ระบบเตือนภัย

(ALARM SYSTEM)

โดย

นาย นิติพงษ์ ล่องแสงศรี

นาย ประพจน์ พลอยสุวรรณ

นาย ประวิทย์ เปลี่ยนสกุลวงศ์

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตร์บัณฑิต

สาขาวิศวกรรมศาสตร์อิเล็กทรอนิกส์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2537

ปริญญานิพนธ์ปีการศึกษา 2537

ภาควิชา อิเลคทรอนิกส์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง ระบบเตือนภัย

ผู้จัดทำ

1.นายนิติพงษ์ ส่องแสงศรี เลขประจำตัว 35103186

2.นายประพจน์ พลอยสุวรรณ เลขประจำตัว 35103188

3.ประวิทย์ เปลี่ยนสกุลวงศ์ เลขประจำตัว 35103189

ดร. กิตินล ชิดสกุล อาจารย์ที่ปรึกษา

(.....)

สารบัญ

ระบบเตือนภัย

นาย นิติพงษ์ ลองแลงศรี

นาย ประพจน์ พลอยสุวรรณ

นาย ประวิทย์ เปลี่ยนลกุลวงศ์

อ. กิตินล ชัดสกุล อาจารย์ที่ปรึกษา

ปีการศึกษา 2537

บทคัดย่อ

จุดประสงค์ของโครงงานคือได้ทำการพัฒนาตัวต้นแบบระบบตรวจจับสัญญาณเตือนภัยต่างๆที่ใช้สำหรับอาคารสูงหรือสถานที่ซึ่งมีขนาดใหญ่ ที่ต้องการติดตั้งระบบตรวจจับสัญญาณเตือนภัยต่างๆ แล้วส่งสัญญาแมาที่ห้องควบคุมเพื่อแสดงผล โดยไม่จำเป็นต้องมีการเดินสายสัญญาณระหว่างห้องควบคุมและจุดตรวจจับ โดยจะใช้การส่งสัญญาณผ่านสายA.C.LINE แทน เพราะในทุกๆส่วนของอาคารจะมีสาย A.C. LINE เดินอยู่แล้ว ข้อมูลที่จะถูกส่งลง A.C. LINE จะถูก MODULATE แบบ A.M.(AMPLITUDE MODULATION) ก่อน ความเร็วของข้อมูลที่ใช้ในการส่ง-รับประมาณ 300 BAUD ซึ่งเหมาะสำหรับการส่ง-รับที่มีขนาดของข้อมูลไม่มากนักและไม่ต้องการความเร็วสูง ด้วยวิธีการนี้จะทำให้การติดตั้งสะดวกรวดเร็วขึ้นและค่าใช้จ่ายในการเดินสายก็จะลดลงตามไปด้วย

ABSTRACT

The purpose of this project are design and construction of prototype of alarm system for high building , grand place without to additional signal cables. The principle is based on high frequency modulation signal transmitted through the existing ac. line.

The data transfers between sensing modules and controller is modulated in A.M.(Amplitude Modulation) and transmitted with 300 bauds. This format of transmission is low speed and low dense data transceived.

The above method provides no complexity of installation and decrease the cost effectiveness for wiring cables.

กิตติกรรมประกาศ

ปริญญานิพนธ์เรื่องนี้ได้แนวความคิดมาจากปริญญานิพนธ์เรื่องระบบป้องกันการโจรกรรม ปีการศึกษา 2529 ซึ่งได้เล็งเห็นข้อดีของระบบป้องกันการโจรกรรมนี้ ดังนั้นจึงได้นำมาพัฒนาระบบให้มีประสิทธิภาพดีขึ้น โดยได้รับคำแนะนำต่างๆ จาก ดร. กิตินล ชิดสกุล มาโดยตลอด จึงขอกราบขอบพระคุณ ดร. กิตินล ชิดสกุล เป็นอย่างสูง ขณะที่ทำปริญญานิพนธ์เรื่องนี้ยังได้รับความช่วยเหลือจากบุคคลหลายๆท่านโดยเฉพาะ

คุณสุมาลี คำลือ

คุณจำลอง พิมพ์สวัสดิ์

คุณจรินทร์ ภูมิดิษฐ์

คุณพิมพ์พิชญ์สร แง้มแก้ว

ท้ายสุดขอขอบคุณ คุณวรกิจ แต้มทอง และ คุณจิรสิญจน์ กระเช้าเพ็ชร ที่ได้ช่วยพิมพ์และตรวจทานปริญญานิพนธ์ฉบับนี้จนเสร็จเรียบร้อย

บทที่ 1

บทนำ (INTRODUCTION)

สังคมปัจจุบัน ทุกคนอยู่ด้วยความหวาดผวากับภัยหลายๆอย่างที่อาจเกิดขึ้นได้ไม่ว่าเวลาใด เช่น อัคคีภัย , การโจรกรรม , ภัยจากธรรมชาติต่างๆมากมาย แต่ภัยที่เกิดขึ้นบ่อยที่สุดและระบบทำความเสียหายให้กับทรัพย์สิน ได้แก่ ภัยจากการโจรกรรม และ อัคคีภัย ดังนั้น ระบบเตือนภัยแบบต่างๆจึงได้ถูกพัฒนาขึ้นมาขายมากมายตามท้องตลาด ปกติทั่วๆไปที่พบเห็นการใช้งานไม่กว้างขวางคือ สามารถป้องกันได้เพียงอย่างใดอย่างหนึ่งเท่านั้น เช่น ป้องกันการโจรกรรม หรือป้องกันอัคคีภัย

ในภาวะปัจจุบันที่อยู่อาศัยส่วนใหญ่จะเป็นห้องชุด , แฟลต หรือ คอนโดมิเนียม ซึ่งจะมีผู้คนเข้าออกพลุกพล่านทำให้ต้องมีการระวังภัยต่างๆที่อาจจะเกิดขึ้นได้มากเป็นเงาตามตัว

ดังนั้น ปริญญานิพนธ์ฉบับนี้จึงได้เสนอ การสร้าง ระบบเตือนภัย ขึ้นมาใช้งานโดยมีจุดประสงค์และคุณสมบัติ ดังต่อไปนี้

1.1 จุดประสงค์

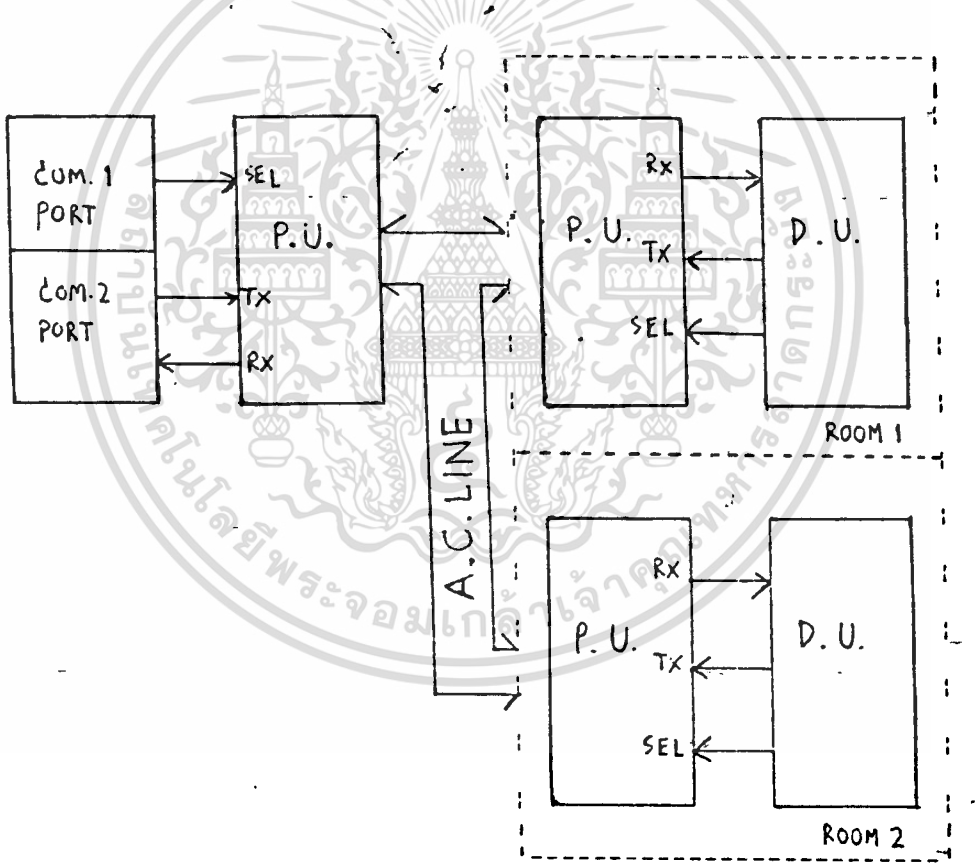
1. เพื่อศึกษาเกี่ยวกับการวางระบบเตือนภัยโดยการส่ง-รับสัญญาณผ่าน A.C.LINE
2. เพื่อศึกษาถึงวิธีการนำเอา MICROCONTROLLER ตระกูล MCS.51 มาใช้งาน
3. เพื่อศึกษาเกี่ยวกับการควบคุมระบบเตือนภัยด้วยการใช้ COMPUTER

1.2 คุณสมบัติของระบบเตือนภัย

1. สามารถตรวจจับสัญญาณเตือนภัยต่างๆภายในห้องได้ 8 จุด จำนวน 256 ห้อง
2. แสดงผลการเตือนภัยออกทางหน้าจอ COMPUTER พร้อมทั้งใช้ควบคุมระบบทั้งหมด
3. ส่งผ่านสัญญาณผ่าน A.C.LINE ทำให้ไม่ต้องเดินสายสัญญาณเพิ่มขึ้นอีก

1.3 โครงสร้างของระบบเตือนภัย

จากแนวคิดในการนำเอา COMPUTER และ MICROCONTROLLER มาควบคุมระบบผ่านทาง A.C.LINE เราสามารถแสดงการทำงานของระบบได้โดย BLOCK DIAGRAM ดังต่อไปนี้



รูปที่ 1.1 แสดงบล็อคไดอะแกรมของระบบเตือนภัย

จาก BLOCK DIAGRAM สามารถแบ่งออกได้เป็น 3 ส่วนใหญ่ๆ คือ

1. CONTROL UNIT (C.U.)
2. P.L.C.T. UNIT (P.U. OR POWER LINE CARRIER TRANSCEIVERS UNIT)
3. DETECT UNIT (D.U.)

อธิบายได้ว่า C.U. จะใช้คอมพิวเตอร์เป็นตัวส่งข้อมูลเลือกห้องออกไปยังพอร์ทคอม 1 และพอร์ทคอม 2 ซึ่งต่ออยู่กับ P.U. โดย P.U. ทำหน้าที่จัดการส่งข้อมูลผ่าน A.C.LINE หลังจากนั้นก็จะคอยรับข้อมูลจาก A.C.LINE ข้อมูลที่อยู่ใน A.C.LINE จะถูกส่งเข้าไปยังห้องแต่ละห้อง ซึ่งมี D.U.ทำหน้าที่ DECODE หมายเลขห้องว่า C.U. ต้องการติดต่อกับห้องใดและทำการตรวจจับสัญญาณจากตัว SENSOR ต่างๆที่ใช้ในการเตือนภัยด้วย

เมื่อ D.U. ทำการ DECODE หมายเลขห้องได้แล้ว จากนั้นก็จะทำการส่งข้อมูลที่ประกอบไปด้วยหมายเลขห้อง , ข้อมูลที่ใช้ในการเตือนภัย กลับไปยัง C.U. เพื่อที่จะได้ใช้ในการแสดงผลได้ว่า ห้องหมายเลขใดกำลังมีเหตุร้ายอะไรเกิดขึ้น ต่อจากนั้น C.U. ก็จะทำการส่งข้อมูลเพื่อใช้ในการเลือกห้องใหม่ต่อไปจนกว่าจะครบ 256 ห้องแล้วก็จะวน LOOP มาที่ห้องแรกใหม่เป็นลักษณะเช่นนี้เรื่อยไป

บทที่ 2.

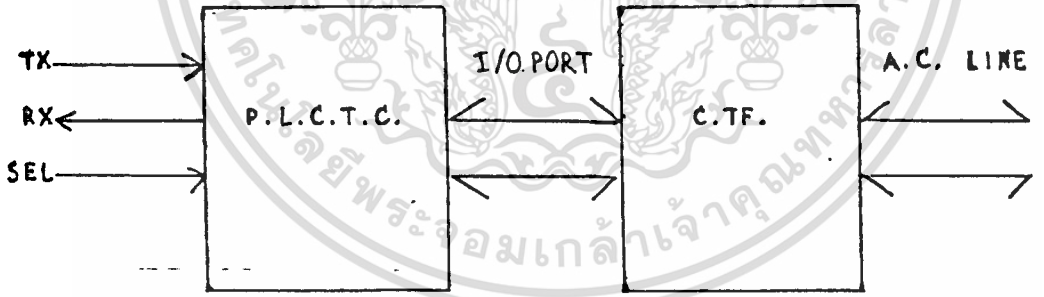
POWER LINE CARRIER TRANSCEIVERS

(P.L.C.T. CIRCUITS)

วงจร P.L.C.T. มีหน้าที่ที่สำคัญคือ เป็นตัวจัดการในการส่งและรับข้อมูลแบบอนุกรมผ่านเข้าไปยัง A.C. LINE 220V 50Hz ซึ่งมีหลักการทำงานประกอบด้วยส่วนใหญ่ๆอยู่ 2 ส่วนคือ

1. P.L.C.T.C.(POWER LINE CARRIER TRANSCERVERS CONTROLLER)
2. C.TF. (COUPLING TRANSFORMER)

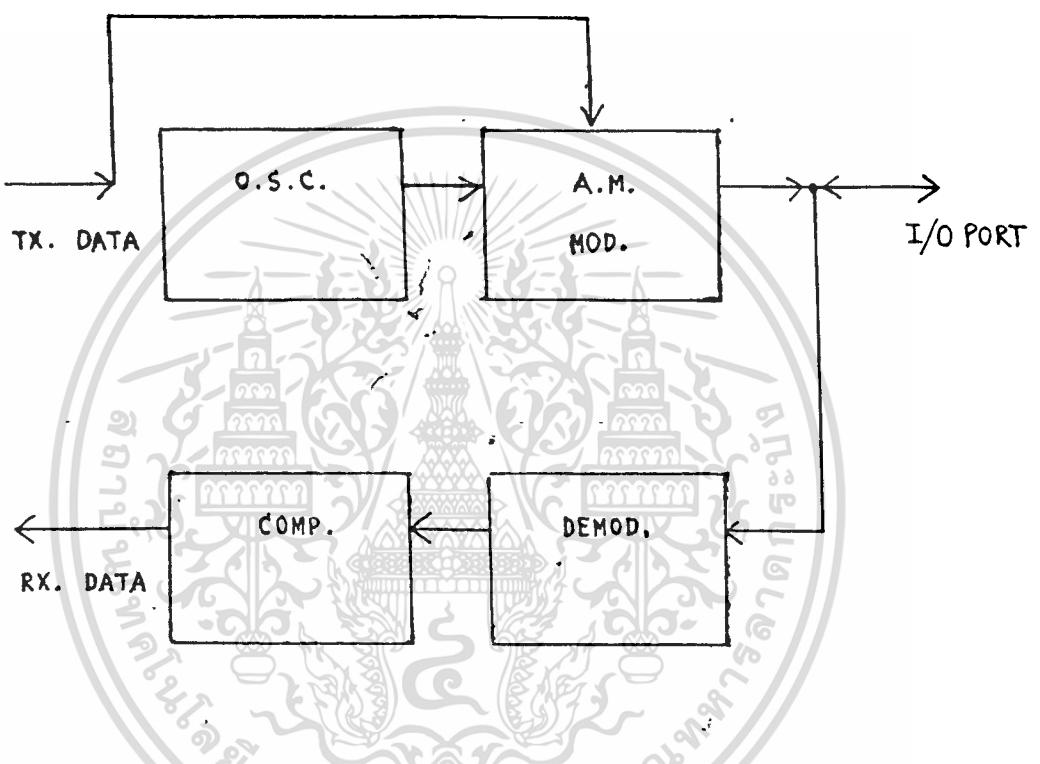
ดังแสดงได้ดัง BLOCK DIAGRAM ต่อไปนี้



รูปที่ 2.1 แสดงหลักการทำงานของวงจร P.L.C.T.

- P.L.C.T.C.

หน้าที่ของส่วน P.L.C.T.C. นี้คือ เป็นตัว MODULATOR & DEMODULATOR ของ SEARIAL DATA ในแบบ HALF DUPLEX สามารถแสดงการทำงานของระบบได้โดย BLOCK DIAGRAM ต่อไปนี้



รูปที่ 2.2 แสดงหลักการทำงานในส่วนของ P.L.C.T.C.

รูปที่ 2.2 อธิบายได้ว่า ข้อมูลที่ต้องการส่งผ่าน A.C. LINE (TX DATA) จะถูกส่งเข้ามายังภาค MODULATOR พร้อมกับสัญญาณพาหะที่มีความถี่ประมาณ 100 KHz ขึ้นไปที่สร้างจากภาค OSCILLATORเพื่อทำการ MODULATION แบบ AM แล้วส่งออกไปยัง I/O PORT

ขณะที่ต้องการรับข้อมูลที่ผ่าน A.C. LINE (RX DATA) ที่สาย I/O PORT ข้อมูลที่ถูก MODULATE มาแล้วนั้นจะถูกส่งผ่านมาเข้ายังภาค DEMODULATOR เพื่อที่จะ DECODE สัญญาณที่ถูก MODULATE ออกมาเป็นข้อมูลที่ต้องการ เพื่อให้ได้ข้อมูลที่ถูกต้องแน่นอน ดังนั้นข้อมูลที่ถูก DECODE แล้ว จะผ่านมาเข้าภาค COMP(COMPARATOR) ซึ่งทำหน้าที่ในการเปรียบเทียบข้อมูล เพื่อที่จะได้กำจัด NOISE ที่ปะปนมาออกไป ดังนั้นก็จะได้ข้อมูลที่ต้องการออกมา

ขั้นตอนที่จะต้องทำต่อไป คือ นำเอาหลักการดังกล่าวมาทำการศึกษา,ทดลองและออกแบบสร้างวงจรต่อไปเพื่อที่จะให้ได้วงจรที่สามารถใช้งานได้ตามความต้องการ จากการศึกษาและทดลองมาแล้วพบว่าอาจจะสร้างวงจรใช้งานได้ใน 2 วิธีคือ

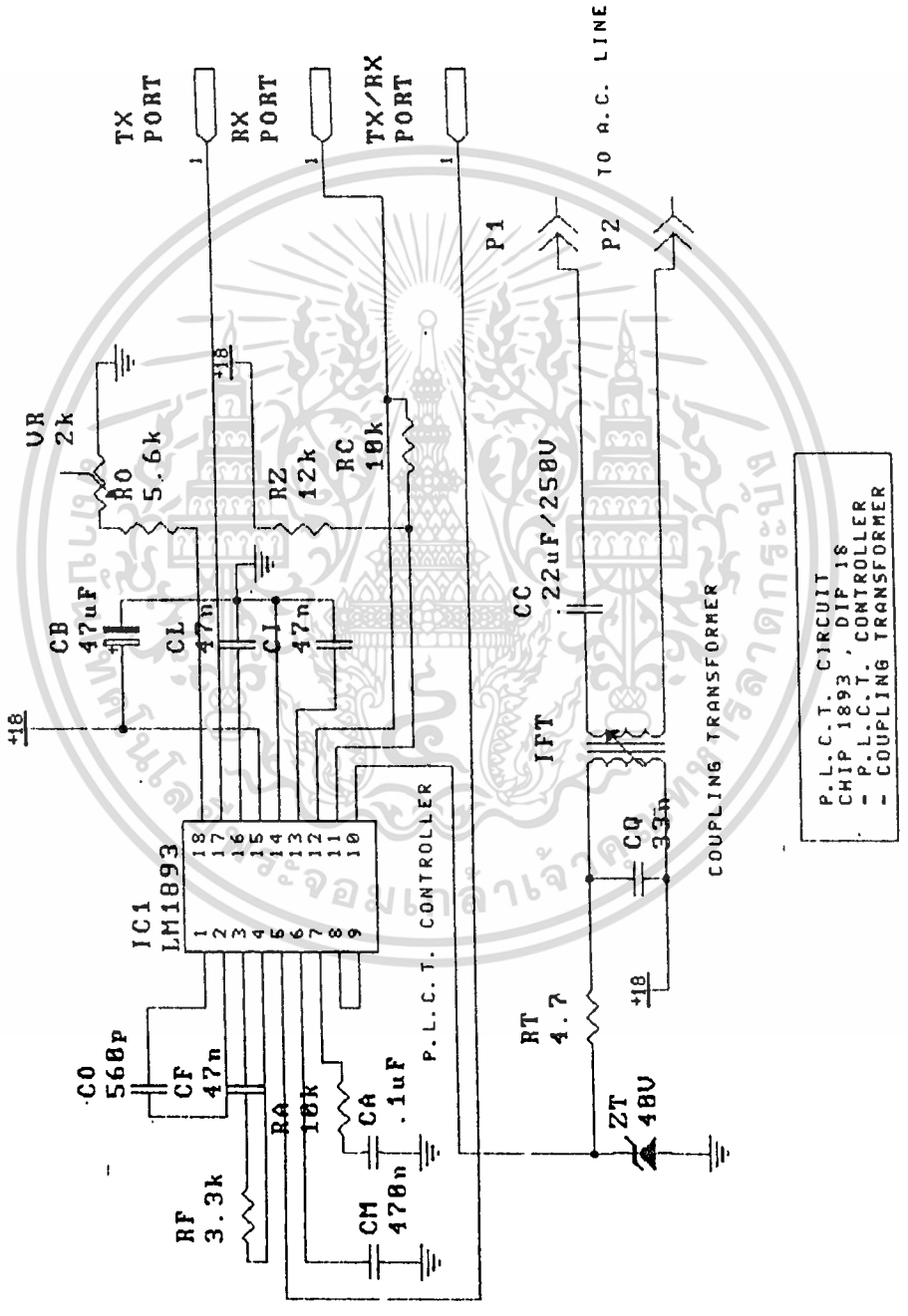
(1) UNUSED I.C. SPECIAL PURPOSE

หลักการคือ นำเอา BLOCK DIAGRAM ใน FIG.2 แต่ละบล็อคมาออกแบบสร้างวงจรไปทีละบล็อค ซึ่งพบว่าเกิดปัญหาขึ้นมาคือ วงจรรวมทั้งหมดมีขนาดใหญ่ ทำให้ต้องใช้อุปกรณ์มาก ขั้นตอนในการออกแบบสร้างวงจรค่อนข้างยุ่งยาก ค่าใช้จ่ายก็สูงตามไปด้วย

(2) USED I.C. SPECTIAL PURPOSE เบอร์ 1893 พบว่ามีข้อดีเหนือกว่าหลายประการด้วยกันคือ วงจรรวมทั้งหมดมีขนาดเล็กลงกว่าเดิมมาก นั่นคือ ใช้อุปกรณ์น้อยลงทำให้ออกแบบวงจรได้ง่ายและสะดวกขึ้นและค่าใช้จ่ายก็ลดลงด้วย

จากเหตุผลดังกล่าวข้างต้น จึงได้เลือกใช้ I.C. SPECTIAL PURPOSE ในการ DESIGN วงจร P.L.C.T. ทั้งหมดดังมีรายละเอียดที่จะได้กล่าวถึงดังต่อไปนี้

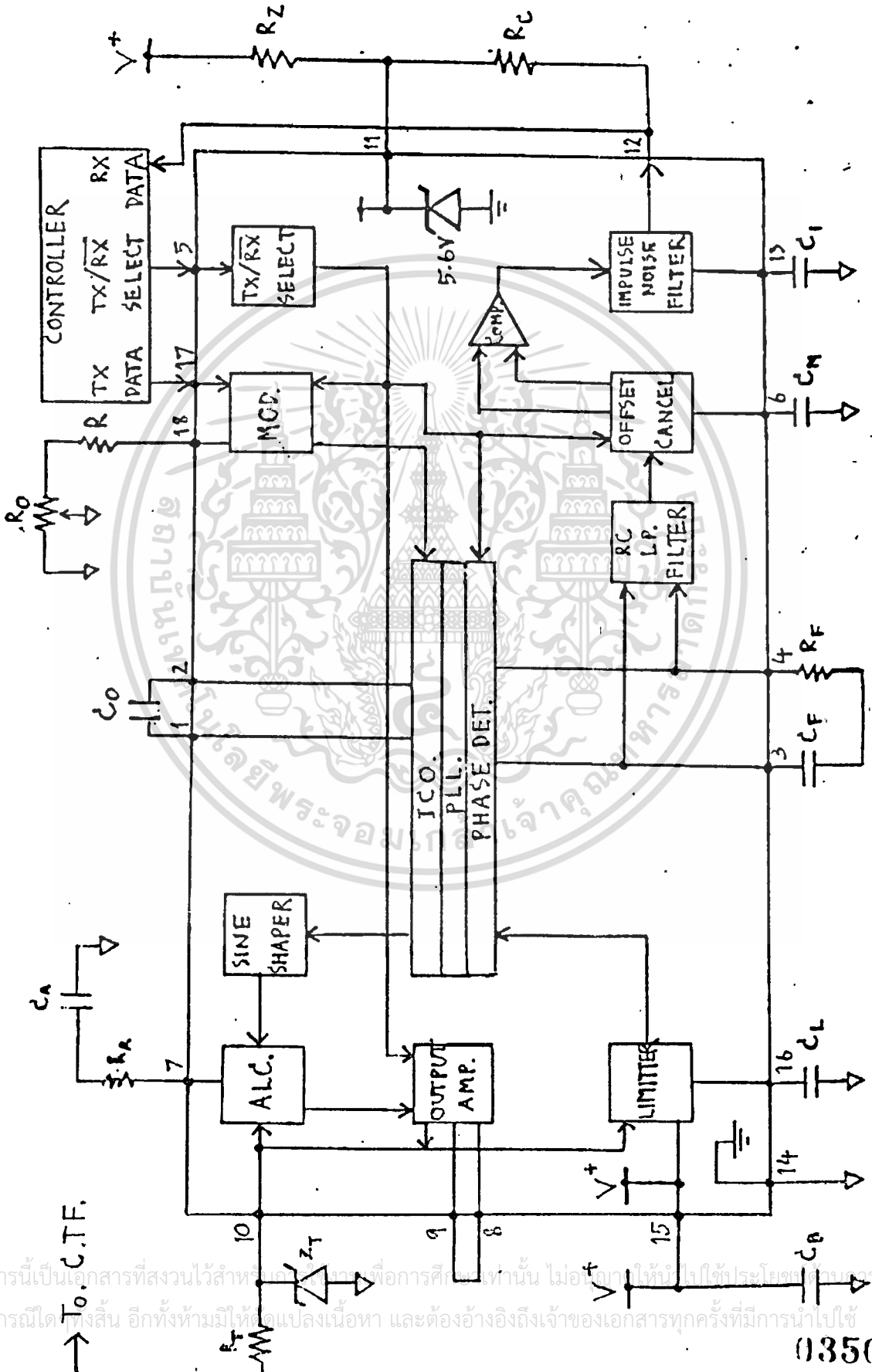
รูปที่ 2.3 แสดงวงจรสมบูรณ์ของส่วน P.L.C.T.C.



ในส่วนของวงจร P.L.C.T. CONTROLLER นี้จะใช้ I.C.เบอร์ 1893 เป็นตัวหลักในการทำหน้าที่ที่สำคัญคือ ส่ง-รับข้อมูลแบบอนุกรมผ่าน A.C. LINE โดยที่ TX DATA จะป้อนเข้าที่ขา 17, RX DATA จะผ่านออกมาที่ขา 12 โดยที่ TX DATA จะถูก MODULATION ในระบบ AM (AMPLITUDE MODULATION) มี CARRIER เป็น SINEWAVE ประมาณ 125 KHz โดยที่ความถี่ของ CARRIER ถูกกำหนดได้จากค่าของ CO และ RO ปรับ ค่าได้ที่ต่อกับขา 1-2 และ 18 ตามลำดับ ซึ่งภายใน I.C.จะมีวงจร MOD&OSCILLATOR ให้ไว้อยู่แล้ว, ควบคุมการส่ง-รับข้อมูลได้ที่ขา 5 ว่าต้องการให้ I.C. ทำงานอยู่ใน MODE TX หรือ RX ซึ่งเป็นการทำงานแบบ HALF DUPLEX นั่นคือ การส่งและรับข้อมูลสามารถทำได้คนละเวลากัน, ข้อมูลจะถูกส่ง-รับกับ PORT ที่ขา 10 เพื่อส่งข้อมูลหรือรับข้อมูลผ่านทาง A.C. LINEได้, ที่ขา 16 จะต่อ CL เพื่อกำจัดความถี่ 50 Hz ของ LINE A.C., ขา 3 และ 4 ต่ออยู่กับ RF,CF ทำหน้าที่ในการกำหนด POLE AND ZERO ให้กับ วงจร PLL LOOP FILTER ภายในตัว I.C., ที่ขา 6 CM จะถูกใช้สำหรับ HOLD RX DATA เอาไว้, CI ที่ขา 13 ใช้สำหรับกำจัด IMPULSE NOISE ของสัญญาณเอ้าท์พุท, ส่วน RA และ CA จะกำหนด POLE AND ZERO ให้กับวงจร ALC ใน I.C., ขา 10 จะต่อ ZT และ RT เพื่อป้องกัน TRANSIENT จาก A.C. LINE ไม่ให้มีค่ามากเกินไปที่ I.C.จะรับได้, RC ที่ขา 12 ทำหน้าที่ เป็น COLLECTOR PULL-UP RESISTOR เพื่อจ่ายกระแสเข้าไป DRIVE วงจรภายใน I.C. ได้อย่างเพียงพอ, RZ จะต่ออยู่ที่ขา 11 ทำหน้าที่ในการจ่ายกระแสเข้าไปเลี้ยง ZENER DIODE เพื่อรักษาระดับแรงดันของเอ้าท์พุทให้มีค่าคงที่ที่ 5V

ในส่วนของวงจร COUPLING TRANSFORMER จะประกอบไปด้วย I.F.T., CC , CQ ซึ่งทำหน้าที่ในการ TUNE ความถี่ CARRIER ที่ต้องการ และใช้ในการลดทอนสัญญาณความถี่ต่างๆ ใน LINE A.C. ให้มีขนาดลดลงด้วย

จากวงจรรูปที่ 2.3 สามารถแสดงให้เห็นถึง PATH การทำงานใน MODE ของการส่ง-รับ ข้อมูลได้ดังต่อไปนี้



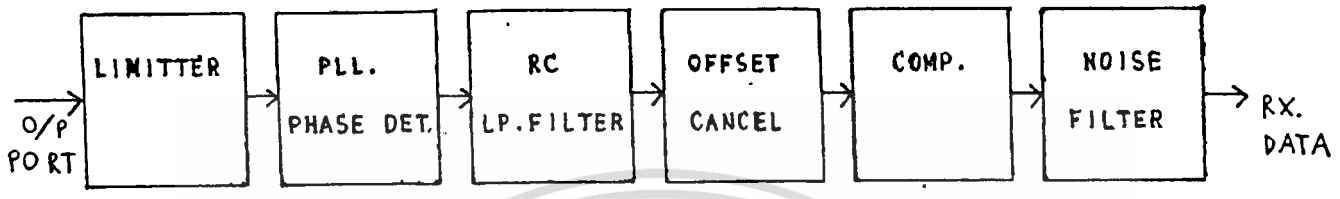
รูปที่ 2.4 Block diagram of a P.L.C.T. system

MODE TX : เป็นโหมดที่ I.C. ใช้ในการทำหน้าที่จัดการส่งข้อมูลผ่าน A.C. LINE ซึ่งมี PATH ในการทำงานดังต่อไปนี้

รูปที่ 2.5 แสดงถึงทางเดินของการส่งข้อมูลผ่านเอซิไลน์

จากรูปที่ 2.5TX DATA ซึ่งมีลักษณะเป็น SEARIAL DIGITAL DATA มีอัตรา 300 BAUD จะผ่านเข้าไปยังวงจร MOD เพื่อที่จะควบคุมให้วงจร ICO.(INTEGRATE CURRENT-CONTROLLED OSCILLATOR)ผลิตความถี่CARRIER TRIANGULAR WAVE ที่แอมปลิจูดของ TRI-WAVE เปลี่ยนไปอย่างได้สัดส่วนกันกับการเปลี่ยนแปลงของแอมปลิจูด TX DATA, MODULATE DATA ที่ได้นี้จะถูกส่งไปเข้าวงจร SINE SHAPER ซึ่งจะนำเอา CARRIER ของ MODULATE DATA มาทำการตบแต่งให้เป็น SINE WAVE เพื่อลดผลทางด้าน HARMONICS ก่อนที่จะส่งต่อไปยังวงจร ALC.(AUTOMATIC LEVEL CONTROL) ทำการควบคุมระดับของสัญญาณให้เหมาะสมก่อนที่จะใช้ในการ DRIVE วงจร OUTPUT AMP. ต่อไป เอ้าท์พุทที่ได้ขณะนี้จะมี GAIN สูงขึ้นโดยประมาณ 200 เท่าที่ขา OUTPUT PORT

MODE RX : เป็นโหมดที่ I.C. ใช้ในการจัดการเกี่ยวกับการรับข้อมูลผ่านทาง A.C. LINE ซึ่งมี PATH การทำงานดังแสดงต่อไปนี้



รูปที่ 2.6 แสดงถึงทางเดินของการรับข้อมูลผ่านเอซีไลน์

รูปที่ 2.6 อธิบายได้ว่า RX DATA ซึ่งมีลักษณะคือ MODULATE DATA ที่เดินทางผ่าน A.C.LINEแล้วบวกกับ NOISE ซึ่งจะปะปนเข้ามากับ DATA ด้วยจะถูกส่งผ่านมาเข้ายังวงจร LIMITTER ก่อนเพื่อที่จะลดทอนความถี่ของ A.C.LINE, กำจัด D.C.OFFSET และจำกัดแอมปลิจูดของ DATA ที่จะเข้าไป DRIVE วงจร PLL.(PHASE LOCKED LOOP),PHASE DETECTOR ซึ่งจะทำหน้าที่ในการ DEMODULATE สัญญาณที่ผ่านเข้าไปออกมาเป็น DEMODULATED DATA ที่ต้องการ เอ้าท์พุทที่ได้นี้จะประกอบไปด้วย A.C. AND D.C. DATA SIGNAL, NOISE, SYSTEM DC OFFSOETS และ TWICE-THE-CARRIER FREQUENCY ซึ่งสัญญาณเหล่านี้จะผ่านเข้าไปยังวงจร 3-STATE RC LOWPASS FILTER เพื่อกำจัด NOISE และความถี่ที่ไม่ต้องการออกไป หลังจากนั้น DATA จะไหลผ่านวงจร OFFSET CANCEL ซึ่งใช้ทำหน้าที่ยกเลิก DC OFFSET ต่อจากนั้น DATA จะถูกนำมาเปรียบเทียบข้อมูลเพื่อความถูกต้องในวงจร COMPARATOR ต่อไป ในการ COMPARE ข้อมูลนั้น ที่เอ้าท์พุทจะมีการ SWICHING เกิดขึ้นทำให้เกิด IMPULSE NOISE ขึ้นได้ ดังนั้นเพื่อที่จะทำการกรองเอา IMPULSE NOISE ทิ้งไป DATA จะผ่านมาเข้าวงจร IMPULSE NOISE FILTER ได้เอ้าท์พุทออกมาเป็น RX DATA ที่ต้องการนำไปใช้งานนั่นเอง

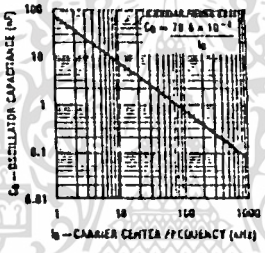
การเลือกค่าของอุปกรณ์ :

THE TRANSMITTER

- C_o

$$C_o = [\ 70.6 * 10^{-6}\ ] / F_o$$

เมื่อ F_o = ความถี่ CARRIER (100 KHz - 300 KHz)



รูปที่ 2.7 กราฟความสัมพันธ์ของ F_o กับ C_o

จากสูตร เมื่อที่จะหาค่าของ C_o จะต้องกำหนดความถี่พาหะของระบบที่ใช้ในการส่ง-รับข้อมูลขึ้นมาก่อน ในที่นี้ SET ให้เท่ากับ 125 KHz ดังนั้นจะได้

$$C_o = \frac{70.6 * 10^{-6}}{125 * 10^{3}} = 560\ pF$$

- C_A AND R_A

C_A = 0.1 uF [ควรมีค่ามากกว่าหรือเท่ากับ 100 pF]

R_A = 10 K

- Z_T

BREAKDOWN VOLTAGE	มากกว่าหรือเท่ากับ 44V น้อยกว่า 60V_{peak} @ 1mA
MAXIMUM LEAKAGE	1uA @ 40V
CAPACITANCE	300pF @ 8V
MAXIMUM CLAMP VOLTAGE	64.5V @ 7.8A
PEAK NON-REPETITIVE PULSE POWER	10KW FOR 1uS
SURGE CURRENT	70A FOR 1/120s

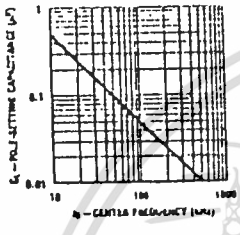
- R_T

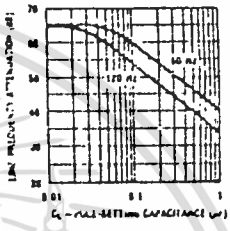
ควรใช้ค่าที่อยู่ระหว่าง 4.7 - 100 โอห์ม

THE RECEIVER

- C_L

ค่าของ C_L สามารถหาได้จากรูปกราฟต่อไปนี้



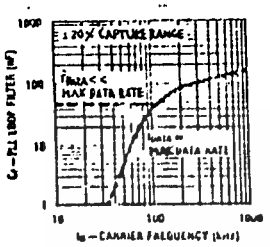


รูปที่ 2.8.1 กราฟความสัมพันธ์ของ F_o กับ C_L รูปที่ 2.8.2 กราฟความสัมพันธ์ของ C_L กับ LOSS

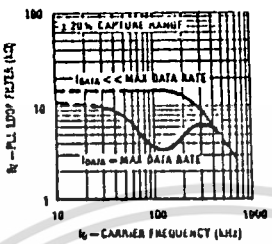
จาก รูปที่ 2.8.1 เมื่อ SET ความถี่พาหะของระบบเอาไว้ที่ 125 KHz, ก็จะได้ค่า C_L ประมาณ 47 nF และจาก รูปที่ 2.8.2 เมื่อทราบค่า C_L แล้วก็นำมาหาค่าของ LINE FREQUENCY ATTENUATETION ได้ประมาณ 70 dB

- C_F AND R_F

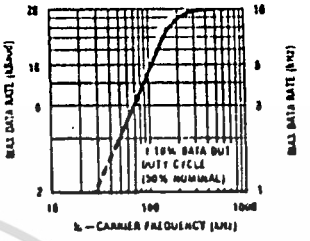
ค่าของ C_F AND R_F สามารถหาได้จากรูปกราฟดังต่อไปนี้



รูปที่ 2.9.1 กราฟความสัมพันธ์ของ F_o กับ C_F



รูปที่ 2.9.2 กราฟความสัมพันธ์ของ F_o กับ R_F

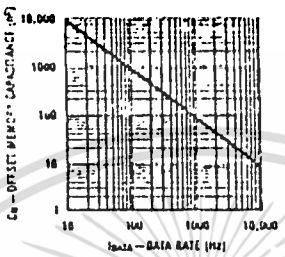


รูปที่ 2.9.3 กราฟความสัมพันธ์ของ F_o กับ DATA RATE

รูปที่ 2.9.3 ที่ความถี่พาหะเท่ากับ 100 KHz จะได้ MAXIMUM DATA RATE ประมาณเท่ากับ 10 KBAUD (5 KHz) นำค่า MAX DATA RATE นี้ไปหาค่า C_F จาก รูปที่ 2.9.1 ได้ประมาณเท่ากับ 47 nF ที่ F_{DATA} << MAX DATA RATE และ F_{DATA} = MAX DATA RATE ส่วน R_F ก็สามารถหาได้ รูปที่ 2.9.2 โดยที่ F_{DATA} << MAX DATA RATE : R_F = 3.3 K และที่ F_{DATA} = MAX DATA RATE : R_F = 20 K

- C_M

ค่าของ C_M หาได้จากรูปกราฟดังต่อไปนี้

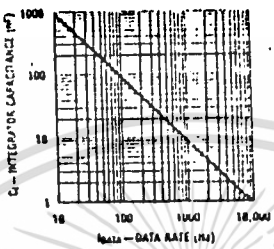


รูปที่ 2.10 กราฟความสัมพันธ์ของ DATA RATE กับ C_M

จาก รูปที่ 2.10 เมื่อ SET DATA RATE เอาไว้ที่ 150 Hz จะได้ค่าของ C_M ประมาณ เท่ากับ 470 nF

- C_1

สามารถหาค่าของ C_1 ได้จากรูปกราฟดังต่อไปนี้



รูปที่ 2.11 กราฟความสัมพันธ์ของ DATA RATE กับ C_1

จากรูปที่ 2.11 เมื่อได้กำหนด F_{DATA} (DATA RATE) ได้แล้วคือ 150 Hz ดังนั้นจะทำให้ทราบค่าของ C_1 ประมาณเท่ากับ 47 nF

- R_z

ค่าของ R_z หาได้จากสมการดังต่อไปนี้คือ

$$R_z = \frac{[\, V_{IN(MIN)} - V_{Z(MIN)} \,]}{[\, I_{Z(MIN)} + I_{L(MAX)} \,]}$$

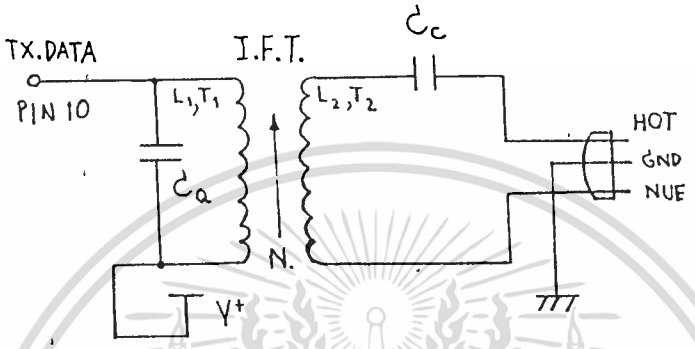
- R_c

ค่าของ R_c สามารถหาได้จากสมการต่อไปนี้คือ

$$R_c = \frac{V_{Z(MIN)} - V_{LZ(SAT)}}{I_{OL}}$$

- C.TF.

หน้าที่หลักของวงจรนี้คือ ใช้เป็นตัวกลางในการ INTETFACE ระหว่างวงจร P.L.C.T.C. กับ A.C.LINE เพื่อทำให้สามารถส่ง-รับข้อมูลผ่าน A.C.LINE ได้โดยสะดวก



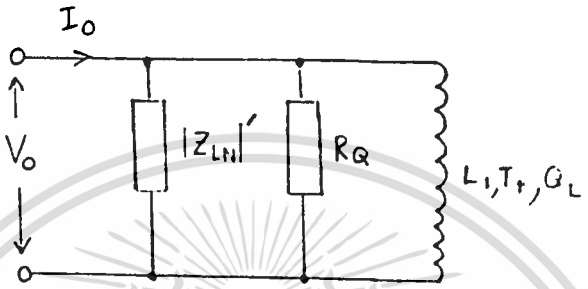
C.TF.

รูปที่ 2.12 แสดงวงจรสมบูรณ์ของส่วน C.TF.

จากวงจรในรูปที่ 2.12 อธิบายได้ว่า TX DATA จะผ่านขดลวด PRIMARY ของ I.F.T. กับ C_a ที่ต่อเป็นวงจร TANK เพื่อทำหน้าที่ในการ TUNE ความถี่ CARRIER ที่ต้องการพร้อมกับทำการ ISOLATE GROUND จาก A.C.LINE เพื่อป้องกันอันตรายที่เกิดจากไฟดูดและขจัด NOISE ที่เกิดขึ้นเนื่องจาก GROUND LOOP ได้ด้วย หลังจากนั้นข้อมูลจะถูก INDUCED ผ่านเข้ามายังขดลวด SECONDARY ของ I.F.T. ซึ่งจะมี C_c ต่ออยู่โดย ทำหน้าที่ให้ความถี่ CARRIER ผ่านเข้าออกในสาย A.C. ได้โดยสะดวกพร้อมกับลดทอนความถี่ของ A.C.LINE และความถี่ของ NOISE ที่อยู่ใน A.C.LINE ก่อนที่จะทำให้ I.C. เสียหายได้

การเลือกค่าของอุปกรณ์

- I.F.T.



รูปที่ 2.13 แสดงวงจรสมมูลย์ของรูปที่ 2.12

จากสมการ $$L_1 = \frac{R}{2*Pi*F*Q} \quad .....(1)$$

ดังที่ทราบแล้วว่า $F_Q = F_o$ ผลที่ได้คือ วงจจะมี MINIMUM BANDWIDTH หรือ MAXIMUM Q ซึ่งมีความจำเป็นที่จะใช้ในการส่งผ่านข้อมูลลง A.C.LINE

ในรูปที่ 2.12 เมื่อคิดที่สภาวะ FULL LOAD จะได้ดังรูปที่ 2.13 ดังนั้น

$$L_1 = \frac{R_Q \,/\!/\, |Z_{LN}|'}{2*Pi*F_o*Q_L} \quad .....(2)$$

โดยที่ F_o = RESONANT FREQUENCY ของวงจร C.TF.

L_L = PRIMARY INDUCTANCE

R_o = MODELS ALL TRANSFORMER LOSSES

Z_{LN} = LOWEST POWER LINE IMPEDANCE

$\left|Z_{LN}\right|'$ = REFLECTED Z_{LN}

Q_L = LOADED Q

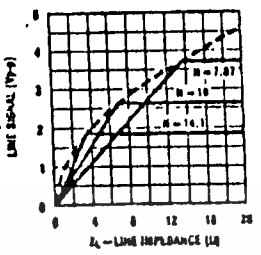
จาก $$P_o = I_o * V_o = \frac{I_{OPP}}{2\sqrt{2}} * \frac{2*(-V_{ALC} + V+)}{2\sqrt{2}}$$

$$= \frac{(-4.7+ V_+)*I_{OPP}}{4} \quad .....(3)$$

$$P_o = \frac{(18 - 4.7)*0.06}{4} = 0.2\ W$$

$$R_o /\!/ \left|Z_{LN}\right|' = V_o^2 / P_o = (-V_{ALC} + V_+)*\sqrt{2} = 442\ \Omega$$

ซึ่ง I_{OPP} = AMPS PEAK-TO-PEAK



รูปที่ 2.14 กราฟความสัมพันธ์ของ LINE SIGNAL กับ LINE IMPEDANCE

รูปที่ 2.14 สามารถหาค่าของ Z_{LN} ได้จากการเลือกค่า N (TURN RATIO) = 7.07 จากกราฟจะได้ Z_{LN} ประมาณ 13.9 ดังนั้น

$$\left|Z_{LN}\right|' = N^2 * Z_{LN} \qquad .....(4)$$

$$= (7.07)^2 * 13.9 = 695 \ \Omega$$

$$R_a = \frac{1}{1 / (K_a \left|\left|Z_{LN}\right|\right|) - 1 / \left|Z_{LN}\right|'} \qquad .....(5)$$

$$= \frac{1}{1/442 - 1/695} = 1210 \ \Omega$$

$$Q_L = \frac{1}{B.W.(\%\ OF\ F_o)} \quad .....(6)$$

โดยปกติแล้ว B.W. ควรทำให้มีค่าอยู่ประมาณ 8.7% ของ F_o ดังนั้นจะได้

$$Q_L = 1/.087 = 11.5$$

เพราะฉะนั้น

$$L_1 = \frac{442}{2*Pi*125000*11.5} = 49.0\ uH$$

$$T_1 = \sqrt{L_1/L} \quad .....(7)$$

$$T_2 = T_1/N \quad .....(8)$$

เมื่อ T_1 = NUMBER OF PRIMARY TURNS

T_2 = NUMBER OF SECONDARY TURNS

L = CORE INDUCTANCE PER TURN

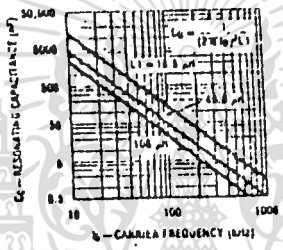
L_2 = SECONDARY INDUCTANCE

$$L_2 = (T_2 * \sqrt{L})^2 \quad .....(9)$$

- C_a

$$C_a = \frac{1}{(2*Pi*F_o)^2 * L_t} = 33 \text{ nF}$$

หรือ สามารถหาค่า C_a ได้จากกราฟต่อไปนี้

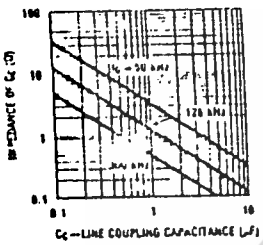


รูปที่ 2.15 กราฟความสัมพันธ์ของ F_o กับ C_a

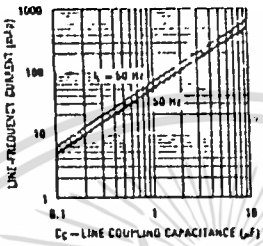
จากกราฟ จะเห็นว่าเมื่อกำหนดความถี่พาหะเท่ากับ 125 KHz และ L_t = 49.0 uH ซึ่งได้จากการคำนวน ก็ทำให้ทราบค่าของ C_a ประมาณ 33 nF

- C_c

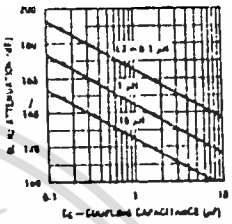
ค่าของ C_c สามารถหาได้จากกราฟดังต่อไปนี้



รูปที่ 2.16.1 กราฟความสัมพันธ์ของ C_c กับ อิมพีแดนซ์



รูปที่ 2.16.2 กราฟความสัมพันธ์ของ C_c กับ LINE CURRENT



รูปที่ 2.16.3 กราฟความสัมพันธ์ของ C_c กับ LOSS

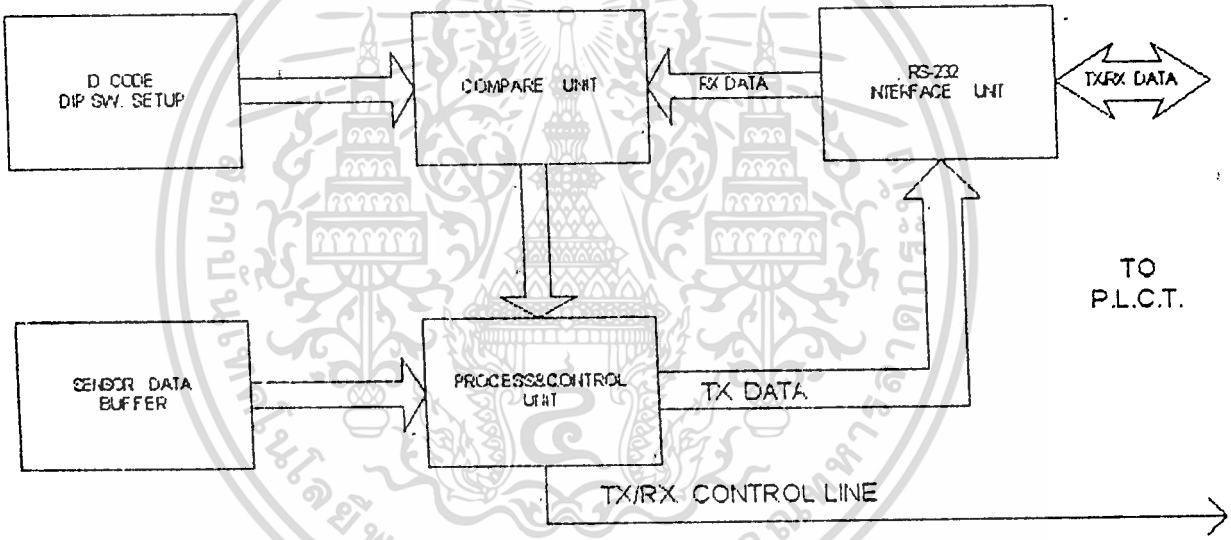
รูปที่ 2.16.1 AND 16.2 ใช้ประกอบกันในการหาค่าของ C_c ดังนี้คือ ต้องสุ่มเลือกค่าของ IMPEDAANCE OF C_c เช่น 4 โอห์ม โดยมีหลักว่า ต้องมีค่าน้อยกว่า LINE IMPEDANCE ที่ความถี่พาหะ 125 KHz ก็ทำให้ทราบค่าของ C_c ประมาณ 0.3 uF หลังจากนั้นนำค่าของ C_c นี้ไปตรวจสอบกับ รูปที่ 2.16.2 เพื่อดูว่าที่ค่าของ C_c เท่ากับ 0.3 uF ที่ตรงกับ LINE FREQUENCY เท่ากับ 50 Hz จะได้ค่าของ LINE-FREQUENCY CURRENT ประมาณ 20 mApeak ซึ่งจะต้องเป็นค่าที่น้อยเพียงพอเพื่อที่จะไม่ทำให้ I.F.T. อยู่ในสภาวะ SATURATION

หรือสามารถหาค่า C_c ได้ รูปที่ 2.16.3 ซึ่งจะใช้ได้ง่ายกว่า โดยเมื่อทราบค่าของ L_2 จากการคำนวณที่ได้กล่าวมาแล้วและทำการกำหนด ATTENUATION ของ 50 Hz ไว้ที่ค่าหนึ่งเช่น 140 dB ก็จะทำให้ทราบค่าของ C_c ประมาณเท่ากับ 0.3 uF

บทที่ 3

DETECTOR UNIT

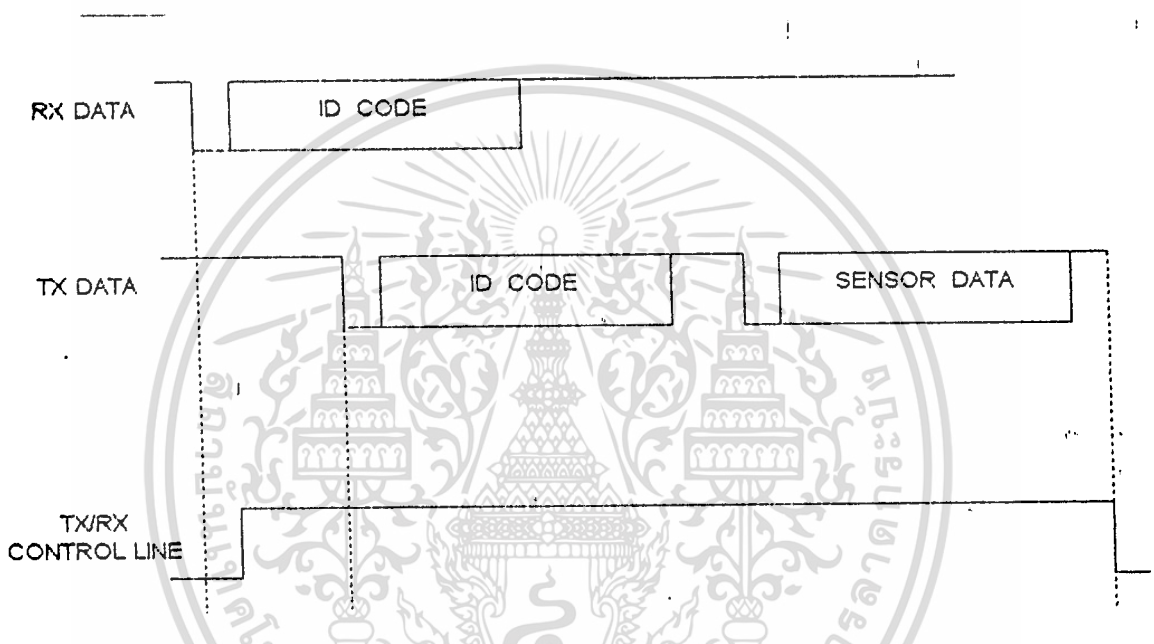
ในส่วนของภาค DETECTOR UNIT จะทำหน้าที่รับสัญญาณจากตัว SENSOR เข้ามา ซึ่งแต่ละ DETECTOR UNIT จะสามารถต่อ SENSOR ได้ 8 ตัว จากนั้นก็จะส่งสัญญาณซึ่งแสดงสภาวะของ SENSOR แต่ละตัว ออกไปเป็น SERIAL DATA เพื่อไป MODULATE ลง AC LINE อีกทอดหนึ่ง



รูปที่ 3.1 BLOCK DIAGRAM DETECTOR UNIT

รูปข้างบนแสดง BLOCK DIAGRAM ในส่วนของภาค DETECTOR UNIT การทำงานของวงจรจะเริ่มจากรับสัญญาณ SERIAL DATA เข้ามาทาง RX INPUT เพื่อมาทำการเปรียบเทียบกับ ID CODE ซึ่ง SET โดย DIP SW. ซึ่งก็คือหมายเลขของห้องนั่นเอง ถ้าหาก DATA INPUT

กับ ID CODE มีค่าตรงกัน ก็จะ ENABLE ให้ในส่วนของ TRANSMIT ทำงาน โดยข้อมูลที่ส่งออกไปชุดแรกจะเป็นการส่ง ID CODE ออกไปก่อน ส่วนข้อมูลชุดต่อไปจะเป็นการส่ง DATA ซึ่งเป็นสถานะของ SENSOR ตัวต่าง ๆ ออกไป



รูปที่ 3.2 TIMING DIAGRAM การส่งและรับข้อมูล

ในส่วนของการติดต่อกับภาค P.L.C.T. จะต้องมีสายสัญญาณ control เพื่อบอกให้ภาค P.L.C.T. รู้ว่าขณะต้องการส่งข้อมูลหรือรับข้อมูล

การทำงานของวงจร

จาก BLOCK DIAGRAM ดังที่ได้กล่าวมาแล้ว ในส่วนของการควบคุมการทำงานของเครื่องจะใช้ CPU เบอร์ 8031 และยังทำหน้าที่ทั้งในการรับและส่งข้อมูล ควบคุมการทำงานต่าง ๆ ได้โดยตัวของมันเอง ทำให้วงจรมีขนาดเล็กและอุปกรณ์น้อยชิ้น สะดวกในการปรับปรุง SOFTWARE เนื่องจากโปรแกรม MONITOR บรรจุอยู่ใน EPROM การออกแบบวงจรจะแบ่งออกเป็น 2 ส่วนคือ

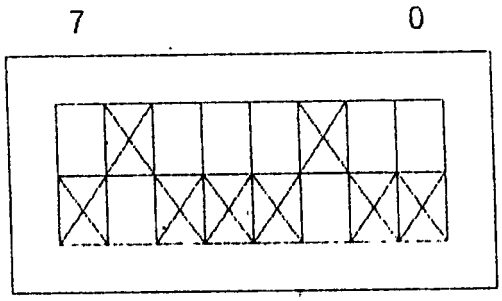
- ส่วนของ HARDWARE
- ส่วนของ SOFTWARE

-ส่วนของ HARDWARE

ในส่วนนี้หัวใจของวงจรก็อยู่ที่ MPU 8031 ซึ่งก็เป็น MPU ในตระกูล MCS-51 ภายในตัว 8031 จะมี PORT อยู่ภายใน 4 PORT และมีวงจรรับส่งข้อมูลแบบอนุกรมอยู่ภายในตัว PORT 1 จะถูกใช้เป็น INPUT PORT ต่ออยู่กับ DIP SW. S1 ที่แต่ละขาของ PORT 1 (P1.0-P1.7) จะถูกต่อ PULL UP อยู่ภายใน เพราะฉะนั้นที่แต่ละขาจะมีสถานะเป็น LOGIC "1" ในขณะที่ไม่ได้กับอะไร เมื่อ DIP SW. ที่ตำแหน่งใดอยู่ในตำแหน่ง ON จะทำให้ขานั้นถูกต่อลง GROUND จะได้เป็น LOGIC "0" ที่ขานั้น DIP SW.S1 ใช้เพื่อ SET ค่า ID CODE ซึ่งก็คือเบอร์ห้องนั่นเอง เพราะฉะนั้นจึงมีจำนวนห้องได้ $2^8-1 = 255$ ห้อง เมื่อนำ DETECTOR UNIT ไปติดตั้งตามห้องต่าง ๆ ก็เพียงแต่ SET ค่าของ DIP SW. S1 ให้มีค่าตรงกับห้องนั้น ๆ

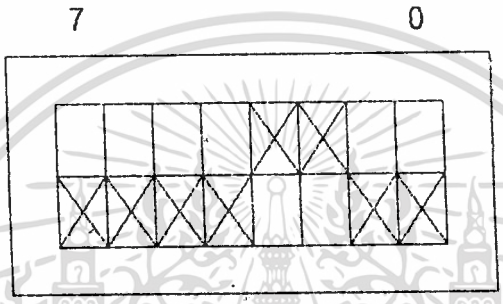
ส่วน PORT 0 และ 2 ถูกใช้เป็นขา ADDRESS และ DATA เนื่องจากใช้ EPROM ในการเก็บโปรแกรม MONITOR อยู่ภายนอก เพราะฉะนั้นขา EA ของ 8031 จะถูกต่อกับ LOGIC "0" เพื่อ ENABLE ให้เป็นการ RUN โปรแกรมจากหน่วยความจำภายนอก

EXAMPLE SET DIP SW.



รูปที่ 3.3 DIP SW. = 44H ROOM = #68

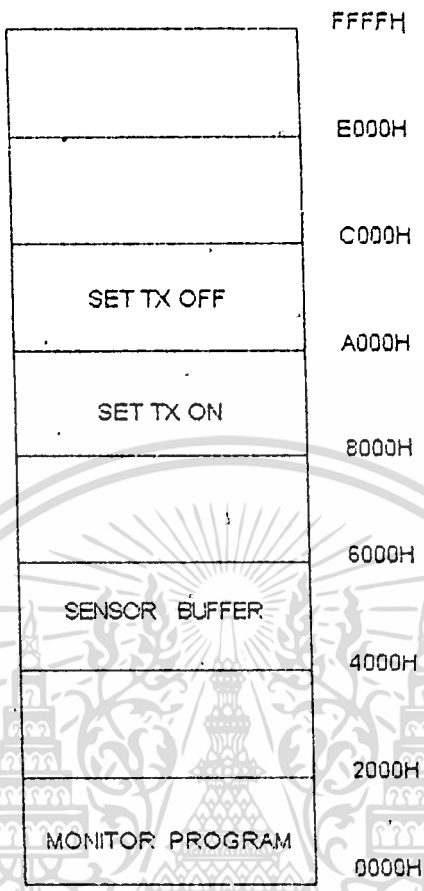
การเซ็ท DIP SW.



DIP SW. = 0CH ROOM = #12

เนื่องจากขา ADDRESS และ DATA ที่ PORT 0 ถูก MULTIPLEX อยู่ในขาเดียวกัน จึงจำเป็นต้องใช้ IC2 เพื่อทำการ DEMULTIPLEX ขา DATA และ ADDRESS ออกจากกัน โดยมีขา ALE ของ 8031(pin 30) เป็นตัวควบคุมในการแสดงสภาวะว่าเป็น ADDRESS หรือ DATA เมื่อขา ALE เป็น LOGIC "0" หมายถึงว่าขณะนี้ PORT 0 จะเป็น DATA ถ้าหากขา ALE เป็น LOGIC "1" แสดงว่า PORT 0 เป็น ADDRESS ขา ALE ถูกต่ออยู่กับขา C ของ IC2 เมื่อขา C เป็น 1 DATA ทาง INPUT ของ IC2 (1D-8D) จะถูก LATCH ไว้ที่เอาท์พุท (1Q-8Q) ส่วนที่ PORT 2 จะเป็นขา ADDRESS เพียงอย่างเดียว

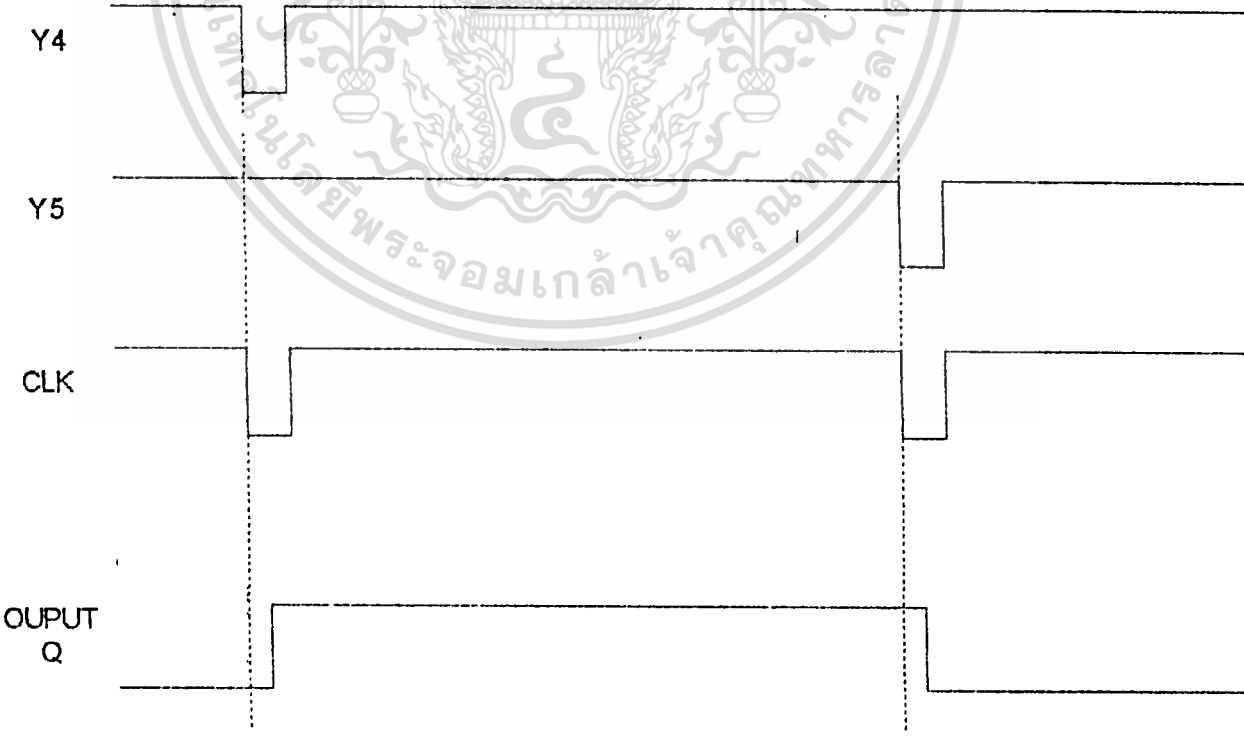
ในส่วนของการจัด MEMORY MAP จะใช้ IC4 74LS138 เป็นตัว DECODE ADDRESS ตัว 8031 สามารถอ้าง ADDRESS ได้สูงสุด 64 KB โดยที่ IC4 จะแบ่ง MEMORY ออกเป็น 8 SEGMENT SEGMENT ละ 8 KB ใน SEGMENT แรกเป็นตำแหน่งของโปรแกรม MONITOR SEGMENT ที่ 3 ใช้อ้างถึง PORT SENSOR SEGMENT ที่ 5,6 ใช้ในการควบคุมขา CONTROL TX/RX โดย MAP ของ MEMORY จะแสดงดังรูป



รูปที่ 3.4 การจัด MEMORY MAP ของหน่วยความจำ

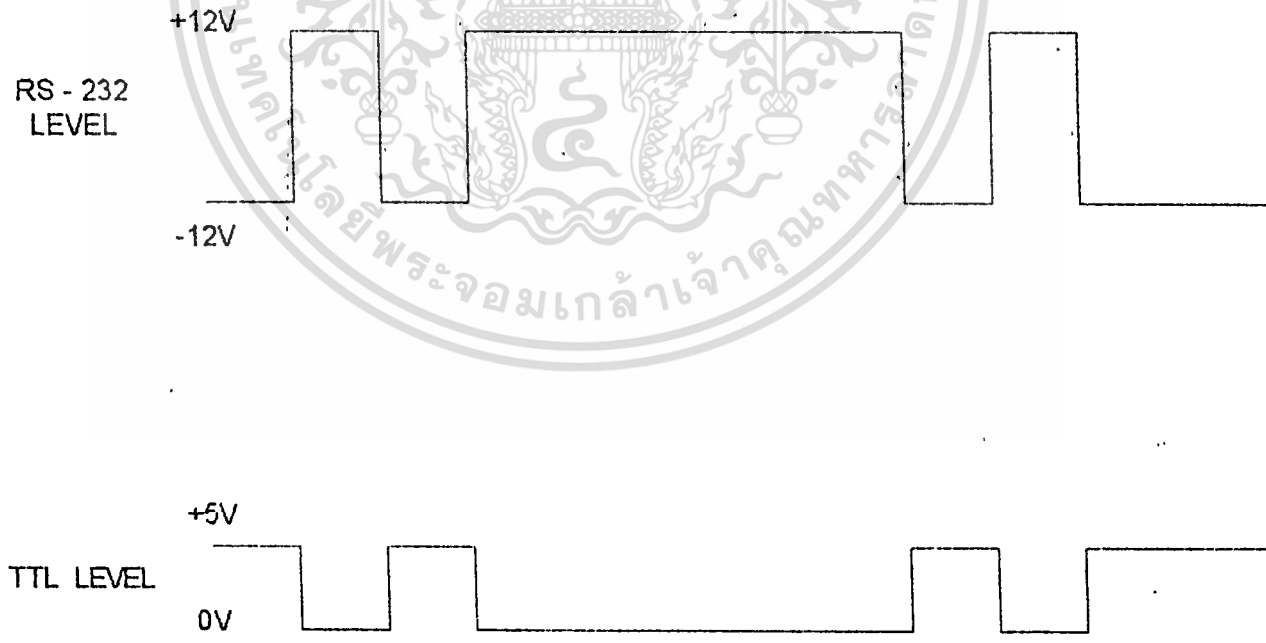
ขา ADDRESS A13-A15 ถูกต่อเข้ากับ INPUT ของ IC4 เพื่อใช้ในการ DECODE เมื่อขา A,B,C เป็น 0 ทั้งหมด จะทำให้ Y0 มีค่าเป็น 0 เพื่อไป ENABLE ให้ EPROM ทำงาน เมื่อขา A14 มีค่าเป็น 1 A13 และ A15 เป็น 0(ADDRESS 4000-4FFF) จะทำให้ Y2 มีค่าเป็น 0 ไป ENABLE ให้ PORT SENSOR ทำการรับข้อมูลมา LATCH ไว้ที่ขา OUTPUT ของ IC5 74LS541 ซึ่งต่ออยู่กับขา DATA ของ 8031 D1,D2,R1 ที่ต่ออยู่ที่ขา Y4 และ Y5 ทำงานเป็นเหมือน AND GATE ตัวหนึ่ง ปกติ LOGIC ที่ขาอาโนดของ D1,D2 จะมีค่าเป็น "1" เมื่อใดก็ตามที่ขา Y4 หรือ Y5 มีค่าเป็น 0 จะทำให้ที่ขาอาโนดของ D1,D2 มีค่าเป็น 0 ด้วย เอาท์พุทที่ได้ที่ขาอาโนดของ D1,D2 จะถูกต่อเข้ากับขา CLK ของ IC5 เพื่อใช้ในการสร้างเป็นขา CONTROL TX/RX ต่อไป

IC6 74ALS112 เป็น J-K FLIPFLOP ทำงานที่ขอบลบของสัญญาณที่ขา CLK E1 กับ R8 ต่อเป็นวงจร POWER ON RESET เมื่อเริ่มจ่ายไฟให้วงจรจะทำให้ขา OUTPUT Q มีค่าเป็น 0 ซึ่งก็คือสัญญาณ CONTROL นั่นเอง ถ้าหาก LOGIC ที่ขา CONTROL มีค่าเป็น 0 จะทำให้ภาค P.L.C.T. อยู่ในสภาวะรับข้อมูล ปกติที่ขา CLK จะมีระดับ LOGIC เป็น 1 เมื่อใดก็ตามที่ Y4 หรือ Y5 เป็น 0 จะทำให้ระดับ LOGIC ที่ขา CLK เปลี่ยนจาก 1 เป็น 0 นั่นก็คือขอบลบของสัญญาณ CLK ทำให้ FLIP FLOP ทำงาน อินพุทที่ขา J จะไปปรากฏที่ เอาท์พุท Q เนื่องจากขา J ต่ออยู่กับขา /Q ซึ่งมีค่าเป็น 1 เนื่องจากเมื่อตอนเริ่มทำงานได้ทำการรีเซทให้ขา Q มีค่าเป็น 0 เพราะฉะนั้นสัญญาณที่ขา CONTROL จึงมีสภาวะเป็น 1 ซึ่งจะทำให้ภาค P.L.C.T. อยู่ในสภาวะส่งข้อมูล เมื่อต้องการให้ขา CONTROL เป็น 0 อีกครั้ง ก็ให้สัญญาณที่ขา CLK เปลี่ยนจาก 1 ไปเป็น 0 อีกครั้งหนึ่ง โดยการให้ LOGIC ที่ขา Y4 หรือ Y5 มีค่าเป็น 0 ด้วยเหตุนี้สัญญาณที่ขา CONTROL จะ TOGGLE ไปมาระหว่าง 0 กับ 1 เมื่อสัญญาณที่ขา CLK เป็นพัลส์ลบเข้ามา



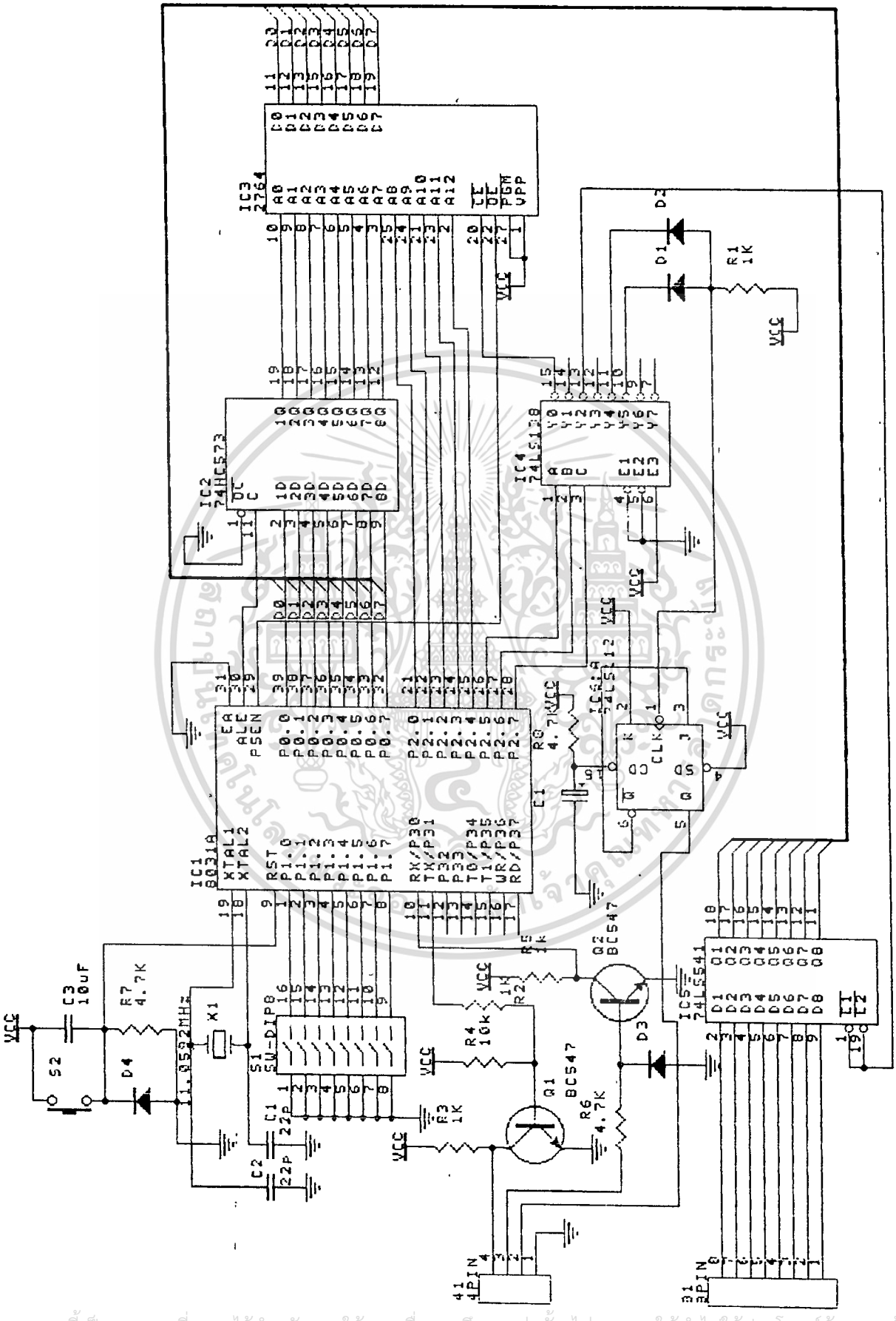
รูปที่ 3.5 TIMING DIAGRAM สัญญาณ CONTROL

ในส่วนของการรับส่งข้อมูลแบบอนุกรม จะใช้ขา P3.0 และ P3.1 ทำหน้าที่เป็นขา RX และ TX ตามลำดับ สัญญาณที่ขา P3.0 และ P3.1 เป็นสัญญาณแบบ TTL แต่ในการต่อ INTER FACE ต้องการสัญญาณแบบ RS232 จึงจำเป็นต้องมีภาค INTERFACE โดยประกอบด้วย Q1,Q2, R2-R6,D3 การเปลี่ยนสัญญาณจาก TTL ไปเป็น RS232 ก็เพียงแต่ INVERT สัญญาณเท่านั้น Q1,R3,R4,R5 ต่อเป็นวงจร COMMON EMITTER ได้สัญญาณเอาท์พุทออกไปที่ขา C ในลักษณะที่ INVERT กับสัญญาณที่เข้ามาทางขา B ส่วนทางด้านขา RX สัญญาณ RS232 ที่เข้ามามีค่าแรงดันได้ตั้งแต่ -15 ถึง +15 V สัญญาณ RS232 ที่เข้ามาผ่าน R6 เพื่อจำกัดกระแสทางขา B ของ Q2 โดยมี D3 ทำหน้าที่กั้นไฟลบที่จะมาทำความเสียหายให้กับ Q2 ได้ Q2 ต่อเป็นวงจร COMMON EMITTER เอาท์พุทที่ได้ที่ขา C จะ INVERT กับสัญญาณที่ขา B สัญญาณที่ออกไปที่ขา C จะมีแรงดันอยู่ในช่วง 0-5V



รูปที่ 3.6 แสดงระดับสัญญาณ RS-232 เทียบกับ TTL

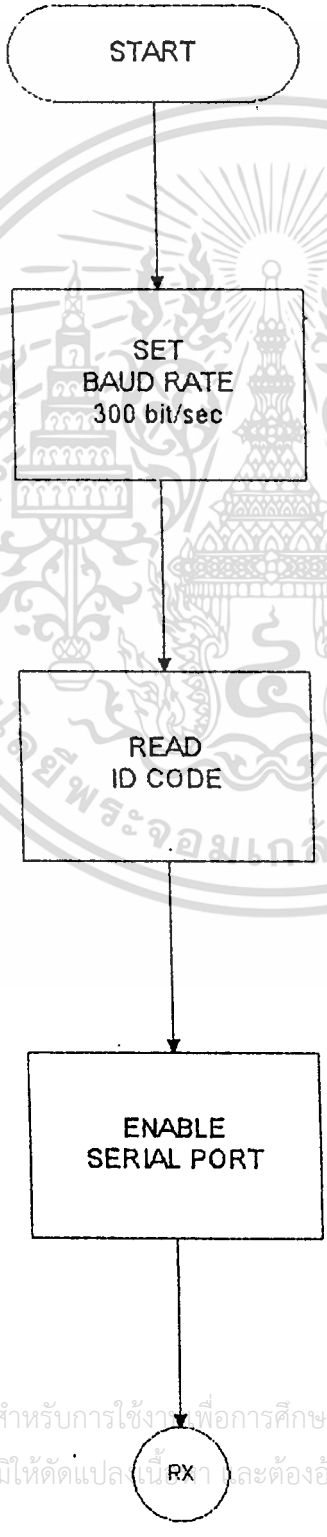
C3,D4,R7 และ S2 เป็นวงจร POWER ON RESET สำหรับ 8031 เนื่องจาก 8031 ต้อง LOGIC 1 เพื่อใช้ในการ RESET ทันทีที่จ่ายไฟให้กับวงจร จะมีกระแสไหลผ่าน C3 อย่างรวดเร็ว ทำให้แรงดันที่ตกคร่อม R7 มีค่าประมาณ V_{cc} หลังจากนั้นกระแสจะไหลผ่าน C3 น้อยลงจนกระทั่งหยุดไหล เมื่อกระแสหยุดไหลแรงดันที่ตกคร่อม R7 จะมีค่าเป็น 0 การ RESET จึงเกิดขึ้นในช่วงแรกที่เริ่มจ่ายไฟเข้าวงจร ถ้าหากต้องการ RESET เป็นแบบ MANUAL ก็ใช้โดยการกดสวิทช์ S2 เมื่อกด S2 ก็จะทำให้ที่ขา RST มีค่าเป็น 1

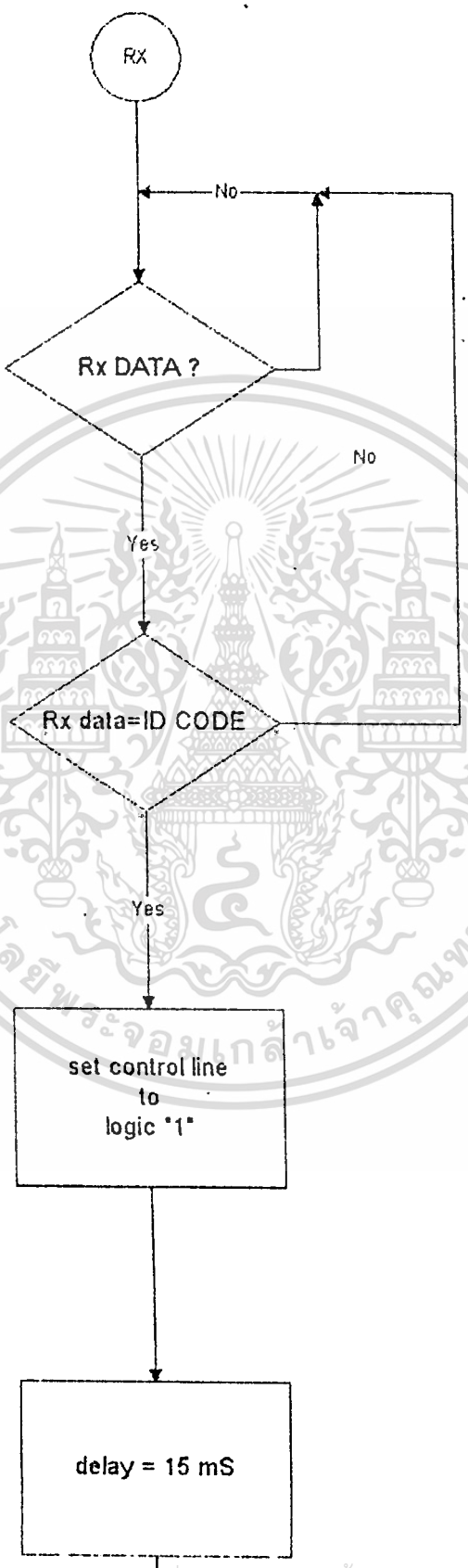


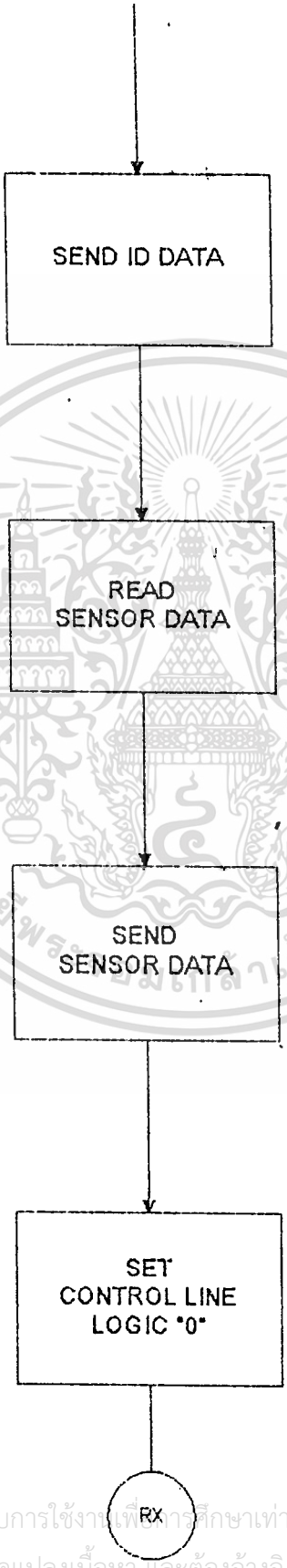
รูปที่ 3.7 วงจร DETECTOR UNIT

-ส่วนของ SOFTWARE

ส่วนของโปรแกรม MONITOR จะถูกเก็บอยู่ใน EPROM การทำงานของโปรแกรม MONITOR เขียนเป็น FLOWCHART ดังรูป







เริ่มต้นการทำงานของโปรแกรมจะทำการ SET BAUD RATE ในการส่งข้อมูลก่อนโดยมีค่าเป็น 300 bit/sec ซึ่งจะต้องคำนวณค่าที่ load ลงไปในรีจิสเตอร์ TH1 ดังนี้

$$TH1 = 256 - \frac{K \times \text{Osc Freq.}}{384 \times \text{baud rate}}$$

เมื่อ K มีค่า = 2 โดยการ set รีจิสเตอร์ PCON เป็น 80H

Osc Freq. เป็นความถี่ของสัญญาณนาฬิกาที่ป้อนให้ CPU

Baud rate มีหน่วยเป็น bit/sec

ดังนั้น

$$TH1 = 256 - \frac{2 \times 11.592 \times 10^{6}}{384 \times 300}$$

$$= 64_{10}$$

$$= 40H$$

ขั้นต่อไปเป็นการอ่านค่า ID CODE จาก DIP SW. เข้ามาเก็บไว้ในรีจิสเตอร์ R0 ในรีจิสเตอร์ BLANK 0

การทำงานของ serial port จะอยู่ใน mode 1 โดยการโหลดค่า 50H ลงในรีจิสเตอร์ SCON และให้รีจิสเตอร์ TMOD = 20H เพื่อให้ทำงานแบบ AUTO-RELOAD คือวนรอบการทำงานตลอดเวลา แล้วจึงเริ่มให้ TIMER ทำงานโดย set TR1

เมื่อทำการ initial CPU เรียบร้อยแล้ว ก็จะทำงานรอรับข้อมูลที่เข้ามาทางขา Rx แล้วทำการเปรียบเทียบข้อมูลที่ได้กับค่าในรีจิสเตอร์ R0 จนกว่าจะตรงกัน จึงกระโดดไปทำงานในส่วนของการส่งข้อมูล

เมื่อตรวจนบว่าข้อมูลที่รับเข้ามาตรงกับค่าใน R0 ก็จะเข้าสู่ในสวนของโปรแกรม TX: ก่อนที่จะทำการส่งข้อมูลออกต้อง set CONTROL LINE ให้เป็น 1 ก่อน โดยอ้างไปที่แอดเดรส 8000H แล้วหน่วงเวลาประมาณ 15 mS โดยเรียกโปรแกรมย่อย DELAY แล้วจึงส่ง DATA ออกไป 2 ชุด ชุดแรกเป็น ID CODEชุดที่ 2 เป็น ALARM STATUS เมื่อส่งข้อมูลครบ 2 ชุดแล้ว จะไปทำการ CLEAR CONTROL LINE เป็น 0 แล้วจึงวนกลับไปรอรับข้อมูลใหม่

บทที่ 4

CONTROL UNIT

- การสื่อสารข้อมูลอนุกรมแบบอะซิงโครนัส
(ASYNCRONOUS SERIAL DATA COMMUNICATION)

ภายในไมโครคอมพิวเตอร์ การติดต่อสื่อสารข้อมูลระหว่างส่วนต่างๆ ของระบบจะเป็นแบบขนาน เพราะว่าการส่งข้อมูลแบบขนานนี้มีความเร็วในการส่งข้อมูลสูง สำหรับการส่งถ่ายข้อมูลระยะทางไกล ๆ การส่งข้อมูลแบบขนานจะเกิดปัญหาขึ้น กล่าวคือการส่งข้อมูลแบบขนานต้องใช้ตัวกลางในการส่งเป็นจำนวนมาก เช่น ถ้าต้องการส่งข้อมูลขนาด 8 บิท จะต้องใช้สายไฟอย่างน้อยสุดคือ 9 เส้น (ข้อมูล 8 เส้น ระดับอ้างอิงอีก 1 เส้น) อีกปัญหาหนึ่งคือ เรื่องสัญญาณรบกวนตามทฤษฎีสายไฟทุกเส้นมีค่าความเหนี่ยวนำอยู่ในตัวเอง และสายไฟที่วางไว้เป็นกลุ่มจะเกิดค่าความจุขึ้นระหว่างสายไฟ คุณสมบัติของตัวเหนี่ยวนำและตัวเก็บประจุที่เกิดขึ้น ทำให้ลักษณะของรูปคลื่น (Wareform) ที่ส่งมาตามสายมีลักษณะผิดเพี้ยนไปจากเดิม

ดังนั้น เมื่อต้องการที่จะส่งข้อมูลไปเป็นระยะทางไกล ๆ เราจึงจำเป็นที่ต้องเปลี่ยนลักษณะของข้อมูลจากแบบขนานไปเป็นแบบอนุกรม ซึ่งการส่งข้อมูลในลักษณะอนุกรมนี้ จำนวนตัวกลางที่ใช้ (ซึ่งอาจเป็นสายไฟ ฯลฯ) จะมีเพียงหนึ่งหรือสองเส้นเท่านั้นเมื่อถึงปลายทางข้อมูลอนุกรมที่รับจากตัวส่งต้นทาง จะถูกเปลี่ยนให้อยู่ในลักษณะข้อมูลแบบขนาน ก่อนที่จะส่งผ่านเข้าไปยังระบบบัสของไมโครคอมพิวเตอร์

การสื่อสารข้อมูลอนุกรมแบบอะซิงโครนัส เราสามารถแบ่งออกได้ 3 ลักษณะคือ

1) Simplex มีลักษณะคือ สายสัญญาณข้อมูลสามารถส่งได้ทิศทางเดียว เช่น ตัวเซนเตอร์วัดคลื่นความสั่นสะเทือนจะส่งข้อมูลกลับมายังเครื่องวัด

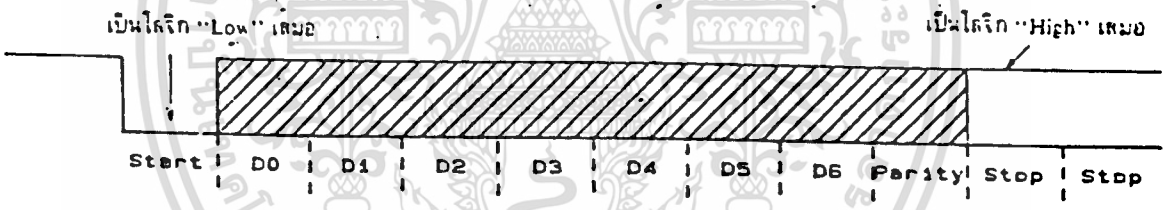
2) Half-duplex การสื่อสารข้อมูลเป็นไปได้ 2 ทิศทางบนสายส่งเพียงชุดเดียว แต่มีข้อจำกัดคือในช่วงเวลานั้น จะต้องเป็นการรับหรือเป็นการส่งข้อมูลอย่างใดอย่างหนึ่งเท่านั้น หรืออีกนัยหนึ่งไม่สามารถที่จะรับหรือส่งข้อมูลไปตามสายในขณะเวลาเดียวกันได้ ตัวอย่างการส่งข้อมูลแบบ Half-duplex คือระบบวิทยุของตำรวจ

3) Full-duplex เป็นระบบที่เราสามารถทำการรับและส่งข้อมูลในเวลาเดียวกันได้

ตัวอย่างของ Full duplex คือ ระบบโทรศัพท์ ลักษณะการส่งข้อมูลแบบอะซิงโครนัส ข้อมูลอักขระแต่ละตัว จะประกอบด้วย 1 บิท เริ่มต้น (Start bit) 5,6,7, หรือ 8 บิท

ข้อมูล (Data bit) 1 ถึง 2 บิทจบ (Stop bit) การส่งจะทำการส่งครั้งละ 1 อักขระ ไปเรื่อย ๆ จนกว่าจะหมดข้อมูลที่ทำการส่ง

รูปที่ 4.1 แสดงถึงรูปแบบของบิทที่ใช้สำหรับการส่งข้อมูลอนุกรมแบบอะซิงโครนัส ขณะที่เรายังไม่ทำการส่งข้อมูลสัญญาณในสายจะมีค่าเป็นโลจิก "High" หรือที่เราเรียกว่า Marking State เริ่มต้นของการส่งข้อมูลเริ่มจากสัญญาณ "High" ของ Marking State เปลี่ยนเป็นโลจิ "Low" ขนาด 1 บิท สเตทที่เปลี่ยนจากโลจิก "High" มาเป็นโลจิก "Low" เราเรียกว่า "Start bit" หลังจาก Start bit บิท ข้อมูลจะถูกส่งตามมา มีข้อสังเกตว่าบิทที่มีนัยสำคัญต่ำ



รูปที่ 4.1 รูปแบบของบิทที่ใช้สำหรับส่งข้อมูลอนุกรมในอะซิงโครนัสโหมด

สุด (least significant bit) จะถูกส่งออกมาก่อน จำนวนของบิทข้อมูลอาจเป็น 5,6,7 หรือ 8 บิท ขึ้นอยู่กับระบบ แต่ที่นิยมใช้คือ 7 รือ 8 บิท หลังจากบิทข้อมูลถูกส่งออกมาครบแล้ว (ตามที่กำหนดในระบบ) พาริตี้บิท (Parity bit) จะถูกส่งออกมา Parity bit มีประโยชน์คือ ช่วยในการตรวจสอบว่าบิทข้อมูลที่อุปกรณ์ปลายทางจะนำเอาบิทข้อมูลทั้งหมดมารวมกัน เพื่อดูค่าที่ได้ว่าตรงกับ Parity bit ให้มีลักษณะเป็นบิทคู่, บิทคี่ หรือไม่สนใจ Parity bit ก็ได้ ขึ้นอยู่กับระบบที่เราใช้

หลังจากบิทข้อมูลและพาริตี้บิท สัญญาณในสายจะถูกทำให้กลับมาสู่ระดับ "High" สภาวะนี้อย่างน้อยที่สุดต้องมีขนาด 1 บิท เพื่อกำหนดการสิ้นสุดของการส่งอักขระ บิทสุดท้ายนี้ต้องมีระดับ เป็น "High" เสมอ ซึ่งเราเรียกบิทนี้ว่า "Stopbit" ระบบบางระบบจะกำหนดให้บิทสิ้นสุดการส่งข้อมูลนี้มีขนาด 2 บิท แต่รูปแบบของข้อมูลที่ใช้บิทสิ้นสุดจำนวน 2 บิท ไม่เป็นที่นิยมเพราะมีประสิทธิภาพต่ำ เนื่องจากใน 1 ครั้งของการส่งอักขระ 1 ตัว ต้องใช้จำนวนบิททั้งหมดถึง 10 หรือ 11

หลังจากที่เราพูดถึงรูปแบบของข้อมูลในการส่งข้อมูลอนุกรมมาแล้ว สิ่งที่เราจะพูดถึงกันต่อไป คือ อัตราของข้อมูลอนุกรมที่จะถูกส่งออกไป ซึ่งเราเรียกว่า baud rate คำว่า baud rate ถูกกำหนดไว้ดังนี้ คือ

baudrate = 1/(เวลาระหว่างการเปลี่ยนแปลงของสัญญาณ) เช่นถ้าสัญญาณเปลี่ยนแปลงทุก ๆ 3.33 ms band rate คือ 1/3.33 ms หรือ 300 Bd. การกำหนดในลักษณะแบบนี้มักไม่นิยมใช้แต่นิยมเขียนอยู่ในรูป 300 bit/s. (300 บิทต่อ 1 วินาที) ในบางกรณี การเขียนทั้ง 2 แบบนี้ มีความหมายเหมือนกัน แต่ในกรณีที่พูดถึง bit/s แต่บิทข้อมูลถูกเข้ารหัสไว้ กรณีนี้คำว่า baud rate และ bit/s จะมีความหมายไม่เหมือนกัน

ตามปกติ baud rate จะถูกกำหนดไว้ดังนี้ 300, 600, 1,200, 2,400, 4,800, 9,600 และ 19,200

การเชื่อมต่อไมโครคอมพิวเตอร์กับสายข้อมูลอนุกรม ข้อมูลจะต้องถูกเปลี่ยนจากข้อมูลแบบขนานให้ไปอยู่ในแบบอนุกรม โดยการใช้อุปกรณ์ที่เรียกว่า parallel-in-serial-out shift register และใช้อุปกรณ์ serial-in parallel-out shift register เป็นตัวรับเพื่อแปลงข้อมูลแบบอนุกรม ให้เป็นข้อมูลแบบขนานก่อนที่จะส่งผ่านเข้าสู่บัสของระบบ

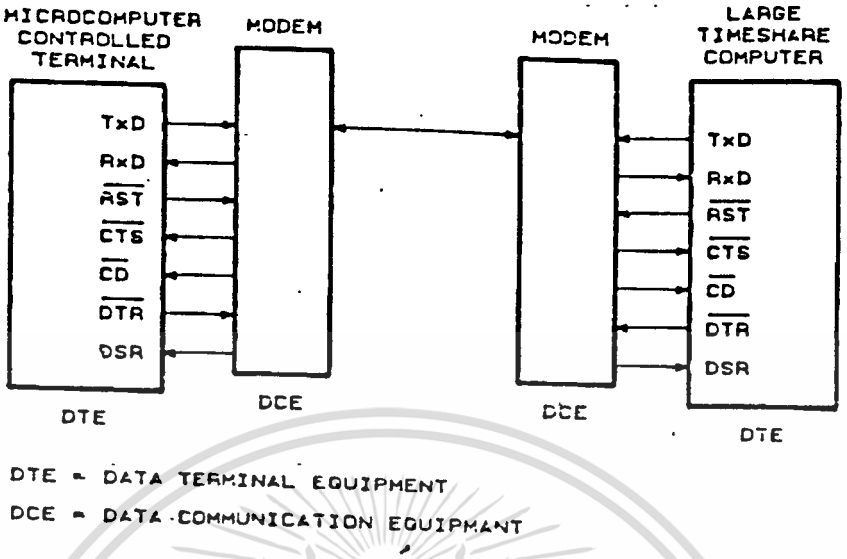
ในบางครั้งการส่งผ่านข้อมูลแบบอนุกรมมีวงจรตรวจสอบความพร้อม (Handshaking circuitry) เพื่อให้มั่นใจว่าตัวส่งข้อมูลจะไม่ส่งข้อมูลเร็วเกินไปจนตัวรับไม่สามารถรับข้อมูลได้ทัน

ในปัจจุบันมีอุปกรณ์ LSI หลาย ๆ ตัวที่ใช้กับการติดต่อสื่อสารแบบอนุกรม เช่น INS 8250 ของบริษัท National เป็นไอซี LSI ที่ออกแบบมาเป็น Asynchronous communication เพียงอย่างเดียว ซึ่งอุปกรณ์แบบนี้มีชื่อเรียกอีกอย่างหนึ่งคือ Univeral asynchronous receiver-transmitter หรือ UART ไอซี LSI ที่ใช้งานในการสื่อสารแบบอนุกรมอีกเบอร์หนึ่งคือ Intel 8251 A ไอซีเบอร์นี้สามารถโปรแกรมให้ทำงานเป็นแบบอะซิงโครนัส หรือซิงโครนัสโหมดก็ได้ เราเรียกอุปกรณ์แบบนี้ว่า Universal synchronous-asynchronous receiver transmitter หรือ USART

ข้อมูลที่ถูกเปลี่ยนให้อยู่ในรูปอนุกรมจะถูกส่งจาก UART ตัวส่งไปยัง UART ตัวรับลักษณะการส่งที่ส่งออกไปนี้มีหลายแบบด้วยกัน วิธีหนึ่งคือการใช้กระแสไฟฟ้าส่งไปตามสายส่ง กล่าวคือถ้ามีกระแสไฟฟ้าปรากฏอยู่ในสายส่งก็จะแทนระดับโลจิก "1" หรือโลจิก "High" ส่วนโลจิก "0" หรือโลจิก "Low" แทนด้วยสภาวะที่ไม่มีกระแสไฟฟ้าปรากฏอยู่ในสาย วิธีการนี้เราเรียกว่า Current Loop Method วิธีนี้เราจะพูดถึงรายละเอียดในหัวข้อวิธีการและมาตรฐานของการสื่อสารข้อมูลแบบอนุกรม อีกวิธีหนึ่งคือการเพิ่ม "line drive" ไว้ที่เอ้าต์พุต ของ UART เพื่อที่จะผลิตสัญญาณแรงดันไฟฟ้าแรงดันสูงออกมาส่งไปตามสายส่ง วิธีนี้เป็นวิธีที่นิยม แต่ถูกจำกัดไว้ที่ระยะทางของการเชื่อมต่อจะได้เพียงไม่กี่ร้อยฟุตเท่านั้น

แต่ถ้าเราต้องการส่งข้อมูลไปเป็นระยะทางไกล ๆ เช่น คอมพิวเตอร์ตัวหนึ่งอยู่ที่จังหวัดเชียงใหม่ ต้องการติดต่อกับระบบเมนเฟรมซึ่งตั้งอยู่ที่กรุงเทพฯ การลากสายอนุกรมจากเชียงใหม่มายังกรุงเทพฯ คงเป็นไปไม่ได้ วิธีที่น่าจะดีที่สุดคือการใช้ระบบโทรศัพท์เป็นทางเดินของสัญญาณแทน ข้อจำกัดของระบบโทรศัพท์ที่เราควรรู้คือความกว้างของสัญญาณหรือที่เราเรียกว่าแบนด์วิดธ์ (Bandwidth) ของระบบโทรศัพท์นั้น มีความกว้างเพียง 300 ถึง 3,000 Hz เท่านั้น ดังนั้นสัญญาณดิจิตอล รูปที่ 4.1 จึงไม่สามารถส่งไปตามสายโทรศัพท์ได้โดยตรง เนื่องจากข้อจำกัดของแบนด์วิดธ์ดังที่ได้กล่าวมาแล้วในเบื้องต้น

วิธีแก้ปัญหาในการส่งข้อมูลไปตามสายโทรศัพท์ สามารถทำได้โดยเปลี่ยนสัญญาณดิจิตอลให้เป็นความถี่เสียง ซึ่งเป็นช่วงความถี่ที่สามารถส่งผ่านไปตามสายโทรศัพท์ได้



รูปที่ 4.2 การส่งข้อมูลผ่านโมเด็มไปตามสายโทรศัพท์

อุปกรณ์ที่ทำเปลี่ยนสัญญาณดิจิตอลให้เป็นความถี่เสียงและเปลี่ยนความถี่เสียงนี้ กลับมาเป็นสัญญาณดิจิตอล เราเรียกอุปกรณ์ตัวนี้ว่าโมเด็ม (Modem) คำว่า Modem มาจากคำว่า Modulator-Demodulator เราจะพูดถึงการทำงานของโมเด็มและชนิดของโมเด็มกันอีกครั้งในบทความตอนต่อไป

รูปที่ 4.2 แสดงถึงการใช้โมเด็ม 2 ตัว เชื่อมต่อระหว่างเทอร์มินอล เข้ากับตัวเมนเฟรมผ่านสายโทรศัพท์ ตัวโมเด็มหรือเครื่องมืออื่น ๆ ที่ใช้ส่งข้อมูลแบบอนุกรมไปเป็นระยะทางไกลเราเรียกอุปกรณ์เหล่านี้ว่า Data Communication Equipment หรือ DCE ตัวเทอร์มินอลและคอมพิวเตอร์ ซึ่งเป็นตัวส่งหรือตัวรับข้อมูล เราเรียกว่า Data Terminal Equipment หรือ DTE

ชื่อของสัญญาณข้อมูลและชื่อของสัญญาณตรวจสอบความพร้อม ตามที่แสดงอยู่ในรูปที่ 4.2 เป็นส่วนหนึ่งของมาตรฐานการสื่อสารข้อมูลแบบอนุกรมที่เรียกว่า RS-232C ทิศทางของลูกศรที่แสดงอยู่ในรูป 4.2 หมายถึง ทิศทางของสัญญาณต่างๆ ของตัวอุปกรณ์ DTE และ DCE เมื่อมีการเชื่อมต่ออุปกรณ์ทั้ง 2 แบบนี้เข้าด้วยกัน

ลำดับของสัญญาณซึ่งเกิดขึ้นเมื่อผู้ใช้เทอร์มินอลต้องการที่จะส่งข้อมูล ไปยังอุปกรณ์ปลายทางอีกตัวหนึ่งโดยผ่านทางโมเด็ม มีขั้นตอนการทำงานดังนี้

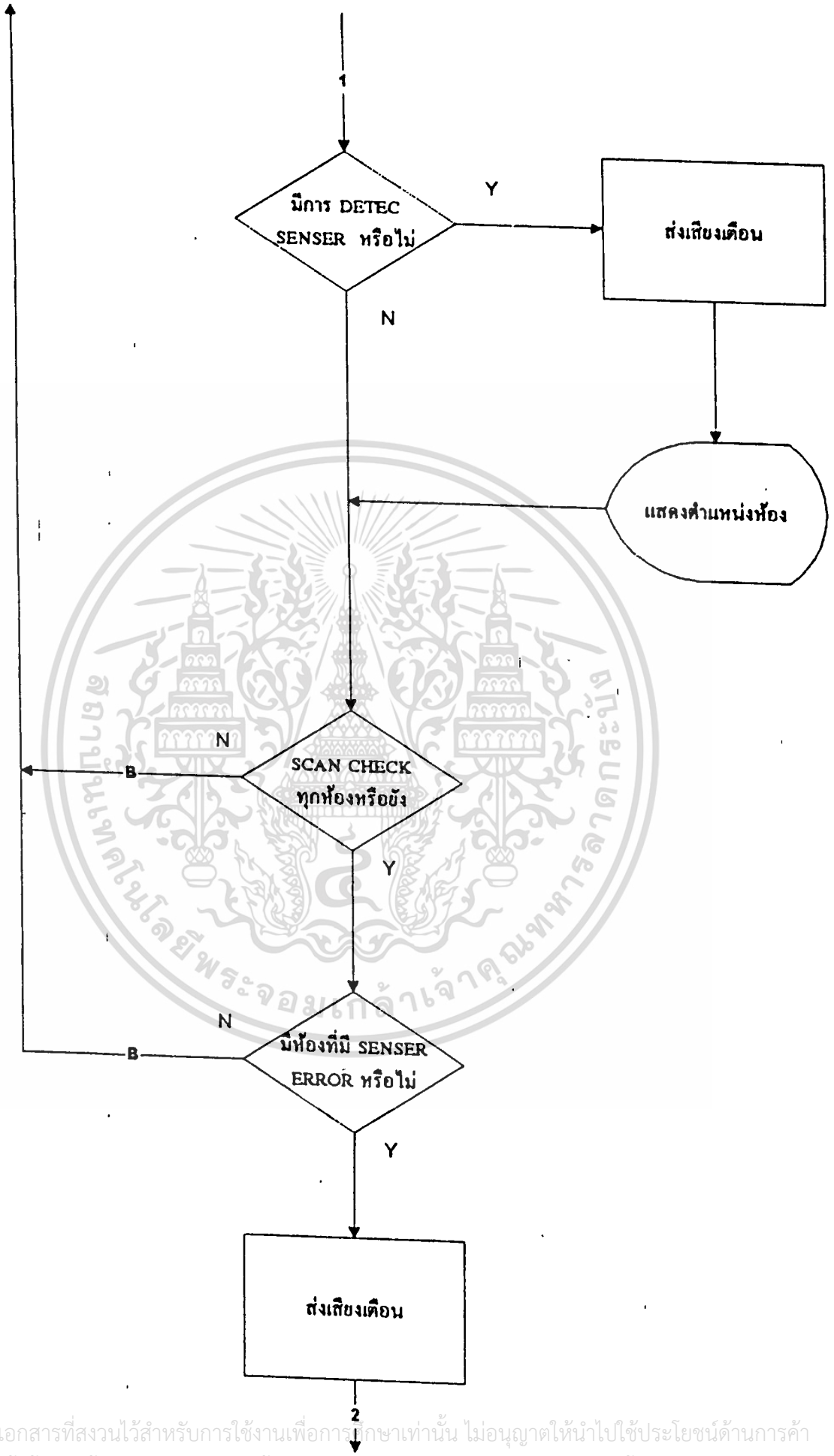
Step 1). หลังจากที่เทอร์มินอลเข้าสู่สภาวะพร้อมที่จะทำงานแล้วตัวเทอร์มินอลจะทำการส่งสัญญาณ data-treminal-ready (DTR) เพื่อบอกให้ตัวโมเด็มทราบว่าเทอร์มินอลพร้อมที่จะทำการรับหรือส่งแล้ว ขณะเดียวกันที่ตัวโมเด็ม เมื่อโมเด็มพร้อมที่จะทำงานมันจะส่งสัญญาณมายังขาสัญญาณ data-set-ready (DSR) ที่ตัวเทอร์มินอล เพื่อบอกให้เทอร์มินอลทราบว่าตัวโมเด็มพร้อมที่จะทำงานเช่นกัน

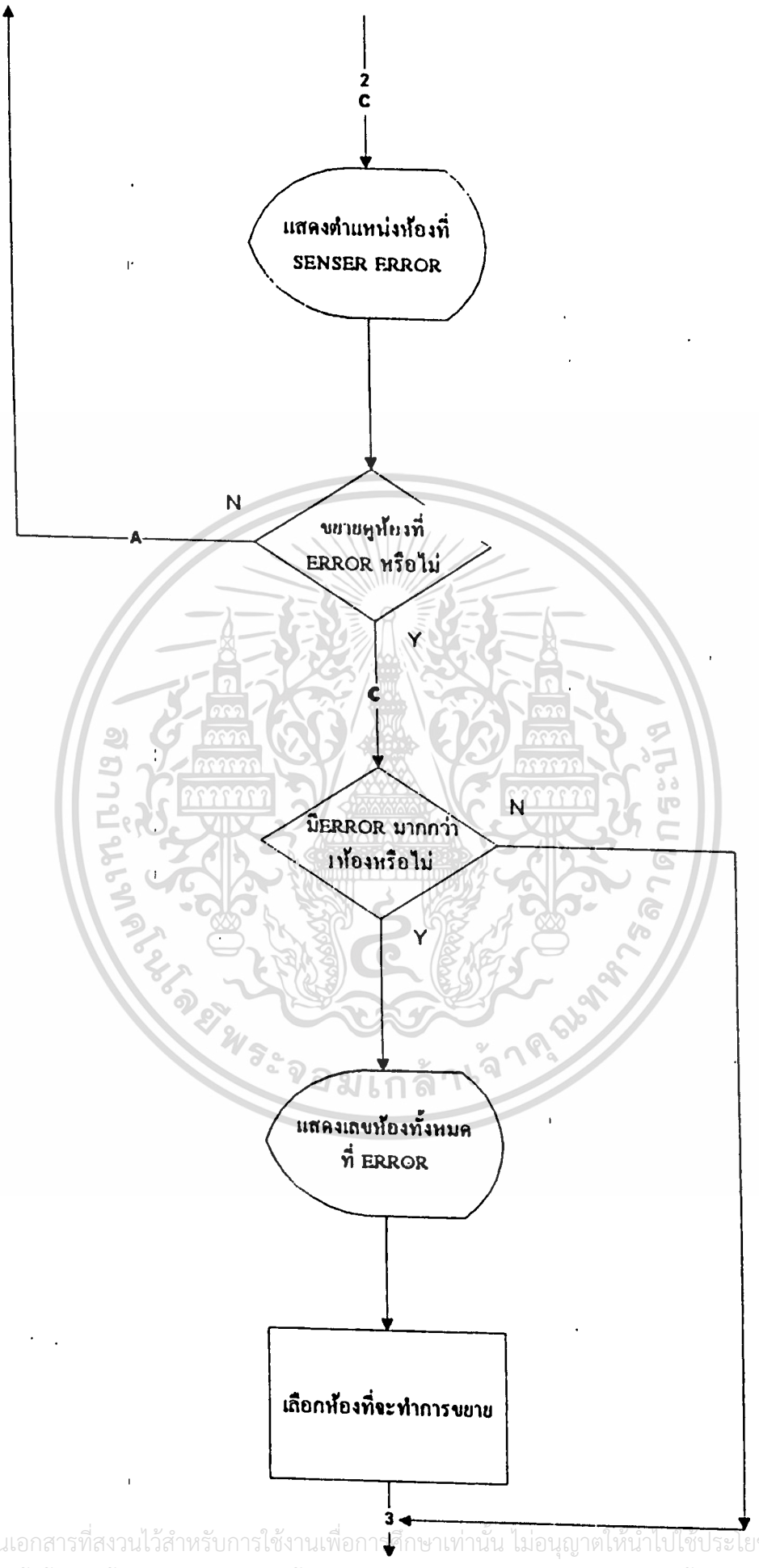
Step 2). เป็นขั้นตอนการ "dials up" ไปยังอุปกรณ์ปลายทาง ขั้นตอนนี้คล้ายกับเราหมุนโทรศัพท์ไปยังเลขหมายปลายทางวิธีการ "dials up" โมเด็ม มี 2 วิธีคือ ผู้ใช้เป็นคนทำเองโดยผ่านตัวโทรศัพท์ซึ่งต่อพ่วงกับโมเด็ม หรือทำการ "didls up" ผ่านตัวเทอร์มินอลก็ได้

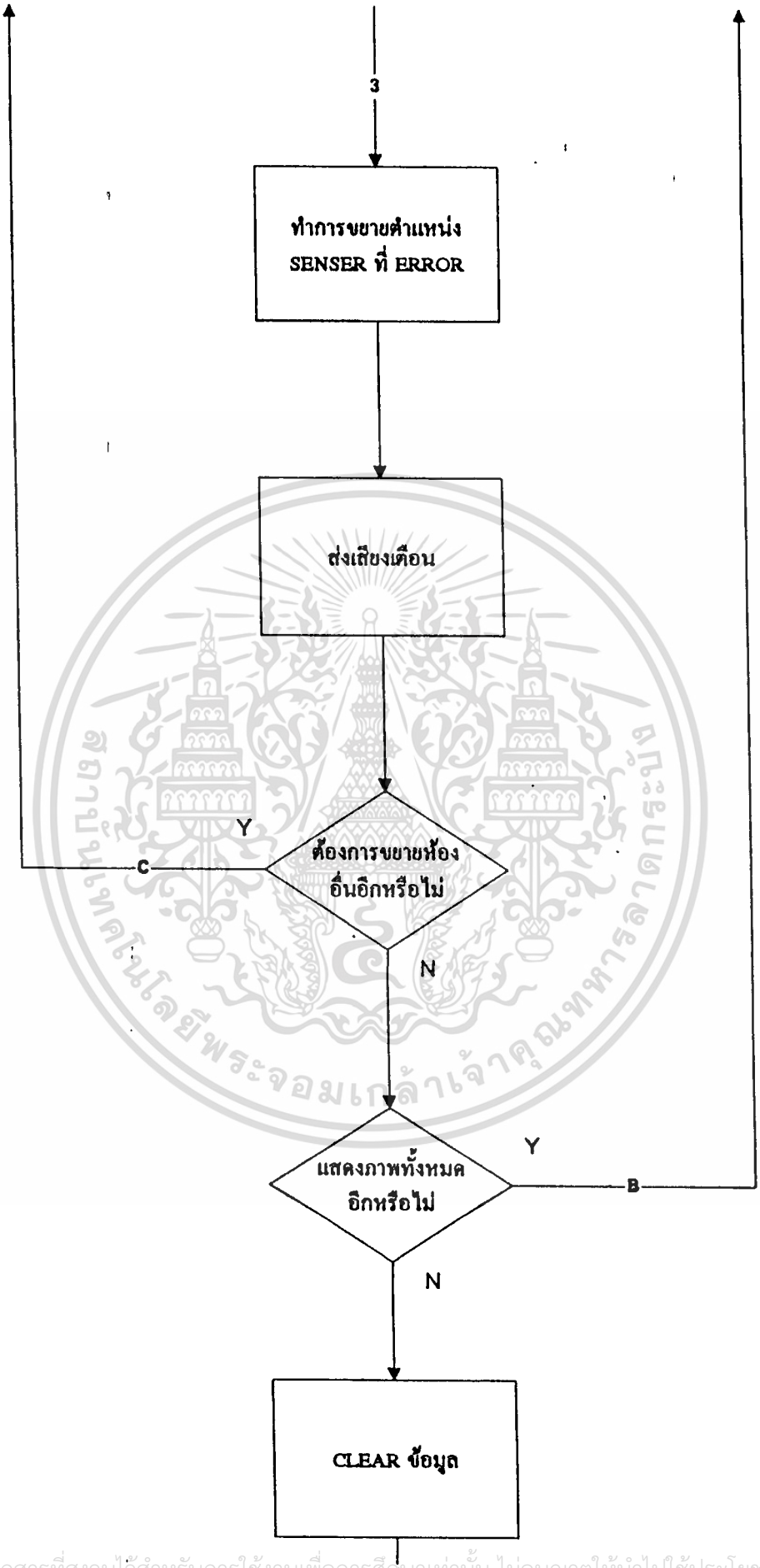
Step 3). หลังจากที่ทำการ "dials up" และโมเด็มต้นทางและโมเด็มปลายทางติดต่อกันได้แล้ว เมื่อเทอร์มินอลต้องการส่งข้อมูลออกไปยังอุปกรณ์ปลายทาง เริ่มแรก ตัวเทอร์มินอลจะส่งสัญญาณออกมาทางขาสัญญาณ request-to-send (RTS) ไปยังโมเด็ม ถ้าตัวโมเด็มยังคงติดต่ออยู่กับตัวโมเด็มปลายทางได้ (มี Carrier ที่จะส่งข้อมูลผ่านไปอยู่) และตัวโมเด็มเองพร้อมที่จะทำงานตัวโมเด็มก็จะส่งสัญญาณกลับมาทางขาสัญญาณ carrier-detect (CD) และขาสัญญาณ clear-to-send (CTS) ที่ตัวเทอร์มินอล

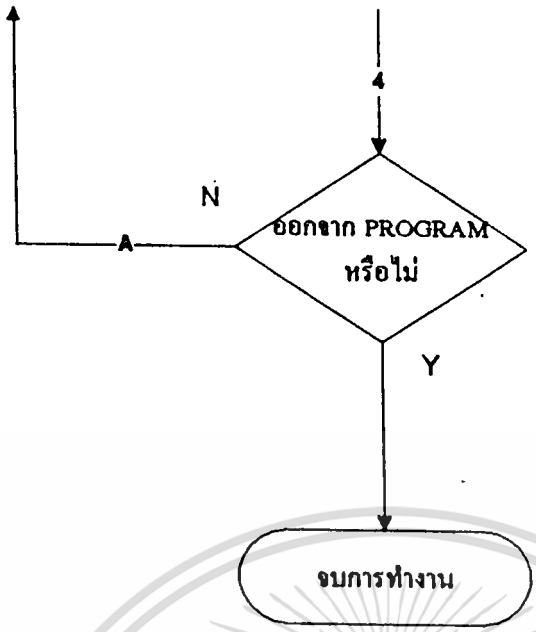
Step 4). หลังจาก step ที่ 3 ตัวเทอร์มินอลก็เริ่มทำการส่งข้อมูลออกมาทางขาสัญญาณ Transmitt (TXD) เข้าไปยังตัวโมเด็ม ขณะที่ตัวเทอร์มินอลทำการส่งข้อมูลอยู่นั้น ขาสัญญาณ RTS จะอยู่ในสภาวะ "High" เป็นเหตุให้ตัวโมเด็มไม่ส่งสัญญาณเข้ามาทางขาสัญญาณ CTS ที่ตัวเทอร์มินอลและหยุดการส่งข้อมูล (ถ้าตัวโมเด็มส่งสัญญาณมายังขาสัญญาณ CTS ที่ตัวเทอร์มินอลขณะนั้นตัวโมเด็มจะไม่ทำการส่งข้อมูลออกไปทางสายโทรศัพท์) การตรวจสอบลักษณะแบบนี้จะเกิดขึ้นเหมือนกันกับอุปกรณ์ปลายทางอีกฝั่งหนึ่ง จุดที่สำคัญของ step ที่กล่าวมาข้างต้นนี้เป็นการเซ็ตของกลุ่มสัญญาณตรวจสอบความพร้อมของการส่งข้อมูล เป็นการส่งระหว่างตัว DTE และตัว DCE (ซึ่งในที่นี้หมายถึงตัวโมเด็ม

FLOWCHRT อธิบายการทำงาน PROGRAM









บทที่ 5

การทดลองและผลการทดลอง

1. ส่วน P.L.C.T. UNIT

การทดลองในส่วนนี้ คือ ใช้ไอซิเบอร์ 1893 เป็นหลักในการส่ง-รับข้อมูลผ่านสายเอซีไลน์ แทนการต่อวงจรแบบแยกทีละส่วนซึ่งทำให้วงจรรวมมีขนาดเล็กลงอย่างมากและวงจรที่ได้สามารถทำงานได้อย่างมีประสิทธิภาพดีกว่าด้วย

ในการทำการปรับแต่งวงจรเริ่มจากทำการต่อขา 5 เข้ากับ +18 โวลท์ซึ่งเป็นไฟที่ใช้สำหรับเลี้ยงวงจร โดยที่ขา 5 ใช้สำหรับการเลือกว่าจะให้ไอซีทำงานอยู่ในโหมดของส่งหรือรับข้อมูล ถ้าต้องการส่งข้อมูลขา 5 จะต้องเป็น +18 โวลท์ ส่วนถ้าต้องการใช้รับข้อมูลขา 5 จะต้องต่อลงกราวน์ เมื่อทำการต่อขา 5 เข้ากับไฟเลี้ยงวงจรเรียบร้อยแล้ว ขณะนี้ไอซีจะถูกเซ็ทให้ทำงานอยู่ในโหมดส่งข้อมูล จากนั้นทำการต่อขา 12 ซึ่งเป็นขาที่ใช้ในการรับข้อมูลที่ได้ทำการดิโคดออกมาจากเอซีไลน์แล้วลงกราวน์และก็ทำการต่อขา 17 ซึ่งเป็นขาที่ใช้สำหรับส่งข้อมูลลงเอซีไลน์ลงกราวน์เช่นกัน ในสภาวะนี้ก็จะทำให้ไอซีผลิตความถี่พาหะที่มีลักษณะเป็น SINE WAVE ออกมาที่ขา 10 ซึ่งเป็นขาที่ใช้สำหรับทำหน้าที่เป็นพอร์ทในการส่ง-รับข้อมูลผ่านเอซีไลน์ โดยมีค่าของความถี่อยู่ในช่วง 100 กิโลเฮิร์ท บวกลบ 25 เปอร์เซ็นต์ ต่อไปก็ทำการปรับความถี่พาหะให้มีค่าประมาณ 125 กิโลเฮิร์ทได้จากการปรับโวลุ่มที่ขา 18 เมื่อปรับความถี่ได้แล้ว จากนั้นทำการปรับระดับแรงดันของความถี่พาหะให้มีค่ามากที่สุดจากการจูนที่วงจรแทงค์ที่ต่อกับขา 10 ระดับแรงดันที่ได้จะมีค่าประมาณอยู่ในช่วงบวกลบ 12 โวลท์ เป็นอันว่าขณะนี้เราก็จะได้วงจร P.L.C.T. ที่พร้อมจะสามารถใช้ในการส่งและรับข้อมูลผ่านเอซีไลน์แล้ว

ขั้นตอนต่อไปคือทำการปรับแต่งวงจรจูนที่ต่ออยู่กับขา 10 ของไอซีโดยที่วงจรนี้จะทำหน้าที่จูนให้ความถี่พาหะที่ใช้ในการส่งและรับผ่านเข้าออกกับเอซีไลน์ได้โดยสะดวก ส่วนความถี่อื่นๆที่ไม่ไม่ใช่ความถี่พาหะจะถูกลดทอนลงหรือกำจัดทิ้งไปโดยเฉพาะความถี่และแรงดันของไฟเอซีไลน์เนื่อป้องกันไม่ให้วงจรเสียหาย

จากบทที่ 2 ที่ความถี่พาหะเท่ากับ 125 กิโลเฮิร์ท เราสามารถที่จะนำไปคำนวนหาค่าของอินดัคแต้นซ์ของคอยล์ที่ด้าน PRIMARY และ SECONDARY ของวงจรจูนได้คือ L_1 = 49.0 uH และ L_2 = 0.98 uH ส่วนค่าของ C_2 = 33 nF และ C_c = 0.3 uF ซึ่งจะเห็นว่าค่าของคาปาซิเตอร์สามารถพอหาซื้อได้ง่าย แต่ส่วนที่เป็นปัญหาก็คือ ค่าอินดัคแต้นซ์ของวงจรจูนซึ่งหาซื้อตามท้องตลาดได้ลำบากที่จะให้ได้ค่าตรงกับที่ได้คำนวณไว้เพราะมิขายอยู่ในต่างประเทศนั่นเอง ดังนั้นจึงเกิดทางเลือกขึ้นมา 2 ทางคือ ทำการซื้อลวด และ แกนเฟอร์ไร้ท์ มาทำการพันเอง หรือ ลองทำการหาซื้อ I.F. TRANSFORMER 455 กิโลเฮิร์ท ที่ใช้กับวิทยุเอเอ็ม ซึ่งมีอยู่หลายสี เช่น สีดำ , สีเหลือง , สีขาว , สีแดง มาทดลองใช้ดู จากผลการทดลองพบว่า ถ้าเลือกใช้กรณีแรก ในทางปฏิบัติจะทำได้ยากเนื่องจาก เกิดความยุ่งยากในการพัน เมื่อพันเสร็จแล้วค่าที่ได้ก็อาจจะมีค่าผิดพลาดไปทำให้ไม่สามารถปรับเปลี่ยนได้โดยสะดวก ดังนั้นการปรับแต่งวงจรจึงทำได้ลำบาก เมื่อทำการเปลี่ยนมาใช้ I.F. TRANSFORMER แล้วปรากฏว่า I.F. กระป๋องสีดำจะมีค่าอินดัคแต้นซ์ที่ใกล้เคียงกับที่ได้คำนวณไว้มากกว่า I.F. TRANSFORMER กระป๋องสีอื่นๆมาก โดยที่ค่าอินดัคแต้นซ์ทางด้านไพรมารี่จะมีค่าประมาณ 50 uH ส่วนทางด้านเซกกันดารี่จะมีค่าอินดัคแต้นซ์ประมาณ 15 uH ซึ่งผิดจากที่คำนวนไว้ แต่ก็ยังมีข้อดีที่สามารถทำการปรับเปลี่ยนค่าอินดัคแต้นซ์ได้โดยสะดวกกว่าจากการจูนได้ที่ตัวกระป๋อง ซึ่งสามารถจูนให้มีระดับแรงดันของความถี่พาหะมีค่ามากสุดที่บวกลบ 12 โวลท์ ถ้าต้องการขนาดแรงดันที่มากกว่านั้นสามารถทำได้โดยทำการปรับเปลี่ยนเบอร์ของ I.F. TRANSFORMER ใหม่

ขั้นตอนต่อไป คือได้ทำการทดลองต่อเป็นระบบเตือนภัยที่สมบูรณ์ ปรากฏว่ามีปัญหาเกิดขึ้นที่เอ้าท์พุทขา 12 ของไอซีคือเอ้าท์พุทในสภาวะปกติที่ยังไม่มีการส่งข้อมูลผ่านเอซีไลน์ เอ้าท์พุทที่ได้มีค่าไม่คงที่สม่ำเสมอมีการเปลี่ยนแปลงไปตลอดเวลาทำให้เมื่อนำเอาเอ้าท์พุทที่ขา 12 นี้ไปใส่เข้าคอมพิวเตอร์เพื่อใช้ในการแสดงผลทำให้การแสดงผลเกิดความผิดพลาดขึ้นมา เมื่อทำการทดลองหาสาเหตุอยู่นานจึงรู้ว่าเกิดจากการที่ไม่มีความถี่พาหะอยู่ในเอซีไลน์นั่นเอง จากนั้นจึงได้ทำการใส่ความถี่พาหะให้ STAND BY อยู่ในเอซีไลน์ ผลปรากฏว่า สามารทำให้เอ้าท์พุทที่ขา 12 ของไอซีคงที่ขึ้นมาได้และเมื่อลองทำการส่งรับข้อมูลดูก็ยังไม่สามารถทำงานได้ ซึ่งจะต้องมีการปรับแต่งความถี่พาหะที่ใช้ในการ STAND BY ให้มีค่าอย่างเหมาะสมมากๆด้วยการปรับโวลุ่มที่ขา 18 อย่างละเอียดเสียก่อน เมื่อปรับโวลุ่มจนได้ที่ดีแล้วก็ได้ทำการทดลองส่งและรับข้อมูลอีกครั้งหนึ่งก็ปรากฏว่า ระบบเตือนภัยนี้สามารถทำงานได้แล้ว

เมื่อระบบเตือนภัยทำงานได้ถูกต้องแล้วแต่ยังเป็นการรับส่งข้อมูลในระยะใกล้ๆอยู่ประมาณสองถึงสามเมตร จากนั้นได้ทำการทดลองรับส่งข้อมูลผ่านเอซีไลน์ในระยะที่ไกลออกไปอีกประมาณหนึ่งร้อยเมตรโดยประมาณผลที่ได้ก็ยังสามารถรับส่งข้อมูลได้อยู่ ในการที่จะทำให้ระบบเตือนภัยสามารถทำงานในระยะที่ไกลออกไปได้อีก POWER ที่ใช้ในการส่งข้อมูลจะเป็นสิ่งที่จะต้องคำนึงถึงโดยมีสิ่งที่จะต้องพิจารณาอยู่สองประการคือ ต้องมีการเพิ่ม GAIN ของ OUTPUT AMP. ภายในตัวไอซีให้มีค่าสูงขึ้นโดยต้องมีการต่อวงจรเพิ่มที่ขา 8 และ 9 ของไอซีซึ่งวงจรนี้มีหน้าที่ในการ BOOST ให้กับ OUTPUT AMP. ในระบบเตือนภัยนี้ไม่ได้มีการ BOOST OUTPUT AMP. ข้อมูลก็สามารถเดินทางได้ไม่ไกลมาก ส่วนอีกกรณีที่จะต้องมีการพิจารณาคือ ขนาดของตัว I.F. TRANSFORMER ซึ่งจะต้องมีขนาดใหญ่ตามไปด้วยเมื่อต้องการ POWER ที่ใช้ในการส่งมากขึ้น ทำให้ไม่สะดวกที่จะนำมาใช้งานในทางปฏิบัติ ดังนั้นในระบบเตือนภัยนี้จึงไม่ได้ใช้จึงทำให้ POWER ที่ได้มีค่าประมาณ 0.2 W ซึ่งไม่สูงมากนัก นอกจากนั้นยังจะมีปัญหาที่เกิดจาก NOISE อีกคือ กรณีที่ข้อมูลผ่านไปในเอซีไลน์จริงๆที่ไม่ได้ใช้สายเอซีไลน์ที่ซื้อมาทดลอง ข้อมูลจะผ่าน LOAD มากมายที่ผลิต NOISE ออกมามีความถี่อย่าง RANDOM ถ้าเป็น LOAD พวก SWITCHING อาจจะผลิต NOISE ที่มีความถี่ใกล้เคียงกับความถี่พาหะทำให้เข้าไปรบกวนระบบเตือนภัยให้ทำงานผิดพลาดไปได้

2.ส่วน DETECTOR UNIT

ในส่วนจะประกอบด้วย 2 ส่วนหลักคือ ส่วนของการ INTERFACE กับ RS232 และส่วนของการประมวลผล

การทดลองการ INTERFACE RS232 เพื่อต่อเชื่อมเข้ากับ ส่วน P.L.C.T. ทำการแปลงสัญญาณจาก RS232 ให้เป็น TTL ในการทดลองตอนแรกได้ใช้ IC เบอร์ MAX232ซึ่งเป็น IC สำเร็จรูปที่ใช้ในการแปลงจาก RS232 เป็น TTL โดยเฉพาะการใช้งานของ IC ก็ต่อร่วมกับ CAPACITOR ภายนอกอีกเพียง 3-4 ตัว ก็สามารถใช้งานได้ ภายในตัว IC จะมีส่วนของ INTERFACE ใช้ได้ 2 CHANNEL แต่ใน DETECTOR UNIT จะใช้เพียง CHANNEL เดียว จึงเหลือเปล่าอีกหนึ่ง CHANNEL จากนั้นจึงได้เปลี่ยนมาใช้ทรานซิสเตอร์แทน ซึ่งใช้ทรานซิสเตอร์เพียง 2 ตัว ต่อร่วมกับตัวต้านทานอีก 4-5 ตัวก็ใช้งานได้ดีเช่นกัน

การทดลองส่วนของการประมวลผล ได้เลือกใช้ไมโครคอนโทรลเลอร์ในตระกูล MCS-51 เบอร์ 8031 มาใช้งานในส่วนนี้เพราะภายในตัว IC ประกอบด้วย ตัวควบคุมการสื่อสารแบบอนุกรม , หน่วยความจำภายใน และ PORT ขนาน 4 ตัว จึงทำให้สะดวกในการออกแบบใช้งาน ในส่วนของโปรแกรมควบคุมการทำงานจะถูกเก็บอยู่ใน EPROM ภายนอก และจะมี DIP SW. ใช้ในการเซ็ทค่า ID CODE ในการทดลองเพื่อเขียนโปรแกรมควบคุมการทำงานของเครื่องจำเป็นที่ต้องใช้ RAMPACK ต่อทดลองก่อนที่ BURN ลง EPROM จริง ๆ เพราะในการทดลองจำเป็นที่จะต้องมีการแก้ไขปรับปรุงโปรแกรมอยู่หลายครั้ง จนในที่สุดได้โปรแกรมที่สามารถใช้งานได้จริงจึงนำมาBURN ลงใน EPROM ในส่วนของ DETECTOR UNIT จะต้องมีการสร้างสัญญาณ CONTROL TX/RX เพื่อป้อนให้กับภาค P.L.C.T. โดยการใช้ J-K FLIP-FLOP เบอร์ 74112 สัญญาณที่ใช้ในการ TRIG FLIP-FLOP จะได้มาจาก IC เบอร์ 74LS138 ในการทดลองตอนแรกใช้ FLIP-FLOP เบอร์ 74LS112 แต่ไม่สามารถทำงานได้ จากการวิเคราะห์สาเหตุ พบว่าสัญญาณ

TRIG ที่ป้อนให้ กับ FLIP-FLOP มีขอบของสัญญาณเร็วเกินไป 74LS112 ทำงานไม่ทัน จึงต้องใช้ IC ที่ทำงานได้เร็วกว่าคือเบอร์ 74ALS112 หรือ 74F112 หลังจากที่ใช้ IC 2 เบอร์หลังใส่แทน IC 74LS112 ก็สามารถทำงานได้ มีสัญญาณ CONTROL TX/RX ไปป้อนให้กับภาค P.L.C.T.

ผลการทดลองเมื่อนำส่วน DETECTOR UNIT ไปต่อทดลองโดยส่งสัญญาณจำลองการทดสอบจากเครื่อง PC ใช้โปรแกรม PROCOMM เป็นตัวสร้างสัญญาณ SERIAL DATA ไปให้กับ DETECTOR UNIT และรับข้อมูลที่จาก DETECTOR UNIT มาแสดงผล ผลการทดลองเมื่อส่งสัญญาณ SERIAL DATA ไปตรงกับ DATA ที่เซ็ทไว้โดย DIP SW. บน DETECTOR UNIT ยกตัวอย่างเช่น DIP SW. เซ็ทค่าเป็น 41H เมื่อส่ง SERIAL DATA จาก PC ออกมาเป็น 41Hท ก็จะมีสัญญาณตอบกลับจาก DETECTOR UNIT เป็น 41H ไปปรากฏบนจอPC เป็น A (ASCII A = 41H) แต่ถ้าหากส่งข้อมูล SERIAL DATA เป็นค่าอื่นที่ไม่ใช่ 41H ก็จะไม่มีอะไรปรากฏบนหน้าจอ PC แสดงว่าการทำงานของ DETECTOR UNITถูกต้อง

จากนั้นจึงนำไปต่อร่วมกับภาค P.L.C.T. เพื่อทดสอบการทำงานจริง ๆ ตัว DETECTOR UNIT สามารถรับและส่งข้อมูลตอบกลับได้ แต่ข้อมูลที่ตอบกลับไม่สามารถผ่าน P.L.C.T. ออกไป AC LINE ได้หรือถ้าได้ข้อมูลก็จะผิดเพี้ยน จากการวิเคราะห์หาสาเหตุพบว่าเมื่อส่งสัญญาณ CONTROL TX/RX ไปบอกให้ P.L.C.T. รับรู้ว่าขณะนี้ภาค DETECTOR UNIT จะส่งข้อมูลตอบกลับ แล้วจะต้องรอหน่วงเวลาประมาณ 15 mS จึงจะส่ง DATA ออกไปได้หลังจากที่ได้หน่วงเวลาแล้ว ข้อมูลที่ตอบกลับไปก็จะมีความถูกต้อง การทำงานของ DETECTOR UNIT จึงสมบูรณ์

3. ส่วนของ CONTROL UNIT

ลำดับขั้นที่ 1.การเขียน FLOWCHART เพื่อวางระบบให้กับ PROGRAM

ได้ทำการคิดและเขียน FLOWCHAT ตามรูปแบบการทำงานของระบบ HARDWEAR ที่ได้ทำขึ้น โดยลักษณะการทำงานตามลำดับขั้นจะเป็นดังนี้

1.ให้ COMPUTER แสดงรูปห้องบนหน้าจอ

2.ส่งสัญญาณเลขห้องไป SCAN ในตำแหน่งห้องแรก

3.รอรับสัญญาณที่เป็นเลขห้องนั้นกลับมาพร้อมกับสัญญาณของตัว SENSER

4.แสดงตำแหน่งห้องที่รับสัญญาณเข้ามาถูกต้อง

5.แสดงการเกิด ERROR ของ SENSER ในห้องนั้น

6.ให้ทำการ SCAN ต่อตามหัวข้อที่ 2 จนครบทุกห้อง

7.ให้แสดง ERROR ทั้งหมดที่ตรวจจับได้และพร้อมกับที่จะสามารถดูตำแหน่งที่ เกิด ERROR นั้นได้ในแต่ละห้อง

สรุป

ได้ทำการเขียน FLOWCHART ตามขั้นตอนที่ได้กำหนดไว้เป็นที่เรียนร้อย ดังแสดงในรูป FLOWCHART อธิบายการทำงานของ PROGRAM ALARM SYSTEM

ลำดับขั้นที่ 2.การเขียน SUB PROGRAM วาดรูปตำแหน่งของห้อง

ทำการเขียน PROGRAM โดยการศึกษาการเขียน PROGRAM ภาษา QUICKBASIC ซึ่งได้มีพื้นฐานการเขียน PROGRAM ภาษา BASIC อยู่บ้าง

สรุป

ได้ทำการเขียน PROGRAM ภาพรูปห้องที่ชื่อว่า MAKESCREEN และทดลอง RUN มีปัญหาทางด้านการใช้รูปแบบประโยคคำสั่ง ได้ทำการแก้ไขจน PROGRAM สามารถ RUN ได้

ลำดับขั้นที่ 3.การเขียน SUB PROGRAM แสดงผลการ SCAN CHECK ตำแหน่งห้องต่างๆ

ทำการเขียน PROGRAM สร้างภาพห้องที่แสดงว่าห้องนั้นเกิด ERROR โดยแสดงใน SUB PROGRAM ที่ ชื่อว่า LCHKROW และ LCHKCO

สรุป

ทำการทดลอง RUN PROGRAM ซึ่ง PROGRAM สามารถ RUN ได้ โดย PROGRAM จะแสดง ERROR ให้เห็นเป็นหลังสีแดงออกมา

ลำดับขั้นที่ 4.การเขียน SUB PROGRAM แสดงผลการ ERROR ในการตรวจจับของ SENSER ในตำแหน่งต่างๆ

ทำการเขียน PROGRAM สร้างภาพห้องเตี่ยวที่แสดงตำแหน่งของตัวตรวจจับ โดยอยู่ใน SUB PROGRAM ที่ชื่อว่า DIROOM และทำการเขียน PROGRAM CHECK ERROR ในตัว SENSER ต่างๆ ใน SUB PROGRAM ที่ชื่อว่า CONKEY

สรุป

ทำการทดลอง RUN PROGRAM ทั้งสอง โดยกำหนดค่า ERROR สมมุติให้กับ PROGRAM ซึ่ง PROGRAM สามารถ RUN ได้ปรกติ โดยตำแหน่งของ SENSER ที่ ERROR จะแสดงให้เห็นเป็นรูปของหน้าต่างและประตูสีแดง

ลำดับขั้นที่ 5.การเขียน SUB PROGRAM ส่งเสียงเตือนเมื่อเกิด ERROR

ทำการเขียน PROGRAM ส่งเสียงเตือน เมื่อเกิด ERROR ใน SUB PROGRAM ที่ชื่อว่า DELA

สรุป

ทำการทดลอง RUN PROGRAM ร่วมกับ SUB DIROOM สามารถ RUN ได้

ลำดับขั้นที่ 6.การเขียน SUB PROGRAM ติดต่อกับชุดตรวจจับ SENSER

ทำการเขียน SUB PROGRAM LCHKPOST1 เพื่อให้ติดต่อกับชุดตรวจจับ SENSER

สรุป

ทำการทดลอง RUN PROGRAM ปรากฏว่า PROGRAM มีปัญหาไม่สามารถ ส่งรับกับชุดตรวจได้ ได้ทำการทดสอบหาจุดผิดพลาด และพบว่าการใช้คำสั่งส่งออก COM PORT ไม่ถูกต้อง ทำให้ขนาดของข้อมูลที่ส่งออกไปไม่ถูกต้อง ทำการแก้ไขศึกษาหาวิธีใช้คำสั่งใหม่ พบว่า ฟังก์ชัน INT สามารถนำข้อมูลที่ถูกต้องออกจาก COM PORT ได้ และทำการทดลองใหม่อีกครั้ง RUN PROGRAM สามารถติดต่อกับชุดตรวจจับได้

ลำดับขั้นที่ 7.การเขียน SUB PROGRAM ติดต่อกับชุดตรวจจับโดยผ่านไป ในระบบชุดส่งสัญญาณผ่านทาง AC LINE

เขียนแก้ไข PROGRAM LCHKPOST เพื่อการทดลองติดต่อกับชุดตรวจจับโดยผ่านไป ในระบบชุดส่งสัญญาณผ่าน AC LINE

สรุป

จากการทดลอง ต้องทำการทดลองและแก้ไข ในส่วนของการหน่วงการรับส่งหลายครั้งด้วยกัน จึงสามารถรับส่งข้อมูลได้ถูกต้อง เพราะเกิดปัญญาณหาในเรื่องการ DELAY ของชุดส่งข้อมูลลงใน AC LINE

ลำดับขั้นที่ 8.การเขียน MAIN PROGRAM โดยรวมชุด SUB PROGRAM ต่างๆ ที่ได้เขียนไว้

ทำการเขียน MAIN PROGRAM และทำการรวมชุดของ SUB PROGRAM ต่างๆ ให้สามารถทำงานร่วมกันได้ทั้งระบบ

สรุป

ทำการทดลอง RUN TEST PROGRAM ทั้งระบบ และทำการแก้ไข PROGRAM อีกนิดหน่อย PROGRAM ก็สามารถทำงานได้

บทที่ 6

สรุปและวิจารณ์ผลการทดลอง

โครงงาน ALARM SYSTEM ที่ได้ทำขึ้นมานี้ จากจุดประสงค์ที่ต้องการคือให้สามารถเชื่อมต่อการส่งสัญญาณในแต่ละห้องโดยผ่านทางสาย AC LINE เพื่อลดการเดินสายส่งสัญญาณ และยังสะดวกในการติดตั้งด้วย ในทางปฏิบัติแล้วต้องการนำโครงงานนี้ไปใช้เป็นระบบเตือนภัยในอาคาร เช่น คอนโดมิเนียม อพาร์ตเมนท์ เป็นต้น

ในการส่งสัญญาณผ่าน AC LINE จะมีปัญหาที่สำคัญคือเรื่องของสัญญาณรบกวนที่ปนเข้ามากับสาย AC LINE ซึ่งเราไม่สามารถควบคุมได้ แต่จากการทดลองแล้วโอกาสที่สัญญาณรบกวนจะมีความถี่ ตรงกับความถี่ที่ใช้ในระบบ ALARM SYSTEM ค่อนข้างน้อย ฉะนั้นโอกาสที่จะเกิด ERROR จึงน้อยหรือแทบจะไม่มีเลย ปัญหาอีกอย่างหนึ่งคือ ในสภาวะปกติแล้ววงจรในส่วน P.L.C.T. จะถูกเซ็ทให้อยู่ในสภาวะรับทุกตัว ในขณะนี้เองสัญญาณรบกวนสามารถที่จะเข้ามาทางส่วน P.L.C.T. ได้ เกิดเป็นข้อมูลที่ไม่ต้องการขึ้น ปัญหานี้สามารถแก้ไขด้วยการสร้างสัญญาณความถี่ให้มีความถี่เดียวกับความถี่ที่ใช้ในระบบ ALARM SYSTEM ในโครงงานนี้คือ 125 KHz ส่งเข้าไปใน AC LINE ตลอดเวลา โดยให้แอมปลิจูดต่ำกว่าสัญญาณที่ใช้จริง ก็จะทำให้ปัญหานี้หมดไป สำหรับระยะทางในการส่งสัญญาณนั้น ในระยะทางที่ไกลมาก ๆ ประมาณมากกว่า 100 เมตรขึ้นไปอาจจะยังไม่ค่อยดีนัก จึงอาจนำไปใช้ได้กับอาคารที่มีขนาดไม่สูงมากนัก อุปกรณ์ที่ใช้ในการคัปปลิ้งสัญญาณลง AC LINE คือ TRANSFORMER ซึ่งต้องใช้แกนที่ใช้ได้กับความถี่สูง ๆ ไม่ต่ำกว่า 125 KHz นั้น ไม่สามารถหาได้ จึงต้องใช้สลักจุนของ IFT แทน ซึ่งมีขนาดเล็ก ด้วยเหตุนี้เองจึงเป็นไปได้ว่าหากสามารถหาแกนของ TRANSFORMER ที่มีขนาดใหญ่ได้ ก็จะทำให้การส่งไปได้ไกลขึ้น

ดังนั้นโครงงานนี้ก็ให้ผลที่น่าพอใจในส่วนหนึ่ง คือสามารถนำไปใช้งานในอาคารขนาดเล็ก ๆ ได้ ซึ่งก็ทำให้บรรลุถึงจุดประสงค์ของโครงงานนี้คือ การลดการเดินสายส่งสัญญาณและสะดวกในการติดตั้งระบบ

ภาคผนวก

```
;**********************************
;*                                *
;*          PROGRAM               *
;*            FOR                 *
;*      DETECTOR UNIT             *
;*                                *
;**********************************

;START PROGRAM

        ORG 0000H
        LJMP MAIN

        ORG 0100H

MAIN:   ORL PCON,#80H           ;SET DOUBLE FREQ.
        MOV TH1,#4DH            ;SET BAUD RATE = 300
        CLR RS0                 ;SET REGISTER BANK 0
        CLR RS1
        MOV R0,P1               ;CHECK ID CODE
        SETB ES                 ;ENABLE SERIAL
        SETB EA                 ;ENABLE INTERRUPT
        MOV SCON,#50H           ;SET SERIAL MODE 1
        ORL TMOD,#20H           ;TIMER AUTO-RELOAD
        SETB TR1                ;TURN TIMER 1 ON

;RECEIVE DATA

RX:     JNB RI,$                ;IF DATA IS RECEIVED
        CLR RI                  ;
        MOV A,R0                ;COMPARE DATA,ID CODE
        CJNE A,SBUF,RX          ;DATA=ID CODE


;TRANSMITT DATA

TX:     MOV DPTR,#8003H         ;SET TX
        MOVX A,@DPTR
        CALL DELAY
        MOV SBUF,R0             ;;SEND DATA OUT
        JNB TI,$
        CLR TI
        MOV DPTR,#4000H
        MOVX A,@DPTR            ;READ ALARM STATUS
        MOV SBUF,A
        JNB TI,$
        CLR TI
        MOV DPTR,#0A000H        ;CLEAR TX
        MOVX A,@DPTR
        SJMP RX
```

```
                ;DELAY DATA
DELAY:          MOV R5,#14H             ;DELAY ABOUT 15 mS
DELAY0:         MOV R4,#0C8H            ;BEFORE DATA IS SEND
DELAY1:         DJNZ R4,DELAY1
                DJNZ R5,DELAY0
                RET

                ;INTERRUPT ROUTINE

                ORG 0023H
SLI:            RETI
                END
```

PROGRAM
CONTROL ALARM SYSTEM

```
' $INCLUDE: 'QB.BI'
DECLARE SUB Lchkpostl (q%, n%, f1%, f2%, f3%, f4%)
DECLARE SUB LgetError (gerr%(), n%, e%, f1%, f2%, f3%, f4%)
DECLARE SUB ConKey (gerr%(), k%)
DECLARE SUB Diroom (gerr%(), e%)
'DECLARE SUB Lchkpost (gerr%(), p%, e%)
DECLARE SUB Lchkrow (u%, w%)
DECLARE SUB Dela ()
DECLARE SUB lchkco (u%, m%)
DECLARE SUB MakeScreen ()
DEFINT A-Z
DEFDBL S-T
DIM gerr(44, 5) AS INTEGER
SCREEN 1: COLOR 0, 0, 2
CLS

'room scan
h = 1
i = 2

LOCATE 3, 5
PRINT "WELCOM TO PROJECT ALARM SYSTEM"
LOCATE 25, 5
PRINT "Press any key to continue .... "
DO

a$ = INKEY$
LOOP UNTIL a$ >= CHR$(10) AND a$ <= CHR$(255)
CLS
```

```
'make screen
Lscan1:
CALL MakeScreen

Lscan:
a% = 0: b% = 0: n% = 0: e% = 0: a$ = " "
r% = 0:  p% = 0: g% = 0: u% = 0: w% = 0
m% = 0: z% = 0: y% = 0: f1% = 0: f2% = 0
f3% = 0: f4% = 0: q% = 0
'scan error
DO

FOR a% = 1 TO 4
   'chkrow
   SELECT CASE a%
         CASE 4
            r% = 11
         CASE 3
            r% = 41
         CASE 2
            r% = 95
         CASE 1
            r% = 125

   END SELECT

   FOR b% = 14 TO 274 STEP 26
      n% = n% + 1                'No. room
      IF b% = 144 AND (r% = 125 OR r% = 41) THEN
            n% = n% - 1
      ELSE
            PAINT (b, r), 1, 3'cHR$(&HAA)
      IF h = n OR i = n THEN
            CALL Lchkpost1(q, n%, f1%, f2%, f3%, f4%)
      ELSE
            q = 0
```

```
    END IF
    s = VAL(MID$(TIME$, 7, 2))
    f = INT(RND(8) * 9) + 1              'error room
        IF q = 1 THEN
           e = e + 1
           gerr(0, 0) = e
           gerr(e, 0) = n                'get  no. room
           p = INT(RND(12) * 4) + 1      'error post

           CALL LgetError(gerr(), n, e, f1, f2, f3, f4)
        'CALL Lchkpost(gerr(), p, e)     'call test
           CALL Dela
           SOUND 2000, 2
        ELSE
           'gerr(E, 0) = 0
        END IF
        CALL Dela
        PAINT (b, r), 0, 3
        IF e > 0 THEN
           FOR g = 1 TO e
                u = gerr(g, 0)      'no romm
                CALL Lchkrow(u, w)
                CALL lchkco(u, m)
                PAINT (m + 1, w + 1), 2, 3
                CALL Dela
           NEXT g

           FOR g = 1 TO e
                u = gerr(g, 0)      'no romm
                CALL Lchkrow(u, w)
                CALL lchkco(u, m)
                PAINT (m + 1, w + 1), 0, 3
           NEXT g
        END IF

    END IF
NEXT b
```

```
NEXT a
n = 0
LOOP UNTIL q = 1

Lerror:
IF h = 1 THEN          'jump  work
  CLS
  CALL MakeScreen
  h = 0
END IF
DO
   FOR g = 1 TO e
       'u = gerr(g, 0)      'no romm
        CALL Lchkrow(u, w)
        CALL lchkco(u, m)
        PAINT (m + 1, w + 1), 2, 3
    NEXT g
    CALL Dela
    CALL Dela
    SOUND 2000, 1
    FOR g = 1 TO e
        u = gerr(g, 0)      'no romm
        CALL Lchkrow(u, w)
        CALL lchkco(u, m)
        PAINT (m + 1, w + 1), 0, 3
    NEXT g
    CALL Dela
    CALL Dela
 Lease:
    LOCATE 23, 15
    PRINT "Zoom in Y/N"
    a$ = INKEY$
    SELECT CASE a$
          CASE "n", "N"
             z = 0
             EXIT DO
          CASE "Y", "y"
```

```
            z = 1
            EXIT DO
    END SELECT

LOOP WHILE e > 0

a$ = " "
IF z = 0 THEN       'clear data
    FOR x = 0 TO 44
        FOR y = 0 TO 5
            gem(x, y) = 0
        NEXT
    NEXT
    GOTO Lscan
END IF

CALL Diroom(gem(), e)

'ClearMem:

LclearMem:
VIEW (1, 18)-(317, 186), , 0
CLS
LOCATE 23, 15
INPUT "Zoon out Y/N ", a$
SELECT CASE a$
        CASE "y", "Y"
            CLS
            h = 1
            GOTO Lerror
        CASE "n", "N"
            CLS
            GOTO LtoN:
        CASE ELSE
            BEEP
            GOTO LclearMem
END SELECT
```

```
LtoN:
  'clear data
FOR x = 0 TO 44
    FOR y = 0 TO 5
      .     gerr(x, y) = 0
    NEXT
NEXT

LTo:
a$ = " "
CLS
LOCATE 22, 15
INPUT "Exit Program Y/N ", a$
SELECT CASE a$
      CASE "n", "N"
            CLS
            GOTO Lscan1
      CASE "Y", "y"
            END
      CASE ELSE
            CLS
            GOTO LTo
  END SELECT
  END

  DEFINT S
  SUB ConKey (gerr(), k)
  a$ = ""
    k = 0
    LOCATE 5, 15
    PRINT "Room Error No. "
    LOCATE 7, 5
    FOR a = 1 TO gerr(0, 0)
       PRINT gerr(a, 0);
    NEXT
    PRINT
```

```
y = CSRLIN
x = POS(0)
KeyLoop:
LOCATE y + 2, 6
INPUT "Seclect Room Error No. ", a$
z = VAL(a$)
IF z = 0 THEN
      BEEP
      LOCATE y + 2, 6
      PRINT "Seclect Room Error No.      "
      GOTO KeyLoop
END IF
FOR a = 1 TO gerr(0, 0)
   IF a = gerr(0, 0) AND gerr(gerr(0, 0), 0) <> z THEN
        BEEP
        LOCATE y + 2, 6
        PRINT "Seclect Room Error No.      "
        GOTO KeyLoop
     END IF
     IF gerr(a, 0) = z THEN
          k = a
          a = gerr(0, 0)
     END IF
  NEXT

END SUB

SUB Dela
FOR x = 1 TO 7000: NEXT
END SUB

SUB Diroom (gerr(), e)
a$ = ""
ff% = 0

CLS
VIEW (1, 1)-(313, 190), , 0
```

```
SCREEN 1: COLOR 0, 0, 2: CLS
room$ = "s4r110d601110u60"
doorex$ = "s4r39d44139u44"
doorin$ = "s4r37d42137u42"
winex$ = "s4r23d24123u24"
winin$ = "s4r21d22121u22"
VIEW (1, 18)-(317, 186), , 2

DO
m = 0
IF e <> 1 THEN
   CALL ConKey(gerr(), k)
   CLS
END IF
a = gerr(k, 1)
b = gerr(k, 2)
c = gerr(k, 3)
d = gerr(k, 4)
DO
   m = m + 1
   IF m = 1 THEN ff = 2
   IF m = 2 THEN ff = 0
    'room background
    PSET (75, 35)
    DRAW "c3"
    DRAW "x" + VARPTR$(room$)

     'DOOR BACKGROUND
     PSET (75, 51)
     DRAW "c3"
     DRAW "x" + VARPTR$(doorex$)

     IF (k >= 1 AND k <= 10) OR (k >= 22 AND k <= 31) THEN
       T = d
     ELSE
       T = a
     END IF
```

```
IF T = 1 THEN
  PAINT (77, 53), ff, 3
  PAINT (113, 52), ff, 3
END IF

'WINDOWS  BACKGROUND #1
IF (k >= 11 AND k <= 21) OR (k >= 32 AND k <= 42) THEN
      PSET (114, 51)
      DRAW "c3"
      DRAW "x" + VARPTR$(winex$)

      IF b = 1 THEN
        PAINT (116, 53), ff, 3
        PAINT (118, 73), ff, 3
        PAINT (135, 73), ff, 3
      END IF

      PSET (137, 51)
      DRAW "c3"
      DRAW "x" + VARPTR$(winex$)
      IF c = 1 THEN
        PAINT (139, 53), ff, 3
        PAINT (139, 73), ff, 3
      END IF
  END IF

  LINE (75, 35)-(130, 64), 3
  LINE (185, 35)-(239, 64), 3
  LINE (75, 95)-(130, 124), 3
  LINE (185, 95)-(239, 124), 3

   'fore room
   PSET (129, 64)
   DRAW "c3"
   DRAW "x" + VARPTR$(room$)

   ' door  FOREGROUND
```

```
PSET (200, 80)
DRAW "C3"
DRAW "x" + VARPTR$(doorex$)
IF (k >= 1 AND k <= 10) OR (k >= 22 AND k <= 31) THEN
  w = a
ELSE
  w = d
END IF
IF w = 1 THEN
  PAINT (202, 82), ff, 3
  PAINT (202, 122), ff, 3
  IF m = 2 THEN
    FOR z = 1 TO 9000: NEXT
  END IF
END IF

'windows FOREGROUND #2
IF (k >= 1 AND k <= 10) OR (k >= 22 AND k <= 31) THEN
    PSET (154, 80)
    DRAW "c3"
    DRAW "x" + VARPTR$(winex$)
 IF c = 1 THEN
      PAINT (156, 82), ff, 3
      PAINT (156, 98), ff, 3
    END IF

    PSET (177, 80)
    DRAW "c3"
    DRAW "x" + VARPTR$(winex$)

  IF b = 1 THEN
      PAINT (179, 82), ff, 3
      PAINT (179, 97), ff, 3
      PAINT (198, 82), ff, 3
    END IF
  END IF
```

```
CALL Dela
SOUND 2000, 2

IF m = 2 THEN m = 0
LOCATE 22, 13
PRINT "Zoon in room  Y/n "
a$ = INKEY$

IF a$ = "n" OR a$ = "N" THEN EXIT DO
LOOP UNTIL a$ = "Y" OR a$ = "y"
m = 0
CLS
LOOP UNTIL a$ = "n" OR a$ = "N"
END SUB

SUB lchkco (u%, m%)
SELECT CASE u
      CASE 1, 11, 22, 32
      m = 13
      CASE 2, 12, 23, 33
      m = 39
      CASE 3, 13, 24, 34
      m = 65
      CASE 4, 14, 25, 35
      m = 91
      CASE 5, 15, 26, 36
      m = 117
      CASE 16, 37
      m = 143
      CASE 6, 17, 27, 38
      m = 169
      CASE 7, 18, 28, 39
      m = 195
      CASE 8, 19, 29, 40
      m = 221
      CASE 9, 20, 30, 41
```

```
        m = 247
        CASE 10, 21, 31, 42
        m = 273
END SELECT

END SUB

SUB Lchkpost (gerr(), p, e)
SELECT CASE p
        CASE 1
        gerr(e, INT(RND(30) * 4) + 1) = 1

        CASE 2
      p = INT(RND(30) * 4) + 1
      gerr(e, p) = 1
lchkpot:
      j = INT(RND(30) * 4) + 1
      IF j = p THEN GOTO lchkpot
      gerr(e, p) = 1

        CASE 3
      p = INT(RND(30) * 4) + 1
      FOR h = 1 TO 4
      IF h = p THEN
            h = h + 1
      ELSE
            gerr(e, h) = 1
      END IF
      NEXT

        CASE 4
      FOR h = 1 TO 4
      gerr(e, h) = 1
      NEXT
  END SELECT
  END SUB
```

```
DEFSNG A-Z
SUB Lchkpost1 (q%, n%, f1%, f2%, f3%, f4%)
DEFINT A-Z
DEFDBL S-T

DIM inreg AS RegType
DIM outreg  AS RegType

'set initialize communications port  #1
inreg.ax = &H43
inreg.dx = &H0
CALL INTERRUPT(&H14, inreg, outreg)

'set initialize communications port  #2
inreg.ax = &H43
inreg.dx = &H1
CALL INTERRUPT(&H14, inreg, outreg)
'LOCATE 2, 15
'PRINT "Program test send com1 and control com2"
DO
'LOCATE 5, 5

'INPUT "Data send to com1 (&Hnn or &H999 to exit prgram ) ", a
b = &H100 + n
'LOCATE 7, 20
'PRINT "                "
'F a = &H999 THEN EXIT DO
'LOCATE 7, 5
'INPUT "Delay time com1 or com2 (1/100 Seconds) ", d

'Write character to communications port #2
inreg.ax = &H1FF
inreg.dx = &H1
CALL INTERRUPT(&H14, inreg, outreg)

'LOCATE 7, 10
'PRINT "Rescive com2 :", HEX$(outreg.ax)
```

```
s = TIMER
DO
  T = TIMER - s
LOOP UNTIL T >= .5

'Write No room to communications port #1
inreg.ax = b
inreg.dx = &H0
CALL INTERRUPT(&H14, inreg, outreg)

s# = TIMER
DO
  T# = TIMER - s#
LOOP UNTIL T# >= .5

'Read No room to communications port #1
inreg.ax = &H200
inreg.dx = &H0
CALL INTERRUPT(&H14, inreg, outreg)
'LOCATE 9, 10
m = outreg.ax
'PRINT "Rescive com1 B1 ", HEX$(outreg.ax)

'Read Error  to communications port #1
inreg.ax = &H200
inreg.dx = &H0
CALL INTERRUPT(&H14, inreg, outreg)
'LOCATE 12, 10
e = outreg.ax
'PRINT "rescive com1 B2", HEX$(outreg.ax)

inreg.ax = &H200
inreg.dx = &H0
CALL INTERRUPT(&H14, inreg, outreg)
LOCATE 15, 10
'PRINT "re com1 B3", HEX$(outreg.ax)
```

```
inreg.ax = &H200
inreg.dx = &H0
CALL INTERRUPT(&H14, inreg, outreg)
'LOCATE 17, 10
'PRINT "b4", HEX$(outreg.ax)

LOOP UNTIL m <> n

bb1$ = LTRIM$(RTRIM$(STR$(&H800 - e)))
b1$ = ""
SELECT CASE bb1$
      CASE "0"
           b1$ = "0000"
     f = 0
      CASE "1"
           b1$ = "0001"
     f = 1
      CASE "2"
           b1$ = "0010"
     f = 1
      CASE "3"
           b1$ = "0011"
     f = 1
      CASE "4"
           b1$ = "0100"
     f = 1
      CASE "5"
           b1$ = "0101"
     f = 1
      CASE "6"
           b1$ = "0110"
     f = 1
      CASE "7"
           b1$ = "0111"
     f = 1
```

```
    CASE "8"
        b1$ = "1000"
    f = 1
    CASE "9"
        b1$ = "1001"
   f = 1
    CASE "10"
        b1$ = "1010"
   f = 1
    CASE "11"
       b1$ = "1011"
  f = 1
       CASE "12"
              b1$ = "1100"
       f = 1
       CASE "13"
       b1$ = "1101"
       f = 1
       CASE "14"
             b1$ = "1110"
       f = 1
       CASE "15"
             b1$ = "1111"
       f = 1
END SELECT

IF VAL(MID$(b1$, 1, 1)) = 1 THEN
   f1 = 1
ELSE
   f1 = 0
END IF
IF VAL(MID$(b1$, 2, 1)) = 1 THEN
   f2 = 1
ELSE
   f2 = 0
END IF
IF VAL(MID$(b1$, 3, 1)) = 1 THEN
```

```
    f3 = 1
ELSE
    f3 = 0
END IF
IF VAL(MID$(b1$, 4, 1)) = 1 THEN
    f4 = 1
ELSE
    f4 = 0
END IF

IF f1 = 1 OR f2 = 1 OR f3 = 1 OR f4 = 1 THEN q = 1

END SUB

DEFINT S

SUB Lchkrow (u%, w%)
SELECT CASE u
    CASE 32 TO 42
    w = 10
    CASE 22 TO 31
    w = 40

    CASE 11 TO 21
    w = 94
    CASE 1 TO 10
    w = 124

END SELECT

END SUB

DEFDBL S
' get to Dim gerr()
SUB LgetError (gerr(), n%, e%, f1%, f2%, f3%, f4%)
gerr(e, 0) = n
```

```
IF f1 = 1 THEN
  gen(e, 1) = 1
ELSE
  gen(e, 1) = 0
END IF

IF f2 = 1 THEN
  gen(e, 2) = 1
ELSE
  gen(e, 2) = 0
END IF

IF f3 = 1 THEN
  gen(e, 3) = 1
ELSE
  gen(e, 3) = 0
END IF

IF f4 = 1 THEN
  gen(e, 4) = 1
ELSE
  gen(e, 4) = 0
END IF

END SUB

DEFINT S
SUB MakeScreen
DIM GROOM(100) AS DOUBLE      'GET ROOM
SCREEN 1: COLOR 0, 0, 2
'make screen
SETBOX$ = "a0s25c3"
room$ = "r4d4l4u4"
VIEW (1, 1)-(313, 190), , 1
PSET (5, 5)
DRAW "C2"
DRAW "r301d82l301u82"
```

```
LOCATE 10, 17
PRINT "FLOOR 2 "
PSET (5, 90)
DRAW "C2"
DRAW "r301d821301u82"
LOCATE 21, 17
PRINT "FLOOR 1"

PSET (13, 10)
DRAW "X" + VARPTR$(SETBOX$)
DRAW "X" + VARPTR$(room$)

GET (13, 10)-(39, 35), GROOM
FOR b = 39 TO 273 STEP 26
   PUT (b, 10), GROOM
NEXT b
FOR b = 13 TO 273 STEP 26
   IF a = 5 THEN b = b + 26
   PUT (b, 40), GROOM
   PUT (b, 124), GROOM
   a = a + 1
NEXT b
FOR b = 13 TO 273 STEP 26
   PUT (b, 94), GROOM
NEXT b

END SUB
```

หนังสืออ้างอิง

(1) NATIONAL SEMICONDUCTOR INC., "LINEAR I.C.3 DATABOOK", U.S.A., 1989

(2) เซมิคอนดัคเตอร์ อิเลคทรอนิคส์, ลำโพงไร้สาย, ฉบับที่ 86

(3) คู่มือ/เทียบเบอร์ไอซี ทีทีแอล, บริษัท ซีเอ็ดยูเคชั่น จำกัด

(4) คู่มือการใช้โปรแกรม ORCAD/SDT & PROTEL, บริษัท ซีเอ็ดยูเคชั่น จำกัด

(5) ไมโครคอนโทรลเลอร์ MCS-48 MCS 51, ภาควิชา วัดคุม พระจอมเกล้าฯ ลาดกระบัง

(6) ภาษาเบสิกสำหรับไมโครคอมพิวเตอร์, ภาควิชา คอมพิวเตอร์ จุฬาลงกรณ์มหาวิทยาลัย