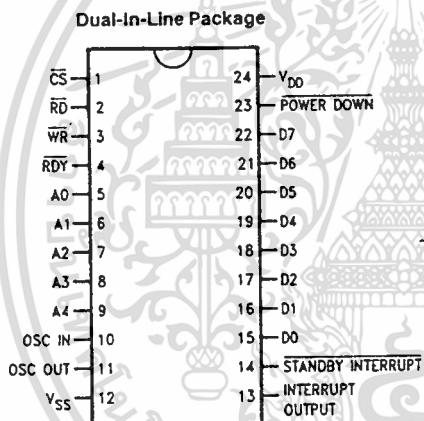
The MM58167B is a low threshold metal gate CMOS circuit that functions as a real time clock in bus oriented microprocessor systems. The device includes an addressable real time counter, 56 bits of RAM, and two interrupt outputs. A **POWER DOWN** input allows the chip to be disabled from the rest of the system for standby low power operation. The time base is a 32.768 Hz crystal oscillator.

Features

- Microprocessor compatible (8-bit data bus)
- Milliseconds through month counters

- 56 bits of RAM with comparator to compare the real time counter to the RAM data
- 2 INTERRUPT OUTPUTS with 8 possible interrupt signals
- **POWER DOWN** input that disables all inputs and outputs except for one of the interrupts
- Status bit to indicate rollover during a read
- 32.768 Hz crystal oscillator
- Four-year calendar (no leap year)
- 24-hour clock

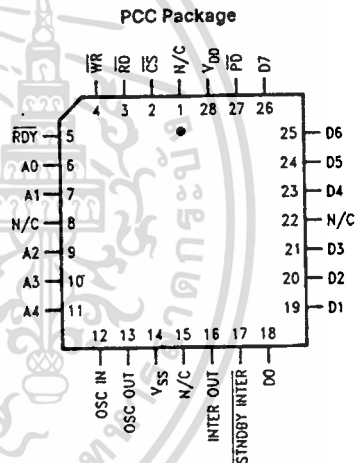
Connection Diagrams



Top View

Order Number MM58167BN
See NS Package Number N24A

TL/F/6148-1



Top View

Order Number MM58167BV
See NS Package Number V28A

TL/F/6148-2

Absolute Maximum Ratings

If Military/Aerospace specified devices are required, please contact the National Semiconductor Sales Office/Distributors for availability and specifications.

Voltage at All Pins $V_{SS} - 0.3V$ to $V_{DD} + 0.3V$
Operating Temperature $0^{\circ}C$ to $70^{\circ}C$

Storage Temperature

$-65^{\circ}C$ to $+150^{\circ}C$

$V_{DD} - V_{SS}$

6.0V

Lead Temperature (Soldering, 10 sec.)

300°C

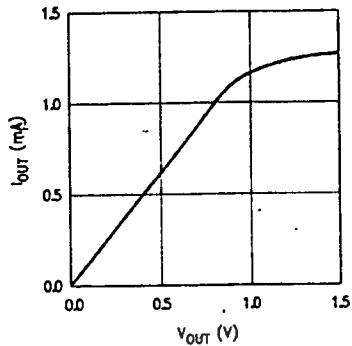
ESD rating is to be determined.

Electrical Characteristics $V_{SS} = 0V, 0^{\circ}C \leq T_A \leq 70^{\circ}C$

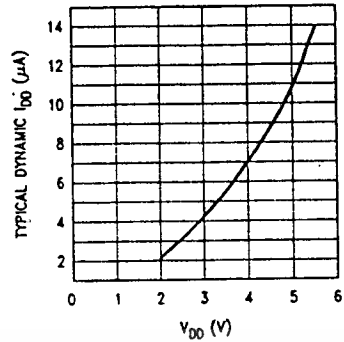
Parameter	Conditions	Min	Max	Units
Supply Voltage V_{DD}	Outputs Enabled	4.5	5.5	V
V_{DD}	POWER DOWN Mode	2.2	5.5	V
Supply Current I_{DD} , Dynamic	Outputs TRI-STATE® $f_{IN} = 32.768$ kHz, $V_{DD} = 5.5V$ $V_{IH} \geq V_{DD} - 0.3V$ $V_{IL} \leq V_{SS} + 0.3V$		20	μA
I_{DD} , Dynamic	Outputs TRI-STATE $f_{IN} = 32.768$ kHz, $V_{DD} = 5.5V$ $V_{IH} = 2.0V, V_{IL} = 0.8V$		5	mA
Input Voltage Logical Low		0.0	0.8	V
Logical high		2.0	V_{DD}	V
Input Leakage Current	$V_{SS} \leq V_{IN} \leq V_{DD}$	-1	1	μA
Output Impedance Logical Low	I/O and INTERRUPT OUT $V_{DD} = 4.5V, I_{OL} = 1.6$ mA		0.4	V
Logical High	$V_{DD} = 4.5V, I_{OH} = -400$ μA $I_{OH} = -10$ μA	2.4 0.8 V_{DD}		V
TRI-STATE	$V_{SS} \leq V_{OUT} \leq V_{DD}$	-1	1	μA
Output Impedance	RDY and STANDBY INTERRUPT (Open Drain Devices)			
Logical Low, Sink	$V_{DD} = 4.5V, I_{OL} = 1.6$ mA		0.4	V
Logical High, Leakage	$V_{OUT} \leq V_{DD}$		10	μA

Functional Description (Continued)

Standby Interrupt
Typical Characteristics



Typical Supply Current
vs Supply Voltage
during Power Down



TL/F/6148-4

TL/F/6148-5

FIGURE 2

Interrupt Timing $0^{\circ}\text{C} \leq T_A \leq 70^{\circ}\text{C}$, $4.5\text{V} \leq V_{DD} \leq 5.5\text{V}$, $V_{SS} = 0\text{V}$

Symbol	Parameter	Min	Max	Units
t _{INTON}	Status Register Clock to INTERRUPT OUTPUT (Pin 13) High (Note 1)		5	μs
t _{SBYON}	Compare Valid to STANDBY INTERRUPT (Pin 14) Low (Note 1)		5	μs
t _{INTOFF}	Trailing Edge of Status Register Read to INTERRUPT OUTPUT Low		5	μs
t _{SBYOFF}	Trailing Edge of Write Cycle (D0 = 0; Address = 16H) to STANDBY INTERRUPT Off (High Impedance State)		5	μs

Note 1: The status register clocks are: the corresponding counter's rollover to its reset state or the compare becoming valid. The compare becomes valid 61 μs after the 1/10,000 of a second counter is clocked, if the real time counter data matches the RAM data.

Read Cycle Timing $0^{\circ}\text{C} \leq T_A \leq 70^{\circ}\text{C}$, $4.5\text{V} \leq V_{DD} \leq 5.5\text{V}$, $V_{SS} = 0\text{V}$

Symbol	Parameter	Min	Max	Units
t _{AR}	Address Bus Valid to Read Strobe (Note 3)	100		ns
t _{CSR}	Chip Select to Read Strobe (Note 2)	0		ns
t _{RRY}	Read Strobe to Ready Strobe		150	ns
t _{RYD}	Ready Strobe to Data Valid		800	ns
t _{AD}	Address Bus Valid to Data Valid		1050	ns
t _{RH}	Data Hold Time from Trailing Edge of Read Strobe	0		ns
t _{HZ}	Trailing Edge of Read Strobe to TRI-STATE Mode		250	ns
t _{RYH}	Read Hold Time after Ready Strobe	0		ns
t _{RA}	Address Bus Hold Time from Trailing Edge of Read Strobe	50		ns
t _{RYDV}	Rising Edge of Ready to Data Valid		100	ns

Note 2: When reading, a deselect time of 500 ns minimum must occur between counter reads. Deselect is: $\overline{\text{CS}} = 1$ or $(\overline{\text{WR}}) \cdot (\overline{\text{RD}}) = 1$.

Note 3: If t_{AR} = 0 and Chip Select, Address Valid or Read are coincident then they must exist for 1050 ns.

Write Cycle Timing $0^{\circ}\text{C} \leq T_A \leq 70^{\circ}\text{C}$, $4.5\text{V} \leq V_{DD} \leq 5.5\text{V}$, $V_{SS} = 0\text{V}$

Symbol	Parameter	Min	Max	Units
t_{AW}	Address Valid to Write Strobe	100		ns
t_{CSW}	Chip Select to Write Strobe	0		ns
t_{DW}	Data Valid before Write Strobe	100		ns
t_{WRY}	Write Strobe to Ready Strobe		150	ns
t_{RY}	Ready Strobe Width		800	ns
t_{RYH}	Write Hold Time after Ready Strobe	0		ns
t_{WD}	Data Hold Time after Write Strobe	110		ns
t_{WA}	Address Hold Time after Write Strobe	50		ns

Note 4: If data changes while CS and WR are low, then they must remain coincident for 1050 ns after the data change to ensure a valid write. Data bus loading is 100 pF. Ready output loading is 50 pF and 3 k Ω pull-up.

Input and output AC timing levels:

Logical one = 2.0V

Logical zero = 0.8V

Read and Write Cycle Timing Diagrams

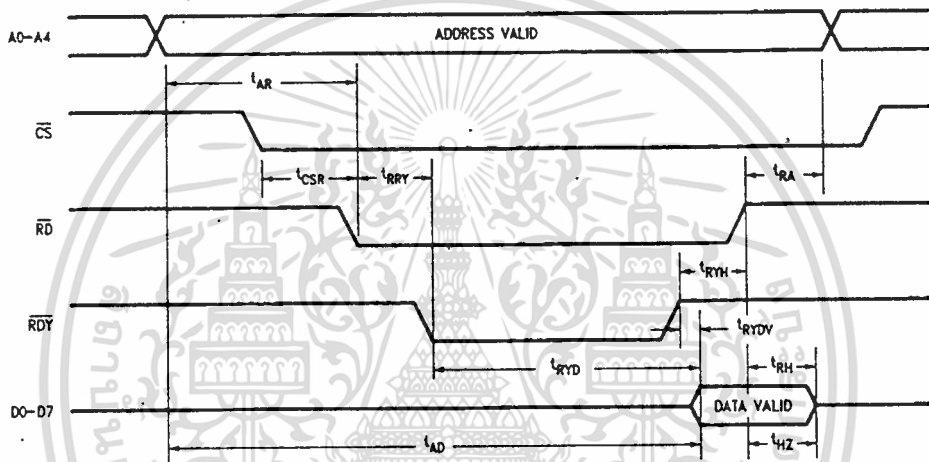


FIGURE 3. Read Cycle Timing

TL/F/614B-6

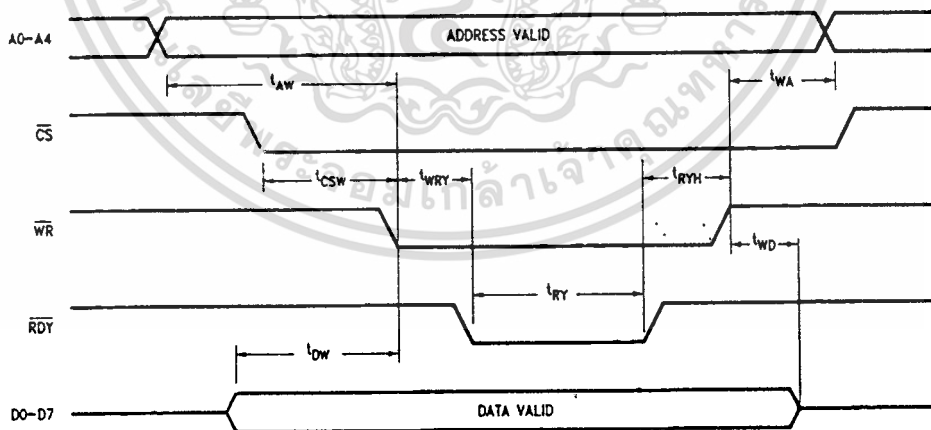
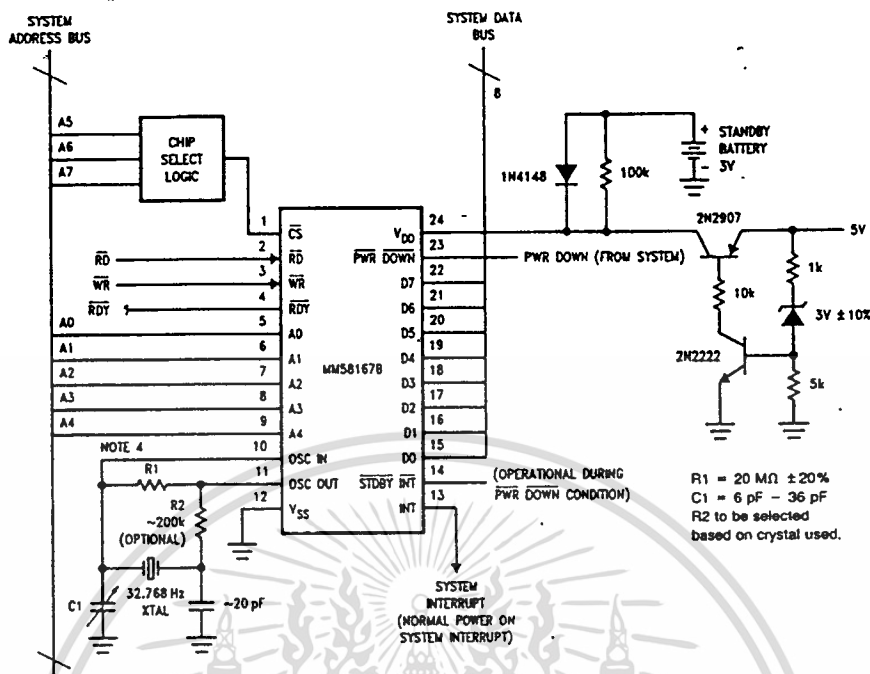


FIGURE 4. Write Cycle Timing

TL/F/614B-7

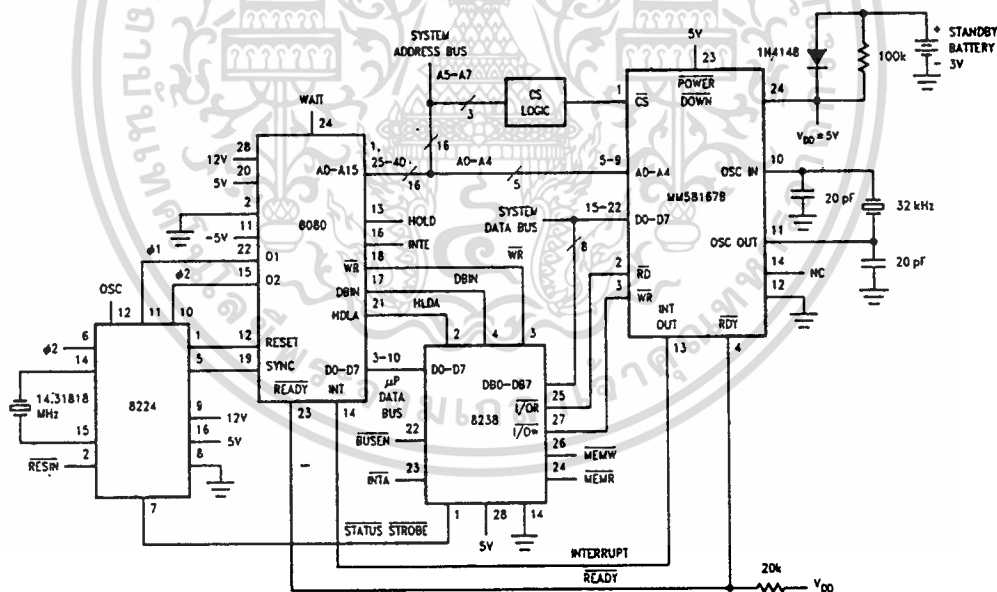
Typical Applications



TL/F/6148-8

Note 5: A ground line or ground plane guard trace should be included between pins 9 and 10 to insure the oscillator is not disturbed by the address line.

FIGURE 5. Typical Connection Diagram



TL/F/6148-9

Note 6: Must use 8238 or equivalent logic to insure advanced I/O pulse; so that the ready output of the MM58167B is valid by the end of $\phi 2$ during the T2 microcycle.

Note 7: L₆₂ ≥ IRS080 + IDL8238 + IWAY58167B.

FIGURE 6. 8080 System Interface with Battery Backup

Block Diagram

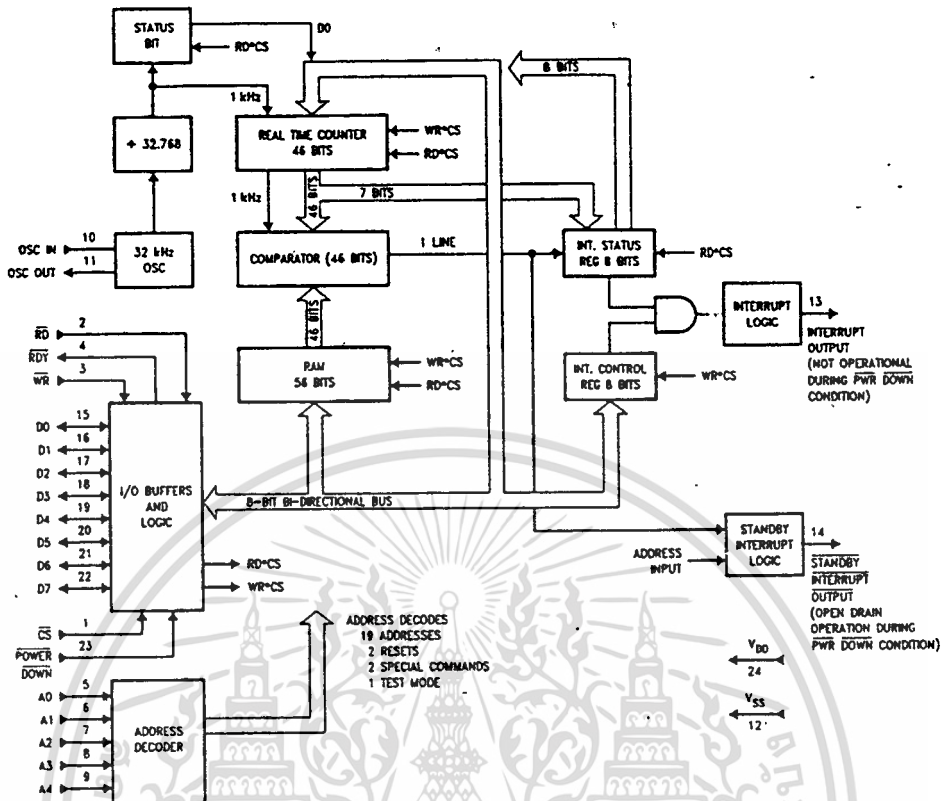


FIGURE 7

TL/F/6148-10

SILICON GATE MOS 8251

8251 BASIC FUNCTIONAL DESCRIPTION

General

The 8251 is a Universal Synchronous/Asynchronous Receiver/Transmitter designed specifically for the 8080 Microcomputer System. Like other I/O devices in the 8080 Microcomputer System its functional configuration is programmed by the systems software for maximum flexibility. The 8251 can support virtually any serial data technique currently in use (including IBM "bi-sync").

In a communication environment an interface device must convert parallel format system data into serial format for transmission and convert incoming serial format data into parallel system data for reception. The interface device must also delete or insert bits or characters that are functionally unique to the communication technique. In essence, the interface should appear "transparent" to the CPU, a simple input or output of byte-oriented system data.

Data Bus Buffer

This 3-state, bi-directional, 8-bit buffer is used to interface the 8251 to the 8080 system Data Bus. Data is transmitted or received by the buffer upon execution of INPUT or OUTPUT instructions of the 8080 CPU. Control words, Command words and Status information are also transferred through the Data Bus Buffer.

Read/Write Control Logic

This functional block accepts inputs from the 8080 Control bus and generates control signals for overall device operation. It contains the Control Word Register and Command Word Register that store the various control formats for device functional definition.

RESET (Reset)

A "high" on this input forces the 8251 into an "Idle" mode. The device will remain at "Idle" until a new set of control words is written into the 8251 to program its functional definition.

CLK (Clock)

The CLK input is used to generate internal device timing and is normally connected to the Phase 2 (TTL) output of the 8224 Clock Generator. No external inputs or outputs are referenced to CLK but the frequency of CLK must be greater than 30 times the Receiver or Transmitter clock inputs for synchronous mode (4 times for asynchronous mode).

WR (Write)

A "low" on this input informs the 8251 that the CPU is outputting data or control words, in essence, the CPU is writing out to the 8251.

RD (Read)

A "low" on this input informs the 8251 that the CPU is inputting data or status information, in essence, the CPU is reading from the 8251.

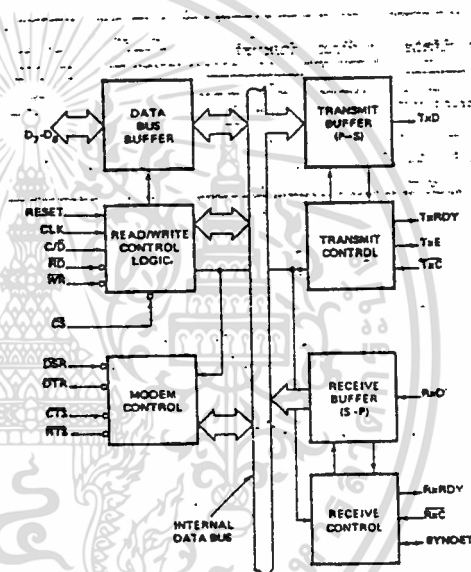
C/D (Control/Data)

This input, in conjunction with the WR and RD inputs informs the 8251 that the word on the Data Bus is either a data character, control word or status information.

1 = CONTROL 0 = DATA

CS (Chip Select)

A "low" on this input enables the 8251. No reading or writing will occur unless the device is selected.



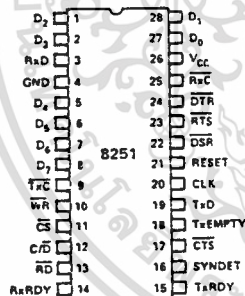
C/D	RD	WR	CS	
0	0	1	0	8251 = DATA BUS
0	1	0	0	DATA BUS = 8251
1	0	1	0	STATUS = DATA BUS
1	1	0	0	DATA BUS = CONTROL
X	X	X	1	DATA BUS = 3-STATE

PROGRAMMABLE COMMUNICATION INTERFACE

- Synchronous and Asynchronous Operation
 - Synchronous:
 - 5-8 Bit Characters
 - Internal or External Character Synchronization
 - Automatic Sync Insertion
 - Asynchronous:
 - 5-8 Bit Characters
 - Clock Rate — 1, 16 or 64 Times Baud Rate
 - Break Character Generation
 - 1, 1½, or 2 Stop Bits
 - False Start Bit Detection
- Baud Rate — DC to 56k Baud (Sync Mode)
DC to 9.6k Baud (Async Mode)
- Full Duplex, Double Buffered, Transmitter and Receiver
- Error Detection — Parity, Overrun, and Framing
- Fully Compatible with 8080 CPU
- 28-Pin DIP Package
- All Inputs and Outputs Are TTL Compatible
- Single 5 Volt Supply
- Single TTL Clock

The 8251 is a Universal Synchronous/Asynchronous Receiver / Transmitter (USART) Chip designed for data communications in microcomputer systems. The USART is used as a peripheral device and is programmed by the CPU to operate using virtually any serial data transmission technique presently in use (including IBM Bi-Sync). The USART accepts data characters from the CPU in parallel format and then converts them into a continuous serial data stream for transmission. Simultaneously it can receive serial data streams and convert them into parallel data characters for the CPU. The USART will signal the CPU whenever it can accept a new character for transmission or whenever it has received a character for the CPU. The CPU can read the complete status of the USART at any time. These include data transmission errors and control signals such as SYNDET, TxEMPT. The chip is constructed using N-channel silicon gate technology.

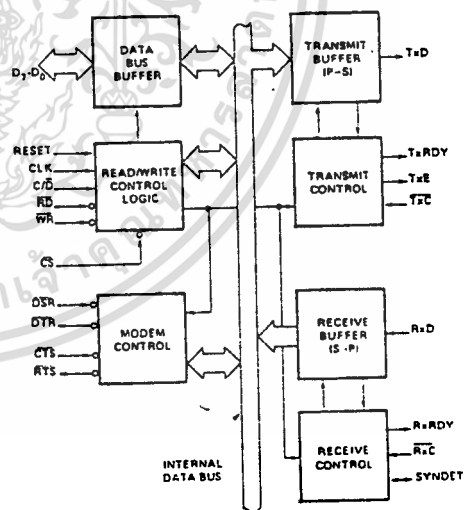
PIN CONFIGURATION



Pin Name	Pin Function
D ₇ -D ₀	Data Bus (8 bits)
C/D	Control or Data in to be Written or Read
RD	Read Data Command
WR	Write Data or Control Command
CS	Chip Enable
CLK	Clock Pulse (TTL)
RESET	Reset
TxC	Transmitter Clock
TxD	Transmitter Data
RxC	Receiver Clock
RxD	Receiver Data
RxRDY	Receiver Ready (has character for 8080)
TxRDY	Transmitter Ready (ready for char. from 8080)

Pin Name	Pin Function
DSR	Data Set Ready
DTR	Data Terminal Ready
SYNDET	Sync Detect
RTS	Request to Send Data
CTS	Clear to Send Data
TxE	Transmitter Empty
Vcc	+5 Volt Supply
GND	Ground

BLOCK DIAGRAM



Mode Instruction Definition

The 8251 can be used for either Asynchronous or Synchronous data communication. To understand how the Mode Instruction defines the functional operation of the 8251 the designer can best view the device as two separate components sharing the same package. One Asynchronous the other Synchronous. The format definition can be changed "on the fly" but for explanation purposes the two formats will be isolated.

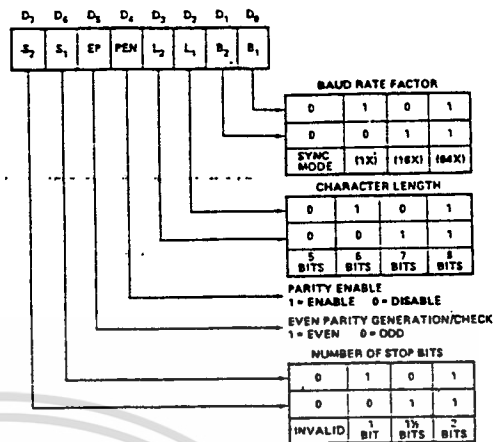
Asynchronous Mode (Transmission)

Whenever a data character is sent by the CPU the 8251 automatically adds a Start bit (low level) and the programmed number of Stop bits to each character. Also, an even or odd Parity bit is inserted prior to the Stop bit(s), as defined by the Mode Instruction. The character is then transmitted as a serial data stream on the TxD output. The serial data is shifted out on the falling edge of TxC at a rate equal to 1, 1/16, or 1/64 that of the TxC, as defined by the Mode Instruction. BREAK characters can be continuously sent to the TxD if commanded to do so.

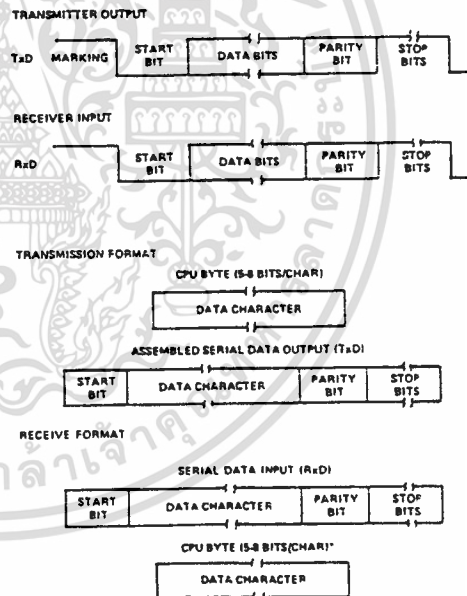
When no data characters have loaded into the 8251 the TxD output remains "high" (marking) unless a Break (continuously low) has been programmed.

Asynchronous Mode (Receive)

The RxD line is normally high. A falling edge on this line triggers the beginning of a START bit. The validity of this START bit is checked by again strobing this bit at its nominal center. If a low is detected again, it is a valid START bit, and the bit counter will start counting. The bit counter locates the center of the data bits, the parity bit (if it exists) and the stop bits. If parity error occurs, the parity error flag is set. Data and parity bits are sampled on the RxD pin with the rising edge of RxC. If a low level is detected as the STOP bit, the Framing Error flag will be set. The STOP bit signals the end of a character. This character is then loaded into the parallel I/O buffer of the 8251. The RxDY pin is raised to signal the CPU that a character is ready to be fetched. If a previous character has not been fetched by the CPU, the present character replaces it in the I/O buffer, and the OVERRUN flag is raised (thus the previous character is lost). All of the error flags can be reset by a command instruction. The occurrence of any of these errors will not stop the operation of the 8251.



Mode Instruction Format, Asynchronous Mode



*NOTE: IF CHARACTER LENGTH IS DEFINED AS 5, 6 OR 7 BITS THE UNUSED BITS ARE SET TO "ZERO"

Asynchronous Mode

DETAILED OPERATION DESCRIPTION

General

The complete functional definition of the 8251 is programmed by the systems software. A set of control words must be sent out by the CPU to initialize the 8251 to support the desired communications format. These control words will program the: BAUD RATE, CHARACTER LENGTH, NUMBER OF STOP BITS, SYNCHRONOUS or ASYNCHRONOUS OPERATION, EVEN/ODD PARITY etc. In the Synchronous Mode, options are also provided to select either internal or external character synchronization.

Once programmed, the 8251 is ready to perform its communication functions. The TxRDY output is raised "high" to signal the CPU that the 8251 is ready to receive a character. This output (TxRDY) is reset automatically when the CPU writes a character into the 8251. On the other hand, the 8251 receives serial data from the MODEM or I/O device, upon receiving an entire character the RxRDY output is raised "high" to signal the CPU that the 8251 has a complete character ready for the CPU to fetch. RxRDY is reset automatically upon the CPU read operation.

The 8251 cannot begin transmission until the TxEN (Transmitter Enable) bit is set in the Command Instruction and it has received a Clear To Send (CTS) input. The TxRDY output will be held in the marking state upon Reset.

Programming the 8251

Prior to starting data transmission or reception, the 8251 must be loaded with a set of control words generated by the CPU. These control signals define the complete functional definition of the 8251 and must immediately follow a Reset operation (internal or external).

The control words are split into two formats:

1. Mode Instruction
2. Command Instruction

Mode Instruction

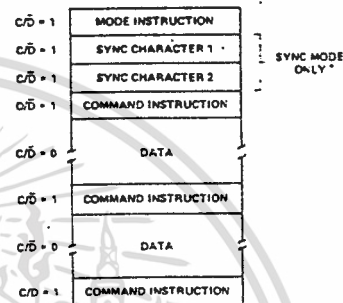
This format defines the general operational characteristics of the 8251. It must follow a Reset operation (internal or external). Once the Mode instruction has been written into the 8251 by the CPU, SYNC characters or Command instructions may be inserted.

Command Instruction

This format defines a status word that is used to control the actual operation of the 8251.

Both the Mode and Command instructions must conform to a specified sequence for proper device operation. The Mode Instruction must be inserted immediately following a Reset operation, prior to using the 8251 for data communication.

All control words written into the 8251 after the Mode Instruction will load the Command Instruction. Command Instructions can be written into the 8251 at any time in the data block during the operation of the 8251. To return to the Mode Instruction format a bit in the Command Instruction word can be set to initiate an internal Reset operation which automatically places the 8251 back into the Mode Instruction format. Command Instructions must follow the Mode Instructions or Sync characters.



*The second SYNC character is skipped if MODE instruction has programmed the 8251 to single character (internal SYNC Mode). Both SYNC characters are skipped if MODE instruction has programmed the 8251 to ASYNC mode.

Typical Data Block

SILICON GATE MOS 8251

COMMAND INSTRUCTION DEFINITION

Once the functional definition of the 8251 has been programmed by the Mode Instruction and the Sync Characters are loaded (if in Sync Mode) then the device is ready to be used for data communication. The Command Instruction controls the actual operation of the selected format. Functions such as: Enable Transmit/Receive, Error Reset and Modem Controls are provided by the Command Instruction.

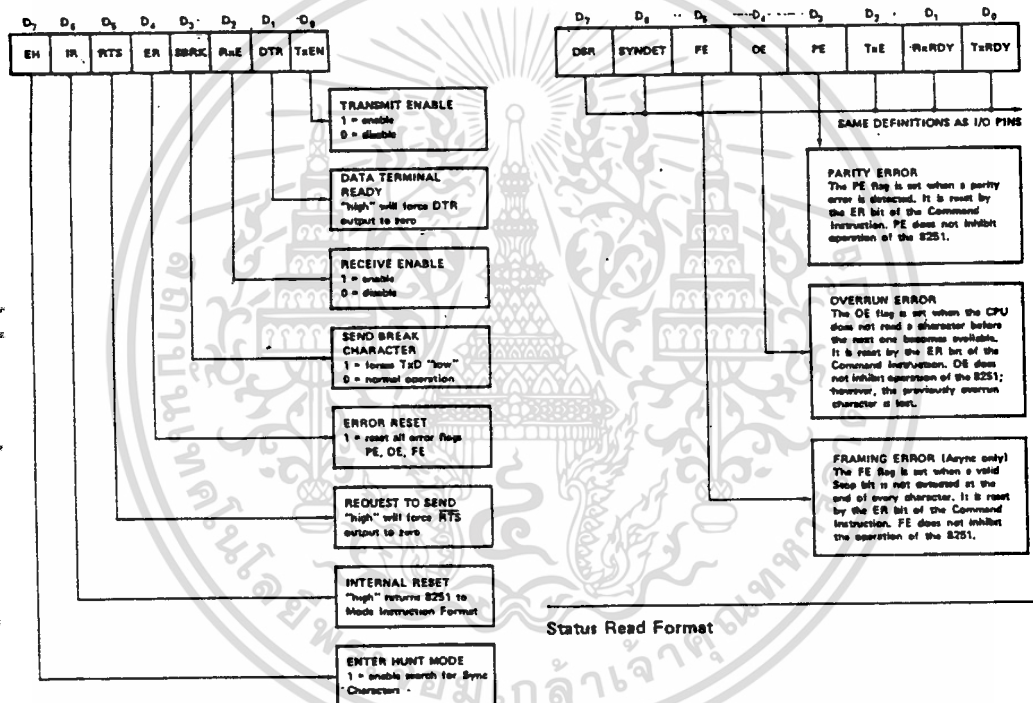
Once the Mode Instruction has been written into the 8251 and Sync characters inserted, if necessary, then all further "control writes" ($C/D = 1$) will load the Command Instruction. A Reset operation (internal or external) will return the 8251 to the Mode Instruction Format.

STATUS READ DEFINITION

In data communication systems it is often necessary to examine the "status" of the active device to ascertain if errors have occurred or other conditions that require the processor's attention. The 8251 has facilities that allow the programmer to "read" the status of the device at any time during the functional operation.

A normal "read" command is issued by the CPU with the C/D input at one to accomplish this function.

Some of the bits in the Status Read Format have identical meanings to external output pins, so that the 8251 can be used in a completely Polled environment or in an interrupt driven environment.



Command Instruction Format

Status Read Format

SILICON GATE MOS 8251

Synchronous Mode (Transmission)

The TxD output is continuously high until the CPU sends its first character to the 8251 which usually is a SYNC character. When the CTS line goes low, the first character is serially transmitted out. All characters are shifted out on the falling edge of TxC. Data is shifted out at the same rate as the TxC.

Once transmission has started, the data stream at TxD output must continue at the TxC rate. If the CPU does not provide the 8251 with a character before the 8251 becomes empty, the SYNC characters (or character if in single SYNC word mode) will be automatically inserted in the TxD data stream. In this case, the TxEMPTY pin is raised high to signal that the 8251 is empty and SYNC characters are being sent out. The TxEMPTY pin is internally reset by the next character being written into the 8251.

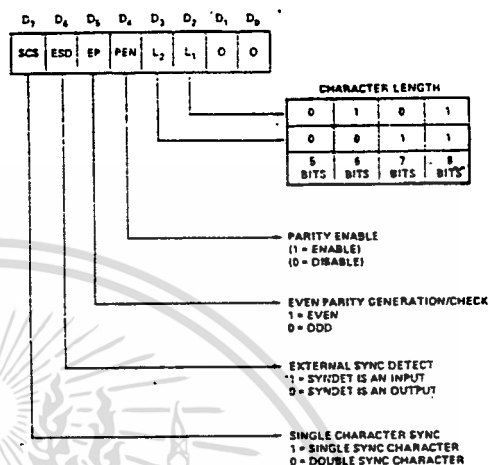
Synchronous Mode (Receive)

In this mode, character synchronization can be internally or externally achieved. If the internal SYNC mode has been programmed, the receiver starts in a HUNT mode. Data on the RxD pin is then sampled in on the rising edge of RxC. The content of the Rx buffer is continuously compared with the first SYNC character until a match occurs. If the 8251 has been programmed for two SYNC characters, the subsequent received character is also compared; when both SYNC characters have been detected, the USART ends the HUNT mode and is in character synchronization. The SYNDET pin is then set high, and is reset automatically by a STATUS READ.

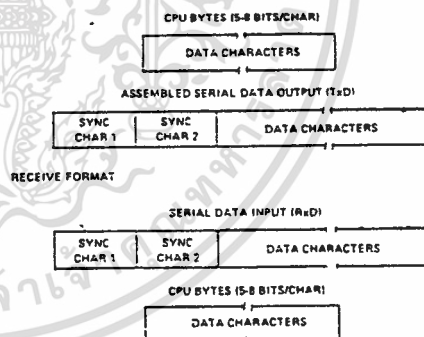
In the external SYNC mode, synchronization is achieved by applying a high level on the SYNDET pin. The high level can be removed after one RxC cycle.

Parity error and overrun error are both checked in the same way as in the Asynchronous Rx mode.

The CPU can command the receiver to enter the HUNT mode if synchronization is lost.



Mode Instruction Format, Synchronous Mode



Synchronous Mode, Transmission Format

SILICON GATE MOS 8251

D.C. Characteristics: $T_A = 0^\circ\text{C to } 70^\circ\text{C}$; $V_{CC} = +5V \pm 5\%$; $V_{SS} = 0V$

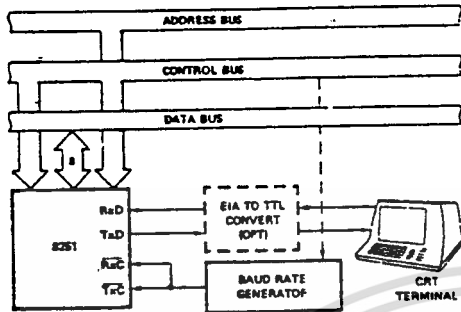
Symbol	Parameter	Min.	Typ.	Max.	Unit	Test Conditions
V_{IL}	Input Low Voltage			.8	V	
V_{IH}	Input High Voltage	2.0			V	
V_{OL}	Output Low Voltage			.4	V	$I_{OL} = 2.0\text{mA (DB}_{0-7}\text{)}$ 1.6mA (Others)
V_{OH}	Output High Voltage	2.4			V	$I_{OH} = 250\mu\text{A (DB}_{0-7}\text{)}$, $100\mu\text{A (Others)}$
I_{CC}	Power Supply Current		80		mA	
I_{LI}	Input Load Current		10		μA	$V_{IN} = 0V \text{ to } 5.25V$
I_{LOL}	Output Leakage Current (DB Low)		-100		μA	$V_{OUT} = 0.4V$
I_{LOH}	Output Leakage Current (DB High)		+10		μA	$V_{OUT} = V_{CC}$

A.C. Characteristics: $T_A = 0^\circ\text{C to } 70^\circ\text{C}$; $V_{CC} = +5V \pm 5\%$; $V_{SS} = 0V$

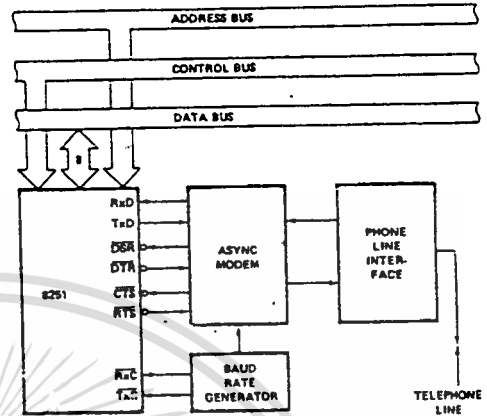
Symbol	Parameter	Min.	Typ.	Max.	Unit	Test Conditions
t_{CY}	Clock Period		400		ns	
t_{CW}	Clock Pulse Width		220		ns	
t_{CD}	Clock Pulse Delay		200		ns	
t_R, t_F	Clock Rise and Fall Time		50		ns	
t_{WR}	Write Pulse Width		400		ns	
t_{DS}	Data Set-Up Time for WRITE		300		ns	
t_{DH}	Data Hold Time for WRITE		20		ns	
t_{DD}	Data Delay from READ		350		ns	$C_L = 150\text{pF}$
t_{FD}	READ to Data Floating		160		ns	$C_L = 150\text{pF}$
t_{AW}	Address Stable before WRITE	0			ns	
t_{WA}	Address Hold Time for WRITE		40		ns	
t_{AR}	Address Stable before READ		0		ns	
t_{RA}	Address Hold Time for READ		0		ns	
t_{DTx}	TxD Delay from Rising Edge of TxCl		300		ns	
t_{SRx}	Rx Data Set-Up Time to Sampling Pulse		500		ns	
t_{HRx}	Rx Data Hold Time to Sampling Pulse		6		CLK Period	
f_{Tx}	Transmitter Input Clock Frequency 1X Baud Rate 16X and 64X Baud Rate		56 615		KHz KHz	
f_{Rx}	Receiver Input Clock Frequency 1X Baud Rate 16X and 64X Baud Rate		56 615		KHz KHz	
t_{Tx}	TxRDY Delay from Center of Data Bit		6		CLK Period	$C_L = 50\text{pF}$
t_{Rx}	RxRDY Delay from Center of Data Bit		6		CLK Period	
t_{IS}	Internal SYNDY Delay from Center of Data Bit		6		CLK Period	
t_{ES}	External SYNDY Set-Up Time before Rising Edge of RxCl		6		CLK Period	

SILICON GATE MOS 8251

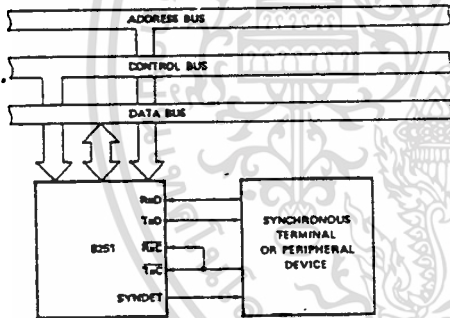
APPLICATIONS OF THE 8251



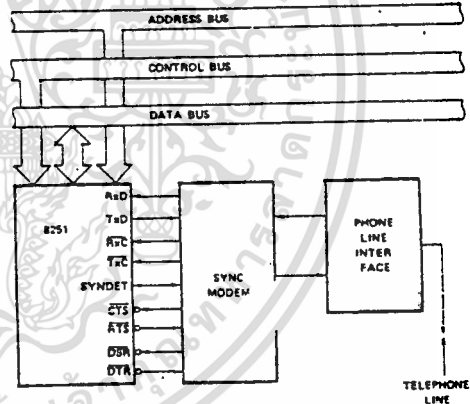
Asynchronous Serial Interface to CRT Terminal,
DC-9600 Baud



Asynchronous Interface to Telephone Lines



Synchronous Interface to Terminal or Peripheral Device



Synchronous Interface to Telephone Lines

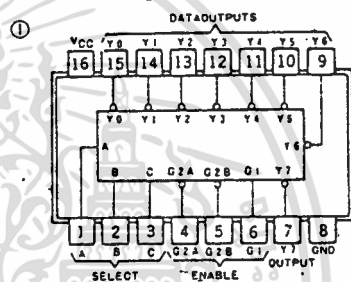
54138/74138 3-Line-to-8-Line Decoder

	Schottky TTL				High-Speed TTL				Low-Power Schottky TTL				Standard TTL				Low-Power TTL			
	Device Type	Package			Device Type	Package			Device Type	Package			Device Type	Package			Device Type	Package		
		C	P	MICF		C	P	MICF		C	P	MICF		C	P	MICF		C	P	MICF
T.I.	SN54LS138	J	U	W					SN54LS138	J	U	W								
	SN74LS138	J	U	W					SN74LS138	J	U	W								
FAIRCHILD	74ALS138	ND							74ALS138	ND										
	74ALS138	ND							74ALS138	ND										
MOTOROLA									SN74LS138	ND										
N.S.C.	DM74LS138								DM74LS138											
PHILIPS	N74LS138								N74LS138											
SIGNETICS	SS4LS138	F	D	B																
	N74LS138	F	D	B					N74LS138	F	D	B								
SIEMENS																				
FUJITSU									74LS138											
HITACHI									HD74LS138											
MTSUBISHI	M74LS138								M74LS138											
NEC									74LS138											
TOSHIBA																				

Electrical Characteristics SN54LS138/SN74LS138

absolute maximum ratings over operating free-air temperature range					
Supply voltage, V_{CC}	TV	Operating free-air temperature range	SN54LS	-55°C to 125°C	
Input voltage	TV		SN74LS	0°C to 70°C	
		Storage temperature range		-65°C to 150°C	
recommended operating conditions					
	SN54LS138			SN74LS138	
	MIN	NOM	MAX	MIN	NOM
Supply voltage, V_{CC}	4.5	5	5.5	4.75	5
High-level output current, I_{OH}			400		400
Low-level output current, I_{OL}			4		8
Operating free-air temperature, T_A	-55	125	0		70
electrical characteristics over recommended operating free-air temperature range					
PARAMETER	TEST CONDITIONS			MIN	TYP
V_{IH} High-level input voltage				2	
V_{IL} Low-level input voltage				0.8	
V_I Input clamp voltage	$V_{CC} = \text{MIN.}$, $I_I = -18 \text{ mA}$			1.5	
V_{OH} High-level output voltage	$V_{CC} = \text{MIN.}$, $V_{IH} = 2 \text{ V}$, $V_{II} = 0.8 \text{ V}$, $I_{OH} = 400 \mu\text{A}$	SN54LS	2.5	3.4	
		SN74LS	2.7	3.4	
V_{OL} Low-level output voltage	$V_{CC} = \text{MIN.}$, $V_{IH} = 2 \text{ V}$, $V_{II} = 0.8 \text{ V}$, $I_{OL} = 8 \text{ mA}$		0.35	0.5	
I_I Input current at maximum input voltage	$V_{CC} = \text{MAX.}$, $V_I = \text{TV}$			0.1	
I_{IH} High-level input current	$V_{CC} = \text{MAX.}$, $V_I = 2.7 \text{ V}$			20	
I_{IL} Low-level input current	$V_{CC} = \text{MAX.}$, $V_I = 0.8 \text{ V}$			0.4	
I_{OS} Short-circuit output current	$V_{CC} = \text{MAX.}$		-20	-100	
I_{CC} Supply current	$V_{CC} = \text{MAX.}$, Outputs enabled and open		5.3	10	
t_{PLH} from Binary select	$V_{CC} = 5 \text{ V}$, $T_A = 25^\circ\text{C}$, $C_L = 15 \text{ pF}$, $R_L = 2 \text{ k}\Omega$	2	13	20	
t_{PHL} to Any output		3	27	41	
t_{PLH} from Enable		2	12	18	
t_{PHL} to Any output		3	21	32	
t_{PLH} from Enable		2	17	26	
t_{PHL} to Any output		3	25	38	

Pin Assignment (Top View)



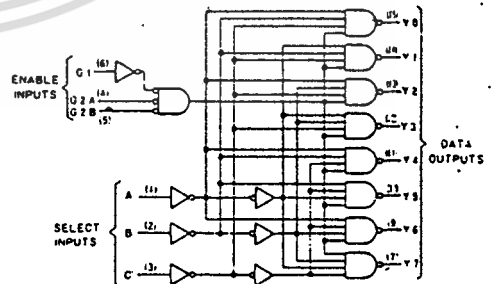
positive logic:
see function table

Function Table

INPUTS		OUTPUTS							
ENABLE	SELECT								
G_1	G_2^A C B A	Y_0	Y_1	Y_2	Y_3	Y_4	Y_5	Y_6	Y_7
X	H X X X	H	H	H	H	H	H	H	H
L	X X X X	H	H	H	H	H	H	H	H
H	L L L L	L	H	H	H	H	H	H	H
H	L L L H	H	L	H	H	H	H	H	H
H	L L H L	H	H	L	H	H	H	H	H
H	L L H H	H	H	H	L	H	H	H	H
H	L H L L	H	H	H	H	L	H	H	H
H	L H L H	H	H	H	H	H	L	H	H
H	L H H L	H	H	H	H	H	H	L	H
H	L H H H	H	H	H	H	H	H	H	L

$G_2 = G_2^A + G_2^B$
H = high level, L = low level, X = irrelevant

Functional Block Diagram



SN54LS138 DECODER/DEMULPLEXER

† For conditions shown as MIN or MAX, use the appropriate value specified under recommended operating conditions for the applicable device type.

‡ All typical values are at $V_{CC} = 5 \text{ V}$, $T_A = 25^\circ\text{C}$.

§ Not more than one output should be shorted at a time, and duration of the short-circuit test should not exceed one second.

¶ t_{PLH} = propagation delay time, low-to-high-level output

‖ t_{PHL} = propagation delay time, high-to-low-level output

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5447A/7447A BCD-to-Seeven Segment Decder/Driver (15V Output)

	Schottky TTL				High-Speed TTL				Low-Power Schottky TTL				Standard TTL				Low-Power TTL			
	Device Type	Package			Device Type	Package			Device Type	Package			Device Type	Package			Device Type	Package		
		C	P	M/C/F		C	P	M/C/F		C	P	M/C/F		C	P	M/C/F		C	P	M/C/F
T.I.								SN54LS47				SN5447A				SN54LS47				
FAIRCHILD								SN74LS47				SN7447A				SN74LS47				
MOTOROLA												MC5447								
N.S.C.								DM54LS47				DM5447A								
PHILIPS								DM74LS47				DM7447A								
SIGNETICS												N7447								
SIEMENS												N7447								
FUJITSU												FLL171W								
HTACHI								HTLS47												
MTSUBISHI								MT74LS47				MT7447A / MT2532								
NEC								MTLS47				MS3247								
TOSHIBA												μPB7047								
												TC3447A								

Electrical Characteristics SN54LS47A-SN74LS47A

absolute maximum ratings over operating
free-air temperature range

Supply voltage V_{CC}	7V	Operating free-air temperature range	SN54LS	55°C to 125°C
Input voltage	7V		SN74LS	0°C to 70°C
Current forced into any output in the off state	1 mA	Storage temperature range		-65°C to 150°C

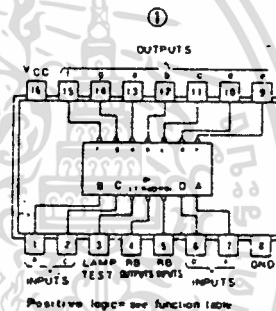
recommended operating conditions

	SN54LS47A				SN74LS47A				UNIT
	MIN	NOM	MAX	MIN	NOM	MAX			
Supply voltage V_{CC}	4.5	5	5.5	4.75	5	5.25	V		
On-state output voltage V_{OL} (on)	8 mA		15			15	V		
On-state output current I_{OL} (on)	8 mA		17			24	mA		
High-level output current I_{OH}	BI RBO		50			50	mA		
Low-level output current I_{OL}	BI RBO		1.6			3.2	mA		
Operating free-air temperature, T_A		-55	125	0	70	175	°C		

electrical actics over recommended operating
free-air temperature range

PARAMETER	TEST CONDITIONS	MIN	TY	MAX	UNIT
V_{IH}	High-level input voltage		2		V
V_{IL}	Low-level input voltage			0.8	V
V_I	Input clamp voltage, any input except BI RBO	$V_{CC} - MIN.$		1.5	V
V_{OH}	High-level output voltage BI RBO	$V_{CC} - MIN.$		2.4	V
V_{OL}	Low-level output voltage BI RBO	$V_{CC} - MIN.$		0.25	V
I_O (off)	Off-state output current	$V_{CC} - MAX.$		250	μA
I_O (on)	On-state output current	$V_{CC} - MAX.$		0.25	V
I_i	Input current at any input maximum input voltage except BI RBO	$V_{CC} - MAX.$		0.1	mA
I_{IH}	High-level input current	$V_{CC} - MAX.$		20	μA
I_{IL}	Low-level input current	$V_{CC} - MAX.$		-0.4	mA
I_{OS}	Short-circuit output current	$V_{CC} - MAX.$		-2	mA
I_{CC}	Supply current	$V_{CC} - MAX.$		7	mA
t_{off}	Turnoff time from input			100	ns
t_{on}	Turnon time from input			100	ns
t_{off}	Turnoff time from BI RBO input			100	ns
t_{on}	Turnon time from BI RBO input			100	ns

Pin Assignment (Top View)



Positive logic = function table

- NOTES:
- The blanking input (BI) must be open or held at a high logic level when output functions 0 through 15 are desired. The reprogram input (RBI) must be open or held at a high logic level when a decimal zero is not desired.
 - When a low logic level is applied directly to the blanking input (BI), all segment outputs are off regardless of the level of any other input.
 - When reprogram input (RBI) and inputs A, B, C, and D are at a low level with the same test input high, all segment outputs go off and the reprogram output (RBO) goes to a low level (reprogram condition).
 - When the blanking input/reprogram output (BI/RBO) is open or held high and a low is applied to the blanking input, all segment outputs are on.
 - BI RBO is a NAND logic driving as blanking input (BI) and/or reprogram output (RBO).

For conditions shown as MIN or MAX, use the appropriate value specified under recommended operating conditions for the applicable type.

All typical values are at $V_{CC} = 5V$, $T_A = 25°C$.

t_{off} measured with all outputs open and all inputs at 4.5V.

t_{on} corresponds to t_{PL} and t_{off} corresponds to t_{PHL} .

CONTINUED ON NEXT PAGE

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

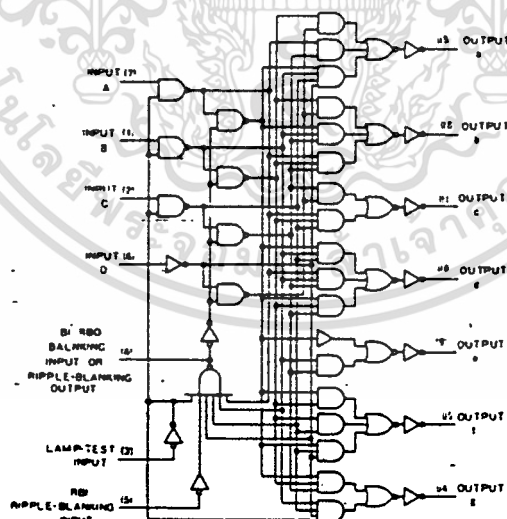
5447A/7447A(CONTINUED)

Function Table

DECIMAL OR FUNCTION	INPUTS						B ¹ RB0≡	OUTPUTS								NOTES
	LT	RB	D	C	B	A		a	b	c	d	e	f	g		
0	H	H	L	L	L	L	H	ON	ON	ON	ON	ON	ON	OFF	1	
1	H	X	L	L	L	M	H	OFF	ON	ON	OFF	OFF	OFF	OFF	1	
2	H	X	L	L	M	L	H	ON	ON	OFF	ON	ON	OFF	ON		
3	H	X	L	L	M	M	H	ON	ON	ON	ON	OFF	OFF	ON		
4	H	X	L	M	L	L	H	OFF	ON	ON	OFF	OFF	ON	ON		
5	H	X	L	M	L	M	H	ON	OFF	ON	ON	OFF	ON	ON		
6	H	X	L	M	M	L	H	OFF	OFF	ON	ON	ON	ON	ON		
7	H	X	L	M	M	M	H	ON	ON	ON	OFF	OFF	OFF	OFF		
8	H	X	M	L	L	L	H	ON	ON	ON	ON	ON	ON	ON		
9	H	X	M	L	L	M	H	ON	ON	ON	OFF	OFF	ON	ON		
10	H	X	M	L	M	L	H	OFF	OFF	OFF	ON	ON	OFF	ON		
11	H	X	M	L	M	M	H	OFF	OFF	ON	ON	OFF	OFF	ON		
12	H	X	M	M	L	L	H	OFF	ON	OFF	OFF	OFF	ON	ON		
13	H	X	M	M	L	M	H	ON	OFF	OFF	ON	OFF	ON	ON		
14	H	X	M	M	M	L	H	OFF	OFF	OFF	ON	ON	ON	ON		
15	H	X	M	M	M	M	H	OFF	OFF	OFF	OFF	OFF	OFF	OFF		
B:	X	X	X	X	X	X	L	OFF	OFF	OFF	OFF	OFF	OFF	OFF	2	
RB:	L	L	L	L	L	L	L	OFF	OFF	OFF	OFF	OFF	OFF	OFF	3	
LT	L	X	X	X	X	X	H	ON	ON	ON	ON	ON	ON	ON	4	

H = high level, L = low level, X = irrelevant

Schematic



'47A 'LSA47 'L47 BCD-TO-SEVEN-SEGMENT DECODER DRIVER

5474/7474 Dual D-Type Positive-Edge-Triggered Flip-Flop with Preset and Clear

	Schottky TTL			High-Speed TTL			Low-Power Schottky TTL			Standard TTL			Low-Power TTL		
	Device Type	Package		Device Type	Package		Device Type	Package		Device Type	Package		Device Type	Package	
T.I.	SN54S74	JG	W2	SN54H74	JG	W2	SN54LS74	JG	W2	SN5474	JG	W2	SN54L74	JG	W2
	SN74S74	JG	W2	SN74H74	JG	W2	SN74LS74	JG	W2	SN7474	JG	W2	SN74L74	JG	W2
FAIRCHILD	FMS5474	FM5474	OJ	FMS5474	FM5474	OJ	FMS5474	FM5474	OJ	FMS5474	FM5474	OJ	FMS5474	FM5474	OJ
	FC74S74	FC9574	OJ	FC74H74	FC9474	OJ	FC74LS74	FC9474	OJ	FC7474	FC9474	OJ	FC74L74	FC9474	OJ
MOTOROLA															
N.S.C.	DM74S74			DM54H74	JG	W2	DM54LS74			DM5474	JG	W2	DM54L74	JG	W2
	DM7474			DM74H74	JG	W2	DM74LS74			DM7474	JG	W2	DM74L74	JG	W2
PHILIPS															
SIGNETICS	N74S74			GJ74H74			N74LS74			FJ74S74			FJ74L74		
	S54S74			S54H74	FJ	W2	S54LS74			S5474	FJ	W2	S54L74	FJ	W2
SIEMENS	N74S74			N74H74	FJ	W2	N74LS74			N7474	FJ	W2	N74L74	FJ	W2
FUJITSU															
HITACHI	HD74S74						HD74LS74			HD7474			HD74L74		
MITSUBISHI	M74S74						M74LS74			M7474			M74L74		
NEC	74S74						74LS74			7474			74L74		
TOSHIBA															

Electrical Characteristics SN54LS74/SN74LS74

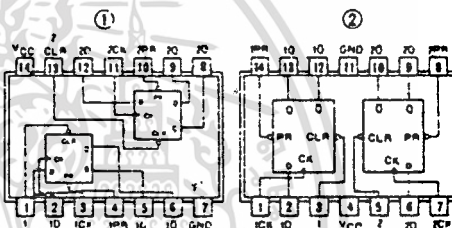
absolute maximum ratings over operating temperature range

parameter	typical	maximum	unit
V _{CC} supply voltage	5.0	7.0	V
V _{IL} input voltage	0.8	1.5	V
V _{OL} output voltage	0.4	0.5	V
I _{IL} input current	-1.0	-1.5	mA
I _{OL} output current	10	15	mA
t _{PLH} propagation delay time	15	20	ns
t _{PHL} propagation delay time	15	20	ns
t _{CL} clock-to-output delay time	15	20	ns
t _{CH} clock-to-output delay time	15	20	ns
t _{SET} setup time	10	15	ns
t _{HD} hold time	10	15	ns
f _{CLK} clock frequency	10	15	kHz

electrical characteristics over recommended operating free-air temperature range

parameter	test conditions	min	typ	max	unit
V _{IL} High-level input voltage	V _{CC} = 5V, I _{IL} = -1.5mA	0.8	1.0	1.5	V
V _{OL} Low-level output voltage	V _{CC} = 5V, I _{OL} = 15mA	0.4	0.5	0.5	V
I _{IL} input current	V _{CC} = 5V, V _{IL} = 0.8V	-1.0	-1.5	-1.5	mA
I _{OL} output current	V _{CC} = 5V, V _{OL} = 0.5V	10	15	15	mA
t _{PLH} propagation delay time	V _{CC} = 5V, V _{IL} = 0.8V, V _{OL} = 0.5V	15	20	20	ns
t _{PHL} propagation delay time	V _{CC} = 5V, V _{IL} = 0.8V, V _{OL} = 0.5V	15	20	20	ns
t _{CL} clock-to-output delay time	V _{CC} = 5V, V _{IL} = 0.8V, V _{OL} = 0.5V	15	20	20	ns
t _{CH} clock-to-output delay time	V _{CC} = 5V, V _{IL} = 0.8V, V _{OL} = 0.5V	15	20	20	ns
t _{SET} setup time	V _{CC} = 5V, V _{IL} = 0.8V, V _{OL} = 0.5V	10	15	15	ns
t _{HD} hold time	V _{CC} = 5V, V _{IL} = 0.8V, V _{OL} = 0.5V	10	15	15	ns
f _{CLK} clock frequency	V _{CC} = 5V, V _{IL} = 0.8V, V _{OL} = 0.5V	10	15	15	kHz

Pin Assignments (Top View)

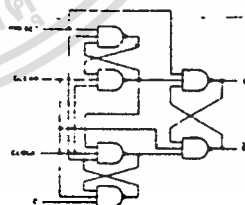


Functional Table

74, 74A, LS74, 74A (See Note 2)

INPUTS			OUTPUTS	
PRESET	CLEAR	CLOCK	Q	Q'
L	H	X	X	H
H	L	X	X	L
L	L	X	X	H
H	H	X	X	L
H	H	L	X	H
H	H	L	X	L

Functional Block Diagram



74, 74A, LS74, 74A Dual D-Type Flip-Flop with Clear and Preset

- With all outputs open, Q is measured with the Q and Q' outputs high in turn. At the time of measurement, the clock input is grounded.
- H = high level (steady state), L = low level (steady state), X = irrelevant, * = transition from low to high level. Q = the value of Q before the indicated input conditions were established. This configuration is nonstable. Set is 1, if not preset when preset and clear inputs return to their negative (high) level.

* For conditions shown as MIN or MAX, use the appropriate value specified under recommended operating conditions.
 † All typical values are at V_{CC} = 5V, T_A = 25°C.
 ‡ Not more than one output should be shorted at a time.
 § t_{PLH} = propagation delay time, low-to-high-level output.
 ¶ t_{PHL} = propagation delay time, high-to-low-level output.
 †† The arrow indicates the edge of the clock pulse used for reference to the rising edge.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

54192/74192 Synchronous Decade Up/Down Counter (Dual clock with Clear)

	Schottky TTL			High-Speed TTL			Low-Power Schottky TTL			Standard TTL			Low-Power TTL		
	Device Type	Package		Device Type	Package		Device Type	Package		Device Type	Package		Device Type	Package	
		C P M CF			C P M CF			C P M CF			C P M CF			C P M CF	
T. I.							SN54LS192	J J	W1	SN54192	J J	W1	SN54LS192	J J	W1
FAIRCHILD							SN74LS192	J J	W1	SN74192	J J	W1	SN74LS192	J J	W1
MOTOROLA							DM54LS192	J J	W1	DM54192	J J	W1	DM74LS192	J J	W1
N. S. C.							DM74LS192	J J	W1	DM74192	J J	W1	DM74LS192	J J	W1
PHILIPS							N74LS192	J J	W1	N74192	J J	W1	N74LS192	J J	W1
SIGNETICS							N74LS192	J J	W1	N74192	J J	W1	N74LS192	J J	W1
SIEMENS							N74LS192	J J	W1	N74192	J J	W1	N74LS192	J J	W1
FUJITSU							74LS192	J J	W1	74192	J J	W1	74LS192	J J	W1
HITACHI							HD74LS192	J J	W1	HD74192	J J	W1	HD74LS192	J J	W1
mitsubishi							M74LS192	J J	W1	M74192	J J	W1	M74LS192	J J	W1
NEC							74LS192	J J	W1	74192	J J	W1	74LS192	J J	W1
TOSHIBA							74LS192	J J	W1	74192	J J	W1	74LS192	J J	W1

Electrical Characteristics SN54LS192/SN74LS192

absolute maximum ratings over operating free-air temperature range

Supply voltage V_{CC}	TV	Operating free-air temperature range	SN54LS	-55°C to 125°C
Input voltage	TV	Storage temperature range	SN74LS	0°C to 70°C
				-65°C to 150°C

recommended operating conditions

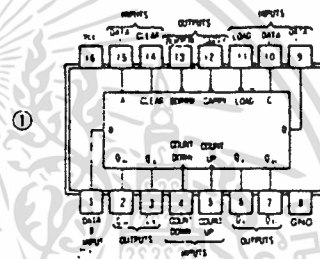
	SN54LS192			SN74LS192			UNIT
	MIN	NOM	MAX	MIN	NOM	MAX	
Supply voltage V_{CC}	4.5	5	5.5	4.75	5	5.25	V
Maximum output current I_{OH}			-400			-400	μ A
Low-level output current I_{OL}			4			8	mA
Count frequency f_{CLK}	0		25	0		25	MHz
Set-up time t_{SU}	20		20				ns
Data input rise time t_{RI}	20		20				ns
Data hold time t_{HD}	0		0				ns
Operating temperature T_A	-55		125	0		70	°C

electrical characteristics over recommended operating free-air temperature range

PARAMETER*	TEST CONDITIONS†	MIN	TYP‡	MAX	UNIT
V_{IH} High-level input voltage		2			V
V_{IL} Low-level input voltage				0.8	V
V_{OL} Low-level output voltage	$V_{CC} = \text{MIN.}$, $I_{OH} = -18 \text{ mA}$			-1.5	V
V_{OH} High-level output voltage	$V_{CC} = \text{MIN.}$, $V_{IH} = 2 \text{ V}$, $I_{OL} = -400 \mu\text{A}$	2.7	3	X	V
V_{OL} Low-level output voltage	$V_{CC} = \text{MIN.}$, $V_{IH} = 2 \text{ V}$, $I_{OH} = 8 \text{ mA}$	0.35	0.5		V
I_{OH} Output current at maximum low-level voltage	$V_{CC} = \text{MAX.}$, $V_{IL} = 0.8 \text{ V}$			0.1	mA
I_{OL} Output current at maximum high-level voltage	$V_{CC} = \text{MAX.}$, $V_{IH} = 2 \text{ V}$			20	μ A
I_{OH} Output current at maximum high-level voltage	$V_{CC} = \text{MAX.}$, $V_{IL} = 0.8 \text{ V}$			0.8	mA
I_{OS} Short-circuit output current	$V_{CC} = \text{MAX.}$	SN54LS: 20 SN74LS: 20	100	100	mA
I_{CC} Supply current	$V_{CC} = \text{MAX.}$, See Note 1	SN54LS: 19 SN74LS: 19	34	34	mA
f_{MAX} Maximum clock frequency		25	32		MHz
t_{PLH} Propagation delay time from Count to output	$V_{CC} = 5 \text{ V}$		17	26	ns
t_{PLC} Propagation delay time from Carry to output			18	24	ns
t_{PLD} Propagation delay time from Data to output	$T_A = 25^\circ\text{C}$		16	24	ns
t_{PLQ} Propagation delay time from Clear to output			15	24	ns
t_{PLH} Propagation delay time from Count to output	$C_L = 15 \text{ pF}$		27	38	ns
t_{PLC} Propagation delay time from Carry to output			30	47	ns
t_{PLD} Propagation delay time from Data to output	$P_L = 2$		24	40	ns
t_{PLQ} Propagation delay time from Clear to output			25	40	ns
t_{PLH} Propagation delay time from Count to output			23	35	ns

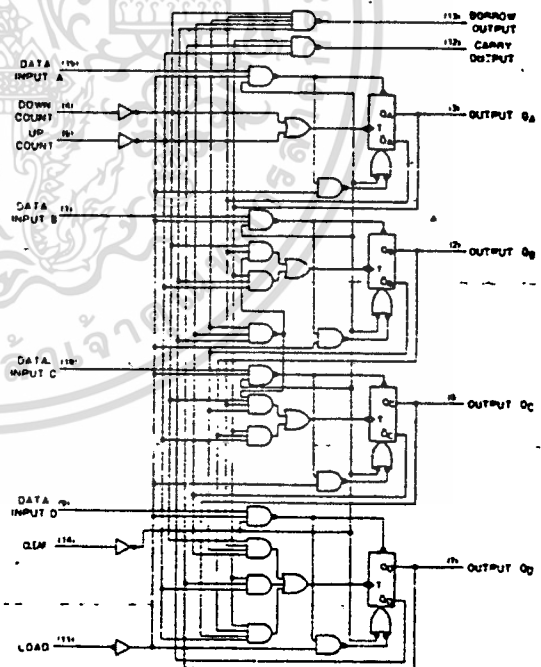
NOTES
 * I_{CC} is measured with all outputs open, clear and load inputs grounded, and all other inputs at 5V.
 † Clear overrides load, data and count inputs.
 ‡ When counting up, count-down input must be high when counting down, count-up input must be high.

Pin Assignment (Top View)



Positive logic:
 Low input to load and $Q_A = A$,
 $Q_B = B$, $Q_C = C$, and $Q_D = D$

Functional Block Diagram



... Dynamic input activated by a transition from a high level to a low level.
 * 192, 74192, 74192 SYNCHRONOUS 4-BIT UP/DOWN COUNTER

† For conditions shown as MIN or MAX, use the appropriate value specified under recommended operating conditions for the applicable type.
 ‡ At 100% duty cycle at $V_{CC} = 5 \text{ V}$, $T_A = 25^\circ\text{C}$.
 * Not more than one output should be shorted at a time.
 † t_{PLH} = propagation delay time, low-to-high-level output.
 ‡ t_{PLC} = propagation delay time, high-to-low-level output.

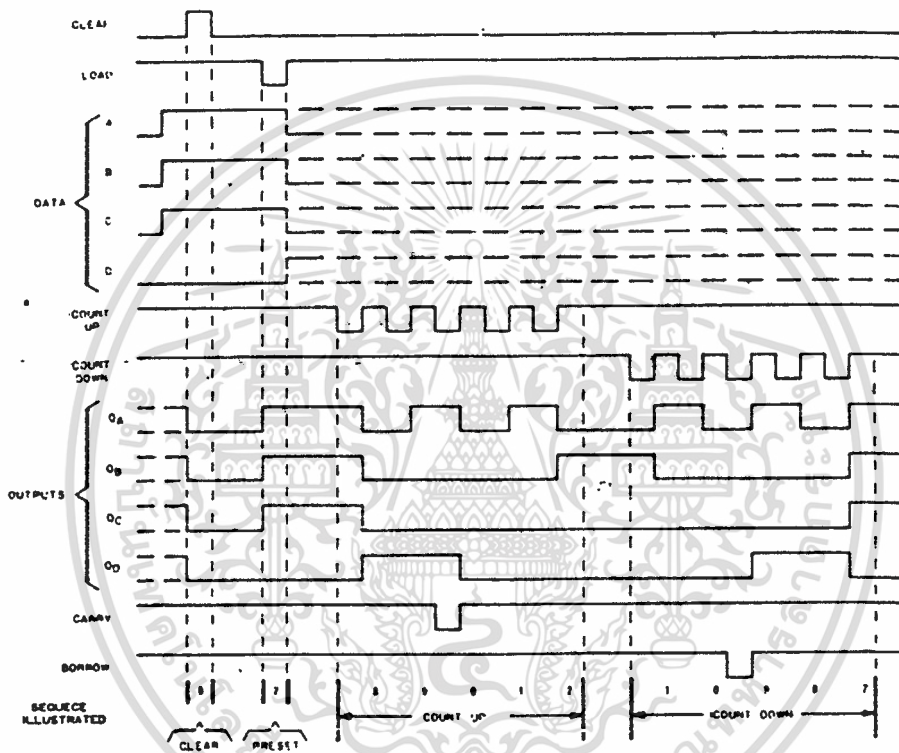
CONTINUED ON NEXT PAGE

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

typical clear, load, and count sequences

Illustrated below is the following sequence

- 1 Clear outputs to zero
- 2 Load (preset) to BCD seven
- 3 Count up to eight, nine, carry, zero, one, and two
- 4 Count down to one, zero, borrow, nine, eight and seven



- NOTES
- A Clear overrides load, data, and count inputs
 - B When counting up, count down input must be high when counting down, count up input must be high