ปีการศึกษา 2532

เรื่องการสื่อสารข้อมูลผ่านสายส่งกำลัง

จัดทำโดย

1. นาย ณรงค์ศักดิ์ ตันติวิริยพันธ์ 31.3303

2. นาย ภานุพันธ์ งานเจริญกุล 31.3309

3. นาย สมชาย เกิดแก้ว 31.3317

อาจารย์ที่ปรึกษา

ผศ.ดร. กนก เจนจิระพงศ์เวช

อาจารย์ กฤดากร กล่อมการ

026993

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ปริญญาญานิพนธ์ปีการศึกษา 2532

ภาควิชา เทคนิคอุตสาหกรรม

เรื่อง การสื่อสารข้อมูลผ่านสายส่งกำลัง

จัดทำโดย

1. นาย ณรงค์ศักดิ์ ตันติวิริยพันธุ์
2. นาย ภานุพันธ์ งานเจริญกุล
3. นาย สมชาย เกิดแก้ว

..... อาจารย์ที่ปรึกษา

(.....)

..... อาจารย์ที่ปรึกษา

(.....)

..... อาจารย์ที่ปรึกษา

(.....)

026993

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การสื่อสารข้อมูลผ่านทางสายส่งกำลังไฟฟ้า

นาย ณรงค์ศักดิ์ ตันติวิริยพันธุ์
นาย ภานุพันธ์ งานเจริญกุล
นาย สมชาย เกิดแก้ว
ผศ.ดร. กนก เจนจิระพงศ์เวช
อ. กฤดากร กล่อมการ
อาจารย์ที่ปรึกษา
ปีการศึกษา 2532

บทคัดย่อ

การสื่อสารข้อมูลภายในสายส่งกำลัง 220V สิ่งที่จำเป็นมากซึ่งต้องพิจารณาถึง ในการออกแบบระบบคือ สัญญาณรบกวนต่างๆ รวมทั้งแรงดันกระชากเนื่องจาก การเปิดปิดของอุปกรณ์กำลัง

โครงงานนี้เสนอแนวทาง และทดลองส่งข้อมูล คอมพิวเตอร์ผ่านสายส่งกำลังไฟฟ้า โดยออกแบบระบบเป็น spread spectrum แบบ direct sequence สัญญาณรบกวนต่างๆจะถูกลดลงโดยการ สะสมพลังงานและ พิจารณาดูว่าเป็นสภาวะใด การรบกวนที่เกิดขึ้นในระหว่างบิทข้อมูล จะลดลงได้

แนวทางการประยุกต์ ใช้งานอื่นๆ ได้ออกแบบวงจร ควบคุมอุปกรณ์ไฟฟ้า โดยใช้โปรแกรมควบคุมด้วยคอมพิวเตอร์ และคาดว่าจะเป็นประโยชน์มาก ในการควบคุมอัตโนมัติต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

AC LINE MODEM.

MR. NARONGSAK TUNTIVIRIYAPHAN

MR. PANOOPAN NGANJALEANKUN

MR. SOMCHAI GIRDGAEW

ASS.PROF.DR. KANOK JENJIRAPONGWET

MR. KIDDAKON KLOMKARN

ADVISOR.

1989

ABSTRACT

In case of power line telecommunication the interference and it pollution are discutted to design system.

This thesis presents data communicated within power line AC 220V by direct sequence spread spectrum system. It noise and interference are reduced by integrator and dump circuit.

It application we guide in electric equipment controller by personal computer it aspected many usefull in automatic control in futher.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

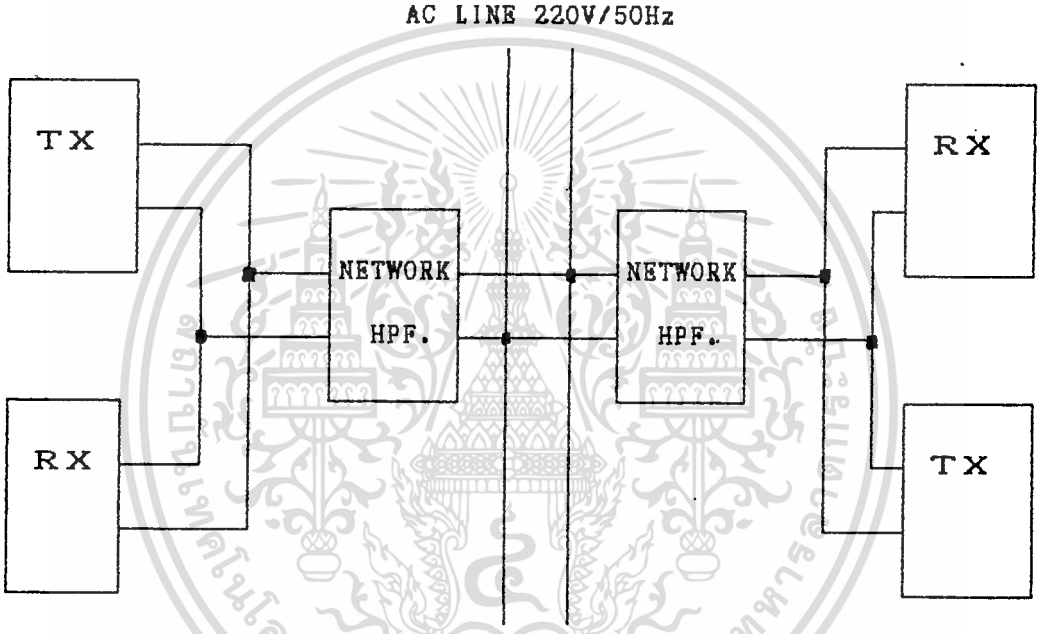
เรื่อง	หน้า
บทนำ	
หลักการทำงานของเครื่องสื่อสารข้อมูลทาง AC LINE	1
การสื่อสารข้อมูลแบบอนุกรม	
ประเภทของการเชื่อมโยงเพื่อการสื่อสาร	3
มาตรฐาน RS 232C	5
อักขระควบคุมการสื่อสาร	12
โปรโตคอลและมาตรฐานการเชื่อมโยง	16
ทฤษฎีและหลักการ	
ระบบ Spread Spectrum คืออะไร	18
การ Demodulation สัญญาณ Spread Spectrum	21
แนวทางการออกแบบวงจร	
การออกแบบ Max Code Sequence	25
การสร้างสัญญาณ NRZI	26
การ Decode สัญญาณ NRZI	27
การ Recovery สัญญาณ Clock จากข้อมูล Biphase Code	28
BPSK Modulation	30
วงจรและการทำงาน	
วงจรภาครับ	32
วงจรภาคส่ง	32
วงจรควบคุมอุปกรณ์ไฟฟ้า	33
การทดลองและผลการทดลอง	39
ภาคผนวก	
โปรแกรมควบคุมการสื่อสารและควบคุมอุปกรณ์ไฟฟ้า	ก
เอกสารอ้างอิง	ข

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

หลักการทำงานของเครื่องสื่อสารข้อมูลทาง AC LINE

หลักการทำงานของเครื่องนี้ ถ้าจะให้สมบูรณ์แล้วต้องมีการติดต่อสื่อสารแบบ Full Duplex ซึ่งติดต่อได้ 2 ทิศทาง คือ มีทั้งรับและส่งในตัวเดียวกัน ตาม block diagram รูป A

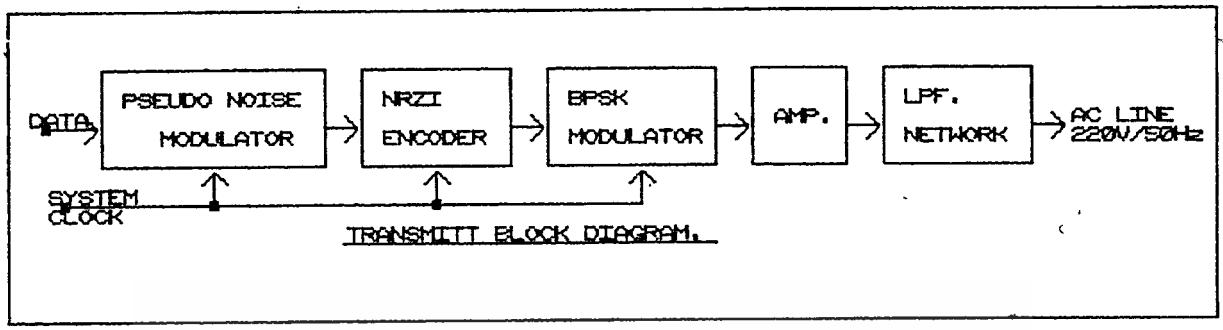


รูป A

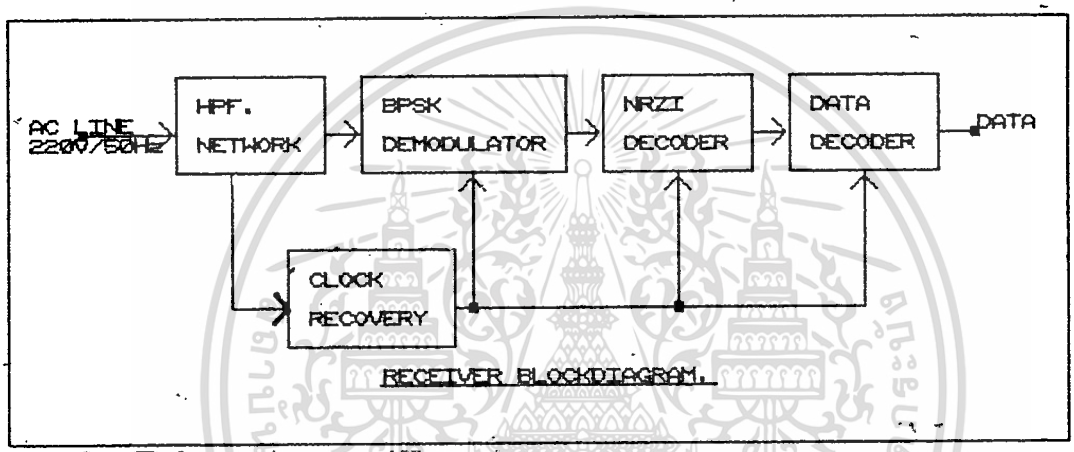
แต่ใน project นี้ จะทำการสื่อสารทิศทางเดียวเท่านั้น มีหลักการทำงานดัง BLOCK DIAGRAM ต่อไปนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

SYSTEM BLOCK DIAGRAM.



TRANSMITT BLOCK DIAGRAM.



RECEIVER BLOCKDIAGRAM.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2

การสื่อสารข้อมูลแบบอนุกรม

ประเภทของการเชื่อมโยงเพื่อการติดต่อสื่อสาร

การเชื่อมโยงแบบจุดต่อจุด (Point to Point) ถือได้ว่าเป็นแบบพื้นฐานของข่ายการสื่อสารโดยมีอุปกรณ์รับส่งข้อมูล 2 ชุดที่ต้องการสื่อสารถึงกันโดยเชื่อมโยงกันด้วยสายสื่อสาร ในข่ายการสื่อสารประเภทนี้จะไม่พิจารณากันที่ความยาวของสายส่งผ่านข้อมูลซึ่งอาจจะยาวเพียง 3 เมตร หรือ 10000 กิโลเมตรก็ได้ ส่วนสายส่งข้อมูลที่ใช้อาจจะเป็นได้ทั้งสายส่งทางเดียว สายส่งกึ่งทางคู่หรือสายส่งทางคู่สมบูรณ์ก็ได้ และวิธีการส่งผ่านข้อมูล สามารถเป็นไปได้ทั้งสองรูปแบบซิงโครนัสหรืออะซิงโครนัสก็ได้ บางครั้งข่ายการสื่อสารแบบนี้อาจจัดให้มีการรวมศูนย์ และมีสถานีรับส่งปลายทางหลายๆจุดซึ่งในรูปแบบนี้จะมีชื่อเรียกกันว่า "Star Network"

การเชื่อมโยงแบบหลายจุดบนช่องทางส่งผ่านข้อมูลช่องเดียว (Multidrop or Multipoint) เพื่อต้องการค่าใช้จ่ายค่าเช่าของช่องทางการส่งผ่านข้อมูล

อุปกรณ์รับส่งข้อมูลที่เชื่อมโยงกับศูนย์กลาง อาจจะใช้อุปกรณ์รับส่งข้อมูลชนิดที่ไม่มีบัฟเฟอร์ แต่ส่วนใหญ่แล้วจะใช้อุปกรณ์รับส่งข้อมูลชนิดที่มีบัฟเฟอร์ทั้งสิ้น โดยมีเหตุผลอยู่ 2 ประการ

1.เพื่อเพิ่มประสิทธิภาพในการรับส่งข้อมูลด้วยการเก็บข้อมูลไว้ ก่อนที่จะมีการส่งได้หรือเมื่อมีข้อมูลส่งมาก็จะสามารถเก็บไว้ได้ทันที

2.ประหยัดช่วงเวลาใช้ การใช้ช่องทางการสื่อสารในขณะที่ผู้ใช้กำลังป้อนข้อมูลจะยังไม่ต้องส่งผ่านไปทันทีจะรอจนกว่าข้อมูลเต็มบัฟเฟอร์แล้วส่งไปทีเดียวพร้อม ๆ กันครั้งละมาก เลยทีเดียว

3.การเชื่อมโยงเครือข่ายแบบสลับช่องช่องทางการสื่อสารได้(Switched Network)

สำหรับเครือข่ายการสื่อสารระบบนี้มีใช้กัน 4 รูปแบบ

1.เครือข่ายสารโทรศัพท์(The telephone Networks)

2.เครือข่ายสื่อสารเทลเล็กซ์(The telex/TWX Networks)

3.เครือข่ายสื่อสาร packet switching(Packet Switching Networks)

4.เครือข่ายสื่อสาร specialized digital (Specialized Digital Networks)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เราได้กล่าวแล้วว่าถ้าหากช่องสัญญาณมีแบนด์วิดธ์กว้างเราก็สามารถส่งสัญญาณอัตราข้อมูล (Data rate)ได้มากขึ้น เมื่อข้อมูลหรือสัญญาณมีค่าระดับของสัญญาณมีมากกว่า 2 ระดับสัญญาณ เช่นในการมอดดูเลทสัญญาณหลายระดับ อัตราบิทจะมากกว่าอัตราบอด เมื่อเรามาพิจารณาถึงช่องทางการสื่อสารที่ไม่มีสัญญาณรบกวน(Niose)เกิดขึ้น ซึ่งมีแบนด์วิดธ์เท่ากับ X ถูกสุ่มด้วยอัตรา 2X/วินาที และถ้าช่องสัญญาณมีแบนด์วิดธ์ x เฮิร์ทซ์แล้วเราสามารถส่งสัญญาณที่แตกต่างกันได้ 2x ค่าได้ สำหรับข้อมูลดิจิตอลซึ่งระดับโวลเต็จต่างกันอยู่ 2 ค่า(0,1) เมื่อเป็นเช่นนี้การส่งสัญญาณพร้อมกัน 2 บิทข้อมูลซึ่งมีค่าโวลเต็จต่างกัน 4 ค่า ในหนึ่งช่วงเวลาเมื่อระดับสัญญาณเท่ากับ 2 ยกกำลัง n เราสามารถเข้ารหัสได้ n บิท ยิ่งกว่านั้นเราสามารถส่งข้อมูล 2nx บิทต่อวินาทีผ่านช่องส่งสัญญาณที่มีแบนด์วิดธ์ x เฮิร์ทซ์ ถ้ากำหนดให้ $L = 2^n$

L = จำนวนระดับสัญญาณ(signal level)

n = จำนวนบิท

เราสามารถแสดงความสัมพันธ์ได้ว่า

$C = 2nx$

$= 2x\text{Log}_2\ L$ บิทต่อวินาที

C = อัตราข้อมูลสูงสุดส่งผ่านช่องสัญญาณที่ไม่มีสัญญาณรบกวน

x = แบนด์วิดธ์ของสายโทรศัพท์ที่ไม่มีสัญญาณรบกวน

ตาราง แสดงความจุ(capacity)ของช่องสัญญาณสายโทรศัพท์ตามอุดมคติ

จำนวนบิทต่อการสุ่ม number of bit/sam	จำนวนระดับสัญญาณ signal level	อัตราข้อมูลสูงสุด maximum date rate
1	2	5.4
2	4	10.8
3	8	16.2

ตารางที่ 2.1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การสื่อสารข้อมูลรูปแบบต่างๆมักจะประกอบด้วยส่วนสำคัญ 3 ส่วน

1.คอมพิวเตอร์หรือเทอร์มินอลสำหรับส่งหรือรับ

2.อุปกรณ์อินเตอร์เฟสแบบอนุกรม(serial interface)ซึ่งมักจะเรียกกันว่า DTE (Data terminal equipment)

3.อุปกรณ์สื่อสารข้อมูล(Data communication equipment)หรืออุปกรณ์ซึ่งเรียกกันว่า DCE

รูปแบบของการติดต่อสื่อสารแบบอนุกรม

1.การส่งสัญญาณทางเดียว(Simplex transmission)หรือที่มักเรียกกันว่า"SPX" ซึ่งเป็นการส่งสัญญาณจาก A ไปสู่ B แต่ไม่สามารถจะส่งสัญญาณข้อมูลจาก B มาสู่ A ได้

2.การส่งสัญญาณกึ่งทางคู่ (Half duplex) มักเรียกกันว่า "HDX" สามารถส่งสัญญาณจาก A ไปสู่ B ได้และก็สามารถที่จะส่งสัญญาณจาก B ไปหา A ได้เหมือนกันแต่ว่าเหตุการณ์ที่เกิดขึ้นทั้งสองจะต้องไม่เกิดขึ้นในเวลาเดียวกัน

3.การส่งสัญญาณทางคู่ (Full duplex) ใช้เรียกคำย่อว่า "FDX"เป็นการส่งสัญญาณจาก A ไปสู่ B และB ส่งสัญญาณไปสู่ Aได้ในเวลาเดียวกันซึ่งเกิดขึ้นได้พร้อมๆกันทั้งสองเหตุการณ์

มาตรฐาน RS-232C

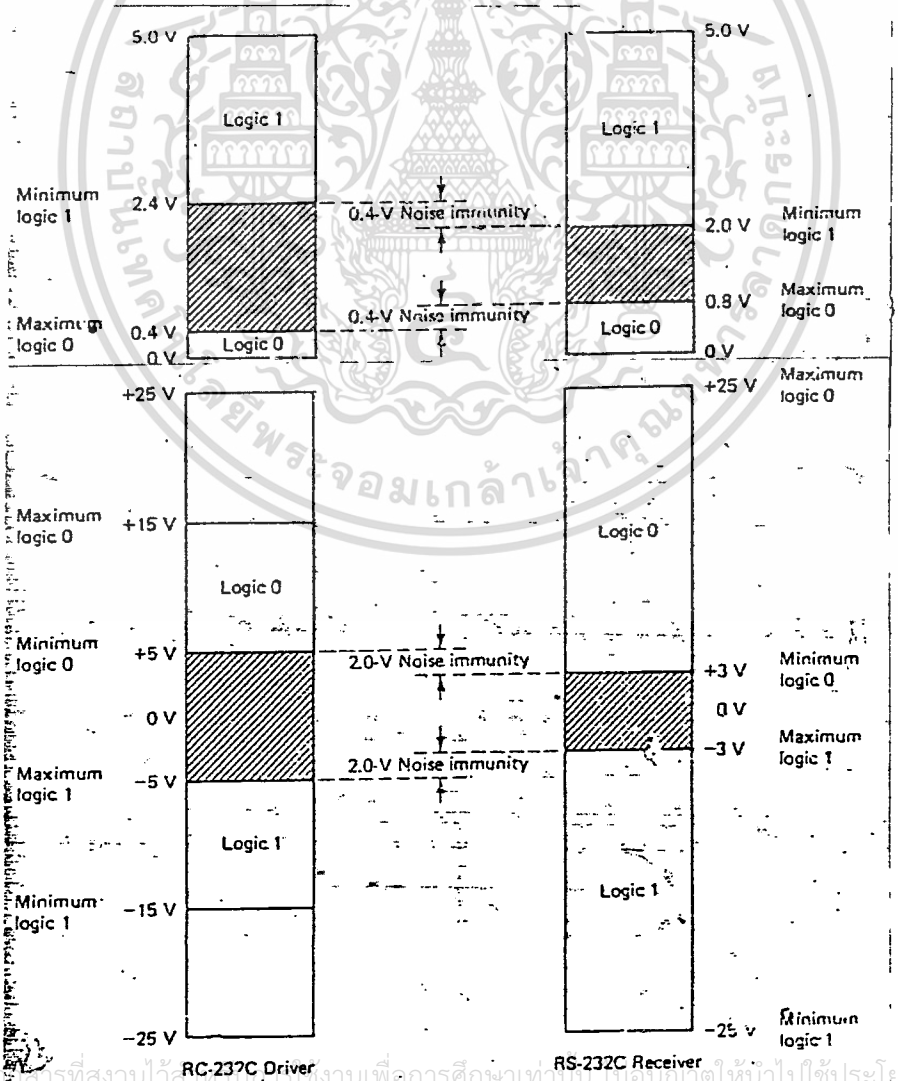
มาตรฐาน RS-232C มีใช้ครั้งแรกเมื่อปี พ.ศ 2505 ซึ่งใช้ในการอินเตอร์เฟสระหว่างอุปกรตัวส่งหรืออุปกรณ์ต้นทาง (SENDER DRIVER) และอุปกรณ์ปลายทาง (RECIVER TERMINATOR) โดยทางคัพท์ที่ใช้งานคอมพิวเตอร์จะหมายถึงเทอร์มินอลและโมเด็มตามลำดับ โดยเทอมินอลมักจะถูกเรียกว่าอุปกรณ์ DEC (DATA TERMINAL EQUIPMENT) และโมเด็มจะถูกเรียกว่า DCE (DATA COMUNICATION EQUIPMENT) ในทางทฤษฎีข้อกำหนดหลักที่บ่งบอกถึงข้อกำจัดการใช้งาน RS 232 c คืออัตราความเร็วในการส่งข้อมูล(บอดเรด)สูงสุดสำหรับสายเคเบิ้ลความยาว 50 ฟุต จะมีได้ไม่เกิน 19,200 บอด แต่ในทางปฏิบัติสามารถให้สายยาวกว่านี้ได้ แต่จะต้องส่งข้อมูลในอัตราที่ต่ำกว่าที่กำหนดนี้ หลายคนคงจะทราบกันมาบ้างแล้วว่า มาตรฐาน rs232c จะให้ระดับลอจิกแรงดัน -12 v และลอจิก 0 ด้วยแรงดัน +12 v

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คุณสมบัติทางไฟฟ้าของ RS 232 C

โดยแท้จริงแล้วรูปแบบของคอนเนคเตอร์ที่เป็นมาตรฐาน RS 232 C นั้นไม่ได้มีการระบุไว้แต่ต้น แต่มาภายหลังมักนิยมใช้ใช้คอนเนคเตอร์ ที่มีลักษณะคล้ายตัว D จำนวน 25 ขาเป็นหลักดังแสดง ให้เห็นทั้งตัวผู้(DB-25P)และแบบตัวเมีย(DB-25S) ซึ่งเรามักเรียกติดปากว่า คอนเนคเตอร์แบบ D-PIN 25 ขา

จาก รูปที่ 2.1 จะเห็นว่าผลจากการ เปรียบเทียบระดับลอจิกของมาตรฐานทั้งสอง มาตรฐานของ TTL จะใช้ระดับแรงดัน บวก แทนลอจิก 1 และมี NOISE IMMUNITY เท่ากับ 0.4 โวลท์ และมาตรฐาน RS 232C จะแตกต่างกันไปอย่างมาก โดยจะใช้แรงดันสูงขับออกทางเอาว์พุท (±12 โวลท์) ซึ่งจำเป็นต่อการใช้งานทั้งนี้เพื่อให้แน่ใจว่าการทำงานกับสายเคเบิ้ลที่ยาวจะมีความถูกต้องและเป็นไปได้ สำหรับ NOISE IMMUNTY จะกำหนดให้มีค่า 2 โวลท์ ทั้งนี้ก็เพื่อหลีกเลี่ยงสัญญาณรบกวนจากแหล่งอื่น ๆ ภายนอกที่สร้างปัญหาให้ระบบมาตรฐาน TTL



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ความหมายของ DCE และ DTE

DCE คือ อุปกรณ์ที่มีระบบการทำงานทำให้เกิดการเชื่อมต่อแหล่งที่เป็น DTE ตั้งแต่ 2 ชุด นอกจากนี้ยังใช้เปลี่ยนลักษณะของสัญญาณและสร้างรหัสสัญญาณต่าง ๆ ที่จำเป็นต่อการสื่อสารข้อมูลระหว่าง DTE ดั้งได้ยกตัวอย่างที่เห็นง่าย ๆ ของ DCE ก็คือ MODEM ส่วน DCE จะมีความหมายว่าเป็นอุปกรณ์ที่ประกอบด้วยตัวส่งขอมูล (DATA SOURCE) หรือตังรับข้อมูล (DATA SINK) หรือจะเป็นตัวรับข้อมูลหรือตัวรับข้อมูลก็ได้ โดยจะมีฟังก์ชั่น การทำงานต่าง ๆ อาทิเช่นหน่วยควบคุมลอจิก,บัฟเฟอร์และอุปกรณ์ อินพุท เอาว์พุท หรืออาจพูดง่าย ๆ คือ คอมพิวเตอร์นั้นเอง และโดยสภาพแท้จริงแล้ว DTE มักจะแทนด้วยแหล่งกำเนิดข้อมูลแหล่งแรก และแหล่งรับ ข้อมูลสุดท้าย อาทิเช่นเครื่องพิมพ์ จอมอนิเตอร์ เทอร์มินอล เป็นต้น

คุณสมบัติทางไฟฟ้าอีกข้อหนึ่งที่ถือว่าเป็นมาตรฐาน RS 232C ก็คือค่าของเวลาทรานซิชั่น (TRANSITOIN TIME) ก็หมายถึงช่วงเวลาที่เปลี่ยนจากระดับหนึ่งไปสู่อีกระดับหนึ่งจะมีได้ไม่เกิน 4% ของเวลา 1 บิท ดังนั้นถ้าอัตราความเร็วในการส่งเป็น 19,200 บอดจะมีทรานซิชั่นเกิดขึ้นได้ไม่เกิน 0.04 x (1 / 19,200) = 2.1 uSEC (2.1×10^{-6} วินาที) หรืออาจจะพูดได้ว่า ความยาวสายเคเบิ้ลที่เป็นแฟคเตอร์อย่างหนึ่งที่กำจัดเวลาทรานซิชั่นเพราะสายเคเบิ้ลยิ่งยาวเท่าใด ความจุ (CAPACITIVE LOAD) ของสายเคเบิ้ลยิ่งมาก อันเป็นผลให้เกิดทรานซิชั่นถูกทำให้ช้าลง ซึ่งในเรื่องนี้ได้มีการทดสอบกันมาแล้วว่าความยาวสายเคเบิ้ลมิได้ไม่เกิน 50 ฟุต

สัญญาณและการอินเตอร์เฟสทางแมคคานิคของ RS_232C

ในตารางที่ 2.1 จะแสดงถึงรายชื่อของสัญญาณทั้ง 25 ขา ซึ่งถือเป็นมาตรฐานการอินเตอร์เฟล RS 232C ขาที่เป็นข้อมูลก็คือกลุ่มที่เป็นแชนแนลปฐมภูมิ (primary channel) และ แชนแนลทุติยภูมิ(secondary channel) ส่วนขาสัญญาณควบคุม ซึ่งนับได้ว่ามีความสำคัญที่สร้างขึ้นมาเพื่อเป็นโปรโตคอลระหว่างโมเด็ม(DEC) และเทอร์มินอล(DTE) ก็คือขาสัญญาณ DTR,DSR,CTS,RTS และ DCD ซึ่งการทำงานของสัญญาณเหล่านี้จะนำไปอธิบายภายหลัง สำหรับขาสัญญาณ ring indicator (ขา 22) จะใช้กับโมเด็มที่เป็นแบบออโต้ไดอัล และขา baud-rate select (ขา 23) จะใช้สำหรับเลือกบอดเรตให้แก่โมเด็มทั้งภาคตัวส่งและตัวรับ

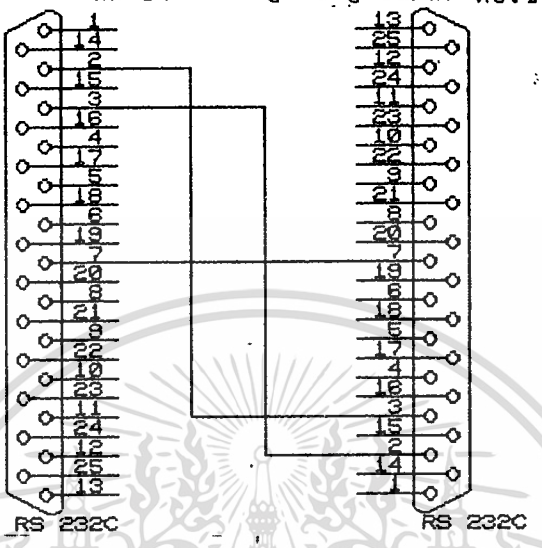
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Pin	Signal name	Data		Control	
		From DTE to DCE	To DTE from DCE	From DTE to DCE	To DTE from DCE
1	Protective ground				
2	Transmitted data	×			
3	Received data		×		
4	Request to send (RTS)			×	
5	Clear to send (CTS)				×
6	Data set ready (DSR)				×
7	Signal ground				
8	Data carrier detect (DCD)				×
9/10	Reserved for data set testing				
11	Unassigned				
12	Secondary data carrier detect				×
13	Secondary clear to send				×
14	Secondary transmitted data	×			
15	Transmit signal element timing				×
16	Secondary received data		×		
17	Receive signal element timing				×
18	Unassigned				
19	Secondary request to send			×	
20	Data terminal ready (DTR)			×	
21	Signal-quality detector (indicates probability of error)				×
22	Ring indicator				×
23	Data signal rate select (allows selection of two different baud rates)				×
24	Transmit signal element timing			×	
25	Unassigned				

ตารางที่ 2.1 แสดงสัญญาณทั้ง 25 ของมาตรฐาน RS 232C

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

วิธีที่ง่ายที่สุดในการเชื่อมต่อโดยใช้จำนวนน้อยที่สุด เพื่ออินเตอร์เฟสระหว่าง ไมโครคอมพิวเตอร์ผ่านทางพอรท์อนุกรม(DCE) ไปยังวีดีโอเทอร์ที่เป็นพอรท์อนุกรม(DTE) คือจะอาศัยขาสัญญาณจาก RS232C เพียง 3 สัญญาณคือ ขา 2 (transmitted data) ขา 3 (receiver data) และขา 7 (signal ground) ต่อเข้าโดยตรง



รูปที่ 2.2

โปรโตคอลสำหรับสำหรับการแลกเปลี่ยนสัญญาณข้อมูล RS 232C

สัญญาณควบคุม 5 สัญญาณ ที่ใช้ร่วมกับ MODEM เพื่อใช้เป็นโปรโตคอลแบบโต้ตอบสัญญาณตามมาตรฐาน RS 232c มีดังต่อไปนี้คือ

1.Data charrier detect (DCD) สัญญาณนี้เป็นสายจาก DCE เพื่อแสดงให้โมเด็มทราบว่าในขณะนั้นตรวจเจอว่าสัญญาณพาหะเกิดการเปลี่ยนแปลงขึ้น(valid carrier)

2.Data terminal ready (DTR) สัญญาณนี้เป็นสายจาก DTE เพื่อแสดงให้โมเด็มทราบว่าในขณะนั้น อยู่ในสภาพที่พร้อม

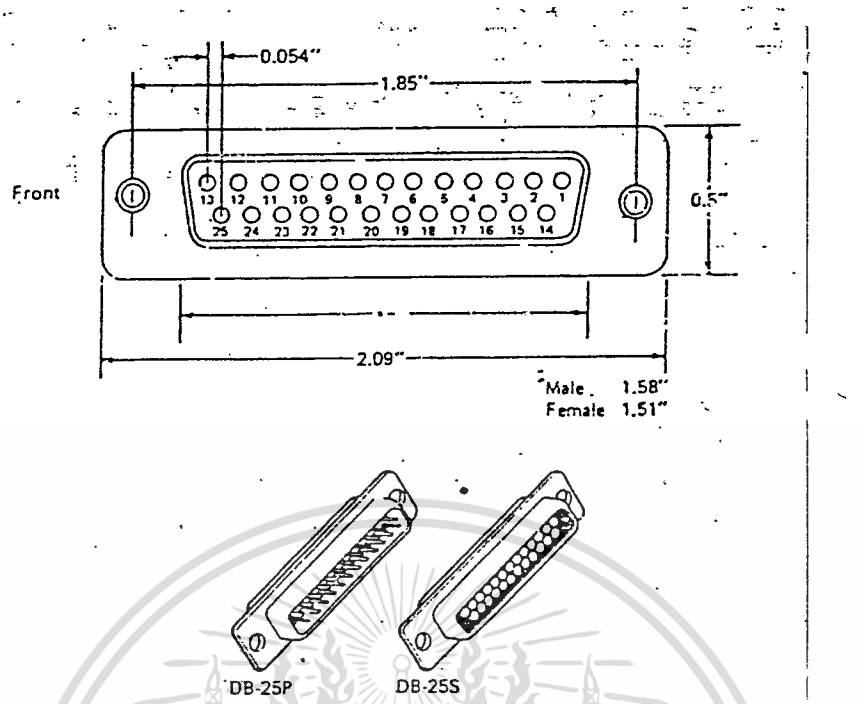
3.Data set ready (DSR) สัญญาณนี้เป็นสายเพื่อตอบสนองสัญญาณ DTR และเป็นการแสดงตัวว่า DCE ถูก ON โดยต่ออยู่กับแชนแนลที่ทำการสื่อสารอยู่

4.Request to send (RTS) สัญญาณนี้เป็นสัญญาณเอาว์พุทที่เกิดขึ้นจาก DCE เพื่อตอบสนองสัญญาณ DTE และเป็นการแสดงว่าขณะนี้ DTE มีความพร้อมที่จะส่งข้อมูล

5.Clear to send (CTS) สัญญาณนี้เป็นสายเพื่อตอบสนองสัญญาณ RTS และเป็นการแสดงตัวว่ามีความพร้อมเกิดขึ้นแล้ว

026993

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.3 แสดงรูปร่างของคอนเนคเตอร์ DB-25P และ DB-25S

อัตราบิท(BIT RATE)กับ อัตราบอด(BAUD RATE)

อัตราบอด คือ อัตราการเปลี่ยนแปลงของสัญญาณใน 1 วินาที สัญญาณที่เปลี่ยนแปลงไป หนึ่งครั้งอาจจะให้ข้อมูลมากกว่า 1 บิท ในกรณีเช่นนี้อัตราบิทจะต่างไปจากอัตราบอดขึ้นอยู่กับจำนวนสภาวะของสัญญาณ ถ้าสัญญาณมี 4 สถานภาพ การเปลี่ยนแปลงสัญญาณ 1 ครั้งจะทำให้ได้ข้อมูล 2 บิท ซึ่งในกรณีนี้เราจะได้อัตราบิทเป็นสองเท่าของอัตราบอด

รหัสที่ใช้ในการส่งสัญญาณ(TRANSMISSION CODE)

ภาษาอังกฤษเป็นภาษาที่ใช้กันแพร่หลายดังนั้นถ้าหากว่า ใช้ตัวอักขระภาษาอังกฤษในการสื่อสารก็ย่อมที่จะใช้ได้ทั่วโลกเช่นเดียวกัน มีรหัสมาตราฐานที่ใช้ในการสื่อสารข้อมูลอยู่หลายรูปแบบแต่ที่เราพบเห็นอยู่บ่อยคือ รหัสแอสกี้(ASCII Code)

รหัสแอสกี้(ASCII Code)

ASCII ย่อมาจากคำเต็มว่า American Standard Code for Information Interchange ซึ่งก็คือรหัสมาตราฐานอเมริกาที่ใช้ที่ใช้ในการส่งข่าวสาร ซึ่งเป็นรหัส

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขนาด 8 บิท โดยใช้ 7 บิทเข้าแทนรหัสตัวอักษรและอีก 1 บิทเป็นบิทตรวจสอบ(Parity bit check) และถือเป็นรหัสแบบหนึ่งที่ใช้ในการสื่อสารข้อมูลกันอย่างกว้างขวางและได้รับมาตราฐานของ CCITT โดยถือว่าเป็นมาตราฐานของ CCITT หมายเลข 5 ในบางครั้งก็เรียกว่า International Alphabet No.5 และในองค์กรสื่อสารระหว่างประเทศบางแห่งก็เรียกว่า "International Standards Organization"หรือ ISO และกลุ่มตัวอักษรที่สร้างขึ้นด้วยรหัสแอสกี้นี้มีชื่อเต็มเรียกว่า "ISO Seven-bit Code Character Set for Information Processing Interchange" นอกจากนี้ ยังมีส่วนที่ให้แต่ละประเทศเลือกกำหนดใช้รหัสบางส่วนได้ตามความเหมาะสมของประเทศนั้นๆ เนื่องจากระหัสแอสกี้เป็นรหัส 7 บิทแทนหนึ่งตัวอักษรจึงสามารถสร้างตัวอักษรได้ถึง 128 ตัวอักษร (เท่ากับ 2 ยกกำลัง 7) จึงช่วยให้เข้ารหัสได้ทั้งตัวอักษรตัวใหญ่และตัวอักษรตัวเล็กได้เพียงพอและยังใส่ลักษณะของกราฟฟิค และตัวอักขระพิเศษที่ใช้ในการควบคุมเข้าไปด้วย

อักขระพิเศษที่ใช้ในการควบคุม

-รหัสที่ใช้ควบคุมการสื่อสารข้อมูล(communication control)

-รหัสจัดรูปฟอร์แมท(format effectors)

-รหัสไว้ใช้แยกข้อความ(information separator)

-รหัสควบคุมพิเศษ(special control characters)

รหัสบีซีดี(BCD Code)

เป็นรหัสแทนตัวอักษรที่ใช้การเข้ารหัสแบบ 6 บิทแทน 1 ตัวอักษร และถือว่าเป็นรหัสมาตราฐานในการเข้ารหัสของเครื่องคอมพิวเตอร์ในยุคที่ 2(1959-1965) จากการที่ใช้ 6 บิทแทนตัวอักษรจึงสามารถเข้ารหัสตัวอักษรได้เท่ากับ 2 ยกกำลัง 6 =64 ตัวอักษรเท่านั้น และเมื่อมีความจำเป็นต้องส่งผ่านข้อมูลภายใต้ระบบการสื่อสาร มักจะเพิ่มบิทตรวจสอบเข้าไปอีก 1 บิทรวมเป็น 7 บิท

รหัสเอ็บซีดิค(EBCDIC)

EBCDIC เป็นคำย่อมาจากคำว่า Extended Binary Coded Decimal Interchange Code เป็นรหัสแทนตัวอักษรที่ใช้การเข้ารหัส 8 บิทแทน 1 ตัวอักษร เป็นรหัสมาตราฐานในการเข้ารหัสตัวอักษรของเครื่องคอมพิวเตอร์ที่ใช้ในยุคปัจจุบัน (ใช้ใน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมนเฟรมคอมพิวเตอร์และมินิคอมพิวเตอร์ แต่ไม่ใช้ในเครื่องไมโครคอมพิวเตอร์) แต่ถึงจะเป็นรหัส 8 บิท ซึ่งดูคล้ายกับรหัสแอสกี้ที่เพิ่มบิทตรวจสอบก็ตาม แต่จะมีการเข้ารหัสที่แตกต่างกันไป รหัสเอ็บซีดิคนี้ความจริงแล้วก็คือ การนำเอารหัสบีซีดีมาขยาย เพื่อให้เข้ารหัสแทนตัวอักษรได้มากขึ้นนั่นเอง จากการที่ใช้ 8 บิทแทนตัวอักษรจึงสามารถเข้ารหัสได้ถึง 2 ยกกำลัง 8 หรือเท่ากับ 256 รูปแบบที่ไม่ซ้ำกันเลย

บิทตรวจสอบ(The parity bit)

บิทตรวจสอบทำหน้าที่ตรวจสอบความถูกต้องของข้อมูล ที่ส่งผ่านมาตามสายส่งสัญญาณ ซึ่งมีอยู่ 2 รูปแบบด้วยกันคือ การตรวจจำนวนคี่(Odd parity) และการตรวจสอบจำนวนคู่ (Even parity)

การตรวจสอบจำนวนคี่หมายถึง การส่งข้อมูลไป 1 ตัวอักษรที่ถูกเข้ารหัสเป็นรหัสแอสกี้ 7 บิทเพิ่มด้วยบิทที่ 8 ซึ่งเป็นบิทตรวจสอบจะต้องนับบิทที่มีค่าของ "1" อยู่เป็นจำนวนคี่ สำหรับการตรวจสอบจำนวนคู่จะมีหลักการเดียวกัน เพียงแต่นับว่าจำนวนบิทที่มีค่าเป็น"1" รวมทั้งสิ้น 8 บิท จะต้องเป็นจำนวนคู่

อักขระควบคุมการสื่อสาร

NUL เป็นอักขระว่างคือ ไม่มีอะไรเลย อย่าสับสนกับคำว่า Blank เพราะว่า Blank ยังมีอักขระอยู่คือ " " ในภาษาเบสิก ส่วน NUL ="" ในภาษาเบสิค อักขระนี้มีประโยชน์สำหรับอุปกรณ์เครื่องพิมพ์ซึ่งต้องการเวลาที่แน่นอนในการเลื่อนหัวพิมพ์

STX(start of text) ใช้ใน Bisyn เป็นการเตือนให้ฝ่ายรับได้รู้ว่าข้างหลังที่ตามมาเป็นข้อความที่ต้องการจะส่งให้ และเป็นการบอกว่าส่วนหัวได้สิ้นสุดลงแล้ว

SOH (Start of heading) ใช้ในการควบคุมสายข้อมูลของ Bisyn ซึ่งหมายถึง จุดเริ่มต้นของส่วนนำหน้าของข้อมูล สถานีในข่ายการสื่อสารจะตรวจสอบอักขระที่ตามหลัง SOH สำหรับตีความว่า จะเป็นผู้รับข้อความที่ตามมาข้างหลังนี้หรือไม่ SOH บางครั้งก็ใช้ในแบบอะซิงโครนัสสำหรับการถ่ายโอนข้อมูลหลาย ๆแฟ้มติดต่อกันโดยไม่แยกแฟ้มส่ง SOH ใช้บอกจุดเริ่มต้นของแต่ละแฟ้ม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ETX (End of text) ใช้ใน Bisynเหมือนกันเป็นการบอกฝ่ายรับว่าข้อความที่ต้องการส่งในกลุ่มนี้หมดแล้ว

EOT (End of transmission) เป็นอักขระควบคุมการสื่อสารเพื่อเป็นการให้รู้ว่าสิ้นสุดการส่งข้อมูลไปยังอุปกรณ์ที่บ่งบอกแล้ว

ETB (End of transmission block) เป็นอักขระควบคุมการสื่อสารเป็นตัวบ่งชี้ถึงการสิ้นสุดกลุ่มข้อมูลที่ส่ง

ENQ (Enquire) อักขระนี้ใช้ในการเรียกการตอบสนองจากฝ่ายรับอาจจะใช้ในการเรียกให้อุปกรณ์ในข่ายการสื่อสารแสดงตัวเอง หรืออาจจะใช้ในการบอกสถานภาพของการสื่อสาร

ACK (Acknowledge) อักขระควบคุมใช้ในการบ่งบอกถึงการสื่อสาร และถ่ายโอนข้อมูลอย่างไม่ผิดพลาด ระหว่างสถานีรับและสถานีส่งหลังจากรับกลุ่มของข้อมูลอย่างถูกต้องแล้วฝ่ายรับจะต้องส่ง ACK ให้ฝ่ายส่งได้รับรู้ว่าข้อความที่ส่งไปได้รับอย่างถูกต้องแล้ว

NAK (Negative Acknowledge) ใช้ระหว่างฝ่ายรับกับฝ่ายส่งหรือบ่งบอกว่ามีการผิดพลาดเกิดขึ้นในข่ายการสื่อสาร

BEL (Bell) เป็นอักขระควบคุมที่ทำงานเหมือนชื่อ ในIBM PC อาจจะกระทำได้โดยการกด CTRL และ G พร้อมกัน

BS (Back spaec) เป็นอักขระสำหรับจัดรูปร่างเพื่อควบคุมเครื่องพิมพ์และการแสดงผลบนหน้าจอ โดยมีผลทำให้เคอร์เซอร์หรือหัวพิมพ์เลื่อนไปทางซ้าย 1 ตำแหน่ง

HT (Horizontal Tabulation) เป็นอักขระสำหรับจัดรูปร่าง ทำให้เครื่องพิมพ์และการแสดงผลบนหน้าจอ เลื่อนไปตำแหน่งที่ตั้งค่าเอาไว้แล้ว

LF (Line Feed) เป็นอักขระควบคุมทำให้ตำแหน่งที่พิมพ์เลื่อนไป 1 บรรทัดนับต่อไปในแถวตั้งอันเดียวกัน

VT (Vertical tabulation) เป็นอักขระควบคุมการจัดรูปร่าง ซึ่งทำให้เครื่องพิมพ์หรือการแสดงผลบนหน้าจอเลื่อนไปยังแถวตั้งเดิมแต่จำนวนบรรทัดห่างออกไป เท่ากับจำนวนบรรทัดที่ตั้งเอาไว้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อ[illegible]ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

FF (Form Feed) เป็นอักขระควบคัมรูปร่าง ใช้ในการเลื่อนหัวพิมพ์ไปยังส่วนบนของแบบฟอร์มที่ตั้งเอาไว้ของหน้าถัดไป

CR (Carriage Return) เป็นอักขระควบคุมการจัดรูปร่าง ซึ่งจะทำให้เครื่องพิมพ์หรือเคอร์เซอร์ของการแสดงผลบนหน้าจอ เลื่อนไปยังตำแหน่งที่ 1 ของบรรทัดเดียวกัน

SO (Shift Out) เป็นอักขระควบคุมพิเศษ ซึ่งทำหน้าที่ในการขยายอักขระมาตราฐานของ ASCIIออกไป

SI (Shift in) เป็นอักขระพิเศษซึ่งอาจจะใช้ในการตั้งค่าใหม่ของเครื่องรับกลับไปตามมาตราฐานของรหัส ASCII

DLE (Data Link Escape) เป็นอักขระควบคุมการสื่อสารใช้ในการแก้ไขดัดแปลงความหมายของอักขระที่ตามหลังมา ใช้ใน Bisyn

DC1 (Device Control 1) เป็นอักขระสำหรับโยกสวิทช์ทางอิเลคทรอนิคส์

DC2 (Device Control 2) เช่นเดียวกับ DC1

DC3 (Device Control 3) เป็นอักขระสำหรับโยกสวิทช์อิเลคทรอนิคส์ อีกอันหนึ่งส่วนมากจะใช้ร่วมกับ DC1 เสมอ

DC4 (Device control 4) ก็เช่นเดียวกับ DC2

SYN (Synchronuos Idle) เป็นอักขระควบคุมการสื่อสารใช้ในโปรโตคอล bisyn สำหรับเริ่มกับการสื่อสาร หรือเมื่อยังไม่มีข้อมูลจะส่งสำหรับให้หน่วยรับปรับสัญญาณนาฬิกาให้เข้ากับฝ่ายส่งเสมอ

CAN (Cancel) เป็นอักขระควบคุมพิเศษ อาจจะนำไปใช้งานต่างๆกันสำหรับในผู้ใช้แต่ละราย

EM (End of medium) เป็นอักขระควบคุมพิเศษใช้ในการบ่งบอกการสิ้นสุดของตัวกลางในทางฟิสิคส์

SUB (Substitute) อักขระนี้ใช้ในการสื่อสารเพื่อควบคุมความแม่นยำเครื่องรับใช้ SUB แทนอักขระที่ถูกตีความว่าผิดพลาด ไม่ถูกต้อง

ESC (Escape) เป็นอักขระควบคุมใช้กันอย่างแพร่หลายในการสื่อสาร

FS (File saparator) เป็นตัวรับข่าวสารใช้สำหรับแสดงถึงขอบเขตในทางลอจิกระหว่างแฟ้มที่ถูกถ่ายโอน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

GS (Group Separator) เป็นอักขระคั่นกลางใช้ในการแสดงขอบเขตในทางลอจิกของกลุ่มข้อมูลที่กำลังส่งออกไป

RS (Record Separator) เป็นอักขระสำหรับคั่นกลางอีกเหมือนกันใช้ในการคั่นระหว่างเร็คคอร์ดที่กำลังส่งออก

US (Unit Separator) ใช้ในการคั่นระหว่างข้อมูลที่แตกต่างกัน

DEL (Delete) ใช้ในการลบอักขระ สำหรับเครื่องพิมพ์ใช้ในการลบอักขระที่ได้รับตัวสุดท้าย ถ้าเป็นหน่วยแสดงผลก็จะเป็นการลบอักขระใต้เคอร์เซอร์ออก

อินเทอเฟส(INTERFACE)

การอินเทอเฟส คือการกำหนดในเรื่องของสายส่งสัญญาณสื่อสารด้วยการถือว่าอุปกรณ์จะถูกแบ่งออกเป็น 2 ชิ้น โดยชิ้นส่วนทั้งสองนี้ต้องทำงานสอดคล้องกันไปพร้อมๆกัน และเป็นไปเพาะตามคุณสมบัติที่กำหนดในการอินเทอเฟสนั้น ๆ สำหรับคุณสมบัติเฉพาะในการอินเทอเฟสมีด้วยกันหลายระดับ แต่การจัดทำอุปกรณ์ทั้งสองชิ้นให้สอดคล้องกันก็เป็นไปตามระดับของการอินเทอเฟสในระดับนั้น ๆ

ระดับต่ำสุดของการอินเทอเฟสก็คือความสอดคล้องในระดับกลไกของอุปกรณ์ที่ใช้ เช่น ปลั๊กที่จะเสียบลงไปในช่องรับสัญญาณ(Socket) ของการสื่อสารจะต้องสอดคล้องกัน ซึ่งจะเป็นการเชื่อมต่อระหว่างอุปกรณ์แต่ละตัวเป็นไปอย่างสมบูรณ์ ในระดับสูงไปอีกขั้นหนึ่งก็คือความสอดคล้องกันทางไฟฟ้า โดยการเชื่อมต่อกันระหว่างอุปกรณ์แต่ละคู่จะต้องจำนวนสายที่เท่ากันและต่อกันอย่างถูกต้องตามคู่สายซึ่งกันและกัน เพื่อจะใช้ส่งสัญญาณได้ถูกต้องตามระยะเวลาที่กำหนดด้วย ดังนั้นจึงอาจจะมีพิธีการ(protocal) ได้มากกว่าหนึ่งระดับที่จะกำหนดลงไปอย่างแน่นอนว่าการส่งผ่านข้อมูลระหว่างอุปกรณ์ ซึ่งจะต้องมีการอินเทอเฟสในลักษณะใดบ้าง ในการอินเทอเฟสระหว่างโมเด็มและเทอร์มินอลจะใช้มาตราฐานของ CCITT คือ V24 และมาตราฐานของ EIA(Eleectronic Industrials association)คือ RS232 โดยที่มีรุ่นล่าสุดคือ RS232Cซึ่งมีการใช้งานกันอย่างแพร่หลาย

เนื่องจากในความเป็นจริงแล้วมรเทอร์มินอลนับร้อยชนิดที่แตกต่างกัน และโมเด็มก็มีอยู่ด้วยกันหลายประเภท ดังนั้นจึงจำเป็นต้องการกำหนดมาตราฐานในการอินเทอร์เฟสระหว่างอุปกรณ์ทั้งสองขึ้น และมาตราฐานที่เป็นที่รู้จักกันดีก็คือ V24 ของ CCITT ซึ่งสามารถรองรับการส่งผ่านข้อมูลได้ด้วยความเร็วถึง 20,000 บิท/วินาทีแต่ถ้าเราต้องการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การส่งผ่านข้อมูลด้วยความเร็วที่สูงกว่านี้สามารถใช้มาตราฐาน V35 ของ CCITT แทนได้ ส่วนในอเมริกานิยมใช้มาตราฐานการอินเทอเฟสของ EIA คือ RS232 ซึ่งทั้ง V24 และ RS232 ได้กำหนดสายที่จะใช้เชื่อมระหว่างโมเด็มและเทอร์มินอล โดยให้สัญญาณไฟฟ้าต่าง ๆ ส่งผ่านไปได้ตามสายที่เชื่อมโยงนี้ รวมทั้งกำหนดระดับของสัญญาณที่ใช้นี้ด้วย

โปรโตคอลและมาตราฐานการเชื่อมโยง

การติดต่อสื่อสารระหว่างสองหน่วยที่จะทำให้ประสบผลสมบูรณ์ได้ก็ต่อเมื่อโปรโตคอลที่ใช้เป็นมาตราฐามเดียวกัน การเชื่อมโยงโครงข่ายจึงจำเป็นต้องอ้างอิงตามมาตราฐานใดมาตราฐานหนึ่ง ในระบบเครือข่าย BBS ของแต่ละแห่งที่สร้างกันขึ้น การเชื่อมต่อผ่านเข้าไปด้วยโมเด็ม เราจำเป็นต้องมีซอร์ฟแวร์อิลูมิเนเทอร์มินอลเพื่อให้การรับส่งข้อมูลเป็นไปตามมาตราฐานที่กำหนด

โปรโตคอลสำหรับการถ่ายโอนแฟ้มข้อมูล

แฟ้มข้อมูลถ้ามองในแง่ของการถ่ายโอนแฟ้มข้อมูลจะมีอยู่ 2 ชนิดคือแฟ้มที่เป็นข้อความ (text file) คือแฟ้มที่มีเฉพาะรหัสที่เป็นอักขระเท่านั้น ไม่รวมถึงรหัสควบคุมที่นอกเหนือไปจากรหัสควบคุมการแสดงข้อความ เช่น LF, CR, FF, HT, VT ตัวอย่างเช่น แฟ้มที่สร้างขึ้นมาโดย Word Star หรือ Note Pad ของ Sidekick แฟ้มอีกประเภทหนึ่งเป็นแฟ้มที่มีข้อมูลที่อาจจะมีค่าได้ตั้งแต่ 0-255 เราเรียกแฟ้มประเภทนี้ว่าแฟ้มเลขฐานสอง (Binary file) เช่น แฟ้มคำสั่งที่ลงท้ายด้วย COM หรือ EXE

การรับส่งแฟ้มประเภทข้อความนั้นไม่มีปัญหาเรื่องที่จะทำให้ตัวโมเด็มเอง หรือเทอร์มินอลตีความหมายผิดไป สำหรับแฟ้มประเภท Binary แล้วรหัสข้อมูลภายในแฟ้มมีโอกาสเป็นไปได้ทุกชนิด หากส่งในลักษณะธรรมดาเหมือนกับข้อความอาจจะทำให้เทอรีมินอลตีความหมายผิดๆไปได้ ดังนั้นในการรับส่งแฟ้มข้อมูลไม่ว่าจะเป็นแฟ้มข้อความหรือแฟ้มเลขฐานสอง ทางที่ดีควรจะมีระเบียบกติกาสำหรับการรับส่งที่เป็นที่เข้าใจทั้งฝ่ายส่งและฝ่ายรับ ระเบียบหรือกติกาที่ว่านี้ก็คือ โปรโตคอลในการรับส่งนั่นเอง

กติกาในการติดต่อข้อมูล

- ตรวจสอบข้อผิดพลาดการรับส่ง
- มีขบวนการที่แน่นอน ที่จะต้องปฏิบัติเมื่อมีข้อผิดพลาดในการรับส่ง
- มีวิธีการเรียงลำดับของกลุ่มข้อมูลที่จะส่ง
- มีวิธีการที่จะทำให้ข้อมูลประเภทแฟ้มเลขฐานสองไปถึงฝ่ายรับได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรโตคอลในการรับส่งข้อมูลส่วนมากจะอยู่ในการสื่อสารอนุกรมแบบ Sunchronous เช่น Bisynch,Hdlc,Sdlc แต่เมื่อการรับส่งเป็นแบบ Asynchronous มีปัญหาก็จำเป็นต้องมีโปรโตคอลในการรับส่งข้อมูลได้ โปรโตคอลสำหรับการรับส่งข้อมูลผ่านทางโมเด็มที่รู้จักกันดี คือ XMODEM, KERMIT, YMODEM เป็นต้น XMODEM เป็นโปรโตคอลตัวแรกที่ใช้ใน Bulletin board ส่วน KERMIT นั้นพัฒนาขึ้นมาทีหลัง นัยว่าเป็มการเพิ่มประสิทธิภาพในการรับส่งผ่านคู่สายที่มีระยะทางไกลๆ เช่นว่าต้องผ่านข่ายการสื่อสารดาวเทียมได้ดีกว่า XMODEM

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3

ทฤษฎี และ หลักการ

ระบบ Spread spectrum คืออะไร

ระบบ spread specturm เป็นการส่งสัญญาณโดยการกระจายย่านความถี่ ออกให้กว้างขึ้นโดยที่ข้อมูลที่จะส่งนั้น จริง ๆ แล้วมี bandwidth ไม่มากนัก เช่นการส่งสัญญาณเสียง แบบ AM จะมี bandwidth เพียงสองเท่าของข้อมูล แต่ในระบบ Spread spectrum นั้น baseband signal (หรือ voice channal) มี bandwidth ไม่กี่ KH_z แต่ส่งออกเป็น MH_z โดยการ Modulated ข้อมูที่จะส่งด้วยสัญญาณ wideband encoding

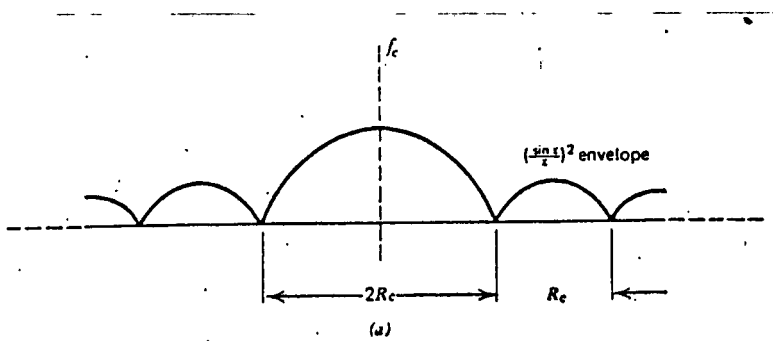
การ modulation ทั่ว ๆ ไปในระบบ spread spectrum มีดังนี้

1.direct sequence เป็นการ mod สัญญาณ carrier ด้วย digital codesequnce ที่มี bit rate มากกว่า bandwidth ของข้อมูลมาก ๆ

2.frequency hopper สัญญาณ carrier ถูก shift ตามรูปแบบที่กำหนดโดย Code sequence ควบคุม การ shift

3.pulse - FM หรือ chriep สัญญาณ carrier จะกวาดไป ในย่าน wide band ภายในช่วงของ pulse interval ที่ให้

ยังมีวิธีการที่คล้ายกับ frequency hopper คือระบบ time hopping และ time - freguency hopping มีข้อแตกต่าง คือเวลาของการส่ง (มักเป็น low duty cycle และ short duration) ซึ่งควบคุมด้วย code sequence ในระบบ time - frequency hopper จะเปลี่ยนแปลงทั้งความถี่ และเวลาที่ใช้ในการส่งตาม code sequence รูปที่ 3.1 แสดง output spectrum ของระบบ direct sequence และ frequency hopping ซึ่งมีย่าน 10 - 100 MH_z



รูป 3.1 direct sequence signal spectrum

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระบบ spread spectrum มีคุณสมบัติที่สำคัญ 2 อย่าง

1.Transmitted bandwidth ต้องมากกว่า bandwidth ของข้อมูลมาก ๆ

2.ข้อมูลที่จะส่งจะถูกนำมา mod ทำให้เกิด RF bandwidth

ทำไมต้องใช้ Spread spectrum

ระบบ Spread spectrum มีคุณสมบัติที่ได้เปรียบมากมายคือ

1.เลือก addressing capability ได้

2.เป็น code division multiplexing ทำให้สามารถทำงาน multiple access ได้

3. low density power spectra เพื่อหลบซ่อนสัญญาณ

4. Message ได้รับการป้องกันจากการลอบฟัง

5. มี resolution ranging สูง

6. ป้องกันการรบกวนจาก interference

Maximal Sequences [56]

เป็นความยาวของ code ซึ่งสามารถสร้างได้จาก shift register หรืออุปกรณ์อื่น binany shift register sequence generator ที่จะกล่าวถึงมี maximum length sequence เป็น 2^n-1bit โดย n เป็นจำนวน stage ของ shift regiter ซึ่งประกอบด้วย shift regiter นำมาต่อร่วมใน logic ที่เหมาะสม มา feedback โดยนำ logic จาก stage ที่ 2 หรือมากกว่า มาเข้า input output ของ sequence generator และข้อมูลของ n stage ที่แต่ละเวลาเป็น function ของ output ของ stage feedback ที่แต่ละเวลา การต่อป้อนกลับจะสร้างความยาวต่าง ๆ จาก 2^{35}-1 bit แสดงในตาราง 3.1

คุณสมบัติของ maximal code requence

1. จำนวนของภาวะ "1" จะเท่ากับภาวะ "0"ใน 1 bit ข้อมูล 1023 bit code จะมีbit "1" 512 bit และ bit "0" 511 bit ถ้า "1" แทนด้วย +v และ "0" เป็น "-v" จำนวนของ offset ของความยาว code เป็นส่วนกลับกับความยาวของ code หรือ $v/(2^n-1)$

2. โอกาสในการกระจาย "1" และ "0" จะกำหนดและเกิดขึ้นเหมือนเดิม

3. Autocorrelation ของ code หรือค่าทั้งหมดของ phase shift

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

correlation เป็น -1 ยกเว้นแต่ 0± 1 bit phase shift area

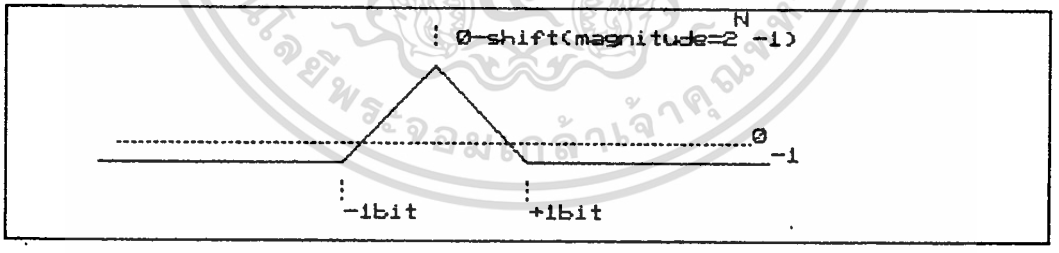
เช่น maximal linear code มี 7 bit (3 stage shift regiter)

$$S = S_1S_2S_3 \quad , \quad S_3 + S_2 \text{ ---}>S_1$$

โดยมี reference sequence :1110010

shift	Sequence	Agreement	Disagreement	A-D
1	0111001	3	4	-1
2	1011100	3	4	-1
3	0101110	3	4	-1
4	0010111	3	4	-1
5	1001011	3	4	-1
6	1100101	3	4	-1
0	1110010	7	0	7

autocorrelation จะแสดงได้จากค่า agreement หรือจำนวน bit ที่คงค่าเดิม ณ ตำแหน่งเดิม ลบกับ disagreement ซึ่งเป็นจำนวน bit ที่เปลี่ยนแปลงไป แต่ละครั้งนำมา plote จะได้ดังรูป 3.2 (magnitude = 2^n-1)



รูป 3.2

นั่นคือค่าของ autocorrelation จะเป็น -1 ทุกค่าการ shift เว้นที่ตำแหน่ง 0 ± 1 ซึ่งจะมีค่าเป็น 2^n-1 ถ้า 1023 bit maximal code $(2^{10}-1)$ จะได้ค่า peak_average ของ correlation = 1024

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

bit rate และความยาว code

Code bit_rate ในระบบ spread spectrum มีผลต่อระบบหลายอย่าง โดยเฉพาะอย่างยิ่งระบบ direct sequence เพราะว่า bandwidth ของการส่งเป็นฟังก์ชั่นโดยตรงกับ code bit_rate (main lobe RF bandwidth เป็น 2 เท่าของ code bit_rate)

code repetition rate เป็นฟังก์ชั่นของ bit rate ด้วยโดย

$$Rc = \frac{\text{clock rate (bps)}}{\text{code length (bit)}}$$

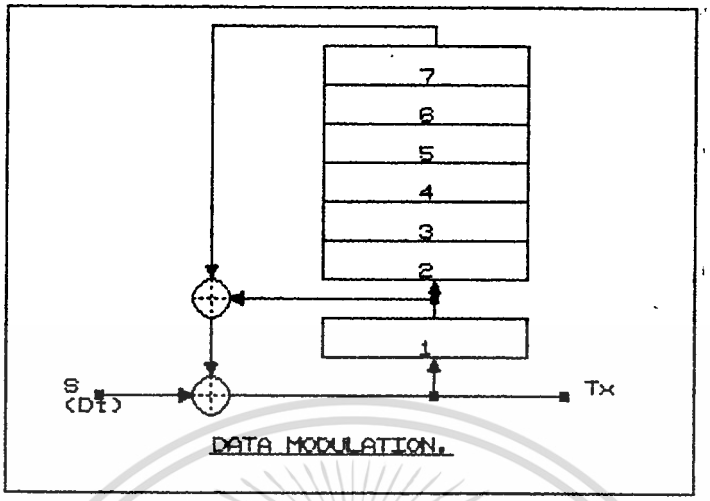
ซึ่งจะทำให้เกิด line spacing ใน RF output spectrum และเป็นส่วนที่ต้องพิจารณาถึงใน การเลือก code repetition rate มีหลักที่ว่า คาบเวลาของ code จะต้องมาก ๆ

การ Demodulation สัญญาณ Spread spectrum

ทำได้โดยการคูณสัญญาน input ด้วย local reference ที่ตรงกัน และ sync กับเครื่องส่งที่ด้านเครื่องรับ สัญญาน spread spectrum จะถูกยุบจากสัญญานมากมายที่กระจาย ในย่านความถี่กว้างมาก มาเป็นสัญญานในระดับ band ซึ่งมี Band width แคบ ๆ ซึ่งทำงานลักษณะนี้เรียกว่า correlation สัญญานนี้จะถูกคูณกันแล้ว นำไปผ่านการอินทิเกรต จะได้สัญญานที่ต้องการ ในprojectนี้ การ Demodulation สามารถพิจารณาได้จากทฤษฎี Pseudo random encoder/decoder ต่อไปนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ทฤษฎี PSEUDO RANDOM ENCODER/DECODER.



รูป 3.3

จากวงจรเขียนเป็นสมการสภาวะได้ดังนี้

$$T = S + DT + D^7T$$

$$= S + (D + D^7)T \quad ..(1)$$

$$S = [\,1 + (D + D^7)\,]T$$

ให้ $F = D + D^7$

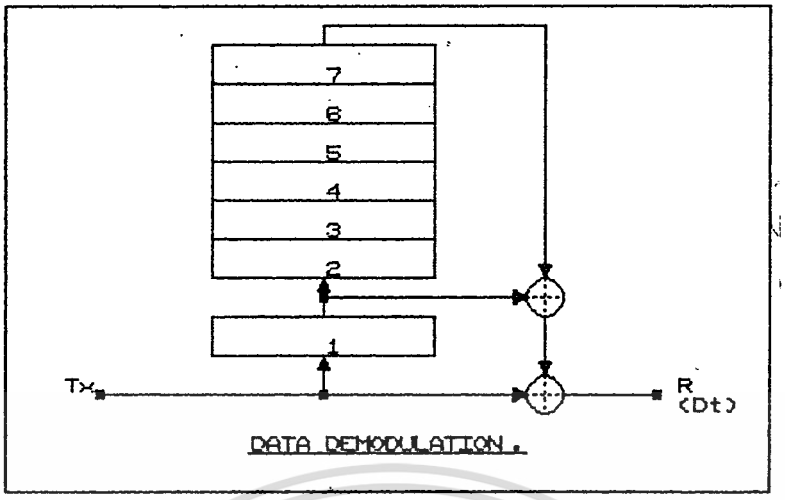
จะได้ $S = T(1 + F)$

เมื่อ F คือจุดที่ถูก Tap สัญญาณออกมา

เขียนอยู่ในรูปสมการทั่วไปได้

$$T = S + FS + F^2S + F^3S + \ldots\ldots$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูป 3.4

ในวงจร decoder เขียนเป็นสมการสภาวะได้คือ

$$R = DT + D^7T + T$$

$$= T + (D + D^7)T \quad ...(2)$$

เขียนอยู่ในรูปทั่วไปได้คือ

$$R = T + FT + F^2T + F^3T + ...$$

แทนค่า (1) ใน (2)

$$R = S + (D + D^7)T + (D + D^7)T$$

$$= S + 0$$

$$R = S$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การ Integrator and Dump

เป็นการสะสมพลังงานทั้งหมดใน 1 bit ของข้อมูลที่ถูกกระจาย ทางด้านส่งซึ่งการรบกวนในระหว่างภายใน 1bit จะถูกตัดทิ้งไปโดยวงจร integrator ซึ่งก็คือวงจรกรองความถี่ต่ำผ่าน นั่นเอง และ dump ข้อมูลคืนเป็นสัญญาณ square wave โดยวงจรเปรียบเทียบแรงดัน (voltage comparator)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

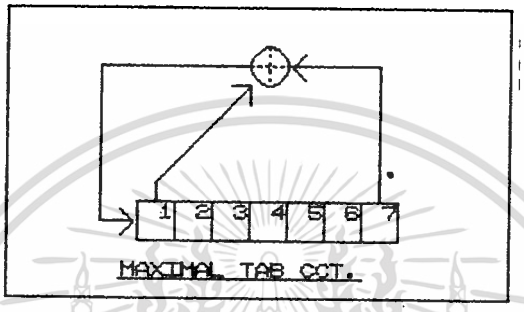
บทที่ 4

แนวทาง และการออกแบบวงจร

การออกแบบ Max Code Sequence

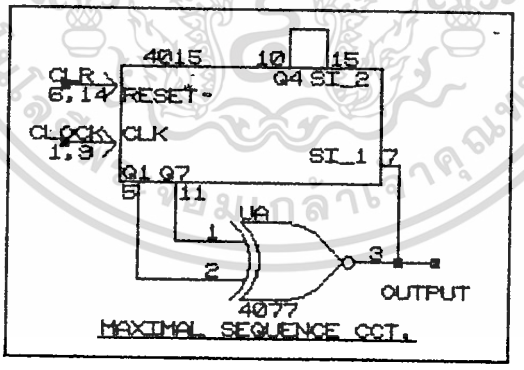
ใช้ n=7 คือ 7 stage shift register ใช้การ feedback จากตาราง 3.1 - feedback Connections for Linear m- Sequences

ที่ 7 stage เลือก Maximal taps ที่ [7,1] มาใช้ ดังรูป4.1



รูป 4.1

ใช้ CMOS 4015 ออกแบบได้ดังนี้ (รูป4.2)

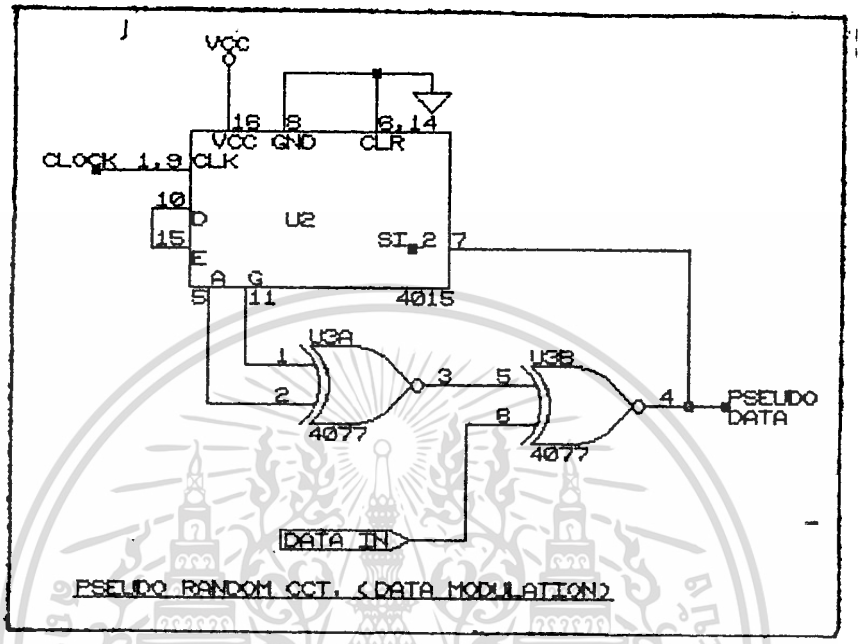


รูป 4.2

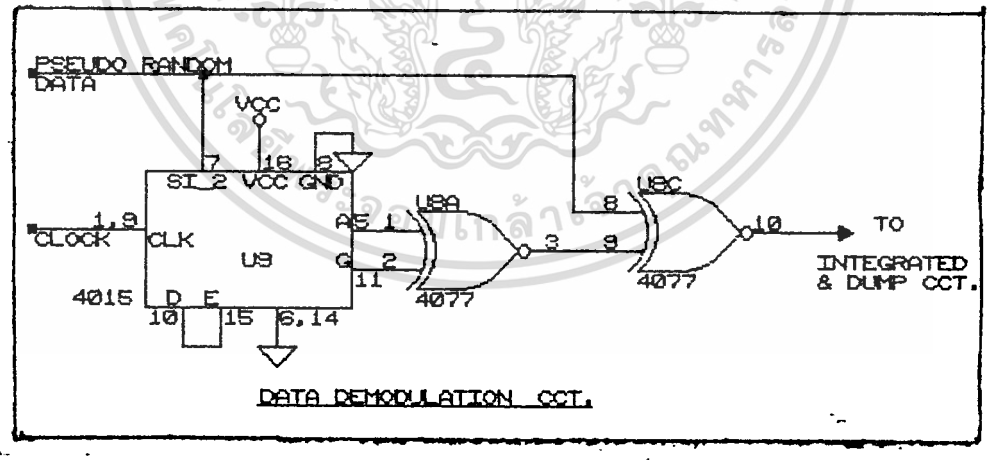
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

DATA Modulation and Demodulation

จากรูป3.3 ออกแบบสร้างโดยใช้ XNOR เกทและ Shift Register ได้ดังรูป4.3 data modulation และ รูป4.4 เป็น data demodulation ที่ออกแบบจากรูป3.4



รูป 4.3



รูป 4.4

การสร้างสัญญาณ NRZI

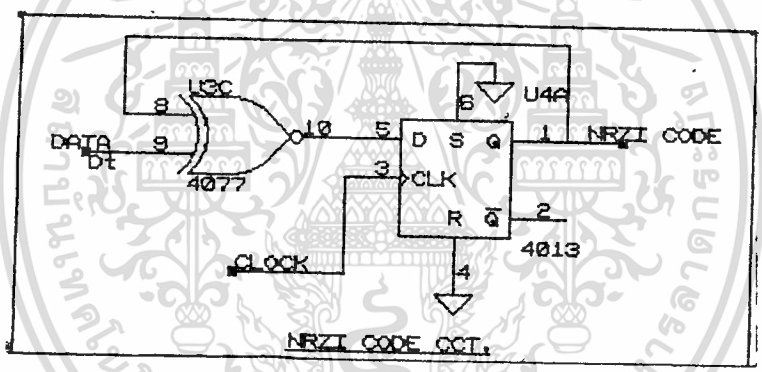
พิจารณาการเกิดสภาวะต่างๆ ของสัญญาณNRZI (Non-Return to Zero Inverted) จากตารางข้างล่างจะเห็นว่า ข้อมูลถัดไปจะมีสภาวะเป็น H ถ้าข้อมูลอินพุทมี สภาวะเหมือนสภาวะของเอาท์พุทเดิม แต่ถ้าข้อมูลอินพุท มีสภาวะต่างจากเอาท์พุทเดิมแล้ว สภาวะถัดไปของเอาท์พุทจะเป็น L ดังนั้นเราสามารถสร้างขึ้น โดยใช้

เอกสารนี้สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เกท XOR และอาศัย D-F/F เพื่อทำให้เป็น Non-Return to Zero ดังรูป4.5

Dt	Qt	Qt+1
0	0	1
0	1	0
1	0	0
1	1	1

ตารางแสดงสภาวะของสัญญาณ NRZI

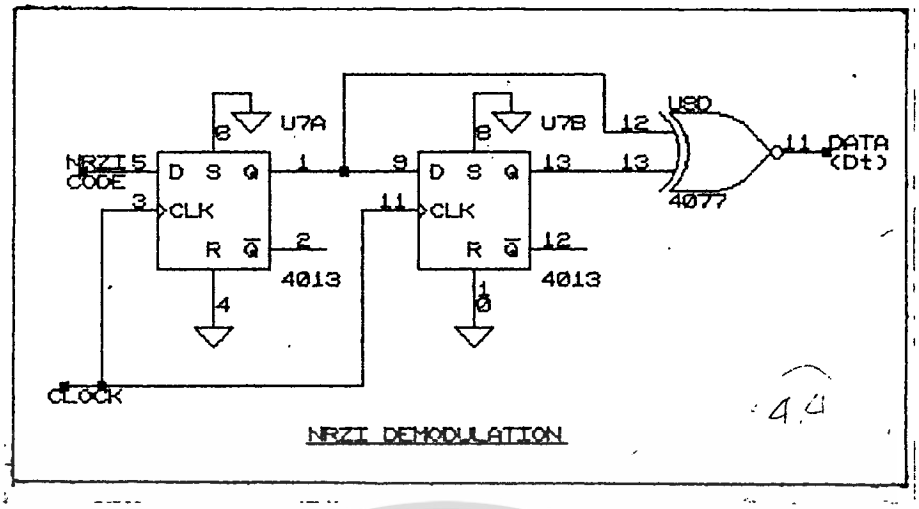


รูป 4.5

การ DECODE สัญญาณ NRZI

การ decode สัญญาณ NRZI สามารถทำได้โดยอาศัย D-F/F และ เกท XNOR การออกแบบ พิจารณาจากการสร้าง NRZI code เกิดจากการ XNOR ระหว่างข้อมูลเดิมกับข้อมูลอินพุท และใช้ D-F/F ทำเป็น NRZ ดังนั้นเราสามารถ decode ข้อมูลกลับได้อย่างง่ายๆ โดยใช้ XNOR สัญญาณ NRZI กับข้อมูลเดิมดังรูป โดยอาศัย D-F/F และสัญญาณ Clock ที่ sync กับทางด้านส่ง ช่วยจัดสัญญาณ NRZI ให้ sync พอดีกันกับสัญญาณ Clock ของทางด้านรับ ดังรูป 4.6

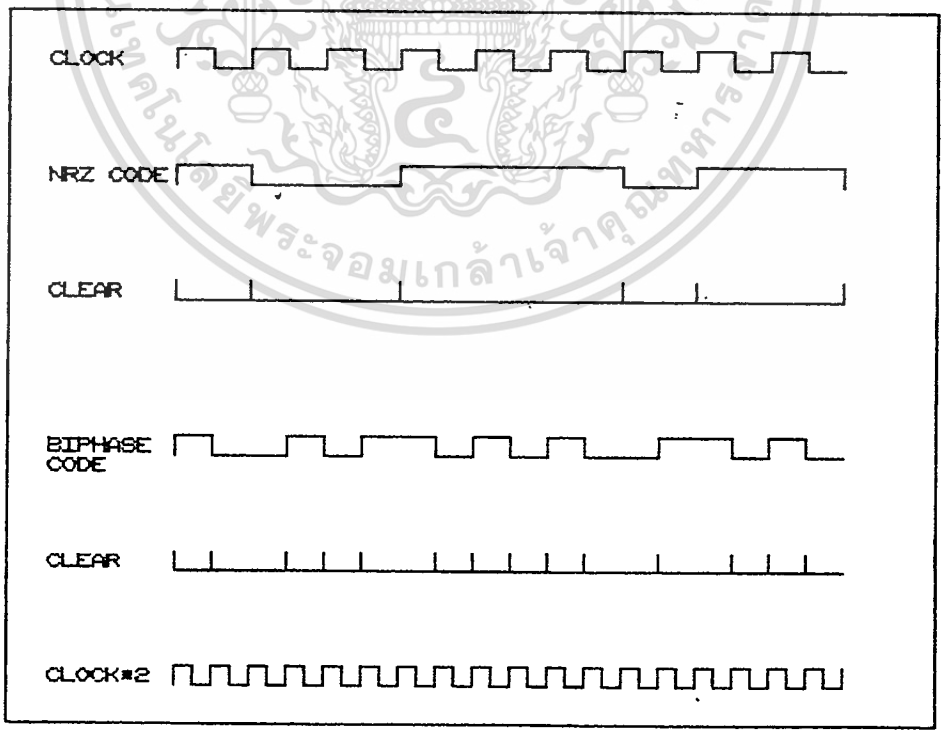
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูป 4.6

การ Recovery สัญญาณ Clock จากข้อมูล Biphase code

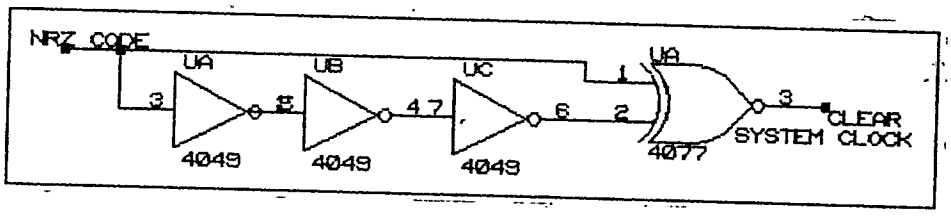
ในกรณีที่สัญญาณรับเข้ามา เป็น NRZ code การ recovery สัญญาณ clock ทำได้ง่ายๆโดยอาศัย การ reset วงจร clock free running ของเครื่องรับทุกๆครั้งที่มีการเปลี่ยนสภาวะ ซึ่งจะตรงกันกับขอบขาขึ้น ของสัญญาณ clock พอดีทุกครั้งดัง timming diagram ข้างล่าง



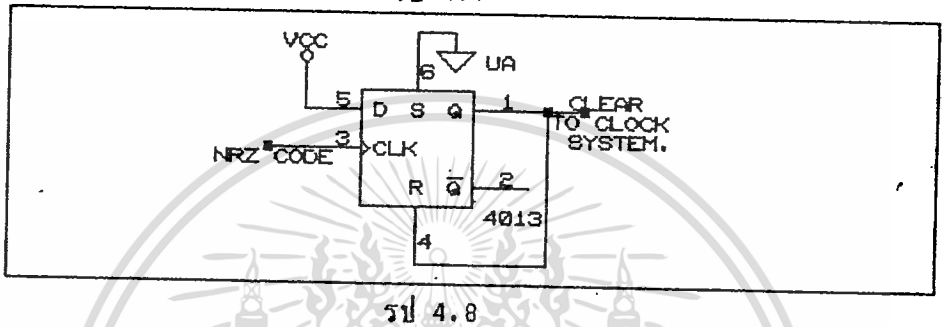
การสร้างสัญญาณ reset ที่ขอบของการเปลี่ยนแปลง ทำได้โดยอาศัยการหน่วง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เวลาข้อมูลช่วงสั้นๆ แล้วนำมา XNOR กับข้อมูลนั้น ดังวงจรข้างล่าง ในกรณีที่ clock มีความถี่สูงมาก ช่วงเวลาในการรีเซตต้องสั้นมากๆ เราใช้ D-F/F สร้างขึ้นได้ดังรูป 4.7 และ 4.8

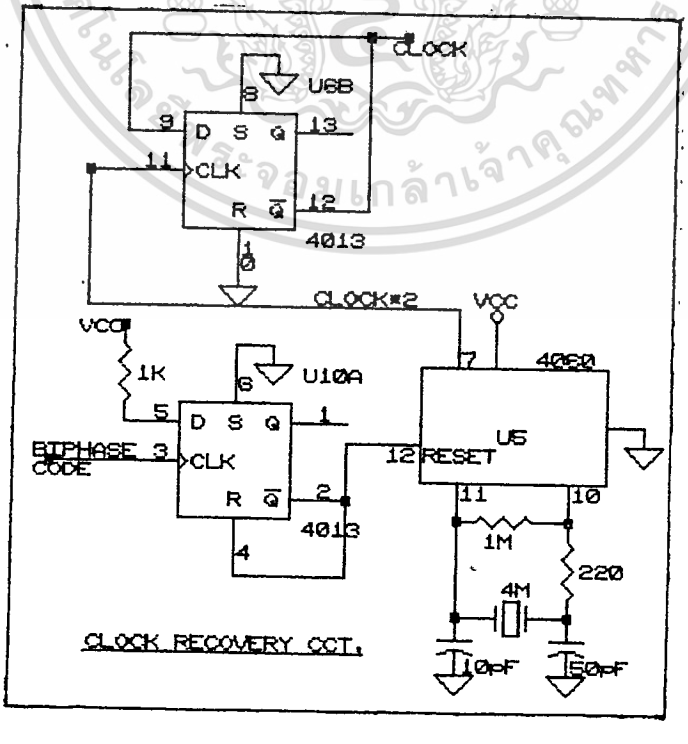


รูป 4.7



รูป 4.8

พิจารณาเช่นเดียวกัน ถ้าอินพุทเป็น Biphase code การสร้างสัญญาณ reset ที่ขอบของการเปลี่ยนแปลง จะตรงกันกับขอบขาขึ้น ของสัญญาณ clock ที่มีความถี่เป็น 2 เท่า ดังนั้นการ recovery สัญญาณ clock จะได้ความถี่เป็น 2 เท่า จึงต้องใช้วงจรหารสองก็จะได้ สัญญาณ clock ที่ sync ตามต้องการดังรูป4.9

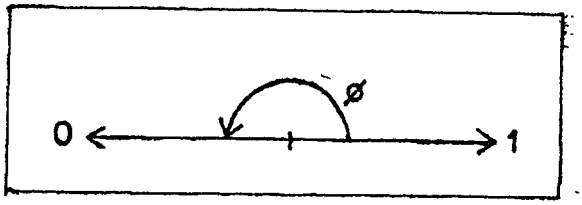


รูป 4.9

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

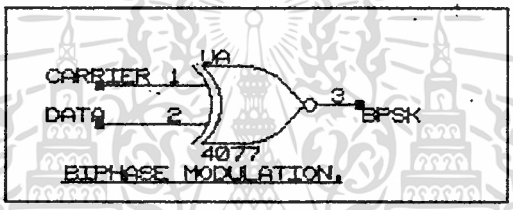
BPSK modulation

หลักการของ bpsk คือ phase ของ carrier จะคงเดิมถ้าสัญญาน input เป็น "1" แต่ถ้าสัญญาน input เป็น "0" phaes จะถูก shift ไป 180° ดังรูป4.10



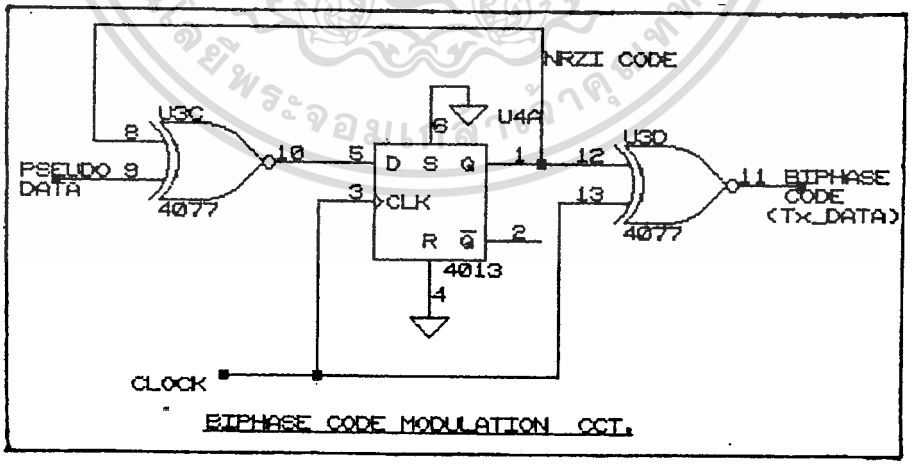
รูป 4.10

ในการปฏิบัติจริงสามารถใช้ logic gate แบบง่าย ๆ ได้ คือ exclusive nor ดังรูป4.11



รูป 4.11

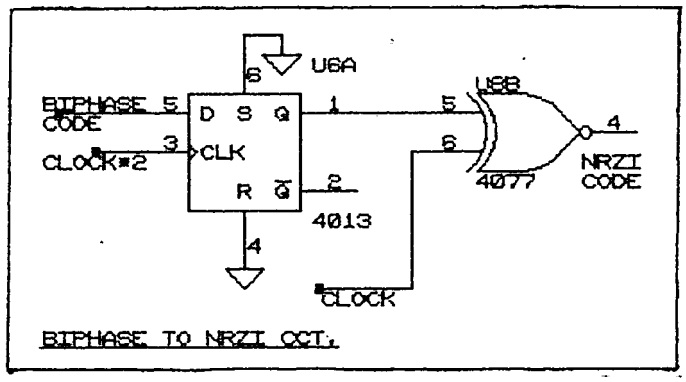
ลักษณะวงจรจริงที่ใช้เป็นการสร้างสัญญาณ BPSK จากสัญญาณที่เป็น NRZI ดังรูป4.12



รูป 4.12

และการคืนรูปเป็นสัญญาณ NRZI ทำได้ดังรูป 4.13

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูป 4.13

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

วงจรและการทำงาน

วงจรภาคส่ง

ประกอบด้วย free-running system clock U1 กำเนิดความถี่ 32kHz และวงจร data encoder เป็น pseudo random หรือ pseudo noise modulation ประกอบด้วย U2,U3a,U3b สัญญาณ modulated จะถูกทำเป็น NRZI code โดย U3c, U4a และท้ายสุด U3d จะทำ NRZI code เป็น Biphase code(BPSK)

วงจรภาค driver เข้า AC line อาศัยวงจรไอซี LM383 และชุด Network ป้องกันสัญญาณ 50 Hz (HPF) ส่งออกสู่สายส่งกำลัง 220V

วงจรภาครับ

U11 รับสัญญาณ inputซึ่งเป็น Biphase code มาจาก Network ส่วนหน้าจัดรูปให้เป็นสัญญาณคลื่นสี่เหลี่ยม

วงจร colck recovery ประกอบด้วย U5,U6a,U6b ซึ่งเป็น free-running ที่จะถูกรีเซตให้ตรงกับสัญญาณอินพุทโดย U10a

สัญญาณ NRZI จาก U8b จะคืนรูปเป็น pseudo code โดย U7a,U7b,U8d วงจร data decoder ประกอบด้วย U8a,U8c,U9 และท้ายสุดเป็น integrator & dump ส่งข้อมูลไป RS232c หรือวงจร control

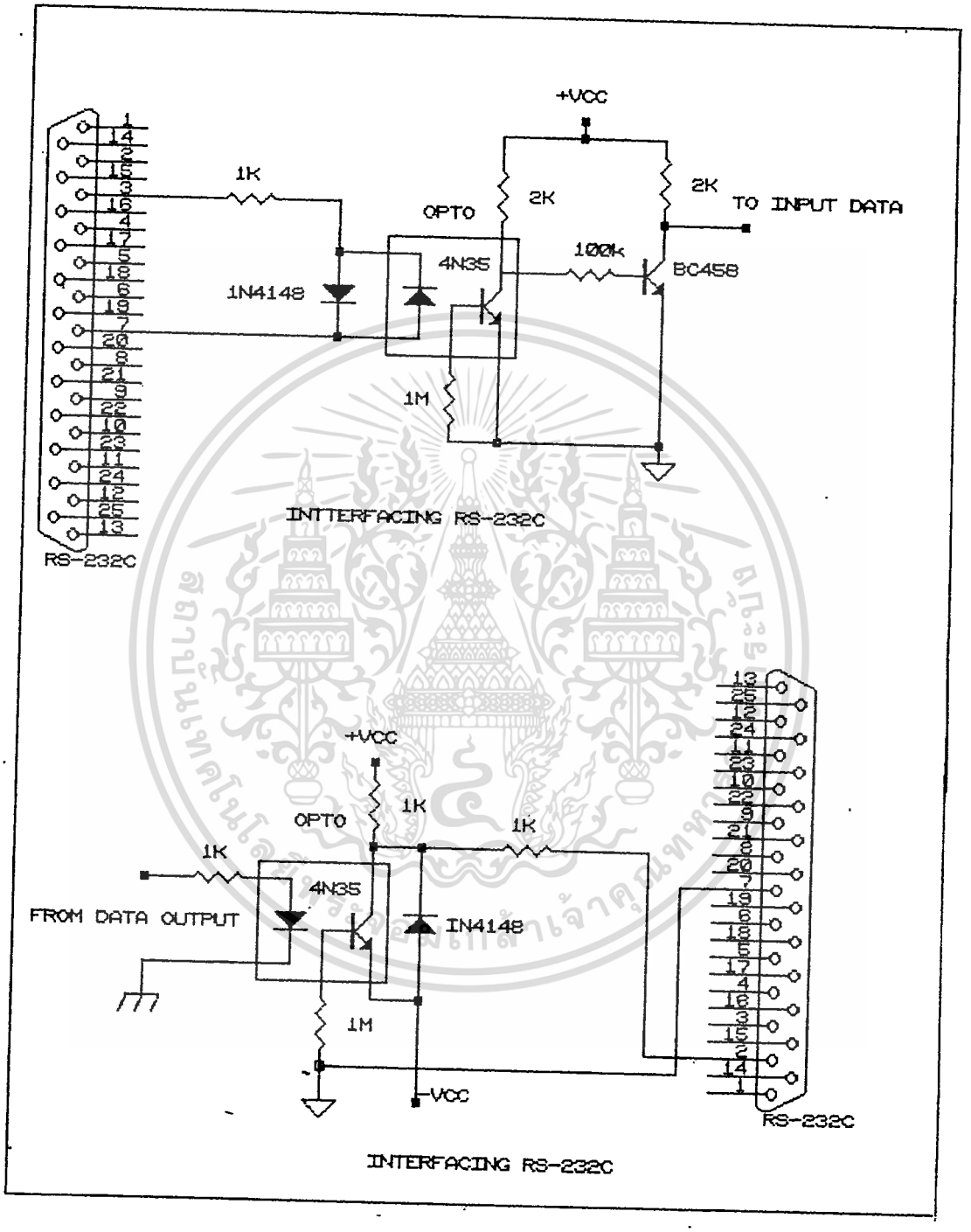
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทำงานของวงจรควบคุมอุปกรณ์ไฟฟ้า

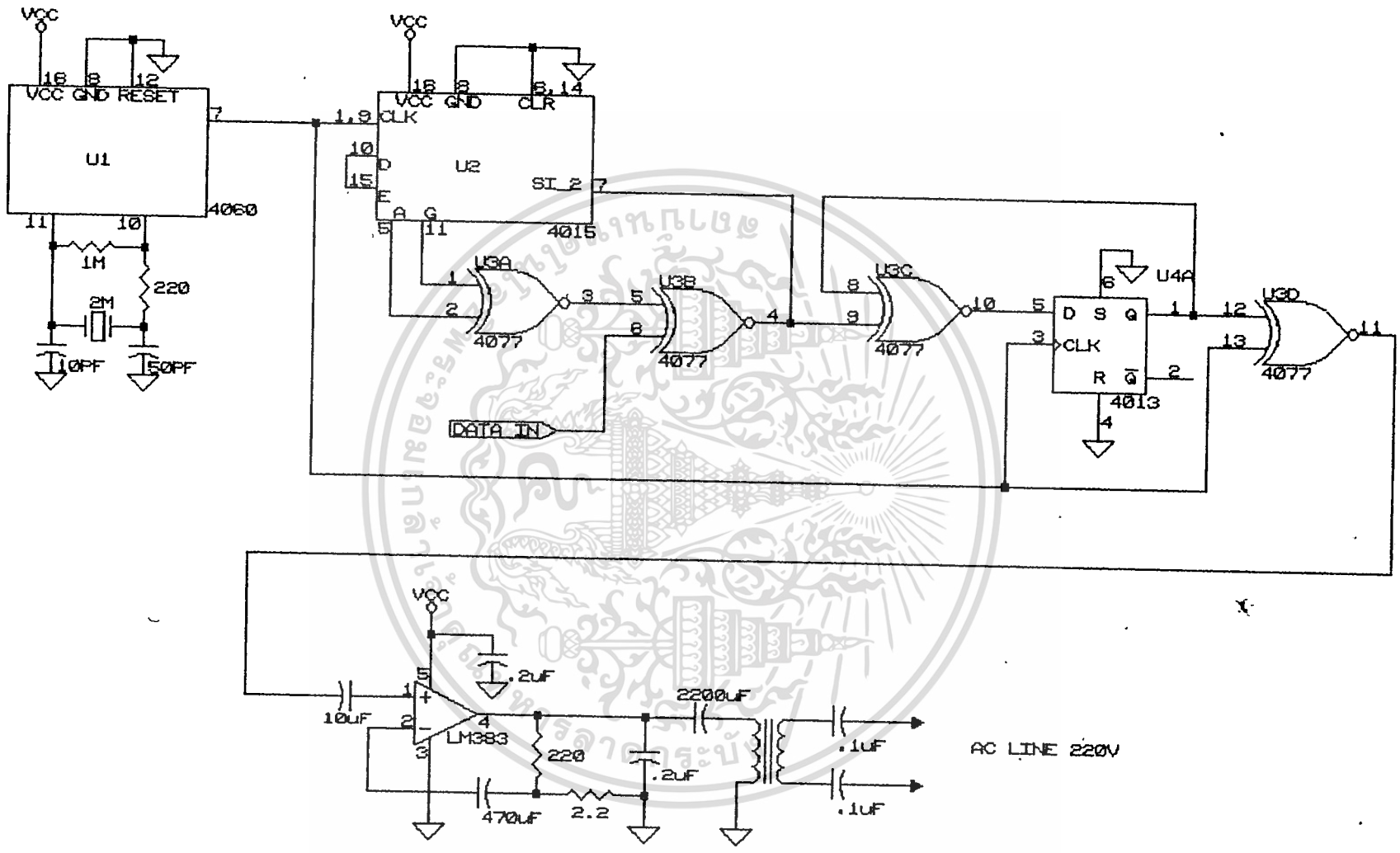
จากวงจรใน sheet 4 U1A และ U1B ทำหน้าที่ผลิตความถี่ 7.42 KHz ป้อนเข้า U9 ซึ่งเป็นวงจรหารความถี่โดยมี U2A ควบคุมการ CLEAR U9 เมื่อขา 1 และ 2 เป็นลอจิก '0' U1D ควบคุมการผลิตความถี่ของ U1A และ U1B ให้หยุดผลิตความถี่เมื่อขา11 และ13เป็นลอจิก '1' U9 ส่งความถี่ 50 Hz มาให้ U7 ซึ่งเป็นสัญญาณนาฬิกา ให้ชิพรีจิสเตอร์(shife register) ซึ่งมีสัญญาเข้าทางขา 7 หน้าที่ของชิพรีจิสเตอร์ คือ เปลี่ยนข้อมูลเข้าที่เป็นข้อมูลอนุกรมเป็นข้อมูลขนาน การทำงานของอุปกรณ์ที่กล่าวมานี้เป็นอุปกรณ์รับข้อมูล หลักการทำงานในวงจรคือ เมื่อ U1Aและ U1B ผลิตความถี่ป้อนให้ U9 จนกระทั้งขา 2 และ 14 (Q13 และ Q10 ตามลำดับ) เป็นลอจิก '1' ขา 1 ของ U1A เป็นลอจิก '0' มีผลทำให้ U1Aและ U1B หยุดผลิตความถี่ ในขณะเดียวกันขา 2 ของ U2A เป็นลอจิก '0' ไม่มีสัญญาณนาฬิกาป้อนให้แก่ U7เมื่อ ข้อมูลเป็น '1' (stop bit) เมื่อมีข้อมูลเข้ามา เอาว์พุทของ U2Aเป็นลอจิก '1' clear ขา 12 และ 13 มีลอจิก '0' เอาว์พุท ลอจิก '1' เอาว์พุทลอจิก '1' U1Aและ U1Bผลิตความถี่มีสัญญาณนาฬิกาป้อน U7 ทำให้ชิพรีจิสเตอร์ ทำงานเปลี่ยนข้อมูลอนุกรมเป็นข้อมูลแบบขนาน

U6,U4C,U2B,U5A,U4B,U2D,T1,T2,T3 เป็นวงจรเลือกพอรท์(decode port) โดยมีรหัส $1F เป็นบิทคอนโทน(bit control) ส่งมาก่อนแล้วตามมาด้วยพอรท์คอนโทน(PORT CONTROL)เพื่อควบคุมอุปกรณ์ โดยมี U8 เป็นตัวแปลงข้อมูล 4 บิท เป็น 16 (4 TO 6 LINE DECODE) และในการควบคุมอุปกรณ์จะใช้บิท 0 ตัวควบคุมการปิดเปิดอุปกรณ์ และใช้บิท 1,2,3,4 เป็นตัวเข้ารหัสเลือกพอรท์ ดังนั้นจะได้อุปกรณที่ควบคุม 16 ตัว (แต่ถ้านำบิท 1,2,3,4,5,6,7 มาเข้ารหัสจะได้พอรท์สูงสุด 127 อุปกรณ์) U10A,U10B,U11A,U11Bเป็นตัวจำข้อมูล(latch)เพื่อไม่ให้เกิดการเปลี่ยนแปลงในพอรท์ที่ไม่ต้องการให้เปลี่ยนในขณะเปลี่ยนสภาวะที่บิท 0 และมี T5ถึง T8 เป็นตัวขับรีเรย์(rerey) เมื่อควบคุมการปิดเปิดอุปกรณ์

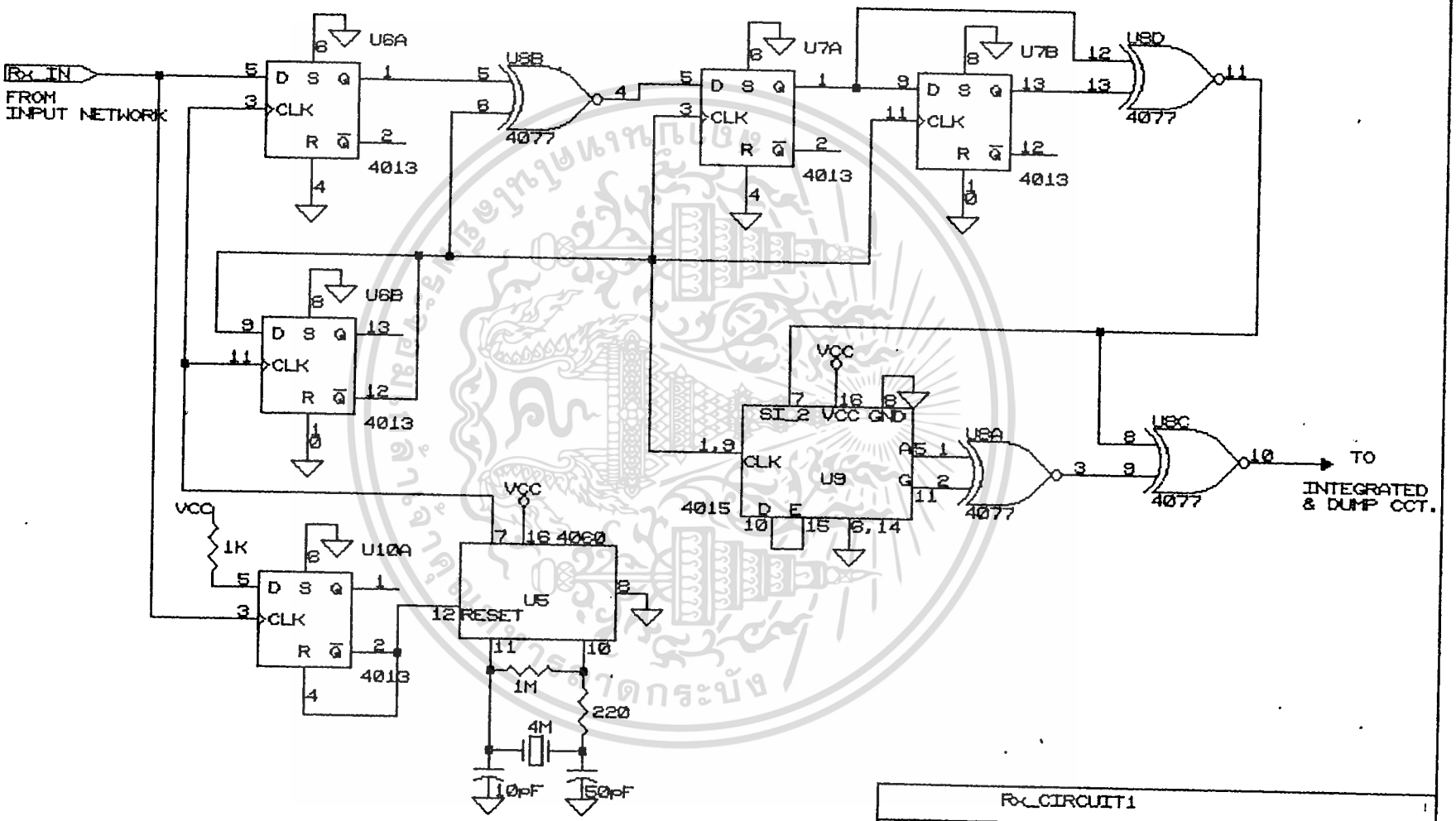
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



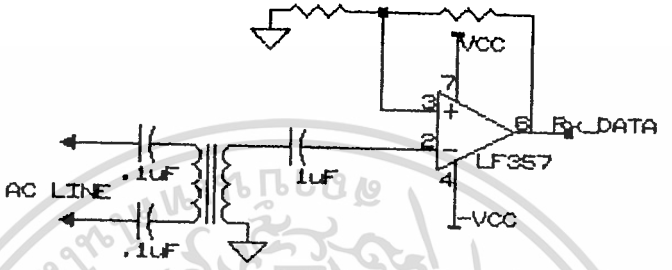
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



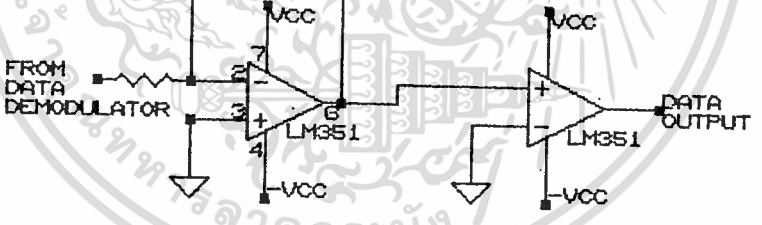
Tx_CIRCUIT1		
Title: AC LINE MODEM AND CONTROL		
Size: A	Document Number:	REV
Date: April 13, 1990	Sheet 1 of	



Rx_CIRCUIT1		
Title: AC LINE MODEM AND CONTROL		
Size: A	Document Number	REV
Date: April 13, 1990	Sheet 2 of	

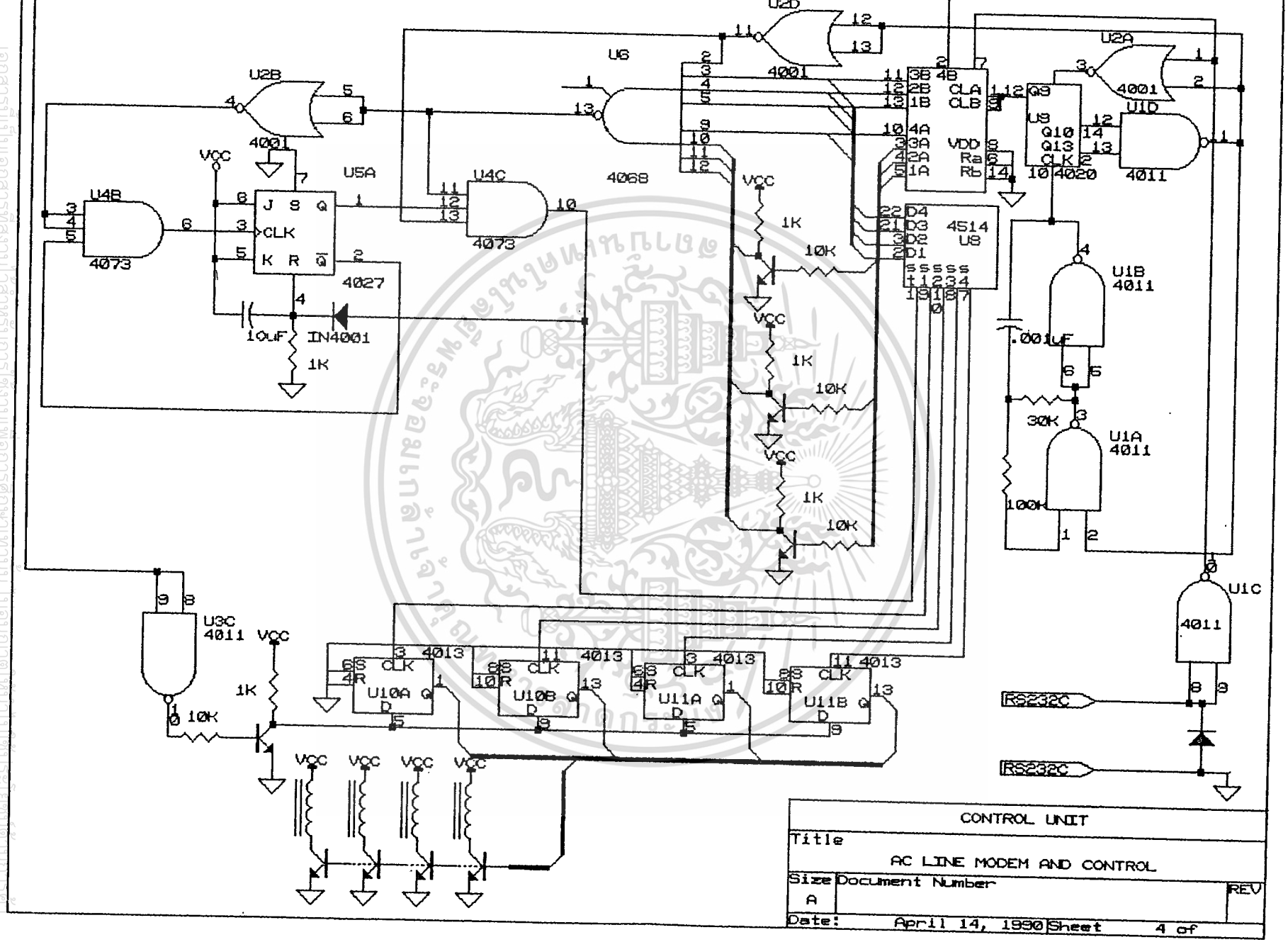


INPUT NETWORK CCT.



INTEGRATOR & DUMP CCT.

Rx_CIRCUIT2		
Title: AC LINE MODEM AND CONTROL		
Size: A	Document Number:	REV:
Date: April 13, 1990	Sheet 3 of	


U2D
4001
U6
4068
U2B
4001
U5A
4027
U4B
4073
U4C
4073
U2A
4001
U1D
4011
U9
4020
4514
U8
U1B
4011
U1A
4011
U1C
4011
U3C
4011
U10A
U10B
U11A
U11B
4013
1K
10K
10uF
IN4001
.001uF
30K
100K
VCC
RS232C
RS232C
CONTROL UNIT
AC LINE MODEM AND CONTROL
Size
A
Document Number
REV
Date:
April 14, 1990
Sheet
4 of

ภาคผนวก ก.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
program ACLINE-MODEM-CONTROL;

uses crt,DOS,graph,com ;

{-------------------------------------------------------}

{ **********************  THAI  ************************}

{-------------------------------------------------------}

const Bit : array[0..7] of byte = (1,2,4,8,16,32,64,128);

      incDec =20;

      NOMAX = 20;

      Xa= 11 ; Ya = 41 ; Xb = 419 ; Yb = 169;

type Font = array[1..44] of byte;

     fontRecord = record

           Fontbuff : font;

        end;

     FileType = file of FontRecord;

           filen = string[30];

var

       drive : string[2];

      exits,endofproject : boolean;

     FontFile : filetype;

     k,Fontcode,mode : byte;

     Font1,font2 : array[0..223] of font;

     fontDisp : font;

     ch : char;

     dirinfo : searchRec;

     xdown,ydown,xUp,yUp,blockNum : array [1..3] of integer;

     dis,dis1,L,row,dirNum,blockMain : integer;

      size3,size4,size5 ,size1,size2,dir : word;

    funckey,first : boolean;

    pic1,pic2,pic3,pic4,pass1,pass2,clearP1,clearP2 : pointer;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        NAME  : filen;
        NOPC : filen;
        NAMEFILE,NAMEDATA : filen;
        dirname,dirsize : array [1..100 ] of string[70];
        comnum,baudbit,databit,paritybit,stopbit : char;
        brat,bitd,sbit,pbit : string[6];
               bang,   b1,d,s,p: integer;
  {-----------------------END  THA1----------------------------}

  {--------------------------}
  {***** ac line control *****}
  {--------------------------}
   var
         num,i : integer;
         xu,yu,xd,yd : word;

         on,off ,open : boolean;
         system1,PIVNUM : array [1..9] of boolean;
         PIC : WORD;

         PIV : ARRAY[1..9] OF POINTER;
 {-------END AC CONTROL--------}

 {-----------------------------}
 {************ MODEM *************}
 {-----------------------------}
const maxheep = 25521 ;
      cr  = #13;
      codefile = '$FILE';
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        controler= $1f ;
        endfile = '$$END';
        nofile = '$$NOT';
        dire ='$$DIR';
        star = 'STAR';
        maxno_ = 127;
 type   str30  = string[30];
        database = array[1..maxheep] of char;
        strt = string[12];

        filevar = file of strt;
        fivart = file of char;
        baseT =^database;
 var

     out : boolean;
     path : string[20];
     shift : string[5];
     filename : filen;
     filepro : fivart;
     base : baset;
     wantname : filen;
      No_ : integer;

{----------------------------------------}
{************ END MODEM ****************}
{----------------------------------------}

{-----------------------------------------------}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
{*****************   GRAPH   *******************}
{-------------------------------------------}

const
  { The five fonts available }
  Fonts : array[0..4] of string[13] =
  ('DefaultFont', 'TriplexFont', 'SmallFont', 'SansSerifFont', 'GothicFont');

  { The five predefined line styles supported }
  LineStyles : array[0..4] of string[9] =
  ('SolidLn', 'DottedLn', 'CenterLn', 'DashedLn', 'UserBitLn');

  { The twelve predefined fill styles supported }
  FillStyles : array[0..11] of string[14] =
  ('EmptyFill', 'SolidFill', 'LineFill', 'LtSlashFill', 'SlashFill',
   'BkSlashFill', 'LtBkSlashFill', 'HatchFill', 'XHatchFill',
   'InterleaveFill', 'WideDotFill', 'CloseDotFill');

  { The two text directions available }
  TextDirect : array[0..1] of string[8] = ('HorizDir', 'VertDir');

  { The Horizontal text justifications available }
  HorizJust  : array[0..2] of string[10] = ('LeftText', 'CenterText', 'RightText');

  { The vertical text justifications available }
  VertJust   : array[0..2] of string[10] = ('BottomText', 'CenterText', 'TopText');

var
  GraphDriver : integer;  { The Graphics device driver }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    GraphMode   : integer;  { The Graphics mode value }
    MaxX, MaxY,X,Y  : word;     { The maximum resolution of the screen }
    ErrorCode   : integer;  { Reports any graphics errors }
    MaxColor    : word;     { The maximum color value available }
    OldExitProc : Pointer;  { Saves exit procedure address }

{$F+}
procedure MyExitProc;
begin
  ExitProc := OldExitProc; { Restore exit procedure address }
  CloseGraph;              { Shut down the graphics system }
end; { MyExitProc }
{$F-}

procedure Initialize;
{ Initialize graphics and report any errors that may occur }
var
  InGraphicsMode : boolean; { Flags initialization of graphics mode }
  PathToDriver   : string;  { Stores the DOS path to *.BGI & *.CHR }
begin
  { when using Crt and graphics, turn off Crt's memory-mapped writes }
  DirectVideo := False;
  OldExitProc := ExitProc;              { save previous exit proc }
  ExitProc := @MyExitProc;              { insert our exit proc in chain }
  PathToDriver := '';
  repeat

{$IFDEF Use8514}                        { check for Use8514 $DEFINE }
    GraphDriver := IBM8514;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    GraphMode := IBM8514Hi;
{$ELSE}
    GraphDriver := Detect;                { use autodetection }
{$ENDIF}

    InitGraph(GraphDriver, GraphMode, PathToDriver);
    ErrorCode := GraphResult;             { preserve error return }
    if ErrorCode <> grOK then             { error? }
    begin
      Writeln('Graphics error: ', GraphErrorMsg(ErrorCode));
      if ErrorCode = grFileNotFound then  { Can't find driver file }
      begin
        Writeln('Enter full path to BGI driver or type <Ctrl-Break> to quit:');
        Readln(PathToDriver);
        Writeln;
      end
      else
        Halt(1);                          { Some other error: terminate }
    end;
  until ErrorCode = grOK;
  Randomize;                  { init random number generator }
  MaxColor := GetMaxColor;    { Get the maximum allowable drawing color }
  maxx := getmaxx;
  maxy := 199;
end; { Initialize }

{------------------------- END GRAPH ----------------------------}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
 procedure opengraph;
 var
    grdriver,grmode : integer;
 begin
     grdriver := 7 ; grmode := 0 ;
     initgraph(grdriver,grmode,'');
 end;
{-------------------------------------------------}
{******************** THAI ***************************}
{-------------------------------------------------}

Procedure savefont;
var FontRec : fontRecord;
    dir : string;
Begin

  Assign(Fontfile,'font1.Nlq');
   reset (fontFile);
    for k := 0 to 223 do
       begin
           seek (fontFile,k);read(fontFile,FontRec);
          with fontRec do font1[k] := fontBuff;
       end;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  close(fontfile);
{ writeln('LOADING FONT2.NLQ');
   Assign(Fontfile,'font2.nlq');
  reset (fontFile);
    for k := 0 to 223 do
       begin
          seek (fontFile,k);read(fontFile,FontRec);
          with fontRec do font2[k] := fontBuff;
       end;
 close(fontfile);}  fontcode := 0; mode := 0;
  end;

procedure dot16x16(xx,yy : integer);
const   Ypix : array[0..7] of byte = (1,3,5,7,9,11,13,15);
var byte_No ,dot : integer ; Onebyte,check : byte;

begin
   byte_no :=1;   check := 0;
    repeat
     oneByte := fontDisp[Byte_No];
     for dot := 0 to 7 do
       if oneByte and Bit[Dot]<>0 then
         putpixel(xx,yy-yPix[dot]+check mod 2,1);
         inc(check) ; xx :=   xx+1-check mod 2;
     if check mod 2 = 1 then Byte_no := Byte_No+22;
     If check mod 2 = 0 then Byte_No := byte_No-21;
    until byte_no >= 44;
end;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
procedure dot8x16(xx,yy : integer);
const   Ypix : array[0..7] of byte = (1,3,5,7,9,11,13,15);
var byte_No ,dot : integer ; Onebyte,check : byte;

begin
   byte_no :=1;   check := 0;
    repeat
     oneByte := fontDisp[Byte_No];
     for dot := 0 to 7 do
       if oneByte and Bit[Dot]<>0 then
         putpixel(xx,yy-yPix[dot]+check mod 2,1);
         inc(check) ;
     if check mod 2 = 1 then Byte_no := Byte_No+22;
     If check mod 4 = 0 then Byte_No := byte_No-21;
     If check mod 4 = 0 then inc(xx);
   until byte_no >= 43;
end;
procedure dot8x8(xx,yy : integer);
const   Ypix : array[0..7] of byte = (1,2,3,4,5,6,7,8);
var byte_No ,dot,x,y : integer ; Onebyte,check : byte;

begin
   byte_no :=2; x := xx; y := yy;
    repeat
     oneByte := fontDisp[Byte_No];
     for dot := 0 to 7 do
       if oneByte and Bit[Dot]<>0 then
         putpixel(x,y-yPix[dot],1);
         byte_no := Byte_no+2 ; inc(x); y := yy;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
      If byte_no  = 24 then begin x := xx ; inc(yy) ;y := yy; end;
     until byte_no >= 44;
 end;
 procedure dot16x8(xx,yy : integer);
 const   Ypix : array[0..7] of byte = (1,2,3,4,5,6,7,8);
 var byte_No ,dot : integer ; Onebyte,check : byte;

 begin
    byte_no :=1;      check := 0;
     repeat
      oneByte := fontDisp[Byte_No];
      for dot := 0 to 7 do
        if oneByte and Bit[Dot]<>0 then
          putpixel(xx,yy-yPix[dot]+check mod 2,1);
          inc(check) ; xx :=   xx+1-check mod 2;
      if check mod 2 = 1 then Byte_no := Byte_No+22;
      If check mod 2 = 0 then Byte_No := byte_No-21;
     until byte_no >= 44;
end;

procedure  printFont(Xstart,Ystart:integer; content : string);
const xpix : array [0..3] of byte = (22,11,11,22);
      ypix : array [0..3] of byte = (14,14,8,8);
var   xx, Ydot :integer; ASCII,pass,Efont: byte;
begin
  xx := xstart;content := content +'  ';
  for  k := 1 to length (content)-2 do
     begin
  repeat
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    if (content[k]<#5) then Mode := ord(content[k])-1;
    if (content[k] >#10 )and(content[k] < #13 ) then
          fontcode := ord(content[k])-11;
    if (content[k] < #13) then inc(k);

until content[k] >#31;
  if (content[k] >#223)and(content[k-1] =#207 ) then   inc(k);
      ASCII := ord(content[k]);ydot := 0; pass := 0;efont :=0;
  if (content[k+1] >#223) then
     begin
       case content[k] of
       #217 : begin ASCII := (ord(content[k+1])+14 ); inc(k) end;
       #218 : begin ASCII := (ord(content[k+1])+19 ); inc(k) end;
       #219 : begin ASCII := (ord(content[k+1])+23 ); inc(k) end;
       #220 : begin ASCII := (ord(content[k+1])+27 ); inc(k) end;
       #221 : begin ASCII := (ord(content[k+1])+10 ); inc(k) end;
       end;
     end;
  if (content[k] >#214) then begin xx := xx- xpix[mode];
     ydot := ypix[mode]; end;
  if (content[k] >#216) then   ydot := -ypix[mode];
  if (content[k] <#127) then efont :=2;
  if (efont = 2) and( mode>1)   then efont := 1;
  if (content[k] >#209)  and (content[k]<#213) then ASCII := 131;
 repeat
       inc (pass);
       case fontcode of
        0 : fontdisp := font1[ASCII-32];
        1 : fontdisp := font2[ASCII-32];
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        end;
        case mode of
        0 : dot16x16(xx,yStart + ydot + efont);
        1 : dot8x16(xx,yStart + ydot + efont);
        2 : dot8x8 (xx,yStart + ydot + efont);
        3 : dot16x8(xx,yStart + ydot + efont);
        end;
      if (content[k] >#170)  and (content[k]<#175) and (pass = 1) then
       begin   ASCII := ASCII -36;  ydot := ypix[mode] ; pass := 2 ; end;
      if (content[k] >#209)  and (content[k]<#213) and (pass = 1) then
       begin   ASCII := ord(content[k])-78;ydot :=-ypix[mode] ; pass := 2 ; end;
      if (content[k] = #194)  and (pass = 1) then
       begin   ASCII := 139;  ydot := ypix[mode] ; pass := 2 ; end;
      if (content[k+1] = #207)  and (content[k+2]<#215) and (pass = 1) then
       begin   ASCII := 222;  ydot := -ypix[mode] ; pass := 2 ; end;
      if (content[k+1] = #207)  and (content[k+2]>#223) and (pass = 1) then
       begin   ASCII := ord(content[k+2])+6;  ydot := -ypix[mode] ; pass := 2 ; end;
   until  (pass = 1) or (pass = 3);
        xx := xx + xpix[mode];
    end;
  end;


{-----------------END  THAI ---------------------}

 Procedure textShow(font,size : word ;Mode : char);
 begin
  case mode of
  'v','V' : setTextStyle(Font,vertDir,size);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  'h','H' : setTextStyle(Font,horizDir,size);
  end;
 end;
 procedure readfuncKey (var key : char);
begin
   key := readkey ;
   if key = #0 then
   begin
     funckey := true ;
     key := readkey;
   end
 else
   funckey := false;
end;
procedure waitToGo;
 var ch : char;
 begin
    repeat until keyPressed;
    ch := readkey ;
    if ch = #27 then  begin end
 end;

 procEdure arrowUp(x,y : integer);
 begin
       moveto(x,y);
       lineto(x,Y-10);
       lineto(x-5,y -5);
       moveto(x,y-10);
       lineto(X+5,Y-5);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
END;

procEdure arrowDOWN(x,y : integer);
begin
      moveto(x,y);
      lineto(x,Y-10);
      moveto(x,y);
      lineto(x-5,y-5);
      moveto(x,y);
      lineto(X+5,Y-5);
END;

 procEdure arrowR(x,y : integer);
begin
      moveto(x,y-5);
      lineto(x+15,Y-5);
      lineto(x+7,y -8);
      moveto(x+15,y-5);
      lineto(X+7,Y-2);
END;


 procEdure arrowL(x,y : integer);
begin
      moveto(x,y-5);
      lineto(x+15,Y-5);
      moveto(x,y-5);
      lineto(x+5,y -8);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        moveto(x,y-5);
        lineto(X+5,Y-2);
   END;
   PROCEDURE  ARROW(X,Y : INTEGER);

   BEGIN

      ARROWUP(X,Y);
      ARROWDOWN(X+25,Y);
      ARROWR(X+50,Y);
      ARROWL(X+75,Y);
  END;

 {-----------------------------------------}
 {*********** control ac line  *************}
 {-----------------------------------------}

  procedure clear(x1,y1,x2,y2:word);
  begin
    setviewport(x1,y1,x2,y2,clipon);
    clearviewport;
    setviewport(0,0,getmaxx,getmaxy,clipon);
  end;

PROCEDURE  moveblock0(x1,y1,x2,y2 : word);
var
pix1,pix2 : pointer;
size : word ;
x: integer;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
begin
size := imageSize(x1,y1,x2,y2);
getmem(pix1,size); getimage(x1,y1,x2,y2,pix1^);
putimage(x1,y1,pix1^,4);
freemem(pix1,size);
end;

Procedure mempic(x1,y1,x2,y2 : word;VAR PIX : POINTER );

var size : integer;
begin
  size := imagesize(x1,y1,x2,y2);
  getmem(pix,size);
  getimage(x1,y1,x2,y2,pix^);
end;

procedure block;
var i,j,x1,x2,y1,y2,y3 : integer;k : char;
begin
 if graphdriver = 1 then begin pic := 55 ;y3 := 78; y2 := 80 end
  else begin y3 := 88 ;y2 := 88 ;pic := 55 end;
   textshow(0,3,'h');k := '0';
   x1 := 20 ;x2 := 70 ;
   y1 := 40 ;
   for i := 1 to 3 do begin
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    for j := 1 to 3 do
      begin
      inc(k);
        rectangle (x1,y1,x2,y2); outtextxy(x1+15,y1+10,k);
        x1 := x1 +PIC;
        x2 := x2 +PIC ;
        end;
         x1 := 20 ;
         x2 := 70;
         y1 := y1 +PIC;
         y2 := y2 +PIC;
           end;
       moveblock0(22,42,68,y3 );
end;
procedure setpat(x,y,p : word);

begin
 setfillstyle(p,1);
 floodfill(x,y,blue);
end;

procedure showpan (y,z,t,w : integer);
var x : integer;

begin
  x := y;
  repeat
      setcolor(1); circle(getmaxx-z,t,x);
      setcolor(0); circle(getmaxx-z,t,x);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        x := x+3;
        if x > w then x := y;
    until keypressed;
       setcolor(1);
end;

procedure showredio;
var x,y : integer;

begin
    x := 0 ;
   repeat
        setcolor(1); circle(getmaxx-253,154,x);
        setcolor(0); circle(getmaxx-253,154,x);
        setcolor(1); circle(getmaxx-53,154,x);
        setcolor(0); circle(getmaxx-53,154,x);
        x := x+2;
        if x > 30 then x := 0;
   until keypressed;
      setcolor(1);
end;

procedure show ;
var x,y,z,x1,r : integer;
begin
x := 250;       y := 120;
z :=0;       x1 :=0;
 if graphdriver = 1 then r := 50  else r:=40 ;
  repeat
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
      textshow(0,z,'h');
      setcolor(1) ;
      outtextxy(getmaxx-x,y,'KMITL');   delay(10);
      setcolor(0); outtextxy(getmaxx-x,y,'KMITL');
      z := z+1;
      textshow(0,z,'h');
      if z = 3 then z := 0;
       setcolor(1);outtextxy(getmaxx-x,y,'KMITL');
       setcolor(0); outtextxy(getmaxx-x,y,'KMITL');
       setcolor(1); circle(getmaxx-190,135,x1);
       setcolor(0); circle(getmaxx-190,135,x1);
        x1 := x1+3;
        if x1 > r then x1 := 0;
   until     keypressed;
   setcolor(1);
end;

procedure   showlame;
var x,y,z : integer;
begin

  repeat
       setpat(getmaxx-210,90,1);  delay(70);
       clear(getmaxx-214,86,getmaxx-96,99); delay(70);
  until     keypressed;
end;

procedure finey(r,x : integer;var y : integer);
 begin
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    until keypressed;

    setcolor(1);

    end;

procedure Tv;
begin
  rectangle(getmaxx-259,96,getmaxx-91,179);
  rectangle(getmaxx-253,103,getmaxx-135,173);
  rectangle(getmaxx-130,103,getmaxx-96,128);
  circle(getmaxx-113,153,15);
  line(getmaxx-176,96,getmaxx-200,52);
  line(getmaxx-180,96,getmaxx-30,52);
  setpat(getmaxx-93,98,0);

end;

procedure redio;
begin
    circle(getmaxx-115,171,3);  SETPAT(getmaxx-115,171,1);
    circle(getmaxx-145,171,3);  SETPAT(getmaxx-145,171,1);
    circle(getmaxx-135,171,3);  SETPAT(getmaxx-135,171,1);
    circle(getmaxx-125,171,3);  SETPAT(getmaxx-125,171,1);
    circle(getmaxx-253,154,20); SETPAT(getmaxx-253,154,0);
    circle(getmaxx-253,119,10); SETPAT(getmaxx-253,119,0);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
      circle(getmaxx-53,154,20);  SETPAT(getmaxx-53,154,0);
      circle(getmaxx-53,119,10);  SETPAT(getmaxx-53,119,0);
      rectangle(getmaxx-285,96,getmaxx-220,179); {SETPAT(getmaxx-284,97,10);}
      rectangle(getmaxx-215,150,getmaxx-91,179);
      rectangle(getmaxx-160,154,getmaxx-95,163);
      rectangle(getmaxx-210,154,getmaxx-175,170);
      rectangle(getmaxx-85,96,getmaxx-20,179);
   end;
Procedure pan;
var r,r1,y : integer;
  begin
  if graphdriver = 1 then begin r1 := 20;  r := 75; y := 115 end
  else begin r := 50 ; r1 := 15;y := 110 end;
     circle(getmaxx-155,95,r);
     inc(r) ;
     circle(getmaxx-155,95,r);
     circle(getmaxx-155,95,5);
     setpat(getmaxx-155,95,1);
     line(getmaxx-160,132,getmaxx-180,170);
     line(getmaxx-150,132,getmaxx-130,170);
     rectangle(getmaxx-215,170,getmaxx-95,185);
     circle(getmaxx-155,75,r1);
     circle(getmaxx-129,y,r1);
     circle(getmaxx-180,y,r1);
     circle(getmaxx-155,95,10);

   end;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
procedure   lame;
 begin
    rectangle(getmaxx-215,85,getmaxx-95,100);
    circle(getmaxx-155,65,7);
    line(getmaxx-222,65,getmaxx-162,65);
    line(getmaxx-222,65,getmaxx-222,89);
    line(getmaxx-222,89,getmaxx-215,89);
    line(getmaxx-148,65,getmaxx-88,65);
    line(getmaxx-88,65,getmaxx-88,89);
    line(getmaxx-88,89,getmaxx-95,89);
    rectangle(getmaxx-130,110,getmaxx-95,120);
    line(getmaxx-88,94,getmaxx-88,112);
    line(getmaxx-88,94,getmaxx-95,94);
    line(getmaxx-88,112,getmaxx-95,112);
    line(getmaxx-88,117,getmaxx-95,117); moveto(getmaxx-88,117);
    lineto(getmaxx-88,155);
    lineto(getmaxx-150,155);
    lineto(getmaxx-150,160);
    moveto(getmaxx-215,94);
    lineto(getmaxx-222,94);
    lineto(getmaxx-222,155);
    lineto(getmaxx-155,155);
    lineto(getmaxx-155,160);
  end;


procedure up;
begin
  if num > 3 then
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    begin
       moveblock0(xu+2,yu+2,xd-2,yd-2);
       num := num -3 ;  yu := yu-PIC; yd := yd-PIC;
       moveblock0(xu+2,yu+2,xd-2,yd-2);
    end
    else begin sound(1000); delay(100); nosound end;
 end;
procedure down;
begin
    if num < 7 then
    begin
      moveblock0(xu+2,yu+2,xd-2,yd-2);
      num := num + 3 ; yu := yu+PIC; yd := yd+PIC;
      moveblock0(xu+2,yu+2,xd-2,yd-2);
    end
    else begin sound(1000) ;delay(100); nosound end;
 end;
procedure r;
begin
    if num <> 3 then
    if num <> 6 then
    if num <> 9 then
    begin
     moveblock0(xu+2,yu+2,xd-2,yd-2);
     num := num +1 ;    xu := xu + PIC;xd := xd +PIC;
     moveblock0(xu+2,yu+2,xd-2,yd-2);
    end
    else begin sound(1000); delay(100); nosound end;
 end;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
procedure l1;
begin
   if num <> 1 then
   if num <> 4 then
   if num <> 7 then
   begin
        moveblock0(xu+2,yu+2,xd-2,yd-2);
        num := num -1 ;    xu := xu - PIC;xd := xd -PIC;
        moveblock0(xu+2,yu+2,xd-2,yd-2);
   end
   else begin sound(1000); delay(100); nosound end;
 end;

 procedure  testkey0(ch : char);
 begin
    if funckey then
       case  ch of
    { up } #72 :  up;
    {down} #80 :  Down;
    { -> } #77 :  R;
    { <- } #75 :  L1;

       end ;

 end;

  procedure order (active : integer);
  begin
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
   o(chr($1f));  delay(100);
active := (Num * 2 ) or active ;
  o(chr(active));
end;

procedure ONorOFF1 ;
begin
      clear(getmaxx-460,96,getmaxx-320,126);
        textshow(0,2,'h');
     if system1[num]  = on then
       begin
        order($01);
        outtextxy(getmaxx-458,100,'  ON   ');
        case num of
         1 : show;
         2 : showpan(55,155,95,65);
         3 : showradio;
         4 : showlame;
         5 : show;
         6 : show;
         7 : showpan(53,155,95,65);
         8 : showlame;
         9 : showradio;
        end ;


       end
      else
       begin
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
      order($00);
      outtextxy(getmaxx-458,100,'  OFF    ');
     end;
end;

procedure ONorOFF ;
begin
   clear(getmaxx-399,96,getmaxx-320,126);
       textshow(0,2,'h');
     if system1[num]  = on then
       begin
        order($01);
        outtextxy(getmaxx-400,100,'ON ');
        case num of
         1 : show;
         2 : showpan(70,155,95,90);
         3 : showredio;
         4 : showlame;
         5 : show;
         6 : show;
         7 : showpan(70,155,95,90);
         8 : showlame;
         9 : showredio;
        end ;

       end
       else
       begin
        order($00);
```

เอกสารนี้เป็นสงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
      outtextxy(getmaxx-400,100,'OFF ');
    end;
end;

procedure control(class : byte);
  begin
    rectangle(0,0,getmaxx ,maxY);
    line(0,30,getmaxx,30);
    clear(1,1,getmaxx-1,29);
    textshow(0,2,'h');
    printfont(5,23,#11+#2 +'เลือกโดย KEY    ');
 ARROW(150,23);
    printfont(5,23,#11+#2 +'                    ถ้าต้องการกด ENTER KEY  และ  ESC ออ
    textshow(0,1,'h');
    block;
    rectangle(getmaxx-300,45,getmaxx-10,195(200));
    rectangle(getmaxx-299,46,getmaxx-11,199);
      if graphdriver = 1 then  yd := 80
  else yd := 90;
    xu := 20 ;xd := 70 ;
    yu := 40 ;
    TV;
    num := 1; on := true; off := false;
    open := off;

    repeat
      readfunckey(ch);
      testkey0(ch);
      clear(getmaxx-299,46,getmaxx-11,194(199));
```

เอกสารนี้จัดทำขึ้นเพื่อประโยชน์ทางการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
      case num of
      1 : tv;
      2 : pan;
      3 : redio;
      4 : lame;
      5 : tv;
      6 : Tv;
      7 : pan ;
      8 : lame;
      9 : redio;
       end;
     if ch = #13 then begin system1[num] :=  not(system1[num]);
    if graphdriver = 1 then  onoroff  else onoroff1 ;    end;
     until ch = #27;
     clear(1,1,getmaxx-1,29);
  end;

{-----------------END AC LINE CONTROL---------------}



{-------------------------------------------------}
{******************  MENU ***************************}
{-------------------------------------------------}


procedure getsize;
var size3,size4,size5 : integer;
begin
    size1 := imageSize(10,40,200,190);
    size2 := imageSize(210,40,410,150);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    getmem(pass1,size1);
    getmem(pass2,size2);
    getimage(10,40,200,190,pass1^);
    getimage(210,40,410,150,pass2^);
    size3 := imageSize(10,40,200,190);

    size4 := imageSize(210,40,410,150);
    size5 := size3;
    getmem(pic1,size3);
    getmem(pic2,size4);
    getmem(pic3,size5);
    getmem(pic4,size5)

end;

procedure freememory;

begin
  freemem(pic1,size3);
    freemem(pic2,size4);
    freemem(pic3,size5);
    freemem(pic4,size5);
    freemem(pass1,size1);
    freemem(pass2,size2);

 end;
 procedure clearpix(k : integer);
 begin
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    if k = 1 then begin
       setviewport(10,40,200,190,true);
       clearviewport; end
    else
    begin
       setviewport(210,40,410,150,true);
    clearviewport end ;
    setviewport(0,0,getmaxx,getmaxy,true);
end;

PROCEDURE  moveblock;
var
pix1,pix2 : pointer;
size : word ;
x: integer;

begin
size := imageSize(xup[blockmain1],yup[blockmain1],xdown[blockmain1],ydown[blockmain]);
getmem(pix1,size); getimage(xup[blockmain],yup[blockmain1],xdown[blockmain1],ydown[bloc
putimage(xup[blockmain1,yup[blockmain1,pix1^,4);
freemem(pix1,size);
end;

Procedure block1;
begin
clear(10,40,200,190);
  RECTANGLE(10,40,200,190);
 MOVETO(10,60);LINETO(200,60);
printfont(20,56,#11+#3+' ติดต่อ MODEM');
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
printfont(20,80,#11+#3+'1.ส่งข้อความ ');
printfont(20,100,#11+#3+'2.ดูข้อความ ');
printfont(20,120,#11+#3+'3.Drive      '+drive);
printfont(20,140, #11+#3+'4.Directory');
printfont(20,160, #11+#3+'5.ติดตั้งระบบสื่อสาร');
printfont(20,180, #11+#3+'6.ออกจาก PROGRAM ');
end;
procedure block2;
begin


    getimage(10,40,200,190,pic1^); clear(10,40,200,190);
    getimage(210,40,410,150,pic2^);
    putimage(10,40,pass1^,0);
      blockmain := 2;
        clear(210,40,410,150);


        rectangle(210,40,410,150);
  line(210,60,410,60);
  printfont(220,56,#11+#3+ '    ควบคุมอุปกรณ์ ');
  printfont(220,80,#11+#3+ '      ชั้นที่ 1');
  printfont(220,100,#11+#3+'      ชั้นที่ 2');
  printfont(220,120,#11+#3+'      ชั้นที่ 3');
  printfont(220,140,#11+#3+'      ชั้นที่ 4');
  first := false;
    moveblock;
END;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Procedure memorypic(x1,y1,x2,y2,z : word);

var size : integer; pix : pointer;
begin
  size := imagesize(x1,y1,x2,y2);
  getmem(pix,size);
  getimage(x1,y1,x2,y2,pix^);
  clear(x1,y1,x2,y2);
  if z = 1  then putimage(x1,y1-20,pix^,0)
  else  putimage(x1,y1+20,pix^,0) ;
  freemem(pix,size);
end;
{----------END MENU----------}

{----------DIRECTORY----------}
procedure  directory ;
var py ,px,i  : integer;
begin
for  i := 1 to 100 do   dirname[i] := ('            ' );
rectangle ( 10,40,420,167);
dir := 1; dis := 0;
px := 30;
py := 50;

findfirst(drive+'*.*' ,anyfile,dirinfo);
while doserror = 0 do begin
   findnext (dirinfo);
   if (doserror = 0) then  begin
    dirname[dir] := dirinfo.name;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  if py < 169 then
    begin
     outtextxy(px,py,dirNAME[DIR]);  px := px + 120;    inc(dis);
   if px > 300  then begin  py := py + 20 ;  px := 30 end
     end;
  inc(dir);
  end;
end;
end;

procedure writeDir( x,y,s : integer; var z : integer);
var i,j,k : integer;
begin
   for i := 0 to 2 do
   begin
   outtextxy(x,y,dirname[z+i]);
                  x := x + 120;
   end;

end;

procedure scrollup;

begin
   memorypic(xa,ya+21,Xb,Yb-4,1);
   writeDir(30,150,1,dis1);
end;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
procedure scrolldown;


var  picdown : pointer;
begin
  memorypic(xa,ya,Xb,Yb-23,2);
  writedir( 30,50,2,dis1);
end;



procedure sound1;
begin
 sound(3000);
 delay(100);
 nosound;
end;



procedure moveUp ;
var i : integer; pass : boolean;
begin
  pass := true;
   case blockmain of
   1 : i := 1;
   2 : i := 1;
   3 :  begin i := 3;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            if  row = 1 then if  (dirnum-3)>0 then pass := false
            else pass := true;
        end;
    end;

if pass then
  If BlockNum[BlockMain] > i then
  begin
     moveBlock; dec(row); dis1 := dis1 -3;
     ydown[blockmain] := ydown[blockmain] - incdec ;
     yup[blockmain] := yup[blockmain] -incdec;
     IF BLOCKmain = 3 then
      begin
        dirNum := dirNum -3;
        blocknum[3] := blocknum[3] -3;
      end
     else
     dec(BlockNum[BlockMain]);
     moveblock;

  end
  else   sound1
  else
    if blockmain = 3 then
    begin
     moveblock;  dis1 := dis1 - 3; scrolldown; ydown[3] :=  60; yup[3] := 45;
     moveblock;
       dirNum := dirNum - 3;  blocknum[3] := blocknum[3] -3;
     end;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
end;
procedure movedown;
var i :byte; down : integer;  pass : boolean;
begin
    pass := true;
 case blockmain of
 1 : i := 6;
 2 : i := 4;
 3 : begin  i := 100 ;
        down := dir - dirnum;
        if down > 3    then
        pass := true else pass := false;
        if row = 6 then pass := false;
     end;
 end;


 IF  pass then

   If BlockNum[BlockMain] <> i then
  begin
     moveblock;
     inc(row);dis1 := dis1 +3;
     ydown[blockmain] := ydown[blockmain] + incdec ;
     yup[blockmain] := yup[blockmain] +incdec;
   IF BLOCKmain = 3 then
      begin
        dirNum := dirNum +3;
        blocknum[3] := blocknum[3] +3;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
      end
    else
        inc(BlockNum[BlockMain]);
      moveblock;

  end
  else
       sound1
       else   if blockmain = 3 then

          begin
          moveblock ;          dis1 := dis1 +3;
          scrollup  ;moveblock;
            dirNum := dirNum +3;
        blocknum[3] := blocknum[3] +3;

          end;

end;


procedure mover1;
var i : integer;
begin
  If 1 < 3 then
  begin
     1 := 1+1;
     moveBlock;
     xdown[blockmain] := xdown[blockmain] + 125 ;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    xup[blockmain] := xup[blockmain] +125;
    inc(BlockNum[3]);
    moveblock;
    inc(dirnum)
  end
  else
    sound1;
end;

procedure move11;
var i : integer;
begin
  if l  > 1 then
  begin
    l := l-1;
    moveBlock;
    xdown[blockmain] := xdown[blockmain] - 125 ;
    xup[blockmain] := xup[blockmain] -125;
    dec(BlockNum[BlockMain]);
    moveblock;
    dec(Dirnum);
  end
  else
    sound1;
end;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
procedure moveR;
begin

if  blockmain = 1 then
  begin

if  not first then
  begin
     blockmain := 2;
    getimage(10,40,200,190,pic1^);
    clear(10,40,200,190);
    putimage(10,40,pass1^,0);
    getimage(210,40,410,150,pic2^);
    clear(210,40,410,150);
    putimage(210,40,pass2^,0);
   end
 else
   block2
    end
   else sound1 ;
   end;



procedure moveL;
begin
  if blockmain = 2 then
    begin

     blockmain := 1;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    getimage(210,40,410,150,pass2^);     clear(210,40,410,150);
    putimage(210,40,pic2^,0);
    getimage(10,40,200,190,pass1^);        clear(10,40,200,190);
    putimage(10,40,pic1^,0);
   end
  else
     sound1;
end;
{----------------------END DIRECTORY-----------------------------}

procedure decchar(var xD,yD : integer);
begin
    clear(xD-17,yD-17,xD,yD+17);
    xD := xD-17;
end;

procedure writechar(x,y : integer; VAR data : filen);
var line ,i : integer;
    CH : CHAR ; EXIT : BOOLEAN;
BEGIN
     DATA := ' ';             exit := false;
    I := 1;
      REPEAT
       ch := readkey;
      if ch = #08 then BEGIN  decchar(x,y); DATA[I-1] := ' '; I := I-1 END;
      if ch > #32 then
           begin
            textshow(0,2,'h');
            DATA[I] := CH ;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            outtextxy(X,Y-14,DATA[I]);
            X := X + 17 ; INC(I);
           end;
      if ch = #13 then exit := true;
      if ch = #27 then exit := true;
      UNTIL exit;
    textshow(0,1,'h');   for line := 1 to i-1 do  namedata[line] := data[line];
END;

PROCEDURE WRITELNCHAR(X,Y : INTEGER ; DATA : STRING);
BEGIN
     PRINTFONT(X,Y,#11+#3+DATA);
END;

PROCEDURE SENDDATA;
VAR X,Y,LINE : INTEGER;

BEGIN
   LINE := 20;
   X := (MaxX DIV 2)- 128;
   Y := 60 ;
   NAMEDATA := '              ';
   NAME     := '      ';
   NOPC     := '    ';
  PRINTFONT(X,Y,#11+#1 +'FOME   ');  Y := Y + LINE;
  PRINTFONT(20,Y,#11+#2 +'NAME    ');
             X := 100 ; ;
             WRITECHAR(90,Y,NAME);
             Y := Y + LINE;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  PRINTFONT(20,Y,#11+#2 +'NO.    ');
  WRITECHAR(90,Y,NOPC);
  Y := Y +LINE;
  PRINTFONT(20,Y,#11+#2 +'ชื่อ    ');
  WRITECHAR(120,Y,NAMEDATA);
  Y := Y + LINE;
  PRINTFONT(20,Y,#11+#2 +'ต้องการดู DIRECTORY    ');
END;



procedure  display;
var size3,rowp : integer;  out : boolean;
begin
blockmain := 3 ;
DIRNUM := 1;
blocknum[3] := 1; dis1 := 1; dis := 1 ;
  rowp  := row;
  row := 1;  l := 1;
     xdown[3] := 120;
     ydown[3] := 60;
     xup[3]   := 15;
     yup[3]   := 45;
  clear( 10,40,MaxX-29,MaxY-31);
directory;   moveblock;
RECTANGLE(MaxX-200,60,MaxX-60,90);  {show dir}
CLEAR(MaxX-199,61,MaxX-61,89);
   OUTTEXTXY(MaxX-193,74,#11+#3+DIRNAME[1]);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
   CLEAR(MaxX-199,61,MaxX-179,89);
   CLEAR(20,MaxY-29 ,MaxX-30,MaxY-1);
   printfont(20,MaxY-3,#11+#2 +' เลือกโดย KEY ');
 ARROW(185,MAXY-3);
   printfont(20,MaxY-3,#11+#2 +'                                    ถ้าต้องการกด RETRUE KEY');
   SOUND1; out := false;
repeat
  readfunckey(ch);
   if funckey then
      case  ch of
   { up } #72 :  moveup;
   {down} #80 :  movedown;
   { -> } #77 :  moveR1;
   { <- } #75 :  moveL1;
       ELSE SOUND1
      end
   ELSE
      IF CH = #13 THEN  begin NAMEDATA := DIRNAME[DIRNUM];
          out := true end;
        if ch = #27 then out := true;
   CLEAR(MaxX-199,61,MaxX-61,89);
   OUTTEXTXY(MaxX-193,74,#11+#3+DIRNAME[DIRNUM]);
   CLEAR(MaxX-199,61,MaxX-179,89);

until  out;
  blockmain := 1;  row := rowp;
end;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
procedure clearpic;
begin

    getimage(10,40,200,190,pic3^);    clear(10,40,200,190);
    putimage(10,40,pic4^,0);
end;


{----------------------------------------}
{****************RS 232*******************}
{----------------------------------------}

PROCEDURE  movebank(x1,y1,x2,y2 : integer);
var
pix1,pix2 : pointer;
size : word ;
x: integer;

begin
size := imageSize(x1,y1,x2,y2);
getmem(pix1,size); getimage(x1,y1,x2,y2,pix1^);
putimage(x1,y1,pix1^,4);
freemem(pix1,size);
end;

procedure up1(var k1 ,Y1 ,Y2 : INTEGER);
begin
  if k1 > 1 then
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    begin
       movebank(92,Y1+2,178,Y2-2);
         dec(K1);
         Y1 := Y1 - 35;
         Y2 := Y2 - 35;
       movebank(92,Y1+2,178,Y2-2)
    END;
end;
procedure DN1(var k1 ,Y1 ,Y2 : INTEGER);
begin
  if k1 < 4 then
    begin
       movebank(92,Y1+2,178,Y2-2) ;
         inc(K1);
         Y1 := Y1 + 35;
         Y2 := Y2 + 35;
       movebank(92,Y1+2,178,Y2-2)
    END;

end;

procedure baudR(var  baud :integer);
begin
    if baud = 8 then baud :=  1 else inc(baud);
   clear(92,34,178,58);
   case baud of
   1  : begin baudbit := '1' ;brat := ' 50'   end;
   2  : begin baudbit := '5' ;brat := ' 150'  end;
   3  : begin baudbit := '6' ;brat := ' 300'  end;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
   4  : begin baudbit := '7' ;brat := ' 600'  end;
   5  : begin baudbit := '8' ;brat := '1200'  end;
   6  : begin baudbit := 'b' ;brat := '2400'  end;
   7  : begin baudbit := 'd' ;brat := '4800'  end;
   8  : begin baudbit := 'f' ;brat := '9600'  end;
   end;
   outtextxy(20,45,'BAUD         '+brat );
   movebank(92,34,178,58);
end;

procedure bitR(var  bit1 :integer);



begin
   if bit1 = 2 then bit1 :=  1 else inc(bit1);
   clear(92,32+37,178,60+33);
   case bit1 of
   1  : begin  databit := '7' ;bitd := '7' end;
   2  : begin  databit := '8' ;bitd := '8' end;
   end;
   outtextxy(20,45+35,'Bit           '+bitd );
   movebank (92,32+37,178,60+33);
end;

procedure stopR(var  stop :integer);
begin
   if stop = 3 then stop :=  1 else inc(stop);
    clear(92,32+72,178,60+68);
    case stop of
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
   1  : begin  stopbit := '0' ; sbit :=  ' 1';  end;
   2  : begin  stopbit := '1' ; sbit :=  '1.5'; end;
   3  : begin  stopbit := '2' ; sbit := ' 2';  end;
   end;
   outtextxy(20,45+70,'Stop          '+sbit );
   movebank  (92,32+72,178,60+68);
end;

procedure paraR(var  para :integer);
var b: string[6];
begin
  if para = 3 then para :=  1 else inc(para);
   clear(92,32+107,178,60+103);
   case para of
   1  : begin  paritybit := '1'; pbit := ' odd';  end;
   2  : begin  paritybit := '2'; pbit := 'even';  end;
   3  : begin  paritybit := '3'; pbit := 'none';  end;
   end;
   outtextxy(20,45+35+70,'Parity       '+pbit );
   movebank(92,32+107,178,60+103);
end;

procedure increas( k1 : integer;var baud ,bit1 ,stop,para : integer);

begin

   case  k1 of
   1 :  baudR(baud);
   2 :  bitR(bit1) ;
```

เอกสารนี้เป็นเอกสารสงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  3 :   stopR(stop);
  4 :   paraR(para);

  end;
end;

procedure computer;
begin

rectangle(maxx-415,55,maxx-5,35);
rectangle(maxx-415,165,maxx-5,60);
rectangle(maxx-350,130,maxx-250,150);
rectangle(maxx-410,145,maxx-355,150);
bar3d(maxx-250,120,maxx-160,110,10,true);
{rectangle(maxx-250,120,maxx-160,110);}
rectangle(maxx-160,130,maxx-60,150);
rectangle(maxx-55,145,maxx-10,150);
moveto(maxx-250,140);
lineto(maxx-230,140);
lineto(maxx-230,120);
moveto(maxx-160,140);lineto(maxx-180,140);
lineto(maxx-180,120);
moveto(maxx-320,130);
lineto(maxx-320,125);
lineto(maxx-345,125);
lineto(maxx-345,89);
lineto(maxx-280,89);
lineto(maxx-280,95);
lineto(maxx-260,115);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
{-------------------------------------------------}
{******************MODEM AC LINE*********************}
{-------------------------------------------------}

procedure checkcom(var command : str30);
var i : integer;
begin
   command := '                                ';
   i := 1 ;
  repeat
    recive(ch1);
    if ch1  <> #13 then command[i] := ch1[1];
        inc(i);
 until ch = #13 ;
end;

procedure  directory1(mask : string;writ : boolean);
begin
   findfirst(mask,anyfile,dirinfo);
     while doserror = 0 do
         begin
       findNext(dirinfo);
         if (dosError = 0) then

              o(dirinfo.name+#13);
         end;
      end;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
outtextxy(20,y,'Parity      '+pbit );
movebank(92,34,178,58);
y1 := 32; y2 := 60;
computer;
printfont(20,MaxY-3,#11+#2 +'เลื่อกโดย key  ');
ARROWUP(160,MAXY-3);
ARROWDOWN(185,MAXY-3);
printfont(20,MaxY-3,#11+#2 +'                    และ ENTER KEY  ,   ESC ออกจากการติดตั้ง
sound1;
   repeat
     readfunckey (ch);
     if funckey then
     case ch of
      { up } #72 :  Up1(k1,Y1,Y2);
      {down} #80 :  dn1(k1,Y1,Y2);
      end
      else
        if ch = #13 then increas(k1,b,d,s,p);

   until ch = #27;
 init_com(comnum,baudbit,databit,paritybit,stopbit);
 clear(21,31,MaxX-4,MaxY-31);
CLEAR(20,MaxY-29 ,MaxX-1 ,MaxY-1);putimage(10,40,pic3^,0);
end;

{-----------------END RS_232 ----------------------}


procedure  controlac(num : byte);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
begin
     init_com('1','1','8','3','2');

    getimage(210,40,410,150,pic3^);        clear(210,40,410,150);
    putimage(210,40,pic4^,0);
      clear(1,31,MaxX-1,MaxY-5);
     control(num);

   clear(1,31,MaxX-1,MaxY-31);LINE(0,MaxY-30,MaxX,MaxY-30);
CLEAR(1,MaxY-29 ,MaxX-1 ,MaxY-1);putimage(210,40,pic3^,0);
 printfont(100,23,#11+#1+'AC LINE MODEM  V 1.0 ');
     init_com(comnum,baudbit,databit,paritybit,stopbit);
     textshow(0,1,'h');
end;

function create(var fi : fivart ) : boolean;
begin
  {$I-} reset (fi) ;{$I+}
  create := ioresult = 0 ;
end;


procedure writefile( filename : filen);
var data : char; out1,out2 : boolean;
begin
    out1 := false;
    out2 := false;

   assign (FILEPRO,drive+filename);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
if create(FILEPRO) then
  begin
repeat
   while not eof(FILEPRO)  and (not keypressed) do
       begin
         read (FILEPRO,data);
         write(data);
       end;
   if eof(filepro) then out1 := true else
   begin
     ch := readkey;
     if ch = ^s then
       begin
         repeat
          ch := readkey ;
           case ch of
           ^q : out2 := true;
           ^c : out2 := true;
           end;
         until out2;
         out2 := false;
       end;
     if ch = ^c then out1 := true ;
   end;
   until out1;
   end
   else
    writeln(' NO  FILE');
end;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
lineto(maxx-260,125);
lineto(maxx-310,125);
lineto(maxx-310,130);
moveto(maxx-90,130);
lineto(maxx-90,125);
lineto(maxx-65,125);
lineto(maxx-65,89);
lineto(maxx-140,89);
lineto(maxx-140,95);
lineto(maxx-150,115);
lineto(maxx-150,125);
lineto(maxx-100,125);
lineto(maxx-100,130);
printfont(maxx-410,53,#11+#2 +'          ชุดอ่าน RS 232C      ');

end;

procedure rs232(var b,d,s,p : integer);
var y,y1,y2,k1 : integer;
begin
    y := 60; k1 := 1;
     clearpic;
textshow(0,1,'h');
y1 := 32; y2 := 60;
for y := 1 to 4 do    begin
rectangle(90,y1,180,y2);    y1 := 35 +y1; y2 := y2 + 35;end;y := 45;
outtextxy(20,y,'BAUD         '+brat );  y := y +35;
outtextxy(20,y,'Bit            '+bitd );   y := y+35;
outtextxy(20,y,'Stop           '+sbit );  y := y+35;
```

เอกสารนี้มีไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
procedure shiftdata(cha : char);
var i : integer;

begin
 for i := 2 to 5 do shift[i-1] := shift[i] ;
     shift[5] := cha;
 end;
PROCEDURE  SHIFEDATA;
VAR PIX1 : POINTER;
    SIZE : INTEGER;
BEGIN
size := imageSize(130,80,GETMAXX-110,100);
getmem(pix1,size); getimage(130,80,GETMAXX-110,100,pix1^);
CLEAR(120,80,GETMAXX-2,100);
putimage(120,80,pix1^,0);
freemem(pix1,size);
END;

PROCEDURE  WRITECHARTOSEND(CHAREC : CHAR);
BEGIN
  SHIFEDATA;
  OUTTEXTXY(GETMAXX-120,80, +CHAREC);

 END;

 procedure sendfile( filename : filen);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
var data : char; out1,out2 : boolean;i : integer;
begin
    out1 := false;
    out2 := false;
   assign (FILEPRO,drive+filename);
   if create(FILEPRO) then
      begin
   CLEAR(20,MaxY-29 ,MaxX-30,MaxY-1);
   printfont(20,MaxY-3,#11+#2 +'กำลังส่ง  FILE DATA  ชื่อ    ' +namedata);
   SOUND1;
    o(star + cr);
    DELAY(1000);
    O(FILENAME + CR);
    for i := 1 to length(filename)  do
    WRITECHARTOSEND(filename[i]);
    for i := 1 to 5 do
    writechartosend(' ');
    DELAY(1000);
    repeat
      while (not eof(FILEPRO)) and (not keypressed)  do
          begin
            read (FILEPRO,data);
            WRITECHARTOSEND(DATA);
            o(data);
          end;
    if eof(filepro) then out1 := true else
      begin
        ch := readkey;
        if ch = ^s then
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        Begin
          repeat
           ch := readkey ;
            case ch of
            ^q : out2 := true;
            ^c : out2 := true;
            end;
           until out2;
          out2 := false;
        end;
      if ch = ^c then out1 := true ;
    end;
 until out1;


          o(endfile);
        CLEAR(120,80,GETMAXX-2,100);
     CLEAR(20,MaxY-29 ,MaxX-30,MaxY-1);
     printfont(120,100,#11+#2 +'จบการอ่านข้อมูล ขอขอบคุณที่ใช้บริการ AC LINE MODEM ');
     waittogo;
      end
    else
    begin
    {o(nofile +cr);}
    CLEAR(20,MaxY-29 ,MaxX-30,MaxY-1);
    printfont(120,100,#11+#2 +'NO HAVE FILE  '+FILENAME);
    WAITTOGO;
    end;
end;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Procedure outToconT( var id : word );
begin
 repeat
   0(base^[id]);
   inc(id);
 until (id-1 ) =$0d ;
end;



{ write data form memory to disk }

Procedure writedisk(ix:word;filename : string);
var is : word;
begin
  is := 1;
  assign(filepro,drive+filename);
  rewrite(filepro);
  repeat
    write(Filepro,base^[is]);
    inc(is);
  until  is = iX - 5 ;
  close(filepro);
  freemem(base,maxheep);
end;

{ write data form controller to memory }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Procedure writeMen(var iy : word);
var chr1 : integer; chr2 : char; out : boolean;
begin
  shift := '      ';
  out := false;
  repeat
    recive(ch1);
     shiftdata (ch1[1]);
     chr1 := ord(ch1[1]);    chr2 := chr(chr1);
     base^[iy] := chr2;
        inc (iy);
       if shift = endfile then out := true;
       if keypressed then  begin  ch := readkey ;
         if ch = ^c then out := true; end;
  until  out;
end;

procedure recivefile;
var id,N : word; out1 : boolean;
begin
  id := 1 ;   N := 1;  out1 := false;
  namedata := '            ';
  ch := ' ';
    REPEAT
        recive(ch1) ;
        NAMEDATA[N] := CH1[1];
        INC(N);  if ch1[1] = cr then out1 := true;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
      if keypressed then  begin ch := readkey;
         if  ch = ^c then out1 := true end;;
   UNTIL out1 ;
 if ch <> ^c then begin
        NAMEDATA[N-1] := ' ';
        getmem(base,maxheep);
 printfont(120,100,#11+#2 +'โปรดรอสักครู่ กำลังอ่านข้อมูลเข้า MEMORY    ');
 writemem(id) ;
 clear(120,70,maxx-3,110);
 printfont(120,100,#11+#2 +'โปรดรอสักครู่ กำลัง เก็บข้อมูลลง DISK    ');
 writedisk(id,NAMEDATA);
 clear(2,70,maxx-3,110);
 printfont(120,100,#11+#2 +'จบการอ่านข้อมูล ขอขอบคุณที่ใช้บริการ AC LINE MODEM ');
          CLEAR(20,MaxY-29 ,MaxX-30,MaxY-1);
   printfont(20,MaxY-3,#11+#2 +'                    กด  ANY KEY เพื่อกลับ MENU หลัก ');
 ch := readkey;
 clear(2,70,maxx-3,110);
      CLEAR(20,MaxY-29 ,MaxX-30,MaxY-1);
     putimage(10,40,pic3^,0);
end;
  end;
procedure reciveformCOm;
var buffer : array [1..6] of char ;
    i,J  :  integer   ;
    exit : boolean ;
    order : str30;
    COMM : STRING[123];
    out : boolean;
begin
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    clearpic;
    shift := '    ';
    printfont(20,MaxY-3,#11+#2 +'        ติดต่อขอ File  ' );
    i := 1; J := 1;ch1 := '';
    out := false;exit := false;
 repeat
     while  ((not out) and (not keypressed)) do
      begin
       recive(ch1) ;
       shiftdata(ch1[1]);
       if ch1 = cr then out := true;
      end;
    if keypressed then begin ch := readkey ;
    if ch = ^c then exit := true  end;
    if out   then exit := true;
  until exit;
if shift =  STAR then  begin recivefile; exit := true; end;
{if shift = nofile then directory('*.*',false);}

      CLEAR(20,MaxY-29 ,MaxX-30,MaxY-1);
      putimage(10,40,pic3^,0);
end;



procedure wantfile ;
var y,num,error: integer;
    exit :  boolean; namedata1 : filen;
begin
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    clearpic;  namedata := '                    ';
    Y := 80;
    if graphdriver = 1 then  y :=80  else y := 74 ;
    printfont(20,100,#11+#2 +'FILE  NAME      ');
    CLEAR(20,MaxY-29 ,MaxX-30,MaxY-1);
    printfont(20,MaxY-3,#11+#2 +'<ถ้าต้องการ อ่าน  DIRECTORY ให้กด RETRUE KEY> ');
    SOUND1;
    WRITECHAR(200,Y+20,NAMEDATA1);
    IF NAMEDATA[1] = ' ' THEN DISPLAY;
     clear(2,40,MaxX-2,MaxY-31);



      sendfile(namedata);
       clear(2,40,MaxX-2,MaxY-31);
      CLEAR(20,MaxY-29 ,MaxX-30,MaxY-1);
       putimage(10,40,pic3^,0);
END;



procedure textfile;


begin
  clearpic;
  display;


  IF CH = #13 THEN
BEGIN
  MyExitProc;
  clrscr;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  writeLN('FILE NAME IS  ',namedata);
  writefile(namedata);
  WRITELN;
  WRITELN('~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~');
  WRITELN('               END OF FILE NAME ',NAMEDATA              );
  WRITELN('..............................................................');
  WRITELN('::::::::::::::::::PRESS ANY KEY:::::::::::::::::::::::::::::::::');
  writeln(':::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::');
      ch := readkey;
  Initialize;
  RECTANGLE(0,0,MaxX,MaxY);
  LINE(0,30,MaxX,30);LINE(0,MaxY-30,MaxX,MaxY-30);
  printfont(100,23,#11+#1+'AC LINE MODEM  V 1.0 ');
END;
   clear(20,40,maxx-2,MAXY-31);
   clear(20,MAXY-29,maxx-2,MAXY-1);
  putimage(10,40,pic3^,0);
end;



{--------------------END OF MODEM-----------------}


procedure changdrive;


var out : boolean;
begin
out := true;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
rectangle(205,105,400,125);
printfont(210,120,#11+#3+'เปลี่ยน '+drive +' เป็น ');
repeat
  ch := readkey;
  case ch of
   'a','A' : begin out := true  ;drive := 'a:'  end;
   'b','B' : begin out := true  ;drive := 'b:'  end;
   'c','C' : begin out := true  ;drive := 'c:'  end;
   'd','D' : begin out := true  ;drive := 'd:'  end;
  else
    begin  out := false;sound1 end;
  end;
until out;

clear(205,105,400,125);
clear(xup[1],yup[1],xdown[1],ydown[1]);
printfont(20,120,#11+#3+'3.Drive      '+drive);
moveblock;
end;



procedure active;
begin
if blockmain = 1 then
  case blocknum[1] of
  1 :wantfile;
  2 :reciveformcom;
  3 : changdrive;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  4 : textfile;
  5 : rs232(b1,d,s,p);
  6 : exits := true;
  end
  else
    case blocknum[2] of
  1 : controlac(1);
  2 : controlac(2);
  3 : controlac(3);
  4 : controlac(4);

    end;
end;



procedure  testkey(ch : char);
begin
   if funckey then
      case  ch of
   { up } #72 :  moveUp;
   {down} #80 :  moveDown;
   { -> } #77 :  moveR;
   { <- } #75 :  moveL;
      end
   else
     if  ch = #13   then    active;
     if ch  = #27 then exits := true;
end;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
procedure SelecBlock;
var ch : char;
   begin
     getsize;
     getimage(10,40,200,190,pic4^);        clear(10,40,200,190);
     blockmain := 1 ;
     block1;
     first := true;
     BlockNum[BlockMain] := 1;
     blocknum[2] := 1;
     blocknum[3] := 1;
     moveblock;
repeat
    exits := false;
  repeat
     readfunckey (ch);
     testkey(ch) ;
  until exits;
  CLEAR(240,MaxY-29 ,MaxX-30,MaxY-1);
  printfont(240,MaxY-3,#11+#2 +'< You sure to exit of program Y or N > ');
repeat
  ch := readkey;
  case ch of
  'y','Y' : endofproject := true;
  'n','N' : endofproject := false;
  else  begin sound1   ; ch := cr end;
  end;
until ch <> cr ;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
CLEAR(240,MaxY-29 ,MaxX-30,MaxY-1);
until endofproject;
end;



begin
     drive := 'a:';
     l := 1; row := 1;
     dirnum := 1;dis1 :=1;
     xdown[1] := 195 ;
     ydown[1] := 82;
     xup[1]   := 15;
     yup[1]   := 68;
     xdown[2] := 405 ;
     ydown[2] := 82;
     xup[2]   := 215;
     yup[2]   := 68;
     xdown[3] := 120;
     ydown[3] := 60;
     xup[3]   := 15;
     yup[3]   := 45;
            X := MaxX DIV 4;
     {init rs_232}
      baudbit := 'f';  brat := '9600';
      databit := '8';  bitd :='8';
      stopbit := '2';  sbit := '2';
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
      paritybit := '2'; pbit := 'even';
       comnum   := '1';   b1:= 8;d := 2 ;s := 3;p := 2;
      {inti_colse control ac line}
       for i := 1 to 9 do BEGIN    PIVNUM[I] := ON;
      system1[i] := off;            END;
  init_com(comnum,baudbit,databit,paritybit,stopbit);
   Initialize;
   RECTANGLE(0,0,MaxX,MaxY);
LINE(0,30,MaxX,30);LINE(0,MaxY-30,MaxX,MaxY-30);

textshow(4,3,'h');
outtextxy(20,90,'    WELCOME TO AC LINE MODEM AND CONTROL');
sound1;
textshow(0,1,'h');

 savefont;
  clear(20,90,maxx-2,120);
printfont(100,23,#11+#1+'AC LINE MODEM  V 1.0 ');
if graphdriver = 1 then  bang :=  138  else bang := 170 ;
RECTANGLE(120,50,MaxX-120,MaxY-40);
printfont(bang,80,#11+#1+'  สถาบันเทคโนโลยี');
printfont(bang,110,#11+#1+'   พระจอมเกล้าฯ ');
printfont(bang,140,#11+#1+'    ลาดกระบัง ');
delay(500);
CLEAR(121,51,MaxX-121,MaxY-41);
printfont(bang-50,80,#11+#3+'     จัดทำโดยนักศึกษา ภาคเทคนิคอุตสาหกรรม   ');
 printfont(bang,100,#11+#3+' 1.  นาย  ภานุพันธ์   งานเจริญกุล  ');
printfont(bang,120,#11+#3+' 2.  นาย  ณรงค์ศักดิ์  ตันติวิริยะพันธุ์  ');
printfont(bang,140,#11+#3+' 3.  นาย  สมชาย    เกิดแก้ว ');
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
sound1;


waittogo;
CLEAR(120,50,MaxX-120,MaxY-40);
endofproject := false;
selecblock;



CLOSEGRAPH;


clrscr;  textcolor(0); textBackground(7);
 writeln('                                                          ');
 writeln('            GOOD BYE  FROM  AC LINE MODEM AND CONTROL     ');
 writeln('                                                          ');
 textcolor (3) ;; textBackground(0);
 WRITELN;
 WRITELN;
end.
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ข.

หนังสืออ้างอิง

1. ชัชวาล ยนต์หงค์ ,แนะนำภาษา Pascal โดย Turbo Pascal ,2532
2. น.ต.ดร.ไพศาล สงวนหมู่, ร.ศ ยืน ภู่วรวรรณ , การสื่อสารข้อมูลและไมโครคอมพิวเตอร์ , 2531
3. เอกสิทธิ์ ฐิติวรการ ,"เทคโนโลยีการสื่อสารข้อมูล",คอมพิวเตอร์อีเล็คทรอนิคส์เวิลด์,2532
4. HERBERT TAUB , DONALD L. SCHILLING , PRINCIPLES OF COMMUNICATION SYSTEMS ,1987
5. R.C.DIXON ,SPREAD SPECTRUM SYSTEMS ,1976

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

ปริญญานิพนธ์นี้ได้สำเร็จลุล่วงไปด้วยดี เนื่องจากได้รับความอนุเคราะห์จากอาจารย์ กฤดากร กล่อมการ ที่ให้คำปรึกษาแนะนำ ตลอดการทำงาน พร้อมทั้งเพื่อนๆ ที่ให้ความช่วยเหลือ อันเป็นประโยชน์แก่คณะผู้จัดทำ จึงขอบพระคุณอย่างสูงมา ณ ที่นี้

นาย ณรงค์ศักดิ์ ตันติวิริยพันธุ์
นาย ภานุพันธ์ งานเจริญกุล
นาย สมชาย เกิดแก้ว

นักศึกษาภาควิชาเทคนิคอุตสาหกรรม
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้า
เจ้าคุณทหาร ลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้