สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

เครื่องควบคุมการปิด-เปิด อัตโนมัติ โดยใช้ MCS-51

Automatic on-off system by MCS-51

T046540

โดย

นายธวัชชัย ทองสมุทร 43015772

นายวินัย ห่างภัย 43015784

อาจารย์ที่ปรึกษา

อาจารย์ สมภพ แก้วมีชัย

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรอุตสาหกรรมศาสตรบัณฑิต

สาขาเทคโนโลยีอิเล็กทรอนิกส์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2545

เลขหมู่.....
เลขทะเบียน 46540
วัน, เดือน, ปี 4 เม.ย. 2546

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

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อโครงงาน	**เครื่องควบคุมการปิด-เปิด อัตโนมัติ โดยใช้ MCS-51**	
	Automatic on-off system by MCS-51	
ชื่อนักศึกษา	นายธวัชชัย ทองสมุทร	43015772
	นายวินัย ห่างภัย	43015784
อาจารย์ที่ปรึกษา	อาจารย์ สมภพ แก้วมีชัย	
ปริญญา	อุตสาหกรรมศาสตรบัณฑิต	
ภาควิชา	วิศวกรรมสารสนเทศ	
สาขาวิชา	เทคโนโลยีอิเล็กทรอนิกส์	
ปีการศึกษา	2545	

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง อนุมัติให้ปริญญานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของการศึกษาหลักสูตรอุตสาหกรรมศาสตรบัณฑิต

คณะกรรมการสอบปริญญานิพนธ์

.....ประธานกรรมการ
( )

.....กรรมการ
( )

.....กรรมการ
( )

.....กรรมการ
( )

.....กรรมการ
( )

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์ปีการศึกษา 2545

ภาควิชา วิศวกรรมสารสนเทศ

คณะ วิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง เครื่องควบคุมการเปิด-ปิด อัตโนมัติ โดยใช้ MCS-51

(Automatic on-off system by MCS-51)

ผู้จัดทำ

นายวินัย ห่างภัย 43015784

นายธวัชชัย ทองสมุทร 43015772

..... อาจารย์ที่ปรึกษา

(อาจารย์ สมภพ แก้วมีชัย)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เครื่องควบคุมการปิด-เปิด อัตโนมัติ โดยใช้ MCS-51

Automatic on-off system by MCS-51

จัดทำโดย	นายวินัย ห่างภัย	43015784
	นายธวัชชัย ทองสมุทร	43015772

อาจารย์ที่ปรึกษา อาจารย์ สมภพ แก้วมีชัย

ปีการศึกษา 2545

บทคัดย่อ

ปริญญานิพนธ์ฉบับนี้ จะกล่าวถึงการควบคุมการเปิด-ปิด อุปกรณ์ไฟฟ้าภายในบ้านอัตโนมัติ โดยใช้ MCS-51 มาควบคุม มีจุดมุ่งหมายคือ ใช้ควบคุมอุปกรณ์ไฟฟ้าได้ 8 ช่องที่เป็นอิสระต่อกันและยังสามารถที่จะช่วยลดความเสียหายที่อาจจะเกิดขึ้นจากความประมาทของผู้ใช้ที่ใช้อุปกรณ์ไฟฟ้าได้อีกด้วยเช่น การใช้อุปกรณ์ไฟฟ้าเป็นระยะเวลานานๆ การเปิดอุปกรณ์ไฟฟ้าทิ้งไว้แล้วลืมปิด เป็นต้น โครงงานนี้จะใช้ไมโครคอนโทรลเลอร์มาทำการประมวลผลเพื่อควบคุมการเปิดปิดโดยจะใช้ไมโครเบอร์ AT89S8252 เพื่อให้เหมาะสมกับการใช้งานและยังมีจอ LCD โมดูลแบบ 16 ตัวอักษร 2 แถว มาแสดงผลเพื่อให้การใช้งานนั้นใช้ได้ง่ายยิ่งขึ้น ส่วนการแสดงผลทางโหลดจะใช้หลอดไฟ 220V มาแสดงผล

Abstract

This study is about controlling switching on and off of the home electrical appliances automatically with MCS-51. The aim is to control 8 independent electrical appliances and to reduce the damages caused by carelessness such as using the electrical appliances for a long time, turning on the appliances for a long time, etc. This project employs microcontroller to analyze the data in order to control switching on and off. The AT89S8252 microcontroller is used because of appropriateness. The 2 – row – 16 – digit LCD module moniter is used so that controlling can be easier. Moreover, the 220v bulbs are used to present the results.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

โครงงานนี้สำเร็จลงได้ด้วยดี ทางผู้จัดทำต้องขอขอบพระคุณ อาจารย์ สมภพ แก้วมีชัย อาจารย์ที่ปรึกษาโครงงานนี้ ที่คอยให้คำแนะนำ คำชี้แนะ ซึ่งเป็นประโยชน์ต่อการทำโครงงานนี้เป็นอย่างยิ่ง

ขอขอบพระคุณ ผู้เขียนเอกสาร หนังสือ และตำราอ้างอิงต่างๆทุกท่าน ที่ผู้จัดทำได้นำข้อมูลมาทำโครงงานนี้ จนเป็นผลสำเร็จ

ขอขอบพระคุณ บิดา มารดาของผู้จัดทำ ที่ได้อุปการะผู้จัดทำ ช่วยเหลือและให้กำลังใจแก่ผู้จัดทำ

ขอขอบคุณ เพื่อนๆพี่ๆทุกคน ที่คอยให้คำปรึกษา คำแนะนำต่างๆ แนวทางในการเขียนโปรแกรม ตลอดจนอุปกรณ์ในการทำงานมาโดยตลอด

ขอขอบพระคุณ ภาควิชาวิศวกรรมสารสนเทศ คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ที่ได้ให้ผู้จัดทำได้มีโอกาสเข้ามาศึกษา ณ สถาบันแห่งนี้

สุดท้ายนี้ ทางผู้จัดทำขอขอบพระคุณ คณาจารย์ทุกท่านที่ได้ให้คำชี้แนะ คำปรึกษา คำแนะนำ และประสิทธิ์ประสาทวิชาความรู้ให้กับผู้จัดทำ จนทำให้โครงงานนี้สำเร็จลงได้ด้วยดี

ผู้จัดทำ
นาย วินัย ห่างภัย
นาย ธวัชชัย ทองสมุทร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

เรื่อง	หน้า
บทที่ 1 บทนำ	1
1.1 แนวความคิดและที่มา	1
1.2 วัตถุประสงค์ของโครงงาน	1
1.3 ผลที่คาดว่าจะได้รับจากโครงงาน	1
1.4 ขอบเขตการทำงาน	1
1.5 ขั้นตอนการดำเนินโครงงาน	2
บทที่ 2 ไมโครคอนโทรลเลอร์	3
2.1 ความหมายของไมโครคอนโทรลเลอร์	3
2.2 โครงสร้างของไมโครคอนโทรลเลอร์	3
2.3 การจัดขาของไมโครคอนโทรลเลอร์ MCS-51	5
2.4 โครงสร้างและการทำงานของพอร์ต	8
2.5 การใช้งานเป็นพอร์ตอินพุต	10
2.6 การใช้งานเป็นพอร์ตเอาท์พุต	11
2.7 หน่วยความจำโปรแกรม (Program memory)	11
2.8 หน่วยความจำข้อมูล (Data memory)	14
2.9 การสื่อสารผ่านพอร์ตอนุกรมของไมโครคอนโทรลเลอร์ MCS-51	16
บทที่ 3 รีลไทม์คล็อก (RTC)	18
3.1 ไอซีสร้างฐานเวลาจริงหรือรีลไทม์คล็อก (RTC)	18
3.2 ลายละเอียดขาต่อการใช้งานของ DS1307	18
3.3 การทำงานของ DS1307	19
3.4 โหมดการทำงานของ DS1307	20
บทที่ 4 หน่วยแสดงผลข้อมูล (LCD MODULE)	23
4.1 ชุดอ่านหรือเขียนข้อมูลและการแสดงข้อความ	25
4.2 ความเข้าใจพื้นฐาน	26
4.3 ลายละเอียดของแต่ละคำสั่ง	27
4.4 การอ่านและเขียนข้อมูลกับ DDRAM/CGRAM	31

เอกสารนี้เป็นเอกสาร... เพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

เรื่อง	หน้า
4.5 แนวทางการเขียนโปรแกรมควบคุม	32
บทที่ 5 การวิเคราะห์และการออกแบบ	33
5.1 ส่วนของฮาร์ดแวร์(Hardware)	33
5.1.1 การออกแบบส่วนสวิตช์ คีย์แพด (Switch Keypad)	34
5.1.2 การออกแบบส่วนไมโครคอนโทรลเลอร์ (Microcontroller)	35
5.1.3 การออกแบบส่วนการแสดงผล (Display)	36
5.1.4 การออกแบบส่วนควบคุมอุปกรณ์ไฟฟ้า (Load control)	37
5.1.5 การออกแบบส่วนภาคจ่ายไฟ (Supply)	39
5.1.6 การออกแบบรูปลักษณะภายนอกของโครงงาน (Box)	40
5.2 ส่วนของซอฟท์แวร์ (Software)	42
บทที่ 6 การทดลองและผลการทดลอง	44
6.1 การทดลองและผลการทดลองทางด้านฮาร์ดแวร์ (Hardware)	44
6.2 การทดลองและผลการทดลองทางด้านซอฟท์แวร์ (Software)	47
6.2.1 โปรแกรมของเครื่องควบคุมการปิด-เปิด อัตโนมัติ โดยใช้ MCS-51	48
ภาคผนวก	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูปภาพ

รูป หน้า

รูปที่ 2.1 โครงสร้างพื้นฐานของไมโครคอนโทรลเลอร์ MCS-51แบบแฟลช AT89Sxx 4
รูปที่ 2.2 รายละเอียดโครงสร้างหลักของไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลช 6
รูปที่ 2.3 การจัดขามาตรฐานของไมโครคอนโทรลเลอร์เบอร์ AT89S8252 6
รูปที่ 2.4 วงจรภายในของพอร์ตทุกพอร์ตในไมโครคอนโทรลเลอร์ MCS-51 9
รูปที่ 2.5 วงจรพูลอัปของไมโครคอนโทรลเลอร์ MCS-51 10
รูปที่ 2.6 การจัดสรรหน่วยความจำโปรแกรมของไมโครคอนโทรลเลอร์ MCS-51 12
(ก) การจัดสรรหน่วยความจำโปรแกรมของไมโครคอนโทรลเลอร์เบอร์ AT89C51 12
(ข) การจัดสรรหน่วยความจำโปรแกรมของไมโครคอนโทรลเลอร์เบอร์ AT89C52 12
รูปที่ 2.7 การเชื่อมต่อหน่วยความจำโปรแกรมภายนอกของ MCS-51 13
รูปที่ 2.8 การเชื่อมต่อหน่วยความจำข้อมูลภายนอกของ MCS-51 14
รูปที่ 2.9 การจัดสรรพื้นที่ของหน่วยความจำข้อมูลภายใน MCS-51 15
รูปที่ 2.10 การจัดสรรพื้นที่ของหน่วยความจำข้อมูลภายในส่วนล่างของ MCS-51 15
รูปที่ 3.1 การจัดขาของไอซี DS1307 18
รูปที่ 3.2 โครงสร้างภายในของไอซี DS1307 20
รูปที่ 3.3 รูปแบบของข้อมูลสำหรับติดต่อกับ DS1307 ในโหมดการเขียนข้อมูล 21
รูปที่ 3.4 รูปแบบของข้อมูลสำหรับติดต่อกับ DS1307 ในโหมดการอ่านข้อมูล 21
รูปที่ 3.5 การเชื่อมต่อไมโครคอนโทรลเลอร์กับไอซีรีลไทม์คล็อก DS1307 22
รูปที่ 4.1 แสดงวงจรภาคแสดงผล LCD 24
รูปที่ 5.1 บล็อกไดอะแกรมของเครื่องควบคุมการเปิด-ปิดอุปกรณ์ไฟฟ้าอัตโนมัติ 33
รูปที่ 5.2 การต่อสวิตช์ คีย์แพด กับไมโครคอนโทรลเลอร์ 34
รูปที่ 5.3 วงจรของสวิตช์แบบเมตริกซ์ หรือคีย์แพด 34
รูปที่ 5.4 การใช้งานของพอร์ตของไมโครคอนโทรลเลอร์ 36
รูปที่ 5.5 ลายวงจรของภาคไมโครคอนโทรลเลอร์ 36
รูปที่ 5.6 การต่อจอ LCD โมดูล กับ MCS-51 37
รูปที่ 5.7 วงจรภายในของไอซี ULN2803A 37

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูปภาพ (ต่อ)

รูป หน้า

รูปที่ 5.8 วงจรควบคุมรีเลย์ เพื่อใช้ตัดต่อการเปิด-ปิดอุปกรณ์ไฟฟ้า 38
รูปที่ 5.9 ลายวงจรของภาคควบคุมโหลด 39
รูปที่ 5.10 วงจรของภาคจ่ายไฟ 39
รูปที่ 5.11 ลายวงจรของภาคจ่ายไฟ 40
รูปที่ 5.12 ภาพด้านบนของลักษณะภายนอกของโครงงาน 40
รูปที่ 5.13 ภาพด้านหลังของลักษณะภายนอกของโครงงาน 41
รูปที่ 5.14 ภาพมุมเอียงของลักษณะภายนอกของโครงงาน 41
รูปที่ 6.1 วงจรจ่ายไฟที่บัดกรีเสร็จแล้ว 44
รูปที่ 6.2 วงจรไมโครคอนโทรลเลอร์ที่บัดกรีเสร็จแล้ว 45
รูปที่ 6.3 วงจรที่ใช้ควบคุมโหลดที่บัดกรีเสร็จแล้ว 45
รูปที่ 6.4 การต่อวงจรก่อนการทดลอง 46
รูปที่ 6.5 แสดงผลการทดลองที่ไม่มีความผิดพลาดในวงจร 46
รูปที่ 6.6 แสดง flowchart ของโปรแกรม 47

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

ตาราง **หน้า**

ตารางที่ 2.1 หน้าที่พิเศษของพอร์ต 1 ในไมโครคอนโทรลเลอร์ MCS-51 8
ตารางที่ 4.1 หน้าที่การทำงานของแต่ละขาของจอแสดงผล LCD 25
ตารางที่ 5.1 ความหมายของปุ่มต่างๆของสวิตช์ คีย์แพด 35

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1 บทนำ

1.1 แนวความคิดและที่มา

เนื่องจากในตอนนี้ เมื่อคนเราต้องการจะใช้อุปกรณ์ไฟฟ้า หรืออุปกรณ์อิเล็กทรอนิกส์ เช่น พัดลม วิทยุ ทีวี หลอดไฟ ฯลฯ นั้น เราจำเป็นต้องควบคุม การปิด-เปิดอุปกรณ์นั้นๆเอง หรืออุปกรณ์ไฟฟ้านั้นๆ อาจจะมีวงจรควบคุมการปิด-เปิดอยู่แล้วก็ได้ แต่ในบางครั้งอาจจะเกิดจากความประมาทของผู้ใช้นั้นๆเอง เช่น เปิดอุปกรณ์ไฟฟ้าทิ้งไว้ แล้วเกิดลืมปิดอุปกรณ์นั้นๆ ซึ่งความประมาท ต่างๆเหล่านี้อาจจะก่อให้เกิดความเสียหายต่อไปได้ แต่ถ้าเมื่อเทียบกับระบบการควบคุมการเปิด-ปิด อัตโนมัติ ที่ใช้ MCS-51 นี้แล้ว ระบบนี้จะสามารถเปิดและปิด อุปกรณ์ไฟฟ้า หรืออุปกรณ์อิเล็กทรอนิกส์ เองได้ โดยง่าย และอีกทั้งยังลดความเสียหายที่จะเกิดขึ้นได้มากอีกด้วย และที่สำคัญ ยังเป็นการฝึกให้ผู้จัดทำเกิดความชำนาญและมีความรู้เพิ่มเติมในทฤษฎีของ MCS-51 ได้อีกด้วย

1.2 วัตถุประสงค์ของโครงงาน

1. เพื่อใช้ตั้งเวลาการเปิด-ปิดอุปกรณ์ไฟฟ้าโดยอัตโนมัติโดยผ่านการควบคุมจากไมโครคอนโทรลเลอร์ MCS-51
2. เพื่อลดความเสียหายที่อาจจะเกิดขึ้นจากความประมาทของผู้ใช้ได้
3. เพื่อเพิ่มเติมความรู้แก่ผู้จัดทำและผู้ใช้งานทั้งในด้านทฤษฎีและปฏิบัติ
4. เพื่อนำโครงงานนี้ไปปฏิบัติงานได้จริง

1.3 ผลที่คาดว่าจะได้รับจากโครงงาน

1. สามารถใช้ควบคุมเวลาในการเปิด-ปิดของอุปกรณ์ไฟฟ้านั้นได้ทั้งหมด 8 CHANNEL
2. สามารถลดความผิดพลาดและความเสียหายที่อาจจะเกิดขึ้นแก่ผู้ใช้ได้
3. ทำให้ผู้จัดทำและผู้ใช้งานมีความรู้จากทฤษฎีของโครงงานนี้เพิ่มเติมและสามารถนำไปประยุกต์ใช้งานต่างๆไปได้อีกด้วย
4. สามารถนำโครงงานนี้ไปใช้งานควบคู่กับอุปกรณ์ไฟฟ้าได้จริง

1.4 ขอบเขตการทำงาน

1. สามารถโปรแกรมตั้งเวลาการเปิด-ปิดเครื่องใช้ไฟฟ้าได้ 8 CHANNEL
2. แต่ละช่องสามารถโปรแกรมได้อย่างอิสระต่อกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. ความละเอียดของโปรแกรมสามารถตั้งเวลามากที่สุดเป็นนาที เช่น 9.30 , 5.51เป็นต้น
4. สามารถโปรแกรมแต่ละช่องให้เปิด-ปิดได้เป็นชั่วโมงและนาที
5. ป้องกันการเกิดอุบัติเหตุจากความประมาทของผู้ใช้
6. ไม่ต้องใช้ COMPUTER มาเกี่ยวข้องในเวลาใช้งานจริงเหมาะสำหรับผู้ใช้งานที่ไม่ถนัดด้าน COMPUTER

1.5 ขั้นตอนการดำเนินโครงงาน

1. ศึกษาหาข้อมูลเพื่อเป็นแนวทางในการทำโครงงาน
2. ทำการทดลองจากแนวทางที่ได้ศึกษามา
3. วิเคราะห์ผลที่ได้จากการทดลองว่าเป็นไปตามเป้าหมายที่ต้องการของโครงงานหรือไม่ และควรจะลดหรือเพิ่มเติมส่วนใดบ้าง
4. นำข้อมูลที่ได้จากการวิเคราะห์ที่เป็นไปตามเป้าหมายของโครงงาน มาดำเนินการทำโครงงาน
5. สรุปปัญหาต่างๆที่เกิดขึ้นในระหว่างการทดลองและการทำโครงงาน
6. นำปัญหาต่างๆที่ได้มาทำการแก้ไข
7. ทำการวิเคราะห์ถึงแนวทางในการพัฒนาโครงงานต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2 ไมโครคอนโทรลเลอร์

2.1 ความหมายของไมโครคอนโทรลเลอร์

ไมโครคอนโทรลเลอร์ (microcontroller) เป็นชื่อของอุปกรณ์อิเล็กทรอนิกส์แบบหนึ่งที่บรรจุความสามารถมากมายไม่ว่าจะเป็นหน่วยประมวลผล หน่วยคำนวณทางคณิตศาสตร์และลอจิก วงจรรับสัญญาณอินพุต วงจรขับสัญญาณออกทางเอาต์พุต หน่วยความจำ วงจรกำเนิดสัญญาณนาฬิกาทำให้ไมโครคอนโทรลเลอร์สามารถนำไปประยุกต์ใช้งานแทนวงจรอิเล็กทรอนิกส์ที่ซับซ้อนได้เป็นอย่างดีโดยช่วยลดจำนวนของอุปกรณ์และขนาดของระบบลงในขณะที่มีขีดความสามารถสูงขึ้น ภายใต้งบประมาณที่เหมาะสม

ไมโครคอนโทรลเลอร์มาจากคำ 2 คำรวมกันคือ" ไมโคร " (micro) ซึ่งหมายถึงไมโครโปรเซสเซอร์ (microprocesser) ซึ่งเป็นอุปกรณ์ประมวลผลข้อมูลขนาดเล็กซึ่งภายในประกอบด้วยหน่วยประมวลผลกลางหรือซีพียู (CPU:Central Processing Unit) หน่วยคำนวณทางคณิตศาสตร์และลอจิก (ALU:Arithmatic Logic Unit) วงจรเชื่อมต่อหน่วยความจำ และวงจรเชื่อมต่อสัญญาณนาฬิกา อีกคำหนึ่งคือคำว่า" คอนโทรลเลอร์ " (controller) หมายถึง อุปกรณ์ควบคุม ดังนั้น ไมโครคอนโทรลเลอร์จึงเป็นอุปกรณ์ที่ใช้ในการควบคุม โดยที่สามารถเขียนโปรแกรมเพื่อกำหนดรูปแบบการควบคุมได้อย่างอิสระ

2.2 โครงสร้างของไมโครคอนโทรลเลอร์

2.2.1 คุณสมบัติของไมโครคอนโทรลเลอร์ตระกูล MCS-51

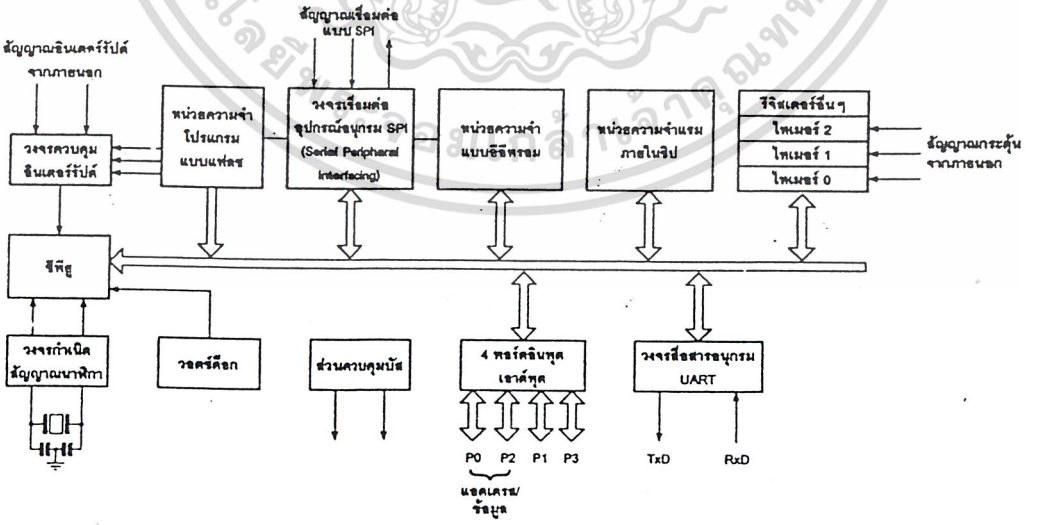
1. ต้องการแหล่งจ่ายไฟ +5V ชุดเดียว
2. มีหน่วยความจำโปรแกรม(Program Memory)ขนาด 4 กิโลไบต์สำหรับเบอร์ 8051 และ 8031,8032 ไม่มีหน่วยความจำชุดนี้ ส่วน 8052 มีหน่วยความจำถึง 8 กิโลไบต์
3. มีหน่วยความจำสำหรับเก็บข้อมูล(Data Memory) ขนาด 128 ไบต์ สำหรับ 8052 มีถึง 256 ไบต์
4. หน่วยความจำสำหรับโปรแกรมและข้อมูล (Program Memory) และ Data Memory แยกจากกันอย่างละ 64 กิโลไบต์
5. คำสั่งที่ใช้เวลาน้อยที่สุดประมาณ 1uS เมื่อทำงานที่ความถี่ 12MHZ
6. มี Time/Counter ขนาด 16 บิต 2 ชุด
7. รับอินเตอร์รัปต์ได้ 6 แหล่ง 5 เวคเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

8. มีพอร์ตสำหรับส่งข้อมูลอนุกรม(UART) 2 พอร์ต ทั้งรับและส่งในเวลาเดียวกันได้ (Full Duplex) เลือกรูปแบบการส่งข้อมูลได้ 4 โหมด
9. มีคำสั่งในการทำ AND,OR หรือ COMPLEMENT ได้ทั้งแบบ 8 บิต และ 1 บิต

2.2.2 คุณสมบัติทางเทคนิคของไมโครคอนโทรลเลอร์ตระกูล MCS-51 เบอร์ AT89xx

- เป็นไมโครคอนโทรลเลอร์ที่ใช้ซีพียูขนาด 8 บิต
- ภายในมีหน่วยความจำโปรแกรมเป็นแบบแฟลชสามารถลบและเขียนใหม่ได้พันครั้ง
- หน่วยความจำข้อมูลพื้นฐานเป็นหน่วยความจำแบบแรม ในบางเบอร์จะมีหน่วยความจำแบบอีอีพรอมเพิ่มเติม
- ขาพอร์ตเป็นแบบสองทิศทาง สามารถใช้งานเป็นได้ทั้งอินพุตและเอาต์พุต
- มีวงจรสื่อสารอนุกรมแบบฟูลดูเพล็กซ์
- ไทเมอร์/เคาร์เตอร์ขนาด 16 บิตอย่างน้อย 2 ตัว
- สามารถรองรับแหล่งกำเนิดอินเตอร์รัปต์ได้ 6 ประเภท
- สามารถขยายหน่วยความจำภายนอกเพิ่มเติมได้สูงสุด 64 กิโลไบต์
- มีวงจรกำเนิดสัญญาณนาฬิกาอยู่ในภายชิป
- มีวงจรสื่อสารอนุกรมแบบ SPI สำหรับในอนุกรม AT89Sxx
- มีวอตช์ด็อกไทเมอร์ในตัว สำหรับในอนุกรม AT89Sxx



รูปที่ 2.1 โครงสร้างพื้นฐานของไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลช AT89Sxx

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในรูปที่ 2.2 เป็นโครงสร้างพื้นฐานของอนุกรม AT89Sxx จะเห็นได้ว่ามีส่วนประกอบที่เพิ่มเติมแตกต่างจาก AT89Cxx เช่น วงจรเชื่อมต่ออนุกรมแบบ SPI ซึ่งในไมโครคอนโทรลเลอร์อนุกรมนี้ใช้ในการเขียนข้อมูลลงในหน่วยความจำโปรแกรมโดยไม่ต้องถอดตัว ชิปออกไปจากระบบหรือเรียกว่า การโปรแกรมในวงจร ไทเมอร์/เคาน์เตอร์ขนาด 16 บิตที่เพิ่มเติมเข้ามาอีกหนึ่งตัวเป็นไทเมอร์ 2 และวงจรวอตช์ด็อกที่ใช้ในการตรวจสอบการทำงานผิดพลาดของซีพียู

2.3 การจัดขาของไมโครคอนโทรลเลอร์ MCS-51

ไมโครคอนโทรลเลอร์ MCS-51 ทุกเบอร์จะมีสถาบัตยกรรมและขาใช้งานพื้นฐานเหมือนกันดังแสดงในรูปที่ 2.2 และ 2.3 โดยมีรายละเอียดขั้นต้น ดังนี้

ขา Vcc ใช้สำหรับต่อไฟเลี้ยง +5V

ขา GND เป็นขากราวด์ สำหรับต่อกับกราวด์ของระบบ

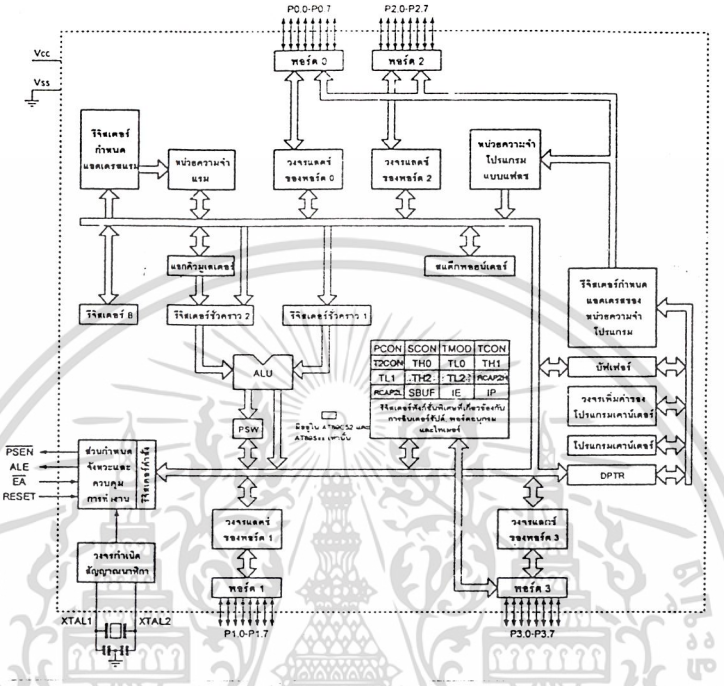
ขาพอร์ต 0 (P0.0-P0.7) มี 8 ขาแต่ละขาสามารถกำหนดให้เป็นได้ทั้งอินพุตและเอาต์พุต สำหรับใช้งานทั่วไปถ้าหากต้องการกำหนดให้ขาพอร์ต 0 ขาใดขาหนึ่งเป็นอินพุตสามารถทำได้โดยการเขียนข้อมูล"1"ไปยังแต่ละบิตของพอร์ตที่ต้องการติดต่อด้วยส่งผลให้ขาพอร์ตนั้นมีสถานะปล่อยลอย (float) จึงมีอินพุตอิมพีแดนซ์สูง สามารถใช้งานเป็นขาพอร์ตอินพุตได้ นอกจากนั้นขาพอร์ตนี้ยังถูกใช้งานในการติดต่อกับขาแอดเดรสไบต์ต่ำของหน่วยความจำภายนอก (A0-A7) และขาข้อมูล (D0-D7) โดยใช้กระบวนการมัลติเพล็กซ์เข้าช่วย เพื่อสลับการทำงานให้เป็นได้ทั้งขาติดต่อแอดเดรสและขาข้อมูล

ขาพอร์ต 1 (P1.0-P1.7) มี 8 ขาแต่ละขาสามารถกำหนดให้เป็นได้ทั้งอินพุตและเอาต์พุต สำหรับใช้งานทั่วไป ถ้าหากต้องการกำหนดให้ขาพอร์ต 0 ขาใดขาหนึ่งเป็นอินพุตสามารถทำได้โดยการเขียนข้อมูล "1" ไปยังแต่ละบิตของพอร์ตที่ต้องการติดต่อด้วย นอกจากนั้นในอนุกรม AT89Sxx จะใช้ขา P1.0 เป็นขาอินพุตสำหรับนับค่าของไทเมอร์ 2 และ P1.1 เป็นขาอินพุตทริกเกอร์ของไทเมอร์ 2 ในขณะที่ขา P1.4 ถึง P1.7 เป็นขาสำหรับเชื่อต่อแบบ SPI เพื่อทำการโปรแกรมข้อมูลในระบบ

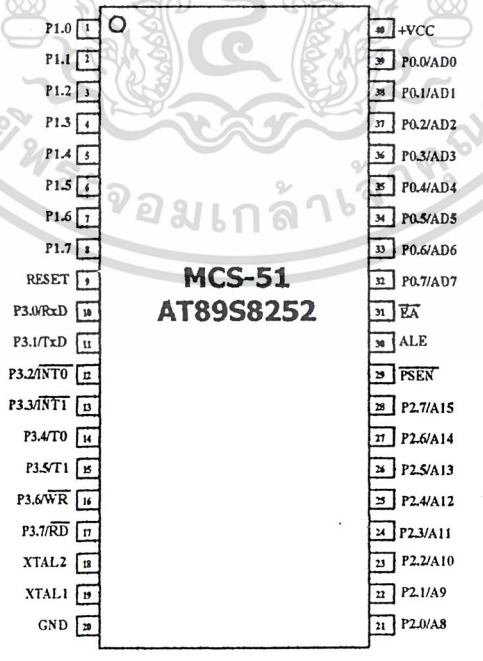
ขาพอร์ต 2 (P2.0-P2.7) มี 8 ขาแต่ละขาสามารถกำหนดให้เป็นได้ทั้งอินพุตและเอาต์พุต สำหรับใช้งานทั่วไป ถ้าหากต้องการกำหนดให้ขาพอร์ต 0 ขาใดขาหนึ่งเป็นอินพุตสามารถทำได้โดยการเขียนข้อมูล"1"ไปยังแต่ละบิตของพอร์ตที่ต้องการติดต่อด้วย ส่งผลให้ขาพอร์ตนั้นมีสถานะปล่อยลอย (float) จึงมีอินพุตอิมพีแดนซ์สูง สามารถใช้งานเป็นขาพอร์ตอินพุตได้ นอกจากนั้นขาพอร์ตนี้ยังถูกใช้งานในการติดต่อกับขาแอดเดรสไบต์สูงของหน่วยความจำภายนอก (A8-A15)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขาพอร์ต 3 (P3.0-P3.7) มี 8 ขาแต่ละขาสามารถกำหนดให้เป็นได้ทั้งอินพุตและเอาต์พุต สำหรับใช้งานทั่วไป ถ้าหากต้องการกำหนดให้ขาพอร์ต 0 ขาใดขาหนึ่งเป็นอินพุตสามารถทำได้



รูปที่ 2.2 รายละเอียดโครงสร้างหลักของไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลช



รูปที่ 2.3 การจัดขามาตรฐานของไมโครคอนโทรลเลอร์เบอร์ AT89S8252

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยการเขียนข้อมูล"1"ไปยังแต่ละบิตของพอร์ตที่ต้องการติดต่อด้วย ส่งผลให้ขาพอร์ตนั้นมีสถานะปล่อยลอย (float) จึงมีอินพุตอิมพีแดนซ์สูง สามารถใช้งานเป็นขาพอร์ตอินพุตได้ นอกจากนั้นขาพอร์ต 3 ยังเป็นขาที่มีหน้าที่การใช้งานพิเศษ ดังมีรายละเอียดขั้นต้นดังต่อไปนี้

P3.0 ใช้เป็นขาอินพุตสำหรับรับข้อมูลจากการสื่อสารแบบอนุกรม หรือขา RxD

P3.1 ใช้เป็นขาอินพุตสำหรับส่งข้อมูลจากการสื่อสารแบบอนุกรม หรือขา TxD

P3.2 ใช้เป็นขาอินพุตสำหรับรับสัญญาณอินเตอร์รัปต์จากภายนอกช่องที่ 0 หรือขา $\overline{\text{INT0}}$

P3.3 ใช้เป็นขาอินพุตสำหรับรับสัญญาณอินเตอร์รัปต์จากภายนอกช่องที่ 1 หรือขา $\overline{\text{INT1}}$

P3.4 ใช้เป็นขาอินพุตสำหรับรับสัญญาณไทเมอร์จากภายนอกช่องที่ 0 หรือขา T0

P3.5 ใช้เป็นขาอินพุตสำหรับรับสัญญาณอินเตอร์รัปต์จากภายนอกช่องที่ 1 หรือขา T1

P3.6 ใช้เป็นขาสัญญาณ $\overline{\text{WR}}$ ในกรณีที่ใช้เชื่อมต่อกับหน่วยความจำภายนอก

P3.7 ใช้เป็นขาสัญญาณ $\overline{\text{RD}}$ ในกรณีที่ใช้เชื่อมต่อกับหน่วยความจำภายนอก

ขารีเซต ใช้ในการรีเซตการทำงานของไมโครคอนโทรลเลอร์ โดยในการป้อนสัญญาณเพื่อรีเซต สถานะที่ขานี้ต้องอยู่ในระดับรีเซตอย่างน้อย2 แมชชีนไซเกิล โดยที่วงจรกำเนิดสัญญาณนาฬิกายังคงทำงานต่อเนื่องไปอย่างเป็นปกติ

ขา $\overline{\text{ALE/PROG}}$(Address Latch Enable/Program pulse input) เป็นขาที่ใช้ในการควบคุมการแลตซ์ของขาพอร์ต 0 เมื่อมีการใช้งานหน่วยความจำภายนอก นอกจากนั้นขานี้ยังใช้เป็นขาสำหรับรับพัลส์ของการโปรแกรมสำหรับโปรแกรมข้อมูลลงในไมโครคอนโทรลเลอร์ MCS-51 ในรุ่นที่มีหน่วยความจำเป็นแบบอีอีพรอม

ขา PSEN(ProgramStore Enable) ขานี้ใช้ในการส่งสัญญาณเพื่อร้องขอติดต่อกับหน่วยความจำโปรแกรมภายนอก เมื่อไมโครคอนโทรลเลอร์ต้องการอ่านข้อมูลจากหน่วยความจำโปรแกรมภายนอกตัวไมโครคอนโทรลเลอร์ ต้องการข้อมูลจากหน่วยความจำโปรแกรมภายนอกตัวไมโครคอนโทรลเลอร์ จะส่งสัญญาณออกมาที่ขานี้ 2 ครั้งในแต่ละแมชชีนไซเกิล แต่ถ้าหากติดต่อกับหน่วยความจำข้อมูลภายนอก ขานี้จะไม่มีการส่งสัญญาณใดๆออกมา

ขา $\overline{\text{EA}}$ / Vpp(External Access enable/Programming voltage input) ใช้สำหรับเลือกการติดต่อหน่วยความจำโปรแกรมภายนอกหรือภายในตัวไมโครคอนโทรลเลอร์ ถ้าหากขานี้เป็น "0"เป็นการเลือกให้ไมโครคอนโทรลเลอร์ติดต่อกับหน่วยความจำโปรแกรมภายนอก แต่หากขานี้เป็น"1"เป็นการเลือกให้ไมโครคอนโทรลเลอร์ติดต่อกับหน่วยความจำภายในตัวไมโครคอนโทรลเลอร์นอกจากนี้ที่ขานี้ยังใช้เป็นขาอินพุตสำหรับแรงดันไฟสูงสำหรับการโปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หน่วยความจำภายในไมโครคอนโทรลเลอร์ สำหรับไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลช ต้องการแรงดันสำหรับการโปรแกรม +12V

ขา XTAL1 และ XTAL2 เป็นขาสำหรับต่อคริสตอลเพื่อสร้างสัญญาณนาฬิกาในการกำหนดจังหวะการทำงานของไมโครคอนโทรลเลอร์

2.4 โครงสร้างและการทำงานของพอร์ต

ไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลชมีพอร์ตให้ใช้งานทั้งสิ้น 4 พอร์ตคือ พอร์ต 0 ถึงพอร์ต 3 แต่ละพอร์ตมีขนาด 8 บิต เป็นพอร์ตแบบ 2 ทิศทาง กล่าวคือ สามารถเป็นได้ทั้งอินพุตสำหรับสัญญาณข้อมูลเข้าและเอาต์พุตสำหรับส่งสัญญาณข้อมูลออกทุกพอร์ตของไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลชมีวงแลตช์และวงจรขับตลอดจนบัฟเฟอร์อินพุต ดังแสดงให้เห็นในสถาปัตยกรรมรูปที่ 2.2

ขา	เบอร์ของ ไมโครคอนโทรลเลอร์	หน้าที่พิเศษ
P1.0	AT89C52/AT89Sxx	ขาT2 เป็นขาอินพุตนับค่าของไทเมอร์/เคาน์เตอร์ 2 และเป็นขาเอาต์พุตของการกำเนิดสัญญาณนาฬิกาโดยไทเมอร์2 (clock out)
P1.1	AT89C52/AT89Sxx	ขาT2EX เป็นขาอินพุตทริกเกอร์สำหรับการแคปเจอร์/รีโหลด และควบคุมทิศทางของสัญญาณ
P1.4	AT89Sxx	ขา $\overline{SS}$(Slave Select)เป็นขาเลือกการติดต่อในกรณีที่ไมโครคอนโทรลเลอร์เป็นอุปกรณ์สเลฟ ในระบบติดต่อแบบ SPI
P1.5	AT89Sxx	ขา MOSI (Master data input, Slave data input) ใช้ในการติดต่อกับพอร์ต SPI
P1.6	AT98Sxx	ขา MISO(Master data input, Slave data output) ใช้ในการติดต่อกับพอร์ต SPI
P1.7	AT89Sxx	ขา SCK (Master clock output) เป็นขาสัญญาณนาฬิกาของการติดต่อกับพอร์ต SPI

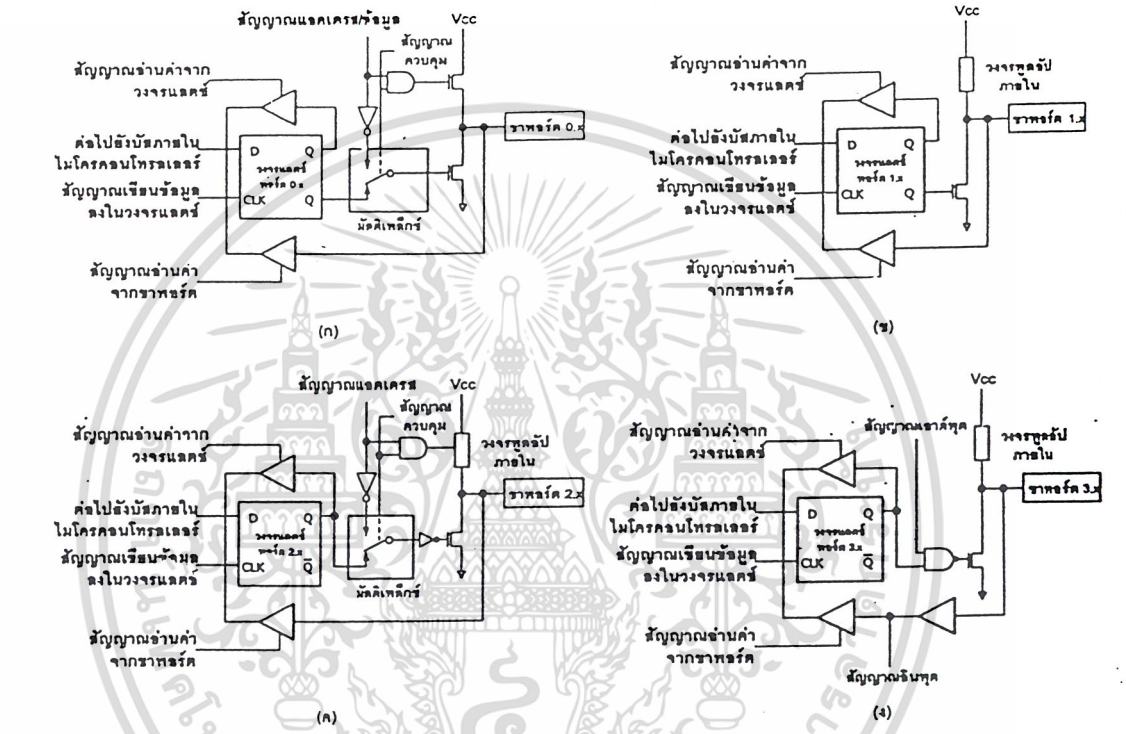
ตารางที่ 2.1 หน้าที่พิเศษของพอร์ต 1 ในไมโครคอนโทรลเลอร์ MCS-51

ที่พอร์ต 0 และพอร์ต 2 จะใช้งานเป็นพอร์ตอินพุตและเอาต์พุตสำหรับงานทั่วไป และในการติดต่อกับหน่วยความจำภายนอก สำหรับพอร์ต 3 ทั้งพอร์ตและพอร์ต 1 บางขานอกจากจะใช้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เป็นขาพอร์ตอินพุตเอาต์พุตตามปกติแล้ว ยังสามารถใช้งานในหน้าที่พิเศษได้อีก ขึ้นอยู่กับว่าเป็นไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลชเบอร์ใด ดังสรุปได้ในตารางที่ 2.1

ในรูปที่ 2.4 แสดงวงจรภายในของแต่ละพอร์ตของไมโครคอนโทรลเลอร์MCS-51 แบบแฟลชโดยในรูปที่ 2.4(ก) เป็นวงจรของพอร์ต 0 วงจรแลตช์ของแต่ละบิต ในแต่ละพอร์ตก็คือวงจร



รูปที่ 2.4 วงจรภายในของพอร์ตทุกพอร์ตในไมโครคอนโทรลเลอร์ MCS-51

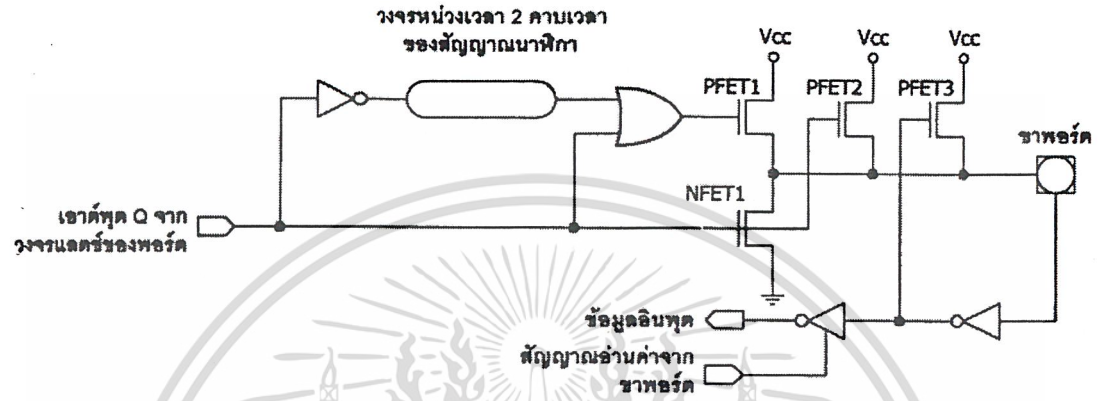
ดีฟลิฟฟลอปนั่นเองการอ่านค่าสถานะของพอร์ตและสถานะของวงจรแลตช์สามารถกระทำได้อย่างอิสระต่อกันด้วยสัญญาณที่แยกจากกัน นั่นคือ สัญญาณอ่านข้อมูลจากขาพอร์ต และสัญญาณอ่านข้อมูลจากวงจรแลตช์ ส่วนการเขียนข้อมูลมายังพอร์ตต้องส่งข้อมูลมายังขา CLK ของดีฟลิฟลอปในขณะที่ข้อมูลจะส่งผ่านมายังขาบัสข้อมูลภายในเข้าสู่ขา D ของดีฟลิปฟลอป

ที่พอร์ตนี้มีวงจรมัลติเพล็กซ์สำหรับกำหนดลักษณะการทำงานของพอร์ตว่าต้องการใช้งานเป็นขาพอร์ตอินพุตเอาต์พุตปกติหรือใช้ในการติดต่อกับหน่วยความจำภายนอกไมโครคอนโทรลเลอร์

เนื่องจากที่ขาพอร์ต 0 ไม่มีวงจรพูลอัปภายใน หากมีการนำพอร์ต 0 ไปใช้งานเป็นพอร์ตอินพุตจะต้องต่อตัวต้านทานพูลอัปภายนอกเข้าที่ขาพอร์ต 0 ทุกขาด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในรูปที่ 2.4 (ข) เป็นวงจรของพอร์ต 1 ซึ่งมีลักษณะโดยทั่วไปคล้ายกับพอร์ต 0 หากแต่ไม่มีวงจรมัลติเพล็กซ์เนื่องจากพอร์ตนี้จะไม่ใช้ในการติดต่อกับหน่วยความจำภายนอก แต่จะมีวงจรพูลอัปภายในที่แต่ละบิตของพอร์ตนี้แทน สำหรับรายละเอียดของวงจรพูลอัปแสดงในรูปที่ 2.5



วงจรพูลอัปประกอบด้วยเฟตชนิดพีแชนเนล 3 ตัวคือ PFET1-PFET3 โดย NFET1 จะทำงานเมื่อได้รับลอจิก "1" จากขา Q และหยุดทำงานเมื่อได้รับลอจิก "0" วงจรพูลอัปจะเริ่มต้นทำงานเมื่อ NFET1 ได้รับลอจิก "1" PFET1 จะทำงานนานประมาณ 2 คาบเวลาของสัญญาณนาฬิกาภายใน หลังจากที่เกิดการเปลี่ยนแปลงจากลอจิก "0" เป็นลอจิก "1" ในขณะที่ PFET1 ทำงาน จะทำให้ PFET3 ทำงานตามไปด้วย ทำให้เกิดการพูลอัปขาพอร์ต

รูปที่ 2.5 วงจรพูลอัปภายในพอร์ตของไมโครคอนโทรลเลอร์ MCS-51

ในรูปที่ 2.4(ค) เป็นวงจรภายในของพอร์ต 2 จะคล้ายกับพอร์ต 0 มากต่างกันเพียงมีวงจรพูลอัปเพิ่มเติมเข้ามา ส่วนในรูปที่ 2.4(ง) เป็นวงจรภายในของพอร์ต 3 จะเห็นได้ว่าคล้ายกับพอร์ต 1 มีการเพิ่มเติมวงจรบัฟเฟอร์ และวงจรอินพุตเอาต์พุตเมื่อทำงานในฟังชั่นพิเศษเข้ามา เนื่องจากพอร์ต 3 สามารถนำไปใช้งานในหน้าที่พิเศษได้ทุกขา

2.5 การใช้งานเป็นพอร์ตอินพุต

เนื่องจากพอร์ตทั้งหมดของไมโครคอนโทรลเลอร์ MCS – 51 แบบแฟลชสามารถเป็นได้ทั้งอินพุตและเอาต์พุต ดังนั้นจึงมีความจำเป็นอย่างยิ่งต้องทำความเข้าใจถึงการกำหนดลักษณะการทำงานให้แก่พอร์ตของไมโครคอนโทรลเลอร์ MCS – 51 แบบแฟลช

ในการกำหนดให้เป็นพอร์ตอินพุต ต้องเริ่มต้นด้วยการเขียนข้อมูล “1” มาที่แต่ละบิตของพอร์ตที่ต้องการใช้งานเป็นอินพุต เพื่อหยุดการทำงานของเฟตที่ใช้ในการขับสัญญาณเอาต์พุตของบิตนั้น ๆ ทำให้ขาสัญญาณของพอร์ตเชื่อมต่อเข้ากับวงจรพูลอัปภายในโดยตรง ส่งผลให้ขาพอร์ตนั้นมีลอจิกเป็น “1” สามารถรับสัญญาณลอจิก “0” จากอุปกรณ์ภายนอกได้ง่าย สัญญาณข้อมูลจากอุปกรณ์ภายนอกจะถูกส่งเข้ามาแล้วเก็บไว้ในวงจรบัฟเฟอร์ภายในพอร์ต แล้วรอให้ซีพียูมาอ่านค่า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เข้าไป เมื่อเป็นเช่นนี้ อุปกรณ์ภายนอกที่เชื่อมต่อกับพอร์ตอินพุตของไมโครคอนโทรลเลอร์ MCS – 51 แบบแฟลชควรกำหนดให้ทำงานในสภาวะลอจิก “0” จะดีและสะดวกที่สุด (ซึ่งในปัจจุบันอุปกรณ์อินพุตที่เชื่อมต่อไมโครคอนโทรลเลอร์แทบทั้งหมดทำงานที่ลอจิก “0” แล้ว)

2.6 การใช้งานเป็นพอร์ตเอาต์พุต

โดยปกติแล้ว ขาพอร์ตจะกำหนดให้มีลักษณะเป็นเอาต์พุตอยู่แล้ว ดังนั้นจึงสามารถส่งข้อมูลออกไปได้อย่างง่ายดายและตรงไปตรงมากล่าวคือ เมื่อต้องการส่งข้อมูล “0” ออกไปทางเอาต์พุต ก็ให้เขียนข้อมูล “0” ไปยังวงจรแลตช์ ซึ่งก็จะส่งต่อไปขับเฟต ทำให้เฟตทำงาน ที่ขาพอร์ตที่กำหนดให้ทำงานก็จะเกิดลอจิก “0” ขึ้น ในทางตรงข้ามหากต้องการส่งข้อมูล “1” ออกไป ก็ให้เขียนข้อมูล “1” ไปยังวงจรแลตช์ วงจรขับก็จะหยุดทำงาน ทำให้ที่ขาพอร์ตเชื่อมต่อกับวงจรพูลอัปภายในเกิดเป็นลอจิก “1” ที่ขาพอร์ตนั้น ซึ่งจะคล้ายกับการกำหนดให้เป็นขาอินพุตจะมีสัญญาณมาอ่านข้อมูลที่บัฟเฟอร์ แต่ถ้าเป็นเอาต์พุตจะไม่มีการอ่านข้อมูลที่บัฟเฟอร์แต่อย่างใด เว้นแต่ในกรณีที่ต้องการตรวจสอบข้อมูลที่ส่งออกมาทางเอาต์พุต

เมื่อใช้งานพอร์ตของไมโครคอนโทรลเลอร์ MCS - 51 แบบแฟลชเป็นพอร์ตเอาต์พุต แต่ละขา (หรือแต่ละบิต) ของแต่ละพอร์ตมีความสามารถในการจ่ายกระแสหรือที่เรียกว่า กระแสซอร์ส (source current) ได้สูงสุด 10 mA และทุกขารวมกันในแต่ละพอร์ต (ทั้ง 8 บิต) สูงสุด 26 mA สำหรับพอร์ต 0 และ 15 mA สำหรับพอร์ต 1 – 3 ในกรณีที่ใช้งานทุกพอร์ตเอาต์พุตจะสามารถจ่ายกระแสได้รวมกันสูงสุด 71 mA ดังนั้นในการใช้งานเป็นพอร์ตเอาต์พุตเพื่อไม่ให้เกิดปัญหาเกี่ยวกับความสามารถในการจ่ายกระแสจึงควรต่อวงจรบัฟเฟอร์ทางเอาต์พุตเพื่อช่วยในการขับกระแสอีกทางหนึ่ง

2.7 หน่วยความจำโปรแกรม (Program memory)

ในรูปที่ 2.6 แสดงการจัดหน่วยความจำโปรแกรมของไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลชในเบอร์ต่างๆ ที่นิยมใช้งาน อันประกอบด้วยเบอร์ AT89C51 และ AT89C52 จะเห็นได้ว่า ทั้งสองเบอร์สามารถติดต่อหน่วยความจำโปรแกรมได้สูงสุด 64 กิโลไบต์โดยสามารถเลือกใช้หน่วยความจำโปรแกรมภายในอย่างเดียวหรือรวมกับภายนอกหรือเลือกใช้หน่วยความจำภายนอกอย่างเดียวก็ได้ดังในรูปที่ 2.6(ก)โดยภายใน AT89C51 จะมีหน่วยความจำโปรแกรมภายใน 4 กิโลไบต์ ในขณะที่ AT89C52 จะมีขนาด 8 กิโลไบต์

ในกรณีที่ใช้หน่วยความจำภายในและภายนอกรวมกัน หากใช้ AT89C51 ก็จะสามารถติดต่อกับหน่วยความจำภายนอกได้ 60 กิโลไบต์ และถ้าใช้เบอร์ AT89C52 จะสามารถติดต่อกับหน่วยความจำโปรแกรมภายนอกได้ 56 กิโลไบต์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หน่วยความจำโปรแกรมใช้เก็บข้อมูลของโปรแกรมควบคุมการทำงานของไมโครคอนโทรลเลอร์หรือที่เรียกว่า โปรแกรมมอนิเตอร์(monitor program) หากใช้หน่วยความจำภายนอกมักจะบรรจุอยู่ในหน่วยความจำชนิดอีพรอม(EPROM : Erasable Programmable Read-Only Memory) ซึ่งสามารถกระทำการอ่านได้เพียงอย่างเดียว

(ก) การจัดสรรหน่วยความจำโปรแกรมของไมโครคอนโทรลเลอร์เบอร์ AT89C51

(ข) การจัดสรรหน่วยความจำโปรแกรมของไมโครคอนโทรลเลอร์เบอร์ AT89C52

รูปที่ 2.6 การจัดสรรหน่วยความจำโปรแกรมของไมโครคอนโทรลเลอร์ MCS-51

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หน่วยความจำโปรแกรมมีแอดเดรสเริ่มต้นที่ 0000H เมื่อซีพียูได้รับการรีเซตให้เริ่มต้นการทำงานจะต้องเริ่มต้นที่แอดเดรส 0000H นี้เสมอ อย่างไรก็ตาม ในพื้นที่ของหน่วยความจำโปรแกรมไม่ว่าจะใช้งานภายในหรือภายนอกก็ตามต้องมีการสงวนพื้นที่บางตำแหน่งเอาไว้สำหรับการบริการอินเตอร์รัปต์ 6 ประเภท ประเภทละ 8 ไบต์ ประกอบด้วย

พื้นที่สำหรับบริการอินเตอร์รัปต์ 0 จากภายนอก กำหนดไว้ที่แอดเดรส 0003H

พื้นที่สำหรับบริการอินเตอร์รัปต์จากไทเมอร์ 0 กำหนดไว้ที่แอดเดรส 000BH

พื้นที่สำหรับบริการอินเตอร์รัปต์ 1 จากภายนอก กำหนดไว้ที่แอดเดรส 0013H

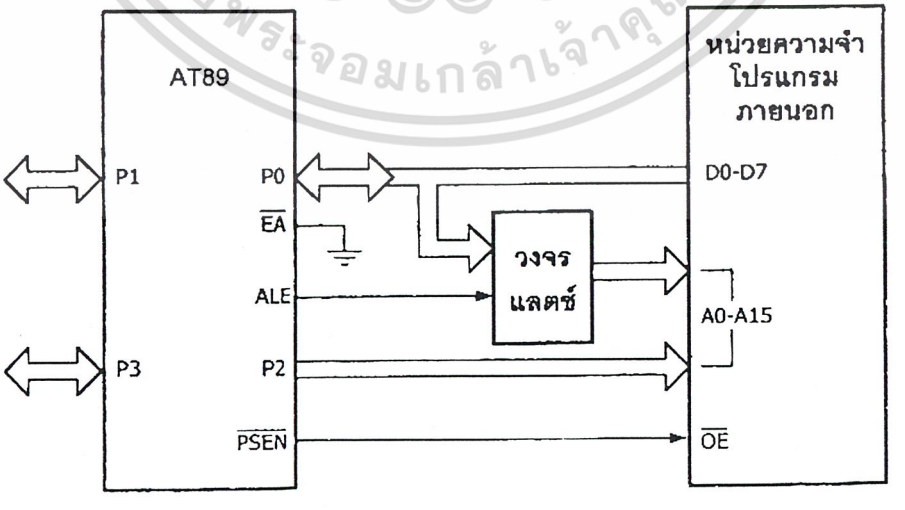
พื้นที่สำหรับบริการอินเตอร์รัปต์จากไทเมอร์ 1 กำหนดไว้ที่แอดเดรส 001BH

พื้นที่สำหรับบริการอินเตอร์รัปต์ของการสื่อสารอนุกรม กำหนดไว้ที่แอดเดรส 0023H

พื้นที่สำหรับบริการอินเตอร์รัปต์จากไทเมอร์ 2 กำหนดไว้ที่แอดเดรส 002BH

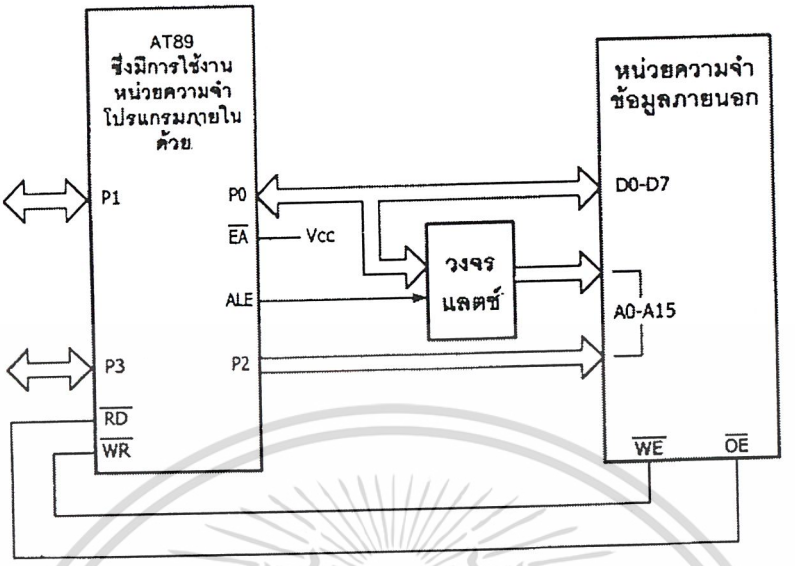
กรณีที่ใช้ไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลชที่มีหน่วยความจำโปรแกรมภายในแต่ต้องการติดต่อกับหน่วยความจำโปรแกรมภายนอกด้วย สามารถทำได้โดยต้องกำหนดแอดเดรสของหน่วยความจำโปรแกรมให้ต่อจากแอดเดรสสุดท้ายของหน่วยความจำโปรแกรมภายในของไมโครคอนโทรลเลอร์ ยกตัวอย่าง ไมโครคอนโทรลเลอร์ AT89C51 มีหน่วยความจำโปรแกรมขนาด 4 กิโลไบต์ มีแอดเดรสอยู่ระหว่าง 0000H-0FFFH เมื่อต่อหน่วยความจำโปรแกรมภายนอกต้องกำหนดให้แอดเดรสอยู่ในช่วง 1000H-FFFFH

การต่อหน่วยความจำภายนอกแสดงดังรูปที่ 2.7 จะเห็นได้ว่าขาพอร์ต P0.0-P0.7 ถูกใช้เป็นขาข้อมูล D0-D7 และขาแอดเดรสไบต์ต่ำ โดยผ่านวงจรแลตช์ ซึ่งปกติใช้ไอซีเบอร์ 74HC573



รูปที่ 2.7 การเชื่อมต่อหน่วยความจำโปรแกรมภายนอกของ MCS-51

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.8 การเชื่อมต่อหน่วยความจำข้อมูลภายนอกของ MCS-51

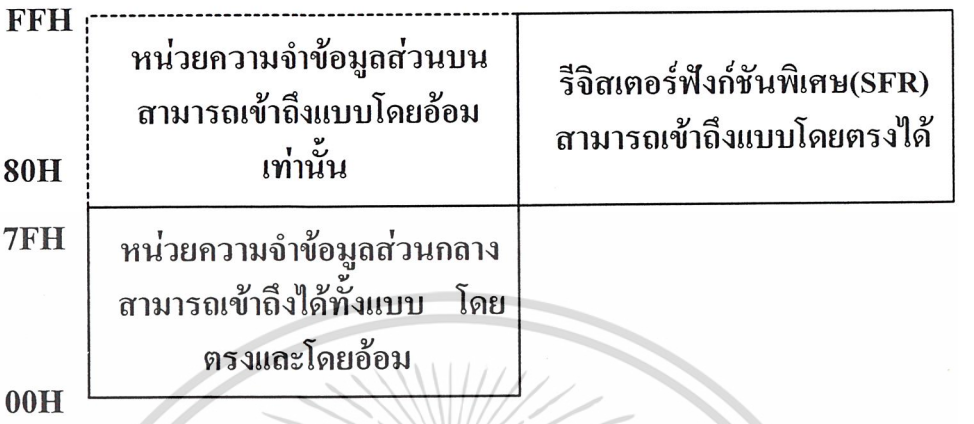
และใช้สัญญาณ ALE และ $\overline{\text{PSEN}}$ ในการเลือกว่า ต้องการใช้ขา P0.0-P0.7 เพื่อเป็นขาข้อมูลหรือขาแอดเดรสในขณะที่ขา P2.0-P2.7 ใช้ในการเชื่อมต่อกับขาแอดเดรสไบต์สูง A8-A15 ดังนั้นเมื่อมีการติดต่อกับหน่วยความจำโปรแกรมภายนอก ไมโครคอนโทรลเลอร์จะเหลือขาพอร์ตใช้งานเพียง 16 บิต คือที่ขาพอร์ต P1.0- P1.7 และ P3.0-P3.7

2.8 หน่วยความจำข้อมูล (Data memory)

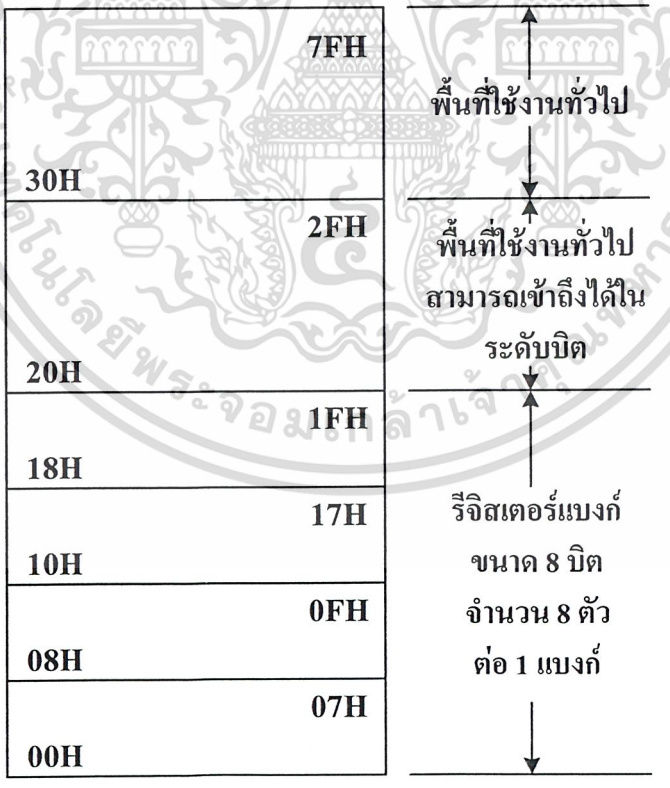
มีด้วยกัน 2 แบบคือ หน่วยความจำข้อมูลภายนอกและภายในโดยไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลชในอนุกรม AT89 สามารถติดต่อกับหน่วยความจำข้อมูลภายนอกได้สูงสุด 64 กิโลไบต์ โดยการใช้คำสั่ง MOVX ในการติดต่อกับหน่วยความจำข้อมูลภายนอก การติดต่อกับหน่วยความจำข้อมูลภายนอกของไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลชแสดงดังในรูปที่ 2.8 จะเห็นได้ว่า มีลักษณะคล้ายกับการติดต่อหน่วยความจำโปรแกรมภายนอกแตกต่างกันที่มีสัญญาณที่ใช้สำหรับการอ่านและเขียนหน่วยความจำข้อมูลภายนอก นั่นคือ ขา $\overline{\text{RD}}$ และ $\overline{\text{WR}}$

สำหรับไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลชในอนุกรม AT89 ทุกเบอร์จะมีหน่วยความจำข้อมูลภายในเป็นแบบแรม (RAM : Random Access Memory) โดยแต่ละเบอร์จะมีขนาดแตกต่างกันออกไป ในเบอร์ AT89C51 มีหน่วยความจำข้อมูลภายในขนาด 128 ไบต์ ในขณะที่เบอร์ AT89C52 มีขนาด 256 ไบต์ สำหรับการจัดสรรหน่วยความจำข้อมูลภายในแบ่งเป็น 3 ส่วน คือ หน่วยความจำข้อมูลส่วนล่าง (lower), ส่วนบน (upper) และรีจีสเตอร์ฟังชั่นพิเศษ (SFR : Special Function Register) แต่ละส่วนมีขนาด 128 ไบต์ ดังแสดงจัดสรรในรูปที่ 2.9

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.9 การจัดสรรพื้นที่ของหน่วยความจำข้อมูลภายใน MCS-51



รูปที่ 3.0 การจัดสรรพื้นที่ของหน่วยความจำข้อมูลภายในส่วนล่างของ MCS-51

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จะเห็นว่า หน่วยความจำข้อมูลส่วนบนและรีจิสเตอร์ฟังชั่นพิเศษมีตำแหน่งทับซ้อนกัน แต่จะใช้การติดต่อที่แตกต่างกัน และในไมโครคอนโทรลเลอร์ MCS-51 บางเบอร์จะไม่มีหน่วยความจำข้อมูลส่วนบน

ขนาดของหน่วยความจำข้อมูลของไมโครคอนโทรลเลอร์MCS-51 แบบแฟลชโดยแท้จริงแล้วมีเพียง 256 ไบต์ แต่ด้วยการจัดการเข้าถึงที่แตกต่างกัน จึงดูเสมือนว่า ไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลชมีหน่วยความจำข้อมูลภายในสูงถึง 384 ไบต์ โดยในหน่วยความจำข้อมูลส่วนล่างขนาด 128 ไบต์ มีแอดเดรสอยู่ที่ 00H-7FH สามารถเข้าถึงได้โดยตรงและโดยอ้อม สำหรับหน่วยความจำข้อมูลส่วนบนมีขนาด 128 ไบต์เช่นกัน มีแอดเดรสอยู่ที่ 80H-FFH สามารถเข้าถึงแบบโดยอ้อมเท่านั้น ในขณะที่รีจิสเตอร์ SFR มีแอดเดรสอยู่ที่ 80H-FFH เช่นเดียวกับหน่วยความจำข้อมูลส่วนบน แต่สำหรับรีจิสเตอร์ SFR ใช้การเข้าถึงแบบโดยตรง

ดังนั้นเพื่อความสะดวกและง่ายตลอดจนป้องกันความสับสนในการเขียนโปรแกรมสำหรับผู้เริ่มต้นจึงควรใช้หน่วยความจำข้อมูลภายในเพียง 128 ไบต์จากหน่วยความจำข้อมูลส่วนล่างร่วมกับรีจิสเตอร์ SFR

ในรูปที่ 3.0 แสดงการจัดสรรหน่วยความจำข้อมูลส่วนล่าง หน่วยความจำ 32 ไบต์ต่ำสุดที่แอดเดรส 00H-1FH แบ่งเป็น 4 กลุ่ม เรียกว่า 4 แบงก์ (bank) แต่ละแบงก์มีรีจิสเตอร์ 8 ตัวคือ R0-R7 การติดต่อกับหน่วยความจำในแบงก์ใดให้กำหนดที่รีจิสเตอร์ PWS (Program Status Word register)

หน่วยความจำข้อมูล 16 ไบต์ถัดมาที่แอดเดรส 20H-2FH เป็นพื้นที่สำหรับใช้งานทั่วไป สามารถเข้าถึงได้ในระดับบิต (Bit addressable)และหน่วยความจำข้อมูลที่เหลือ 80 ไบต์ จะต้องแบ่งส่วนหนึ่งสำรองไว้เป็นสแต็ก (stack : ที่พักข้อมูลชั่วคราวในกรณีที่ซีพียูมีการกระโดดไปทำงานในโปรแกรมย่อย) การเข้าถึงหน่วยความจำข้อมูลส่วนนี้ต้องใช้การเข้าถึงในระดับไบต์

2.9 การสื่อสารผ่านพอร์ตอนุกรมของไมโครคอนโทรลเลอร์ MCS – 51

การสื่อสารข้อมูลผ่านพอร์ตอนุกรมในไมโครคอนโทรลเลอร์นั้น ทำได้ 2 วิธี คือ

1.ใช้อินเตอร์รัปต์ เป็นวิธีที่ให้ผลการทำงานเร็วที่สุด แต่มีความยุ่งยากในการทำงานมากกว่าเนื่องจากตำแหน่งของการอินเตอร์รัปต์ทั้งการรับและการส่งข้อมูลนั้นอยู่ที่ตำแหน่งเดียวกัน ต้องพิจารณาจากแฟลก TI หรือ RI ก่อนว่าเกิดการอินเตอร์รัปต์จากสาเหตุใด และต้องพิจารณาการใช้รีจิสเตอร์ในช่วงเวลานั้น ๆ ด้วยว่ามีโอกาสซ้อนทับกันหรือไม่ ทำให้โปรแกรมของการทำงานในส่วนนี้มีความซับซ้อนมากกว่า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

2. วนโปรแกรมตรวจสอบแฟลก เป็นวิธีที่มีความซับซ้อนน้อยกว่า โดยเขียนโปรแกรมให้วนตรวจสอบแฟลกอยู่ตลอดเวลาจนกว่าจะเกิดการเปลี่ยนแปลง ยกตัวอย่าง เมื่อต้องการตรวจสอบการส่งข้อมูล ให้ทำการวนตรวจสอบแฟลก TI ว่าถูกเซตหรือไม่ เมื่อถูกเซต แสดงว่า มีการส่งข้อมูลเกิดขึ้นเรียบร้อยแล้ว จากนั้นให้ทำการเคลียร์แฟลก TI แล้วทำการส่งข้อมูลตัวถัดไป หรือทำงานในคำสั่งต่อไปได้

ในกรณีที่ต้องการตรวจสอบการรับข้อมูล ให้ทำการตรวจสอบแฟลก RI ว่าถูกเซตหรือไม่ เมื่อตรวจสอบได้ว่าถูกเซต แสดงว่า เกิดการรับข้อมูลขึ้น ให้ทำการเคลียร์แฟลก RI แล้วนำค่าจากรีจิสเตอร์ SBUF มาใช้ได้ทันที แต่วิธีการนี้มีข้อเสียตรงที่เป็นการทำงานแบบเรียงลำดับ ทำให้ขั้นตอนในการทำงานช้ากว่าการทำงานด้วยอินเตอร์รัปต์

อย่างไรก็ตามหัวใจสำคัญของการสื่อสารข้อมูลผ่านพอร์ตอนุกรมคือ การกำหนดอัตราบอดและรูปแบบของข้อมูลว่า มีจำนวนบิตเริ่มต้น, บิตของข้อมูล, บิตหยุด หรือว่ามีการตรวจสอบบิตพารีตี้หรือไม่ถ้าหากข้อกำหนดเหล่านี้ในตัวส่งและตัวรับไม่ตรงกันจะทำให้การถ่ายทอดข้อมูลเกิดความผิดพลาดได้อย่างง่ายดาย ส่งผลให้การสื่อสารข้อมูลล้มเหลวอย่างสิ้นเชิง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

46540

บทที่ 3 รีลไทม์คล็อก (RTC)

3.1 ไอซีสร้างฐานเวลาจริงหรือรีลไทม์คล็อก (RTC)

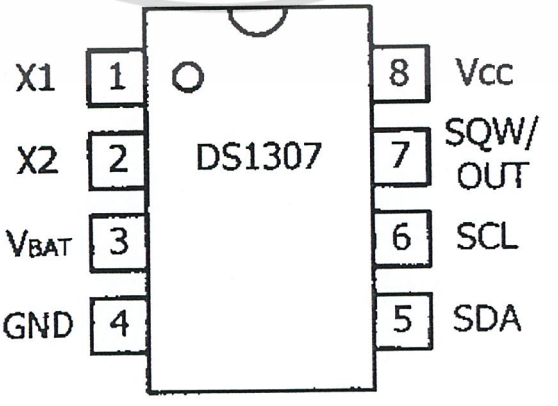
มีหน้าที่สร้างฐานเวลาจริงให้แก่ระบบไมโครคอนโทรลเลอร์ โดย DS1307 จะให้ข้อมูลเกี่ยวกับเวลาทั้งหมด ไม่ว่าจะเป็นค่าของเวลาที่ละเอียดถึงหลักวินาที, นาที, ชั่วโมง, วันที่, วันในสัปดาห์, เดือนและปี โดยสามารถปรับวันเดือนปีให้ตรงตามปฏิทินได้อย่างถูกต้อง รวมถึงการกำหนดวันในปีอธิกสุรทินด้วย คุณสมบัติทางเทคนิคที่สำคัญมีดังนี้

- เป็นไอซีรีลไทม์คล็อกให้ข้อมูลตั้งแต่วินาทีจนถึงปี รวมถึงการกำหนดวันในปีอธิกสุรทินด้วย สามารถให้ข้อมูลเวลาได้อย่างเที่ยงตรงถึงปี ค.ศ. 2100
- มีหน่วยความจำนอนโวลาไทล์แรม 56 ไบต์อยู่ภายใน สามารถใช้เก็บข้อมูลทั่วไปได้
- ใช้การเชื่อมต่อแบบระบบบัส I^2C
- มีวงจรตรวจจับไฟเลี้ยงต่ำหรือหายไปอย่างอัตโนมัติ และสามารถรักษาข้อมูลเวลาไว้ได้แม้ไม่มีไฟเลี้ยงไอซี

3.2 รายละเอียดขาต่อการใช้งานของ DS1307

Vcc, GND (ขา 8,4) ต่อกับไฟเลี้ยง +5V

V_{BAT} (ขา 3) ใช้ต่อกับแบตเตอรี่ 3V เพื่อรักษาการทำงานของวงจรสร้างฐานเวลาของ DS1307 ให้คงอยู่ต่อไป แม้ว่าไม่มีไฟเลี้ยงจ่ายให้แก่ DS1307 ชนิดของแบตเตอรี่ที่เหมาะสมคือแบตเตอรี่แบบลิเธียม ซึ่งมีความจุ 40 mAhr หรือมากกว่า จะสามารถรักษาข้อมูลได้นาน 10 ปีที่อุณหภูมิ 25 องศาเซลเซียส



รูปที่ 3.1 การจัดขาของไอซี DS1307

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

SDA, SCL (ขา 5 และ 6) เป็นขาสำหรับเชื่อมต่อกับไมโครคอนโทรลเลอร์บนระบบบัส I^2C

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

X1, X2 (ขา 1 และ 2) ใช้ต่อกับคริสตอลความถี่มาตรฐาน 32.768kHz เพื่อใช้เป็นฐานเวลาในการสร้างค่าเวลาจริง ในการใช้งานต้องต่อคริสตอลเข้ากับขาทั้งสองนี้และที่แต่ละขาต้องมีตัวเก็บประจุค่าต่ำๆประมาณ 15pF คร่อมกับขากราวด์ด้วย

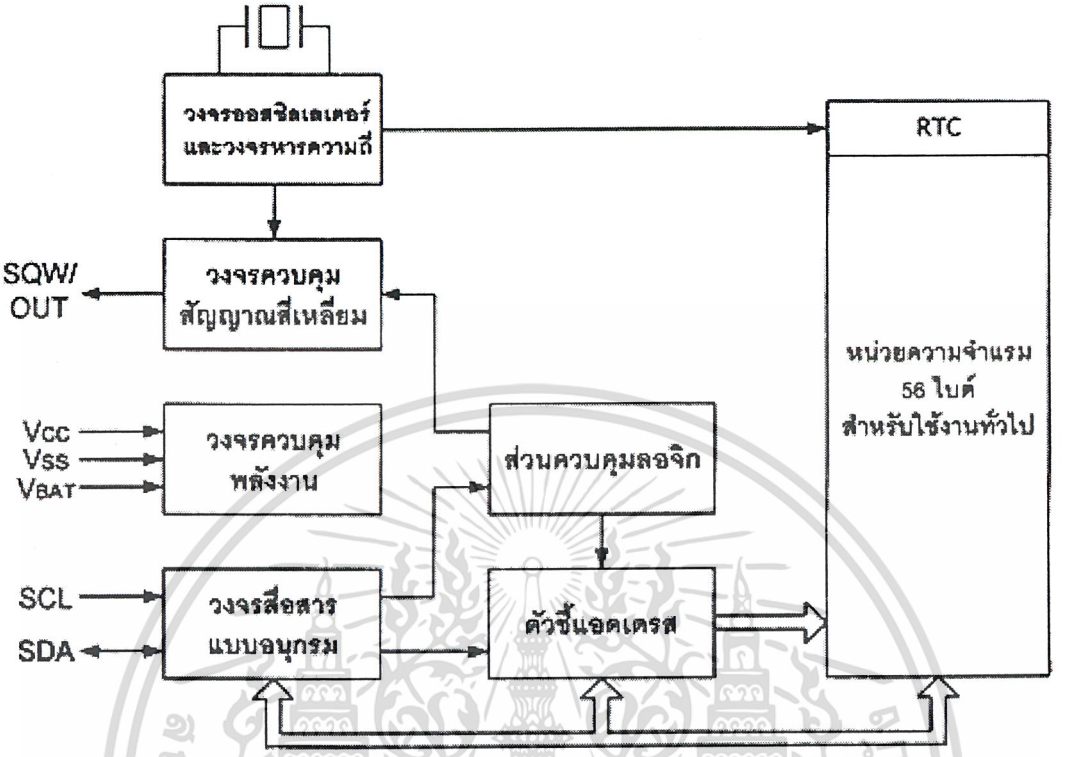
3.3 การทำงานของ DS1307

ไอซี DS1307 จัดการเชื่อมต่อในแบบบัส I^2C โดยจะทำงานเป็นอุปกรณ์สเลฟเสมอ ดังนั้นการติดต่อเพื่อใช้งานจึงต้องกำหนดรูปแบบตามที่กำหนดไว้ในการติดต่อแบบ I^2C ในรูปที่ 3.2 แสดงส่วนประกอบหลักที่สำคัญและไดอะแกรมการทำงานของ DS1307 วงจรออสซิลเลเตอร์ถือเป็นหัวใจหลักของไอซี เนื่องจากเป็นจุดเริ่มต้นของการสร้างข้อมูลเวลาจริง ในขณะที่ DS1307 ทำงานที่ขา SQW/OUT จะมีสัญญาณพัลส์สี่เหลี่ยมส่งออกมาตลอดเวลาที่มีการอีนาเบิลวงจรกำเนิดสัญญาณพัลส์ที่รีจิสเตอร์ควบคุม ค่าความถี่ของสัญญาณนี้สามารถเลือกได้ 4 ค่าคือ 1Hz, 4.096kHz, 8.192kHz และ 32kHz พร้อมกันนั้นก็จะมีการเก็บค่าของเวลาไว้ในหน่วยความจำนอนโวลาไทล์แรม ซึ่งมีขนาดรวม 64 ไบต์ แต่จัดสรรให้ใช้เก็บข้อมูลเวลา 8 ไบต์ และเป็นหน่วยความจำสำหรับเก็บข้อมูลทั่วไปสำหรับผู้ใช้งาน 56 ไบต์

วงจรควบคุมพลังงานไฟฟ้าจะคอยตรวจสอบสภาวะของไฟเลี้ยงไอซี หากไฟเลี้ยงต่ำกว่า $1.25 \times V_{BAT}$ ก็จะควบคุมให้ DS1307 หยุดการทำงานรีเซตค่าตัวนับแอดเดรสภายในทำให้ไม่สามารถติดต่อกับ DS1307 ได้ดังนั้นในการใช้งานต้องระมัดระวังอย่าให้ไฟเลี้ยงต่ำกว่า $1.25 \times V_{BAT}$ หรือ 3.75 V ในกรณีที่ใช้ V_{BAT} เท่ากับ 3 V ถ้าหากไฟเลี้ยงมีค่าต่ำกว่า V_{BAT} ไอซี DS1307 จะเข้าสู่โหมดสำรองข้อมูลกระแสต่ำทันที จะไม่มีการส่งสัญญาณพัลส์ออกมาที่ขา SQW/OUT แต่วงจรสร้างฐานเวลายังคงทำงานเพื่อให้ค่าของเวลาเดินไปอย่างไม่ผิดพลาด เมื่อมีไฟเลี้ยงปรากฏขึ้นอีกครั้ง DS1307 ก็จะสามารถให้ค่าของเวลาที่เป็นจริงแก่ผู้ใช้งานได้ต่อไป

วงจรสื่อสารอนุกรมภายใน DS1307 ได้รับการกำหนดให้ทำงานตามรูปแบบของบัส I^2C เป็นช่องทางการสื่อสารระหว่าง DS1307 กับอุปกรณ์มาสเตอร์ ผู้ใช้งานสามารถเข้าถึงหน่วยความจำที่ใช้เก็บค่าเวลาและหน่วยความจำใช้งานทั่วไปได้โดยการเขียนข้อมูลตามรูปแบบที่กำหนดในระบบบัส I^2C

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.2 โครงสร้างภายในของไอซี DS1307

3.4 โหมดการทำงานของ DS1307

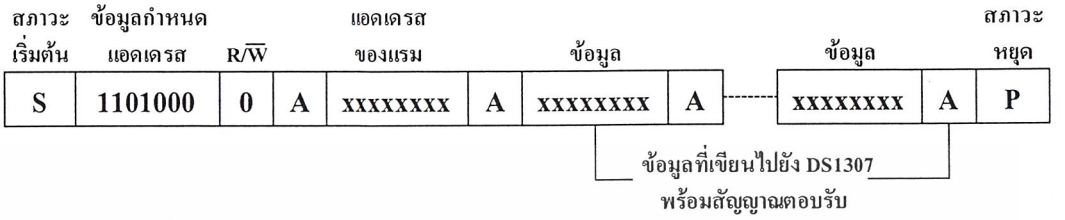
มีด้วยกัน 2 โหมดคือ โหมดเขียนข้อมูลและโหมดอ่านค่าข้อมูล ในการใช้งาน DS1307 ตามปกติจะใช้งานเฉพาะโหมดอ่านข้อมูลเท่านั้น เนื่องจากไมโครคอนโทรลเลอร์จะติดต่อกับ DS1307 เพื่ออ่านข้อมูลของเวลาไปใช้งาน โหมดการเขียนข้อมูลจะถูกใช้งานก็ต่อเมื่อต้องการตั้งค่าเวลาใหม่และต้องการเขียนข้อมูลลงในหน่วยความจำที่ใช้งานทั่วไป อย่างไรก็ตามเมื่อเริ่มติดต่อกับ DS1307 จำเป็นอย่างยิ่งที่จะต้องเข้าสู่โหมดการเขียนข้อมูลก่อนเพื่อกำหนดแอดเดรสที่ต้องการอ่านข้อมูล จากนั้นจึงเปลี่ยนโหมดการทำงานมาเป็นโหมดการอ่านข้อมูลต่อไป

3.4.1โหมดการเขียนข้อมูล

มีรูปแบบดังรูป 3.3 เริ่มต้นเมื่อไมโครคอนโทรลเลอร์ ทำการกำหนดสภาวะเริ่มต้น (START : S) จากนั้นส่งข้อมูลกำหนดแอดเดรส 1101000 ตามด้วยข้อมูลเลือกการเขียนคือค่า 0 จากนั้นจะรอการตอบรับจาก DS1307 ขั้นตอนต่อมาคือ ส่งข้อมูลเพื่อเลือกแอดเดรสที่ต้องการเขียน จากนั้นรอการตอบรับจาก DS1307 เมื่อมีการตอบรับมาเรียบร้อย ก็เริ่มทยอยเขียนข้อมูลลงไปครั้งละแอดเดรสหลังจากเขียนข้อมูลในแต่ละแอดเดรส จะต้องหยุดรอการตอบรับจาก DS1307 ทุกครั้ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

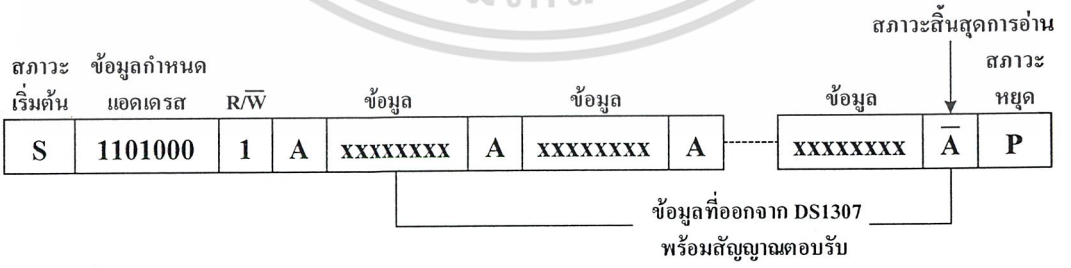
จึงจะสามารถเขียนข้อมูลต่อไปได้เมื่อเขียนข้อมูลเรียบร้อยแล้วให้ส่งสภาวะหยุด (STOP : P) เป็นอันสิ้นสุดกระบวนการเขียนข้อมูล



รูปที่ 3.3 รูปแบบของข้อมูลสำหรับติดต่อกับ DS1307 ในโหมดการเขียนข้อมูล

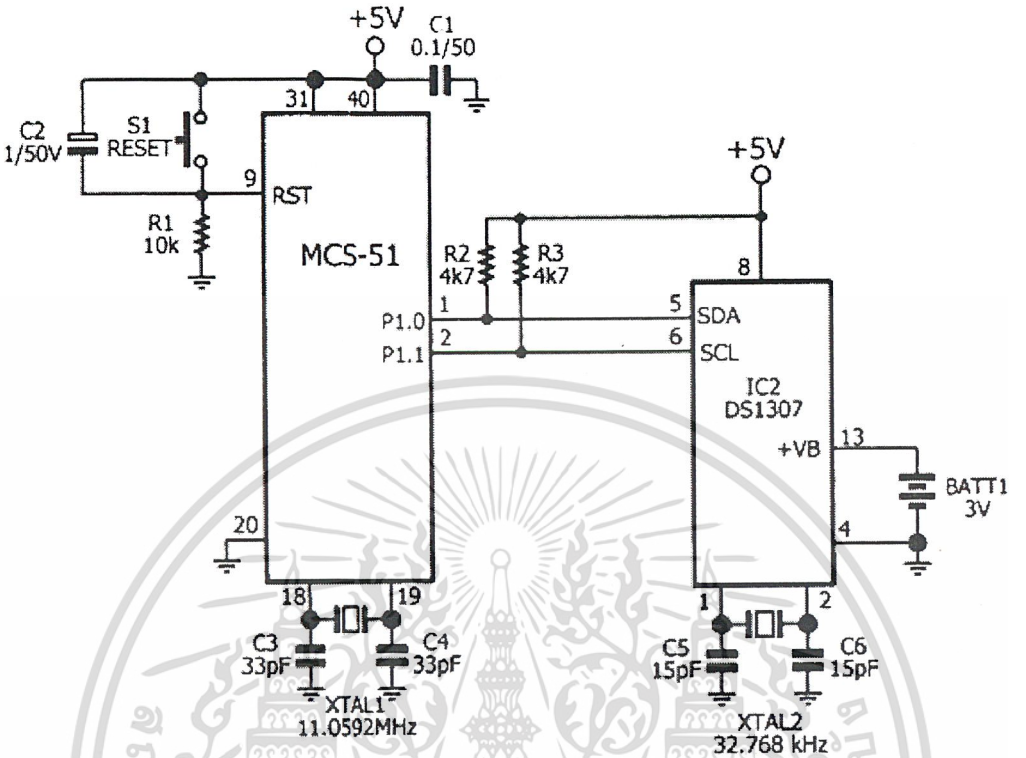
3.4.2โหมดการอ่านข้อมูล

มีรูปแบบแสดงดังรูป 3.4 เริ่มต้นการทำงานเหมือนกับโหมดการเขียนข้อมูลคือ ไมโครคอนโทรลเลอร์กำหนดสภาวะเริ่มต้นแล้วส่งข้อมูลกำหนดแอดเดรสตามด้วยข้อมูลเลือกการอ่านซึ่งเท่ากับ 1 จากนั้นรอการตอบรับจาก DS1307 เมื่อตอบรับเรียบร้อยแล้ว DS1307 จะทยอยส่งข้อมูลออกมาให้ไมโครคอนโทรลเลอร์คราวละ 1 แอดเดรสหรือ 1 ไบต์ โดยแอดเดรสที่เลือกอ่านข้อมูลจะต้องมีการกำหนดมาก่อนล่วงหน้าด้วยโหมดการเขียนข้อมูล วิธีการง่ายๆคือ เข้าสู่โหมดการเขียนข้อมูลก่อน เมื่อถึงจังหวะที่ต้องเขียนข้อมูล ให้ทำการสร้างสภาวะเริ่มต้นและส่งข้อมูลกำหนดแอดเดรสใหม่อีกครั้ง ตามด้วยเลือกโหมดการอ่านข้อมูล ข้อมูลที่ออกมาจาก DS1307 จะเป็นข้อมูลจากแอดเดรสที่กำหนดไว้ก่อนหน้านี้



รูปที่ 3.4 รูปแบบของข้อมูลสำหรับติดต่อกับ DS1307 ในโหมดการอ่านข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.5 การเชื่อมต่อไมโครคอนโทรลเลอร์กับไอซีรีลไทม์คล็อก DS1307

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4 หน่วยแสดงผลข้อมูล (LCD MODULE)

ปัจจุบัน LCD เป็นที่นิยมกันอย่างมาก สำหรับการแสดงผลในเครื่องมือเครื่องใช้ต่าง ๆ ทั้งนี้เนื่องจากมีความเหมาะสมด้วยประการทั้งปวง ทั้งในด้านการกินกระแสต่ำ สามารถแสดงผลเป็นตัวอักษร และตัวเลข หรือแสดงเป็นกราฟฟิคได้ (เฉพาะรุ่น) จะติดปัญหาก็คือในด้านวงจร ซึ่งมีระบบการทำงานที่ซับซ้อน และหาอุปกรณ์ได้ค่อนข้างยาก แต่ขณะนี้ผู้ผลิต LCD จะทำรุ่นที่เป็น LCD MODULE ออกมา คือเป็น MODULE ที่มีตัว LCD และวงจรควบคุมให้พร้อม (เรียกว่า LCM) ซึ่งทำให้ผู้สามารถต่อเข้ากับระบบไมโครได้ง่ายและสะดวกสำหรับการเขียนโปรแกรม รวมทั้งมีจำหน่ายกันอย่างกว้างขวาง และมีราคาที่เหมาะสม ทำให้ผู้ใช้งานทางด้านไมโครหันมาใช้แผงแสดงด้วย LCD MODULE กันมากขึ้น LCD MODULE มีอยู่มากมายหลายรุ่น และมีคุณสมบัติแตกต่างกันไป ซึ่งแบ่งได้เป็น 2 กลุ่มหลักคือ

1. DOT MATRIXจะแสดงผลเป็นตัวอักษรขนาด 5*8 DOT และมีจำนวนอักษรและบรรทัดแตกต่างกันไปในแต่ละรุ่น

2. GRAPHIC จะสามารถแสดงผลในแบบ BIT-MAP คือจะสร้างเป็นภาพใด ๆ ก็ได้ตามต้องการ

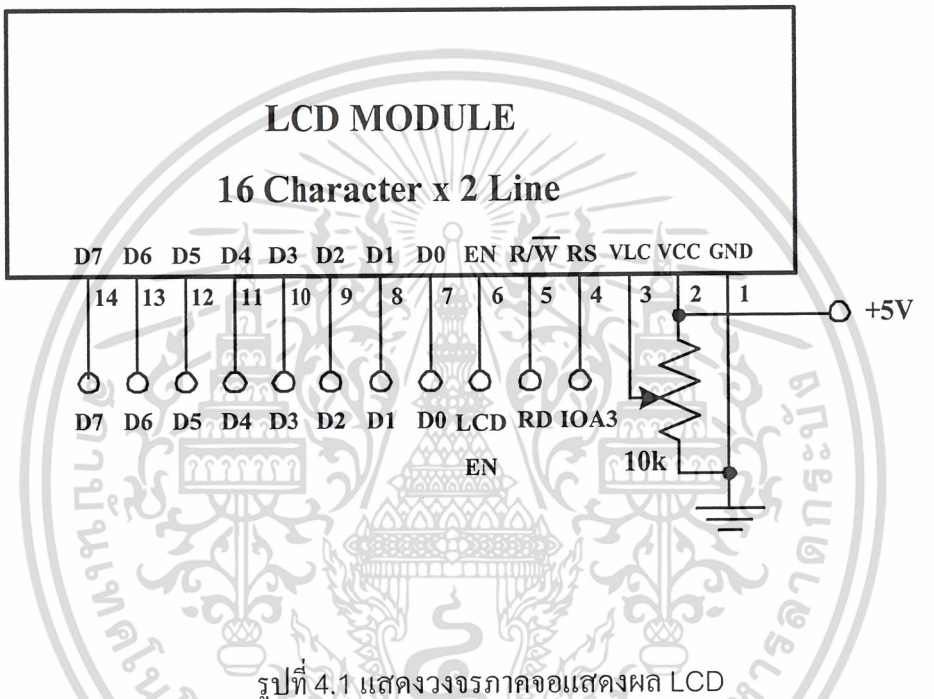
แนวทางในการใช้งานของทั้งสองแบบ จะมีลักษณะใกล้เคียงกัน การใช้งานโดยทั่วไปมักจะใช้แบบ DOT MATRIX มากกว่าเนื่องจากว่ามีราคาถูก และเพียงพอต่องานส่วนใหญ่ คุณสมบัติของ DOT MATRIX LCE MODULE สามารถสรุปเป็นข้อ ๆ ได้ดังนี้

1. มีเลือกหลายรุ่นตามการใช้งาน โดยมีจำนวนตัวอักษรและบรรทัดที่แตกต่างกันไป
2. ตัวอักษรแสดงด้วย DOT MATRIX ขนาด 5*8 DOT
3. สามารถต่อเข้ากันระบบไมโครได้ 2 ลักษณะ คือแบบ MEMORY MAP (20 PIN LCD BUS) และแบบผ่าน 8255 PORT (26 – PIN 8255 BUS โดยกรณี 26 – PIN 8255 BUS จะใช้แผ่น PCP (DMCAD) เป็นตัว ADAPTER ทำให้เป็น 8255 BUS อีกที
4. การใช้งานง่ายและสะดวก ระบบไมโครเพียงแค่ส่งข้อมูลให้กับ LCD MODULE เท่านั้น ข้อความก็จะปรากฏบนแผงแสดง และจะค้างไว้ตลอด ทำให้ไม่ต้องเสียเวลาหลักของระบบไมโคร
5. มีคำสั่งพิเศษสำหรับอำนวยความสะดวกมากมาย เช่น CLEAR DISPLAY,HOME RSOR,ON OFF CURSOR, BLINK CHARACTER และอื่น ๆ อีก
6. สามารถแสดงผลเป็นตัวอักษรภาษาอังกฤษและตัวเลขได้ 160 ตัว และสัญลักษณ์พิเศษอีก 32 ตัว รวมทั้งสามารถกำหนดอักษรที่ออกแบบเองได้อีก 8 ตัว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

7. กินกระแสน้อย และมีน้ำหนักเบา รวมทั้งทำงานด้วยไฟเลี้ยงระดับ 5 V เท่านั้น

การแสดงผลการทำงานของเครื่องควบคุมนี้จะใช้จอแสดงผล LCD ขนาด 16 ตัวอักษร 2 บรรทัดซึ่งสามารถแสดงข้อความตัวอักษรได้อย่างชัดเจนและทำให้ผู้ใช้งานทำความเข้าใจได้ง่าย การเชื่อมต่อจอแสดงผล LCD เข้ากับภาคของไมโครคอนโทรลเลอร์แสดงดังรูปที่ 4.1



รูปที่ 4.1 แสดงวงจรภาคจอแสดงผล LCD

จอแสดงผล LCD มีขาต่อใช้งานทั้งหมด 14 ขา แสดงการทำงานของแต่ละขาในตารางที่ 4.1 ตัวต้านทานปรับค่าได้ที่ต่อที่ขา 3 ของจอแสดงผล LCD ทำหน้าที่ในการปรับความเข้มในการแสดงผลและจากวงจรที่เชื่อมต่อขา IOA_3 เข้ากับขา RS (Register Select) ซึ่งทำหน้าที่ในการเลือกรีจิสเตอร์ที่ใช้รับคำสั่งและรีจิสเตอร์ที่ใช้รับข้อมูลการแสดงผล และขา EN ที่เชื่อมต่อเข้ากับขาสัญญาณ LCD EN จึงทำให้ตำแหน่งแอดเดรสของจอแสดงผล LCD ที่ใช้งานมีอยู่ 2 ตำแหน่งคือ ตำแหน่ง C001H และ C009H ซึ่งถ้าอ้างในตำแหน่งพอร์ต C001H จะเป็นการติดต่อเพื่อรับหรือส่งข้อมูลคำสั่งกับจอแสดงผล LCD แต่ถ้าอ้างในตำแหน่งพอร์ต C009H จะเป็นการติดต่อเพื่อรับหรือส่งข้อมูลกับจอแสดงผล LCD

สำหรับการเชื่อมต่อเพื่อรับหรือส่งข้อมูลใดๆ กับจอแสดงผล LCD จะใช้ขา D0-D7 (ขา 7-14) เชื่อมต่อกับข้อมูลของภาคไมโครคอนโทรลเลอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตำแหน่งขา	สัญลักษณ์	หน้าที่การทำงาน
1	Vss	เป็นกราวด์
2	Vdd	ต่อไฟตรง +5 โวลต์
3	Vo	ปรับความสว่าง (โดยปรับที่ VR1)
4	RS	ขา เลือกรีจิสเตอร์ ที่ต่อกับ IOA3 0 = ข้อมูลที่ได้รับเป็นคำสั่ง 1 = ข้อมูลที่ได้รับเป็นข้อมูล
5	R/W	เป็นการเลือกการอ่านหรือการเขียน 0 = การเขียนข้อมูล 1 = การอ่านข้อมูล
6	E	อีนาเบิลการอ่านหรือเขียน LCD
7	DB0	ดาต้าอินพุท/เอาท์พุตบิตที่ 1
8	DB1	ดาต้าอินพุท/เอาท์พุตบิตที่ 2
9	DB2	ดาต้าอินพุท/เอาท์พุตบิตที่ 3
10	DB3	ดาต้าอินพุท/เอาท์พุตบิตที่ 4
11	DB4	ดาต้าอินพุท/เอาท์พุตบิตที่ 5
12	DB5	ดาต้าอินพุท/เอาท์พุตบิตที่ 6
13	DB6	ดาต้าอินพุท/เอาท์พุตบิตที่ 7
14	DB7	ดาต้าอินพุท/เอาท์พุตบิตที่ 8

ตารางที่ 4.1 หน้าที่การทำงานของแต่ละขาของจอแสดงผล LCD

4.1 ชุดหรืออ่านเขียนข้อมูลและการแสดงข้อความ

การเขียนหรืออ่านข้อมูล LCD MODULE ก็คือ การกำหนดคุณสมบัติต่าง ๆ ในการใช้งานของชุดของ LCD ตามชุดคำสั่งควบคุมและรวมถึงการเขียนข้อมูลที่เป็นข้อความ เพื่อให้ปรากฏบนแผงแสดงด้วย โดยมีรายละเอียดดังต่อไปนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

INSTRUCTION	RS	R/W	DATA BIT								EXE. TIME ( )
			7	6	5	4	3	2	1	0	
CLEAR DISPLAY	0	0	0	0	0	0	0	0	0	1	1640
CURSOR AT HOME	0	0	0	0	0	0	0	0	1	*	1640
ENTRY MODE SET	0	0	0	0	0	0	0	1	I/D	S	40
DISPLAY ON/OFF	0	0	0	0	0	0	1	D	C	B	40
DISPLAY SHIFT	0	0	0	0	0	1	S/C	R/L	*	*	40
FUNCTION SET	0	0	0	0	1	DL	N	F	*	*	40
SET CGRAM ADD.	0	0	0	1	CGRAM ADDRESS						40
SET DDRAM ADD.	0	0	1	DDRAM ADDRESS							40
BUSY,ADD. READ	0	1	BF	ADDRESS							0
CGRAM,DDRAM WR	1	0	WRITE DATA								40
CGRAM,DDRAM RD	1	1	READ DATA								40

4.2 ความเข้าใจพื้นฐาน

1. การเขียนข้อมูลให้กับ LCD MODULE จะแบ่งเป็น 2 ลักษณะ คือ INSTRUCTION และ DATA โดยจะกำหนดด้วยขาสัญญาณ RS คือถ้า RS = 0 จะหมายถึงส่งสัญญาณควบคุม (INSTRUCTION) หรืออ่านค่า FLAG สภาพการทำงานของ LCD MODULE
2. หลักการในการเขียนข้อมูลให้ LCD MODULE นี้ คือเมื่อมีการเขียนข้อมูลเข้าไปแล้วตัว LCD MODULE จะต้องใช้เวลาในการทำงานชั่วขณะหนึ่ง (ตามค่า EXECUTE TIME ในตาราง) ซึ่งระบบไมโครสามารถตรวจสอบได้จาก BUSY FLAG (BF) และถ้าเรียบร้อยแล้วจึงสามารถเขียนข้อมูลอันต่อไปได้ ในกรณีที่การต่อวงจรเป็นแบบ I/O PORT คือไม่สามารถอ่านข้อมูลย้อนกลับได้ ระบบไมโครก็จะต้องใช้วิธีหน่วงเวลาแทน
3. การเขียนข้อมูลให้กับ LCD MODULE สามารถทำได้ทั้งแบบ 8 BIT และ 4 BIT โดยกรณี 4 BIT จะใช้สายสัญญาณ DATA เพียง 4 เส้น คือ DB4 – DB7(ใช้สำหรับระบบไมโครแบบ 4 BIT หรือเพื่อการประหยัดสาย) การเขียนข้อมูลจะกระทำเหมือนกับ 8 BIT เพียงแต่ให้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เขียน 2 ครั้งคือ DB4 – DB7 ก่อน แล้วตามด้วย DB0 – DB3 และจะต้องกำหนดคุณสมบัติตามค่า DL ในคำสั่ง FUNCTION SET ด้วย

4. DDRAM (DISPLAY DATA RAM) คือหน่วยความจำภายในตัว LCD MODULE ที่เป็น BUFFER ของข้อมูล โดยถ้าเขียนรหัส ASCII ใด ๆ ลงไปในหน่วยความจำนี้ ก็จะปรากฏเป็นตัวอักษรที่เป็นแผงแสดงทันที
5. CGRAM (CHARACTER GENERATOR RAM) คือหน่วยความจำภายในตัว LCD MODULE สำหรับเก็บภาพตัวอักษรที่ผู้ใช้สามารถสร้างได้เอง(8ตัว)โดยจะอ้าง ADDRESS ได้ทั้งหมด 64 BYTE คือ 8 ตัวอักษร คูณกับ 8 ROW

4.3 รายละเอียดของแต่ละคำสั่ง

1. CLEAR DISPLAY

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	0	0	0	0	0	0	1

สำหรับการ CLEAR DISPLAY โดยจะทำการเขียนตัวอักษร SPACE ลงใน DDRAM ทั้งหมด และกำหนดค่า DDRAM ADDRESS ให้เป็น 0 พร้อมทั้ง CURSOR จะกลับไปที่ตำแหน่งซ้ายสุดของจอภาพ

2. CURSOR AT HOME

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	0	0	0	0	0	1	*

สำหรับกำหนดค่า DDRAM ADDRESS ให้เป็น 0 พร้อมทั้ง CURSOR จะไปอยู่ที่ตำแหน่งซ้ายบนสุดของจอภาพ โดยที่ข้อมูลใน DDRAM ไม่มีการเปลี่ยนแปลง

3. ENTRY MODE SET

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	0	0	0	0	1	I/D	S

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

I/D = 0 กำหนดทิศทางของ CURSOR และ DDRAM ให้เป็นแบบ DECREMENT

I/D = 1 กำหนดทิศทางของ CURSOR และ DDRAM ให้เป็นแบบ INCREMENT

S = 0 เมื่อเขียนข้อมูลแล้ว ตัว CURSOR จะถูกไปทิศทางตามค่า I/D

S = 1 เมื่อเขียนข้อมูลแล้ว ตัว CURSOR จะอยู่กับที่และตัวอักษรจะถูกดันไปทิศทางตามค่า I/D

การกำหนด I/D และ S นี้ ให้กำหนดก่อนการเขียนข้อมูลใน DDRAM และเมื่อกำหนดแล้ว จะต้องไม่ใช้คำสั่ง CLEAR DISPLAY อีก

4. DISPLAY ON/OF

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	0	0	0	0	1	C	B

D = 0 กำหนดให้ OFF DISPLAY

D = 1 กำหนดให้ ON DISPLAY

C = 0 กำหนดให้ OFF CURSOR

C = 1 กำหนดให้ ON CURSOR โดย CURSOR จะเป็นเส้นขีดใต้ตัวอักษร

B = 0 กำหนดให้ไม่มีการกระพริบที่ตำแหน่ง CURSOR

B = 1 กำหนดให้มีการกระพริบที่ตำแหน่ง CURSOR

5. DISPLAY SHIFT

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	0	0	0	S/C	R/L	*	*

S/C = 0 กำหนดให้เลื่อน CURSOR ตามทิศทาง R/L ไป 1 ตำแหน่ง

S/C = 1 กำหนดให้เลื่อนข้อความแผงแสดงตามทิศทาง R/L ไป 1 COLUMN

R/L = 0 กำหนดให้มีทิศทางไปทางซ้าย

R/L = 0 กำหนดให้มีทิศทางไปทางขวา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6. FUNCTION SET

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	0	1	DL	N	F	*	*

DL = 0 กำหนดให้ติดต่อกับ LCD MODULE เป็นแบบ 4 BIT

DL = 1 กำหนดให้ติดต่อกับ LCD MODULE เป็นแบบ 8 BIT

จะสังเกตว่า การกำหนดค่า DL นี้ สามารถกระทำได้ที่ DB4 – DB7 ซึ่งถ้ามีการกำหนดให้เป็นแบบ 4 BIT ตั้งแต่แรก หลังจากจ่ายไฟเลี้ยง ก็จะทำให้ LCD MODULE มีการรับข้อมูลแบบ 4 BIT ทันที

N = 0 กำหนดจำนวนบรรทัดแบบ 1/8 DUTY และ 1/11 DUTY

N = 1 กำหนดจำนวนบรรทัดแบบ 1/6 DUTY

F = 0 กำหนดจำนวนบรรทัดแบบ 5*7 DOTS

F = 1 กำหนดจำนวนบรรทัดแบบ 5*10 DOTS (กรณีที่ LCD MODULE เป็นแบบ 5*7 อยู่แล้ว ก็จะไม่มีผลอะไร)

7. SET CGRAM ADDRESS

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	1	CGRAM ADDRESS					

สำหรับการกำหนด ADDRESS ของ CGRAM เมื่อได้ทำการกำหนดไว้แล้ว การอ่านและเขียน DATA ที่ต่อจากนี้ จะเป็นไปตาม ADDRESS ที่กำหนดทันที

8. SET DDRAM ADDRESS

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	1	DDRAM ADDRESS						

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำหรับการกำหนด ADDRESS ของ DDRAM เมื่อได้ทำการกำหนดไว้แล้ว การอ่านและเขียน DATA ที่ต่อจากนี้จะ เป็นไปตาม ADDRESS ในแต่ละรุ่นจะมีความแตกต่างกันบ้าง เพราะจำนวนอักษรต่อบรรทัดไม่เท่ากัน ซึ่ง แสดงดังต่อไปนี้

แบบการจัด ADDRESS ของ DD RAM หน้าจอแบบต่างๆ

16 ตัวอักษร 1 บรรทัด

00	01	02	03	04	05	06	04	40	41	42	43	44	45	46	47

16 ตัวอักษร 2 บรรทัด

00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F
40	41	42	43	44	45	46	47	48	49	4A	4B	4C	4D	4E	4F

16 ตัวอักษร 4 บรรทัด

00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F
40	41	42	43	44	45	46	47	48	49	4A	4B	4C	4D	4E	4F
10	11	12	13	14	15	16	17	18	19	1A	1B	1C	1D	1E	1F
50	51	52	53	54	55	56	57	58	59	5A	5B	5C	5D	5E	5F

20 ตัวอักษร 1 บรรทัด

00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	10	11	12	13

20 ตัวอักษร 2 บรรทัด

00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	10	11	12	13
40	41	42	43	44	45	46	47	48	49	4A	4B	4C	4D	4E	4F	50	51	52	53

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

9. BUS FLAG AND ADDRESS READ

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	1	BF	ADDRESS						

สำหรับการอ่านค่า BF (BUSY FLAGE) ซึ่งบอกถึงความพร้อมของ LCD MODULE ในการรับข้อมูล ถ้า BF = 0 หมายถึงว่าพร้อม ที่จะรับข้อมูลต่อไปได้ แต่ถ้า BF = 1 หมายถึงว่ายังไม่พร้อม นอกจากนี้ยังเป็นการอ่าน ADDRESS ของ CGRAM หรือ ADDRAM ด้วย

4.4 การอ่านและเขียนข้อมูลกับ DDRAM/CGRAM

1. WRITE DATA TO DDRAM OR CGRAM

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
1	1	DATA							

สำหรับการเขียนข้อมูลลงหน่วยความจำ DDRAM หรือ CGRAM โดยเมื่อทำการเขียนแล้ว ADDRESS จะถูกเพิ่มหรือลดลงโดยอัตโนมัติ ตามที่กำหนดจากค่า I/D ในคำสั่ง ENTRY MODE SET และการเขียนจะเป็น DDRAM หรือ CGRAM ก็ขึ้นกับว่า ก่อนหน้าคำสั่งนี้ มีการกำหนด ADDRESS ที่ใด

2. READ DATA FROM DDRAM OR CGRAM

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
1	1	DATA							

สำหรับการอ่านข้อมูลจากหน่วยความจำ DDRAM หรือ CGRAM โดยเมื่อทำการเขียนแล้ว ADDRESS จะถูกเพิ่มหรือลงโดยอัตโนมัติ ตามที่กำหนดจากค่า I/D ในคำสั่ง ENTRY MODE SET และการอ่านจะเป็น DDRAM หรือ CGRAM ก็ขึ้นอยู่กับว่า ก่อนหน้าคำสั่งนี้ มีการกำหนด ADDRESS ที่ใด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.5 แนวทางการเขียนโปรแกรมควบคุม

1. เมื่อจ่ายไฟเลี้ยงให้กับ LCD MODULE ครั้งแรก ภายในจะมีการ RESET ระบบโดยอัตโนมัติ ซึ่งจะใช้เวลา 10 ms หลังจากที่ระดับแรงไฟขึ้นถึง 4.5 V แล้ว ทั้งนี้ระบบ RESET ดังกล่าวจะกระทำสิ่งต่าง ๆ ดังต่อไปนี้
   - ทำการ CLEAR จอภาพ ทั้งหมด
   - กำหนดคุณสมบัติด้วยคำสั่ง FUNCTION SET คือ DL = 1 (ติดต่อกับระบบไมโครในแบบ 8BIT), N = 0 (แสดงผล 1 บรรทัด) , โชว์ 0 (กำหนดตัวอักษรแบบ 5*7 DOT)
   - กำหนดคุณสมบัติด้วยคำสั่ง DISPLAY ON/OFF คือ D= 0 (ไม่แสดงข้อมูล) , C = 0 (CURSOR OFF), B = 0 (BLANK OFF)
   - กำหนดคุณสมบัติด้วยคำสั่ง ENTRY MODE SET คือ I/D = 1 (INCREMENT), S = 0 (NO SHIFT)

การใช้งาน LCD MODULE ต้องรอให้ขบวนการ RESET ภายในทำงานเรียบร้อยก่อนซึ่งจะตรวจสอบได้ด้วย BF (BUSY FLAGE) หรืออาจจะใช้การหน่วงเวลาก็ได้

2. การใช้งาน LCD MODULE จะเกี่ยวข้องกับทางด้านโปรแกรมเป็นส่วนใหญ่ ชุดคำสั่งต่าง ๆรวมทั้งการอ่านหรือเขียนข้อมูลนั้น จะถูกกำหนดด้วยขาสัญญาณทั้งหมดที่มีอยู่ปกติโปรแกรมจะต้องกำหนดคุณสมบัติต่าง ๆ ที่ต้องการไว้ที่ส่วนต้น และจากนั้นก็จะเป็นการอ่านหรือเขียนข้อมูลลงใน DDRAM ซึ่งก็คือข้อความที่จะให้แสดงนั่นเอง

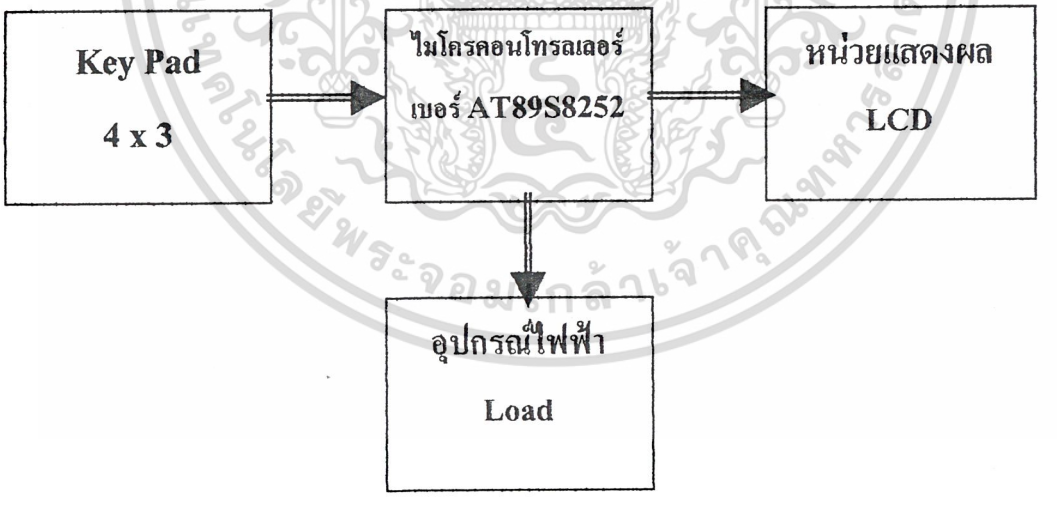
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5 การวิเคราะห์และการออกแบบ

ในการออกแบบโครงงานนี้จะแยกการออกแบบเป็น 2 ส่วนคือ ทางด้านฮาร์ดแวร์ (Hard ware) และทางด้านซอฟท์แวร์ (Software) เพื่อง่ายต่อการทดลอง โดยส่วนฮาร์ดแวร์จะอธิบายถึงการออกแบบวงจร การสร้างวงจร และการออกแบบรูปลักษณะภายนอกของโครงงาน ส่วนทางด้านซอฟท์แวร์ จะกล่าวถึงแนวทางในการเขียนโปรแกรมเพื่อที่จะให้เป็นไปตามรูปแบบของโครงงานที่กำหนดไว้

5.1 ส่วนของฮาร์ดแวร์ (Hardware)

โครงงานจะทำงานจากการป้อนข้อมูลทางสวิตช์ คีย์แพด ว่าให้อุปกรณ์ตัวใดทำงานและเปิด-ปิดที่เวลาใด แล้วทำการส่งข้อมูลไปยังตัวไมโครคอนโทรลเลอร์ เบอร์ AT89S8252 ซึ่งไมโครตัวนี้จะทำการสแกนคีย์เพื่อนำข้อมูลที่ส่งมา มาทำการประมวลผลเพื่อให้ทำงานได้ตรงตามที่กำหนด โดยจะมีอุปกรณ์บางตัวที่ช่วยให้การทำงานนั้นเป็นไปตามที่กำหนดไว้ เช่น ไอซี DS1307 ไอซีตัวนี้เป็นไอซีรีลไทม์คล็อก ซึ่งจะช่วยให้เวลามีความมาตรฐานและเที่ยงตรงมากยิ่งขึ้น, LELAY เป็นอุปกรณ์ที่ช่วยในการตัดต่อเมื่อถึงเวลาที่กำหนดว่าจะให้เปิด-ปิดเวลาใด, จอ LCD เป็นอุปกรณ์ที่แสดงผลเพื่อช่วยให้การใช้งานมีความง่ายยิ่งขึ้น

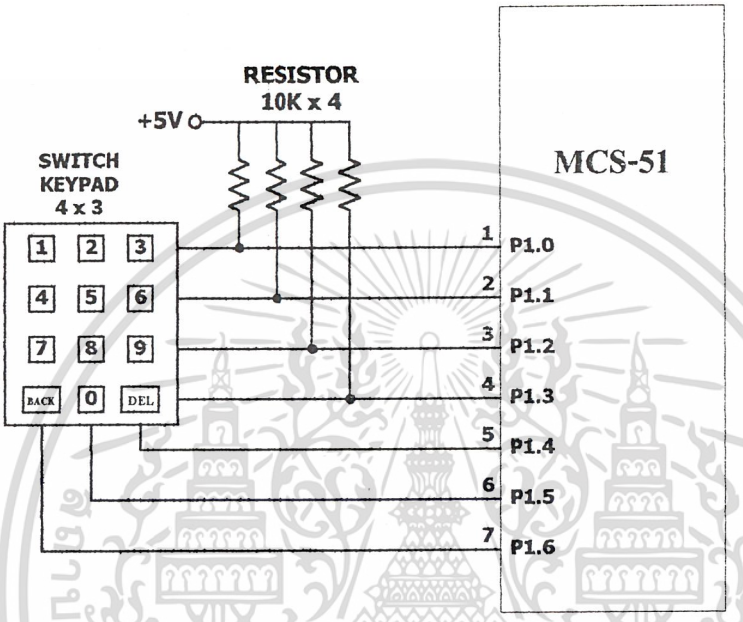


รูปที่ 5.1 บล็อกไดอะแกรมของเครื่องควบคุมการเปิด-ปิดอุปกรณ์ไฟฟ้าอัตโนมัติ

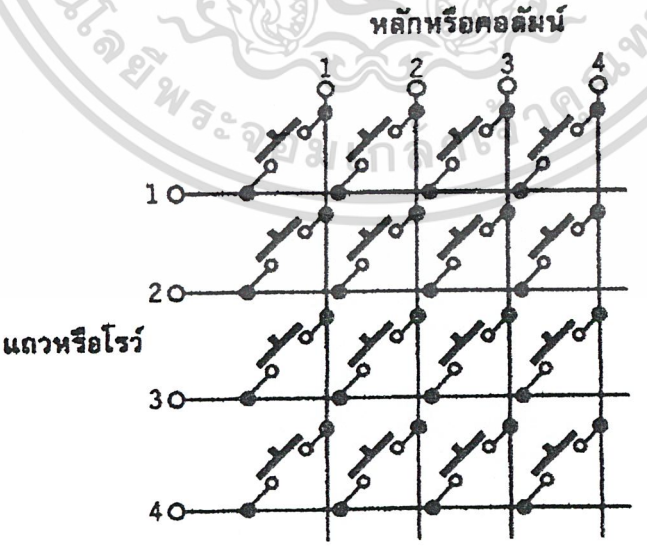
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.1.1 การออกแบบส่วนสวิตช์ คีย์แพด (Switch Keypad)

จะทำการต่อสวิตช์เป็นแบบเมตริกซ์ 4 x 3 โดยใช้สวิตช์ทั้งหมด 12 ตัว ซึ่งจะต่ออยู่กับ MCS-51 ที่พอร์ต 1 โดยต่อเป็นสายทั้งหมด 7 เส้นคือ สายของคอลัมน์ 3 เส้น และสายทางแถวอีก 4 เส้นและจะต้องใช้ตัวต้านทานพูลอัปไว้เพื่อกำหนดสภาวะเริ่มต้นที่ไม่มีการกดคีย์ ดังรูปที่ 5.2



รูปที่ 5.2 การต่อ สวิตช์ คีย์แพดกับไมโครคอนโทรลเลอร์



รูปที่ 5.3 วงจรของสวิตช์แบบเมตริกซ์หรือคีย์แพด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

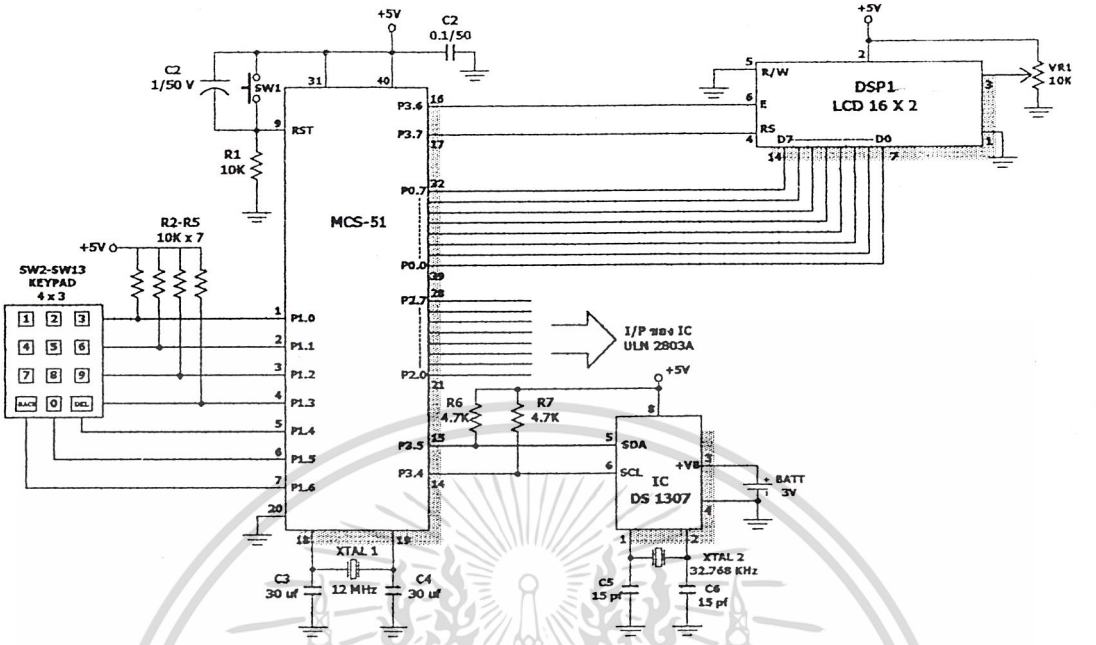
หมายเลข	ความหมาย
1	ใช้เป็นเลข 1
2	ใช้เป็นเลข 2
3	ใช้เป็นเลข 3
4	ใช้เป็นเลข 4
5	ใช้เป็นเลข 5
6	ใช้เป็นเลข 6
7	ใช้เป็นเลข 7
8	ใช้เป็นเลข 8
9	ใช้เป็นเลข 9
0	ใช้เป็นเลข 0 และใช้ SET (เวลา, ตั้งเวลา)
BACK	ใช้ย้อนกลับ ไป เมื่อไม่ทำภารกิจนั้นๆ
DEL	ใช้ลบเวลาเปิดปิด เพื่อป้อนเวลาใหม่

ตารางที่ 5.1 ความหมายของปุ่มต่างๆของสวิตช์ คีย์แพด

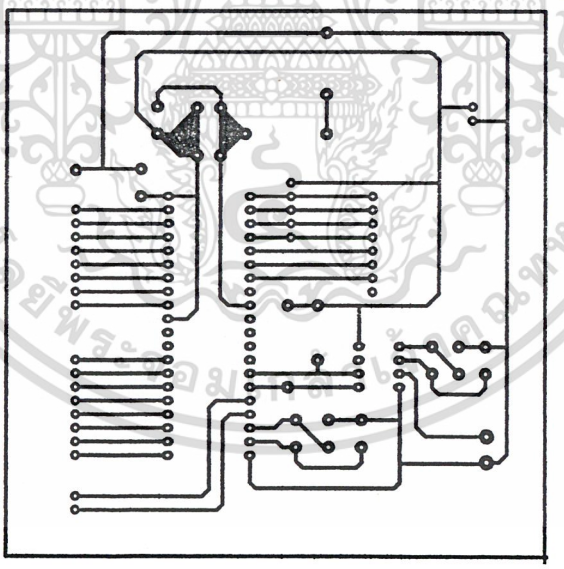
5.1.2 การออกแบบส่วนไมโครคอนโทรลเลอร์ (Microcontroller)

จะเป็นส่วนที่สำคัญที่สุด โดยไมโครคอน โทรลเลอร์ที่ใช้จะใช้เบอร์ AT89S8252 โดย พอร์ต p1.0-p1.6 จะใช้ต่ออยู่กับสวิตช์ คีย์แพด เพื่อรับข้อมูลเข้ามาแล้วทำการประมวลผล พอร์ต p0.0-p0.7 จะต่ออยู่กับจอ LCD โมดูล แบบ 16ตัวอักษร 2แถว เพื่อแสดงผลเมื่อ ไม โครคอนโทรลเลอร์ส่งข้อมูลออกไป พอร์ต p2.0-p2.7 จะต่ออยู่กับไอซี ULN2803A ซึ่งเป็นไอซีไดรเวอร์ที่ใช้ขับรีเลย์เพื่อทำการตัดต่ออุปกรณ์ไฟฟ้าตามที่กำหนด และ p3.4 กับ p3.5 จะต่ออยู่กับไอซี DS1307 ซึ่งเป็นไอซีสร้างฐานเวลาจริงหรือรีลไทม์คล็อก จะมีหน้าที่สร้างฐานเวลาจริงให้แก่ระบบไมโครฯ เพื่อให้รีเลย์สามารถตัดต่ออุปกรณ์ตามเวลาที่เที่ยงตรงได้ โดยจะมีแบตเตอรี่ 3V มาต่อกับ DS1307 อีกด้วยเพื่อรักษาการทำงานของวงจรสร้างฐานเวลาให้คงอยู่ต่อไป แม้ว่าไม่มีไฟเลี้ยงจ่ายให้แก่ DS1307 ดังรูปที่ 5.4

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5.4 การใช้งานของพอร์ตของไมโครคอนโทรลเลอร์



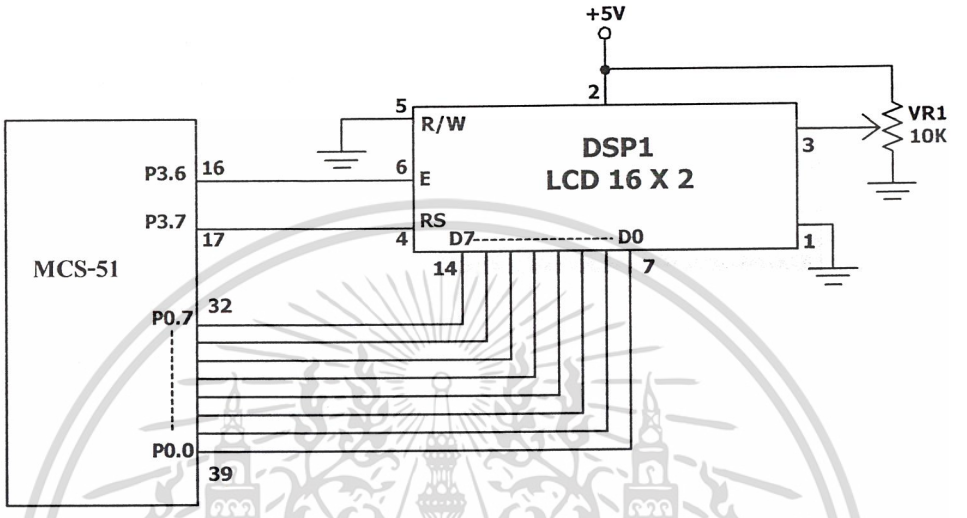
รูปที่ 5.5 ลายวงจรของภาคไมโครคอนโทรลเลอร์

5.1.3 การออกแบบส่วนการแสดงผล (Display)

จะใช้ LCD โมดูล แบบ 16 ตัวอักษร 2 แถว มาเป็นหน่วยแสดงผล เพื่อให้ผู้ใช้งานสามารถดูได้เมื่อเวลาใช้งานจริง จะช่วยให้การใช้งานนั้นง่ายขึ้น โดยจะต่ออยู่กับพอร์ต 0 ของ MCS-51 และ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

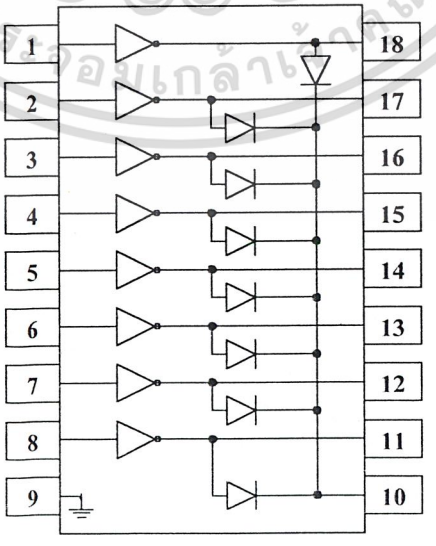
จะมีตัวต้านทานปรับค่าได้ ค่า 10k มาช่วยในการปรับความเข้มของหน้าจอ LCD เมื่อเวลาใช้งาน และจะนำขา 6 ซึ่งเป็นขา Enable และขา 4 ซึ่งเป็นขา Register select ของ LCD ไปต่อกับพอร์ต p3.6 และ พอร์ต p3.7 ของ MCS-51 ตามลำดับ ซึ่งใช้ในการควบคุม ดังรูปที่ 5.6



รูปที่ 5.6 การต่อจอ LCD โมดูลกับ MCS-51

5.1.4 การออกแบบส่วนควบคุมอุปกรณ์ไฟฟ้า (Load control)

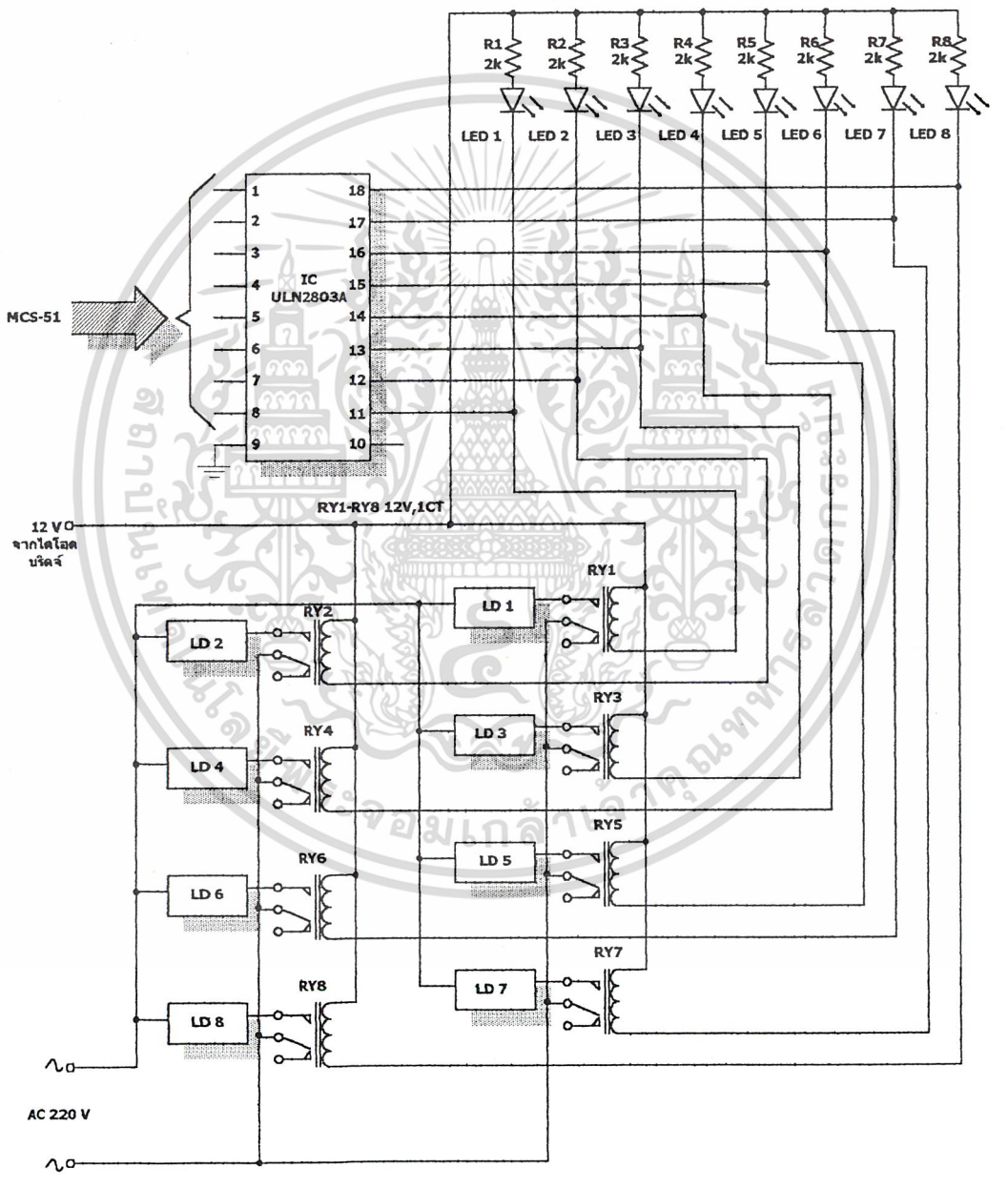
จะมีไอซีเบอร์ ULN2803A มาต่ออยู่กับพอร์ต 2 ของ MCS-51 โดยไอซีตัวนี้เป็นไอซีไดรเวอร์แบบอินเวอร์เตอร์ที่สามารถจ่ายกระแสได้สูงสุด 200 mA และมีไดโอดป้องกันการยุบตัวของสนามแม่เหล็กอยู่ภายใน ดังรูปที่ 5.7 สามารถนำสัญญาณเอาท์พุตที่ออกจากไอซีนี้ ไปใช้ขับรีเลย์



รูปที่ 5.7 วงจรภายในของไอซี ULN2803A

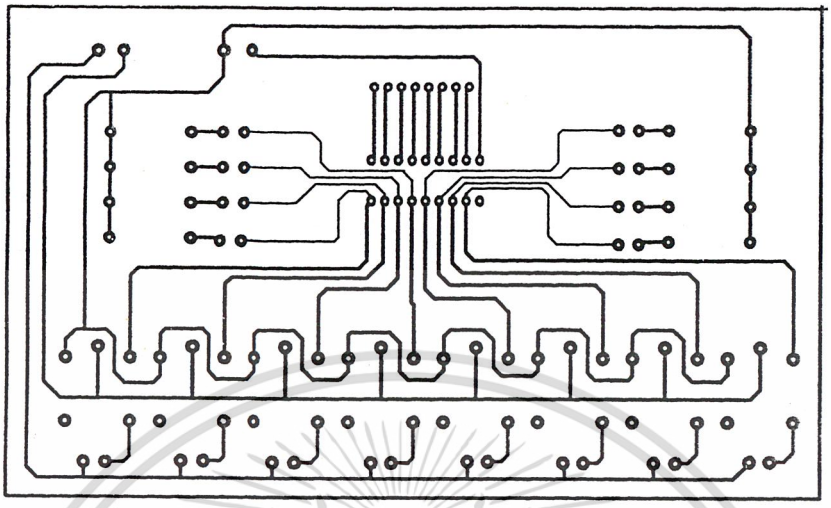
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

12V ทั้ง 8 ตัว ได้โดยตรงเพื่อใช้ตัดต่ออุปกรณ์ไฟฟ้า (ในที่นี้จะใช้หลอดไฟ 220V มาเป็นโหลดแทนอุปกรณ์ไฟฟ้า เพื่อให้เห็นชัดและง่ายต่อการทดลอง แต่จะมีเต้ารับที่ไว้สำหรับต่อใช้งานกับอุปกรณ์ไฟฟ้าจริง) ส่วนอีกข้างหนึ่งของรีเลย์ทั้ง 8 ตัว จะต่ออยู่กับ LED1-LED8 เพื่อแสดงผลเมื่ออุปกรณ์นั้นๆทำงานตามเวลาที่ได้ตั้งไว้ โดยจะมีตัวต้านทานต่อไว้ เพื่อจำกัดกระแสที่ไหลผ่าน LED ให้พอเหมาะ ดังรูปที่ 5.8



รูปที่ 5.8 วงจรควบคุมรีเลย์เพื่อใช้ตัดต่อการเปิด-ปิดอุปกรณ์ไฟฟ้า

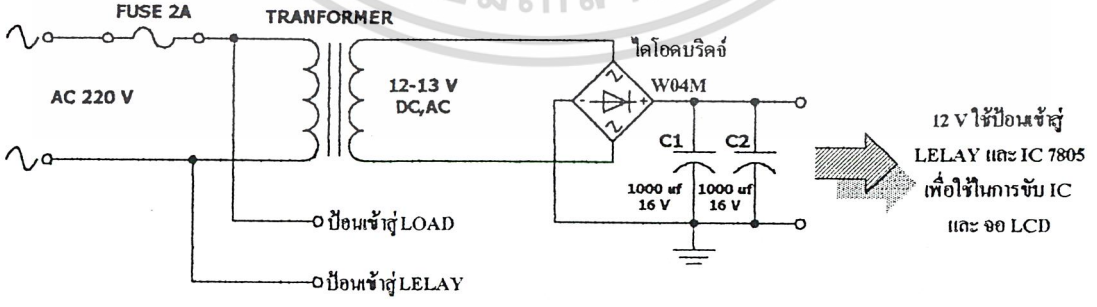
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5.9 ลายวงจรของภาคควบคุมโหลด

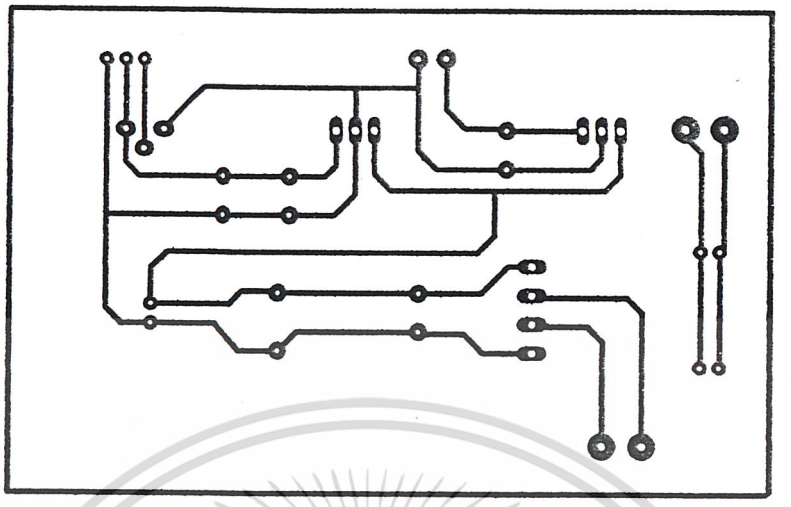
5.1.5 การออกแบบส่วนภาคจ่ายไฟ (Supply)

ในวงจรจะใช้ไฟเลี้ยงตั้งแต่ 220V, 12V, 5V เพราะฉะนั้นจึงใช้หม้อแปลงแบบ เข้า 220V ออก 12V มาใช้ในการจ่ายไฟให้กับวงจร โดยมีไอซี 7805 ซึ่งเป็นไอซีเร็กกูเลท มีหน้าที่แปลงไฟจาก 12V เป็น 5V มาใช้ในการจ่ายไฟให้กับ ไอซีต่างๆภายในวงจร และจอ LCD ด้วย ส่วน 12V จะใช้ในการจ่ายไฟให้กับ LELAY เพื่อใช้ในการตัดต่ออุปกรณ์ไฟฟ้า และไฟ 220V จะใช้ต่อให้กับโหลดของวงจร ดังรูปที่ 5.10



รูปที่ 5.10 วงจรของภาคจ่ายไฟ

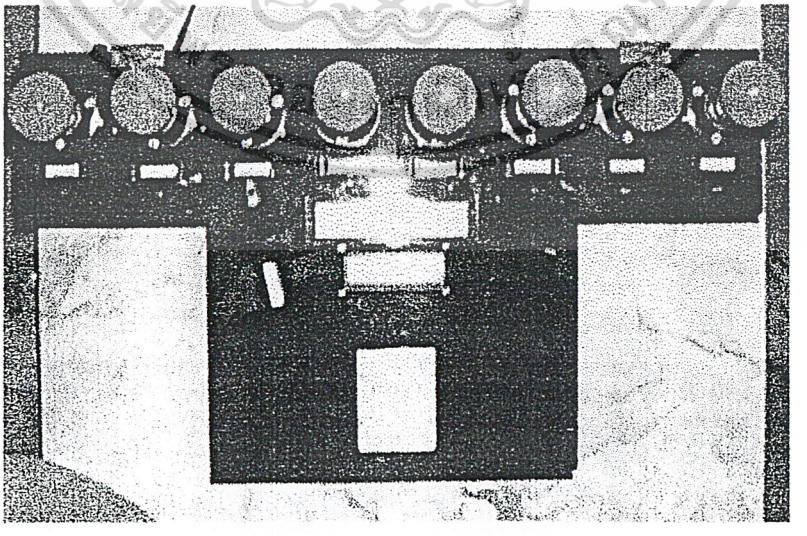
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5.11 ลายวงจรของภาคจ่ายไฟ

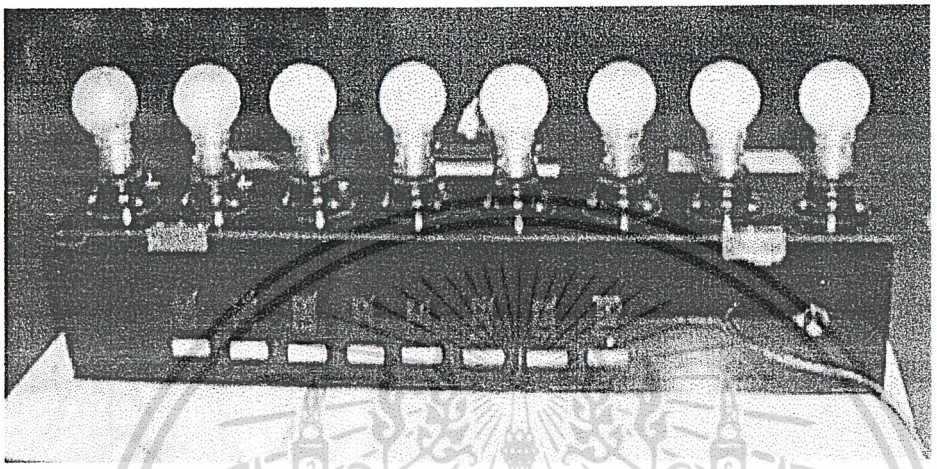
5.1.6 การออกแบบรูปลักษณะภายนอกของโครงงาน (Box)

จะออกแบบให้เป็นรูปตัว T เพื่อให้เหมาะกับการทดลอง ซึ่งด้านบนจะมี หลอดไฟ 220V 8 ดวง LED 8 ดวง ที่ใช้ช่วยในการทดลอง และจะมีสวิตช์ คีย์แพด กับ จอ LCD ไว้เพื่อให้ใช้งานได้ง่ายยิ่งขึ้น ดังรูปที่ 5.12 ส่วนด้านหลัง จะทำการใสเต้ารับไว้ เพื่อใช้ในการต่อกับอุปกรณ์ไฟฟ้าในเวลาใช้งานจริง ดังรูปที่ 5.13

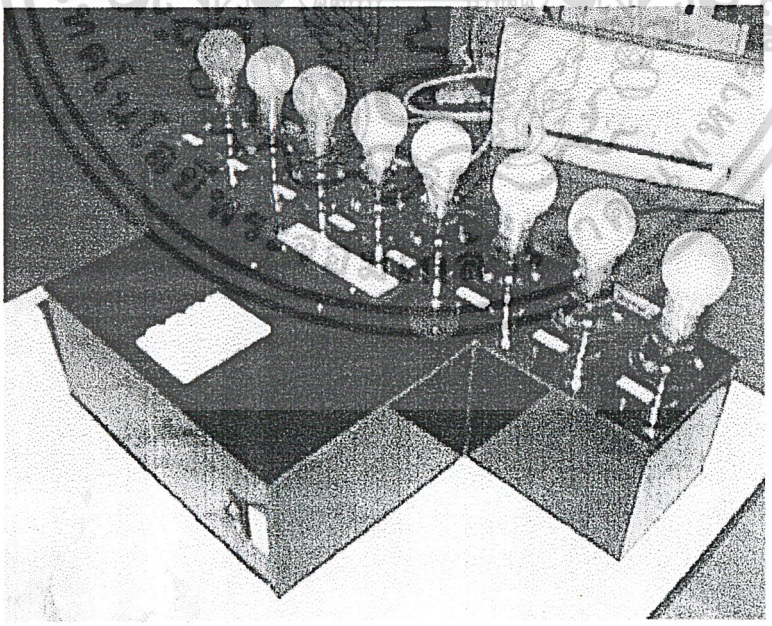


รูปที่ 5.12 ภาพด้านบนของลักษณะภายนอกของโครงงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5.13 ภาพด้านหลังของลักษณะภายนอกของโครงงาน



รูปที่ 5.14 ภาพมุมเอียงของลักษณะภายนอกของโครงงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

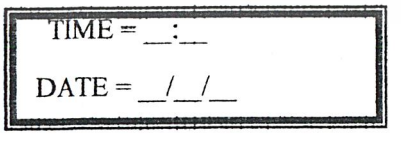
5.2 ส่วนของซอฟท์แวร์ (Software)

ในการเขียนโปรแกรมแต่ละครั้งก็ต้องมีเป้าหมายในการเขียนโปรแกรมอยู่แล้ว และในโครงงานนี้ก็มีเป้าหมายหรือความต้องการของโครงงานดังนี้

เมื่อเปิดเครื่องขึ้นมาให้หน้าจอแสดงคำว่า KMITL ในบรรทัดที่ 1 และ ENGINEERING ในบรรทัดที่ 2 ต่อมาก็ขึ้นค่าเวลา และวันที่ ที่ถูกต้อง ณ ตำแหน่งทางขวาของจอ LCD เพื่อรอการเข้าสู่โหมดการทำงานต่อไป

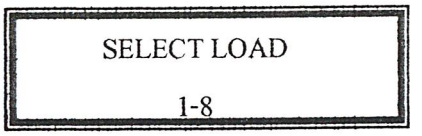
การเข้าสู่โหมดการทำงานนั้น จะใช้ปุ่มหมายเลข 0 เข้าสู่โหมดการทำงาน โดยเมื่อกดปุ่มหมายเลข 0 หน้าจอจะขึ้นคำว่า 1. SET TIME ในบรรทัดที่ 1 ก็คือ ใช้ตั้งค่าเวลาและวันที่ใหม่เพื่อให้ตรงกับความเป็นจริง และขึ้นคำว่า 2. SET LOAD ในบรรทัดที่ 2 ก็คือ ใช้ตั้งเวลาการเปิด-ปิดอุปกรณ์ไฟฟ้า

ถ้ากดหมายเลข 1 ก็คือจะเข้าสู่โหมดตั้งเวลาและวันที่ใหม่ หน้าจอจะขึ้นคำว่า TIME = __:__ ในบรรทัดที่ 1 เพื่อใช้ตั้งค่าเวลา และขึ้น DATE = __/__/__ ในบรรทัดที่ 2 เพื่อใช้ตั้งวันที่ เดือน ปี (ในที่นี้ ปีจะแสดงเป็นเลข ค.ศ.) จากนั้นก็ป้อนเวลาและวันที่เข้าไป โดยจะเรียงจากซ้ายไปขวา เท่านี้ก็เป็นอันเสร็จสิ้นหน้าจอจะขึ้นเวลาและวันที่ ที่ตั้งเอาไว้อยู่ทางมุมขวามือเหมือนตอนแรก



ถ้ากดหมายเลข 2 ก็คือจะเข้าสู่โหมดการตั้งเวลาการเปิด-ปิดอุปกรณ์ไฟฟ้า หน้าจอจะขึ้นคำว่า SELECT LOAD ในบรรทัดที่ 1 และ ขึ้น 1-8 ในบรรทัดที่ 2 ก็คือ ให้ผู้ใช้งานได้เลือกว่าจะตั้งเวลาการเปิด-ปิดอุปกรณ์ตัวใด

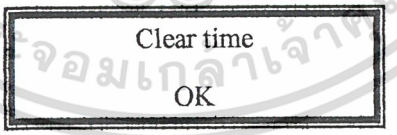
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เมื่อกดหมายเลขอุปกรณ์ที่ทำการเลือกแล้ว ก็จะเข้าสู่การตั้งเวลาเปิดและปิด โดยหน้าจอขึ้นคำว่า START TIME __:__ ในบรรทัดที่ 1 และ STOP TIME __:__ ในบรรทัดที่ 2 จากนั้นก็ป้อนเวลาที่ต้องการที่จะให้อุปกรณ์ไฟฟ้านั้นๆเปิด และเวลาที่ต้องการจะให้ปิด เท่านี้เป็นการเสร็จสิ้นในการตั้งเวลาการเปิด-ปิด หน้าจอจะกลับมาขึ้นคำว่า SELECT LOAD ในบรรทัดที่ 1 และ 1-8 ในบรรทัดที่ 2 อีกครั้งเพื่อให้ผู้ใช้งานได้ทำการตั้งเวลาการเปิด-ปิด อุปกรณ์ตัวอื่นอีกต่อไป

ในการใช้งาน เมื่อกรณีที่ผู้ใช้งาน กดเข้ามาแล้วแต่เกิดไม่อยากใช้งาน ก็สามารถใช้ปุ่ม BACK เพื่อใช้ในการย้อนกลับไปหน้าเก่าก่อนที่จะเข้ามาได้อีกด้วย

ในกรณีที่ผู้ใช้งานต้องการจะตั้งเวลาเปิด ปิดอุปกรณ์ไฟฟ้านั้น บางครั้งอาจจะมีเวลาที่ตั้งเอาไว้ ก่อนๆอยู่ สามารถลบได้ด้วย ปุ่ม DEL แล้วถึงจะป้อนเวลาเปิด ปิดเข้าไป แต่ถ้าอยากจะให้เวลานั้นหายไปเลย ก็กดปุ่ม DEL 2 ครั้ง เท่านี้เวลาที่ขึ้นมาก่อนๆ ก็จะหายไป โดยที่หน้าจอจะขึ้นคำว่า Clear time ในบรรทัดที่ 1 และ OK ในบรรทัดที่ 2

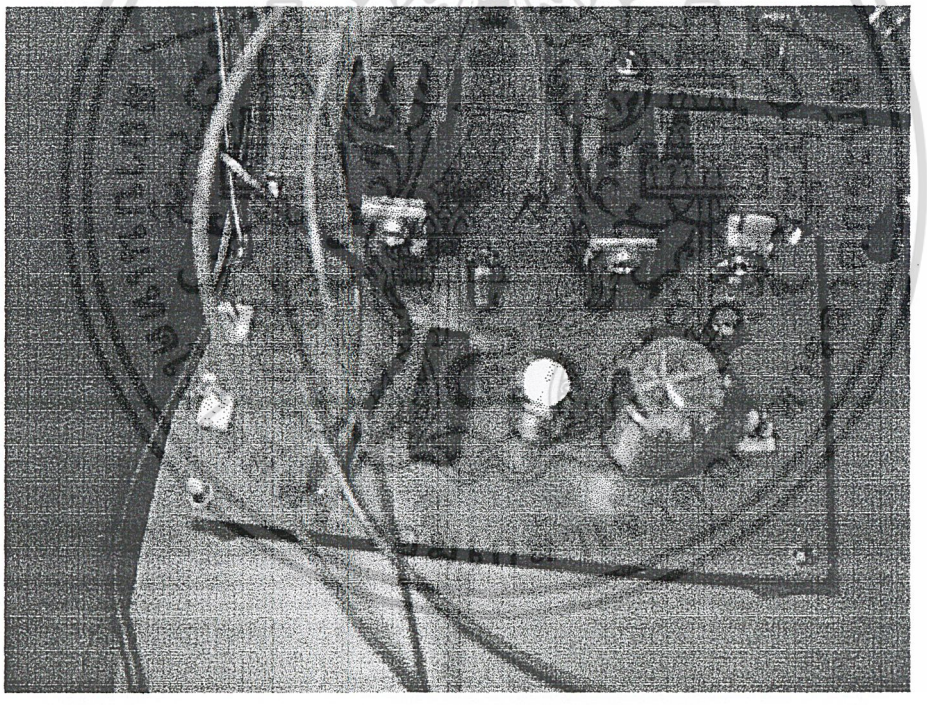


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 6 การทดลองและผลการทดลอง

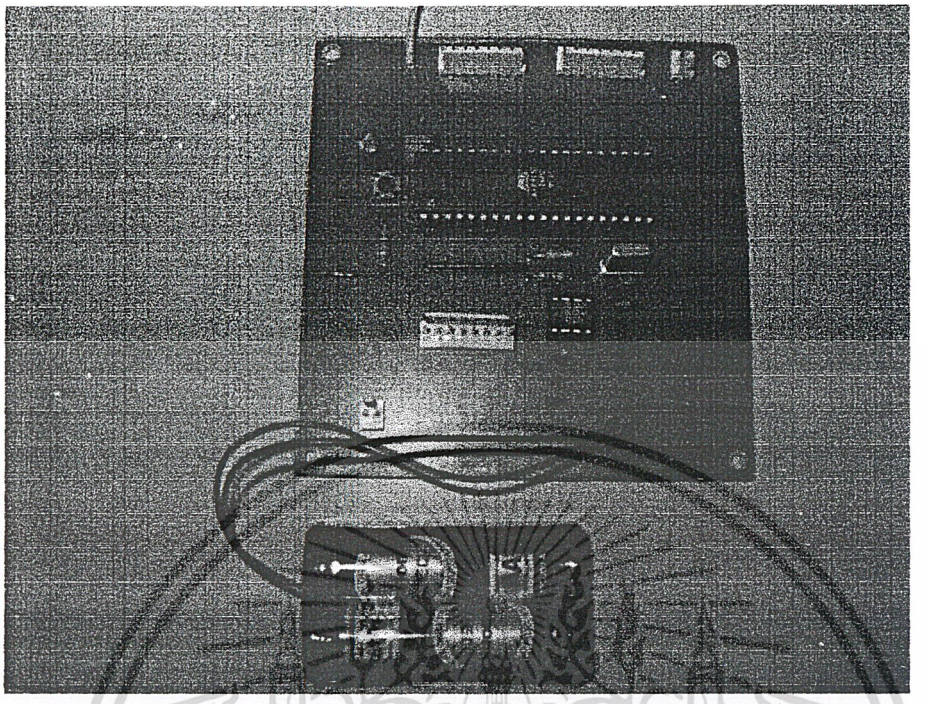
6.1 การทดลองและผลการทดลองทางด้านฮาร์ดแวร์ (Hardware)

นำวงจรที่บัดกรีเสร็จแล้ว ทั้งวงจรจ่ายไฟ วงจรไมโครคอนโทรลเลอร์ และวงจรควบคุมโหลด มาทดลองต่อ เพื่อทำการทดสอบ LELAY กับอุปกรณ์ภายนอก และหลอดไฟ 220V โดยการต่อไฟ 220V เข้าไป ไฟจะผ่านหม้อแปลง แล้วออกมาเป็น 12V แล้วจากนั้นนำไฟ 12V เข้าสู่ LELAY และไอซี 7805 เพื่อแปลงไฟเป็น 5V เพื่อจ่ายในการทดสอบ และนำไฟ 220V จ่ายให้กับ Load ด้วย จากนั้นนำไฟ 5V ที่ได้จาก ไอซี 7805 มาลองทดสอบ โดยการป้อนเข้าทางขา 1-ขา 8 ของไอซี ULN2803A ทีละขา แล้วดูการเปลี่ยนแปลงที่ LED และหลอดไฟ 220V ถ้า LED และ หลอดไฟ 220V ติด ตามการป้อนไฟ 5V ที่ขาไอซี ULN2803A ทีละขา แสดงว่าวงจรที่ทำออกไม่มีความผิดพลาด ดังรูปที่ 6.5

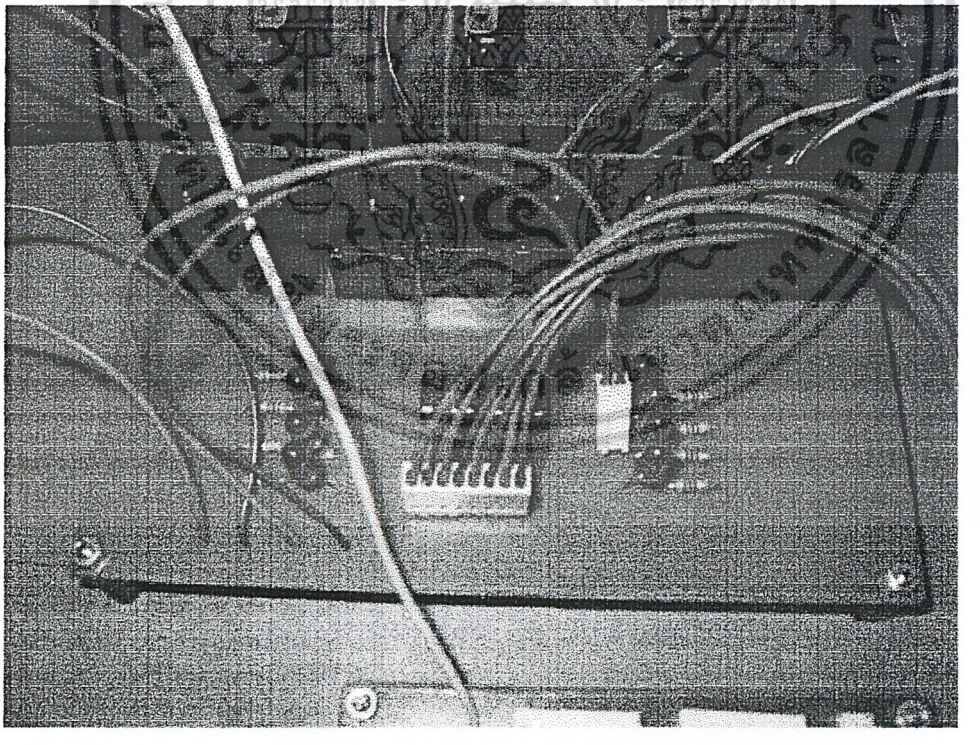


รูปที่ 6.1 วงจรจ่ายไฟที่บัดกรีเสร็จแล้ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

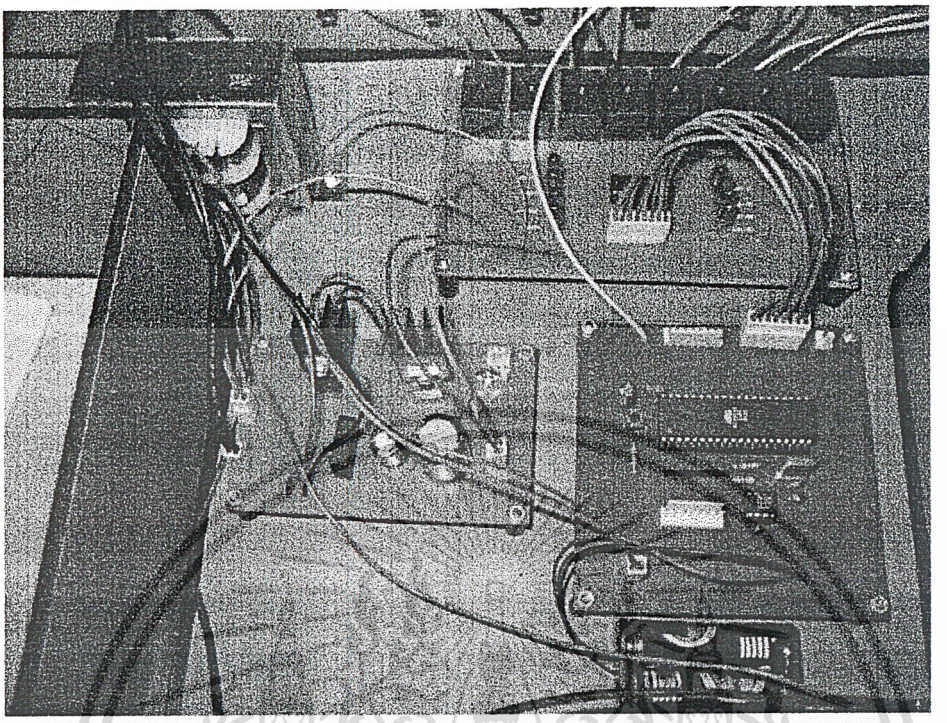


รูปที่ 6.2 วงจรไมโครคอนโทรลเลอร์ที่บัดกรีเสร็จแล้ว

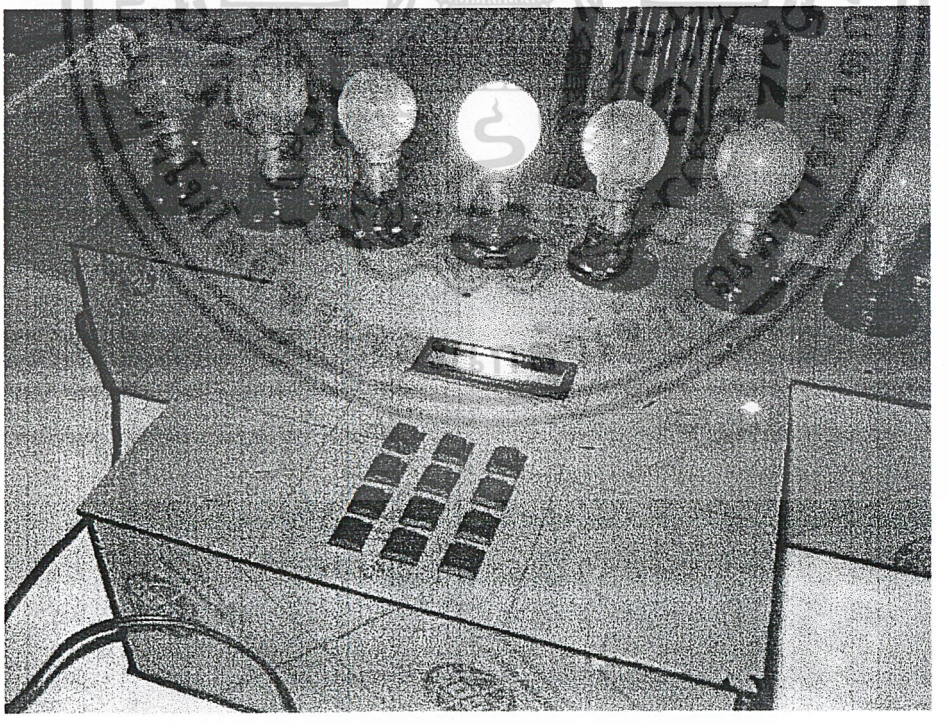


รูปที่ 6.3 วงจรที่ใช้ควบคุมโหลดที่บัดกรีเสร็จแล้ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 6.4 การต่อวงจรก่อนการทดลอง

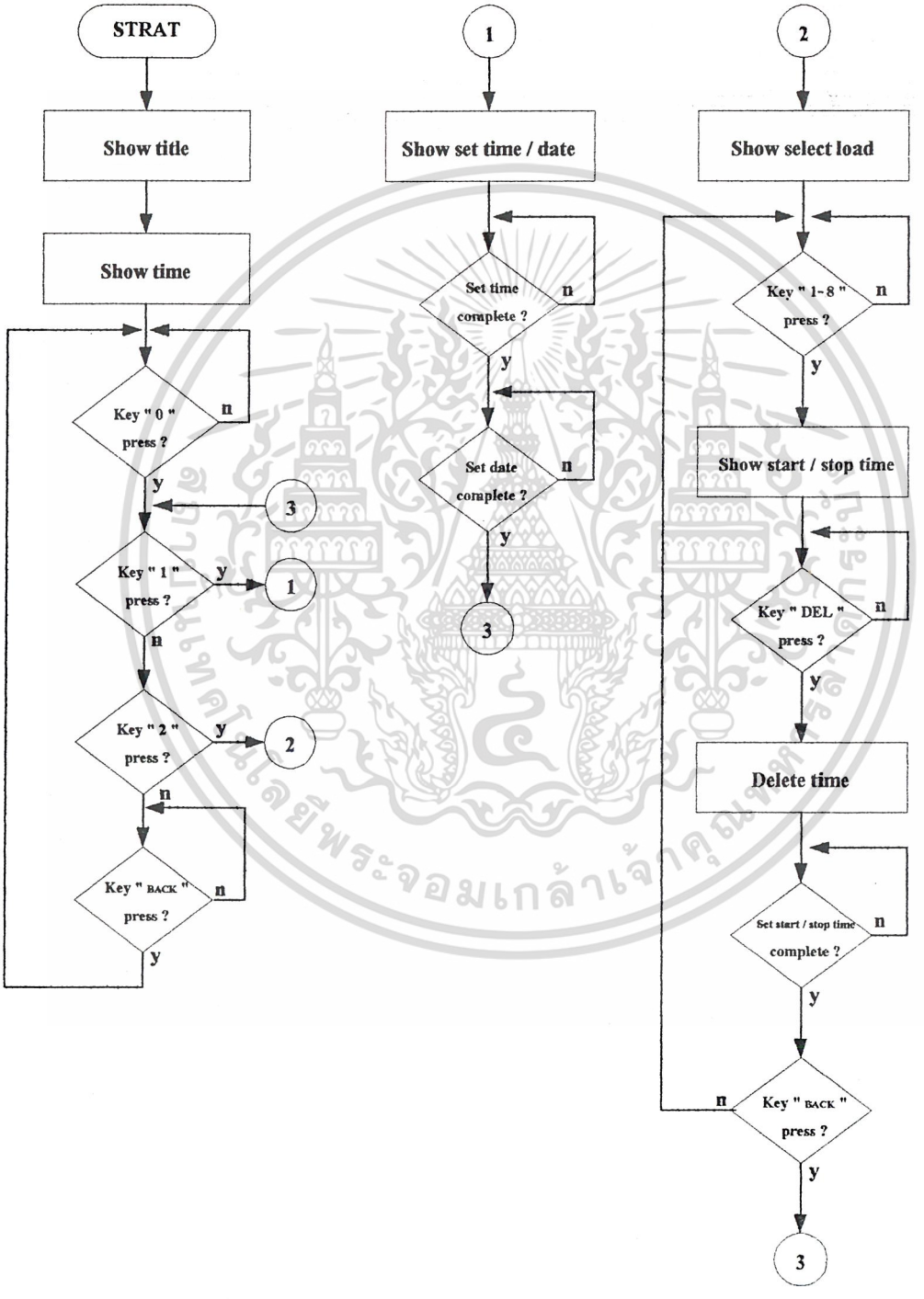


รูปที่ 6.5 แสดงผลการทดลองที่ไม่มีความผิดพลาดในวงจร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6.2 การทดลองและผลการทดลองทางด้านซอฟท์แวร์ (Software)

นำวงจรหรือฮาร์ดแวร์ที่ได้จากการทดลองทางด้านบนที่สมบูรณ์แล้ว มาทำการทดลองทางด้านซอฟท์แวร์ โดยนำไอซีไมโครคอนโทรลเลอร์ไปทำการโหลดโปรแกรมลงไปในตัวไอซี แล้วทำการทดลองตามคู่มือการใช้งาน



รูปที่ 6.6 แสดง flowchart ของโปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6.2.1 โปรแกรมของเครื่องควบคุมการปิด-เปิด อัตโนมัติ โดยใช้ MCS-51

 # ตัวอย่างโปรแกรม #

```
#include <at89x52.h>
#include <stdio.h>
#include <absacc.h>
#include <ctype.h>
#include <stdlib.h>
#include <intrins.h>
#include <string.h>

#define   BUFSIZE   16        /* ASCII buffer size */
#define   FALSE     0
#define   TRUE      1
#define   LF        10        /* line feed chr */
#define   CR        13        /* return chr */
#define   RTCADD    0xd0      /* ds1307 RTC address */

#define   k_cancel  10
#define   k_ent     11

sbit      SCL   =  P3^4;      /* I2c SCL bit */
sbit      SDA   =  P3^5;      /* I2c SDA bit */

sbit      lcdrs  =  P3^7;
sbit      lcdrw  =  P1^7;
sbit      lcden  =  P3^6;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
char readkbd (void);

/************************/

/* Gobal variable */

/************************/

char   data   a, b, c, d, k_a, k_b, TimeMin, TimeHour;
char   data   ASCBUF[BUFSIZE];
char   code   scankey[]={0xbf, 0xdf, 0xef};
char   code   keycode[]={1, 4, 7, 10, 2, 5, 8, 0, 3, 6, 9, 11};
char   code   tbl1[]={0x01, 0x02, 0x04, 0x08, 0x10, 0x20, 0x40, 0x80};

bit    data   TimeFlag;

typedef struct CLOCK                    /* Structure for RTC */
        {
                char sec;
                char min;
                char hour;
                char day;
                char date;
                char month;
                char year;
        } CLOCK;

CLOCK RTC;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
/**********************/
/*Bit delay*/
/**********************/

void Bitdl (void)
    {
        _nop_();                    /* Delay time for read EEPROM */
        _nop_();
    }

/**************************/
/*Clear(L) SCL line*/
/**************************/

void Clrscl (void)
    {
        SCL = 0;                    /* clear line  */
        Bitdl();
    }

/************************/
/*Set(H) SCL line*/
/************************/

void Setscl (void)
    {
        SCL = 1;                    /* set line   */
        while(!SCL);
        Bitdl();
    }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
/*****************************/
/*Pulse gen for SCL line*/
/*****************************/

void Plscl (void)
    {
        Setscl();
        Clrscl();
    }

/*************************/
/*Set stop condition*/
/*************************/

void Stop (void)
    {
    SDA = 0;Setscl();
    SDA = 1;Bitdl();
    }

/*************************/
/*Set start condition*/
/*************************/

void Start (void)
    {
        SDA = 1;
        Setscl();
        SDA = 0;
        Bitdl();
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        Clrscl();
    }

                /*************************************/
                    /*Receive data(byte) from I2C bus*/
                /*************************************/

char Rxbyte (void)
    {
        unsigned char tmp, dat;
        dat = 0;
    for(tmp=1; tmp<=8; tmp++)                    /* Send 1 byte to EEPROM */
        {
            dat <<= 1;Setscl();
            dat |= SDA;Clrscl();
        }
    return(dat);
    }

                /*************************************/
                    /*Transmit data(byte) to I2C bus*/
                /*************************************/

bit Txbyte (unsigned com)
    {
        bit erflag;
         unsigned char tmp;
    for(tmp=1; tmp<=8; tmp++)
        {
            SDA   = com & 0x80;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                com <<= 1;
                Plscl();
            }

    SDA = 1;
    Setscl();
    if(SDA) erflag = 1;
    else    erflag = 0;
    Clrscl();
    return(erflag);
}
/*  Get data from any register  */
/*  in -->>  address(addr)     */
/*  in -->>  register(reg)     */
/*  out <<--  data(*dat)       */
/*  err <<--  0 = no error,1 = error */

bit Getreg (char addr,char reg,char *dat)
        {
            EA=0;
            if((SCL==0) || (SDA==0))
                {                                   /* check line first     */
                    EA=1;
                    return(1);
        }

    Start();
        if(Txbyte(addr))
            {                                       /* Xmit register address  */
                Stop();
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                EA=1;
                return(1);
            }
        if(Txbyte(reg))
            {                                   /* Xmit register command  */
                Stop();
                EA=1;
                return(1);
            }
    Start();                                    /* Set start condition    */
        if(Txbyte(addr|0x01))
            {                                   /* Xmit register address + receive command */
                Stop();
                EA=1;
                return(1);
            }
                *dat = Rxbyte();                /* Recv data byte         */
                SDA = 1;
                Plscl();
                Stop();
                EA=1;
                return(0);                      /* return data to caller  */
            }
```

```
/*  Put data to any register   */
/*  in -->>  address(addr)     */
/*  in -->>  register(reg)     */
/*  in -->>  data(dat)         */
/*  err <<--  0 = no error,1 = error */
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
bit Putreg(char addr,char reg,char dat)
        {
                EA=0;
                 if((SCL==0) || (SDA==0))
        {                                                   /* Ckeck line first     */
                EA=1;
                return(1);                                  /* line is busy return 1 */
        }

   Start();                                                 /* Set Start condition   */
        if(Txbyte(addr))
                {                                           /* Xmit register address */
                        Stop();
                        EA=1;
                        return(1);
                }
        if(Txbyte(reg))
                {                                           /* Xmit register command */
                        Stop();
                        EA=1;
                        return(1);
                }
Txbyte(dat);                                                /* Xmit data          */
Stop();                                                     /* Set stop condition    */
                        EA=1;
                        return(0);
                }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
/*********************************/
/*Get time from RTC(Ds1307)*/
/*********************************/

bit GetRTC (void)
    {
        Getreg(RTCADD,0,&RTC.sec);          /* get second     (00 - 59) */
        Getreg(RTCADD,1,&RTC.min);          /* get minute     (00 - 59) */
        Getreg(RTCADD,2,&RTC.hour);         /* get hour       (00 - 23) */
        Getreg(RTCADD,3,&RTC.day);          /* get day of week (01 - 07) */
        Getreg(RTCADD,4,&RTC.date);         /* get date       (01 - 31) */
        Getreg(RTCADD,5,&RTC.month);        /* get month      (01 - 12) */
        Getreg(RTCADD,6,&RTC.year);         /* get year       (00 - 99) */
        return (0);
    }

/***********************/
/*Set time to rtc*/
/***********************/

void SetRTC (void)
    {
        Putreg(RTCADD,0,RTC.sec);           /* Put to RTC */
        Putreg(RTCADD,1,RTC.min);
        Putreg(RTCADD,2,RTC.hour);
        Putreg(RTCADD,4,RTC.date);          /* Put to RTC */
        Putreg(RTCADD,5,RTC.month);
        Putreg(RTCADD,6,RTC.year);
    }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
/*****************************/
/*Delay function*/
/*input : int loop counter*/
/*****************************/

void delay (unsigned int count)
    {
        for (; count; count--)                  /* Loop Delay */
        {
            if (RI==1)                          /* Check Serial port recieve bit */
            break;
        }
    }

        /*****************************/
        /*Check Keyboard hit*/
        /*****************************/

bit kbhit (void)
    {
        P1=0x0f;
        if ((P1&0x0f)!=0x0f)                    /* Key hit return true */
        return (TRUE);
    else
        return (FALSE);                         /* Non return false */
    }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
/**********************/
/*Read keyboard*/
/**********************/

char readKbd (void)
    {
        while (1)
        {
            for (k_a=0; k_a<3; k_a++)              /* Loop scan 4 bit */
            {
                P1=scankey[k_a];                   /* Check key press */

                k_b=P1&0x0f;
                if (k_b==0x0f)
                continue;
            else
            {
                delay (3000);                      /* Key debouce */
                k_b=P1&0x0f;
                if (k_b==0x0f)
                continue;
                if (k_b==0x0e)                     /* bit 1 */
                    k_b=0;
            if (k_b==0x0d)                         /* bit 2 */
                    k_b=1;
            if (k_b==0x0b)                         /* bit 3 */
                    k_b=2;
             if (k_b==0x07)                        /* bit 4 */
                    k_b=3;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                    k_a*=4;
                    k_b+=k_a;
                    k_b=keycode[k_b];              /* Key encode */

                    while ((P1&0x0f)!=0x0f);       /* Loop while depress */
                    return (k_b);                  /* return key */
                    }
                }
        }
}

                    /***********************/
                    /* Write Instruction */
                    /***********************/

void lcdwi (char cmd)
        {
                P0=cmd;
                lcdrs=0;
                lcdrw=0;
                lcden=0;
                lcden=1;
                _nop_ ();
                _nop_ ();
                lcden=0;
                delay (128);
        }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
/*****************/
/* Write Data */
/*****************/

void lcdwd (char dta)
    {
        P0=dta;
        lcdrs=1;
        lcdrw=0;
        lcden=0;
        _nop_ ();
        _nop_ ();
        lcden=1;
        lcden=0;
        delay (128);
    }

/******************/
/* Initial LCD */
/******************/

void initlcd (void)
    {
        lcdwi (0x38);
        lcdwi (0x0c);
        lcdwi (1);
        lcdwi (0x80);
    }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
/*********************/
/*Outtext to LCD*/
/*********************/

void text (char *msg)
    {
        while (*msg!=0L)
        {
            lcdwd (*msg);
            msg++;
        }
    }

/************************/
/*Settime function*/
/************************/

void Settime (void)
    {
        initlcd ();
        text ("TIME = __:__ ");
        lcdwi (0xc0);
        text ("DATE = __/__/__");
        lcdwi (0x0f);  /* on cursor */
        lcdwi (0x87);
        b=0;
    while (1)
    {
        a=readkbd ();
    if (a==k_cancel)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            break;
 if ((a>=0) && (a<=9))
 {
            ASCBUF[b]=a;
            b++;
            a+=0x30;
            lcdwd (a);
 if (b==2)
            lcdwi (0x8a);
if (b==4)
            lcdwi (0xc7);
if (b==6)
            lcdwi (0xca);
if (b==8)
            lcdwi (0xcd);
if (b==10)
{
            ASCBUF[0]<<=4;
            ASCBUF[0]|=ASCBUF[1];
            RTC.hour=ASCBUF[0];

            ASCBUF[2]<<=4;
            ASCBUF[2]|=ASCBUF[3];
            RTC.min=ASCBUF[2];

            ASCBUF[4]<<=4;
            ASCBUF[4]|=ASCBUF[5];
            RTC.date=ASCBUF[4];

            ASCBUF[6]<<=4;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                ASCBUF[6]|=ASCBUF[7];
                RTC.month=ASCBUF[6];

                ASCBUF[8]<<=4;
                ASCBUF[8]|=ASCBUF[9];
                RTC.year=ASCBUF[8];

                RTC.sec=0;

        SetRTC ();
        break;
                }
            }
        }
    }

                /*******************/
                /*Load time*/
                /*******************/

void Loadtime (char n)
    {
                n--;
                n*=4;
                n+=10;
                Getreg (RTCADD, n, &a);
                Getreg (RTCADD, n+1, &b);
                Getreg (RTCADD, n+2, &c);
                Getreg (RTCADD, n+3, &d);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
if (a!=0xff)
        {
                initlcd ();
                sprintf (ASCBUF, "START TIME %02bx:%02bx", a, b);
                text (ASCBUF);
                lcdwi (0xc0);
                sprintf (ASCBUF, " STOP TIME %02bx:%02bx", c, d);
                text (ASCBUF);
                lcdwi (0x0f);                           /* on cursor */
                lcdwi (0x8b);

if (readkbd ()!=k_ent)
                return;
}
        initlcd ();
        text ("START TIME __:__");
        lcdwi (0xc0);
        text (" STOP TIME __:__");
        lcdwi (0x0f);                                   /* on cursor */
        lcdwi (0x8b);
        b=0;
        while (1)
        {
                a=readkbd ();
        if (a==k_cancel)
                break;
        if (a==k_ent)
        {
                Putreg (RTCADD, n, 0xff);
                Putreg (RTCADD, n+1, 0xff);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            Putreg (RTCADD, n+2, 0xff);
            Putreg (RTCADD, n+3, 0xff);
            initlcd ();
            text ("   Clear time   ");
            lcdwi (0xc0);
            text ("       OK       ");
            delay (60000);
            break;
    }
    if ((a>=0) && (a<=9))
    {
            ASCBUF[b]=a;
            b++;
            a+=0x30;
            lcdwd (a);
    if (b==2)
            lcdwi (0x8e);
    if (b==4)
            lcdwi (0xcb);
    if (b==6)
            lcdwi (0xce);
    if (b==8)
   {
            ASCBUF[0]<<=4;
            ASCBUF[0]|=ASCBUF[1];

            ASCBUF[2]<<=4;
            ASCBUF[2]|=ASCBUF[3];

            ASCBUF[4]<<=4;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                ASCBUF[4]|=ASCBUF[5];

                ASCBUF[6]<<=4;
                ASCBUF[6]|=ASCBUF[7];

                Putreg (RTCADD, n, ASCBUF[0]);
                Putreg (RTCADD, n+1, ASCBUF[2]);
                Putreg (RTCADD, n+2, ASCBUF[4]);
                Putreg (RTCADD, n+3, ASCBUF[6]);

                break;
            }
        }
    }
}

                /************************/
                /*SetLoad function*/
                /************************/

void Setload (void)
    {
    while (1)
    {
            initlcd ();
            text ("  SELECT LOAD  ");
            lcdwi (0xc0);
            text ("      1-8    ");
            a=readkbd ();
    if (a==k_cancel)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                break;
        if ((a>=1) && (a<=8))
        Loadtime (a);
        }
}

                        /************************/
                        /*CSettime function*/
                        /************************/

void CSet (void)
        {
        while (1)
        {
                initlcd ();
                text ("1.SET TIME");
                lcdwi (0xc0);
                text ("2.SET LOAD");
                a=readkbd ();
                switch (a)
                {
        case    1:
                Settime ();
                break;

        case    2:
                Setload ();
                break;

        case    k_cancel:
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                initlcd ();
                return;
        }
    }
}

/**********************/
/*main program*/
/**********************/

void main (void)
    {
        lcdrs=lcdrw=lcden=0;                    /* LCD set */
        SCL    = 0;                             /* Excite I2C line */
        SDA    = 0;
        delay (256);
        SCL    = 1;
        SDA    = 1;
        delay (256);
        initlcd ();
        P2=0;
        text ("    KMITL.   ");
        lcdwi (0xc0);
        text ("  ENGINEERING. ");
        delay (60000);
        delay (60000);
        delay (60000);

        Getreg (RTCADD, 9, &a);                 /* Check first RTC */
    if  (a!=0x99)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
{
        RTC.sec=0x00;                          /* default RTC to my birth day */
        RTC.min=0x30;
        RTC.hour=0x03;
        RTC.date=0x23;
        RTC.month=0x11;
        RTC.year=0x02;

        SetRTC ();                             /* set to RTC */
        Putreg (RTCADD, 9, 0x99);              /* Put first time flag */
}
        initlcd ();

while (1)                                      /* never die loop */
{
if (!GetRTC ())
{
        sprintf (ASCBUF, "%02bx/%02bx/%02bx", RTC.date, RTC.month, RTC.year);
        lcdwi (0x88);
        text (ASCBUF);
        sprintf (ASCBUF, "%02bx:%02bx:%02bx", RTC.hour, RTC.min, RTC.sec);
        lcdwi (0xc8);
        text (ASCBUF);

for (a=0; a<16; a++)                           /* Check alarm */
{
        b=a*2;
        b+=10;
        Getreg (RTCADD, b, &TimeHour);         /* Get alarm */
        Getreg (RTCADD, b+1, &TimeMin);        /* Get alarm */
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            if ((RTC.hour!=TimeHour) || (RTC.min!=TimeMin))
                continue;
                b=a/2;
            if ((a&0x01)==0)
                P2|=tbl1[b];
            else
            {
                c=tbl1[b];
                c^=0xff;
                P2&=c;
            }
        }

//          TimeHour=RTC.hour;
//          TimeMin=RTC.min;

        }
        if (kbhit()==TRUE)                          /* If key press */
        {
            a=readKbd ();                           /* Get keyboard */
        if (a==0)
            CSet ();
            initlcd ();
        }
    }
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คู่มือการใช้งาน

1. เมื่อทำการเปิดสวิตช์ เครื่องจะทำการแสดงเวลาและวันที่ขึ้นมา ที่จอแสดงผล

2. กดปุ่มหมายเลข 0 เพื่อเลือกว่าจะทำการตั้งค่าเวลาใหม่ หรือจะทำการตั้งเวลาการเปิด-ปิด อุปกรณ์ไฟฟ้า โดยจอจะขึ้นคำว่า 1.SET TIME กับ 2.SET LOAD

3. หลังจากกดเลข 0 แล้วถ้าต้องการตั้งค่าเวลาใหม่ ให้กดปุ่มหมายเลข 1 โดยจอแสดงผลขะขึ้นคำว่า TIME = __:__ และ DATE = __/__/__ จากนั้นป้อนเวลาและวันที่ตามมาตราฐานลงไป โดยที่เวลาจะเป็น ชั่วโมงกับนาที ส่วนวันที่จะเป็น วันที่ / เดือนเป็นตัวเลข / และปี ค.ศ. 2 ตัวท้าย

4. ถ้าต้องการตั้งเวลาการเปิด-ปิด อุปกรณ์ไฟฟ้า ให้กดปุ่มหมายเลข 2 โดยจอจะขึ้นคำว่า SELECT LOAD 1-8

5. เลือกหมายเลขของอุปกรณ์ไฟฟ้าที่ต้องการจะตั้งเวลาการเปิด-ปิด (มี 1-8) เสร็จแล้วจอแสดงผลจะขึ้นคำว่า START TIME __:__ และ STOP TIME __:__

6. ป้อนเวลาที่ต้องการให้อุปกรณ์ไฟฟ้านั้นๆเปิด โดยยึดตามเวลาจริงเป็นหลัก ที่ตรง STRAT TIME

7. ป้อนเวลาที่ต้องการให้อุปกรณ์ไฟฟ้านั้นๆปิด โดยยึดตามเวลาจริงเป็นหลัก ที่ตรง STOP TIME

8. หลังจากป้อนเวลาเสร็จแล้ว จอแสดงผลจะขึ้นคำว่า SELECT LOAD 1-8 เพื่อให้เลือกหมายเลขของอุปกรณ์ที่จะต้องการจะตั้งเวลาการเปิด-ปิดของอุปกรณ์ไฟฟ้าอีก จากนั้นก็ทำตามข้อ 6 กับข้อ 7

9. ถ้าไม่ต้องการที่จะตั้งเวลาการเปิด-ปิดอุปกรณ์แล้ว ก็กดปุ่ม BACK กลับไปยังหน้าแรกเพื่อคอยเวลาเปิดและปิดของอุปกรณ์ตามที่ได้ตั้งไว้

<u>หมายเหตุ</u>

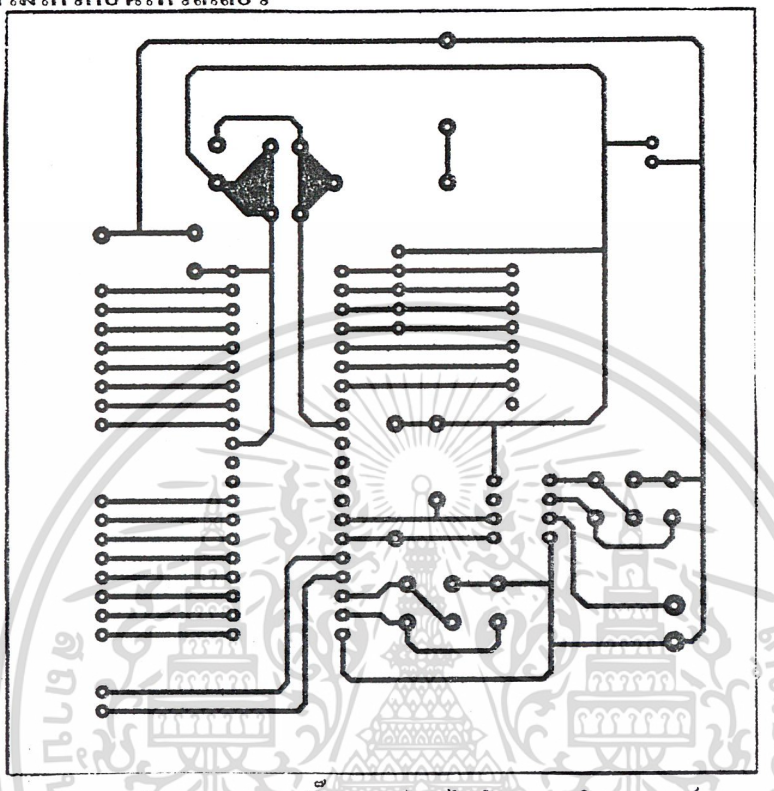
1. ถ้าในกรณีที่ไม่อยากทำภารกิจในหน้านั้นๆ ก็สามารถใช้ปุ่ม BACK เพื่อย้อนกลับไปหน้าเก่า ก่อนที่จะเข้ามาหน้านี้ได้

2. ในกรณีที่ผู้ใช้งานต้องการจะตั้งเวลาเปิด ปิดอุปกรณ์ไฟฟ้านั้น บางครั้งอาจจะมีเวลาที่ตั้งเอาไว้ ก่อนๆอยู่ สามารถลบได้ด้วย ปุ่ม DEL แล้วถึงจะป้อนเวลาเปิด ปิดเข้าไป แต่ถ้าอยากจะให้เวลานั้นหายไปเลย ก็กดปุ่ม DEL 2 ครั้ง เท่านี้เวลาที่ขึ้นมาก่อนๆ ก็จะหายไป โดยที่หน้าจอจะขึ้นคำว่า Clear time OK

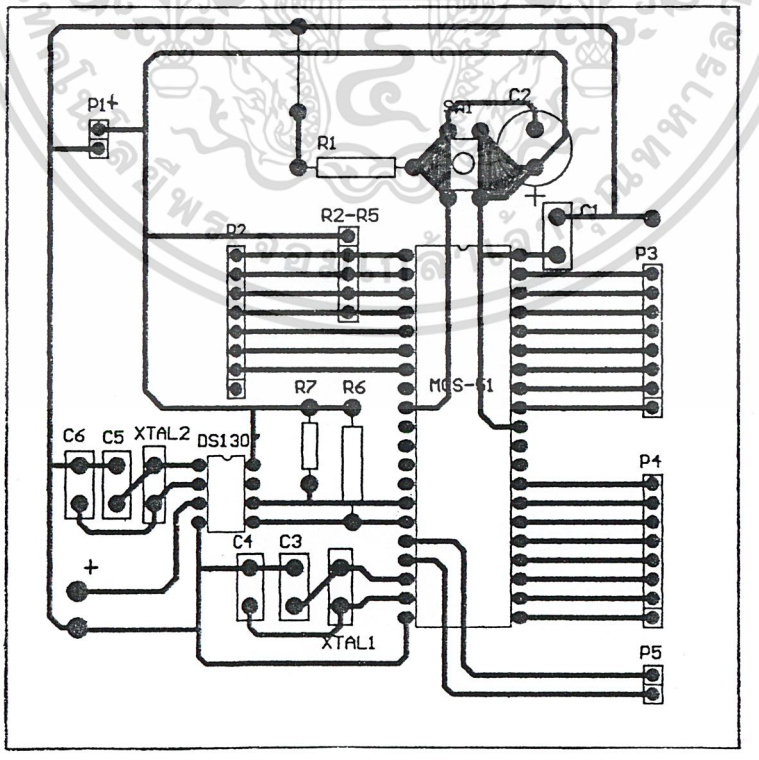
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปแสดงลาย PCB และการลงอุปกรณ์ของส่วนต่างๆ

1. ส่วนของไมโครคอนโทรลเลอร์

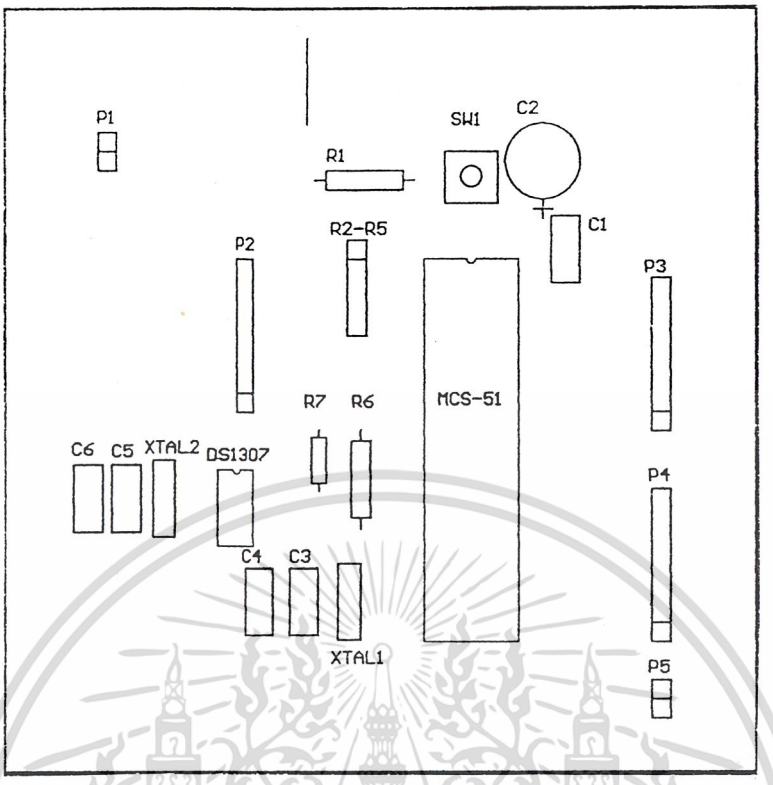


รูป ก. แสดงลายปริ๊นของส่วนไมโครคอนโทรลเลอร์



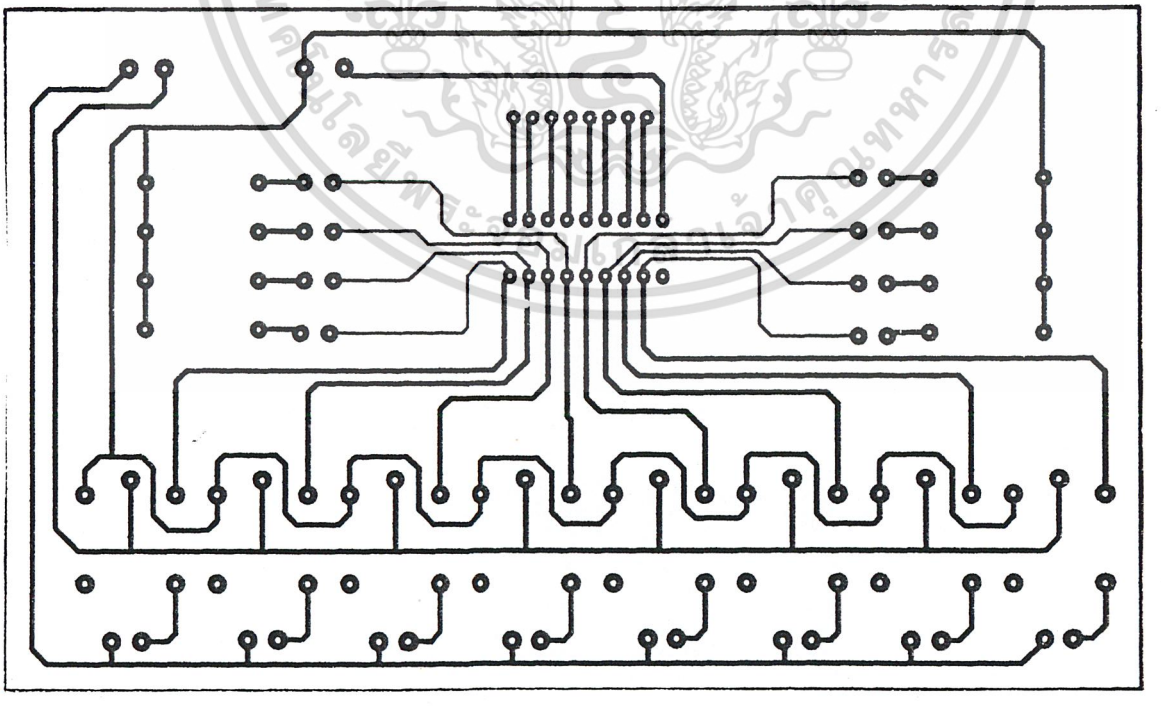
รูป ข. แสดงการลงอุปกรณ์ของส่วนไมโครคอนโทรลเลอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



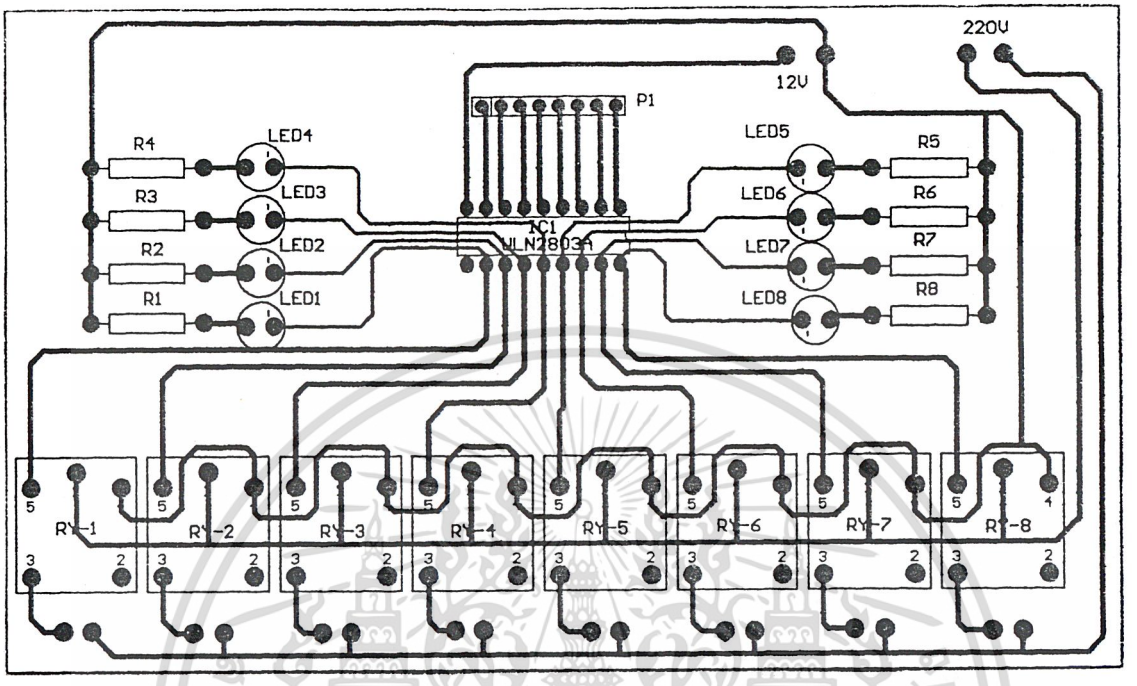
รูป ค. แสดงตำแหน่งของการวางอุปกรณ์ของส่วนไมโครคอนโทรลเลอร์

2. ส่วนควบคุมอุปกรณ์ไฟฟ้า

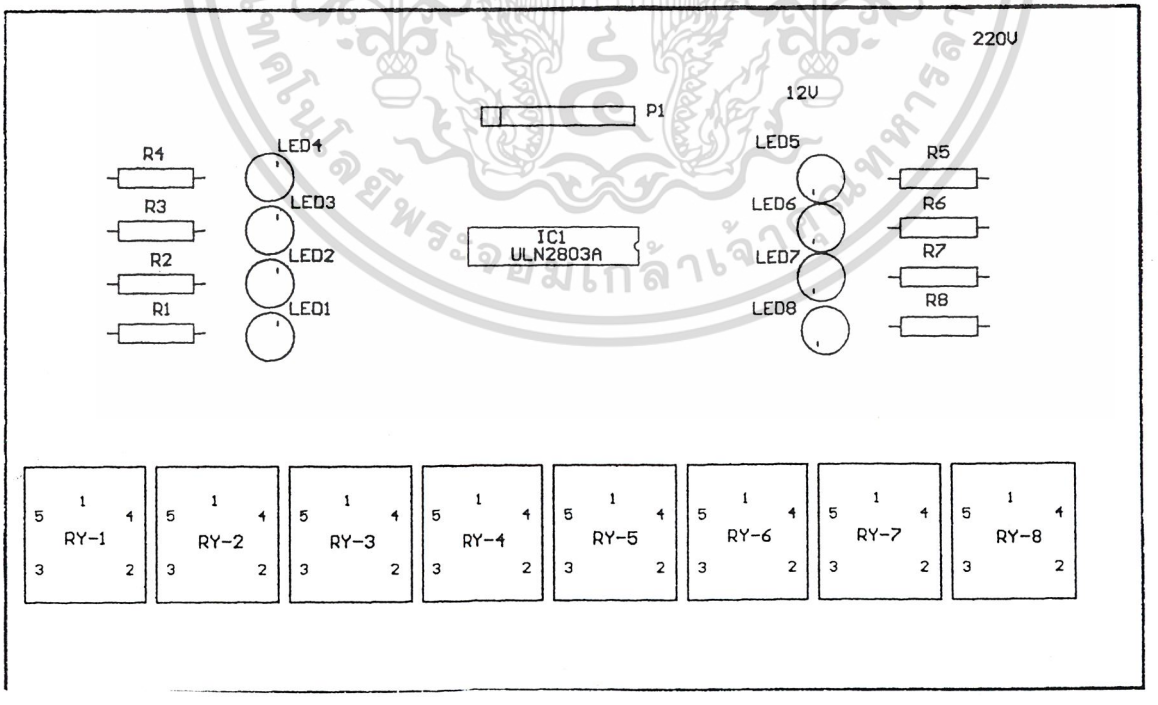


รูป ง. แสดงลายปริ๊นของส่วนควบคุมอุปกรณ์ไฟฟ้า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



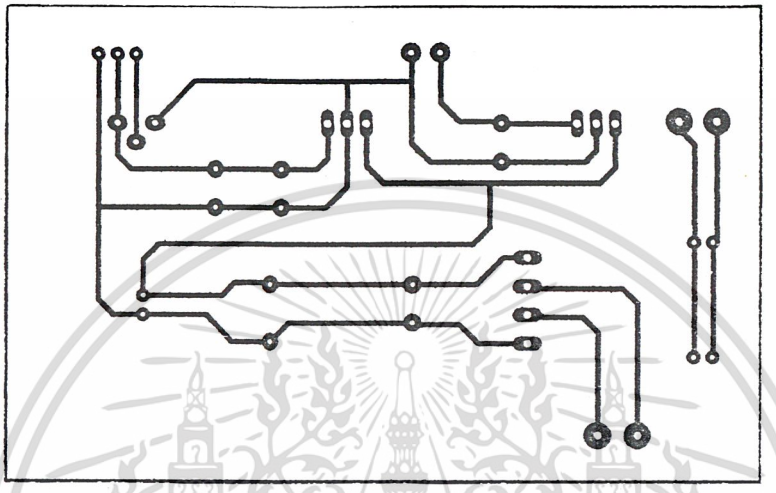
รูป จ. แสดงการลงอุปกรณ์ของส่วนควบคุมอุปกรณ์ไฟฟ้า



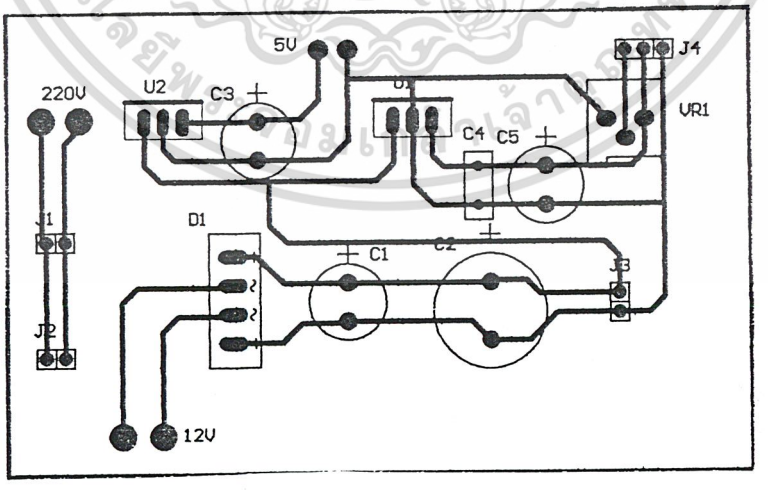
รูป ฉ. แสดงตำแหน่งของการวางอุปกรณ์ของส่วนควบคุมอุปกรณ์ไฟฟ้า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. ส่วนภาคจ่ายไฟ

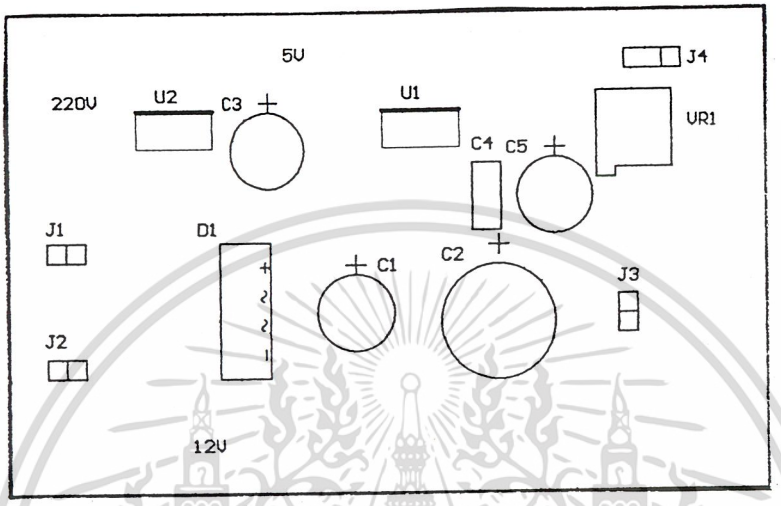


รูป ช. แสดงลายปริ๊นของส่วนภาคจ่ายไฟ



รูป ฆ. แสดงการลงอุปกรณ์ของส่วนภาคจ่ายไฟ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูป ฌ. แสดงตำแหน่งของการวางอุปกรณ์ของส่วนภาคจ่ายไฟ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Features

- Compatible with MCS-51™ Products
- 8K bytes of In-System Reprogrammable Downloadable Flash Memory
  - SPI Serial Interface for Program Downloading
  - Endurance: 1,000 Write/Erase Cycles
- 2K bytes EEPROM
  - Endurance: 100,000 Write/Erase Cycles
- 2.7V to 6V Operating Range
- Fully Static Operation: 0 Hz to 24 MHz
- Three-Level Program Memory Lock
- 256 x 8 bit Internal RAM
- 32 Programmable I/O Lines
- Three 16 bit Timer/Counters
- Nine Interrupt Sources
- Programmable UART Serial Channel
- SPI Serial Interface
- Low Power Idle and Power Down Modes
- Interrupt Recovery From Power Down
- Programmable Watchdog Timer
- Dual Data Pointer
- Power Off Flag

8 bit Microcontroller with 8K bytes Flash

AT89S8252 Preliminary

Description

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

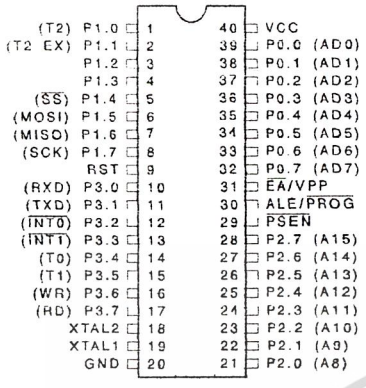
The AT89S8252 provides the following standard features: 8K bytes of Downloadable Flash, 2K bytes EEPROM, 256 bytes of RAM, 32 I/O lines, programmable watchdog timer, two Data Pointers, three 16 bit timer/counters, a six-vector two-level interrupt architecture, a full duplex serial port, on-chip oscillator,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

The Downloadable Flash can be changed a single byte at a time and is accessible through the SPI serial interface. Holding RESET active forces the SPI bus into a slave input mode and allows the program memory to be written-from or read-to unless Lock Bit 3 has been activated.

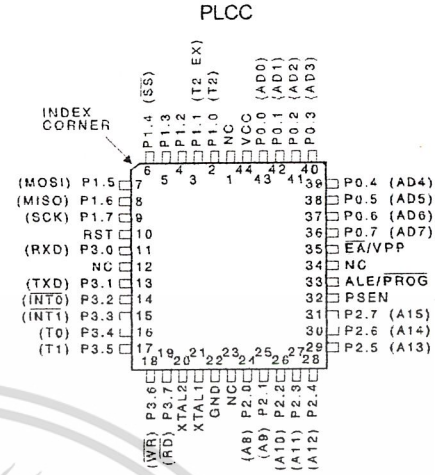
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Pin Configurations

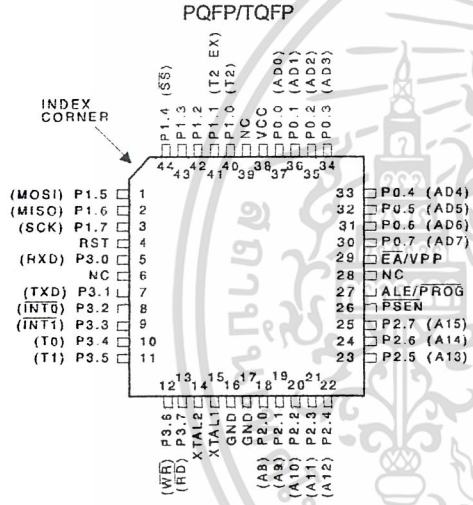
PDIP



PLCC



PQFP/TQFP



Pin Description

VCC

Supply voltage.

GND

Ground.

Port 0

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

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

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

Port 1

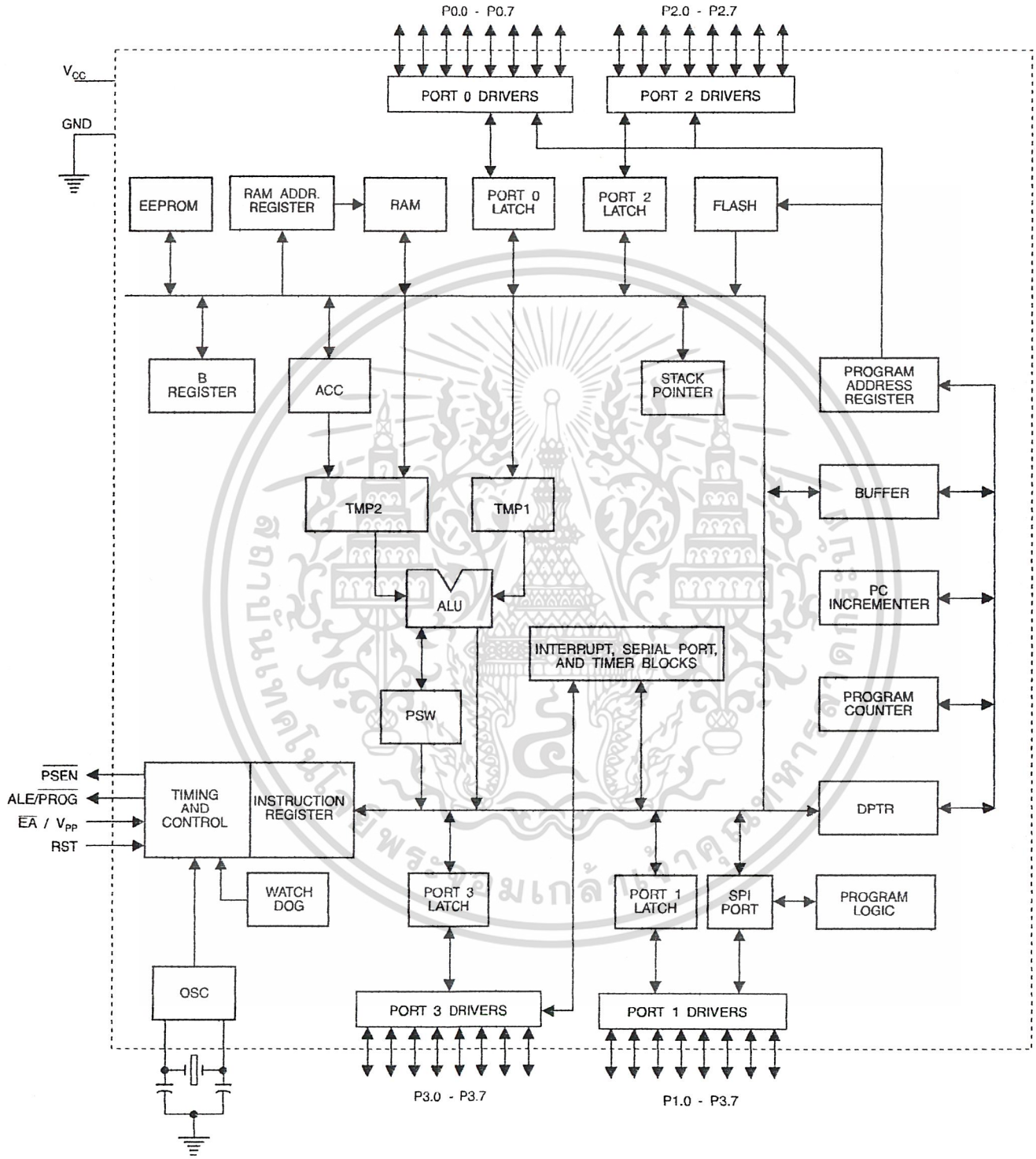
Port 1 is an 8 bit bidirectional I/O port with internal pullups. The Port 1 output buffers can sink/source four TTL inputs. When 1s are written to Port 1 pins, they are pulled high by the internal pullups and can be used as inputs. As inputs, Port 1 pins that are externally being pulled low will source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

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

(continued)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Block Diagram



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Pin Description (Continued)

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

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

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

Port 2

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

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

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

Port 3

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

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

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

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

RST

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

ALE/$\overline{PROG}$

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

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

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

$\overline{PSEN}$

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

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

$\overline{EA}$/V_{PP}

External Access Enable. $\overline{EA}$ must be strapped to GND in order to enable the device to fetch code from external program memory locations starting at 0000H up to FFFFH. Note, however, that if lock bit 1 is programmed, $\overline{EA}$ will be internally latched on reset.

$\overline{EA}$ should be strapped to V_{CC} for internal program executions.

(continued)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Pin Description (Continued)

This pin also receives the 12-volt programming enable voltage (V_{PP}) during Flash programming when 12-volt programming is selected.

XTAL1

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

XTAL2

Output from the inverting oscillator amplifier.

Special Function Registers

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

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

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

(continued)

Table 1. AT89S8252 SFR Map and Reset Values

0F8H									0FFH
0F0H	B 00000000								0F7H
0E8H									0EFH
0E0H	ACC 00000000								0E7H
0D8H									0DFH
0D0H	PSW 00000000					SPCR 000001XX			0D7H
0C8H	T2CON 00000000	T2MOD XXXXXXX0	RCAP2L 00000000	RCAP2H 00000000	TL2 00000000	TH2 00000000			0CFH
0C0H									0C7H
0B8H	IP XX000000								0BFH
0B0H	P3 11111111								0B7H
0A8H	IE 0X000000		SPSR 00000000						0AFH
0A0H	P2 11111111								0A7H
98H	SCON 00000000	SBUF XXXXXXXX							9FH
90H	P1 11111111						WMCON 00000010		97H
88H	TCON 00000000	TMOD 00000000	TL0 00000000	TL1 00000000	TH0 00000000	TH1 00000000			8FH
80H	P0 11111111	SP 00000111	DP0L 00000000	DP0H 00000000	DP1L 00000000	DP1H 00000000	SPDR XXXXXXXX	PCON 0XXX0000	87H

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Special Function Registers (Continued)

Timer 2 Registers Control and status bits are contained in registers T2CON (shown in Table 2) and T2MOD (shown in Table 4) for Timer 2. The register pair (RCAP2H, RCAP2L) are the Capture/Reload registers for Timer 2 in 16 bit capture mode or 16 bit auto-reload mode.

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

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

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

(continued)

Table 2. T2CON—Timer/Counter 2 Control Register

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

Bit Addressable

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

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

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Special Function Registers (Continued)

Dual Data Pointer Registers To facilitate accessing both internal EEPROM and external data memory, two banks of 16 bit Data Pointer Registers are provided: DP0 at SFR address locations 82H-83H and DP1 at 84H-85H. Bit DPS = 0 in SFR WMCON selects DP0 and DPS = 1 selects DP1.

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

Table 3. WMCON—Watchdog and Memory Control Register

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

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

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

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Table 4. SPCR—SPI Control Register

SPCR Address = D5H — Reset Value = 0000 01XXB

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

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

Table 5. SPSR—SPI Status Register

SPCR Address = AAH — Reset Value = 0000 0000B

	SPIF	WCOL	—	—	—	—	—	—
Bit	7	6	5	4	3	2	1	0

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

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Table 6. SPDR—SPI Data Register

SPDR Address = 86H — Reset Value = unchanged

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

Data Memory—EEPROM and RAM

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

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

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

MOV 0A0H, #data

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

MOV @R0, #data

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

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

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

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

Programmable Watchdog Timer

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

Table 7. Watchdog Timer Period Selection

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

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

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Timer 0 and 1

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

Timer 2

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

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

In the Counter function, the register is incremented in response to a 1-to-0 transition at its corresponding external input pin, T2. In this function, the external input is sampled during S5P2 of every machine cycle. When the samples show a high in one cycle and a low in the next cycle, the count is incremented. The new count value appears in the register during S3P1 of the cycle following the one in which the transition was detected. Since two machine cycles (24 oscillator periods) are required to recognize a 1-to-0 transition, the maximum count rate is 1/24 of the oscillator frequency. To ensure that a given level is sampled at least once before it changes, the level should be held for at least one full machine cycle.

Table 8. Timer 2 Operating Modes

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

Capture Mode

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

Auto-Reload (Up or Down Counter)

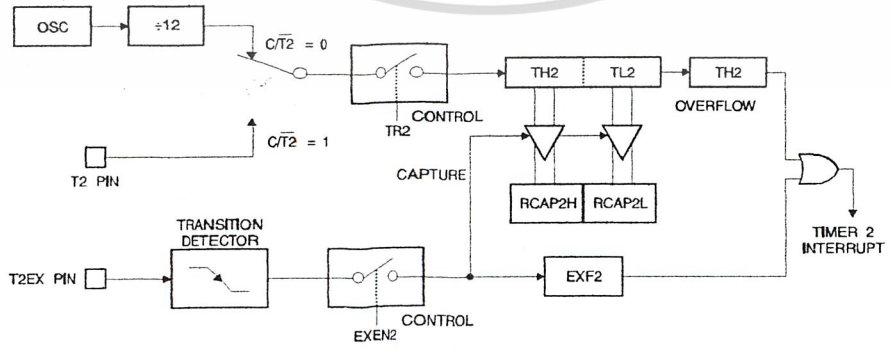
Timer 2 can be programmed to count up or down when configured in its 16 bit auto-reload mode. This feature is invoked by the DCEN (Down Counter Enable) bit located in the SFR T2MOD (see Table 9). Upon reset, the DCEN bit is set to 0 so that timer 2 will default to count up. When DCEN is set, Timer 2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

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

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

(continued)

Figure 1. Timer 2 in Capture Mode



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Auto-Reload (Up or Down Counter) (Continued)

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

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

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

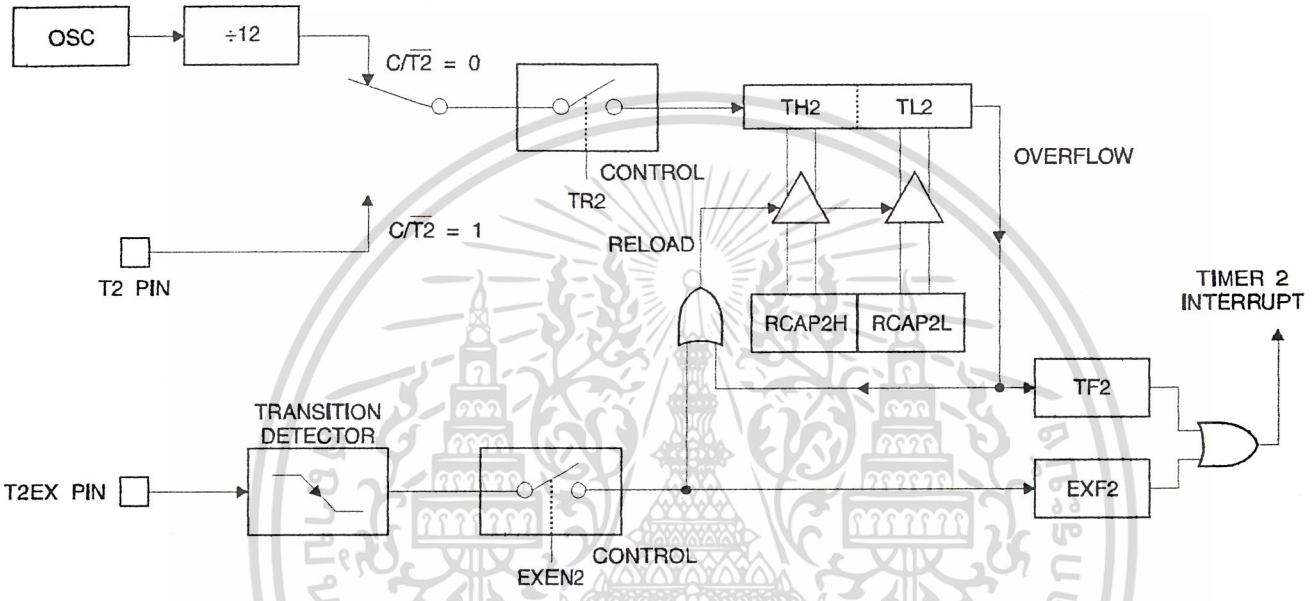


Table 9. T2MOD—Timer 2 Mode Control Register

T2MOD Address = 0C9H — Reset Value = XXXX XX00B

Not Bit Addressable

	—	—	—	—	—	—	T2OE	DCEN
Bit	7	6	5	4	3	2	1	0

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

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

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

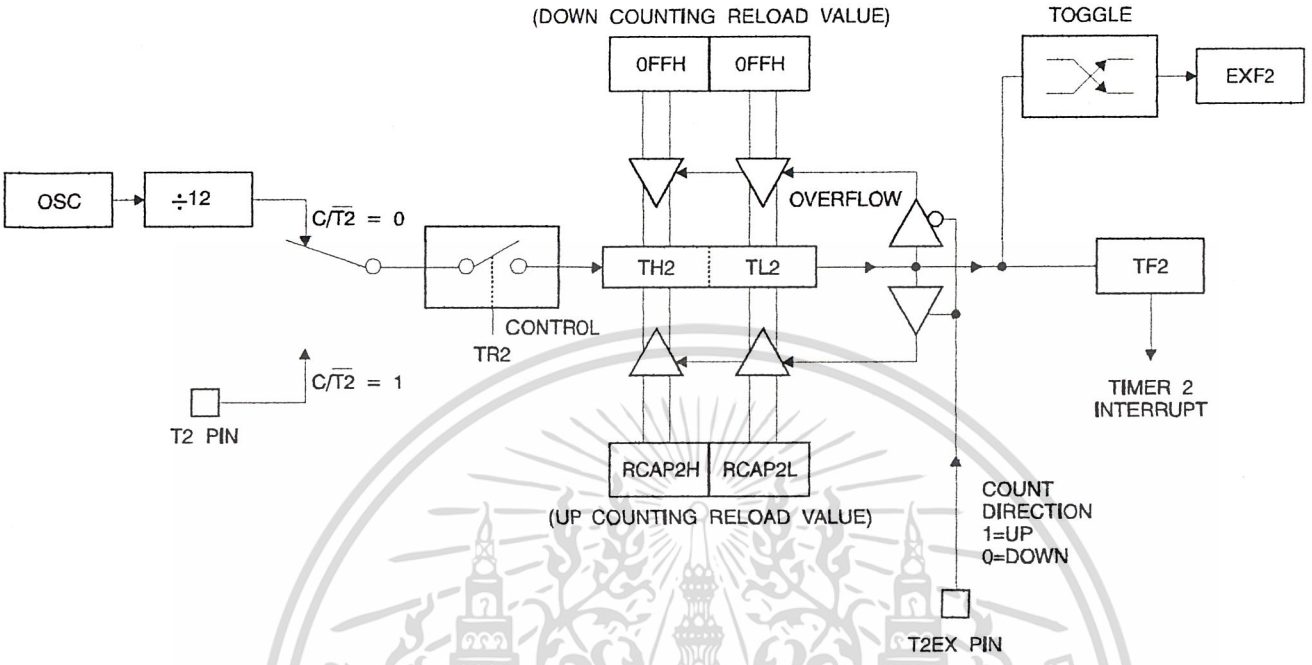
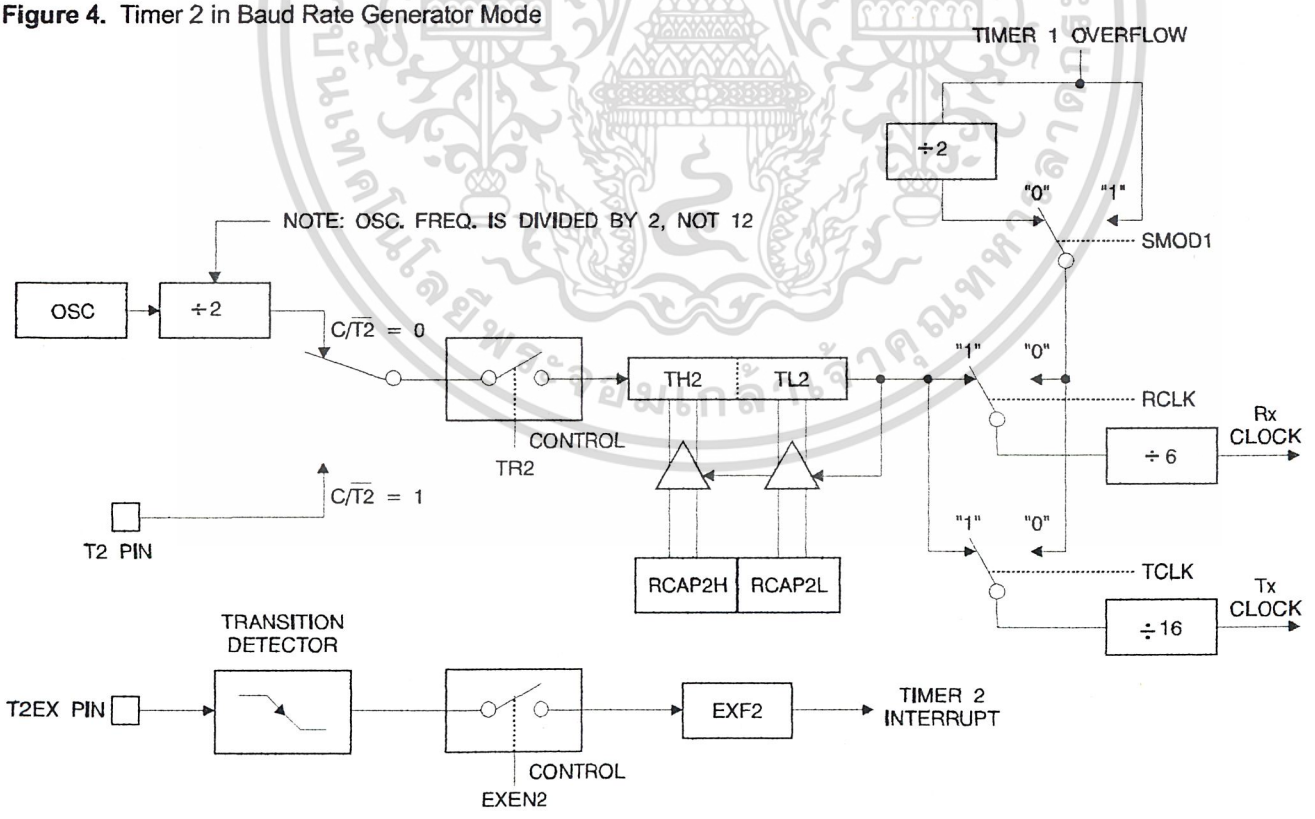


Figure 4. Timer 2 in Baud Rate Generator Mode



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Baud Rate Generator

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

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

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

$$\text{Modes1 and 3 Baud Rates} = \frac{\text{Timer2 OverflowRate}}{16}$$

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

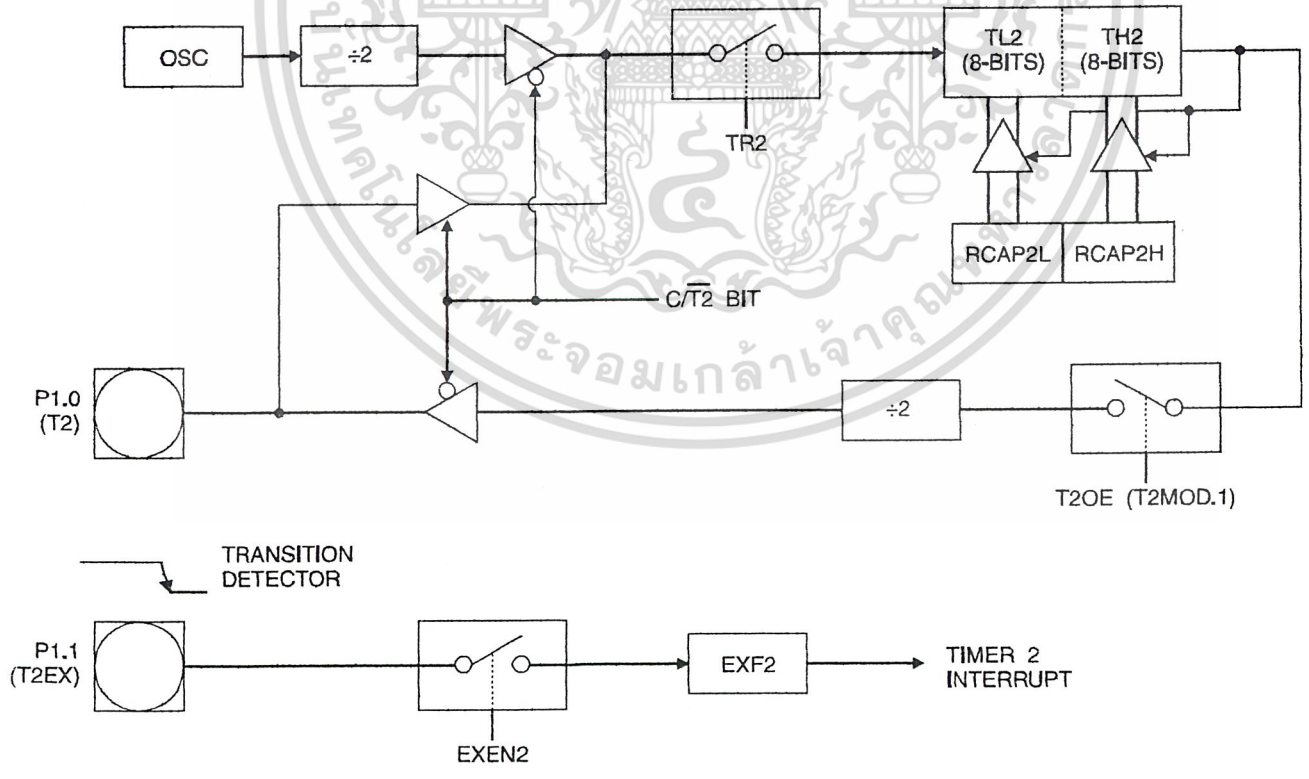
$$\frac{\text{Modes1 and 3}}{\text{Baud Rate}} = \frac{\text{Oscillator Frequency}}{32 \times [65536 - (\text{RCAP2H, RCAP2L})]}$$

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

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

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

Figure 5. Timer 2 in Clock-Out Mode



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Programmable Clock Out

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

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

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

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

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

UART

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

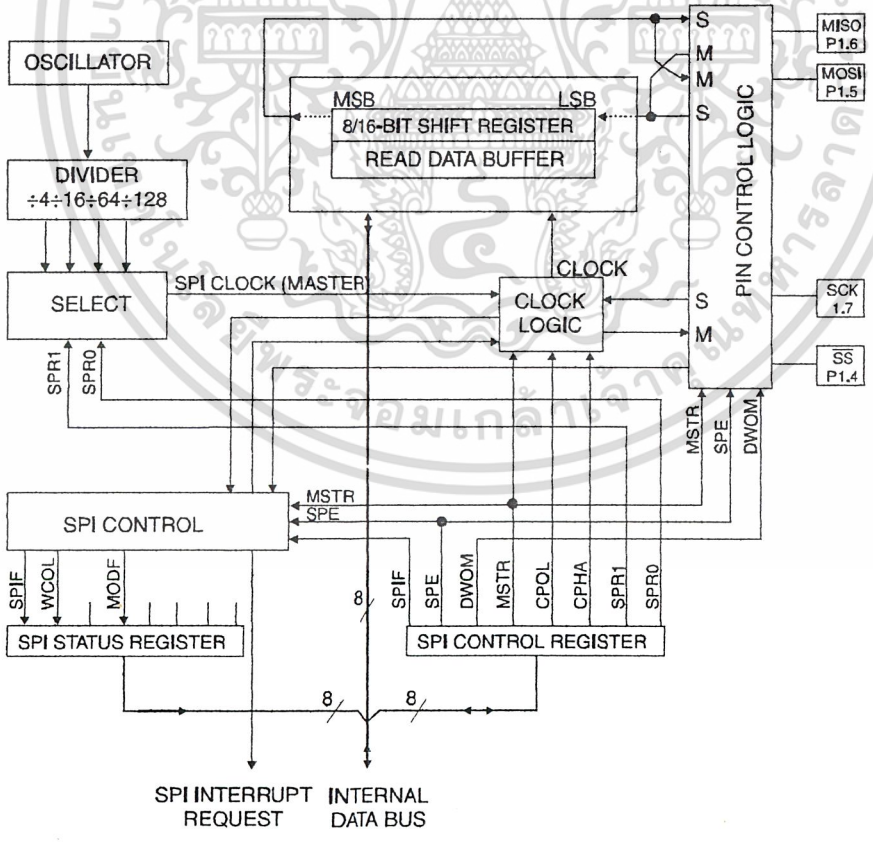
Serial Peripheral Interface

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

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

(continued)

Figure 6. SPI Block Diagram



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Serial Peripheral Interface (Continued)

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

The Slave Select input, $\overline{SS}$/P1.4, is set low to select an individual SPI device as a slave. When $\overline{SS}$/P1.4 is set high, the SPI port is deactivated and the MOSI/P1.6 pin can be used as an input.

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

Figure 7. SPI Master -Slave Interconnection

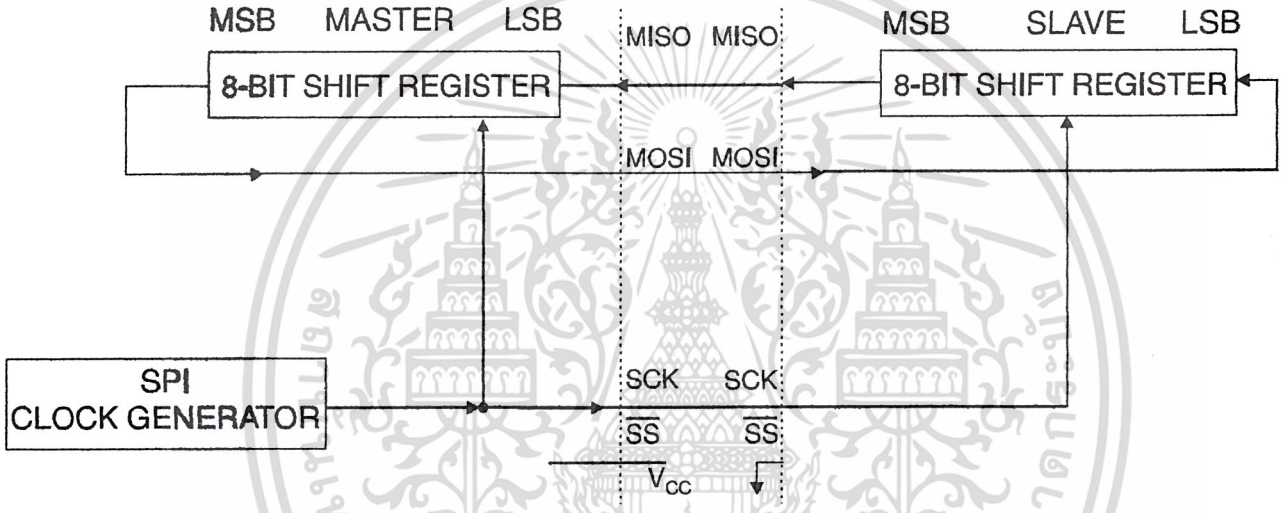
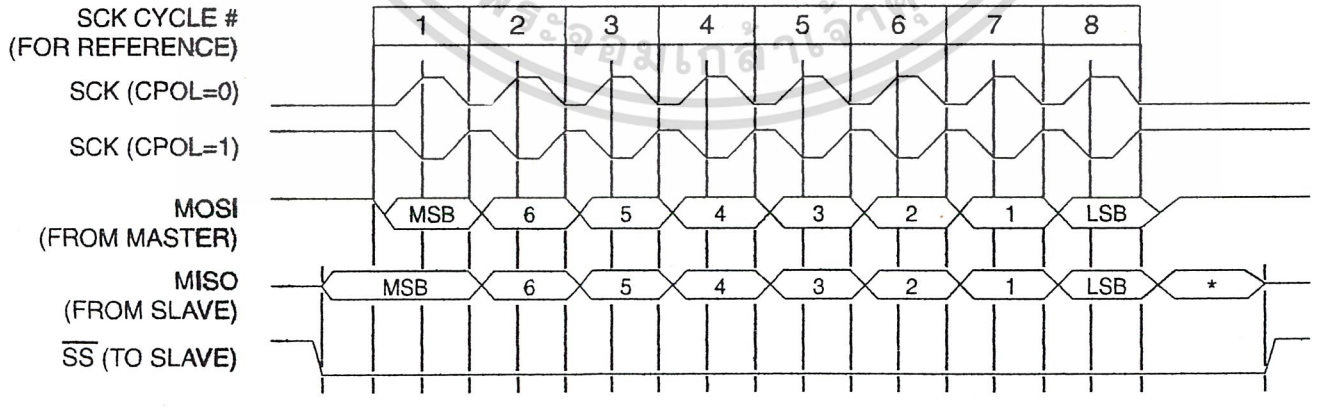


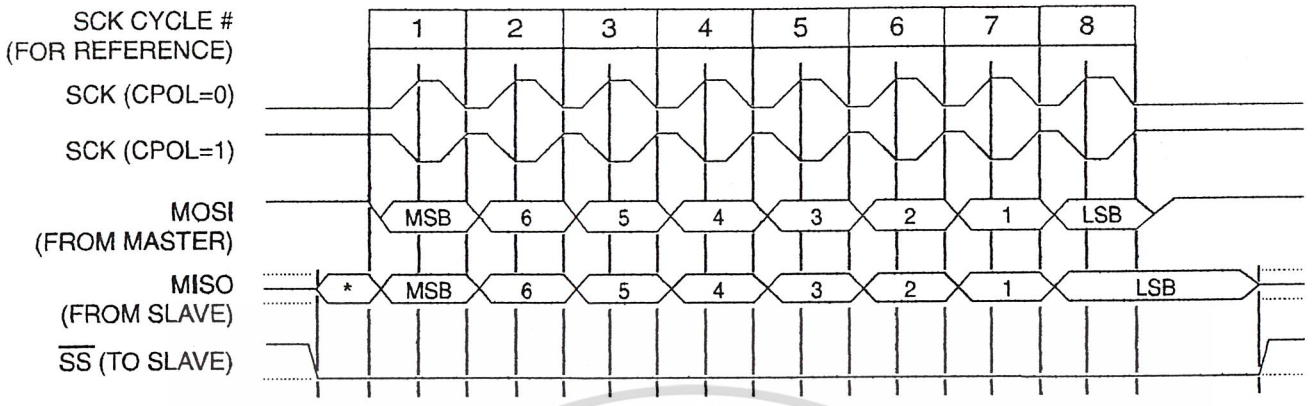
Figure 8. SPI Transfer Format with CPHA = 0



* Not defined but normally MSB of character just received.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Figure 9. SPI Transfer Format with CPHA = 1



* Not defined but normally LSB of previously transmitted character.

Interrupts

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

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

Note that Table 10 shows that bit position IE.6 is unimplemented. In the AT89C51, AT89C52 and AT89C55, bit position IE.5 is also unimplemented. User software should not write 1s to these bit positions, since they may be used in future AT89 products.

Table 10. Interrupt Enable (IE) Register

(MSB)							(LSB)
EA	—	ET2	ES	ET1	EX1	ET0	EX0

Enable Bit = 1 enables the interrupt.

Enable Bit = 0 disables the interrupt.

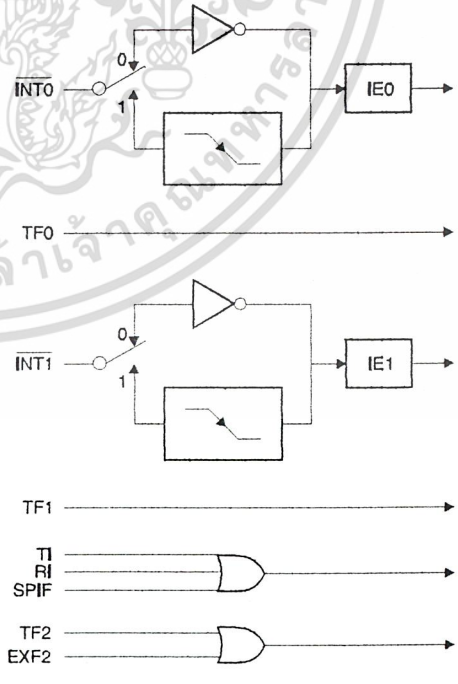
Symbol	Position	Function
EA	IE.7	Disables all interrupts. If EA = 0, no interrupt is acknowledged. If EA = 1, each interrupt source is individually enabled or disabled by setting or clearing its enable bit.
—	IE.6	Reserved.
ET2	IE.5	Timer 2 interrupt enable bit.
ES	IE.4	SPI and UART interrupt enable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

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

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

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

Figure 10. Interrupt Sources

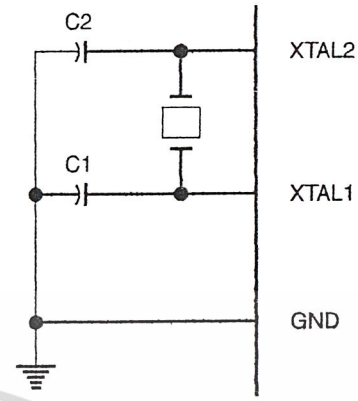


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Oscillator Characteristics

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

Figure 11. Oscillator Connections



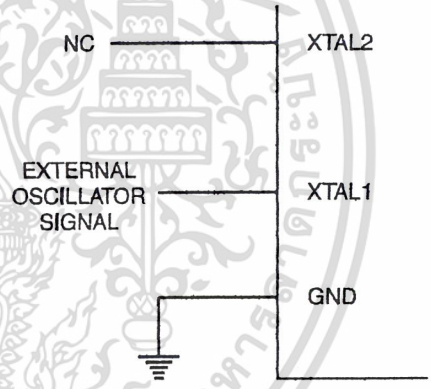
Notes: C1, C2 = 30 pF ± 10 pF for Crystals
= 40 pF ± 10 pF for Ceramic Resonators

Idle Mode

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

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

Figure 12. External Clock Drive Configuration



Power Down Mode

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

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

Status of External Pins During Idle and Power Down

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

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Program Memory Lock Bits

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

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

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

Lock Bit Protection Modes (1, 2)

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

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

Programming the Flash and EEPROM

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

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

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

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

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

Parallel Programming Algorithm

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

1. Power-up sequence:
   Apply power between V_{CC} and GND pins with all other pins floating.
   Set RST pin to 'H'.
   Apply a 4 MHz to 24 MHz clock to XTAL1 pin and wait for
   at least 10 milliseconds.
2. Set $\overline{PSEN}$ pin to 'H'
   ALE pin to 'H'
   $\overline{EA}$ pin to 'H' and all other pins to 'H'.
3. Apply the appropriate combination of 'H' or 'L' logic levels to pins P2.6, P2.7, P3.6, P3.7 to select one of the programming operations shown in the PEROM Programming Modes table.
4. Apply the desired byte address to pins P1.0 to P1.7 and P2.0 to P2.5.
   Apply data to pins P0.0 to P0.7 for Write Code operation.
5. Raise $\overline{EA}$/VPP to 12V to enable Flash programming, erase or verification.
6. Pulse ALE/$\overline{PROG}$ once to program a byte in the Code memory array, the Data memory array or the lock bits. The byte-write cycle is self-timed and typically takes 1.5 ms.

(continued)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Programming the Flash and EEPROM (Continued)

7. To verify the byte just programmed, bring pin P2.7 to 'L' and read the programmed data at pins P0.0 to P0.7.
8. Repeat steps 3 through 7 changing the address and data for the entire 2K or 8K bytes array or until the end of the object file is reached.
9. Power-off sequence:

   Set XTAL1 to 'L'.

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

   Float all other I/O pins.

   Turn V_{CC} power off.

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

$\overline{DATA}$ Polling

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

Ready/$\overline{Busy}$

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

Program Verify

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

Chip Erase

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

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

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

Serial Programming Fuse

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

The AT89S8252 is shipped with the Serial Programming Mode enabled.

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

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

Programming Interface

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

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

Serial Downloading

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

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

The Code and Data memory arrays have separate address spaces:

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

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

(continued)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Serial Downloading (Continued)

24 MHz oscillator clock, the maximum SCK frequency is 600 kHz.

Serial Programming Algorithm

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

1. Power-up sequence:

   Apply power between V_{CC} and GND pins with all other pins floating.

   Set RST pin to 'H'.

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

   Set XTAL1 to 'L' (if a crystal is not used).

   Set RST to 'L'.

   Float all other I/O pins.

   Turn V_{CC} power off.

Serial Programming Instruction

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

Instruction Set

Instruction	Input Format		Operation
	MSB	LSB	
Programming Enable	1010 0101 xxxx	1100 0011 xxxx	Enable serial programming interface after RST goes high.
Chip Erase	1010 xxxx xxxx	1100 x100 xxxx	Chip erase both 8K & 2K memory arrays.
Read Code Memory	aaaa low xxxx	a001 addr xxxx	Read data from Code memory array at the selected address. The 5 MSBs of the first byte are the high order address bits. The low order address bits are in the second byte. Data are available at pin MISO during the third byte.
Write Code Memory	aaaa low data	a010 addr in	Write data to Code memory location at selected address. The address bits are the 5 MSBs of the first byte together with the second byte.
Read Data Memory	00aa low xxxx	a101 addr xxxx	Read data from Data memory array at selected address. Data are available at pin MISO during the third byte.
Write Data Memory	00aa low data	a110 addr in	Write data to Data memory location at selected address.
Write Lock Bits	1010 LLLx BBB 123 xxxx	1100 x111 xxxx	Write lock bits. Set LB1, LB2 or LB3 = '0' to program lock bits.

Notes: 1. $\overline{DATA}$ polling is used to indicate the end of a write cycle, which typically takes less than 2.5 ms.
2. 'aaaaa' = high order address.
3. 'x' = don't care.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Flash and EEPROM Parallel Programming Modes

Mode	RST	$\overline{PSEN}$	ALE/$\overline{PROG}$	$\overline{EA}$/V_{PP}	P2.6	P2.7	P3.6	P3.7	Data I/O P0.7:0	Address P2.5:0 P1.7:0
Serial Prog. Modes	H	h[1]	h	x	(See detail timing)					
Chip Erase	H	L	[2]	12V	H	L	L	L	X	X
Write (10K bytes) Memory	H	L		12V	L	H	H	H	DIN	ADDR
Read (10K bytes) Memory	H	L	H	12V	L	L	H	H	DOUT	ADDR
Write Lock Bits:	H	L		12V	H	L	H	L	DIN	X
Bit - 1									P0.7 = 0	X
Bit - 2									P0.6 = 0	X
Bit - 3									P0.5 = 0	X
Read Lock Bits:	H	L	H	12V	H	H	L	L	DOUT	X
Bit - 1									@P0.2	X
Bit - 2									@P0.1	X
Bit - 3									@P0.0	X
Read Atmel Code	H	L	H	12V	L	L	L	L	DOUT	30H
Read Device Code	H	L	H	12V	L	L	L	L	DOUT	31H
Serial Prog. Enable	H	L	[2]	12V	L	H	L	H	P0.0 = 0	X
Serial Prog. Disable	H	L	[2]	12V	L	H	L	H	P0.0 = 1	X
Read Serial Prog. Fuse	H	L	H	12V	H	H	L	H	@P0.0	X

Notes: 1. 'h' = weakly pulled 'High' internally.
2. Chip Erase and Serial Programming Fuse require a 10 ms PROG pulse. Chip Erase needs to be performed first before reprogramming any byte with a content other than FFH.
3. P3.4 is pulled Low during programming to indicate RDY/$\overline{BSY}$.
4. 'X' = don't care.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Figure 13. Programming the Flash Memory

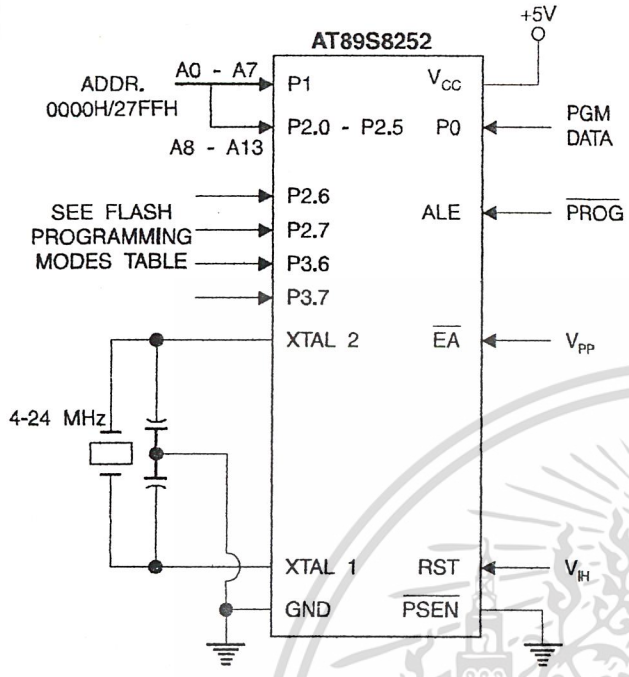


Figure 14. Verifying the Flash Memory

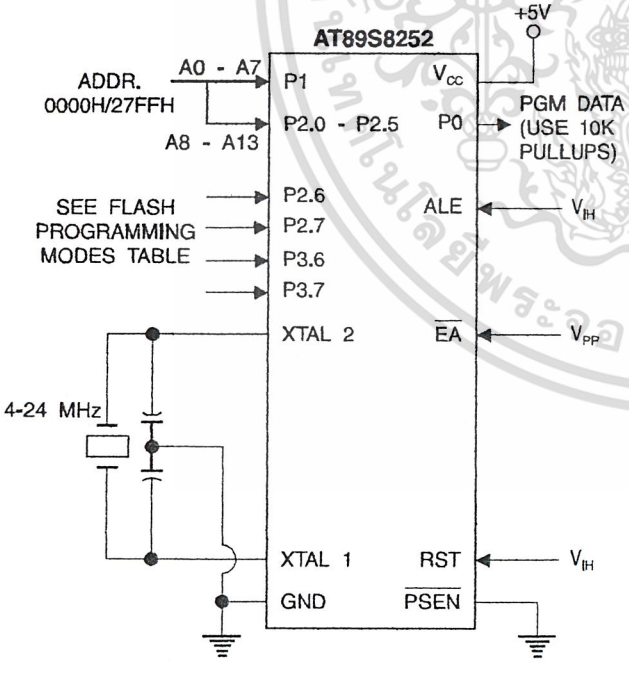
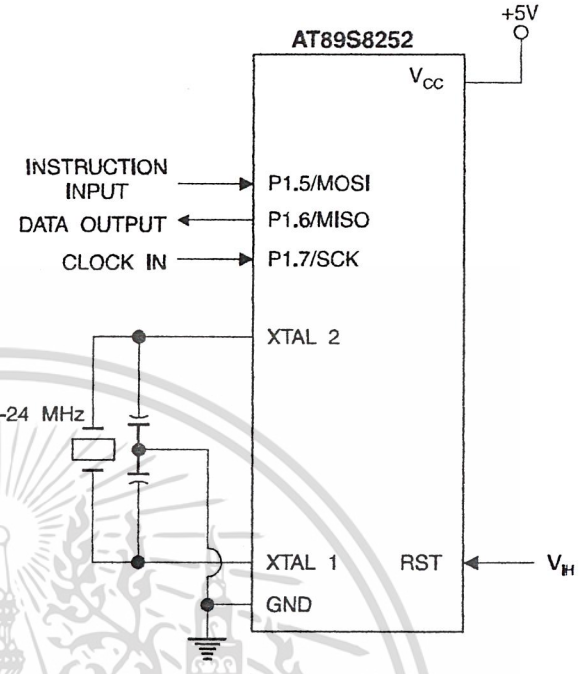


Figure 15. Flash/EEPROM Serial Downloading



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

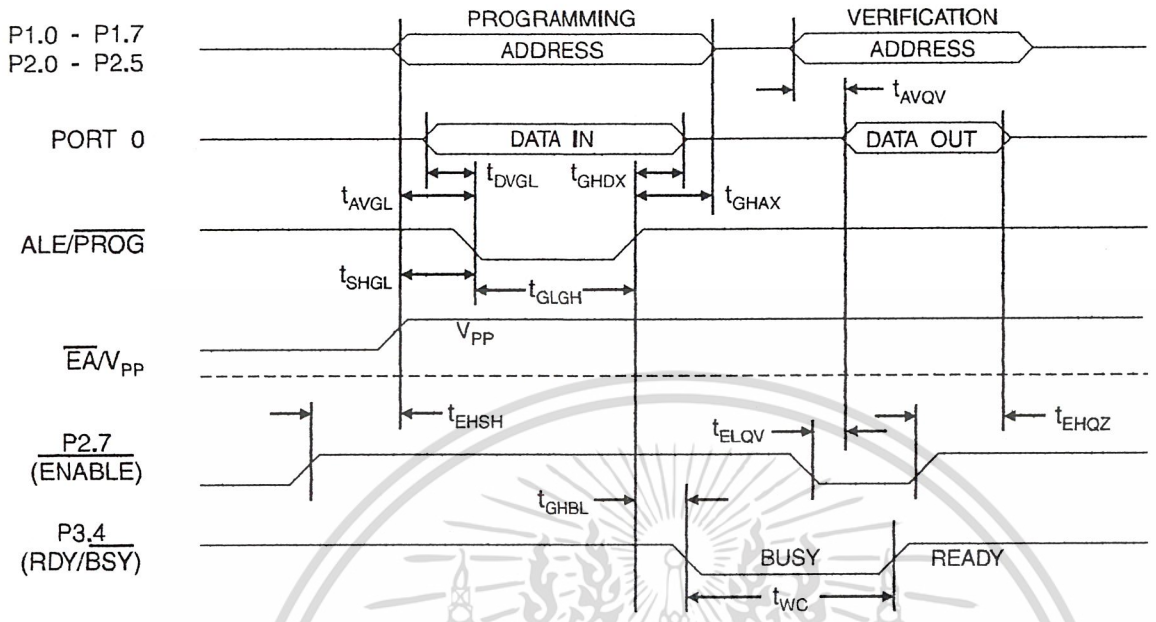
Flash Programming and Verification Characteristics

T_A = 21°C to 27°C, V_{CC} = 5.0V ± 10%

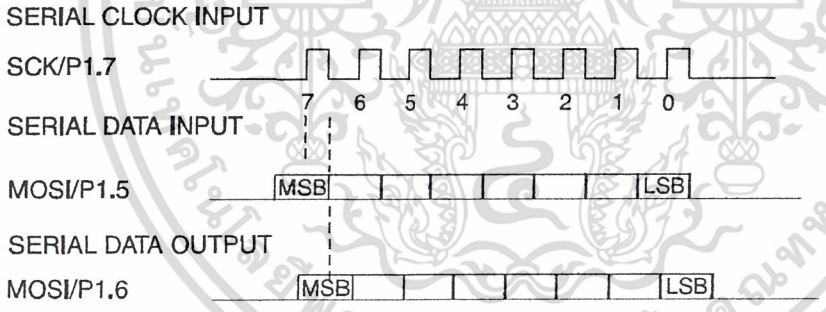
Symbol	Parameter	Min	Max	Units
V_{PP}	Programming Enable Voltage	11.5	12.5	V
I_{PP}	Programming Enable Current		1.0	mA
$1/t_{CLCL}$	Oscillator Frequency	4	24	MHz
t_{AVGL}	Address Setup to $\overline{PROG}$ Low	$48t_{CLCL}$		
t_{GHAX}	Address Hold After $\overline{PROG}$	$48t_{CLCL}$		
t_{DVGL}	Data Setup to $\overline{PROG}$ Low	$48t_{CLCL}$		
t_{GHDX}	Data Hold After $\overline{PROG}$	$48t_{CLCL}$		
t_{EHSH}	P2.7 ($\overline{ENABLE}$) High to V_{PP}	$48t_{CLCL}$		
t_{SHGL}	V_{PP} Setup to $\overline{PROG}$ Low	10		µs
t_{GLGH}	$\overline{PROG}$ Width	1	110	µs
t_{AVQV}	Address to Data Valid		$48t_{CLCL}$	
t_{ELQV}	$\overline{ENABLE}$ Low to Data Valid		$48t_{CLCL}$	
t_{EHQV}	Data Float After $\overline{ENABLE}$	0	$48t_{CLCL}$	
t_{GHBL}	$\overline{PROG}$ High to $\overline{BUSY}$ Low		1.0	µs
t_{WC}	Byte Write Cycle Time		2.0	ms

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Flash/EEPROM Programming and Verification Waveforms - Parallel Mode



Serial Downloading Waveforms



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Absolute Maximum Ratings*

Operating Temperature	-55°C to +125°C
Storage Temperature	-65°C to +150°C
Voltage on Any Pin with Respect to Ground	-1.0V to +7.0V
Maximum Operating Voltage	6.6V
DC Output Current	15.0 mA

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

DC Characteristics

The values shown in this table are valid for T_A = -40°C to 85°C and V_{CC} = 2.7V to 6.0V, unless otherwise noted.

Symbol	Parameter	Condition	Min	Max	Units
V_{IL}	Input Low Voltage	(Except $\overline{EA}$)	-0.5	0.2 V_{CC} - 0.1	V
V_{IL1}	Input Low Voltage ($\overline{EA}$)		-0.5	0.2 V_{CC} - 0.3	V
V_{IH}	Input High Voltage	(Except XTAL1, RST)	0.2 V_{CC} + 0.9	V_{CC} + 0.5	V
V_{IH1}	Input High Voltage	(XTAL1, RST)	0.7 V_{CC}	V_{CC} + 0.5	V
V_{OL}	Output Low Voltage (1) (Ports 1,2,3)	I_{OL} = 1.6 mA		0.5	V
V_{OL1}	Output Low Voltage (1) (Port 0, ALE, $\overline{PSEN}$)	I_{OL} = 3.2 mA		0.5	V
V_{OH}	Output High Voltage (Ports 1,2,3, ALE, $\overline{PSEN}$)	I_{OH} = -60 µA, V_{CC} = 5V ± 10%	2.4		V
		I_{OH} = -25 µA	0.75 V_{CC}		V
		I_{OH} = -10 µA	0.9 V_{CC}		V
V_{OH1}	Output High Voltage (Port 0 in External Bus Mode)	I_{OH} = -800 µA, V_{CC} = 5V ± 10%	2.4		V
		I_{OH} = -300 µA	0.75 V_{CC}		V
		I_{OH} = -80 µA	0.9 V_{CC}		V
I_{IL}	Logical 0 Input Current (Ports 1,2,3)	V_{IN} = 0.45V		-50	µA
I_{TL}	Logical 1 to 0 Transition Current (Ports 1,2,3)	V_{IN} = 2V		-650	µA
I_{LI}	Input Leakage Current (Port 0, EA)	0.45 < V_{IN} < V_{CC}		±10	µA
RRST	Reset Pulldown Resistor		50	300	kΩ
C_{IO}	Pin Capacitance	Test Freq. = 1 MHz, T_A = 25°C		10	pF
I_{CC}	Power Supply Current	Active Mode, 12 MHz		25	mA
		Idle Mode, 12 MHz		6.5	mA
	Power Down Mode (2)	V_{CC} = 6V		100	µA
		V_{CC} = 3V		40	µA

Notes: 1. Under steady state (non-transient) conditions, I_{OL} must be externally limited as follows:
Maximum I_{OL} per port pin: 10 mA
Maximum I_{OL} per 8 bit port:
Port 0: 26 mA
Ports 1,2,3: 15 mA
Maximum total I_{OL} for all output pins: 71 mA
If I_{OL} exceeds the test condition, V_{OL} may exceed the related specification. Pins are not guaranteed to sink current greater than the listed test conditions.
2. Minimum V_{CC} for Power Down is 2V.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

AC Characteristics

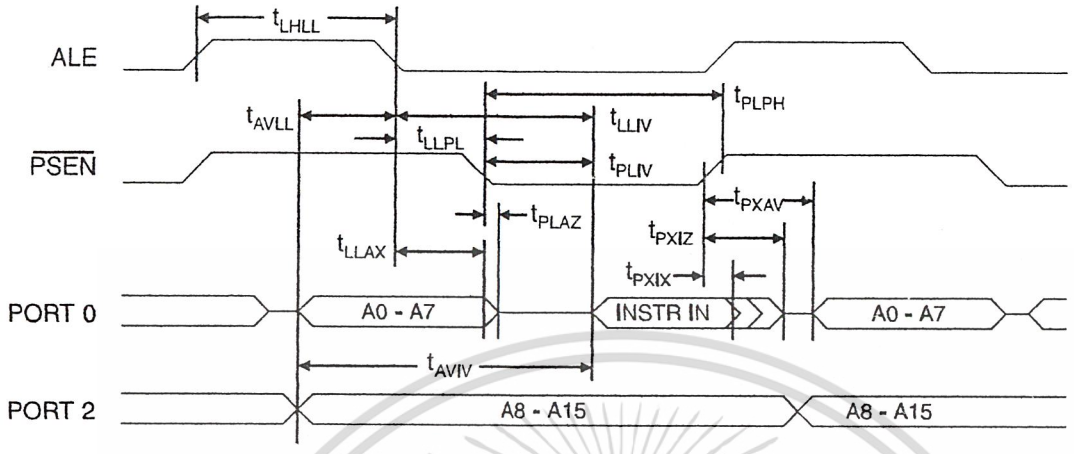
Under operating conditions, load capacitance for Port 0, ALE/$\overline{PROG}$, and $\overline{PSEN}$ = 100 pF; load capacitance for all other outputs = 80 pF.

External Program and Data Memory Characteristics

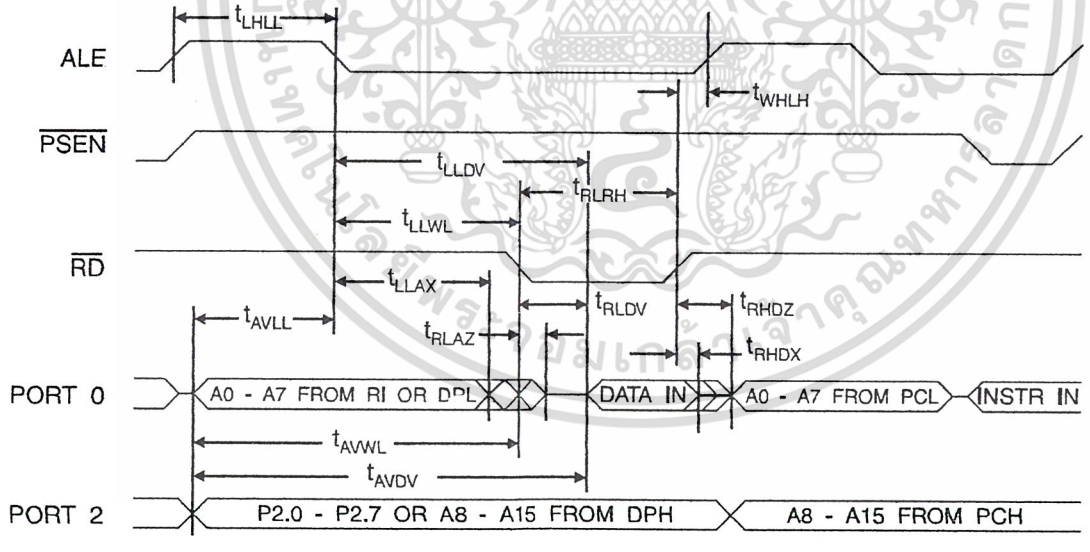
Symbol	Parameter	12 MHz Oscillator		Variable Oscillator		Units
		Min	Max	Min	Max	
$1/t_{CLCL}$	Oscillator Frequency			0	24	MHz
t_{LHLL}	ALE Pulse Width	127		$2t_{CLCL} - 40$		ns
t_{AVLL}	Address Valid to ALE Low	28		$t_{CLCL} - 13$		ns
t_{LLAX}	Address Hold After ALE Low	48		$t_{CLCL} - 20$		ns
t_{LLIV}	ALE Low to Valid Instruction In		233		$4t_{CLCL} - 65$	ns
t_{LLPL}	ALE Low to $\overline{PSEN}$ Low	43		$t_{CLCL} - 13$		ns
t_{PLPH}	$\overline{PSEN}$ Pulse Width	205		$3t_{CLCL} - 20$		ns
t_{PLIV}	$\overline{PSEN}$ Low to Valid Instruction In		145		$3t_{CLCL} - 45$	ns
t_{PXIX}	Input Instruction Hold After $\overline{PSEN}$	0		0		ns
t_{PXIZ}	Input Instruction Float After $\overline{PSEN}$		59		$t_{CLCL} - 10$	ns
t_{PXAV}	$\overline{PSEN}$ to Address Valid	75		$t_{CLCL} - 8$		ns
t_{AVIV}	Address to Valid Instruction In		312		$5t_{CLCL} - 55$	ns
t_{PLAZ}	$\overline{PSEN}$ Low to Address Float		10		10	ns
t_{RLRH}	$\overline{RD}$ Pulse Width	400		$6t_{CLCL} - 100$		ns
t_{WLWH}	$\overline{WR}$ Pulse Width	400		$6t_{CLCL} - 100$		ns
t_{RLDV}	$\overline{RD}$ Low to Valid Data In		252		$5t_{CLCL} - 90$	ns
t_{RHDX}	Data Hold After $\overline{RD}$	0		0		ns
t_{RHDZ}	Data Float After $\overline{RD}$		97		$2t_{CLCL} - 28$	ns
t_{LLDV}	ALE Low to Valid Data In		517		$8t_{CLCL} - 150$	ns
t_{AVDV}	Address to Valid Data In		585		$9t_{CLCL} - 165$	ns
t_{LLWL}	ALE Low to $\overline{RD}$ or $\overline{WR}$ Low	200	300	$3t_{CLCL} - 50$	$3t_{CLCL} + 50$	ns
t_{AVWL}	Address to $\overline{RD}$ or $\overline{WR}$ Low	203		$4t_{CLCL} - 75$		ns
t_{QVWX}	Data Valid to $\overline{WR}$ Transition	23		$t_{CLCL} - 20$		ns
t_{QVWH}	Data Valid to $\overline{WR}$ High	433		$7t_{CLCL} - 120$		ns
t_{WHQX}	Data Hold After $\overline{WR}$	33		$t_{CLCL} - 20$		ns
t_{RLAZ}	$\overline{RD}$ Low to Address Float		0		0	ns
t_{WHLH}	$\overline{RD}$ or $\overline{WR}$ High to ALE High	43	123	$t_{CLCL} - 20$	$t_{CLCL} + 25$	ns

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

External Program Memory Read Cycle

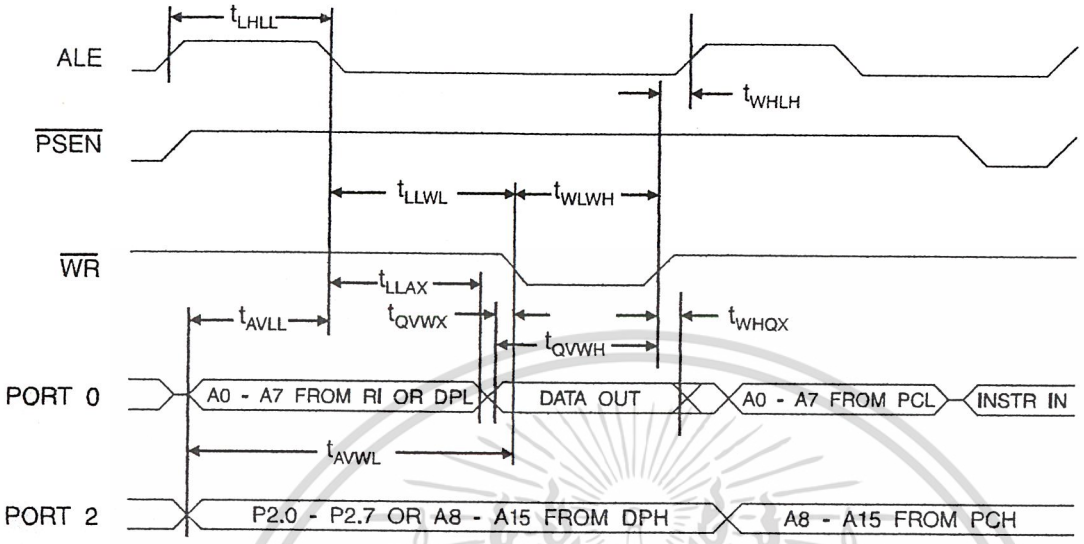


External Data Memory Read Cycle

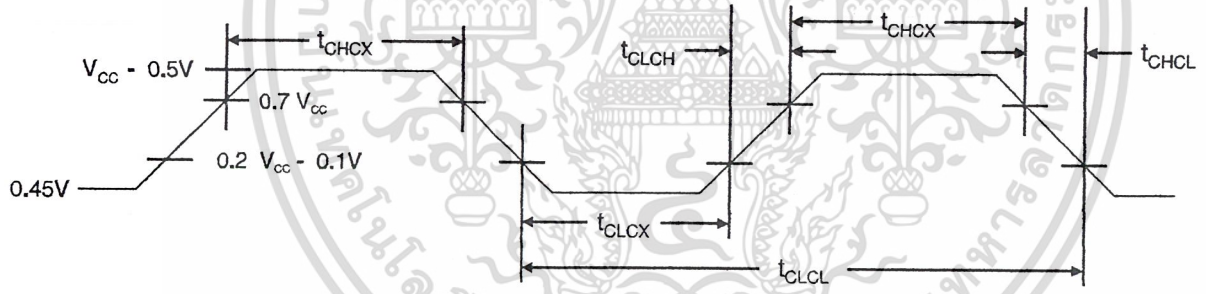


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

External Data Memory Cycle



External Clock Drive Waveforms



External Clock Drive

Symbol	Parameter	Vcc = 4.0V to 6.0V		Units	Vcc = 2.7V to 6.0V	
		Min	Max		Min	Max
1/t_{CLCL}	Oscillator Frequency	0	24	MHz	0	12
t_{CLCL}	Clock Period	41.6		ns	83.3	
t_{CHCX}	High Time	15		ns	30	
t_{CLCX}	Low Time	15		ns	30	
t_{CLCH}	Rise Time		20	ns		20
t_{CHCL}	Fall Time		20	ns		20

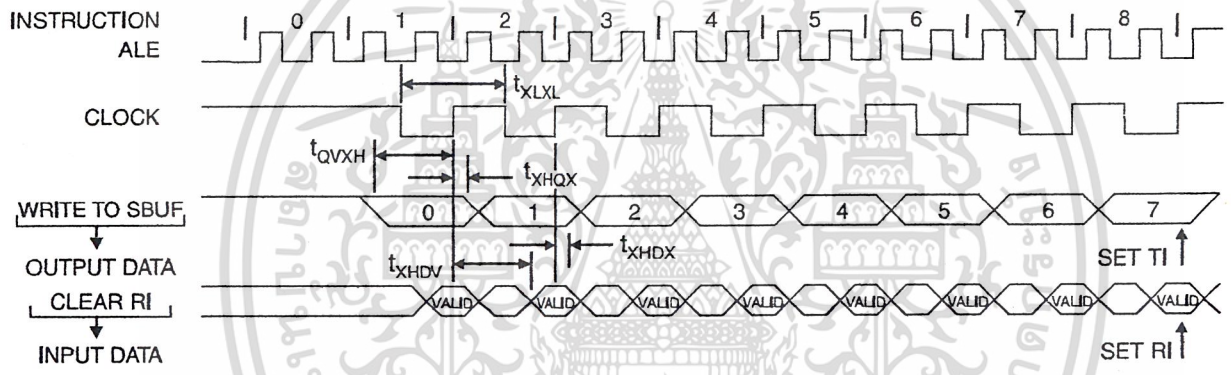
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Serial Port Timing: Shift Register Mode Test Conditions

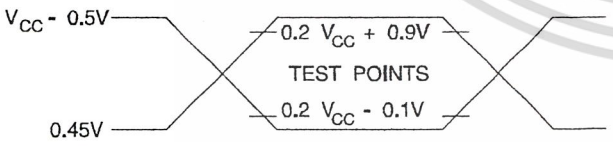
The values in this table are valid for V_{CC} = 2.7V to 6V and Load Capacitance = 80 pF.

Symbol	Parameter	12 MHz Osc		Variable Oscillator		Units
		Min	Max	Min	Max	
t_{XLXL}	Serial Port Clock Cycle Time	1.0		$12t_{CLCL}$		µs
t_{QVXH}	Output Data Setup to Clock Rising Edge	700		$10t_{CLCL}$ - 133		ns
t_{XHQX}	Output Data Hold After Clock Rising Edge	50		$2t_{CLCL}$ - 33		ns
t_{XHDX}	Input Data Hold After Clock Rising Edge	0		0		ns
t_{XHDV}	Clock Rising Edge to Input Data Valid		700		$10t_{CLCL}$ - 133	ns

Shift Register Mode Timing Waveforms

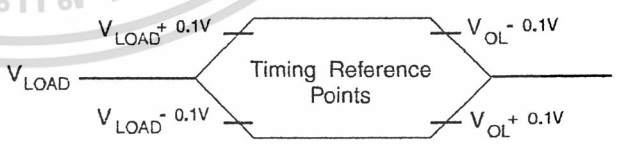


AC Testing Input/Output Waveforms (1)



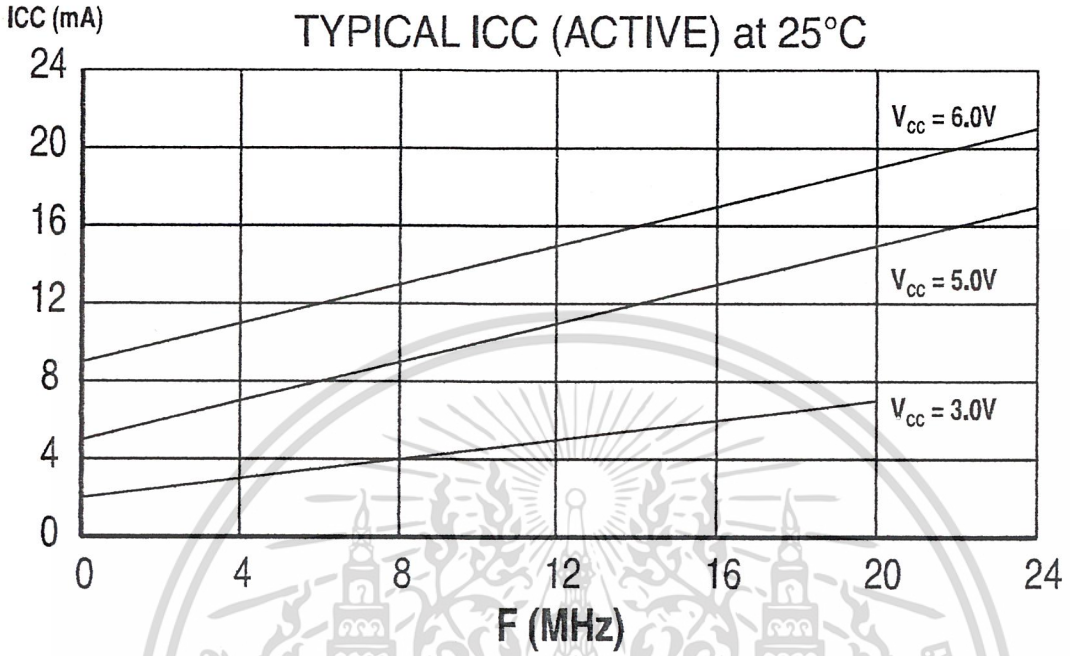
Note: 1. AC Inputs during testing are driven at V_{CC} - 0.5V for a logic 1 and 0.45V for a logic 0. Timing measurements are made at V_{IH} min. for a logic 1 and V_{IL} max. for a logic 0.

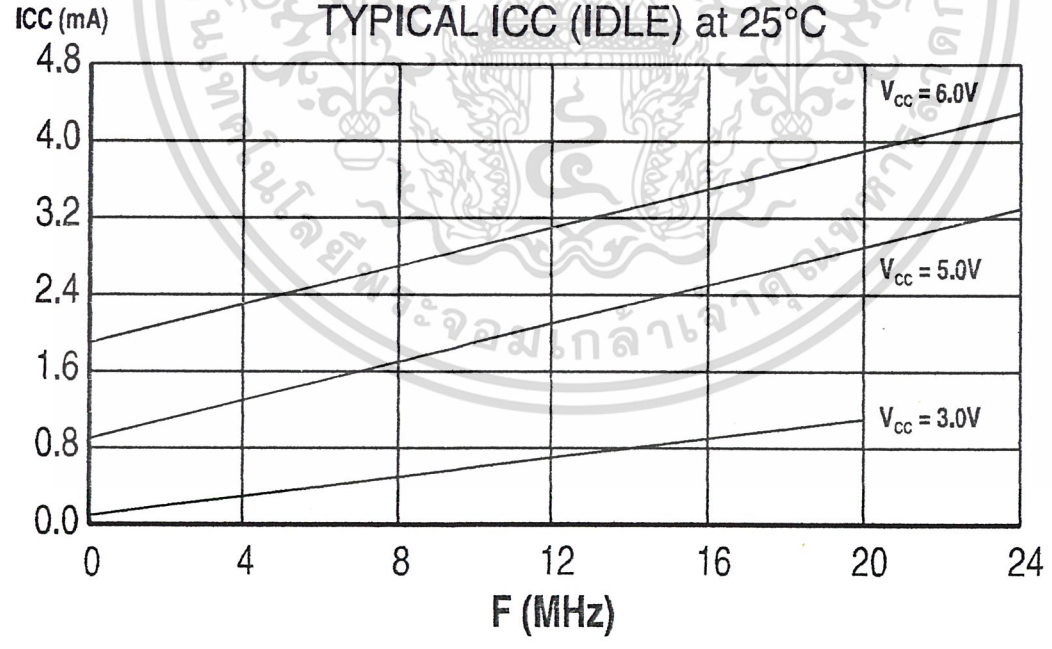
Float Waveforms (1)



Note: 1. For timing purposes, a port pin is no longer floating when a 100 mV change from load voltage occurs. A port pin begins to float when a 100 mV change from the loaded V_{OH}/V_{OL} level occurs.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

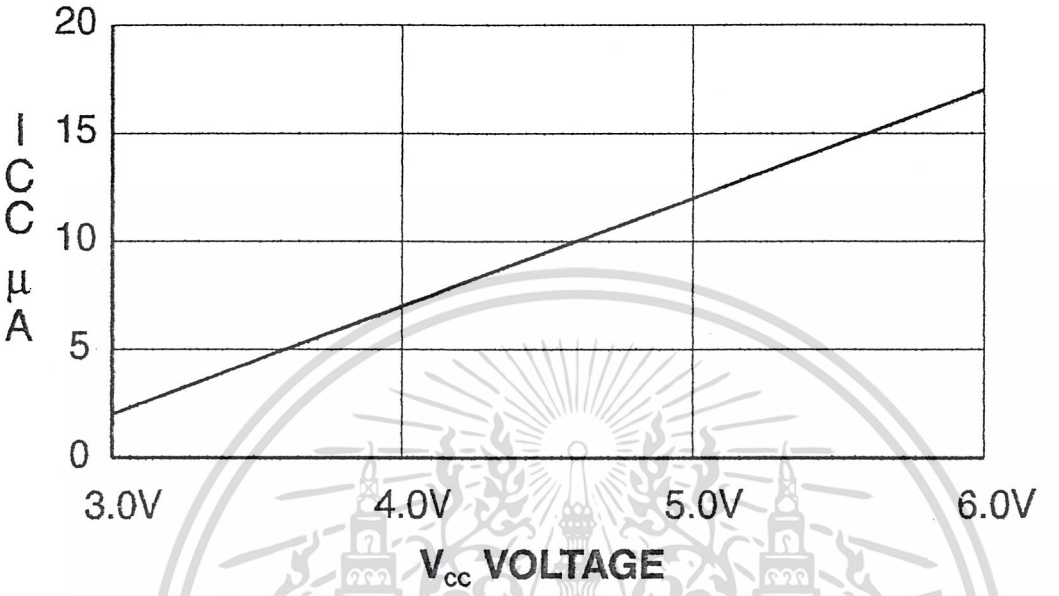




เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

AT89S8252

TYPICAL ICC vs. VOLTAGE - POWER DOWN (85°C)



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Ordering Information

Speed (MHz)	Power Supply	Ordering Code	Package	Operation Range
12	2.7V to 6.0V	AT89S852-12AC AT89S852-12JC AT89S852-12PC AT89S852-12QC	44A 44J 40P6 44Q	Commercial (0°C to 70°C)
		AT89S852-12AI AT89S852-12JI AT89S852-12PI AT89S852-12QI	44A 44J 40P6 44Q	Industrial (-40°C to 85°C)
16	4.0V to 6.0V	AT89S852-16AA AT89S852-16JA AT89S852-16PA AT89S852-16QA	44A 44J 40P6 44Q	Automotive (-40°C to 125°C)
20	4.0V to 6.0V	AT89S852-20AC AT89S852-20JC AT89S852-20PC AT89S852-20QC	44A 44J 40P6 44Q	Commercial (0°C to 70°C)
		AT89S852-20AI AT89S852-20JI AT89S852-20PI AT89S852-20QI	44A 44J 40P6 44Q	Industrial (-40°C to 85°C)
24	4.0V to 6.0V	AT89S852-24AC AT89S852-24JC AT89S852-24PC AT89S852-24QC	44A 44J 44P6 44Q	Commercial (0°C to 70°C)
		AT89S852-24AI AT89S852-24JI AT89S852-24PI AT89S852-24QI	44A 44J 44P6 44Q	Industrial (-40°C to 85°C)

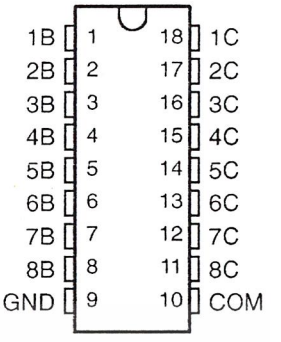
Package Type	
44A	44 Lead, Thin Plastic Gull Wing Quad Flatpack (TQFP)
44J	44 Lead, Plastic J-Leaded Chip Carrier (PLCC)
40P6	40 Lead, 0.600" Wide, Plastic Dual Inline Package (PDIP)
44Q	44 Lead, Plastic Gull Wing Quad Flatpack (PQFP)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

SLRS049 – FEBRUARY1997

- **500 mA Rated Collector Current (Single Output)**
- **High-Voltage Outputs . . . 50 V**
- **Output Clamp Diodes**
- **Inputs Compatible With Various Types of Logic**
- **Relay Driver Applications**
- **Compatible with ULN2800A Series**

N PACKAGE
(TOP VIEW)



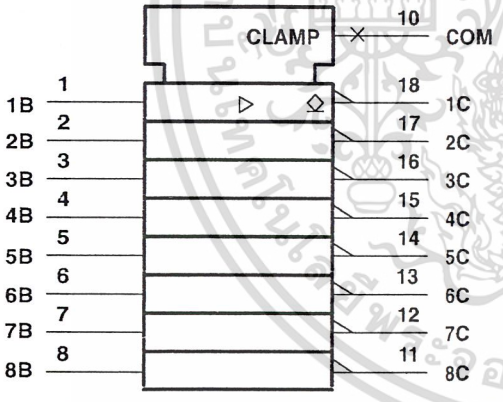
description

The ULN2803A is a monolithic high-voltage, high-current Darlington transistor array. The device consists of eight npn Darlington pairs that feature high-voltage outputs with common-cathode clamp diodes for switching inductive loads. The collector-current rating of each Darlington pair is 500 mA. The Darlington pairs may be paralleled for higher current capability.

Applications include relay drivers, hammer drivers, lamp drivers, display drivers (LED and gas discharge), line drivers, and logic buffers. The ULN2803A has a 2.7-kΩ series base resistor for each Darlington pair for operation directly with TTL or 5-V CMOS devices.

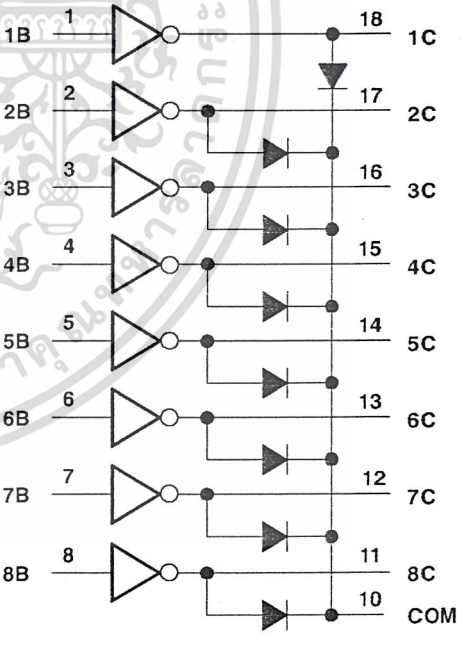
The ULN2803A is offered in a standard 18-pin dual in-line (N) package. The device is characterized for operation over the temperature range of –20°C to 85°C.

logic symbol†



† This symbol is in accordance with ANSI/IEEE Std 91-1984 and IEC Publication 617-12.

logic diagram (positive logic)



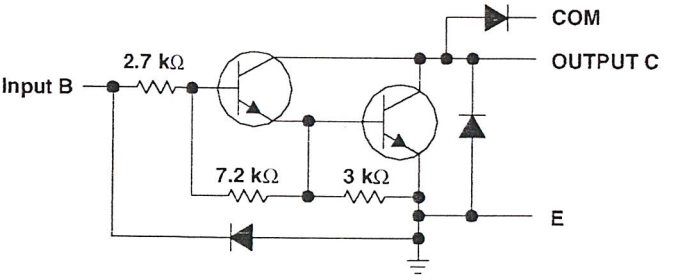
Please be aware that an important notice concerning availability, standard warranty, and use in critical applications of Texas Instruments semiconductor products and disclaimers thereto appears at the end of this data sheet.

PRODUCTION DATA information is current as of publication date. Products conform to specifications per the terms of Texas Instruments standard warranty. Production processing does not necessarily include testing of all parameters.

Copyright © 1997, Texas Instruments Incorporated

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

schematic (each Darlington pair)



absolute maximum ratings at 25°C free-air temperature (unless otherwise noted)†

Collector-emitter voltage	50 V
Input voltage (see Note 1)	30 V
Continuous collector current	500 mA
Output clamp diode current	500 mA
Total substrate-terminal current	–2.5 A
Continuous dissipation at (or below) 25°C free-air temperature	1150 mW
Operating free-air temperature range, T_A	–20°C to 85°C
Storage temperature range, T_{stg}	–65°C to 150°C
Lead temperature 1,6 mm (1/16 inch) from case for 10 seconds:	260°C

† Stresses beyond those listed under "absolute maximum ratings" may cause permanent damage to the device. These are stress ratings only, and functional operation of the device at these or any other conditions beyond those indicated under "recommended operating conditions" is not implied. Exposure to absolute-maximum-rated conditions for extended periods may affect device reliability.

NOTES: 1. All voltages values, unless otherwise noted, are with respect to the emitter/substrate terminal GND.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

electrical characteristics at 25°C free-air temperature (unless otherwise noted)

PARAMETER		TEST CONDITIONS		MIN	TYP	MAX	UNIT
I_{CEX}	Collector cutoff current	V_{CE} = 50 V, See Figure 1	I_I = 0,			50	μA
$I_{I(off)}$	Off-state input current	V_{CE} = 50 V, T_A = 70°C,	I_C = 500 μA, See Figure 2	50	65		μA
$I_{I(on)}$	Input current	V_I = 3.85 V,	See Figure 3		0.93	1.35	mA
$V_{I(on)}$	On-state input voltage	V_{CE} = 2 V, See Figure 4	I_C = 200 mA			2.4	V
			I_C = 250 mA			2.7	
			I_C = 300 mA			3	
$V_{CE(sat)}$	Collector emitter saturation voltage	I_I = 250 μA, See Figure 5	I_C = 100 mA,		0.9	1.1	V
		I_I = 350 μA, See Figure 5	I_C = 200 mA,		1	1.3	
		I_I = 500 μA, See Figure 5	I_C = 350 mA,		1.3	1.6	
I_R	Clamp diode reverse current	V_R = 50 V,	See Figure 6			50	μA
V_F	Clamp diode forward voltage	I_F = 350 mA,	See Figure 7		1.7	2	V
C_i	Input capacitance	V_I = 0 V,	f = 1 MHz		15	25	pF

switching characteristics at 25°C free-air temperature

PARAMETER		TEST CONDITIONS		MIN	TYP	MAX	UNIT
t_{PLH}	Propagation delay time, low-to-high-level output	V_S = 50 V, C_L = 15 pF,	R_L = 163 Ω, See Figure 8		130		ns
t_{PHL}	Propagation delay time, high-to-low level output				20		
V_{OH}	High-level output voltage after switching	V_S = 50 V, See Figure 9	I_O = 300 mA,	V_S–20			mV

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งาน ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

PARAMETER MEASUREMENT INFORMATION

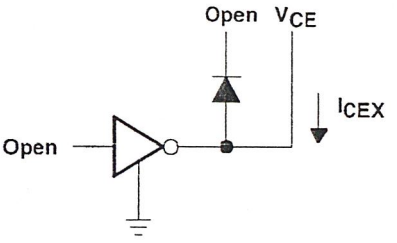


Figure 1. I_{CEX} Test Circuit

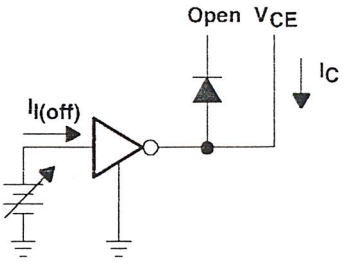


Figure 2. $I_{I(off)}$ Test Circuit

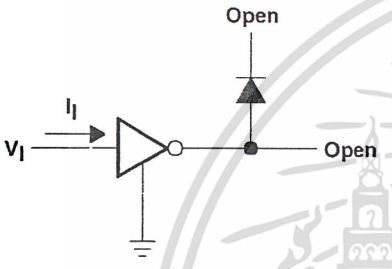


Figure 3. $I_{I(on)}$ Test Circuit

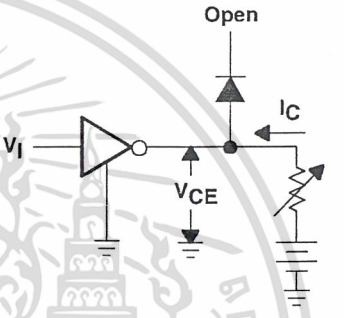


Figure 4. $V_{I(on)}$ Test Circuit

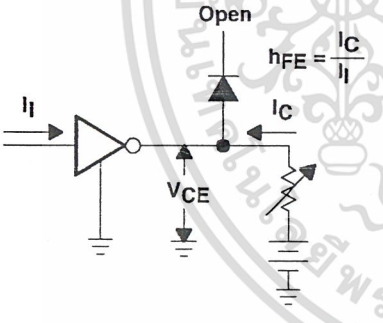


Figure 5. h_{FE}, $V_{CE(sat)}$ Test Circuit

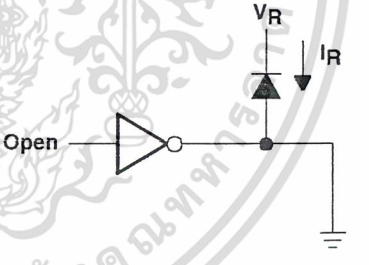


Figure 6. I_R Test Circuit

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

PARAMETER MEASUREMENT INFORMATION

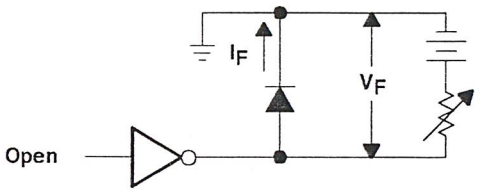
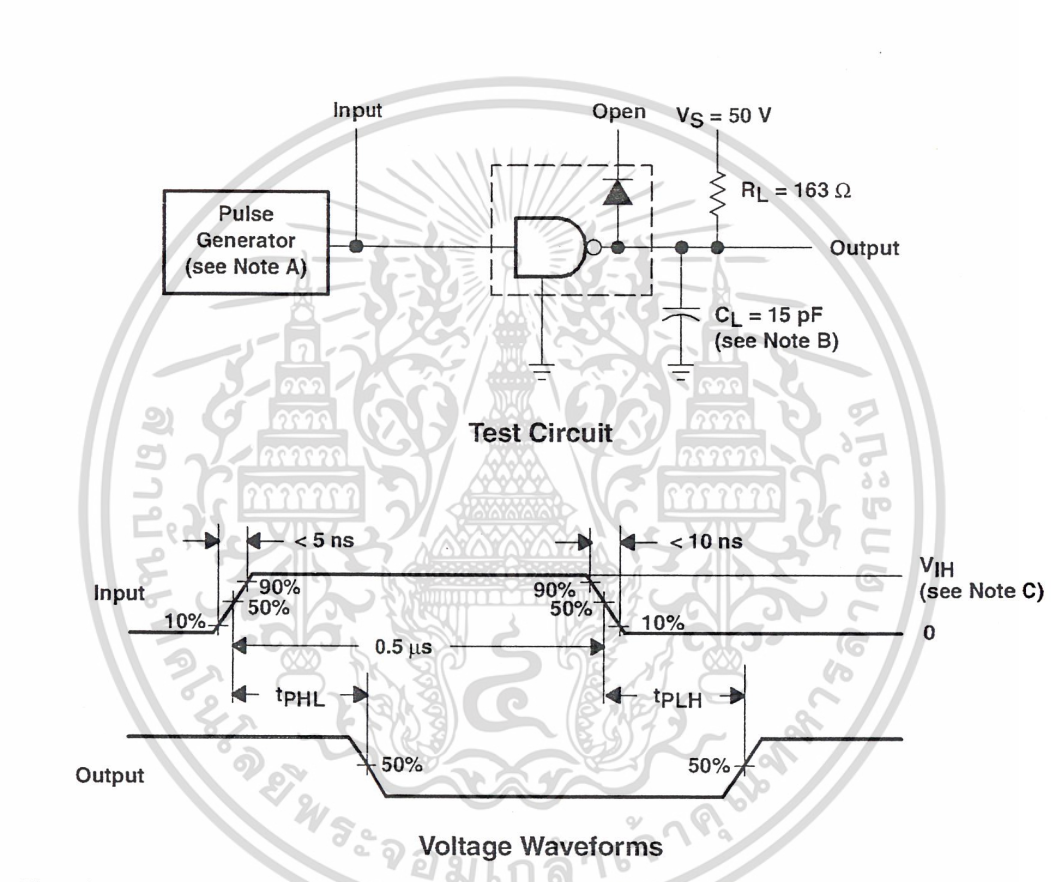


Figure 7. V_F Test Circuit

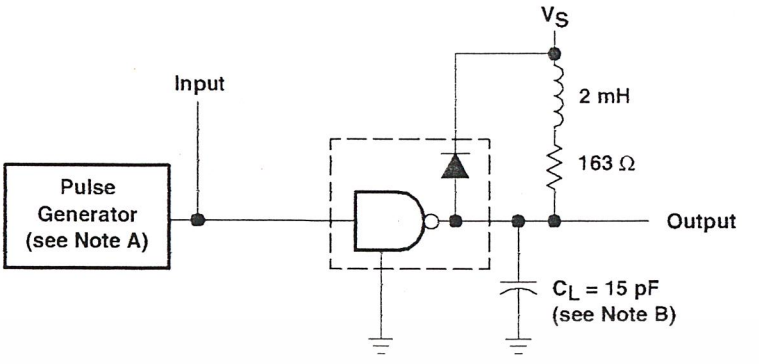


NOTES: A. The pulse generator has the following characteristics: PRR = 1 MHz, Z_O = 50 Ω.
B. C_L includes probe and jig capacitance.
C. V_{IH} = 3 V

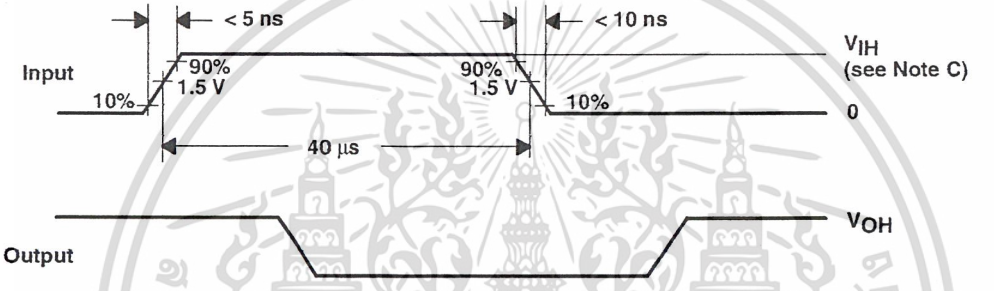
Figure 8. Propagation Delay Times

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

PARAMETER MEASUREMENT INFORMATION



Test Circuit



Voltage Waveforms

NOTES: A. The pulse generator has the following characteristics: PRR = 12.5 KHz, Z_O = 50 Ω.
B. C_L includes probe and jig capacitance.
C. V_{IH} = 3 V

Figure 9. Latch-Up Test

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษานั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

IMPORTANT NOTICE

Texas Instruments (TI) reserves the right to make changes to its products or to discontinue any semiconductor product or service without notice, and advises its customers to obtain the latest version of relevant information to verify, before placing orders, that the information being relied on is current.

TI warrants performance of its semiconductor products and related software to the specifications applicable at the time of sale in accordance with TI's standard warranty. Testing and other quality control techniques are utilized to the extent TI deems necessary to support this warranty. Specific testing of all parameters of each device is not necessarily performed, except those mandated by government requirements.

Certain applications using semiconductor products may involve potential risks of death, personal injury, or severe property or environmental damage ("Critical Applications").

TI SEMICONDUCTOR PRODUCTS ARE NOT DESIGNED, INTENDED, AUTHORIZED, OR WARRANTED TO BE SUITABLE FOR USE IN LIFE-SUPPORT APPLICATIONS, DEVICES OR SYSTEMS OR OTHER CRITICAL APPLICATIONS.

Inclusion of TI products in such applications is understood to be fully at the risk of the customer. Use of TI products in such applications requires the written approval of an appropriate TI officer. Questions concerning potential risk applications should be directed to TI through a local SC sales office.

In order to minimize risks associated with the customer's applications, adequate design and operating safeguards should be provided by the customer to minimize inherent or procedural hazards.

TI assumes no liability for applications assistance, customer product design, software performance, or infringement of patents or services described herein. Nor does TI warrant or represent that any license, either express or implied, is granted under any patent right, copyright, mask work right, or other intellectual property right of TI covering or relating to any combination, machine, or process in which such semiconductor products or services might be or are used.

Copyright © 1996, Texas Instruments Incorporated

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

DALLAS SEMICONDUCTOR

DS1307
64 X 8 Serial Real Time Clock

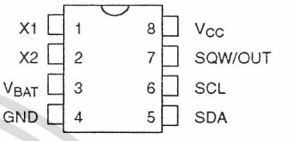
FEATURES

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

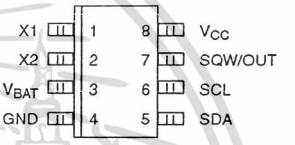
ORDERING INFORMATION

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

PIN ASSIGNMENT



DS1307 8–PIN DIP (300 MIL)



DS1307Z 8–PIN SOIC (150 MIL)

PIN DESCRIPTION

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

DESCRIPTION

The DS1307 Serial Real Time Clock is a low power full BCD clock calendar plus 56 bytes of nonvolatile SRAM. Address and data are transferred serially via a 2–wire bi–directional bus. The clock/calendar provides seconds, minutes, hours, day, date, month, and year information. The end of the month date is automatically adjusted for months with less than 31 days, including corrections for leap year. The clock operates in either the 24–hour or 12–hour format with AM/PM indicator. The DS1307 has a built–in power sense circuit which detects power failures and automatically switches to the battery supply.

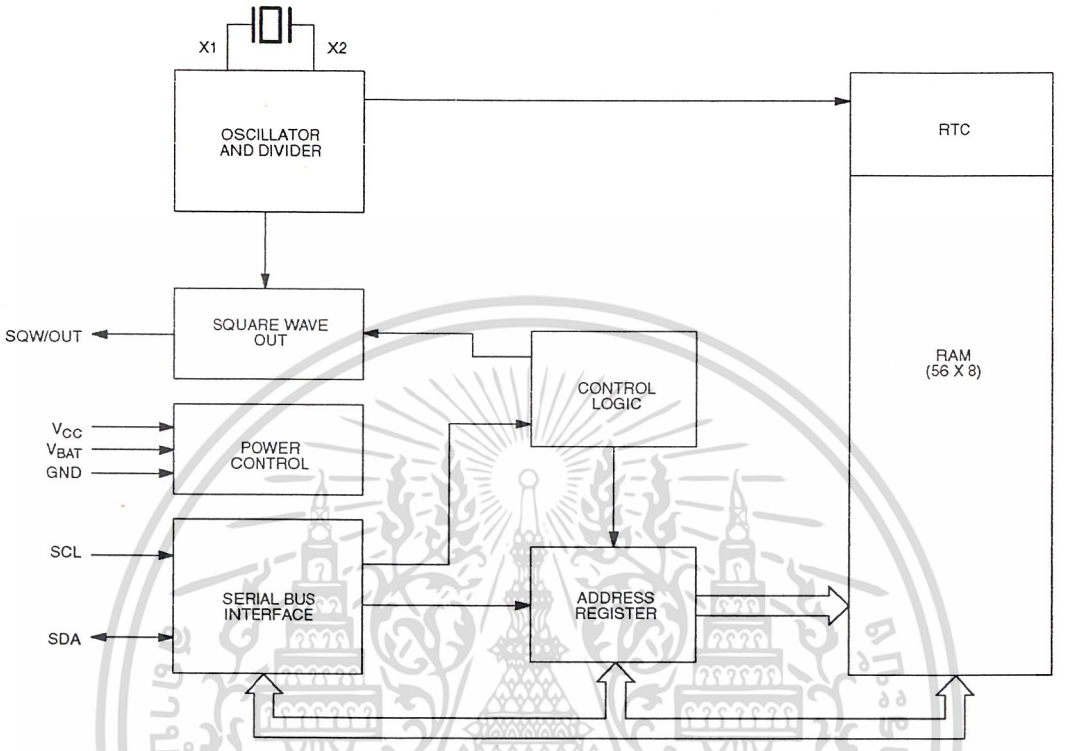
OPERATION

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

©Copyright 1995 by Dallas Semiconductor Corporation.
All Rights Reserved. For important information regarding patents and other intellectual property rights, please refer to Dallas Semiconductor data books.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

DS1307 BLOCK DIAGRAM Figure 1



SIGNAL DESCRIPTIONS

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

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

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

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

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

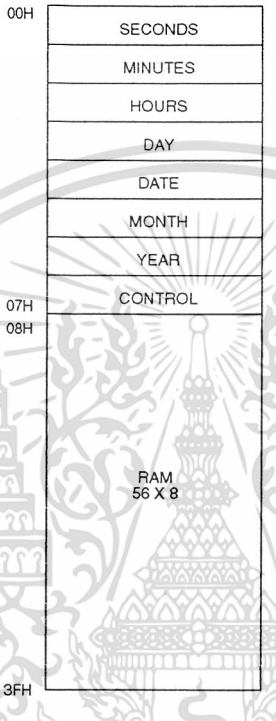
X1, X2 – Connections for a standard 32.768 KHz quartz crystal. The internal oscillator circuitry is designed for operation with a crystal having a specified load capacitance (CL) of 12.5 pF.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

RTC AND RAM ADDRESS MAP

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

DS1307 ADDRESS MAP Figure 2



CLOCK AND CALENDAR

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

The DS1307 can be run in either 12–hour or 24–hour mode. Bit 6 of the hours register is defined as the 12– or 24–hour mode select bit. When high, the 12–hour mode is selected. In the 12–hour mode, bit 5 is the AM/PM bit with logic high being PM. In the 24–hour mode, bit 5 is the second 10 hour bit (20–23 hours).

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

DS1307 TIMEKEEPER REGISTERS Figure 3

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

CONTROL REGISTER

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

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

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

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

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

SQUAREWAVE OUTPUT FREQUENCY Table 1

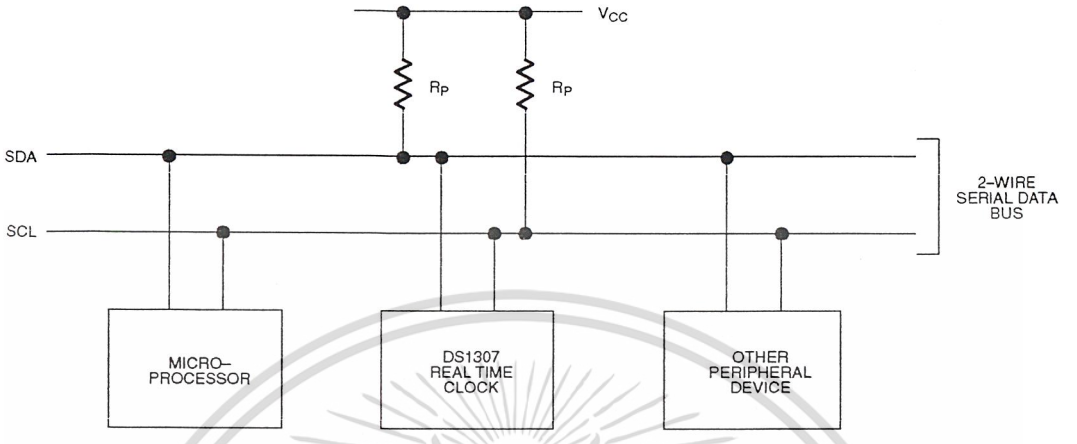
RS1	RS0	SQW OUTPUT FREQUENCY
0	0	1 Hz
0	1	4 KHz
1	0	8 KHz
1	1	32 KHz

2–WIRE SERIAL DATA BUS

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

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

TYPICAL 2–WIRE BUS CONFIGURATION Figure 4



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

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

Accordingly, the following bus conditions have been defined:

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

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

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

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

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

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

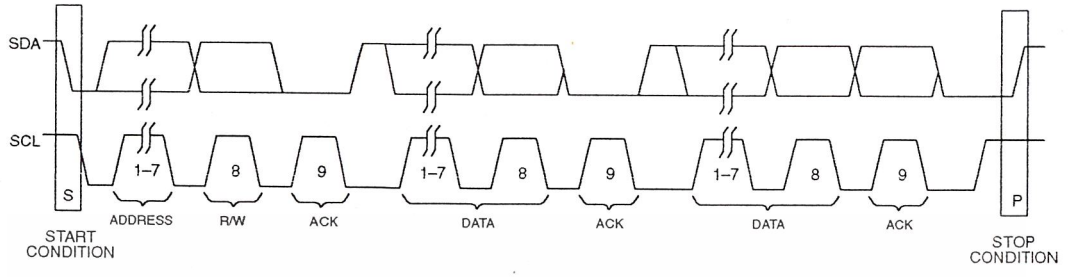
A device that acknowledges must pull down the SDA line during the acknowledge clock pulse in such a way that the SDA line is stable low during the high period of the acknowledge related clock pulse. Of course, setup and hold times must be taken into account. When receiving data from a slave a master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

DATA TRANSFER

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

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

DATA TRANSFER ON 2–WIRE SERIAL BUS Figure 5



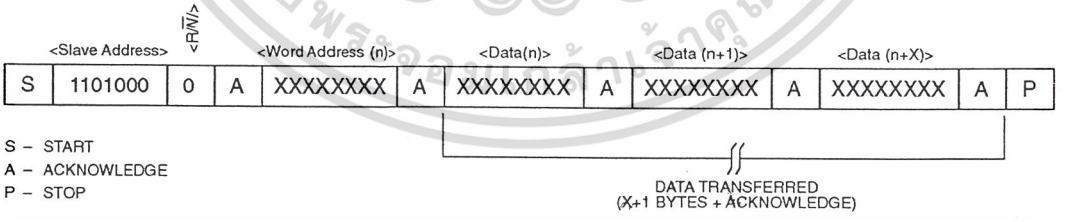
1. Data transfer from a master transmitter to a slave receiver. The first byte transmitted by the master is the slave address. Next follows a number of data bytes. The slave returns an acknowledge bit after each received byte. Data is transfered with the most significant bit (MSB) first.
2. Data transfer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

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

The DS1307 may operate in the following two modes:

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

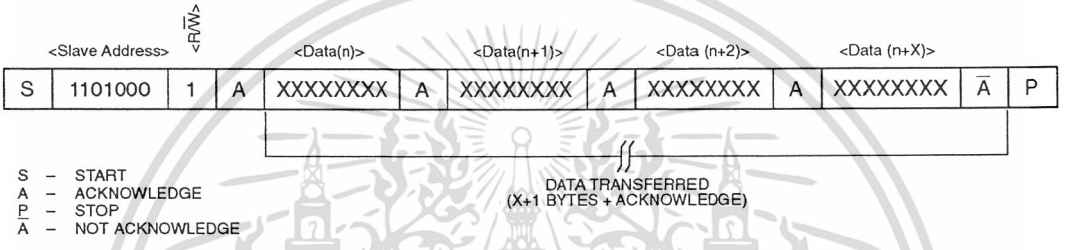
DATA WRITE – SLAVE RECEIVER MODE Figure 6



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

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

DATA READ – SLAVE TRANSMITTER MODE Figure 7



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

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ABSOLUTE MAXIMUM RATINGS*

Voltage on Any Pin Relative to Ground	–0.5V to +7.0V
Operating Temperature	0°C to 70°C
Storage Temperature	–55°C to +125°C
Soldering Temperature	260°C for 10 seconds

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

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

RECOMMENDED DC OPERATING CONDITIONS

(0°C to 70°C)

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

DC ELECTRICAL CHARACTERISTICS

(0°C to 70°C; V_{CC}=4.5V to 5.5V)

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

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

AC ELECTRICAL CHARACTERISTICS

(0°C to 70°C; V_{CC}=4.5V to 5.5V)

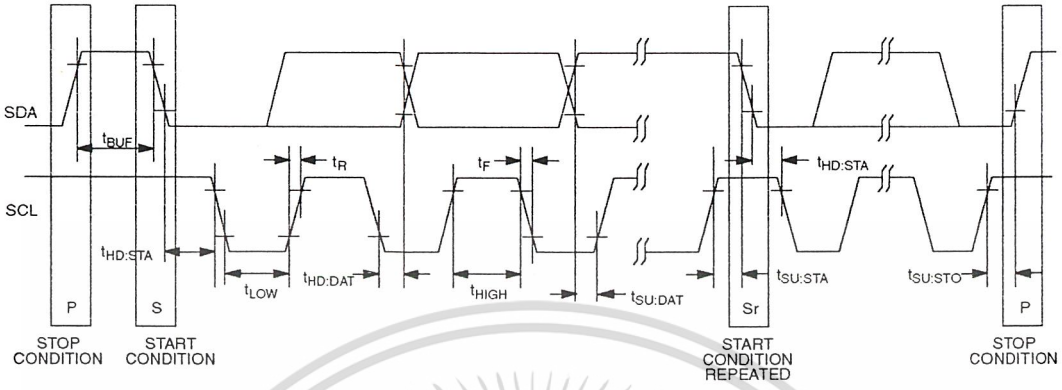
PARAMETER	SYMBOL	MIN	TYP	MAX	UNITS	NOTES
SCL Clock Frequency	f_{SCL}	0		100	KHz	
Bus Free Time Between a STOP and START Condition	t_{BUF}	4.7			µs	
Hold Time (Repeated) START Condition	$t_{HD:STA}$	4.0			µs	5
LOW Period of SCL Clock	t_{LOW}	4.7			µs	
HIGH Period of SCL Clock	t_{HIGH}	4.0			µs	
Set–up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

NOTES:

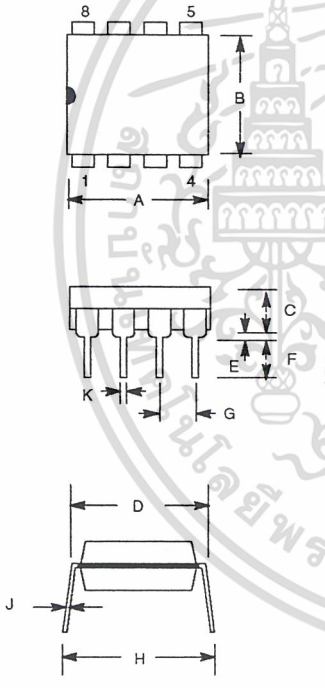
1. All voltages are referenced to ground.
2. Logic zero voltages are specified at a sink current of 5 mA at V_{CC}=4.5V, V_{OL}=GND for capacitive loads.
3. I_{CCS} specified with V_{CC}=5.0V and SDA, SCL=5.0V.
4. V_{CC}=0V, V_{BAT}=3V.
5. After this period, the first clock pulse is generated.
6. A device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

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

TIMING DIAGRAM



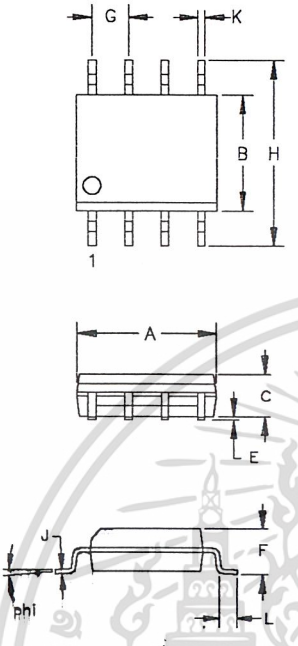
DS1307 64 X 8 SERIAL REAL TIME CLOCK 8–PIN DIP



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

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

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



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

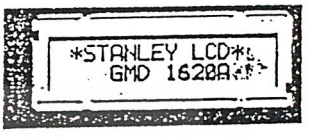
56–G2008–001

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

▼ LCD Type

TN Type

This is the most typical general-purpose type LCD, being used in applications ranging from watches and calculators to Dot Matrix Modules.

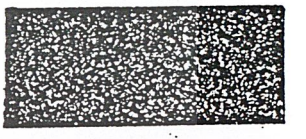


● **TN Positive Image Type (Basic Type)**

The most typical general-purpose LCD of the TN types.
This type features black characters (or drawings) on a neutral gray background.
Usable in three modes: Reflective mode, Transmissive mode and Transflective mode.

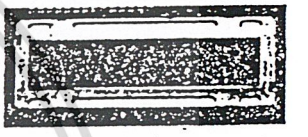
● **TN Negative Image Type**

This has reversed image of positive type, with a white display on a black background.
Usable in Transmissive mode only.



● **Wide Temperature Type**

In contrast with the operating temperature range of 0 to 50°C for normal TN types, this type features a wide operating temperature range of −20 to 70°C. (This LCD is particularly effective in use with backlighting which generates heat.)

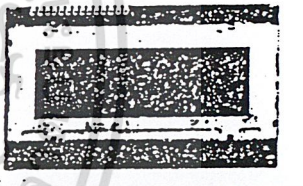


STN Type

Compared with TN types, this high-resolution display LCD intended for Dot Matrix Display has an increased twisted angle, resulting in a more than two-fold improvement in contrast and viewing angle characteristics.

● **STN Yellow Type**

Dark blue display on a yellow-green background.
This is the most typical general-purpose type of high-contrast STN types.
Usable in three modes: Reflective mode, Transmissive mode and Transflective mode.



● **STN Blue Type**

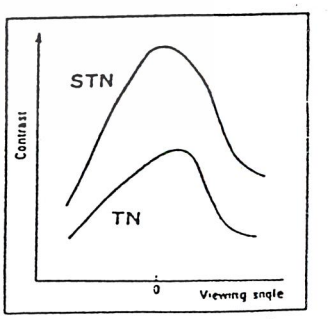
Backlight color display on a blue background.
This type intended specifically for use in combination with a white light source.
Usable in transmissive mode only.

● **STN Gray Type**

Dark blue display on a blue-gray background. This type features an improvement that provides a display which appears natural with its surround colors, when the yellow mode background is blanked. Usable in reflective mode

Features of Each Mode

	LCD Type	Background Color	Display Color	Viewing Direction	Contrast	LCD mode			Temperature	
						Reflective	Transmissive	Transflective	Opr.	Stg.
TN	Positive	Neutral gray	Black	○	○	●	○	○	0~ +50	−20~ +70
TN	Negative	Black	Clear	○	○	×	△	●		
TN	Wide Tem.	—	—	○	○	—	—	—	−20~+70	−30~+80
STN	Yellow	Yellow-Green	Dark blue	●	●	●	○	○	0~ +40	−20~ +60
STN	Blue	Blue	Clear	●	●	×	△	●		
STN	Gray	Blue-Gray	Dark blue	○	○	○	△	×		



Example of Contrast-Viewing Angle Curve

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

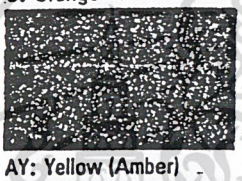
▼ Backlighting

STANLEY Dot Matrix LCDs feature three types of backlighting:
high-intensity, long-life LEDs, CFL with brightness and good color rendition, and slim-line EL.

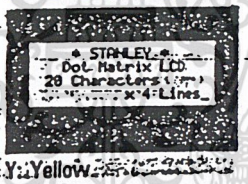
LED Backlighting

R: Red

O: Orange



AY: Yellow (Amber)



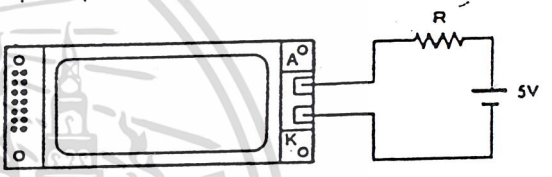
Y: Yellow

G: Green

■ **Features**
- Long-life, bright and uniform backlighting
- Operating voltage: 4.4-4.7V (Using external resister)
- Five colors are available
- The use of high-intensity LEDs and unique illumination technology enable effective backlighting with a small number of chips.

■ **Specifications**
Please see each page for standard types.

■ **Example of power supply circuit for LED Backlighting Unit.**



R: Forward current limiting resister

* Always use an externally connected resister (R) for the backlighting unit.

■ **Precaution for usage**
The operating temperature range of LCD modules with LED backlighting is specified with the assumption that the forward current supplied to the LED units will be varied according to the ambient temperature.

Fig. 1 Example of Relative Luminous Intensity vs. Forward Current (For GMO1616FLY)

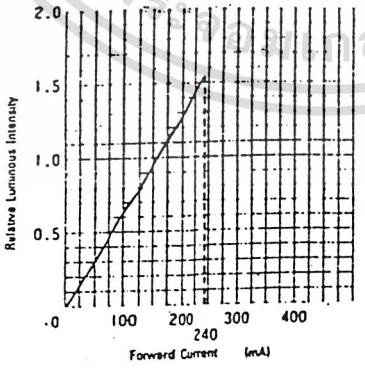
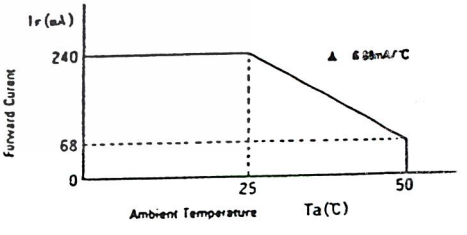


Fig. 2 Example of Forward Current vs. Ambient Temperature (For GMO1616FLY)



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ALPHANUMERIC DOT MATRIX MODULES

DISPLAY CHARACTER POSITION AND DD RAM ADDRESS

1x8 DMM, 1/8 MUX

N=0 : 1-LINE DISPLAY
F=0 : 5X7 DOTS

	1	2	3	4	5	6	7	8	
DISPLAY POSITION									
FIRST LINE	00	01	02	03	04	05	06	07	DD RAM ADDRESS

WHEN THE DISPLAY SHIFT OPERATION IS PERFORMED, THE DD RAM ADDRESS MOVED AS FOLLOW :

AFTER THE LEFT SHIFT INSTRUCTION

	1	2	3	4	5	6	7	8	
DISPLAY POSITION									
DD RAM ADDRESS	01	02	03	04	05	06	07	08	

AFTER THE RIGHT SHIFT INSTRUCTION

	1	2	3	4	5	6	7	8	
DISPLAY POSITION									
DD RAM ADDRESS	4F	00	01	02	03	04	05	06	

1x16 DMM, 1/16 MUX

N=1 : 2-LINE DISPLAY
F=0 : 5X7 DOTS

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	
DISPLAY POSITION																	
FIRST LINE	00	01	02	03	04	05	06	07	40	41	42	43	44	45	46	47	DD RAM ADDRESS

WHEN THE DISPLAY SHIFT OPERATION IS PERFORMED, THE DD RAM ADDRESS MOVED AS FOLLOW :

AFTER THE LEFT SHIFT INSTRUCTION

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	
DISPLAY POSITION																	
DD RAM ADDRESS	01	02	03	04	05	06	07	08	41	42	43	44	45	46	47	48	

AFTER THE RIGHT SHIFT INSTRUCTION

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	
DISPLAY POSITION																	
DD RAM ADDRESS	27	00	01	02	03	04	05	06	67	40	41	42	43	44	45	45	

1x16 DMM, 1/8 MUX

N=0 : 1-LINE DISPLAY
F=0 : 5X7 DOTS

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	
DISPLAY POSITION																	
FIRST LINE	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	DD RAM ADDRESS

1x20 DMM, 1/8 MUX

N=0 : 1-LINE DISPLAY
F=0 : 5X7 DOTS

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	
DISPLAY POSITION																					
FIRST LINE	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	10	11	12	13	DD RAM ADDRESS

1x40 DMM, 1/11 MUX

N=0 : 1-LINE DISPLAY
F=1 : 5X10 DOTS

	1	2	3	4	5	6	7	8	9	10	11	12	13		33	34	35	36	37	38	39	40	
DISPLAY POSITION																							
FIRST LINE	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	---	28	21	22	23	24	25	26	27	DD RAM ADDRESS

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ALPHANUMERIC DOT MATRIX MODULES

DISPLAY CHARACTER POSITION AND DD RAM ADDRESS (CONTINUE)

2x16 DMM, 1/16 MUX N=1 : 2-LINE DISPLAY F=0 : 5X7 DOTS

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	← DISPLAY POSITION
FIRST LINE	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	← DD RAM ADDRESS
SECOND LINE	40	41	42	43	44	45	46	47	48	49	4A	4B	4C	4D	4E	4F	

WHEN THE DISPLAY SHIFT OPERATION IS PERFOMED, THE DD RAM ADDRESS MOVED AS FOLLOW :

AFTER THE LEFT SHIFT INSTRUCTION

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	← DISPLAY POSITION
	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	10	← DD RAM ADDRESS
	41	42	43	44	45	46	47	48	49	4A	4B	4C	4D	4E	4F	50	

AFTER THE RIGHT SHIFT INSTRUCTION

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	← DISPLAY POSITION
	27	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	← DD RAM ADDRESS
	67	40	41	42	43	44	45	46	47	48	49	4A	4B	4C	4D	4E	

2x20 DMM, 1/16 MUX N=1 : 2-LINE DISPLAY F=0 : 5X7 DOTS

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	← DISPLAY POSITION
FIRST LINE	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	10	11	12	13	← DD RAM ADDRESS
SECOND LINE	40	41	42	43	44	45	46	47	48	49	4A	4B	4C	4D	4E	4F	50	51	52	53	

2x24 DMM, 1/16 MUX N=1 : 2-LINE DISPLAY F=0 : 5X7 DOTS

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	← DISPLAY POSITION
FIRST LINE	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	10	11	12	13	14	15	16	17	← DD RAM ADDRESS
SECOND LINE	40	41	42	43	44	45	46	47	48	49	4A	4B	4C	4D	4E	4F	50	51	52	53	54	55	56	57	

2x40 DMM, 1/16 MUX N=1 : 2-LINE DISPLAY F=0 : 5X7 DOTS

	1	2	3	4	5	6	7	8	9	10	11	12	13	...	33	34	35	36	37	38	39	40	← DISPLAY POSITION
FIRST LINE	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	...	20	21	22	23	24	25	26	27	← DD RAM ADDRESS
SECOND LINE	40	41	42	43	44	45	46	47	48	49	4A	4B	4C	...	60	61	62	63	64	65	66	67	

4x16 DMM, 1/16 MUX N=1 : 2-LINE DISPLAY F=0 : 5X7 DOTS

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	← DISPLAY POSITION
FIRST LINE	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	← DD RAM ADDRESS
SECOND LINE	40	41	42	43	44	45	46	47	48	49	4A	4B	4C	4D	4E	4F	
THIRD LINE	10	11	12	13	14	15	16	17	18	19	1A	1B	1C	1D	1E	1F	
FOURTH LINE	50	51	52	53	54	55	56	57	58	59	5A	5B	5C	5D	5E	5F	

4x20 DMM, 1/16 MUX N=1 : 2-LINE DISPLAY F=0 : 5X7 DOTS

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	← DISPLAY POSITION
FIRST LINE	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	10	11	12	13	← DD RAM ADDRESS
SECOND LINE	40	41	42	43	44	45	46	47	48	49	4A	4B	4C	4D	4E	4F	50	51	52	53	
THIRD LINE	14	15	16	17	18	19	1A	1B	1C	1D	1E	1F	20	21	22	23	24	25	26	27	
FOURTH LINE	54	55	56	57	58	59	5A	5B	5C	5D	5E	5F	60	61	62	63	64	65	66	67	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

DMC162

● Display Format(16character ×2line) ● Display Fonts(5×8dots) ● Driving method(1/16D)

ABSOLUTE MAXIMUM RATINGS

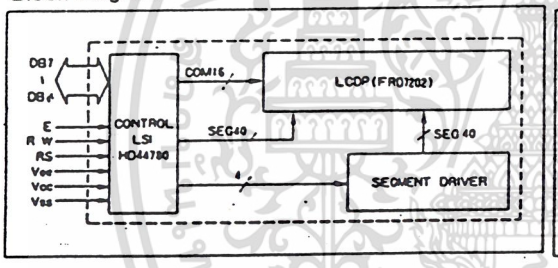
Item	Symbol	Test Condition	Standard Value min.	typ.	max.	Unit
Power Supply Voltage for Logic	Vcc - Vss		0		7	V
Power Supply Voltage for LCD Drive	Vcc - Vee		0		13.5	V
Input "High" Voltage	VI		Vss		Vcc	V
Operating Temperature	Ta		0		+50	℃
Storage Temperature	Tstg		−20		+70	℃

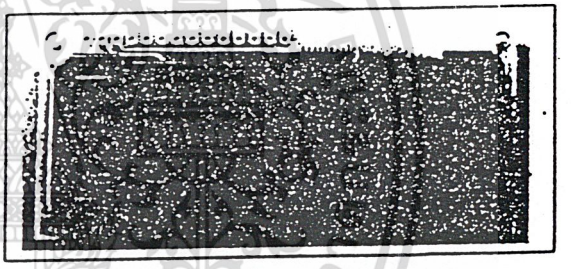
ELECTRICAL CHARACTERISTICS

Item	Symbol	Test Condition	Standard Value min.	typ.	max.	Unit
Input Voltage	VIH	—	2.2	—	Vcc	V
Input "Low" Voltage	VIL	—	−0.3	—	0.6	V
Output "High" Voltage	VOH	−IOH=0.205mA	2.4			V
Output "Low" Voltage	VOL	IOH=1.2mA	—		0.4	V
Power Supply Current	Icc	Vcc=5.0V		0.5	2.0	mA

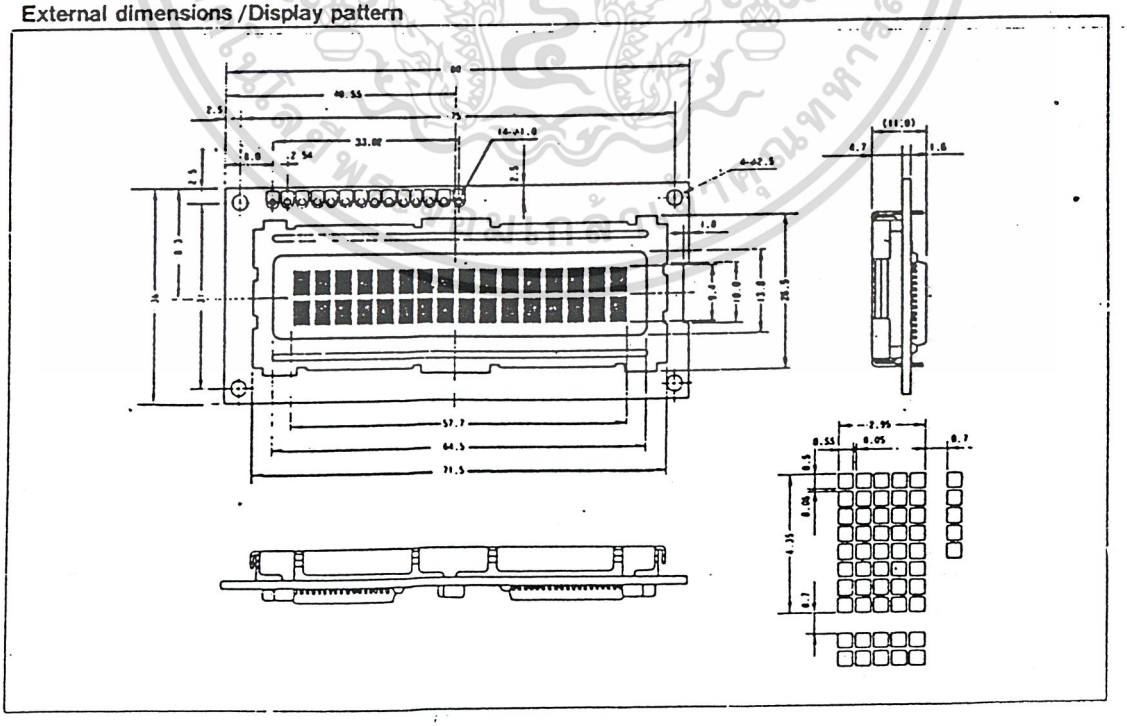
* Vcc=5.0V±5%, Ta=25°C

Block diagram





External dimensions /Display pattern



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

DV-16236

16 chars × 2 lines TN/STN Reflective/EL/LED Backlight

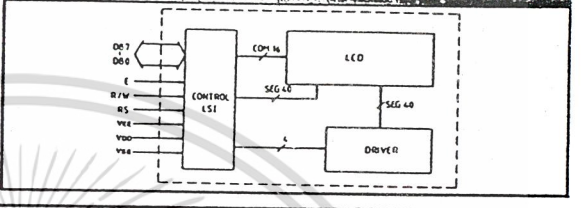
Item	Symbol	Standard Value			Unit
		Min	Typ.	Max.	
Supply Voltage for Logic	VDD-Vss	0	–	7.0	V
Supply Voltage for LCD Driver	VDD-VEE	–	–	13.5	V
Input Voltage	Vi	Vss	–	VDD	V
Operature Temp.	Topr	0	–	50	°C
Storage Temp.	Tstg	−20	–	70	°C

ELECTRICAL CHARACTERISTIC (REFLECTIVE TYPE)

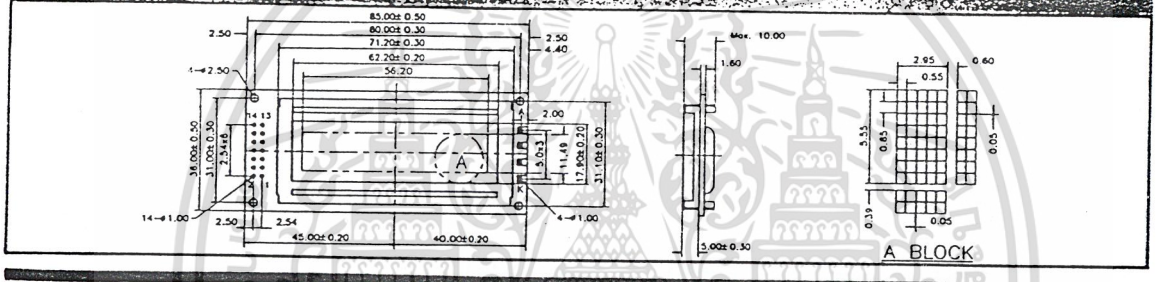
Item	Symbol	Test Condition	Standard Value			Unit
			Min.	Typ.	Max.	
Input "High" Voltage	VIH	–	2.2	–	VEE	V
Input "Low" Voltage	VIL	–	–	–	0.6	V
Output "High" Voltage	VOH	IOH = 0.2mA	2.2	–	–	V
Output "Low" Voltage	VOL	IOL = 1.2mA	–	–	0.4	V
Supply Current	IDD	VDD = 5.0A	–	1.5	4.0	mA

No	Symbol	Function	No	Symbol	Function
1	Vss	GND, OV	8	DB1	Data Bus
2	VDD	+5V	9	DB2	–
3	VEE	for LCD Drive	10	DB3	–
4	RS	Function Select	11	DB4	–
5	R/W	Read/Write	12	DB5	–
6	E	Enable Signal	13	DB6	–
7	DB0	Data Bus Line	14	DB7	–

BLOCK DIAGRAM



OUTLINE



DV-16244

16 chars × 2 lines TN/STN Reflective/EL/LED Backlight

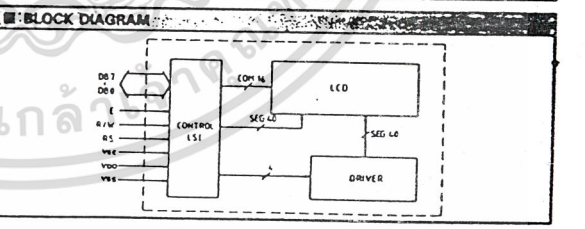
Item	Symbol	Standard Value			Unit
		Min	Typ.	Max.	
Supply Voltage for Logic	VDD-Vss	0	–	7.0	V
Supply Voltage for LCD Driver	VDD-VEE	–	–	13.5	V
Input Voltage	Vi	Vss	–	VDD	V
Operature Temp.	Topr	0	–	50	°C
Storage Temp.	Tstg	−20	–	70	°C

ELECTRICAL CHARACTERISTIC (REFLECTIVE TYPE)

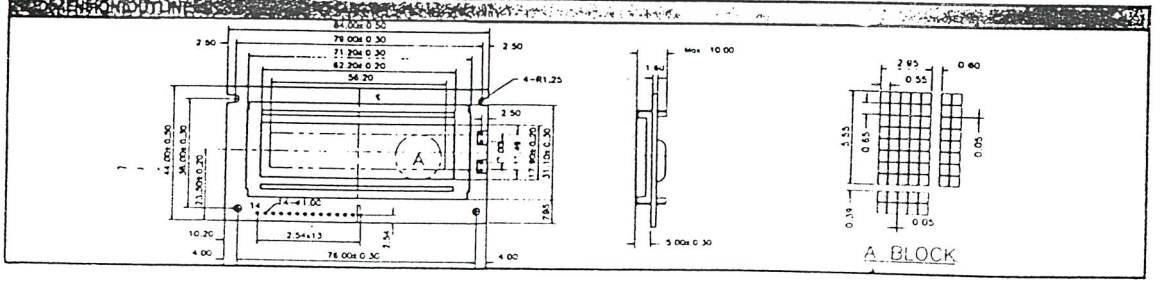
Item	Symbol	Test Condition	Standard Value			Unit
			Min.	Typ.	Max.	
Input "High" Voltage	VIH	–	2.2	–	VEE	V
Input "Low" Voltage	VIL	–	–	–	0.6	V
Output "High" Voltage	VOH	IOH = 0.2mA	2.2	–	–	V
Output "Low" Voltage	VOL	IOL = 1.2mA	–	–	0.4	V
Supply Current	IDD	VDD = 5.0A	–	1.5	4.0	mA

No	Symbol	Function	No	Symbol	Function
1	Vss	GND, OV	8	DB1	Data Bus
2	VDD	+5V	9	DB2	–
3	VEE	for LCD Drive	10	DB3	–
4	RS	Function Select	11	DB4	–
5	R/W	Read/Write	12	DB5	–
6	E	Enable Signal	13	DB6	–
7	DB0	Data Bus Line	14	DB7	–

BLOCK DIAGRAM



OUTLINE



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

DV series

DV-16244 16 characters x 2 lines

■ Mechanical Data

Item	Standard Value	Unit
Module Dimensions	84.0(W) X 44.0(H) X 10.0 (D)	mm
Viewing Area	62.2(W) X 17.9(H)	mm
Dot Size	0.55(W) X 0.65(H)	mm
Dot Pitch	0.60(W) X 0.70(H)	mm

■ Absolute Maximum Ratings

Item	Symbol	Standard Value			Unit
		Min.	Typ.	Max.	
Supply Voltage for Logic	V_{dd}-V_{ss}	-0.3	–	7.0	V
Supply Voltage for LCD Drive	V_{dd}-V_{ee}	V_{dd}-13.5	–	V_{dd}+0.3	V
Input Voltage	V_I	-0.3	–	V_{dd}+0.3	V
Operation Temperature	Topr	0	–	50	°C
Storage Temperature	Tstg	-20	–	70	°C

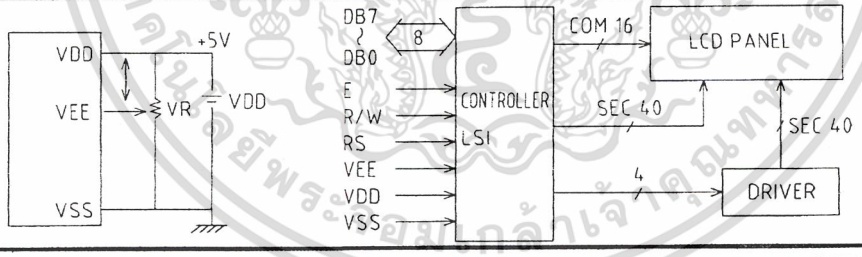
■ Electrical Characteristics

Item	Symbol	Condition	Standard Value			Unit
			Min.	Typ.	Max.	
Input "High" Voltage	V_{IH}		2.2	–	V_{dd}	V
Input "Low" Voltage	V_{IL}	$-I_{OH}$=0.205mA	0	–	0.6	V
Output "High" Voltage	V_{OH}	I_{OL}=1.2mA	2.4	–	V_{dd}	V
Output "Low" Voltage	V_{OL}		0	–	0.4	V
Supply Current	I_{dd}	V_{dd}= 5V	–	1.8	5.0	mA

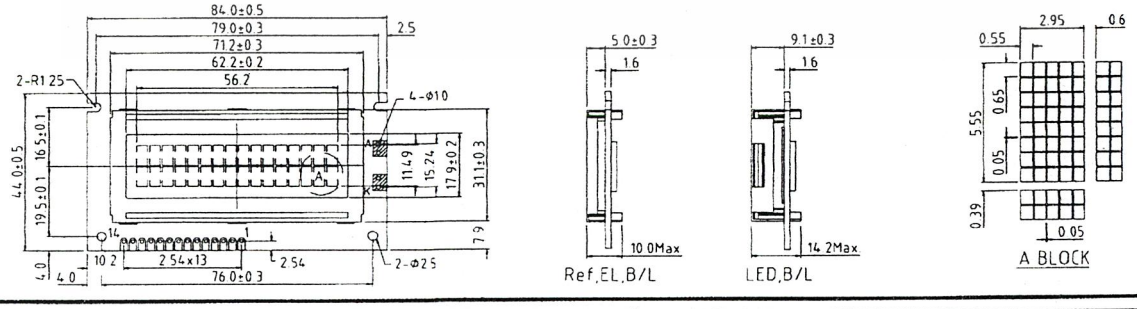
■ Pin Assignment

No.	Symbol	Function	No.	Symbol	Function
1	Vss	GND,0V	10	DB3	Data Bus Line
2	V_{DD}	+5V	11	DB4	Data Bus Line
3	V_{EE}	For LCD Drive	12	DB5	Data Bus Line
4	RS	Function select	13	DB6	Data Bus Line
5	R/W	Read/Write	14	DB7	Data Bus Line
6	E	Enable signal	15	LED A	LED power supply
7 -9	DB0-DB2	Data bus line	16	LED K	LED power supply

■ Block Diagram



■ External Dimensions



■ Option

LCD Type								Backlight Type				Built-in Control LSI
S1	S2	W	N	R	F	M	H	EL	LED	CCFL	Touch Panel	
•	•			•	•		•	•	•			KS-0066U or equivalent

Remarks: S1 : yellow-green STN LCD , S2 : gray STN LCD, W : black & white LCD, N : negative type LCD, R : reflective type LCD, F : transflective type LCD, M : transmissive type LCD, H : extended temperature type LCD (-20 °C ~ 70 °C)

ทั้งสิน อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้