สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

# เครื่องวัดและบันทึกข้อมูลแบบสองโหมด

# ( DUAL-MODE DATA LOGGER )

T050192

จัดทำโดย

นาย ประชันศักดิ์ เกิดสำราญ รหัส 43015311

นาย อภิสิทธิ์ รูปสูง รหัส 43015345

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชา วิศวกรรมระบบควบคุม

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2545

เลขหมู่.....
เลขทะเบียน.....50192
วัน,เดือน,ปี.....27 เม.ย. 2547

.b.....
.i.....

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์ ปีการศึกษา 2545

ภาควิชาวิศวกรรมระบบควบคุม

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง เครื่องวัดและบันทึกข้อมูลแบบสองโหมด


ผู้จัดทำ

1. นาย ประชันศักดิ์ เกิดสำราญ รหัส 43015311
2. นาย อภิสิทธิ์ รูปสูง รหัส 43015345

.....อาจารย์ที่ปรึกษา

( ผศ. วรพงศ์ ตั้งศรีรัตน์ )



สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**หัวข้อปริญญานิพนธ์** เครื่องวัดและบันทึกข้อมูลแบบสองโหมด

**นักศึกษา** นาย ประชันศักดิ์ เกิดสำราญ

นาย อภิสิทธิ์ รูปสูง

**อาจารย์ผู้ควบคุมปริญญานิพนธ์** ผศ. วรพงศ์ ตั้งศรีรัตน์

**ระดับการศึกษา** วิศวกรรมศาสตรบัณฑิตสาขาวิชาวิศวกรรมระบบควบคุม

**ภาควิชา** วิศวกรรมระบบควบคุม

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

**ปีการศึกษา** 2545

## บทคัดย่อ

โครงงานนี้ได้นำเสนอเครื่องวัดและบันทึกข้อมูลแบบสองโหมด ซึ่งสามารถวัดและบันทึกข้อมูลที่ได้จากเซนเซอร์ได้ในเวลาเดียวกัน พร้อมกับบันทึกวันและเวลาทำงาน โดยข้อมูลที่เก็บไว้จะไม่สูญหาย แม้ในขณะปิดเครื่องหรือเกิดไฟฟ้าขัดข้อง อีกทั้งยังสามารถกำหนดคาบเวลาในการบันทึกข้อมูลได้ละเอียดถึง 1 วินาที การใช้งานสามารถกระทำได้สองโหมดดังนี้คือ

1. **Remote control mode** สามารถสั่งงาน ควบคุม และแสดงผลทางเครื่องคอมพิวเตอร์ เช่น การปรับเปลี่ยนค่าเริ่มต้นในการวัดค่า การดูผลการวัดแบบเวลาจริง (Real time) และการเรียกดูค่าที่บันทึกไว้โดยผ่านทางหน้าจอคอมพิวเตอร์

2. **Stand-alone mode** ใช้กรณีที่ต้องการวัดหรือบันทึกค่าข้อมูล ในบริเวณที่ไม่สามารถติดต่อกับเครื่องคอมพิวเตอร์ได้ โดยข้อมูลที่บันทึกจะถูกเก็บไว้ในหน่วยความจำข้อมูลภายในของเครื่องซึ่งสามารถนำข้อมูลเหล่านี้ไปเรียกดูได้ในภายหลัง

## ABSTRACT

The object of this project is to present the dual-mode data logger, which is employed for simutaneously measuring and recording any data from the sensor signal. Although the electrical signal dropping has occurred ,the recording data are not disappear. Moreover, the smallest range of the recording period is about 1 second. There are two operation modes as follows.

1. Remote control mode is based on the use of personal computer,which can be ordered , controlled and displayed via the user interface monitors.
2. Stand-alone mode is based on the use of microcontrolled on-board, which can be portable operated at the working site.The recording data will be download in the future.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# กิตติกรรมประกาศ

ปริญญานิพนธ์ฉบับนี้สำเร็จลุล่วงได้ด้วยดีจากความช่วยเหลือจากอาจารย์และบุคคลหลายท่าน

ขอขอบคุณผู้ช่วยศาสตราจารย์ วรพงศ์ ตั้งศรีรัตน์ อาจารย์ที่ปรึกษา ที่ให้คำปรึกษาชี้แนะแนวทางในการแก้ไขปัญหาต่างๆ ตลอดจนสถานที่และอุปกรณ์ที่ใช้ในโครงงาน และความเอาใจใส่อย่างดีเสมอมา

ขอขอบคุณพี่ๆ บริษัท อีทีที. จำกัด ที่ให้ความเอื้อเฟื้อในด้านข้อมูลเอกสารประกอบการใช้งานบอร์ดคอนโทรลเลอร์และเอกสารการใช้งานไอซี

ขอขอบคุณสำหรับพระคุณจาก คุณพ่อ คุณแม่ และครอบครัวของเราทั้งสองที่คอยห่วงใยและให้กำลังใจ ตลอดจนให้ความช่วยเหลือในทุกๆเรื่องแก่พวกเราเสมอมา

ประชันศักดิ์ เกิดสำราญ

อภิสิทธิ์ รูปสูง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# สารบัญ


| | หน้า |
|---|---|
| บทคัดย่อ | I |
| กิตติกรรมประกาศ | II |
| สารบัญ | III |
| สารบัญตาราง | V |
| สารบัญรูปภาพ | VI |
| **บทที่ 1 บทนำ** | 1 |
| 1.1 ความสำคัญและความเป็นมาของโครงงาน | 1 |
| 1.2 จุดประสงค์ของปริญญานิพนธ์ | 1 |
| 1.3 ขอบเขตโครงงาน | 1 |
| **บทที่ 2 ไมโครคอนโทรลเลอร์ AT89S8252** | 2 |
| 2.1 คุณสมบัติทั่วไป | 2 |
| 2.2 โครงสร้างภายในของ AT89S8252 | 3 |
| 2.3 พอร์ตของ AT89S8252 | 4 |
| 2.4 วงจรสร้างสัญญาณนาฬิกา | 6 |
| 2.5 การแบ่งประเภทของหน่วยความจำ | 7 |
| **บทที่ 3 ทฤษฎีการสื่อสารแบบ $I^2C$** | 8 |
| 3.1 ความรู้เบื้องต้นเกี่ยวกับ $I^2C$ | 8 |
| 3.2 คุณสมบัติทั่วไปของบัส $I^2C$ | 9 |
| 3.3 หลักการของบัส $I^2C$ | 10 |
| 3.4 สภาวะที่เกิดขึ้นบนบัส $I^2C$ | 10 |
| 3.5 การทำงานบนบัส $I^2C$ | 12 |
| **บทที่ 4 ทฤษฎีของ ADC และ DAC** | 14 |
| 4.1 ทฤษฎีของ Data Acquisition and Conversion | 14 |
| 4.2 ทฤษฎีการ Sampling | 14 |
| 4.3 การสุ่มและคงค่าสัญญาณ | 16 |
| 4.4 Quantizing Theory | 18 |
| 4.5 Quantizes Resolution and Error | 19 |
| 4.6 รหัสตัวเลขสำหรับการเปลี่ยนข้อมูล | 20 |
| 4.7 วงจรสุ่มและคงค่าสัญญาณ | 21 |


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**บทที่ 5 ทฤษฎีระบบฐานเวลาจริง ( Real Time Clock )** 23

5.1 DS1307 ไอซีสร้างฐานเวลาจริง 23

5.2 การทำงานของ DS1307 24

5.3 การจัดสรรหน่วยความจำใน DS1307 25

5.4 โหมดการทำงานของ DS1307 27

**บทที่ 6 การสื่อสารข้อมูลแบบอนุกรม** 29

6.1 มาตรฐาน EIA RS-232 29

6.2 RS-232 31

6.3 รูปแบบของข้อมูลอนุกรมและอัตราบอดในการสื่อสารข้อมูลอนุกรม 33

6.4 การเชื่อมต่อแบบอนุกรมและ UART 34

6.5 MCS-51 กับการรับส่งข้อมูล 38

6.6 Serial Port Control Register 39

6.7 Mode of Operation 41

6.8 การกำหนดค่าเริ่มต้นให้รีจิสเตอร์ในการรับส่งข้อมูล 44

6.9 อัตราการส่งข้อมูลทางพอร์ทอนุกรม 46

**บทที่ 7 การออกแบบฮาร์ดแวร์** 48

7.1 หน่วยการทำงานต่างๆของอุปกรณ์ 48

**บทที่ 8 การออกแบบซอฟท์แวร์** 54

**บทที่9 บทวิจารณ์และสรุป** 63

10.1 ผลการทดลอง 63

10.2 วิจารณ์และสรุปผลการทดลอง 74

**บรรณานุกรม**

**ภาคผนวก**

**ก. โปรแกรมควบคุม**

**ข. คู่มือไอซีของไอซีที่ใช้งาน**

**ค. คู่มือการใช้งานเครื่องวัดและบันทึกข้อมูลแบบสองโหมด**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# สารบัญตาราง


| ตารางที่ | | หน้า |
|---|---|---|
| 6.1 | ตัวเชื่อมต่อที่นิยมใช้สายส่งสัญญาณอนุกรมแบบมาตรฐาน RS-232 | 29 |
| 6.2 | ขาสัญญาณ RS-232 ทั้งแบบ 9 ขาและ 25 ขา | 32 |
| 6.5 | อัตราบอดและช่วงเวลาของแต่ละบิตข้อมูลในการสื่อสารอนุกรม | 34 |
| 6.9 | อัตราบอดทั่วไปที่ใช้ในการโอนย้ายข้อมูลอนุกรม | 36 |
| 6.14 | บิตต่างๆของรีจีสเตอร์ SCON | 40 |
| 6.15 | โหมดต่างๆของการรับส่งข้อมูลอนุกรม | 41 |
| 6.21 | ความถี่สัญญาณนาฬิกาที่ใช้กำหนดอัตราบอดค่าต่างๆ | 47 |


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# สารบัญรูปภาพ


| รูปที่ | | หน้า |
|---|---|---|
| 2.1 | ตำแหน่งของรีจีสเตอร์ต่างๆ | 3 |
| 2.2 | การจัดขาใช้งานของ AT89S8252 | 3 |
| 2.3 | การต่อขารีเซ็ตให้กับ AT89S8252 | 5 |
| 2.4 | วงจรสร้างสัญญาณนาฬิกาของ AT89S8252 | 6 |
| 2.5.1 | ผังเก็บความจำสำหรับเก็บโปรแกรมสำหรับเบอร์ 8051 | 7 |
| 2.5.2 | ผังเก็บความจำสำหรับเก็บโปรแกรมสำหรับเบอร์ 8052 | 7 |
| 3.1 | ผังการเชื่อมต่อของอุปกรณ์ต่างๆใน $I^2C$ บัส | 8 |
| 3.2 | วงจรเอาท์พุทของอุปกรณ์ในระบบ $I^2C$ บัส | 8 |
| 3.3 | การต่อตัวต้านทานพูลอัปบนสายสัญญาณระบบ $I^2C$ บัส | 9 |
| 3.4 | การต่อตัวต้านทาน RS เพื่อลดสัญญาณรบกวนขนาดใหญ่ที่อาจเข้าไปในระบบ $I^2C$ บัส | 9 |
| 3.5 | ผังเวลาแสดงสถานะต่างๆในบัส $I^2C$ | 11 |
| 3.6 | รูปแบบของข้อมูลกำหนดแอดเดรสที่ใช้ในการอ้างถึงแบบ 7 บิต | 11 |
| 3.7 | รูปแบบข้อมูลอนุกรมที่ใช้ติดต่อกับอุปกรณ์บัส $I^2C$ เมื่อใช้การอ้างถึงแบบ 7 บิต | 12 |
| 3.8 | รูปแบบข้อมูลอนุกรมที่ใช้ติดต่อกับอุปกรณ์บัส $I^2C$ เมื่อใช้การอ้างถึงแบบ 10 บิต | 13 |
| 4.1 | ระบบควบคุมที่มีการประมวลผลข้อมูลแบบดิจิตอล | 15 |
| 4.2 | ค่าความผิดพลาดจากการวัดใน Aperture time | 15 |
| 4.3 | การสุ่มสัญญาณ | 17 |
| 4.4 | ทรานสเฟอร์ฟังก์ชั่นของ Quantize 3 บิต ตามทฤษฎี | 18 |
| 4.5 | ทรานสเฟอร์ฟังก์ชั่นของ ADC 3 บิตที่ใช้รหัสออฟเซ็ทไบนารี่ | 20 |
| 4.6-(ก) | พื้นฐานของ S&H | 21 |
| 4.6-(ข) | ไดอะแกรมของวงจร S&H | 21 |
| 4.7 | รูปคลื่นเอาท์พุทของ S&H | 22 |
| 5.1 | การจัดขาของไอซี DS1307 ไอซีสร้างฐานเวลาจริง (RTC) | 23 |
| 5.2 | โครงสร้างภายในของไอซีรีลไทม์คล็อกเบอร์ DS1307 | 25 |
| 5.3-(ก) | การจัดสรรหน่วยความจำแรมภายใน DS1307 | 26 |


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

| | | |
|---|---|---|
| 5.3-(ข) | รายละเอียดของรีจีสเตอร์เก็บค่าเวลาและรีจีสเตอร์ควบคุม DS1307 | 26 |
| 5.4 | รูปแบบข้อมูลสำหรับติดต่อกับ DS1307 ในโหมดการเขียนข้อมูล | 28 |
| 5.5 | รูปแบบข้อมูลสำหรับติดต่อกับ DS1307 ในโหมดการอ่านข้อมูล | 28 |
| 6.3 | การเชื่อมต่อสัญญาณของมาตรฐาน RS-232 | 33 |
| 6.4 | รูปแบบของสัญญาณข้อมูลอนุกรมที่ใช้ในการสื่อสารข้อมูลอนุกรม | 33 |
| 6.6 | การนำ optoisolator มาใช้กับสายส่งสัญญาณ RS-232 | 34 |
| 6.7 | การแปลงข้อมูลแบบขนานเป็นอนุกรม | 35 |
| 6.8 | เวิร์ดข้อมูลขนาด 8 บิตกับ Start bit และ Stop bit ที่ใช้ในการโอนย้ายข้อมูลแบบอนุกรม | 36 |
| 6.10 | การใช้บิตพาริตี้คี่เพื่อตรวจสอบความผิดพลาดในการโอนย้ายข้อมูลแบบอนุกรม | 37 |
| 6.11 | บล็อกไดอะแกรมของ UART | 37 |
| 6.12 | สัญญาณดาต้าอินพุทส โตรบ | 38 |
| 6.13 | การรับส่งข้อมูลระหว่างรีจีสเตอร์กับบัสภายใน | 39 |
| 6.16 | ผังสัญญาณเวลาการส่งข้อมูล | 42 |
| 6.17 | ผังสัญญาณเวลาการรับข้อมูล | 42 |
| 6.18 | การส่งข้อมูลออกโดยใช้ชิฟต์รีจีสเตอร์ช่วย | 43 |
| 6.19 | การรับส่งข้อมูลในโหมด 1 | 43 |
| 6.20 | การกำหนด Baud Rate ในโหมดต่างๆ | 46 |
| 7.1 | โครงสร้างบอร์ด DUAL-MODE DATA LOGGER | 48 |
| 7.2 | วงจรสมบูรณ์ของโครงงาน | 51 |
| 7.3 | โครงงานสมบูรณ์ที่ทำการออกแบบและติดตั้ง | 52 |
| 8.1 | รูปแบบการเก็บค่าเริ่มต้นในหน่วยความจำ RAM RTC | 55 |
| 8.2 | รูปแบบการเก็บค่าเริ่มต้นในหน่วยความจำ EEPROM | 55 |
| 8.3 | ผังโปรแกรมควบคุมการทำงานหลัก | 56 |
| 8.4 | ผังโปรแกรมย่อย SETTING | 57 |
| 8.5 | ผังโปรแกรมย่อย VIEW_SETTING | 58 |
| 8.6 | ผังโปรแกรมย่อย MEASURE_TEST | 59 |
| 8.7 | ผังโปรแกรมย่อย LOGGER_DATA | 60 |
| 8.8 | ผังโปรแกรมย่อย DOWNLOAD | 61 |
| 8.9 | ผังการกำหนดรูปแบบการติดต่อผ่าน RS-232 ในโครงงาน | 62 |
| 9.1 | หน้าจอหลักของ User Interface | 63 |


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

| | | |
|---|---|---|
| 9.2 | ข้อความที่หน้าจอว่ายังมีข้อมูลเก่าที่ยังบันทึกไม่แล้วเสร็จ | 64 |
| 9.3 | ข้อความที่หน้าจอว่าข้อมูลเก่าที่ต้องการบันทึกครบถ้วนแล้ว | 65 |
| 9.4.1 | การตั้งค่า RECORD เริ่มต้น | 66 |
| 9.4.2 | การตั้งค่าจำนวน NUM OF RECORD ที่ต้องการ | 67 |
| 9.4.3 | การตั้งค่า Sampling-rate period ในการบันทึก | 67 |
| 9.4.4 | การตั้งค่าเวลาจริงของการปฏิบัติงาน | 68 |
| 9.4.5 | การตั้ง วัน/เดือน/ปี ตามเวลาปฏิบัติงาน | 68 |
| 9.5 | การเรียกดูค่าที่ตั้งไว้ใน VIEW_SETTING | 69 |
| 9.6 | การวัดค่าอุณหภูมิและสัญญาณทั้งสองช่องใน MEASURE_TEST | 70 |
| 9.7 | ข้อมูลอุณหภูมิและสัญญาณทั้งสองช่องที่ทำการบันทึก | 71 |
| 9.8 | การเรียกดูข้อมูลที่บันทึกเสร็จสิ้นแล้ว | 72 |
| 9.9 | กราฟผลข้อมูลที่ได้จากการบันทึกค่า | 73 |


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# บทที่ 1
# บทนำ

## 1.1 ความสำคัญและความเป็นมาของโครงงาน

ในปัจจุบันโครงงานทางด้านเครื่องมือวัดและบันทึกข้อมูลหรือดาต้าล็อกเกอร์ ( Data Log ger ) มักมีรูปแบบที่เป็นการแยกใช้งานอย่างใดอย่างหนึ่งคือ วัดหรือไม่ก็บันทึกค่าได้อย่างเดียว ซึ่งพบว่ามีความจำกัดอยู่มากสำหรับการใช้งานบางอย่าง เช่น ในกระบวนการผลิต หรือการทดลองทางด้านวิทยาศาสตร์ ที่บางครั้งต้องมีการวัดและบันทึกข้อมูลในเวลาเดียวกัน เพื่อนำข้อมูลเหล่านี้ไปวิเคราะห์หาความสัมพันธ์กันได้อย่างสะดวก

ปริญญานิพนธ์ฉบับนี้ได้นำเสนอ เครื่องวัดและบันทึกข้อมูลแบบสองโหมด ที่สามารถวัดและบันทึกข้อมูลได้ในเวลาและเครื่องเดียวกัน พร้อมทั้งบันทึกวันเวลาที่เริ่มทำงาน โดยที่ข้อมูลจะถูกเก็บไว้แม้ขณะเครื่องดับหรือเกิดปัญหาไฟฟ้าขัดข้อง สามารถกำหนดคาบเวลาในการบันทึกข้อมูลได้ละเอียดถึง 1 วินาที รวมถึงยังสามารถใช้คอมพิวเตอร์ควบคุมการทำงานและเรียกดูข้อมูลได้

## 1.2 จุดประสงค์ของปริญญานิพนธ์

1. เพื่อศึกษาการทำงานของไมโครคอนโทรลเลอร์ทั้งทางด้านโครงสร้างสถาปัตยกรรมและการเขียนโปรแกรมประยุกต์ใช้งาน
2. เพื่อศึกษาการทำงานและรูปแบบการติดต่อกับพอร์ตอนุกรม
3. ศึกษาการออกแบบนำส่วนฮาร์ดแวร์และซอฟท์แวร์มาประยุกต์ใช้งานจริงร่วมกัน

## 1.3 ขอบเขตโครงงาน

1. สามารถใช้ตรวจวัดและบันทึกค่าสัญญาณที่ได้จากเซนเซอร์แบบต่างๆ ซึ่งทำงานแบบเวลาจริง ( Real time ) โดยระบุวันที่และเวลาขณะที่ทำการวัดด้วย
2. สามารถใช้งานได้สองโหมด คือ

   **Remote control mode** สามารถควบคุมและแสดงผลข้อมูลผ่านทางคอมพิวเตอร์และสามารถเรียกดูข้อมูลที่บันทึกไว้โดยใช้โปรแกรมเทอร์มินอลทั่วไปได้

   **Stand alone mode** สามารถทำการวัดและบันทึกข้อมูล ในบริเวณที่ไม่สามารถติดต่อกับเครื่องคอมพิวเตอร์ได้
3. การควบคุมโดยเครื่องคอมพิวเตอร์จะผ่านทางพอร์ตอนุกรมมาตรฐาน RS-232
4. สามารถเก็บข้อมูลและนำไปดาวน์โหลดดูได้ภายหลัง และสามารถนำข้อมูลที่ได้ไปทำการแสดงผลแบบกราฟโดยใช้โปรแกรม Microsoft Excel ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# บทที่ 2

# ไมโครคอนโทรลเลอร์ AT89S8252

## 2.1 คุณสมบัติทั่วไป

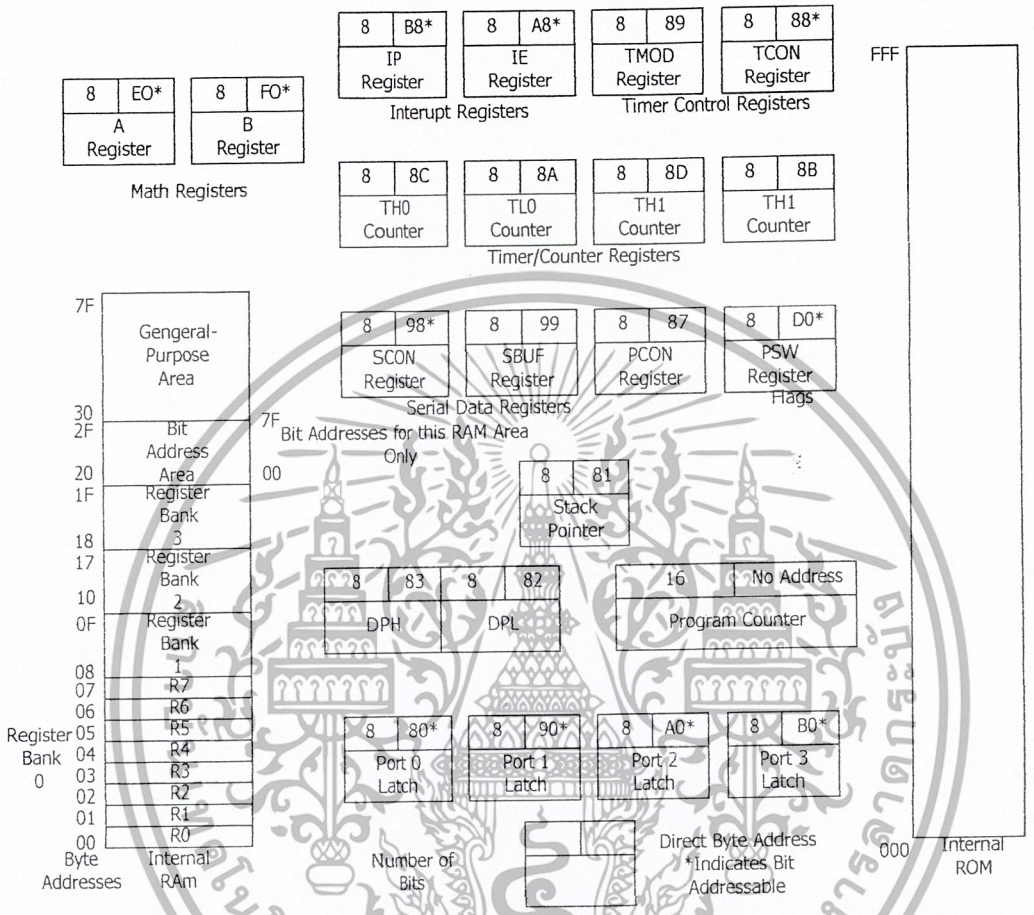
คุณสมบัติทั่วไปของไมโครคอนโทรลเลอร์ มีรายละเอียดสำคัญดังต่อไปนี้

- แหล่งจ่ายไฟ + 5 V (รองรับการทำงานได้ 4-6 โวลต์)
- มีหน่วยความจำโปรแกรม (Code Memory) แบบแฟลชขนาด 8 กิโลไบต์ที่สามารถเขียนและลบโปรแกรมได้โดยตรง โดยไม่ต้องนำตัวชิพไปทำการโปรแกรมด้วยเครื่องโปรแกรมได้ถึง 1000 ครั้ง
- หน่วยความจำข้อมูล (Data Memory) แบบ EEPROM ขนาด 2 กิโลไบต์ ที่สามารถเขียนและลบโปรแกรมได้กว่า 100,000 ครั้ง
- ใช้สัญญาณนาฬิกาได้สูงสุด 24 MHz
- สามารถกำหนดรูปแบบการป้องกันข้อมูลโปรแกรมในตัวชิพได้ 3 ระดับ
- มีระบบ Watchdog Timer
- มีระบบประหยัดพลังงาน
- มีการติดต่อกับอุปกรณ์รอบข้างแบบอนุกรมหรือ SPI ( Serial Peripheral Interface )
- แหล่งกำเนิดอินเตอร์รัพท์ 9 แหล่ง
- วงจรไทม์เมอร์และตัวนับขนาด 16 บิต จำนวน 3 ตัว
- มีหน่วยความจำแรม ( RAM ) ภายใน 256 ไบต์ ( 256x 8 bit Internal RAM )
- มีพอร์ตอินพุท/เอาท์พุท ( I/O Port ) จำนวน 32 บิต

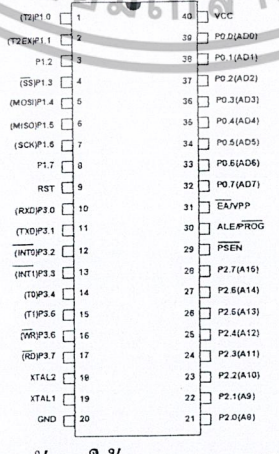
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 2.2 โครงสร้างภายในของ AT89S8252

โครงสร้างสำหรับเบอร์ AT89S8252 สามารถแสดงได้ดังในรูปที่ 2.1 โดยแสดงให้เห็นหน่วยความจำภายในเพื่อใช้ประกอบการเขียนโปรแกรมและในรูปที่ 2.2 จะแสดงการจัดขาใช้งาน




**รูปที่ 2.1 ตำแหน่งของรีจีสเตอร์ต่างๆ**




**รูปที่ 2.2 การจัดขาใช้งานของ AT89S8252**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**2.3 พอร์ตของ AT89S8252**

AT89S8252 เป็นไมโครคอนโทรลเลอร์ขนาด 40 ขา ซึ่งมีขาต่าง ๆ ดังนี้

- **Vcc** (ขา 40) ต่อกับ + 5 V
- **Vss** (ขา 20) เป็นขา GND
- **พอร์ต 0** (ขา 32-39) มีทั้งหมด 8 บิต คือ (P0.0-P0.7) ใช้งานได้ 2 หน้าที่คือส่งแอดเดรสและดาต้าออกไปให้หน่วยความจำภายนอกเมื่อทำการเขียนข้อมูลลงในหน่วยความจำภายนอกควบคุมด้วยขา Control และอีกหน้าที่หนึ่งหน้าที่ก็คือ เป็นพอร์ต I/O ถ้าต้องการให้ทำงานเป็นอินพุตพอร์ตต้องส่งลอจิก " 1 " ไปยังพอร์ตนี้ จะมีผลให้ $\overline{Q}$ ของ D-FF เป็น " 0 " ทำให้ FET ตัวล่างมีสถานะ OFF สัญญาณที่ใช้อ่านอินพุทพอร์ต PIN (พอร์ต P0.X PIN) จะใช้สัญญาณ READ LATCH เมื่อถูกกระตุ้นที่ Tri-State Buffer ตัวบน
- **พอร์ต 1** (ขา 1-8) มีทั้งหมด 8 บิต คือ (P1.0-P1.7) มีโครงสร้างคล้าย พอร์ต 0 แต่จะใช้ความต้านทานภายในพูลอัพแทน (Internal Pull Up Register)
- **พอร์ต 2** (ขา 21-28) มีทั้งหมด 8 บิต คือขา (P2.0-P2.7) มีโครงสร้างคล้าย PORT 0 โดยมี FET ตัวล่างตัวเดียวส่วนตัวด้านบนใช้ความต้านทานพูลอัพแทน (Internal Pull up) พอร์ตนี้ทำงาน 2 หน้าที่คือ สามารถใช้เป็นพอร์ตสำหรับส่งแอดเดรส 8 บิตบน (A8-A15) และเป็น I/O พอร์ตใช้งานทั่วไป เมื่อจะใช้งานเป็นอินพุตพอร์ต ต้องส่งลอจิก "1" มาที่พอร์ตนี้ก่อนเพื่อบังคับให้ FET อยู่ในสภาวะ OFF
- **พอร์ต 3** (ขา 10-17) มีทั้งหมด 8 บิต คือขา (P3.0-P3.7) มีโครงสร้างคล้าย พอร์ต 1 พอร์ตนี้ทำหน้าที่เป็น I/O พอร์ต ถ้าจะให้พอร์ตนี้เป็น I/O PORT ก็ส่งลอจิก " 1 " มาที่พอร์ตนี้และอีกหน้าที่หนึ่งคือ ส่งสัญญาณควบคุมออกมา และรับสัญญาณเข้าไป สัญญาณต่าง ๆ มีดังนี้

  **P3.0**/RXD (Serial Input Port) เป็นขาที่ใช้รับข้อมูลแบบอนุกรม

  **P3.1**/RXD (Serial Output Port) เป็นขาที่ใช้ส่งข้อมูลแบบอนุกรม

  **P3.2**/$\overline{INT0}$ (External Interrupt) ใช้รับสัญญาณขัดจังหวะจากภายนอก

  **P3.3**/$\overline{INT1}$ (External Interrupt) ใช้รับสัญญาณขัดจังหวะจากภายนอก

  **P3.4**/T0 (Timer/Counter 0 External Input) ขารับสัญญาณเข้าไปยัง Timer/Counter ที่ทำหน้าที่นับสัญญาณไซเคิลของสัญญาณ T1 นี้หรือสัญญาณนาฬิกา

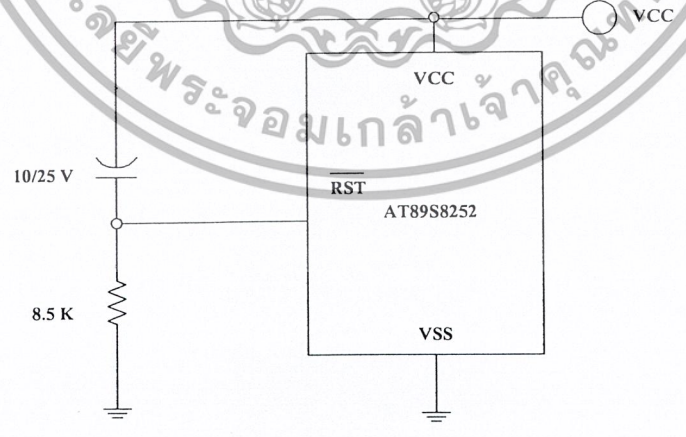
  **P3.5**/T1 (Timer/Counter 1 External Input) ขารับสัญญาณเข้าไปยัง Timer/Counter 1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**P3.6/$\overline{WR}$** (External Data Memory Write Strobe) ขาสัญญาณควบคุมการเขียนข้อมูลไปยังหน่วยความจำภายนอก

**P3.7/$\overline{RD}$** (External Data Memory Read Strobe) ขาสัญญาณควบคุมการอ่านข้อมูลมาจากหน่วยความจำสำหรับข้อมูลภายนอก

- **ALE** (ขา 30) เป็นขาส่งสโตรบสำหรับใช้ในการแลตซ์แอดเดรสไบต์ต่ำ (A0-A7) ที่ส่งออกมาจาก (พอร์ท 0) สัญญาณนี้จะแอคทีฟทุก ๆ 2 ครั้ง ใน 1 แมชชีนไซเคิล (1/6 ของสัญญาณนาฬิกา)
- **$\overline{PSEN}$** (ขา 29) เป็นขาที่ใช้ส่งสโตรบสำหรับอ่านข้อมูลจาก Program Memory ภายนอก (หน่วยความจำประเภท ROM EPROM) สัญญาณนี้จะส่งออกมา 2 ครั้งในแต่ละแมชชีนไซเคิลแต่ถ้าเป็นการอ่าน Internal Program Memory จะไม่มีสัญญาณออกที่ขานี้
- **$\overline{EA}$** (ขา 30) ถ้าป้อนลอจิก " 0 " เข้ามาที่ขานี้ซีพียูจะอ่านค่าจาก Program Memory ภายนอกชิพเท่านั้น แต่ถ้าถูกป้อนด้วยลอจิก " 1 " ก็จะอ่านโปรแกรมภายในชิพ
- **RST** (ขา 9) เป็นขารีเซ็ตซีพียูจะรีเซ็ต ได้ก็ต่อเมื่อ ป้อนลอจิก " 1 " เข้ามาที่ขานี้อย่างน้อย 2 แมชชีนไซเคิล
- **XTAL1** (ขา 19) ใช้ต่อคริสตอลภายนอกโดยเป็นอินพุตเข้าสู่วงจรออสซิเลเตอร์
- **XTAL2** (ขา 18) ใช้ต่อคริสตอลภายนอกโดยเป็นอินพุตเข้าสู่วงจรออสซิเลเตอร์

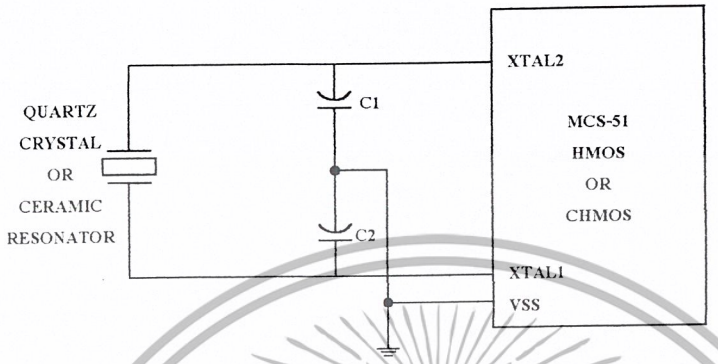



รูปที่ 2.3 การต่อขารีเซ็ทให้กับ AT89S8252

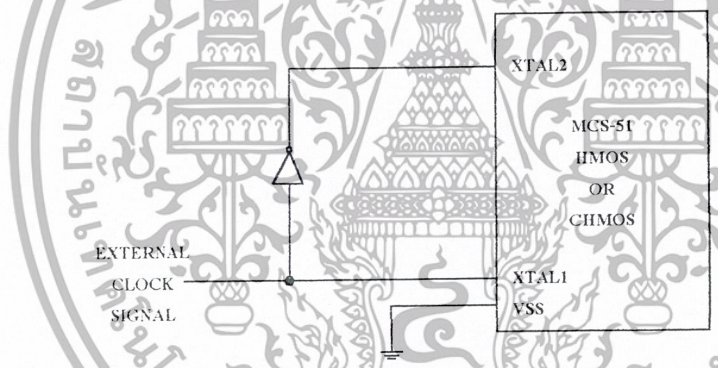
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**2.4 วงจรสร้างสัญญาณนาฬิกาของ AT89S8252**

การต่อวงจรสร้างสัญญาณนาฬิกามีอยู่ด้วยกัน 2 รูปแบบ คือแบบภายในและภายนอกแสดงดังรูปที่ 2.4




(ก) Using the on-chip Oscillator



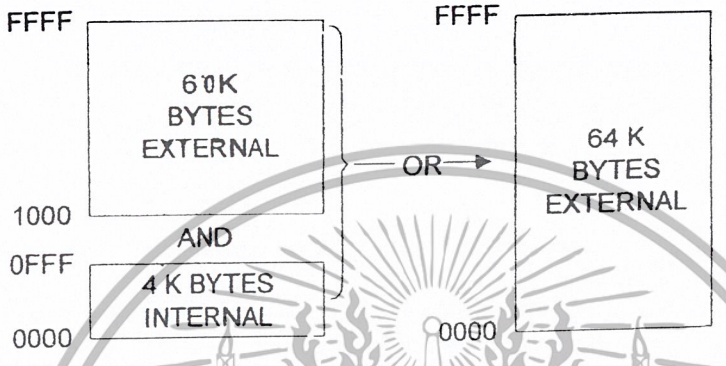

(ข) Using the External Clock

**รูปที่ 2.4 วงจรสร้างสัญญาณนาฬิกาของ AT89S8252**

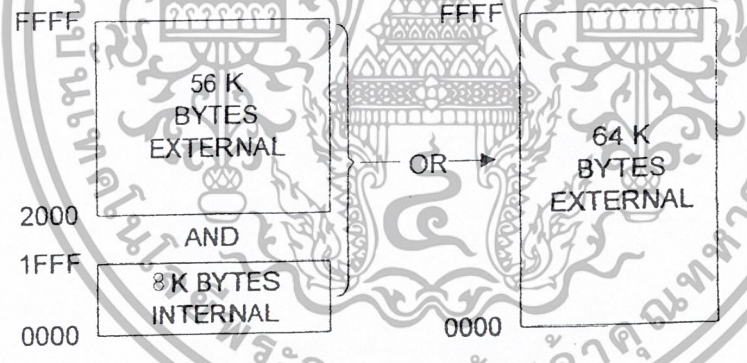
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 2.5 การแบ่งประเภทของหน่วยความจำ

หน่วยความจำที่ใช้กับ MCS-51 มีอยู่ด้วยกัน 2 ชนิด คือ Program Memory และ Data Memory ซึ่งเป็นหน่วยความจำที่ใช้เก็บโปรแกรมสั่งงานบรรจุอยู่ในชิพ 8051 ส่วนที่เป็น Program Memory ก็คือ ROM ขนาด 4 กิโลไบต์นั่นเอง แต่ถ้าเป็นเบอร์ 8052 ก็คือ ROM ขนาด 8 กิโลไบต์ ดังแสดงในรูปที่ 2.5.1 และ ที่ 2.5.2




**รูปที่ 2.5.1** ผังเก็บความจำสำหรับเก็บโปรแกรมสำหรับเบอร์ 8051



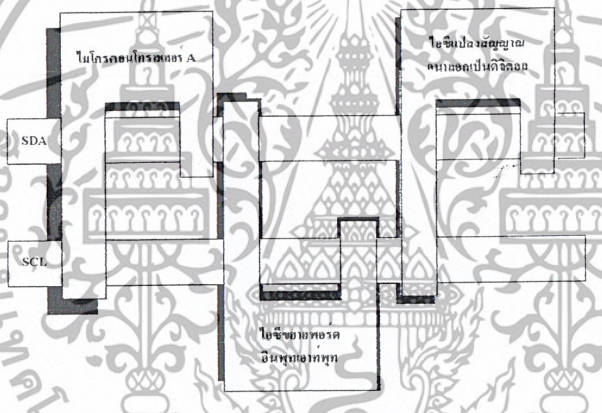

**รูปที่ 2.5.2** ผังแสดงหน่วยความจำสำหรับเก็บโปรแกรมสำหรับเบอร์ 8052

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

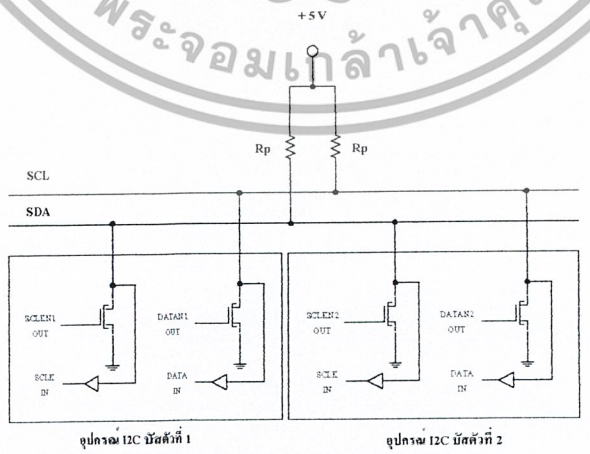
# บทที่ 3
# ทฤษฎีการสื่อสารแบบ $I^2C$

## 3.1 ความรู้เบื้องต้นเกี่ยวกับ $I^2C$

$I^2C$ ย่อมาจาก Inter – IC Communication หมายถึง การติดต่อสื่อสารระหว่างไอซีโดยบัส $I^2C$ ได้รับการพัฒนาโดยฟิลิปส์ (Philips) ด้วยจุดมุ่งหมายหลักคือ ต้องการให้ไอซีหรือโมดูลนั้นสามารถติดต่อ สั่งงาน และควบคุมภายใต้สายสัญญาณเพียง 2 เส้น โดยที่เส้นหนึ่งคือสายข้อมูลอีกเส้นหนึ่งคือ สายสัญญาณนาฬิกาที่ใช้ในการกำหนดจังหวะการทำงานการติดต่อทำงานร่วมกันของอุปกรณ์บนบัส $I^2C$ ทำได้ง่ายมาก เพียงต่อสายข้อมูลและสายสัญญาณนาฬิกาของอุปกรณ์แต่ละตัวขนานหรือพ่วงกันไป ส่วนการกำหนดแอดเดรสหรือตำแหน่งสำหรับติดต่ออุปกรณ์แต่ละตัวจะใช้รหัสข้อมูลและการกำหนดสภาวะลอจิกที่ขาแอดเดรสของอุปกรณ์แต่ละตัว




**รูปที่ 3.1 ผังการเชื่อมต่อของอุปกรณ์ต่างๆ บนระบบบัส $I^2C$**




**รูปที่ 3.2 วงจรเอาต์พุทของอุปกรณ์ในระบบบัส $I^2C$**

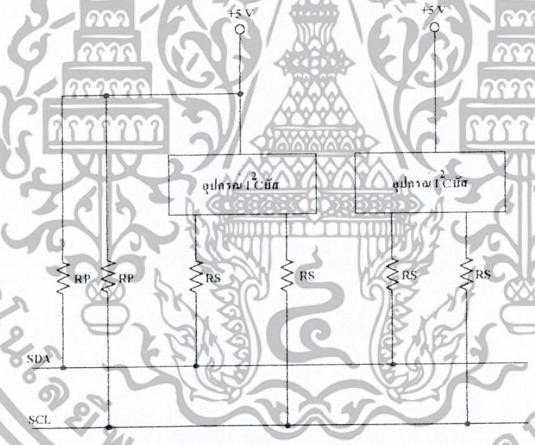
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สายข้อมูลบนบัส $I^2C$ มีชื่อเรียกอย่างเป็นทางการว่า สายข้อมูลอนุกรมหรือ SDA ( Serial Data line ) ส่วนสายสัญญาณนาฬิกามีชื่อเรียกว่า สายสัญญาณนาฬิกาอนุกรมหรือ SCL ( Serial Clock line)

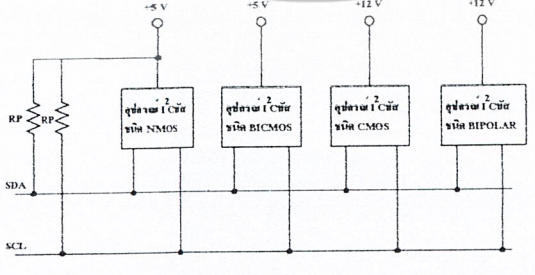
### 3.2 คุณสมบัติโดยทั่วไปของบัส $I^2C$

เนื่องสาย SDA และสาย SCL เป็นสายสัญญาณ 2 ทิศทาง(bi-directional line) จึงต้องมีการต่อตัวต้านพูลอัพกับแรงดัน +5V ไว้ตลอดเวลา เพื่อให้สายมีสถานะลอจิกสูงในขณะที่ไม่มีการติดต่อใช้งาน ทั้งยังช่วยในการป้องกันสัญญาณที่อาจมีเข้ามาในสายสัญญาณทั้งสอง วงจรเอาต์พุทของอุปกรณ์ที่ต่ออยู่อยู่บนบัส I2C ต้องมีลักษณะเป็นวงจรเดรนเปิด (open-drain) หรือคอลเล็กเตอร์เปิด

อัตราการถ่ายทอดข้อมูลบนบัส $I^2C$ สูงถึง 100 กิโลบิตต่อวินาทีในโหมดปกติ (standard mode) และสูงถึง 400 กิโลบิตต่อวินาทีใน โหมดความเร็วสูง ( fast mode ) โดยอุปกรณ์ที่ต่อร่วมอยู่อยู่บนบัส $I^2C$ จะต้องมีค่าความจุไฟฟ้ารวมที่เกิดขึ้นระหว่างสาย SDA และ SCL ไม่เกิน 400 pF การเข้าถึงอุปกรณ์บนบัส $I^2C$ ใช้ข้อมูลสำหรับการเข้าถึง 2 ค่าคือ 7บิต( 7-bit addressing ) หรือ 10 บิต (10-bit addressing)




**รูปที่ 3.3 การติดต่อตัวต้านทานพูลอัปบนสายสัญญาณในระบบบัส $I^2C$**




**รูปที่ 3.4 การต่อตัวต้านทาน RS เพื่อลดสัญญาณรบกวนขนาดใหญ่ที่อาจเข้าในบัส $I^2C$**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.3 หลักการของบัส $I^2C$

บัส $I^2C$ ประกอบด้วยสายสัญญาณ 2 เส้นคือ SDA และ SCL โดยมีอุปกรณ์ที่ต่อพ่วงบนบัสสามารถมีได้มากมาย ดังนั้นต้องมีการกำหนดรูปแบบของการติดต่อบนบัสที่เรียกว่าโปรโตคอล (protocol) เพื่อให้ผู้ใช้งานได้ทราบว่าขณะนี้มีอุปกรณ์ใดติดต่อกันอยู่และมีอุปกรณ์ตัวใดเป็นตัวรับหรือตัวส่ง

ข้อกำหนด 2 ประการสำคัญของการติดต่อบนบัส $I^2C$ คือ

(1) การถ่ายทอดข้อมูลจะเกิดขึ้นเมื่อบัสว่างเท่านั้น

(2) ในระหว่างการถ่ายทอดข้อมูล เมื่อใดก็ตามที่สาย SCL มีสถานะเป็นลอจิกสูงสายข้อมูลต้องรักษาข้อมูลไว้ อย่าให้เกิดมีการเปลี่ยนแปลงขึ้นอย่างเด็ดขาด มิฉะนั้นสัญญาณที่เกิดขึ้นจะได้รับการแปลความหมายเป็นสัญญาณควบคุมแทน

### 3.4 สภาวะที่เกิดขึ้นบนบัส $I^2C$

มีด้วยกัน 5 สภาวะ ดังนี้

(1) บัสว่าง (Bus not busy) สภาวะนี้เกิดขึ้นเมื่อสถานะลอจิกบนสายสัญญาณ SDA และ SCL เป็นลอจิกทั้งคู่ซึ่งหมายความว่า การถ่ายทอดข้อมูลเริ่มต้นขึ้น

(2) เริ่มการถ่ายทอดข้อมูล (Start data transfer) เกิดเมื่อสาย SDA มีการเปลี่ยนแปลงของระดับลอจิกจากสูงไปเป็นลอจิกต่ำในขณะที่สาย SCL มีสถานะลอจิกสูง เราจะเรียกสภาวะที่เกิดขึ้นนี้ว่า สภาวะเริ่มต้น (START)

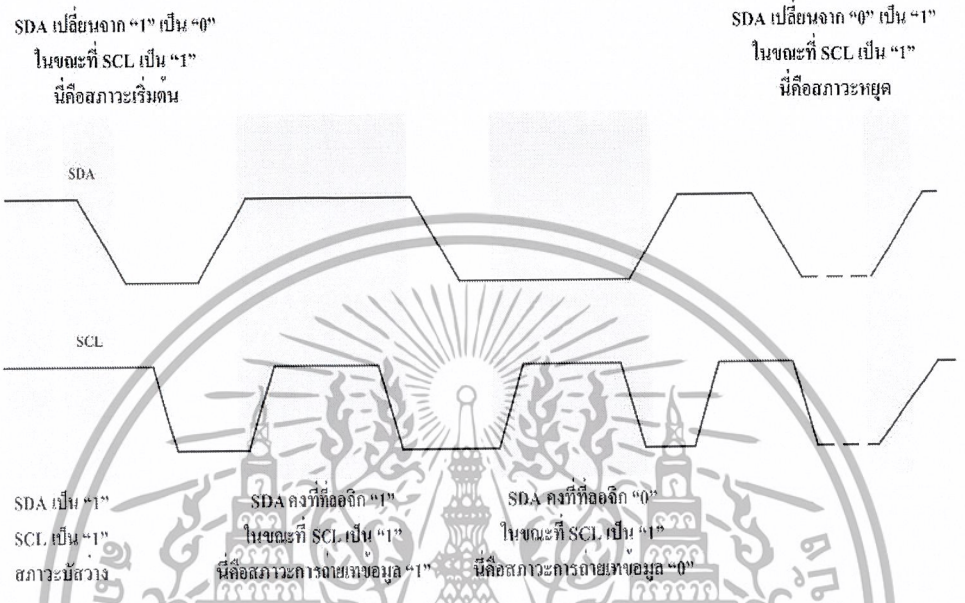
(3) หยุดการถ่ายทอดข้อมูล (Stop data transfer) เกิดขึ้นเมื่อสาย SDA มีการเปลี่ยนแปลงระดับลอจิกจากต่ำไปสูง ในขณะที่สาย SCL มีสถานะลอจิกสูงอยู่ เราเรียกว่า สภาวะหยุด (STOP)

(4) ข้อมูลดำรงอยู่บนบัส (Data valid) สภาวะนี้จะเกิดขึ้นถัดมาหลังจากที่เกิดสภาวะเริ่มต้น สถานะลอจิกบนสาย SDA ก็คือข้อมูลที่ทำการถ่ายทอดเมื่อสาย SCL เป็นลอจิกสูง สถานะที่สาย SDA ต้องคงที่ เพื่อให้อุปกรณ์รับรู้ข้อมูลในจังหวะนั้นว่าเป็น “0” หรือ “1” ข้อมูลอาจเกิดการเปลี่ยนแปลงได้ในขณะที่สาย SCL เป็นลอจิกต่ำ แต่ถ้าเมื่อใดก็ตามที่ต้องการให้เกิดการถ่ายทอดข้อมูลขึ้นอย่างสมบูรณ์ สถานะลอจิกขา SDA ต้องยังคงที่ตลอดช่วงเวลาที่สาย SCL มีสถานะลอจิกสูง หากว่าเกิดมีการเปลี่ยนแปลงสถานะลอจิกในขณะที่สาย SCL มีลอจิกสูงอยู่นั้น อุปกรณ์มาสเตอร์ที่ทำหน้าที่ในการควบคุมการถ่ายทอดข้อมูลจะแปลความหมายเป็นสภาวะหยุดหรือสภาวะเริ่มต้นก็ได้

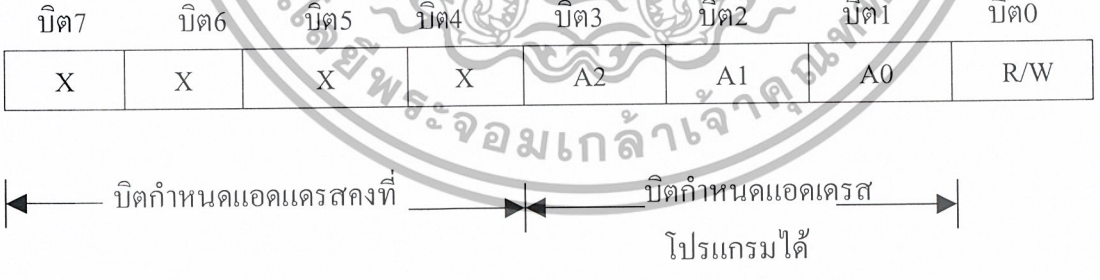
(5) รับรู้ข้อมูล (Acknowledge) เกิดขึ้นหลังจากที่มีการถ่ายทอดข้อมูลมาจากตัวส่งไปยังตัวรับเกิดขึ้นได้อย่างสมบูรณ์ โดยตัวส่งจะทำการส่งข้อมูลมา 1 บิตที่เรียกว่า บิตรับรู้ (acknowledge bit) มีสถานะลอจิกสูงหลังจากที่ส่งข้อมูลมาครบ ส่วนอุปกรณ์มาสเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จะทำการส่งสัญญาณรับรู้พิเศษซึ่งสัมพันธ์กับสัญญาณนาฬิกาเพื่อตอบสนองบิตรับรู้ที่ส่งมาจากตัวส่ง ทางด้านตัวรับส่งบิตรับรู้ที่มีสถานะลอจิกต่ำลงบนบัสอุปกรณ์สเลฟที่ถูกอ้างถึงในการติดต่ออยู่ในขณะนั้น ก็จะกำเนิดบิตรับรู้เพื่อตอบสนองให้ทราบว่ารับข้อมูลในแต่ละไบต์เรียบร้อยแล้ว




**รูปที่ 3.5 ผังเวลาแสดงสถานะต่างๆ ในบัส $I^2C$**




**รูปที่ 3.6 รูปแบบของข้อมูลกำหนดแอดเดรสที่ใช้ในการอ้างถึงแบบ 7 บิต**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.5 การทำงานบนบัส $I^2C$

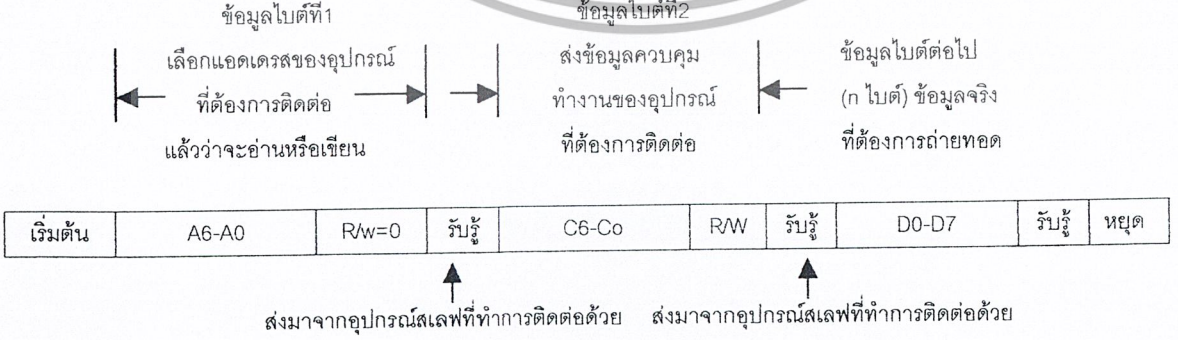
ก่อนที่จะเริ่มต้นการถ่ายข้อมูลระหว่างอุปกรณ์ที่ต่ออยู่บนบัสต้องมีการอ้างถึงเสียก่อน โดยการอ้างถึงอุปกรณ์บนบัส $I^2C$ นั้นจะใช้การอ้างแบบ 7 บิต หรือ 10 บิต ในกรณีที่มีอุปกรณ์ต่ออยู่บนบัสไม่มากใช้การอ้างถึงแบบ 7 บิตก็เพียงพอแต่ถ้ามีอุปกรณ์ต่ออยู่บนบัสมากกว่า 127 แอดเดรสก็จำเป็นที่ต้องใช้การอ้างแบบ 10 บิต และหลังจากที่ติดต่ออุปกรณ์แต่ละตัวได้เรียบร้อยแล้ว ก็จะเริ่มต้นการถ่ายทอดข้อมูลกันต่อไป

**การอ้างถึงแบบ 7 บิต (7-bit addressing)**

ข้อมูลไบต์แรกที่เกิดขึ้นหลังจากสภาวะเริ่มต้นคือ ข้อมูลที่ใช้ในการอ้างถึงอุปกรณ์ที่ต้องการติดต่อหรือข้อมูลกำหนดแอดเดรส โดยมีรูปแบบแสดงในรูป 3.7 ใน 7 บิตบน รวมทั้งบิต MSB ด้วยนั้นจะเป็นข้อมูลแอดเดรสของอุปกรณ์สเลฟที่ต้องการติดต่อ โดยแบ่งออกเป็นบิตกำหนดแอดเดรสคงที่ (fixed address bit) จำนวน 4บิต ซึ่งข้อมูลนี้อุปกรณ์แต่ละตัวจะถูกกำหนดมาจากผู้ผลิตที่ไม่สามารถทำการเปลี่ยนแปลงแก้ไขได้ ถัดมาอีก 3 บิตจะเป็นบิตกำหนดแอดเดรสสามารถทำการโปรแกรมได้ (programmable address bit) โดยผู้ใช้งานต้องกำหนดสถานะลอจิกให้แก่ขา A0-A2 ของอุปกรณ์ที่มีการเชื่อมต่อแบบบัส $I^2C$ ส่วนบิต LSB เป็นบิตที่ใช้กำหนดการอ่านหรือเขียนข้อมูลกับอุปกรณ์สเลฟตัวนั้นๆ หากบิต LSB เป็น "0" หมายถึง ต้องการเขียนข้อมูลไปยังอุปกรณ์นั้น ถ้าเป็น "1" จะเป็นการอ่านข้อมูลจากอุปกรณ์สเลฟ

ข้อมูลในไบต์ต่อมาคือ ข้อมูลควบคุม (control byte) ซึ่งในอุปกรณ์แต่ละตัวจะต้องมีการกำหนดข้อมูลควบคุมที่แตกต่างกันไปข้อมูลในไบต์ต่อมาคือ ข้อมูลที่ทำการถ่ายทอดจริง

เมื่อมีการถ่ายทอดข้อมูลในแต่ละ ไบต์ อุปกรณ์สเลฟที่รับการติดต่อต้องทำการส่งสัญญาณรับรู้ตอบกลับมาด้วยทุกครั้ง เพื่อให้กระบวนการถ่ายทอดข้อมูลสามารถดำเนินต่อไปได้ ดังรูปที่3.7 แสดงถึงรูปแบบข้อมูลอนุกรมที่เกิดขึ้นในการติดต่อบนบัส $I^2C$ ของการอ้างถึงแบบ 7 บิต

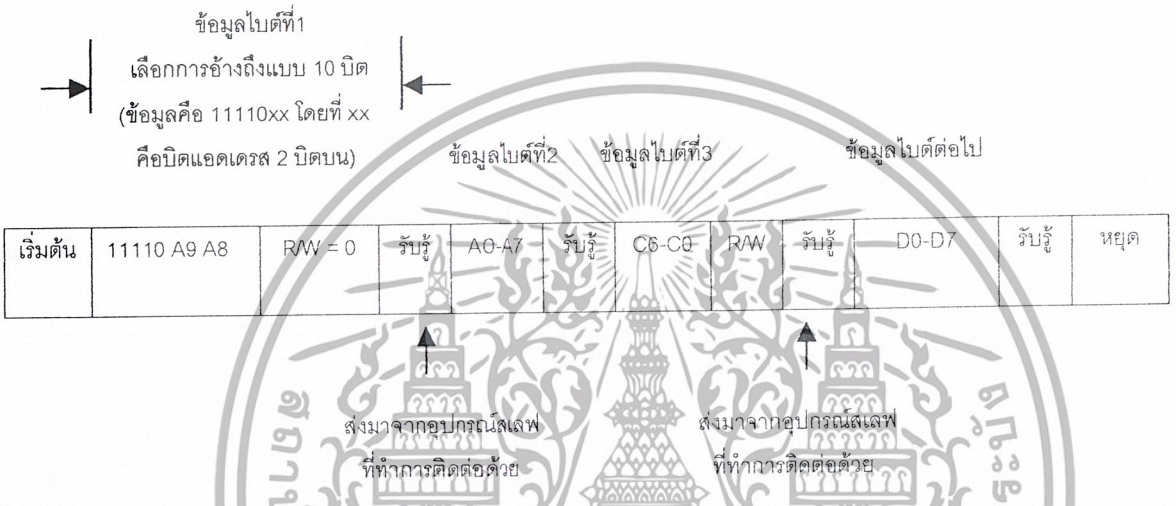



**รูปที่ 3.7 รูปแบบข้อมูลอนุกรมที่ใช้ติดต่อกับอุปกรณ์บัส $I^2C$ เมื่อใช้การอ้างถึงแบบ 7 บิต**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**การอ้างถึงแบบ 10 บิต**

ในการอ้างถึงแบบนี้ยังคงใช้รูปแบบของข้อมูลอนุกรมที่เหมือนกับแบบ 7 บิต หากแต่จะมีข้อมูลเพิ่มเติมขึ้นมาเล็กน้อย โดยในข้อมูลไบต์แรกหลังจากที่เกิดสภาวะเริ่มต้น ต้องกำหนดให้ 5บิตบนมีข้อมูลเป็น 11110 ส่วนอีก 2 บิตถัดมาเป็นบิตแอดเดรสของอุปกรณ์ที่ต้องการติดต่อด้วยข้อมูลไบต์ต่อมาเป็นข้อมูลแอดเดรสในไบต์ที่ 2 ของอุปกรณ์ที่ต้องการติดต่อด้วย ข้อมูลไบต์ถัดไปจึงเป็นข้อมูลควบคุม ข้อมูลหลังจากนั้นจะเป็นข้อมูลจริงที่ใช้ในการติดต่อ ในรูปที่ 3.8 จะแสดงรูปแบบข้อมูลของการอ้างแบบ 10 บิต




**รูปที่ 3.8** รูปแบบข้อมูลอนุกรมที่ใช้ติดต่อกับอุปกรณ์บัส $I^2C$ เมื่อใช้การอ้างถึงแบบ 10 บิต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# บทที่ 4
# ทฤษฎีของ ADC และ DAC

## 4.1 ทฤษฎีของ Data Acquisition and Conversion

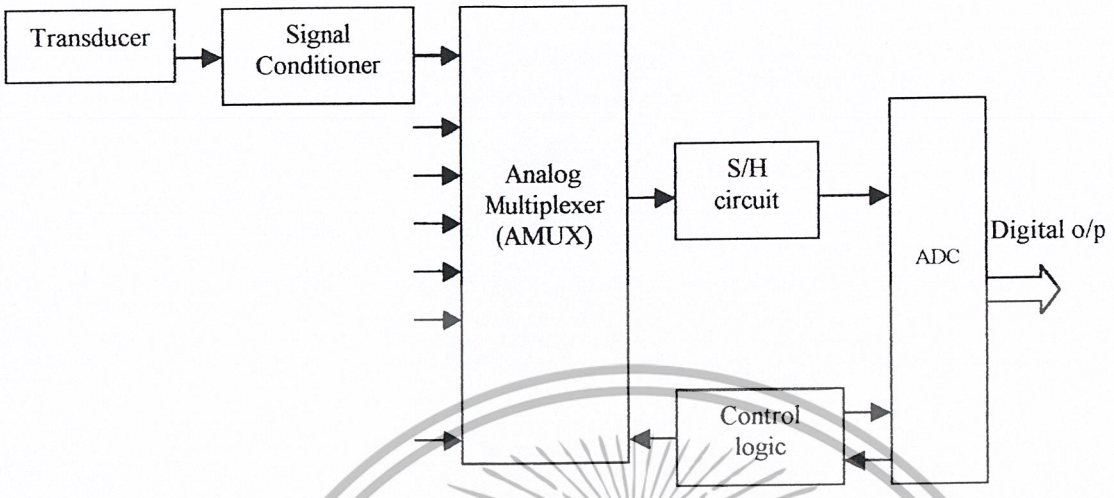
รูปแบบสัญญาณไฟฟ้าที่เราพบเห็นและคุ้นเคยในชีวิตประจำวันนั้น ส่วนมากจะอยู่ในรูปแบบสัญญาณที่มีความต่อเนื่องหรือที่เรียกว่า สัญญาณอนาลอก ( Analog signal ) ซึ่งแต่เดิมการนำเอาสัญญาณไฟฟ้ามาทำการประมวลผล (Process) เพื่อให้มีรูปแบบที่เหมาะสมจะกระทำในรูปแบบของอนาลอกนั่นเอง แต่เมื่อเทคนิคและอุปกรณ์ในการประมวลผลสัญญาณดิจิตอลได้รับการพัฒนาขึ้นมา เนื่องจากพบว่าในรูปแบบดิจิตอล การประมวลเก็บข้อมูลสื่อสารและการนำเสนอกระทำได้ง่ายกว่า ดังนั้นในการเปลี่ยนรูปแบบของสัญญาณ(Conversion) จึงได้มีความจำเป็นขึ้นมาดังแสดงในรูปที่ 4.1 เป็นตัวอย่างแสดงระบบควบคุมที่ใช้การประมวลข้อมูลในระบบดิจิตอล ซึ่งในระบบที่ยกมาเป็นตัวอย่างนี้เป็นการเปลี่ยนแปลงทางกายภาพ (Physical process) ในลักษณะใดๆก็ตามเช่น ค่าความดัน อุณหภูมิ จะถูกเปลี่ยนแปลงให้เป็นสัญญาณไฟฟ้าที่มีความต่อเนื่อง (สัญญาณอนาลอก) โดยทรานสดิวเซอร์ที่มีคุณสมบัติเหมาะสมกับรูปแบบทางกายภาพนั้น สัญญาณไฟฟ้าจะถูกปรับให้อยู่ในรูปและขนาดที่เหมาะสมก่อน โดยวงจรส่วนที่ทำหน้าที่ปรับแต่งสัญญาณ เช่น วงจรขยาย วงจรฟิลเตอร์ เป็นต้น ADC จะทำหน้าที่เปลี่ยนรูปแบบสัญญาณอนาลอกเป็นดิจิตอล ตัวประมวลผลทางดิจิตอล (Digital process) เช่น คอมพิวเตอร์ จะมีการจัดการกับข้อมูลเพื่อนำเสนอหรือถูกเปลี่ยนกลับมาให้อยู่ในรูปแบบของสัญญาณอนาลอกโดยวงจร DAC เพื่อทำการป้อนกลับไปควบคุม Physical process

ในระบบที่มีข้อมูลที่ต้องการประมวลผลในเวลาเดียวกันหลายๆ ข้อมูล หาก ADC ทำงานได้เร็วพอก็ไม่จำเป็นต้องใช้ ADC หลายๆตัวทำงานแยกกันสำหรับข้อมูลแต่ละชุด แต่จะใช้วิธีการแบ่งเวลา (Timesharing) โดยวิธี Multiplex วงจรสุ่มตัวอย่างและคงค่า (Sampling and hole,S/H) จะสุ่ม (Sample) ขนาดของสัญญาณอนาลอกมาและเก็บ (hold) ไว้ชั่วขณะ เพื่อรอให้ส่วน ADC รับไปเปลี่ยนให้สัญญาณดิจิตอลจนเรียบร้อยแล้วค่อยสุ่มสัญญาณใหม่ทั้งนี้เพื่อที่ไม่จำเป็นต้องใช้ส่วน ADC ที่มีความเร็วสูงราคาแพง ข้อมูลดิจิตอลจะถูกส่งต่อไปยัง System bus และถูกประมวลผลโดย Process ผลของการประมวลผมจะถูกส่งกลับออกมาเพื่อเปลี่ยนกลับมาเป็นสัญญาณอนาลอก

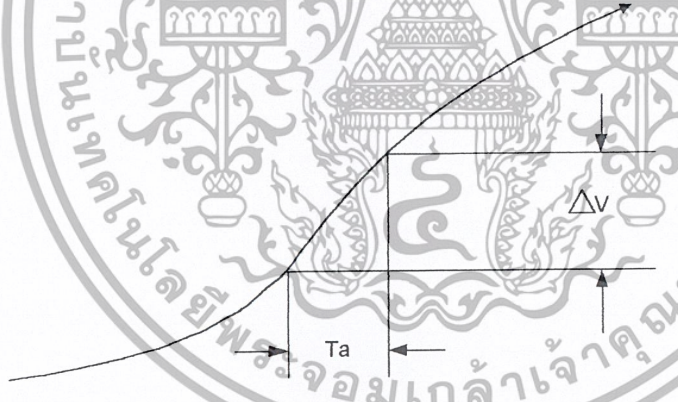
## 4.2 ทฤษฎีการ Sampling

ในการแปลงสัญญาณอนาลอกเป็นดิจิตอลนั้น ส่วนวงจร ADC นั้นจะต้องใช้เวลาช่วงหนึ่งในการจัดการ ซึ่งช่วงเวลาดังกล่าวนั้นขึ้นยังอยู่กับหลายๆ ปัจจัย อาทิเช่น เทคนิคการเปลี่ยนระดับสัญญาณและความเร็วในการทำงานของอุปกรณ์ร่วมอื่นๆ การกำหนดความเร็วของการแปลงระดับของสัญญาณนี้ขึ้นอยู่กับประยุกต์ใช้งานเฉพาะอย่างและความแม่นยำที่ต้องการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้




รูปที่ 4.1 ระบบควบคุมที่มีการประมวลผลข้อมูลแบบดิจิตอล




รูปที่ 4.2 ค่าความผิดพลาดจากการวัดใน Aperture time

ช่วงเวลาในการแปลงสัญญาณบางครั้ง อาจเรียกว่า Aperture time ซึ่งความหมายโดยทั่วไป หมายถึงช่วงเวลาที่เกิดความไม่แน่นอนขึ้นในการวัด และผลก็คือความผิดพลาด (error) ต่อค่าที่วัดได้

จากรูปที่ 4.2 สัญญาณอนาลอก V(t) มีอัตราการเปลี่ยนแปลง dv/dt ในช่วง Aperture time, Ta ดังนั้นช่วงการเปลี่ยนแปลงอนาลอกเท่ากับ $\Delta V$ โดย

$$\Delta V = Ta\ dv(t)/dt$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ดังนั้นหากเวลาที่ ADC ใช้ในการเปลี่ยนแปลงสัญญาณในช่วงเวลา Ta นี้ สัญญาณดิจิตอลที่ได้อาจจะตรงกับขนาดของสัญญาณอนาลอกค่าใดค่าหนึ่งในช่วงเวลานี้ และอาจจัดทำได้ง่ายและมีราคาถูกกว่าที่เกิดขึ้น แน่นอนว่าในบางครั้งเป็นไปได้ที่สัญญาณดิจิตอลจะตรงกับขนาดสัญญาณ - อนาลอกที่เกิดขึ้นซึ่งเรียกว่า ความผิดพลาดที่เกิดขึ้นนี้ว่า Aperture time error

ตัวอย่างในกรณีสัญญาณอินพุตเป็นรูปคลื่นซายน์ โดยที่อัตราการเปลี่ยนแปลงบนรูปคลื่นจะเกิดขึ้นสูงสุดตรงบริเวณจุดตัดแกนเวลารอบๆ จุดศูนย์โวลต์ (Zero crossing) และ Aperture time error คือ

$$\Delta V = Ta\ d(A\sin \omega t)_{t=0}/dt = TaA\omega$$

และค่าความผิดพลาดรวม ($\varepsilon$) คิดจากอัตราส่วนของขนาดเต็มสเกล คือ

$$\varepsilon = \Delta V/2A = \pi Ta$$

ดังนั้นถ้าต้องการเปลี่ยนสัญญาณเป็นรูปซายน์ความถี่ 1 กิโลเฮิร์ต ให้เป็นสัญญาณดิจิตอล 10 บิต ซึ่งยอมให้ค่าความผิดพลาดไม่เกินกว่าค่าความละเอียด (Resolution) คือ $1/2^{10}$ LSB หรือ 0.001 ดังนั้นเวลา Aperture time จะต้องอยู่ในช่วง

$$Ta = \varepsilon/\pi = 0.001/3.14x\ 1000 = 320 \times 10^{-9}$$

ถึงแม้สัญญาณ 1 กิโลเฮิร์ตจะ ไม่ใช่ความถี่สูงก็จริง แต่ ADC ที่ใช้นั้นต้องการเวลาในการปรับเปลี่ยนในเวลา 320 นาโนวินาทีให้เป็นรหัส 10 บิต ทั้งนี้ยังมีวิธีอื่นที่ไม่จำเป็นจะต้องใช้ ADC ความเร็วสูงคือการใช้การสุ่มและคงค่าสัญญาณ ซึ่งการสุ่มและคงค่าสัญญาณที่มี Aperture time ค่าน้อยๆ นั้นทำได้ง่ายและราคาถูกกว่า

**4.3 การสุ่มและคงค่าสัญญาณ (Sample and Hold) และ Aperture error**

วงจรสุ่มและคงค่า (Sample and Hold) จะทำการสุ่ม (Sampling) สัญญาณอินพุตและนำมาเก็บ(Hold)ไว้ในช่วงเวลาหนึ่งได้ ซึ่งส่วนใหญ่จะใช้การประจุแรงดันนั้นไว้ในตัวเก็บประจุที่รั่วไหลต่ำ Aperture time ของการสุ่มและคงค่าสัญญาณ คือเวลาตั้งแต่เริ่มสุ่มสัญญาณจนเก็บประจุค่าแรงดันจนถึงค่าที่สุ่มซึ่งขึ้นอยู่กับแบนด์วิดธ์และช่วงเวลาในการสวิทซ์ (Switching time)

ในการสุ่มสัญญาณอนาลอกจะถูกสุ่มเป็นระยะๆ ซึ่งคงที่ตามรูปที่ 4.3 การสุ่มจะเป็นการตัดต่อสัญญาณอนาลอกในช่วงเวลาอันสั้นด้วยสวิทซ์ที่ทำงานด้วยความเร็วสูง โดยที่ผลของการสุ่ม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

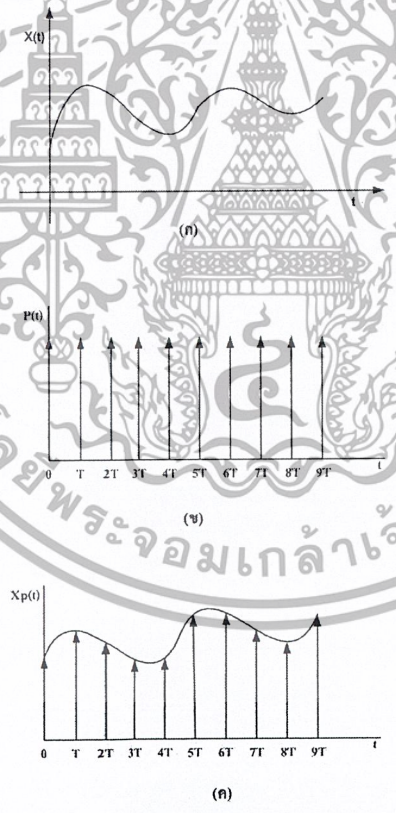
สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง



**4.3 การสุ่มและคงค่าสัญญาณ (Sample and Hold) และ Aperture error**

วงจรสุ่มและคงค่า (Sample and Hold) จะทำการสุ่ม (Sampling) สัญญาณอินพุตและนำมาเก็บ(Hold)ไว้ในช่วงเวลาหนึ่งได้ ซึ่งส่วนใหญ่จะใช้การประจุแรงดันนั้นไว้ในตัวเก็บประจุที่รั่วไหลต่ำ Aperture time ของการสุ่มและคงค่าสัญญาณ คือเวลาตั้งแต่เริ่มสุ่มสัญญาณจนเก็บประจุค่าแรงดันจนถึงค่าที่สุ่มซึ่งขึ้นอยู่กับแบนด์วิดธ์และช่วงเวลาในการสวิทช์ (Switching time)

ในการสุ่มสัญญาณอนาลอกจะถูกสุ่มเป็นระยะๆ ซึ่งคงที่ตามรูปที่ 4.3 การสุ่มจะเป็นการตัดต่อสัญญาณอนาลอกในช่วงเวลาอันสั้นด้วยสวิทช์ที่ทำงานด้วยความเร็วสูง โดยที่ผลของการสุ่มสัญญาณด้วยความเร็วจะเสมือนกับการคูณขบวนสัญญาณพัลส์แคบๆกับสัญญาณอนาลอก ซึ่งจะได้เป็นสัญญาณมอดูเลท (modulate) ระหว่างขบวนการพัลส์กับสัญญาณอนาลอก ปัญหาที่ว่าอัตราการสุ่มสัญญาณนั้นควรมีขนาดเท่าใดจึงจะไม่ทำให้ข้อมูลสูญเสียไปนั้น พบว่าเมื่อสัญญาณนั้นถูกเปลี่ยนกลับมาเช่นเดิม คำตอบคือ ขึ้นอยู่กับความถี่ของสัญญาณอนาลอก ทฤษฎีของการสุ่มกล่าวไว้ว่า “ถ้าสัญญาณต่อเนื่องซึ่งมีความถี่และฮาร์โมนิคส์ไม่เกิน $f_c$ ถูกสุ่มด้วยอัตราการสุ่มไม่น้อยกว่า $2f_c$ แล้วสัญญาณดังกล่าวจะสามารถเปลี่ยนกลับมาได้อย่างเดิมโดยผิดเพี้ยนไป”




**รูปที่ 4.3 การสุ่มสัญญาณ**

**(ก) สัญญาณอนาลอกอินพุต**

**(ข) พัลส์ที่มาสุ่มสัญญาณ**

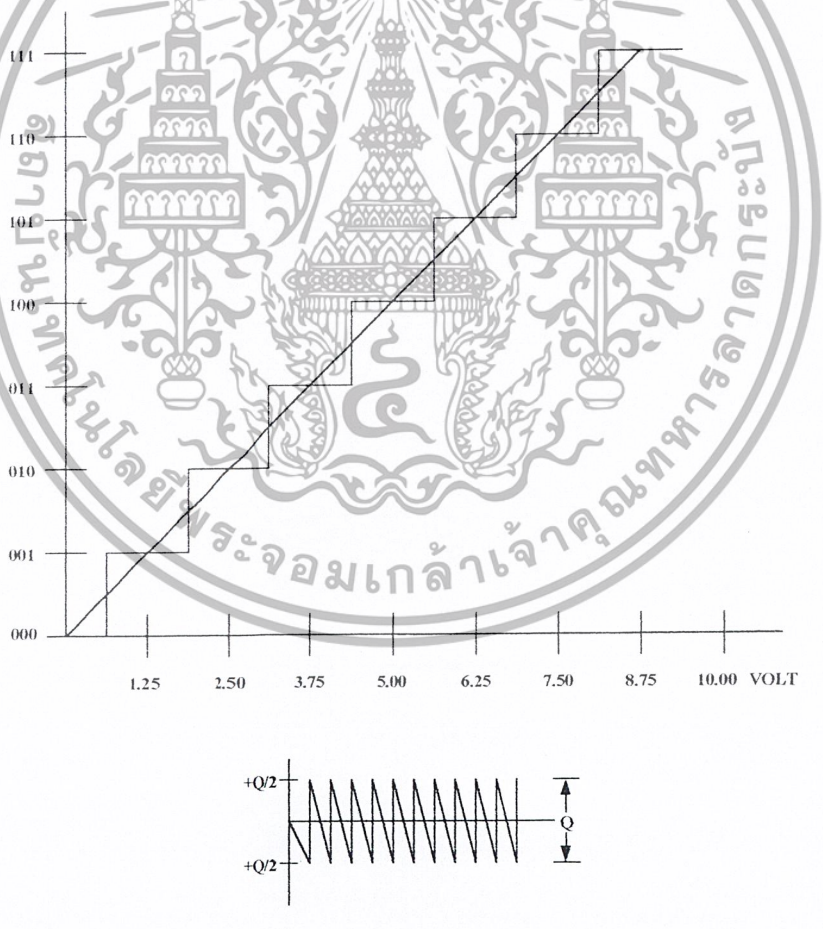
**(ค) สัญญาณอนาลอกหลังการสุ่ม**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
50192

**4.4 Quantizing Theory**

Quantizing เป็นขบวนการที่เปลี่ยนแปลงของสัญญาณอนาลอกเป็นสัญญาณที่ไม่ต่อเนื่องหลังการสุ่ม โดยผ่านขบวนการเข้ารหัส (Coding) จัดให้สัญญาณที่ไม่ต่อเนื่องนั้นอยู่ในรูปที่ง่ายต่อการประมวลผลและเป็นสัดส่วนสัมพันธ์กับสัญญาณอนาลอก เช่น ในรูปของรหัสไบนารี(Binary) เป็นต้น หากนำเอาขนาดสัญญาณอนาลอกและสัญญาณดิจิตอลที่ได้รับการ Quantize มาเขียนกราฟก็จะได้กราฟแสดง Quantize transfer function ดังรูปที่ 4.4

รูปกราฟแสดงถึงความสัมพันธ์ระหว่างสัญญาณอนาลอกขนาดอยู่ระหว่าง 0 ถึง +10 โวลท์ ถูก Quantize และ Encode เป็นรหัสไบนารี่ (Binary) 3 บิต ได้ 8 ระดับ จาก 000 ถึง 111 เนื่องจากในระบบไบนารี่รหัสดิจิตอลแต่ละค่าจะแทนขนาดของสัญญาณอนาลอก แต่ละค่าที่เป็นสัดส่วนกับค่าเต็มสเกลโดยค่าสูงสุดของรหัสดิจิตอลคือ ทุกบิตที่เป็น 1 จะเท่ากับสัญญาณอนาลอกเต็มสเกลคูณด้วย $(1-2^{-n})$ โดย n เป็นจำนวนบิตของรหัสดิจิตอลแต่ละบิตที่เป็น 1 จะเท่ากับขนาดเต็มสเกลของอนาลอกคูณกับค่าน้ำหนัก (weighting) ของรหัสชนิดนั้น หารด้วย $2^n$




**รูปที่ 4.4 ทรานส์เฟอร์ฟังชันของ Quantize 3 บิต ตามทฤษฎี**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวอย่างเช่น ค่าเต็มสเกลของสัญญาณอนาลอกเป็น 10 โวลท์ ( รหัส 1011 ) จะแทนขนาดสัญญาณอนาลอกอินพุต

$$Vin = (Rs/2^n) \times \{(1x2^3)+(0x2^2)+(1x2^1)+(1x2^0)\}$$

จุดสำคัญเกี่ยวกับกราฟทรานเฟอร์ฟังก์ชันรูปที่ 4.4 อันดับแรกได้แก่ ความละเอียดของ Quantizer ซึ่งสามารถกำหนดได้จากจำนวนบิตของรหัสดิจิตอล หรือจากกราฟคือขนาดกว้างของระดับ (Step) ทางแกนอนาลอกอินพุตว่าเป็นสัดส่วนเท่าใดระหว่างค่าเต็มสเกลอนาลอกกับค่า $2^n$

จำนวนสถานะเอาท์พุตกำหนดได้จากจำนวนบิตคือ เท่ากับ $2^n$ สถานะ ตัวอย่างกรณี ADC 8 บิต Quantizer จะให้เอาท์พุต 256 สถานะและ 12 บิตให้ 4096 สถานะต่อค่าเต็มสเกลอนาลอกในไดอะแกรมทรานเฟอร์ฟังก์ชั่นจะเห็นจุดแบ่ง ระดับ (Decision point หรือ Theshold level) พบว่าสัญญาณอนาลอกจะมีจำนวน $2^n$-1 จุด ที่อยู่ที่ 0.625, 1.875, 3.125, 4.375, 5.625 และ 8.125 โวลท์ ระหว่างจุดดังกล่าวเป็นสัญญาณอนาลอกซึ่งแปลงเป็นรหัสดิจิตอล 1 สถานะดังนั้นค่าเหล่านี้จะต้องปรับให้ถูกต้องมากที่สุด เพื่อแปลงขนาดของ อนาลอกให้ตรงกับค่าที่ทำการ Quantizer แรงดันที่ 1.25, 2.50, 3.75, 5.0, 6.25, 7.5, 8.75 โวลท์ เป็นจุดกึ่งกลางของรหัสดิจิตอลสถานะสุดท้าย สังเกตว่าในทางทฤษฎีแล้ว เส้นตรงนี้จะต้องผ่านจุดกึ่งกลางของรหัสดิจิตอลสถานะสุดท้าย สังเกตว่าในทางทฤษฎีแล้วเส้นตรงนี้จะต้องผ่านจุดกึ่งกลางของทุกระดับดิจิตอล

### 4.5 Quantizer Resolution and Error

ในแต่ละสถานะของสัญญาณดิจิตอลเอาท์พุต จะแทนขนาดของสัญญาณอนาลอกค่าใดค่าหนึ่งในช่วงเล็กๆ ระหว่างจุดแบ่งระดับ เราเรียกช่วงเล็กๆ นี้ว่าเป็นขนาดหนึ่ง Analog quantization หรือหนึ่งควันตัม (Quantum) หรือ 1LSB (Least significant bit) ของการแปลงสัญญาณตัวอย่างในรูปที่ 4.4(บน) ควันตัมคือ 1.25 โวลท์ ค่านี้ได้จากการคำนวณจาก

$$Q = FSR / 2^n$$

โดยที่ FRS คือช่วงเต็มสเกลของแรงดันอนาลอก ( Full Scale Range ) และ n คือ จำนวนบิตของรหัสดิจิตอล

จากสมการจะเห็นได้ว่า หากจำนวนบิตมากขนาดของควันตัมก็จะลดลงและถ้าให้สัญญาณอินพุตของ Quantizer กวาดไปตลอดช่วงของสัญญาณอนาลอก ก็จะเห็นช่วงผลต่างของสัญญาณ

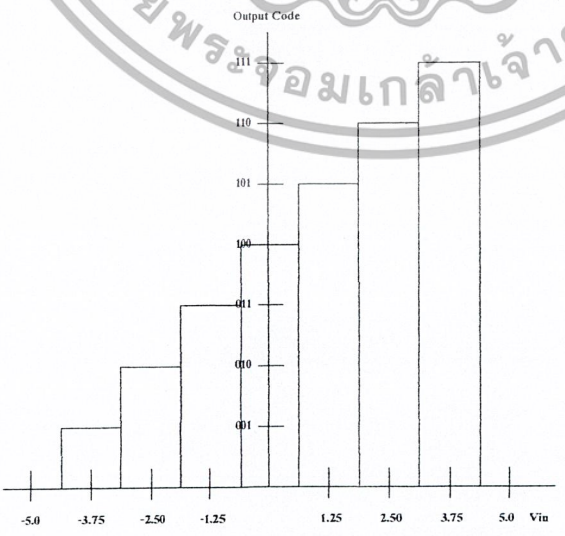
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อนาลอกอินพุตและดิจิตอลเอาท์พุตเป็นช่วงพล๊อตได้เป็นรูปฟันเลื่อย ดังรูปที่ 4.4 ด้านล่าง ซึ่งเรียกว่า Quantizing error ซึ่งค่าความผิดพลาดแบบนี้ก็คือ 1 ช่วงสัญญาณอนาลอกที่แปลงไปเป็นรหัสดิจิตอล 1 สถานะ ดังกล่าวมาแล้วนั่นเอง

ค่าความผิดพลาดนี้เป็นธรรมชาติของ Quantizing จะทำการแก้ไขไม่ได้นอกจากการเพิ่มจำนวนของ Quantizer ให้มากขึ้น และเอาท์พุตค่าความผิดพลาดจะอยู่ระหว่าง 0-Q/2 Error อาจจะเป็นที่ศูนย์ เมื่อสัญญาณอนาลอกค่าที่จุดกึ่งกลางของควันตัมพอดี ลักษณะฟังก์ชั่นของค่าความผิดพลาดจะสามารถพิจารณาเป็นสัญญาณรบกวนทางอินพุต ซึ่งมีค่าเป็น Q $V_{p-p}$ และค่าเฉลี่ยเป็นศูนย์ ค่าอาร์เอ็มเอส (root mean square,rms) เป็น Q/2 3 ซึ่งจะได้จากการพิจารณาวิเคราะห์สัญญาณรูปคลื่นฟันเลื่อย

**4.6 รหัสตัวเลขสำหรับการเปลี่ยนข้อมูล**

รหัสตัวเลขที่นิยมนำมาใช้ในระบบเปลี่ยนข้อมูลได้แก่รหัสไบนารีหรือที่เรียกว่า Straight binary โดยที่รหัสไบนารีสถานะสูงจะแทนจากสัญญาณอนาลอก FSR ($1-2^{-n}$) โวลท์ ตัวอย่างเช่น หากสัญญาณอนาลอกเต็มสเกล (FSR) เท่ากับ 20 โวลท์ สำหรับวงจร ADC ขนาด 12 บิต รหัส 1111 1111 1111 จะแทนสัญญาณอนาลอกขนาด $20(1-2^{-12})$ หรือ 19.39951171 โวลท์ นอกจากรหัสไบนารีธรรมดาดังกล่าวยังมีการใช้ระบบไบนารี แบบอื่นๆ ในระบบการแปลงสัญญาณ ได้แก่ ออฟเซ็ทไบนารี, Two's complement, BCD ซึ่งในแต่ละชนิดมีข้อดีและความเหมาะสมต่างกัน ตัวอย่างเช่น ระบบ BCD เหมาะสำหรับการคำนวณทางคณิตศาสตร์ลอจิกและ สำหรับระบบออฟเซ็ตไบนารีนั้น เหมาะสำหรับการคำนวณทางคณิตศาสตร์ลอจิก และสำหรับในแบบระบบออฟเซ็ตไบนารีเหมาะสำหรับการแปลงสัญญาณอินพุตที่ทั้งช่วงบวกและลบ ในรูปที่4.5 แสดงทรานเฟอร์ฟังก์ชั่นของ ADC 3 บิตที่ใช้รหัสออฟเซ็ตไบนารี



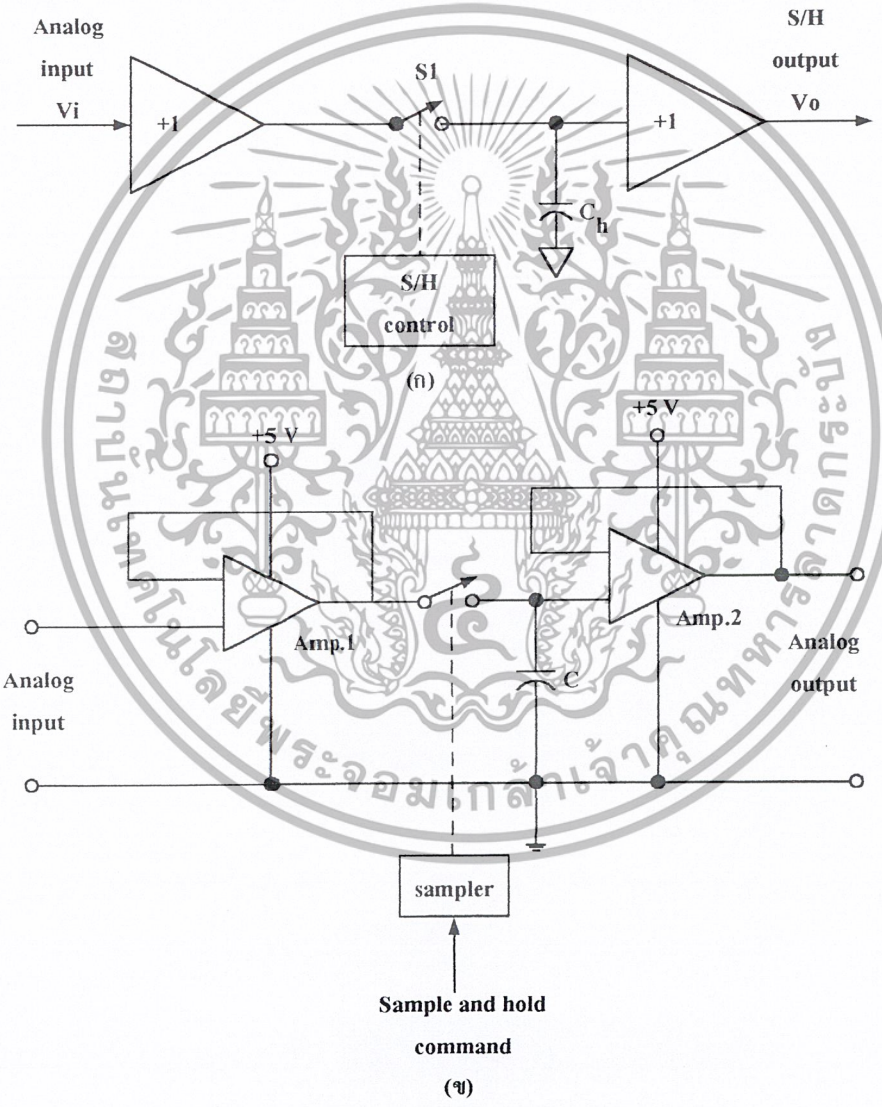

**รูปที่4.5 ทรานเฟอร์ฟังก์ชั่นของ ADC 3 บิตที่ใช้รหัสออฟเซ็ตไบนารี**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**4.7 วงจรสุ่มและคงค่าสัญญาณ (Sample and Hold,S&H)**

ที่ผ่านมาได้กล่าวถึงจุดมุ่งหมายในการใช้วงจร S&H กับ ADC และต่อจากนี้จะได้กล่าวถึงรายละเอียดของวงจร S&H บางแบบที่นิยมใช้ในปัจจุบัน

ความจริงแล้ววงจร S&H มิได้มีใช้ในเฉพาะกับ ADC เท่านั้น แต่ยังใช้กันทั่วไปในระบบ Data distribution, Sampling scope, DVM, Reconstruction filter และอนาลอกคอมพิวเตอร์เป็นต้น วงจร S&H โดยพื้นฐานแล้วอุปกรณ์หรือวงจรเก็บแรงดัน (Voltage memory) ซึ่งมีการใช้อุปกรณ์ร่วมสำคัญคือตัวเก็บประจุ ในรูปที่ 4.6 (ก) แสดงวงจรพื้นฐาน S&H จากลักษณะอนาลอกอินพุต สัญญาณการสุ่มและเอาท์พุต

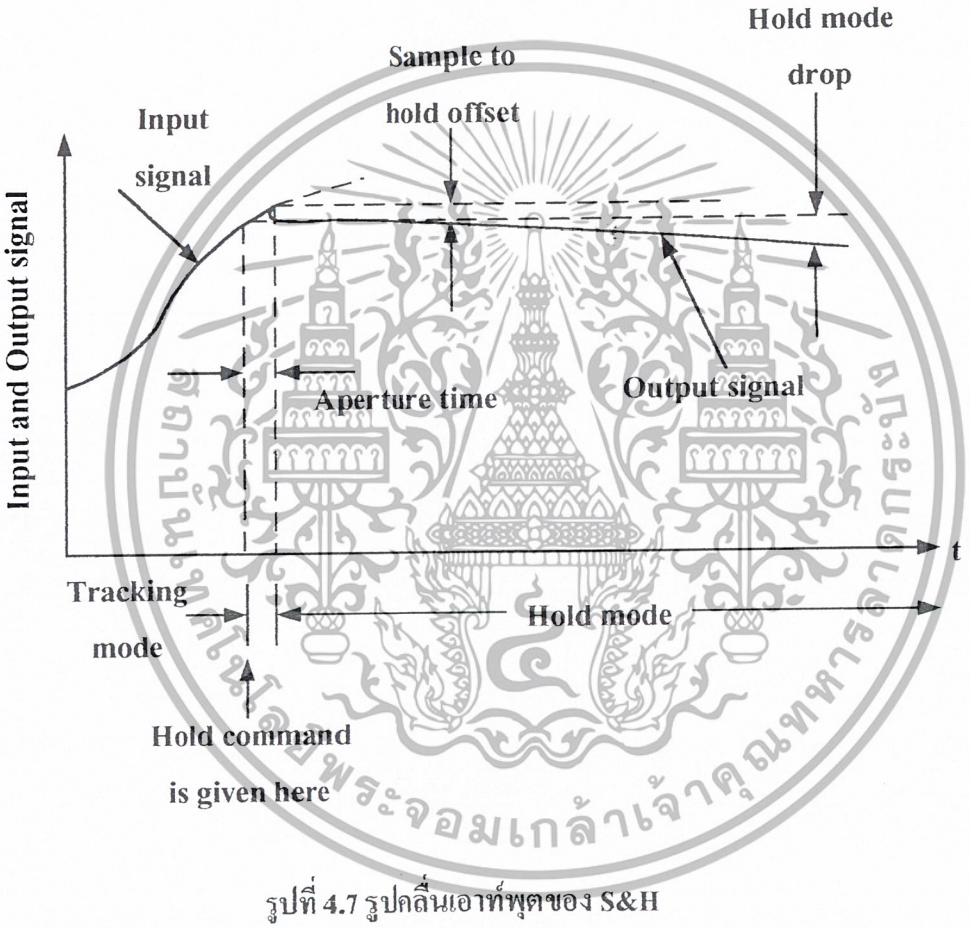



**รูปที่ 4.6 (ก) พื้นฐานของ S&H**

**(ข) ไดอะแกรมของวงจร S&H**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 4.6 ( ข) แสดงวงจรใกล้เคียงกับวงจรจริงๆที่ใช้ในทางปฏิบัติ โดยเพิ่มเติมบัฟเฟอร์แอมป์ปลิไฟล์เข้าทางส่วนอินพุตและเอาท์พุตของ S&H พื้นฐาน แอมปลิไฟล์เออร์ด้านอินพุตช่วยทำให้วงจรมีอินพุตอิมพีแดนซ์สูง สะดวกต่อการใช้งานและยังสามารถเพิ่มกระแสเพื่อทำการประจุ $C_h$ ได้เร็วขึ้น ส่วนทางเอาท์พุตช่วยทำให้เอาท์พุตอิมพีแดนซ์สามารถขับ ADC ได้ง่าย มีจุดสำคัญที่ต้องพิจารณาคือ ในส่วนของแอมปลิไฟร์เออร์เหล่านี้ปกติแล้ว จำเป็นต้องเป็นแอมปลิไฟร์เออร์ที่ใช้กระแสอินพุตต่ำ ทั้งนี้เพื่อให้ดึงมีการกระแสจากตัวเก็บประจุในช่วงที่คงค่าสัญญาณให้น้อยที่สุดดังที่แสดงอยู่ในรูปที่ 4.7 ซึ่งตามปกติแล้วมักใช้แอมปลิไฟร์เออร์ที่มี FET หรือ MOSFET เป็นอินพุตเพราะการไบอัสด้วยแรงดันทำให้กระแสอินพุตต่ำด้วย




รูปที่ 4.7 รูปคลื่นเอาท์พุตของ S&H

วงจร S&H ในระบบ Data acquisition ที่นิยมใช้สองแบบคือ Sample-hold และ Track-hold และวงจร S&H จะใช้วิธีสุ่มสัญญาณอย่างรวดเร็วแล้วเข้าสู่ Holding period ซึ่งหมายความว่าสวิทซ์ควบคุมจะต้องตัดต่อในช่วงเวลาอันสั้นอย่างต่อเนื่อง ในส่วนของ Track-holds จะตัดสวิทซ์สุ่มช้ากว่า

การจัดวงจร S&H มีได้หลายลักษณะ ซึ่งอาจนำไอซีหรือทรานซิสเตอร์มาประกอบเป็นวงจร ตลอดจนการสร้างวงจรทั้งหมดของ S&H ลงบนชิปไอซีเดียว เช่น เบอร์ LF398

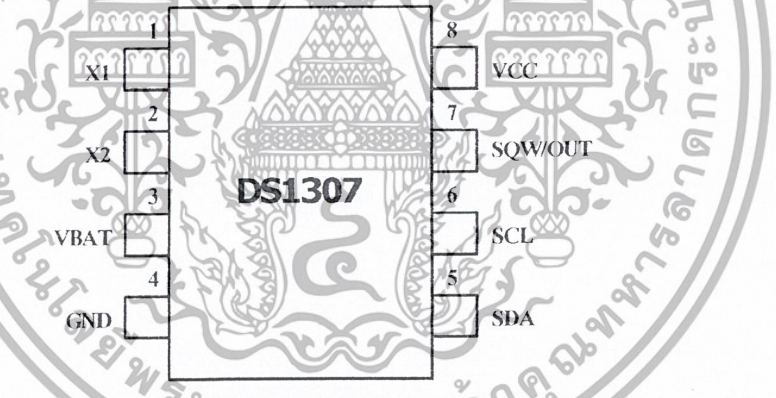
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# บทที่ 5
# ทฤษฎีระบบฐานเวลาจริง

## 5.1 DS1307 ไอซีสร้างฐานเวลาจริงหรือ Real Time Clock ( RTC )

มีหน้าที่สร้างฐานเวลาจริงให้แก่ระบบไมโครคอนโทรลเลอร์ โดยที่ DS1307 จะให้ข้อมูลเกี่ยวกับเวลาทั้งหมด ไม่ว่าจะเป็นค่าของเวลาที่ละเอียดถึงหลัก วินาที,นาที,ชั่วโมง,วันที่,วันสัปดาห์, เดือน และปี โดยสามารถที่จะปรับวันเดือนปีให้ตรงตามปฏิทินได้อย่างถูกต้อง รวมถึงการกำหนดวันในปีอธิกสุรทินด้วย คุณสมบัติทางเทคนิคที่สำคัญมีดังนี้

- เป็นไอซีสร้างฐานเวลาให้ข้อมูลตั้งแต่วินาทีจนถึงปี รวมถึงการปรับวันในปีอธิกสุรทินด้วยสามารถให้ข้อมูลเวลาได้อย่างเที่ยงตรงถึงปีคริสตศักราช 2100
- มีหน่วยความจำนอนโวลาไทม์แรม 56 ไบต์อยู่ภายใน ซึ่งสามารถใช้เก็บข้อมูลทั่วไปได้
- ใช้ในการเชื่อมต่อแบบระบบบัส $I^2C$
- มีวงจรตรวจจับไฟเลี้ยงต่ำหรือหายไปอย่างอัตโนมัติ และสามารถที่จะรักษาข้อมูลเวลาได้ แม้ไม่มีไฟเลี้ยงไอซี




รูปที่ 5.1 การจัดขาของไอซี DS1307 ไอซีสร้างฐานเวลาจริง (RTC)

**รายละเอียดขาต่อใช้งานของ DS1307**

การจัดขาของ DS1307 แสดงได้ดังรูปที่ 5.1 โดยแต่ละขามีหน้าที่และการใช้งานดังนี้

$V_{CC}$, **GND** (ขา 8,4) : ต่อไฟเลี้ยง +5V

$V_{BAT}$ (ขา 3) : ใช้ต่อแบตเตอรี่ 3V เพื่อรักษาการทำงานของวงจรสร้างฐานเวลาจริงของ DS1307 ให้คงอยู่ต่อไป แม้ว่าในขณะที่ไม่มีไฟเลี้ยงจ่ายให้แก่ DS1307 โดยชนิดของ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แบตเตอรี่ที่เหมาะสมนั้นคือ แบตเตอรี่แบบลิเธียมซึ่งมีความจุ 40mAhr หรือมากกว่า ซึ่งจะสามารถรักษาข้อมูลได้นาน 10 ปี ที่อุณหภูมิ 25 องศาเซลเซียส

**SDA, SCL** (ขา5 และ 6) เป็นขาสำหรับเชื่อมต่อกับระบบไมโครคอนโทรลเลอร์บนระบบบัส $I^2C$

**SQW/OUT** (ขา 7) ที่ขานี้จะมีสัญญาณรูปสี่เหลี่ยมส่งออกมา โดยที่สามารถเลือกความถี่ได้ 1KHz, 4.096KHz, 8.192KHz และ 32KHz ในการใช้งานต้องต่อตัวต้านทาน 1k พูลอัปที่ขานี้ด้วย

**X1, X2** (ขา 1 และ 2 ) ใช้ต่อกับคริสตอลความถี่มาตรฐาน 32.768KHz เพื่อใช้เป็นฐานเวลาในการสร้างค่าเวลาจริง ในการใช้งานต้องต่อคริสตอลเข้ากับขาทั้งสองนี้และที่แต่ละขาต้องต่อตัวเก็บประจุค่าต่ำๆ ประมาณ 15pF คร่อมกับกราวด์ด้วย

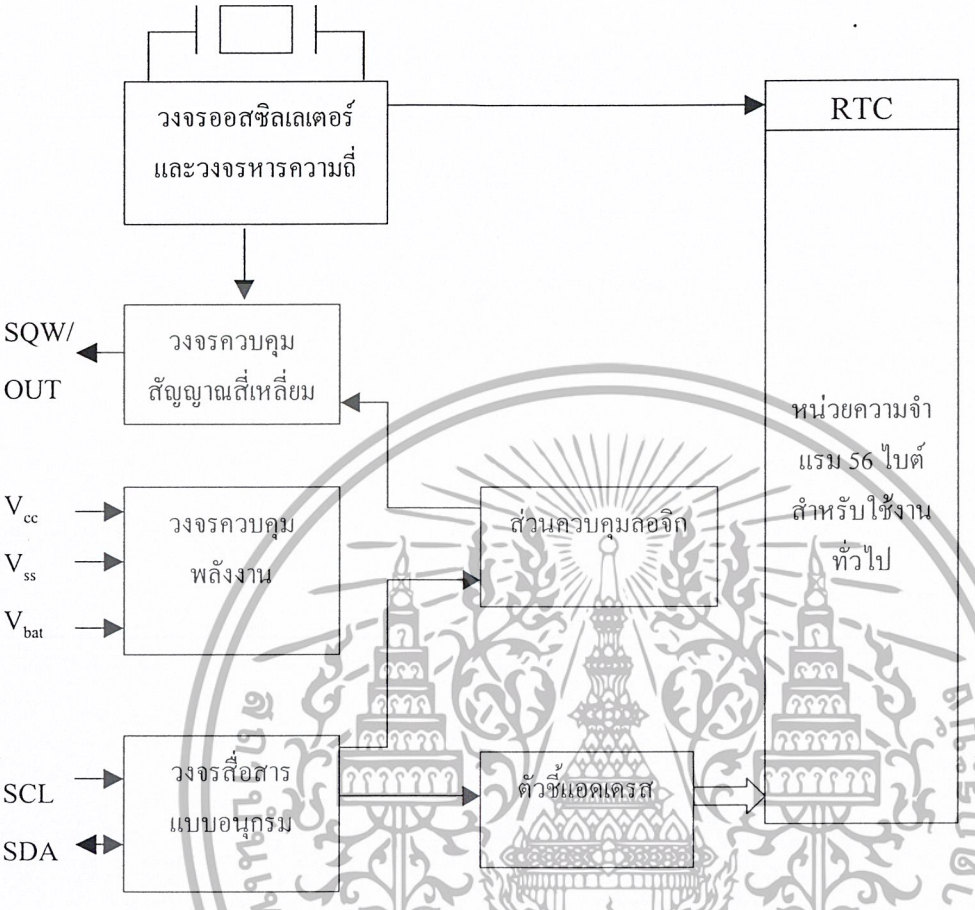
## 5.2 การทำงานของ DS1307

ไอซี DS1307 จัดการเชื่อมต่อในแบบบัส $I^2C$ โดยจะทำงานเป็นอุปกรณ์สเลฟเสมอ ดังนั้นการติดต่อเพื่อใช้งานจึงต้องกำหนดรูปแบบตามที่กำหนดไว้ในการติดต่อแบบ $I^2C$ ดังในรูปที่ 5.2 แสดงส่วนประกอบหลักที่สำคัญและไดอะแกรมการทำงานของ DS1307 วงจรออสซิลเลเตอร์ถือเป็นหัวใจหลักของไอซี อันเนื่องจากเป็นจุดเริ่มต้นของการสร้างข้อมูลเวลาจริง ในขณะที่ DS1307 ทำงานที่ขา SQW/OUT จะมีสัญญาณพัลส์สี่เหลี่ยมส่งออกมาตลอดเวลาในกรณีมีการอินาเบิล วงจรกำเนิดสัญญาณพัลส์ที่รีจิสเตอร์ควบคุม ค่าความถี่ของสัญญาณนี้สามารถเลือกได้ 4 ค่า พร้อมกันนั้นก็จะมีการเก็บค่าของเวลาไว้ในหน่อยความจำโวลาไทม์แรมซึ่งมีขนาดรวม 64 ไบต์ แต่จัดสรรให้ใช้เก็บข้อมูลเวลา 8 ไบต์ และเป็นหน่วยความจำใช้เก็บข้อมูลทั่วไปสำหรับผู้ใช้งานอีก 56 ไบต์

วงจรควบคุมพลังงานไฟฟ้าจะคอยตรวจสอบสภาวะของไฟเลี้ยงไอซี หากว่ามีไฟเลี้ยงต่ำกว่า $1.25*V_{BAT}$ ก็จะควบคุมให้ DS1307 หยุดการทำงาน และรีเซ็ตค่าตัวนับแอดเดรสภายใน ทำให้ไม่สามารถติดต่อกับ DS1307 ได้ดังนั้นในการใช้งาน DS1307 ต้องระมัดระวังอย่าให้ไฟเลี้ยงตกต่ำกว่า $1.25*V_{BAT}$ หรือประมาณ 3.75V ในกรณีที่ใช้ VBAT เท่ากับ 3V ถ้าหากไฟเลี้ยงมีค่าต่ำกว่าค่า $V_{BAT}$ ไอซี DS1307 จะเข้าสู่โหมดสำรองข้อมูลกระแสต่ำทันทีจะไม่มีการส่งสัญญาณพัลส์ออกมาที่ขา SQW/OUT แต่วงจรสร้างฐานเวลายังคงทำงานเพื่อให้ค่าของเวลาเดินไปอย่างไม่ผิดพลาด เมื่อมีไฟเลี้ยงปรากฏขึ้นอีกครั้ง DS1307 ก็จะสามารถให้ค่าของเวลาที่เป็นจริงแก่ผู้ใช้งานได้ต่อไป

วงจรสื่อสารอนุกรมภายใน DS1307 ได้รับการกำหนดให้ทำงานตามรูปแบบบัส $I^2C$ เป็นช่องทางการสื่อสารระหว่าง DS1307 กับอุปกรณ์มาสเตอร์ ผู้ใช้งานสามารถเข้าถึงหน่วยความจำที่ใช้เก็บค่าเวลาและหน่วยความจำใช้งานทั่วไปได้ โดยการเขียนข้อมูลตามรูปแบบที่กำหนดในระบบบัส $I^2C$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้




**รูปที่ 5.2 โครงสร้างภายในของไอซีสร้างฐานเวลาจริงเบอร์ DS1307**

### 5.3 การจัดสรรหน่วยความจำใน DS1307

รูปที่ 5.3 (ก) แสดงการจัดสรรพื้นที่ของหน่วยความจำภายใน DS1307 พื้นที่ 7 ไบต์แรกตั้งแต่แอดเดรส 00H-06H เป็นพื้นที่ของรีจีสเตอร์ค่าเวลาใช้ในการเก็บข้อมูลเกี่ยวกับเวลา ไบต์ต่อมาที่แอดเดรส 07H เป็นพื้นที่ของรีจีสเตอร์ควบคุมการทำงานของ DS1307 ในรูปที่ 5.3 (ข) แสดงรายละเอียดของรีจีสเตอร์ค่าเวลาและรีจิสเตอร์ควบคุมของ DS1307

ด้วยการจัดสรรพื้นที่แบบนี้ ทำให้ผู้ใช้งานสามารถเรียกข้อมูลเวลาออกมาได้ตามที่ต้องการ โดยไม่จำเป็นต้องอ่านออกมาทั้งหมดก็ได้ ค่าของเวลาทั้งหมดจะอยู่ในรูปของเลขฐานสิบ สำหรับการแสดงเวลาในรูปชั่วโมงสามารถเลือกได้ว่าต้องการแบบ 12 หรือ 24 ชั่วโมง โดยกำหนดที่บิต 6 ของแอดเดรส 02H และเมื่อเลือกแบบ 12 ชั่วโมง ที่บิต 5 ในแอดเดรสเดียวกันจะใช้ในการแสดงค่า AM/PM โดยถ้าบิตนี้เป็น "1" หมายถึง ค่าชั่วโมงในขณะนี้เป็นช่วงเวลาหลังเที่ยงวันในกรณีที่เป็นแบบ 24 ชั่วโมง บิตนี้จะใช้ในการแสดงค่า 2 ของหลักสิบในหนึ่งชั่วโมง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**รีจีสเตอร์ควบคุม**

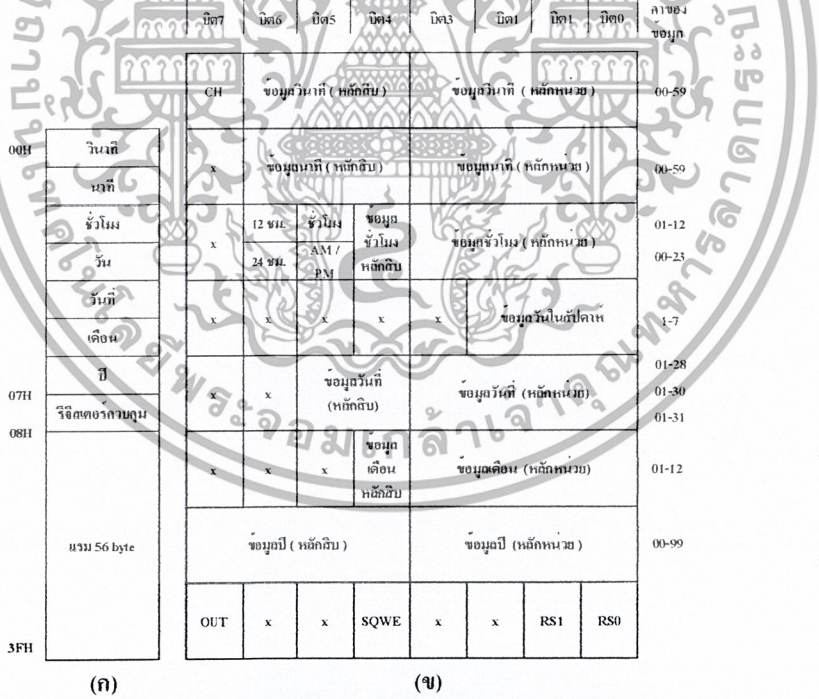
มีแอดเดรสอยู่ที่ 07H มีรายละเอียดแต่ละบิตดังนี้

**OUT** (Output control) : ใช้ในการควบคุมระดับลอจิกที่ขา SQW/OUT ในกรณีที่ดิสเอเบิลกำเนิดสัญญาณสี่เหลี่ยม โดยถ้าบิตนี้เป็น "1" ที่ขา SQW/OUT ก็จะเป็น "1" ถ้าบิตนี้เป็น "0" ที่ขา SQW/OUT ก็จะเป็น "1" ถ้าบิตนี้เป็น "0" ที่ขา SQW/OUT ก็จะเป็น "0"

**SQWE** (Square wave Enable) : ใช้ในการอีเนเบิลส่วนของวงจรกำเนิดสัญญาณสี่เหลี่ยมที่ขา SQW/OUT ถ้าต้องให้มีสัญญาณสี่เหลี่ยมออกให้กำหนดบิตนี้เป็น "1"

**RS1,RS0** (Rate Select) : ใช้เลือกความถี่ของสัญญาณสี่เหลี่ยมที่ออกจากขา SQW/OUT ดังมีรายละเอียดต่อไปนี้

| RS1 | RS0 | ค่าความถี่ของสัญญาณสี่เหลี่ยม |
|---|---|---|
| 0 | 0 | 1Hz |
| 0 | 1 | 4.096kHz |
| 1 | 0 | 8.192kHz |
| 1 | 1 | 32.768kHz |




**รูปที่ 5.3 (ก) การจัดสรรหน่วยความจำแรมภายใน DS1307**

**(ข) รายละเอียดของรีจีสเตอร์เก็บค่าเวลาและรีจีสเตอร์ควบคุม DS1307**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**5.4 โหมดการทำงานของ DS1307**

มีอยู่ด้วยกัน 2 โหมดคือ โหมดการเขียนข้อมูลและโหมดการอ่านข้อมูล ซึ่งในการใช้งาน DS1307 ตามปกติจะเป็นการใช้งานเฉพาะโหมดอ่านข้อมูลเท่านั้นเนื่องจากไมโครคอนโทรลเลอร์จะติดต่อกับ DS1307 เพื่ออ่านข้อมูลของเวลาไปใช้งาน โหมดการเขียนข้อมูลจะถูกใช้งานก็ต่อเมื่อการต้องการตั้งค่าเวลาใหม่และต้องการเขียนข้อมูลลงในหน่วยความจำใช้งานทั่วไป อย่างไรก็ตามเมื่อเริ่มต้นติดต่อกับ DS1307 จำเป็นอย่างยิ่งที่จะต้องเข้าสู่โหมดการเขียนข้อมูลก่อน เพื่อกำหนดแอดเดรสที่ต้องการอ่านข้อมูล จากนั้นจึงเปลี่ยนโหมดการทำงานมาเป็นโหมดการอ่านข้อมูล

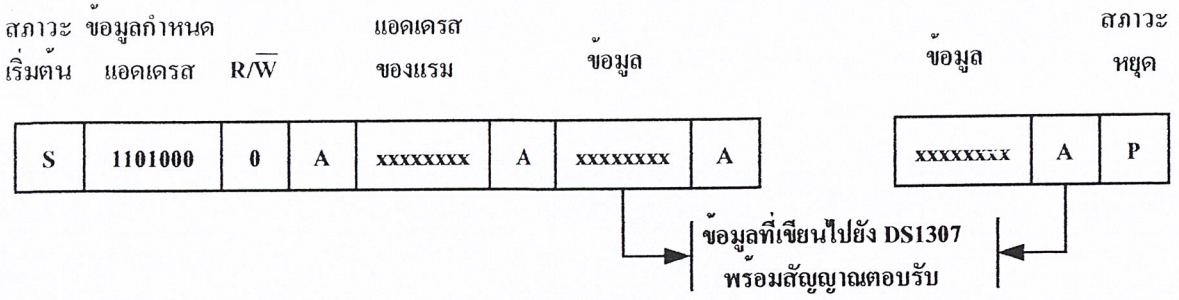
**โหมดการเขียนข้อมูล**

มีรูปแบบดังในรูปที่ 5.4 เริ่มต้นเมื่อไมโครคอนโทรลเลอร์จะทำการกำหนดสภาวะเริ่มต้น (START :S) จากนั้นส่งข้อมูลกำหนดแอดเดรส 1101000 ตามด้วยข้อมูลเลือกการเขียน นั่นคือค่า 0 จากนั้นจะรอการตอบรับจาก DS1307 ขั้นตอนต่อมาคือ ส่งข้อมูลเพื่อเลือกแอดเดรสที่ต้องการเขียน จากนั้นรอการตอบรับจาก DS1307 เมื่อมีการตอบรับมาเรียบร้อย ก็เริ่มทยอยเขียนข้อมูลลงไปครั้งละแอดเดรสหลังจากเขียนข้อมูล ในแต่ละแอดเดรสจะต้องหยุดรอการตอบรับจาก DS1307 ทุกครั้งจึงสามารถเขียนข้อมูลต่อไปได้ เมื่อเขียนเรียบร้อยแล้วให้ส่งสภาวะหยุด (STOP :P) เป็นอันสิ้นสุดกระบวนการเขียนข้อมูล

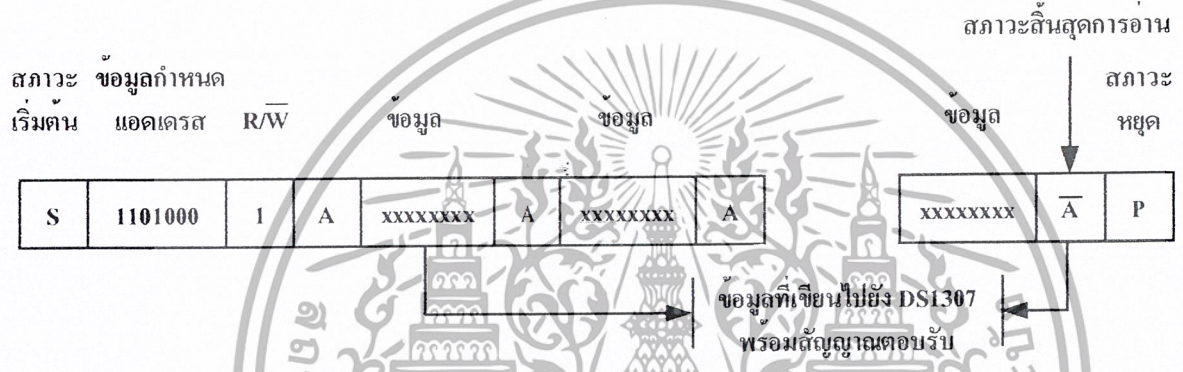
**โหมดการอ่านข้อมูล**

มีรูปแบบแสดงดังรูป P5.5 เริ่มต้นการทำงานจะเหมือนกับโหมดการเขียนข้อมูลคือ ไมโครคอนโทรลเลอร์จะกำหนดสภาวะเริ่มต้นแล้วส่งข้อมูลกำหนดแอดเดรส ตามด้วยข้อมูลเลือกการอ่าน ซึ่งเท่ากับ 1 จากนั้นรอการตอบรับจาก DS1307 เมื่อมีการตอบรับเรียบร้อย DS1307 จะทยอยส่งข้อมูลออกมาให้ไมโครคอนโทรลเลอร์คราวละ 1 แอดเดรสหรือ 1 ไบต์ โดยแอดเดรสที่เลือกอ่านข้อมูลจะต้องมีการกำหนดมาก่อนล่วงหน้าด้วยโหมดการเขียนข้อมูล วิธีง่ายๆ คือ การเข้าสู่โหมดการเขียนข้อมูลก่อน เมื่อถึงจังหวะที่ต้องการเขียนข้อมูลให้ทำการสร้างสภาวะเริ่มต้นก่อนและส่งข้อมูลกำหนดแอดเดรสใหม่อีกครั้ง ตามด้วยเลือกโหมดการอ่านข้อมูล ข้อมูลที่ออกจาก DS1307 ก็จะเป็นข้อมูลจากแอดเดรสที่กำหนดไว้ก่อนหน้านี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้




รูปที่ 5.4 รูปแบบของข้อมูลสำหรับติดต่อกับ DS1307 ในโหมดการเขียนข้อมูล




รูปที่ 5.5 รูปแบบของข้อมูลสำหรับติดต่อกับ DS1307 ในโหมดการอ่านข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# บทที่ 6
# การสื่อสารข้อมูลแบบอนุกรม

การเชื่อมต่อระหว่างพอร์ตอนุกรมแต่ละพอร์ตจะใช้เส้นสัญญาณแบบอนุกรมมาตรฐานอยู่ 3 แบบ คือ มาตรฐาน EIA RS-422 ,ระบบวงรอบกระแส และมาตรฐาน EIA RS-232 โดยทั่วไปเราจะใช้เส้นส่งสัญญาณอนุกรมแบบมาตรฐาน EIA RS-232 มากที่สุด ซึ่งเราจะเรียกว่า RS-232 สายส่งสัญญาณ RS-232 นี้ ได้ถูกนำไปใช้ในหน่อยแสดงผล เครื่องพิมพ์โมเดม และอุปกรณ์อื่นๆ ซึ่งจะมีความยาวของสายไม่เกิน 50 ฟุต

## 6.1 มาตรฐาน EIA RS-232

มาตรฐาน EIA RS-232 ซึ่งเราจะเรียก RS-232 ได้กำหนดให้ค่าสัญญาณไฟฟ้าที่ระดับของศักดาไฟฟ้าเท่ากับ 3 โวลท์หรือสูงกว่า มีค่าทางตรรกะเป็น 1 และกำหนดให้ค่าสัญญาณไฟฟ้าที่มีระดับศักดาไฟฟ้าเท่ากับ -3 โวลท์หรือต่ำกว่า มีค่าทางตรรกะเป็น 0 วงจรไอซีที่ใช้สร้างสัญญาณเหล่านี้ต้องการแหล่งจ่ายไฟขนาด +12 V RS-232 จะใช้สาย 1 เส้น สำหรับส่งข้อมูลและใช้สาย อีก 1 เส้นสำหรับรับข้อมูล โดยสัญญาณในแต่ละสายนี้จะถูกอ้างอิงเทียบกับกราวนด์ ( ขาเบอร์7 ) มาตรฐาน RS-232 นี้ยังได้กำหนด สัญญาณตอบรับเพื่อ ใช้ในการควบคุมการรับส่งข้อมูลด้วย

มาตรฐาน RS-232 จะสามารถใช้ส่งข้อมูลได้ไกลที่สุด 50 เมตรด้วยอัตราการรับส่งข้อมูล 9600 บอด แต่ถ้าเราต้องการให้ส่งได้ไกลกว่านี้เราต้องส่งข้อมูลด้วยอัตราส่งข้อมูลที่ช้ากว่านี้และถ้าต้องการส่งข้อมูลในระยะทางที่ไกลกว่านี้ เราสามารถส่งข้อมูลด้วยอัตราที่มากกว่าอัตราการรับส่งข้อมูล 9600 บอดในที่นี้จะกล่าวถึงพอร์ตสื่อสารอนุกรม RS-232C

**ตารางที่ 6.1 ตัวเชื่อมต่อที่นิยมใช้สายส่งสัญญาณอนุกรมแบบมาตรฐาน RS-232**

| 9 Pin | EIA RS-232 Circuit | CCIT V.24 Circuit | RS-232 Description | Signal type & Direction |
|---|---|---|---|---|
| 5 | AB | 102 | Signal group/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 |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

| | | | | |
|---|---|---|---|---|
| 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 |

**(ก) ตัวเชื่อมต่อ DB-9**

| 25 Pin | EIA RS-232 Circuit | CCIT V.24 Circuit | RS-232 Description | Signal type & 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 |
| 17 | DD | 115 | (DCE) 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 |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

| | | | | |
|---|---|---|---|---|
| 12 | SCF | 122 | Secondary received line signal detector | Control from DCE |
| 11<br>18<br>25 | | | Undefined | |

**(ข) ตัวเชื่อมต่อ DB-25**

**6.2 RS-232C**

**1. คุณสมบัติของ RS-232C**

อัตรารับส่งข้อมูล : 0 - 20000 บิต/วินาที

ระดับแรงดันเอาต์พุตสูงสุดในสภาวะไม่มีโหลด : -25 โวลท์ (ลอจิก 1)

: +25 โวลท์ (ลอจิก 0)

ระดับแรงดันเอาต์พุตสำหรับโหลด 3 - 7กิโลโอห์ม : ลอจิก"1" -15โวลท์(7กิโลโอห์ม)

-5 โวลท์(3กิโลโอห์ม)

ลอจิก"0" +15โวลท์(7กิโลโอห์ม)

+5 โวลท์(3กิโลโอห์ม)

กระแสเอาต์พุตเมื่อลัดวงจร : สูงสุด 500 มิลลิแอมป์

เอาต์พุตอิมพีแดนซ์เมื่อไม่จ่ายไฟเลี้ยง : ต่ำสุด 300 โอห์ม

สลูว์เรตทางเอาต์พุตสูงสุด : 30 โวลท์/ไมโครวินาที

ความต้านทานอินพุตของภาครับ : สูงสุด 7 กิโลโอห์ม

ต่ำสุด 3 กิโลโอห์ม

ค่าความจุอินพุตของภาครับ : สูงสุด 2500 กิโลฟารัด

ย่านแรงดันอินพุตของภาครับ : -25 โวลท์ถึง +25 โวลท์

ค่าความจุอินพุตภาครับ : สูงสุด 2500 กิโลฟารัด

**ย่านแรงดันอินพุตของภาครับ** : -25 โวลท์ถึง +25 โวลท์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**2. การจัดขาสัญญาณของ RS-232C**

มีด้วยกัน 2 แบบคือ แบบ 9 ขา และ แบบ 25 ขา ดังแสดงในตารางที่ 6.2

**ตารางที่ 6.2 ขาสัญญาณ RS-232 ทั้งแบบ 9 ขาและ 25 ขา**

| ชื่อสัญญาณ | หมายเลขขาในแบบ 9 ขา | หมายเลขขาในแบบ 25 ขา |
|---|---|---|
| TD Transmitted Data | 3 | 2 |
| RD Received Data | 2 | 3 |
| RTS Request to Send | 7 | 4 |
| CTS Clear to Send | 8 | 5 |
| DSR Data Set Ready | 6 | 6 |
| SG Signal Ground | 5 | 7 |
| CD Carrier Detect | 1 | 8 |
| DTR Data terminal Ready | 4 | 20 |
| RI Ring Indicator | 9 | 22 |

**3. การเชื่อมต่อสัญญาณของ RS-232C**

มีลักษณะเชื่อมต่อ 2 แบบด้วยกัน

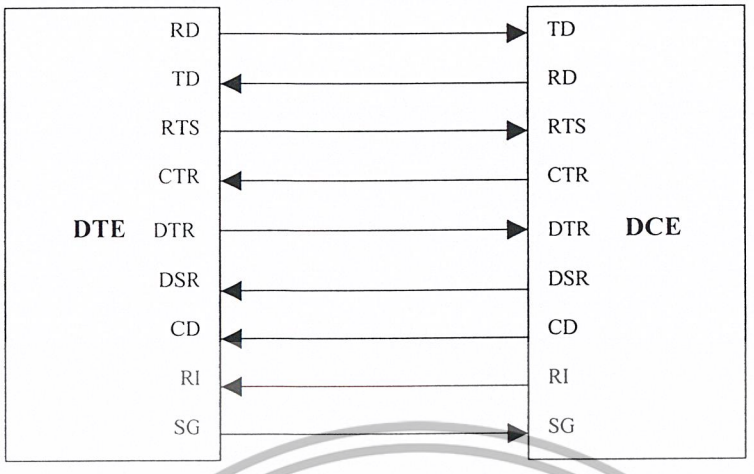
1. การเชื่อมต่อกันระหว่างอุปกรณ์ DTE (Data Terminal Equipment) เช่น คอมพิวเตอร์กับอุปกรณ์ DCE (Data Circuit Terminal) เช่น โมเดม แสดงดังรูปที่ 6.3

## 6.3 รูปแบบของข้อมูลอนุกรมและอัตราบอดในการสื่อสารข้อมูลอนุกรม

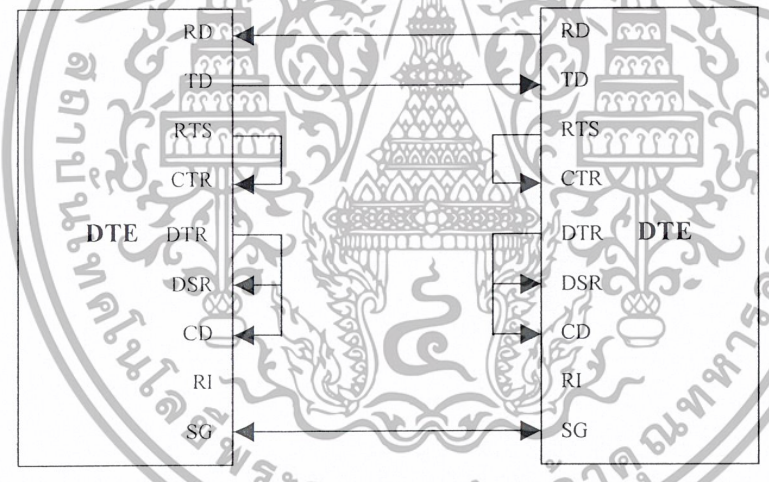
อัตราบอด (baud rate) คือ ความเร็วในการรับส่งข้อมูลอนุกรม มีหน่วยเป็นบิตต่อวินาที

เมื่อเราไม่ต้องการให้อุปกรณ์ภายนอกเชื่อมต่อโดยตรงให้กับระบบที่มีไมโครโปรเซสเซอร์เป็นพื้นฐาน เราจะต้องนำ opt isolator มาใช้เป็นตัวแยก opt isolator จะช่วยป้องกันไมโครโปรเซสเซอร์ UART และวงจรอื่นๆในระบบจากศักดาไฟฟ้าสูงๆ หรือป้องกันไฟฟ้าไม่ให้เข้าที่สายกราวนด์ ซึ่งอาจเกิดจากอุบัติเหตุ จึงทำให้มีการเชื่อมต่อระหว่างสายส่งสัญญาณกับสายกราวด์โดยเราอาจพบเหตุการณ์เช่นนี้ได้ในงานด้านอุตสาหกรรมดังในรูปที่ 6.5 ซึ่งแสดงโครงสร้างของตัวรับสัญญาณและตัวส่งสัญญาณในระบบ EIA- RS-232 ซึ่งมีการนำ optoisolator มาใช้นอกจากนี้ยังได้มีการนำ optisolator มาใช้ในการส่งสัญญาณแบบขนาน และในการสื่อสารรูปแบบอื่นๆในไมโครโปรเซสเซอร์ด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้




(ก)



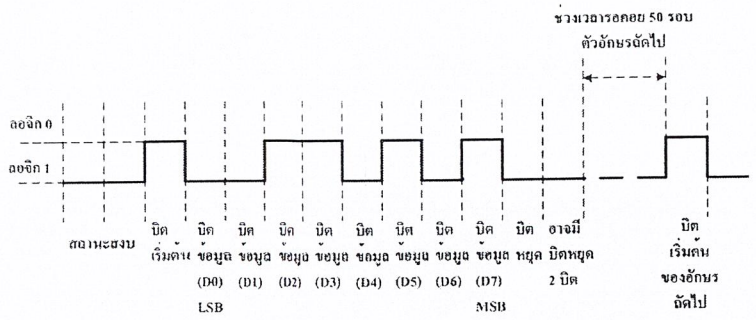

(ข)

**รูปที่6.3 การเชื่อมต่อสัญญาณของมาตรฐาน RS-232**

**(ก) การต่ออุปกรณ์ DTE เข้ากับ DCE**

**(ข) การต่ออุปกรณ์ DTE เข้ากับ DTE**

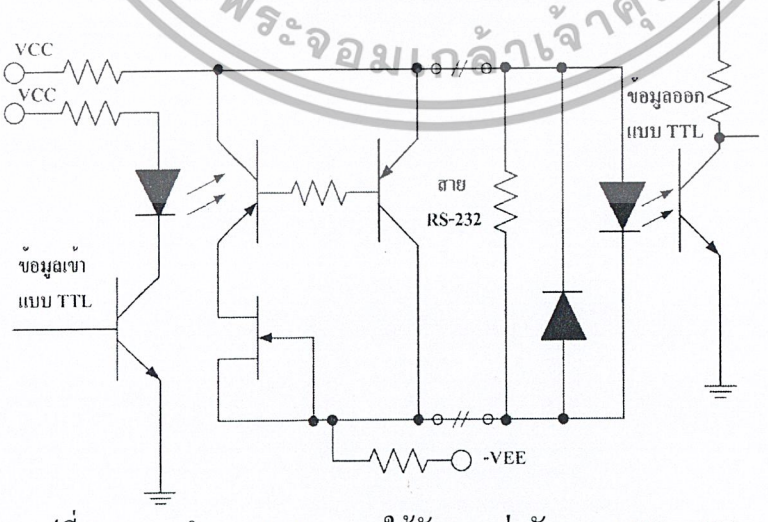
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้




**รูปที่ 6.4 รูปแบบของสัญญาณข้อมูลอนุกรมที่ใช้ในการสื่อสารข้อมูลอนุกรม**

**ตารางที่ 6.5 อัตราบอดและช่วงเวลาของแต่ละบิตข้อมูลในการสื่อสารข้อมูลอนุกรม**

| อัตราบอด | ช่วงเวลาของแต่ละบิต |
|---|---|
| 110 | 9.91 ms |
| 150 | 6.67 ms |
| 300 | 3.33 ms |
| 600 | 1.67 ms |
| 1200 | 0.833 ms |
| 2400 | 0.417 ms |
| 4800 | 0.208 ms |
| 9600 | 0.104 ms |
| 19200 | 0.052 ms |

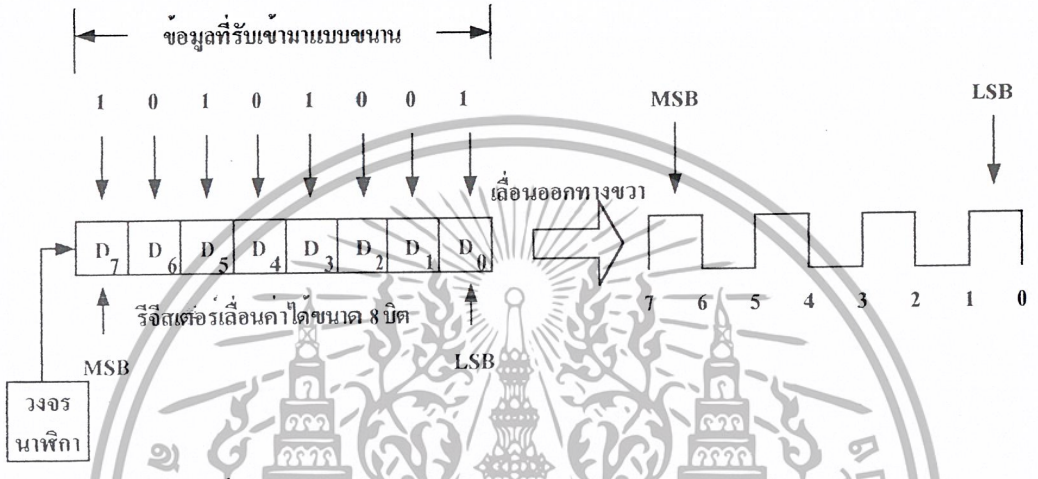



**รูปที่ 6.6 การนำ optisolator มาใช้กับสายส่งสัญญาณ RS – 232**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**6.4 การเชื่อมต่อแบบอนุกรมและ UART**

ในรูปที่ 6.7 แสดงถึงการแปลงข้อมูลแบบขนานไปเป็นข้อมูลแบบอนุกรม โดยที่เริ่มแรกข้อมูลแบบขนานจะถูกเก็บไว้ในรีจิสเตอร์ Shift Register จากนั้นจะใช้สัญญาณนาฬิกาในการเลื่อนค่าในรีจีสเตอร์ออกมาทีละบิต (โดยเลื่อนค่าไปทางขวามือ) บิตแรกที่ถูกเลือกออกมาคือบิต LSB ของข้อมูลและบิตที่สองที่ถูกเลื่อนออกมาก็คือบิตที่อยู่ถัดจากบิต LSB และบิตต่อไป สำหรับบิตสุดท้ายที่ถูกเลื่อนออกมาก็คือบิตMSBของข้อมูล



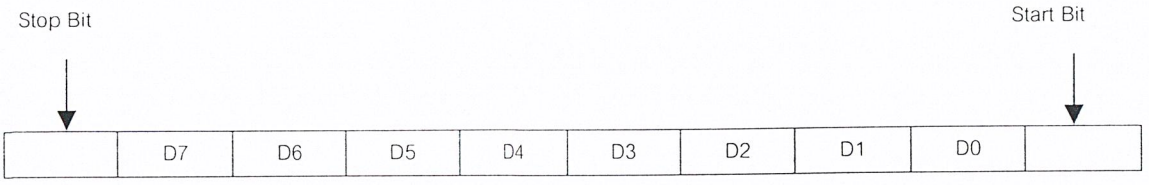

**รูปที่ 6.7 การแปลงข้อมูลแบบขนานเป็นอนุกรม**

การแปลงข้อมูลแบบอนุกรมไปเป็นข้อมูลขนานนั้นจะมีขั้นตอนตรงกันกับที่กล่าวมา นั่นคือข้อมูลแบบอนุกรมจะถูกเลื่อนเข้าไปเก็บใน Shift Register โดยใช้สัญญาณนาฬิกาเป็นตัวควบคุม และหลังจากที่ได้มีการเลื่อนข้อมูลทุกบิตเข้าใน Shift Register ได้หมดแล้ว ข้อมูลในรีจิสเตอร์นี้จะถูกนำออกมาแบบขนานเพื่อนำไปใช้งานต่อไป

อุปกรณ์ที่ทำหน้าที่ในการแปลงข้อมูลแบบอนุกรมเป็นข้อมูลแบบขนาน และแปลงข้อมูลจากแบบขนานไปเป็นข้อมูลแบบอนุกรม เราเรียกว่า UART ( Universal Asynchronous Receiver - Transmitter ) ซึ่งเป็นวงจร LSI ซึ่งนอกจากจะมีหน้าที่ในการแปลงข้อมูลแล้ว UART ยังมีหน่วยควบคุมและหน่วยตรวจสอบการทำงานอีกด้วย

ในการส่งข้อมูลขนาด 8 บิตแบบอนุกรมนี้จะต้องมีบิตสตาร์ท ( Start Bit ) และบิตสต๊อป( Stop Bit ) เพิ่มเข้ามา ซึ่งจะทำให้ข้อมูลที่ถูกส่งไปจริงๆนั้นมีขนาด 10 บิต ในรูปที่ 6.8 แสดงเวิร์ดข้อมูลที่มีบิตข้อมูล 8 บิต ประกอบด้วย บิตสตาร์ท 1 บิต และบิตสต๊อป 1 บิต โดยที่บิตสตาร์ทมีค่า 0 บอก UART ที่รับข้อมูลให้รู้ว่ามีข้อมูลกำลังเข้ามา และบิตสต๊อปที่มีค่า 1 จะบอก UART ที่รับข้อมูลให้รู้ว่าการส่งข้อมูลได้เสร็จสิ้นลง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้




**รูปที่ 6.8 เวิร์ดข้อมูลขนาด 8 บิตกับ Start Bit และ Stop Bit ที่ใช้ในการโอนย้ายข้อมูลแบบอนุกรม**

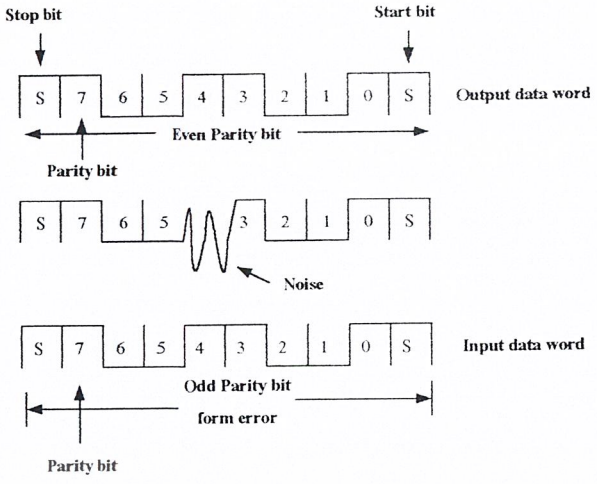
**ตารางที่ 6.9 อัตราบอดทั่วไปที่ใช้ในการโอนย้ายข้อมูลแบบอนุกรม**

| อัตราบอด | ไบต์/วินาที |
|---|---|
| 110 | 10 |
| 150 | 15 |
| 300 | 30 |
| 600 | 60 |
| 1200 | 120 |
| 2400 | 240 |
| 4800 | 480 |
| 9600 | 960 |
| 19200 | 1920 |
| 38400 | 3840 |

การส่งข้อมูลด้วยอัตราบอด 110 นั้นจะแตกต่างไปจากอัตราบอดอื่น ซึ่งอัตราบอดนี้ต้องใช้บิตสตาร์ท 1 บิตและบิตสต๊อป 2 บิต ดังนั้นจึงทำให้ต้องส่งข้อมูลขนาด 11 บิต

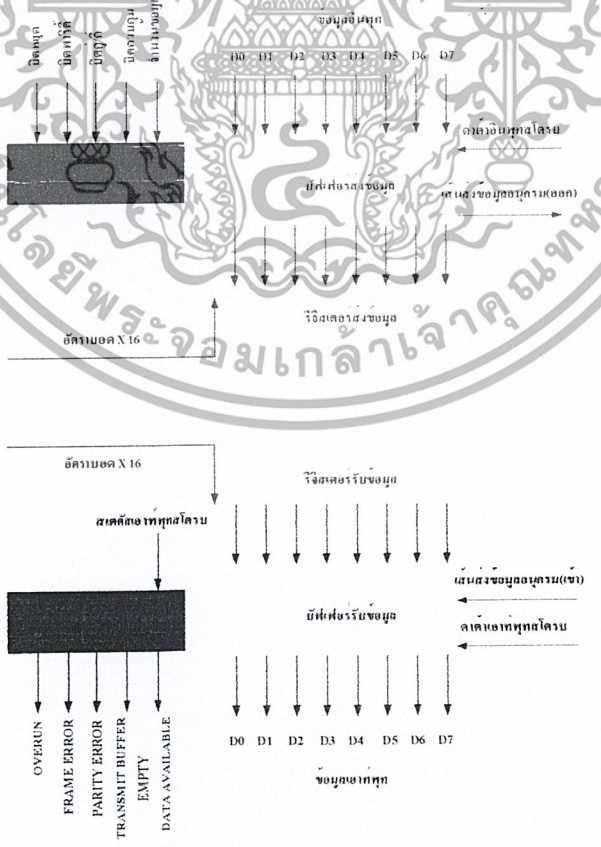
ในรูปที่ 6.10 แสดงถึงการส่งข้อมูลแบบอนุกรมที่มีคาพารีตี้เปลี่ยนไปอันเนื่องจากสัญญาณรบกวน (noise) โดยบิตที่ 4 ในรูปจะ ถูกสัญญาณรบกวนทำให้มีค่าจาก 1 ไปเป็น 0 ซึ่งทำให้ข้อมูลที่ได้รับนั้นไม่ถูกต้อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



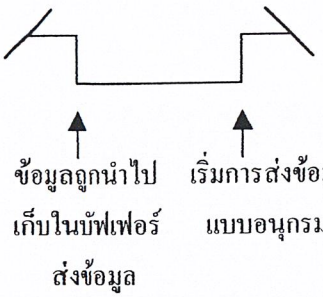

**รูปที่ 6.10** การใช้บิตพาริตี้เพื่อตรวจสอบความผิดพลาดในการโอนย้ายข้อมูลแบบอนุกรม

ในการส่งข้อมูลของ UART แยกออกเป็น 2 ส่วนซึ่งได้แก่ บัฟเฟอร์ส่งข้อมูล (Transmitter data output buffer) กับรีจิสเตอร์ส่งข้อมูล (Transmitter register) โดยบิตข้อมูล 8 บิต จะถูกนำไปเก็บในบัฟเฟอร์ส่งข้อมูล เมื่อสัญญาณที่ควบคุมคาต้าอินพุตสโตรบ (Data input strobe) เปลี่ยนค่าจาก 1 เป็น 0 และการส่งข้อมูลแบบอนุกรมจะเริ่มขึ้นเมื่อสัญญาณควบคุมนี้เปลี่ยนค่าจาก 0 เป็น 1 ดังรูปที่ 6.11




**รูปที่ 6.11** บล็อกไดอะแกรมของ UART

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้




**รูปที่ 6.12 สัญญาณดาต้าอินพุตสโตรบ**

เราจะสามารถนำข้อมูลในรีจีสเตอร์สถานะของ UART ไปใช้ได้ โดยการส่งสัญญาณควบคุมสเตตัสเอาท์พุตสโตรบ (Status output strobe) ซึ่งในรีจีสเตอร์สถานะของ UART จะมีบิตบอกสถานะต่างๆดังนี้

**1.บิต OR** (Overrun) บิตนี้จะมีค่าเป็น 1 เมื่อข้อมูลชิ้นใหม่เข้ามาทับข้อมูลชั้นเดิมที่เก็บอยู่ ซึ่งข้อมูลชิ้นเดิมยังไม่ได้ถูกนำไปเก็บในบัฟเฟอร์รับข้อมูล

**2.บิต FE** (Framing error) บิตนี้จะมีค่าเป็น 1 ถ้า UART ไม่พบบิตสต๊อป ซึ่งอาจเป็นเพราะว่า UART ไม่ได้อ่านบิตสตาร์ทจากตำแหน่งที่ถูกต้อง ซึ่งหมายความว่า UART เริ่มทำงานกับบิตที่ไม่ใช่บิตสตาร์ท

**3.บิต PE** (Parity error) บิตนี้จะมีค่าเป็น 1 เมื่อ UART ได้ทำการตรวจสอบพบว่าข้อมูลที่ได้รับมีพาริตี้ไม่ตรงกับค่าในพาริตี้ของข้อมูล เราจะกำหนดให้ UART ทำการทดสอบค่าแบบพาริตี้คู่หรือพาริตี้คี่ได้ โดยระบุไว้ที่วงจรควบคุมก่อนที่จะนำ UART มาทำการรับข้อมูล

**4.บิต TBE** (Transmit buffer empty) บิตนี้จะมีค่าเป็น 1 เมื่อ UART ได้ส่งข้อมูลออกไปแล้ว ซึ่งทำให้เราสามารถนำข้อมูลชิ้นต่อไปที่จะส่งเก็บลงในบัฟเฟอร์ส่งข้อมูลได้

**5.บิต DA** (Data available) บิตนี้จะมีค่าเป็น 1 เมื่อ UART ได้รับข้อมูลชิ้นใหม่เข้ามาและสามารถทำการอ่านข้อมูลนี้ได้ โดยส่งสัญญาณดาต้าเอาท์พุตสโตรบ

## 6.5 MCS-51 กับการรับส่งข้อมูลอนุกรม

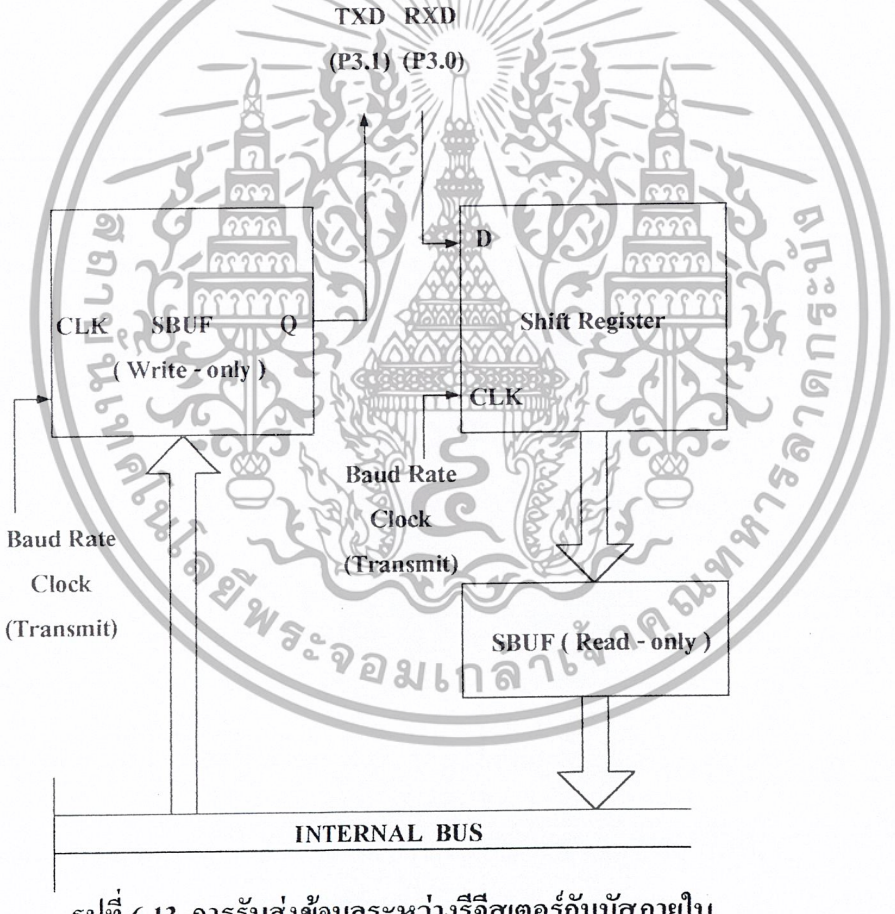
การรับส่งข้อมูลแบบอนุกรมกับไมโครคอนโทรลเลอร์ MCS-51 นั้น ภายในชิพ MCS-51 จะมี UART อยู่ในตัว ซึ่งเป็นข้อดีของไมโครคอนโทรลเลอร์ โดยพอร์ตอนุกรมของ MCS-51 จะใช้ขา TxD และ RxD ในการรับส่งข้อมูล โดยขาทั้งสองจะอยู่ในพอร์ต 3 คือ P3.1 หรือขา 11 เป็น TxD และ P3.0 หรือขา 10 เป็น RxD พอร์ตอนุกรมของ MCS-51 สามารถทำงานแบบ Full Duplex

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ได้คือ สามารถส่งข้อมูลและรับข้อมูลในเวลาเดียวกันได้ โดยในการรับและส่งข้อมูล จะมีบัฟเฟอร์สำหรับเก็บข้อมูลให้ใช้

รีจิสเตอร์ที่สำคัญในการรับส่งข้อมูลคือ รีจีสเตอร์ SBUF และรีจีสเตอร์ SCON ซึ่งเป็นรีจีสเตอร์ที่อยู่ใน Special Function Registers โดยที่รีจีสเตอร์ Serial Port Buffer (SBUF) จะอยู่ในตำแหน่ง 99H ถ้าเขียนข้อมูลไปที่ตำแหน่งนี้จะเป็นการส่งข้อมูลออกทางพอร์ตอนุกรม และถ้าอ่านข้อมูลจากตำแหน่งนี้ จะเป็นการรับข้อมูลจากพอร์ตอนุกรม

สำหรับที่รีจีสเตอร์ Serial Port Control Register (SCON) ซึ่งอยู่ที่ตำแหน่ง 98H จะเป็นรีจีสเตอร์ที่สามารถเข้าถึงข้อมูลระดับบิตได้ รีจีสเตอร์นี้จะทำหน้าที่ในการควบคุมและบอกสถานะต่างๆของภาครับส่งข้อมูลแบบอนุกรม สำหรับความเร็วของการส่งข้อมูล (Baud Rate) สามารถหาได้จากการหารสัญญาณนาฬิกาที่ใช้กับ MCS-51




**รูปที่ 6.13 การรับส่งข้อมูลระหว่างรีจีสเตอร์กับบัสภายใน**

### 6.6 Serial Port Control Register

MCS-51 มีโหมดการทำงานของพอร์ตอนุกรมหลายโหมด ซึ่งสามารถโปรแกรมโหมดการทำงานได้โดยการเขียนข้อมูลควบคุมไปยังรีจีสเตอร์ SCON โดยที่ความหมายของแต่ละบิตจะแสดงได้ดังตารางที่ 6.14 และ 6.15

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**ตารางที่ 6.14 บิตต่างๆ ของรีจีสเตอร์ SCON**

| บิต | ชื่อ | ตำแหน่ง | ความหมาย |
|---|---|---|---|
| SCON.7 | SM0 | 9FH | บิตเลือกโหมด 0 |
| SCON.6 | SM1 | 9EH | บิตเลือกโหมด1 |
| SCON.5 | SM2 | 9DH | บิตเลือกโหมด2 |
| SCON.4 | REN | 9CH | บิตแฟลกกำหนดยอมให้มีการรับข้อมูล |
| SCON.3 | TB8 | 9BH | ค่าของบิต 9 สำหรับการส่งข้อมูล ในโหมด 2 และ 3 สามารถ set และ clear ได้โดย software |
| SCON.2 | RB8 | 9AH | ค่าของบิต 9 เมื่อรับข้อมูลเข้ามา |
| SCON.1 | TI | 99H | บิตแฟลกแสดงการอินเตอร์รัพท์ ภายหลังจากส่งข้อมูลออกไป โดยจะ set เมื่อส่งข้อมูลออกไปหมดแล้ว และสามารถ clear ได้ด้วย software |
| SCON.0 | RI | 98H | แฟลกแสดงการอินเตอร์ รัพท์ภายหลังรับข้อมูลเข้ามาสามารถ set ได้ด้วย software |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**ตารางที่ 6.15 โหมดต่างๆ ของการรับส่งแบบอนุกรม**

| SM0 | SM1 | MODE | ความหมาย | BAUD RATE |
|---|---|---|---|---|
| 0 | 0 | 0 | Shift Register | เปลี่ยนแปลงไม่ได้(Oscillator Frequency/12) |
| 0 | 1 | 1 | 8 – bit UART | สามารถเปลี่ยนแปลงได้โดยกำหนดจาก Timer |
| 1 | 0 | 2 | 9 – bit UATR | เปลี่ยนแปลงไม่ได้(Oscillator Frequency/12หรือ /64) |
| 1 | 1 | 3 | 9 – bit UART | สามารถเปลี่ยนแปลงได้โดยกำหนดจาก Timer |

ก่อนจะใช้พอร์ตอนุกรมจะต้องโปรแกรมให้กับ SCON เสียก่อนเพื่อเป็นการกำหนดโหมดการทำงานและลักษณะต่างๆ เช่น

MOV SCON,#01010010B

เป็นการกำหนดให้พอร์ตอนุกรมทำงานในโหมด 1 และอีนาเบิ้ลให้มีการรับรู้ข้อมูลพร้อมกับกำหนดให้ TI เป็น 1

ในการส่งข้อมูลทุกโหมดสามารถทำได้โดยเขียนไปยัง SBUF เมื่อข้อมูลถูกส่งไปแล้ว บิต TI จะถูก Set ให้เป็น "1" ในการส่งข้อมูล จะต้องคอยตรวจสอบบิต TI เพราะว่าถ้า TI ยังไม่เป็น "1" แสดงว่าข้อมูลยังส่งไปไม่หมด ถ้าหากมีการเขียนข้อมูลไบต์ต่อไปไปยัง SBUF จะทำให้เกิดข้อผิดพลาดขึ้น สำหรับในการรับข้อมูลบิต REN จะต้อง Set ให้เป็น "1" ยกเว้นโหมด 0 เพื่ออนุญาติให้รับข้อมูลได้ เมื่อข้อมูลรับเข้ามาเรียบร้อยแล้วบิต RI จะถูก Set เป็น "1"

## 6.7 Mode of Operation

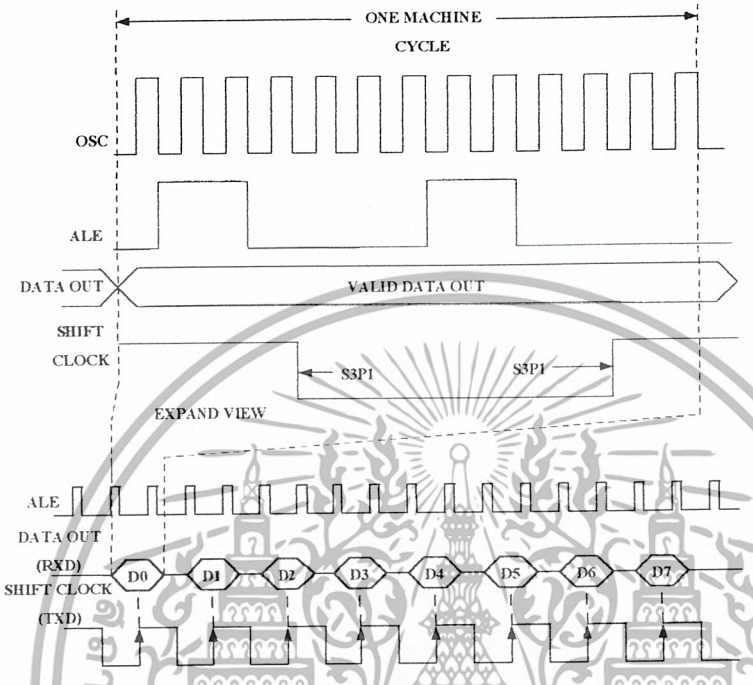
ใน MCS-51 การสื่อสารทางพอร์ตอนุกรมจะมีอยู่ 4 ประเภท หรือ 4 โหมดซึ่งจะกำหนดได้ที่บิต SM0 และ SM1 ใน SCON โดยจะมีทั้งสิ้น 3 โหมด เป็นการสื่อสารแบบ Asynchronous โดยลักษณะของข้อมูลที่ส่ง จะมีบิตเริ่มต้น (Start Bit) และบิตจบ (Stop Bit) ซึ่งคล้ายกับการสื่อสารแบบ RS-232 ในระบบคอมพิวเตอร์ อีกโหมดหนึ่งจะเป็นการใช้พอร์ตอนุกรมในลักษณะชิพรีจีสเตอร์

### 1. 8 - Bit Shift Register (Mode 0)

การทำงานในโหมดนี้จะใช้ขา RxD ในการรับส่งข้อมูลโดยต่อกับ Shift Register ภายนอก ส่วนขา TxD จะเป็น Output Shift Clock เพื่อกระตุ้นรีจีสเตอร์ภายนอกให้เลื่อนบิต ถ้ามีการส่งข้อมูลหรือรับข้อมูล 8 บิต จะเริ่มต้นที่บิตต่ำสุดก่อน โดยมีค่า Baud Rate เท่ากับ 1/12 ของความถี่ที่ใช้บนชิพ

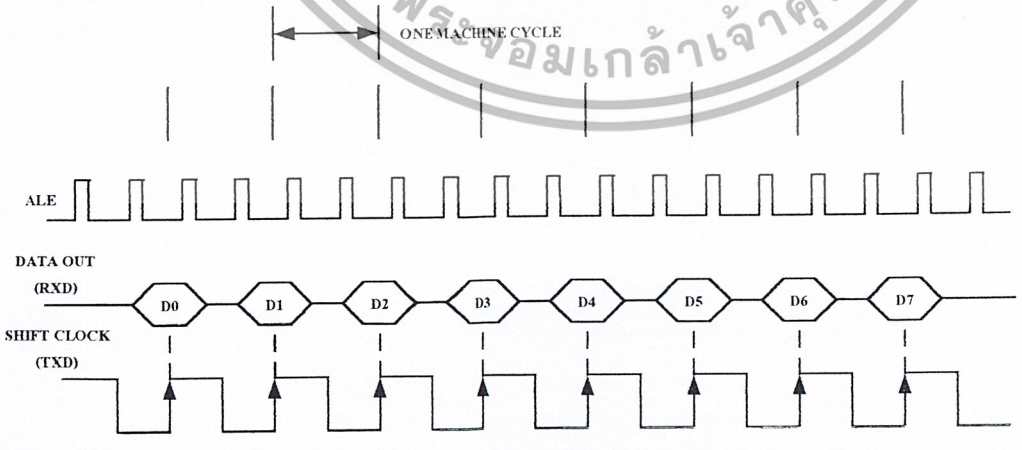
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในการส่งข้อมูลจะทำโดยการเขียนข้อมูลไปที่รีจีสเตอร์ SBUF ข้อมูลจะถูกส่งออกมาทางขา RxD (P3.0) โดยจะสอดคล้องกับสัญญาณที่ออกมาทางขา TxD ซึ่งสัญญาณของขา TxD จะถูกส่งออกมาทุกๆ Machine Cycle โดยจะเป็นลอจิก "0" ใน S3P1 และกลับเป็นลอจิก "1" ใน S6P1 ซึ่งแสดงได้ดังรูปที่ 6.16




**รูปที่ 6.16 ผังสัญญาณเวลาการส่งข้อมูล**

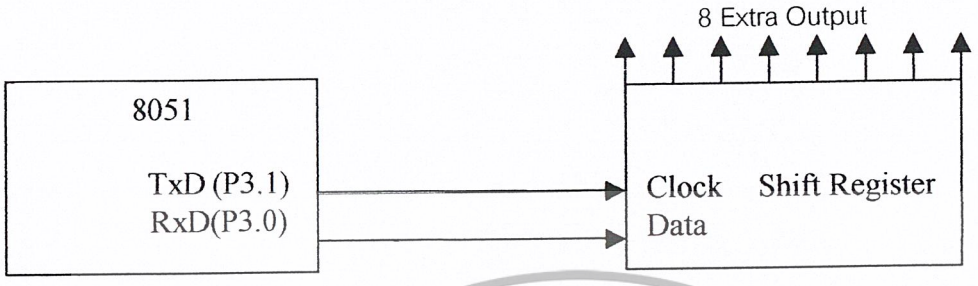
สำหรับการรับข้อมูลจะรับเมื่อ Set Receiver Enable Bit (REN) เป็น "1" และ Clear ที่ขา Receiver Interrup Bit (BI) เป็น "0" ข้อมูลจะเข้าสู่ MCS – 51 เมื่อ Clock Shift ถูกส่งออกไปทาง TxD ที่ขอบขาขึ้น ของ Clock Shift บิตต่ำจะถูกส่งเข้ามาก่อนดังรูปที่ 6.17




**รูปที่ 6.17 ผังสัญญาณเวลารับข้อมูล**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในการประยุกต์ใช้งานในโหมดนี้ ต้องมีไอซีชิพรีจีสเตอร์มาต่อร่วมกันที่ภายนอก เช่น ถ้าหากต้องการส่งข้อมูลออกมาทางพอร์ตอนุกรม อาจจะต่อวงจรได้ดังรูปที่ 6.18 โดยใช้ไอซี Serial - to - Parallel Shift Register โดยข้อมูลส่งออกมาทาง RxD และใช้ TxD เป็น Clock

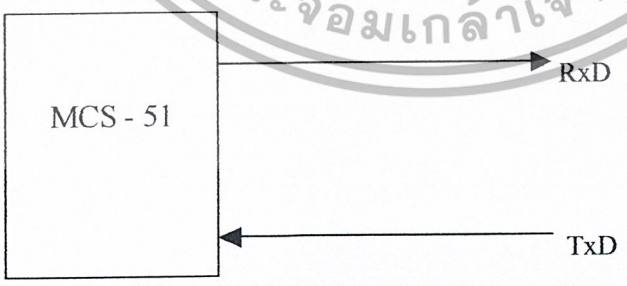



รูปที่ 6.18 การส่งข้อมูลออกโดยใช้ชิพต์รีจิสเตอร์ช่วย

**2. 8 - Bit UART with Variable Baud Rate (Made 1)**

ในโหมดนี้จะเป็นการส่งข้อมูลแบบ 10 บิต ซึ่งประกอบด้วยบิตเริ่มต้น (เป็น "0") ข้อมูล 8 บิต และบิตจบ (เป็น "1") นอกจากนี้ยังสามารถกำหนดค่า Baud Rate ได้โดยค่า Baud Rate นี้จะแปรตามตัวจับเวลาตัวที่ 1 ในโหมดนี้จะส่งข้อมูลออกทาง TxD และรับข้อมูลทาง RxD ถ้าเป็นการรับข้อมูลเข้าตัว Stop Bit จะเข้ามายังบิต RB8 ใน SCON

ถ้าหาก Baud Rate ในการรับส่งข้อมูลกำหนดโดย Timer 1 หลังจากโปรแกรมไปใน Timer 1 แล้วสามารถเลือกค่า Baud Rate ได้อีกสองค่า คือค่าจาก Timer 1 Overflow หาร32 กับค่าจาก Timer 1 Overflow หาร 16




**รูปที่ 6.19 การรับส่งข้อมูลในโหมด 1**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การส่งข้อมูลทำได้โดยการเขียนข้อมูล 8 บิต ไปที่ SBUF โดยบิตที่9 (Stop Bit) ให้เขียนลงใน TB8 ใน SCON จากนั้นข้อมูลจะถูกส่งออกมาทางขา TxD โดยส่ง Start Bit ออกมาก่อนตามด้วยข้อมูล 8 บิตและจบด้วย Stop Bit เมื่อข้อมูลถูกส่งออกไปหมดแล้ว บิต Interrupt Flag (TI) จะเป็น "1" ดังนั้นในการเขียนข้อมูลใหม่ลงไป จะต้องทำการตรวจสอบบิตนี้

ในการรับข้อมูลจะเริ่มจากมีการเปลี่ยนแปลงลอจิกจาก 1 เป็น 0 ทางขา RxD หมายความว่าเริ่มรับบิตเริ่มต้น จากนั้นข้อมูลอีก 8 บิตจะถูกเก็บลงใน SBUF และ Stop Bit จะถูกเก็บในบิต RB8 ของรีจีสเตอร์ SCON เมื่อรับข้อมูลเข้ามาครบแล้วบิต Interrupt Flag (TI) จะถูก Set ดังนั้นในการอ่านข้อมูลจะอ่านเมื่อบิต RI ถูก Set แล้ว เมื่ออ่านข้อมูลไปแล้ว จะต้อง Clear บิตนี้

**3. 9 - Bit UART with Fixed Baud Rate (Mode 2)**

การทำงานในโหมดนี้ไม่สามารถกำหนดค่า Baud Rate ได้ ซึ่งค่า Baud Rate จะมี 2 ค่าคือ 1/64 และ 1/32 ของสัญญาณนาฬิกาบนชิพ การรับส่งข้อมูลจะเป็นชุดข้อมูล 9 บิตประกอบด้วยบิตเริ่มต้น บิตหยุดรวมเป็น 11 บิต โดยข้อมูล 9 บิตจะเป็นจำนวนข้อมูล 8 บิต และบิตที่โปรแกรมได้อีก 1 บิต โดยบิตนี้จะเป็นบิตที่ 9 ซึ่งจะใช้ Parity Bit ในการรับส่งข้อมูลต้องเขียนไปที่บิต TB8 ในรีจีสเตอร์ SCON สำหรับการรับข้อมูลบิตที่ 9 จะถูกเก็บในบิต RB8

**4. 9 - Bit UART with Variable Baud Rate (Mode 3)**

การทำงานในโหมดนี้จะคล้ายกับโหมด 2 แต่สามารถกำหนดค่า Baud Rate ได้ โดยการทำการโปรแกรมไปที่ Timer 1 หลังจากโปรแกรมแล้ว ยังสามารถเลือกค่าได้อีก 2 ค่าคือ ค่าความถี่การ Overflow ของ Timer 1 หารด้วย 16 และ หารด้วย 32

## 6.8 การกำหนดค่าเริ่มต้นให้รีจีสเตอร์ในการรับส่งข้อมูล

การรับข้อมูล ถ้าให้ MCS-51 รับข้อมูลทางพอร์ตอนุกรม จะต้องทำการโปรแกรมไปที่บิต Receiver Enable (REN) ในรีจีสเตอร์ SCON ให้เป็นลอจิก "1" ซึ่งอาจทำได้สองวิธีดังนี้

```
SET     REN
```

เป็นการ Set บิต REN ให้เป็น "1" หรืออาจทำโดยใช้คำสั่ง

```
MOV     SCON,#xxx1xxxxB
```

ซึ่งเป็นการย้ายข้อมูลที่ทำให้บิต REN เป็น 1 สำหรับค่า x หมายความว่าเป็นอะไรก็ได้ขึ้นกับการใช้งานในโหมดต่างๆ

ข้อมูลแบบ 9 บิตในการรับส่งข้อมูลที่มีบิตข้อมูลแบบ 9 บิต ได้แก่ การใช้งานในโหมด 2 และโหมด 3 การส่งข้อมูลบิตที่ 9 จะถูกเขียนในบิต TB8 โดยการเขียนโปรแกรมสำหรับรับข้อมูลเมื่อข้อมูลเข้ามาถึงบิตที่ 9 จะถูกเขียนลงในบิต RB8

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การเพิ่มบิต Parity การส่งข้อมูลแบบ 9 บิตสามารถใช้บิตที่ 9 เป็นบิต Parity ได้ ซึ่งที่บิต Parity จะอยู่ใน Program Status Word (PSW) โดยจะถูก Set หรือ Clear ในทุกๆ Machine Cycle ที่เกี่ยวข้องกับ Accumulator เช่น ถ้าจะทำการส่งข้อมูลแบบ 8 บิต ตามด้วย บิต Even Parity เป็นบิตที่ 9 สามารถเขียนโปรแกรมได้ดังนี้

```
MOV   C,P          ; อ่านค่าจากบิต P มาเก็บในบิต C
MOV   TB8,C        ; นำค่าบิต Parity เขียนลงใน TB8
MOV   SBUF,A       ; ส่งข้อมูลไปทางพอร์ตอนุกรม
```

ถ้าเป็นแบบ Odd Parity ให้ทำการแก้ไขข้อมูลที่อ่านได้จากบิต Parity เสียก่อนที่จะส่งออกไป ซึ่งเขียนโปรแกรมได้ดังนี้

```
MOV   C,P          ; อ่านค่าบิต P มาเก็บใน C
CPL   C            ; กลับค่าให้เป็น Odd Parity
MOV   TB8,C        ; เขียนค่าลงใน TB8
MOV   SBUF,A       ; ส่งข้อมูลไปพอร์ตอนุกรม
```

การส่งข้อมูลแบบมี Parity Bit ด้วยไม่ใช่จะส่งได้แบบบิตหรือโหมด 2 และโหมด 3 เท่านั้น ในโหมด 1 ก็สามารถทำได้อย่างเช่น การส่ง รหัส ASCII จะใช้บิตข้อมูล 7 บิต สำหรับบิตที่เหลืออีกหนึ่งบิตจะเป็นบิต Parity รวมเป็น 8 บิต ซึ่ง สามารถเขียนโปรแกรมได้ดังนี้

```
CLR   ACC.7        ; เคลียร์ค่าบิต 7 เพื่อใช้เป็น Parity Bit
MOV   C,P          ; อ่านค่าบิต P มาเก็บใน C
MOV   ACC.7,C      ; เขียนค่าบิต Parity ลงในรีจิสเตอร์ A
MOV   SBUF,A       ; ส่งข้อมูลไปพอร์ตอนุกรม
```

**แฟลกอินเตอร์รัพท์** เมื่อมีการรับส่งข้อมูลเสร็จ จะมีผลต่อแฟลกอินเตอร์รัพท์ (RIและTI) ในรีจีสเตอร์ SCON ซึ่งบิตเหล่านี้ จะถูก Set โดย Hardware แต่ต้อง Clear ด้วย Software

บิต RI ถ้าถูก Set หมายความว่าบัฟเฟอร์ใช้รับข้อมูลเต็มให้อ่านไปได้แล้วและบิตนี้สามารถใช้อินเตอร์รัพท์ MCS-51 ได้ แต่ถ้าเขียนโปรแกรม จะใช้วิธีตรวจเช็คบิตนี้ ถ้าเป็น "1" หมายความว่าให้อ่านข้อมูลมาเก็บในรีจีสเตอร์ A ได้แต่ก่อนอ่านจะต้อง Clear RI เสียก่อน เพื่อจะได้รับข้อมูลถัดไป ซึ่งเขียนโปรแกรมได้ดังนี้

```
WAIT:   JNB   RI,WAIT      ; ถ้าบิตนี้ไม่เป็น "1" จะทำงานอยู่ที่เดิม
        CLR   RI           ; เคลียร์ RI
        MOV   A,SBUF       ; อ่านค่ามาเก็บใน A
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บิต TI เมื่อส่งข้อมูลออกไปแล้ว บิตนี้จะถูก Set เป็นการบอกว่าบัฟเฟอร์ส่งข้อมูลว่างแล้ว ให้ส่งข้อมูลเข้าไปได้ ซึ่งสามารถใช้บิตนี้อินเตอร์รัพท์ MCS - 51 ได้เช่นกัน แต่ถ้าเขียนโปรแกรมคอยตรวจเช็คอาจเขียนได้ดังนี้

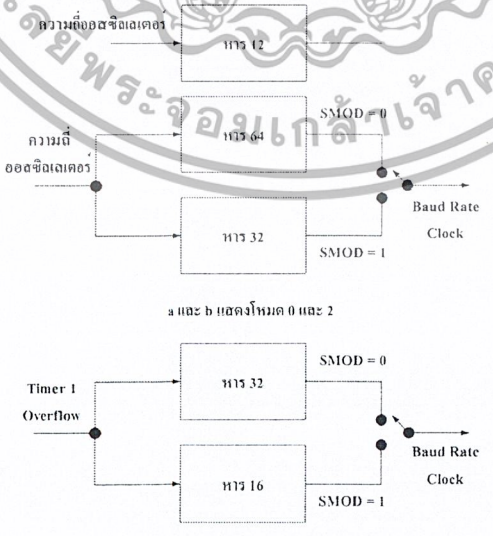
```
WAIT:   JNB   TI,WAIT      ; ตรวจบิต TI ว่าเป็น "1" หรือยัง
        CLR   TI           ; เคลียร์ TI
        MOV   SBUF,A       ; เขียนข้อมูลลงไป
```

**6.9 อัตราการส่งข้อมูลทางพอร์ตอนุกรม**

จากการศึกษาการรับส่งข้อมูลในโหมดต่างๆพบว่าในโหมด 0 และ โหมด 2 ไม่สามารถกำหนดอัตราบอดเองได้ โดยในโหมด 0 ค่าอัตราบอดจะมีค่าเท่ากับความถี่ของ Oscillator หารด้วย 12 ในโหมด 1 จะมีสองค่าคือ ความถี่ Oscillator หารด้วย 32 และหารด้วย 64 สองค่านี้เรียกว่า SMOD0 และ SMOD1 ซึ่งสามารถกำหนดได้ในรีจีสเตอร์ PCON บิตที่ 7 ในรีจีสเตอร์ PCON นี้ไม่สามารถเข้าถึงข้อมูลในระดับบิตได้ การเขียนข้อมูลลงไปทีละบิต จะต้องใช้วิธีที่เรียกว่า "Read - Modify - Write" คืออ่านค่าขึ้นมาแก้ไขแล้วเขียนลงไปใหม่ตัวอย่างเช่น

```
        MOV   A,PCON       ; อ่านค่าจาก PCON มาเก็บในรีจีสเตอร์ A
        SETB  ACC.7        ; เซตบิต 7 (SMOD)
        MOV   PCON,A       ; เขียนค่าลงไปใหม่ใน PCON
```

สำหรับโหมด 1 และ โหมด 3 สามารถกำหนดค่าอัตราบอดได้ โดยทำการโปรแกรมลงใน Timer 1 ในการโปรแกรมแต่ละครั้งจะมี SMOD สองค่าเช่นกัน ค่าของอัตราบอดของโหมดต่างๆ แสดงได้ดังรูปที่ 6.20




**รูปที่ 6.20 การกำหนดอัตราบอด ในโหมดต่างๆ**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**การใช้ Timer 1 กำหนด Baud Rate Clock**

การกำหนดค่าลงใน Timer 1 ทำได้โดยการโปรแกรมไปที่ TMOD ให้ทำงานแบบ 8 Bit Auto Reload Mode (Mode 2) โดยเขียนค่าไปที่ TH1 ซึ่งโปรแกรมที่รีจีสเตอร์ TMOD ได้ดังนี้

MOD TMOD,#0010xxxxB

ค่า x หมายความว่าเป็นอะไรก็ได้ เพราะบิตเหล่านี้ใช้ใน Timer 0

ถ้าต้องการอัตราบอดต่ำๆ สามารถใช้ 16 - Bit Mode ได้โดยทำการโปรแกรมเป็น TMOD = 0001xxxxB ค่าอัตราบอดที่ส่งออกมาจะมีค่าเท่ากับความถี่ของ Timer 1 ที่เกิด Overflow หารด้วย 32 (หรือหารด้วย16 ถ้าเป็น SMOD = 1)

รูปแบบทั่วไปของการหาค่าอัตราบอดในโหมด 1 และ โหมด 3 สามารถหาได้ดังนี้

BAUD RATE = TIMER 1 OVERFLOW RATE/32

ถ้าต้องการอัตราบอดเท่ากับ 1200 สามารถที่จะคำนวณค่าความถี่ Overflow ของ Timer 1 ได้ดังนี้

1200 = Timer 1 Overflow Rate/32

จะได้ Timer 1 Overflow Rate เท่ากับ 38.4 KHz

ถ้าระบบ MCS-51 ใช้ความถี่สัญญาณนาฬิกาจาก Crystal เท่ากับ 12 MHz Timer 1 จะได้รับ Clock เท่ากับ 1 MHz หรือ 1000 KHz ถ้าเราต้องการ Timer 1 Overflow เท่ากับ 38.4 KHz ดังนั้นค่าอัตรา Overflow มีค่าเท่ากับ 1000/38.4 = 26.04 Clock โดยค่า Overflow จะเกิดขึ้นเมื่อเกิดการเปลี่ยนจาก FFH เป็น 00H ดังนั้นจะต้องทำให้ Timer 1 นับไป 26 Count ดังนั้นค่าที่จะให้รีจีสเตอร์ TH1 มีค่าเท่ากับ -26 ซึ่งใช้เป็นค่า Reload ดังนั้นเขียนคำสั่งได้ดังนี้

MOV TH1,# -26

ตัวโปรแกรมแอสแซมเบอร์ทั่วไปจะแปลงค่า –26 เป็น OE6H เอง โดยจากที่ผ่านมาจะเห็นว่าความถี่ อัตราบอด จะมีความสัมพันธ์กับค่าสัญญาณนาฬิกาที่ใช้จาก Crystal ในตารางที่ 6.21 จะเป็นค่าที่ต้องกำหนดในTimer 1 เมื่อต้องการค่าอัตราบอดต่างๆ

**ตารางที่ 6.21 ความถี่สัญญาณนาฬิกาที่ใช้กำหนดอัตราบอดค่าต่างๆ**

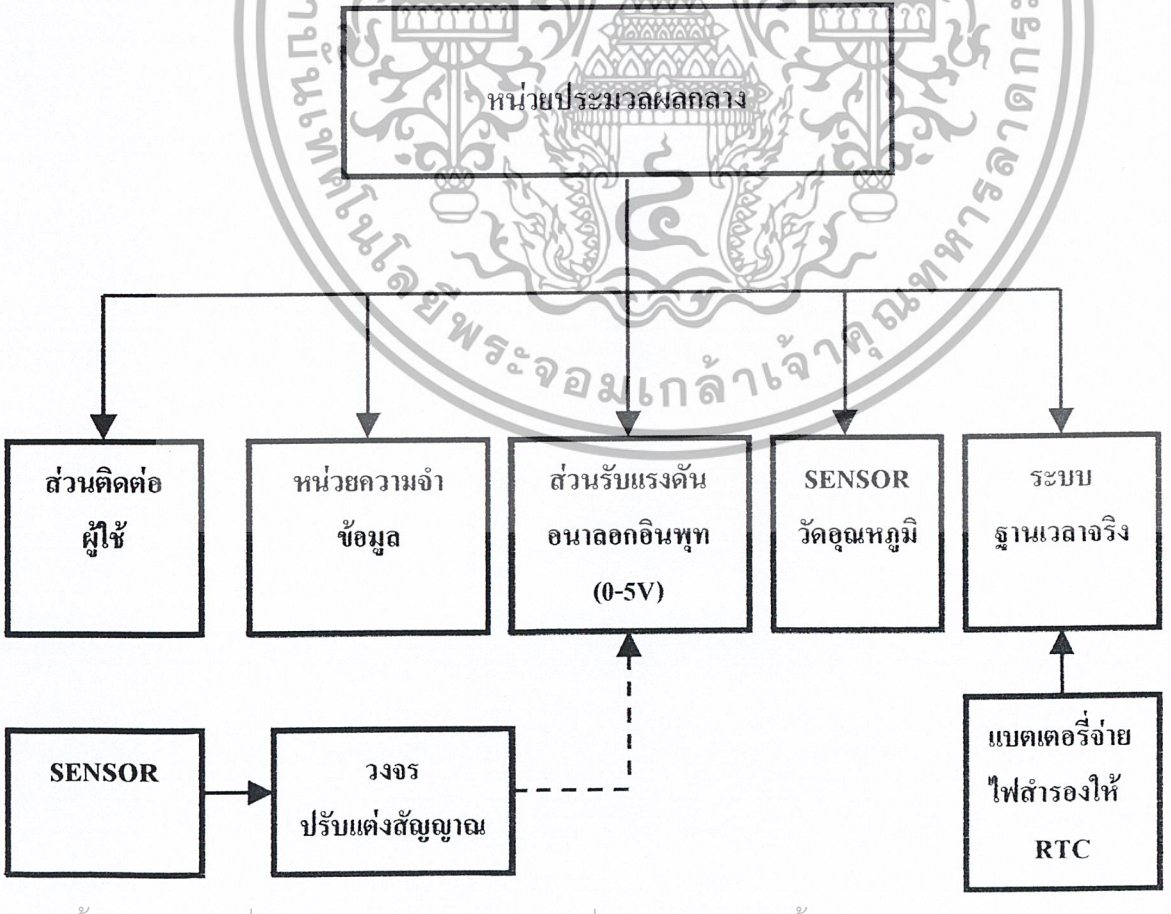
| ค่าอัตราบอด | Crystal | SMOD โหมด | ค่าใน TH1 | ค่าอัตราบอดที่ได้ | ผิดพลาด |
|---|---|---|---|---|---|
| 9600 | 12.000 | 1 | -7(F9H) | 8923 | 7% |
| 2400 | 12.000 | 0 | -13(F3H) | 2402 | 0.16% |
| 1200 | 12.000 | 0 | -16(E6H) | 1202 | 0.16% |
| 19200 | 11.059 | 1 | -3(FDH) | 19200 | 0 |
| 9600 | 11.059 | 0 | -3(FDH) | 9600 | 0 |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# บทที่ 7
## การออกแบบฮาร์ดแวร์

ในการออกแบบวงจรของโครงงานโดยรวมนั้น จะเป็นไปตามโครงสร้างของการทำงานแสดงดังรูปที่ 7.1 ซึ่งมีหลักการทำงานดังนี้

- ตัวควบคุมและประมวลผลส่วนกลางจะทำหน้าที่ในการควบคุมการทำงานของอุปกรณ์ในส่วนต่างๆอันได้แก่ หน่วยความจำข้อมูล , ส่วนรับแรงดันอนาลอกอินพุท ( 0-5 โวลต์ ) , เซนเซอร์วัดอุณหภูมิและระบบฐานเวลาจริง
- เมื่อเริ่มทำงานตัวควบคุมและประมวลผลกลาง จะอ่านค่าอุณหภูมิจากเซนเซอร์วัดอุณหภูมิและค่าสัญญาณอนาลอกอินพุทจากส่วนรับแรงดันอนาลอกอินพุท (0-5 โวลต์) จากนั้นจะแปลงข้อมูลให้อยู่ในรูปแบบที่เหมาะสมแล้วเก็บข้อมูลนั้นไว้ในหน่วยความจำข้อมูลอัตรา การเก็บข้อมูลจะอิงเวลาจากระบบฐานเวลาจริง เพื่อให้ได้เวลาที่แน่นอนและตรงกับความเป็นจริง
- การติดต่อข้อมูลกับอุปกรณ์ดังกล่าวมานั้นจะใช้การติดต่อแบบบัสข้อมูลอนุกรม( $I^2C$) เพื่อเป็นการลดจำนวนสายสัญญาณในระบบ และประหยัดพลังงานด้วย




**รูปที่ 7.1 โครงสร้างของบอร์ด DUAL-MODE DATA LOGGER**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**7.1 หน่วยการทำงานต่างๆของอุปกรณ์**

สามารถแสดงการทำงานในส่วนต่างๆดังรายละเอียดดังนี้

**1. หน่วยประมวลผลกลาง**

ใช้ไมโครคอนโทรลเลอร์เบอร์ AT89S8252 เป็นหน่วยประมวลผลกลาง ซึ่งทำงานด้วยความถี่ 18.345 เมกกะเฮิรตซ์ คอยควบคุมอุปกรณ์ในหน่วยต่างๆผ่านทางบัสข้อมูลอนุกรม

**2. ส่วนตรวจรับแรงดันอนาลอกอินพุท ( 0-5V )**

ตัวแปลงสัญญาณอนาลอกไปเป็นสัญญาณดิจิตอล(Analog to Digital Converter : ADC) ใช้เบอร์ LTC1298 มีค่าความละเอียดในการวัด 12 บิต สื่อสารข้อมูลโดยใช้สายสัญญาณ 3 เส้น โดยใช้สาย DI ร่วมกับ DO สามารถวัดสัญญาณแบบขั้วเดียว ( Single End ) ได้ 2 ช่องสัญญาณ การสื่อสารข้อมูลจะใช้แบบ $I^2C$ บัส

**3. SENSOR วัดอุณหภูมิ**

ใช้เซนเซอร์วัดอุณหภูมิเบอร์ DS1620 ซึ่งจะสามารถทำการวัดอุณหภูมิได้ตั้งแต่ –55 องศาเซลเซียสจนถึง +125 องศาเซลเซียส มีความละเอียด 0.5 องศาเซลเซียส โดยจะแปลค่าอุณหภูมิที่วัดได้ เป็นข้อมูลดิจิตอลขนาด 9 บิต ทำการสื่อสารข้อมูลผ่านการติดต่อแบบอนุกรม โดยใช้สายสัญญาณ 3 เส้น ซึ่งจากวงจรสมบูรณ์ใช้งานจริงของโครงงาน จะพบว่าจะใช้การติดต่อในรูปแบบ $I^2C$ บัส

**4. ระบบฐานเวลาจริง**

ใช้ตัวสร้างสัญญาณนาฬิกาแบบเรียลไทม์ (Real Time Clock : RTC) เบอร์ DS1307 ที่มีหน้าที่สร้างฐานเวลา ( โดยใช้คริสตอล 32.768 กิโลเฮิรตซ์ในการสร้าง )

ภายในยังมีหน่วยความจำ RAM ไว้ใช้งานได้อีก 56 ไบต์ ซึ่งในการทำงานเป็นที่เก็บค่าคอนฟิกต่างๆของบอร์ด การสื่อสารจะใช้เป็นแบบ $I^2C$ บัส

**5. หน่วยความจำข้อมูล**

ได้ออกแบบให้ใช้หน่วยความจำเก็บข้อมูลถาวร (Eeprom Memory Unit) เบอร์ 24C256 ขนาด 32 กิโลไบต์ มีการกำหนดให้มีการติดต่อสื่อสารกันระหว่าง Eeprom กับ CPU ผ่านบัส $I^2C$

จากวงจรสมบูรณ์ของโครงงานเราจะกำหนดตำแหน่งของ EEPROM นี้ให้อยู่ที่ตำแหน่ง 0A0h โดยกำหนดให้ที่ขา A2,A1,A0 ลงกราวนด์ทั้งหมด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**6. ส่วนติดต่อกับผู้ใช้**

ส่วนที่1 กระทำผ่านพอร์ตอนุกรม RS-232 ที่ใช้สำหรับติดต่อกับคอมพิวเตอร์ โดยที่ผู้ใช้สามารถสั่งงานผ่านคีย์บอร์ดคอมพิวเตอร์และเรียกดูข้อมูลต่างๆได้ทางหน้าจอ โดยใช้โปรแกรมเทอร์มินอลทั่วไปหรือผ่านทางหน้าจอ User interface

ส่วนที่ 2 ใช้เมื่ออยู่ในบริเวณที่ไม่สามารถทำการสั่งงานติดต่อกับคอมพิวเตอร์ได้ ซึ่งในที่นี้ SW – START จะทำหน้าที่ในการทำงานในสภาวะ Stand-alone โดย LED Display จะกระพริบทุกครั้งที่มีการเก็บข้อมูล

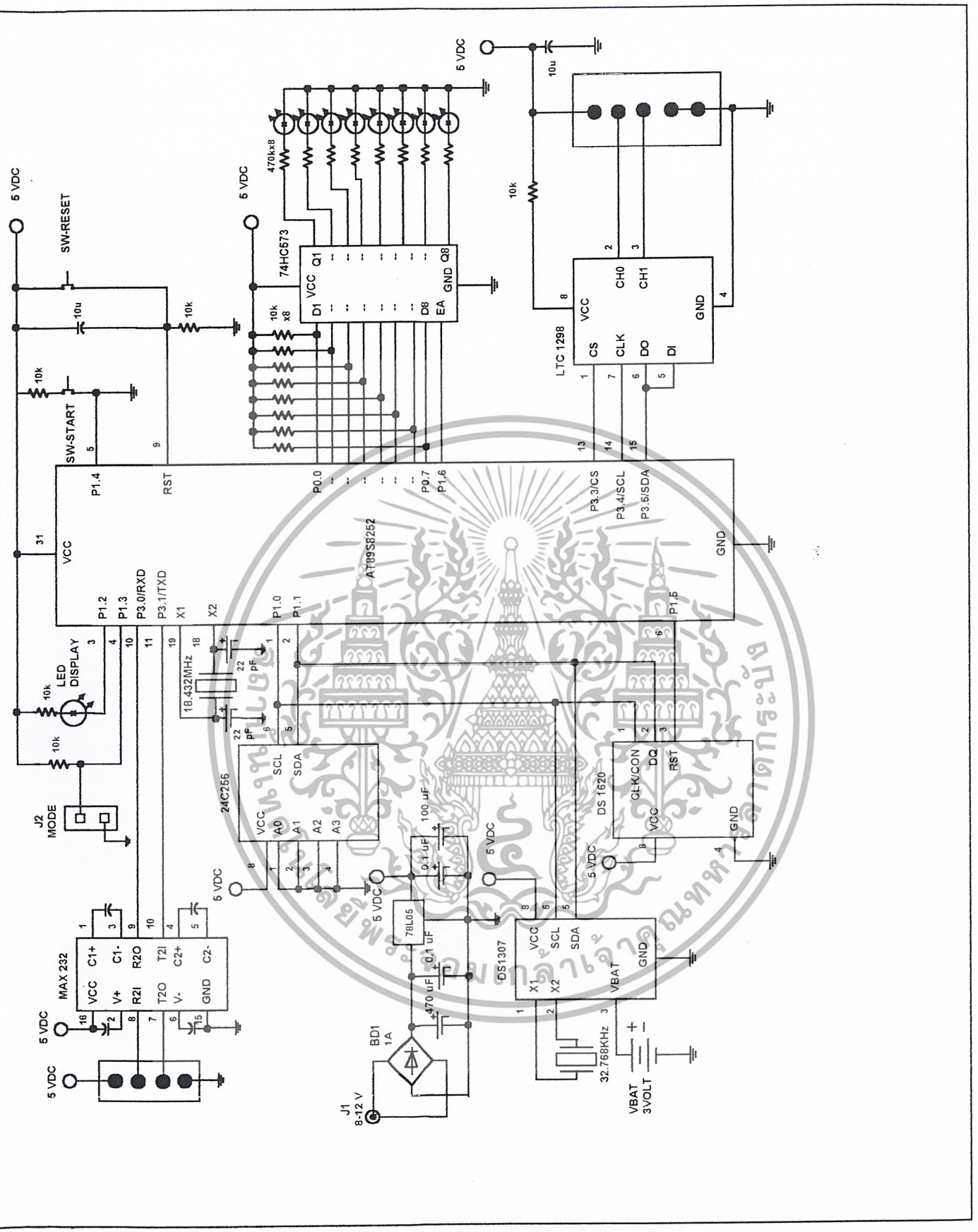
นอกจากนี้ยังพบว่าในส่วนของวงจรขับ LED 8 ตัวนั้น ในส่วนนี้จะมีการแสดงผลไฟวิ่งไปกลับทุกครั้งๆละ 2 รอบในทุกครั้งที่จะเริ่มมีการบันทึกค่าข้อมูล

**7. แหล่งจ่ายไฟ**

มีอยู่ 2 ส่วนคือ

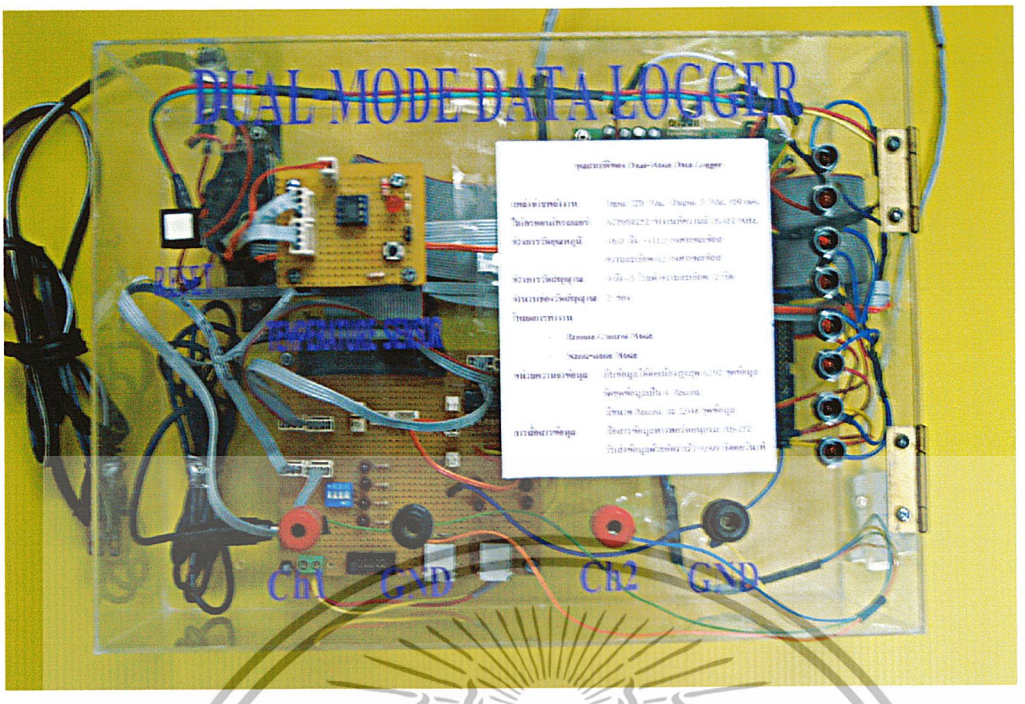
- ไฟเลี้ยงสำหรับวงจรใช้แหล่งจ่ายไฟฟ้ากระแสตรง DC 8-12 โวลต์ ต่อเข้าที่แจ็ค J1 ภายในบอร์ดจะมีไดโอดเรกติไฟร์ช่วยป้องกันการต่อไฟเลี้ยงผิดขั้ว นอกจากนี้ยังมีไอซีเรกูเลเตอร์เบอร์ 78L05 รักษาระดับแรงดันไฟเลี้ยง 5 โวลต์ไว้จ่ายให้กับวงจรทั้งหมด
- ไฟสำรองจ่ายให้กับวงจร RTC เนื่องจากวงจร RTC จำเป็นต้องทำงานอยู่ตลอดเวลา เพื่อจะรักษาค่าเวลาให้ถูกต้องจึงจำเป็นต้องมีไฟเลี้ยงสำรองจ่ายให้กับวงจร RTC ในขณะที่ไม่มีไฟเลี้ยงตามปกติ ( ภาวะไม่ได้ใช้งานหรือไฟฟ้าขัดข้อง ) โดยใช้แบตเตอรี่ลิเธียม-ไอออนขนาดแรงดัน 3 โวลต์ต่อไว้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

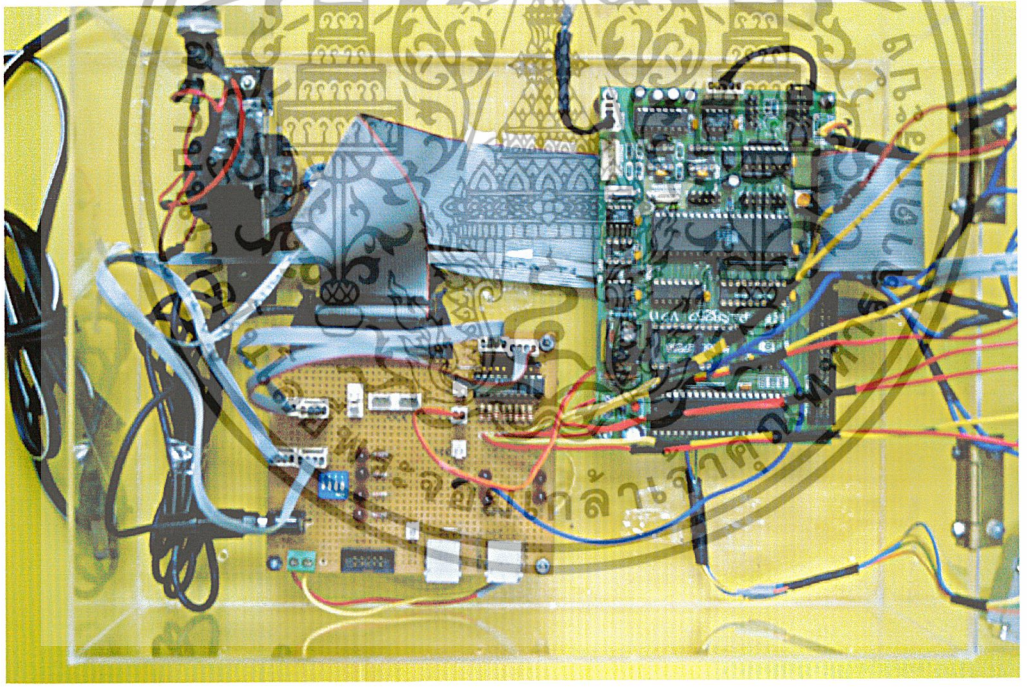



**รูปที่ 7.2 วงจรสมบูรณ์ของโครงงาน**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้




(ก)



(ข)

**รูปที่ 7.3 โครงงานสมบูรณ์ที่ทำการออกแบบและติดตั้ง**

**(ก) ภาพด้านบนแสดงหน้าปัดโครงงาน**

**(ข) ภาพการเดินสายภายในโครงงาน**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**7.2 คุณสมบัติของโครงงาน Dual – Mode Data Logger**

**แหล่งจ่ายพลังงาน** Input 220 VAC. Output 5 VDC 850 mA

**ไมโครคอนโทรลเลอร์** AT89S8252 ทำงานที่ความถี่ 18.432 MHz

**ช่วงการวัดอุณหภูมิ** -16.0 ถึง +111.5 องศาเซลเซียส ความละเอียด 0.5 องศาเซลเซียส

**ช่วงการวัดสัญญาณ** 0 ถึง –5 โวลต์ ความละเอียด 12 บิต

**จำนวนช่องวัดสัญญาณ** 2 ช่อง

**โหมดการทำงาน**

- **Remote Control Mode**
- **Stand-alone Mode**

**หน่วยความจำข้อมูล** เก็บข้อมูลได้ต่อเนื่องสูงสุด 8,192 ชุดข้อมูล ( ชุดข้อมูลละ 4 ไบต์ ) เก็บได้เป็น 4 Record ขนาดRecordละ 2,048 ชุดข้อมูล

**การสื่อสารข้อมูล** สื่อสารข้อมูลทางพอร์ต RS-232 ด้วยอัตราเร็ว 9,600 บิตต่อวินาที ใช้เวลาดาวน์โหลดข้อมูลประมาณ 16 วินาทีต่อ 1 Record

**กระบวนการทำงาน**

- **SETTING** สามารถตั้งค่าเริ่มต้นในการทำงานได้ดังนี้
  - RECORD เป็นการเลือกตำแหน่งชุดข้อมูลเริ่มต้น ที่ต้องทำการวัดหรือบันทึกข้อมูล (1-4)
  - NUM OF RECORD การระบุจำนวนชุดข้อมูล ที่ต้องการทำการวัดหรือบันทึกข้อมูล (1-4)
  - PERIOD เลือกอัตราการวัดหรือบันทึกข้อมูล (1-255 วินาทีต่อครั้ง) มีอัตราการวัดสูงสุดประมาณ 28 ครั้งต่อวินาที ( เลือกที่ 000)
  - CLOCK ปรับเวลาให้ตรงกับเวลาปัจจุบันที่วัดหรือบันทึกข้อมูลอยู่ (hour/minute/second)
  - DATE ปรับวันที่ให้ตรงกับเวลาปัจจุบันที่วัดหรือบันทึกข้อมูลอยู่ (Date/Month/Years)
- **VIEW_SETTING** เรียกดูค่าต่างๆที่การตั้งไว้
- **MEASURE_TEST** วัดและแสดงผลข้อมูลผ่านหน้าจอคอมพิวเตอร์ โดยยังไม่มีการบันทึกลงหน่วยความจำข้อมูล
- **LOGGER_DATA** วัดและบันทึกข้อมูลลงในหน่วยความจำข้อมูล ตามตำแหน่งชุดข้อมูลเริ่มต้น,จำนวนของชุดข้อมูลที่ต้องการ,และคาบเวลาที่กำหนด ในรูปแบบการทำงานในสภาวะ Stand-alone จะเป็นการเข้าสู่กระบวนการนี้ในทันทีเมื่อมีการกด SW-START
- **DOWNLOAD** เรียกดูข้อมูลที่บันทึกไว้มาทำการวิเคราะห์ต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# บทที่ 8
# การออกแบบซอฟท์แวร์

หลังจากที่ได้สร้างฮาร์ดแวร์ของโครงงานเสร็จเรียบร้อยแล้ว ยังไม่สามารถนำโครงงานไปใช้งานได้ทันที จำเป็นที่ต้องมีการเขียนโปรแกรมควบคุมให้กับตัวไมโครคอนโทรลเลอร์ในหน่วยประมวลผลกลาง เพื่อกำหนดหน้าที่และลำดับการทำงานของโครงงานโดยพิจารณาตามลำดับดังนี้

1. ต้องทราบถึงจุดมุ่งหมายและความต้องการใช้งานในกระบวนการต่างๆคือ
   1.1 ต้องสามารถใช้งานในการวัดค่าอุณหภูมิและสัญญาณได้ในเวลาเดียวกัน
   1.2 ต้องสามารถบันทึกค่าที่ทำการวัดได้
   1.3 สามารถใช้งานได้ 2 โหมดการทำงานคือ
      - **REMOTE CONTROL MODE**
      - **STAND-ALONE MODE**
   1.4 ต้องมีกระบวนการต่างๆดังนี้

- **SETTING** สามารถตั้งค่าเริ่มต้นในการทำงานได้ดังนี้
  - RECORD เป็นการเลือกตำแหน่งชุดข้อมูลเริ่มต้นที่ต้องการวัดหรือบันทึกข้อมูล (1-4)
  - NUM OF RECORD การระบุจำนวนชุดข้อมูลที่ต้องการวัดหรือบันทึกข้อมูล (1-4)
  - SAMPLING-RATE PERIOD เลือกอัตราการวัดหรือบันทึกข้อมูล (1-255 วินาทีต่อครั้ง) โดยมีอัตราการวัดสูงสุดประมาณ 28 ครั้งต่อวินาที (เลือกที่ 000)
  - CLOCK ปรับเวลาให้ตรงกับเวลาปัจจุบันที่วัดหรือบันทึกข้อมูล (hour/minute/second)
  - DATE ปรับวันที่ให้ตรงกับเวลาปัจจุบันที่วัดหรือบันทึกข้อมูลอยู่(Date/Month/Years)
- **VIEW_SETTING** เรียกดูค่าต่างๆที่การตั้งไว้
- **MEASURE_TEST** วัดและแสดงผลข้อมูลผ่านหน้าจอคอมพิวเตอร์ โดยยังไม่มีการบันทึกลงในหน่วยความจำข้อมูล
- **LOGGER_DATA** วัดและบันทึกข้อมูลลงในหน่วยความจำข้อมูล ตามตำแหน่งชุดข้อมูลเริ่มต้น, จำนวนของชุดข้อมูลที่ต้องการ และคาบเวลาที่ได้กำหนด
  ในรูปแบบการทำงานในสภาวะ Stand-alone จะเข้าสู่กระบวนการนี้ในทันทีเมื่อมีการกด SW-START
- **DOWNLOAD** เรียกดูข้อมูลที่บันทึกไว้มาทำการวิเคราะห์ต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. เมื่อพิจารณาความต้องการดังกล่าวข้างต้นแล้ว ต่อมาต้องมาทำความความเข้าใจในการจัดสรรหน่วยความจำของข้อมูลที่จะทำการบันทึกค่าดังรูปที่ 8.1 และรูปที่ 8.2

| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | |
|---|---|---|---|---|---|---|---|---|---|
| 38H | | | | | | | NUMREC | RECORD | 3FH |
| 30H | | | | | | | | | 37H |
| 28H | | | | | | | | | 2FH |
| 20H | DATE4 | MONTH 4 | YEAR4 | HOUR4 | MIN4 | SEC4 | PER4 | | 27H |
| 18H | DATE3 | MONTH 3 | YEAR3 | HOUR3 | MIN3 | SEC3 | PER3 | | 1FH |
| 10H | DATE2 | MONTH 2 | YEAR2 | HOUR2 | MIN2 | SEC2 | PER2 | | 17H |
| 08H | DATE1 | MONTH 1 | YEAR1 | HOUR1 | MIN1 | SEC1 | PER1 | | 0FH |

**RECORD** = ชุดข้อมูลเริ่มต้นที่ต้องการบันทึก

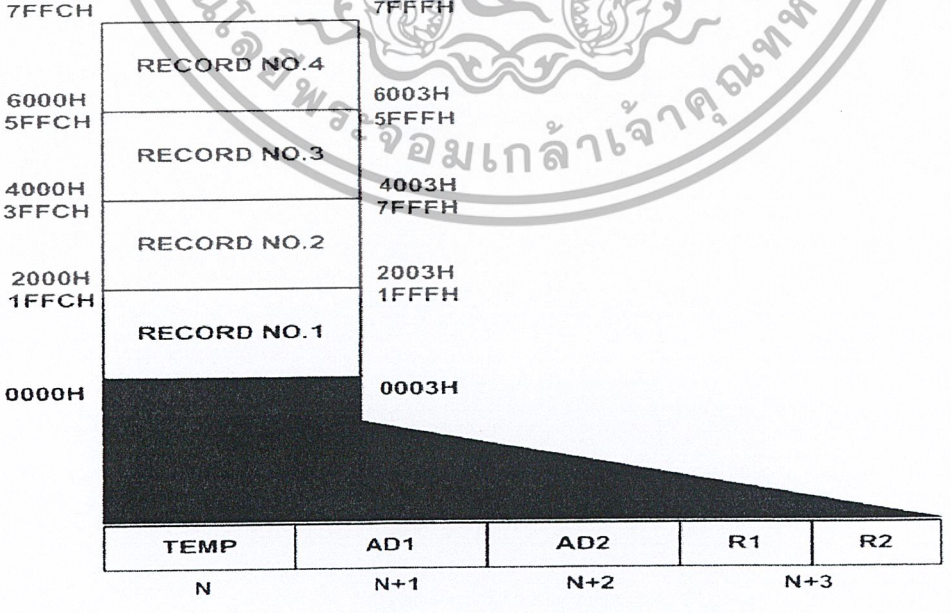
**NUMREC** = จำนวนชุดข้อมูลที่ต้องการให้มีการบันทึกอย่างต่อเนื่อง

**DATEx/MONTHx/YEARx** = วันที่เริ่มทำการบันทึกข้อมูลของชุดข้อมูลที่ x

**HOURx/MINx/SECx** = เวลาที่เริ่มทำการบันทึกข้อมูลของชุดข้อมูลที่ x

**PERx** = คาบเวลาในการบันทึกข้อมูลในชุดข้อมูล x

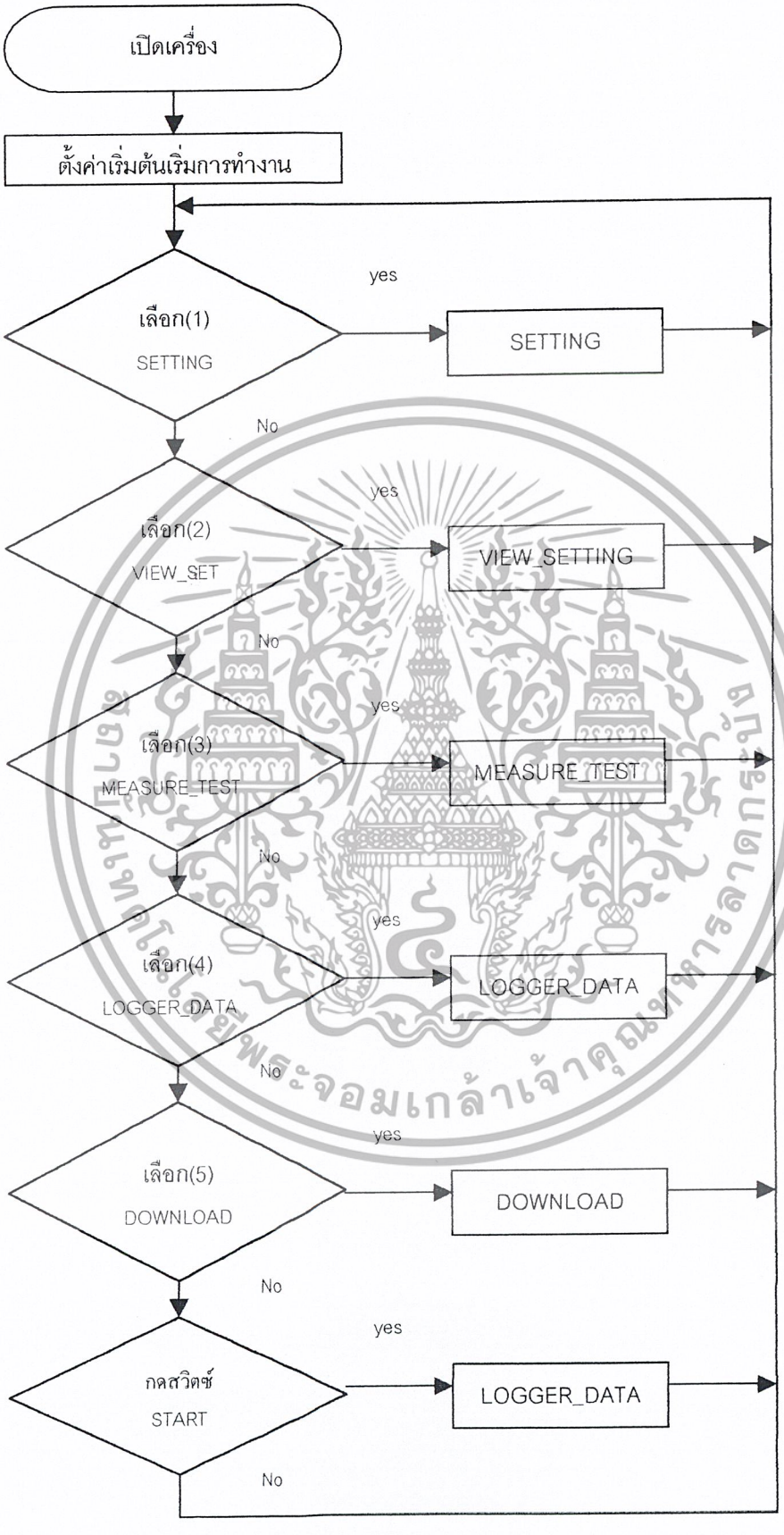
**รูปที่ 8.1 รูปแบบการเก็บค่าเริ่มต้นในหน่วยความจำ RAM RTC**




**รูปที่ 8.2 รูปแบบการเก็บค่าเริ่มต้นในหน่วยความจำ EEPROM**

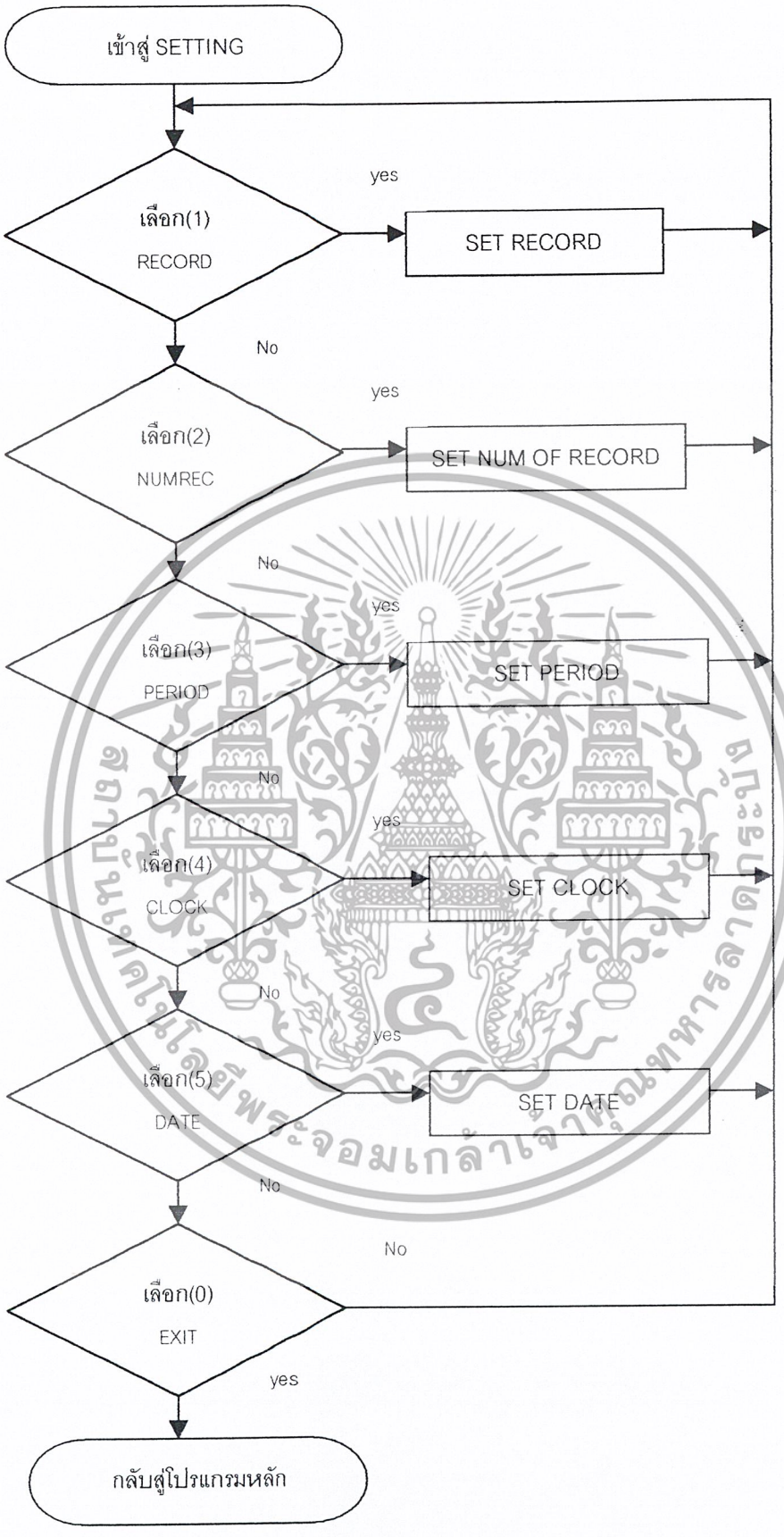
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. จากโครงสร้างโปรแกรมสามารถนำมาเขียนผังโดยรวมของโครงงานได้ดังรูปที่ 8.3



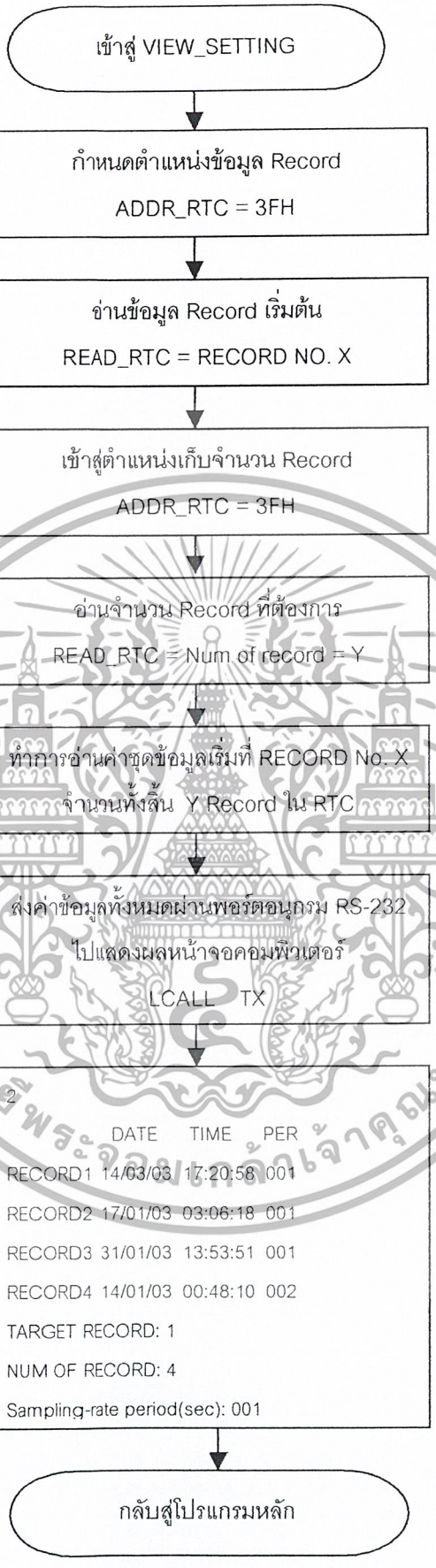

**รูปที่ 8.3 ผังโปรแกรมควบคุมการทำงานหลัก**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



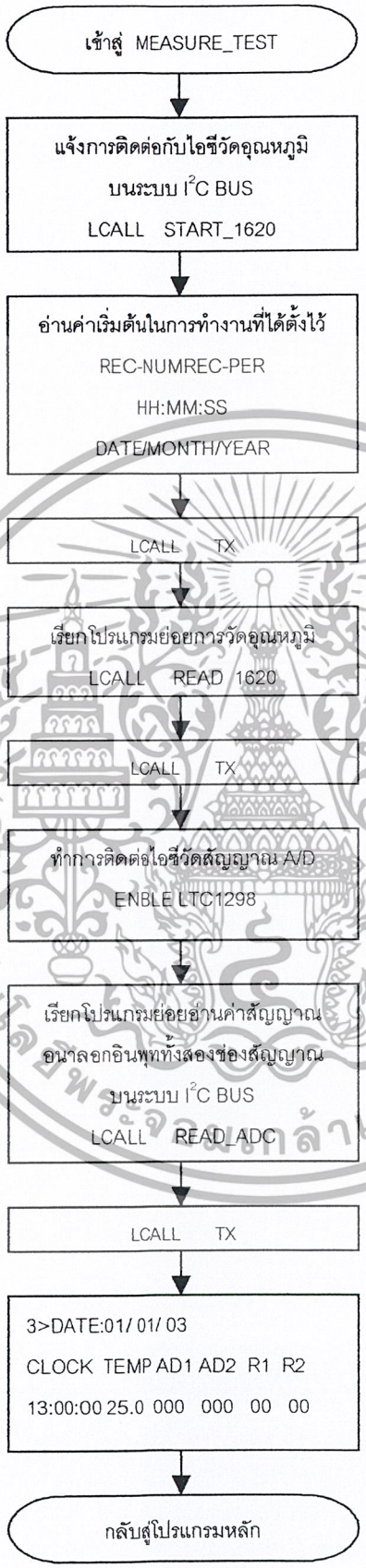

**รูปที่ 8.4 ผังโปรแกรมย่อย SETTING**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



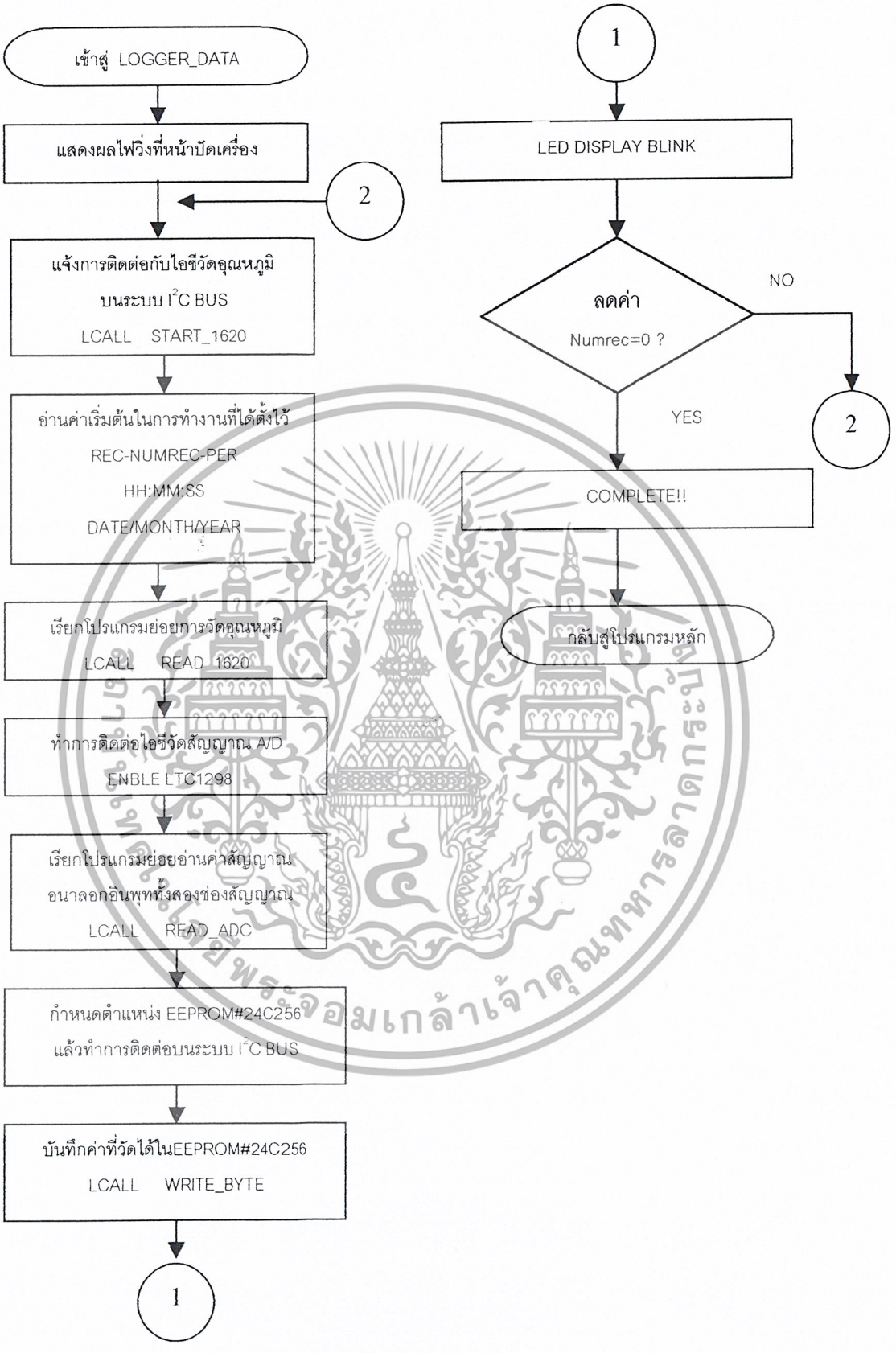

**รูปที่ 8.5 ผังโปรแกรมย่อย VIEW_SETTING**

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



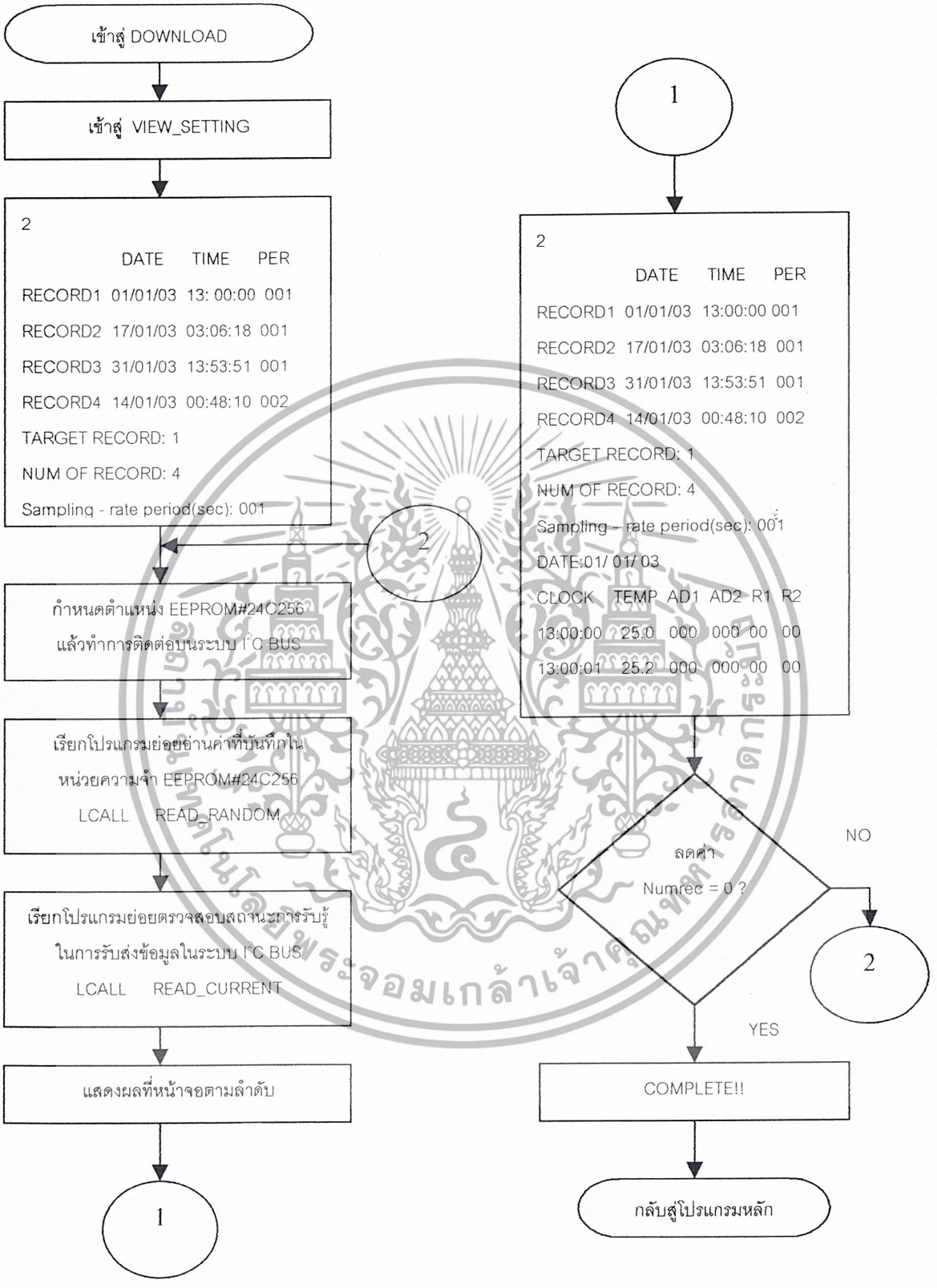

**รูปที่ 8.6 ผังโปรแกรมย่อย MEASURE_TEST**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



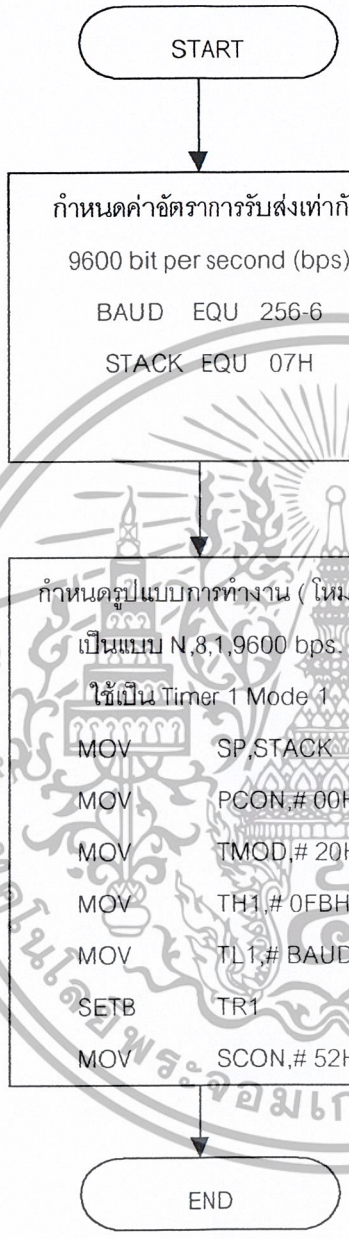

**รูปที่ 8.7 ผังโปรแกรมย่อย LOGGER_DATA**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้




**รูปที่ 8.8 ผังโปรแกรมย่อย DOWNLOAD**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



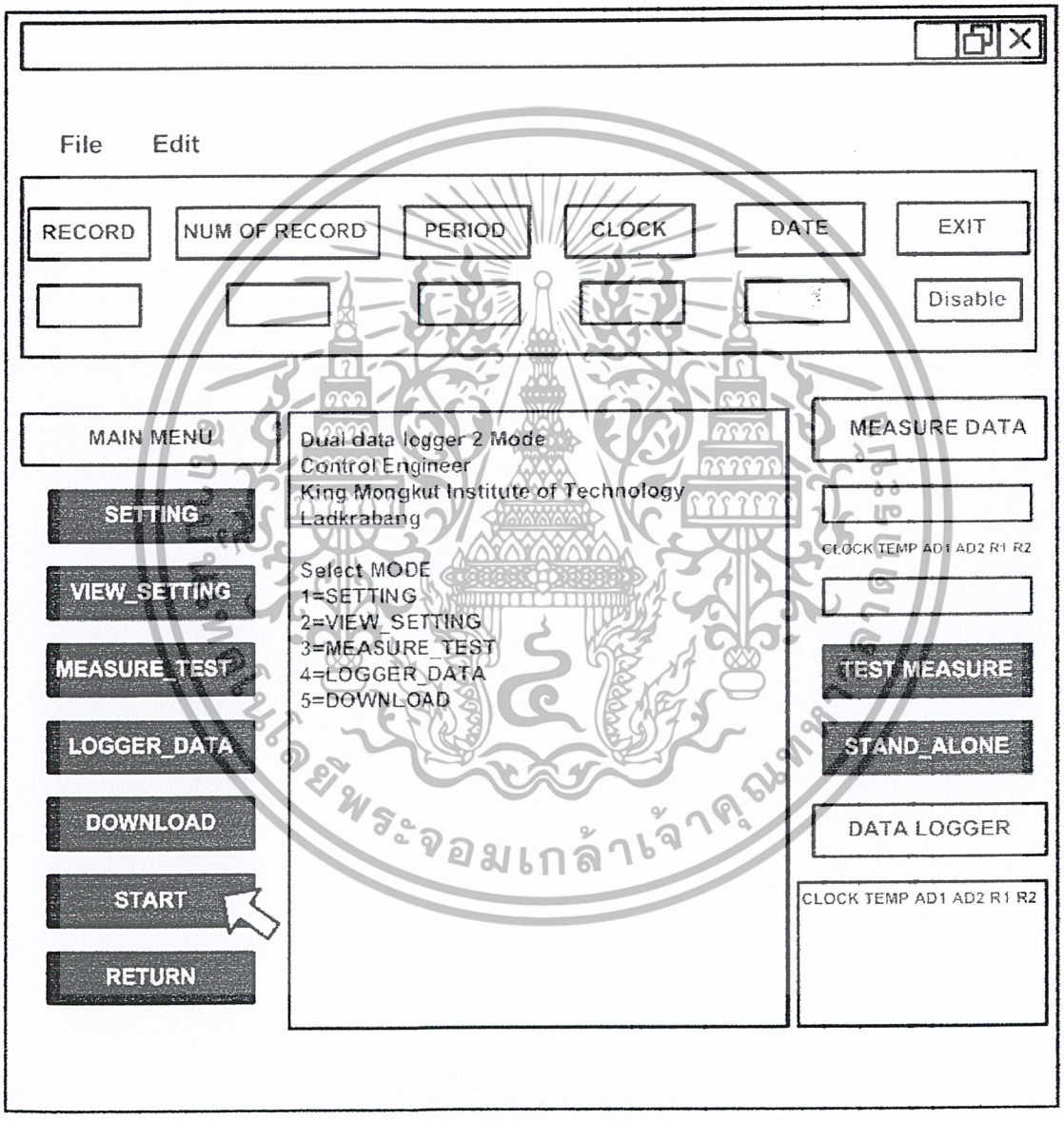

**รูปที่ 8.9 ผังการกำหนดรูปแบบการติดต่อผ่าน RS-232 ในโครงงาน**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# บทที่ 9
# บทวิจารณ์และสรุปผล

## 9.1 ผลการทดลอง

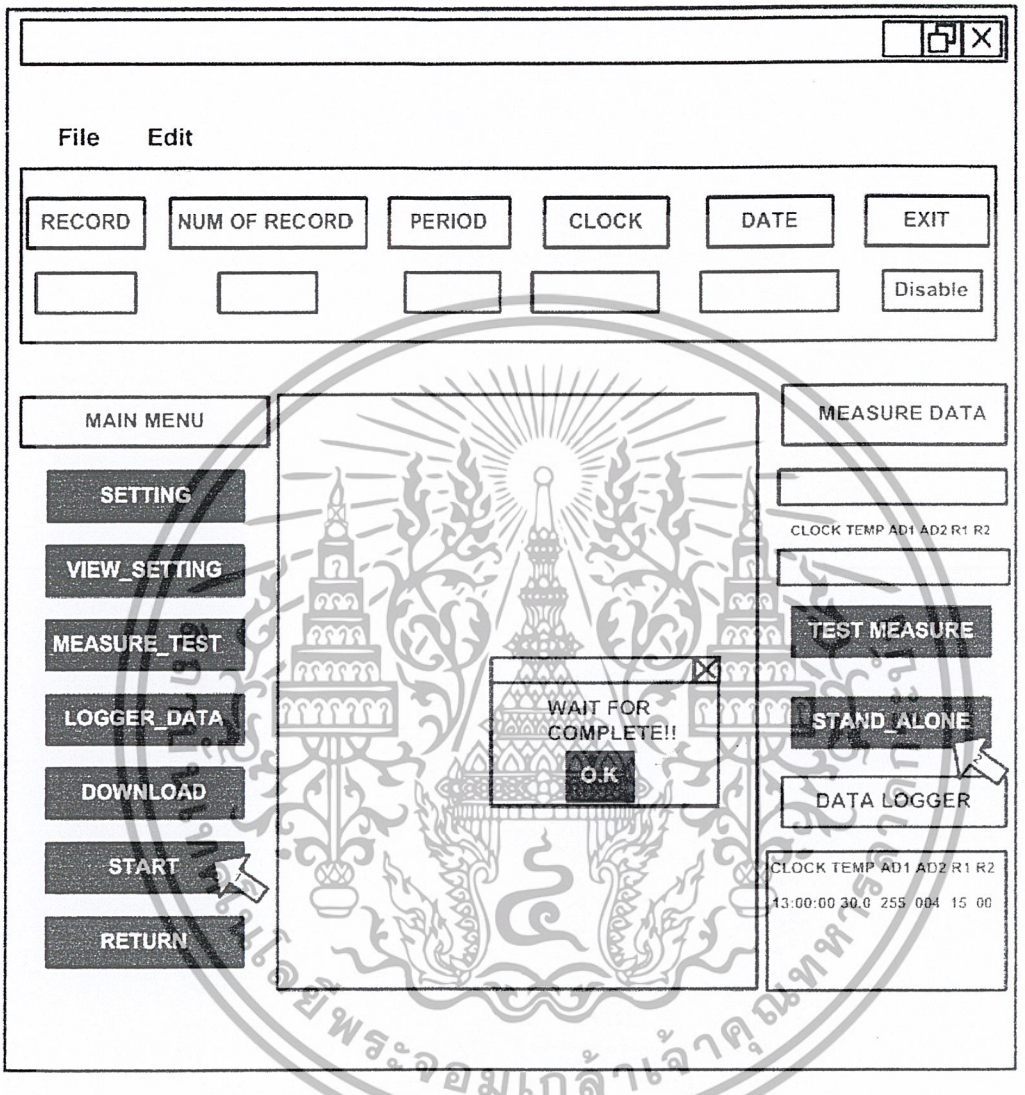
1. เมื่อทำการประกอบเครื่องเข้ากับคอมพิวเตอร์และทำการเปิดโปรแกรมใช้งาน พร้อมทั้งกดปุ่ม **START** ที่หน้าจอจะปรากฏหน้าจอ User Interface หลักดังรูปที่ 9.1




**รูปที่ 9.1 หน้าจอหลักของ User Interface**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

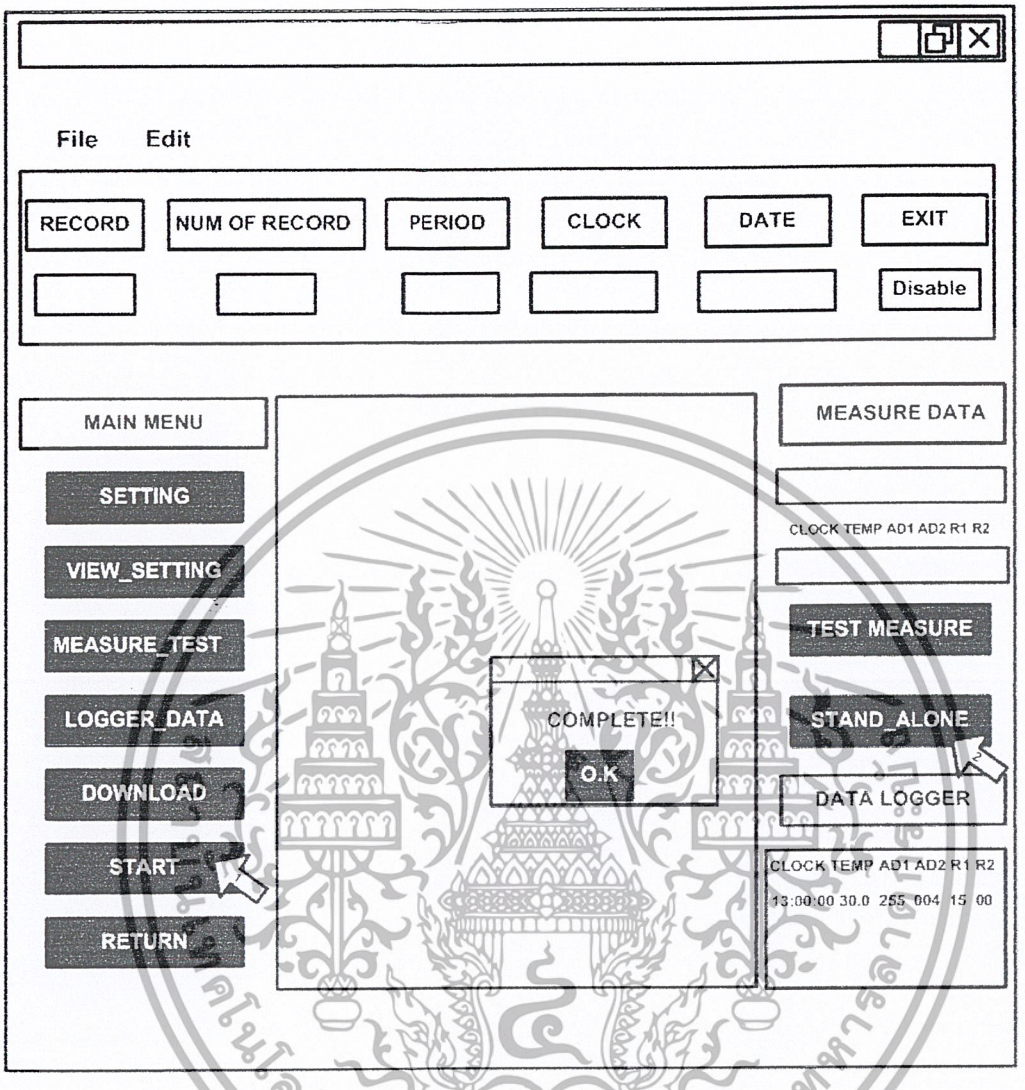
2. ให้ทำการตรวจสอบก่อนว่ายังมีการบันทึกค่าในสภาวะ Stand-alone อยู่หรือไม่ โดยการกดปุ่มตรวจสอบ **STAND_ALONE** สังเกตผลพบว่ายังมีการเก็บข้อมูลเดิมอยู่พบจากข้อความที่หน้าจอดังรูปที่ 9.2




**รูปที่ 9.2 ข้อความที่หน้าจอว่ายังมีข้อมูลเก่าที่ยังบันทึกไม่แล้วเสร็จ**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

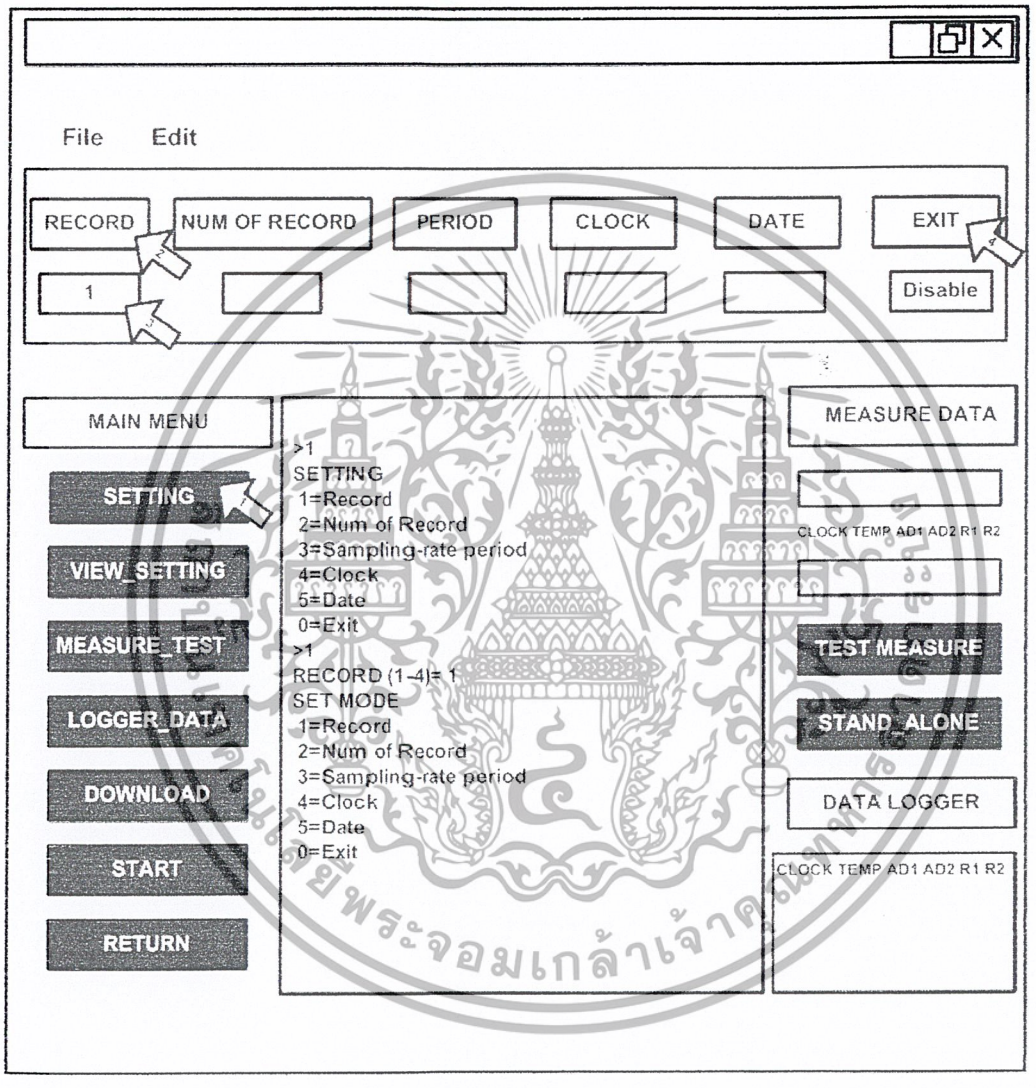
3. เมื่อรอจนข้อมูลเดิมบันทึกครบถ้วนแล้วจะปรากฏข้อความดังรูปที่ 9.3




รูปที่ 9.3 ข้อความที่หน้าจอว่าข้อมูลเก่าที่ต้องการบันทึกครบถ้วนแล้ว

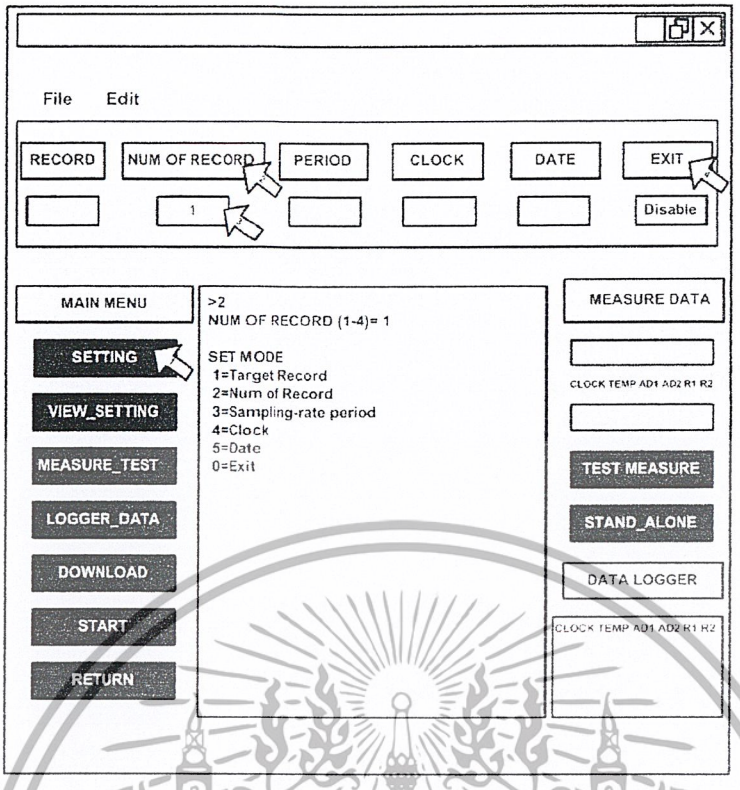
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. ตั้งค่าเริ่มต้นในการทำงานโดยเข้าสู่โปรแกรมย่อย **SETTING** ( กดที่ปุ่ม **SETTING** ที่หน้าจอ) โดยเริ่มที่ **Record 1 , Num of Record 1, Sampling-rate period (PER) 000** และตั้งค่าเวลา **13:00:00 น.** วันที่ทำการบันทึก **01/01/03** ตามวันเวลาจริง จากนั้นปรับระดับแรงดันที่ต้องการวัดไว้ที่ 0 โวลต์ทั้งสองช่องรับสัญญาณแรงดันอนาลอกอินพุท จะปรากฏผลที่หน้าจอดังรูปที่ 9.4 .1 – 9.4.5

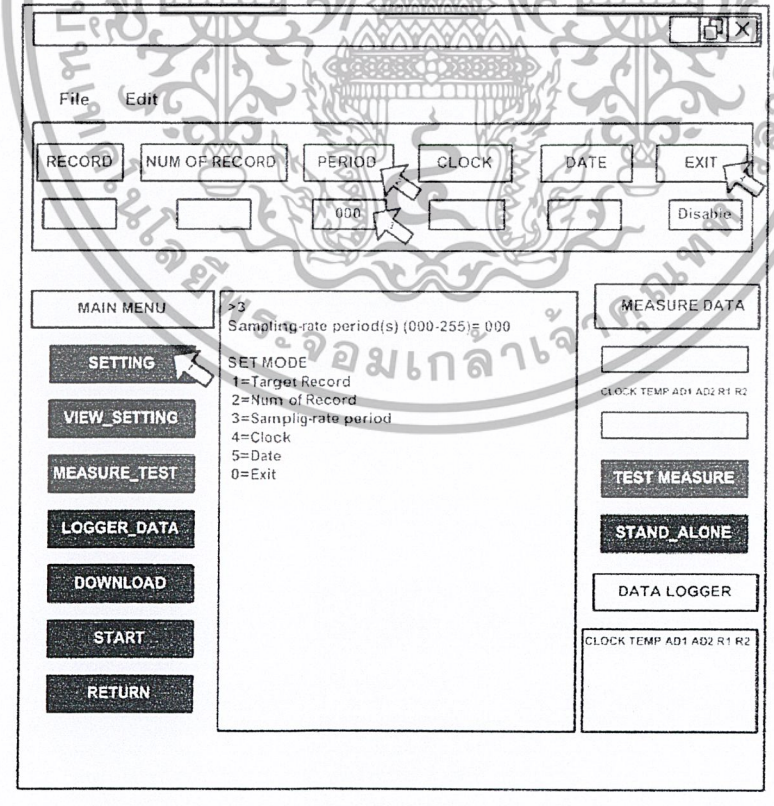



**รูปที่ 9.4.1 การตั้งค่า RECORD เริ่มต้น**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

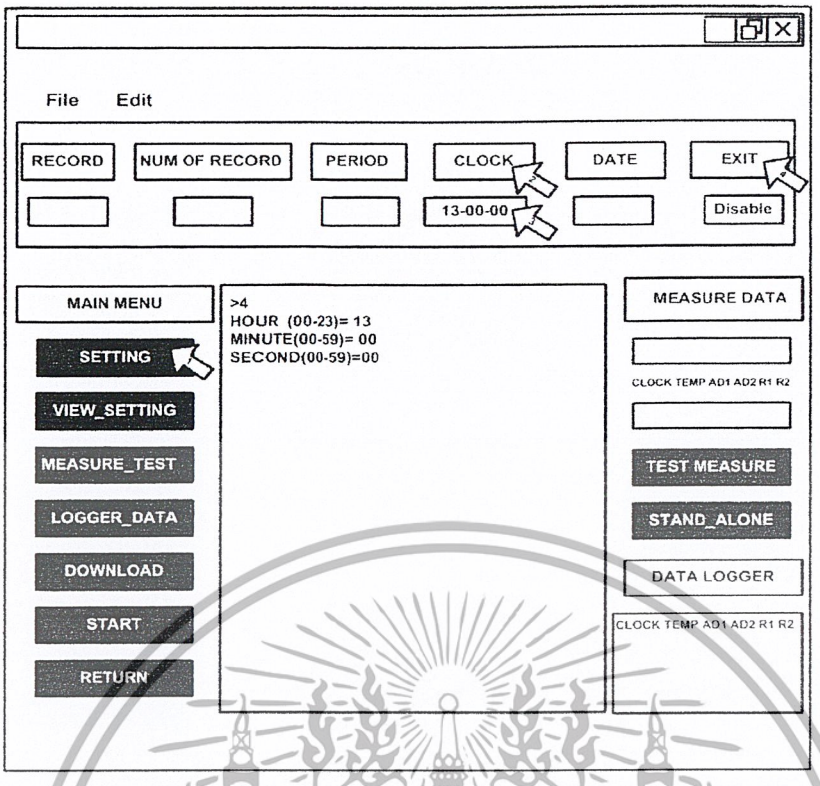



**รูปที่ 9.4.2 การตั้งค่าจำนวนของ NUM OF RECORD ที่ต้องการ**

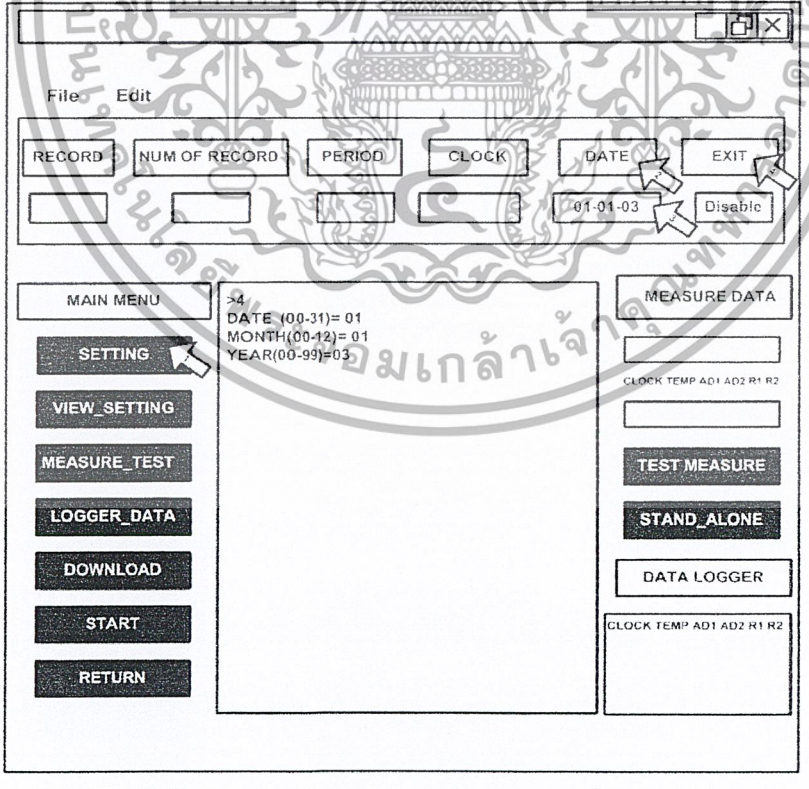



**รูปที่ 9.4.3 การตั้งค่า Sampling-rate period ในการบันทึก**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



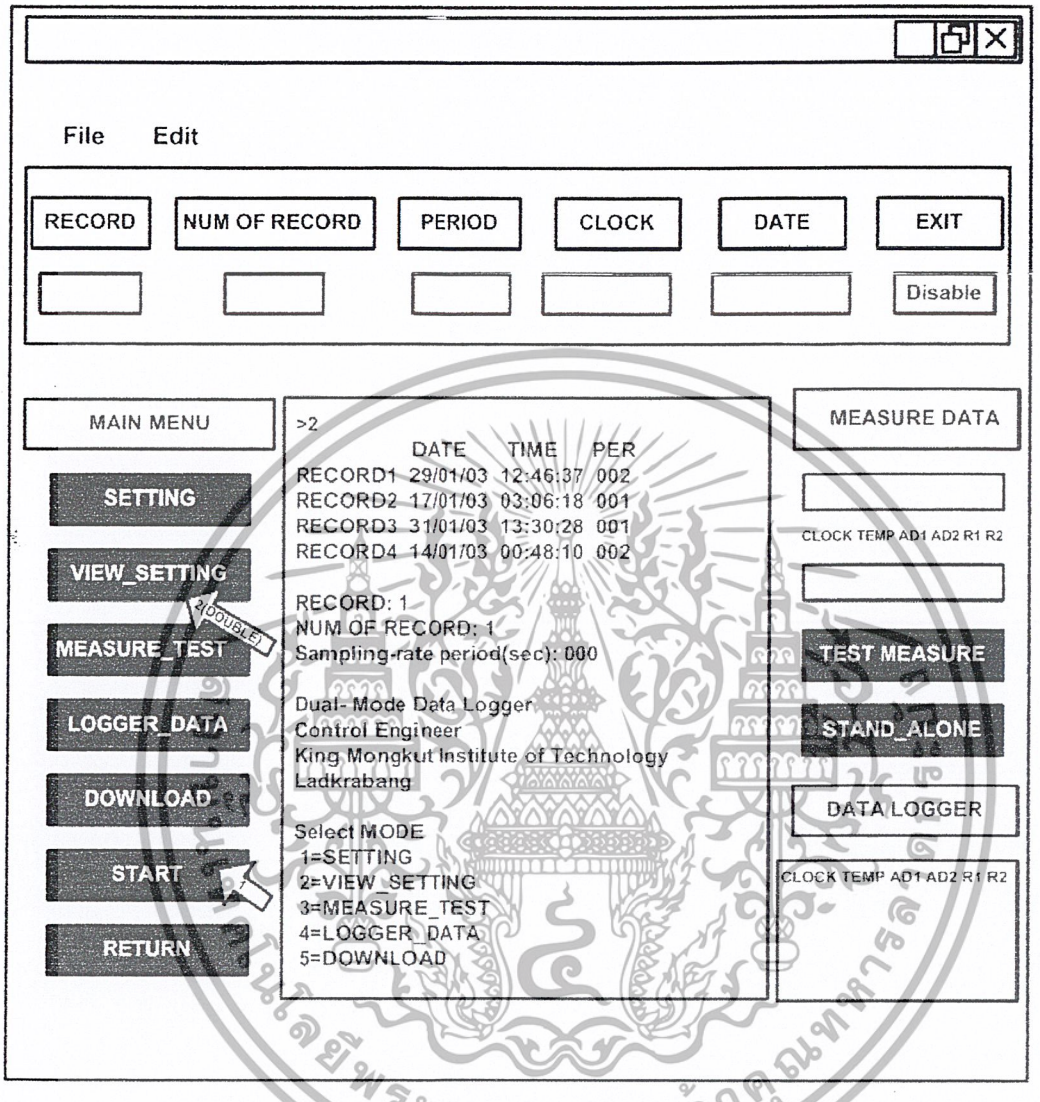

**รูปที่ 9.4.4 การตั้งค่าเวลาจริงขณะปฏิบัติงาน**




**รูปที่ 9.4.5 การตั้งวัน/เดือน/ปีตามเวลาปฏิบัติงานจริง**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

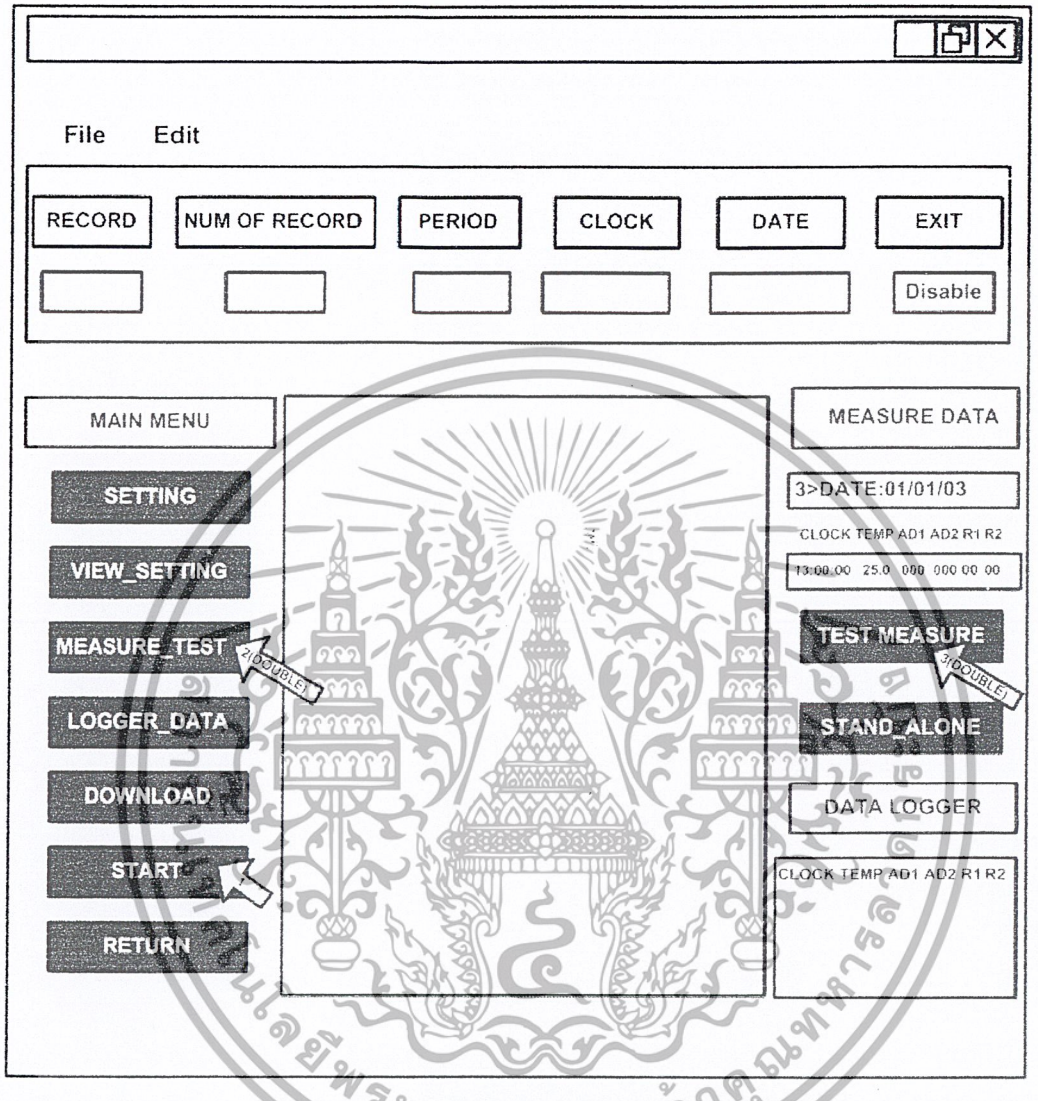
5. เมื่อเรียกดูค่าที่ตั้งไว้โดยเข้าสู่ **VIEW_SETTING** (กดที่ปุ่ม **VIEW_SETTING** ที่หน้าจอ) แล้วสังเกตข้อความที่ปรากฏที่หน้าจอได้ดังรูปที่ 9.5




**รูปที่ 9.5 การเรียกดูค่าที่ตั้งไว้ใน VIEW_SETTING**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

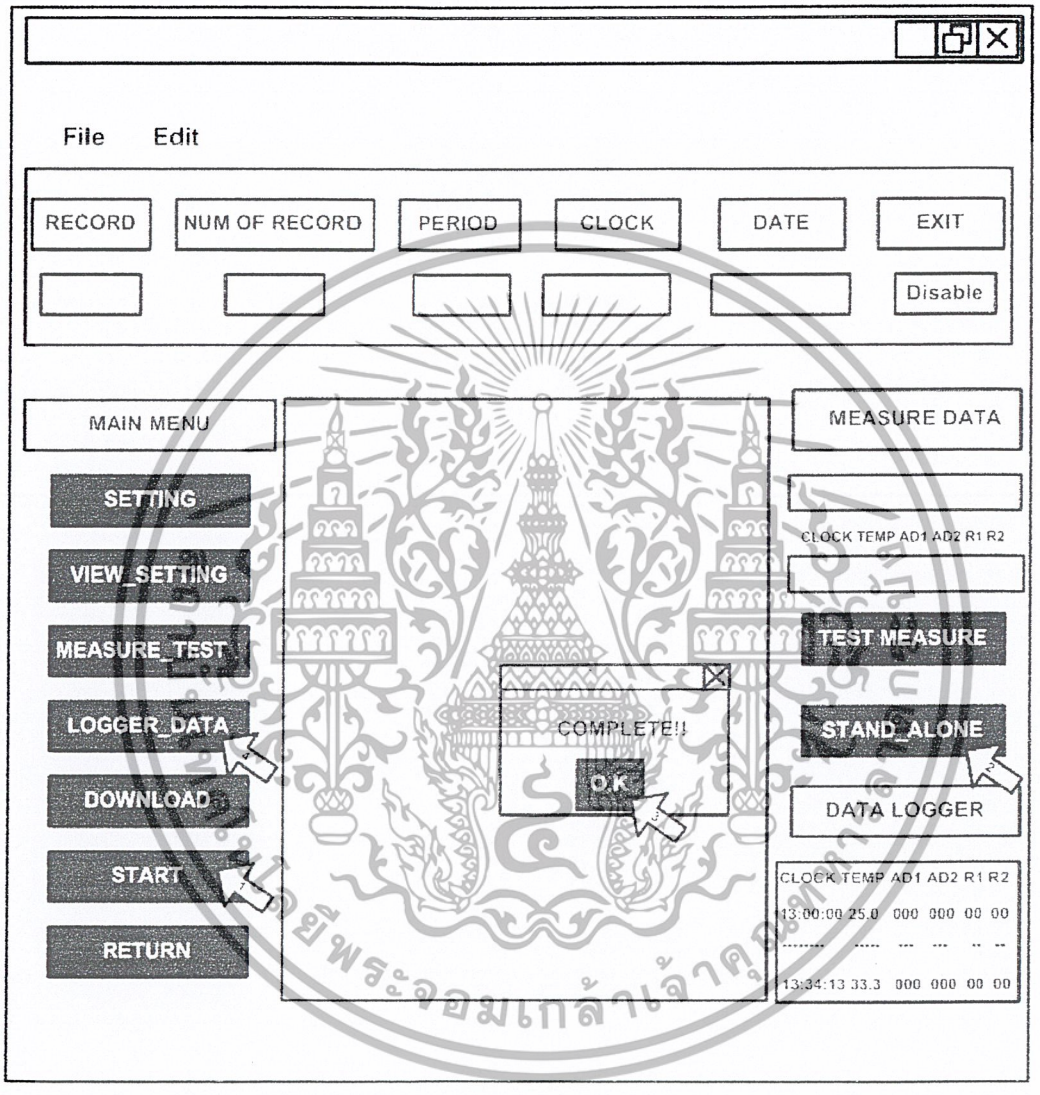
6. เมื่อทำการวัดค่าอุณหภูมิและสัญญาณทั้งสองช่องโดยเข้าสู่ **MEASURE_TEST** ( กดที่ปุ่ม **MEASURE_TEST** ที่หน้าจอ ) แล้วจะปรากฏข้อความที่หน้าจอดังรูปที่ 9.6




รูปที่ 9.6 การวัดค่าอุณหภูมิและสัญญาณทั้งสองช่องใน MEASURE_TEST

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

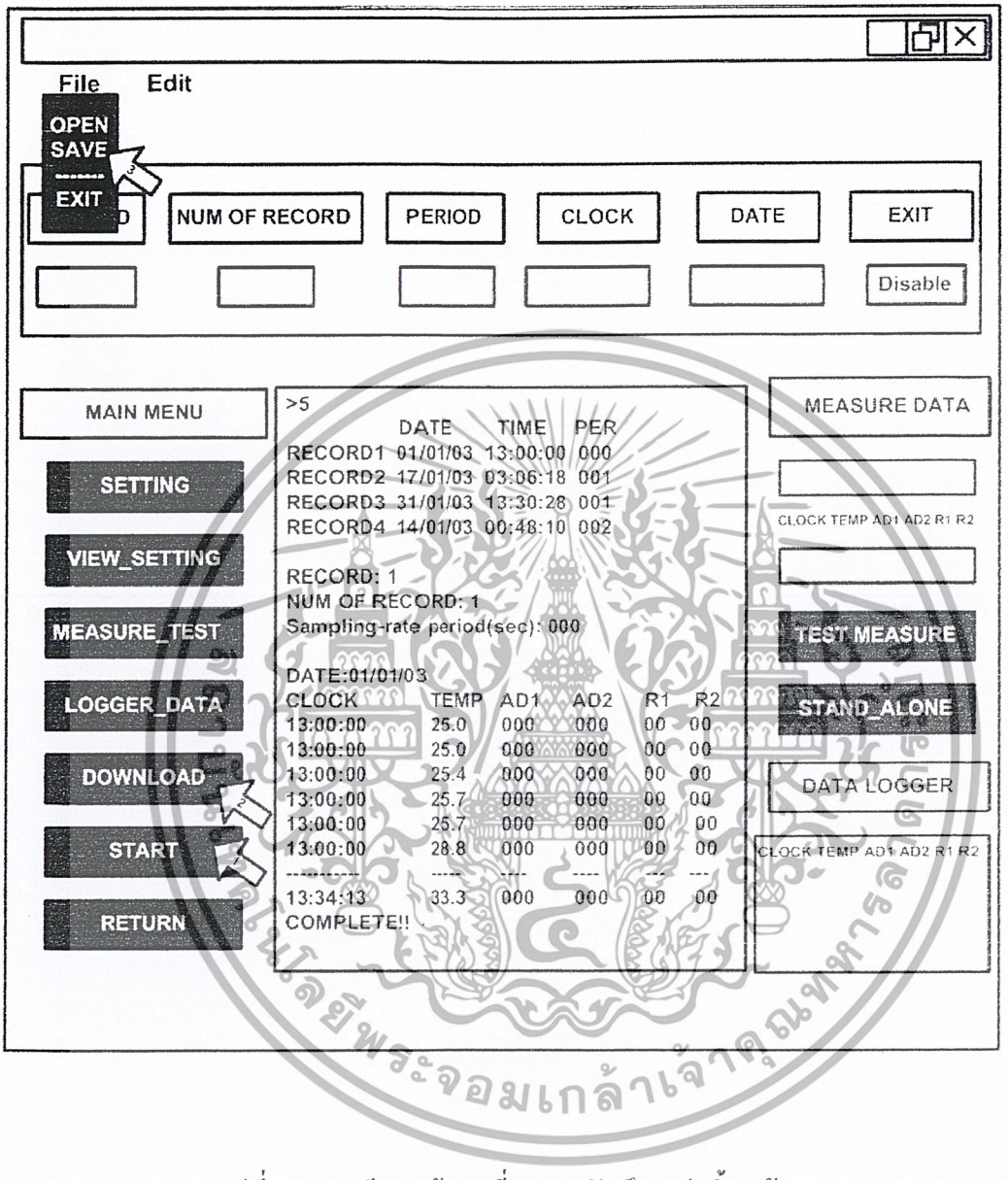
7. เมื่อทำการบันทึกค่าอุณหภูมิและสัญญาณทั้งสองช่องโดยเข้าสู่ **LOGGER_DATA** ( กดที่ปุ่ม **LOGGER_DATA** ที่หน้าจอ ) แล้วพบว่าที่หน้าปัดเครื่อง **LED 8 ตัว** จะวิ่งในลักษณะไปกลับ 2 รอบ พร้อมทั้ง **LED-Display** จะกระพริบทุกๆครั้งที่มีการเก็บข้อมูล อีกทั้งยังปรากฏข้อความที่หน้าจอดังรูปที่ 9.7




**รูปที่ 9.7 ข้อมูลอุณหภูมิและสัญญาณทั้งสองช่องที่ทำการบันทึก**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

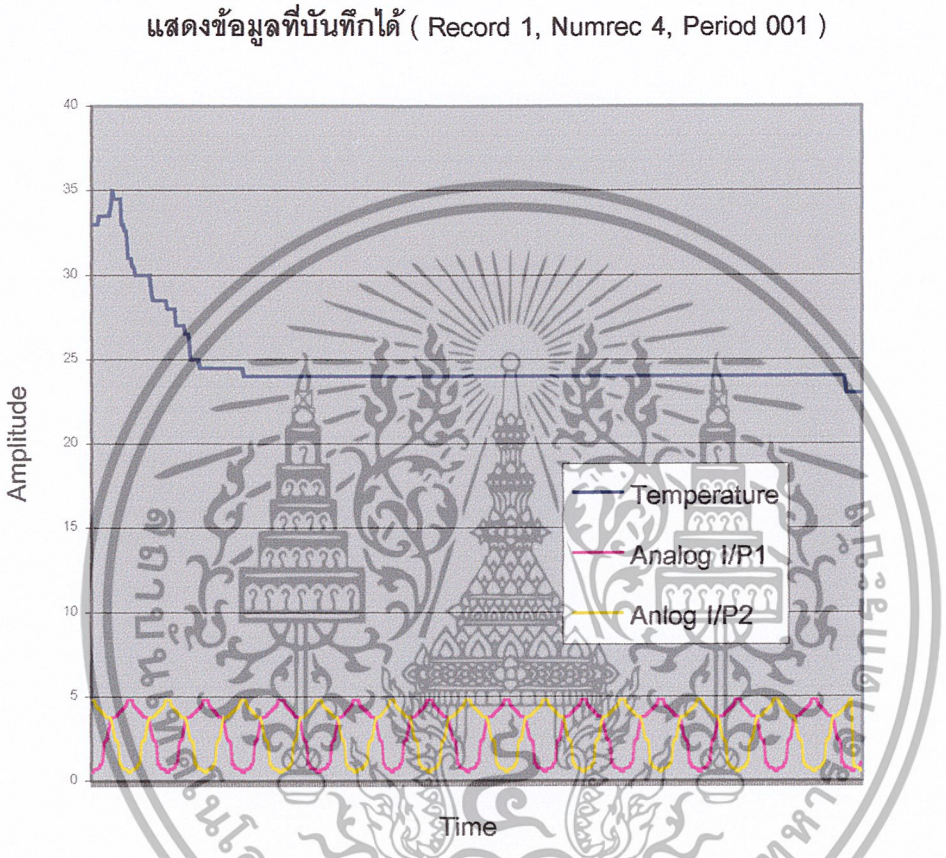
8. เมื่อทำการเรียกดูข้อมูลที่ทำการบันทึกค่าไว้โดยกดปุ่ม **DOWNLOAD** ที่หน้าจอจะพบข้อความปรากฏที่หน้าจอหลักดังรูปที่ 9.8




**รูปที่9.8 การเรียกดูข้อมูลที่ทำการบันทึกเสร็จสิ้นแล้ว**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

9. ทำการทดลองการวัดค่าอุณหภูมิและสัญญาณทั้งสองช่องสัญญาณใหม่ โดยทำการตั้งค่าเริ่มต้นทำงานเป็น **Record = 1, Num of Record = 4 ,Sampling-rate period = 001** และตั้งค่าเวลา,วันที่ทำการบันทึกตามวันและเวลาจริง
จากนั้นทำการบันทึกข้อมูลเป็น **textfile.text** แล้วนำไปวิเคราะห์ข้อมูลแบบกราฟโดยโปรแกรม **Microsoft Excel** แสดงดังรูปที่ 9.9




**รูปที่ 9.9 กราฟผลข้อมูลที่ได้จากการบันทึกค่า**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**สรุปและวิจารณ์ผลการทดลอง**

1. การเก็บข้อมูลที่ได้จากการวัดจะแบ่งการจัดเก็บได้เป็น 4 ชุดข้อมูล ( Record ) โดยมีขนาดของการเก็บข้อมูลในแต่ละ Record ได้ 2,048 ชุดข้อมูล ดังนั้นหากเราต้องการทำการบันทึกข้อมูลทั้งหมด 4 Record จึงสามารถบันทึกข้อมูลได้ถึง 8,192 ชุดข้อมูล
2. ระยะเวลาในการบันทึกข้อมูลสามารถเลือกได้จากค่าเริ่มต้นในการทำงาน เช่น

พิจารณาการเลือก Sampling-rate period = 001 ( 1 วินาทีต่อครั้ง )

| Record | Num of Record | Sampling-rate period | Time |
|---|---|---|---|
| 1 | 1 | 001 | 34.13 นาที |
| 1 | 2 | 001 | 1 ชั่วโมง 8 นาที 26วินาที |
| 1 | 3 | 001 | 1 ชั่วโมง 42 นาที 4วินาที |
| 1 | 4 | 001 | 2 ชั่วโมง 16 นาที 53วินาที |

พิจารณาการเลือก Sampling-rate period = 005 ( 5 วินาทีต่อครั้ง )

| Record | Num of Record | Sampling-rate period | Time |
|---|---|---|---|
| 1 | 1 | 005 | 2 ชั่วโมง 50 นาที 24วินาที |
| 1 | 2 | 005 | 5 ชั่วโมง 40 นาที 48วินาที |
| 1 | 3 | 005 | 8 ชั่วโมง 53 นาที 45วินาที |
| 1 | 4 | 005 | 11 ชั่วโมง 22 นาที 40วินาที |

นั่นหมายความว่า ถ้าต้องการเก็บข้อมูลตลอด 24 ชั่วโมง เราต้องจะทำการตั้งค่าเริ่มต้นในการทำงานไว้ที่

Record = 1, Num of Record = 4, Sampling-rate period = 012

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บรรณานุกรม


[1] ยืน ภู่วรวรรณ “ เทคโนยีฮาร์ดแวร์ไอบีเอ็มพีซี ” บริษัท ซีเอ็ดยูเคชั่น จำกัด พ.ศ. 2533

[2] สุนทร วิทูสุรพจน์ “ การใช้งานไมโครคอนโทรลเลอร์ตระกูล 8051 ” บริษัท ซีเอ็ดยูเคชั่น จำกัด พ.ศ. 2533

[3] ธีรวัฒน์ ประกอบผล “ การประยุกต์ใช้งานไมโครคอนโทรลเลอร์ ” สมาคมส่งเสริมเทคโนโลยี ( ไทย-ญี่ปุ่น ) พ.ศ. 2540

[4] กิตติ องค์คุณารักษ์ “ แนะนำแอสแซมบลี ” บริษัท ซีเอ็ดยูเคชั่น จำกัด พ.ศ. 2537


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# ภาคผนวก ก.

## โปรแกรมควบคุม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
;*****************************************
;*    Dual-Mode  Data Logger    *
;*        Used with AT89S8252   *
;*        Control Engineering   *
;*****************************************

; Parameter definitions
BAUD            EQU  256-6          ; 256-1 -> baud rate = 57,600 bps
                                    ; 256-3 -> baud rate = 19,200 bps
                                    ; 256-6 -> baud rate = 9,600 bps
                                    ; 256-24 -> baud rate = 2,400 bps
ADDR24          EQU  0A0h           ; device address for 24C256
EEPROM
ADDR13          EQU  0D0h           ; device address for DS1307 RTC
YEAR            EQU  06h
MONTH           EQU  05h
DATE            EQU  04h
DAY             EQU  03h
HOUR            EQU  02h
MINUTE          EQU  01h
SECOND          EQU  00h
NUMREC          EQU  3Eh
TARGET          EQU  3Fh

; Register definitions
INDEX           EQU  0              ; index register
ZDATA           EQU  1              ; data transfer register

; I/O pin definitions
SCL             BIT  P1.0           ; serial clock
SDA             BIT  P1.1           ; serial data
RST             BIT  P1.5           ; reset DS1620
CLK             EQU  P3.4           ;clock(SCL)
DIO             EQU  P3.5           ;data in & data out(SDA)
CS              BIT  P3.3           ; chip select LTC1298
SW              BIT  P1.4           ; switch input
LED             BIT  P1.2           ; LED output
MODE            BIT  P1.3           ; mode select
DRIVER_LE       BIT  P1.6

; Memory allocations
STACK           EQU  07h            ; first stack
ADDR_RTC        EQU  30h            ; RTC address register
ADDR_LO         EQU  31h            ; 2-byte EEPROM address register
ADDR_HI         EQU  32h            ;
DATA_LO         EQU  33h
DATA_HI         EQU  34h
COUNT      EQU  35h
BUFF1           EQU  36h
BUFF2           EQU  37h
BUFFD           EQU  38h
HOU             EQU  39h
MIN             EQU  3Ah
SEC             EQU  3Bh
PER             EQU  3Ch
NUM             EQU  3Dh
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                ORG     0000h                   ; power-on reset vector
ON_RESET:
                mov     SP, #STACK
                mov     PCON,#00h
                mov     TMOD,#20h
                mov     TH1,#0FBH
                mov     TL1,#BAUD
                setb    TR1
                mov     SCON,#52h

                setb    CS
                clr     RST
                setb    SDA
                setb    SCL

                clr     LED
                lcall   DELAY

MAIN:
                setb    LED
                lcall   PRINT_SER
                db      0Dh,0Ah,0Dh,0Ah
                db      "Dual-Mode Data Logger 2 Mode",0Dh,0Ah
                db      "Control Engineer",0Dh,0Ah
                db      "Faculty of Engineering",0Dh,0Ah
                db      "King Mongkut Institute of Technology Ladkrabang",0Dh,0Ah
                db      0Dh,0Ah
                db      "Select MODE",0Dh,0Ah
                db      " 1=SETTING",0Dh,0Ah
                db      " 2=VIEW_SETTING",0Dh,0Ah
                db      " 3=MEASURE_TEST",0Dh,0Ah
                db      " 4=LOGGER_DATA",0Dh,0Ah
                db      " 5=DOWNLOAD",0Dh,0Ah
                db      0Dh,0Ah
                db      ">",0

        READY:
                jnb     SCON.0, CHK_SW
                lcall   RX
        CHK_KEY1:
                cjne    A,#31h,CHK_KEY2
                mov     A,SBUF
                lcall   TX
                lcall   SET_MODE
                jmp     MAIN
        CHK_KEY2:
                cjne    A,#32h,CHK_KEY3
                mov     A,SBUF
                lcall   TX
                lcall   VIEW_MODE
                jmp     MAIN
        CHK_KEY3:
                cjne    A,#33h,CHK_KEY4
                mov     A,SBUF
                lcall   TX
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                lcall   MEASURE_MODE
                jmp     MAIN
        CHK_KEY4:
                cjne    A,#34h,CHK_KEY5
                mov     A,SBUF
                lcall   TX
                lcall   Display
                lcall   LOGGER_MODE
                mov     P0,#0FFh
                lcall   DELAY_100ms
                mov     P0,#0Fh
                lcall   DELAY_100ms
                mov     P0,0F0h
                lcall   DELAY_100ms
                mov     P0,#00h
                lcall   DELAY_100ms
                mov     P0,#0FFh
                lcall   DELAY_100ms
                mov     P0,#00h
                lcall   DELAY_100ms
                mov     P0,#0FFh
                lcall   DELAY_100ms
                jmp     MAIN
        CHK_KEY5:
                cjne    A,#35h,CHK_SW
                mov     A,SBUF
                lcall   TX
                lcall   DUMP_MODE
                jmp     MAIN

        CHK_SW:
                JB      SW,READY
                mov     A,#34h
                lcall   TX
                lcall   LOGGER_MODE
                jmp     MAIN

SET_MODE:
        lcall           PRINT_SER
        db              0Dh,0Ah,0Dh,0Ah
        db              "SETTING",0Dh,0Ah
        db              " 1=Record",0Dh,0Ah
        db              " 2=Num of Record",0Dh,0Ah
        db              " 3=Sampling-rate period",0Dh,0Ah
        db              " 4=Clock",0Dh,0Ah
        db              " 5=Date",0Dh,0Ah
        db              " 0=Exit",0Dh,0Ah
        db              0Dh,0Ah
        db              ">",0
        SET_SLC:
                lcall   RX
        CHK_KEY1S:
                cjne    A,#31h,CHK_KEY2S
                mov     A,SBUF
                lcall   TX
                lcall   SET_TARGET
                jmp     SET_MODE
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        CHK_KEY2S:
                cjne    A,#32h,CHK_KEY3S
                mov     A,SBUF
                lcall   TX
                lcall   SET_NUMREC
                jmp     SET_MODE
        CHK_KEY3S:
                cjne    A,#33h,CHK_KEY4S
                mov     A,SBUF
                lcall   TX
                call    SET_PERIOD
                jmp     SET_MODE
        CHK_KEY4S:
                cjne    A,#34h,CHK_KEY5S
                mov     A,SBUF
                lcall   TX
                call    SET_CLOCK
                jmp     SET_MODE
        CHK_KEY5S:
                cjne    A,#35h,CHK_KEY0S
                mov     A,SBUF
                lcall   TX
                lcall   SET_DATE
                jmp     SET_MODE
        CHK_KEY0S:
                cjne    A,#30h,SET_SLC
                ret

SET_TARGET:
                lcall   LINE_FEED
                lcall   PRINT_SER
                db      "RECORD (1-4)= ",0
                lcall   RX
                lcall   TX
                dec     A
                anl     A,#03h
                mov     ZDATA,A
                mov     ADDR_RTC,#TARGET
                lcall   WRITE_RTC
                ret

SET_NUMREC:
                lcall   LINE_FEED
                lcall   PRINT_SER
                db      "NUM OF RECORD (1-4)= ",0
                lcall   RX
                lcall   TX
                dec     A
                anl     A,#03h
                mov     ZDATA,A
                mov     ADDR_RTC,#NUMREC
                lcall   WRITE_RTC
                ret

SET_PERIOD:
                lcall   LINE_FEED
                lcall   PRINT_SER
                db      "Sampling-rate period(s) (000-255)= ",0
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        lcall   RX
        lcall   TX
        anl     A,#0Fh
        mov     B,#100
        mul     AB
        mov     ZDATA,A
        lcall   RX
        lcall   TX
        anl     A,#0Fh
        mov     B,#10
        mul     AB
        add     A,ZDATA
        mov     ZDATA,A
        lcall   RX
        lcall   TX
        anl     A,#0Fh
        add     A,ZDATA
        mov     ZDATA,A
        mov     ADDR_RTC,#TARGET
        lcall   READ_RTC
        anl     A,#03h
        mov     B,#8
        mul     AB
        add     A,#14
        mov     ADDR_RTC,A
        lcall   WRITE_RTC
        ret

SET_CLOCK:
        lcall   LINE_FEED
        lcall   PRINT_SER
        db      "HOUR  (00-23)= ",0
        lcall   RX_NUM
        mov     ZDATA,A
        mov     ADDR_RTC,#HOUR
        lcall   WRITE_RTC
        lcall   LINE_FEED
        lcall   PRINT_SER
        db      "MINUTE(00-59)= ",0
        lcall   RX_NUM
        mov     ZDATA,A
        mov     ADDR_RTC,#MINUTE
        lcall   WRITE_RTC
        lcall   LINE_FEED
        lcall   PRINT_SER
        db      "SECOND(00-59)= ",0
        lcall   RX_NUM
        mov     ZDATA,A
        mov     ADDR_RTC,#SECOND
        lcall   WRITE_RTC
        lcall   RTC_CLOCK_ENABLE
        lcall   CONFIG_1620
        ret

SET_DATE:
        lcall   LINE_FEED
        lcall   PRINT_SER
        db      "DATE (01-31)= ",0
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        lcall   RX_NUM
        mov     ZDATA,A
        mov     ADDR_RTC,#DATE
        lcall   WRITE_RTC
        lcall   LINE_FEED
        lcall   PRINT_SER
        db      "MONTH(01-12)= ",0
        lcall   RX_NUM
        mov     ZDATA,A
        mov     ADDR_RTC,#MONTH
        lcall   WRITE_RTC
        lcall   LINE_FEED
        lcall   PRINT_SER
        db      "YEAR (00-99)= ",0
        lcall   RX_NUM
        mov     ZDATA,A
        mov     ADDR_RTC,#YEAR
        lcall   WRITE_RTC
        ret

VIEW_MODE:
        lcall   LINE_FEED
        lcall   PRINT_SER
        db      "DATE    TIME    PER",0
        mov     COUNT,#4
    LPV1:
        lcall   LINE_FEED
        lcall   PRINT_SER
        db      "RECORD",0
        mov     A,#4
        clr     C
        subb    A,COUNT
        add     A,#31h
        lcall   TX
        clr     C
        subb    A,#31h
        mov     B,#8
        mul     AB
        add     A,#8
        mov     ADDR_RTC,a
        mov     A,#20h
        lcall   TX
        lcall   TX
        lcall   READ_RTC
        inc     ADDR_RTC
        lcall   TX_BYTE
        mov     A,#'/'
        lcall   TX
        lcall   READ_RTC
        inc     ADDR_RTC
        lcall   TX_BYTE
        mov     A,#'/'
        lcall   TX
        lcall   READ_RTC
        inc     ADDR_RTC
        lcall   TX_BYTE
        mov     A,#20h
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        lcall   TX
        lcall   TX
        lcall   READ_RTC
        inc     ADDR_RTC
        lcall   TX_BYTE
        mov     A,#':'
        lcall   TX
        lcall   READ_RTC
        inc     ADDR_RTC
        lcall   TX_BYTE
        mov     A,#':'
        lcall   TX
        lcall   READ_RTC
        inc     ADDR_RTC
        lcall   TX_BYTE
        mov     A,#20h
        lcall   TX
        lcall   TX
        lcall   READ_RTC
        lcall   TX_BASE10
        djnz    COUNT,LPV2
jmp LPV3
    LPV2:
        jmp     LPV1
    LPV3:
        lcall   LINE_FEED
        lcall   LINE_FEED
        lcall   PRINT_SER
        db      "RECORD: ",0
        mov     ADDR_RTC,#TARGET
        lcall   READ_RTC
        anl     A,#03h
        mov     BUFF1,a
        add     A,#31h
        lcall   TX
        lcall   LINE_FEED
        lcall   PRINT_SER
        db      "NUM OF RECORD: ",0
        mov     ADDR_RTC,#NUMREC
        lcall   READ_RTC
        anl     A,#03h
        add     A,#31h
        lcall   TX
        lcall   LINE_FEED
        lcall   PRINT_SER
        db      "PERIOD(sec): ",0
        mov     A,BUFF1
        mov     B,#8
        mul     AB
        add     A,#14
        mov     ADDR_RTC,a
        lcall   READ_RTC
        mov     PER,a
        lcall   TX_BASE10
        ret
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
MEASURE_MODE:
        lcall   START_1620
        lcall   LINE_FEED
        lcall   PRINT_SER
        db      "DATE: ",0
        mov     ADDR_RTC,#DATE
        lcall   READ_RTC
        lcall   TX_BYTE
        mov     A,#'/'
        lcall   TX
        mov     ADDR_RTC,#MONTH
        lcall   READ_RTC
        lcall   TX_BYTE
        mov     A,#'/'
        lcall   TX
        mov     ADDR_RTC,#YEAR
        lcall   READ_RTC
        lcall   TX_BYTE

        lcall   LINE_FEED
        lcall   PRINT_SER
        db      "CLOCK   TEMP AD1 AD2 R1 R2",0
        lcall   LINE_FEED
LPM:
        mov     A,#0Dh
        lcall   TX
        mov     ADDR_RTC,#HOUR
        lcall   READ_RTC
        lcall   TX_BYTE
        mov     A,#':'
        lcall   TX
        mov     ADDR_RTC,#MINUTE
        lcall   READ_RTC
        lcall   TX_BYTE
        mov     A,#':'
        lcall   TX
        mov     ADDR_RTC,#SECOND
        lcall   READ_RTC
        lcall   TX_BYTE

        mov     A,#20h
        lcall   TX
        lcall   TX
        lcall   READ_1620
        mov     ZDATA,a
        clr     C
        subb    A,#20h
        jnc     JUMP1
        cpl     A
        inc     A
        push    ACC
        mov     A,#'-'
        lcall   TX
        pop     ACC
JUMP1:
        clr     C
        rrc     A
        mov     B,#10
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        div     AB
        swap    A
        add     A,B
        lcall   TX_BYTE
        mov     A,#'.'
        lcall   TX
        mov     A,ZDATA
        rrc     A
        mov     A,#30h
        jnc     JUMP2
        add     A,#5
JUMP2:
        lcall   TX

        mov     A,#20h
        lcall   TX
        lcall   TX
        clr     C
        lcall   READ_ADC
        mov     A,DATA_LO
        swap    A
        mov     BUFFD,A
        mov     A,DATA_HI
        lcall   TX_BASE10

        mov     A,#20h
        lcall   TX
        lcall   TX
        setb    C
        lcall   READ_ADC
        mov     A,DATA_LO
        add     A,BUFFD
        mov     ZDATA,A
        mov     A,DATA_HI
        lcall   TX_BASE10
        lcall   SHOW_REM
        jnb     SCON.0,JUMP3
        clr     SCON.0
        ret
JUMP3:
        lcall   DELAY
        jmp     LPM


LOGGER_MODE:
        lcall   START_1620
        lcall   LINE_FEED
        lcall   PRINT_SER
        db      "RECORD: ",0
        lcall   Display
        mov     A,#00h
        mov     P0,A

        mov     ADDR_RTC,#TARGET
        lcall   READ_RTC
        anl     A,#03h
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        mov     BUFF1,A
        add     A,#31h
        lcall   TX
        lcall   LINE_FEED
        lcall   PRINT_SER
        db      "NUM OF RECORD: ",0
        lcall   Display
        mov     A,#00h
        mov     P0,A

        mov     ADDR_RTC,#NUMREC
        lcall   READ_RTC
        anl     A,#03h
        add     A,#31h
        lcall   TX
        lcall   LINE_FEED
        lcall   PRINT_SER
        db      "PERIOD(sec): ",0
        lcall   Display
        mov     A,#00h
        mov     P0,A

        mov     A,BUFF1
        mov     B,#8
        mul     AB
        add     A,#14
        mov     ADDR_RTC,a
        lcall   READ_RTC
        mov     PER,A
        lcall   TX_BASE10
LOGGER:
        lcall   LINE_FEED
        lcall   LINE_FEED
        mov     ADDR_RTC,#TARGET
        lcall   READ_RTC
        anl     A,#03h
        mov     B,#8
        mul     AB
        add     A,#8
        mov     BUFF1,A
        lcall   PRINT_SER
        db      "DATE: ",0
        mov     ADDR_RTC,#DATE
        lcall   READ_RTC
        mov     ZDATA,A
        lcall   TX_BYTE
        mov     ADDR_RTC,BUFF1
        lcall   WRITE_RTC
        inc     BUFF1
        mov     A,#'/'
        lcall   TX
        mov     ADDR_RTC,#MONTH
        lcall   READ_RTC
        mov     ZDATA,A
        lcall   TX_BYTE
        mov     ADDR_RTC,BUFF1
        lcall   WRITE_RTC
        inc     BUFF1
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        mov     A,#'/'
        lcall   TX
        mov     ADDR_RTC,#YEAR
        lcall   READ_RTC
        mov     ZDATA,A
        lcall   TX_BYTE
        mov     ADDR_RTC,BUFF1
        lcall   WRITE_RTC
        inc     BUFF1

        mov     ADDR_RTC,#HOUR
        lcall   READ_RTC
        mov     ZDATA,A
        mov     ADDR_RTC,BUFF1
        lcall   WRITE_RTC
        inc     BUFF1
        mov     ADDR_RTC,#MINUTE
        lcall   READ_RTC
        mov     ZDATA,A
        mov     ADDR_RTC,BUFF1
        lcall   WRITE_RTC
        inc     BUFF1
        mov     ADDR_RTC,#SECOND
        lcall   READ_RTC
        mov     ZDATA,A
        mov     ADDR_RTC,BUFF1
        lcall   WRITE_RTC
        mov     ADDR_RTC,#TARGET
        lcall   READ_RTC
        anl     A,#03h
        rl      A
        swap    A
        mov     ADDR_HI,A
        mov     ADDR_LO,#0
        mov     INDEX,#0
        mov     ADDR_RTC,#NUMREC
        lcall   READ_RTC
        inc     A
        mov     NUM,A

        lcall   LINE_FEED
        lcall   PRINT_SER
        db      "CLOCK   TEMP AD1 AD2 R1 R2",0
LPL:
        clr     LED
        lcall   LINE_FEED
        mov     ADDR_RTC,#HOUR
        lcall   READ_RTC
        lcall   TX_BYTE
        mov     A,#':'
        lcall   TX
        mov     ADDR_RTC,#MINUTE
        lcall   READ_RTC
        lcall   TX_BYTE
        mov     A,#':'
        lcall   TX
        mov     ADDR_RTC,#SECOND
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        lcall   READ_RTC
        lcall   TX_BYTE
        mov     A,#20h
        lcall   TX
        lcall   TX
        lcall   read_1620
        mov     ZDATA,A
WA1:
        lcall   write_byte
        jc      WA1
        inc     ADDR_LO
        mov     A,ZDATA
        clr     C
        subb    A,#20h
        jnc     SKIP1
        cpl     A
        inc     A
        push    ACC
        mov     A,#'-'
        lcall   TX
        pop     ACC
SKIP1:
        clr     C
        rrc     A
        mov     B,#10
        div     AB
        swap    A
        add     A,B
        lcall   TX_BYTE
        mov     A,#'.'
        lcall   TX
        mov     A,ZDATA
        rrc     A
        mov     A,#30h
        jnc     SKIP2
        add     A,#5
SKIP2:
        lcall   TX
        mov     A,#20h
        lcall   TX
        lcall   TX
        clr     C
        lcall   READ_ADC
        mov     ZDATA,DATA_HI
WA2:
        lcall   WRITE_BYTE
        jc      WA2
        inc     ADDR_LO
        mov     A,DATA_LO
        swap    A
        mov     BUFFD,A
        mov     A,DATA_HI
        lcall   TX_BASE10
        mov     A,#20h
        lcall   TX
        lcall   TX
        setb    C
        lcall   READ_ADC
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        mov     ZDATA,DATA_HI
WA3:
        lcall   WRITE_BYTE
        jc      WA3
        inc     ADDR_LO
        mov     A,DATA_LO
        add     A,BUFFD
        mov     ZDATA,a
        mov     A,DATA_HI
        lcall   TX_BASE10
WA4:
        lcall   WRITE_BYTE
        jc      WA4
        inc     ADDR_LO
        lcall   SHOW_REM
        clr     A
        cjne    A,ADDR_LO,SKIP3
        inc     ADDR_HI
        inc     INDEX
        mov     A,INDEX
        cjne    A,#20h,SKIP3
        mov     INDEX,#0
        djnz    NUM,SKIP3
        lcall   LINE_FEED
        lcall   PRINT_SER
        db      "COMPLETE!",0
        setb    LED
        jnb     SCON.0,$
        clr     SCON.0
        ret
SKIP3:
        setb    LED
        jnb     SCON.0,SKIP4
        clr     SCON.0
        ret
SKIP4:
        lcall   WAIT
        jmp     LPL

DUMP_MODE:
        lcall   VIEW_MODE
DUMP:
        lcall   LINE_FEED
        lcall   LINE_FEED
        lcall   PRINT_SER
        db      "DATE: ",0
        mov     ADDR_RTC,#TARGET
        lcall   READ_RTC
        anl     A,#03h
        mov     B,#8
        mul     AB
        add     A,#8
        mov     ADDR_RTC,A
        lcall   READ_RTC
        inc     ADDR_RTC
        lcall   TX_BYTE
        mov     A,#'/'
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        lcall   TX
        lcall   READ_RTC
        inc     ADDR_RTC
        lcall   TX_BYTE
        mov     A,#'/'
        lcall   TX
        lcall   READ_RTC
        lcall   TX_BYTE
        inc     ADDR_RTC
        lcall   READ_RTC
        mov     HOU,A
        inc     ADDR_RTC
        lcall   READ_RTC
        mov     MIN,A
        inc     ADDR_RTC
        lcall   READ_RTC
        mov     SEC,A
        mov     ADDR_RTC,#TARGET
        lcall   READ_RTC
        anl     A,#03h
        rl      A
        swap    A
        mov     ADDR_HI,A
        mov     ADDR_LO,#0
        mov     INDEX,#0
        mov     ADDR_RTC,#NUMREC
        lcall   READ_RTC
        inc     A
        mov     NUM,A
        lcall   LINE_FEED
        lcall   PRINT_SER
        db      "CLOCK   TEMP  AD1  AD2  R1  R2",0
LPD:
        lcall   LINE_FEED
        mov     A,HOU
        lcall   TX_BYTE
        mov     A,#':'
        lcall   TX
        mov     A,MIN
        lcall   TX_BYTE
        mov     A,#':'
        lcall   TX
        mov     A,SEC
        lcall   TX_BYTE
        mov     A,#20h
        lcall   TX
        lcall   TX
        lcall   READ_RANDOM
        mov     ZDATA,A
        inc     ADDR_LO
        clr     C
        subb    A,#20h
        jnc     SKIP6
        cpl     A
        inc     A
        push    ACC
        mov     A,#'-'
        lcall   TX
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        pop     ACC
SKIP6:
        clr     C
        rrc     A
        mov     B,#10
        div     AB
        swap    A
        add     A,B
        lcall   TX_BYTE
        mov     A,#'.'
        lcall   TX
        mov     A,ZDATA
        rrc     A
        mov     A,#30h
        jnc     SKIP7
        add     A,#5
SKIP7:
        lcall   TX

        mov     A,#20h
        lcall   TX
        lcall   TX
        lcall   READ_RANDOM
        inc     ADDR_LO
        lcall   TX_BASE10

        mov     A,#20h
        lcall   TX
        lcall   TX
        lcall   READ_RANDOM
        inc     ADDR_LO
        lcall   TX_BASE10
        lcall   READ_RANDOM
        inc     ADDR_LO
        mov     ZDATA,A
        lcall   SHOW_REM
        clr     A
        cjne    A,ADDR_LO,SKIP8
        inc     ADDR_HI
        inc     INDEX
        mov     A,INDEX
        cjne    A,#20h,SKIP8
        mov     INDEX,#0
        djnz    NUM,SKIP8
        lcall   LINE_FEED
        lcall   PRINT_SER
        db      "COMPLETE!",0
        jnb     SCON.0,$
        clr     SCON.0
        ret
SKIP8:
        lcall   NEXT_TIME
        jnb     SCON.0,SKIP9
        clr     SCON.0
        ret
SKIP9:
        jmp     LPD
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
SHOW_REM:
        mov     A,#20h
        lcall   TX
        lcall   TX
        mov     A,ZDATA
        anl     A,#0F0h
        swap    A
        mov     B,#10
        div     AB
        swap    A
        add     A,B
        lcall   TX_BYTE

        mov     A,#20h
        lcall   TX
        lcall   TX
        mov     A,ZDATA
        anl     A,#0Fh
        mov     B,#10
        div     AB
        swap    A
        add     A,B
        lcall   TX_BYTE
        ret

LINE_FEED:
        mov     A,#0Dh
        lcall   TX
        mov     A,#0Ah
        lcall   TX
        ret

TX:
SE1:
        jnb     SCON.1,SE1
        clr     SCON.1
        mov     SBUF,A
        ret

RX:
RE1:
        jnb     SCON.0,RE1
        clr     SCON.0
        mov     A,SBUF
        ret

NUM2HEX:
        clr     C
        subb    A,#30h
        jc      N2HX
        subb    A,#10
        jnc     N2HX
        add     A,#10
        clr     C
        ret
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
N2HX:
        setb    C
        ret

RX_NUM:
RB1:
        lcall   RX
        lcall    NUM2HEX
        jc      RB1
        swap    A
        mov     B,A
        mov     A,SBUF
        lcall   TX
RB2:
        lcall   RX
        lcall   NUM2HEX
        jc      RB2
        orl     A,B
        mov     B,A
        mov     A,SBUF
        lcall   TX
        mov     A,B
        ret

HEX2CHR:
        push    ACC
        anl     A,#0Fh
        clr     C
        subb    A,#0Ah
        jnc     HC1
        add     A,#3Ah
        jmp     HC2
HC1:
        add     A,#41h
HC2:
        mov     B,A
        pop     ACC
        swap    A
        anl     A,#0Fh
        clr     C
        subb    A,#0Ah
        jnc     HC3
        add     A,#3Ah
        jmp     HC4
HC3:
        add     A,#41h
HC4:
        ret

TX_BYTE:
        lcall   HEX2CHR
        lcall   TX
        mov     A,B
        lcall   TX
        ret
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
PRINT_SER:
        pop     DPH
        pop     DPL
PS1:
        clr     A
        movc    A,@A+DPTR
        cjne    A,#00,PS2
        jmp     PSX
PS2:
        lcall   TX
        inc     DPTR
        jmp     PS1
PSX:
        push    DPL
        push    DPH
        ret

TX_BASE10:
        mov     B,#100
        div     AB
        add     A,#30h
        lcall   TX
        mov     A,B
        mov     B,#10
        div     AB
        add     A,#30h
        lcall   TX
        mov     A,B
        add     A,#30h
        lcall   TX
        ret

NEXT_TIME:
        mov     A,PER
        mov     B,#60
        div     AB
        mov     BUFF1,A
        mov     BUFF2,B
        mov     A,SEC
        lcall   BCD2HEX
        add     A,BUFF2
        clr     C
        subb    A,#60
        jnc     NT1
        add     A,#60
        lcall   HEX2BCD
        mov     SEC,A
        mov     A,MIN
        jmp     NT2
NT1:
        lcall   HEX2BCD
        mov     SEC,A
        mov     A,MIN
        inc     A
NT2:
        add     A,BUFF1
        lcall   BCD2HEX
        clr     C
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        subb    A,#60
        jnc     NT3
        add     A,#60
        lcall   HEX2BCD
        mov     MIN,A
        ret
NT3:
        lcall   HEX2BCD
        mov     MIN,A
        mov     A,HOU
        lcall   BCD2HEX
        inc     A
        cjne    A,#24,NT4
        mov     HOU,#0
        ret
NT4:
        lcall   HEX2BCD
        mov     HOU,A
        ret

BCD2HEX:
        push    ACC
        anl     A,#0F0h
        swap    A
        mov     B,#10
        mul     AB
        mov     B,A
        pop     ACC
        anl     A,#0Fh
        add     A,B
        ret

HEX2BCD:
        mov     B,#10
        div     AB
        swap    A
        add     A,B
        ret

DELAY:
        mov     R7,#0
DL1:
        mov     R6,#0
        djnz    R6,$
        djnz    R7,DL1
        ret

WAIT:
        clr     A
        cjne    A,PER,WT1
        ret
    WT1:
        mov     ADDR_RTC,#SECOND
        lcall   READ_RTC
        mov     BUFF1,A
        mov     COUNT,#0
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
WT2:
        lcall   READ_RTC
        cjne    A,BUFF1,WT3
        jmp     WT2
WT3:
        mov     BUFF1,A
        inc     COUNT
        mov     A,COUNT
        cjne    A,PER,WT2
        ret

WRITE_BYTE:
        mov     A,#ADDR24
        call    START
        jc      X49
        clr     ACC.0
        call    SHOUT
        jc      X48
        mov     A, ADDR_HI
        call    SHOUT
        jc      X48
        mov     A, ADDR_LO
        call    SHOUT
        jc      X48
        mov     A, ZDATA
        call    SHOUT
        jc      X48
        clr     C
X48:
        call    STOP
X49:
        ret

READ_CURRENT:
        call    START
        jc      X45
        setb    ACC.0
        call    SHOUT
        jc      X44
        call    SHIN
        call    NAK
        clr     C
X44:
        call    STOP
X45:
        ret

READ_RANDOM:
        mov     A,#ADDR24
        call    START
        jc      X47
        clr     ACC.0
        call    SHOUT
        jc      X46
        mov     A,ADDR_HI
        call    SHOUT
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        jc      X46
        mov     A, ADDR_LO
        call    SHOUT
        jc      X46
        mov     A,#ADDR24
        call    READ_CURRENT
        jmp     x47
X46:
        call    STOP
X47:
        ret

START:
        setb    SDA
        setb    SCL
        jnb     SDA, x40
        jnb     SCL, x40
        nop
        clr     SDA
        nop
        nop
        nop
        nop
        nop
        clr     SCL
        clr     C
        jmp     X41
X40:
        setb    C
X41:
        ret

STOP:
        clr     SDA
        nop
        nop
        setb    SCL
        nop
        nop
        nop
        nop
        nop
        setb    SDA
        ret

SHOUT:
        push    B
        mov     B,#8
X42:
        rlc     A
        mov     SDA,C
        nop
        setb    SCL
        nop
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        nop
        nop
        nop
        clr     SCL
        djnz    B,X42
        setb    SDA
        nop
        nop
        setb    SCL
        nop
        nop
        nop
        nop
        mov     C, SDA
        clr     SCL
        pop     B
        ret

SHIN:
        setb    SDA
        push    B
        mov     B, #8
X43:
        nop
        nop
        nop
        setb    SCL
        nop
        nop
        mov     C, SDA
        rlc     A
        clr     SCL
        djnz    B, x43
        pop     B
        ret

ACK:
        clr     SDA
        nop
        nop
        setb    SCL
        nop
        nop
        nop
        nop
        clr     SCL
        ret

NAK:
        setb    SDA
        nop
        nop
        setb    SCL
        nop
        nop
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                nop
                nop
                clr     SCL
                ret


RTC_CLOCK_ENABLE:
                mov     ADDR_RTC,#0
                call    READ_RTC
                clr     ACC.7
                mov     ZDATA,A
                call    WRITE_RTC
                ret


;RTC_CLOCK_DISABLE:
;               mov     ADDR_RTC,#0
;               call    READ_RTC
;               setb    ACC.7
;               mov     ZDATA,A
;               call    WRITE_RTC
;               ret

WRITE_RTC:
                mov     A,#ADDR13
                call    START
                jc      X59
                clr     ACC.0
                call    SHOUT
                jc      X58
                mov     A, ADDR_RTC
                call    SHOUT
                jc      X58
                mov     A, ZDATA
                call    SHOUT
                jc      X58
                clr     C
        X58:
                call    STOP
        X59:
                ret


READ_RTC:
                mov     A,#ADDR13
                call    START
                jc      X55
                clr     ACC.0
                call    SHOUT
                jc      X54
                mov     A, ADDR_RTC
                call    SHOUT
                jc      X54
                call    STOP
                mov     A,#ADDR13
                call    READ_CURRENT
                jmp     X55
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        X54:
                call    STOP
        X55:
                ret


READ_ADC:
                setb    CS
                clr     CS
                mov     A,#0dfh
                mov     ACC.5,C
                mov     B,#4
        x60:
                clr     SCL
                rlc     A
                mov     SDA,C
                setb    SCL
                djnz    B,X60
                setb    SDA
                mov     B,#9
        X61:
                clr     SCL
                setb    SCL
                mov     C,SDA
                rlc     A
                djnz    B,X61
                mov     DATA_HI,A
                clr     A
                mov     B,#4
        X62:
                clr     SCL
                setb    SCL
                mov     C,SDA
                rlc     A
                djnz    B,x62
                anl     A,#0Fh
                mov     DATA_LO,A
                setb    CS
                setb    SCL
                setb    SDA
                ret


CONFIG_1620:
                clr     RST
                setb    RST
                mov     A,#0ch
                mov     B,#8
        X64:
                clr     SCL
                rrc     A
                mov     SDA,C
                setb    SCL
                djnz    B,x64
                mov     A,#0ah
                mov     B,#8
        X65:
                clr     SCL
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        rrc     A
        mov     SDA,C
        setb    SCL
        djnz    B,x65
        setb    SCL
        setb    SDA
        clr     RST
        ret

START_1620:
        clr     RST
        setb    RST
        mov     A,#0eeh
        mov     B,#8
    x66:
        clr     SCL
        rrc     A
        mov     SDA,C
        setb    SCL
        djnz    B,x66
        setb    SCL
        setb    SDA
        clr     RST
        ret

READ_1620:
        clr     RST
        setb    RST
        mov     A,#0aah
        mov     B,#8
    X68:
        clr     SCL
        rrc     A
        mov     SDA,C
        setb    SCL
        djnz    B,x68
        setb    SDA
        mov     B,#8
    X69:
        clr     SCL
        mov     C,SDA
        rrc     A
        setb    SCL
        djnz    B,x69
        add     A,#20h
        setb    SCL
        setb    SDA
        clr     RST
        ret
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Display:        MOV     P0,#00000000B           ; Clear Databus
                SETB    DRIVER_LE               ; Enable Driver
Latch
                MOV     R0,#7                   ; Set 7 times loop
                MOV     A,#00000001B            ; Initial ACC. Value

LOOP_LEFT:      MOV     P0,A            ; Out ACC. Value to
DATABUS
                ACALL   DELAY_100ms             ; Delay 1 s
                RL      A                       ; Rotate ACC. Left
                DJNZ    R0,LOOP_LEFT            ; Do loop until 7
times
                MOV     R0,#7                   ; Set 7 times loop

LOOP_RIGHT:     MOV     P0,A                    ; Out ACC. Value to
DATABUS
                ACALL   DELAY_100ms             ; Delay 1 s
                RR      A               ; Rotate ACC. Right
                DJNZ    R0,LOOP_RIGHT           ; Do loop until 7
times
                RET

;-------------------------------------------------------
; Dummy Delay time 10ms,1s
;-------------------------------------------------------
DELAY_100ms:    MOV     R7,#100                 ; Do 100 times
DELAY_100ms_1:  MOV     R6,#0E6H                ; Each loop = 1 ms
DELAY_100ms_2:  NOP
                NOP
                DJNZ    R6,DELAY_100ms_2
                DJNZ    R7,DELAY_100ms_1
                RET

END
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# ภาคผนวก ข.

# คู่มือของไอซีที่ใช้งาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## Features

- Compatible with MCS-51™ Products
- 8K Bytes of In-System Reprogrammable Downloadable Flash Memory
  - SPI Serial Interface for Program Downloading
  - Endurance: 1,000 Write/Erase Cycles
- 2K Bytes EEPROM
  - Endurance: 100,000 Write/Erase Cycles
- 4V to 6V Operating Range
- Fully Static Operation: 0 Hz to 24 MHz
- Three-level Program Memory Lock
- 256 x 8-bit Internal RAM
- 32 Programmable I/O Lines
- Three 16-bit Timer/Counters
- Nine Interrupt Sources
- Programmable UART Serial Channel
- SPI Serial Interface
- Low-power Idle and Power-down Modes
- Interrupt Recovery From Power-down
- Programmable Watchdog Timer
- Dual Data Pointer
- Power-off Flag

## Description

The AT89S8252 is a low-power, high-performance CMOS 8-bit microcomputer with 8K bytes of downloadable Flash programmable and erasable read only memory and 2K bytes of EEPROM. The device is manufactured using Atmel's high-density nonvolatile memory technology and is compatible with the industry-standard 80C51 instruction set and pinout. The on-chip downloadable Flash allows the program memory to be reprogrammed in-system through an SPI serial interface or by a conventional nonvolatile memory programmer. By combining a versatile 8-bit CPU with downloadable Flash on a monolithic chip, the Atmel AT89S8252 is a powerful microcomputer which provides a highly-flexible and cost-effective solution to many embedded control applications.

The AT89S8252 provides the following standard features: 8K bytes of downloadable Flash, 2K bytes of EEPROM, 256 bytes of RAM, 32 I/O lines, programmable watchdog timer, two data pointers, three 16-bit timer/counters, a six-vector two-level interrupt architecture, a full duplex serial port, on-chip oscillator, and clock circuitry. In addition, the AT89S8252 is designed with static logic for operation down to zero frequency and supports two software selectable power saving modes. The Idle Mode stops the CPU while allowing the RAM, timer/counters, serial port, and interrupt system to continue functioning. The Power-down mode saves the RAM contents but freezes the oscillator, disabling all other chip functions until the next interrupt or hardware reset.

The downloadable Flash can be changed a single byte at a time and is accessible through the SPI serial interface. Holding RESET active forces the SPI bus into a serial programming interface and allows the program memory to be written to or read from unless Lock Bit 2 has been activated.

# 8-bit Microcontroller with 8K Bytes Flash

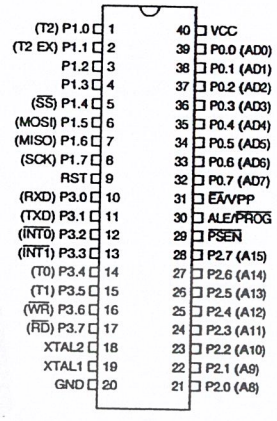
## AT89S8252

Rev. 0401E–02/00

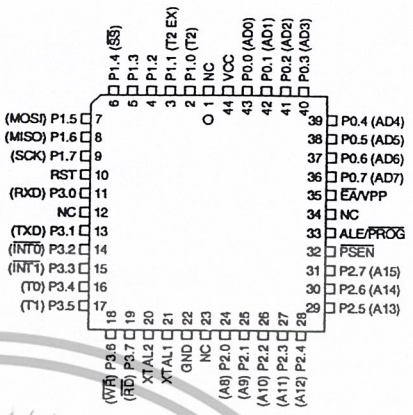
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## Pin Configurations

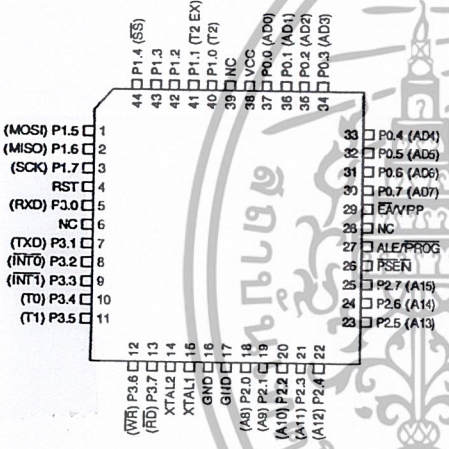
PDIP




PLCC




PQFP/TQFP




## Pin Description

### VCC

Supply voltage.

### GND

Ground.

### Port 0

Port 0 is an 8-bit open drain bbi-didirectional I/O port. As an output port, each pin can sink eight TTL inputs. When 1s are written to port 0 pins, the pins can be used as high-impedance inputs.

Port 0 can also be configured to be the multiplexed low-order address/data bus during accesses to external program and data memory. In this mode, P0 has internal pullups.

Port 0 also receives the code bytes during Flash programming and outputs the code bytes during program verification. External pullups are required during program verification.

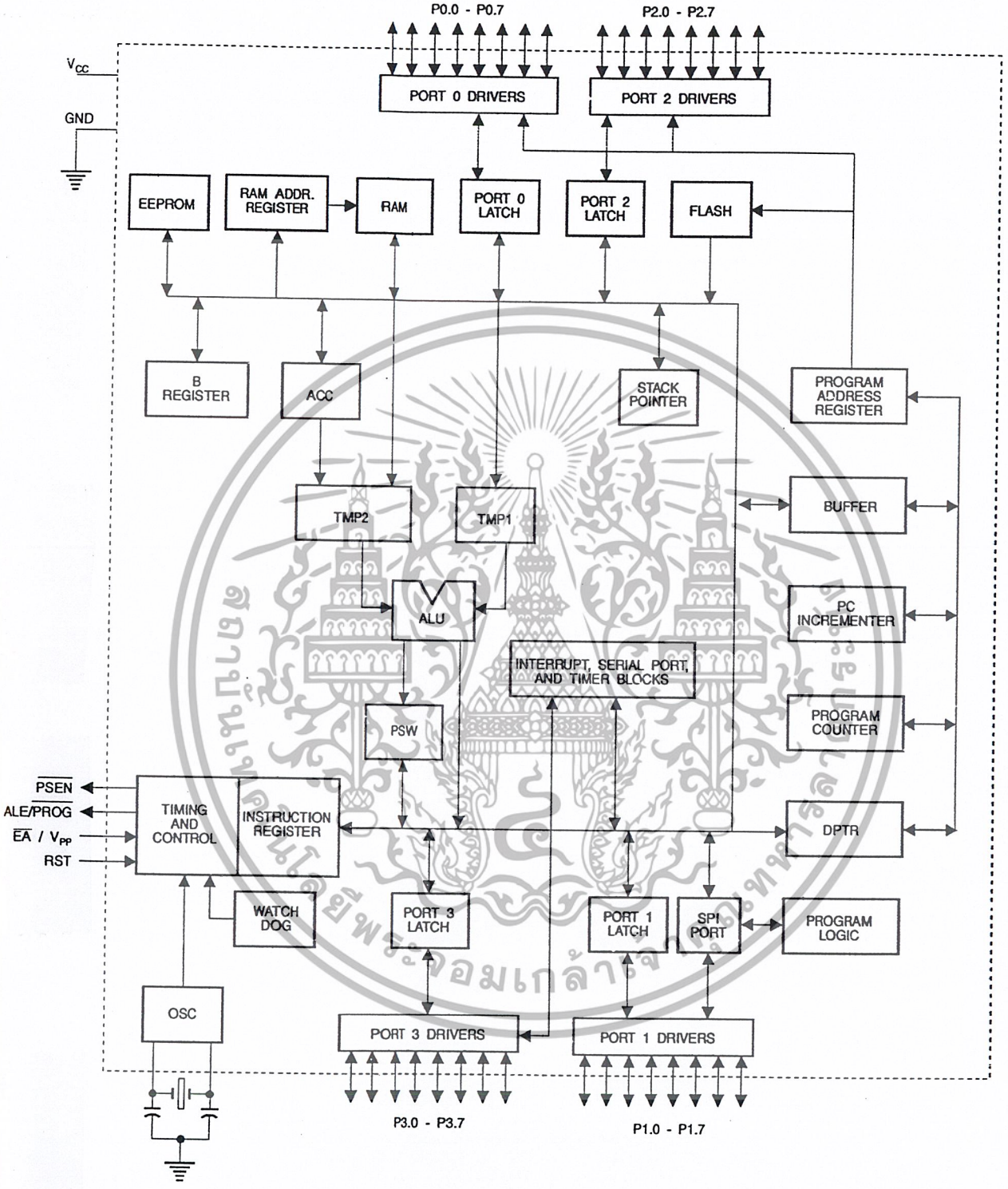
### Port 1

Port 1 is an 8-bit bi-directional I/O port with internal pullups. The Port 1 output buffers can sink/source four TTL inputs. When 1s are written to Port 1 pins, they are pulled high by the internal pullups and can be used as inputs. As inputs, Port 1 pins that are externally being pulled low will source current ($I_{IL}$) because of the internal pullups.



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## Block Diagram






เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Some Port 1 pins provide additional functions. P1.0 and P1.1 can be configured to be the timer/counter 2 external count input (P1.0/T2) and the timer/counter 2 trigger input (P1.1/T2EX), respectively.

## Pin Description

Furthermore, P1.4, P1.5, P1.6, and P1.7 can be configured as the SPI slave port select, data input/output and shift clock input/output pins as shown in the following table.

| Port Pin | Alternate Functions |
|---|---|
| P1.0 | T2 (external count input to Timer/Counter 2), clock-out |
| P1.1 | T2EX (Timer/Counter 2 capture/reload trigger and direction control) |
| P1.4 | $\overline{SS}$ (Slave port select input) |
| P1.5 | MOSI (Master data output, slave data input pin for SPI channel) |
| P1.6 | MISO (Master data input, slave data output pin for SPI channel) |
| P1.7 | SCK (Master clock output, slave clock input pin for SPI channel) |

Port 1 also receives the low-order address bytes during Flash programming and verification.

### Port 2

Port 2 is an 8-bit bi-directional I/O port with internal pullups. The Port 2 output buffers can sink/source four TTL inputs. When 1s are written to Port 2 pins, they are pulled high by the internal pullups and can be used as inputs. As inputs, Port 2 pins that are externally being pulled low will source current ($I_{IL}$) because of the internal pullups.

Port 2 emits the high-order address byte during fetches from external program memory and during accesses to external data memory that use 16-bit addresses (MOVX @ DPTR). In this application, Port 2 uses strong internal pullups when emitting 1s. During accesses to external data memory that use 8-bit addresses (MOVX @ RI), Port 2 emits the contents of the P2 Special Function Register.

Port 2 also receives the high-order address bits and some control signals during Flash programming and verification.

### Port 3

Port 3 is an 8 bit bi-directional I/O port with internal pullups. The Port 3 output buffers can sink/source four TTL inputs. When 1s are written to Port 3 pins, they are pulled high by the internal pullups and can be used as inputs. As inputs, Port 3 pins that are externally being pulled low will source current ($I_{IL}$) because of the pullups.

Port 3 also serves the functions of various special features of the AT89S8252, as shown in the following table.

Port 3 also receives some control signals for Flash programming and verification.

| Port Pin | Alternate Functions |
|---|---|
| P3.0 | RXD (serial input port) |
| P3.1 | TXD (serial output port) |
| P3.2 | $\overline{INT0}$ (external interrupt 0) |
| P3.3 | $\overline{INT1}$ (external interrupt 1) |
| P3.4 | T0 (timer 0 external input) |
| P3.5 | T1 (timer 1 external input) |
| P3.6 | $\overline{WR}$ (external data memory write strobe) |
| P3.7 | $\overline{RD}$ (external data memory read strobe) |

### RST

Reset input. A high on this pin for two machine cycles while the oscillator is running resets the device.

### ALE/$\overline{PROG}$

Address Latch Enable is an output pulse for latching the low byte of the address during accesses to external memory. This pin is also the program pulse input ($\overline{PROG}$) during Flash programming.

In normal operation, ALE is emitted at a constant rate of 1/6 the oscillator frequency and may be used for external timing or clocking purposes. Note, however, that one ALE pulse is skipped during each access to external data memory.

If desired, ALE operation can be disabled by setting bit 0 of SFR location 8EH. With the bit set, ALE is active only during a MOVX or MOVC instruction. Otherwise, the pin is weakly pulled high. Setting the ALE-disable bit has no effect if the microcontroller is in external execution mode.

### $\overline{PSEN}$

Program Store Enable is the read strobe to external program memory.

When the AT89S8252 is executing code from external program memory, $\overline{PSEN}$ is activated twice each machine cycle, except that two $\overline{PSEN}$ activations are skipped during each access to external data memory.

### $\overline{EA}$/VPP

External Access Enable. $\overline{EA}$ must be strapped to GND in order to enable the device to fetch code from external pro-



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

gram memory locations starting at 0000H up to FFFFH. Note, however, that if lock bit 1 is programmed, $\overline{EA}$ will be internally latched on reset.

$\overline{EA}$ should be strapped to $V_{CC}$ for internal program executions. This pin also receives the 12-volt programming enable voltage ($V_{PP}$) during Flash programming when 12-volt programming is selected.

**XTAL1**

Input to the inverting oscillator amplifier and input to the internal clock operating circuit.

**XTAL2**

Output from the inverting osciliator amplifier.

**Table 1.** AT89S8252 SFR Map and Reset Values

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 0F8H | | | | | | | | | 0FFH |
| 0F0H | B<br>00000000 | | | | | | | | 0F7H |
| 0E8H | | | | | | | | | 0EFH |
| 0E0H | ACC<br>00000000 | | | | | | | | 0E7H |
| 0D8H | | | | | | | | | 0DFH |
| 0D0H | PSW<br>00000000 | | | | | SPCR<br>000001XX | | | 0D7H |
| 0C8H | T2CON<br>00000000 | T2MOD<br>XXXXXX00 | RCAP2L<br>00000000 | RCAP2H<br>00000000 | TL2<br>00000000 | TH2<br>00000000 | | | 0CFH |
| 0C0H | | | | | | | | | 0C7H |
| 0B8H | IP<br>XX000000 | | | | | | | | 0BFH |
| 0B0H | P3<br>11111111 | | | | | | | | 0B7H |
| 0A8H | IE<br>0X000000 | | SPSR<br>00XXXXXX | | | | | | 0AFH |
| 0A0H | P2<br>11111111 | | | | | | | | 0A7H |
| 98H | SCON<br>00000000 | SBUF<br>XXXXXXXX | | | | | | | 9FH |
| 90H | P1<br>11111111 | | | | | | WMCON<br>00000010 | | 97H |
| 88H | TCON<br>00000000 | TMOD<br>00000000 | TL0<br>00000000 | TL1<br>00000000 | TH0<br>00000000 | TH1<br>00000000 | | | 8FH |
| 80H | P0<br>11111111 | SP<br>00000111 | DP0L<br>00000000 | DP0H<br>00000000 | DP1L<br>00000000 | DP1H<br>00000000 | SPDR<br>XXXXXXXX | PCON<br>0XXX0C00 | 87H |



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## Special Function Registers

A map of the on-chip memory area called the Special Function Register (SFR) space is shown in Table 1.

Note that not all of the addresses are occupied, and unoccupied addresses may not be implemented on the chip. Read accesses to these addresses will in general return random data, and write accesses will have an indeterminate effect.

User software should not write 1s to these unlisted locations, since they may be used in future products to invoke new features. In that case, the reset or inactive values of the new bits will always be 0.

**Timer 2 Registers** Control and status bits are contained in registers T2CON (shown in Table 2) and T2MOD (shown in Table 9) for Timer 2. The register pair (RCAP2H, RCAP2L) are the Capture/Reload registers for Timer 2 in 16 bit capture mode or 16-bit auto-reload mode.

**Table 2.** T2CON—Timer/Counter 2 Control Register

T2CON Address = 0C8H — Reset Value = 0000 0000B

Bit Addressable

| Bit | TF2 | EXF2 | RCLK | TCLK | EXEN2 | TR2 | C/$\overline{T2}$ | CP/$\overline{RL2}$ |
|---|---|---|---|---|---|---|---|---|
| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |

| Symbol | Function |
|---|---|
| TF2 | Timer 2 overflow flag set by a Timer 2 overflow and must be cleared by software. TF2 will not be set when either RCLK = 1 or TCLK = 1. |
| EXF2 | Timer 2 external flag set when either a capture or reload is caused by a negative transition on T2EX and EXEN2 = 1. When Timer 2 interrupt is enabled, EXF2 = 1 will cause the CPU to vector to the Timer 2 interrupt routine. EXF2 must be cleared by software. EXF2 does not cause an interrupt in up/down counter mode (DCEN = 1). |
| RCLK | Receive clock enable. When set, causes the serial port to use Timer 2 overflow pulses for its receive clock in serial port Modes 1 and 3. RCLK = 0 causes Timer 1 overflows to be used for the receive clock. |
| TCLK | Transmit clock enable. When set, causes the serial port to use Timer 2 overflow pulses for its transmit clock in serial port Modes 1 and 3. TCLK = 0 causes Timer 1 overflows to be used for the transmit clock. |
| EXEN2 | Timer 2 external enable. When set, allows a capture or reload to occur as a result of a negative transition on T2EX if Timer 2 is not being used to clock the serial port. EXEN2 = 0 causes Timer 2 to ignore events at T2EX. |
| TR2 | Start/Stop control for Timer 2. TR2 = 1 starts the timer. |
| C/$\overline{T2}$ | Timer or counter select for Timer 2. C/$\overline{T2}$ = 0 for timer function. C/$\overline{T2}$ = 1 for external event counter (falling edge triggered). |
| CP/$\overline{RL2}$ | Capture/Reload select. CP/$\overline{RL2}$ = 1 causes captures to occur on negative transitions at T2EX if EXEN2 = 1. CP/$\overline{RL2}$ = 0 causes automatic reloads to occur when Timer 2 overflows or negative transitions occur at T2EX when EXEN2 = 1. When either RCLK or TCLK = 1, this bit is ignored and the timer is forced to auto-reload on Timer 2 overflow. |



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**Watchdog and Memory Control Register** The WMCON register contains control bits for the Watchdog Timer (shown in Table 3). The EEMEN and EEMWE bits are used to select the 2K bytes on-chip EEPROM, and to enable byte-write. The DPS bit selects one of two DPTR registers available.

**Table 3.** WMCON—Watchdog and Memory Control Register

WMCON Address = 96H — Reset Value = 0000 0010B

| Bit | PS2 | PS1 | PS0 | EEMWE | EEMEN | DPS | WDTRST | WDTEN |
|---|---|---|---|---|---|---|---|---|
| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |

| Symbol | Function |
|---|---|
| PS2<br>PS1<br>PS0 | Prescaler Bits for the Watchdog Timer. When all three bits are set to "0", the watchdog timer has a nominal period of 16 ms. When all three bits are set to "1", the nominal period is 2048 ms. |
| EEMWE | EEPROM Data Memory Write Enable Bit. Set this bit to "1" before initiating byte write to on-chip EEPROM with the MOVX instruction. User software should set this bit to "0" after EEPROM write is completed. |
| EEMEN | Internal EEPROM Access Enable. When EEMEN = 1, the MOVX instruction with DPTR will access on-chip EEPROM instead of external data memory. When EEMEN = 0, MOVX with DPTR accesses external data memory. |
| DPS | Data Pointer Register Select. DPS = 0 selects the first bank of Data Pointer Register, DP0, and DPS = 1 selects the second bank, DP1 |
| WDTRST<br>RDY/$\overline{BSY}$ | Watchdog Timer Reset and EEPROM Ready/$\overline{Busy}$ Flag. Each time this bit is set to "1" by user software, a pulse is generated to reset the watchdog timer. The WDTRST bit is then automatically reset to "0" in the next instruction cycle. The WDTRST bit is Write-Only. This bit also serves as the RDY/$\overline{BSY}$ flag in a Read-Only mode during EEPROM write. RDY/$\overline{BSY}$ = 1 means that the EEPROM is ready to be programmed. While programming operations are being executed, the RDY/$\overline{BSY}$ bit equals "0" and is automatically reset to "1" when programming is completed. |
| WDTEN | Watchdog Timer Enable Bit. WDTEN = 1 enables the watchdog timer and WDTEN = 0 disables the watchdog timer. |

**SPI Registers** Control and status bits for the Serial Peripheral Interface are contained in registers SPCR (shown in Table 4) and SPSR (shown in Table 5). The SPI data bits are contained in the SPDR register. Writing the SPI data register during serial data transfer sets the Write Collision bit, WCOL, in the SPSR register. The SPDR is double buffered for writing and the values in SPDR are not changed by Reset.

**Interrupt Registers** The global interrupt enable bit and the individual interrupt enable bits are in the IE register. In addition, the individual interrupt enable bit for the SPI is in the SPCR register. Two priorities can be set for each of the six interrupt sources in the IP register.

**Dual Data Pointer Registers** To facilitate accessing both internal EEPROM and external data memory, two banks of 16 bit Data Pointer Registers are provided: DP0 at SFR address locations 82H-83H and DP1 at 84H-85H. Bit DPS = 0 in SFR WMCON selects DP0 and DPS = 1 selects DP1. The user should always initialize the DPS bit to the appropriate value before accessing the respective Data Pointer Register.

**Power Off Flag** The Power Off Flag (POF) is located at bit_4 (PCON.4) in the PCON SFR. POF is set to "1" during power up. It can be set and reset under software control and is not affected by RESET.



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**Table 4.** SPCR—SPI Control Register

SPCR Address = D5H

Reset Value = 0000 01XXB

| Bit | SPIE | SPE | DORD | MSTR | CPOL | CPHA | SPR1 | SPR0 |
|---|---|---|---|---|---|---|---|---|
| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |

| Symbol | Function |
|---|---|
| SPIE | SPI Interrupt Enable. This bit, in conjunction with the ES bit in the IE register, enables SPI interrupts: SPIE = 1 and ES = 1 enable SPI interrupts. SPIE = 0 disables SPI interrupts. |
| SPE | SPI Enable. SPI = 1 enables the SPI channel and connects $\overline{SS}$, MOSI, MISO and SCK to pins P1.4, P1.5, P1.6, and P1.7. SPI = 0 disables the SPI channel. |
| DORD | Data Order. DORD = 1 selects LSB first data transmission. DORD = 0 selects MSB first data transmission. |
| MSTR | Master/Slave Select. MSTR = 1 selects Master SPI mode. MSTR = 0 selects Slave SPI mode. |
| CPOL | Clock Polarity. When CPOL = 1, SCK is high when idle. When CPOL = 0, SCK of the master device is low when not transmitting. Please refer to figure on SPI Clock Phase and Polarity Control. |
| CPHA | Clock Phase. The CPHA bit together with the CPOL bit controls the clock and data relationship between master and slave. Please refer to figure on SPI Clock Phase and Polarity Control. |
| SPR0<br>SPR1 | SPI Clock Rate Select. These two bits control the SCK rate of the device configured as master. SPR1 and SPR0 have no effect on the slave. The relationship between SCK and the oscillator frequency, $F_{osc}$, is as follows:<br>SPR1 SPR0 SCK = $F_{osc}$ divided by<br>0 0 4<br>0 1 16<br>1 0 64<br>1 1 128 |

**Table 5.** SPSR – SPI Status Register

SPSR Address = AAH

Reset Value = 00XX XXXXB

| Bit | SPIF | WCOL | – | – | – | – | – | – |
|---|---|---|---|---|---|---|---|---|
| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |

| Symbol | Function |
|---|---|
| SPIF | SPI Interrupt Flag. When a serial transfer is complete, the SPIF bit is set and an interrupt is generated if SPIE = 1 and ES = 1. The SPIF bit is cleared by reading the SPI status register with SPIF and WCOL bits set, and then accessing the SPI data register. |
| WCOL | Write Collision Flag. The WCOL bit is set if the SPI data register is written during a data transfer. During data transfer, the result of reading the SPDR register may be incorrect, and writing to it has no effect. The WCOL bit (and the SPIF bit) are cleared by reading the SPI status register with SPIF and WCOL set, and then accessing the SPI data register. |

**Table 6.** SPDR – SPI Data Register

SPDR Address = 86H

Reset Value = unchanged

| Bit | SPD7 | SPD6 | SPD5 | SPD4 | SPD3 | SPD2 | SPD1 | SPD0 |
|---|---|---|---|---|---|---|---|---|
| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## Data Memory – EEPROM and RAM

The AT89S8252 implements 2K bytes of on-chip EEPROM for data storage and 256 bytes of RAM. The upper 128 bytes of RAM occupy a parallel space to the Special Function Registers. That means the upper 128 bytes have the same addresses as the SFR space but are physically separate from SFR space.

When an instruction accesses an internal location above address 7FH, the address mode used in the instruction specifies whether the CPU accesses the upper 128 bytes of RAM or the SFR space. Instructions that use direct addressing access SFR space.

For example, the following direct addressing instruction accesses the SFR at location 0A0H (which is P2).

```
MOV 0A0H, #data
```

Instructions that use indirect addressing access the upper 128 bytes of RAM. For example, the following indirect addressing instruction, where R0 contains 0A0H, accesses the data byte at address 0A0H, rather than P2 (whose address is 0A0H).

```
MOV @R0, #data
```

Note that stack operations are examples of indirect addressing, so the upper 128 bytes of data RAM are available as stack space.

The on-chip EEPROM data memory is selected by setting the EEMEN bit in the WMCON register at SFR address location 96H. The EEPROM address range is from 000H to 7FFH. The MOVX instructions are used to access the EEPROM. To access off-chip data memory with the MOVX instructions, the EEMEN bit needs to be set to "0".

The EEMWE bit in the WMCON register needs to be set to "1" before any byte location in the EEPROM can be written. User software should reset EEMWE bit to "0" if no further EEPROM write is required. EEPROM write cycles in the serial programming mode are self-timed and typically take 2.5 ms. The progress of EEPROM write can be monitored by reading the RDY/$\overline{BSY}$ bit (read-only) in SFR WMCON. RDY/$\overline{BSY}$ = 0 means programming is still in progress and RDY/$\overline{BSY}$ = 1 means EEPROM write cycle is completed and another write cycle can be initiated.

In addition, during EEPROM programming, an attempted read from the EEPROM will fetch the byte being written with the MSB complemented. Once the write cycle is completed, true data are valid at all bit locations.

## Programmable Watchdog Timer

The programmable Watchdog Timer (WDT) operates from an independent oscillator. The prescaler bits, PS0, PS1 and PS2 in SFR WMCON are used to set the period of the Watchdog Timer from 16 ms to 2048 ms. The available timer periods are shown in the following table and the actual timer periods (at $V_{CC}$ = 5V) are within ±30% of the nominal.

The WDT is disabled by Power-on Reset and during Power-down. It is enabled by setting the WDTEN bit in SFR WMCON (address = 96H). The WDT is reset by setting the WDTRST bit in WMCON. When the WDT times out without being reset or disabled, an internal RST pulse is generated to reset the CPU.

**Table 7.** Watchdog Timer Period Selection

| WDT Prescaler Bits | | | |
|---|---|---|---|
| PS2 | PS1 | PS0 | Period (nominal) |
| 0 | 0 | 0 | 16 ms |
| 0 | 0 | 1 | 32 ms |
| 0 | 1 | 0 | 64 ms |
| 0 | 1 | 1 | 128 ms |
| 1 | 0 | 0 | 256 ms |
| 1 | 0 | 1 | 512 ms |
| 1 | 1 | 0 | 1024 ms |
| 1 | 1 | 1 | 2048 ms |

## Timer 0 and 1

Timer 0 and Timer 1 in the AT89S8252 operate the same way as Timer 0 and Timer 1 in the AT89C51, AT89C52 and AT89C55. For further information, see the October 1995 Microcontroller Data Book, page 2-45, section titled, "Timer/Counters."

## Timer 2

Timer 2 is a 16 bit Timer/Counter that can operate as either a timer or an event counter. The type of operation is selected by bit C/$\overline{T2}$ in the SFR T2CON (shown in Table 2). Timer 2 has three operating modes: capture, auto-reload (up or down counting), and baud rate generator. The modes are selected by bits in T2CON, as shown in Table 8.

Timer 2 consists of two 8-bit registers, TH2 and TL2. In the Timer function, the TL2 register is incremented every machine cycle. Since a machine cycle consists of 12 oscillator periods, the count rate is 1/12 of the oscillator frequency.

In the Counter function, the register is incremented in response to a 1-to-0 transition at its corresponding external input pin, T2. In this function, 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



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

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. To ensure that a given level is sampled at least once before it changes, the level should be held for at least one full machine cycle.

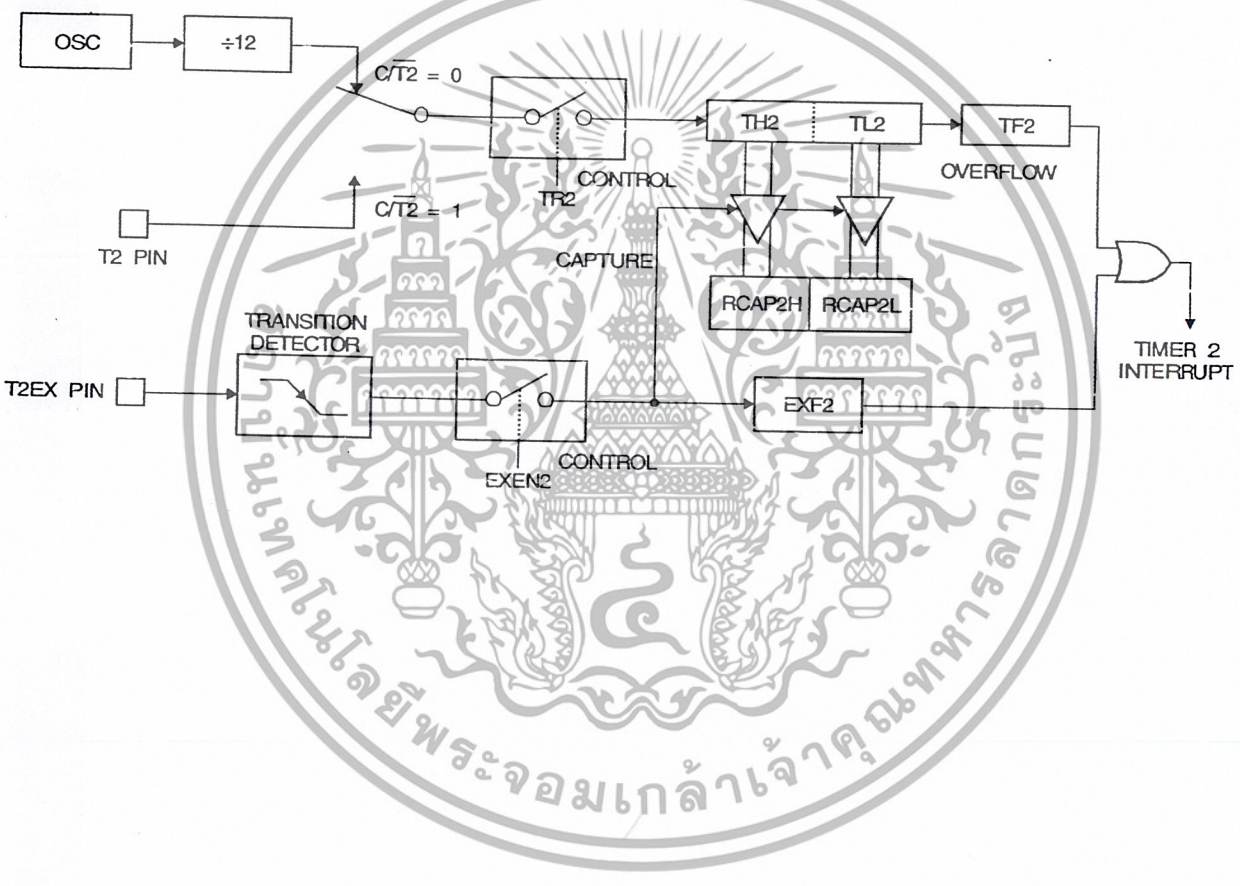
**Table 8.** Timer 2 Operating Modes

| RCLK + TCLK | CP/$\overline{RL2}$ | TR2 | MODE |
|---|---|---|---|
| C | 0 | 1 | 16-bit Auto-reload |
| 0 | 1 | 1 | 16-bit Capture |
| 1 | X | 1 | Baud Rate Generator |
| X | X | 0 | (Off) |

## Capture Mode

In the capture mode, two options are selected by bit EXEN2 in T2CON. If EXEN2 = 0, Timer 2 is a 16 bit timer or counter which upon overflow sets bit TF2 in T2CON. This bit can then be used to generate an interrupt. If EXEN2 = 1, Timer 2 performs the same operation, but a 1-to-0 transition at external input T2EX also causes the current value in TH2 and TL2 to be captured into RCAP2H and RCAP2L, respectively. In addition, the transition at T2EX causes bit EXF2 in T2CON to be set. The EXF2 bit, like TF2, can generate an interrupt. The capture mode is illustrated in Figure 1.

**Figure 1.** Timer 2 in Capture Mode






เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## Auto-reload (Up or Down Counter)

Timer 2 can be programmed to count up or down when configured in its 16 bit auto-reload mode. This feature is invoked by the DCEN (Down Counter Enable) bit located in the SFR T2MOD (see Table 9). Upon reset, the DCEN bit is set to 0 so that timer 2 will default to count up. When DCEN is set, Timer 2 can count up or down, depending on the value of the T2EX pin.

Figure 2 shows Timer 2 automatically counting up when DCEN = 0. In this mode, two options are selected by bit EXEN2 in T2CON. If EXEN2 = 0, Timer 2 counts up to 0FFFFH and then sets the TF2 bit upon overflow. The overflow also causes the timer registers to be reloaded with the 16 bit value in RCAP2H and RCAP2L. The values in RCAP2H and RCAP2L are preset by software. If EXEN2 = 1, a 16 bit reload can be triggered either by an overflow or by a 1-to-0 transition at external input T2EX. This transition also sets the EXF2 bit. Both the TF2 and EXF2 bits can generate an interrupt if enabled.

Setting the DCEN bit enables Timer 2 to count up or down, as shown in Figure 3. In this mode, the T2EX pin controls the direction of the count. A logic 1 at T2EX makes Timer 2 count up. The timer will overflow at 0FFFFH and set the TF2 bit. This overflow also causes the 16 bit value in RCAP2H and RCAP2L to be reloaded into the timer registers, TH2 and TL2, respectively.

A logic 0 at T2EX makes Timer 2 count down. The timer underflows when TH2 and TL2 equal the values stored in RCAP2H and RCAP2L. The underflow sets the TF2 bit and causes 0FFFFH to be reloaded into the timer registers.

The EXF2 bit toggles whenever Timer 2 overflows or underflows and can be used as a 17th bit of resolution. In this operating mode, EXF2 does not flag an interrupt.

**Figure 2.** Timer 2 in Auto Reload Mode (DCEN = 0)

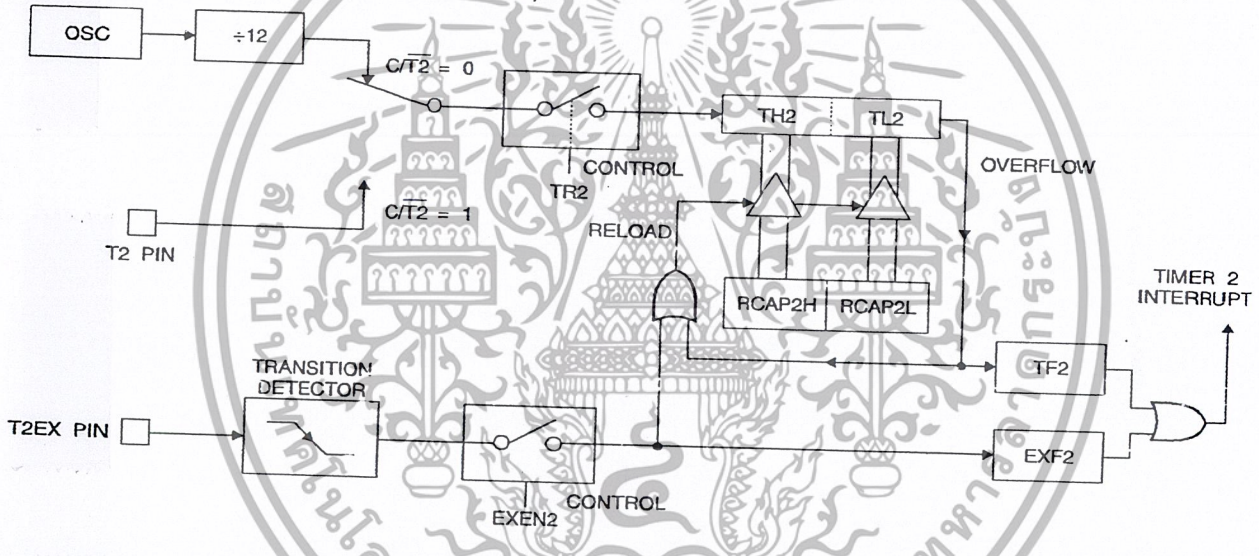



**Table 9.** T2MOD – Timer 2 Mode Control Register

T2MOD Address = 0C9H — Reset Value = XXXX XX00B

Not Bit Addressable

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | – | – | – | – | – | – | T2OE | DCEN |

| Symbol | Function |
|---|---|
| – | Not implemented, reserved for future use. |
| T2OE | Timer 2 Output Enable bit. |
| DCEN | When set, this bit allows Timer 2 to be configured as an up/down counter. |



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**Figure 3.** Timer 2 Auto Reload Mode (DCEN = 1)

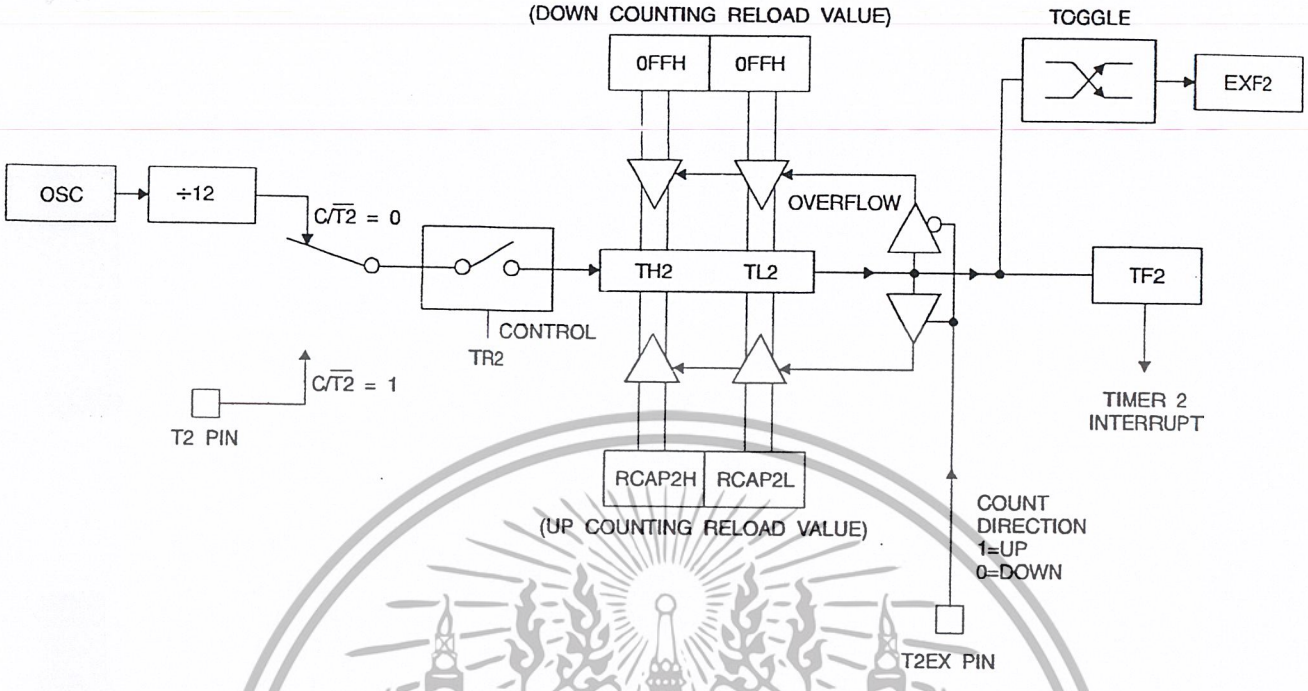
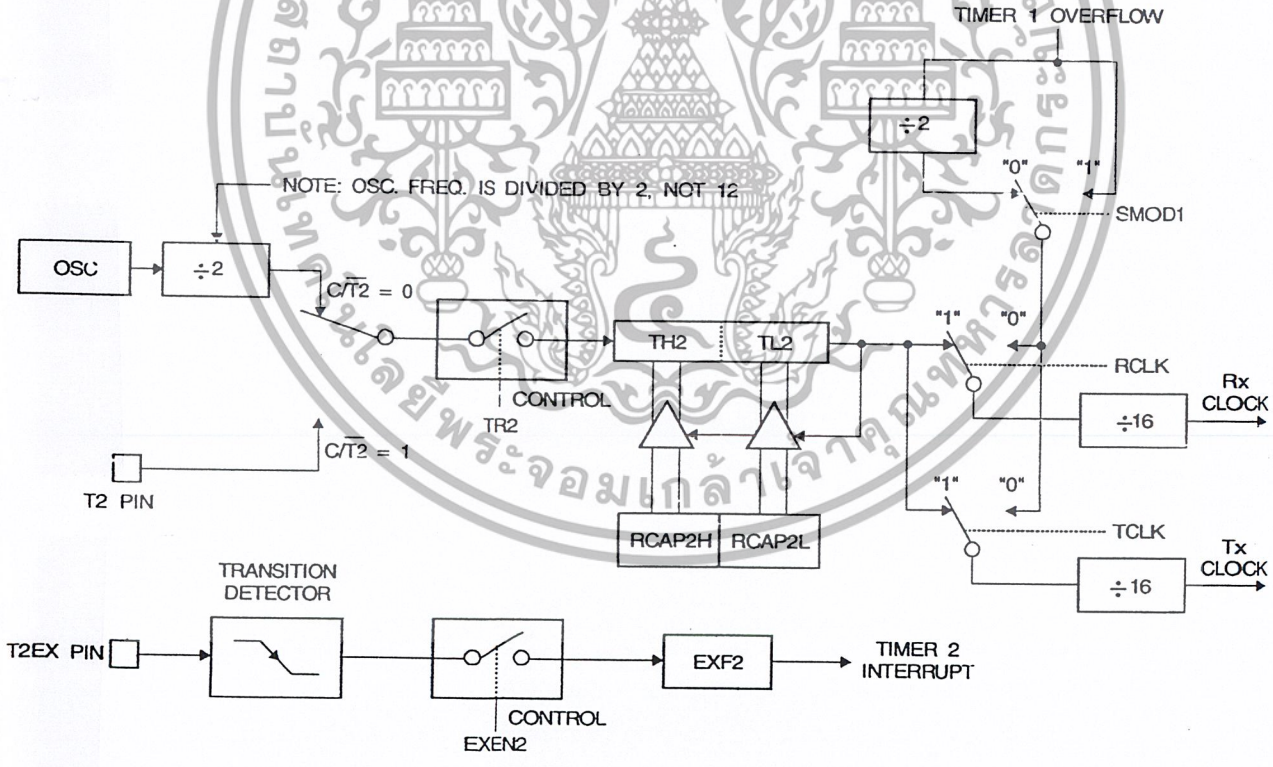



**Figure 4.** Timer 2 in Baud Rate Generator Mode






เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## Baud Rate Generator

Timer 2 is selected as the baud rate generator by setting TCLK and/or RCLK in T2CON (Table 2). Note that the baud rates for transmit and receive can be different if Timer 2 is used for the receiver or transmitter and Timer 1 is used for the other function. Setting RCLK and/or TCLK puts Timer 2 into its baud rate generator mode, as shown in Figure 4.

The baud rate generator mode is similar to the auto-reload mode, in that a rollover in TH2 causes the Timer 2 registers to be reloaded with the 16 bit value in registers RCAP2H and RCAP2L, which are preset by software.

The baud rates in Modes 1 and 3 are determined by Timer 2's overflow rate according to the following equation.

$$\text{Modes 1 and 3 Baud Rates} = \frac{\text{Timer 2 Overflow Rate}}{16}$$

The Timer can be configured for either timer or counter operation. In most applications, it is configured for timer operation ($C/\overline{T2} = 0$). The timer operation is different for Timer 2 when it is used as a baud rate generator. Normally, as a timer, it increments every machine cycle (at 1/12 the oscillator frequency). As a baud rate generator, however, it increments every state time (at 1/2 the oscillator frequency). The baud rate formula is given below.

$$\frac{\text{Modes 1 and 3}}{\text{Baud Rate}} = \frac{\text{Oscillator Frequency}}{32 \times [65536 - (\text{RCAP2H,RCAP2L})]}$$

where (RCAP2H, RCAP2L) is the content of RCAP2H and RCAP2L taken as a 16 bit unsigned integer.

Timer 2 as a baud rate generator is shown in Figure 4. This figure is valid only if RCLK or TCLK = 1 in T2CON. Note that a rollover in TH2 does not set TF2 and will not generate an interrupt. Note too, that if EXEN2 is set, a 1-to-0 transition in T2EX will set EXF2 but will not cause a reload from (RCAP2H, RCAP2L) to (TH2, TL2). Thus when Timer 2 is in use as a baud rate generator, T2EX can be used as an extra external interrupt.

Note that when Timer 2 is running (TR2 = 1) as a timer in the baud rate generator mode, TH2 or TL2 should not be read from or written to. Under these conditions, the Timer is incremented every state time, and the results of a read or write may not be accurate. The RCAP2 registers may be read but should not be written to, because a write might overlap a reload and cause write and/or reload errors. The timer should be turned off (clear TR2) before accessing the Timer 2 or RCAP2 registers.

## Programmable Clock Out

A 50% duty cycle clock can be programmed to come out on P1.0, as shown in Figure 5. This pin, besides being a regular I/O pin, has two alternate functions. It can be programmed to input the external clock for Timer/Counter 2 or to output a 50% duty cycle clock ranging from 61 Hz to 4 MHz at a 16 MHz operating frequency.

To configure the Timer/Counter 2 as a clock generator, bit $C/\overline{T2}$ (T2CON.1) must be cleared and bit T2OE (T2MOD.1) must be set. Bit TR2 (T2CON.2) starts and stops the timer.

The clock-out frequency depends on the oscillator frequency and the reload value of Timer 2 capture registers (RCAP2H, RCAP2L), as shown in the following equation.

$$\text{Clock Out Frequency} = \frac{\text{Oscillator Frequency}}{4 \times [65536 - (\text{RCAP2H,RCAP2L})]}$$

In the clock-out mode, Timer 2 rollovers will not generate an interrupt. This behavior is similar to when Timer 2 is used as a baud-rate generator. It is possible to use Timer 2 as a baud-rate generator and a clock generator simultaneously. Note, however, that the baud-rate and clock-out frequencies cannot be determined independently from one another since they both use RCAP2H and RCAP2L.



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**Figure 5.** Timer 2 in Clock-out Mode

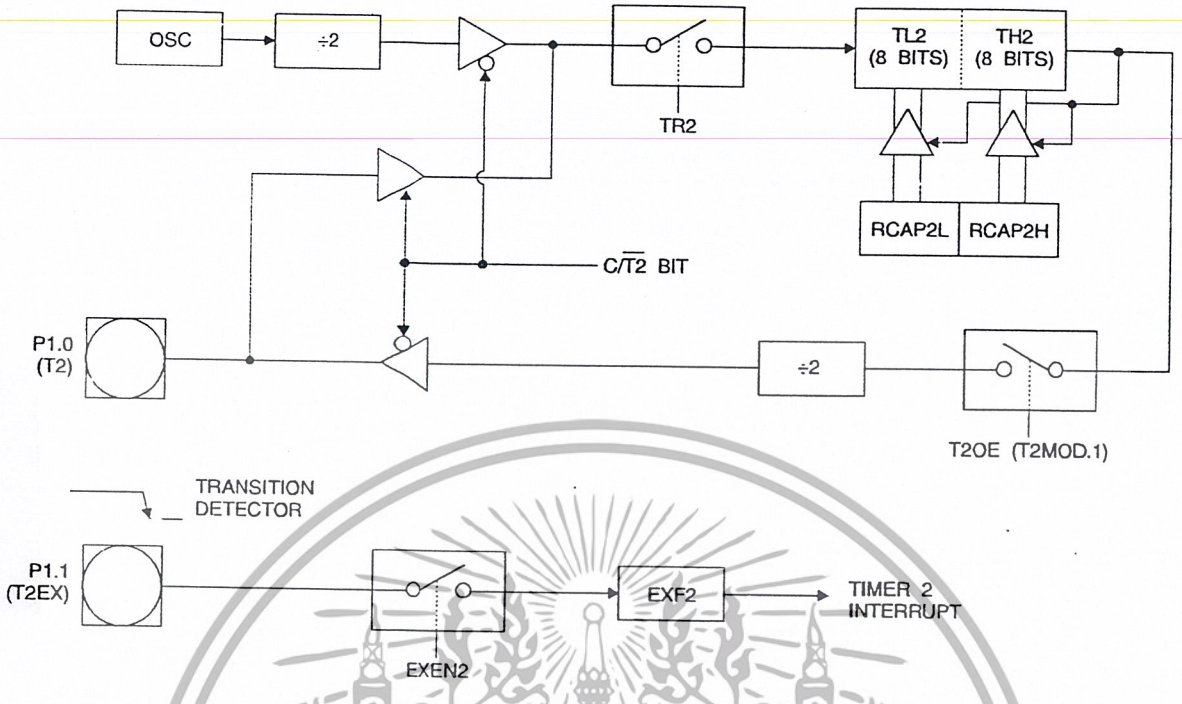
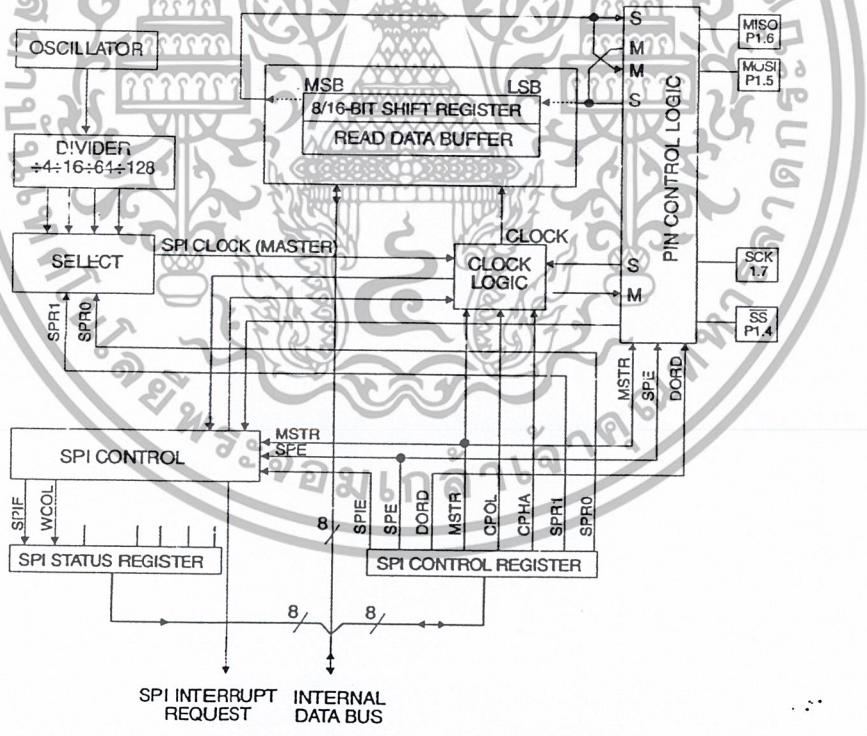



**Figure 6.** SPI Block Diagram






เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## UART

The UART in the AT89S8252 operates the same way as the UART in the AT89C51, AT89C52 and AT89C55. For further information, see the October 1995 Microcontroller Data Book, page 2-49, section titled, "Serial Interface."

## Serial Peripheral Interface

The serial peripheral interface (SPI) allows high-speed synchronous data transfer between the AT89S8252 and peripheral devices or between several AT89S8252 devices. The AT89S8252 SPI features include the following:

- Full-Duplex, 3-Wire Synchronous Data Transfer
- Master or Slave Operation
- 1.5 MHz Bit Frequency (max.)
- LSB First or MSB First Data Transfer
- Four Programmable Bit Rates
- End of Transmission Interrupt Flag
- Write Collision Flag Protection
- Wakeup from Idle Mode (Slave Mode Only)

The interconnection between master and slave CPUs with SPI is shown in the following figure. The SCK pin is the clock output in the master mode but is the clock input in the slave mode. Writing to the SPI data register of the master CPU starts the SPI clock generator, and the data written shifts out of the MOSI pin and into the MOSI pin of the slave CPU. After shifting one byte, the SPI clock generator stops, setting the end of transmission flag (SPIF). If both the SPI interrupt enable bit (SPIE) and the serial port interrupt enable bit (ES) are set, an interrupt is requested.

The Slave Select input, $\overline{SS}$/P1.4, is set low to select an individual SPI device as a slave. When $\overline{SS}$/P1.4 is set high, the SPI port is deactivated and the MOSI/P1.5 pin can be used as an input.

There are four combinations of SCK phase and polarity with respect to serial data, which are determined by control bits CPHA and CPOL. The SPI data transfer formats are shown in Figure 8 and Figure 9.

**Figure 7.** SPI Master-slave Interconnection

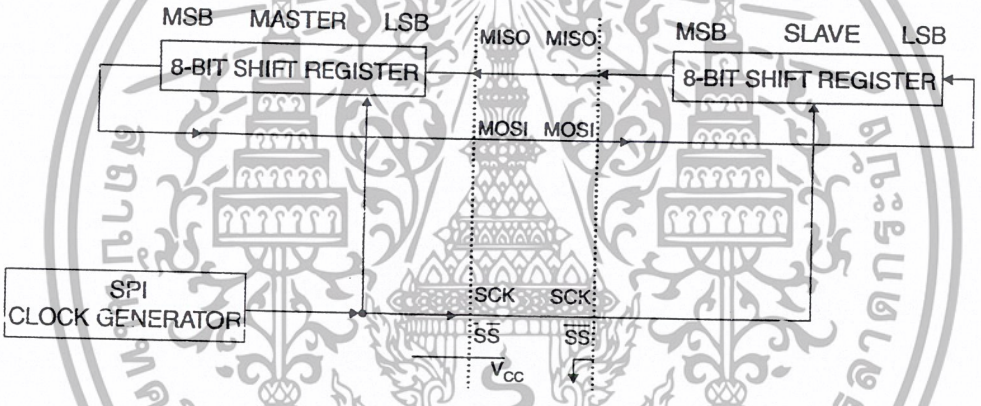
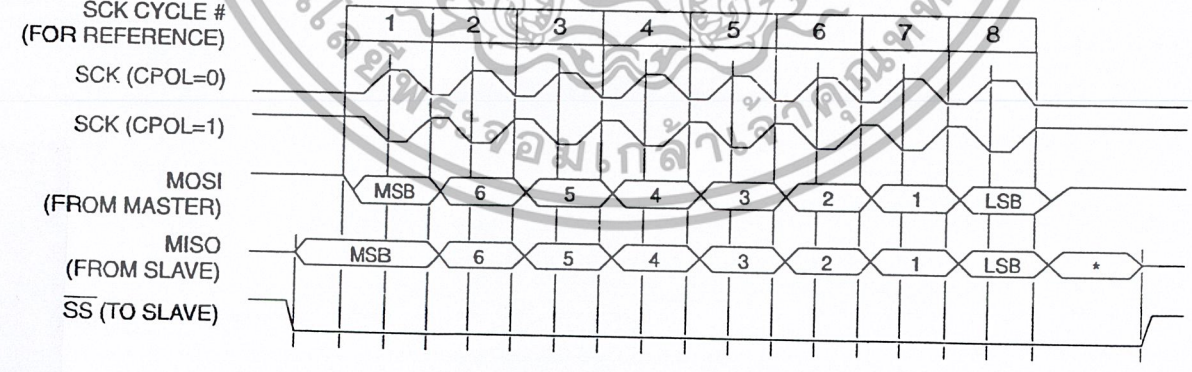



**Figure 8.** SPI transfer Format with CPHA = 0

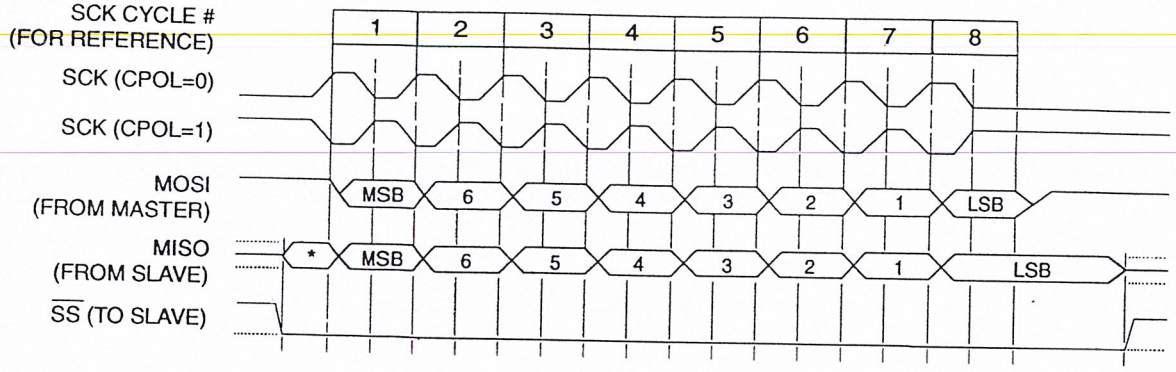



*Not defined but normally MSB of character just received



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**Figure 9.** SPI Transfer Format with CPHA = 1




*Not defined but normally LSB of previously transmitted character

## Interrupts

The AT89S8252 has a total of six interrupt vectors: two external interrupts ($\overline{INT0}$ and $\overline{INT1}$), three timer interrupts (Timers 0, 1, and 2), and the serial port interrupt. These interrupts are all shown in Figure 10.

Each of these interrupt sources can be individually enabled or disabled by setting or clearing a bit in Special Function Register IE. IE also contains a global disable bit, EA, which disables all interrupts at once.

Note that Table 10 shows that bit position IE.6 is unimplemented. In the AT89C51, bit position IE.5 is also unimplemented. User software should not write 1s to these bit positions, since they may be used in future AT89 products.

Timer 2 interrupt is generated by the logical OR of bits TF2 and EXF2 in register T2CON. Neither of these flags is cleared by hardware when the service routine is vectored to. In fact, the service routine may have to determine whether it was TF2 or EXF2 that generated the interrupt, and that bit will have to be cleared in software.

The Timer 0 and Timer 1 flags, TF0 and TF1, are set at S5P2 of the cycle in which the timers overflow. The values are then polled by the circuitry in the next cycle. However, the Timer 2 flag, TF2, is set at S2P2 and is polled in the same cycle in which the timer overflows.

**Table 10.** Interrupt Enable (IE) Register

(MSB)(LSB)

| EA | – | ET2 | ES | ET1 | EX1 | ET0 | EX0 |
|---|---|---|---|---|---|---|---|

Enable Bit = 1 enables the interrupt.

Enable Bit = 0 disables the interrupt.

| Symbol | Position | Function |
|---|---|---|
| EA | IE.7 | Disables all interrupts. If EA = 0, no interrupt is acknowledged. If EA = 1, each interrupt source is individually enabled or disabled by setting or clearing its enable bit. |
| – | IE.6 | Reserved. |
| ET2 | IE.5 | Timer 2 interrupt enable bit. |
| ES | IE.4 | SPI and UART interrupt enable bit. |
| ET1 | IE.3 | Timer 1 interrupt enable bit. |
| EX1 | IE.2 | External interrupt 1 enable bit. |
| ET0 | IE.1 | Timer 0 interrupt enable bit. |
| EX0 | IE.0 | External interrupt 0 enable bit. |

User software should never write 1s to unimplemented bits, because they may be used in future AT89 products.



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**Figure 10.** Interrupt Sources

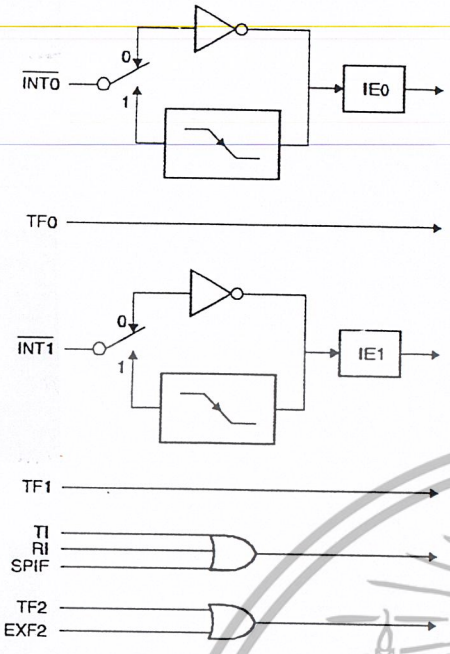
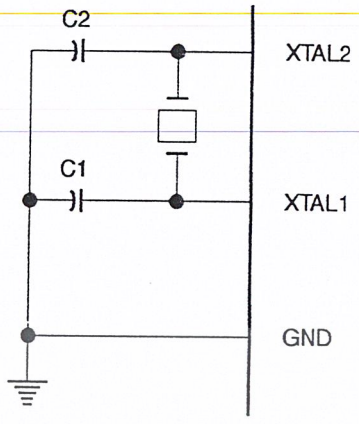
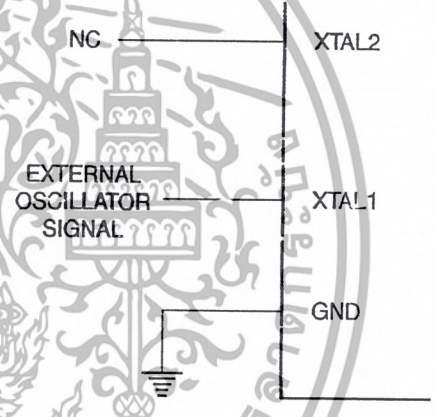



**Figure 11.** Oscillator Connections




Note: Note: C1, C2 = 30 pF ± 10 pF for Crystals
= 40 pF ± 10 pF for Ceramic Resonators

**Figure 12.** External Clock Drive Configuration




## Oscillator Characteristics

XTAL1 and XTAL2 are the input and output, respectively, of an inverting amplifier that can be configured for use as an on-chip oscillator, as shown in Figure 11. Either a quartz crystal or ceramic resonator may be used. To drive the device from an external clock source, XTAL2 should be left unconnected while XTAL1 is driven, as shown in Figure 12. There are no requirements on the duty cycle of the external clock signal, since the input to the internal clocking circuitry is through a divide-by-two flip-flop, but minimum and maximum voltage high and low time specifications must be observed.



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## Idle Mode

In idle mode, the CPU puts itself to sleep while all the on-chip peripherals remain active. The mode is invoked by software. The content of the on-chip RAM and all the special functions registers remain unchanged during this mode. The idle mode can be terminated by any enabled interrupt or by a hardware reset.

Note that when idle mode is terminated by a hardware reset, the device normally resumes program execution from where it left off, up to two machine cycles before the internal reset algorithm takes control. On-chip hardware inhibits access to internal RAM in this event, but access to the port pins is not inhibited. To eliminate the possibility of an unexpected write to a port pin when idle mode is terminated by a reset, the instruction following the one that invokes idle mode should not write to a port pin or to external memory.

## Status of External Pins During Idle and Power-down Modes

| Mode | Program Memory | ALE | $\overline{PSEN}$ | PORT0 | PORT1 | PORT2 | PORT3 |
|---|---|---|---|---|---|---|---|
| Idle | Internal | 1 | 1 | Data | Data | Data | Data |
| Idle | External | 1 | 1 | Float | Data | Address | Data |
| Power-down | Internal | 0 | 0 | Data | Data | Data | Data |
| Power-down | External | 0 | 0 | Float | Data | Data | Data |

## Power-down Mode

In the power-down mode, the oscillator is stopped and the instruction that invokes power-down is the last instruction executed. The on-chip RAM and Special Function Registers retain their values until the power-down mode is terminated. Exit from power-down can be initiated either by a hardware reset or by an enabled external interrupt. Reset redefines the SFRs but does not change the on-chip RAM. The reset should not be activated before $V_{CC}$ is restored to its normal operating level and must be held active long enough to allow the oscillator to restart and stabilize.

To exit power-down via an interrupt, the external interrupt must be enabled as level sensitive before entering power-down. The interrupt service routine starts at 16 ms (nominal) after the enabled interrupt pin is activated.

## Program Memory Lock Bits

The AT89S8252 has three lock bits that can be left unprogrammed (U) or can be programmed (P) to obtain the additional features listed in the following table.

When lock bit 1 is programmed, the logic level at the $\overline{EA}$ pin is sampled and latched during reset. If the device is powered up without a reset, the latch initializes to a random value and holds that value until reset is activated. The latched value of $\overline{EA}$ must agree with the current logic level at that pin in order for the device to function properly.

Once programmed, the lock bits can only be unprogrammed with the Chip Erase operations in either the parallel or serial modes.

## Lock Bit Protection Modes(1)(2)

| Program Lock Bits | | | | |
|---|---|---|---|---|
| | LB1 | LB2 | LB3 | Protection Type |
| 1 | U | U | U | No internal memory lock feature. |
| 2 | P | U | U | MOVC instructions executed from external program memory are disabled from fetching code bytes from internal memory. $\overline{EA}$ is sampled and latched on reset and further programming of the Flash memory (parallel or serial mode) is disabled. |
| 3 | P | P | U | Same as Mode 2, but parallel or serial verify are also disabled. |
| 4 | P | P | P | Same as Mode 3, but external execution is also disabled. |

Notes: 1. U = Unprogrammed
2. P = Programmed



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## Programming the Flash and EEPROM

Atmel's AT89S8252 Flash Microcontroller offers 8K bytes of in-system reprogrammable Flash Code memory and 2K bytes of EEPROM Data memory.

The AT89S8252 is normally shipped with the on-chip Flash Code and EEPROM Data memory arrays in the erased state (i.e. contents = FFH) and ready to be programmed. This device supports a High-voltage (12V) Parallel programming mode and a Low-voltage (5V) Serial programming mode. The serial programming mode provides a convenient way to download the AT89S8252 inside the user's system. The parallel programming mode is compatible with conventional third party Flash or EPROM programmers.

The Code and Data memory arrays are mapped via separate address spaces in the serial programming mode. In the parallel programming mode, the two arrays occupy one contiguous address space: 0000H to 1FFFH for the Code array and 2000H to 27FFH for the Data array.

The Code and Data memory arrays on the AT89S8252 are programmed byte-by-byte in either programming mode. An auto-erase cycle is provided with the self-timed programming operation in the serial programming mode. There is no need to perform the Chip Erase operation to reprogram any memory location in the serial programming mode unless any of the lock bits have been programmed.

In the parallel programming mode, there is no auto-erase cycle. To reprogram any non-blank byte, the user needs to use the Chip Erase operation first to erase both arrays.

**Parallel Programming Algorithm:** To program and verify the AT89S8252 in the parallel programming mode, the following sequence is recommended:

1. Power-up sequence:

   Apply power between $V_{CC}$ and GND pins.

   Set RST pin to "H".

   Apply a 3 MHz to 24 MHz clock to XTAL1 pin and wait for at least 10 milliseconds.
2. Set $\overline{PSEN}$ pin to "L"

   ALE pin to "H"

   $\overline{EA}$ pin to "H" and all other pins to "H".
3. Apply the appropriate combination of "H" or "L" logic levels to pins P2.6, P2.7, P3.6, P3.7 to select one of the programming operations shown in the Flash Programming Modes table.
4. Apply the desired byte address to pins P1.0 to P1.7 and P2.0 to P2.5.

   Apply data to pins P0.0 to P0.7 for Write Code operation.
5. Raise $\overline{EA}/V_{PP}$ to 12V to enable Flash programming, erase or verification.
6. Pulse ALE/$\overline{PROG}$ once to program a byte in the Code memory array, the Data memory array or the lock bits. The byte-write cycle is self-timed and typically takes 1.5 ms.
7. To verify the byte just programmed, bring pin P2.7 to "L" and read the programmed data at pins P0.0 to P0.7.
8. Repeat steps 3 through 7 changing the address and data for the entire 2K or 8K bytes array or until the end of the object file is reached.
9. Power-off sequence:

   Set XTAL1 to "L".

   Set RST and $\overline{EA}$ pins to "L".

   Turn $V_{CC}$ power off.

In the parallel programming mode, there is no auto-erase cycle and to reprogram any non-blank byte, the user needs to use the Chip Erase operation first to erase both arrays.

**Data Polling:** The AT89S8252 features $\overline{DATA}$ Polling to indicate the end of a write cycle. During a write cycle in the parallel or serial programming mode, an attempted read of the last byte written will result in the complement of the written datum on P0.7 (parallel mode), and on the MSB of the serial output byte on MISO (serial mode). Once the write cycle has been completed, true data are valid on all outputs, and the next cycle may begin. $\overline{DATA}$ Polling may begin any time after a write cycle has been initiated.

**Ready/Busy:** The progress of byte programming in the parallel programming mode can also be monitored by the RDY/$\overline{BSY}$ output signal. Pin P3.4 is pulled Low after ALE goes High during programming to indicate $\overline{BUSY}$. P3.4 is pulled High again when programming is done to indicate READY.

**Program Verify:** If lock bits LB1 and LB2 have not been programmed, the programmed Code or Data byte can be read back via the address and data lines for verification. The state of the lock bits can also be verified directly in the parallel programming mode. In the serial programming mode, the state of the lock bits can only be verified indirectly by observing that the lock bit features are enabled.

**Chip Erase:** Both Flash and EEPROM arrays are erased electrically at the same time. In the parallel programming mode, chip erase is initiated by using the proper combination of control signals and by holding ALE/$\overline{PROG}$ low for 10 ms. The Code and Data arrays are written with all "1"s in the Chip Erase operation.



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

In the serial programming mode, a chip erase operation is initiated by issuing the Chip Erase instruction. In this mode, chip erase is self-timed and takes about 16 ms.

During chip erase, a serial read from any address location will return 00H at the data outputs.

**Serial Programming Fuse:** A programmable fuse is available to disable Serial Programming if the user needs maximum system security. The Serial Programming Fuse can only be programmed or erased in the Parallel Programming Mode.

*The AT89S8252 is shipped with the Serial Programming Mode enabled.*

**Reading the Signature Bytes:** The signature bytes are read by the same procedure as a normal verification of locations 030H and 031H, except that P3.6 and P3.7 must be pulled to a logic low. The values returned are as follows:

(030H) = 1EH indicates manufactured by Atmel
(031H) = 72H indicates 89S8252

## Programming Interface

Every code byte in the Flash and EEPROM arrays can be written, and the entire array can be erased, by using the appropriate combination of control signals. The write operation cycle is self-timed and once initiated, will automatically time itself to completion.

All major programming vendors offer worldwide support for the Atmel microcontroller series. Please contact your local programming vendor for the appropriate software revision.

## Serial Downloading

Both the Code and Data memory arrays can be programmed using the serial SPI bus while RST is pulled to $V_{CC}$. The serial interface consists of pins SCK, MOSI (input) and MISO (output). After RST is set high, the Programming Enable instruction needs to be executed first before program/erase operations can be executed.

An auto-erase cycle is built into the self-timed programming operation (in the serial mode ONLY) and there is no need to first execute the Chip Erase instruction unless any of the lock bits have been programmed. The Chip Erase operation turns the content of every memory location in both the Code and Data arrays into FFH.

The Code and Data memory arrays have separate address spaces:

0000H to 1FFFH for Code memory and 000H to 7FFH for Data memory.

Either an external system clock is supplied at pin XTAL1 or a crystal needs to be connected across pins XTAL1 and XTAL2. The maximum serial clock (SCK) frequency should be less than 1/40 of the crystal frequency. With a 24 MHz oscillator clock, the maximum SCK frequency is 600 kHz.

## Serial Programming Algorithm

To program and verify the AT89S8252 in the serial programming mode, the following sequence is recommended:

1. Power-up sequence:
   Apply power between VCC and GND pins.
   Set RST pin to "H".
   If a crystal is not connected across pins XTAL1 and XTAL2, apply a 3 MHz to 24 MHz clock to XTAL1 pin and wait for at least 10 milliseconds.
2. Enable serial programming by sending the Programming Enable serial instruction to pin MOSI/P1.5. The frequency of the shift clock supplied at pin SCK/P1.7 needs to be less than the CPU clock at XTAL1 divided by 40.
3. The Code or Data array is programmed one byte at a time by supplying the address and data together with the appropriate Write instruction. The selected memory location is first automatically erased before new data is written. The write cycle is self-timed and typically takes less than 2.5 ms at 5V.
4. Any memory location can be verified by using the Read instruction which returns the content at the selected address at serial output MISO/P1.6.
5. At the end of a programming session, RST can be set low to commence normal operation.

Power-off sequence (if needed):
Set XTAL1 to "L" (if a crystal is not used).
Set RST to "L".
Turn $V_{CC}$ power off.

## Serial Programming Instruction

The Instruction Set for Serial Programming follows a 3-byte protocol and is shown in the following table:



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**DALLAS SEMICONDUCTOR**

# DS1307
## 64 X 8 Serial Real Time Clock

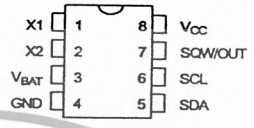
### FEATURES

- Real time clock counts seconds, minutes, hours, date of the month, month, day of the week, and year with leap year compensation valid up to 2100
- 56 byte nonvolatile RAM for data storage
- 2–wire serial interface
- Programmable squarewave output signal
- Automatic power fail detect and switch circuitry
- Consumes less than 500 nA in battery backup mode at 25°C
- Optional industrial temperature range –40°C to +85°C (IND)
- Available in 8–pin DIP or SOIC

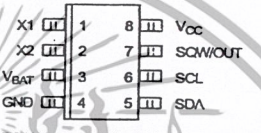
### ORDERING INFORMATION

| | |
|---|---|
| DS1307 | Serial Timekeeping Chip; 8–pin DIP |
| DS1307Z | Serial Timekeeping Chip; 8–pin SOIC (150 mil) |
| DS1307N | 8–pin DIP (IND) |
| DS1307ZN | 8–pin SOIC (IND) |

### PIN ASSIGNMENT




DS1307 8–PIN DIP (300 MIL)




DS1307Z 8–PIN SOIC (150 MIL)

### PIN DESCRIPTION

| | | |
|---|---|---|
| $V_{CC}$ | – | Primary Power Supply |
| X1, X2 | – | 32.768 KHz Crystal Connection |
| $V_{BAT}$ | – | +3 Volt Battery Input |
| GND | – | Ground |
| SDA | – | Serial Data |
| SCL | – | Serial Clock |
| SQW/OUT | – | Square wave/Output Driver |

### DESCRIPTION

The DS1307 Serial Real Time Clock is a low power full BCD clock calendar plus 56 bytes of nonvolatile SRAM. Address and data are transferred serially via a 2–wire bi–directional bus. The 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 AM/PM indicator. The DS1307 has a built–in power sense circuit which detects power failures and automatically switches to the battery supply.

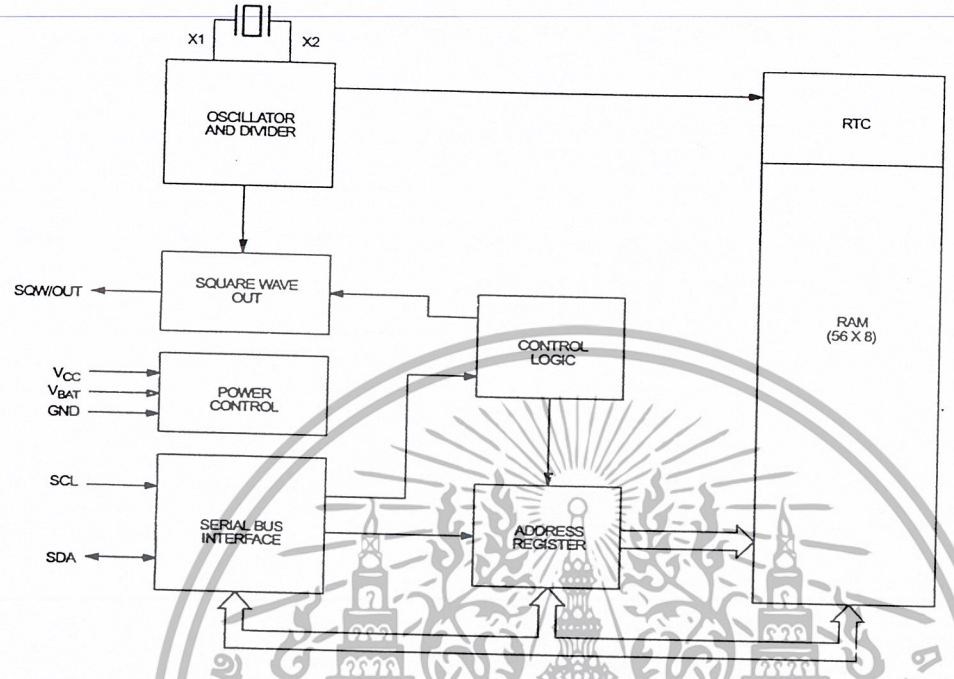
### OPERATION

The DS1307 operates as a slave device on the serial bus. Access is obtained by implementing a START condition and providing a device identification code followed by a register address. Subsequent registers can be accessed sequentially until a STOP condition is executed. When $V_{CC}$ falls below 1.25 x $V_{BAT}$ the device terminates an access in progress and resets the device address counter. Inputs to the device will not be recognized at this time to prevent erroneous data from being written to the device from an out of tolerance system. When $V_{CC}$ falls below $V_{BAT}$ the device switches into a low current battery backup mode. Upon power up, the device switches from battery to $V_{CC}$ when $V_{CC}$ is greater than $V_{BAT}$+0.2V and recognizes inputs when $V_{CC}$ is greater than 1.25 x $V_{BAT}$. The block diagram in Figure 1 shows the main elements of the Serial Real Time Clock. The following paragraphs describe the function of each pin.



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**DS1307 BLOCK DIAGRAM** Figure 1




## SIGNAL DESCRIPTIONS

**$V_{CC}$, GND** – DC power is provided to the device on these pins. $V_{CC}$ is the +5 volt input. When 5 volts are applied within normal limits, the device is fully accessible and data can be written and read. When a 3 volt battery is connected to the device and $V_{CC}$ is below 1.25 x $V_{BAT}$, reads and writes are inhibited. However, the Timekeeping function continues unaffected by the lower input voltage. As $V_{CC}$ falls below $V_{BAT}$ the RAM and timekeeper are switched over to the external 3 volt battery.

**$V_{BAT}$** – Battery input for any standard 3 volt lithium cell or other energy source. Battery voltage must be held between 2.5 and 3.5 volts for proper operation. The nominal write protect trip point voltage at which access to the real time clock and user RAM is denied is set by the internal circuitry as 1.25 x $V_{BAT}$ nominal. A Lithium battery with 35 mAh or greater will back up the DS1307 for more than 10 years in the absence of power.

**SCL (Serial Clock Input)** – SCL is used to synchronize data movement on the serial interface.

**SDA (Serial Data Input/Output)** – SDA is the input/output pin for the 2–wire serial interface. The SDA pin is open drain which requires an external pull–up resistor.

**SQW/OUT (Square Wave/ Output Driver)** – When enabled, the SQWE bit set to 1, the SQW/OUT pin outputs one of four square wave frequencies (1 Hz, 4 KHz, 8 KHz, 32 KHz). The SQW/OUT pin is open drain which requires an external pull–up resistor.

**X1, X2** – Connections for a standard 32.768 KHz quartz crystal. The internal oscillator circuitry is designed for operation with a crystal having a specified load capacitance (CL) of 12.5 pF.

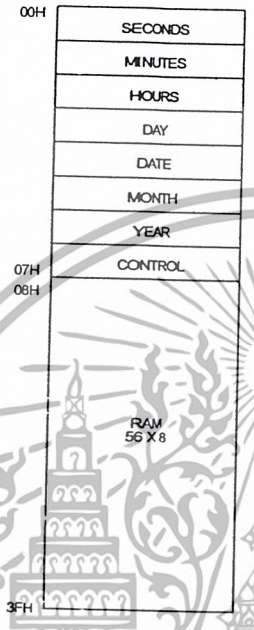


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## RTC AND RAM ADDRESS MAP

The address map for the RTC and RAM registers of the DS1307 is shown in Figure 2. The real time clock registers are located in address locations 00h to 07h. The RAM registers are located in address locations 08h to 3Fh. During a multibyte access, when the address pointer reaches 3Fh, the end of RAM space, it wraps around to location 00h, the beginning of the clock space.

**DS1307 ADDRESS MAP** Figure 2




## CLOCK AND CALENDAR

The time and calendar information is obtained by reading the appropriate register bytes. The real time clock registers are illustrated in Figure 3. The time and calendar are set or initialized by writing the appropriate register bytes. The contents of the time and calendar registers are in the Binary–Coded Decimal (BCD) format. Bit 7 of Register 0 is the Clock Halt (CH) bit. When this bit is set to a one, the oscillator is disabled. When cleared to a zero, the oscillator is enabled.

The DS1307 can be run in either 12–hour or 24–hour mode. Bit 6 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).



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## DS1307 TIMEKEEPER REGISTERS Figure 3

| | BIT7 | | | | | | BIT0 | |
|---|---|---|---|---|---|---|---|---|
| 00H | CH | 10 SECONDS | | SECONDS | | | | 00–59 |
| | X | 10 MINUTES | | MINUTES | | | | 00–59 |
| | X | 12/24 | 10 HR A/P | 10 HR | HOURS | | | 01–12<br>00–23 |
| | X | X | X | X | X | DAY | | 1–7 |
| | X | X | 10 DATE | | DATE | | | 01–28/29<br>01–30<br>01–31 |
| | X | X | 10 MONTH | | MONTH | | | 01–12 |
| | 10 YEAR | | | | YEAR | | | 00–99 |
| 07H | OUT | X | X | SQWE | X | X | RS1 | RS0 |

### CONTROL REGISTER

The DS1307 Control Register is used to control the operation of the SQW/OUT pin.

| BIT 7 | BIT 6 | BIT 5 | BIT 4 | BIT 3 | BIT 2 | BIT 1 | BIT 0 |
|---|---|---|---|---|---|---|---|
| OUT | X | X | SQWE | X | X | RS1 | RS0 |

OUT (Output control): This bit controls the output level of the SQW/OUT pin when the square wave output is disabled. If SQWE = 0, the logic level on the SQW/OUT pin is 1 if OUT = 1 and is 0 if OUT = 0.

SQWE (Square wave Enable): This bit when set to a logic 1 will enable the oscillator output. The frequency of the square wave output depends on the value of the RS0 and RS1 bits.

RS (Rate Select): These bits control the frequency of the square wave output when the square wave output has been enabled. Table 1 lists the square wave frequencies that can be selected with the RS bits.

### SQUAREWAVE OUTPUT FREQUENCY Table 1

| RS1 | RS0 | SQW OUTPUT FREQUENCY |
|---|---|---|
| 0 | 0 | 1 Hz |
| 0 | 1 | 4 KHz |
| 1 | 0 | 8 KHz |
| 1 | 1 | 32 KHz |

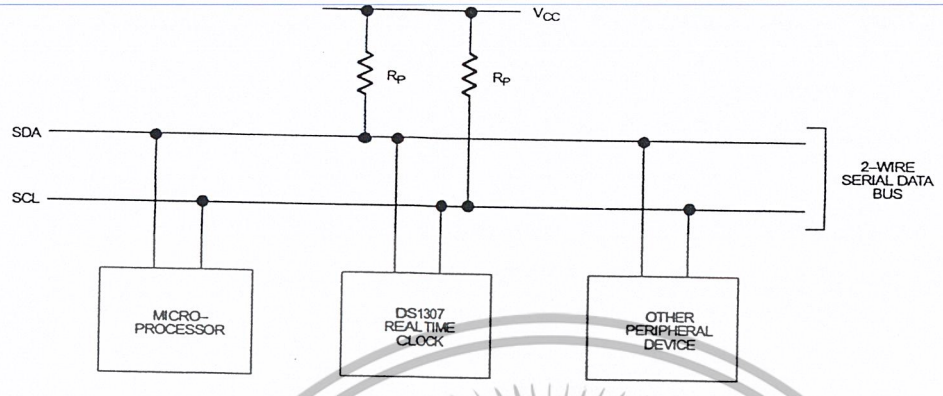
### 2–WIRE SERIAL DATA BUS

The DS1307 supports a bi–directional 2–wire bus and data transmission protocol. A device that sends data onto the bus is defined as a transmitter and a device receiving data as a receiver. The device that controls the message is called a master. The devices that are controlled by the master are slaves. The bus must be controlled by a master device which generates the serial clock (SCL), controls the bus access, and generates the START and STOP conditions. The DS1307 operates as a slave on the 2–wire bus. A typical bus configuration using this 2–wire protocol is show in Figure 4.



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**TYPICAL 2–WIRE BUS CONFIGURATION** Figure 4




The following bus protocol has been defined (see Figure 5).

- Data transfer may be initiated only when the bus is not busy.
- During data transfer, the data line must remain stable whenever the clock line is HIGH. Changes in the data line while the clock line is high will be interpreted as control signals.

Accordingly, the following bus conditions have been defined:

Bus not busy: Both data and clock lines remain HIGH.

Start data transfer: A change in the state of the data line from high to low, while the clock line is high, defines a START condition.

Stop data transfer: A change in the state of the data line from low to high, while the clock line is high defines the STOP condition.

Data valid: The state of the data line represents valid data when, after a START condition, the data line is stable for the duration of the high period of the clock signal. The data on the line must be changed during the low period of the clock signal. There is one clock pulse per bit of data.

Each data transfer is initiated with a START condition and terminated with a STOP condition. The number of data bytes transferred between the START and the STOP conditions is not limited, and is determined by the master device. The information is transferred byte–wise and each receiver acknowledges with a ninth bit.

Acknowledge: Each receiving device, when addressed, is obliged to generate an acknowledge after the reception of each byte. The master device must generate an extra clock pulse which is associated with this acknowledge bit.

A device that acknowledges must pull down the SDA line during the acknowledge clock pulse in such a way that the SDA line is stable low during the high period of the acknowledge related clock pulse. Of course, setup and hold times must be taken into account. When receiving data from a slave a master must signal an end of data to the slave by not generating an acknowledge bit on the last byte that has been clocked out of the slave. In this case, the slave must leave the data line high to enable the master to generate the STOP condition.

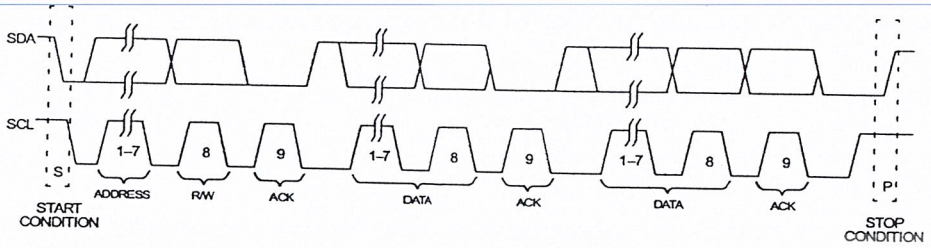
## DATA TRANSFER

Figures 5, 6, and 7 detail how data transfer is accomplished on the 2–wire bus. Depending on the state of the R/$\overline{W}$ bit in the transmission protocols as shown in Figures 6 and 7, two types of data transfer are possible:



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**DATA TRANSFER ON 2–WIRE SERIAL BUS** Figure 5



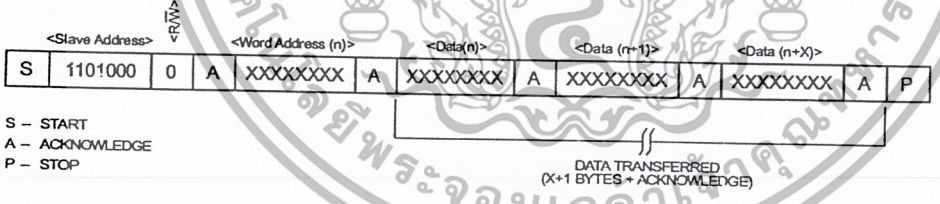

1. Data transfer from a master transmitter to a slave receiver. The first byte transmitted by the master is the slave address. Next follows a number of data bytes. The slave returns an acknowledge bit after each received byte. Data is transfered with the most significant bit (MSB) first.
2. Data transfer from a slave transmitter to a master receiver. The first byte (the slave address) is transmitted by the master. The slave then returns an acknowledge bit. This is followed by the slave transmitting a number of data bytes. The master returns an acknowledge bit after all received bytes other than the last byte. At the end of the last received byte, a 'not acknowledge' is returned.

The master device generates all of the serial clock pulses and the START and STOP conditions. A transfer is ended with a STOP condition or with a repeated START condition. Since a repeated START condition is also the beginning of the next serial transfer, the bus will not be released. Data is transfered with the most significant bit (MSB) first.

The DS1307 may operate in the following two modes:

1. Slave receiver mode (DS1307 write mode): Serial data and clock are received through SDA and SCL. After each byte is received an acknowledge bit is transmitted. START and STOP conditions are recognized as the beginning and end of a serial transfer. Address recognition is performed by hardware after reception of the slave address and direction bit (See Figure 6). The address byte is the first byte received after the start condition is generated by the master. The address byte contains the 7 bit DS1307 address, which is 1101000, followed by the direction bit (R/W\) which for a write is a 0. After receiving and decoding the address byte the DS1307 outputs an acknowledge on the SDA line. After the DS1307 acknowledges the slave address + write bit, the master transmits a register address to the DS1307. This will set the register pointer on the DS1307. The master will then begin transmitting each byte of data with the DS1307 acknowledging each byte received. The master will generate a stop condition to terminate the data write.

**DATA WRITE – SLAVE RECEIVER MODE** Figure 6

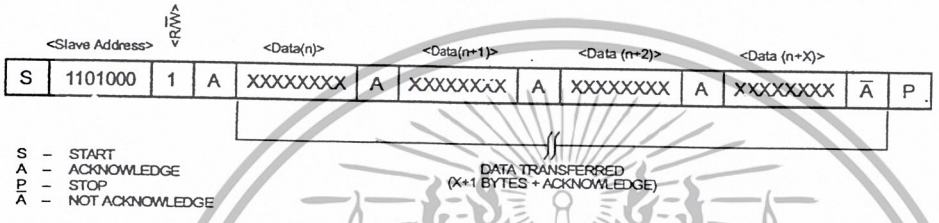





เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. Slave transmitter mode (DS1307 read mode): The first byte is received and handled as in the slave receiver mode. However, in this mode, the direction bit will indicate that the transfer direction is reversed. Serial data is transmitted on SDA by the DS1307 while the serial clock is input on SCL. START and STOP conditions are recognized as the beginning and end of a serial transfer (See Figure 7). The address byte is the first byte received after the start condition is generated by the master. The address byte contains the 7 bit DS1307 address, which is 1101000, followed by the direction bit (R/W) which for a read is a 1. After receiving and decoding the address byte the DS1307 inputs an acknowledge on the SDA line. The DS1307 then begins to transmit data starting with the register address pointed to by the register pointer. If the register pointer is not written to before the initiation of a read mode the first address that is read is the last one stored in the register pointer. The DS1307 must receive a Not Acknowledge to end a read.

**DATA READ – SLAVE TRANSMITTER MODE** Figure 7




S – START
A – ACKNOWLEDGE
P – STOP
Ā – NOT ACKNOWLEDGE



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## ABSOLUTE MAXIMUM RATINGS*

| | |
|---|---|
| Voltage on Any Pin Relative to Ground | –0.5V 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.

The Dallas Semiconductor DS1307 is built to the highest quality standards and manufactured for long term reliability. All Dallas Semiconductor devices are made using the same quality materials and manufacturing methods. However, standard versions of the DS1307 are not exposed to environmental stresses, such as burn–in, that some industrial applications require. Products which have successfully passed through this series of environmental stresses are marked IND or N, denoting their extended operating temperature and reliability rating. For specific reliability information on this product, please contact the factory at (972) 371–4448.

## RECOMMENDED DC OPERATING CONDITIONS

(0°C to 70°C)

| PARAMETER | SYMBOL | MIN | TYP | MAX | UNITS | NOTES |
|---|---|---|---|---|---|---|
| Supply Voltage | $V_{CC}$ | 4.5 | 5.0 | 5.5 | V | 1 |
| Logic 1 | $V_{IH}$ | 2.2 | | $V_{CC}$+0.3 | V | 1 |
| Logic 0 | $V_{IL}$ | –0.3 | | +0.8 | V | 1 |
| $V_{BAT}$ Battery Voltage | $V_{BAT}$ | 2.5 | | 3.5 | V | 1 |

## DC ELECTRICAL CHARACTERISTICS

(0°C to 70°C; $V_{CC}$=4.5V to 5.5V)

| PARAMETER | SYMBOL | MIN | TYP | MAX | UNITS | NOTES |
|---|---|---|---|---|---|---|
| Input Leakage | $I_{LI}$ | | | 1 | µA | 10 |
| I/O Leakage | $I_{LO}$ | | | 1 | µA | 11 |
| Logic 0 Output | $V_{OL}$ | | | 0.4 | V | 2 |
| Active Supply Current | $I_{CCA}$ | | | 1.5 | mA | 9 |
| Standby Current | $I_{CCS}$ | | | 200 | µA | 3 |
| Battery Current (OSC ON); SQW/OUT OFF | $I_{BAT1}$ | | 300 | 500 | nA | 4 |
| Battery Current (OSC ON); SQW/OUT ON (32 KHz) | $I_{BAT2}$ | | 480 | 800 | nA | 4 |



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## AC ELECTRICAL CHARACTERISTICS

(0°C to 70°C; $V_{CC}$=4.5V to 5.5V)

| PARAMETER | SYMBOL | MIN | TYP | MAX | UNITS | NOTES |
|---|---|---|---|---|---|---|
| SCL Clock Frequency | $f_{SCL}$ | 0 | | 100 | KHz | |
| Bus Free Time Between a STOP and START Condition | $t_{BUF}$ | 4.7 | | | µs | |
| Hold Time (Repeated) START Condition | $t_{HD:STA}$ | 4.0 | | | µs | 5 |
| LOW Period of SCL Clock | $t_{LOW}$ | 4.7 | | | µs | |
| HIGH Period of SCL Clock | $t_{HIGH}$ | 4.0 | | | µs | |
| Set–up Time for a Repeated START Condition | $t_{SU:STA}$ | 4.7 | | | µs | |
| Data Hold Time | $t_{HD:DAT}$ | 0 | | | µs | 6, 7 |
| Data Set–up Time | $t_{SU:DAT}$ | 250 | | | ns | |
| Rise Time of Both SDA and SCL Signals | $t_R$ | | | 1000 | ns | |
| Fall Time of Both SDA and SCL Signals | $t_F$ | | | 300 | ns | |
| Set–up Time for STOP Condition | $t_{SU:STO}$ | 4.7 | | | µs | |
| Capacitive Load for each Bus Line | $C_B$ | | | 400 | pF | 8 |
| I/O Capacitance | $C_{I/O}$ | | 10 | | pF | |
| Crystal Capacitance | | | 12.5 | | pF | |

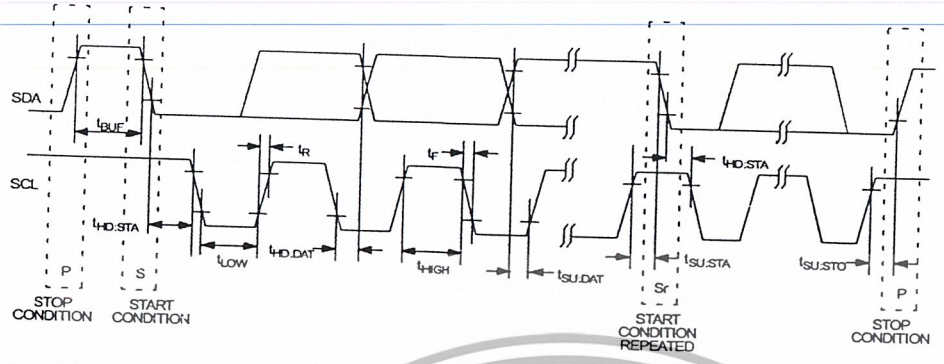
**NOTES:**

1. All voltages are referenced to ground.
2. Logic zero voltages are specified at a sink current of 5 mA at $V_{CC}$=4.5V, $V_{CL}$=GND for capacitive loads.
3. $I_{CCS}$ specified with $V_{CC}$=5.0V and SDA, SCL=5.0V.
4. $V_{CC}$=0V, $V_{BAT}$=3V.
5. After this period, the first clock pulse is generated.
6. A device must internally provide a hold time of at least 300 ns for the SDA signal (referred to the $V_{IHMIN}$ of the SCL signal) in order to bridge the undefined region of the falling edge of SCL.
7. The maximum $t_{HD:DAT}$ has only to be met if the device does not stretch the LOW period ($t_{LOW}$) of the SCL signal.
8. $C_B$ – total capacitance of one bus line in pF.
9. $I_{CCA}$ – SCL clocking at max frequency = 100 KHz.
10. SCL only.
11. SDA and SQW/OUT

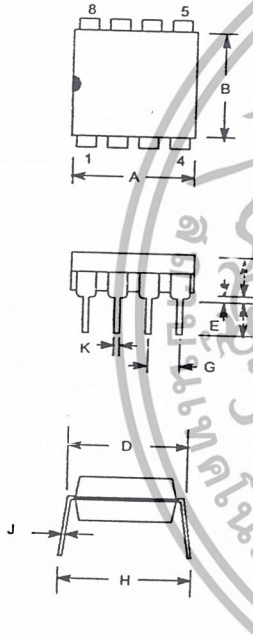


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## TIMING DIAGRAM




## DS1307 64 X 8 SERIAL REAL TIME CLOCK 8–PIN DIP

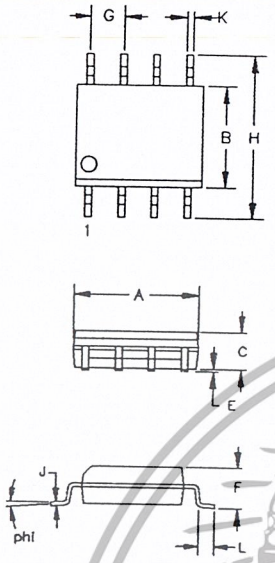



| PKG | 8–PIN | |
|---|---|---|
| DIM | MIN | MAX |
| A IN. | 0.360 | 0.400 |
| MM | 9.14 | 10.16 |
| B IN. | 0.240 | 0.260 |
| MM | 6.10 | 6.60 |
| C IN. | 0.120 | 0.140 |
| MM | 3.05 | 3.56 |
| D IN. | 0.300 | 0.325 |
| MM | 7.62 | 8.26 |
| E IN. | 0.015 | 0.040 |
| MM | 0.38 | 1.02 |
| F IN. | 0.120 | 0.140 |
| MM | 3.04 | 3.56 |
| G IN. | 0.090 | 0.110 |
| MM | 2.29 | 2.79 |
| H IN. | 0.320 | 0.370 |
| MM | 8.13 | 9.40 |
| J IN. | 0.008 | 0.012 |
| MM | 0.20 | 0.30 |
| K IN. | 0.015 | 0.021 |
| MM | 0.38 | 0.53 |



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## DS1307Z 64 X 8 SERIAL REAL TIME CLOCK 8–PIN SOIC (150 MIL)




| PKG | 8–PIN (150 MIL) | |
|---|---|---|
| DIM | MIN | MAX |
| A IN. | 0.188 | 0.196 |
| MM | 4.78 | 4.98 |
| B IN. | 0.150 | 0.158 |
| MM | 3.81 | 4.01 |
| C IN. | 0.048 | 0.062 |
| MM | 1.22 | 1.57 |
| E IN. | 0.004 | 0.010 |
| MM | 0.10 | 0.25 |
| F IN. | 0.053 | 0.069 |
| MM | 1.35 | 1.75 |
| G IN. | 0.050 BSC | |
| MM | 1.27 BSC | |
| H IN. | 0.230 | 0.244 |
| MM | 5.84 | 6.20 |
| J IN. | 0.007 | 0.011 |
| MM | 0.18 | 0.28 |
| K IN. | 0.012 | 0.020 |
| MM | 0.30 | 0.51 |
| L IN. | 0.016 | 0.050 |
| MM | 0.41 | 1.27 |
| phi | 0° | 8° |

56–G2008–001



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

HARRIS SEMICONDUCTOR

# ICL232

## +5V Powered Dual RS-232 Transmitter/Receiver

December 1993

### Features

- Meets All RS-232C Specifications
- Requires Only Single +5V Power Supply
- Onboard Voltage Doubler/Inverter
- Low Power Consumption
- 2 Drivers
  - ±9V Output Swing for +5V Input
  - 300Ω Power-off Source Impedance
  - Output Current Limiting
  - TTL/CMOS Compatible
  - 30V/μs Maximum Slew Rate
- 2 Receivers
  - ±30V Input Voltage Range
  - 3kΩ to 7kΩ Input Impedance
  - 0.5V Hysteresis to Improve Noise Rejection
- All Critical Parameters are Guaranteed Over the Entire Commercial, Industrial and Military Temperature Ranges

### Applications

- Any System Requiring RS-232 Communications Port
  - Computer - Portable and Mainframe
  - Peripheral - Printers and Terminals
  - Portable Instrumentation
  - Modems
  - Dataloggers

### Description

The ICL232 is a dual RS-232 transmitter/receiver interface circuit that meets all EIA RS-232C specifications. It requires a single +5V power supply, and features two onboard charge pump voltage converters which generate +10V and -10V supplies from the 5V supply.

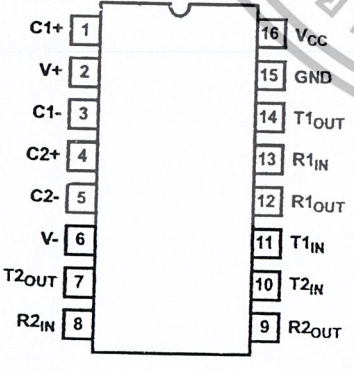
The drivers feature true TTL/CMOS input compatibility, slew-rate-limited output, and 300Ω power-off source impedance. The receivers can handle up to +30V, and have a 3kΩ to 7kΩ input impedance. The receivers also have hysteresis to improve noise rejection.

### Ordering Information

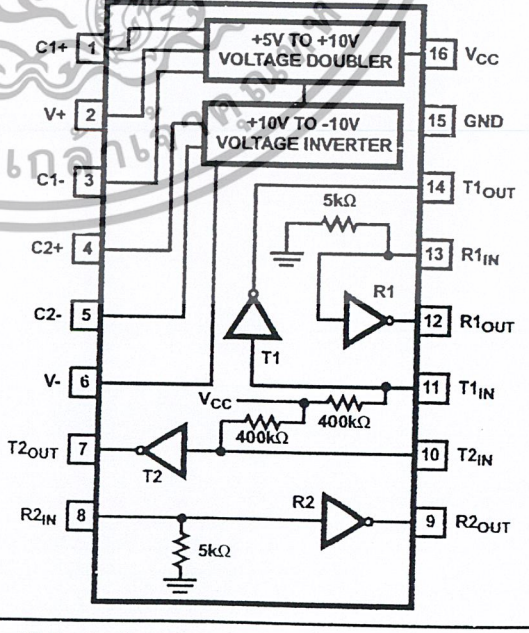
| PART NUMBER | TEMPERATURE RANGE | PACKAGE |
|---|---|---|
| ICL232CPE | 0°C to +70°C | 16 Lead Plastic DIP |
| ICL232CJE | 0°C to +70°C | 16 Lead Ceramic DIP |
| ICL232CBE | 0°C to +70°C | 16 Lead SOIC (W) |
| ICL232IPE | -40°C to +85°C | 16 Lead Plastic DIP |
| ICL232IJE | -40°C to +85°C | 16 Lead Ceramic DIP |
| ICL232IBE | -40°C to +85°C | 16 Lead SOIC (W) |
| ICL232MJE | -55°C to + 125°C | 16 Lead Ceramic DIP |

### Pinouts

ICL232
(PDIP, CDIP, SOIC)
TOP VIEW




### Functional Diagram




CAUTION. These devices are sensitive to electrostatic discharge. Users should follow proper I.C. Handling Procedures.

Copyright © Harris Corporation 1993

File Number **3020.2**



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# Specifications ICL232

## Absolute Maximum Ratings

$V_{CC}$ to Ground ..... (GND -0.3V) <$V_{CC}$ < 6V
V+ to Ground ..... ($V_{CC}$ -0.3V) <V+ < 12V
V- to Ground ..... -12V < V- < (GND +0.3V)
Input Voltages
$T1_{IN}$, $T2_{IN}$ ..... (V- -0.3V) < $V_{IN}$ < (V+ +0.3V)
$R1_{IN}$, $R2_{IN}$ ..... ±30V
Output Voltages
$T1_{OUT}$, $T2_{OUT}$ ..... (V- -0.3V) < $V_{TXOUT}$ < (V+ +0.3V)
$R1_{OUT}$, $R2_{OUT}$ ..... (GND -0.3V) < $V_{RXOUT}$ < ($V_{CC}$ +0.3V)
Short Circuit Duration
$T1_{OUT}$, $T2_{OUT}$ ..... Continuous
$R1_{OUT}$, $R2_{OUT}$ ..... Continuous
Storage Temperature Range ..... -65°C to +150°C
Lead Temperature (Soldering 10s) ..... +300°C

## Thermal Information

| Thermal Resistance | $\theta_{JA}$ | $\theta_{JC}$ |
|---|---|---|
| Ceramic DIP Package | 80°C/W | 24°C/W |
| Plastic DIP Package | 100°C/W | - |
| SOIC Package | 100°C/W | - |

Maximum Power Dissipation ..... 250mW
Operating Temperature Range
ICL232C ..... 0°C to +70°C
ICL232I ..... -40°C to +85°C
ICL232M ..... -55°C to +125°C

*CAUTION: Stresses above those listed in "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress only rating and operation of the device at these or any other conditions above those indicated in the operational sections of this specification is not implied.*

## Electrical Specifications

Test Conditions: $V_{CC}$ = +5V ±10%, $T_A$ = Operating Temperature Range. Test Circuit as in Figure 8 Unless Otherwise Specified

| PARAMETER | TEST CONDITIONS | LIMITS MIN | LIMITS TYP | LIMITS MAX | UNITS |
|---|---|---|---|---|---|
| Transmitter Output Voltage Swing, $T_{OUT}$ | $T1_{OUT}$ and $T2_{OUT}$ loaded with 3kΩ to Ground | ±5 | ±9 | ±10 | V |
| Power Supply Current, $I_{CC}$ | Outputs Unloaded, $T_A$ = +25°C | - | 5 | 10 | mA |
| $T_{IN}$, Input Logic Low, $V_{IL}$ | | - | - | 0.8 | V |
| $T_{IN}$, Input Logic High, $V_{IH}$ | | 2.0 | - | - | V |
| Logic Pullup Current, $I_P$ | $T1_{IN}$, $T2_{IN}$ = 0V | - | 15 | 200 | µA |
| RS-232 Input Voltage Range, $V_{IN}$ | | -30 | - | +30 | V |
| Receiver Input Impedance, $R_{IN}$ | $V_{IN}$ = ±3V | 3.0 | 5.0 | 7.0 | kΩ |
| Receiver Input Low Threshold, $V_{IN}$ (H-L) | $V_{CC}$ = 5.0V, $T_A$ = +25°C | 0.8 | 1.2 | - | V |
| Receiver Input High Threshold, $V_{IN}$ (L-H) | $V_{CC}$ = 5.0V, $T_A$ = +25°C | - | 1.7 | 2.4 | V |
| Receiver Input Hysteresis, $V_{HYST}$ | | 0.2 | 0.5 | 1.0 | V |
| TTL/CMOS Receiver Output Voltage Low, $V_{OL}$ | $I_{OUT}$ = 3.2mA | - | 0.1 | 0.4 | V |
| TTL/CMOS Receiver Output Voltage High, $V_{OH}$ | $I_{OUT}$ = -1.0mA | 3.5 | 4.6 | - | V |
| Propagation Delay, $t_{PD}$ | RS-232 to TTL | - | 0.5 | - | µs |
| Instantaneous Slew Rate, SR | $C_L$ = 10pF, $R_L$ = 3kΩ, $T_A$ = +25°C (Notes 1, 2) | - | - | 30 | V/µs |
| Transition Region Slew Rate, $SR_T$ | $R_L$ = 3kΩ, $C_L$ = 2500pF Measured from +3V to -3V or -3V to +3V | - | 3 | - | V/µs |
| Output Resistance, $R_{OUT}$ | $V_{CC}$ = V+ = V- = 0V, $V_{OUT}$ = ±2V | 300 | - | - | Ω |
| RS-232 Output Short Circuit Current, $I_{SC}$ | $T1_{OUT}$ or $T2_{OUT}$ shorted to GND | - | ±10 | - | mA |

NOTES:

1. Guaranteed by design.
2. See Figure 4 for definition.



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## Typical Performance Curves

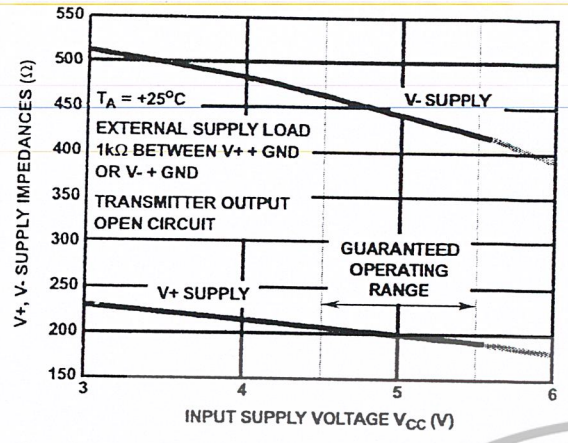



FIGURE 1. V+, V- OUTPUT IMPEDANCES vs $V_{CC}$

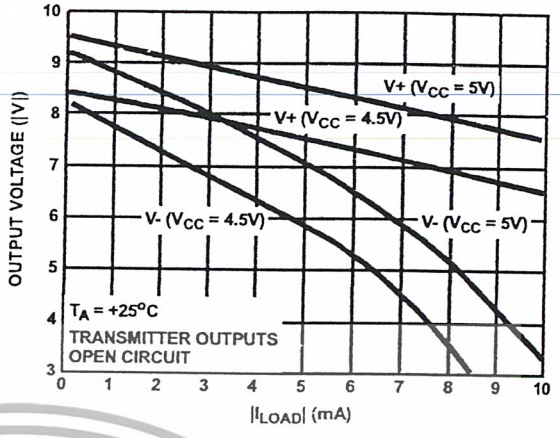



FIGURE 2. V+, V- OUTPUT VOLTAGES vs LOAD CURRENT

## Pin Descriptions

| PLASTIC DIP, CERAMIC DIP | SOIC | PIN NAME | DESCRIPTION |
|---|---|---|---|
| 1 | 1 | C1+ | External capacitor "+" for internal voltage doubler. |
| 2 | 2 | V+ | Internally generated +10V (typical) supply. |
| 3 | 3 | C1- | External capacitor "-" for internal voltage doubler. |
| 4 | 4 | C2+ | External capacitor "+" internal voltage inverter. |
| 5 | 5 | C2- | External capacitor "-" internal voltage inverter. |
| 6 | 6 | V- | Internally generated -10V (typical) supply. |
| 7 | 7 | $T2_{OUT}$ | RS-232 Transmitter 2 output ±10V (typical). |
| 8 | 8 | $R2_{IN}$ | RS-232 Receiver 2 input, with internal 5K pulldown resistor to GND. |
| 9 | 9 | R2out | Receiver 2 TTL/CMOS output. |
| 10 | 10 | $T2_{IN}$ | Transmitter 2 TTL/CMOS input, with internal 400K pullup resistor to $V_{CC}$. |
| 11 | 11 | $T1_{IN}$ | Transmitter 1 TTL/CMOS input, with internal 400K pullup resistor to $V_{CC}$. |
| 12 | 12 | $R1_{OUT}$ | Receiver 1 TTL/CMOS output. |
| 13 | 13 | $R1_{IN}$ | RS-232 Receiver 1 input, with internal 5K pulldown resistor to GND. |
| 14 | 14 | $T1_{OUT}$ | RS-232 Transmitter 1 output ±10V (typical). |
| 15 | 15 | GND | Supply Ground. |
| 16 | 16 | VCC | Positive Power Supply +5V ±10% |



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## Detailed Description

The ICL232 is a dual RS-232 transmitter/receiver powered by a single +5V power supply which meets all EIA RS232C specifications and features low power consumption. The functional diagram illustrates the major elements of the ICL232. The circuit is divided into three sections: a voltage doubler/inverter, dual transmitters, and dual receivers.

### Voltage Converter

An equivalent circuit of the dual charge pump is illustrated in Figure 3.

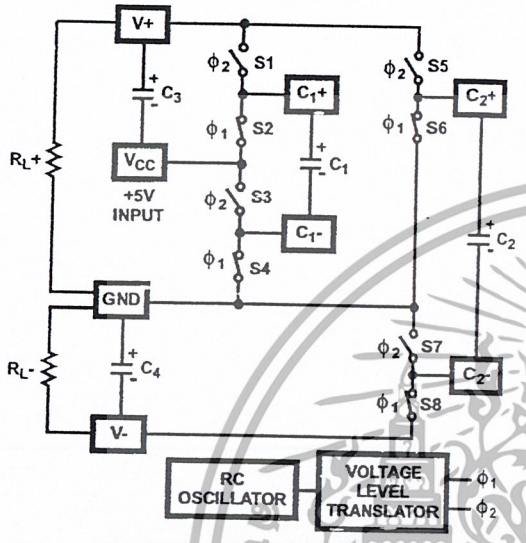
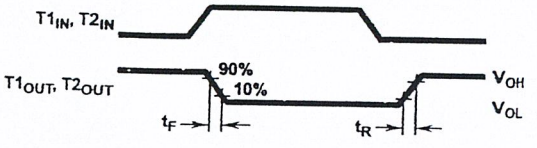



FIGURE 3. DUAL CHARGE PUMP

The voltage quadrupler contains two charge pumps which use two phases of an internally generated clock to generate +10V and -10V. The nominal clock frequency is 16kHz. During phase one of the clock, capacitor C1 is charged to $V_{CC}$. During phase two, the voltage on C1 is added to $V_{CC}$, producing a signal across C2 equal to twice $V_{CC}$. At the same time, C3 is also charged to $2V_{CC}$, and then during phase one, it is inverted with respect to ground to produce a signal across C4 equal to $-2V_{CC}$. The voltage converter accepts input voltages up to 5.5V. The output impedance of the doubler (V+) is approximately 200Ω, and the output impedance of the inverter (V-) is approximately 450Ω. Typical graphs are presented which show the voltage converters output vs input voltage and output voltages vs load characteristics. The test circuit (Figure 8) uses 1µF capacitors for C1-C4, however, the value is not critical. Increasing the values of C1 and C2 will lower the output impedance of the voltage doubler and inverter, and increasing the values of the reservoir capacitors, C3 and C4, lowers the ripple on the V+ and V- supplies.




Instantaneous Slew Rate (SR) = $\frac{(0.8)(V_{OH} - V_{OL})}{t_R}$ or $\frac{(0.8)(V_{OL} - V_{OH})}{t_F}$

FIGURE 4. SLEW RATE DEFINITION

### Transmitters

The transmitters are TTL/CMOS compatible inverters which translate the inputs to RS-232 outputs. The input logic threshold is about 26% of $V_{CC}$, or 1.3V for $V_{CC}$ = 5V. A logic 1 at the input results in a voltage of between -5V and V- at the output, and a logic 0 results in a voltage between +5V and (V+ -0.6V). Each transmitter input has an internal 400kΩ pullup resistor so any unused input can be left unconnected and its output remains in its low state. The output voltage swing meets the RS-232C specification of ±5V minimum with the worst case conditions of: both transmitters driving 3kΩ minimum load impedance, $V_{CC}$ = 4.5V, and maximum allowable operating temperature. The transmitters have an internally limited output slew rate which is less than 30V/µs. The outputs are short circuit protected and can be shorted to ground indefinitely. The powered down output impedance is a minimum of 300Ω with ±2V applied to the outputs and $V_{CC}$ = 0V.

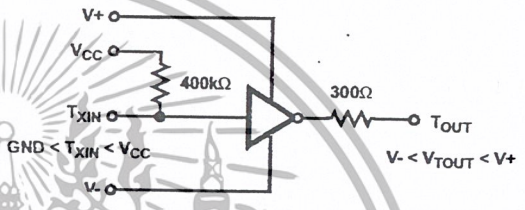



FIGURE 5. TRANSMITTER

### Receivers

The receiver inputs accept up to ±30V while presenting the required 3kΩ to 7kΩ input impedance even it the power is off ($V_{CC}$ = 0V). The receivers have a typical input threshold of 1.3V which is within the ±3V limits, known as the transition region, of the RS-232 specification. The receiver output is 0V to $V_{CC}$. The output will be low whenever the input is greater than 2.4V and high whenever the input is floating or driven between +0.8V and -30V. The receivers feature 0.5V hysteresis to improve noise rejection.

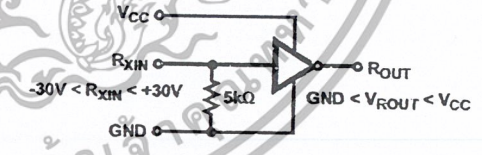
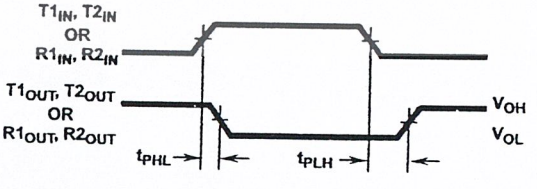



FIGURE 6. RECEIVER




Average Propagation Delay = $\frac{t_{PHL} + t_{PLH}}{2}$

FIGURE 7. PROPAGATION DELAY DEFINITION



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## Test Circuits

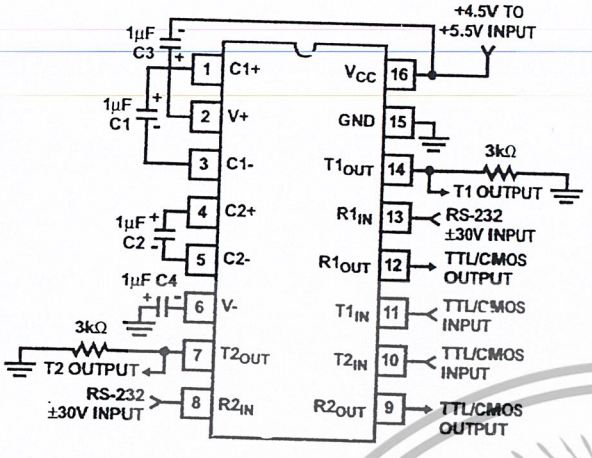



FIGURE 8. GENERAL TEST CIRCUIT

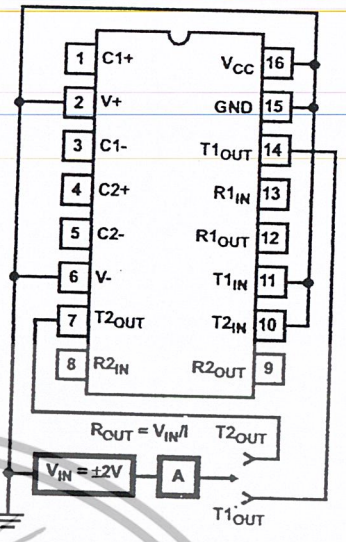



FIGURE 9. POWER-OFF SOURCE RESISTANCE CONFIGURATION

## Applications

The ICL232 may be used for all RS-232 data terminal and communication links. It is particularly useful in applications where ±12V power supplies are not available for conventional RS-232 interface circuits. The applications presented represent typical interface configurations.

A simple duplex RS-232 port with CTS/RTS handshaking is illustrated in Figure 10. Fixed output signals such as DTR (data terminal ready) and DSRS (data signaling rate select) is generated by driving them through a 5kΩ resistor connected to V+.

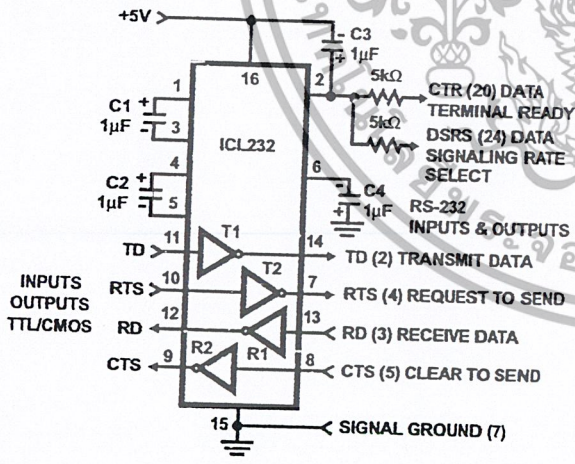



FIGURE 10. SIMPLE DUPLEX RS-232 PORT WITH CTS/RTS HANDSHAKING

In applications requiring four RS-232 inputs and outputs (Figure 11), note that each circuit requires two charge pump capacitors (C1 and C2) but can share common reservoir capacitors (C3 and C4). The benefit of sharing common reservoir capacitors is the elimination of two capacitors and the reduction of the charge pump source impedance which effectively increases the output swing of the transmitters.

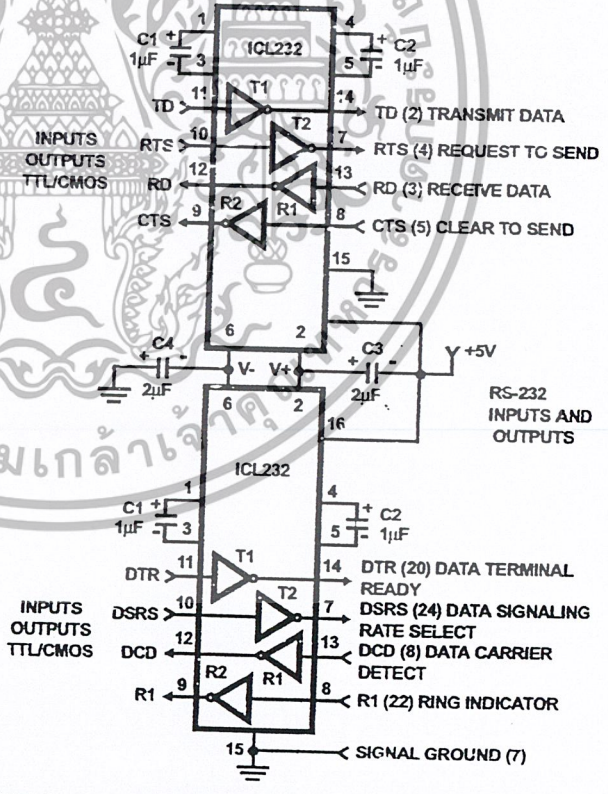



FIGURE 11. COMBINING TWO ICL232s FOR 4 PAIRS OF RS-232 INPUTS AND OUTPUTS



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## Features

- Low Voltage and Standard Voltage Operation
  - 5.0 ($V_{CC}$ = 4.5V to 5.5V)
  - 2.7 ($V_{CC}$ = 2.7V to 5.5V)
  - 1.8 ($V_{CC}$ = 1.8V to 3.6V)
- Internally Organized 16,384 x 8 and 32,768 x 8
- 2-Wire Serial Interface
- Schmitt Trigger, Filtered Inputs for Noise Suppression
- Bidirectional Data Transfer Protocol
- 1 MHz (5V), 400 kHz (2.7V) and 100 kHz (1.8V) Compatibility
- Write Protect Pin for Hardware and Software Data Protection
- 64-Byte Page Write Mode (Partial Page Writes Allowed)
- Self-Timed Write Cycle (5 ms typical)
- High Reliability
  - Endurance: 100,000 Write Cycles
  - Data Retention: 40 Years
  - ESD Protection: > 4000V
- Automotive Grade and Extended Temperature Devices Available
- 8-Pin JEDEC PDIP, 8-Pin JEDEC and EIAJ SOIC, 14-Pin TSSOP, and 8-Pin Leadless Array Packages

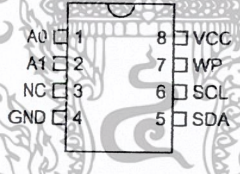
## Description

The AT24C128/256 provides 131,072/262,144 bits of serial electrically erasable and programmable read only memory (EEPROM) organized as 16,384/32,768 words of 8 bits each. The device's cascadable feature allows up to 4 devices to share a common 2-wire bus. The device is optimized for use in many industrial and commercial applications where low power and low voltage operation are essential. The devices are available in space-saving 8-pin JEDEC PDIP, 8-pin EIAJ, 8-pin JEDEC SOIC, 14-pin TSSOP, and 8-pin LAP packages. In addition, the entire family is available in 5.0V (4.5V to 5.5V), 2.7V (2.7V to 5.5V) and 1.8V (1.8V to 3.6V) versions.

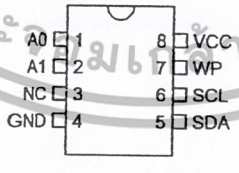
## Pin Configurations

| Pin Name | Function |
|---|---|
| A0 - A1 | Address Inputs |
| SDA | Serial Data |
| SCL | Serial Clock Input |
| WP | Write Protect |
| NC | No Connect |

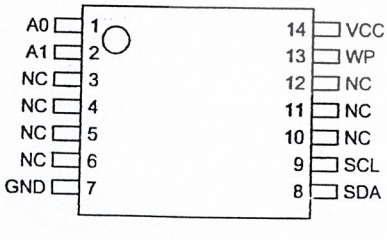
8-Pin PDIP



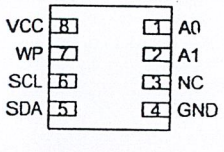

8-Pin SOIC




14-Pin TSSOP




8-Pin Leadless Array




Bottom View

# 2-Wire Serial EEPROMs

128K (16,384 x 8)

256K (32,768 x 8)

# AT24C128
# AT24C256

Rev. 0670C–08/98

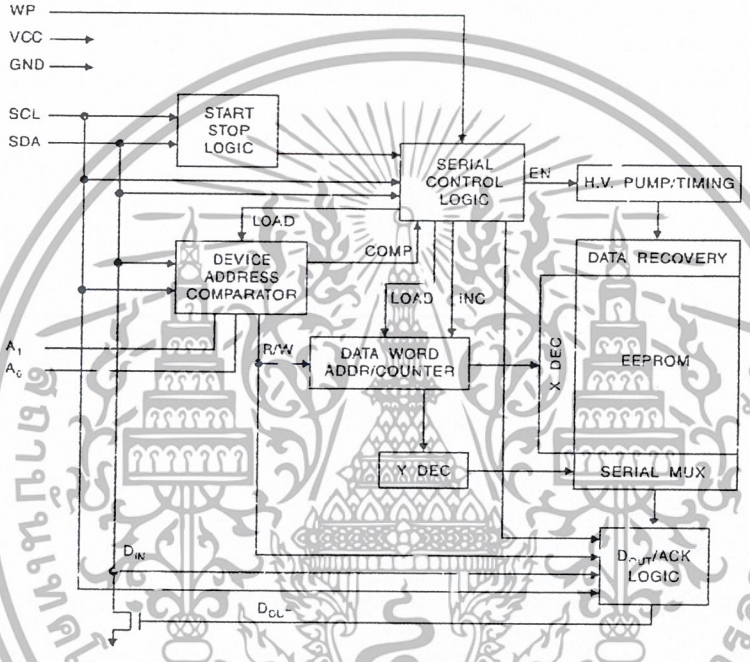
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## Absolute Maximum Ratings*

| | |
|---|---|
| Operating Temperature | -55°C to +125°C |
| Storage Temperature | -65°C to +150°C |
| Voltage on Any Pin with Respect to Ground | -1.0V to +7.0V |
| Maximum Operating Voltage | 6.25V |
| DC Output Current | 5.0 mA |

*NOTICE: Stresses beyond those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other conditions beyond those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

## Block Diagram




## Pin Description

**SERIAL CLOCK (SCL):** The SCL input is used to positive edge clock data into each EEPROM device and negative edge clock data out of each device.

**SERIAL DATA (SDA):** The SDA pin is bidirectional for serial data transfer. This pin is open-drain driven and may be wire-ORed with any number of other open-drain or open collector devices.

**DEVICE/PAGE ADDRESSES (A1, A0):** The A1 and A0 pins are device address inputs that are hardwired or left not connected for hardware compatibility with AT24C32/64. When the pins are hardwired, as many as four 128K/256K devices may be addressed on a single bus system (device addressing is discussed in detail under the Device Addressing section). When the pins are not hardwired, the default $A_1$ and $A_0$ are zero.

**WRITE PROTECT (WP):** The write protect input, when tied to GND, allows normal write operations. When WP is tied high to $V_{CC}$, all write operations to the memory are inhibited. If left unconnected, WP is internally pulled down to GND. Switching WP to $V_{CC}$ prior to a write operation creates a software write protect function.

## Memory Organization

**AT24C128/256, 128K/256K SERIAL EEPROM:** The 128K/256K is internally organized as 256/512 pages of 64-bytes each. Random word addressing requires a 14/15-bit data word address.



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## Pin Capacitance[1]

Applicable over recommended operating range from $T_A$ = 25°C, f = 1.0 MHz, $V_{CC}$ = +1.8V.

| Symbol | Test Condition | Max | Units | Conditions |
|---|---|---|---|---|
| $C_{I/O}$ | Input/Output Capacitance (SDA) | 8 | pF | $V_{I/O}$ = 0V |
| $C_{IN}$ | Input Capacitance ($A_0$, $A_1$, SCL) | 6 | pF | $V_{IN}$ = 0V |

Note: This parameter is characterized and is not 100% tested.

## DC Characteristics

Applicable over recommended operating range from: $T_{AI}$ = -40°C to +85°C, $V_{CC}$ = +1.8V to +5.5V, $T_{AC}$ = 0°C to +70°C, $V_{CC}$ = +1.8V to +5.5V (unless otherwise noted).

| Symbol | Parameter | Test Condition | Min | Typ | Max | Units |
|---|---|---|---|---|---|---|
| $V_{CC1}$ | Supply Voltage | | 1.8 | | 3.6 | V |
| $V_{CC2}$ | Supply Voltage | | 2.7 | | 5.5 | V |
| $V_{CC3}$ | Supply Voltage | | 4.5 | | 5.5 | V |
| $I_{CC1}$ | Supply Current | $V_{CC}$ = 5.0V READ at 400 kHz | | 1.0 | 2.0 | mA |
| $I_{CC2}$ | Supply Current | $V_{CC}$ = 5.0V WRITE at 400 kHz | | 2.0 | 3.0 | mA |
| $I_{SB1}$ | Standby Current (1.8V option) | $V_{CC}$ = 1.8V $V_{IN}$ = $V_{CC}$ or $V_{SS}$ | | | 0.2 | µA |
| | | $V_{CC}$ = 3.6V | | | 2.0 | |
| $I_{SB2}$ | Standby Current (2.7V option) | $V_{CC}$ = 2.7V $V_{IN}$ = $V_{CC}$ or $V_{SS}$ | | | 0.5 | µA |
| | | $V_{CC}$ = 5.5V | | | 6.0 | |
| $I_{SB3}$ | Standby Current (5.0V option) | $V_{CC}$ = 4.5 - 5.5V $V_{IN}$ = $V_{CC}$ or $V_{SS}$ | | | 6.0 | µA |
| $I_{LI}$ | Input Leakage Current | $V_{IN}$ = $V_{CC}$ or $V_{SS}$ | | 0.10 | 3.0 | µA |
| $I_{LO}$ | Output Leakage Current | $V_{OUT}$ = $V_{CC}$ or $V_{SS}$ | | 0.05 | 3.0 | µA |
| $V_{IL}$ | Input Low Level[1] | | -0.6 | | $V_{CC}$ x 0.3 | V |
| $V_{IH}$ | Input High Level[1] | | $V_{CC}$ x 0.7 | | $V_{CC}$ + 0.5 | V |
| $V_{OL2}$ | Output Low Level | $V_{CC}$ = 3.0V $I_{OL}$ = 2.1 mA | | | 0.4 | V |
| $V_{OL1}$ | Output Low Level | $V_{CC}$ = 1.8V $I_{OL}$ = 0.15 mA | | | 0.2 | V |

Note: 1. $V_{IL}$ min and $V_{IH}$ max are reference only and are not tested



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## AC Characteristics

Applicable over recommended operating range from $T_A$ = -40°C to +85°C, $V_{CC}$ = +1.8V to +5.5V, CL = 100 pF (unless otherwise noted). Test conditions are listed in Note 2.

| Symbol | Parameter | 1.8-volt | | 2.7-volt | | 5.0-volt | | Units |
|---|---|---|---|---|---|---|---|---|
| | | Min | Max | Min | Max | Min | Max | |
| $f_{SCL}$ | Clock Frequency, SCL | | 100 | | 400 | | 1000 | kHz |
| $t_{LOW}$ | Clock Pulse Width Low | 4.7 | | 1.3 | | 0.6 | | µs |
| $t_{HIGH}$ | Clock Pulse Width High | 4.0 | | 1.0 | | 0.4 | | µs |
| $t_{AA}$ | Clock Low to Data Out Valid | 0.1 | 4.5 | 0.05 | 0.9 | 0.05 | 0.55 | µs |
| $t_{BUF}$ | Time the bus must be free before a new transmission can start(1) | 4.7 | | 1.3 | | 0.5 | | µs |
| $t_{HD.STA}$ | Start Hold Time | 4.0 | | 0.6 | | 0.25 | | µs |
| $t_{SU.STA}$ | Start Set-up Time | 4.7 | | 0.6 | | 0.25 | | µs |
| $t_{HD.DAT}$ | Data In Hold Time | 0 | | 0 | | 0 | | µs |
| $t_{SU.DAT}$ | Data In Set-up Time | 200 | | 100 | | 100 | | ns |
| $t_R$ | Inputs Rise Time(1) | | 1.0 | | 0.3 | | 0.3 | µs |
| $t_F$ | Inputs Fall Time(1) | | 300 | | 300 | | 100 | ns |
| $t_{SU.STO}$ | Stop Set-up Time | 4.7 | | 0.6 | | 0.25 | | µs |
| $t_{DH}$ | Data Out Hold Time | 100 | | 50 | | 50 | | ns |
| $t_{WR}$ | Write Cycle Time | | 20 | | 10 | | 10 | ms |
| Endurance(1) | 5.0V, 25°C, Page Mode | 100K | | 100K | | 100K | | Write Cycles |

Notes:
1. This parameter is characterized and is not 100% tested.
2. AC measurement conditions:
   $R_L$ (connects to $V_{CC}$): 1.3KΩ (2.7V, 5V), 10KΩ (1.8V)
   Input pulse voltages: $0.3V_{CC}$ to $0.7V_{CC}$
   Input rise and fall times: ≤50ns
   Input and output timing reference voltages: $0.5V_{CC}$

## Device Operation

**CLOCK and DATA TRANSITIONS:** The SDA pin is normally pulled high with an external device. Data on the SDA pin may change only during SCL low time periods (refer to Data Validity timing diagram). Data changes during SCL high periods will indicate a start or stop condition as defined below.

**START CONDITION:** A high-to-low transition of SDA with SCL high is a start condition which must precede any other command (refer to Start and Stop Definition timing diagram).

**STOP CONDITION:** A low-to-high transition of SDA with SCL high is a stop condition. After a read sequence, the stop command will place the EEPROM in a standby power mode (refer to Start and Stop Definition timing diagram).

**ACKNOWLEDGE:** All addresses and data words are serially transmitted to and from the EEPROM in 8-bit words. The EEPROM sends a zero during the ninth clock cycle to acknowledge that it has received each word.

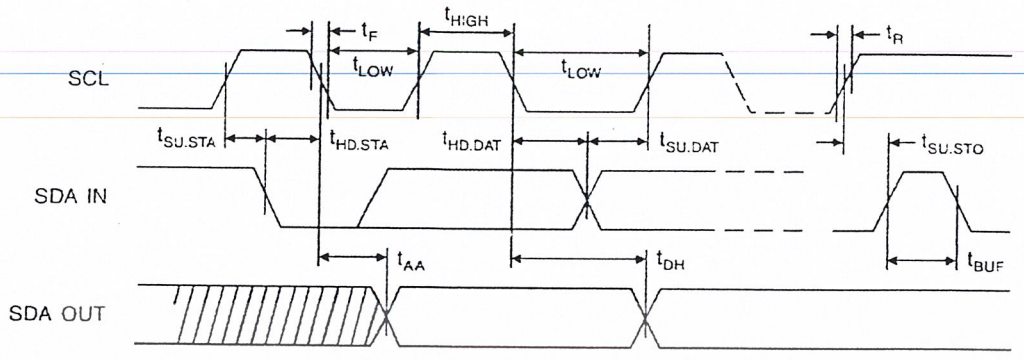
**STANDBY MODE:** The AT24C128/256 features a low power standby mode which is enabled: a) upon power-up and b) after the receipt of the STOP bit and the completion of any internal operations.

**MEMORY RESET:** After an interruption in protocol, power loss or system reset, any 2-wire part can be reset by following these steps: (a) Clock up to 9 cycles, (b) look for SDA high in each cycle while SCL is high and then (c) create a start condition as SDA is high.

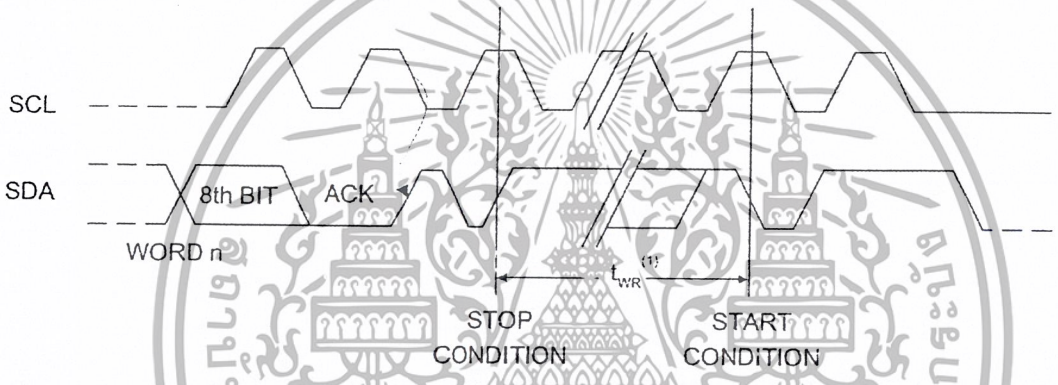


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## Bus Timing (SCL: Serial Clock, SDA: Serial Data I/O)




## Write Cycle Timing (SCL: Serial Clock, SDA: Serial Data I/O)

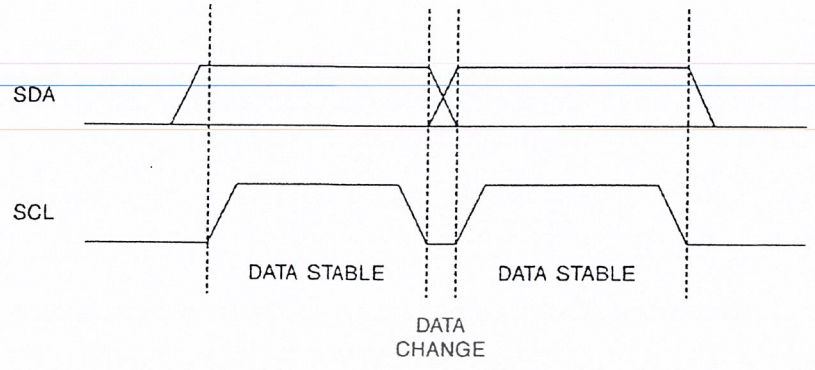



Note: 1. The write cycle time $t_{WR}$ is the time from a valid stop condition of a write sequence to the end of the internal clear/write cycle.

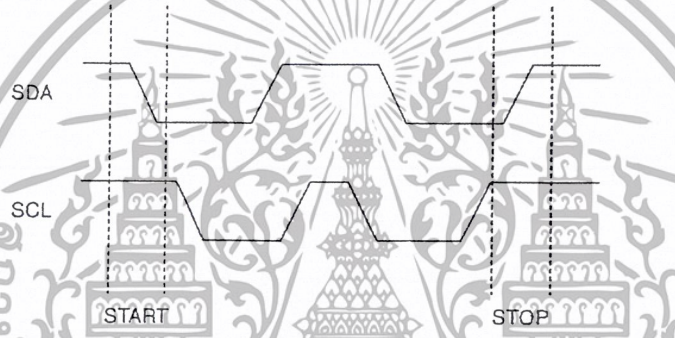


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

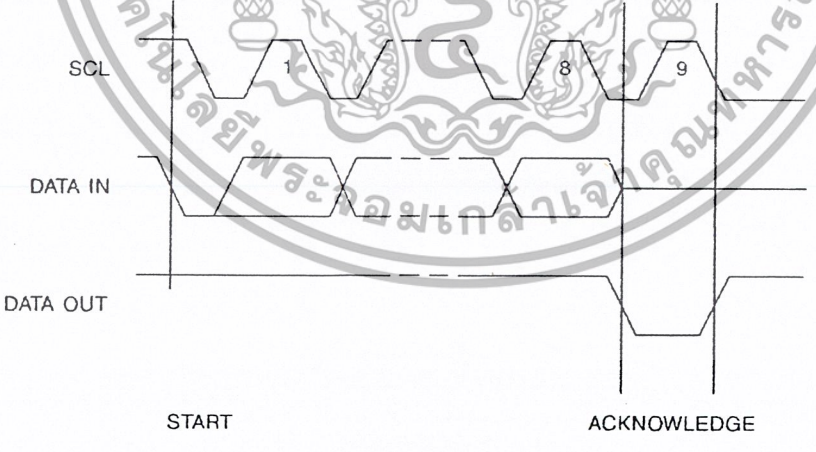
## Data Validity




## Start and Stop Definition




## Cutput Acknowledge






เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## Device Addressing

The 128K/256K EEPROM requires an 8-bit device address word following a start condition to enable the chip for a read or write operation (refer to Figure 1). The device address word consists of a mandatory one, zero sequence for the first five most significant bits as shown. This is common to all 2-wire EEPROM devices.

The 128K/256K uses the two device address bits A1, A0 to allow as many as four devices on the same bus. These bits must compare to their corresponding hardwired input pins. The A1 and A0 pins use an internal proprietary circuit that biases them to a logic low condition if the pins are allowed to float.

The eighth bit of the device address is the read/write operation select bit. A read operation is initiated if this bit is high and a write operation is initiated if this bit is low.

Upon a compare of the device address, the EEPROM will output a zero. If a compare is not made, the device will return to a standby state.

**DATA SECURITY:** The AT24C128/256 has a hardware data protection scheme that allows the user to write protect the whole memory when the WP pin is at $V_{CC}$.

## Write Operations

**BYTE WRITE:** A write operation requires two 8-bit data word addresses following the device address word and acknowledgment. Upon receipt of this address, the EEPROM will again respond with a zero and then clock in the first 8-bit data word. Following receipt of the 8-bit data word, the EEPROM will output a zero. The addressing device, such as a microcontroller, then must terminate the write sequence with a stop condition. At this time the EEPROM enters an internally-timed write cycle, $t_{WR}$, to the nonvolatile memory. All inputs are disabled during this write cycle and the EEPROM will not respond until the write is complete (refer to Figure 2).

**PAGE WRITE:** The 128K/256K EEPROM is capable of 64-byte page writes.

A page write is initiated the same way as a byte write, but the microcontroller does not send a stop condition after the first data word is clocked in. Instead, after the EEPROM acknowledges receipt of the first data word, the microcontroller can transmit up to 63 more data words. The EEPROM will respond with a zero after each data word received. The microcontroller must terminate the page write sequence with a stop condition (refer to Figure 3).

The data word address lower 6 bits are internally incremented following the receipt of each data word. The higher data word address bits are not incremented, retaining the memory page row location. When the word address, internally generated, reaches the page boundary, the following byte is placed at the beginning of the same page. If more than 64 data words are transmitted to the EEPROM, the data word address will "roll over" and previous data will be overwritten. The address "roll over" during write is from the last byte of the current page to the first byte of the same page.

**ACKNOWLEDGE POLLING:** Once the internally-timed write cycle has started and the EEPROM inputs are disabled, acknowledge polling can be initiated. This involves sending a start condition followed by the device address word. The read/write bit is representative of the operation desired. Only if the internal write cycle has completed will the EEPROM respond with a zero, allowing the read or write sequence to continue.

## Read Operations

Read operations are initiated the same way as write operations with the exception that the read/write select bit in the device address word is set to one. There are three read operations: current address read, random address read and sequential read.

**CURRENT ADDRESS READ:** The internal data word address counter maintains the last address accessed during the last read or write operation, incremented by one. This address stays valid between operations as long as the chip power is maintained. The address "roll over" during read is from the last byte of the last memory page, to the first byte of the first page.

Once the device address with the read/write select bit set to one is clocked in and acknowledged by the EEPROM, the current address data word is serially clocked out. The microcontroller does not respond with an input zero but does generate a following stop condition (refer to Figure 4).

**RANDOM READ:** A random read requires a "dummy" byte write sequence to load in the data word address. Once the device address word and data word address are clocked in and acknowledged by the EEPROM, the microcontroller must generate another start condition. The microcontroller now initiates a current address read by sending a device address with the read/write select bit high. The EEPROM acknowledges the device address and serially clocks out the data word. The microcontroller does not respond with a zero but does generate a following stop condition (refer to Figure 5).

**SEQUENTIAL READ:** Sequential reads are initiated by either a current address read or a random address read. After the microcontroller receives a data word, it responds with an acknowledge. As long as the EEPROM receives an acknowledge, it will continue to increment the data word address and serially clock out sequential data words. When the memory address limit is reached, the data word address will "roll over" and the sequential read will continue. The sequential read operation is terminated when the microcontroller does not respond with a zero but does generate a following stop condition (refer to Figure 6).



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**Figure 1.** Device Address

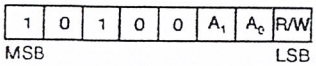



**Figure 2.** Byte Write

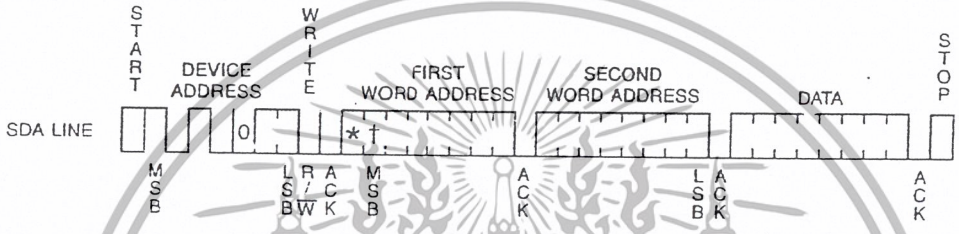
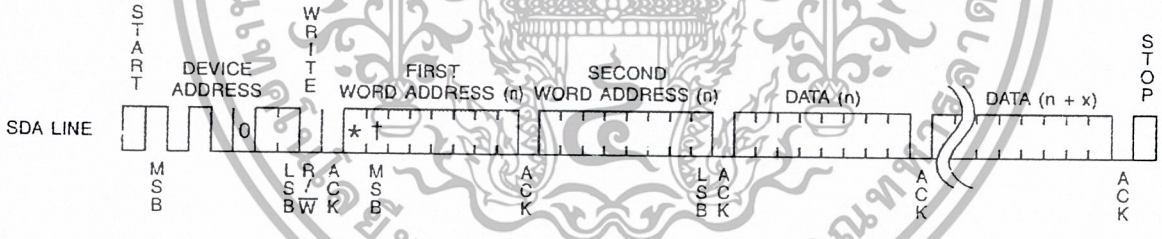



**Figure 3.** Page Write




(* = DON'T CARE bit)
(† = DON'T CARE bit for the 128K)



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**Figure 4.** Current Address Read

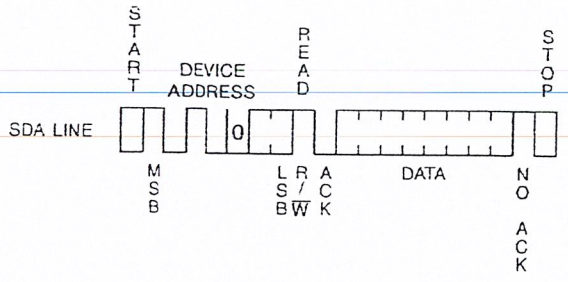
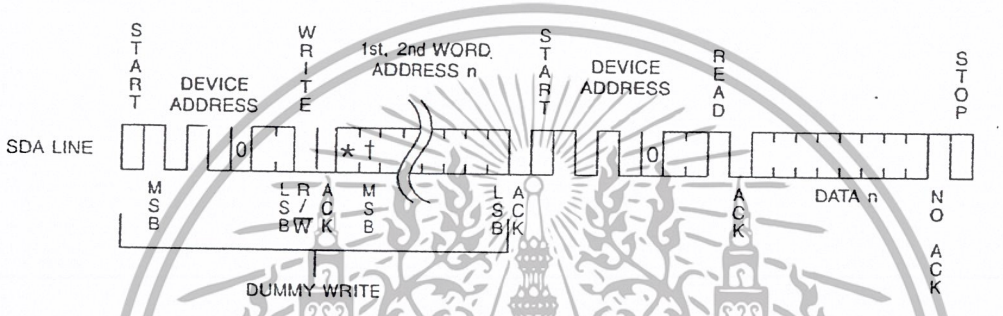
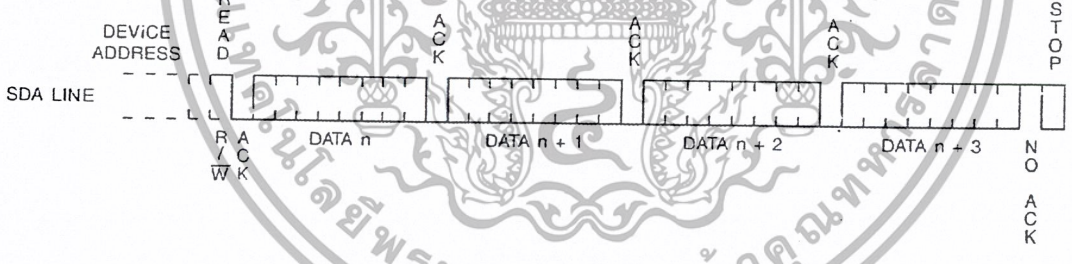



**Figure 5.** Random Read




(* = DON'T CARE bit)
(† = DON'T CARE bit for the 128K)

**Figure 6.** Sequential Read






เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## AT24C128 Ordering Information

| $t_{WR}$ (max) (ms) | $I_{CC}$ (max) (µA) | $I_{SB}$ (max) (µA) | $f_{MAX}$ (kHz) | Ordering Code | Package | Operation Range |
|---|---|---|---|---|---|---|
| 10 | 3000 | 6.0 | 1000 | AT24C128-10PC | 8P3 | Commercial (0°C to 70°C) |
| | | | | AT24C128N-10SC | 8S1 | |
| | | | | AT24C128W-10SC | 8S2 | |
| | | | | AT24C128-10CC | 8C | |
| | | | | AT24C128C1-10CC | 8C1 | |
| | | | | AT24C128T1-10TC | 14T | |
| | 3000 | 6.0 | 1000 | AT24C128-10PI | 8P3 | Industrial (-40°C to 85°C) |
| | | | | AT24C128N-10SI | 8S1 | |
| | | | | AT24C128W-10SI | 8S2 | |
| | | | | AT24C128-10CI | 8C | |
| | | | | AT24C128C1-10CI | 8C1 | |
| | | | | AT24C128T1-10TI | 14T | |
| 10 | 1500 | 0.5 | 400 | AT24C128-10PC-2.7 | 8P3 | Commercial (0°C to 70°C) |
| | | | | AT24C128N-10SC-2.7 | 8S1 | |
| | | | | AT24C128W-10SC-2.7 | 8S2 | |
| | | | | AT24C128-10CC-2.7 | 8C | |
| | | | | AT24C128C1-10CC-2.7 | 8C1 | |
| | | | | AT24C128T1-10TC-2.7 | 14T | |
| | 1500 | 0.5 | 400 | AT24C128-10PI-2.7 | 8P3 | Industrial (-40°C to 85°C) |
| | | | | AT24C128N-10SI-2.7 | 8S1 | |
| | | | | AT24C128W-10SI-2.7 | 8S2 | |
| | | | | AT24C128-10CI-2.7 | 8C | |
| | | | | AT24C128C1-10CI-2.7 | 8C1 | |
| | | | | AT24C128T1-10TI-2.7 | 14T | |

| Package Type | |
|---|---|
| 8C | 8-Lead, 0.230" Wide, Leadless Array Package (LAP) |
| 8C1 | 8-Lead, 0.300" Wide, Leadless Array Package (LAP) |
| 8P3 | 8-Lead, 0.300" Wide, Plastic Dual Inline Package (PDIP) |
| 8S1 | 8-Lead, 0.150" Wide, Plastic Gull Wing Small Outline Package (JEDEC SOIC) |
| 8S2 | 8-Lead, 0.200" Wide, Plastic Gull Wing Small Outline Package (EIAJ SOIC) |
| 14T | 14-Lead, 0.170" Wide, Thin Shrink Small Outline Package (TSSOP) |

| Options | |
|---|---|
| Blank | Standard Operation (4.5V to 5.5V) |
| -2.7 | Low-Voltage (2.7V to 5.5V) |
| -1.8 | Low-Voltage (1.8V to 3.6V) |



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## AT24C128 Ordering Information (Continued)

| $t_{WR}$ (max) (ms) | $I_{CC}$ (max) (µA) | $I_{SB}$ (max) (µA) | $f_{MAX}$ (kHz) | Ordering Code | Package | Operation Range |
|---|---|---|---|---|---|---|
| 20 | 800 | 0.2 | 100 | AT24C128-10PC-1.8 | 8P3 | Commercial (0°C to 70°C) |
| | | | | AT24C128N-10SC-1.8 | 8S1 | |
| | | | | AT24C128W-10SC-1.8 | 8S2 | |
| | | | | AT24C128-10CC-1.8 | 8C | |
| | | | | AT24C128C1-10CC-1.8 | 8C1 | |
| | | | | AT24C128T1-10TC-1.8 | 14T | |
| | 800 | 0.2 | 100 | AT24C128-10PI-1.8 | 8P3 | Industrial (-40°C to 85°C) |
| | | | | AT24C128N-10SI-1.8 | 8S1 | |
| | | | | AT24C128W-10SI-1.8 | 8S2 | |
| | | | | AT24C128-10CI-1.8 | 8C | |
| | | | | AT24C128C1-10CI-1.8 | 8C1 | |
| | | | | AT24C128T1-10TI-1.8 | 14T | |

| Package Type | |
|---|---|
| 8C | 8-Lead, 0.230" Wide, Leadless Array Package (LAP) |
| 8C1 | 8-Lead, 0.300" Wide, Leadless Array Package (LAP) |
| 8P3 | 8-Lead, 0.300" Wide, Plastic Dual Inline Package (PDIP) |
| 8S1 | 8-Lead, 0.150" Wide, Plastic Gull Wing Small Outline Package (JEDEC SOIC) |
| 8S2 | 8-Lead, 0.200" Wide, Plastic Gull Wing Small Outline Package (EIAJ SOIC) |
| 14T | 14-Lead, 0.170" Wide, Thin Shrink Small Outline Package (TSSOP) |
| **Options** | |
| Blank | Standard Operation (4.5V to 5.5V) |
| -2.7 | Low-Voltage (2.7V to 5.5V) |
| -1.8 | Low-Voltage (1.8V to 3.6V) |



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## AT24C256 Ordering Information

| $t_{WR}$ (max) (ms) | $I_{CC}$ (max) (µA) | $I_{SB}$ (max) (µA) | $f_{MAX}$ (kHz) | Ordering Code | Package | Operation Range |
|---|---|---|---|---|---|---|
| 10 | 3000 | 6.0 | 1000 | AT24C256-10PC | 8P3 | Commercial (0°C to 70°C) |
| | | | | AT24C256N-10SC | 8S1 | |
| | | | | AT24C256W-10SC | 8S2 | |
| | | | | AT24C256-10CC | 8C | |
| | | | | AT24C256C1-10CC | 8C1 | |
| | | | | AT24C256T1-10TC | 14T | |
| | 3000 | 6.0 | 1000 | AT24C256-10PI | 8P3 | Industrial (-40°C to 85°C) |
| | | | | AT24C256N-10SI | 8S1 | |
| | | | | AT24C256W-10SI | 8S2 | |
| | | | | AT24C256-10CI | 8C | |
| | | | | AT24C256C1-10CI | 8C1 | |
| | | | | AT24C256T1-10TI | 14T | |
| 10 | 1500 | 0.5 | 400 | AT24C256-10PC-2.7 | 8P3 | Commercial (0°C to 70°C) |
| | | | | AT24C256N-10SC-2.7 | 8S1 | |
| | | | | AT24C256W-10SC-2.7 | 8S2 | |
| | | | | AT24C256-10CC-2.7 | 8C | |
| | | | | AT24C256C1-10CC-2.7 | 8C1 | |
| | | | | AT24C256T1-10TC-2.7 | 14T | |
| | 1500 | 0.5 | 400 | AT24C256-10PI-2.7 | 8P3 | Industrial (-40°C to 85°C) |
| | | | | AT24C256N-10SI-2.7 | 8S1 | |
| | | | | AT24C256W-10SI-2.7 | 8S2 | |
| | | | | AT24C256-10CI-2.7 | 8C | |
| | | | | AT24C256C1-10CI-2.7 | 8C1 | |
| | | | | AT24C256T1-10TI-2.7 | 14T | |

| Package Type | |
|---|---|
| 8C | 8-Lead, 0.230" Wide, Leadless Array Package (LAP) |
| 8C1 | 8-Lead, 0.300" Wide, Leadless Array Package (LAP) |
| 8P3 | 8-Lead, 0.300" Wide, Plastic Dual Inline Package (PDIP) |
| 8S1 | 8-Lead, 0.150" Wide, Plastic Gull Wing Small Outline Package (JEDEC SOIC) |
| 8S2 | 8-Lead, 0.200" Wide, Plastic Gull Wing Small Outline Package (EIAJ SOIC) |
| 14T | 14-Lead, 0.170" Wide, Thin Shrink Small Outline Package (TSSOP) |
| **Options** | |
| Blank | Standard Operation (4.5V to 5.5V) |
| -2.7 | Low-Voltage (2.7V to 5.5V) |
| -1.8 | Low-Voltage (1.8V to 3.6V) |



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## AT24C256 Ordering Information (Continued)

| $t_{WR}$ (max) (ms) | $I_{CC}$ (max) (µA) | $I_{SB}$ (max) (µA) | $f_{MAX}$ (kHz) | Ordering Code | Package | Operation Range |
|---|---|---|---|---|---|---|
| 20 | 800 | 0.2 | 100 | AT24C256-10PC-1.8 | 8P3 | Commercial (0°C to 70°C) |
| | | | | AT24C256N-10SC-1.8 | 8S1 | |
| | | | | AT24C256W-10SC-1.8 | 8S2 | |
| | | | | AT24C256-10CC-1.8 | 8C | |
| | | | | AT24C256C1-10CC-1.8 | 8C1 | |
| | | | | AT24C256T1-10TC-1.8 | 14T | |
| | 800 | 0.2 | 100 | AT24C256-10PI-1.8 | 8P3 | Industrial (-40°C to 85°C) |
| | | | | AT24C256N-10SI-1.8 | 8S1 | |
| | | | | AT24C256W-10SI-1.8 | 8S2 | |
| | | | | AT24C256-10CI-1.8 | 8C | |
| | | | | AT24C256C1-10CI-1.8 | 8C1 | |
| | | | | AT24C256T1-10TI-1.8 | 14T | |

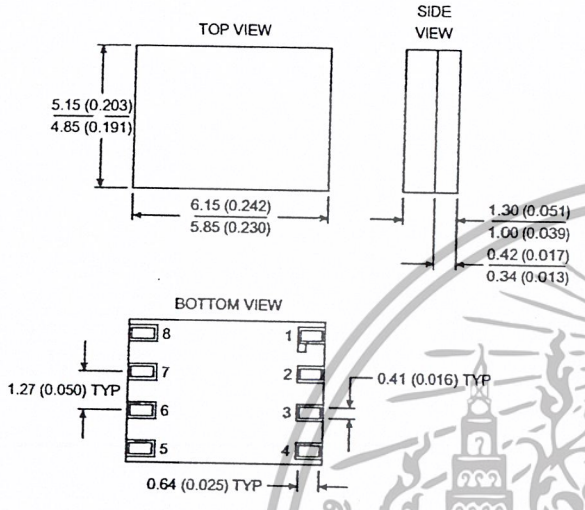
| Package Type | |
|---|---|
| 8C | 8-Lead, 0.230" Wide, Leadless Array Package (LAP) |
| 8C1 | 8-Lead, 0.300" Wide, Leadless Array Package (LAP) |
| 8P3 | 8-Lead, 0.300" Wide, Plastic Dual Inline Package (PDIP) |
| 8S1 | 8-Lead, 0.150" Wide, Plastic Gull Wing Small Outline Package (JEDEC SOIC) |
| 8S2 | 8-Lead, 0.200" Wide, Plastic Gull Wing Small Outline Package (EIAJ SOIC) |
| 14T | 14-Lead, 0.170" Wide, Thin Shrink Small Outline Package (TSSOP) |
| **Options** | |
| Blank | Standard Operation (4.5V to 5.5V) |
| -2.7 | Low-Voltage (2.7V to 5.5V) |
| -1.8 | Low-Voltage (1.8V to 3.6V) |



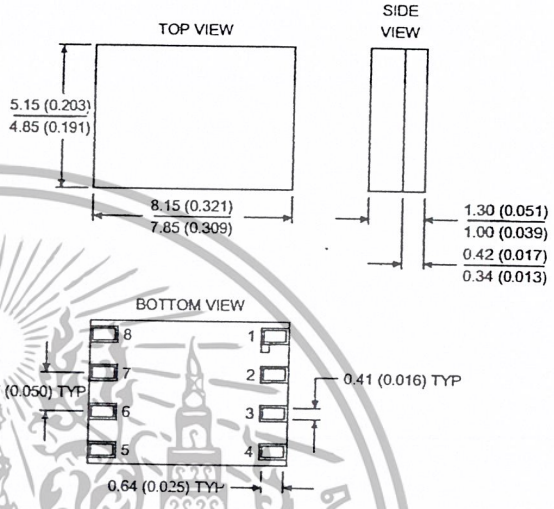
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## Packaging Information

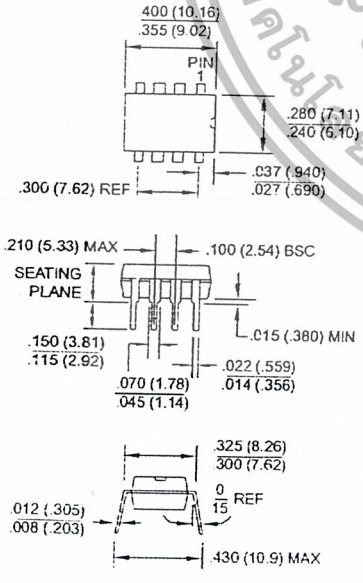
**8C**, 8-Lead, 0.230" Wide, Leadless Array Package (LAP)
Dimensions in Inches and (Millimeters)



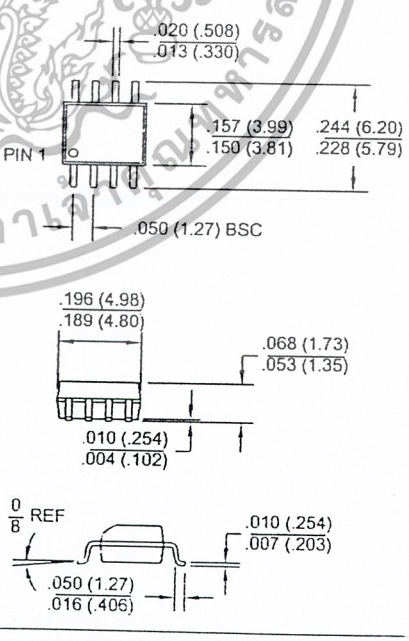

**8C1**, 8-Lead, 0.300" Wide, Leadless Array Package (LAP)
Dimensions in Inches and (Millimeters)




**8P3**, 8-Lead, 0.300" Wide,
Plastic Dual Inline Package (PDIP)
Dimensions in Inches and (Millimeters)
JEDEC STANDARD MS-001 BA




**8S1**, 8-Lead, 0.150" Wide,
Plastic Gull Wing Small Outline (JEDEC SOIC)
Dimensions in Inches and (Millimeters)

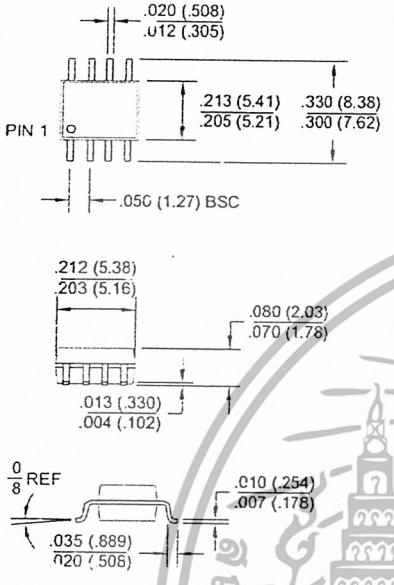




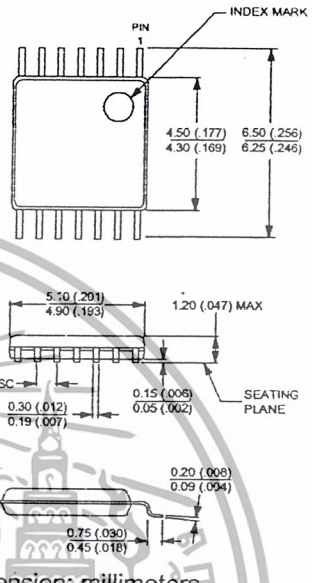

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## Packaging Information

**8S2,** 8-Lead, 0.200" Wide,
Plastic Gull Wing Small Outline (EIAJ SOIC)
Dimensions in Inches and (Millimeters)




**14T,** 14-Lead, 0.170" Wide, Thin Shrink Small
Outline Package (TSSOP)
Dimensions in Inches and (Millimeters)




*Controlling dimension: millimeters



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

DALLAS SEMICONDUCTOR

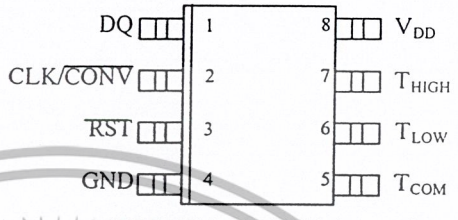
# DS1620
# Digital Thermometer and Thermostat

www.dalsemi.com

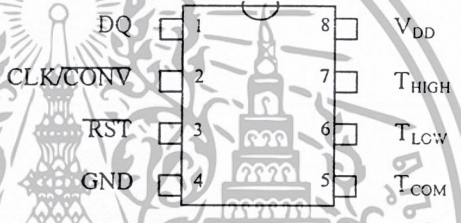
## FEATURES

- Requires no external components
- Supply voltage range covers from 2.7V to 5.5V
- Measures temperatures from -55°C to +125°C in 0.5°C increments; Fahrenheit equivalent is -67°F to +257°F in 0.9°F increments
- Temperature is read as a 9-bit value
- Converts temperature to digital word in 1 second (max)
- Thermostatic settings are user-definable and nonvolatile
- Data is read from/written via a 3-wire serial interface (CLK, DQ, $\overline{RST}$)
- Applications include thermostatic controls, industrial systems, consumer products, thermometers, or any thermally sensitive system
- 8-pin DIP or SOIC (208-mil) packages

## PIN ASSIGNMENT




DS1620S 8-Pin SOIC (208-miil)
See Mech Drawings Section




DS1620 8-Pin DIP (300-mil)
See Mech Drawings Section

## PIN DESCRIPTION

| | |
|---|---|
| DQ | - 3-Wire Input/Output |
| CLK/$\overline{CONV}$ | - 3-Wire Clock Input and Stand-alone Convert Input |
| $\overline{RST}$ | - 3-Wire Reset Input |
| GND | - Ground |
| $T_{HIGH}$ | - High Temperature Trigger |
| $T_{LOW}$ | - Low Temperature Trigger |
| $T_{COM}$ | - High/Low Combination Trigger |
| $V_{DD}$ | - Power Supply Voltage (3V - 5V) |

## DESCRIPTION

The DS1620 Digital Thermometer and Thermostat provides 9–bit temperature readings which indicate the temperature of the device. With three thermal alarm outputs, the DS1620 can also act as a thermostat. $T_{HIGH}$ is driven high if the DS1620's temperature is greater than or equal to a user–defined temperature TH. $T_{LOW}$ is driven high if the DS1620's temperature is less than or equal to a user–defined temperature TL. $T_{COM}$ is driven high when the temperature exceeds TH and stays high until the temperature falls below that of TL.



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

User–defined temperature settings are stored in nonvolatile memory, so parts can be programmed prior to insertion in a system, as well as used in standalone applications without a CPU. Temperature settings and temperature readings are all communicated to/from the DS1620 over a simple 3–wire interface.

## OPERATION-MEASURING TEMPERATURE

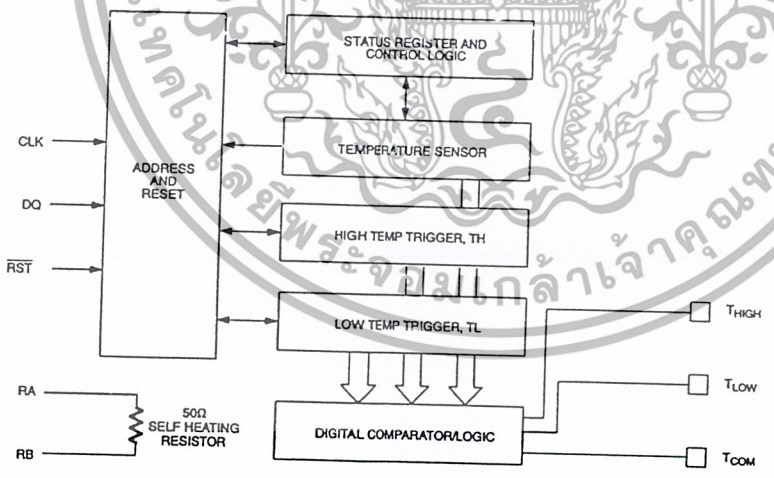
A block diagram of the DS1620 is shown in Figure 1. The DS1620 measures temperatures through the use of an onboard proprietary temperature measurement technique. A block diagram of the temperature measurement circuitry is shown in Figure 2.

The DS1620 measures temperature by counting the number of clock cycles that an oscillator with a low temperature coefficient goes through during a gate period determined by a high temperature coefficient oscillator. The counter is preset with a base count that corresponds to –55°C. If the counter reaches 0 before the gate period is over, the temperature register, which is also preset to the –55°C value, is incremented, indicating that the temperature is higher than –55°C.

At the same time, the counter is then preset with a value determined by the slope accumulator circuitry. This circuitry is needed to compensate for the parabolic behavior of the oscillators over temperature. The counter is then clocked again until it reaches 0. If the gate period is still not finished, then this process repeats.

The slope accumulator is used to compensate for the nonlinear behavior of the oscillators over temperature, yielding a high-resolution temperature measurement. This is done by changing the number of counts necessary for the counter to go through for each incremental degree in temperature. To obtain the desired resolution, therefore, both the value of the counter and the number of counts per degree C (the value of the slope accumulator) at a given temperature must be known.

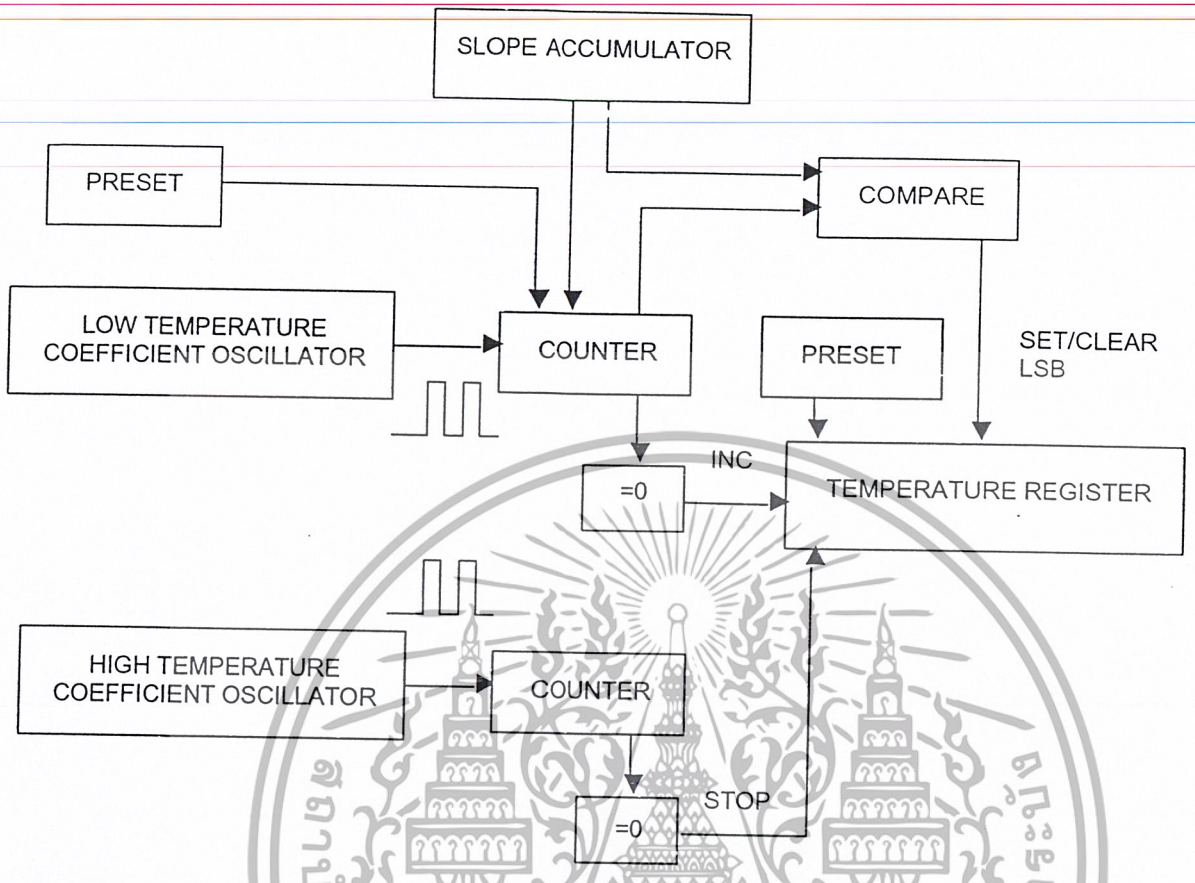
## DS1620 FUNCTIONAL BLOCK DIAGRAM Figure 1






เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## TEMPERATURE MEASURING CIRCUITRY Figure 2




This calculation is done inside the DS1620 to provide 0.5°C resolution. The temperature reading is provided in a 9–bit, two's complement reading by issuing a READ TEMPERATURE command. Table 1 describes the exact relationship of output data to measured temperature. The data is transmitted serially through the 3–wire serial interface, LSB first. The DS1620 can measure temperature over the range of -55°C to +125°C in 0.5°C increments. For Fahrenheit usage, a lookup table or conversion factor must be used.

## TEMPERATURE/DATA RELATIONSHIPS Table 1

| TEMP | DIGITAL OUTPUT (Binary) | DIGITAL OUTPUT (Hex) |
|---|---|---|
| +125°C | 0 11111010 | 00FA |
| +25°C | 0 00110010 | 0032h |
| +½°C | 0 00000001 | 0001h |
| +0°C | 0 00000000 | 0000h |
| -½°C | 1 11111111 | 01FFh |
| -25°C | 1 11001110 | 01CEh |
| -55°C | 1 10010010 | 0192h |

Since data is transmitted over the 3–wire bus LSB first, temperature data can be written to/read from the DS1620 as either a 9–bit word (taking $\overline{RST}$ low after the 9$^{th}$ (MSB) bit), or as two transfers of 8–bit



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

words, with the most significant 7 bits being ignored or set to 0, as illustrated in Table 1. After the MSB, the DS1620 will output 0s.

Note that temperature is represented in the DS1620 in terms of a ½°C LSB, yielding the following 9–bit format:

| MSB | | | | | | | | LSB | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| X | X | X | X | X | X | X | 1 | 1 | 1 | 0 | 0 | 1 | 1 | 1 | 0 |

T = -25°C

Higher resolutions may be obtained by reading the temperature, and truncating the 0.5°C bit (the LSB) from the read value. This value is TEMP_READ. The value left in the counter may then be read by issuing a READ COUNTER command. This value is the count remaining (COUNT_REMAIN) after the gate period has ceased. By loading the value of the slope accumulator into the count register (using the READ SLOPE command), this value may then be read, yielding the number of counts per degree C (COUNT_PER_C) at that temperature. The actual temperature may be then be calculated by the user using the following:

$$\text{TEMPERATURE} = \text{TEMP_READ} - 0.25 + \frac{(\text{COUNT_PER_C} - \text{COUNT_REMAIN})}{\text{COUNT_PER_C}}$$

## DETAILED PIN DESCRIPTION Table 2

| PIN | SYMBOL | DESCRIPTION |
|---|---|---|
| 1 | DQ | **Data Input/Output pin** for 3-wire communication port. |
| 2 | CLK/$\overline{\text{CONV}}$ | **Clock input pin** for 3-wire communication port. When the DS1620 is used in a stand-alone application with no 3–wire port, this pin can be used as a convert pin. Temperature conversion will begin on the falling edge of $\overline{\text{CONV}}$. |
| 3 | $\overline{\text{RST}}$ | **Reset input pin** for 3-wire communication port. |
| 4 | GND | **Ground pin.** |
| 5 | $T_{COM}$ | **High/Low Combination Trigger.** Goes high when temperature exceeds TH; will reset to low when temperature falls below TL. |
| 6 | $T_{LOW}$ | **Low Temperature Trigger.** Goes high when temperature falls below TL. |
| 7 | $T_{HIGH}$ | **High Temperature Trigger.** Goes high when temperature exceeds TH. |
| 8 | $V_{DD}$ | **Supply Voltage.** 2.7V – 5.5V input power pin. |

## OPERATION–THERMOSTAT CONTROLS

Three thermally triggered outputs, $T_{HIGH}$, $T_{LOW}$, and $T_{COM}$, are provided to allow the DS1620 to be used as a thermostat, as shown in Figure 3. When the DS1620's temperature meets or exceeds the value stored in the high temperature trip register, the output $T_{HIGH}$ becomes active (high) and remains active until the DS1620's measured temperature becomes less than the stored value in the high temperature register, TH. The $T_{HIGH}$ output can be used to indicate that a high temperature tolerance boundary has been met or exceeded, or it can be used as part of a closed loop system to activate a cooling system and deactivate it when the system temperature returns to tolerance.

The $T_{LOW}$ output functions similarly to the $T_{HIGH}$ output. When the DS1620's measured temperature equals or falls below the value stored in the low temperature register, the $T_{LOW}$ output becomes active. $T_{LOW}$ remains active until the DS1620's temperature becomes greater than the value stored in the low temperature register, TL. The $T_{LOW}$ output can be used to indicate that a low temperature tolerance

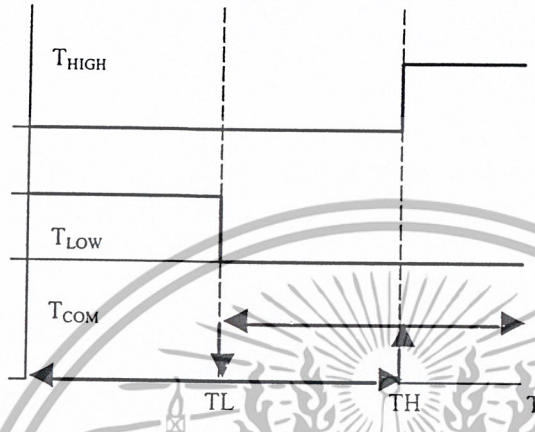


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

boundary has been met or exceeded, or as part of a closed loop system it can be used to activate a heating system and deactivate it when the system temperature returns to tolerance.

The $T_{COM}$ output goes high when the measured temperature meets or exceeds TH, and will stay high until the temperature equals or falls below TL. In this way, any amount of hysteresis can be obtained.

## THERMOSTAT OUTPUT OPERATION Figure 3




## OPERATION AND CONTROL

The DS1620 must have temperature settings resident in the TH and TL registers for thermostatic operation. A configuration/status register also determines the method of operation that the DS1620 will use in a particular application and indicates the status of the temperature conversion operation. The configuration register is defined as follows:

CONFIGURATION/STATUS REGISTER

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| DONE | THF | TLF | NVB | 1 | 0 | CPU | 1SHOT |

where

DONE = Conversion Done Bit. 1=conversion complete, 0=conversion in progress.

THF = Temperature High Flag. This bit will be set to 1 when the temperature is greater than or equal to the value of TH. It will remain 1 until reset by writing 0 into this location or by removing power from the device. This feature provides a method of determining if the DS1620 has ever been subjected to temperatures above TH while power has been applied.

TLF = Temperature Low Flag. This bit will be set to 1 when the temperature is less than or equal to the value of TL. It will remain 1 until reset by writing 0 into this location or by removing power from the device. This feature provides a method of determining if the DS1620 has ever been subjected to temperatures below TL while power has been applied.

NVB = Nonvolatile Memory Busy Flag. 1=write to an $E^2$ memory cell in progress. 0=nonvolatile memory is not busy. A copy to $E^2$ may take up to 10 ms.

CPU = CPU Use Bit. If CPU=0, the CLK/$\overline{CONV}$ pin acts as a conversion start control, when $\overline{RST}$ is low. If CPU is 1, the DS1620 will be used with a CPU communicating to it over the 3–wire port, and the



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

operation of the CLK/$\overline{\text{CONV}}$ pin is as a normal clock in concert with DQ and $\overline{\text{RST}}$. This bit is stored in nonvolatile $E^2$ memory, capable of at least 50,000 writes. The DS1620 is shipped with CPU=0.

1SHOT = One–Shot Mode. If 1SHOT is 1, the DS1620 will perform one temperature conversion upon reception of the Start Convert T protocol. If 1SHOT is 0, the DS1620 will continuously perform temperature conversion. This bit is stored in nonvolatile $E^2$ memory, capable of at least 50,000 writes. The DS1620 is shipped with 1SHOT=0.

For typical thermostat operation, the DS1620 will operate in continuous mode. However, for applications where only one reading is needed at certain times or to conserve power, the one–shot mode may be used. Note that the thermostat outputs ($T_{HIGH}$, $T_{LOW}$, $T_{COM}$) will remain in the state they were in after the last valid temperature conversion cycle when operating in one–shot mode.

## OPERATION IN STAND–ALONE MODE

In applications where the DS1620 is used as a simple thermostat, no CPU is required. Since the temperature limits are nonvolatile, the DS1620 can be programmed prior to insertion in the system. In order to facilitate operation without a CPU, the CLK/$\overline{\text{CONV}}$ pin (pin 2) can be used to initiate conversions. Note that the CPU bit must be set to 0 in the configuration register to use this mode of operation. Whether CPU=0 or 1, the 3–wire port is active. Setting CPU=1 disables the stand–alone mode.

To use the CLK/$\overline{\text{CONV}}$ pin to initiate conversions, $\overline{\text{RST}}$ must be low and CLK/$\overline{\text{CONV}}$ must be high. If CLK/$\overline{\text{CONV}}$ is driven low and then brought high in less than 10 ms, one temperature conversion will be performed and then the DS1620 will return to an idle state. If CLK/$\overline{\text{CONV}}$ is driven low and remains low, continuous conversions will take place until CLK/$\overline{\text{CONV}}$ is brought high again. With the CPU bit set to 0, the CLK/$\overline{\text{CONV}}$ will override the 1SHOT bit if it is equal to 1. This means that even if the part is set for one–shot mode, driving CLK/$\overline{\text{CONV}}$ low will initiate conversions.

## 3–WIRE COMMUNICATIONS

The 3–wire bus is comprised of three signals. These are the $\overline{\text{RST}}$ (reset) signal, the CLK (clock) signal, and the DQ (data) signal. All data transfers are initiated by driving the $\overline{\text{RST}}$ input high. Driving the $\overline{\text{RST}}$ input low terminates communication. (See Figures 4 and 5.) A clock cycle is a sequence of a falling edge followed by a rising edge. For data inputs, the data must be valid during the rising edge of a clock cycle. Data bits are output on the falling edge of the clock and remain valid through the rising edge.

When reading data from the DS1620, the DQ pin goes to a high-impedance state while the clock is high. Taking $\overline{\text{RST}}$ low will terminate any communication and cause the DQ pin to go to a high-impedance state.

Data over the 3–wire interface is communicated LSB first. The command set for the 3–wire interface as shown in Table 3 is as follows.

### Read Temperature [AAh]

This command reads the contents of the register which contains the last temperature conversion result. The next nine clock cycles will output the contents of this register.



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### Write TH [01h]

This command writes to the TH (HIGH TEMPERATURE) register. After issuing this command the next nine clock cycles clock in the 9–bit temperature limit which will set the threshold for operation of the $T_{HIGH}$ output.

### Write TL [02h]

This command writes to the TL (LOW TEMPERATURE) register. After issuing this command the next nine clock cycles clock in the 9–bit temperature limit which will set the threshold for operation of the $T_{LOW}$ output.

### Read TH [A1h]

This command reads the value of the TH (HIGH TEMPERATURE) register. After issuing this command the next nine clock cycles clock out the 9–bit temperature limit which sets the threshold for operation of the $T_{HIGH}$ output.

### Read TL [A2h]

This command reads the value of the TL (LOW TEMPERATURE) register. After issuing this command the next nine clock cycles clock out the 9–bit temperature limit which sets the threshold for operation of the $T_{LOW}$ output.

### Read Counter [A0h]

This command reads the value of the counter byte. The next nine clock cycles will output the contents of this register.

### Read Slope [A9h]

This command reads the value of the slope counter byte from the DS1620. The next nine clock cycles will output the contents of this register.

### Start Convert T [EEh]

This command begins a temperature conversion. No further data is required. In one–shot mode the temperature conversion will be performed and then the DS1620 will remain idle. In continuous mode this command will initiate continuous conversions.

### Stop Convert T [22h]

This command stops temperature conversion. No further data is required. This command may be used to halt a DS1620 in continuous conversion mode. After issuing this command the current temperature measurement will be completed and then the DS1620 will remain idle until a Start Convert T is issued to resume continuous operation.

### Write Config [0Ch]

This command writes to the configuration register. After issuing this command the next eight clock cycles clock in the value of the configuration register.

### Read Config [ACh]

This command reads the value in the configuration register. After issuing this command the next eight clock cycles output the value of the configuration register.



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## DS1620 COMMAND SET Table 3

| INSTRUCTION | DESCRIPTION | PROTOCOL | 3-WIRE BUS DATA AFTER ISSUING PROTOCOL | NOTES |
|---|---|---|---|---|
| **TEMPERATURE CONVERSION COMMANDS** | | | | |
| Read Temperature | Reads last converted temperature value from temperature register. | AAh | <read data> | |
| Read Counter | Reads value of count remaining from counter. | A0h | <read data> | |
| Read Slope | Reads value of the slope accumulator. | A9h | <read data> | |
| Start Convert T | Initiates temperature conversion. | EEh | Idle | 1 |
| Stop Convert T | Halts temperature conversion. | 22h | Idle | 1 |
| **THERMOSTAT COMMANDS** | | | | |
| Write TH | Writes high temperature limit value into TH register. | 01h | <write data> | 2 |
| Write TL | Writes low temperature limit value into TL register. | 02h | <write data> | 2 |
| Read TH | Reads stored value of high temperature limit from TH register. | A1h | <read data> | 2 |
| Read TL | Reads stored value of low temperature limit from TL register. | A2h | <read data> | 2 |
| Write Config | Writes configuration data to configuration register. | 0Ch | <write data> | 2 |
| Read Config | Reads configuration data from configuration register. | ACh | <read data> | 2 |

**NOTES:**

1. In continuous conversion mode, a Stop Convert T command will halt continuous conversion. To restart, the Start Convert T command must be issued. In one-shot mode, a Start Convert T command must be issued for every temperature reading desired.
2. Writing to the $E^2$ typically requires 10 ms at room temperature. After issuing a write command no further writes should be requested for at least 10 ms.



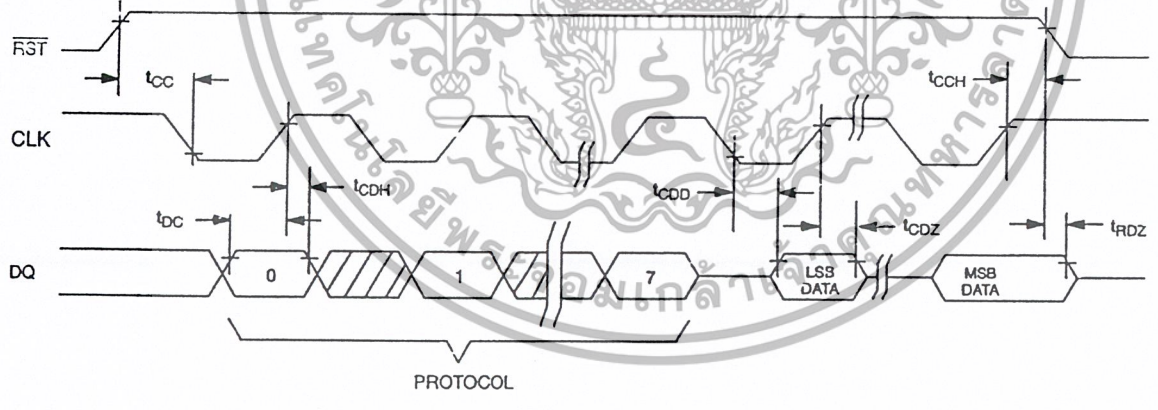
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## FUNCTION EXAMPLE

Example: CPU sets up DS1620 for continuous conversion and thermostatic function.

| CPU MODE | DS1620 MODE (3-WIRE) | DATA (LSB FIRST) | COMMENTS |
|---|---|---|---|
| TX | RX | 0Ch | CPU issues Write Config command |
| TX | RX | 00h | CPU sets DS1620 up for continuous conversion |
| TX | RX | Toggle $\overline{RST}$ | CPU issues Reset to DS1620 |
| TX | RX | 01h | CPU issues Write TH command |
| TX | RX | 0050h | CPU sends data for TH limit of +40°C |
| TX | RX | Toggle $\overline{RST}$ | CPU issues Reset to DS1620 |
| TX | RX | 02h | CPU issues Write TL command |
| TX | RX | 0014h | CPU sends data for TL limit of +10°C |
| TX | RX | Toggle $\overline{RST}$ | CPU issues Reset to DS1620 |
| TX | RX | A1h | CPU issues Read TH command |
| RX | TX | 0050h | DS1620 sends back stored value of TH for CPU to verify |
| TX | RX | Toggle $\overline{RST}$ | CPU issues Reset to DS1620 |
| TX | RX | A2h | CPU issues Read TL command |
| RX | TX | 0014h | DS1620 sends back stored value of TL for CPU to verify |
| TX | RX | Toggle $\overline{RST}$ | CPU issues Reset to DS1620 |
| TX | RX | EEh | CPU issues Start Convert T command |
| TX | RX | Drop $\overline{RST}$ | CPU issues Reset to DS1620 |

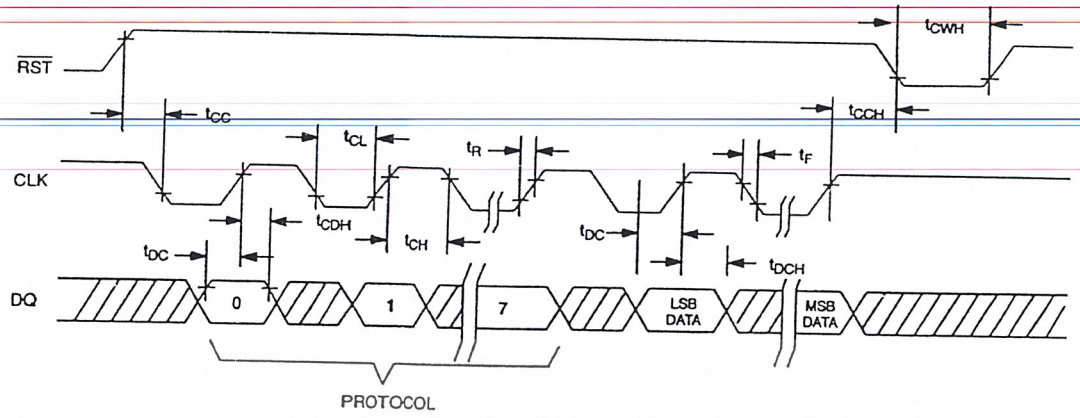
## READ DATA TRANSFER Figure 4






เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## WRITE DATA TRANSFER Figure 5




NOTE: $t_{CL}$, $t_{CH}$, $t_R$, and $t_F$ apply to both read and write data transfer.

## RELATED APPLICATION NOTES

The following Application Notes can be applied to the DS1620. These notes can be obtained from the Dallas Semiconductor "Application Note Book", via our Web site at http:\\www.dalsemi.com/, or through our faxback service at (972) 371–4441.

Application Note 67: "Applying and Using the DS1620 in Temperature Control Applications"

Application Note 85: "Interfacing the DS1620 to the Motorola SPI Bus"

Application Note 105: "High Resolution Temperature Measurement with Dallas Direct–to–Digital Temperature Sensors"

Sample DS1620 subroutines that can be used in conjunction with AN105 can be downloaded from the web-site or our Anonymous FTP Site.

## ABSOLUTE MAXIMUM RATINGS*

| | |
|---|---|
| Voltage on Any Pin Relative to Ground | –0.5V to +7.0V |
| Operating Temperature | –55°C to +125°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

| PARAMETER | SYMBOL | MIN | TYP | MAX | UNITS | NOTES |
|---|---|---|---|---|---|---|
| Supply | $V_{DD}$ | 2.7 | | 5.5 | V | 1 |
| Logic 1 | $V_{IH}$ | 2.0 | | $V_{CC}$+0.3 | V | 1 |
| Logic 0 | $V_{IL}$ | -0.3 | | +0.6 | V | 1 |

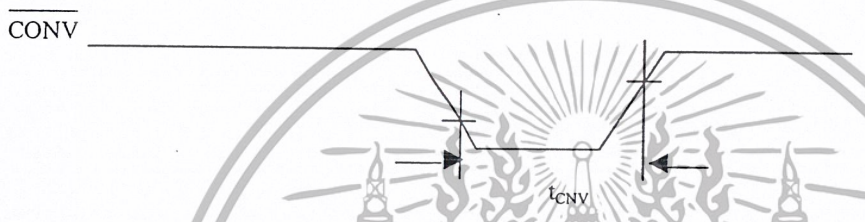


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## DC ELECTRICAL CHARACTERISTICS (-55°C to +125°C; $V_{DD}$=2.7V to 5.5V)

| PARAMETER | SYMBOL | CONDITION | MIN | MAX | UNITS | NOTES |
|---|---|---|---|---|---|---|
| Thermometer Error | $T_{ERR}$ | 0°C to +70°C<br>-55°C to +0°C and 70°C to 125°C | See Typical Curve | | ±½ °C | 10, 11 |
| Logic 0 Output | $V_{OL}$ | | | 0.4 | V | 3 |
| Logic 1 Output | $V_{OH}$ | | 2.4 | | V | 2 |
| Input Resistance | $R_I$ | $\overline{RST}$ to GND<br>DQ, CLK to $V_{DD}$ | 1<br>1 | | MΩ<br>MΩ | |
| Active Supply Current | $I_{CC}$ | 0°C to +70°C | | 1 | mA | 4, 5 |
| Standby Supply Current | $I_{STBY}$ | 0°C to +70°C | | 1 | µA | 4, 5 |

## SINGLE CONVERT TIMING DIAGRAM (STAND-ALONE MODE)




## AC ELECTRICAL CHARACTERISTICS (-55°C to +125°C; $V_{DD}$=2.7V to 5.5V)

| PARAMETERS | SYMBOL | MIN | TYP | MAX | UNITS | NOTES |
|---|---|---|---|---|---|---|
| Temperature Conversion Time | $T_{TC}$ | | 400 | 1000 | ms | |
| Data to CLK Setup | $t_{DC}$ | 35 | | | ns | 6 |
| CLK to Data Hold | $t_{CDH}$ | 40 | | | ns | 6 |
| CLK to Data Delay | $t_{CDD}$ | | | 100 | ns | 6, 7, 8 |
| CLK Low Time | $t_{CL}$ | 285 | | | ns | 6 |
| CLK High Time | $t_{CH}$ | 285 | | | ns | 6 |
| CLK Frequency | $f_{CLK}$ | DC | | 1.75 | MHz | 6 |
| CLK Rise and Fall | $t_R$, $t_F$ | | | 500 | ns | |
| $\overline{RST}$ to CLK Setup | $t_{CC}$ | 100 | | | ns | 6 |
| CLK to $\overline{RST}$ Hold | $t_{CCH}$ | 40 | | | ns | 6 |
| $\overline{RST}$ Inactive Time | $t_{CWH}$ | 125 | | | ns | 6, 9 |
| CLK High to I/O High-Z | $t_{CDZ}$ | | | 50 | ns | 6 |
| $\overline{RST}$ Low to I/O High-Z | $t_{RDZ}$ | | | 50 | ns | 6 |
| Convert Pulse Width | $t_{CNV}$ | 250 ns | | 500 ms | | |
| NV Write Cycle Time | $t_{WR}$ | | 10 | 50 | ms | 12 |

## AC ELECTRICAL CHARACTERISTICS (-55°C to +125°C; $V_{DD}$=2.7V to 5.5V)

| PARAMETER | SYMBOL | MIN | TYP | MAX | UNITS | NOTES |
|---|---|---|---|---|---|---|
| Input Capacitance | $C_I$ | | 5 | | pF | |
| I/O Capacitance | $C_{I/O}$ | | 10 | | pF | |



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## NOTES:

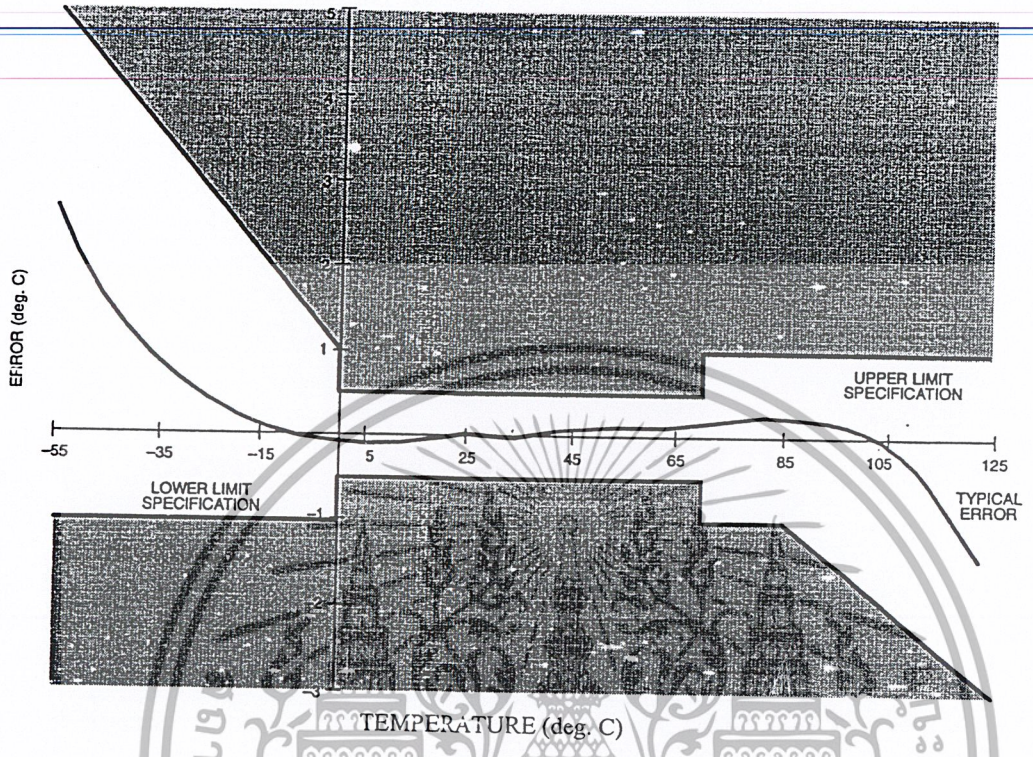
1. All voltages are referenced to ground.
2. Logic 1 voltages are specified at a source current of 1 mA.
3. Logic 0 voltages are specified at a sink current of 4 mA.
4. $I_{CC}$ specified with DQ pin open and CLK pin at $V_{DD}$.
5. $I_{CC}$ specified with $V_{CC}$ at 5.0V and $\overline{RST}$ =GND.
6. Measured at $V_{IH}$ = 2.0V or $V_{IL}$ = 0.8V.
7. Measured at $V_{OH}$ = 2.4V or $V_{OL}$ = 0.4V.
8. Load capacitance = 50 pF.
9. $t_{CWH}$ must be 10 ms minimum following any write command that involves the $E^2$ memory.
10. See typical curve for specification limits outside 0°C to 70°C range.
11. Thermometer error reflects temperature accuracy as tested during calibration.
12. Writing to the nonvolatile memory should only take place in the 0°C to 70°C temperature range.
13. Valid for design revisions D1 and above. The supply range for Rev. C2 and below is 4.5V ≤ 5.5V.



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## TYPICAL PERFORMANCE CURVE

DS1620 DIGITAL THERMOMETER AND THERMOSTAT
TEMPERATURE READING ERROR






เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

LINEAR TECHNOLOGY

# LTC1286/LTC1298

## Micropower Sampling 12-Bit A/D Converters In SO-8 Packages

## FEATURES

- **12-Bit Resolution**
- **8-Pin SOIC Plastic Package**
- **Low Cost**
- **Low Supply Current: 250µA Typ.**
- Auto Shutdown to 1nA Typ.
- Guaranteed ±3/4LSB Max DNL
- Single Supply 5V to 9V Operation
- On-Chip Sample-and-Hold
- 60µs Conversion Time
- Sampling Rates:
  - 12.5 ksps (LTC1286)
  - 11.1 ksps (LTC1298)
- I/O Compatible with SPI, Microwire, etc.
- Differential Inputs (LTC1286)
- 2-Channel MUX (LTC1298)
- 3V Versions Available: LTC1285/LTC1288

## APPLICATIONS

- Battery-Operated Systems
- Remote Data Acquisition
- Battery Monitoring
- Handheld Terminal Interface
- Temperature Measurement
- Isolated Data Acquisition

## DESCRIPTION

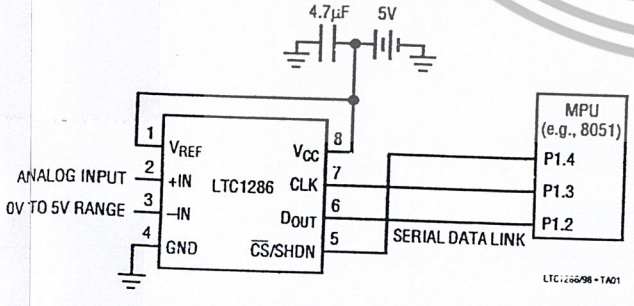
The LTC1286/LTC1298 are micropower, 12-bit, successive approximation sampling A/D converters. They typically draw only 250µA of supply current when converting and automatically power down to a typical supply current of 1nA whenever they are not performing conversions. They are packaged in 8-pin SO packages and operate on 5V to 9V supplies. These 12-bit, switched-capacitor, successive approximation ADCs include sample-and-holds. The LTC1286 has a single differential analog input. The LTC1298 offers a software selectable 2-channel MUX.

On-chip serial ports allow efficient data transfer to a wide range of microprocessors and microcontrollers over three wires. This, coupled with micropower consumption, makes remote location possible and facilitates transmitting data through isolation barriers.

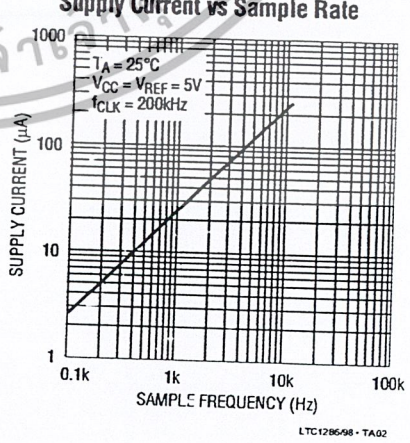
These circuits can be used in ratiometric applications or with an external reference. The high impedance analog inputs and the ability to operate with reduced spans (to 1.5V full scale) allow direct connection to sensors and transducers in many applications, eliminating the need for gain stages.

## TYPICAL APPLICATIONS

**25µW, SO-8 Package, 12-Bit ADC Samples at 200Hz and Runs Off a 5V Supply**




**Supply Current vs Sample Rate**






เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## ABSOLUTE MAXIMUM RATINGS (Notes 1 and 2)

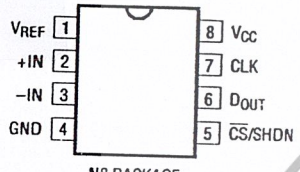
Supply Voltage ($V_{CC}$) to GND ..... 12V
Voltage
- Analog and Reference ..... −0.3V to $V_{CC}$ + 0.3V
- Digital Inputs ..... −0.3V to 12V
- Digital Output ..... −0.3V to $V_{CC}$ + 0.3V

Power Dissipation ..... 500mW
Operating Temperature Range
- LTC1286C/LTC1298C ..... 0°C to 70°C
- LTC1286I/LTC1298I ..... −40°C to 85°C

Storage Temperature Range ..... −65°C to 150°C
Lead Temperature (Soldering, 10 sec.) ..... 300°C

## PACKAGE/ORDER INFORMATION

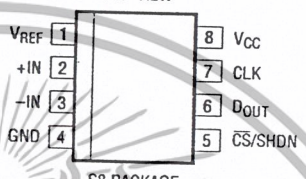
TOP VIEW




N8 PACKAGE
8-LEAD PLASTIC DIP
$T_{JMAX}$ = 150°C, $\theta_{JA}$ = 130°C/W

| ORDER PART NUMBER |
|---|
| LTC1286CN8 |
| LTC1286IN8 |

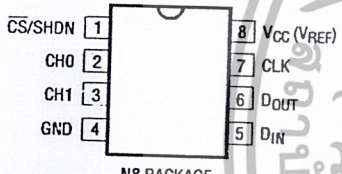
TOP VIEW




S8 PACKAGE
8-LEAD PLASTIC SOIC
$T_{JMAX}$ = 150°C, $\theta_{JA}$ = 175°C/W

| ORDER PART NUMBER | PART MARKING |
|---|---|
| LTC1286CS8<br>LTC1286IS8 | 1286C<br>1286I |

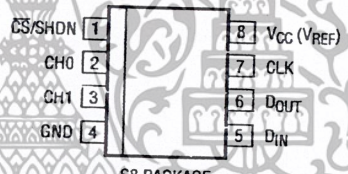
TOP VIEW




N8 PACKAGE
8-LEAD PLASTIC DIP
$T_{JMAX}$ = 150°C, $\theta_{JA}$ = 130°C/W

| ORDER PART NUMBER |
|---|
| LTC1298CN8 |
| LTC1298IN8 |

TOP VIEW




S8 PACKAGE
8-LEAD PLASTIC SOIC
$T_{JMAX}$ = 150°C, $\theta_{JA}$ = 175°C/W

| ORDER PART NUMBER | PART MARKING |
|---|---|
| LTC1298CS8<br>LTC1298IS8 | 1298C<br>1298I |

Consult factory for military grade parts.

## RECOMMENDED OPERATING CONDITIONS

| SYMBOL | PARAMETER | CONDITIONS | MIN | TYP | MAX | UNITS |
|---|---|---|---|---|---|---|
| $V_{CC}$ | Supply Voltage (Note 3) | LTC1286<br>LTC1298 | 4.5<br>4.5 | | 9.0<br>5.5 | V<br>V |
| $f_{CLK}$ | Clock Frequency | $V_{CC}$ = 5V | (Note 4) | | 200 | kHz |
| $t_{CYC}$ | Total Cycle Time | LTC1286, $f_{CLK}$ = 200kHz<br>LTC1298, $f_{CLK}$ = 200kHz | 80<br>90 | | | µs<br>µs |
| $t_{hDI}$ | Hold Time, $D_{IN}$ After CLK↑ | $V_{CC}$ = 5V | 150 | | | ns |
| $t_{suCS}$ | Setup Time $\overline{CS}$↓ Before First CLK↑ (See Operating Sequence) | LTC1286, $V_{CC}$ = 5V<br>LTC1298, $V_{CC}$ = 5V | 2<br>2 | | | µs<br>µs |
| $t_{suDI}$ | Setup Time, $D_{IN}$ Stable Before CLK↑ | $V_{CC}$ = 5V | 400 | | | ns |
| $t_{WHCLK}$ | CLK High Time | $V_{CC}$ = 5V | 2 | | | µs |
| $t_{WLCLK}$ | CLK Low Time | $V_{CC}$ = 5V | 2 | | | µs |
| $t_{WHCS}$ | $\overline{CS}$ High Time Between Data Transfer Cycles | $V_{CC}$ = 5V | 2 | | | µs |
| $t_{WLCS}$ | $\overline{CS}$ Low Time During Data Transfer | LTC1286, $f_{CLK}$ = 200kHz<br>LTC1298, $f_{CLK}$ = 200kHz | 75<br>85 | | | µs<br>µs |



LINEAR TECHNOLOGY

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## CONVERTER AND MULTIPLEXER CHARACTERISTICS (Note 5)

| PARAMETER | CONDITIONS | | LTC1286 MIN | LTC1286 TYP | LTC1286 MAX | LTC1298 MIN | LTC1298 TYP | LTC1298 MAX | UNITS |
|---|---|---|---|---|---|---|---|---|---|
| Resolution (No Missing Codes) | | • | 12 | | | 12 | | | Bits |
| Integral Linearity Error | (Note 6) | • | | ±3/4 | ±2 | | ±3/4 | ±2 | LSB |
| Differential Linearity Error | | • | | ±1/4 | ±3/4 | | ±1/4 | ±3/4 | LSB |
| Offset Error | | • | | 3/4 | ±3 | | 3/4 | ±3 | LSB |
| Gain Error | | • | | ±2 | ±8 | | ±2 | ±8 | LSB |
| Analog Input Range | (Note 7 and 8) | • | | −0.05V to $V_{CC}$ + 0.05V | | | | | V |
| REF Input Range (LTC1286) (Notes 7, 8, and 9) | $4.5 \le V_{CC} \le 5.5V$ | | | 1.5V to $V_{CC}$ + 0.05V | | | | | V |
| | $5.5V < V_{CC} \le 9V$ | | | 1.5V to 5.55V | | | | | V |
| Analog Input Leakage Current (Note 10) | | • | | ±1 | | | ±1 | | µA |

## DIGITAL AND DC ELECTRICAL CHARACTERISTICS (Note 5)

| SYMBOL | PARAMETER | CONDITIONS | | MIN | TYP | MAX | UNITS |
|---|---|---|---|---|---|---|---|
| $V_{IH}$ | High Level Input Voltage | $V_{CC}$ = 5.25V | • | 2 | | | V |
| $V_{IL}$ | Low Level Input Voltage | $V_{CC}$ = 4.75V | • | | | 0.8 | V |
| $I_{IH}$ | High Level Input Current | $V_{IN} = V_{CC}$ | • | | | 2.5 | µA |
| $I_{IL}$ | Low Level Input Current | $V_{IN}$ = 0V | • | | | −2.5 | µA |
| $V_{OH}$ | High Level Output Voltage | $V_{CC}$ = 4.75V, $I_O$ = 10µA | • | 4.0 | 4.64 | | V |
| | | $V_{CC}$ = 4.75V, $I_O$ = 360µA | • | 2.4 | 4.62 | | V |
| $V_{OL}$ | Low Level Output Voltage | $V_{CC}$ = 4.75V, $I_O$ = 1.6mA | • | | | 0.4 | V |
| $I_{OZ}$ | Hi-Z Output Leakage | $\overline{CS}$ = High | • | | | ±3 | µA |
| $I_{SOURCE}$ | Output Source Current | $V_{OUT}$ = 0V | | | −25 | | mA |
| $I_{SINK}$ | Output Sink Current | $V_{OUT} = V_{CC}$ | | | 45 | | mA |
| $R_{REF}$ | Reference Input Resistance (LTC1286) | $\overline{CS} = V_{CC}$ | | | 5000 | | MΩ |
| | | $\overline{CS}$ = GND | | | 55 | | kΩ |
| $I_{REF}$ | Reference Current (LTC1286) | $\overline{CS} = V_{CC}$ | | | 0.001 | 2.5 | µA |
| | | $t_{CYC} \ge 640µs$, $f_{CLK} \le 25kHz$ | • | | 90 | 140 | µA |
| | | $t_{CYC}$ = 80µs, $f_{CLK}$ = 200kHz | • | | 90 | 140 | µA |
| $I_{CC}$ | Supply Current | $\overline{CS} = V_{CC}$ | • | | 0.001 | ±3.0 | µA |
| | | LTC1286, $t_{CYC} \ge 640µs$, $f_{CLK} \le 25kHz$ | • | | 200 | 400 | µA |
| | | LTC1286, $t_{CYC}$ = 80µs, $f_{CLK}$ = 200kHz | • | | 250 | 500 | µA |
| | | LTC1298, $t_{CYC} \ge 720µs$, $f_{CLK} \le 25kHz$ | • | | 290 | 490 | µA |
| | | LTC1298, $t_{CYC}$ = 90µs, $f_{CLK}$ = 200kHz | • | | 340 | 640 | µA |

## DYNAMIC ACCURACY $f_{SMPL}$ = 12.5kHz (LTC1286), $f_{SMPL}$ = 11.1kHz (LTC1298) (Note 5)

| SYMBOL | PARAMETER | CONDITIONS | MIN | TYP | MAX | UNITS |
|---|---|---|---|---|---|---|
| S/(N +D) | Signal-to-Noise Plus Distortion Ratio | 1kHz/7kHz Input Signal | | 71/68 | | dB |
| THD | Total Harmonic Distortion (Up to 5th Harmonic) | 1kHz/7kHz Input Signal | | −84/−80 | | dB |
| SFDR | Spurious-Free Dynamic Range | 1kHz/7kHz Input Signal | | 90/86 | | dB |
| | Peak Harmonic or Spurious Noise | 1kHz/7kHz Input Signal | | −90/−86 | | dB |



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## AC CHARACTERISTICS (Note 5)

| SYMBOL | PARAMETER | CONDITIONS | | MIN | TYP | MAX | UNITS |
|---|---|---|---|---|---|---|---|
| $t_{SMPL}$ | Analog Input Sample Time | See Operating Sequence | | | 1.5 | | CLK Cycles |
| $f_{SMPL(MAX)}$ | Maximum Sampling Frequency | LTC1286 | ● | 12.5 | | | kHz |
| | | LTC1298 | ● | 11.1 | | | kHz |
| $t_{CONV}$ | Conversion Time | See Operating Sequence | | | 12 | | CLK Cycles |
| $t_{dDO}$ | Delay Time, CLK↓ to $D_{OUT}$ Data Valid | See Test Circuits | ● | | 250 | 600 | ns |
| $t_{dis}$ | Delay Time, $\overline{CS}$↑ to $D_{OUT}$ Hi-Z | See Test Circuits | ● | | 135 | 300 | ns |
| $t_{en}$ | Delay Time, CLK↓ to $D_{OUT}$ Enable | See Test Circuits | ● | | 75 | 200 | ns |
| $t_{hDO}$ | Time Output Data Remains Valid After CLK↓ | $C_{LOAD}$ = 100pF | | | 230 | | ns |
| $t_f$ | $D_{OUT}$ Fall Time | See Test Circuits | ● | | 20 | 75 | ns |
| $t_r$ | $D_{OUT}$ Rise Time | See Test Circuits | ● | | 20 | 75 | ns |
| $C_{IN}$ | Input Capacitance | Analog Inputs, On Channel | | | 20 | | pF |
| | | Analog Inputs, Off Channel | | | 5 | | pF |
| | | Digital Input | | | 5 | | pF |

The ● denotes specifications which apply over the full operating temperature range.

**Note 1:** Absolute maximum ratings are those values beyond which the life of a device may be impaired.

**Note 2:** All voltage values are with respect to GND.

**Note 3:** These devices are specified at 5V. For 3V specified devices, see LTC1285 and LTC1288.

**Note 4:** Increased leakage currents at elevated temperatures cause the S/H to droop, therefore it is recommended that $f_{CLK} \geq 120kHz$ at 85°C, $f_{CLK} \geq 75kHz$ at 70° and $f_{CLK} \geq 1kHz$ at 25°C.

**Note 5:** $V_{CC} = 5V$, $V_{REF} = 5V$ and CLK = 200kHz unless otherwise specified.

**Note 6:** Linearity error is specified between the actual end points of the A/D transfer curve.

**Note 7:** Two on-chip diodes are tied to each reference and analog input which will conduct for reference or analog input voltages one diode drop below GND or one diode drop above Vcc. This spec allows 50mV forward bias of either diode for $4.5V \leq Vcc \leq 5.5V$. This means that as long as the reference or analog input does not exceed the supply voltage by more than 50mV the output code will be correct. To achieve an absolute 0V to 5V input voltage range will therefore require a minimum supply voltage of 4.950V over initial tolerance, temperature variations and loading. For $5.5V < Vcc \leq 9V$, reference and analog input range cannot exceed 5.55V. If reference and analog input range are greater than 5.55V, the output code will not be guaranteed to be correct.

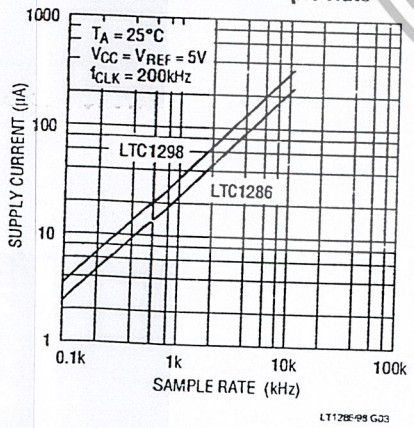
**Note 8:** The supply voltage range for the LTC1286 is from 4.5V to 9V, but the supply voltage range for the LTC1298 is only from 4.5V to 5.5V.

**Note 9:** Recommended operating conditions

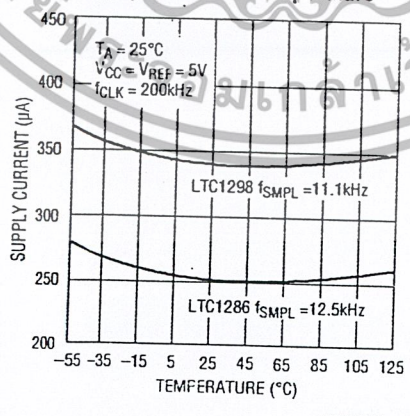
**Note 10:** Channel leakage current is measured after the channel selection.

## TYPICAL PERFORMANCE CHARACTERISTICS

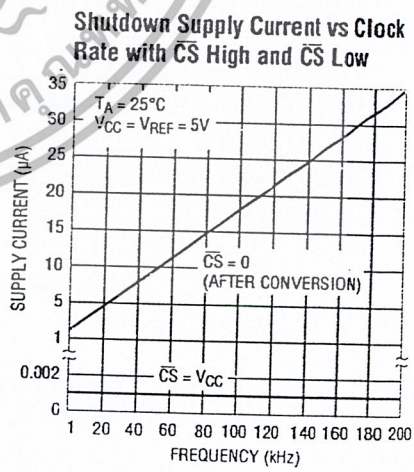
**Supply Current vs Sample Rate**




**Supply Current vs Temperature**




**Shutdown Supply Current vs Clock Rate with $\overline{CS}$ High and $\overline{CS}$ Low**



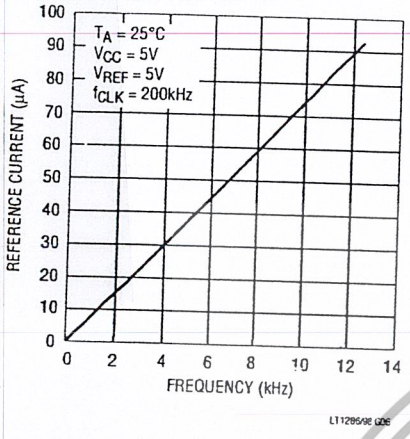



LINEAR TECHNOLOGY

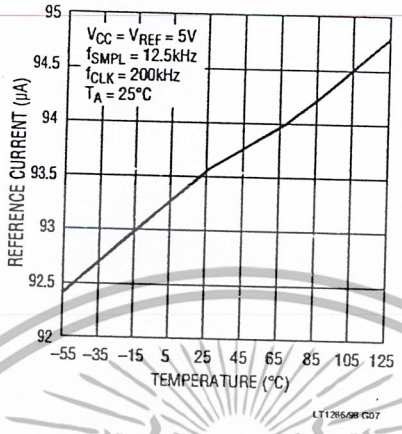
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# TYPICAL PERFORMANCE CHARACTERISTICS

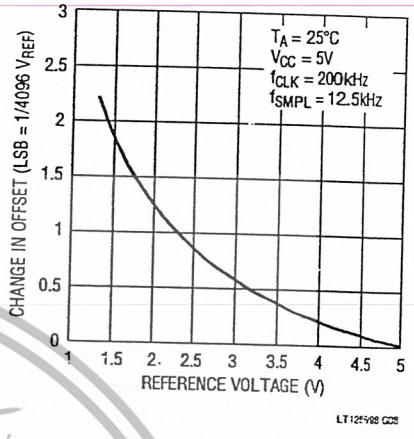
**Reference Current vs Sample Rate (LTC1286)**



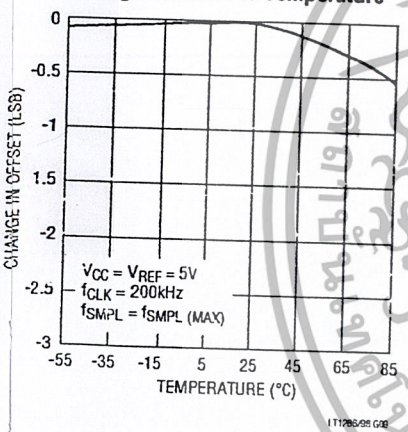

**Reference Current vs Temperature**



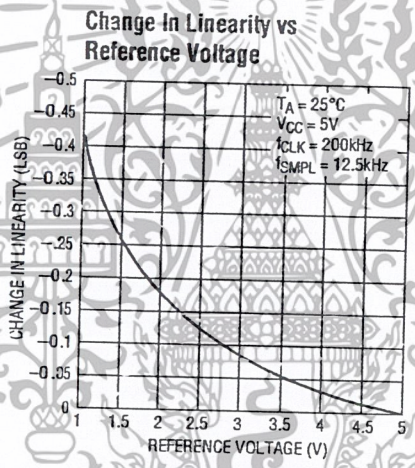

**Change in Offset vs Reference Voltage**



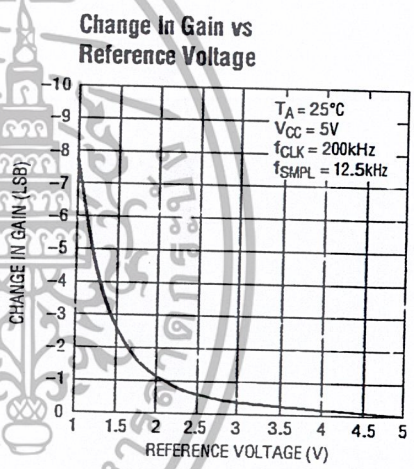

**Change in Offset vs Temperature**



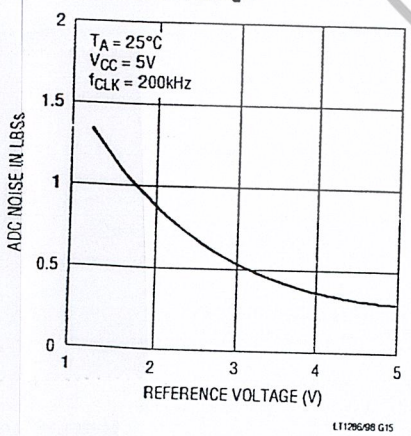

**Change in Linearity vs Reference Voltage**



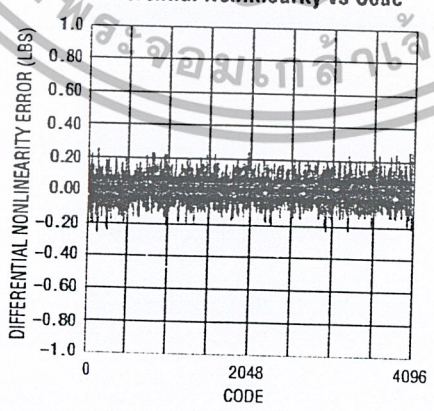

**Change in Gain vs Reference Voltage**



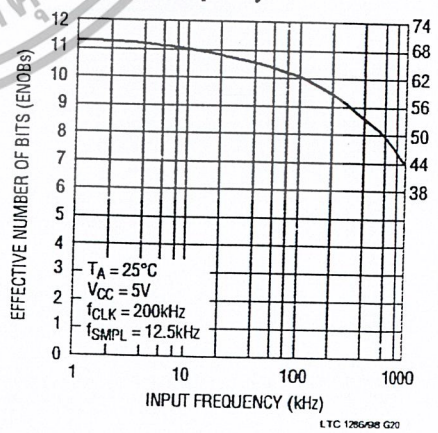

**Peak-to-Peak ADC Noise vs Reference Voltage**




**Differential Nonlinearity vs Code**




**Effective Bits and S/(N + D) vs Input Frequency**

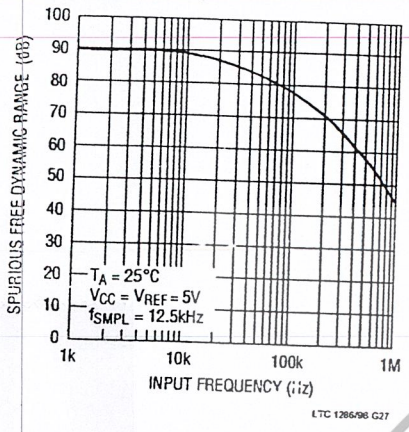




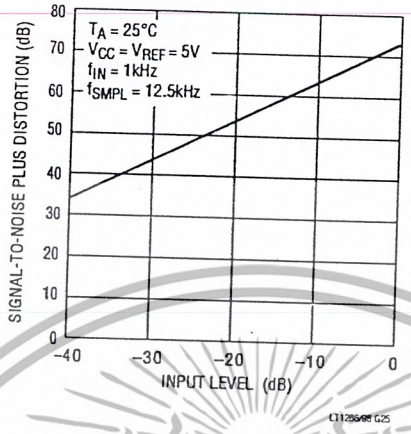

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## TYPICAL PERFORMANCE CHARACTERISTICS

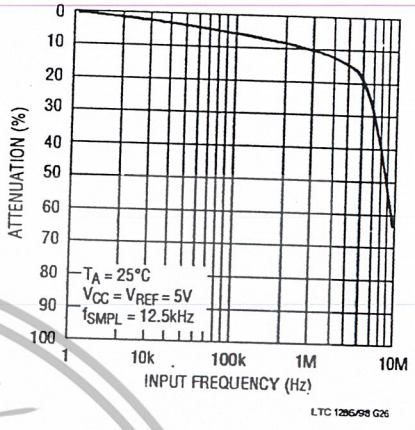
### Spurious Free Dynamic Range vs Frequency



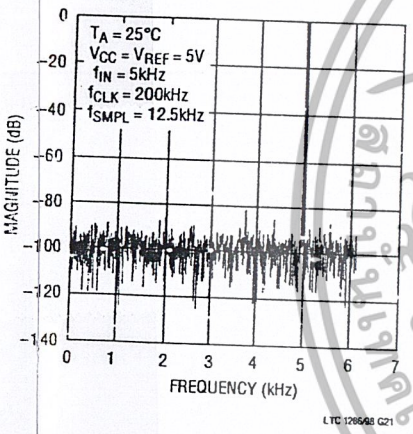

### S/(N+D) vs Input Level



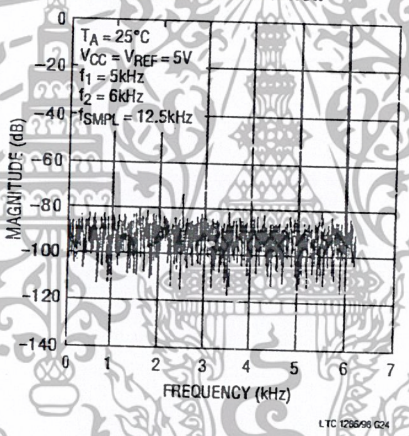

### Attenuation vs Input Frequency



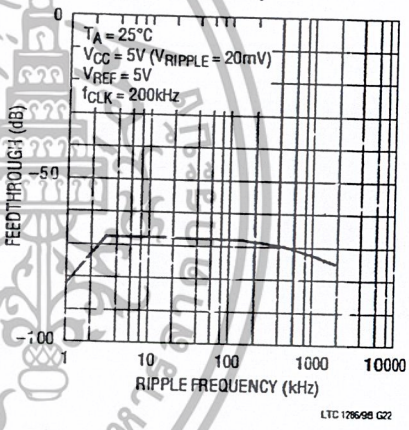

### 4096 Point FFT Plot



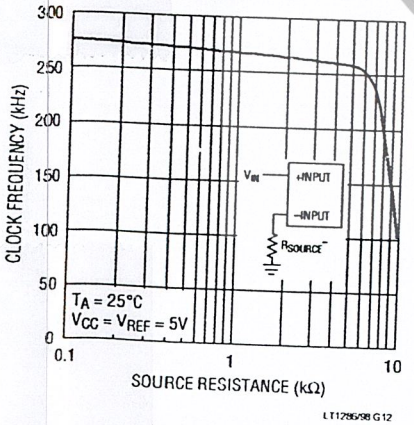

### Intermodulation Distortion



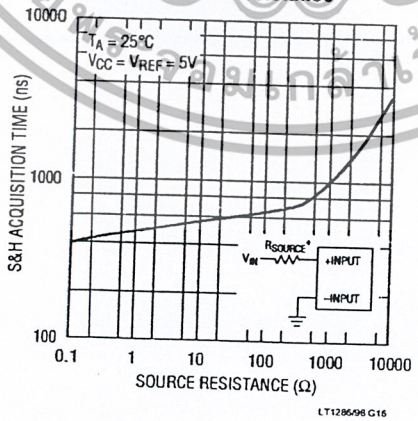

### Power Supply Feedthrough vs Ripple Frequency



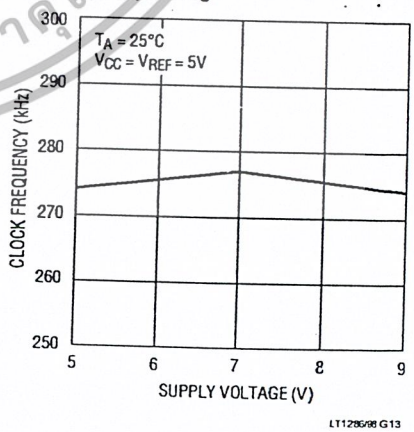

### Maximum Clock Frequency vs Source Resistance




### Sample and Hold Aquisition Time vs Source Resistance




### Maximum Clock Frequency vs Supply Voltage

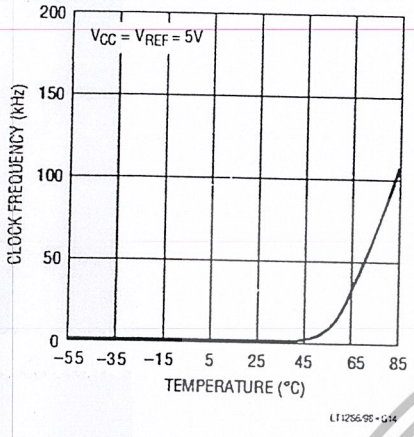




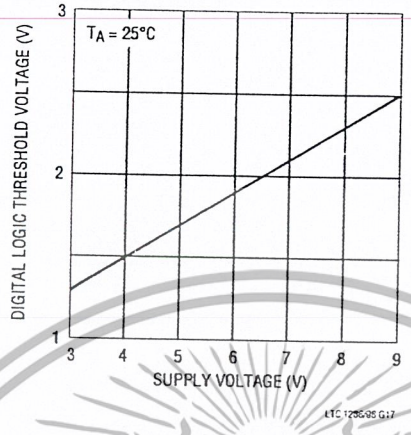

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## TYPICAL PERFORMANCE CHARACTERISTICS

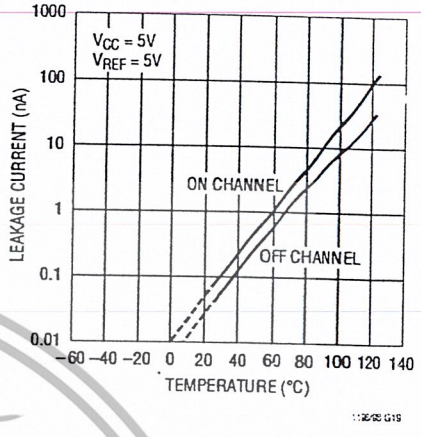
**Minimum Clock Frequency for 0.1 LSB Error vs Temperature**




**Digital Input Logic Threshold vs Supply Voltage**




**Input Channel Leakage Current vs Temperature**




## PIN FUNCTIONS

### LTC1286

**$V_{REF}$ (Pin 1):** Reference Input. The reference input defines the span of the A/D converter.

**$IN^+$ (Pin 2):** Positive Analog Input.

**$IN^-$ (Pin 3):** Negative Analog Input.

**GND (Pin 4):** Analog Ground. GND should be tied directly to an analog ground plane.

**$\overline{CS}$/SHDN (Pin 5):** Chip Select Input. A logic low on this input enables the LTC1286. A logic high on this input disables and powers down the LTC1286.

**$D_{OUT}$ (Pin 6):** Digital Data Output. The A/D conversion result is shifted out of this output.

**CLK (Pin 7):** Shift Clock. This clock synchronizes the serial data transfer and determines conversion speed.

**$V_{CC}$ (Pin 8):** Power Supply Voltage. This pin provides power to the A/D converter. It must be kept free of noise and ripple by bypassing directly to the analog ground plane.

### LTC1298

**$\overline{CS}$/SHDN (Pin 1):** Chip Select Input. A logic low on this input enables the LTC1298. A logic high on this input disables and powers down the LTC1298.

**CH0 (Pin 2):** Analog Input.

**CH1 (Pin 3):** Analog Input.

**GND (Pin 4):** Analog Ground. GND should be tied directly to an analog ground plane.

**$D_{IN}$ (Pin 5):** Digital Data Input. The multiplexer address is shifted into this input.

**$D_{OUT}$ (Pin 6):** Digital Data Output. The A/D conversion result is shifted out of this output.

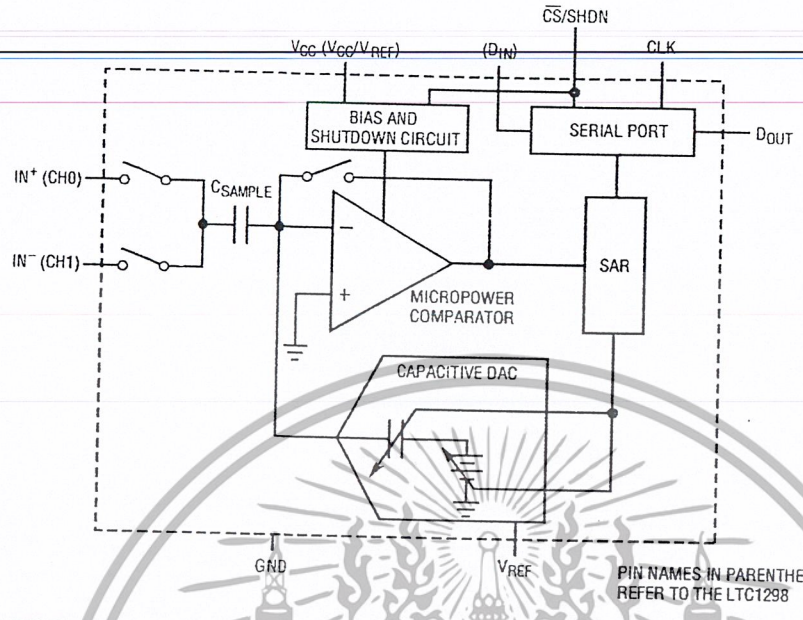
**CLK (Pin 7):** Shift Clock. This clock synchronizes the serial data transfer and determines conversion speed.

**$V_{CC}/V_{REF}$ (Pin 8):** Power Supply and Reference Voltage. This pin provides power and defines the span of the A/D converter. It must be kept free of noise and ripple by bypassing directly to the analog ground plane.



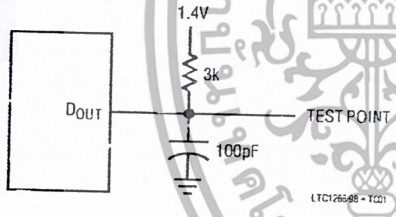
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## BLOCK DIAGRAM

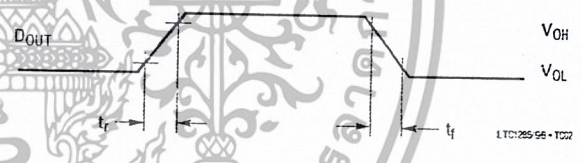



## TEST CIRCUITS

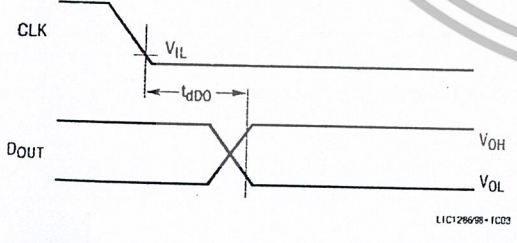
**Load Circuit for $t_{DDO}$, $t_r$ and $t_f$**



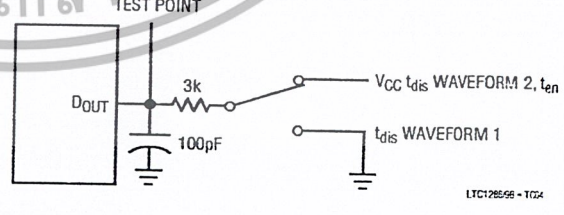

**Voltage Waveforms for $D_{OUT}$ Rise and Fall Times, $t_r$, $t_f$**




**Voltage Waveforms for $D_{OUT}$ Delay Times, $t_{DDO}$**




**Load Circuit for $t_{dis}$ and $t_{en}$**

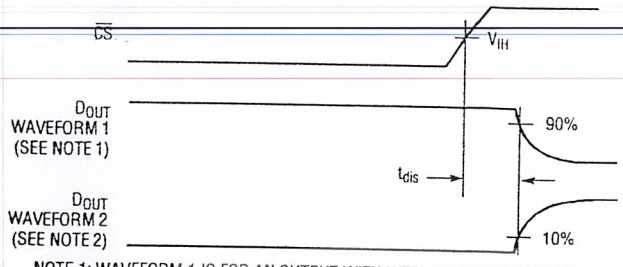





เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## TEST CIRCUITS

**Voltage Waveforms for $t_{dis}$**

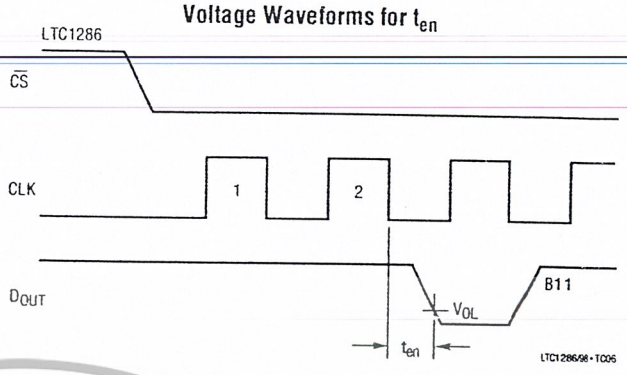



NOTE 1: WAVEFORM 1 IS FOR AN OUTPUT WITH INTERNAL CONDITIONS SUCH THAT THE OUTPUT IS HIGH UNLESS DISABLED BY THE OUTPUT CONTROL.

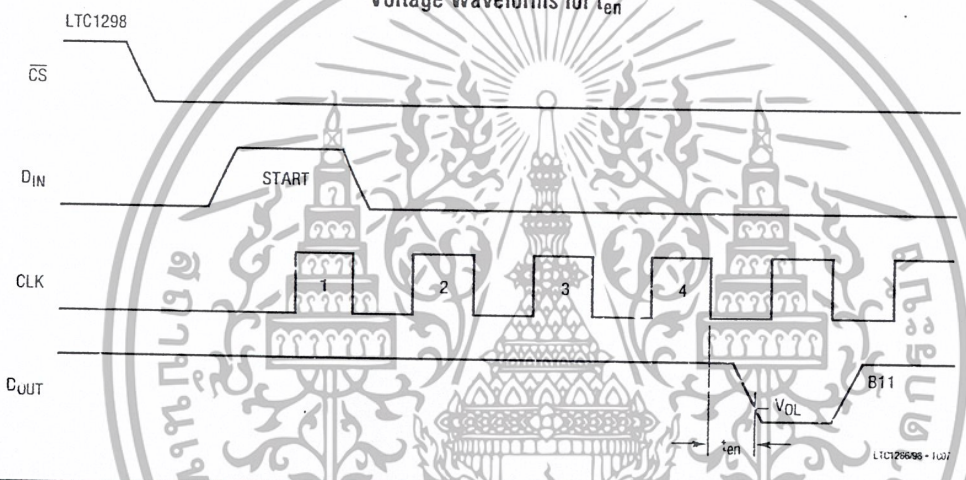
NOTE 2: WAVEFORM 2 IS FOR AN OUTPUT WITH INTERNAL CONDITIONS SUCH THAT THE OUTPUT IS LOW UNLESS DISABLED BY THE OUTPUT CONTROL.

LTC1286/98 • TC05

**Voltage Waveforms for $t_{en}$**




**Voltage Waveforms for $t_{en}$**




## APPLICATION INFORMATION

### OVERVIEW

The LTC1286 and LTC1298 are micropower, 12-bit, successive approximation sampling A/D converters. The LTC1286 typically draws 250μA of supply current when sampling at 12.5kHz while the LTC1298 nominally consumes 350μA of supply current when sampling at 11.1 kHz. The extra 100μA of supply current on the LTC1298 comes from the reference input which is intentionally tied to the supply. Supply current drops linearly as the sample rate is reduced (see Supply Current vs Sample Rate). The ADCs automatically power down when not performing conversions, drawing only leakage current. They are packaged in 8-pin SO and DIP packages. The LTC1286 operates on a single supply from 4.5V to 9V, while the LTC1298 operates from a 4.5V to 5.5V supply.

Both the LTC1286 and the LTC1298 contain a 12-bit, switched-capacitor ADC, a sample-and-hold, and a serial port (see Block Diagram). Although they share the same basic design, the LTC1286 and LTC1298 differ in some respects. The LTC1286 has a differential input and has an external reference input pin. It can measure signals floating on a DC common-mode voltage and can operate with reduced spans to 1V. Reducing the spans allows it to achieve 244μV resolution. The LTC1298 has a two-channel input multiplexer and can convert either channel with respect to ground or the difference between the two. The reference input is tied to the supply pin.



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# APPLICATION INFORMATION

## SERIAL INTERFACE

The 2-channel LTC1298 communicates with microprocessors and other external circuitry via a synchronous, half duplex, 4-wire serial interface. The single channel LTC1286 uses a 3-wire interface (see Operating Sequence in Figures 1 and 2).

### Data Transfer

The CLK synchronizes the data transfer with each bit being transmitted on the falling CLK edge and captured on the rising CLK edge in both transmitting and receiving systems.

The LTC1286 does not require a configuration input word and has no $D_{IN}$ pin. A falling $\overline{CS}$ initiates data transfer as shown in the LTC1286 operating sequence. After $\overline{CS}$ falls the second CLK pulse enables $D_{OUT}$. After one null bit the A/D conversion result is output on the $D_{OUT}$ line. Bringing $\overline{CS}$ high resets the LTC1286 for the next data exchange.

The LTC1298 first receives input data and then transmits back the A/D conversion result (half duplex). Because of the half duplex operation, $D_{IN}$ and $D_{OUT}$ may be tied together allowing transmission over just 3 wires: $\overline{CS}$, CLK and DATA ($D_{IN}/D_{OUT}$).

Data transfer is initiated by a falling chip select ($\overline{CS}$) signal. After $\overline{CS}$ falls the LTC1298 looks for a start bit. After the start bit is received, the 3-bit input word is shifted into the $D_{IN}$ input which configures the LTC1298 and starts the conversion. After one null bit, the result of the conversion is output on the $D_{OUT}$ line. At the end of the data exchange $\overline{CS}$ should be brought high. This resets the LTC1298 in preparation for the next data exchange.

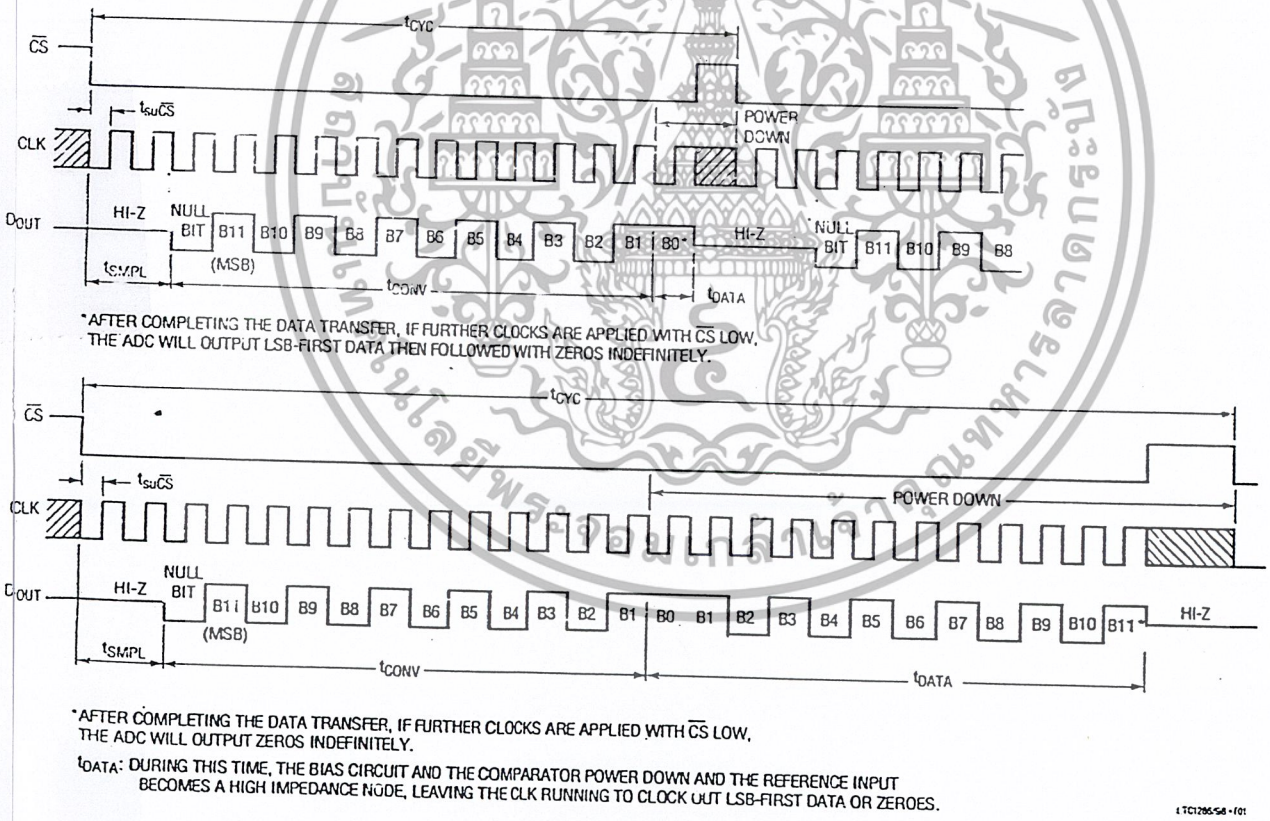
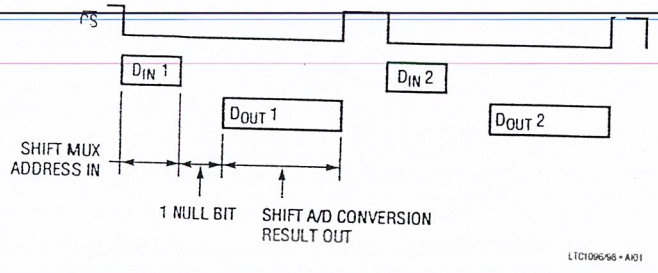



**Figure 1. LTC1286 Operating Sequence**

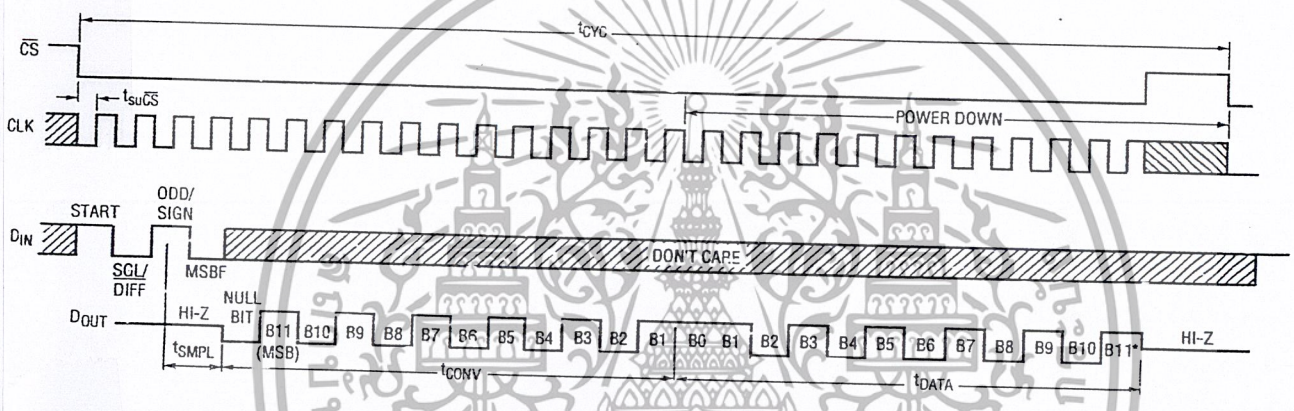


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

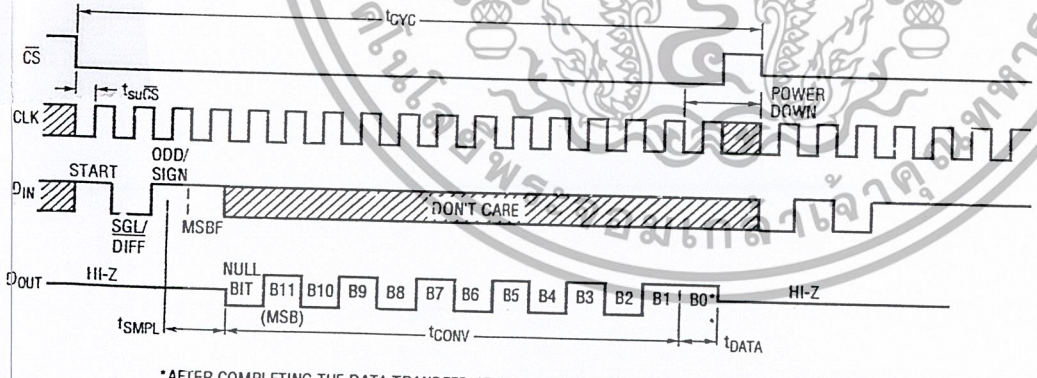
## APPLICATION INFORMATION




**MSB-First Data (MSBF = 0)**




**MSB-First Data (MSBF = 1)**




*AFTER COMPLETING THE DATA TRANSFER, IF FURTHER CLOCKS ARE APPLIED WITH $\overline{CS}$ LOW, THE ADC WILL OUTPUT ZEROS INDEFINITELY.

$t_{DATA}$: DURING THIS TIME, THE BIAS CIRCUIT AND THE COMPARATOR POWER DOWN AND THE REFERENCE INPUT BECOMES A HIGH IMPEDANCE NODE, LEAVING THE CLK RUNNING TO CLOCK OUT LSB-FIRST DATA OR ZEROES.

LTC1286/98 • H02

**Figure 2. LTC1298 Operating Sequence Example: Differential Inputs ($CH^+$, $CH^-$)**

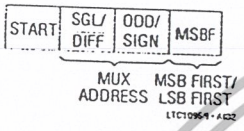


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# APPLICATION INFORMATION

## Input Data Word

The LTC1286 requires no $D_{IN}$ word. It is permanently configured to have a single differential input. The conversion result appears on the $D_{OUT}$ line. The data format is MSB first followed by the LSB sequence. This provides easy interface to MSB or LSB first serial ports. For MSB first data the $\overline{CS}$ signal can be taken high after B0 (see Figure 1). The LTC1298 clocks data into the $D_{IN}$ input on the rising edge of the clock. The input data words are defined as follows:




## Start Bit

The first "logical one" clocked into the $D_{IN}$ input after $\overline{CS}$ goes low is the start bit. The start bit initiates the data transfer. The LTC1298 will ignore all leading zeros which precede this logical one. After the start bit is received, the remaining bits of the input word will be clocked in. Further inputs on the $D_{IN}$ pin are then ignored until the next $\overline{CS}$ cycle.

## Multiplexer (MUX) Address

The bits of the input word following the START bit assign the MUX configuration for the requested conversion. For a given channel selection, the converter will measure the voltage between the two channels indicated by the + and – signs in the selected row of the following tables. In single-ended mode, all input channels are measured with respect to GND.

LTC1298 Channel Selection

| | MUX ADDRESS SGL/DIFF | MUX ADDRESS ODD/SIGN | CHANNEL # 0 | CHANNEL # 1 | GND |
|---|---|---|---|---|---|
| SINGLE-ENDED MUX MODE | 1 | 0 | + | | – |
| | 1 | 1 | | + | – |
| DIFFERENTIAL MUX MODE | 0 | 0 | + | – | |
| | 0 | 1 | – | + | |

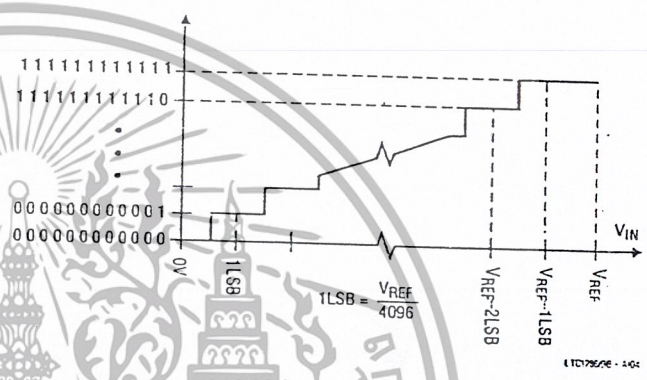
## MSB First/LSB First (MSBF)

The output data of the LTC1298 is programmed for MSB first or LSB first sequence using the MSBF bit. When the MSBF bit is a logical one, data will appear on the $D_{OUT}$ line in MSB first format. Logical zeros will be filled in indefinitely following the last data bit. When the MSBF bit is a logical zero, LSB first data will follow the normal MSB first data on the $D_{OUT}$ line. (see Operating Sequence)

## Transfer Curve

The LTC1286/LTC1298 are permanently configured for unipolar only. The input span and code assignment for this conversion type are shown in the following figures.

Transfer Curve




Output Code

| OUTPUT CODE | INPUT VOLTAGE | INPUT VOLTAGE ($V_{REF}$ = 5.000V) |
|---|---|---|
| 111111111111 | $V_{REF}$ – 1LSB | 4.99878V |
| 111111111110 | $V_{REF}$ – 2LSB | 4.99756V |
| • | • | • |
| • | • | • |
| • | • | • |
| 000000000001 | 1LSB | 0.00122V |
| 000000000000 | 0V | 0V |

## Operation with $D_{IN}$ and $D_{OUT}$ Tied Together

The LTC1298 can be operated with $D_{IN}$ and $D_{OUT}$ tied together. This eliminates one of the lines required to communicate to the microprocessor (MPU). Data is transmitted in both directions on a single wire. The processor pin connected to this data line should be configurable as either an input or an output. The LTC1298 will take control of the data line and drive it low on the 4th falling CLK edge after the start bit is received (see Figure 3). Therefore the processor port line must be switched to an input before this happens to avoid a conflict.

In the Typical Applications section, there is an example of interfacing the LTC1298 with $D_{IN}$ and $D_{OUT}$ tied together to the Intel 8051 MPU.



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## APPLICATION INFORMATION

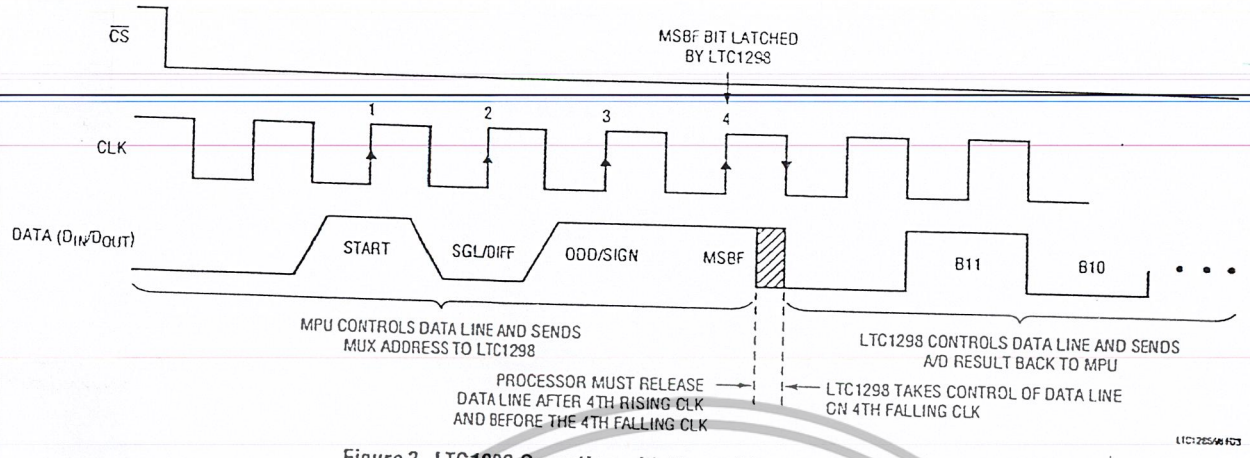



Figure 3. LTC1298 Operation with $D_{IN}$ and $D_{OUT}$ Tied Together

### ACHIEVING MICROPOWER PERFORMANCE

With typical operating currents of 250µA and automatic shutdown between conversions, the LTC1286/LTC1298 achieves extremely low power consumption over a wide range of sample rates (see Figure 4). The auto-shutdown allows the supply curve to drop with reduced sample rate. Several things must be taken into account to achieve such a low power consumption.

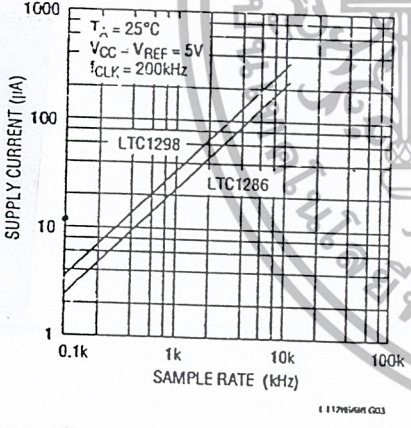



Figure 4. Automatic Power Shutdown Between Conversions Allows Power Consumption to Drop with Sample Rate.

#### Shutdown

The LTC1286/LTC1298 are equipped with automatic shutdown features. They draw power when the $\overline{CS}$ pin is low and shut down completely when that pin is high. The bias circuit and comparator powers down and the reference input becomes high impedance at the end of each conversion leaving the CLK running to clock out the LSB first data or zeroes (see Figures 1 and 2). If the $\overline{CS}$ is not running rail-to-rail, the input logic buffer will draw current. This current may be large compared to the typical supply current. To obtain the lowest supply current, bring the $\overline{CS}$ pin to ground when it is low and to supply voltage when it is high.

When the $\overline{CS}$ pin is high (= supply voltage), the converter is in shutdown mode and draws only leakage current. The status of the $D_{IN}$ and CLK input have no effect on supply current during this time. There is no need to stop $D_{IN}$ and CLK with $\overline{CS}$ = high; they can continue to run without drawing current.

#### Minimize $\overline{CS}$ Low Time

In systems that have significant time between conversions, lowest power drain will occur with the minimum $\overline{CS}$ low time. Bringing $\overline{CS}$ low, transferring data as quickly as possible, and then bringing it back high will result in the lowest current drain. This minimizes the amount of time the device draws power. After a conversion the ADC automatically shuts down even if $\overline{CS}$ is held low (see Figures 1 and 2). If the clock is left running to clock out LSB-data or zero, the logic will draw a small current. Figure 5 shows that the typical supply current with $\overline{CS}$ = ground varies from 1µA at 1kHz to 35µA at 200kHz. When $\overline{CS}$ = $V_{CC}$, the logic is gated off and no supply current is drawn regardless of the clock frequency.



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## APPLICATION INFORMATION

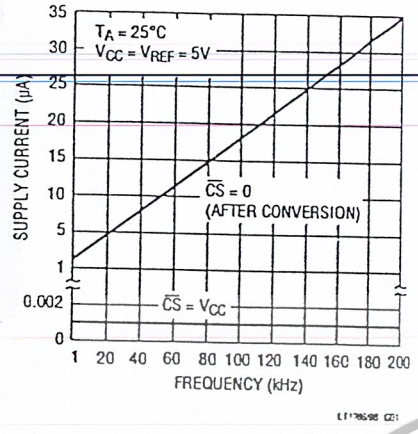



**Figure 5. Shutdown current with $\overline{CS}$ high is 1nA typically, regardless of the clock. Shutdown current with $\overline{CS}$ = ground varies from 1μA at 1kHz to 35μA at 200kHz.**

### $D_{OUT}$ Loading

Capacitive loading on the digital output can increase power consumption. A 100pF capacitor on the $D_{OUT}$ pin can add more than 50μA to the supply current at a 200kHz clock frequency. An extra 50μA or so of current goes into charging and discharging the load capacitor. The same goes for digital lines driven at a high frequency by any logic. The C × V × f currents must be evaluated and the troublesome ones minimized.

### OPERATING ON OTHER THAN 5V SUPPLIES (LTC1286)

The LTC1286 operates from 4.5V to 9V supplies and the LTC1298 operates from a 5V supply. To operate the LTC1286 on other than 5V supplies a few things must be kept in mind.

### Input Logic Levels

The input logic levels of $\overline{CS}$, CLK and $D_{IN}$ are made to meet TTL on a 5V supply. When the supply voltage varies, the input logic levels also change. For the LTC1286 to sample and convert correctly, the digital inputs have to be in the proper logical low and high levels relative to the operating supply voltage (see typical curve of Digital Input Logic Threshold vs Supply Voltage). If achieving micropower consumption is desirable, the digital inputs must go rail-to-rail between supply voltage and ground (see ACHIEVING MICROPOWER PERFORMANCE section).

### Clock Frequency

The maximum recommended clock frequency is 200kHz for the LTC1286/LTC1298 running off a 5V supply. With the supply voltage changing, the maximum clock frequency for the devices also changes (see the typical curve of Maximum Clock Rate vs Supply Voltage). If the maximum clock frequency is used, care must be taken to ensure that the device converts correctly.

### Mixed Supplies

It is possible to have a microprocessor running off a 5V supply and communicate with the LTC1286 operating on a 9V supply. The requirement to achieve this is that the outputs of $\overline{CS}$ and CLK from the MPU have to be able to trip the equivalent inputs of the LTC1286 and the output of $D_{OUT}$ from the LTC1286 must be able to toggle the equivalent input of the MPU (see typical curve of Digital Input Logic Threshold vs Supply Voltage). With the LTC1286 operating on a 9V supply, the output of $D_{OUT}$ may go between 0V and 9V. The 9V output may damage the MPU running off a 5V supply. The way to get around this possibility is to have a resistor divider on $D_{OUT}$ (Figure 6) and connect the center point to the MPU input. It should be noted that to get full shutdown, the $\overline{CS}$ input of the LTC1286 must be driven to the $V_{CC}$ voltage to keep the $\overline{CS}$ input buffer from drawing current. An alternative is to leave $\overline{CS}$ low after a conversion, clock data until $D_{OUT}$ outputs zeros, and then stop the clock low.

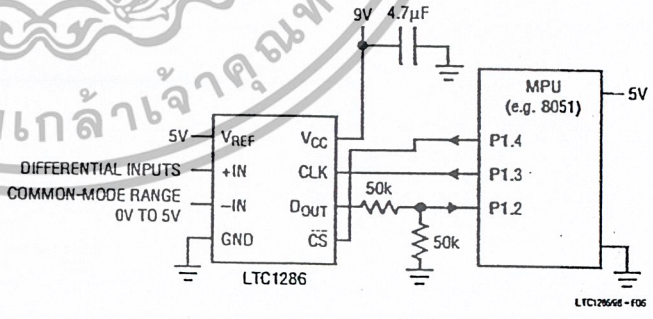



**Figure 6. Interfacing a 9V Powered LTC1286 to a 5V System**



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

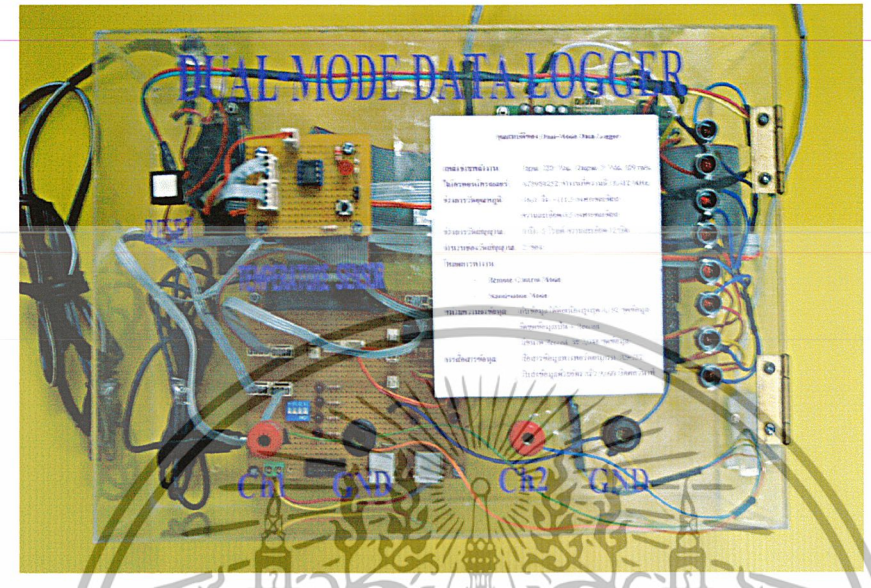
# ภาคผนวก ค.

# คู่มือการใช้งานเครื่องวัดและบันทึกข้อมูลแบบสองโหมด

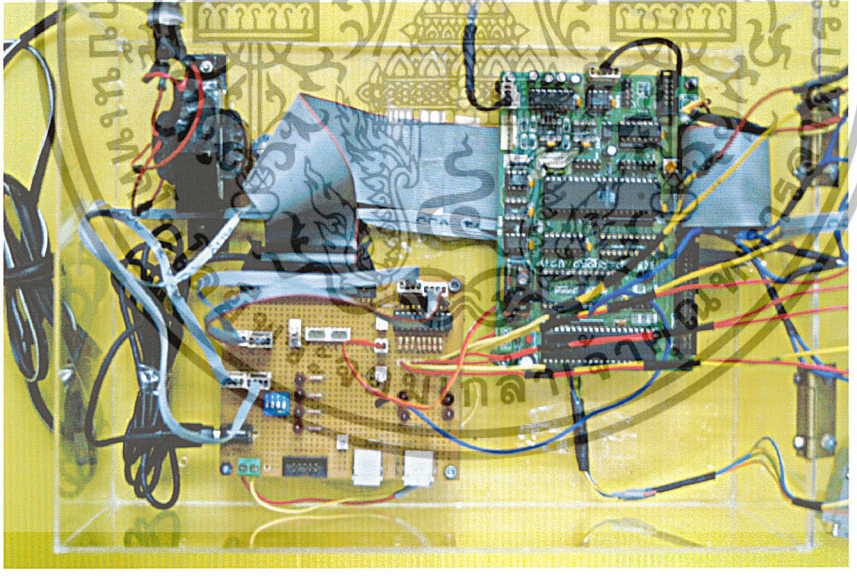
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# คู่มือการใช้งานเครื่องวัดและบันทึกข้อมูลแบบสองโหมด

1. ให้ทำการติดตั้งเครื่องวัดและบันทึกข้อมูลแบบสองโหมดให้สมบูรณ์ดังรูปที่ 1




(ก)



(ข)

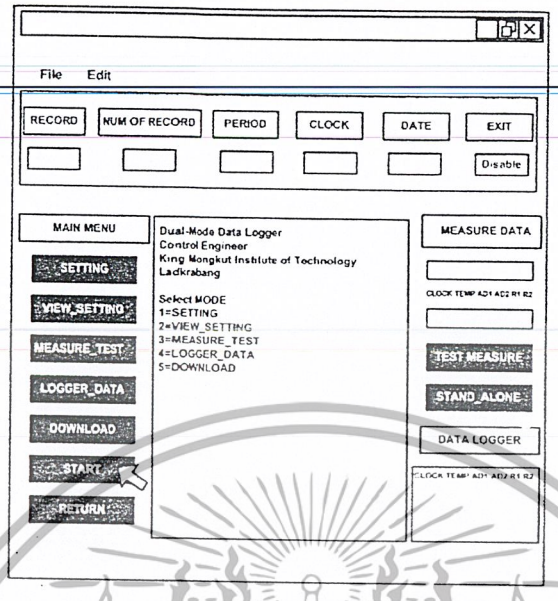
**รูปที่ 1 โครงงานสมบูรณ์ที่ทำการออกแบบและติดตั้ง**

**(ก) ภาพด้านบนแสดงหน้าปัดโครงงาน**

**(ข) ภาพการเดินสายภายในโครงงาน**

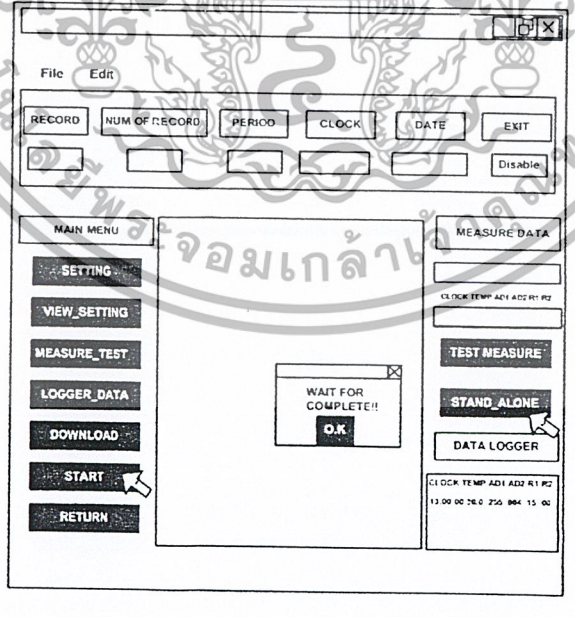
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. ติดตั้งโปรแกรมใช้งานที่อยู่ในแผ่นดิสก์ประกอบการใช้งานและทำการเปิดหน้าจอการใช้งานหลักจะปรากฏหน้าจอดังรูปที่ 2




**รูปที่ 2 หน้าจอหลักของหน้าจอ User Interface**

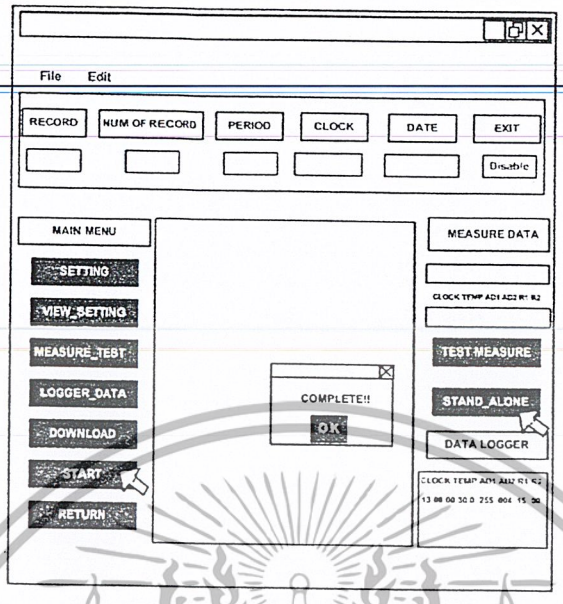
3. การใช้งานทุกครั้งให้ตรวจสอบว่าการบันทึกค่าในสภาวะ Stand-alone เดิมก่อน โดยกดปุ่ม **STAND_ALONE** ถ้ายังมีการบันทึกข้อมูลอยู่จะพบจากข้อความที่หน้าจอดังรูปที่ 3




**รูปที่ 3 ข้อความที่หน้าจอว่ายังมีข้อมูลเก่าที่ยังบันทึกไม่แล้วเสร็จ**

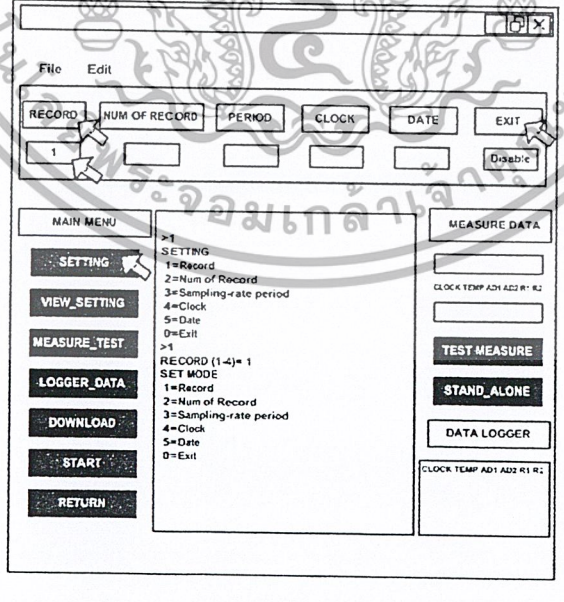
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. หากพร้อมสำหรับการใช้งานต่อไป (ข้อมูลเดิมบันทึกสมบูรณ์แล้ว) จะปรากฏหน้าจอดังรูปที่ 4



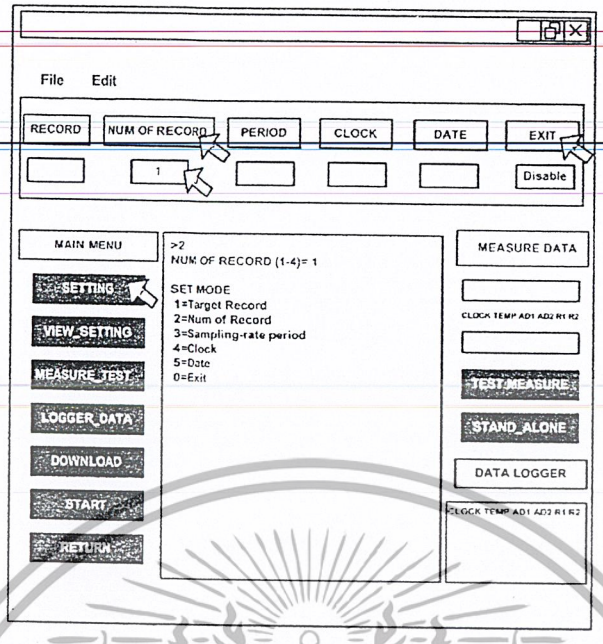

**รูปที่ 4 ข้อความที่หน้าจอว่าข้อมูลเก่าที่ต้องการบันทึกครบถ้วนแล้ว**

5. ทำการตั้งค่าเริ่มต้นในการทำงานที่ต้องการใหม่โดยการกดที่ปุ่ม **SETTING** ที่หน้าจอ และให้ทำตามลำดับขั้นดังรูปที่ 5-9 โดยในตัวอย่างแสดงการเลือกค่าเริ่มต้นที่ **RECORD = 1, NUM OF RECORD = 1, SAMPLING-RATE PERIOD = 000**

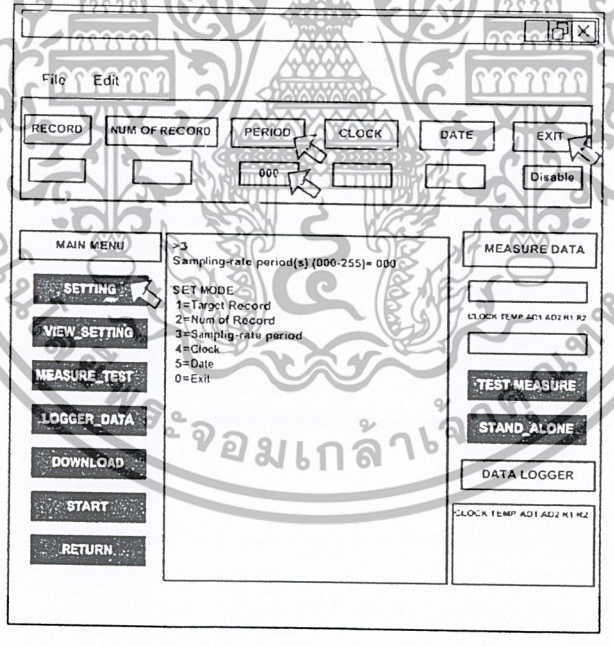



**รูปที่ 5 การตั้งค่า RECORD เริ่มต้น**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

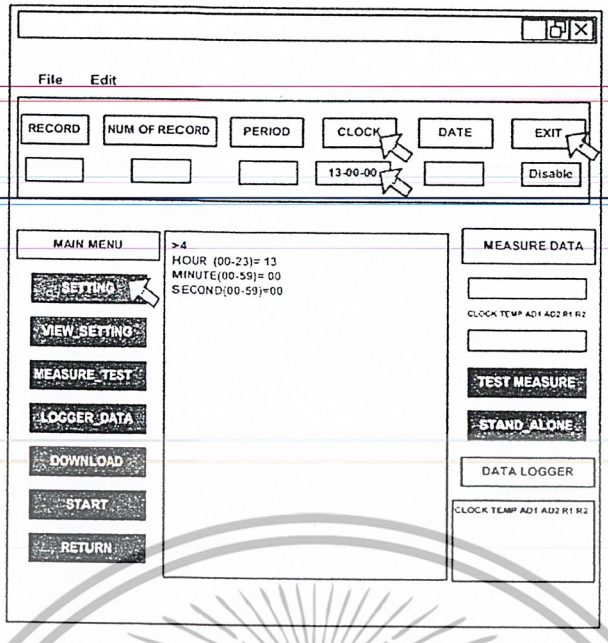



**รูปที่ 6 การตั้งค่าจำนวนของ NUM OF RECORD ที่ต้องการ**

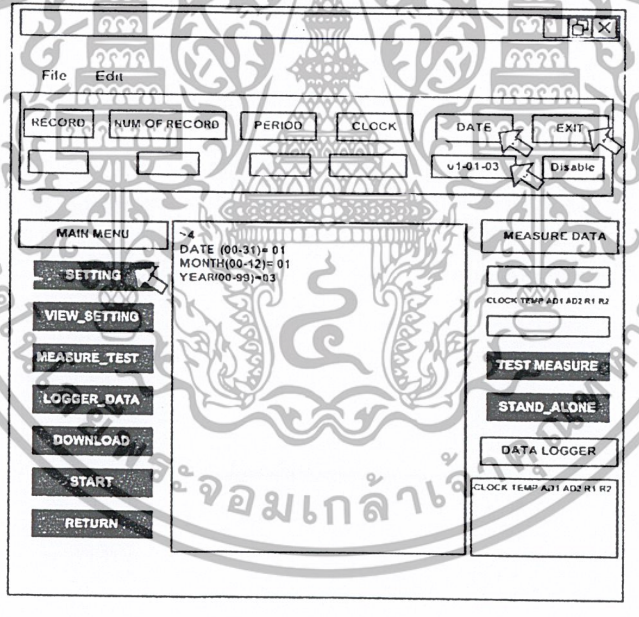



**รูปที่ 7 การตั้งค่า Sampling-rate period ในการบันทึก**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



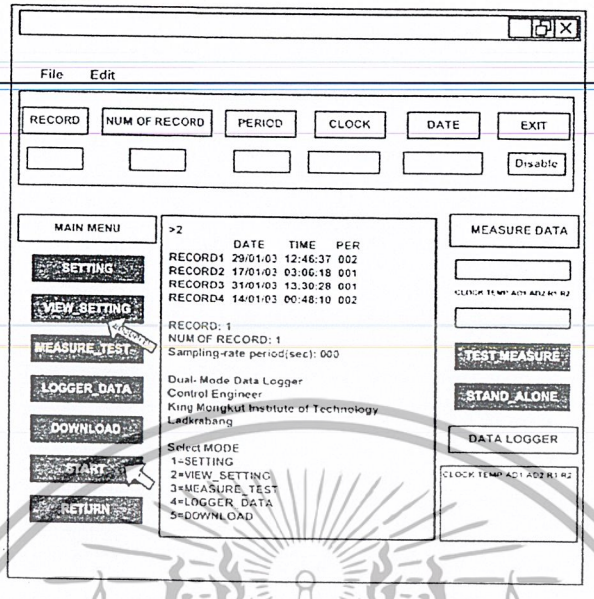

รูปที่ 8 การตั้งค่าเวลาจริงขณะปฏิบัติงาน




**รูปที่ 9 การตั้งวัน/เดือน/ปีตามเวลาปฏิบัติงานจริง**

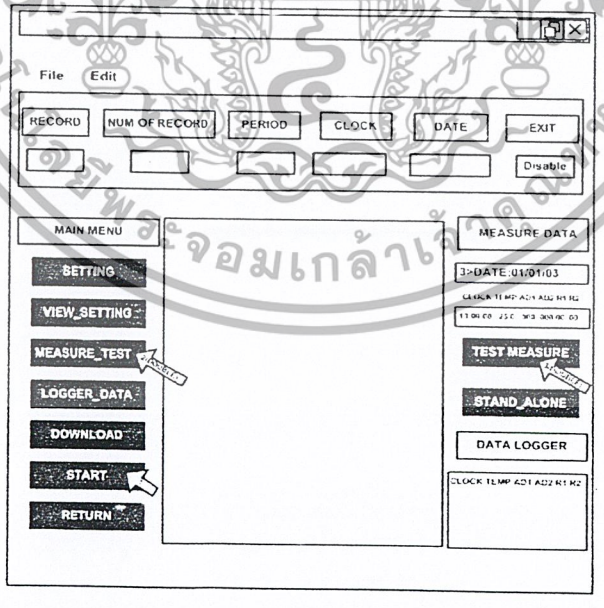
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6. เพื่อเรียกดูค่าที่ตั้งไว้โดยกดที่ปุ่ม VIEW_SETTING ที่หน้าจอ จากนั้นแล้วสังเกตข้อความจะปรากฏที่หน้าจอได้ดังรูปที่ 10




**รูปที่ 10 การเรียกดูค่าที่ตั้งไว้ใน VIEW_SETTING**

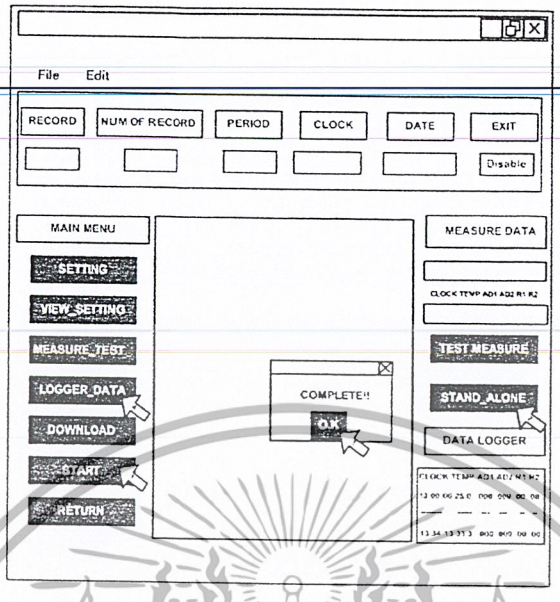
7. ทดลองทำการวัดค่าอุณหภูมิและสัญญาณทั้งสองช่องโดยการกดที่ปุ่ม MEASURE_TEST ที่หน้าจอ แล้วจะปรากฏข้อความที่หน้าจอดังรูปที่ 11




**รูปที่ 11 การทดลองวัดค่าอุณหภูมิและสัญญาณทั้งสองช่องใน MEASURE_TEST**

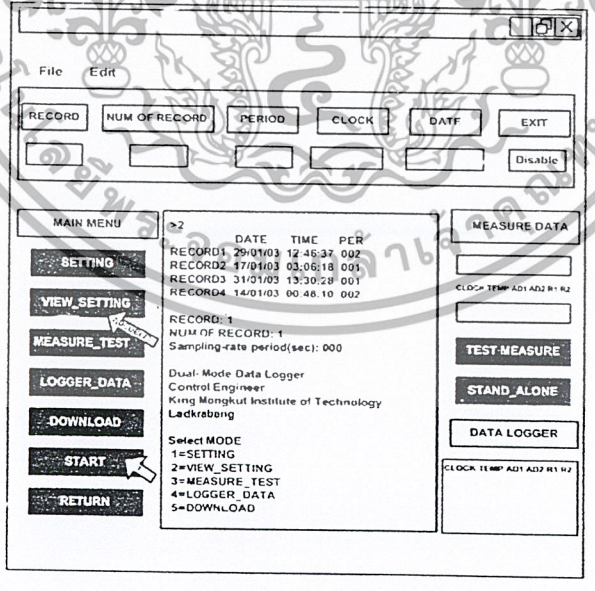
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

8. เมื่อจะทำการบันทึกค่าอุณหภูมิและส่วนรับสัญญาณอนาลอกอินพุททั้งสองช่อง โดยกดที่ปุ่ม **LOGGER_DATA** ที่หน้าจอ จะปรากฏข้อความที่หน้าจอดังรูปที่ 12




**รูปที่ 12 ข้อมูลอุณหภูมิและสัญญาณทั้งสองช่องที่ทำการบันทึก**

9. เพื่อทำการเรียกดูข้อมูลที่ทำการบันทึกค่าไว้ให้กดปุ่ม **DOWNLOAD** ที่หน้าจอจะพบข้อความปรากฏที่หน้าจอหลักดังรูปที่ 13




**รูปที่ 13 การเรียกดูข้อมูลที่ทำการบันทึกเสร็จสิ้นแล้ว**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้