

บอร์ดเก็บข้อมูลแบบไร้สาย
WIRELESS DATA LOGGER



โดย
นาย สุกรีย์ ทรงบุญญา 41014465
นาย สถาพร ศิริศักดิ์สถาพร 41014437

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิชาอิเล็กทรอนิกส์
คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เลขหมู่.....
เลขทะเบียน..... 46285
วัน, เดือน, ปี 21 ส.ค. 2546

ปีการศึกษา 2544

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

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บอร์ดเก็บข้อมูลแบบไร้สาย
WIRELESS DATA LOGGER



โดย
นาย สุกรีัย ทรงบุญญา 41014465
นาย สถาพร ศิริศักดิ์สถาพร 41014437

อาจารย์ที่ปรึกษา
ผศ.ดร. สุวิมล สมควรพาณิชย์

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาค้นคว้าหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิชาอิเล็กทรอนิกส์
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2544

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บอร์ดเก็บข้อมูล

DATA LOGGER

นาย สุกรีย์ ทรงบุญญา 41014465

นาย สถาพร ศิริศักดิ์สถาพร 41014437

ปริญญานิพนธ์นี้ได้ผ่านการตรวจสอบแล้ว พร้อมทั้งจะทำการสอบได้

ลงชื่อ.....อาจารย์ที่ปรึกษา
(ผศ.ดร. สุริณ สมควรพาณิชย์)

วันที่ 29 / 3 / 45

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

โครงการงานออกแบบและสร้างบอร์ดเก็บข้อมูลไว้สายนี้นำสำเร็จลุล่วงได้ด้วยดีเนื่องจากตลอดระยะเวลาการทำงานมี ผศ.ดร. สุริภณ สมนครพาณิชย์ อาจารย์ที่ปรึกษาโครงการ และอาจารย์ประจำภาควิชาอิเล็กทรอนิกส์ทุกท่าน ที่ให้ความเอื้อเฟื้อทางด้านอุปกรณ์และเครื่องมือต่าง ๆ ที่ใช้ในการทดลอง รวมทั้งให้คำแนะนำและเป็นที่ปรึกษาเมื่องานประสบปัญหาที่ไม่อาจแก้ไขได้ด้วยตนเอง นอกจากนี้ยังได้รับความช่วยเหลือจากรุ่นพี่และเพื่อนๆ ในภาควิชาอิเล็กทรอนิกส์

ผู้จัดทำจึงขอขอบคุณท่านอาจารย์ รุ่นพี่และเพื่อนๆ ทุกคนมา ณ โอกาสนี้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บอร์ดเก็บข้อมูลแบบไร้สาย

นาย สุกรีย์ ทรงบุญญา

นาย สถาพร สิริศักดิ์สถาพร

ผศ.ดร. สุริภณ สมควรพาณิชย์ (อาจารย์ที่ปรึกษา)

ปีการศึกษา 2544

บทคัดย่อ

บอร์ดเก็บข้อมูลแบบไร้สาย เป็นอุปกรณ์ที่บันทึกสัญญาณดิจิทัลที่ได้รับมาจากหน่วยแปลงสัญญาณอนาล็อกเป็นดิจิทัล(Analog to Digital Converter : ADC) โดยจะใช้ไมโครคอนโทรลเลอร์เป็นหน่วยควบคุมการทำงานต่างๆ เช่น การแสดงผลด้วยจอแสดงผลผลึกเหลว(Liquid Crystal Display) การนำข้อมูลที่ได้ไปเก็บในหน่วยเก็บข้อมูล(Memory) และการเชื่อมต่อกับคอมพิวเตอร์ทางพอร์ตอนุกรม(Serial Port) เป็นต้น โดยข้อมูลที่นำมาประมวลผลนี้จะอ้างอิงจากหน่วยฐานเวลาจริง(Real Time Clock) และได้มีการเพิ่มความสามารถให้สามารถเก็บข้อมูลในที่ห่างไกลจากคอมพิวเตอร์(Wireless) โดยที่ยังสามารถส่งข้อมูลมาแสดงและเก็บไว้บนฐานข้อมูลได้เหมือนการนำบอร์ดเก็บข้อมูลมาต่อกับพอร์ตอนุกรมโดยตรง โดยต้องอยู่ในขอบเขตของคลื่นวิทยุที่ใช้ส่ง

WIRELESS DATA LOGGER

Mr. Sukree Throngboonya

Mr. Sataporn Sirisaksataporn

Dr. Suripon Somkwonpanit (Advisor)

Education Year 2001

Abstract

The wireless data logger is a component that can be records ditital signal from analog to digital converter unit.It controled by Microcontroller to process any work such as display data in LCD unit, store data in memory unit and interface with computer with serial port etc. .The data will be identify the time value by Real Time Clock.The wireless data logger can connect to computer form any location where the radio signal can be detected.

สารบัญ

	หน้า
กิตติกรรมประกาศ	-
บทคัดย่อ	-
Abstract	-
สารบัญ	-
สารบัญรูป	-
สารบัญตาราง	-
บทที่ 1 บทนำ	1
บทที่ 2 ทฤษฎีและหลักการพื้นฐาน	3
2.1 อุปกรณ์เซ็นเซอร์	3
2.2 การแปลงสัญญาณอนาล็อกเป็นดิจิทัล	5
2.3 ไมโครคอนโทรลเลอร์ 8051	14
2.4 การติดต่อสื่อสารผ่านพอร์ตอนุกรม	26
2.5 การจับโมดูลแสดงผลแบบผลึกเหลว (LCD)	33
2.6 การโปรแกรมแบบวิซวล (Visual Programming)	41
2.7 DS1307 ไอซีสร้างฐานเวลาจริงหรือรีลไทม์คล็อก (RTC)	45
2.8 การเชื่อมต่อคีย์บอร์ด	53
2.9 การใช้งานหน่วยความจำแบบ EEPROM	55
2.10 การส่งข้อมูลแบบไร้สาย	55
2.11 ขั้นตอนการติดต่อข้อมูลระหว่าง MICROCONTROLLER กับ DS1820	55
บทที่ 3 การออกแบบและการสร้าง	56
3.1 การออกแบบส่วนฮาร์ดแวร์ (Hardware)	56
3.2 ส่วนของซอฟต์แวร์และโปรแกรมควบคุม	61
บทที่ 4 การทดลอง	65
4.1 ผลการทดลองการแปลงอนาล็อกเป็นดิจิทัล	65
4.2 ผลการทดลองการเทียบค่าอุณหภูมิที่วัดได้จาก DS1820 กับค่าอุณหภูมิจริง	66
4.3 FSK Modulation ด้วย XR2206	67
4.4 FSK Demodulation ด้วย XR2211	67

4.5 FM Transmitter	68
4.6 RF Amplifier	69
4.7 FM Receiver	69
4.8 สัญญาณ IF	69
4.9 สัญญาณข้อมูลหลังการ Demodulation IF	70
บทที่ 5 สรุปผลและวิจารณ์	71
เอกสารอ้างอิง	
ภาคผนวก	



สารบัญรูป

	หน้า
รูปที่ 1.1 บล็อกไคอะแกรมบอร์ดเก็บข้อมูลแบบไร้สาย	1
รูปที่ 2.1 แสดงตำแหน่งขาสัญญาณของ LM335	3
รูปที่ 2.2 แสดงการต่อวงจร	4
รูปที่ 2.3 Parallel Comparator ADC แบบ 2 บิต	8
รูปที่ 2.4 Dual Slope ADC	8
รูปที่ 2.5 กราฟเอาต์พุตของ Integrator	9
รูปที่ 2.6 Successive Approximation ADC ขนาด 8 บิต	10
รูปที่ 2.7 การต่ออุปกรณ์บนระบบบัส I ² C	11
รูปที่ 2.8 รูปแบบของการกำหนดแอดเดรสแบบ 7 บิต	12
รูปที่ 2.9 รูปแบบการกำหนดข้อมูลควบคุม	13
รูปที่ 2.10 แสดงโครงสร้างภายในของ 8051	15
รูปที่ 2.11 แสดงขาของ 8051	18
รูปที่ 2.12 โครงสร้างภายในพอร์ตอินพุต/เอาต์พุตของ 8051	19
รูปที่ 2.13 แสดงหน่วยความจำโปรแกรมของ 8051	20
รูปที่ 2.14 แสดงหน่วยความจำข้อมูลของ 8051	21
รูปที่ 2.15 แสดงหน่วยความจำข้อมูลภายใน	22
รูปที่ 2.16 แสดงรายละเอียดของ Special Function Register	23
รูปที่ 2.17 ข้อมูลอนุกรม 10 บิต	27
รูปที่ 2.17.ก Timer Mode 2	29
รูปที่ 2.17.ข Serial Port Mode 1	30
รูปที่ 2.18 ไคอะแกรมการทำงานของ โมดูล LCD แบบอักขระ	34
รูปที่ 2.19 รูปร่างและการจัดขาโมดูล LCD	36
รูปที่ 2.20 ตำแหน่ง DDRAM	40
รูปที่ 2.21 ลักษณะของ โปรแกรมที่ทำงานด้วยฮิวเอนต์	44
รูปที่ 2.22 แสดงการจัดขาของ DS1307	46
รูปที่ 2.23 แสดงส่วนประกอบหลักที่สำคัญและไคอะแกรมการทำงานของ DS1307	47

รูปที่ 2.24 แสดงการจัดสรรพื้นที่ของหน่วยความจำภายใน DS1307	49
รูปที่ 2.25 แสดงรายละเอียดของรีจิสเตอร์ค่าเวลาและรีจิสเตอร์ควบคุมของ DS1307	49
รูปที่ 2.26 โหมดการอ่าน/เขียนข้อมูล	52
รูปที่ 2.27 ตัวอย่างวงจร	53
รูปที่ 2.28 การอ่านค่าคีย์บอร์ดและการกระพือของหน้าสัมผัสสวิตช์	54
รูปที่ 2.29 การต่อคีย์บอร์ด	55
รูปที่ 3.1 วงจรส่วนประกอบทั้งหมดของ Data Logger	57
รูปที่ 3.2 วงจรการ โมดูเลท และ ดีโมดูเลทโดยใช้การติดต่อแบบอนุกรมผ่าน RS232	58
รูปที่ 3.3 วงจรภาคส่ง FM (วงจร Oscillator และ Modulator)	59
รูปที่ 3.4 วงจร RF Amplifier	60
รูปที่ 3.5 วงจรภาครับ FM	60
รูปที่ 3.6 วงจร Limiter & Detector	61
รูปที่ 3.7 Flow Chart การทำงานของ โปรแกรม	62
รูปที่ 3.8 หน้าต่างของตัวโปรแกรมหลังรัน โปรแกรม	63
รูปที่ 4.1 สัญญาณ FSK ที่ได้จากเอาต์พุตของ XR2206	67
รูปที่ 4.2 สัญญาณเอาต์พุตที่ได้จาก XR2211	68
รูปที่ 4.3 สัญญาณที่ได้หลังจาก FM Modulation	68
รูปที่ 4.4 สัญญาณเอาต์พุตจากภาครับ FM	69
รูปที่ 4.5 สัญญาณ IF 10.7 MHz	70
รูปที่ 4.6 สัญญาณ FSK ที่ได้หลังจากการดีโมดูเลท	70

สารบัญตาราง

	หน้า
ตารางที่ 2.1 แสดงความสัมพันธ์ระหว่างค่าอุณหภูมิหน่วย $^{\circ}\text{C}$, K และ V_{out}	4
ตารางที่ 2.2 แสดงการเข้ารหัสสัญญาณ	7
ตารางที่ 2.3 แสดงหน้าที่พิเศษของพอร์ต 3	17
ตารางที่ 4.1 ผลการทดลองวงจรแปลงสัญญาณอนาล็อกเป็นดิจิทัล	64
ตารางที่ 4.2 แสดงการเทียบค่าอุณหภูมิที่วัดได้จาก IC กับที่ได้จากเทอร์โมมิเตอร์	66



บทที่ 1

บทนำ

การออกแบบและสร้างโครงงานนี้จะแบ่งออกเป็นสองส่วนใหญ่ๆ คือ ส่วนบอร์ดเก็บข้อมูล และส่วนการส่งและรับข้อมูลผ่านคลื่นวิทยุ

โครงงานนี้มีกระบวนการ โดยคร่าวๆเริ่มตั้งแต่การศึกษาวงจรในภาคต่างๆของบอร์ดเก็บข้อมูล โดยนำส่วนประกอบต่าง ๆ ของวงจรมาศึกษาโดยละเอียดตั้งแต่ส่วนวงจรแปลงค่าอุณหภูมิเป็นสัญญาณไฟฟ้า วงจรแปลงสัญญาณอนาลอก เป็นสัญญาณดิจิทัล ตัวไมโครคอนโทรลเลอร์ การเขียนโปรแกรมการทำงานไมโครคอนโทรลเลอร์ การใช้รีเลย์บอร์ด การใช้ไอซีฐานเวลาจริง การเก็บข้อมูลและการอ่านข้อมูลจากหน่วยความจำ การติดต่อกับคอมพิวเตอร์ผ่านทางพอร์ตอนุกรม การแสดงข้อมูลโดยใช้จอผลึกเหลว ลำดับต่อมาเป็นการทดลองต่อวงจรตามที่ได้ออกแบบไว้บนโปรโตบอร์ด เพื่อตรวจสอบคุณสมบัติให้เป็นไปตามที่ได้ออกแบบไว้ เมื่อได้วงจรตามที่ได้ออกแบบไว้แล้ว จึงทำการย้ายวงจรมาต่อลงบนบอร์ดคอนเนกประสงค์ซึ่งจะใช้ในการสร้างชิ้นงานจริง ต่อมาก็ทำการประกอบอุปกรณ์ต่าง ๆ เข้าด้วยกัน สุดท้ายก็ทดสอบประสิทธิภาพว่าสามารถทำงานได้ตามที่ได้กำหนดไว้หรือไม่ โครงงานนี้ประกอบด้วยส่วนวงจรสองส่วน ดังรูป



รูปที่ 1.1 บล็อกไดอะแกรมบอร์ดเก็บข้อมูลแบบไร้สาย

วัตถุประสงค์ของโครงการ

สามารถออกแบบและสร้างบอร์ดเก็บข้อมูลแบบไร้สายที่สามารถเก็บข้อมูลอุณหภูมิพร้อมทั้งระยะเวลาที่เก็บได้และสามารถติดต่อเพื่อส่งข้อมูลไปเก็บบนคอมพิวเตอร์โดยใช้คลื่นความถี่วิทยุเพื่อสามารถนำข้อมูลที่ได้ไปแสดงผลและวิเคราะห์ต่อไป

ประโยชน์และผลที่คาดว่าจะได้รับ

1. เข้าใจการทำงานของวงจรส่วนต่างๆ ที่เป็นส่วนประกอบของบอร์ดเก็บข้อมูล เช่น การใช้ไอซีเปลี่ยนอุณหภูมิเป็นสัญญาณไฟฟ้า วงจรแปลงสัญญาณอนาลอกเป็นสัญญาณดิจิทัล การใช้งานไมโครคอนโทรลเลอร์ควบคุมส่วนประกอบต่างๆ การติดต่อกับคอมพิวเตอร์ผ่านทางพอร์ตอนุกรม การเขียนโปรแกรมบนคอมพิวเตอร์เพื่อรับข้อมูลมาแสดงผลและเก็บ เป็นต้น
2. ได้รับความรู้ความเข้าใจและเกิดทักษะความชำนาญในการใช้เครื่องมือ และอุปกรณ์ต่าง ๆ ทางอิเล็กทรอนิกส์ เช่น ไมโครคอนโทรลเลอร์ การแปลงสัญญาณอนาลอกเป็นดิจิทัล การแสดงผลบนจอผลึกเหลว เป็นต้น
3. สามารถนำบอร์ดเก็บข้อมูลที่ได้ไปใช้งานในการเก็บข้อมูลอุณหภูมิในที่ต่างโดยมีความสะดวกคือสามารถติดต่อส่งข้อมูลมาเก็บบนฐานข้อมูลบนเครื่องคอมพิวเตอร์จากจุดที่วัดได้เลยโดยไม่ต้องนำเครื่องมาต่อกับกับคอมพิวเตอร์โดยตรง

บทที่ 2

ทฤษฎีและหลักการพื้นฐาน

2.1 อุปกรณ์เซ็นเซอร์

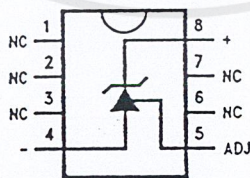
เซ็นเซอร์คืออุปกรณ์ที่ทำหน้าที่ในการแปลงข้อมูลทางกายภาพ เช่น อุณหภูมิ ความชื้น ความสว่าง เป็นต้นให้อยู่ในรูปของสัญญาณทางไฟฟ้า เพื่อสามารถนำสัญญาณที่วัดมาทำการประมวลผลและแสดงผลได้ เซ็นเซอร์ที่ใช้ในโครงการ คือ เซ็นเซอร์อุณหภูมิ ซึ่งจะกล่าวถึงรายละเอียดของเซ็นเซอร์แต่ละชนิดดังต่อไปนี้

2.1.1 อุปกรณ์เซ็นเซอร์อุณหภูมิ

ในการวัดอุณหภูมิ จะใช้เซ็นเซอร์คานธีลทอนิกส์ ซึ่งมีโครงสร้างภายในเป็นซีเนอริไดโอดที่สามารถเปลี่ยนแปลงแรงดันพังทลาย (Breakdown Voltage) ตามระดับอุณหภูมิ ดังนั้น เมื่ออุณหภูมิมีการเปลี่ยนแปลง แรงดันที่ตกคร่อมอุปกรณ์เซ็นเซอร์ดังกล่าวจะเปลี่ยนแปลงตามไปด้วย เราจึงสามารถนำการเปลี่ยนแปลงดังกล่าวมาเปรียบเทียบและประมวลผล

สำหรับโครงการนี้เราใช้คานธีลทอนิกส์เซ็นเซอร์เบอร์ LM335 (Precision Temperature Sensor) ซึ่งมีอัตราเปลี่ยนแปลงแรงดันต่ออุณหภูมิเท่ากับ $+10 \text{ mV/K}$ และทำงานที่ย่านอุณหภูมิ -40 องศาเซลเซียส มีความเที่ยงตรงสูงในย่านที่ใช้งาน ราคาถูก หาซื้อได้ง่าย จากรูปที่

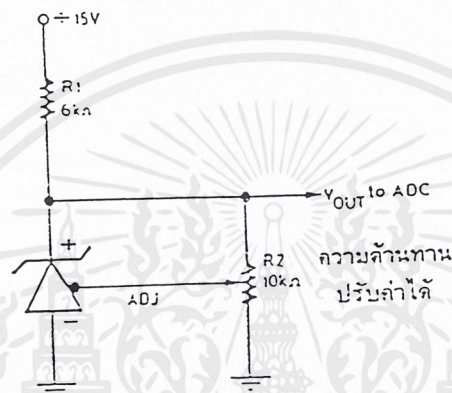
2.1 แสดงลักษณะการใช้งานของ LM335



รูปที่ 2.1 แสดงตำแหน่งขาสัญญาณของ LM335

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในการนำอิเล็กทรอนิกส์เซ็นเซอร์ดังกล่าวมาใช้งาน จำเป็นจะต้องมีวงจรต่อร่วมเพื่อการใช้งานได้อย่างมีประสิทธิภาพ โดยวงจรที่ใช้แสดงดังรูปที่ 2.2 และจากวงจรรูปที่ 2.2 สามารถหาความสัมพันธ์ระหว่างแรงดันที่ตกคร่อมตัวเซ็นเซอร์กับค่าอุณหภูมิ ได้ดังตารางที่ 2.1 โดยในวงจรยังสามารถปรับค่าแรงดันได้ที่ VR อีกด้วย



รูปที่ 2.2 แสดงการต่อวงจร

อุณหภูมิ		
องศาเซลเซียส	องศาเคลวิน	V_{out}
0	273.2	2.73
25	298.2	2.98
50	323.3	3.23
75	348.2	3.48
100	373.2	3.73

ตารางที่ 2.1 แสดงความสัมพันธ์ระหว่างค่าอุณหภูมิหน่วย °C , K และ V_{out}

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2 การแปลงสัญญาณอนาลอกเป็นดิจิตอล (Analog to Digital Converter : ADC)

วงจรแปลงสัญญาณอนาลอกเป็นสัญญาณดิจิตอล(ADC) ทำหน้าที่แปลงขนาดของสัญญาณที่เป็นแรงดัน หรือกระแสที่เปลี่ยนแปลงอย่างต่อเนื่องให้เป็นค่าข้อมูลดิจิตอล สัญญาณที่ได้จาก ADC สามารถนำไปใช้เป็นสัญญาณอินพุตให้กับไมโครคอนโทรลเลอร์ได้ สัญญาณนี้จะถูกไมโครคอนโทรลเลอร์อ่านไปเก็บไว้ในหน่วยความจำเพื่อใช้ในการประมวลผลต่อไป

การนำสัญญาณอนาลอก (Analog) มาประมวลผลในวงจรดิจิตอล (Digital) จำเป็นต้องอาศัยตัวแปลงสัญญาณอนาลอกเป็นสัญญาณดิจิตอล หรือเรียกว่า ADC ซึ่งความละเอียดของการแปลงสัญญาณขึ้นอยู่กับจำนวนบิตของ ADC ในการแปลงสัญญาณอนาลอกเป็นดิจิตอลนั้นมีขั้นตอนต่างๆ ที่ซับซ้อนอยู่พอสมควร ว่าจะได้เป็นสัญญาณดิจิตอลที่มีค่าผิดพลาด (error) ต่ำ ซึ่งโดยส่วนใหญ่แล้วจะประกอบด้วยขั้นตอนที่สำคัญอยู่ 3 ขั้นตอน คือ การสุ่ม (Sampling) , การจัดระดับสัญญาณ (Quantizing) และการเข้ารหัส (Encoder) โดยทั้งหมดนั้นถือเป็นทฤษฎีพื้นฐานในการแปลงสัญญาณอนาลอกเป็นดิจิตอล

2.2.1 ทฤษฎีของการสุ่ม

ในการแปลงสัญญาณอนาลอกมาเป็นรหัสดิจิตอลนั้น ADC จำเป็นต้องใช้เวลาช่วงหนึ่งในการจัดการ ซึ่งช่วงเวลาดังกล่าวขึ้นอยู่กับหลายๆ แฟกเตอร์ เช่น ความละเอียดของการแปลงสัญญาณ (จำนวนดิจิตอล) เทคนิคของการแปลงสัญญาณ และความเร็วในการทำงานของอุปกรณ์ร่วมอื่นๆ การกำหนดความเร็วของการแปลงสัญญาณขึ้นอยู่กับ การประยุกต์ใช้งานเฉพาะอย่างและความแม่นยำที่ต้องการ

ช่วงเวลาในการแปลงสัญญาณบางครั้งอาจเรียกว่า Aperture time ซึ่งหมายถึงช่วงเวลาที่เกิดความไม่แน่นอนขึ้นในการวัด และผลก็คือความผิดพลาด (error) ต่อค่าที่วัดได้

2.2.2 การจัดระดับสัญญาณ (Quantizing)

อีกขั้นตอนหนึ่งที่สำคัญหลังจากสัญญาณผ่านการสุ่มมาแล้วก็คือ ขั้นตอนการจัดระดับ (Quantizing) ของสัญญาณ โดยสัญญาณที่ผ่านการสุ่มมาแล้วจะมีลักษณะที่ไม่ต่อเนื่อง (Discrete Signal) และเนื่องจากสัญญาณอนาลอกส่วนใหญ่จะมีสัญญาณรบกวน (noise) ปะปนมาด้วยเสมอ จึงจำเป็นจะต้องกำจัดสัญญาณเหล่านี้ออกไปเพื่อไม่ให้เกิดความผิดพลาดในการประมวลผลแบบดิจิตอลและสามารถนำไปเข้ารหัสได้ง่าย

เมื่อผ่านการสุ่ม (Sampling) และการจัดระดับ (Quantizing) แล้วข้อมูลที่เป็นสัญญาณอนาลอกจะถูกเปลี่ยนเป็นระดับสัญญาณที่แน่นอนและมีลักษณะไม่ต่อเนื่องเป็นลำดับต่างๆ เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กัน จะมีลักษณะสัญญาณที่ประมาณออกมา ซึ่งแน่นอนว่าจะต้องมีส่วนที่ผิดพลาด (error) อยู่ด้วย และอีกอย่างที่สำคัญมากก็คือ ความละเอียด (Resolution) ของตัวแปลงว่ามีกี่บิตเพราะจำนวนบิตของตัวแปลง จะบอกถึงจำนวนระดับสัญญาณคือ

$$\text{จำนวนระดับสัญญาณ (Q)} = 2^n \quad ; \quad n \text{ คือจำนวนบิต} \quad (2.1)$$

ถ้าเราใช้ตัวแปลงขนาด 8 บิต กับขนาดสัญญาณอนาล็อกที่สูงสุด 10 โวลต์ จะได้จำนวนระดับสัญญาณเท่ากับ = 256 ระดับ ดังนั้นค่าของหนึ่งบิตค่าน้ำ (Least Significant Bit : LSB) จะมีค่าเป็น

$$\text{ค่าหนึ่งบิตค่าน้ำ} = \text{แรงดันสัญญาณสูงสุด/จำนวนระดับสัญญาณ} \quad (2.2)$$

ในบางครั้งเราจะเรียก LSB ว่า “ Step Size ” โดยใช้สัญลักษณ์จากตัวแปลง 8 บิต และขนาดสัญญาณ 10 โวลต์ จะมีค่า $\Delta = 0.039\text{V}$. แล้วอาจเขียนสมการในการหาค่า Δ ได้ใหม่ว่า

$$\Delta = \text{Full Scale Signal} / Q \quad (2.3)$$

จะเห็นว่า จำนวนบิตยิ่งมากค่า Δ จะยิ่งลดลง จะทำให้ค่าความผิดพลาดลดลงไปด้วยโดยค่าความผิดพลาดจะอยู่ระหว่าง 0 ถึง $\Delta/2$ ซึ่งค่าความผิดพลาดอาจเป็นศูนย์ถ้าสัญญาณอนาล็อกมีค่าอยู่ที่จุดกึ่งกลางพอดี

2.2.3 การเข้ารหัส (Encoded)

ในการเข้ารหัสสัญญาณที่ผ่านการสุ่มและการวัดระดับสัญญาณมาแล้วนั้น ส่วนใหญ่จะแปลงให้อยู่ในรูปของรหัสตัวเลขฐานสอง (Binary Code) แล้วเปลี่ยนจากข้อมูลแบบขนานให้เป็นแบบอนุกรมเพื่อสามารถลดช่องสัญญาณในการส่งข้อมูลให้เหลือเพียงช่องสัญญาณเดียวได้

สมมติว่าสัญญาณอนาล็อก $x(t)$ มีขนาดสูงสุดอยู่ที่ $\pm 4\text{ Vpp}$ และมีอัตราการสุ่มที่เป็น 2 เท่า ของความถี่สูงสุด มีค่า $Q = 16$ ดังนั้นจึงมี $\Delta = 0.5\text{ V}$. แล้วจุดระดับ X_i จะอยู่ที่ -4, -3.5, -3, -2.5,, 0,, 2.5, 3, 3.5, 4 V และแรงดันระดับอ้างอิง (mi) จะเป็น -3.75, -3.25,, 3.25, 3.75 V. ตามลำดับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากตารางที่ 2.2 จะแสดงลำดับการเข้ารหัสของระดับแรงดันจากการสุ่มและผ่านการจัดระดับสัญญาณมาแล้ว ซึ่งที่ระดับแรงดันค่าที่สุดก็จะมีรหัสเป็นเลขฐานสอง 0000 ส่วนระดับแรงดันสูงที่สุดก็จะมีรหัสเป็นเลขฐานสอง 1111 เป็นต้น

ระดับแรงดัน	รหัสเป็นเลขฐานสอง	ระดับแรงดัน	รหัสเป็นเลขฐานสอง
-3.75	0000	0.25	1000
-3.25	0001	0.75	1001
-2.75	0010	1.25	1010
-2.25	0011	1.75	1011
-1.75	0100	2.25	1100
-1.25	1010	2.75	1101
-0.75	0110	3.25	1110
-0.25	0111	3.75	1111

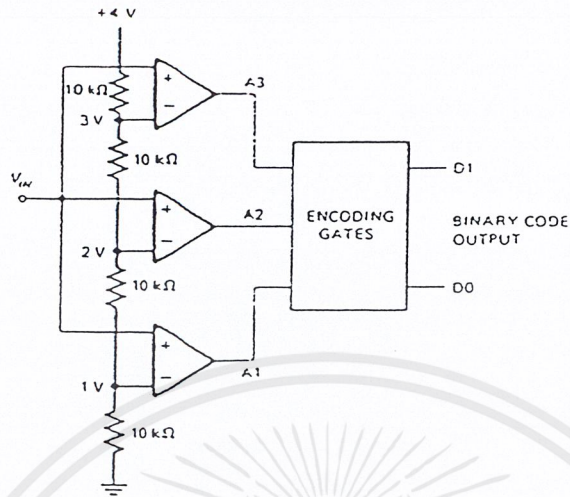
ตารางที่ 2.2 แสดงการเข้ารหัสสัญญาณ

จากขั้นตอนทั้งหมดของการแปลงสัญญาณอนาลอกเป็นดิจิทัลนั้น เราสามารถกำหนดความ สามารถหรือคุณภาพ (Quality) ในการแปลงได้ ด้วยปัจจัยหนึ่งที่สำคัญมากก็คือ “ ความละเอียด (Resolution) ” ซึ่งขึ้นอยู่กับจำนวนบิตนั่นเอง แต่ในการใช้งานจริงแล้วยังมีเรื่องความเร็วในการแปลงอีกด้วยและที่ใช้งานจริงจะมีตัวแปลงอยู่หลายชนิดอีกด้วย

ADC มีคุณสมบัติที่สำคัญคือ Conversion Time หมายถึงเวลาที่ใช้ในการแปลงสัญญาณอนาลอกที่ได้มาเป็นสัญญาณดิจิทัล ซึ่งเวลาที่ใช้จะขึ้นอยู่กับวิธีที่ ADC ใช้แปลงสัญญาณ โดยสามารถแบ่งได้ดังนี้

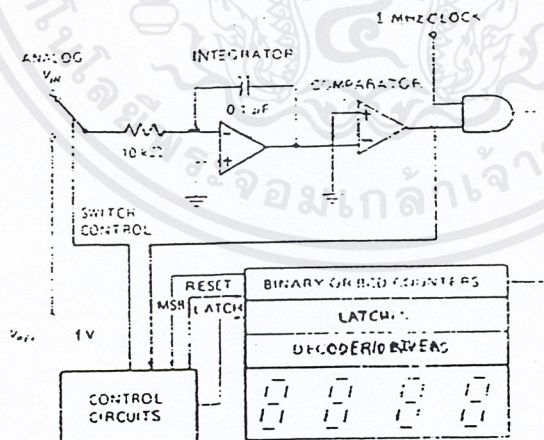
2.6.1 Parallel Comparator ADC

จากรูปที่ 2.3 จะเป็น ADC 2 บิตโดยแรงดันที่อินพุตที่เข้ามา จะถูกนำไปเปรียบเทียบกับแรงดันอ้างอิงค่าต่างๆและผลลัพธ์ที่ได้จะถูกส่งต่อไปยัง Encoding Gate เพื่อสร้างรหัสของเอาต์พุตที่ต้องการซึ่งสามารถจะให้รหัสเป็นอย่างไรก็ได้ขึ้นกับการออกแบบ Gate ข้อดีคือแปลงสัญญาณได้เร็ว แต่มีข้อเสียคือราคาแพง



รูปที่ 2.3 Parallel Comparator ADC แบบ 2 บิต

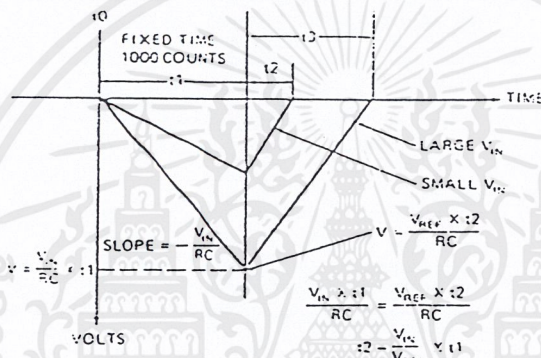
2.6.2 Dual Slope ADC



รูปที่ 2.4 Dual Slope ADC

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

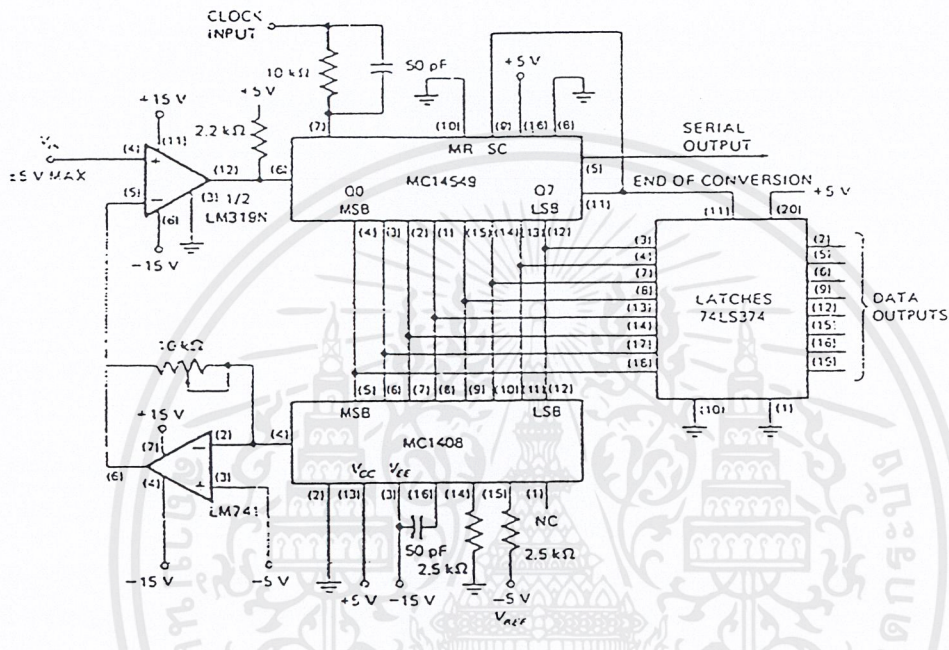
หลักการทำงานคือ เมื่อเริ่มส่วนของ Control Circuit จะทำการ Reset ค่าของ Counter ให้เป็น “0” และทำการค่ออินพุทของส่วน Integrator เข้ากับสัญญาณนาฬิกาที่จะทำการแปลง ถ้าสมมุติให้ แรงดันที่ต้องการแปลงมีค่าเป็น V_{IN} บวก ดังนั้นเอาต์พุทของ Integrator ก็จะลดลงตามเวลาเป็นเส้นตรงดัง รูปที่ 2.5



รูปที่ 2.5 กราฟเอาต์พุทของ Integrator

เมื่อเอาต์พุทของ Integrator เริ่มเป็นลบส่วน Comparator ก็จะให้เอาต์พุทที่เป็น “1” ออกมา ซึ่งจะทำให้ AND Gate ส่งสัญญาณ Clock 1 MHz เข้าไปยังส่วนของ Counter หลังจาก Counter นับไปได้เป็นจำนวนที่กำหนด ส่วนของ Control Circuit ก็จะสลับอินพุทของ Integrator ให้ไปที่ แรงดันอ้างอิงที่เป็นลบและ Reset ค่าของ Counter เมื่อมีแรงดันไฟที่เป็น ลบ ป้อนเข้าที่ Integrator ก็จะทำให้อเอาต์พุทของมันค่อยๆเพิ่มขึ้นด้วยอัตราคงที่และเมื่อ เอาต์พุทของ Integrator นี้มีค่ากลับถึง “0” ก็จะทำให้ AND Gate หยุดส่งสัญญาณ Clock ให้กับ Counter ค่าจำนวนที่นับได้ของ Counter นี้จึงขึ้นอยู่กับอินพุท Voltage ที่ป้อน ข้อดีคือสามารถทำให้มีจำนวนบิตที่มากได้ในราคาถูก ข้อเสียคือใช้เวลาในการแปลงสัญญาณที่นานมาก

2.6.3 Successive Approximation ADC



รูปที่ 2.6 Successive Approximation ADC ขนาด 8 บิต

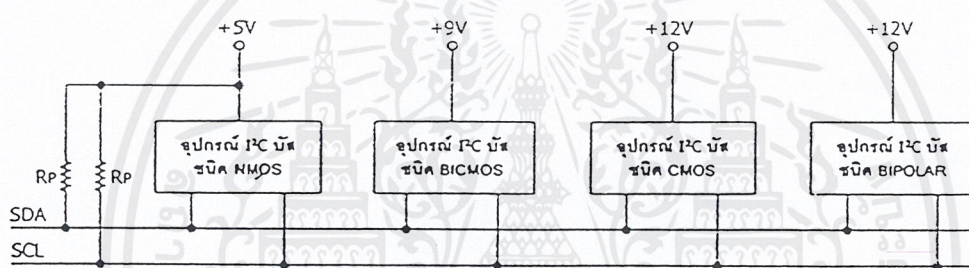
จากรูปที่ 2.6 ส่วนประกอบที่สำคัญคือ SAR(MC14549) โดยจะมีการทำงานเมื่อมีสัญญาณ Start of Conversion(SC) ที่สัญญาณ Clock ลูกแรก SAR จะทำให้ MSB มีค่าเป็น “1” (1000 0000) ซึ่งค่านี้จะเข้าไปยัง DAC(MC1408) ได้เป็นระดับแรงดันออกมา และนำไปเปรียบเทียบกับสัญญาณอินพุทของ ADC ถ้าแรงดันนี้มากกว่าสัญญาณอินพุทส่วนของ Comparator จะส่งสัญญาณไปที่ SAR ว่าค่าที่ MSB นี้มากเกินไป SAR จะทำให้บิตนี้เป็น “0” แต่ถ้าค่าที่ได้น้อยกว่าสัญญาณอินพุทส่วน SAR จะคงค่าข้อมูลเป็น “1” ไว้ ในสัญญาณ Clock ลูกต่อมา SAR ก็จะทำให้ข้อมูลบิตถัดจาก MSB มีค่าเป็น “1” และจะเกิดขบวนการเหมือนบิตที่แล้วจนถึง LSB โดยใช้ Clock 9 ลูก แล้ว SAR จะส่งสัญญาณ End of Conversion ไปที่ Latch เพื่อให้ทำการเก็บข้อมูลไว้ และในการที่จะให้ ADC ติดต่อกับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กับตัวไมโครคอนโทรลเลอร์นั้นมีการติดต่อได้หลายแบบแต่ในการออกแบบนี้ได้เลือกใช้ระบบบัส I²C ซึ่งมีรายละเอียดดังนี้

การใช้งานไอซี ADC บนระบบบัส I²C

I²C ย่อมาจาก Inter IC-Communication หมายถึงการติดต่อสื่อสารระหว่าง IC ด้วยบัส I²C จุดมุ่งหมายคือต้องการให้ไอซีสามารถ ติดต่อ ตั้งงานและควบคุมผ่านสายสัญญาณเพียง 2 เส้น โดยเส้นหนึ่งเป็นข้อมูล(SDA)และอีกเส้นเป็นสายสัญญาณนาฬิกา(SCL) โดยจะมีการต่ออุปกรณ์บนบัสดังรูปที่ 2.7



รูปที่ 2.7 การต่ออุปกรณ์บนระบบบัส I²C

ที่ขาทั้ง SDA และ SCL ต้องต่อตัวต้านทานพูลอัพกับแรงดัน+5 โวลต์ ไว้เพื่อให้สายมีลอจิกสูงในขณะที่ไม่มีการใช้งานและเพื่อลดสัญญาณรบกวนสถานะที่เกิดขึ้นบนบัส I²C

มีด้วยกัน 5 สถานะดังนี้

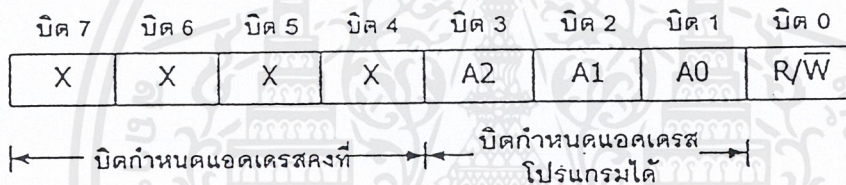
1. บัสว่าง เกิดเมื่อลอจิกบน SDA และ SCL เป็นสูงทั้งคู่หมายถึงการถ่ายทอดข้อมูลสามารถเริ่มได้
2. เริ่มต้นการถ่ายทอดข้อมูล เกิดเมื่อสาย SDA เปลี่ยนจากลอจิกสูงไปต่ำในขณะที่สาย SCL มีลอจิกสูง
3. หยุดการถ่ายทอดข้อมูล เกิดเมื่อสาย SDA เปลี่ยนจากลอจิกต่ำไปสูงในขณะที่สาย SCL มีลอจิกสูง

4. ข้อมูลอยู่บนบัส เกิดจากสถานะเริ่มต้นโดยลอจิกที่อยู่บนสาย SDA คือข้อมูลที่ต้องการถ่ายทอดเมื่อสาย SCL เป็นลอจิกสูงโดยข้อมูลที่ส่งอาจจะเปลี่ยนแปลงแต่ต้องอยู่ในสถานะที่ SCL มีลอจิกต่ำ

5. รับรู้ข้อมูล เกิดหลังจากมีการส่งข้อมูลอย่างสมบูรณ์โดยตัวส่งจะส่งข้อมูลเป็นลอจิกสูงมา 1 บิตเป็นบิตรับรู้หลังจากส่งข้อมูลครบถ้วนอุปกรณ์มาสเตอร์จะส่งสัญญาณรับรู้พิเศษที่สัมพันธ์กับสัญญาณนาฬิกาเพื่อตอบสนองตัวส่ง และทางด้านตัวรับจะส่งบิตรับรู้ที่มีลอจิกต่ำลงบนบัสการทำงานบนบัส I²C

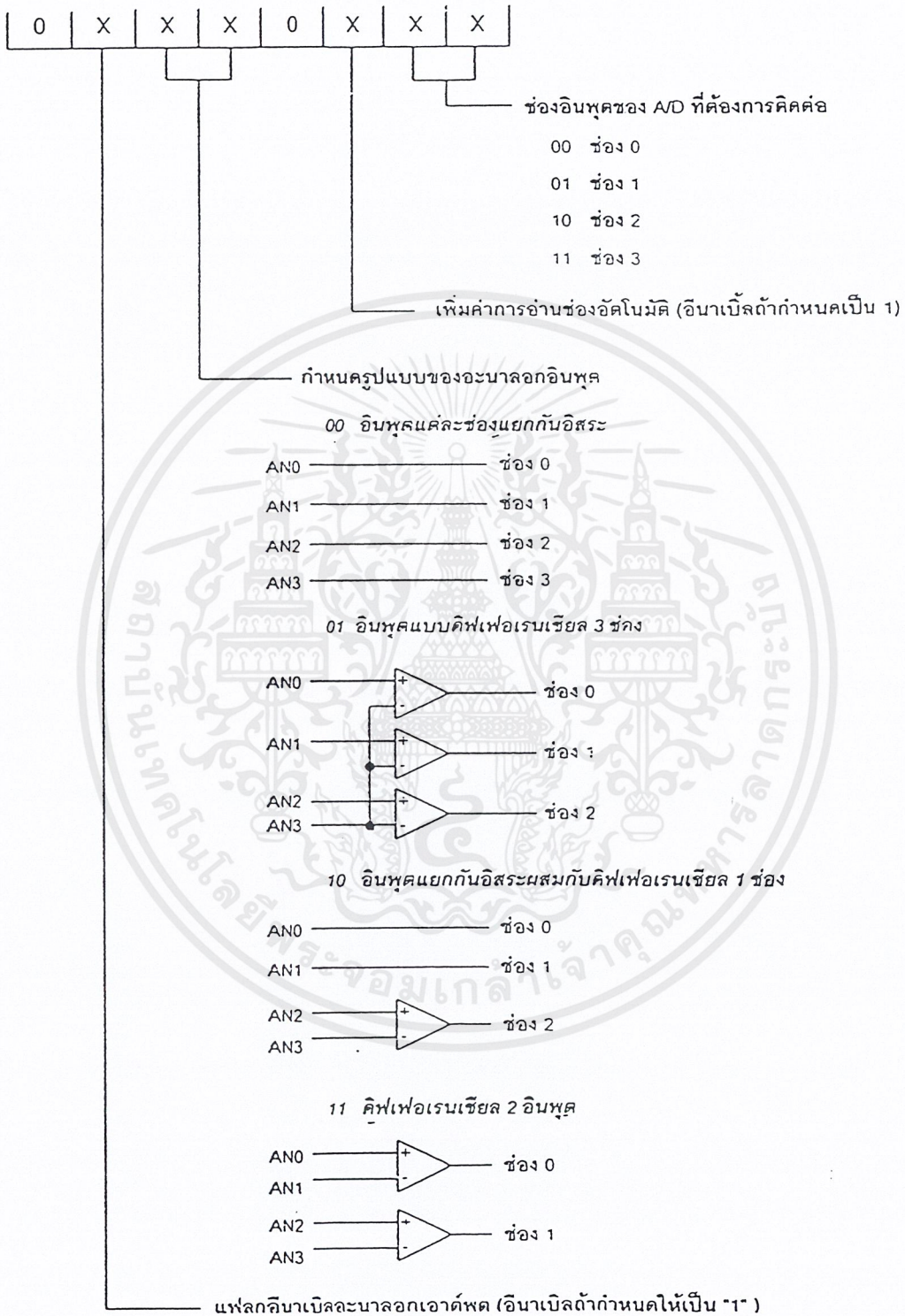
ก่อนการส่งข้อมูลจะต้องมีการอ้างตำแหน่งของอุปกรณ์บนบัสก่อนโดยมีแบบ 7 บิต(มีอุปกรณ์น้อย)และ 10 บิต(มีอุปกรณ์มาก) ซึ่งในที่นี้จะขอพูดถึงแค่แบบ 7 บิต

การอ้างถึงแบบ 7 บิต มีรูปแบบดังรูปที่ 2.8



รูปที่ 2.8 รูปแบบของการกำหนดแอดเดรสแบบ 7 บิต

บิตแอดเดรสคงที่เป็นค่าของอุปกรณ์ซึ่งผู้ผลิตกำหนดมาไม่สามารถแก้ไขได้ ส่วน 3 บิตถัดมาเป็นบิตกำหนดแอดเดรสที่กำหนดได้ ส่วน LSB เป็นบิตกำหนดการอ่านหรือเขียน โดย “0” หมายถึงต้องการเขียนข้อมูลไปยังอุปกรณ์นั้น ส่วนข้อมูลในไบต์ถัดมา คือ ข้อมูลควบคุมโดยในอุปกรณ์แต่ละตัวจะกำหนดข้อมูลควบคุมแตกต่างกันไป ซึ่งในที่นี้จะยกมา แค่ ADC เบอร์ PCF8591 ซึ่งเป็นดังรูปที่ 2.9



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้ภายในหน่วยงานเท่านั้น ไม่ควรนำออกเผยแพร่โดยไม่ได้รับอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

และข้อมูลในไบต์ต่อมา คือข้อมูลที่ใช้อย่างทอคงจริง (Data)

2.3 ไมโครคอนโทรลเลอร์ 8051

ไมโครคอนโทรลเลอร์ 8051 เป็น ซีพียูในตระกูล MCS – 51 ของบริษัทอินเทลถูกผลิตขึ้นมาให้มีคุณสมบัติเหมาะสมกับงานด้านระบบการควบคุมต่างๆ ในลักษณะที่เรียกว่า Single Chip Microcontroller หรือ ไมโครคอนโทรลเลอร์ชิปเดียว

2.3.1 คุณสมบัติของ 8051

- เป็นไมโครโพรเซสเซอร์ขนาด 8 บิต สำหรับงานควบคุมระบบต่างๆ
- มีความสามารถประมวลผลของลอจิกระดับบิต
- มีขนาดของหน่วยความจำสำหรับโปรแกรมทำงานได้ถึง 64 Kbytes (Program Memory)
- มีขนาดของหน่วยความจำสำหรับเก็บข้อมูลได้ถึง 64 Kbytes (Data Memory)
- มีหน่วยความจำสำหรับความจำภายในขนาด 4,8 Kbytes
- มีพอร์ตสำหรับควบคุม 4 พอร์ต และสามารถอ้างอิงพอร์ตได้ระดับบิตต่อบิต
- มีชุด Time/Counter ขนาด 16 บิต 2,3 ชุด
- มี Full duplex UART
- มีโครงสร้างการอินเตอร์รัพท์จาก 6 แหล่งกำเนิดสัญญาณ และ 5 ตำแหน่งโปรแกรม ทำงานตอบรับการอินเตอร์รัพท์โดยสามารถจัดระดับความสำคัญได้ 2 ระดับ
- มีแหล่งกำเนิดความถี่อ้างอิงการทำงานในตัว

โครงสร้างพื้นฐานของไมโครคอนโทรลเลอร์ 8051 แสดงในรูปที่ 2.10

2.3.2 รายละเอียดของขาสัญญาณ 8051

Vcc : ขาแหล่งจ่ายไฟฟ้า (+5 V)

Vss : ขากราวนค์

PO : เป็นขาพอร์ต 0 ของ 8051 ที่มีขนาด 8 บิต ชนิด 2 ทิศทางโดยแต่ละสัญญาณสามารถต่อพ่วงอุปกรณ์ TTL แบบ LS ได้ 8 ตัว และเป็นขาให้สัญญาณ Multiplex ระหว่างสัญญาณข้อมูลกับแปดบิตล่างของสัญญาณแอดเดรส ในกรณีที่ใช้นี้หน่วยความจำภายนอก

P1 : เป็นขาพอร์ต 1 ของ 8051 ขนาด 8 บิต ชนิด 2 ทิศทางแบบ Quasi-Bidirectional โดยที่ถ้าต้องการให้พอร์ตเสกนใดเป็นอินพุตจะให้ค่า 1 ที่บิตนั้น และสามารถต่อพ่วงกับอุปกรณ์ TTL แบบ LS ได้ 4 ตัว

P2 : เป็นขาพอร์ต 2 ของ 8051 ขนาด 8 บิต ชนิด 2 ทิศทางแบบ Quasi-bidirectional เช่นเดียวกับพอร์ต 1 นอกจากนี้ พอร์ต 2 นี้ยังทำหน้าที่ให้สัญญาณแอดเดรส 8 บิตบนในกรณีที่ใช้นี้หน่วยความจำภายนอก ในกรณีอ้างแอดเดรส หน่วยความจำขนาด 16 บิตคั้งนั้นขณะที่ใช้นี้หน่วยความจำภายนอก จะต้องไม่มีการเขียนข้อมูลใดๆ ไปที่พอร์ต 2 จะทำให้เกิดความผิดพลาดในการทำงานได้

P3 : เป็นขาพอร์ต 3 ของ 8051 ขนาด 8 บิต ชนิดสองทิศทางแบบ Quasi-bidirectional เช่นเดียวกับพอร์ต 1 และ พอร์ต 2 แต่พอร์ต 3 นี้จะมีหน้าที่พิเศษคั้งตารางข้างล่างคั้งนั้นเมื่อมีการใช้สัญญาณคั้งกล่าว จึงไม่ควรเขียนข้อมูลไปที่พอร์ต 3 จะทำให้การทำงานของ 8051 ผิดพลาดได้

ขาพอร์ต	หน้าที่พิเศษ
P3.0	3 RxD (สำหรับรับข้อมูลแบบอนุกรม)
P3.1	TxD (สำหรับส่งข้อมูลแบบอนุกรม)
P3.2	INT0 (ขาอินเทอร์รัพท์ภายนอก 1)
P3.3	INT1 (ขาอินเทอร์รัพท์ภายนอก 0)
P3.4	T0 (ขาอินพุตของ Timer 0)
P3.5	T1 (ขาอินพุตของ Timer 1)
P3.6	WR (สำหรับสัญญาณเขียนหน่วยความจำข้อมูลภายนอก)
P3.7	RD (สำหรับสัญญาณอ่านหน่วยความจำข้อมูลภายนอก)

ตารางที่ 2.3 แสดงหน้าที่พิเศษของพอร์ต 3

RST : ขาดำหรับรีเซ็ตการทำงานของ 8051 โดยการให้ลอจิก 1 อย่างน้อย 2 ช่วง Machine Cycle

ALE : ขาสัญญาณออกของ แอคเตอเรต Latch Enable สำหรับ Latch ค่า แอคเตอเรต 8 บิตที่ได้จากพอร์ต 0 โดยจะมีความถี่ออกมาที่ 1/6 ของความถี่อ้างอิงของ 8051

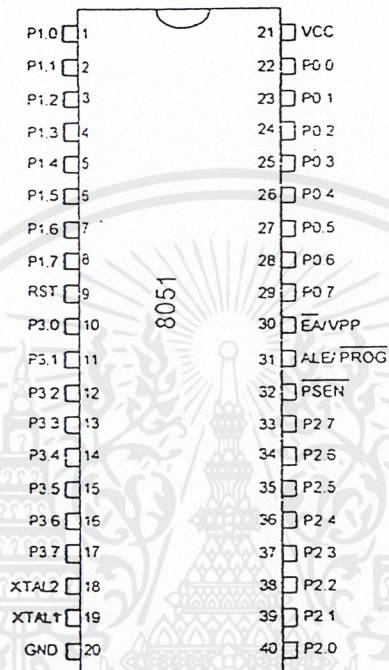
PSEN : ขาสัญญาณ Program Store Enable ใช้สำหรับหน่วยความจำโปรแกรมภายนอก

EA : ขาสัญญาณ External Access Enable สำหรับกำหนดให้ 8051 ให้อ่านหน่วยความจำโปรแกรมภายในหรือภายนอก โดยที่ให้ลอจิก 1 จะอ่านหน่วยความจำโปรแกรมภายใน 8051

XTAL1: ขาเข้าของวงจรกำเนิดความถี่อ้างอิงภายในของ 8051

XTAL2: ขาออกของวงจรกำเนิดความถี่อ้างอิงภายในของ 8051

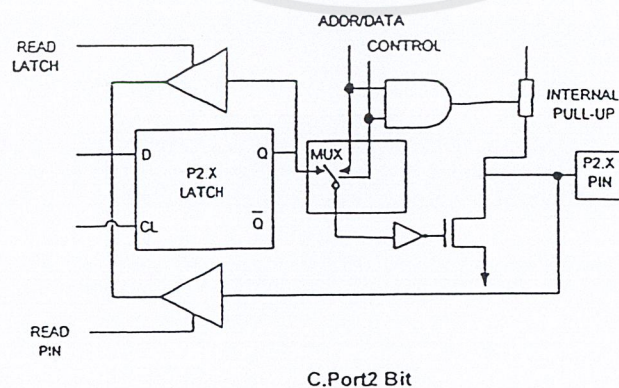
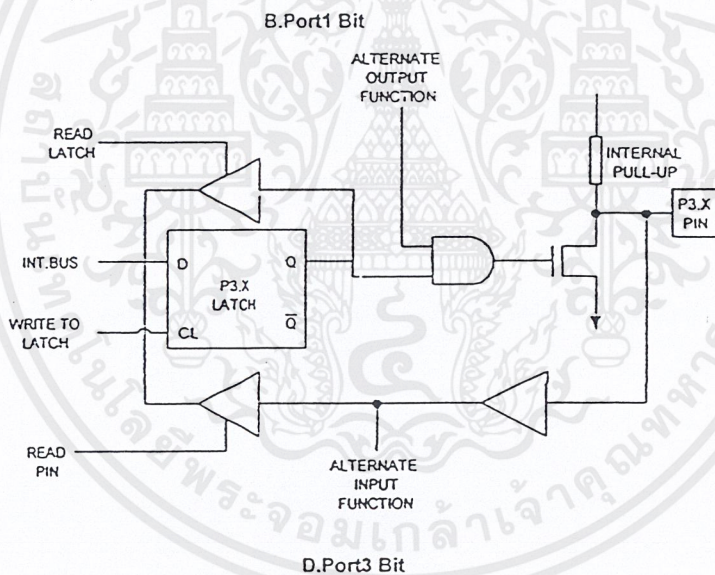
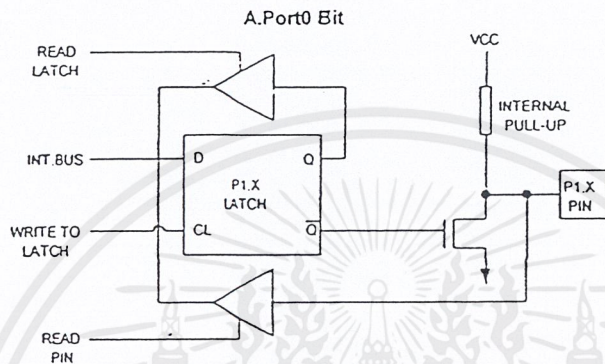
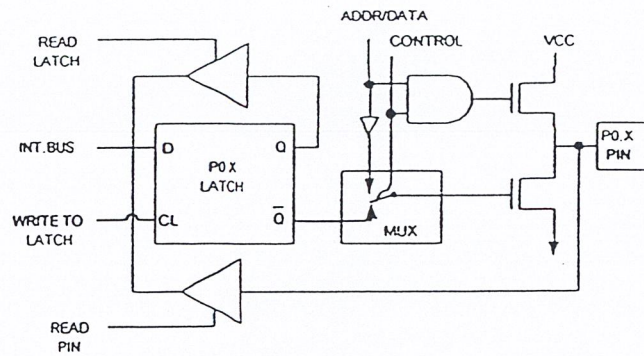
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.11 แสดงขาของ 8051

2.3.3 โครงสร้างภายในของพอร์ต 8051

ในรูปที่ 2.12 แสดงโครงสร้างภายในของพอร์ตแต่ละพอร์ตของ 8051



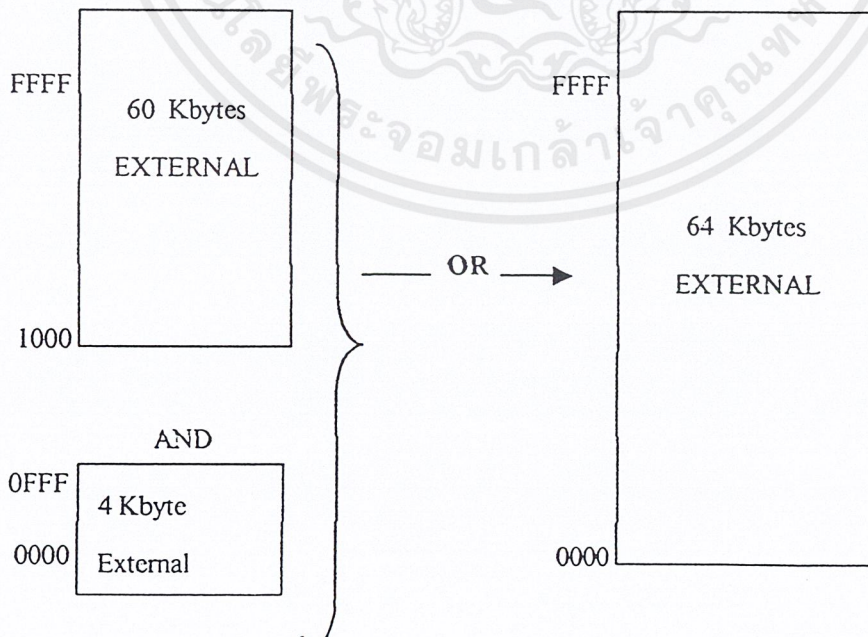
รูปที่ 2.12 โครงสร้างภายในพอร์ตอินพุต/เอาต์พุตของ 8051

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จะเห็นว่าพอร์ต 0 จะเป็นพอร์ตแบบสองทิศทางอย่างแท้จริง โดยที่ข้อมูลที่ส่งออกและอ่านเข้าจะไม่รบกวนกันแต่โครงสร้างของพอร์ต 1, พอร์ต 2 และ พอร์ต 3 จะเป็นเอาท์พุตให้เฟสเพียงตัวเดียว ดังนั้นข้อมูลที่ส่งออกไปโดยเฉพาะลอจิก 0 จะรบกวนข้อมูลที่อ่านได้ ดังนั้นจึงไม่สามารถทำงานแบบสองทิศทางอย่างแท้จริง โดยที่ถ้าต้องการจะใช้บิตหนึ่งของพอร์ตดังกล่าวเป็นอินพุตจะต้องเขียนลอจิก 1 ไปที่บิตดังกล่าว

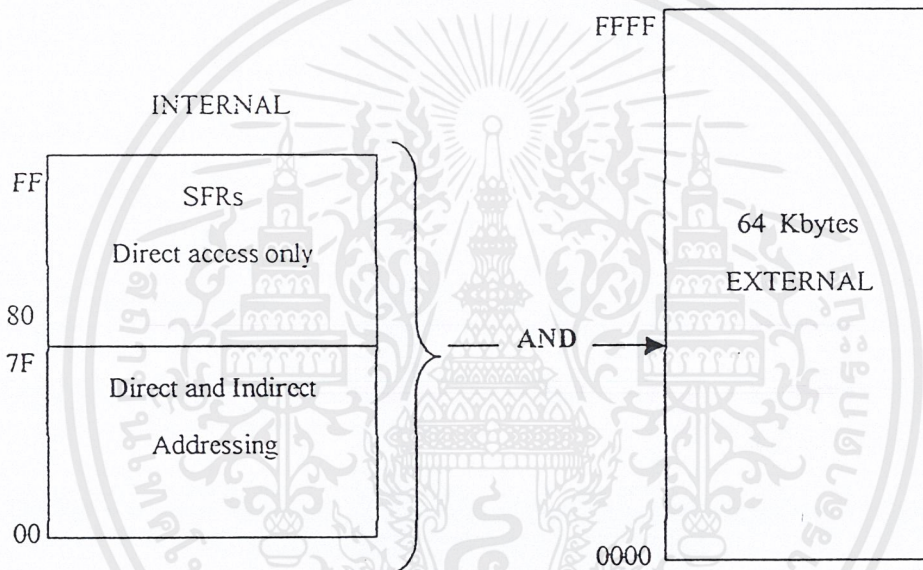
2.3.4 โครงสร้างของหน่วยความจำของ 8051

ดังที่กล่าวมาแล้ว 8051 จะแบ่งหน่วยความจำออกเป็นสองส่วน ได้แก่ หน่วยความจำสำหรับโปรแกรมและหน่วยความจำสำหรับเก็บข้อมูล โดยที่ขนาดของแต่ละส่วนเท่ากับ 64 Kbytes ในส่วนของหน่วยความจำโปรแกรมจะเป็นหน่วยความจำสำหรับอ่านเพียงอย่างเดียว โดยที่ 8051 จะใช้สัญญาณ PSEN ในการอ่านเท่านั้น แต่หน่วยความจำข้อมูลของ 8051 จะสามารถอ่านและเขียนโดยใช้สัญญาณ RD และ WR ตามลำดับ แต่อย่างไรก็ตาม ผู้ใช้สามารถรวมหน่วยความจำโปรแกรมและหน่วยความจำของข้อมูลเข้าด้วยกันได้ โดยนำสัญญาณ RD และ PSEN มา AND กัน สำหรับสร้างสัญญาณในการอ่านหน่วยความจำ นอกจากนี้หน่วยความจำดังกล่าวยังแบ่งออกเป็นภายนอกและภายในของ 8051 ดังแสดงในรูปที่ 2.13 ในด้านซ้ายมือเป็นส่วนของหน่วยความจำโปรแกรมภายในที่มีขนาด 4 Kbytes ของ 8051 ส่วนที่เหลือจะเป็นหน่วยความจำภายนอก ส่วนด้านขวามือแสดงหน่วยความจำโปรแกรมเมื่อเลือกให้ติดต่อหน่วยความจำภายนอกทั้งหมด



รูปที่ 2.13 แสดงหน่วยความจำโปรแกรมของ 8051

สำหรับหน่วยความจำข้อมูลของ 8051 สามารถแบ่งออกเป็นภายนอกและภายใน โดยหน่วยความจำภายนอกแสดงไว้ด้านขวามือของรูปที่ 2.14 ซึ่งมีขนาด 64 Kbytes ส่วนหน่วยความจำข้อมูลภายในแสดงไว้ด้านซ้ายมือของรูปที่ 2.14 โดยหน่วยความจำภายในของ 8051 แบ่งออกได้เป็นสองส่วนได้แก่ ส่วนหน่วยความจำข้อมูลที่สามารถอ้างอิงแบบ Direct และ Indirect ซึ่งมีขนาด 128 Kbytes กับหน่วยความจำที่อ้างอิงได้เฉพาะแบบ Direct เท่านั้น หรือในส่วนที่เรียกอีกแบบหนึ่งว่า SFR (Special Function Register) โดยจะแบ่งกล่าวได้ดังนี้



รูปที่ 2.14 แสดงหน่วยความจำข้อมูลของ 8051

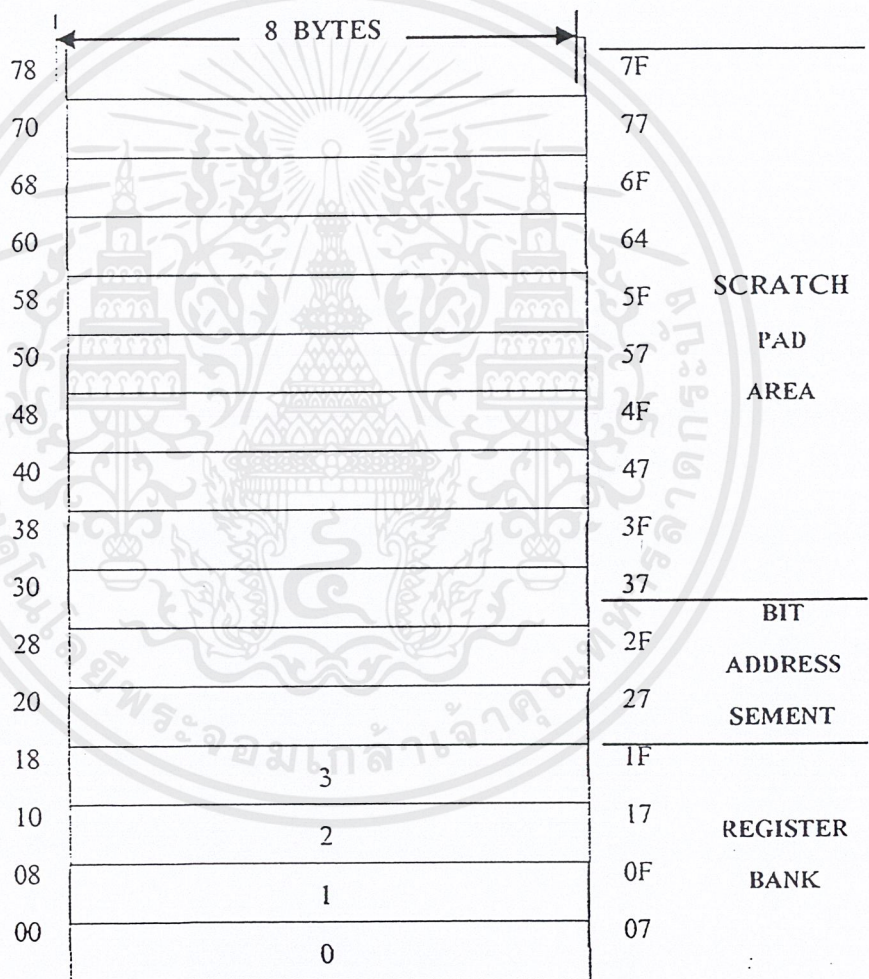
ในส่วนของหน่วยความจำข้อมูลภายในที่อ้างอิงแบบ Direct และ Indirect นั้นจะสามารถแบ่งออกได้ 3 ส่วน ดังแสดงในรูปที่ 2.15 โดยมีรายละเอียดดังนี้

- ส่วนที่ 1 เรียกว่า Register Banks 0-3 ซึ่งอยู่ที่ตำแหน่งหน่วยความจำข้อมูลภายใน ตั้งแต่ 00H ถึง 1FH จำนวน 32 byte โดยจะแบ่งออกเป็นชุด ชุดละ 8 byte จำนวน 4 ชุด ซึ่งแต่ละชุดจะมีชื่อเรียกเป็น R0 ถึง R7 เป็น Register ที่ใช้งานร่วมในแต่ละคำสั่ง โดยเมื่อ 8051 ถูกรีเซ็ต Register Banks 0 จะถูกเลือก
- ส่วนที่ 2 เรียกว่า Bit แอดเดรส Area ซึ่งมีขนาด 16 byte ที่ตำแหน่งหน่วยความจำข้อมูล 20H ถึง 2FH ในส่วนนี้สามารถใช้อ้างอิงถึงข้อมูลได้เป็น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระดังบิตถึง 128 บิต โดยการอ้างอิงตำแหน่งโดยตรงในลักษณะบิต ตั้งแต่ตำแหน่ง 00H ถึง 7FH

- ส่วนที่ 3 เรียกว่า Scratch Pad Area จะอยู่ที่ตำแหน่งตั้งแต่ 30H ถึง 7FH ซึ่งเป็นบริเวณหน่วยความจำข้อมูลภายในเอนกประสงค์ที่ผู้ใช้สามารถใช้ได้โดยตรง นอกจากนี้ยังสามารถใช้หน่วยความจำข้อมูลบริเวณนี้สำหรับการเก็บข้อมูลแบบ Stack ได้ด้วย



รูปที่ 2.15 แสดงหน่วยความจำข้อมูลภายใน

ในส่วนของการเข้าถึงข้อมูลภายในที่ใช้อ้างอิงแบบ Direct เพียงอย่างเดียวหรือที่เรียก

ว่า SFR ซึ่งเป็นส่วนสำหรับเก็บหรือกำหนดการทำงานของงานภายในของ 8051 ดังแสดงในรูปที่ 2.16

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

F8								FF
F0	B							F7
E8								EF
E0	ACC							E7
D8								DF
D0	PSW							D7
C8	T2CON		RCAP2H	RCAP2H	TL2	TH2		CF
C0								C7
B8	IP							BF
B0	P3							B7
A8	IE							AF
A0	P2							A7
98	SCON	SBUF						9F
90	P1							97
88	TCON	TMOD	TL0	TL1	TH0	TH1		8F
80	P0	SP	DPL	DPH			PCON	87

รูปที่ 2.16 แสดงรายละเอียดของ Special Function Register

ในส่วนของบริเวณนี้จะมีขนาด 128 byte แต่ในการใช้งานนั้นใช้ได้เฉพาะตำแหน่งซึ่งแสดงไว้ในรูปที่ 2.16 เท่านั้น หากผู้ใช้อ้างตำแหน่งนอกเหนือจากนี้นั้นจะได้ข้อมูลที่คาดเดาไม่ได้ โดยแต่ละตำแหน่งจะมีหน้าที่ดังนี้

ACC : เป็น Accumulator ซึ่งเป็นรีจิสเตอร์สำหรับการประมวลผลทางลอจิกและคณิตศาสตร์ โดยผู้ใช้สามารถอ้างอิงได้ในรูปแบบของ byte หรือระดับบิตได้

B : เป็นรีจิสเตอร์พิเศษสำหรับใช้กับคำสั่งในการคูณหรือหาร นอกจากนี้ยังใช้เป็นรีจิสเตอร์สำหรับเก็บพักข้อมูลได้

เอกสารนี้เป็นเอกสารสงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- PSW : เป็นรีจิสเตอร์ Program Status Word จะแสดงสถานะการทำงาน
ของ 8051 สำหรับการตรวจสอบได้ ซึ่งจะอธิบายรายละเอียดภายหลัง
- SP : เป็นรีจิสเตอร์สำหรับชี้หน่วยความจำข้อมูลสำหรับการเก็บแบบ
Stack
- DPTR : เป็นรีจิสเตอร์ขนาด 16 บิต โดยแบ่งเป็น 8 บิตบนและ 8 บิตล่าง
ใช้สำหรับชี้ตำแหน่งของหน่วยความจำข้อมูลภายนอกหรือสำหรับการอ่านตารางข้อมูล
ของหน่วยความจำโปรแกรมภายใน
- P0 : เป็นรีจิสเตอร์สำหรับพอร์ต 0 ของ 8051
- P1 : เป็นรีจิสเตอร์สำหรับพอร์ต 1 ของ 8051
- P2 : เป็นรีจิสเตอร์สำหรับพอร์ต 2 ของ 8051
- P3 : เป็นรีจิสเตอร์สำหรับพอร์ต 3 ของ 8051
- IP : เป็นรีจิสเตอร์สำหรับกำหนดลำดับความสำคัญของการอินเทอร์รัพต์
ของ 8051
- IE : เป็นรีจิสเตอร์สำหรับการจะรับหรือไม่รับการอินเทอร์รัพต์ของ
8051
- TMOD : เป็นรีจิสเตอร์สำหรับควบคุมหน้าที่ของ 8051
- TCON : เป็นรีจิสเตอร์สำหรับควบคุมการทำงานของ Timer/Counter
ของ 8051
- T2CON : เป็นรีจิสเตอร์สำหรับ
- TH0 : เป็นรีจิสเตอร์สำหรับเก็บข้อมูลของ Timer/Counter 0 8 บิต
บนของ 8051
- TL0 : เป็นรีจิสเตอร์สำหรับเก็บข้อมูลของ Timer/Counter 0 8 บิต
ล่างของ 8051
- TH1 : เป็นรีจิสเตอร์สำหรับเก็บข้อมูลของ Timer/Counter 1 8 บิต
บนของ 8051
- TL1 : เป็นรีจิสเตอร์สำหรับเก็บข้อมูลของ Timer/Counter 1 8 บิต
ล่างของ 8051
- TH2 : เป็นรีจิสเตอร์สำหรับเก็บข้อมูลของ Timer/Counter 2 8 บิต
บนของ 8051

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- TL2 : เป็นรีจิสเตอร์สำหรับเก็บข้อมูลของ Timer/Counter 2 8 บิตล่างของ 8051
- RCAP2H : เป็น Capture Register ของ Timer/Counter 2 8 บิตล่างของ 8051
- RCAP2L : เป็น Capture Register ของ Timer/Counter 2 8 บิตบนของ 8051
- SCON : เป็นรีจิสเตอร์สำหรับควบคุมการรับส่งข้อมูลแบบอนุกรมของ 8051
- SBUF : เป็นรีจิสเตอร์สำหรับเก็บพักข้อมูล ที่ได้จากการรับส่งข้อมูลแบบอนุกรมของ 8051

2.3.5 แฟลชเมโมรีไมโครคอนโทรลเลอร์

AT89C51 เป็นประติรูปกรรมของอุปกรณ์ไมโครคอนโทรลเลอร์ ที่มีหน่วยความจำแบบแฟลช อยู่ภายในตัว 4 Kbytes ที่สามารถใช้เครื่องคอมพิวเตอร์พีซี ทำการลบและเขียนโปรแกรมได้โดยตรง โดยไม่ต้องถอด MCU ออกจากบอร์ดหรือแผงวงจรในลักษณะที่เรียกว่า In System Programming หรือจะใช้เครื่องโปรแกรม (Universal Programmer) โดยตรงก็ได้ ซึ่งสามารถโปรแกรมได้เป็นพันครั้ง

ไมโครคอนโทรลเลอร์ตระกูล AT89C51 สามารถใช้งานร่วมและแทนกันได้กับไมโครคอนโทรลเลอร์ในตระกูล MCS-51 ได้แก่เบอร์ 80C51/52, 87C51/52 เป็นต้น ซึ่ง AT89C51 เหมือนกับตระกูลเหล่านี้ของอินเทล ทำให้ด้านของชุดคำสั่งและการจัดเรียงขา นั้นคือเราสามารถนำ AT89C51 มาใช้งานแทน MCS-51 ได้เลย

คุณสมบัติที่สำคัญของ AT89C51

- ขนาดหน่วยความจำแบบแฟลช 4 Kbytes
- ขนาดหน่วยความจำ Ram 128 ไบต์
- ชาติ่อใช้งาน 40 ขา
- พอร์ตอินพุต/เอาต์พุต 32 I/O
- ตัวถังบรรจุ PDIP
- ย่านแรงดันไฟฟ้าเลี้ยง 2.7 – 6.0 V.
- ย่านความถี่การทำงาน 0 – 24 MHz

2.4 การติดต่อสื่อสารผ่านพอร์ตอนุกรม

ลักษณะของการส่งข้อมูลแบบอนุกรมนั้น ข้อมูลจะถูกส่งออกมาทีละบิต จากอุปกรณ์ส่งไปยังอุปกรณ์รับ ช่องสัญญาณในการส่งข้อมูลอาจใช้เพียง 1 หรือ 2 ช่องสัญญาณเท่านั้น ทำให้ค่าใช้จ่ายในการสื่อสารถูกกว่าแบบขนาน แต่อัตราการส่งข้อมูลจะล่าช้ากว่าการส่งแบบขนาน ในการส่งข้อมูลแบบอนุกรม ข้อมูลที่ต้องการส่งจะอยู่ในลักษณะเป็นไบนารีจะถูกทยอยส่งทีละบิต และทางอุปกรณ์รับจะต้องรับข้อมูลเข้ามาทีละบิต แล้วมารวมกันเป็นไบนารีซึ่งทางด้านอุปกรณ์รับจะต้องคอยตรวจสอบว่า บิตใดเป็นบิตเริ่มแรกของไบนารีนั้น การตรวจสอบขึ้นอยู่กับรูปแบบของรหัสของบิตที่ใช้ ซึ่งในการรับส่งข้อมูลแบบอนุกรมระหว่างไมโครคอมพิวเตอร์กับอุปกรณ์ภายนอกนั้น จำเป็นต้องมีมาตรฐานในการรับส่ง ซึ่งมาตรฐานที่นิยมมากที่สุดก็คือ มาตรฐาน RS-232

มาตรฐาน RS-232

เพื่อที่จะทำให้อุปกรณ์จากผู้ผลิตต่างกันทำงานร่วมกันได้ มาตรฐานหลายชนิดจึงได้รับการออกแบบขึ้น มาตรฐานที่ใช้กันอย่างกว้างขวางที่สุดคือ RS-232C ซึ่งโดยปกติไมโครคอมพิวเตอร์จะมีพอร์ตที่เป็นอนุกรมอยู่ในตัวอยู่แล้ว และจะทำหน้าที่รับส่งข้อมูลในแบบอนุกรมตามจุดประสงค์ของมาตรฐาน RS-232C นั้นเพื่อจะสามารถเชื่อมต่อกันระหว่างอุปกรณ์รับส่งข้อมูลปลายทาง (Data Terminal Equipment : DTE) เช่น พอร์ตของคอมพิวเตอร์หลักหรืออุปกรณ์ปลายทาง กับอุปกรณ์สื่อสารข้อมูล (Data Communication Equipment : DCE) หมายถึงอุปกรณ์ที่สามารถแปลงรูปคลื่นดิจิทัลไปเป็นสัญญาณที่เหมาะสมสำหรับการส่งผ่านสายโทรศัพท์ หรือ ตัวกลางการสื่อสารอย่างอื่น โดยกระบวนการที่เรียกว่า มอดูเรชั่น (Modulation)

การอินเตอร์เฟสตามมาตรฐาน RS-232

มาตรฐาน RS-232C ใช้สายสัญญาณเพียงเส้นเดียวในการส่งสัญญาณ โดยสัญญาณที่ส่งส่งไปได้ทิศทางเดียว สำหรับความเร็วและระยะทางของการเชื่อมต่อ RS-232C สามารถเชื่อมต่อการถ่ายข้อมูลได้จาก 0 - 20 K bps (บิตต่อวินาที) ซึ่งเพียงพอสำหรับไมโครคอมพิวเตอร์ที่มีขนาดอัตราบอด (Baud Rate) 110 ถึง 9600 ความยาวของสายเชื่อมต่อสัญญาณระหว่าง DTE และ DCE โดยสัญญาณตามมาตรฐานของ RS-232 จำกัดอยู่แค่ 50 ฟุต หรือประมาณ 15 เมตร ซึ่งเพียงพอสำหรับการสื่อสารไมโครคอมพิวเตอร์กับอุปกรณ์ภายนอก สำหรับแรงดันของระดับสัญญาณจะถูกกำหนดลงในสองบริเวณคือ

สำหรับ MCS 8051 สามารถส่งและรับข้อมูลอนุกรมแบบ Universal Asynchronous Receiver Transmitter (UART) แบบ Full Duplex ที่สามารถเลือกรูปแบบในการรับส่งได้ถึง 4 แบบ

The diagram illustrates a serial data frame. It begins with a 'Start bit' (represented by a high-to-low transition). This is followed by eight data bits, labeled 0 through 7, each in its own box. Above the data boxes are two inverted triangles (▽) indicating the data period. Below the data boxes is the label 'DATA'. The frame ends with a 'Stop bit' (represented by a low-to-high transition), indicated by an upward triangle (△) below the line.

โดยข้อมูลที่รับส่ง 1 ชุด มีขนาด 10 บิต

- | | | |
|-------------|---|-----|
| - Start bit | 1 | บิต |
| - DATA bit | 8 | บิต |
| - Stop bit | 1 | บิต |

โดยอัตราความเร็วในการรับส่งข้อมูล คำนวณได้จากสมการ

$$\text{Baud rate} = \frac{2^{\text{SMOD}}}{32} \times \frac{\text{Oscillator Frequency}}{12 \times [256 - (\text{TH } 1)]}$$

จากโครงงานได้กำหนดใช้ Timer 1 ทำงานใน Mode 2 และความเร็วระบบเท่ากับ 11.059 MHz บิต SMOD = 0 และ Register TH 1 = FDH, (253) D

$$\text{Baud rate} = \frac{2^0}{32} \times \frac{11.059 \text{ M}}{12 \times [256 - 253]}$$

$$\text{Baud rate} = 9600 \text{ b/s}$$

ในการทำงานในการส่งและรับข้อมูล และอัตราความเร็วในการส่งข้อมูล จะต้องทำการใช้งาน SFR ต่างๆ ดังนี้

- **SCON (Serial Port Control Register)**

SCON เป็น Register เพื่อควบคุมการส่งและรับข้อมูลแบบ Serial Port ในที่นี้ ได้กำหนดให้การรับส่งข้อมูลทำงานใน MODE 1 โดยข้อมูล 1 ชุด มีขนาด 10 บิต ซึ่งประกอบด้วย Start bit 1 bit (Logic 0), Data bit 8 bit และ Stop bit 1 bit (Logic 1) และเป็น Register เพื่อให้มีการรับข้อมูลเข้ามาทาง RXD

- **TMOD (Timer/Counter Mode Register)**

TMOD เป็น Register ที่ทำหน้าที่ควบคุมการทำงานของ Timer หรือ Counter ในการทำงานนี้ได้กำหนดการใช้งานให้ใช้ Timer 1 และเลือกการทำงานของ Timer 1 ใน Mode 2 โดยทำงานแบบ “Reload” และเป็น Register ที่เป็นตัวกำหนดให้ วงจรในการรับส่งทำงาน

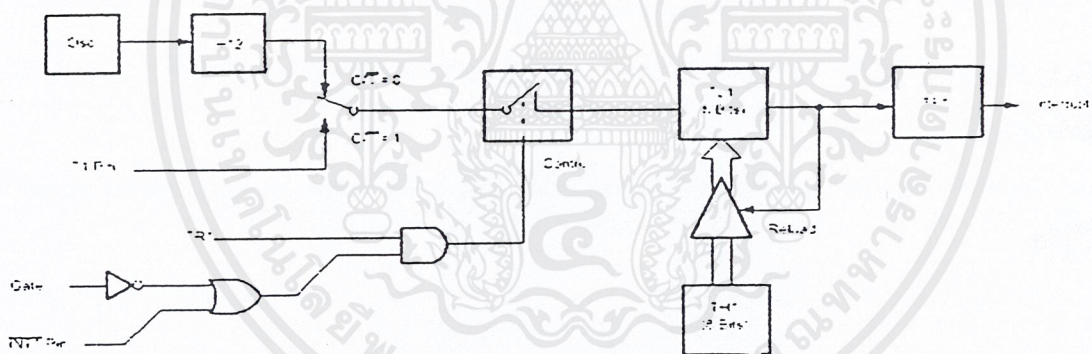
- **TCON (Timer/Control Register)**

TCON เป็น Register เพื่อใช้ตรวจสอบสถานะการทำงานของ Timer ในที่นี้ได้ใช้ Register นี้ ในการตรวจสอบว่ามีข้อมูลเข้ามาทาง RXD หรือไม่ และใช้ตรวจสอบในการส่งข้อมูล

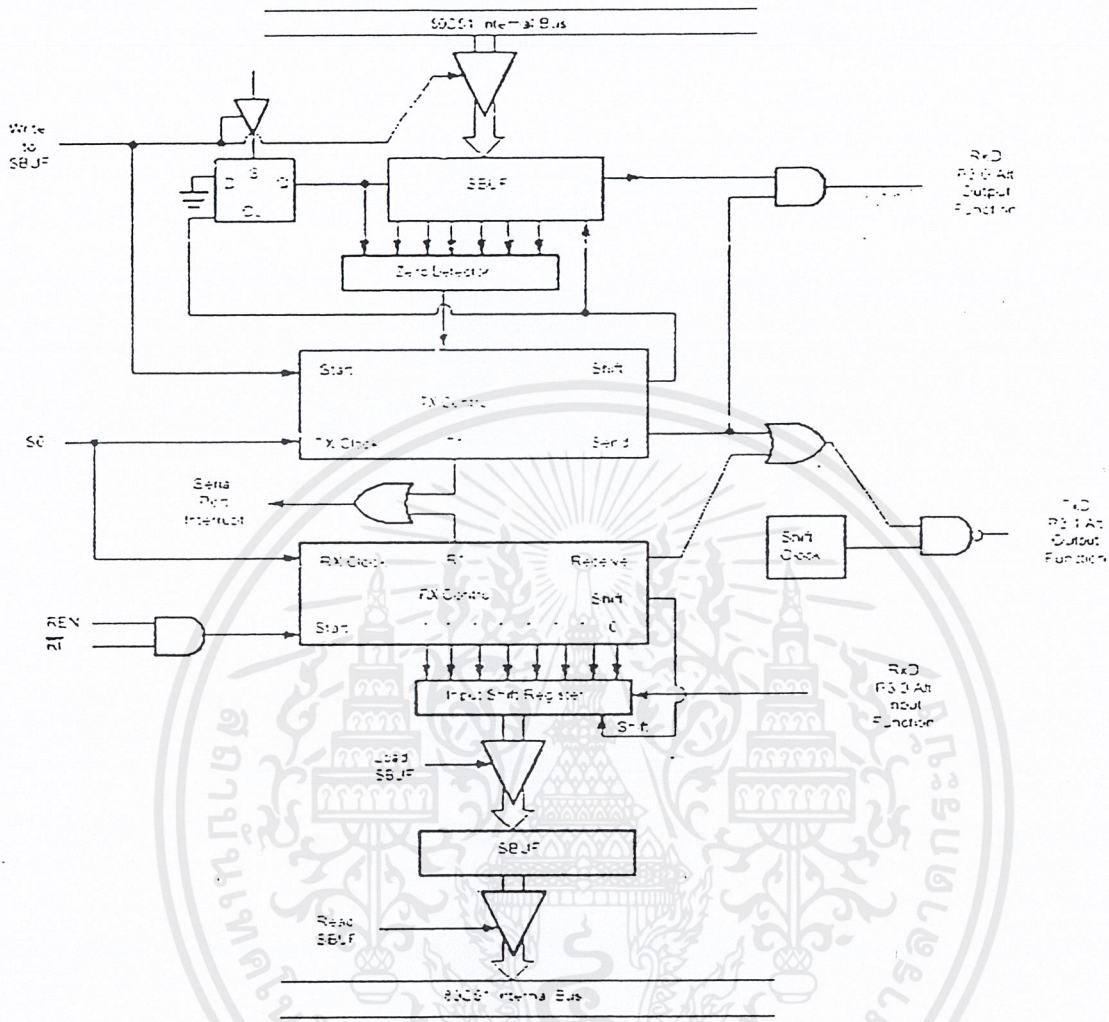
Register ทั้งหมดที่กล่าวมานี้จะต้องทำงานที่สอดคล้องกันหมด จึงจะทำให้การรับส่งข้อมูลแบบ Universal Asynchronous Receiver Transmitter (UART) ทำงานในอัตราความเร็ว (Baud Rate) ที่กำหนด

ในรูปที่ 2.17กเป็น Diagram ของวงจร Timer 1 ใน 8051 ที่ทำงานใน Mode 2 ซึ่งจะใช้ในการกำหนด อัตราความเร็วในการส่งและรับข้อมูล โดยการทำงานต้องกำหนดใน Register SFR.

การกำหนด Baud Rate นั้นได้กำหนดค่าที่ TH 1 และวงจรทำงานใน Mode Timer 1 โดยกำหนดที่ขา C/T โดยความถี่ OSC จะถูกวงจรหาร 12 มารออยู่ที่ Control ดังนั้นการ START ให้วงจรทำงาน ก็ขึ้นอยู่กับ Output ของ AND Gate ที่มี Logic “1” โดยถูกควบคุมด้วย THR 1 และ Gate หรือ INT



รูปที่ 2.17.ก Timer Mode 2



รูปที่ 2.17.ข Serial Port Mode 1

การทำงานโหมด 1

การส่งข้อมูล

จากรูปที่ 2.19 บิต SMOD จะเป็นตัวเลือกว่า สัญญาณ Timer 1 Overflow ที่ส่งไปยังวงจรหาร 16 จะถูกหาร 2 ก่อนหรือไม่ ถ้า SMOD เป็น 1 สัญญาณ Timer 1 Overflow จะไม่ถูกหารแต่ถ้า SMOD เป็น 0 สัญญาณ Timer1 Overflow จะถูกหาร 2 ก่อนที่จะเข้าวงจรหาร 16 การส่งข้อมูลจะเริ่มจากการที่ไมโครคอนโทรลเลอร์เขียนข้อมูลไปยังรีจิสเตอร์ SBUF และทำให้เอาต์พุตของ D FLIP FLOP ทางซ้ายของ SBUF มีค่าเป็น 1 และเป็นบิตที่ 9 ของการส่งข้อมูล สัญญาณ Write to SBUF ยังส่งไปยัง TX Control ด้วย ขณะนี้ข้อมูลในวงจรหาร 16 มีค่าเป็นอะไรไม่ทราบจึงจะรอจนกว่าข้อมูลในวงจรหาร 16 นับเพิ่มขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Overflow จะไม่ถูกหารแต่ถ้า SMOD เป็น 0 สัญญาณ Timer1 Overflow จะถูกหาร 2 ก่อนที่จะเข้าวงจรหาร 16 การส่งข้อมูลจะเริ่มจากการที่มีค่าตั้งเขียนข้อมูลไปยังรีจิสเตอร์ SBUF และทำให้เอาต์พุตของ D FLIP FLOP ทางซ้ายของ SBUF มีค่าเป็น 1 และเป็นบิตที่ 9 ของการส่งข้อมูล สัญญาณ Write to SBUF ยังส่งไปยัง TX Control ด้วย ขณะนี้ข้อมูลในวงจรหาร 16 มีค่าเป็นอะไรไม่ทราบจึงจะรอนกว่าข้อมูลในวงจรหาร 16 นับเพิ่มขึ้นจนถึงค่าสูงสุดแล้ววนกลับเป็น 0 คือเกิดการวนกลับทำให้เริ่มการส่งข้อมูลที่เวลา SIPI ของไซเคิลเครื่องถัดไป (การส่งข้อมูลออกจะสัมพันธ์กับการเกิด Overflow ในวงจรหาร 16) สัญญาณ SEND จาก TX Control เปลี่ยนสถานะลอจิกเป็น 0 แล้วเริ่มส่งข้อมูลที่เป็น Start bit (0) ออกไป เมื่อส่ง Start bit ออกไปแล้ว วงจร TX Control ก็จะทำให้สัญญาณ DATA เป็น 1 เพื่อเลื่อนข้อมูลใน SBUF ออกไป เริ่มจากบิต 0 จนถึง บิตที่ 7 การส่งข้อมูลนี้จะเกิดขึ้นเมื่อสัญญาณ TX Clock เปลี่ยนสถานะจาก 0 เป็น 1 ขณะที่ข้อมูลถูกเลื่อนออกไปนั้น จะมี 0 ถูกเลื่อนเข้ามาทางซ้ายของรีจิสเตอร์ SBUF เมื่อข้อมูลเลื่อนออกไปทั้ง 8 บิต แล้วบิตที่ 9 ซึ่งเป็น 1 และคอนคันอยู่ทางซ้ายสุดจะถูกเลื่อนมาอยู่ในตำแหน่งสุดท้ายทางขวาของรีจิสเตอร์ SBUF และทางซ้ายของหลักนี้จะมี 0 อยู่ทั้ง 8 บิต ใน SBUF ทำให้ Zero Detector รู้ว่าเป็นข้อมูลบิตสุดท้ายแล้วที่ส่งออกโดยจะมีสัญญาณมาบอกกับวงจร TX Control ด้วย เมื่อ TX Control ส่งสัญญาณ Shift ออกไปเป็นการส่งข้อมูลบิตสุดท้าย (บิต 7) ออกไป ก็จะรออีก 1 TX Clock (Bit Clock) ก็จะทำงานให้ขา TXD ส่งข้อมูล Stop bit (1) ออกมา สัญญาณ DATA ซึ่งมีสถานะลอจิกเป็น “ 1 ” มาตั้งแต่เริ่มส่งข้อมูลบิต 0 ก็จะกลับเป็น 0 และบิต TI จะเป็น 1 เพื่อบอกการสิ้นสุดการส่งข้อมูลทั้งหมดจะสิ้นสุดลงเมื่อสัญญาณ TX Clock ไซเคิลที่ 10 นับตั้งแต่สัญญาณ SEND เปลี่ยนสถานะลอจิกเป็น 0

การรับข้อมูล

การรับข้อมูลจะขึ้นกับอัตราการเกิด Overflow ใน Timer แล้วหาร 2 หรือไม่ขึ้นกับค่าของ บิต SMOD สัญญาณนี้จะไปเข้าวงจรหาร 16 และเป็นตัวกำหนดอัตราการรับข้อมูล การรับข้อมูลจะเริ่มจากวงจร 1 – TO – 0 Transmitter Detector พบว่าสัญญาณที่ขา RXD เปลี่ยนจาก 1 เป็น 0 ซึ่งหมายถึงมีข้อมูล Start bit เข้ามา การตรวจสอบนี้จะกระทำด้วยอัตราเดียวกับสัญญาณที่เข้าวงจรหาร 16 เมื่อพบการเปลี่ยนสถานะลอจิกที่ขา RXD ก็ จะเริ่มการรับข้อมูล ขณะนี้จะรีเซ็ตวงจรหาร 16 ให้มีค่าเป็น 0 เพื่อสร้าง RX Clock ให้เข้าจังหวะ (Synchronous) กับข้อมูลที่เข้ามา โดยสัญญาณ RX Clock จะเป็น 1 เมื่อการ

เอกสารนี้เป็นฉบับของวงจรหาร 16 มีค่าเป็น 15 ขณะที่วงจรหาร 16 นั้น นับถึง 7, 8 และ 9 จะมีการตรวจการคำนวณว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สับข้อมูลที่เข้ามาเป็นค่านั้น ถ้าในการตรวจสอบ Start bit แล้วพบว่าผิดพลาด คือไม่เป็น 0 ก็จะมีรีเซ็ต การทำงานเพื่อไปตรวจสอบการเปลี่ยนสถานะจาก 1 เป็น 0 ของข้อมูลที่ขา RXD ใหม่ แต่ถ้าพบ Start bit ก็จะเก็บข้อมูลทั้งหมดที่เข้ามาโดยเลื่อนข้อมูลเข้าไปยัง Input Shift Register ที่มีสัญญาณควบคุมการเลื่อนข้อมูล (Shift) ส่งมาจาก RX Control ในตอนเริ่มต้นการรับข้อมูลเข้าไปทางขาของ Input Shift Register ก็จะมี 1 ถูกเลื่อนออกไปทางซ้ายทุกครั้งที่มีข้อมูลเข้ามา เมื่อ Start bit ที่รับเข้ามาถูกเลื่อนไปถึงซ้ายสุดของ Input Shift Register ก็จะมีสัญญาณไปบอก RX Control Block หลังจากข้อมูลบิตสุดท้ายเข้ามาแล้วก็จะโหลด (Load) เอาข้อมูล 8 บิต ไปเก็บในรีจิสเตอร์ SBUF พร้อมทั้ง Set ค่าในบิต RI และ RB8 ของรีจิสเตอร์ SCON แต่การโหลดข้อมูลไปเก็บนี้จะเกิดขึ้นได้ก็ต่อเมื่อ

1. RI = 0 และ
 2. SM2 = 0 หรือถ้า SM2 = 1 จะต้องได้รับ Stop bit เป็น 1
- ถ้าไม่มีสถานะใดสถานะหนึ่งดังกล่าวแล้ว ข้อมูลที่รับเข้ามาก็จะถูกทิ้งไป คือไม่

โหลดไปเก็บในรีจิสเตอร์ SBUF ถ้ามีสถานะดังกล่าวถูกต้อง Stop bit จะถูกนำไปเก็บในรีจิสเตอร์ RB8 และ บิต RI จะเป็น 1

แต่ไม่ว่าทั้ง 2 กรณี จะเกิดหรือไม่ก็จะกลับไปสู่การตรวจสอบสถานะเปลี่ยนจาก 1 เป็น 0 ที่ขา RXD เพื่อรับข้อมูลต่อไป

ในการรับข้อมูลแบบอนุกรม Mode 1 นี้ อัตราการส่งข้อมูลแต่ละบิต (Baud Rate) จะขึ้นกับอัตราการเกิด Overflow ใน Timer 1 ดังสมการ

$$\text{Baudrate} = \frac{2^{\text{SMOD}}}{X} \times (\text{Timer 1 Overflow Rate})$$

32

ในขณะที่ใช้ Timer 1 เป็นตัวกำหนด Baud Rate นี้จะต้อง Disable ไม่ให้เกิดการขัดจังหวะเนื่องจากการ Overflow Timer 1 อาจใช้ใน Mode ของ Timer หรือ Counter ก็ได้ ซึ่งเมื่อการนับในรีจิสเตอร์ตัวนับมีค่าสูงสุดแล้วกลับมาเป็น 0 ก็เกิด Overflow เช่นเดียวกัน แต่โดยปกติแล้วจะใช้ Timer 1 ใน Mode ของ Timer ที่มีการทำงานแบบ Auto Reload Mode 2 เพื่อว่าเมื่อค่าในการนับโดยรีจิสเตอร์ TL1 ถึงค่าสูงสุดก็จะโหลดค่าในรีจิสเตอร์ TH1 มาไว้ใน TL1 สำหรับเป็นค่าเริ่มต้นการนับต่อไปซึ่ง Baud Rate จะมีค่า

$$\text{Baudrate} = \frac{2^{\text{SMOD}}}{32} \times \frac{\text{Oscillator Frequency}}{12 \times [256 - (\text{TH1})]}$$

โดยที่ SMOD เป็นบิตหนึ่งในรีจิสเตอร์ PCON

เช่นความถี่ของออสซิลเลเตอร์เท่ากับ 11.059 MHz บิต SMOD = 0 รีจิสเตอร์ TH1 มีค่า FDH ,Timer1 ทำงานใน Mode 2 จะได้ว่าอัตราการส่ง – รับข้อมูลแบบอนุกรม

$$= \frac{2^0}{32} \times \frac{11.059 \times 10^6}{12 \times [256 - 253]}$$

$$= \frac{1}{32} \times \frac{11.059 \times 10^6}{12 \times 24}$$

$$= 9600 \text{ บิต/วินาที}$$

2.5 การขับโมดูลแสดงผลแบบผลึกเหลว (LCD)

อุปกรณ์ในปัจจุบันนี้ในส่วนแสดงผลนั้นจะใช้ LCD (Liquid Crystal Display) เสียเป็นส่วนใหญ่ ไม่ว่าจะเป็นเครื่องเล่นวีดีโอ เครื่องวัดคุมต่างๆ เราพอจะแบ่งออกเป็นออกเป็นพวกๆ ได้ดังนี้

1. Character LCD Module
2. Graphic LCD Module
3. Segment Display Type LED Module

โมดูล LCD จะมีส่วนประกอบหลักๆ 3 ส่วนดังนี้

ตัวแสดงผล (Display)

ภายในเป็นผลึกเหลวที่สามารถแสดงผลให้เห็นโดยอาศัยแสงจากภายนอก ดังนั้นจึงต้องมีมุมในการมองข้อมูลที่แสดงผลบนจอ LCD

ตัวควบคุม (Controller)

เป็นตัวรับข้อมูลจากอุปกรณ์ภายนอกมาควบคุมการทำงานของโมดูล LCD เช่น ลบจอภาพ แสดงตัวอักษร หรือเลื่อนเคอร์เซอร์ เป็นต้น ตัวควบคุมนี้ใช้ชิปควบคุมโดยเฉพาะ ชิปที่นิยมใช้คือ

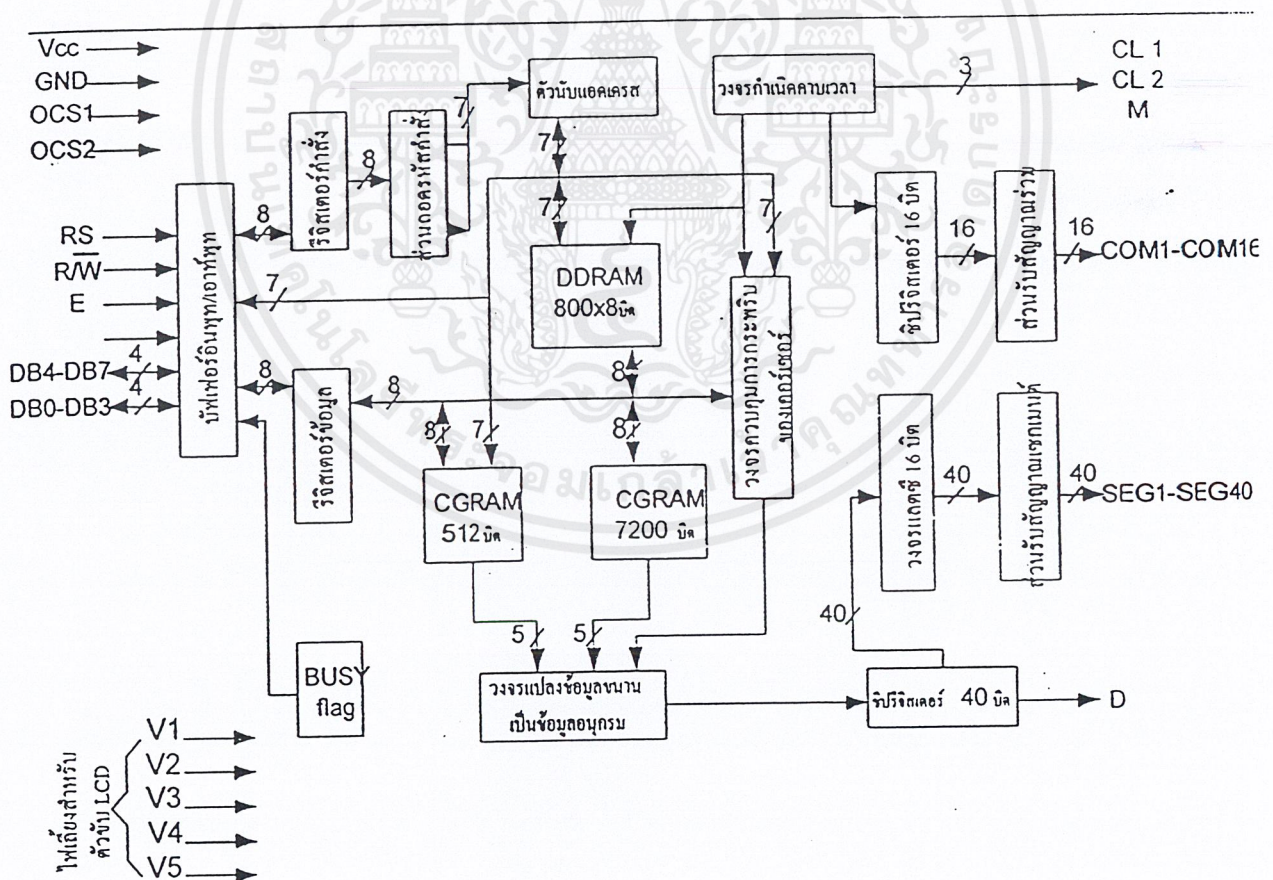
เบอร์ HD44780 และ HD61830 โดย HD44780 จะใช้ควบคุม LCD แบบอักษร ส่วน HD61830 ใช้ควบคุม LCD แบบกราฟฟิก

ตัวขับ (Driver)

เป็นตัวรับสัญญาณจากตัวควบคุมมาขับให้ตัวแสดงผล แสดงผลตามที่กำหนด ชิปที่ทำหน้าที่เป็นตัวขับได้แก่ เบอร์ HD44100H และ MSM5259 เป็นต้น

2.5.1 โครงสร้างภายในของตัวควบคุมโมดูล LCD

ในการใช้งานโมดูล LCD จำเป็นต้องทำความเข้าใจเกี่ยวกับโครงสร้างและคำศัพท์ที่ใช้ในการควบคุมให้ดียิ่งขึ้น โดยโมดูล LCD แบบอักษร สามารถเข้าใจได้ง่าย รูปที่ 2.18 เป็น Block Diagram ภายในของชิปควบคุม LCD เบอร์ HD44780 ซึ่งใช้โมดูล LCD แบบอักษรประกอบด้วย



เอกสารนี้เป็นเอกสารที่สรุปที่ 2.18 ไคอะแกรมการทำงานของ โมดูล LCD แบบอักษร

บัพเฟอร์อินพุต/เอาต์พุตเป็นส่วนที่ใช้ในการคัดลอกรับส่งข้อมูลกับอุปกรณ์ภายนอก เพื่อที่จะถ่ายเทข้อมูลเข้าออกภายในตัวควบคุม

รีจิสเตอร์คำสั่ง (Instruction Register : IR)

เป็นรีจิสเตอร์ใช้รับข้อมูลคำสั่งจากอุปกรณ์ภายนอกเพื่อนำไปควบคุมการแสดงผล

รีจิสเตอร์ข้อมูล (Data Register : DR)

เป็นรีจิสเตอร์ใช้รับข้อมูลจากอุปกรณ์ภายนอกเพื่อถ่ายเทต่อไปยังหน่วยความจำที่ทำหน้าที่เก็บข้อมูลแสดงผล หรือนำข้อมูลไปสร้างตัวอักษรเพิ่มเติมในหน่วยความจำเก็บตัวอักษร

RAM เก็บข้อมูลแสดงผล (Display Data Ram : DDRAM)

เป็นหน่วยความจำ RAM ที่ทำหน้าที่เก็บข้อมูลที่มาจากรีจิสเตอร์ DR ตัวควบคุมจะนำข้อมูลใน DDRAM นี้ไปเปิดตาราง (Lock up - lable) ของตัวอักษรที่เก็บไว้ในหน่วยความจำ ROM และ RAM เก็บตัวอักษร เพื่อนำไปที่ตัวแสดงผล

ROM เก็บตัวอักษร (Character Generator ROM : CGROM)

เป็นหน่วยความจำ ROM ที่ใช้เก็บข้อมูลตัวอักษรหรือสัญลักษณ์ที่สามารถอ่านออกไปแสดงผลได้ มีขนาด 7200 บิต โดยจะถูกอ่านด้วยค่าของข้อมูลใน DDRAM

RAM เก็บตัวอักษร (Character Generator RAM : CGRAM)

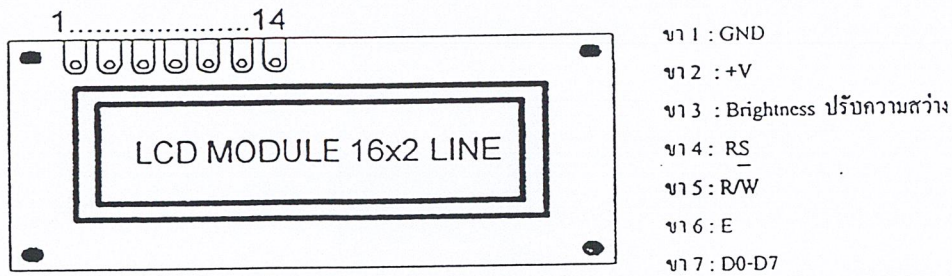
เป็นหน่วยความจำ RAM ที่ใช้เก็บอักษรที่มีการสร้างเพิ่มเติมขึ้นใหม่ ในกรณีที่ตัวอักษรใน CGRAM ไม่เพียงพอ มีขนาด 512 บิต การเขียนและอ่านค่าไปใช้นั้นทำได้เช่นเดียวกับ CGROM คือเขียนข้อมูลลงใน DDRAM แล้วตัวควบคุมจะมาอ่านค่าจาก CGRAM เอง

FLAG BUSY

เป็นส่วนหนึ่งที่ทำหน้าที่แจ้งสถานะการทำงานของตัวควบคุมให้อุปกรณ์ภายนอกทราบว่า ตัวควบคุมพร้อมที่จะรับข้อมูลหรือคำสั่งหรือไม่ ดังนั้นก่อนการส่งข้อมูลหรือคำสั่งมายังตัวควบคุมต้องตรวจสอบสถานะของ FLAG BUSY นี้เสียก่อน

2.5.2 โมดูล LCD ขนาด 16 ตัวอักษร 2 บรรทัด (LCD 16X2)

สำหรับโมดูล LCD ที่ใช้ในการเรียนรู้ในการทดลอง เป็นขนาด 16 ตัวอักษร 2 บรรทัด เนื่องจากราคาถูก และเป็นโมดูล LCD ที่มีโครงสร้างเป็นมาตรฐาน มีผู้ผลิตหลายราย และการระบุเบอร์แตกต่างกันออกไปตามผู้ผลิต อาทิเช่น LM020L ของ Hitachi , DMC - 16117A ของ คอปเทร็กซ์ เป็นต้น แต่อย่างไรก็ตามคนโทรลเลอร์ที่ใช้คือเบอร์เดียวกัน คือ HD44780 ของ Hitachi



รูปที่ 2.19 รูปร่างและการจัดขาโมดูล LCD

โมดูลขนาด 16 X 2 มีขาที่ต่อใช้งานทั้งสิ้น 14 ขา มีการจัดขาตั้งในรูป สำหรับรายละเอียดการทำงานของแต่ละขามีดังนี้

- ผล
- Vss (ขา 1) : ต่อกราวด์
 - Vdd (ขา 2) : ต่อไฟเลี้ยง +5 โวลต์
 - Vo (ขา 3) : เป็นขาอินพุตรับแรงดันเพื่อปรับความเข้มของการแสดงผล
 - RS (ขา 4) : เป็นขาอินพุตใช้ในการแยกชนิดของข้อมูลที่ทำการประมวลผลในขณะนั้นว่าเป็นคำสั่งสำหรับรีจิสเตอร์ IR หรือเป็นข้อมูลสำหรับรีจิสเตอร์ DR โดยถ้าขาเป็น “ 0 ” ข้อมูลที่ส่งมาจะเป็นคำสั่ง แต่ถ้าขาเป็น “ 1 ” ข้อมูลที่ส่งมาจะเป็นข้อมูลสำหรับการแสดงผล
 - R/W (ขา 5) : เป็นขาที่ใช้เลือกการอ่านหรือเขียนข้อมูลกับ LCD ถ้าเป็น “ 0 ” เป็นการกำหนดให้เขียนข้อมูล แต่ถ้าเป็น “ 1 ” จะเป็นการอ่านข้อมูล
 - E (ขา 6) : เป็นขา Enable “ LCD ” ให้ทำงาน
 - D0 - D7 (ขา 7- 14) : เป็นขาที่ใช้เป็นทางผ่านของข้อมูลระหว่าง LCD กับอุปกรณ์ภายนอกขนาด 8 บิต

2.5.3 คำสั่งควบคุมโมดูล LCD

ในการเขียนคำสั่งลงในตัวควบคุม แน่นอนว่าจะต้องกำหนดให้ขา RS, R/W เป็น 0 แล้วเขียนคำสั่งตามไป คำสั่งควบคุมโมดูล LCD ของชิปควบคุม HD44780 ที่สำคัญมี 10 คำสั่ง ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. คำสั่งเคลียร์ตัวแสดงผล (Clear Display)

มีข้อมูลคำสั่งเป็น 01H เป็นคำสั่งที่ใช้เขียนข้อมูลช่องว่างหรือ Space เข้าไปใน DDRAM ทั้งหมด เมื่อตัวควบคุม Excute คำสั่งนี้ จะทำการกำหนดแอดเดรส ของ DDRAM เป็น “0” เคอร์เซอร์จะกลับไปอยู่ที่ตำแหน่งซ้ายสุดของจอแสดงผล แล้ว Set บิต I/D ให้เป็น “1”

2. คำสั่ง return home

ต้องกำหนดข้อมูลของบิตที่ 1 ของข้อมูลให้เป็น “1” เป็นคำสั่งให้เคอร์เซอร์เคลื่อนที่กลับไปยังตำแหน่งซ้ายสุดของจอแสดงผลแต่ข้อมูลบนจอแสดงผลไม่เปลี่ยนแปลง นั่นคือข้อมูลของคำสั่งนี้จะเป็น 02H หรือ 03H ก็ได้

3. คำสั่งเลือกโหมดการป้อนข้อมูล (Entry Mode Set)

มีรายละเอียดของข้อมูลรูปแบบคำสั่งดังนี้

บิต 7 บิต 6 บิต 5 บิต 4 บิต 3 บิต 2 บิต 1 บิต 0

0	0	0	0	0	1	I/D	S
---	---	---	---	---	---	-----	---

บิต S เป็นบิตที่ใช้ในการกำหนดลักษณะของการแสดงผล เมื่อมีการป้อนข้อมูล ถ้าหากบิต S เป็น “1” เมื่อเกิดข้อมูลใหม่บนจอแสดงผล ตัวเคอร์เซอร์จะอยู่กับที่ แต่ตัวอักษรข้อมูลเดิมจะถูกดันไปทางซ้าย แต่ถ้าหากบิตนี้เป็น “0” เมื่อเกิดข้อมูลตัวเคอร์เซอร์จะเลื่อนไปทางขวามือ

บิต I/D เป็นบิตที่ใช้กำหนดว่า เมื่อเขียนหรืออ่านข้อมูลแล้ว ทำให้แอดเดรสของ DDRAM เพิ่มขึ้นหรือลดลงหนึ่งแอดเดรส โดยถ้าบิตนี้เป็น “1” แอดเดรสของ DDRAM จะเพิ่มขึ้น แต่ถ้าบิตนี้เป็น “0” แอดเดรสจะลดลง

ดังนั้น ข้อมูลคำสั่งที่เกิดขึ้นสำหรับคำสั่งนี้ได้แก่ 04H – 07H (4 ข้อมูลคำสั่ง) และที่ใช้บ่อยคือ 06H หมายถึง กำหนดให้เมื่อเกิดข้อมูลใหม่ เคอร์เซอร์จะเลื่อนไปทางขวามือ และแอดเดรสของ DDRAM เพิ่มขึ้น

4. คำสั่งควบคุมการแสดงผล

มีรายละเอียดรูปแบบข้อมูลคำสั่งดังนี้

บิต 7 บิต 6 บิต 5 บิต 4 บิต 3 บิต 2 บิต 1 บิต 0

0	0	0	0	0	D	C	B
---	---	---	---	---	---	---	---

บิต D ใช้ควบคุมการเปิดปิดจอแสดงผล ถ้าบิตนี้เป็น “ 1 ” จะเป็นการเปิดจอแสดงผล ถ้าบิตนี้เป็น “ 0 ” จะเป็นการปิดจอแสดงผล

บิต C ใช้ควบคุมตัวแสดงเคอร์เซอร์บนจอแสดงผล ถ้าต้องการให้มีเคอร์เซอร์แสดงบนจอแสดงผล ต้องกำหนดให้บิตนี้เป็น “ 1 ” แต่ถ้ากำหนดให้บิตนี้เป็น “ 0 ” จะเป็นการปิดเคอร์เซอร์

บิต B ใช้ควบคุมการกระพริบของเคอร์เซอร์ ถ้าบิตนี้เป็น “ 1 ” เคอร์เซอร์จะกระพริบ

ดังนั้นจะมีข้อมูลคำสั่งได้ตั้งแต่ 08H – 0FH (8 รูปคำสั่ง) ที่ใช้บ่อยคือ 0CH เป็นการสั่งให้เปิดจอแสดงผล แต่ไม่แสดงเคอร์เซอร์ และ 0FH เป็นการสั่งให้เปิดจอแสดงผล และแสดงเคอร์เซอร์ รวมถึงสั่งให้เคอร์เซอร์กระพริบด้วย

5. คำสั่งควบคุมการเลื่อนเคอร์เซอร์และข้อมูลตัวอักษร

มีรายละเอียดของรูปแบบข้อมูลคำสั่งดังนี้

บิต 7 บิต 6 บิต 5 บิต 4 บิต 3 บิต 2 บิต 1 บิต 0

0	0	0	1	S/C	R/L	*	*
---	---	---	---	-----	-----	---	---

การควบคุมการเลื่อนเคอร์เซอร์และตัวอักษรบนจอแสดงผลขึ้นกับการกำหนดบิต S/C และ R/L ซึ่งสามารถสรุปได้ดังนี้

S/C	R/L	ลักษณะการเลื่อน	ข้อมูลคำสั่ง
0	0	เลื่อนเคอร์เซอร์ไปทางซ้าย	10H – 13H
0	1	เลื่อนเคอร์เซอร์ไปทางขวา	14H – 17H
1	0	เลื่อนเคอร์เซอร์ใหม่ไปทางซ้าย	18H – 1BH
1	1	เลื่อนเคอร์เซอร์ใหม่ไปทางขวา	1CH – 1FH

6. คำสั่งกำหนดฟังก์ชันการทำงาน

มีรายละเอียดของรูปแบบข้อมูลคำสั่งดังนี้

บิต 7 บิต 6 บิต 5 บิต 4 บิต 3 บิต 2 บิต 1 บิต 0

0	0	1	DL	N	F	*	*
---	---	---	----	---	---	---	---

บิต DL ใช้กำหนดจำนวนบิตที่ใช้ติดต่อส่งผ่านข้อมูล ถ้าบิตนี้เป็น “ 0 ” จะเป็นการติดต่อแบบ 4 บิต แต่ถ้าบิตนี้เป็น “ 1 ” จะเป็นแบบ 8 บิต

บิต N ใช้กำหนดจำนวนบรรทัดของการแสดงผล 1 บรรทัด ถ้าเป็น “ 1 ” จะแสดง 2 บรรทัด ในกรณีที่จอแสดงผลสามารถที่จะแสดงผลได้มากกว่า 2 บรรทัด และต้องการให้แสดงผลมากกว่า 2 บรรทัด ก็กำหนดให้บิต N เป็น “ 1 ”

บิต F ใช้เลือกความละเอียดของตัวอักษรให้การแสดงผล ถ้าบิตนี้เป็น “ 0 ” จะเป็นการแสดงผลแบบ 5X7 จด และถ้าเป็น “ 1 ” จะแสดงผลเป็น 5X10 จด

ข้อมูลค่าตั้งที่ใช้บ่อยคือ 38H เป็นการกำหนดให้โมดูล LCD ทำงานในแบบ 8 บิต แสดงผล 2 บรรทัด และเลือกความละเอียดเป็น 5X7 จด

7. คำสั่งเลือกแอดเดรสของ CGRAM

เมื่อต้องการกำหนดแอดเดรสของ CGRAM ต้องกำหนดให้บิตที่ 7 เป็น “ 0 ” บิตที่ 6 เป็น “ 1 ” ส่วนอีก 6 บิต ที่เหลือจะแทนด้วยค่าแอดเดรสของ CGRAM จะต้องทำการกำหนดแอดเดรสด้วยคำสั่งนี้ ก่อนที่จะอ่านหรือเขียนข้อมูลให้ CGRAM โดยแอดเดรสของ CGRAM อยู่ระหว่าง 00H – 3FH

8. คำสั่งเลือกแอดเดรสของ DDRAM

ใช้ในการเลือกแอดเดรสของ DDRAM ก่อนที่จะทำการอ่านหรือเขียนข้อมูล โดยบิตที่ 7 ต้องเป็น “ 1 ” และข้อมูลอีก 7 บิตที่เหลือจะเป็นค่าแอดเดรสของ DDRAM ซึ่งแอดเดรสของ DDRAM จะอยู่ระหว่าง 8CH – 0FFH ทั้งนี้จำนวนแอดเดรสวิ่งขึ้นอยู่กับการกำหนดสถานะบิต N ด้วย หากบิต N เป็น “ 0 ” แอดเดรสของ DDRAM จะอยู่ระหว่าง 80H – 0CFH บิต N เป็น “ 1 ” แอดเดรสของ DDRAM จะมี 2 ช่วงคือ 80CH – 87H และ 0C0H – 0C7H ดังรูป

4.7 2 LINE X 16 CHARACTERS PER LINE

Char	4.8	4.9	4.10	4.11	4.12	4.13	4.14	4.15	4.16	4.17	4.18	4.19	4.20	4.21	4.22	4.23
Line 1	4.24	4.25	4.26	4.27	4.28	4.29	4.30	4.31	4.32	4.33	4.34	4.35	4.36	4.37	4.38	4.39
Line 2	4.40	4.41	4.42	4.43	4.44	4.45	4.46	4.47	4.48	4.49	4.50	4.51	4.52	4.53	4.54	4.55

รูปที่ 2.20 ตำแหน่ง DDRAM

9. คำสั่งอ่าน Flag Busy ของ แอดเดรส
มีรายละเอียดของรูปแบบข้อมูลคำสั่งดังนี้

บิต 7 บิต 6 บิต 5 บิต 4 บิต 3 บิต 2 บิต 1 บิต 0

BF	A	A	A	A	A	A	A
----	---	---	---	---	---	---	---

เป็นคำสั่งที่ใช้อ่าน Flag Busy (BF) โดยแฟล็กนี้จะเป็นตัวบอกสถานะตัวควบคุม LCD ว่าพร้อมจะรับข้อมูลหรือไม่ ถ้าหากบิต BF เป็น “0” แสดงว่าตัวควบคุม LCD พร้อมรับข้อมูล หรือคำสั่ง แต่ถ้าเป็น “1” แสดงว่า ขณะนี้ตัวควบคุม LCD ยังอยู่ในกระบวนการภายใน หรือกำลังประมวลผลข้อมูลอยู่ และยังไม่พร้อมรับข้อมูลหรือคำสั่ง

เมื่อต้องการอ่านแฟล็กต้องกำหนดให้ขา R/W เป็น “1” ด้วย แต่สัญญาณที่ RS ยังต้องเป็น “0” อยู่เพราะข้อมูลนี้เป็นข้อมูลคำสั่ง

นอกจากนี้ ยังใช้เป็นคำสั่งอ่านข้อมูลแอดเดรสของ CGRAM และ DDRAM ด้วย โดยบิต 0-6 เป็นค่าข้อมูลของแอดเดรสที่ต้องการอ่าน

2.5.5 การเขียนคำสั่งและข้อมูลให้แก่มอูล

ในการเขียนข้อมูลเพื่อควบคุมให้โมดูล LCD แสดงผลตามที่ใช้ต้องการ ต้องส่งคำสั่ง (Instruction) แล้วกำหนดโหมดการทำงานให้แก่มอูล LCD ก่อน จากนั้นจึงค่อยส่งข้อมูล (Data) ที่ต้องการแสดงผลเนื่องจากบิตของโมดูล LCD มี 8 เส้นคือ D0 - D7 และใช้เป็นทางผ่านของทั้งคำสั่งและข้อมูล ดังนั้นในการส่งคำสั่งและข้อมูลจึงต้องอาศัยการกำหนดสัญญาณลอจิกที่ขาเอกสารเป็นเอกสารที่ส่งงานไว้สำหรับการเขียนเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

RS ถ้าหากที่ขา RS ได้ลอจิก “ 0 ” หมายความว่า ข้อมูลที่ป้อนให้แก่โมดูล LCD ขณะนั้นเป็นคำสั่งในทางตรงกันข้าม หากขานี้ได้รับลอจิกเป็น “ 1 ” ข้อมูลที่ป้อนให้ขณะนั้นเป็นข้อมูลที่ใช้ในการแสดงผล

เมื่อต้องการเขียนหรืออ่านข้อมูลใน CGRAM และ DDRAM เริ่มต้นต้องกำหนดแอดเดรสที่ต้องการอ่านหรือเขียนก่อน โดยใช้คำสั่งเลือกแอดเดรส จากนั้นกำหนดให้ขา RS เป็น “ 1 ” เพื่อแจ้งให้ตัวควบคุมภายในโมดูล LCD ทราบว่าข้อมูลที่ปรากฏต่อไปนี้เป็นข้อมูลปกติ ไม่ใช่ข้อมูลคำสั่ง

ในกรณีที่ต้องการอ่านข้อมูลต้องกำหนดให้ขา R/W เป็น “ 1 ” ข้อมูลขนาด 8 บิต (หรือ 4 บิต) ก็จะปรากฏบนบัสข้อมูล โดยข้อมูลที่อ่านออกมาได้จะเป็นข้อมูลจากแอดเดรสของ CGRAM หรือ DDRAM ตามที่ต้องการ

ในกรณีที่ต้องการเขียนข้อมูล เมื่อกำหนดแอดเดรสและป้อนลอจิก “ 1 ” ให้ขา RS แล้วต้องกำหนดให้ขา R/W เป็น “ 0 ” ข้อมูลที่อยู่บนบัสข้อมูลจะถูกเขียนลงบนรีจิสเตอร์ DR จากนั้นจึงถ่ายทอดลงใน DDRAM ต่อไป

2.6 การโปรแกรมแบบวิซวล (Visual Programming)

วิซวล (Visual) หมายถึง ภาพหรือสิ่งที่เรามองเห็น ดังนั้น วิซวลโปรแกรมมิง (Visual Programming – การเขียนโปรแกรมแบบวิซวล) จึงหมายถึง การเขียนโปรแกรมด้วยภาพ หรือการเขียนโปรแกรมด้วยสิ่งที่เรามองเห็น

ภาพประกอบเบื้องต้นสำหรับใช้เขียนโปรแกรมแบบวิซวลมี 4 รายการดังต่อไปนี้

1. ฟอर्म (Form) ฟอर्मเป็นวินโดวที่ว่างที่ใช้สำหรับออกแบบโปรแกรม การทำงานต่างๆ ของโปรแกรมจะปรากฏอยู่บนฟอर्म

2. คอมโพเนนต์ (Component) คอมโพเนนต์เป็นส่วนประกอบต่างๆ ที่ต้องใช้ในโปรแกรม

เช่น เมนู (Menu) , ปุ่มหรือบัตตอน (Button – ปุ่มสำหรับกดให้โปรแกรมทำงานชนิดหนึ่ง) , เมสเสจบ็อกซ์ (Message Box – กรอบแสดงข้อความไคอะล็อกบ็อกซ์) , ไคอะล็อกบ็อกซ์ (Dialog Box – กรอบสำหรับโต้ตอบกับโปรแกรม)

3. คำสั่งสำหรับจัดลักษณะและการทำงานของคอมโพเนนต์

4. คำสั่งสำหรับควบคุมการทำงานของโปรแกรม

วิซวลโปรแกรมมิงช่วยให้เราสามารถเขียนโปรแกรมได้ง่าย สะดวก รวดเร็ว และถูกต้อง

ปาสคาล (Pascal) เป็นภาษาคอมพิวเตอร์ที่สร้างขึ้นโดยมีจุดมุ่งหมายเพื่อให้ผู้เขียน

เอกรัโปรแกรมได้เรียนรู้เกี่ยวกับวิธีการเขียนโปรแกรมที่ดี ซึ่งจะเป็นพื้นฐานในการเขียนโปรแกรมขนาดใหญ่ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งยังมีให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ใหญ่ที่มีความซับซ้อนได้โดยสะดวก ปาสคาลที่นิยมใช้กันมากในไมโครคอมพิวเตอร์ได้แก่ เทอร์โบปาสคาล (Turbo Pascal) ซึ่งเป็นผลงานของบริษัท บอร์แลนด์อินเตอร์เนชันแนล (Borland International , Inc.) ประเทศสหรัฐอเมริกา เทอร์โบปาสคาลเวอร์ชัน 1.0 วางจำหน่ายในปี 1984 ตั้งแต่เวอร์ชัน 5.5 ซึ่งวางจำหน่ายในปี 1999 เป็นต้นมา เทอร์โบปาสคาลได้เพิ่มวิธีการเขียนโปรแกรมแบบออบเจกต์ ซึ่งเรียกว่า การเขียนโปรแกรมแบบ OOP (Object - Oriented Programming) และเรียกชื่อปาสคาลชนิดนี้ว่า ออบเจกต์ปาสคาล (Object Pascal) ออบเจกต์ปาสคาลมีทั้งชนิดที่รันในคอสมและรันในวินโดวส์

ออบเจกต์ปาสคาลและเดลไฟล์ (Delphi) ต่างก็เป็นภาษาคอมพิวเตอร์ซึ่งใช้สำหรับเขียนโปรแกรมเพื่อสั่งให้คอมพิวเตอร์ทำงานต่างๆ ตามที่เราต้องการ ออบเจกต์ปาสคาลและเดลไฟล์มีคำสั่งและกฎเกณฑ์ในการเขียนโปรแกรมเหมือนกันเกือบทุกประการ แต่มีวิธีเขียนโปรแกรมต่างกัน กล่าวคือ ออบเจกต์ปาสคาลใช้วิธีเขียนโปรแกรมด้วยอักษรและเครื่องหมาย แต่เดลไฟล์ใช้วิธีเขียนโปรแกรมแบบวิชวล ดังนั้นผู้ที่เขียนโปรแกรมด้วยออบเจกต์ปาสคาลจึงสามารถเปลี่ยนมาเขียนโปรแกรมด้วยเดลไฟล์ได้อย่างรวดเร็ว เพราะเปลี่ยนแปลงเฉพาะวิธีการเท่านั้น ซึ่งวิธีการของเดลไฟล์จะทำให้เราเขียนโปรแกรมได้ง่าย และเร็วกว่าวิธีการของออบเจกต์ปาสคาล เดลไฟล์ผลิตโดยบริษัท บอร์แลนด์อินเตอร์เนชันแนล เดลไฟล์เวอร์ชันแรกนั้นวางจำหน่ายในปี 1995 เดลไฟล์เวอร์ชัน 4 ซึ่งมีชื่อว่า Delphi 4 วางจำหน่ายประมาณกลางปี 1998 โดยแบ่งการจำหน่ายเป็น 4 ชุดคือ

1. Delphi Standard (ชุดมาตรฐาน)
2. Delphi Professional (ชุดระดับมืออาชีพ)
3. Delphi Client/Server (ชุดสำหรับเครือข่ายคอมพิวเตอร์)
4. Delphi Enterprise (ชุดสำหรับองค์กร)

เดลไฟล์เป็นวิชวลโปรแกรมมิ่ง ซึ่งมีวิธีการเขียนโปรแกรมโดยการนำเอาคอมโพเนนต์ที่เรามองเห็นมาวางในฟอร์ม แล้วกำหนดลักษณะและการทำงานให้กับคอมโพเนนต์นั้น ในบางโปรแกรมอาจจะต้องเขียนคำสั่งควบคุมการทำงานของโปรแกรมเพิ่มเติม ต่อจากนั้นจึงคอมไพล์ (Compile) และรัน (Run) โปรแกรมนั้นได้ ผลจากการคอมไพล์โปรแกรมจะได้ไฟล์ชนิด .EXE ซึ่งสามารถนำไปรันในคอมพิวเตอร์เครื่องอื่นได้อย่างอิสระ โดยไม่ต้องเกี่ยวข้องกับเดลไฟล์อีก

เดลไฟล์เป็นออบเจกต์ปาสคาลประเภทที่สามารถสร้างโปรแกรมใช้งานได้อย่างรวดเร็ว ภาษาคอมพิวเตอร์ประเภทนี้มีชื่อย่อว่า RAD (Rapid Application Development) ซึ่งหมายความว่าด้วยเดลไฟล์เราจะสามารถเขียนโปรแกรมเพื่อรันในวินโดวส์ได้เร็วกว่าและง่ายกว่าการเขียน

เอกปรแกรมออบเจกต์ปาสคาลแบบธรรมดาเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อีเวนต์ (Even)

อีเวนต์ หมายถึง ทุกสิ่งทุกอย่างที่เกิดขึ้นในระบบคอมพิวเตอร์ เช่น การเลื่อนเมาส์ , การคลิกเมาส์ , การกดคีย์ , การเลื่อนวินโดวส์ และการเปลี่ยนขนาดของวินโดวส์ วินโดวส์จะเก็บอีเวนต์ทั้งหมดไว้ในคิว (Queue) ที่มีชื่อว่า โกลบอลอีเวนต์คิว (Global Even Queue – คิวของอีเวนต์ทั้งหมดซึ่งบางทีเรียกว่า System – Wide Queue หรือ Hardware Even Queue) โกลบอลอีเวนต์คิวใช้สำหรับเก็บอีเวนต์ของทุก Application ที่รันในวินโดวส์

เมสเสจ (Message)

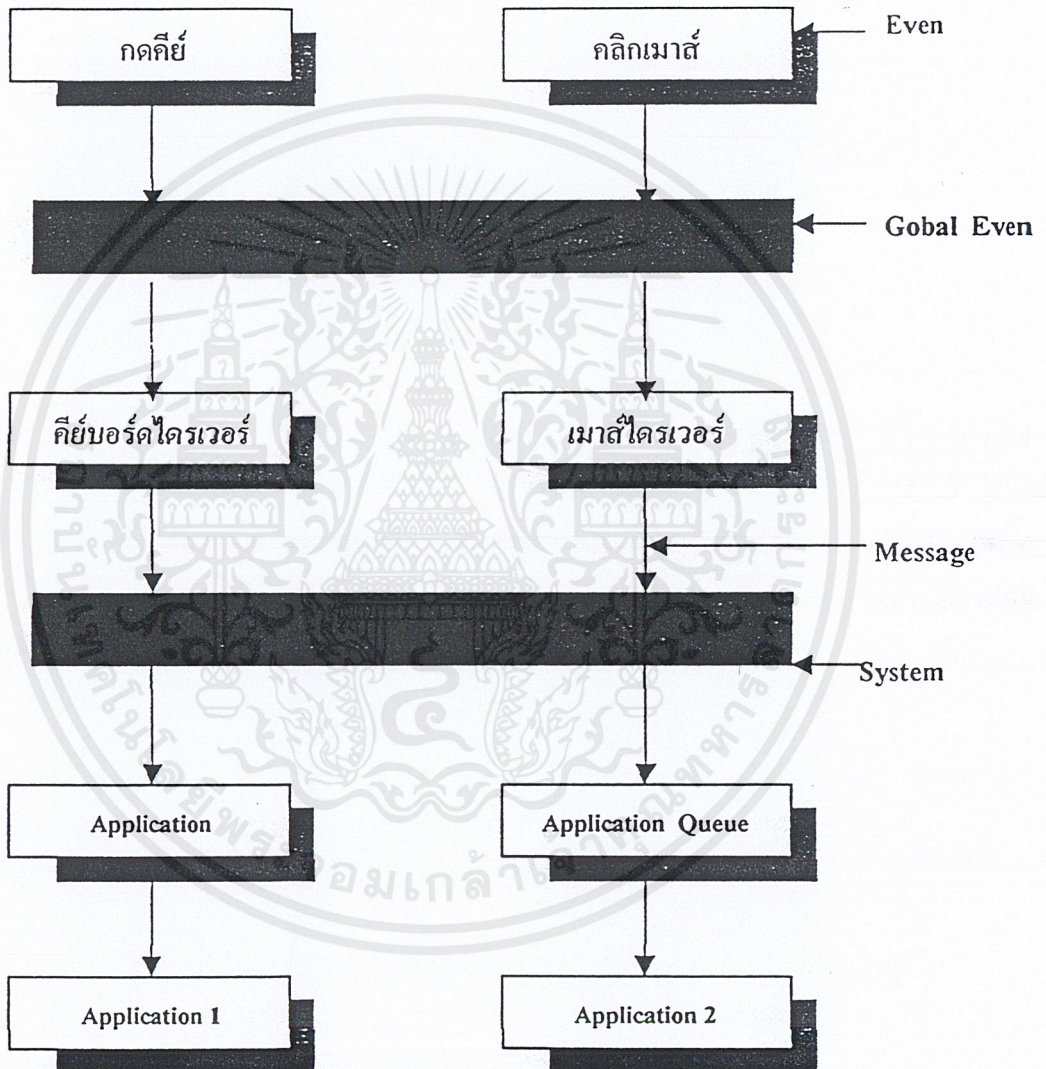
เมสเสจ หมายถึง ชุดของข่าวสารที่เป็นรายละเอียดของแต่ละอีเวนต์ วินโดวส์จะสร้างเมสเสจของอีเวนต์ โดยการส่งอีเวนต์ที่เกิดขึ้นให้แก่ ไดรเวอร์ (Driver – โปรแกรมที่คอมพิวเตอร์ใช้ติดต่อกับอุปกรณ์ชนิดนั้น) ที่สอดคล้องกับอีเวนต์นั้น เช่น เมื่อเราคลิกเมาส์จะเกิดอีเวนต์คลิกเมาส์ วินโดวส์จะส่งอีเวนต์นี้ให้แก่เมาส์ไดรเวอร์ และเมาส์ไดรเวอร์จะสร้างเมสเสจคลิกเมาส์ ไดรเวอร์จะเก็บเมสเสจจากนี้ไว้ในซิสเต็มคิว (System Queue - คิวของระบบ) ต่อจากนั้นวินโดวส์จึงส่งเมสเสจจากซิสเต็มคิวไปที่แอปพลิเคชันคิว (Application Queue – คิวเก็บเมสเสจของแอปพลิเคชัน) ของแอปพลิเคชันต่างๆ ที่กำลังทำงานอยู่ในขณะนั้น ซึ่งเป็นผลให้แอปพลิเคชันได้รับเมสเสจจากวินโดวส์ เนื่องจากวินโดวส์สามารถรันแอปพลิเคชันคิวสำหรับแอปพลิเคชันนั้นได้ โดยเฉพาะความสัมพันธ์ระหว่างอีเวนต์กับเมสเสจ

หลักการทำงานของแอปพลิเคชันที่รันในวินโดวส์ก็คือ การดำเนินการกับเมสเสจ การรับส่งเมสเสจกับวินโดวส์หรือกับแอปพลิเคชันอื่น เมสเสจทั้งหมดของวินโดวส์กำหนดไว้เป็นคอนสแตนต์ (Constant – ค่าที่ไม่เปลี่ยนแปลง) เช่น เมื่อเราเลือกการหนึ่งจากเมนูที่ปรากฏบนจอภาพ จะมีเมสเสจ “ WM_COMMAND ” เกิดขึ้น ในบรรดาเมสเสจเหล่านี้มีหลายเมสเสจที่เกิดขึ้นพร้อมกับ “ แวริเอเบิล ” (Variable – ค่าที่เปลี่ยนแปลงได้) จำนวนหนึ่งซึ่งนำมาใช้ในแอปพลิเคชันได้ เช่น เมื่อเรากดปุ่มซ้ายของเมาส์ วินโดวส์จะสร้างเมสเสจ WM_LBUTTONDOWN ขึ้นมา เมสเสจนี้จะประกอบด้วยแวริเอเบิล ซึ่งเป็นโคออร์ดิเนต (Coordinate – ตำแหน่ง) ของเมาส์พอยน์เตอร์ เราสามารถนำค่าดังกล่าวนี้มาใช้ได้ เช่น ใช้สำหรับแสดงผลบนจอภาพ ใช้ในการตรวจสอบตำแหน่งของเมาส์พอยน์เตอร์

โปรแกรมที่ทำงานด้วยอีเวนต์

แอปพลิเคชันที่รันในวินโดวส์ จะใช้เวลาส่วนใหญ่เพื่อการรับและส่งเมสเสจชนิดต่างๆ จึงนิยมเรียกแอปพลิเคชัน (หรือโปรแกรม) ที่ทำงานในลักษณะนี้ว่าโปรแกรมที่ทำงานด้วยอีเวนต์ (Even - Drive Program) ซึ่งเป็นโปรแกรมที่มีหลักการทำงานดังต่อไปนี้คือ ในโปรแกรมที่ทำงานนี้ ไม่มีการเฝ้าดูคีย์บอร์ด หรือเมาส์ให้คอยตรวจจับการกดคีย์หรือการคลิกเมาส์ แต่การตรวจจับการกดคีย์หรือการคลิกเมาส์จะถูกส่งมาในรูปของอีเวนต์ให้กับโปรแกรมที่ทำงานนี้

ด้วยอีเวนต์จะประกอบด้วย เมธอด (Method – สเตตเมนต์ชุดหนึ่งซึ่งทำงานอย่างหนึ่ง และมีชื่อเรียกเฉพาะสำหรับให้ส่วนอื่นของโปรแกรมเรียกใช้ได้) จำนวนหนึ่งซึ่งเรียกว่า เมธอดตอบสนองอีเวนต์ ซึ่งแต่ละเมธอดจะมีหน้าที่ตอบสนองกับแต่ละอีเวนต์โดยเฉพาะและเป็นอิสระต่อกัน ทั้งนี้เนื่องจากวินโดวส์จะมีอีเวนต์เกิดขึ้นเวลาใดก็ได้ เราจึงไม่ทราบล่วงหน้าว่าจะเรียกใช้เมธอดอะไร เมื่อใด



รูปที่ 2.21 ลักษณะของโปรแกรมที่ทำงานด้วยอีเวนต์

โครงสร้างโปรแกรมของเดลไฟล์ (The Structure of a Delphi Program)

การเขียนโปรแกรมเดลไฟล์ให้ได้ประโยชน์มากขึ้น ถ้านักพัฒนาโปรแกรมเข้าใจโครงสร้างของ Object Pascal ซึ่งเป็นรากฐานของเดลไฟล์ การศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

IDE เป็นเครื่องมือที่ช่วยให้สามารถเขียนโปรแกรมได้เร็วขึ้นโดยที่ IDE ช่วยเขียนคำสั่งที่จำเป็นต้องใช้ และช่วยให้นักพัฒนาสามารถออกแบบจอภาพและเขียนคำสั่งที่ต้องการให้ทำงานเมื่อมีเหตุการณ์ (Event) เกิดขึ้น

IDE จะช่วยเขียนโค้ดคำสั่งงานซึ่งอยู่ภายใต้กฎเกณฑ์ของ Object Pascal และช่วยสร้าง Resource ที่ต้องใช้ตามกฎการทำงานของระบบปฏิบัติการแบบ Windows

โพรซีเจอร์ (Procedure)

โพรซีเจอร์ เป็นกลุ่มคำสั่งงานที่ใช้ในการทำงานอย่างใดอย่างหนึ่ง อาจจะเรียกอีกอย่างหนึ่งว่า “ โปรแกรมย่อย ” เช่น การคำนวณภาษี , การแสดงจอภาพเพื่อรับค่าจากผู้ใช้งาน เป็นต้น

ตัวอย่างของโพรซีเจอร์

1. procedure Button1Click;
2. begin
3. Edit1.Text:='Hello World';
4. End;

ในโพรซีเจอร์ตัวอย่างนี้จะแสดงข้อความว่า 'Hello World' ในคอนโทรล Edit1 เมื่อมีผู้ใช้งานกดปุ่ม Button1 โดยที่ชื่อโพรซีเจอร์จะอยู่บรรทัดที่ 1 และมีคำสั่ง begin และ end คลุมคำสั่งที่ใช้ในการทำงานของโพรซีเจอร์นี้

- คำสั่งที่อยู่ในโพรซีเจอร์อาจมีได้ตั้งแต่คำสั่งเดียวจนถึงหลายๆ หน้ากระดาษ
- เนื่องจากภาษา Object Pascal เป็นภาษาที่ Case – Insensitive ดังนั้นการตั้งชื่อต่างๆ ที่ใช้อักษรตัวใหญ่และอักษรตัวเล็กจะไม่แตกต่างกันเช่น Edit1 หรือ edit1 จะเป็นการอ้างถึงชื่อเดียวกัน
- เพื่อความสะดวกในการอ่านโปรแกรมและความง่ายในการแก้ไขโปรแกรม ควรจะใช้อักษรตัวใหญ่ผสมกับตัวเล็ก

2.7 DS1307ไอซีสร้างฐานเวลาจริงหรือรีลไทม์คล็อก (RTC)

ผู้ผลิตคือ ดัลลัสเซมิคอนดักเตอร์ (Dallas semiconductor) มีหน้าที่สร้างฐานเวลาให้แก่ระบบไมโครคอนโทรลเลอร์โดย DS1307 จะให้ข้อมูลเกี่ยวกับเวลาทั้งหมด ไม่ว่าจะเป็นค่าของเวลาที่ละเอียดถึงหลักวินาที, นาที, ชั่วโมง, วันที่ (date), วันในสัปดาห์ (day), เดือน และปี โดยสามารถปรับวันเดือนปีให้ตรงตามปฏิทินได้อย่างถูกต้องรวมถึงการกำหนดวันในปีอธิกสุรทินด้วย คุณสมบัติทางเทคนิคที่สำคัญมีดังนี้

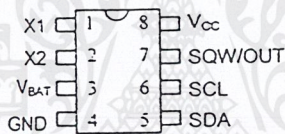
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- เป็นไอซีรีลไทม์คล็อกให้ข้อมูลตั้งแต่วินาทีจนถึงปี รวมถึงการปรับวันในปีอธิกสุรทินด้วยสามารถให้ข้อมูลเวลาได้อย่างเที่ยงตรงถึงปีคริสตศักราช 2100 มีหน่วยความจำอนโวลตาไทล์แรม 56 ไบต์อยู่ภายใน สามารถใช้เก็บข้อมูลทั่วไปได้
- ใช้การเชื่อมต่อแบบระบบบัส I2C มีวงจรตรวจจับไฟเลี้ยงต่ำหรือหายไปอย่างอัตโนมัติและสามารถรักษาข้อมูลเวลาไว้ได้แม้ไม่มีไฟเลี้ยงไอซี

รายละเอียดการใช้งานของ DS1307

ในรูปที่ 2.22 แสดงการจัดขาของ DS1307 แต่ละขา มีหน้าที่และการใช้งานดังนี้

Vcc, GND (ขา 8,4) ต่อกับไฟเลี้ยง +5V



รูปที่ 2.22 แสดงการจัดขาของ DS1307

Vbat (ขา 3) ใช้ต่อกับแบตเตอรี่ 3V เพื่อรักษาการทำงานของวงจรสร้างฐานเวลาของ DS1307 ให้คงอยู่ต่อไป แม้ว่าไม่มีไฟเลี้ยงจ่ายให้แก่ DS1307 ชนิดของแบตเตอรี่ที่เหมาะสมคือ แบตเตอรี่แบบลิเทียม ซึ่งมีความจุ 40mAh หรือมากกว่า จะสามารถรักษาข้อมูลได้นาน 10 ปีที่อุณหภูมิ 25 องศาเซลเซียส

SDA, SCL (ขา 5 และ 6) เป็นขาสำหรับเชื่อมต่อกับไมโครคอนโทรลเลอร์บนระบบบัส I2C

SQW/OUT (ขา 7) ที่ขานี้จะมีสัญญาณรูปสี่เหลี่ยมส่งออกมา โดยสามารถเลือกความถี่ได้ 1Hz, 4.096kHz, 8.192kHz และ 32kHz ในการใช้งานต้องต่อตัวต้านทาน 1k พูลอัพที่ขานี้ด้วย

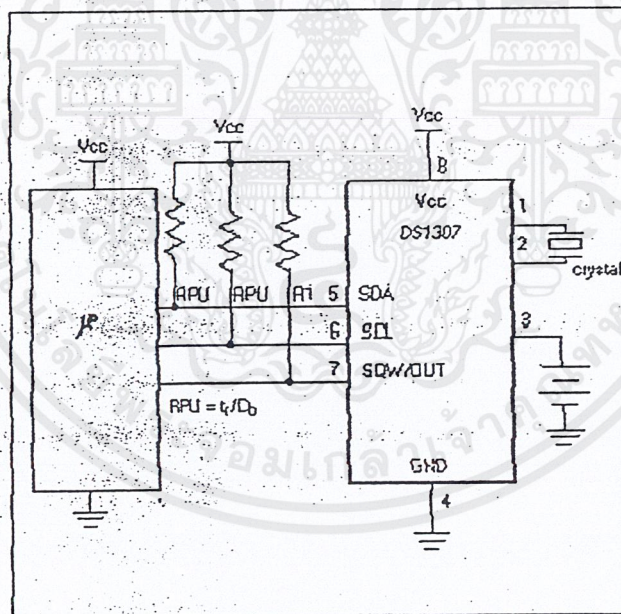
X1, X2 (ขา 1 และ 2) ใช้ต่อกับคริสตัลความถี่มาตรฐาน 32.768kHz เพื่อใช้เป็นฐานเวลาในการสร้างค่าเวลาจริงในการใช้งานต้องต่อคริสตัลเข้ากับขาทั้งสองนี้และที่แต่ละขาต้องต่อเก็บประจุ

ค่าต่ำๆ ประมาณ 15pF พร้อมกันจากราวด์ด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทำงานของ DS1307

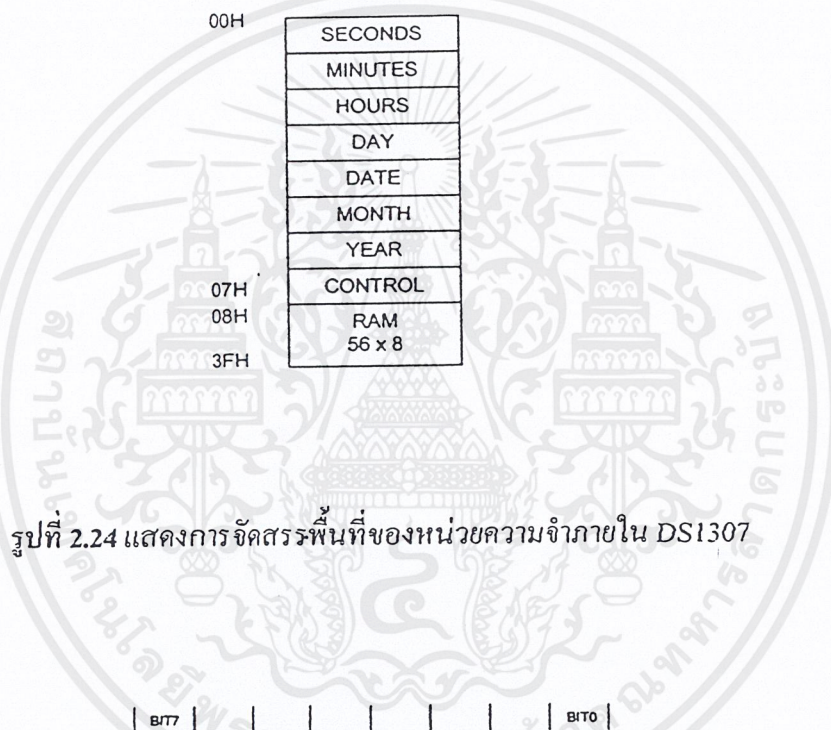
ไอซี DS1307 จัดการเชื่อมต่อในแบบบัส I2C โดยจะทำงานเป็นอุปกรณ์สเตลเฟสเสมอ ดังนั้น การติดต่อเพื่อใช้งานจึงต้องกำหนดรูปแบบตามที่กำหนดไว้ในการติดต่อแบบ I2C (มีรายละเอียดเพิ่มเติมในการทดลอง) ในรูปที่ 2.23 แสดงส่วนประกอบหลักที่สำคัญและไอะแกรมการทำงานของ DS1307 วงจรออสซิลเลเตอร์ถือเป็นหัวใจหลักของไอซี เนื่องจากเป็นจุดเริ่มต้นของการสร้างข้อมูลเวลาจริง ในขณะที่ DS1307 ทำงานที่ขา SQW/OUT จะมีสัญญาณพัลส์สี่เหลี่ยมส่งออกมาตลอดเวลา ในกรณีที่มีการอินาเบลวงจรกำเนิดสัญญาณพัลส์ที่รีจิสเตอร์ควบคุม ค่าความถี่ของสัญญาณนี้ สามารถเลือกได้ 4 ค่าคือ 1Hz, 4.096kHz, 8.192kHz และ 32kHz พร้อมกันนั้นก็จะมีการเก็บค่าของเวลาไว้ในหน่วยความจำอนโวลตาไทล์เริ่มซึ่งมีขนาดรวม 64 ไบต์ แต่จัดสรรให้เก็บข้อมูลเวลา 8 ไบต์ และเป็นหน่วยความจำสำหรับเก็บข้อมูลทั่วไปสำหรับผู้ใช้งานอีก 56 ไบต์



รูปที่ 2.23 แสดงส่วนประกอบหลักที่สำคัญและไอะแกรมการทำงานของ DS1307

การจัดสรรหน่วยความจำใน DS1307

ในรูป 2.24 แสดงการจัดสรรพื้นที่ของหน่วยความจำภายใน DS1307 พื้นที่ 7 ไบต์แรกตั้งแต่แอดเดรส 00H-06H เป็นพื้นที่ของรีจิสเตอร์ค่าเวลาใช้ในเก็บข้อมูลเกี่ยวกับเวลา ไบต์ต่อมาที่แอดเดรส 07H เป็นพื้นที่ของรีจิสเตอร์ควบคุมการทำงานของ DS1307 ในรูปที่ 2.25 แสดงรายละเอียดของรีจิสเตอร์ค่าเวลาและรีจิสเตอร์ควบคุมของ DS1307



รูปที่ 2.24 แสดงการจัดสรรพื้นที่ของหน่วยความจำภายใน DS1307

00H		CH		10 SECONDS				SECONDS				00-59	
		0		10 MINUTES				MINUTES				00-59	
		0		12 24	10 HR A/P	10 HR		HOURS				01-12 00-23	
		0		0	0	0	0	DAY				1-7	
		0		0	10 DATE			DATE				01-28/29 01-30 01-31	
		0		0	0	10 MONTH		MONTH				01-12	
		10 YEAR						YEAR				00-99	
07H		OUT		0	0	SQWE		0	0	RS1	RS0		

รูปที่ 2.25 แสดงรายละเอียดของรีจิสเตอร์ค่าเวลาและรีจิสเตอร์ควบคุมของ DS1307

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 2.25 แสดงรายละเอียดของรีจิสเตอร์ค่าเวลาและรีจิสเตอร์ควบคุมของ DS1307

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

รูปที่ 2.25 แสดงรายละเอียดของรีจิสเตอร์ค่าเวลาและรีจิสเตอร์ควบคุมของ DS1307

ด้วยการจัดสรรพื้นที่แบบนี้ ทำให้ผู้ใช้งานสามารถเรียกข้อมูลเวลาออกมาได้ตามที่ต้องการ โดยไม่จำเป็นต้องอ่านออกมาทั้งหมดก็ได้ค่าของเวลาที่หมดจะอยู่ในรูปของเลขฐานสิบ สำหรับการแสดงเวลาในรูปของชั่วโมง สามารถเลือกได้ว่าต้องการแบบ 12 หรือ 24 ชั่วโมง โดยกำหนดที่บิต 6 ของแอดเดรส 02H และเมื่อเลือกแบบ 12 ชั่วโมง ที่บิต 5 ในแอดเดรส เดียวกันจะใช้ในการแสดงค่า AM/PM โดยถ้าบิตนี้เป็น 1 หมายถึง ค่าชั่วโมงในขณะนี้เป็นเวลาหลังเที่ยงวัน ในกรณีที่แบบ 24 ชั่วโมง บิตนี้จะใช้ในการแสดงค่า 2 ชั่วโมง บิตนี้จะใช้ในการแสดงค่า 2 ของหลักสิบในหน่วยชั่วโมง

รีจิสเตอร์ควบคุม

มีแอดเดรสอยู่ที่ 07H มีรายละเอียดของแต่ละบิตดังนี้

OUT (Outcontrol) : ใช้ในการควบคุมระดับลอจิกที่ขา SQW/OUT ในกรณีที่คิสเอบิลการกำหนดสัญญาณสี่เหลี่ยม โดยถ้าบิตนี้เป็น 1 ที่ขา SQW/OUT ก็จะเป็น 1 ถ้าบิตนี้เป็น 0 ที่ขา SQW/OUT ก็จะเป็น 0

SQWE (Square Wave Enable) : ใช้ในการอินาเบิลวงจรกำหนดสัญญาณสี่เหลี่ยมที่ขา SQW/OUT ถ้าต้องการให้มีสัญญาณสี่เหลี่ยมออกให้กำหนดบิตนี้เป็น 1

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

RS1	RS0	ความถี่ของสัญญาณสี่เหลี่ยม
0	0	1Hz
0	1	4.906kHz
1	0	8.192kHz
1	1	32.768kHz

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โหมดการทำงานของ DS1307

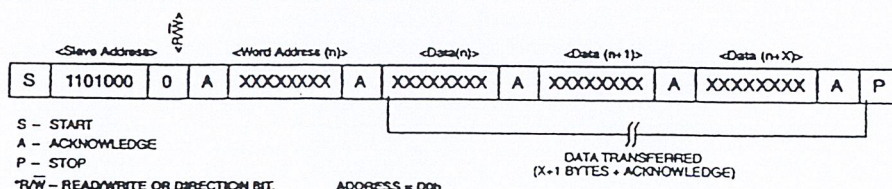
มีด้วยกัน 2 โหมดคือ โหมดเขียนข้อมูลและโหมดอ่านข้อมูล ในการใช้งาน DS 1307 ตามปกติจะใช้งานเฉพาะโหมดอ่านข้อมูลเท่านั้น เนื่องจากไมโครคอนโทรลเลอร์จะติดต่อกับ DS1307 เพื่ออ่านข้อมูลของเวลาไปใช้งาน โหมดการเขียนข้อมูลจะถูกใช้งานก็ต่อเมื่อต้องการตั้งค่าเวลาใหม่และต้องการเขียนข้อมูลลงไปใช้งาน โหมดการเขียนข้อมูลก่อนเพื่อกำหนดแอดเดรสที่ต้องการอ่านข้อมูล จากนั้นจึงเปลี่ยนโหมดการทำงานมาเป็นโหมดการอ่านข้อมูลต่อไป

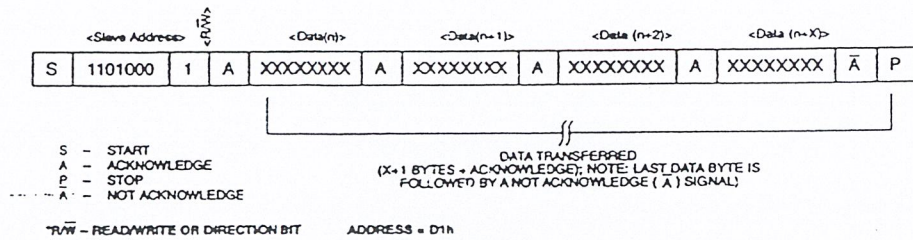
โหมดการเขียนข้อมูล

มีการเริ่มต้นเมื่อไมโครคอนโทรลเลอร์ ทำการกำหนดสถานะเริ่มต้น (START:S) จากนั้นส่งข้อมูลกำหนดแอดเดรส 1101000 ตามด้วยข้อมูลเลือกการเขียนนั่นคือค่า 0 จากนั้นจะรอการตอบรับจาก DS1307 ขึ้นตอนต่อมาคือ ส่งข้อมูลเพื่อเลือกแอดเดรสที่ต้องการเขียน จากนั้นรอการตอบรับจาก DS1307 เมื่อมีการตอบรับมาเรียบร้อยแล้ว ก็เริ่มทยอยเขียนข้อมูลลงไปครั้งละแอดเดรส หลังจากเขียนข้อมูลในแต่ละแอดเดรส จะต้องหยุดรอการตอบรับจาก DS1307 ทุกครั้งจึงจะสามารถเขียนข้อมูลต่อไปได้ เมื่อเขียนเรียบร้อยแล้วให้ส่งสถานะหยุด (STOP:S) เป็นอันสิ้นสุดการเขียนข้อมูล

โหมดการอ่านข้อมูล

มีรูปแบบแสดงในรูปที่ 2.26 เริ่มต้นการทำงานเหมือนกับโหมคการเขียนข้อมูล คือ ไมโครคอนโทรลเลอร์กำหนดสถานะเริ่มต้นแล้วส่งข้อมูลกำหนดแอดเดรสตามด้วยข้อมูล เลือกการอ่านซึ่งเท่ากับ 1 จากนั้นรอการตอบรับจาก DS1307 เมื่อตอบรับเรียบร้อยแล้ว DS1307 จะทยอยส่งข้อมูลออกมาให้ไมโครคอนโทรลเลอร์คราวละ 1 แอดเดรสหรือ 1 ไบต์ โดยแอดเดรสที่เลือกอ่านข้อมูลจะต้องมีการกำหนดมาก่อนล่วงหน้าด้วยโหมคการเขียนข้อมูลให้ทำการสร้างสถานะเริ่มต้นและส่งข้อมูลกำหนดแอดเดรสใหม่อีกครั้ง ตามด้วยเลือก โหมคการอ่านข้อมูล ข้อมูลที่ออกมาจาก DS1307 ก็จะเป็นข้อมูลจากแอดเดรสที่กำหนดไว้ ก่อนหน้านี

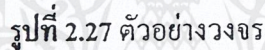




รูปที่ 2.26 โหมดการอ่าน/เขียนข้อมูล

การเชื่อมต่อDS1307กับไมโครคอนโทรลเลอร์ MCS-51

มีตัวอย่างวงจรแสดงในรูปที่ 2.27 เริ่มต้นการทำงานเหมือนกับโหมดการเขียนข้อมูลคือ ไมโครคอนโทรลเลอร์กำหนดสถานะเริ่มต้นแล้วส่งข้อมูลกำหนดแอดเดรสตามด้วยข้อมูลเลือกการอ่าน ซึ่งเท่ากับ 1 จากนั้นรอการตอบรับจาก DS1307 เมื่อตอบรับเรียบร้อยแล้ว DS1307 จะทยอยส่งข้อมูลออกมาให้ไมโครคอนโทรลเลอร์คราวละ 1 แอดเดรสหรือ 1 ไบต์โดยแอดเดรสที่เลือกอ่านข้อมูลจะต้องมีการกำหนดมาก่อนล่วงหน้าด้วยโหมดการเขียนข้อมูล วิธีการง่าย ๆ คือ เข้าสู่โหมดการเขียนข้อมูลก่อนเมื่อถึงจังหวะที่ต้องเขียนข้อมูล ให้ทำการสร้างสถานะเริ่มต้นและส่งข้อมูลกำหนดแอดเดรสใหม่อีกครั้งตามด้วยเลือกโหมดการอ่านข้อมูล ข้อมูลที่ออกมาจาก DS1307 ก็จะเป็นข้อมูลจากแอดเดรสที่กำหนดไว้ก่อนหน้านี้



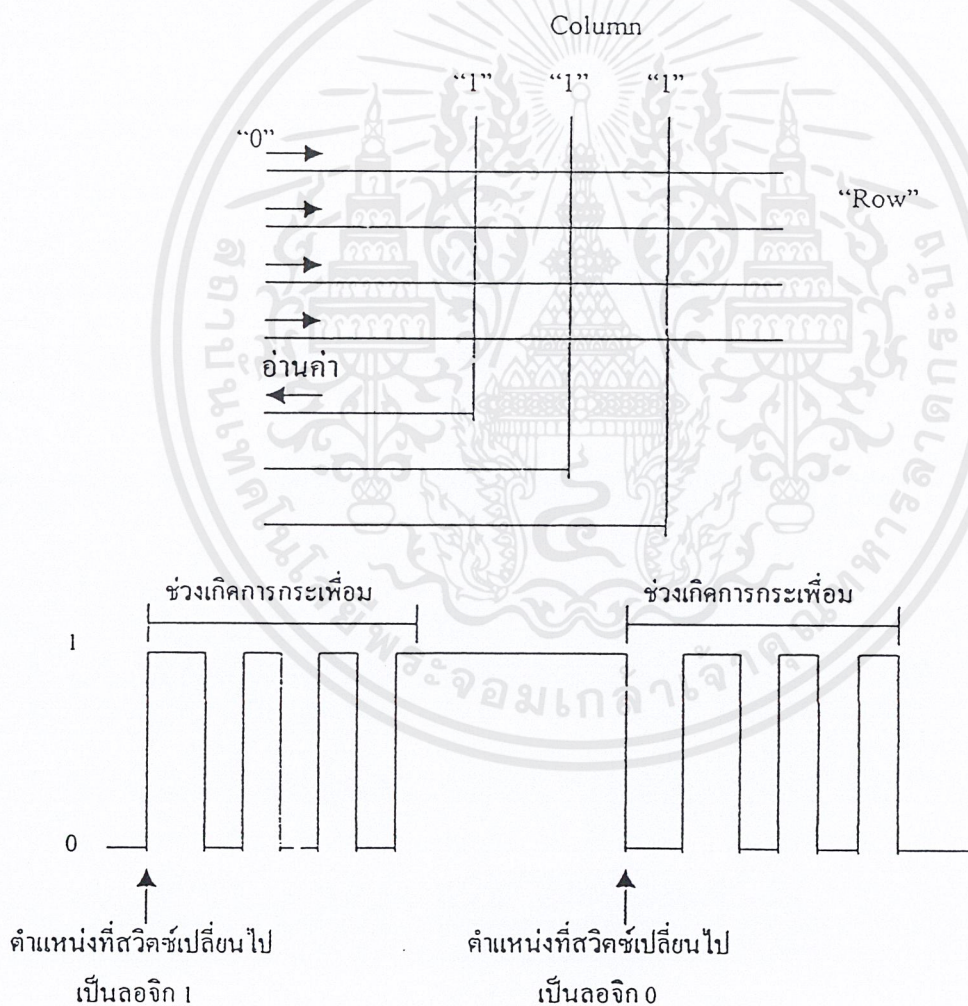
มีตัวอย่างวงจรแสดงในรูปที่ 2.27 จะเห็นได้ว่ามีลักษณะการต่อเหมือนกับอุปกรณ์บนระบบบัส I2C ตัวอื่น ๆ ทุกประการ และสามารถที่จะต่อไอซีทั้งหมดร่วมกันบนสาย SDA และ SCL ได้ เป็นการช่วยให้เห็นถึงความสามารถพิเศษของระบบบัส I2C ที่ผู้ใช้งานสามารถเชื่อมต่ออุปกรณ์ที่มีความต่างกันในหน้าที่การทำงานบนสายสัญญาณเดียวกันได้ ถึงการทดลองนี้ผู้ใช้งานสามารถเชื่อมต่ออุปกรณ์ระบบบัส I2C ได้ถึง 3 ตัว 3 ลักษณะการทำงานโดยใช้สายสัญญาณเพียง 2 เส้น

2.8 การเชื่อมต่อเครือข่าย

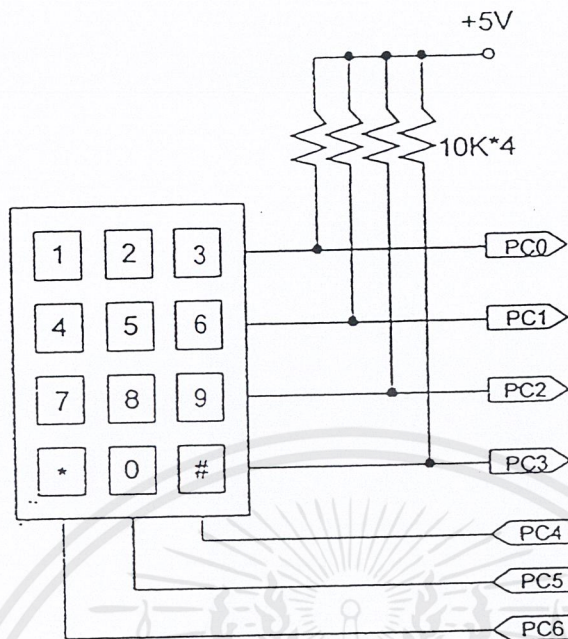
หลักการของโปรแกรมในการตรวจสอบสภาพการกวดวิชา จะใช้การสแกนคีย์บอร์ดโดยเริ่มต้นต้องทำให้สถานะลอจิกในแนวคอลัมน์ทุกเส้นสัญญาณมีค่า 1 ก่อน จากนั้นจึงเริ่มกำหนดสัญญาณ

ของลอจิกในแถวแรกเป็นลอจิก 0 เมื่อรับค่าสัญญาณของคอลัมน์เข้ามาตรวจสอบ หากพบว่ามีเส้นสัญญาณใดมีค่าลอจิกต่ำ แสดงว่ามีการกดสวิทช์ที่ต่ออยู่ระหว่างคอลัมน์นั้นกับเส้นสัญญาณใน แถวแรก หากไม่มีเส้นใดเป็นลอจิกต่ำเลยแสดงว่าไม่มีการกดคีย์บอร์ดในแถวนั้น จากนั้นก็ทำเช่นเดียวกันกับแถวอื่นๆจนครบทุกแถว

ในการนำไปใช้ต้องเพิ่มโปรแกรมหน่วงเวลาเข้าไปด้วยสักระยะหนึ่ง เนื่องจากเมื่อมีการกดสวิทช์ จะมีการกระเพื่อมของหน้าสัมผัส ซึ่งมีสภาพเหมือนการเปิดปิดหน้าสัมผัสหลายๆครั้งซึ่งมีผลทำให้โปรแกรมดำเนินการผิดพลาดได้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.29 การต่อคีย์บอร์ด

2.9 การใช้หน่วยความจำแบบ EEPROM

การใช้หน่วยความจำในการเก็บข้อมูลมีผลทำให้การทำงานทางด้านการประมวลผลสัญญาณดิจิทัลในปัจจุบันมีความสะดวกขึ้นมากเนื่องจากสามารถเก็บข้อมูลไว้ได้ จึงสามารถเพิ่มขีดความสามารถให้กับหน่วยกระบวนการทางดิจิทัลต่างๆ ดังเช่นในไมโครคอนโทรลเลอร์

การใช้หน่วยความจำแบบ Read Only Memory หรือ ROM จะมีประโยชน์มากกว่า Read Only Memory หรือ Ram เนื่องจาก สามารถเก็บข้อมูลไว้ได้แม้จะ ไม่มีไฟเลี้ยงวงจรทำให้สามารถเก็บข้อมูลไว้ได้ปลอดภัยมากกว่า แต่ก็มีข้อเสีย คือราคาจะแพงกว่ามาก

2.10 การส่งข้อมูลแบบไร้สาย

การ โมดูเลต คือ การนำเอาคลื่นความถี่สูงค่าหนึ่งที่เราเรียกว่า สัญญาณพาหะ (Carrier Signal) ไปรวมกับคลื่นความถี่ที่เราต้องการส่งซึ่งโดยทั่วไปจะมีอยู่ 3 แบบ คือ

1. แอมพลิจูด โมดูเลชัน (Amplitude Modulation : AM)
2. เฟิควนซี โมดูเลชัน (Frequency Modulation : FM)
3. เฟสโมดูเลชัน (Phase Modulation : PM)

การ โมดูเลชันแบบ FSK เป็นการ โมดูเลชันแบบ FM ชนิดหนึ่งโดยจะ ใช้กับการส่งสัญญาณดิจิทัล เนื่องจากสัญญาณแบบดิจิทัลจะมีสองสถานะคือ 1 กับ 0 เมื่อผ่านการ โมดูเลชันแบบ FSK แล้วจะได้ความถี่ที่แตกต่างกันสองความถี่แทนสองสถานะของสัญญาณดิจิทัล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.11 ขั้นตอนการติดต่อข้อมูลระหว่าง MICROCONTROLLER กับ DS18B20

รูปแบบการติดต่อ

เป็นการติดต่อ แบบสายสัญญาณเดี่ยว (1-wire serial bus) ถ่ายข้อมูลแบบ อะซิงโครนัส ในระดับบิต คล้ายระบบสื่อสารแบบ half-duplex ลักษณะของรูปสัญญาณที่ปรากฏบนสายสัญญาณในแต่ละช่วงของเวลา เรียก time slot

ลักษณะของ time slot แบบต่างๆ

1. การรีเซตและการตอบสนอง

เริ่มจากอุปกรณ์ master me ทำให้เกิดการรีเซตบนสายสัญญาณ ทำให้สายสัญญาณเป็น ลอจิกต่ำ อย่างน้อยเป็นเวลา 480 ไมโครวินาที และทำให้สายสัญญาณกลับมาเป็นลอจิกสูงถ้าหากมีอุปกรณ์สเลฟต่ออยู่ จะมีการตอบสนองสัญญาณรีเซตด้วยสัญญาณตอบสนอง โดยอุปกรณ์สเลฟจะทำให้สายสัญญาณเป็นลอจิกต่ำประมาณ 15-60 ไมโครวินาทีแล้วจึงทำกลับเป็นลอจิกสูง

2. การอ่านของอุปกรณ์มาสเตอร์หรือการเขียนของอุปกรณ์สเลฟอุปกรณ์มาสเตอร์ จะทำให้สายสัญญาณมีลอจิกต่ำประมาณ 1-15 ไมโครวินาทีจากนั้นทำมาเป็นลอจิกสูง ช่วงนี้อุปกรณ์สเลฟจะทำการส่งข้อมูลมาโดยถ้าสเลฟส่งข้อมูล 1 สเลฟจะทำให้สายสัญญาณเป็นลอจิกสูงต่อเนื่องไปอีก 45 ไมโครวินาทีถ้าสเลฟส่งข้อมูล 0 สเลฟจะทำให้สายสัญญาณเป็นลอจิกต่ำประมาณ 45 ไมโครวินาที หลังจากทีสเลฟส่งข้อมูลแล้วจะทำให้สายสัญญาณมีลอจิกสูง

3. การเขียนข้อมูลของอุปกรณ์มาสเตอร์หรือการอ่านข้อมูลของอุปกรณ์สเลฟอุปกรณ์มาสเตอร์จะทำให้สายสัญญาณเป็นลอจิกต่ำประมาณ 1-15 ไมโครวินาทีจากนั้นทำเป็นลอจิกสูงแล้วทำการเขียนข้อมูลทันทีถ้าเขียนข้อมูล 1 มาสเตอร์จะทำให้สายสัญญาณเป็นลอจิกสูงต่อเนื่องไปอีก 45 ไมโครวินาทีถ้าเขียนข้อมูล 0 มาสเตอร์จะทำให้สายสัญญาณเป็นลอจิกต่ำประมาณ 45 ไมโครวินาทีหลังจากที่อุปกรณ์มาสเตอร์ทำการเขียนเสร็จแล้วจะต้องทำให้สายสัญญาณเป็นลอจิกสูง

บทที่ 3

การออกแบบและการสร้าง

ในการออกแบบวงจรในระบบการเก็บข้อมูลนี้ สิ่งที่จะต้องพิจารณา คือ ขนาดที่เล็กเพื่อความสะดวกในการนำไปใช้งานในสถานที่ต่างๆ อุปกรณ์ที่ใช้ในวงจรต้องกินไฟน้อยเพื่อการประหยัดไฟและให้สามารถใช้แบตเตอรี่เป็นไฟเลี้ยงได้

การออกแบบแบ่งเป็นส่วนฮาร์ดแวร์และซอฟต์แวร์ ดังนี้

3.1 การออกแบบส่วนฮาร์ดแวร์ (Hardware)

ส่วนประมวลผลกลาง (Central Control Unit)

ไมโครคอนโทรลเลอร์เลือกใช้ไมโครคอนโทรลเลอร์เบอร์ AT89C52 ซึ่งมีหน่วยความจำแบบแฟลช (Flash Memory) ภายในตัว 8 กิโลไบต์ เหตุที่เลือกใช้เบอร์นี้ก็เพราะว่าภายในตัวมันเองมีหน่วยความจำแบบแฟลชขนาด 8 Kbyte ซึ่งสามารถเก็บข้อมูลโปรแกรมได้มากกว่า AT89C51

EEPROM เบอร์ 24LC256 ขนาด 32 Kbyte X 8 bit เอาไว้เก็บข้อมูลโดยสามารถคงค่าไว้ได้แม้ไม่มีไฟเลี้ยง

การส่งข้อมูลผ่านพอร์ตอนุกรม ใช้ไอซี MAX 232 เป็นตัวแปลงสัญญาณข้อมูลให้ได้มาตรฐานในการส่งแบบอนุกรม

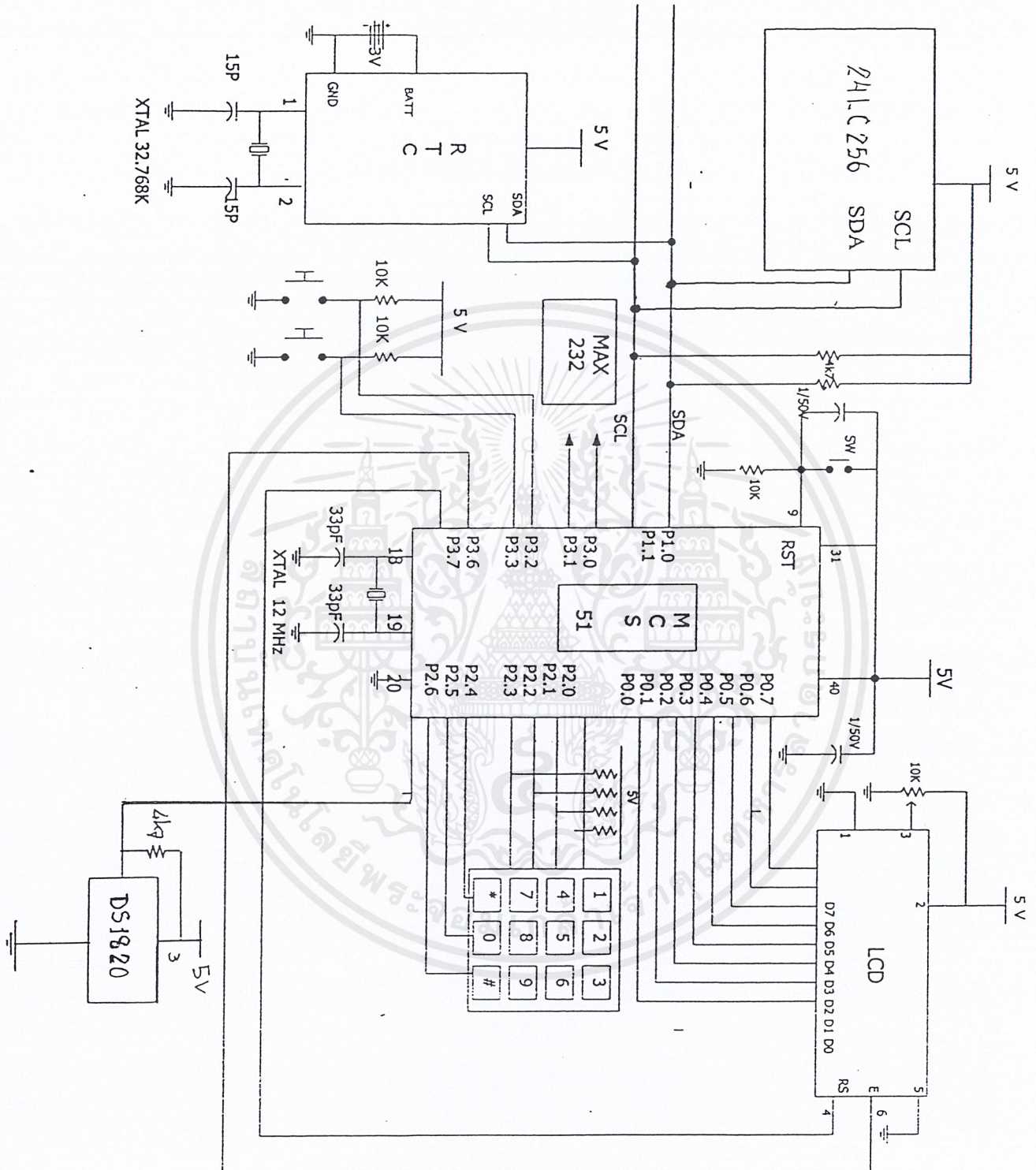
คีย์บอร์ด แบบ 12 ปุ่ม (4 Row 3 Column) สำหรับตั้งเวลาให้กับ Real Time Clock ในตอนที่เริ่มใช้งาน

เซ็นเซอร์อุณหภูมิ ใช้ไอซี DS1307 เนื่องจากช่วงอุณหภูมิที่ไอซีบอกค่าได้อยู่ในช่วงที่เหมาะสมและสัญญาณ

เรียลไทม์คล็อก ใช้ไอซี DS1307 เนื่องจากมีการติดต่อแบบบัส I²C และสามารถบอกได้ละเอียดตั้งแต่ค่าวินาทีจนถึงปี

การส่งข้อมูลไร้สาย ใช้ไอซี เบอร์ XR 2206 เป็นตัวโมดูลสัญญาณ และ ไอซี XR 2211 เป็นตัวดีโมดูลโดยจะทำการต่อวงจรตามค่าอ้างอิง

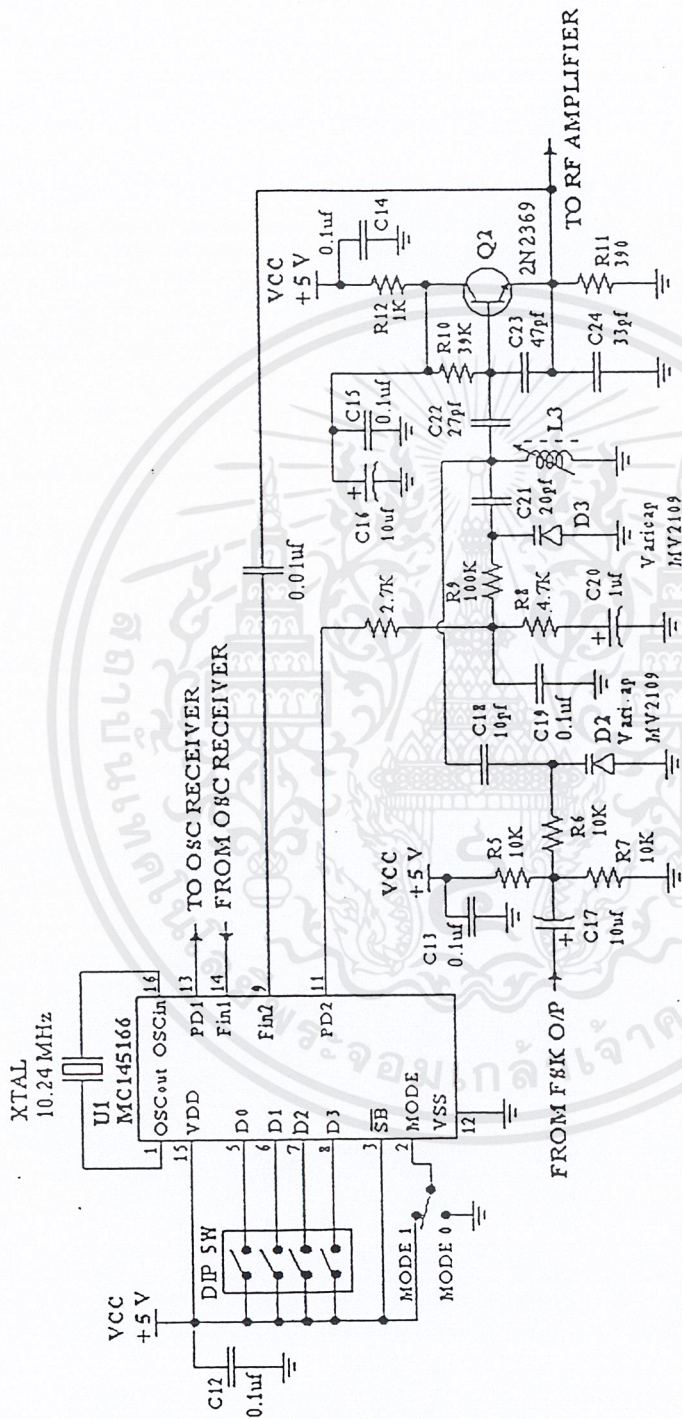
ส่วนประกอบหลักของวงจรทั้งหมดแสดงดังรูปที่ 3.1



รูปที่ 3.1 วงจรส่วนประกอบทั้งหมดของ Data Logger

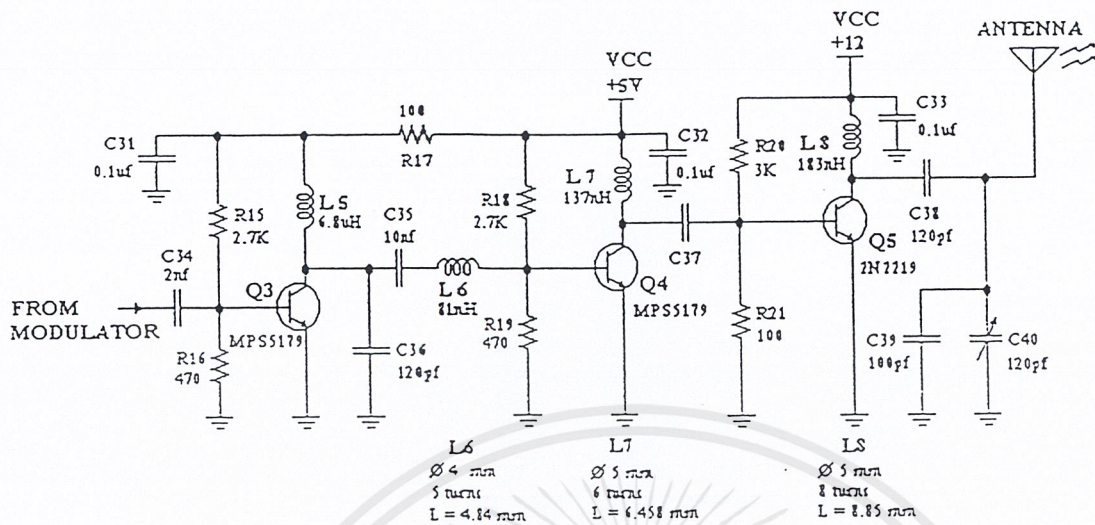
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

XR-2206 / XR-2211 300 AND 1200 BAUD MODEM WITH RS-232 INTERFACE

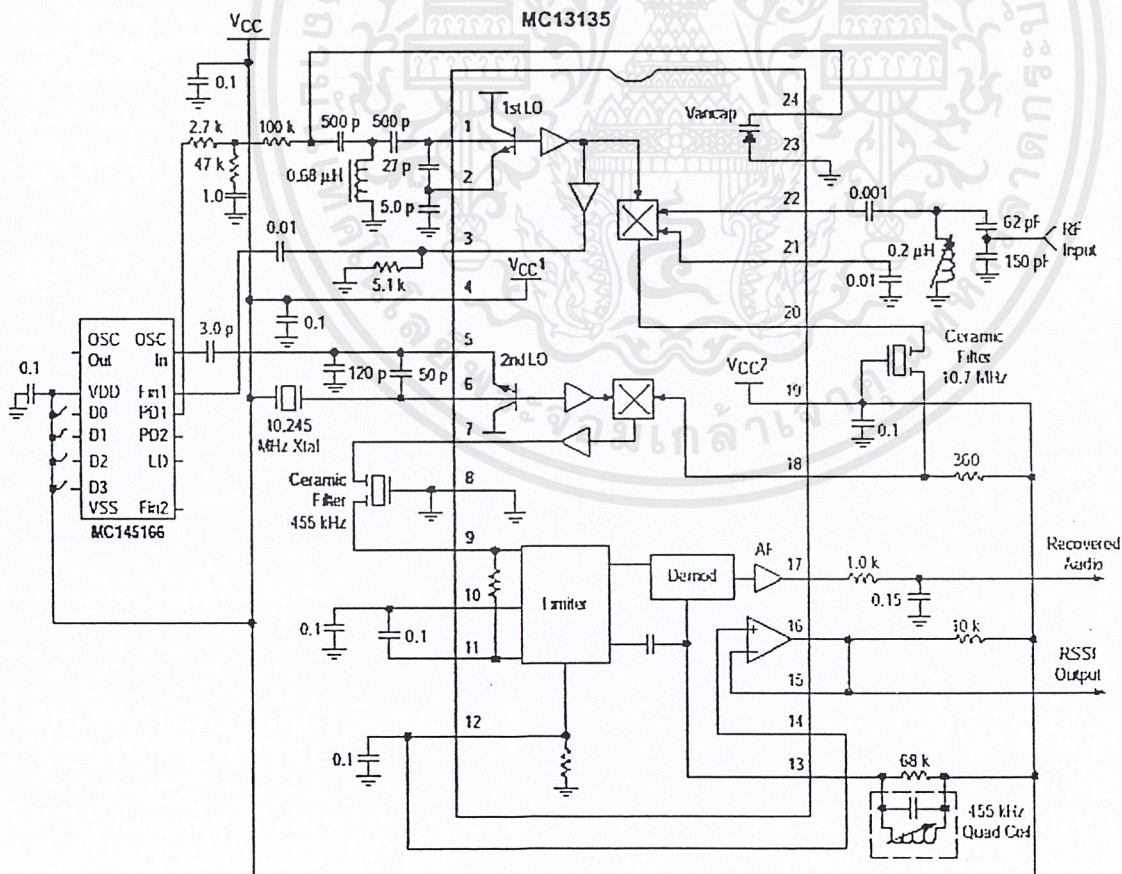


รูปที่ 3.3 วงจรภาคส่ง FM (วงจร Oscillator และ Modulator)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

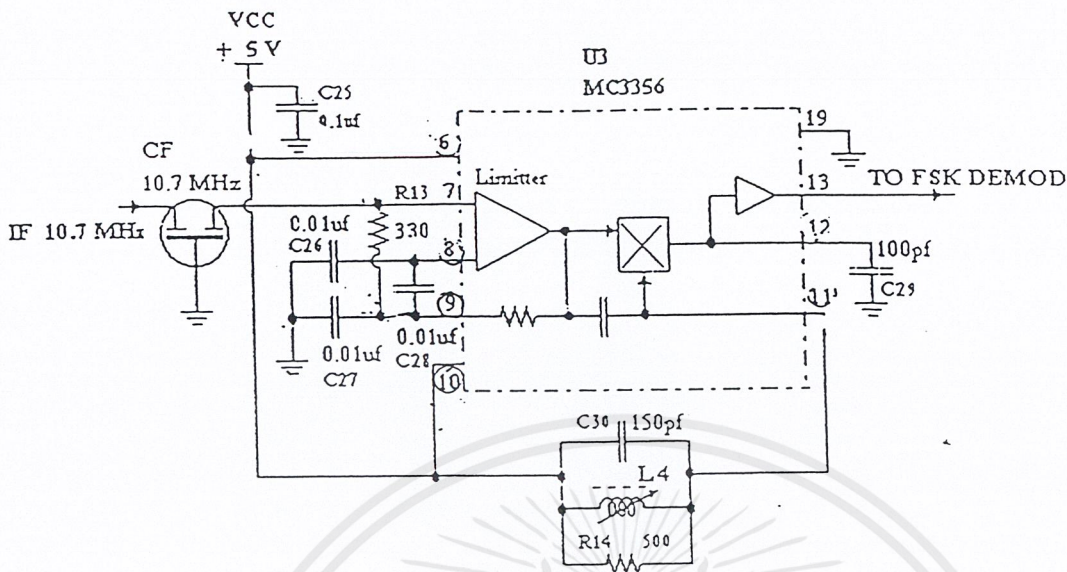


รูปที่ 3.4 วงจร RF Amplifier



รูปที่ 3.5 วงจรภาครับ FM

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.6 วงจร Limiter & Detector

3.2 ส่วนของซอฟต์แวร์และโปรแกรมควบคุม

3.2.1 โปรแกรมไมโครคอนโทรลเลอร์

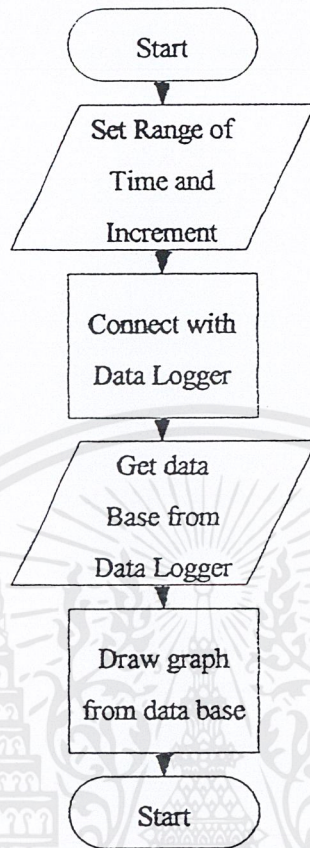
ในการเขียนโปรแกรมเพื่อควบคุมการทำงานของไมโครคอนโทรลเลอร์นี้ เราจำเป็นต้องใช้ภาษาแอสเซมบลี (Assembly) เฉพาะสำหรับไมโครคอนโทรลเลอร์เบอร์นี้ ซึ่งในการทำงานเราจะต้องเขียนโปรแกรมควบคุมส่วนหลักๆ ดังนี้

- การแสดงคีย์บอร์ด
- การโปรแกรม Real Time Clock
- การเก็บข้อมูลและการดึงข้อมูลออกจากหน่วยความจำ EEPROM
- การกำหนดตำแหน่งของอุปกรณ์ของ บัส I²C
- การติดต่อกับชุดไมโครคอมพิวเตอร์ผ่านพอร์ตอนุกรม
- การโปรแกรม LCD DISPLAY

3.2.2 โปรแกรมคอมพิวเตอร์

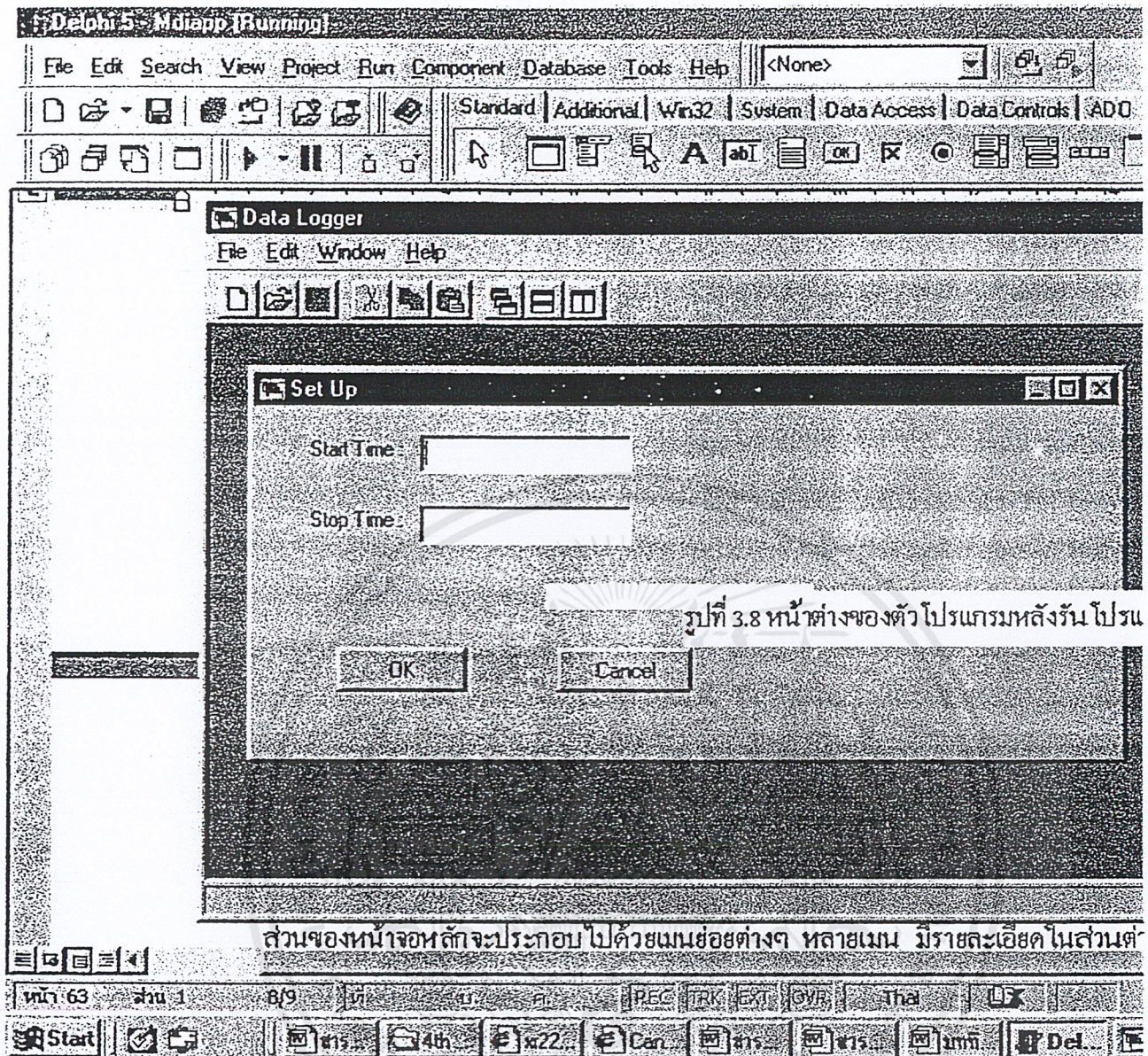
ซอฟต์แวร์ที่เขียนขึ้นใหม่ในโครงงานนี้มีการทำงานแบบ MDI (Multiple Document Interface) กล่าวคือเป็นโปรแกรมที่มีหน้าต่างการทำงานหลายหน้าต่าง โดยจะมีหน้าต่างใดหน้าต่างหนึ่งเป็นหลักผู้ใช้งานสามารถที่จะทำงานได้มากกว่า 1 หน้าต่างย่อยในเวลาเดียวกัน โครงสร้างของโปรแกรมนี้นี้ดังรูปที่ 3.2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.7 Flow Chart การทำงานของโปรแกรม

ซึ่งในโครงงานนี้ ได้ออกแบบให้ หน้าต่างหลัก คือ หน้าต่างชื่อ Data Logger และหน้าต่างย่อย ที่ใช้ในการแสดงผลและจัดการเกี่ยวกับข้อมูล คือ Graph โดยจะอ้างอิงจากรากฐานข้อมูลที่มาจาก Data Logger



ส่วนของหน้าจอหลักจะประกอบไปด้วยเมนูย่อยต่างๆ หลายเมนู มีรายละเอียดในส่วนต

ส่วนของหน้าจอหลักจะประกอบไปด้วยเมนูย่อยต่างๆ หลายเมนู มีรายละเอียดในส่วนต่างๆ ดังนี้

1. เมนู - นิว (Menu - New)

เป็นเมนูที่ทำการสร้างหน้าต่างกราฟแสดงผลขึ้นมาใหม่ ซึ่งสามารถบันทึกสัญญาณจากกราฟได้

2. ส่วนโพสิชันเคอร์เซอร์ (Position Cursor)

ใช้ในกรณีที่เรากำลังต้องการทราบค่าพิกัดในแนวแกน X และแนวแกน Y โดยในแนวแกน X จะเป็นแกนของเวลา และแนวแกน Y จะเป็นแกนของขนาดสัญญาณ

3. เมนู - สเกลเซตติง (Menu - Scale Setting)

จะปรากฏขึ้นมาในตอนที่เราเริ่มโปรแกรมโดยจะใช้ตั้งค่าเวลาเริ่มต้น , เวลาสิ้นสุด และการเพิ่มของเวลาในแกน X ของ Graph

4. เมนู - พิมพ์ (Menu - Print)

เมื่อต้องการที่จะทำการพิมพ์รูปสัญญาณออกจากเครื่องพิมพ์ ก็สามารถทำได้ โดยใช้เมนูนี้

บทที่ 4

การทดลองและผลการทดลอง

4.1 ผลการทดลองการแปลงอนาลอกเป็นดิจิทัล

ในการทดลอง A/D เบอร์ PCF8591

ทดสอบป้อนไฟ V_{in} เข้าไปซึ่งเป็นไฟตรงโดยจะตั้งแรงดัน Full scale ไว้ที่ 5 โวลต์ โดยตั้งค่าแรงดันอ้างอิง 5 โวลต์ (คุณสมบัติไอซีตัวนี้ได้จาก Data Sheets ในภาคผนวก) ได้ผลดังตารางที่ 4.1

V_{in}	D7	D6	D5	D4	D3	D2	D1	D0
0	0	0	0	0	0	0	0	0
1	0	0	1	1	0	0	1	1
2	0	1	1	0	0	1	0	1
3	1	0	0	1	1	0	0	1
4	1	1	1	0	0	0	0	0
5	1	1	1	1	1	1	1	1

ตารางที่ 4.1 ผลการทดลองวงจรแปลงสัญญาณอนาลอกเป็นดิจิทัล

จากตารางผลการทดลองเราสามารถที่จะจัดหาค่า Voltage จากสัญญาณ Digital ที่ทดลองได้ซึ่งจะเป็นตัวชี้ให้เห็นว่าวงจร A/D ของเรามีความผิดพลาดหรือไม่

จากสัญญาณ 8 บิต เราสามารถแยกระดับสัญญาณได้ $2^8 = 256$ Increments

จากค่า Full Scale Resolution = 256 divisions

Full Scale Resolution = 5 Volts.

เราจะคำนวณหาค่า Analog อินพุตจากค่า Digital ที่วัดได้ดังนี้

1. เมื่อ ค่า Digital ที่วัดได้ = 00110011

ค่า division = $32+16+2+1 = 51$

ดังนั้นค่า Analog I/P Voltage = $(51/256) \times 5 = 1.0$ Volt

[จะเห็นว่าค่าที่คำนวณมีค่าใกล้เคียงกับค่าแรงดันที่ป้อนเข้าไปคือ 1 Volt I/P]

$$\begin{aligned}
 2. \text{ เมื่อ ค่า Digital ที่วัดได้} &= 01100101 \\
 \text{ค่า division} &= 64+32+4+1 = 101 \\
 \text{ดังนั้นค่า Analog I/P Voltage} &= (101/256) \times 5 = 1.98 \text{ Volt}
 \end{aligned}$$

$$\begin{aligned}
 3. \text{ เมื่อ ค่า Digital ที่วัดได้} &= 10011001 \\
 \text{ค่า division} &= 128+16+8+1 = 153 \\
 \text{ดังนั้นค่า Analog I/P Voltage} &= (153/256) \times 5 = 3.0 \text{ Volt}
 \end{aligned}$$

$$\begin{aligned}
 4. \text{ เมื่อ ค่า Digital ที่วัดได้} &= 11100000 \\
 \text{ค่า division} &= 128+64+32 = 224 \\
 \text{ดังนั้นค่า Analog I/P Voltage} &= (224/256) \times 5 = 4.392 \text{ Volt}
 \end{aligned}$$

จะเห็นว่าจากการทดลองป้อนค่าอินพุตโวลต์เตจ เข้าที่อินพุตและการคำนวณค่าควอลท์เตจ จากค่าดิจิตอล จะมีค่าใกล้เคียงกันมาก ซึ่งในการทดลองมีค่าแรงดันต่ำๆ ในส่วนของบิตต่ำๆ จะมีสัญญาณไม่อยู่นิ่ง แต่ก็ไม่ค่อยมีผลมากนักเนื่องจากมีค่านัยสำคัญต่ำๆ และช่วงใช้งานของวงจรนี้จะ ขึ้นอยู่กับแรงดัน

4.2 ผลการทดลองการเทียบค่าอุณหภูมิที่วัดได้จาก DS1820 กับค่าอุณหภูมิจริง

โดยในการทดลองนี้จะวัดค่าอุณหภูมิในสถานที่ที่ต่างกัน โดยจะเทียบกับค่าที่วัดได้จาก เทอร์โมมิเตอร์ได้ผลดังตารางที่ 4.2

สถานที่	ค่าที่ได้จากเทอร์โมมิเตอร์	ค่าที่ได้จาก IC	ค่า error (%)
ในที่ร่ม	32	31.5	1.56
กลางแจ้ง	37	36.5	1.35
ในห้องที่มีเครื่องปรับอากาศ	23	23	0

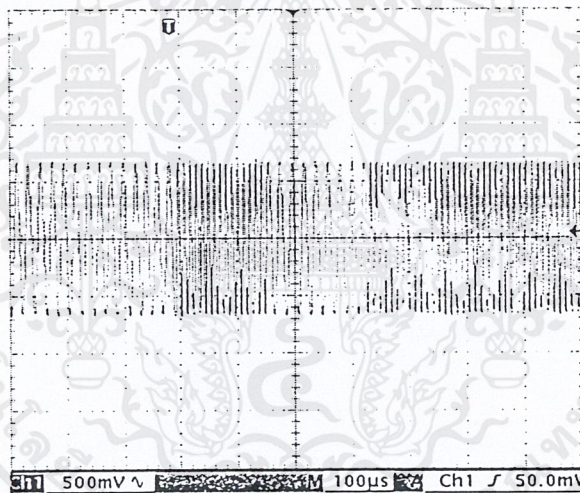
ตารางที่ 4.2 แสดงการเทียบค่าอุณหภูมิที่วัดได้จาก IC กับที่ได้จากเทอร์โมมิเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากตารางจะเห็นได้ค่าที่ได้จากไอซีจะมีความใกล้เคียงกับค่าที่วัดได้จากเทอร์โมมิเตอร์มาก แม้จะผิดพลาดก็เป็นค่าที่ยอมรับได้ เนื่องจากการวัดนี้ไม่ได้นำไปใช้กับงานที่ต้องการความละเอียดสูงและเน้นการประหยัดจึงถือว่าค่าที่ได้คือค่าที่ใช้ได้

4.3 FSK Modulation ด้วย XR2206

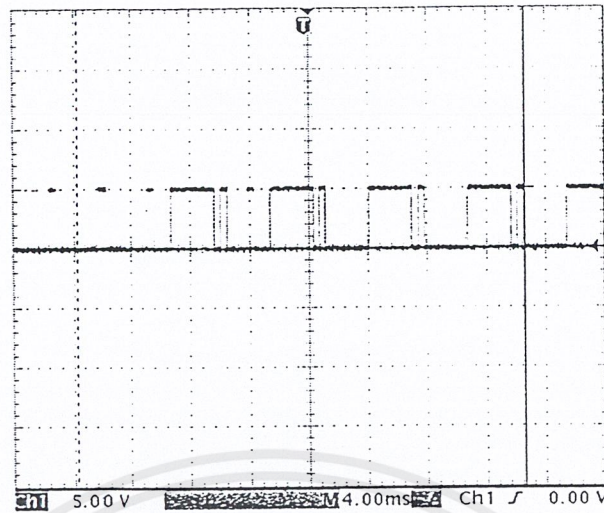
นำสัญญาณข้อมูลอนุกรม 8 Bit ที่ได้ไปเข้าขาที่ 9 ของ XR2206 ซึ่งก็คือขา I/P แล้ววัดสัญญาณที่ขา O/P (ขา 2) จะได้สัญญาณ FSK ดังรูป ถ้ามีการส่งลอจิก 1 จะได้ความถี่ที่ต่ำ แต่ถ้ามีการส่งลอจิกสูงจะได้ความถี่ที่สูง



รูปที่ 4.1 สัญญาณ FSK ที่ได้จากเอาต์พุตของ XR2206

4.4 FSK Demodulation ด้วย XR2211

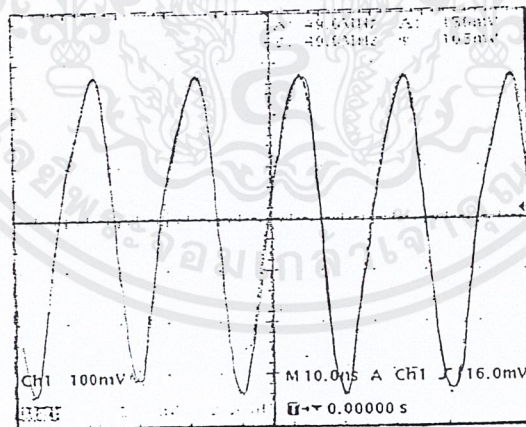
รูปสัญญาณเอาต์พุตที่ได้จากขา 7 ของ XR2211 จะเหมือนกับสัญญาณอินพุตของ XR2206 ดังรูปที่ 4.2



รูปที่ 4.2 สัญญาณเอาต์พุตที่ได้จาก XR2211

4.5 FM Transmitter

เมื่อนำสัญญาณที่ได้จาก XR2206 ไปทำการ Modulation แบบ FM ที่ความถี่ 49 MHz จะได้สัญญาณที่เอาต์พุตดังรูป



รูปที่ 4.3 สัญญาณที่ได้หลังจาก FM Modulation

4.6 RF Amplifier

จากการใช้สโคปวัดค่าจะทำให้หาค่าดังขยายได้คือ

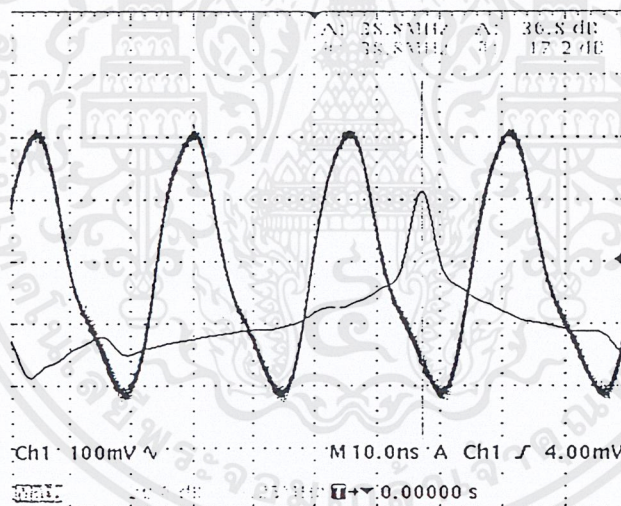
$$V_{in} = 64.5 \text{ mVpp}$$

$$V_{out} = 365.7 \text{ mVpp}$$

$$A_v = 365.7/64.5 = 5.67$$

4.7 FM Receiver

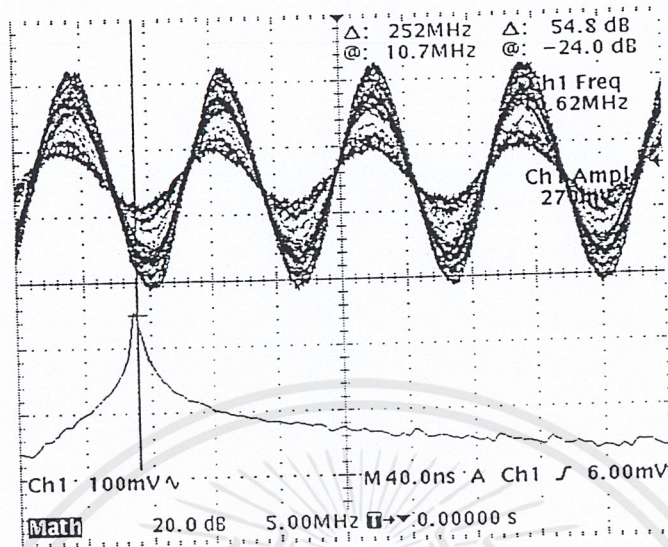
สัญญาณเอาต์พุตที่ได้มีดังรูป



รูปที่ 4.4 สัญญาณเอาต์พุตจากภาครับ FM

4.8 สัญญาณ IF

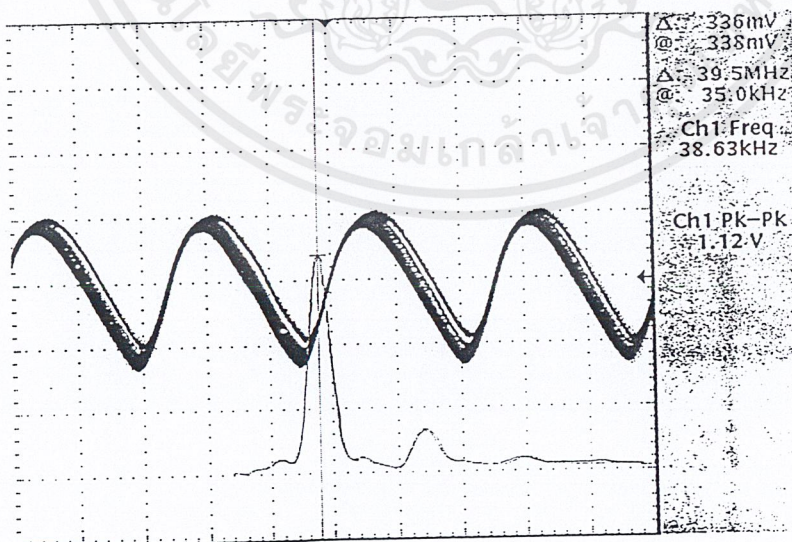
วัดสัญญาณ IF 10.7 MHz ที่ได้จากวงจรมิกเซอร์หลังจากผ่านเซรามิกฟิลเตอร์ 10.7 MHz ที่ต่ออยู่กับขาที่ 20 ของ MC13135



รูปที่ 4.5 สัญญาณ IF 10.7 MHz

4.9 สัญญาณข้อมูลหลังการ Demodulation IF

สัญญาณที่วัดได้จากขาที่ 13 ของ MC3356 ซึ่งเป็นสัญญาณที่วัดได้หลังจากการ คีมอคูเลต โดยวิธี Quarature Detector ซึ่งจะทำให้ได้สัญญาณ FSK ออกมา



รูปที่ 4.6 สัญญาณ FSK ที่ได้หลังจากการคีมอคูเลต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

สรุปผลและวิจารณ์

โครงการนี้มีลักษณะของการสุ่มเก็บข้อมูลเป็นค่าเวลาตามที่ใช้จะกำหนด โดยขอบเขตในการเก็บค่าขึ้นอยู่กับ A/D จึงทำให้โครงการนี้มีข้อจำกัดที่รายละเอียดของสัญญาณเป็น 8 บิต ในช่วงค่าของแรงดันที่เข้ามาจะต้องอยู่ในช่วง 0 – 5 Volt และสามารถเก็บข้อมูลที่ได้รับเข้ามาได้ 32 Kbytes ความสามารถในการตั้งค่าช่วงของเวลาในการเก็บข้อมูล และสามารถตั้งค่าเวลานาฬิกาได้

การส่งข้อมูลแบบ FM มีข้อดีคือ สามารถส่งได้ระยะทางไกล ส่วนข้อเสียคือจะมีสัญญาณรบกวนมากเมื่อเทียบกับวงจรที่ใช้งานกับความถี่ต่ำ ซึ่งจะทำให้เกิดความผิดพลาดได้ง่าย

แนวทางในการพัฒนา ควรเพิ่มความสามารถให้สามารถรับอินพุตได้หลายเซนแนล และสามารถเก็บข้อมูลรายละเอียดที่มากขึ้น โดยพัฒนาการเขียนโปรแกรม และเพิ่มวงจรตรวจจับสัญญาณนอก นอกจากนี้ยังควรเพิ่มหน่วยความจำให้มากขึ้นกว่านี้เพื่อสามารถที่จะเก็บข้อมูลได้มากขึ้น และที่สำคัญเป็นอย่างยิ่งคือการเพิ่มระยะในการส่งให้ได้มากขึ้นแต่ก็เป็นการทำได้ยากเนื่องจากที่ความถี่สูงนั้นค่าต่างๆที่คลาดเคลื่อนไปแม้เพียงเล็กน้อย เช่น ค่าความเหนี่ยวนำซึ่งต้องทำเองไม่สามารถหาซื้อมาได้ มีผลต่อการส่งสัญญาณมาก จนสามารถทำให้การส่งสัญญาณล้มเหลวได้

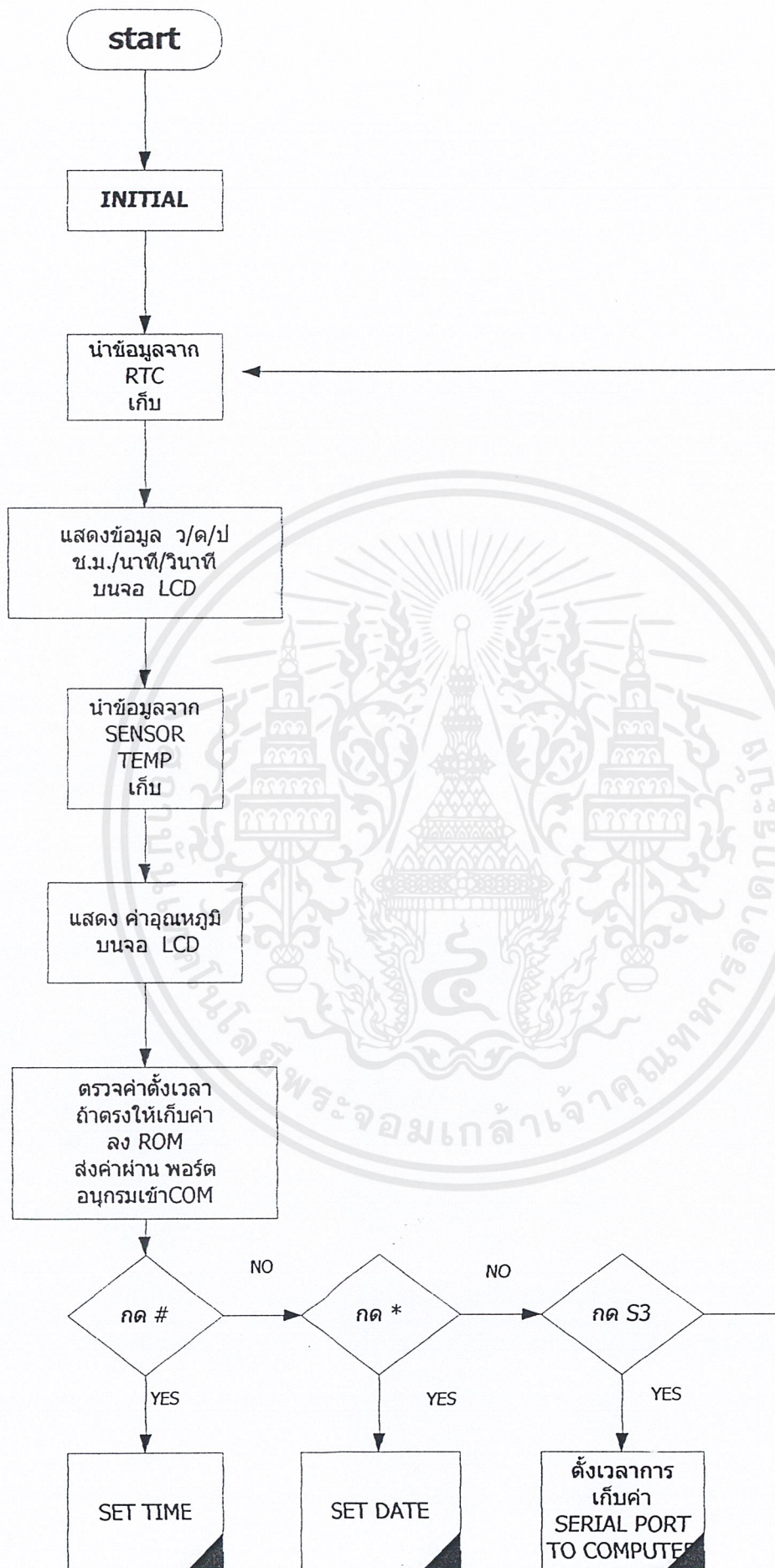
เอกสารอ้างอิง

1. กฤษฎา วิศวธีรานนท์ ,เขียน เล่น ใช้ไอซีคิดจคคอด ,บริษัท ซีเอ็ดดูเคชั่น จำกัด
2. กมลมาศ ถิ่นกรกิจกร ,คู่มือ Borland Delphi 5 ฉบับสมบูรณ์ ,กรุงเทพฯ :สำนักพิมพ์โปรวิชั่น ,2543
3. ชัยวัฒน์ ลิ้มพรจิตรวิไล , เร็วรู้ และ ปฏิบัติการไมโครคอนโทรลเลอร์ MCS-51, บริษัท อินโนเวทีฟ เอ็กเพอริเมนต์ จำกัด ,2542





เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



14/03/02	11:52	31.5°C
14/03/02	11:53	32.0°C
14/03/02	11:54	32.0°C
14/03/02	11:55	32.0°C
14/03/02	11:56	32.0°C
14/03/02	11:57	32.0°C
14/03/02	11:58	32.0°C
14/03/02	11:59	32.0°C
14/03/02	12:00	32.0°C
14/03/02	12:01	32.0°C
14/03/02	12:02	32.0°C
14/03/02	12:03	32.0°C
14/03/02	12:04	32.0°C
14/03/02	12:05	32.0°C
14/03/02	12:06	32.0°C
14/03/02	12:07	32.0°C
14/03/02	12:08	32.0°C
14/03/02	12:09	32.0°C
14/03/02	12:10	32.0°C
14/03/02	12:11	32.0°C
14/03/02	12:12	32.0°C
14/03/02	12:13	32.0°C
14/03/02	12:14	32.0°C
14/03/02	12:15	32.0°C
14/03/02	12:16	32.0°C

รูปการ ส่งข้อมูล ไปเก็บในคอมพิวเตอร์

Source Code

ONEWIRE	BIT	P2.7
LCD_EN	BIT	P3.6
LCD_RS	BIT	P3.7
SDA	BIT	P1.0
SCL	BIT	P1.1
KPAD_ROW0	BIT	P2.0
KPAD_ROW1	BIT	P2.1
KPAD_ROW2	BIT	P2.2
KPAD_ROW3	BIT	P2.3
KPAD_COL2	BIT	P2.4
KPAD_COL1	BIT	P2.5
KPAD_COL0	BIT	P2.6
SET_TLOG_SW	BIT	P3.2
SET_TIME_SW	BIT	P3.3
F	EQU	02FH
I2C_ACK	BIT	F.0
FLAG	EQU	02eH
BUSY	BIT	FLAG.0
LCD_ADDR	EQU	030H
LCD_DATA	EQU	031H
LCD_PTR	EQU	032H
I2C_ADDR	EQU	033H
I2C_DATA	EQU	034H
KPAD_DATA	EQU	035H
BUFFER	EQU	036H
ONEWIRE_DATA	EQU	039H
TEMP	EQU	03aH

SECONDS	EQU	040H	
MINUTES		EQU	041H
HOURS	EQU	042H	
DAY	EQU	043H	
DATE	EQU	044H	
MONTH		EQU	045H
YEAR	EQU	046H	
CONTROL	EQU	047H	
SH	EQU	048H	;@@@
SM	EQU	049H	
M	EQU	04AH	
CT	EQU	04BH	
CM	EQU	04CH	
TIMEH	EQU	04DH	
PHI	EQU	04EH	
PLO	EQU	04FH	
SE_DATA		EQU	050H
;-----			
RTC_ID		EQU	11010000B
;-----			
ID_EEP	EQU	0A0H	;READ &

```

ORG 0000H
MOV P0,#00000000B
MOV P1,#11111011B ;clr p1.2
MOV P3,#11111111B
MOV SM,#00H
MOV SH,#00H
MOV CT,#00H

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้


```

MOV CM,#00H
MOV TIMEH,#00H
MAIN:  ACALLINIT_LCD
MOV TMOD,#021H
MOV TH1,#0FDH
MOV TL1,#0FDH
SETB TR1
MOV SCON,#040H

MOV LCD_ADDR,#000H
ACALLSET_ADDR_LCD
MOV DPTR,#TITLE_1
ACALLWRLINE_LCD
MOV LCD_ADDR,#040H
ACALLSET_ADDR_LCD
MOV DPTR,#TITLE_2
ACALLWRLINE_LCD
ACALLDELAY_1S
ACALLDELAY_1S
ACALLLCD_CLR

LOOP:  ACALLRTC_RD          ;READ RTC
ACALLDS1820_T_RD ;READ TEMP
ACALLCHECK_T_LOG          ;@@@

MOV LCD_ADDR,#000H
ACALLSET_ADDR_LCD
MOV LCD_DATA,#' '
ACALLWRCHAR_LCD
MOV LCD_DATA,DATE
ACALLBCD2LCD

```



```

MOV LCD_DATA,#' '
ACALL WRCHAR_LCD
MOV A,MONTH
CJNE A,#010H,WR_CHK_M_1
MOV A,#00AH
AJMP WRITE_M_NX
WR_CHK_M_1: CJNE A,#011H,WR_CHK_M_2
MOV A,#00BH
AJMP WRITE_M_NX
WR_CHK_M_2: CJNE A,#012H,WRITE_M_NX
MOV A,#00CH
WRITE_M_NX:MOV LCD_PTR,A
MOV DPTR,#MONTH_JAN
ACALL WR3CHAR_LCD
MOV LCD_DATA,#' '
ACALL WRCHAR_LCD
MOV LCD_DATA,YEAR
ACALL BCD2LCD
MOV LCD_DATA,#' '
ACALL WRCHAR_LCD
MOV A,HOURS
ANL A,#00110000B
JZ WRITE_TIME_HN
SWAP A
ADD A,#030H
AJMP WRITE_TIME_HH
WRITE_TIME_HN: MOV A,#' '
WRITE_TIME_HH: MOV LCD_DATA,A
ACALL WRCHAR_LCD
MOV A,HOURS

```

```
ANL A,#00001111B
```



```

        ADD    A,#030H
        MOV    LCD_DATA,A
        ACALL WRCHAR_LCD
        MOV    A,SECONDS
        ANL    A,#001H
        JNZ    WRITE_SPACE
        MOV    LCD_DATA,#':'
        ACALL WRCHAR_LCD
        AJMP   WRITE_MINUTES
WRITE_SPACE:  MOV    LCD_DATA,#' '
        ACALL WRCHAR_LCD
WRITE_MINUTES: MOV    LCD_DATA,MINUTES
        ACALL BCD2LCD
        MOV    LCD_ADDR,#040H           ;temp ##
        ACALL SET_ADDR_LCD
        MOV    DPTR,#SCR_TEMP
        ACALL WRLINE_LCD
        MOV    LCD_ADDR,#047H
        ACALL SET_ADDR_LCD
        MOV    A,TEMP
        CLR    C
        RRC    A
        MOV    LCD_DATA,A
        ACALL HEX2LCD
        MOV    LCD_ADDR,#04bH
        ACALL SET_ADDR_LCD
        MOV    A,TEMP
        JNB    ACC.0,WRITE_0C
        MOV    LCD_DATA,#'5'
        AJMP   WRITE_NEXT

```


WRITE_0C: MOV LCD_DATA,#'0'

WRITE_NEXT: ACALL WRCHAR_LCD

; ACALLDELAY_100MS

JNB SET_TLOG_SW,SET_LOG

ACALLGET_KPAD ;scan *,#

MOV A,KPAD_DATA

CJNE A,#10,OU

AJMP SET_DATE

OU: CJNE A,#12,OUT

AJMP SET_TIME

OUT: NOP

AJMP LOOP

SET_LOG: ACALLSET_T_LOG ;@@@

SET_DATE: ACALLLCD_CLR

MOV LCD_ADDR,#000H ;ตั้งค่า วันเดือนปี

ACALLSET_ADDR_LCD

MOV DPTR,#SCR_SET_DATE

ACALLWRLINE_LCD

MOV LCD_ADDR,#040H

ACALLSET_ADDR_LCD

ACALLLCD_BLINK

ACALLWAIT_KEYPRESS

MOV BUFFER,KPAD_DATA

MOV A,KPAD_DATA

ADD A,#030H

MOV LCD_DATA,A

เอกสารนี้เป็นเอกสารสงวนลิขสิทธิ์การใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้


```

ACALL WRCHAR_LCD
ACALL LCD_BLINK
ACALL WAIT_KEY
ACALL WAIT_KEYPRESS
MOV  BUFFER+1,KPAD_DATA
MOV  A,KPAD_DATA
ADD  A,#030H
MOV  LCD_DATA,A
ACALL WRCHAR_LCD
ACALL WAIT_KEY
MOV  A,BUFFER
ANL  A,#00FH
SWAP A
MOV  B,A
MOV  A,BUFFER+1
ANL  A,#00FH
ADD  A,B
MOV  DATE,A
MOV  LCD_ADDR,#043H
ACALL SET_ADDR_LCD
ACALL LCD_BLINK
ACALL WAIT_KEYPRESS
MOV  BUFFER,KPAD_DATA
MOV  A,KPAD_DATA
ADD  A,#030H
MOV  LCD_DATA,A
ACALL WRCHAR_LCD
ACALL LCD_BLINK
ACALL WAIT_KEY
ACALL WAIT_KEYPRESS
MOV  BUFFER+1,KPAD_DATA

```



```

MOV  A,KPAD_DATA
ADD  A,#030H
MOV  LCD_DATA,A
ACALL WRCHAR_LCD
ACALL WAIT_KEY
MOV  A,BUFFER
ANL  A,#00FH
SWAP A
MOV  B,A
MOV  A,BUFFER+1
ANL  A,#00FH
ADD  A,B
MOV  MONTH,A
MOV  LCD_ADDR,#046H
ACALL SET_ADDR_LCD
ACALL LCD_BLINK
ACALL WAIT_KEYPRESS
MOV  BUFFER,KPAD_DATA
MOV  A,KPAD_DATA
ADD  A,#030H
MOV  LCD_DATA,A
ACALL WRCHAR_LCD
ACALL LCD_BLINK
ACALL WAIT_KEY
ACALL WAIT_KEYPRESS
MOV  BUFFER+1,KPAD_DATA
MOV  A,KPAD_DATA
ADD  A,#030H
MOV  LCD_DATA,A
ACALL WRCHAR_LCD
ACALL WAIT_KEY

```



```

MOV  A,BUFFER
ANL  A,#00FH
SWAP A
MOV  B,A
MOV  A,BUFFER+1
ANL  A,#00FH
ADD  A,B
MOV  YEAR,A
ACALL RTC_WR
ACALL LCD_CLR
AJMP LOOP

```

SET_TIME:

```

ACALL LCD_CLR
MOV  LCD_ADDR,#000H ;ตั้งค่าเวลา
ACALL SET_ADDR_LCD
MOV  DPTR,#SCR_SET_TIME
ACALL WRLINE_LCD
MOV  LCD_ADDR,#040H
ACALL SET_ADDR_LCD
ACALL LCD_BLINK
ACALL WAIT_KEYPRESS
MOV  BUFFER,KPAD_DATA
MOV  A,KPAD_DATA
ADD  A,#030H
MOV  LCD_DATA,A
ACALL WRCHAR_LCD
ACALL LCD_BLINK
ACALL WAIT_KEY
ACALL WAIT_KEYPRESS
MOV  BUFFER+1,KPAD_DATA

```



```

MOV A,KPAD_DATA
ADD A,#030H
MOV LCD_DATA,A
ACALLWRCHAR_LCD
ACALLWAIT_KEY
MOV A,BUFFER
ANL A,#00FH
SWAP A
MOV B,A
MOV A,BUFFER+1
ANL A,#00FH
ADD A,B
MOV HOURS,A
MOV LCD_ADDR,#043H
ACALLSET_ADDR_LCD
ACALLLCD_BLINK
ACALLWAIT_KEYPRESS
MOV BUFFER,KPAD_DATA
MOV A,KPAD_DATA
ADD A,#030H
MOV LCD_DATA,A
ACALLWRCHAR_LCD
ACALLLCD_BLINK
ACALLWAIT_KEY
ACALLWAIT_KEYPRESS
MOV BUFFER+1,KPAD_DATA
MOV A,KPAD_DATA
ADD A,#030H
MOV LCD_DATA,A
ACALLWRCHAR_LCD
ACALLWAIT_KEY

```



```

MOV  A,BUFFER
ANL  A,#00FH
SWAP A
MOV  B,A
MOV  A,BUFFER+1
ANL  A,#00FH
ADD  A,B
MOV  MINUTES,A
MOV  LCD_ADDR,#046H
ACALLSET_ADDR_LCD
ACALLLCD_BLINK
ACALLWAIT_KEYPRESS
MOV  BUFFER,KPAD_DATA
MOV  A,KPAD_DATA
ADD  A,#030H
MOV  LCD_DATA,A
ACALLWRCHAR_LCD
ACALLLCD_BLINK
ACALLWAIT_KEY
ACALLWAIT_KEYPRESS
MOV  BUFFER+1,KPAD_DATA
MOV  A,KPAD_DATA
ADD  A,#030H
MOV  LCD_DATA,A
ACALLWRCHAR_LCD
ACALLWAIT_KEY
MOV  A,BUFFER
ANL  A,#00FH
SWAP A
MOV  B,A
MOV  A,BUFFER+1

```



```

ANL  A,#00FH
ADD  A,B
MOV  SECONDS,A
ACALLRTC_WR
ACALLLCD_CLR      ;*****
AJMP LOOP

```

```

;@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@

```

```

SET_T_LOG:  ACALLLCD_CLR
MOV  LCD_ADDR,#000H
ACALLSET_ADDR_LCD
MOV  DPTR,#TLOG
ACALLWRLINE_LCD
ACALLDELAY_1S
ACALLDELAY_1S

MOV  LCD_ADDR,#000H
ACALLSET_ADDR_LCD
MOV  DPTR,#SLOG
ACALLWRLINE_LCD
MOV  LCD_ADDR,#040H
ACALLSET_ADDR_LCD

```

```

ACALLLCD_BLINK

```

```

ACALLWAIT_KEYPRESS
MOV  BUFFER,KPAD_DATA
MOV  A,KPAD_DATA
ADD  A,#030H
MOV  LCD_DATA,A
ACALLWRCHAR_LCD

```

```

ACALLLCD_BLINK

```

เอกสารนี้เป็นเอกสารทสงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ACALL WAIT_KEY

ACALL WAIT_KEYPRESS

MOV BUFFER+1,KPAD_DATA

MOV A,KPAD_DATA

ADD A,#030H

MOV LCD_DATA,A

ACALL WRCHAR_LCD

ACALL WAIT_KEY

MOV A,BUFFER

ANL A,#00FH

SWAP A

MOV B,A

MOV A,BUFFER+1

ANL A,#00FH

ADD A,B

MOV SH,A ;STORE SET HOUR

MOV LCD_ADDR,#042H

ACALL SET_ADDR_LCD

MOV LCD_DATA,#':'

ACALL WRCHAR_LCD ;SHOW :

MOV LCD_ADDR,#043H

ACALL SET_ADDR_LCD

ACALL LCD_BLINK

ACALL WAIT_KEYPRESS

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

MOV BUFFER,KPAD_DATA

MOV A,KPAD_DATA

ADD A,#030H

MOV LCD_DATA,A

ACALL WRCHAR_LCD

ACALL LCD_BLINK

ACALL WAIT_KEY

ACALL WAIT_KEYPRESS

MOV BUFFER+1,KPAD_DATA

MOV A,KPAD_DATA

ADD A,#030H

MOV LCD_DATA,A

ACALL WRCHAR_LCD

ACALL WAIT_KEY

MOV A,BUFFER

ANL A,#00FH

SWAP A

MOV B,A

MOV A,BUFFER+1

ANL A,#00FH

ADD A,B

MOV SM,A ;STORE SET MINUTE

MOV M,MINUTES ;STORE DATA START

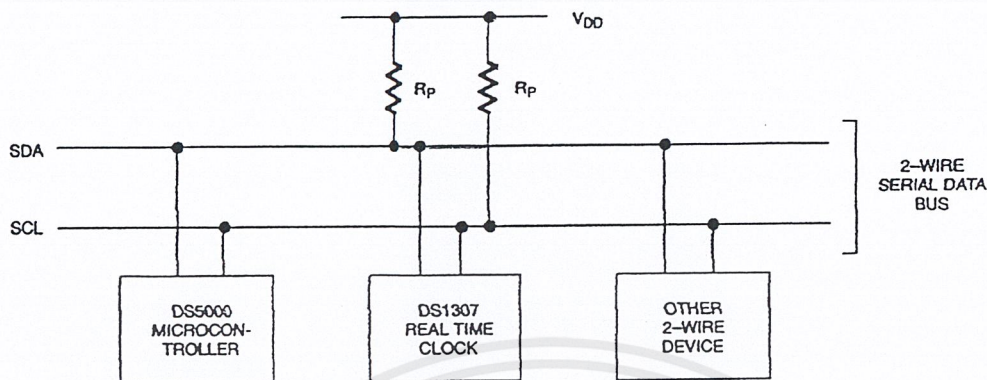
mov CT,#00H

mov CM,#00H

AJMP LOOP

;@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@

TYPICAL 2-WIRE BUS CONFIGURATION Figure 1



The following bus protocol has been defined (see Figure 2).

- 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:

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.

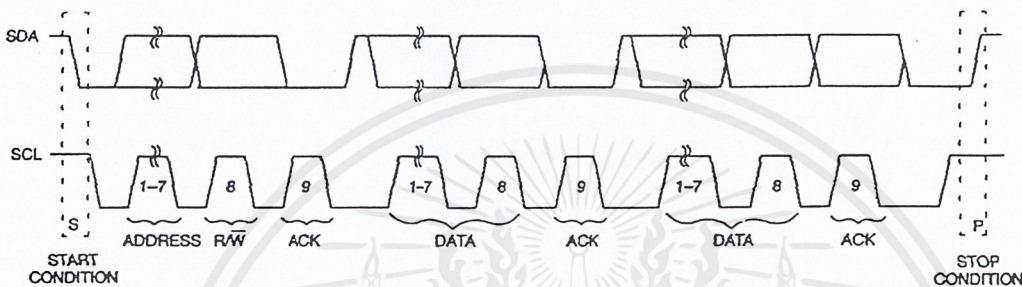
Acknowledge: Each receiving device, when addressed, is obliged to generate an acknowledge after the reception of each byte. The master device must generate an extra clock pulse which is associated with this acknowledge bit.

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. 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.

Figure 2 details how data transfer is accomplished on the 2-wire bus. Depending on the state of the R/ $\overline{W}$ bit, two types of data transfer are possible:

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 transferred 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.

DATA TRANSFER ON 2-WIRE SERIAL BUS Figure 2



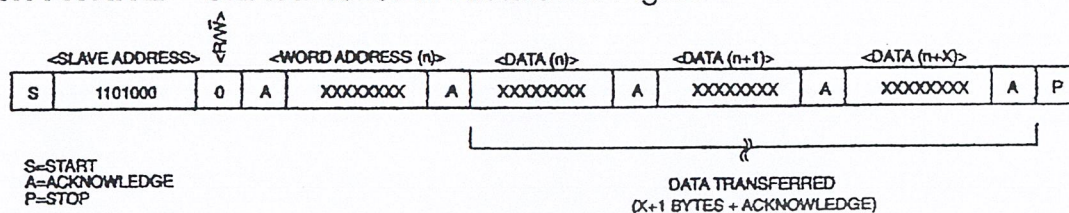
The master device generates all of the serial clock pulses and the START and STOP conditions. A transfer is ended with a STOP condition or with a repeated START condition. Since a repeated START condition is also the beginning of the next serial transfer, the bus will not be released. Data is transferred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

The DS1307 may operate in the following two modes:

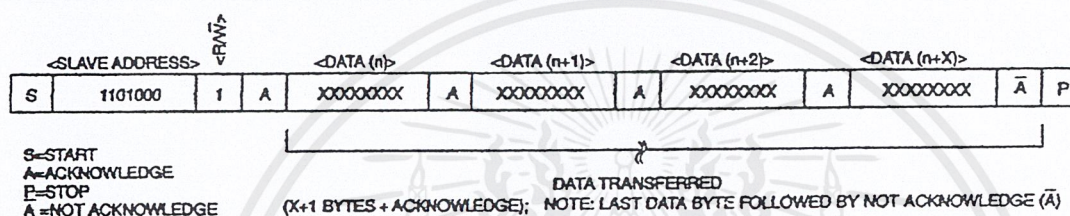
1. **Slave receiver mode (DS1307 write mode):** Serial data and clock are received through SDA and SCL. After each byte is received, an acknowledge bit is transmitted. START and STOP conditions are recognized as the beginning and end of a serial transfer. Address recognition is performed by hardware after reception of the slave address and direction bit (see Figure 3). 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 ($R/\bar{W}$) 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.
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 4). 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 ($R/\bar{W}$), 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 be sent a Not-Acknowledge bit by the master to terminate a read.

DATA WRITE – SLAVE RECEIVER MODE Figure 3



DATA READ – SLAVE TRANSMITTER MODE Figure 4



SOFTWARE OPERATION

DS5000 INTERFACE

The software presented in Appendix 1 is written to interface the DS5000 with the DS1307 over the 2-wire interface. The DS5000 was programmed using Dallas Semiconductor's DS5000T Evaluation Kit, which allows a PC to be used as a dumb terminal. The KIT5K software environment supplied with the DS5000T Evaluation Kit provides a high-level interface for loading application software to the DS5000 or for setting its configuration parameters via the Program command. The KIT5K software includes a dumb terminal emulator to allow users to run application software in the DS5000, which communicates with the user via a PC COM port.

DS1307 SOURCE CODE

The first section of the code found in the Appendix is used to configure the DS5000 for serial communication with the PC. Also at the beginning of the code is the MASTER_CONTROLLER subroutine which is used to control the demonstration software.

The subroutines that immediately follow the MASTER_CONTROLLER subroutine are the low level drivers for controlling the 2-wire interface. They are not specific to the DS1307 but can be used with any 2-wire compatible slave-only device. These subroutines are:

SEND_START

This subroutine is used to generate the Start condition on the 2-wire bus.

SEND_STOP

This subroutine is used to generate the Stop condition on the 2-wire bus.

SEND_BYTE

This subroutine sends an 8-bit word, MSB first, over the 2-wire bus with a 9th clock pulse for the Acknowledge pulse.

READ_BYTE

This subroutine reads an 8-bit word over the 2-wire bus. It checks for the LASTREAD flag to be cleared indicating when the last read from the slave device is to occur. If it is not the last read, the DS5000 sends an Acknowledge pulse on the 9th clock and if it is the last read from the slave device, the DS5000 sends a Not-Acknowledge.

SCL_HIGH

This subroutine transitions the SCL line low-to-high and ensures the SCL line is high before continuing.

DELAY and DELAY_4

These two subroutines have been included to ensure that the 2-wire bus timing is maintained.

The rest of the code included in the appendix is specifically designed to demonstrate the functions of the DS1307. The functions that are demonstrated are:

Setting Time

The time is read in from the keyboard and stored in the DS5000 scratchpad memory. It is then transferred, over the 2-wire interface, to the DS1307.

Set RAM

A single hex byte is read in from the keyboard and written to the entire user RAM of the DS1307.

Read Date/Time

The date and time are read, over the 2-wire bus, and stored in the DS5000 scratchpad memory. It is then written to the screen. This continues until a key is pressed on the keyboard.

Read RAM

The entire user RAM of the DS1307 is read into the DS5000 scratchpad memory and then written to the PC monitor.

OSC On/ OSC Off

The DS1307 clock oscillator can be turned on or off.

SQW/OUT On/ SQW/OUT Off

The SQW/OUT can be turned on or off. It will toggle at 1 Hz.



MICROCHIP

24AA256/24LC256/24FC256

256K Bit I²C™ CMOS Serial EEPROM

DEVICE SELECTION TABLE

Part Number	Vcc Range	Max Clock Frequency	Temp Ranges
24AA256	1.8-5.5V	400 kHz [†]	I
24LC256	2.5-5.5V	400 kHz [‡]	I, E
24FC256	2.5-5.5V	1 MHz	I

[†]100 kHz for Vcc < 2.5V.
[‡]100 kHz for E temperature range.

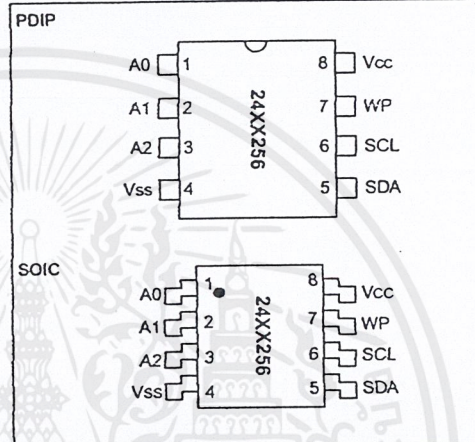
FEATURES

- Low power CMOS technology
 - Maximum write current 3 mA at 5.5V
 - Maximum read current 400 μ A at 5.5V
 - Standby current 100 nA typical at 5.5V
- 2-wire serial interface bus, I²C compatible
- Cascadable for up to eight devices
- Self-timed ERASE/WRITE cycle
- 64-byte page-write mode available
- 5 ms max write-cycle time
- Hardware write protect for entire array
- Output slope control to eliminate ground bounce
- Schmitt trigger inputs for noise suppression
- 100,000 erase/write cycles guaranteed
- Electrostatic discharge protection > 4000V
- Data retention > 200 years
- 8-pin PDIP and SOIC (208 mil) packages
- Temperature ranges:
 - Industrial (I): -40°C to +85°C
 - Automotive (E): -40°C to +125°C

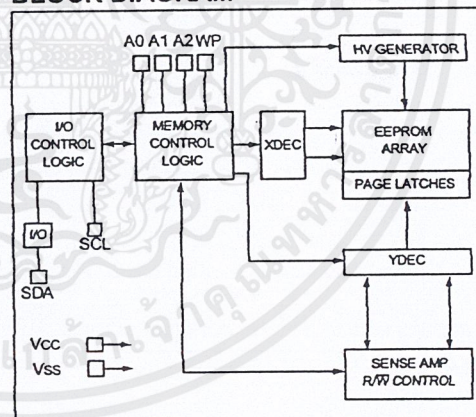
DESCRIPTION

The Microchip Technology Inc. 24AA256/24LC256/24FC256 (24XX256*) is a 32K x 8 (256K bit) Serial Electrically Erasable PROM, capable of operation across a broad voltage range (1.8V to 5.5V). It has been developed for advanced, low power applications such as personal communications or data acquisition. This device also has a page-write capability of up to 64 bytes of data. This device is capable of both random and sequential reads up to the 256K boundary. Functional address lines allow up to eight devices on the same bus, for up to 2M bit address space. This device is available in the standard 8-pin plastic DIP and 8-pin SOIC (208 mil) packages.

PACKAGE TYPE



BLOCK DIAGRAM



I²C is a trademark of Philips Corporation.

*24XX256 is used in this document as a generic part number for the 24AA256/24LC256/24FC256 devices.

24AA256/24LC256/24FC256

1.0 ELECTRICAL CHARACTERISTICS

1.1 Maximum Ratings*

VCC7.0V
All inputs and outputs w.r.t. VSS-0.6V to VCC +1.0V
Storage temperature-65°C to +150°C
Ambient temp. with power applied-65°C to +125°C
Soldering temperature of leads (10 seconds)+300°C
ESD protection on all pins≥ 4 kV

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

TABLE 1-1 PIN FUNCTION TABLE

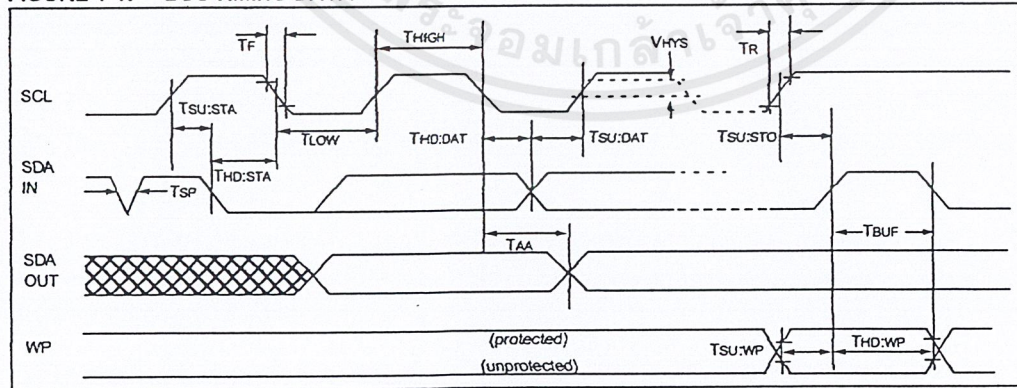
Name	Function
A0, A1, A2	User Configurable Chip Selects
VSS	Ground
SDA	Serial Data
SCL	Serial Clock
WP	Write Protect Input
VCC	+1.8 to 5.5V (24AA256) +2.5 to 5.5V (24LC256, 24FC256)

TABLE 1-2 DC CHARACTERISTICS

All parameters apply across the specified operating ranges, unless otherwise noted.					
Industrial (I): VCC = +1.8V to 5.5V Tamb = -40°C to +85°C Automotive (E): VCC = +4.5V to 5.5V Tamb = -40°C to 125°C					
Parameter	Symbol	Min	Max	Units	Conditions
A0, A1, A2, SCL, SDA, and WP pins: High level input voltage	V _{IH}	0.7 VCC	—	V	VCC 2.5V VCC < 2.5V VCC 2.5V (Note)
Low level input voltage	V _{IL}	—	0.3 VCC	V	
			0.2 VCC	V	
Hysteresis of Schmitt Trigger inputs (SDA, SCL pins)	V _{HYS}	0.05 VCC	—	V	VCC 2.5V (Note)
Low level output voltage	V _{OL}	—	0.40	V	I _{OL} = 3.0 mA @ VCC = 4.5V I _{OL} = 2.1 mA @ VCC = 2.5V
Input leakage current	I _{LI}	-10	10	μA	V _{IN} = VSS or VCC, WP = VSS V _{IN} = VSS or VCC, WP = VCC
Output leakage current	I _{LO}	-10	10	μA	V _{OUT} = VSS or VCC
Pin capacitance (all inputs/outputs)	C _{IN} , C _{OUT}	—	10	pF	VCC = 5.0V (Note) Tamb = 25°C, f _c = 1 MHz
Operating current	I _{CC} Read	—	400	μA	VCC = 5.5V, SCL = 400 kHz
	I _{CC} Write	—	3	mA	VCC = 5.5V
Standby current	I _{CCS}	—	1	μA	SCL = SDA = VCC = 5.5V A0, A1, A2, WP = VSS

Note: This parameter is periodically sampled and not 100% tested.

FIGURE 1-1: BUSTIMING DATA



24AA256/24LC256/24FC256

2.0 PIN DESCRIPTIONS

2.1 A0, A1, A2 Chip Address Inputs

The A0, A1, A2 inputs are used by the 24XX256 for multiple device operations. The levels on these inputs are compared with the corresponding bits in the slave address. The chip is selected if the compare is true.

Up to eight devices may be connected to the same bus by using different chip select bit combinations. If left unconnected, these inputs will be pulled down internally to VSS.

2.2 SDA Serial Data

This is a bi-directional pin used to transfer addresses and data into and data out of the device. It is an open-drain terminal, therefore, the SDA bus requires a pullup resistor to Vcc (typical 10 k Ω for 100 kHz, 2 k Ω for 400 kHz and 1 MHz).

For normal data transfer SDA is allowed to change only during SCL low. Changes during SCL high are reserved for indicating the START and STOP conditions.

2.3 SCL Serial Clock

This input is used to synchronize the data transfer from and to the device.

2.4 WP

This pin can be connected to either VSS, Vcc or left floating. An internal pull-down resistor on this pin will keep the device in the unprotected state if left floating. If tied to VSS or left floating, normal memory operation is enabled (read/write the entire memory 0000-7FFF).

If tied to Vcc, WRITE operations are inhibited. Read operations are not affected.

3.0 FUNCTIONAL DESCRIPTION

The 24XX256 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 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 while the 24XX256 works as a slave. Both master and slave can operate as a transmitter or receiver, but the master device determines which mode is activated.

4.0 BUS CHARACTERISTICS

The following bus protocol has been defined:

- 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 a START or STOP condition.

Accordingly, the following bus conditions have been defined (Figure 4-1).

4.1 Bus not Busy (A)

Both data and clock lines remain HIGH.

4.2 Start Data Transfer (B)

A HIGH to LOW transition of the SDA line while the clock (SCL) is HIGH determines a START condition. All commands must be preceded by a START condition.

4.3 Stop Data Transfer (C)

A LOW to HIGH transition of the SDA line while the clock (SCL) is HIGH determines a STOP condition. All operations must end with a STOP condition.

4.4 Data Valid (D)

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 bit of data per clock pulse.

Each data transfer is initiated with a START condition and terminated with a STOP condition. The number of the data bytes transferred between the START and STOP conditions is determined by the master device.

4.5 Acknowledge

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

Note: The 24XX256 does not generate any acknowledge bits if an internal programming cycle is in progress.

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. During reads, 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 (24XX256) will leave the data line HIGH to enable the master to generate the STOP condition.

24AA256/24LC256/24FC256

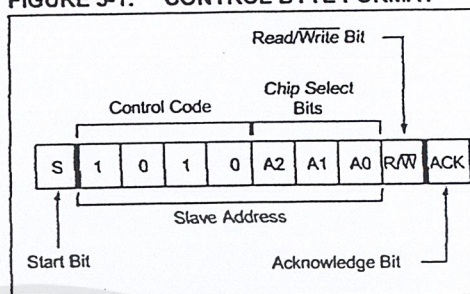
5.0 DEVICE ADDRESSING

A control byte is the first byte received following the start condition from the master device (Figure 5-1). The control byte consists of a 4-bit control code; for the 24XX256 this is set as 1010 binary for read and write operations. The next three bits of the control byte are the chip select bits (A2, A1, A0). The chip select bits allow the use of up to eight 24XX256 devices on the same bus and are used to select which device is accessed. The chip select bits in the control byte must correspond to the logic levels on the corresponding A2, A1, and A0 pins for the device to respond. These bits are in effect the three most significant bits of the word address.

The last bit of the control byte defines the operation to be performed. When set to a one a read operation is selected, and when set to a zero a write operation is selected. The next two bytes received define the address of the first data byte (Figure 5-2). Because only A14...A0 are used, the upper address bits is a don't care. The upper address bits are transferred first, followed by the less significant bits.

Following the start condition, the 24XX256 monitors the SDA bus checking the device type identifier being transmitted. Upon receiving a 1010 code and appropriate device select bits, the slave device outputs an acknowledge signal on the SDA line. Depending on the state of the R/W bit, the 24XX256 will select a read or write operation.

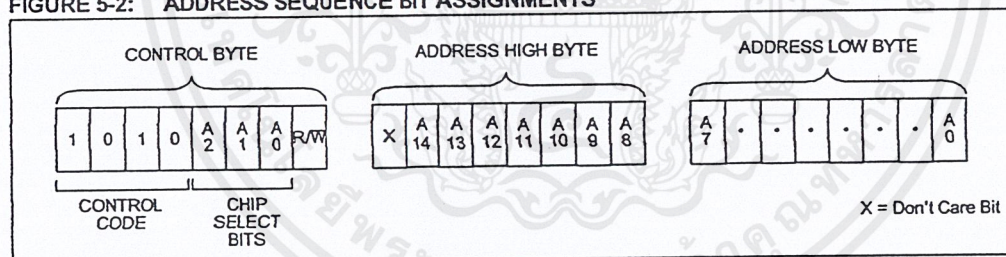
FIGURE 5-1: CONTROL BYTE FORMAT



5.1 Contiguous Addressing Across Multiple Devices

The chip select bits A2, A1, A0 can be used to expand the contiguous address space for up to 2 Mbit by adding up to eight 24XX256's on the same bus. In this case, software can use A0 of the control byte as address bit A15; A1, as address bit A16; and A2, as address bit A17. It is not possible to sequentially read across device boundaries.

FIGURE 5-2: ADDRESS SEQUENCE BIT ASSIGNMENTS



24AA256/24LC256/24FC256

6.0 WRITE OPERATIONS

6.1 Byte Write

Following the start condition from the master, the control code (four bits), the chip select (three bits), and the R/W bit (which is a logic low) are clocked onto the bus by the master transmitter. This indicates to the addressed slave receiver that the address high byte will follow after it has generated an acknowledge bit during the ninth clock cycle. Therefore, the next byte transmitted by the master is the high-order byte of the word address and will be written into the address pointer of the 24XX256. The next byte is the least significant address byte. After receiving another acknowledge signal from the 24XX256, the master device will transmit the data word to be written into the addressed memory location. The 24XX256 acknowledges again and the master generates a stop condition. This initiates the internal write cycle, and, during this time, the 24XX256 will not generate acknowledge signals (Figure 6-1). If an attempt is made to write to the array with the WP pin held high, the device will acknowledge the command but no write cycle will occur, no data will be written, and the device will immediately accept a new command. After a byte write command, the internal address counter will point to the address location following the one that was just written.

6.2 Page Write

The write control byte, word address, and the first data byte are transmitted to the 24XX256 in the same way as in a byte write. But instead of generating a stop condition, the master transmits up to 63 additional bytes, which are temporarily stored in the on-chip page buffer and will be written into memory after the master has transmitted a stop condition. After receipt of each word, the six lower address pointer bits are internally incremented by one. If the master should transmit more than 64 bytes prior to generating the stop condition, the address counter will roll over and the previously received data will be overwritten. As with the byte write operation, once the stop condition is received, an internal write cycle will begin (Figure 6-2). If an attempt is made to write to the array with the WP pin held high, the device will acknowledge the command but no write cycle will occur, no data will be written, and the device will immediately accept a new command.

6.3 Write Protection

The WP pin allows the user to write-protect the entire array (0000-7FFF) when the pin is tied to Vcc. If tied to Vss or left floating, the write protection is disabled. The WP pin is sampled at the STOP bit for every write command (Figure 1-1). Toggling the WP pin after the STOP bit will have no effect on the execution of the write cycle.

FIGURE 6-1: BYTE WRITE

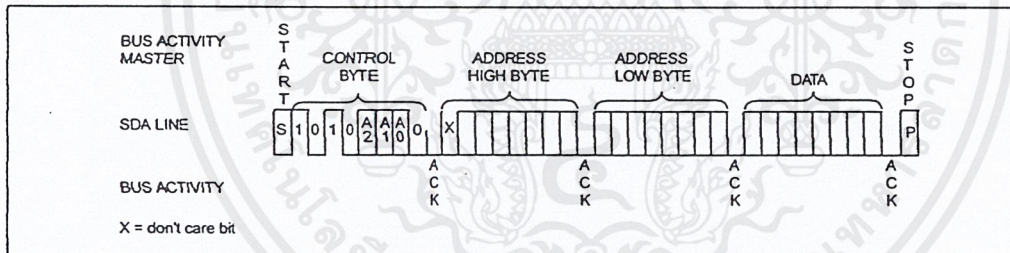
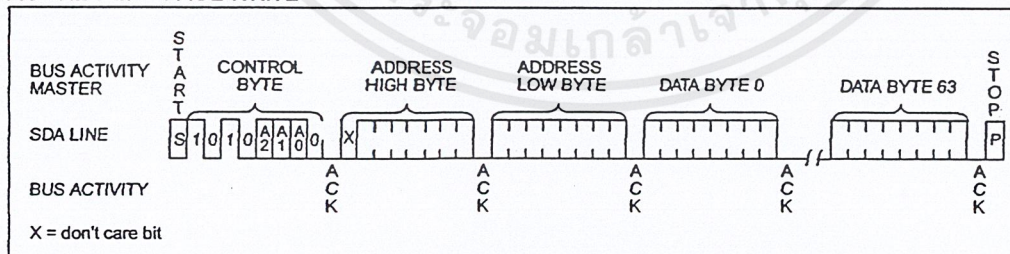


FIGURE 6-2: PAGE WRITE



24AA256/24LC256/24FC256

8.0 READ OPERATION

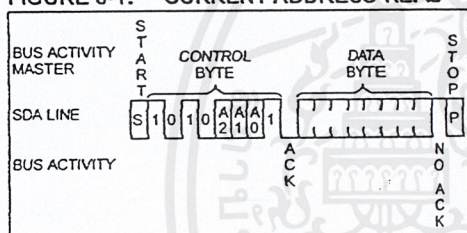
Read operations are initiated in the same way as write operations with the exception that the $R/\bar{W}$ bit of the control byte is set to one. There are three basic types of read operations: current address read, random read, and sequential read.

8.1 Current Address Read

The 24XX256 contains an address counter that maintains the address of the last word accessed, internally incremented by one. Therefore, if the previous read access was to address n (n is any legal address), the next current address read operation would access data from address $n + 1$.

Upon receipt of the control byte with $R/\bar{W}$ bit set to one, the 24XX256 issues an acknowledge and transmits the 8-bit data word. The master will not acknowledge the transfer but does generate a stop condition and the 24XX256 discontinues transmission (Figure 8-1).

FIGURE 8-1: CURRENT ADDRESS READ



8.2 Random Read

Random read operations allow the master to access any memory location in a random manner. To perform this type of read operation, first the word address must be set. This is done by sending the word address to the 24XX256 as part of a write operation ($R/\bar{W}$ bit set to 0). After the word address is sent, the master generates a start condition following the acknowledge. This terminates the write operation, but not before the internal address pointer is set. Then, the master issues the control byte again but with the $R/\bar{W}$ bit set to a one. The 24XX256 will then issue an acknowledge and transmit the 8-bit data word. The master will not acknowledge the transfer but does generate a stop condition which causes the 24XX256 to discontinue transmission (Figure 8-2). After a random read command, the internal address counter will point to the address location following the one that was just read.

8.3 Sequential Read

Sequential reads are initiated in the same way as a random read except that after the 24XX256 transmits the first data byte, the master issues an acknowledge as opposed to the stop condition used in a random read. This acknowledge directs the 24XX256 to transmit the next sequentially addressed 8-bit word (Figure 8-3). Following the final byte transmitted to the master, the master will NOT generate an acknowledge but will generate a stop condition. To provide sequential reads, the 24XX256 contains an internal address pointer which is incremented by one at the completion of each operation. This address pointer allows the entire memory contents to be serially read during one operation. The internal address pointer will automatically roll over from address 7FFF to address 0000 if the master acknowledges the byte received from the array address 7FFF.

FIGURE 8-2: RANDOM READ

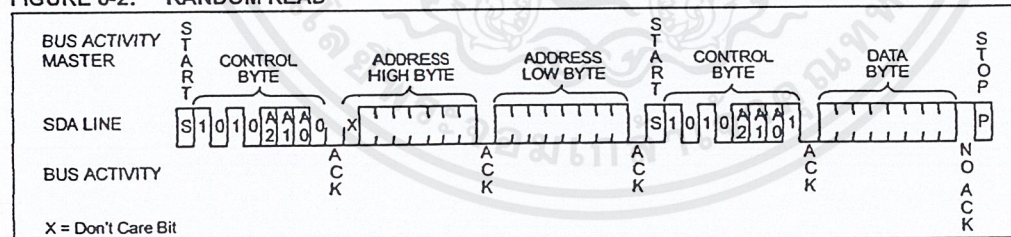
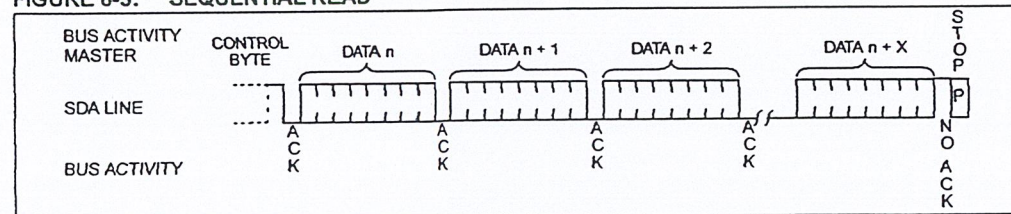


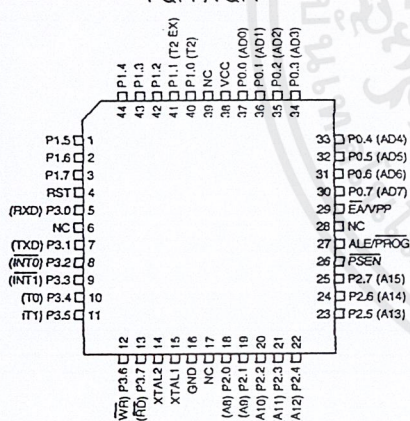
FIGURE 8-3: SEQUENTIAL READ



- Compatible with MCS-51™ Products
- 8K Bytes of In-System Reprogrammable Flash Memory
- Endurance: 1,000 Write/Erase Cycles
- 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
- Eight Interrupt Sources
- Programmable Serial Channel
- Low-power Idle and Power-down Modes

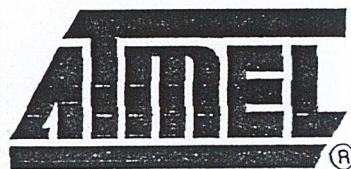
The AT89C52 is a low-power, high-performance CMOS 8-bit microcomputer with 8K bytes of Flash programmable and erasable read only memory (PEROM). The device is manufactured using Atmel's high-density nonvolatile memory technology and is compatible with the industry-standard 80C51 and 80C52 instruction set and pinout. The on-chip Flash allows the program memory to be reprogrammed in-system or by a conventional nonvolatile memory programmer. By combining a versatile 8-bit CPU with Flash on a monolithic chip, the Atmel AT89C52 is a powerful microcomputer which provides a highly-flexible and cost-effective solution to many embedded control applications.

POFR/TQFR



(T2) P1.0	1	40	VCC
(T2 EX) P1.1	2	39	P0.0 (AD0)
P1.2	3	38	P0.1 (AD1)
P1.3	4	37	P0.2 (AD2)
P1.4	5	36	P0.3 (AD3)
P1.5	6	35	P0.4 (AD4)
P1.6	7	34	P0.5 (AD5)
P1.7	8	33	P0.6 (AD6)
RST	9	32	P0.7 (AD7)
(RxD) P3.0	10	31	E _{AVPP}
(TxD) P3.1	11	30	ALE/PROG
(INT0) P3.2	12	29	PSEN
(INT1) P3.3	13	28	P2.7 (A15)
(T0) P3.4	14	27	P2.6 (A14)
(T1) P3.5	15	26	P2.5 (A13)
(WR) P3.6	16	25	P2.4 (A12)
(RD) P3.7	17	24	P2.3 (A11)
XTAL2	18	23	P2.2 (A10)
XTAL1	19	22	P2.1 (A9)
GND	20	21	P2.0 (A8)

Pin diagram of the PIC16C50 microcontroller. The chip is shown with pins numbered 1 to 40. Pin 1 is labeled (TXD) P3.0, Pin 2 is (RXD) P3.1, Pin 3 is P1.1 (T2 EX), Pin 4 is P1.2, Pin 5 is P1.3, Pin 6 is P1.4, Pin 7 is P1.5, Pin 8 is P1.6, Pin 9 is P1.7, Pin 10 is RST, Pin 11 is P3.0, Pin 12 is NC, Pin 13 is P3.1, Pin 14 is (INT) P3.2, Pin 15 is (INT) P3.3, Pin 16 is (T0) P3.4, Pin 17 is (T1) P3.5, Pin 18 is (WR) P3.6, Pin 19 is (RD) P3.7, Pin 20 is XTAL, Pin 21 is XTAL, Pin 22 is GND, Pin 23 is NC, Pin 24 is (A0) P2.0, Pin 25 is (A1) P2.1, Pin 26 is (A2) P2.2, Pin 27 is (A3) P2.3, Pin 28 is (A4) P2.4, Pin 29 is (A5) P2.5, Pin 30 is (A6) P2.6, Pin 31 is (A7) P2.7, Pin 32 is PSEN, Pin 33 is EAV/PROG, Pin 34 is NC, Pin 35 is EAV/PP, Pin 36 is P0.7 (AD7), Pin 37 is P0.6 (AD6), Pin 38 is P0.5 (AD5), Pin 39 is P0.4 (AD4), Pin 40 is P0.3 (AD3).



8-bit Microcontroller with 8K Bytes Flash

AT89C52



MC78XX/LM78XX/MC78XXA

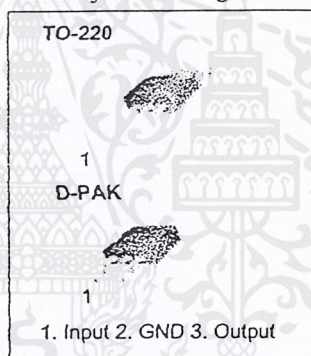
3-Terminal 1A Positive Voltage Regulator

Features

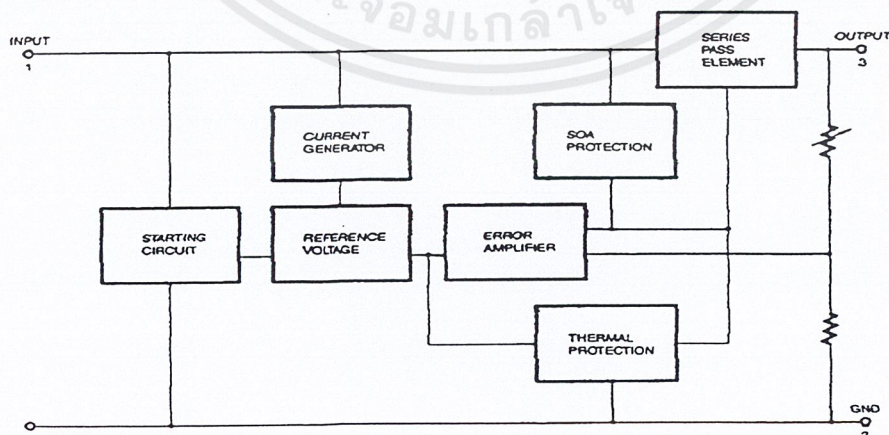
- Output Current up to 1A
- Output Voltages of 5, 6, 8, 9, 10, 12, 15, 18, 24V
- Thermal Overload Protection
- Short Circuit Protection
- Output Transistor Safe Operating Area Protection

Description

The MC78XX/LM78XX/MC78XXA series of three terminal positive regulators are available in the TO-220/D-PAK package and with several fixed output voltages, making them useful in a wide range of applications. Each type employs internal current limiting, thermal shut down and safe operating area protection, making it essentially indestructible. If adequate heat sinking is provided, they can deliver over 1A output current. Although designed primarily as fixed voltage regulators, these devices can be used with external components to obtain adjustable voltages and currents.



Internal Block Diagram



Rev. 1.0.0

Absolute Maximum Ratings

Parameter	Symbol	Value	Unit
Input Voltage (for $V_O = 5V$ to $18V$) (for $V_O = 24V$)	V_I	35	V
	V_I	40	V
Thermal Resistance Junction-Cases (TO-220)	$R_{\theta JC}$	5	$^{\circ}C/W$
Thermal Resistance Junction-Air (TO-220)	$R_{\theta JA}$	65	$^{\circ}C/W$
Operating Temperature Range	T_{OPR}	0 ~ +125	$^{\circ}C$
Storage Temperature Range	T_{STG}	-65 ~ +150	$^{\circ}C$

Electrical Characteristics (MC7805/LM7805)

(Refer to test circuit , $0^{\circ}C < T_J < 125^{\circ}C$, $I_O = 500mA$, $V_I = 10V$, $C_I = 0.33\mu F$, $C_O = 0.1\mu F$, unless otherwise specified)

Parameter	Symbol	Conditions		MC7805/LM7805			Unit
				Min.	Typ.	Max.	
Output Voltage	Vo	TJ=+25 °C		4.8	5.0	5.2	V
		5.0mA ≤ Io ≤ 1.0A, PO ≤ 15W VI = 7V to 20V		4.75	5.0	5.25	
Line Regulation (Note1)	Regline	TJ=+25 °C	VO = 7V to 25V	-	4.0	100	mV
			VI = 8V to 12V	-	1.6	50	
Load Regulation (Note1)	Regload	TJ=+25 °C	IO = 5.0mA to1.5A	-	9	100	mV
			IO =250mA to 750mA	-	4	50	
Quiescent Current	Iq	TJ =+25 °C		-	5.0	8.0	mA
Quiescent Current Change	ΔIq	Io = 5mA to 1.0A		-	0.03	0.5	mA
		VI= 7V to 25V		-	0.3	1.3	
Output Voltage Drift	ΔVo/ΔT	Io= 5mA		-	-0.8	-	mV/ °C
Output Noise Voltage	VN	f = 10Hz to 100KHz, TA=+25 °C		-	42	-	μV/Vo
Ripple Rejection	RR	f = 120Hz VO = 8V to 18V		62	73	-	dB
Dropout Voltage	VDrop	Io = 1A, TJ =+25 °C		-	2	-	V
Output Resistance	ro	f = 1KHz		-	15	-	mΩ
Short Circuit Current	IsC	VI = 35V, TA =+25 °C		-	230	-	mA
Peak Current	IPK	TJ =+25 °C		-	2.2	-	A

Note:

1. Load and line regulation are specified at constant junction temperature. Changes in V_O due to heating effects must be taken into account separately. Pulse testing with low duty is used.

+5V-Powered, Multichannel RS-232 Drivers/Receivers

MAX220-MAX249

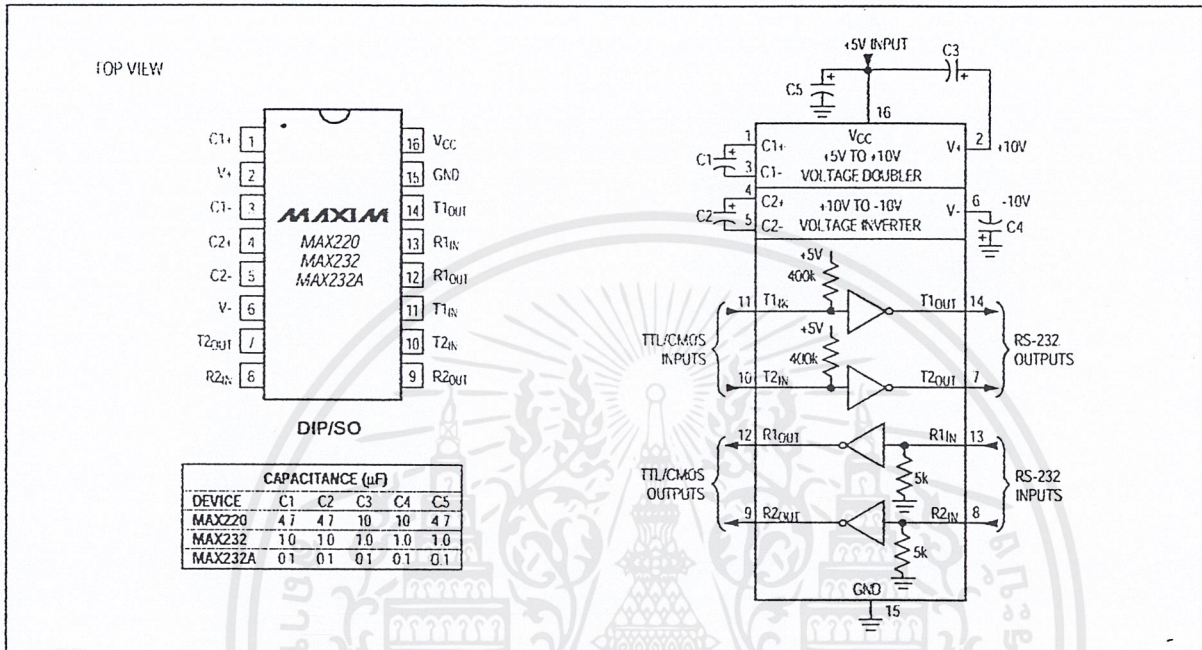


Figure 5. MAX220/MAX232/MAX232A Pin Configuration and Typical Operating Circuit

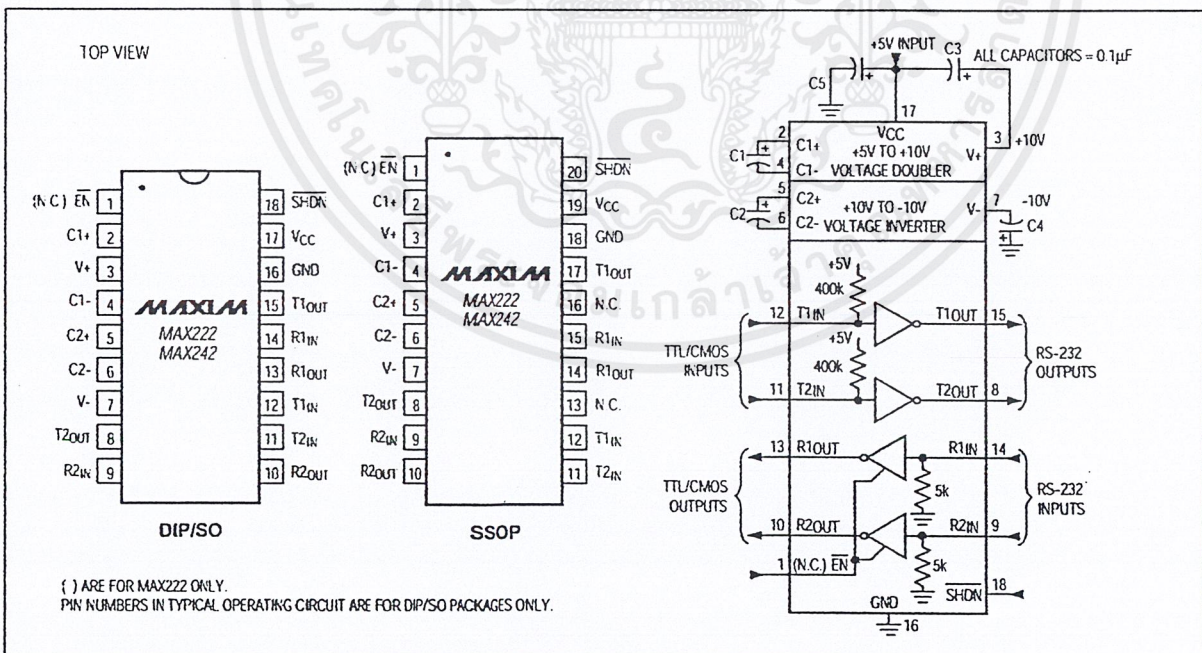


Figure 6. MAX222/MAX242 Pin Configurations and Typical Operating Circuit



TECHNICAL DATA

AN EXCLUSIVE RADIO SHACK SERVICE TO THE EXPERIMENTER

XR-2206 Monolithic Function Generator

Description

The XR-2206 is a monolithic function generator integrated circuit capable of producing high quality sine, square, triangle, ramp and pulse waveforms of high stability and accuracy. The output waveforms can be both amplitude and frequency modulated by an external voltage. Frequency of operation can be selected externally over a range of 0.01 Hz to more than 1 MHz.

The XR-2206 is ideally suited for communications, instrumentation, and function generator applications requiring sinusoidal tone, AM, FM or FSK generation. It has a typical drift specification of 20 ppm/°C. The oscillator frequency can be linearly swept over a 2000:1 frequency range with an external control voltage with very little effect on distortion.

As shown in Figure 1, the monolithic circuit is comprised of four functional blocks: a voltage-controlled oscillator (VCO); an analog multiplier and sine-shaper; a unity gain buffer amplifier; and a set of current switches. The internal current switches transfer the oscillator current to any one of the two external timing resistors to produce two discrete frequencies selected by the logic level at the FSK input terminal (pin 9).

Features

- Low Sinewave Distortion (THD .5%), insensitive to signal sweep
- Excellent Stability (20 ppm/°C, typ)
- Wide Sweep Range (2000:1, typ)
- Low Supply Sensitivity (0.01%/V, typ)
- Linear Amplitude Modulation
- Adjustable Duty-Cycle (1% to 99%)
- TTL Compatible FSK Controls
- Wide Supply Range (10V to 26V)

Applications

- Waveform Generation
Sine, Square, Triangle, Ramp
- Sweep Generation
- AM/FM Generation
- FSK and PSK Generation
- Voltage-to-Frequency Conversion
- Tone Generation
- Phase-Locked Loops

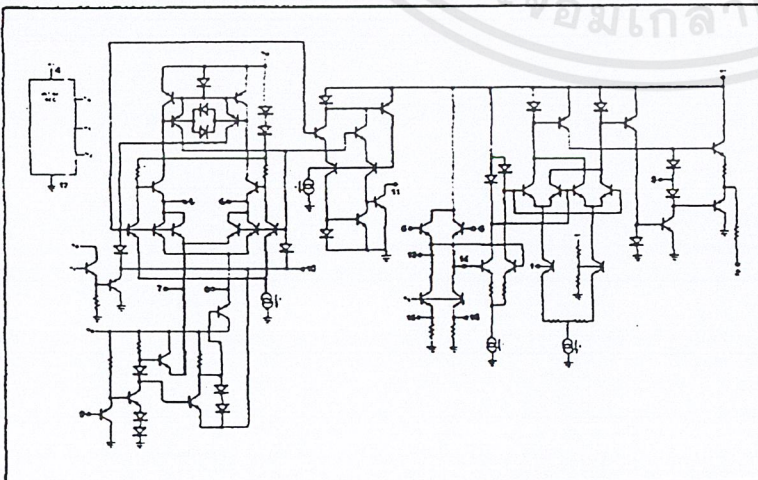
Absolute Maximum Ratings

Power Supply	26V
Power Dissipation	625mW
Derate above +25°C	5 mW/°C
Storage Temperature Range	-65°C to +150°C
Operating Temperature Range	0°C to +75°C

EQUIVALENT SCHEMATIC DIAGRAM

FUNCTIONAL BLOCK DIAGRAM

EQUIVALENT SCHEMATIC DIAGRAM



FUNCTIONAL BLOCK DIAGRAM

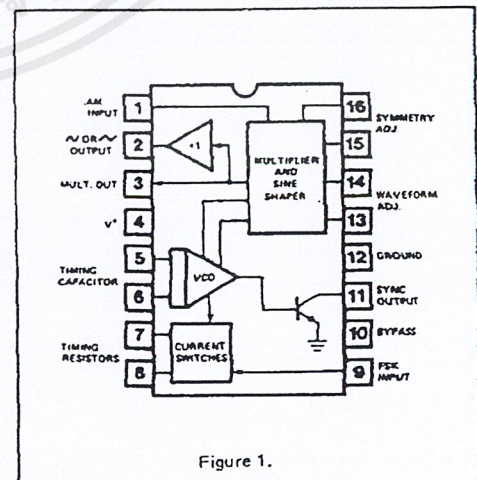


Figure 1.

CUSTOM PACKAGED IN USA BY RADIO SHACK, A DIVISION OF TANDY CORPORATION

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

The circuit of Figure 8 can be converted to split supply operation simply by replacing all ground connections with V^- . For split supply operation, R_3 can be directly connected to ground.

B) With External Adjustment

The harmonic content of sinusoidal output can be reduced to $\approx 0.5\%$ by additional adjustments as shown in Figure 9. The potentiometer R_A adjusts the sine-shaping resistor; and R_B provides the fine-adjustment for the waveform symmetry. The adjustment procedure is as follows:

1. Set R_B at mid-point and adjust R_A for minimum distortion.
2. With R_A set as above, adjust R_B to further reduce distortion.

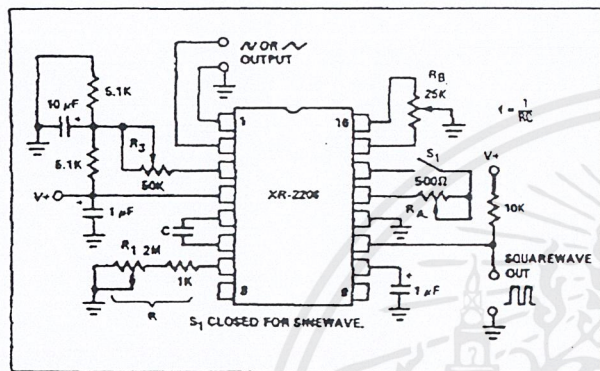


Figure 9: Circuit for Sinewave Generation With Minimum Harmonic Distortion. (R_3 Determines output Swing — See Fig. 3)

Triangle Wave Generation

The circuits of Figures 8 and 9 can be converted to triangle wave generation by simply open circuiting pins 13 and 14 (i.e., S_1 open). Amplitude of the triangle is approximately twice the sinewave output.

FSK Generation

Figure 10 shows the circuit connection for sinusoidal FSK signal generation. Mark and space frequencies can be independently adjusted by the choice of timing resistors R_1 and R_2 ; and the output is phase-continuous during transitions. The keying signal is applied to pin 9. The circuit can be converted to split-supply operation by simply replacing ground with V^- .

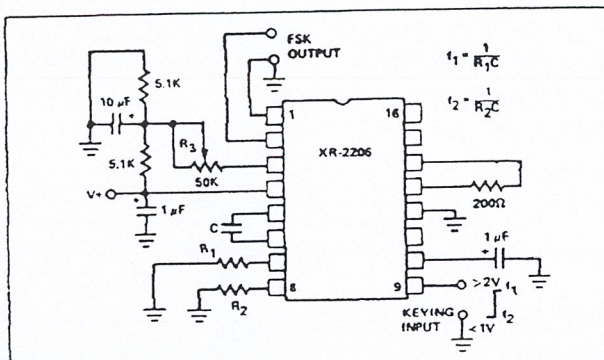


Figure 10: Sinusoidal FSK Generator

Pulse And Ramp Generation

Figure 11 shows the circuit for pulse and ramp waveform generation. In this mode of operation, the FSK keying terminal (pin 9) is shorted to the square-wave output (pin 11); and the circuit automatically frequency-shift keys itself between two separate frequencies during the positive and negative going output waveforms. The pulse-width and the duty cycle can be adjusted from 1% to 99% by the choice of R_1 and R_2 . The values of R_1 and R_2 should be in the range of $1\text{ K}\Omega$ to $2\text{ M}\Omega$.

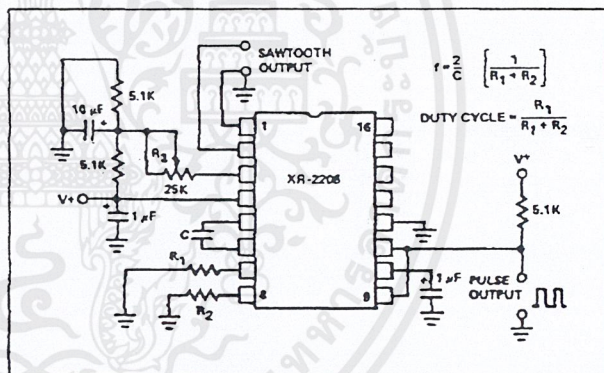


Figure 11: Circuit for Pulse and Ramp Generation

RADIO SHACK, A DIVISION OF TANDY CORPORATION

U.S.A.: FORT WORTH, TEXAS 76102
CANADA: BARRIE, ONTARIO L4M 4W5

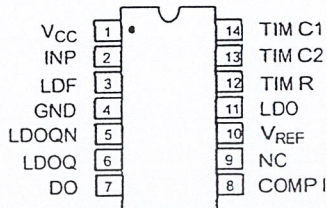
TANDY CORPORATION

AUSTRALIA
 91 KURRAJONG AVENUE
 MOUNT DRUITT, N.S.W. 2770

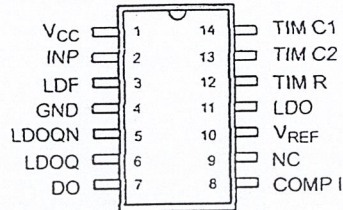
BELGIUM
 PARC INDUSTRIEL DE NANINNE
 5140 NANINNE

U. K.
 BILSTON ROAD WEDNESBURY
 WEST MIDLANDS WS10 7JN

PIN CONFIGURATION



14 Lead CDIP, PDIP (0.300")



14 Lead SOIC (Jedec, 0.150")

PIN DESCRIPTION

Pin #	Symbol	Type	Description
1	VCC		Positive Power Supply.
2	INP	I	Receive Analog Input.
3	LDF	O	Lock Detect Filter.
4	GND		Ground Pin.
5	LDOQN	O	Lock Detect Output Not. This output will be low if the VCO is in the capture range.
6	LDOQ	O	Lock Detect Output. This output will be high if the VCO is in the capture range.
7	DO	O	Data Output. Decoded FSK output.
8	COMP I	I	FSK Comparator Input.
9	NC		Not Connected.
10	VREF	O	Internal Voltage Reference. The value of VREF is $V_{CC}/2 - 650\text{mV}$.
11	LDO	O	Loop Detect Output. This output provides the result of the quadrature phase detection.
12	TIM R	I	Timing Resistor Input. This pin connects to the timing resistor of the VCO.
13	TIM C2	I	Timing Capacitor Input. The timing capacitor connects between this pin and pin 14.
14	TIM C1	I	Timing Capacitor Input. The timing capacitor connects between this pin and pin 13.

ELECTRICAL CHARACTERISTICS

Test Conditions: $V_{CC} = 12V$, $T_A = +25^\circ C$, $R_0 = 30K\Omega$, $C_0 = 0.033\mu F$, unless otherwise specified.

Parameter	Min.	Typ.	Max.	Unit	Conditions
General					
Supply Voltage	4.5		20	V	
Supply Current		4	7	mA	$R_0 \geq 10K\Omega$. See Figure 4.
Oscillator Section					
Frequency Accuracy		± 1	± 3	%	Deviation from $f_0 = 1/R_0 C_0$
Frequency Stability					
Temperature		± 20	± 50	ppm/ $^\circ C$	See Figure 8.
Power Supply		0.05	0.5	%/V	$V_{CC} = 12 \pm 1V$. See Figure 7.
		0.2		%/V	$V_{CC} = \pm 5V$. See Figure 7.
Upper Frequency Limit	100	300		kHz	$R_0 = 8.2K\Omega$, $C_0 = 400pF$
Lowest Practical Operating Frequency			0.01	Hz	$R_0 = 2M\Omega$, $C_0 = 50\mu F$
Timing Resistor, R_0 - See Figure 5					
Operating Range	5		2000	$K\Omega$	
Recommended Range	5			$K\Omega$	See Figure 7 and Figure 8.
Loop Phase Detector Section					
Peak Output Current	± 150	± 200	± 300	μA	Measured at Pin 11
Output Offset Current		1		μA	
Output Impedance		1		$M\Omega$	
Maximum Swing	± 4	± 5		V	Referenced to Pin 10
Quadrature Phase Detector					
					Measured at Pin 3
Peak Output Current	100	300		μA	
Output Impedance		1		$M\Omega$	
Maximum Swing		11		V_{PP}	
Input Preamplifier Section					
					Measured at Pin 2
Input Impedance		20		$K\Omega$	
Input Signal					
Voltage Required to Cause Limiting		2	10	mV rms	

Notes

Parameters are guaranteed over the recommended operating conditions, but are not 100% tested in production.

Bold face parameters are covered by production test and guaranteed over operating temperature range.

DC ELECTRICAL CHARACTERISTICS (CONT'D)

Test Conditions: $V_{CC} = 12V$, $T_A = +25^\circ C$, $R_O = 30K\Omega$, $C_O = 0.033\mu F$, unless otherwise specified.

Parameter	Min.	Typ.	Max.	Unit	Conditions
Voltage Comparator Section					
Input Impedance		2		M Ω	Measured at Pins 3 and 8
Input Bias Current		100		nA	
Voltage Gain	55	70		dB	$R_L = 5.1K\Omega$
Output Voltage Low		300	500	mV	$I_C = 3mA$
Output Leakage Current		0.01	10	μA	$V_O = 20V$
Internal Reference					
Voltage Level	4.9	5.3	5.7	V	Measured at Pin 10
Output Impedance		100		Ω	AC Small Signal
Maximum Source Current		80		μA	

Notes

Parameters are guaranteed over the recommended operating conditions, but are not 100% tested in production. Bold face parameters are covered by production test and guaranteed over operating temperature range.

Specifications are subject to change without notice

ABSOLUTE MAXIMUM RATINGS

Power Supply 20V
 Input Signal Level 3V rms
 Power Dissipation 900mW

Package Power Dissipation Ratings

CDIP 750mW
 Derate Above $T_A = 25^\circ C$ 8mW/ $^\circ C$
 PDIP 800mW
 Derate Above $T_A = 25^\circ C$ 60mW/ $^\circ C$
 SOIC 390mW
 Derate Above $T_A = 25^\circ C$ 5mW/ $^\circ C$

SYSTEM DESCRIPTION

The main PLL within the XR-2211 is constructed from an input preamplifier, analog multiplier used as a phase detector and a precision voltage controlled oscillator (VCO). The preamplifier is used as a limiter such that input signals above typically 10mV rms are amplified to a constant high level signal. The multiplying-type phase detector acts as a digital exclusive or gate. Its output (unfiltered) produces sum and difference frequencies of the input and the VCO output. The VCO is actually a current controlled oscillator with its normal input current (f_O) set by a resistor (R_O) to ground and its driving current with a resistor (R_1) from the phase detector.

The output of the phase detector produces sum and difference of the input and the VCO frequencies

(internally connected). When in lock, these frequencies are $f_{IN} + f_{VCO}$ (2 times f_{IN} when in lock) and $f_{IN} - f_{VCO}$ (0Hz when lock). By adding a capacitor to the phase detector output, the 2 times f_{IN} component is reduced, leaving a DC voltage that represents the phase difference between the two frequencies. This closes the loop and allows the VCO to track the input frequency.

The FSK comparator is used to determine if the VCO is driven above or below the center frequency (FSK comparator). This will produce both active high and active low outputs to indicate when the main PLL is in lock (quadrature phase detector and lock detector comparator).

DESIGN EQUATIONS

(All resistance in Ω , all frequency in Hz and all capacitance in farads, unless otherwise specified)

(See *Figure 3* for definition of components)

1. VCO Center Frequency, f_0 :

$$f_0 = \frac{1}{R_0 \cdot C_0}$$

2. Internal Reference Voltage, V_{REF} (measured at pin 10):

$$V_{REF} = \left(\frac{V_{CC}}{2} \right) - 650mV \text{ in volts}$$

3. Loop Low-Pass Filter Time Constant, τ :

$$\tau = C_1 \cdot R_{PP} \text{ (seconds)}$$

where:

$$R_{PP} = \left(\frac{R_1 \cdot R_F}{R_1 + R_F} \right)$$

if R_F is ∞ or C_F reactance is ∞ , then $R_{PP} = R_1$

4. Loop Damping, ζ :

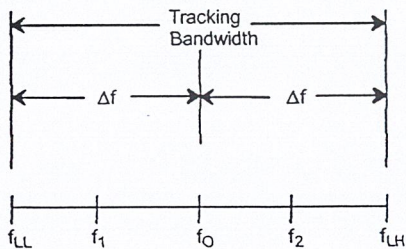
$$\zeta = \sqrt{\left(\frac{1250 \cdot C_0}{R_1 \cdot C_1} \right)}$$

Note: For derivation/explanation of this equation, please see TAN-011.

5. Loop-tracking

bandwidth, $\pm = \frac{\Delta f}{f_0}$

$$\frac{\Delta f}{f_0} = \frac{R_0}{R_1}$$



6. FSK Data filter time constant, t_F :

$$\tau_F = \frac{R_B \cdot R_F}{(R_B + R_F)} \cdot C_F \text{ (seconds)}$$

7. Loop phase detector conversion gain, K_d : (K_d is the differential DC voltage across pin 10 and pin11, per unit of phase error at phase detector input):

$$K_d = \frac{V_{REF} \cdot R_1}{10,000 \cdot \pi} \left[\frac{\text{volt}}{\text{radian}} \right]$$

Note: For derivation/explanation of this equation, please see TAN-011.

8. VCO conversion gain, K_o : (K_o is the amount of change in VCO frequency, per unit of DC voltage change at pin 11):

$$K_o = \frac{-2\pi}{V_{REF} \cdot C_o \cdot R_1} = \left(\frac{\text{radian/second}}{\text{volt}} \right)$$

9. The filter transfer function:

$$F(s) = \frac{1}{1 + SR_1 \cdot C_1} \text{ at } 0 \text{ Hz.} \quad S = j\omega \text{ and } \omega = 0$$

10. Total loop gain, K_T :

$$K_T = K_o \cdot K_d \cdot F(s) = \left(\frac{R_F}{5,000 \cdot C_o \cdot (R_1 + R_F)} \right) \left[\frac{1}{\text{seconds}} \right]$$

11. Peak detector current I_A :

$$I_A = \frac{V_{REF}}{20,000} \text{ (} V_{REF} \text{ in volts and } I_A \text{ in amps)}$$

Note: For derivation/explanation of this equation, please see TAN-011.

APPLICATIONS INFORMATION

FSK Decoding

Figure 10 shows the basic circuit connection for FSK decoding. With reference to Figure 3 and Figure 10, the functions of external components are defined as follows: R_0 and C_0 set the PLL center frequency, R_1 sets the system bandwidth, and C_1 sets the loop filter time constant and the loop damping factor. C_F and R_F form a one-pole post-detection filter for the FSK data output. The resistor R_B from pin 7 to pin 8 introduces positive feedback across the FSK comparator to facilitate rapid transition between output logic states.

Design Instructions:

The circuit of Figure 10 can be tailored for any FSK decoding application by the choice of five key circuit components: R_0 , R_1 , C_0 , C_1 and C_F . For a given set of FSK mark and space frequencies, f_0 and f_1 , these parameters can be calculated as follows:

(All resistance in Ω 's, all frequency in Hz and all capacitance in farads, unless otherwise specified)

- a) Calculate PLL center frequency, f_0 :

$$f_0 = \sqrt{F_1 \cdot F_2}$$

- b) Choose value of timing resistor R_0 , to be in the range of 10K Ω to 100K Ω . This choice is arbitrary. The recommended value is $R_0 = 20K\Omega$. The final value of R_0 is normally fine-tuned with the series potentiometer, R_X .

$$R_o = R_0 + \frac{R_X}{2}$$

- c) Calculate value of C_0 from design equation (1) or from Figure 7:

$$C_o = \frac{1}{R_o \cdot f_0}$$

- d) Calculate R_1 to give the desired tracking bandwidth (See design equation 5).

$$R_1 = \frac{R_o \cdot f_0}{(f_1 - f_2)} \cdot 2$$

- e) Calculate C_1 to set loop damping. (See design equation 4):

Normally, $\zeta = 0.5$ is recommended.

$$C_1 = \frac{1250 \cdot C_0}{R_1 \cdot \zeta^2}$$

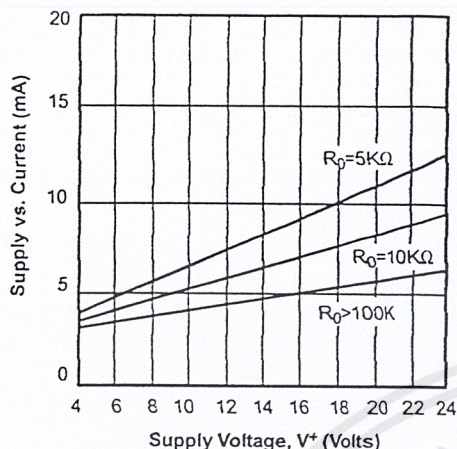


Figure 5. Typical Supply Current vs. V_+
(Logic Outputs Open Circuited)

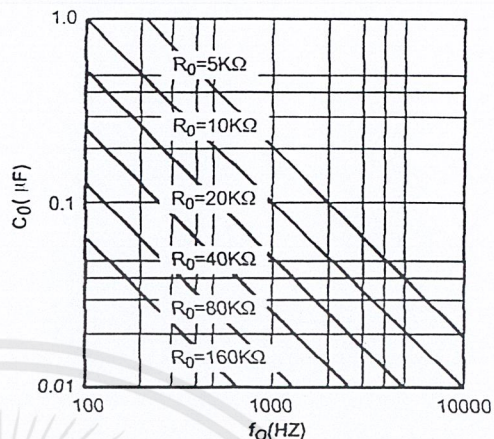


Figure 6. VCO Frequency vs. Timing Resistor

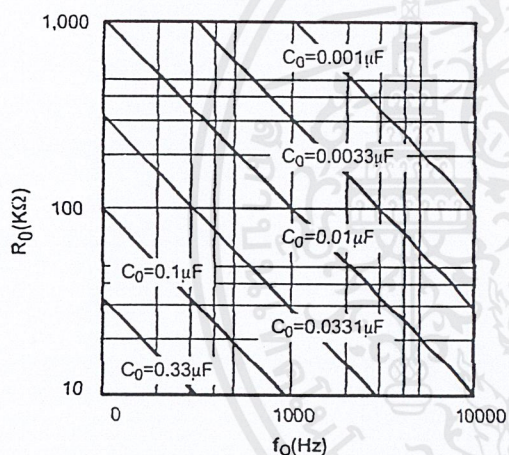


Figure 7. VCO Frequency vs. Timing Capacitor

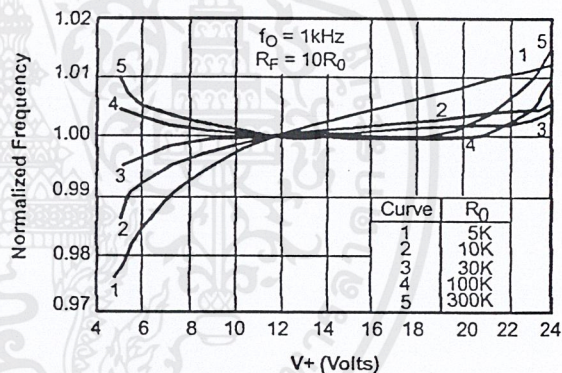


Figure 8. Typical f_0 vs. Power Supply
Characteristics

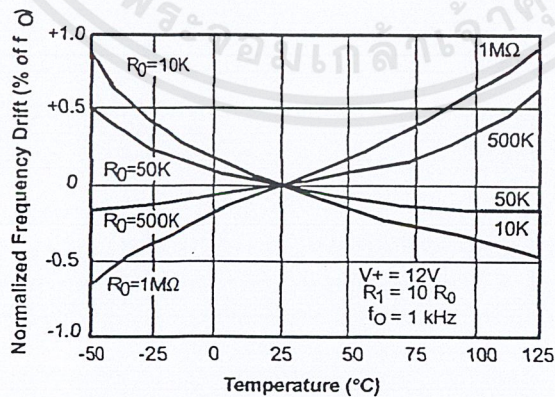


Figure 9. Typical Center Frequency Drift vs. Temperature

DALLAS
 SEMICONDUCTOR

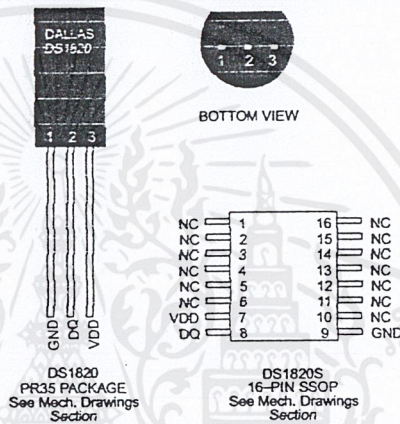
DS1820

1-Wire™ Digital Thermometer

FEATURES

- Unique 1-Wire™ interface requires only one port pin for communication
- Multidrop capability simplifies distributed temperature sensing applications
- Requires no external components
- Can be powered from data line
- Zero standby power required
- Measures temperatures from -55°C to $+125^{\circ}\text{C}$ in 0.5°C increments. Fahrenheit equivalent is -67°F to $+257^{\circ}\text{F}$ in 0.9°F increments
- Temperature is read as a 9-bit digital value.
- Converts temperature to digital word in 200 ms (typ.)
- User-definable, nonvolatile temperature alarm settings
- Alarm search command identifies and addresses devices whose temperature is outside of programmed limits (temperature alarm condition)
- Applications include thermostatic controls, industrial systems, consumer products, thermometers, or any thermally sensitive system

PIN ASSIGNMENT



PIN DESCRIPTION

GND	—	Ground
DQ	—	Data In/Out
VDD	—	Optional V _{DD}
NC	—	No Connect

DESCRIPTION

The DS1820 Digital Thermometer provides 9-bit temperature readings which indicate the temperature of the device.

Information is sent to/from the DS1820 over a 1-Wire interface, so that only one wire (and ground) needs to be connected from a central microprocessor to a DS1820. Power for reading, writing, and performing temperature conversions can be derived from the data line itself with no need for an external power source.

Because each DS1820 contains a unique silicon serial number, multiple DS1820s can exist on the same 1-Wire bus. This allows for placing temperature sensors in many different places. Applications where this feature is useful include HVAC environmental controls, sensing temperatures inside buildings, equipment or machinery, and in process monitoring and control.

DETAILED PIN DESCRIPTION

PIN 16-PIN SSOP	PIN PR35	SYMBOL	DESCRIPTION
9	1	GND	Ground.
8	2	DQ	Data Input/Output pin. For 1-Wire operation: Open drain. (See "Parasite Power" section.)
7	3	V _{DD}	Optional V _{DD} pin. See "Parasite Power" section for details of connection.

DS1820S (16-pin SSOP): All pins not specified in this table are not to be connected.

OVERVIEW

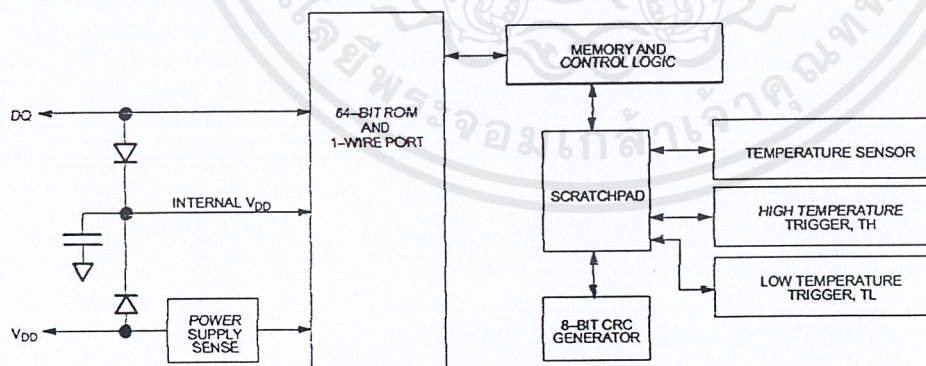
The block diagram of Figure 1 shows the major components of the DS1820. The DS1820 has three main data components: 1) 64-bit lasered ROM, 2) temperature sensor, and 3) nonvolatile temperature alarm triggers TH and TL. The device derives its power from the 1-Wire communication line by storing energy on an internal capacitor during periods of time when the signal line is high and continues to operate off this power source during the low times of the 1-Wire line until it returns high to replenish the parasite (capacitor) supply. As an alternative, the DS1820 may also be powered from an external 5 volts supply.

Communication to the DS1820 is via a 1-Wire port. With the 1-Wire port, the memory and control functions will not be available before the ROM function protocol has been established. The master must first provide one of five ROM function commands: 1) Read ROM, 2) Match ROM, 3) Search ROM, 4) Skip ROM, or 5) Alarm Search. These commands operate on the 64-bit lasered ROM portion of each device and can single out

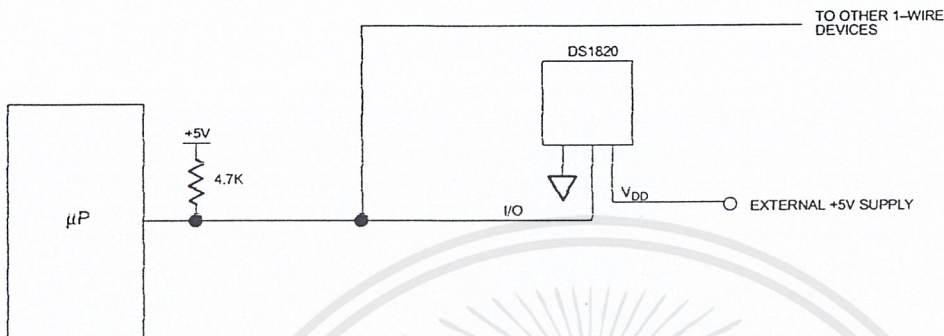
a specific device if many are present on the 1-Wire line as well as indicate to the Bus Master how many and what types of devices are present. After a ROM function sequence has been successfully executed, the memory and control functions are accessible and the master may then provide any one of the six memory and control function commands.

One control function command instructs the DS1820 to perform a temperature measurement. The result of this measurement will be placed in the DS1820's scratchpad memory, and may be read by issuing a memory function command which reads the contents of the scratchpad memory. The temperature alarm triggers TH and TL consist of one byte EEPROM each. If the alarm search command is not applied to the DS1820, these registers may be used as general purpose user memory. Writing TH and TL is done using a memory function command. Read access to these registers is through the scratchpad. All data is read and written least significant bit first.

DS1820 BLOCK DIAGRAM Figure 1



USING V_{DD} TO SUPPLY TEMPERATURE CONVERSION CURRENT Figure 3



OPERATION – MEASURING TEMPERATURE

The DS1820 measures temperature through the use of an on-board proprietary temperature measurement technique. A block diagram of the temperature measurement circuitry is shown in Figure 4.

The DS1820 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 zero 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 zero. If the gate period is still not finished, then this process repeats.

The slope accumulator is used to compensate for the non-linear 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.

Internally, this calculation is done inside the DS1820 to provide 0.5°C resolution. The temperature reading is

provided in a 16-bit, sign-extended two's complement reading. Table 1 describes the exact relationship of output data to measured temperature. The data is transmitted serially over the 1-Wire interface. The DS1820 can measure temperature over the range of -55°C to $+125^{\circ}\text{C}$ in 0.5°C increments. For Fahrenheit usage, a lookup table or conversion factor must be used.

Note that temperature is represented in the DS1820 in terms of a $1/2^{\circ}\text{C}$ LSB, yielding the following 9-bit format:

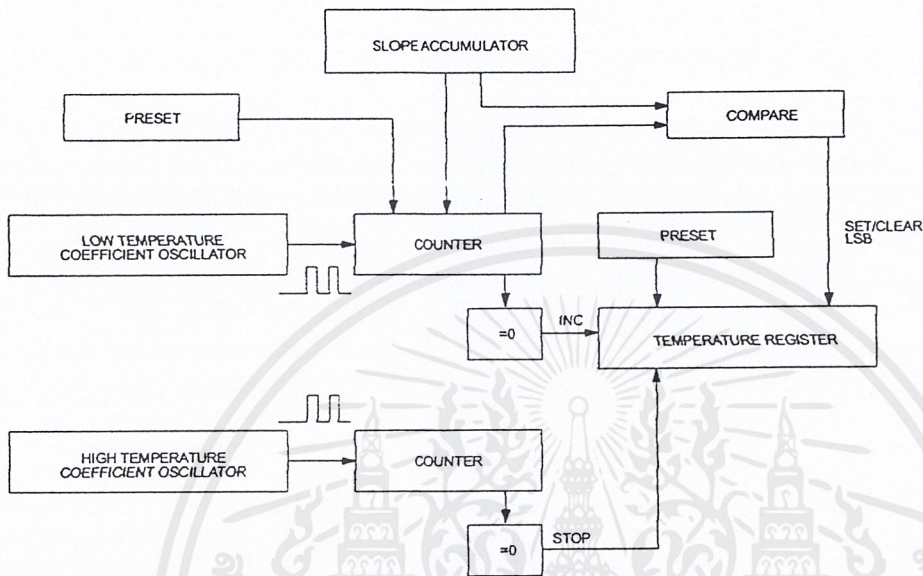
MSB								LSB
1	1	1	0	0	1	1	1	0
= -25°C								

The most significant (sign) bit is duplicated into all of the bits in the upper MSB of the two-byte temperature register in memory. This "sign-extension" yields the 16-bit temperature readings as shown in Table 1.

Higher resolutions may be obtained by the following procedure. First, read the temperature, and truncate 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. This value is the count remaining (COUNT_REMAIN) after the gate period has ceased. The last value needed is 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:

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

TEMPERATURE MEASURING CIRCUITRY Figure 4



TEMPERATURE/DATA RELATIONSHIPS Table 1

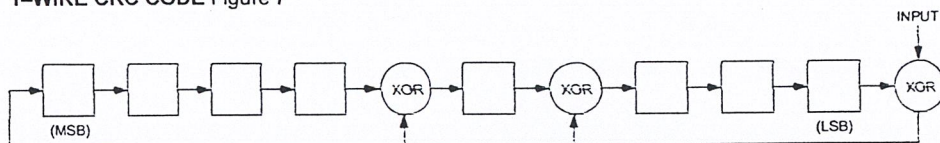
TEMPERATURE	DIGITAL OUTPUT (Binary)	DIGITAL OUTPUT (Hex)
+125°C	00000000 11111010	00FA
+25°C	00000000 00110010	0032h
+1/2°C	00000000 00000001	0001h
+0°C	00000000 00000000	0000h
-1/2°C	11111111 11111111	FFFFh
-25°C	11111111 11001110	FFCEh
-55°C	11111111 10010010	FF92h

OPERATION – ALARM SIGNALING

After the DS1820 has performed a temperature conversion, the temperature value is compared to the trigger values stored in TH and TL. Since these registers are 8-bit only, the 0.5°C bit is ignored for comparison. The most significant bit of TH or TL directly corresponds to the sign bit of the 16-bit temperature register. If the result of a temperature measurement is higher than TH or lower than TL, an alarm flag inside the device is set.

This flag is updated with every temperature measurement. As long as the alarm flag is set, the DS1820 will respond to the alarm search command. This allows many DS1820s to be connected in parallel doing simultaneous temperature measurements. If somewhere the temperature exceeds the limits, the alarming device(s) can be identified and read immediately without having to read non-alarming devices.

1-WIRE CRC CODE Figure 7

**MEMORY**

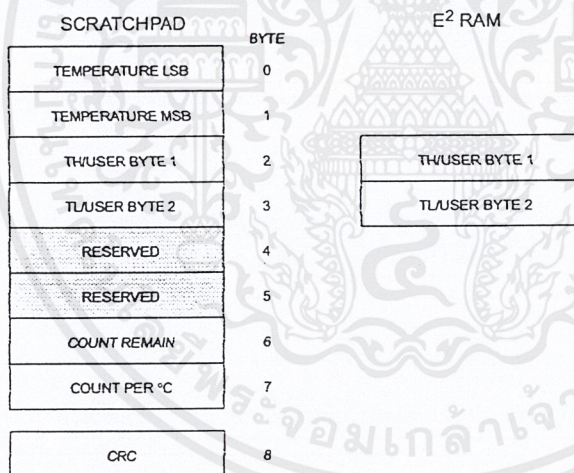
The DS1820's memory is organized as shown in Figure 8. The memory consists of a scratchpad RAM and a nonvolatile, electrically erasable (E²) RAM, which stores the high and low temperature triggers TH and TL. The scratchpad helps insure data integrity when communicating over the 1-Wire bus. Data is first written to the scratchpad where it can be read back. After the data has been verified, a copy scratchpad command will transfer the data to the nonvolatile (E²) RAM. This process insures data integrity when modifying the memory.

The scratchpad is organized as eight bytes of memory. The first two bytes contain the measured temperature

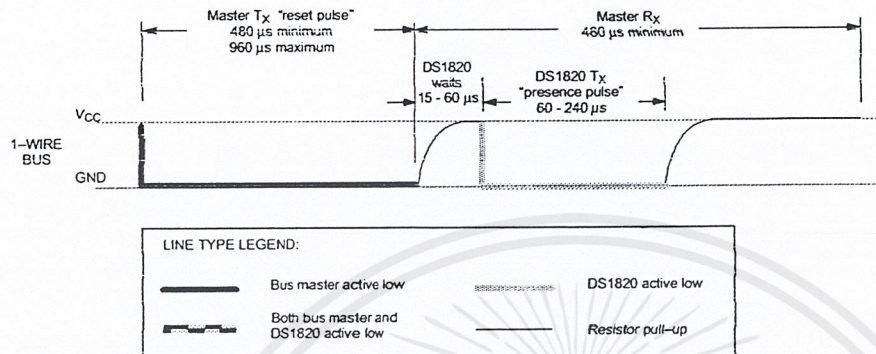
information. The third and fourth bytes are volatile copies of TH and TL and are refreshed with every power-on reset. The next two bytes are not used; upon reading back, however, they will appear as all logic 1's. The seventh and eighth bytes are count registers, which may be used in obtaining higher temperature resolution (see "Operation—measuring Temperature" section).

There is a ninth byte which may be read with a Read Scratchpad command. This byte contains a cyclic redundancy check (CRC) byte which is the CRC over all of the eight previous bytes. This CRC is implemented in the fashion described in the section titled "CRC Generation".

DS1820 MEMORY MAP Figure 8



INITIALIZATION PROCEDURE "RESET AND PRESENCE PULSES" Figure 11



DS1820 COMMAND SET Table 2

INSTRUCTION	DESCRIPTION	PROTOCOL	1-WIRE BUS AFTER ISSUING PROTOCOL	NOTES
TEMPERATURE CONVERSION COMMANDS				
Convert T	Initiates temperature conversion.	44h	<read temperature busy status>	1
MEMORY COMMANDS				
Read Scratchpad	Reads bytes from scratchpad and reads CRC byte.	BEh	<read data up to 9 bytes>	
Write Scratchpad	Writes bytes into scratchpad at addresses 2 and 3 (TH and TL temperature triggers).	4Eh	<write data into 2 bytes at addr. 2 and addr. 3>	
Copy Scratchpad	Copies scratchpad into nonvolatile memory (addresses 2 and 3 only).	48h	<read copy status>	2
Recall E ²	Recalls values stored in nonvolatile memory into scratchpad (temperature triggers).	B8h	<read temperature busy status>	
Read Power Supply	Signals the mode of DS1820 power supply to the master.	B4h	<read supply status>	

NOTES:

- Temperature conversion takes up to 500 ms. After receiving the Convert T protocol, if the part does not receive power from the V_{DD} pin, the I/O line for the DS1820 must be held high for at least 500 ms to provide power during the conversion process. As such, no other activity may take place on the 1-Wire bus for at least this period after a Convert T command has been issued.
- After receiving the Copy Scratchpad protocol, if the part does not receive power from the V_{DD} pin, the I/O line for the DS1820 must be held high for at least 10 ms to provide power during the copy process. As such, no other activity may take place on the 1-Wire bus for at least this period after a Copy Scratchpad command has been issued.

Read Scratchpad [BEh]

This command reads the contents of the scratchpad. Reading will commence at byte 0, and will continue through the scratchpad until the 9th (byte-8, CRC) byte is read. If not all locations are to be read, the master may issue a reset to terminate reading at any time.

Copy Scratchpad [48h]

This command copies the scratchpad into the E² memory of the DS1820, storing the temperature trigger bytes in nonvolatile memory. If the bus master issues read time slots following this command, the DS1820 will output "0" on the bus as long as it is busy copying the scratchpad to E²; it will return a "1" when the copy process is complete. If parasite powered, the bus master has to enable a strong pull-up for at least 10 ms immediately after issuing this command.

Convert T [44h]

This command begins a temperature conversion. No further data is required. The temperature conversion will be performed and then the DS1820 will remain idle. If the bus master issues read time slots following this command, the DS1820 will output "0" on the bus as long as it is busy making a temperature conversion; it will return a "1" when the temperature conversion is complete. If parasite powered, the bus master has to enable a strong pullup for 500 ms immediately after issuing this command.

Recall E2 [B8h]

This command recalls the temperature trigger values stored in E² to the scratchpad. This recall operation happens automatically upon power-up to the DS1820 as well, so valid data is available in the scratchpad as soon as the device has power applied. With every read data time slot issued after this command has been sent, the device will output its temperature converter busy flag "0"=busy, "1"=ready.

Read Power Supply [B4h]

With every read data time slot issued after this command has been sent to the DS1820, the device will signal its power mode: "0"=parasite power, "1"=external power supply provided.

READ/WRITE TIME SLOTS

DS1820 data is read and written through the use of time slots to manipulate bits and a command word to specify the transaction.

Write Time Slots

A write time slot is initiated when the host pulls the data line from a high logic level to a low logic level. There are two types of write time slots: Write One time slots and Write Zero time slots. All write time slots must be a minimum of 60 μ s in duration with a minimum of a one μ s recovery time between individual write cycles.

The DS1820 samples the I/O line in a window of 15 μ s to 60 μ s after the I/O line falls. If the line is high, a Write One occurs. If the line is low, a Write Zero occurs (see Figure 12).

For the host to generate a Write One time slot, the data line must be pulled to a logic low level and then released, allowing the data line to pull up to a high level within 15 μ s after the start of the write time slot.

For the host to generate a Write Zero time slot, the data line must be pulled to a logic low level and remain low for 60 μ s.

Read Time Slots

The host generates read time slots when data is to be read from the DS1820. A read time slot is initiated when the host pulls the data line from a logic high level to logic low level. The data line must remain at a low logic level for a minimum of one μ s; output data from the DS1820 is valid for 15 μ s after the falling edge of the read time slot. The host therefore must stop driving the I/O pin low in order to read its state 15 μ s from the start of the read slot (see Figure 12). By the end of the read time slot, the I/O pin will pull back high via the external pull-up resistor. All read time slots must be a minimum of 60 μ s in duration with a minimum of a one μ s recovery time between individual read slots.

Figure 13 shows that the sum of T_{INIT}, T_{RC}, and T_{SAMPLE} must be less than 15 μ s. Figure 14 shows that system timing margin is maximized by keeping T_{INIT} and T_{RC} as small as possible and by locating the master sample time towards the end of the 15 μ s period.

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	CONDITION	MIN	TYP	MAX	UNITS	NOTES
Supply Voltage	V_{DD}	I/O Functions	2.8	5.0	5.5	V	1, 2
		$\pm 1/2^\circ\text{C}$ Accurate Temperature Conversions	4.3		5.5		
Data Pin	I/O		-0.5		+5.5	V	2
Logic 1	V_{IH}		2.0		$V_{CC}+0.3$	V	2, 3
Logic 0	V_{IL}		-0.3		+0.8	V	2, 4

DC ELECTRICAL CHARACTERISTICS(-55°C to +125°C; $V_{DD}=3.6\text{V}$ to 5.5V)

PARAMETER	SYMBOL	CONDITION	MIN	TYP	MAX	UNITS	NOTES
Thermometer Error	t_{ERR}	-0°C to +70°C			$\pm 1/2$	°C	1, 9, 10
		-55°C to 0°C and +70°C to +125°C		See Typical Curve			
Input Logic High	V_{IH}		2.2		5.5	V	2, 3
Input Logic Low	V_{IL}		-0.3		+0.8	V	2, 4
Sink Current	I_L	$V_{I/O}=0.4V$	-4.0			mA	2
Standby Current	I_Q			200	350	nA	8
Active Current	I_{DD}			1	1.5	mA	5, 6
Input Load Current	I_L			5		μA	7

AC ELECTRICAL CHARACTERISTICS:

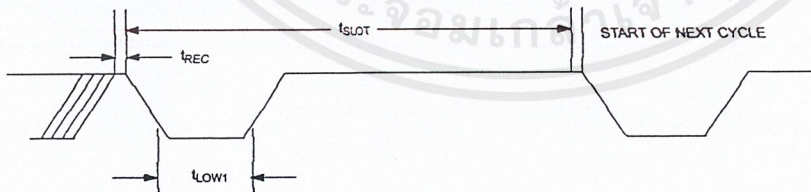
(-55°C to +125°C; $V_{DD}=3.6V$ to 5.5V)

PARAMETER	SYMBOL	MIN	TYP	MAX	UNITS	NOTES
Temperature Conversion Time	t_{CONV}		200	500	ms	
Time Slot	t_{SLOT}	60		120	μs	
Recovery Time	t_{REC}	1			μs	
Write 0 Low Time	t_{LOW0}	60		120	μs	
Write 1 Low Time	t_{LOW1}	1		15	μs	
Read Data Valid	t_{RDV}			15	μs	
Reset Time High	t_{RSTH}	480			μs	
Reset Time Low	t_{RSTL}	480		4800	μs	
Presence Detect High	t_{PDHIGH}	15		60	μs	
Presence Detect Low	t_{PDLOW}	60		240	μs	
Capacitance	$C_{IN/OUT}$			25	pF	

NOTES:

1. Temperature conversion will work with $\pm 2^\circ C$ accuracy down to $V_{DD} = 3.4$ volts.
2. All voltages are referenced to ground.
3. Logic one voltages are specified at a source current of 1 mA.
4. Logic zero voltages are specified at a sink current of 4 mA.
5. I_{DD} specified with V_{CC} at 5.0 volts.
6. Active current refers to either temperature conversion or writing to the E^2 memory. Writing to E^2 memory consumes approximately 200 μA for up to 10 ms.
7. Input load is to ground.
8. Standby current specified up to 70°C. Standby current typically is 5 μA at 125°C.
9. See Typical Curve for specification limits outside the 0°C to 70°C range. Thermometer error reflects sensor accuracy as tested during calibration.
10. Typical accuracy curve valid for $4.3V \leq V_{DD} \leq 5.5V$.

1-WIRE WRITE ONE TIME SLOT



TYPICAL PERFORMANCE CURVE

DS1820 DIGITAL THERMOMETER AND THERMOSTAT
TEMPERATURE READING ERROR