



ปริญญาโททางการศึกษา 2533

ภาควิชา วิศวกรรมโทรคมนาคม

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง ข่ายข้อมูลผ่านระบบวิทยุ

ผู้จัดทำ

1. นาย ชันฉัตร จิตนาวงค์ 90-1058
2. นาย ไชยเชษฐ์ จามรมา 90-1066



(Handwritten signature)

อาจารย์ที่ปรึกษา

(ผศ. ณรงค์ เหมกรณ์)

ขายข้อมูลผ่านระบบวิทยุ

ชั้นใต้ดิน จินตนาเวงค์ 301058

ซอยน้ำจามรนาถ 301066

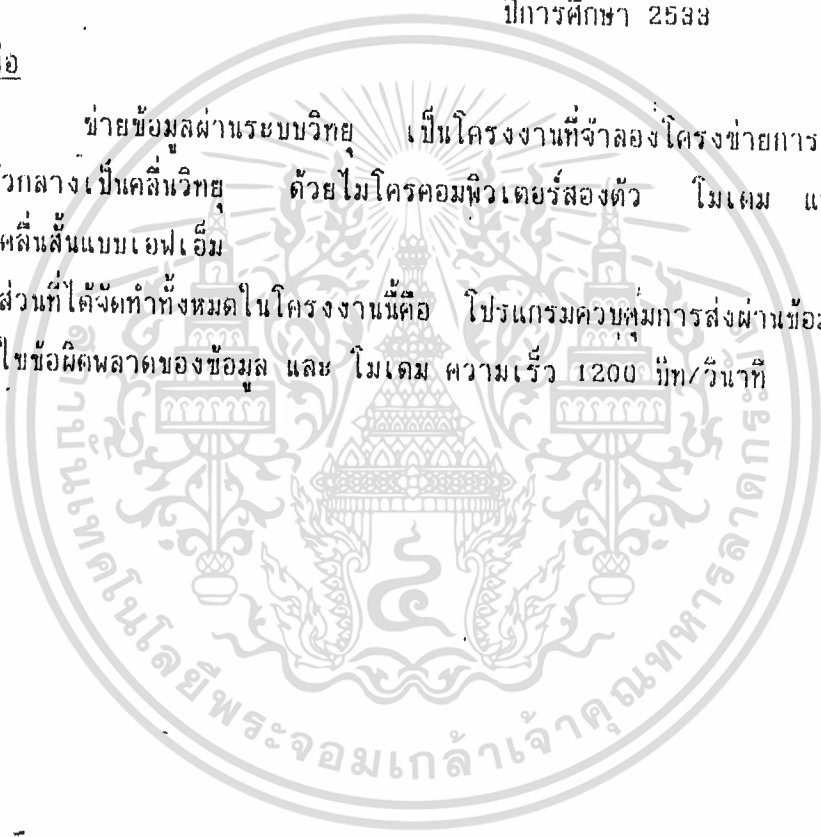
ผศ. ดร. เกรียงศักดิ์ เหมกรรณ อาจารย์ที่ปรึกษา

ปีการศึกษา 2538

บทคัดย่อ

ขายข้อมูลผ่านระบบวิทยุ เป็นโครงการที่จำลองโครงข่ายการสื่อสารข้อมูลแบบมีตัวกลางเป็นคลื่นวิทยุ ด้วยไมโครคอมพิวเตอร์สองตัว โมเด็ม และเครื่องรับส่งวิทยุคลื่นสั้นแบบเอฟเอ็ม

ส่วนที่ได้จัดทำทั้งหมดในโครงการนี้คือ โปรแกรมควบคุมการส่งผ่านข้อมูล โปรแกรมการแก้ไขข้อผิดพลาดของข้อมูล และ โมเด็ม ความเร็ว 1200 บิต/วินาที



DATA LINK BY RADIO

Chanchat Chintanawong 301058

Chaiyanath Chamoraman 301066

Assistant Professor Narong Heamkorn Advisor
1990

Abstract

DATA LINK BY RADIO is the model of data communication network that compose of a couple of micro computer, modems and FM transceiver.

This thesis separate into two section. One is the file transfers program another is the modems 1200 b/s.

สารบัญ

	หน้า
บทที่ 1 บทนำ	1
บทที่ 2 ทฤษฎีและหลักการ	2
2.1 การสื่อสารข้อมูล	2
2.2 การสื่อสารภายในระบบคอมพิวเตอร์	2
2.3 การโอนถ่ายข้อมูล	3
2.4 รูปแบบของการติดต่อสื่อสารแบบอนุกรม	5
2.5 ความเร็วในการถ่ายโอนข้อมูลแบบอนุกรม	6
2.6 การสื่อสารแบบอซิงโครนัส	6
2.7 ข่ายข้อมูลผ่านระบบวิทยุ	7
2.8 ฮาร์ดแวร์ที่เกี่ยวข้องกับการสื่อสารข้อมูล	8
2.9 โมเด็ม	12
2.10 ความผิดพลาดและการแก้ไขความผิดพลาด	15
2.11 เอกซ์โมเด็ม โปรโตคอล (XMODEM PROTOCOL)	17
บทที่ 3 การออกแบบและการสร้าง	20
3.1 การออกแบบโมเด็มที่ใช้กับระบบวิทยุ	20
3.2 การต่อโมเด็มเข้ากับเครื่องรับส่งวิทยุ	20
3.3 การเขียนโปรแกรมโดยใช้ภาษาแอสเซมบลี 8086	22
3.4 การเขียนโปรแกรมส่งข้อมูล	27
3.5 การเขียนโปรแกรมรับข้อมูล	28
บทที่ 4 การทดลองและผลการทดลอง	29
4.1 การทดลองวัดค่าระดับต่าง ๆ ของโมเด็ม	29
4.2 การทดลองโปรแกรมรับ-ส่งข้อมูล	29
บทที่ 5 บทสรุปและวิจารณ์	32
ภาคผนวก ก.	33
ภาคผนวก ข.	70

บทที่ 1

บทนำ

การสื่อสารข้อมุลนั้นมีหลายแบบ ซึ่งในแต่ละแบบจะใช้ในประโยชน์ต่างกันไป ดังนั้นเราจึงจำแนก แบบของการสื่อสารข้อมูลเป็น 2 ประเภทคือ

1. แบบ ซิงโครนัส (synchronous)
2. แบบ อะซิงโครนัส (asynchronous)

และชนิดของการส่งผ่านข้อมูล แบ่งได้เป็น

1. การส่งผ่านข้อมูลแบบอนุกรม (serial data communication)
2. การส่งผ่านข้อมูลแบบขนาน (parallel data communication)

ในโครงการนี้ได้เลือกใช้ การสื่อสารข้อมูลแบบซิงโครนัสและการส่งผ่านข้อมูลแบบอนุกรม ด้วยเหตุผลดังต่อไปนี้

1. การสื่อสารแบบ อะซิงโครนัส ไม่จำเป็นต้องอาศัยสัญญาณนาฬิกา ที่ตรงกัน และการกำหนด โปรโตคอล (protocol) ก็ซับซ้อนน้อยกว่าแบบ ซิงโครนัส
2. การส่งผ่านข้อมูลแบบอนุกรม ใช้ช่องสัญญาณน้อยกว่า ถึงแม้ว่าการส่งผ่านข้อมูลชนิดนี้ จะช้ากว่าแบบขนาน แต่อัตราการผิดพลาดของข้อมูลน้อยกว่าและ สามารถตรวจสอบความผิดพลาดได้ง่ายกว่า

การต่ออุปกรณ์ปลายทางเข้าด้วยกันเป็นข่ายข้อมูลมีหลายวิธี เช่น การต่อแบบ โมเด็ม (modem) การต่อแบบใช้ โมเด็ม ผ่านชุมสายโทรศัพท์ การต่อแบบใช้ โมเด็ม ผ่านข่ายสายเส้นใยแสง และการต่อแบบใช้ โมเด็ม ผ่านระบบวิทยุ แต่ละแบบก็จะมีข้อดีต่าง ๆ กันไป

การต่อแบบ โมเด็ม ใช้สำหรับการโอนข้อมูลระหว่างคอมพิวเตอร์สองเครื่องที่มีระยะทางไกล ๆ (ไม่เกิน 50 ฟุต) ส่วนการต่อผ่านชุมสายโทรศัพท์ นี้เป็นแบบปกติที่ใช้กันแพร่หลาย และในอนาคตก็จะมีข่ายสายของเส้นใยแสงในการสื่อสารข้อมูลด้วย

ในกรณีที่ต้องการข่ายโทรศัพท์ไม่สามารถวางไปถึง หรือเป็นสถานีชั่วคราวที่ไม่คุ้มกับการวางสายโทรศัพท์ การใช้ระบบวิทยุมาแทนที่จะสามารถตัดปัญหาเหล่านั้นได้.

บทที่ 2

ทฤษฎีและหลักการ

2.1 การสื่อสารข้อมูล

การสื่อสารข้อมูลหมายถึง การแลกเปลี่ยนข่าวสารระหว่างอุปกรณ์ปลายทาง โดยอาศัยสายส่ง หรือตัวกลางอื่นๆ โดยที่ลักษณะของข้อมูลเป็นรหัสเลขฐานสอง และการเข้ารหัสต้องเป็นมาตรฐาน มาตรฐานโดยทั่วไปที่ใช้กันอย่างแพร่หลายคือ รหัส แอสกี (ASCII) ซึ่งเป็นรหัส 7 บิต ข้อมูลที่เข้ารหัส แอสกีในแต่ละไบต์จะมี 8 บิต โดยที่อีกบิต จะเป็นบิตพาริตี

2.1.1 รหัสแอสกี (ASCII)

ในการสื่อสารข้อมูล ตัวอักษร ตัวเลข อักขระต่าง ๆ จะถูกแทนด้วยรหัสแอสกี ค่าของรหัสมีดังตารางที่ 2.1

ตารางที่ 2.1 แสดงรหัสแอสกี

7	6	5	4	3	2	1	
x	x	x	0	0	0	0	NUL DLE SP 0 @ P
x	x	x	0	0	0	1	SOH DC1 ! 1 A Q
x	x	x	0	0	1	0	STX DC2 " 2 B R
x	x	x	0	1	1	1	ETX DC3 # 3 C S
x	x	x	0	1	0	0	EOT DC4 \$ 4 D T
x	x	x	0	1	0	1	ENQ NAK % 5 E U
x	x	x	0	1	1	0	ACK SYN & 6 F V
x	x	x	0	1	1	1	BEL ETB ' 7 G W
x	x	x	1	0	0	0	BS CAN (8 H X
x	x	x	1	0	0	1	HT EM , 9 I Y
x	x	x	1	0	1	0	LF SUB * : J Z
x	x	x	1	0	1	1	VT ESC + ; K [
x	x	x	1	1	0	0	FF FS , < L \
x	x	x	1	1	0	1	CR GS - = M]
x	x	x	1	1	1	0	SO RS . > N ^
x	x	x	1	1	1	1	SI US / ? O _ o DEL

2.2 การสื่อสารภายในระบบคอมพิวเตอร์

โดยทั่วไปไมโครคอมพิวเตอร์ ประกอบด้วยอุปกรณ์เรือนอกหลายแบบ แต่ที่จำเป็นทั้งเราพบเห็นอยู่บ่อยๆ ได้แก่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. แป้นพิมพ์
2. จอภาพ
3. เครื่องพิมพ์
4. เครื่องรับจานแม่เหล็ก
5. อุปกรณ์สื่อสารข้อมูล

การติดต่อสื่อสารกับอุปกรณ์รอบนอกเหล่านี้เป็นไปได้ทั้งแบบอนุกรมและแบบขนาน แต่ ในโครงการนี้เป็น การโอนถ่ายข้อมูลแบบอนุกรมกับอุปกรณ์สื่อสารข้อมูล

2.3 การโอนถ่ายข้อมูล

2.3.1 การโอนถ่ายข้อมูลแบบขนาน

ลักษณะของการส่งข้อมูลแบบขนาน ทำให้โดยการส่งข้อมูลทีละ 1 ไบต์ คือ 8 บิต จากอุปกรณ์ส่งไปยังอุปกรณ์รับ ตัวกลางระหว่าง 2 เครื่องจะต้องมี ช่องทางให้ข้อมูลเดินทางอย่างน้อย 8 ช่องทาง โดยมากจะเป็นสายขนานให้กระแสไฟฟ้าวิ่งมากกว่าจะเป็นตัวกลางชนิดอื่น เนื่องจากมีสัญญาณสูญหายไปกับความต้านทานของสายระยะทางระหว่าง 2 เครื่องไม่ควรจะเกิน 100 ฟุต ปัญหาที่เกิดขึ้นหากระยะทางสายมากกว่านี้คือ ระดับของกราวด์ในทางไฟฟ้าที่จุดรับผิดไปจากจุดส่ง ทำให้เกิดผิดพลาดในการรับสัญญาณแอสจิกทางฝ่ายรับ

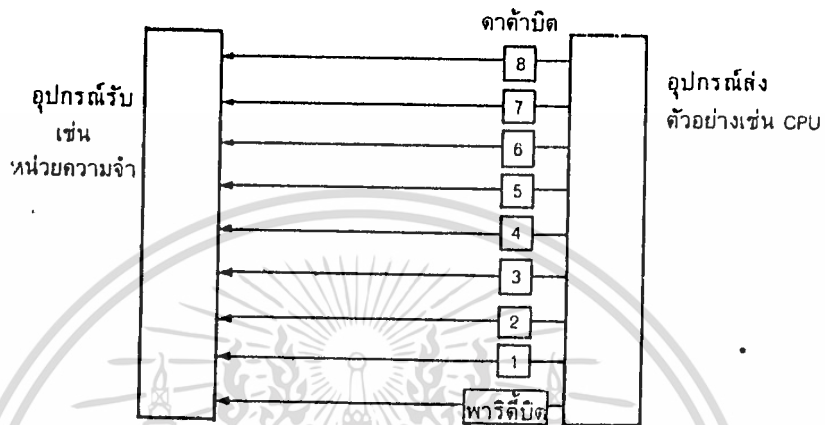
นอกจากสายที่เป็นทางเดินของข้อมูลแล้วอาจจะมีทางเดินของสัญญาณควบคุมอื่น ๆ อีก เป็นต้นว่า บิตที่บอกพาริตีของสัญญาณ เพื่อเป็นการตรวจสอบความผิดพลาดของการรับสัญญาณที่ปลายทาง หรือสายที่ควบคุมการโต้ตอบ (Hand-shake) ดังที่กล่าวมาแล้วจะเห็นว่าการส่งแบบขนานส่วนมากจะทำในระยะใกล้ ๆ เนื่องจากจะต้องมี ช่องทางเดินมากกว่า 8 สาย และอุปกรณ์ที่ติดต่อแบบขนานกับคอมพิวเตอร์ก็เห็นจะได้แก่ เครื่องพิมพ์ที่ดังกล่าวมาแล้ว

2.3.2. การถ่ายโอนข้อมูลแบบอนุกรม

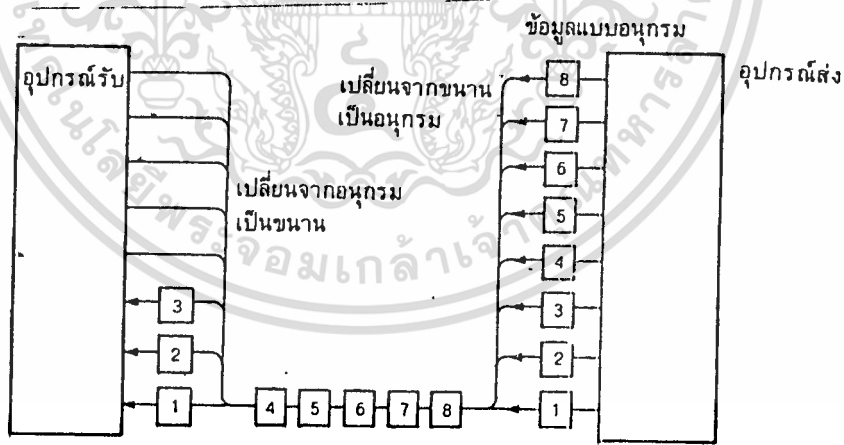
ในการถ่ายโอนข้อมูลแบบอนุกรม ข้อมูลถูกส่งออกมาทีละบิต ระหว่าง จุดส่งและจุดรับ จะเห็นว่าการส่งข้อมูลแบบนี้จะช้ากว่าแบบขนานที่กล่าวมาแล้วแน่นอน แล้วทำไมต้องส่งแบบนี้ คำตอบก็คือ ตัวกลางการสื่อสารต้องการเพียงช่องเดียวหรือ สายเพียงคู่เดียว ค่าใช้จ่ายในสื่อกลางจะต้องถูกกว่าแบบขนานอย่างแน่นอนสำหรับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การส่งระยะไกล ๆ โดยเฉพาะเมื่อเรามีระบบการสื่อสารทางโทรศัพที่ไว้ใช้งานอยู่แล้วย่อมจะเป็นการประหยัดกว่าที่จะทำการติดต่อสื่อสารทีละ 8 ช่อง เพื่อการถ่ายโอนข้อมูลแบบขนานอย่างแน่นอน



รูปที่ 2.1 แสดงการส่งข้อมูลแบบขนาน



รูปที่ 2.2 แสดงการส่งข้อมูลแบบอนุกรม

รูปที่ 2.2 แสดงให้เห็นการส่งข้อมูลแบบอนุกรม ข้อมูลจากจุดส่งจะถูกเปลี่ยนให้เป็นอนุกรมเสียก่อน แล้วค่อยทยอยส่งออกทีละบิตไปยังจุดรับ ณ ที่จุดรับจะต้องมีกลไกในการเปลี่ยนข้อมูลที่ส่งมาทีละบิต ให้เป็นสัญญาณแบบขนานซึ่งส่งตัวพอนี้แต่คือ บิต 1 ลงที่บัส เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ข้อมูลเส้นที่ 1 พอดี การที่จะทำให้การแปลงสัญญาณจากอนุกรมทีละบิต ให้ลงพอดีนั้นจำเป็นจะต้องมีกลไกที่เหมาะสม เพื่อป้องกันการผิดพลาดในการรับ กลไกที่ว่านี้แบ่งเป็น 2 แบบ คือ

1. การสื่อสารแบบซิงโครนัส
2. การสื่อสารแบบอะซิงโครนัส

2.4 รูปแบบของการติดต่อสื่อสารแบบอนุกรม

การติดต่อแบบอนุกรมอาจจะแบ่งตามรูปลักษณะได้ 3 แบบตามรูปที่ 2.3

1. แบบซิมเพล็กซ์ (Simplex) ข้อมูลส่งได้ในทางเดียวเท่านั้น บางครั้งก็เรียกว่าการส่งทิศทางเดียว (Unidirectional data bus)
2. แบบฮาล์ฟดูเพล็กซ์ (Half duplex) ข้อมูลสามารถส่งได้ทั้งสองสถานี แต่จะต้องผลัดกันส่งผลัดกันรับ จะส่งและรับพร้อมกันไม่ได้
3. แบบฟูลดูเพล็กซ์ (Full duplex) ทั้งสองสถานีสามารถรับและส่งได้ในเวลาเดียวกัน



รูปที่ 2.3 รูปแบบของการติดต่อสื่อสารข้อมูลแบบอนุกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การส่งแบบพЛУตูลูเพิ้ลัษัและแบบอาล์ฟูตูลูเพิ้ลัษั ไม่ขึ้นกับจำนวนของสายในการติดต่อ บางครั้งคำว่า ทุวัวร์ (two wire) หรือสองเส้น และไฟร์วัวร์ (four หรือสี่เส้น) ใช้ในการบรรยายถึงลักษณะการสื่อสารข้อมูลซึ่งอาจจะทำให้เข้าใจว่า สายโทรศัพท์ทั่วไปเป็นแบบ 2 เส้น ส่วนในสายที่เป็นแบบเช่า (lease line) นั้นส่วนมากจะเป็นสี่เส้น

2.5 ความเร็วในการถ่ายโอนข้อมูลแบบอนุกรม

ความเร็วของการถ่ายโอนข้อมูลแบบอนุกรม หน่วยวัดเป็นบิตต่อวินาที เรียกว่าบอดเรต (baud rate) หรืออัตราบอด

$$\text{อัตราบิต (bit rate)} = \text{อัตราบอด (baud rate)} \times \text{บิตใน 1 บอด}$$

2.6 การสื่อสารแบบอะซิงโครนัส

การส่งแบบอะซิงโครนัสนี้ พัฒนามาจากการส่งโทรพิมพ์ในสมัยก่อนลักษณะของสัญญาณแสดงไว้ในรูป 2.4 เพื่อเพิ่มกลไกในการรับส่งอย่างถูกต้อง สัญญาณอะซิงโครนัส จะประกอบด้วยบิตเริ่มต้น (start bit) และบิตสิ้นสุด (stop bit)



รูปที่ 2.4 รูปแบบการสื่อสารแบบอะซิงโครนัส

ขณะที่สถานะของการส่งเป็นแบบว่าง (idle) คือ ยังไม่มีสัญญาณส่งออกมา จะมีสัญญาณหรือแรงดัน (หรือกระแส) ตลอดเวลา เพื่อความแน่ใจว่าฝ่ายรับยังติดต่ออยู่กับฝ่ายส่ง เมื่อเริ่มส่งข้อมูล สัญญาณอะซิงโครนัสจะเป็น 0 ในช่วงสัญญาณนาฬิกา บิตนี้เรียกว่า บิตเริ่มต้น ตามหลังของบิตเริ่มต้นก็จะเป็นข้อมูลสำหรับ 1 ตัวอักษร ซึ่งอาจจะมีความยาวตั้งแต่ 5 บิต จนถึง 8 บิต โดยบิตที่มีค่าน้อยที่สุด (LSB) จะถูกส่งออกมาก่อนไล่ไปจนถึงบิตที่มีค่ามากที่สุด (MSB) การเข้ารหัสอักขระนี้ส่วนมากจะนิยมใช้รหัส แอสกี แรกเริ่มทีเดียวในงานของโทรพิมพ์เขาใช้รหัส โบตทอ (Baudot) ซึ่งใช้ 5 บิต ในการแทนอักขระ 1 ตัว ตามหลังข้อมูลก็จะเพิ่มพาริตีบิต ซึ่งอาจจะใช้หรือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ไม่ใช่ก็ได้ พาริตีบิตทำหน้าที่เป็นตัวตรวจสอบความถูกต้องของสัญญาณที่ได้รับ พาริตีบิต อาจจะเป็นแบบคู่ (Even) หรือแบบคี่ (Odd) หมายความว่าถ้าหากเป็นพาริตีคี่ จำนวนบิตที่ 1 ในช่วงบิตข้อมูลกับบิตพาริตีรวมแล้วจะต้องเป็นจำนวนคู่ ผู้ส่งจะต้องทำหน้าที่ ตรวจสอบข้อมูลแล้วใส่พาริตีบิตเอง ฝ่ายรับเมื่อรับแล้วจะต้องตรวจสอบว่าเป็นจริงดัง สถานการณ์ที่ตั้งเอาไว้หรือไม่ หากผิดพลาดก็หมายความว่าสัญญาณที่รับนั้นผิดพลาดไป จากสถานีส่งส่งออกมาทั้งนี้ทั้งนั้นจะต้องผิดเป็นจำนวนคี่เท่านั้น คือผิดไป 1 บิต 3 บิต หรือ 5 บิตพร้อมกัน จึงจะตรวจสอบได้ว่าผิด มองเห็นง่ายๆ ว่ามาผิดเป็นจำนวนคู่ ผลรวม ของจำนวนหนึ่งก็ยังคงเป็นคู่อยู่ดี ทั้งนี้ทั้งนั้นไม่ได้หมายความว่าพาริตีคี่ (Odd parity) จะ ตรวจสอบความผิดพลาดเป็นจำนวนคี่ ความจริงแล้วตรวจสอบความผิดพลาดได้เหมือนกับ พาริตีคู่ (Even parity) แต่แทนที่จะตรวจสอบว่าสัญญาณที่รับเข้ามามีจำนวนคู่ ก็ตรวจสอบ ว่ามีจำนวนคี่หรือเปล่า อย่างไรก็ตามโอกาสที่จะผิดพลาด 2 บิตพร้อมกันมีน้อยมาก

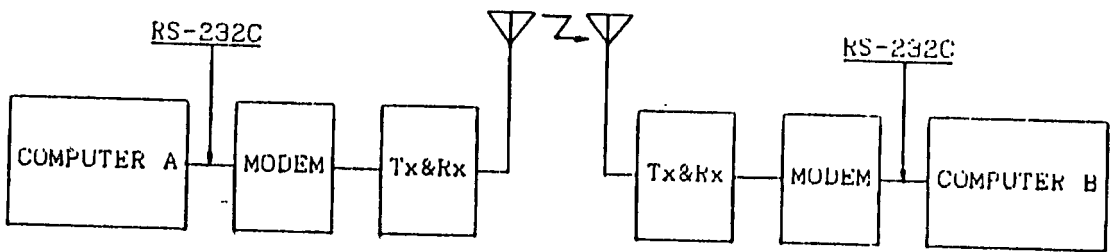
ย้อนกลับมาดูสัญญาณเอชซีโครนัลใหม่ หลังจากบิตพาริตีแล้วก็จะต้องมี สตอปบิตซึ่ง เป็น 1 ความกว้างของสตอปบิตอาจจะเป็น 1, 1.5 หรือ 2 พัลส์ของ สัญญาณนาฬิกา แล้วแต่ผู้รับและผู้ส่งจะตกลงใช้กันเอง การเริ่มใช้ฮาร์ดแวร์จนกระทั่ง จำเป็นต้องตั้งค่าต่างๆ สำหรับการส่งแบบอนุกรม อันได้แก่

1. ความเร็วในการส่ง
2. ความยาวรหัส 1 อักขระ
3. บิตตรวจสอบ
4. จำนวนสตอปบิต

2.7 ข่ายข้อมูลผ่านระบบวิทยุ

ข่ายข้อมูลระบบวิทยุคือ ข่ายการสื่อสารข้อมูลที่นำเอาระบบวิทยุมาใช้เป็นตัว พาสัญญาณข้อมูลให้ไปได้ไกลๆ แทนข่ายสายโทรศัพท์ ในกรณีที่มีบริเวณนั้นไม่มีสายโทรศัพท์ วางไปถึง

โครงสร้างหลักๆ ของระบบมีไมโครคอมพิวเตอร์ เป็น อุปกรณ์ปลายทาง โดยมี โมเด็ม 1200 บิตต่อวินาที เป็นตัวแปลงสัญญาณข้อมูลให้เป็นสัญญาณ อนุลอก โดย มอดดูเลข สัญญาณแบบ เข้ารหัสความถี่ (Frequency Shift Keying (FSK)) ใช้ มาตรฐานการอินเตอร์เฟส RS-232C ระหว่าง โมเด็มและอุปกรณ์ปลายทาง ส่งออกอา ภาคโดยระบบวิทยุ FM Half Duplex ตั้งแสดงในรูป 2.5



รูปที่ 2.5 แสดงบล็อกไดอะแกรมของระบบข่ายข้อมูลผ่านระบบวิทยุ

2.8 อาร์ตเวิร์กที่เกี่ยวข้องกับการสื่อสารข้อมูล

2.8.1 พอร์ต RS 232C

โดยปกติไมโครคอมพิวเตอร์จะมีพอร์ตที่เป็นแบบอนุกรม เรียกว่า RS 232C อยู่ในตัวเองอยู่แล้ว หลายเครื่องไม่มีมากับเครื่อง อย่างเช่น IBM PC จำเป็นจะต้องมีการดัดที่เรียกว่าอะซิงโครนัสอะแดปเตอร์มาเปลี่ยนใส่

พอร์ต RS 232C นี้ทำหน้าที่รับและส่งข้อมูลในแบบอนุกรม เรียกว่า อะซิงโครนัส อะแดปเตอร์ (Asynchronous Adapter) เทคโนโลยีที่ชื่อเรียกว่า RS 232C ก็เนื่องจาก สมาคมผู้ผลิตอุปกรณ์อิเล็กทรอนิกส์ของอเมริกาเหนือ หรือ EIA ได้กำหนดมาตรฐานของอุปกรณ์การสื่อสารแบบอนุกรมเอาไว้ภายใต้ชื่อว่า RS 232C ความจริงมาตรฐานของการส่งข้อมูลแบบอนุกรมมีหลายมาตรฐาน แต่ที่นิยมกันมากที่สุดสำหรับไมโครคอมพิวเตอร์ก็คือ RS 232C

หน้าที่สำคัญของการสื่อสารแบบอะซิงโครนัสก็คือ

รับสัญญาณ

1. เปลี่ยนสัญญาณเข้ามาแบบอนุกรมให้เป็นแบบขนาน
2. ตรวจสอบความผิดพลาดของสัญญาณที่รับ
3. ตัดสตอปบิตและพาริตีบิตออก
4. ส่งสัญญาณให้ทีนียูรู้ว่าได้รับสัญญาณไว้แล้ว



ส่งสัญญาณ

1. เปลี่ยนสัญญาณแบบขนานจากซีพียูค้อยทอยส่งออกเป็นแบบอนุกรม
2. เพิ่มสล็อตบิตและหาริตบิต
3. เพิ่มสัญญาณควบคุมโมเด็มที่เชื่อมต่อ (ถ้ามี)

2.8.2 ลักษณะสัญญาณ RS 232C

เพื่อเป็นหลักประกันว่าข้อมูลถูกส่งออกไปอย่างถูกต้อง และอุปกรณ์
 ถูกควบคุมอย่างถูกต้อง จำเป็นจะต้องมีข้อตกลงกันในเรื่องของสัญญาณที่ใช้ มาตรฐาน RS
 232C กำหนดค่าของแรงดันไฟฟ้าในสัญญาณเพื่อสนองจุดประสงค์ดังกล่าว ดังแสดงใน
 ตารางที่ 2.2 และรูปที่ 2.6

ตารางที่ 2.2 แสดงมาตรฐานการใช้แรงดันไฟฟ้าของ RS-232C

มาตรฐานการใช้แรงดันไฟฟ้า			
แรงดันไฟฟ้า	สถานภาพลอจิก	สถานภาพของสัญญาณ	ฟังก์ชันในการควบคุม
บวก	0	สเปซ	ออน
ลบ	1	มาร์ค	ออฟ

+15V -----

ย่านแรงดันไฟฟ้าบวก

+3V -----

ย่านเปลี่ยนแปลงระดับ

-3V -----

ย่านแรงดันไฟฟาลบ

-15V -----

รูปที่ 2.6 ย่านของแรงดันไฟฟ้าที่ใช้ในสัญญาณ RS 232C

สำหรับไมโครคอมพิวเตอร์บางเครื่องใช้แต่สัญญาณลอจิกออกมาเป็นสัญญาณของ RS 232C เลย อย่างเช่น ยะซิงโครนัสสายเคเบิลเตอร์ของ IBM PC ในกรณีเช่นนี้ ระยะทางของสายที่เชื่อมอาจจะไปได้สั้นกว่า 50 ฟุต ดังที่กล่าวเอาไว้เนื่องจากระดับของกราวด์เปลี่ยนแปลงไป อันเนื่องจากการสูญเสียไปในความต้านทานของสาย ผู้ที่เคยใช้ IBM PC อาจจะเคยประสบกับปัญหาว่า เวลาต่อสัญญาณ RS 232C เกินกว่า 10 ฟุตแล้วใช้งานไม่ได้ แต่อย่างไรก็ตาม RS 232C ของ IBM PC ยังมีโอกาสเลือกรับใช้ 20 มิลลิแอมแปร์ กระแสกลับแทนแรงดันไฟฟ้า

2.0.3 การกำหนดจุดต่อของ RS 232C

ในทางฟิสิกส์แล้ว มาตรฐานของ RS 232C กำหนดข้อต่อแบบ DB-25 แต่ละขาของข้อต่อกำหนดไว้ดังรูปที่ 2.7 สัญญาณต่าง ๆ ของขาในหัวต่อ RS 232C กำหนดหน้าที่ดังนี้

-Transmit Data (TD ขาที่ 2)

เป็นสัญญาณที่ส่งออกจาก DTE (ไมโครคอมพิวเตอร์) ไปยังโมเด็ม หรือต่อเข้าโดยตรงกับไมโครคอมพิวเตอร์ตัวอื่น หรือเครื่องพิมพ์ เมื่อไม่มีสัญญาณส่งออก สถานะของลอจิกที่ขาจะมีค่าเท่ากับ 1 หรือเทียบเท่ากับสตีอปีต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

-Receive Data (RD ขาที่ 3)

เป็นทางของสัญญาณเข้าไปยัง DTE หรือ ไมโครคอมพิวเตอร์ เมื่อไม่มีสัญญาณรับเข้ามา ขานี้จะมีสถานะทางลอจิก เป็น 1

-Request To Send (RTS ขาที่ 4)

ใช้สำหรับส่งสัญญาณไปยังโมเด็ม หรือเครื่องพิมพ์ เป็นการเรียกร้องที่จะส่งสัญญาณเข้ามาทางขาที่ 2 สัญญาณนี้ใช้คู่กับ CTS หรือ Clear To Send อุปกรณ์รับ หากได้รับสัญญาณ RTS จะตรวจสอบตัวเองว่าพร้อมจะรับสัญญาณได้หรือยัง หากพร้อมที่จะรับ ก็จะส่งสัญญาณออกไปที่สาย CTS

-Clear To Send (CTS ขาที่ 5)

ตั้งอธิบายไว้ใน RTS เมื่อสัญญาณอยู่ในสถานะออฟ (negative voltage หรือลอจิก 1) หมายความว่า อุปกรณ์รับกำลังบอกว่าการพร้อมที่จะรับข้อมูลแล้ว

-Data Set Ready (DSR ขาที่ 6)

เมื่อสัญญาณสายอยู่ในสถานะออน (ลอจิก 0) เป็นการบอกไมโครคอมพิวเตอร์ว่า โมเด็มต่อเข้ากับสายโทรศัพท์เรียบร้อยแล้ว พร้อมที่จะส่งได้แล้ว

-Signal Ground (SG ขาที่ 7)

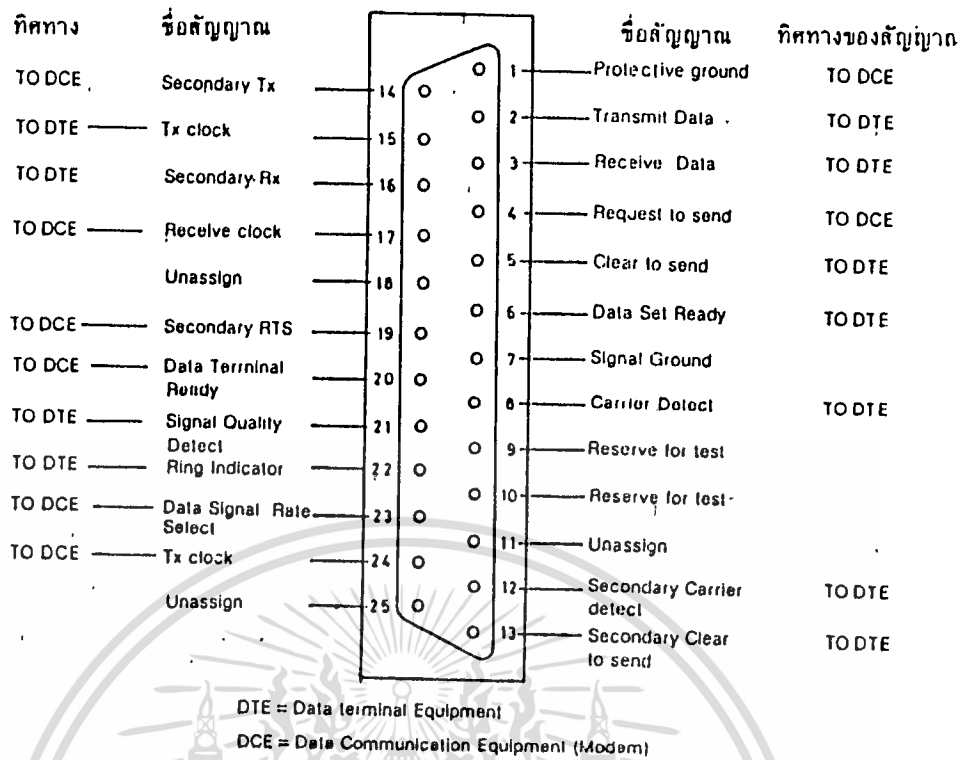
SG ทำหน้าที่เป็นระดับแรงดันอ้างอิงกับทุก ๆ สายของสัญญาณ จะมีแรงดันเป็น 0 เมื่อเทียบกับสัญญาณตัวอื่น

-Carrier Detect (CD ขาที่ 8)

โมเด็มจะส่งสัญญาณที่อยู่ในสถานะออนไปบอกไมโครคอมพิวเตอร์ เมื่อได้รับสัญญาณจากโมเด็มของอีกฝ่ายหนึ่ง

-Data Terminal Ready (DTR ขาที่ 20)

-คอมพิวเตอร์เปิดสัญญาณสายนี้ให้ออนเมื่อนพร้อมที่จะติดต่อกับโมเด็ม โมเด็มส่วนมากจะไม่รายงานสถานะภาพของตัวเอง (CD, DSR และ CTS) ให้คอมพิวเตอร์รู้ หากคอมพิวเตอร์ไม่เปิดสัญญาณ DTR



รูปที่ 2.7 แสดงการกำหนดขั้วต่อแบบ DB-25 ของ RS-232C

2.9 โมเด็ม

สัญญาณข้อมูลจะมีลักษณะเป็น เลขฐานสอง (Binary) การที่ส่งสัญญาณข้อมูลเช่นนี้ไปในสายส่งจะไม่สามารถทำได้ ดังนั้นจะต้องมีการ มอดูเลชันให้เป็นสัญญาณแอนะล็อก แทนค่าสัญญาณข้อมูลด้วยการเปลี่ยนแปลงความถี่หรือเปลี่ยนแปลงเฟสก่อน ทางฝ่ายรับก็จะต้องมีการเปลี่ยนแปลงของความถี่หรือเฟส ให้กลับเป็นสัญญาณอย่างเต็ม กระบวนการเปลี่ยนแปลงของความถี่หรือเฟสเรียกว่ามอดูเลชัน เพราะฉะนั้นอุปกรณ์ที่ใช้ทั้งมอดูเลชัน และ ดีมอดูเลชัน (Demodulation) เรียกว่า โมเด็ม (Modem)

2.9.1 การแบ่งชนิดของโมเด็ม ตามอัตราส่งข้อมูล

1. อัตราการส่งข้อมูลต่ำ (Low-speed)
มีอัตราการส่งข้อมูลไม่เกิน 600 bps
2. อัตราการส่งข้อมูลปานกลาง (Medium-speed)
มีอัตราการส่งข้อมูลระหว่าง 1200 ถึง 9600 bps
3. อัตราการส่งข้อมูลสูง (High-speed)
มีอัตราการส่งข้อมูลมากกว่า 9600 bps

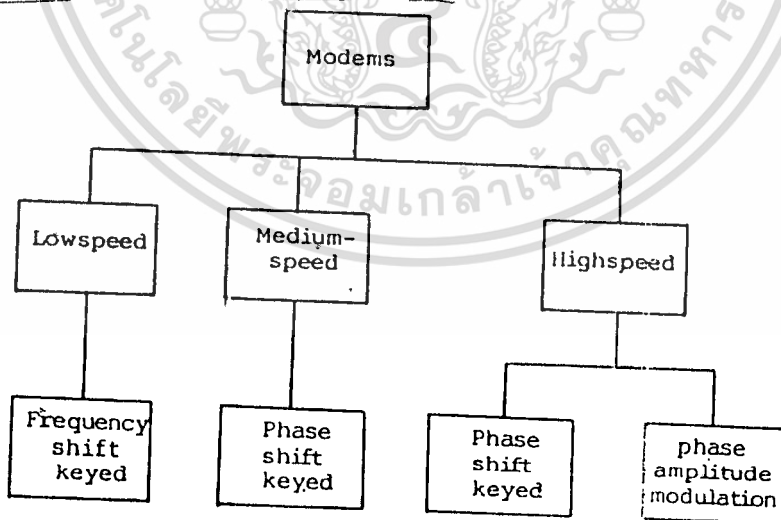
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ซึ่งในปัจจุบันโมเด็มที่มีจำหน่ายในตลาดอุปกรณ์อิเล็กทรอนิกส์ จะอยู่ใน 3 ชนิดนี้แต่ในอนาคตอันใกล้นี้ เครื่องคอมพิวเตอร์ทั่วไปมีแนวโน้มที่จะต้องมีโมเด็มประกอบ มาในเครื่อง (Built In Modem) และในการพัฒนาขั้นต่อไปก็คือโมเด็มที่ประกอบมา กับเครื่องคอมพิวเตอร์จะสามารถปรับอัตราเร็วการส่งข้อมูลได้ (Speed Select built in Modem) ในการพัฒนารูปแบบดังกล่าวนี้คงจะมีในอนาคตอันใกล้นี้

2.9.2 การแบ่งชนิดโมเด็มตามเทคนิคการโมเด็มเลข

1. เฟสชิฟคีย์อิง (Phase Shift Keying : PSK)
2. เฟรควเอนซีชิฟคีย์อิง (Frequency Shift Keying : FSK)
3. เฟสแอมพลิจูดโมดูเลชัน (Phase Amplitude Modulation : PAM)

สำหรับวิธีการมอดูเลตยังมีเทคนิคอื่นๆ อีกหลายแบบ แต่ก็ค่อนข้างจะกันน้อย ตามการแบ่งโดยลักษณะต่างๆ อาจสรุปเป็นไดอะแกรมได้ดังรูป



รูปที่ 2.8 การแบ่งชนิดของโมเด็ม

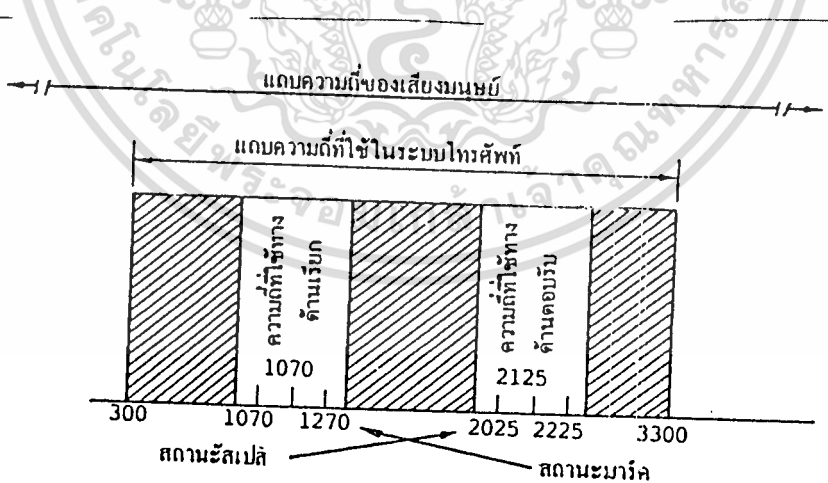
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.9.9 เทคนิคการโมดูเลชัน

การกำหนดความถี่ของด้านผู้เรียกกับผู้ตอบรับจากข้างต้นเราได้กล่าวถึงแล้วว่าสัญญาณดิจิทัลประกอบด้วยสัญญาณ 2 สถานะคือ 0 กับ 1 ดังนั้นระดับโวลเตจหนึ่งแทน 0 อีกระดับโวลเตจหนึ่งก็จะแทนด้วย 1 เช่นกัน จึงได้มีการกำหนดความถี่ของสัญญาณเพื่อใช้แทนระดับโวลเตจของทางด้านผู้เรียกกับผู้ตอบรับ ซึ่งต้องใช้ต่าง ๆ กัน 4 ความถี่ เพื่อไม่ให้เกิดการรบกวนของสัญญาณ (ในกรณีที่มีการสื่อสารแบบฟูลดูเพล็กซ์) โดยทางด้านส่งจะต้องใช้ความถี่ของสัญญาณ 2 ค่า เพื่อแทนโวลเตจระดับ 1 กับระดับ 0 และทางด้านรับก็เช่นเดียวกัน แต่ความถี่อีก 2 ค่าที่ใช้จะต่างกันในกรณีของ Bell-103 ได้มีการกำหนดความถี่ของสัญญาณที่ใช้ไว้ดังนี้คือ

- ความถี่ 1070 Hz สำหรับทางด้านส่ง แทนลอจิก 0
- ความถี่ 1270 Hz สำหรับทางด้านส่ง แทนลอจิก 1
- ความถี่ 2025 Hz สำหรับทางด้านรับ แทนลอจิก 0
- ความถี่ 2225 Hz สำหรับทางด้านรับ แทนลอจิก 1

โดยแสดงเป็นแถบความถี่ได้ดังรูปที่ 2.9



รูปที่ 2.9 แสดงแบนด์วิดท์ของความถี่ที่กำหนดให้ใช้ในโมเด็มชนิด Bell-103 โดยใช้งานร่วมกับ PSTN (PACKET SWITCH TELEPHONE NETWORK) ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปจะเห็นว่า ความถี่ที่ระบุมานี้จะต้องมีความสัมพันธ์กันระหว่างโมเด็มทางต้นส่ง (Originate modem) ที่ทำการส่งสัญญาณด้วยความถี่หนึ่ง ส่วนโมเด็มทางต้นรับ (Answer Modem) ก็จะได้รับสัญญาณความถี่ค่านั้นเข้ามาแล้วตอบกลับไปด้วยสัญญาณที่ความถี่อีกค่าหนึ่ง ซึ่งเทคนิคในการนำสัญญาณแจ้งต่อลมาโมเด็มเลขเข้ากับสัญญาณเวลาออกแล้วส่งผ่านตัวนำออกไป เราเรียกว่าเทคนิคของ FSK หรือ Frequency Shift Keying

2.10 ความผิดพลาดและการแก้ไขความผิดพลาด

2.10.1 จุดเริ่มต้นของความผิดพลาด

ความผิดพลาดเกิดขึ้นเป็นสองแบบใหญ่ๆคือ ความผิดพลาดที่เกิดขึ้นอย่างทราบได้และความผิดพลาดที่เกิดขึ้นแบบลุ่ม การบิดเบือนของสัญญาณและการสูญเสียที่เกิดจากลดทอน เป็นตัวอย่างของความผิดพลาดอย่างทราบได้ ส่วนสนามแม่เหล็กไฟฟ้าในบรรยากาศ เช่น การเกิดฟ้าผ่า หรือสนามหลังจากจุดยอดของตงอาทิตยส์ เหล่านี้เป็นความผิดพลาดแบบลุ่ม

ความผิดพลาดที่เกิดขึ้นแบบทราบได้นี้ง่ายต่อการแก้ไข เพราะเราสามารถพยากรณ์เหตุการณ์ล่วงหน้าได้ เราสามารถชดเชยสัญญาณที่ขาดหายไปได้โดยการขยายสัญญาณ ใช้สายสัญญาณที่คุณภาพดีขึ้น แต่ความผิดพลาดแบบลุ่ม เราไม่สามารถที่จะพยากรณ์ได้เพราะเกิดเป็นคาบเวลาที่ไม่แน่นอน ความผิดพลาดแบบลุ่มนั้นส่วนใหญ่เกิดจากสัญญาณรบกวนทางไฟฟ้า (noise) ในสายการสื่อสาร เช่น dropouts, crosstalk, echoes forth, impulse noise

สัญญาณรบกวนที่พบเห็นกันมากที่สุดคืออิมพัลส์ นอยส์ (impulse noise) หรือ เบอร์ส นอยส์ (burst noise) แม้ว่าในการสื่อสารข้อมูลทุกรูปแบบจะไม่สามารถหลีกเลี่ยงนอยส์ได้แต่เมื่อใช้การสื่อสารต่างรูปแบบกัน ความผิดพลาดก็จะเกิดขึ้นไม่เท่ากัน การสื่อสารแบบอนุกรมจะมีโอกาสเกิดความผิดพลาดมากกว่าแบบขนาน เพราะ สายส่งมีเพียงเส้นเดียว โอกาสที่ข้อมูลจะอยู่ในสายส่งก็มากขึ้นเมื่อมี burst noise เกิดขึ้นก็จะทำให้เกิดความผิดพลาดมากขึ้นเช่น เบอร์ส นอยส์ เกิดขึ้น 0.01 วินาที ไประหว่างการส่งข้อมูลความเร็ว 1,200 บอด จะทำให้ นอยส์ ทำลายข้อมูลได้ 12 บิต และถ้าเพิ่มความเร็วเป็น 9,600 บอด ความผิดพลาดก็จะมิถึง 96 บิต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.10.2 การตรวจสอบความผิดพลาด

การตรวจสอบความผิดพลาดมีหลายวิธีด้วยกัน แต่แบ่งเป็นประเภทได้ 2 ประเภทคือ

1. การตรวจสอบเป็นตัวอักษร (Character redundancy)
2. การตรวจสอบเป็นบล็อก (Block redundancy)

ในโครงการนี้ใช้การตรวจสอบความผิดพลาดแบบเป็นบล็อกโดยใช้ค่าผลบวกของรหัสแอสกีในบล็อกเป็นค่าตรวจสอบ

2.10.2 ค่าผลบวกตรวจสอบ (Checksum)

เป็นการตรวจสอบในแนวตั้งของบล็อกตัวอักษรแทนที่จะใช้ตัวอักษรตัวเดียว เป็นการบวกค่าแอสกีในบล็อกเข้าด้วยกัน ค่าที่ได้จะมากกว่า 0 ทิต เพราะฉะนั้นต้องหารด้วย 256 (modulo - 256) แล้วเศษที่ได้คือค่า ผลบวกตรวจสอบ เมื่อฝ่ายส่งหาค่าผลบวกตรวจสอบในหนึ่งบล็อก ได้แล้วก็จะส่งข้อมูลหนึ่งบล็อก นั้น พร้อมค่าผลบวกตรวจสอบ ไปด้วย ทางด้านรับก็จะรับเอาข้อมูลในบล็อกนั้น มาหาค่า ผลบวกตรวจสอบ และนำมาเปรียบเทียบกับค่าผลบวกตรวจสอบของต้นส่ง ถ้าเท่ากันแสดงว่าไม่มี ความผิดพลาด ถ้าไม่เท่ากันแสดงว่ามีความผิดพลาดเกิดขึ้น

2.10.3 การควบคุมการถ่ายทอดข่าวสาร

ในการส่งผ่านข้อมูลจากคอมพิวเตอร์เครื่องส่งไปยังคอมพิวเตอร์เครื่องรับและผ่านระบบวิทยุนั้น ในแต่ละระบบจะมีการประมวลผลด้วยความเร็วที่ไม่เท่ากัน ถ้าคอมพิวเตอร์มีความเร็วไม่เท่ากันการประมวลผลก็จะไม่เท่ากันด้วย อาจเป็นสาเหตุให้ข้อมูลสูญหายไป ส่วนในระบบวิทยุก็จะมี ความล่าช้า (delay) ต่างๆ เกิดขึ้น เพราะฉะนั้นจะมีโปรแกรมควบคุมการถ่ายทอดข่าวสารหรือ โปรโตคอล

2.10.4 โปรโตคอล ควบคุมการส่งข่าวสาร

ในที่นี้จะกำหนดการส่งเป็นแพ็คเกจ (packet) ในหนึ่งแพ็คเกจจะมี ส่วนของข้อมูล (data field) และส่วนเรื่องสัญลักษณ์ (signature field) ต่างๆ คือ ส่วนหัวจะเป็น หมายเลขแพ็คเกจ (packet-sequence

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นิยมนำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

SOH	PACKET SEQUENCE NO.	1's complement	DATA 128 BYTE	CHECKSUM
-----	------------------------	-------------------	------------------	----------

รูปที่ 2.11 แสดงส่วนประกอบของเอ็กซ์โมเด็มแพคเก็ต

SOH (start of header)	-- จุดเริ่มต้นของ แพคเก็ต
packet sequence No.	-- หมายเลขลำดับที่ของ แพคเก็ต หมายเลขของ แพคเก็ตแรกคือ 1
1's complement of packet sequence	-- หมายเลขลำดับที่ของ แพคเก็ต ก่อนหน้านี้ หนึ่งแพคเก็ต ที่ทำการ คอมพลิเมนต์
DATA	-- ข้อมูล 128 ไบต์
Arithmetic checksum	-- ข้อมูล 1 ไบต์ ที่เกิดจากการบวกค่า แอลเกอริทึมของข้อมูลแล้ว หายเอาเศษด้วย 256

2.11.1 เอ็กซ์โมเด็ม ด้านส่ง (XMODEM SEND)

จาก รูปที่ 2.12 อธิบายได้เป็นขั้นตอนดังนี้

ขั้นเริ่มต้น จะมีการเริ่มต้นติดต่อระหว่างด้านส่งและด้านรับ โดยด้านรับจะส่ง NAK เพื่อเป็นการบอกด้านส่งให้รู้ว่าเริ่มต้นข้อมูล แพคเก็ต แรกได้

ขั้นกลาง เมื่อด้านส่งรับ NAK จากด้านรับแล้วก็จะ จัดเรียง ข้อมูล 128 ไบต์จาก buffer และเข้ารูปแบบของ แพคเก็ต ส่งไปยังด้านรับและรอรับรหัส ACK หรือ NAK ถ้า ด้านส่งได้รับรหัส ACK ก็จะส่งข้อมูล แพคเก็ต ต่อไป แต่ถ้าเป็น NAK ก็จะส่ง แพคเก็ต เดิมไปแทน

ขั้นจบ เมื่อด้านส่งส่งข้อมูลไปหมดแล้ว ก็จะต้องส่งรหัส EOT (End-Of-Transmission) ไปด้านรับ เพื่อบอกว่าการส่งข้อมูลได้ยุติลงแล้ว และเมื่อด้านรับได้รับ EOT ก็จะตอบรับด้านส่งด้วยการส่ง ACK

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.11.2 เอกซ์โมเดม ต้ารับ (XMODEM RECEIVE)

เป็นขั้นตอนที่ยิ่งยากกว่าด้านส่งเพราะจะต้องคอยตรวจสอบว่าข้อมูลที่ส่งมานั้น

ผิดพลาด มีขั้นตอนดังนี้

ขั้นแรก ส่งรหัส NAK ให้ด้านส่งเพื่อขอร้องการส่งข้อมูล แพคเก็ต แรก

ขั้นที่สอง รอข้อมูลแต่ละ แพคเก็ต และหาจุดเริ่มต้นของ แพคเก็ต ได้โดยตรวจหารหัส SOH ในส่วนหัวของ แพคเก็ต และแยกเอา SOH. หมายเลขแพคเก็ต.ค่า 1's คอมพลิเมนท์ ของ แพคเก็ตที่แล้ว เพื่อเปรียบเทียบว่า หมายเลขที่ของ แพคเก็ต ที่แล้วตรงกับค่า 1's คอมพลิเมนท์ ของ แพคเก็ต ปัจจุบันที่ทำการเปลี่ยนกลับเป็นค่าธรรมดาหรือไม่ ถ้าไม่ตรงกันก็ให้ส่งรหัส NAK เพื่อขอร้องการส่งกลับใหม่ (retransmission) ถ้าหมายเลข แพคเก็ต ทุกตัวจะมีการตรวจสอบอีกทีโดย การสอบค่า arithmetic sum ถ้าตรงกันกับค่าที่ส่งมากับ แพคเก็ต ต้ารับจะส่ง NAK ถ้าไม่ตรงกันจะส่ง ACK

ขั้นสุดท้าย เมื่อได้รับ EOT จากด้านส่ง ก็ตอบรับด้วย ACK และทำการปิดข้อมูลและจัดเก็บข้อมูล



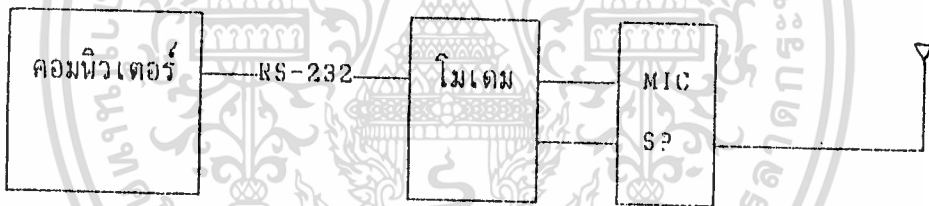
บทที่ 8

การออกแบบและการสร้าง

3.1 การออกแบบโมเด็มที่ใช้กับระบบวิทยุ

โมเด็มที่ใช้กับระบบวิทยุต้องมีคุณสมบัติดังต่อไปนี้

1. สามารถอินเทอร์เฟสกับคอมพิวเตอร์ ด้วยมาตรฐานอินเทอร์เฟสของ RS-232C ได้
2. มอดดูเลขชั้นสัญญาณข้อมูลให้เป็นสัญญาณในย่านความถี่เสียงพูด และตีมอดดูเลขชั้นในทางกลับกัน
3. สามารถตรวจสอบ และควบคุมการออกอากาศของวิทยุรับส่งได้



รูป 3.1 แสดงการต่อโมเด็มกับอุปกรณ์ต่างๆ

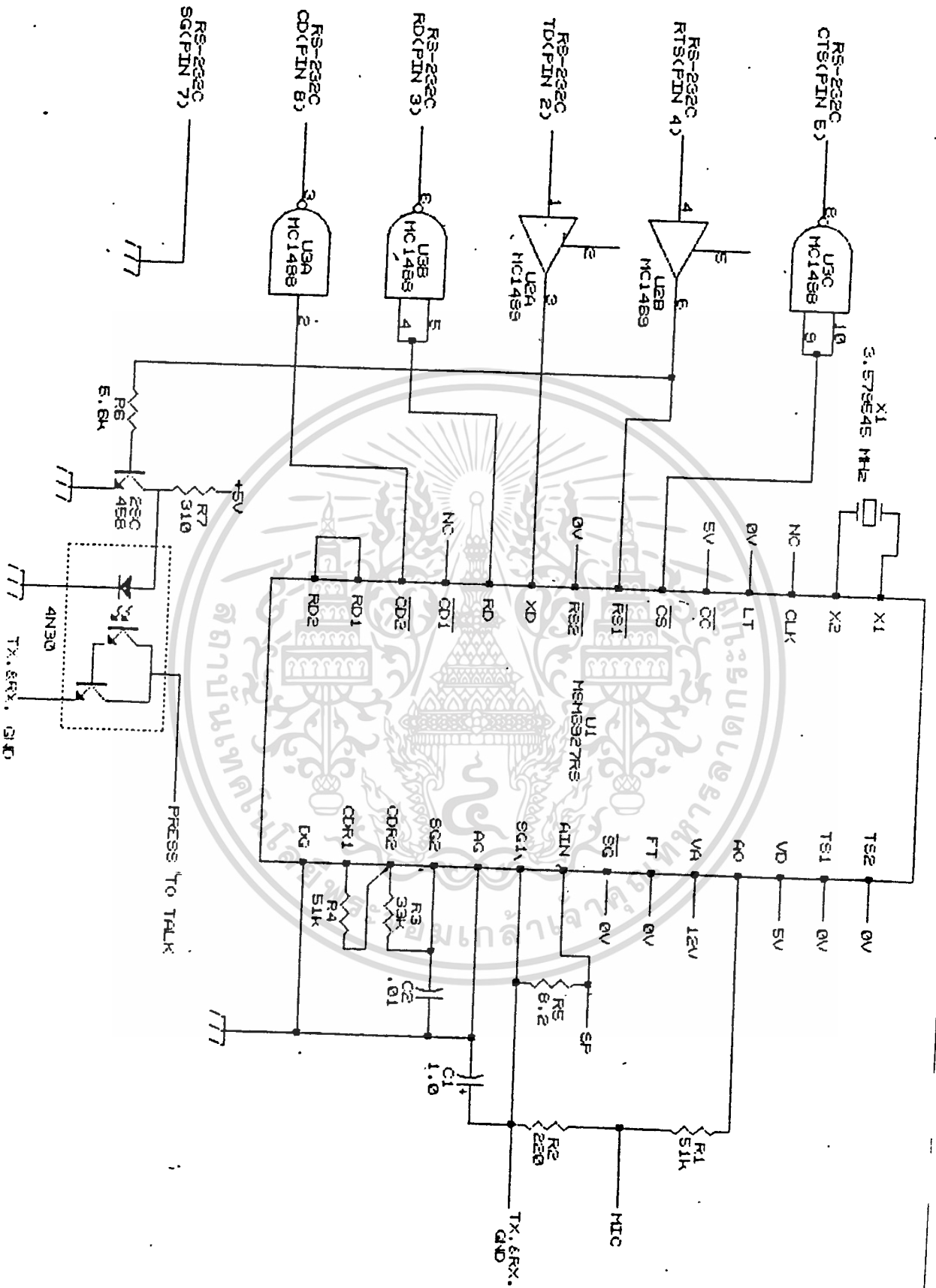
โมเด็มที่ใช้ในโครงการนี้ ใช้ไอซีโมเด็มเบอร์ M 5927 ซึ่งมีสัญญาณแอนต์เซคกับ RS-232 เกือบครบถ้วน และมีวงจรฟิลเตอร์ในตัวเอง เมื่อนำมาออกแบบจะได้วงจรดังรูป 3.2

3.2 การต่อโมเด็มเข้ากับเครื่องรับ-ส่งวิทยุ

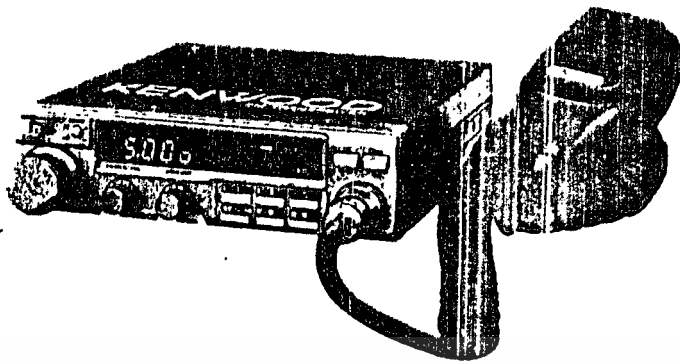
3.2.1 เครื่องรับส่งวิทยุ เอนเอ็ม

เป็นเครื่องยี่ห้อ KENWOOD รุ่น TM-201/D ใช้ในย่านความถี่ 145 เมกกะเฮิรตซ์ ดังรูปที่ 3.3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



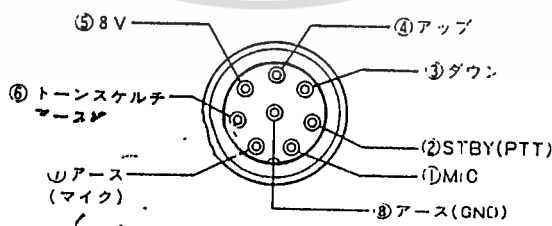
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับรูปที่ 3.2 แสดงวงจรโมเด็ม ไม่นอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.3 แสดงรูปเครื่องรับ-ส่งวิทยุ

การประยุกต์เพื่อนำโมเด็มมาต่อกับวิทยุรับ-ส่ง จะใช้สัญญาณขาออกจากโมเด็มที่มีความถี่ในช่วง 1800-2200 เฮิรตซ์ (มีค่าความถี่ไม่เกินช่วงความถี่เสียงพูด) ต่อเข้ากับขา MIC ของเครื่องรับส่งวิทยุ และใช้สัญญาณจาก SP มาเป็นสัญญาณขาเข้าของโมเด็ม โดยมีการตรวจสอบการออกอากาศโดยใช้ สัญญาณ RTS ความคุมการกดปุ่มพูด (press to talk) โดยเมื่อมีสัญญาณ RTS หมายถึงสถานะแรงดันไฟฟ้ามีค่าบวก (ลอจิก 0) จะทำให้ ทราานซิสเตอร์ 2SC45B ทำงานเพื่อไปขับ ไฟโตไดโอด 4N30 ให้ต่อวงจร กดปุ่มพูดเมื่อออฟสถานะของ RTS ทราานซิสเตอร์ 2SC450 ก็จะมีชุดทำงาน ทำให้วงจรของ 4N30 และ วงจรกดปุ่มพูดถูกตัดออก

เพื่อป้องกันการรบกวนของสัญญาณอื่น เมื่อโมเด็มไม่ได้รับสัญญาณคลื่นพาหะ ขา RD จะมีสัญญาณออกเป็นลอจิก 1 ตลอด จะทำให้ไมมีสัญญาณอื่นมารบกวนได้



รูปที่ 3.4 แสดงสัญญาณที่ขาต่าง ๆ ของหัวต่อเครื่องรับ-ส่งวิทยุ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3 การเขียนโปรแกรมโดยใช้ภาษาแอสเซมบลี 8086

ภาษาแอสเซมบลีสามารถเขียนโดยใช้การเรียกบริการจากระบบไบออส และการควบคุม 8250 โดยตรง

3.3.1 การเรียกบริการจากระบบไบออส

ฟังก์ชันที่ BIOS มีให้เลือกใช้ คือ Interrupt 14 โดยการให้คำสั่ง INT 14H โดยมีความหมายในการทำงานที่แตกต่างกันตามค่าที่อยู่ในรีจิสเตอร์ AH ถ้า AH = 0 จะเป็นการตั้งค่าพารามิเตอร์สำหรับการสื่อสาร โดยที่รีจิสเตอร์ AL มีค่าพารามิเตอร์ที่จะตั้งค่าดังนี้

บิตที่ 1 และบิตที่ 0 เป็นการบอกความยาวของข้อมูล

10 หมายถึง 7 บิต

11 หมายถึง 8 บิต

บิตที่ 2 เป็นการบอกความยาว Stop Bit

" 0 หมายถึง 1 Stop Bit

" 1 หมายถึง 2 Stop Bit

บิตที่ 4 และบิตที่ 5 บอกชนิดของการตรวจสอบ

X 0 หมายถึงไม่มีการตรวจสอบ

0 1 หมายถึงการตรวจสอบแบบคู่

1 1 หมายถึงการตรวจสอบแบบคู่

บิตที่ 7, บิตที่ 6 และบิตที่ 5 ใช้บอกความเร็วของการรับส่ง

000 หมายถึง 110 บอด

001 หมายถึง 150 บอด

010 หมายถึง 300 บอด

011 หมายถึง 600 บอด

100 หมายถึง 1200 บอด

101 หมายถึง 2400 บอด

110 หมายถึง 4800 บอด

111 หมายถึง 9600 บอด

AH = 1 เป็นการส่งอักขระในรีจิสเตอร์ AL ออกไปในสายสื่อสารหลัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากออกมาจากการขัดจังหวะ บิตที่ 7 ของรีจิสเตอร์ AH
จะเป็น 1 ถ้าหากการส่งทำได้สมบูรณ์ บิตที่เหลือของ AH บอกถึง
สถานะของสายสื่อสาร

AH = 2 เป็นการรับอักษระจากสายสื่อสารเข้ามาไว้ใน AL หลังจากที
ออกจากการขัดจังหวะ AH จะบอกสถานะของข้อผิดพลาดที่เกิดขึ้น

AH = 8 เป็นการตรวจสอบสถานะของการสื่อสารแบบอนุกรม เมื่อออกจาก
การขัดจังหวะค่าในรีจิสเตอร์ AH จะเป็นสถานะของสายสื่อสารโดยที่
บิต 7 = timeout (รอจนหมดเวลา)

บิต 6 = Transfer Shift Register Empty

บิต 5 = Transfer Register Empty

บิต 4 = Break Detect

บิต 3 = Framing Error

บิต 2 = Parity Error

บิต 1 = Overrun Error

บิต 0 = Data Ready

ค่าในรีจิสเตอร์ AL จะมีสถานะของโมเด็มดังนี้

บิต 7 = Receive Line Signal Detect

บิต 6 = Ring Indicate

บิต 5 = Data Set Ready

บิต 4 = Clear to Send

บิต 3 = Delta Receive Line Signal Detect

บิต 2 = Tailing Edge Ring Indicator

บิต 1 = Delta Data Set Ready

บิต 0 = Delta Clear to Send

รีจิสเตอร์ DX จะเป็นค่าบอกว่าจะใช้อะซิงครันสออะเตปเตอร์ตัวไหน (1 หรือ 2)

3.3.2 การควบคุม 8250 โดยตรง

-รีจิสเตอร์ควบคุมสายสื่อสาร (SCB)

ในการควบคุมรูปแบบของข้อมูลแบบอะซิงโครนัสนั้น ผู้โปรแกรมจะต้องกำหนดค่าลงในรีจิสเตอร์ควบคุมสายสื่อสาร รีจิสเตอร์ตัวนี้มี 8 บิต โดยแต่ละบิตมีความหมายดังนี้

บิต 0 และ 1 เป็นตัวกำหนดความยาวของข้อมูลในการรับส่ง โดยที่

บิต 0	บิต 1	
0	0	หมายถึงข้อมูลขนาด 5 บิต
0	1	หมายถึงข้อมูลขนาด 6 บิต
1	0	หมายถึงข้อมูลขนาด 7 บิต
1	0	หมายถึงข้อมูลขนาด 8 บิต

บิต 2 เป็นบิตที่ใช้ในการกำหนดจำนวนบิตต่อบิต ถ้าเป็น "0" หมายถึงใช้ต่อบิต บิต ถ้าเป็น "1" หมายถึงต่อบิตมี 2 บิต

บิต 3 เป็นบิตแสดงการอานาเบลให้มีการตรวจสอบพาริตี โดยถ้าบิตนี้มีค่าเป็น 1 จะมีการเพิ่มพาริตี

บิต 4 มีค่าเป็น "0" และบิต 3 มีค่าเป็น "1" จะมีการกำหนดเป็นพาริตีคู่ ถ้าบิตนี้มีค่าเป็น "1" จะเป็นพาริตีคู่

บิต 5 เมื่อบิต 3, 4 และ 5 มีค่าเป็น "1" จะมีการแทรกหรือตรวจสอบพาริตีด้วยเงื่อนไขกำหนดให้เป็น "0" ถ้าบิต 4 มีค่าเป็น "0" จะมีการกำหนดพาริตีเป็น "1"

บิต 6 ควบคุมการเบรค เมื่อบิต 6 มีค่าเป็น "1" ส่วนของ SOUF จะได้รับการกำหนดให้เป็น "0" ตลอด

บิต 7 ทำหน้าที่เป็น DLAB

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

-รีจิสเตอร์แสดงสถานะสายสื่อสาร (SRD)

บิต 0 เป็นบิตที่บอกสถานะการรับข้อมูล ถ้าเป็น 1 แสดงว่ามีข้อมูลเข้ามาในบัฟเฟอร์ครบทุกบิต เมื่อซีพียูอ่านค่าไปแล้ว จะถูกรีเซ็ตให้เป็น 0

บิต 1 ถ้ามีค่าเป็น 1 แสดงว่าเกิด โอเวอร์รัน เออเรอร์ (overflow error)

บิต 2 มีค่าเป็น 1 แสดงว่าเกิด พาริตี เออเรอร์ (parity error)

บิต 3 มีค่าเป็น 1 แสดงว่าเกิดการผิดพลาดของเฟรมข้อมูล ไม่เป็นไปตามที่กำหนด

บิต 4 บิตนี้จะได้รับการเซ็ตให้มีค่าเป็น 1 ถ้าหากว่ารับข้อมูลเป็น 0 เป็นเวลายาวนานกว่าเวร็ดของการสื่อสาร

บิต 5 เป็นบิตที่บอกค่า 8250 พร้อมทั้งจะรับข้อมูลจากสายสื่อสาร บิตนี้จะได้รับการเซ็ตให้มีค่าเป็น 1

บิต 6 เป็นบิตที่จะบอกค่า ซีพียูรีจิสเตอร์ว่างเปล่า บิตนี้จะได้รับการเซ็ตให้มีค่าเป็น 1 เพื่อบอกว่าพร้อมส่ง

บิต 7 จะเป็น 0 ตลอด

-รีจิสเตอร์กำหนดอินเตอร์รัฟ (SFR)

ไอซี 8250 กำหนดความสำคัญของอินเตอร์รัฟไว้ 4 ระดับ คือ

ระดับแรก สถานะการรับข้อมูลจากสายสื่อสาร

ระดับสอง การพร้อมรับข้อมูล

ระดับสาม หมดเรจิสเตอร์ไอเอสตังสำหรับส่งข่าว

ระดับสี่ สัญญาณสถานะโมเด็ม

ในขณะที่มีความต้องการอินเตอร์รัฟหลายระดับพร้อมกัน 8250 จะให้ความสำคัญแก่ระดับแรกมากกว่า

บิต 0 ถ้าบิตนี้เป็น 1 แสดงว่า ไม่มีการอินเตอร์รัฟเกิดขึ้น

บิต 1-2 เป็นบิตที่แสดงว่าการอินเตอร์รัฟเกิดขึ้นจากอินเตอร์รัฟระดับไหน

บิต 3-7 มีค่าเป็น 0

-รหัสเตอร์อิเนาเบิลอินเตอร์รันท (3F9)

บิท 0 เป็น 1 เมื่อต้องการอิเนาเบิลอินเตอร์รันทการพร้อมรับข้อมูล

บิท 1 เป็น 1 เมื่อต้องการอิเนาเบิลอินเตอร์รันทโอสตั้งรหัสเตอร์ว่าง

บิท 2 เป็น 1 เมื่อต้องการอิเนาเบิลอินเตอร์รันทจากสถานะการรับข้อ

มูลจากสายสื่อสาร

บิท 3 เป็น 1 เมื่อต้องการอิเนาเบิลอินเตอร์รันทจากสถานะไมเดม

บิท 4-7 กำหนดให้เป็น 0 เสมอ

-รหัสเตอร์ควบคุมไมเดม (3FC)

บิท 0 เมื่อบิทนี้มีค่าเป็น 1 เอาท์พุทที่ DTR จะได้รับการกำหนดให้เป็น 0 และถ้าบิทนี้มีค่าเป็น 0 เอาท์พุทที่ DTR จะเป็น 1

บิท 1 มีความหมายถึง RTS จะมีผลเหมือนกับบิท 0

บิท 2-3 ใช้ควบคุมเอาท์พุท

บิท 4-7 เป็นบิทสำหรับการตรวจสอบระบบการทำงานของ 0250

-รหัสเตอร์แสดงสถานะไมเดม (3FR)

บิท 0 ใช้สำหรับแสดงการเปลี่ยนแปลงของสัญญาณ CTS กล่าวคือ เมื่อขา CTS ของ 8250 ได้เปลี่ยนสถานะหลังจากที่ขั้วพินได้อ่านสถานะนี้ไปแล้ว บิทนี้จะได้รับการรีเซ็ต และจะได้รับการเซ็ต เมื่อมีการเปลี่ยนสถานะที่ขา CTS

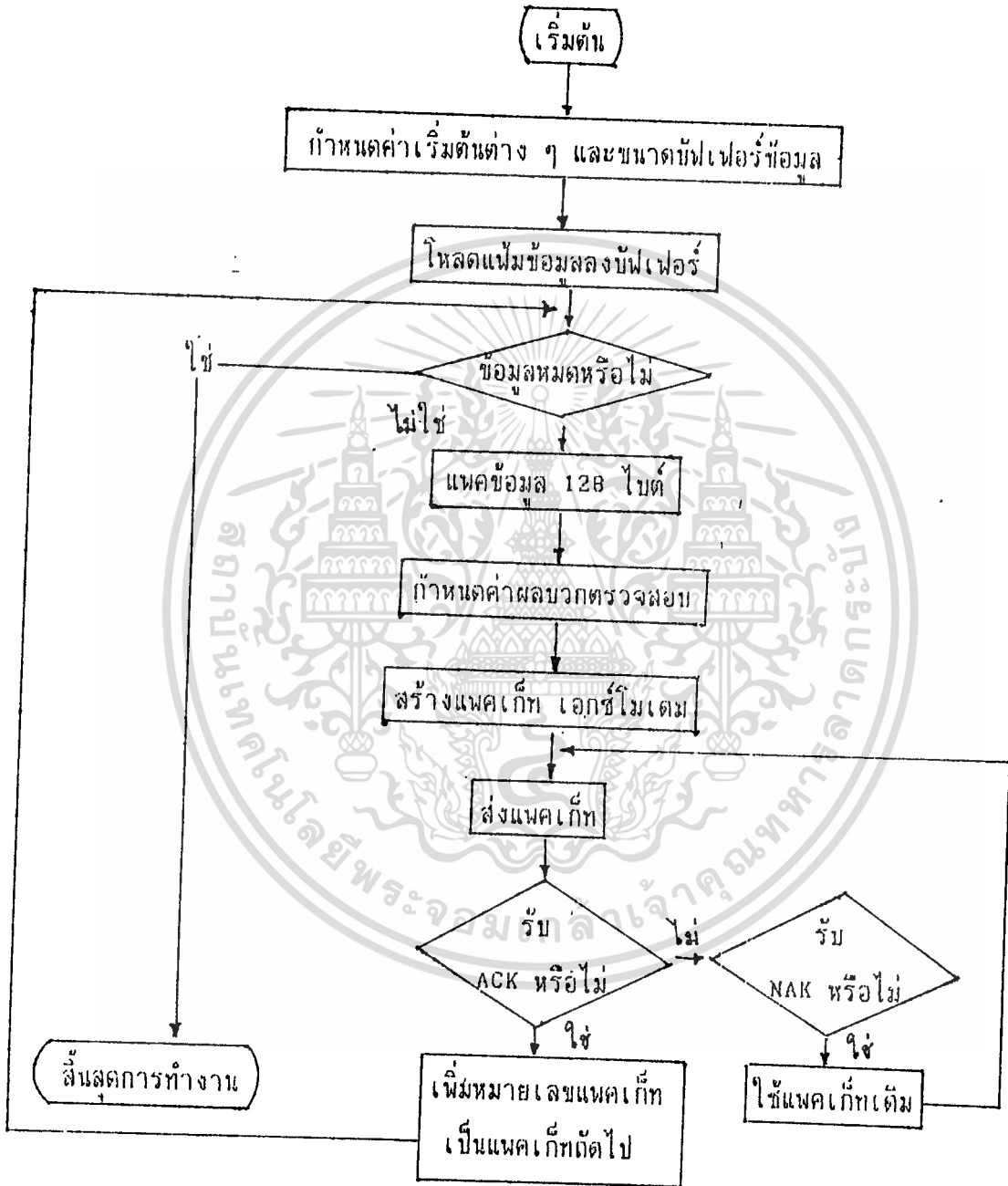
บิท 1 เหมือนบิท 0 แต่แสดงการเปลี่ยนแปลงของ DSR

บิท 2 เป็นบิทที่แสดงว่าสัญญาณ RI ได้รับการรีเซ็ต

บิท 3 เหมือนบิท 0 แต่เป็นการแสดงการเปลี่ยนแปลงของ RLSN

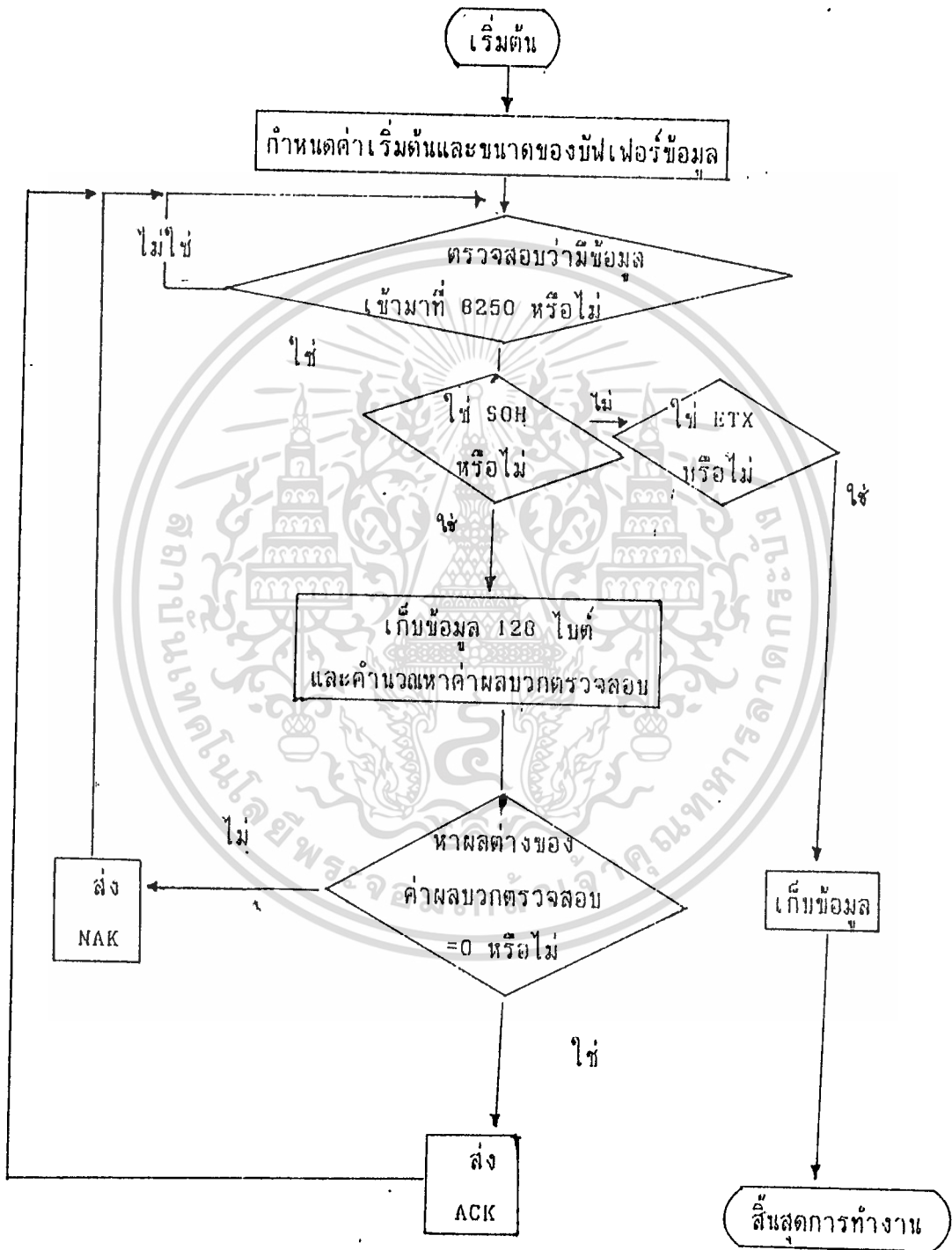
บิท 4-7 เป็นบิทเก็บสัญญาณคอมพลีเมนท์ของ CTS, DSR, RI, RLSN

3.4 การเขียนโปรแกรมส่งข้อมูล



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.5 การเขียนโปรแกรมรับข้อมูล



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

การทดลองและผลการทดลอง

โครงการนี้ได้แบ่งการทำการทดลองออกเป็น 2 ส่วน คือ

1. การทดลองวัดค่าระดับสัญญาณต่าง ๆ ของโมเด็ม
2. การทดลองโปรแกรมรับ-ส่งข้อมูล

4.1 การทดลองวัดค่าระดับสัญญาณต่าง ๆ ของโมเด็ม

ตำแหน่ง	ประเภทสัญญาณ	ค่าที่อ่านได้	หน่วย
RS-232	ไลน์ไดรเวอร์	10-12	โวลต์
โมเด็ม	ไลน์รีซีฟเวอร์	5	โวลต์
โมเด็ม	มอดดูเลขค่า 0	1300	เฮิรต์
โมเด็ม	มอดดูเลขค่า 1	2097	เฮิรต์

4.2 การทดลองโปรแกรมรับ-ส่งข้อมูล

4.2.1 การทดลองใช้โปรแกรม PAC_SND ส่งข้อมูล ขนาด 14981 ไบต์ โดยใช้โมเด็ม และใช้โปรแกรม RECV เป็นโปรแกรมรับ

-ผลการส่งแฟ้มข้อมูลไม่มีข้อผิดพลาดใด ๆ ตัวอย่างแฟ้มข้อมูลดังภาคผนวก

4.2.2 การทดลองใช้โปรแกรม PAC_SND ส่งข้อมูลเดิมในระยะ 10 เมตร โดยผ่านระบบวิทยุ

-พบว่าเกิดโอเวอร์รันเออร์เรอร์เกือบทุกครั้งที่ทดลอง ลักษณะการเกิดโอเวอร์รันเออร์เรอร์แสดงดังตัวอย่าง

ข้อมูลที่ใส่ส่ง ดังภาคผนวก ก..

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ข้อมูลที่รับได้ แสดงดังนี้

ครั้งที่ 1

-แองค์เมื่อรับข้อมูลได้ 384 ไบต์

WELCOME TO TURBO ASSEMBLER 1.0

This README file contains important, last minute information about Turbo Assembler 1.0. The HELPME!.DOC file on the TURBO ASSEMBLER/INSTALLATION disk also answers many common Technical Support questions.

TABLE OF CONTENTS

1. How to Get Help
2. Installation

ครั้งที่ 2

-ไม่สามารถรับข้อมูลได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

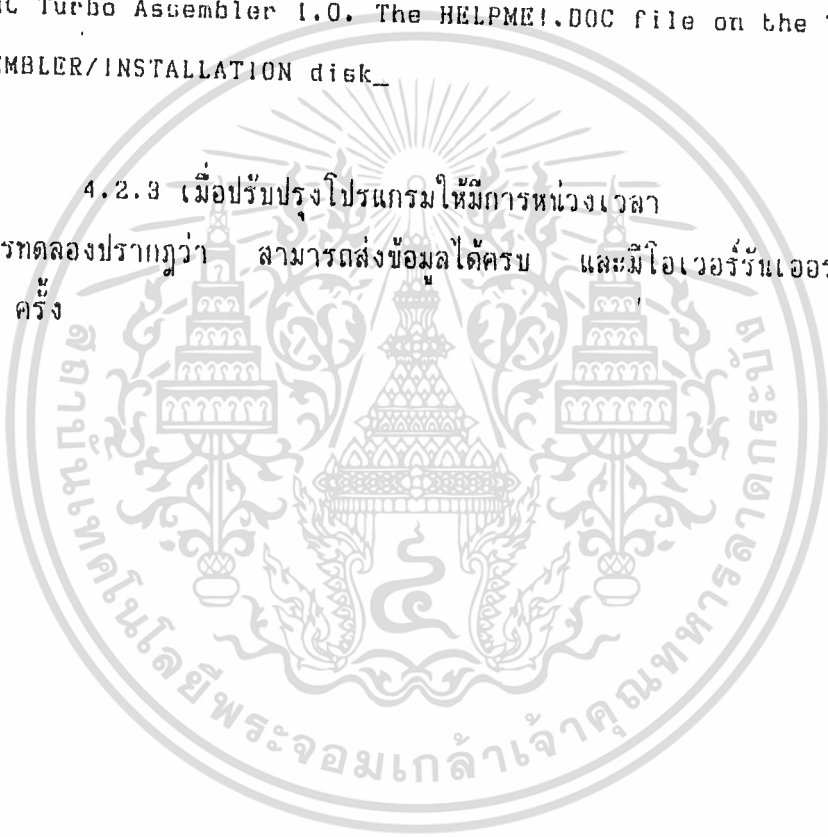
ครั้งที่ ๑

-แองค์เมื่อรับข้อมูลได้ 256 ไบต์

WELCOME TO TURBO ASSEMBLER 1.0

This README file contains important, last minute information about Turbo Assembler 1.0. The HELPME!.DOC file on the TURBO ASSEMBLER/INSTALLATION disk_

4.2.3 เมื่อปรับปรุงโปรแกรมให้มีการทวนวงเวลา
-ผลการทดลองปรากฏว่า สามารถส่งข้อมูลได้ครบ และมีโอเวอร์รันเออร์เรอร์เกิดขึ้น
นาน ๆ ครั้ง



บทสรุปและวิจารณ์

สิ่งที่เป็นตัวกำหนดคุณภาพของระบบ ก็คือค่าความผิดพลาดของข้อมูล ความผิดพลาดของข้อมูลที่เกิดขึ้นส่วนใหญ่เป็น โอเวอร์รันเออเรอร์ คือข้อมูลใหม่ไปทับกับข้อมูลเก่า ทำให้โปรแกรมเกิดการแอ่งค์เหตุเนื่องมาจากการอ่านค่ารีจิสเตอร์ของเครื่องรับช้ากว่าอัตราการส่งของคอมพิวเตอร์เครื่องส่ง เช่นการติดต่อระหว่าง IBM PC /XT กับ IBM PC/AT การแก้ไขก็คือ ตั้งค่าหน่วยเวลา ในการส่งเมื่อขาดเศษค่าเวลาที่สูญเสียไปกับการประมวลผลของโปรแกรม การเดินทางของข้อมูลในตัวกลาง ฯลฯ

ปัญหาอีกหนึ่งอย่างในการพัฒนาก็คือ อุปกรณ์หาไม่ได้ในเมืองไทย เช่น ไอซี M 6927 เป็นไอซี ที่ต้องซื้อจากประเทศญี่ปุ่น และเบอร์ที่หาได้ในประเทศไทยก็คือ MC 145450 เป็นไอซีโมเด็ม 1200 บิตต่อวินาที (ดูภาคผนวก ข.) แต่มีปัญหาที่ค่าออฟซีลเลเตอร์ไม่มีขาย ต้องจ้างโรงงานฝน ซึ่งมีราคาแพง

และสุดท้ายนี้จะขอกล่าวถึงโปรโตคอล เอกซ์โมเด็ม-ซีอาร์ซี (XMODEM-CRC) ซึ่งมีประสิทธิภาพดีกว่า เอกซ์โมเด็มธรรมดา เพื่อการปรับปรุงการลดความผิดพลาด สำหรับผู้สนใจให้หาอ่านได้ใน C Programmer's Guide to Serial Communication ของ Joe Campbell และผู้แต่งหวังว่าโครงการนี้คงจะมีประโยชน์ต่อส่วนรวมบ้างไม่มากก็น้อย.

ภาคผนวก ก.

ก. โปรแกรม PAC_SND

```
joint group cd_seg, data_seg
    assume cs:joint, ds:joint

cd_seg segment public
extrn xmit:near
extrn cls :near
extrn publern:near
extrn cr_lf:near
extrn port_status:near
extrn kbd:near
; #####
; # #
; # This file use for pack data 128 byte and send to #
; # uart 8250 #
; #####
soh equ 01h
ack equ 06h
nak equ 15h

public load_file
org 100h

load_file proc near
```

```

mov ah,00
mov dx,0000
mov al,10000111b
int 14h

mov dx,3f9h      mov dx,3f9h
in al,dx
and al,1fh
out dx,al

mov al,01h
mov dx,3f9h
out dx,al

mov al,02h
mov dx,3feh
out dx,al

cls:mov dx,3feh
in al,dx
and al,10h
jz cls

start:call cls

mov ah,09h ;display statement for user
mov dx,offset joint:string
int 21h

mov dx,offset joint:filename ;keyboard input buffer
mov joint:filename,40

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

mov ah,0ah
int 21h
mov bl,join:filename+1
add bl,02h
mov bh,00
mov joint:filename[bx],00;asciiz format

mov dx,offset joint:filename;get filename from buffer
add dx,02h
mov al,0
mov uh,3dh
int 21h ;open file

;check for file error
jnc fileok
rol ax,1
mov bx,ax
mov dx,join:error_pointer[bx]
mov bh,00
mov ah,09h
int 21h
mov dl,0dh
mov ah,02h
int 21h
jmp start

fileok:
mov handle,ax ;save file handle
mov bx,handle

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

mov cx,30000;define buffer=30kbyte
mov dx,offset joint:file_buf
mov ah,3fh
int 21h;read file

mov filesize,ax;save filesize

mov bx,handle
mov ah,3eh
int 21h

mov dh,1;adjust number of first packet to 1
xor bx,bx

sendheader:mov al,soh;soh
call xmit

numberpac : mov al,dh
            call xmit

onocom :add al,1
not al

            call xmit

mov cx,134;define number of byte in a packet

pac_128 :mov dl,offset joint:file_buf[bx] ;get character from buffer
mov al,dl

call xmit

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

dec filesize
or filesize,0h
jz halt

    inc bx

    mov byte_counter,bx

loop proc_128

inc dh

call chksum

call ack_nak

jmp sendheader

halt: mov al,03h;terminate program
call xmit;and send etx
int 20h

load_file endp

chksum proc near
push ax
push dx
push bx
push cx

sub byte_counter,134

mov hx,byte_counter

mov dl,offset joint:file_buf[bx]
mov al,dl

mov cx,133

rew:inc bx

mov dl,offset joint:file_buf[bx]

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

add al,dl
loop rew
call xmit
pop cx
pop bx
pop dx
push ax
ret
chksum endp

```

```

ack_nakprocnear
push ax
push bx
push dx

incognito:call port_status
mov dx,3f0h
in al,dx
sub al,nak
jz no
jmp yes

```

```

no:pop dx
pop bx
pop ax
sub bx,134
sub dh,1
ret

```

```

yes: pop dx

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

pop bx
pop ax
ret
stop: int 20h
ack_nakendp

```

```

delay procnear
mov cx,0ffffh
fo1:mov bx,0aaaaah
fo2:dec bx
jnz fo2
loop fo1
delay endp

```

```

cd_seg ends
data_seg segment public
extrn tm_out_mess:byte
extrn datap1:byte
extrn datap2:byte
extrn datap3:byte
string db 'enter filename d:\path filename.bin ',0dh,0ah,24h
filename db 40 dup(0)
file_buf db 30000 dup(0)
error_pointer dw 0
dw offset joint:err1
dw offset joint:err2
dw offset joint:err3
err1db 'invalid function number',0dh,0ah,24h
err2db 'file not found' ,0dh,0ah,24h

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

err3db 'path not found',0dh,0ah,24h
handle dw 0
filesizedw 0
byte_counter dw 0
data_seg ends
end load_file

```

ข. โปรแกรม RECV

```

joint group cd_seg,data_seg
assume cs:joint,ds:joint
cd_seg segment public
extrn xmit :near
org 100h
nak equ 15h
sch equ 01h
eot equ 04h
ack equ 06h
etx equ 03h
public receive
receive proc near

```

```

    mov ah,00 ;initialize com 1
    mov dx,0000
    mov al,10010111b ;1200Bd,no parity,2 stop,8-bit
    int 14h

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

mov dx,03FBh ;enable 8250
    in al,dx
    and al,7Fh
out dx,al    out dx,al

mov al,0fh
mov dx,3F9h
out dx,al .

mov dx,3fdh
xor al,al
out dx,al

mov ah,3ch ;create output file
mov dx,offset joint:fname
mov cx,0
int 21h
mov handle,ax

check: mov dx,3fdh
    in al,dx
    and al,01h
    jz check
    jmp recv

recv:xor bx,bx

seh_recv: call port_status

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

mov dx,3f8h
in al,dx
cmp al,soh
jne soh_recv
call port_status
rest_pac:
mov dx,3f8h
in al,dx
mov pac_num1,al
call port_status
mov dx,3f8h
in al,dx
mov pac_num2,al
call port_status
mov cx,134
pac_loop:call port_status
mov dx,3f8h
in al,dx
cmp al,etx
je savedisk
mov offset joint:buffer[bx],al
call dsply
inc bx
mov byte_counter,bx
loop pac_loop
call port_status
mov dx,3f8h

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

in al,dx
mov errchk,al
call checksum
jmp soh_recv

```

```

savedisk:
mov ah,40h.
mov bx,handle
mov dx,offset joint:buffer
add byte_counter,0ah
mov cx,byte_counter
int 21h
jc not_ready ;error message mismatch
mov ah,3eh
mov bx,handle
int 21h
jmp quit
not_ready:
mov dx,offset joint:tm_out_mess
mov ah,09h
int 21h
quit:int 20h
receive endp
public kbd
kbd proc near
push ax
push bx
push cx

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    push dx
    mov ah,1
    int 16h
    cmp al,27
    je quit2
    jmp getback
quit2:int 20h
getback:pop dx

pop cx
pop bx
pop ax
ret
khd endp

checksum proc near
push ax
push bx
push cx
push dx

sub byte_counter,127
mov bx,byte_counter
mov dl,offset joint:buffer[bx]
mov al,dl
mov cx,127
rew:inc bx
mov dl,offset joint:buffer[bx]
add al,dl
loop rew

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    jmp sendack

sendnak:mov al,nak

    call xmit

    pop dx

    pop cx
    pop cx

    pop bx

    pop ax

    sub bx,128

    ret

sendack:mov al,ack
    call xmit

    pop dx

    pop cx

    pop bx

    pop ax

    ret

checksum endp

```

```

dsply proc near

    push bx

    push cx

    push dx

    mov dl,al

    mov ah,02h

    int 21h

    pop dx

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    pop cx

    pop bx

        ret

display endp

public port_status

port_status proc near

push ax
;
push bx
push cx
push dx
gogo:call kbd

mov dx,3fdh
in al,dx
and al,01h
jz gogo

pop dx
pop cx
pop bx
pop ax

ret

port_status endp

delay proc near
mov cx,0ffffh

fool1:loop fool1

delay endp

```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

cd_seg ends

data_seg segment
    lin out mess db'modem not ready or',0dh,0ah
    db'transmit timeout-try again! ',0dh,0ah,24h
    buffer db 30000 dup(0)
    fname db 'noname.ibm',0
    byte_counter dw 0
    handle dw 0
    pac_num1 db 0
    pac_num2 db 0
    errchk db 0
data_seg ends
end receive

```

ค. โปรแกรมลบเส้นอื่น ๆ

```

-XMIT-

joint group cd_seg,data_seg
assume cs:joint,ds:joint

cd_seg segment public
public xmit,
org 100h
xmit proc near
    push bx
push cx

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

push dx
push ax

check:mov dx,3fdh
in al,dx
and al,20h
jz check

    pop ax

    mov dx,3f8h ;send character
out dx,al
clc
jmp done
not_ready:
mov dx,offset joint:tm_out_mess
mov ah,09h
int 21h
etc
pop ax
done: pop dx
pop cx
pop bx
ret

xmit endp
cd_seg ends

    data_seg segment public
public tm_out_mess
tm_out_mess db'transmit timeout-try again! ',0dh,0ah,24h
data_seg ends

end

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

-SCREEN-

```
cr equ 13
lf equ 10
joint grouped_seg
assume cs:join4J      assume ds:joint.
cd_seg segment public
org 100h
public cls
cls proc near
push ax
push bx
push cx
push dx
xor al,al
xor cx,cx
mov dh,24
mov dl,79
mov bh,7
mov ah,6
int 10h
mov dl,0
mov dh,0
mov bh,0
mov ah,2
int 10h
pop dx
pop cx
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

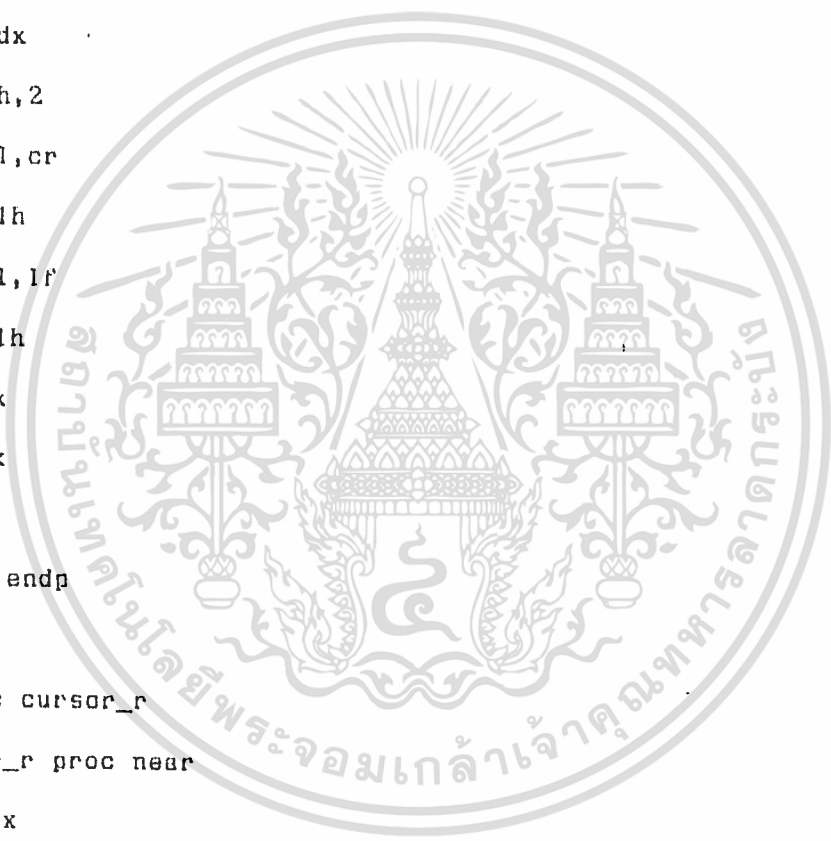
pop bx
pop ax
ret

cls endp

public cr_lf
cr_lf proc near
push ax
push dx
mov ah,2
mov dl,cr
int 21h
mov dl,lf
int 21h
pop dx
pop ax
ret
cr_lf endp

public cursor_r
cursor_r proc near
push ax
push bx
push cx
push dx
mov ax,03
mov bh,0
int 10
mov ah,2
inc dl

```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

cmp dl,79
jbe ok

call cr_lf

jmp done

ok:int 10h

done:pop dx      done:pop dx

pop cx

pop bx

pop ax

ret

cursor_r endp

cd_seg ends

end

-DISPLAY-

joint group cd_seg , data_seg
assume cs:joint , ds:joint

cd_seg segment public
extrn cursor_r:near
extrn cr_lf:near

org 100h

public write_char

write_char proc near

push ax

push bx

push cx

push dx

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

mov ah,9
mov bh,0
mov cx,1
mov al,dl
mov bl,7
int 10h
call cursor_r
pop dx
pop cx
pop bx
pop ax
ret
write_char endp

public repeat

repeat proc near
push cx
repeat1:
call write_char
loop repeat1
pop cx
ret
repeat endp

public pattern

pattern proc near
push ax

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

push cx
push dx
push si
pushf
cld
mov si, dx
pattern x:
lodsb
or al, al
jz end_patt
mov di, al
lodsb
mov cl, al
xor ch, ch
call repeat
jmp patternx
end_patt:
popf
pop dx
pop cx
pop ax
ret
pattern endp

cd_seg ends

```

```
data_seg segment public
```

```
dalapl label byte
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
db0dch,77
db0
datap2 label byte
db 0ddh,1
db 0f7h,75
db 0deh,1
db 0
datap3 label byte
db 0dch,77
db 0
data_seg ends
end
```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ง. แผ่นข้อมูลที่ไ้รับ-ส่ง

WELCOME TO TURBO ASSEMBLER 1.0

This README file contains important, last minute information about Turbo Assembler 1.0. The HELPME!.DOC file on the TURBO ASSEMBLER/INSTALLATION disk also answers many common Technical Support questions.

TABLE OF CONTENTS

1. How to Get Help
2. Installation
3. Important Notes
4. Manual Additions and Corrections
5. Files on the Disks

1. HOW TO GET HELP

If you have any problems, please read this file, the HELPME!.DOC

file, and the Turbo Assembler manuals first. If you still have

a

question and need assistance, help is available from the

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้พิมพ์ไปใช้โดยไม่ผ่านการคำ

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

following

sources:

1. Type GO BOR on the CompuServe bulletin board system for instant access to the Borland forums with their libraries of technical information and answers to common questions.

If you are not a member of CompuServe, see the enclosed special

offer, and write for full details on how to receive a free IntroPak containing a \$15 credit toward your first month's on-line charges.

2. Check with your local software dealer or users' group.
3. Write to us at the following address:

Borland International
Turbo Assembler Technical Support
1800 Green Hills Road
P.O. Box 660001
Scotts Valley, CA 95066-0001

Please remember to include your serial number or we will be unable to process your letter.

4. If you have an urgent problem that cannot wait and you have sent

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

in the license agreement from the front of your manual, you may

call the Borland Technical Support Department at (408) 430-5300.

Please have the following information ready before calling:

- a. Product name and serial number on your original distribution disk. Please have your serial number ready or we will be unable to process your call.
- b. Product version number. The version number for Turbo Assembler is displayed when you run the program.
- c. Computer brand, model, and the brands and model numbers of any additional hardware.
- d. Operating system and version number. (The version number can be determined by typing VER at the MSDOS prompt.)
- e. Contents of your AUTOEXEC.BAT file.
- f. Contents of your CONFIG.SYS file.

2. INSTALLATION

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

This disk contains a program called INSTALL.EXE that will assist you with the installation of Turbo Assembler 1.0. There are two options for installation:

1. Hard Disk - This option allows you to pick the subdirectories

where the files will be loaded.

2. Floppy Disk - This option will install the files necessary to use Turbo Assembler on a two-drive system. Be sure to have your formatted disks ready before you start.

To start the installation, change your current drive to the one that has the install program on it and type INSTALL. You will be given instructions in a box at the bottom of the screen for each prompt. For example, if you will be installing from drive A:, type

A:

INSTALL

You should read the rest of this README file to get further information about this release before you do the installation.

3. IMPORTANT NOTES

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

o Additional tips and answers to many common questions will be found

in the text file HELPME!.DOC, which has been installed to the same

location as this file (README). You can use the README.COM utility

to view the HELPME!.DOC file by giving the following command:

```
README HELPME!.DOC
```

o If you will be running INSTALL on a laptop or any other system

that uses an LCD display, you should set your system into black

and white mode before running INSTALL. You can do this from DOS

with the following command line:

```
mode hw80
```

or, you can force INSTALL to come up in black and white mode by

using the /b switch:

```
install /b
```

o Changed Switches for OBJXREF:

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

The /O option (object files directory) has been changed to the /D (directories) option. The switch now allows for multiple search directories to be specified. The new syntax follows:

```
OBJXREF /Ddir1[;dir2[;dir3]]
```

or

```
OBJXREF /Ddir1 [E/Ddir2] [E/Ddir3]
```

OBJXREF will search each of the directories in the specified order for all object and library files. If no /D option is used, only the current directory will be searched. However, if a /D option is used, the current directory will NOT be searched unless it is included in the directory list. For example, to first search the BORLAND directory for files and then search the current directory, you would type

```
OBJXREF /Dborland;
```

If multiple search directories are specified and a file matching the file specification is found, OBJXREF will include the file as part of the cross-reference. OBJXREF will only continue to search the other directories for the same file specification if the file specification contains wildcards.

A new option has been added to allow you to specify an output file where OBJXREF will send any reports generated. The new option is the /O option, and it has the following syntax:

```
OBJXREF myfile.obj /RU /Ofilename.ext
```

By default, all output is sent to the console.

o Turbo Linker (TLINK) Notes:

- 1) When linking with the /V option, TLINK will initialize all segments. If you have a program that runs differently when linked with debug information, you have an uninitialized variable somewhere.
- 2) TLINK loads last any segments of class 'STACK' even if they are part of DGROUP.

5. MANUAL ADDITIONS AND CORRECTIONS

Additions and corrections to the manual will be found in the text

file MANUAL.DOC, which has been installed to the same location as

this file (README). You can use the README.COM utility to view the

MANUAL.DOC file by giving the following commands:

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นิยมนำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้.

README MANUAL.DOC

6. FILES ON THE DISKS

TURBO ASSEMBLER/INSTALLATION

INSTALL EXE - Installation program
README COM - Reads this README
TASM EXE - Turbo Assembler
TCREF EXE - Turbo Assembler cross-reference utility
MAKE EXE - Program for managing projects
TLINK EXE - Borland Turbo Linker
TLIB EXE - Borland Turbo Librarian
OBJXREF COM - Object file cross-reference utility
GREP COM - Turbo GREP program
TOUCH COM - Program that updates a file's date and time
MANUAL DOC - Last minute additions and corrections to the manual
HELPME! DOC - Text file with the answers to many common questions. Please read HELPME!.DOC before contacting Technical Support
README - This file
MMACROS ARC - MASM Mode Macros
BIOS.INC - Equates representing the BIOS services
DOS.INC - Equates representing the DOS services
KBD.INC - Equates representing the IBM PC keyboard keystrokes

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นิยมนำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

MBIOS.MAC - Macros used to interface with the IBM PC BIOS.

MASM Mode

MDOS.MAC - Macros to interface with DOS. MASM Mode

MMACROS.MAC - General macros. Masm Mode

TURBO DEBUGGER UTILITIES/EXAMPLES

This diskette also contains files for use with the Turbo Debugger.

Please see the Turbo Debugger README and manuals for a complete description of these files.

TDCONVRT EXE - Turbo Debugger utility

TDPACK EXE - Turbo Debugger utility

TDNMI COM - Turbo Debugger utility

UNPACK COM - Program to unpack the .ARC files

TDEXMPL ARC - Packed file that contains Turbo Debugger examples

TAEXMPL1 ARC - Ideal mode example and macros

BIOS.INC - Equates representing the BIOS services

BIOSMAC.DOC - Documentation for macros implemented in ?BIOS.

MAC

DEMO1.BAT - Batch file to make Whereis.exe

DOS.INC - Equates representing the DOS services

DOSMAC.DOC - Documentation for macros implemented in ?DOS.MAC

IASCII2S.ASM - Routine to display an ASCII2 string

IBIOS.MAC - Ideal mode macros used to interface with the IBM

PC BIOS

IBYTECPY.ASM - Routine that copies an array of bytes from one location to another

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

IDELCHAR.ASM - Routine that deletes n characters from a
 Pascal-style string starting at position i

IDOS.INC - Type declarations used to interface with DOS

IDOS.MAC - Macros to interface with DOS. Ideal Mode

IFINDBYT.ASM - Routine that searches an array of bytes
 looking for a
 value; routine that returns the length of an
 ASCIIZ string.

IFINDFIL.ASM - Routine that does a recursive search of a
 hard disk
 looking for a file

IFINDREP.ASM - Routine that searches through an array of
 bytes,
 replacing all instances of a value with
 another value.

ILOCSTRG.ASM - Routine that searches through memory looking
 for the
 nth Pascal-style string

IMACROS.MAC - General macros. Ideal Mode

IPARAM.ASM - Routines to return the # of command-line
 parameters and
 the contents of an individual parameter

IPARSEFN.ASM - Routine that parses a file name into Drive,
 Path, and File spec

ISKPWHIT.ASM - Routine to advance a pointer past any
 whitespace in
 an array of bytes

IWHEREIS.ASM - Main module for Whereis program

IWHGLOBL.INC - Global declarations for Whereis

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

IWRITEPS.ASM - Routine to display a Pascal-style string
KBD.INC - Equates representing the IBM PC keyboard keystrokes
WHEREIS.DOC - Documentation for Whereis file locator program
WHEREIS.MAK - Make file for creating Whereis.exe using the
Make utility
WHUSAGE.INC - Usage screen text

TAEXMPL2.ARC - Simplified (FILT) and standard segmentation
(SHOW87)

examples

CONVERT1.INC - Converts from binary numbers to strings
CONVERT2.INC - Converts from floating point to integer
numbers
DEMO3.BAT - Batch file to make Filtr.exe text filter program
DEMO2.BAT - Batch file to build Show87
FILTR.ASM - Text filter main file
FILTR.DOC - Documentation file for Filtr.exe text filter
program
FILTR.MAK - Make file for creating Filtr.exe using the make
utility
FILTR.RSP - Linker response file for linking Filtr.exe
GLOBALS.INC - Declares procedures and symbols needed by the
modules

of FILT

OBYTEHI.ASM - Processes the current high byte value in AL
OBYTLOW.ASM - Processes the current low byte value in AL
OBYTENRM.ASM - Processes the current normal byte value in AL
OCMDNUM.ASM - Parses the command line
OERREXIT.ASM - Handles the printing of error messages

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

OFCREATE.ASM - Creates or truncates a file; the file is opened

for writing

OFILALOC.ASM - Allocates a read/write buffer

OFILCLOS.ASM - Closes a file

OFILFIL.ASM - Fills a read buffer

OFILFREE.ASM - Deallocates a read/write buffer

OFILREAD.ASM - Reads from a file

OFILWRIT.ASM - Writes to a file

OFLFLUSH.ASM - Flushes the write buffer

OFOPENR.ASM - Opens a file for reading

OFOPENW.ASM - Opens a file for writing

OLOCASE.ASM - Routine converts a letter to lowercase

OPARSCMD.ASM - Parses the command line

OPRCHYTE.ASM - Processes a byte

OPROCDOC.ASM - Processes an entire document, one line at a time

OPROCLIN.ASM - Processes a line

OSETOPTN.ASM - Sets input buffer size; sets output buffer size; sets the byte to end lines; clears all tab stops; sets a tab stop; truncates lines to a specified length; sets the left margin; deletes the left margin; displays a help message

OSPACES.ASM - Stores a specified number of spaces at particular location

OSTORSPC.ASM - Stores any currently stored spaces

OSTORTAB.ASM - Stores a tab character if there are any spaces to compress

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

OTABCLR.ASM - Clears all tab stops
OTABNEXT.ASM - Returns the number of spaces to the next tab
stop
OTABRSET.ASM - Sets tab stops every eight columns
OTABSET.ASM - Sets a tab stop at a specified column location
OUPCASE.ASM - Routine converts a letter to uppercase
SHOW07.ASM - Standard Segmentation Demo Program
SHOW07.DOC - Documentation for Show07 0087 register display
program
VIDEO1.INC - Routines to display text and control the screen
VIDEO2.INC - Routines to display formatted text strings
CHAPXPL.ARC - Examples Programs from the Manual

Note: This file is not automatically unpacked when Turbo
Assembler
is installed to a hard disk. To unpack these files
you will
first need to switch to the drive and subdirectory
containing

this file and then give the command

unpack /r chapxmpl

HELLO.ASM - Pg 9

HELLO2.ASM - Pg 13

HELLOPRN.ASM - Pg 15

REVERSE.ASM - Pg 16

ECHOCHAR.ASM - Pg 56

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

MODCHAR.ASM - Pg 80
DELAY.ASM - Pg 88
DSLYSTR.ASM - Pg 109
USE_ES.ASM - Pg 110
STDSEG.ASM - Pg 115
STRINGS.ASM - Pg 132
PRNTSTR.ASM - Pg 175
CNTWORDS.ASM - Pg 181
MAIN.ASM - Pg 208
SUB1.ASM - Pg 208
PLUSONE.C - Pg 285
PLUSONE.ASM - Pg 285
SQRTBLE.C - Pg 289
SQRTBLE2.C - Pg 289
STRINGUP.C - Pg 295
DOTOTAL.ASM - Pg 308
SHOWTOT.C - Pg 309
DOTOTAL2.ASM - Pg 309
TOGLFLAG.C - Pg 316
TOGLFLAG.ASM - Pg 316
CALLCT.C - Pg 330
COUNT.ASM - Pg 330
COUNTLG.ASM - Pg 332
CALCAVG.C - Pg 337
AVERAGE.ASM - Pg 337
SAMPLE.PAS - Pg 347
ASMPROC.ASM - Pg 347
TSAMPLE.PAS - Pg 348
HEXTEST.PAS - Pg 361

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

HEXSTR.ASM - Pg 360
XCHANGE.PAS - Pg 363
XCHANGE.ASM - Pg 363
ENVTEST.PAS - Pg 366
ENVSTR.ASM - Pg 366
SCROLLH.PRO Pg 391
SCROL.ASM Pg 392
MYPROLOG.PRO - Pg 395
MYASM.ASM - Pg 395
ADDPRO.PRO - Pg 397
ADD.ASM - Pg 397
SHOWNESS.PRO - Pg 399
FROM_ASM.ASM - Pg 400
SHOWNEW.PRO - Pg 400
FROM_NEW.ASM - Pg 401
FUNC.PRO - Pg 402
IFUNC.ASM - Pg 402
MULT_C&D.ASM - Pg 475
PRIMES.ASM - Pg 526
MASEXAMPL.ASM - Pg 555
IDLEXAMPL.ASM - Pg 557

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



MOTOROLA

MC145450

Advance Information

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 RTS-CTS Delay Options
- Soft Turn-Off Capability
- Answer Back Tone Generator (US and CCITT Tones)
- Carrier Detect Input
- 22 Pin Package

CMOS

1200 BAUD FSK MODEM

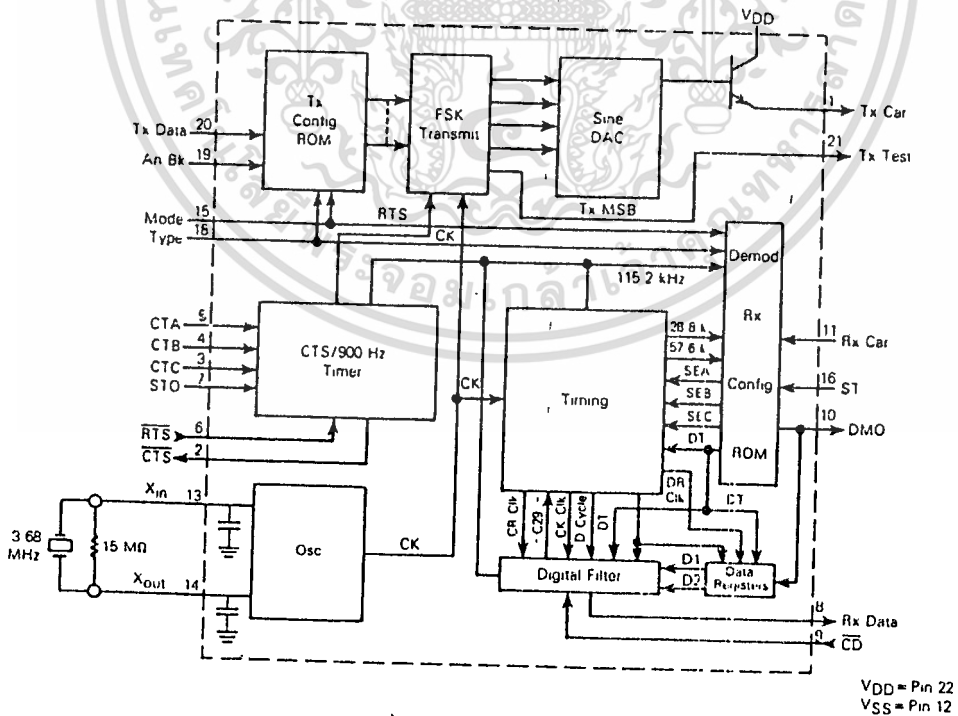


L SUFFIX
CERAMIC PACKAGE
CASE 736



P SUFFIX
PLASTIC PACKAGE
CASE 706

**MC145450 1200 BAUD FSK MODEM
BLOCK DIAGRAM**



VDD = Pin 22
VSS = Pin 12

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

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

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

PIN ASSIGNMENTS

Tx Car	1	22	V _{DD}
CTS	2	21	Tx Test
CTC	3	20	Tx Data
CTB	4	19	An Bk
CTA	5	18	Type
RTS	6	17	N C
STO	7	16	ST
Rx Data	8	15	Mode
CD	9	14	X _{out}
DMO	10	13	X _{in}
Rx Car	11	12	V _{SS}

RECOMMENDED OPERATING CONDITIONS

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

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 Pins 3-7, 9, 15, 16, 18, 19, 20 Pin 13, 11	V _{IH}	2.8 4.0	-	-	V
Input Low Voltage Pins 3-7, 9, 15, 16, 18, 19, 20 Pin 13, 11	V _{IL}	-	-	0.5 0.6	V
Input Current All Inputs (V _{IL} = 0 V) All Inputs Except Pins 11, 13, (V _{IH} > 2.8 V) (Note 1)	I _{in}	-	-	-5.0 600	μA
Output High Current (V _{OH} = 2.4 V) Pins 2, 8 (Test Load A) Pins 10, 21 (Test Load B)	I _{OH}	0.75 0.75	-	-	mA
Output Low Current (V _{OL} = 0.4 V) Pins 2, 8 (Test Load A) Pins 10, 21 (Test Load B)	I _{OL}	1.2 0.6	-	-	mA
Operating Current	I _{OD}	-	2.5	6	mA
Input Capacitance All Except Pin 13 Pin 13 (X _{in})	C _{in}	-	-	12 8	pF
Output Capacitance All Except Pin 14 Pin 14 (X _{out})	C _{ou}	-	-	12 -	pF
Transmit Audio Signal Level (Pin 1 R _L = 10 kΩ (Note 2) Total Harmonic Distortion (2nd to 14th) (Note 2)	- THC	0.428 -	0.5 -50	0.578 -40	Vp-p 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 RTS to CTS	STO = Low t _{d(low)}	-	1	-	μs
Delay From RTS to CTS	STO = High t _{d(high)}	18.3	-	21.7	ns

NOTES:

- 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.
- Measured in any mode using HP-3555B dB meter (or equivalent) with 3 kHz flat filtering.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

MC145450

PIN DESCRIPTIONS

VDD, POSITIVE POWER SUPPLY (PIN 22)
This is nominally 5.0 V.

VSS, 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) ($\pm 10\%$) and offset by a dc bias of $0.5 V_{DD}$ ($\pm 10\%$). The output load should be 10 kilohms or greater.

CTS, CLEAR TO SEND (PIN 2)
The clear to send output goes low in response to a high-to-low transition of RTS following a selected delay (see CTA, CTB, CTC pin description). This output goes high immediately after loss of RTS. During the time following activation of RTS and before the activation of 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.

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 CTS to remain active for 20 ms following the loss of 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 CD is not active.

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}, OSCILLATOR 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 $\pm 0.01\%$. When driven by a crystal, a 15 megohm resistor 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 CTS will go to a high state, regardless of the state of RTS (see Figure 11).

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.

MC145450

FIGURE 1 - An Bk AND RTS-CTS TIMING

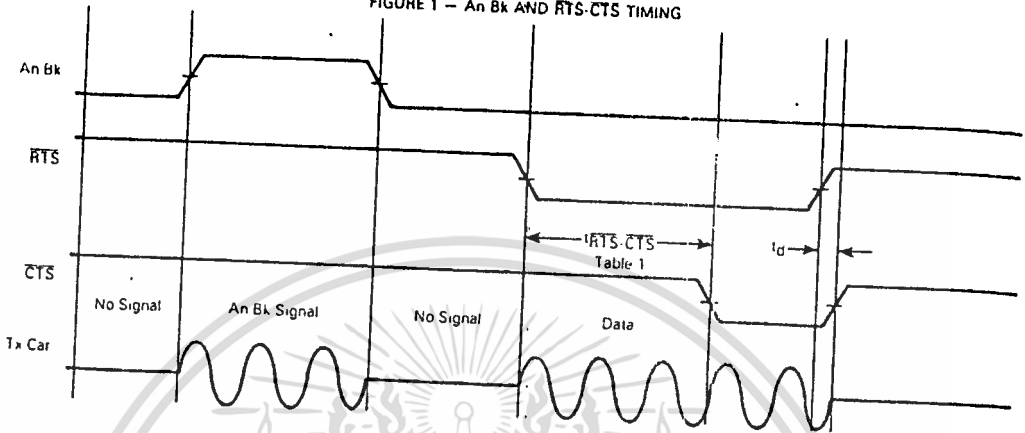


TABLE 1 - RTS-CTS DELAY TIMES

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

Data = 0 = Space
= 1 = Mark

* Crystal frequency = 3.6864 MHz

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

FIGURE 2 — STO TIMING

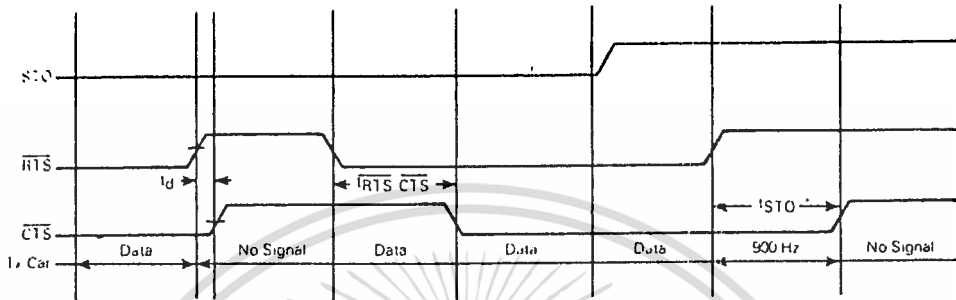
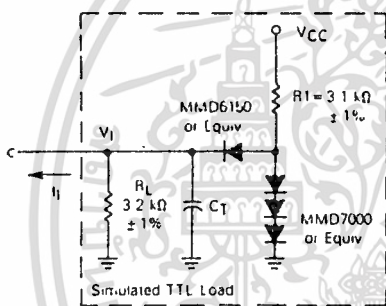
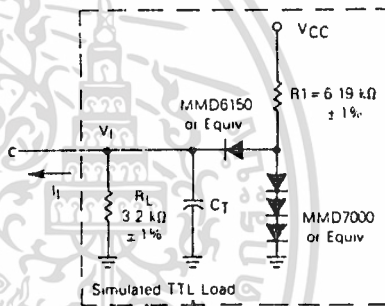


FIGURE 3 — OUTPUT TEST LOAD A



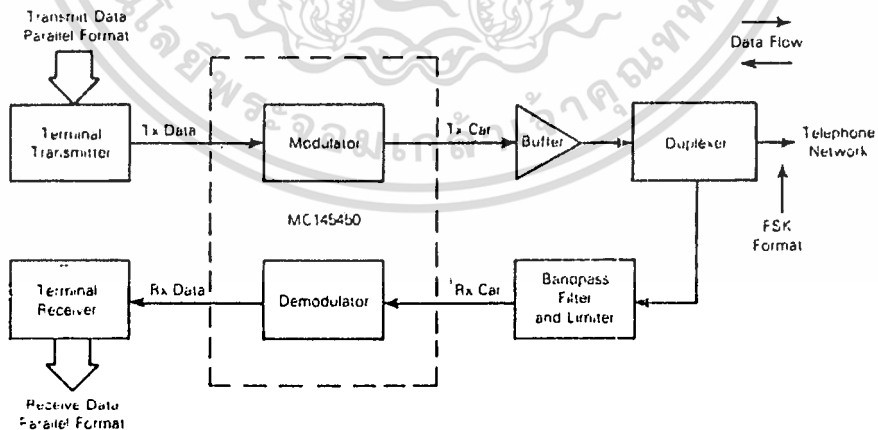
$C_T = 20\text{ pF}$ = total parasitic capacitance, which includes probe, wiring, and load capacitances

FIGURE 4 — OUTPUT TEST LOAD B



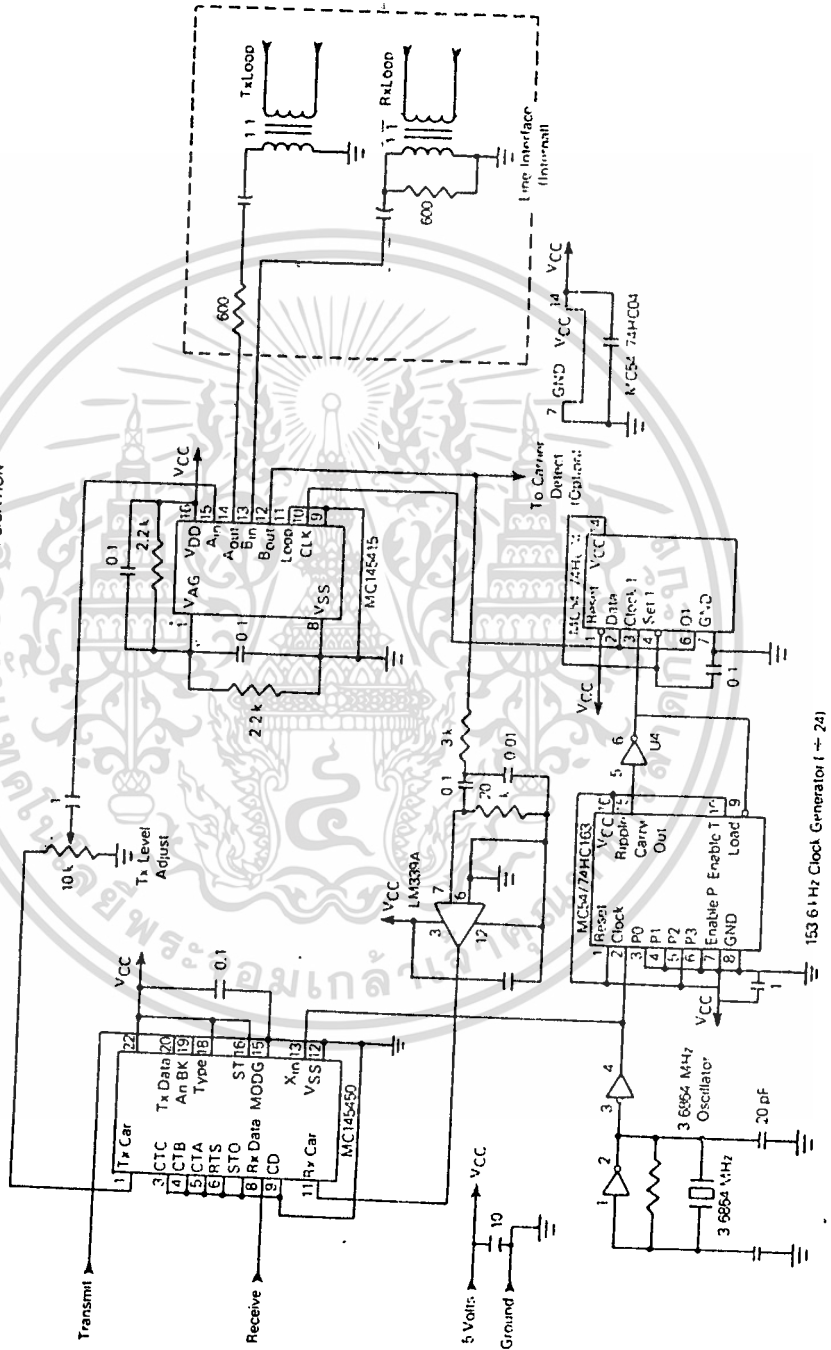
$C_T = 20\text{ pF}$ = total parasitic capacitance, which includes probe, wiring, and load capacitances

FIGURE 5 — TYPICAL MEDIUM-SPEED MODEM APPLICATION



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

FIGURE 6 -- TYPICAL 1200 BAUD 4 WIRE MODEM APPLICATION



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

ปริญญาบัตรเล่มนี้ สามารถสล่งมาโดยดี เนื่องเพราะความเอื้อเฟื้อจากบุคคลหลาย ๆ ท่าน โดยเฉพาะอาจารย์ที่ปรึกษา และเพื่อน ๆ ที่ช่วยให้คำปรึกษา และช่วยพิมพ์รายงาน

ข้อผิดพลาด หรือส่วนใดที่ไม่เรียบร้อย ผู้จัดทำออกภัยและรับผิดชอบ แต่เพียงผู้เดียว



หนังสืออ้างอิง

1. น.ต.ดร. ไนศาล สวงวหนู รศ.ยีนุภาพวรณ, "การสื่อสารข้อมูล และ ไมโครคอมพิวเตอร์เน็ตเวิร์ค, ซีเอ็ด, 2532
2. ปีเตอร์ นอร์ทัน และ จอนห์โซธา, "เทคนิคการเขียน โปรแกรมภาษาแอสเซมบลีสำหรับเครื่อง IBM PC ", ซีเอ็ด, 2532
3. RAY DUNCAN, " ADVANCE MSDOS ", MICROSOFT PRESS, 1986
4. JOE CAMPBELL, " C PROGRAMMER'S GUIDE TO SERIAL COMMUNICATIONS ", HOWARD W. SAM AND COMPANY, 1989
5. DOUGLAS V. HALL, " MICROPROCESSORS AND INTERFACING PROGRAMMING AND HARDWARE ", Mc. GRAW HILL, 1986