

## ปริญญานิพนธ์

วิทยุแพคเกจควบคุมโดยไมโครคอนโทรลเลอร์สำหรับวิทยุสมัครเล่น  
PACKET RADIO CONTROL BY MICROCONTROLLER FOR  
AMATURE RADIO



ปริญญานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรครุศาสตรบัณฑิต

๑.

สาขาวิชาวิศวกรรมโทรคมนาคม

๒ ภาควิชาครุศาสตร์วิศวกรรม คณะครุศาสตร์อุตสาหกรรม

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์โดยพระจอมเกล้าเจ้าคุณทหารลาดกระบังไปใช้ประโยชน์ด้านการค้า

เลขที่.....  
เลขทะเบียน.....  
วันที่.....

เลขที่.....  
เลขทะเบียน.....  
วันที่.....

วัน, เดือน, ปี.....  
10 ส.ย. 2542



ภาควิชาครุศาสตร์วิศวกรรม

คณะครุศาสตร์อุตสาหกรรม

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ใบรับรองปริญญานิพนธ์

ชื่อหัวข้อ วิทย์แพคเกจควบคุมโดยไมโครคอนโทรลเลอร์ สำหรับวิทยุสมัครเล่น

Packet Radio Control by Microcontroller for Amature Radio

ชื่อนักศึกษา

1. นายจิรศักดิ์ นามวงศ์ รหัสประจำตัว 40031103
2. นายนเรศ สว่างช้าง รหัสประจำตัว 40031114
3. นายรชชภัทร เหลืองนาม รหัสประจำตัว 40031124
4. นายจิณห์ พงษ์ขจรศิริ รหัสประจำตัว 40031129

หลักสูตร ครุศาสตร์อุตสาหกรรมบัณฑิต สาขาวิชา วิศวกรรมโทรคมนาคม

อาจารย์ที่ปรึกษา อาจารย์อ้อมรัช ชัยชนะ

อาจารย์ที่ปรึกษาร่วม อาจารย์ปิยะ ศุภวรรธิน

| คณะกรรมการสอบปริญญานิพนธ์ |              | ลายมือชื่อ |
|---------------------------|--------------|------------|
| 1. อาจารย์อ้อมรัช         | ชัยชนะ       |            |
| 2. อาจารย์ปิยะ            | ศุภวรรธิน    |            |
| 3. ผศ.วิสุทธิ             | อิทธิธรรม    |            |
| 4. อาจารย์พีระวุฒิ        | สุวรรณจันทร์ |            |
| 5. อาจารย์สุระชัย         | พิมพ์สาลี    |            |

วัน/เดือน/ปีที่สอบ วันอังคารที่ 20 เมษายน พ.ศ. 2542 เวลา 13.00 น.

สถานที่สอบ ห้อง ค.310 คณะครุศาสตร์อุตสาหกรรม สจล.



ภาควิชารับรองแล้ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานภายในเท่านั้น (This document is reserved for internal use only.)  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลง (In any case, it is strictly prohibited to modify or alter.)  
ภาควิชาครุศาสตร์วิศวกรรม (Faculty of Science, Sakon Nakhon Rajabhat University)  
วันที่.....เดือน.....พ.ศ.....

# ปริญญานิพนธ์

เรื่อง วิทยุแพกเกจควบคุมโดยไมโครคอนโทรลเลอร์สำหรับวิทยุสมัครเล่น

Packet Radio Control By Microcontroller For Amature Radio

## วัตถุประสงค์ของปริญญานิพนธ์

1. เพื่อศึกษาการประยุกต์ใช้ไมโครคอนโทรลเลอร์กับวิทยุแพกเกจ สำหรับงานวิทยุสมัครเล่น
2. เพื่อออกแบบวงจรรับส่งข้อมูลด้วยวิทยุแพกเกจโดยใช้ไมโครคอนโทรลเลอร์ควบคุม
3. เพื่อสร้างวงจรรับส่งข้อมูลสำหรับนักวิทยุสมัครเล่นด้วยวิทยุแพกเกจโดยใช้ไมโครคอนโทรลเลอร์ควบคุม
4. เพื่อทดลองวงจรรับส่งข้อมูลสำหรับนักวิทยุสมัครเล่นด้วยวิทยุแพกเกจโดยใช้ไมโครคอนโทรลเลอร์ควบคุม
5. เพื่อนำวงจรรับส่งข้อมูลสำหรับนักวิทยุสมัครเล่นด้วยวิทยุแพกเกจโดยใช้ไมโครคอนโทรลเลอร์ควบคุมไปใช้งานจริง

## ประโยชน์ที่คาดว่าจะได้รับ

1. มีความรู้ในเรื่องของการประยุกต์ และออกแบบใช้งานไมโครคอนโทรลเลอร์ ร่วมกับวิทยุแพกเกจ
2. ได้รับวงจรส่งข้อมูลด้วยวิทยุแพกเกจโดยใช้ไมโครคอนโทรลเลอร์ควบคุม
3. ได้เครื่องต้นแบบวิทยุแพกเกจ ควบคุมโดยไมโครคอนโทรลเลอร์สำหรับวิทยุสมัครเล่น
4. ได้ผลการทดสอบประสิทธิภาพการใช้งานของวิทยุแพกเกจร่วมกับไมโครคอนโทรลเลอร์
5. ได้นำความรู้ด้านการรับส่งข้อมูลโดยใช้วิทยุแพกเกจไปใช้งานกับไมโครคอนโทรลเลอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

|                      |                                                            |               |
|----------------------|------------------------------------------------------------|---------------|
| ชื่อหัวข้อ           | วิทยุแพกเกจควบคุมโดยไม่โครคอนโทรลเลอร์สำหรับวิทยุสมัครเล่น |               |
| นักศึกษา             | นายจิรศักดิ์                                               | นุชนต์        |
|                      | นายนเรศ                                                    | สุรรองช้าง    |
|                      | นายรองยุทธ                                                 | เหล็งหวาน     |
|                      | นายวินิจ                                                   | พยัคฆชาติ     |
| อาจารย์ที่ปรึกษา     | อาจารย์อมรชัย                                              | ชัยชนะ        |
| อาจารย์ที่ปรึกษาร่วม | อาจารย์ปิยะ                                                | ศุภวราสุวัฒน์ |
| หลักสูตร             | ครุศาสตร์อุตสาหกรรมบัณฑิต                                  |               |
| สาขาวิชา             | วิศวกรรมโทรคมนาคม                                          |               |
| ปีการศึกษา           | 2541                                                       |               |

#### บทคัดย่อ

ปฏิญานิพนธ์ฉบับนี้ นำเสนอวิทยุแพกเกจควบคุมโดยไม่โครคอนโทรลเลอร์สำหรับวิทยุสมัครเล่น ซึ่งประกอบด้วยตัวเครื่องของวิทยุแพกเกจเป็นการส่งข้อความสื่อสารระหว่างจุดต่อจุด โดยอาศัยย่านความถี่วีเอชเอฟ (30-300 เมกะเฮิร์ตซ์) สามารถใช้เป็นพิมพ์ภาษาอังกฤษ ขนาด 64 คีย์ และสามารถรับข้อมูล โดยใช้วิทยุสื่อสารย่านความถี่วิทยุสมัครเล่นแสดงผลบนจอแอลซีดีขนาด 4 บรรทัด 20 ตัวอักษร และสามารถแจ้งระดับพลังงานของแบตเตอรี่ของวงจรได้ แล้วสามารถบันทึกข้อมูลที่ใช้ใน Log Book ได้ ซึ่งวิทยุแพกเกจควบคุมโดยไม่โครคอนโทรลเลอร์ สำหรับวิทยุสมัครเล่นนี้สามารถทำการส่งและรับข้อมูลโดยใช้วิทยุแพกเกจร่วมกับไมโครคอนโทรลเลอร์ได้

|                        |                                                       |                |
|------------------------|-------------------------------------------------------|----------------|
| <b>Thesis Title</b>    | Packet Radio Control by Microcontroller Amature Radio |                |
| <b>Students</b>        | Mr.Jirasak                                            | Nuchanong      |
|                        | Mr.Naret                                              | Sukrongchang   |
|                        | Mr.Rongyuth                                           | Langwan        |
|                        | Mr.Vinit                                              | Phayakhachat   |
| <b>Advisor</b>         | Mr.Amornchai                                          | Chaichana      |
| <b>Co-Advisor</b>      | Mr.Piya                                               | Suppawarasuwat |
| <b>Education Level</b> | Bachelor of Science in Industrial Education           |                |
| <b>Program in</b>      | Telecommunication Engineering                         |                |
| <b>Academic Year</b>   | 1998                                                  |                |

### ABSTRACT

This thesis presents the project of Packet Radio Control by Microcontroller for Amature Radio. This project consist of Packet Radio send data to communicate between point by point with VHF bands (30-300 MHz) Using 64 keys size English keyboard and receive data by amature radio which display on 4 lines-20 strings size LCD, can display power level of battery and record data which use in Log Book. This project can send and receive data by use Packet Radio with Microcontroller.

## กิตติกรรมประกาศ

ปริญญานิพนธ์ฉบับนี้ สำเร็จลุล่วงไปด้วยดีเนื่องมาจากความอนุเคราะห์ช่วยเหลือของ อาจารย์ที่ปรึกษาปริญญานิพนธ์ อาจารย์ประจำภาควิชาครุศาสตร์วิศวกรรมทุกท่าน และบุคคลที่เกี่ยวข้องกับการทำโครงการนี้ ที่ได้กรุณาให้คำแนะนำปรึกษาปัญหาในด้านต่างๆ ไปจนถึงข้อมูล และอุปกรณ์ที่เป็นประโยชน์ต่อการทดลอง

สุดท้ายขอกราบขอบพระคุณ บิดา มารดา ที่ได้ให้ทุกสิ่งทุกอย่างตลอดมาจนถึงปัจจุบัน หนึ่งประโยชน์ และคุณงามความดีที่เกิดขึ้นจากปริญญานิพนธ์ฉบับนี้ขอมอบให้แก่ บิดา มารดา ครูบาอาจารย์ที่ประสิทธิ์ประสาทวิชามาตั้งแต่อดีตจนถึงปัจจุบัน ตลอดจนถึงเพื่อนๆ ที่ได้ให้กำลังใจเสมอมาจนทำให้ปริญญานิพนธ์ฉบับนี้สำเร็จลุล่วงไปได้ด้วยดี



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## สารบัญ

| เรื่อง                                    | หน้า |
|-------------------------------------------|------|
| บทคัดย่อภาษาไทย                           | I    |
| บทคัดย่อภาษาอังกฤษ                        | II   |
| กิตติกรรมประกาศ                           | III  |
| สารบัญ                                    | IV   |
| สารบัญตาราง                               | VIII |
| สารบัญรูป                                 | IX   |
| บทที่ 1 บทนำ                              | 1    |
| 1.1 ความเป็นมาและความสำคัญของปริญญานิพนธ์ | 1    |
| 1.2 ขีดความสามารถของโครงการ               | 1    |
| 1.3 เนื้อหาโดยสังเขป                      | 1    |
| บทที่ 2 ทฤษฎี และหลักการ                  | 3    |
| 2.1 กล่าวนำ                               | 3    |
| 2.2 พื้นฐานการสื่อสารโทรคมนาคม            | 3    |
| 2.2.1 องค์ประกอบชั้นพื้นฐานของระบบ        | 3    |
| 2.2.2 สัญญาณที่ใช้ในระบบสื่อสาร           | 5    |
| 2.2.3 ความถี่และสายนำสัญญาณ               | 6    |
| 2.2.4 ความเหมาะสม                         | 9    |
| 2.3 ระบบสื่อสารแอนะล็อก                   | 9    |
| 2.3.1 การมอดูเลต                          | 10   |
| 2.3.2 การมัลติเพล็กซ์                     | 16   |
| 2.4 ระบบสื่อสารดิจิทัล                    | 20   |
| 2.4.1 การเข้ารหัส                         | 21   |
| 2.4.2 การจัดรูปแบบและการเข้ารหัสข้อมูล    | 22   |
| 2.4.3 รูปแบบการมอดูเลตสัญญาณดิจิทัล       | 24   |
| 2.5 การสื่อสารข้อมูล                      | 26   |
| 2.5.1 อักขระในระบบสื่อสารข้อมูล           | 26   |
| 2.5.2 รูปแบบการสื่อสารข้อมูล              | 31   |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้ภายในเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## สารบัญ (ต่อ)

| เรื่อง                                                                | หน้า |
|-----------------------------------------------------------------------|------|
| 2.5.3 ประเภทการบริการของการสื่อสารข้อมูล                              | 32   |
| 2.5.4 ระบบส่งสัญญาณข้อมูล                                             | 32   |
| 2.5.5 ประเภทของการส่งข้อมูล                                           | 33   |
| 2.5.6 การส่งข้อมูลแบบอนุกรม                                           | 33   |
| 2.5.7 การเชื่อมต่อแบบอนุกรมและ UART                                   | 35   |
| 2.5.8 มาตรฐานการเชื่อมต่อแบบ RS-232                                   | 43   |
| 2.6 หลักการและโครงสร้างของ MCS-51                                     | 43   |
| 2.6.1 คุณสมบัติของไมโครคอนโทรลเลอร์ MCS-51                            | 45   |
| 2.6.2 โครงสร้างภายในของไมโครคอนโทรลเลอร์                              | 46   |
| 2.6.3 หน้าที่หลักของภายนอกของไมโครคอนโทรลเลอร์ MCS-51 แบบ Pin และ Pad | 47   |
| 2.6.4 การจัดหน่วยความจำของ MCS-51                                     | 49   |
| 2.7 การสร้างฐานเวลาจริงให้แก่ MCS-51                                  | 50   |
| 2.7.1 วิธีสร้างฐานเวลาจริงให้แก่ MCS-51 (ใช้ชิป RTC)                  | 50   |
| 2.7.2 รายละเอียดของชิป RTC                                            | 52   |
| 2.7.3 โครงสร้างภายในของ RTC DS1202                                    | 53   |
| 2.7.4 โครงสร้าง Command Byte                                          | 54   |
| 2.7.5 การเลือกใช้คริสตอล                                              | 57   |
| 2.8 แพคเกจเรดิโอ                                                      | 59   |
| 2.8.1 ประวัติความเป็นมาของการสื่อสารแพคเกจเรดิโอ                      | 59   |
| 2.8.2 การส่งข้อมูลแบบแพคเกจเรดิโอ                                     | 60   |
| 2.8.3 เทคนิคการควบคุมการรับส่งข้อมูล                                  | 61   |
| บทที่ 3 การออกแบบ การสร้าง และการทำงาน                                | 65   |
| 3.1 กล่าวนำ                                                           | 65   |
| 3.2 หลักการทำงานของวงจรในระบบรับส่งข้อมูล                             | 66   |
| 3.3 หลักการออกแบบ                                                     | 66   |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## สารบัญ (ต่อ)

| เรื่อง                                                                            | หน้า |
|-----------------------------------------------------------------------------------|------|
| 3.3.1 วงจรแพกเกจเรดิโอ                                                            | 66   |
| 3.3.2 วงจรวัฏจักรดับแรงดัน                                                        | 67   |
| 3.3.3 บอร์ดไมโครคอนโทรลเลอร์                                                      | 68   |
| 3.3.4 แป้นพิมพ์ตัวอักษร                                                           | 71   |
| 3.3.5 เครื่องรับส่งวิทยุ                                                          | 72   |
| บทที่ 4 การทดลอง และผลการทดลอง                                                    | 74   |
| 4.1 การทดลอง                                                                      | 74   |
| 4.2 อุปกรณ์ที่ใช้ในการทดลอง                                                       | 74   |
| 4.3 การทดสอบการทำงานของระบบ                                                       | 75   |
| 4.3.1 การทดลองวงจรมอดูเลตและดีมอดูเลต                                             | 75   |
| 4.3.2 การทดสอบประสิทธิภาพการทำงานของไอซี TCM 3105                                 | 75   |
| 4.3.3 การทดสอบประสิทธิภาพของวงจรวัฏจักรดับแรงดัน                                  | 78   |
| 4.4 แผ่นผังแสดงการใช้งาน                                                          | 80   |
| 4.5 ระบบการทำงานและการทดลองใช้งานจริง                                             | 81   |
| บทที่ 5 บทสรุป ปัญหา แนวทางแก้ไข และพัฒนา                                         | 92   |
| 5.1 บทสรุป                                                                        | 92   |
| 5.2 ปัญหา                                                                         | 93   |
| 5.3 แนวทางแก้ไข                                                                   | 93   |
| 5.4 ข้อเสนอแนะ                                                                    | 94   |
| ภาคผนวก ก รูปต้นแบบของวิทยุแพกเกจ ควบคุมโดยไมโครคอนโทรลเลอร์ สำหรับวิทยุสมัครเล่น | 95   |
| ภาคผนวก ข ผังการทำงาน และโปรแกรมการทำงาน                                          | 97   |
| ภาคผนวก ค วงจร และแผ่นวงจรพิมพ์                                                   | 154  |
| ภาคผนวก ง รายการอุปกรณ์                                                           | 158  |
| ภาคผนวก จ คู่มือการใช้งาน                                                         | 161  |
| ภาคผนวก ฉ รายละเอียด และคุณสมบัติของอุปกรณ์                                       | 178  |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## สารบัญ (ต่อ)

เรื่อง

บรรณานุกรม

ประวัติผู้แต่ง

หน้า

249

250



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## สารบัญตาราง

| ตาราง                                                                          | หน้า |
|--------------------------------------------------------------------------------|------|
| ตารางที่ 2.1 สเปกตรัมที่ความถี่ต่างๆ                                           | 7    |
| ตารางที่ 2.2 การเปรียบเทียบคุณลักษณะของการมอดูเลตแบบ AM, FM และ PM             | 14   |
| ตารางที่ 2.3 รหัสสมอร์ส                                                        | 27   |
| ตารางที่ 2.4 รหัสโบคอต                                                         | 28   |
| ตารางที่ 2.5 รหัสแอสกี                                                         | 29   |
| ตารางที่ 2.6 รหัสเอชซีดิก                                                      | 31   |
| ตารางที่ 2.7 อัตราบอดที่ใช้ในการโอนย้ายข้อมูล                                  | 36   |
| ตารางที่ 2.8 มาตรฐานของเคเบิล RS - 232                                         | 44   |
| ตารางที่ 2.9 คำสั่งการกระโดด                                                   | 45   |
| ตารางที่ 2.10 คำสั่งการคำนวณทางคณิตศาสตร์                                      | 57   |
| ตารางที่ 2.11 คำสั่งจัดการข้อมูลแบบบิต                                         | 58   |
| ตารางที่ 2.12 คำสั่งการเคลื่อนย้ายข้อมูล                                       | 58   |
| ตารางที่ 2.13 คำสั่งการคำนวณทางลอจิก                                           | 61   |
| ตารางที่ 2.14 หลักการใช้ Burst Mode                                            | 71   |
| ตารางที่ 2.15 ลักษณะข้อมูลการทำงานของรีจิสเตอร์และ RAM                         | 72   |
| ตารางที่ 4.1 การเปรียบเทียบค่าผลการทดสอบระดับแรงดันลอจิกกับค่าความถี่ที่ได้รับ | 77   |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## สารบัญรูป

| รูป                                                                                                        | หน้า |
|------------------------------------------------------------------------------------------------------------|------|
| รูปที่ 2.1 องค์ประกอบชั้นพื้นฐานของระบบสื่อสารโทรคมนาคม                                                    | 5    |
| รูปที่ 2.2 การสื่อสารในการรับส่งสัญญาณ                                                                     | 10   |
| รูปที่ 2.3 สัญญาณที่ถูกมอดูเลตแบบ AM                                                                       | 11   |
| รูปที่ 2.4 การมอดูเลตแบบ FM                                                                                | 12   |
| รูปที่ 2.5 ตัวอย่างการใช้เทคนิคเฟสล็อกคูลูป (Phase - Locked Loop)                                          | 13   |
| รูปที่ 2.6 การเปรียบเทียบสัญญาณ PM, AM และ FM กับสัญญาณคิจิตอลเลขฐานสอง                                    | 14   |
| รูปที่ 2.7 การเปลี่ยนของบิตในระบบอะซิงโครนัส                                                               | 15   |
| รูปที่ 2.8 การแก้ปัญหาโดยใช้สัญญาณนาฬิกาทางด้านรับช่วยในระบบซิงโครนัส                                      | 16   |
| รูปที่ 2.9 การมัลติเพล็กซ์สัญญาณความถี่ต่างๆ ในระบบ FDM                                                    | 17   |
| รูปที่ 2.10 การส่งสัญญาณในระบบ FDM                                                                         | 18   |
| รูปที่ 2.11 การมัลติเพล็กซ์สัญญาณที่เวลาต่างๆ โดยใช้สัญญาณซิงค์ช่วยในระบบ TDM                              | 18   |
| รูปที่ 2.12 การส่งสัญญาณในระบบ TDM                                                                         | 19   |
| รูปที่ 2.13 ตัวอย่างการประยุกต์ใช้งานของโทรศัพท์ทางไกล โดยมีการมัลติเพล็กซ์แบบต่างๆ                        | 19   |
| รูปที่ 2.14 ความสัมพันธ์ระหว่างอุปกรณ์รับส่งข้อมูล (DTE) และอุปกรณ์รับส่งปลายทาง (DCE) ในระบบสื่อสารข้อมูล | 32   |
| รูปที่ 2.15 การส่งข้อมูลแบบขนานและแบบอนุกรม                                                                | 34   |
| รูปที่ 2.16 การแปลงข้อมูลแบบขนานเป็นข้อมูลอนุกรม                                                           | 35   |
| รูปที่ 2.17 เวิร์คข้อมูลขนาด 8 บิต กับบิตสตอปที่ใช้ในการโอนย้ายข้อมูลแบบอนุกรม                             | 36   |
| รูปที่ 2.18 เวิร์คข้อมูลขนาด 8 บิต ที่ใช้ในการโอนย้ายข้อมูลแบบอนุกรมด้วยอัตรา 110 บอด                      | 37   |
| รูปที่ 2.19 การใช้บิตพาริตีเพื่อตรวจสอบความผิดพลาดในการโอนย้ายข้อมูลแบบอนุกรม                              | 38   |
| รูปที่ 2.20 เฟรมมิงเออเรอร์ข้อมูลที่รับเข้ามาที่ตำแหน่งเริ่มต้นและตำแหน่งสิ้นสุดไม่ถูกต้อง                 | 38   |
| รูปที่ 2.21 แผนผังการทำงานของ UART                                                                         | 39   |
| รูปที่ 2.22 สัญญาณคาล์อีนพุตสโตป                                                                           | 39   |
| รูปที่ 2.23 การโอนย้ายข้อมูลแบบอนุกรมที่เป็นตัวอักษรในรหัสแอสกี                                            | 41   |
| รูปที่ 2.24 การโอนย้ายข้อมูลแบบอนุกรมที่ประกอบไปด้วยอักขระ 4 ตัว                                           | 42   |
| รูปที่ 2.25 ขั้นตอนการทำงานของไมโครโปรเซสเซอร์                                                             | 42   |

## สารบัญรูป (ต่อ)

| รูป                                                                                    | หน้า |
|----------------------------------------------------------------------------------------|------|
| รูปที่ 2.26 ลักษณะของเค้าเสียบบ DB - 25 ที่ใช้ในการเชื่อมต่อทางกายภาพตามมาตรฐาน RS-232 | 43   |
| รูปที่ 2.27 ลักษณะของเค้าเสียบบ DB - 9 ที่ใช้ในการเชื่อมต่อทางกายภาพตามมาตรฐาน RS-232  | 44   |
| รูปที่ 2.28 โครงสร้างภายในไมโครคอนโทรลเลอร์                                            | 46   |
| รูปที่ 2.29 ลักษณะภายนอกของ MCS-51                                                     | 48   |
| รูปที่ 2.30 การจัดพื้นที่หน่วยความจำโปรแกรม                                            | 49   |
| รูปที่ 2.31 การจัดพื้นที่หน่วยความจำข้อมูล                                             | 50   |
| รูปที่ 2.32 ลักษณะรูปร่างทั้งสองแบบไอซีเบอร์ DS1202                                    | 51   |
| รูปที่ 2.33 โครงสร้างภายในของ RTC DS1202                                               | 53   |
| รูปที่ 2.34 โครงสร้างของ Command Byte                                                  | 54   |
| รูปที่ 2.35 เปรียบเทียบการใช้งานแพ็คเกจเรดิโอกับการใช้งาน โมเด็มรับส่งข้อมูล           | 60   |
| รูปที่ 2.36 การส่งแพ็คเกจโดยใช้เทคนิค TDMA                                             | 62   |
| รูปที่ 2.37 การส่งแพ็คเกจโดยใช้เทคนิค FDMA                                             | 63   |
| รูปที่ 2.38 ลักษณะการเกิดการชนกันของแพ็คเกจระหว่างผู้ใช้ หลายคน                        | 64   |
| รูปที่ 2.39 พฤติกรรมการรับส่งข้อมูลของสถานีหนึ่ง ในกรณีของ CSMA                        | 64   |
| รูปที่ 3.1 ผังของระบบการรับส่งข้อมูลในช่องสัญญาณวิทยุสื่อสาร                           | 65   |
| รูปที่ 3.2 แผนผังการทำงานของระบบการรับส่งข้อมูลในช่องสัญญาณวิทยุสื่อสาร                | 66   |
| รูปที่ 3.3 วงจรแพ็คเกจเรดิโอ                                                           | 67   |
| รูปที่ 3.4 วงจรวัดระดับแรงดัน                                                          | 67   |
| รูปที่ 3.5 โครงสร้างของบอร์ดไมโครคอนโทรลเลอร์ รุ่น ET - BASIC32 V3.0                   | 68   |
| รูปที่ 3.6 วงจรภายในของบอร์ดไมโครคอนโทรลเลอร์ รุ่น ET - BASIC32 V3.0                   | 69   |
| รูปที่ 3.7 โครงสร้างของวงจรควบคุมการทำงานของแป้นพิมพ์ตัวอักษร                          | 71   |
| รูปที่ 3.8 แป้นพิมพ์ตัวอักษร                                                           | 72   |
| รูปที่ 3.9 เครื่องรับส่งวิทยุ                                                          | 73   |
| รูปที่ 4.1 ลักษณะของสัญญาณการมอดูเลตแบบ FSK และผลึกความถี่ออกมาในลักษณะ                | 76   |

## สารบัญรูป (ต่อ)

| รูป                                                                                                                                                | หน้า |
|----------------------------------------------------------------------------------------------------------------------------------------------------|------|
| รูปที่ 4.2 ลักษณะของสัญญาณการมอดูเลตแบบ FSK และผลิตความถี่ออกมาในลักษณะสัญญาณดิจิทัล                                                               | 76   |
| รูปที่ 4.3 ลักษณะของสัญญาณที่ถูกมอดูเลตจากสัญญาณดิจิทัลเป็นสัญญาณแอนะล็อกที่ความถี่ 1,300 เฮิรตซ์                                                  | 77   |
| รูปที่ 4.4 ลักษณะของสัญญาณที่ถูกมอดูเลตจากสัญญาณดิจิทัลเป็นสัญญาณแอนะล็อกที่ความถี่ 1,300 เฮิรตซ์                                                  | 78   |
| รูปที่ 4.5 ลักษณะสถานะของวงจรวัฏระดับแรงดันไฟในช่วงแรงดันที่มีค่าสูงกว่า 9.2 โวลต์                                                                 | 79   |
| รูปที่ 4.6 ลักษณะสถานะของวงจรวัฏระดับแรงดันไฟในช่วงแรงดันประมาณ 7.8 โวลต์                                                                          | 79   |
| รูปที่ 4.7 ลักษณะสถานะของวงจรวัฏระดับแรงดันไฟในช่วงแรงดันที่มีค่าต่ำกว่า 7.2 โวลต์                                                                 | 80   |
| รูปที่ 4.8 แผนผังการใช้งานของวิทยุแพคเกจร่วมกับอุปกรณ์อื่นๆ ที่ใช้ในงานทดลองจริง                                                                   | 80   |
| รูปที่ 4.9 ลักษณะการต่อวิทยุแพคเกจเข้ากับอุปกรณ์ ผ่านทางขั้วต่อชนิดต่างๆ                                                                           | 81   |
| รูปที่ 4.10 การใส่แผ่นดิสก์ที่บรรจุ โปรแกรม PROCOM อยู่ เพื่อเตรียมทำการ Setup                                                                     | 82   |
| รูปที่ 4.11 การ Setup หรือ Install โปรแกรม PROCOM ลงสู่เครื่องคอมพิวเตอร์                                                                          | 82   |
| รูปที่ 4.12 หน้าจอแสดงผล โปรแกรม PROCOM                                                                                                            | 83   |
| รูปที่ 4.13 Menu Setup พอร์ตสื่อสาร COM1 และ COM2                                                                                                  | 83   |
| รูปที่ 4.14 Menu Setup บอกระยะการสื่อสารให้ใช้บอกระยะ 9,600 bps                                                                                    | 84   |
| รูปที่ 4.15 การเข้าสู่โหมด พร้อมรับข้อมูลจากวิทยุแพคเกจ                                                                                            | 84   |
| รูปที่ 4.16 การตั้งย่านความถี่ใช้ในการทดลองสื่อสารที่ย่านความถี่ 145.500 MHz                                                                       | 85   |
| รูปที่ 4.17 รายการเลือกหลักหลังเปิดสวิทซ์ทำงานแก่เครื่องวิทยุแพคเกจ                                                                                | 86   |
| รูปที่ 4.18 สถานะของ Connection พร้อมทั้งจะทำการติดต่อสื่อสารกับคู่สถานี                                                                           | 86   |
| รูปที่ 4.19 การรับข้อมูลจากสถานีคู่เข้ามาโดยรับจาก HS2 BBB ข้อความว่า "HELLO"                                                                      | 87   |
| รูปที่ 4.20 การบันทึกข้อมูลการสนทนาอัตโนมัติ                                                                                                       | 87   |
| รูปที่ 4.21 ข้อความที่บันทึกการสนทนาที่เกิดขึ้น                                                                                                    | 88   |
| รูปที่ 4.22 การถ่ายโอนข้อมูลที่บันทึกการสนทนาจากแฟ้ม 0 ไปสู่เครื่องคอมพิวเตอร์                                                                     | 88   |
| รูปที่ 4.23 การรับข้อมูลของคอมพิวเตอร์ จากการส่งข้อมูลมาจากเครื่องวิทยุแพคเกจ เพื่อเลือกการเก็บข้อมูลไว้ในแผ่นดิสก์หรือพิมพ์ผลผ่านคอมพิวเตอร์ต่อไป | 89   |

## สารบัญรูป (ต่อ)

| รูป                                                                                | หน้า |
|------------------------------------------------------------------------------------|------|
| รูปที่ 4.24 การเลือกบันทึกชื่อโดยใช้คำสั่ง Page Down                               | 89   |
| รูปที่ 4.25 การบันทึกชื่อแฟ้มที่ต้องการบันทึกด้วยชื่อไฟล์ TEST.TXT                 | 90   |
| รูปที่ 4.26 การออกสู่โปรแกรม PCP                                                   | 90   |
| รูปที่ 4.27 การใช้คำสั่งชนิด TEST.TXT เพื่อดูข้อมูลใน Log Book                     | 91   |
| รูปที่ ก.1 รูปต้นแบบของวิทยุแพคเกจ ควบคุมโดยไมโครคอนโทรลเลอร์ สำหรับวิทยุสมัครเล่น | 96   |
| รูปที่ ข.1 แผนผังการทำงานของโปรแกรมหลัก                                            | 98   |
| รูปที่ ข.2 แผนผังการทำงานของโปรแกรม Help                                           | 99   |
| รูปที่ ข.3 แผนผังการทำงานของโปรแกรม Command                                        | 100  |
| รูปที่ ข.4 แผนผังการทำงานของโปรแกรม Mode                                           | 101  |
| รูปที่ ข.5 แผนผังการทำงานของโปรแกรม Setup                                          | 102  |
| รูปที่ ข.6 แผนผังการทำงานของโปรแกรม Q - Code                                       | 103  |
| รูปที่ ข.7 แผนผังการทำงานของโปรแกรมน้อย Introduction                               | 105  |
| รูปที่ ข.8 แผนผังการทำงานของโปรแกรมน้อย Log Book                                   | 107  |
| รูปที่ ข.9 แผนผังการทำงานของโปรแกรมน้อย Standby                                    | 110  |
| รูปที่ ข.10 แผนผังการทำงานของโปรแกรมน้อย Connect                                   | 111  |
| รูปที่ ข.11 แผนผังการทำงานของโปรแกรมน้อย Date                                      | 112  |
| รูปที่ ข.12 แผนผังการทำงานของโปรแกรมน้อย Time                                      | 113  |
| รูปที่ ข.13 แผนผังการทำงานของโปรแกรมน้อย Callsing                                  | 114  |
| รูปที่ ข.14 โปรแกรมควบคุมการทำงานของแพคเกจเรดิโอ                                   | 153  |
| รูปที่ ค.1 ลายวงจรพิมพ์ของเป็นพิมพ์ตัวอักษร                                        | 155  |
| รูปที่ ค.2 ลายวงจรพิมพ์ของวงจรวัดระดับแรงดัน                                       | 156  |
| รูปที่ ค.3 การวางอุปกรณ์ของวงจรวัดระดับแรงดัน                                      | 156  |
| รูปที่ ค.4 ด้านหน้าลายวงจรพิมพ์ของวงจรแพคเกจ                                       | 157  |
| รูปที่ ค.5 ด้านหลังลายวงจรพิมพ์ของวงจรแพคเกจ                                       | 157  |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## สารบัญรูป (ต่อ)

| รูป                                                                                      | หน้า |
|------------------------------------------------------------------------------------------|------|
| รูปที่ 1 ส่วนด้านบนเครื่อง                                                               | 164  |
| รูปที่ 2 ส่วนด้านหลังเครื่อง                                                             | 164  |
| รูปที่ 3 ส่วนด้านหน้าเครื่อง                                                             | 165  |
| รูปที่ 4 ข้อความเริ่มต้น เมื่อเปิดสวิตช์ใช้เครื่องทำงาน                                  | 166  |
| รูปที่ 5 ข้อความหลังจากที่เปิดสวิตช์ต่อจากข้อความแรก                                     | 166  |
| รูปที่ 6 รายการเลือกในการเลือกใช้งาน 4 ฟังก์ชัน                                          | 167  |
| รูปที่ 7 ข้อความ เมื่อกด F1 ในโหมด Help                                                  | 167  |
| รูปที่ 8 ข้อความ เมื่อกด “1” ที่โหมด Q - Code                                            | 167  |
| รูปที่ 9 ข้อความ เมื่อกดปุ่มลูกศรเลื่อนหารหัส Q ที่ต้องการเลือกแล้วกด Enter              | 168  |
| รูปที่ 10 ข้อความ เมื่อกดเลื่อนลูกศรลง ทำการเลือกดูรหัส Q                                | 168  |
| รูปที่ 11 ข้อความ เมื่อกด Esc เมื่อกดกลับเข้าสู่รายการเลือก Help                         | 169  |
| รูปที่ 12 ข้อความ เมื่อกดเลข 2 เมื่อเข้าสู่โหมด Introduction                             | 169  |
| รูปที่ 13 ข้อความ เมื่อเลือก F2 เมื่อเข้าสู่โหมด Command                                 | 169  |
| รูปที่ 14 ข้อความ เมื่อกดเลข 1 เพื่อแสดงผลของเวลา                                        | 170  |
| รูปที่ 15 ข้อความ เมื่อกดเลข 2 เพื่อดูข้อมูลที่บันทึกไว้ในการสนทนา                       | 170  |
| รูปที่ 16 ข้อความ เมื่อกดเลข 0 - 9                                                       | 170  |
| รูปที่ 17 ข้อความ เมื่อกด Esc เพื่อเข้าสู่รายการเลือก Log Book                           | 171  |
| รูปที่ 18 ข้อความ เมื่อกดปุ่ม D เพื่อลบข้อมูล                                            | 171  |
| รูปที่ 19 ข้อความ เมื่อกด F3 เพื่อเข้ารายการเลือกโหมด                                    | 171  |
| รูปที่ 20 ข้อความ เมื่อกดเลข 1 เพื่อทำการติดต่อสื่อสารกัน                                | 172  |
| รูปที่ 21 ข้อความ เมื่อมีคู่สถานีส่งข้อมูลเข้ามา                                         | 172  |
| รูปที่ 22 ข้อความ เมื่อกดเลข 2 เข้าโหมด Standby                                          | 173  |
| รูปที่ 23 ข้อความ เมื่อมีข้อมูลส่งเข้ามา                                                 | 173  |
| รูปที่ 24 ข้อความ เมื่อกด F4 เข้าเลือกรายการ Setup                                       | 173  |
| รูปที่ 25 ข้อความ เมื่อกดเลขรายการเลือก Setup ในการตั้งค่า วัน - เดือน - ปี              | 174  |
| รูปที่ 26 ข้อความ เมื่อกดเลข 2 จากรายการเลือก Setup ในการตั้งค่า ชั่วโมง - นาที - วินาที | 174  |

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์และใช้เฉพาะในการศึกษาเท่านั้น ไม่ควรนำข้อมูลไปเผยแพร่โดยไม่ได้รับอนุญาต  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป (ต่อ)

| รูป                                                                                | หน้า |
|------------------------------------------------------------------------------------|------|
| รูปที่ 27 ข้อความ เมื่อกดเลข 3 จากรายการเลือก Setup ในการตั้งค่าชื่อสถานีของผู้ใช้ | 174  |
| รูปที่ 28 การวางตำแหน่งตัวอักษรของเป็นพิมพ์                                        | 177  |



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# บทที่ 1

## บทนำ

### 1.1 ความเป็นมาและความสำคัญของปริญญานิพนธ์

ในการติดต่อสื่อสารระหว่าง นักวิทยุสมัครเล่นทั้งหลายในปัจจุบัน ได้มีการนำคอมพิวเตอร์มาพัฒนาใช้งานร่วมกับการสื่อสารระบบเก่า แต่ต้องอาศัยการลงทุนที่สูง โดยเฉพาะในส่วนของคอมพิวเตอร์

ดังนั้นเราจึงเห็นว่า เราสามารถที่จะนำไมโครคอนโทรลเลอร์มาประยุกต์ใช้งานแทนคอมพิวเตอร์ได้ โดยจะช่วยลดค่าใช้จ่าย ในการติดต่อสื่อสารระบบวิทยุแพคเกจ ซึ่งให้ประสิทธิภาพการใช้งานใกล้เคียงกับคอมพิวเตอร์ และสามารถพกพาไปใช้งานได้สะดวกกว่าในทุกที่

### 1.2 ขีดความสามารถของโครงการ

โครงการนี้มีขีดความสามารถดังต่อไปนี้

1. สามารถรับส่วนข้อมูล โดยใช้วิทยุสื่อสารย่านวิทยุสมัครเล่นแสดงผลบนจอ LCD 20X4 เป็นอักษรภาษาอังกฤษ
2. สามารถใช้เป็นพิมพ์ภาษาอังกฤษจำนวน 64 คีย์
3. ความเร็วในการรับส่งข้อมูลประมาณ 1,200 bps ระหว่าง วิทยุแพคเกจกับคอมพิวเตอร์
4. สามารถใช้แหล่งจ่ายภายใน และสามารถแจ้งระดับพลังงานของแบตเตอรี่ได้
5. สามารถบันทึกข้อมูลที่ใช้ใน Log Book ได้
6. สามารถส่งข้อมูล Log Book ไปเก็บไว้ในคอมพิวเตอร์ได้

### 1.3 เนื้อหาโดยสังเขป

เนื้อหาภายในปริญญานิพนธ์ฉบับนี้แบ่งออกเป็นบทต่างๆ เพื่อความสะดวกต่อการศึกษา และทำความเข้าใจ ในแต่ละบทจะประกอบด้วยเนื้อหาที่สำคัญดังนี้

บทที่ 2 ทฤษฎี และหลักการ ประกอบด้วยเนื้อหาในทางทฤษฎีที่เกี่ยวข้อง ซึ่งทำให้ผู้อ่านมีความรู้ความเข้าใจที่เป็นพื้นฐาน อันจะเป็นประโยชน์ต่อการทำความเข้าใจกับวงจรที่ใช้งานจริงต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3 การออกแบบ การสร้าง และการทำงาน กล่าวถึงการสร้างการออกแบบฮาร์ดแวร์ และซอฟต์แวร์ รวมทั้งหลักการทำงานในส่วนต่างๆ ซึ่งทำให้ผู้อ่านมีความเข้าใจ การทำงานโดยรวมของโครงการนี้

บทที่ 4 การทดลอง และผลการทดลอง กล่าวถึงขั้นตอนการทดลอง และการทดสอบการทำงานของฮาร์ดแวร์ และซอฟต์แวร์ของโครงการนี้ เพื่อตรวจสอบว่าโครงการนี้มีความสามารถทำงานได้ตรงตามวัตถุประสงค์หรือไม่

บทที่ 5 บทสรุป ปัญหา แนวทางแก้ไข และพัฒนา เป็นการสรุปผลการทำงาน และได้เสนอแนะแนวทางในการแก้ไข และแนวทางในการพัฒนา ให้มีประสิทธิภาพ

ภาคผนวก ก รูปต้นแบบของวิทยุแพคเกจควบคุม โดยไมโครคอนโทรลเลอร์สำหรับวิทยุสมัครเล่น

ภาคผนวก ข ผังการทำงาน และโปรแกรมการทำงาน

ภาคผนวก ค วงจร และแผ่นวงจรพิมพ์

ภาคผนวก ง รายการอุปกรณ์

ภาคผนวก จ คู่มือการใช้งาน

ภาคผนวก ฉ รายละเอียด และคุณสมบัติของอุปกรณ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทที่ 2

### ทฤษฎี และหลักการ

#### 2.1 กล่าวนำ

การพัฒนาของอารยธรรมในปัจจุบันนี้มีมากมายนัก เนื่องมาจากความสามารถของมนุษย์ ในการที่จะแลกเปลี่ยนข้อมูลข่าวสาร และความคิดโดยอาศัยการมองการฟัง และโดยคำที่ถูกเขียน โดยใช้รูปแบบของภาษาหรือรหัส (Codes) ที่เกิดการยอมรับ จากแรกเริ่มมนุษย์ได้แสหาวิธีการส่ง ข้อมูลข่าวสารให้ได้ระยะทางไกลเท่าที่จะทำได้ส่วนใหญ่จะขึ้นอยู่กับวิธีการใช้สัญญาณควัน สัญญาณไฟ สัญญาณท่าทาง หรือสัญญาณวัตถุ เป็นต้น ชาวอินเดียเคยนิยมใช้สัญญาณไฟหรือ สัญญาณธง ชาวประมงก็อาจจะใช้ไฟกระพริบ สิ่งที่ควรจะต้องชี้ให้เห็นถึงความหมายของการสื่อสาร โทรคมนาคมที่ได้นำมาจากรากศัพท์ของคำนั้นๆ คำว่า “Tele” นั้นมาจากภาษากรีกโบราณ ซึ่งแปลว่าระยะทางที่ไกล ส่วนคำว่า “Vision” หมายถึงการดูมอง และคำว่า “Communication” หมายถึง การติดต่อสื่อสาร ซึ่งการสื่อสารโทรคมนาคม (Telecommunication) แปลว่าการติดต่อสื่อสารที่ ระยะทางไกล

ดังนั้นการสื่อสารโทรคมนาคม จึงเป็นขั้นตอนของการส่งผ่านพลังงานของข่าวสารเป็น ระยะทางไกลๆ โดยวิธีการทางไฟฟ้า พลังงานของข่าวสารจะถูกส่งผ่านไปยังปลายทาง อาจจะโดย วิธีใช้เส้นลวดตัวนำหุ้มฉนวน หรือจะใช้วิธีส่งผ่านชั้นบรรยากาศโดยการเชื่อมต่อของสัญญาณวิทยุ โดยไม่ต้องใช้ลวดตัวนำใดๆ ตัวอย่างของการใช้พลังงานไฟฟ้าสำหรับงานต่างๆ ที่รู้จักกันทั่วไป เช่น อุปกรณ์ไฟฟ้า แสงไฟมอเตอร์ไฟฟ้า เป็นต้น พลังงานไฟฟ้าจะถูกเปลี่ยนเป็นพลังงานในรูปแบบ ต่างๆ ตามวัตถุประสงค์นั้นๆ ในด้านการสื่อสารโทรคมนาคม จะพบว่าบางรูปแบบของพลังงาน เมื่อบรรจุข่าวสารลงไป จะถูกเปลี่ยนไปเป็นพลังงานไฟฟ้าเพื่อที่จะส่งผ่านไปยังจุดหมายปลายทาง ที่อยู่ห่างไกลออกไป เมื่อถึงจุดหมายปลายทาง พลังงานไฟฟ้าก็จะถูกเปลี่ยนกลับมาเป็นรูปแบบเดิม มีความหมายเหมือนเดิม ซึ่งเป็นอันว่าได้เกิดการติดต่อสื่อสารขึ้นแล้ว

#### 2.2 พื้นฐานการสื่อสารโทรคมนาคม

##### 2.2.1 องค์ประกอบขั้นพื้นฐานของระบบ

องค์ประกอบขั้นพื้นฐานของระบบการสื่อสารโทรคมนาคมดังแสดงในรูปที่ 2.1 สามารถ อธิบายเป็นเอกสารที่ส่งมอบไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้เผยแพร่ใน การค้า จำแนกออกเป็นส่วนประกอบได้ดังต่อไปนี้

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1) ผู้ส่งข่าวสารหรือแหล่งกำเนิดข่าวสาร (Source) อาจจะเป็นสัญญาณต่างๆ เช่น สัญญาณภาพ ข้อมูล และเสียง เป็นต้น ในการติดต่อสื่อสารสมัยก่อนอาจใช้แสงไฟ คandles หรือท่าทางต่างๆ ก็นับว่าเป็นแหล่งกำเนิดข่าวสาร จัดอยู่ในหมวดหมู่เช่นกัน

2) ผู้รับข่าวสารหรือจุดหมายปลายทางของข่าวสาร (Sink) ซึ่งจะรับรู้จากสิ่งที่ผู้ส่งข่าวสารหรือแหล่งกำเนิดข่าวสารส่งผ่านมาให้ トラバิดที่การติดต่อสื่อสารบรรลุวัตถุประสงค์ ผู้รับสารหรือจุดหมายปลายทางไม่ได้รับข่าวสาร ก็แสดงว่าการสื่อสารนั้นไม่ประสบความสำเร็จ กล่าวคือไม่มี การสื่อสารเกิดขึ้นนั่นเอง

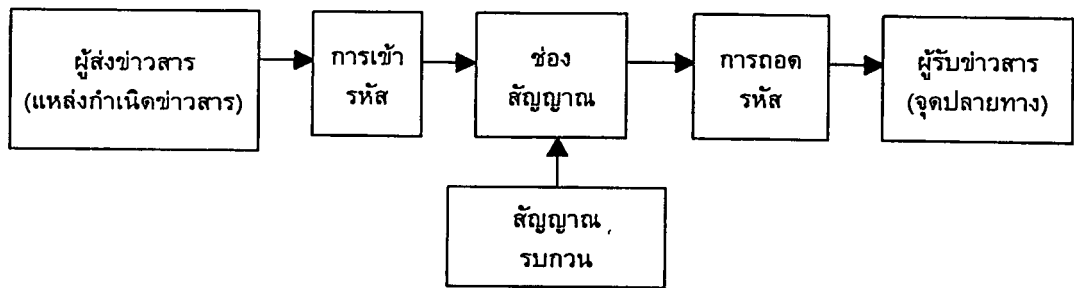
3) ช่องสัญญาณ (Channel) ในที่นี้อาจจะหมายถึง สื่อกลางหรือตัวกลางที่ข่าวสารเดินทางผ่าน อาจจะเป็นอากาศ สายนำสัญญาณต่างๆ หรือแม้กระทั่งของเหลว เช่น น้ำ น้ำมัน เป็นต้น เปรียบเสมือนเป็นสะพานที่จะให้ข่าวสารข้ามจากฝั่งหนึ่งไปยังอีกฝั่งหนึ่ง

4) การเข้ารหัส (Encoding) เป็นการช่วยให้ผู้ส่งข่าวสารและผู้รับข่าวสารมีความเข้าใจตรงกันในการสื่อความหมาย จึงมีความจำเป็นต้องแปลงความหมายนี้ การเข้ารหัสจึงหมายถึงการแปลงข่าวสารให้อยู่ในรูปพลังงานที่พร้อมจะส่งไปในสื่อกลาง และทางผู้รับมีความสามารถในการแปลงพลังงานให้กลับไปอยู่ในรูปข่าวสารที่สามารถเข้าใจได้ ข้อสำคัญคือความเข้าใจต้องตรงกันระหว่างผู้ส่งและผู้รับ หรือมีรหัสเดียวกันการสื่อสารจึงเกิดขึ้นได้

5) การถอดรหัส (Decoding) หมายถึงการที่ผู้รับข่าวสารแปลงพลังงานจากสื่อกลางให้กลับไปอยู่ในรูปข่าวสารที่ส่งมาจากผู้ส่งข่าวสาร โดยมีความเข้าใจหรือรหัสตรงกัน

6) สัญญาณรบกวน (Noise) เป็นสิ่งที่มีอยู่ในธรรมชาติ มักจะลดทอนหรือรบกวนระบบ อาจเกิดขึ้นได้ทั้งทางด้านผู้ส่งข่าวสาร ผู้รับข่าวสาร และช่องสัญญาณ แต่ในการศึกษาขั้นพื้นฐาน มักจะสมมติให้ทางด้านผู้ส่งข่าวสารและผู้รับข่าวสารไม่มีความผิดพลาด ตำแหน่งที่ใช้วิเคราะห์มักจะเป็นตัวกลางหรือช่องสัญญาณเมื่อไรที่รวมสัญญาณรบกวนด้านผู้ส่งข่าวสารและด้านผู้รับข่าวสาร สิ่งที่ส่งมาจะเป็นข่าวสารจากแหล่งกำเนิดรวมกับสัญญาณรบกวนทางด้านส่ง ในทางปฏิบัติมักจะใช้วงจรกรอง (Filter) กรองสัญญาณรบกวนแต่ต้นทาง เพื่อให้การสื่อสารมีคุณภาพดียิ่งขึ้นแล้วค่อยดำเนินการ เช่น การเข้ารหัสแหล่งข้อมูล เป็นต้น

ความต้องการขั้นพื้นฐานของระบบโทรคมนาคมในการส่งสัญญาณนั้น พลังงานที่มีข้อมูลข่าวสารจะต้องถูกเปลี่ยนไปอยู่ในรูปของพลังงานไฟฟ้า เพื่อที่จะผลิตสัญญาณข้อมูลในรูปของสัญญาณแอนะล็อกหรือสัญญาณดิจิทัล ซึ่งสามารถทำได้โดยใช้เครื่องแปลงสัญญาณที่เหมาะสมในการแปลงพลังงานจากรูปหนึ่งไปเป็นอีกรูปแบบหนึ่ง ในการใช้งานจริงอาจจะต้องใช้ชิ้นส่วนอื่นๆ เพิ่มเติมอีกเพื่อความเหมาะสม เช่น วงจรขยาย วงจรกรอง เป็นต้น สิ่งสำคัญที่พึงจะตระหนัก คือ ในระบบโทรคมนาคมจะมีการรบกวนจากสัญญาณรบกวน และยังมีการบิดเบือนของสัญญาณไปใช้



รูปที่ 2.1 องค์ประกอบขั้นพื้นฐานของระบบสื่อสารโทรคมนาคม

เกิดขึ้นด้วย สิ่งเหล่านี้เป็นผลที่เราไม่ต้องการ และควรให้มีน้อยที่สุด ในการพิจารณาระบบเครือข่ายโทรคมนาคม หลักเบื้องต้นที่จะต้องคำนึงถึง คือ ทำอย่างไรให้ข่าวสารหรือข้อมูลที่ได้รับมีความสมบูรณ์และมีสภาพเหมือนข่าวสารที่ส่งมามากที่สุด แล้วค่อยมาพิจารณาถึงประสิทธิภาพของระบบ

## 2.2.2 สัญญาณที่ใช้ในระบบสื่อสาร

สัญญาณที่ใช้ในระบบเครือข่ายการสื่อสารโทรคมนาคม แบ่งออกได้เป็น 2 ประเภท คือ สัญญาณแอนะล็อกและสัญญาณดิจิทัล สัญญาณแอนะล็อกเป็นสัญญาณที่มีขนาดเป็นค่าต่อเนื่อง ส่วนสัญญาณดิจิทัลเป็นสัญญาณที่มีขนาดที่เปลี่ยนแปลงเป็นค่าของเลขลงตัว โดยปกติมักแทนด้วยระดับแรงดันที่แสดงสถานะเป็น “0” และ “1” หรืออาจจะมีหลายสถานะ ข้อดีในการทำให้เกิดความผิดพลาดน้อยลง เนื่องจากสัญญาณรบกวนต้องมีค่าสูงกว่าค่าที่ตั้งไว้สถานะจึงจะเปลี่ยนตัวอย่างเช่น ในระบบดิจิทัล สถานะของข้อมูลเป็น “0” สัญญาณรบกวนมีค่า 0.2 โวลต์ แต่ค่าที่ตั้งไว้เท่ากับ 0.5 โวลต์ สถานะยังคงเดิมคือเป็น “0” ในขณะที่ระบบแอนะล็อก สัญญาณรบกวนจะเดินเข้าไปในสัญญาณจริงโดยตรง กล่าวคือสัญญาณจริงบวกสัญญาณรบกวนเป็นสัญญาณขณะนั้น ทำให้สัญญาณรบกวนมีผลต่อสัญญาณจริงและมีความผิดพลาดเกิดขึ้น

กระแสไฟฟ้าแบ่งออกได้เป็นไฟฟ้ากระแสตรงและกระแสสลับ หลากๆ คนอาจจะคิดว่าไม่มีความเกี่ยวข้องกับระบบสื่อสารโทรคมนาคม เมื่อกล่าวถึงสัญญาณในเชิงประยุกต์ก็อาจจะจำแนกในหมวดหมู่นี้ การไหลของไฟฟ้ากระแสตรงในวงจรอย่างสม่ำเสมอไม่สามารถส่งข่าวสารได้ แต่เมื่อไรที่ต้องการควบคุมกระแสให้เป็นพัลส์โดยการเปิดสวิตช์ กระแสจะลดลงสู่ศูนย์ และปิดสวิตช์ กระแสก็จะมีค่าหนึ่ง พัลส์ของกระแสถูกผลิตตามรหัสที่ใช้แทนแต่ละตัวอักษรหรือตัวเลขโดยการรวมของพัลส์การทำงานของสวิตช์สามารถส่งข้อความใดๆ ได้ ตัวอย่างเช่น รหัสสมอร์ส เป็นต้น ส่วนไฟฟ้ากระแสสลับในรูปของคลื่นอยู่ในจำพวกคลื่นวิทยุ มีการใช้งานอย่างกว้างขวางเป็นที่รู้จักกันดี ไม่ว่าจะเป็นคลื่นใดๆ ทั้งสิ้น อีกทั้งยังมีให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สัญญาณหนึ่งที่น่าสนใจคือรูปคลื่นที่ซับซ้อน ตัวอย่างของรูปคลื่นที่ซับซ้อนที่พบเห็นได้บ่อยๆ ในระบบสื่อสารโทรคมนาคม โดยใช้การวิเคราะห์ทางคณิตศาสตร์เราจะพบว่ารูปคลื่นที่ซับซ้อน ซึ่งสร้างขึ้นจากรูปคลื่นไซน์ที่มีความถี่ที่แน่นอนค่าหนึ่งที่เรียกว่า ความถี่พื้นฐาน (Fundamental Frequency) และจำนวนของรูปคลื่นไซน์อื่นๆ จะมีความถี่เป็นจำนวนเท่าของความถี่พื้นฐาน และมีค่าสูงสุดที่ลดลง จำนวนเท่าเหล่านี้เรียกว่า ฮาร์โมนิกของความถี่พื้นฐาน เช่น รูปคลื่นที่ซับซ้อนมีความถี่พื้นฐาน  $f$  เฮิร์ตซ์ จะมีความถี่ฮาร์โมนิกเป็น  $2f, 3f, 4f, 5f, 6f$  และอื่นๆ ซึ่งมักจะเป็นเลขคู่และเลขคี่ของจำนวนเท่าของความถี่ รูปคลื่นเหล่านี้จะมีประโยชน์มากในการวิเคราะห์สัญญาณต่างๆ โดยเฉพาะสัญญาณรบกวนการวิเคราะห์ทางคณิตศาสตร์ที่ค่อนข้างจะยุ่งยาก เป็นสิ่งจำเป็นในการตีความหมายของสัญญาณเหล่านี้

สัญญาณอีกชนิด ที่ไม่สามารถจัดประเภทได้ ไม่มีรูปแบบ ซึ่งมักจะเป็นสัญญาณรบกวนไม่สามารถนำมาใช้งานได้ เมื่อเกิดขึ้นก็จะไม่คำนึงถึง เกิดขึ้นชั่วคราวแล้วก็หายไป เมื่อไรที่คำนึงหรือมีผลกระทบต่อระบบ ความจำเป็นในการกำหนดรูปแบบ หรือจำลองแบบโดยอาศัยวิธีการทางคณิตศาสตร์ก็สามารถกระทำได้ และจัดการใส่รูปแบบเข้าไปเพื่อผลในการกำจัดหรือลดทอนสัญญาณที่ไม่พึงประสงค์ออกไป วิธีการทางคณิตศาสตร์สามารถวิเคราะห์หรือหารูปแบบที่เหมาะสมในการอธิบายพฤติกรรมของสัญญาณที่ไม่มีรูปแบบนี้ รวมถึงการหาค่าประกอบย่อยๆ ของสัญญาณนั้นได้ แต่ค่อนข้างที่จะยุ่งยากและไม่สามารถนำมาใช้ประโยชน์ได้ ในทางปฏิบัติจึงนิยมที่จะละเลยไป เว้นแต่มีความสำคัญต่อระบบ เช่น เกนฮิต (Gain Hit) หรือเฟสฮิต (Phase Hit) เป็นต้น เกนฮิตและเฟสฮิตนี้มักเกิดจากการกดปุ่มผิดจังหวะ ตามมาตรฐาน CCITT ระบุให้ผลที่เกิดตั้งแต่ 4 มิลลิวินาทีจนถึงนานนับชั่วโมงเข้าข่ายนี้ และกำหนดให้โมเด็มมีมาตรฐานว่าเกิดขึ้นได้กี่ครั้งในเวลา 15 นาที มีขนาดสูงสุดและเฟสสูงสุดเท่าใด ความยากของการจัดรูปแบบอยู่ที่ตำแหน่งที่กดปุ่มว่าอยู่ที่ใดของระบบ ณ จุดนั้นมีการไหลของข้อมูลและสัญญาณอย่างไร การจัดรูปแบบสามารถทำได้เมื่อผ่านการทดสอบทางแอนะล็อก บางครั้งก็อาจจะไม่เพียงพอ ต้องคู่ไปกับการทดสอบดิจิทัลเพื่อดูการทำงานในขั้นการเชื่อมต่อ บางครั้งอาจจะต้องทดสอบทางโปรโตคอล เพื่อดูการทำงานในขั้นประมวลผลการทำงานทั้งระบบ และวิเคราะห์กลับมาสู่รูปแบบของสัญญาณนั้นๆ วิธีการดังกล่าวจะได้กล่าวต่อไป

### 2.2.3 ความถี่และสายนำสัญญาณ

ตามปกติแล้วคลื่นที่ใช้ในการติดต่อสื่อสารได้แก่คลื่นแม่เหล็กไฟฟ้า ซึ่งมีความสัมพันธ์ระหว่างความถี่ ความยาวคลื่น และความเร็วในการเคลื่อนที่ของคลื่น ความถี่เราแทนด้วย  $f$  โดย

เอกส  $\text{เอาแรงดันตั้ง (V)}$  แล้วหารด้วยความยาวคลื่น ซึ่งเราจะแทนด้วยค่าของ  $(\lambda)$  ตามหลักการทางทฤษฎี เราต้องหาความถี่ในการใช้งาน โดยใช้สมการที่เราเคยได้ศึกษามาในการหาความถี่

ความยาวคลื่นนี้เป็นการวัดระหว่างจุดสูงสุด หรือจุดต่ำสุดของคลื่นทั้งสองคลื่นที่อยู่ติดต่อกัน ความเร็วของแสงในอากาศเป็น  $3 \cdot 10$  เมตรต่อวินาที ความถี่มีหน่วยเป็นเฮิรตซ์ ช่วงของความถี่เรียกว่า สเปกตรัม และในสเปกตรัมนี้แบ่งเป็นช่วงๆ เรียกว่า แบนด์ อันเป็นที่มาของความกว้างของช่วงคลื่นหรือแถบความถี่เรียกว่า แบนด์วิธ (Bandwidth) และมักจะพบเสมอเมื่อกล่าวถึงความถี่ สมมติว่าวิทยุ AM ส่งกระจายเสียงที่ความถี่ 1 เมกะเฮิรตซ์ ความยาวคลื่นในการส่งเท่ากับ 300 เมตร เป็นต้น ตารางที่ 2.1 แสดงสเปกตรัมที่ความถี่ต่างๆ ตามมาตรฐานข้อตกลงระหว่างประเทศ ซึ่งกำหนดโดยองค์การโทรคมนาคมนานาชาติ การเคลื่อนที่ของคลื่นเป็นไปตามหลักฟิสิกส์ คือ คลื่นเดินทางเป็นเส้นตรง มีการสะท้อนของสัญญาณ มีการหักเหของสัญญาณ รวมไปถึงการที่มีสัญญาณรบกวนมาเกี่ยวข้องด้วย คุณสมบัติต่างๆ ของคลื่นขึ้นอยู่กับความถี่ที่ใช้งาน ความถี่ที่สูงจะมีความยาวคลื่นที่สั้นกว่า แต่มีอำนาจทะลุทะลวงได้ดีกว่า ในขณะที่ความถี่ต่ำจะมีความยาวคลื่นมากกว่า ทำให้คลื่นเดินทางได้ไกลกว่าสัญญาณข่าวสารที่เป็นคำพูดหรือเสียง จึงมีแบนด์วิธของความถี่ที่ต่ำสุดค่าหนึ่งในการนำพาข้อมูลข่าวสารรูปแบบอื่นๆ ของสัญญาณข่าวสารที่ใช้ในระบบโทรคมนาคม เช่น โทรศัพท์ โทรทัศน์ คนตรี ข้อมูล เป็นต้น มีแบนด์วิธต่ำสุดที่แตกต่างกันไป แบนด์วิธของสัญญาณข่าวสารเป็นส่วนสำคัญในการออกแบบระบบในด้านของวงจร ละตัวกลางสำหรับส่งสัญญาณ ความรู้เรื่องความถี่เป็นสิ่งสำคัญมาก เนื่องจากการประยุกต์ใช้งานที่ความถี่ย่านต่างๆ ในระบบโทรคมนาคม จะกล่าวถึงรายละเอียดเกี่ยวกับคุณสมบัติของคลื่นที่ความถี่นั้นๆ ในหัวข้อการประยุกต์ใช้งานของเทคโนโลยีนั้นๆ

ตารางที่ 2.1 สเปกตรัมที่ความถี่ต่างๆ

| แถบความถี่     | ชื่อเรียก ของความถี่     | ชื่อย่อ |
|----------------|--------------------------|---------|
| ต่ำกว่า 3 kHz  | Extremely Low Frequency  | ELF     |
| 3 - 30 kHz     | Very Low Frequency       | VLF     |
| 30 - 300 kHz   | Low Frequency            | LF      |
| 300 - 3000 kHz | Medium Frequency         | MF      |
| 3 - 30 MHz     | High Frequency           | HF      |
| 30 - 300 MHz   | Very High Frequency      | VHF     |
| 300 - 3000 MHz | Ultra High Frequency     | UHF     |
| 3 - 30 GHz     | Super High Frequency     | SHF     |
| 30 - 300 GHz   | Extremely High Frequency | EHF     |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำ

คลื่นที่กล่าวถึงในการนำไปใช้งานจะมีรูปแบบเป็นคลื่นแม่เหล็กไฟฟ้า ซึ่งมีคุณสมบัติที่ความถี่ต่างๆ ไม่เหมือนกัน และมีผลต่อช่องสัญญาณในการติดต่อสื่อสารด้วย คุณสมบัติและตัวอย่างการใช้ประโยชน์ที่ความถี่ต่างๆ พอกล่าวย่อๆ ได้ดังนี้

ความถี่ ELF เป็นย่านของความถี่ต่ำมาก ได้แก่ เสียงคนตรี เสียงสัตว์ หรือเสียงของมนุษย์ บางส่วน ความถี่ VLF และ LF สมัยก่อนจะใช้กับวิทยุโทรเลข แต่ในปัจจุบันใช้กับงานบางชนิดเท่านั้น ความถี่ย่าน MF นี้ใช้ในการส่งกระจายเสียงในระบบ AM และ HF หรือที่เรียกอีกชื่อว่าคลื่นสั้นใช้ในการส่งกระจายเสียงในระบบ AM แบบคลื่นสั้น และใช้กับกิจการวิทยุสมัครเล่น คุณสมบัติของคลื่นในความถี่ย่าน MF และ HF คือเมื่อคลื่นเดินทางไปในชั้นของบรรยากาศชั้นสูงสุดที่ห่อหุ้มโลกคือไอโอโนสเฟียร์ (Ionosphere) คลื่นก็จะสะท้อนกลับมา อันเป็นเหตุให้คลื่นเดินทางได้ไกลมาก แต่มีการสูญเสียพลังงานพอสมควร เนื่องจากการเปลี่ยนแปลงของชั้นบรรยากาศที่มีผลต่อคลื่น ทำให้ความแรงของสัญญาณมีค่าไม่คงที่ ส่วนคลื่นความถี่ VHF และ UHF เป็นคลื่นที่สามารถเดินทางทะลุผ่านชั้นของบรรยากาศไปได้ เนื่องจากมีความถี่สูง การติดต่อสื่อสารจึงอยู่ในลักษณะเป็นแนวเส้นตรง (Line of Sight) จึงจะทำการสื่อสารได้ การประยุกต์ใช้งานมีอยู่มากมาย เช่น การส่งสัญญาณโทรทัศน์ วิทยุสื่อสาร เซลล์มือถือ หรือแม้กระทั่งดาวเทียมวงโคจรต่ำ เป็นต้น จะกล่าวถึงรายละเอียดของความถี่เมื่อไปถึงหัวข้อการประยุกต์การใช้งานของเทคโนโลยีนั้นๆ

อีกย่านความถี่ที่น่าสนใจคือ ไมโครเวฟ ความหมายของคลื่นไมโครเวฟมักหมายถึงความถี่ที่สูงเกิน 1 เมกะเฮิร์ตซ์ ช่วงความถี่ตั้งแต่ 1 ถึง 30 กิกะเฮิร์ตซ์ จะมีความยาวคลื่น 30 เซนติเมตรลงมาถึง 1 เซนติเมตรเมตรตามลำดับ ดังนั้น คลื่นความถี่ตั้งแต่ 30 กิกะเฮิร์ตซ์แต่ไม่เกิน 300 กิกะเฮิร์ตซ์ ซึ่งนิยมเรียกอีกชื่อว่า คลื่นมิลลิเมตรเวฟ ตามขนาดความยาวคลื่น ซึ่งพออนุโลมว่าเป็นคลื่นไมโครเวฟ การประยุกต์ใช้งานของคลื่นไมโครเวฟมีตั้งแต่การส่งสัญญาณไมโครเวฟ โดยแบ่งเป็นระบบส่งสัญญาณและอุปกรณ์วงจรที่ใช้ความถี่ในย่านนี้ไปจนถึงอุปกรณ์วงจรในยานอวกาศ คลื่นมิลลิเมตรเวฟนิยมใช้ในกิจการของความมั่นคงทางด้านอวกาศและการทหารเป็นหลัก จุดเด่นของคลื่นที่ความถี่สูง คือยิ่งความถี่ยิ่งสูงเท่าไร สายอากาศที่ใช้จะมีขนาดเล็กลงเท่านั้น แต่มีข้อเสียเกี่ยวกับสภาวะภูมิอากาศ เช่น การเปลี่ยนแปลงของสภาพอากาศหรือฝน เป็นต้น ตัวอย่างที่เห็นได้ชัดจากผลกระทบดังกล่าวคือ งานสายอากาศในการสื่อสารผ่านดาวเทียม งานรับสัญญาณย่านความถี่ 4 ถึง 6 กิกะเฮิร์ตซ์ จะต้องใช้จานใหญ่กว่างานรับสัญญาณย่านความถี่ 12 ถึง 14 กิกะเฮิร์ตซ์เสมอ คุณสมบัติต่างๆ เกี่ยวกับคลื่นในย่านไมโครเวฟจะกล่าวถึงในหัวข้อไมโครเวฟและการสื่อสารดาวเทียม

การที่จะติดต่อสื่อสารได้นั้น ข้อมูลข่าวสารในรูปของสัญญาณต่างๆ ต้องเคลื่อนที่ผ่านช่องเอกสารนี้เป็นเอกสารลิขสิทธิ์ไว้สำหรับใช้ในงานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้เผยแพร่โดยไม่ได้รับอนุญาต การนำเอกสารนี้เป็นเอกสารลิขสิทธิ์ไว้สำหรับใช้ในงานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้เผยแพร่โดยไม่ได้รับอนุญาต การนำเอกสารนี้เป็นเอกสารลิขสิทธิ์ไว้สำหรับใช้ในงานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้เผยแพร่โดยไม่ได้รับอนุญาต

เป็นประเภทคลื่นวิทยุที่ความถี่ต่างๆ โดยจะกล่าวต่อไป เมื่อถึงการประยุกต์ใช้งานในหัวข้อนั้นๆ รวมถึงผลกระทบของชั้นบรรยากาศและสภาวะแวดล้อมสำหรับคลื่นที่ความถี่นั้น ส่วนแบบมีสายก็จะได้แก่สายนำสัญญาณ สายนำสัญญาณพ่วงจะจำแนกได้ 3 ประเภทคือ สายทองแดงตีเกลียวคู่ (Twisted Pair) สายโคแอกเชียล (Coaxial) และสายใยแก้วนำแสง (Optic Fiber) สายนำสัญญาณในแต่ละชนิดจะมีข้อดีข้อเสียแตกต่างกัน เช่น ราคา การทนต่อสัญญาณรบกวน รวมไปถึงความสูญเสียของสัญญาณ ความง่ายยากในการใช้งาน รวมถึงความเหมาะสม เป็นต้น นอกจากนี้ยังมีสายนำสัญญาณที่มีลักษณะเป็นท่อเรียกว่า ท่อนำคลื่น (Waveguide) ท่อนำคลื่นนี้มีลักษณะเป็นท่อสี่เหลี่ยม ใช้ความกว้างและความยาวของท่อในการคำนวณกำหนดรูปแบบและคุณสมบัติของคลื่น เหมาะที่จะใช้งานในย่านความถี่สูง ซึ่งจะทดแทนคุณสมบัติบางประการของเคเบิล เช่น การสูญเสียพลังงานของคลื่นพื้นผิว และการสูญเสียการเป็นสื่อนำไฟฟ้า เป็นต้น

#### 2.2.4 ความเหมาะสม

เทคโนโลยีแต่ละชนิดจะมีความเหมาะสมในตัวเอง การที่จะบอกว่าเทคโนโลยีนั้นดีกว่าเทคโนโลยีอย่างอื่นเชิงเป็นไปไม่ได้ ดังตัวอย่างเช่น การสื่อสารผ่านดาวเทียมเมื่อใช้กับพื้นที่ที่เป็นหมู่เกาะ หุบเขา หรือพื้นที่ที่มีการติดต่อสื่อสารน้อยจะเป็นเทคโนโลยีที่เหมาะสมกว่าการลากสายเคเบิลอย่างแน่นอน แต่กรณีใช้งานปกติในพื้นที่ที่ใกล้เคียงกัน การลากสายเคเบิลย่อมดีกว่า เนื่องจากสัญญาณผ่านดาวเทียมต้องใช้เวลาในการเดินทางของสัญญาณ จากสถานีส่งส่งสัญญาณไปยังดาวเทียม และมายังสถานีรับสัญญาณทำให้เกิดการหน่วงเวลา หรือเมื่อการเปรียบเทียบกันระหว่างแบบมีสายนำสัญญาณด้วยกัน ในบางกรณีการใช้สายใยแก้วนำแสงเป็นกรณีดี เนื่องจากมีคุณสมบัติทนต่อสัญญาณรบกวน และสามารถใช้ในการส่งสัญญาณดิจิทัล แต่ในบางกรณีก็ไม่ดี เช่น ถ้ามีการใช้งานน้อยจะทำให้ราคาแพงและเชื่อมต่อลำบาก ซึ่งสายโคแอกเชียลหรือสายทองแดงตีเกลียวคู่จะมีสามารถใช้งานได้ดีกว่า ราคาถูกกว่า และการเชื่อมต่อง่ายกว่าด้วย

### 2.3 ระบบสื่อสารแอนะล็อก

การสื่อสารแอนะล็อกเป็นระบบที่ออกแบบให้ส่งสัญญาณแอนะล็อก เช่น สัญญาณเสียง แต่ได้มีการพัฒนาจนประยุกต์ให้สามารถส่งข่าวสารได้ด้วยในปัจจุบัน ปัญหาสำคัญสำหรับการสื่อสารแอนะล็อกก็คือเรื่องสัญญาณรบกวน ซึ่งในบางครั้งอาจทำให้ระบบไม่สามารถทำงานได้เลย ดังนั้นการสื่อสารดิจิทัลจึงได้เริ่มเข้ามามีแทนที่ อย่างไรก็ตาม การสื่อสารดิจิทัลก็ยังไม่สามารถแทนที่การสื่อสารแบบแอนะล็อกได้ทั้งหมด เนื่องจากสัญญาณในธรรมชาติทั้งหมดเป็นสัญญาณแอนะล็อกจึงยังคงเห็นการพัฒนาของการสื่อสารแบบแอนะล็อกอยู่ในปัจจุบัน ซึ่งรูปแบบการสื่อสารที่มีประสิทธิภาพของการทำงานบางรูปแบบก็ยังคงใช้อยู่ในปัจจุบัน เช่น การมอดูเลตแอมพลิจูด

(Amplitude Modulation หรือ AM) การมอดูเลตความถี่ (Frequency Modulation หรือ FM) สัญญาณโทรศัพท์พื้นฐาน เป็นต้น นอกจากนี้ยังมีสาเหตุอีกหลายประการที่ระบบแอนะล็อกยังคงมีการใช้อยู่ได้แก่

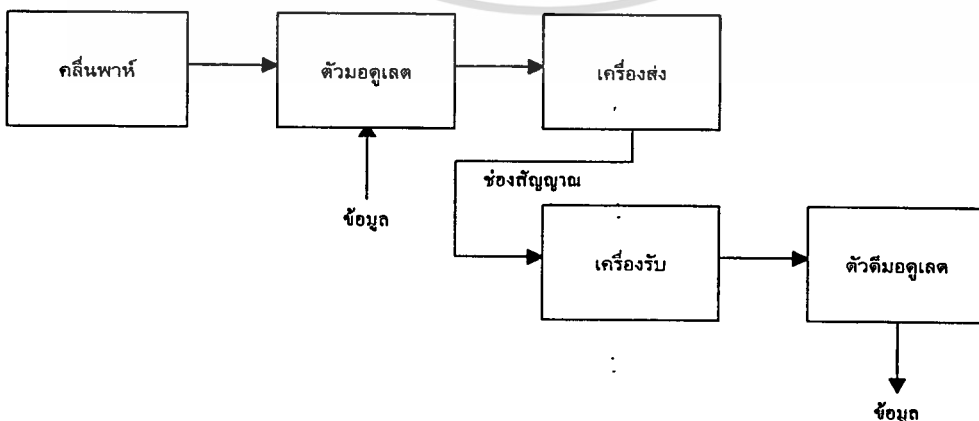
1) สัญญาณดิจิทัลเป็นกรณีพิเศษของสัญญาณแอนะล็อก สัญญาณดิจิทัลที่เป็นสัญญาณแอนะล็อกถูกกำหนดให้มีระดับแรงดันคงที่ที่ระดับต่างๆ ตามสัญลักษณ์ของข่าวสาร นั้นหมายความว่าระบบที่ทำงานกับสัญญาณแอนะล็อกสามารถใช้งานกับสัญญาณดิจิทัลได้

2) ถึงแม้ระบบที่ส่งสัญญาณจะเป็นดิจิทัล แต่สัญญาณก็ยังเป็นแอนะล็อก เช่น โครงข่ายโทรศัพท์ที่ใช้ส่งสัญญาณดิจิทัลของเครื่องคอมพิวเตอร์หรือการส่งสัญญาณไมโครเวฟ เป็นต้น

3) โครงข่ายที่ใช้กันอยู่ในปัจจุบันเกือบทั้งหมดเป็นโครงข่ายแอนะล็อก ซึ่งสามารถใช้งานได้กับทั้งสัญญาณแอนะล็อกและดิจิทัล ทำให้ประหยัดค่าใช้จ่ายในการพัฒนาระบบใหม่ หากระบบนั้นสามารถใช้งานกับโครงข่ายแอนะล็อกที่มีอยู่ในปัจจุบันได้

### 2.3.1 การมอดูเลต

การมอดูเลตเป็นการผสมสัญญาณของข้อมูลเข้ากับสัญญาณอีกสัญญาณหนึ่งเรียกว่า คลื่นพาห์ ซึ่งสัญญาณนี้มีความถี่ที่เหมาะสมกับช่องสัญญาณนั้นๆ เพื่อให้ข้อมูลที่ส่งเข้าไปในช่องสัญญาณเดินทางได้ไกลมากขึ้น รูปที่ 2.2 รูปแบบของการสื่อสารในการรับส่งสัญญาณอย่างง่าย โดยคลื่นพาห์ผสมสัญญาณข้อมูลที่ควมอดูเลต แล้วส่งไปที่เครื่องส่ง จากเครื่องส่งไปยังเครื่องรับจะเป็นช่องสัญญาณสำหรับลำเลียงสัญญาณผสมนี้ ซึ่งสัญญาณผสมจากเครื่องรับจะไปเข้าตัวดีมอดูเลตเพื่อแยกสัญญาณข้อมูลออกมา การเลือกวิธีการมอดูเลตขึ้นอยู่กับปัจจัยหลายประการ เช่น ชนิดของสัญญาณ แบนด์วิดท์ ประสิทธิภาพของระบบที่ต้องการ และความต้านทานต่อสัญญาณรบกวน เป็นต้น เมื่อมีการมอดูเลตที่ภาคส่ง การที่จะรับข้อมูลได้นั้นต้องมีการดีมอดูเลตที่ภาครับด้วย

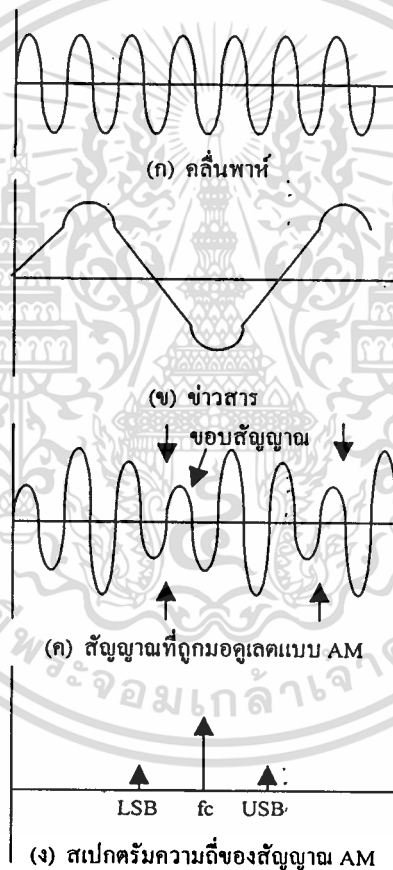


รูปที่ 2.2 การสื่อสารในการรับส่งสัญญาณ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้ในการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การมอดูเลตที่ใช้อยู่ในปัจจุบันมีอยู่ 4 วิธีได้แก่

1) การมอดูเลตแอมพลิจูด หรือที่เรียกว่า การมอดูเลตแบบ AM และเรียกสัญญาณที่ถูกมอดูเลตนี้ว่าสัญญาณ AM วิธีนี้ แอมพลิจูดของคลื่นพาห์จะเปลี่ยนแปลงตามสัญญาณของข้อมูลที่เข้ามา ดังรูปที่ 2.3 การมอดูเลตแบบ AM โดยรูป (ก) เป็นคลื่นพาห์ รูป (ข) เป็นข่าวสาร (ค) เป็นสัญญาณที่ถูกมอดูเลตแบบ AM และมีสเปกตรัมของความถี่ในรูป (ง) การมอดูเลตแบบ AM เป็นวิธีที่ง่ายที่สุดในการมอดูเลต แต่คุณภาพของสัญญาณไม่ดี มีความต้านทานสัญญาณรบกวนต่ำ เหมาะกับข้อมูลที่ไม่ต้องการคุณภาพมากนัก เช่น สัญญาณเสียง เป็นต้น



โดยที่  $f_c$  เป็นความถี่ของคลื่นพาห์ (Carrier Frequency)

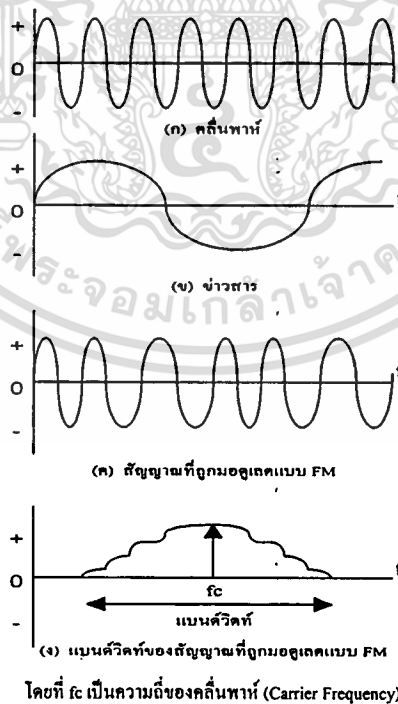
LSB เป็นความถี่แบบต่ำ (Lower Sideband Frequency)

USB เป็นความถี่แบบสูง (Upper Sideband Frequency)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้ภายในหน่วยงานเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

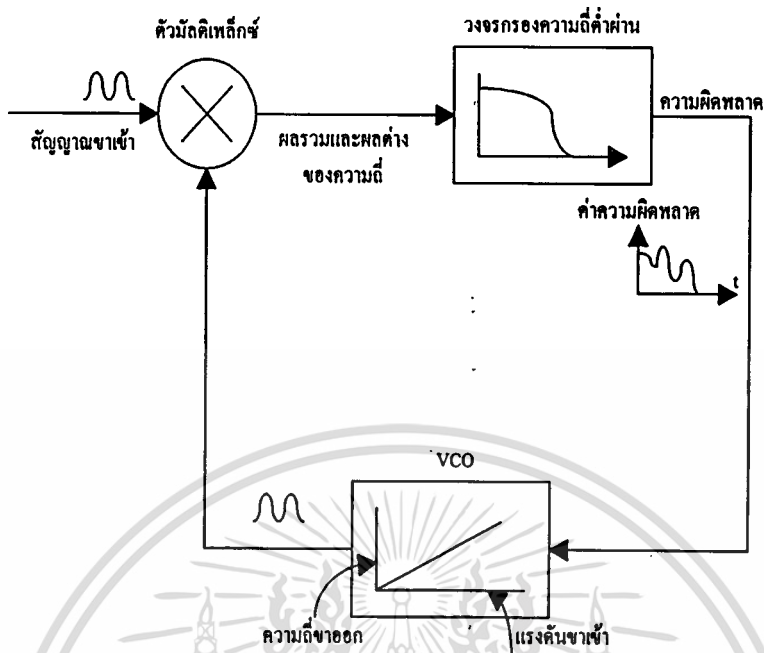
จากรูปที่ 2.3 จะเห็นว่าพลังงานส่วนใหญ่จะอยู่ที่คลื่นพาห้ไม่ได้อยู่ที่สัญญาณของข้อมูล ทำให้สิ้นเปลืองพลังงานในการส่งสัญญาณมาก เช่น เครื่องส่งขนาด 10 กิโลวัตต์อาจจะเป็นพลังงานของข้อมูลเพียง 4 ถึง 5 กิโลวัตต์ เท่านั้น ดังนั้น จึงได้มีการพัฒนาการมอดูเลตแบบ AM เป็นการมอดูเลตแอมพลิจูดระงับคลื่นพาห้ (Amplitude Modulation Suppress Carrier : AM-SC) ซึ่งเป็นการมอดูเลตแบบ AM ที่มีการกำจัดคลื่นพาห้ออกไปหลังจากที่สัญญาณข้อมูลได้ทำการมอดูเลตแล้ว ทำให้ประหยัดพลังงานในการส่งสัญญาณขึ้นมา

2) การมอดูเลตความถี่ หรือที่เรียกกันสั้นๆ ว่าการมอดูเลตแบบ FM วิธีนี้เป็นวิธีการเปลี่ยนแปลงความถี่ของคลื่นพาห้ตามสัญญาณของข้อมูลที่เข้ามา ดังรูปที่ 2.4 การมอดูเลตแบบ FM ให้คุณภาพของสัญญาณที่ดีกว่าการมอดูเลตแบบ AM แต่ระบบจะซับซ้อนกว่า อย่างไรก็ตามจากความก้าวหน้าทางเทคโนโลยีในปัจจุบัน ทำให้การมอดูเลตแบบ FM สามารถทำได้ง่ายขึ้น โดยใช้เทคนิคเฟสล็อกคัล (Phase Lock Loop) ดังรูปที่ 2.5 สัญญาณในการมอดูเลตแบบ FM จะได้จากสัญญาณขาออกของระบบ VCO หรือออสซิลเลเตอร์ควบคุมแรงดัน ซึ่งได้รับสัญญาณความผิดพลาดจากวงจรกรองความถี่ต่ำผ่าน เพื่อพยายามที่จะปรับความถี่และเฟสด้านขาออกกับค่าสัญญาณขาเข้า โดยการคูณกันเพื่อให้ได้ผลต่างระหว่างสัญญาณทั้งสองคงที่



รูปที่ 2.4 การมอดูเลตแบบ FM

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการศึกษาเท่านั้น ไม่สามารถนำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

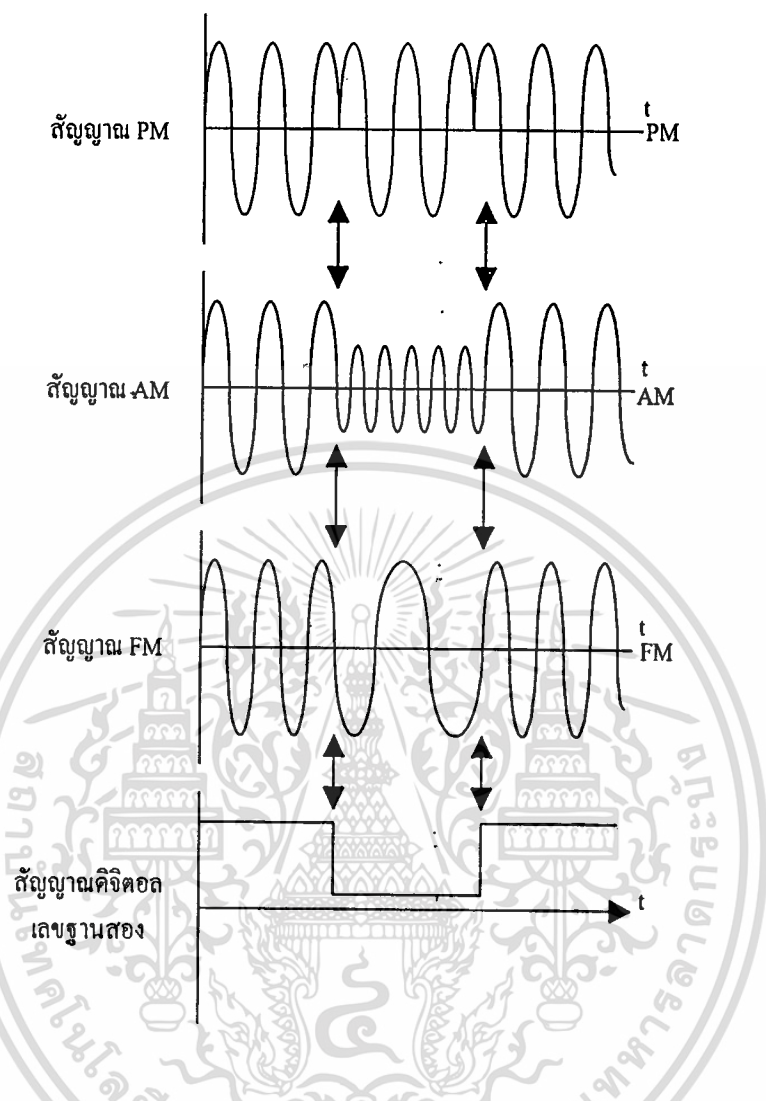


รูปที่ 2.5 ตัวอย่างการใช้เทคนิคเฟสล็อก

3) การมอดูเลตเฟส (Phase Modulation ; PM) หรือที่เรียกกันสั้นๆ ว่าการการมอดูเลตแบบ PM เป็นการมอดูเลตที่ใช้การเปลี่ยนแปลงเฟสของคลื่นพาห้ตามสัญญาณข้อมูลที่เข้ามา ทั้งคุณภาพของสัญญาณและความซับซ้อนไม่ค่อยแตกต่างจากการมอดูเลตแบบ FM ข้อแตกต่างระหว่างการมอดูเลตแบบ FM กับการมอดูเลตแบบ PM คือการมอดูเลตแบบ PM ใช้คลื่นพาห้เพียงความถี่เดียว การมอดูเลตและการดีมอดูเลตสามารถทำได้และประหยัดกว่า แต่ไม่ได้หมายความว่า จะซับซ้อนน้อยกว่า จึงไม่ค่อยพบการใช้การมอดูเลตแบบ PM กับสัญญาณแอนะล็อกเท่าใด แต่มักพบการมอดูเลตแบบ PM ใช้กับสัญญาณดิจิทัล เนื่องจากสัญญาณดิจิทัลมีการเปลี่ยนแปลงแบบทันทีทันใด การเปลี่ยนแปลงความถี่แบบทันทีทันใดในการมอดูเลตแบบ FM ทำได้ยากมาก โดยเฉพาะการดีมอดูเลตเฟส

การเปลี่ยนแปลงเฟสในการมอดูเลตแบบ PM จะทำได้ง่ายกว่าการมอดูเลตแบบ FM ซึ่งจะเห็นได้จากรูปที่ 2.6 การเปรียบเทียบสัญญาณแบบ PM, AM, FM กับสัญญาณดิจิทัลเลขฐานสอง และตารางที่ 2.2 การเปรียบเทียบการมอดูเลตแบบต่างๆ ทางด้านแบนด์วิดท์ รูปแบบสเปกตรัม ประสิทธิภาพของกำลังงานที่ใช้การมอดูเลตและการดีมอดูเลต รวมถึงความต้านทานต่อสัญญาณรบกวน เป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



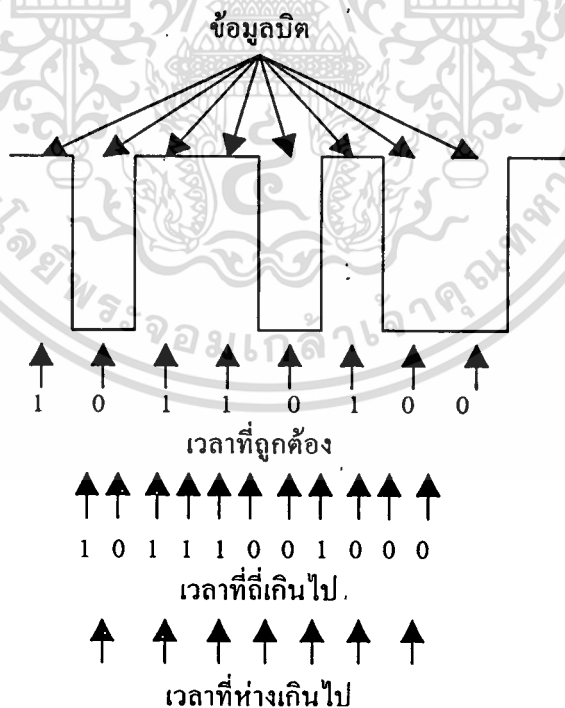
รูปที่ 2.6 การเปรียบเทียบสัญญาณ PM, AM และ FM กับสัญญาณดิจิทัลเลขฐานสอง

ตารางที่ 2.2 การเปรียบเทียบคุณลักษณะของการมอดูเลตแบบ AM, FM และ PM

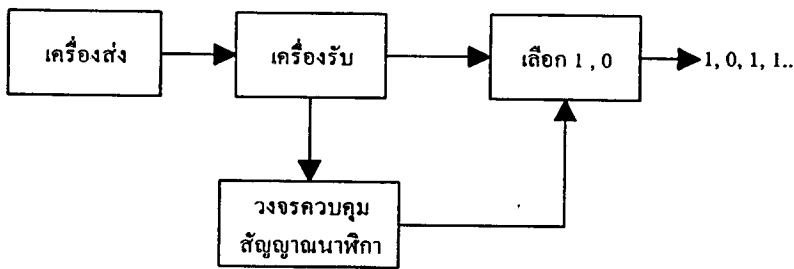
| คุณลักษณะการมอดูเลตแบบต่างๆ | AM    | FM      | PM      |
|-----------------------------|-------|---------|---------|
| แบนด์วิธ                    | แคบ   | กว้าง   | กว้าง   |
| รูปแบบสเปกตรัม              | ง่าย  | ซับซ้อน | ซับซ้อน |
| ประสิทธิภาพของกำลังงาน      | ไม่ดี | ดี      | ดี      |
| การมอดูเลตและดีมอดูเลต      | ง่าย  | ยาก     | ปานกลาง |
| ความต้านทานต่อสัญญาณรบกวน   | ต่ำ   | สูง     | สูง     |

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้เพื่อการใช้งานเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้วยวิธีการ  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4) การมอดูเลตและการดีมอดูเลตแบบซิงโครนัสและอะซิงโครนัส (Synchronous and Asynchronous Modulation and Demodulation) เมื่อสัญญาณถูกส่งผ่านทางช่องสัญญาณสู่ด้านรับ จะมีการแยกสัญญาณข้อมูลออกจากสัญญาณที่ถูกมอดูเลตก่อนถึงด้านรับ ซึ่งเรียกว่าการดีมอดูเลต ในระบบสื่อสารดิจิทัลสัญญาณข้อมูลมักเป็น “0” และ “1” วงจรรับพิจารณาว่าเป็น “0” หรือ “1” อัตราข้อมูลเป็นอย่างไร สัญญาณนาฬิกา ที่ถูกผลิตขึ้นจากทางด้านส่งและด้านรับต่างกันเล็กน้อย เป็นผลให้การดีมอดูเลตเร็วไปหรือช้าไปดังรูปที่ 2.7 เรียกว่า ระบบอะซิงโครนัส ส่วนรูปที่ 2.7 (ข) เป็นการแก้ปัญหาโดยการนำวงจรควบคุมสัญญาณนาฬิกาทางด้านรับ ช่วยให้ได้เวลาที่เหมาะสม ซึ่งเรียกว่าระบบซิงโครนัส เหตุที่ต้องกล่าวถึงระบบเหล่านี้ เนื่องจากการพิจารณาการมอดูเลตทั้งสามแบบสามารถทำได้ เพียงแต่การพิจารณาทางหน่วยของเวลาไม่เหมาะสมกับการมอดูเลตแบบ AM เนื่องจากยุ่งยากมากแต่ก็ทำได้ การออกแบบระบบมักจะกล่าวถึงระบบซิงโครนัสเพื่อการประโยชน์จากช่องสัญญาณ ซึ่งมักจะมีผลกระทบจากสัญญาณรบกวนเสมอในการประยุกต์ใช้งานของระบบ การมอดูเลต การดีมอดูเลต การมัลติเพล็กซ์ และการดีมัลติเพล็กซ์ มักจะมีการผสมผสานระบบกัน ไประหว่างระบบแอนะล็อกและระบบดิจิทัล ตัวอย่างที่เห็นได้ชัดเจน ได้แก่ เทคโนโลยีโทรศัพท์ทางไกล เป็นต้น



เอกสารนี้เป็นเอกสารที่สงวนไว้รูปที่ 2.7 การเปลี่ยนของบิตในระบบอะซิงโครนัสนำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.8 การแก้ปัญหาโดยใช้สัญญาณนาฬิกาทางด้านการรับช่วยในระบบชิงโครนัส

### 2.3.2 การมัลติเพล็กซ์

เนื่องจากความต้องการสื่อสารมีจำนวนมาก ในขณะที่ช่องสัญญาณมีจำนวนจำกัด ทำให้ต้องมีการจัดสรรช่องสัญญาณให้กับผู้ใช้จำนวนมาก การมัลติเพล็กซ์ จึงเป็นวิธีที่จะจัดสรรช่องสัญญาณให้กับผู้ใช้เหล่านั้น ซึ่งมีหลายวิธี การที่จะเลือกใช้วิธีใดนั้นขึ้นอยู่กับองค์ประกอบหลายอย่าง เช่น ธรรมชาติของช่องสัญญาณ จำนวนผู้ใช้ หรือแหล่งกำเนิดแบนด์วิดท์ที่ใช้ได้ หรือ องค์ประกอบทางเศรษฐศาสตร์ บางวิธีมีการลงทุนต่ำ แต่คุณภาพสัญญาณพอใช้ได้ บางวิธีมีคุณภาพสัญญาณดี แต่ลงทุนสูง การมัลติเพล็กซ์มีใช้กันอยู่ 3 วิธี ได้แก่

1) การมัลติเพล็กซ์แบบแบ่งเส้นทาง (Space Division Multiplexing : SDM) เป็นวิธีการมัลติเพล็กซ์เริ่มแรกที่ใช้กันและเป็นวิธีที่ง่ายที่สุด การมัลติเพล็กซ์แบบแบ่งเส้นทางมีเส้นทางระหว่างตัวส่งและตัวรับหลายเส้นทาง ผู้ใช้สามารถใช้ความถี่เดียวกันในการส่งสัญญาณได้ในเวลาเดียวกัน สัญญาณที่ส่งในระบบนี้จะมีลักษณะของสัญญาณเป็นสัญญาณเบสแบนด์ ในการมอดูเลตแอมพลิจูด ตัวอย่างที่เห็นได้ชัดเจนคือโครงข่ายโทรศัพท์พื้นฐาน

ข้อได้เปรียบและข้อจำกัดของการมัลติเพล็กซ์แบบแบ่งเส้นทาง ได้แก่

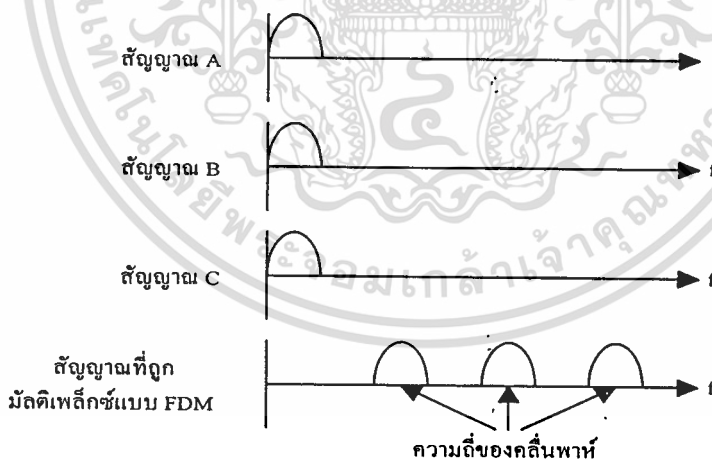
1. การมัลติเพล็กซ์แบบแบ่งเส้นทาง (SDM) เป็นระบบที่สร้างได้ง่าย
2. หากเกิดปัญหาที่เส้นทางใดเส้นทางหนึ่ง เช่น สายขาด ปัญหานั้นจะไม่ส่งผลกระทบต่อผู้อื่นๆ ทำให้การมัลติเพล็กซ์แบบแบ่งเส้นทางถูกใช้งานที่สำคัญ เช่น การควบคุมในระบบที่ต้องการความเชื่อถือได้มากๆ หรือระบบที่มีความเสี่ยงมากจะใช้วิธีส่งสัญญาณเดียวกันไปหลายๆ เส้นทาง
3. โดยหลักการแล้ว การเพิ่มหรือการลดจำนวนผู้ใช้สามารถทำได้ง่าย หรือแม้แต่การเพิ่มความจุของระบบ

4. ประสิทธิภาพของระบบการทำงานสามารถที่จะทำนายได้โดยมีความถูกต้อง สามารถคำนวณจำนวนอุปกรณ์ที่ใช้ได้ ทำให้สามารถหาปริมาณการลงทุนที่เหมาะสม รวมทั้งการให้บริการที่มีประสิทธิภาพ

5. สามารถติดต่อแบบดูเพล็กซ์เต็ม (Full-Duplex) ได้

6. มีการลงทุนที่สูง โดยเฉพาะการลงทุนด้านสายเคเบิล และมีสัญญาณรบกวนระหว่างสายนำสัญญาณ

2) การมัลติเพล็กซ์แบบแบ่งความถี่ (Frequency Division Multiplexing ; FDM) เป็นวิธีที่ใช้กันทั้งระบบที่มีสายและระบบคลื่นวิทยุ หลักการของการมัลติเพล็กซ์แบบแบ่งความถี่ คือการรวมสัญญาณจากแหล่งต่างๆ สัญญาณเหล่านี้สามารถใช้เส้นทางร่วมกันได้ ดังรูปที่ 2.9 และมีการส่งสัญญาณในรูปที่ 2.10 ดังนั้นในแต่ละเส้นทางจึงประกอบด้วย สัญญาณจากแหล่งกำเนิดอยู่มากมายกระจายอยู่ในความถี่ต่างๆ การใช้งานในมัลติเพล็กซ์แบบแบ่งความถี่ ไม่ได้หมายความว่าใช้ได้กับการมอดูเลตความถี่ เท่านั้น แต่สามารถใช้งานได้กับการมอดูเลตทั้งสามแบบกล่าวคือการมอดูเลตแอมพลิจูด การมอดูเลตความถี่ และการมอดูเลตเฟส ตัวอย่างที่เห็นได้ชัดเจนคือการส่งกระจายเสียงแบบมอดูเลตแอมพลิจูดใช้การมอดูเลตแอมพลิจูด และมีการมัลติเพล็กซ์แบบแบ่งความถี่ ทำให้ระบบกระจายเสียงแบบมอดูเลตแอมพลิจูด มีสถานีต่างๆ ที่ความถี่ตั้งแต่ 550 ถึง 1,600 กิโลเฮิรตซ์ หรือการส่งกระจายสัญญาณเสียงแบบมอดูเลตความถี่ มีการมอดูเลตความถี่และมีการมัลติเพล็กซ์แบบแบ่งความถี่ แนวคิดที่ว่าการมัลติเพล็กซ์แบบแบ่งความถี่ จะเท่ากับผลรวมของแนวคิดของแต่ละแหล่งกำเนิดรวมกับแนวคิดของแถบคุม (Guardband)

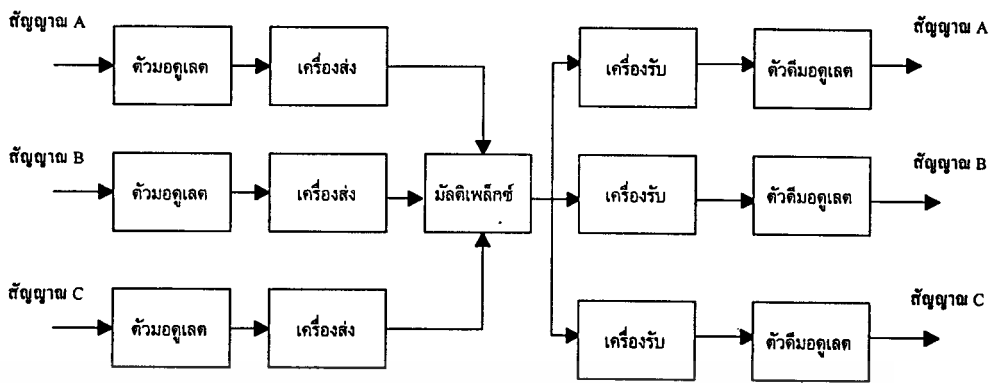


รูปที่ 2.9 การมัลติเพล็กซ์สัญญาณความถี่ต่างๆ ในระบบ FDM

3) การมัลติเพล็กซ์แบบแบ่งเวลา (Time Division Multiplexing ; TDM) เป็นวิธีที่เพิ่งจะได้

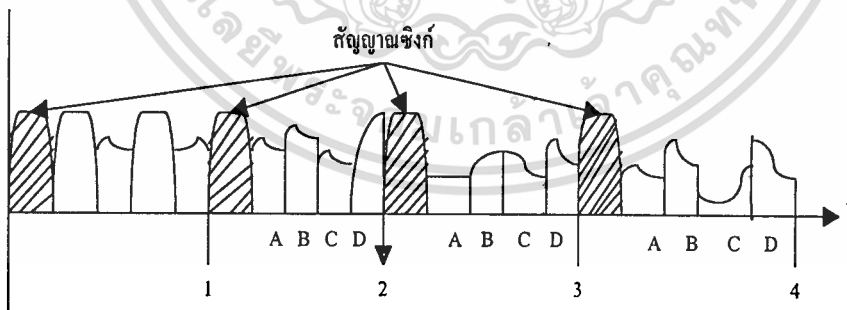
รับการพัฒนาได้ไม่นาน การมัลติเพล็กซ์แบบแบ่งเวลา จะใช้เส้นทางเพียงเส้นทางเดียว และคลื่นพาหความถี่เดียวเท่านั้น ผู้ใช้ แต่ละคนจะได้รับการจัดสรรเวลาในการเข้าใช้ช่องสัญญาณส่วนสวิตซ์

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งหากนำไปใช้



รูปที่ 2.10 การส่งสัญญาณในระบบ FDM

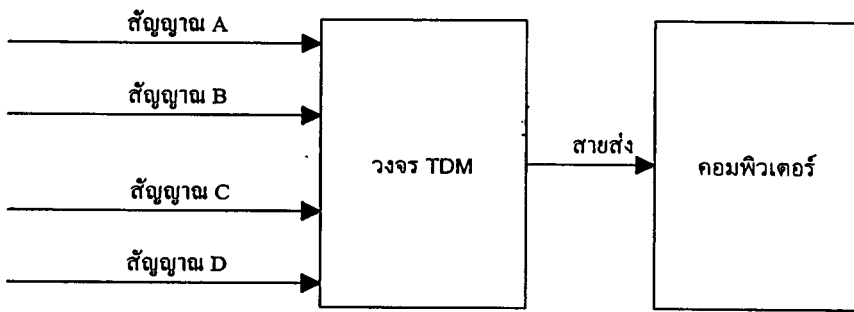
กลางหรือตัวมัลติเพล็กซ์ จะทำหน้าที่ในการกำหนดให้ผู้ใช้งานสามารถเข้าใช้ช่องสัญญาณ ปัญหาหลักของการมัลติเพล็กซ์แบบแบ่งเวลา คือชิงโครไนซ์ เนื่องจากสวิตช์ที่ตัวส่งและสวิตช์ที่ตัวรับต้องทำงานไปพร้อมกันจึงจะได้รับข้อมูลที่ถูกต้อง แต่ในการใช้งานจริงมีเวลาประวิงของช่องสัญญาณ จึงทำให้เกิดปัญหาขึ้น โดยเฉพาะหากสวิตช์ด้วยความเร็วสูง การแก้ไขได้คือการส่งสัญญาณชิงโครไนซ์เข้าไประหว่างช่องสัญญาณที่เป็นข้อมูลของผู้ใช้ ดังรูปที่ 2.11 การมัลติเพล็กซ์สัญญาณที่เวลาต่างๆ โดยใช้สัญญาณชิงค์ ช่วยในระบบ และการส่งสัญญาณในลักษณะใช้สายส่งสัญญาณเพียงเส้นเดียวโดยสัญญาณหลายๆ สัญญาณจะไปรวมกันที่วงจร TDM ก่อน ดังรูปที่ 2.12



รูปที่ 2.11 การมัลติเพล็กซ์สัญญาณที่เวลาต่างๆ โดยใช้สัญญาณชิงค์ช่วยในระบบ TDM

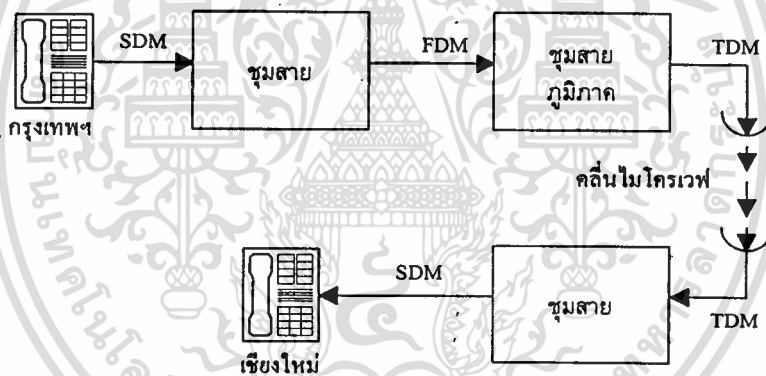
การมัลติเพล็กซ์ทั้งสามระบบข้างต้นเป็นระบบพื้นฐานที่มีอยู่ แต่ในการประยุกต์ใช้งานกับระบบสื่อสาร โดยทั่วไปมักใช้ทั้งสามระบบร่วมกัน เช่น ระบบโทรศัพท์ทางไกล เป็นต้น ดังรูปที่

2.13 ที่ชุมสายท้องถิ่นจะใช้ การมัลติเพล็กซ์แบบแบ่งเส้นทาง ระหว่างชุมสายท้องถิ่นและชุมสายที่ไม่ใช่



รูปที่ 2.12 การส่งสัญญาณในระบบ TDM

ค่อผ่านทางไกล เป็นการมัลติเพล็กซ์แบบแบ่งความถี่ โดยใช้เดินสายเคเบิล และระหว่างชุมสายค่อผ่านทางไกลเป็นการมัลติเพล็กซ์แบบแบ่งเวลา ผ่านระบบการเชื่อมโยงของไมโครเวฟ



รูปที่ 2.13 ตัวอย่างการประยุกต์ใช้งานของโทรศัพท์ทางไกล โดยมีการมัลติเพล็กซ์แบบต่างๆ

ปัญหาหลักของการสื่อสารแอนะล็อกคือสัญญาณรบกวน (Noise) ซึ่งจะส่งผลมาก โดยเฉพาะระบบการมัลติเพล็กซ์แบบแบ่งเวลา การมัลติเพล็กซ์แบบแบ่งความถี่และการมัลติเพล็กซ์แบบแบ่งเส้นทางมากขึ้น ตามลำดับ การที่จะสร้างให้ระบบการทำงานมีความเที่ยงตรงมากเป็นไปได้ยาก ดังนั้น ปกติจึงนิยมใช้กับสัญญาณเสียง หรือภาพที่ต้องการคุณภาพไม่สูงนัก ไม่เหมาะกับการส่งข้อมูลที่ต้องการความเที่ยงตรงมากๆ แม้จะมีการลดสัญญาณรบกวนก็ตาม เช่น การส่งสัญญาณตรวจสอบส่วนซ้ำซ้อน (Redundant Checking) เป็นต้น แต่วิธีเหล่านี้ ก็ยังไม่ดีพอที่จะทำให้การสื่อสารแอนะล็อกมีความเที่ยงตรงมากขึ้น ได้มากนัก

เอกสาคือเป็นเอกสารที่ส่งมาให้อ่านหรือเก็บไว้เพื่อบันทึกข้อมูลหรือเป็นหลักฐาน ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 2.4 ระบบสื่อสารดิจิทัล

ความต้องการรับส่งข้อมูลข่าวสารในปัจจุบันที่มีมากทั้งปริมาณ และชนิดของข้อมูล ทำให้เกิดการพัฒนาและการขยายตัวของระบบสื่อสารข้อมูลเป็นอย่างมาก ทั้งโครงข่ายโทรศัพท์ โครงข่ายข้อมูลแบบดิจิทัล รวมถึงโครงข่ายโทรสารและโทรทัศน์ เป็นต้น เป็นโครงข่ายที่พัฒนาขึ้นมาเพื่อตอบสนองความต้องการรับส่งข้อมูลต่างๆทั้งสิ้น ในยุคที่เริ่มมีการรับส่งข้อมูลระบบต่างๆ ทำงานแบบแอนะล็อกทั้งหมด ต่อมาเมื่อเทคโนโลยีทางด้านดิจิทัลได้ก้าวหน้าขึ้นมา จึงได้เริ่มมีการนำเทคโนโลยีดิจิทัลเข้าไปทดแทนแบบแอนะล็อกเดิม ทั้งการนำไปทดแทนทั้งหมด เช่น สร้างโครงข่ายชนิดใหม่ หรือนำไปทดแทนบางส่วน เช่น โมเด็ม ทั้งนี้เนื่องจากข้อดีของสัญญาณแบบดิจิทัลนั่นเอง เช่น

1. ให้คุณภาพการรับส่งข้อมูลที่เท่ากันหรือดีกว่าแบบแอนะล็อก
2. ยืดอายุการบำรุงรักษา
3. เพิ่มเติมปรับปรุง หรือเปลี่ยนแปลงความสามารถหรือบริการของระบบได้ง่าย
4. มีความเร็วในการรับส่งข้อมูลสูง
5. ทนต่อสัญญาณรบกวนได้ดี
6. สามารถเปลี่ยนแปลงรูปแบบของสัญญาณไปเป็นรูปแบบตามมาตรฐานต่างๆ ได้ง่าย
7. เนื่องจากสัญญาณดิจิทัลสามารถแทนได้ด้วยตัวเลข จึงนำไปทำการคำนวณที่ซับซ้อนได้ จากจุดนี้เองทำให้เกิดวิธีการบีบอัดข้อมูลและการเข้ารหัสแบบต่างๆ มากมาย

สัญญาณดิจิทัลมักจะหมายถึงสัญญาณในระบบเลขฐานสองได้แก่ “0” และ “1” เท่านั้น “0” และ “1” นี้แทนค่าความหมายต่างๆ ของข้อมูลได้ แต่ในความเป็นจริง สัญญาณดิจิทัลเป็นสัญญาณที่มีการแบ่งค่าต่างๆ ออกเป็น 2, 4, 8 หรือ 16 ระดับก็ได้ เหตุผลที่มีระดับมากกว่าสอง (“0” และ “1”) การเพิ่มระดับเป็นการช่วยในการใช้ประโยชน์จากช่องสัญญาณ และทำให้สมรรถนะที่มีความผิดพลาดต่ำอันเป็นเป้าหมายสูงสุดของระบบ

ระบบสื่อสารดิจิทัลสามารถใช้รูปแบบของสัญญาณได้หลายชนิด และการมอดูเลตที่ตรงกันได้อย่างดี เช่น แบ่งระดับแรงดันในการมอดูเลตแอมพลิจูด แบ่งระดับความถี่ในการมอดูเลตความถี่ และแบ่งระดับเฟสในการมอดูเลตเฟส เป็นต้น เพื่อที่จะนำมาใช้งาน อาจจะใช้ในลักษณะแบบเดียวกันก็ได้ ดังตัวอย่างเช่น แอมพลิจูด 2 โวลต์, 4 โวลต์, 6 โวลต์ และ 8 โวลต์ แบ่งออกได้เป็น 4 ระดับ, ความถี่ 1 กิโลเฮิรตซ์, 4 กิโลเฮิรตซ์และ 6 กิโลเฮิรตซ์ แบ่งออกได้เป็น 3 ระดับ ของแอมพลิจูดและความถี่ 1 โวลต์ และ 2 โวลต์ ที่ 1 กิโลเฮิรตซ์ และ 2 กิโลเฮิรตซ์ ตามลำดับ แบ่งออกได้เป็น 2 ระดับ, แอมพลิจูด และเฟส 1 โวลต์และ 2 โวลต์ที่ +90 องศา และ -90 องศา ตามลำดับ แบ่งออกได้เป็น 2 ระดับแอมพลิจูด และ 2 ระดับเฟส เป็นต้น

ข้อมูลในระบบสื่อสารดิจิทัลใช้คำว่า บิต ซึ่งมีรากศัพท์มาจากคำว่า Binary Digit และใช้ในระบบเลขฐานสอง มีสัญลักษณ์เป็น “0” และ “1” กลุ่มของข้อมูล 8 บิต เรียกว่า ไบต์ ในระบบของการสื่อสารดิจิทัลที่มีระดับมากกว่า 2 หรือไม่ได้เป็นเลขฐานสองยังใช้คำว่าบิต เนื่องจากข้อมูลสามารถเปลี่ยนรูปจากแบบหนึ่งไปสู่อีกแบบหนึ่งได้ แต่เนื่องจากระบบคอมพิวเตอร์ทั้งหมดใช้ระบบเลขฐานสองในการวัด และการวิเคราะห์ปัญหา ไม่ว่าระบบสื่อสารดิจิทัลจะมีระดับเท่าใดก็ตามก็ยังต้องใช้คำว่า “บิต”

ข้อได้เปรียบของสัญญาณดิจิทัลที่สำคัญคือ เนื่องจากสามารถนำสัญญาณไปประมวลได้หลายรูปแบบ โดยที่มีคุณภาพเท่ากันหรือดีกว่าสัญญาณแบบแอนะล็อก สัญญาณแบบดิจิทัลจึงใช้งานได้หลากหลายกว่า อย่างไรก็ดี สัญญาณดิจิทัลก็คือสัญญาณแอนะล็อกที่มีการเปลี่ยนแปลงแบบทันทีทันใด ดังนั้นระบบที่ใช้ กับแบบแอนะล็อกจะสามารถใช้กับสัญญาณแบบดิจิทัลได้ด้วย ไม่ว่าจะเป็นการมอดูเลตแอมพลิจูด การมอดูเลตความถี่และการมอดูเลตเฟส หรือการมัลติเพล็กซ์แบบแบ่งเส้นทาง การมัลติเพล็กซ์แบบแบ่งความถี่ และการมัลติเพล็กซ์แบบแบ่งเวลา ซึ่งสัญญาณแอนะล็อกที่มีรูปแบบเฉพาะการใช้เทคนิคต่างๆ ของระบบแอนะล็อกจึงเป็นการสิ้นเปลือง และได้มีเทคนิคของระบบดิจิทัลโดยเฉพาะขึ้นมา

#### 2.4.1 การเข้ารหัส

แหล่งกำเนิดข้อมูล (Source) เช่น ภาพ เสียง หรือสัญญาณพูดที่ใช้ในการสื่อสารมักจะเป็นแหล่งกำเนิดสัญญาณซึ่งมีคุณสมบัติเฉพาะ การที่จะส่งสัญญาณเหล่านั้นไปได้มักจะใช้วิธีการเข้ารหัส (Encoding) เพื่อเปลี่ยนรูปแบบของสัญญาณจากรูปแบบหนึ่งไปยังอีกรูปแบบหนึ่ง ตัวอย่างของการเข้ารหัสที่ใช้กันมากได้แก่

1) การเข้ารหัสแหล่งข้อมูล (Source Encoding) ทำหน้าที่ในการลดขนาดของข้อมูลดิจิทัลที่จะส่งให้มีการสูญเสียของข่าวสารน้อย ซึ่งรวมไปถึงการซ้ำซ้อนของข้อมูล โดยปกติแล้วสัญญาณดิจิทัลที่ได้จากการเข้ารหัสแหล่งข้อมูลมักจะน้อยกว่าสัญญาณดิจิทัลขาเข้าเริ่มแรก

2) การเข้ารหัสช่องสัญญาณ (Channel Encoding) ปกติแล้วหลักการนี้มักจะใช้เพื่อให้สัญญาณดิจิทัลขาออกมีความอ่อนไหวหรือแปรเปลี่ยนตามคุณภาพของช่องสัญญาณในการส่งน้อยลง โดยทั่วไปจะใช้รหัสตรวจจับและแก้ไขความผิดพลาดเข้ามาช่วย และในการใช้งานอาจรวมการเข้ารหัสช่องสัญญาณและการเข้ารหัสแหล่งข้อมูลเป็นกระบวนการเดียวกันหรือแยกกันก็ได้ การเข้ารหัสช่องสัญญาณอาจรวมไปถึงการเปลี่ยนข้อมูลให้เป็นสัญญาณไฟฟ้าที่เหมาะสมกับสายนำสัญญาณ โดยการกระจายกำลังของสัญญาณออกไป

3) การเข้ารหัสสายหรือไลน์โคดดิ้ง (Line Coding) เป็นการเข้ารหัสสัญญาณดิจิทัลที่เป็นตัวเลขให้เป็นสัญญาณทางไฟฟ้า ในบางกรณีสัญญาณทางไฟฟ้าเหล่านี้ สามารถที่จะส่งออกไปที่

สายนำสัญญาณได้เลย ไม่จำเป็นต้องนำไปมอดูเลตกับคลื่นพาห์ด้วยวิธีการต่างๆ เพื่อส่งไปในช่องสัญญาณก็ได้ วิธีนี้เป็นการปรับสัญญาณให้เข้ากับสมบัติทางกายภาพของสายนำสัญญาณ เช่น แบนด์วิดท์ ทิศทางของกระแสไฟฟ้า การส่งผ่านและการจำรูปแบบ ความถี่ของสัญญาณนาฬิกา (Clock Frequency) เป็นต้น การจะเลือกใช้การเข้ารหัสสายแบบใดนั้นขึ้นอยู่กับความเหมาะสมในแต่ละกรณี ยกตัวอย่างเช่น ต้องการส่งสัญญาณในสายวงจรพิมพ์เป็นการส่งสัญญาณในระยะทางสั้นๆ ใช้อุปกรณ์ในการเข้ารหัสน้อย จะพบเป็นแบบ Unipolar NRZ ซึ่งได้แก่การเปลี่ยนระดับแรงดันของตัวคิจิตอลไอซี (Digital IC) ต่างๆ นั้นเอง หรือสัญญาณในคู่สายของ N-ISDN จะเป็นแบบ Pseudo-Ternary ทำให้สามารถส่งสัญญาณคิจิตอลได้ไกลขึ้น

4) การเข้ารหัสลับและการถอดรหัสลับ (Encryption and Decryption) เป็นการป้องกันข่าวสารที่ส่งออกไปให้ผู้ที่ไม่ได้รับอนุญาตรับรู้ข้อมูลได้ โดยปกติมักเป็นการสื่อสารในรูปแบบเครือข่ายที่เป็นสาธารณะ เช่น เคเบิลทีวี เป็นต้น รูปแบบการเข้ารหัสมักจะเก็บไว้เป็นความลับ การเข้ารหัสแบบนี้จะนำไปใช้กับงานที่ต้องการความเชื่อถือของข้อมูลสูงได้

#### 2.4.2 การจัดรูปแบบและการเข้ารหัสข้อมูล

การจัดรูปแบบ (Formatting) เป็นกระบวนการที่ปรับเปลี่ยนข้อมูลให้เป็นสัญลักษณ์ทางคิจิตอล และการเข้ารหัสข้อมูล (Source Encoding) เป็นกระบวนการบีบอัดข้อมูลตามรูปแบบของกระบวนการจัดรูปแบบ ทั้งนี้เนื่องจากแหล่งข้อมูลส่วนใหญ่สัญญาณแอนะล็อกหรือข้อความยกเว้นสัญญาณการติดต่อระหว่างเครื่องคอมพิวเตอร์ที่เป็นสัญลักษณ์ทางคิจิตอลอยู่แล้ว การจัดรูปแบบที่รู้จักกันดีได้แก่ รหัสแอสกี (American Standard Code for Information Interchange หรือ ASCII) ซึ่งใช้กับข้อมูลที่เป็นตัวอักษรและตัวเลข เป็นต้น หากเป็นสัญญาณแอนะล็อกต้องถูกเปลี่ยนเป็นสัญญาณคิจิตอลโดยอาศัยทฤษฎีของไนควิสต์ กล่าวคือ สัญญาณทุกสัญญาณในทางปฏิบัติจะประกอบด้วยช่วงความถี่ต่างๆ ที่จำกัด ช่วงความถี่ที่ประกอบขึ้นเป็นสัญญาณนั้นเรียกว่า แบนด์วิดท์ของสัญญาณ ทฤษฎีของไนควิสต์กล่าวไว้ว่า หากสุ่มสัญญาณใดๆ โดยความถี่ที่นำมาสุ่มนั้นมากกว่า 2 เท่า ของความถี่สูงสุดในสัญญาณนั้น สัญญาณที่สุ่มมานั้นจะประกอบด้วยข้อมูลของสัญญาณเดิมทุกประการ เช่น สัญญาณเสียงมีแบนด์วิดท์ประมาณ 4 กิโลเฮิรตซ์ ดังนั้น ความถี่ที่ใช้ในการสุ่มสัญญาณเสียงต้องไม่ต่ำกว่า 8 กิโลเฮิรตซ์ เพื่อให้ข้อมูลของสัญญาณเสียงครบถ้วน เมื่อเราเปลี่ยนสัญญาณเสียงที่ถูกสุ่มแล้วกลับมาเป็นสัญญาณเสียงที่เป็นสัญญาณแอนะล็อก โดยการผ่านวงจรกรองความถี่ต่ำผ่าน จะได้สัญญาณเสียงที่เหมือนเดิม ทฤษฎีของไนควิสต์ นี้สามารถพิสูจน์ได้ด้วยคณิตศาสตร์เกี่ยวกับผลแปลงฟูริเยร์ (Fourier Transform) หลังจากที่ได้สัญญาณคิจิตอลแล้ว ไม่ว่าจะ ได้จากการสุ่มสัญญาณแอนะล็อกหรือจากการจัดรูปแบบให้เป็นไปตามมาตรฐานต่างๆ มีวิธีนำสัญญาณเหล่านั้นมาเข้ารหัสข้อมูล ซึ่งมีหลายวิธี เช่น

1) การมอดูเลตแอมพลิจูดของพัลส์ (Pulse Amplitude Modulation ; PAM) การมอดูเลตแบบนี้ แอมพลิจูดของพัลส์ที่นำไปส่งจะเปลี่ยนไปตามสัญญาณที่เข้ามา

2) การมอดูเลตแบบช่วงกว้างพัลส์ (Pulse Duration Modulation ; PDM) ความกว้างของพัลส์ที่นำไปส่งจะเปลี่ยนไปตามสัญญาณที่เข้ามา

3) การมอดูเลตตำแหน่งพัลส์ (Pulse Position Modulation ; PPM) เนื่องจากการมอดูเลตแบบช่วงกว้างพัลส์ (PDM) นั้นมีข้อเสีย คือพัลส์ที่มีความกว้างมากจะใช้พลังงานมากขึ้น โดยไม่ได้มีจำนวนข่าวสารที่ส่งไปมากขึ้นตาม วิธีนี้ใช้วิธีเปลี่ยนตำแหน่งของพัลส์ที่นำไปส่งตามสัญญาณที่เข้ามา

4) การมอดูเลตรหัสพัลส์ (Pulse Code Modulation ; PCM) วิธีนี้เป็นวิธีที่ใช้กันแพร่หลายที่สุด ด้วยการเปลี่ยนค่าของสัญญาณที่สุ่มได้เป็นกลุ่มของเลขฐานสองแล้วจึงส่งไปเป็นสัญญาณพัลส์ที่มีแอมพลิจูดเท่ากัน เมื่อสุ่มข้อมูลและได้สัญญาณแบบมอดูเลตทางแอมพลิจูดของพัลส์มาแล้ว ค่าของกลุ่มสัญญาณที่เข้ากับสัญญาณมอดูเลตทางแอมพลิจูดของพัลส์นั้นจะถูกส่งออกไป เนื่องจากแอมพลิจูดของการมอดูเลตทางแอมพลิจูดของพัลส์มีการเปลี่ยนแบบแอนะล็อก แต่การแทนค่าด้วยกลุ่มของเลขฐานสองทำให้เกิดความผิดพลาดขึ้น ซึ่งเรียกว่าความผิดพลาดจากการควอนไทซ์ (Quantization Error) เนื่องจากกลุ่มของเลขฐานสองจำนวน  $n$  หลักนี้จะแทนระดับแรงดันได้สำหรับเครื่องโคเดกเชิงเส้น (Linear Codec) ที่มีเลขฐานสองจำนวน  $n$  หลักต่อตัวอย่าง (Sample) อัตราส่วนระหว่างกำลังของสัญญาณและกำลังของความเพี้ยนจากการควอนไทซ์ (Quantizing Distortion ; S/D) เป็น

$$S/D = 6n + 1.8 \text{ dB}$$

จากสมการบอกว่าการเพิ่มหลักของจำนวนเลขฐานสองจำนวน 1 หลักจะทำให้ S/D สูงขึ้น 6 เดซิเบล หรือการเพิ่มหลักของเลขฐานสองจะทำให้สามารถส่งข้อมูลได้รายละเอียดมากขึ้น

สำหรับสัญญาณเสียงนั้น การส่งแบบมอดูเลตรหัสพัลส์ ซึ่งจะมีเลขฐานสองจำนวน 8 หลักก็เพียงพอต่อการใช้งาน หากพิจารณาองค์ประกอบของสัญญาณเสียงจะพบว่า ช่วงกลางของสัญญาณหรือช่วงใกล้แรงดัน 0 โวลต์ จะมีการเปลี่ยนแปลงมากช่วงสูงสุดและต่ำสุดของสัญญาณจะมีการเปลี่ยนแปลงน้อยหรือไม่เกิดขึ้นเลย ดังนั้นการแทนค่าเลขฐานสองด้วยระดับแรงดันต่างๆ ซึ่งเพิ่มเป็นแบบเชิงเส้นจึงไม่ค่อยมีประสิทธิภาพ ทำให้มีการแบ่งช่วงแรงดันสำหรับการแทนค่าของเลขฐานสองใหม่ โดยแบ่งให้ช่วงใกล้แรงดัน 0 โวลต์ให้ดีกว่าปกติ ช่วงค่าสูงสุดและค่าต่ำสุดกว้างกว่า

เอกส  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปกติเรียกว่า การคอมแพนด์ (Companding) ซึ่งความกว้างมาตรฐานที่ใช้ในการแบ่งนั้นปัจจุบันมีใช้ อยู่ 2 ระบบ ได้แก่ กฎเอ (A-Law) และ กฎมิวส์ ( $\mu$ -Law)

5) การมอดูเลตแบบปรับตัว PCM (Adaptive PCM ; APCM) ถึงแม้การมอดูเลตรหัสพัลส์ จะมีการใช้การคอมแพนด์มาช่วยเพิ่มประสิทธิภาพ แต่การที่มีขนาดของชั้นแบบเชิงเส้น ทำให้ เหมาะกับสัญญาณที่มีรายละเอียดอยู่ที่ช่วงกลางของสัญญาณเท่านั้น ไม่สามารถใช้กับสัญญาณได้ ทุกชนิด การมอดูเลตแบบปรับตัว PCM (APCM) เป็นการมอดูเลตแบบรหัสพัลส์ที่สามารถปรับ ขนาดของชั้นได้โดยอัตโนมัติ ซึ่งทำให้สามารถใช้กับรูปแบบของสัญญาณที่หลากหลายมากขึ้น

6) การมอดูเลตแบบผลต่าง PCM (Differential PCM ; DPCM) สำหรับสัญญาณเสียง สัญญาณจะไม่เปลี่ยนแปลงโดยทันทีทันใด ดังนั้นแทนที่จะทำการเข้ารหัสที่ค่าของแอมพลิจูดของ สัญญาณ วิธีนี้จะใช้วิธีการเข้ารหัสที่ค่าความแตกต่างของแอมพลิจูดของพัลส์ ที่สุ่มได้ในปัจจุบันกับ ค่าแอมพลิจูดของพัลส์ก่อนหน้า

7) การมอดูเลตแบบปรับตัวผลต่าง PCM (Adaptive Differential PCM ; ADPCM) วิธีนี้ เป็นวิธีที่ผสมระหว่างการมอดูเลตแบบผลต่าง PCM (DPCM) กับการมอดูเลตแบบปรับตัว PCM (APCM) ซึ่งทำให้ขนาดของชั้นของการมอดูเลตแบบผลต่าง PCM (DPCM) จะเปลี่ยนแปลงได้ เหมาะสมกับสัญญาณ

8) การมอดูเลตแบบเดลต้า (Delta Modulation ; DM) เป็นกรณีเฉพาะของการมอดูเลต แบบผล PCM ที่มีการเปลี่ยนแปลงของบิตไม่เกิน 1 บิต จากเหตุนี้เองทำให้การมอดูเลตแบบเดลต้า อาจเกิดปัญหาการเกินของความชัน (Slope Overload) ได้ กล่าวคือค่าที่ได้ไม่สามารถเปลี่ยนแปลง ตามสัญญาณได้ทัน เนื่องจากสัญญาณมีการเปลี่ยนแปลงเร็วเกินไป

9) การมอดูเลตแบบเชิงเส้น (Linear Predictive Code ; LPC) วิธีนี้ตั้งอยู่บนพื้นฐานที่ว่า สัญญาณทุกสัญญาณสามารถจำลองได้เป็นระบบแบบเชิงเส้น ดังนั้น ส่วนที่ส่งผลต่อสัญญาณจะถูก เข้ารหัสแทนที่จะต้องเข้ารหัสสัญญาณทั้งหมด เช่น สัญญาณเสียง สามารถแบ่งออกได้เป็นช่วงที่มี เสียง ช่วงที่ไม่มีเสียง อัตราการเปลี่ยนแปลงความถี่หรือความแหลมของเสียง เป็นต้น การเข้ารหัส แบบนี้จะทำให้จำนวนบิตต่อตัวอย่างไม่เท่ากัน แต่จะทำให้ค่าเฉลี่ยของจำนวนบิตต่อตัวอย่างต่ำกว่า ทุกวิธี

### 2.4.3 รูปแบบการมอดูเลตสัญญาณดิจิทัล

การมอดูเลตสัญญาณดิจิทัลเข้ากับคลื่นพาห์ที่เป็นสัญญาณไซน์ นั้นมีอยู่หลายรูปแบบ ทั้งนี้เพื่อต้องการให้สัญญาณดิจิทัล เหล่านั้นสามารถส่งผ่านตัวกลางที่ออกแบบมาสำหรับระบบของ สัญญาณแอนะล็อกได้ เช่น โครงข่ายโทรศัพท์พื้นฐาน ไมโครเวฟ เป็นต้น การมอดูเลตที่ใช้กันทั่ว ไม่ไปไม่ได้แก่ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1) การมอดูเลตเชิงเลขทางแอมพลิจูด (Amplitude Shift Keying ; ASK) เป็นลักษณะการมอดูเลตที่อาศัยการเปลี่ยนแปลงแอมพลิจูดของคลื่นพาห์ไปตามสัญญาณที่เข้ามา มีดัชนีมอดูเลชัน (Modulation Index) เป็น 1

2) การมอดูเลตเชิงเลขทางความถี่ (Frequency Shift Keying ; FSK) เป็นการมอดูเลตที่อาศัยการเปลี่ยนแปลงความถี่ของคลื่นพาห์ไปตามสัญญาณที่เข้ามา การสร้างสัญญาณ FSK สามารถสร้างได้ 2 แบบคือ

2.1) ในสวิตช์แบบดิจิทัล สวิตช์ให้คลื่นพาห์ที่กำเนิดจากแหล่งกำเนิดหลายๆ แหล่งออกไปที่ด้านขาออกตามค่าของสัญญาณที่เข้ามา

2.2) ในแหล่งกำเนิดสัญญาณตัวเดียว แต่แหล่งกำเนิดนั้น สามารถเปลี่ยนแปลงความถี่ได้ สัญญาณที่ได้จากวิธีนี้จะมีความต่อเนื่องของสัญญาณ จึงเรียกว่า FSK เฟสต่อเนื่อง (Continuous Phase FSK หรือ CPFSK)

3) การมอดูเลตเชิงเลขทางเฟส (Phase Shift Keying ; PSK) เป็นการมอดูเลตที่อาศัยการเปลี่ยนแปลงของเฟสของคลื่นพาห์ไปตามสัญญาณที่เข้ามา หากสัญญาณที่เข้ามี  $M$  ระดับ จะสามารถแบ่งเฟสของคลื่นพาห์ได้เป็น  $M$  ส่วนเรียกว่า M-PSK

4) การมอดูเลตแบบควอดเรเจอร์แคเรียร์แอมพลิจูด (Quadrature Carrier Amplitude Modulation ; QAM) เป็นการมอดูเลตแบบแอมพลิจูด ที่ใช้คลื่นพาห์ที่มีเฟสต่างกัน 90 องศา หากพิจารณาในเชิงคณิตศาสตร์จะเปรียบเสมือนค่าของตำแหน่งใดๆ จะเกิดจากขนาดของค่าในแกน  $X$  และแกน  $Y$  ซึ่งทั้งสองแกนทำมุมกัน 90 องศา การมอดูเลตควอดเรเจอร์แคเรียร์แอมพลิจูด (QAM) นี้ก็เช่นกัน ค่าของสัญญาณดิจิทัลจะถูกแทนด้วยคลื่นพาห์ที่มีเฟสเท่ากับ 0 องศา (Inphase) และ 90 องศา (Quadrature Phase) ซึ่งหากสามารถแทนสัญญาณดิจิทัลได้  $M$  สัญลักษณ์นี้เราเรียกว่า M-QAM และแต่ละสัญลักษณ์จะมีจำนวน  $n$  บิต เช่น แต่ละสัญลักษณ์ของ 16-QAM ประกอบด้วยสัญญาณดิจิทัล 4 บิต เป็นต้น

จากความก้าวหน้าทางเทคโนโลยีและข้อได้เปรียบต่างๆ ของสัญญาณดิจิทัลที่ทำให้การสื่อสารในยุคปัจจุบันเริ่มก้าวสู่ยุคของดิจิทัล เริ่มมีการนำสัญญาณดิจิทัลมาใช้งานแทน สัญญาณแอนะล็อกและมีการนำมาใช้งานมากขึ้นเรื่อยๆ อย่างไรก็ตามสัญญาณดิจิทัลยังไม่สามารถทดแทนสัญญาณแอนะล็อกได้ทั้งหมด เนื่องจากในการเปลี่ยนสัญญาณแอนะล็อกให้เป็นสัญญาณทางดิจิทัลต้องเกิดความผิดพลาดทุกครั้ง ซึ่งไม่สามารถกำจัดได้ และสัญญาณต่างๆ ตามธรรมชาติต่างเป็นสัญญาณแอนะล็อกทั้งสิ้น ดังนั้นในปัจจุบันยังคงเห็นการใช้งานสัญญาณดิจิทัลแอนะล็อกผสมผสานกันไป ทั้งนี้เพื่อให้การสื่อสารเป็นไปอย่างมีประสิทธิภาพมากที่สุดนั่นเอง

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้สำหรับใช้ในการศึกษาเท่านั้น ไม่สามารถนำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 2.5 การสื่อสารข้อมูล

ในการรับส่งข้อมูลข่าวสารจากอีกแหล่งหนึ่งด้วยความถูกต้องสมบูรณ์ ซึ่งในอดีตนั้นเริ่มจากการติดต่อโดยใช้วิธีทางธรรมชาติ เช่น ควันไฟ แสง เสียง นก เป็นต้น ต่อมาแซมมอล มอร์สได้คิดค้นโทรเลขขึ้นในปี 1854 จากนั้นอเล็กซานเดอร์เกรแฮมเบลได้คิดค้นโทรศัพท์ในปี 1876 และในปี 1950 ก็ได้มีการพัฒนาระบบดิจิทัลคอมพิวเตอร์ ซึ่งช่วยเพิ่มประสิทธิภาพและความเชื่อถือเป็นอย่างมาก ทำให้การติดต่อสื่อสารมีประสิทธิภาพสูงขึ้น คำว่า “Data Communication” หมายถึง การสื่อสารข้อมูล ซึ่งหมายถึงการแลกเปลี่ยนข้อมูลกันระหว่างอุปกรณ์ 2 ชนิด โดยมีการส่งสัญญาณแบบอิเล็กทรอนิกส์ ข้อมูลในที่นี้มักได้แก่ข้อมูลดิจิทัลที่มีค่า “0” และ “1” ทำอย่างไรที่จะส่ง “0” และ “1” นี้จากอุปกรณ์ชนิดหนึ่งไปยังอุปกรณ์อีกชนิดหนึ่งได้ อุปกรณ์สื่อสารข้อมูลที่ใช้กันมากได้แก่ ระบบคอมพิวเตอร์ ซึ่งมักจะประกอบด้วยคอมพิวเตอร์แม่เทอร์มินัลเป็นอุปกรณ์หลัก การที่จะทำให้การสื่อสารประสบความสำเร็จนั้น ผู้ส่งสารและผู้รับสารต้องทำความเข้าใจร่วมกัน เช่น ทิศทางในการส่งข้อมูล สัญลักษณ์ที่ใช้ อัตราความเร็วที่ใช้ในการถ่ายเทข้อมูล เป็นต้น โดยทั่วไปการสื่อสารข้อมูลหมายถึง วิธีการที่มีประสิทธิภาพที่รวมเอาการประมวลผล และการส่งข้อมูลเข้าด้วยกันโดยใช้การสื่อสารเชื่อมโยงระหว่างหน่วยอินพุต/เอาต์พุต (Input/Output) ที่อยู่ห่างไกลกัน ระบบที่ถูกสร้างขึ้นมาเพื่อการสื่อสารข้อมูลที่เรียกว่า ระบบสื่อสารข้อมูล (Data Communication System)

### 2.5.1 อักขระในระบบสื่อสารข้อมูล

ในระบบสื่อสารข้อมูลจะส่งข้อมูลเป็น “0” หรือ “1” ซึ่งแต่ละหน่วยจะเรียกว่า บิต (Bit) กลุ่มของบิต ซึ่งปกติมักเป็น 8 หน่วย เรียกว่า ไบต์ ในการใช้งานแต่ละครั้ง ผู้ใช้ที่แต่ละเทอร์มินัลจะเกาะแป้นเพื่อส่งบิตจำนวนหนึ่งไปยังคอมพิวเตอร์แม่ เมื่อคอมพิวเตอร์แม่ได้รับข้อมูลก็จะทำการพิจารณาว่าอักขระ (Character) เป็นอย่างไร เช่น ตัวเลข ตัวอักษรและเครื่องหมายต่างๆ เป็นต้น การสื่อความหมายจะทำอยู่ในรูปรหัสอักขระ (Character Code) ซึ่งมีด้วยกันหลายรูปแบบ เช่น รหัสมอร์ส (Morse Code) รหัสโบคอต (Baudot Code) รหัสแอสกี (ASCII Code) และรหัสแอบซีดิก (EBCDIC Code) เป็นต้น

รหัสมอร์สเป็นอักขระที่ใช้ในการส่งโทรเลข โดยใช้จุด (Dot) และขีด (Dash) เป็นรหัสในการก่อให้เกิดอักขระ ซึ่งประเภทของงานที่นำไปใช้มีมากมาย ทางด้านการทหาร โดยจะเป็นการส่งข้อมูลที่มีการความถี่สูงและคนปกติไม่สามารถรับรู้ถึงอักษรแล้วเสียงที่ส่งออกไป นอกจากคนเรานั้นได้ศึกษาหรือได้เรียนถึงอักขระที่ใช้ในการสื่อสารแบบรหัสมอร์ส ดังตารางที่ 2.3 จะเป็นการบอกถึงลักษณะของตัวอักษรที่ใช้และการแปลอักษรเป็นรหัสมอร์สเพื่อนำไปใช้ในการส่งข้อมูล เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

แม้ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.3 รหัสมอร์ส

| อักษร | รหัส   |
|-------|--------|
| A     | .-.    |
| B     | ....   |
| C     | ..--.  |
| D     | ...-   |
| E     | .      |
| F     | ..-.   |
| G     | ...-   |
| H     | ....   |
| I     | ..     |
| J     | .-.-.  |
| K     | -.-    |
| L     | ..-.   |
| M     | --     |
| N     | ..-    |
| O     | ---    |
| P     | ..-.   |
| Q     | ..-.   |
| R     | ..-    |
| S     | ...-   |
| T     | -      |
| U     | ...-   |
| V     | ...-   |
| W     | ..-    |
| X     | ..-    |
| Y     | ..-    |
| Z     | ..-    |
| 1     | .----  |
| 2     | ..---  |
| 3     | ...--  |
| 4     | ....-  |
| 5     | .....  |
| 6     | -.---- |
| 7     | --.... |
| 8     | ---... |
| 9     | ----.  |
| 0     | -----  |
| .     | ...-.- |
| ?     | ...-.- |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น มิอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเนื้อหาและข้อมูลใดๆ ที่ปรากฏในเอกสารทุกครั้งที่มีการนำไปใช้

รหัสโบคอตจะใช้รหัสเป็น “0” และ “1” โดยแต่ละอักขระจะประกอบด้วย 5 บิต ดังตารางที่ 2.4 ในส่วนของลำดับบิต ซึ่งจะเห็นว่าส่วนใหญ่จะใช้อุปกรณ์เทเล็กซ์ ในการส่งข้อมูลมีความผิดพลาดบิตเกิดขึ้นได้ง่าย และทางอุปกรณ์ด้านรับไม่สามารถบอกได้ว่ามีความผิดพลาดของบิตเกิดขึ้น อันนี้จึงเป็นข้อเสียประการหนึ่งที่ทำให้การส่งข้อมูลมีการผิดพลาดเกิดขึ้น จึงมีการปรับปรุงและตรวจจับความผิดพลาดนั้น ในการตรวจจับความผิดพลาด อักขระที่ปรับปรุงใหม่นี้เรียกว่า รหัสโบคอตนานาชาติ (International Baudot Code) เมื่อทำการแก้ไขปัญหาทางด้านรับได้ การส่งข้อมูลก็มีความผิดพลาดของการส่งน้อยลง แต่ยังมีผิดพลาดของการส่งข้อมูลน้อยกว่าเดิม ทำให้การสื่อสารในการส่งข้อมูลมีประสิทธิภาพมากขึ้น

ตารางที่ 2.4 รหัสโบคอต

| ระดับล่าง | ระดับบน | ลำดับบิต |   |   |   |   |
|-----------|---------|----------|---|---|---|---|
|           |         | 5        | 4 | 3 | 2 | 1 |
| A         | -       | 0        | 0 | 0 | 1 | 1 |
| B         | ?       | 1        | 1 | 0 | 0 | 1 |
| C         | :       | 0        | 1 | 1 | 1 | 0 |
| D         | \$      | 0        | 1 | 0 | 0 | 1 |
| E         | 3       | 0        | 0 | 0 | 0 | 1 |
| F         | !       | 0        | 1 | 1 | 0 | 1 |
| G         | &       | 1        | 1 | 0 | 1 | 0 |
| H         | #       | 1        | 0 | 1 | 0 | 0 |
| I         | 8       | 0        | 0 | 1 | 1 | 0 |
| J         | .       | 0        | 1 | 0 | 1 | 1 |
| K         | (       | 0        | 1 | 1 | 1 | 1 |
| L         | )       | 1        | 0 | 0 | 1 | 0 |
| M         | .       | 1        | 1 | 1 | 0 | 0 |
| N         | ,       | 0        | 1 | 1 | 0 | 0 |
| O         | 9       | 1        | 1 | 0 | 0 | 0 |
| P         | 0       | 1        | 0 | 1 | 1 | 0 |
| Q         | 1       | 1        | 0 | 1 | 1 | 1 |
| R         | 4       | 0        | 1 | 0 | 1 | 0 |
| S         | BELL    | 0        | 0 | 1 | 0 | 1 |
| T         | 5       | 1        | 0 | 0 | 0 | 0 |

ตารางที่ 2.4 (ต่อ) รหัสไบนารี

| ระดับล่าง           | ระดับบน | ลำดับบิต |   |   |   |   |
|---------------------|---------|----------|---|---|---|---|
|                     |         | 5        | 4 | 3 | 2 | 1 |
| U                   | 7       | 0        | 0 | 1 | 1 | 1 |
| V                   | :       | 1        | 1 | 1 | 1 | 0 |
| W                   | 2       | 1        | 0 | 0 | 1 | 1 |
| X                   | /       | 1        | 1 | 1 | 0 | 1 |
| Y                   | 6       | 1        | 0 | 1 | 0 | 1 |
| Z                   | "       | 1        | 0 | 0 | 0 | 1 |
| ตัวหนังสือ          |         | 1        | 1 | 1 | 1 | 1 |
| รูป                 |         | 1        | 1 | 0 | 1 | 1 |
| เว้นวรรค            |         | 0        | 0 | 1 | 0 | 0 |
| ย้อนกลับ (RETURN)   |         | 0        | 1 | 0 | 0 | 0 |
| ป้อนกลับ (LINEFEED) |         | 0        | 0 | 0 | 1 | 0 |
| ที่ว่าง             |         | 0        | 0 | 0 | 0 | 0 |

รหัสแอสกี (ASCII; American National Standard Code for Information Interchange) ผลิตโดยสถาบันมาตรฐานแห่งชาติของสหรัฐอเมริกา (American National Standard Institute) แต่ละอักขระจะมี 7 บิต ซึ่งเป็นบิต “0” และ “1” จำนวนอักขระจะเท่ากับ 128 อักขระ รหัสนี้จะมีบิตที่ 8 เป็นบิต ภาวะเสมอมูล (Parity Check Bit) โดยที่บิตที่ 1 ถึง 7 เป็นบิตข้อมูล ดังตารางที่ 2.5

ตารางที่ 2.5 รหัสแอสกี

| ลำดับบิต | อักขระ | ลำดับบิต | อักขระ | ลำดับบิต | อักขระ | ลำดับบิต | อักขระ |
|----------|--------|----------|--------|----------|--------|----------|--------|
| 7654321  |        | 7654321  |        | 7654321  |        | 7654321  |        |
| 0000000  | NUL    | 0100000  | PS     | 1000000  | @      | 1100000  | ,      |
| 0000001  | SIH    | 0100001  | !      | 1000001  | A      | 1100001  | a      |
| 0000010  | STX    | 0100010  | "      | 1000010  | B      | 1100010  | b      |
| 0000011  | ETX    | 0100011  | #      | 1000011  | C      | 1100011  | c      |
| 0000100  | EOT    | 0100100  | \$     | 1000100  | D      | 1100100  | d      |
| 0000101  | ENQ    | 0100101  | %      | 1000101  | E      | 1100101  | e      |
| 0000110  | ACK    | 0100110  | &      | 1000110  | F      | 1100110  | f      |

### ตารางที่ 2.5 (ต่อ) รหัสแอสกี

| ลำดับบิต | อักขระ | ลำดับบิต | อักขระ | ลำดับบิต | อักขระ | ลำดับบิต | อักขระ |
|----------|--------|----------|--------|----------|--------|----------|--------|
| 7654321  |        | 7654321  |        | 7654321  |        | 7654321  |        |
| 0000111  | BEL    | 0100111  | .      | 1000111  | G      | 1100111  | g      |
| 0001000  | BS     | 0101000  | (      | 1001000  | H      | 1101000  | h      |
| 0001001  | HT     | 0101001  | )      | 1001001  | I      | 1101001  | i      |
| 0001010  | LF     | 0101010  | *      | 1001010  | J      | 1101010  | j      |
| 0001011  | VT     | 0101011  | +      | 1001011  | K      | 1101011  | k      |
| 0001100  | FF     | 0101100  | ,      | 1001100  | L      | 1101100  | l      |
| 0001101  | CR     | 0101101  | -      | 1001101  | M      | 1101101  | m      |
| 0001110  | SO     | 0101110  | .      | 1001110  | N      | 1101110  | n      |
| 0001111  | SI     | 0101111  | /      | 1001111  | O      | 1101111  | o      |
| 0010000  | DLE    | 0110000  | 0      | 1010000  | P      | 1110000  | p      |
| 0010001  | DC1    | 0110001  | 1      | 1010001  | Q      | 1110001  | q      |
| 0010010  | DC2    | 0110010  | 2      | 1010010  | R      | 1110010  | r      |
| 0010011  | DC3    | 0110011  | 3      | 1010011  | S      | 1110011  | s      |
| 0010100  | DC4    | 0110100  | 4      | 1010100  | T      | 1110100  | t      |
| 0010101  | NAK    | 0110101  | 5      | 1010101  | U      | 1110101  | u      |
| 0010110  | SYN    | 0110110  | 6      | 1010110  | V      | 1110110  | v      |
| 0010111  | ETB    | 0110111  | 7      | 1010111  | W      | 1110111  | w      |
| 0011000  | CAN    | 0111000  | 8      | 1011000  | X      | 1111000  | x      |
| 0011001  | EM     | 0111001  | 9      | 1011001  | Y      | 1111001  | y      |
| 0011010  | SUB    | 0111010  | :      | 1011010  | Z      | 1111010  | z      |
| 0011011  | ESC    | 0111011  | ;      | 1011011  | [      | 1111011  | {      |
| 0011100  | FS     | 0111100  | <      | 1011100  | \      | 1111100  |        |
| 0011101  | GS     | 0111101  | =      | 1011101  | ]      | 1111101  | }      |
| 0011110  | RS     | 0111110  | >      | 1011110  | ^      | 1111110  | ~      |

รหัสเอชซีดีค (EBCDIC; Extended Binary Code Decimal Interchange Code) พัฒนาโดยบริษัท IBM แต่ละอักขระประกอบด้วย 8 บิต ซึ่งเป็น บิต “0” หรือ “1” จำนวนอักขระจะเท่ากับ 256 อักขระ เวลาอ่านจะอ่านบิตที่ 0, 1, 2, 3, ทางด้านซ้าย และอ่านบิตที่ 4, 5, 6, 7 ทางด้านบนของแนวตั้ง ดังตารางที่ 2.6

เอกสารนี้จัดทำขึ้นเพื่อใช้ในการเรียนการสอนเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.6 รหัสเอ็บซีดิก

|          |     |     |     |     |     |    |     |     |     |    |    |    |     |     |     |     |
|----------|-----|-----|-----|-----|-----|----|-----|-----|-----|----|----|----|-----|-----|-----|-----|
| บิตที่ 4 | 0   | 0   | 0   | 0   | 0   | 0  | 0   | 0   | 1   | 1  | 1  | 1  | 1   | 1   | 1   | 1   |
| 5        | 0   | 0   | 0   | 0   | 1   | 1  | 1   | 1   | 0   | 0  | 0  | 0  | 1   | 1   | 1   | 1   |
| 6        | 0   | 0   | 1   | 1   | 0   | 0  | 1   | 1   | 0   | 0  | 1  | 1  | 0   | 0   | 1   | 1   |
| 7        | 0   | 1   | 0   | 1   | 0   | 1  | 0   | 1   | 0   | 1  | 0  | 1  | 0   | 1   | 0   | 1   |
| 0123     |     |     |     |     |     |    |     |     |     |    |    |    |     |     |     |     |
| 0000     | NUL | SOH | ETX | PE  | HT  | LC | DEL |     |     |    |    |    | FF  | CR  | SO  | SI  |
| 0001     | DEL | DC1 | DC2 | DC3 | RES | NL | BS  | IL  | CAN | EM |    |    | IFS | IGS | TRS | IUS |
| 0010     |     |     | FS  |     | BYP | LF | EOB | PRE |     |    | SM |    |     | ENQ | ACK | BEL |
| 0011     |     |     | SYN |     | PN  | RS | UC  | EOT |     |    |    |    | DC4 | NAK |     | SUB |
| 0100     | SP  |     |     |     |     |    |     |     |     |    | Q  |    | <   | (   | +   |     |
| 0101     | &   |     |     |     |     |    |     |     |     |    | !  | \$ | *   | )   | ;   | ]   |
| 0110     | -   | /   |     |     |     |    |     |     |     |    |    | .  | %   | -   | >   | ?   |
| 0111     |     |     |     |     |     |    |     |     |     | \  | :  | #  | 2   | ,   | =   | "   |
| 1000     |     | a   | b   | c   | d   | e  | f   | g   | h   | i  |    |    |     |     |     |     |
| 1001     |     | j   | k   | l   | m   | n  | o   | p   | q   | r  |    |    |     |     |     |     |
| 1010     |     | ~   | s   | t   | u   | v  | w   | x   | y   | z  |    |    |     |     |     |     |
| 1011     |     |     |     |     |     |    |     |     |     |    |    |    |     |     |     |     |
| 1100     | {   | A   | B   | C   | D   | E  | F   | G   | H   | I  |    |    |     |     |     |     |
| 1101     | }   | J   | K   | L   | M   | N  | O   | P   | Q   | R  |    |    |     |     |     |     |
| 1110     |     |     | S   | T   | U   | V  | W   | X   | Y   | Z  |    |    |     |     |     |     |
| 1111     | 0   | 1   | 2   | 3   | 4   | 5  | 6   | 7   | 8   | 9  |    |    |     |     |     |     |

โดยที่อักขระพิเศษจะมีมากกว่า 1 ตัวอักษร เช่น NUL, SOH, STX, ETX, และ PF เป็นต้น

### 2.5.2 รูปแบบการสื่อสารข้อมูล

รูปแบบของการสื่อสารอาจทำได้หลายวิธี เช่น การสื่อสารระหว่างบุคคล การสื่อสารระหว่างคอมพิวเตอร์กับอุปกรณ์ข้างเคียง การสื่อสารระหว่างคอมพิวเตอร์และการสื่อสารภายในคอมพิวเตอร์ เป็นต้น การสื่อสารระหว่างบุคคลส่วนใหญ่แล้ว ข่าวสารจะถูกส่งในรูปแบบของเสียง เช่น วิทยุหรือโทรศัพท์ เป็นต้น เสียงเหล่านี้สามารถถูกแปลงเป็นข้อมูลโดยการพิมพ์ได้ การสื่อสารระหว่างคอมพิวเตอร์กับอุปกรณ์ข้างเคียง อุปกรณ์ข้างเคียงหมายถึง เครื่องพิมพ์หรืออุปกรณ์เชื่อมต่อก็ได้ เมื่อเครื่องพิมพ์พิมพ์งานตามคำสั่งที่ส่งมา การสื่อสารข้อมูลนั้นก็จะบรรลุตามต้องการ การสื่อสารระหว่างคอมพิวเตอร์มักจะเป็นการส่งทอดข้อมูล ในลักษณะเพิ่มข้อมูลฐานข้อมูล หรือการติดต่อแบบโต้ตอบในทันที ส่วนการสื่อสารภายในระบบคอมพิวเตอร์ เป็นการส่งข้อมูลระหว่างชั้น (Layer) ตามแบบจำลอง OSI

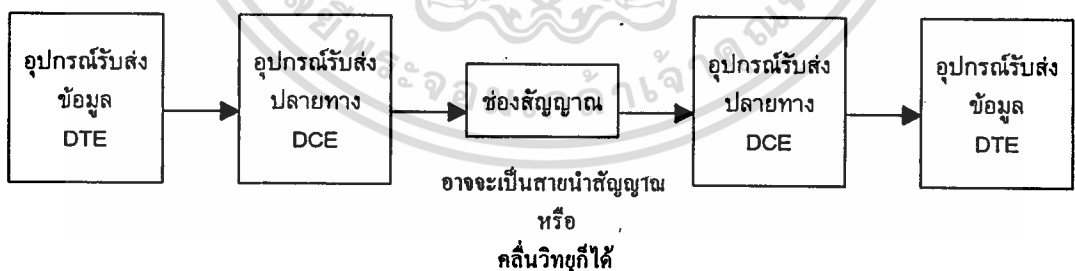
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 2.5.3 ประเภทการบริการของการสื่อสารข้อมูล

ประเภทของการให้บริการสื่อสารข้อมูลอาจจะแบ่งออกได้เป็นหลายรูปแบบ เช่น บริการที่ขึ้นกับวงจรสื่อสาร บริการที่ขึ้นอยู่กับวัตถุประสงค์ในการส่งข้อมูล และบริการที่ขึ้นอยู่กับวัตถุประสงค์ของการใช้ เป็นต้น บริการที่ขึ้นอยู่กับวงจรสื่อสารข้อมูลนี้อาจเป็นวงจรที่อนุญาตให้ใช้แตกต่างกัน บางวงจรมีไว้เฉพาะผู้ใช้งานบางประเภทใช้ได้ บางวงจรมีไว้สำหรับผู้ใช้งานโดยทั่วไปร่วมกัน ใช้ ด้านบริการที่ขึ้นอยู่กับวัตถุประสงค์ในการส่งข้อมูล มีหลายประเภท เช่น เป็นการกระจายข่าวสาร เป็นการแลกเปลี่ยนข่าวสารหรือการตอบคำถาม เป็นต้น ส่วนบริการที่ขึ้นอยู่กับวัตถุประสงค์ของการใช้นั้นมีอยู่มากมาย น่าจะเหมือนกับบริการที่ขึ้นอยู่กับวัตถุประสงค์ในการส่งข้อมูล แต่บางครั้งก็มีข้อมูลปลีกย่อยที่ต่างกัน

### 2.5.4 ระบบส่งสัญญาณข้อมูล

การส่งสัญญาณข้อมูลคือ การส่งข่าวสารให้แก่อุปกรณ์ประมวลข่าวสาร (Information Processing Equipment) โดยอุปกรณ์ประมวลข่าวสารที่รู้จักกันดีก็คือ คอมพิวเตอร์ ดังนั้นระบบส่งสัญญาณ จึงเป็นระบบที่เชื่อมโยงคอมพิวเตอร์กับอุปกรณ์รับส่งข้อมูลปลายทาง ให้สามารถติดต่อกันได้ ในระบบสื่อสารข้อมูลนั้นอุปกรณ์รับส่งปลายทาง (Data Terminal Equipment) ซึ่งสามารถเรียกย่อๆ ว่า DTE เป็นอุปกรณ์ที่กำเนิดและรับส่งข้อมูลในเครือข่ายคอมพิวเตอร์ ส่วนอุปกรณ์สื่อสารข้อมูล (Data Circuit Terminating Equipment) หรือเรียกย่อๆ ว่า DCE โดยจะใช้ร่วมกับอุปกรณ์รับส่งปลายทาง ดังรูปที่ 2.14



รูปที่ 2.14 ความสัมพันธ์ระหว่างอุปกรณ์รับส่งข้อมูล (DTE) และอุปกรณ์รับส่งปลายทาง (DCE) ในระบบสื่อสารข้อมูล

ในการสื่อสารข้อมูลนั้น มักจะมีอุปกรณ์รับส่งปลายทางอยู่หลายชุดวางอยู่กระจัดกระจายตามที่ต่างๆ ในการเชื่อมต่อกัน การจัดอุปกรณ์รับส่งปลายทาง เป็นรูปแบบของเครือข่ายที่เหมาะสมจะช่วยให้การประหยัดได้ รูปแบบของเครือข่ายที่นิยมใช้กันมากได้แก่

1) การต่อตรงระหว่างอุปกรณ์รับส่งปลายทางเข้าด้วยกัน ในลักษณะเป็นเส้นตรง หรือเรียกว่า จุดต่อจุด (Point-to-Point) ซึ่งมีผลดีคือเสถียรภาพของระบบสูง และก่อนที่จะทำการส่งสัญญาณข้อมูลก็ไม่ต้องมีการจองช่องสัญญาณ เหมาะที่จะใช้ในกรณีที่มีข่าวสารที่ต้องการความหนาแน่นสูงและมีการติดต่อกันมาก

2) การต่อระหว่างอุปกรณ์รับส่งปลายทางแบบหลายจุด (Multi-Point) หรือแบบหลายจุดร่วมตัวกลาง (Multi-Drop) หรือแบบบัส (Bus) มักใช้กับกรณีที่เป็นการติดต่อระหว่างคอมพิวเตอร์แม่กับอุปกรณ์รับส่งปลายทางหลายๆเครื่อง ซึ่งเป็นระบบประมวลผลแบบศูนย์รวม โดยใช้สายคู่เดียวต่อระหว่างคอมพิวเตอร์แม่กับอุปกรณ์รับส่งปลายทางการต่อแบบนี้ สถานีแม่หรืออุปกรณ์ที่ศูนย์รวมในกรณีนี้ได้แก่ คอมพิวเตอร์แม่ จะเวียนถามสถานีลูกหรืออุปกรณ์รับส่งปลายทางว่ามีความต้องการในการส่งหรือไม่ เรียกวิธีนี้ว่า การเลือกการหยั่งสัญญาณ (Polling Selecting) ถ้าสถานีลูกมีความต้องการในการส่งก็จะจัดช่องสัญญาณให้ใช้ในการติดต่อระหว่างสถานี

3) การต่อตรงระหว่างอุปกรณ์รับส่งปลายทางแบบสวิตชิง ทำได้โดยการติดต่อต่างๆ จะต้องผ่านอุปกรณ์สวิตชิง โดยจะไม่มีการแบ่งเป็นสถานีแม่และสถานีลูกเหมือนแบบบัส และจะต่อถึงกันระหว่างอุปกรณ์รับส่งปลายทางในช่วงที่มีความต้องการเท่านั้น เหมาะที่จะใช้กับกรณีที่มีจำนวนอุปกรณ์รับส่งปลายทางจำนวนมากๆ และมีการส่งข่าวสารในลักษณะกระจายออกไป

### 2.5.5 ประเภทของการส่งสัญญาณข้อมูล

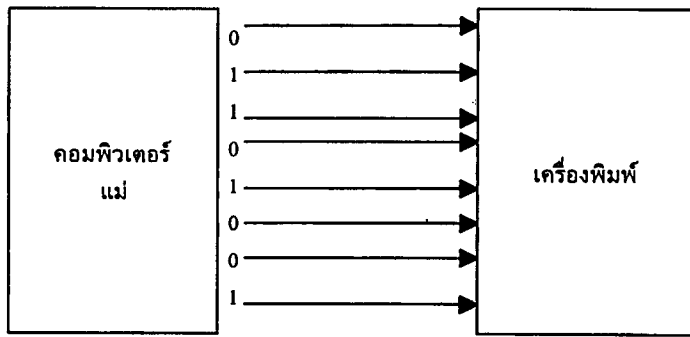
การส่งสัญญาณข้อมูลจะแบ่งออกได้เป็น 2 ประเภทคือการส่งแบบอนุกรมและแบบขนาน การส่งแบบขนานนั้นจะทำการส่งข้อมูลที่ละหลายๆบิต เช่น ส่ง 10011110 ทั้ง 8 บิตออกไปพร้อมกันโดยผ่านสายส่งข้อมูลที่มี 8 เส้น ส่วนการส่งข้อมูลแบบอนุกรม ข้อมูลจะถูกส่งออกไปทีละบิตต่อเนื่องกันไป เช่น ถ้าข้อมูลคือ 10011110 เลข 0 ทางขวามือสุดเป็นบิตที่ 1 เรียงลำดับไปจนครบ 8 บิต โดยการส่งนั้นจะใช้สายส่งเส้นเดียวเท่านั้น ดังรูปที่ 2.15 การส่งข้อมูลแบบขนานและแบบอนุกรม

### 2.5.6 การส่งข้อมูลแบบอนุกรม

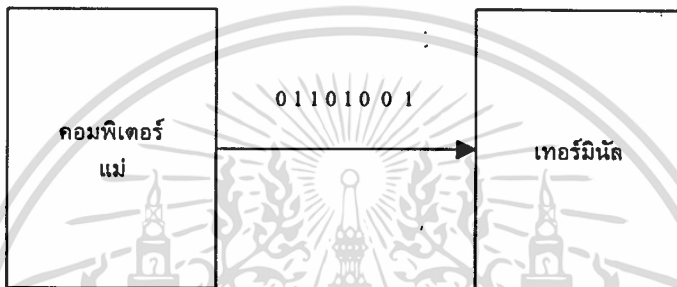
การส่งข้อมูลแบบอนุกรมแบ่งออกได้เป็น 2 ประเภทได้แก่

1. การส่งข้อมูลแบบอะซิงโครนัส (Asynchronous Data Transmission)
2. การส่งข้อมูลแบบซิงโครนัส (Synchronous Data Transmission)

การส่งข้อมูลแบบอะซิงโครนัสมักจะใช้กับเทอร์มินัลธรรมดา (Dumb Terminal) ไว้สำหรับรับข้อมูลจากคอมพิวเตอร์แม่และแสดงผลที่จอ โดยไม่สามารถเปลี่ยนแปลงข้อมูลได้ การส่งข้อมูลแบบนี้มักจะมีอัตราการรับส่งข้อมูลที่แน่นอนมีหน่วยเป็นบิตต่อวินาที (Bit Per Second) เมื่ออุปกรณ์อะซิงโครนัสจะส่งข้อมูล 1 ไบต์ ก็จะต้องส่งบิตเริ่มต้น (Start Bit) ก่อน ซึ่งมักจะเป็น "0" และตามด้วย



(ก) แบบขนาน



(ข) แบบอนุกรม

รูปที่ 2.15 การส่งข้อมูลแบบขนานและแบบอนุกรม

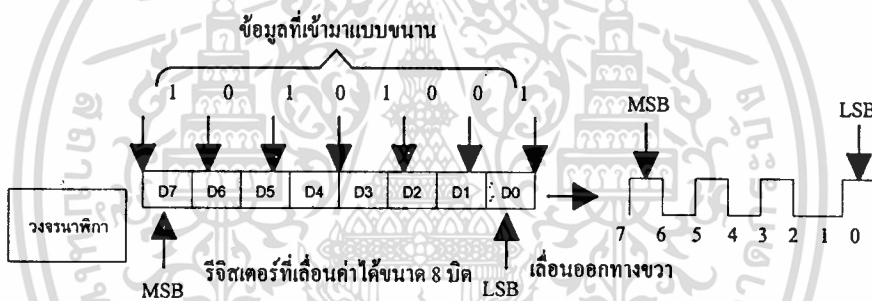
ข้อมูลทั้ง 8 บิตใน 1 ไบต์ แล้วจึงจะส่งบิตหยุด (Stop Bit) ซึ่งมักจะเป็น “1” บิตทั้งหมดนี้จะรวมกันเป็น 10 บิต ในการส่งข้อมูลเรียงลำดับดังนี้ 1 บิตเริ่มต้น 7 บิตข้อมูล (Data Bit) 1 บิตภาวะเสมอมูล และ 1 บิตหยุด กระบวนการเหล่านี้จะห่างกัน 1 วินาทีที่จะส่งข้อมูลชุดต่อไป ซึ่งหมายความว่าเมื่อคอมพิวเตอร์แม่ได้รับบิตเริ่มต้น ก็คาดหวังว่าจะได้รับอีก 9 บิตภายในเวลา 1 วินาที ในระบบนี้จะเกี่ยวข้องกับเวลาว่าเมื่อไรบิตต่อไปจะมาถึง ถ้าไม่ตรงตามที่กำหนดไว้ การส่งข้อมูลก็จะล้มเหลว ระบบนี้จะเกี่ยวข้องกับเวลาว่าเมื่อไรบิตต่อไปจะมาถึง ถ้าไม่ตรงตามที่กำหนดไว้ การส่งข้อมูลจะล้มเหลว ระบบนี้เหมาะในการส่งอักขระจากเทอร์มินัลมายังคอมพิวเตอร์แม่ทันที ซึ่งอัตราการส่งข้อมูลอาจจะเป็น 110, 300, 1,200, 2,400, 4,800, 9,600, 19,200 บิตต่อวินาที โดยที่ทางด้านส่งและด้านรับจะต้องมีการตั้งค่าความเร็วให้เท่ากัน

การส่งข้อมูลแบบซิงโครนัสจะไม่ใช้บิตเริ่มต้นและบิตหยุด จะไม่มีการหยุดชั่วขณะระหว่างอักขระ จะใช้วิธีให้จังหวะเวลาทั้งสองทางที่ติดต่อกัน มีอยู่สองวิธีที่ปฏิบัติกันคือ ใช้อักขระซิงก์ (Sync Character) หรือใช้สัญญาณนาฬิกา (Clock Signal) การใช้อักขระซิงก์ไว้หน้าบล็อกรหัส (block) ของอักขระที่ใหญ่ โดยการใส่อักขระซิงก์ไว้หน้าบล็อกรหัสของข้อมูล อักขระซิงก์นี้เป็นบิตใช้

จำนวนหนึ่งที่ทางอุปกรณ์เครื่องรับสามารถใช้ในการกำหนดอัตราเร็วของข้อมูลให้ตรงกับทางอุปกรณ์เครื่องส่ง การใช้สัญญาณนาฬิกาของด้านรับจะใช้คนละสายหรือคนละช่องสัญญาณในการส่งข่าวสารเกี่ยวกับเวลาของข้อมูลที่จะส่ง โดยทั่วไปการส่งข้อมูลแบบซิงโครนัสจะทำงานภายใต้การควบคุมของโปรโตคอลในระบบนั้นๆ และนิยมใช้กับเทอร์มินัลตลาดและเทอร์มินัลอัจฉริยะ

### 2.5.7 การเชื่อมต่อแบบอนุกรมและ UART

จากรูปที่ 2.16 การแปลงข้อมูลแบบขนานเป็นข้อมูลแบบอนุกรม โดยเริ่มแรกข้อมูลแบบขนานจะถูกนำไปเก็บไว้ในรีจิสเตอร์ที่เลื่อนค่าได้ (Shift Register) จากนั้นเราจะใช้สัญญาณนาฬิกาในการเลื่อนค่าในรีจิสเตอร์ออกมาทีละบิต (โดยจะเลื่อนค่าไปทางขวามือ) โดยบิตแรกที่ถูกเลื่อนออกมาคือ บิต LSB ของข้อมูลและบิตทั้งสองที่ถูกเลื่อนออกมาก็คือ บิตที่อยู่ถัดจากบิต LSB และบิตต่อไป สำหรับบิตสุดท้ายที่ถูกเลื่อนออกมาก็คือ บิต MSB ของข้อมูล



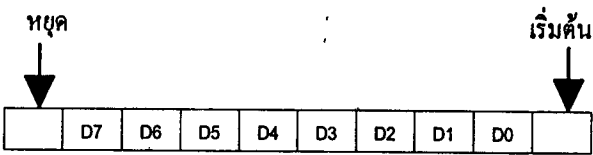
รูปที่ 2.16 การแปลงข้อมูลแบบขนานเป็นข้อมูลอนุกรม

การแปลงข้อมูลแบบอนุกรมไปเป็นข้อมูลแบบขนานนั้นจะมีขั้นตอนตรงกันข้ามกับที่กล่าวมา นั่นคือข้อมูลแบบอนุกรมจะถูกเลื่อนเข้าไปเก็บในรีจิสเตอร์ที่เลื่อนค่าได้ โดยใช้สัญญาณนาฬิกาเป็นตัวควบคุม และหลังจากที่ได้มีการเลื่อนข้อมูลทุกบิตเข้าไปในรีจิสเตอร์ที่เลื่อนค่าได้หมดแล้ว ข้อมูลในรีจิสเตอร์นี้ก็จะถูกนำออกมาแบบขนานเพื่อนำไปให้กับไมโครโปรเซสเซอร์

อุปกรณ์ที่ทำหน้าที่แปลงข้อมูลแบบอนุกรมเป็นข้อมูลแบบขนานและแปลงข้อมูลแบบขนานเป็นข้อมูลแบบอนุกรมเราเรียกว่า UART (Universal Asynchronous Receiver Transmitter) UART เป็นวงจร LSI ซึ่งนอกจากจะมีหน้าที่ในการแปลงข้อมูลแล้วมันยังมีหน่วยควบคุมและหน่วยตรวจสอบการทำงานอีกด้วย

ในการส่งข้อมูลขนาด 8 บิตแบบอนุกรมนี้จะต้องมีบิตเริ่มต้น (Start Bit) และบิตหยุด (Stop Bit) เพิ่มขึ้นมา ซึ่งจะทำให้ข้อมูลที่ถูกส่งไปจริงนั้นมีขนาด 10 บิต ในรูปที่ 2.17 เวิร์ดข้อมูลขนาด 8 บิตที่มีบิตเริ่มต้น 1 บิต และบิตหยุด 1 บิต เพิ่มขึ้นมา โดยบิตเริ่มต้น (มีค่าเป็น 0 เสมอ) จะบอก

UART ที่รับข้อมูลให้รู้ว่ามีข้อมูลกำลังเข้ามา และบิตหยุด (มีค่าเป็น 1 เสมอ) จะบอก UART ที่รับข้อมูลให้รู้ว่าการส่งข้อมูลได้เสร็จสิ้นลงแล้ว



รูปที่ 2.17 เวิร์ดข้อมูลขนาด 8 บิตกับบิตเริ่มต้นและบิตหยุดที่ใช้ในการโอนย้ายข้อมูลแบบอนุกรม

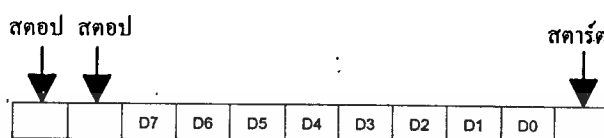
เราเรียกความเร็วในการส่งข้อมูลของ UART ว่าอัตราบอด (Baud Rate) ซึ่งอัตราบอดนี้จะบอกจำนวนบิตที่ถูกส่งไปใน 1 วินาที เช่น การส่งข้อมูลด้วยอัตรา 1200 บอด ก็คือการส่งตัวอักขระขนาด 10 บิต (ประกอบด้วยบิตเริ่มต้น 1 บิต บิตข้อมูล 8 บิต และบิตหยุด 1 บิต) ได้ 120 ตัวอักขระใน 1 วินาที ตารางที่ 2.7 อัตราบอดของ UART ที่ใช้กันโดยทั่วไป

ตารางที่ 2.7 อัตราบอดที่ใช้ในการโอนย้ายข้อมูล

| อัตราบอด | ไบต์/วินาที |
|----------|-------------|
| 110      | 1           |
| 150      | 15          |
| 300      | 30          |
| 600      | 60          |
| 1200     | 120         |
| 2400     | 240         |
| 4800     | 480         |
| 9600     | 960         |
| 19200    | 1920        |
| 38400    | 3840        |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การส่งข้อมูลด้วยอัตรา 110 บอดนั้นจะมีรูปแบบแตกต่างจากอัตราบอดอื่น ซึ่งในอัตราบอดนี้จะต้องใช้บิตเริ่มต้น 1 บิต และบิตหยุด 2 บิต ดังนั้นจึงทำให้เราต้องส่งข้อมูลที่มีขนาด 11 บิต ในรูปที่ 2.18 ลักษณะเวิร์คขนาด 11 บิตที่ใช้ระบบอัตรา 110 บอด การส่งข้อมูลด้วยอัตรานี้จะพบได้ในเครื่องโทรพิมพ์ แต่ไม่ได้รับความนิยมในปัจจุบัน



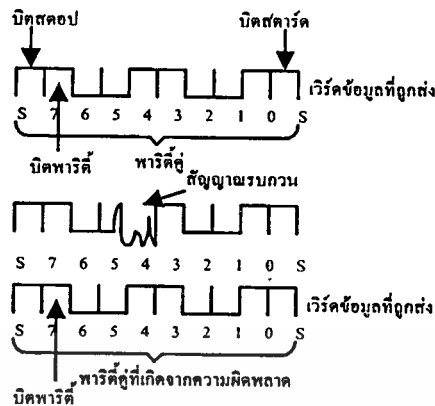
รูปที่ 2.18 เวิร์คข้อมูลขนาด 8 บิต ที่ใช้ในการโอนย้ายข้อมูลแบบอนุกรมด้วยอัตรา 110 บอด

เมื่อเรานำบิตที่ 8 ของข้อมูลมาใช้ในการตรวจสอบความผิดพลาดในการสื่อสารข้อมูลซึ่งเราเรียกว่า บิตพาริตี (Parity Bit) UART ส่วนใหญ่สามารถสร้างและทำการตรวจสอบข้อมูลนั้นว่าเป็นพาริตีคู่หรือเป็นพาริตีคี่ได้ ในการสร้างพาริตีคู่ UART จะทำการเซตหรือเคลียร์ค่าในบิตพาริตีเพื่อให้ข้อมูลทั้ง 8 บิตนั้นมีเลข 1 จำนวนคู่ตัว และในการสร้างพาริตีคี่ UART จะทำการเซตหรือเคลียร์ค่าในบิตพาริตีเพื่อให้ข้อมูล 8 บิตนั้นมีเลข 1 จำนวนคี่ตัว

เราจะใช้พาริตีคู่หรือพาริตีคี่ในการตรวจสอบความผิดพลาดในการส่งข้อมูล เมื่อ UART ได้รับข้อมูลมันก็จะทำการทดสอบว่าข้อมูลนั้นเป็นพาริตีคู่หรือพาริตีคี่ ถ้า UART พบว่าข้อมูลที่มันทดสอบมีค่าพาริตีไม่ตรงตามที่กำหนด บิตพาริตีในรีจิสเตอร์สถานะของ UART ก็จะถูกเซตเพื่อแสดงว่าข้อมูลที่ได้รับเข้ามามีข้อผิดพลาด และโปรแกรมที่ทำการรับข้อมูลนั้นก็ขอให้มีการส่งข้อมูลให้กับ UART ใหม่อีกครั้ง การใช้พาริตีนี้จะสามารถตรวจจับความผิดพลาดได้เพียง 50 เปอร์เซ็นต์ของความผิดพลาดทั้งหมด ทั้งนี้เนื่องจากมันจับจะความผิดพลาดได้เฉพาะในกรณีที่ข้อมูลนั้นเกิดข้อผิดพลาดเป็นจำนวนคี่บิต แต่ถ้าข้อมูลนั้นเกิดความผิดพลาดเป็นจำนวนเลขคู่บิตก็จะไม่ทำให้ค่าพาริตีเปลี่ยนแปลง และความผิดพลาดนี้ก็จะไม่ถูกตรวจพบ

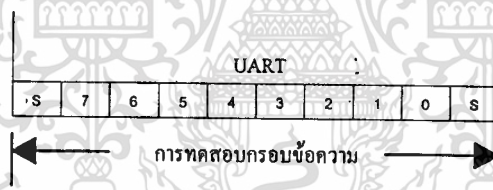
ในรูปที่ 2.19 เป็นการส่งข้อมูลแบบอนุกรมที่มีค่าพาริตีเปลี่ยนไป เนื่องจากสัญญาณรบกวน (Noise) โดยบิตที่ 4 ในรูปจะถูกสัญญาณรบกวนทำให้มีการเปลี่ยนค่าจาก 1 ไปเป็น 0 ซึ่งทำให้ข้อมูลที่ได้รับนั้นไม่ถูกต้อง

UART ยัง สามารถตรวจจับการเฟรมมิ่งเออเรอร์ (Framing Error) ได้ การเฟรมมิ่งเออเรอร์จะเกิดขึ้นเมื่อ UART พลาดบิตเริ่มต้นซึ่งหมายความว่า UART นั้นได้รับข้อมูลที่มีบิตเริ่มต้นและบิตหยุดอยู่ในตำแหน่งที่ไม่ถูกต้อง การใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.19 การใช้พาริตีเพื่อตรวจสอบความผิดพลาดในการโอนย้ายข้อมูลแบบอนุกรม

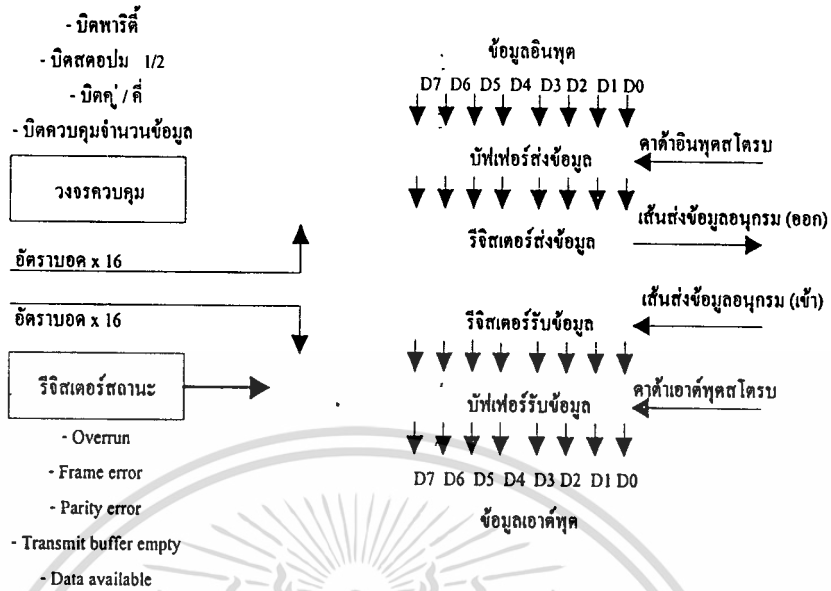
ในรูปที่ 2.20 การรับส่งข้อมูลของ UART ที่เกิดการเฟรมมิ่งเออเรอร์ จากรูปเราจะเห็นได้ว่า ข้อมูลที่รับเข้ามานั้นเรียงกันอย่างไม่ถูกต้องในรีจิสเตอร์ที่เลื่อนค่าได้ สัญญาณรบกวนในเส้นส่งสัญญาณก็เป็นสาเหตุหนึ่งที่ทำให้เกิดการเฟรมมิ่งเออเรอร์



รูปที่ 2.20 เฟรมมิ่งเออเรอร์ ข้อมูลที่รับเข้ามาที่ตำแหน่งเริ่มต้นและตำแหน่งสิ้นสุดไม่ถูกต้อง

จากรูปที่ 2.21 แผนผังการทำงานของ UART แบบง่าย ๆ จากรูปเราจะพบว่าเราสามารถแบ่ง UART ออกเป็น 4 ส่วนได้แก่ ส่วนที่หน้าที่ส่งข้อมูล ส่วนที่ทำหน้าที่รับข้อมูลส่วนที่กำหนดสถานะและส่วนที่เป็นวงจรควบคุม

ส่วนที่มีหน้าที่ส่งข้อมูลของ UART แยกออกเป็นได้เป็น 2 ส่วน ซึ่งได้แก่ บัฟเฟอร์ส่งข้อมูล (Transmitted-Data Output Buffer) กับรีจิสเตอร์ส่งข้อมูล (Transmit Register) โดยรีจิสเตอร์ส่งข้อมูลจะมีหน้าที่เลื่อนข้อมูลส่งออกไปยังเส้นส่งข้อมูลอนุกรม (ออก) โดยเริ่มจากบิตเริ่มต้นถัดมาเป็นบิตข้อมูล D0 ถึง D7 และบิตสุดท้ายคือ บิตหยุดข้อมูลขนาด 8 บิตที่จะส่งจะถูกนำไปเก็บในบัฟเฟอร์ส่งข้อมูล เมื่อสัญญาณควบคุมดาต้าอินพุต สโตรบ (Data Input Strobe) เปลี่ยนค่าจาก 1 เป็น 0 และการส่งข้อมูลแบบอนุกรมจะเริ่มขึ้นเมื่อสัญญาณควบคุมนี้เปลี่ยนค่าจาก 0 เป็น 1 ดังรูปที่ 2.22 ส่วนที่ทำหน้าที่รับข้อมูลของ UART จะเหมือนเป็นภาพสะท้อนของส่วนที่ทำหน้าที่ส่งข้อมูลที่เข้า



รูปที่ 2.21 แผนผังการทำงานของ UART

ข้อมูลถูกนำไป  
เก็บในบัฟเฟอร์ส่งข้อมูล

เริ่มการส่งข้อมูล  
แบบอนุกรม

รูปที่ 2.22 สัญญาณคาล์เอาต์พุตสโตรบ

มาทางเส้นรับข้อมูลอนุกรม (เข้า) จะถูกเลื่อนเข้าไปเก็บในรีจิสเตอร์รับข้อมูล โดยการเลื่อนค่า 10 หรือ 11 ครั้ง ขึ้นตอณดังกล่าวจะเริ่มเมื่อมีบิตเริ่มต้นเข้ามา และทันทีที่ข้อมูลทั้งหมดถูกเลื่อนเข้าไปเก็บในรีจิสเตอร์รับข้อมูล (Receive Register) แล้ว ข้อมูลในรีจิสเตอร์นี้จะถูกนำไปเก็บในบัฟเฟอร์รับข้อมูล (Received-Data Output Buffer) เมื่อมีสัญญาณควบคุมคาล์เอาต์พุตสโตรบ

ทั้งรีจิสเตอร์รับข้อมูลและรีจิสเตอร์ส่งข้อมูลจะได้รับสัญญาณนาฬิกาขนาด 16 หรือ 64 เท่าของอัตราบอดที่เราเลือกสำหรับการเลื่อนค่าในรีจิสเตอร์นั้นๆ

พิจารณาในรูปที่ 2.21 เราจะพบว่าข้อมูลในรีจิสเตอร์สถานะของ UART สามารถถูกนำไปเก็บในบัฟเฟอร์ข้อมูลได้ ซึ่งทำได้โดยการส่งสัญญาณควบคุมสเคตต์เอาต์พุตสโตรบ (Status Output Strobe) ซึ่งขึ้นตอณดังกล่าวจะทำให้ไมโครโปรเซสเซอร์สามารถอ่านข้อมูลจากรีจิสเตอร์สถานะได้

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รีจิสเตอร์สถานะของ UART จะมีบิตบอกสถานะดังต่อไปนี้

1. บิต OR (Overrun) บิตนี้จะมีค่าเป็น 1 เมื่อข้อมูลชิ้นใหม่เข้ามาทับข้อมูลชิ้นเดิมที่เก็บอยู่ ซึ่งข้อมูลชิ้นเดิมนี้อย่างไม่ได้ถูกนำไปเก็บในบัฟเฟอร์รับข้อมูล

2. บิต FE (Framing Error) บิตนี้จะมีค่าเป็น 1 ถ้า UART ไม่พบบิตหยุด ซึ่งอาจเป็นเพราะว่า UART ไม่ได้อ่านบิตเริ่มต้นจากตำแหน่งที่ถูกตั้งซึ่งก็หมายความว่ามันเริ่มทำงานกับบิตที่ไม่ใช่บิตเริ่มต้น

3. บิต PE (Parity Error) บิตนี้จะมีค่าเป็น 1 เมื่อ UART ได้ตรวจสอบพบว่าข้อมูลที่มันได้รับมีค่าพาริตีไม่ตรงกับค่าในบิตพาริตีของข้อมูล เราจะสามารถให้ UART ทำการทดสอบค่าแบบพาริตีคู่หรือพาริตีคี่ได้โดยระบุไว้ที่วงจรควบคุมก่อนที่จะนำ UART นี้มาทำการรับข้อมูล

4. บิต TBE (Transmit Buffer Empty) บิตนี้จะมีค่าเป็น 1 เมื่อ UART ได้ทำการส่งข้อมูลออกไปแล้ว ซึ่งทำให้เราสามารถนำข้อมูลชิ้นต่อไปที่จะส่งเก็บลงในบัฟเฟอร์ส่งข้อมูลได้

5. บิต DA (Data Available) บิตนี้จะมีค่าเป็น 1 เมื่อ UART ได้รับข้อมูลชิ้นใหม่เข้ามาและไมโครโปรเซสเซอร์สามารถทำการอ่านข้อมูลนี้ได้โดยส่งสัญญาณค่าเอาต์พุตสโตรบ

หลังจากที่ข้อมูลในรีจิสเตอร์สถานะของ UART ถูกนำไปเก็บในบัฟเฟอร์รับข้อมูลแล้ว เราสามารถเคลื่อนย้ายข้อมูลนี้เข้าไปยังแอสคิวเลเตอร์ของไมโครโปรเซสเซอร์ได้ซึ่งในที่นี้จะทำให้เราสามารถใช้งานคำสั่งเปรียบเทียบและการทดสอบอื่นๆ เพื่อว่ามีบิตบอกสถานะใดบ้างที่ถูกเซต

เราสามารถนำค่าในรีจิสเตอร์สถานะของ UART ที่เก็บในแอสคิวเลเตอร์มาเปรียบเทียบกับค่า 02 H ได้โดยใช้คำสั่งเปรียบเทียบ คำสั่งการเปรียบเทียบนี้จะทำให้เกิดการเซตค่าในบิตที่ศูนย์ของรีจิสเตอร์สถานะในไมโครโปรเซสเซอร์ ถ้าบิต DA ของรีจิสเตอร์สถานะใน UART เท่านั้นที่ถูกเซต แต่ถ้าบิตอื่นๆของรีจิสเตอร์สถานะใน UART ถูกเซตแสดงว่าได้เกิดปัญหาขึ้นกับข้อมูลที่ UART รับเข้ามา เราสามารถใช้ซอฟต์แวร์เพื่อตรวจหาปัญหาที่เกิดขึ้นหรือให้มีการส่งข้อมูลชิ้นเดิมมาให้กับ UART อีกครั้ง

รีจิสเตอร์ควบคุมการทำงานของ UART ทำให้เราสามารถกำหนดโหมดการทำงานของ UART ได้ (ดูรูปที่ 2.21) บิตควบคุมจำนวนข้อมูลทั้งสองบิต จะช่วยให้เราสามารถเลือกจำนวนของข้อมูลจริงที่รับมาว่าเป็น 5 บิต 7 บิต หรือ 8 บิต โดยเราจะใช้ข้อมูลจริงขนาด 7 บิต ในการส่งข้อมูลที่ไอร์แลนด์สกี ข้อมูลจริงขนาด 5 บิตจะใช้ในการส่งข้อมูลของเครื่องโทรพิมพ์ ข้อมูลจริงขนาด 6 บิตจะใช้ในการส่งข้อมูลที่มีขนาดเล็ก ข้อมูลจริงขนาด 8 บิตจะใช้เมื่อไม่มีบิตพาริตี แต่ผู้ใช้ต้องการที่จะส่งข้อมูลขนาด 1 ไบต์เต็ม

บิตควบคุมคู่ที่จะใช้ในการกำหนดพาริตีที่ใช้ว่าเป็นพาริตีคู่หรือพาริตีคี่ แต่บิตควบคุมนี้จะถูกใช้ก็ต่อเมื่อเรากำหนดให้มีการใช้ค่าพาริตีเท่านั้น (โดยการกำหนดค่าบิตพาริตี) ในการส่งข้อมูล

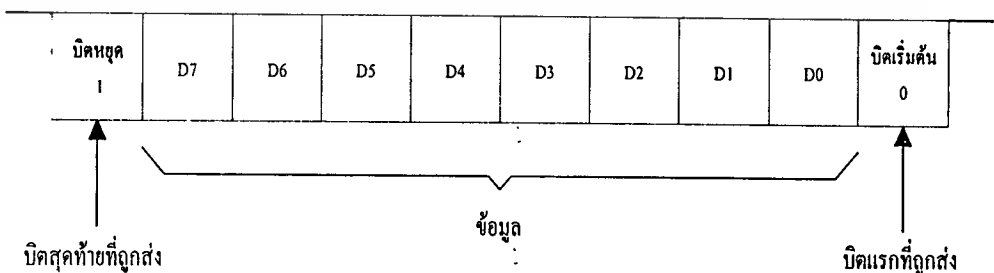
รหัสแอสกีขนาด 7 บิต บิตที่ 8 จะเป็นบิตพาริตีและเป็นบิตสุดท้ายของข้อมูลที่ส่งการกำหนดบิตคู่/คี่นี้จะเป็นการบอก UART ว่าให้ทำการเช็คค่าข้อมูลที่มันได้รับแบบพาริตีคู่หรือพาริตี

บิตหยุดของวงจรควบคุมจะกำหนดจำนวนบิตหยุดที่ใช้ในการส่งข้อมูล โดยการส่งข้อมูลด้วยอัตรา 110 บอด จะต้องใช้บิตหยุดจำนวน 2 บิตเสมอ บางครั้งอาจมีบิตควบคุมพิเศษเพิ่มเข้ามาซึ่งทำให้เราสามารถเลือกใช้บิตสตอปจำนวน 1.5 บิต ซึ่งจะใช้ในการส่งข้อมูลมาตรฐานแบบเก่าขนาด 5 บิต

อัตราการส่งข้อมูล UART จะไม่ขึ้นอยู่กับการทำงานของไมโครโปรเซสเซอร์ โดยไมโครโปรเซสเซอร์และ UART จะมีวงจรสร้างสัญญาณนาฬิกาเป็นของตนเอง แต่ถ้าเราพบว่า ไมโครโปรเซสเซอร์และ UART ใช้สัญญาณนาฬิกาจากวงจรตัวเดียวกันไม่ได้หมายความว่าไมโครโปรเซสเซอร์และ UART ทำงานร่วมกันอย่างเข้าจังหวะ แต่การทำงานเช่นนี้ก็เพื่อ เป็นการลดส่วนของวงจรฮาร์ดแวร์ที่ใช้สร้างสัญญาณนาฬิกา

การส่งข้อมูลของ UART จะเป็นแบบอะซิงโครนัส ซึ่งก็หมายความว่าเวลาระหว่างเวิร์ดข้อมูลแต่ละตัวที่รับเข้ามาจะไม่แน่นอน เช่น ในคีย์บอร์ดเราจะพบว่าเวลาระหว่างการกดปุ่มแต่ละบิตที่ส่งออกไปจะเป็นแบบอะซิงโครนัส แต่เวลาระหว่างตัวอักษรแต่ละตัวจะเป็นแบบอะซิงโครนัส เนื่องจากในการส่งตัวอักษรตัวหนึ่งๆ บิตแต่ละบิตของตัวอักษรนี้จะถูกส่งไปด้วยอัตราบอดเดียวกัน แต่สำหรับเวลาระหว่างตัวอักษรแต่ละตัวนั้นจะขึ้นอยู่กับอัตราการกดปุ่มคีย์บอร์ดของผู้ใช้

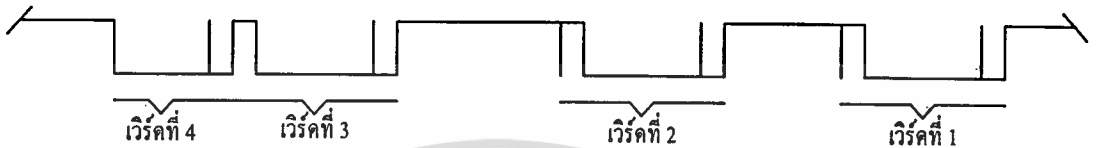
ในรูปที่ 2.23 เป็นการส่งตัวอักษรในรหัสแอสกีจาก UART โดยใช้การส่งแบบอนุกรม เราจะพบว่าบิตเริ่มต้นมีค่าเป็น 0 เสมอ บิตข้อมูล D0 ถึง D7 จะมีทั้งบิตที่เป็น 0 หรือ เป็น 1 โดยจะขึ้นอยู่กับข้อมูลที่เราส่งไป บิตพาริตีอาจมีค่าเป็น 0 หรือ 1 ก็ได้ และบิตหยุดจะมีค่าเป็น 1 เสมอ หลังจากที่เราส่งตัวอักษรหนึ่งๆ ออกไปแล้ว สายส่งสัญญาณนั้นจะคงค่าเป็น 1 เสมอจนกว่าจะมีการส่งบิตเริ่มต้นตัวใหม่เข้าไป



รูปที่ 2.23 การโอนย้ายข้อมูลแบบอนุกรมที่เป็นตัวอักษรในรหัสแอสกี

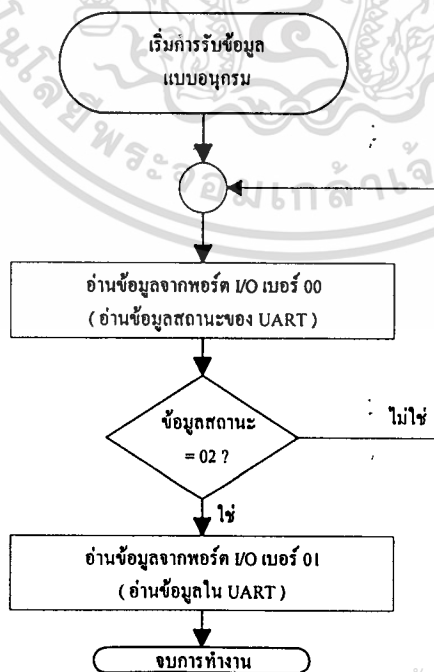
เอกสารนี้เป็นเอกสารลิขสิทธิ์ของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี อนุญาตให้นำไปใช้  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในรูปที่ 2.24 แสดงการส่งตัวอักขระ 4 ตัว โดยใช้การส่งแบบอนุกรม เราจะพบว่า การส่งข้อมูลนี้เป็นแบบอะซิงโครนัส และมีลักษณะเฉพาะ 2 ประการ ประการแรก จะเห็นได้ว่าเวลา ระหว่างตัวอักขระแต่ละตัวนั้นจะไม่แน่นอน ประการที่สอง ค่าระหว่างตัวอักขระแต่ละตัวจะมีค่าตามบิตหยุดคือมีค่าเป็น 1



รูปที่ 2.24 การโอนย้ายข้อมูลแบบอนุกรมที่ประกอบไปด้วยตัวอักขระ 4 ตัว

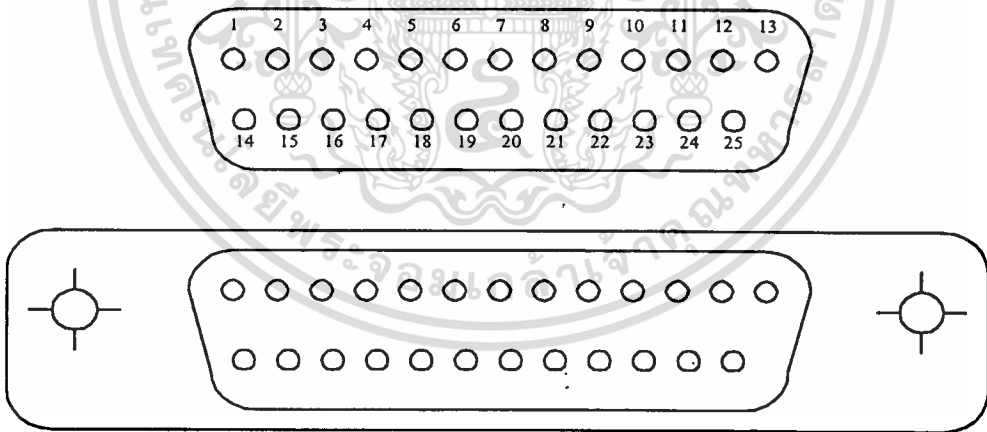
ไมโครโปรเซสเซอร์จะทำการตรวจสอบค่าในรีจิสเตอร์สถานะของ UART เป็น ช่วงๆ ซึ่งถ้ามันพบว่าบิต DA (Data Available) ในรีจิสเตอร์สถานะของ UART มีค่าเป็น 1 มันก็จะทำการอ่านข้อมูลจาก UART แต่ถ้าบิตนี้มีค่าเป็น 0 ไมโครโปรเซสเซอร์ก็จะต้องทำการตรวจสอบค่าที่มีอยู่ในตัวรีจิสเตอร์สถานะต่อไปจนกว่าบิต DA จะถูกเซตให้เป็น 1 จากรูปที่ 2.25 จะเป็นลักษณะขั้นตอนการทำงานของไมโครโปรเซสเซอร์



รูปที่ 2.25 ขั้นตอนการทำงานของไมโครโปรเซสเซอร์

### 2.5.8 มาตรฐานการเชื่อมต่อแบบ RS-232

RS-232 หรือ V.24 เป็นมาตรฐานการเชื่อมต่อระหว่างอุปกรณ์รับส่งปลายทาง (DTE) กับ อุปกรณ์สื่อสารข้อมูล (DCE) เพื่อใช้ในการส่งทอดข้อมูลไบนารี (Binary) แบบอนุกรม โดยที่ RS-232 เป็นมาตรฐานที่กำหนดขึ้นมาจากสมาคมอิเล็กทรอนิกส์อุตสาหกรรม (Electronic Industries Association : EIA) ซึ่งเกิดขึ้นมาด้วยความร่วมมือของบริษัท Bell ซึ่งเป็นบริษัทผู้ผลิตคอมพิวเตอร์ และโมเด็มเพื่อใช้เป็นมาตรฐานการเชื่อมต่อ ตัวอักษรภาษาอังกฤษของคำว่า “RS” ย่อมาจากคำว่า “Recommended Standard” แล้วตามด้วยเลข 3 ตัว และปิดท้ายตัวเลขทั้ง 3 ด้วยตัวอักษรภาษาอังกฤษ A, B, C, และ D เป็นการแสดงถึงรุ่นที่ได้รับ การพัฒนา มาตรฐาน RS-232 จะเทียบเท่ากับ มาตรฐาน V ที่สร้างขึ้นโดยองค์การสหภาพโทรคมนาคมระหว่างประเทศหรือ CCITT คุณสมบัติทางกลของ RS-232 ได้กำหนดให้ใช้เคเบิลที่มีขนาด 25 เข็มและ 9 เข็มดังรูปที่ 2.26 และรูปที่ 2.27 เข็มแต่ละเข็มจะมีสายเคเบิลต่ออยู่ 1 เส้น ทำให้ต้องใช้สายเคเบิลจำนวน 25 เส้นและ 9 เส้นที่มีลักษณะเป็นแพ คุณสมบัติทางไฟฟ้าของ RS-232 ที่ใช้ในการเชื่อมต่อสัญญาณดิจิทัลนั้น แรงดันไฟบวกในช่วง +3 ถึง +25 โวลต์ แทนบิต “0” หรือ “on” และใช้แรงดันไฟลบในช่วง -3 ถึง -25 โวลต์แทนบิต “1” หรือ “off” ส่วนแรงดันไฟฟ้าในช่วง -3 ถึง +3 โวลต์นั้นเป็นช่วงที่เปลี่ยนแปลง และไม่มีเสถียรภาพ



(ก) ตัวเชื่อมต่อ DB-25

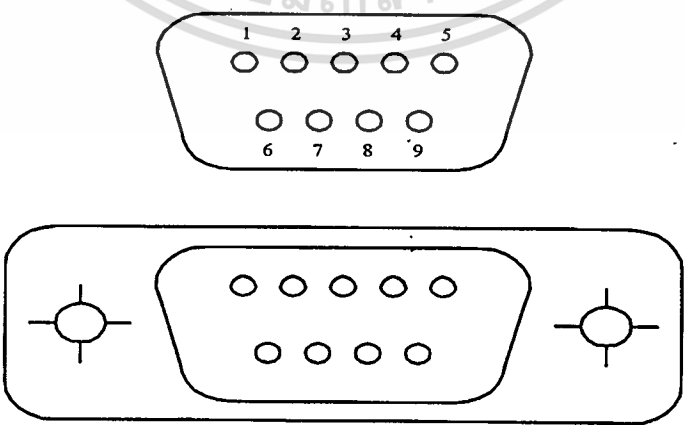
รูปที่ 2.26 ลักษณะของเคเบิล DB - 25 ที่ใช้ในการเชื่อมต่อทางกายภาพตามมาตรฐาน RS-232

## 2.6 หลักการและโครงสร้างของ MCS-51

ไมโครคอนโทรลเลอร์ MCS-51 ผลิตโดยบริษัทอินเทล มีการนำมาใช้ประโยชน์กันอย่างมากมายในปี ค.ศ.1980 ต่อมาบริษัท Phillips และ Siemens ได้รับลิขสิทธิ์ในการผลิตจำหน่ายและใช้

ตารางที่ 2.8 มาตรฐานของเค้าเสียบบ DB - 25

| 25<br>Pin | EIA<br>RS-232<br>Circuit | CCITT<br>V.24<br>Circuit | RS-232<br>DESCRIPTION                   | Signal type &<br>direction |
|-----------|--------------------------|--------------------------|-----------------------------------------|----------------------------|
| 1*        | AA                       | 101                      | Protective ground                       | Ground                     |
| 7*        | AB                       | 102                      | Signal ground / common return           | Ground/common              |
| 2*        | BA                       | 103                      | Transmitted data                        | Data to DCE                |
| 3*        | BB                       | 104                      | Received data                           | Data from DCE              |
| 4*        | CA                       | 105                      | Request to send                         | Control to DCE             |
| 5*        | CB                       | 106                      | Clear to send                           | Control from DCE           |
| 6*        | CC                       | 107                      | Data set ready                          | Control from DCE           |
| 20*       | CD                       | 108.2                    | Data terminal ready                     | Control to DCE             |
| 22        | CE                       | 125                      | Ring indicator                          | Control from DCE           |
| 8         | CF                       | 109                      | Received line signal detector           | Control from DCE           |
| 21        | CG                       | 110                      | Signal quality detector                 | Control from DCE           |
| 23        | CH                       | 111                      | Data signal rate selector (DTE)         | Control to DCE             |
| 23        | CI                       | 112                      | Data signal rate selector (DCE)         | Control from DCE           |
| 24        | DA                       | 113                      | Transmitter signal element timing (DTE) | Timing to DCE              |
| 15        | DB                       | 114                      | Transmitter signal element timing (DCE) | Timing from DCE            |
| 17        | DD                       | 115                      | Receiver signal element timing (DCE)    | Timing from DCE            |
| 14        | SBA                      | 118                      | Secondary transmitted data              |                            |
| 16        | SBB                      | 119                      | Secondary received data                 |                            |
| 19        | SCA                      | 120                      | Secondary request to send               | Control to DCE             |
| 13        | SCB                      | 121                      | Secondary clear to send                 | Control from DCE           |
| 12        | SCF                      | 122                      | Secondary received line signal detector | Control from DCE           |
| 11        |                          |                          | Undefined                               |                            |
| 18        |                          |                          |                                         |                            |
| 25        |                          |                          |                                         |                            |



(ข) ตัวเชื่อมต่อ DB-9

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและตัวอย่างอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 2.27 ลักษณะของเค้าเสียบบ DB - 9 ที่ใช้ในการเชื่อมต่อทางกายภาพตามมาตรฐาน RS-232

ตารางที่ 2.9 มาตรฐานของเค้าเทียบ DB - 9

| 25<br>Pin | EIA<br>RS-232<br>Circuit | CCITT<br>V.24<br>Circuit | RS-232<br>DESCRIPTION         | Signal type &<br>direction |
|-----------|--------------------------|--------------------------|-------------------------------|----------------------------|
| 5         | AB                       | 102                      | Signal ground / common return | Ground/common              |
| 2         | BB                       | 104                      | Received data                 | Data from DCE              |
| 3         | BA                       | 103                      | Transmitted data              | Data to DCE                |
| 1         | CF                       | 109                      | Received line signal detector | Control from DCE           |
| 4         | CD                       | 108.2                    | Data terminal ready           | Control to DCE             |
| 6         | CC                       | 107                      | Data set ready                | Control from DCE           |
| 7         | CA                       | 105                      | Request to send               | Control to DCE             |
| 8         | CB                       | 106                      | Clear to send                 | Control from DCE           |
| 9         | CE                       | 125                      | Ring indicator                | Control from DCE           |

ได้มีการเพิ่มประสิทธิภาพมากขึ้น ทำให้ในปัจจุบันมีไมโครคอนโทรลเลอร์ที่มีมาตรฐานมาจาก MCS-51 ของบริษัทอินเทลเป็นจำนวนมาก

ไมโครคอนโทรลเลอร์ MCS-51 ประกอบด้วยไมโครคอนโทรลเลอร์หลายรุ่น ซึ่งมีสถาปัตยกรรมพื้นฐานที่เหมือนกัน จะต่างเพียงขนาดของหน่วยความจำภายในและหน่วยทำงานภายในเท่านั้น ในการใช้งานสามารถเลือกใช้ได้ตามความต้องการและความเหมาะสม

### 2.6.1 คุณสมบัติของไมโครคอนโทรลเลอร์ MCS-51

1. หน่วยประมวลผลการขนาด 8 บิต
2. หน่วยความจำโปรแกรมภายใน ( Program Memory ) ขนาด 4 กิโลไบต์
3. หน่วยความจำข้อมูลภายใน (Data Memory) ขนาด 128 ไบต์
4. อ้า่งตำแหน่งของหน่วยความจำโปรแกรมได้ถึง 64 กิโลไบต์
5. อ้า่งตำแหน่งของหน่วยความจำข้อมูลได้ถึง 64 กิโลไบต์
6. หน่วยความจำโปรแกรมและข้อมูลที่อยู่ภายนอกชิป แยกจากกันอย่างละ 64 กิโลไบต์
7. มีพอร์ตอินพุต / เอาต์พุต แบบขนานจำนวน 4 พอร์ต (32 บิต) แยกกันอย่างอิสระ
8. มีวงจรนับ / จับเวลา ขนาด 16 บิต 2 ชุด ทำงานได้ 4 โหมด
9. มีพอร์ตการสื่อสารอนุกรม UART รับส่งข้อมูลได้ในเวลาเดียวกัน และสามารถเลือกรูปการส่งได้ 4 รูปแบบ

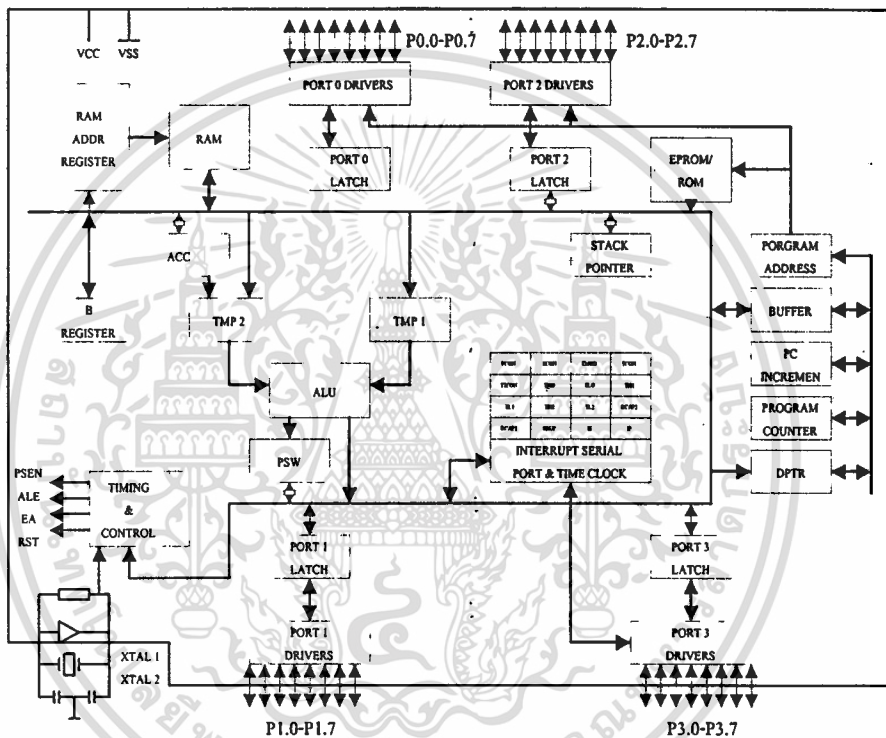
10. รับสัญญาณอินเตอร์รัปต์ได้ 6 แหล่งกระโดดไปทำงานตอบสนองได้ 5 ตำแหน่ง

เอกสารนี้เป็นเอกสารของบริษัทเอกชน ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ

11. มีวงจรลอจิกเกตภายใน เพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
12. นำข้อมูลมา AND ,OR หรือทำ Complement ได้ทั้งแบบ 8 บิต และ 1 บิต

## 2.6.2 โครงสร้างภายในของไมโครคอนโทรลเลอร์

ภายในของไมโครคอนโทรลเลอร์ จะประกอบขึ้นด้วยเกตต่างๆ เช่น AND , OR , NOT ซึ่งเกตเหล่านี้จะนำมาออกแบบให้มีหน้าที่การทำงานต่างๆ เช่น วงจรลอกรหัสคำสั่ง วงจรสร้างสัญญาณนาฬิกา เป็นต้น โครงสร้างภายในของไมโครคอนโทรลเลอร์ จะประกอบด้วยส่วนย่อยๆ ดังรูปที่ 2.28



รูปที่ 2.28 โครงสร้างภายในไมโครคอนโทรลเลอร์

ซึ่งโครงสร้างภายในไมโครคอนโทรลเลอร์ จะประกอบด้วย 3 ส่วนหลักๆดังนี้

1) CPU (Central Processing Unit) ส่วนนี้จะทำหน้าที่สร้างสัญญาณควบคุมในการติดต่อกับส่วนอื่นๆ เรียกว่าวงจรควบคุม (Control Unit) สัญญาณที่สร้างจากวงจรควบคุมได้แก่ สัญญาณสำหรับการติดต่อกับหน่วยความจำ , อุปกรณ์รับข้อมูลเข้าหรือส่งข้อมูลออก ซึ่งส่วนควบคุมการขัดจังหวะ และส่วนควบคุมบัสก็เป็นส่วนหนึ่งของวงจรควบคุมด้วย การสร้างสัญญาณจากวงจรควบคุมจาก CPU นี้จะทำการสร้างสัญญาณโดยการลอกรหัสจากคำสั่งที่มีการกำหนดไว้ และสัญญาณที่สร้างขึ้นมานี้จะอ้างอิงกับสัญญาณนาฬิกาที่สร้างจากวงจรออสซิลเลเตอร์เพื่อให้ทุกๆ ส่วนทำงานประสานกันอย่างถูกต้อง

2) หน่วยความจำ (Memory) มีไว้สำหรับจัดจำข้อมูล ซึ่งในการนำข้อมูลเข้าและออกจากหน่วยความจำ เราจำเป็นต้องรู้จักตำแหน่งของหน่วยความจำ (Address) ในการนำข้อมูลไปเก็บในหน่วยความจำ เรียกว่า การเขียนข้อมูล และการนำข้อมูลออกจากหน่วยความจำ เรียกว่า การอ่านข้อมูล ในไมโครคอนโทรลเลอร์ MCS-51 ข้อมูลในแต่ละตำแหน่งจะมีขนาด 8 บิต ดังนั้นแต่ละตำแหน่งของหน่วยความจำจะสามารถเก็บข้อมูลมีค่าได้ระหว่าง 00000000 (2) ถึง 11111111 (2) หรือ 00H ถึง 0FFH ในการติดต่อกับหน่วยความจำจะต้องมีสัญญาณ 3 กลุ่มคือ

1) ตำแหน่งที่ต้องการติดต่อกับหน่วยความจำ ซึ่ง MCS-51 สามารถติดต่อกับหน่วยความจำโปรแกรมและหน่วยความจำข้อมูลได้สูงสุดชนิดละ 65,536 ตำแหน่ง (64 kBytes) ดังนั้นการอ้างตำแหน่งของหน่วยความจำจะต้องใช้เส้นแสดงตำแหน่งในเลขฐาน 2 ทั้งหมด 16 เส้น

2) ข้อมูลที่อ่านหรือเขียนกับหน่วยความจำ ในตำแหน่งที่เราต้องการ

3) สัญญาณควบคุม ที่จะส่งไปยังหน่วยความจำ เพื่อบอกกับหน่วยความจำว่าต้องการอ่านหรือเขียนข้อมูล โดยวงจรลอจิกคำสั่งจะทำการสร้างสัญญาณควบคุมจากคำสั่งที่อ่านเข้ามาจากหน่วยความจำโปรแกรม

3) อุปกรณ์อินพุต / เอาต์พุต (Input / Output Device) เป็นส่วนที่ใช้ส่งข้อมูลเข้าหรือนำข้อมูลออกจาก MCS-51 ทำให้สามารถติดต่อกับอุปกรณ์ภายนอกได้ อุปกรณ์อินพุต / เอาต์พุต ได้แก่ 4 I/O Port, Timer / Counter 0, Timer Counter 1, Serial Port

### 2.6.3 หน้าที่หลักของภายนอกของไมโครคอนโทรลเลอร์ MCS-51

จากรูปแสดงลักษณะภายนอกของไมโครคอนโทรลเลอร์ MCS-51 โดยสามารถแบ่งได้เป็น 2 แบบ คือ แบบ Pin มี 40 ขา และแบบ Pad มี 44 ขา ซึ่งทั้ง 2 แบบ มีการทำงานที่เหมือนกัน ผู้ใช้สามารถเลือกใช้ได้ตามความต้องการ จากรูปที่ 2.29 เป็นโครงสร้างภายนอกของไมโครคอนโทรลเลอร์ 8051 ซึ่งในการควบคุมการทำงานแพคเกจนี้ใช้ ไมโครคอนโทรลเลอร์ในการควบคุมระบบต่างๆให้มีการจัดลำดับขั้นตอนการทำงาน ตามลักษณะของผังการทำงานของระบบ

VCC : (ขา 40) ต่อไฟเลี้ยง + 5V

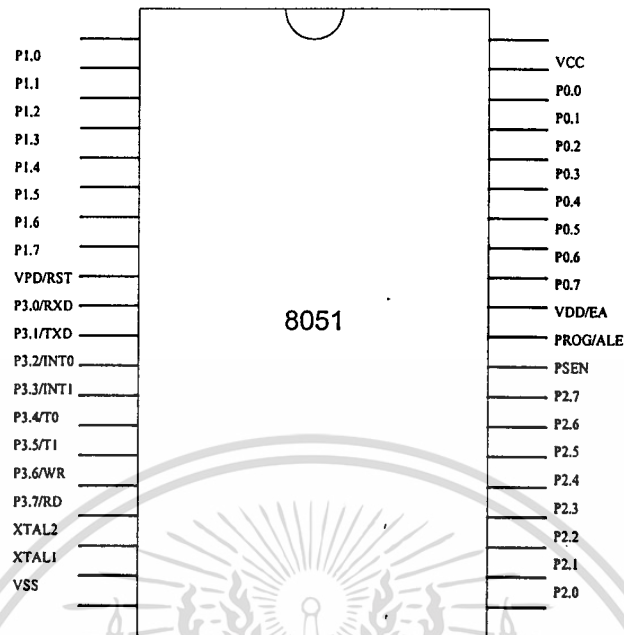
VSS : (ขา 20) ต่อลงกราวด์

Port 0 : (ขา 32 - 29) มีทั้งหมด 8 บิต คือ P0.0 - P0.7 ใช้งานเป็นอินพุต / เอาต์พุตพอร์ตทั่วไป ใช้เป็นตัวส่งแอดเดรสไบต์ต่ำ (A0-A7) และรับส่งข้อมูล (D0-D7) จากหน่วยความจำภายนอก

Port 1 : (ขา 1-8) มีทั้งหมด 8 บิต คือ P1.0-P1.7 ใช้งานเป็นอินพุต / เอาต์พุตพอร์ตทั่วไป

Port 2 : (ขา 21 - 28) มีทั้งหมด 8 บิต คือ P2.0 - P2.7 ใช้งานเป็นอินพุต / เอาต์พุตพอร์ตทั่วไป

และใช้เป็นตัวส่งแอดเดรสไบต์สูง (A8-A15) เพื่อใช้ติดต่อกับหน่วยความจำภายนอก  
เอกสารนี้เป็นเอกสารลิขสิทธิ์ของสถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.29 ลักษณะภายนอกของ MCS-51

Port 3 : (ขา 10-17) มีทั้งหมด 8 บิต คือ P3.0-P3.7 ใช้งานเป็นอินพุต / เอาต์พุตพอร์ตทั่วไป และใช้งานหน้าที่พิเศษดังนี้

P3.0/RXD (Serial Input Port) : ใ้รับข้อมูลแบบอนุกรม

P3.1/TXD (Serial Output Port) : ใ้ส่งข้อมูลแบบอนุกรม

P3.2/INT0 (External Interrupt) : ใ้เป็นอินพุตเพื่อรับสัญญาณขัดจังหวะจากภายนอก

P3.3/INT1 (External Interrupt) : ใ้เป็นอินพุตเพื่อรับสัญญาณขัดจังหวะจากภายนอก

P3.4/T0 (Timer/Counter 0 External Input) : ใ้เป็นวงจรรนับ/จับเวลา ชุดที่0

P3.5/T1 (Timer/Counter 0 External Input) : ใ้เป็นวงจรรนับ/จับเวลา ชุดที่1

P3.6/WR (External Data Memory White Strobe) : ควบคุมการเขียนข้อมูลไปยังหน่วยความจำภายนอก

P3.7/RD (External Data Memory Read Strobe) : ควบคุมการอ่านข้อมูลไปยังหน่วยความจำภายนอก

RST : (ขา 9) Reset ใ้สำหรับเช้ควงจรทุกอย่างภายในชิป เพื่อเริ่มต้นการทำงานใหม่ในการรีเซตต้องป้อน ลอจิก “1” นานอย่างน้อย 2 แมกซีนไซเคิล

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ (ข่า 30) Address Latch Enable เป็นขาส่งสัญญาณออกไปภายนอก เพื่อควบคุมไม่ให้การเกิดค่าแอดเดรสไบต์ต่ำจากพอร์ต 0 นื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใ้

PSEN (ขา 29) Program Strobe Enable เป็นขาส่งสัญญาณเพื่ออ่านคำสั่งจากหน่วยความจำโปรแกรมภายนอก เมื่อขา Active มีลอจิกเป็น “ 0 ” จะอ่านโปรแกรมจากหน่วยความจำโปรแกรมภายนอก และถ้าเป็นการอ่านหน่วยความจำโปรแกรมภายในขาจะไม่ Active

EA (ขา 19) External Access เป็นขาที่ใช้สำหรับเลือกให้ทำงานจากหน่วยความจำโปรแกรมภายในหรือหน่วยความจำภายนอกชิป เมื่อขา Active มีลอจิกเป็น “0” จะเป็นการทำงานตามคำสั่งในหน่วยความจำโปรแกรมภายนอก

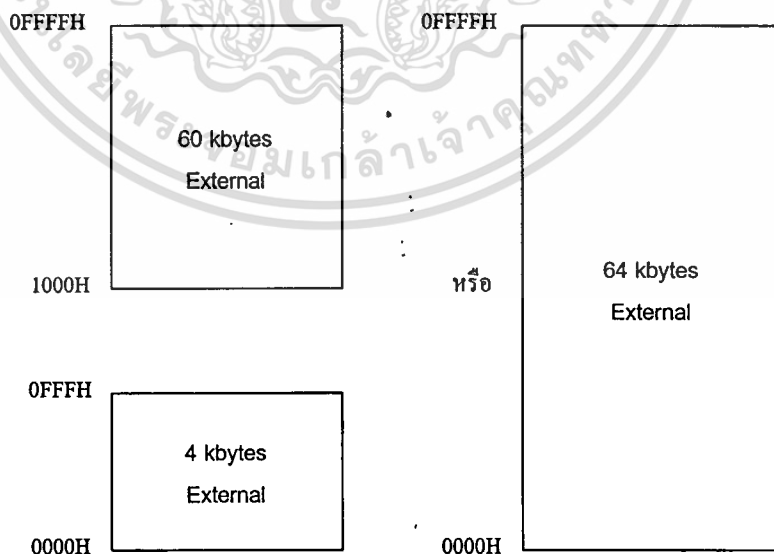
XTAL1 (ขา 19) ใช้ต่อคริสตัลภายนอก โดยเป็นอินพุตเข้าวงจรออสซิลเลเตอร์

XTAL1 (ขา 18) ใช้ต่อคริสตัลภายนอก โดยเป็นเอาต์พุตออกจากวงจรออสซิลเลเตอร์

#### 2.6.4 การจัดหน่วยความจำของ MCS-51

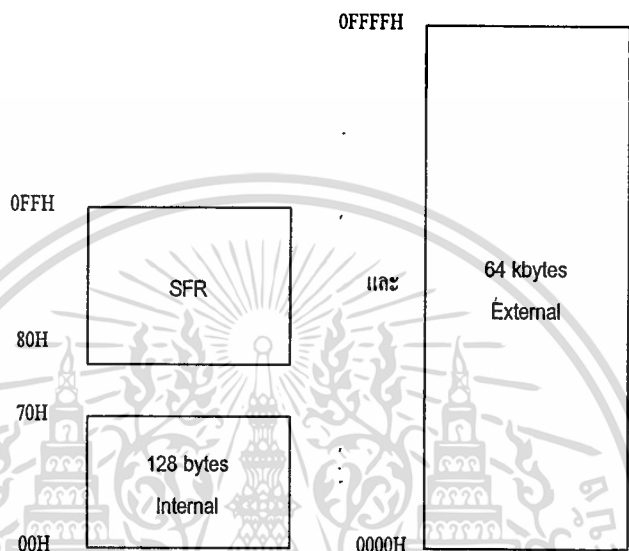
หน่วยความจำของ MCS-51 แบ่งออกเป็น 2 แบบตามลักษณะการใช้งาน ดังนี้

1) หน่วยความจำโปรแกรม (Program Memory) เป็นหน่วยความจำที่ใช้เก็บคำสั่งในรูปของภาษาเครื่องซึ่งต้องการให้ MCS-51 ทำงาน เมื่อ MCS-51 ทำงานก็จะอ่านข้อมูลที่เก็บในหน่วยความจำประเภทนี้ไปทำการถอดรหัสแล้ว สร้างสัญญาณควบคุมส่วนอื่นๆ ตามการทำงานของแต่ละคำสั่งนั้น หน่วยความจำแบบนี้เป็นแบบ ROM และผู้ใช้งานต้องเขียนข้อมูลในแต่ละตำแหน่งของหน่วยความจำเป็นรหัสภาษาเครื่องของ MCS-51 ตามลำดับการทำงานที่ต้องการ ส่วนที่เป็นหน่วยความจำโปรแกรมก็คือ ROM ขนาด 4 กิโลไบต์นั่นเอง ดังรูปที่ 2.30



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้ในกรณีฉุกเฉินเท่านั้น ไม่ควรนำเอกสารนี้ไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2) หน่วยความจำข้อมูล (Data Memory) เป็นหน่วยความจำที่ใช้เก็บข้อมูล สามารถอ่านและเขียนข้อมูลได้ ซึ่งหน่วยความจำภายในมีขนาดเพียง 128 ไบต์ ส่วนหน่วยความจำข้อมูลภายนอกชิป มีขนาด 64 กิโลไบต์ ดังรูปที่ 2.31 เป็นการจัดพื้นที่หน่วยความจำข้อมูล



รูปที่ 2.31 การจัดพื้นที่หน่วยความจำข้อมูล

2.7 การสร้างฐานเวลาให้แก่ MCS-51

ในระบบที่ใช้ไมโครคอนโทรลเลอร์เป็นตัวควบคุม ส่วนใหญ่ต้องใช้เวลาเป็นตัวกำหนดการทำงาน เช่น การกำหนดการทำงานให้เครื่องจักรเปิดหรือปิดในเวลาที่กำหนด การบันทึกเวลาการใช้งานเครื่องจักร ดังนั้นหากในระบบควบคุมที่ใช้ไมโครคอนโทรลเลอร์มีวงจรกำเนิดฐานเวลารวมอยู่ด้วย จะทำให้การออกแบบระบบและการเขียนโปรแกรมควบคุมสะดวกมากขึ้น

ในการสร้างฐานเวลาให้แก่ไมโครคอนโทรลเลอร์รวม2วิธี วิธีแรกจะใช้ชิปที่ทำหน้าที่สร้างฐานเวลาเพื่อส่งค่าเวลาปัจจุบันในขณะนี้ ให้แก่ไมโครคอนโทรลเลอร์ ส่วนวิธีที่สองจะใช้วงจรกำเนิดความถี่ที่มีการป้อนให้แก่เคาน์เตอร์ของไทม์เมอร์ 0 หรือไทม์เมอร์ 1 เพื่อให้ไทม์เมอร์ทั้งสองอินเตอร์รัปต์ซีพียูในช่วงเวลาที่กำหนดจากวงจรภายในได้

2.7.1 วิธีสร้างฐานเวลาจริงให้แก่ MCS-51 (ใช้ชิป RTC)

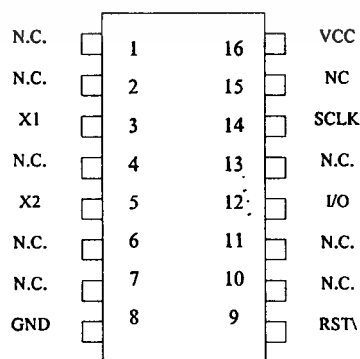
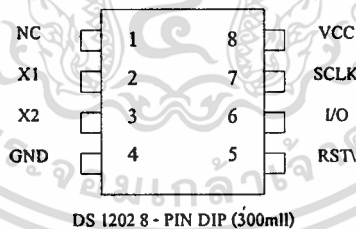
วิธีแรกจะใช้ชิปทำหน้าที่เป็นนาฬิกาที่สามารถส่งข้อมูลเวลาในขณะนี้ (ชิป RTC) ให้แก่ไมโครคอนโทรลเลอร์ได้ เวลาที่ได้จากวิธีนี้จะเป็นวิธีจริงๆ ที่เดินอยู่ตลอดเวลาอย่างเที่ยงตรง ชิปที่ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ใช้ส่วนใหญ่จะเป็นเบอร์ DS1202 ของบริษัท Dallas Semiconductor จากรูปที่ 2.32 เป็นลักษณะรูปร่างทั้งสองแบบของไอซีเบอร์ DS1202 ซึ่งจะมีเนื้อหาภายในดังต่อไปนี้

ชิป RTC จะมีอยู่ด้วยกันหลายชนิด บางชนิดสามารถอินเทอร์เฟซได้ไมโครคอนโทรลเลอร์ภายในช่วงเวลาที่กำหนดได้ ชิป RTC ที่ใช้จะทำได้เพียงแค่ให้ข้อมูลที่เป็นเวลาในขณะใดๆ แก่ไมโครคอนโทรลเลอร์เท่านั้น ไม่สามารถอินเทอร์เฟซซีพียูได้ การเลือกใช้งานชิป RTC ประเภทใดขึ้นอยู่กับความต้องการว่าต้องการความสามารถมากน้อยเพียงใด

ชิป RTC เบอร์ DS1202 เป็นชิปที่มีความเที่ยงตรงในการทำงานสูงมาก สามารถนำมาต่อร่วมกับระบบเพื่อบอกเวลาให้แก่ไมโครคอนโทรลเลอร์ได้สะดวก เพราะใช้จำนวนสายในการติดต่อระหว่างตัวชิปเองกับไมโครคอนโทรลเลอร์เพียง 3 เส้นเท่านั้น เนื่องจากชิป RTC จะใช้การติดต่อรับส่งข้อมูลแบบอนุกรม คุณสมบัติคร่าวๆของชิป RTC มีดังนี้

1. ทำหน้าที่นับวินาที นาฬิกา ชั่วโมง วันที่ของเดือน เดือน ปี รวมทั้งคำนวณปีอธิกสุรทินให้เองโดยอัตโนมัติ
2. มีหน่วยความจำขนาด 24 ไบต์สำหรับเก็บข้อมูลต่างๆไป ส่วนใหญ่ไว้เก็บข้อมูลที่ต้องการสำรองในกรณีที่ไม่มีพลังงานจ่ายให้แก่ระบบ เช่น รหัสผ่านที่เปลี่ยนค่าได้ เวลาที่ต้องการให้เครื่องจักรทำงาน ทำให้ไม่จำเป็นต้องสำรองหน่วยความจำทั้งระบบนั่นเอง
3. ใช้การติดต่อแบบอนุกรม จึงใช้จำนวนสายในการเชื่อมต่อกับระบบเพียง 3 เส้นเท่านั้น



DS 1202 16 - PIN DIP (300mll)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 2.32 ลักษณะรูปร่างทั้งสองแบบของไอซีเบอร์ DS1202

4. ใช้แรงดันไฟฟ้าเพียง 20 ถึง 5.5 โวลต์ และการใช้กระแสเพียง 300 นาโนแอมแปร์ที่ระดับ แรงดัน 2.0
5. การโอนย้ายข้อมูลสามารถกระทำได้ทั้งในแบบครั้งละ 1 ไบต์ (signal byte) หรือ ครั้งละหลายๆ ไบต์ ไม่ว่าจะเป็นการเขียนหรือการอ่านข้อมูล
6. ตัวชิปเองมีให้เลือกทั้งแบบ 8 PIN DIP หรือ PIN SOIC เพื่อใช้สำหรับแผ่นวงจรชนิด Surface Mount
7. ระดับสัญญาณ TTL Compatible ( $V_{cc} = 5$  โวลต์)
8. ช่วงอุณหภูมิในการใช้งานกว้างมาก ระหว่าง  $-40$  องศา  $\pm 88$  องศาเซลเซียส

### 2.7.2 รายละเอียดของชิป RTC

ชิป RTC เบอร์ DS1202 “Serial Time Keeper Chip” มี Real Time Clock/Calendar และ Static RAM ขนาด 24 ไบต์ ใช้สายเพียง 3 เส้นในการเชื่อมต่อกับไมโครคอนโทรลเลอร์ เพื่อรับข้อมูลเกี่ยวกับเวลา ข้อมูลที่ชิป RTC DS1202 มีให้ประกอบด้วย วินาที , นาที , ชั่วโมง , วันที่ , เดือน , ปี , วัน

วันที่ในวันสุดท้ายของเดือนจะถูกปรับโดยอัตโนมัติ สำหรับเดือนที่มีจำนวนวันน้อยกว่า 31 วัน และมีการคำนวณจำนวนวันของเดือนกุมภาพันธ์ในปีอธิกสุรทินให้เอง ข้อมูลที่ส่งให้แก่ไมโครคอนโทรลเลอร์สามารถเลือกรูปแบบได้ทั้งแบบ 24 ชั่วโมง (000-23.59 นาฬิกา) หรือแบบ 12 ชั่วโมง (000-12.00 นาฬิกา โดยมีข้อมูลเพิ่มเพื่อบอกให้ทราบว่า เป็นเวลาในช่วงกลางวันหรือกลางคืน)

การเชื่อมต่อชิป RTC DS1202 เข้ากับระบบมีความสะดวกมากเนื่องจากใช้จำนวนสายเพียง 3 เส้นเท่านั้น เพราะใช้การติดต่อแบบอนุกรมชนิด Synchronous Serial Communication ขาที่ต้องใช้ในการรับส่งข้อมูลทั้งสามคือ RST (Reset) , I/O (Data Line) , SCLK (Serial Clock)

เนื่องจากใช้ชิป RTC DS1202 มีนาฬิกาหรือเวลาที่เดินอยู่ตลอดเวลา รวมทั้งมีหน่วยความจำจำนวนหนึ่ง ดังนั้นในการติดต่อกับชิป RTC DS1202 ผู้ใช้สามารถเลือกข้อมูลจากนาฬิกาหรือจากหน่วยความจำในชิป (CLOCK/RAM) การรับส่งข้อมูลสามารถกระทำได้ทั้งแบบทีละไบต์ หรือรับส่งกันคราวละหลายๆไบต์ นอกจากนี้ RTC DS1202 ยังถูกออกแบบให้ใช้พลังงานน้อยมาก และสิ้นเปลืองพลังงานจากแบตเตอรี่น้อยที่สุดเพื่อความสะดวกในการสำรองพลังงาน โดยชิปตัวนี้สามารถเก็บรักษาข้อมูลในหน่วยความจำและเวลาที่เดินอยู่ตลอดเวลาได้ที่ใช้กำลังไฟฟ้าน้อยกว่า 1 ไมโครวัตต์ เราจะมาดูโครงสร้างของชิป RTC DS1202 ดังต่อไปนี้

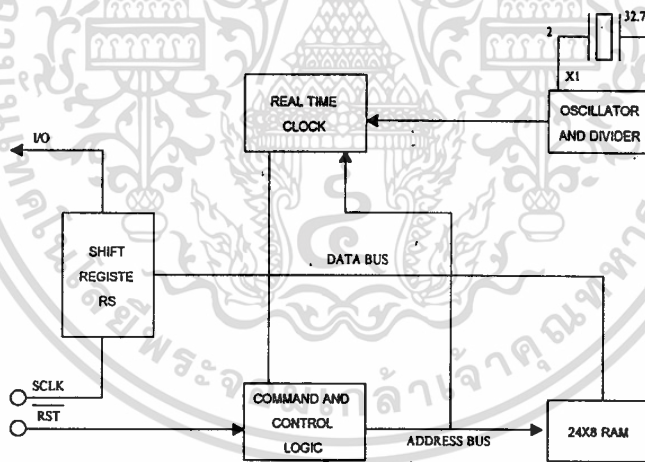
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 2.7.3 โครงสร้างภายในของ RTC DS1202

โครงสร้างภายในของ RTC DS1202 ประกอบไปด้วยส่วนที่สำคัญๆ ดังรูปที่ 2.33 จะเห็นว่าชิป RTC เบอร์นี้ประกอบไปด้วยส่วนที่สำคัญๆ ดังนี้

1. Shift Register
2. Control Logic
3. Oscillator
4. Real Time Clock
5. RAM

ในการรับหรือส่งข้อมูลใดๆ ให้แก่ RTC DS1202 ไมโครคอนโทรลเลอร์หรือไมโครโปรเซสเซอร์ที่ต้องการติดต่อดำเนินการจะต้องส่งข้อมูลที่เป็นคำสั่งควบคุมการติดต่อซึ่งมีขนาด 8 บิตเสียก่อน โดยเริ่มต้นด้วยการให้ขา RST มีสถานะเป็น 1 (อยู่ในช่วงการติดต่อ) จากนั้นส่งข้อมูลจำนวน 8 บิตเข้าไปไว้ใน Shift Register ของ RTC ข้อมูลขนาด 8 บิตจะประกอบด้วยคำสั่งในการควบคุมชิป



รูปที่ 2.33 โครงสร้างภายในของ RTC DS1202

RTC และตำแหน่งของหน่วยความจำที่ต้องการติดต่อ (Address/Command Byte) ในแต่ละครั้ง การรับข้อมูลจากไมโครโปรเซสเซอร์แต่ละบิตจะกระทำในช่วงขอบขาขึ้นของสัญญาณที่ขา SCLK (Serial Clock)

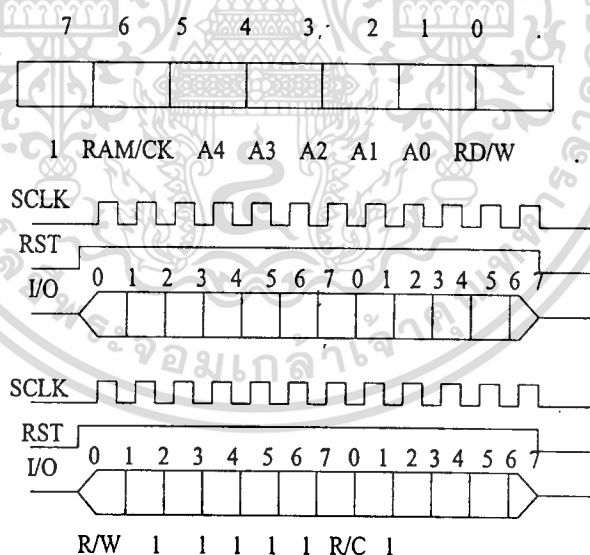
ภายในชิป RTC เบอร์ DS1202 ประกอบด้วยหน่วยความจำขนาด 24 ไบต์ และรีจิสเตอร์ ที่ทำหน้าที่เก็บเวลาของชิปในขณะปัจจุบันจำนวน 8 ตัว ดังรูปที่ 2.34 รีจิสเตอร์ 8 ตัวนี้ สามารถเข้าถึงไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ได้เสมือนเป็นหน่วยความจำตำแหน่งหนึ่ง ดังนั้นต่อไปเราจะมองว่าชิป RTC เบอร์นี้มีหน่วยความจำรวมทั้งสิ้น 32 ตำแหน่ง โดยประกอบขึ้นจากรีจิสเตอร์ 8 ตำแหน่งและหน่วยความจำ 24 ตำแหน่ง

ข้อมูลขนาด 8 บิตแรก (Address/Command Byte) จะระบุตำแหน่งหน่วยความจำที่ต้องการติดต่อ (ทั้งตำแหน่งของหน่วยความจำทั่วไปและตำแหน่งของรีจิสเตอร์สำหรับเก็บเวลา) และบอกว่าเป็นการเขียนหรืออ่านข้อมูลจากหน่วยความจำตำแหน่งนั้นๆ รวมทั้งระบุว่าการรับส่งข้อมูลเป็นแบบครั้งละ 1 ไบต์ หรือครั้งละหลายๆ ไบต์

หลังจากที่มีสัญญาณนาฬิกาเกิดขึ้น 8 ครั้ง (สัญญาณ SCLK) ในระหว่างการเริ่มเขียนข้อมูล 8 บิตแรกเข้าไปใน Shift Register สัญญาณนาฬิกาต่อไปที่จะเกิดขึ้นจะเป็นการนำข้อมูลออกจากชิป RTC สำหรับการอ่านข้อมูลหรือนำข้อมูลเข้าไปยังชิป RTC สำหรับการเขียนข้อมูล จำนวนของสัญญาณนาฬิกาทั้งหมดที่เกิดขึ้นในการติดต่อครั้งหนึ่งๆ จึงเท่ากับ  $8+8$  ในการส่งข้อมูลครั้งละ 1 ไบต์ Single Byte Mode หรือ 8 บวกมากที่สุด 192 ( $8 \times 24$ ) สำหรับการส่งข้อมูลครั้งละหลายๆ ไบต์ โครงสร้างของ Command Byte ดังแสดงในรูปที่ 2.34

#### 2.7.4 โครงสร้าง Command Byte



รูปที่ 2.34 โครงสร้างของ Command Byte

การรับหรือส่งข้อมูลระหว่างชิป RTC DS1202 เริ่มแรก ไมโครคอนโทรลเลอร์จะต้องส่งข้อมูลเพื่อกำหนดการทำงานให้แก่ RTC เสียก่อน ข้อมูลที่ RTC ได้รับเริ่มแรกนั้นจะมีขนาด 1 ไบต์ ซึ่งมีชื่อว่า Command Byte และเนื่องจากข้อมูลเป็นตัวกำหนดการทำงานของ RTC ดังนั้นแต่ละบิตในไบต์นี้จะมีควมหมายแตกต่างกันไปดังนี้

1. MSB (บิต 7) ต้องเป็น 1 เสมอ ถ้าเป็น 0 การทำงานต่อจากนี้จะถูกหยุดไว้หมด
2. บิต 6 ถ้าเป็น 0 จะระบุว่าต้องการติดต่อกับรีจิสเตอร์สำหรับเก็บเวลา (Clock/Calendar Register) ดังนั้นข้อมูลที่รับส่งกันจะเป็นเวลา หากบิตนี้มีค่าเป็น 1 จะระบุว่าต้องการติดต่อกับหน่วยความจำ
3. บิต 1 ถึง 5 เป็นตัวระบุตำแหน่งหน่วยความจำ (ทั้งหน่วยความจำที่ใช้เป็นรีจิสเตอร์สำหรับเก็บเวลาและหน่วยความจำทั่วไป) ที่ต้องการเข้าถึง ไม่ว่าจะเป็นการอ่านหรือเขียนข้อมูล ซึ่งควบคุมด้วยบิต 0
4. บิต 0 จะระบุว่าเป็นการหรืออ่านข้อมูล ถ้าเป็น 0 หมายถึงการเขียนข้อมูลลงไปในชิป หากเป็น 1 หมายถึงการอ่านข้อมูลจากชิป

ในการส่ง Command Byte ไปยังชิป RTC จะเริ่มต้นด้วยบิต 0 ก่อนเสมอ (LSB First)

Burst Mode หมายถึงการรับหรือส่งข้อมูลครั้งละหลายไบต์ในการติดต่อแต่ละครั้ง โดยสามารถกำหนดได้ว่าข้อมูลที่ต้องการติดต่อเป็นรีจิสเตอร์สำหรับเก็บเวลาหรือหน่วยความจำที่ใช้เก็บข้อมูล หากเป็นการรับส่งข้อมูลกับรีจิสเตอร์สำหรับเก็บเวลาก็จะรับส่งกันครั้งละ 8 ไบต์ หากเป็นการรับส่งกับหน่วยความจำจะรับส่งกันครั้งละ 24 ไบต์ ซึ่งการกำหนดให้รับส่งข้อมูลครั้งละหลายไบต์

ในการรับหรือส่งใน Burst Mode จะเริ่มต้นที่บิต 0 ของหน่วยความจำตำแหน่ง 0 ก่อนเสมอ ไม่ว่าจะเป็นรีจิสเตอร์สำหรับเก็บเวลาหรือหน่วยความจำสำหรับเก็บมูลทั่วไป

Burst Mode นี้มีไว้เพื่อความสะดวกในการรับ หรือส่งข้อมูลครั้งละจำนวนมากๆทำให้ไม่ต้องส่ง Command Byte หลายครั้งนั่นเอง

Write Protect Command Byte เวลาที่เดินอยู่ภายในรีจิสเตอร์ที่ทำหน้าที่เก็บเวลาและข้อมูลที่อยู่ในหน่วยความจำทั้ง 24 ตำแหน่ง สามารถป้องกันมิให้เขียนข้อมูลใดๆ ซ้อนลงไปได้ เพื่อป้องกันเหตุการณ์ที่เกิดขึ้น โดยไม่ได้ตั้งใจ ทั้งนี้โดยการควบคุมจาก Write Protect Register ซึ่งเป็นรีจิสเตอร์ ตำแหน่งที่ 7 ของรีจิสเตอร์ที่ทำหน้าที่เก็บเวลา โดยมี Write Protect Bit ซึ่งเป็นบิตที่ 7 ของรีจิสเตอร์ตัวนี้เป็นตัวกำหนดการทำงาน หาก Write Protect Bit เป็น 0 หมายถึงสามารถเขียนข้อมูลใดๆ ลงไปยังรีจิสเตอร์ที่เก็บเวลาหรือหน่วยความจำได้ หากบิตนี้เป็น 1 หมายถึงชิป RTC อยู่ในสถานะป้องกันการเขียนข้อมูล ดังนั้นก่อนการเขียนข้อมูลใดๆ ไปยังรีจิสเตอร์สำหรับเก็บเวลาหรือหน่วยความจำ Write Protect Bit ต้องเป็น 0 เสมอ โดยการให้ RST เป็น 1 (อยู่ในช่วงการติดต่อ) และโหลด Write Protection Command Byte (8EH) ตามด้วยข้อมูลที่มีค่า 00H หลังจากนั้น RST

ต้องกลับมาเป็นสถานะเป็น 0 ก่อนที่คำสั่งอื่นๆ จะเริ่มต้นทำงานได้

เอกสารนี้เป็นเอกสารที่เผยแพร่ในเชิงวิชาการเพื่อเป็นข้อมูลเท่านั้น ไม่สามารถนำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ส่วนในการบังคับให้ชิป RTC อยู่ในสถานะการเขียนข้อมูล ก็ต้องให้ RST เป็น 1 แล้ว โหลด Command ที่มีค่า 8EH ตามด้วยข้อมูล 80H (ให้ Write Protect Bit เป็น 1)

ในการทำงานแบบ Burst Mode เราไม่สามารถเขียนค่าใดๆ เข้าไปใน Write Protect Bit ได้

Reset and Clock Control การรับหรือส่งข้อมูลทั้งหมดจะต้องเริ่มโดยให้ขา RST มีสถานะเป็น 1 ก่อนเสมอ โดย RST มีหน้าที่หลักอยู่ 2 ประการดังนี้

1. RST ใช้ควบคุมการเขียนหรืออ่านข้อมูลใน Shift Register

2. RST ใช้เป็นสัญญาณในการหยุดทำงานใดๆ กับชิป RTC DS1202 โดยปกติการเขียนข้อมูลเข้าไปใน RTC DS1202 จะเกิดขึ้นในช่วงขอบขาขึ้นของสัญญาณที่ขา SCLK ส่วนการอ่านข้อมูลจาก RTC DS1202 จะเกิดขึ้นในช่วงขอบขาลงของสัญญาณที่ขา SCLK โดยในระหว่างการติดต่อ ขา RST ต้องมีสถานะเป็น 1 ตลอดเวลา หากขา RST มีสถานะเป็น 0 หมายถึงยกเลิกการติดต่อ หรือสิ้นสุดการติดต่อ

Data Input ในตอนเริ่มต้นติดต่อระหว่างชิป RTC DS1202 กับไมโครคอนโทรลเลอร์ไบต์แรกจะต้องเป็น Command Byte เสมอ หากใน Command Byte ระบุว่าเป็นการเขียนข้อมูลไปในชิป ข้อมูลจะถูกรับเข้ามาในช่วงขอบขาขึ้น (Rising Edge) ของ SCLK เท่านั้น โดยเริ่มต้นด้วยบิต 0 ก่อนเสมอ และหากเป็นคำสั่งให้รับส่งครั้งละ 1 ไบต์ เมื่อข้อมูลได้รับเข้ามานครบแล้ว สัญญาณ SCLK ที่รับได้เกินจะถูกทะเลยไป หากเป็นคำสั่งให้รับส่งแบบ Burst Mode ซึ่งรับส่งครั้งละ 24 ไบต์ ก็มีลักษณะเช่นเดียวกันคือ เมื่อรับข้อมูลครบ 24 ไบต์แล้ว สัญญาณ SCLK ที่รับได้เกินจะถูกทะเลยเช่นกัน

Data Output หลังจากรับ Command Byte แล้ว หากมีการระบุว่าเป็นการอ่านข้อมูลจากชิป RTC DS1202 ข้อมูลจะถูกส่งออกจากชิปสู่ภายนอกในช่วงขอบขาลง (Falling Edge) ของ SCLK หลังจากมีการรับ Command Byte เรียบร้อยแล้ว นั่นคือบิตแรกที่จะถูกส่งออกจาก RTC จะเกิดขึ้นในช่วงขอบขาลงของสัญญาณนาฬิกาอยู่ที่ 9 นั่นเอง (ต่อจาก Command Byte)

Clock/Calendar คือ รีจิสเตอร์ 8 ตัว โดยข้อมูลในรีจิสเตอร์เหล่านี้จะอยู่ในรูปของรหัส BCD เท่านั้น

Clock Halt Flag บิต 7 ของรีจิสเตอร์ที่เก็บค่าวินาที ซึ่งจะเป็นตัวบอกให้ชิป RTC DS1202 หยุดการทำงานของวงจรในส่วนของออสซิลเลเตอร์เมื่อบิตนี้มีค่าเป็น 1 ซึ่งเป็นผลทำให้นาฬิกาภายในชิปหยุดทำงานไปด้วย และจะบังคับให้ชิปอยู่ในสถานะ Low Power Standby Mode โดยใช้กระแสไม่เกิน 100 นาโนแอมแปร์ และเมื่อบิตนี้เป็น 0 อีกครั้ง วงจรออสซิลเลเตอร์ จะเริ่มทำงานต่อ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

AM-PM/12-24 Mode บิต 7 ของรีจิสเตอร์ที่เก็บค่าชั่วโมงถูกกำหนดให้เป็น 12/24 Hour Mode Select Bit นั่นคือ เป็นตัวเลือกที่จะให้รีจิสเตอร์นี้เก็บค่าชั่วโมงแบบ 12 ชั่วโมงหรือ 24 ชั่วโมง โดย

- 1. บิตที่ 7 เป็น 1 จะเป็นการเลือกให้เก็บค่าแบบ 12 ชั่วโมง โดยมีบิต 5 เป็นตัวบอกว่า เป็นช่วงกลางวันหรือกลางคืน (AM/PM Indicator) โดย จะหมายถึง PM และ หมายถึง AM
- 2. บิตที่ 7 เป็น 0 จะเป็นการเลือกให้เก็บค่าแบบ 24 ชั่วโมง และบิต 5 ที่แสดงหลักสิบตัวที่ 2 ของชั่วโมง (20-30)

Write Protect บิต 7 ของ Write Protect Register จะเป็น Write Protect Bit โดย 7 บิตแรกถูกบังคับให้เป็น 0 หหมด ทำให้อ่านได้ค่าเป็น 0 เสมอ

Clock/Calendar Burst Mode Command Byte ที่มีค่า BEH จะเป็นการระบุให้มีการเขียนข้อมูลในรีจิสเตอร์ที่เก็บเวลาในแบบ Burst Mode หาก Command Byte มีค่า BFH จะระบุให้มีการอ่านข้อมูลในรีจิสเตอร์นี้ในแบบ Burst Mode เช่นกัน

RAM Burst Mode Command Byte ที่มีค่า FEH จะเป็นการระบุให้มีการเขียนข้อมูลในหน่วยความจำแบบ Burst Mode หาก Command Byte มีค่า FFH จะระบุให้มีการอ่านข้อมูลในหน่วยความจำแบบ Burst Mode ทำนองเดียวกับใน Clock/Calendar Burst Mode Command Byte

2.7.5 การเลือกใช้คริสตอล

การเลือกใช้คริสตอล ชิป RTC DS1202 ใช้คริสตอลความถี่ 32.768 กิโลเฮิร์ตซ์ เป็นตัวกำหนดคาบเวลาในการทำงาน

คริสตอลความถี่ 32.768 กิโลเฮิร์ตซ์นี้สามารถต่อโดยตรงเข้ากับขา 2 และ 3 (x1,x2) ของ DS1202 ได้เลย โดยค่าอิมพีแดนซ์ของตัวคริสตอลเองควรเป็นค่าพาซิแตนซ์ (CL) ขนาด 6 พิโคฟาร์ด

ตารางที่ 2.10 หลักการทำงานของ Burst Mode

| BURST FUNCTION | DATA (BYTE) | SCLK |
|----------------|-------------|------|
| CLOCK          | 8           | 72   |
| RAM            | 24          | 200  |

ตารางที่ 2.11 ลักษณะข้อมูลการทำงานของรีจิสเตอร์

| รีจิสเตอร์ | ฟังก์ชัน                 | COMMAND ADDRESS (HEX) | เขียน = W<br>อ่าน = R | RANGE DATA (BCD) | รีจิสเตอร์ที่กำหนด |                |                  |        |              |   |   |   |  |
|------------|--------------------------|-----------------------|-----------------------|------------------|--------------------|----------------|------------------|--------|--------------|---|---|---|--|
|            |                          |                       |                       |                  | 7                  | 6              | 5                | 4      | 3            | 2 | 1 | 0 |  |
| 0          | วินาที                   | 80<br>81              | W<br>R                | 00-59            | CH                 | 10<br>วินาที   |                  | วินาที |              |   |   |   |  |
| 1          | นาฬิกา                   | 82<br>83              | W<br>R                | 00-59            | 0                  | 10 นาฬิกา      |                  | นาฬิกา |              |   |   |   |  |
| 2          | 12 ชั่วโมง<br>24 ชั่วโมง | 84<br>85              | W<br>R                | 01-12            | 12/<br>24          | 0              | AP 10<br>ชั่วโมง |        | ชั่วโมง      |   |   |   |  |
| 3          | วัน                      | 86<br>87              | W<br>R                | 00-23<br>01-31   | 0                  | 0              | 10 วัน           |        | วัน          |   |   |   |  |
| 4          | เดือน                    | 88<br>89              | W<br>R                | 01-12            | 0                  | 0              | 0                | 10     | เดือน<br>วัน |   |   |   |  |
| 5          | วัน                      | 8A<br>8B              | W<br>R                | 01-07            | 0                  | 0              | 0                | 0      | วัน          |   |   |   |  |
| 6          | ปี                       | 8C<br>8D              | W<br>R                | 00-99            | 10 ปี              |                |                  | ปี     |              |   |   |   |  |
| 7          | WRITE<br>PROTECT         | 8E<br>8F              | W<br>R                | 00-80            | WP                 | เป็น 0 ทั้งหมด |                  |        |              |   |   |   |  |

ตารางที่ 2.12 ลักษณะข้อมูลการทำงานของ RAM

|    |        |    |   |
|----|--------|----|---|
| 31 | CLOCK  | BE | W |
|    | BURST  | BF | R |
| 0  | RAM0   | CO | W |
|    |        | C1 | R |
|    |        | :  | : |
| 23 | RAM 23 | EE | W |
| 31 | RAM    | FE | W |
|    | BURST  | FF | R |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้เพื่อวัตถุประสงค์ทางการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 2.8 แพกเกจเรดิโอ

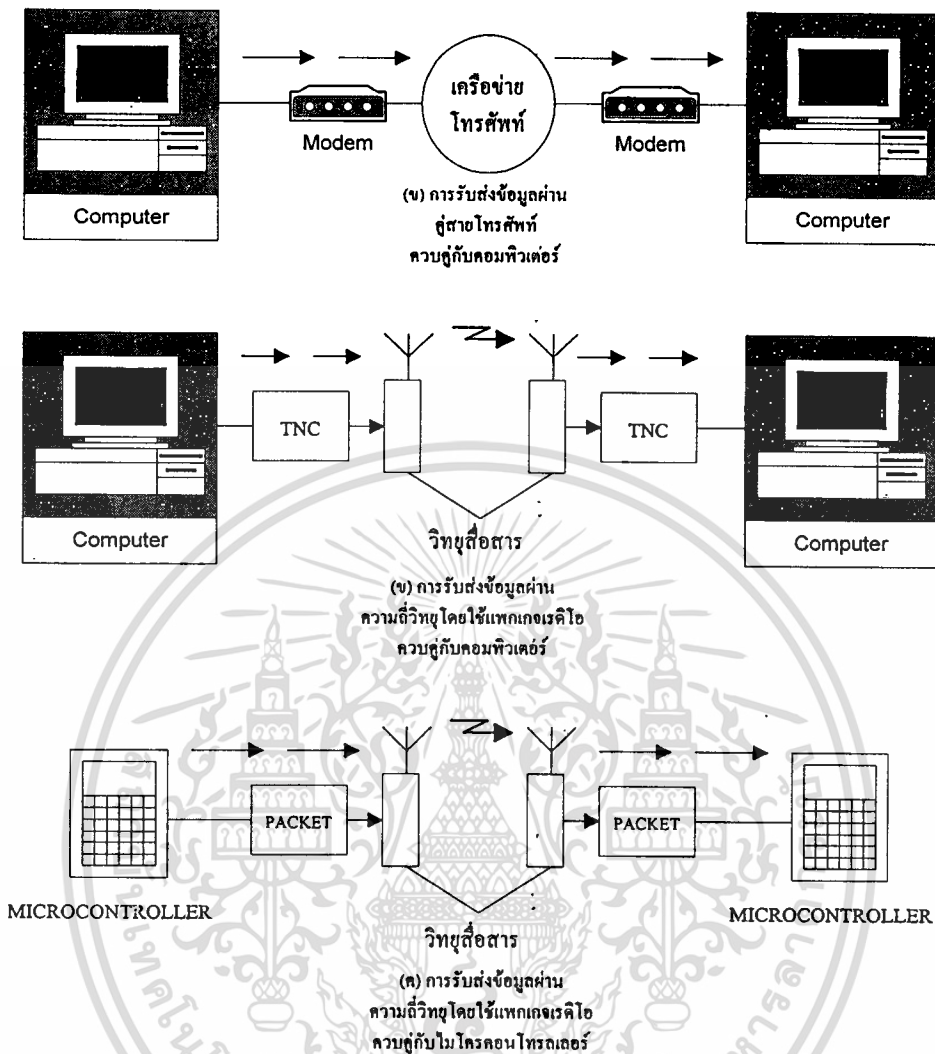
แพกเกจเรดิโอเป็นเทคโนโลยีการสื่อสารข้อมูลแบบดิจิทัล ซึ่งนิยมใช้งานในหมู่นักวิทยุสมัครเล่น (HAM) โดยอาศัยการสื่อสารข้อมูลคอมพิวเตอร์เป็นพื้นฐาน ซึ่งเปรียบได้กับการนำเครื่องคอมพิวเตอร์มาเชื่อมต่อกับโมเด็ม เพื่อใช้โมเด็มเป็นตัวกลางในการแปลงสัญญาณดิจิทัลจากเครื่องคอมพิวเตอร์มาเป็นสัญญาณแอนะล็อกก่อนที่จะส่งผ่านไปสู่อุปกรณ์สายโทรศัพท์ แต่สำหรับแพกเกจเรดิโอคู่สายโทรศัพท์จะถูกแทนที่ด้วยเครื่องรับส่งวิทยุ ซึ่งรับส่งสัญญาณคลื่นความถี่สูงในย่าน VHF หรือ UHF อุปกรณ์โมเด็มก็จะถูกแทนที่ด้วยอุปกรณ์เชื่อมต่อชนิดหนึ่ง มีชื่อเรียกว่า Terminal Node Controller หรือ TNC ซึ่งการสื่อสารแบบแพกเกจเรดิโอเป็นการนำข้อมูลไปนาริจากเครื่องคอมพิวเตอร์ มาทำการมอดูเลตก่อนที่จะส่งไปออกอากาศผ่านทางเครื่องรับส่งวิทยุไปสู่สถานีรับปลายทาง ดังรูปที่ 2.31 เปรียบเทียบการใช้งานแพกเกจเรดิโอกับการใช้งานโมเด็มรับส่งข้อมูล โดยข้อมูลจากเครื่องคอมพิวเตอร์จะถูกนำมาแบ่งเป็นกลุ่มอาจจะเป็นกลุ่มละ 100 ไบต์ เป็นต้น ก่อนที่จะส่งออกอากาศเราเรียกกลุ่มข้อมูลเหล่านี้ว่า แพกเกจ

### 2.8.1 ประวัติความเป็นมาของการสื่อสารแพกเกจเรดิโอ

เทคโนโลยีของเครือข่ายการสื่อสารข้อมูลแบบแพกเกจ ได้ถูกพัฒนาขึ้นมาใช้งานในช่วงกลางปี พ.ศ. 2503 และถูกกำหนดให้ใช้งานในเครือข่าย ARPANET (เครือข่ายอินเทอร์เน็ตในปัจจุบัน) ในปีพ.ศ. 2512 ต่อมาในปี พ.ศ. 2513 ทีมมหาวิทยาลัยฮาวายได้มีการพัฒนาเครือข่าย Alohanet ซึ่งนับได้ว่าเป็นโครงการแรก ที่ได้สร้างแพกเกจเรดิโอที่มีขนาดใหญ่มาก การนำแพกเกจเรดิโอไปใช้งานในวงการวิทยุสมัครเล่นได้เริ่มจัดตั้งขึ้นที่นครมอนทรีออล ประเทศแคนาดาในวันที่ 31 พฤษภาคม พ.ศ. 2521 หลังจากนั้นไม่นานก็ได้มีการดำเนินการสร้างอุปกรณ์ TNC ซึ่งเป็นอุปกรณ์เชื่อมต่อระหว่างเครื่องมินิเทอร์มินอลและวิทยุสื่อสารตามขึ้นมา โดยกลุ่มสื่อสารดิจิทัลสมัครเล่นแห่งนครแวนคูเวอร์ (Vancouver Communication Group - VADCCG) ในปี พ.ศ. 2523

มาตรฐาน TNC ได้รับการพัฒนามาขึ้นนับตั้งแต่ได้มีการจัดสัมมนา โดยสมาคมคอมพิวเตอร์ของ IEEE ในปี พ.ศ. 2524 โดยหลังจากการสัมมนาประมาณ 1 สัปดาห์ได้มีผู้ร่วมสัมมนาจำนวน 6 หน่วยงานร่วมกันจัดอภิปรายถึงความเป็นไปได้ในการจัดสร้างและพัฒนาอุปกรณ์ TNC สำหรับใช้งานในกิจการวิทยุสมัครเล่นด้วยราคาต่อเครื่องที่ไม่แพงมากนัก ซึ่งบริษัท Tucson Amateur Packet Radio (TAPR) เป็นผู้รับโครงการนี้ไปดำเนินการ ในวันที่ 26 มิถุนายน พ.ศ. 2525 ไนน์ จอห์น และเคน คอนเนอร์ส ก็สามารถดำเนินการสร้างเครื่องต้นแบบที่สามารถติดต่อกับ TAPR ได้สำเร็จ โครงการได้ดำเนินต่อไปจากเครื่องต้นแบบได้ถูกพัฒนามาเป็น TNC-1 และเป็น TNC-2 ซึ่งเป็นพื้นฐานของอุปกรณ์ TNC ที่ใช้กันทั่วโลกในที่สุด

เอกสารนี้เป็นเอกสารที่งานปฐมนิเทศน์เพื่อใช้ในการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.35 เปรียบเทียบการใช้งานแพคเกจเรดิโอกับการใช้งานโมเด็มรับส่งข้อมูล

## 2.8.2 การส่งข้อมูลแบบแพคเกจเรดิโอ

การรับส่งข้อมูลเป็นแพคเกจมีข้อดีเหนือกว่าการรับส่งข้อมูลแบบดิจิทัลแบบอื่นๆ อยู่ 3 ประการ คือ ความโปร่งใสของข้อมูลที่ส่งผ่านระหว่างอุปกรณ์ต้นทางและปลายทาง การควบคุมความผิดพลาดของข้อมูลและการควบคุมการรับส่งแบบอัตโนมัติ

การทำงานของสถานีรับส่งแพคเกจเรดิโอจะเป็นลักษณะที่โปร่งใสต่อผู้ใช้งาน ก็คือว่าผู้ใช้งานเพียงแค่เชื่อมต่ออุปกรณ์ต่างๆ เข้าด้วยกัน พิมพ์หรือเตรียมข้อความที่ต้องการจะส่ง ข้อความนั้นจะถูกส่งไปโดยอัตโนมัติไม่จำเป็นที่ผู้ใช้งานจะต้องมาเป็นธุระในการควบคุมกระบวนการเปลี่ยนแปลงรูปแบบข้อมูลให้เป็นแพคเกจ หรือควบคุมลำดับการรับส่ง ซึ่งอุปกรณ์ TNC จะทำหน้าที่แบ่งเอกสารเป็นเอกสารท่อนสั้นไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นับว่าเหมาะไปใช้ประโยชน์ด้านการค้า ข้อมูลที่ต้องการจะส่งออกเป็นข้อมูลแพคเกจย่อยๆ ติดต่อกับเครื่องรับวิทยุ และส่งแพคเกจนั้น โดยไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตีแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อัตโนมัติ ในด้านสถานีรับข้อมูลอุปกรณ์ TNC จะทำหน้าที่ถอดรหัสข้อมูลจากแพคเกจที่ได้รับ ตรวจสอบความผิดพลาดของข้อมูลและแสดงข้อความที่ได้รับ โดยผ่านเครื่องคอมพิวเตอร์หรือ โดยผ่านตัวไมโครคอนโทรลเลอร์ ที่เชื่อมต่ออยู่ปลายทางในการรับข้อมูลแสดงผลจากการส่งข้อความสู่ผู้รับปลายทาง

การสื่อสารแบบแพคเกจเรดิโอเป็นการสื่อสารที่ปราศจากความผิดพลาดของข้อมูล หรือ หากจะมีก็อยู่ในระดับต่ำมาก ๆ ทั้งนี้เนื่องจากการกำหนดกระบวนการวิธีการตรวจสอบความผิดพลาดของข้อมูลในอุปกรณ์ TNC โดยที่หากสถานีวิทยุได้รับแพคเกจข้อมูลมาจากสถานีส่งผ่านทาง จะมีการตรวจหาความผิดพลาดของข้อมูล และแสดงข้อความที่ได้รับผ่านทางเครื่องคอมพิวเตอร์หรือตัวไมโครคอนโทรลเลอร์แสดงที่จอ LCD ในกรณีที่ข้อมูลได้รับไม่มีความผิดปกติใดๆ นอกจากนั้นยังสามารถนำอุปกรณ์ TNC มาใช้งานเป็นสถานีทวนสัญญาณสำหรับการสื่อสารแบบแพคเกจเรดิโอได้อีกประการหนึ่ง ซึ่งบางครั้งเรียกว่า Digipeater นับเป็นการเพิ่มระยะทางในการติดต่อสื่อสารข้อมูลให้กว้างไกลขึ้น

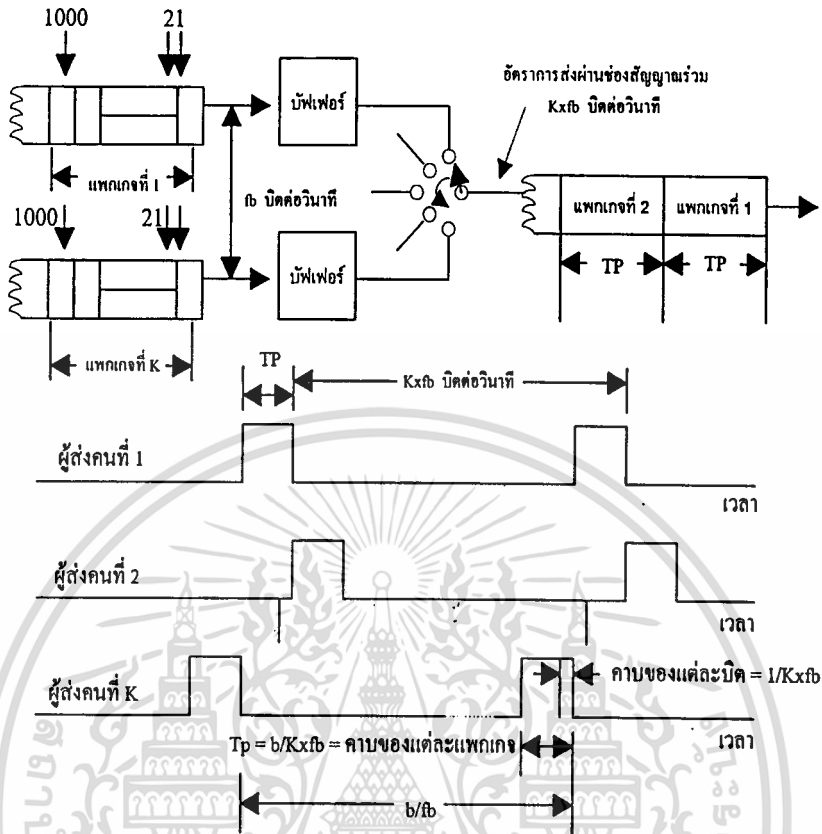
ผู้ใช้งานสามารถทำการติดต่อสื่อสารไปยังอุปกรณ์ TNC ของผู้อื่นได้ตลอดเวลาตามที่ต้องการ อุปกรณ์ TNC บางรุ่นสามารถรองรับการฝากข้อความไว้ในเครื่องได้ด้วย ดังนั้นผู้ใช้บริการรายอื่นๆ จึงสามารถฝากข่าวสารผ่านมาเก็บไว้ใน TNC ได้ ในกรณีที่เจ้าของเครื่องไม่อยู่บ้าน ข้อดีอีกประการหนึ่งของแพคเกจเรดิโอคือ ประสิทธิภาพในการใช้งานความถี่ร่วมกัน ทั้งนี้เนื่องจากระบบการส่งข้อมูลเป็นแบบแพคเกจซึ่งส่งกันเป็นช่วงๆ ดังนั้นผู้ใช้งานหลายๆรายจึงสามารถใช้ความถี่วิทยุเดียวกันเป็นศูนย์กลางในการรับส่งข้อมูลได้

### 2.8.3 เทคนิคการควบคุมการรับส่งข้อมูล

#### 1) การรับส่งข้อมูลแบบ TDMA (Time Division Multiple Access)

สำหรับวิธีการนี้จะเป็นลักษณะที่ผู้ใช้งานแต่ละคนผลัดกันส่งข้อมูล ดังรูปที่ 2.36 ซึ่งถ้ามีผู้ทำการส่งข้อมูลออกอากาศในความถี่เดียวกันอยู่  $K$  คน แต่ละคนถูกกำหนดให้ส่งข้อมูลออกมาเป็นแพคเกจๆ ละ 1,000 บิต ข้อมูลที่ส่งออกมาจะถูกเก็บไว้ในบัฟเฟอร์ของแต่ละช่องข้อมูล จนกว่าสวิตช์ตัดต่อ SW1 ซึ่งวิ่งวนจากช่องข้อมูล 1 ถึงช่องที่  $K$  เคลื่อนมาถึงช่องข้อมูลดังกล่าว ข้อมูลที่เก็บไว้ในบัฟเฟอร์จะถูกส่งผ่านออกไปยังช่องสัญญาณรวม และบัฟเฟอร์จะถูกทำให้ว่างเพื่อจะทำการรองรับแพคเกจใหม่ หากผู้ใช้งานแต่ละคนส่งข้อมูลออกมาด้วยอัตรา  $b$  บิตต่อวินาทีเท่าๆ กันจะพบว่าอัตราการส่งข้อมูลนั้นมีขีดจำกัด ลักษณะของการส่งข้อมูลนั้นจะขึ้นอยู่กับพื้นที่ของการใช้ส่งด้วยว่ามีการรบกวนของสัญญาณมากน้อยเพียงใด ซึ่งผลกระทบที่เกิดขึ้นต้องทำการแก้ไขและหลีกเลี่ยง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.36 การส่งแพ็กเกจโดยใช้เทคนิค TDMA

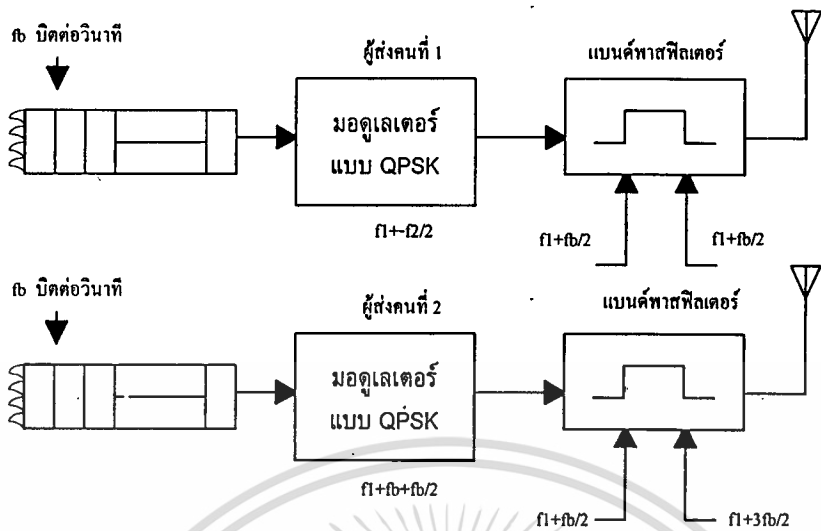
การส่งผ่านข้อมูลผ่านช่องสัญญาณร่วมมีค่าเป็น  $Kfb$  บิตต่อวินาที ซึ่งหากพิจารณาถึงข้อมูลที่วิ่งผ่านช่องสัญญาณร่วม จะเห็นว่าการส่งแพ็กเกจของช่องสัญญาณ 1 ตามด้วยช่องสัญญาณ 2 ไปจนถึงช่องสัญญาณ K แล้ววกกลับมาเป็นช่องสัญญาณ 1 อีกจะเป็นเช่นนี้เรื่อยไป

## 2) การรับส่งข้อมูลแบบ FDMA (Frequency Division Multiple Access)

การส่งแบบ FDMA จะแตกต่างกับแบบ TDMA ตรงที่ผู้ส่งข้อมูลแบบ TDMA จะผลัดกันส่งข้อมูลแต่แบบ FDMA อนุญาตให้แต่ละคนทำการส่งข้อมูลได้ตลอดเวลา โดยข้อมูลของแต่ละคนจะถูกส่งไปคนละความถี่ซึ่งมีแบนด์วิดท์  $B_u$  ภายในขอบเขตแบนด์วิดท์รวม  $B$  ซึ่งแท้จริงแล้วอัตราการส่งข้อมูลแบบ FDMA จะมีค่าเท่ากับแบบ TDMA คือ  $Kfb$  บิตต่อวินาที

รูปที่ 2.37 ลักษณะของการส่งข้อมูลแบบ FDMA ข้อมูลจากแต่ละช่องจะถูกส่งไปยังวงจรมอดูเลตแบบ QPSK (Quadrature Phase Shift Keying) โดยความถี่พาหะของแต่ละช่องสัญญาณจะถูกกำหนดให้ห่างกันเท่ากับ  $fb$  และแต่ละช่องข้อมูลใช้แบนด์วิดท์  $fb$  แล้วจึงทำการส่งออกอากาศที่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับอาจารย์งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
เครื่องรับแต่ละช่องจะถูกจูนให้รับคลื่นความถี่ที่ตรงกันกับช่องสัญญาณ และมีแบนด์วิดท์  $fb$  เช่น  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.37 การส่งแพ็คเกจโดยใช้เทคนิค FDMA

เดียวกัน ซึ่งข้อดีของการส่งแบบ FDMA ก็คือ การที่ผู้ส่งไม่ต้องเสียเวลารอในการส่ง แต่ละช่องสามารถส่งข้อมูลได้ตามอิสระ อีกทั้งยังเป็นการกำจัดบัพเฟอร์ออกไป ซึ่งบัพเฟอร์เป็นหน่วยความจำที่มีราคาสูง ทำให้ลดต้นทุนลงได้

3) การรับส่งข้อมูลแบบ ALOHA

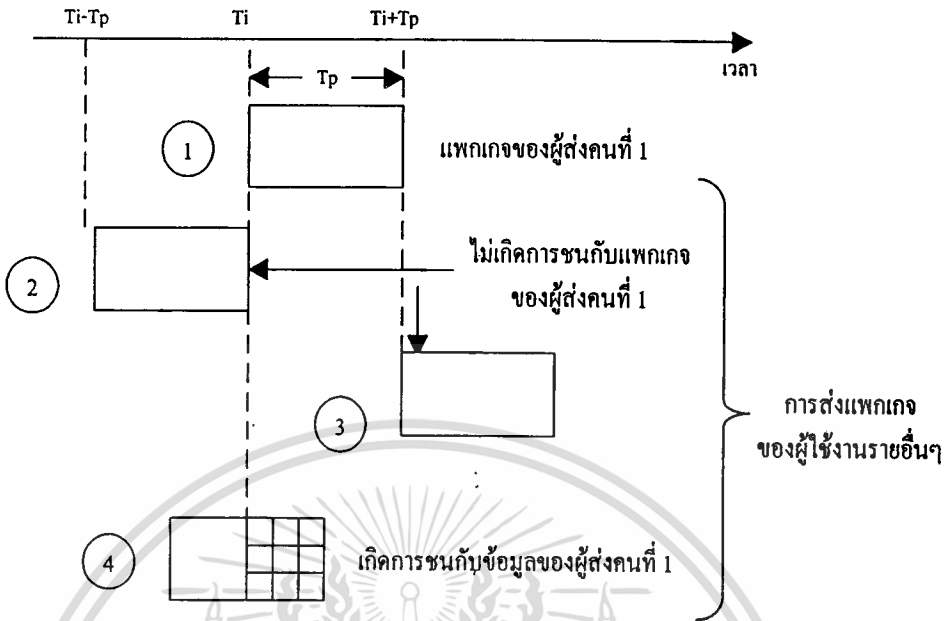
ในระบบ ALOHA จะมีช่องสัญญาณเดียว คือช่องสัญญาณร่วม ซึ่งโดยปกติจะว่างถ้าไม่มีการส่งข้อมูลจากช่องข้อมูลใดๆ แต่หากมีช่องข้อมูลใดต้องการส่งก็สามารถทำการส่งได้ทันที ด้วยเหตุนี้จึงทำให้แบนด์วิดท์ของช่องสัญญาณร่วมมีความกว้างเพียง 2b ซึ่งเท่ากับอัตราการส่งข้อมูลจากผู้ส่งรายเดียว จะเห็นได้ว่าผู้ส่งไม่ต้องรอส่งข้อมูลดังเช่นในรูปแบบ TDMA

ปัญหาที่จะเกิดขึ้นในการส่งข้อมูลจากผู้ใช้งานหลายๆ ช่องพร้อมกัน กรณีนี้เครื่องรับจะรับแพ็คเกจหลายแพ็คเกจในเวลาเดียวกัน โดยไม่สามารถแยกแยะแพ็คเกจเหล่านั้นออกจากกันได้ เรียกว่าเกิด “การชน” (Collision) ดังรูปที่ 2.38 ซึ่งความน่าจะเป็นที่ข้อมูลจะเกิดการชนกัน ขึ้นอยู่กับพฤติกรรมกรรมการส่งข้อมูลของผู้ใช้ช่องสัญญาณนั่นเอง

4) การรับส่งข้อมูลแบบ CSMA (Carrier Sense Multiple Access)

การส่งข้อมูลแบบ CSMA ผู้ใช้งานแต่ละคนจะทำการปรับเครื่องรับของตนไปยังความถี่เฉพาะเดียวกันคือ fc ก่อนที่จะมีการส่งข้อมูลออกมา ผู้ใช้แต่ละรายจะทำการตรวจสอบการใช้งานช่องสื่อสาร เพื่อดูว่ามีผู้ใ้รายใดกำลังทำการส่งข้อมูลอยู่หรือไม่ หากไม่มีสถานงานใดส่งข้อมูล จะมีผู้ใ้รายหนึ่งเริ่มทำการส่งแพ็คเกจทันที ในขั้นแรกนี้จะไม่คิดเวลาประวิงที่ใช้ในกาเดินทางของแพ็คเกจจากผู้ส่งไปยังผู้รับ

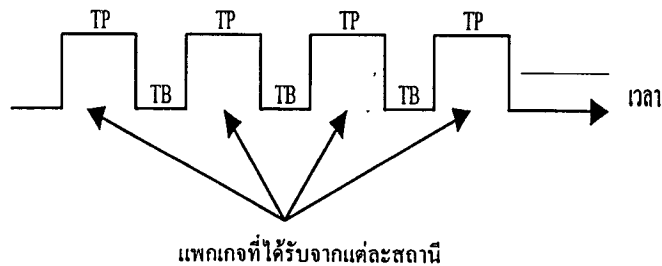
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.38 ลักษณะการเกิดการชนกันของแพคเกจระหว่างผู้ใช้หลายคน

หากมีการกำหนดค่าอัตราการส่งข้อมูล  $S$  ของการรับส่งข้อมูล CSMA ซึ่งมีรูปแบบของการส่งข้อมูลที่ถูกรับ ณ เครื่องรับดังรูปที่ 2.39 พฤติกรรมการรับข้อมูลของสถานีงานหนึ่งๆ ในกรณีของ CSMA

สำหรับการรับส่งแบบ CSMA ผู้ใช้งานแต่ละแห่งมิได้ทำการตรวจสอบการใช้งานช่องสื่อสารข้อมูลอยู่ตลอดเวลา แต่การใช้งานช่องสื่อสารของผู้ใช้งานจะเป็นไปในทันทีที่ต้องการใช้ ซึ่งบางครั้งอาจพบว่าช่องสื่อสารกำลังถูกใช้งานอยู่ ในกรณีนี้ผู้ใช้บริการจะทำการรอด้วยช่วงเวลาหนึ่ง ซึ่งเป็นค่าสุ่ม ก่อนที่จะทำการส่งข้อมูลสู่ช่องสื่อสารใหม่อีกครั้ง



รูปที่ 2.39 พฤติกรรมการรับข้อมูลของสถานีงานหนึ่งๆ ในกรณีของ CSMA

เอกสารนี้เป็นเอกสารตัวอย่าง ไม่สามารถนำมาใช้เพื่อวัตถุประสงค์อื่นใดโดยไม่ได้รับอนุญาตจากเจ้าของเอกสาร  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทที่ 3

### การออกแบบ การสร้าง และการทำงาน

#### 3.1 กล่าวนำ

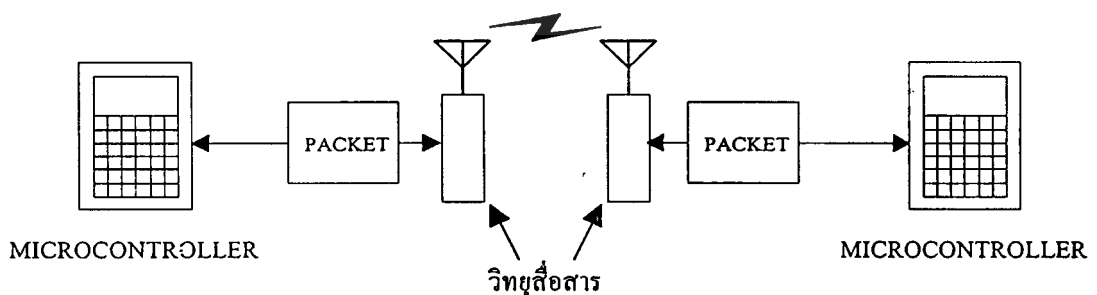
การออกแบบระบบวิทยุแพคเกจ ควบคุมโดยไมโครคอนโทรลเลอร์ สำหรับวิทยุสมัครเล่น นี้ ในการทำงานของระบบสามารถที่จะส่งข้อมูลไปกับระบบวิทยุสื่อสารได้ เพียงแค่ทำการเพิ่มวงจรแพคเกจเรดิโอ โดยใช้หลักการของเอฟเอสเคโมเด็มสำหรับการส่งสัญญาณดิจิทัลในช่องสัญญาณสื่อสาร ซึ่งสัญญาณดิจิทัลที่นำที่นำมามอดูเลตแบบเอฟเอสเค แล้วนำสัญญาณที่มอดูเลตได้ส่งไปพร้อมกับความถี่ในช่องสัญญาณวิทยุสื่อสาร

ส่วนทางด้านรับก็จะใช้วิทยุสื่อสารมาต่อร่วมกับวงจรแพคเกจเรดิโอมาทำการดีเทคสัญญาณดิจิทัลออกจากช่องสัญญาณสื่อสาร เพื่อนำไปถอดรหัสสัญญาณดิจิทัลอีกครั้งหนึ่งแล้วนำรหัสสัญญาณที่ได้ไปแปลงเป็นสัญญาณข้อมูลเพื่อแสดงผลทางหน้าจอต่อไป

ระบบการส่งข้อมูลและการรับข้อมูล ในช่องสัญญาณระบบวิทยุสื่อสาร ทั้งเครื่องส่งและเครื่องรับจะต่อร่วมกับชุดคีย์บอร์ดและจอแสดงผล เพื่อใช้เป็นอุปกรณ์ส่งข้อมูลและแสดงผลข้อมูล ซึ่งในโครงงานนี้จะประกอบด้วย 2 ระบบใหญ่ๆ คือ

1. ระบบส่งข้อมูล
2. ระบบรับข้อมูล

ซึ่งโครงสร้างการทำงานของระบบการส่งข้อมูลและรับข้อมูลจะแสดง ได้ดังรูปที่ 3.1



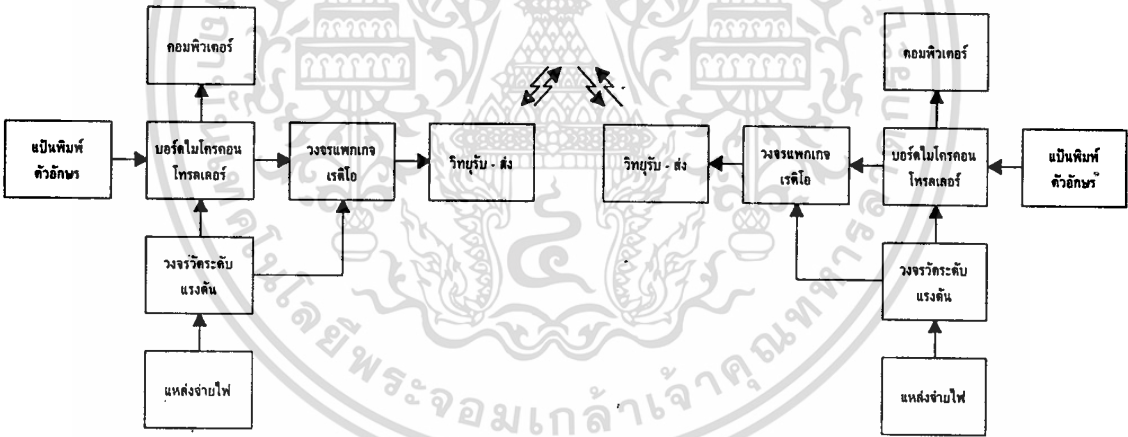
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น รูปที่ 3.1 ผังของระบบการรับส่งข้อมูลในช่องสัญญาณวิทยุสื่อสาร ครั้งที่มีการนำไปใช้

3.2 หลักการทำงานของวงจรในระบบรับส่งข้อมูล

ระบบหลักการทำงานของวงจรในระบบรับส่งข้อมูล ของปริญญานิพนธ์นี้มีส่วนประกอบของระบบรับส่งข้อมูลในส่วนต่างๆ ดังนี้

- 1. วงจรแพกเกจเรดิโอ
- 2. วงจรวัดระดับแรงดัน
- 3. บอร์ดไมโครคอนโทรลเลอร์
- 4. เป็นพิมพ์ตัวอักษร
- 5. เครื่องรับส่งวิทยุ

ในการทำงานของระบบรับส่งข้อมูลในช่องสัญญาณวิทยุสื่อสารนั้น จะมีส่วนประกอบของวงจรและส่วนต่างๆ ในการรับส่งข้อมูลข่าวสารระหว่างสถานีส่งและสถานีรับดังที่ได้กล่าวมาข้างต้น และสามารถเขียนบล็อกไดอะแกรมการทำงานของระบบการรับส่งข้อมูล ในช่องสัญญาณวิทยุสื่อสารได้ดังรูปที่ 3.2



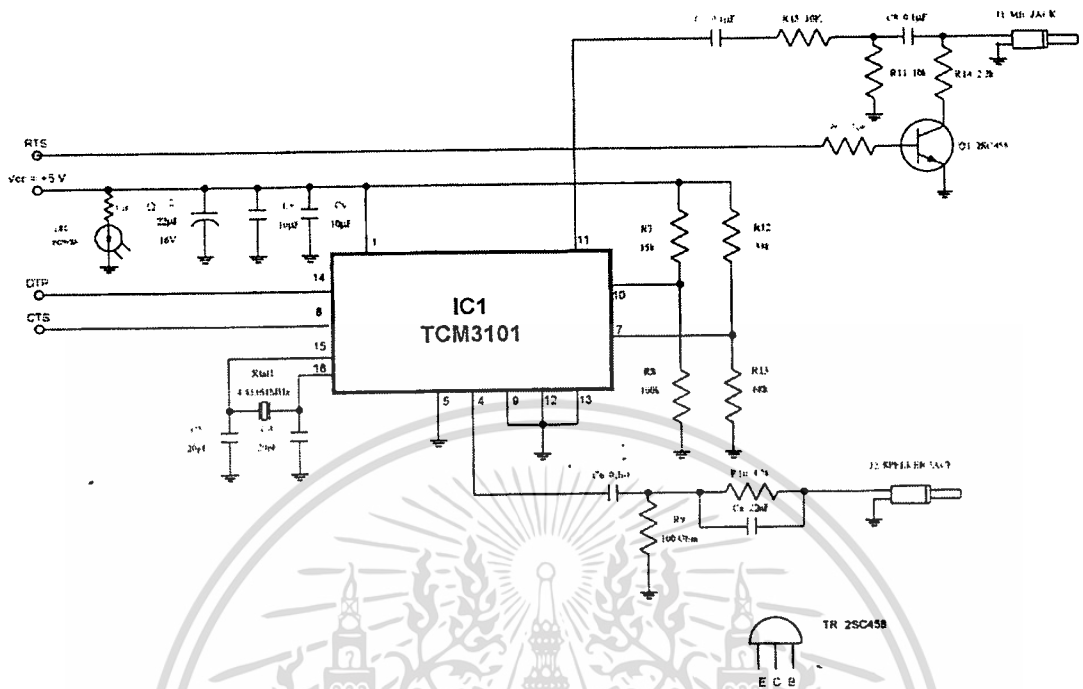
รูปที่ 3.2 แผนผังการทำงานของระบบการรับส่งข้อมูลในช่องสัญญาณวิทยุสื่อสาร

3.3 หลักการออกแบบ

3.3.1 วงจรแพกเกจเรดิโอ

วงจรแพกเกจเรดิโอเป็นรูปแบบหนึ่งในหลายๆ รูปแบบของการติดต่อสื่อสารกัน ในกิจการวิทยุสมัครเล่นตามปกติเราจะคุ้นเคยกับการติดต่อสื่อสารใน Mode เสียง การติดต่อสื่อสารใน Mode

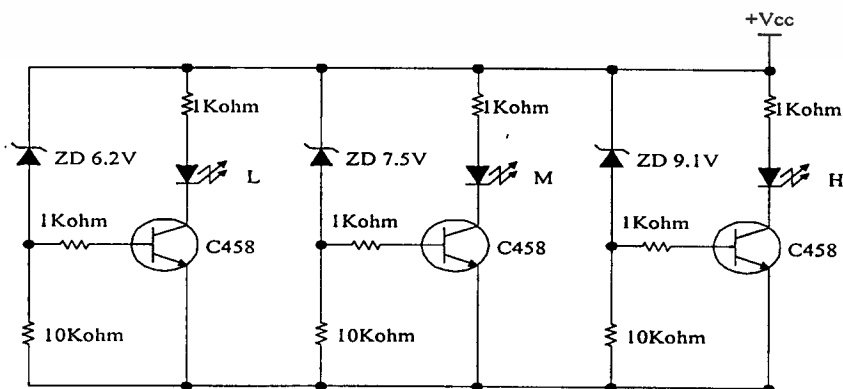
Packet Radio พูดง่ายก็คือ แทนที่เราจะใช้เสียงในการติดต่อสื่อสาร เราใช้ไมโครคอนโทรลเลอร์ ไม่มาควบคุมในการรับส่งข้อมูล ซึ่งรูปแบบของวงจรแพกเกจเรดิโอจะเป็นดังรูปที่ 3.3 ทุกครั้งที่มีการนำไปใช้



รูปที่ 3.3 วงจรแพคเกจรีโอ

### 3.3.2 วงจรวัดระดับแรงดัน

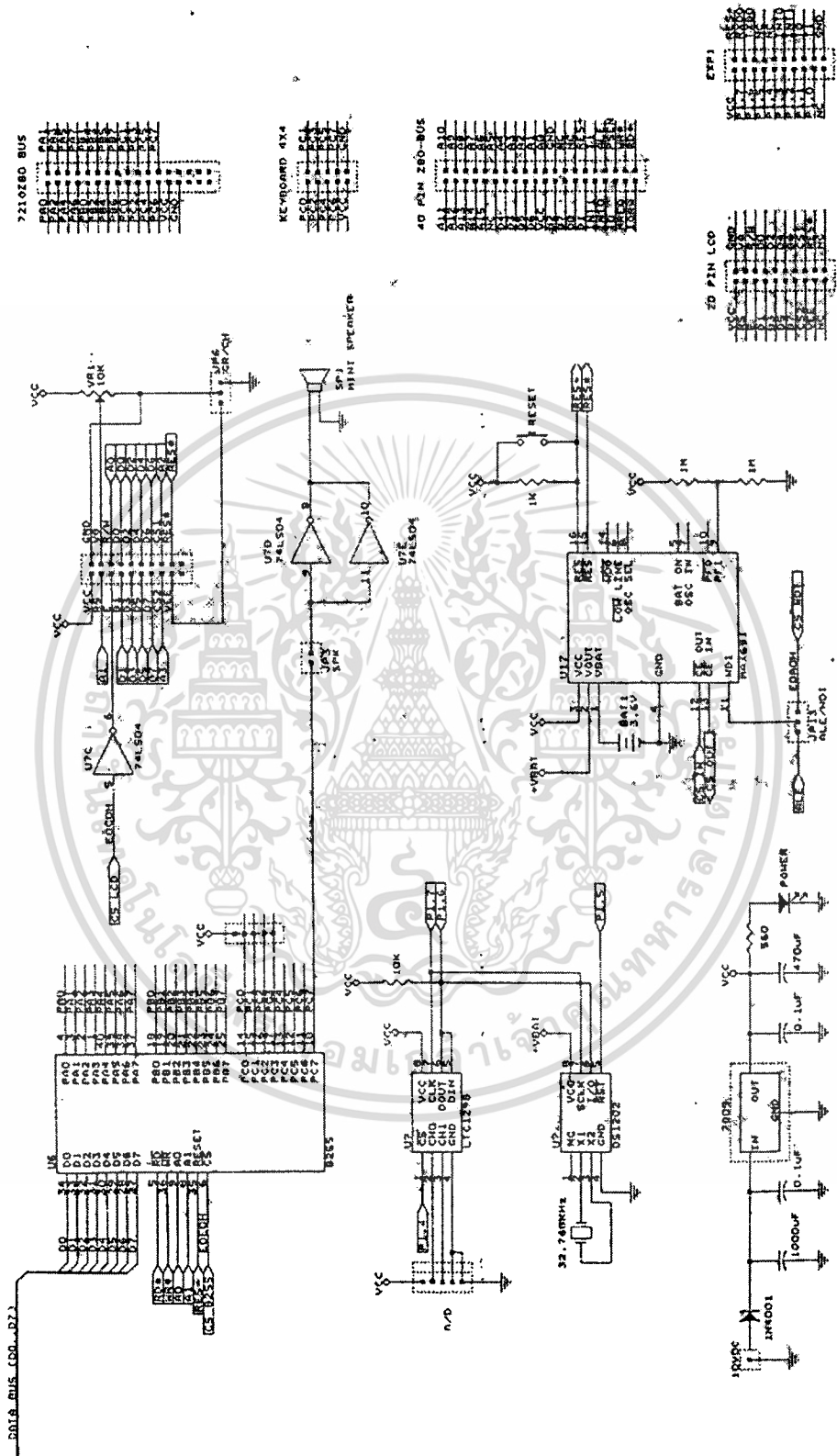
ในส่วนของวงจรวัดระดับแรงดันนี้ จะเป็นการตรวจสอบระดับแรงดันไฟที่ใช้ในวงจร ว่ามีระดับที่จะใช้จ่ายภายในวงจรในระดับใด ซึ่งอาจจะเป็นตัวบอกขอบเขตการใช้แรงดันไฟฟ้าควรปรับใช้ไม่เกินระดับใด และถ้าทำการปรับแรงดันไฟจนเกินจากแรงดันไฟที่ต้องการ ไดโอดเปล่งแสงก็จะแสดงผลเพื่อบอกว่าตอนนี้แรงดันไฟที่ปรับในการใช้งานเกิน ควรที่จะทำการปรับแรงดันไฟให้ลดลง ซึ่งรูปแบบของวงจรจะเป็นดังรูปที่ 3.4



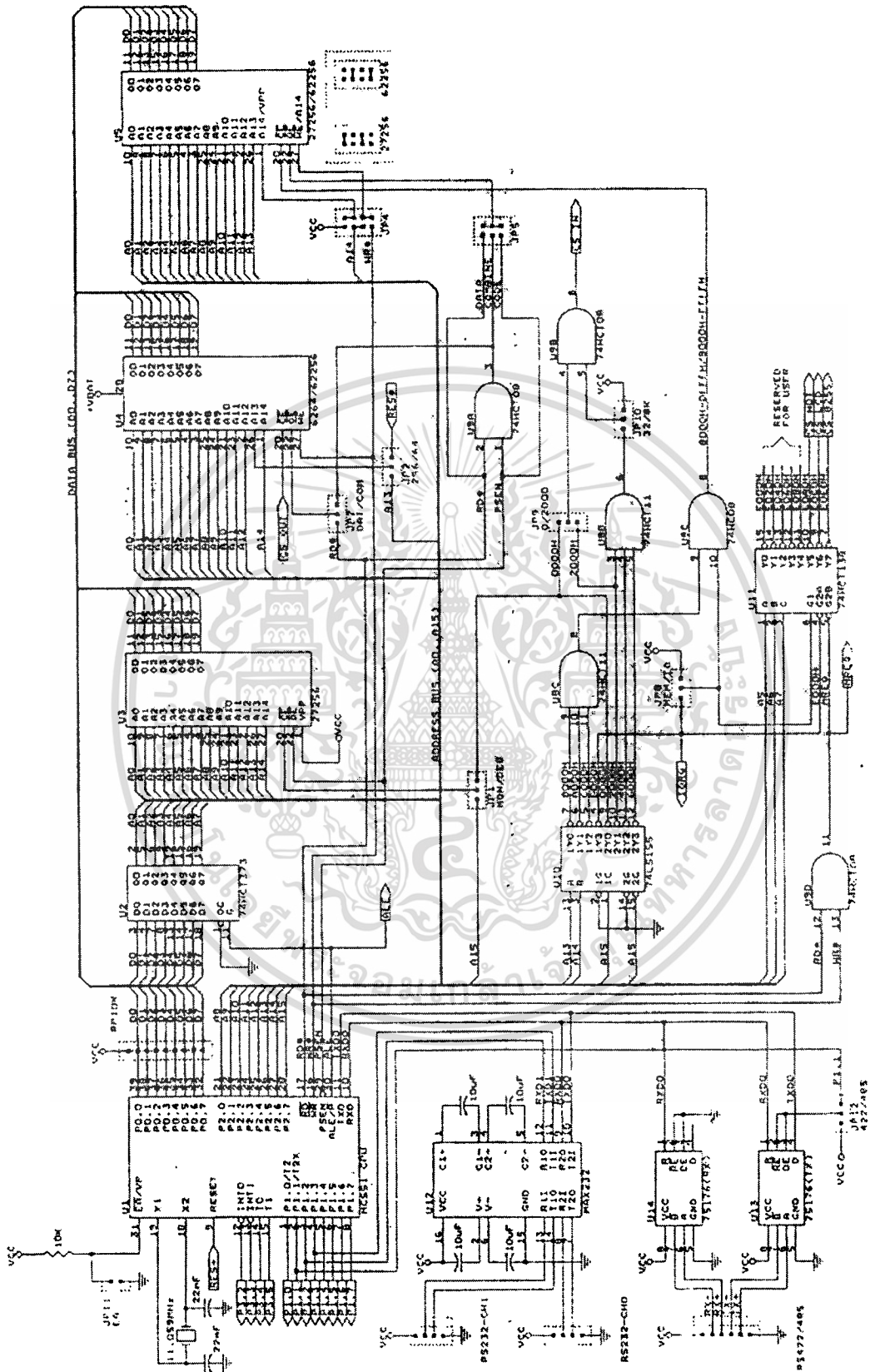
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้แก้ไขหรือดัดแปลงเนื้อหาใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกหรือเผยแพร่เอกสารนี้โดยไม่ได้รับอนุญาตจากเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 3.4 วงจรวัดระดับแรงดัน เจ้าของเอกสารทุกครั้งที่มีการนำไปใช้





เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 รูปที่ 3.6 วงจรภายในของบอร์ดไมโครคอนโทรลเลอร์ รุ่น ET - BASIC32 V3.0  
 ไม่ว่ากรณีใดๆ ก็ตาม หากมีข้อผิดพลาดหรือข้อสงสัย กรุณาติดต่อฝ่ายบริการลูกค้า



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆก็ตาม หากต้องการนำเอกสารนี้ไปใช้ กรุณาติดต่อขอสงวนลิขสิทธิ์ไว้ก่อน

รูปที่ 3.6 (ต่อ) วงจรภายในของบอร์ดไมโครคอนโทรลเลอร์ รุ่น ET - BASIC32 V3.0



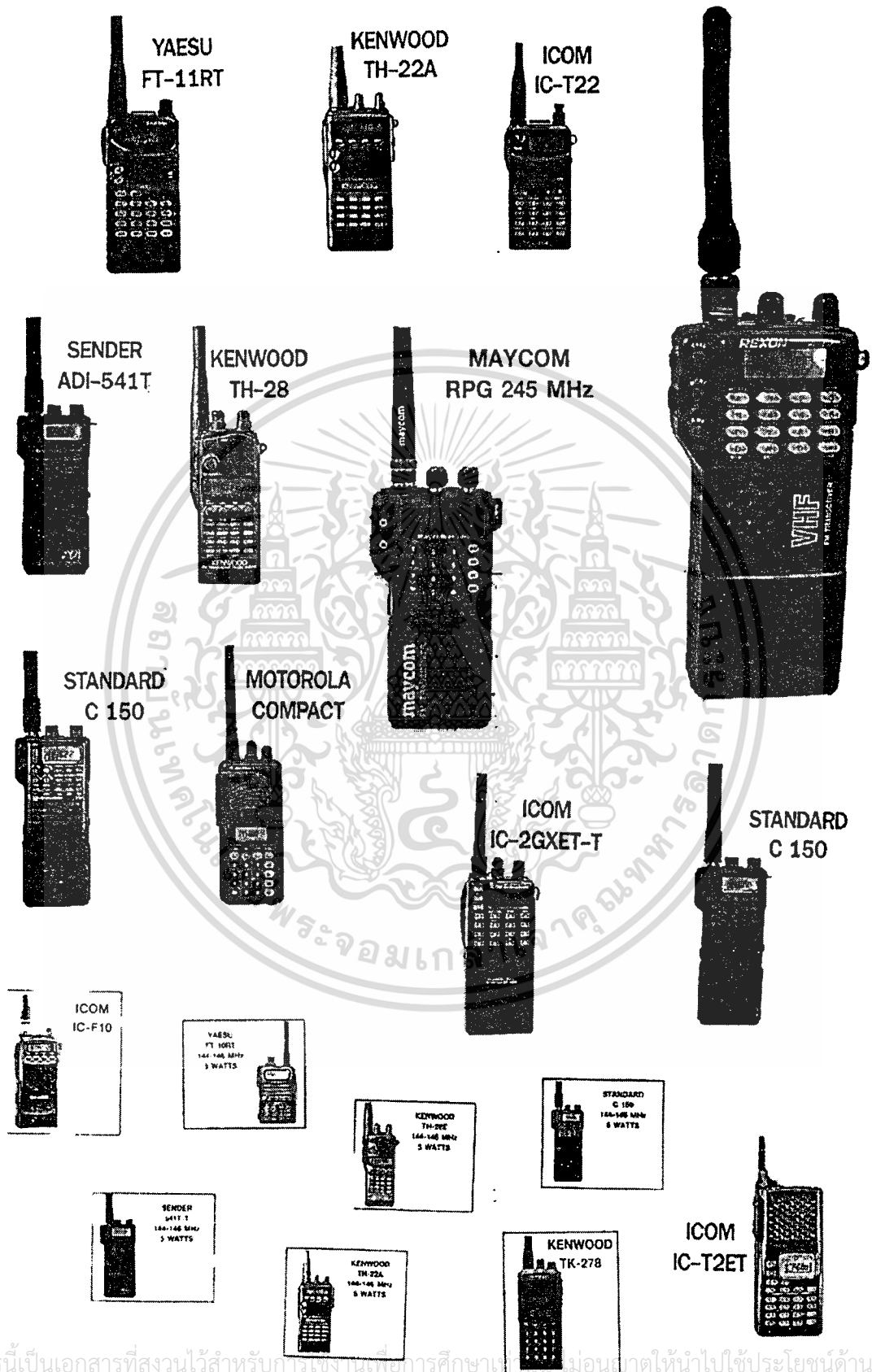
โดยจะมีปุ่มควบคุมการใช้งานทั้งหมด 64 ปุ่ม ซึ่งจะประกอบไปด้วยปุ่มตัวเลขตั้งแต่ 0 - 9 และตัวอักษรภาษาอังกฤษ A - Z ทั้งหมด 26 ตัว แล้วก็พยัญชนะต่างๆ ที่ใช้ในคอมพิวเตอร์ทั้งหมด 28 ปุ่ม ซึ่งใช้เป็นปุ่มในการคีย์ข้อความหรือข้อมูลในการสนทนาระหว่างสถานีของผู้ส่งและสถานีของผู้รับ ซึ่งรูปแบบของแป้นพิมพ์ตัวอักษรที่ใช้จะเป็นดังรูปที่ 3.8

|     |    |   |   |   |    |       |   |
|-----|----|---|---|---|----|-------|---|
| 0   | 1  | 2 | 3 | 4 | 5  | 6     | 7 |
| 8   | 9  | A | B | C | D  | E     | F |
| G   | H  | I | J | K | L  | M     | N |
| O   | P  | Q | R | S | T  | U     | V |
| W   | X  | Y | Z | ( | )  | !     | ← |
| F1  | F2 | : | ' | " | ?  | .     | & |
| F3  | F4 | ; | ↑ | % | \$ | *     | = |
| Esc | Sp | ← | ↓ | → | #  | Enter |   |

รูปที่ 3.8 แป้นพิมพ์ตัวอักษร

### 3.3.5 เครื่องรับส่งวิทยุ

เครื่องรับส่งวิทยุโดยทั่วไปเรามักจะเห็นจากวิทยุสมัครเล่นบ้าง จากหน่วยงานของรัฐบ้างที่นำมาใช้งานในการควบคุมงานหรือการติดต่อสื่อสารระหว่างกัน ซึ่งเครื่องรับวิทยุสื่อสารที่ใช้กันอยู่เป็นรูปแบบที่เรานำมาใช้ในปริญญานิพนธ์ฉบับนี้ โดยโครงงานนี้สามารถที่จะนำมาใช้ได้กับเครื่องรับวิทยุส่งได้ทุกรุ่นทุกยี่ห้อ ไม่ว่าจะเป็นแบบมือถือ ดิครดยนต์ หรือประจำสถานี เพียงแต่มีช่องสำหรับต่อไมโครโฟน และลำโพงภายนอกเท่านั้น และใช้งานได้กับทุกความถี่ไม่ว่าจะเป็นช่องความถี่ไหนก็ได้ แต่ในการใช้งานควรที่จะเลือกช่องความถี่ที่ไม่มีใครสนทนากันเพื่อทำให้การสนทนามีประสิทธิภาพเพิ่มมากขึ้น ซึ่งรูปแบบและลักษณะของเครื่องรับส่งวิทยุจะมีด้วยกันหลายแบบ หลายยี่ห้อดังรูปที่ 3.9



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิใช้รูปที่ 3.9 รูปแบบของเครื่องรับส่งวิทยุของเอกสารทุกครั้งที่มีการนำไปใช้

## บทที่ 4

### การทดลอง และผลการทดลอง

ในบทนี้จะกล่าวถึงการทดลอง การทำงาน ในวงจรส่วนต่างๆ ของปริยญาณิพนธ์ฉบับนี้ ว่า มีผลการทดลองที่ได้เป็นไปตามที่ออกแบบไว้หรือไม่ โดยใช้เครื่องมือวัดทำการตรวจสอบผลของรูปสัญญาณ การทำงานในแต่ละส่วนของวงจรที่ได้สร้างขึ้น ซึ่งสามารถแบ่งออกเป็นการทดลองในแต่ละส่วน ได้ดังนี้

#### 4.1 การทดลอง

การทดลองและทดสอบการทำงานแบ่งแยกเป็นลักษณะของการทดสอบวัดสัญญาณ บริเวณจุดการทำงานที่สำคัญต่างๆ และเป็นการทดลองประสิทธิภาพของวงจรวัดระดับแรงดันและการทดลองโดยรวม ซึ่งเราจะมาทำการทดลองการทำงานของระบบการรับส่งข้อมูลและรูปแบบการต่อใช้งานในจุดต่างๆ รวมถึงรูปแบบของเครื่องในส่วนต่างๆ ที่ใช้งานกัน อาจจะแยกเป็นส่วนประกอบของเครื่องส่งข้อมูลและเครื่องรับข้อมูล โดยมีการต่อเข้ากับเครื่องรับส่งวิทยุสื่อสาร ในการสนทนาหรืออาจจะเป็นการส่งข้อมูลทางช่องการสื่อสารระหว่างกันก็ได้

#### 4.2 อุปกรณ์ที่ใช้ในการทดลอง

- |                                                                      |           |
|----------------------------------------------------------------------|-----------|
| 1. เครื่องวิทยุพกพา ความถี่โดยไมโครคอนโทรลเลอร์ สำหรับวิทยุสมัครเล่น | 2 ชุด     |
| 2. วิทยุรับส่งไอคอม รุ่น P2-ET กำลังส่ง 5 วัตต์พร้อมแบตเตอรี่        | 2 ตัว     |
| 3. เส้าอากาศสไลด์                                                    | 2 ตัว     |
| 4. เครื่องคอมพิวเตอร์                                                | 2 เครื่อง |
| 5. แผ่นโปรแกรม Procom                                                | 2 แผ่น    |
| 6. เครื่องพิมพ์                                                      | 1 เครื่อง |
| 7. ออสซิลโลสโคป                                                      | 1 เครื่อง |
| 8. เพาเวอร์ซัพพลาย                                                   | 2 ชุด     |
| 9. ชุดแบตเตอรี่ชนิดนิเกิลแคดเมียม ขนาด AA 9.6 V 800mA                | 2 ชุด     |
| 10. ชุดแบตเตอรี่สำรองชนิดนิเกิลแคดเมียม ขนาด AA 3.6 - 4.5 V 800mA    | 2 ชุด     |

เอกสารนี้เป็นเอกสารต้นฉบับของงานวิจัยที่จัดทำขึ้นเพื่อใช้ในการศึกษาวิจัยเท่านั้น ไม่ควรนำเอกสารนี้ไปใช้ประโยชน์อื่นใดโดยไม่ได้รับอนุญาตจากเจ้าของเอกสาร

- |                                                          |           |
|----------------------------------------------------------|-----------|
| 11. สายต่อแฉีกไมโครโฟนและลำโพงพร้อมพอร์ตคอมพิวเตอร์ต่างๆ | 2 ชุด     |
| 12. ฟังก์ชันเจนเนอเรเตอร์                                | 1 เครื่อง |

## 4.3 การทดสอบการทำงานของระบบ

### 4.3.1 การทดลองวงจรมอดูเลตและดีมอดูเลต

พิจารณาการทดสอบของระบบการมอดูเลตและการดีมอดูเลต แบบ FSK ซึ่งเป็นหัวใจสำคัญของการสื่อสารแบบวิทยุแพกเกจ โดยการทำงานที่สำคัญจะอยู่ในส่วนของไอซีเบอร์ TCM 3105 ซึ่งมีตำแหน่งขาใช้งานที่สำคัญคือ

1. ขา VDD (ขาที่ 1) เป็นขาที่ต่อกับเพาเวอร์ซัพพลาย
2. ขา VSS (ขาที่ 9) เป็นขาที่ต่อกับกราวด์
3. ขา DTR (ขาที่ 14) เป็นขาที่รับข้อมูลดิจิทัล จากบอร์ดประมวลผลสัญญาณข้อมูล ET - BASIC32 V3.0 มาเพื่อทำการมอดูเลตแปลงเป็นสัญญาณแอนะล็อกต่อไป
4. ขา CTS (ขาที่ 8) เป็นขาที่ทำหน้าที่ส่งสัญญาณดิจิทัลที่ผ่านจากการดีมอดูเลต สัญญาณแอนะล็อกของสัญญาณข่าวสารที่รับมาจากวิทยุสื่อสาร ส่งต่อไปให้กับบอร์ดประมวลผลสัญญาณข้อมูล ET - BASIC32 V3.0 เพื่อประมวลผลเป็นตัวอักษรแสดงที่จอ LCD ต่อไป
5. ขา RXA (ขาที่ 4) เป็นขาที่ทำหน้าที่รับสัญญาณแอนะล็อกของวิทยุสื่อสารเข้ามา เพื่อจะทำการดีมอดูเลตต่อไป
6. ขา TXA (ขาที่ 11) เป็นขาที่ทำหน้าที่ส่งสัญญาณแอนะล็อกที่ผ่านการมอดูเลตแบบ FSK ไปยังเครื่องวิทยุสื่อสารเพื่อส่งอากาศต่อไป ซึ่งคุณสมบัติ และหน้าที่ของการทำงานต่างๆ ของไอซีเบอร์ TCM3105 มีรายละเอียดที่สมบูรณ์ศึกษาได้จากภาคผนวก

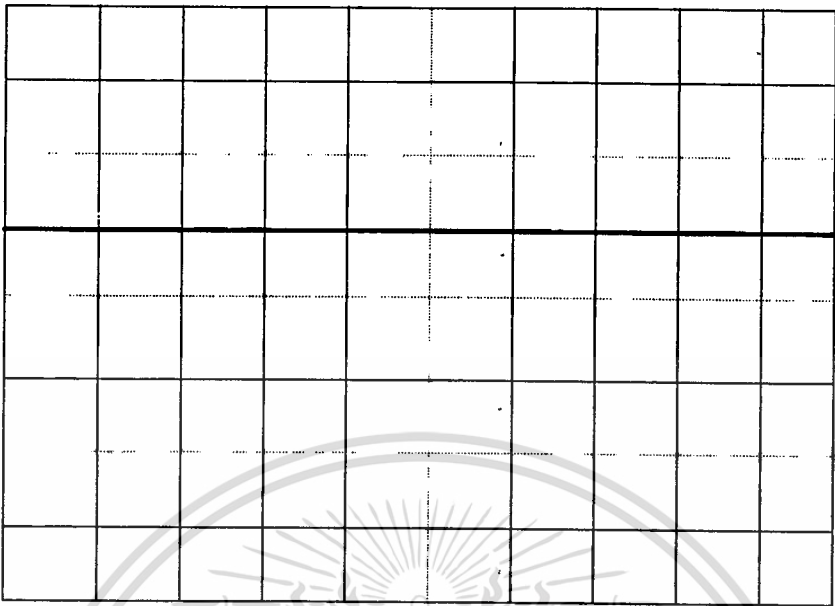
### 4.3.2 การทดสอบประสิทธิภาพการทำงานของไอซี TCM 3105

ทำการทดสอบโดยป้อนระดับแรงดันไฟ DC 5 โวลต์ เข้าที่ขา DTR (ขา 14) วัดค่าสัญญาณความถี่ที่ผลิตออกมายังขา TXA (ขา 11) ได้ ค่าความถี่ประมาณ 1,300 เฮิร์ตซ์ ดังรูปที่ 4.1 ลักษณะของสัญญาณการมอดูเลตแบบ FSK และผลิตความถี่ออกมาในลักษณะสัญญาณดิจิทัล

จากนั้นเปลี่ยนค่าระดับที่ป้อนยังขา DTR ใหม่ เป็นระดับแรงดันไฟ DC 0 โวลต์ เข้าที่ขา DTR (ขา 14) วัดค่าสัญญาณความถี่ที่ผลิตออกมายังขา TXA (ขา 11) ได้ ค่าความถี่ประมาณ 2,100 เฮิร์ตซ์ ดังรูปที่ 4.2 ลักษณะของสัญญาณการมอดูเลตแบบ FSK และผลิตความถี่ออกมาในลักษณะสัญญาณดิจิทัล

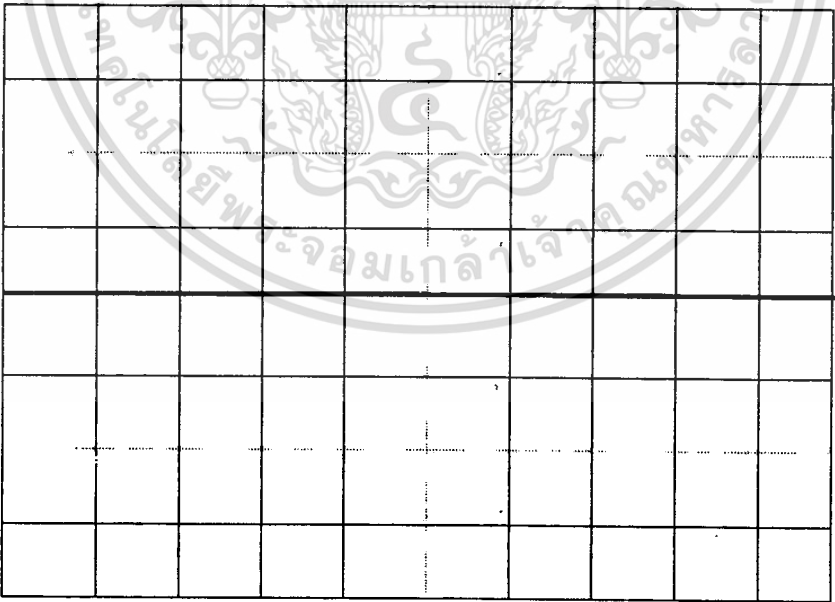
ทำการใช้ฟังก์ชันเจนเนอเรเตอร์ ผลิตสัญญาณความถี่ ไซน์เวฟที่ความถี่ 1,300 เฮิร์ตซ์ และ 2,100 เฮิร์ตซ์ แอมพลิจูด 5 โวลต์ เข้าที่ขา RXA (ขา 4) ทำการวัดสัญญาณที่ได้จากขา CTS (ขา 8) เมื่อความถี่ที่ต่อกัน จะได้สัญญาณดังรูปที่ 4.3 และรูปที่ 4.4

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



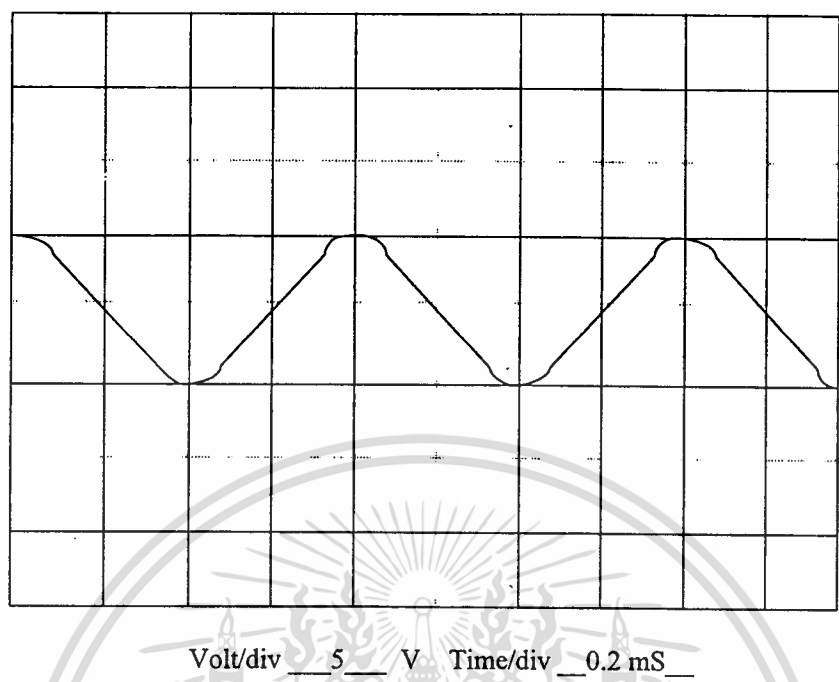
Volt/div 5 V Time/div 0.2 mS

รูปที่ 4.1 ลักษณะของสัญญาณการมอดูเลตแบบ FSK และผลิตความถี่ออกมาในลักษณะสัญญาณ  
ดิจิทัล



Volt/div 5 V Time/div 0.2 mS

เอกสาร **รูปที่ 4.2** ลักษณะของสัญญาณการมอดูเลตแบบ FSK และผลิตความถี่ออกมาในลักษณะสัญญาณการคำ  
ไม่ว่ากรณีใด **ดิจิทัล** อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.3 ลักษณะของสัญญาณที่ถูกมอดูเลตจากสัญญาณดิจิตอล เป็นสัญญาณแอนะล็อกที่มีความถี่ 1,300 เฮิรตซ์

จากการพิจารณาผลการทดสอบสามารถบันทึกผลการทดลองได้ดังนี้

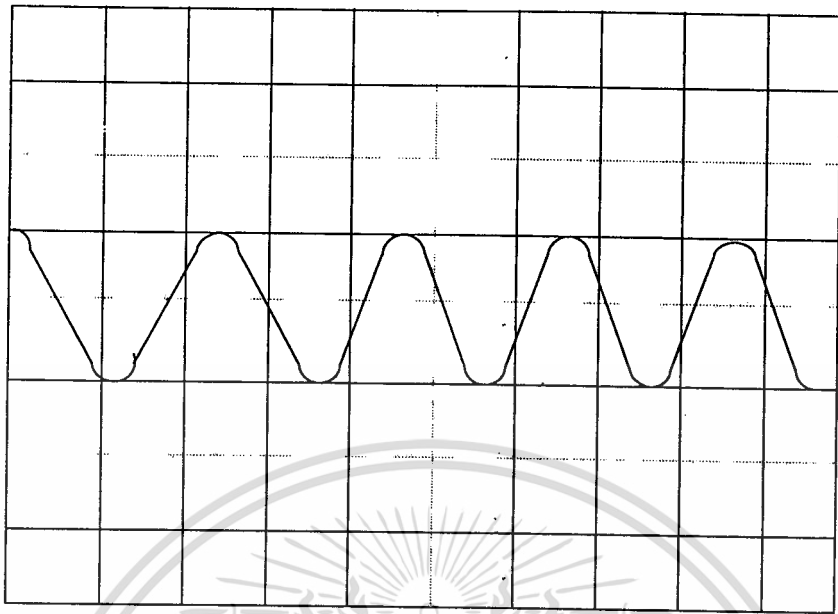
การมอดูเลตแบบ FSK โดยไอซี เบอร์ TCM3105 ทำหน้าที่มอดูเลตและดีมอดูเลตสัญญาณจากสัญญาณแอนะล็อก เป็นสัญญาณดิจิตอลและจากสัญญาณดิจิตอลเป็นสัญญาณแอนะล็อกที่มีความถี่ 1,300 เฮิรตซ์และความถี่ 2,100 เฮิรตซ์

ตารางที่ 4.1 การเปรียบเทียบค่าผลการทดสอบระดับแรงดันลอจิกกับค่าความถี่ที่ได้รับ

การเทียบค่าระดับแรงดันระดับลอจิก " 0 " และระดับลอจิก " 1 " พิจารณาผลได้จาก

| ระดับลอจิก | ความถี่ที่มอดูเลตออกมาและความถี่ที่ดีมอดูเลตได้ |
|------------|-------------------------------------------------|
| 0          | 2,100 Hz                                        |
| 1          | 1,300 Hz                                        |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



Volt/div 5 V Time/div 0.2 ms

รูปที่ 4.4 ลักษณะของสัญญาณที่ถูกมอดูเลตจากสัญญาณคิจิตอล เป็นสัญญาณแอนะล็อกที่มีความถี่ 2,100 เฮิรตซ์

#### 4.3.3 การทดสอบประสิทธิภาพของวงจรวัดระดับแรงดัน

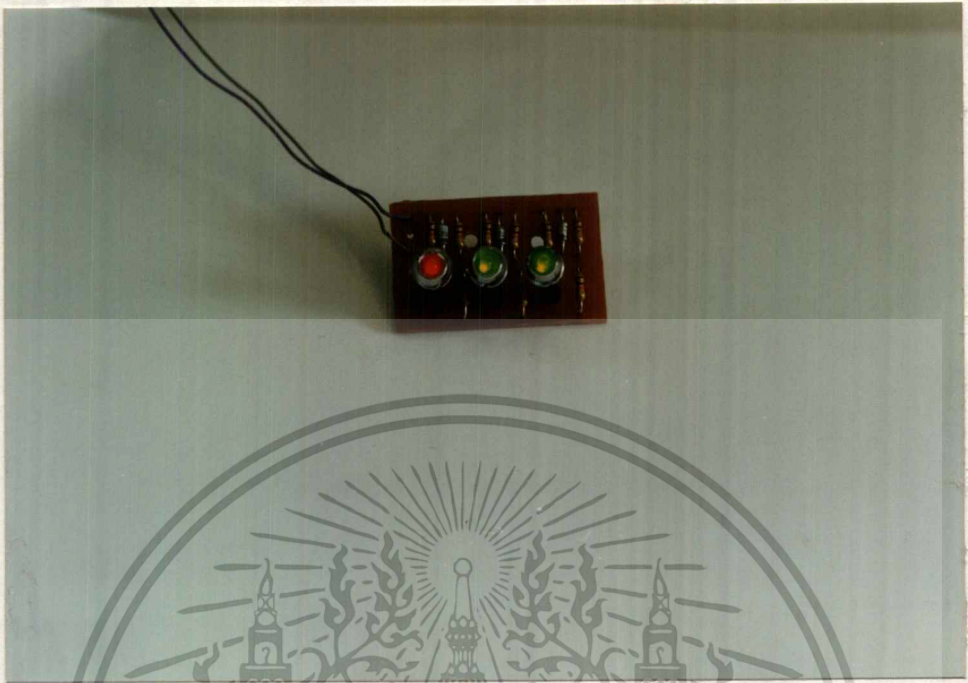
1) ใช้เพาเวอร์ซัพพลาย ปรับระดับแรงดันตั้งแต่ 9.6 โวลต์ - 4 โวลต์ ที่กระแส 800 - 1200 มิลลิแอมป์ พิจารณาผลของโหลดไฟทั้ง 3 ดวงได้ผลดังรูปที่ 4.5 ลักษณะสถานะของวงจรวัดระดับแรงดันไฟในช่วงแรงดันที่มีค่าสูงกว่า 9.2 โวลต์

2) ใช้เพาเวอร์ซัพพลาย ปรับระดับแรงดันตั้งแต่ 9.6 โวลต์ - 4 โวลต์ ที่กระแส 800 - 1200 มิลลิแอมป์ พิจารณาผลของโหลดไฟทั้ง 3 ดวงได้ผลดังรูปที่ 4.6 ลักษณะสถานะของวงจรวัดระดับแรงดันไฟในช่วงแรงดันประมาณ 7.8 โวลต์

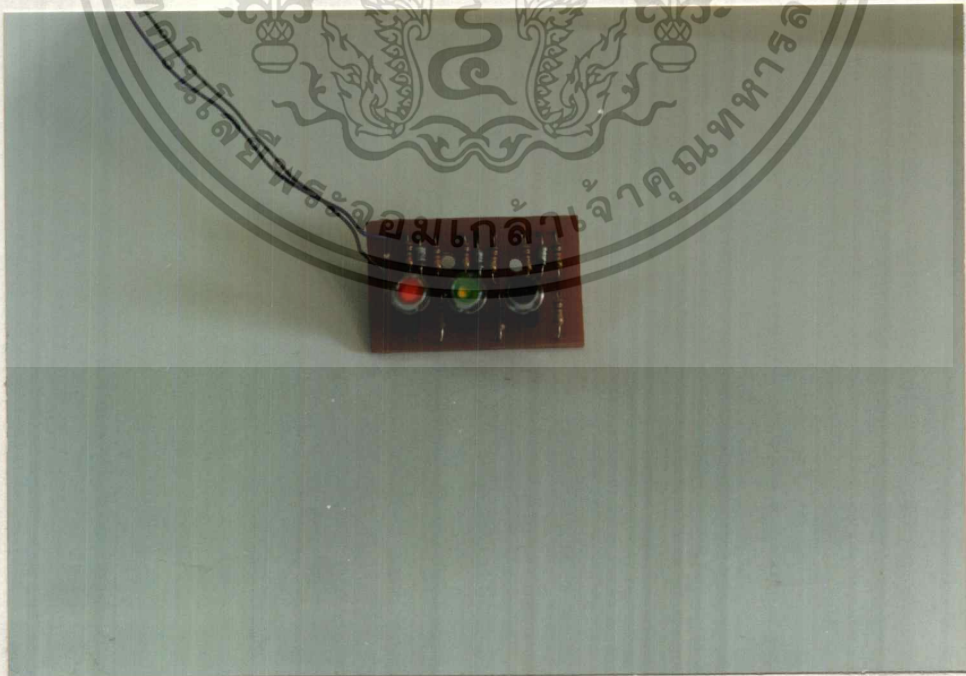
3) ใช้เพาเวอร์ซัพพลาย ปรับระดับแรงดันตั้งแต่ 9.6 โวลต์ - 4 โวลต์ ที่กระแส 800 - 1200 มิลลิแอมป์ พิจารณาผลของโหลดไฟทั้ง 3 ดวงได้ผลดังรูปที่ 4.7 ลักษณะสถานะของวงจรวัดระดับแรงดันไฟในช่วงแรงดันที่มีค่าต่ำกว่า 7.2 โวลต์

ซึ่งรูปทั้ง 3 รูปจะเป็นการบอกถึงระดับแรงดันที่ควรจะใช้งานในระดับใดเท่านั้น ไม่ควรใช้แรงดันเกินจากที่ได้กำหนดไว้ในการทดลอง หรือว่าเป็นการตรวจสอบให้เห็นถึง ระดับแรงดันที่ใช้ขนาดนี้อยู่ในช่วงเท่าใด การใช้งานควรอ่านคู่มือใช้งานประกอบการทดลองด้วย

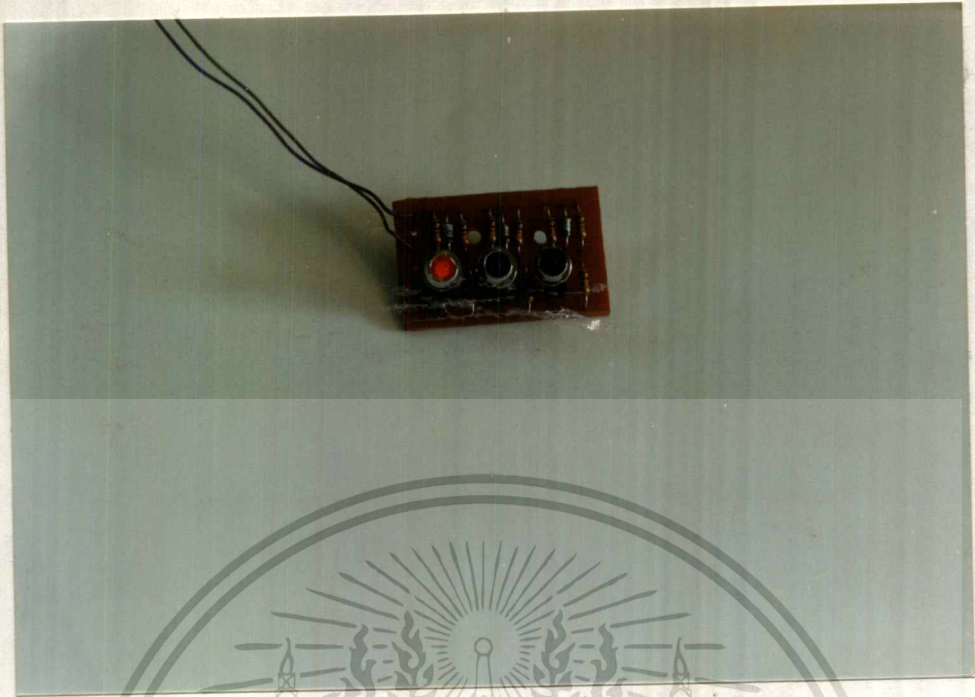
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.5 ลักษณะสภาวะของวงจรวัดระดับแรงดันไฟในช่วงแรงดันที่มีค่าสูงกว่า 9.2 โวลต์



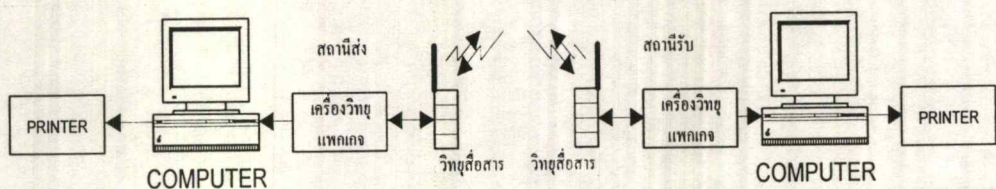
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีรูปที่ 4.6 ลักษณะสภาวะของวงจรวัดระดับแรงดันไฟในช่วงแรงดันประมาณ 7.8 โวลต์



รูปที่ 4.7 ลักษณะสภาวะของวงจรวัดระดับแรงดันไฟในช่วงแรงดันที่มีค่าต่ำกว่า 7.2 โวลต์

#### 4.4 แผนผังแสดงการใช้งาน

ลักษณะของแผนผังการทำงานของระบบการรับส่งข้อมูลทางช่องสัญญาณการสื่อสาร โดยทั่วไปแล้วจะมีส่วนประกอบที่เราพอจะเห็นกันเพียงไม่กี่ส่วนเท่านั้น ซึ่งจะมีสถานีผู้ส่งข้อมูลและสถานีผู้รับข้อมูลทั้งสองสถานีสามารถที่จะติดต่อสื่อสารหรือว่าได้ตอบในการสนทนากันได้เลย การทำงานในช่วงนี้จะมีลักษณะการคุยทางอินเตอร์เน็ตนั่นเอง ในส่วนนี้จะใช้โมเด็มในการสนทนากัน แต่ที่เราจะใช้แพคเกจเรดิโอส่งสัญญาณออกจากช่องสัญญาณสื่อสารในการสนทนากัน รูปแบบที่ใช้จะแตกต่างกัน ดังรูปที่ 4.8 แผนผังการใช้งานของวิทยุแพคเกจร่วมกับอุปกรณ์อื่นๆ ที่ใช้ในงานทดลองจริง



รูปที่ 4.8 แผนผังการใช้งานของวิทยุแพคเกจร่วมกับอุปกรณ์อื่นๆ ที่ใช้ในงานทดลองจริง

เอกสารนี้เป็นเอกสารของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี ไม่ควรนำเอกสารนี้ไปใช้โดยไม่ได้รับอนุญาตจากมหาวิทยาลัย

#### 4.5 ระบบการทำงานและการทดลองใช้งานจริง

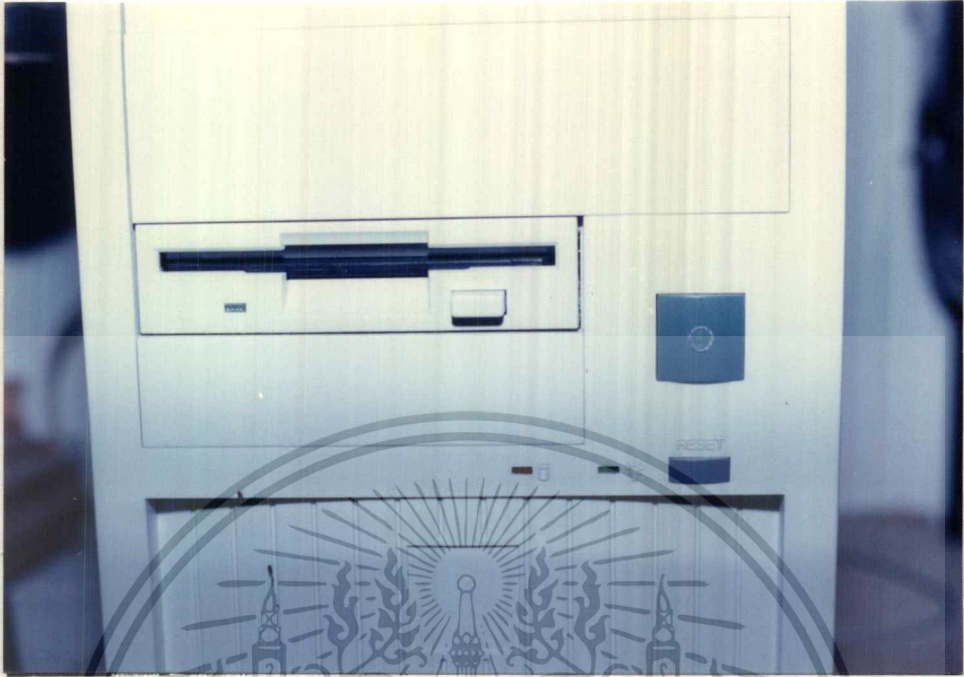
จากรูปที่ 4.8 แสดงให้เห็นถึงลักษณะการทำงานของระบบโดยรวมของการสื่อสาร โดยเครื่องวิทยุแพกเกจที่มีคู่สถานี 2 สถานี คือ สถานีผู้ส่ง และสถานีผู้รับ ซึ่งส่วนประกอบที่สำคัญที่สถานีทั้งสองต้องมี คือ เครื่องวิทยุแพกเกจและวิทยุสื่อสารย่านวิทยุสมัครเล่นพร้อมสายอากาศ แล้วก็คอมพิวเตอร์กับปรินเตอร์ด้วย

การทำงานเริ่มจากเมื่อทั้ง 2 สถานีทำการประกอบตามรูปที่ 4.8 แล้ว โดยใช้สายต่อกับหัว DB-9 และ DB-25 ต่อระหว่างเครื่องวิทยุแพกเกจของพอร์ต COM1 หรือ COM2 ซึ่งเป็นการสื่อสารแบบอนุกรมและต่อสายปรินเตอร์เข้ากับคอมพิวเตอร์ของพอร์ต COM1 หรือ COM2 ที่เหลือจากการเลือกใช้ ในขั้นแรก ต่อสายไมโครโฟนและสายลำโพงจากวิทยุสื่อสารเข้ากับเครื่องวิทยุแพกเกจ พิจารณาการต่ออุปกรณ์ต่างๆ

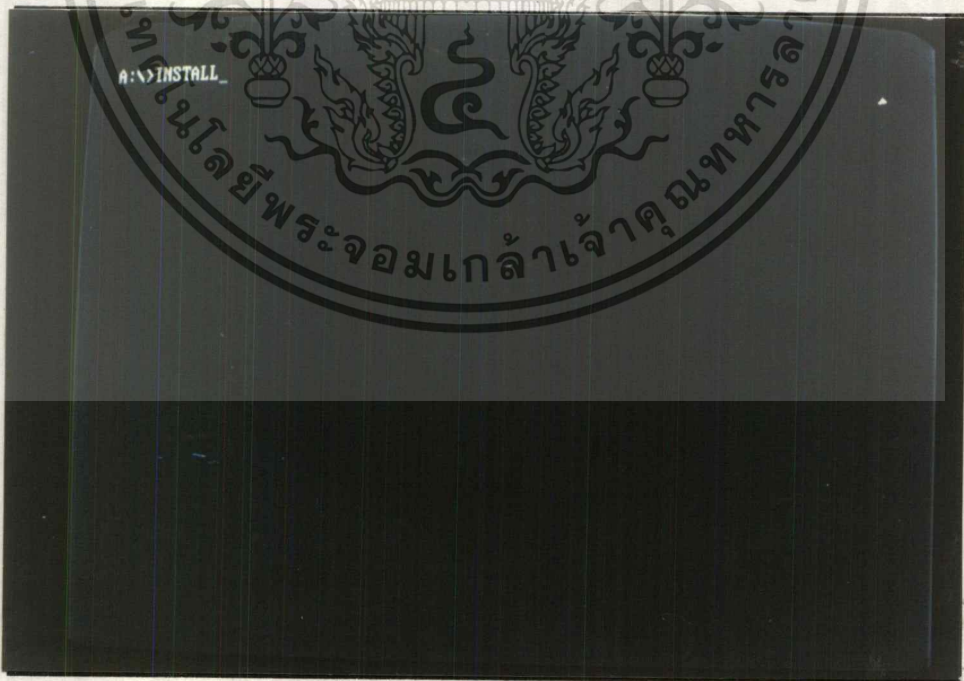
พิจารณาอีกส่วนของการ Setup โปรแกรมในระบบคอมพิวเตอร์ แล้วทำการใส่แผ่นดิสก์ที่มีโปรแกรมโปรคอมอยู่และทำการเรียกคำสั่ง Install ก็จะเข้าทำการ Setup โปรแกรมโปรคอมลงยังเครื่องคอมพิวเตอร์และเข้าสู่โปรแกรม โดยการพิมพ์ คำสั่ง PCP.EXE จะเข้าสู่โปรแกรมโดยสมบูรณ์ทำการ Setup พอร์ตสื่อสาร คือพอร์ต COM1 และ COM2 เข้าสู่โหมดเตรียมรับข้อมูล



เอกสารนี้เป็นเอกสารที่รูปที่ 4.9 ลักษณะการต่อวิทยุแพกเกจเข้ากับอุปกรณ์ ผ่านทางหัวต่อชนิดต่างๆ ด้านการคำ  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

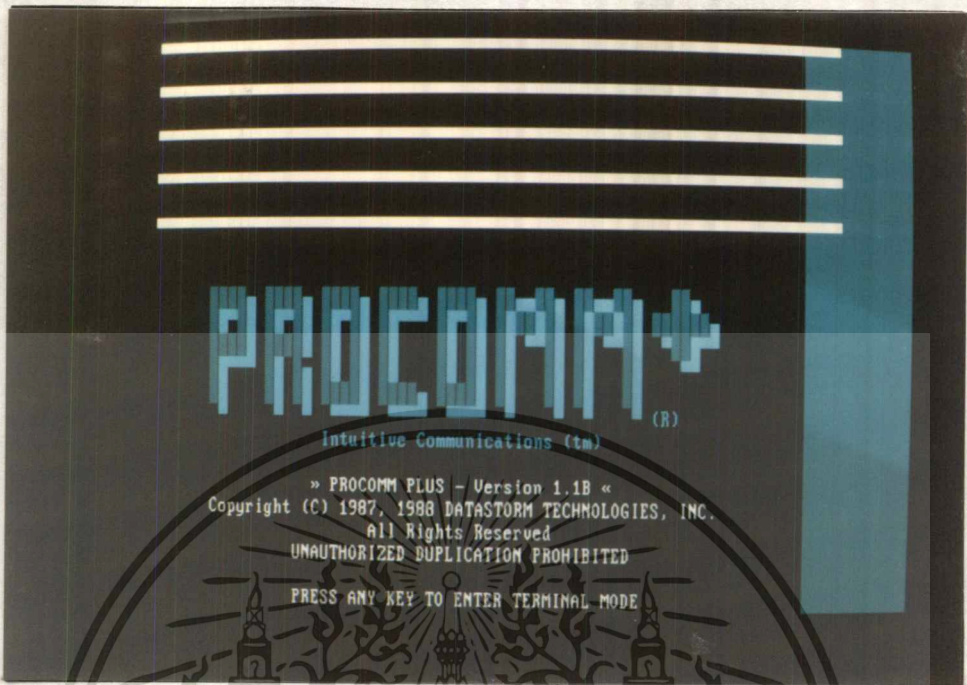


รูปที่ 4.10 การใส่แผ่นดิสก์ที่บรรจุ โปรแกรม PROCOM อยู่ เพื่อเตรียมทำการ Setup

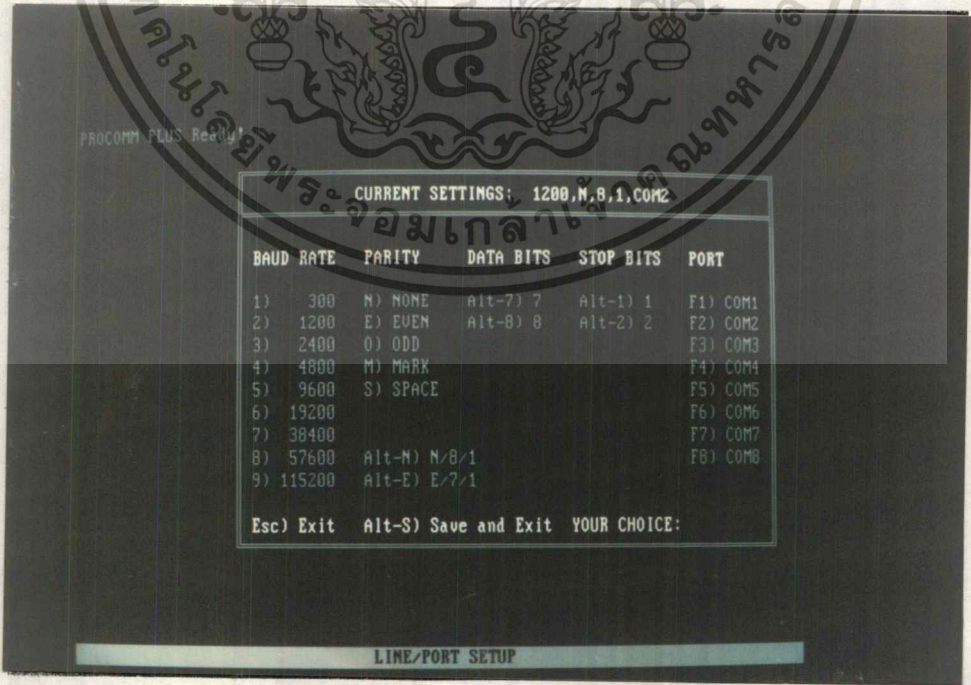


รูปที่ 4.11 การ Setup หรือ Install โปรแกรม PROCOM ลงสู่เครื่องคอมพิวเตอร์

เอกสารนี้เป็นเอกสารที่วางไว้สำหรับการใช้และเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้ทำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

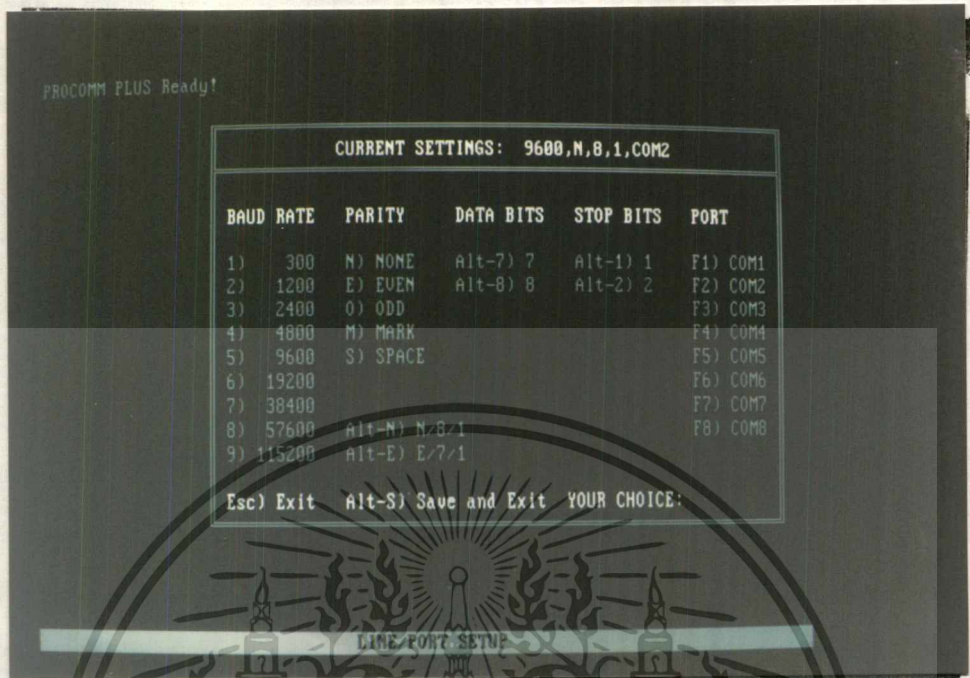


รูปที่ 4.12 หน้าจอแสดงผล โปรแกรม PROCOM ที่ผ่านการ Install แล้วพร้อมที่จะใช้งาน

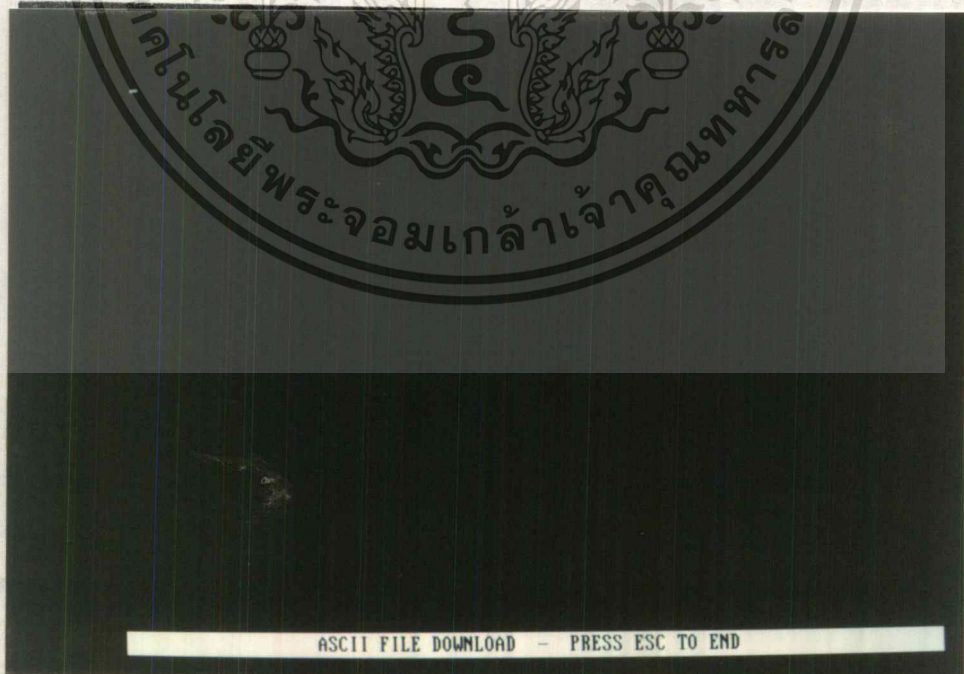


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น

รูปที่ 4.13 Menu Setup พอร์ตการสื่อสาร COM1 และ COM2 ที่มีการนำไปใช้



รูปที่ 4.14 Menu Setup บอกระบบการสื่อสารให้ใช้บอดเรท 9,600 bps



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามเผยแพร่โดยไม่ได้รับอนุญาตจากวิทยุแพคเกจ

รูปที่ 4.15 การเข้าสู่โหมด พร้อมรับข้อมูลจากวิทยุแพคเกจ

เมื่อเปิดสวิตช์ใช้งานเครื่องวิทยุสื่อสารทั้ง 2 สถานี โดยทดลอง ให้มีการติดต่อสื่อสารที่ช่องความถี่ผ่าน 145.500 MHz และเปิดสวิตช์เครื่องวิทยุแพคเกจของทั้ง 2 สถานีเช่นกัน และทำการตั้งค่า วัน - เดือน - ปี , เวลาปัจจุบัน, ชื่อสถานี (Callsign) โดยพิจารณาการใช้งานจากคู่มือการใช้งานเข้าสู่โหมด Connection ทั้ง 2 เครื่อง

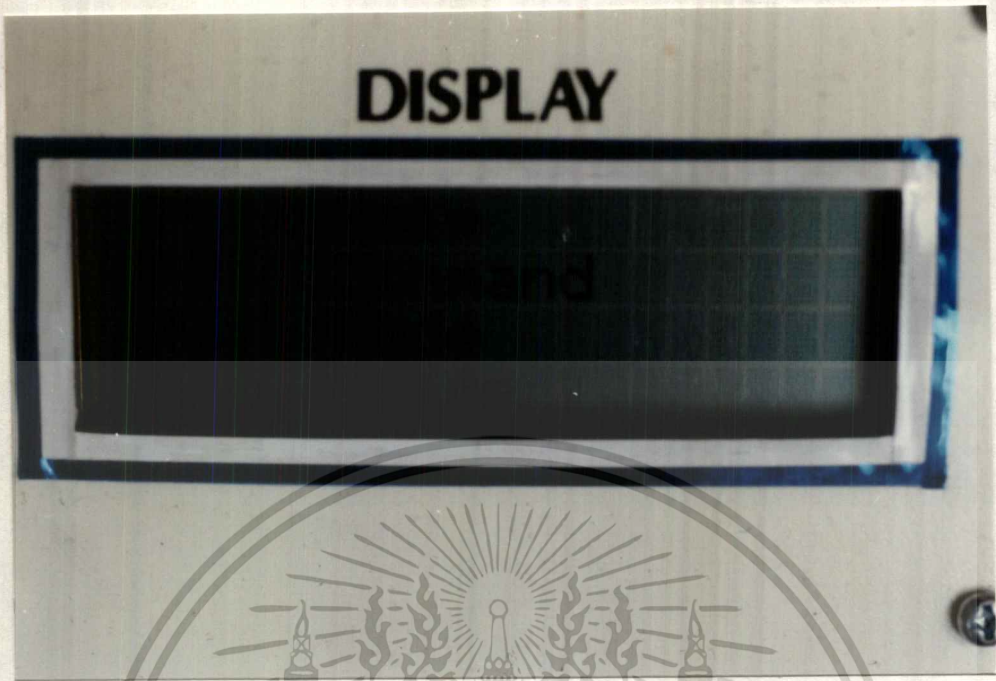
ทำการทดลองสื่อสาร โดยส่งข้อมูลข่าวสารโต้ตอบกันทั้ง 2 สถานี เมื่อทำการสื่อสารเสร็จเรียบร้อย เข้าสู่ Menu Command และ Menu Log - Book เข้าไปดู File O เพื่อดูข้อมูลที่บันทึกการติดต่อสื่อสารว่า บันทึกข้อมูลทั้งหมดถูกต้องหรือไม่ พร้อมทั้งทำการทดลองส่งข้อมูล โดยใช้คำสั่ง Link เพื่อถ่ายโอนข้อมูลมาเสนอผลที่เครื่องคอมพิวเตอร์ และพิมพ์ผลออกมาทางเครื่องพิมพ์ หรือเก็บข้อมูล LOG - BOOK ทั้งหมดไว้ในแผ่นดิสก์

การบันทึกข้อมูลสามารถทำได้โดยการใช้คำสั่ง Page - down ตามด้วยเลข 4 จากนั้นเติมชื่อ File ที่ต้องการตั้งชื่อพร้อมนามสกุล .TXT จากนั้นส่งข้อมูลจาก Log - Book สู่อุปกรณ์คอมพิวเตอร์ โดยใช้คำสั่ง LINK แล้วกด Esc ออกจากโปรแกรมก็สามารถได้ File ที่แสดงข้อมูล Log - Book เป็น File สามารถใช้คำสั่ง Type ตามด้วยชื่อ File.TXT ก็จะได้ข้อมูลที่ต้องการดังรูปต่อไปนี้



รูปที่ 4.16 การตั้งย่านความถี่ที่ใช้ในการทดลองสื่อสารที่ย่านความถี่ 145.500 MHz

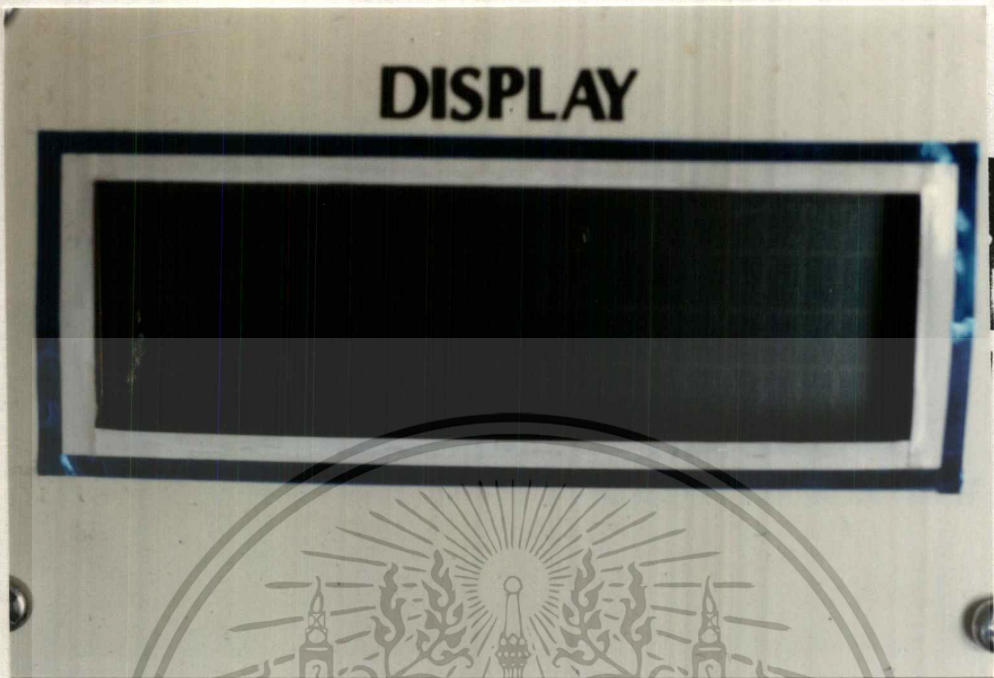
เอกสารนี้เป็นเอกสารสงวนลิขสิทธิ์สำหรับการใช้ในเพื่อการศึกษาเท่านั้น เมื่อผู้ยืมได้พ้นจากขอบเขตของการยืมแล้ว  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



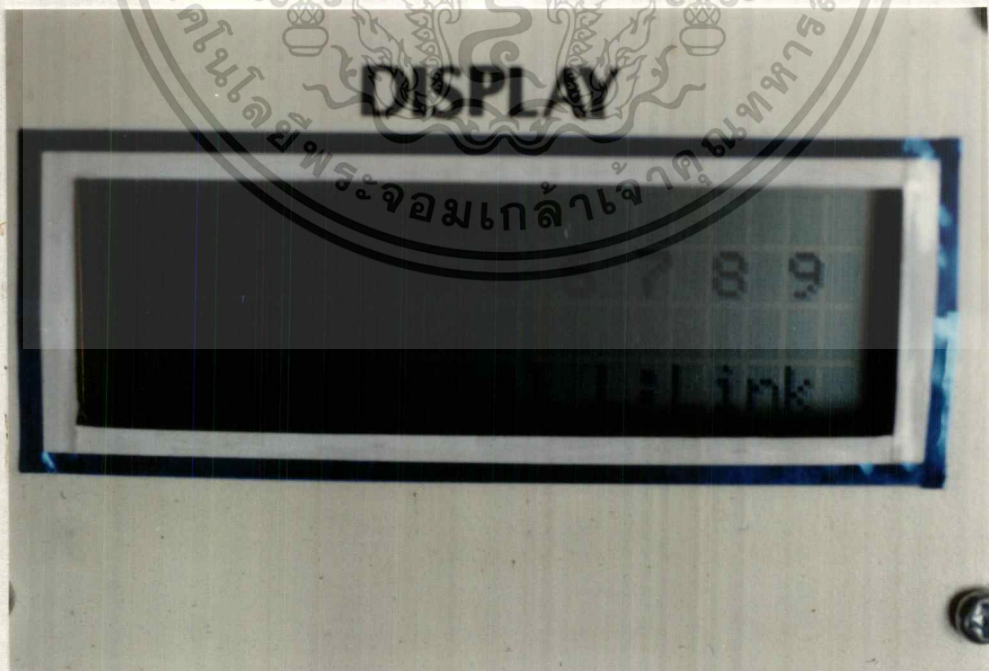
รูปที่ 4.17 รายการเลือกหลักหลังเปิดสวิตซ์ทำงานแก่เครื่องวิทยุแพกเกจ พร้อมทั้งจะใช้งาน



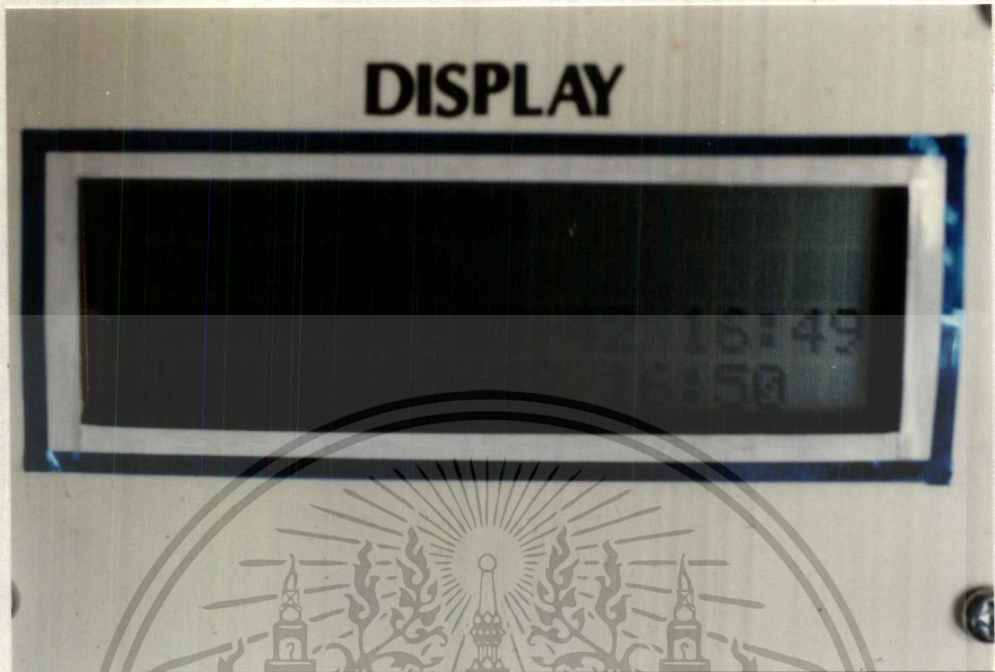
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ รูปที่ 4.18 สภาวะของ Connection พร้อมทั้งจะทำการติดต่อสื่อสารกับคู่สถานีที่มีการนำไปใช้



รูปที่ 4.19 การรับข้อมูลจากสถานีคู่เข้ามาโดยรับจาก HS2 BBB ข้อความว่า “HELLO”



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมีรูปที่ 4.20 กว๊านบันทึกข้อมูลการสนทนาอัตโนมัติ ทุกครั้งที่มีการนำไปใช้

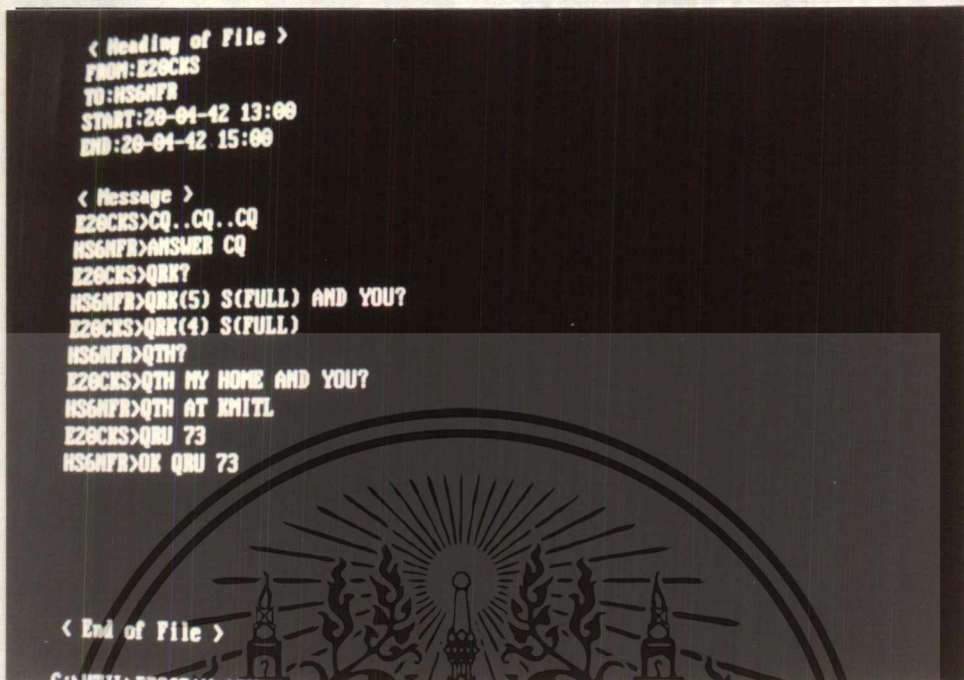


รูปที่ 4.21 ข้อความที่บันทึกการสนทนาที่เกิดขึ้น

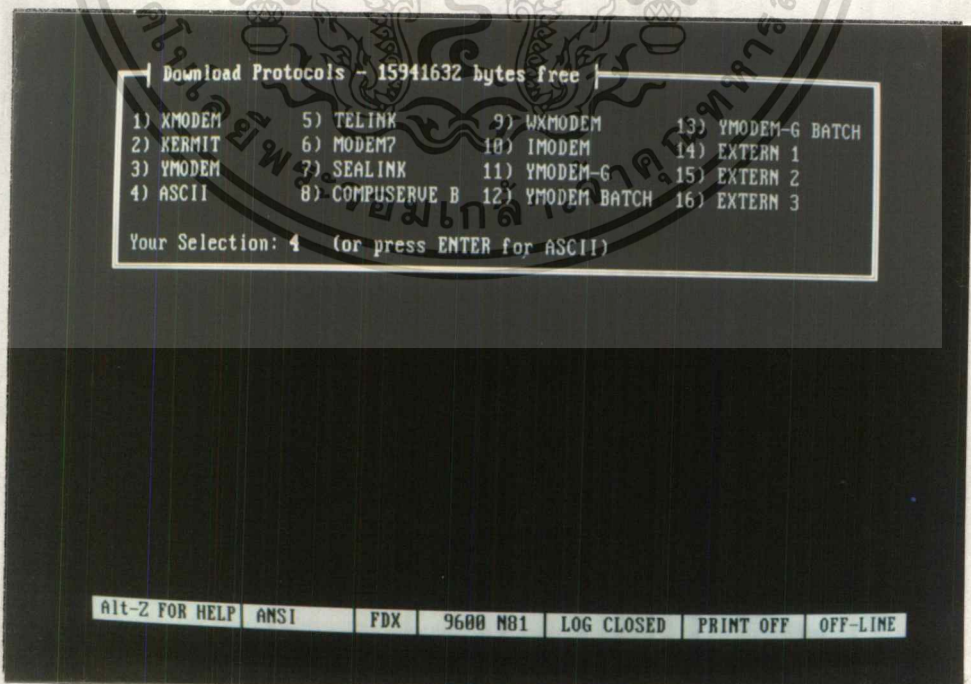


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
รูปที่ 4.22 การถ่ายโอนข้อมูลที่บันทึกการสนทนาจากแฟ้ม 0 ไปสู่เครื่องคอมพิวเตอร์

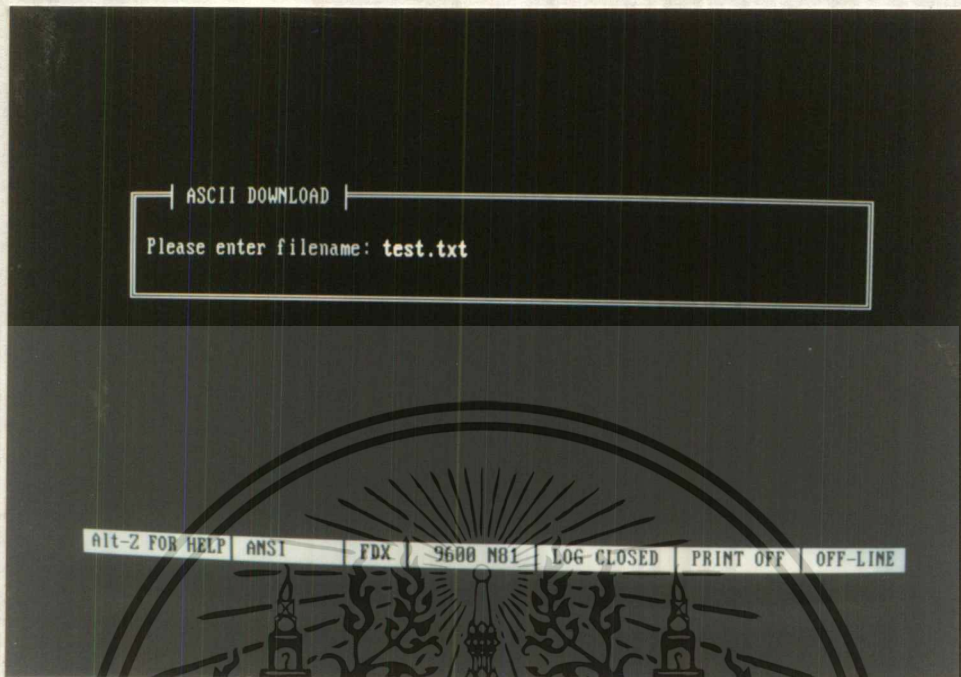
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามเผยแพร่ข้อมูลนี้ และต้องแจ้งเจ้าของสิทธิ์ที่สงวนไว้ให้นำไปใช้



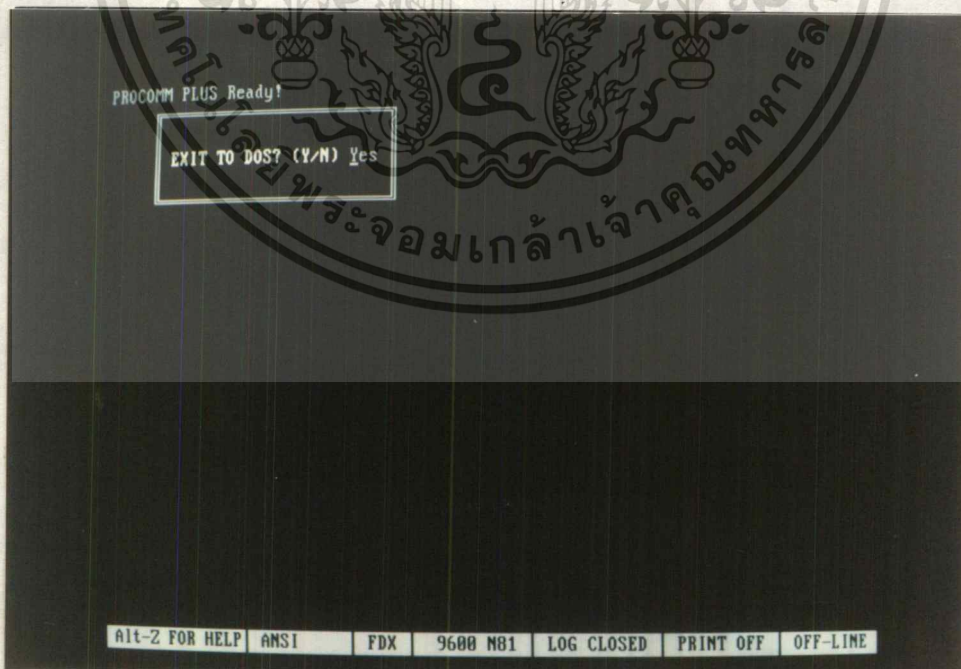
รูปที่ 4.23 การรับข้อมูลของคอมพิวเตอร์ จากการส่งข้อมูลมาจากเครื่องวิทยุแพคเกจ เพื่อเลือกการเก็บข้อมูลไว้ในแผ่นดิสก์หรือพิมพ์ผลผ่านคอมพิวเตอร์ต่อไป



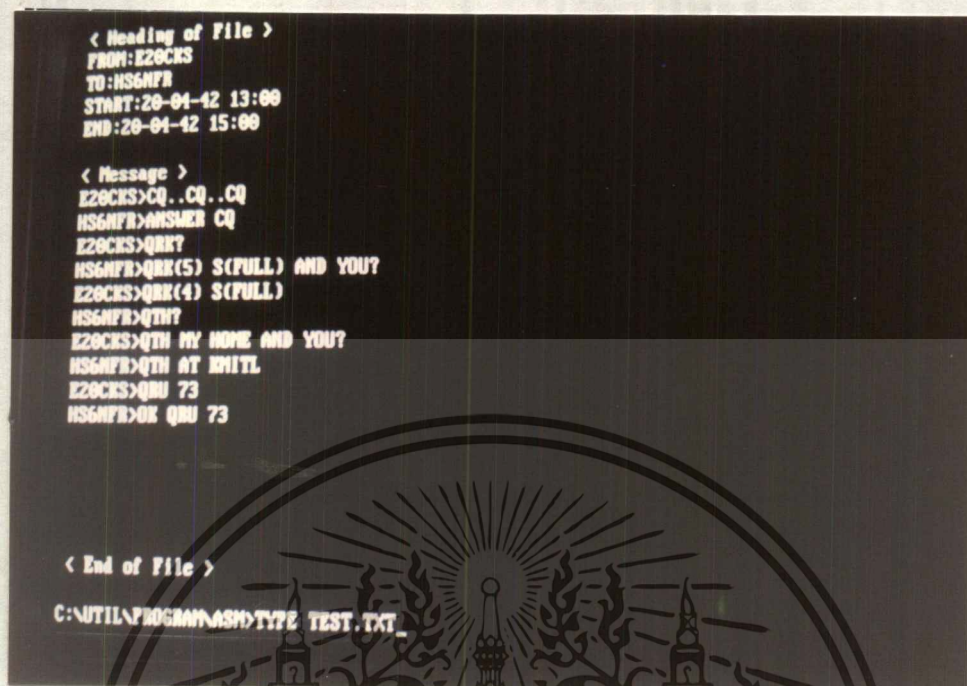
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามเผยแพร่โดยไม่ได้รับอนุญาต และต้องอยู่ภายใต้เงื่อนไขของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.25 การบันทึกชื่อแฟ้มที่ต้องการบันทึกตัวอย่างชื่อไฟล์ TEST.TXT



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
รูปที่ 4.26 การออกสู่โปรแกรม PCP  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.27 การใช้คำสั่งชนิด TEST.TXT เพื่อดูข้อมูลใน Log - Book

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทที่ 5

### บทสรุป ปัญหา แนวทางแก้ไข และพัฒนา

#### 5.1 บทสรุป

จากการทดลองและการวิเคราะห์และการทดลองที่เกิดขึ้น ทำให้ทราบถึงลักษณะประสิทธิภาพในการทำงานที่มีขีดจำกัด ตลอดจนส่วนที่บ่งชี้ให้เห็นถึงแนวทางในการปรับปรุงและพัฒนาต่อไปในอนาคต ซึ่งสามารถได้ข้อสรุปต่อไปนี้

1. ประสิทธิภาพของการรับส่งข้อมูลถือว่าอยู่ในเกณฑ์ดี ซึ่งสามารถรับส่งข้อมูลได้มีประสิทธิภาพประมาณ 90% เกิดความผิดพลาดหรือข้อความไม่สมบูรณ์น้อยมากเนื่องจากมีระบบที่กำหนดการชิงโครน์สที่ดี โดยมีการใส่รหัสป้องกันการเกิดสัญญาณแปลกปลอมถึง 2 ชั้น

2. ระยะทางในการรับส่งข้อมูลข่าวสารนั้นไม่เป็นอุปสรรคเท่าใดนัก แต่ทั้งนี้และทั้งนั้นต้องขึ้นอยู่กับระบบการรับส่งอันได้แก่ ขนาดของเครื่องส่งและสายนำสัญญาณ สายอากาศ เป็นต้น ถ้าเครื่องส่งมีกำลังส่งแรง สายนำสัญญาณมีการลดทอนน้อยสายอากาศมีอัตราขยายที่ดี และสภาพอากาศเอื้ออำนวย ก็สามารถติดต่อสื่อสารได้ โดยประสิทธิภาพสูงสุดที่จะเป็นไปได้จะใกล้เคียงกับการติดต่อสื่อสารระบบวิทยุสมัครเล่นทั่วไป

3. ความไวในการรับอยู่ในช่วงที่เร็วพอประมาณ ไม่ช้าจนเกินไปตามการออกแบบระบบไว้ที่อัตราบอดเรทในการรับส่งข้อมูล 1,200 bps

4. การติดต่อสื่อสารระบบเชื่อมโยงกับเครื่องคอมพิวเตอร์ได้ผลเป็นที่น่าพอใจมีประสิทธิภาพมาก โดยอ้างอิงการทดลองรับส่งข้อมูลหรือระบบถ่ายโอนข้อมูล Log Book โดยโปรแกรม PROCOM ทั่วไป มีการใช้งานที่ง่ายมาก สามารถพิมพ์ผลที่ต้องการออกมาได้ การติดตั้งง่ายไม่เกิดความซับซ้อน

5. การทดสอบการติดต่อรับส่งข้อมูลกับระบบแพคเกจรีโอ ระบบความถี่ในย่านความถี่วิทยุสมัครเล่น 144-146 MHz ไม่เป็นผลทั้งนี้เนื่องจากการที่ระบบทั้งสองมีการเขียนโปรแกรม เพื่อควบคุมการติดต่อสื่อสารไม่ตรงกันทั้งทางด้านการถอดรหัสที่ไม่สามารถถอดรหัสเพื่อให้ใช้ร่วมกันได้ เนื่องจากผู้เขียนโปรแกรมเป็นชาวต่างชาติ ทำให้ไม่สามารถทดลองแก้ไขโปรแกรม ให้ใช้งานร่วมกันได้ อีกทั้งข้อจำกัดในหน่วยความจำของเครื่องวิทยุแพคเกจรีโอมีขนาดเล็กน้อยทำให้เกิดข้อจำกัด ทางด้านขีดความสามารถในการเขียนโปรแกรมและการเก็บข้อมูลให้มากได้ แต่การทำงานของระบบวิทยุแพคเกจรีโอยังคงติดต่อสื่อสารได้ผลเช่นเดียวกับระบบแพคเกจรีโอ หรือรีโอโมเด็ม ทั่วไป ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6. ลักษณะของการพิมพ์ข้อความอาจยังมีความคล่องตัวน้อย เนื่องจากมีปุ่มเป็นพิมพ์ขนาดเล็กและการเรียงตัวอักษรที่ต่างจากคีย์บอร์ดคอมพิวเตอร์ทั่วไป ทำให้ยังไม่เคยชินอยู่บ้าง แต่การใช้งานยังคงสามารถใช้งานได้ดีอีกทั้งจอแสดงผลที่มีขนาดเล็ก เนื่องจากเป็นเครื่องต้นแบบอาจมีการพัฒนาให้มีขนาดใหญ่รองรับการใช้งานที่สะดวกต่อไปในอนาคตได้

7. วงจรตรวจเช็คสถานะแรงดันไฟสามารถทำงานได้อย่างมีประสิทธิภาพตรงความถี่ได้ออกแบบไว้ทุกประการ

## 5.2 ปัญหา

การจัดทำเครื่องวิทยุแพคเกจ มีอุปสรรคเกี่ยวกับการทำงานดังต่อไปนี้

1. การมอดูเลตสัญญาณแบบ FSK เกิด Bit Error Rate สูง
2. วงจรแพคเกจเรดิโอไม่สามารถเพิ่มความเร็วในการรับส่งข้อมูลให้สูงกว่า 1,200 bps ได้ เนื่องจากใช้อุปกรณ์ไอซี มอดูเลตแบบ FSK ที่สำเร็จรูป (TCM 3105)
3. รูปแบบของโปรโตคอล ในการสื่อสารของแพคเกจเรดิโอที่จัดทำขึ้นนี้ ไม่สามารถติดต่อกับโปรโตคอลของโปรแกรมบายคอม หรือแพคเกจเรดิโอ ตัวอื่นๆ ที่ทำงานบนเครื่องคอมพิวเตอร์ทั่วไปได้
4. ในขณะที่มีการสื่อสารด้วยระบบวิทยุแพคเกจเรดิโออยู่นี้ เมื่อมีสถานีอื่นทำการสื่อสารระหว่างกันในระบบเสียงที่ตรงกัน ช่องความถี่ที่เราใช้งานอยู่ และมีความแรงของสัญญาณมากกว่าคู่สถานีของเราจะทำให้เกิดการรบกวนกัน ไม่สามารถรับข้อมูลที่ติดต่อกันได้
5. ไม่สามารถติดต่อสื่อสารระบบแพคเกจเรดิโอร่วมกับระบบสื่อสารดาวเทียมได้ เนื่องจากมีโปรโตคอลที่แตกต่างกัน และไม่สามารถศึกษาค้นคว้าโปรแกรมในการถอดรหัสข้อมูลที่ส่งมาจากดาวเทียมได้

## 5.3 แนวทางแก้ไข

1. ทำการใช้ซอฟต์แวร์ สร้างรหัสไคด์ ที่กลุ่มข้อมูลก่อนส่งออกอากาศ
2. ถ้าหากต้องการลด Bit Error Rate ให้ต่ำลง และเพิ่มความเร็วในการส่งข้อมูล ควรจะทำการเปลี่ยนการมอดูเลตให้เป็นแบบ PSK ,QPSK และ QAM แต่อาจมีความซับซ้อนเพิ่มมากขึ้น
3. แนวทางแก้ไขหรือป้องกันระบบการรบกวนจากคู่สถานีอื่น อาจทำการส่งด้วยกำลังที่สูงขึ้น เพื่อเพิ่มประสิทธิภาพของสายอากาศและสายนำสัญญาณ ตลอดจนจัดช่วงความถี่ที่เหมาะสมในการติดต่อสื่อสารย่านแพคเกจเรดิโอโดยเฉพาะถ้าเป็นไปได้ ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 5.4 ข้อเสนอแนะ

จากการทดลองใช้งานวิทยุแพคเกจควบคุมโดยไม่โครคอนดักเตอร์สำหรับนักวิทยุสมัครเล่น ในการส่งข้อมูลข่าวสารผ่านเครื่องวิทยุสมัครเล่น ย่านความถี่ 144 - 146 MHz ชนิดหนึ่ง ทั้งนี้ และทั้งนั้น ยังคงมีบางส่วนที่อาจมีการปรับปรุง เปลี่ยนแปลงเพิ่มเติม เพื่อให้เกิดเป็นแนวทางในการพัฒนาที่ดีขึ้นได้ในอนาคต ซึ่งได้แก่

1. การใช้เป็นฟิร์มแวร์คอมพิวเตอร์ภายนอกร่วมกับเครื่องวิทยุแพคเกจนี้ได้ โดยอาจมีการเพิ่มพอร์ตเข้าไปมีโปรแกรมควบคุมเพื่อรองรับการใช้งาน ทำให้ผู้ใช้มีทางเลือกประกอบกับความเคยชินกับเป็นฟิร์มแวร์คอมพิวเตอร์ทั่วไป ไม่ต้องสร้างความคุ้นเคยกับเป็นฟิร์มแวร์บนตัวเครื่องมากนัก ทำให้สามารถพิมพ์ข้อความได้สะดวกและรวดเร็วยิ่งขึ้น

2. การพัฒนาการแสดงผลและสามารถพิมพ์ข้อความเป็นอักษรภาษาไทยได้ ช่วยให้เพิ่มความน่าสนใจ การใช้งานที่สะดวกต่อคนที่ถนัดภาษาไทยมากกว่าภาษาอังกฤษ เป็นการเพิ่มทางเลือกให้กับผู้ใช้อีกทางหนึ่งด้วย

3. การเพิ่มปุ่มเป็นพิมพ์พิเศษ เช่น ปุ่มเป็นพิมพ์แสดงรหัส Q - Code ที่นิยมใช้งานให้เลือกใช้ได้ทันที โดยไม่ต้องเสียเวลาในการพิมพ์ สะดวกรวดเร็ว ยิ่งขึ้น อาจไม่ต้องปรับปรุงที่ฮาร์ดแวร์ แต่อาจเขียนโปรแกรมควบคุมสวิตช์ ให้สามารถใช้งานได้ 2 หน้า ที่ โดยอาศัยปุ่ม Shift เข้าช่วยเป็นต้น

4. เพิ่มประสิทธิภาพในการเก็บข้อมูล ที่สำคัญให้มากกว่าเดิม เช่น ใน Log Book ที่สามารถเก็บเพิ่มได้มากกว่า 10 แฟ้มหรือแต่ละแฟ้มเก็บข้อมูลรายละเอียดได้มากกว่าเป็นต้น โดยมีโหมดชาร์จ์ให้เลือกใช้ทำให้สะดวกไม่ต้องหาเครื่องชาร์จ์ถ่านมาชาร์จ์จนนอกเครื่อง เป็นต้น

5. การประหยัดพลังงานในอนาคต อาจมีโหมด Power Save ให้เลือกใช้กล่าวคือ เมื่อผู้ใช้เข้าสู่โหมด Standby หรือยังไม่มีมีการติดต่อสื่อสารเกิดขึ้น จะให้หน้าจอดับจนกว่าจะมีข้อความส่งเข้ามา เพื่อตัวผู้ใช้ปลดออกสวิตช์ใดๆ ก็ตาม หน้าจอก็จะปรากฏข้อความพร้อมที่จะใช้งานต่อไป เป็นต้น ทำให้เครื่องสามารถใช้งานได้นานขึ้น

6. มีระบบชาร์จ์แบตเตอรี่ในตัว เมื่อแบตเตอรี่หมด ให้สามารถชาร์จ์ได้ โดยมีโหมดชาร์จ์ให้เลือกใช้ ทำให้สะดวกไม่ต้องหาเครื่องชาร์จ์ถ่านมาชาร์จ์จนนอกเครื่อง เป็นต้น

แนวทางทั้งหมดที่กล่าวมาในข้างต้น อาจเป็นแนวในการปรับปรุงเพิ่มประสิทธิภาพการทำงานให้สูงขึ้นในอนาคต แก่ผู้สนใจแต่ต้องมีการศึกษานำความเข้าใจ และศึกษาความเป็นไปได้ควบคู่กันไปด้วย จึงจะทำให้สามารถปฏิบัติได้จริงหรือไม่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

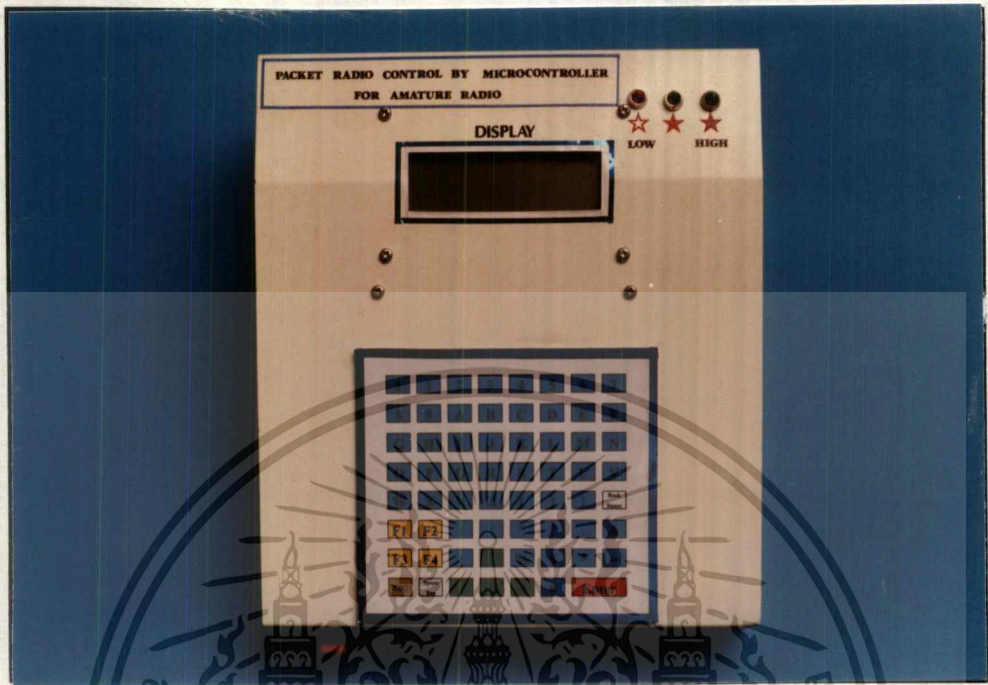


ภาคผนวก ก

**รูปต้นแบบของวิทยุแพกเกจ ควบคุมโดยไมโครคอนโทรลเลอร์**

**สำหรับวิทยุสมัครเล่น**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

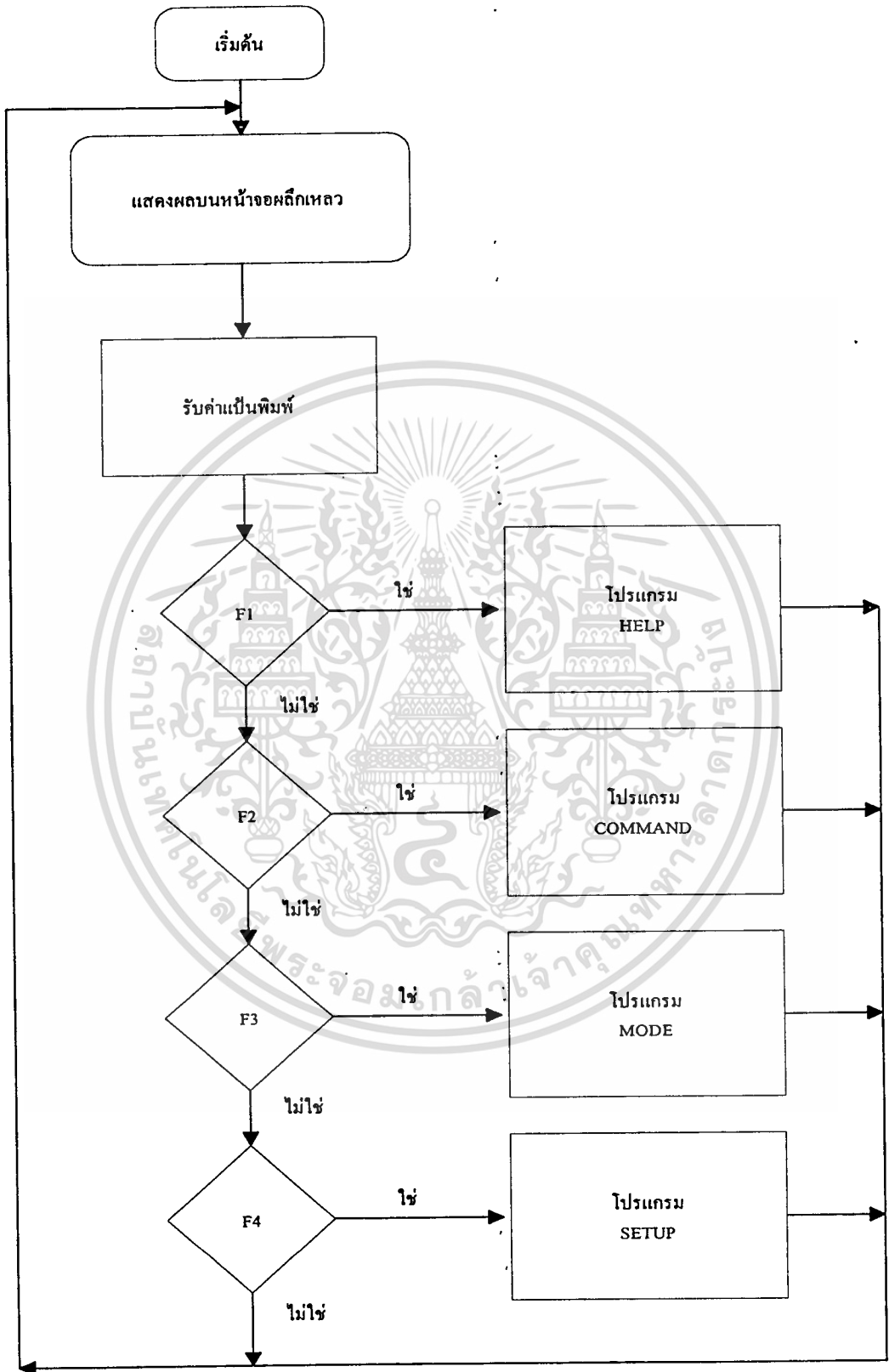


รูปที่ ก.1 รูปต้นแบบของวิทยุแพคเกจ ควบคุมโดยไมโครคอนโทรลเลอร์ สำหรับวิทยุสมัครเล่น

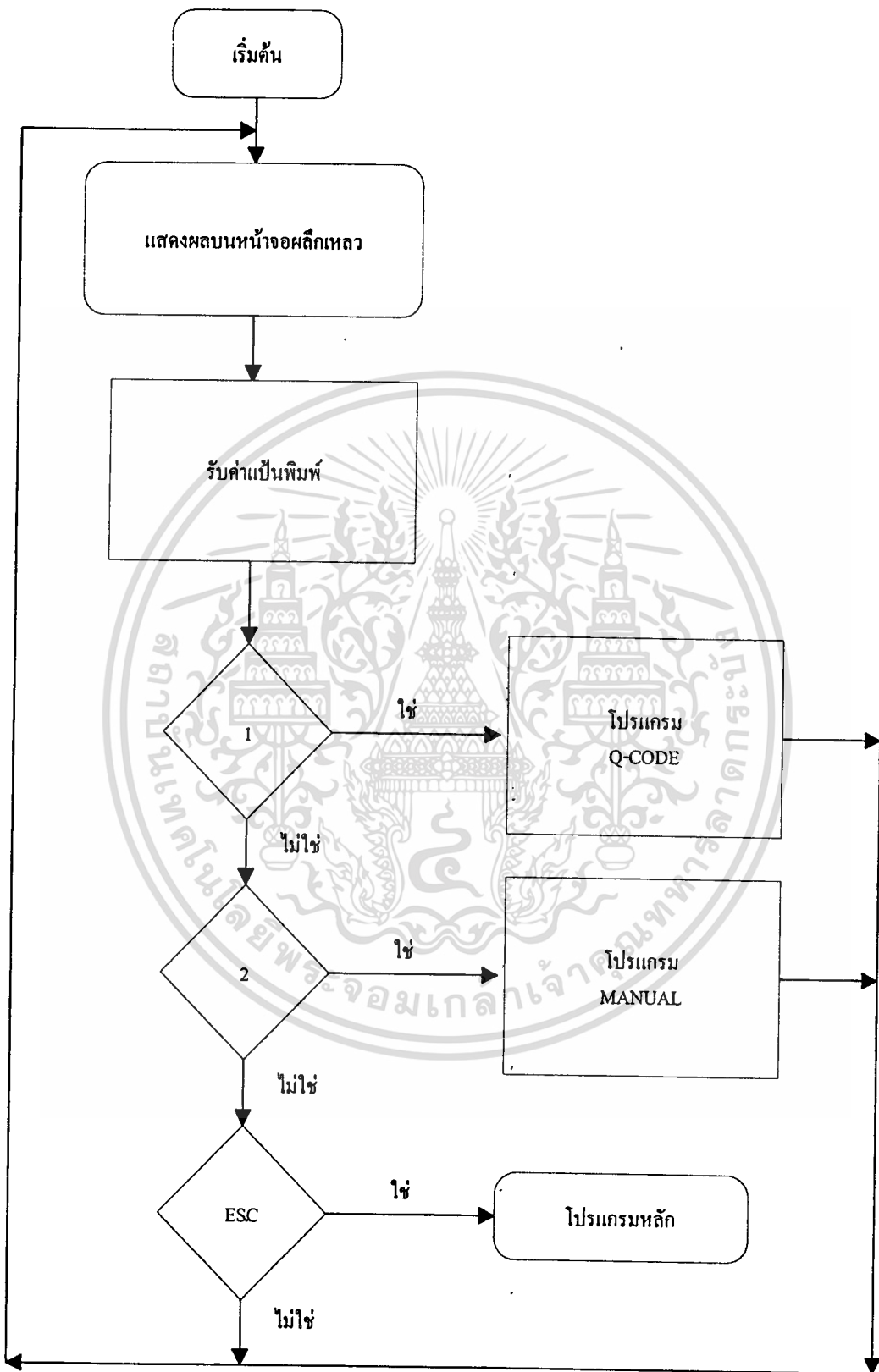
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



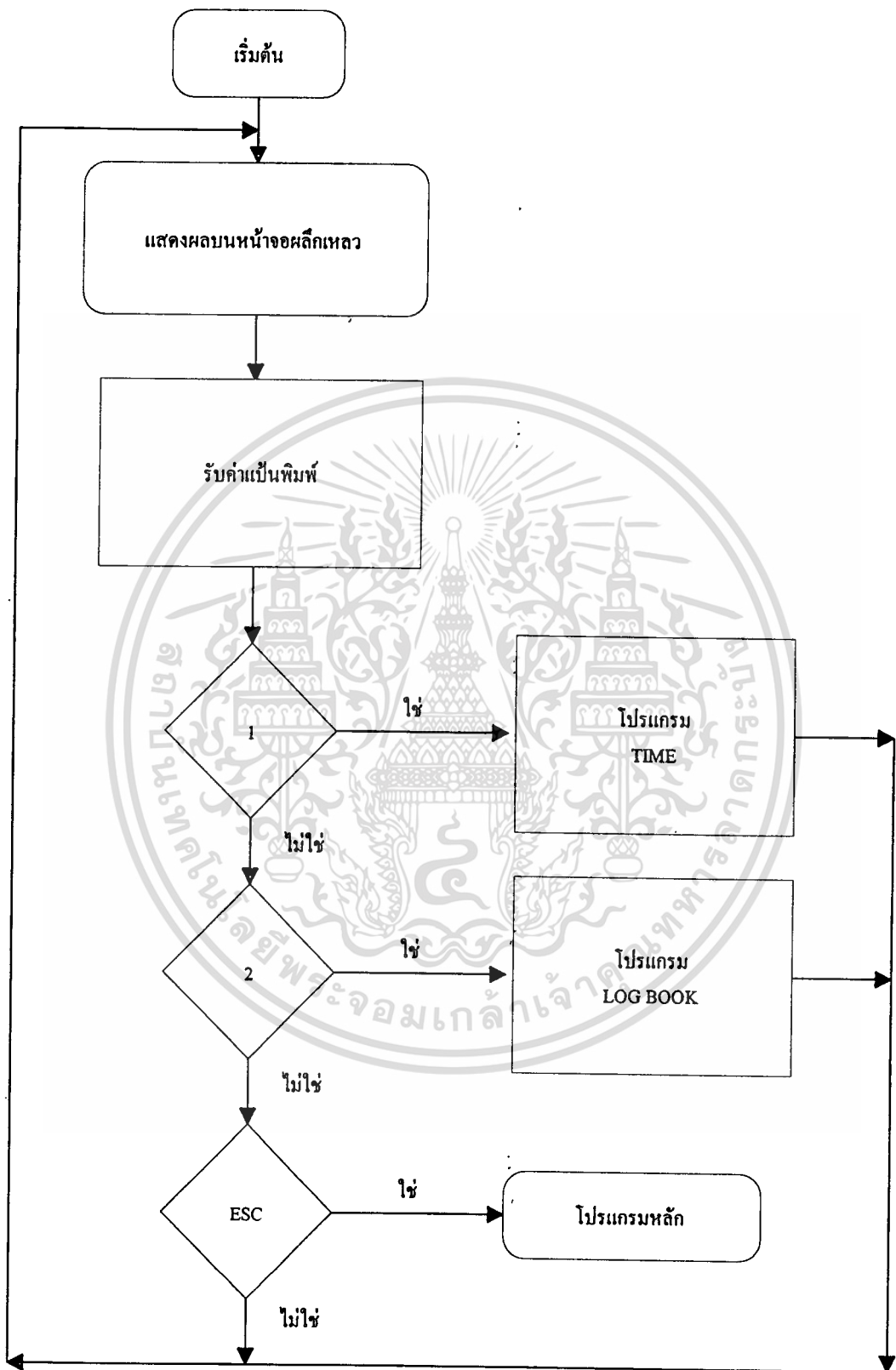
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



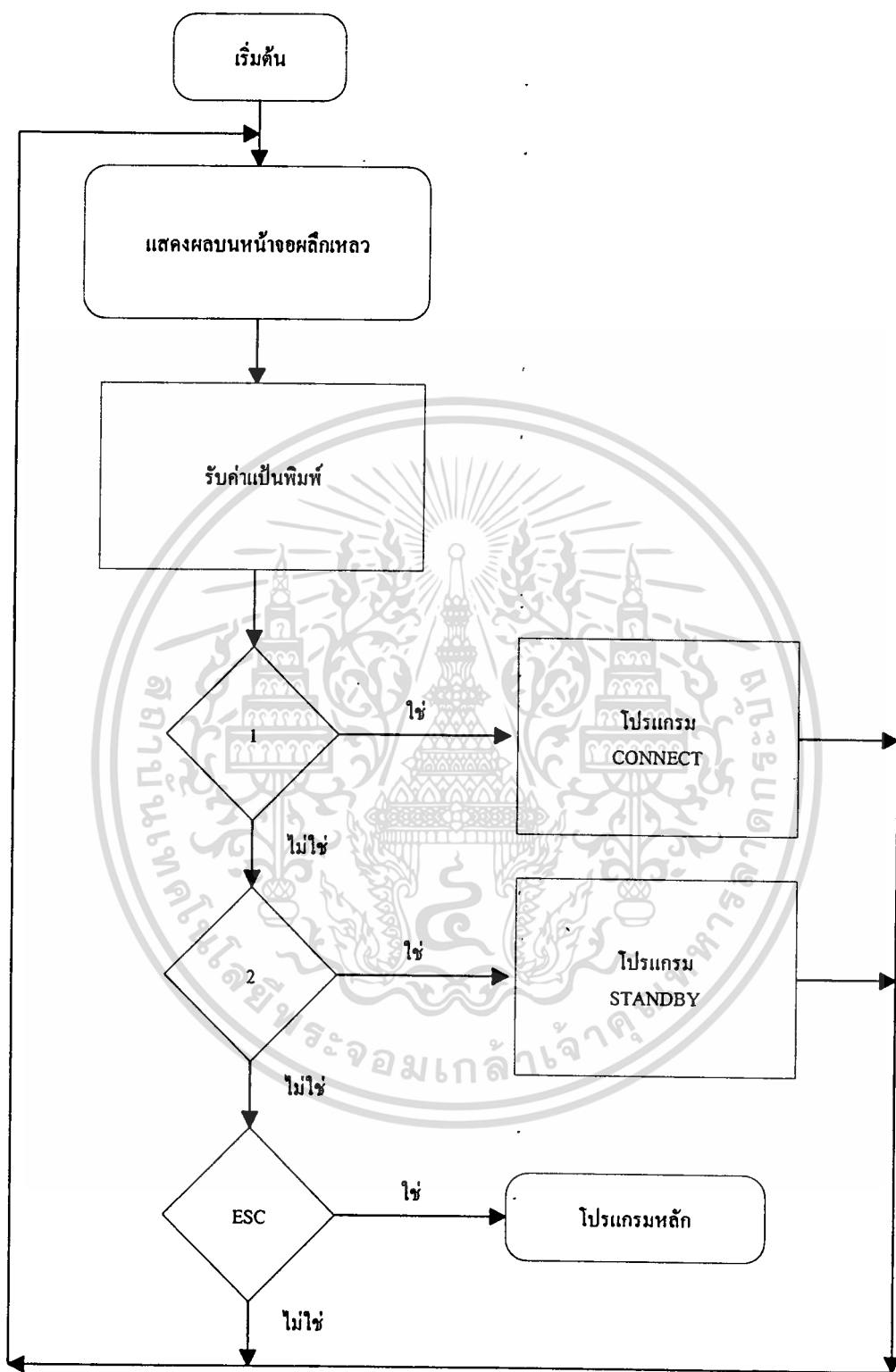
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้าม **รูปที่ ข.1** แผนผังการทำงานของโปรแกรมหลัก



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งยังสงวนสิทธิ์ในเนื้อหาและรูปแบบของโปรแกรม Help เอกสารทุกครั้งที่มีการนำไปใช้

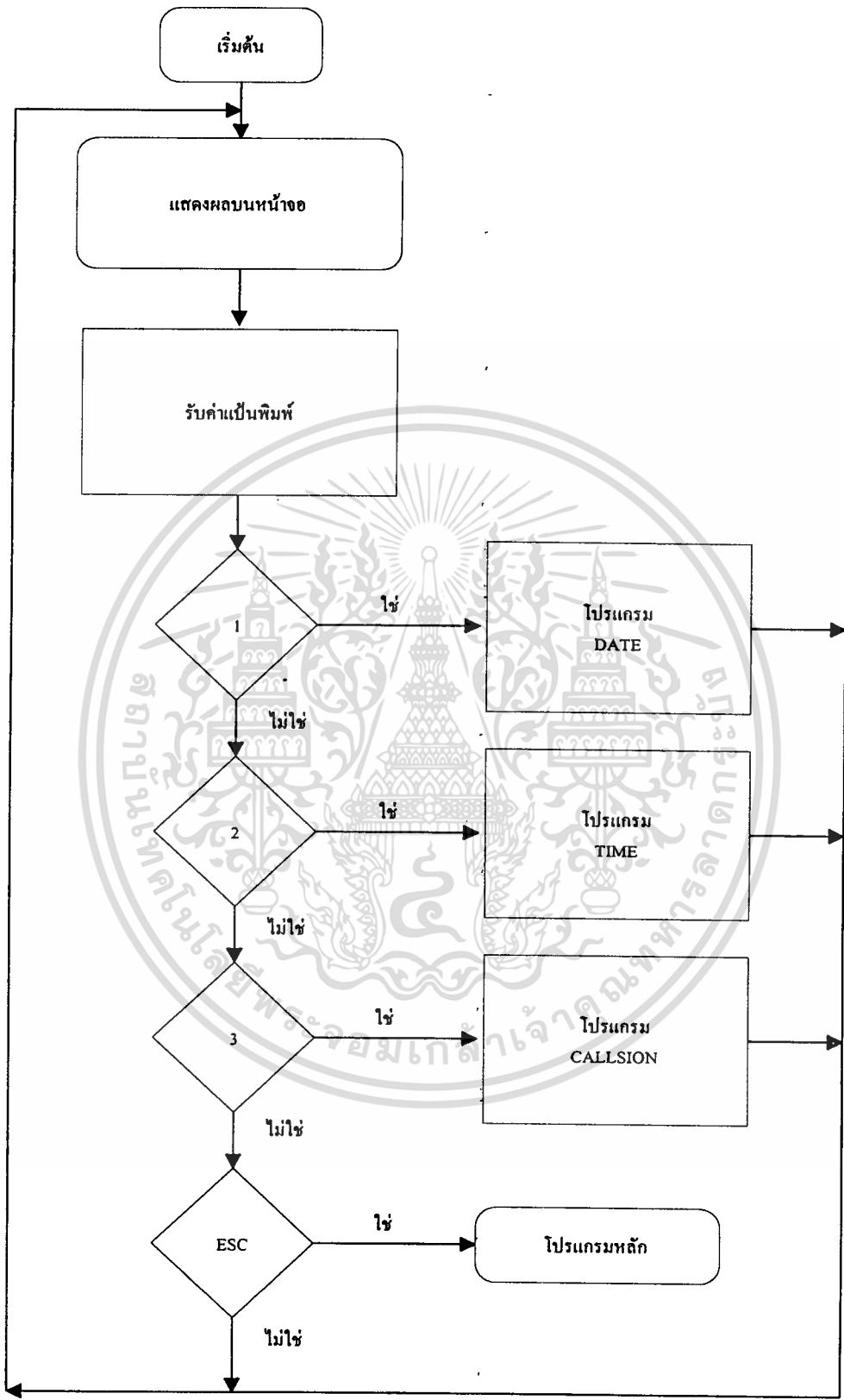


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 รูปที่ ข.3 แผนผังการทำงานของโปรแกรม Command  
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

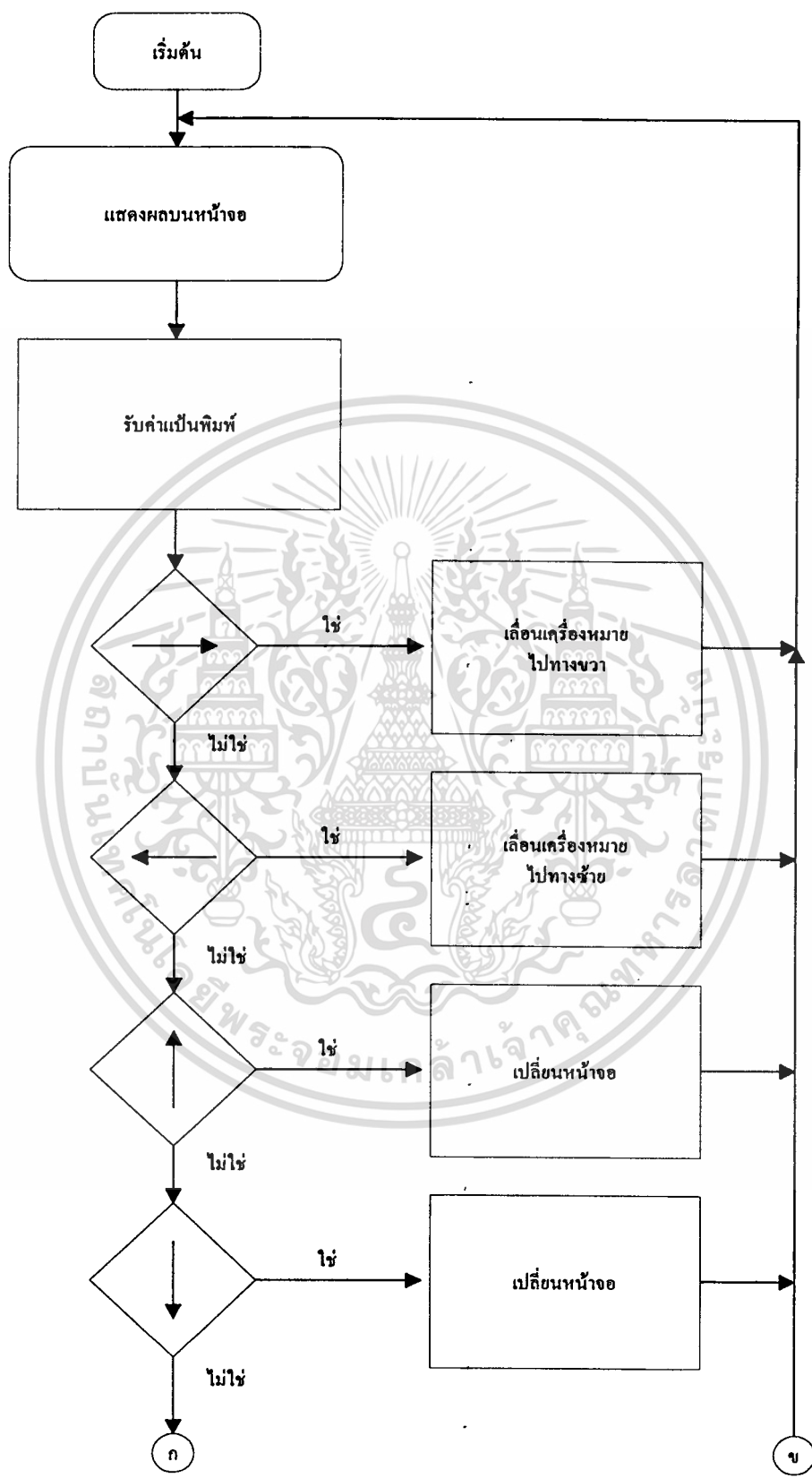


รูปที่ ข.4 แผนผังการทำงานของโปรแกรม Mode

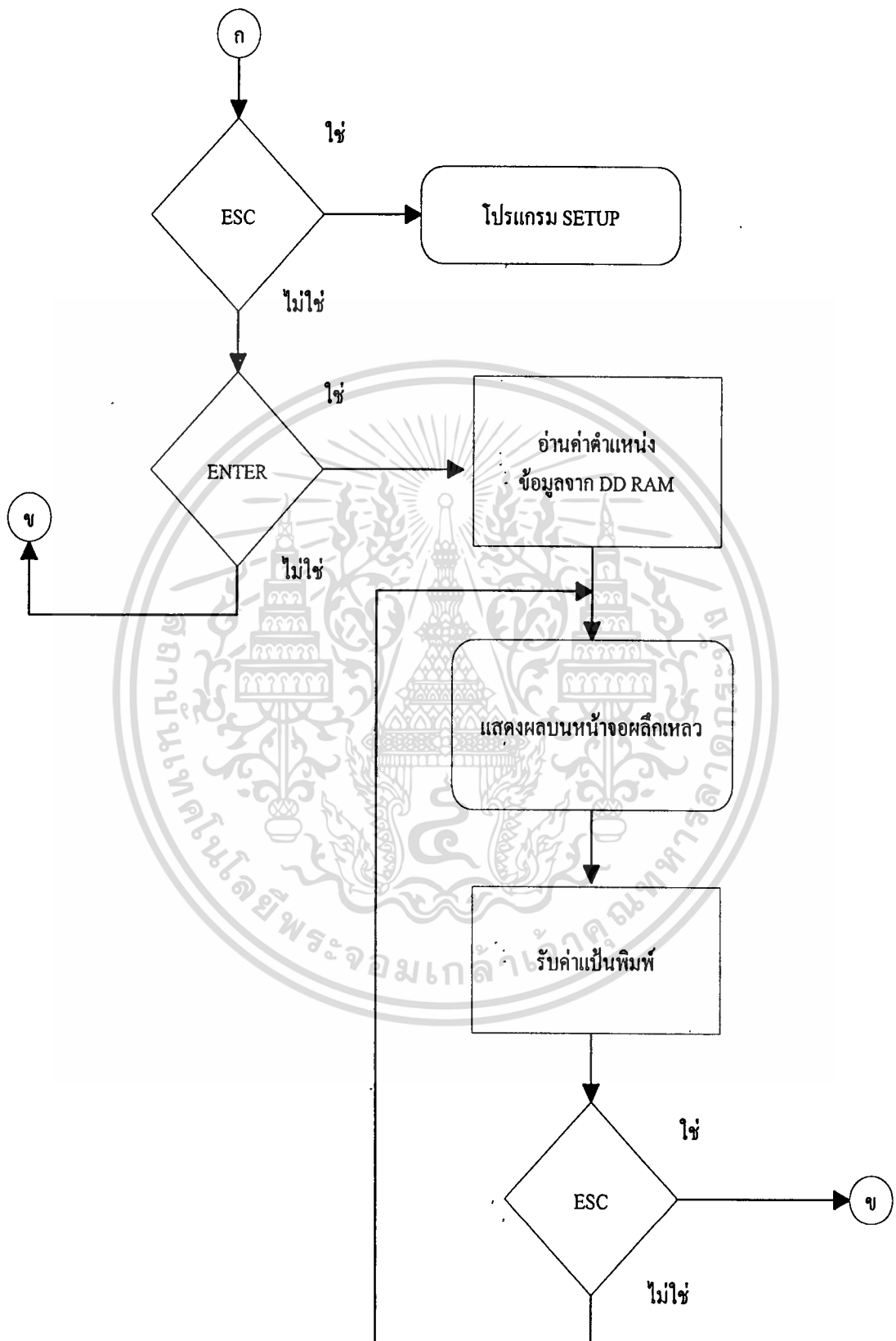
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
รูปที่ ข.5 แผนผังการทำงานของโปรแกรม Setup  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

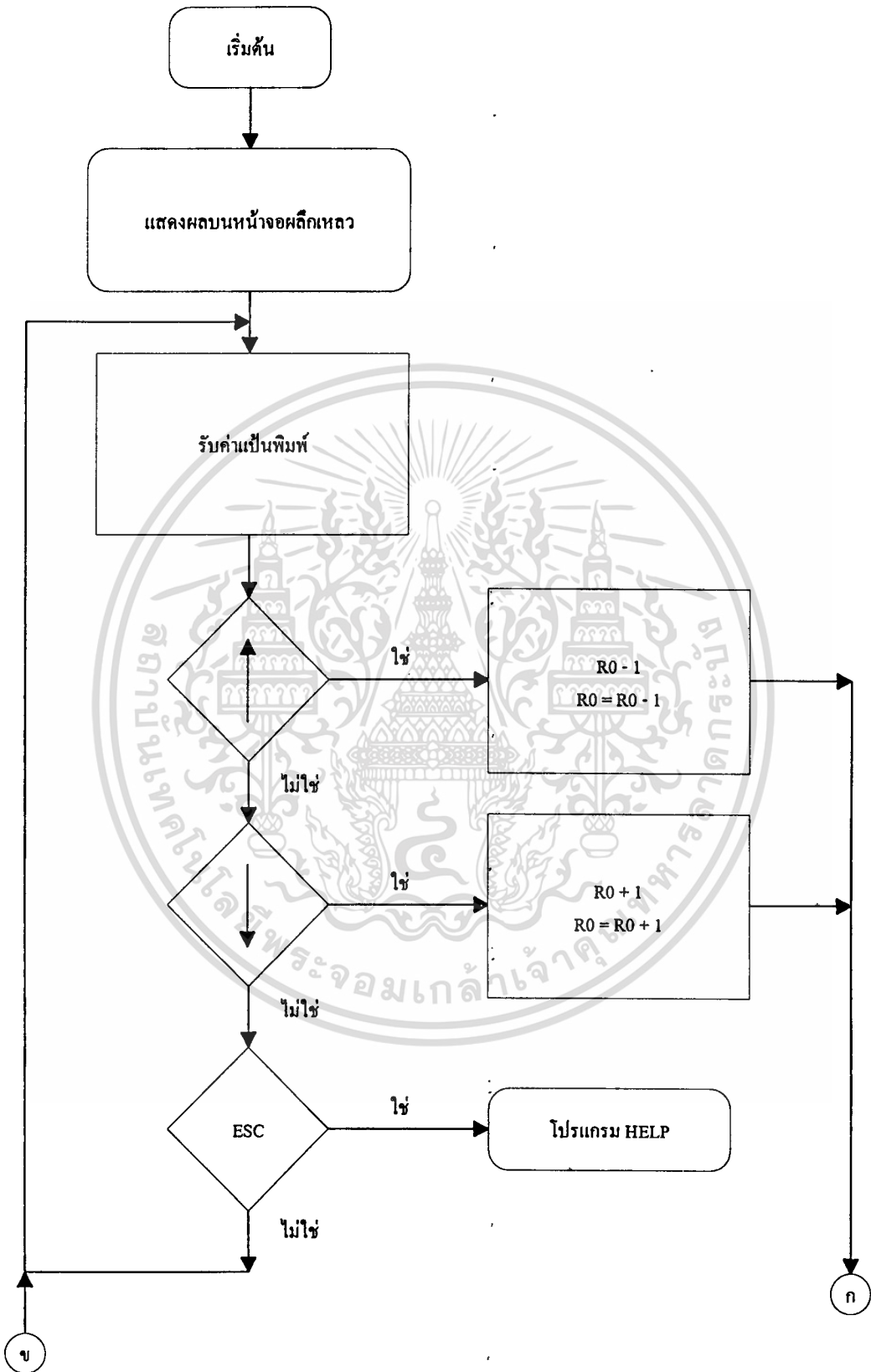


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
รูปที่ ข.6 แผนผังการทำงานของโปรแกรม Q - Code  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งยังมีเหตุที่เปลี่ยนแปลงเนื้อหา และต้องอยู่ภายใต้เงื่อนไขของเอกสารทุกครั้งที่มีการนำไปใช้



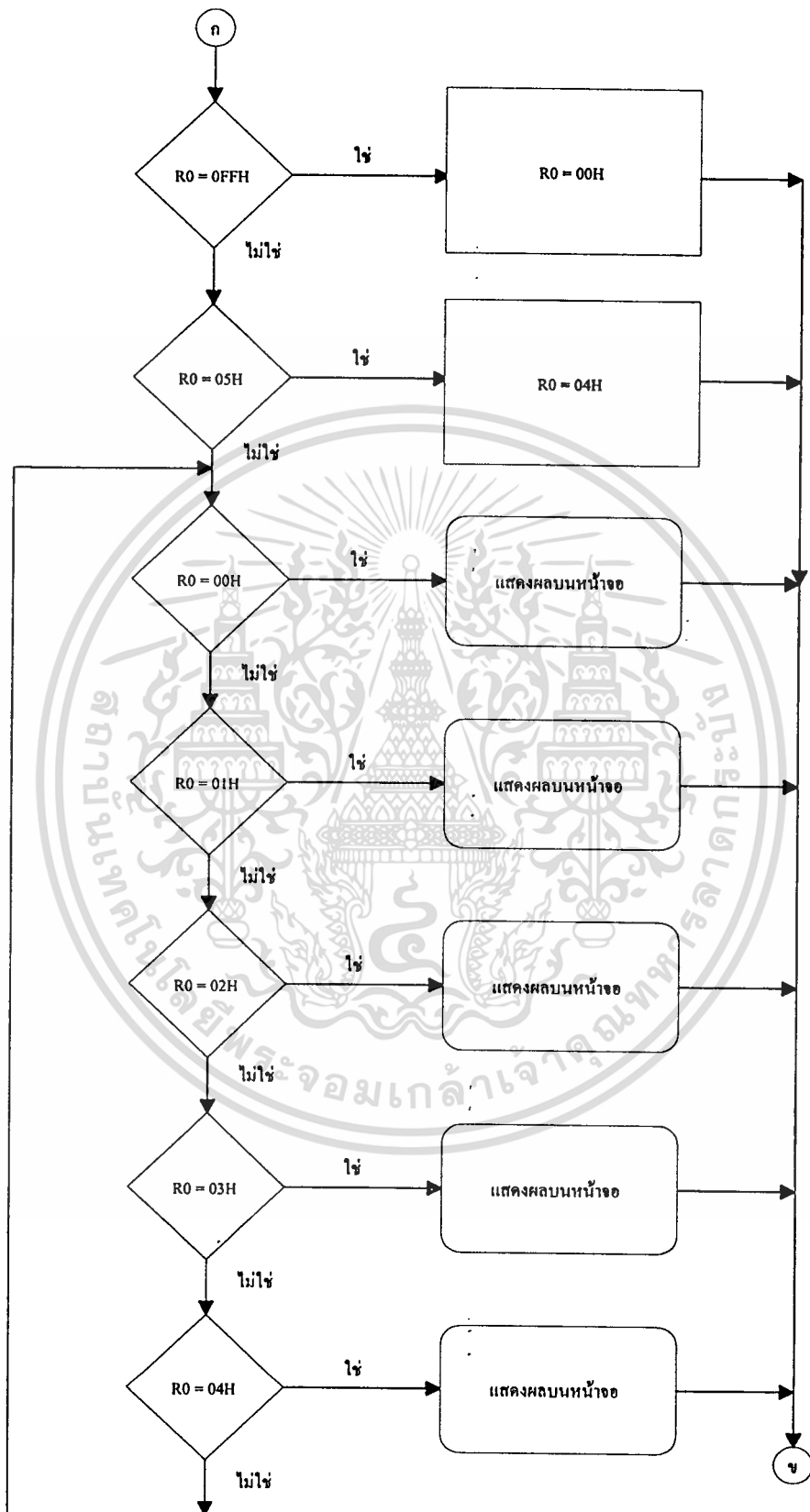
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งยังมิให้คัดลอกข้อมูลและตั้งชื่อเรียกเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ ข.6 (ต่อ) แผนผังการทำงานของโปรแกรม Q - Code



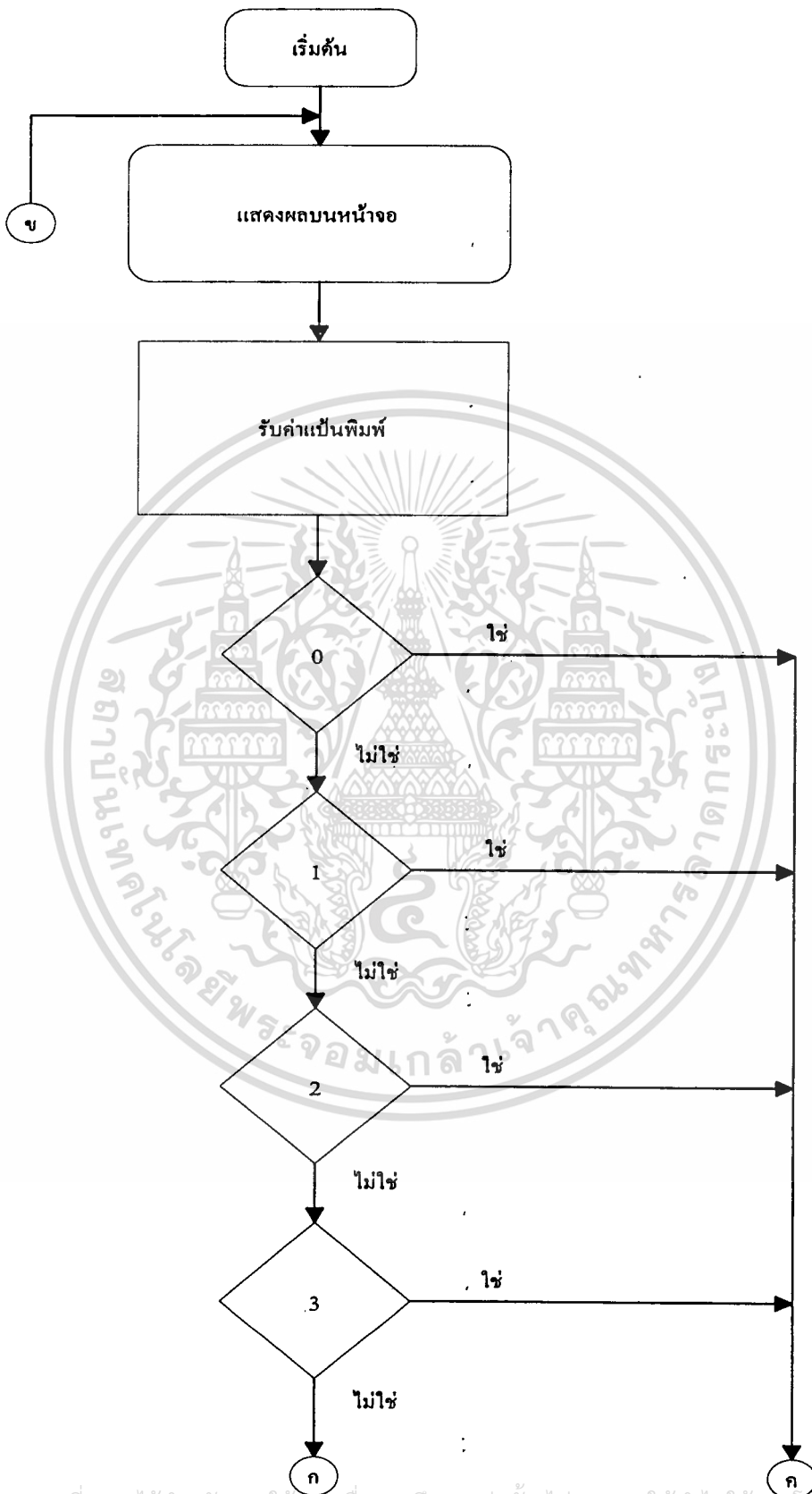
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกครั้งที่ไม่อนุญาตให้เผยแพร่ข้อมูลใดๆ ของเอกสารนี้ไปใช้

รูปที่ ข.7 แผนผังการทำงานของโปรแกรมย่อย Introduction



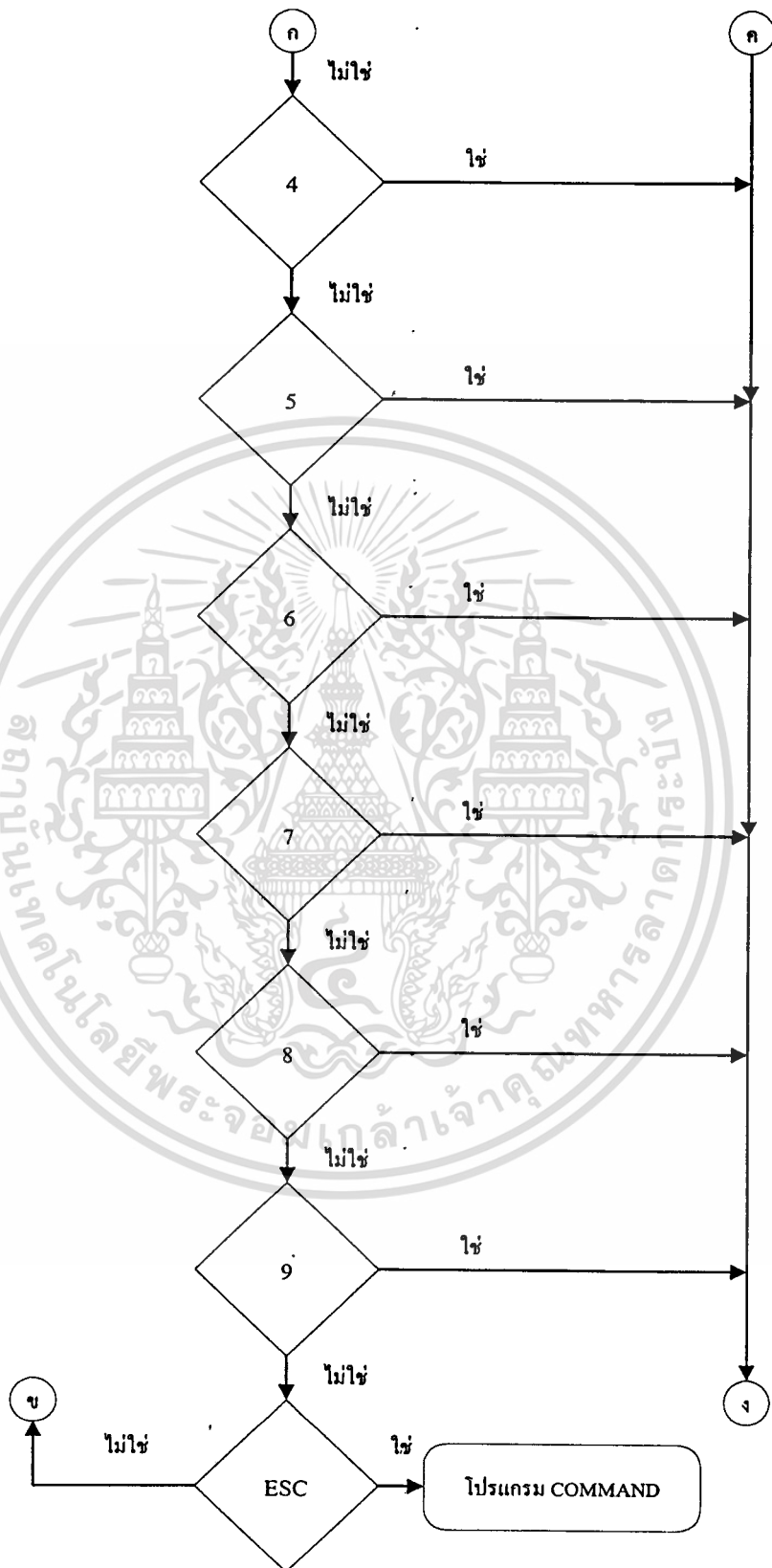
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น ถือว่าผู้จัดทำเอกสารนี้ขอสงวนสิทธิ์ในเนื้อหาและข้อมูลอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ ข.7 (ต่อ) แผนผังการทำงานของโปรแกรมย่อย Introduction

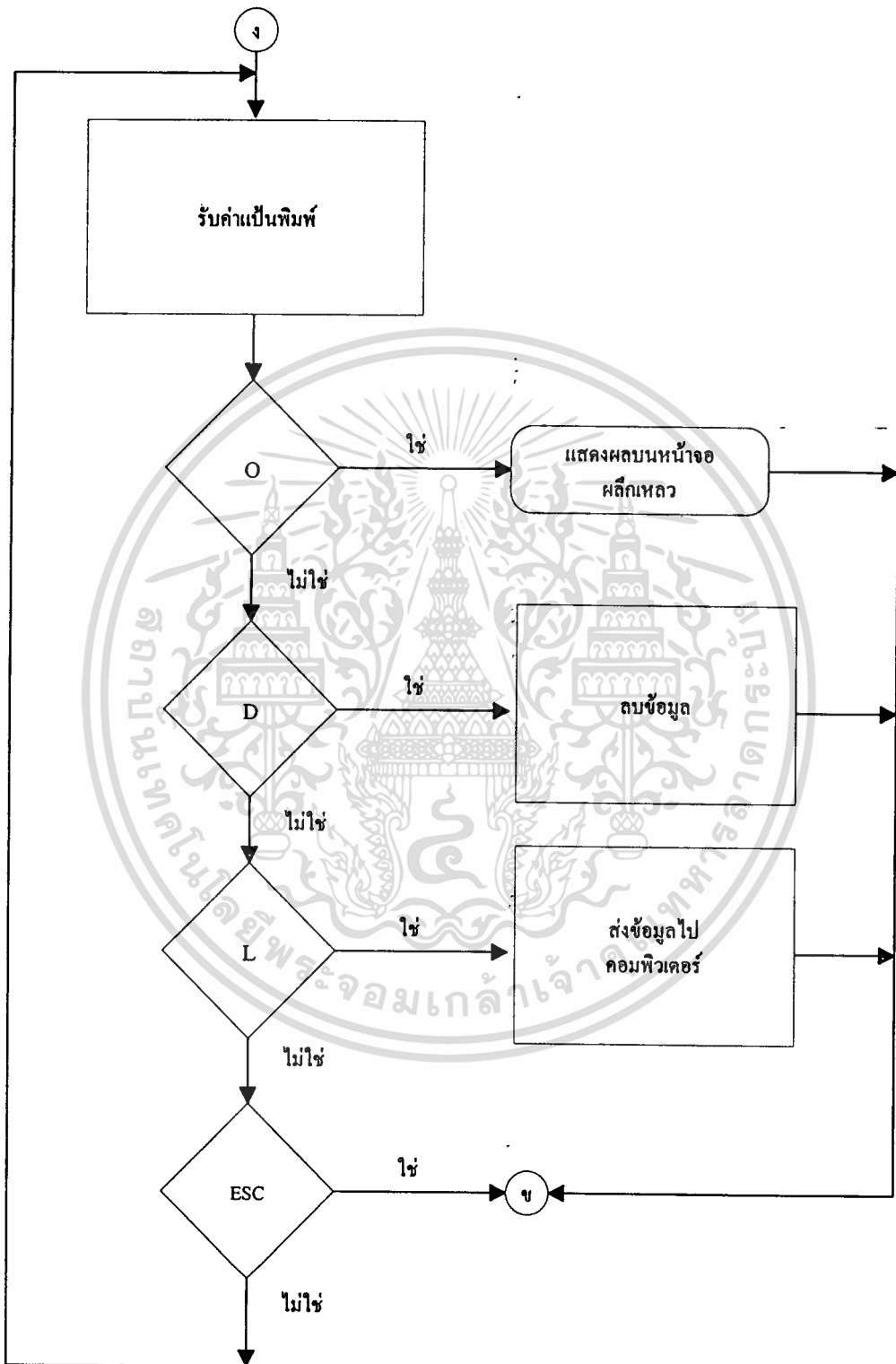


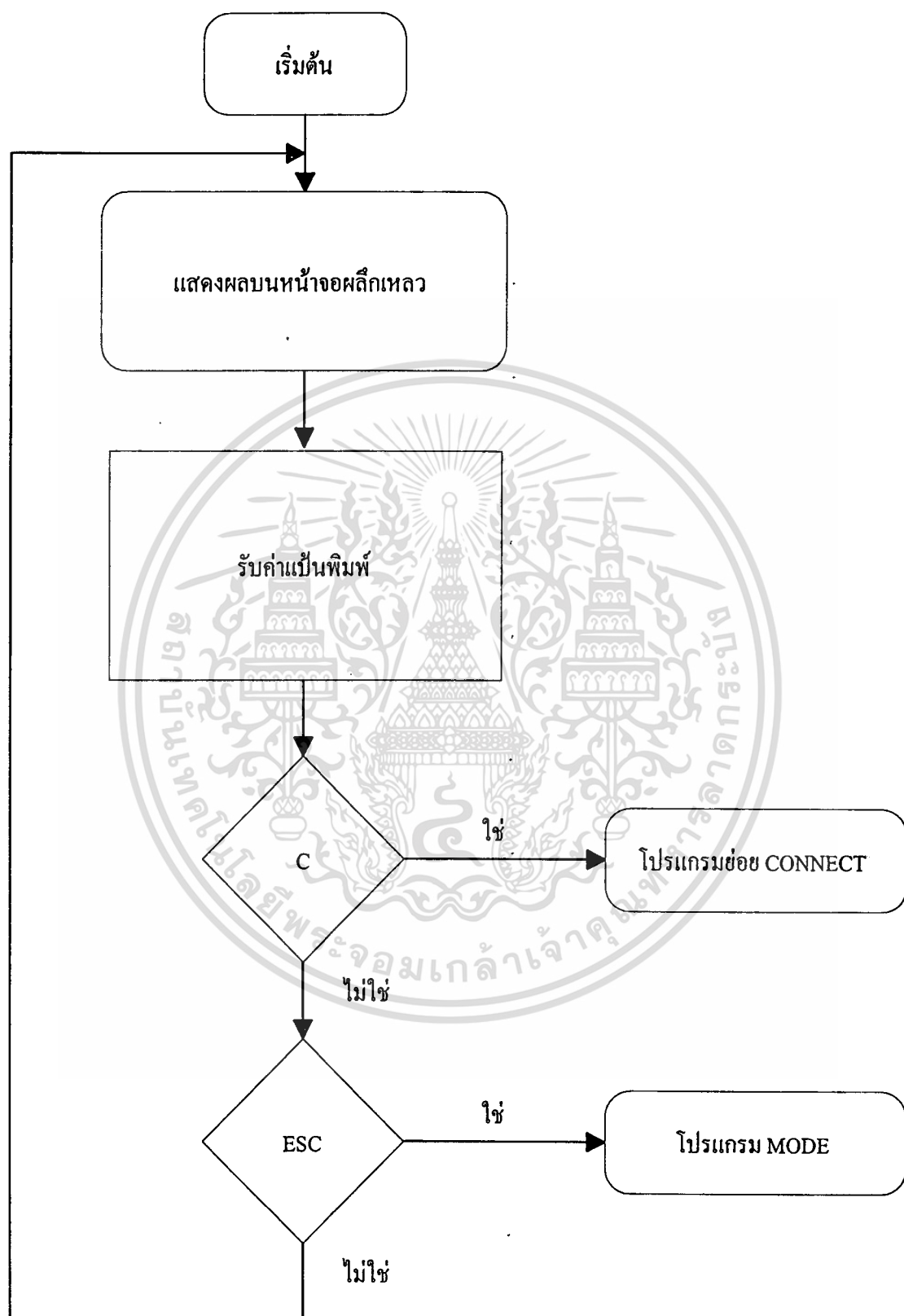
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งยังมิให้ดัดแปลงเนื้อหาและตัวอย่างอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ ข.8 แผนผังการทำงานของโปรแกรมย่อย Log Book

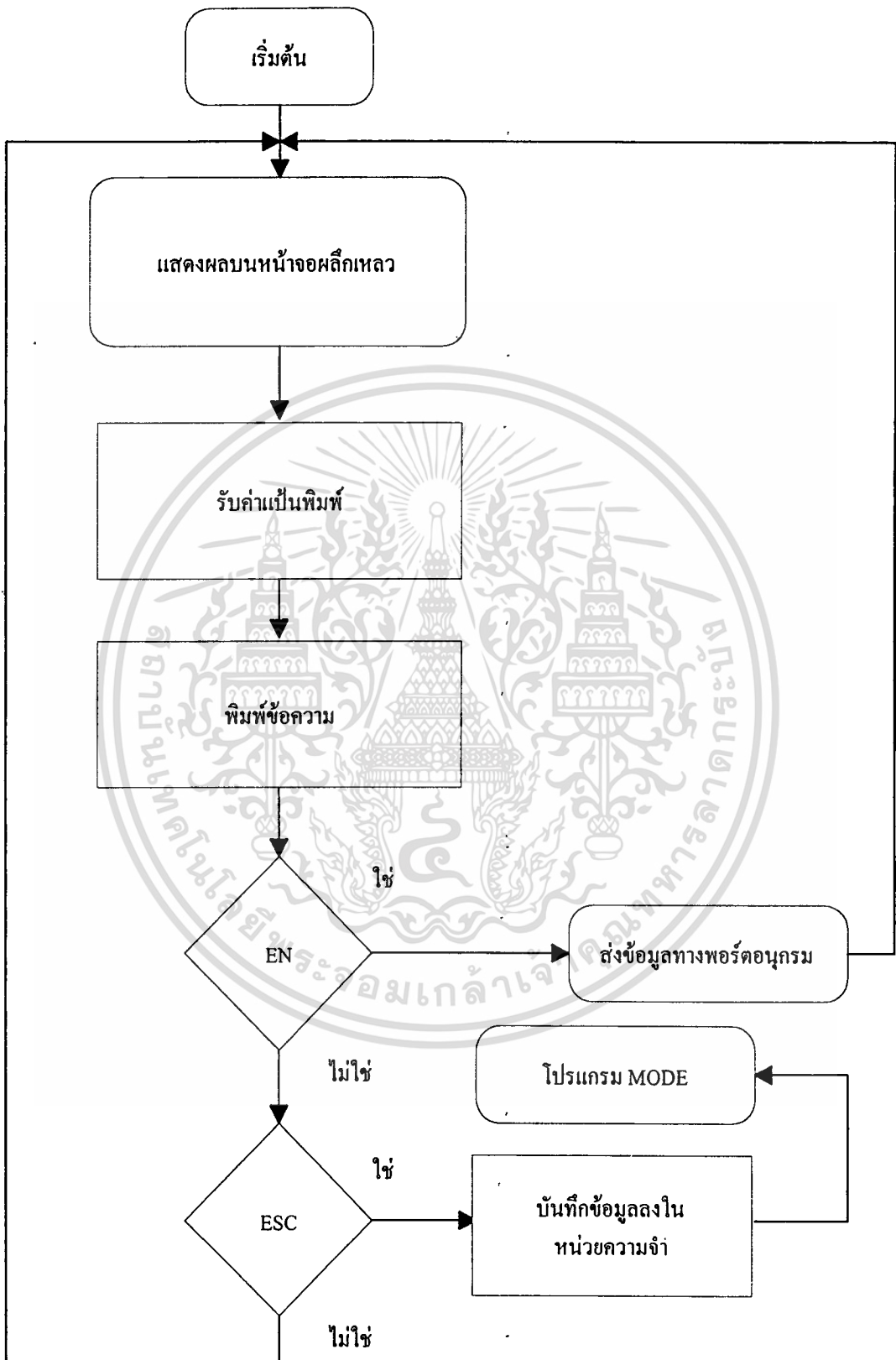


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น **รูปที่ ข.8 (ต่อ) แผนผังการทำงานของโปรแกรมย่อย Log Book** ทุกครั้งที่มีการนำไปใช้

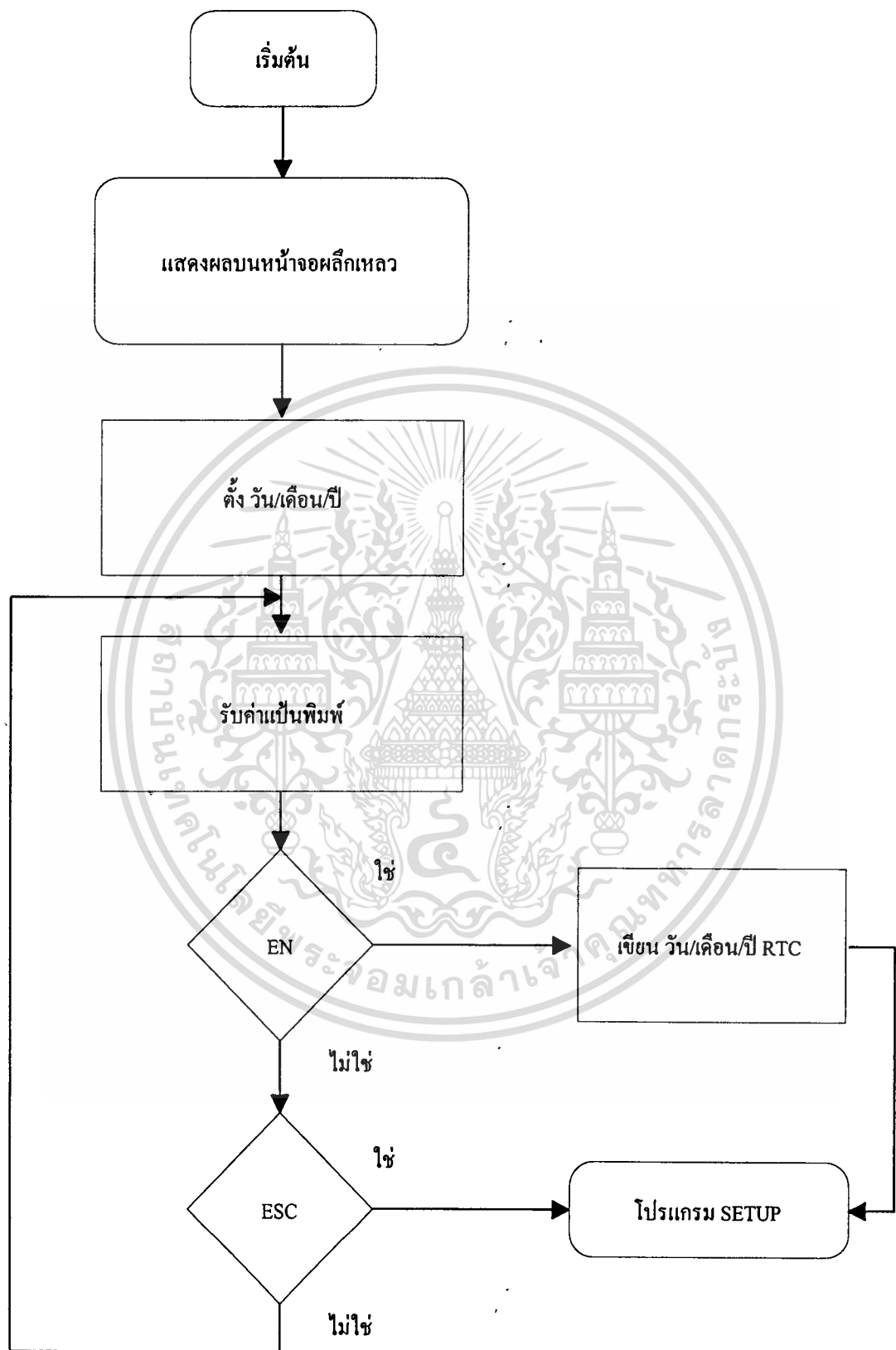




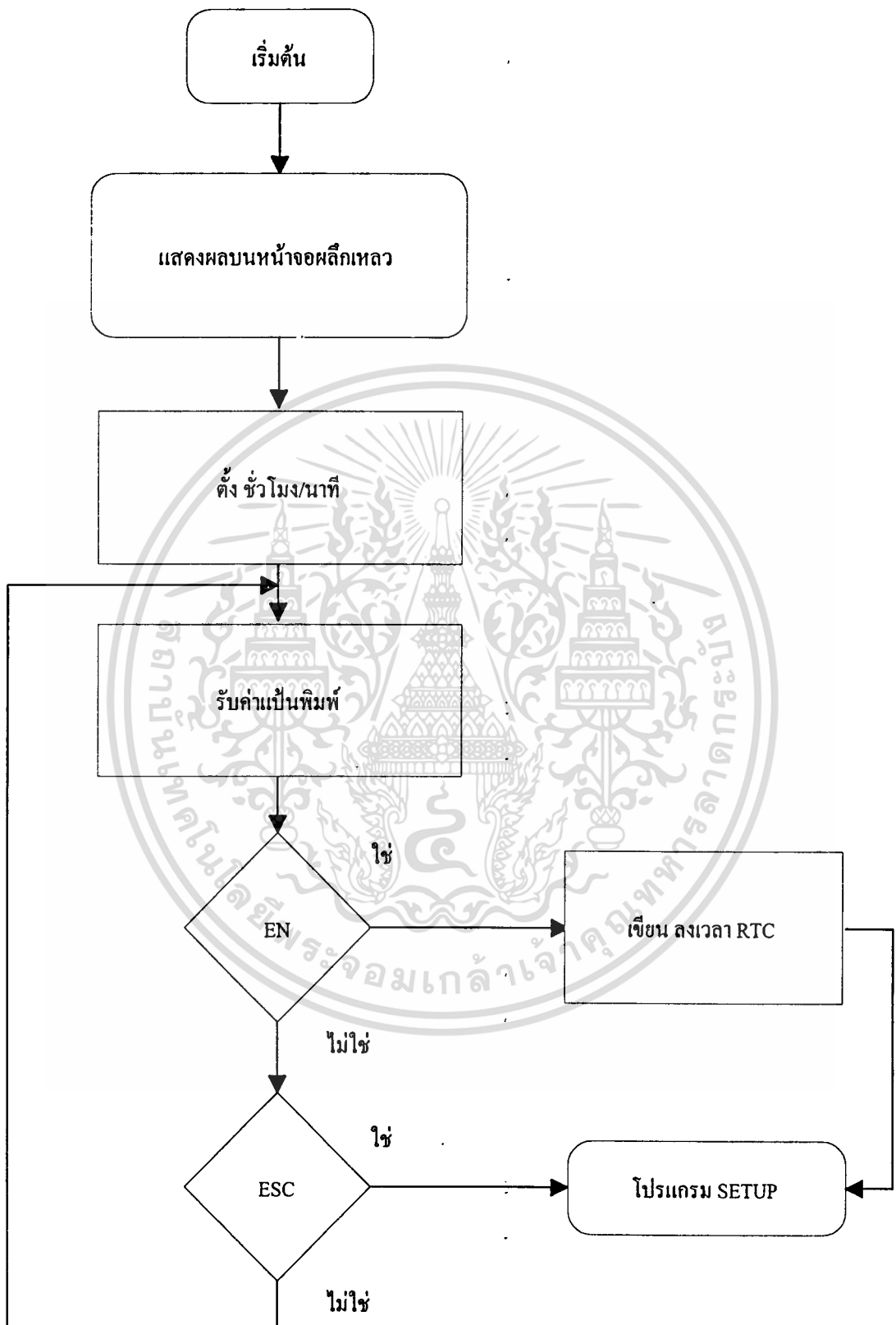
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้ง **รูปที่ ข.9** แผนผังการทำงานของโปรแกรมย่อย Standby ทุกครั้งที่มีการนำไปใช้



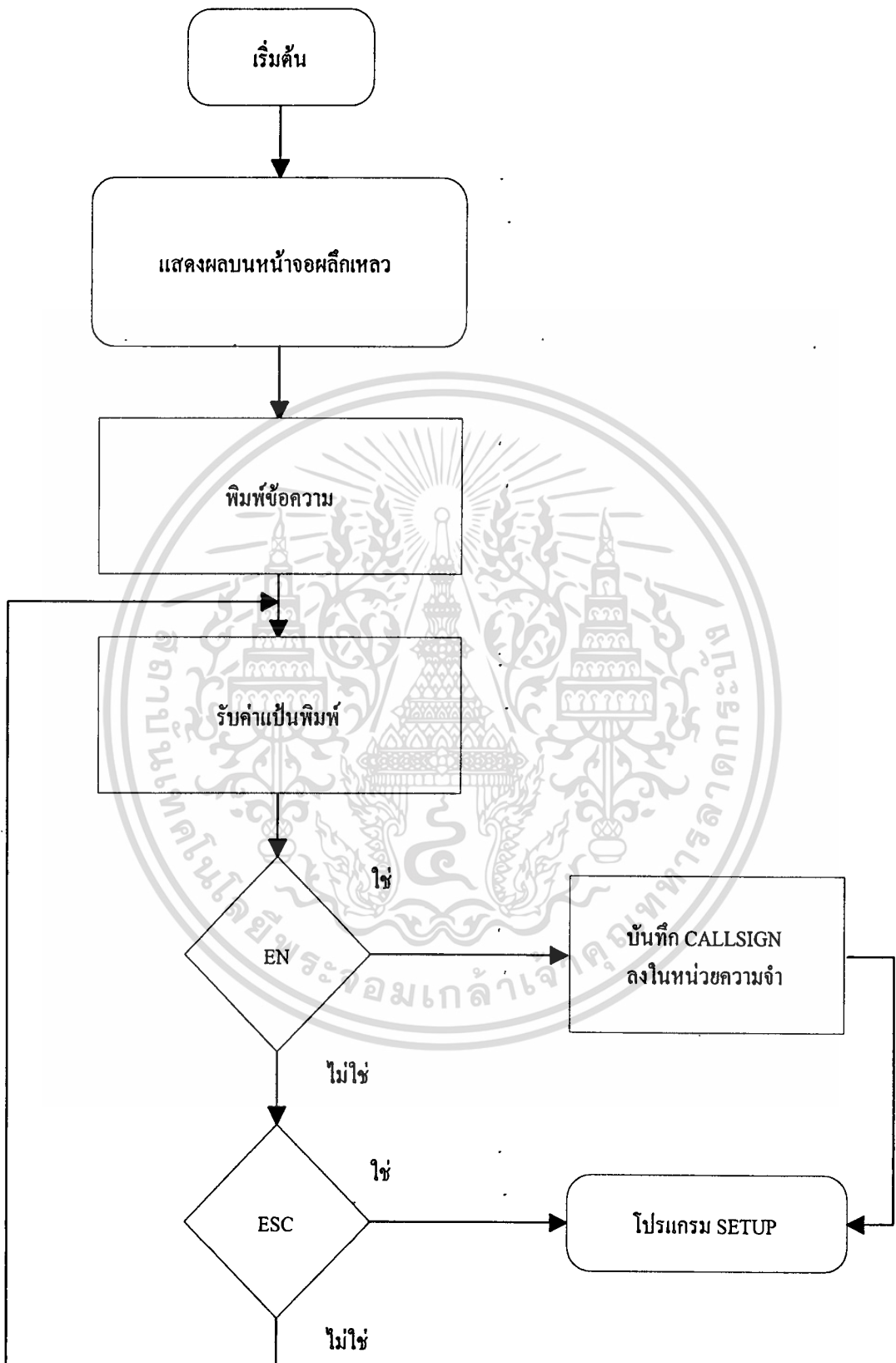
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งรูปที่ ข.10 แผนผังการทำงานของโปรแกรมน้อย Connect ทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้ง **รูปที่ ข.11** แผนผังการทำงานของโปรแกรมย่อย Date การทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งยังไม่มีเหตุเปลี่ยนแปลงเนื้อหา และต้องอยู่ใต้อำนาจของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งรูปที่ ข.13 นี้เป็นผังการทำงานของโปรแกรมย่อย Callsing ทุกครั้งที่มีการนำไปใช้

```

; EDIT VERSION 3.0
; SAVE MESSAGE TX&RX
; CLOCK NO SWAP ":", ""
; PROGRAMER BY.MR.NARET SUKROGNCHARNG ED.19
;-----
; 8255
PORTA EQU 0E0E0H ; PORT A 8255
PORTB EQU 0E0E1H ; PORT B 8255
PORTC EQU 0E0E2H ; PORT C 8255
CONTROL EQU 0E0E3H ; CONTROL PORT 8255
;-----
; LCD
COMMAND EQU 0E0C0H ; READ-WRITE REGISTER
READBUSY EQU 0E0C1H ; READ BF AND ADDRESS
WRITEDATA EQU 0E0C2H ; WRITE CHARACTER
READDATA EQU 0E0C3H ; READ DATA FROM DD RAM
BF EQU ACC.7 ; BIT BUSY FLAG
;-----
; RTC
RST EQU P1.5
DATA_IO EQU P1.6
CLK EQU P1.7
COL1 EQU 4FH
COL2 EQU 50H
;-----
; DATA TRANFER (RTC)
COMMAND_RTC EQU 08H ; WRITE TO RTC
OUTPUT_RTC EQU 09H ; READ DATA FROM RTC
DATA_RTC EQU 0AH ; DATA
ORG 0000H
LJMP MAIN
;-----
; ISR
ORG 0003H ; INT0
ORG 000BH ; TIMER0
ORG 0013H ; INT1
ORG 001BH ; TIMER1
ORG 0023H ; SERIAL PORT
LJMP RX
;-----
ORG 0030H
MAIN: MOV SP,#30H
MOV R0,#0
DJNZ R0,$
DJNZ R0,$
CLR P3.4 ; STAN A
MOV 59H,#14H ; STORE ADDRESS DD RAM
MOV DPTR,#CONTROL
MOV A,#90H ; PORT A = INPUT , PORT B C =
OUTPUT
MOVX @DPTR,A
LCALL INIT ; SETUP LCD
LCALL FONT ; ADD FONT
MOV DPTR,#COMMAND
MOV A,#00H
ORL A,#80H
MOVX @DPTR,A
LCALL WAIT
LCALL DISPLAY
;-----
PAGE_3:
MOV DPTR,#DIS_4 ; PAGE 3
LCALL PG_LCD
SCAN_GET: LCALL SCANKEY
CJNE A,#01H,S_1
LJMP PG_1 ; F1 HELP
S_1: CJNE A,#02H,S_2
LJMP PG_2 ; F2 COMMAND
S_2: CJNE A,#03H,S_3
LJMP PG_3 ; F3 MODE
S_3: CJNE A,#04H,SCAN_GET
LJMP PG_4 ; F4 SETUP
PG_1: MOV DPTR,#DIS_5 ; F1 HELP
LCALL PG_LCD
LP_1: LCALL SCANKEY
CJNE A,#27H,LPT_2
LJMP PAGE_3
LPT_2: CJNE A,#31H,LPT_3
LJMP Q_CODE
LPT_3: CJNE A,#32H,LP_1
LJMP INTRO
PG_2: MOV DPTR,#DIS_6 ; F2 COMMAND
LCALL PG_LCD
LP_2: LCALL SCANKEY
CJNE A,#27H,NXT_1
LJMP PAGE_3
NXT_1: CJNE A,#31H,NXT_2
LJMP RTC
NXT_2: CJNE A,#32H,LP_2
LJMP LOG_BOOK
PG_3: MOV DPTR,#DIS_7 ; F3 MODE
LCALL PG_LCD
LP_3: LCALL SCANKEY
CJNE A,#27H,NLP_1
LJMP PAGE_3
NLP_1: CJNE A,#31H,NLP_2
LJMP CONNECT
NLP_2: CJNE A,#32H,LP_3

```

|                      |       |                                      |                                                   |       |                        |
|----------------------|-------|--------------------------------------|---------------------------------------------------|-------|------------------------|
|                      | LJMP  | PAGE_3                               |                                                   | POP   | DPL                    |
| NLP_1:               | CJNE  | A,#31H,NLP_2                         |                                                   | POP   | ACC                    |
|                      | LJMP  | CONNECT                              |                                                   | RET   |                        |
| NLP_2:               | CJNE  | A,#32H,LP_3                          |                                                   |       |                        |
|                      | LJMP  | STANBY                               |                                                   |       |                        |
|                      |       |                                      | ; SCAN KEYBOARD AND CONVER KEYCODE TO ASCII CODES |       |                        |
| PG_4:                | MOV   | DPTR,#DIS_8 ; F4 SETUP               | SCANKEY:                                          | PUSH  | DPH                    |
|                      | LCALL | PG_LCD                               |                                                   | PUSH  | DPL                    |
| LP_4:                | LCALL | SCANKEY                              |                                                   | PUSH  | 00H                    |
| NXP_1:               | CJNE  | A,#27H,NXP_2                         |                                                   | PUSH  | 02H                    |
|                      | LJMP  | PAGE_3                               |                                                   | PUSH  | 05H                    |
| NXP_2:               | CJNE  | A,#31H,NXP_3                         |                                                   | MOV   | R2,#00H                |
|                      | LJMP  | S_DATE                               |                                                   |       |                        |
| NXP_3:               | CJNE  | A,#32H,NXP_4                         | LOOP_SCAN:                                        | MOV   | DPTR,#PORTA            |
|                      | LJMP  | S_TIMER                              |                                                   | MOVX  | A,@DPTR                |
| NXP_4:               | CJNE  | A,#33H,LP_4                          |                                                   | CJNE  | A,#0FFH,SCAN           |
|                      | LJMP  | S_CALLSIGN                           |                                                   |       |                        |
|                      |       |                                      | SCAN_COL:                                         | MOV   | A,R2                   |
| ; INITIALIZATION     |       |                                      |                                                   | MOV   | DPTR,#PORTB            |
| INIT:                | PUSH  | ACC                                  |                                                   | MOVX  | @DPTR,A                |
|                      | PUSH  | DPL                                  |                                                   | MOV   | 52H,R2                 |
|                      | PUSH  | DPH                                  |                                                   | INC   | R2                     |
|                      | MOV   | DPTR,#COMMAND                        |                                                   | SJMP  | LOOP_SCAN              |
|                      | MOV   | A,#38H ; DATA 8 BIT > 2 LINE 5X7 DOT |                                                   |       |                        |
|                      | MOVX  | @DPTR,A                              | SCAN:                                             | MOV   | 55H,A ; STORE A IN RAM |
|                      | LCALL | WAIT                                 | ADDRESS 55H                                       |       |                        |
|                      | MOV   | A,#0CH ; DISPLAY ON/OFF CONTROL      | SCAN_OPEN:                                        | MOV   | DPTR,#PORTA            |
|                      | MOVX  | @DPTR,A                              |                                                   | MOVX  | A,@DPTR                |
|                      | LCALL | WAIT                                 |                                                   | LCALL | DELAY_SW               |
|                      | MOV   | A,#06H ; INCREMENT CURSOR-ENTRY      |                                                   | CJNE  | A,#0FFH,SCAN_OPEN      |
| MODE SET             |       |                                      |                                                   |       |                        |
|                      | MOVX  | @DPTR,A                              |                                                   | MOV   | 53H,#00H               |
|                      | LCALL | WAIT                                 |                                                   | MOV   | R0,#53H                |
|                      | MOV   | A,#01H ; CLEAR DISPLAY               |                                                   | MOV   | 51H,55H ; PUSH 55H =   |
|                      | MOVX  | @DPTR,A                              | ADDRESS 31H ; ROW                                 |       |                        |
|                      | LCALL | WAIT                                 |                                                   | MOV   | A,51H                  |
|                      | POP   | DPH                                  |                                                   | SWAP  | A                      |
|                      | POP   | DPL                                  |                                                   | XCHD  | A,@R0                  |
|                      | POP   | ACC                                  |                                                   | MOV   | 51H,A                  |
|                      | RET   |                                      |                                                   | MOV   | 53H,#00H               |
|                      |       |                                      |                                                   | MOV   | A,52H                  |
| ; BUSY FLAG FROM LCD |       |                                      |                                                   | SWAP  | A                      |
| WAIT:                | PUSH  | ACC                                  |                                                   | XCHD  | A,@R0                  |
|                      | PUSH  | DPL                                  |                                                   | SWAP  | A                      |
|                      | PUSH  | DPH                                  |                                                   | ORL   | A,51H                  |
|                      | MOV   | DPTR,#READBUSY                       |                                                   | MOV   | 54H,A                  |
| READBF:              | MOVX  | A,@DPTR                              | LOOP_A:                                           | MOV   | R5,#64                 |
|                      | JB    | BF,READBF                            |                                                   | MOV   | DPTR,#TABEL_KEY        |
|                      | POP   | DPH                                  |                                                   |       |                        |

เอกสารนี้เป็นเอกสารที่สร้างขึ้นเพื่อสำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 ไม่ว่ากรณีใดๆ ทั้งสิ้น หากท่านมีให้ดัดแปลงเนื้อหาและตัวอย่างอื่นใด กรุณาแจ้งเจ้าของเอกสารให้ทราบเพื่อที่จะมีการนำไปใช้

```

LOOP_S:  CLR    A
          MOVC   A,@A+DPTR
          CJNE   A,54H,NEXT
          SJMP   NEXT_TO

NEXT:    INC     DPTR
          DJNZ   R5,LOOP_S
          SJMP   LOOP_A

NEXT_TO: ; KEY MATCH
          LCALL  CONVERS
          POP     05H
          POP     02H
          POP     00H
          POP     DPL
          POP     DPH
          RET

SCANKEY_P: PUSH  DPH
          PUSH   DPL
          PUSH   00H
          PUSH   02H
          PUSH   05H
          MOV    R2,#00H

LOOP_SCAN_A: MOV  DPTR,#PORTA
          MOVX   A,@DPTR
          CJNE   A,#0FFH,SCAN_A

SCAN_COL_A: MOV  A,R2
          MOV    DPTR,#PORTB
          MOVX   @DPTR,A
          MOV    52H,R2
          INC    R2
          CJNE   R2,#10H,LOOP_SCAN_A
          AJMP   N_END

SCAN_A:   MOV    55H,A ; STORE A IN RAM ADDRESS 55H
SCAN_OPEN_A: MOV  DPTR,#PORTA
          MOVX   A,@DPTR
          CJNE   A,#0FFH,SCAN_OPEN_A
          MOV    53H,#00H
          MOV    R0,#53H
          MOV    51H,55H ; PUSH 55H = ADDRESS 31H ;

ROW
          MOV    A,51H
          SWAP   A
          XCHD   A,@R0
          MOV    51H,A

          MOV    53H,#00H
          MOV    A,52H
          SWAP   A
          XCHD   A,@R0
          SWAP   A
          ORL    A,51H
          MOV    54H,A

LOOP_AA:  MOV    R5,#64
          MOV    DPTR,#TABEL_KEY

LOOP_SS:  CLR    A
          MOVC   A,@A+DPTR
          CJNE   A,54H,NEXT_I
          SJMP   NEXT_TO_I

NEXT_I:   INC     DPTR
          DJNZ   R5,LOOP_SS
          SJMP   LOOP_AA

NEXT_TO_I: ; KEY MATCH
          LCALL  CONVERS
          POP     05H
          POP     02H
          POP     00H
          POP     DPL
          POP     DPH
          RET

; CONVER KEYCODES TO ASCII CODES
CONVERS:  MOV    55H,#00H ; SWAP XXXX-0000
          MOV    R0,#55H
          XCHD   A,@R0
NEXT_1:   CJNE   A,#0E0H,NEXT_2 ; E0 - EF
          AJMP   CONVER_E0
NEXT_2:   CJNE   A,#0D0H,NEXT_3 ; D0 - DF
          AJMP   CONVER_D0
NEXT_3:   CJNE   A,#0B0H,NEXT_4 ; B0 - BF
          AJMP   CONVER_B0
NEXT_4:   CJNE   A,#070H,NEXT_1 ; 70 - 7F
          AJMP   CONVER_70

;*****
; CONVER E0
CONVER_E0: MOV    A,54H
C_E0:     CJNE   A,#0E0H,C_E1 ; 0
          MOV    A,#30H
          AJMP   NEXT_END
C_E1:     CJNE   A,#0E1H,C_E2 ; 1
          MOV    A,#31H

```

```

      AJMP     NEXT_END
C_E2:  CJNE    A,#0E2H,C_E3  ;2
      MOV     A,#32H
      AJMP     NEXT_END
C_E3:  CJNE    A,#0E3H,C_E4  ;3
      MOV     A,#33H
      AJMP     NEXT_END
C_E4:  CJNE    A,#0E4H,C_E5  ;4
      MOV     A,#34H
      AJMP     NEXT_END
C_E5:  CJNE    A,#0E5H,C_E6  ;5
      MOV     A,#35H
      AJMP     NEXT_END
C_E6:  CJNE    A,#0E6H,C_E7  ;6
      MOV     A,#36H
      AJMP     NEXT_END
C_E7:  CJNE    A,#0E7H,C_E8  ;7
      MOV     A,#37H
      AJMP     NEXT_END
C_E8:  CJNE    A,#0E8H,C_E9  ;8
      MOV     A,#38H
      AJMP     NEXT_END
C_E9:  CJNE    A,#0E9H,C_EA  ;9
      MOV     A,#39H
      AJMP     NEXT_END
C_EA:  CJNE    A,#0EAH,C_EB  ;A
      MOV     A,#41H
      AJMP     NEXT_END
C_EB:  CJNE    A,#0EBH,C_EC  ;B
      MOV     A,#42H
      AJMP     NEXT_END
C_EC:  CJNE    A,#0ECH,C_ED  ;C
      MOV     A,#43H
      AJMP     NEXT_END
C_ED:  CJNE    A,#0EDH,C_EE  ;D
      MOV     A,#44H
      AJMP     NEXT_END
C_EE:  CJNE    A,#0EEH,C_EF  ;E
      MOV     A,#45H
      AJMP     NEXT_END
C_EF:  CJNE    A,#0EFH,C_E0  ;F
      MOV     A,#46H
      AJMP     NEXT_END
;*****
;CONVER D0
CONVER_D0: MOV     A,54H
C_D0:  CJNE    A,#0D0H,C_DI  ;G
      MOV     A,#47H
      AJMP     NEXT_END
      CJNE    A,#0D1H,C_D2  ;H
      MOV     A,#48H
      AJMP     NEXT_END
      CJNE    A,#0D2H,C_D3  ;I
      MOV     A,#49H
      AJMP     NEXT_END
      CJNE    A,#0D3H,C_D4  ;J
      MOV     A,#4AH
      AJMP     NEXT_END
      CJNE    A,#0D4H,C_D5  ;K
      MOV     A,#4BH
      AJMP     NEXT_END
      CJNE    A,#0D5H,C_D6  ;L
      MOV     A,#4CH
      AJMP     NEXT_END
      CJNE    A,#0D6H,C_D7  ;M
      MOV     A,#4DH
      AJMP     NEXT_END
      CJNE    A,#0D7H,C_D8  ;N
      MOV     A,#4EH
      AJMP     NEXT_END
      CJNE    A,#0D8H,C_D9  ;O
      MOV     A,#4FH
      AJMP     NEXT_END
      CJNE    A,#0D9H,C_DA  ;P
      MOV     A,#50H
      AJMP     NEXT_END
      CJNE    A,#0DAH,C_DB  ;Q
      MOV     A,#51H
      AJMP     NEXT_END
      CJNE    A,#0DBH,C_DC  ;R
      MOV     A,#52H
      AJMP     NEXT_END
      CJNE    A,#0DCH,C_DD  ;S
      MOV     A,#53H
      AJMP     NEXT_END
      CJNE    A,#0DDH,C_DE  ;T
      MOV     A,#54H
      AJMP     NEXT_END
      CJNE    A,#0DEH,C_DF  ;U
      MOV     A,#55H
      AJMP     NEXT_END
      CJNE    A,#0DFH,C_D0  ;V
      MOV     A,#56H
      AJMP     NEXT_END
;*****
CONVER_B0: MOV     A,54H
C_B0:  CJNE    A,#0B0H,C_B1  ;W

```

|       |                               |                                      |
|-------|-------------------------------|--------------------------------------|
|       | MOV A,#57H                    | CONVER_70: MOV A,54H                 |
|       | AJMP NEXT_END                 | C_70: CJNE A,#070H,C_71 ;F3          |
| C_B1: | CJNE A,#0B1H,C_B2 ;X          | MOV A,#03H                           |
|       | MOV A,#58H                    | AJMP NEXT_END                        |
|       | AJMP NEXT_END                 | C_71: CJNE A,#071H,C_72 ;F4          |
| C_B2: | CJNE A,#0B2H,C_B3 ;Y          | MOV A,#04H                           |
|       | MOV A,#59H                    | AJMP NEXT_END                        |
|       | AJMP NEXT_END                 | C_72: CJNE A,#072H,C_73 ;;           |
| C_B3: | CJNE A,#0B3H,C_B4 ;Z          | MOV A,#3BH                           |
|       | MOV A,#5AH                    | AJMP NEXT_END                        |
|       | AJMP NEXT_END                 | C_73: CJNE A,#073H,C_74 ;ARROW UP    |
| C_B4: | CJNE A,#0B4H,C_B5 ;(          | MOV A,#18H                           |
|       | MOV A,#28H                    | AJMP NEXT_END                        |
|       | AJMP NEXT_END                 | C_74: CJNE A,#074H,C_75 ;%           |
| C_B5: | CJNE A,#0B5H,C_B6 ;)          | MOV A,#25H                           |
|       | MOV A,#29H                    | AJMP NEXT_END                        |
|       | AJMP NEXT_END                 | C_75: CJNE A,#075H,C_76 ;\$          |
| C_B6: | CJNE A,#0B6H,C_B7 ;!          | MOV A,#24H                           |
|       | MOV A,#21H                    | AJMP NEXT_END                        |
|       | AJMP NEXT_END                 | C_76: CJNE A,#076H,C_77 ;*           |
| C_B7: | CJNE A,#0B7H,C_B8 ;BACK SPACE | MOV A,#2AH                           |
|       | MOV A,#08H                    | AJMP NEXT_END                        |
|       | AJMP NEXT_END                 | C_77: CJNE A,#077H,C_78 ;=           |
| C_B8: | CJNE A,#0B8H,C_B9 ;F1         | MOV A,#3DH                           |
|       | MOV A,#01H                    | AJMP NEXT_END                        |
|       | AJMP NEXT_END                 | C_78: CJNE A,#078H,C_79 ;ESC         |
| C_B9: | CJNE A,#0B9H,C_BA ;F2         | MOV A,#27H                           |
|       | MOV A,#02H                    | AJMP NEXT_END                        |
|       | AJMP NEXT_END                 | C_79: CJNE A,#079H,C_7A ;SPACE       |
| C_BA: | CJNE A,#0BAH,C_BB ;:          | MOV A,#20H                           |
|       | MOV A,#3AH                    | AJMP NEXT_END                        |
|       | AJMP NEXT_END                 | C_7A: CJNE A,#07AH,C_7B ;ARROW LEFT  |
| C_BB: | CJNE A,#0BBH,C_BC ;,          | MOV A,#1BH                           |
|       | MOV A,#2CH                    | AJMP NEXT_END                        |
|       | AJMP NEXT_END                 | C_7B: CJNE A,#07BH,C_7C ;ARROW DOWN  |
| C_BC: | CJNE A,#0BCH,C_BD ;"          | MOV A,#19H                           |
|       | MOV A,#22H                    | AJMP NEXT_END                        |
|       | AJMP NEXT_END                 | C_7C: CJNE A,#07CH,C_7D ;ARROW RIGHT |
| C_BD: | CJNE A,#0BDH,C_BE ;?          | MOV A,#1AH                           |
|       | MOV A,#3FH                    | AJMP NEXT_END                        |
|       | AJMP NEXT_END                 | C_7D: CJNE A,#07DH,C_7E ;#           |
| C_BE: | CJNE A,#0BEH,C_BF ;.          | MOV A,#23H                           |
|       | MOV A,#2EH                    | AJMP NEXT_END                        |
|       | AJMP NEXT_END                 | C_7E: CJNE A,#07EH,C_7F ;ENTER       |
| C_BF: | CJNE A,#0BFH,C_B0 ;&          | MOV A,#0DH                           |
|       | MOV A,#26H                    | AJMP NEXT_END                        |
|       | AJMP NEXT_END                 | C_7F: CJNE A,#07FH,C_70 ;ENTER       |
|       |                               | MOV A,#0DH                           |
|       |                               | AJMP NEXT_END                        |

\*\*\*\*\*  
 เอกสารนี้เป็นเอกสารที่จัดทำขึ้นเพื่อการใช้งานเพื่อการศึกษาเท่านั้น ไม่ใช่นโยบายด้านการค้า  
 ไม่สามารถใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้  
 ;CONVER 70

```

;*****
NEXT_END: MOV    54H,A
            RET

;-----
; REAL TIME CLOCK
RTC:        LCALL  CLEAR
            MOV    DPTR,#DIS_CLK ;DISPLAY CLOCK
            LCALL  PG_LCD

RTC_A:      MOV    COMMAND_RTC,#81H ;SEC
            ACALL  READ_RTC
            MOV    A,OUTPUT_RTC
            LCALL  CONVER_ASCII
            MOV    DPTR,#COMMAND
            MOV    A,#21H
            ORL    A,#80H
            MOVX   @DPTR,A
            LCALL  WAIT
            MOV    DPTR,#WRITEDATA
            MOV    A,COL1
            MOVX   @DPTR,A
            LCALL  WAIT
            MOV    DPTR,#COMMAND
            MOV    A,#20H
            ORL    A,#80H
            MOVX   @DPTR,A
            LCALL  WAIT
            MOV    DPTR,#WRITEDATA
            MOV    A,COL2
            MOVX   @DPTR,A
            LCALL  WAIT

DOT:        MOV    DPTR,#COMMAND ;;
            MOV    A,#1FH
            ORL    A,#80H
            MOVX   @DPTR,A
            LCALL  WAIT
            MOV    DPTR,#WRITEDATA
            MOV    A,#:"
            MOVX   @DPTR,A
            LCALL  WAIT

MIN:        MOV    COMMAND_RTC,#83H ; MIN
            ACALL  READ_RTC
            MOV    A,OUTPUT_RTC
            LCALL  CONVER_ASCII
            MOV    DPTR,#COMMAND
            MOV    A,#1EH
            ORL    A,#80H
            MOVX   @DPTR,A
            LCALL  WAIT
            MOV    DPTR,#WRITEDATA
            MOV    A,COL1
            MOVX   @DPTR,A
            LCALL  WAIT
            MOV    DPTR,#COMMAND
            MOV    A,#1DH
            ORL    A,#80H
            MOVX   @DPTR,A
            LCALL  WAIT
            MOV    DPTR,#WRITEDATA
            MOV    A,COL2
            MOVX   @DPTR,A
            LCALL  WAIT

DQT1:      MOV    DPTR,#COMMAND ;:"
            MOV    A,#1CH
            ORL    A,#80H
            MOVX   @DPTR,A
            LCALL  WAIT
            MOV    DPTR,#WRITEDATA
            MOV    A,#:"
            MOVX   @DPTR,A
            LCALL  WAIT

HR:        MOV    COMMAND_RTC,#85H ; HR
            ACALL  READ_RTC
            MOV    A,OUTPUT_RTC
            LCALL  CONVER_ASCII
            MOV    DPTR,#COMMAND
            MOV    A,#IBH
            ORL    A,#80H
            MOVX   @DPTR,A
            LCALL  WAIT
            MOV    DPTR,#WRITEDATA
            MOV    A,COL1
            MOVX   @DPTR,A
            LCALL  WAIT
            MOV    DPTR,#COMMAND
            MOV    A,#IAH
            ORL    A,#80H
            MOVX   @DPTR,A
            LCALL  WAIT
            MOV    DPTR,#WRITEDATA
            MOV    A,COL2
            MOVX   @DPTR,A

```

4

```

MOV R2,#08 ; VOL LOOP
CLR CY ; CLEAR CARRY FLAG
MOV A,COMMAND_RTC ; INPUT IN ACC
SETB RST ; START RTC (ACTIVE)
LOOP_COMM: RRC A ; ROTATE ACC TO CARRY FLAG
MOV DATA_IO,C ; OUT INPUT IO TO RTC
SETB CLK
NOP
CLR CLK ; CLOCK LOW
DJNZ R2,LOOP_COMM ; 8 LOOP
MOV R2,#08H ; VOL LOOP
CLR CY ; CLEAR CARRY FLAG
MOV A,DATA_RTC ; DATA TO ACC
LOOP_DATA: RRC A ; ROTATE ACC TO CARRY FLAG
MOV DATA_IO,C ; OUT INPUT IO TO RTC
SETB CLK
NOP
CLR CLK ; CLOCK LOW
DJNZ R2,LOOP_DATA
CLR RST
POP ACC
RET

;-----
;READ DATA FROM RTC
READ_RTC: PUSH ACC
MOV P1,#00H ; CLEAR PORT 1
MOV R2,#08H ; VOL LOOP
CLR CY ; CLEAR CARRY FLAG
MOV A,COMMAND_RTC ; INPUT IN ACC
SETB RST ; START RTC (ACTIVE)
LOOP_COMM1:RRC A ; ROTATE ACC TO CARRY FLAG
MOV DATA_IO,C ; OUT INPUT IO TO RTC
SETB CLK
NOP
CLR CLK ; CLOCK LOW
DJNZ R2,LOOP_COMM1 ; 8 LOOP
MOV R2,#08H
CLR A
CLR CY ; CLEAR CARRY FLAG
LOOP_DATA1:SETB CLK
NOP
MOV C,DATA_IO ; READ OUTPUT RTC TO A
CLR CLK
RRC A ; ROTATE ACC TO CARRY FLAG
SETB CLK ; CLOCK HIGH
DJNZ R2,LOOP_DATA1
CLR RST
MOV OUTPUT_RTC,A
POP ACC

RET

;-----
; ASCII CODE CONVER SEC
CONVER_ASCII: PUSH ACC
PUSH 00H
MOV A,OUTPUT_RTC
SWAP A
MOV 4EH,#00H
MOV R0,#4EH
XCHD A,@R0
SWAP A
MOV COL1,A
XCHD A,@R0
MOV COL2,A
ORL COL1,#30H
ORL COL2,#30H
POP 00H
POP ACC
RET

;-----
;LOG BOOK
LOG_BOOK: MOV DPTR,#DIS_13 ; DISPLAY LOG BOOK
LCALL PG_LCD
LCALL OFF_CURSOR
LCALL CHK
LNK: LCALL SCANKEY
CJNE A,#27H,LNX0
LJMP PG_2
LNK0: CJNE A,#30H,LNX1
MOV R0,#40H ; ADDRESS DD RAM
MOV R1,#30H ; 0
MOV R2,#01H ; DPH
MOV R3,#30H ; DPL
MOV R4,#01H ; DPH INDEX
MOV R5,#20H ; DPL
MOV R6,#08H ; DPH MESSAGE
MOV R7,#00H ; DPL
LJMP TO_LOG
LNK1: CJNE A,#31H,LNX2
MOV R0,#42H ; ADDRESS DD RAM
MOV R1,#31H ; 1
MOV R2,#01H ; DPH
MOV R3,#80H ; DPL
MOV R4,#01H ; DPH INDEX
MOV R5,#21H ; DPL
MOV R6,#0AH ; DPH MESSAGE
MOV R7,#58H ; DPL
LJMP TO_LOG

```

[illegible]

```

NXLI:  CJNE  A,#4FH,NXL2  ; OPEN LOG BOOK
        MOV   DPH,R4      ; INDEX MEMORY CHK
        MOV   DPL,R5
        MOVX  A,@DPTR
        CJNE  A,#00H,NL_LOG1
        LJMP  OPEN

NL_LOG1: LJMP  OPEN_EP

NXLI2:  CJNE  A,#44H,NXL3  ; DELETE
        LJMP  DEL

NXLI3:  CJNE  A,#4CH,NX_LOG ; LINK
        LJMP  LINK

NX_LOG: MOV   A,R0        ; DD RAM
        ORL   A,#80H
        MOV   DPTR,#COMMAND ; SET ADDRESS
        DD RAM
        MOVX  @DPTR,A
        LCALL WAIT
        MOV   A,R1        ; CHAR
        MOV   DPTR,#WRITEDATA
        MOVX  @DPTR,A
        LCALL WAIT
        LCALL DELAY_LOG
        LJMP  TO_LOG

; OPEN LOG BOOK
OPEN:   MOV   DPH,R2      ; MAIN
        MOV   DPL,R3
        LCALL PG_LOG
        LCALL SCANKEY
        CJNE  A,#27H,OPEN
        LJMP  LOG_BOOK

; WRITE DATA TO LCD = 1 PAGE FOR LOG BOOKS
PG_LOG: PUSH  00H
        PUSH  ACC
        MOV   R0,#00H

LOG_1:  PUSH  DPL
        PUSH  DPH
        MOV   DPTR,#COMMAND
        MOV   A,R0
        ORL   A,#80H
        MOVX  @DPTR,A
        LCALL WAIT
        POP   DPH

POP      DPL
MOVX     A,@DPTR
INC      DPTR
PUSH     DPH
PUSH     DPL
MOV      DPTR,#WRITEDATA
MOVX     @DPTR,A
LCALL    WAIT
POP      DPL
POP      DPH
INC      R0
CJNE     R0,#14H,LNN_1
MOV      R0,#40H
SJMP     LOG_1

LNN_1:  CJNE  R0,#54H,LNN_2
        MOV   R0,#14H
        SJMP  LOG_1

LNN_2:  CJNE  R0,#28H,LNN_3
        MOV   R0,#54H
        SJMP  LOG_1

LNN_3:  CJNE  R0,#68H,LOG_1
        POP   ACC
        POP   00H
        RET

; OPEN LOG BOOK EPROM
OPEN_EP: MOV   DPTR,#FORM
        LCALL PG_EP
        LCALL SCANKEY
        CJNE  A,#27H,OPEN_EP
        LJMP  LOG_BOOK

; WRITE DATA TO LCD = 1 PAGE FOR LOG BOOKS
PG_EP:  PUSH  00H
        PUSH  01H
        PUSH  ACC
        MOV   R0,#00H
        MOV   R1,#00H

EOG_1:  PUSH  DPL
        PUSH  DPH
        PUSH  ACC
        MOV   DPTR,#COMMAND
        MOV   A,R0
        ORL   A,#80H
        MOVX  @DPTR,A
        LCALL WAIT
        POP   ACC
        POP   DPH

```

```

POP      DPL
MOV      A,R1
MOVC     A,@A+DPTR
INC      R1
PUSH     DPH
PUSH     DPL
MOV      DPTR,#WRITEDATA
MOVSX    @DPTR,A
LCALL    WAIT
POP      DPL
POP      DPH
INC      R0
CJNE     R0,#14H,ANN_1
MOV      R0,#40H
LJMP     EOG_1
ANN_1:   CJNE     R0,#54H,ANN_2
MOV      R0,#14H
LJMP     EOG_1
ANN_2:   CJNE     R0,#28H,ANN_3
MOV      R0,#54H
LJMP     EOG_1
ANN_3:   CJNE     R0,#68H,EOG_1
POP      ACC
POP      01H
POP      00H
RET

; DELETE
DEL:      MOV      DPH,R2 ; ADDRESS RAM
MOV      DPL,R3
PUSH     00H
PUSH     01H
MOV      R0,#00H
MOV      R1,#00H

DEL_STORE: PUSH     DPL
PUSH     DPH
MOV      DPTR,#FORM
MOV      A,R0
MOVC     A,@A+DPTR ; GET DATA
INC      R0
POP      DPH
POP      DPL
MOVSX    @DPTR,A
INC      DPTR
INC      R1 ; COUNTER
CJNE     R1,#50H,DEL_STORE
POP      01H
POP      00H

PUSH     00H;CLEAR DATA OF EXTERNAL RAM
PUSH     01H
MOV      R0,#40
MOV      R1,#15
MOV      DPH,R6 ; MESSAGE
MOV      DPL,R7
MOV      A,#20H ; DATA = SPACE
MEI:      MOVSX    @DPTR,A
INC      DPTR
DJNZ     R0,MEI
MOV      R0,#40
DJNZ     R1,MEI
POP      01H
POP      00H
MOV      DPH,R4 ; CLEAR INDEX MEMORY
MOV      DPL,R5
MOV      A,#0FFH
MOVSX    @DPTR,A
LJMP     LOG_BOOK

; LINK DATA TO PC
LINK:      MOV      DPH,R2
MOV      DPL,R3
MOV      SCON,#40H
MOV      TMOD,#20H
MOV      TH1,#-3 ; 9600 BPS
SETB     TR1 ; STRAT LINK
LCALL    NEW_LINE
LCALL    PC_H ; HEADING OF FILE
LCALL    NEW_LINE
PUSH     00H
PUSH     07H
MOV      R7,#04H

LOOP_LINK1: MOV      R0,#00H
LOOP_LINK: MOVSX    A,@DPTR
MOV      SBUF,A
JNB      TI,$
CLR      TI
INC      DPTR
INC      R0
CJNE     R0,#20,LOOP_LINK
LCALL    NEW_LINE
DJNZ     R7,LOOP_LINK1
POP      07H
POP      00H
LCALL    NEW_LINE
LCALL    PC_M
LCALL    NEW_LINE

```

```

MOV    DPH,R6    ;START OF ADDRESS
MESSAGE
MOV    DPL,R7
PUSH   00H
PUSH   01H
MOV    R0,#40
MOV    R1,#15
LPI:   MOVX   A,@DPTR
INC    DPTR      ; DPTR+1
MOV    SBUF,A
JNB    TI,$
CLR    TI
DJNZ   R0,LPI
MOV    R0,#40
ACALL  NEW_LINE
DJNZ   R1,LPI
POP    01H
POP    00H
LCALL  PC_E
LCALL  NEW_LINE
CLR    TR1      ; STOP LINK
LJMP   LOG_BOOK

; NEW LINE
NEW_LINE: MOV  A,#10 ; LF
MOV    SBUF,A
JNB    TI,$
CLR    TI
MOV    A,#13 ; CR
MOV    SBUF,A
JNB    TI,$
CLR    TI
RET

; CHECK INDEX MEMORY
CHK:   MOV    R0,#14H    ; DD RAM
MOV    R1,#01H    ; DPH
MOV    R2,#20H    ; DPL

L_CHK: MOV    DPH,R1
MOV    DPL,R2
MOVSX  A,@DPTR
CJNE   A,#00H,E_CHK
MOV    DPTR,#COMMAND
MOV    A,R0
ORL    A,#80H
MOVSX  @DPTR,A
LCALL  WAIT
MOV    DPTR,#WRITEDATA
MOV    A,COL1
MOVSX  @DPTR,A
LCALL  WAIT
MOV    DPTR,#COMMAND
MOV    A,#46H
ORL    A,#80H
MOVSX  @DPTR,A
LCALL  WAIT
MOV    DPTR,#WRITEDATA
MOV    A,COL2
MOVSX  @DPTR,A
LCALL  WAIT

;STORE DATE
STO_DATE: MOV  COMMAND_RTC,#87H ; DATE
LCALL  READ_RTC
MOV    A,OUTPUT_RTC
LCALL  CONVER_ASCII
MOV    DPTR,#COMMAND
MOV    A,#47H
ORL    A,#80H
MOVSX  @DPTR,A
LCALL  WAIT
MOV    DPTR,#WRITEDATA
MOV    A,COL1
MOVSX  @DPTR,A
LCALL  WAIT
MOV    DPTR,#COMMAND
MOV    A,#46H
ORL    A,#80H
MOVSX  @DPTR,A
LCALL  WAIT
MOV    DPTR,#WRITEDATA
MOV    A,COL2
MOVSX  @DPTR,A
LCALL  WAIT

STO_MONTH: MOV  COMMAND_RTC,#89H ; MONTH
LCALL  READ_RTC
MOV    A,OUTPUT_RTC
LCALL  CONVER_ASCII

```

|               |                         |       |      |
|---------------|-------------------------|-------|------|
| MOV           | DPTR,#COMMAND           | LCALL | WAIT |
| MOV           | A,#4AH                  | RET   |      |
| ORL           | A,#80H                  |       |      |
| MOVX          | @DPTR,A                 |       |      |
| LCALL         | WAIT                    |       |      |
| MOV           | DPTR,#WRITEDATA         |       |      |
| MOV           | A,COL1                  |       |      |
| MOVX          | @DPTR,A                 |       |      |
| LCALL         | WAIT                    |       |      |
| MOV           | DPTR,#COMMAND           |       |      |
| MOV           | A,#49H                  |       |      |
| ORL           | A,#80H                  |       |      |
| MOVX          | @DPTR,A                 |       |      |
| LCALL         | WAIT                    |       |      |
| MOV           | DPTR,#WRITEDATA         |       |      |
| MOV           | A,COL2                  |       |      |
| MOVX          | @DPTR,A                 |       |      |
| LCALL         | WAIT                    |       |      |
| MOV           | DPTR,#COMMAND ; -       |       |      |
| MOV           | A,#4BH                  |       |      |
| ORL           | A,#80H                  |       |      |
| MOVX          | @DPTR,A                 |       |      |
| LCALL         | WAIT                    |       |      |
| MOV           | DPTR,#WRITEDATA         |       |      |
| MOV           | A,#'                    |       |      |
| MOVX          | @DPTR,A                 |       |      |
| LCALL         | WAIT                    |       |      |
| STO_YEAR: MOV | COMMAND_RTC,#8DH ; YEAR |       |      |
| LCALL         | READ_RTC                |       |      |
| MOV           | A,OUTPUT_RTC            |       |      |
| LCALL         | CONVER_ASCII            |       |      |
| MOV           | DPTR,#COMMAND           |       |      |
| MOV           | A,#4DH                  |       |      |
| ORL           | A,#80H                  |       |      |
| MOVX          | @DPTR,A                 |       |      |
| LCALL         | WAIT                    |       |      |
| MOV           | DPTR,#WRITEDATA         |       |      |
| MOV           | A,COL1                  |       |      |
| MOVX          | @DPTR,A                 |       |      |
| LCALL         | WAIT                    |       |      |
| MOV           | DPTR,#COMMAND           |       |      |
| MOV           | A,#4AH                  |       |      |
| ORL           | A,#80H                  |       |      |
| MOVX          | @DPTR,A                 |       |      |
| LCALL         | WAIT                    |       |      |
| MOV           | DPTR,#WRITEDATA         |       |      |
| MOV           | A,COL2                  |       |      |
| MOVX          | @DPTR,A                 |       |      |
| LCALL         | WAIT                    |       |      |
| STO_MIN: MOV  | COMMAND_RTC,#83H ; MIN  |       |      |
| LCALL         | READ_RTC                |       |      |
| MOV           | A,OUTPUT_RTC            |       |      |
| LCALL         | CONVER_ASCII            |       |      |
| MOV           | DPTR,#COMMAND           |       |      |
| MOV           | A,#4BH                  |       |      |
| ORL           | A,#80H                  |       |      |
| MOVX          | @DPTR,A                 |       |      |
| LCALL         | WAIT                    |       |      |
| MOV           | DPTR,#WRITEDATA         |       |      |
| MOV           | A,COL1                  |       |      |
| MOVX          | @DPTR,A                 |       |      |
| LCALL         | WAIT                    |       |      |
| MOV           | DPTR,#COMMAND           |       |      |
| MOV           | A,#4AH                  |       |      |
| ORL           | A,#80H                  |       |      |
| MOVX          | @DPTR,A                 |       |      |
| LCALL         | WAIT                    |       |      |
| MOV           | DPTR,#WRITEDATA         |       |      |
| MOV           | A,COL2                  |       |      |
| MOVX          | @DPTR,A                 |       |      |
| LCALL         | WAIT                    |       |      |

```

STO_HR: MOV    COMMAND_RTC,#85H ; HR
        LCALL   READ_RTC
        MOV     A,OUTPUT_RTC
        LCALL   CONVER_ASCII
        MOV     DPTR,#COMMAND
        MOV     A,#48H
        ORL     A,#80H
        MOVX    @DPTR,A
        LCALL   WAIT
        MOV     DPTR,#WRITEDATA
        MOV     A,COL1
        MOVX    @DPTR,A
        LCALL   WAIT
        MOV     DPTR,#COMMAND
        MOV     A,#47H
        ORL     A,#80H
        MOVX    @DPTR,A
        LCALL   WAIT
        MOV     DPTR,#WRITEDATA
        MOV     A,COL2
        MOVX    @DPTR,A
        LCALL   WAIT
        RET

; STORE CALLSIGN
STO_CSIGN: MOV    R0,#4AH
        MOV     DPTR,#0100H
AX_1:  MOVX    A,@DPTR
        MOV     R1,A
        PUSH    DPH
        PUSH    DPL
        MOV     DPTR,#COMMAND
        MOV     A,R0
        ORL     A,#80H
        MOVX    @DPTR,A
        LCALL   WAIT
        MOV     DPTR,#WRITEDATA
        MOV     A,R1
        MOVX    @DPTR,A
        LCALL   WAIT
        POP     DPL
        POP     DPH
        INC     R0
        INC     DPL
        CJNE    R0,#50H,AX_1
        RET

; SETUP DATE
S_DATE: MOV    COMMAND_RTC,#8EH ; WRITE PROTECT
        MOV     DATA_RTC,#00H
        LCALL   SEND_RTC
        MOV     DPTR,#DIS_9
        LCALL   PG_LCD
        LCALL   STO_DATE
        MOV     DPTR,#COMMAND ; SET ADDRESS
        DD RAM 1AH START
        MOV     R0,#1AH
        MOV     A,#1AH
        ORL     A,#80H
        MOVX    @DPTR,A
        LCALL   WAIT
        LCALL   ON_CURSOR
S_SCAN: LCALL   SCANKEY ; DATE
        CJNE    A,#27H,N_SCAN
        LCALL   OFF_CURSOR
        LJMP    PG_4
N_SCAN: CJNE    A,#30H,N_S1
        LJMP    S_SCAN_0
N_S1:  CJNE    A,#31H,N_S2
        LJMP    S_SCAN_1
N_S2:  CJNE    A,#32H,N_S3
        LJMP    S_SCAN_2
N_S3:  CJNE    A,#33H,S_SCAN
        LJMP    S_SCAN_3
S_SCAN_0: MOV    DPTR,#WRITEDATA ; 0,1-9
        MOVX    @DPTR,A
        LCALL   WAIT
        LCALL   COMPARE_0
        CJNE    A,#27H,S_NXT_0
        LCALL   OFF_CURSOR
        LCALL   PG_4
S_NXT_0: MOVX    @DPTR,A
        LCALL   WAIT
        LJMP    S_SCAN_NXT
S_SCAN_1: MOV    DPTR,#WRITEDATA ; 1,0-9
        MOVX    @DPTR,A
        LCALL   WAIT
        LCALL   COMPARE_1
        CJNE    A,#27H,S_NXT_1
        LCALL   OFF_CURSOR
        LCALL   PG_4
S_NXT_1: MOVX    @DPTR,A
        LCALL   WAIT
        LJMP    S_SCAN_NXT

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้เผยแพร่ไปใช้ประโยชน์ด้านการค้า  
โดยไม่ได้รับอนุญาต ทั้งนี้ หักห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของลิขสิทธิ์ที่ทำการนำไปใช้

เอกสารนี้เป็นเอกสารที่ส่งมาเพื่อสำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

|       |                |         |       |                |
|-------|----------------|---------|-------|----------------|
| CJNE  | A,#27H,S_NXT_4 | LJMP    | PG_4  |                |
| LCALL | OFF_CURSORM    | SD_NX1: | CJNE  | A,#0DH,SD_SCAN |
| LCALL | PG_4           |         | LCALL | OFF_CURSOR     |

ไม่ทำการตีพิมพ์ซ้ำโดยไม่ได้รับอนุญาต การนำเอกสารนี้ไปใช้โดยไม่ได้รับอนุญาตถือว่าผิดกฎหมาย



```

LJMP PG_4
S_T2: MOV DPTR,#WRITEDATA
      MOVX @DPTR,A
      LCALL WAIT
      LJMP NXS_END

SE_T3: MOV DPTR,#WRITEDATA
      MOVX @DPTR,A
      LCALL WAIT
      LCALL COMPARE_4
      CJNE A,#27H,S_T3
      LCALL OFF_CURSOR
      LJMP PG_4

S_T3: MOV DPTR,#WRITEDATA
      MOVX @DPTR,A
      LCALL WAIT

NXS_END: LCALL OFF_CURSOR
          MOV DPTR,#COMMAND ; SET ADDRESS
DD RAM 1EH START
      MOV A,#1EH
      ORL A,#80H
      MOVX @DPTR,A
      LCALL WAIT
      LCALL ON_CURSOR
      LCALL COMPARE_5
      CJNE A,#27H,ST_1
      LCALL OFF_CURSOR
      LJMP PG_4

ST_1: MOV DPTR,#WRITEDATA
      MOVX @DPTR,A
      LCALL WAIT
      LCALL COMPARE_1
      CJNE A,#27H,ST_2
      LCALL OFF_CURSOR
      LJMP PG_4

ST_2: MOV DPTR,#WRITEDATA
      MOVX @DPTR,A
      LCALL WAIT

SS_SCAN: LCALL SCANKEY
          CJNE A,#27H,SS_NX1
          LCALL OFF_CURSOR
          LJMP PG_4

SS_NX1: CJNE A,#0DH,SS_SCAN
          LCALL OFF_CURSOR
          MOV A,#1BH ; HR
          LCALL READ_DD
          SWAP A

      MOV 57H,#00H
      MOV R0,#57H
      XCHD A,@R0 ; XXXX-0000
      MOV 58H,A ; STORE A IN 58H
      MOV A,#1CH
      LCALL READ_DD
      SWAP A
      MOV 57H,#00H
      XCHD A,@R0 ; XXXX-0000
      SWAP A
      ORL A,58H
      MOV COMMAND_RTC,#84H ; SEND DATA
      TO RTC
      MOV DATA_RTC,A
      LCALL SEND_RTC
      MOV A,#1EH ; MIN
      LCALL READ_DD
      SWAP A
      MOV 57H,#00H
      MOV R0,#57H
      XCHD A,@R0 ; XXXX-0000
      MOV 58H,A ; STORE A IN 58H
      MOV A,#1FH
      LCALL READ_DD
      SWAP A
      MOV 57H,#00H
      XCHD A,@R0 ; XXXX-0000
      SWAP A
      ORL A,58H
      MOV COMMAND_RTC,#82H ; SEND DATA
      TO RTC
      MOV DATA_RTC,A
      LCALL SEND_RTC
      MOV COMMAND_RTC,#80H ; SEC
      MOV DATA_RTC,#00H
      LCALL SEND_RTC
      LJMP PG_4 ; <SETUP>

; CALLSIGN
S_CALLSIGN: MOV DPTR,#DIS_11
            LCALL PG_LCD
            LCALL STO_CSIGN
            MOV R0,#1EH
            MOV DPTR,#COMMAND ; SET ADDRESS
DD RAM 1EH START
      MOV A,#1EH
      ORL A,#80H
      MOVX @DPTR,A
      LCALL WAIT

```

|           |       |                             |        |       |                 |
|-----------|-------|-----------------------------|--------|-------|-----------------|
|           | LCALL | ON_CURSOR                   |        | MOVX  | @DPTR,A         |
| S_SCAN_C: | LCALL | SCANKEY                     |        | LCALL | WAIT            |
|           | CJNE  | A,#27H,S_S_A                | B_S:   | LJMP  | S_SCAN_C        |
| S_GO:     | LCALL | OFF_CURSOR                  |        | CJNE  | R0,#1EH,B_SI    |
|           | LJMP  | PG_4                        | B_SI:  | LJMP  | S_SCAN_C        |
| S_S_A:    | CJNE  | A,#01H,S_S_B ; F1           |        | DEC   | R0              |
|           | LJMP  | S_SCAN_C                    |        | PUSH  | DPL             |
| S_S_B:    | CJNE  | A,#02H,S_S_C ; F2           |        | PUSH  | DPH             |
|           | LJMP  | S_SCAN_C                    |        | PUSH  | ACC             |
| S_S_C:    | CJNE  | A,#03H,S_S_D ; F3           |        | MOV   | DPTR,#COMMAND   |
|           | LJMP  | S_SCAN_C                    |        | MOV   | A,R0            |
| S_S_D:    | CJNE  | A,#04H,S_S_E ; F4           |        | ORL   | A,#80H          |
|           | LJMP  | S_SCAN_C                    |        | MOVX  | @DPTR,A         |
| S_S_E:    | CJNE  | A,#18H,S_S_F ; UP           |        | LCALL | WAIT            |
|           | LJMP  | S_SCAN_C                    |        | MOV   | A,'#'           |
| S_S_F:    | CJNE  | A,#19H,S_S_G ; DOWN         |        | MOV   | DPTR,#WRITEDATA |
|           | LJMP  | S_SCAN_C                    |        | MOVX  | @DPTR,A         |
| S_S_G:    | CJNE  | A,#1AH,S_S_H ; RIGHT        |        | LCALL | WAIT            |
|           | LJMP  | S_SCAN_C                    |        | LCALL | OFF_CURSOR      |
| S_S_H:    | CJNE  | A,#1BH,S_S_I ; LIFT         |        | MOV   | DPTR,#COMMAND   |
|           | LJMP  | S_SCAN_C                    |        | MOV   | A,R0            |
| S_S_I:    | CJNE  | A,#08H,S_S_J ; BACK SPACE   |        | ORL   | A,#80H          |
|           | LJMP  | B_S                         |        | MOVX  | @DPTR,A         |
| S_S_J:    | CJNE  | A,#0DH,S_S_K ; ENTER        |        | LCALL | WAIT            |
|           | LJMP  | ENTER                       |        | LCALL | ON_CURSOR       |
| S_S_K:    | PUSH  | DPH                         |        | POP   | ACC             |
|           | PUSH  | DPL                         |        | POP   | DPH             |
|           | PUSH  | ACC                         |        | POP   | DPL             |
|           | MOV   | DPTR,#COMMAND               | ENTER: | LJMP  | S_SCAN_C        |
|           | MOV   | A,R0                        |        | CJNE  | R0,#1EH,E_N     |
|           | ORL   | A,#80H                      | E_N:   | LJMP  | S_SCAN_C        |
|           | MOVX  | @DPTR,A                     |        | MOV   | R0,#1EH         |
|           | LCALL | WAIT                        | E_NI:  | MOV   | DPTR,#0100H     |
|           | POP   | ACC                         |        | PUSH  | DPL             |
|           | POP   | DPL                         |        | PUSH  | DPH             |
|           | POP   | DPH                         |        | MOV   | A,R0            |
|           | INC   | R0                          |        | LCALL | READ_DD         |
|           | CJNE  | R0,#25H,GO_T                |        | POP   | DPH             |
| S_N_N:    | LCALL | SCANKEY                     |        | POP   | DPL             |
|           | CJNE  | A,#27H,S_N_I                |        | MOVX  | @DPTR,A         |
|           | LJMP  | S_GO                        |        | INC   | R0              |
| S_N_I:    | CJNE  | A,#0DH,S_N_2                |        | INC   | DPL             |
|           | LJMP  | ENTER                       |        | CJNE  | R0,#24H,E_NI    |
| S_N_2:    | CJNE  | A,#08H,S_N_N                |        | LJMP  | S_GO            |
|           | DEC   | R0                          |        |       |                 |
|           | LJMP  | B_S                         |        |       |                 |
| GO_T:     | MOV   | DPTR,#WRITEDATA ; NEXT STEP |        |       |                 |

|                  |                              |         |       |                            |
|------------------|------------------------------|---------|-------|----------------------------|
| MOV              | TMOD,#20H                    | C_S_C:  | CJNE  | A,#03H,C_S_D ; F3          |
| MOV              | TH1,#-24                     |         | LJMP  | C_SC                       |
| SETB             | TR1                          | C_S_D:  | CJNE  | A,#04H,C_S_E ; F4          |
| MOV              | 66H,#05H ; START ADDRESS OF  |         | LJMP  | C_SC                       |
| MESSAGE          |                              | C_S_E:  | CJNE  | A,#18H,C_S_F ; UP          |
| MOV              | 67H,#00H                     |         | LJMP  | C_SC                       |
| LCALL            | C_RAM ; CLEAR BUFFER MESSAGE | C_S_F:  | CJNE  | A,#19H,C_S_G ; DOWN        |
| LCALL            | CLEAR                        |         | LJMP  | C_SC                       |
| A_CON:           | LCALL CLEAR_TWO              | C_S_G:  | CJNE  | A,#1AH,C_S_H ; RIGHT       |
| MOV              | DPTR,#COMMAND ; SET ADDRESS  |         | LJMP  | C_SC                       |
| DD RAM 00H START |                              | C_S_H:  | CJNE  | A,#1BH,C_S_I ; LIFT        |
|                  |                              |         | LJMP  | C_SC                       |
| MOV              | A,#00H                       | C_S_I:  | CJNE  | A,#08H,C_S_J ; BACK SPACE  |
| ORL              | A,#80H                       |         | LJMP  | BSI                        |
| MOVX             | @DPTR,A                      | C_S_J:  | CJNE  | A,#0DH,C_S_K ; ENTER       |
| LCALL            | WAIT                         |         | LJMP  | ENTERI                     |
| MOV              | R0,#00H                      | C_S_K:  | MOV   | R1,A ; STORE A IN R1       |
| MOV              | DPTR,#0100H                  |         | PUSH  | DPH                        |
| CN_NI:           | MOVX A,@DPTR                 |         | PUSH  | DPL                        |
| MOV              | R1,A                         |         | PUSH  | ACC                        |
| PUSH             | DPL                          |         | MOV   | DPTR,#COMMAND              |
| PUSH             | DPH                          |         | MOV   | A,R0                       |
| MOV              | DPTR,#WRITEDATA              |         | ORL   | A,#80H                     |
| MOV              | A,R1                         |         | MOVX  | @DPTR,A                    |
| MOVX             | @DPTR,A                      |         | LCALL | WAIT                       |
| LCALL            | WAIT                         |         | MOV   | DPTR,#WRITEDATA            |
| POP              | DPH                          |         | MOV   | A,R1                       |
| POP              | DPL                          |         | MOVX  | @DPTR,A                    |
| INC              | R0                           |         | LCALL | WAIT                       |
| INC              | DPTR                         |         | POP   | ACC                        |
| CJNE             | R0,#06H,CN_NI                |         | POP   | DPL                        |
| MOV              | R0,#07H ; R0 = 07H           |         | POP   | DPH                        |
| MOV              | DPTR,#WRITEDATA              |         | INC   | R0                         |
| MOV              | A,#>                         |         | CJNE  | R0,#14H,GO_T1 ; TO 40H-53H |
| MOVX             | @DPTR,A                      |         | LCALL | OFF_CURSOR                 |
| LCALL            | WAIT                         |         | MOV   | R0,#40H                    |
| LCALL            | ON_CURSOR                    |         | MOV   | DPTR,#COMMAND              |
| C_SC:            | LCALL SCANKEY                |         | MOV   | A,R0                       |
| CJNE             | A,#27H,C_S_A                 |         | ORL   | A,#80H                     |
| S_GO1:           | LCALL OFF_CURSOR             |         | MOVX  | @DPTR,A                    |
| CLR              | TR1 ; STOP SERIAL            |         | LCALL | WAIT                       |
| MOV              | IE,#00H ; DISABEL INTERRUPT  |         | LCALL | ON_CURSOR                  |
| LCALL            | ST_MEMORY2 ; STORE PATH 2    |         | LJMP  | C_SC                       |
| LCALL            | SAVE_RAM                     | GO_T1:  | CJNE  | R0,#40H,GO_TN1             |
| LJMP             | PG_3                         |         | SJMP  | GO_NX                      |
| C_S_A:           | CJNE A,#01H,C_S_B ; F1       | GO_TN1: | CJNE  | R0,#41H,GO_TN2             |
| LJMP             | C_SC                         |         | SJMP  | GO_NX                      |
| C_S_B:           | CJNE A,#02H,C_S_C ; F2       | GO_TN2: | CJNE  | R0,#42H,GO_TN3             |
| LJMP             | C_SC                         |         | SJMP  | GO_NX                      |

|          |       |                 |         |                    |                       |
|----------|-------|-----------------|---------|--------------------|-----------------------|
| GO_TN3:  | CJNE  | R0,#43H,GO_TN4  | PUSH    | DPL                |                       |
|          | SJMP  | GO_NX           | PUSH    | ACC                |                       |
| GO_TN4:  | CJNE  | R0,#44H,GO_TN5  | MOV     | A,R0               |                       |
|          | SJMP  | GO_NX           | ORL     | A,#80H             |                       |
| GO_TN5:  | CJNE  | R0,#45H,GO_TN6  | MOV     | DPTR,#COMMAND      |                       |
|          | SJMP  | GO_NX           | MOVX    | @DPTR,A            |                       |
| GO_TN6:  | CJNE  | R0,#46H,GO_TN7  | LCALL   | WAIT               |                       |
|          | SJMP  | GO_NX           | MOV     | A,#'               |                       |
| GO_TN7:  | CJNE  | R0,#47H,GO_TN8  | MOV     | DPTR,#WRITEDATA    |                       |
|          | SJMP  | GO_NX           | MOVX    | @DPTR,A            |                       |
| GO_TN8:  | CJNE  | R0,#48H,GO_TN9  | LCALL   | WAIT               |                       |
|          | SJMP  | GO_NX           | LCALL   | OFF_CURSOR         |                       |
| GO_TN9:  | CJNE  | R0,#49H,GO_TN10 | MOV     | DPTR,#COMMAND      |                       |
|          | SJMP  | GO_NX           | MOV     | A,R0               |                       |
| GO_TN10: | CJNE  | R0,#4AH,GO_TN11 | ORL     | A,#80H             |                       |
|          | SJMP  | GO_NX           | MOVX    | @DPTR,A            |                       |
| GO_TN11: | CJNE  | R0,#4BH,GO_TN12 | LCALL   | WAIT               |                       |
|          | SJMP  | GO_NX           | LCALL   | ON_CURSOR          |                       |
| GO_TN12: | CJNE  | R0,#4CH,GO_TN13 | POP     | ACC                |                       |
|          | SJMP  | GO_NX           | POP     | DPL                |                       |
| GO_TN13: | CJNE  | R0,#4DH,GO_TN14 | POP     | DPH                |                       |
|          | SJMP  | GO_NX           | LJMP    | C_SC               |                       |
| GO_TN14: | CJNE  | R0,#4EH,GO_TN15 |         |                    |                       |
|          | SJMP  | GO_NX           | ENTERI: | LCALL              | OFF_CURSOR ; ENTER TX |
| GO_TN15: | CJNE  | R0,#4FH,GO_TN16 | LCALL   | SAVE_BUF           |                       |
|          | SJMP  | GO_NX           | SETB    | P3.4 ; STAN A      |                       |
| GO_TN16: | CJNE  | R0,#50H,GO_TN17 | LCALL   | DELAY_SEC ; STAN A |                       |
|          | SJMP  | GO_NX           | MOV     | 60H,#'K' ; K       |                       |
| GO_TN17: | CJNE  | R0,#51H,GO_TN18 | SETB    | TI                 |                       |
|          | SJMP  | GO_NX           | LCALL   | DELAY_MS           |                       |
| GO_TN18: | CJNE  | R0,#52H,GO_TN19 | MOV     | 60H,#'A' ; A       |                       |
|          | SJMP  | GO_NX           | SETB    | TI                 |                       |
| GO_TN19: | CJNE  | R0,#53H,GO_NX   | LCALL   | DELAY_MS           |                       |
| G_NX:    | LCALL | SCANKEY         | MOV     | 60H,#'I' ; I       |                       |
|          | CJNE  | A,#27H,G_N1     | SETB    | TI                 |                       |
|          | LCALL | OFF_CURSOR      | LCALL   | DELAY_MS           |                       |
|          | LJMP  | PG_3            | MOV     | R0,#00H            |                       |
| G_N1:    | CJNE  | A,#0DH,G_N2     | EN_X:   | MOV                | A,R0                  |
|          | LJMP  | ENTERI          | LCALL   | READ_DD            |                       |
| G_N2:    | CJNE  | A,#08H,G_NX     | TXN:    | MOV                | 60H,A                 |
|          | LJMP  | BS1             | SETB    | TI                 |                       |
| GO_NX:   | LJMP  | C_SC            | INC     | R0                 |                       |
| BS1:     | CJNE  | R0,#07H,B_NX    | CJNE    | R0,#14H,E_NX1      |                       |
|          | LJMP  | C_SC            | MOV     | R0,#40H            |                       |
| B_NX:    | CJNE  | R0,#40H,B_NXT1  | SJMP    | EN_X               |                       |
|          | MOV   | R0,#13H         | E_NX1:  | CJNE               | R0,#53H,EN_X          |
|          | SJMP  | B_NXT2          | CLR     | P3.4 ; STAN A      |                       |
| B_NXT1:  | DEC   | R0              | LJMP    | A_CON              |                       |
| B_NXT2:  | PUSH  | DPH             |         |                    |                       |

เอกสารนี้เป็นเอกสารที่เผยแพร่ไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่อเผยแพร่ให้ผู้อื่นไปใช้ประโยชน์ด้านการค้า  
 ในรูปแบบใดๆ ทั้งสิ้น ถือว่าผิดกฎหมายและต้องแจ้งเจ้าของลิขสิทธิ์เอกสารทุกครั้งที่มีการนำไปใช้

; STANBY

```

STANBY:  MOV    DPTR,#DIS_12 ; DISPLAY STANBY
          LCALL  PG_LCD
          LCALL  OFF_CURSOR
          MOV    IE,#90H ; SERIAL PORT CONTROL
          MOV    SCON,#50H
          MOV    TMOD,#20H
          MOV    TH1,#-24
          SETB   TR1

```

```

S_SC:    LCALL  OFF_CURSOR
          LCALL  SCANKEY_P
          CJNE   A,#27H,S_S1
          CLR    TR1
          MOV    IE,#00H
          LCALL  OFF_CURSOR
          LJMP   PG_3

```

```

S_S1:    CJNE   A,#43H,S_SC
          CLR    TR1
          MOV    IE,#00H
          LCALL  OFF_CURSOR
          LJMP   CONNECT

```

; RX SERIAL PORT MCS-51

```

RX:       PUSH  ACC
          PUSH  DPH
          PUSH  DPL
          JB    TI,TX
          SJMP  RX_X

```

```

TX:       CLR    TI
          MOV    A,60H ; 60H TRANSFER DATA
          MOV    SBUF,A
          JNB    TI,$
          CLR    TI
          LJMP   RETURN

```

```

RX_X:     JNB    RI,$ ; K
          CLR    RI
          MOV    A,SBUF
          CJNE   A,#4BH,RETURN
          JNB    RI,$ ; A
          CLR    RI
          MOV    A,SBUF
          CJNE   A,#41H,RETURN
          JNB    RI,$ ; I
          CLR    RI
          MOV    A,SBUF

```

```

CJNE      A,#49H,RETURN
MOV       R7,#39

```

```

RX_A:     JNB    RI,$
          CLR    RI
          MOV    A,SBUF
          MOV    R3,A
          LCALL  OFF_CURSOR
          MOV    R2,59H
          PUSH   DPH
          PUSH   DPL
          MOV    DPTR,#COMMAND ; WRITE ADDRESS

```

DD RAM START 14H

```

          MOV    A,R2
          ORL    A,#80H
          MOVX   @DPTR,A
          LCALL  WAIT
          MOV    DPTR,#WRITEDATA
          MOV    A,R3
          MOVX   @DPTR,A
          LCALL  WAIT
          POP    DPL
          POP    DPH
          INC    59H
          MOV    R2,59H
          CJNE   R2,#28H,NRX1
          MOV    59H,#54H
          NRX1:  CJNE   R2,#67H,RETURN_I
          MOV    59H,#14H
          MOV    DPTR,#COMMAND
          MOV    A,R0
          ORL    A,#80H
          MOVX   @DPTR,A
          LCALL  WAIT
          LCALL  ON_CURSOR

```

```

RETURN_I:  DJNZ   R7,RX_A
          :       LCALL  SAVE_RX
RETURN:    POP    DPL
          POP    DPH
          POP    ACC
          RETI

```

; STORE MEMORY PATH1

```

ST_MEMORY1: MOV  DPTR,#0000H ; START OF EXTERNAL
RAM
          MOV    R0,#00H
          MOV    R1,#00H

```

```

LOOP_STORE: PUSH    DPL
              PUSH    DPH
              MOV     DPTR,#FORM
              MOV     A,R0
              MOVC    A,@A+DPTR ; GET DATA
              INC     R0
              POP     DPH
              POP     DPL
              MOVX    @DPTR,A
              INC     DPTR
              INC     R1 ; COUNTER
              CJNE    R1,#50H,LOOP_STORE
              PUSH    DPL
              PUSH    DPH
              PUSH    ACC
              MOV     DPTR,#0120H
              MOV     R0,#00H
              MOVX    A,@DPTR ; GET DPTR START = 120-
129H
              CJNE    A,#00H,STEP_1
STEP_2: INC     DPTR
              INC     R0
              MOVX    A,@DPTR
              CJNE    A,#00H,STEP_1
              CJNE    R0,#09H,STEP_2
              LCALL   FULL ; MEMORY FULL

STEP_1: CJNE    R0,#00H,N_STEP1
              MOV     DPTR,#0120H
              LJMP    TO_STEP
N_STEP1: CJNE    R0,#01H,N_STEP2
              MOV     DPTR,#0121H
              LJMP    TO_STEP
N_STEP2: CJNE    R0,#02H,N_STEP3
              MOV     DPTR,#0122H
              LJMP    TO_STEP
N_STEP3: CJNE    R0,#03H,N_STEP4
              MOV     DPTR,#0123H
              LJMP    TO_STEP
N_STEP4: CJNE    R0,#04H,N_STEP5
              MOV     DPTR,#0124H
              LJMP    TO_STEP
N_STEP5: CJNE    R0,#05H,N_STEP6
              MOV     DPTR,#0125H
              LJMP    TO_STEP
N_STEP6: CJNE    R0,#06H,N_STEP7
              MOV     DPTR,#0126H
              LJMP    TO_STEP
N_STEP7: CJNE    R0,#07H,N_STEP8
              MOV     DPTR,#0127H
              LJMP    TO_STEP
N_STEP8: CJNE    R0,#08H,N_STEP9
              MOV     DPTR,#0128H
              LJMP    TO_STEP
N_STEP9: CJNE    R0,#09H,STEP_1
              MOV     DPTR,#0129H

TO_STEP: MOV     61H,R0 ; INDEX MEMORY
              MOV     62H,DPH
              MOV     63H,DPL

; STORE CALLSIGN USER
              MOV     DPTR,#0100H
              MOV     R0,#00H ; COUNTER
              MOV     R1,#00H ; DPH
              MOV     R2,#05H ; DPL
NS_STEP1: MOVX    A,@DPTR
              INC     DPTR
              PUSH    DPL
              PUSH    DPH
              MOV     DPH,R1
              MOV     DPL,R2
              MOVX    @DPTR,A
              INC     R0
              INC     R2
              POP     DPH
              POP     DPL
              CJNE    R0,#06H,NS_STEP1

; STORE START DATE & TIME
SC_DATE: MOV     COMMAND_RTC,#87H; DATE
              LCALL   READ_RTC
              MOV     A,OUTPUT_RTC
              LCALL   CONVER_ASCII
              MOV     DPTR,#002FH ; COL1
              MOV     A,COL1
              MOVX    @DPTR,A
              MOV     DPTR,#002EH; ; COL2
              MOV     A,COL2
              MOVX    @DPTR,A
              MOV     A,#" ; -
              MOV     DPTR,#0030H
              MOVX    @DPTR,A

```

SC\_MONTH: MOV COMMAND\_RTC,#89H ; MONTH

LCALL READ\_RTC

MOV A,OUTPUT\_RTC

LCALL CONVER\_ASCII

MOV DPTR,#0032H ; COL1

MOV A,COL1

MOVX @DPTR,A

MOV DPTR,#0031H, ; COL2

MOV A,COL2

MOVX @DPTR,A

MOV A,#' ' ; -

MOV DPTR,#0033H

MOVX @DPTR,A

SC\_YEAR: MOV COMMAND\_RTC,#8DH ; YEAR

LCALL READ\_RTC

MOV A,OUTPUT\_RTC

LCALL CONVER\_ASCII

MOV DPTR,#0035H ; COL1

MOV A,COL1

MOVX @DPTR,A

MOV DPTR,#0034H, ; COL2

MOV A,COL2

MOVX @DPTR,A

MOV A,#' ' ; SPACE

MOV DPTR,#0036H

MOVX @DPTR,A

SC\_HR: MOV COMMAND\_RTC,#85H ; HR

LCALL READ\_RTC

MOV A,OUTPUT\_RTC

LCALL CONVER\_ASCII

MOV DPTR,#0038H ; COL1

MOV A,COL1

MOVX @DPTR,A

MOV DPTR,#0037H, ; COL2

MOV A,COL2

MOVX @DPTR,A

MOV A,#' ' ; :

MOV DPTR,#0039H

MOVX @DPTR,A

SC\_MIN: MOV COMMAND\_RTC,#83H ; MIN

LCALL READ\_RTC

MOV A,OUTPUT\_RTC

LCALL CONVER\_ASCII

MOV DPTR,#003BH ; COL1

MOV A,COL1

MOVX @DPTR,A

MOV DPTR,#003AH, ; COL2

MOV A,COL2

MOVX @DPTR,A

POP ACC

POP DPH

POP DPL

RET

; MEMORY FULL

FULL: MOV DPTR,#0090H ; STORE COUNTER

MOVX A,@DPTR

FU0: CJNE A,#00H,FU1

MOV R0,#00H

LJMP TO\_FU

FU1: CJNE A,#01H,FU2

MOV R0,#01H

LJMP TO\_FU

FU2: CJNE A,#02H,FU3

MOV R0,#02H

LJMP TO\_FU

FU3: CJNE A,#03H,FU4

MOV R0,#03H

LJMP TO\_FU

FU4: CJNE A,#04H,FU5

MOV R0,#04H

LJMP TO\_FU

FU5: CJNE A,#05H,FU6

MOV R0,#05H

LJMP TO\_FU

FU6: CJNE A,#06H,FU7

MOV R0,#06H

LJMP TO\_FU

FU7: CJNE A,#07H,FU8

MOV R0,#07H

LJMP TO\_FU

FU8: CJNE A,#08H,FU9

MOV R0,#08H

```

LJMP TO_FU
FU9: CJNE A,#09H,FU10
MOV R0,#09H
LJMP TO_FU

```

```

FU10: MOV A,#00H
MOV DPTR,#0090H
MOVX @DPTR,A
LJMP FULL

```

```

TO_FU: INC A
MOVX @DPTR,A

```

```
RET
```

```

LCALL READ_RTC
MOV A,OUTPUT_RTC
LCALL CONVER_ASCII

```

```

MOV DPTR,#0044H ;COL1
MOV A,COL1
MOVX @DPTR,A
MOV DPTR,#0043H ;COL2
MOV A,COL2
MOVX @DPTR,A

```

```

MOV A,#' ' ; -
MOV DPTR,#0045H
MOVX @DPTR,A

```

```
;STORE MEMORY PATH 2
```

```

ST_MEMORY2: MOV R0,#14H ;STORE CALLSIGN
MOV DPTR,#0017H
LOOP_STM2: MOV A,R0
PUSH DPL
PUSH DPH
LCALL READ_DD
POP DPH
POP DPL
MOVX @DPTR,A
INC DPTR
INC R0
CJNE R0,#1AH,LOOP_STM2

```

```
SE_YEAR: MOV COMMAND_RTC,#8DH ; YEAR
```

```

LCALL READ_RTC
MOV A,OUTPUT_RTC
LCALL CONVER_ASCII

```

```

MOV DPTR,#0047H ;COL1
MOV A,COL1
MOVX @DPTR,A
MOV DPTR,#0046H ;COL2
MOV A,COL2
MOVX @DPTR,A
MOV A,#' ' ; SPACE
MOV DPTR,#0048H
MOVX @DPTR,A

```

```
;STORE END DATE & TIME
```

```
SE_DATE: MOV COMMAND_RTC,#87H ; DATE
```

```

LCALL READ_RTC
MOV A,OUTPUT_RTC
LCALL CONVER_ASCII
MOV DPTR,#0041H ;COL1
MOV A,COL1
MOVX @DPTR,A
MOV DPTR,#0040H ;COL2
MOV A,COL2
MOVX @DPTR,A

```

```

MOV A,#' ' ; -
MOV DPTR,#0042H
MOVX @DPTR,A

```

```
SE_HR: MOV COMMAND_RTC,#85H ; HR
```

```

LCALL READ_RTC
MOV A,OUTPUT_RTC
LCALL CONVER_ASCII

```

```

MOV DPTR,#004AH ;COL1
MOV A,COL1
MOVX @DPTR,A
MOV DPTR,#0049H ;COL2
MOV A,COL2
MOVX @DPTR,A

```

```

MOV A,#' ' ; :
MOV DPTR,#004BH
MOVX @DPTR,A

```

```
SE_MONTH: MOV COMMAND_RTC,#89H ; MONTH
```

```
SE_MIN: MOV COMMAND_RTC,#83H ; MIN
```

เอกสารนี้เป็นเอกสารตัวอย่างสำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่ควรนำไปใช้ประโยชน์ด้านการค้า  
 ไม่ควรแก้ไข หรือดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

LCALL READ_RTC
MOV A,OUTPUT_RTC
LCALL CONVER_ASCII

MOV DPTR,#004DH ;COL1
MOV A,COL1
MOVX @DPTR,A
MOV DPTR,#004CH, ;COL2
MOV A,COL2
MOVX @DPTR,A

MOV R0,61H

S_MT: MOV DPH,R0
MOV DPL,R1
MOVX A,@DPTR
INC R1

MOV DPH,64H ; START MEMORY ADDRESS
MOV DPL,65H
MOVX @DPTR,A
INC DPTR
MOV 64H,DPH
MOV 65H,DPL

CJNE R1,#50H,S_MT

MT0: CJNE R0,#00H,MT1
MOV DPTR,#0130H
LJMP TO_MT
MT1: CJNE R0,#01H,MT2
MOV DPTR,#0180H
LJMP TO_MT
MT2: CJNE R0,#02H,MT3
MOV DPTR,#01D0H
LJMP TO_MT
MT3: CJNE R0,#03H,MT4
MOV DPTR,#0220H
LJMP TO_MT
MT4: CJNE R0,#04H,MT5
MOV DPTR,#0270H
LJMP TO_MT
MT5: CJNE R0,#05H,MT6
MOV DPTR,#02C0H
LJMP TO_MT
MT6: CJNE R0,#06H,MT7
MOV DPTR,#0310H
LJMP TO_MT
MT7: CJNE R0,#07H,MT8
MOV DPTR,#0360H
LJMP TO_MT
MT8: CJNE R0,#08H,MT9
MOV DPTR,#03B0H
LJMP TO_MT
MT9: CJNE R0,#09H,MT0
MOV DPTR,#0400H

TO_MT: MOV 64H,DPH ; STORE MEMORY
MOV 65H,DPL

MOV R0,#00H ; START BUFFER (DPH)
MOV R1,#00H ; DPL

MOV DPH,62H ; INDEX MEMORY
MOV DPL,63H
MOV A,#00H
MOVX @DPTR,A

RET

; SAVE MESSAGE OF DATA MEMORY
SAVE_BUF: MOV DPH,66H ; START OF ADDRESS
MESSAGE
MOV DPL,67H
PUSH 00H
PUSH 01H
PUSH 02H

MOV R0,#00H
L_SAVEI: MOV A,R0
PUSH DPH
PUSH DPL
LCALL READ_DD
POP DPL
POP DPH

MOVX @DPTR,A
INC DPTR
MOV 66H,DPH
MOV 67H,DPL
INC R0

```

```

CJNE R0,#14H,L_SAVE2
MOV R0,#40H
L_SAVE2: CJNE R0,#54H,L_SAVE1

MOV R1,DPH
MOV R2,DPL
CJNE R1,#07H,CHK_S1
CJNE R2,#58H,CHK_S1
MOV 66H,#05H
MOV 67H,#00H

CHK_S1: POP 02H
POP 01H
POP 00H

RET

; SAVE MESSAGE OF DATA MEMORY (RX)

SAVE_RX: LCALL OFF_CURSOR
MOV DPH,66H ; START OF ADDRESS
MESSAGE
MOV DPL,67H

PUSH 00H
PUSH 01H
PUSH 02H

MOV R0,#14H
R_SAVE1: MOV A,R0

PUSH DPH
PUSH DPL
LCALL READ_DD
POP DPL
POP DPH

MOVX @DPTR,A
INC DPTR
MOV 66H,DPH
MOV 67H,DPL
INC R0

CJNE R0,#28H,R_SAVE2
MOV R0,#54H
R_SAVE2: CJNE R0,#68H,R_SAVE1

MOV R1,DPH
MOV R2,DPL
CJNE R1,#07H,CHK_R1
CJNE R2,#58H,CHK_R1
MOV 66H,#05H
MOV 67H,#00H

CHK_R1: POP 02H
POP 01H
POP 00H

MOV DPTR,#COMMAND
MOV A,R0
ORL A,#80H
MOVX @DPTR,A
LCALL WAIT
LCALL ON_CURSOR

RET

; SAVE MESSAGE OF RAM

SAVE_RAM: PUSH ACC
PUSH DPH
PUSH DPL
PUSH 00H
PUSH 01H
PUSH 02H
PUSH 03H
PUSH 04H
PUSH 05H

MOV R0,61H ; INDEX MEMORY

S_R0: CJNE R0,#00H,S_R1
MOV DPTR,#0800H
LJMP S_R_NEXT
S_R1: CJNE R0,#01H,S_R2
MOV DPTR,#0A58H
LJMP S_R_NEXT
S_R2: CJNE R0,#02H,S_R3
MOV DPTR,#0CB0H
LJMP S_R_NEXT
S_R3: CJNE R0,#03H,S_R4
MOV DPTR,#0F08H

```

```

LJMP S_R_NEXT
S_R4: CJNE R0,#04H,S_R5
      MOV DPTR,#1160H
      LJMP S_R_NEXT
S_R5: CJNE R0,#05H,S_R6
      MOV DPTR,#13B8H
      LJMP S_R_NEXT
S_R6: CJNE R0,#06H,S_R7
      MOV DPTR,#1610H
      LJMP S_R_NEXT
S_R7: CJNE R0,#07H,S_R8
      MOV DPTR,#1868H
      LJMP S_R_NEXT
S_R8: CJNE R0,#08H,S_R9
      MOV DPTR,#1AC0H
      LJMP S_R_NEXT
S_R9: CJNE R0,#09H,S_R0
      MOV DPTR,#1D18H
S_R_NEXT: MOV R0,#05H ; START OF BUFFER
(MESSAGE)
      MOV R1,#00H
      MOV R2,DPH ; START OF ADDRESS RAM
      MOV R3,DPL
      MOV R4,#60 ; LOOP 600
      MOV R5,#10
SAVE_DATA: MOV DPH,R0
      MOV DPL,R1
      MOVX A,@DPTP.
      INC DPTR
      MOV R0,DPH
      MOV R1,DPL
      MOV DPH,R2
      MOV DPL,R3
      MOVX @DPTR,A
      INC DPTR
      MOV R2,DPH
      MOV R3,DPL
      DJNZ R4,SAVE_DATA ; LOOP 600
      MOV R4,#60
      DJNZ R5,SAVE_DATA
      POP 05H
      POP 04H
      POP 03H
      POP 02H
      POP 01H
      POP 00H
      POP DPL
      POP DPH
      POP ACC
      RET
C_RAM: PUSH 00H ; CLEAR DATA OF EXTERNAL
RAM
      PUSH 01H
      PUSH DPH
      PUSH DPL
      MOV R0,#40
      MOV R1,#15
      MOV DPTR,#0500H ; STRAT ADDRESS OF
MESSAGE
      MOV A,#20H ; DATA = SPACE
C_R1: MOVX @DPTR,A
      INC DPTR
      DJNZ R0,C_R1
      MOV R0,#40
      DJNZ R1,C_R1
      POP DPL
      POP DPH
      POP 01H
      POP 00H
      RET
; STORE A
READ_DD: MOV DPTR,#COMMAND
      ORL A,#80H
      MOVX @DPTR,A
      LCALL WAIT
      MOV DPTR,#READDATA
      MOVX A,@DPTR
      LCALL WAIT
      RET

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

;*****
;SCANKEY 1-9
COMPARE_0: LCALL SCANKEY
NT_1: CJNE A,#31H,NT_2
      LJMP NT_END
NT_2: CJNE A,#32H,NT_3
      LJMP NT_END
NT_3: CJNE A,#33H,NT_4
      LJMP NT_END
NT_4: CJNE A,#34H,NT_5
      LJMP NT_END
NT_5: CJNE A,#35H,NT_6
      LJMP NT_END
NT_6: CJNE A,#36H,NT_7
      LJMP NT_END
NT_7: CJNE A,#37H,NT_8
      LJMP NT_END
NT_8: CJNE A,#38H,NT_9
      LJMP NT_END
NT_9: CJNE A,#39H,NT_10
      LJMP NT_END
NT_10: CJNE A,#27H,COMPARE_0
NT_END: RET

;*****
;SCANKEY 0-9
COMPARE_1: LCALL SCANKEY
NTC_0: CJNE A,#30H,NTC_1
      LJMP NTC_END
NTC_1: CJNE A,#31H,NTC_2
      LJMP NTC_END
NTC_2: CJNE A,#32H,NTC_3
      LJMP NTC_END
NTC_3: CJNE A,#33H,NTC_4
      LJMP NTC_END
NTC_4: CJNE A,#34H,NTC_5
      LJMP NTC_END
NTC_5: CJNE A,#35H,NTC_6
      LJMP NTC_END
NTC_6: CJNE A,#36H,NTC_7
      LJMP NTC_END
NTC_7: CJNE A,#37H,NTC_8
      LJMP NTC_END
NTC_8: CJNE A,#38H,NTC_9
      LJMP NTC_END
NTC_9: CJNE A,#39H,NTC_10
      LJMP NTC_END
NTC_10: CJNE A,#27H,COMPARE_1
NTC_END: RET

;*****
;SCANKEY 0-1
COMPARE_2: LCALL SCANKEY
NXC_1: CJNE A,#30H,NXC_2
      LJMP NXC_END
NXC_2: CJNE A,#31H,NXC_3
      LJMP NXC_END
NXC_3: CJNE A,#27H,COMPARE_2
NXC_END: RET

;*****
;SCANKEY 0-2
COMPARE_3: LCALL SCANKEY
NC_1: CJNE A,#30H,NC_2
      LJMP NC_END
NC_2: CJNE A,#31H,NC_3
      LJMP NC_END
NC_3: CJNE A,#32H,NC_4
      LJMP NC_END
NC_4: CJNE A,#27H,COMPARE_3
NC_END: RET

;*****
;SCANKEY 0-3
COMPARE_4: LCALL SCANKEY
NBC_0: CJNE A,#30H,NBC_1
      LJMP NBC_END
NBC_1: CJNE A,#31H,NBC_2
      LJMP NBC_END
NBC_2: CJNE A,#32H,NBC_3
      LJMP NBC_END
NBC_3: CJNE A,#33H,NBC_4
      LJMP NBC_END
NBC_4: CJNE A,#27H,COMPARE_4
NBC_END: RET

;*****
;SCANKEY 0-5
COMPARE_5: LCALL SCANKEY
NAC_0: CJNE A,#30H,NAC_1
      LJMP NAC_END
NAC_1: CJNE A,#31H,NAC_2
      LJMP NAC_END
NAC_2: CJNE A,#32H,NAC_3
      LJMP NAC_END
NAC_3: CJNE A,#33H,NAC_4
      LJMP NAC_END
NAC_4: CJNE A,#34H,NAC_5
      LJMP NAC_END

```

```

NAC_5: CJNE A,#35H,NAC_6
        LJMP NAC_END
NAC_6: CJNE A,#27H,COMPARE_5
NAC_END: RET

;*****
; HELP SECTION IS Q CODE
Q_CODE: MOV R0,#00H ; START VALUE
        MOV DPTR,#DIS_Q1 ; DISPLAY Q CODE
        LCALL PG_LCD
        LCALL DISP_Q1 ; LIFT UP RIGHT ENTER
        LCALL SET_DD ; FIRST ADDRESS DD RAM
        MOV A,#* ; WRITE * DD RAM
        LCALL WRITE_DD

Q_NEXT: LCALL SCANKEY
        CJNE A,#27H,Q_NEXT1 ; ESC
        LJMP PG_1

Q_NEXT1: CJNE A,#0DH,Q_NEXT2 ; ENTER
        LJMP EN_PAGE1

Q_NEXT2: CJNE A,#1AH,Q_NEXT3 ; RIGHT
        LJMP RIGHT

Q_NEXT3: CJNE A,#1BH,Q_NEXT4 ; LIFT
        LJMP LIFT

Q_NEXT4: CJNE A,#19H,Q_NEXT ; DOWN
        LJMP DOWN

; RIGHT
RIGHT: CJNE R0,#64H,NX_RG1
        LJMP Q_NEXT

NX_RG1: LCALL SET_DD
        MOV A,#'
        LCALL WRITE_DD
        MOV A,R0 ; R0=R0+4
        ADD A,#04H
        MOV R0,A
        CJNE R0,#14H,NX_ADD1
        MOV R0,#40H
        SJMP NX_ADD4

NX_ADD1: CJNE R0,#54H,NX_ADD2
        MOV R0,#14H
        SJMP NX_ADD4

NX_ADD2: CJNE R0,#28H,NX_ADD3
        MOV R0,#54H
        SJMP NX_ADD4

NX_ADD3: CJNE R0,#64H,NX_ADD4
        MOV R0,#60H
        NX_ADD4: LCALL SET_DD
        MOV A,#*
        LCALL WRITE_DD
        LJMP Q_NEXT

; LIFT
LIFT: CJNE R0,#00H,NX_LI1
        :
        LJMP Q_NEXT
NX_LI1: LCALL SET_DD
        MOV A,#'
        LCALL WRITE_DD
        MOV A,R0 ; R0=R0-4
        SUBB A,#04H
        MOV R0,A
        CJNE R0,#50H,NX_SUB1
        MOV R0,#24H
        SJMP NX_SUB3

NX_SUB1: CJNE R0,#10H,NX_SUB2
        MOV R0,#50H
        SJMP NX_SUB3

NX_SUB2: CJNE R0,#3CH,NX_SUB3
        MOV R0,#10H
        NX_SUB3: LCALL SET_DD
        MOV A,#*
        LCALL WRITE_DD
        LJMP Q_NEXT

; DOWN
DOWN: MOV R0,#00H
        MOV DPTR,#DIS_Q2 ; DISPLAY Q CODE
        LCALL PG_LCD
        LCALL DISP_Q2
        LCALL SET_DD ; FIRST ADDRESS DD RAM
        MOV A,#* ; WRITE * DD RAM
        LCALL WRITE_DD

NX_DOWN: LCALL SCANKEY
        CJNE A,#27H,NX_DOWN1 ; ESC
        LJMP PG_1

NX_DOWN1: CJNE A,#18H,NX_DOWN2 ; UP
        LJMP Q_CODE

NX_DOWN2: CJNE A,#1AH,NX_DOWN3 ; RIGHT
        LJMP RIGHT_D

NX_DOWN3: CJNE A,#1BH,NX_DOWN4 ; LIFT
        LJMP LIFT_D

NX_DOWN4: CJNE A,#0DH,NX_DOWN ; ENTER
        LJMP EN_PAGE2

```

```

; RIGHT DOWN
RIGHT_D: CJNE R0,#1CH,NX_RGD1
        LJMP NX_DOWN
NX_RGD1: LCALL SET_DD
        MOV A,#' '
        LCALL WRITE_DD
        MOV A,R0 ; R0=R0+4
        ADD A,#04H
        MOV R0,A
        CJNE R0,#14H,ND_ADD1
        MOV R0,#40H
        SJMP ND_ADD3
ND_ADD1: CJNE R0,#54H,ND_ADD2
        MOV R0,#14H
        SJMP ND_ADD3
ND_ADD2: CJNE R0,#1CH,ND_ADD3
        MOV R0,#18H
ND_ADD3: LCALL SET_DD
        MOV A,#'*'
        LCALL WRITE_DD
        LJMP NX_DOWN

; LIFT DOWN
LIFT_D: CJNE R0,#00H,ND_LI1
        LJMP NX_DOWN
ND_LI1: LCALL SET_DD
        MOV A,#' '
        LCALL WRITE_DD
        MOV A,R0 ; R0=R0-4
        SUBB A,#04H
        MOV R0,A
        CJNE R0,#10H,ND_SUB1
        MOV R0,#50H
        SJMP ND_SUB2
ND_SUB1: CJNE R0,#3CH,ND_SUB2
        MOV R0,#10H
ND_SUB2: LCALL SET_DD
        MOV A,#''
        LCALL WRITE_DD
        LJMP NX_DOWN

; INTRODUCTION
INTRO: PUSH 00H
        MOV R0,#00H
        MOV DPTR,#INTR1 ; PAGE 1
        LCALL PG_LCD
        LCALL DISP_INT1
INT_SCAN: LCALL SCANKEY
        CJNE A,#27H,N_INT1 ; ESC
        LJMP N_INT_END
N_INT1: CJNE A,#19H,N_INT2 ; UP
        INC R0
        AJMP N_NEXT_A
N_INT2: CJNE A,#18H,INT_SCAN ; DOWN
        DEC R0
N_NEXT_A: CJNE R0,#00H,N_P1
        MOV DPTR,#INTR1
        AJMP INTRX
N_P1: CJNE R0,#01H,N_P2
        MOV DPTR,#INTR2
        AJMP INTRX
N_P2: CJNE R0,#02H,N_P3
        MOV DPTR,#INTR3
        AJMP INTRX
N_P3: CJNE R0,#03H,N_P4
        MOV DPTR,#INTR4
        AJMP INTRX
N_P4: CJNE R0,#04H,N_P5
        MOV DPTR,#INTR5
        AJMP INTRX
N_P5: CJNE R0,#0FFH,N_P6
        MOV R0,#00H
        AJMP N_NEXT_A
N_P6: CJNE R0,#05H,N_NEXT_A
        MOV R0,#04H
        AJMP N_NEXT_A
INTRX: LCALL PG_LCD
N_K0: CJNE R0,#00H,N_K1
        LCALL DISP_INT1
        AJMP NNK
N_K1: CJNE R0,#01H,N_K2
        LCALL DISP_INT2
        AJMP NNK
N_K2: CJNE R0,#02H,N_K3
        LCALL DISP_INT2
        AJMP NNK
N_K3: CJNE R0,#03H,N_K4
        LCALL DISP_INT2
        AJMP NNK
N_K4: CJNE R0,#04H,N_K0
        LCALL DISP_INT3
NNK: AJMP INT_SCAN
N_INT_END: POP 00H

```

```

LJMP PG_I
;*****
; ENTER PAGE1
EN_PAGE1: CJNE R0,#00H,EN_NX1
MOV DPTR,#QRA
LJMP EN_END
EN_NX1: CJNE R0,#04H,EN_NX2
MOV DPTR,#QRB
LJMP EN_END
EN_NX2: CJNE R0,#08H,EN_NX3
MOV DPTR,#QRD
LJMP EN_END
EN_NX3: CJNE R0,#0CH,EN_NX4
MOV DPTR,#QRG
LJMP EN_END
EN_NX4: CJNE R0,#10H,EN_NX5
MOV DPTR,#QRH
LJMP EN_END
EN_NX5: CJNE R0,#40H,EN_NX6
MOV DPTR,#QRI
LJMP EN_END
EN_NX6: CJNE R0,#44H,EN_NX7
MOV DPTR,#QRJ
LJMP EN_END
EN_NX7: CJNE R0,#48H,EN_NX8
MOV DPTR,#QRK
LJMP EN_END
EN_NX8: CJNE R0,#4CH,EN_NX9
MOV DPTR,#QRL
LJMP EN_END
EN_NX9: CJNE R0,#50H,EN_NX10
MOV DPTR,#QRM
LJMP EN_END
EN_NX10: CJNE R0,#14H,EN_NX11
MOV DPTR,#QRN
LJMP EN_END
EN_NX11: CJNE R0,#18H,EN_NX12
MOV DPTR,#QRO
LJMP EN_END
EN_NX12: CJNE R0,#1CH,EN_NX13
MOV DPTR,#QRP
LJMP EN_END
EN_NX13: CJNE R0,#20H,EN_NX14
MOV DPTR,#QRQ
LJMP EN_END
EN_NX14: CJNE R0,#24H,EN_NX15
MOV DPTR,#QRS
LJMP EN_END
EN_NX15: CJNE R0,#54H,EN_NX16
MOV DPTR,#QRT
LJMP EN_END
EN_NX16: CJNE R0,#58H,EN_NX17
MOV DPTR,#QRU
LJMP EN_END
EN_NX17: CJNE R0,#5CH,EN_NX18
MOV DPTR,#QRX
LJMP EN_END
EN_NX18: CJNE R0,#60H,EN_END
MOV DPTR,#QRZ
EN_END: LCALL PG_LCD
LCALL SCANKEY
CJNE A,#27H,EN_END
LJMP Q_CODE
;*****
; ENTER PAGE2
EN_PAGE2: CJNE R0,#00H,EN_ND1
MOV DPTR,#QSA
LJMP EN_END_D
EN_ND1: CJNE R0,#04H,EN_ND2
MOV DPTR,#QSB
LJMP EN_END_D
EN_ND2: CJNE R0,#08H,EN_ND3
MOV DPTR,#QSD
LJMP EN_END_D
EN_ND3: CJNE R0,#0CH,EN_ND4
MOV DPTR,#QSL
LJMP EN_END_D
EN_ND4: CJNE R0,#10H,EN_ND5
MOV DPTR,#QSM
LJMP EN_END_D
EN_ND5: CJNE R0,#40H,EN_ND6
MOV DPTR,#QSO
LJMP EN_END_D
EN_ND6: CJNE R0,#44H,EN_ND7
MOV DPTR,#QSP
LJMP EN_END_D
EN_ND7: CJNE R0,#48H,EN_ND8
MOV DPTR,#QSV
LJMP EN_END_D
EN_ND8: CJNE R0,#4CH,EN_ND9
MOV DPTR,#QSX
LJMP EN_END_D
EN_ND9: CJNE R0,#50H,EN_ND10
MOV DPTR,#QSZ
LJMP EN_END_D
EN_ND10: CJNE R0,#14H,EN_ND11

```

```

MOV DPTR,#QTH
LJMP EN_END_D
EN_ND11: CJNE R0,#18H,EN_END_D
MOV DPTR,#QTR
EN_END_D: LCALL PG_LCD
LCALL SCANKEY
CJNE A,#27H,EN_END_D
LJMP DOWN

;*****
;SET ADDRESS DD RAM
SET_DD: PUSH DPL
PUSH DPH
PUSH ACC
MOV DPTR,#COMMAND
MOV A,R0
ORL A,#80H
MOVX @DPTR,A
LCALL WAIT
POP ACC
POP DPH
POP DPL
RET

; WRITE DD RAM (USE R0=ADDRESS)
WRITE_DD: PUSH DPL
PUSH DPH
MOV DPTR,#WRITEDATA
MOVX @DPTR,A
LCALL WAIT
POP DPH
POP DPL
RET

; ON/OFF CURSOR
ON_CURSOR: PUSH DPH
PUSH DPL
MOV DPTR,#COMMAND
MOV A,#0FH ; DISPLAY ON CONTROL
MOVX @DPTR,A
LCALL WAIT
POP DPL
POP DPH
RET

OFF_CURSOR: PUSH DPH
PUSH DPL

MOV DPTR,#COMMAND
MOV A,#0CH ; DISPLAY OFF CONTROL
MOVX @DPTR,A
LCALL WAIT
POP DPH
POP DPL
RET

; DISPLAY
DISPLAY: MOV R0,#02H
BACK_A: MOV DPTR,#DIS_1 ; PAGE 1
LCALL PG_LCD
LCALL DELAY_SEC
MOV DPTR,#DIS_2
LCALL PG_LCD
LCALL DELAY_SEC
DJNZ R0,BACK_A
MOV DPTR,#DIS_1
LCALL PG_LCD
LCALL DELAY_SEC
LCALL CLEAR
LCALL DELAY_SEC
MOV DPTR,#DIS_3 ; PAGE 2
LCALL PG_LCD
LCALL DELAY_SEC
LCALL DELAY_SEC
LCALL DELAY_SEC
RET

; DISP Q-CODE
DISP_Q1: PUSH 00H
PUSH ACC
MOV R0,#64H ; LIFT
LCALL SET_DD
MOV A,#7FH
LCALL WRITE_DD
MOV R0,#65H ; DOWN
LCALL SET_DD
MOV A,#01H
LCALL WRITE_DD
MOV R0,#66H ; RIGHT
LCALL SET_DD
MOV A,#7EH
LCALL WRITE_DD

```

```

MOV R0,#67H ;UP
LCALL SET_DD
MOV A,#02H
LCALL WRITE_DD
POP ACC
POP 00H
RET

```

```

MOV R0,#67H ;DOWN
LCALL SET_DD
MOV A,#01H
LCALL WRITE_DD
POP ACC
POP 00H

```

```
RET
```

```
DISP_Q2: PUSH 00H
```

```

PUSH ACC
MOV R0,#64H ;LIFT
LCALL SET_DD
MOV A,#7FH
LCALL WRITE_DD
MOV R0,#65H ;UP
LCALL SET_DD
MOV A,#00H
LCALL WRITE_DD
MOV R0,#66H ;RIGHT
LCALL SET_DD
MOV A,#7EH
LCALL WRITE_DD
MOV R0,#67H ;UP
LCALL SET_DD
MOV A,#02H
LCALL WRITE_DD
POP ACC
POP 00H
RET

```

```
DISP_INT3: PUSH 00H
```

```
PUSH ACC
```

```

MOV R0,#66H ;UP
LCALL SET_DD
MOV A,#00H
LCALL WRITE_DD
POP ACC
POP 00H
RET

```

```
; WRITE DATA TO LCD = 1 PAGE
```

```
PG_LCD: PUSH 00H
```

```

PUSH ACC
PUSH DPL
PUSH DPH
MOV DPTR,#COMMAND
MOV A,#00H
ORL A,#80H
MOVX @DPTR,A
LCALL WAIT

```

```
POP DPH
```

```
POP DPL
```

```
MOV R0,#00H
```

```
LCD_I: MOV A,R0
```

```
MOVC A,@A+DPTR
```

```
PUSH DPH
```

```
PUSH DPL
```

```
MOV DPTR,#WRITEDATA
```

```
MOVX @DPTR,A
```

```
LCALL WAIT
```

```
POP DPL
```

```
POP DPH
```

```
INC R0
```

```
CJNE R0,#80,LCD_I
```

```
POP ACC
```

```
POP 00H
```

```
RET
```

```
; DISPLAY INTRODUCTION
```

```
DISP_INT1: PUSH 00H
```

```
PUSH ACC
```

```
MOV R0,#67H ;DOWN
```

```
LCALL SET_DD
```

```
MOV A,#01H
```

```
LCALL WRITE_DD
```

```
POP ACC
```

```
POP 00H
```

```
RET
```

```
DISP_INT2: PUSH 00H
```

```
PUSH ACC
```

```
MOV R0,#66H ;UP
```

```
LCALL SET_DD
```

```
MOV A,#00H
```

```
LCALL WRITE_DD
```

```

; CLEAR LCD = 1 PAGE
CLEAR:  PUSH DPH
        PUSH DPL
        PUSH ACC
        MOV DPTR,#COMMAND
        MOV A,#01
        MOVX @DPTR,A
        LCALL WAIT
        POP ACC
        POP DPL
        POP DPH
        RET

; END OF FILE
PC_E:   PUSH DPH
        PUSH DPL
        PUSH 00H
        MOV R0,#20
        MOV DPTR,#PC2
L_PC_E: CLR A
        MOVC A,@A+DPTR
        INC DPTR
        MOV SBUF,A
        JNB TI,$
        CLR TI
        DJNZ R0,L_PC_E
        POP 00H
        POP DPL
        POP DPH
        RET

; MESSAGE
PC_M:   PUSH DPH
        PUSH DPL
        PUSH 00H
        MOV R0,#20
        MOV DPTR,#PC3
L_PC_M: CLR A
        MOVC A,@A+DPTR
        INC DPTR
        MOV SBUF,A
        JNB TI,$
        CLR TI
        DJNZ R0,L_PC_M
        POP 00H
        POP DPL

; LINK DATA TO PC
; HEADING OF FILE
PC_H:   PUSH DPH
        PUSH DPL

```

```

POP DPH
RET

;-----
; DELAY 1 SEC

DELAY_SEC:  PUSH 02H
            PUSH 01H
            PUSH 00H
            MOV  R2,#100

DEL_2:      MOV  R1,#100
DEL_1:      MOV  R0,#50
            DJNZ R0,$
            DJNZ R1,DEL_1
            DJNZ R2,DEL_2
            POP  00H
            POP  01H
            POP  02H
            RET

;-----
; DELAY 1 mS

DELAY_MS:   PUSH 00H
            PUSH 01H
            MOV  R0,#0FH

DEL_MS:     MOV  R1,#00H
            DJNZ R1,$
            DJNZ R0,DEL_MS
            POP  01H
            POP  00H
            RET

;-----
; DELAY LOG

DELAY_LOG:  PUSH 00H
            PUSH 01H
            MOV  R0,#00H

DEL_LOG:    MOV  R1,#00H
            DJNZ R1,$
            DJNZ R0,DEL_LOG
            POP  01H
            POP  00H
            RET

;-----
DELAY_SW:   PUSH 00H
            PUSH 01H
            MOV  R1,#100

LP_SW:      MOV  R0,#0
            DJNZ R0,$
            DJNZ R1,LP_SW
            POP  01H
            POP  00H
            RET

;-----
; ADD FONT
; 00H = UP
; 01H = DOWN
; 02H = ENTER

FONT:       PUSH ACC
            PUSH DPH
            PUSH DPL
            PUSH 00H
            PUSH 01H
            PUSH 02H
            MOV  R0,#00H ; UP
            MOV  R1,#00H
            MOV  R2,#08H

LF1:        MOV  DPTR,#UP
            MOV  A,R0
            MOVC A,@A+DPTR
            MOV  30H,A
            PUSH DPH
            PUSH DPL
            MOV  DPTR,#COMMAND
            MOV  A,R1
            ORL  A,#40H
            MOVX @DPTR,A
            LCALL WAIT
            MOV  DPTR,#WRITEDATA
            MOV  A,30H
            MOVX @DPTR,A
            LCALL WAIT
            INC  R1
            POP  DPL
            POP  DPH
            INC  R0
            DJNZ R2,LF1
            MOV  R0,#00H ; DOWN
            MOV  R1,#48H
            MOV  R2,#08H

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับกองในงานเพื่อการศึกษาเท่านั้น ไม่สามารถเผยแพร่โดยไม่ได้รับอนุญาตให้ใช้ประโยชน์ด้านการค้า  
 ไม่มีการแก้ไข ทังสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงเอกสารทุกครั้งที่มีการนำไปใช้

```

LF2:    MOV  DPTR,#DW
        MOV  A,R0
        MOVC A,@A+DPTR
        MOV  30H,A
        PUSH DPH
        PUSH DPL
        MOV  DPTR,#COMMAND
        MOV  A,R1
        ORL  A,#40H
        MOVX @DPTR,A
        LCALL WAIT
        MOV  DPTR,#WRITEDATA
        MOV  A,30H
        MOVX @DPTR,A
        LCALL WAIT
        INC  R1
        POP  DPL
        POP  DPH
        INC  R0
        DJNZ R2,LF2
        MOV  R0,#00H ; ENTER
        MOV  R1,#50H
        MOV  R2,#08H

LF3:    MOV  DPTR,#ENT
        MOV  A,R0
        MOVC A,@A+DPTR
        MOV  30H,A
        PUSH DPH
        PUSH DPL
        MOV  DPTR,#COMMAND
        MOV  A,R1
        ORL  A,#40H
        MOVX @DPTR,A
        LCALL WAIT
        MOV  DPTR,#WRITEDATA
        MOV  A,30H
        MOVX @DPTR,A
        LCALL WAIT
        INC  R1
        POP  DPL
        POP  DPH
        INC  R0
        DJNZ R2,LF3
        POP  02H
        POP  01H
        POP  00H
        POP  DPL

        POP  DPH
        POP  ACC
        RET

;*****
; SCAN KEYBOARD 16x4 (COLxROW) ROW=MSB(PA) :
; COL=LSB(PB)
TABEL_KEY: DB 0E0H,0E1H,0E2H,0E3H ; 00,01,02,03
            DB 0E4H,0E5H,0E6H,0E7H ; 04,05,06,07
            DB 0E8H,0E9H,0EAH,0EBH ; 08,09,10,11
            DB 0ECH,0EDH,0EEH,0EFH ; 12,13,14,15
            DB 0D0H,0D1H,0D2H,0D3H ; 16,17,18,19
            DB 0D4H,0D5H,0D6H,0D7H ; 20,21,22,23
            DB 0D8H,0D9H,0DAH,0DBH ; 24,25,26,27
            DB 0DCH,0DDH,0DEH,0DFH ; 28,29,30,31
            DB 0B0H,0B1H,0B2H,0B3H ; 32,33,34,35
            DB 0B4H,0B5H,0B6H,0B7H ; 36,37,38,39
            DB 0B8H,0B9H,0BAH,0BBH ; 40,41,42,43
            DB 0BCH,0BDH,0BEH,0BFH ; 44,45,46,47
            DB 070H,071H,072H,073H ; 48,49,50,51
            DB 074H,075H,076H,077H ; 52,53,54,55
            DB 078H,079H,07AH,07BH ; 56,57,58,59
            DB 07CH,07DH,07EH,07FH ; 60,61,62,63

;*****
; DISPLAY LCD
DIS_1: DB "*****"
        DB " ED.ENGINEER 19 "
        DB " K M I T'L "
        DB "*****"

DIS_2: DB "*****"
        DB " "
        DB " "
        DB "*****"

DIS_3: DB "Packet Radio Control"
        DB "for Amature Radio. "
        DB "by Microcontroller"
        DB " "

DIS_4: DB " F1 Help "
        DB " F3 Mode "
        DB " F2 Command "
        DB " F4 Setup "

DIS_5: DB "<Help>" ; F1
        DB " 2. Introduction "
        DB " 1. Q-Code "

```

|          |                                                                                   |                          |               |         |                        |                                         |
|----------|-----------------------------------------------------------------------------------|--------------------------|---------------|---------|------------------------|-----------------------------------------|
| DB       | "                                                                                 |                          |               | DB      | "END:                  | "                                       |
| DIS_6:   | DB                                                                                | "<Command>               | " ; F2        | DIS_Q1: | DB                     | "QRA QRB QRD QRG QRH" ; PAGE1 QR?       |
| DB       | "2. Log Book                                                                      | "                        |               | DB      | "QRN QRO QRP QRQ QRS"  |                                         |
| DB       | "1. Time                                                                          | "                        |               | DB      | "QRI QRJ QRK QRL QRM"  |                                         |
| DB       | "                                                                                 | "                        |               | DB      | "QRT QRU QRX QRZ "     |                                         |
| DIS_7:   | DB                                                                                | "<Mode>                  | " ; F3        | DIS_Q2: | DB                     | "QSA QSB QSD QSL QSM" ; PAGE2 QS? & QT? |
| DB       | "2. Standby                                                                       | "                        |               | DB      | "QTH QTR "             |                                         |
| DB       | "1. Connect                                                                       | "                        |               | DB      | "QSO QSP QSV QSX QSZ"  |                                         |
| DB       | "                                                                                 | "                        |               | DB      | "                      |                                         |
| DIS_8:   | DB                                                                                | "<Setup>                 | " ; F4        | QRA:    | DB                     | "What is the name of "                  |
| DB       | "2. Timer                                                                         | "                        |               | DB      | "                      |                                         |
| DB       | "1. Date                                                                          | "                        |               | DB      | "your station? "       |                                         |
| DB       | "3. Callsign                                                                      | "                        |               | DB      | "Esc:Return "          |                                         |
| DIS_9:   | DB                                                                                | "**** SETUP DATE ****"   |               | QRB:    | DB                     | "How far approximate "                  |
| DB       | "*DATE: - - *                                                                     | "                        |               | DB      | "station? "            |                                         |
| DB       | "*DATE:DD-MM-YY *                                                                 | "                        |               | DB      | "are you from my "     |                                         |
| DB       | "*****"                                                                           |                          |               | DB      | "Esc:Return "          |                                         |
| DIS_10:  | DB                                                                                | "**** SETUP TIMER ****"  |               | QRD:    | DB                     | "Where are you bound "                  |
| DB       | "*TIMER: : *                                                                      | "                        |               | DB      | "from? "               |                                         |
| DB       | "*TIMER: : : *                                                                    | "                        |               | DB      | "and where are you"    |                                         |
| DB       | "*****"                                                                           |                          |               | DB      | "Esc:Return "          |                                         |
| DIS_11:  | DB                                                                                | "*** SETUP CALLSIGN ***" |               | QRG:    | DB                     | "Will you tell me my "                  |
| DB       | "*CALLSIGN: *                                                                     | "                        |               | DB      | "KHz? "                |                                         |
| DB       | "*CALLSIGN: *                                                                     | "                        |               | DB      | "exact frequency in "  |                                         |
| DB       | "*****"                                                                           |                          |               | DB      | "Esc:Return "          |                                         |
| DIS_12:  | DB                                                                                | "<Standby>               | "             | QRH:    | DB                     | "Does my frequency "                    |
| DB       | "                                                                                 | "                        |               | DB      | "                      |                                         |
| DB       | "                                                                                 | "                        |               | DB      | "vary? "               |                                         |
| DB       | "                                                                                 | "                        |               | DB      | "Esc:Return "          |                                         |
| DIS_13:  | DB                                                                                | "<Log Book>              | "             | QRI:    | DB                     | "Is my note good? "                     |
| DB       | "                                                                                 | "                        |               | DB      | "                      |                                         |
| DB       | "0 1 2 3 4 5 6 7 8 9"                                                             |                          |               | DB      | "                      |                                         |
| DB       | "O:Open D:Del L:Link "                                                            |                          |               | DB      | "Esc:Return "          |                                         |
| DIS_CLK: | DB                                                                                | "***** TIME *****"       | DISPLAY CLOCK | QRJ:    | DB                     | "Do you receive me"                     |
| DB       | "* *                                                                              | "                        |               | DB      | "Are my signals weak?" |                                         |
| DB       | "* *                                                                              | "                        |               | DB      | "badly? "              |                                         |
| DB       | "*****"                                                                           |                          |               | DB      | "Esc:Return "          |                                         |
| FORM:    | DB                                                                                | "FROM: "                 |               | QRK:    | DB                     | "Do you receive me"                     |
| DB       | "TO: "                                                                            |                          |               |         |                        |                                         |
| DB       | "START: อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและตัวอักษรถึงแม้จะขอจนกระทั่งมีการนำไปใช้" |                          |               |         |                        |                                         |

|         |                        |         |                        |
|---------|------------------------|---------|------------------------|
| DB      | "Are my signals good?" | DB      | "me again? "           |
| DB      | "well? "               | DB      | "When will you Call "  |
| DB      | "Esc:Return "          | DB      | "Esc:Return "          |
| QRL: DB | "Are you busy? "       | QRZ: DB | "Who is calling me? "  |
| DB      | " "                    | DB      | " "                    |
| DB      | " "                    | DB      | " "                    |
| DB      | "Esc:Return "          | DB      | "Esc:Return "          |
| QRM: DB | "Ary you being "       | QSA: DB | "What is the strength" |
| DB      | " "                    | DB      | " "                    |
| DB      | "interfered with? "    | DB      | "of my signals? "      |
| DB      | "Esc:Return "          | DB      | "Esc:Return "          |
| QRN: DB | "Are you troubled by " | QSB: DB | "Does the strength of" |
| DB      | " "                    | DB      | " "                    |
| DB      | "atmosherics? "        | DB      | "my signals vary? "    |
| DB      | "Esc:Return "          | DB      | "Esc:Return "          |
| QRO: DB | "Shall I increase "    | QSD: DB | "Is my keying correct" |
| DB      | " "                    | DB      | "Signals distinct? "   |
| DB      | "power? "              | DB      | "? "                   |
| DB      | "Esc:Return "          | DB      | "Esc:Return "          |
| QRP: DB | "Shall I decrease "    | QSL: DB | "Can you give me "     |
| DB      | " "                    | DB      | "receipt? "            |
| DB      | "power? "              | DB      | "acknowledgement of "  |
| DB      | "Esc:Return "          | DB      | "Esc:Return "          |
| QRQ: DB | "Shall I send faster?" | QSM: DB | "Shall I repeat the "  |
| DB      | " "                    | DB      | "you? "                |
| DB      | " "                    | DB      | "last telegram I sent" |
| DB      | "Esc:Return "          | DB      | "Esc:Return "          |
| QRS: DB | "Shall I send more"    | QSO: DB | "Can you communicate " |
| DB      | " "                    | DB      | " "                    |
| DB      | "slowly? "             | DB      | "with...direct? "      |
| DB      | "Esc:Return "          | DB      | "Esc:Return "          |
| QRT: DB | "Shall I stop sending" | QSP: DB | "Will you relay to..?" |
| DB      | " "                    | DB      | " "                    |
| DB      | "? "                   | DB      | " "                    |
| DB      | "Esc:Return "          | DB      | "Esc:Return "          |
| QRU: DB | "Have you anything "   | QSV: DB | "Shall I send a "      |
| DB      | "Are you ready? "      | DB      | " "                    |
| DB      | "for me? "             | DB      | "series of V's? "      |
| DB      | "Esc:Return "          | DB      | "Esc:Return "          |
| QRX: DB | "Shall I wait? "       | QSX: DB | "Will you listen for." |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

|        |                       |                        |      |                        |
|--------|-----------------------|------------------------|------|------------------------|
| DB     | "                     | INTR5:                 | DB   | "supported by the "    |
| DB     | ".(calling) on..KHz?" |                        | DB   | "radio. "              |
| DB     | "Esc:Return "         |                        | DB   | "bandwidth of amature" |
|        |                       |                        | DB   | "Esc:Return "          |
| QSZ:   | DB                    | "Shall I send each"    |      |                        |
|        | DB                    | "                      | PC1: | DB                     |
|        | DB                    | "word or group twice?" | PC2: | DB                     |
|        | DB                    | "Esc:Return "          | PC3: | DB                     |
|        |                       |                        |      |                        |
| QTH:   | DB                    | "What's your position" |      |                        |
|        | DB                    | "longitude? "          |      |                        |
|        | DB                    | "in latitude and "     |      |                        |
|        | DB                    | "Esc:Return "          |      |                        |
|        |                       |                        |      |                        |
| QTR:   | DB                    | "What is the exact"    |      |                        |
|        | DB                    | "                      |      |                        |
|        | DB                    | "time? "               |      |                        |
|        | DB                    | "Esc:Return "          |      |                        |
|        |                       |                        |      |                        |
| INTR1: | DB                    | "Packet radio control" |      |                        |
|        | DB                    | "for amature radio. "  |      |                        |
|        | DB                    | "by microcontroller"   |      |                        |
|        | DB                    | "Esc:Return "          |      |                        |
|        |                       |                        |      |                        |
| INTR2: | DB                    | "It's half duplex "    |      |                        |
|        | DB                    | "contain a single "    |      |                        |
|        | DB                    | "system. This section" |      |                        |
|        | DB                    | "Esc:Return "          |      |                        |
|        |                       |                        |      |                        |
| INTR3: | DB                    | "chip asynchronous "   |      |                        |
|        | DB                    | "applicable CCITT V23" |      |                        |
|        | DB                    | "FSK with the "        |      |                        |
|        | DB                    | "Esc:Return "          |      |                        |
|        |                       |                        |      |                        |
| INTR4: | DB                    | "standards. Operation" |      |                        |
|        | DB                    | " This equipment "     |      |                        |
|        | DB                    | "up to 1200 bps TX&RX" |      |                        |
|        | DB                    | "Esc:Return "          |      |                        |
|        |                       |                        |      |                        |

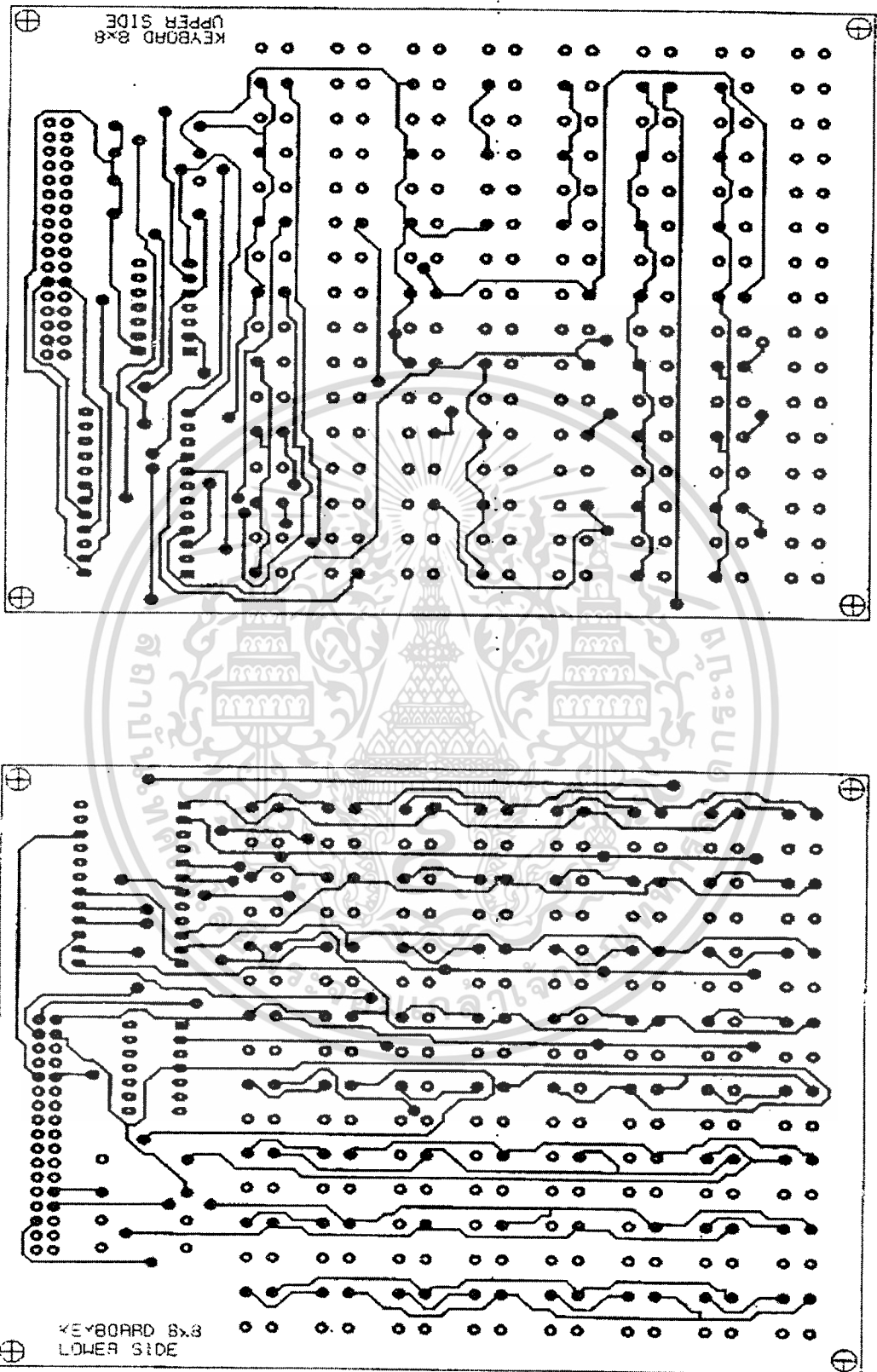
เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ของ **รูปที่ ข.14 โปรแกรมควบคุมการทำงานของแพคเกจรีโอ** โดยใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



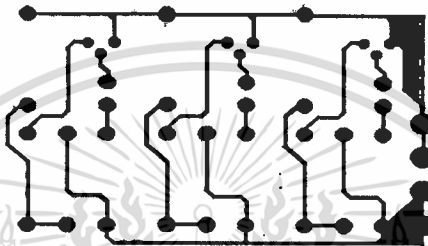
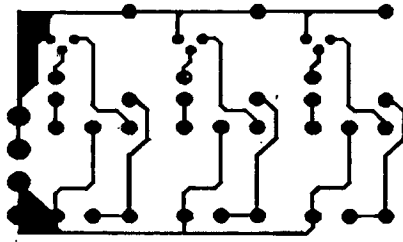
ภาคผนวก ก

วงจร และแผ่นวงจรพิมพ์

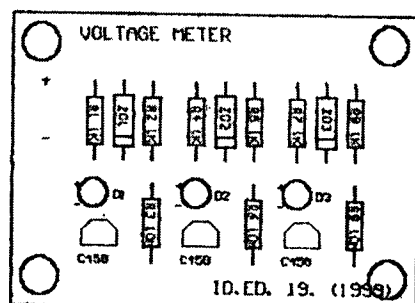
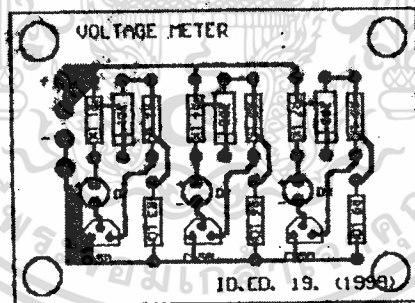
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



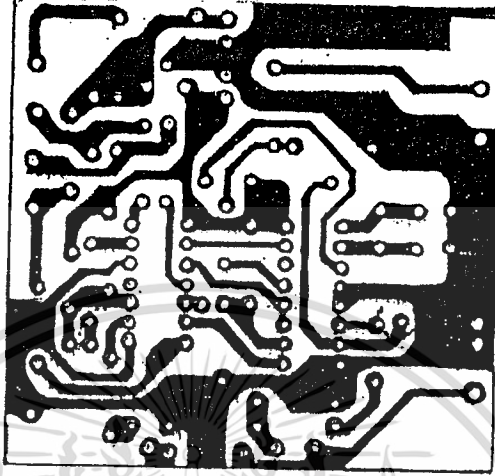
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามนำรูปที่ ค.1 ลายวงจรพิมพ์ซึ่งเป็นพิมพ์ตัวอักษรเอกสารทุกครั้งที่มีการนำไปใช้



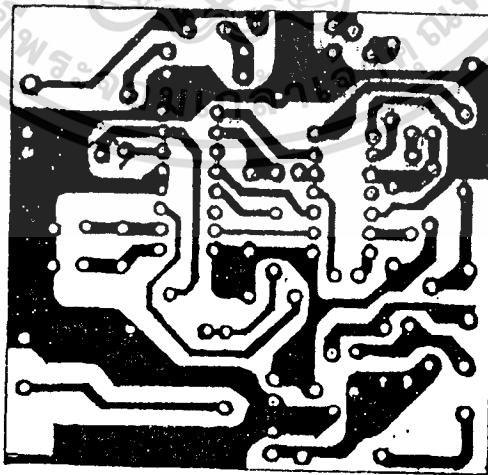
รูปที่ ค.2 ลายวงจรพิมพ์ของวงจรวัดระดับแรงดัน



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งรูปที่ ค.3 การวางอุปกรณ์ของวงจรวัดระดับแรงดัน สารทุกครั้งที่มีการนำไปใช้



รูปที่ ค.4 ด้านหน้าลายวงจรพิมพ์ของวงจรเพกเกจ



รูปที่ ค.5 ด้านหลังลายวงจรพิมพ์ของวงจรเพกเกจ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งยังห้ามมิให้เผยแพร่เอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 1. วงจรแพกเกจ

อุปกรณ์ที่ใช้ในวงจรแพกเกจมีดังนี้

|                        |       |
|------------------------|-------|
| 1. R 100 $\Omega$      | 2 ตัว |
| 2. R 2.2 K $\Omega$    | 2 ตัว |
| 3. R 4.7 K $\Omega$    | 2 ตัว |
| 4. R 10 K $\Omega$     | 6 ตัว |
| 5. R 15 K $\Omega$     | 2 ตัว |
| 6. R 33 K $\Omega$     | 2 ตัว |
| 7. R 68 K $\Omega$     | 2 ตัว |
| 8. R 100 K $\Omega$    | 2 ตัว |
| 9. C 0.1 $\mu$ F       | 6 ตัว |
| 10. C 10 $\mu$ F       | 2 ตัว |
| 11. C 22 $\mu$ F       | 2 ตัว |
| 12. C 20 pF            | 2 ตัว |
| 13. C 0.1 nF           | 2 ตัว |
| 14. C 22 nF            | 2 ตัว |
| 15. Jack Specker       | 2 ตัว |
| 16. Crystal 4.43361MHz | 2 ตัว |
| 17. LED Power          | 2 ตัว |
| 18. Jack Mic           | 2 ตัว |
| 19. TR 2SC458          | 2 ตัว |
| 20. IC TCM3101         | 2 ตัว |

## 2. วงจรวัดระดับแรงดัน

อุปกรณ์ที่ใช้ในวงจรวัดระดับแรงดันมีดังนี้

|                                 |        |
|---------------------------------|--------|
| 1. Zener Diode 6.2V, 7.5V, 9.1V | 6 ตัว  |
| 2. TR C458                      | 6 ตัว  |
| 3. R 1 K $\Omega$               | 12 ตัว |
| 4. R 10 K $\Omega$              | 6 ตัว  |
| 5. LED เปล่งแสง                 | 6 ตัว  |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีกรนำไปใช้

3. แป้นพิมพ์ตัวอักษร

อุปกรณ์ที่ใช้ในแป้นพิมพ์ตัวอักษรมีดังนี้

- |               |         |
|---------------|---------|
| 1. IC 74LS154 | 2 ตัว   |
| 2. IC 74LS21  | 2 ตัว   |
| 3. R 10 KΩ    | 8 ตัว   |
| 4. คริปสวิทช์ | 128 ตัว |

4. บอร์ดไมโครคอนโทรลเลอร์

|                                               |         |
|-----------------------------------------------|---------|
| บอร์ดไมโครคอนโทรลเลอร์ รุ่น ET - BASIC32 V3.0 | 2 บอร์ด |
|-----------------------------------------------|---------|

5. จอ LCD แสดงผล

|                           |       |
|---------------------------|-------|
| จอ LCD แสดงผล รุ่น DMC204 | 2 อัน |
|---------------------------|-------|



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



## คู่มือประกอบการใช้งาน

วิทยุแพคเกจ ควบคุมโดยไมโครคอนโทรลเลอร์ สำหรับวิทยุสมัครเล่น  
PACKET RADIO CONTROL BY MICROCONTROLLER FOR  
AMATEUR RADIO

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# คู่มือประกอบการใช้งาน

## วิทยุแพกเกจ ควบคุมโดยไมโครคอนโทรลเลอร์ สำหรับวิทยุสมัครเล่น

### 1. บทนำ

เครื่องควบคุมการส่งข้อมูล วิทยุแพกเกจ ควบคุมโดยไมโครคอนโทรลเลอร์ สำหรับวิทยุสมัครเล่น อาศัยหลักการประยุกต์แบบไมโครคอนโทรลเลอร์ เพื่อช่วยในการควบคุม ประมวลผล สัญญาณข่าวสารเช่นเดียวกับระบบไมโครคอมพิวเตอร์ทั่วไปที่นำมาใช้ ในการสื่อสารระบบวิทยุแพกเกจสำหรับวิทยุสมัครเล่น ช่วยให้ลดทอนในการใช้การสื่อสารวิทยุแพกเกจเป็นอย่างมาก และเป็นพื้นฐานในการศึกษาเพื่อพัฒนาการสื่อสารในระบบวิทยุแพกเกจต่อไปในอนาคต นอกจากนี้ยังมีคุณสมบัติอื่นๆอีก คือ

1. รับส่งสัญญาณข่าวสาร ผ่านเครื่องวิทยุรับส่ง ย่านความถี่สมัครเล่น
2. บันทึกข้อมูลการสนทนา (Log Book) ได้ 10 ครั้ง
3. ถ่ายโอนข้อมูลบันทึกการสนทนา (Log Book) สู่ระบบคอมพิวเตอร์ เพื่อบันทึกและพิมพ์ และพิมพ์ออกมาในรูปแบบลายลักษณ์อักษร
4. ป้อนข้อมูลและรับข้อความเป็นตัวอักษรและตัวเลขได้ 20 ตัวอักษร
5. แสดงเมนูช่วยบอกรหัส Q - Code ต่างๆ ได้
6. เลือกตั้งค่า วัน - เดือน - ปี เวลาปัจจุบันและชื่อสถานีได้เอง
7. มีหลอดไฟฉายให้การแสดงผลในเวลากลางคืนชัดเจนยิ่งขึ้น

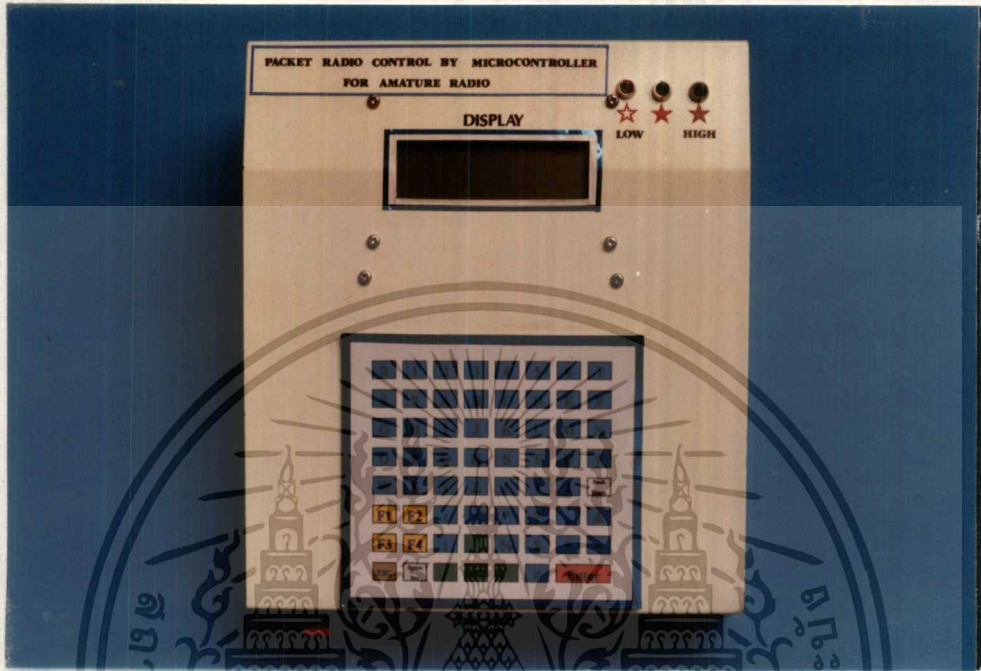
### 2. ส่วนประกอบและอุปกรณ์

1. เครื่องควบคุมการส่งข้อมูลวิทยุแพกเกจ ควบคุมโดยไมโครคอนโทรลเลอร์ สำหรับวิทยุสมัครเล่น
2. หม้อแปลง แรงดัน DC 9 V
3. แบตเตอรี่ภายนอก 9.6 V 800mA
4. สายต่อกับวิทยุสื่อสาร
5. สายต่อกับพอร์ต COM1 และ COM2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

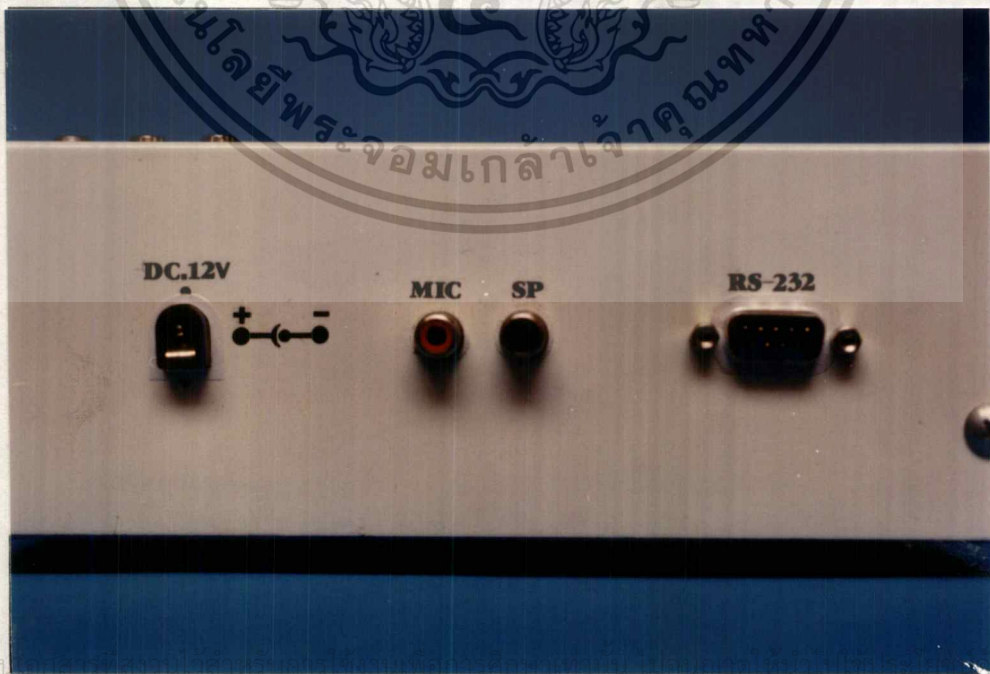
### 3. ส่วนด้านต่างๆของเครื่องวิทยุแพคเกจ

#### 3.1 ส่วนด้านบนเครื่อง



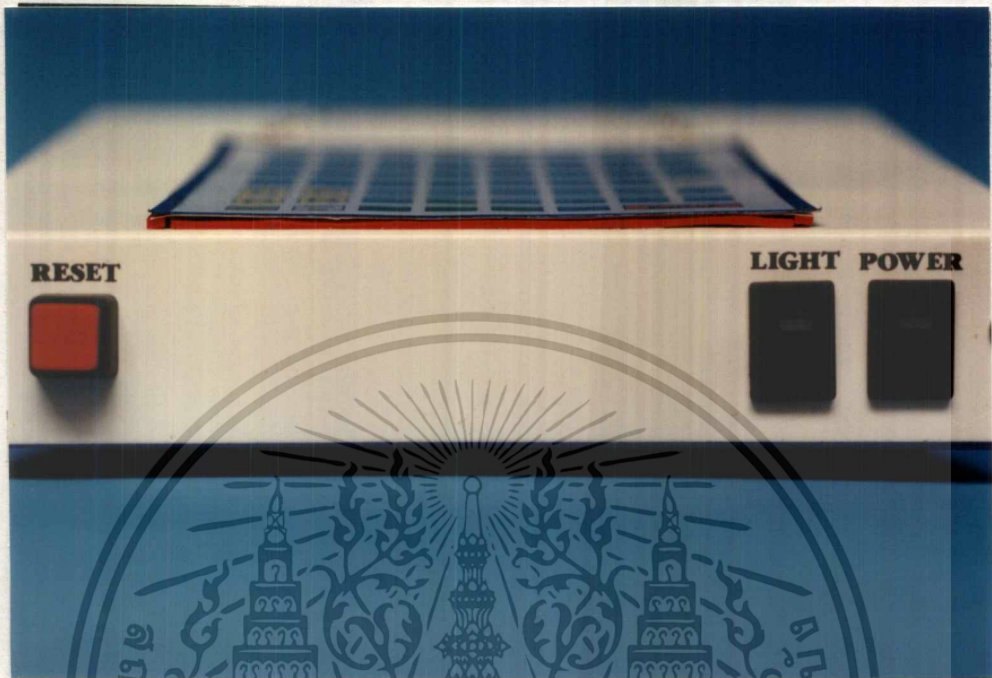
รูปที่ 1 ส่วนด้านบนเครื่อง

#### 3.2 ส่วนด้านหลังเครื่อง



เอกสารนี้เป็นเอกสารของสำนักงานคณะกรรมการการ  
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกหรือทำซ้ำโดยไม่ได้รับอนุญาตจากเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.3 ส่วนด้านหน้าเครื่อง



รูปที่ 3 ส่วนด้านหน้าเครื่อง

### 4. หน้าที่ของอุปกรณ์ต่างๆ

1. Reset Switch ทำหน้าที่ Reset การทำงานของเครื่องทั้งหมด เมื่อเริ่มต้นการทำงานใหม่ทั้งหมดอีกครั้งหนึ่ง ซึ่งจะใช้ในกรณีที่ผลการทำงานเกิดผลขัดข้องแล้วทำการ Reset เครื่องใหม่

2. Light Switch ทำหน้าที่เปิดไฟ เพื่อให้จอแสดงผลมีความสว่างเพียงพอที่จะแสดงผลในที่มืดได้ ในที่นี้จะป็นลักษณะการเพิ่มความชัดเจนของตัวอักษร ขณะทำงานอยู่ในที่ที่มีแสงไฟไม่เพียงพอ

3. Power Switch ทำหน้าที่ เปิด/ปิด เครื่องทั้งหมดให้สามารถทำงานได้

4. ขั้วต่อหม้อแปลงไฟฟ้า เพื่อรับพลังงานจากภายนอก รวมถึงต่อกับแบตเตอรี่ ที่ระดับแรงดันประมาณ 7.2 V - 9.6 V กระแส 800 mA - 1,200 mA

5. ขั้วต่อ RCA เพื่อต่อกับเครื่องรับวิทยุรับส่งย่านความถี่วิทยุสมัครเล่น 144 - 146 MHz ใช้

งานร่วมกับสายต่อวิทยุสื่อสารเข้าทางไมโครโฟน และ ลำโพง ของวิทยุสื่อสาร

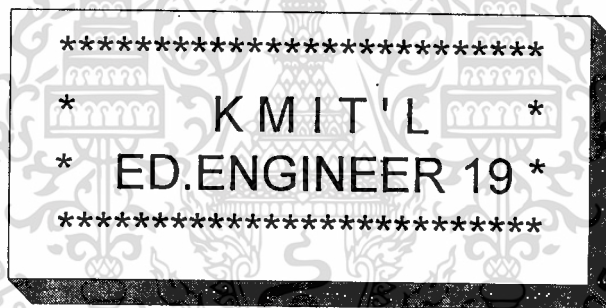
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6. ขั้วต่อ DB - 9 ทำหน้าที่ต่อสัญญาณ เมื่อ LINK กับเครื่องคอมพิวเตอร์ผ่านทางพอร์ต COM1 และ COM2
7. จอแสดงผล LCD 20 ตัวอักษร X 4 บรรทัด ทำหน้าที่แสดงผลการทำงานและแสดงสถานะการทำงานต่างๆ ของเครื่อง
8. แป้นพิมพ์ ทำหน้าที่ ป้อนคำสั่งต่างๆ เพื่อใช้ควบคุมการทำงานทั้งหมดของเครื่อง
9. ไดโอดเปล่งแสง แสดงผลสถานะ พลังงานของแบตเตอรี่ที่มีอยู่ 3 สถานะ คือ High และ Medium และ Low

## 5. การใช้งานของแพคเกจรีโอ

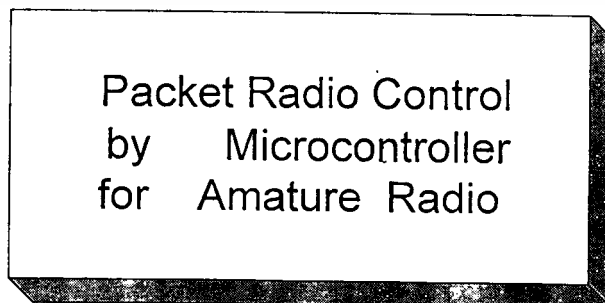
การใช้งานของแพคเกจรีโอ จะมีลำดับขั้นตอนการทำงานของระบบ เป็นขั้นดังต่อไปนี้

1. เมื่อเปิดสวิตช์ให้เครื่องทำงาน จอแสดงผลจะมีข้อความกระพริบ 3 ครั้ง ดังรูปที่ 4



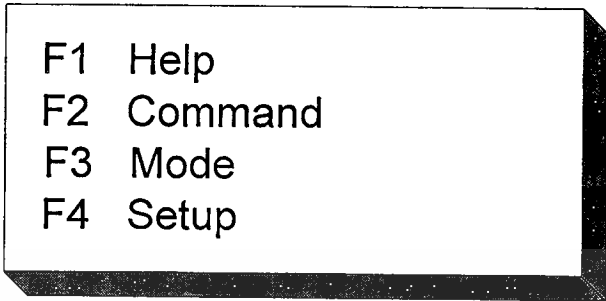
รูปที่ 4 ข้อความเริ่มต้น เมื่อเปิดสวิตช์ให้เครื่องทำงาน

2. หลังจากนั้นจอแสดงผลจะแสดงข้อความเกี่ยวกับชื่อปริณญานพนธ์ที่ทำการรูปที่ 5



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
รูปที่ 5 ข้อความหลังจากที่เปิดสวิตช์ต่อจากข้อความแรก  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

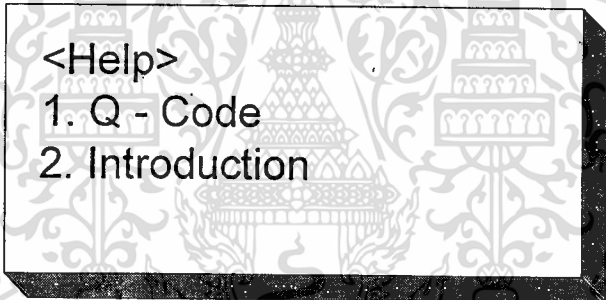
3. เครื่องจะเข้าสู่รายการเลือกให้ผู้ใช้เลือกการทำงาน 4 ฟังก์ชันดังรูปที่ 6



F1 Help  
F2 Command  
F3 Mode  
F4 Setup

รูปที่ 6 รายการเลือกในการเลือกใช้งาน 4 ฟังก์ชัน

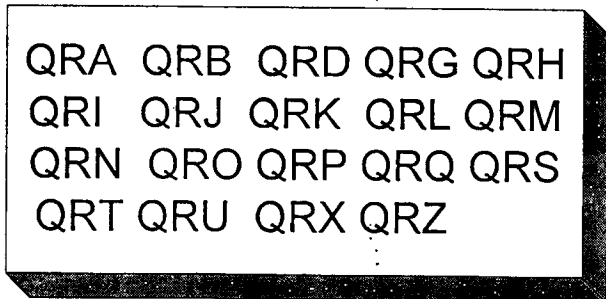
4. เมื่อผู้ใช้เลือกกด F1 จะเข้าสู่โหมด Help ดังรูปที่ 7 ซึ่งมี 2 เมนูย่อยให้ทำการเลือกใช้



<Help>  
1. Q - Code  
2. Introduction

รูปที่ 7 ข้อความ เมื่อกด F1 ที่โหมด Help

5. ถ้าผู้ใช้เลือกเข้าสู่โหมด Q - Code โดยการกดเลข “ 1 ” จะมีการบอกความหมายของรหัส Q ต่างๆ ให้เลือก ดังรูปที่ 8



QRA QRB QRD QRG QRH  
QRI QRJ QRK QRL QRM  
QRN QRO QRP QRQ QRS  
QRT QRU QRX QRZ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องแจ้งถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 8 ข้อความ เมื่อกดเลข “1” ที่โหมด Q - Code

6. จากข้อ 5 ถ้าผู้ใช้ต้องการทราบความหมายของรหัส Q ใดๆ สามารถเลื่อน โดยการกดปุ่ม ลูกศรเลื่อนไปทางด้านขวามือ เมื่อเลื่อนได้ตามที่ต้องการแล้วให้กด Enter จะได้รูปตัวอย่างดังรูปที่ 9 ถ้าต้องการกลับไปเลือกดูความหมายของรหัส Q ตัวอื่นๆ สามารถทำได้โดยการกดปุ่ม Esc

7. ถ้าเลื่อนลูกศรลงมาข้างล่างจะเป็นการเลือกรหัส Q ในหน้าที่ 2 ดังรูปที่ 10

8. เมื่อกด Esc เพื่อกลับเข้าสู่รายการเลือก Help ดังรูปที่ 11

9. ทำการกดเลข 2 เพื่อเข้าสู่โหมด Introduction ดังรูปที่ จ.12 ซึ่งสามารถเลือกอ่านได้ โดยการกดลูกศรเลื่อนลง หรือกด Esc เพื่อออกจากโหมด Introduction เข้าสู่เมนูหลักดังรูปที่ 12

10. ทำการเลือกกด F2 เพื่อเข้าสู่โหมด Command ซึ่งมี 2 เมนูย่อยให้เลือกดูเวลาหรือคู่มือ ความ Log Book ดังรูปที่ 13

11. ถ้าเลือกกดเลข 1 จะแสดงผลเวลาปัจจุบัน ซึ่งประกอบไปด้วย วัน - เดือน - ปี และแสดงผลเป็น ชั่วโมง - นาที - วินาที ดังรูปที่ 14

12. ถ้าทำการเลือกกดเลข 2 จะเข้าสู่การดูข้อมูลที่บันทึกการสนทนาโดยแบ่งการบันทึกได้ 10 ครั้ง สามารถเปิด, ลบ หรือถ่ายโอนข้อมูลไปยังคอมพิวเตอร์ได้ ดังรูปที่ 15

เครื่องหมาย \* ได้ ตัวเลข 0 - 9 แสดงว่ามีการสนทนาทั้งหมดเป็นจำนวนครั้งเท่ากับจำนวน \* ที่มีอยู่

What is the name of your station?

Esc : Return

รูปที่ 9 ข้อความ เมื่อกดปุ่มลูกศรเลื่อนหารหัส Q ที่ต้องการแล้วกด Enter

QSA QSB QSD QSL QSM  
QSO QSP QSV QSX QSZ  
QTH QTR

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งยังห้ามนำไปเผยแพร่และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 10 ข้อความ เมื่อกดเลื่อนลูกศรลง ทำการเลือกรหัส Q

<Help>  
 1. Q - Code  
 2. Introduction

รูปที่ 11 ข้อความ เมื่อกด Esc เพื่อกลับเข้าสู่รายการเลือก Help

Packet Radio Control  
 by Microcontroller  
 for Amature Radio  
 Esc : Return

รูปที่ 12 ข้อความ เมื่อกดเลข 2 เพื่อเข้าสู่โหมด Introduction

<Command>  
 1. Time  
 2. Log Book

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ โดยผู้จัดทำไว้เพื่อเผยแพร่ความรู้ให้แก่ประชาชนโดยไม่หวังผลตอบแทน  
 รูปที่ 13 ข้อความ เมื่อเลือกกด F2 เพื่อเข้าสู่โหมด Command  
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

*****Time*****
*           15-02-42           *
*           12-03:16           *
*****

```

รูปที่ 14 ข้อความ เมื่อกดเลข 1 เพื่อแสดงผลของเวลา

```

<Log Book>
0 1 2 3 4 5 6 7 8 9
* * * * *
O : Open D : Del L : Link

```

รูปที่ 15 ข้อความ เมื่อกดเลข 2 เพื่อดูข้อมูลที่บันทึกไว้ในการสนทนา

13. ถ้าต้องการเข้าไปดูข้อมูลที่บันทึกการสนทนาให้กดเลข 0 - 9 เลขใดเลขหนึ่งเครื่องหมาย \* ได้เลขที่เลือกนั้นจะกระพริบขึ้น ต้องทำการกดปุ่ม O ก็จะแสดงผลข้อมูลดังรูปที่ 16 โดยจะแสดงชื่อของกลุ่มสนทนา เวลาที่เริ่มต้นสนทนา และเวลาที่สิ้นสุดการสนทนา

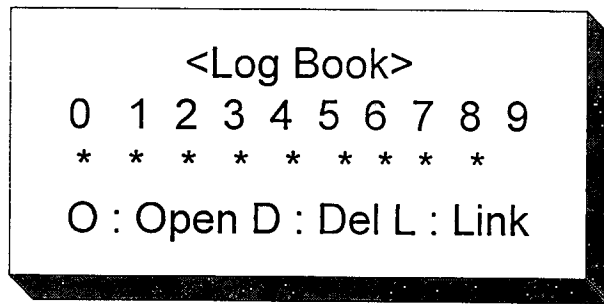
```

FROM : AS6BCD
TO : HS5XXX
START : 15-02-42 12 : 34
END : 15-02-42 12 : 42

```

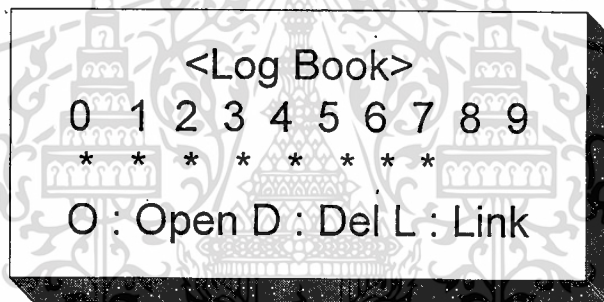
รูปที่ 16 ข้อความ เมื่อกดเลข 0 - 9

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
14. กด Esc เพื่อเข้าสู่รายการเลือก Log Book ตัวเดิม ถ้าต้องการลบเพิ่มหมายเลข 8 ให้กด  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งยังมีให้คำแนะนำและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้  
เลข 8 ที่เครื่องหมาย \* ได้เลข 8 จะกระพริบดังรูปที่ 17



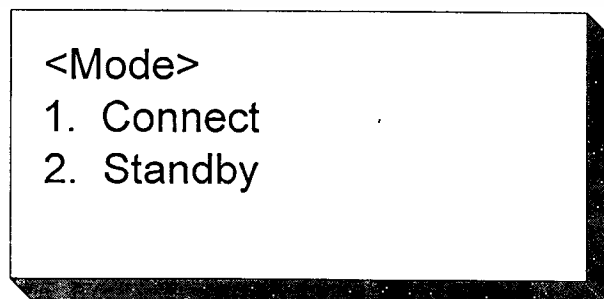
รูปที่ 17 ข้อความ เมื่อกด Esc เพื่อเข้าสู่รายการเลือก Log Book

15. กดปุ่ม D เพื่อลบข้อมูล เครื่องหมาย \* ได้เลข 8 จะหายไป แสดงว่าข้อมูลในแฟ้มที่ 8 พร้อมทั้งจะบันทึกข้อมูลที่จะเข้ามาใหม่ได้ ดังรูปที่ 18



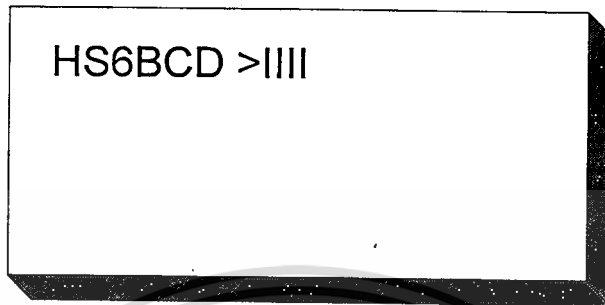
รูปที่ 18 ข้อความ เมื่อกดปุ่ม D เพื่อลบข้อมูล

16. จากรายการเลือกหลัก เลือกกด F3 เพื่อเข้าสู่รายการเลือก Mode โดยมี 2 รายการเลือกย่อยให้เลือก ดังรูปที่ 19



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
รูปที่ 19 ข้อความ เมื่อกด F3 เพื่อเข้าสู่รายการเลือก Mode  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

17. ถ้าเลือกกดเลข 1 เพื่อทำการติดต่อสื่อสารระหว่างกันจะมีชื่อสถานะของผู้ใช้ ซึ่งสามารถพิมพ์ข้อมูลที่ต้องการส่งได้ภายใน 2 บรรทัด ดังรูปที่ 20:



รูปที่ 20 ข้อความ เมื่อกดเลข 1 เพื่อทำการติดต่อสื่อสารกัน

18. เมื่อมีคู่สถานีส่งข้อมูลมาจะปรากฏข้อความที่ 2 บรรทัดล่าง โดยมีชื่อของคู่สถานีนำหน้ามาด้วยทุกครั้ง พร้อมทั้งข้อความตัวอย่างดังรูปที่ 21

จากรูปคู่สถานีชื่อ HS5XXX ว่าข้อความคำว่า “HELLO” มาให้แก่ HS6BCD



รูปที่ 21 ข้อความ เมื่อมีคู่สถานีส่งข้อมูลเข้ามา

19. จากรายการเลือก Mode ถ้าเลือกกดเลข 2 เข้าสู่รายการเลือกโหมด Standby เพื่อรองรับข้อมูลเพียงอย่างเดียว ไม่สามารถส่งข้อมูลออกไปได้ดังรูปที่ 22

20. ถ้ามีข้อมูลส่งเข้ามาจะแสดงผลที่ 2 บรรทัดด้านล่าง ตัวอย่างดังรูปที่ 23 จากรูปแสดงว่า HS5XXX ส่งข้อความคำว่า “HELLO” มา ถ้าผู้ใช้ต้องการติดต่อกลับไปให้กด Esc และเลือกเข้าสู่โหมด Connect จากรายการเลือกโหมด

21. จากเมนูหลักกด F4 เพื่อเข้าสู่รายการเลือก Setup และมี 3 รายการเล็กน้อยให้เลือก ดังรูปไม่ที่ 24

22. ถ้าต้องการเลือกตั้งค่าของ วัน- เดือน - ปี ให้กดเลข 1 จากเมนู Setup จะแสดงผลดังรูปที่ 25 จากรูปแสดง วัน - เดือน - ปี ในการเดิม วัน - เดือน - ปี ที่ต้องการควรตั้งค่าใหม่ ถ้าตั้งค่าของ วัน - เดือน - ปี ใหม่แล้ว ให้กด Enter หรือ กด Esc เมื่อไม่ต้องการตั้งค่าใหม่

23. ถ้าต้องการเลือกตั้งค่าของ ชั่วโมง - นาที - วินาที ให้กดเลข 2 จากรายการเลือก Setup จะแสดงผลดังรูปที่ 26 จากรูปจะแสดงเวลาเดิมและเวลาที่ต้องการตั้งค่าใหม่ มีกระบวนการทำเช่นเดียวกับข้อ 22

<Standby>

รูปที่ 22 ข้อความ เมื่อกดเลข 2 เพื่อเข้าโหมด Standby

<Standby>

HS5XXX > HELLO

รูปที่ 23 ข้อความ เมื่อมีข้อมูลส่งเข้ามา

<Setup>

1. Date
2. Timer
3. Callsign

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งยังมีให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 24 ข้อความ เมื่อกด F4 เพื่อเข้าสู่รายการเลือก Setup

## 6. คำสั่ง Q-Code ในรายการเลือก Help

คำสั่ง Q-Code ที่มีอยู่ในรายการเลือก Help มีดังต่อไปนี้

**QRA**

What is the name of your station ?

**QRB**

How far approximately are you from my station ?

**QRD**

Where are you bound and where are you from ?

**QRG**

Will you tell me my exact frequency in KHz ?

**QRH**

Does my frequency vary ?

**QRI**

Is my note good ?

**QRJ**

Does you receive me badly ?

Are my signals weak ?

**QRK**

Do you receive me well ?

Are my signals good ?

**QRL**

Are you busy ?

**QRM**

Are you being interfered with ?

**QRN**

Are you troubled by atmospheric ?

**QRO**

Shall I increase power ?

**QRP**

Shall I decrease power ?

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**QRQ**

Shall I send faster ?

**QRS**

Shall I send more slowly ?

**QRT**

Shall I stop sending ?

**QRU**

Have you anything for me ?

Are you ready ?

**QRX**

Shall I wait ?

When will you call me again ?

**QRZ**

Who is calling me ?

**QSA**

What is the strength of my signals ?

**QSB**

Does the strength of my signal vary ?

**QSD**

Is my keying correct ?

Signals distinct ?

**QSL**

Can you give me as knowledge of receipt ?

**QSM**

Shall I repeat the last telegram (message) I sent you ?

**QSO**

Can you communicate with .... Direct ?

**QSP**

Will you relay to ..... ?

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

QSZ

Shall I sand each word or group twice ?

QTH

What is your position in latitude and longitude ?

QTR

What is the exact time ?

## 7. รูปแสดงตำแหน่งของแป้นพิมพ์ ต่าง ๆ

ลักษณะรูปแบบของแป้นพิมพ์เกี่ยวกับการวางตำแหน่งตัวอักษรของแป้นพิมพ์ อักษรที่ใช้งานจะมีจำนวนดังรูปที่ 28 ซึ่งขนาดของการออกแบบนั้นออกแบบตามความเหมาะสมของงานและความสมดุลของกล่องที่ใช้งานด้วย

|     |       |   |   |   |    |       |               |
|-----|-------|---|---|---|----|-------|---------------|
| 0   | 1     | 2 | 3 | 4 | 5  | 6     | 7             |
| 8   | 9     | A | B | C | D  | E     | F             |
| G   | H     | I | J | K | L  | M     | N             |
| O   | P     | Q | R | S | T  | U     | V             |
| W   | X     | Y | Z | ( | )  | !     | Back<br>Space |
| F1  | F2    | : | , | “ | ?  | .     | &             |
| F3  | F4    | ; |   | % | \$ | *     | =             |
| Esc | Space | — |   | — | #  | Enter |               |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับรูปที่ 28 การวางตำแหน่งตัวอักษรของแป้นพิมพ์ นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# DALLAS

SEMICONDUCTOR

## DS1202, DS1202S

### Serial Timekeeping Chip

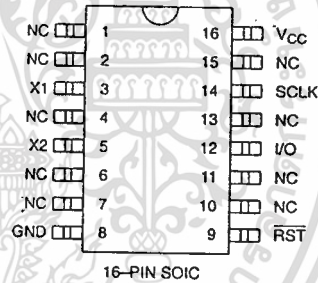
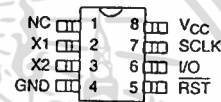
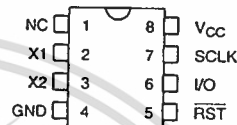
#### FEATURES

- Real time clock counts seconds, minutes, hours, date of the month, month, day of the week, and year with leap year compensation
- 24 x 8 RAM for scratchpad data storage
- Serial I/O for minimum pin count
- 2.0-5.5 volt full operation
- Uses less than 300 nA at 2 volts
- Single-byte or multiple-byte (burst mode) data transfer for read or write of clock or RAM data
- 8-pin DIP or optional 16-pin SOIC for surface mount
- Simple 3-wire interface
- TTL-compatible ( $V_{CC} = 5V$ )
- Optional industrial temperature range  $-40^{\circ}C$  to  $+85^{\circ}C$

#### ORDERING INFORMATION

DS1202      8-pin DIP  
 DS1202S    16-pin SOIC  
 DS1202S8   8-pin SOIC

#### PIN ASSIGNMENT



#### PIN DESCRIPTION

NC      - No Connection  
 X1, X2    - 32.768 KHz Crystal Input  
 GND      - Ground  
 $\overline{RST}$      - Reset  
 I/O       - Data Input/Output  
 SCLK     - Serial Clock  
 $V_{CC}$      - Power Supply Pin

#### DESCRIPTION

The DS1202 Serial Timekeeping Chip contains a real time clock/calendar and 24 bytes of static RAM. It communicates with a microprocessor via a simple serial interface. The real time clock/calendar provides seconds, minutes, hours, day, date, month, and year information. The end of the month date is automatically adjusted for months with less than 31 days, including corrections for

leap year. The clock operates in either the 24-hour or 12-hour format with an AM/PM indicator. Interfacing the DS1202 with a microprocessor is simplified by using synchronous serial communication. Only three wires are required to communicate with the clock/RAM: (1)  $\overline{RST}$  (Reset), (2) I/O (Data line), and (3) SCLK (Serial clock). Data can be transferred to and from the clock/

RAM one byte at a time or in a burst of up to 24 bytes. The DS1202 is designed to operate on very low power and retain data and clock information on less than 1 micro-watt.

### OPERATION

The main elements of the Serial Timekeeper are shown in Figure 1: shift register, control logic, oscillator, real time clock, and RAM. To initiate any transfer of data,  $\overline{RST}$  is taken high and eight bits are loaded into the shift register providing both address and command information. Data is serially input on the rising edge of the SCLK. The first eight bits specify which of 32 bytes will be accessed, whether a read or write cycle will take place, and whether a byte or burst mode transfer is to occur. After the first eight clock cycles have occurred which

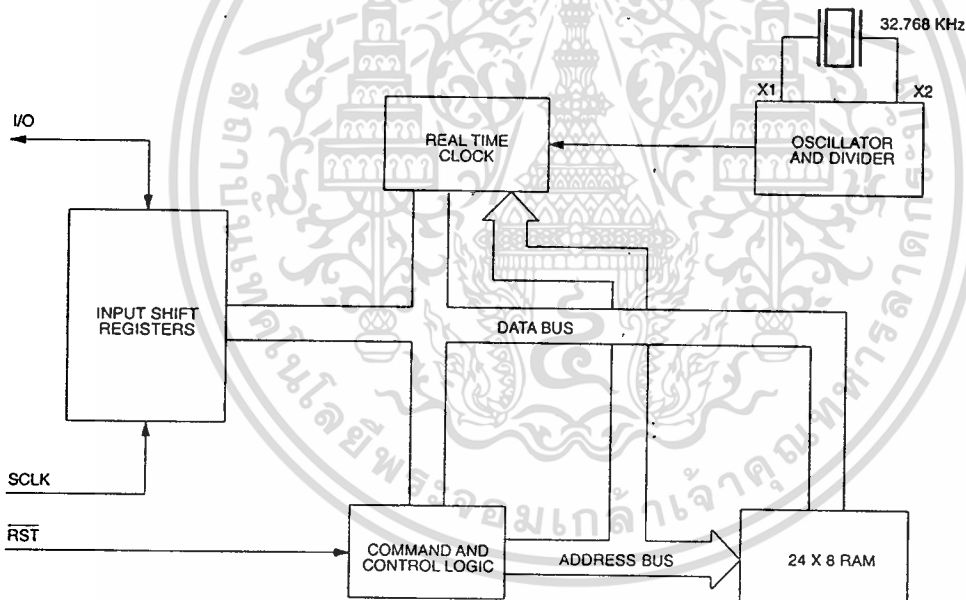
load the command word into the shift register, additional clocks will output data for a read or input data for a write. The number of clock pulses equals eight plus eight for byte mode or eight plus up to 192 for burst mode.

### COMMAND BYTE

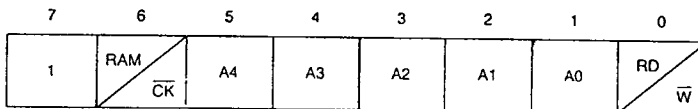
The command byte is shown in Figure 2. Each data transfer is initiated by a command byte. The MSB (Bit 7)

rated. Bit 6 specifies clock/calendar data if logic 0 or RAM data if logic 1. Bits one through five specify the designated registers to be input or output, and the LSB (Bit 0) specifies a write operation (input) if logic 0 or read operation (output) if logic 1. The command byte is always input starting with the LSB (bit 0).

DS1202 BLOCK DIAGRAM Figure 1



ADDRESS/COMMAND BYTE Figure 2



## RESET AND CLOCK CONTROL

All data transfers are initiated by driving the  $\overline{\text{RST}}$  input high. The  $\overline{\text{RST}}$  input serves two functions. First,  $\overline{\text{RST}}$  turns on the control logic which allows access to the shift register for the address/command sequence. Second, the  $\overline{\text{RST}}$  signal provides a method of terminating either single byte or multiple byte data transfer. A clock cycle is a sequence of a falling edge followed by a rising edge. For data inputs, data must be valid during the rising edge of the clock and data bits are output on the falling edge of clock. All data transfer terminates if the  $\overline{\text{RST}}$  input is low and the I/O pin goes to a high impedance state. Data transfer is illustrated in Figure 3.

## DATA INPUT

Following the eight SCLK cycles that input a write command byte, a data byte is input on the rising edge of the next eight SCLK cycles. Additional SCLK cycles are ignored should they inadvertently occur. Data is input starting with bit 0.

## DATA OUTPUT

Following the eight SCLK cycles that input a read command byte, a data byte is output on the falling edge of the next eight SCLK cycles. Note that the first data bit to be transmitted occurs on the first falling edge after the last bit of the command byte is written. Additional SCLK cycles retransmit the data bytes should they inadvertently occur so long as  $\overline{\text{RST}}$  remains high. This operation permits continuous burst mode read capability. Data is output starting with bit 0.

## BURST MODE

Burst mode may be specified for either the clock/calendar or the RAM registers by addressing location 31 decimal (address/command bits one through five = logical one). As before, bit six specified clock or RAM and bit 0 specifies read or write. There is no data storage capacity at locations 8 through 31 in the Clock/Calendar Registers or locations 24 through 31 in the RAM registers. When writing to the clock registers in the burst mode, the first eight registers must be written in order for the data to be transferred.

However, when writing to RAM in burst mode it is not necessary to write all 24 bytes for the data to transfer. Each byte that is written to will be transferred to RAM regardless of whether all 24 bytes are written or not.

## CLOCK/CALENDAR

The clock/calendar is contained in eight write/read registers as shown in Figure 4. Data contained in the clock/calendar registers is in binary coded decimal format (BCD).

## CLOCK HALT FLAG

Bit 7 of the seconds register is defined as the clock halt flag. When this bit is set to logic 1, the clock oscillator is stopped, and the DS1202 is placed into a low-power standby mode with a current drain of not more than 100 nanoamps. When this bit is written to logic 0, the clock will start.

## AM-PM/12-24 MODE

Bit 7 of the hours register is defined as the 12- or 24-hour mode select bit. When high, the 12-hour mode is selected. In the 12-hour mode, bit 5 is the AM/PM bit with logic high being PM. In the 24-hour mode, bit 5 is the second 10 hour bit (20-23 hours).

## WRITE PROTECT REGISTER

Bit 7 of write protect register is the write protect bit. The first seven bits (bits 0-6) are forced to zero and will always read a zero when read. Before any write operation to the clock or RAM, bit 7 must be zero. When high, the write protect bit prevents a write operation to any other register.

## CLOCK/CALENDAR BURST MODE

The clock/calendar command byte specifies burst mode operation. In this mode the eight clock/calendar registers can be consecutively read or written (see Figure 4) starting with bit 0 of address 0.

## RAM

The static RAM is 24 x 8 bytes addressed consecutively in the RAM address space.

## RAM BURST MODE

The RAM command byte specifies burst mode operation. In this mode, the 24 RAM registers can be consecutively read or written (see Figure 4) starting with bit 0 of address 0.

REGISTER SUMMARY

A register data format summary is shown in Figure 4.

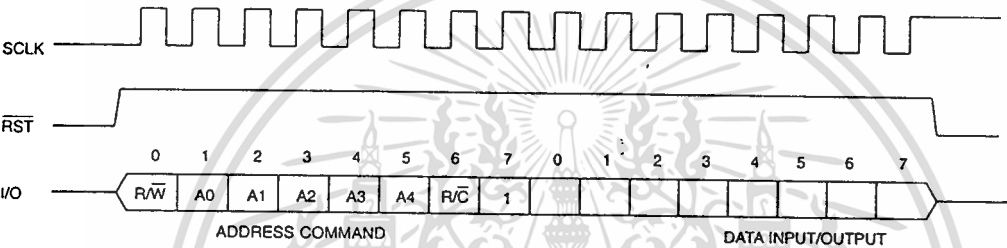
CRYSTAL SELECTION

A 32.768 KHz crystal, Daiwa Part No. DT26S, Seiko Part No. DS-VT-200 or equivalent, can be directly connected to the DS1202 via pins 2 and 3 (X1, X2). The crystal selected for use should have a specified load ca-

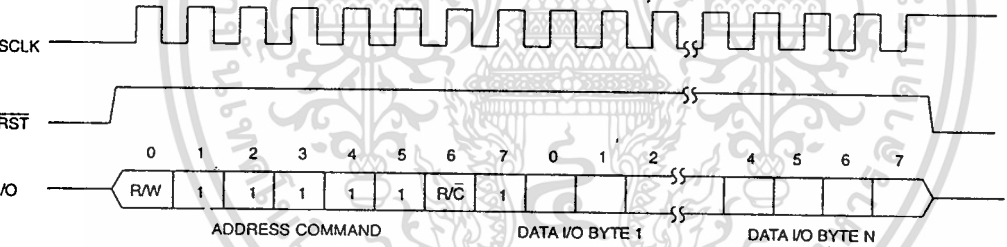
pacitance (CL) of 6 pF. The crystal is connected directly to the X1 and X2 pins. There is no need for external capacitors or resistors. Note: X1 and X2 are very high impedance nodes. It is recommended that they and the crystal be guard-ringed with ground and that high frequency signals be kept away from the crystal area. For more information on crystal selection and crystal layout considerations, please consult Application Note 58, "Crystal Considerations with Dallas Real Time Clocks".

DATA TRANSFER SUMMARY Figure 3

SINGLE BYTE TRANSFER



BURST MODE TRANSFER



| FUNCTION | BYTE N | SCLK n |
|----------|--------|--------|
| CLOCK    | 8      | 72     |
| RAM      | 24     | 200    |

REGISTER ADDRESS/DEFINITION Figure 4

# REGISTER ADDRESS

## A. CLOCK

# REGISTER DEFINITION

|     |   |   |   |   |   |   |   |         |
|-----|---|---|---|---|---|---|---|---------|
|     | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0       |
| SEC | 1 | 0 | 0 | 0 | 0 | 0 | 0 | RD<br>W |

|       |    |        |     |
|-------|----|--------|-----|
| 00-59 | CH | 10 SEC | SEC |
|-------|----|--------|-----|

|     |   |   |   |   |   |   |   |         |
|-----|---|---|---|---|---|---|---|---------|
| MIN | 1 | 0 | 0 | 0 | 0 | 0 | 1 | RD<br>W |
|-----|---|---|---|---|---|---|---|---------|

|       |   |        |     |
|-------|---|--------|-----|
| 00-59 | 0 | 10 MIN | MIN |
|-------|---|--------|-----|

|    |   |   |   |   |   |   |   |         |
|----|---|---|---|---|---|---|---|---------|
| HR | 1 | 0 | 0 | 0 | 0 | 1 | 0 | RD<br>W |
|----|---|---|---|---|---|---|---|---------|

|                |           |   |           |    |    |
|----------------|-----------|---|-----------|----|----|
| 01-12<br>00-23 | 12/<br>24 | 0 | 10<br>A/P | HR | HR |
|----------------|-----------|---|-----------|----|----|

|      |   |   |   |   |   |   |   |         |
|------|---|---|---|---|---|---|---|---------|
| DATE | 1 | 0 | 0 | 0 | 0 | 1 | 1 | RD<br>W |
|------|---|---|---|---|---|---|---|---------|

|                            |   |   |         |      |
|----------------------------|---|---|---------|------|
| 01-28/29<br>01-30<br>01-31 | 0 | 0 | 10 DATE | DATE |
|----------------------------|---|---|---------|------|

|       |   |   |   |   |   |   |   |         |
|-------|---|---|---|---|---|---|---|---------|
| MONTH | 1 | 0 | 0 | 0 | 1 | 0 | 0 | RD<br>W |
|-------|---|---|---|---|---|---|---|---------|

|       |   |   |   |         |       |
|-------|---|---|---|---------|-------|
| 01-12 | 0 | 0 | 0 | 10<br>M | MONTH |
|-------|---|---|---|---------|-------|

|     |   |   |   |   |   |   |   |         |
|-----|---|---|---|---|---|---|---|---------|
| DAY | 1 | 0 | 0 | 0 | 1 | 0 | 1 | RD<br>W |
|-----|---|---|---|---|---|---|---|---------|

|       |   |   |   |   |   |     |
|-------|---|---|---|---|---|-----|
| 01-07 | 0 | 0 | 0 | 0 | 0 | DAY |
|-------|---|---|---|---|---|-----|

|      |   |   |   |   |   |   |   |         |
|------|---|---|---|---|---|---|---|---------|
| YEAR | 1 | 0 | 0 | 0 | 1 | 1 | 0 | RD<br>W |
|------|---|---|---|---|---|---|---|---------|

|      |         |      |
|------|---------|------|
| 0-99 | 10 YEAR | YEAR |
|------|---------|------|

|         |   |   |   |   |   |   |   |         |
|---------|---|---|---|---|---|---|---|---------|
| CONTROL | 1 | 0 | 0 | 0 | 1 | 1 | 1 | RD<br>W |
|---------|---|---|---|---|---|---|---|---------|

|    |                |
|----|----------------|
| WP | FORCED TO ZERO |
|----|----------------|

|                |   |   |   |   |   |   |   |         |
|----------------|---|---|---|---|---|---|---|---------|
| CLOCK<br>BURST | 1 | 0 | 1 | 1 | 1 | 1 | 1 | RD<br>W |
|----------------|---|---|---|---|---|---|---|---------|

## B. RAM

|       |   |   |   |   |   |   |   |         |
|-------|---|---|---|---|---|---|---|---------|
| RAM 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | RD<br>W |
|-------|---|---|---|---|---|---|---|---------|

|            |  |  |  |  |  |  |  |
|------------|--|--|--|--|--|--|--|
| RAM DATA 0 |  |  |  |  |  |  |  |
|------------|--|--|--|--|--|--|--|

⋮

⋮

|        |   |   |   |   |   |   |   |         |
|--------|---|---|---|---|---|---|---|---------|
| RAM 23 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | RD<br>W |
|--------|---|---|---|---|---|---|---|---------|

|             |  |  |  |  |  |  |  |
|-------------|--|--|--|--|--|--|--|
| RAM DATA 23 |  |  |  |  |  |  |  |
|-------------|--|--|--|--|--|--|--|

|              |   |   |   |   |   |   |   |         |
|--------------|---|---|---|---|---|---|---|---------|
| RAM<br>BURST | 1 | 1 | 1 | 1 | 1 | 1 | 1 | RD<br>W |
|--------------|---|---|---|---|---|---|---|---------|

DS1202, DS1202S

ABSOLUTE MAXIMUM RATINGS\*

|                                       |                      |
|---------------------------------------|----------------------|
| Voltage on Any Pin Relative to Ground | −0.3V to +7.0V       |
| Operating Temperature                 | 0°C to 70°C          |
| Storage Temperature                   | −55°C to +125°C      |
| Soldering Temperature                 | 260°C for 10 seconds |

\* This is a stress rating only and functional operation of the device at these or any other conditions above those indicated in the operation sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods of time may affect reliability.

RECOMMENDED DC OPERATING CONDITIONS (0°C to 70°C)

| PARAMETER      | SYMBOL          | MIN                   | TYP  | MAX                  | UNITS | NOTES |
|----------------|-----------------|-----------------------|------|----------------------|-------|-------|
| Supply Voltage | V <sub>CC</sub> | 2.0                   |      | 5.5                  | V     | 1     |
| Logic 1 Input  | V <sub>IH</sub> | 2.0                   |      | V <sub>CC</sub> +0.3 | V     | 1     |
| Logic 0 Input  | V <sub>IL</sub> | V <sub>CC</sub> =2.0V | −0.3 | +0.3                 | V     | 1     |
|                |                 | V <sub>CC</sub> =5V   | −0.3 | +0.8                 |       |       |

DC ELECTRICAL CHARACTERISTICS (0°C to 70°C; V<sub>CC</sub> = 2.0 to 5.5V\*)

| PARAMETER             | SYMBOL           | MIN                 | TYP | MAX  | UNITS | NOTES |
|-----------------------|------------------|---------------------|-----|------|-------|-------|
| Input Leakage         | I <sub>LI</sub>  |                     |     | +500 | μA    | 6     |
| I/O Leakage           | I <sub>LO</sub>  |                     |     | +500 | μA    | 6     |
| Logic 1 Output        | V <sub>OH</sub>  | V <sub>CC</sub> =2V | 1.6 |      | V     | 2     |
|                       |                  | V <sub>CC</sub> =5V | 2.4 |      |       |       |
| Logic 0 Output        | V <sub>OL</sub>  | V <sub>CC</sub> =2V |     | 0.4  | V     | 3     |
|                       |                  | V <sub>CC</sub> =5V |     | 0.4  |       |       |
| Active Supply Current | I <sub>CC</sub>  | V <sub>CC</sub> =2V |     | .4   | mA    | 5     |
|                       |                  | V <sub>CC</sub> =5V |     | 1.2  |       |       |
| Timekeeping Current   | I <sub>CC1</sub> | V <sub>CC</sub> =2V |     | 0.3  | μA    | 4     |
|                       |                  | V <sub>CC</sub> =5V |     | 1    |       |       |
| Leakage Current       | I <sub>CC2</sub> | V <sub>CC</sub> =2V |     | 100  | nA    | 10    |
|                       |                  | V <sub>CC</sub> =5V |     | 100  |       |       |

\*Unless otherwise noted.

CAPACITANCE (t<sub>A</sub> = 25°C)

| PARAMETER           | SYMBOL           | CONDITION | TYP | MAX | UNITS | NOTES |
|---------------------|------------------|-----------|-----|-----|-------|-------|
| Input Capacitance   | C <sub>I</sub>   |           | 5   |     | pF    |       |
| I/O Capacitance     | C <sub>I/O</sub> |           | 10  |     | pF    |       |
| Crystal Capacitance | C <sub>X</sub>   |           | 6   |     | pF    |       |

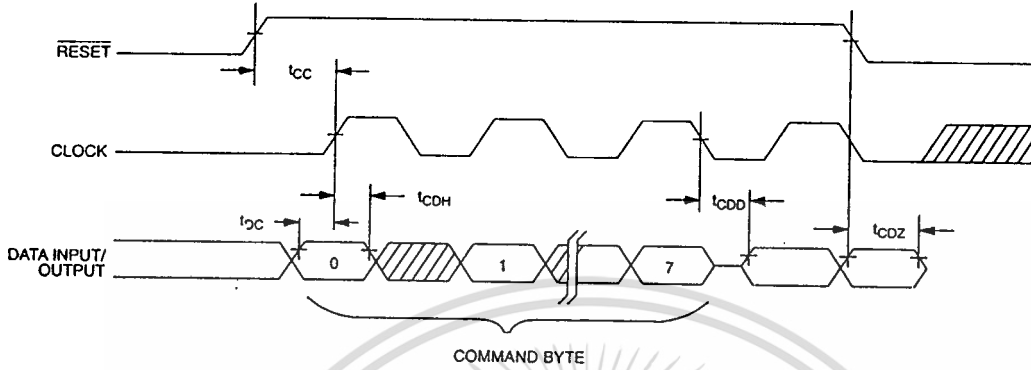
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

AC ELECTRICAL CHARACTERISTICS (0°C to 70°C; V<sub>CC</sub> = 2.0 to 5.5V\*)

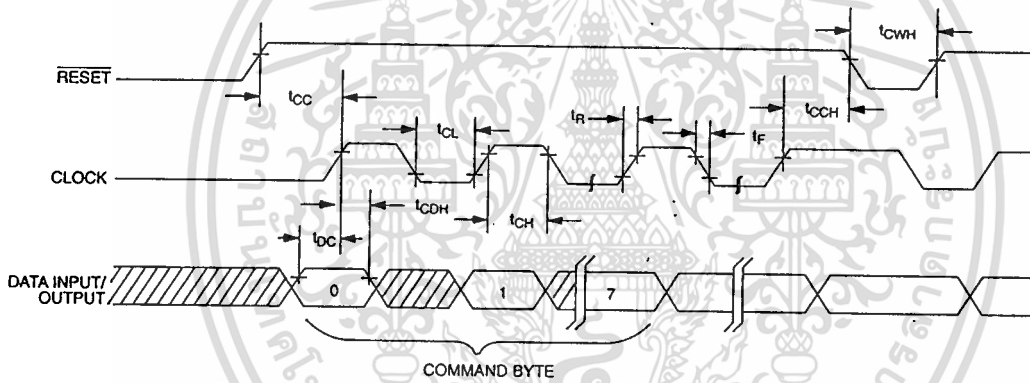
| PARAMETER         | SYMBOL                          | MIN                 | TYP  | MAX  | UNITS | NOTES   |
|-------------------|---------------------------------|---------------------|------|------|-------|---------|
| Data to CLK Setup | t <sub>DC</sub>                 | V <sub>CC</sub> =2V | 200  |      | ns    | 7       |
|                   |                                 | V <sub>CC</sub> =5V | 50   |      |       |         |
| CLK to Data Hold  | t <sub>CDH</sub>                | V <sub>CC</sub> =2V | 280  |      | ns    | 7       |
|                   |                                 | V <sub>CC</sub> =5V | 70   |      |       |         |
| CLK to Data Delay | t <sub>CDD</sub>                | V <sub>CC</sub> =2V |      | 800  | ns    | 7, 8, 9 |
|                   |                                 | V <sub>CC</sub> =5V |      | 200  |       |         |
| CLK Low Time      | t <sub>CL</sub>                 | V <sub>CC</sub> =2V | 1000 |      | ns    | 7       |
|                   |                                 | V <sub>CC</sub> =5V | 250  |      |       |         |
| CLK High Time     | t <sub>CH</sub>                 | V <sub>CC</sub> =2V | 1000 |      | ns    | 7, 12   |
|                   |                                 | V <sub>CC</sub> =5V | 250  |      |       |         |
| CLK Frequency     | f <sub>CLK</sub>                | V <sub>CC</sub> =2V |      | 0.5  | MHz   | 7, 12   |
|                   |                                 | V <sub>CC</sub> =5V | DC   | 2.0  |       |         |
| CLK Rise and Fall | t <sub>R</sub> , t <sub>F</sub> | V <sub>CC</sub> =2V |      | 2000 | ns    |         |
|                   |                                 | V <sub>CC</sub> =5V |      | 500  |       |         |
| RST to CLK Setup  | t <sub>CC</sub>                 | V <sub>CC</sub> =2V | 4    |      | μs    | 7       |
|                   |                                 | V <sub>CC</sub> =5V | 1    |      |       |         |
| CLK to RST Hold   | t <sub>CCH</sub>                | V <sub>CC</sub> =2V | 1000 |      | ns    | 7       |
|                   |                                 | V <sub>CC</sub> =5V | 250  |      |       |         |
| RST Inactive Time | t <sub>CWH</sub>                | V <sub>CC</sub> =2V | 4    |      | μs    | 7       |
|                   |                                 | V <sub>CC</sub> =5V | 1    |      |       |         |
| RST to I/O High Z | t <sub>CDZ</sub>                | V <sub>CC</sub> =2V |      | 280  | ns    | 7       |
|                   |                                 | V <sub>CC</sub> =5V |      | 70   |       |         |

\*Unless otherwise noted.

## TIMING DIAGRAM: READ DATA TRANSFER Figure 5



## TIMING DIAGRAM: WRITE DATA TRANSFER Figure 6

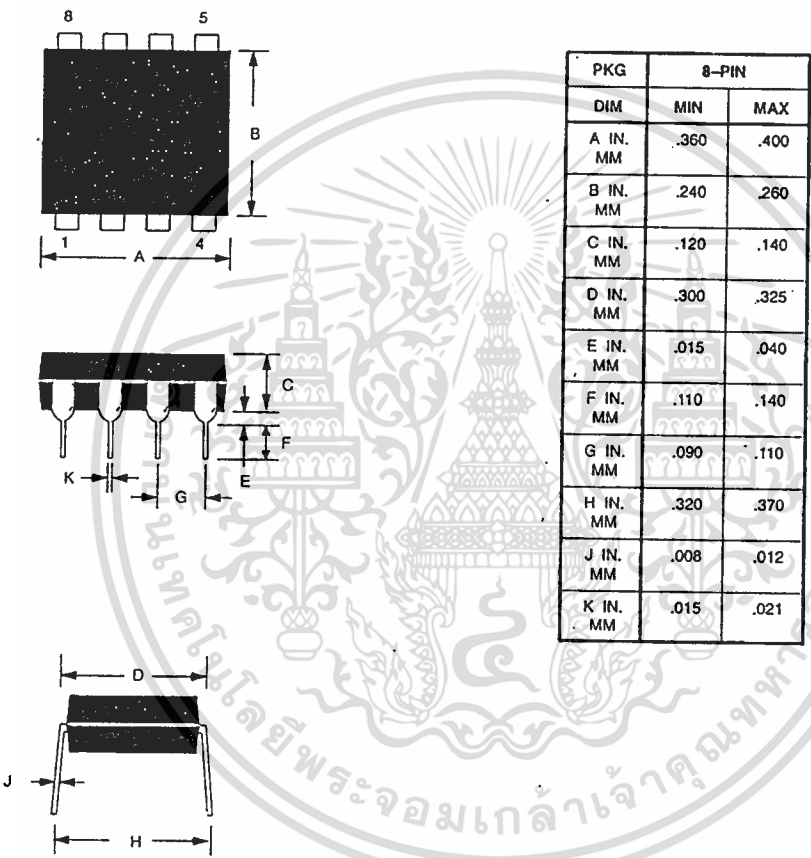


## NOTES:

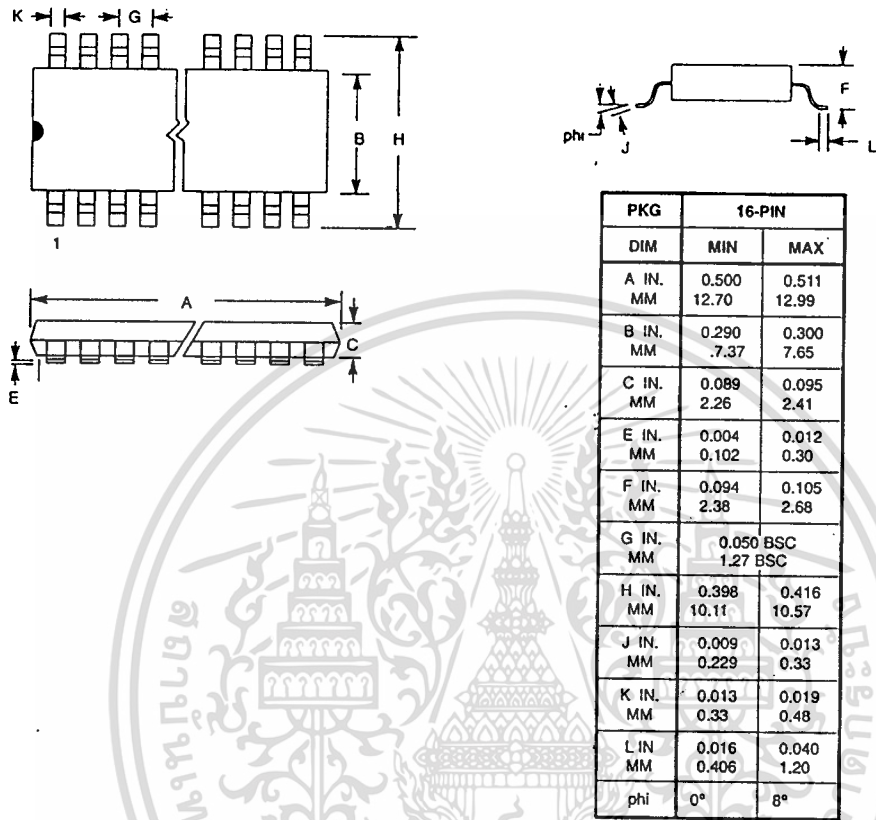
1. All voltages are referenced to ground.
2. Logic one voltages are specified at a source current of 1 mA at  $V_{CC}=5V$  and .4 mA at  $V_{CC}=2V$ ,  $V_{OH}=V_{CC}$  for capacitive loads.
3. Logic zero voltages are specified at a sink current of 4 mA at  $V_{CC}=5V$  and 1.5 mA at  $V_{CC}=2V$ .
4.  $I_{CC1}$  is specified with I/O open,  $\overline{RST}$  set to a logic 0, and clock halt flag=0 (oscillator enabled).
5.  $I_{CC}$  is specified with the I/O pin open,  $\overline{RST}$  high,  $SCLK=2\text{ MHz}$  at  $V_{CC}=5V$ ;  $SCLK=500\text{ KHz}$ ,  $V_{CC}=2V$  and clock halt flag=0 (oscillator enabled).
6.  $\overline{RST}$ ,  $SCLK$ , and I/O all have 40K $\Omega$  pulldown resistors to ground.
7. Measured at  $V_{IH}=2.0V$  or  $V_{IL}=0.8V$  and 10 ms maximum rise and fall time.
8. Measured at  $V_{OH}=2.4V$  or  $V_{OL}=0.4V$ .
9. Load capacitance = 50 pF.

- 10.  $I_{CC2}$  is specified with  $\overline{RST}$ , I/O, and SCLK open. The clock halt flag must be set to logic one (oscillator disabled).
- 11. At power-up,  $\overline{RST}$  must be at a logic 0 until  $V_{CC} \geq 2$  volts. Also, SCLK must be at a logic 0 when  $\overline{RST}$  is driven to a logic one state.
- 12. If  $t_{CH}$  exceeds 100 ms with  $\overline{RST}$  in a logic one state, then  $I_{CC}$  may briefly exceed  $I_{CC}$  specification.

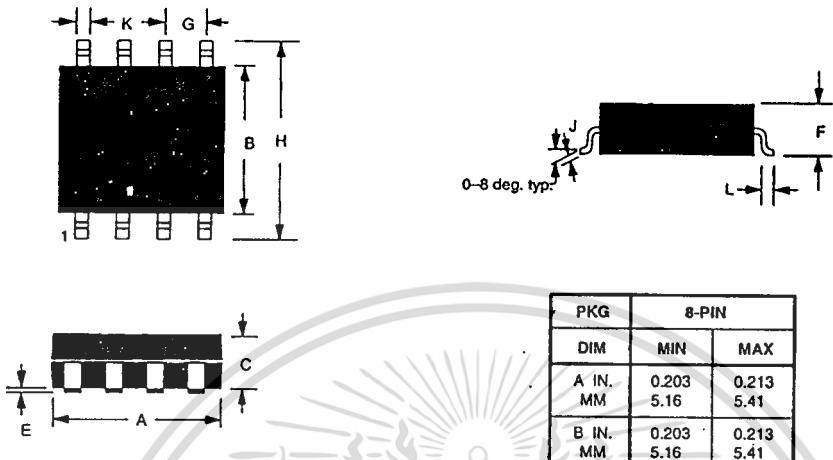
DS1202 SERIAL TIMEKEEPER 8-PIN DIP



DS1202S SERIAL TIMEKEEPER 16-PIN SOIC



DS1202S8 8-PIN SOIC 200 MIL



| PKG   | 8-PIN     |       |
|-------|-----------|-------|
|       | DIM       |       |
|       | MIN       | MAX   |
| A IN. | 0.203     | 0.213 |
| MM    | 5.16      | 5.41  |
| B IN. | 0.203     | 0.213 |
| MM    | 5.16      | 5.41  |
| C IN. | 0.070     | 0.074 |
| MM    | 1.78      | 1.88  |
| E IN. | 0.004     | 0.010 |
| MM    | 0.102     | 0.390 |
| F IN. | 0.074     | 0.84  |
| MM    | 1.88      | 2.13  |
| G IN. | 0.050 BSC |       |
| MM    | 1.27 BSC  |       |
| H IN. | 0.302     | 0.318 |
| MM    | 7.67      | 8.07  |
| J IN. | 0.006     | 0.010 |
| MM    | 0.152     | 0.254 |
| K IN. | 0.013     | 0.020 |
| MM    | 0.33      | 0.508 |
| L IN. | 0.19      | 0.030 |
| MM    | 4.83      | 0.762 |

19 0175, Rev. 3, 5/96



**±15kV ESD-Protected, +5V RS-232 Transceivers**

**General Description**

The MAX202E-MAX213E, MAX232E/MAX241E line drivers/receivers are designed for RS-232 and V.28 communications in harsh environments. Each transmitter output and receiver input is protected against ±15kV electrostatic discharge (ESD) shocks, without latchup. The various combinations of features are outlined in the *Selection Guide*. The drivers and receivers for all ten devices meet all EIA/TIA-232E and CCITT V.28 specifications at data rates up to 120kbps, when loaded in accordance with the EIA/TIA-232E specification.

The MAX211E/MAX213E/MAX241E are available in 28-pin SO packages, as well as a 28-pin SSOP that uses 60% less board space. The MAX202E/MAX232E come in 16-pin narrow SO, wide SO, and DIP packages. The MAX203E comes in a 20-pin DIP/SO package, and needs no external charge-pump capacitors. The MAX205E comes in a 24-pin wide DIP package, and also eliminates external charge-pump capacitors. The MAX206E/MAX207E/MAX208E come in 24-pin SO, SSOP, and narrow DIP packages. The MAX232E/MAX241E operate with four 1µF capacitors, while the MAX202E/MAX206E/MAX207E/MAX208E/MAX211E/MAX213E operate with four 0.1µF capacitors, further reducing cost and board space.

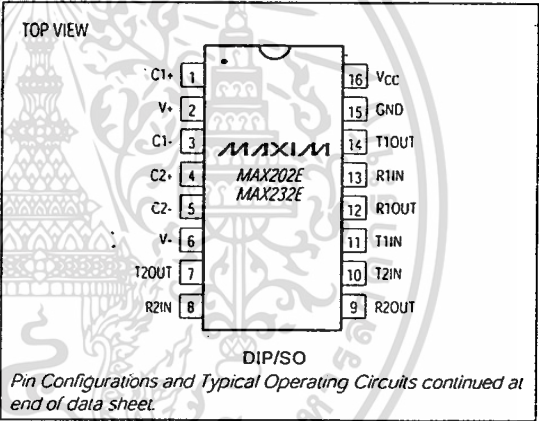
**Applications**

Notebook, Subnotebook, and Palmtop Computers  
Battery-Powered Equipment  
Hand-Held Equipment

**Features**

- ♦ ESD Protection for RS-232 I/O Pins:  
±15kV—Human Body Model  
±8kV—IEC1000-4-2, Contact Discharge  
±15kV—IEC1000-4-2, Air-Gap Discharge
- ♦ Latchup Free (unlike bipolar equivalents)
- ♦ Guaranteed 120kbps Data Rate—LapLink™ Compatible
- ♦ Guaranteed 3V/µs Min Slew Rate
- ♦ Operate from a Single +5V Power Supply

**Pin Configurations**



Ordering Information appears at end of data sheet.

**Selection Guide**

| PART    | No. of RS-232 DRIVERS | No. of RS-232 RECEIVERS | RECEIVERS ACTIVE IN SHUTDOWN | No. of EXTERNAL CAPACITORS | LOW-POWER SHUTDOWN | TTL THREE-STATE |
|---------|-----------------------|-------------------------|------------------------------|----------------------------|--------------------|-----------------|
| MAX202E | 2                     | 2                       | 0                            | 4 (0.1µF)                  | No                 | No              |
| MAX203E | 2                     | 2                       | 0                            | None                       | No                 | No              |
| MAX205E | 5                     | 5                       | 0                            | None                       | Yes                | Yes             |
| MAX206E | 4                     | 3                       | 0                            | 4 (0.1µF)                  | Yes                | Yes             |
| MAX207E | 5                     | 3                       | 0                            | 4 (0.1µF)                  | No                 | No              |
| MAX208E | 4                     | 4                       | 0                            | 4 (0.1µF)                  | No                 | No              |
| MAX211E | 4                     | 5                       | 0                            | 4 (0.1µF)                  | Yes                | Yes             |
| MAX213E | 4                     | 5                       | 2                            | 4 (0.1µF)                  | Yes                | Yes             |
| MAX232E | 2                     | 2                       | 0                            | 4 (1µF)                    | No                 | No              |
| MAX241E | 4                     | 5                       | 0                            | 4 (1µF)                    | Yes                | Yes             |

LapLink is a registered trademark of Traveling Software, Inc.

**MAXIM** Maxim Integrated Products

For free samples & the latest literature: <http://www.maxim-ic.com>, or phone 1-800-998-8800

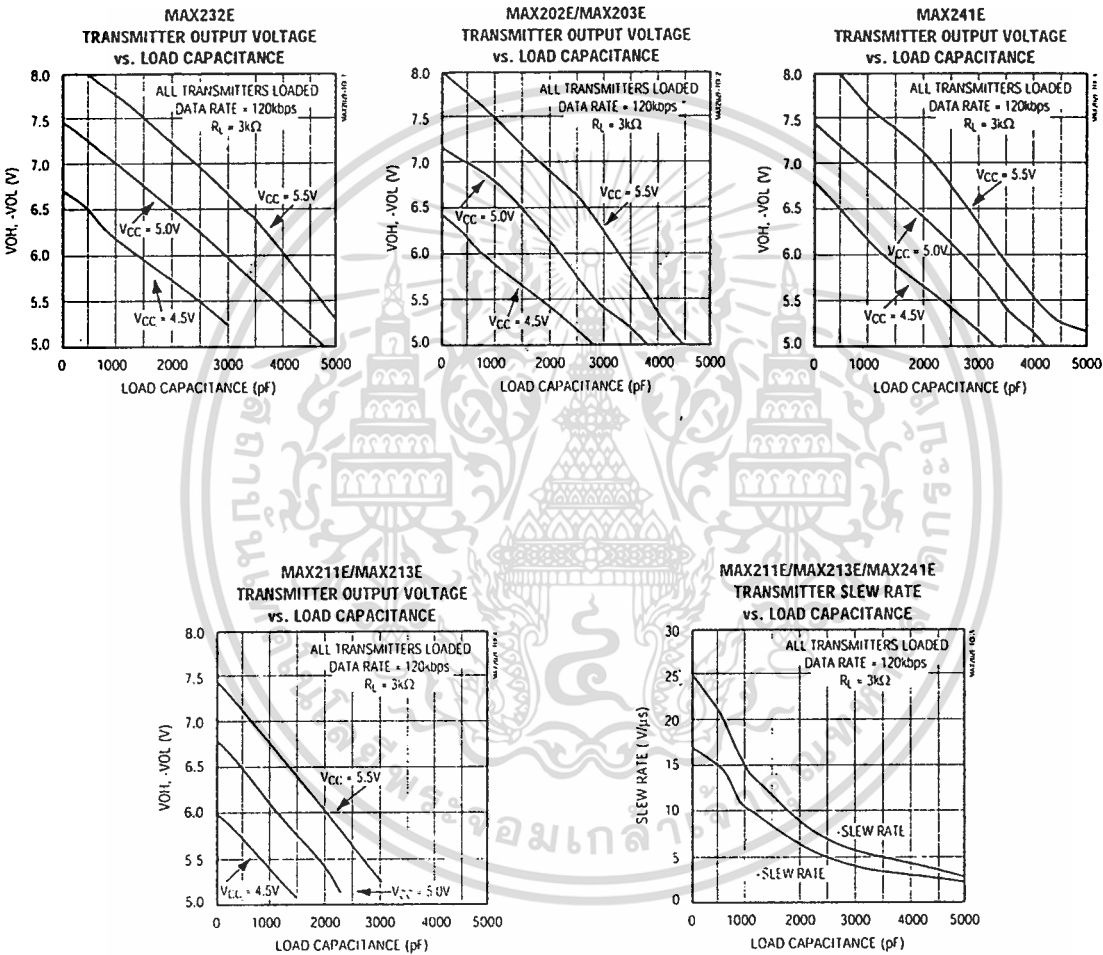
MAX202E-MAX213E, MAX232E/MAX241E

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

MAX202E-MAX213E, MAX232E/MAX241E

**±15kV ESD-Protected, +5V RS-232 Transceivers**

**Typical Operating Characteristics**  
(Typical Operating Circuits. V<sub>CC</sub> = +5V, T<sub>A</sub> = +25°C, unless otherwise noted.)



MAXIM

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### ***Pin Descriptions (continued)***

| PIN              | NAME                     | FUNCTION                                                                                                                                                                      |
|------------------|--------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1, 2, 3, 28      | T_OUT                    | RS-232 Driver Outputs                                                                                                                                                         |
| 4, 9, 18, 23, 27 | R_IN                     | RS-232 Receiver Inputs                                                                                                                                                        |
| 5, 8, 19, 22, 26 | R_OUT                    | TTL/CMOS Receiver Outputs. For the MAX213E, receivers R4 and R5 are active in shutdown mode when EN = 1. For the MAX211E and MAX241E, all receivers are inactive in shutdown. |
| 6, 7, 20, 21     | T_IN                     | TTL/CMOS Driver Inputs. Only the MAX211E, MAX213E, and MAX241E have internal pull-ups to V <sub>CC</sub>                                                                      |
| 10               | GND                      | Ground                                                                                                                                                                        |
| 11               | V <sub>CC</sub>          | +4.5V to +5.5V Supply Voltage                                                                                                                                                 |
| 12, 14           | C1+, C1-                 | Terminals for positive charge-pump capacitor                                                                                                                                  |
| 13               | V+                       | +2V <sub>CC</sub> voltage generated by the charge pump                                                                                                                        |
| 15, 16           | C2+, C2-                 | Terminals for negative charge-pump capacitor                                                                                                                                  |
| 17               | V-                       | -2V <sub>CC</sub> voltage generated by the charge pump                                                                                                                        |
| 24               | $\overline{\text{EN}}$   | Receiver Enable—active low (MAX211E, MAX241E)                                                                                                                                 |
|                  | EN                       | Receiver Enable—active high (MAX213E)                                                                                                                                         |
| 25               | SHDN                     | Shutdown Control—active high (MAX211E, MAX241E)                                                                                                                               |
|                  | $\overline{\text{SHDN}}$ | Shutdown Control—active low (MAX213E)                                                                                                                                         |

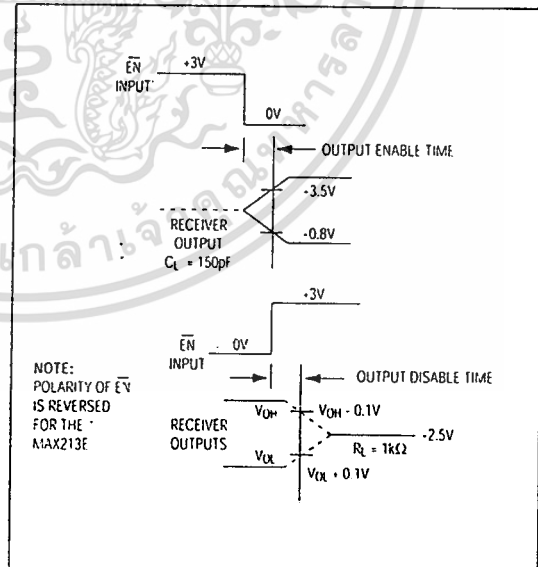


Figure 2 Receiver Output Enable and Disable Timing  
(A14X205E/A14X206E/MAX211E/MAX213E/MAX241E)

MAXIMI

## **$\pm 15\text{kV}$ ESD-Protected, $+5\text{V}$ RS-232 Transceivers**

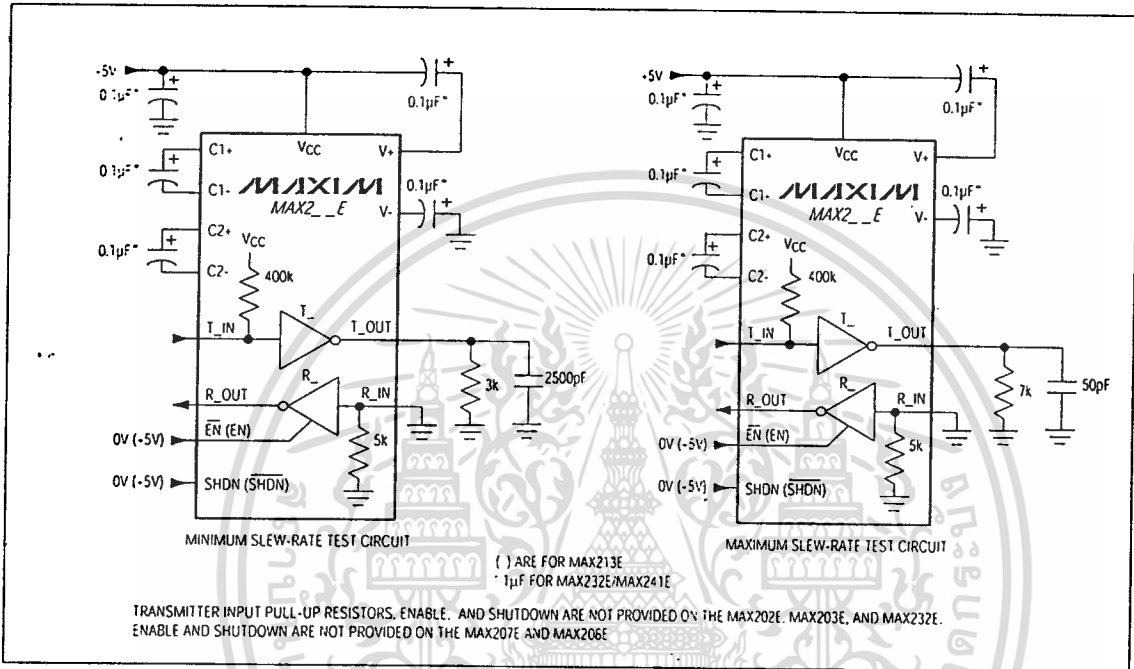


Figure 3. Transition Slew-Rate Circuit

### **Detailed Description**

The MAX202E-MAX213E, MAX232E/MAX241E consist of three sections: charge-pump voltage converters, drivers (transmitters), and receivers. These E versions provide extra protection against ESD. They survive  $\pm 15\text{kV}$  discharges to the RS-232 inputs and outputs, tested using the Human Body Model. When tested according to IEC1000-4-2, they survive  $\pm 8\text{kV}$  contact discharges and  $\pm 15\text{kV}$  air-gap discharges. The rugged E versions are intended for use in harsh environments or applications where the RS-232 connection is frequently changed (such as notebook computers). The standard (non-"E") MAX202, MAX203, MAX205-MAX208, MAX211, MAX213, MAX232, and MAX241 are recommended for applications where cost is critical.

#### **$+5\text{V}$ to $\pm 10\text{V}$ Dual Charge-Pump Voltage Converter**

The  $-5\text{V}$  to  $\pm 10\text{V}$  conversion is performed by dual charge-pump voltage converters (Figure 4). The first charge-pump converter uses capacitor C1 to double the  $-5\text{V}$  into  $+10\text{V}$ , storing the  $+10\text{V}$  on the output filter capacitor C3. The second uses C2 to invert the  $+10\text{V}$

into  $-10\text{V}$ , storing the  $-10\text{V}$  on the  $V_-$  output filter capacitor, C4.

In shutdown mode,  $V_+$  is internally connected to  $V_{CC}$  by a  $1\text{k}\Omega$  pull-down resistor, and  $V_-$  is internally connected to ground by a  $1\text{k}\Omega$  pull-up resistor.

#### **RS-232 Drivers**

With  $V_{CC} = 5\text{V}$ , the typical driver output voltage swing is  $\pm 8\text{V}$  when loaded with a nominal  $5\text{k}\Omega$  RS-232 receiver. The output swing is guaranteed to meet EIA/TIA-232E and V.28 specifications that call for  $\pm 5\text{V}$  minimum output levels under worst-case conditions. These include a  $3\text{k}\Omega$  load, minimum  $V_{CC}$ , and maximum operating temperature. The open-circuit output voltage swings from  $(V_+ - 0.6\text{V})$  to  $V_-$ .

Input thresholds are CMOS/TTL compatible. The unused drivers' inputs on the MAX205E-MAX208E, MAX211E, MAX213E, and MAX241E can be left unconnected because  $400\text{k}\Omega$  pull-up resistors to  $V_{CC}$  are included on-chip. Since all drivers invert, the pull-up resistors force the unused drivers' outputs low. The MAX202E, MAX203E, and MAX232E do not have pull-up resistors on the transmitter inputs.

MAX202E-MAX213E, MAX232E/MAX241E

MAXIM

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## $\pm 15\text{kV}$ ESD-Protected, +5V RS-232 Transceivers

When in low-power shutdown mode, the MAX205E/MAX206E/MAX211E/MAX213E/MAX241E driver outputs are turned off and draw only leakage currents—even if they are back-driven with voltages between 0V and 12V. Below -0.5V in shutdown, the transmitter output is diode-clamped to ground with a 1k $\Omega$  series impedance.

### RS-232 Receivers

The receivers convert the RS-232 signals to CMOS-logic output levels. The guaranteed 0.8V and 2.4V receiver input thresholds are significantly tighter than the  $\pm 3\text{V}$  thresholds required by the EIA/TIA-232E specification. This allows the receiver inputs to respond to TTL/CMOS-logic levels, as well as RS-232 levels.

The guaranteed 0.8V input low threshold ensures that receivers shorted to ground have a logic 1 output. The 5k $\Omega$  input resistance to ground ensures that a receiver with its input left open will also have a logic 1 output.

Receiver inputs have approximately 0.5V hysteresis. This provides clean output transitions, even with slow rise/fall-time signals with moderate amounts of noise and ringing.

In shutdown, the MAX213E's R4 and R5 receivers have no hysteresis.

### Shutdown and Enable Control (MAX205E/MAX206E/MAX211E/ MAX213E/MAX241E)

In shutdown mode, the charge pumps are turned off, V<sub>+</sub> is pulled down to V<sub>CC</sub>, V<sub>-</sub> is pulled to ground, and the transmitter outputs are disabled. This reduces supply current typically to 1 $\mu\text{A}$  (15 $\mu\text{A}$  for the MAX213E). The time required to exit shutdown is under 1ms, as shown in Figure 5:

### Receivers

For MAX213E receivers, except R4 and R5, are put into a high-impedance state in shutdown mode (see Tables 1a and 1b). The MAX213E's R4 and R5 receivers still function in shutdown mode. These two awake-in-shutdown receivers can monitor external activity while maintaining minimal power consumption.

The enable control is used to put the receiver outputs into a high-impedance state, to allow wire-OR connection of two EIA/TIA-232E ports (or ports of different types) at the UART. It has no effect on the RS-232 drivers or the charge pumps.

Note: The enable control pin is active low for the MAX211E/MAX241E (EN), but is active high for the MAX213E (EN). The shutdown control pin is active high for the MAX205E/MAX206E/MAX211E/MAX241E (SHDN), but is active low for the MAX213E (SHDN).

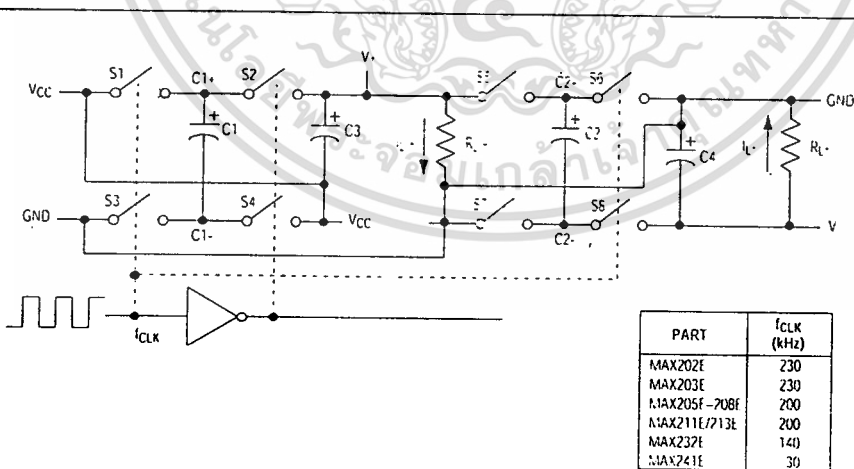


Figure 4 Charge-Pump Diagram

MAXIM

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# ±15kV ESD-Protected, +5V RS-232 Transceivers

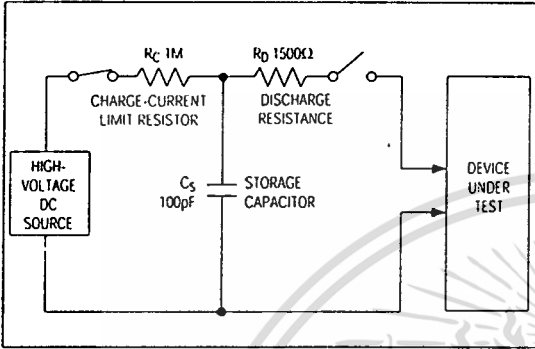


Figure 6a. Human Body ESD Test Model

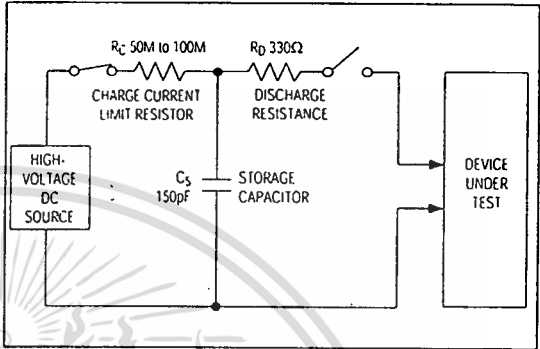


Figure 7a. IEC1000-4-2 ESD Test Model

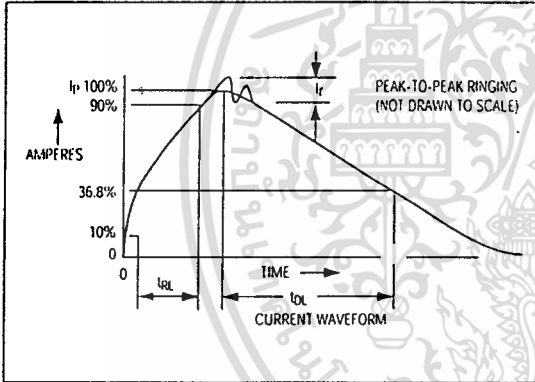


Figure 6b. Human Body Model Current Waveform

## IEC1000-4-2

The IEC1000-4-2 standard covers ESD testing and performance of finished equipment; it does not specifically refer to integrated circuits. The MAX202E/MAX203E-MAX213E, MAX232E/MAX241E help you design equipment that meets level 4 (the highest level) of IEC1000-4-2, without the need for additional ESD-protection components.

The major difference between tests done using the Human Body Model and IEC1000-4-2 is higher peak current in IEC1000-4-2, because series resistance is lower in the IEC1000-4-2 model. Hence, the ESD withstand voltage measured to IEC1000-4-2 is generally lower than that measured using the Human Body Model. Figure 7b shows the current waveform for the 8kV IEC1000-4-2 level-four ESD contact-discharge test.

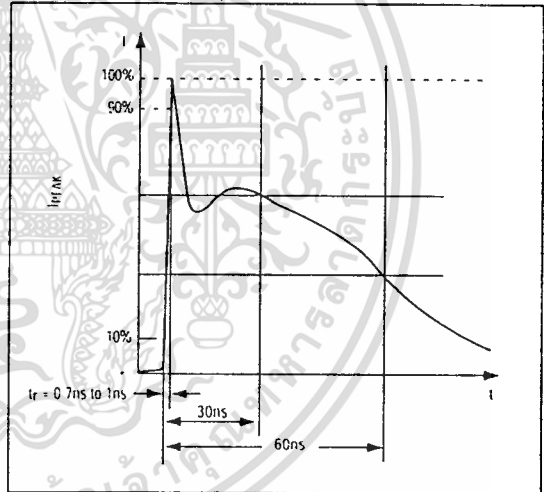


Figure 7b. IEC1000-4-2 ESD Generator Current Waveform

The air-gap test involves approaching the device with a charged probe. The contact-discharge method connects the probe to the device before the probe is energized.

## Machine Model

The Machine Model for ESD tests all pins using a 200pF storage capacitor and zero discharge resistance. Its objective is to emulate the stress caused by contact that occurs with handling and assembly during manufacturing. Of course, all pins require this protection during manufacturing, not just RS-232 inputs and outputs. Therefore, after PC board assembly, the Machine Model is less relevant to I/O ports.

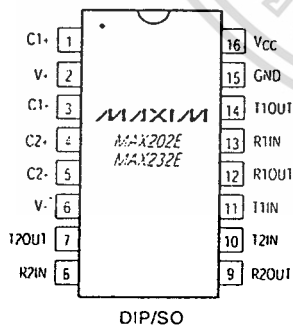
MAXIM

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

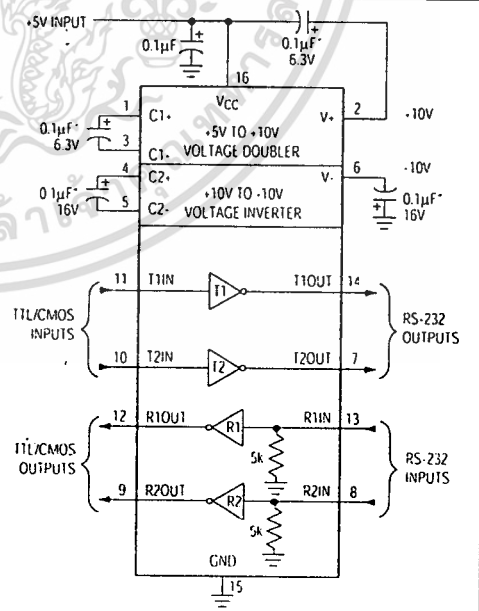
**Table 3. DB9 Cable Connections  
Commonly Used for EIA/TIAE-232E and  
V.24 Asynchronous Interfaces**

| PIN | CONNECTION                                                           |                             |
|-----|----------------------------------------------------------------------|-----------------------------|
| 1   | Received Line Signal Detector (sometimes called Carrier Detect. DCD) | Handshake from DCE          |
| 2   | Receive Data (RD)                                                    | Data from DCE               |
| 3   | Transmit Data (TD)                                                   | Data from DTE               |
| 4   | Data Terminal Ready                                                  | Handshake from DTE          |
| 5   | Signal Ground                                                        | Reference point for signals |
| 6   | Data Set Ready (DSR)                                                 | Handshake from DCE          |
| 7   | Request to Send (RTS)                                                | Handshake from DTE          |
| 8   | Clear to Send (CTS)                                                  | Handshake from DCE          |
| 9   | Ring Indicator                                                       | Handshake from DCE          |

TOP VIEW



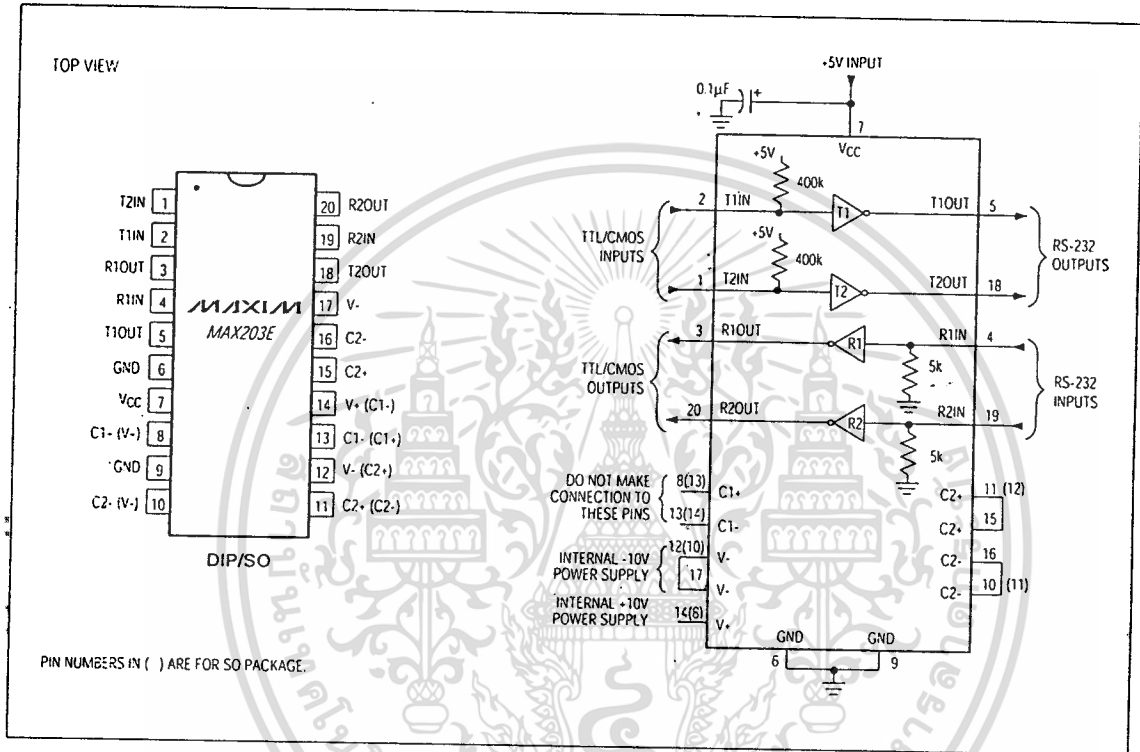
PIN NUMBERS ON TYPICAL OPERA-1.7 CIRCUIT REFER TO DIP/SO PACKAGE, NOT LCC  
100nF CAPACITORS, MAX232E ONLY



MAXIM

# **$\pm 15\text{kV}$ ESD-Protected, +5V RS-232 Transceivers**

## **Pin Configurations and Typical Operating Circuits (continued)**



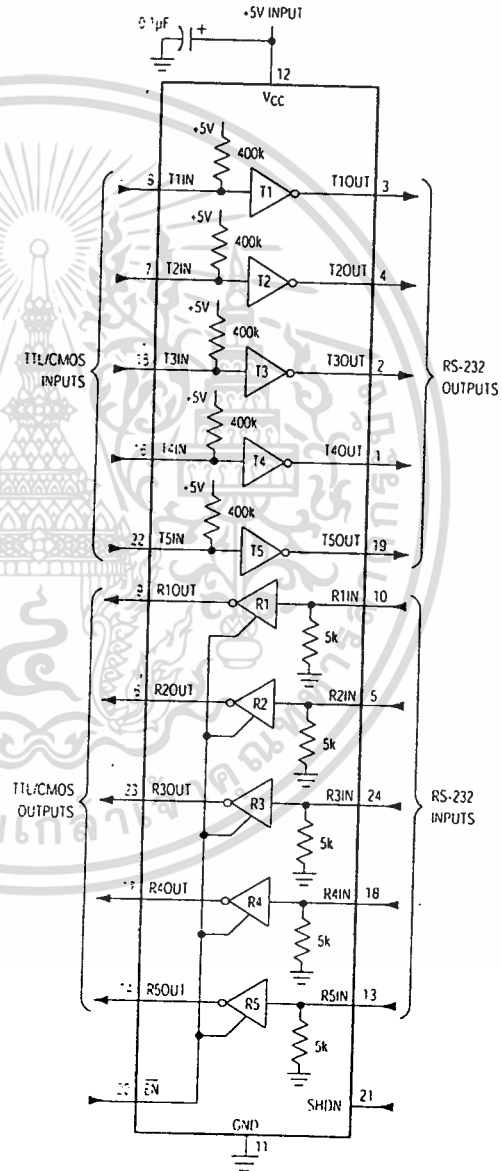
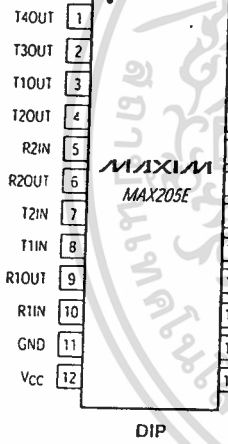
MAXIM

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# $\pm 15\text{kV}$ ESD-Protected, +5V RS-232 Transceivers

## Pin Configurations and Typical Operating Circuits (continued)

TOP VIEW



MAXIM

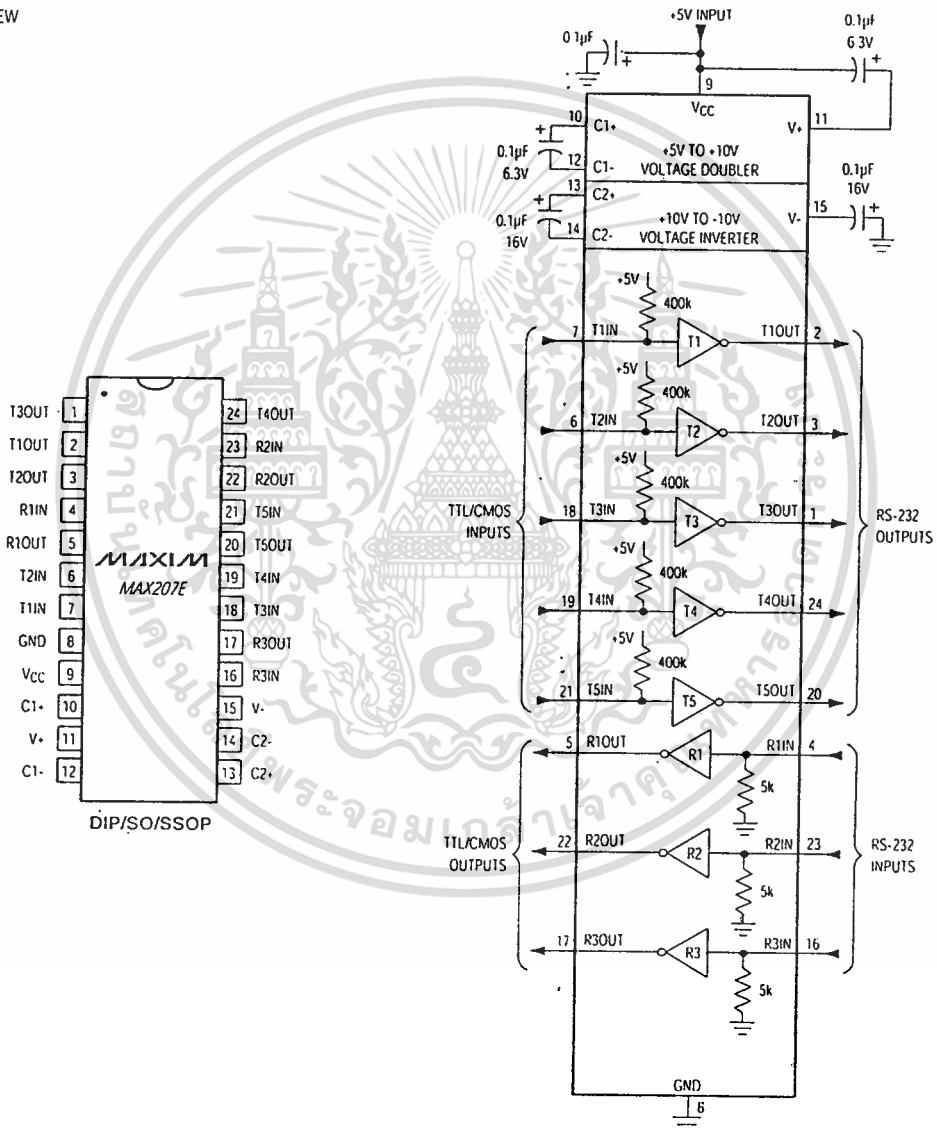
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



# $\pm 15\text{kV}$ ESD-Protected, +5V RS-232 Transceivers

## Pin Configurations and Typical Operating Circuits (continued)

TOP VIEW



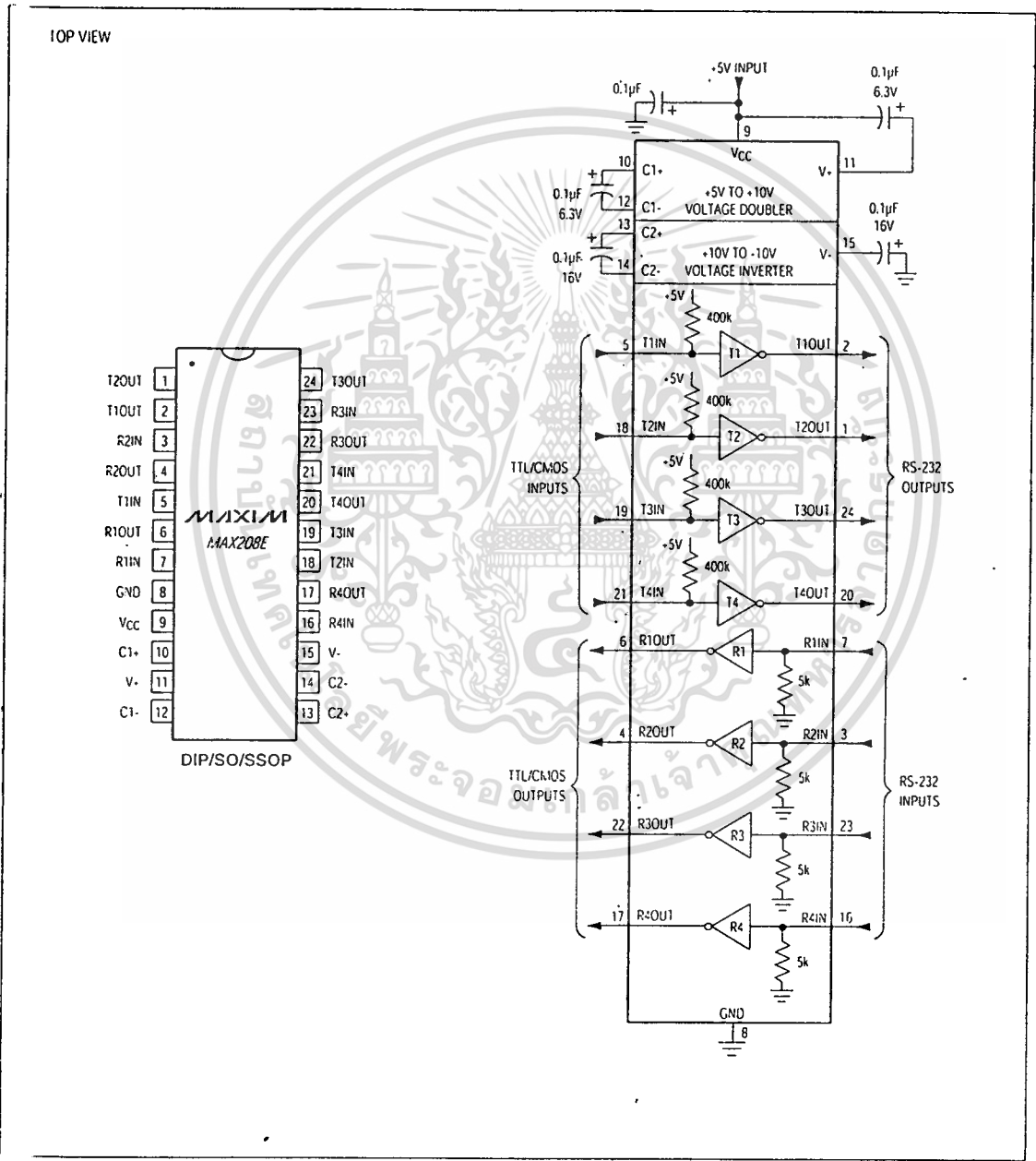
MAXIM

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**$\pm 15\text{kV}$  ESD-Protected, +5V RS-232 Transceivers**

**Pin Configurations and Typical Operating Circuits (continued)**

**MAX202E-MAX213E, MAX232E/MAX241E**



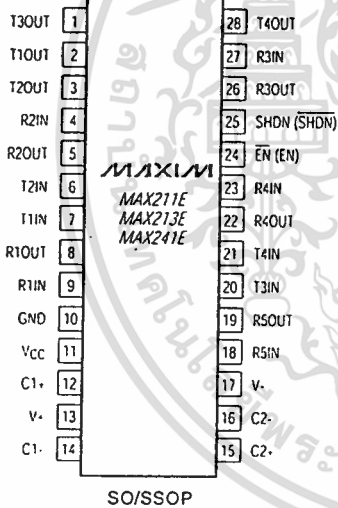
MAXIM

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

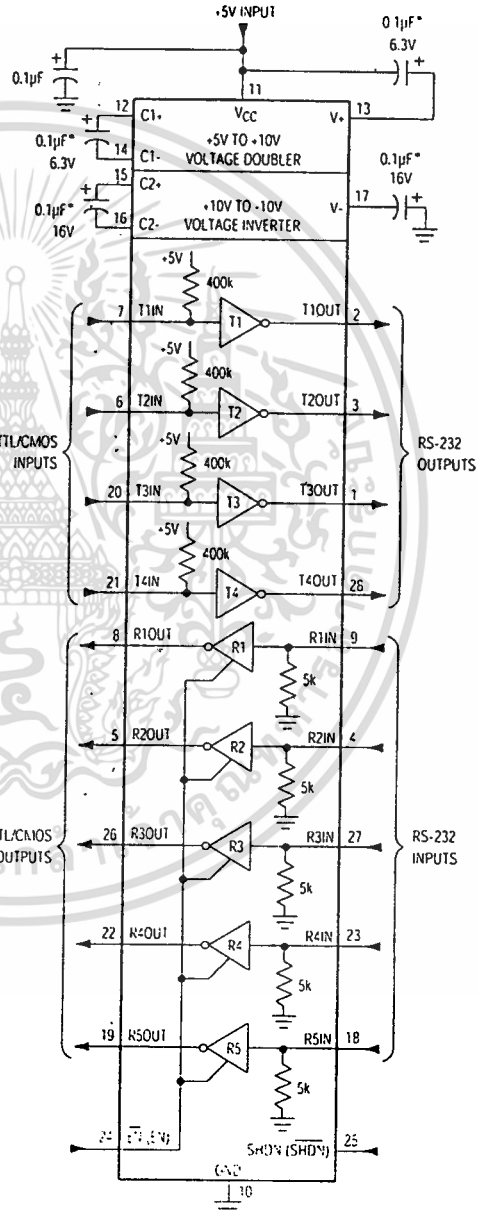
# ±15kV ESD-Protected, +5V RS-232 Transceivers

## Pin Configurations and Typical Operating Circuits (continued)

TOP VIEW



ARE FOR MAX213E ONLY  
10pF CAPACITORS, MAX241E ONLY



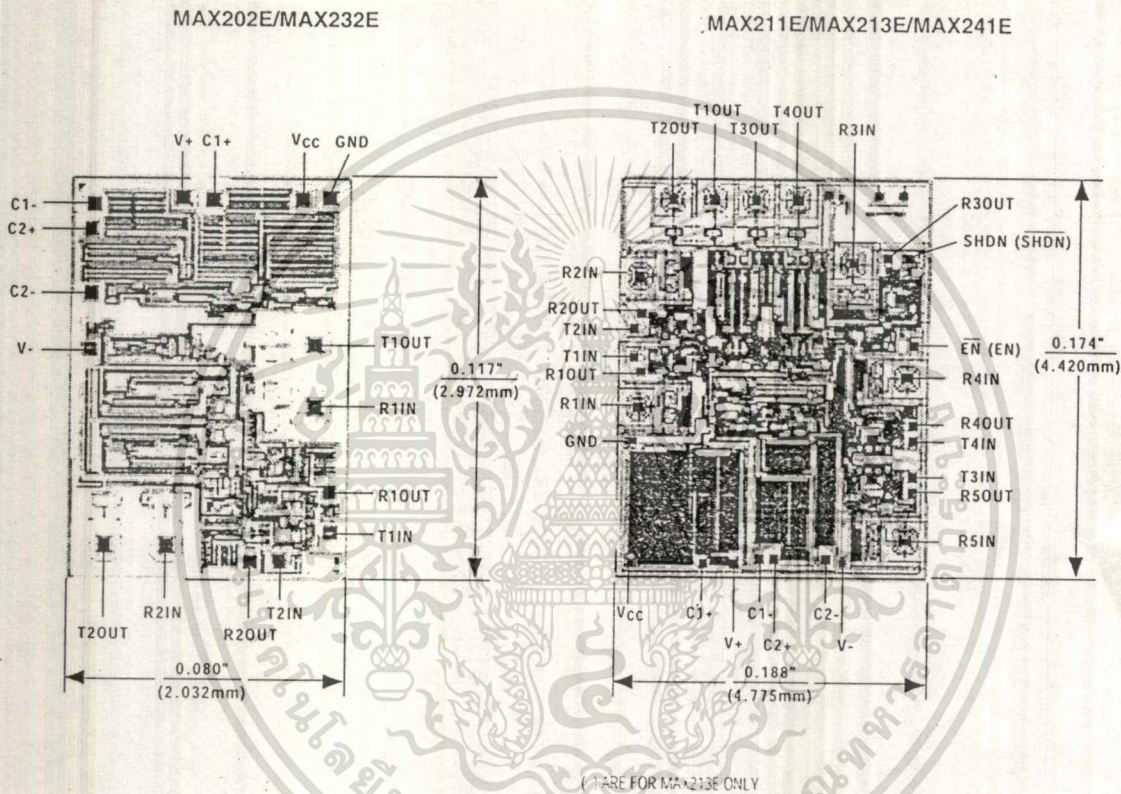
MAXIM

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

MAX202E-MAX213E, MAX232E/MAX241E

**$\pm 15\text{kV}$  ESD-Protected, +5V RS-232 Transceivers**

*Chip Topographies*



**Chip Information**

MAX205E/MAX206E/MAX207E/MAX208E

TRANSISTOR COUNT: 328

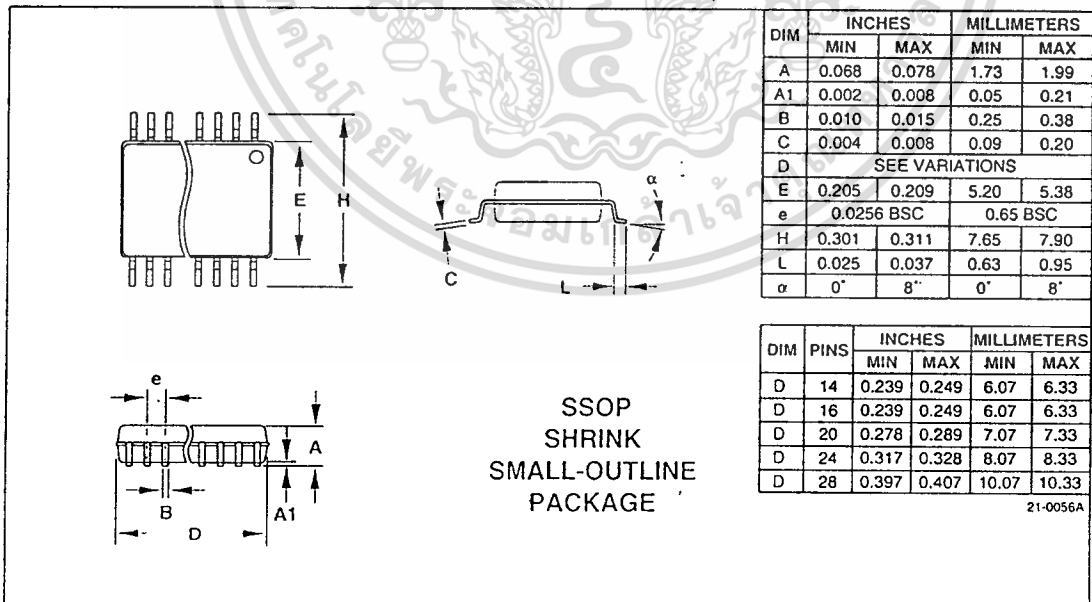
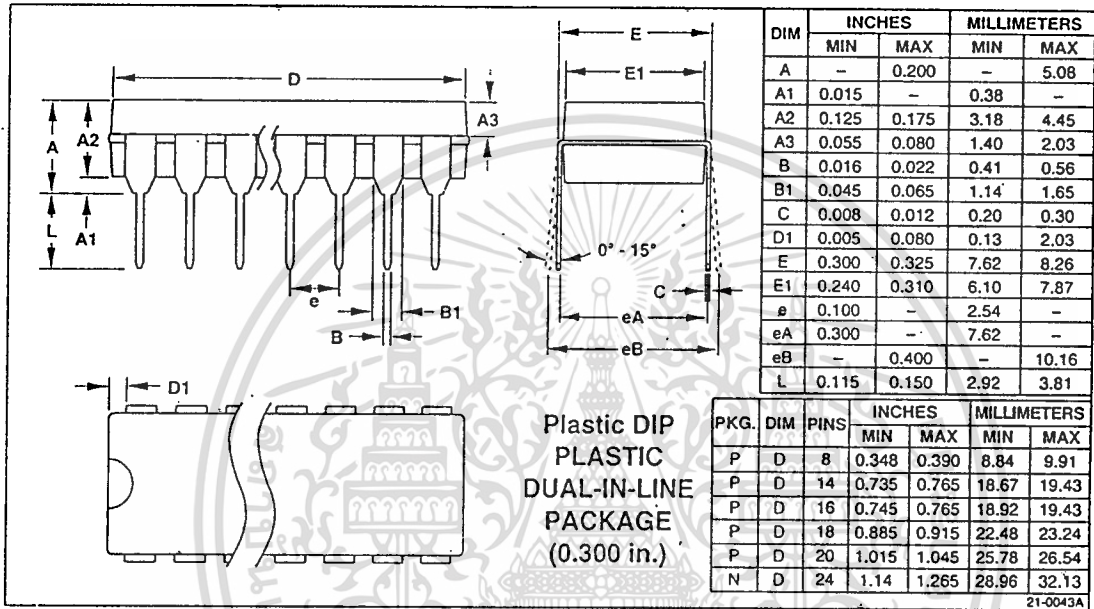
SUBSTRATE CONNECTED TO GND

MAXIM

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# $\pm 15\text{kV}$ ESD-Protected, +5V RS-232 Transceivers

## Package Information



MAX202E-MAX213E, MAX232E/MAX241E

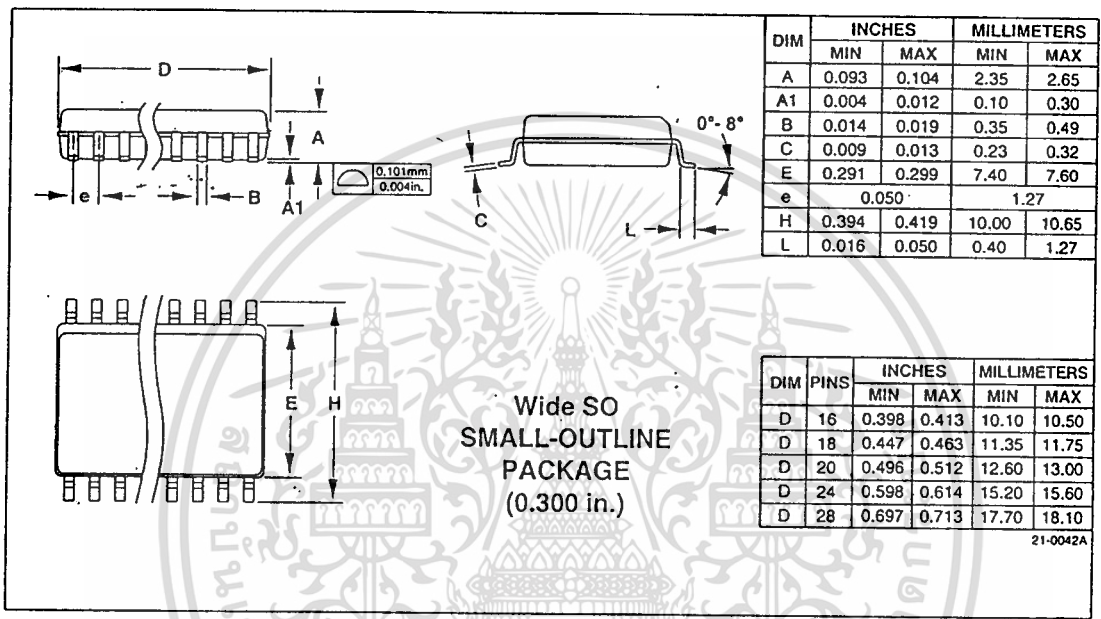
MAXIM

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

MAX202E-MAX213E, MAX232E/MAX241E

±15kV ESD-Protected, +5V RS-232 Transceivers

Package Information (continued)



Maxim cannot assume responsibility for use of any circuitry other than circuitry entirely embodied in a Maxim product. No circuit patent licenses are implied. Maxim reserves the right to change the circuitry and specifications without notice at any time.

Maxim Integrated Products, 120 San Gabriel Drive, Sunnyvale, CA 94086 (408) 737-7600

© 1996 Maxim Integrated Products Printed USA MAXIM is a registered trademark of Maxim Integrated Products.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# ALTERA

## a8255

### Programmable Peripheral Interface Adapter

September 1996, ver. 1.0

## Features

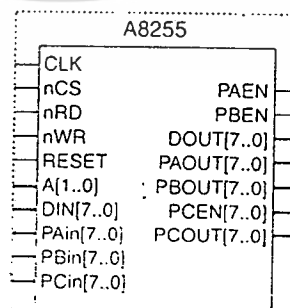
- a8255 MegaCore function implementing a programmable peripheral interface adapter
- Optimized for FLEX<sup>®</sup> and MAX<sup>®</sup> architectures
- 24 programmable inputs/outputs
- Static read/write or handshaking modes
- Direct bit set/reset capability
- Synchronous design
- Uses approximately 194 FLEX logic elements (LEs)
- Functionally based on the Intel 8255A and Harris 82C55A devices, except as noted in the "Variations & Clarifications" section on page 56

## General Description

The a8255 MegaCore function implements a programmable peripheral interface adapter (see Figure 1). The a8255 has 24 I/O signals that can be programmed in two groups of 12. This MegaCore function operates in the following three modes:

- *Mode 0: Basic Input/Output*—Port A, port B, and port C (upper and lower) can be independently configured as inputs or outputs to read or hold static data. Outputs are registered; inputs are not registered.
- *Mode 1: Strobed Input/Output*—Port A and port B can be independently configured as strobed input or output buses. Signals from port C are dedicated as control signals for data handshaking.
- *Mode 2: Bidirectional Bus*—Port A can be configured as a bidirectional bus with the majority of port C providing the control signals. In this configuration, port B can still implement mode 0 or mode 1.

Figure 1. a8255 Symbol



Altera Corporation

A-DS-8255-01

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## a8255 Programmable Peripheral Interface Adapter Data Sheet

Table 1 describes the input and output ports of the a8255.

*Table 1. a8255 Ports*

| Name        | Type   | Polarity | Description                                                                                                                  |
|-------------|--------|----------|------------------------------------------------------------------------------------------------------------------------------|
| clk         | Input  | –        | Clock.                                                                                                                       |
| ncs         | Input  | Low      | Chip select. When ncs is asserted, the a8255 is selected and read and write transactions to internal registers are possible. |
| nrd         | Input  | Low      | Read control. When nrd is asserted and the a8255 is selected, read transactions from internal registers are possible.        |
| nwr         | Input  | Low      | Write control. When nwr is asserted and the a8255 is selected, write transactions to internal registers are possible.        |
| reset       | Input  | High     | Reset. Initializes the control and port C output registers, and sets the port A, B, and C registers to input mode.           |
| a[1..0]     | Input  | High     | Register address bus. This bus selects one of the internal registers.                                                        |
| din[7..0]   | Input  | High     | Data input bus. The CPU writes data to the internal control, port A, port B, or port C register via the din[7..0] bus.       |
| pain[7..0]  | Input  | High     | Port A input data bus.                                                                                                       |
| pbin[7..0]  | Input  | High     | Port B input data bus.                                                                                                       |
| pcin[7..0]  | Input  | High     | Port C input data bus.                                                                                                       |
| paen        | Output | High     | Port A data enable. Output enable for the port A output data bus.                                                            |
| pben        | Output | High     | Port B data enable. Output enable for the port B output data bus.                                                            |
| dout[7..0]  | Output | High     | Data output bus. The CPU reads data from the internal control, port A, port B, or port C register via the dout[7..0] bus.    |
| paout[7..0] | Output | High     | Port A output data bus.                                                                                                      |
| pbout[7..0] | Output | High     | Port B output data bus.                                                                                                      |
| pcen[7..0]  | Output | High     | Port C data enable bus. Output enable for each bit of the port C output data bus.                                            |
| pcout[7..0] | Output | High     | Port C output data bus.                                                                                                      |

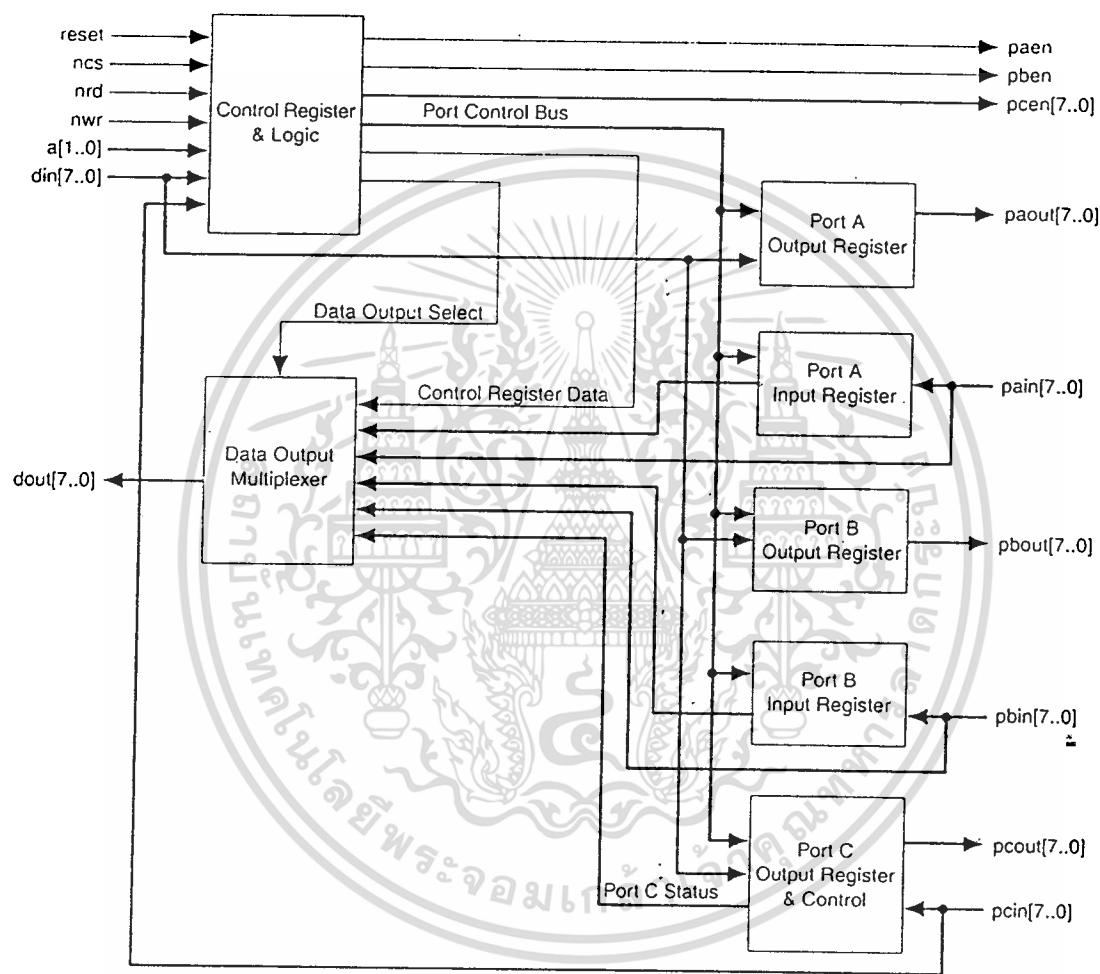
Altera Corporation

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Functional Description

Figure 2 shows a block diagram of the a8255.

Figure 2. a8255 Block Diagram



## Register Address Map

Table 2 shows the register address map for the a8255.

| <i>Table 2. Register Address Map</i> |    |                                                           |
|--------------------------------------|----|-----------------------------------------------------------|
| a1                                   | a0 | Register                                                  |
| 0                                    | 0  | Port A data (all modes)                                   |
| 0                                    | 1  | Port B data (all modes)                                   |
| 1                                    | 0  | Port C data (mode 0) and status (modes 1 and 2)           |
| 1                                    | 1  | Control register mode definition and port C bit set/reset |

## Registers

This section describes the following a8255 registers:

- Control
- Port A, B & C

### Control Register

The control register sets the mode and signal direction for the three 8-bit I/O ports. Control of the I/O ports is split into two groups. Group A consists of port A and the upper four bits of port C; group B consists of port B and the lower four bits of port C. Group A can be set to mode 0, mode 1, or mode 2, but group B can be set to only mode 0 or mode 1.

Writing to the control register address with bit 7 set is the mode definition format, which allows control of the mode and direction of the three I/O ports (see Table 3). Writing to the control register address with bit 7 reset is the port C bit set/reset format, which allows single-bit control of port C (see Table 4). The CPU reads the control register using the mode definition format.

**Table 3. Control Register Mode Definition Format**

| Bit  | Description                                                                           |
|------|---------------------------------------------------------------------------------------|
| 0    | Port C (lower) I/O direction:<br>1 = input<br>0 = output                              |
| 1    | Port B I/O direction:<br>1 = input<br>0 = output                                      |
| 2    | Group B mode select:<br>1 = mode 1<br>0 = mode 0                                      |
| 3    | Port C (upper) I/O direction:<br>1 = input<br>0 = output                              |
| 4    | Port A I/O direction:<br>1 = input<br>0 = output                                      |
| 6..5 | Group A mode select:<br>00 = mode 0<br>01 = mode 1<br>1X = mode 2, <i>Note (1)</i>    |
| 7    | 1 when writing = mode definition format<br>Always 1 when reading the control register |

**Note:**

- (1) The X indicates "don't care."

**Table 4. Port C Bit Set/Reset Format *Note (1)***

| Bit  | Description                                  |
|------|----------------------------------------------|
| 0    | Bit set/reset:<br>1 = set<br>0 = reset       |
| 3..1 | Bit select address                           |
| 6..4 | XXX, <i>Note (2)</i>                         |
| 7    | 0 when writing = port C bit set/reset format |

**Notes:**

- (1) For example, to reset bit 3 of port C, bit 7 is reset to indicate that the write is in the port C bit set/reset format. Bits 6 through 4 are "don't care." Bits 3 through 1 are 011 to address bit 3 and bit 0 is 0 to indicate a reset operation. The complete data word is 0XXX011.
- (2) The X indicates "don't care."

Port A, B & C Registers

Depending on the configured input and output directions that are set in the control register, the microprocessor either reads or writes data to/from the port A, B, or C registers. Ports A and B have separate input and output registers.

In mode 0, the port C register functions identically to the port A and B registers. In modes 1 and 2, the port C register has a specialized role; a write to port C has no effect—the register bits must be altered individually using the port C bit set/reset format. Reading the port C status bits in modes 1 or 2 provides the CPU with the status of the control signals and flags, as shown in Tables 5 through 7. However, the modes and port directions can be mixed in more combinations than these tables illustrate.

| Table 5. Port C Status Bits with Ports A & B in Mode 1 Input    Note (1) |        |                               |
|--------------------------------------------------------------------------|--------|-------------------------------|
| Bit                                                                      | Signal | Description                   |
| 0                                                                        | intrb  | Port B interrupt request      |
| 1                                                                        | ibfb   | Port B input buffer full flag |
| 2                                                                        | inteb  | Port B interrupt enable       |
| 3                                                                        | intra  | Port A interrupt request      |
| 4                                                                        | intea  | Port A interrupt enable       |
| 5                                                                        | ibfa   | Port A input buffer full flag |
| 6                                                                        | I/O    | Note (2)                      |
| 7                                                                        | I/O    | Note (2)                      |

Notes:

- (1) These bits are defined in Table 9.
- (2) Bits 6 and 7 effectively operate in mode 0. The I/O direction is dependent on bit 3 of the control register.

| Table 6. Port C Status Bits with Ports A & B in Mode 1 Output    Note (1) |        |                                |
|---------------------------------------------------------------------------|--------|--------------------------------|
| Bit                                                                       | Signal | Description                    |
| 0                                                                         | intrb  | Port B interrupt request       |
| 1                                                                         | nobfb  | Port B output buffer full flag |
| 2                                                                         | inteb  | Port B interrupt enable        |
| 3                                                                         | intra  | Port A interrupt request       |
| 4                                                                         | I/O    | Note (2)                       |
| 5                                                                         | I/O    | Note (2)                       |
| 6                                                                         | intea  | Port A interrupt enable        |
| 7                                                                         | nobfa  | Port A output buffer full flag |

- Notes:
- (1) These bits are defined in Table 9.
  - (2) Bits 4 and 5 effectively operate in mode 0. The I/O direction is dependent on bit 3 of the control register.

| Table 7. Port C Status Bits with Port A in Mode 2    Note (1) |        |                                |
|---------------------------------------------------------------|--------|--------------------------------|
| Bit                                                           | Signal | Description                    |
| 0                                                             | -      | Note (2)                       |
| 1                                                             | -      | Note (2)                       |
| 2                                                             | -      | Note (2)                       |
| 3                                                             | intra  | Port A interrupt request       |
| 4                                                             | inte2  | Interrupt enable 2             |
| 5                                                             | ibfa   | Port A input buffer full flag  |
| 6                                                             | intel  | Interrupt enable 1             |
| 7                                                             | nobfa  | Port A output buffer full flag |

- Notes:
- (1) These bits are defined in Table 11.
  - (2) Depending on the lower three bits of the control register, bits 0 through 2 either operate in mode 0 or function as status bits for port B in mode 1.

Operation

The a8255 operates in the following three modes:

- Mode 0: basic input/output
- Mode 1: strobed input/output
- Mode 2: strobed bidirectional bus

### Mode 0: Basic Input/Output

Mode 0 is used to perform simple reads or writes of relatively static signals, such as switches or status displays. Port A, port B, port C (upper), and port C (lower) can be independently configured as inputs or outputs without requiring handshake signals. Data written by the CPU to a port configured as an output is registered; data read by the CPU from a port configured as an input is not registered.

### Mode 1: Strobed Input/Output

Mode 1 is used to perform reads or writes of data controlled by handshake signals. Ports A and B are the data ports, configured independently as either inputs or outputs. Port C provides the three handshaking signals for each of the data ports. Both input and output data are registered.

Table 8 shows how the handshaking signals are configured for mode 1 input.

| Table 8. Handshaking Signal Configuration (Mode 1 Input) |                      |                                                                                                                                                                      |
|----------------------------------------------------------|----------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Name                                                     | Signal Type          | Description                                                                                                                                                          |
| nstb                                                     | Input                | Strobe. Enable for input register.                                                                                                                                   |
| ibf                                                      | Output               | Input buffer full flag. When set, indicates that data has been loaded into the input register. Set by nstb going low, and reset by the rising edge of the nrd input. |
| intr                                                     | Output               | Interrupt request. Can be used as the interrupt signal to the CPU. Set on the rising edge of nstb when inte is high. Reset by the falling edge of nrd.               |
| inte                                                     | Internal control bit | Interrupt enable. Set by bit set to PC4 for port A and PC2 for port B.                                                                                               |

Table 9 shows how the handshaking signals are configured for mode 1 output.

| Table 9. Handshaking Signal Configuration (Mode 1 Output) |                      |                                                                                                                                                                                                                                  |
|-----------------------------------------------------------|----------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Name                                                      | Signal Type          | Description                                                                                                                                                                                                                      |
| nobf                                                      | Output               | Output buffer full flag. Indicates that data has been written to the port. Goes low on the rising edge of nwr, and returns high when nack is asserted. The rising edge of nobf should be used to latch data into the peripheral. |
| nack                                                      | Input                | Acknowledge. Indicates the peripheral is ready to latch the output data.                                                                                                                                                         |
| intr                                                      | Output               | Interrupt request. Can be used as the interrupt signal to the CPU, which indicates that the peripheral device has latched the data. Reset on the falling edge of nwr; set on the rising edge of nack when inte is high.          |
| inte                                                      | Internal control bit | Interrupt enable. Set by bit set to PC6 for port A and PC2 for port B.                                                                                                                                                           |

Table 10 summarizes the configuration of port C when both port A and port B are configured as mode 1.

| Table 10. Port C with Port A & Port B Both Configured as Mode 1 |              |               |                                                                               | Note (1) |
|-----------------------------------------------------------------|--------------|---------------|-------------------------------------------------------------------------------|----------|
| Bit                                                             | Mode 1 Input | Mode 1 Output | Description                                                                   |          |
| PC0                                                             | intrb        | intrb         | Always output.                                                                |          |
| PC1                                                             | ibfb         | nobfb         | Always output.                                                                |          |
| PC2                                                             | nstbb        | nackb         | Always input.                                                                 |          |
| PC3                                                             | intra        | intra         | Always output.                                                                |          |
| PC4                                                             | nstba        | I/O           | I/O direction configured by bit 3 of the control register in "mode 1 output." |          |
| PC5                                                             | ibfa         | I/O           | I/O direction configured by bit 3 of the control register in "mode 1 output." |          |
| PC6                                                             | I/O          | nacka         | I/O direction configured by bit 3 of the control register in "mode 1 input."  |          |
| PC7                                                             | I/O          | nobfa         | I/O direction configured by bit 3 of the control register in "mode 1 input."  |          |

Note:  
(1) The interrupt enable control bits (intea and inteb) are stored in the output register bits PC2, PC4, and PC6.

Mode 2: Strobed Bidirectional Bus

Mode 2 is used to perform reads and writes of data over a bidirectional bus controlled by handshake signals. Port A is the only data port capable of mode 2 operation, while port C provides the five control signals for this data port. Both input and output data are registered. Table 11 shows the configuration for the bidirectional bus.

| Table 11. Bidirectional Bus Configuration |                      |                                                                                                                                                                                                                                                                                     |
|-------------------------------------------|----------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Name                                      | Signal Type          | Description                                                                                                                                                                                                                                                                         |
| nstb                                      | Input                | Strobe. Enable for input register.                                                                                                                                                                                                                                                  |
| nack                                      | Input                | Acknowledge. Indicates that the peripheral is ready to latch the output data. Acts as the tri-state enable for port A.                                                                                                                                                              |
| ibf                                       | Output               | Input buffer full flag. When set, indicates data has been loaded into the input register. Set by nstb going low, and reset by the rising edge of the nrd input.                                                                                                                     |
| nobf                                      | Output               | Output buffer full flag. Indicates that data has been written to port A. Reset on the rising edge of nwr, and set when nack goes low.                                                                                                                                               |
| intr                                      | Output               | Interrupt request. Can be used as the interrupt signal to the CPU that indicates the peripheral has latched the data. Reset on the falling edge of nwr or falling edge of nrd. Set on the rising edge of nack when inte1 is high, or on the rising edge of nstb when inte2 is high. |
| inte1                                     | Internal control bit | Interrupt enable 1. Set by bit set to PC6.                                                                                                                                                                                                                                          |
| inte2                                     | Internal control bit | Interrupt enable 2. Set by bit set to PC4.                                                                                                                                                                                                                                          |

Table 12 summarizes the configuration of port C in mode 2.

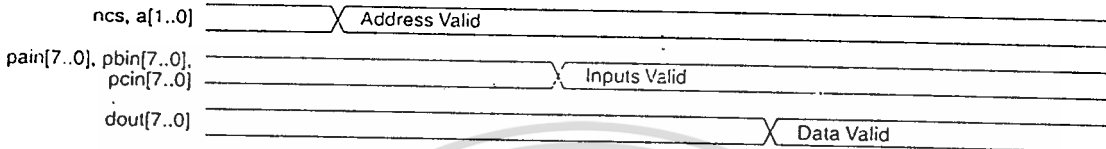
| Table 12. Port C Configuration in Mode 2 |        |                                    |
|------------------------------------------|--------|------------------------------------|
| I/O                                      | Mode 2 | Description                        |
| PC0                                      | I/O    | Dependent on group B configuration |
| PC1                                      | I/O    | Dependent on group B configuration |
| PC2                                      | I/O    | Dependent on group B configuration |
| PC3                                      | intra  | Output                             |
| PC4                                      | nstba  | Input                              |
| PC5                                      | ibfa   | Output                             |
| PC6                                      | nacka  | Input                              |
| PC7                                      | nobfa  | Output                             |

## Timing Waveforms

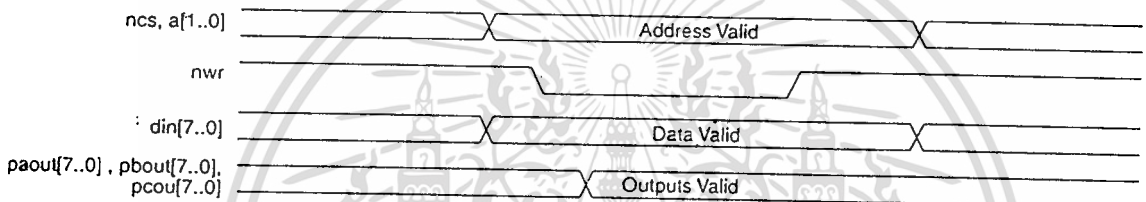
Figures 3 and 4 shows the functional timing waveforms for the a8255.

Figure 3. a8255 Mode 0 and Mode 1 Functional Timing Waveforms

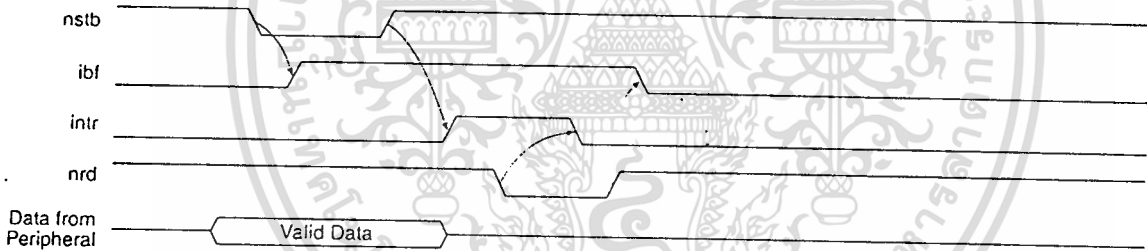
### Mode 0: Basic Input



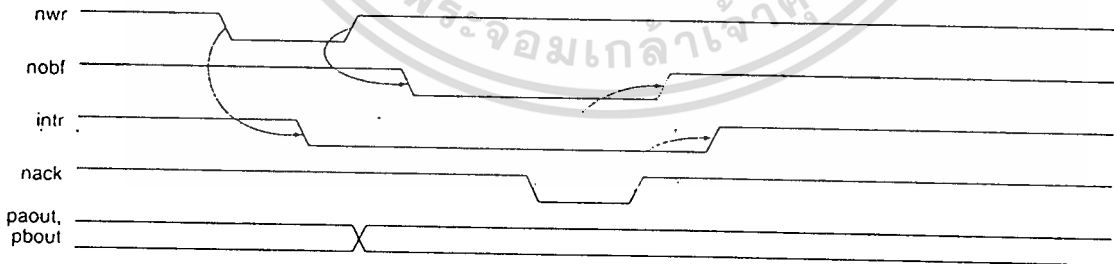
### Mode 0: Basic Output



### Mode 1: Strobed Input

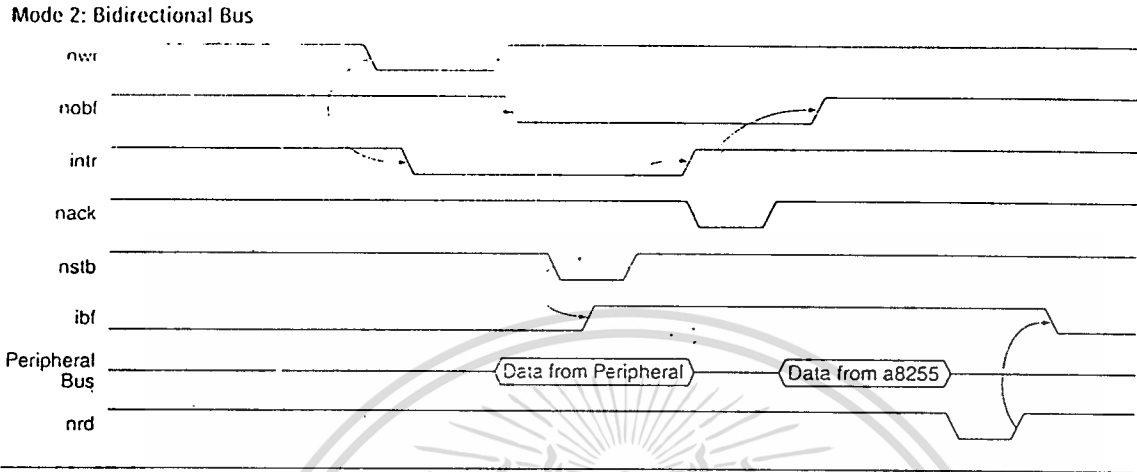


### Mode 1: Strobed Output



## a8255 Programmable Peripheral Interface Adapter Data Sheet

Figure 4. a8255 Mode 2 Functional Timing Waveforms



## Variations & Clarifications

The following characteristics distinguish the Altera a8255 from the Intel 8255A and Harris 82C55A devices:

- To allow synchronous design, a **clk** input was added as a system clock to the a8255. This capability requires that all strobes (**nrd**, **nwr**, **nstb**, and **nack**) have a minimum pulse width of one **clk** cycle.
- In the a8255, the **reset** input resets the port A, B, and C registers. In the Intel 8255A and Harris 82C55A devices, the port A, B, and C registers are unaffected by the **reset** input.
- The bidirectional buses in the Intel 8255A and Harris 82C55A devices (**d**, **pa**, **pb**, and **pc**) are split into input, output, and enable signals in the Altera a8255.
- The a8255 has no "bus hold" passive pull-ups on port signals. Because the port I/O signal is usually tied to the I/O of the Altera device, pull-ups or pull-downs can be added.
- In the a8255, the control register can be read. The control register can be read on the Harris 82C55A device, but the Intel 8255A device does not have this capability.
- In the a8255, the **reset** signal initializes the control register such that all ports are set to mode 0 inputs. Reading the control register after initialization will return the value of 9B in hexadecimal.
- After initialization in mode 1, the pertinent control signals in the port C register should be configured via the port C bit set/reset commands.
- In mode 2, every read or write of port A resets the **intra** interrupt signal.
- The Intel and Harris data sheets state that output registers and status flipflops should be reset in the event of a mode change. This feature was not included in the a8255.

Altera Corporation

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# HM6264BI Series

8,192-word  $\times$  8-bit High Speed CMOS Static RAM

# HITACHI

ADE-203-492A (Z)

Rev. 1.0

Sep. 5, 1996

## Description

The Hitachi HM6264BI is 64k-bit static RAM organized 8-kword  $\times$  8-bit. It realizes higher performance and low power consumption by 1.5  $\mu$ m CMOS process technology. The device, packaged in 450 mil SOP (foot print pitch width), 600 mil plastic DIP, is available for high density mounting.

## Features

- High speed
  - Fast access time: 100/120 ns (max)
- Low power
  - Standby: 10  $\mu$ W (typ)
  - Operation: 15 mW (typ) ( $f = 1$  MHz)
- Single 5 V supply
- Completely static memory
  - No clock or timing strobe required
- Equal access and cycle times
- Common data input and output
  - Three state output
- Directly TTL compatible
  - All inputs and outputs
- Battery backup operation capability
- Operating temperature range
  - $-40^{\circ}\text{C}$  to  $+85^{\circ}\text{C}$

## Ordering Information

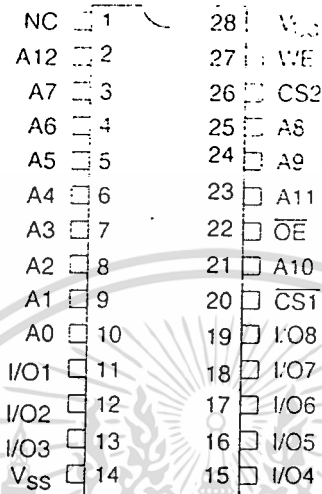
| Type No.        | Access time | Package                               |
|-----------------|-------------|---------------------------------------|
| HM6264BLPI-10   | 100 ns      | 600-mil, 28-pin plastic DIP (DP-28)   |
| HM6264BLPI-12   | 120 ns      |                                       |
| HM6264BLFPI-10T | 100 ns      | 450-mil, 28-pin plastic SOP (FP-28DA) |
| HM6264BLFPI-12T | 120 ns      |                                       |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

HM6264BI Series

Pin Arrangement

HM6264BLPI/BLFPI Series



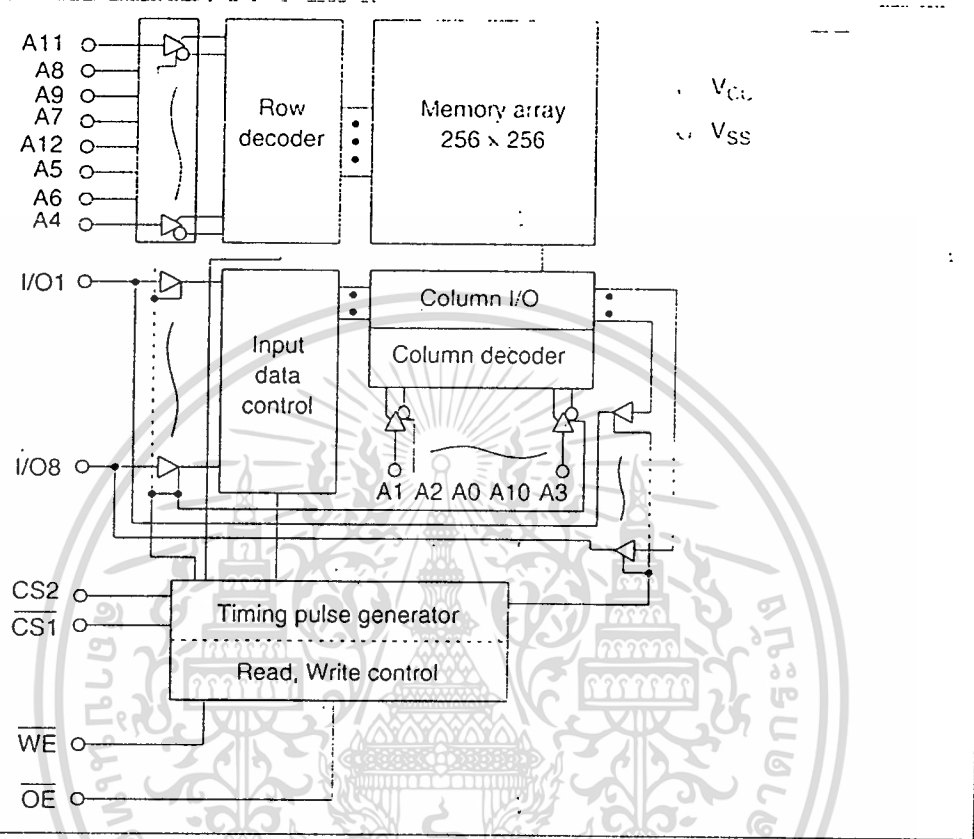
(Top view)

Pin Description

| Pin name        | Function          |
|-----------------|-------------------|
| A0 to A12       | Address input     |
| I/O1 to I/O8    | Data input/output |
| CS1             | Chip select 1     |
| CS2             | Chip select 2     |
| WE              | Write enable      |
| OE              | Output enable     |
| NC              | No connection     |
| V <sub>cc</sub> | Power supply      |
| V <sub>ss</sub> | Ground            |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Block Diagram



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

HM6264BI Series

Function Table

| WE | CS1 | CS2 | OE | Mode                      | V <sub>CC</sub> current            | I/O pin | Ref. cycle         |
|----|-----|-----|----|---------------------------|------------------------------------|---------|--------------------|
| x  | H   | x   | x  | Not selected (power down) | I <sub>SB</sub> , I <sub>SB1</sub> | High-Z  | —                  |
| x  | x   | L   | x  | Not selected (power down) | I <sub>SB</sub> , I <sub>SB1</sub> | High-Z  | —                  |
| H  | L   | H   | H  | Output disable            | I <sub>CC</sub>                    | High-Z  | —                  |
| H  | L   | H   | L  | Read                      | I <sub>CC</sub>                    | Dout    | Read cycle (1)–(3) |
| L  | L   | H   | H  | Write                     | I <sub>CC</sub>                    | Din     | Write cycle (1)    |
| L  | L   | H   | L  | Write                     | I <sub>CC</sub>                    | Din     | Write cycle (2)    |

Note: x: H or L

Absolute Maximum Ratings

| Parameter                         | Symbol            | Value                                                   | Unit |
|-----------------------------------|-------------------|---------------------------------------------------------|------|
| Power supply voltage <sup>1</sup> | V <sub>CC</sub>   | –0.5 to +7.0                                            | V    |
| Terminal voltage <sup>1</sup>     | V <sub>T</sub>    | –0.5 <sup>2</sup> to V <sub>CC</sub> + 0.3 <sup>3</sup> | V    |
| Power dissipation                 | P <sub>T</sub>    | 1.0                                                     | W    |
| Operating temperature             | T <sub>opr</sub>  | –40 to +85                                              | °C   |
| Storage temperature               | T <sub>stg</sub>  | –55 to +125                                             | °C   |
| Storage temperature under bias    | T <sub>bias</sub> | –40 to +85                                              | °C   |

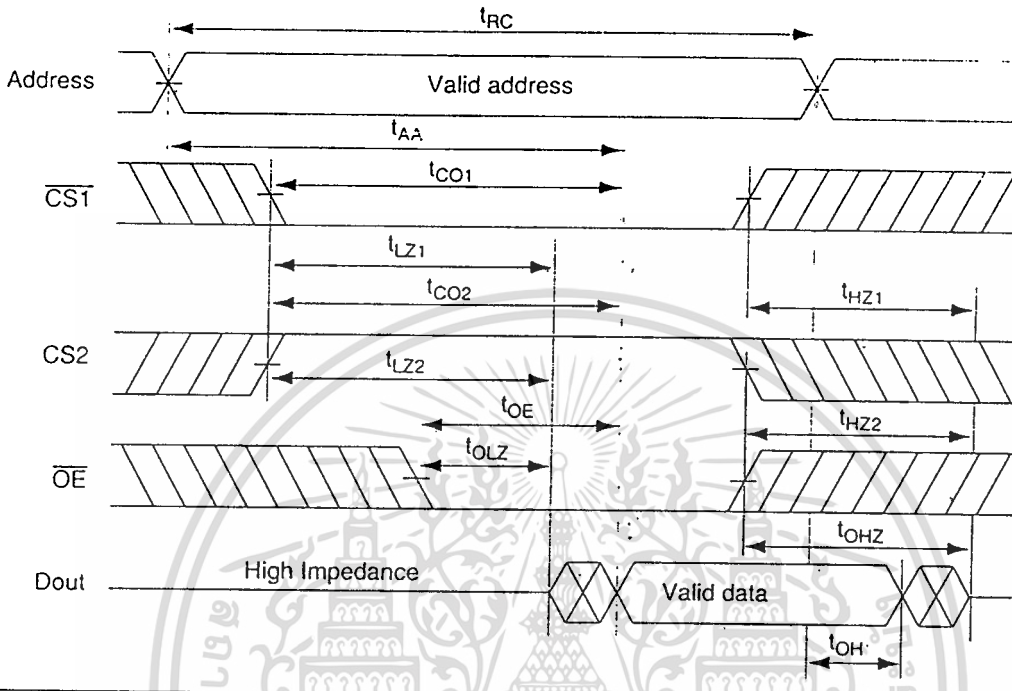
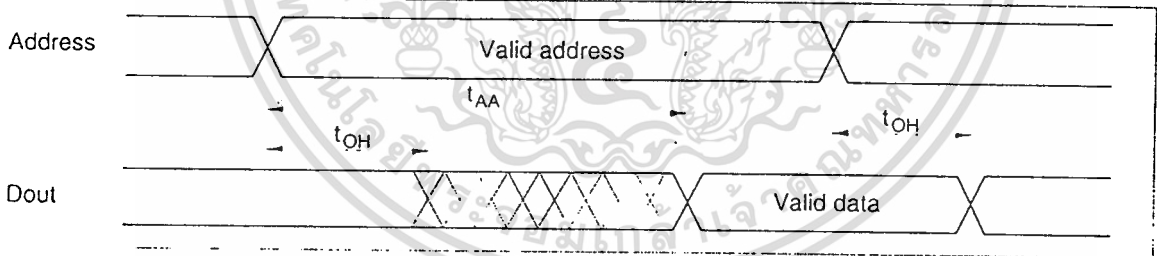
- Notes: 1. Relative to V<sub>SS</sub>  
2. V<sub>T</sub> min: –3.0 V for pulse half-width ≤ 50 ns  
3. Maximum voltage is 7.0 V

Recommended DC Operating Conditions (Ta = –40 to +85°C)

| Parameter          | Symbol          | Min               | Typ | Max                   | Unit |
|--------------------|-----------------|-------------------|-----|-----------------------|------|
| Supply voltage     | V <sub>CC</sub> | 4.5               | 5.0 | 5.5                   | V    |
|                    | V <sub>SS</sub> | 0                 | 0   | 0                     | V    |
| Input high voltage | V <sub>IH</sub> | 2.4               | —   | V <sub>CC</sub> + 0.3 | V    |
| Input low voltage  | V <sub>IL</sub> | –0.3 <sup>1</sup> | —   | 0.6                   | V    |

Note: 1. V<sub>IL</sub> min: –3.0 V for pulse half-width ≤ 50 ns

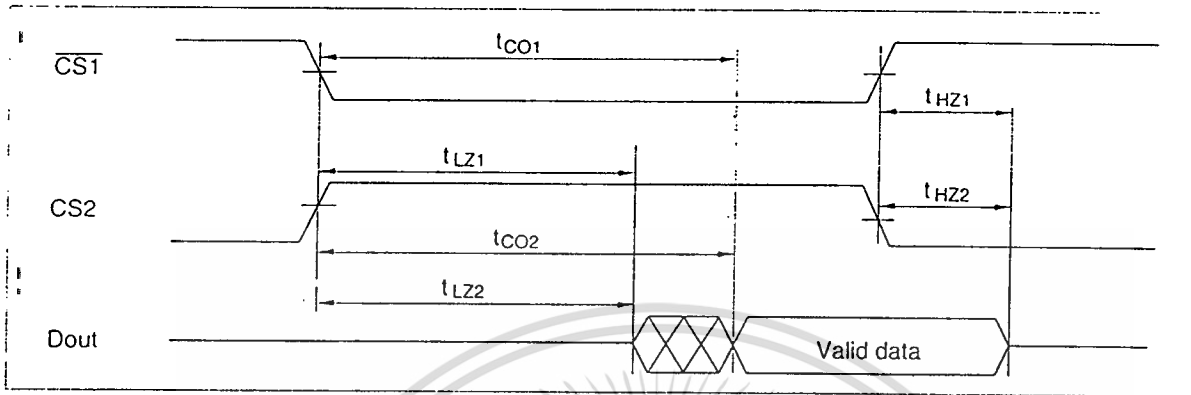
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Read Timing Waveform (1) ( $\overline{WE} = V_{IH}$ )Read Timing Waveform (2) ( $\overline{WE} = V_{IH}$ ,  $\overline{OE} = V_{IL}$ )

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

HM6264BI Series

Read Timing Waveform (3) ( $\overline{WE} = V_{IH}, \overline{OE} = V_{IL}$ )\*3



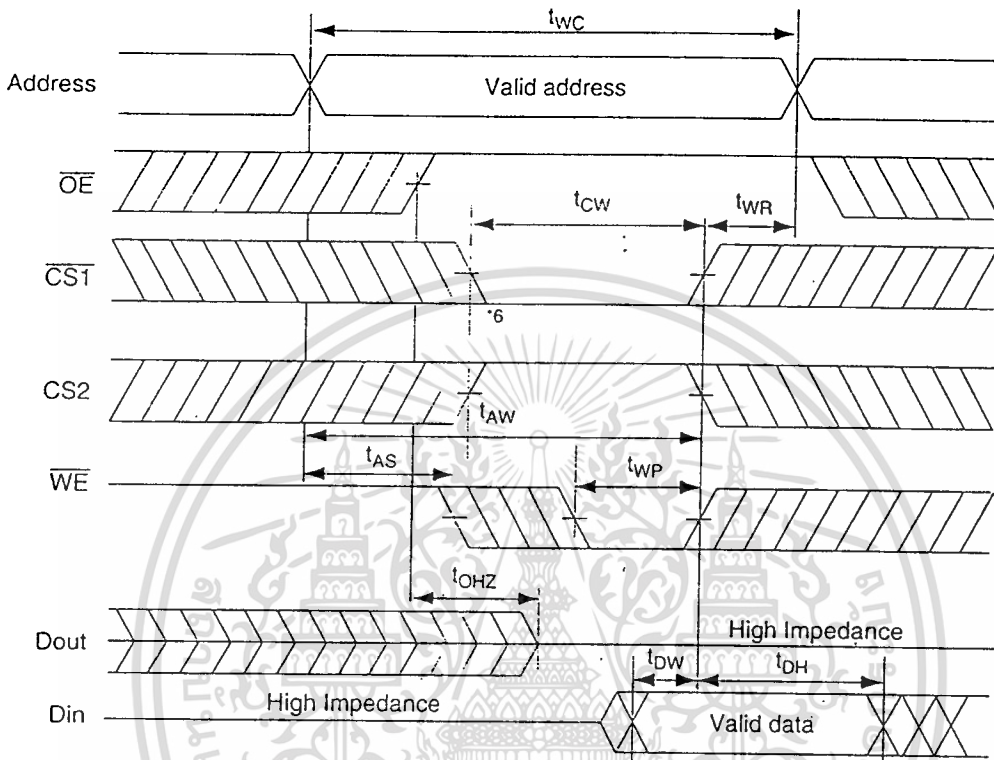
Write Cycle

| Parameter                           | Symbol    | HM6264BI-10 |     | HM6264BI-12 |     | Unit | Notes |
|-------------------------------------|-----------|-------------|-----|-------------|-----|------|-------|
|                                     |           | Min         | Max | Min         | Max |      |       |
| Write cycle time                    | $t_{WC}$  | 100         | —   | 120         | —   | ns   |       |
| Chip selection to end of write      | $t_{CW}$  | 80          | —   | 85          | —   | ns   | 2     |
| Address setup time                  | $t_{AS}$  | 0           | —   | 0           | —   | ns   | 3     |
| Address valid to end of write       | $t_{AW}$  | 80          | —   | 85          | —   | ns   |       |
| Write pulse width                   | $t_{WP}$  | 60          | —   | 70          | —   | ns   | 1, 9  |
| Write recovery time                 | $t_{WR}$  | 0           | —   | 0           | —   | ns   | 4     |
| $\overline{WE}$ to output in high-Z | $t_{WHZ}$ | 0           | 35  | 0           | 40  | ns   | 5     |
| Data to write time overlap          | $t_{DW}$  | 40          | —   | 40          | —   | ns   |       |
| Data hold from write time           | $t_{DH}$  | 0           | —   | 0           | —   | ns   |       |
| Output active from end of write     | $t_{OW}$  | 5           | —   | 5           | —   | ns   |       |
| Output disable to output in high-Z  | $t_{OHZ}$ | 0           | 35  | 0           | 40  | ns   | 5     |

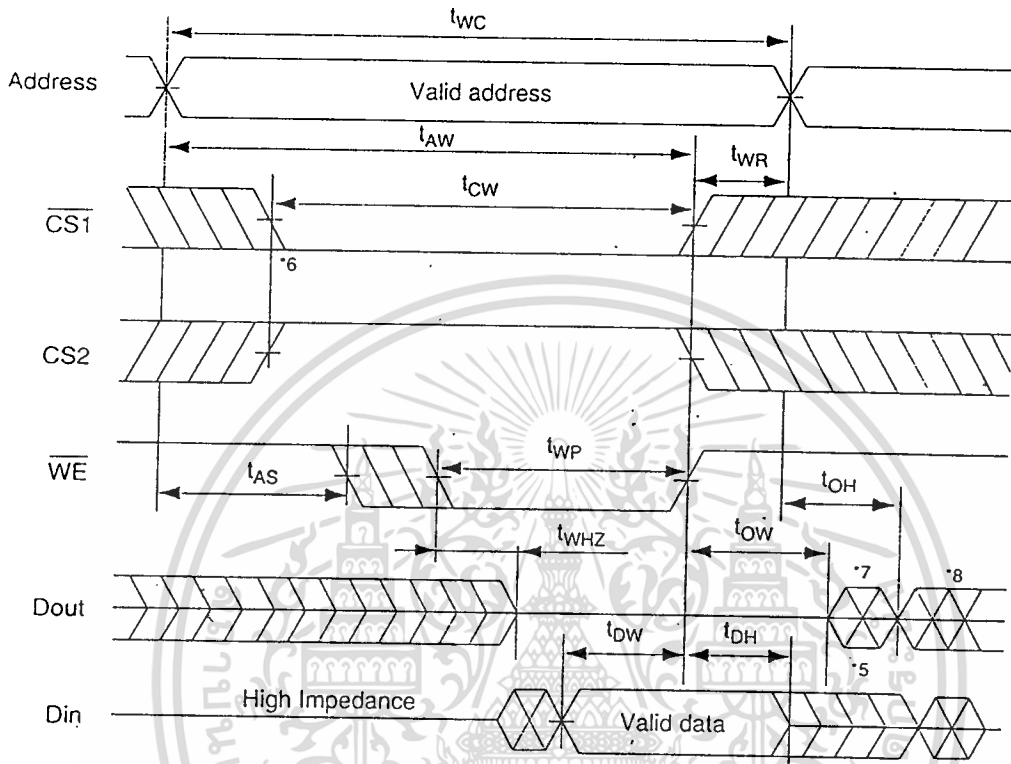
- Notes: 1 A write occurs during the overlap of a low  $\overline{CS1}$ , and high  $\overline{CS2}$ , and a high  $\overline{WE}$ . A write begins at the latest transition among  $\overline{CS1}$  going low,  $\overline{CS2}$  going high and  $\overline{WE}$  going low. A write ends at the earliest transition among  $\overline{CS1}$  going high,  $\overline{CS2}$  going low and  $\overline{WE}$  going high. Time  $t_{WP}$  is measured from the beginning of write to the end of write.
2.  $t_{CW}$  is measured from the later of  $\overline{CS1}$  going low or  $\overline{CS2}$  going high to the end of write.
3.  $t_{AS}$  is measured from the address valid to the beginning of write.
4.  $t_{WR}$  is measured from the earliest of  $\overline{CS1}$  or  $\overline{WE}$  going high or  $\overline{CS2}$  going low to the end of write cycle.
5. During this period, I/O pins are in the output state, therefore the input signals of the opposite phase to the outputs must not be applied.
6. If  $\overline{CS1}$  goes low simultaneously with  $\overline{WE}$  going low after  $\overline{WE}$  goes low, the outputs remain in high impedance state.
7. Dout is the same phase of the written data in this write cycle.
8. Dout is the read data of the next address
9. In the write cycle with  $\overline{OE}$  low fixed,  $t_{WP}$  must satisfy the following equation to avoid a problem of data bus contention

## HM6264BI Series

### Write Timing Waveform (1) ( $\overline{\text{OE}}$ Clock)



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Write Timing Waveform (2) ( $\overline{OE}$  Low Fixed) ( $\overline{OE} = V_{IL}$ )

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

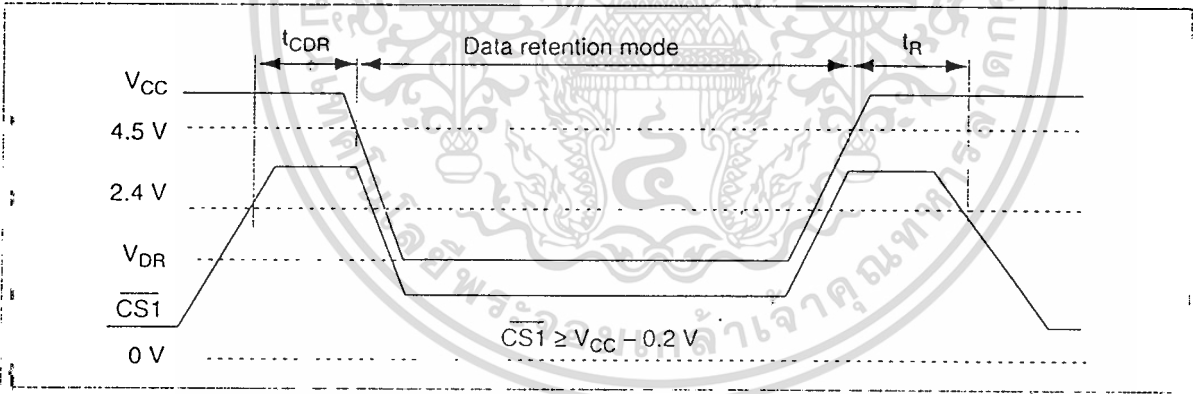
HM6264BI Series

Low V<sub>CC</sub> Data Retention Characteristics (Ta = -40 to +85°C)

| Parameter                            | Symbol            | Min | Typ <sup>*1</sup> | Max               | Unit | Test conditions <sup>*3</sup>                                                                                                                                                                  |
|--------------------------------------|-------------------|-----|-------------------|-------------------|------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| V <sub>CC</sub> for data retention   | V <sub>DR</sub>   | 2.0 | —                 | —                 | V    | $\overline{CS1} \geq V_{CC} - 0.2\text{ V}$ ,<br>$CS2 \geq V_{CC} - 0.2\text{ V}$ or $CS2 \leq 0.2\text{ V}$<br>$V_{in} \geq 0\text{ V}$                                                       |
| Data retention current               | I <sub>CCDR</sub> | —   | 1 <sup>*1</sup>   | 100 <sup>*2</sup> | μA   | $V_{CC} = 3.0\text{ V}$ , $0\text{ V} \leq V_{in} \leq V_{CC}$<br>$\overline{CS1} \geq V_{CC} - 0.2\text{ V}$ , $CS2 \geq V_{CC} - 0.2\text{ V}$<br>or $0\text{ V} \leq CS2 \leq 0.2\text{ V}$ |
| Chip deselect to data retention time | t <sub>CDR</sub>  | 0   | —                 | —                 | ns   | See retention waveform                                                                                                                                                                         |
| Operation recovery time              | t <sub>R</sub>    | 5   | —                 | —                 | ms   |                                                                                                                                                                                                |

- Notes: 1. Reference data at Ta = 25°C.  
2. V<sub>IL</sub> min = -0.3V.  
3. CS2 controls address buffer,  $\overline{WE}$  buffer,  $\overline{CS1}$  buffer,  $\overline{OE}$  buffer, and Din buffer. If CS2 controls data retention mode, Vin levels (address,  $\overline{WE}$ ,  $\overline{OE}$ ,  $\overline{CS1}$ , I/O) can be in the high impedance state. If  $\overline{CS1}$  controls data retention mode, CS2 must be  $CS2 \geq V_{CC} - 0.2\text{ V}$  or  $0\text{ V} \leq CS2 \leq 0.2\text{ V}$ . The other input levels (address,  $\overline{WE}$ ,  $\overline{OE}$ , I/O) can be in the high impedance state.

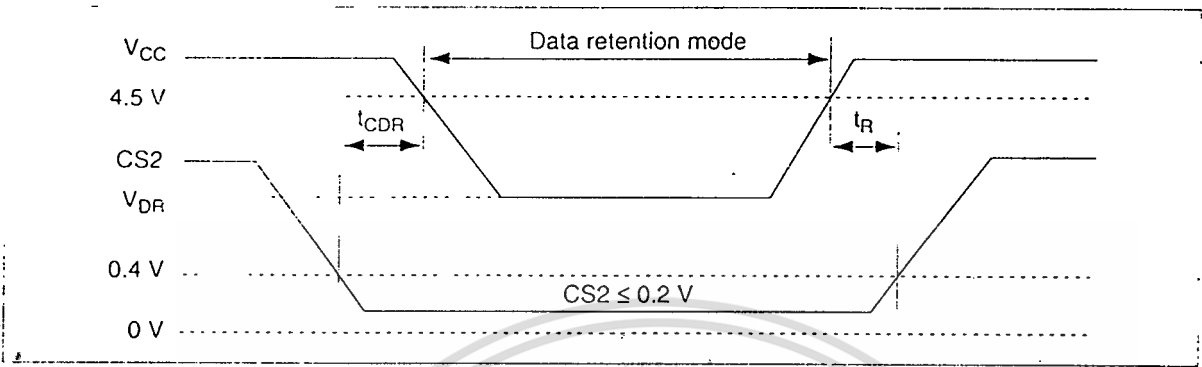
Low V<sub>CC</sub> Data Retention Timing Waveform (1) ( $\overline{CS1}$  Controlled)



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

HM6264BI Series

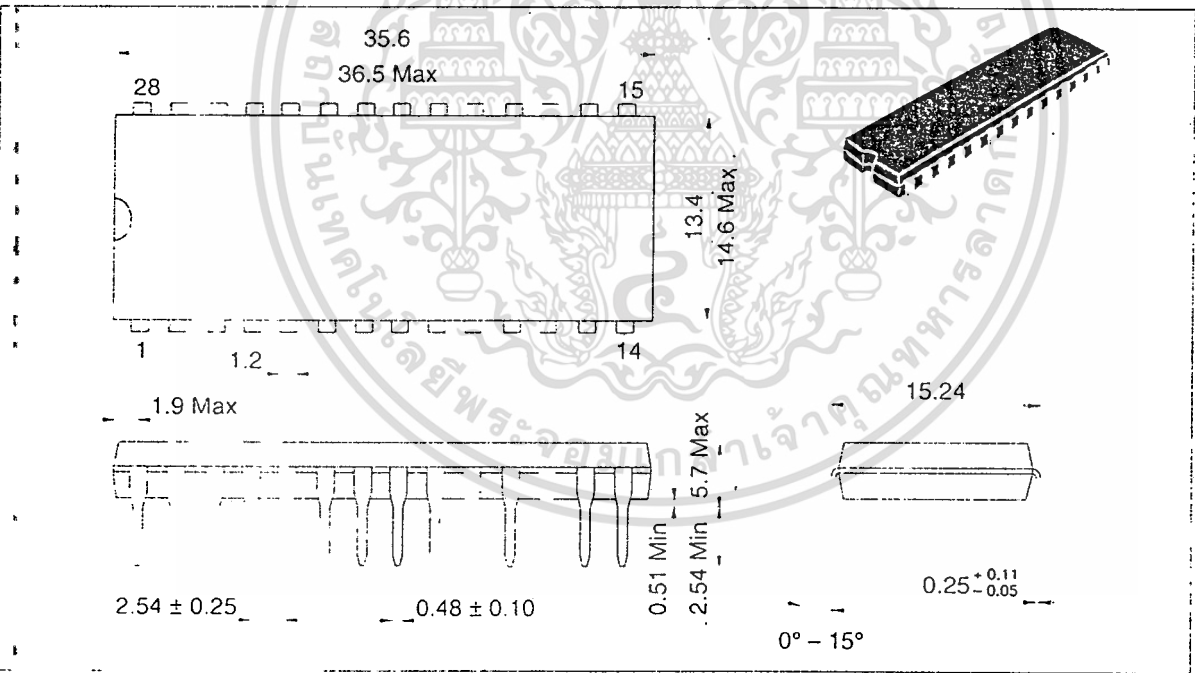
Low V<sub>CC</sub> Data Retention Timing Waveform (2) (CS2 Controlled)



Package Dimensions

HM6264BLFPI Series (DP-28)

Unit: mm

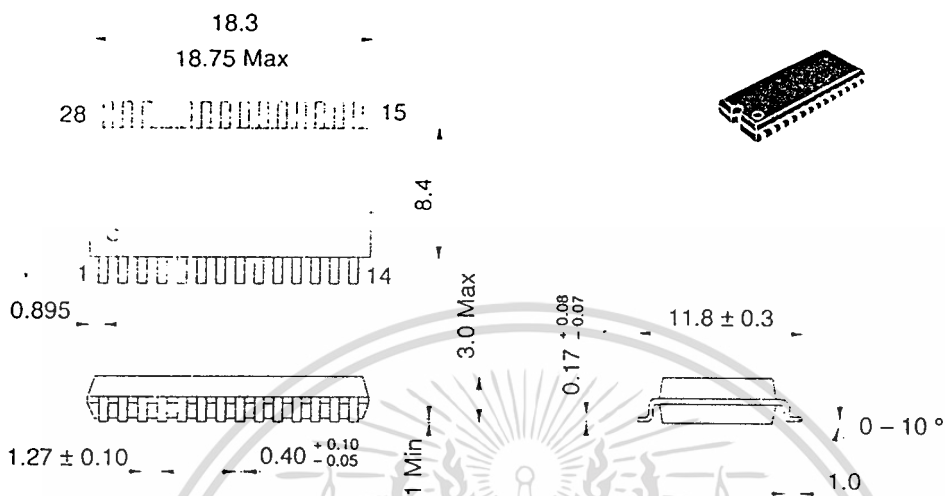


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

HM6264BI Series

HM6264BLPI Series (FP-28DA)

Unit: mm



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# IS80C51 IS80C31

**ISSI**

## CMOS SINGLE CHIP 8-BIT MICROCONTROLLER

NOVEMBER 1998

### FEATURES

- 80C51 based architecture
- 4K x 8 ROM (IS80C51 only)
- 128 x 8 RAM
- Two 16-bit Timer/Counters
- Full duplex serial channel
- Boolean processor
- Four 8-bit I/O ports, 32 I/O lines
- Memory addressing capability
  - 64K ROM and 64K RAM
- Power save modes:
  - Idle and power-down
- Six interrupt sources
- Most instructions execute in 0.3  $\mu$ s
- CMOS and TTL compatible
- Maximum speed: 40 MHz @  $V_{CC} = 5V$
- Industrial temperature available
- Packages available:
  - 40-pin DIP
  - 44-pin PLCC
  - 44-pin PQFP

### GENERAL DESCRIPTION

The ISSI IS80C51 and IS80C31 are high-performance microcontrollers fabricated using high-density CMOS technology. The CMOS IS80C51/31 is functionally compatible with the industry standard 80C51 microcontrollers.

The IS80C51/31 is designed with 4K x 8 ROM (IS80C51 only); 128 x 8 RAM; 32 programmable I/O lines; a serial I/O port for either multiprocessor communications, I/O expansion or full duplex UART; two 16-bit timer/counters; a six-source, two-priority-level, nested interrupt structure; and an on-chip oscillator and clock circuit. The IS80C51/31 can be expanded using standard TTL compatible memory.

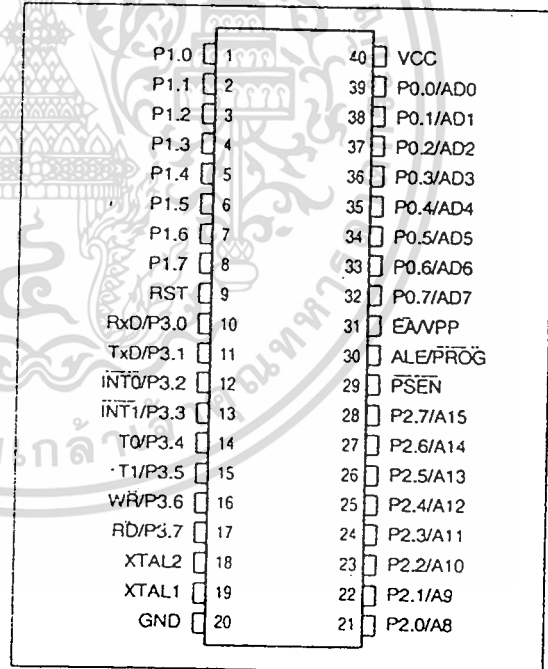


Figure 1. IS80C51/31 Pin Configuration: 40-pin PDIP

ISSI reserves the right to make changes to its products at any time without notice in order to improve design and support the best possible product. We assume no responsibility for any errors which may appear in this publication. © Copyright 1998, Integrated Silicon Solutions, Inc.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



IS80C51  
IS80C31

ISSI

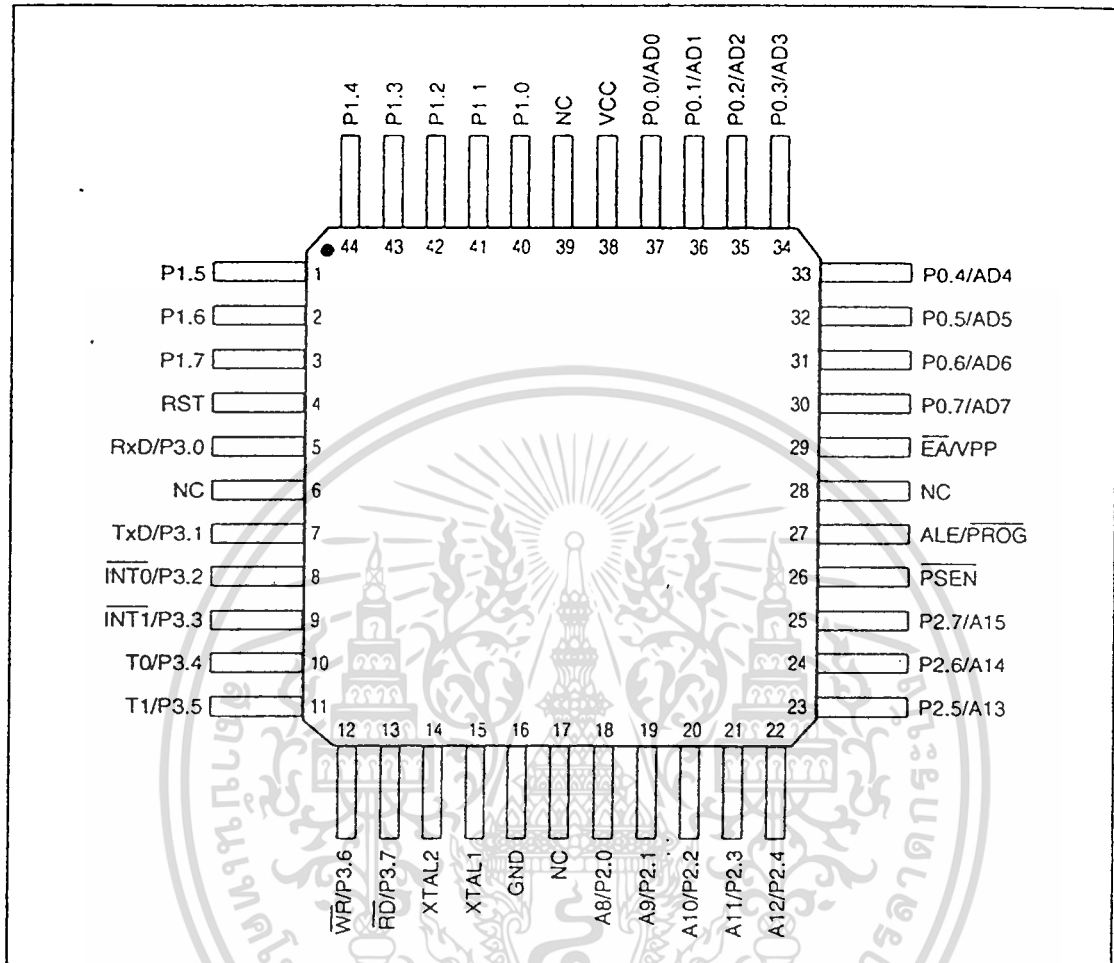


Figure 3. IS80C51/31 Pin Configuration: 44-pin PQFP

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

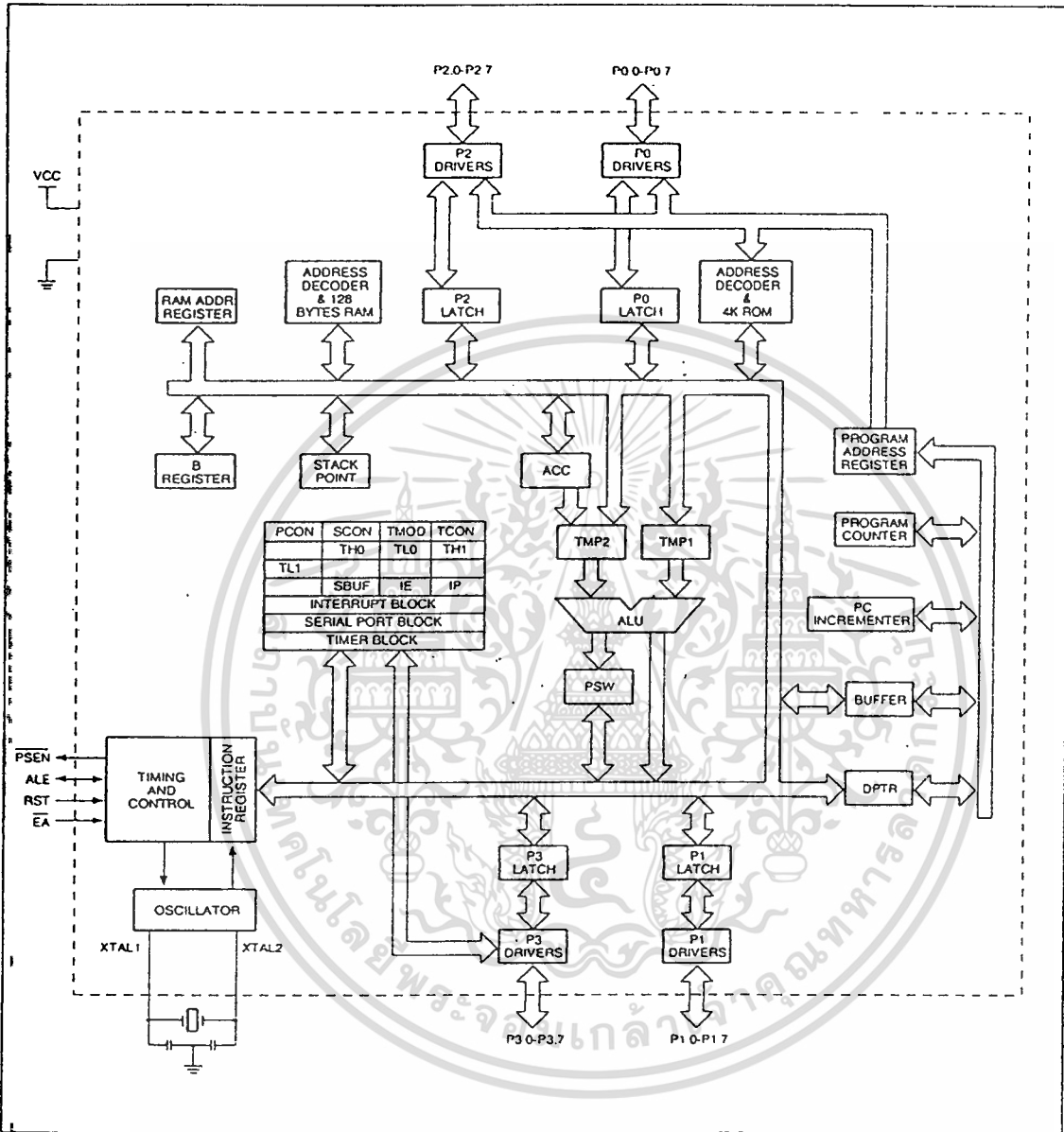
**ISSI**

Figure 4. IS80C51/31 Block Diagram

## TIMER/COUNTERS

The IS80C51/31 has two 16-bit Timer/Counter registers: Timer 0 and Timer 1. Both can be configured to operate either as Timers or event Counters.

As a Timer, the register is incremented every machine cycle. Thus, the register counts machine cycles. Since a machine cycle consists of 12 oscillator periods, the count rate is 1/12 of the oscillator frequency.

As a Counter, the register is incremented in response to a 1-to-0 transition at its corresponding external input pin, T0 and T1. The external input is sampled during S5P2 of every machine cycle. When the samples show a high in one cycle and a low in the next cycle, the count is incremented. The new count value appears in the register during S3P1 of the cycle following the one in which the transition was detected. Since two machine cycles (24 oscillator periods) are required to recognize a 1-to-0 transition, the maximum count rate is 1/24 of the oscillator frequency. There are no restrictions on the duty cycle of the external input signal, but it should be held for at least one full machine cycle to ensure that a given level is sampled at least once before it changes.

In addition to the Timer or Counter functions, Timer 0 and Timer 1 have four operating modes: (13-bit timer, 16-bit timer, 8-bit auto-reload, split timer).

### Timer 0 and Timer 1

Timer/Counters 0 and 1 are present in both the IS80C51/31 and IS80C52/32. The Timer or Counter function is selected by control bits C/T in the Special Function Register TMOD. These two Timer/Counters have four operating modes, which are selected by bit pairs (M1, M0) in TMOD. Modes 0, 1, and 2 are the same for both Timer/Counters, but Mode 3 is different. The four modes are described in the following sections.

#### Mode 0:

Both Timers in Mode 0 are 8-bit Counters with a divide-by-32 prescaler. Figure 8 shows the Mode 0 operation as it applies to Timer 1.

In this mode, the Timer register is configured as a 13-bit register. As the count rolls over from all 1s to all 0s, it sets the Timer interrupt flag TF1. The counted input is enabled to the Timer when TR1 = 1 and either GATE = 0 or INT1 = 1. Setting GATE = 1 allows the Timer to be controlled by external input INT1, to facilitate pulse width measurements. TR1 is a control bit in the Special Function Register TCON. Gate is in TMOD.

The 13-bit register consists of all eight bits of TH1 and the lower five bits of TL1. The upper three bits of TL1 are indeterminate and should be ignored. Setting the run flag (TR1) does not clear the registers.

Mode 0 operation is the same for Timer 0 as for Timer 1, except that TR0, TF0 and INT0 replace the corresponding Timer 1 signals in Figure 7. There are two different GATE bits, one for Timer 1 (TMOD.7) and one for Timer 0 (TMOD.3).

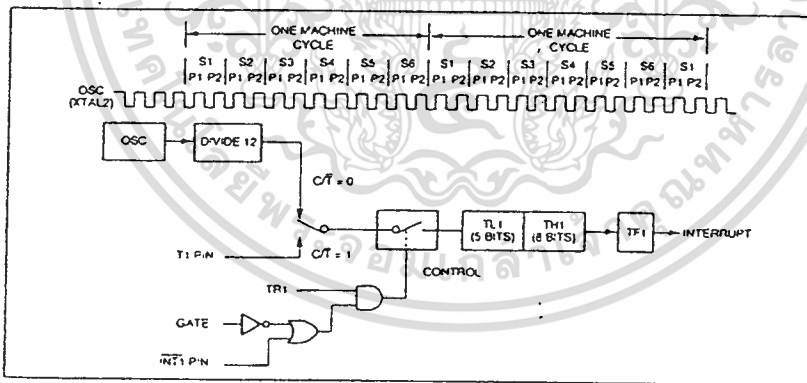
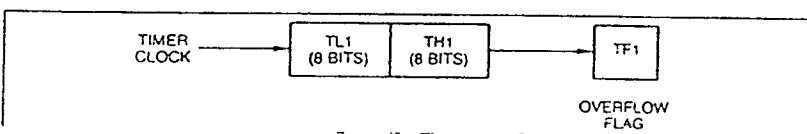


Figure 8. Timer/Counter 1 Mode 0: 13-Bit Counter



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**IS80C51**  
**IS80C31**
**ISSI**
**Mode 1:**

Mode 1 is the same as Mode 0, except that the Timer register is run with all 16 bits. The clock is applied to the combined high and low timer registers (TL1/TH1). As clock pulses are received, the timer counts up: 0000H, 0001H, 0002H, etc. An overflow occurs on the FFFFH-to-0000H overflow flag. The timer continues to count. The overflow flag is the TF1 bit in TCON that is read or written by software (see Figure 9).

**Mode 2:**

Mode 2 configures the Timer register as an 8-bit Counter (TL1) with automatic reload, as shown in Figure 10. Overflow from TL1 not only sets TF1, but also reloads TL1 with the contents of TH1, which is preset by software. The reload leaves the TH1 unchanged. Mode 2 operation is the same for Timer/Counter 0.

**Mode 3:**

Timer 1 in Mode 3 simply holds its count. The effect is the same as setting TR1 = 0. Timer 0 in Mode 3 establishes TL0 and TH0 as two separate Timer/Counters. The logic for Mode 3 on Timer 0 is shown in Figure 11. TL0 uses the Timer 0 control bits: C/T, GATE, TR0,  $\overline{\text{INT0}}$ , and TF0. TH0 is locked into a timer function (counting machine cycles) and over the use of TR1 and TF1 from Timer 1. Thus, TH0 now controls the Timer 1 interrupt.

Mode 3 is for applications requiring an extra 8-bit timer or counter. With Timer 0 in Mode 3, the IS80C51/31 can appear to have three Timer/Counters. When Timer 0 is in Mode 3, Timer 1 can be turned on and off by switching it out of and into its own Mode 3. In this case, Timer 1 can still be used by the serial port as a baud rate generator or in any application not requiring an interrupt.

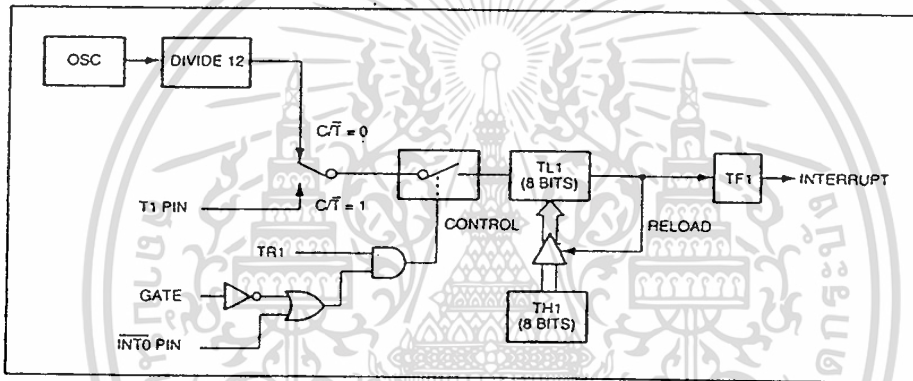


Figure 10. Timer/Counter 1 Mode 2: 8-Bit Auto-Reload

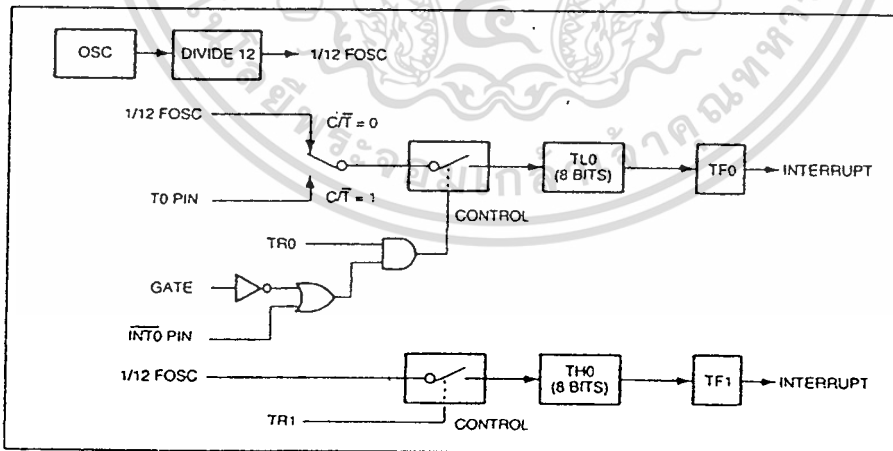


Figure 11. Timer/Counter 0 Mode 3: Two 8-Bit Counters

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

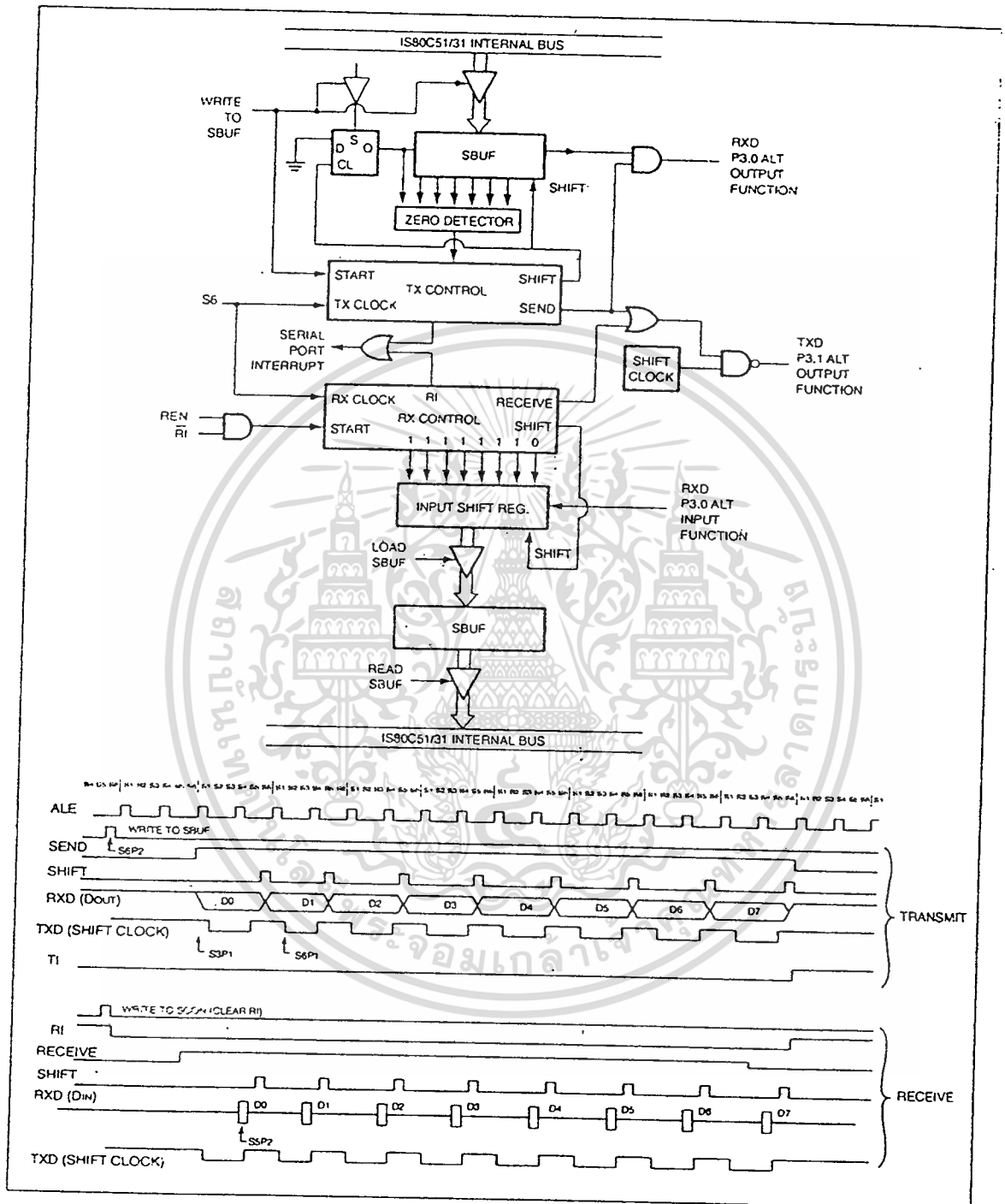


Figure 12. Serial Port Mode 0

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

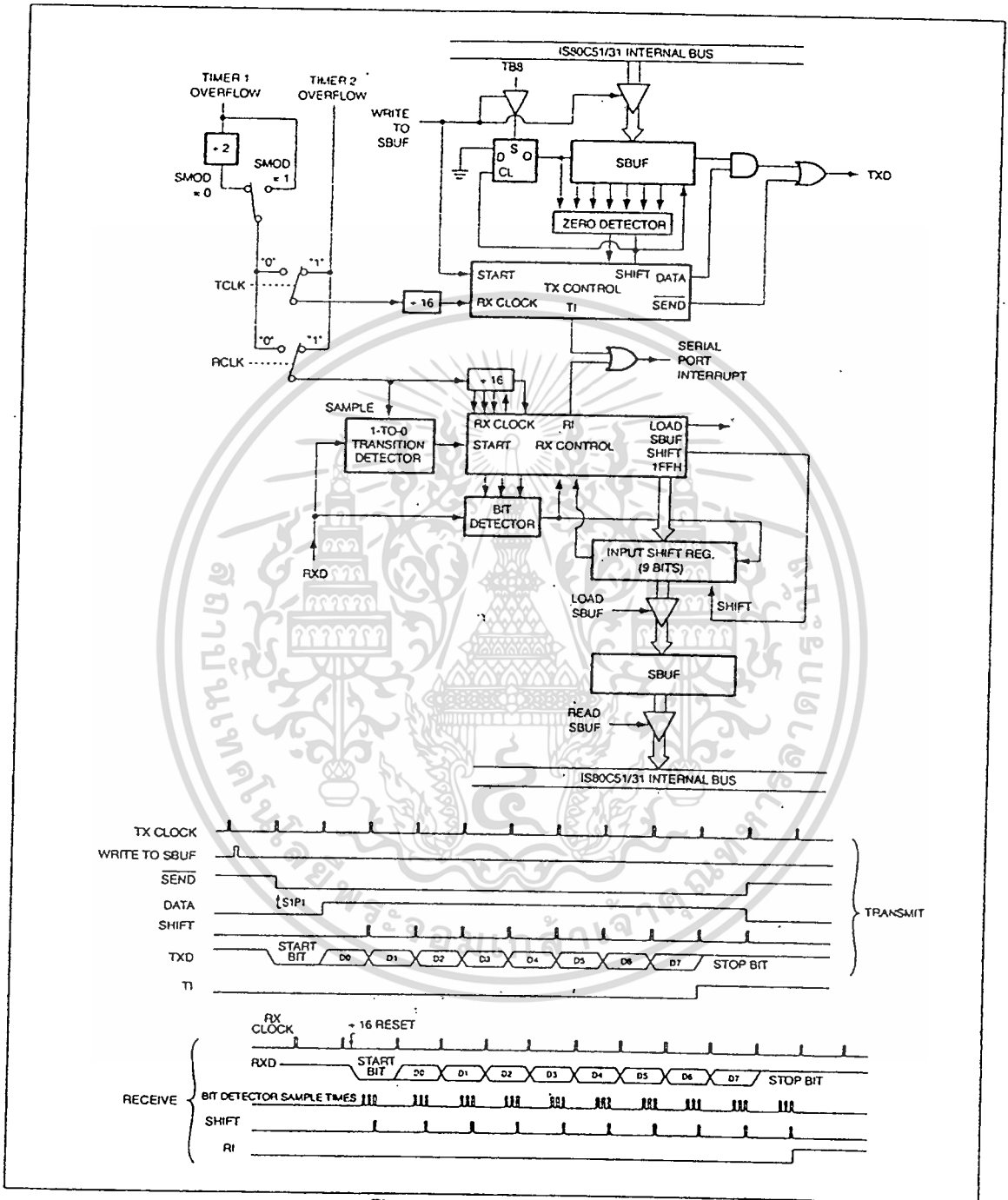


Figure 13. Serial Port Mode 1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

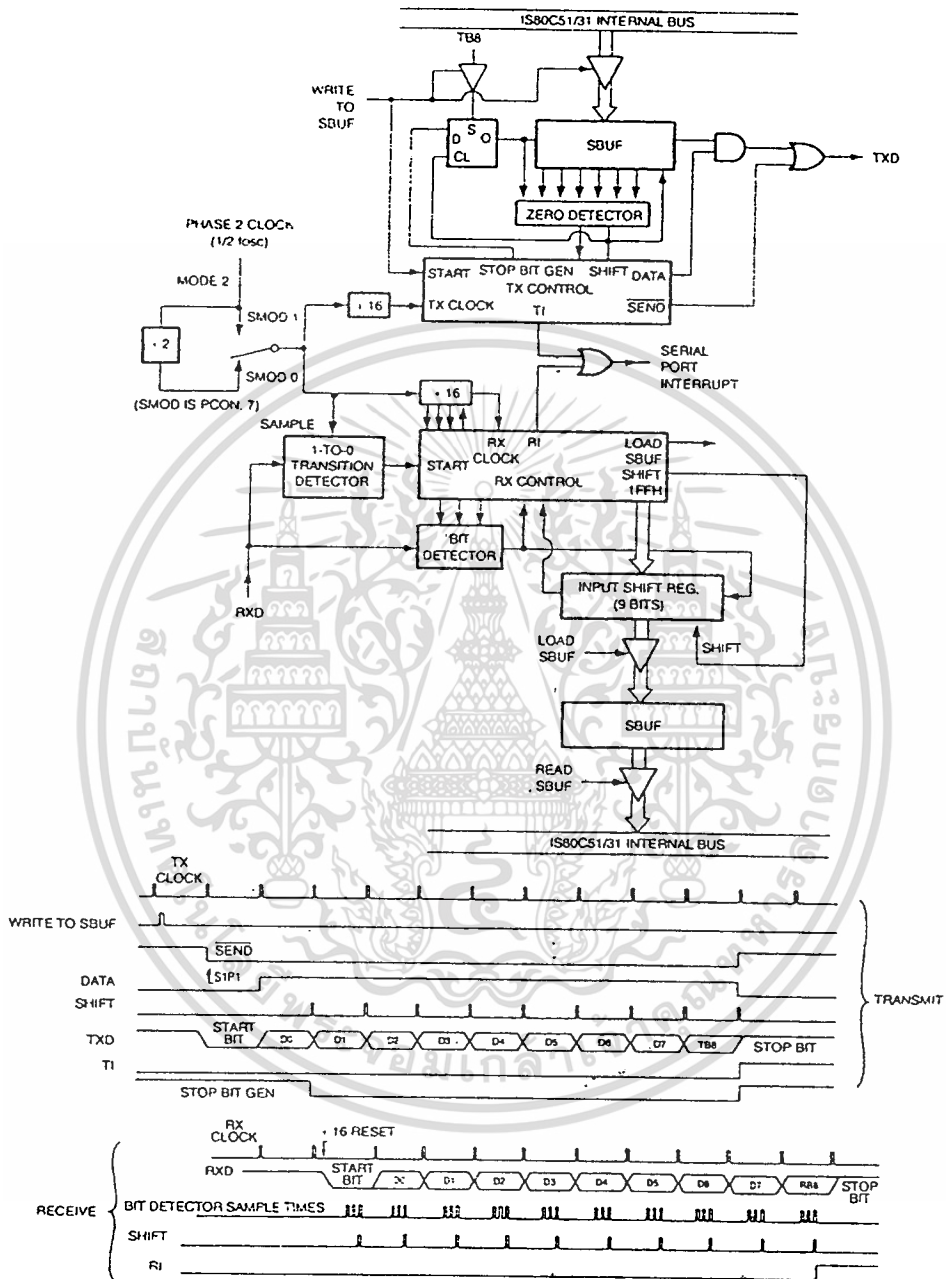


Figure 14. Serial Port Mode 2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

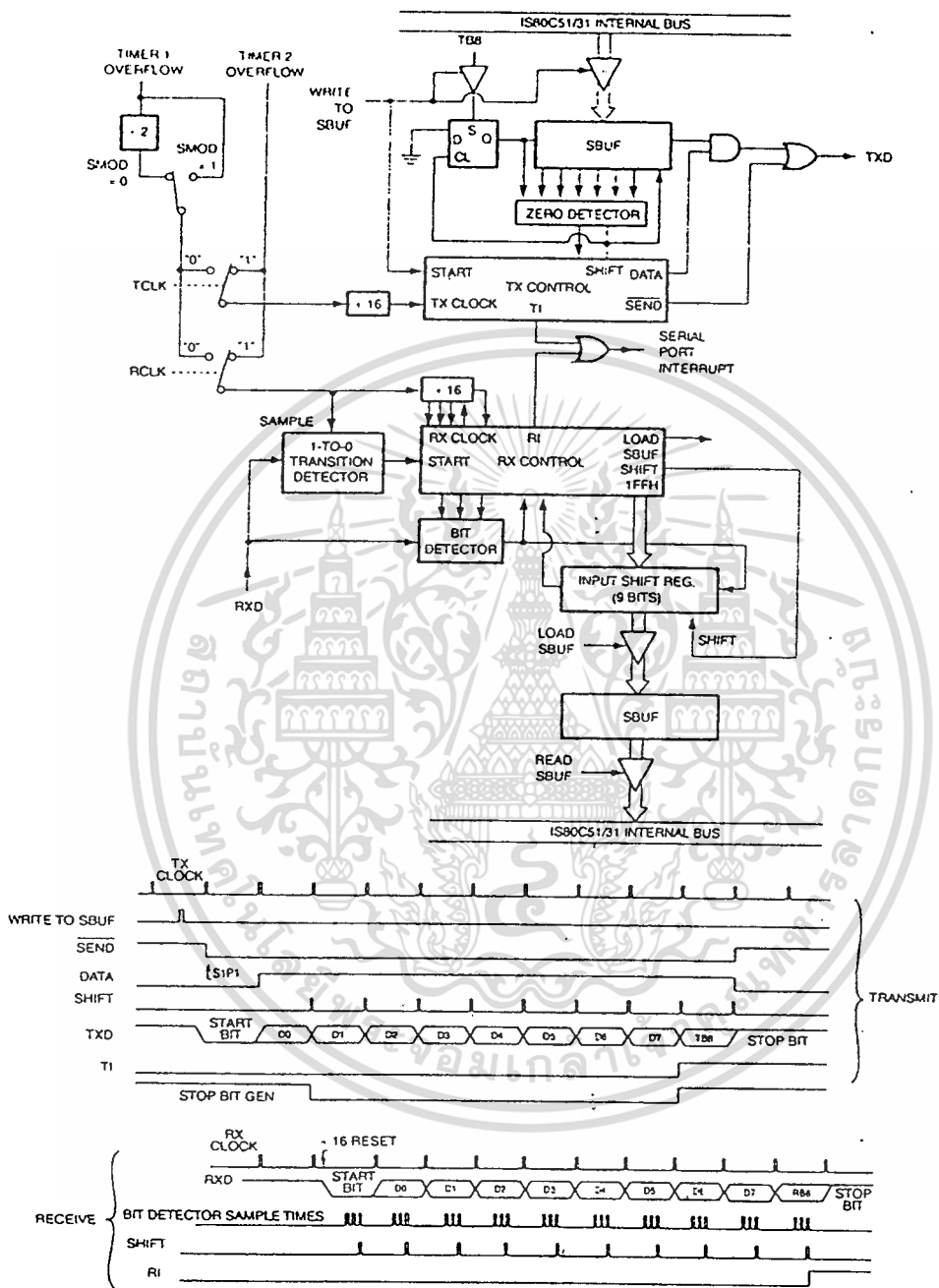


Figure 15. Serial Port Mode 3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## INTERRUPT SYSTEM

The IS80C51/31 provides 6 interrupt sources: two external interrupts, three timer interrupts, and a serial port interrupt. These are shown in Figure 16.

The External Interrupts  $\overline{INT0}$  and  $\overline{INT1}$  can each be either level-activated or transition-activated, depending on bits IT0 and IT1 in Register TCON. The flags that actually generate these interrupts are the IE0 and IE1 bits in TCON. When the service routine is vectored to, hardware clears the flag that generated an external interrupt only if the interrupt was transition-activated. If the interrupt was level-activated, then the external requesting source (rather than the on-chip hardware) controls the request flag.

The Timer 0 and Timer 1 Interrupts are generated by TF0 and TF1, which are set by a rollover in their respective Timer/Counter registers (except for Timer 0 in Mode 3). When a timer interrupt is generated, the on-chip hardware clears the flag that generated it when the service routine is vectored to.

The Serial Port Interrupt is generated by the logical OR of RI and TI. Neither of these flags is cleared by hardware when the service routine is vectored to. In fact, the service routine normally must determine whether RI or TI generated the interrupt, and the bit must be cleared in software.

All of the bits that generate interrupts can be set or cleared by software, with the same result as though they had been set or cleared by hardware. That is, interrupts can be generated and pending interrupts can be canceled in software.

Each of these interrupt sources can be individually enabled or disabled by setting or clearing a bit in Special Function Register IE (interrupt enable) at address 0A8H. As well as individual enable bits for each interrupt source, there is a global enable/disable bit that is cleared to disable all interrupts or set to turn on interrupts (see SFR IE).

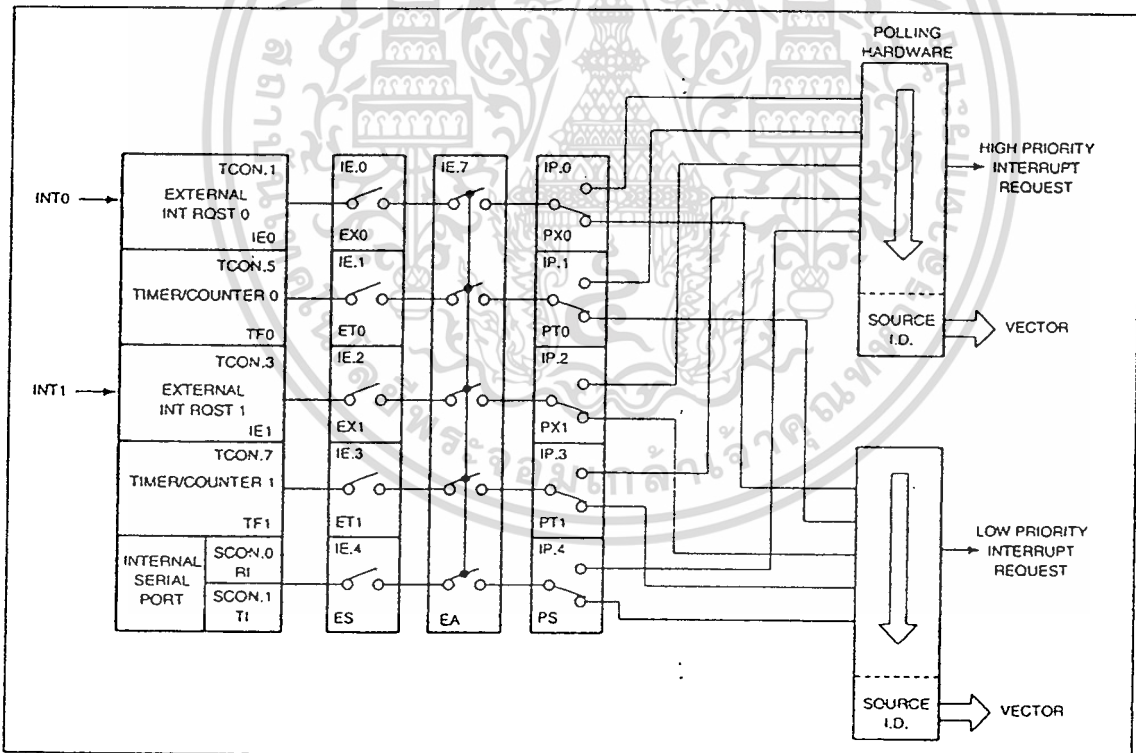


Figure 16. Interrupt System

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## OTHER INFORMATION

## Reset

The reset input is the RST pin, which is the input to a Schmitt Trigger.

A reset is accomplished by holding the RST pin high for at least two machine cycles (24 oscillator periods), while the oscillator is running. The CPU responds by generating an internal reset, with the timing shown in Figure 18.

The external reset signal is asynchronous to the internal clock. The RST pin is sampled during State 5 Phase 2 of every machine cycle. The port pins will maintain their current activities for 19 oscillator periods after a logic 1 has been sampled at the RST pin; that is, for 19 to 31 oscillator periods after the external reset signal has been applied to the RST pin.

The internal reset algorithm writes 0s to all the SFRs except the port latches, the Stack Pointer, and SBUF. The port latches are initialized to FFH, the Stack Pointer to 07H, and SBUF is indeterminate. Table 9 lists the SFRs and their reset values.

Then internal RAM is not affected by reset. On power-up the RAM content is indeterminate.

Table 9. Reset Values of the SFR's

| SFR Name | Reset Value   |
|----------|---------------|
| PC       | 0000H         |
| ACC      | 00H           |
| B        | 00H           |
| PSW      | 00H           |
| SP       | 07H           |
| DPTR     | 0000H         |
| P0-P3    | FFH           |
| IP       | XXX00000B     |
| IE       | 0XX00000B     |
| TMOD     | 00H           |
| TCON     | 00H           |
| TH0      | 00H           |
| TL0      | 00H           |
| TH1      | 00H           |
| TL1      | 00H           |
| SCON     | 00H           |
| SBUF     | Indeterminate |
| PCON     | 0XXX0000B     |

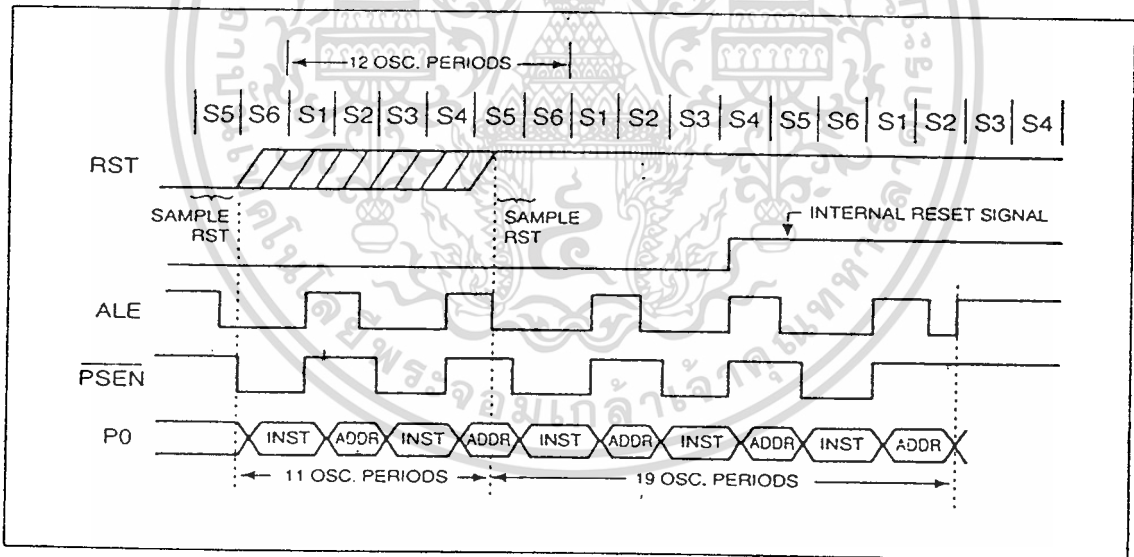


Figure 18. Reset Timing

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### ROM Verification

The on-chip memory can be read out for ROM verification. The address of the program memory location to be read is applied to Port 1 and pins P2.3-P2.0. The other pins should be held at the "Verify" level. The contents of the addressed locations will be emitted on Port 0. External pullups are required on Port 0 for this operation. Figure 24 shows the setup to verify the program memory.

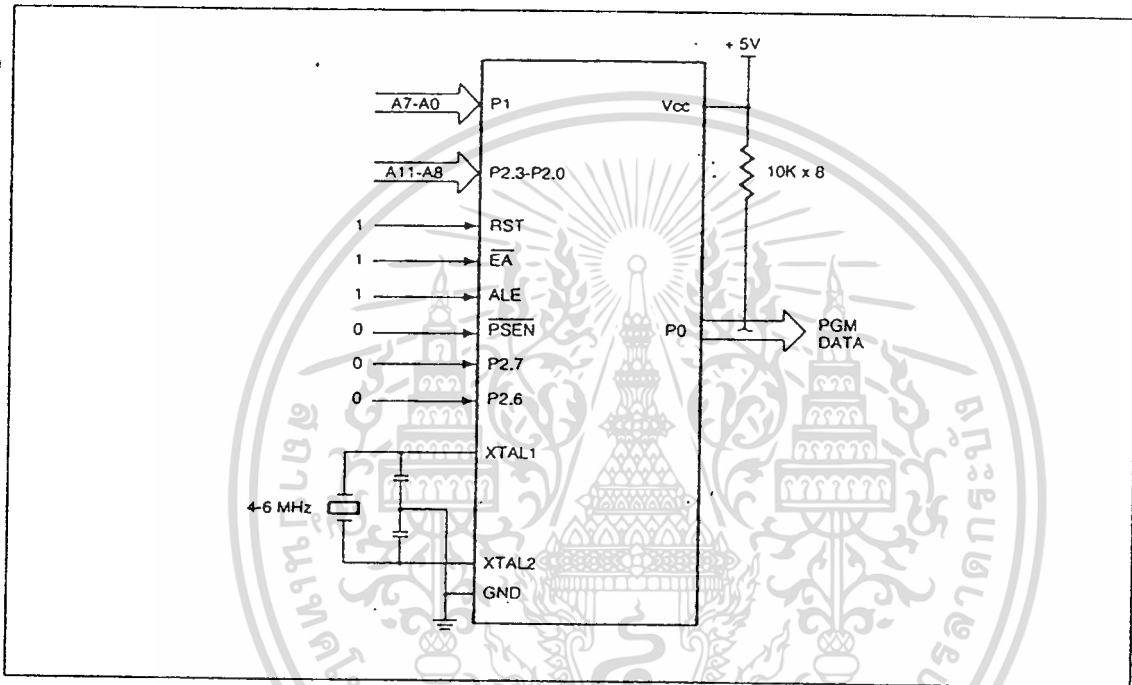


Figure 24. ROM Verification

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

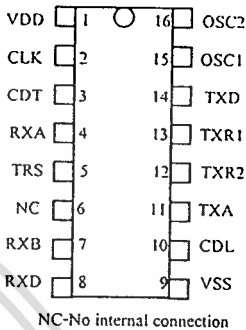
TCM3105JE, TCM3105JL

FSK MODEM

D2662. NOVEMBER 1985 - REVISED APRIL

1986

J DUAL-IN-LINE PACKAGE:  
(TOP VIEW)



- Single-Chip Frequency-Shift-Keying (FSK) Modem
- Meets Both Bell 202 and CCITT V23 Specifications
- Transmit Modulation at 75, 150, 600, and 1200 Baud
- Receive Demodulation at 5, 75, 150, 600, and 1200 Baud
- Half-Duplex Operation Up to 1200 Baud Transmit and Receive
- Full-Duplex Operation Up to 1200 Baud Transmit and 150 Baud Receive
- On-Chip Ground Delay Equalization and Transmit/Receive Filtering
- Carrier-Detect-Level Adjustment and Carrier-Fail Output
- Single 5-V Power Supply
- Low Power Consumption
- Reliable CMOS Silicon Gate Technology

PIN FUNCTIONAL DESCRIPTION

| NO | NAME | DESCRIPTION                                                                                                                                                                                                                                                 |
|----|------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1  | VDD  | Positive supply voltage                                                                                                                                                                                                                                     |
| 2  | CLK  | Output for a continuous clock signal at 16 times the highest selected Transmit or receive bit rate                                                                                                                                                          |
| 3  | CDT  | Carrier-Detect Output A low-level output indicates carrier failure                                                                                                                                                                                          |
| 4  | RXA  | Receive Analog Input to which the receive line signal must be ac coupled                                                                                                                                                                                    |
| 5  | TRS  | Transmit/Receive Standard Select Input. Which, With TXR1 and TXR2 sets the standard bit rates and mark/space frequencies                                                                                                                                    |
| 6  | NC   | No internal connection                                                                                                                                                                                                                                      |
| 7  | RXB  | Receive Bias Adjust for external adjustment of the decision threshold of the final comparator to minimize bias distortion                                                                                                                                   |
| 8  | RXD  | Receive Digital Output for the demodulated received data in positive logic. The high logic level is a mark and the low logic level is a space                                                                                                               |
| 9  | VSS  | Most negative supply voltage (normally ground); connected to substrate                                                                                                                                                                                      |
| 10 | CDL  | Carrier Detect Level Adjust for external adjustment of carrier detect threshold                                                                                                                                                                             |
| 11 | TXA  | Transmit Analog Output for the modulated signal. Which must be ac coupled                                                                                                                                                                                   |
| 12 | TXR2 | Bit Rate Select 2 input which, Along with TXR1 and TRS. Sets the bit rates and mark/space frequencies                                                                                                                                                       |
| 13 | TXR3 | Bit Rate Select 1 input which, Along with TXR2 and TRS. Sets the bit rates and mark/space frequencies                                                                                                                                                       |
| 14 | TXD  | Transmit Digital Input for input data to the transmitter in positive logic logic. The high logic level is a mark and the low logic level is a space. The data can be accepted at any speed from zero to the selected speed and may be totally asynchronous. |
| 15 | OSC1 | Oscillator connections. The crystal (typically 4.4336 MHz) is connected to these. If an external clock is used, OSC2 is left open and the clock is connected to OSC1.                                                                                       |
| 16 | OSC2 | Oscillator connections. The crystal (typically 4.4336 MHz) is connected to these. If an external clock is used, OSC2 is left open and the clock is connected to OSC1.                                                                                       |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลง INSTRUMENTS อ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

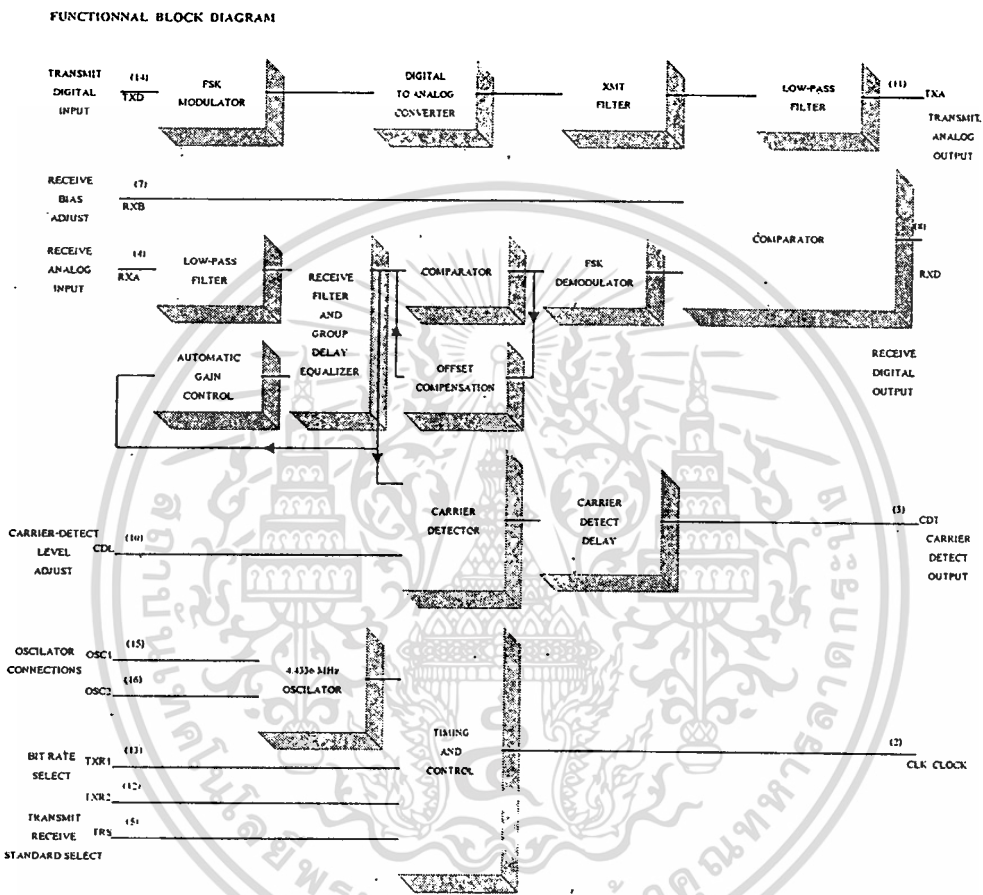
TEXAS

TCM3105JE, TCM3105JL

FSK MODEM

D2662, NOVEMBER 1985 - REVISED APRIL

1986



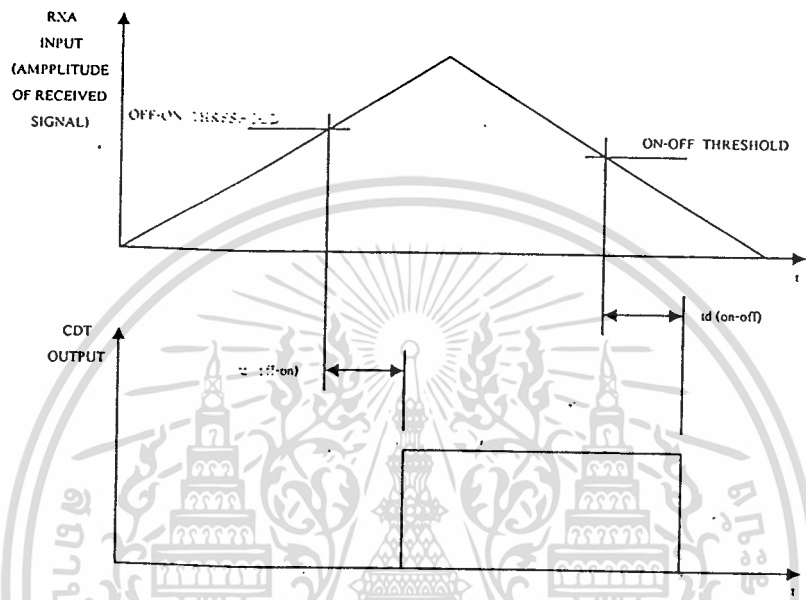
recommended operating condition

|                                         | TCM3105JE |        |        | TCM3105JL |        |        | UNIT |
|-----------------------------------------|-----------|--------|--------|-----------|--------|--------|------|
|                                         | MIN       | NOM    | MAX    | MIN       | NOM    | MAX    |      |
| Supply voltage. VDD                     | 4         | 5      | 5.5    | 4         | 5      | 5.5    | V    |
| High-level input voltage. VIH           | 2         |        | VDD    | 2         |        | VDD    | V    |
| Low-level input voltage. VIL            | 0         |        | 0.8    | 0         |        | 0.8    | V    |
| Analog input level. Peak (ac coupled)   |           | 0.3    | 0.78   |           | 0.30   | 0.78   | V    |
| Clock frequency. Clock                  | 4.4334    | 4.4336 | 4.4338 | 4.4334    | 4.4336 | 4.4338 | MHz  |
| Analog load impedance at TXA            | 50        |        |        | 50        |        |        | kH   |
| Operating free-air temperature rang. TA | -40       |        | 85     | 0         |        | 70     | °C   |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานที่ TEXAS INSTRUMENTS ขอสงวนสิทธิ์ในไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1986

TIMING DIAGRAM



electrical characteristics over recommended ranges of supply voltage and operating free-air temperature (unless otherwise noted)

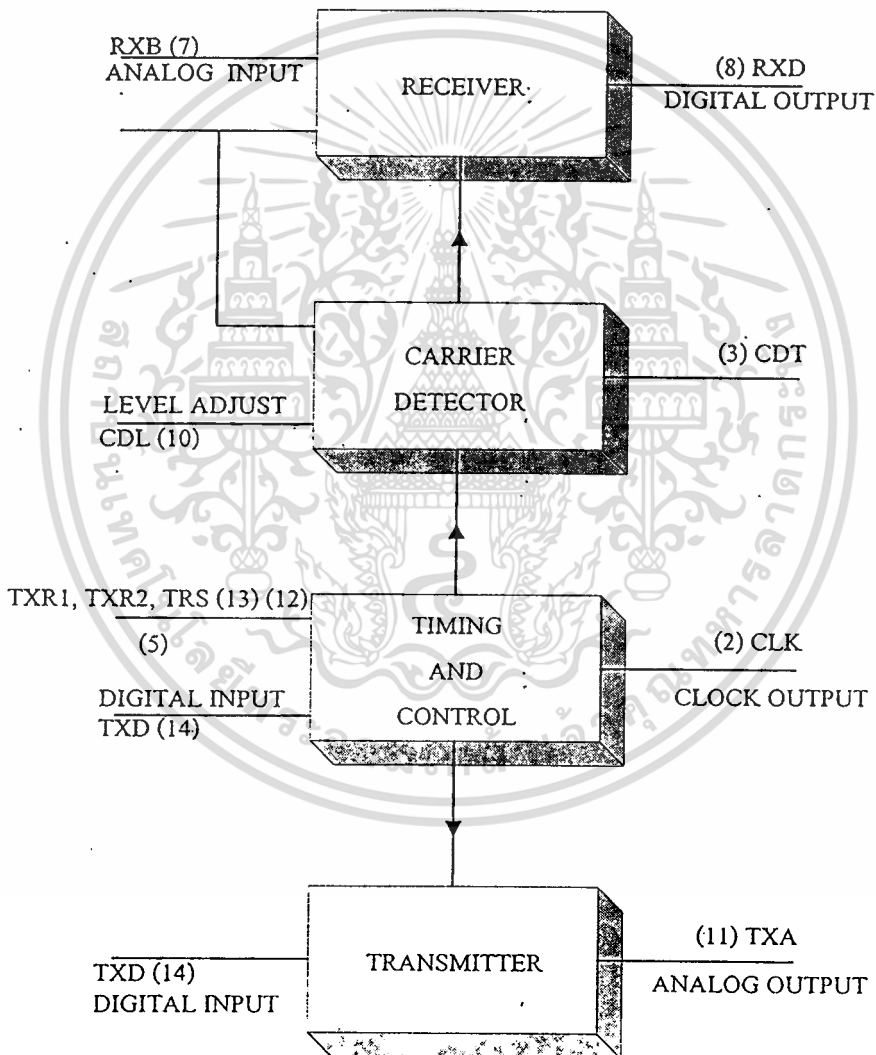
| PARAMETER                                |                                | TEST CONDITION               | TCM3105JE          |     |       | TCM3105JL  |     |       | UNIT    |
|------------------------------------------|--------------------------------|------------------------------|--------------------|-----|-------|------------|-----|-------|---------|
|                                          |                                |                              | MIN                | TYP | MAX   | MIN        | TYP | MAX   |         |
| VOH                                      | High level output voltage      | RXP, CDT, TXR                | IOH = -100 $\mu$ S |     |       | 2.4        |     | VDC   | V       |
| VOL                                      | Low level output voltage       | RXP, CDT, TXR                | IOL = 1.6 mA       |     |       | VSS        |     | 0.4   | V       |
| Analog output voltage level peak to peak | TXA                            | VDD = 4 V                    | 1.55               |     |       | 1.55       |     |       |         |
|                                          |                                | VDD = 5 V RL = 50 k $\Omega$ | 1.4                | 1.9 | 2.3   | 1.4        | 1.9 | 2.3   | V       |
|                                          |                                | VDD = 5.5 V CL = 100 nF      | 2.1                |     |       | 2.1        |     |       |         |
| Adjust voltage                           | RXB                            | VDD = 5 V                    | 2.3                | 2.7 | 3.1   | 2.3        | 2.7 | 3.1   | V       |
|                                          | CTM                            |                              | 2.8                | 3.3 | 3.9   | 2.8        | 3.3 | 3.9   |         |
| Analog output dc offset                  | TXA                            |                              | VDD/2              |     |       | VDD/2      |     |       | V       |
| Digital input current                    | TXD, TSS, TXR1, TXR2           | VI = 0 TO VDD                | $\pm 1$            |     |       | $\pm 1$    |     |       | $\mu$ A |
| Analog input current                     | RXA                            |                              | $\pm 15$           |     |       | $\pm 15$   |     |       | $\mu$ A |
| Bias input current                       | RXB, CTM                       | VI = 3 V                     | $\pm 150$          |     |       | $\pm 150$  |     |       | $\mu$ A |
| IDD                                      | Supply current                 | VDD = 4 V                    | 3                  |     | 6     | 3          |     | 6     |         |
|                                          |                                | VDD = 5 V                    | 5                  |     | 10    | 5          |     | 10    | mA      |
|                                          |                                | VDD = 5.5 V                  | 8                  |     | 16    | 8          |     | 16    |         |
| C1                                       | Input capacitance, all inputs  | f = 1 MHz                    | 10                 |     |       | 10         |     |       | pF      |
| C0                                       | Output capacitance, all inputs | f = 1 MHz                    | 10                 |     |       | 10         |     |       | pF      |
| Phase filter                             |                                |                              | 200                |     |       | 200        |     |       | $\mu$ S |
| Bias distortion                          |                                |                              | $\pm 15\%$         |     |       | $\pm 15\%$ |     |       |         |
| Carrier detect threshold, off-on         |                                |                              | -45.5              |     | -43   | -45.5      |     | -43   | dBm     |
| Carrier detect threshold, on-off         |                                |                              | -48                |     | -45.5 | -48        |     | -45.5 | dBm     |
| Carrier detect hysteresis                |                                |                              | 2.5                | 2.8 |       | 2.5        | 2.8 |       | dBm     |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้เผยแพร่โดยไม่ได้รับอนุญาต  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

TEXAS  
INSTRUMENTS

## PRINCIPLES OF OPERATION

The TCM3105 FSK modem is made up of four functional circuits. The circuits are the transmitter. The receiver a carrier detection and control and timing (See Figure 1)



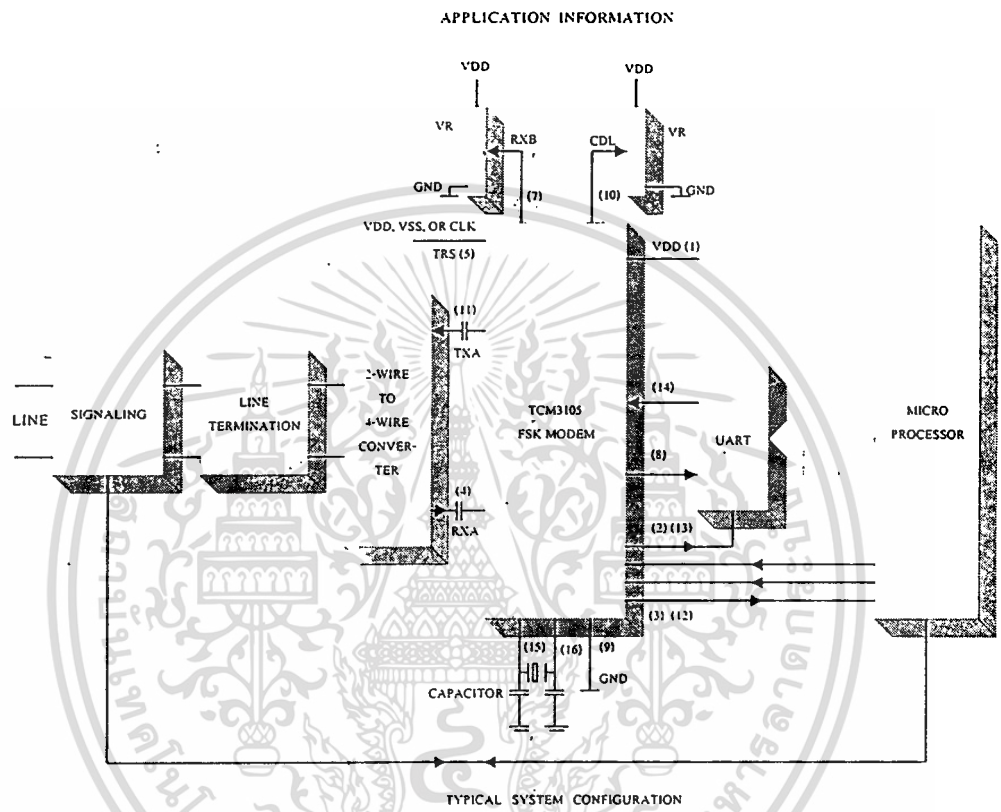
TCM3105 SYSTEM  
PARTITIONING

TCM3105JE, TCM3105JL

FSK MODEM

D2662. NOVEMBER 1985 - REVISED APRIL

1986



TCM3105JE, TCM3105JL

FSK MODEM

D2662. NOVEMBER 1985 - REVISED APRIL

1986

The TCM3105

The TCM3105 provides a majority of the functions required of a medium speed FSK. Modem in a single 16 - pin DIP. The device is manufactured using silicon-gate complementary consumption of approximately 40 mW. This makes the device ideally suited for use in battery operated equipment applications. As well as in standard applications. The TCM3105 device significance of each pin.

The TCM3105 is characterized for operation from 0°c to 70°c (JL suffix) as well as over the extended free-air temperature range of -40°c to +85°c (JE suffix).

Table 1. Pinout Description

| PIN  |    | I/O | DESCRIPTION                                                                                                                                                                                                                                                          |
|------|----|-----|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| NAME | NO |     |                                                                                                                                                                                                                                                                      |
| CDL  | 10 | I   | Carrier Detect Level-Sets the threshold level for the carrier detect decision Refer to Description of the Carrier Detect Adjustment paragraph                                                                                                                        |
| CDT  | 3  | O   | Carrier Detect - A high logic level output indicates the presence of a carrier at the RXA pin                                                                                                                                                                        |
| CLK  | 2  | O   | CLOCK - Continuous output clock signal at 16 times the highest selected transmit or receive baud rate                                                                                                                                                                |
| OSC1 | 15 | I   | Oscillator 1 and 2 - input connections for external 4.4336 MHz crystal. See Table2 for list of crystal manufacturers.                                                                                                                                                |
| OSC2 | 16 |     | If an external clock input is provided. Then the OSC1 pin is left open and the clock is connected to the OSC2 pin.                                                                                                                                                   |
| RXA  | 4  | I   | Receive Analog - This input is referenced to an internal voltage and must be ac coupled                                                                                                                                                                              |
| RXB  | 7  | I   | Receive Bias Adjust - This input sets the threshold level of the slicer that allows the bias distortion on the RXD pin to be minimized. Refer to Description of the Receive Bias Adjustment paragraph.                                                               |
| RXD  | 8  | O   | Receive Digital Output - Outputs the demodulated receive data in positive logic. i.e., a mark is indicated by a high level and a space is indicated by a low level. The RXD output pin will remain high if there is no analog input on the RXA pin.                  |
| TRS  | 5  | I   | Transmit/Receive Standard Select Input - This pin along with TXR1 and TXR2 select the standard and mode to be used. See Table 1.                                                                                                                                     |
| TXD  | 14 | I   | Transmit Digital - Digital input to the modulator in positive logic, i.e., a mark is indicated by a high level and a space is indicated by a low level. The data can be accepted at any rate from zero up to the selected baud rate and may be totally asynchronous. |
| TXR1 | 13 | I   | Transmit Rate 1 and 2 - These signals along with TRS set the standard and mode to be used. See Table 1.                                                                                                                                                              |
| TXR2 | 12 | I   |                                                                                                                                                                                                                                                                      |
| VDD  | 1  |     | Positive supply voltage - 5 volts nominal                                                                                                                                                                                                                            |

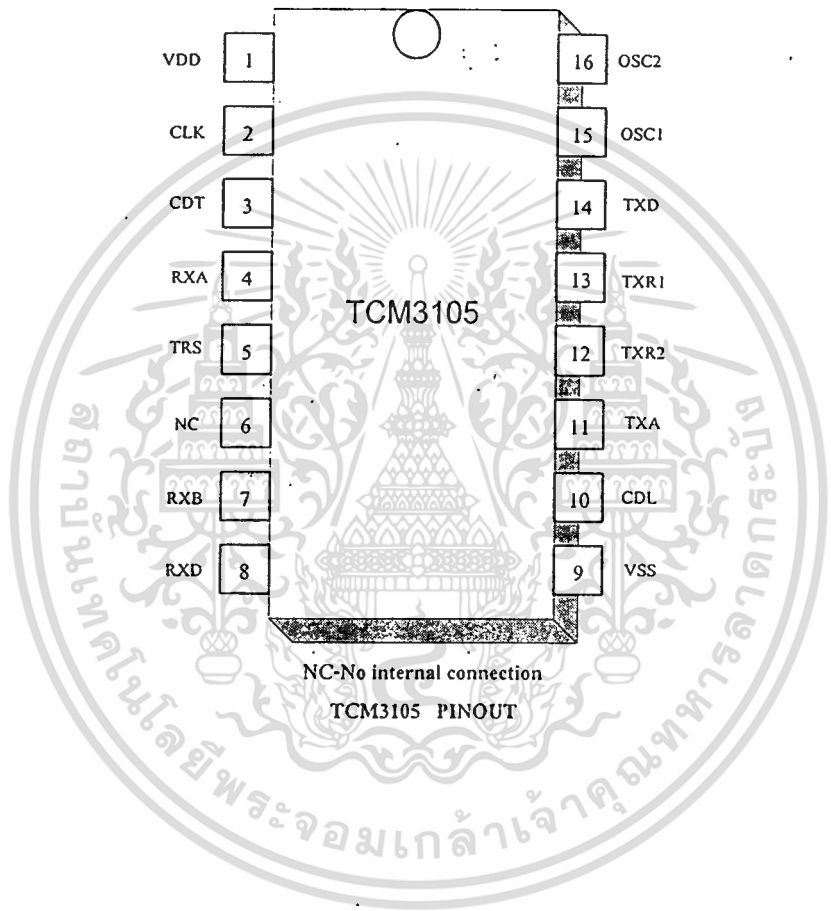
TCM3105JE, TCM3105JL

FSK MODEM

D2662. NOVEMBER 1985 - REVISED APRIL

1986

J DUAL-IN-LINE PACKAGE  
(TOP VIEW)



TEXAS  
INSTRUMENTS

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บรรณานุกรม

- จิตพงษ์ ศรีสว่าง. การเข้ารหัสสัญญาณเสียงอย่างมีประสิทธิภาพ. กรุงเทพฯ : สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง. 2537
- จิรฐา ไชยเดชาร. เครื่องป้องกันการดักฟังสำหรับเครื่องรับ - ส่งวิทยุ 27 เมกะเฮิร์ตซ์. กรุงเทพฯ : สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง. 2537
- ประสิทธิ์ ทิมพุดิ. การสื่อสารโทรคมนาคมภาคพื้นฐาน. กรุงเทพฯ : บริษัทซีเอ็ดยูเคชั่น จำกัด (มหาชน). 2539
- รัชนัย อินทุไส. การสื่อสารดาวเทียม. กรุงเทพฯ : สำนักพิมพ์ฟิสิกส์เซ็นเตอร์. 2539
- ไพโรจน์ ไวนิชกิจ และ กมล เขมะรังษี. เปิดโลกการสื่อสารไร้สาย. กรุงเทพฯ : บริษัทซีเอ็ดยูเคชั่น จำกัด (มหาชน). 2539



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## ประวัติ



|                                    |                                                                                   |
|------------------------------------|-----------------------------------------------------------------------------------|
| ชื่อผู้ทำปริญญานิพนธ์              | นายจิรศักดิ์ นุชนงค์                                                              |
| วันเดือนปีเกิด                     | 25 พฤษภาคม 2519                                                                   |
| สถานที่เกิด                        | จังหวัดเพชรบุรี                                                                   |
| ภูมิลำเนาเดิม                      | 42/2 หมู่ 6 ต.บ้านแหลม อ.บ้านแหลม จ.เพชรบุรี 76110                                |
| ที่อยู่ปัจจุบัน                    | 42/2 หมู่ 6 ต.บ้านแหลม อ.บ้านแหลม จ.เพชรบุรี 76110                                |
| โทรศัพท์                           | 032-481844 , 01-2534549                                                           |
| ประวัติการศึกษา                    |                                                                                   |
| ประถมศึกษา                         | โรงเรียนวัดลักษณาราม                                                              |
| มัธยมศึกษาตอนต้น                   | โรงเรียนบ้านแหลมวิทยา                                                             |
| ประกาศนียบัตรวิชาชีพ (ปวช.)        | วิทยาลัยเทคนิคเพชรบุรี                                                            |
| ประกาศนียบัตรวิชาชีพชั้นสูง (ปวส.) | วิทยาลัยเทคนิคเพชรบุรี                                                            |
| ปริญญาตรี                          | สาขาวิชาวิศวกรรมโทรคมนาคม<br>ภาควิชาครุศาสตร์อุตสาหกรรม<br>คณะครุศาสตร์อุตสาหกรรม |
| ผลงานที่ได้รับ                     | -                                                                                 |
| ทุนการศึกษา                        | -                                                                                 |
| คติพจน์                            | จงทำความฝันของเราให้เป็นจริง                                                      |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# ประวัติ



|                                    |                                                                            |
|------------------------------------|----------------------------------------------------------------------------|
| ชื่อผู้ทำปริญญานิพนธ์              | นายณเรศ สุรอรังช้าง                                                        |
| วันเดือนปีเกิด                     | 3 กันยายน 2519                                                             |
| สถานที่เกิด                        | จังหวัดเชียงใหม่                                                           |
| ภูมิลำเนาเดิม                      | 110/240 ถ.อินทวโรรส ต.ศรีภูมิ อ.เมือง จ.เชียงใหม่<br>50200                 |
| ที่อยู่ปัจจุบัน                    | 110/240 ถ.อินทวโรรส ต.ศรีภูมิ อ.เมือง จ.เชียงใหม่<br>50200                 |
| โทรศัพท์                           | 053-947296 , 053-210122 - 7296                                             |
| ประวัติการศึกษา                    |                                                                            |
| ประถมศึกษา                         | โรงเรียนศรีธนาวิทยาลัย                                                     |
| มัธยมศึกษาตอนต้น                   | โรงเรียนยุพราชวิทยาลัย                                                     |
| ประกาศนียบัตรวิชาชีพ (ปวช.)        | วิทยาลัยเทคนิคเชียงใหม่                                                    |
| ประกาศนียบัตรวิชาชีพชั้นสูง (ปวส.) | วิทยาลัยเทคนิคเชียงใหม่                                                    |
| ปริญญาตรี                          | สาขาวิชาวิศวกรรมโทรคมนาคม<br>ภาควิชาวิศวกรรมอุตสาหการ<br>คณะวิศวกรรมศาสตร์ |
| ผลงานที่ได้รับ                     | -                                                                          |
| ทุนการศึกษา                        | -                                                                          |
| คติพจน์                            | คนเราเลือกเกิดไม่ได้ แต่เลือกที่จะเป็นได้                                  |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## ประวัติ



|                                    |                                                                                   |
|------------------------------------|-----------------------------------------------------------------------------------|
| ชื่อผู้ทำปริญญานิพนธ์              | นายรองยุทธ เหลียงหวาน                                                             |
| วันเดือนปีเกิด                     | 2 พฤศจิกายน 2518                                                                  |
| สถานที่เกิด                        | กรุงเทพมหานคร                                                                     |
| ภูมิลำเนาเดิม                      | 134 หมู่ 3 ต.พิมลราช อ.บางบัวทอง จ.นนทบุรี<br>11110                               |
| ที่อยู่ปัจจุบัน                    | 134 หมู่ 3 ต.พิมลราช อ.บางบัวทอง จ.นนทบุรี<br>11110                               |
| โทรศัพท์                           | 02-5710442                                                                        |
| ประวัติการศึกษา                    |                                                                                   |
| ประถมศึกษา                         | โรงเรียนนันทนารวิทย์                                                              |
| มัธยมศึกษาตอนต้น                   | โรงเรียนบางบัวทองนนทบุรี                                                          |
| ประกาศนียบัตรวิชาชีพ (ปวช.)        | วิทยาลัยเทคนิคเทคโนโลยีสยาม                                                       |
| ประกาศนียบัตรวิชาชีพชั้นสูง (ปวส.) | วิทยาลัยเทคนิคเซนต์จอห์น โปล์เทคนิค                                               |
| ปริญญาดริ                          | สาขาวิชาวิศวกรรมโทรคมนาคม<br>ภาควิชาครุศาสตร์อุตสาหกรรม<br>คณะครุศาสตร์อุตสาหกรรม |
| ผลงานที่ได้รับ                     | -                                                                                 |
| ทุนการศึกษา                        | ทุนกู้ยืมรัฐบาล                                                                   |
| คติพจน์                            | ตั้งใจ จริงใจ พยายาม                                                              |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ประวัติ



ชื่อผู้ทำปริญญานิพนธ์

นายวินิจ พัยคมชาติ

วันเดือนปีเกิด

7 มกราคม 2519

สถานที่เกิด

จังหวัดตาก

ภูมิลำเนาเดิม

301/5 หมู่ 3 ต.สามเงา อ.สามเงา จ.ตาก 63130

ที่อยู่ปัจจุบัน

409/1 ถ.ตากสิน ต.หนองหลวง อ.เมือง จ.ตาก

63000

โทรศัพท์

055-540382

ประวัติการศึกษา

ประถมศึกษา

โรงเรียนชุมชนชลประทานรังสรรค์

มัธยมศึกษาตอนต้น

โรงเรียนตากพิทยาคม

ประกาศนียบัตรวิชาชีพ (ปวช.)

สถาบันเทคโนโลยีราชมงคลวิทยาเขตตาก

ประกาศนียบัตรวิชาชีพชั้นสูง (ปวส.)

สถาบันเทคโนโลยีราชมงคลวิทยาเขตตาก

ปริญญาดรี

สาขาวิชาวิศวกรรมโทรคมนาคม

ภาควิชาครุศาสตร์อุตสาหกรรม

คณะครุศาสตร์อุตสาหกรรม

ผลงานที่ได้รับ

ทุนการศึกษา

ทุนยกเว้นหน่วยกิต

คติพจน์

ทำวันนี้ให้ดีที่สุด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้