สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ระบบรายงานค่าพารามิเตอร์ของดาวเทียม

TELEMETRY SYSTEM

โดย

นางสาวอิฎฐพร จันทรสถาพร

นายอิทธิพงษ์ วงศ์แสนสุข

นายอิทธิพร เมฆวิบูลย์

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิศวกรรมโทรคมนาคม

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2537

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

034948

ระบบรายงานค่าพารามิเตอร์ของดาวเทียม

TELEMETRY SYSTEM

โดย

นางสาวอิฏฐพร จันทรสถาพร

นายอิทธิพงษ์ วงศ์แสนสุข

นายอิทธิพร เมฆวิบูลย์

อาจารย์ที่ปรึกษา

รศ.ณรงค์ เหมกรณ์

ปริญญานิพนธ์สำหรับปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิศวกรรมโทรคมนาคม

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2537

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระบบรายงานค่าพารามิเตอร์ของดาวเทียม

Telemetry system

โดย นางสาวอิฏฐพร จันทรสถาพร
นายอิทธิพงษ์ วงศ์แสนสุข
นายอิทธิพร เมฆวิบูลย์
อาจารย์ที่ปรึกษา รศ.ณงค์ เหมกรณ์
ปีการศึกษา 2537

บทคัดย่อ

ปริญญานิพนธ์ฉบับนี้เสนอเกี่ยวกับโครงงานการจำลองระบบรายงานค่าพารามิเตอร์ของดาวเทียม (Telemetry system) ซึ่งเป็นระบบย่อยที่สำคัญสำหรับดาวเทียมในส่วนของโครงงานที่ได้ทำได้นำการ์ด Z84C11_PLUS มาประยุกต์ใช้เป็นหน่วยประมวลผลกลาง(Central Processing Unit:CPU) ที่จะควบคุมการทำงานของระบบวัดทางไกลที่จำลองขึ้น โดยได้ทำการทดลองเก็บข้อมูล 3 ชุดพร้อมทั้งชั่วโมง วัน เดือน ปี เก็บลงหน่วยความจำบนการ์ดแล้วส่งข้อมูลอนุกรมของสัญญาณดิจิตอลเหล่านั้นไปยังภาคมอดูเลทแบบฟรีเควนซีชิฟคีย์อิง(Frequency Shift Keying:FSK) ซึ่งจะเปลี่ยนสัญญาณอินพุทที่เป็นดิจิตอลได้เป็นสัญญาณเอ้าพุทที่เป็นสัญญาณคนาล็อกส่งเข้าเครื่องส่งวิทยุออกไปทางสายอากาศ สัญญาณวิทยุจะเดินทางเข้าไปยังเครื่องรับวิทยุโดยสมมติให้เป็นส่วนภาคพื้นดิน สัญญาณที่เข้ามานี้จะถูกนำไปดีมอดูเลทเพื่อให้ได้ข้อมูลดิจิตอลเดิมออกมา ต่อจากนั้นก็จะส่งเข้าเครื่องคอมพิวเตอร์เพื่อทำการแสดงผลออกทางหน้าจอ โดยสามารถแสดงผลได้ 2 แบบคือ แบบตารางข้อมูลและแบบกราฟ นอกจากนี้ยังมีฟังก์ชันการเรียกดูข้อมูลย้อนหลัง(Backward Information:Back Info)สำหรับให้ผู้ใช้ ใช้เรียกดูข้อมูลในอดีตที่ถูกเก็บไว้ในแผ่นดิสก์(disk) โดยผู้ใช้สามารถกำหนดวัน เวลาของข้อมูลที่ต้องการทราบได้เอง

ABSTRACT

This thesis presents about the telemetry simulation system an important subsystem for satellite communication. For this project we have used CPZ84C11 PLUS gard applied for central processing unit(CPU)controlling the telemetry simulation system. Initially we store 3 data with hour,day,month and year in gard's memory then send them to modulator which is Frequency Shift Keying type. It will change digital input data to analog output signal and then send it to radio transmitter for transmission radio wave in the air. At Ground Station radio receiver transmits radio wave then passes it to demodulator which detects the data from FSK signal. The data will be sent to computer and displayed on monitor. It can be displayed in two ways ,one is table type and the other is graphic type . In addition to those,there is "Backward Information" function for user to request Backward Information saved in disk.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์ปีการศึกษา 2537

ภาควิชาวิศวกรรมโทรคมนาคม

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง ระบบรายงานค่าพารามิเตอร์ของดาวเทียม

ผู้จัดทำ

1.นางสาวอิฏฐพร จันทรสถาพร 34109513

2.นายอิทธิพงษ์ วงศ์แสนสุข 34109515

3.นายอิทธิพร เมฆวิบูลย์ 34109516

---------- อาจารย์ที่ปรึกษา

(รศ.ณรงค์ เหมกรณ์)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

	หน้า
บทที่ 1 บทนำ	1
บทที่ 2 ทฤษฎีและหลักการ	2
บทที่ 3 การทดลองและผลการทดลอง	19
บทที่ 4 สรุปผลและวิจารณ์	35
บทที่ 5 บทแทรก	37
ภาคผนวก	73
กิตติกรรมประกาศ	81
เอกสารอ้างอิง	82

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่1
บทนำ

ในยุคโลกาภิวัฒน์ทุกวันนี้ ซึ่งเป็นโลกของการไร้ขอบเขต ไร้การปิดกั้นสามารถติดต่อสื่อสารกันได้ทั่วโลกภายในเวลาอันรวดเร็ว ระบบสื่อสารที่จัดว่าทันสมัยที่สุดระบบหนึ่ง ได้ถูกพัฒนาอย่างต่อเนื่องเป็นลำดับนั่นคือ การสื่อสารดาวเทียม(Sattellite communication)

ระบบการสื่อสารดาวเทียมประกอบด้วยระบบย่อยต่างๆมากมายซับซ้อน ซึ่งในแต่ละส่วนก็ได้ถูกพัฒนา แก้ไข ปรับปรุงประสิทธิภาพของระบบ ให้สูงขึ้นทุกขณะ ตลอดจนความสามารถในการส่งดาวเทียมให้โคจรอยู่รอบโลกได้เป็นระยะทางที่ไกลขึ้น ดังนั้นความสามารถในการควบคุมและตรวจสอบดูแลสถานะต่างๆของดาวเทียมหรือระบบรายงานค่าพารามิเตอร์ของดาวเทียม(Telemetry system) จึงถูกสร้างขึ้นเพื่อเป็น ระบบย่อย(Subsystem) ในการทำงานหน้าที่ดังกล่าว

ในส่วนของโครงงานที่ได้ทำ ได้นำการ์ด CP-Z84C11 PLUS มาประยุกต์ใช้เพื่อเป็นหน่วยประมวลผลกลางให้กับภาคส่งข้อมูล(บนดาวเทียม) ซึ่งจะทำการรับข้อมูล 3 ชุด สมมติให้เป็นข้อมูลจากเซ็นเซอร์ 3 ตัวที่ติดตั้งไว้ พร้อมทั้งอ่านค่าการเก็บลงไปไว้ในหน่วยความจำบนการ์ด พร้อมทั้งส่งข้อมูลนั้นไปยังภาคมอดูเลท แบบ ฟรีเควนซี่ชิพคีย์อิง ซึ่งจะเปลี่ยนสัญญาณอินพุทที่เป็นดิจิตอลได้สัญญาณเอาท์พุทเป็นสัญญาณอนาล็อก แล้วส่งผ่านเข้าไปยังเครื่องส่งวิทยุเพื่อทำการมอดูเลทกับความถี่พาหะ (Carrier frequency) ย่านคลื่นวิทยุ(Radio Wave) จากนั้นส่งสัญญาณออกทางสายอากาศ ส่วนทางด้านที่สมมติให้เป็นภาคพื้นดิน จะมีเครื่องรับวิทยุ ซึ่งจะรับสัญญาณวิทยุเข้ามา ทำการดีเทคเอาสัญญาณแอฟเอสเค(FSK Signal)ออกมา ส่งผ่านไปยังวงจรดีมอดูเลเตอร์ เพื่อแยกข้อมูล(data)ดิจิตอลซึ่งทางภาคส่งข้อมูลได้ส่งลงมา จากนั้นข้อมูลจะถูกส่งไปยังเครื่องคอมพิวเ ตอร์ เพื่อทำการแสดงผลออกทางหน้าจอ ซึ่งผู้ใช้สามารถเลือกดูได้ 2 แบบ คือ แสดงผลในรูปของตาราง และแสดงผลในรูปของกราฟ นอกจากนี้ ยังมีฟังก์ชันเรียกดูข้อมูลย้อนหลัง เป็นฟังก์ที่สร้างขึ้นสำหรับให้ผู้ใช้สามารถเรียกขอดูข้อมูลในอดีตซึ่งถูกเก็บไว้ในแผ่นดิสก์ โดยผู้ใช้เพียงแต่ ป้อนวัน เวลา ของข้อมูลที่ต้องการทราบลงไป คอมพิวเตอร์ก็จะค้นหาข้อมูลแล้วทำการแสดงข้อมูลให้ทราบออกทางหน้าจออีกอย่างหนึ่งในการเก็บข้อมูลโดยเปิดไฟล์ลงไว้ในแผ่นดิสก์ จะกระทำทุกๆ 24 ชั่วโมง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2
ทฤษฎีและหลักการ

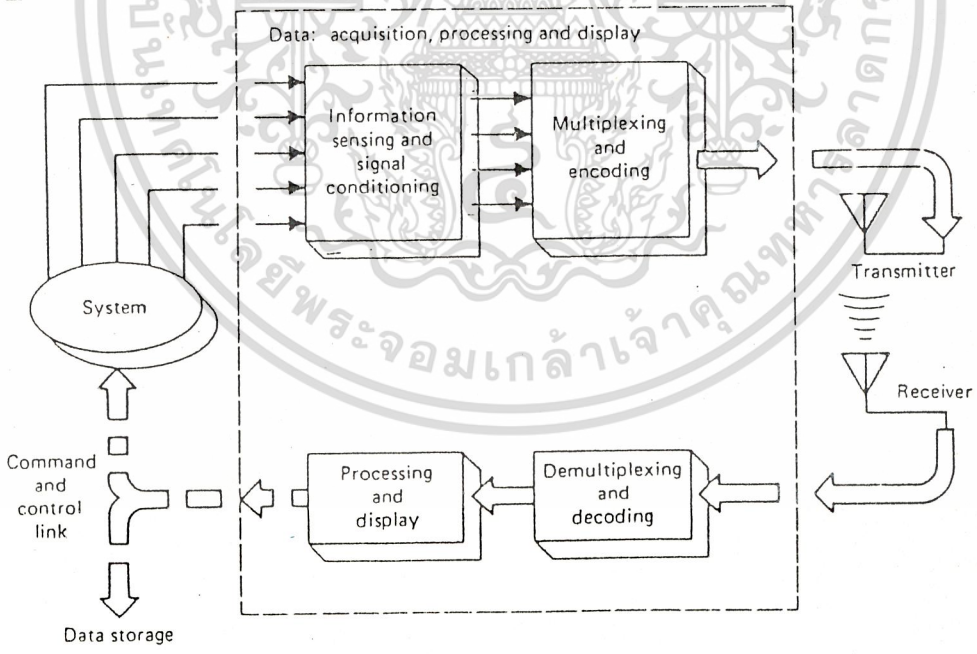
2.1 ทฤษฎีและหลักการของระบบรายงานค่าพารามิเตอร์ของดาวเทียม

2.1.1 ความหมายของระบบรายงานค่าพารามิเตอร์ของดาวเทียม

ระบบวัดรายงานค่าพารามิเตอร์ของดาวเทียม หมายถึง ขบวนการได้รับข้อมูลจากปรากฏการณ์ต่างๆโดยปราศจากการควบคุมโดยมนุษย์ ข้อมูลที่จะวัดอาจถูกบันทึกลงในเทป และถูกเปลี่ยนในเวลาที่เหมาะสม ถ้าข้อมูลถูกส่งผ่านทางคลื่นวิทยุก็จะเรียกว่า "Radio telemetry" ระบบนี้เริ่มขึ้นในระหว่างช่วงสงครามโลกครั้งที่ 2 และถูกพัฒนาเพื่อรับข้อมูลเกี่ยวกับการบินจากยานบินและจรวด เนื่องจากสิ่งที่เราต้องการจะดูระยะไกลนั้น ปกติแล้วจะวัดมากกว่า 1 สิ่งเสมอ ดังนั้นสัญญาณหลายๆ สัญญาณจึงถูกนำมามัลติเพลกซ์รวมกันซึ่งทำให้สามารถใช้เครื่องส่งและเครื่องรับเพียงเครื่องเดียวได้

2.1.2 บล็อกไดอะแกรมของระบบรายงานค่าพารามิเตอร์ของดาวเทียม

ขบวนการวัดเริ่มต้นจากระบบที่ต้องการ จะแสดงผลจากตัวเซ็นเซอร์ซึ่งจะแปลงสิ่งที่เราต้องการวัด ให้อยู่ในรูปของสัญญาณไฟฟ้า ตัวอย่างของสิ่งที่เราต้องการจะวัด เช่น อุณหภูมิ ความดัน การเคลื่อนที่หรือความเร่งของวัตถุจากในรูปที่ 2.1 จะมีเอาท์พุท จากเซ็นเซอร์ 5 ชุด



รูปที่ 2.1 แสดงบล็อกไดอะแกรมของระบบรายงานค่าพารามิเตอร์ของดาวเทียม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากบล็อคไดอะแกรมจะพบว่า ข้อมูล 5 ชุดจะถูกตรวจจับโดยเซ็นเซอร์ (บล็อก Information sensing and Signal conditioning) ต่อจากนั้นข้อมูลทั้ง 5 ผ่านไปยังบล็อกมัลติเพล็กและเข้ารหัส ซึ่งทำการมัลติเพล็กซ์ และ เข้ารหัสข้อมูลทั้ง 5 ส่งไปเข้าเครื่องส่ง เพื่อส่งข้อมูลออกไปที่เครื่องรับจะรับสัญญาณที่ส่งมา แล้วนำสัญญาณนั้นมาดีมัลติเพลกซ์และถอดรหัส แล้วส่งต่อไปที่ส่วนแสดงผล ซึ่งในส่วนสั่งงานและควบคุมนั้น จะมีไว้สำหรับส่งสัญญาณควบคุม กลับไปยังระบบที่ถูกวัดเพื่อที่จะควบคุมระบบให้อยู่ในระดับที่ต้องการ ซึ่งในระบบที่สมบูรณ์จะมีส่วนที่ใช้สำหรับเก็บข้อมูลที่วัดได้ ดังแสดงในรูปที่ 2.1 ข้างต้น

2.2 หลักการเบื้องต้นของการรับส่งข้อมูลแบบอนุกรม

สำหรับหลักการเบื้องต้นของการรับส่งข้อมูลแบบอนุกรมที่ได้กล่าวถึงนั้นสามารถที่จะสรุปเป็นข้อๆได้ดังนี้

1. ข้อมูลแบบขนานจากระบบจะถูกเปลี่ยนให้เป็นข้อมูลแบบอนุกรมเพื่อเตรียมที่จะส่งออกไปให้กับส่วนรับข้อมูล

2.ข้อมูลจะถูกส่งออกไปด้วยอัตราคงที่ค่าหนึ่งซึ่งเรียกว่า "อัตราการมอดูเลชั่น(Baud rate)"คือ ถ้าทำการส่งข้อมูลด้วยอัตรา 1200 บอร์ด ก็แสดงว่าเป็นการส่งข้อมูลด้วยอัตรา 1200 บิทต่อ 1 วินาที ซึ่งก็คือ การส่งข้อมูลโดยใช้ความถี่ 1200 เฮริตซ์ นั่นเอง

3.ข้อมูลอนุกรมจะถูกส่งออกไปทีละบิท โดยทำการส่งบิท D0 เป็นบิทแรก และบิท D7 เป็นบิทสุดท้าย

4.ในขณะที่ยังไม่มีการส่งข้อมูลเข้าไปในสายส่ง สายส่งจะถูกทำให้อยู่ในสภาวะลอจิกใดลอจิกหนึ่ง และเราเรียกสภาวะนี้ว่า "Marking"

5.อุปกรณ์ส่งข้อมูลจะเพิ่มข้อมูลอีก 1 บิทเข้าไปหน้าบิท D0 ของข้อมูลที่จะส่งให้กับเครื่องรับ บิทที่เพิ่มเข้าไปนี้เรียกว่า"สตาร์ทบิท(Start Bit)" สำหรับบิทนี้จะมีลอจิกตรงข้ามกับลอจิกของ Marking เช่น ถ้าลอจิกของ Marking เป็น"1"ลอจิกของบิทนี้ก็จะเป็น "0"

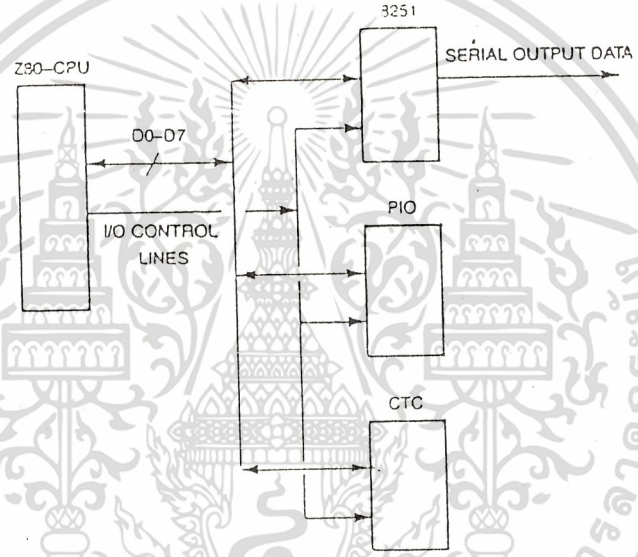
6.อุปกรณ์ส่งข้อมูลจะทำการเพิ่มบิทพาร์ริตี้เข้าไปหลังบิท D7 ของข้อมูลเพื่อใช้ในการตรวจสอบความผิดพลาดของข้อมูลที่เครื่องรับ(สำหรับบิทนี้เครื่องส่งอาจจะเพิ่มเข้าไปหรือไม่เพิ่มก็ได้ ขึ้นอยู่กับผู้ออกแบบ)

7.สำหรับบิทสุดท้ายที่ถูกเพิ่มเข้าไปหลังบิทพาร์ริตี้เรียกว่า สต็อปบิท(Stop bit) ซึ่งอาจจะมีจำนวน 1,1.5 หรือ 2บิทก็ได้ และลอจิกของบิทนี้จะเป็นลอจิกเดียวกับลอจิกของ Marking

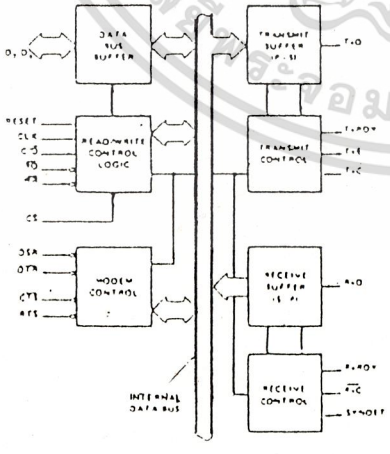
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.1 8251 USART

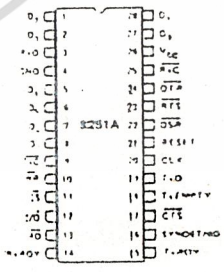
ดังที่ได้กล่าวไว้แล้วว่า การส่งข้อมูลแบบอนุกรมไปตามสายส่งได้นั้น เราจำเป็นที่จะต้องทำการเปลี่ยนรูปแบบของข้อมูลเสียก่อน แต่โดยวิธีที่กล่าวมาแล้วนั้น เป็นวิธีที่ยังมีประสิทธิภาพไม่ดีพอ เนื่องจากว่าซีพียูจะต้องทำการรับส่งข้อมูลเองในช่วงเวลาที่เหมาะสม และในส่วนรับข้อมูล ซีพียู ก็จะต้องทำการตรวจสอบความผิดพลาดของข้อมูลที่รับเข้ามาเองทุกอย่าง ทำให้เกิดความยุ่งยากในการออกแบบและในส่วนรับข้อมูลก็ยังจะต้องมีอุปกรณ์ใช้ในการเปลี่ยนข้อมูลกลับมาเป็นแบบขนานอีก ทำให้เกิดความสิ้นเปลือง ดังนั้นจึงจำเป็นที่จะต้องใช้อุปกรณ์ที่มีความสามารถที่จะเป็นได้ทั้งอุปกรณ์รับและส่งข้อมูลในตัวเดียวกัน สำหรับในบทนี้จะกล่าวถึงลักษณะและวิธีการใช้งานไอซี 8251USART(UNIVERSAL SYNCHRONOUS/ASYNCHRONOUS RECEIVER/TRANSMITTER) ซึ่งเป็นพอร์ทที่ใช้ในการรับส่งข้อมูลแบบอนุกรมที่มีประสิทธิภาพมาก



รูปที่ 2.2 แสดงการใช้งาน8251 ร่วมกับ chip support ของZ80



รูปที่ 2.3 แสดงบล็อกไดอะแกรมของ8251



รูปที่ 2.4 แสดงไอซี 8251

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 2.2 จะเป็นว่าเราสามารถที่จะนำ 8251 ไปเชื่อมต่อกับ Z80 เช่น PIO หรือ CTC และในรูปที่ 2.3 จะแสดงถึงบล็อกไดอะแกรมของ 8251 สำหรับส่วนแรกที่จะกล่าวถึงก็คือ DATA BUS BUFFER ซึ่งส่วนนี้ 8251 จะใช้ในการเชื่อมต่อระหว่าง8251กับบัสข้อมูลของ Z80 ส่วนต่อไปก็คือ อ่าน/เขียน ลอจิกควบคุม ซึ่งจะทำหน้าที่ในการควบคุมการรับส่งข้อมูลภายในของ 8251 ให้เป็นไปอย่างถูกต้อง

ส่วนที่จะกล่าวถึงต่อไปก็คือ บัฟเฟอร์การส่ง (P-S) และ ตัวควบคุมการส่ง (P-S;PARALLEL TO SERIAL CONVERSION) ซึ่งใช้ในการส่งและควบคุมการส่งข้อมูลไปตามสายส่ง

สำหรับส่วนสุดท้ายที่จะกล่าวถึงก็คือ บัฟเฟอร์ด้านรับ (S-P;SERIAL TO _PARALLEL CONVERSION) ซึ่งทำหน้าที่ในการรับและควบคุมการรับข้อมูลของ 8251

2.2.2 การจัดเรียงขาและหน้าที่

8251เป็นไอซีขนาด 28 ขา ซึ่งได้แสดงไว้ในรูปที่2.4 และเราสามารถที่จะแบ่งขาของ 8251 ออกเป็นกลุ่มๆได้ดังนี้คือ

ก.กลุ่มที่ใช้ในการติดต่อกับซีพียู

1) D0-D7 : ใช้ในการติดต่อกับ บัสข้อมูล ของ ซีพียู โดยตรงซึ่งจะทำหน้าที่ในการรับส่งข้อมูลและคำสั่งต่างๆระหว่าง 8251 กับ ซีพียู

2) รีเซ็ท : 8251 จะถูกรีเซ็ทเมื่อขานี้ได้รับลอจิก "1" ซึ่งเราอาจจะต่อมาจากขา รีเซ็ท ของ Z80โดยผ่าน INVERTER ก่อนก็ได้

3) CLK(CLOCK) : ใช้ในการควบคุมช่วงเวลาการทำงานภายในของ _8251 สำหรับการใช้งานนั้นจะต่อเข้าโดยตรงกับระบบ อย่างไรก็ตามสัญญาณที่ขา CLK นี้ไม่เกี่ยวข้องกับ อัตราการรับส่งข้อมูลหรือ BAUD RATE แต่อย่างใด

4) RD : เมื่อขานี้ได้รับลอจิก "0" 8251 จะทำการส่งข้อมูล_แบบขนานออกมาที่ บัสข้อมูล เพื่อส่งให้กับ ซีพียู

5) WR : เมื่อขานี้ได้รับลอจิก "0" 8251 จะทำการรับข้อมูลแบบขนานจาก DATA BUS ของระบบ

6) C/D (CONTROL/DATA):ขา C/D นี้จะใช้ในการทำให้ 8251ทราบว่า ซีพียู ต้องการที่จะติดต่อกับ รีจิสเตอร์ ควบคุม หรือ รีจิสเตอร์ข้อมูล

7) CS (CHIP SELECT): ในกรณีที่ขานี้ได้รับลอจิก"0" ก็จะเป็นการ ทำให้ 8251ทำงาน โดยทั่วไปแล้วสัญญาณ ที่ขานี้จะได้มาจากการถอดรหัสพอร์ทแอดเดรส ดังที่ใช้กับ CHIP SUPPORT อื่นๆ

ข.กลุ่มที่ใช้ในการส่งข้อมูล

1) TxD (TRANSMIT DATA OUTPUT) : เป็นขาที่ใช้ในการส่งข้อมูลไปตามสายส่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2) TxC (TRANSMIT BAUD RATE CLOCK) : ขานี้เป็นขา ที่ใช้ในการส่งสัญญาณคล็อกที่ใช้ในการส่งข้อมูลซึ่งก็คือความถี่ที่ใช้ในการกำหนด อัตราการมีอดดูเลชันนั่นเอง โดยปกติแล้ว จะต้องช้ากว่า สัญญาณคล็อกของระบบไม่น้อยกว่า 30 เท่า

3) TxRDY : ขานี้จะใช้ในการทำให้ซีพียูทราบว่า 8251 พร้อมที่จะรับข้อมูลจากซีพียูเพื่อที่จะทำการส่งต่อไปแล้วหรือยัง และขานี้อาจจะนำไปใช้ในการขออินเทอร์รัพท์ก็ได้

4) TxEMPTY :ขานี้จะใช้ในการแสดงว่าข้อมูลที่ ซีพียู ส่งให้กับ8251 นั้นได้ถูกส่งออกไปให้อุปกรณ์อื่นๆหมดแล้ว โดยที่ 8251 จะทำให้ขานี้เป็น "1" และเมื่อ ซีพียู ทำการส่งข้อมูลชุดต่อไปให้กับ 8251 ขา TxEMPTY ก็จะเป็น "0" จนกว่า 8251 จะทำการส่งข้อมูลนี้ออกไปหมด 8251 ก็จะทำให้ขานี้กลับเป็น "0" อีกครั้ง

ค.กลุ่มที่ใช้ในการรับข้อมูล

1) RXD : ใช้ในการรับข้อมูลแบบอนุกรมจากสายส่ง

2) RxC : เป็นขาที่ใช้ในการรับสัญญาณคลอ็กที่ใช้ในการรับข้อมูลโดยปกติแล้วจะทำการต่อเข้ากับ TxC โดยตรง

3) RxRDY :จะใช้ในการแสดงว่า8251พร้อมที่จะส่งข้อมูลให้กับซีพียูและขานี้อาจจะใช้ในการขออินเทอร์รัพท์ ได้เช่นเดียวกับขา TxRDY

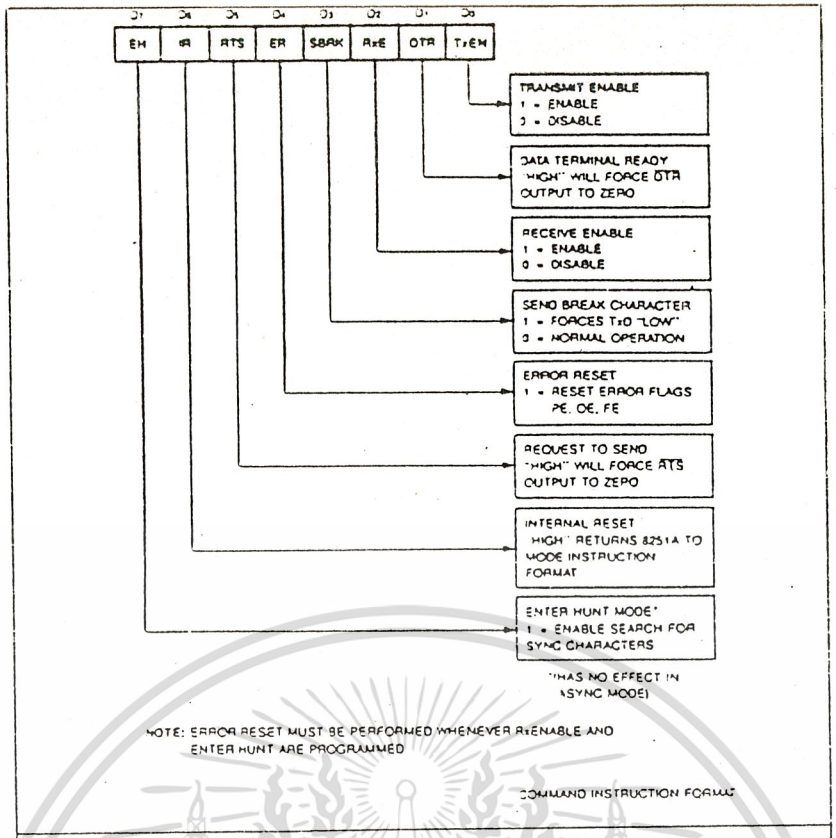
4) SYNDET : ขานี้จะใช้ในการรับข้อมูลแบบซิงโครนัสเท่านั้น (8251 สามารถที่จะทำการรับส่งข้อมูลได้ทั้งแบบ ซิงโครนัส และแบบ อซิงโครนัสซึ่งจะได้กล่าวถึงต่อไป) โดยที่เราสามารถที่จะโปรแกรมให้ขานี้เป็นอินพุทหรือเอาท์พุทก็ได้ โดยที่เมื่อขา SYNDET นี้ถูกโปรแกรมเป็นเอาท์พุทนั้นขา SYNDET จะให้ลอจิก "1" เมื่อซีพียูทำการอ่าน รีจิสเตอร์สถานะสำหรับขา SYNDET นี้จะให้ลอจิก "1" ไปอีกกรณีหนึ่งคือ เมื่อ 8251 ได้รับข้อมูลจากสายส่งเป็น "0" หมดตั้งแต่ สตาร์ทบิท จนถึง สต็อปบิท

ในกรณีที่ขาSYNDETถูกโปรแกรมให้เป็นอินพุทนั้น ถ้าขานี้ได้รับสัญญาณขอบขาขึ้น (สัญญาณเปลี่ยนจาก ลอจิก "0" เป็น "1") 8251 ก็จะถือว่าข้อมูลที่ขา RxD เป็นข้อมูลทันที และเราสามารถที่จะทำให้ลอจิกที่ขานี้กลับเป็น "0" ได้ในสัญญาณ RXC ลูกต่อไป

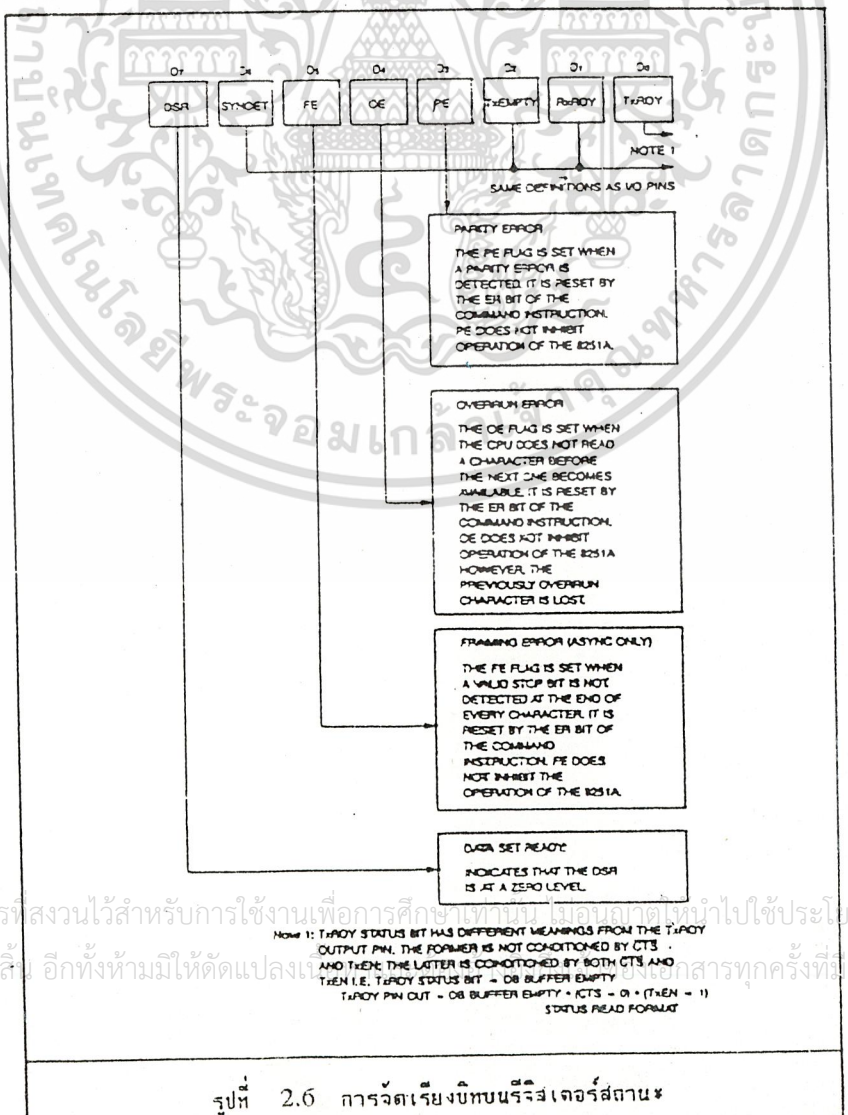
ง.กลุ่มไฟเลี้ยงของ 8251

8251 ใช้ไฟเลี้ยงเพียงชุดเดียว คือ + 5V กับ GND เท่านั้น ดังนั้นขาไฟเลี้ยงของ 8251 จึงมีเพียง 2 ขา คือ v_{cc}กับ GND

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

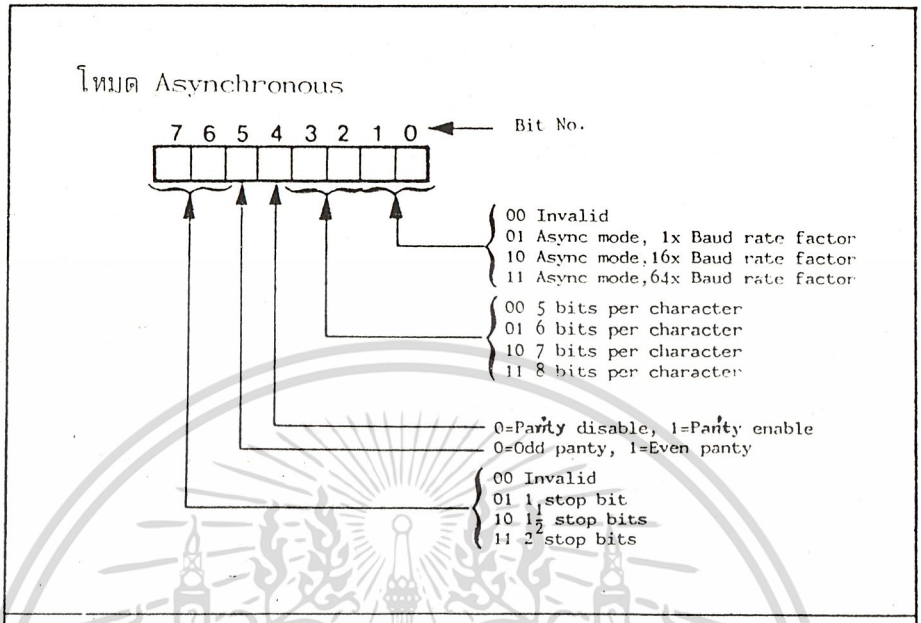


รูปที่ 2.5 การจัดเรียงบิทใน COMMAND WORD



รูปที่ 2.6 การจัดเรียงบิทของรีจิสเตอร์สถานะ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.7 การจัดเรียงบิทใน MODE WORD

2.3 เครื่องรับส่งวิทยุสื่อสาร

เครื่องรับวิทยุสื่อสารที่จะอธิบายถึงต่อไปนี้ เป็นเครื่องรับวิทยุสื่อสารที่ใช้งานในย่านความถี่ HF (ประมาณ 3 ถึง 30 เมกะเฮิรตซ์) ย่านความถี่ VHF (ประมาณ 150 เมกะเฮิรตซ์และย่านความถี่ UHF (ประมาณ 450 เมกะเฮิรตซ์)ซึ่งผู้ใช้งานการสื่อสารในย่านความถี่ที่กล่าวมาข้างต้น มีอยู่เป็นจำนวนมาก

ปกติเครื่องรับวิทยุสื่อสารดังกล่าว มักจะประกอบภาคเครื่องรับและภาคเครื่องส่ง รวมไว้ในตัวเครื่องเดียวกัน เราจึงเรียกว่า เครื่องรับส่งวิทยุ(transceiver) ซึ่งมีทั้งประเภทติดตั้งประจำที่ติดตั้งประจำที่ ติดตั้งในรถยนต์หรือในเรือ และประเภทหิ้วถือไปมาหรือมือถือซึ่งนิยมเรียกว่า วอล์คกี้ทอล์คกี้ (walkie-talkie)ข่าวสารที่ใช้ในการรับส่งกันมักจะเป็นเสียงพูด

เครื่องรับส่งวิทยุในย่าน HF ส่วนใหญ่ออกแบบให้ใช้รับส่งสัญญาณในระบบ SSB และ CW (อาจมี AM ด้วย) บางเครื่องสามารถต่ออแดปเตอร์(adapter)เพื่อใช้รับส่งสัญญาณวิทยุโทรพิมพ์(radio-teletype หรือ RTTY) หรือสัญญาณโทรทัศน์สแกนช้าด้วย การสื่อสารในย่านความถี่ HFนี้ จะเป็นการสื่อสารระยะไกลและเป็นการเชื่อมโยงระหว่างจุดต่อจุด(point to point)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

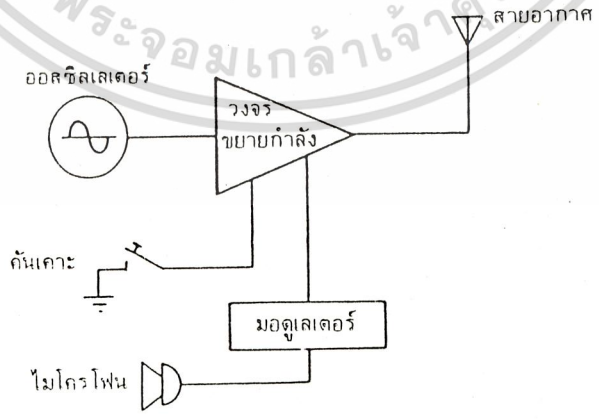
สำหรับเครื่องรับส่งวิทยุในย่าน VHF และ UHF นี้ส่วนใหญ่จะเป็นแบบแบนด์วิดท์แคบ (narrow band) ซึ่งใช้ในการรับส่งสัญญาณ FM (อาจมี AM, SSB หรือ สัญญาณอื่นๆด้วย) วิธีมอดูเลตแบบ FM มีข้อดีตรงที่มีภูมิต้านทานต่อนอยส์(noise immunity)ดี และสัญญาณFM ผลิตได้ง่าย ทำให้ขนาดเครื่องเล็กกะทัดรัดและน้ำหนักเบา การสื่อสารในย่านความถี่ VHF และ UHF มีทั้งแบบเชื่อมโยงระหว่างจุดต่อจุดและการสื่อสารโมบายล์(mobile) ระยะทางที่ติดต่อสื่อสารกันมักจะไม่เกิน 50 กิโลเมตร

ความถี่ใช้งานของเครื่องรับส่งวิทยุข้างต้นมักจัดไว้เป็นช่องๆตายตัว ถ้าต้องการเปลี่ยนความถี่ก็ใช้วิธีปรับสวิตช์เปลี่ยนช่องโดยที่แต่ละช่องจะมีแร่บังคับความถี่คริสตอลเฉพาะของตัวเอง สำหรับเครื่องรับส่งวิทยุในรุ่นใหม่ๆ จะนิยมใช้ระบบสังเคราะห์ความถี่แทนการใช้แร่ ทำให้สามารถเลือกหรือปลี่ยนความถี่ใช้งานได้สะดวก ส่วนเครื่องรับส่งวิทยุในกิจการวิทยุสมัครเล่น(amature reaio)นั้น จะนิยมใช้ระบบสังเคราะห์ความถี่(synthesizer) ทำให้สามารถปรับจูนความถี่ได้ตามต้องการ ตลอดย่านความถี่

2.3.1 เครื่องรับส่งวิทยุ SSB

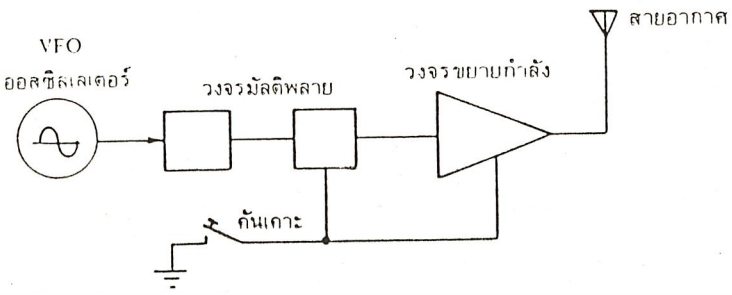
เครื่องรับส่งวิทยุในสมัยก่อนจะใช้งานกันแต่เฉพาะในระบบCWและระบบ AM เท่านั้น ภาคเครื่องส่งจะเป็นแบบง่ายๆกล่าวคือใช้ออสซิลเลเตอร์ผลิตความถี่ออกมาแล้วขยายส่งออกอากาศโดยเคาะรหัสมอร์ส(Morse หรือ CW หรืออาจใช้ออสซิลเลเตอร์ VFO(variable frequendy oscillator)ป้อนให้วงจรคูณความถี่ (มัลติพลาย) เพื่อขยายออกอากาศต่อไป ถ้าหากมีการมอดูเลต ก็จะทำที่ภาคสุดท้าย ซึ่งสัญญาณในภาคนี้มีขนาดใหญ่เพราะต้องใช้กำลังมากในการมอดูเลต(ดังรูปที่ 2.8)

เครื่องส่งอีกชนิดหนึ่งซึ่งได้รับการปรับปรุงใหม่ โดยใช้งานในย่านความถี่ 1.65 เมกะเฮิรตซ์แล้วคูณความถี่ให้สูงขึ้นไปเป็นความถี่ที่ต้องการ(ดังรูปที่ 2.9) เครื่องส่งทั้งสองแบบที่กล่าวมาข้างต้นใช้ประโยชน์ก็แต่เฉพาะการส่งสัญญาณ CW หรือ AM เท่านั้น อย่างไรก็ดีการมอดูเลตที่ขนาดสัญญาณขนาดใหญ่ๆนั้นไม่ค่อยสะดวกเพราะต้องใช้กำลังมาก ยิ่งถ้าเป็นสัญญาณ SSB ด้วยแล้วการกำจัดพาหะออกไปจะทำได้ลำบากยิ่งขึ้น



รูปที่ 2.8 เครื่องส่ง CW อย่างง่าย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
034948



รูปที่ 2.9 เครื่องส่ง CW แบบใช้วงจรมัลติพลาย

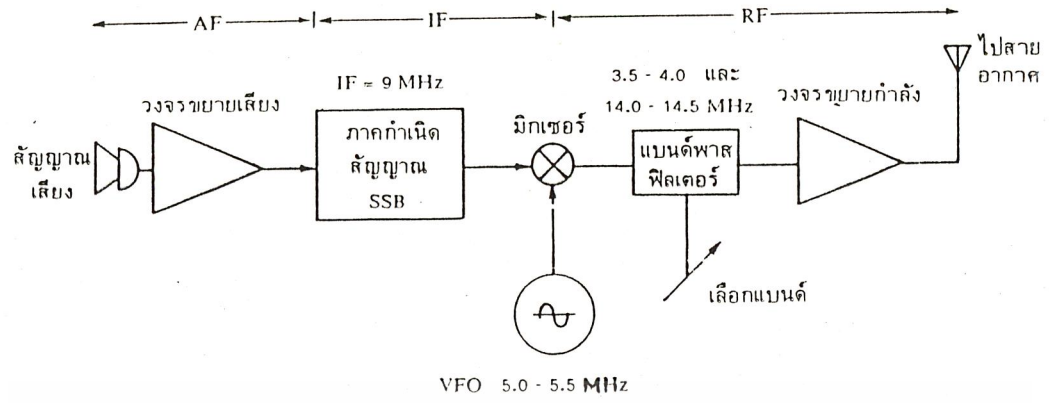
ก. ภาคเครื่องส่ง

ในปัจจุบันเครื่องส่งทั้งสองชนิดที่อธิบายในตอนต้นได้ค่อยๆลดความนิยมลงไปบ้าง เนื่องจากส่วนใหญ่เราห้นมาใช้ระบบ SSB ซึ่งมีประสิทธิภาพในการส่งดีกว่า

สัญญาณ SSB กำเนิดได้หลายวิธี ดังที่ได้อธิบายมาแล้ว เอาต์พุตจากภาคกำเนิดไซด์แบนด์(side-band generator) จะมีความถี่ซึ่งถือว่าเป็นความถี่ IF ของภาคเครื่องส่ง การที่จะแปลงความถี่ของสัญญาณSSB ให้มีค่าตามความถี่ใช้งานที่ต้องการนั้น เราไม่สามารถใช้วิธีคูณความถี่ได้ เพราะสัญญาณ SSB จะมีความเพี้ยน การแปลงความถี่ของสัญญาณ SSB สามารถทำได้โดยวิธีผสมคลื่น(mix)หรือเฮตเทอโรดายน์

รูปที่ 2.10 แสดงดังตัวอย่างแผนผังของเครื่องส่งแบบซิงเกิลคอนเวอร์ชั่น จะเห็นว่าเอาต์พุตของภาคกำเนิดสัญญาณจะทำการผสมคลื่นกับ VFO แล้วก็ผ่านฟิลเตอร์ชนิดแบนด์พาสเพื่อกรองความถี่ที่ไม่ต้องการทิ้งเอาแต่เฉพาะสัญญาณที่ต้องการขยายส่งออกอากาศต่อไป สัญญาณ SSB ที่ผ่านกระบวนการกรอง และขยายต่างๆ ที่กล่าวมานั้นจะต้องไม่ทำให้สัญญาณ SSB มีการเพี้ยนเกิดขึ้น ฉะนั้นเราต้องให้ภาคขยายและภาคมิกเซอร์ทำงานในช่วงที่เป็นลิเนียร์(จะใช้วงจรมัลติพลายหรือวงจรขยายคลาส C เช่นเดียวกับการขยายสัญญาณ FM ไม่ได้) เพราะวงจรประเภทนอนลิเนียร์จะทำให้สัญญาณ SSB เกิดการเพี้ยนขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.10 เครื่องส่ง SSB แบบซิงเกิลคอนเวอร์ชั่น

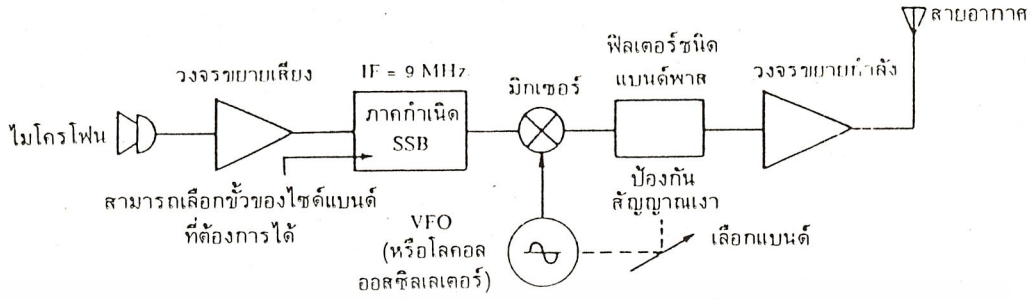
สังเกตว่าเครื่องส่งในรูปที่ 2.10 นั้น เราใช้สัญญาณทั้งความถี่ผลรวมและความถี่ผลต่างโดยการเลือกแบนด์ความถี่ที่ต้องการ และตัดความถี่ที่ไม่ต้องการออกไป โดยฟิลเตอร์(BPF) ที่ใช้ในที่นี้ทำหน้าที่กำจัดสัญญาณความถี่ที่ไม่ต้องการหรือสัญญาณเงาออกไป ความถี่ใช้งานของเครื่องส่งดังกล่าวจำกัดอยู่ในช่วง3.5 ถึง4.0 เมกะเฮิรตซ์ และ 14.0 ถึง 14.5 เมกะเฮิรตซ์ ถ้าต้องการให้ครอบคลุมความถี่กว้างขึ้นกว่านี้เราต้องปรับปรุงการผสมคลื่นใหม่และ ใช้วิธีสลับขั้วของไซด์แบนด์ร่วมด้วย (ดูรูปที่ 2.11)

ปกติภาคกำเนิดไซด์แบนด์มักจะใช้พาหะIF ที่ความถี่สูงๆเนื่องจากถ้าใช้ความถี่ต่ำจะกำจัดสัญญาณเงาได้ลำบาก เช่น ถ้าใช้ความถี่ IF เท่ากับ 455กิโลเฮิรตซ์ สัญญาณเงาจะห่างออกไป 910 กิโลเฮิรตซ์ที่ความถี่ต่ำๆ เช่น IF 3.5 หรือ 7 เมกะเฮิรตซ์ การกำจัดสัญญาณเงาสามารถทำได้ อย่างไรก็ดีที่ความถี่สูงขึ้น เช่น ที่ความถี่ IF 21 หรือ 28 เมกะเฮิรตซ์ การกำจัดสัญญาณเงาทำได้ลำบากขึ้น (เพราะต้องใช้วงจรฟิลเตอร์ที่มีค่า Q สูงมากขึ้นด้วย)

ในกรณีที่กำเนิดไซด์แบนด์ที่มีความถี่สูง เช่น ที่ความถี่ IF 9 เมกะเฮิรตซ์ เราอาจเลือกใช้ความถี่ VFO หรือโลคอลออสซิลเลเตอร์ได้ทั้งแบบป้อนด้านสูงอย่างเดียว หรือ แบบป้อนด้านสูงและด้านต่ำผสมกันยกตัวอย่างเช่น เครื่องส่งในย่านความถี่ 3.5 ถึง 30 เมกะเฮิรตซ์ ถ้าใช้วิธีการป้อนด้านสูงจะต้องใช้ความถี่โลคอลออสซิลเลเตอร์ระหว่าง 12.5 ถึง 39เมกะเฮิรตซ์ ถ้าใช้วิธีการป้อนด้านสูงและด้านต่ำผสมกันจะต้องใช้ความถี่VFOระหว่าง 5 ถึง 20.7เมกะเฮิรตซ์ ดังตารางที่ 2.12

สมมติว่าเราต้องการส่งความถี่ 3.5 เมกะเฮิรตซ์ USB เราจะต้องผลิตไซด์แบนด์ 9 เมกะเฮิรตซ์ LSB และบีตกับโลคอลออสซิลเลเตอร์ 12.5 เมกะเฮิรตซ์(ด้านสูง) เป็น 3.5 เมกะเฮิรตซ์ USB ตามต้องการ ถ้าต้องการส่ง 21 เมกะเฮิรตซ์ USB เราจะต้องผลิตไซด์แบนด์ 9 เมกะเฮิรตซ์ USB เพื่อบีตกับ VFO 12 เมกะเฮิรตซ์(ด้านต่ำ) จะได้ 21 เมกะเฮิรตซ์ USB ตามต้องการ สังเกตว่าอาจมีการสลับขั้วของไซด์แบนด์เมื่อเปลี่ยนย่านความถี่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.11 เครื่องส่ง SSB แบบซิงเกิลคอนเวอร์ชั่นที่ครอบคลุมความถี่กว้างขึ้น

ความถี่ใช้งาน	การป้อนความถี่ในการบีต	ความถี่ของ VFO
3.5 - 4.0 MHz	ป้อนด้านสูง	12.5 - 13.0 MHz
7.0 - 7.5 MHz	ป้อนด้านสูง	16.0 - 16.5 MHz
10.0 - 10.5 MHz	ป้อนด้านสูง	19.0 - 19.5 MHz
14.0 - 14.5 MHz	ป้อนด้านต่ำ	5.0 - 5.5 MHz
18.05 - 18.55 MHz	ป้อนด้านต่ำ	9.05 - 9.55 MHz
21.0 - 21.5 MHz	ป้อนด้านต่ำ	12.0 - 12.55 MHz
24.5 - 25.0 MHz	ป้อนด้านต่ำ	15.5 - 16.0 MHz
28.0 - 28.5 MHz	ป้อนด้านต่ำ	19.0 - 19.5 MHz
28.5 - 29.0 MHz	ป้อนด้านต่ำ	19.5 - 20.0 MHz
29.0 - 29.5 MHz	ป้อนด้านต่ำ	20.0 - 20.5 MHz
29.5 - 29.7 MHz	ป้อนด้านต่ำ	20.5 - 20.7 MHz

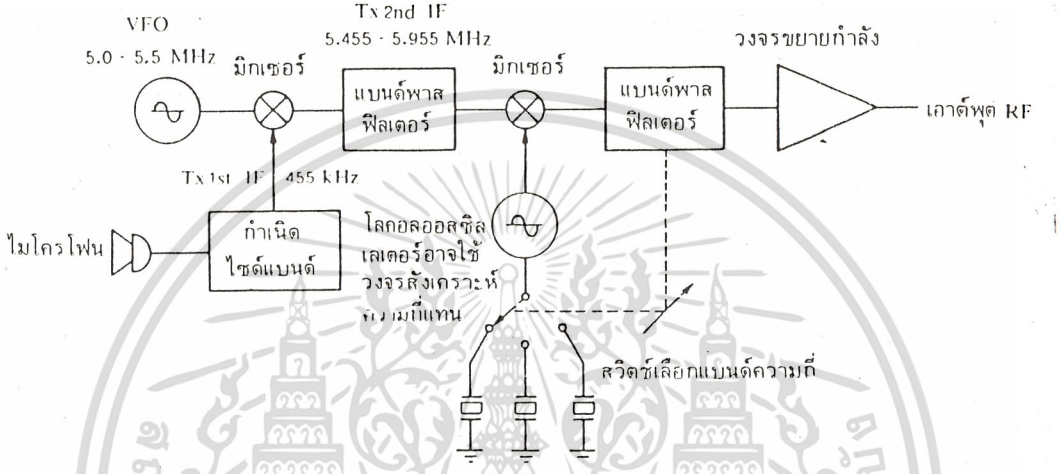
รูปที่ 2.12 ตารางความถี่ของออสซิลเลเตอร์ที่ต้องใช้ในย่านความถี่ใช้งานต่างๆ

เครื่องส่ง SSB อีกชนิดหนึ่งเป็นชนิดดับเบิลคอนเวอร์ชั่น ซึ่งภาคกำเนิดไซด์แบนด์ผลิตไซด์แบนด์ที่ความถี่ 455 กิโลเฮิรตซ์ แล้วผสม 2 ครั้ง คือ ครั้งแรกผสมกับ 5.0 ถึง 5.5 เมกะเฮิรตซ์ แล้วผ่านฟิลเตอร์ชนิดแบนด์พาส 5.455 ถึง 5.955 เมกะเฮิรตซ์ และครั้งที่สองผสมกับโลคอลออสซิลเลเตอร์แล้วไปฟิลเตอร์ชนิดแบนด์พาสและวงจรขยายกำลังต่อไป ช่วงปรับความถี่ของ VFO กว้างเพียง 0.5 เมกะเฮิรตซ์ ซึ่งก็คือในช่วงความถี่ระหว่าง 5.0 ถึง 5.5 เมกะเฮิรตซ์ ฉะนั้นถ้าต้องการให้เครื่องส่งทำงานตลอดย่านความถี่ 3.5 ถึง 30 เมกะเฮิรตซ์ จะต้องใช้แร่เพื่อเปลี่ยนย่านความถี่ถึง 50 ก้อน

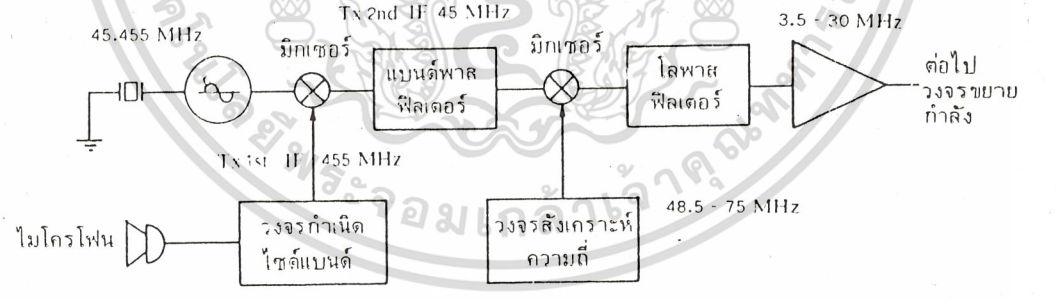
ในรูปที่ 2.14 แสดงเครื่องส่งแบบอัพคอนเวอร์ชั่น(upconversion) ซึ่งความถี่ของโลคอลออสซิลเลเตอร์สูงกว่าความถี่ใช้งานสูงสุดของเครื่องส่ง ดังนั้น IF จึงต้องสูงกว่าความถี่ใช้งานสูงสุดด้วย วิธี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การกำเนิดไซด์แบนด์ที่ความถี่ใช้งานสูงสุดของเครื่องส่ง ดังนั้น IF จึงต้องสูงกว่าความถี่ใช้งานสูงสุด ด้วยวิธีการกำเนิดไซด์แบนด์ที่ความถี่สูงเช่นนี้จะต้องใช้วิธีเฮตเทอร์โรดายน์ จะเห็นว่าสัญญาณ SSB กำเนิดที่ความถี่ 455 กิโลเฮิรตซ์ แล้วมิกซ์กับคริสตอลออสซิลเลเตอร์ 45.455 เมกะเฮิรตซ์ ได้สัญญาณ SSB IF เท่ากับ 45 เมกะเฮิรตซ์ ผ่านฟิลเตอร์ชนิดแบนด์พาส 45 เมกะเฮิรตซ์ ซึ่งใช้คริสตอลฟิลเตอร์กำจัดสัญญาณเงา(ที่อยู่ห่างไป 910 กิโลเฮิรตซ์) อย่างไรก็ตามถ้าเราเลือกกำเนิดสัญญาณ SSBที่สูงกว่า 455 กิโลเฮิรตซ์ ขึ้นไปอีกการกรองสัญญาณเงาจะทำได้สะดวกขึ้น



รูปที่ 2.13 เครื่องส่ง SSB แบบดับเบิลคอนเวอร์ชั่น



รูปที่ 2.14 เครื่องส่ง SSB แบบอัพคอนเวอร์ชั่น

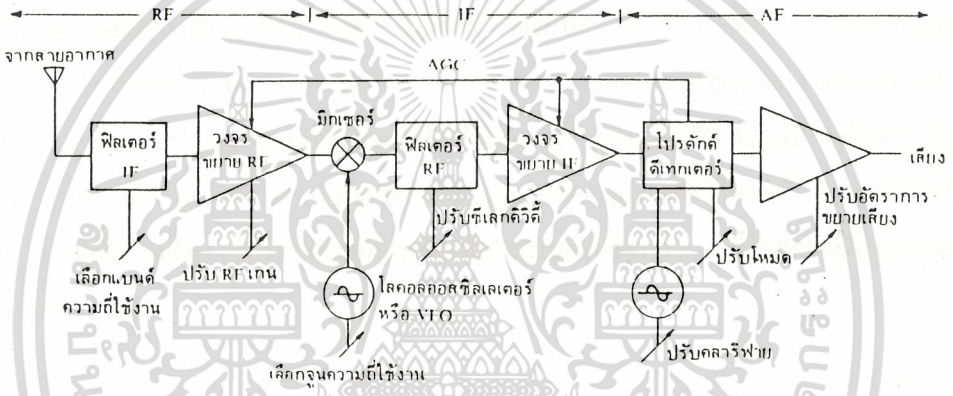
เมื่อสัญญาณ SSB IF (45 เมกะเฮิรตซ์) ผสมกับโลคอลออสซิลเลเตอร์จากซินธิไซเซอร์ ในช่วงของความถี่ระหว่าง 48.5 ถึง 75 เมกะเฮิรตซ์จะให้เอาต์พุทในย่านความถี่ 3.5 ถึง 30 เมกะเฮิรตซ์ตามต้องการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

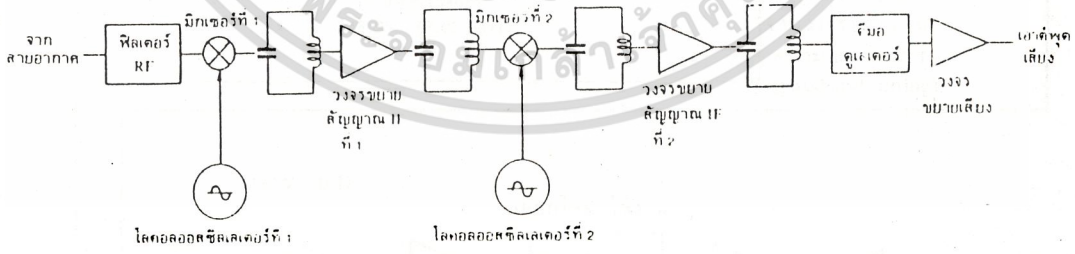
ข.ภาคเครื่องรับ

เครื่องรับในยุคแรกๆ จะเป็นแบบซิงเกิลคอนเวอร์ชั่น ซึ่งมีแผนผังดังแสดงในรูปที่ 2.15 สัญญาณ IF ผ่านเข้าทางสายอากาศขยายและแปลงความถี่เป็น IF ซึ่งอัตราขยายของภาค IF นี้ค่อนข้างสูงแล้วจึงดีมอดปี้อนไปยังวงจรเสียงต่อไป การกำจัดสัญญาณเงาทำได้โดยใช้ฟิลเตอร์ก่อนมิกเซอร์ สัญญาณเงาต้องถูกลดทอนลงไปจนเหลือเล็กมากๆ ทำให้เครื่องรับไม่สามารถรับได้ ปกติอัตราลดทอนจะมีค่าประมาณ 80 ถึง 90 เดซิเบล สังเกตว่าที่ความถี่ใช้งานสูงขึ้น การกรองความถี่เงาจะยุ่งยากขึ้น เว้นแต่เราจะเพิ่มความถี่ IF ให้สูงๆ เข้าไว้

การแก้ปัญหาสปิวเรียสทำได้โดยใช้วิธีดับเบิลคอนเวอร์ชั่น วิธีนี้เราใช้ความถี่ IF ที่หนึ่งซึ่งมีค่าสูงเพื่อกำจัดปัญหาสัญญาณเงาและใช้ความถี่ IF ที่สองมีค่าต่ำรองลงมา ในรูปที่ 2.16มิกเซอร์ที่สองก็ยังมี



รูปที่ 2.15 แผนผังเครื่องรับ SSB แบบซิงเกิลคอนเวอร์ชั่นแสดงปุ่มควบคุมสำหรับภาคต่างๆด้วย

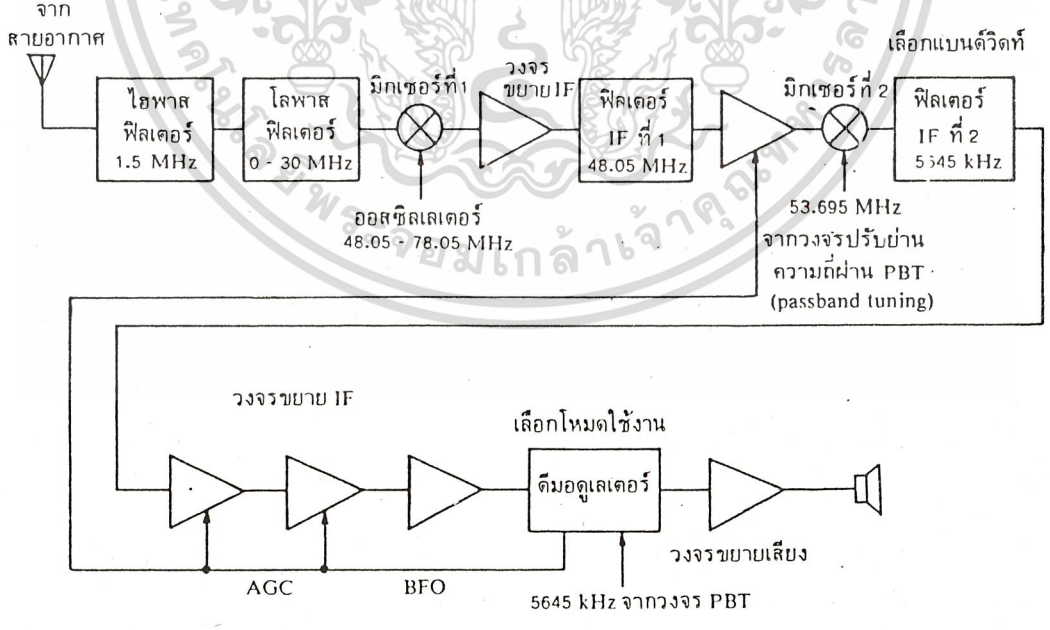


รูปที่ 2.16 แผนผังเครื่องรับ SSB แบบดับเบิลคอนเวอร์ชั่น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สัญญาณเงาเช่นกันแต่ใช้วงจรจูนที่มีค่า Q สูงก็จะสามารถกำจัดไปได้ สำหรับโลคอลออสซิลเลเตอร์ที่หนึ่งอาจเป็นแบบใช้แร่เลือกแบนด์ และโลคอลออสซิลเลเตอร์ที่สองใช้วิธีจูนความถี่ใช้งาน ส่วนในระบบสังเคราะห์ความถี่โลคอลออสซิลเลเตอร์ที่หนึ่ง จะใช้ระบบสังเคราะห์ความถี่ ส่วนโลคอลออสซิลเลเตอร์ที่สองใช้ VFO

เครื่องรับอีกประเภทหนึ่งซึ่งได้รับความนิยมคือ เครื่องรับชนิดอัพคอนเวอร์ชั่น ซึ่งมีแผนผังดังรูปที่ 2.17 สัญญาณ RF จะถูกเฮตเทอโรดายน์แปลงเป็นความถี่IF ที่หนึ่งซึ่งจะมีวามถี่สูงกว่าความถี่อินพุตสูงสุด นั่นคือไม่ว่าความถี่อินพุตจะมีค่าเท่าใดจะถูกเฮตเทอโรดายน์เป็น IF ที่มีความถี่สูงขึ้นเสมอ ในกรณีของเครื่องรับซึ่งใช้งานในย่านความถี่คลุมไปถึง 30 เมกะเฮิรตซ์ ความถี่ IF จะต้องสูงกว่า 30 เมกะเฮิรตซ์ ดังนั้นออสซิลเลเตอร์ที่หนึ่งจะต้องมีความถี่ในย่าน 30 ถึง 60 เมกะเฮิรตซ์เป็นอย่างต่ำ เครื่องรับโดยทั่วไปมักเลือกใช้ความถี่ IF ที่หนึ่งเท่ากับ 45 เมกะเฮิรตซ์ ดังนั้นย่านความถี่โลคอลออสซิลเลเตอร์จะต้องอยู่ระหว่าง 45 ถึง 75 เมกะเฮิรตซ์ เนื่องจากที่ความถี่ 45 เมกะเฮิรตซ์ เราออกแบบให้มีซีเลกติวิตี้ดีได้ยากเราจึงต้องทอนความถี่ให้ต่ำลงเป็นความถี่ IF ที่สอง โดยทั่วไปจะเลือกใช้ความถี่ 455 กิโลเฮิรตซ์ สำหรับในกรณีที่ IF เท่ากับ 45 เมกะเฮิรตซ์ สัญญาณเงาจะอยู่ห่างความถี่ใช้งานเท่ากับ 90 เมกะเฮิรตซ์ ย่านความถี่เงาของเครื่องรับจะอยู่ในย่าน 90 ถึง 120 เมกะเฮิรตซ์ ถ้าความถี่ใช้งานอยู่ระหว่าง 0 ถึง 30 เมกะเฮิรตซ์ เราจะสามารถ กำจัดสัญญาณเงาโดยใช้ฟิลเตอร์ชนิดโลพาส เครื่องรับประเภทนี้จึงไม่ต้องใช้วงจรพิเศษสำหรับเลือกแบนด์ที่เรียกว่า พรีซีเลกเตอร์(preselector) เพื่อกำจัดสัญญาณเงา



<u>รูปที่ 2.17 แผนผังเครื่องรับ SSB แบบอัพคอนเวอร์ชั่น</u>

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

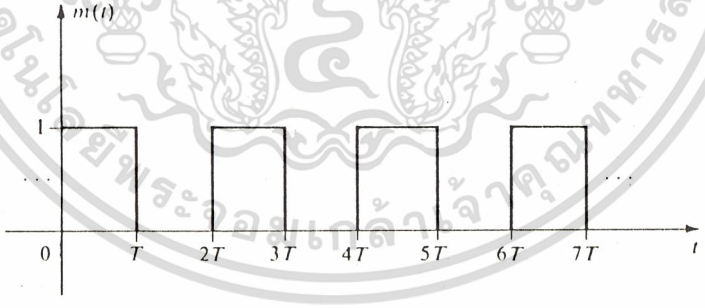
สำหรับการจูนเราอาจใช้วิธีจูนโลคอลออสซิลเลเตอร์ทั้งสองก็ได้ ถ้าเราใช้วิธีเลือกแบนด์โดยปรับโลคอลออสซิลเลเตอร์ที่สอง ภาค IF ที่หนึ่งจะต้องใช้ฟิลเตอร์ที่เป็นแบนด์กว้าง ทำให้กำจัดสัญญาณเงาได้ลำบาก โดยทั่วไปเราไม่นิยมจูนที่โลคอลออสซิลเลเตอร์ที่สอง แต่ใช้จูนที่โลคอลออสซิลเลเตอร์ที่หนึ่งที่เดียว และใช้การสังเคราะห์ความถี่เข้าช่วย

2.4 ฟรีเควนซีชิฟคีย์อิง

เมื่อมีการมอดูเลชันความถี่(frepuency modulation) ในการส่งข่าวสารดิจิตอล โดยเฉพาะอย่างยิ่งในรูปของสัญญาณไบนารี จะเรียกเทคนิคในการมอดูเลทนี้ว่า ฟรีเควนซีชิพคีย์อิง(frequency Shift Keying:FSK)สำหรับสัญญาณไบนารีในรูปสัญญาณเอฟเอสเค จะแทนลอจิก "1"ด้วย พัลส์รูปไซน์ที่มีความถี่ค่าหนึ่ง และ แทนลอจิก"0"ด้วย พัลส์รูปไซน์ที่มีความถี่อีกค่าหนึ่ง รูปร่างของพัลส์(Fulse Shape)ขึ้นอยู่กับการประยุกต์ใช้แต่ถ้าเราพิจารณาในกรณีที่เป็นพัลส์สี่เหลี่ยม(Rectangular Pulse)ที่มีความกว้างเท่ากับ T จะได้สัญญาณที่ถูกส่ง S(t) โดยที่

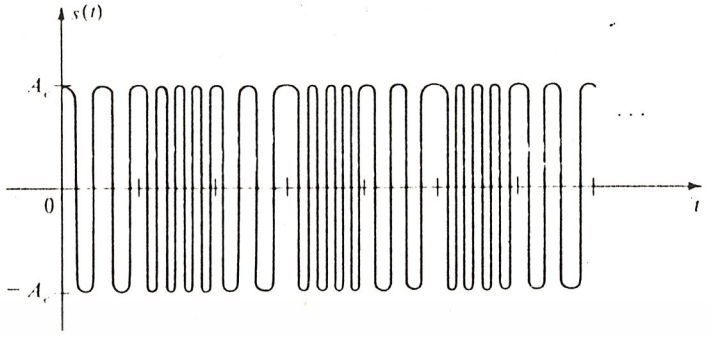
$S(t) = A_0(\cos w_1 t)$ สำหรับไบนารี"1"

$= A_0(\cos w_2 t)$ สำหรับไบนารี"0"

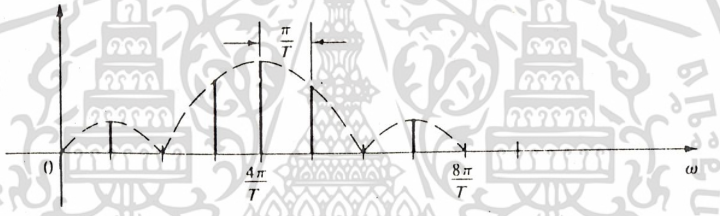


รูปที่ 2.18 แสดงสัญญาณข่าวสารไบนารี

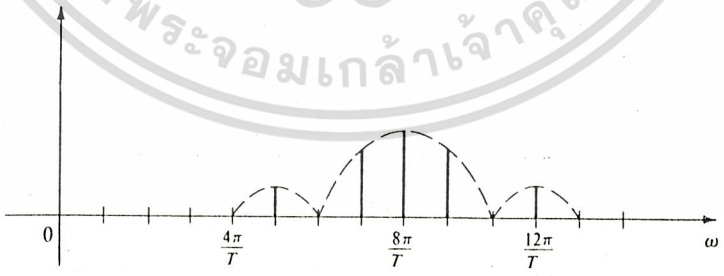
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.19 แสดงเอฟเอสเคเวฟฟอร์มของ m(t) ของสัญญาณข่าวสารไบนารี

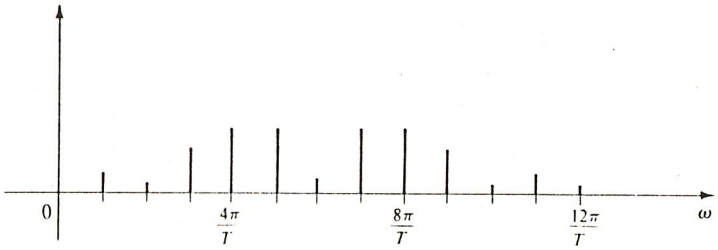


รูปที่ 2.20 แสดงสเปกตรัมของ cos w_1t



รูปที่ 2.21 แสดงสเปกตรัมของ cos w_2t

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.22 แสดงสเปกตรัมของคลื่นเอฟเอสเค

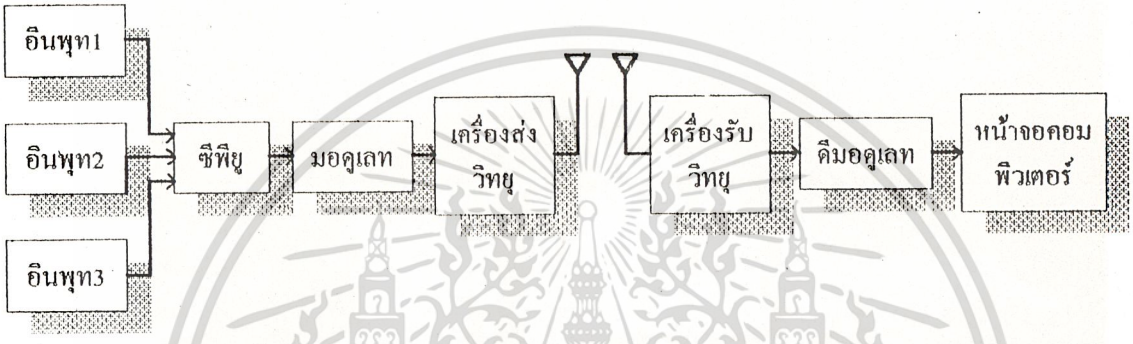
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3

การทดลองและผลการทดลอง

3.1 การทดลองที่ 1

จุดประสงค์การทดลองเพื่อเปลี่ยนจากการส่งข้อมูลอนุกรมผ่านสายตรงมาเป็นการมอดูเลทกับคลื่นวิทยุ แล้วส่งออกสายอากาศโดยการมอดูเลทแบบฟรีเควนซีชิฟคีย์อิง ทางด้านรับจะรับสัญญาณวิทยุเข้ามาทำการดีมอดูเลทเอา ข้อมูลอนุกรมออกมา บล็อกไดอะแกรมของการทดลองแสดงได้ดังนี้



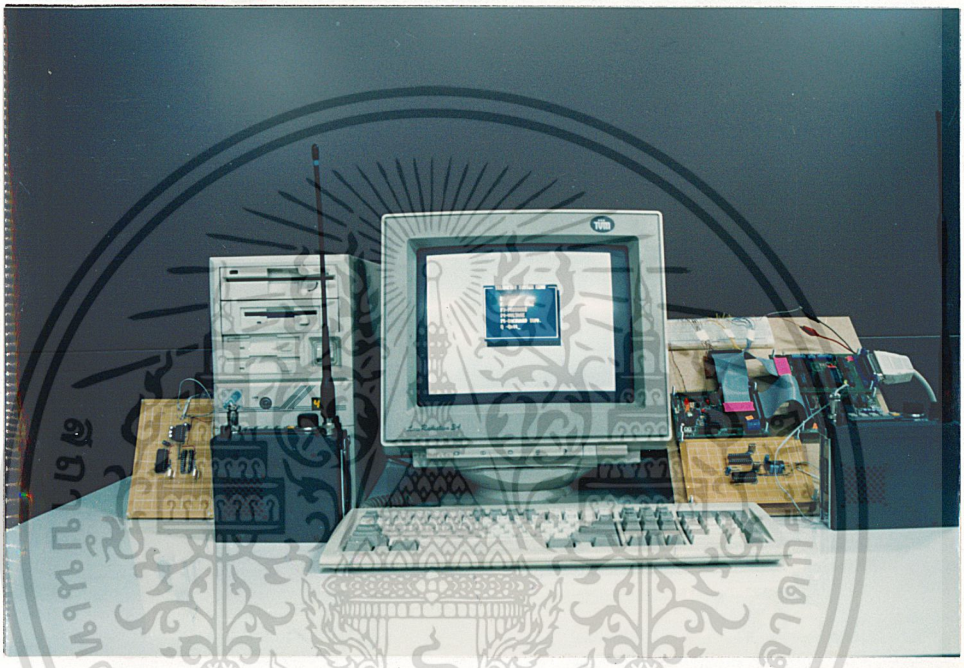
รูปที่ 3.1 แสดงบล็อกไดอะแกรม

เริ่มจากรับข้อมูลจากพอร์ทอินพุททุกๆ 1 ชั่วโมง แล้วส่งออกทางพอร์ทขนาน ผ่านอุปกรณ์เปลี่ยนข้อมูลขนานเป็นอนุกรม โดยมีรูปแบบของข้อมูลดังนี้

ปี	เดือน	วัน	เวลา	ข้อมูล1	ข้อมูล2	ข้อมูล3

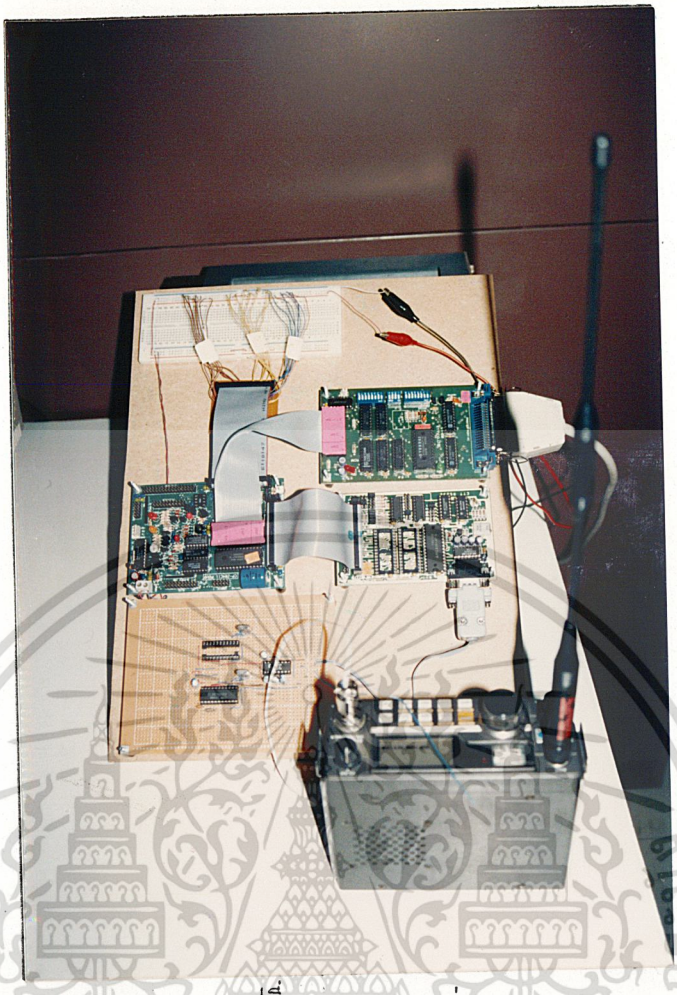
จากนั้นข้อมูลแบบอนุกรมจะส่งไปยังภาคมอดูเลเตอร์เพื่อทำการมอดูเลทระบบฟรีเควนซีชิฟคีย์อิง ซึ่งสัญญาณอินพุทดิจิตอลจะถูกเปลี่ยนเป็นสัญญาณอนาล็อกความถี่ย่านวิทยุแล้วส่งไปยังเครื่องส่งวิทยุเพื่อส่งออกอากาศ โดยให้ระยะห่างระหว่างภาคส่งและภาครับประมาณ 0.5 เมตร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.2แสดงระบบทั้งหมด

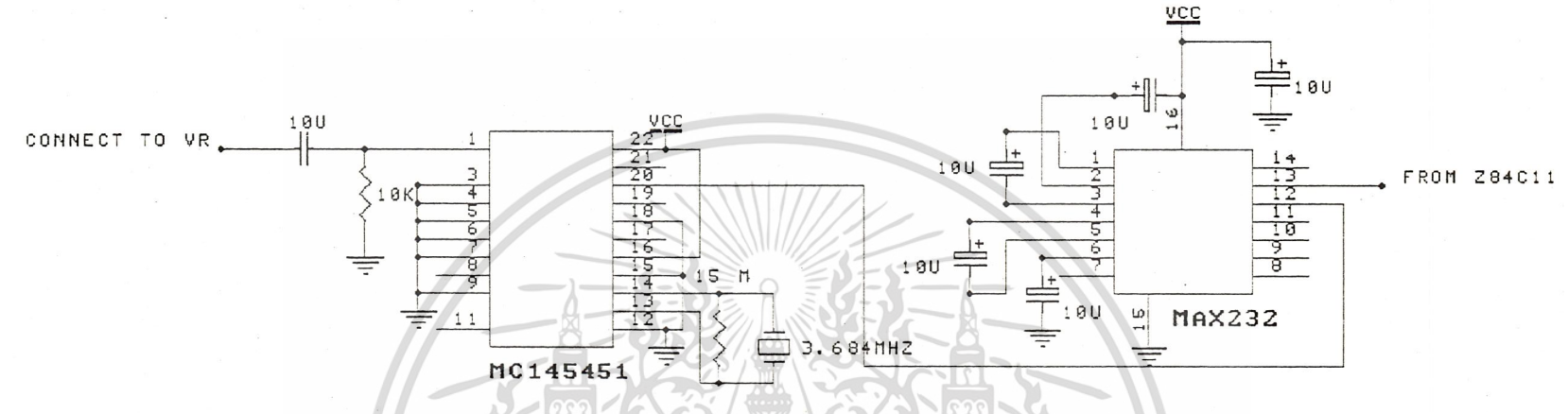
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



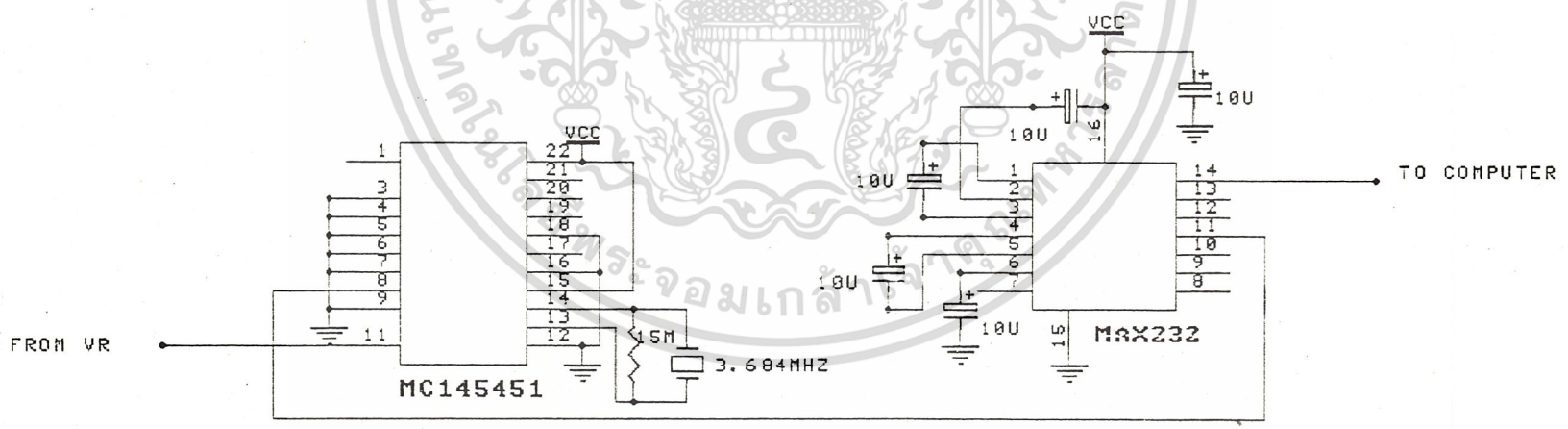
รูปที่ 3.3 แสดงภาคส่ง

รูปที่ 3.4 แสดงภาครับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

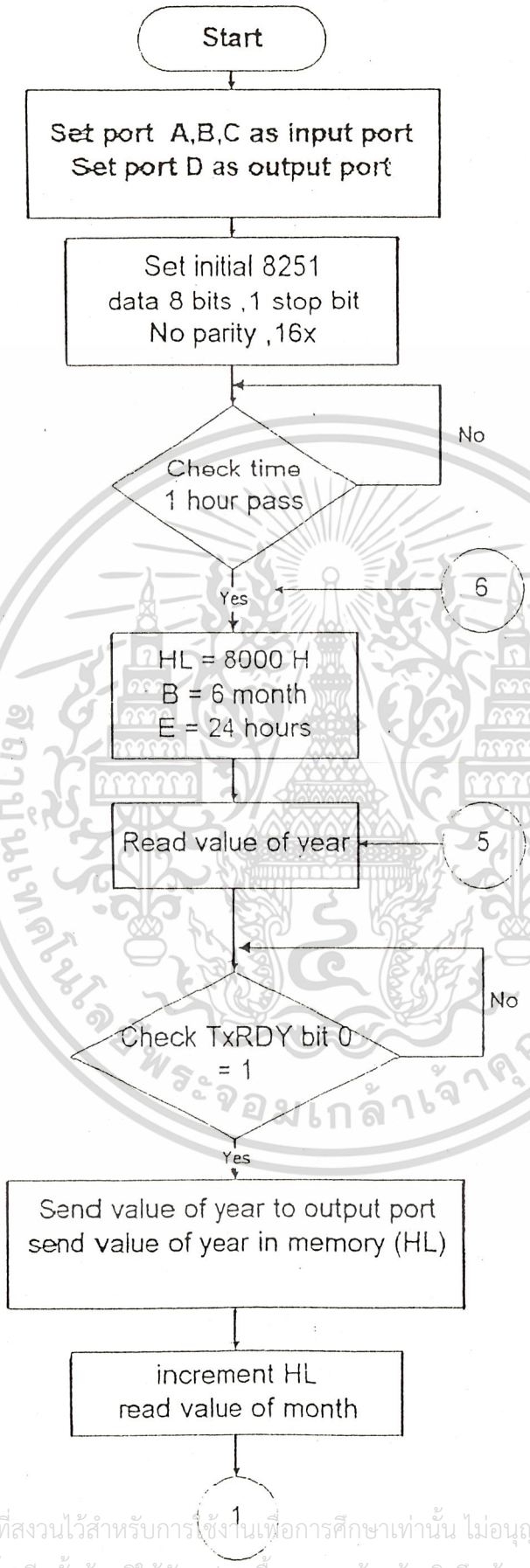


รูปที่ 3.5แสดงวงจรเอฟเอสเคมอดูเลเตอร์

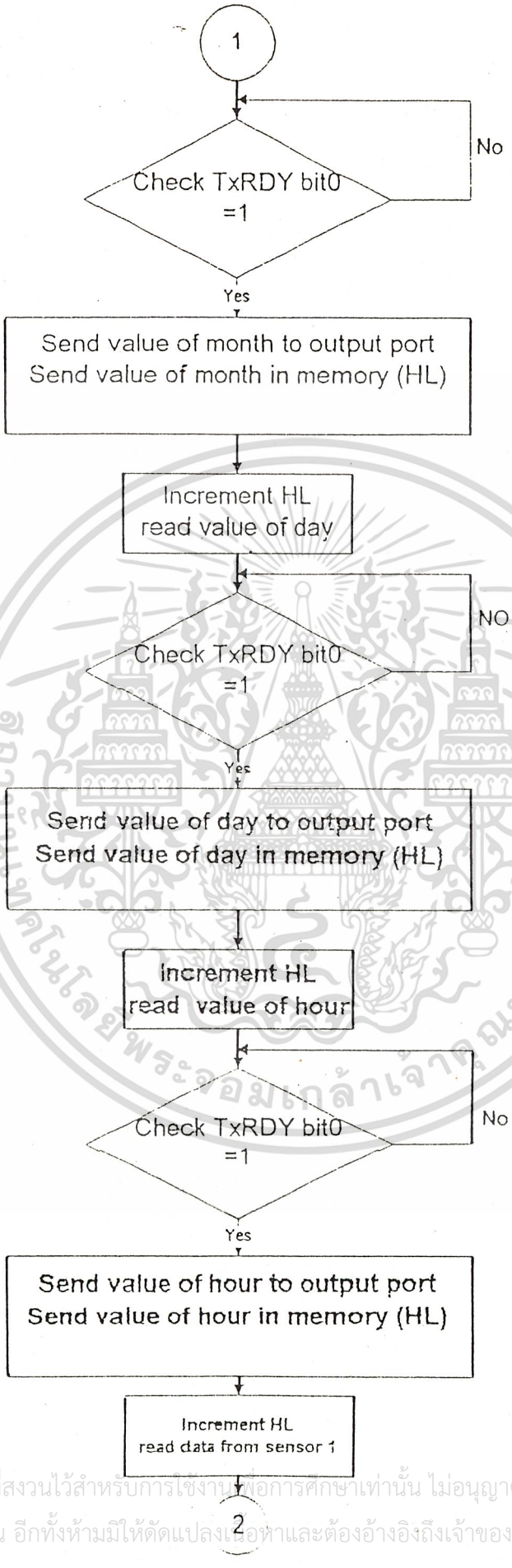


รูปที่ 3.6แสดงวงจรเอฟเอสเคลีมอดูเลเตอร์

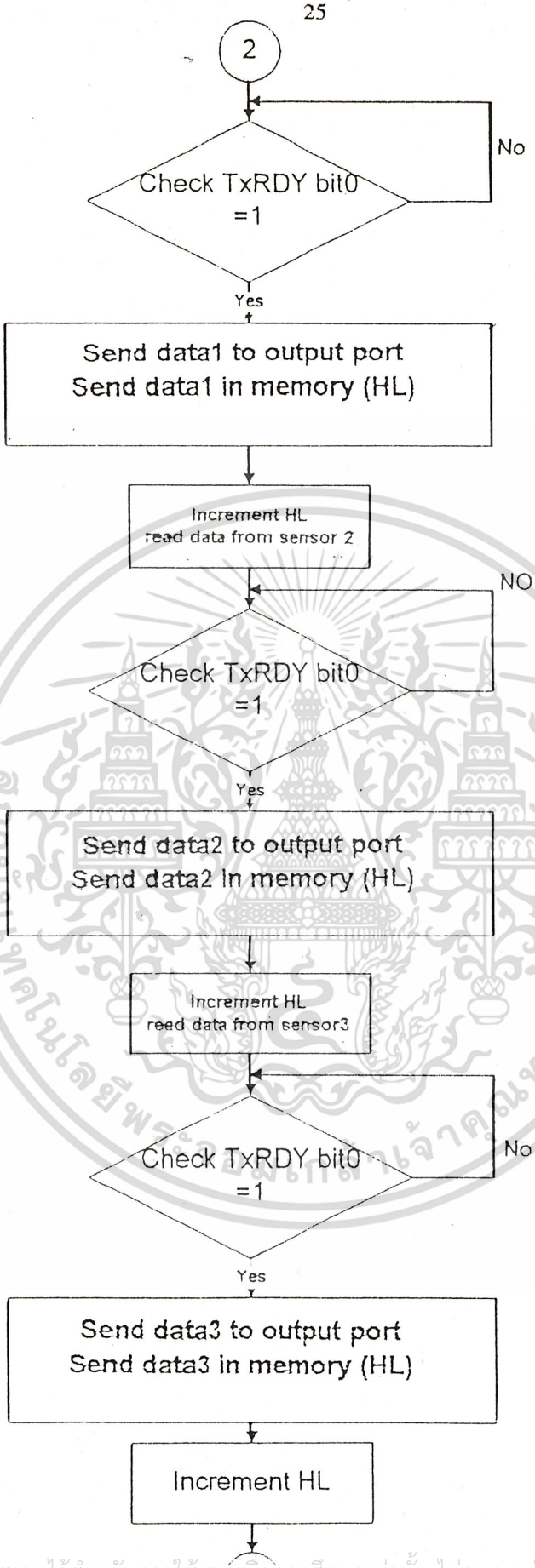
ก. โฟลวชาร์ท(Flow Chart)ของโปรแกรมภาคส่ง



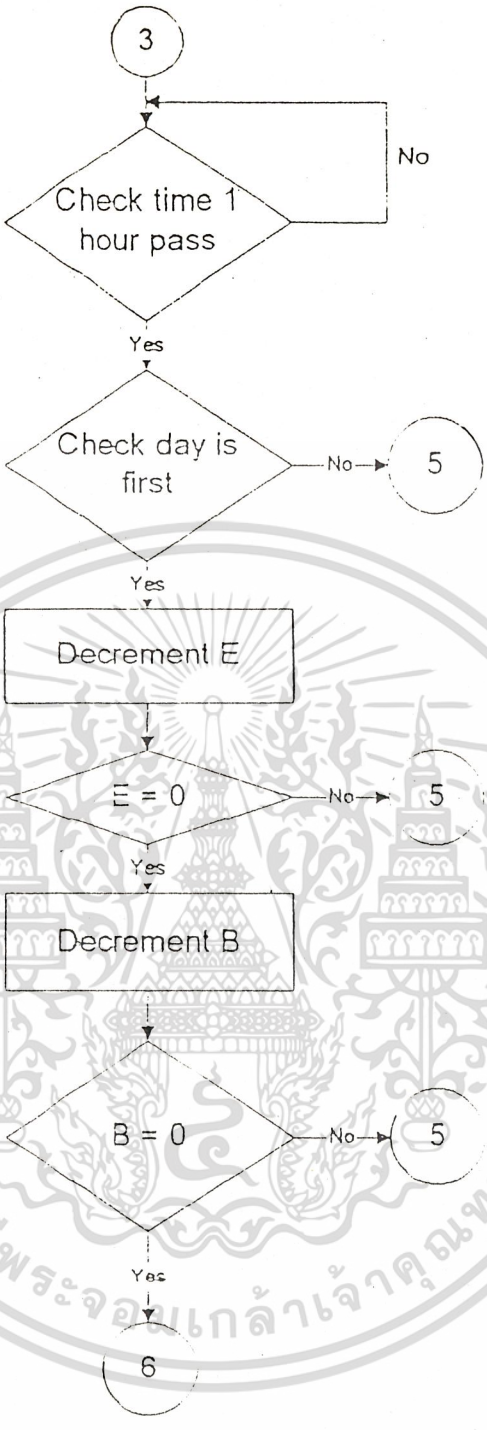
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

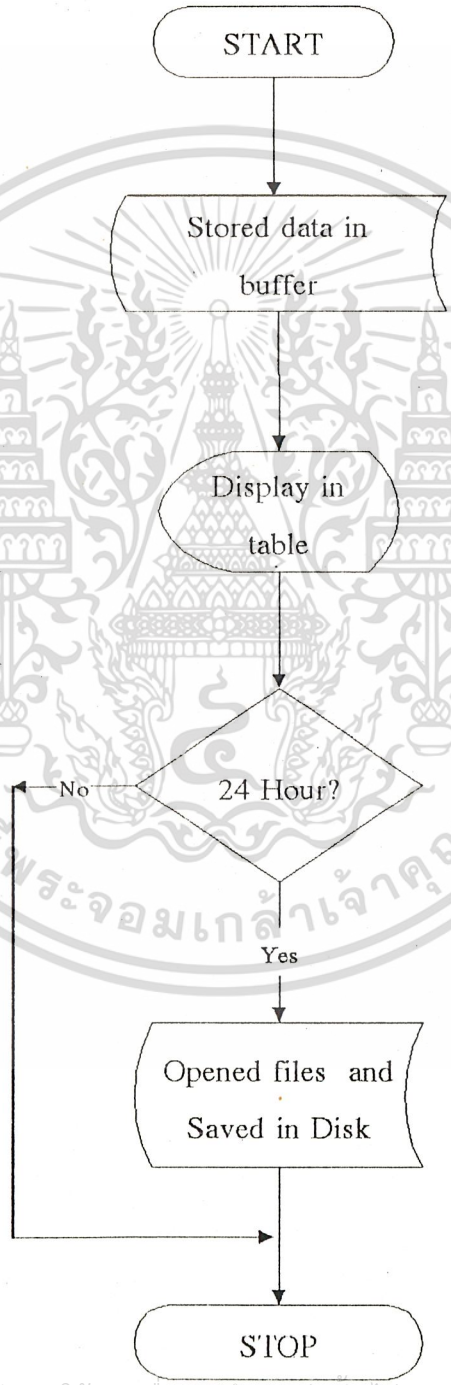


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



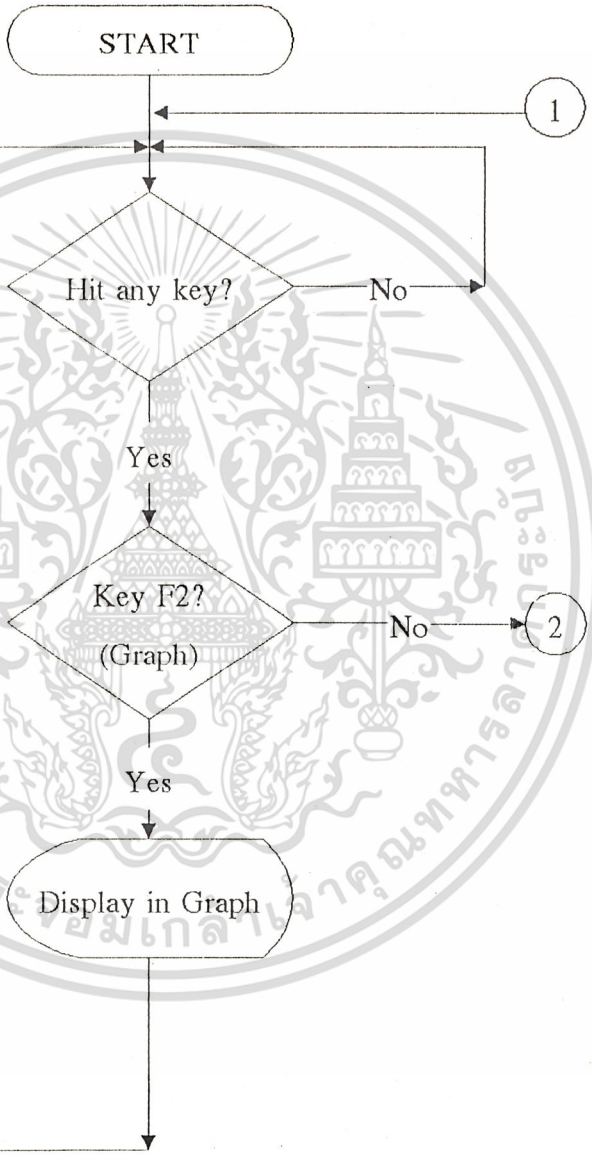
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ข.โฟลวชาร์ท ของโปรแกรมส่วนอินเทอร์รัพท์รับข้อมูล

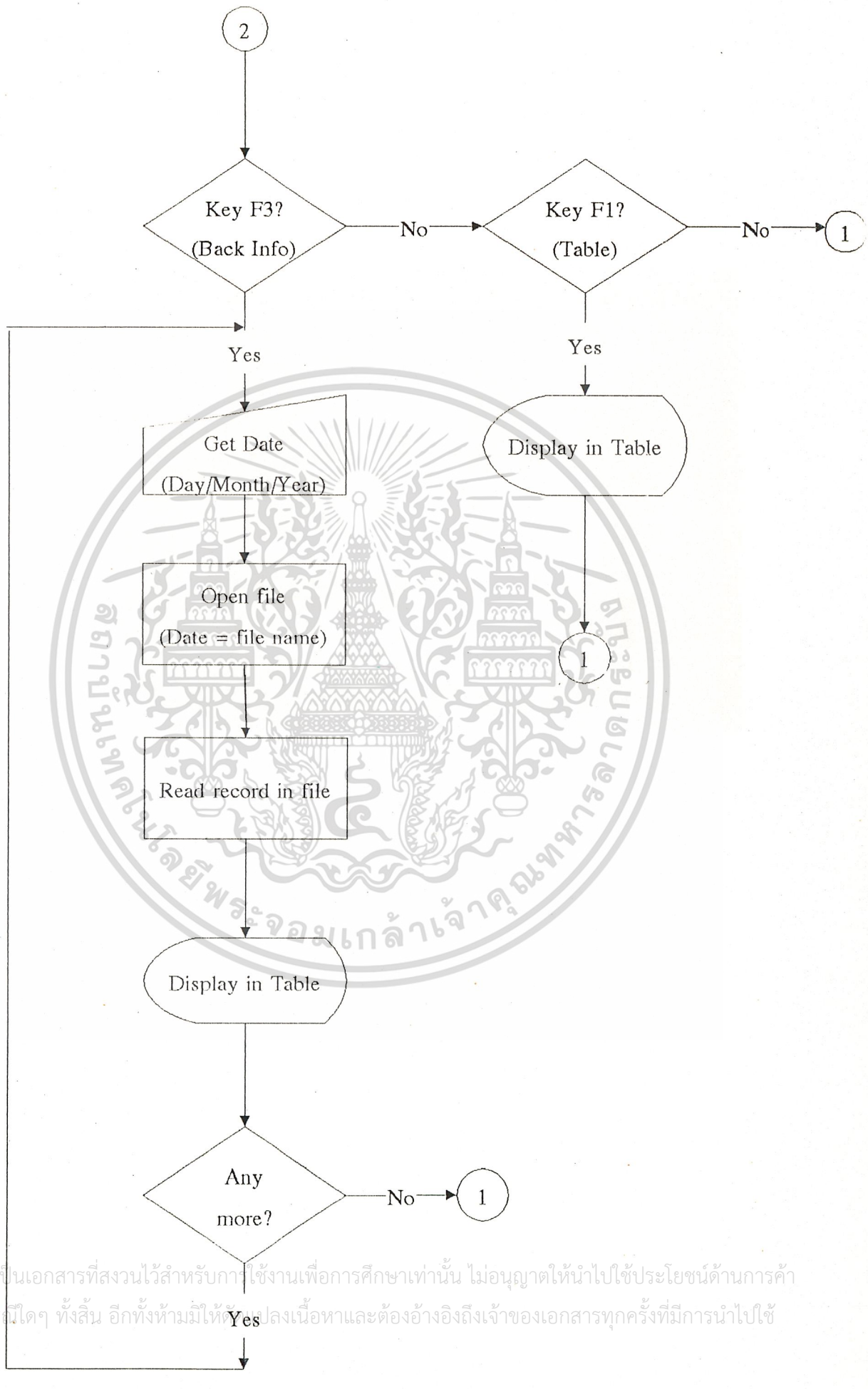


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ค.โฟลวชาร์ท ของโปรแกรมส่วนเลือกรูปแบบการแสดงผล



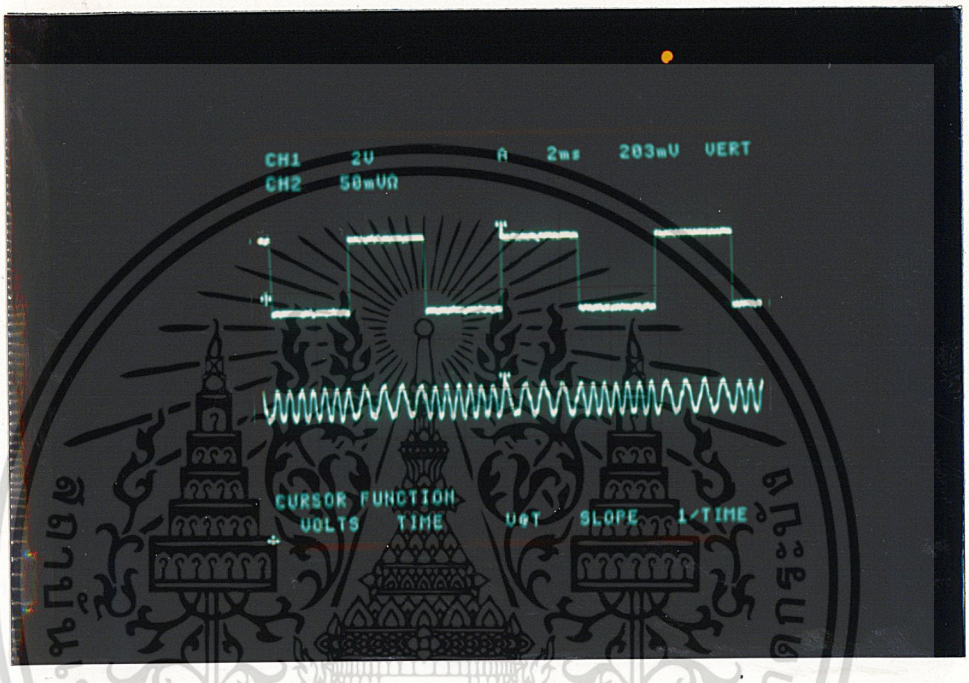
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

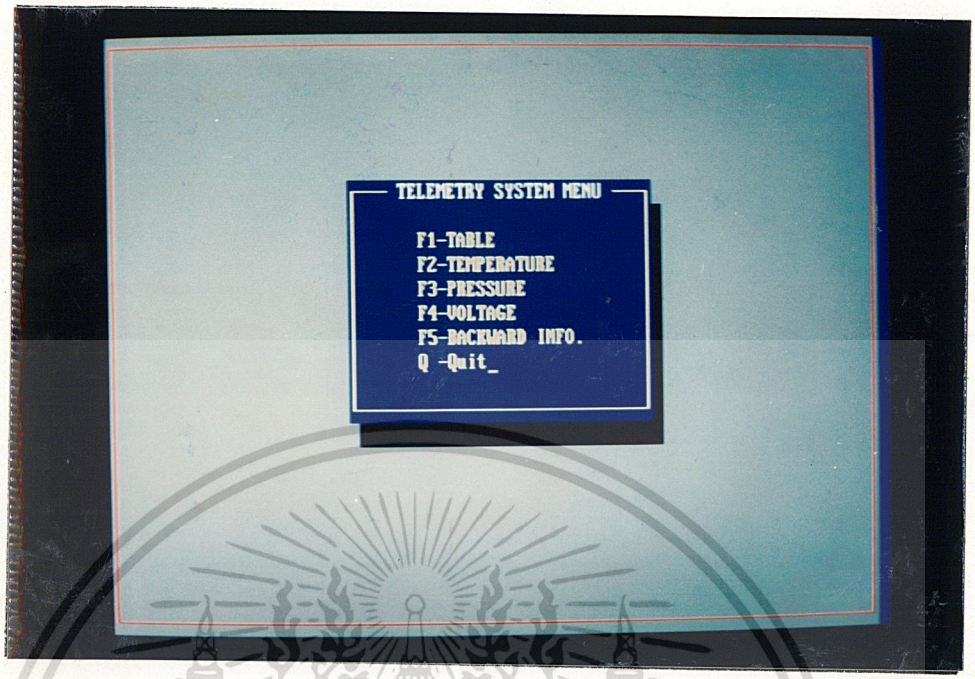
3.2 ผลการทดลองที่ 1

เมื่อทำการรันโปรแกรมจะได้การแสดงผลทางหน้าจอคอมพิวเตอร์ดังรูป 3.8 - 3.11

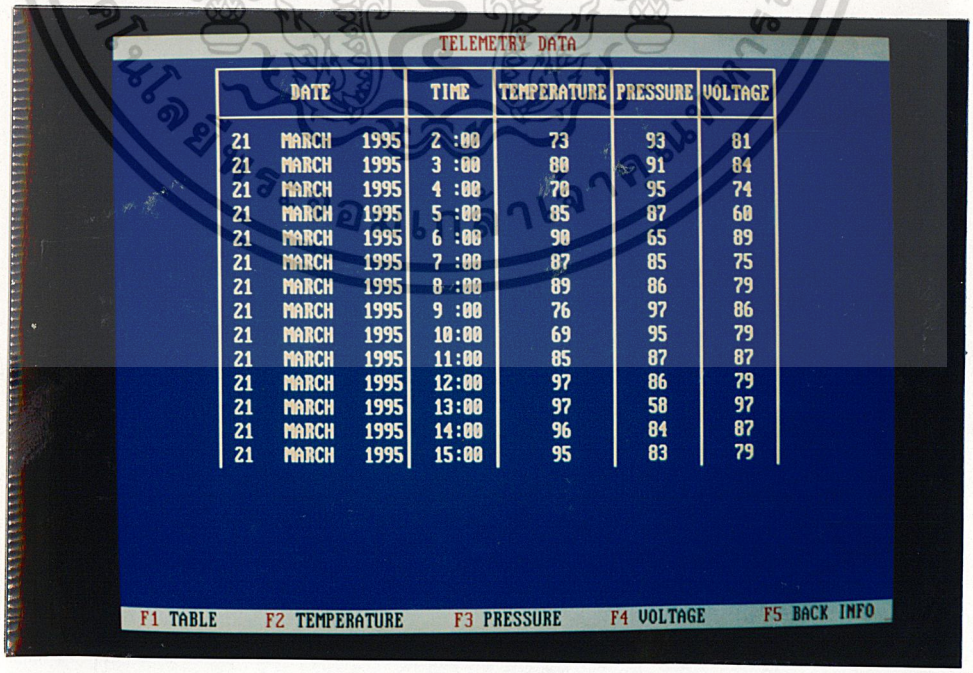


รูปที่ 3.7 แสดงสัญญาณเอฟเอสเอ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

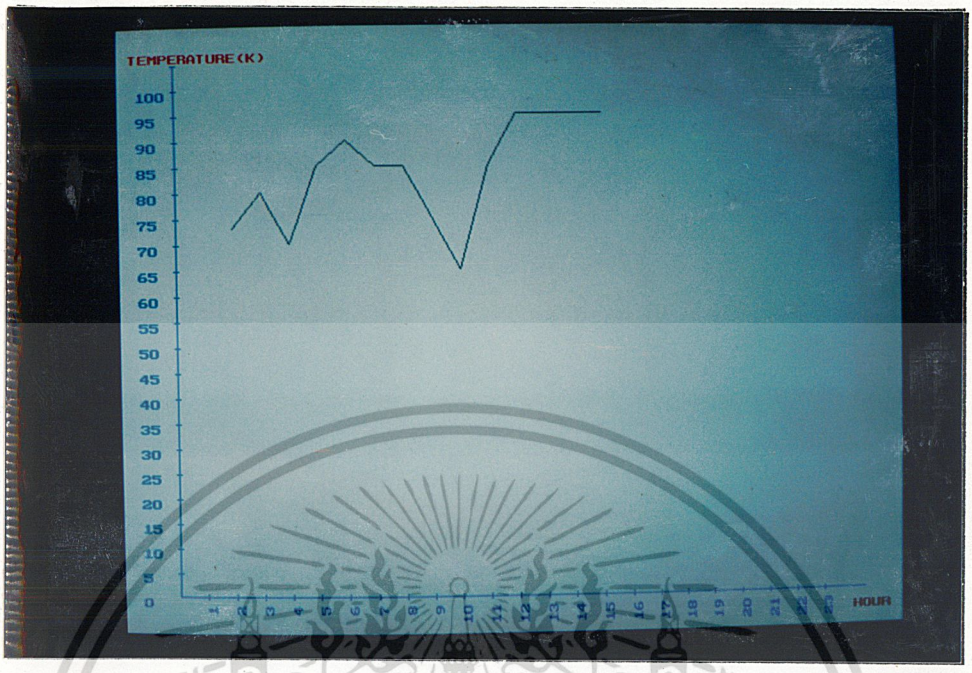


รูปที่3.8 แสดงเมนูต่างๆสำหรับการแสดงผล

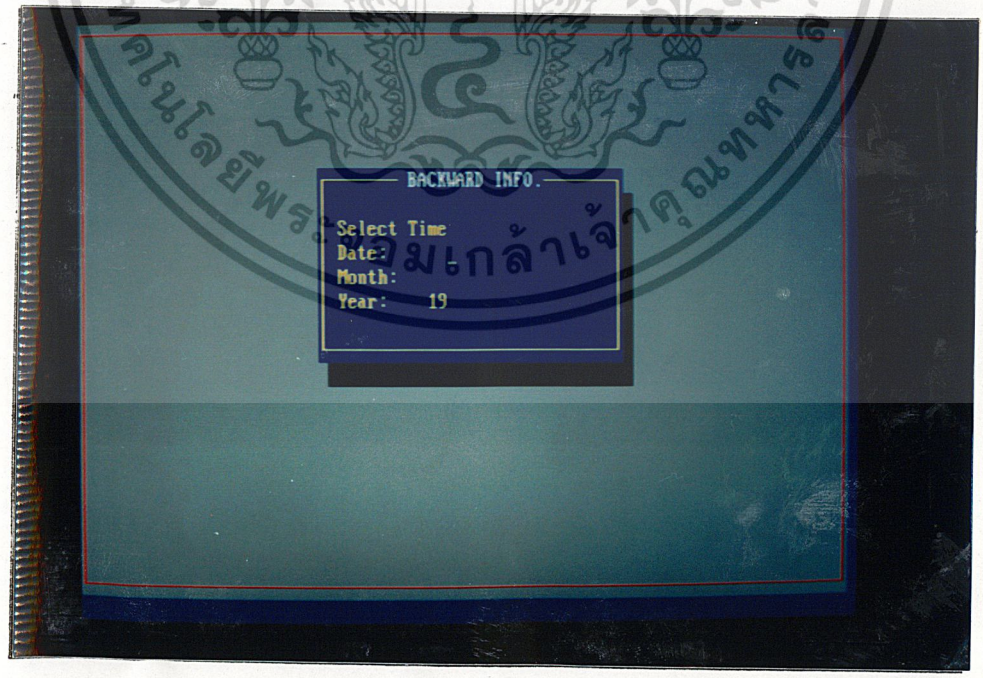


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่3.9 การแสดงผลแบบตาราง



รูปที่ 3.10 แสดงผลแบบกราฟ



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 3.11 แสดงหน้าจอของฟังก์ชัน "Back Info."

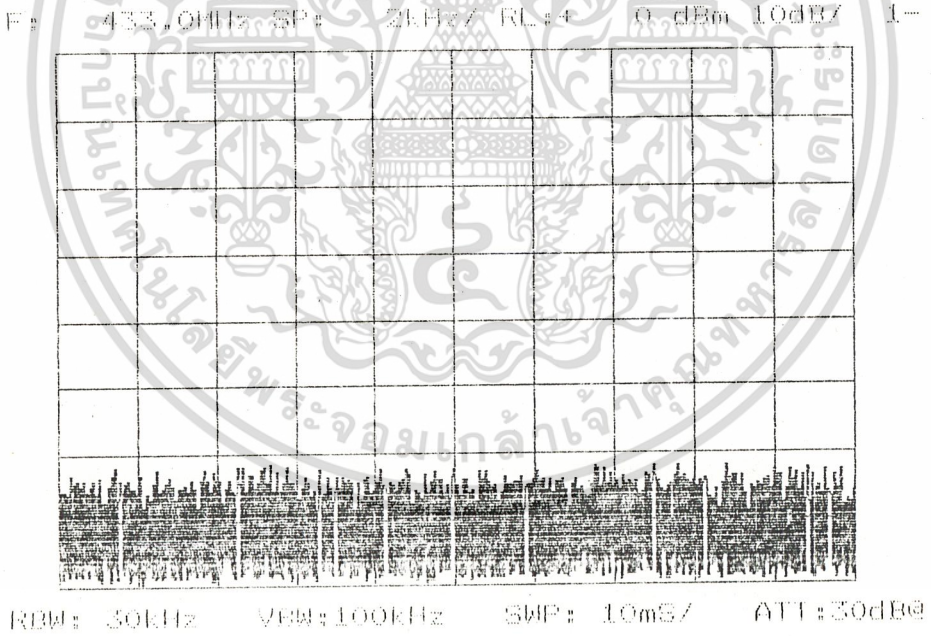
3.3 การทดลองที่ 2

จุดประสงค์เพื่อหาระยะทางสูงสุดที่สามารถส่ง-รับข้อมูลกันได้ ระหว่างภาคส่งและภาครับ

3.4 ผลการทดลองที่ 2

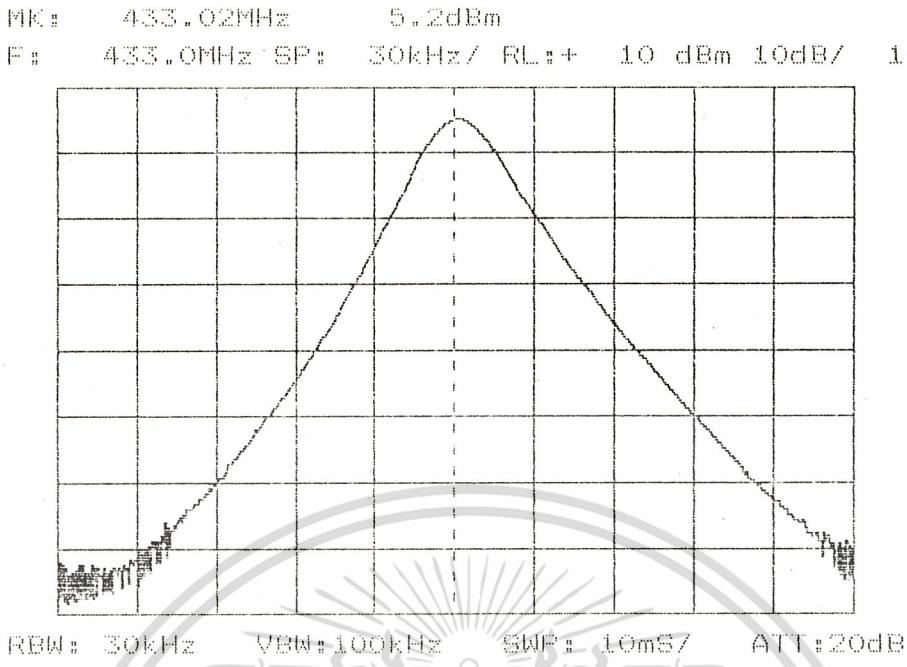
เมื่อได้ทำการจัดให้ภาคส่งและภาครับอยู่ห่างกันเป็นระยะทางต่างๆกันจะได้ว่าระยะห่างสูงสุดระหว่างภาคส่งและภาครับที่สามารถส่ง-รับข้อมูลกันได้อย่างถูกต้องในแนวราบคือ 145.9 เมตรโดยประมาณ

เมื่อทำการนำสเปคตรัมอนาไลซ์เซอร์มาทำการจับสัญญาณทางด้านรับในระยะประมาณ 0.5 เมตร และที่ระยะ 234 เมตรจะได้สเปคตรัมของสัญญาณทางด้านรับดังรูปที่ 3.13 , 3.14 ตามลำดับ ซึ่งจะพบว่าความแรงของสัญญาณสูงสุด(peak)ที่รับได้ในระยะใกล้มีค่าประมาณ 5.2 dBm และที่ระยะทาง 234 เมตร มีค่าประมาณ - 40.2 dBm

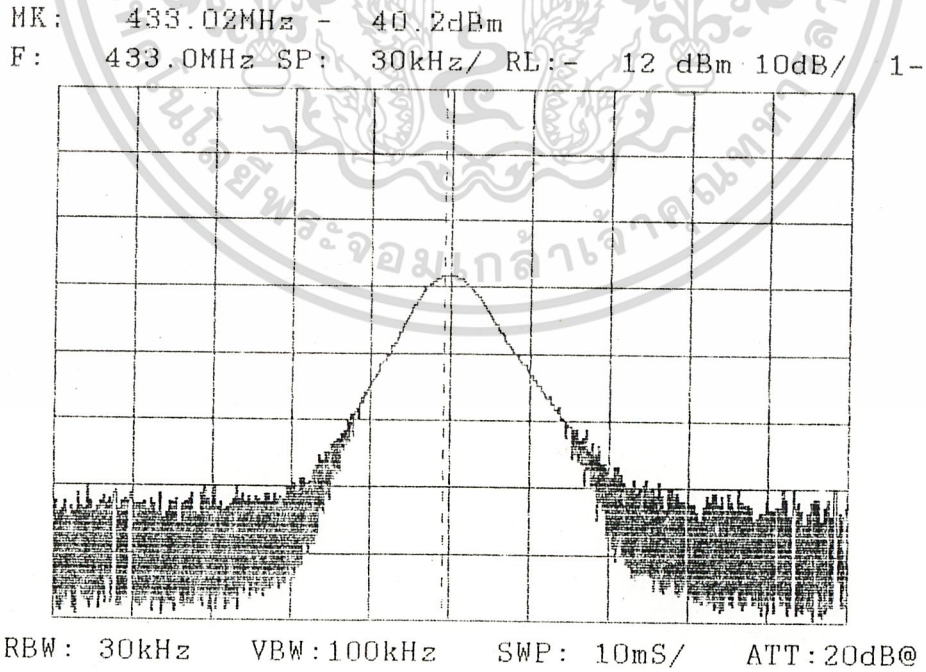


รูปที่ 3.12 แสดงสเปคตรัมในขณะที่ยังไม่มีการส่งคลื่นวิทยุ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.13 แสดงสเปกตรัมของสัญญาณคลื่นวิทยุที่ส่งออกอากาศที่ระยะประมาณ 0.5 เมตร



รูปที่ 3.14 แสดงสเปกตรัมของสัญญาณคลื่นวิทยุที่ส่งออกอากาศที่ระยะ 234 เมตร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่4

สรุปผลและวิจารณ์

4.1 สรุปผลและวิจารณ์

จากการทดลองที่ 1 สามารถทำการนำข้อมูล 3 ชุด ชุดละ 8 บิทมามอดูเลทกับแคร์เรียร์คลื่นวิทยุย่านยู เฮท เอฟ (UHF) ส่งออกอากาศและสามารถรับได้ทางด้านรับและดีเทกข้อมูลทั้ง 3 ชุดนั้นออกมาแสดงทางหน้าจอได้ ทั้งแบบตารางและแบบกราฟ ตามเวลาทุกๆ 1 ชั่วโมงที่ได้ตั้งไว้ ดังผลการทดลอง นอกจากนี้ฟังก์ชันเรียกดูข้อมูลย้อนหลังที่สร้างขึ้นก็สามารถใช้งานได้เป็นอย่างดี

ในการทดลองที่ 2 จากผลการทดลองจะได้ระยะทางมากที่สุดที่สามารถส่งและรับข้อมูลได้อย่างถูกต้องคือ 145.9 เมตร เมื่อเพิ่มระยะทางเป็น 234 เมตรจะพบว่าข้อมูลที่ด้านรับรับได้ไม่ครบทุกไบต์และรับได้ผิดพลาดสันนิษฐานว่าอาจเนื่องมาจากการลดทอนของสัญญาณที่เข้ามาทางเครื่องรับวิทยุซึ่งมีความแรงของสัญญาณลดลงเป็นอย่างมากดังรูปสเปคตรัมที่แสดงในผลการทดลอง เพราะได้ทำการทดสอบการดีมอดูเลทของไอซีเบอร์นี้พบว่า หากสัญญาณอินพุทที่จะมาทำการดีมอดูเลทต่ำกว่า 5 โวลท์จะทำให้การดีมอดูเลทของไอซีเกิดการผิดพลาดขึ้นดังนั้นจึงเป็นไปได้ที่จะสามารถส่งและรับข้อมูลได้อย่างถูกต้องที่ระยะทางสูงสุดเพียงเท่านั้น ทั้งๆที่เครื่องรับส่งวิทยุน่าจะมีประสิทธิภาพในการส่งและรับสัญญาณได้ไกลกว่านี้

4.2 แนวทางในการพัฒนาต่อไป

เนื่องจากโครงงานที่ได้ทำนี้เป็นเพียงการจำลองระบบรายงานค่าพารามิเตอร์ของดาวเทียมเฉพาะในส่วนที่สำคัญๆของระบบทั้งหมดเช่น ส่วนประมวลผลข้อมูลจากเซนเซอร์แต่ละตัว การส่งข้อมูลโดยการใช้ระบบการสื่อสารด้วยคลื่นวิทยุ(Radio Wave Communication System) การเก็บข้อมูลที่ได้รับและการแสดงผลของทางด้านรับ(สถานีภาคพื้นดิน) ดังนั้นขอบเขตของโครงงานนี้จึงมิได้ครอบคลุมระบบรายงานค่าพารามิเตอร์ของดาวเทียมโดยทั้งหมด เช่น ในโครงงานนี้ไม่รวมถึงการทำเซนเซอร์เพื่อป้อนข้อมูลอินพุทให้กับหน่วยประมวลผล(บนดาวเทียม)แต่จะใช้วิธีสมมติ ข้อมูลอินพุทจากเซนเซอร์ 3 ตัว โดยการป้อนไฟให้แก่แต่ละบิทแทน และเปลี่ยนค่าอินพุททุกๆชั่วโมงให้เหมือนกับข้อมูลจากเซนเซอร์ที่อาจมีการเปลี่ยนแปลงทุกชั่วโมง

ดังนั้นแนวทางในการพัฒนาต่อของโครงการนี้จึงเป็นการทำให้โครงงานนี้มีขอบเขตครอบคลุมระบบรายงานค่าพารามิเตอร์ของดาวเทียมได้มากขึ้นกว่านี้จนมากที่สุด ซึ่งทำได้โดยการลดข้อจำกัดดังต่อไปนี้

1.ใช้เซนเซอร์ที่สามารถให้ค่าเอ้าพุทแก่หน่วยประมวลผลเป็นข้อมูลขนาด 8 บิทต่อ 1 ชุด ดังนั้นจึงทำให้ข้อมูลถูกจำกัดเพียงค่า ตั้งแต่ 0 ถึง 255 คือถ้าเป็นข้อมูลอุณหภูมิสามารถส่งได้ค่าอุณหภูมิสูงสุด 255 องศาและไม่สามารถแสดงผลในรูปทศนิยมได้

2.เนื่องจากการ์ด Z84C11 มีพอร์ททั้งหมด 5 พอร์ท ถูกใช้เป็นพอร์ทเอ้าพุท 1 พอร์ทเพื่อส่งข้อมูลออก ดังนั้นจึงเหลือสำหรับใช้เป็นพอร์ทอินพุทจากเซนเซอร์ต่างๆได้เพียง 4 พอร์ท เท่านั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.ส่วนจ่ายกำลังไฟฟ้า(Power Supply Part)เป็นอีกส่วนหนึ่งที่มีผลต่อประสิทธิภาพของระบบเนื่องจากในส่วนของภาคส่งประกอบด้วยการ์ด Z84C11 เครื่องรับวิทยุ ฯลฯ ซึ่งจะถูกติดตั้งไปกับดาวเทียม(ในสภาพใช้งานจริง) ดังนั้นจึงจำเป็นจะต้องได้รับการจ่ายไฟจากแหล่งจ่ายไฟอย่างตลอดเวลา เช่นได้พลังงานไฟฟ้าจากเซลล์สุริยะ(Solar cell) ดังในสภาพการใช้งานจริง แต่เนื่องจากในการทำโครงงานไม่สามารถทำเช่นนั้นได้ ทำได้เพียงใช้แหล่งจ่ายไฟ(Power supply) จ่ายให้แทนดังนั้นประสิทธิภาพในการใช้งานโครงงานจึงถูกจำกัดให้ทำได้ในบริเวณที่สามารถใช้แหล่งจ่ายไฟได้ ไม่สามารถปล่อยขึ้นไปในอากาศได้ อย่างไรก็ตามอาจทำได้โดยการใช้ไฟจากแบตเตอรี่สำรองที่มีอยู่ในการ์ดZ84C11 และใช้แบตเตอรี่ของเครื่องส่งวิทยุดังนั้นอายุการทำงานของระบบด้านส่งจึงขึ้นอยู่กับว่าแบตเตอรี่ทั้ง 2 สามารถจ่ายไฟได้นานแค่ไหน

4.โครงงานที่ได้ทำมีการสื่อสารข้อมูลในแบบทางเดียว(simplex)จากด้านส่งมายังด้านรับจึงเกิดข้อจำกัดในการควบคุมการผิดพลาด(Error Control)คือ หากจะทำการควบคุมการผิดพลาดแบบเออร์เรอร์ คีเทคชัน(Error Detection) โดยการเช็คที่บิทพาร์ริตี้(Parity bit)ก็ไม่สามารถทำได้เพราะเมื่อด้านรับเช็ครู้ว่าไบต์ที่ส่งมาผิดก็จะต้องส่งสัญญาณกลับไปบอกที่ด้านส่งเพื่อขอร้องให้ส่งข้อมูลที่ผิดมาใหม่ซึ่งเป็นการส่งแบบ 2 ทาง(Duplex อาจจะเป็น Full Duplex หรือ Half Duplex)

4.3 แนวทางการประยุกต์ใช้

จากระบบโครงงานที่ได้ทำขึ้นนี้ นอกจากจะเป็นการจำลองระบบรายงานค่าพารามิเตอร์ของดาวเทียมซึ่งเป็นจุดประสงค์หลักของโครงงานนี้ ยังสามารถนำมาประยุกต์ใช้งานอื่นๆได้อีกดังเช่น อาจนำมาประยุกต์ใช้เกี่ยวกับการวัดคุณภาพอากาศ(เช่น เปอร์เซนต์ของก๊าซคาร์บอนมอนอ็อกไซด์:CO) โดยกำหนดตำแหน่งศูนย์กลางการรับคลื่นวิทยุไว้จุดหนึ่งเพื่อติดตั้งภาครับซึ่งมีอุปกรณ์ต่างๆเหมือนในโครงงาน และติดตั้งภาคส่งซึ่งจะเพิ่มเซนเซอร์ที่ใช้สำหรับวัดเปอร์เซนต์ของก๊าซชนิดที่ต้องการวัดนั้นๆไว้ตามจุดต่างๆบริเวณรอบๆภาครับ เช่นตามสี่แยกต่างๆ โดยตั้งความถี่ของคลื่นวิทยุที่ใช้เป็นคลื่นพาหะของภาคส่งแต่ละจุดให้มีความถี่ต่างกัน และในการทำการส่งข้อมูลที่วัดได้จากเซนเซอร์โดยส่งออกอากาศมายังภาครับ(ศูนย์รายงานผล) จะต้องตั้งเวลาให้เรียงตามลำดับกันไปถึงจะต้องไม่ส่งพร้อมกัน ทางภาครับก็สามารถปรับจูนความถี่ให้สอดคล้อง(Synchronous)กับเครื่องส่งแต่ละจุดเพื่อเลือกรับข้อมูลจากจุดต่างๆที่ติดตั้งเซนเซอร์ไว้ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

บทแทรก

5.1โปรแกรมทางด้านส่งบนดาวเทียม

```
        CPU "Z80.TBL"
        HOF "INT8"
        ORG 0000H
;***  DEFINE VARIABLE  ***;
        PRTA:  EQU 50H
        PRTB:  EQU 51H
        PRTC:  EQU 52H
        PRTD:  EQU 30H
        PRTE:  EQU 40H

;***  PRE MAIN PROGRAM  ***;

        LD A,00H        ;SET STATUS INPUT PORT
        OUT (54H),A
        OUT (55H),A
        OUT (56H),A
        LD A,0FFH       ;SET STATUS OUTPUT PORT
        OUT (34H),A

;***  INITIAL 8251 ***;

        LD A,40H        ;RESET
        OUT (31H),A
        LD A,4EH        ;DATA 8 BIT,1 STOP BIT,NO PARITY,16X
        OUT (31H),A
        LD A,37H        ;RTS,ERROR RESET,DTR
        OUT (31H),A
        LD A,21H        ;BOTH RECEIVE AND TRANSMIT
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        OUT (31H),A

;***  MAIN PROGRAM FOR TRANSFER FRAME OUT OFF PORT  ***;

        CALL MIN
MAIN:   LD HL,8000H
        LD B,06H        ;SET FOR 6 MONTHS
        LD E,18H        ;SET F0R 24 TIMES A DAY

FRAME:  CALL RTC        ;PUT/SAVE TIME OUT

CHKSND5: IN A,(31H)
        BIT 0,A         ;CHECK TXRDY BIT 0
        JR Z,CHKSND5

        IN A,(PRTA)     ;IN DATA1
        OUT (PRTD),A    ;OUT DATA1
        LD (HL),A       ;SAVE DATA1
        INC HL

CHKSND6: IN A,(31H)
        BIT 0,A         ;CHECK TXRDY BIT 0
        JR Z,CHKSND6

        IN A,(PRTB)     ;IN DATA2
        OUT (PRTD),A    ;OUT DATA2
        LD (HL),A       ;SAVE DATA2
        INC HL

        CHKSND7: IN A,(31H)
        BIT 0,A         ;CHECK TXRDY BIT 0
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        JR Z,CHKSND7

        IN A,(PRTC)     ;IN DATA3
        OUT (PRTD),A    ;OUT DATA3
        LD (HL),A       ;SAVE DATA3
        INC HL


        CALL MIN

        LD A,D          ;CHECK FOR END OF RAM
        SUB 01H
        JR NZ,FRAME

        DEC E           ;CHECK FOR 24 HOUR PER 1ST
        JR NZ,FRAME

        DEC B
        JR Z,MAIN
        JP FRAME


;***  SET TIME OF RTC  ***;

RTC:    LD A,01H        ;SET 24 HOURS
        OUT (0AFH),A
        LD A,04H
        OUT (0AFH),A
        LD A,00H
        OUT (0AFH),A
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
YEAR:   IN A,(0AAH)
        AND 0FH
        LD C,A
        IN A,(0ABH)
        SLA A
        SLA A
        SLA A
        SLA A
        ADD A,C
        LD C,A

CHKSND1: IN A,(31H)
        BIT 0,A         ;CHECK TXRDY BIT 0
        JP Z,CHKSND1

        LD A,C
        OUT (30H),A
        LD (HL),A       ;SAVE YEAR AT RAM
        INC HL

MONTH:  IN A,(0A8H)
        AND 0FH
        LD C,A
        IN A,(0A9H)
        SLA A
        SLA A
        SLA A
        SLA A
        ADD A,C

        LD C,A
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
CHKSND2: IN A,(31H)
        BIT 0,A         ;CHECK TXRDY BIT 0
        JR Z,CHKSND2

        LD A,C
        OUT (30H),A
        LD (HL),A       ;SAVE MONTH
        INC HL

DAY:    IN A,(0A6H)
        AND 0FH
        LD C,A
        IN A,(0A7H)
        SLA A
        SLA A
        SLA A
        SLA A
        ADD A,C
        LD D,A          ;SAVE DAY IN REG D FOR CHECK LOOP
        LD C,A

CHKSND3: IN A,(31H)
        BIT 0,A         ;CHECK TXRDY BIT 0
        JR Z,CHKSND3

        LD A,C
        OUT (30H),A
        LD (HL),A       ;SAVE DAY
        INC HL

HOUR:   IN A,(0A4H)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        AND 0FH
        LD C,A
        IN A,(0A5H)
        SLA A
        SLA A
        SLA A
        SLA A
        ADD A,C
        LD C,A

CHKSND4: IN A,(31H)
        BIT 0,A          ;CHECK TXRDY BIT 0
        JR Z,CHKSND4

        LD A,C
        OUT (30H),A
        LD (HL),A        ;SAVE HOUR
        INC HL
        RET

;***  PROGRAM FOR LOOP TIME 1 HOUR ***;

MIN:    IN A,(0A2H)      ;READ MINUTE  FOR CHECK 1 HOUR
        AND 0FH
        LD C,A
        IN A,(0A3H)
        SLA A
        SLA A

        SLA A
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        SLA A
        ADD A,C

        SUB 59H
        JR NZ,MIN

MIN1:   IN A,(0A2H)  ;PROTECT SEND MORE ONE TIME
        AND 0FH
        LD C,A
        IN A,(0A3H)
        SLA A
        SLA A
        SLA A
        SLA A
        ADD A,C

        SUB 00H
        JR NZ,MIN1
        RET

END
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.2 โปรแกรมทางด้านรับภาคพื้นดิน

```
#include"stdio.h"
#include"conio.h"
#include"bios.h"
#include"dos.h"
#include"string.h"
#include <graphics.h>
#include <stdlib.h>
#include <time.h>
void TEMP_axis1(int xmax,int ymax);
void TEMP_axis2(int xmax,int ymax);
void Initialize_Graphics_Mode(void);
void TEMP_axis(int xmax,int ymax);
void BKCOLOR(int color);
void WR_gph(int xdev,int dy,int colour);
void Table(int comp,int j,int data[][3],int date[][3]);
void Graph(int comp,int j,int data[][3]);
void Graph1(int comp,int j,int data[][3]);
void Graph2(int comp,int j,int data[][3]);
void BLACKGRND(void);
void ATTRIBUTE(void);
void TextWindow(int x1,int y1,int x2,int y2,int fcolor,int bcolor);
void Backinfo(void);
void Title(void);
void Cancel(void);
void R_File(int idte[3]);
void Error_To_read(int p1,int q1,int p2,int q2,int fcolor,int bcolor);
void ERROR_R(void);
void QuitSerialComm(void);
void O_File(int dta[][3],int dte[][3]);
void InitSerialComm(void);
void interrupt STORE(void);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
void interrupt (*OLD)(void);
void transfer(void);
void Error_To_read(int p1,int q1,int p2,int q2,int fcolor,int bcolor);
void Error_To_write(int l1,int m1,int l2,int m2,int fcolor,int bcolor);
void ETO(void);
void ERROR_W(void);
void ETOP(int t1,int u1,int t2,int u2,int fcolor,int bcolor);

#define DLL  0
#define IER  1
#define DLH  1
#define IIR  2
#define LCR  3
#define MCR  4
#define LSR  5
#define MSR  6

#define EOI  0x20
#define COM1  1
#define IRQ4FLAG 0x10
#define INTV  0x0C     /* INT 0x0C */
#define IMR  0x21
#define CommAddr 0x3F8

#define B1200  0
#define D8  0
#define S1  0
#define PNONE  0

#define false 0
#define true 1
int in_dte[3];
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
int dt[30][3];
struct  {
  int data1;
  int data2;
  int data3;
  }s_dt;

unsigned baud = 96;          /* 96 for 1200 bps,48 :2400 bps */
char Data =  0x03;           /* feature bit of 8-bit */
char stop =  0x00;           /* feature bit of 1-bit */
char pari =  0x00;           /* feature bit of none  */
int  Chck = 1;
int  Buf  = 0;
int  data_[7];
int  CommBuf[7];             /* Buffer for storing data from modem */

        int data_in[7],i,j=-1,k,h=0,z=0;
        int data[24][3];
        int date[24][3];
        char mmm,key,month[9];
        div_t qu;

main()
{
        int keycheck=0,ASCII=0;
        Title();
        InitSerialComm();
        while (1)
        {   keycheck=bioskey(1);     /*Check hit key*/

          while (keycheck)
          {
                ASCII=bioskey(0);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
if (ASCII == 0x3b00)    /*if press F1 ,show Table*/
{   z=1;
     closegraph();
     window(1,1,80,25);
     clrscr();
     textattr(14 + (1 << 4));
     clrscr();
    ATTRIBUTE();
     Table(data_in[3],j,data,date);
}
     if (ASCII == 0x3c00)  /*if press F2,show Graph of data1*/
     {  z=2;
        Graph(data_in[3],j,data);
     }
     if (ASCII == 0x3d00)  /*if press F3,show Graph of data2*/
     {  z=3;
        Graph1(data_in[3],j,data);
     }
     if (ASCII == 0x3e00)  /*if press F4,show Graph of data3*/
     {  z=4;
        Graph2(data_in[3],j,data);
     }
     if (ASCII == 0x3f00)  /*if press F5,show menu of Back information*/
     {
        closegraph();
        window(1,1,80,25);
        clrscr();
        Backinfo();
     }
     if (ASCII == 0x011b)  /*if press ESC,back to main menu*/
     {
        closegraph();
        window(1,1,80,25);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            clrscr();
            Title();
          }
          if (ASCII == 0x1071 ) /*if press q,quit to dos*/
          {  closegraph();
            window(1,1,80,25);
            clrscr();
            exit(0);
          }
  keycheck=0;
 }
if (h==7)  /*if data_in 7 byte*/
{
h=0;
transfer(); /*save data_in to array 2 dimension "data[24][3]"and"date[24][3]*/
switch(date[data_in[3]][1]) /*take data_in 3rd byte for show the name of month*/
{
                case 1:
                  strcpy(month," JANUARY ");
                  break;
                case 2:
                  strcpy(month," FEBRUARY");
                  break;
               case 3:
                  strcpy(month,"  MARCH  ");
                  break;
               case 4:
                  strcpy(month,"  APRIL  ");
                  break;
               case 5:
                  strcpy(month,"   MAY   ");
                  break;
               case 6:
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            strcpy(month," JUNE  ");
            break;
          case 7:
            strcpy(month," JULY  ");
            break;
          case 8:
            strcpy(month," AUGUST  ");
            break;
          case 9:
            strcpy(month,"SEPTEMBER");
            break;
          case 10:
            strcpy(month," OCTOBER ");
            break;
          case 11:
            strcpy(month," NOVEMBER");
            break;
          case 12:
            strcpy(month," DECEMBER");
            break;
        }
if (z==1) /*if have an interupt data,it will fill the interupt data in table*/
  printf("      | %2d %-9s 19%d | %-2d:00 |   %d   |   %d   |   %d \n",
        date[data_in[3]][0],month,date[data_in[3]][2],data_in[3],data[data_in[3]][0],
        data[data_in[3]][1],data[data_in[3]][2]);
if (z==2) /*if have an interupt data,it will fill that data in data1 graph*/
  WR_gph(data_in[3],data[data_in[3]][0],8);
if (z==3) /*if have an interupt data,it will fill that data in data2 graph*/
  WR_gph(data_in[3],data[data_in[3]][1],8);
if (z==4) /*if have an interupt data,it will fill that data in data3 graph*/
  WR_gph(data_in[3],data[data_in[3]][2],8);
}
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        return 0;
}
void Table(int comp,int j,int data[][3],int date[][3]) /*Table funtion*/
{       int k;
        char month[9];
        printf("\n");
        window(1,1,80,1);
        textattr(4+(7<<4));
        cprintf("                                                ");
        cprintf("                ");
        cprintf("                                                TELEMETRY DATA                ");
        cprintf("              ");
        printf("
```

DATE	TIME	TEMPERATURE	PRESSURE	VOLTAGE

```
        \n");for (k=(comp-j);(k<=comp);k++)
        {
        switch(date[k][1])
        {
                        case 1:
                          strcpy(month," JANUARY ");
                          break;
                        case 2:
                          strcpy(month," FEBRUARY");
                          break;
                        case 3:
                          strcpy(month,"  MARCH  ");
                          break;
                        case 4:
                          strcpy(month,"  APRIL  ");
                          break;
                        case 5:
                          strcpy(month,"   MAY   ");
                          break;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                case 6:
                    strcpy(month," JUNE   ");
                    break;
                case 7:
                    strcpy(month," JULY   ");
                    break;
                case 8:
                    strcpy(month," AUGUST  ");
                    break;
                case 9:
                    strcpy(month,"SEPTEMBER");
                    break;
                case 10:
                    strcpy(month," OCTOBER ");
                    break;
                case 11:
                    strcpy(month," NOVEMBER");
                    break;
                case 12:
                    strcpy(month," DECEMBER");
                    break;
            }

        printf("    | %2d %-9s 19%d | %-2d:00 |    %d   |   %d   |   %d  \n",
                date[k][0],month,date[k][2],k,data[k][0],
                data[k][1],data[k][2]);
        }
}
void Graph(int comp,int j,int data[][3])  /*show data1 Graph*/
{
    int dy,xmax,ymax,intrvaly,k,intrvalx,l;
    Initialize_Graphics_Mode();
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    BKCOLOR(7);
    xmax = getmaxx();
    ymax = getmaxy();
    intrvalx = (xmax-75) / 24;
    intrvaly = (ymax-60) / 20;
    TEMP_axis(xmax,ymax);
    l=(comp-j);
    moveto((45+(intrvalx*l)),ymax-((data[l][0]*4)+30));
    for (k=(l+1);(k<=comp);k++)
    {
          dy=data[k][0];
          WR_gph(k,dy,8);
     }
    /* clean up */
}
void Graph1(int comp,int j,int data[][3]) /*show data2 Graph*/
{
    int dy,xmax,ymax,intrvaly,k,intrvalx,l;

    Initialize_Graphics_Mode();
    BKCOLOR(7);
    xmax = getmaxx();
    ymax = getmaxy();
    intrvalx = (xmax-75) / 24;
    intrvaly = (ymax-60) / 20;
    TEMP_axis1(xmax,ymax);
    l=(comp-j);
    moveto((45+(intrvalx*l)),ymax-((data[l][1]*4)+30));
    for (k=(l+1);(k<=comp);k++)
    {
          dy=data[k][1];
          WR_gph(k,dy,8);
     }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
   /* clean up */
}
void Graph2(int comp,int j,int data[][3]) /*show data3 Graph*/
{
   int dy,xmax,ymax,intrvaly,k,intrvalx,l;

   Initialize_Graphics_Mode();
   BKCOLOR(7);
   xmax = getmaxx();
   ymax = getmaxy();
   intrvalx = (xmax-75) / 24;
   intrvaly = (ymax-60) / 20;
   TEMP_axis2(xmax,ymax);
   l=(comp-j);
   moveto((45+(intrvalx*l)),ymax-((data[l][2]*4)+30));
   for (k=(l+1);(k<=comp);k++)
   {
         dy=data[k][2];
         WR_gph(k,dy,8);
   }
   /* clean up */
}

void Initialize_Graphics_Mode(void)
{
   int gdriver = DETECT, gmode, errorcode;

   /* initialize graphics and local variables */
   initgraph(&gdriver, &gmode, "");

   /* read result of initialization */
   errorcode = graphresult();
   /* an error occurred */
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    if (errorcode != grOk)
    {
        printf("Graphics error: %s\n", grapherrormsg(errorcode));
        printf("Press any key to halt:");
        getch();
        exit(1);
    }
}
void TEMP_axis(int xmax,int ymax)  /*make the axis of data1 Graph*/
{
    int xi,i,intrvalx,intrvaly,number,numm;
    char string[25];
    int size=1,style=1;
    /* draw a diagonal line */
    intrvalx = (xmax-75) / 24;   /*the value of interval of dot in x_axis*/
    intrvaly = (ymax-60) / 20;   /*the value of interval of dot in y_axis*/
    setcolor(9);
    line(45, 30,45,ymax-30);     /*draw the axis of x_axis*/
    line(45,ymax-30,xmax-30,ymax-30);   /*draw the axis of y_axis*/
    for (i=1;i<=23;i++)          /*draw the line of interval in x_axis*/
    {
        number = i;
        line((45+(intrvalx*i)),ymax-28,45+((intrvalx*i)),ymax-32);
        settextstyle(style, 1 ,0);
        itoa(number, string, 10);
        outtextxy((30+(intrvalx*(i+1))),ymax-23,string);
    }
    xi=20;
    for (i=0;i<=20;i++)          /*draw the line of interval in y_axis*/
    {
        numm = xi*5;
        xi--;
        line(43,(30+(intrvaly*i)),47,(30+(intrvaly*i)));
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        settextstyle(style, 0 ,size);
        itoa(numm, string, 10);
        outtextxy(15,(31+(intrvaly*(i+1))),string);
    }
    settextjustify(CENTER_TEXT, CENTER_TEXT);
    /* select the text style */
    settextstyle(style, 0 , size);
    /* output a message */
    setcolor(20);
    outtextxy(65,25,"TEMPERATURE(K)");
    settextstyle(style,0,size);
    outtextxy(xmax-25,ymax-15,"HOUR");
}
void TEMP_axis1(int xmax,int ymax)   /*make the axis of data2 Graph*/
{
    int xi,i,intrvalx,intrvaly,number,numm;
    char string[25];
    int size=1,style=1;
    /* draw a diagonal line */
    intrvalx = (xmax-75) / 24;
    intrvaly = (ymax-60) / 20;
    setcolor(9);
    line(45, 30,45,ymax-30);
    line(45,ymax-30,xmax-30,ymax-30);
    for (i=1;i<=23;i++)
    {
        number = i;
        line((45+(intrvalx*i)),ymax-28,45+((intrvalx*i)),ymax-32);
        settextstyle(style, 1 ,0);
        itoa(number, string, 10);
        outtextxy((30+(intrvalx*(i+1))),ymax-23,string);
    }
    xi=20;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    for (i=0;i<=20;i++)
    {
        numm = xi*5;
        xi--;
        line(43,(30+(intrvaly*i)),47,(30+(intrvaly*i)));
        settextstyle(style, 0 ,size);
        itoa(numm, string, 10);
        outtextxy(15,(31+(intrvaly*(i+1))),string);
    }
    settextjustify(CENTER_TEXT, CENTER_TEXT);
    /* select the text style */
    settextstyle(style, 0 , size);
    /* output a message */
    setcolor(20);
    outtextxy(110,21,"2");
    outtextxy(65,25,"PRESSURE(N/M )");
    settextstyle(style,0,size);
    outtextxy(xmax-25,ymax-15,"HOUR");
}
void TEMP_axis2(int xmax,int ymax)   /*make the axis of data3 Graph*/
{
    int xi,i,intrvalx,intrvaly,number,numm;
    char string[25];
    int size=1,style=1;
    /* draw a diagonal line */
    intrvalx = (xmax-75) / 24;
    intrvaly = (ymax-60) / 20;
    setcolor(9);
    line(45, 30,45,ymax-30);
    line(45,ymax-30,xmax-30,ymax-30);
    for (i=1;i<=23;i++)
    {
        number = i;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        line((45+(intrvalx*i)),ymax-28,45+((intrvalx*i)),ymax-32);
        settextstyle(style, 1 ,0);
        itoa(number, string, 10);
        outtextxy((30+(intrvalx*(i+1))),ymax-23,string);
    }
    xi=20;
    for (i=0;i<=20;i++)
    {
        numm = xi*5;
        xi--;
        line(43,(30+(intrvaly*i)),47,(30+(intrvaly*i)));
        settextstyle(style, 0 ,size);
        itoa(numm, string, 10);
        outtextxy(15,(31+(intrvaly*(i+1))),string);
    }
    settextjustify(CENTER_TEXT, CENTER_TEXT);
    /* select the text style */
    settextstyle(style, 0 , size);
    /* output a message */
    setcolor(20);
    outtextxy(65,25,"VOLTAGE(V)");
    settextstyle(style,0,size);
    outtextxy(xmax-25,ymax-15,"HOUR");
}
void BKCOLOR(int color)
{
    /* clear the screen */
    cleardevice();
    /* select a new background color */
    setbkcolor(color);
    /* output a messsage */
}
void WR_gph(int xdev,int dy,int colour)   /*write data of graph*/
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
{       int wr_y,wr_x;
        int xmax,ymax,intrvalx,intrvaly;

        xmax = getmaxx();
        ymax = getmaxy();
        intrvalx = (xmax-75)/24;
        intrvaly = (ymax-60)/20;
        setcolor(colour);
        wr_y=((ymax-(intrvaly*(dy/5)))-30);
        wr_x=(45+(intrvalx*xdev));
        lineto(wr_x,wr_y);
}
void ATTRIBUTE(void)        /*show hot key*/
{
        window(1,25,80,25);
        textattr(4+(7<<4));
        cprintf("                                    ");
        cprintf("                    ");
        cprintf("   F1");
        textcolor(BLACK);
        cprintf(" TABLE");
        textcolor(RED);
        cprintf("   F2");
        textcolor(BLACK);
        cprintf(" TEMPERATURE");

        textcolor(RED);
        cprintf("   F3");
        textcolor(BLACK);
        cprintf(" PRESSURE");
        textcolor(RED);
        cprintf("   F4");
        textcolor(BLACK);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        cprintf(" VOLTAGE ");
        textcolor(RED);
        cprintf("   F5");
        textcolor(BLACK);
        cprintf(" BACKWARD INFO. ");
        window(1,1,80,25);
}
void Title(void)        /*write main menu*/
{
    clrscr();
          TextWindow(1,1,79,25,LIGHTRED,LIGHTGRAY);
          TextWindow(26,8,56,17,BLACK,BLACK);
          TextWindow(25,7,55,16,YELLOW,BLUE);
    textcolor(WHITE);
    gotoxy(29,7);
    cprintf(" TELEMETRY SYSTEM MENU ");
    textcolor(YELLOW);
    gotoxy(32,9);
    cprintf("F1-TABLE");
    gotoxy(32,10);
    cprintf("F2-TEMPERATURE");
    gotoxy(32,11);
    cprintf("F3-PRESSURE");
    gotoxy(32,12);
    cprintf("F4-VOLTAGE");
    gotoxy(32,13);
          cprintf("F5-BACKWARD INFO.");
          gotoxy(32,14);
          cprintf("Q -Quit");
}
void Backinfo(void)     /*show menu of back information*/
{
          char det,temp[40];
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
int s;
BACK:
clrscr();
TextWindow(1,1,79,24,LIGHTRED,LIGHTGRAY);
TextWindow(26,8,56,15,BLACK,BLACK);
TextWindow(25,7,55,14,YELLOW,BLUE);
textcolor(WHITE);
gotoxy(33,7);
cprintf(" BACKWARD INFO.");
textcolor(YELLOW);
gotoxy(27,9);
cprintf("Select Time");
gotoxy(27,10);
cprintf("Date:");
gotoxy(27,11);
cprintf("Month:");
gotoxy(27,12);
cprintf("Year:    19");
gotoxy(38,10);
gets(temp);
s = atoi(temp);
if (!(s>=1 & s<100))
{
 goto BACK;
}
in_dte[0] = s;

gotoxy(38,11);
gets(temp);
s = atoi(temp);
if(!(s>=1 & s<100))
{
 goto BACK;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        }
        in_dte[1] = s;

        gotoxy(38,12);
        gets(temp);
        s = atoi(temp);
        if(!(s>=1 & s<100))
        {
         goto BACK;
        }
        in_dte[2] = s;

        gotoxy(1,24);
        textattr(1+(2<<4));
        cprintf("                    o = O.K. OR  c = Cancel");
        cprintf("                    ");

        while(!kbhit());
        det=getch();
        if(det !='o')       /* C */
        Cancel();
        if(det =='o')       /* O  */
        R_File(in_dte);
}
/*write background of menu*/
void TextWindow(int x1,int y1,int x2,int y2,int fcolor,int bcolor)
{
    int x,y;
    textcolor(fcolor);
    textbackground(bcolor);
    for(y=y1+1;y<y2;y++)
    {
     for(x=x1+1;x<x2;x++)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    {
        gotoxy(x,y);
        putch(0x20);
    }
    gotoxy(x1,y);putch(0xb3);
    gotoxy(x2,y);putch(0xb3);
   }
  gotoxy(x1+1,y1);
  for(x=x1+1;x<x2;x++)putch(0xc4);
  gotoxy(x1+1,y2);
  for(x=x1+1;x<x2;x++)putch(0xc4);
  gotoxy(x1,y1); putch(0xda);
  gotoxy(x2,y1); putch(0xbf);
  gotoxy(x1,y2); putch(0xc0);
  gotoxy(x2,y2); putch(0xd9);
}
void O_File(int dta[][3],int dte[][3])  /*open file to write data which full 24 hour*/
{
    FILE *fp;
    char d[3],m[3],y[3],name[9],file_name[20];
    int e;
    itoa(dte[data_in[3]][0],d,10);    /*name back data file*/
    itoa(dte[data_in[3]][1],m,10);
    itoa(dte[data_in[3]][2],y,10);
    strcpy(name,d);
    strcat(name,"m");
    strcat(name,m);
    strcat(name,"y");
    strcat(name,y);
        sprintf(file_name,"a:\\%s.tel",name);
   /*************************************************************/
    while((fp = fopen(file_name,"wb")) == NULL)
   {
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        printf("\a");
        ETO();
    }
    for(e=0;e<=23;e++)
    {
        s_dt.data1 = dta[e][0];
        s_dt.data2 = dta[e][1];
        s_dt.data3 = dta[e][2];  /* OPEN FILE TO WRITE */
        fwrite(&s_dt,sizeof s_dt,1,fp);

        if(ferror(fp))
        {
          printf("\007");
          printf("\007");
          ERROR_W();
        }
     }
    fclose(fp);
}
void R_File(int idte[3]) /*read file to write back data*/
{
      FILE *fp;
      char d[3],m[3],y[3],name[9],file_name[20];
      int k;

      itoa(idte[0],d,10);
      itoa(idte[1],m,10);
      itoa(idte[2],y,10);
      strcpy(name,d);
      strcat(name,"m");          /* TRANFER DATE TO FILE NAME */
      strcat(name,m);
      strcat(name,"y");
      strcat(name,y);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        sprintf(file_name,"a:\\%s.tel",name);
   printf("%s",file_name);

if((fp = fopen(file_name,"rb")) == NULL)
{
    printf("\a");
    printf("\007");
    printf("\007");
    ERROR_R();
    Cancel();
    goto ssend;
}
k = 0;
while(fread(&s_dt,sizeof s_dt,1,fp) == 1)
 {
     if(ferror(fp))
     {
         printf("\007");
         printf("\007");
         ERROR_R();
      }
         dt[k][0] =  s_dt.data1;
         dt[k][1] =  s_dt.data2;
         dt[k][2] =  s_dt.data3;
         k++;
  }
        clrscr();
        printf("                                %2d/%2d/19%d \n",idte[0],idte[1],idte[2]);
        gotoxy(1,3);
        printf("
```

TIME	TEMPERATURE	PRESSURE	VOLTAGE

```
          \n");
    for (k=0;k<=11;k++)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    {
        if (k<=9)
        printf(" | %d:00 |    %-2d    |  %-2d  |  %-2d  \n",k,dt[k][0],dt[k][1],dt[k][2]);
        else
        printf(" | %d:00|    %-2d    |  %-2d  |  %-2d  \n",k,dt[k][0],dt[k][1],dt[k][2]);
    }
        gotoxy(40,3);
        printf("
```

TIME	TEMPERATURE	PRESSURE	VOLTAGE

```
        \n");
    for (k=12;k<=23;k++)
    {  gotoxy(40,k-6);
        printf("    | %d:00|    %-2d    |  %-2d  |  %-2d  \n",k,dt[k][0],dt[k][1],dt[k][2]);
    }
  ssend:
  fclose(fp);
}
void ERROR_R(void)   /*show menu of error read file if press q,it will quit to main menu*/
  {
        char ex;
        clrscr();
        Error_To_read(1,1,79,24,LIGHTRED,LIGHTGRAY);
        Error_To_read(26,10,56,16,BLACK,BLACK);
        Error_To_read(25,9,55,15,YELLOW,RED);
        ex = getch();
        while(ex != 'q');
      return(0);

  }
  void Error_To_read(int p1,int q1,int p2,int q2,int fcolor,int bcolor)

  {
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
int p,q;
textcolor(fcolor);
textbackground(bcolor);
for(q=q1+1;q<q2;q++)
{
 for(p=p1+1;p<p2;p++)
 {
      gotoxy(p,q);
      putch(0x20);
 }
 gotoxy(p1,q);putch(0xb3);
 gotoxy(p2,q);putch(0xb3);
}
gotoxy(p1+1,q1);
for(p=p1+1;p<p2;p++)putch(0xc4);
gotoxy(p1+1,q2);
for(p=p1+1;p<p2;p++)putch(0xc4);
gotoxy(p1,q1); putch(0xda);
gotoxy(p2,q1); putch(0xbf);
gotoxy(p1,q2); putch(0xc0);
gotoxy(p2,q2); putch(0xd9);
textcolor(YELLOW|BLINK);
gotoxy(38,10);
cprintf("ERROR!");
gotoxy(31,12);
cprintf(" Can not read data.");
gotoxy(36,14);
cprintf("(q = Quit)");
}
void ERROR_W(void)    /*show of error writing file*/
{
      char  n;
      clrscr();
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        Error_To_write(1,1,79,24,LIGHTRED,LIGHTGRAY);
        Error_To_write(26,10,56,16,BLACK,BLACK);
        Error_To_write(25,9,55,15,YELLOW,RED);
        n = getch();
        while(n != 'q')
        n = getch();
        QuitSerialComm();
        exit(1);
    }


void Error_To_write(int l1,int m1,int l2,int m2,int fcolor,int bcolor)
    {

     int l,m;
     textcolor(fcolor);
     textbackground(bcolor);
     for(m=m1+1;m<m2;m++)
     {
      for(l=l1+1;l<l2;l++)
      {
           gotoxy(l,m);
           putch(0x20);
      }
      gotoxy(l1,m);putch(0xb3);
      gotoxy(l2,m);putch(0xb3);
     }
    gotoxy(l1+1,m1);
    for(l=l1+1;l<l2;l++)putch(0xc4);
    gotoxy(l1+1,m2);
    for(l=l1+1;l<l2;l++)putch(0xc4);
    gotoxy(l1,m1); putch(0xda);
    gotoxy(l2,m1); putch(0xbf);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    gotoxy(l1,m2); putch(0xc0);
    gotoxy(l2,m2); putch(0xd9);
    textcolor(YELLOW|BLINK);
    gotoxy(38,10);
    cprintf("ERROR!");
    gotoxy(27,12);
    cprintf(" Can not save data in disk.");
    gotoxy(36,14);
    cprintf("(q = Quit)");
}
void ETO(void)  /*if press q,it will back to main menu*/
{
        char ex;
        clrscr();
        ETOP(1,1,79,24,LIGHTRED,LIGHTGRAY);
        ETOP(26,10,56,16,BLACK,BLACK);
        ETOP(25,9,55,15,YELLOW,RED);
        ex = getch();
        while(ex != 'q')
        ex = getch();
        exit(1);
}
   /*show menu of press q,it will back to main menu*/
   void ETOP(int t1,int u1,int t2,int u2,int fcolor,int bcolor)
   {
      int t,u;
      textcolor(fcolor);
      textbackground(bcolor);
      for(u=u1+1;u<u2;u++)
     {
        for(t=t1+1;t<t2;t++)
        {
           gotoxy(t,u);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            putch(0x20);
          }
          gotoxy(t1,u);putch(0xb3);
          gotoxy(t2,u);putch(0xb3);
      }

     gotoxy(t1+1,u1);
     for(t=t1+1;t<t2;t++)putch(0xc4);
     gotoxy(t1+1,u2);
      for(t=t1+1;t<t2;t++)putch(0xc4);
      gotoxy(t1,u1); putch(0xda);
      gotoxy(t2,u1); putch(0xbf);
      gotoxy(t1,u2); putch(0xc0);
      gotoxy(t2,u2); putch(0xd9);
      textcolor(YELLOW);
      gotoxy(38,10);
      cprintf("ERROR!");
      gotoxy(31,12);
      cprintf(" Can not open file.");
      gotoxy(36,14);
      cprintf("(q = Quit)");
   }
void Cancel()  /*function to cancel open file*/
{
   char dete;
   gotoxy(1,24);
   textattr(1+(2<<4));
   cprintf(" Do you want to read any backward information ?");
   cprintf(" y = Yes , n = No                ");
 while(!kbhit());
 dete=getch();
 if(dete == 'y')
 {/*  getdate()   */
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        closegraph();
        window(1,1,80,25);
        clrscr();
        Backinfo();
    }
    if(dete != 'y')
    {
      closegraph();
      window(1,1,80,25);
      Title();
    }
    /* option() */ ;
}
/* ************************************************************************

    Procedure : InitSerialComm(...) -

      Parameters :

          Name        Available selections
          ========================================================================
          comno       - COM1,COM2,COM3 or COM4
          baudrate    - B1200, B2400, B4800, B9600 or B19200
          databit     - D7 or D8
          stopbit     - S1 or S2
          parity      - PNONE, PODD or PEVEN
************************************************************************ */
void  InitSerialComm(void) /*function initial serial port*/
{
  char low_divisor,high_divisor,comm_spec;
  int a;
  low_divisor = baud;
  high_divisor = 0;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
comm_spec = Data | stop | pari;
outportb( CommAddr + LCR , comm_spec | 0x80 );
outportb( CommAddr + DLL , low_divisor      );
outportb( CommAddr + DLH , high_divisor     );
outportb( CommAddr + LCR , comm_spec & 0x7F );
outportb( CommAddr + IER , 1                ); /* enable int : Rx Ready */
outportb( CommAddr + MCR , 0x0B             ); /* OUT2(3),RTS(1) */
for(a=0;a < 6 ;a++)
  inportb(CommAddr + a);  /*Clear any pending interrupt*/
                OLD = getvect(INTV);
                setvect(INTV,STORE);
                outportb(IMR,inportb(IMR) & (~IRQ4FLAG));
}
void QuitSerialComm(void)
{
                setvect( INTV , STORE );
                outportb( IMR , inportb(IMR) | IRQ4FLAG );
}
void interrupt STORE(void)  /*function if have interrupt*/
{
 data_in[h] = inportb(CommAddr);
 h++;
 outportb( 0x20 , EOI );
 OLD();
}

void transfer(void)        /*function save interrupt data in array 2 dimension*/
{
            for (i=0;i<=3;i++)
            {
                qu = div(data_in[i],16);
                data_in[i]=data_in[i]-(qu.quot*6);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    }
    data[data_in[3]][0]=data_in[4];
    data[data_in[3]][1]=data_in[5];
    data[data_in[3]][2]=data_in[6];
    date[data_in[3]][0]=data_in[2]; /*DAY*/
    date[data_in[3]][1]=data_in[1]; /*MONTH*/
    date[data_in[3]][2]=data_in[0]; /*YEAR*/
    j++;
   if (data_in[3]==23)
   {
    j=0;
    O_File(data,date);
   }
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

MOTOROLA
SEMICONDUCTOR
TECHNICAL DATA

MC145450

Advance Information

CMOS

1200 BAUD
FSK MODEM

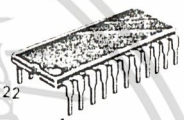
2

1200 BAUD FSK MODEM

The MC145450 is a silicon-gate CMOS frequency shift keying (FSK) modem intended for use in Bell 202 and CCITT V.23 applications. Features of the device include:

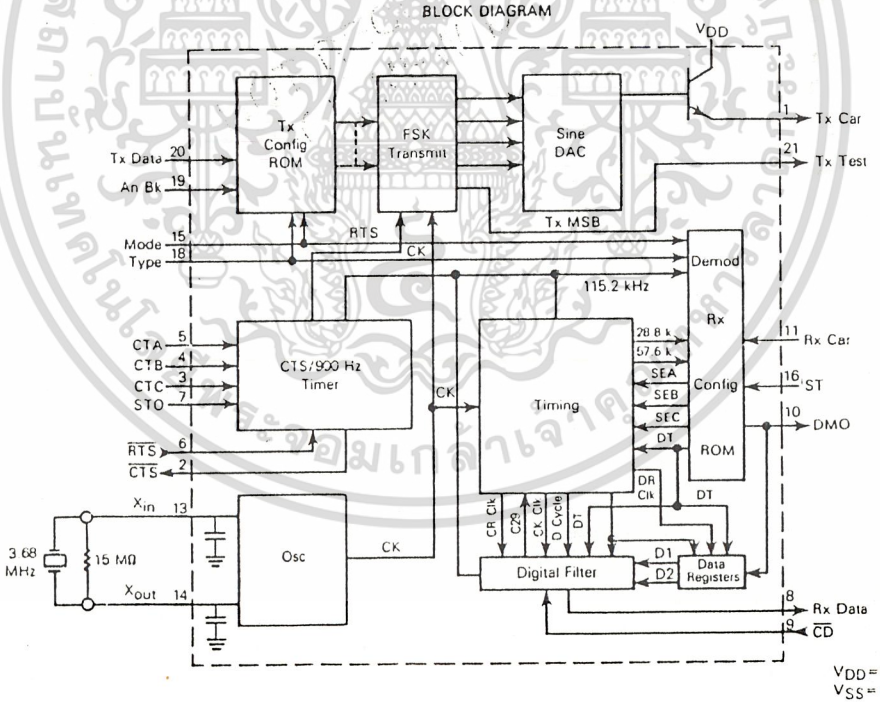
- Bell 202 Compatible 0 to 1800 Baud Main Channel
- 0 to 150 Baud Reverse Channel
- CCITT V.23 Mode 2 Compatible 0 to 1800 Main Channel
- CCITT V.23 0 to 75 Baud Compatible Reverse Channel
- TTL Compatible
- Eight Selectable $\overline{RTS}$-$\overline{CTS}$ Delay Options
- Soft Turn-Off Capability
- Answer Back Tone Generator (US and CCITT Tones)
- Carrier Detect Input
- 22 Pin Package

L SUFFIX
CERAMIC PACKAGE
CASE 736



P SUFFIX
PLASTIC PACKAGE
CASE 708

MC145450 1200 BAUD FSK MODEM
BLOCK DIAGRAM



This document contains information on a new product. Specifications and information herein are subject to change without notice.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

MC145450

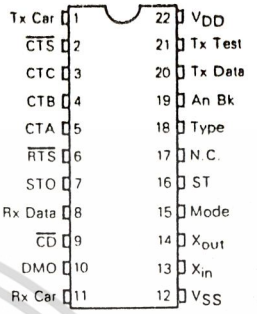
ABSOLUTE MAXIMUM RATINGS

Rating	Symbol	Value	Unit
DC Supply Voltage	V_{DD}	10	V
Input Voltages, All Inputs	V_{in}	$V_{SS}-0.5$ to $V_{DD}+0.5$	V
DC Current Drain per Pin Pin 3-6, 9, 15, 16, 18, 19, 20 Pins 2, 8	I_{out}	 10 35	mA
Operating Temperature Range	T_A	0 to +70	°C
Storage Temperature Range	T_{stg}	−55 to +150	°C

RECOMMENDED OPERATING CONDITIONS

Parameter	Symbol	Min	Typ	Max	Unit
DC Supply Voltage	$V_{DD}-V_{SS}$	4.5	5.0	6.5	V

PIN ASSIGNMENTS



DC ELECTRICAL CHARACTERISTICS ($V_{DD}=5.0$ V ± 5%, $V_{SS}=0$, $T_A=0$ to 70°C)

Characteristics	Symbol	Min	Typ	Max	Unit
Input High Voltage	V_{IH}				V
Pins 3-7, 9, 15, 16, 18, 19, 20	–	2.8	–	–	
Pin 13, 11	–	4.0	–	–	
Input Low Voltage	V_{IL}				V
Pins 3-7, 9, 15, 16, 18, 19, 20	–	–	–	0.5	
Pin 13, 11	–	–	–	0.6	
Input Current	I_{in}				μA
All Inputs (V_{IL} 0 V)	–	–	–	−5.0	
All Inputs Except Pins 11, 13, ($V_{IH}>2.8$ V) (Note 1)	–	–	–	600	
Output High Current ($V_{OH}=2.4$ V)	I_{OH}				mA
Pins 2, 8 (Test Load A)		0.75	–	–	
Pins 10, 21 (Test Load B)		0.75	–	–	
Output Low Current ($V_{OL}=0.4$ V)	I_{OL}				mA
Pins 2, 8 (Test Load A)		1.2	–	–	
Pins 10, 21 (Test Load B)		0.6	–	–	
Operating Current	I_{DD}	–	2.5	5	mA
Input Capacitance	C_{in}				pF
All Except Pin 13	–	–	–	12	
Pin 13 (X_{in})	–	–	8	–	
Output Capacitance	C_{out}				pF
All Except Pin 14	–	–	–	12	
Pin 14 (X_{out})	–	–	13	–	
Transmit Audio Signal Level (Pin 1 $R_L=10$ kΩ (Note 2)	–	0.428	0.5	0.578	Vp-p
Total Harmonic Distortion (2nd to 14th) (Note 2)	THD	–	−50	−40	dB

AC ELECTRICAL CHARACTERISTICS ($V_{DD}=5.0$ V ± 5%, $V_{SS}=0$, $T_A=0$ to 70°C)

Characteristics		Symbol	Min	Typ	Max	Unit
Output Rise Time (Test Load A) (Pins 2, 8)		t_r	–	20	100	ns
Output Rise Time (Test Load B) (Pins 10, 14, 21)		t_r	–	20	100	ns
Output Fall Time (Test Load A) (Pins 2, 8)		t_f	–	20	100	ns
Output Fall Time (Test Load B) (Pins 10, 14, 21)		t_f	–	20	100	ns
Input Rise and Fall Times (Except Pin 13)		t_r, t_f	–	–	1000	μs
Delay From $\overline{RTS}$ to $\overline{CTS}$	STO = Low	$t_{d(low)}$	–	1	–	μs
Delay From $\overline{RTS}$ to $\overline{CTS}$	STO = High	$t_{d(high)}$	18.3	–	21.7	ms

NOTES

1. Active pull-up devices are used on these inputs to allow interfacing to TTL devices. The I_{in} specified is a transitional load (not steady state) which is drawn when the input is brought up to 2.8 V until the internal pull-up device has raised the signal to the V_{DD} level.
2. Measured in any mode using HP-3555B dB meter (or equivalent) with 3 kHz flat filtering.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

PIN DESCRIPTIONS

V_{DD}, POSITIVE POWER SUPPLY (PIN 22)
This is nominally 5.0 V.

V_{SS}, NEGATIVE POWER SUPPLY (PIN 12)
This is usually 0 volts.

Tx Car, TRANSMIT CARRIER (PIN 1)
The transmit carrier output is a 16 step digitally-synthesized sine wave with an amplitude of 0.1 V_{DD} (p-p) (± 10%) and offset by a dc bias of 0.5 V_{DD} (± 10%). The output load should be 10 kilohms or greater.

$\overline{CTS}$, CLEAR TO SEND (PIN 2)
The clear to send output goes low in response to a high-to-low translation of $\overline{RTS}$ following a selected delay (see CTA, CTB, CTC pin description). This output goes high immediately after loss of $\overline{RTS}$. During the time following activation of $\overline{RTS}$ and before the activation of $\overline{CTS}$, Tx Data should be held in the mark condition.

CTA, CLEAR TO SEND SELECT A (PIN 5)
CTB, CLEAR TO SEND SELECT B (PIN 4)
CTC, CLEAR TO SEND SELECT C (PIN 3)

For delay times for clear to send delay select inputs, see Table 1.

$\overline{RTS}$, REQUEST TO SEND (PIN 6)
The request to send input controls data transmission from the modulator. A low level enables the modulator output and a high level will disable the modulator. See Figure 1.

STO, SOFT TURN OFF INPUT (PIN 7)
Activation of STO causes a 900 Hz tone to be transmitted and $\overline{CTS}$ to remain active for 20 ms following the loss of $\overline{RTS}$. See Figure 5.

Rx Data, RECEIVE DATA (PIN 8)
The receive data output is the serial data output from the demodulator. Rx Data is clamped high when $\overline{CD}$ is not active.

$\overline{CD}$, CARRIER DETECT (PIN 9)
When carrier detect input is high (1), the Rx Data output will be clamped to a high state. When carrier detect is low (0), Rx Data output demodulates the Rx carrier input signal.

DMO, DEMODULATOR OUTPUT (PIN 10)
The demodulator output is the output of the differential delay detector. It is used for production testing of the demodulator. In normal operation, this pin should be left open.

Rx Car, RECEIVER CARRIER (PIN 11)
The receiver carrier input is the FSK input to the demodulator. This signal should be the hard-limited output of the receive filter, nominally 50%.

X_{in}, OSCILLATIOR INPUT (PIN 13)
X_{out}, OSCILLATOR OUTPUT (PIN 14)
X_{in} should be driven from either an AT-cut crystal or a digital signal source at 3.6864 MHz ± 0.01%. When driven by a crystal, a 15 megohm resisitor should be connected from X_{in} to X_{out} in parallel with the crystal.

MODE (PIN 15)
The mode pin selects the pair of frequencies used during modulation and demodulation. A "0" on this pin selects forward channel operation; i.e. high-speed transmit and low-speed receive. A "1" on this pin selects reverse channel operation; i.e. low-speed transmit and high-speed receive.

ST, SELF TEST (PIN 16)
When a high level is placed on this pin, the demodulator is switched to the modulator frequencies and baud rate (as determined by Mode and Type pins). The modulator should be looped back through the receive filter to the demodulator for self test (echo back).

N.C. NO CONNECTION (PIN 17)
This pin is not bonded internally and should be left open in normal operation.

TYPE (PIN 18)
This pin is used to select Bell 202 type operation and CCITT V.23 operation. When the type input pin is a "1", Bell operation is selected. When the type input pin is a "0", the CCITT standard is selected.

An Bk, ANSWER BACK (PIN 19)
The answer back input causes the answer back tone to be transmitted. The answer back tone is 2025 Hz for the Bell mode and 2100 Hz for the CCITT modes. When a high level is placed on the An Bk input pin, the Tx Car pin will output an answer back tone and $\overline{CTS}$ will go to a high state, regardless of the state of $\overline{RTS}$ (see Figure 1).

Tx Data, TRANSMIT DATA (PIN 20)
The transmit data input is the serial input to the modulator. A high level causes a mark frequency to be transmitted, a low level causes a space frequency to be transmitted.

Tx Test, TRANSMIT TEST (PIN 21)
The transmit test output is a square wave representation of the modulator transmit frequency. It is used for test purposes and should be left open in normal operation.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

FIGURE 1 – An Bk AND $\overline{RTS}$-$\overline{CTS}$ TIMING

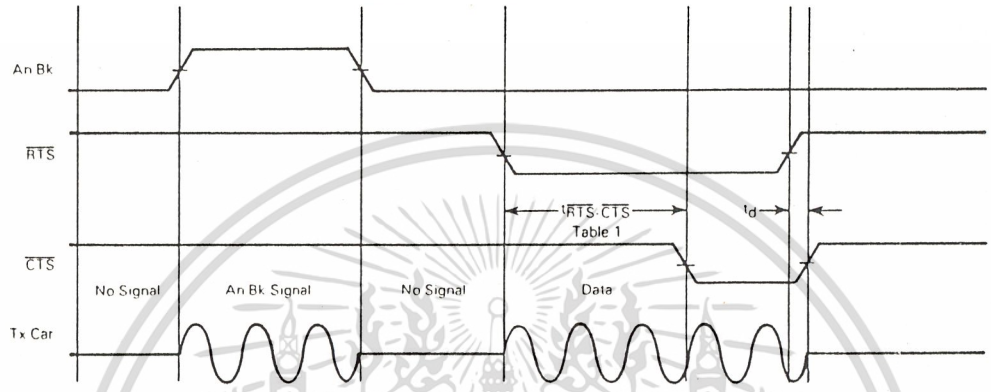


TABLE 1 – $\overline{RTS}$-$\overline{CTS}$ DELAY TIMES

CTC	CTB	CTA	Delay*
0	0	0	0 ms
0	0	1	26.7 ms
0	1	0	40.0 ms
0	1	1	60.0 ms
1	0	0	133.3 ms
1	0	1	213.3 ms
1	1	0	266.7 ms
1	1	1	426.6 ms

* All delays are ± 1.7 ms.

TABLE 2 – OPERATING MODES

Type	Mode	Transmit Data	Transmit Frequency Spec	Transmit Frequency Actual	Answer Back Tone	Application
0	0	0	2100	2099.32	2100	CCITT V.23 75 Baud Receive 1200 Baud Transmit
		1	1300	1299.86		Forward Channel
0	1	0	450	450	2100	CCITT V.23 1200 Baud Receive 75 Baud Transmit
		1	390	390.5		Reverse Channel
1	0	0	2200	2199.52	2025	U.S. 150 Baud Receive 1200 Baud Transmit (Bell 202)
		1	1200	1200		Forward Channel
1	1	0	510	509.73	390	U.S. 1200 Baud Receive (Bell 202) 150 Baud Transmit
		1	390	390.5		Reverse Channel

Data = 0 = Space
= 1 = Mark

* Crystal Frequency = 3.6864 MHz

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

FIGURE 2 — STO TIMING

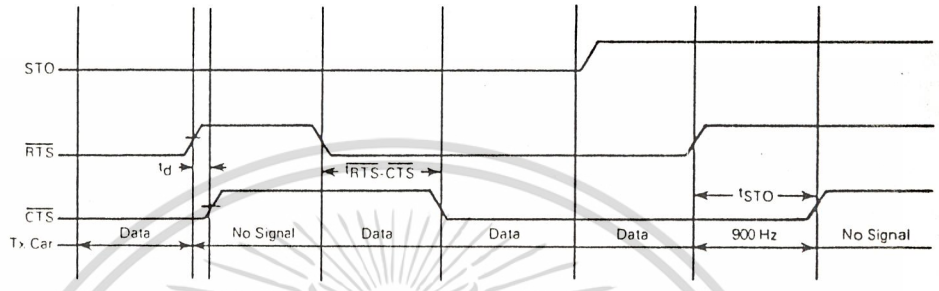
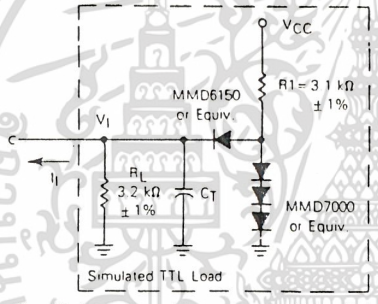
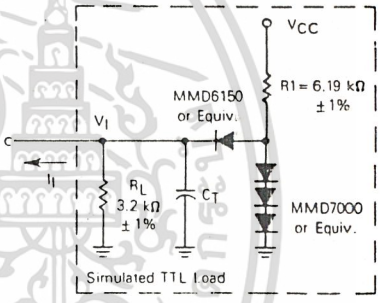


FIGURE 3 — OUTPUT TEST LOAD A



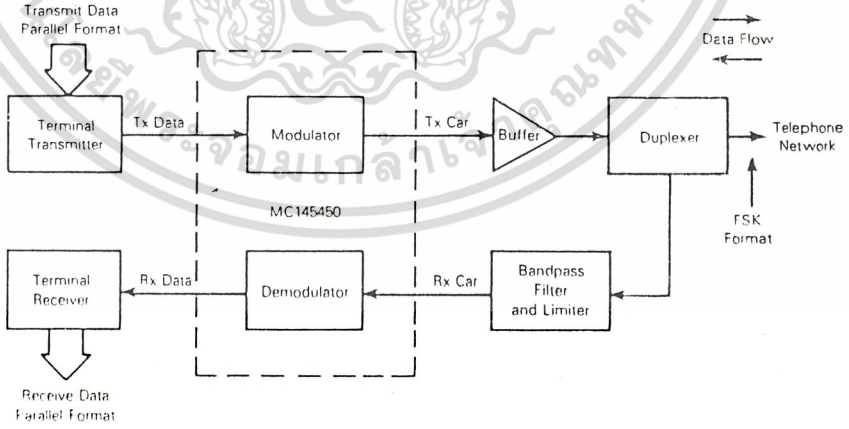
C_T = 20 pF = total parasitic capacitance, which includes probe, wiring, and load capacitances.

FIGURE 4 — OUTPUT TEST LOAD B



C_T = 20 pF = total parasitic capacitance, which includes probe, wiring, and load capacitances.

FIGURE 5 — TYPICAL MEDIUM-SPEED MODEM APPLICATION



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

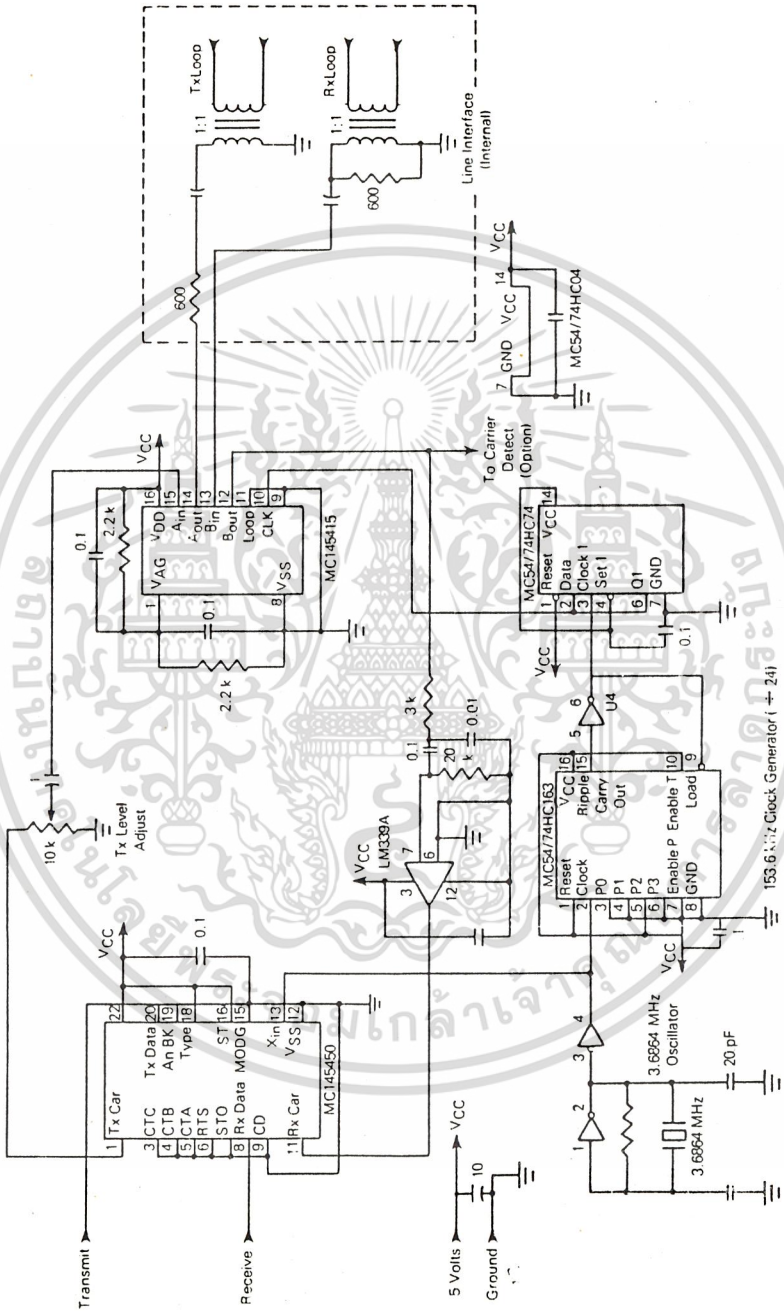
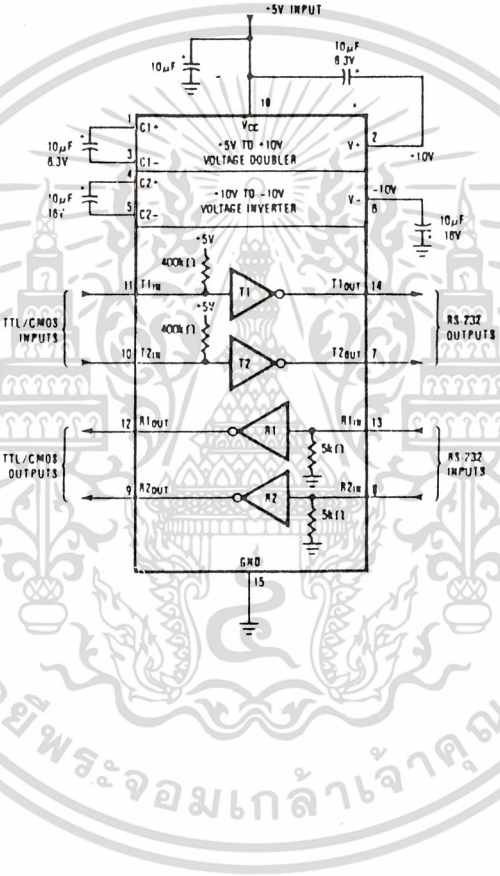


FIGURE 6 — TYPICAL 1200 BAUD 4 WIRE MODEM APPLICATION

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



IC MAX232

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

ขอขอบพระคุณ อาจารย์ณรงค์ เหมกรณ์ และ อาจารย์นิภา ลีลารุจิ ที่ได้ให้คำปรึกษาแนะนำความรู้ต่างๆ เป็นอย่างดี ตลอดจนการให้ยืมอุปกรณ์ต่างๆที่ใช้ในการทำงานซึ่งเป็นประโยชน์อย่างยิ่ง รวมทั้งขอบคุณเพื่อนๆที่ให้ความช่วยเหลือทุกๆคน

จึงขอขอบคุณมา ณ โอกาสนี้ด้วย

อิฏฐพร จันทรสถาพร
อิทธิพงษ์ วงศ์แสนสุข
อิทธิพร เมฆวิบูลย์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เอกสารอ้างอิง

[1] Gary M.Miller"Modern electronic communication" Third edition
: Prentice-Hall International Edition.

[2] Jerry D Gibson "Principle of Digital and Analog Communications"
:Macmillan Publishing Company , a division of Macmillan Inc,1989.

[3] John Uffenbeck "Microcomputers and Microprocessors"
:Prentece-Hall International Editions,1991.

[4] Leon W Couch II "Digital and Analog Communication Systems"
:Macmillan Publishing Company.

[5] B.P.Lathi "modern Digital and Analog Communication Systems"
:Holt,Rinehart and Winston.

[6] K.MIYA "Satellite communications technology"
:KDD Engineering and consulting INC,1982.

[7] ชูชัย ธนสารตั้งเจริญ,ดนัย แสงสุริยศิลป์ ,ทินกร ดุ๊ก ,ธงชัย อุดมกิจจโกศล,
ธานินทร์ พาวรศาสนวงศ์ "การใช้งาน Z80" กรุงเทพฯ
:สำนักพิมพ์ ฟิสิกส์เซ็นเตอร์การพิมพ์.

[8] มนตรี พจนารถกาวัณย์ "การเขียนโปรแกรมคอมพิวเตอร์ด้วยเทอร์โบซี" กรุงเทพฯ
:บริษัท ซีเอ็ดยูเคชั่น จำกัด,2535.

[9] สุชาติ กังวารจิตต์ "หลักการทำงาน เครื่องรับส่งวิทยุและระบบวิทยุสื่อสาร"
:บริษัท ซีเอ็ดยูเคชั่น จำกัด.

[10] ธันวา ศรีประโมง "การเขียนโปรแกรมภาษาซี สำหรับวิศวกรรม"
:สำนักพิมพ์ มหาวิทยาลัยเทคโนโลยีมหานคร,2537.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้