สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

เครื่องควบคุมอุปกรณ์ไฟฟ้าภายในบ้านแบบโปรแกรมได้

PROGRAMMABLE ELECTRIC APPLIANCE CONTROLLER

T 050207

โดย

นายเฉลิมพล บวรกิติวงศ์

นายธีรวัฒน์ คุณโชติ

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เลขหมู่.....
เลขทะเบียน.....50207
วัน,เดือน,ปี.....27 เม.ย. 2547

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

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิศวกรรมระบบควบคุม

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2545

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์ปีการศึกษา2545

ภาควิชาวิศวกรรมระบบควบคุม

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง เครื่องควบคุมอุปกรณ์ไฟฟ้าภายในบ้านแบบโปรแกรมได้

ผู้จัดทำ

1. นายเฉลิมพล บวรกิติวงศ์
2. นายธีรวัฒน์ คุณโชติ

.....อาจารย์ที่ปรึกษา

(.....)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เครื่องควบคุมอุปกรณ์ไฟฟ้าภายในบ้านแบบโปรแกรมได้

นายเฉลิมพล บวรกิติวงศ์
นายธีรวัฒน์ คุณโชติ
ผศ.ดร.เกียรติศักดิ์ คมวัชระ อาจารย์ปรึกษา

ปีการศึกษา 2545

บทคัดย่อ

โครงงานนี้มีวัตถุประสงค์เพื่อใช้ควบคุมเครื่องใช้ไฟฟ้าภายในบ้านให้ทำงานได้ตามเวลาที่กำหนดเพื่อให้เกิดความปลอดภัยและเป็นการใช้พลังงานให้เกิดประโยชน์สูงสุด โดยเป็นการนำไมโครคอนโทรลเลอร์ตระกูล MCS-51 เป็นส่วนประมวลผลการควบคุมเครื่องใช้ไฟฟ้า ซึ่งมีวงจรฐานเวลาอ้างอิงเป็นตัวอ้างอิงการทำงานของเครื่องใช้ไฟฟ้าให้ทำงานได้ตรงตามเวลาที่แท้จริง ลักษณะของงานจะสามารถควบคุมการทำงาน โดยป้อนค่าเวลาที่ต้องการเปิด-ปิดเครื่องใช้ไฟฟ้าผ่านทางคีย์บอร์ด แล้วนำค่าที่ได้ไปแสดงผลทางจอ LCD และนำไปประมวลผลเพื่อควบคุมการเปิด-ปิดวงจรรีเลย์ ให้เครื่องใช้ไฟฟ้าต่ออยู่กับระบบทำงานได้ตามค่าเวลาที่ได้กำหนดไว้

Abstrat

The purpose of this project is to be able to control and manage electric appliance to work in time fixed for safety and able to use energy profit. That used to be the MCS-51 microcontroller in central processing unit for control electric appliance by real time clock is time reference, real time clock is time reference for control electric appliance to work in real time. The control can be select time from keyboard for on or off electric appliance and show time at LCD display. Finally data from keyboard will process in microcontroller for control relay to give electric appliance on or off along time fixed

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

หน้า

สารบัญ ง

สารบัญรูปภาพ ช

สารบัญตาราง ฎ

บทที่ 1 บทนำ 1

1.1 ความเป็นมา 1

1.2 วัตถุประสงค์ของโครงงาน 1

1.3 ขอบเขตการทำงานของโครงงาน 2

1.4 ขั้นตอนการดำเนินงาน 3

1.5 ประโยชน์ที่ได้รับจากโครงงาน 3

บทที่ 2 ทฤษฎีและหลักการ 4

2.1 โครงสร้างและสถาปัตยกรรมของไมโครคอนโทรลเลอร์ MCS-51แบบแฟลช 4

2.2 การนำพอร์ตมาใช้งาน 10

2.3 จังหวะการทำงานของไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลช 12

2.4 กระบวนการรีเซ็ตในไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลช 14

2.5 วงจรนาฬิกาของไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลช 16

2.6 การจัดหน่วยความจำของไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลช 17

2.6.1 หน่วยความจำโปรแกรม (Program Memory) 17

2.6.2 หน่วยความจำข้อมูล (Data Memory) 18

2.7 หลักการของระบบบัส I^2C 24

2.7.1 ความรู้เบื้องต้นเกี่ยวกับ I^2C 24

2.7.2 คุณสมบัติโดยทั่วไปของบัส I^2C 25

2.7.3 หลักการของบัส I^2C 26

2.7.4 สภาวะที่เกิดขึ้นบนบัส I^2C 27

2.7.5 การทำงานบนบัส I^2C 28

2.7.6 การอ้างถึงแบบ 7 บิต (7-bit addressing) 28

2.7.7 การอ้างถึงแบบ 10 บิต 29

2.7.8 อุปกรณ์ที่ใช้เชื่อมต่อแบบบัส I^2C 31

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

หน้า

2.7.9 การต่ออุปกรณ์ระบบบัส I^2C กับไมโครคอนโทรลเลอร์ MCS-51 31

2.8 ไอซีสร้างฐานเวลาจริงเบอร์ DS1307 (RTC : Real Time Clock) 32

2.8.1 รายละเอียดการต่อใช้งานของ DS1307 32

2.8.2 การทำงานของ DS1307 33

2.8.3 การจัดสรรหน่วยความจำใน DS1307 34

2.8.4 รีจิสเตอร์ควบคุม 35

2.8.5 โหมดการทำงานของ DS1307 35

2.9 LCD (Liquid Crystral Displays) 36

2.9.1 รายละเอียดเกี่ยวกับโมดูล LCD 36

2.9.2 โครงสร้างภายในของตัวควบคุมโมดูล LCD 37

2.9.3 โมดูล LCD ขนาด 20 ตัวอักษร 4 บรรทัด (LCD 20X4) 38

2.9.4 คำสั่งควบคุมโมดูล LCD 38

2.9.5 การเขียนคำสั่งและข้อมูลให้แก่ส่วนแสดงผลแบบผลึกเหลว 41

2.9.6 จังหวะการทำงานของส่วนแสดงผลแบบผลึกเหลว 42

บทที่ 3 การคำนวณและการสร้าง 43

3.1 แนวคิดและหลักการทำงาน 43

3.2 วงจรฐานเวลาอ้างอิง (Real Time Clock : RTC) 44

3.2.1 การเขียนข้อมูลลงไอซีสร้างฐานเวลาจริง (Real Time Clock) DS1307 46

3.2.2 การอ่านค่าจากไอซีสร้างฐานเวลาจริง (Real Time Clock) DS1307 47

3.3 วงจร Scan Keyboard 48

3.4 วงจร LCD 51

3.4.1 การติดต่อกับ LCD 52

3.5 วงจรหน่วยความจำข้อมูลภายนอก 54

3.6 วงจร Relay Output 55

บทที่ 4 การทดลองและผลการทดลอง 58

4.1 การตั้งค่าวันและตั้งค่าเวลา 60

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

หน้า

4.2 การเปลี่ยนรหัสผ่าน 62
4.3 การป้อนโปรแกรมเปิด-ปิดอุปกรณ์ไฟฟ้าภายในบ้าน 64
4.3.1 การตรวจสอบโปรแกรม(Check Program) 64
4.3.2 การตั้งค่าหรือการกำหนดโปรแกรม (Edit Program) 66
4.3.3 การลบโปรแกรม (Clear Program) 68
ผลการทดลอง 71
บทที่ 5 บทวิจารณ์และสรุป 72
ภาคผนวก
กิตติกรรมประกาศ ฎ
หนังสืออ้าง ฐ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูปภาพ

หน้า

รูปที่2.1 โครงสร้างพื้นฐานของไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลชในอนุกรม AT89Cxx 5
รูปที่ 2.2 รายละเอียดโครงสร้างหลักของไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลช 6
รูปที่ 2.3 การจัดขามาตรฐานของไมโครคอนโทรลเลอร์ MCS-51 6
รูปที่ 2.4 วงจรภายในของพอร์ต 0 ในไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลช 7
รูปที่ 2.5 วงจรภายในของพอร์ต 1 ในไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลช 7
รูปที่ 2.6 วงจรพูลอัปภายในพอร์ตของไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลช 8
รูปที่ 2.7 วงจรภายในของพอร์ต 2 ในไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลช 8
รูปที่ 2.8 วงจรภายในของพอร์ต 3 ในไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลช 9
รูปที่ 2.9ก ไซเคิลการทำงานของไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลช 12
รูปที่ 2.9ข ไดอะแกรมเวลาแสดงการติดต่อหน่วยความจำโปรแกรมภายนอกของ MCS-51 13
รูปที่ 2.10 ไดอะแกรมเวลาของกระบวนการรีเซตภายในไมโครคอนโทรลเลอร์ MCS-51 14
รูปที่ 2.11 วงจรพาวเวอร์ออนรีเซต 16
รูปที่ 2.12 แสดงการต่อวงจรนาฬิกาแบบภายในชิพของไมโครคอนโทรลเลอร์ MCS-51 16
รูปที่ 2.13 แสดงการต่อวงจรนาฬิกาจากภายนอกของไมโครคอนโทรลเลอร์ MCS-51 17
รูปที่ 2.14 แสดงการจัดพื้นที่หน่วยความจำโปรแกรมสำหรับไมโครคอนโทรลเลอร์เบอร์ AT89S53 18
รูปที่ 2.15 แสดงการจัดสรรพื้นที่ของหน่วยความจำข้อมูลภายไมโครคอนโทรลเลอร์ 19
รูปที่ 2.16 แสดงหน่วยความจำข้อมูลภายในบริเวณที่อ้างถึงได้แบบบิต 21
รูปที่ 2.17 แสดงรีจิสเตอร์ใช้งานพิเศษ 22
รูปที่ 2.18 ผังแสดงการเชื่อมต่อของอุปกรณ์ต่างๆบนบัส I^2C 25
รูปที่ 2.19 แสดงวงจรเอาต์พุตของอุปกรณ์ในระบบบัส I^2C 25
รูปที่ 2.20 การต่อตัวต้านทานพูลอัพบนสายสัญญาณในระบบบัส I^2C 26
รูปที่ 2.21การต่อตัวต้านทาน Rs เพื่อลดสัญญาณรบกวนขนาดใหญ่ที่อาจเข้ามาในบัส I^2C 26
รูปที่ 2.22 ไดอะแกรมเวลาแสดงสถานะต่าง ๆ ในบัส I^2C 28
รูปที่ 2.23 รูปแบบของข้อมูลกำหนดแอดเดรสที่ใช้ในการอ้างถึงแบบ 7 บิต 29

เอกสารนี้สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูปภาพ (ต่อ)

หน้า

รูปที่ 2.24 รูปแบบของข้อมูลอนุกรมที่ใช้ติดต่อกับอุปกรณ์บัส I^2C เมื่อใช้การอ้างถึงแบบ 7 บิต 30
รูปที่ 2.25 รูปแบบของข้อมูลอนุกรมที่ใช้ติดต่อกับอุปกรณ์บัส I^2C เมื่อใช้การอ้างถึงแบบ 10 บิต 30
รูปที่ 2.26 วงจรตัวอย่างการต่อไมโครคอนโทรลเลอร์ MCS-51 กับอุปกรณ์ระบบบัส I^2C 31
รูปที่ 2.27 การจัดขาของไอซี DS1307 32
รูปที่ 2.28.โครงสร้างภายในของไอซีรีลไทม์คล็อกเบอร์ DS1307 33
รูปที่ 2.29 (ก) การจัดสรรหน่วยความจำแรมภายใน DS1307 34
(ข) รายละเอียดของรีจิสเตอร์เก็บค่าเวลาและรีจิสเตอร์ควบคุมของ DS1307
รูปที่ 2.30 รูปแบบ ของข้อมูลสำหรับติดต่อกับ DS1307 ในโหมดการเขียนข้อมูล 36
รูปที่ 2.31 รูปแบบ ของข้อมูลสำหรับติดต่อกับ DS1307 ในโหมดการอ่านข้อมูล 36
รูปที่ 3.1 บล็อกไดอะแกรมของเครื่องควบคุมอุปกรณ์ไฟฟ้าแบบโปรแกรมได้ 43
รูปที่ 3.2 วงจรการต่อใช้งานไอซีสร้างฐานเวลาจริง (Real Tie Clock : RTC) 45
รูปที่ 3.3 Flowchart การเขียนข้อมูลลง Real Time Clock 46
รูปที่ 3.4 Flowchart การอ่านข้อมูลจากไอซี Real Time Clock 47
รูปที่ 3.5 วงจรการเชื่อมต่อ Key Pad โดยใช้ IC Scankey 74C922 48
รูปที่ 3.6 Flowchart การติดต่อกับ IC Scankey 16 Key Encoder 50
รูปที่ 3.7 วงจรการเชื่อมต่อ LCD ขนาด 20 ตัวอักษร 4 บรรทัด แบบ 8 บิต 51
รูปที่ 3.8 Flowchart การกำหนดค่าเริ่มต้นให้กับ LCD 52
รูปที่ 3.9 Flowchart การส่งค่าคำสั่งแบบ 8 บิต 53
รูปที่ 3.10 Flowchart การส่งค่าข้อมูลแบบ 8 บิต 53
รูปที่ 3.11 วงจรการเชื่อมต่อหน่วยความจำข้อมูลภายนอกของไมโครคอนโทรลเลอร์ 54
รูปที่ 3.12 วงจรการต่อใช้งาน Relay ร่วมกับไมโครคอนโทรลเลอร์ 55
รูปที่ 3.13 Flowchart ของระบบทั้งหมด 56
รูปที่ 4.1 ตัวอย่างการแสดงผลที่ LCD Display 59

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูปภาพ (ต่อ)

หน้า

รูปที่ 4.2 ตัวอย่างการแสดงผลที่ LCD Display เมื่อทำการกดคีย์ HELP เพื่อตรวจสอบสถานะของเครื่องใช้ไฟฟ้า 59
รูปที่ 4.3 แสดงหน้าจอเมื่อทำการกดคีย์ใด ๆ (ยกเว้นคีย์ HELP) 60
รูปที่ 4.4 แสดงหน้าจอเพื่อให้กดรหัสผ่าน 60
รูปที่ 4.5 แสดงหน้าจอ LCD เมื่อกดรหัสผ่านผิด 61
รูปที่ 4.6 แสดงหน้าจอLCD เมื่อกดรหัสผ่านถูกต้อง 61
รูปที่ 4.7 แสดงหน้าจอLCD ซึ่งเป็นเมนูการตั้งค่าวันและเวลา 61
รูปที่ 4.8 แสดงหน้าจอ LCD ซึ่งเป็นเมนูการตั้งเวลา 61
รูปที่ 4.9 แสดงหน้าจอ LCDที่เป็นเมนู การตั้งค่าวัน 62
รูปที่ 4.10 แสดงหน้าจอ LCDแจ้งให้ทราบว่าจะทำการเปลี่ยนรหัสผ่าน 63
รูปที่ 4.11 แสดงหน้าจอ LCD รับค่ารหัสผ่านเก่า 63
รูปที่ 4.12 แสดงหน้าจอLCDรับค่ารหัสผ่านใหม่ (เมื่อกดรหัสผ่านเก่าถูกต้อง) 63
รูปที่ 4.13 แสดงหน้าจอLCDรับค่ารหัสผ่านเก่า 63
รูปที่ 4.14 แสดงหน้าจอ LCD แจ้งให้ทราบว่าการเปลี่ยนรหัสผ่านเสร็จสมบูรณ์ 64
รูปที่ 4.15 แสดงหน้าจอ LCD ซึ่งแสดงเมนูการทำงานเกี่ยวกับการเปิดปิดอุปกรณ์ไฟฟ้าภายในบ้าน 64
รูปที่ 4.16 แสดงหน้าจอ LCD ซึ่งใช้เพื่อเลือกอุปกรณ์ไฟฟ้าที่ต้องการติดต่อ (ตรวจสอบหรือตั้งค่าโปรแกรม) 65
รูปที่ 4.17 แสดงหน้าจอ LCD ใช้เพื่อเลือกวัน (SUN – SAT) หลังจากที่เลือกอุปกรณ์ไฟฟ้าแล้ว 65
รูปที่ 4.18 แสดงตัวอย่างหน้าจอ LCD ที่แสดงโปรแกรมสำหรับควบคุมการเปิด-ปิดอุปกรณ์ไฟฟ้าที่ได้ป้อนเอาไว้ 66
รูปที่ 4.19 แสดงหน้าจอ LCD เมื่อกดคีย์ “▽“ 1 ครั้ง 66
รูปที่ 4.20 แสดงหน้าจอ LCD เมื่อกดคีย์ “▽“ 2 ครั้ง 66
รูปที่ 4.21 แสดงหน้าจอ LCD เมื่อต้องการแก้ไขโปรแกรม (ในที่นี้คือโปรแกรมที่ 1) 67
รูปที่ 4.22 แสดงหน้าจอ LCD เมื่อได้ทำการแก้ไขโปรแกรมเรียบร้อยแล้ว 67
รูปที่ 4.23 แสดงหน้าจอ LCD เมื่อต้องการเพิ่มโปรแกรมที่ 4 ให้กับอุปกรณ์ไฟฟ้า 68

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้ในการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูปภาพ (ต่อ)

หน้า

รูปที่ 4.24 แสดงหน้าจอ LCD เมื่อตั้งค่าโปรแกรมเสร็จสมบูรณ์และรอการตั้งค่าครั้งต่อไป 68
รูปที่ 4.25 แสดงหน้าจอ LCD ที่แสดงเมนูย่อยการลบโปรแกรมของอุปกรณ์ไฟฟ้า 69
รูปที่ 4.26 แสดงหน้าจอ LCD เพื่อแจ้งให้ทราบว่าได้ทำการลบโปรแกรมแล้ว 69
รูปที่ 4.27 แสดงหน้าจอ LCD เมื่อทำการลบโปรแกรมที่ 2 เรียบร้อยแล้ว 70
รูปที่ 4.28 แสดงหน้าจอ LCD เมื่อทำการลบโปรแกรมที่ 2 (โปรแกรมที่ 3 เดิม) เรียบร้อยแล้ว 70
รูปที่ 4.29 แสดงการตั้งค่าเวลาของโปรแกรมซ้ำกัน แต่ลักษณะการเปิด-ปิดแตกต่างกัน 71

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

หน้า

ตารางที่ 2.1 รายละเอียดบางส่วนของไมโครคอนโทรลเลอร์ ตระกูล MCS-51แต่ละเบอร์ 5
ตารางที่ 2.2 แสดงผลของการรีเซตที่มีต่อรีจิสเตอร์ภายในไมโครคอนโทรลเลอร์ MCS-51 15
ตารางที่ 2.3 ตารางแสดงกลุ่มแอดเดรสที่ถูกใช้งานในฐานะรีจีสเตอร์ใช้งานทั่วไป 20
ตารางที่ 2.4 ตารางแสดงการสวิตช์เลือกแต่ละกลุ่มของรีจีสเตอร์แบงค์ 20
ตารางที่ 3.1 แสดงค่าเอาท์พุตที่ได้จาก IC Scankey 74C922 49

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

1.1 ความเป็นมา

ปัจจุบันประเทศไทยมีการเจริญเติบทางเศรษฐกิจอย่างรวดเร็ว ประชากรก็มีจำนวนเพิ่มมากขึ้น มีรายได้เพิ่มขึ้น ก่อให้เกิดความเจริญทางด้านวัตถุ มีการนำเอาเทคโนโลยีใหม่ ๆ มาใช้ ทำให้ประชากรส่วนใหญ่มีความสะดวกสบายเพิ่มมากขึ้น

เนื่องจากในปัจจุบันนี้ภายในบ้านหรือสถานที่ต่างๆ ไม่ว่าจะเป็น หอพัก อพาร์ทเม้น คอนโดมิเนียม โรงเรียน ที่ทำงาน ฯลฯ เราจะพบว่ามีอุปกรณ์เครื่องใช้ไฟฟ้าต่างๆ มากมายอยู่ในนั้นและถ้าหากเราสังเกตให้ดีจะพบว่า เราจะต้องทำการเปิด-ปิดอุปกรณ์เครื่องใช้ไฟฟ้าบางอย่าง ในเวลาเดิมทุกวันเป็นประจำ เช่น การเปิด-ปิดไฟที่บ้านหรือสถานที่ทำงาน การเปิดและปิดเครื่องปรับอากาศ ฯลฯ

โครงงานชิ้นนี้จึงได้ออกแบบขึ้นมาเพื่อช่วยอำนวยความสะดวกในการที่เราต้องทำอะไรซ้ำๆ อย่างนี้เกือบจะทุกวัน ซึ่งบางครั้งเราอาจจะลืมปิดไฟ หรือเครื่องปรับอากาศเวลาเราออกจากสถานที่นั้นๆ ไปแล้วทำให้ต้องสูญเสียพลังงาน และค่าใช้จ่ายเพิ่มในการที่เราต้องเปิดเครื่องใช้ไฟฟ้านั้นๆ ทิ้งไว้หรือจะเป็นการช่วยอำนวยความสะดวกขณะที่เราไม่อยู่บ้าน อุปกรณ์นี้จะช่วยเปิดไฟหน้าบ้านในตอนกลางคืนและปิดไฟในตอนเช้า เสมือนว่ามีคนอยู่ที่บ้านของเราเป็นการรักษาความปลอดภัยระดับหนึ่งเมื่อเราไม่อยู่บ้าน

ทางผู้จัดทำจึงหวังว่า โครงงานชิ้นนี้จะเป็นประโยชน์ไม่มากก็น้อยสำหรับผู้ที่สนใจและโครงงาน ชิ้นนี้ยังได้นำเสนอการทำงานของไมโครคอนโทรลเลอร์ ตระกูล 8051 ให้กับผู้ที่สนใจอีกด้วย

1.2 วัตถุประสงค์ของโครงงาน

โครงงานนี้มีวัตถุประสงค์เพื่ออำนวยความสะดวกและให้การใช้ชีวิตประจำวันเกิดความปลอดภัยมากยิ่งขึ้นอีกทั้งยังเป็นการช่วยอนุรักษ์พลังงานการใช้ทรัพยากร พลังงานไฟฟ้าให้คุ้มค่าอีกรูปแบบหนึ่งด้วย ซึ่งเครื่องควบคุมอุปกรณ์ไฟฟ้าภายในบ้านแบบโปรแกรมได้นี้จะสามารถทำการกำหนดค่าเวลาในการทำงานของเครื่องใช้ไฟฟ้าให้ทำงานได้ตามเวลาที่กำหนดถึงแม้ว่าผู้ใช้จะไม่อยู่บ้านโดยผู้ใช้จะสามารถทำการตั้งค่าเวลาต่าง ๆ ไว้ให้กับเครื่องใช้ไฟฟ้าล่วงหน้าได้เป็นวันหรือเป็นสัปดาห์ได้ โดยสามารถควบคุมได้ 8 อุปกรณ์ และแต่ละอุปกรณ์สามารถตั้งโปรแกรมได้อีก 30 โปรแกรมต่อ 1 วัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวอย่างการประยุกต์ใช้งาน

- สามารถเปิด-ปิดอุปกรณ์ในสำนักงาน โดยสามารถตั้งเวลาเปิด-ปิดได้อย่างอิสระ และตั้งให้การทำงานของอุปกรณ์ต่างๆในแต่ละวันแตกต่างกันตามความต้องการได้
- ใช้ควบคุมขบวนการผลิตในโรงงาน โดยตั้งเปิด-ปิดเครื่องจักรต่างๆ ที่ทำงานในช่วงเวลาต่างกัน หรืออาจตั้งเพื่อการอุ่นเครื่องจักรก่อนเวลาเข้างาน เพื่อให้เริ่มงานได้ทันทีเมื่อเข้างาน
- ใช้เพื่อการประหยัดพลังงานในบ้านและสำนักงาน โดยควบคุมเครื่องทำน้ำเย็น เครื่องปรับอากาศ ให้เปิด-ปิดในเวลาที่เหมาะสม ไม่ต้องเปิดทิ้งไว้ให้เปลืองไฟ
- ใช้ควบคุมในการเกษตรต่างๆ เช่นการเปิด-ปิดปั๊มน้ำที่บ่อปลาหรือเครื่องเติมอากาศในบ่อกุ้ง หรือการเปิด-ปิดปั๊มน้ำหรือวาล์วน้ำเพื่อการลดน้ำต้นไม้ต่างๆ
- ใช้เพื่อการเปิด-ปิดในสถานที่ที่ห่างไกลหรือไม่สะดวกต่อการดูแล

1.3 ขอบเขตการทำงานของโครงงาน

1.3.1 สามารถตั้งเวลาปัจจุบันและเวลาที่ควบคุมเครื่องใช้ไฟฟ้าได้จากแป้นสวิตช์ Keypad

1.3.2 ทำงานทั้งแบบ Manual Mode และ แบบ Program Mode

1.3.3 ควบคุมการทำงานด้วยไมโครคอนโทรลเลอร์

1.3.4 แสดงผลทางส่วนแสดงผลแบบผลึกเหลว

1.3.5 มีแหล่งจ่ายไฟสำรองในกรณีที่ไฟฟ้าดับ

คุณสมบัติทางเทคนิคของอุปกรณ์คือ

- ใช้กับอุปกรณ์ไฟฟ้า 220 VAC กระแส 10 แอมแปร์
- แสดงผลโดยส่วนแสดงผลแบบผลึกเหลว ขนาด 20 ตัวอักษร 4 บรรทัด และมีไฟ Black Light เพื่อความสะดวกในการใช้งานในที่มืด
- ส่วนแสดงผลจะทำการแสดง วัน, วันที่, เดือน, ปี, ชั่วโมง, นาที และวินาที ตลอดเวลา
- สามารถกดดูสถานะการเปิด-ปิดของเอาท์พุทแต่ละช่องได้จากหน้าจอ LCD
- มีคีย์บอร์ดเพื่อป้อนค่าต่างๆ ขนาด 16 คีย์ (4X4)
- มีระบบ Backup สามารถเก็บเวลาและโปรแกรมไว้ได้ แม้ขณะที่ไฟฟ้าดับ และมี Watchdog Timer ซึ่งสามารถรีเซตไมโครคอนโทรลเลอร์อัตโนมัติในกรณีที่เกิดสภาวะหยุดทำงานกระทันหันหรือแฮงก์ (Hang)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- Output สามารถต่อพ่วงกับสวิทย์เดิมที่มีอยู่ได้ทันที หรือเลือกให้ Output เป็นแบบ Relay ได้ โดยปรับแก้ที่ Hardware บางส่วนเท่านั้น
- มีฟังก์ชัน Manual Mode ใช้ในกรณีเช่น ทำการซ่อมเครื่องใช้ไฟฟ้า หรือในวันหยุดนักขัตฤกษ์ ที่ไม่ต้องการให้เครื่องใช้ไฟฟ้าทำงานในวันนั้นๆ ก็สามารถกำหนดให้อุปกรณ์ทั้งหมดทำงานแบบ Manual Mode ได้อย่างง่ายดายโดยไม่ต้องตั้งโปรแกรมใหม่ และไม่ต้องถอดปลั๊ก

1.4 ขั้นตอนการดำเนินงาน

1.4.1 ศึกษารายละเอียดของโครงงาน

1.4.2 เสนอโครงงาน ออกแบบและทดสอบเขียนโปรแกรมควบคุมการทำงาน

1.4.3 จัดทำปริญญานิพนธ์

1.5 ประโยชน์ที่ได้รับจากโครงงาน

ประโยชน์ที่ได้รับ จากโครงงานนี้ คือความรู้และประสบการณ์ จากการศึกษาและวิจัยการทำงานของวงจรต่าง ๆ การเขียนภาษาแอสเซมบลี บนไมโครคอนโทรลเลอร์ การใช้โปรแกรมโพล-เทล ในการออกแบบลายวงจร การนำความรู้ที่ได้จากการศึกษาภาคทฤษฎี ในชั้นเรียนมาใช้ในการปฏิบัติงานจริง และยังสามารถนำโครงงานนี้ไปใช้ประโยชน์ในชีวิตประจำวันได้ด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2
ทฤษฎีและหลักการ

2.1 โครงสร้างและสถาปัตยกรรมของไมโครคอนโทรลเลอร์ MCS-51แบบแฟลช

ไมโครคอนโทรลเลอร์แบบชิพเดี่ยวตระกูล MSC-51 ที่นำมาใช้ มีหน่วยความจำภายในเป็นแบบแฟลช (Flash Memory) ของยี่ห้อ Atmel Corporation มีเบอร์ขึ้นต้นด้วย AT89 ซึ่งมีคุณลักษณะพิเศษดังนี้

- หน่วยความจำโปรแกรมภายในตัวไมโครคอนโทรลเลอร์ สามารถลบและเขียนใหม่ได้นับพันครั้งจึงสามารถใช้งานในรูปแบบของไมโครคอนโทรลเลอร์ชิพเดี่ยวไม่ต้องใช้หน่วยความจำภายนอก ส่งผลให้สามารถใช้งานพอร์ตอินพุตและพอร์ตเอาท์พุตของไมโครคอนโทรลเลอร์ได้อย่างเต็มประสิทธิภาพ
- ด้วยการใช้หน่วยความจำภายในตัวไมโครคอนโทรลเลอร์ทำให้สามารถป้องกันการคัดลอกข้อมูลของหน่วยความจำโปรแกรมได้อย่างดี
- ชุดคำสั่ง และสถาปัตยกรรมพื้นฐานเหมือนกับไมโครคอนโทรลเลอร์ MCS-51 ของผู้ผลิตอื่นไม่ว่าจะเป็นอินเทล, ซีเมนส์ หรือดัลลัส

คุณสมบัติทางเทคนิคของไมโครคอนโทรลเลอร์ตระกูล MCS-51 อนุกรม AT89XXX

1) เป็นไมโครคอนโทรลเลอร์ที่ใช้หน่วยประมวลผลกลาง (CPU) ขนาด 8 บิต
2) หน่วยความจำข้อมูลพื้นฐานเป็นหน่วยความจำแบบแรม ในบางเบอร์จะมีความจำแบบอีอี- พรอมเพิ่มเติม
3) ขาพอร์ต (PORT) เป็นแบบสองทิศทาง สามารถใช้งานเป็นได้ทั้งอินพุต และเอาท์พุต
4) มีวงจรสื่อสารอนุกรมแบบฟลูดูเพล็กซ์
5) ไทเมอร์/เคาน์เตอร์ (Timer/Counter) ขนาด 16 บิต อย่างน้อย 2 ตัว
6) สามารถรองรับแหล่งกำเนิดอินเตอร์รัปต์ได้ 6 ประเภท
7) สามารถขยายหน่วยความจำภายนอกเพิ่มเติมได้สูงสุด 64 กิโลไบต์
8) มีวงจรกำเนิดสัญญาณนาฬิกาอยู่ภายในชิพ
9) มีวงจรสื่อสารอนุกรมแบบ SPI สำหรับอนุกรม AT89SXX
10) มีวอตช์ด็อกไทเมอร์ในตัว สำหรับอนุกรม AT89SXX

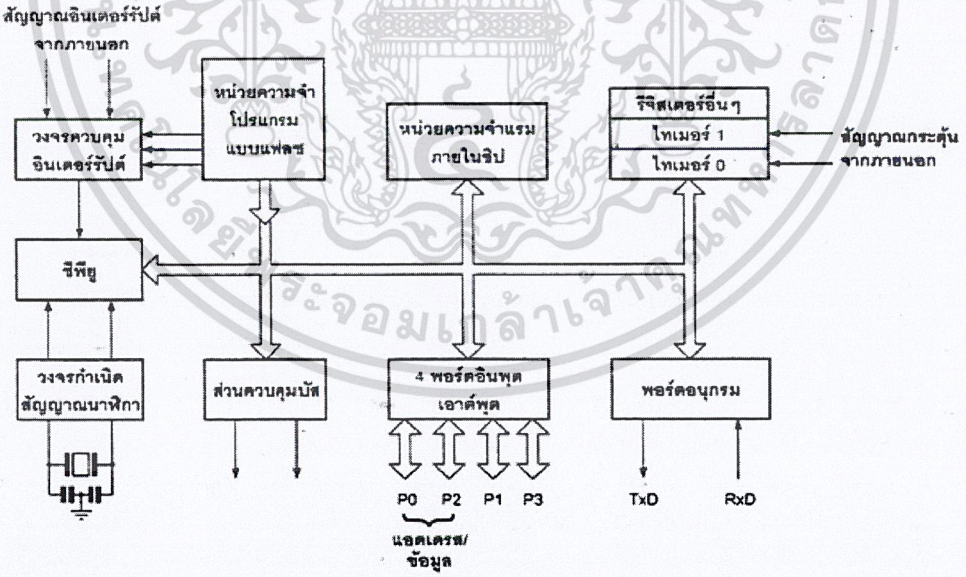
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในตารางที่ 2.1 แสดงรายละเอียดบางส่วนของไมโครคอนโทรลเลอร์ ตระกูล MCS-51 แต่ละเบอร์ที่ Atmel Corporation ผลิตขึ้น และมีใช้งานอยู่ในปัจจุบัน

เบอร์ของไมโครคอนโทรลเลอร์	หน่วยความจำโปรแกรม	หน่วยความจำข้อมูล	จำนวนไทเมอร์/เคาน์เตอร์ 16 บิต
AT89C1051	แบบแฟลช ขนาด 1 กิโลไบต์	แรม 64 ไบต์	1
AT89C2051	แบบแฟลช ขนาด 2 กิโลไบต์	แรม 128 ไบต์	2
AT89C51	แบบแฟลช ขนาด 4 กิโลไบต์	แรม 128 ไบต์	2
AT89C52	แบบแฟลช ขนาด 8 กิโลไบต์	แรม 256 ไบต์	3
AT89C55	แบบแฟลช ขนาด 20 กิโลไบต์	แรม 256 ไบต์	3
AT89S8252	แบบแฟลช ขนาด 8 กิโลไบต์	แรม 256 ไบต์ อีอีพรอม 2 กิโลไบต์	3
AT89S53	แบบแฟลช ขนาด 12 กิโลไบต์	แรม 256 ไบต์	3

ตารางที่ 2.1 รายละเอียดบางส่วนของไมโครคอนโทรลเลอร์ ตระกูล MCS-51แต่ละเบอร์

ในรูปที่ 2.1 แสดงโครงสร้างพื้นฐานของไมโครคอนโทรลเลอร์ MCS-51 ในอนุกรม AT89CXX จะเห็นได้ว่า โครงสร้างของ AT89CXX จะเหมือนกับไมโครคอนโทรลเลอร์ตระกูล MCS-51 พื้นฐาน หากแต่แตกต่างกันเฉพาะหน่วยความจำโปรแกรมแบบแฟลชเพิ่มเติมเข้ามา หากเป็นไมโครคอนโทรลเลอร์ในอนุกรม 87XX หน่วยความจำโปรแกรมภายในจะเป็นแบบอีอีพรอมและบางเบอร์สามารถโปรแกรมได้ครั้งเดียว

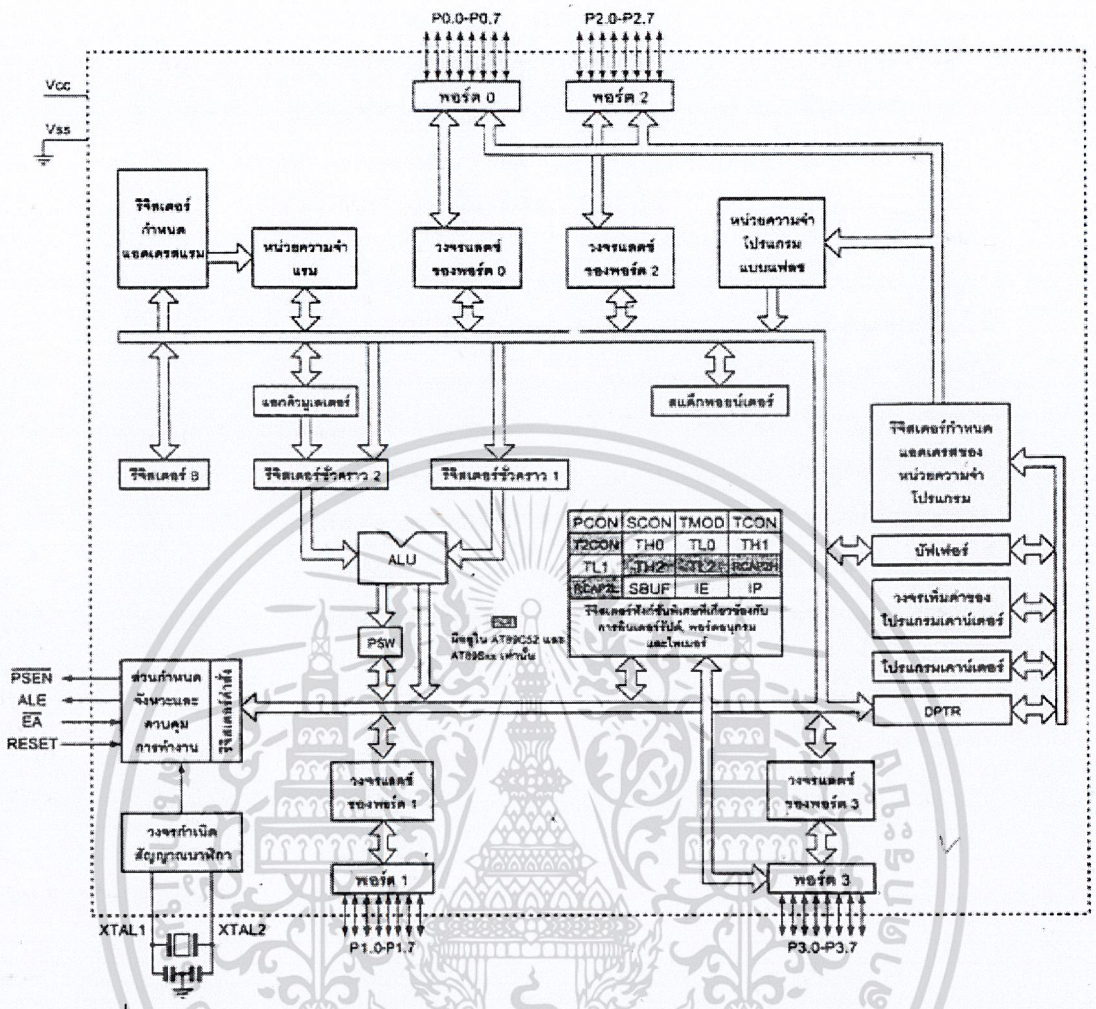


รูปที่2.1 โครงสร้างพื้นฐานของไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลชในอนุกรม AT89Cxx

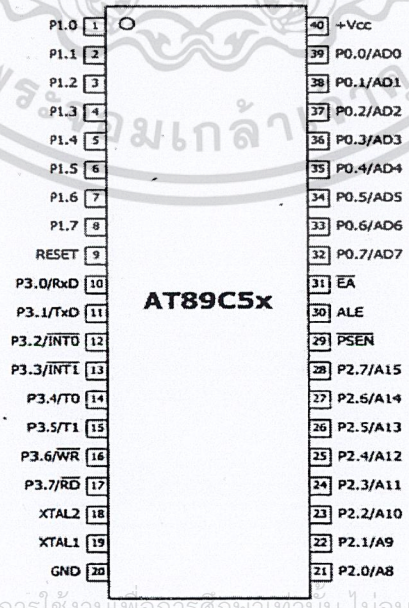
การจัดขาของไมโครคอนโทรลเลอร์ MCS-51

ไมโครคอนโทรลเลอร์ MCS-51 ทุกเบอร์จะมีสถาปัตยกรรมโครงสร้างภายในและการจัดวางขาใช้งานพื้นฐานเหมือนกันดังแสดงในรูปที่ 2.2 และรูปที่ 2.3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.2 รายละเอียดโครงสร้างหลักของไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลช

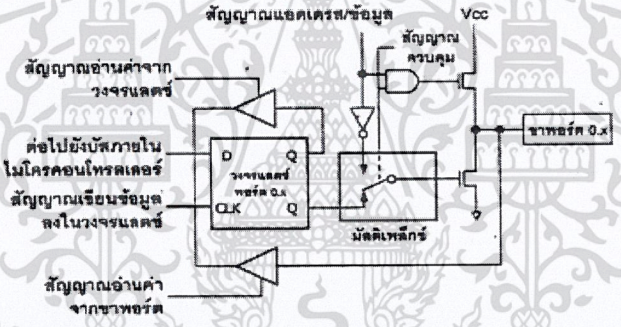


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 2.3 การจัดขามาตรฐานของไมโครคอนโทรลเลอร์ MCS-51

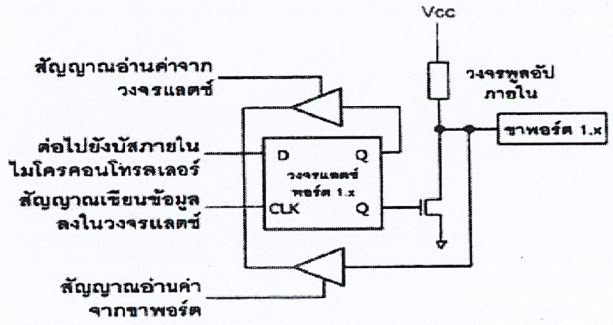
โดยรายละเอียดขาการใช้งานขั้นต้นมีดังนี้

- ขา V_{cc} (40) ต่อกับไฟเลี้ยง +5 V
- ขา GND (20) เป็นขากราวด์ สำหรับต่อกับกราวด์ของระบบ
- ขา PORT 0 (32-39) มีทั้งหมด 8 บิต (P0.0-P0.7) ใช้งานได้ 2 หน้าที่ คือ ส่งแอดเดรสไบต์ต่ำ (A0-A7) และข้อมูล (D0-D7) ออกไปให้หน่วยความจำภายนอก เมื่อทำการเขียนข้อมูลลงในหน่วยความจำภายนอกควบคุมด้วยขาควบคุม (Control) แสดงที่รูป 2.4 โดยใช้กระบวนการมัลติเพล็กซ์เข้าช่วย เพื่อสลับการทำงานให้เป็นได้ทั้งติดต่อแอดเดรสและส่งข้อมูล และอีกหน้าที่หนึ่งคือเป็นอินพุตพอร์ตหรือเอาท์พุตพอร์ตสำหรับใช้งานทั่วไป ถ้าต้องการกำหนดให้พอร์ต 0 ทำงานเป็นอินพุตพอร์ต ต้องส่งละจิก “1” ไปยังพอร์ตนี้ จะมีผลให้ $\overline{Q}$ ของดี-ฟลิปฟลอป (D-FF) เป็น “0” ทำให้เฟ็ต (FET) ตัวล่างมีสภาวะปิด (OFF) สัญญาณที่อ่านขาอินพุตพอร์ต (ขาอินพุตพอร์ต P0.X) จะใช้สัญญาณ READ Latch เมื่อถูกกระตุ้นที่ไตรสเตทบัฟเฟอร์ (Tri-State Buffer) ตัวบน



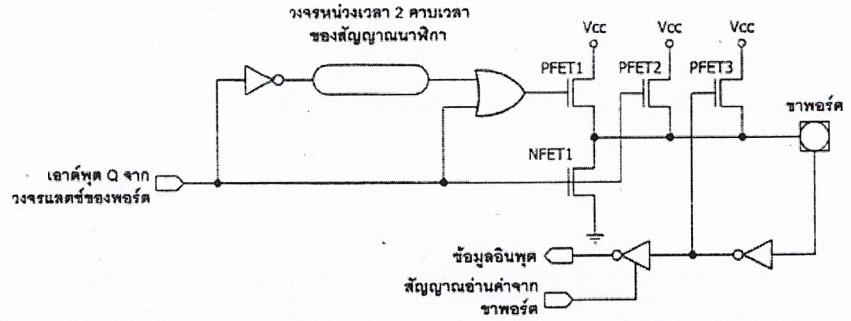
รูปที่ 2.4 วงจรภายในของพอร์ต 0 ในไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลช

- ขา PORT 1(1-8) มีทั้งหมด 8 บิต (P1.0-P1.7) มีโครงสร้างคล้ายกับพอร์ต 0 ดังรูปที่ 2.5 พอร์ตนี้ทำหน้าที่เป็นอินพุตพอร์ตหรือเอาท์พุตพอร์ตสำหรับใช้งานทั่วไป แต่จะใช้ความต้านทานภายในพูลอัพแทน (Internal pull-up) ซึ่งแสดงดังรูปที่ 2.6



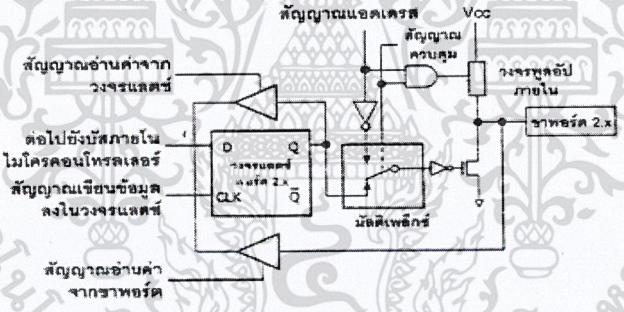
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 2.5 วงจรภายในของพอร์ต 1 ในไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลช



รูปที่ 2.6 วงจรพูลอัปภายในพอร์ตของไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลช

- ขา PORT 2 (21-28) มีทั้งหมด 8 บิต (P2.0-P2.7) มีโครงสร้างคล้ายกับพอร์ต 0 โดยมีเฟ็ตตัวล่างตัวเดียว ส่วนด้านบนใช้ความต้านทานพูลอัพแทน พอร์ตนี้ทำงาน 2 หน้าที่ คือสามารถใช้เป็นพอร์ตสำหรับส่งแอดเดรสไบต์สูง (A8-A15) และเป็นอินพุตพอร์ตหรือเอาท์พุตพอร์ตสำหรับใช้งานทั่วไป เมื่อจะใช้งานเป็นอินพุตพอร์ต ต้องส่งลอจิก “1” มาที่พอร์ตนี้ก่อนเพื่อบังคับให้เฟ็ตอยู่ในสภาวะปิด ดังแสดงในรูปที่ 2.7

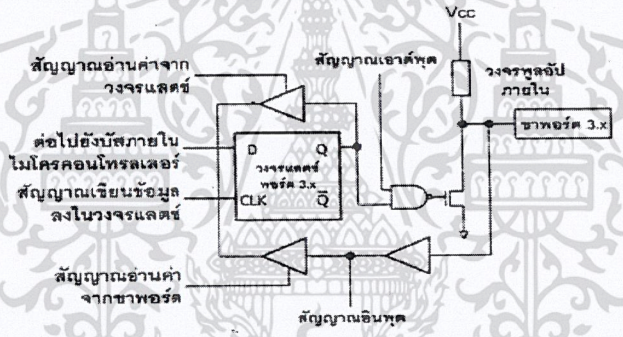


รูปที่ 2.7 วงจรภายในของพอร์ต 2 ในไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลช

- ขา PORT 3 (10-17) มีทั้งหมด 8 บิต (P3.0-P3.7) มีโครงสร้างคล้ายกับพอร์ต 1 แสดงดังรูปที่ 2.8 พอร์ตนี้ทำหน้าที่เป็นอินพุตพอร์ตหรือเอาท์พุตพอร์ตสำหรับใช้งานทั่วไป ถ้าจะใช้งานเป็นอินพุตพอร์ต ต้องส่งลอจิก “1” มาที่พอร์ตนี้ก่อนและนอกจากนั้นยังเป็นพอร์ตที่ทำหน้าที่การใช้งานพิเศษ คือ ส่งสัญญาณควบคุมออกมาและรับสัญญาณเข้าไป ดังมีรายละเอียดขั้นต้นดังนี้
  - P3.0 ใช้เป็นขาอินพุตสำหรับรับข้อมูลจากการสื่อสารแบบอนุกรม หรือขา RXD (Serial Input Port)
  - P3.1 ใช้เป็นขาเอาท์พุตสำหรับส่งข้อมูลจากการสื่อสารแบบอนุกรม หรือขา TXD (Serial Output Port)
  - P3.2 ใช้เป็นขาอินพุตสำหรับรับสัญญาณอินเตอร์รัปต์จากภายนอกช่องที่ 0 หรือขา $\overline{\text{INT}}0$ (External Interrupt)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- P3.3 ใช้เป็นขาอินพุตสำหรับรับสัญญาณอินเตอร์รัปต์จากภายนอกช่องที่ 1 หรือขา $\overline{INT1}$ (External Interrupt)
- P3.4 ใช้เป็นขาสำหรับรับสัญญาณไทเมอร์ ที่ทำหน้าที่นับจำนวนไซเคิลของสัญญาณ T1 หรือสัญญาณนาฬิกาก็ได้ จากภายนอกช่องที่ 0 หรือขา T0 (Timer/Counter 0 External Input)
- P3.5 ใช้เป็นขาสำหรับรับสัญญาไทเมอร์จากภายนอกช่องที่ 1หรือขา T1(Timer/Counter 1 External Input) ซึ่งมีหน้าที่การทำงานเหมือนกับ T0
- P3.6 ใช้เป็นขาสัญญาณควบคุมการเขียนข้อมูลไปยังหน่วยความจำภายนอกในกรณีที่มีการเชื่อมต่อกับหน่วยความจำภายนอก หรือขา $\overline{WR}$ (External Data Memory Write Strobe)
- P3.7 ใช้เป็นขาสัญญาณควบคุมการอ่านข้อมูลไปยังหน่วยความจำภายนอกในกรณีที่มีการเชื่อมต่อกับหน่วยความจำภายนอก หรือขา $\overline{RD}$ (External Data Memory Read Strobe)



รูปที่ 2.8 วงจรภายในของพอร์ต 3 ในไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลช

- ขารีเซต (RST, 9) ใช้ในการรีเซตการทำงานของไมโครคอนโทรลเลอร์ โดยการป้อนสัญญาณเพื่อรีเซต (ป้อนลอจิก “1”) เข้าที่ขานี้โดยคงสถานะของสัญญาณการรีเซตไว้อย่างน้อย 2 แมชชีนไซเคิล โดยที่วงจรกำเนิดสัญญาณนาฬิกายังคงทำงานต่อเนื่องไปอย่างปกติ ภายหลังจากเกิดรีเซตขึ้นแล้ว ค่าต่างๆในรีจิสเตอร์ใดๆ จะเป็นไปตามตารางที่ 2.2 ในหัวข้อที่ 2.4
- ขา ALE/$\overline{PROG}$ (Address Latch Enable/Program Pulse Input, 30) เป็นขาส่งสโตรบสำหรับใช้ในการควบคุมการแลตช์แอดเดรสไบต์ต่ำ (A0-A7) ที่ส่งออกมาจากพอร์ต 0 เมื่อมีการใช้งานหน่วยความจำภายนอก สัญญาณนี้จะแอกทีฟทุกๆ 2 ครั้ง ใน 1 แมชชีนไซเคิล (1/16 ของสัญญาณนาฬิกา) นอกจากนั้นขานี้ยังใช้เป็นขาสำหรับรับพัลส์ของการโปรแกรมสำหรับโปรแกรมข้อมูลลงในไมโครคอนโทรลเลอร์ ในรุ่นที่หน่วยความจำเป็นแบบอีอีพรอม
- ขา $\overline{PSEN}$ (Program Strobe Enable, 29) เป็นขาที่ใช้ในการส่งสัญญาณเพื่อร้องขอการติดต่อกับหน่วยความจำภายนอก เมื่อไมโครคอนโทรลเลอร์ต้องการอ่านข้อมูลจากหน่วยความจำโปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภายนอก ตัวไมโครคอนโทรเลเลอร์จะส่งสัญญาณนี้ออกมา 2 ครั้ง ในแต่ละแมชชีนไซเคิล แต่ถ้าเป็นการอ่านค่าจากหน่วยความจำโปรแกรมภายใน จะไม่มีการส่งสัญญาณใดๆออกที่ขานี้

- ขา $\overline{EA}$/Vpp (External Acess Enable / Programing Voltage Input, 31) ใช้สำหรับการเลือกติดต่อหน่วยความจำโปรแกรมภายนอก หรือภายในตัวไมโครคอนโทรลเลอร์ ถ้าป้อนลอจิก “0” เข้าที่ขานี้ จะเป็นการเลือกให้ไมโครคอนโทรลเลอร์ติดต่อกับหน่วยความจำโปรแกรมภายนอก แต่หากป้อนลอจิก “1” เข้าที่ขานี้ จะเป็นการเลือกให้ไมโครคอนโทรลเลอร์ติดต่อกับหน่วยความจำโปรแกรมภายในตัวไมโครคอนโทรลเลอร์ นอกจากนี้ที่ขานี้ยังใช้เป็นขาอินพุทสำหรับแรงดันไฟสูงสำหรับการโปรแกรมหน่วยความจำภายในไมโครคอนโทรลเลอร์ สำหรับในไมโครคอนโทรลเลอร์แบบแฟลชต้องการแรงดันสำหรับการโปรแกรม +12 V
- ขา XTAL1 ใช้ต่อกับคริสตอลภายนอกโดยเป็นอินพุทเข้าสู่วงจรสร้างสัญญาณนาฬิกา (Oscillator)
- ขา XTAL2 ใช้ต่อกับคริสตอลภายนอกโดยเป็นอินพุทเข้าสู่วงจรสร้างสัญญาณนาฬิกา (Oscillator)

2.2 การนำพอร์ตมาใช้งาน

เนื่องจากพอร์ตทั้งหมดของไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลชสามารถเป็นได้ทั้งอินพุตและเอาท์พุต ดังนั้นจะขอกล่าวถึงการกำหนดลักษณะการทำงานให้แก่พอร์ตของไมโครคอนโทรลเลอร์แบบแฟลช

การใช้งานเป็นพอร์ตอินพุต

ในการกำหนดให้เป็นพอร์ตอินพุตต้องเริ่มต้นด้วยการเขียนข้อมูล “1” มาที่แต่ละบิตของแต่ละพอร์ตที่ต้องการใช้งานเป็นอินพุต เพื่อหยุดการทำงานของเฟ็ตที่ใช้ในการขับสัญญาณเอาท์พุตของบิตนั้น ๆ ทำให้ขาสัญญาณของพอร์ตเชื่อมต่อเข้ากับวงจรพูลอัพโดยตรง ส่งผลให้ขาพอร์ตนั้นมีลอจิกเป็น “1” สามารถรับสัญญาณลอจิก “0” จากอุปกรณ์ภายนอกได้ง่าย สัญญาณข้อมูลจากอุปกรณ์ภายนอกจะถูกส่งเข้ามาแล้วเก็บไว้ในวงจรบัฟเฟอร์ภายในพอร์ต แล้วรอให้หน่วยประมวลผลกลางมาอ่านค่าไป เมื่อเป็นเช่นนี้อุปกรณ์ภายนอกที่เชื่อมต่อกับพอร์ตอินพุตของไมโคร -คอนโทรลเลอร์แบบแฟลชควรกำหนดให้ทำงานในสภาวะลอจิก “0” จะดีและสะดวกสุด (ซึ่งในปัจจุบันอุปกรณ์อินพุตที่เชื่อมต่อกับไมโครคอนโทรลเลอร์แทบทั้งหมดทำงานที่ลอจิก “0” แล้ว)

และเนื่องจากที่ขาพอร์ต 0 ไม่มีวงจรพูลอัพภายใน หากมีการนำพอร์ต 0 ไปใช้งานเป็นพอร์ตอินพุตจะต้องต่อความต้านทานพูลอัพภายนอกเข้าที่ขาพอร์ต 0 ทุกขาด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น และห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การใช้งานเป็นพอร์ตเอาท์พุต

โดยปกติแล้วขาพอร์ตจะกำหนดให้มีลักษณะเป็นเอาท์พุตอยู่แล้ว กล่าวคือ เมื่อต้องการส่งข้อมูล “0” ออกไปทางเอาท์พุตก็ให้เขียนข้อมูล “0” ไปยังวงจรแลตช์ซึ่งก็จะส่งต่อไปขับเฟ็ตทำให้เฟ็ตทำงาน ที่ขาพอร์ตที่กำหนดให้ทำงานก็จะเกิดลอจิก “0” ขึ้น ในทางตรงกันข้ามหากต้องการส่งข้อมูล “1” ออกไป ก็ให้เขียนข้อมูล “1” ออกไปยังวงจรแลตช์ วงจรขับก็จะหยุดทำงาน ทำให้ที่ขาพอร์ตเชื่อมต่อกับวงจรพูลอัพภายในเกิดเป็นลอจิก “1” ที่ขาพอร์ตนั้น ซึ่งคล้ายกับการกำหนดให้เป็นขาอินพุตมาก เพียงแต่แตกต่างกันที่กระบวนการในการเคลื่อนย้ายข้อมูล โดยถ้าเป็นลอจิก “1” ออกไปยังวงจรแลตช์ วงจรขับก็จะหยุดทำงาน ทำให้ที่ขาพอร์ตเชื่อมต่อกับวงจรพูลอัพภายในเกิดเป็นลอจิก “1” ที่ขาพอร์ตนั้น ซึ่งคล้ายกับการกำหนดให้เป็นขาอินพุตมากเพียงแต่แตกต่างกันที่กระบวนการในการเคลื่อนย้ายข้อมูล โดยถ้าเป็นอินพุตจะมีสัญญาณมาอ่านข้อมูลที่บัฟเฟอร์ แต่ถ้าเป็นเอาท์พุตจะไม่มีการอ่านข้อมูลที่บัฟเฟอร์แต่อย่างใด เว้นแต่กรณีที่ต้องการตรวจสอบข้อมูลที่ส่งออกมาทางเอาท์พุต

เมื่อใช้งานพอร์ตของไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลชเป็นพอร์ตเอาท์พุต แต่ละขา (หรือในแต่ละบิต) ของแต่ละพอร์ตมีความสามารถในการจ่ายกระแสหรือที่เรียกว่า กระแสซอร์ส (Source Current) ได้สูงสุด 10 มิลลิแอมแปร์ และทุกขารวมกันในแต่ละพอร์ต (ทั้ง 8 บิต) สูงสุด 26 มิลลิแอมแปร์ สำหรับพอร์ต 0 และ 15 มิลลิแอมแปร์ สำหรับพอร์ต 1ถึง 3 ในกรณีที่ใช้งานทุกพอร์ตเอาท์พุต จะสามารถจ่ายกระแสได้รวมกันสูงสุด 71 มิลลิแอมแปร์ ดังนั้นในการใช้งานเป็นพอร์ตเอาท์พุตเพื่อไม่ให้เกิดปัญหาเกี่ยวกับความสามารถในการจ่ายกระแสจึงควรต่อวงจรบัฟเฟอร์ทางเอาท์พุตเพื่อช่วยในการขับกระแสอีกทางหนึ่ง

การอ่านค่าลอจิกจากพอร์ต

ไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลช สามารถอ่านค่าลอจิกจากพอร์ตได้ 2 ลักษณะคือ อ่านค่าจากพอร์ตโดยตรง และอ่านค่าจากวงจรแลตช์ของแต่ละพอร์ต

ในกรณีที่พอร์ตต่อกับขาเบสทรานซิสเตอร์ชนิด NPN และขาอีมิตเตอร์ของทรานซิสเตอร์ตัวนั้นต่อลงกราวด์ หากมีการส่งข้อมูล “1” ไปยังทรานซิสเตอร์จะทำให้ทรานซิสเตอร์ทำงานสถานะลอจิกที่ขาพอร์ตลอจิกจะเป็น “0” เนื่องจากเมื่อทรานซิสเตอร์ทำงาน จะเสมือนว่าขาพอร์ตถูกต่อลงกราวด์ทำให้หากอ่านค่าลอจิกที่ขาพอร์ตจะได้ผลตรงกันข้ามกับที่ส่งออกมา แต่หากขณะทำงานอ่านค่าลอจิกที่วงจรแลตช์จะได้ค่าตรงกับค่าที่ต้องการส่งจริง ดังนั้นมีการอ่านค่าลอจิกของพอร์ตจึงต้องเลือกวิธีการให้เหมาะสมกับอุปกรณ์ที่นำมาต่อด้วย

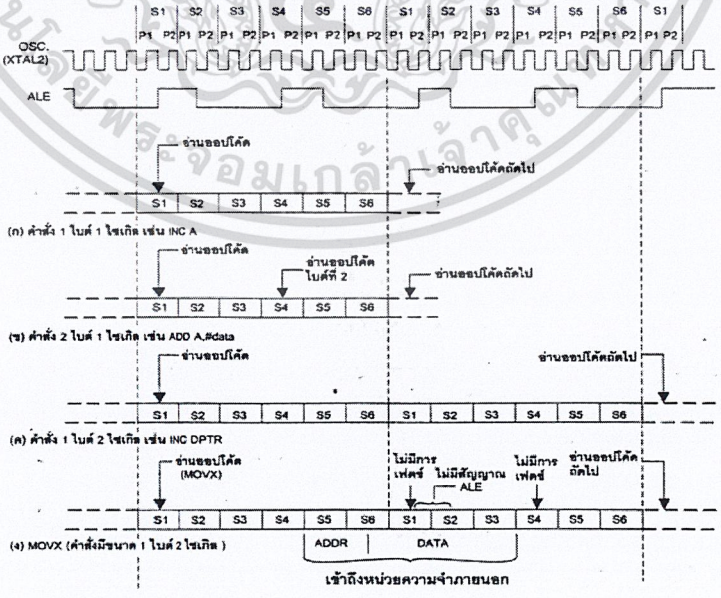
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3 จังหวะการทำงานของไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลช

ในการประมวลคำสั่งของหน่วยประมวลผลกลางจะมีขั้นตอนหลัก ๆ 2 ขั้นตอน คือ กระบวนการเฟ็ต (Fetch) เป็นการเรียกคำสั่งออกจากหน่วยความจำโปรแกรมแล้วทำการแปลงรหัสคำสั่งนั้นเป็นภาษาเครื่องเพื่อเตรียมประมวลผล ขั้นตอนต่อมาคือ กระบวนการเอ็กซีคิวต์ (Execute) เป็นการกระทำการตามคำสั่งที่กำหนดหรือตามที่เฟ็ตขึ้นมา โดยกระบวนการก่อนหน้านี้ เมื่อทำการเอ็กซีคิวต์คำสั่งเรียบร้อยแล้วก็จะเริ่มกระบวนการเฟ็ตคำสั่งใหม่ต่อไป

เมื่อเริ่มจ่ายไฟให้แก่ไมโครคอนโทรลเลอร์ จะเกิดการรีเซตในลักษณะที่เรียกว่า เพาเวอร์ออนรีเซต (Power-on Reset) ทำให้หน่วยประมวลผลกลางไปเริ่มต้นการทำงานที่แอดเดรส 0000H ของหน่วยความจำโปรแกรม จังหวะการทำงานของหน่วยประมวลผลกลางจะเป็นไปตามรูปแบบ โดยได้รับการกำหนดมาจากรอบการทำงาน (Machine Cycle) โดยไดอะแกรมเวลาในรูปที่ 2.9 แสดงจังหวะการทำงานของไมโครคอนโทรลเลอร์ โดยใน 1 รอบการทำงานจะแบ่งออกเป็น 6 สเตต (State) กำหนดชื่อเป็น S1-S6 ในแต่ละสเตตมีค่าเวลาเท่ากับ 2 คาบเวลาของสัญญาณนาฬิกา ถ้าสัญญาณนาฬิกามีความถี่ 12 MHz จะมีคาบเวลาเท่ากับ 1 มิลลิวินาที คาบเวลาทั้งสองภายใน 1 สเตตเรียกว่า เฟส1 (Phase1) และเฟส2(Phase2)

ความเร็วในการทำงานภายในของไมโครคอนโทรลเลอร์เท่ากับ
ความถี่ของสัญญาณนาฬิกา (ค่าของคริสตอลที่ต่ออยู่ที่ขา XTAL1 และ XTAL2) / 12
เวลา 1 รอบการทำงาน = 1 / ความเร็วในการทำงานภายในของไมโครคอนโทรลเลอร์



รูปที่ 2.9ก ไซเคิลการทำงานของไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลช

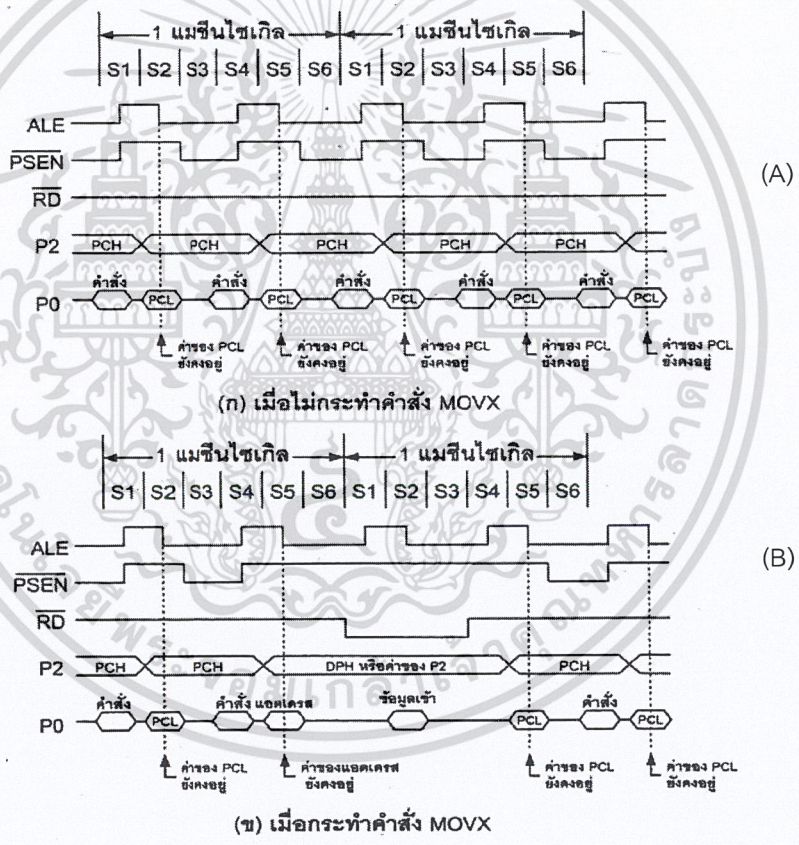
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 2.9ก (ก) แสดงการทำงานของคำสั่ง INC A ซึ่งเป็นคำสั่ง 1 ไบต์ ทำงานเสร็จใน 1รอบการทำงาน

รูปที่ 2.9ก (ข) แสดงการทำงานของคำสั่ง ADD A,#DATA ซึ่งเป็นคำสั่ง 2 ไบต์ แต่ทำงานเสร็จใน 1 รอบการทำงาน

รูปที่ 2.9ก (ค) แสดงการทำงานของคำสั่ง INC DTPR ซึ่งเป็นคำสั่ง 1 ไบต์ แต่ทำงานเสร็จใน 2 รอบการทำงาน

รูปที่ 2.9ก (ง) แสดงการทำงานของคำสั่ง MOVX ซึ่งเป็นคำสั่ง 1 ไบต์ แต่ทำงานเสร็จใน 2 รอบการทำงาน



รูปที่ 2.9ข ไดอะแกรมเวลาแสดงการติดต่อหน่วยความจำโปรแกรมภายนอกของ MCS-51

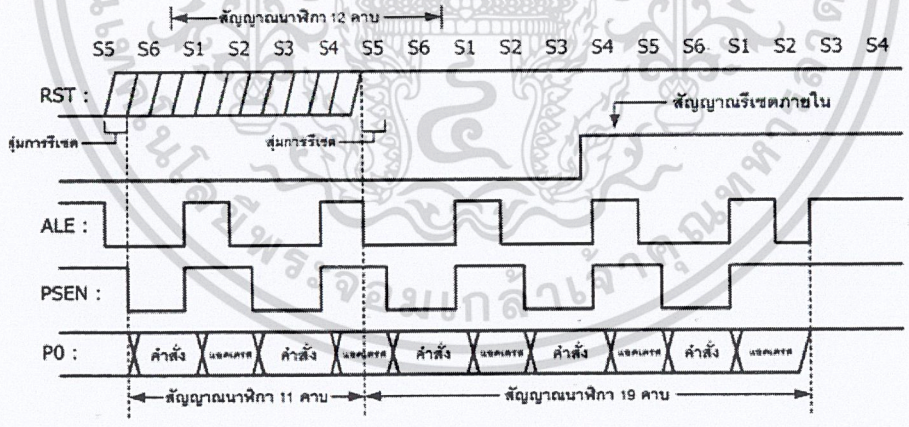
ในรูปที่ 2.9ข แสดงสัญญาณและไดอะแกรมเวลาของการเข้าถึงหน่วยความจำโปรแกรมภายนอก โดยในรูปที่ 2.9ข (A) เป็นไดอะแกรมเวลาในขณะที่ยังไม่มีการกระทำคำสั่ง MOVX สัญญาณที่ขา ALE และ $\overline{\text{PSEN}}$ จะเกิดการแอกทีฟ 2 ครั้งภายใน 1 รอบการทำงาน ในทุกครั้งที่ ALE เกิดการแอกทีฟที่พอร์ต 0 จะมีค่าของรีจิสเตอร์ PC ในไบต์ต่ำออกมา ในขณะที่พอร์ต 2 ก็จะมีค่าของ PC ในไบต์สูงเพื่อชี้ไปยังแอดเดรสต่อไปที่ต้องการไปดำเนินการ สำหรับขา $\overline{\text{PSEN}}$ ก็จะเกิดการแอกทีฟเมื่อมี

การติดต่อกับหน่วยความจำโปรแกรมภายนอก ในกรณีที่กระทำคำสั่ง MOVX เพื่อเข้าถึงหน่วยความจาข้อมูลภายนอก ที่ขา $\overline{\text{PSEN}}$ จะไม่เกิดการแอกทีฟ 2 ครั้ง ภายใน 1 รอบการทำงาน เนื่องจากบัสแอดเดรส และบัสข้อมูลจะถูกใช้ในการติดต่อหน่วยความจำข้อมูลภายนอกแทน คือจะเป็นช่วงเวลาในการอ่านข้อมูลจากหน่วยความจำข้อมูล โดยใช้สัญญาณ RD แต่สำหรับสัญญาณ ALE ยังคงแอกทีฟตามจังหวะการทำงานเหมือนเดิม แสองในรูปที่ 2.9ข (B)

อาจสรุปง่าย ๆ ว่า การอ่านข้อมูลจากหน่วยความจำโปรแกรมจะใช้สัญญาณ $\overline{\text{PSEN}}$ และการอ่านข้อมูลจากหน่วยความจำข้อมูล จะใช้สัญญาณ $\overline{\text{RD}}$ ส่วนสัญญาณ ALE คือสัญญาณที่ใช้ในการแลตช์แอดเดรส A0-A7 นั่นเอง

2.4 กระบวนการรีเซ็ตในไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลช

การรีเซตในไมโครคอนโทรลเลอร์เกิดขึ้นได้เพียงกรณีเดียวคือที่ขารีเซต (ขา 9) ได้รับลอจิก “1” อย่างน้อยเป็นเวลา 2 รอบการทำงาน หรือ 24 คาบเวลาของสัญญาณนาฬิกา โดยที่วงจรกำเนิดสัญญาณยังคงทำงานอยู่เมื่อเกิดการรีเซตขึ้นไมโครคอนโทรลเลอร์จะสร้างสัญญาณรีเซตภายในขึ้นเพื่อตอบสนองการรีเซตจากภายนอก ดังแสดงไดอะแกรมเวลาของกระบวนการรีเซตภายในไมโครคอนโทรลเลอร์ในรูปที่ 2.10



รูปที่ 2.10 ไดอะแกรมเวลาของกระบวนการรีเซตภายในไมโครคอนโทรลเลอร์ MCS-51

สัญญาณรีเซตที่เกิดขึ้นจากภายนอกจะไม่มีความสอดคล้องกับสัญญาณนาฬิกาแต่อย่างใด เนื่องจากการรีเซตอาจเกิดขึ้นเมื่อใดก็ได้ โดยจะมีการสุ่มตรวจสอบสถานะลอจิกที่ขารีเซต ใน สเตต 5 เฟส 2 ของทุกรอบการทำงาน เมื่อสามารถตรวจจับลอจิก ”1” ที่ขารีเซตได้และเมื่อเกิดสัญญาณรีเซตภายในเกิดขึ้น กระบวนการรีเซตของไมโครคอนโทรลเลอร์จะเริ่มต้นขึ้นอย่างเป็นทางการ ขาพอร์ตทั้งหมดยังคงรักษาข้อมูลปัจจุบันไว้ได้อีก 19 คาบเวลาของสัญญาณนาฬิกาหลังจากที่ขารีเซตได้รับลอจิก “1” จากนั้นจะเปลี่ยนแปลงค่ากลายเป็น FFH ทั้งหมด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สัญญาณ ALE และ $\overline{\text{PSEN}}$ จะกลายเป็นลอจิก “1” เมื่อการรีเซตเกิดขึ้นอย่างสมบูรณ์ และเมื่อขารีเซตถูกทำให้กลายเป็นลอจิก “0” อันเป็นการแจ้งให้ทราบว่าสามารถเริ่มต้นกระบวนการทำงานปกติได้แล้ว สัญญาณ ALE และ $\overline{\text{PSEN}}$ จะเกิดขึ้นใหม่อีกครั้งหลังจากที่ขารีเซตเป็นลอจิก “0” 1-2 รอบ การทำงานทั้งนี้เป็นการรอให้อุปกรณ์ที่ต่อร่วมอยู่กับไมโครคอนโทรลเลอร์อยู่ในสภาวะพร้อมทำงานและสามารถทำงานในจังหวะที่สอดคล้องกับไมโครคอนโทรลเลอร์ได้อย่างเหมาะสม

ผลของการรีเซตที่มีต่อรีจิสเตอร์ภายในไมโครคอนโทรลเลอร์นั้นแสดงในตารางที่ 2.2

0F8H								0FFH
0F0H	B 00000000							0F7H
0E8H								0EFH
0E0H	ACC 00000000							0E7H
0D8H								0DFH
0D0H	PSW 00000000							0D7H
0C8H	T2CON 00000000	T2MOD XXXXXX00	RCAP2L 00000000	RCAP2H 00000000	TL2 00000000	TH2 00000000		0CFH
0C0H								0C7H
0B8H	IP XX000000							0BFH
0B0H	P3 11111111							0B7H
0A8H	IE 0X000000							0AFH
0A0H	P2 11111111							0A7H
98H	SCON 00000000	SBUF XXXXXXXX						9FH
90H	P1 11111111							97H
88H	TCON 00000000	TMOD 00000000	TL0 00000000	TL1 00000000	TH0 00000000	TH1 00000000		8FH
80H	P0 11111111	SP 00000111	DPL 00000000	DPH 00000000			PCON 0XXX0000	87H

ตารางที่ 2.2 แสดงผลของการรีเซตที่มีต่อรีจิสเตอร์ภายในไมโครคอนโทรลเลอร์ MCS-51

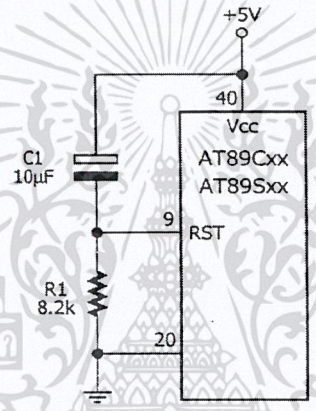
กลไกที่สำคัญอีกประการหนึ่งของกระบวนการรีเซตคือ ค่าของหน่วยความจำภายในจำไม่ได้รับผลกระทบจากการรีเซตแต่อย่างใด ดังนั้นในการเขียนโปรแกรมควบคุมที่ส่วนต้นของโปรแกรมจึงควรมีการเคลียร์ค่าของหน่วยความจำข้อมูลภายในที่ต้องใช้งานเสมอ เพื่อป้องกันการใช้ข้อมูลที่เกิดขึ้นก่อนการรีเซตมาประมวลผล เนื่องจากโอกาสในการรับรู้ว่าข้อมูลของหน่วยความจำก่อนการรีเซตมีค่าเท่าใดมีน้อยมาก

2.4.1 เพาเวอร์ออนรีเซต (Power-On Reset)

เป็นการรีเซตที่เกิดขึ้นหลังจากมีการจ่ายไฟเลี้ยงให้แก่ไมโครคอนโทรลเลอร์ โดยมีวงจรพื้นฐานเพื่อทำให้เกิดการรีเซตแบบนี้ในรูปที่ 2.11

เอกสารนี้เป็นการ... ไม่อนุญาตให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

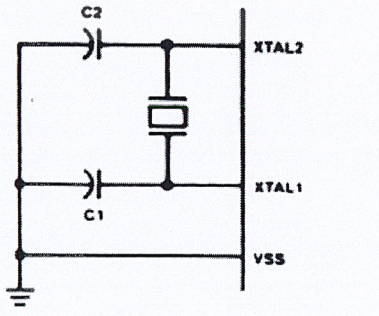
การทำงานของการรีเซตแบบนี้ กล่าวคือ เมื่อเริ่มต้นจ่ายไฟ ตัวเก็บประจุไฟฟ้าทำให้มีกระแสไหลผ่านตัวต้านทาน เกิดแรงดันตกคร่อมตัวต้านทาน ส่งผลให้ขา RST ได้รับแรงดันไฟตรงประมาณ +5 V ซึ่งก็คือระดับแรงดันของลอจิก “1” โดยตรง จนกระทั่งเวลาผ่านไป ตัวเก็บประจุสามารถประจุแรงดันแรงดันไฟฟ้ามากขึ้นเรื่อย ๆ ทำให้แรงดันตกคร่อมตัวต้านทานลดลงตามลำดับ และกลายเป็นศูนย์เมื่อตัวเก็บประจุสามารถประจุแรงดันได้อย่างเต็มที่ ดังนั้นการกำหนดค่าของตัวเก็บประจุและตัวต้านทานในวงจรนี้มีความสำคัญต่อการรีเซตของไมโครคอนโทรลเลอร์อย่างมาก เพราะต้องกำหนดให้ค่าเวลาของการประจุแรงดันของตัวเก็บประจุยาวนานมากพอที่จะทำให้ขารีเซต ได้รับลอจิก “1” อย่างน้อยเป็นเวลา 2 รอบการทำงาน



รูปที่ 2.11 วงจรพาวเวอร์ออนรีเซต

2.5 วงจรนาฬิกาของไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลช

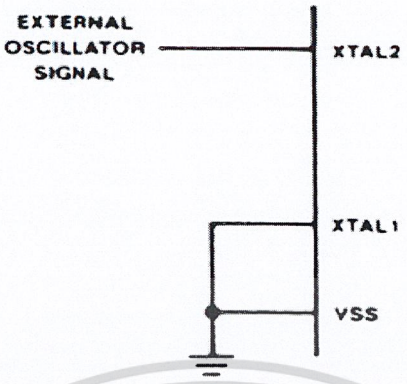
การต่อมีอยู่ด้วยกัน 2 รูปแบบ คือ แบบใช้วงจรนาฬิกาภายในชิพ (On-Chip Oscillator) และแบบต่อกับวงจรสร้างสัญญาณนาฬิกาจากภายนอก (External Clock Source) รูปที่ 2.12 และรูปที่ 2.13 ได้แสดงรูปแบบการต่อทั้งสองแบบตามลำดับ



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 2.12 แสดงการต่อวงจรนาฬิกาแบบภายในชิพของไมโครคอนโทรลเลอร์ MCS-51

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง



รูปที่ 2.13 แสดงการต่อวงจรนาฬิกาจากภายนอกของไมโครคอนโทรลเลอร์ MCS-51

2.6 การจัดหน่วยความจำของไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลช

ในไมโครคอนโทรลเลอร์แบบแฟลชมีหน่วยความจำภายในหลัก ๆ อยู่ 2 ส่วนคือ หน่วยความจำโปรแกรมและหน่วยความจำข้อมูล ซึ่งก็มีขนาดและการจัดสรรแตกต่างกันไปในแต่ละเบอร์ ซึ่งในหัวข้อนี้จะกล่าวถึงรายละเอียดของการจัดสรรหน่วยความจำภายในไมโครคอนโทรล -เลอร์แบบแฟลช การเชื่อมต่อกับหน่วยความจำภายนอก และข้อมูลเบื้องต้นของรีจิสเตอร์ฟังก์ชันพิเศษที่ใช้ควบคุมการทำงานของไมโครคอนโทรลเลอร์แบบแฟลชดังนี้

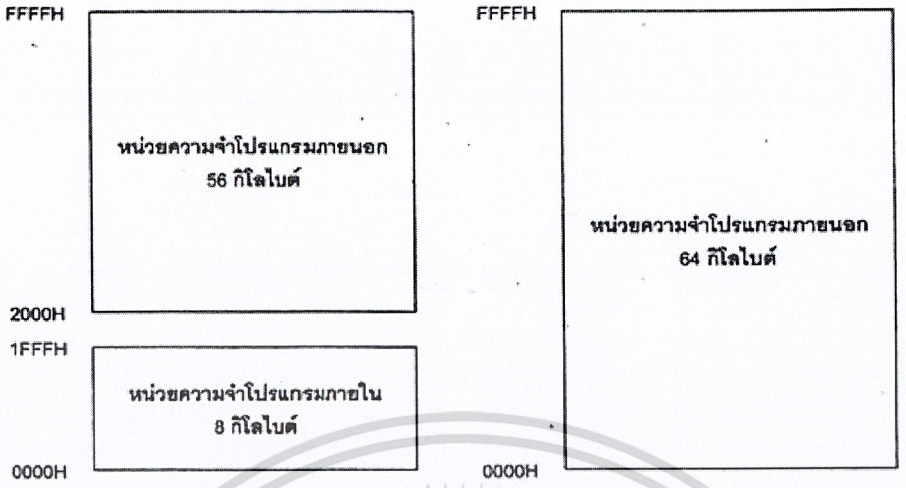
2.6.1 หน่วยความจำโปรแกรม (Program Memory)

ในระบบของไมโครคอนโทรลเลอร์ จำเป็นต้องมีหน่วยความจำสำหรับบรรจุคำสั่ง หรือโปรแกรมที่ผู้ใช้พัฒนาขึ้นจัดเก็บไว้ภายในหน่วยความจำ ที่เรียกว่าหน่วยความจำโปรแกรม โดยอาจจะประกอบอยู่ภายในตัวไอซีของไมโครคอนโทรลเลอร์เอง หรือเป็นไอซีของหน่วยความจำแยกออกต่างหากได้ ในกรณีหลังจำเป็นจะต้องมีการใช้พอร์ตอินพุต/เอาท์พุต ทำหน้าที่เป็นบัสแอดเด-รสและข้อมูลเพื่อให้สามารถทำการเชื่อมต่อเข้ากับไอซีหน่วยความจำมาตรฐานทั่วไปได้

หน่วยความจำโปรแกรมใช้เก็บข้อมูลและคำสั่งใช้งานต่าง ๆ ซึ่งแม้ว่าจะไม่มีกระแสไฟฟ้าให้กับระบบ ข้อมูลเหล่านี้ก็ยังคงอยู่ ไม่สูญหาย โครงสร้างของหน่วยความจำโปรแกรม มีลักษณะเช่นเดียวกับหน่วยความจำที่บรรจุอยู่ในไอซีหน่วยความจำประเภทต่าง ๆ เช่นหน่วยความจำแบบรอม (ROM) หรืออีพีพรอม (EPROM) เป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

50207



รูปที่ 2.14 แสดงการจัดพื้นที่หน่วยความจำโปรแกรมสำหรับไมโครคอนโทรลเลอร์เบอร์ AT89S53

การจัดหน่วยความจำโปรแกรมของไมโครคอนโทรลเลอร์แบบแฟลช สามารถติดต่อหน่วยความจำโปรแกรมได้สูงสุด 64 กิโลไบต์ ซึ่งในไมโครคอนโทรลเลอร์เบอร์ AT89S53 นั้นมีหน่วยความจำโปรแกรมภายในแบบรอมขนาด 12 กิโลไบต์ ฉะนั้นถ้าต้องการใช้หน่วยความจำโปรแกรมภายนอก ก็จะสามารถติดต่อได้อีก 52 กิโลไบต์

ไมโครคอนโทรลเลอร์เบอร์ต่าง ๆ สามารถขยายให้ใช้งานหน่วยความจำภายนอกทั้งสิ้นโดยในกรณีที่มีหน่วยความจำโปรแกรมภายในอยู่แล้ว การอ้างตำแหน่งแอดดรสที่มีทั้งหน่วยความจำโปรแกรมภายในและภายนอกนั้น จะต้องทำการพิจารณาระดับลอจิกของสัญญาณ $\overline{EA}$ ในขณะนั้นด้วย

การเชื่อมต่อหน่วยความจำโปรแกรมภายนอกกับไมโครคอนโทรลเลอร์

เนื่องจากระบบบัสแอดเดรส และข้อมูลของไมโครคอนโทรลเลอร์ เป็นลักษณะแบบใช้การมัลติเพล็กซ์ จากพอร์ตเดียวกัน กล่าวคือ ในระยะเวลาเริ่มต้น เส้นสัญญาณเหล่านี้ของพอร์ตจะใช้ในการส่งค่าแอดเดรส ของตำแหน่งที่ต้องการติดต่อด้วย ในช่วงเวลาต่อมาจึงจะเปลี่ยนไปเป็นสภาวะอิมพีแดนซ์สูงเพื่อใช้งานในฐานะของบัสข้อมูล แต่เนื่องจากว่า EPROM ที่ใช้งานทั่วไปนั้นๆ ไม่ใช้การมัลติเพล็กซ์ และมีขาสัญญาณบัสแอดเดรสและบัสข้อมูลแยกออกจากกันโดยชัดเจน ดังนั้นการเชื่อมต่อ EPROM เพื่อทำหน้าที่เป็นหน่วยความจำโปรแกรม จึงจำเป็นต้องมีวงจรประเภทแลตช์ (Latch) ประกอบขึ้นมาเพื่อทำการค้างค่าของแอดเดรสที่ส่งออกมาจากไมโครคอนโทรล -เลอร์ ในช่วงเวลาแรกให้กับขาสัญญาณแอดเดรสของอีพีรอมต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

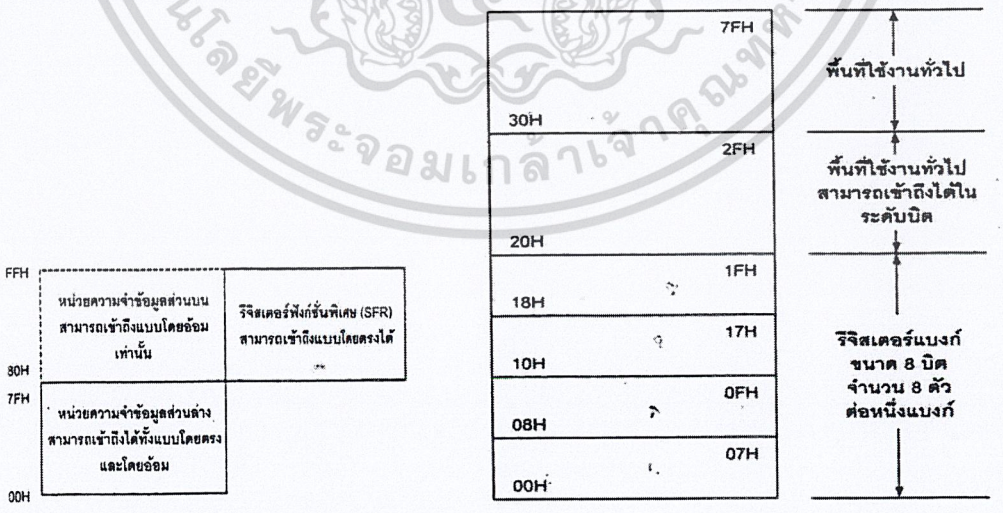
2.6.2 **หน่วยความจำข้อมูล** (Data Memory)

มีหน้าที่สำหรับเก็บข้อมูล หรือตัวแปรที่เกิดขึ้นในขณะที่กำลังประมวลผลโปรแกรมไว้เป็นการชั่วคราว โดยพื้นฐานแล้ว หน่วยความจำข้อมูลจัดว่าเป็นหน่วยความจำแรมแบบสแตติก (Static) ดั้งนั้นเมื่อไม่มีการจ่ายไฟฟ้าให้กับระบบ ก็จะมีผลให้ข้อมูลที่จัดเก็บไว้ภายในหน่วยความจำสูญหายไป พื้นที่ของหน่วยความจำข้อมูลของไมโครคอนโทรลเลอร์แบบแฟลช สามารถมีได้สูงสุดไม่เกิน 64 กิโลไบต์ และแยกเป็นสองลักษณะตามตำแหน่งที่ตั้งของหน่วยความจำนั้น คือหน่วยความจำข้อมูลภายใน ซึ่งเป็นแรมที่อยู่ภายในตัวของไมโครคอนโทรลเลอร์เอง และหน่วยความจำข้อมูลภายนอก (External Data Memory) ซึ่งเป็นการใช้ไอซีหน่วยความจำแรมมาเพิ่มเติมเข้าไปในวงจร ลักษณะเดียวกับการนำไอซีอีพีรอมมาใช้งานเป็นหน่วยความจำโปรแกรมนั่นเอง

หน่วยความจำข้อมูลภายใน (Internal Data Memory)

หน่วยความจำข้อมูลภายในของไมโครคอนโทรลเลอร์แบบแฟลชเบอร์ AT89S53นั้น มีจำนวนทั้งหมด 256 ไบต์ โดยแยกออกได้เป็น 2 ลักษณะ คือพื้นที่เฉพาะสำหรับประมวลผลกลางใช้งานเท่านั้น ซึ่งก็คือ รีจิสเตอร์และพื้นที่ใช้งานทั่วไปสำหรับโปรแกรมใช้งานที่ผู้ใช้สร้างขึ้นมา

สำหรับการจัดสรรหน่วยความจำข้อมูลภายในแบ่งป็น 3 ส่วนคือหน่วยความจำข้อมูลส่วนล่าง (Lower), ส่วนบน (Upper) และรีจิสเตอร์ฟังก์ชันพิเศษ (SFR : Special Function Register) ซึ่งแต่ละส่วนมีขนาด 128 ไบต์



(ก) (ข)

รูปที่ 2.15 แสดงการจัดสรรพื้นที่ของหน่วยความจำข้อมูลภายไมโครคอนโทรลเลอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บริเวณแอเดรส 00H - 1FH จำนวน 32 ไบต์ จำแนกออกเป็นกลุ่ม (หรือแบงค์ Bank) ข้อมูลจำนวน 8 ไบต์ รวมทั้งหมดสี่กลุ่ม พื้นที่ข้อมูลในแต่ละกลุ่มจะถูกใช้งานในฐานะรีจิสเตอร์ใช้งานทั่วไป ซึ่งมีชื่อเรียกว่า รีจิสเตอร์ R0 - R7 ดังตารางที่ 2.3

แอดเดรส	รีจิสเตอร์แบงค์	ชื่อรีจิสเตอร์ใช้งาน
00H – 07H	0	R0 – R7
08H – 0FH	1	R0 – R7
10H – 17H	2	R0 – R7
18H – 1FH	3	R0 – R7

ตารางที่ 2.3 ตารางแสดงกลุ่มแอดเดรสที่ถูกใช้งานในฐานะรีจีสเตอร์ใช้งานทั่วไป

ดังนั้นการใช้งานผู้ใช้จะต้องให้ความระมัดระวังว่าต้องการรีจิสเตอร์นั้น ๆ จากแบงค์ใด การสวิตช์เลือกแต่ละกลุ่มของรีจิสเตอร์นี้ก็ทำได้ง่าย เพียงการกำหนดค่าของบิตที่อยู่ภายในรีจิส-เตอร์ PSW เท่านั้น ตามตารางที่ 2.4

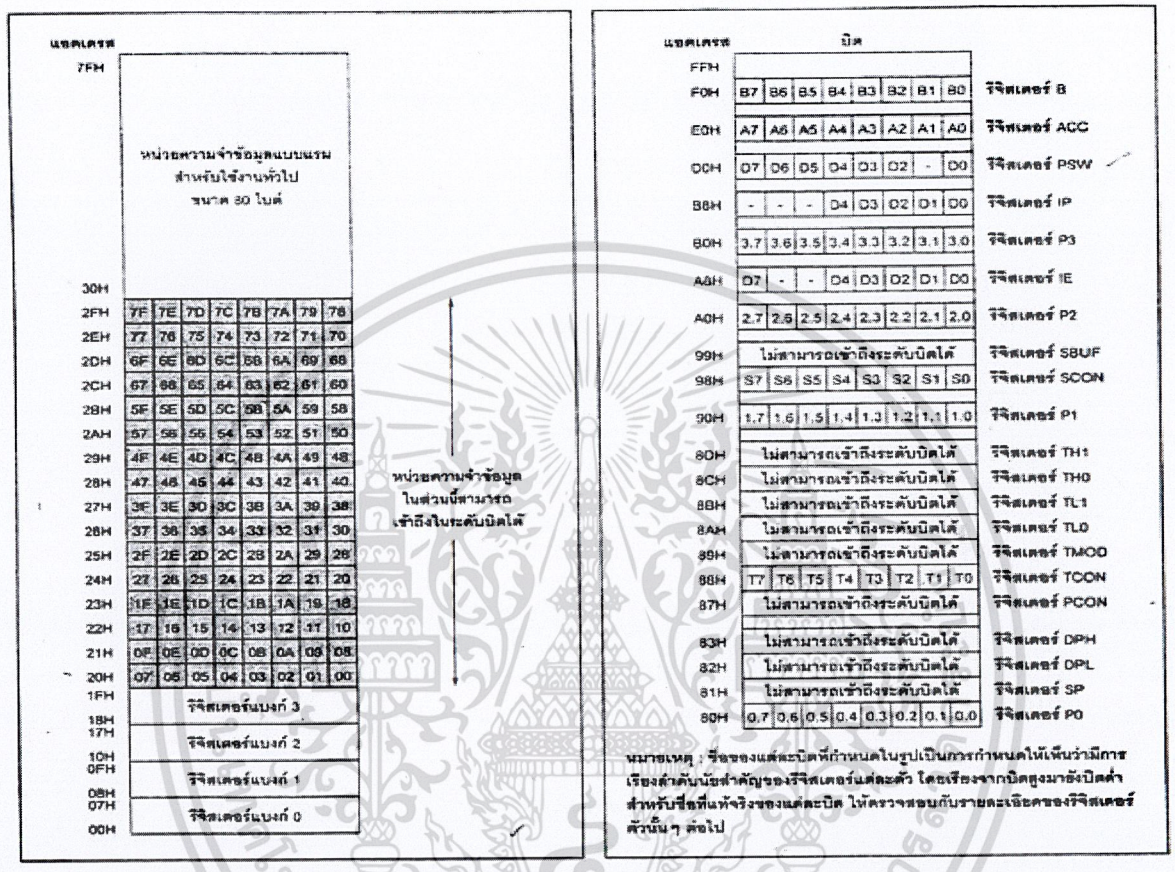
รีจิสเตอร์	บิต RS0	บิต RS1	ตำแหน่งหน่วยความจำ
แบงค์ 0	0	0	0000H
แบงค์ 1	0	1	0008H
แบงค์ 2	1	0	0010H
แบงค์ 3	1	1	0018H

ตารางที่ 2.4 ตารางแสดงการสวิตช์เลือกแต่ละกลุ่มของรีจีสเตอร์แบงค์

อย่างไรก็ตามโดยทั่วไปก็มักจะมีการใช้งานรีจิสเตอร์ R0 – R7 เฉพาะในแบงค์ 0 เท่านั้น ดังนั้นพื้นที่ของแบงค์อื่น ๆ ที่เหลือก็สามารถนำมาใช้ในลักษณะของหน่วยความจำข้อมูลภายในปกติด้วยการอ้างถึงหมายเลขของแอดเดรสนั้น ๆ โดยตรง

บริเวณแอดเดรส 20H – 2FH จำนวน 16 ไบต์ บริเวณพื้นที่เป็นส่วนสำหรับผู้ใช้ซึ่งจะมีความพิเศษต่างไปจากหน่วยความจำส่วนอื่น ๆ เนื่องจากผู้ใช้อาจสามารถอ้างถึงหน่วยความจำบริเวณนี้ได้ทั้งในลักษณะของไบต์ข้อมูลเช่นปกติ หรือจะเป็นบิตข้อมูลได้โดยตรง ดังนั้นหากมองในลักษณะบิตข้อมูลแล้ว ก็จะมีพื้นที่ตัวแปรแบบบิตให้ใช้งานได้มากถึง 128 บิต โดยตำแหน่งแรกของบิตจะเป็นบิตซึ่ง

เริ่มต้นนับจากบิตที่นัยสำคัญต่ำสุด (LSB) ของแอดเดรส 20H เรื่อยไปจนกระทั่งถึงบิตที่ 127 ซึ่งเป็นบิตที่นัยสำคัญสูงสุด (MSB) ของแอดเดรส 2FH



รูปที่ 2.16 แสดงหน่วยความจำข้อมูลภายในบริเวณที่อ้างถึงได้แบบบิต

หน่วยความจำขนาด 128 ไบต์ถัดมา

พื้นที่ตั้งแต่บริเวณแอดเดรส 80H – FFH เป็นบริเวณของหน่วยความจำที่มีการใช้งานเฉพาะจาก AT89C52 และ AT89Sxx เท่านั้น โดยจะนำมาใช้เป็นตำแหน่งของรีจิสเตอร์หน้าที่พิเศษจำนวน 28 ตำแหน่ง ดังแสดงในรูปที่ 2.17

สำหรับไมโครคอนโทรลเลอร์ AT89C52 และ AT89Sxx จะมีหน่วยความจำข้อมูลภายในสำหรับการใช้งานเพิ่มมากขึ้นกว่าเบอร์ AT89C51 อีก 128 ไบต์ โดยจะอยู่บริเวณช่วงแอดเดรส 80H ถึง FFH เช่นกัน ซึ่งแม้ว่าจะเป็นพื้นที่ที่มีหมายเลขแอดเดรสเดียวกับส่วนของรีจิสเตอร์หน้าที่พิเศษแต่ในความเป็นจริงแล้วจะเป็นพื้นที่หน่วยความจำอีกบริเวณหนึ่ง ซึ่งมีการซ้อนเกย (Overlap) กันให้อยู่ในบริเวณแอดเดรสช่วงนี้ ซึ่งหากว่าผู้ใช้งานต้องการจะเก็บข้อมูลในพื้นที่บริเวณนี้ ก็จะต้องใช้การอ้างถึงหน่วยความจำแบบโดยอ้อม (Indirect Addressing) เท่านั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Symbol	Name	Address
*ACC	Accumulator	0E0H
*B	B Register	0F0H
*PSW	Program Status Word	0D0H
SP	Stack Pointer	81H
DPTR	Data Pointer 2 Bytes	
DPL	Low Byte	82H
DPH	High Byte	83H
*P0	Port 0	80H
*P1	Port 1	90H
*P2	Port 2	0A0H
*P3	Port 3	0B0H
*IP	Interrupt Priority Control	0B8H
*IE	Interrupt Enable Control	0A8H
TMOD	Timer/Counter Mode Control	89H
*TCON	Timer/Counter Control	88H
*+T2CON	Timer/Counter 2 Control	0C8H
TH0	Timer/Counter 0 High Byte	8CH
TL0	Timer/Counter 0 Low Byte	8AH
TH1	Timer/Counter 1 High Byte	8DH
TL1	Timer/Counter 1 Low Byte	8BH
+TH2	Timer/Counter 2 High Byte	0CDH
+TL2	Timer/Counter 2 Low Byte	0CCH
+RCAP2H	T/C 2 Capture Reg. High Byte	0CBH
+RCAP2L	T/C 2 Capture Reg. Low Byte	0CAH
*SCON	Serial Control	98H
SBUF	Serial Data Buffer	99H
PCON	Power Control	87H

* = Bit addressable
+ = 8052 only

รูปที่ 2.17 แสดงรีจิสเตอร์ใช้งานพิเศษ

รีจิสเตอร์ทำหน้าที่พิเศษ

รีจิสเตอร์ทำหน้าที่พิเศษ เป็นรีจิสเตอร์สำหรับการควบคุมหน้าที่และการทำงานของอุปกรณ์หรือพอร์ตของไมโครคอนโทรลเลอร์ทั้งหมด โดยมีตำแหน่งอยู่ในบริเวณแอดเดรส 80H- FFH การใช้งานรีจิสเตอร์หน้าที่พิเศษเหล่านี้สามารถทำได้ทั้งการระบุถึงชื่อของรีจิสเตอร์ หรือตำแหน่งแอดเดรสที่เป็นของรีจิสเตอร์นั้นก็ได้

สำหรับรีจิสเตอร์หน้าที่พิเศษนี้ มีข้อสังเกตว่ารีจิสเตอร์ที่อยู่ที่ตำแหน่งแอดเดรสที่เป็นจำนวนทวีคูณของค่า 8 จะสามารถอ้างถึงในระดับบิตได้ด้วย (นั่นคือแอดเดรส 80H, 88H, 90H, 98H, A0H, A8H, B0H, B8H, D0H, E0H และ F0H)

แอกคิวมูเลเตอร์ (Accumulator) หรือ ACC

เป็นรีจิสเตอร์ขนาด 8 บิต ทำหน้าที่ในการเก็บข้อมูลที่จะส่งให้กับหน่วยทำงานภายในซีพียูและเก็บผลลัพธ์ที่ได้จากการทำงานนั้น การทำงานจริงของรีจิสเตอร์นี้มีลักษณะเช่นเดียวกับตัวแอกคิวมูเลเตอร์ของโปรเซสเซอร์ทั่วไป การใช้งานภายในโปรแกรมจะเรียกว่า "รีจิสเตอร์ A"

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รีจิสเตอร์ B

เป็นรีจิสเตอร์ที่ใช้สำหรับการทำคำสั่งการคูณและหารตัวเลข ในกรณีที่ไม่ใช้ในการคำนวณทางด้านคณิตศาสตร์ ก็สามารถนำไปใช้งานเช่นเดียวกับรีจิสเตอร์ทั่วไปได้

โปรแกรมเคาน์เตอร์ (Program Counter)

เป็นรีจิสเตอร์ที่ใช้ในการชี้ตำแหน่งแอดเดรสของหน่วยความจำโปรแกรม ซึ่งจะต้องไปทำงานในลำดับถัดไป การใช้งานภายในโปรแกรมจะเรียกว่า "รีจิสเตอร์ PC"

สแต็กพอยน์เตอร์ (Stack Pointer)

เป็นรีจิสเตอร์ขนาด 8 บิต ทำหน้าที่เก็บตำแหน่งของตัวชี้หรือพอยเตอร์ (Pointer) ของบริเวณสแต็ก (Stack) สำหรับเก็บข้อมูลแอกคิวมูเลเตอร์ รีจิสเตอร์ต่าง ๆ รวมทั้งข้อมูลจากโปรแกรมโดยปกติแล้วเมื่อทำการเริ่มต้นระบบใหม่ภายหลังจากการเริ่มจ่ายไฟฟ้า หรือมีการรีเซต (Reset) เกิดขึ้นภายในสแต็กพอยน์เตอร์ จะมีค่า 07H ซึ่งเป็นตำแหน่งแอดเดรสภายในบริเวณเนื้อที่ 128 ไบต์แรกของหน่วยความจำข้อมูลภายใน การใช้งานภายในโปรแกรมจะเรียกว่า "รีจิสเตอร์ SP "

ตัวชี้ข้อมูลหรือดาต้าพอยน์เตอร์ (Data Pointer)

เป็นรีจิสเตอร์ขนาด 16 บิต ซึ่งเรียกว่า รีจิสเตอร์ DPTR และสามารถใช้งานแยกออกเป็นรีจิสเตอร์ขนาด 8 บิต สองตัว คือ รีจิสเตอร์ DPH และ DPL เพื่อเก็บค่าของแอดเดรสของหน่วยความจำที่จะต้องใช้งานภายในโปรแกรม หรืออาจจะเป็นแอดเดรสของอุปกรณ์ภายนอก ซึ่งกำหนดให้ติดต่อกันโดยใช้ตำแหน่งของหน่วยความจำนั้นภายในโปรแกรม

โปรแกรมสเตตัสเวิร์ด (PSW)

รีจิสเตอร์นี้ทำหน้าที่บอกถึงแฟล็กสภาวะการทำงานต่าง ๆ รวมทั้งบิตสำหรับการกำหนดเลือกแบงค์ของรีจิสเตอร์ที่ใช้งานด้วย

รีจิสเตอร์ที่เกี่ยวข้องกับพอร์ต (Port Register)

รีจิสเตอร์เหล่านี้จะมีความเกี่ยวข้องกับการทำงานของพอร์ตอินพุต/เอาท์พุตโดยตรง ซึ่งแต่ละตัวจะเป็นรีจิสเตอร์ขนาด 8 บิต สามารถใช้งานได้ทั้งในลักษณะของการอินพุต หรือการเอาท์พุตข้อมูลได้ การดำเนินการใด ๆ ที่เกี่ยวข้องกับพอร์ตทั้งสี่ตัวนี้จะมีผลทำให้ข้อมูลที่ตำแหน่งของพอร์ตเหล่านี้เปลี่ยนแปลงไปเช่นกัน นอกจากนี้พอร์ต P0 และ P2 ยังสามารถนำมาใช้ในการติดต่อกับหน่วยความจำโปรแกรมหรือหน่วยความจำข้อมูลภายนอกได้ โดยพอร์ต P2 จะเป็นค่าของแอดเด-รส 8 บิต บน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หน่วยความจำ ส่วนพอร์ต P0 นั้นในช่วงเริ่มแรกจะเป็นค่าของแอดเดรส 8 บิตล่างของหน่วยความจำ ช่วงเวลาต่อมาจึงจะนำพอร์ต P0 ไปใช้เป็นบัสสำหรับการรับหรือส่งข้อมูลกับหน่วยอุปกรณ์ภายนอก สำหรับพอร์ต P3 นั้นนอกเหนือจากจะใช้ในฐานะของพอร์ตอินพุต/เอาท์พุตเช่นปกติแล้ว ยังนำมาใช้ในฐานะบัสควบคุมเกี่ยวกับสัญญาณอินเตอร์รัปต์ได้อีกด้วย

รีจิสเตอร์ SBUF

เป็นบัฟเฟอร์ขนาด 8 บิตสำหรับการสื่อสารข้อมูลแบบอนุกรมทั้งการรับและส่งข้อมูล ซึ่งตามความเป็นจริงแล้วบัฟเฟอร์มีอยู่ด้วยกันสองชุด และแยกออกจากกันอย่างชัดเจน สำหรับการส่งและรับโดยซีพียูจะทำการจัดการเลือกบัฟเฟอร์ที่เหมาะสมให้โดยอัตโนมัติ

รีจิสเตอร์ PCON

เป็นรีจิสเตอร์ที่ใช้ในการควบคุมหน้าที่การทำงานในสามลักษณะได้แก่ การควบคุมการทำงานของโปรเซสเซอร์ (บิต IDL และ PD) การกำหนดอัตราการทวีคูณของอัตราเร็วในการสื่อสารข้อมูลอนุกรม (บิต SMOD) และแฟล็กสภาวะสำหรับการใช้งานทั่วไป (บิต GR0 และ GR1)

รีจิสเตอร์ IP, IE, TMOD, SCON

เป็นกลุ่มของรีจิสเตอร์ที่ทำหน้าที่กำหนดการควบคุม และการทำงานของการอินเตอร์รัปต์ต่าง ๆ ของ AT89C53 และ AT89Sxx

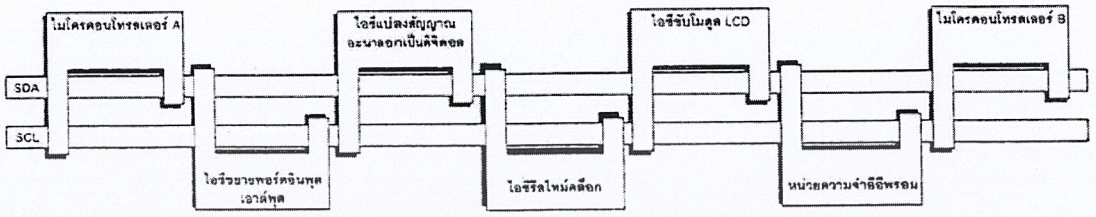
2.7 หลักการของระบบบัส I^2C

2.7.1 ความรู้เบื้องต้นเกี่ยวกับ I^2C

I^2C ย่อมาจาก Inter-IC Commuication หมายถึง การติดต่อสื่อสารระหว่างไอซีโดยบัส I^2C ได้รับการพัฒนาขึ้นโดยฟิลิปส์ (Philips) ด้วยจุดม่งหมายหลัก คือ ต้องการให้ไอซีหรือโมดูลสามารติดต่อสั่งงาน และควบคุมภายใต้สัญญาณเพียง 2 เส้น เส้นหนึ่ง คือ สายข้อมูล อีกเส้นหนึ่งคือ สายสัญญาณนาฬิกาที่ใช้ในการกำหนดจังหวะการทำงาน การต่อร่วมกันของอุปกรณ์บนบัส I^2C ทำได้ง่ายมาก เพียงต่อสายข้อมูลและสายสัญญาณของอุปกรณ์แต่ละตัวขนานหรือพ่วงกันไป ส่วนการกำหนดแอดเดรสหรือตำแหน่งสำหรับติดต่ออุปกรณ์แต่ละตัว จะใช้รหัสข้อมูลและการกำหนดสภาวะลอจิกที่ขาแอดเดรสของอุปกรณ์แต่ละตัว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สายข้อมูลบนบัส I^2C มีชื่อเรียกอย่างเป็นทางการว่า สายข้อมูลอนุกรม หรือ SDA (Serial Data Line) ส่วนสายสัญญาณนาฬิกามีชื่อเรียกว่า สายสัญญาณนาฬิกาอนุกรมหรือ SCL (Serial Clock Line)

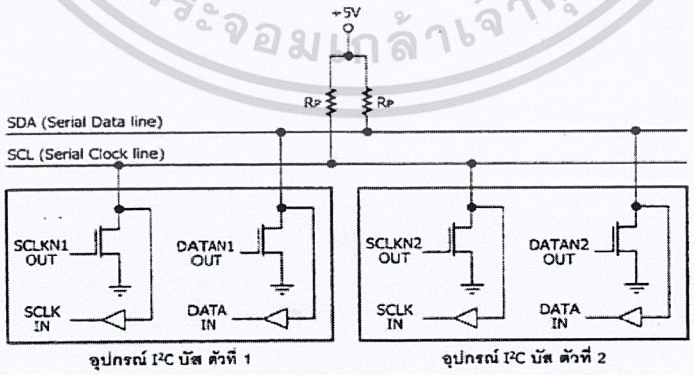


รูปที่ 2.18 ผังแสดงการเชื่อมต่อของอุปกรณ์ต่างๆ บนบัส I^2C

ในรูปที่ 2.18 แสดงผังการเชื่อมต่อของอุปกรณ์ต่างๆ บนบัส I^2C จะเห็นได้ว่าอุปกรณ์ที่ทำการเชื่อมต่อบนบัส I^2Cมีหลากหลาย ไม่ว่าจะเป็นไอซีขยายพอร์ตอินพุตเอาต์พุต (I/O Expander), ไอซีแปลงสัญญาณอนาลอกเป็นดิจิตอล(ADC) และแปลงสัญญาณดิจิตอลเป็นอนาลอก (DAC), ไอซีรีลไทม์คล็อก (RTC), ไอซีขับโมดูล LCD , หน่วยความจำอีอีพรอมและไมโครคอนโทรลเลอร์

2.7.2 คุณสมบัติโดยทั่วไปของบัส I^2C

สาย SDA และ SCL เป็นสายสัญญาณ 2 ทิศทาง (bi-directional-line) ต้องมีการต่อตัวต้านทานพูลอัพกับแรงดัน +5V ไว้ตลอดเวลาเพื่อให้สายมีสถานะลอจิกสูงในขณะที่ไม่มีการติดต่อใช้งาน ทั้งยังช่วยในการป้องกันสัญญาณรบกวนที่อาจมีเข้ามาในสายสัญญาณทั้งสอง วงจรเอาต์พุตของอุปกรณ์ที่ต่ออยู่บนบัส I^2C ต้องมีลักษณะเป็นวงจรเดรนเปิด (open-drain) หรือคอลเล็ก-เตอร์เปิด (open-collector) ดังแสดงรายละเอียดในรูปที่ 2.19



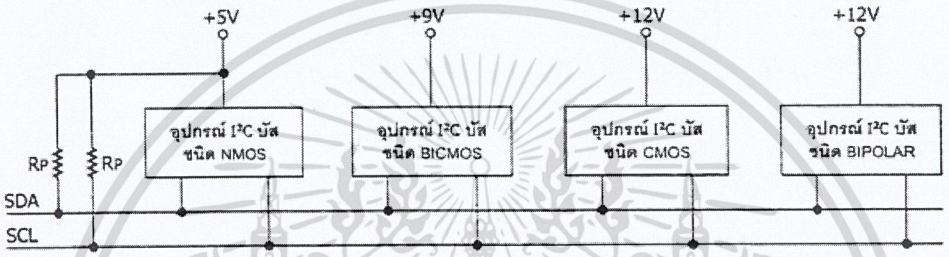
รูปที่ 2.19 แสดงวงจรเอาต์พุตของอุปกรณ์ในระบบบัส I^2C

อัตราการถ่ายทอดข้อมูลบนบัส I^2C สูงถึง 100 กิโลบิตต่อวินาทีในโหมดปกติ (stsndard - mode) และสูงถึง 400 กิโลบิตต่อวินาทีในโหมดความเร็วสูง (fast mode) อุปกรณ์ที่ต่อร่วมอยู่บนบัส I^2C จะต้องมีค่าความจุไฟฟ้ารวมที่เกิดขึ้นระหว่างสาย SDA และ SCL ไม่เกิน 400 pF การเข้า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

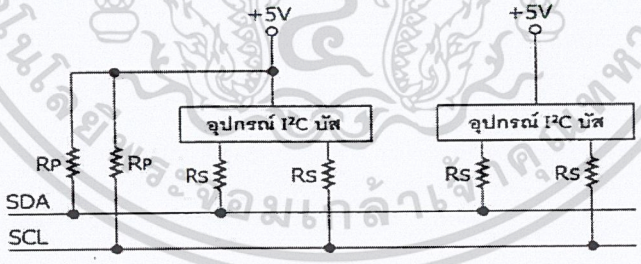
ถึงอุปกรณ์บนบัส I^2C ใช้ข้อมูลสำหรับการเข้าถึง 2 ค่า คือ 7 บิต (7-bit addressing) หรือ 10 บิต (10-bit addressing)

ข้อเด่นอีกประการหนึ่งของบัส I^2C คือ สามารถเชื่อมต่ออุปกรณ์ที่ใช้ไฟเลี้ยงไม่เท่ากันให้สามารถติดต่อสื่อสารกันได้ โดยอุปกรณ์บนบัส I^2C ตัวหนึ่งอาจใช้ไฟเลี้ยง +5V ในขณะที่อีกตัวหนึ่งใช้ไฟเลี้ยง +12V การต่อร่วมกันบนบัส I^2C สามารถกระทำได้ในลักษณะเดียวกับกรณีที่อุปกรณ์ทั้งสองใช้ไฟเลี้ยงเท่ากัน กล่าวคือ ให้ต่อสาย SDA และ SCL ของอุปกรณ์แต่ละตัวเข้าด้วยกัน และต้องต่อตัวต้านทานพูลอัพ (R_p) เข้ากับแรงดัน +5V ไว้ด้วยเสมอ ดังแสดงในรูปที่ 2.20



รูปที่ 2.20 การต่อตัวต้านทานพูลอัพบนสายสัญญาณในระบบบัส I^2C

ในกรณีที่อาจมีแรงดันไฟกระชากขนาดใหญ่ปะปนเข้ามาในบัส I^2C ที่ขา SDA และ SCL ของอุปกรณ์แต่ละตัวต้องต่อตัวต้านทานอนุกรมกับขา SDA และ SCL เรียกว่า Rs ก่อนต่อเข้าสู่บัส I^2C ดังแสดงในรูปที่ 2.21



รูปที่ 2.21การต่อตัวต้านทาน Rs เพื่อลดสัญญาณรบกวนขนาดใหญ่ที่อาจเข้ามาในบัส I^2C

2.7.3 หลักการของบัส I^2C

บัส I^2C ประกอบด้วยสายสัญญาณ 2 เส้น ดังที่ได้กล่าวมาแล้วคือ SDA และ SCL อุปกรณ์ที่ต่อพ่วงบนบัสสามารถมีได้มากมาย ดังนั้นจึงต้องมีการกำหนดรูปแบบของการติดต่อบนบัส หรือเรียกว่า โปรโตคอล (protocol) เพื่อให้ผู้ใช้งานทราบว่า ขณะนี้อุปกรณ์ใดติดต่อกันอยู่และอุปกรณ์ตัวใดเป็นตัวรับหรือตัวส่ง ต่อไปนี้จะอธิบายลักษณะ หน้าที่ และนิยามของอุปกรณ์ที่ต่ออยู่บนบัส I^2C เพื่อเป็นข้อตกลงพื้นฐานก่อนที่จะอธิบายการทำงานของบัส I^2C ต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น และไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อุปกรณ์ที่เป็นผู้สร้างข้อมูลหรือส่งข้อมูล เรียกว่า ตัวส่ง (transmitter)

อุปกรณ์ที่เป็นผู้รับข้อมูล เรียกว่า ตัวรับ (receiver) ในอุปกรณ์บนบัส I^2C สามารถเป็นได้ทั้งตัวรับและตัวส่ง บางอุปกรณ์ทำหน้าที่เป็นตัวรับเพียงอย่างเดียว จะไม่มีอุปกรณ์ใดบนบัส I^2C ที่ทำหน้าที่เป็นตัวส่งเพียงอย่างเดียว

อุปกรณ์ที่ทำหน้าที่ควบคุมจังหวะการติดต่อบนบัส I^2C เรียกว่า มาสเตอร์(Master)

อุปกรณ์ที่ถูกควบคุมหรืออุปกรณ์ที่ต่อพ่วงเข้าไปบนบัส I^2C เรียกว่า สเลฟ(Slave)

ข้อกำหนด 2 ประการสำคัญของการติดต่อบนบัส I^2C คือ

(1) การถ่ายทอดข้อมูลจะเกิดขึ้นได้เมื่อบัสว่างเท่านั้น

(2) ในระหว่างการถ่ายทอดข้อมูล เมื่อใดก็ตามที่สาย SCL มีสถานะเป็นลอจิกสูง สายข้อมูลต้องรักษาข้อมูลไว้ อย่าให้เกิดการเปลี่ยนแปลงขึ้นเด็ดขาด มิฉะนั้น สัญญาณที่เกิดขึ้นจะได้รับการแปลความหมายเป็นสัญญาณควบคุมแทน

2.7.4 **สภาวะที่เกิดขึ้นบนบัส I^2C**

มีด้วยกัน 5 สภาวะ ดังนี้

(1) **บัสว่าง** (bus not busy) สภาวะนี้เกิดขึ้นเมื่อสถานะลอจิกบนสาย SDA และ SCL เป็นลอจิกสูงทั้งคู่ นั่นหมายความว่า การถ่ายทอดข้อมูลสามารถเริ่มต้นขึ้นได้

(2) **เริ่มต้นการถ่ายทอดข้อมูล** (start data transfer) เกิดขึ้นเมื่อสาย SDA มีการเปลี่ยน แปลงระดับลอจิกจากสูงไปต่ำ ในขณะที่สาย SCL มีสถานลอจิกสูง เรียกสภาวะที่เกิดขึ้นนี้ว่า “สภาวะเริ่มต้น (START)”

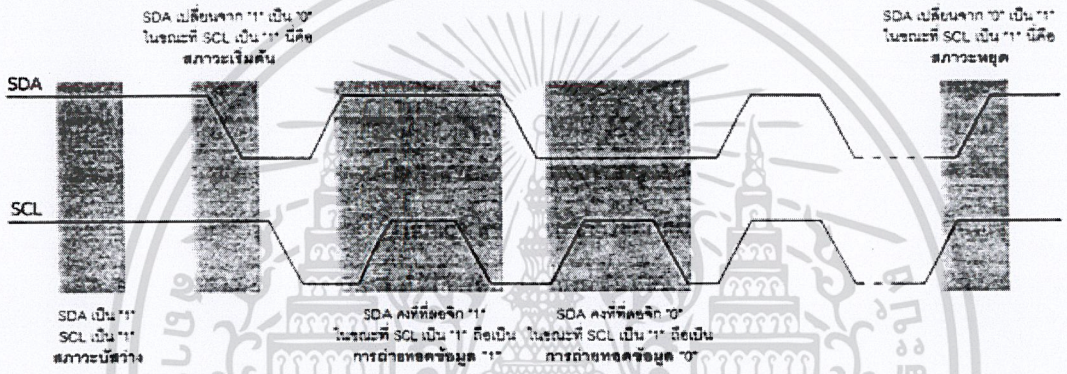
(3) **หยุดการถ่ายทอดข้อมูล** (stop data transfer) เกิดขึ้นเมื่อสาย SDA มีการเปลี่ยนแปลงระดับลอจิกจากต่ำไปสูง ในขณะที่สาย SCL มีสถานลอจิกสูง เรียกสภาวะที่เกิดขึ้นนี้ว่า “สภาวะหยุด (STOP)”

(4) **ข้อมูลดำรงอยู่บนบัส** (data valid) สภาวะนี้เกิดขึ้นถัดจากสภาวะเริ่มต้น โดยสถานะลอจิกที่เกิดขึ้นบนสาย SDA ก็คือข้อมูลที่ทำการถ่ายทอด เมื่อสาย SCL เป็นลอจิกสูง สถานะที่สาย SDA ต้องคงที่ เพื่อให้อุปกรณ์รับรู้ข้อมูลในจังหวะนั้นว่า เป็น “0” หรือ “1” ข้อมูลอาจเกิดการเปลี่ยนแปลงได้ในขณะที่สาย SCL เป็นลอจิกต่ำ แต่เมื่อใดก็ตามที่ต้องการให้เกิดการถ่ายทอดข้อมูลอย่างสมบูรณ์ สถานะลอจิกที่ขา SDA ต้องคงที่ ตลอดช่วงเวลาที่สาย SCL มีสถานะลอจิกสูง หากเกิดการเปลี่ยนแปลงสถานะลอจิกในขณะที่สาย SCL มีลอจิกสูงอยู่นั้น อุปกรณ์มาสเตอร์ที่ทำการควบคุมการถ่ายทอดข้อมูลจะแปลความหมายเป็นสภาวะหยุดหรือสภาวะเริ่มต้นก็ได้ ทำให้ข้อมูลที่ทำการถ่ายทอดนั้นเกิดความผิดพลาดขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

(5) **รับรู้ข้อมูล** (acknowledge) เกิดขึ้นหลังจากที่การถ่ายทอดข้อมูลจากตัวส่งมายังตัวรับเกิดขึ้นอย่างสมบูรณ์ โดยตัวส่งจะทำการส่งข้อมูลมา 1 บิตเรียกว่า “บิตรับรู้ (acknowledge bit)” มีสถานะเป็นลอจิกสูง หลังจากส่งข้อมูลมาครบถ้วน ส่วนอุปกรณ์มาสเตอร์จะทำการส่งสัญญาณรับรู้พิเศษซึ่งสัมพันธ์กับสัญญาณนาฬิกา เพื่อตอบสนองบิตรับรู้ที่ส่งมาจากตัวส่ง ทางด้านตัวรับจะส่งบิตรับรู้ที่ลอจิกต่ำลงบนบัส อุปกรณ์สเลฟที่ถูกอ้างถึงในการติดต่ออยู่ในขณะนั้นก็จะกำเนิดบิตรับรู้เพื่อตอบสนองให้ทราบว่าได้รับข้อมูลในแต่ละไบต์เรียบร้อยแล้ว

ในรูปที่ 2.22 เป็นไดอะแกรมเวลาที่แสดงถึงการเกิดสภาวะต่าง ๆ บนบัส I^2C ไม่ว่าจะเป็นสภาวะว่าง, เริ่มต้น, ถ่ายทอดข้อมูล, รับรู้ และ หยุดการถ่ายทอดข้อมูล



รูปที่ 2.22 ไดอะแกรมเวลาแสดงสถานะต่าง ๆ ในบัส I^2C

2.7.5 การทำงานบนบัส I^2C

ก่อนที่จะเริ่มต้นการถ่ายทอดข้อมูลระหว่างอุปกรณ์ต่าง ๆ ที่ต่ออยู่บนบัส ต้องมีการอ้างถึงเสียก่อน โดยการอ้างถึงอุปกรณ์บนบัส I^2C นั้นจะใช้การอ้างถึงแบบ 7 บิต หรือ 10 บิต ในกรณีที่มีอุปกรณ์ต่อยู่บนบัสไม่มาก ใช้การอ้างถึงแบบ 7 บิตก็เพียงพอ แต่ถ้ามีอุปกรณ์ต่ออยู่บนบัสมาก กว่า 127 แอดเดรส จำเป็นต้องใช้การอ้างถึงแบบ 10 บิต หลังจากที่ติดต่ออุปกรณ์แต่ละตัวได้เรียบร้อยแล้วก็จะเริ่มต้นการถ่ายทอดข้อมูลกันต่อไป

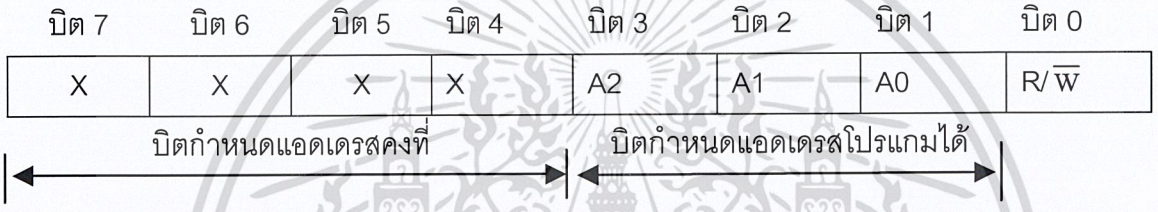
ดังนั้นหัวใจสำคัญในอันดับแรกของการทำงานบนบัส I^2C คือ การอ้างถึงอุปกรณ์แต่ละตัวซึ่งในที่นี้จะอธิบายรายละเอียดของการอ้างถึงทั้ง 2 รูปแบบ

2.7.6 การอ้างถึงแบบ 7 บิต (7-bit addressing)

ข้อมูลไบต์แรกที่เกิดขึ้นหลังจากสภาวะเริ่มต้นคือข้อมูลที่ใช้ในการอ้างถึงอุปกรณ์ที่ต้องการติดต่อ หรือ ข้อมูลกำหนดแอดเดรส โดยมีรูปแบบแสดงในรูปที่ 2.23 ใน 7 บิตบนรวมทั้งบิต MSB ด้วยจะเป็นข้อมูลแอดเดรสของอุปกรณ์สเลฟที่ต้องการติดต่อ โดยแบ่งเป็น บิตกำหนดแอดเดรสคงที่ (fixed address bit) จำนวน 4 บิต ซึ่งข้อมูลนี้อุปกรณ์แต่ละตัวจะถูกกำหนดมาจากผู้ผลิต ไม่สามารถเปลี่ยน

แปลงแก้ไขได้ ถัดมาอีก 3 บิต เป็นบิตกำหนดแอดเดรสที่สามารถโปรแกรมได้ (programmable adderss bit) โดยผู้ใช้งานต้องกำหนดสถานะลอจิกในบิต LSB เป็นบิตที่ใช้กำหนดการอ่านหรือเขียนข้อมูลกับอุปกรณ์สเลฟตัวนั้น ๆ หากบิต LSB เป็น "0" หมายถึงต้องการเขียนข้อมูลไปยังอุปกรณ์นั้น ถ้าเป็น "1" จะเป็นการอ่านข้อมูลจากอุปกรณ์สเลฟ

ข้อมูลในไบต์ต่อมา คือ ข้อมูลควบคุม (control byte) ในอุกรณ์แต่ละตัวมีการกำหนดข้อมูลควบคุมที่แตกต่างกันไป ยกตัวอย่างไอซีขยายพอร์ตมีข้อมูลควบคุมที่ใช้กำหนดว่า บิตใดเป็นอินพุต บิตใดเป็นเอาต์พุต ในขณะที่ไอซี ADC/DAC ต้องการข้อมูลควบคุมเพื่อกำหนดให้ทำงานเป็นวงจร ADC หรือ DAC เป็นต้น



รูปที่ 2.23 รูปแบบของข้อมูลกำหนดแอดเดรสที่ใช้ในการอ้างถึงแบบ 7 บิต

ข้อมูลในไบต์ต่อมา คือ ข้อมูลที่ทำการถ่ายทอดจริง (data)

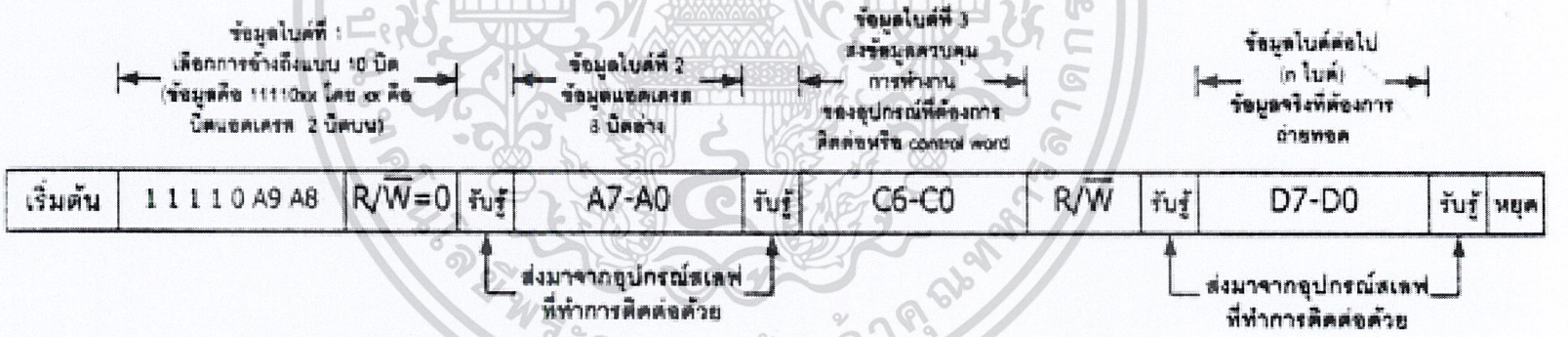
หลังจากที่มีการถ่ายทอดข้อมูลในไบต์ อุปกรณ์สเลฟที่ได้รับการติดต่อต้องส่งสัญญาณรับรู้ตอบกลับมาด้วยทุกครั้ง เพื่อให้กระบวนการถ่ายทอดข้อมูลสามารถดำเนินต่อไปได้ ในรูปที่ 2.24 แสดงรูปแบบข้อมูลอนุกรมที่เกิดขึ้นในการติดต่อบนบัส I^2C ของการอ้างถึงแบบ 7 บิต

2.7.7 การอ้างถึงแบบ 10 บิต

ในการอ้างถึงแบบนี้ยังคงใช้รูปแบบข้อมูลอนุกรมที่เหมือนกันแบบ 7 บิต หากแต่จะมีข้อมูลเพิ่มเติมขึ้นมาเล็กน้อย โดยในข้อมูลไบต์แรกหลังจากเกิดสภาวะเริ่มต้น ต้องกำหนดให้ 5 บิตบนมีข้อมูล เป็น 11110 ส่วนอีก 2 บิตถัดมาเป็นบิตแอดเดรสของอุปกรณ์ที่ต้องการติดต่อ ในบิต LSB ของข้อมูลเป็นไบต์แรกยังคงเป็นการกำหนดว่าต้องการอ่านหรือเขียนข้อมูลกับอุปกรณ์สเลฟตัวที่ต้องการติดต่อด้วย ข้อมูลไบต์ต่อมาข้อมูลแอดเดรสในไบต์ที่ 2 ของอุปกรณ์ที่ต้องการติดต่อด้วย ข้อมูลไบต์ถัดไปจึงเป็นข้อมูลควบคุม ข้อมูลหลังจากนั้นจะเป็นข้อมูลจริงที่ใช้ในการติดต่อเช่นเดียวกับการอ้างถึงแบบ 7 บิต หลังจากถ่ายทอดข้อมูลครบทุกไบต์ ต้องมีสภาวะรับรู้เกิดขึ้น เพื่อให้กระบวนการถ่ายทอดข้อมูลสามารถดำเนินต่อไปได้ ในรูปที่ 2.25 แสดงรูปแบบขอมูลอนุกรมของการอ้างถึงแบบ 10 บิต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 2.24 รูปแบบของข้อมูลอนุกรมที่ใช้ติดต่อกับอุปกรณ์บัส I^2C เมื่อใช้การอ้างถึงแบบ 7 บิต



รูปที่ 2.25 รูปแบบของข้อมูลอนุกรมที่ใช้ติดต่อกับอุปกรณ์บัส I^2C เมื่อใช้การอ้างถึงแบบ 10 บิต

2.7.8 อุปกรณ์ที่ใช้เชื่อมต่อแบบบัส I^2C

ในปัจจุบันบัส I^2C ได้รับความนิยมมากขึ้นเรื่อยๆด้วยข้อดีที่ชัดเจนคือ ใช้สายสัญญาณเพียง 2 เส้นเท่านั้น และการขยายระบบไมโครคอนโทรลเลอร์ที่มีจำนวนอินพุตเอาท์พุตและหน่วยความจำจำกัดสามารถทำได้ง่ายขึ้นด้วยระบบบัส I^2C เมื่อเป็นเช่นนี้จึงมีอุปกรณ์เพอริเฟอรัลที่ใช้การเชื่อมต่อแบบบัส I^2C มากมายจากหลายผู้ผลิตออกมาให้ใช้งานกัน ดังมีตัวอย่างต่อไปนี้

ไอซีขยายพอร์ตอินพุตเอาท์พุต (I/O expander) : PCF8574, PCF8582, PCF8584

ไอซีหน่วยความจำอีอีพรอมอนุกรม (Serial EEPROM) : 24Cxx, PCF8570, PCF72/73, PCF8582

ไอซี ADC/DAC : PCF8591

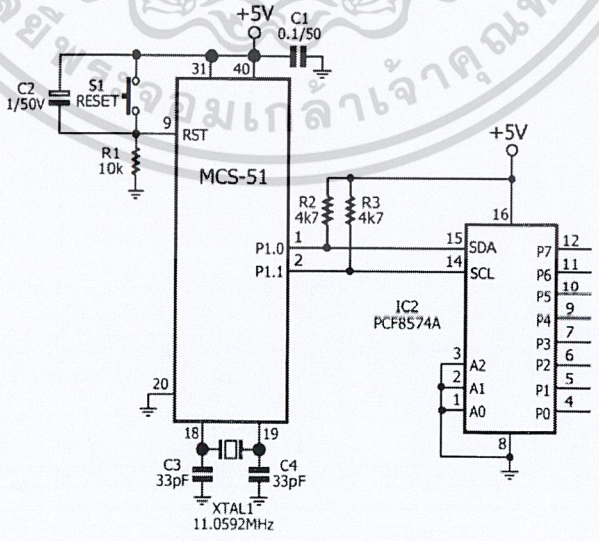
ไอซีรีลไทม์คล็อก (Real Time Clock : RTC) : PCF8583, PCF8593, PCF8598, 41T56C

ไอซีขับ LCD โมดูล (LCD driver) : PCF8466, PCF8576, PCF8577/78, PCF8579, SAA1064

ไอซีกำเนิดสัญญาณ DTMF (DTMF generator) : PCD3311/12

2.7.9 การต่ออุปกรณ์ระบบบัส I^2C กับไมโครคอนโทรลเลอร์ MCS-51

สามารถทำได้ง่ายมาก เพียงใช้ขาพอร์ต 2 ขา โดยกำหนดให้ขาหนึ่งเป็น SDA อีกขาหนึ่งเป็น SCL และต่อตัวต้านทานค่าประมาณ 4.7K พูลอัพที่ขาพอร์ตทั้ง 2 ขา เพียงเท่านี้ก็สามารถติดต่อกับอุปกรณ์ระบบบัส I^2C ได้แล้ว



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 2.26 วงจรตัวอย่างการต่อไมโครคอนโทรลเลอร์ MCS-51 กับอุปกรณ์ระบบบัส I^2C

ในรูปที่ 2.26 เป็นวงจรตัวอย่างการต่อไมโครคอนโทรลเลอร์ MCS-51 เข้ากับระบบบัส I^2C จากวงจรจะใช้ขาพอร์ต P1.0 เป็นขา SDA และ P1.1 เป็นขา SCL อุปกรณ์ที่ทำการติดต่อด้วยคือ ไอซีขยายพอร์ตอินพุตเอาท์พุตเบอร์ PCF8574

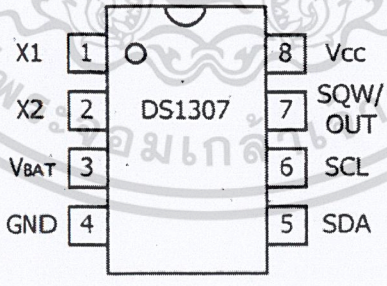
2.8 ไอซีสร้างฐานเวลาจริงเบอร์ DS1307 (RTC : Real Time Clock)

การทำงานคือจะทำหน้าที่สร้างฐานเวลาจริงให้แก่ระบบไมโครคอนโทรลเลอร์โดย DS1307 จะให้ข้อมูลเกี่ยวกับเวลาทั้งหมดไม่ว่าจะเป็นค่าของเวลาที่ละเอียดถึงหลักวินาที, นาที, ชั่วโมง, วันที่ (date), วันในสัปดาห์ (day), เดือน และปี โดยสามารถปรับวันเดือนปีให้ตรงตามปฏิทินได้อย่างถูกต้อง รวมถึงการกำหนดวันในปีอธิการทินด้วย คุณสมบัติทางเทคนิคที่สำคัญมีดังนี้

- เป็นไอซีรีลไทม์คล็อก (RTC) ให้ข้อมูลตั้งแต่วินาทีจนถึงปี รวมถึงการปรับวันในปีอธิกสุรทินด้วย สามารถให้ข้อมูลเวลาได้อย่างเที่ยงตรงถึงปีคริสตศักราช 2100
- มีหน่วยความจำนอนโวลาไทล์แรม 56 ไบต์อยู่ภายใน สามารถใช้เก็บข้อมูลทั่วไปได้
- ใช้การเชื่อมต่อแบบระบบบัส I^2C
- มีวงจรตรวจจับไฟเลี้ยงต่ำหรือหายไปอย่างอัตโนมัติ และสามารถรักษาข้อมูลเวลาไว้ได้แม้ไม่มีไฟเลี้ยงไอซี

2.8.1 รายละเอียดการต่อใช้งานของ DS1307

ในรูปที่ 2.27 แสดงการจัดขาของ DS1307 แต่ละขามีหน้าที่และการใช้งานดังนี้



รูปที่ 2.27 การจัดขาของไอซี DS1307

V_{cc} , GND (ขา 8,4) ต่อกับไฟเลี้ยง +5 V

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

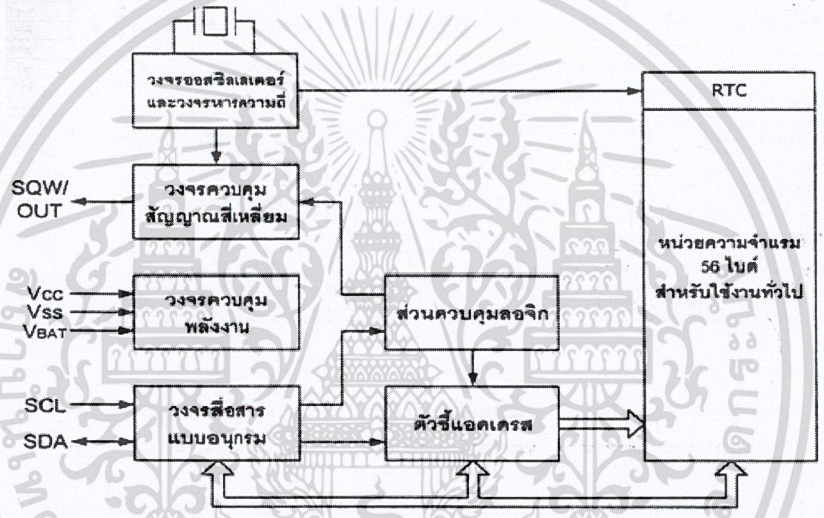
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

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

SQW/OUT (ขา 7) ที่ขานี้จะมีสัญญาณรูปสี่เหลี่ยมออกมา โดยสามารถเลือกความถี่ได้ 1 Hz, 4.096 kHz, 8.192kHzและ 32KHz ในการใช้งานต้องต่อตัวต้านทาน 1 k Pull up ที่ขานี้ด้วย

X1, X2 (ขา 1, 2) ใช้ต่อกัยคริสตอลความถี่มาตรฐาน 32.798 kHz เพื่อใช้เป็นฐานเวลาในการสร้างค่าเวลาจริง ในการใช้งานต้องต่อคริสตอลเข้ากับขาทั้งสองนี้และที่แต่ละขาต้องต่อตัวเก็บประจุค่าต่ำ ๆ ประมาณ 15 pF คร่อมกับขากราวด์ด้วย

2.8.2 การทำงานของ DS1307



รูปที่ 2.28.โครงสร้างภายในของไอซีรีลไทม์คล็อกเบอร์ DS1307

ไอซี DS1307 จัดการเชื่อมต่อในแบบบัส I^2C โดยจะทำงานเป็นอุปกรณ์สเลฟเสมอ ดังนั้นการติดต่อเพื่อใช้งานจึงต้องกำหนดรูปแบบตามที่กำหนดไว้ในการติดต่อแบบ I^2C ในรูปที่ 2.28 แสดงส่วนประกอบหลักที่สำคัญและไดอะแกรมการทำงานของ DS1307 วงจรออสซิลเลเตอร์ถือเป็นหัวใจหลักของไอซี เนื่องจากเป็นจุดเริ่มต้นของการสร้างข้อมูลเวลาจริง ในขณะที่ DS1307 ทำงานที่ขา SQW/OUT จะมีสัญญาณพัลส์สี่เหลี่ยมส่งออกมาตลอดเวลาในกรณีที่มีการอีนาเบิล วงจรกำเนิดสัญญาณพัลส์ที่รีจิสเตอร์ควบคุม ค่าความถี่ของสัญญาณนี้สามารถเลือกได้ 4 ค่า คือ 1 Hz, 4.096 kHz, 8.192 kHz และ 32 kHz พร้อมกันนั้นก็จะมีการเก็บค่าของเวลาไว้ในหน่วยความจำนอนโวลาไทล์แรม ซึ่งมีขนาดรวม 64 ไบต์ แต่จัดสรรให้ใช้เก็บข้อมูลเวลา 8 ไบต์ และเป็นหน่วยความจำสำหรับเก็บข้อมูลทั่วไปสำหรับผู้ใช้งานอีก 56 ไบต์

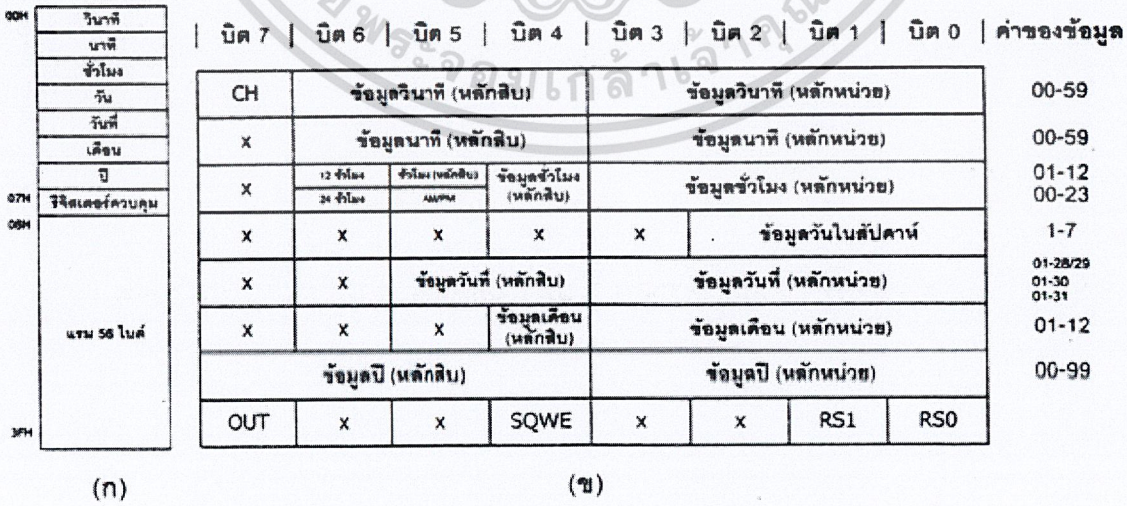
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

วงจรควบคุมพลังงานไฟฟ้าจะคอยตรวจสอบสภาวะของไฟเลี้ยงไอซี หากไฟเลี้ยงต่ำกว่า 1.25x V_{BAT} ก็จะควบคุมให้ DS1307 หยุดการทำงานรีเซตค่าตัวนับแอดเดรสภายในทำให้ไม่สามารถติดต่อกับ DS1307 ได้ ดังนั้นในการใช้งาน DS1307 ต้องระวังอย่าให้ไฟเลี้ยงต่ำกว่า V_{BAT} ไอซี DS1307 จะเข้าสู่โหมดสำรองข้อมูลกระแสต่ำทันที จะไม่มีการส่งสัญญาณพัลส์ออกมาที่ขา SQW/OUT แต่วงจรสร้างฐานเวลายังคงทำงานเพื่อให้ค่าของเวลาเดินไปอย่างไม่ผิดพลาด เมื่อมีไฟเลี้ยงปรากฏขึ้นอีกครั้ง DS1307 ก็จะสามารถให้ค่าของเวลาที่เป็นจริงแก่ผู้ใช้งานได้ต่อไป

วงจรสื่อสารอนุกรมใน DS1307 ได้รับการกำหนดให้ทำงานตามรูปแบบของบัส I^2C เป็นช่องทางการสื่อสารระหว่าง DS1307 กับอุปกรณ์มาสเตอร์ ผู้ใช้งานสามารถเข้าถึงหน่วยความจำที่ใช้เก็บค่าเวลาและหน่วยความจำใช้งานทั่วไปได้โดยการเขียนข้อมูลตามรูปแบบที่กำหนดในระบบ I^2C

2.8.3 การจัดสรรหน่วยความจำใน DS1307

ในรูปที่2.29 (ก) แสดงการจัดสรรพื้นที่ของหน่วยความจำภายใน DS1307 พื้นที่ 7 ไบต์แรกตั้งแต่แอดเดรส 00H-06H เป็นพื้นที่ของรีจิสเตอร์ค่าเวลาใช้ในการเก็บข้อมูลเกี่ยวกับเวลาไบต์ต่อมาที่แอดเดรส 07H เป็นพื้นที่แบบนี้ ทำให้ผู้ใช้งานสามารถเรียกข้อมูลเวลาออกมาได้ตามที่ต้องการโดยไม่จำเป็นต้องอ่านออกมาทั้งหมดก็ได้ ค่าของเวลาทั้งหมดจะอยู่ในรูปของเลขฐานสิบ สำหรับการแสดงเวลาในรูปของชั่วโมง สามารถเลือกได้ว่าต้องการแบบ 12 หรือ 24 ชั่วโมง โดยกำหนดที่บิต 5 ของแอดเดรส 02H และเมื่อเลือกแบบ 12 ชั่วโมง ที่บิต 5 ในแอดเดรสเดียวกันจะใช้ในการแสดงค่า AM/PM โดยถ้าบิตนี้เป็น “1” หมายถึง ค่าชั่วโมงในขณะนี้เป็นช่วงเวลาหลังเที่ยงวัน ในกรณีที่เป็นแบบ 24 ชั่วโมง บิตนี้จะใช้ในการแสดงค่า 2 ของหลักสิบในหน่วยชั่วโมง



รูปที่ 2.29 (ก) การจัดสรรหน่วยความจำแรมภายใน DS1307
(ข) รายละเอียดของรีจิสเตอร์เก็บค่าเวลาและรีจิสเตอร์ควบคุมของ DS1307

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.8.4 รีจิสเตอร์ควบคุม

มีแอดเดรสอยู่ที่ 07H มีรายละเอียดของแต่ละบิตดังนี้

OUT (Output Control) : ใช้ในการควบคุมระดับลอจิกที่ขา SQW/OUT ในกรณีที่ดิสเอ-เบิลการกำเนิดสัญญาณสี่เหลี่ยม โดยถ้าบิตนี้ป็น “1” ที่ขา SQW/OUT ก็จะเป็น “1” ถ้าบิตนี้เป็น “0” ที่ขา SQW/OUT ก็จะเป็น “0”

SQWE (Square Wave Enable) : ใช้ในการอีนาเบิลวงจรกำเนิดสัญญาณสี่เหลี่ยมที่ขา SQW/OUT ถ้าต้องการให้มีสัญญาณสี่เหลี่ยมออกให้กำหนดบิตนี้เป็น “1”

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

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

2.8.5 โหมดการทำงานของ DS1307

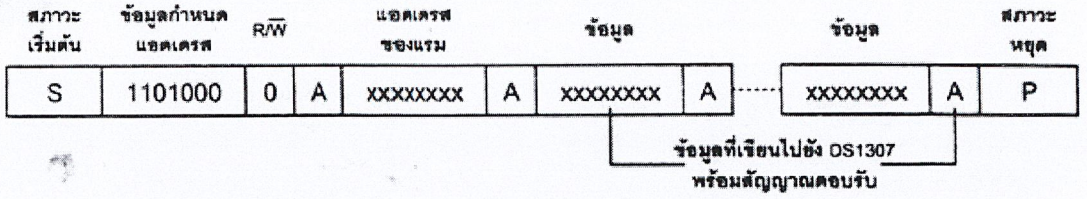
มีด้วยกัน 2 โหมด คือ โหมดเขียนข้อมูลและโหมดอ่านข้อมูลในการใช้งาน DS1307 ตามปกติจะใช้งานเฉพาะโหมดอ่านข้อมูลเท่านั้น เนื่องจากไมโครคอนโทรลเลอร์จะติดต่อกับ DS1307 เพื่ออ่านข้อมูลของเวลาไปใช้งาน โหมดการเขียนข้อมูลจะถูกใช้งานก็ต่อเมื่อต้องการตั้งค่าเวลาใหม่และต้องการเขียนข้อมูลลงในหน่วยความจำใช้งานทั่วๆไป อย่างไรก็ตามเมื่อเริ่มต้นติดต่อกับ DS1307 จำเป็นอย่างยิ่งที่จำต้องเข้าสู่โหมดการเขียนข้อมูลก่อนเพื่อกำหนดแอดเดรสที่ต้องการอ่านข้อมูลจากนั้นจึงเปลี่ยนโหมดการทำงานเป็นโหมดการอ่านข้อมูลต่อไป

(1) โหมดการเขียนข้อมูล

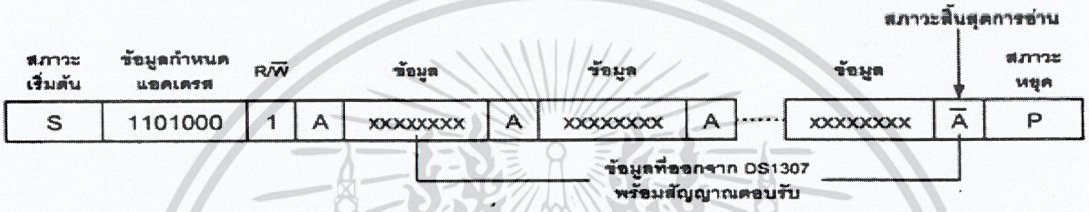
มีรูปแบบดังในรูปที่ 2.30 เริ่มต้นเมื่อไมโครคอนโทรลเลอร์ทำการกำหนดสภาวะเริ่มต้น (START : S) จากนั้นส่งข้อมูลกำหนดแอดเดรส 1101000 ตามด้วยข้อมูลเลือกการเขียน นั่นคือค่า 0 จากนั้นจะรอการตอบรับจาก DS1307 ขั้นตอนต่อมา คือ ส่งข้อมูลเพื่อเลือกแอดเดรสที่ต้องการเขียนจากนั้นรอการตอบรับจาก DS1307 เมื่อมีการตอบรับมาเรียบร้อย ก็เริ่มทยอยเขียนข้อมูลลงไปครั้งละแอดเดรส หลังจากเขียนข้อมูลในแต่ละแอดเดรส จะต้องหยุดรอการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตอบรับจาก DS1307 ทุกครั้งจึงจะสามารถเขียนข้อมูลต่อไปได้ เมื่อเขียนเรียบร้อยแล้วให้ส่งสภาวะหยุด (STOP : P) เป็นอันสิ้นสุดกระบวนการเขียนข้อมูล



รูปที่ 2.30 รูปแบบ ของข้อมูลสำหรับติดต่อกับ DS1307 ในโหมดการเขียนข้อมูล



รูปที่ 2.31 รูปแบบ ของข้อมูลสำหรับติดต่อกับ DS1307 ในโหมดการอ่านข้อมูล

(2) โหมดการเขียนข้อมูล

มีรูปแบบแสดงในรูปที่ 2.31เริ่มต้นการทำงานเหมือนกับโหมดการเขียนข้อมูล คือ ไมโครคอนโทรลเลอร์ กำหนดสภาวะเริ่มต้นแล้วส่งข้อมูลกำหนดแอดเดรสตามด้วยข้อมูลเลือกการอ่านซึ่งค่าเท่ากับ 1 จากนั้นรอการตอบรับจาก DS1307 เมื่อตอบรับเรียบร้อยแล้ว DS1307 จะทยอยส่งข้อมูลออกมาให้ไมโครคอนโทรลเลอร์คราวละ 1 แอดเดรสหรือ 1 ไบต์ โดยแอดเดรสที่เลือกอ่านข้อมูลจะต้องมีการกำหนดมาก่อนล่วงหน้าด้วยโหมดการเขียนข้อมูล วิธีการง่ายๆ คือ เข้าสู่โหมดการเขียนข้อมูลก่อน เมื่อถึงจังหวะที่ต้องเขียนข้อมูลให้ทำการสร้างสภาวะเริ่มต้นและส่งข้อมูลกำหนดแอดเดรสใหม่อีกครั้ง ตามด้วยเลือกโหมดการอ่านข้อมูล ข้อมูลที่ออกมาจาก DS1307 ก็จะเป็นข้อมูลจากแอดเดรสที่กำหนดไว้ก่อนหน้านี้

2.9 LCD (Liquid Crystral Displays)

2.9.1 **รายละเอียดเกี่ยวกับโมดูล** LCD

ในส่วนแสดงผลแบบผลึกเหลวจะมีส่วนประกอบหลักๆ 3 ส่วนดังนี้

ตัวแสดงผล (Display) ภายในเป็นผลึกเหลวที่สามารถแสดงผลให้เห็นโดยอาศัยแสงจากภายนอก ดังนั้นจึงต้องมีมุมในการมองข้อมูลที่แสดงผลบนส่วนแสดงผลแบบผลึกเหลว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวควบคุม (Controller) เป็นตัวรับข้อมูลจากอุปกรณ์ภายนอกมาควบคุมการทำงานของส่วนแสดงผลแบบผลึกเหลว เช่นลบจอภาพ แสดงตัวอักษรหรือเลื่อนเคอร์เซอร์ เป็นต้น ตัวควบคุมนั้นใช้ชิปควบคุมโดยเฉพาะชิปที่นิยมใช้ คือ เบอร์ HD4478 และ HD61830 โดย HD44780 จะใช้ควบคุม LCD แบบ อักขระ ส่วน HD61830 ใช้ควบคุม LCD แบบกราฟฟิก

ตัวขับ (Driver) เป็นตัวรับสัญญาณจากตัวควบคุมมาขับให้ตัวแสดงผลแสดงข้อมูลตามที่กำหนด ชิปที่ใช้ทำหน้าที่เป็นตัวขับนี้ได้แก่ เบอร์ HD44100H และ MSM5259 เป็นต้น

2.9.2 โครงสร้างภายในของตัวควบคุมโมดูล LCD

ในการใช้งานโมดูล LCD จำเป็นต้องทำความเข้าใจเกี่ยวกับโครงสร้างและคำสั่งที่ใช้ในการควบคุมให้ดีเสียก่อน ในที่นี้ของยกตัวอย่างโมดูล LCD แบบอักขระ ซึ่งเป็น LCD ที่ใช้ในตัว project จริงๆ เพราะสามารถเข้าใจได้ง่าย ให้รูปที่ ... เป็นบล็อกไดอะแกรมภายในของชิปควบคุม LCD เบอร์ HD44780 ซึ่งใช้ในโมดูล LCD แบบอักขระ ประกอบด้วย

บัฟเฟอร์อินพุทเอาท์พุท เป็นส่วนที่ใช้ในการติดต่อรับส่งข้อมูลกับอุปกรณ์ภายนอก เพื่อที่จะถ่ายทอดข้อมูลเข้าออกภายในตัวควบคุม

รีจิสเตอร์คำสั่ง (Instryction Register : IR) เป็นรีจิสเตอร์ใช้รับข้อมูลคำสั่งจากอุปกรณ์ภายนอก เพื่อนำไปควบคุมการแสดงผล

รีจิสเตอร์ข้อมูล (Data Register : DR) เป็นรีจิสเตอร์ใช้รับข้อมูลจากอุปกรณ์ภายนอกเพื่อถ่ายทอดต่อไปยังหน่วยความจำที่ทำหน้าที่เก็บข้อมูลแสดงผล หรือนำข้อมูลไปสร้างตัวอักษรเพิ่มเติมในแรมตัวอักษร

แรมเก็บข้อมูลแสดงผล (Display Data RAM : DDRAM) เป็นหน่วยความจำแรมทำหน้าที่เก็บข้อมูลที่มาจากรีจิสเตอร์ DR ตัวควบคุมจะนำข้อมูลใน DDRAM นี้ไปเปิดตาราง (Look up lable) ของตัวอักษรที่เก็บไว้ในหน่วยความจำรอมและแรมเก็บตัวอักษร เพื่อนำไปแสดงที่ตัวแสดงผล

รอมเก็บตัวอักษร (Character Generator ROM : CGROM) เป็นหน่วยความจำรอมที่ใช้เก็บข้อมูลตัวอักษรหรือสัญลักษณ์ที่สามารถอ่านออกไปแสดงผลที่ตัวแสดงผลได้ มีขนาด 7,200 บิต โดยจะถูกอ่านด้วยค่าของข้อมูลใน DDRAM

แรมเก็บตัวอักษร (Character Generator RAM : CGRAM) เป็นหน่วยความจำแรมที่ใช้เก็บอักษรที่มีการสร้างเพิ่มเติมขึ้นใหม่ ในกรณีที่ตัวอักษรใน CGROM ไม่เพียงพอ มีขนาด 512 บิต การเขียนและอ่านค่าไปใช้นั้นทำได้เช่นเดียวกับ CGROM คือเขียนข้อมูลลงใน DDRAM แล้วตัวควบคุมจะมาอ่านค่าจาก CGRAM เอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แฟลก BUSY เป็นส่วนที่ทำหน้าที่แจ้งสถานะการทำงานของตัวควบคุมให้อุปกรณ์ภายนอกทราบว่า ตัวควบคุมพร้อมที่จะรับข้อมูลหรือคำสั่งหรือไม่ ดังนั้นก่อนการส่งข้อมูลหรือคำสั่งมายังตัวควบคุมต้องตรวจสอบสถานะของแฟลก BUSY นี้เสียก่อน

2.9.3 โมดูล LCD ขนาด 20 ตัวอักษร 4 บรรทัด (LCD 20X4)

ส่วนแสดงผลแบบผลึกเหลวที่นำมาใช้เป็นขนาด 20 ตัวอักษร 4 บรรทัด มีขาต่อใช้งาน 16 ขา ซึ่งมีรายละเอียดดังต่อไปนี้

Vss (ขา1) ต่อกราวด์

Vdd (ขา2) ต่อไฟเลี้ยง +5 โวลต์

Vo (ขา3) เป็นขาอินพุตรับแรงดันเพื่อปรับความเข้มของการแสดงผล

RS (ขา4) เป็นขาอินพุตใช้ในการแยกชนิดข้อมูลที่ทำการประมวลผลในขณะนั้นว่าเป็นคำสั่งสำหรับรีจิสเตอร์ IR หรือเป็นข้อมูลสำหรับรีจิสเตอร์ DR โดยถ้าขานี้เป็น “0” ข้อมูลที่ส่งมาจะเป็นคำสั่ง แต่ถ้าขานี้เป็น “1” ข้อมูลที่ส่งมาจะเป็นข้อมูลสำหรับการแสดงผล

R/$\overline{W}$ (ขา5) เป็นขาที่ใช้เลือกการอ่านหรือเขียนข้อมูลกับ LCD ถ้าเป็น “0” เป็นการกำหนดให้เขียนข้อมูล แต่ถ้าเป็น “1” จะเป็นการอ่านข้อมูล

E (ขา6) เป็นขาอีนาเบิล LCD ทำงาน

D0-D7 (ขา 7-14) เป็นขาที่ใช้เป็นทางผ่านของข้อมูลระหว่าง ส่วนแสดงผลแบบผลึกเหลวกับอุปกรณ์ภายนอกขนาด 8 บิต

BLACK LIGHT (ขา15-16) เป็นขาที่ใช้สำหรับให้แสงสว่างบนหน้าจอ LCD โดยที่ขา 15 จะต้องต่อเข้ากับไฟบวก ส่วนขา 16 ต่อลงกราวด์

2.9.4 คำสั่งควบคุมโมดูล LCD

ในการเขียนคำสั่งลงในตัวควบคุมแน่นอนว่าต้องกำหนดให้ขา RS และ R/$\overline{W}$ เป็น “0” แล้วเขียนคำสั่งตามไป คำสั่งควบคุมโมดูล LCD ของชิบควบคุม HD44780 ที่สำคัญมี 10 คำสั่งดังนี้

1. **คำสั่ง เคลียร์ตัวแสดงผล** (Clear display)

มีข้อมูลคำสั่งเป็น 01H เป็นคำสั่งที่ใช้เขียนข้อมูลช่องว่าง หรือ space เข้าไปใน DDRAM ทั้งหมด เมื่อตัวควบคุมเอ็กซิคิวท์คำสั่งนี้ จะทำการกำหนดแอดเดรสของ DDRAM เป็น 0 เคอร์เซอร์จะกลับไปอยู่ที่ตำแหน่งซ้ายมือสุดของจอแสดงผล แล้วเซตบิต I/D (ซึ่งจะกล่าวถึงภายหลัง) ให้เป็น ”1”

2. **คำสั่ง** Return home

ต้องกำหนดให้บิต 1 ของข้อมูลเป็น “1” เป็นคำสั่งให้เคอร์เซอร์เคลื่อนที่กลับไปยังตำแหน่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ซ้ายสุดของจอแสดงผล แต่ข้อมูลบนจอแสดงผลไม่เปลี่ยนแปลงนั่นคือ ข้อมูลของคำสั่งนี้จะเป็น 02H หรือ 03H ก็ได้

3. **คำสั่งเลือกโหมดการป้อนข้อมูล** (Entry mode Set)

มีรายระเอียดของรูปแบบข้อมูลคำสั่งดังนี้

บิต7	บิต6	บิต5	บิต4	บิต3	บิต2	บิต1	บิต0
0	0	0	0	0	1	I/D	S

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

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

ดังนั้น ข้อมูลคำสั่งที่เกิดขึ้นสำหรับคำสั่งนี้ได้แก่ 04H-07H (4 ข้อมูลคำสั่ง) และที่ใช้บ่อย คือ 06H หมายถึงกำหนดให้เมื่อเกิดข้อมูลใหม่ เคอร์เซอร์จะเลื่อนไปทางขวามือ และแอดเดรสของ DDRAM เพิ่มขึ้น

4. **คำสั่งควบคุมการแสดงผล**

มีรายระเอียดของรูปแบบข้อมูลคำสั่งดังนี้

บิต7	บิต6	บิต5	บิต4	บิต3	บิต2	บิต1	บิต0
0	0	0	0	1	D	C	B

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

บิต C ใช้ควบคุมการแสดงตัวเคอร์เซอร์บนจอแสดงผล ถ้าต้องการให้มีตัวเคอร์เซอร์บนจอแสดงผล ต้องกำหนดให้บิตนี้เป็น “1” ถ้ากำหนดให้เป็น “0” จะเป็นการปิดเคอร์เซอร์ หรือไม่แสดงเคอร์เซอร์

บิต B ใช้ควบคุมการกระพริบของเคอร์เซอร์ ถ้าบิตนี้เป็น “1” เคอร์เซอร์จะกระพริบ

ดังนั้นข้อมูลคำสั่งนี้จะเป็นได้ตั้งแต่ 08H-0FH (8 รูปแบบคำสั่ง) ที่ใช้บ่อยคือ 0CH เป็นคำสั่งให้เปิดจอแสดงผล แต่ไม่แสดงเคอร์เซอร์ และ 0FH เป็นคำสั่งให้เปิดจอแสดงผล แสดงเคอร์เซอร์และคำสั่งให้เคอร์เซอร์กระพริบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5. **คำสั่งควบคุมการเลื่อนเคอร์เซอร์และข้อมูลตัวอักษร**

มีรายระเอียดของรูปแบบข้อมูลคำสั่งดังนี้

บิต7	บิต6	บิต5	บิต4	บิต3	บิต2	บิต1	บิต0
0	0	0	1	S/C	R/L	*	*

การควบคุมการเลื่อนเคอร์เซอร์และตัวอักษรบนจอแสดงผลขึ้นอยู่กับการกำหนดบิต S/C และ R/L ซึ่งสามารถสรุปได้ดังนี้

S/C	R/L	ลักษณะการเลื่อน	ข้อมูลคำสั่ง
0	0	เลื่อนเคอร์เซอร์ไปทางซ้าย	10H-12H
0	1	เลื่อนเคอร์เซอร์ไปทางขวา	14H-17H
1	0	เลื่อนตัวอักษรใหม่ไปทางซ้าย	18H-1BH
1	1	เลื่อนอักษรใหม่ไปทางขวา	1CH-1FH

6. **คำสั่งกำหนดฟังก์ชันการทำงาน**

มีรายระเอียดของรูปแบบข้อมูลคำสั่งดังนี้

บิต7	บิต6	บิต5	บิต4	บิต3	บิต2	บิต1	บิต0
0	0	1	D/L	N	F	*	*

บิต D/L ใช้กำหนดจำนวนบิตที่ใช้ติดต่อส่งผ่านข้อมูล ถ้าบิตนี้เป็น “0” จะเป็นการติดต่อแบบ 4 บิต แต่ถ้าเป็น “1” จะเป็นแบบ 8 บิต

บิต N ใช้กำหนดจำนวนบรรทัดของการแสดงผล ถ้าเป็น “0” จะแสดงผล 1 บรรทัด ถ้าเป็น “1” จะแสดงผล 2 บรรทัด ในกรณีที่จอแสดงผลสามารถแสดงได้มากกว่า 2 บรรทัด และต้องการให้แสดงผลมากกว่า 2 บรรทัด ก็กำหนดบิต N นี้ให้เป็น “1”

บิต F ใช้เลือกความละเอียดของตัวอักษรในการแสดงผล ถ้าบิตนี้เป็น “0” จะเป็นการแสดงผลแบบ 5X7 จุด และถ้าเป็น “1” จะแสดงผลเป็นแบบ 5X10 จุด

ข้อมูลคำสั่งที่ใช้บ่อยคือ 38H เป็นการกำหนดให้โมดูล LCD ทำงานในแบบ 8 บิต แสดงผล 2 บรรทัด และเลือกความระเอียดเป็นแบบ 5X7 จุด

7. **คำสั่งเลือกแอดเดรสของ** CGRAM

เมื่อต้องการกำหนดแอดเดรสของ CGRAM ต้องกำหนดให้บิต 7 เป็น “0” บิต 6 เป็น “1” ส่วนอีก 6 บิตที่เหลือจะแทนด้วยค่าแอดเดรสของ CGRAM จะต้องทำการกำหนดแอดเดรสด้วยคำสั่งนี้ก่อนที่จะอ่านหรือเขียนข้อมูลให้ CGRAM อยู่ระหว่าง 00H-3FH

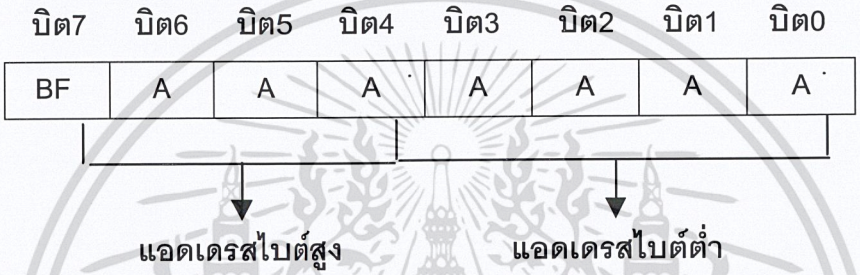
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

8. **คำสั่งเลือกแอดเดรสของ** DDRAM

ใช้ในการเลือกแอดเดรสของ DDRAM ก่อนที่จะทำการอ่านหรือเขียนข้อมูล โดยบิต 7 ต้องเป็น “1” และข้อมูลอีก 7 บิตที่เหลือจะเป็นค่าแอดเดรสของ DDRAM ซึ่งแอดเดรสของ DDRAM จะอยู่ระหว่าง 8CH-0FFH ทั้งนี้แอดเดรสยังขึ้นอยู่กับการกำหนดสถานะที่บิต N ด้วย หากบิต N เป็น “0” แอดเดรสของ DDRAM 80H-0CFH และถ้าบิต N เป็น “1” แอดเดรสของ DDRAM จะมี 2 ช่วงคือ 8CH-87H และ 0C0H-0C7H

9. **คำสั่งอ่านแฟลก** BUSY **และแอดเดรส**

มีรายระเอียดของรูปแบบข้อมูลคำสั่งดังนี้



เป็นคำสั่งที่ใช้อ่านแฟลก BUSY (BF) โดยแฟลกนี้เป็นตัวบอกสถานะของตัวควบคุม LCD ว่าพร้อมจะรับข้อมูลอยู่หรือไม่ ถ้าหากบิต BF เป็น “0” แสดงว่าตัวควบคุม LCD พร้อมรับข้อมูลหรือคำสั่ง แต่ถ้าเป็น “1” แสดงว่าขณะนี้ตัวควบคุม LCD ยังอยู่ในกระบวนการทำงานภายในหรือกำลังประมวลผลข้อมูลอยู่ ยังไม่พร้อมรับข้อมูลหรือคำสั่ง

เมื่อต้องการอ่านแฟลกต้องกำหนดให้ขา R/$\overline{W}$ เป็น “1” ด้วย แต่สัญญาณที่ RS ยังต้องเป็น “0” อยู่เพราะข้อมูลนี้เป็นข้อมูลคำสั่ง

นอกจากนี้ยังใช้เป็นคำสั่งอ่านข้อมูลแอดเดรสของ CGRAM และ DDRAM ด้วย โดย บิต 0 – บิต 6 เป็นค่าข้อมูลของแอดเดรสที่ต้องการอ่าน

2.9.5 การเขียนคำสั่งและข้อมูลให้แก่ส่วนแสดงผลแบบผลึกเหลว

ในการเขียนข้อมูลเพื่อควบคุมให้ส่วนแสดงผลแบบผลึกเหลวแสดงผลตามที่ต้องการ ต้องส่งคำสั่งแล้วกำหนดโหมดการทำงานให้แก่ส่วนแสดงผลแบบผลึกเหลวก่อน จากนั้นจึงค่อยส่งข้อมูลที่ต้องการ แสดงผลเนื่องจากบัสข้อมูลของส่วนแสดงผลแบบผลึกเหลวมี 8 เส้นคือ D0-D7 และใช้เป็นทางผ่านของทั้งคำสั่งและข้อมูล ดังนั้นในการส่งคำสั่งและข้อมูลจึงต้องอาศัยการกำหนดสัญญาณลอจิกที่ขา RS หากที่ขา RS ได้ลอจิก “0” หมายความว่า ข้อมูลที่ป้อนให้แก่ส่วนแสดงผลแบบผลึกเหลวขณะนั้นเป็นคำสั่ง ในทางตรงกันข้ามหากขา RS หากได้รับลอจิก “1” ข้อมูลที่ป้อนให้ขณะนั้นเป็นข้อมูลที่ใช้ในการแสดงผล

เอกสารนี้สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อต้องการเขียนหรืออ่านข้อมูลใน CGRAM และ DDRAM เริ่มต้น ต้องกำหนดแอดเดรสที่ต้องการอ่านหรือเขียนก่อน โดยใช้คำสั่งเลือกแอดเดรส จากนั้นกำหนดให้ขา RS เป็น “1” เพื่อแจ้งให้ตัวควบคุมภายในส่วนแสดงผลแบบผลึกเหลวทราบว่าข้อมูลที่ปรากฏต่อไปนี้เป็นข้อมูลปกติไม่ใช่คำสั่ง

ในกรณีที่ต้องการอ่านข้อมูลต้องกำหนดให้ขา R/$\overline{w}$ เป็น ”1” ข้อมูลขนาด 8 บิต (หรือ 4 บิต) ก็จะปรากฏบนบัสข้อมูล โดยข้อมูลที่อ่านออกมาได้จะเป็นข้อมูลจากแอดเดรสของ CGRAM หรือ DDRAM ตามที่ต้องการ

ในกรณีที่ต้องการเขียนข้อมูล เมื่อกำหนดแอดเดรสและป้อนกำหนดให้ขา R/$\overline{w}$ เป็น “0” ข้อมูลที่อยู่บนบัสข้อมูลจะถูกเขียนลงในรีจิสเตอร์ DR จากนั้นจึงถ่ายทอดลงใน DDRAM ต่อไป

2.9.6 จังหวะการทำงานของส่วนแสดงผลแบบผลึกเหลว

ในการติดต่อกับส่วนแสดงผลแบบผลึกเหลว จะต้องมีการหน่วงอวลาหลังจากที่ทำการส่งรหัสคำสั่งหรือข้อมูลเนื่องจากต้องรอให้คอนโทรลเลอร์ภายในส่วนแสดงผลแบบผลึกเหลว แปลความหมายของรหัสคำสั่งและทำงานตามคำสั่งให้เรียบร้อยก่อน จากนั้นจึงจะรับข้อมูลหรือดำเนินการต่อไป

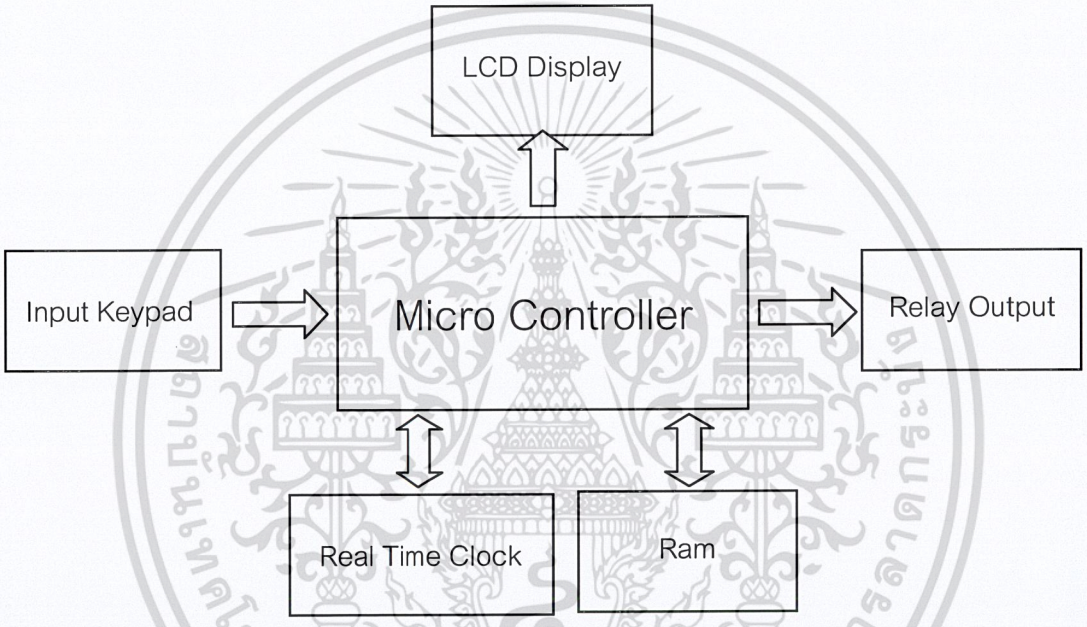
ดังนั้นในการใช้งานส่วนแสดงผลแบบผลึกเหลว ต้องมีการหน่วงเวลารอให้ส่วนแสดงผลแบบผลึกเหลวพร้อมทำงาน จากนั้นก็จะกำหนดลอจิกให้แก่ขา RS ของส่วนแสดงผลแบบผลึกเหลวแล้วควรหน่วงแวลาไว้อีกซักพักเพื่อให้คอนโทรลเลอร์ภายในส่วนแสดงผลแบบผลึกเหลวแปลความหมายของลอจิกที่ RS ว่าข้อมูลต่อไปที่จะได้รับนั้นเป็นรหัสคำสั่งหรือเป็นข้อมูลที่ต้องการแสดงผล จากนั้นจะเป็นการส่งข้อมูลมารอที่บัสข้อมูล D0-D7 ขั้นต่อไปจะเป็นการส่งสัญญาณพัลส์ไปที่ขา E เพื่ออีนาเบิลส่วนแสดงผลแบบผลึกเหลว ให้รับข้อมูลจากบัสข้อมูลเข้าไป โดยพัลส์ที่ป้อนเข้าที่ขา E ของส่วนแสดงผลแบบผลึกเหลวต้องเป็นพัลส์ขาขึ้น จากนั้นให้หน่วงเวลาไว้สักครู่

ทั้งหมดที่กล่าวมาคือขั้นตอนและจังหวะในการทำงาน 1 รอบของโมดูล LCD จะเห็นได้ว่ามีโปรแกรมย่อยที่สำคัญอยู่ 3 โปรแกรมย่อยคือ โปรแกรมอินนิเชียล LCD , โปรแกรมหน่วงเวลาและโปรแกรมย่อยการส่งพัลส์เพื่ออีนาเบิลโมดูล LCD

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3
การคำนวณและการสร้าง

3.1 แนวคิดและหลักการทำงาน



รูปที่ 3.1 บล็อกไดอะแกรมของเครื่องควบคุมอุปกรณ์ไฟฟ้าแบบโปรแกรมได้

จากรูปที่ 3.1 แสดงบล็อกไดอะแกรมของเครื่องควบคุมอุปกรณ์ไฟฟ้าภายในบ้านแบบโปรแกรมได้ ซึ่งจะเห็นได้ว่ามีไมโครคอนโทรลเลอร์เป็นหน่วยประมวลผลกลาง (CPU) ซึ่งจะทำหน้าที่ติดต่อกับอุปกรณ์ต่างๆ ที่นำมาต่อร่วม คือ คีย์แพด (Key Pad), ส่วนแสดงผล (LCD), วงจรฐานเวลาอ้างอิง (Real Time Clock), หน่วยความจำข้อมูลภายนอก (RAM), เอาท์พุต (Relay Output) ซึ่งในการออกแบบและหลักการทำงานของวงจรต่างๆสามารถอธิบายได้ดังนี้

ในระบบจะมีวงจรฐานเวลาอ้างอิง (Real Time Clock : RTC) เป็นตัวสร้างเวลาจริงให้กับระบบซึ่งเป็นหัวใจหลักของระบบ โดยที่วงจรนี้จะทำการผลิตสัญญาณนาฬิกาไปประมวลผลที่ไม-โครคอนโทรลเลอร์ เพื่อไปอ้างอิงในการเปิด-ปิด (ON-OFF) เอาท์พุตที่เป็นอุปกรณ์ไฟฟ้าภายในบ้าน ในส่วนของวงจรนี้จะต้องทำงานตลอดเวลาเพื่อให้สามารถสร้างเวลาที่แท้จริงตลอดเวลา ทำให้การอ้างอิงเพื่อไปเปิด-ปิด (ON-OFF) สามารถทำได้อย่างถูกต้องตามเวลาที่แท้จริง ในส่วนของอินพุต (Key Pad)

โดยจะเป็นคีย์แพดแบบ 4X4 จะเป็นส่วนที่ป้อนค่าเข้าไปตามที่ต้องการ โดยที่เมื่อทำการป้อนค่าเข้าไปจะไปทำการกำหนดค่าต่างๆ แล้วดึงเอาโปรแกรมต่างๆที่ถูกเขียนอยู่ในไมโครคอนโทรลเลอร์มาใช้งาน กล่าวคือ การกำหนดให้เอาท์พุต เปิด-ปิด จะต้องกำหนดที่คีย์แพด ในส่วนของหน่วยความจำข้อมูลภายนอก (RAM) จะใช้ในการเก็บข้อมูลเวลาที่ได้จากการป้อนค่าผ่านทาง Key Pad และข้อมูลที่ได้นี้จะถูกนำไปเปรียบเทียบกับฐานเวลาอ้างอิง ซึ่งใช้ในการเปิด-ปิด (ON-OFF) เอาท์พุตที่เป็นอุปกรณ์ไฟฟ้าภายในบ้าน และในส่วนของส่วนแสดงผล (LCD) ก็เป็นเพียงการแสดงค่าต่างๆ โดยหลักๆแล้วจะทำการแสดงเวลา, วัน, วันที่, เดือน และปีอยู่ตลอดเวลาและแสดงค่าการกำหนดเอาท์พุตต่างๆนั่นเอง และในบล็อกไดอะแกรมสุดท้ายคือ เอาท์พุต (Relay Output) เป็นวงจรรีเลย์ที่ทำหน้าที่เป็นสวิตช์เปิด-ปิด (ON-OFF) อุปกรณ์ไฟฟ้า โดยอาศัยหลักการกำหนดค่าจากคีย์แพดแล้วสัญญาณจะถูกประมวลผลที่ไมโครคอนโทรลเลอร์ซึ่งสัญญาณที่ถูกประมวลผลจะถูกส่งมาควบคุมให้รีเลย์ทำการเปิด-ปิดได้ตามเวลาที่ต้องการ โดยอาศัยโปร-แกรมในไมโครคอนโทรลเลอร์เป็นตัวอ้างอิง

จากหลักการทั้งหมดจะเห็นได้ว่า แนวคิดในการสร้างเครื่องควบคุมอุปกรณ์ไฟฟ้าภายในบ้านแบบโปรแกรมได้นี้จะใช้เปิด-ปิด (ON-OFF) ให้ได้ตามเวลาที่ต้องการโดยให้ระบบมีฐานเวลาอ้างอิงเท่านั้น ซึ่งในการสร้างจะเป็นการเขียนโปรแกรมรูปแบบต่างๆ เพื่อเป็นการกำหนดให้เอาท์พุตทำงานได้ตามเวลาที่กำหนด ซึ่งในแต่ละเอาท์พุตนั้นจะสามารถกำหนดให้ทำงานได้ในรอบสัปดาห์หรือรอบวัน โดยแต่ละเอาท์พุทสามารถตั้งโปรแกรมให้ทำงานได้ 30 โปรแกรมต่อวัน (หรือ 210 โปรแกรมต่อสัปดาห์) นอกจากนี้ยังสามารถกำหนดโหมดการทำงานได้ทั้งสิ้น 2 แบบคือ แบบ Program Mode ซึ่งเป็นโหมดการทำงานที่กำหนดให้อุปกรณ์ไฟฟ้าสามารถเปิด-ปิดได้ตามเวลาที่กำหนดไว้ และแบบ Manual Mode ซึ่งเป็นโหมดการทำงานที่สามารถเปิด-ปิดอุปกรณ์ไฟฟ้าด้วยมือทุกตัวตามที่ต้องการโดยไม่ขึ้นกับโปรแกรมที่ตั้งไว้

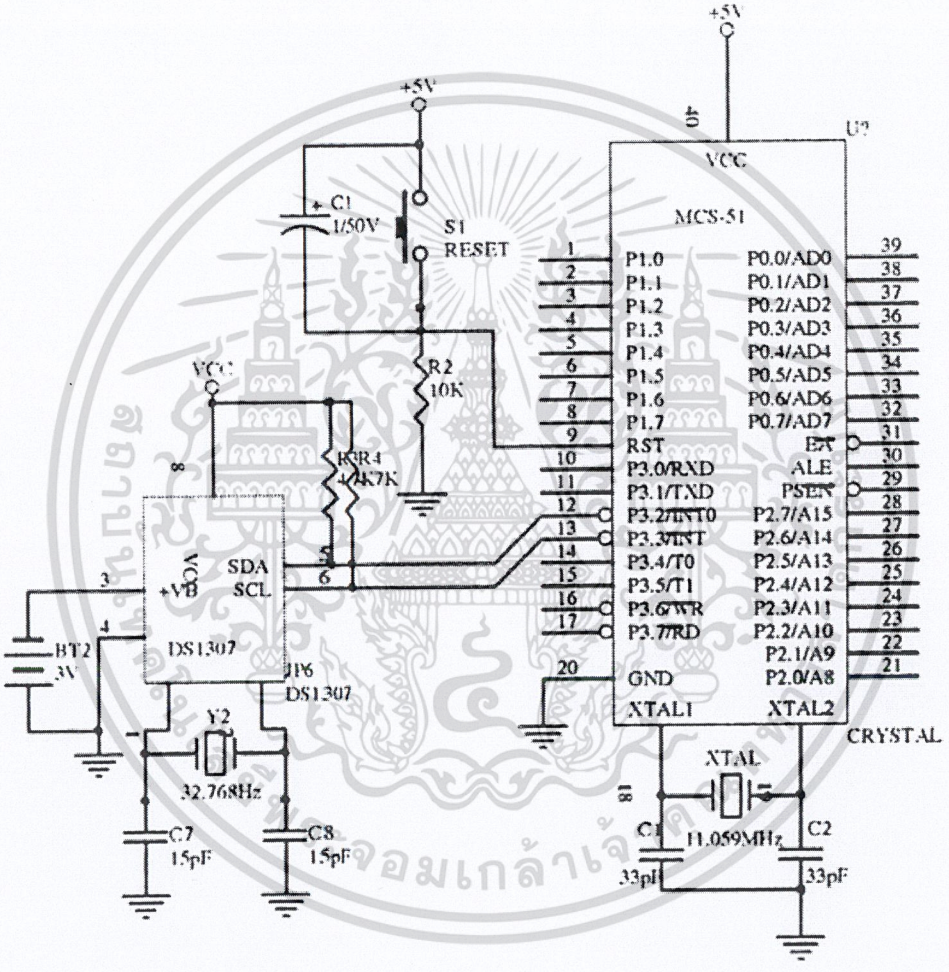
3.2 **วงจรฐานเวลาอ้างอิง** (Real Time Clock : RTC)

ในรูปที่ 3.2 แสดงวงจรการต่อใช้งานไอซีสร้างฐานเวลาจริง (Real Time Clock : RTC) เบอร์ DS1307 ซึ่งจะต้องมี cystral ขนาด 32.768 KHz และตัวเก็บประจุค่าต่ำๆประมาณ 15 pF ต่ออยู่ที่ขา X1 และ X2 เพื่อใช้เป็นตัวกำเนิดสัญญาณนาฬิกาในการสร้างฐานเวลาจริง สำหรับการต่อร่วมกับไมโครคอนโทรลเลอร์มีลักษณะการต่อใช้งานในลักษณะอุปกรณ์ระบบบัส I^2C (ดูรายระเอียดอุปกรณ์ระบบบัส I^2C ได้ในหัวข้อที่ 2.2) ซึ่งรายระเอียดการต่อใช้งานนั้นมีการต่อขา SDA และขา SCL เพื่อทำการติดต่อกับไมโครคอนโทรลเลอร์เพียง 2 เส้นเท่านั้น ซึ่งเป็นความสามารถพิเศษของอุปกรณ์ระบบบัส I^2C และจะต้องต่อความต้านทานพูลอัพที่ 2 ขานี้ด้วยเพื่อคงสภาวะเริ่มต้นของ DS1307 ให้เป็น high

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยหากไม่มีการติดต่อระหว่าง DS1307 กับไมโครคอนโทรล –เลอร์ ที่พอร์ต 3.2 และ 3.3 จะเป็น high แต่ถ้ามีการติดต่อระหว่าง DS1307 กับ ไมโครคอนโทรล - เลอร์ ที่พอร์ต 3.2 และ 3.3 จะเป็น low

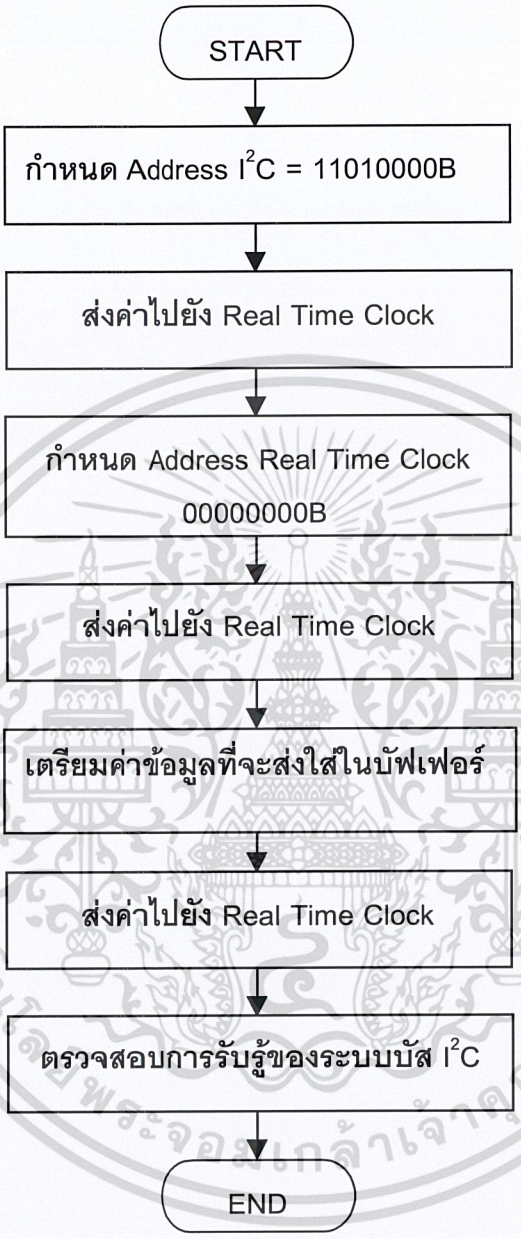
จากวงจรมีการต่อแบตเตอรี่ที่ขา V_{BAT} ไว้ตลอดเวลาไม่ว่าจะใช้งานหรือไม่ ทั้งนี้เพื่อรักษาการทำงานของวงจรภายใน DS1307 ไว้ให้ยังคงทำงานต่อเนื่องไปได้ เมื่อใดที่ไมโครคอนโทรล เลอร์ ต้องการอ่านค่าข้อมูลก็จะได้ข้อมูลที่เป็นเวลาจริงอยู่ตลอดเวลา



รูปที่ 3.2 วงจรการต่อใช้งานไอซีสร้างฐานเวลาจริง (Real Time Clock : RTC)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.1 การเขียนข้อมูลลงไอซีสร้างฐานเวลาจริง (Real Time Clock) DS1307

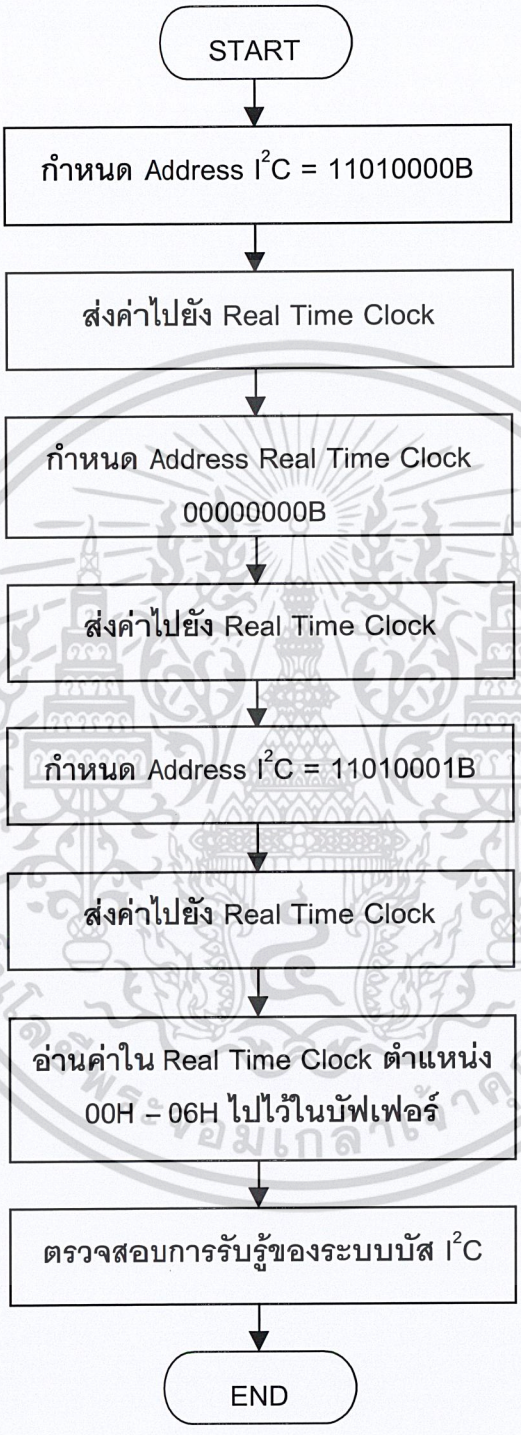


รูปที่ 3.3 Flowchart การเขียนข้อมูลลง Real Time Clock

การทำงานของ Flowchart ดังรูปที่ 3.3 เป็นการติดต่อกับไอซีสร้างฐานเวลาจริง (Real Time Clock) ในโหมดการเขียนค่า ต้องกำหนด Address ในการเขียน (Address = 11010000B) จากนั้นส่งค่าไปให้กับไอซีสร้างฐานเวลาจริง เมื่อส่งค่าข้อมูลเสร็จสิ้น ก็กำหนดแอดเดรส Real Time Clock ที่ต้องการจะเขียนข้อมูลลงไป ในที่นี้เราต้องการเขียนข้อมูลลง ณ ตำแหน่ง 00H – 06H ดังนั้นเราจึงนำค่าข้อมูลในบัฟเฟอร์เขียนลงในตำแหน่ง 00H – 06H โดยการเขียนข้อมูลลงแต่ละครั้งจะทำการตรวจสอบการรับรู้ของระบบบัส I2C ว่าพร้อมทำงานต่อไปหรือยัง

เอกสารนี้สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.2 การอ่านค่าจากไอซีสร้างฐานเวลาจริง (Real Time Clock) DS1307



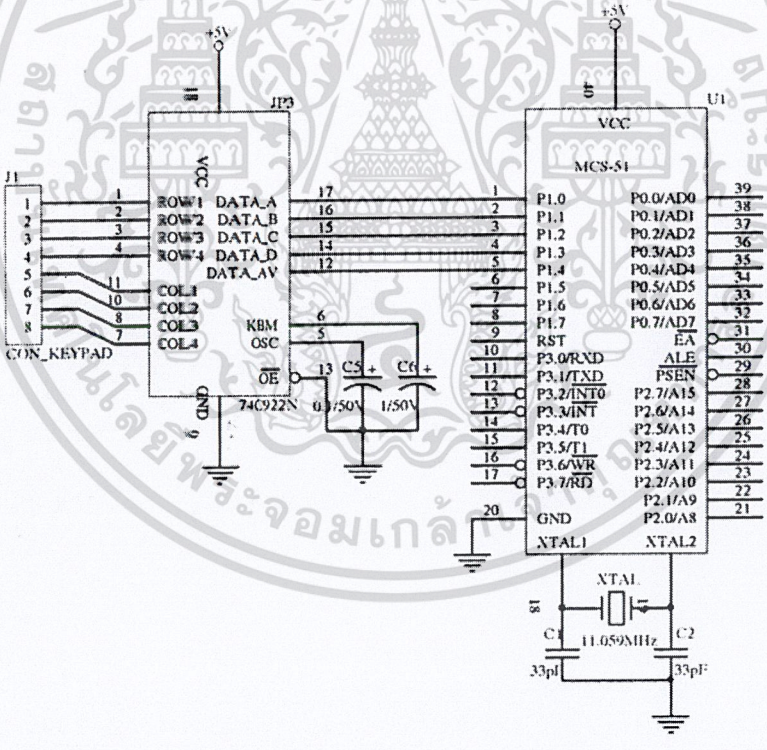
รูปที่ 3.4 Flowchart การอ่านข้อมูลจากไอซี Real Time Clock

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทำงานของ Flowchart ดังรูปที่ 3.4 เป็นการติดต่อกับไอซีสร้างฐานเวลาจริง ในโหมดการอ่านค่าต้องกำหนดแอดเดรสในการเขียน (Address = 11010000B) จากนั้นส่งค่าไปให้ไอซีสร้างฐานเวลาจริง ซึ่งการส่งค่าจะต้องส่งค่า start bit ตามด้วยข้อมูลจากนั้นต้องส่ง stop bit เมื่อการส่งค่าข้อมูลเสร็จสิ้น จากนั้นทำการกำหนดแอดเดรสในไอซีสร้างฐานเวลาจริง ที่จะอ่านค่าออกมาซึ่งในที่นี้เราต้องการอ่านค่าตั้งแต่ตำแหน่ง 00H – 06H ซึ่งเป็นตำแหน่งที่เก็บค่า วินาที, นาที, ชั่วโมง, วัน, วันที่, เดือน, ปี ตามลำดับ

การอ่านค่าข้อมูลแต่ละครั้งจะนำมาเก็บในบัฟเฟอร์เพื่อนำไปใช้ต่อไป และการอ่านค่าแต่ละครั้งก็จะทำการตรวจสอบการรับรู้ของระบบบัส I^2C ด้วยว่าพร้อมทำงานหรือยัง การที่จะอ่านค่าต้องกำหนดแอดเดรสในการอ่านก่อน

3.3 **วงจร** Scan Keyboard



รูปที่ 3.5 วงจรการเชื่อมต่อ Key Pad โดยใช้ IC Scankey 74C922

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คีย์แพดแบบเมตริกซ์ 4x4 เป็นเทคนิคการนำสวิตช์จำนวน 16 ตัวมาต่อกันแบบแนวแกนตั้งและแนวแกนนอน จะเรียกแนวแกนตั้งว่า คอลัมน์ (Column) และแนวแกนนอนว่าโรว์ (Row) ซึ่งเมื่อต่อแล้วจะได้สายสัญญาณทั้งหมด 8 เส้น แบ่งเป็นแนวแกนตั้ง 4 เส้น และแนวแกนนอน 4 เส้น จะเห็นว่าถ้าต่อสวิตช์โดยตรงจะต้องมีสายสัญญาณถึง 16 เส้น ดังนั้น การใช้คีย์แพด 4x4 นี้ มีข้อดีคือ ช่วยลดสายสัญญาณที่จะเชื่อมต่อกับไมโครคอนโทรลเลอร์ให้น้อยลงได้ทำให้สามารถนำพอร์ตที่เหลือของไมโครคอนโทรลเลอร์ไปใช้งานอย่างอื่นได้

ในรูปที่ 3.5 แสดงการเชื่อมต่อคีย์แพด 4x4 กับไมโครคอนโทรลเลอร์โดยจะมี IC Scankey เบอร์ 74C922 ซึ่งเป็น 16 key encoder เป็นตัวเชื่อมต่อระหว่างคีย์แพด 4x4 กับไมโครคอนโทรล - เลอร์ โดยที่เอาท์พุตของไอซีนี้จะมี 4 เส้น เพื่อต่อกับไมโครคอนโทรลเลอร์ดังรูป ข้อดีของ IC Scan – key นี้คือ จะช่วยลดความยุ่งยากในการเขียนโปรแกรมทำให้โปรแกรมมีขนาดสั้นลงไม่เปลือง Flash Memory ของไมโครคอนโทรลเลอร์ ทั้งนี้เพราะว่าเราไม่จำเป็นต้องไปสร้าง Table เพื่อทำการจดจำค่าคีย์ของแต่ละคีย์นั่นเอง จากวงจรในรูปที่ 2.5 ถ้าทำการกดคีย์ต่าง ๆ ที่คีย์แพดจะได้ค่าเอาท์พุต A-D ของ IC Scankey ดังในตารางที่ 3.1

Key	1	2	3	↑	4	5	6	↓	7	8	9	NOP	BACK	0	HELP	ENTER
A	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
B	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
C	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
D	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1

ตารางที่ 3.1 แสดงค่าเอาท์พุตที่ได้จาก IC Scankey 74C922

จากค่าเอาท์พุตของ IC Scankey ในตารางที่ 3.1 จะเป็นตัวกำหนดว่า คีย์ใดถูกกดเช่น ถ้าทำการกดเลข 1 จะได้ลอจิกเป็น 0000 ดังนั้นเมื่อลอจิก 0000 ถูกส่งเข้าไมโครคอนโทรลเลอร์ ไม-โครคอนโทรลเลอร์ก็จะทำการประมวลผลว่าคีย์หมายเลข 1 ถูกกด จากนั้นตัวไมโครคอนโทรลเลอร์ก็จะทำงานตามที่โปรแกรมไว้

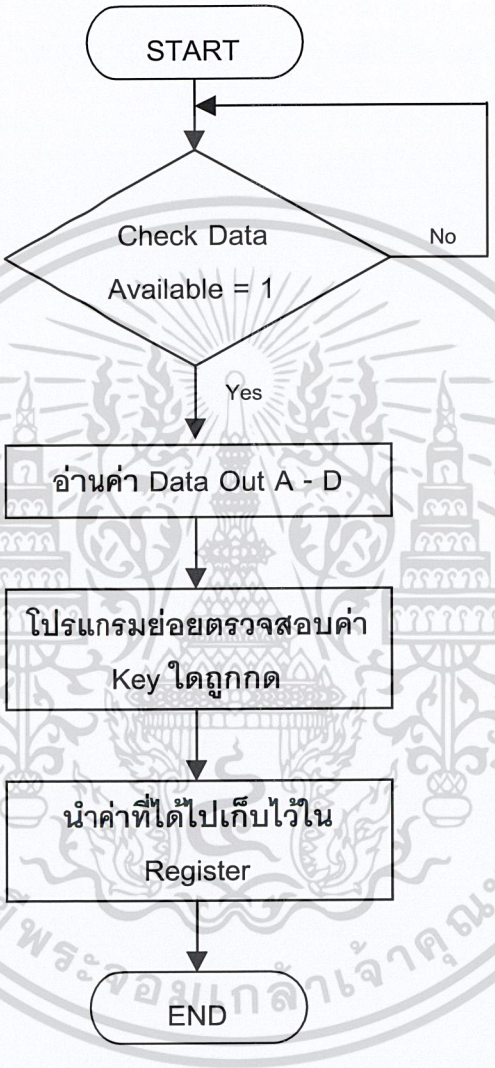
จากวงจรในรูปที่ 3.5 มีวิธีการกำหนดค่าตัวเก็บประจุที่ต่ออยู่ที่ขา OSC และ KEY B กล่าวคือ ตัวเก็บประจุที่ขา OSC เป็นตัวกำหนดค่าความถี่ในการสแกนคีย์ (F_{scan} (Hz)) และตัวเก็บประจุที่ขา KEY B จะเป็นตัวกำหนดค่าเวลาในการกดว่าให้ระยะเวลาในการกดคีย์แรกและคีย์ถัดไปมีระยะเวลาห่างกันเท่าใด มีหน่วยเป็น second โดยตัวเก็บประจุที่ขา KEY B จะต้องเป็น 10 เท่าของตัวเก็บประจุที่ขา OSC จากรูปที่เลือกใช้ค่าตัวเก็บประจุดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

C_{osc} = 0.1 uF ⇨ F_{scan} = 600 Hz

C_{KBM} = 1 uF ⇨ Debounce Period = 0.01 sec

(รายละเอียดการเลือกค่าตัวเก็บประจุดูได้จากภาคผนวก)

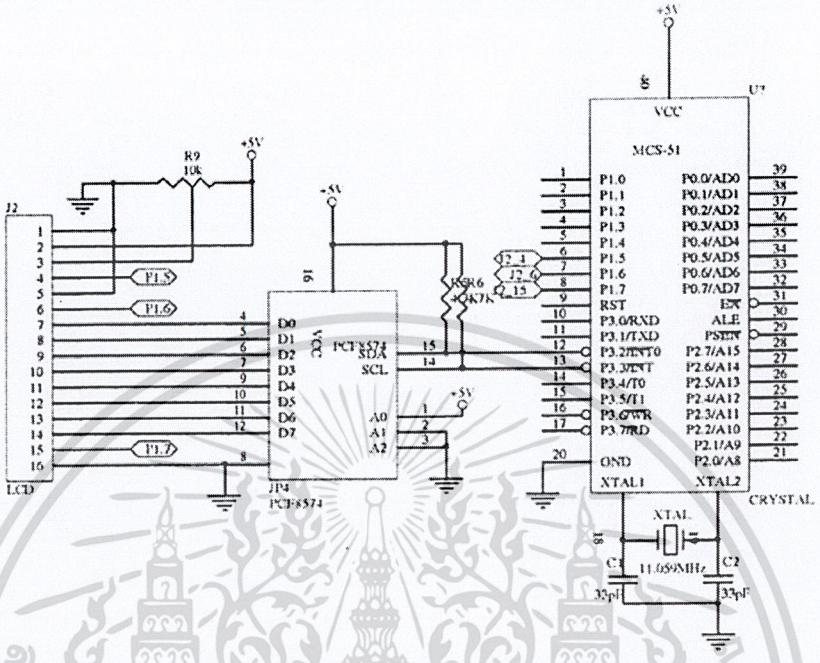


รูปที่ 3.6 Flowchart การติดต่อกับ IC Scankey 16 Key Encoder

การทำงานของ Flowchart ดังรูปที่ 3.6 เป็นการติดต่อกับ IC Scankey 16 Key Encoder โปรแกรมจะทำการตรวจสอบค่า Key Data ว่ามีค่าเป็น “1” หรือไม่ ถ้าเป็น “1” แสดงว่ามีการกดคีย์ใด ๆ เกิดขึ้น แต่ถ้า Key Data มีค่าเป็น “0” แสดงว่าไม่มีการกดคีย์ใด ๆ เมื่อมีการกดค่าคีย์ใด ๆ ก็จะอ่านค่าจาก Data Out A-D เข้ามาเก็บไว้ เพื่อนำไปตรวจสอบว่าได้ค่าคีย์ที่กดมีค่าเท่าใด เพื่อนำไปใช้ต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4 วงจร LCD



รูปที่ 3.7 วงจรการเชื่อมต่อ LCD กับไมโครคอนโทรลเลอร์

รูปที่ 3.7 วงจรการเชื่อมต่อ LCD ขนาด 20 ตัวอักษร 4 บรรทัด แบบ 8 บิต (รายละเอียด LCD ขนาด 20x4 ดูในหัวข้อ 2.9) ในการเลือกการต่อร่วมกับไมโครคอนโทรลเลอร์มีให้เลือกทั้งแบบ 8 บิต และ 4 บิตซึ่งจะมีข้อดีข้อเสียต่างกันคือ

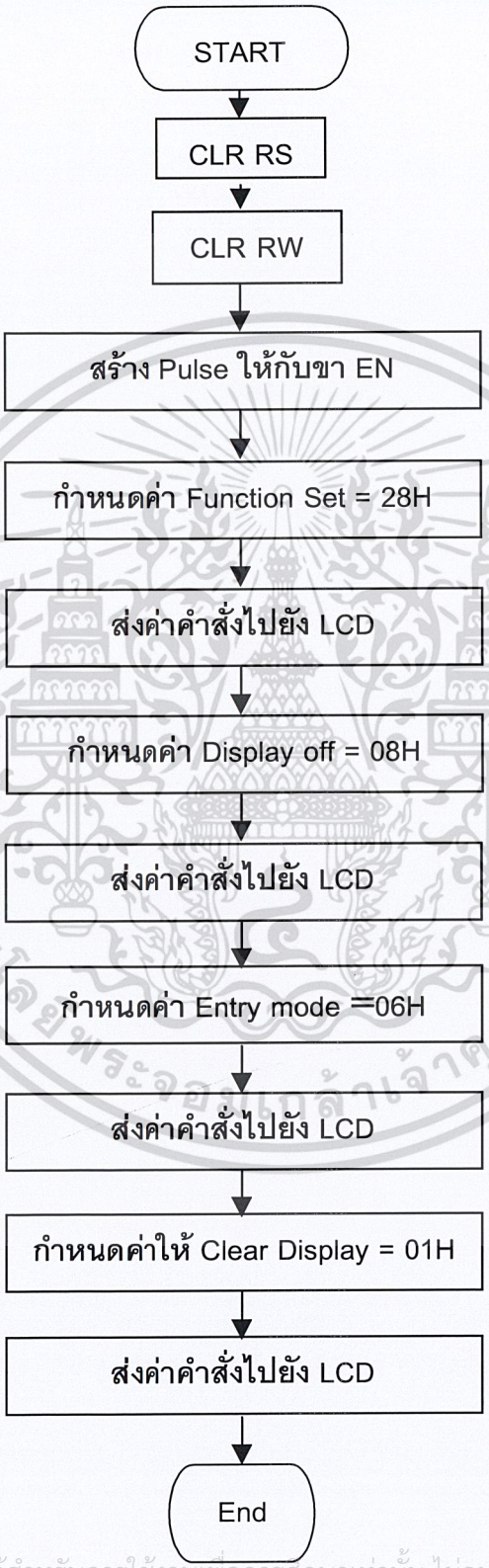
1. ความเร็วในการแสดงผลแบบ 8 บิตจะเร็วกว่าแบบ 4 บิต
2. การใช้สายสัญญาณติดต่อกับไมโครคอนโทรลเลอร์แบบ 4 บิต จะใช้น้อยกว่าแบบ 8บิต

จากข้อดีข้อเสียดังกล่าวถึงแม้ว้าแบบ 4 บิตจะมีอัตราการแสดงผลที่ช้ากว่าแบบ 8 บิต แต่ข้อดีที่ใช้สายสัญญาณน้อยกว่าเป็นข้อได้เปรียบที่มีประโยชน์มากเพราะการเหลือพอร์ตไว้มาก ๆ ของไมโครคอนโทรลเลอร์จะช่วยให้สามารถทำการต่ออุปกรณ์อื่น ๆ เพิ่มขึ้นมาได้อีก เป็นการขยายประสิทธิภาพของระบบให้มีการใช้งานได้หลากหลายขึ้นด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4.1 การติดต่อกับ LCD

- การกำหนดค่าเริ่มต้นให้กับ LCD



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 3.8 Flowchart การกำหนดค่าเริ่มต้นให้กับ LCD

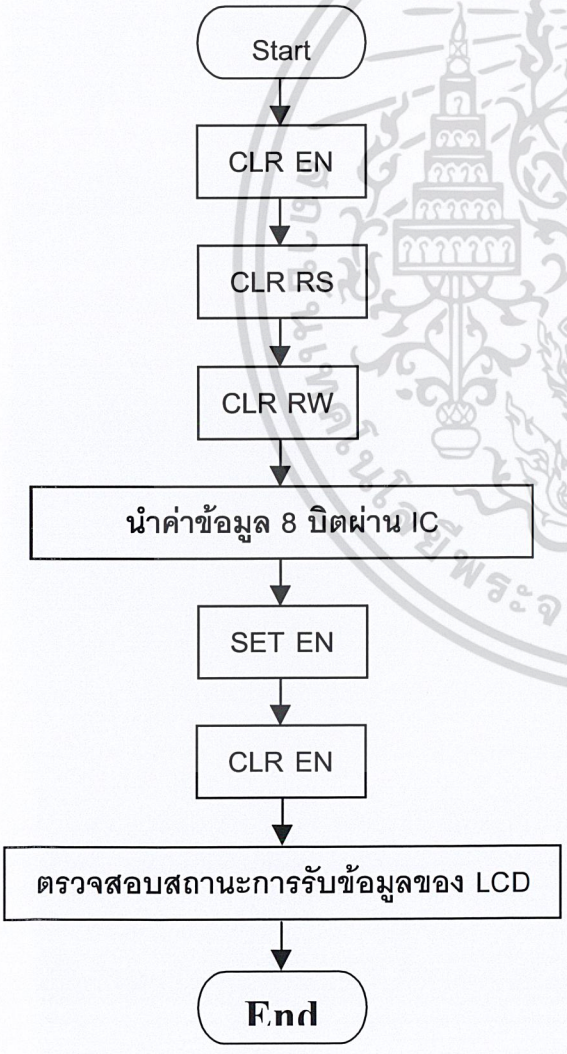
การทำงานของ Flowchart ดังรูปที่ 3.8 เราจะต้องกำหนดค่าต่าง ๆ ก่อนเพื่อให้ LCD พร้อมที่จะทำงานโดยการเขียนกำหนดดังนี้

RS = 0 เพื่อบอกว่าข้อมูลที่จะส่งเป็นคำสั่ง

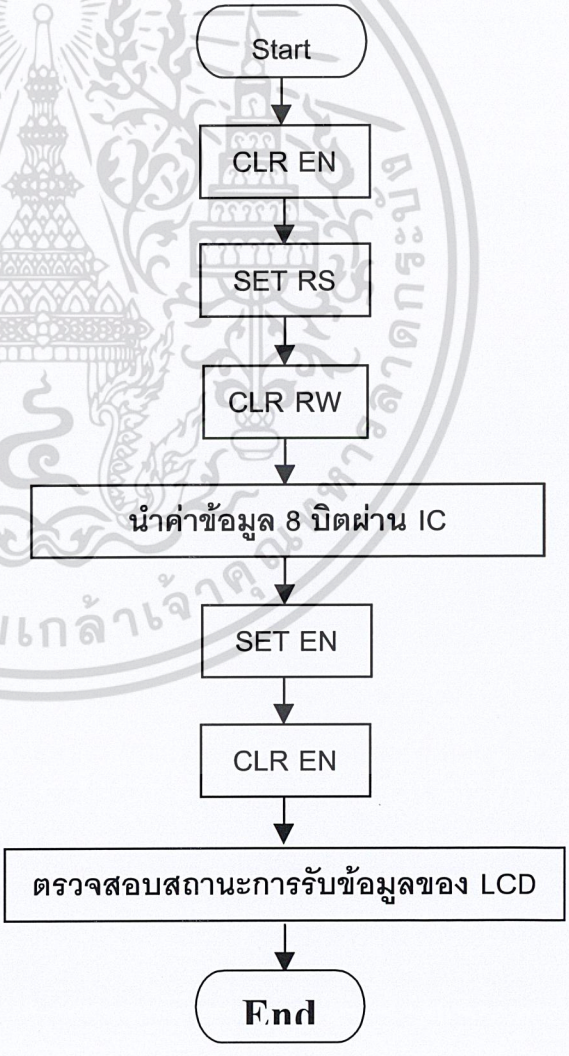
RW = 0 เพื่อบอกว่าเป็นการเขียนข้อมูล

จากนั้นสร้างสัญญาณพัลส์ให้กับขา EN เพื่อที่จะนำค่าสั่งต่าง ๆ ส่งให้กับส่วนควบคุมของ LCD ซึ่งคำสั่งควบคุมที่ส่งก็มี function set, Display off, Entry mode, Clear display

- การส่งคำสั่งและการส่งค่าข้อมูลแบบ 8 บิต



รูปที่ 3.9 Flowchart การส่งค่าคำสั่งแบบ 8 บิต



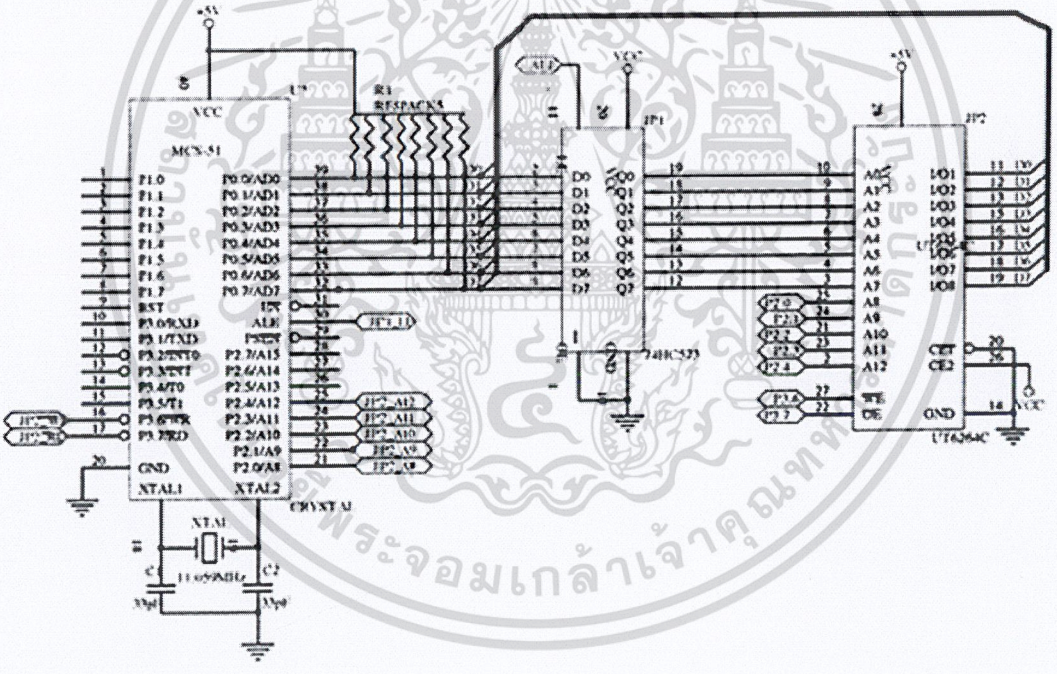
รูปที่ 3.10 Flowchart การส่งค่าข้อมูลแบบ 8 บิต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทำงานของ Flowchart ดังรูปที่ 3.9 เป็นการส่งค่าคำสั่งแบบ 4 บิต จะต้องทำให้ขา EN = 0 และขา RS = 0 เพื่อเป็นการบอกว่าข้อมูลที่จะส่งไปให้ส่วนควบคุมของ LCD เข้าใจว่าเป็นคำสั่งจากนั้นให้ขา RW = 0 เพื่อเขียนค่าลงในส่วนของ LCD โดยการส่งข้อมูลจะส่งแบบ 4 บิต ซึ่งจะส่ง 4 บิต แรกก่อนจากนั้นค่อยส่งข้อมูล 4 บิตล่างตามมา จากนั้นจะทำการตรวจสถานะการรับรู้ข้อมูลเมื่อได้รับข้อมูลแล้วก็จะสามารถรับข้อมูลใหม่ได้

การทำงานของ Flowchart ดังรูปที่ 3.10 เป็นการส่งค่าข้อมูลแบบ 4 บิตจะมีลักษณะเหมือนกับการส่งค่าคำสั่งแต่จะต่างกันตรงที่ขา RS จะต้องทำการกำหนดให้เป็น 1 เพื่อให้ตัวควบคุมของ LCD ได้เข้าใจว่าเป็นข้อมูล

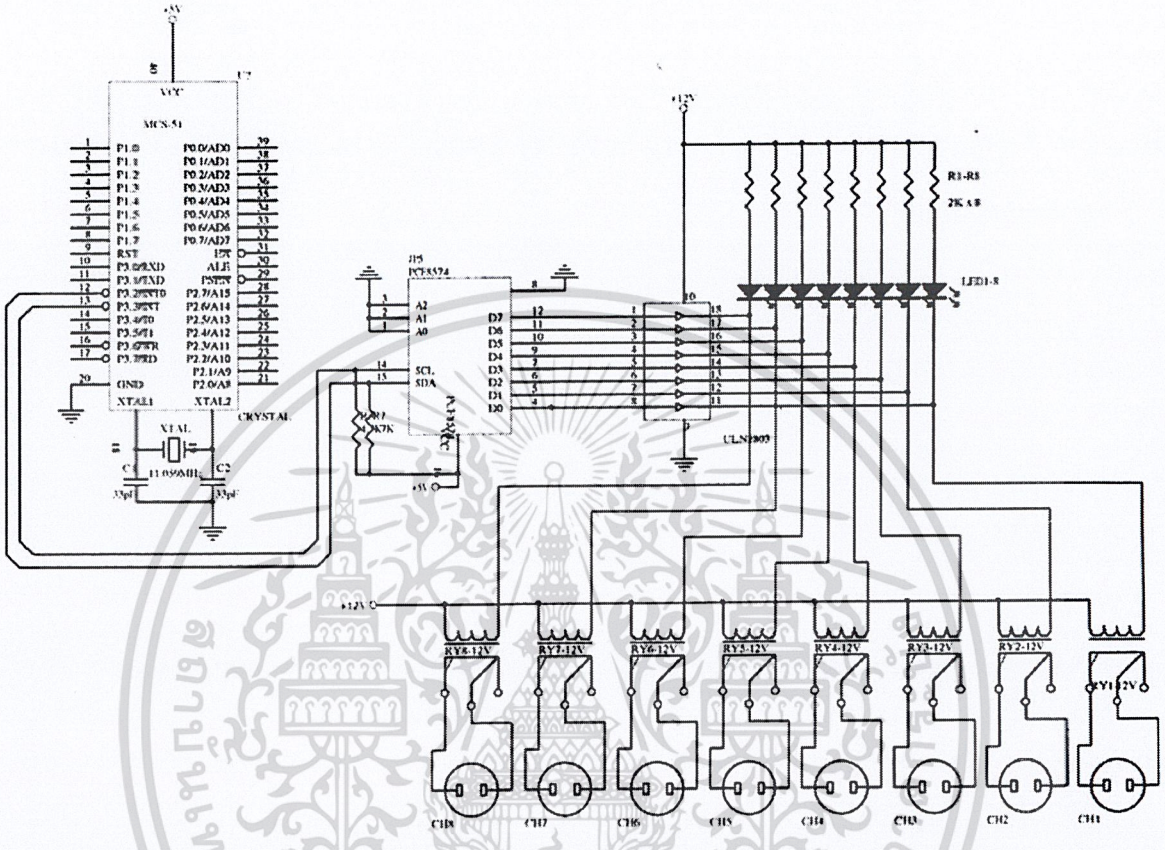
3.5 วงจรหน่วยความจำข้อมูลภายนอก



รูปที่ 3.11 วงจรการเชื่อมต่อหน่วยความจำข้อมูลภายนอกของไมโครคอนโทรลเลอร์

จากรูปที่ 3.11 แสดงการต่อหน่วยความจำข้อมูลภายนอกขนาด 8 K (2^{13}) จะเห็นได้ว่าขาพอร์ต P0.0 – P0.7 ถูกใช้เป็นขาข้อมูล D0 – D7 และขาแอดเดรสไบต์ต่ำ (A0 – A7) โดยผ่านวงจรแลตช์ ซึ่งใช้ไอซีเบอร์ 74HC573 และใช้สัญญาณ ALE ในการเลือกค่าว่า ต้องการใช้งานขา P0.0 – P0.7 เพื่อเป็นขาข้อมูลหรือขาแอดเดรส ในขณะที่ P2.0 – P2.4 ใช้ในการเชื่อมต่อกับขาแอดเดรสไบต์สูง A8 – A12 (เนื่องจาก RAM มีขนาด 8 K จึงใช้สายแอดเดรสทั้งหมด 13เส้น คือ A0 – A12) และใช้สัญญาณสำหรับการอ่านและเขียนหน่วยความจำข้อมูลภายนอก นั่นคือ ขา $\overline{RD}$ และ $\overline{WR}$

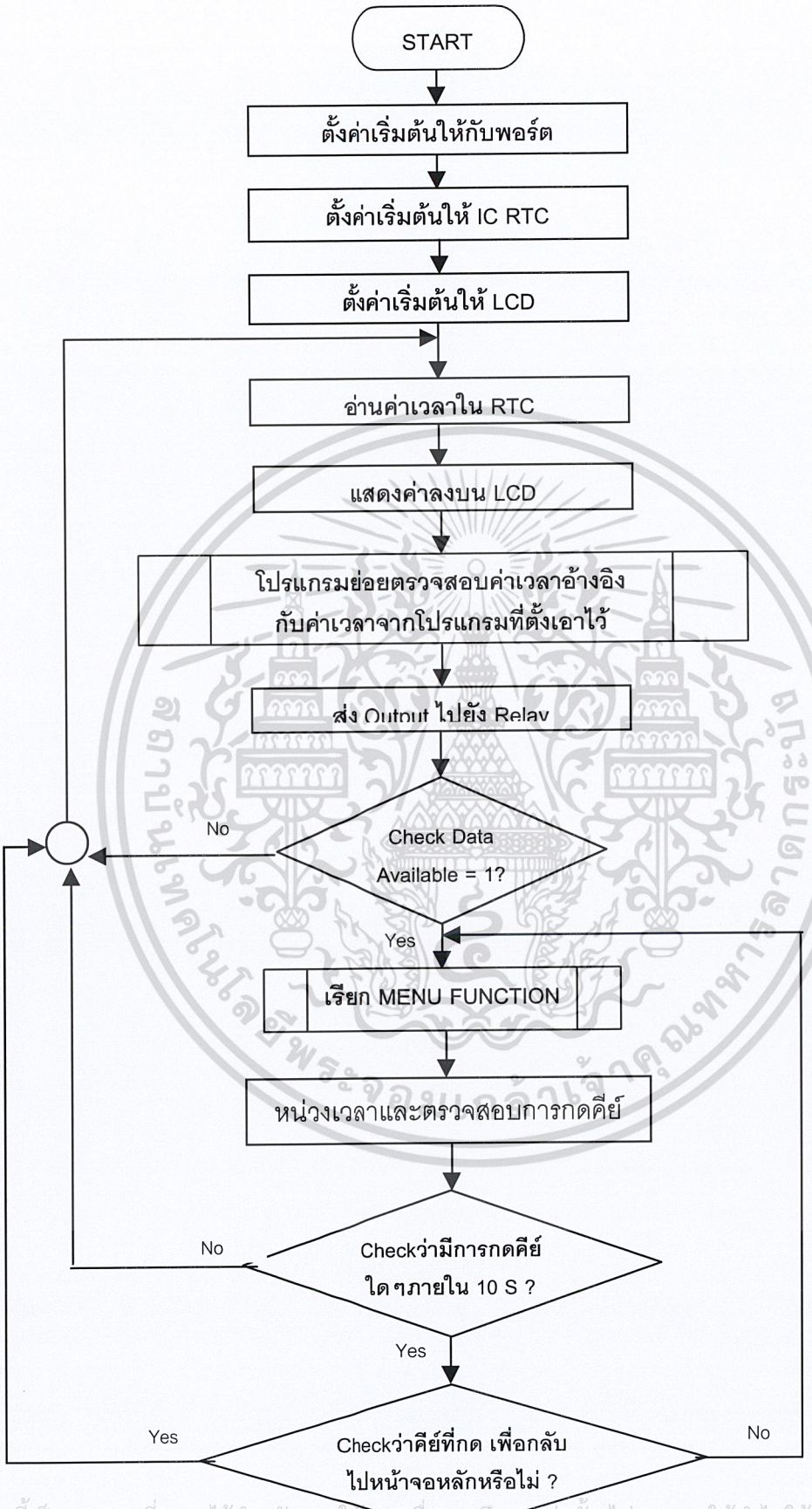
3.6 วงจร Relay Output



รูปที่ 3.12 วงจรการต่อใช้งาน Relay ร่วมกับไมโครคอนโทรลเลอร์

ในรูปที่ 3.12 แสดงการต่อใช้งานรีเลย์เป็นสวิตช์เปิด-ปิด แรงดันไฟ AC 220 V ซึ่งมีการทำงาน คือ สัญญาณที่ถูกประมวลผลจากไมโครคอนโทรลเลอร์เพื่อไปควบคุมการเปิด-ปิดรีเลย์จะออกจากพอร์ต P3.2 และ P3.3 ไปยังขา SDA และ SCL ของไอซีขยายพอร์ตอินพุตเอาท์พุตเบอร์ PCF8574 จากนั้นสัญญาณทั้ง 8 เส้นจะถูกส่งเข้าไปที่อินพุตของ ULN2803A ซึ่งเป็นไอซีไดรเวอร์แบบอินเวอร์เตอร์ที่สามารถจ่ายกระแสได้สูงสุด 500 mA และมีไดโอดป้องกันแรงดันย้อนกลับจากการยุบตัวของสนามแม่เหล็กในขดลวดรีเลย์ อยู่ภายใน และสามารถนำสัญญาณเอาท์พุตทั้ง 8 เส้นไปใช้ขับรีเลย์ 12 V ทั้ง 8 ตัวได้โดยตรงส่วนอีกขั้วของรีเลย์จะถูกต่ออยู่กับไฟบวก 12 V รอไว้อยู่แล้ว โดยรีเลย์ที่ใช้จะเป็นแบบ 1 หน้าสัมผัสแบบปกติเปิด (Normal Open : NO) จะถูกต่ออยู่กับไฟ AC 220 V เพื่อนำไปต่อกับอุปกรณ์ไฟฟ้า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 7 Flowchart ของระบบทั้งหมด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทำงานของโฟล์ชาร์ต ดังรูปที่ 3.13 เริ่มแรกจะทำการตั้งค่าเริ่มต้นให้กับพอร์ตต่างๆ จากนั้นจะทำการส่งค่าเริ่มต้นให้กับ Real Time Clock และจอ LCD และจะเข้าสู่โปรแกรมส่วนของหลัก การทำงานคือ จะทำการอ่านค่าเวลาจริงจาก RTC ไปแสดงผลบนจอ LCD จากนั้นจะทำการเรียกโปรแกรมย่อยตรวจสอบเวลาอ้างอิงกับค่าเวลาจากโปรแกรมที่ตั้งเอาไว้ โดยในส่วนของโปรแกรมย่อยนี้จะตรวจสอบทั้ง 8 เอาท์พุตโดยแต่ละเอาท์พุต จะมีทั้งหมด 30 โปรแกรมในแต่ละวัน หลังจากทำการตรวจสอบค่าเวลาอ้างอิงกับค่าเวลาที่ตั้งเอาไว้แล้ว จะทำการส่งข้อมูลที่ได้จากการตรวจสอบไปยัง Output Relay จากนั้นจะทำการตรวจสอบว่ามีการกดคีย์ใดๆหรือไม่ จากขา Data Available ถ้าไม่มีการกดคีย์ใดๆ จะกลับไปทำงานในส่วนของการอ่านค่าเวลาจาก RTC ต่อไป

แต่ถ้ามีการกดคีย์ใดๆ ก็จะทำการอ่านค่าจาก P1.0 – P1.3 เข้ามาเก็บไว้จากนั้นจะนำไปตรวจสอบว่าคีย์ใดถูกกด และจะทำการเรียกโปรแกรมย่อย MENU FUNCTION ซึ่ง MENU FUNCTION จะมีหลายเมนูด้วยกันเช่น Setting (Date/Time), Change Password, About Program และ Check Load Status ซึ่งจะเรียกใช้เมนูไหนนั้นก็ขึ้นอยู่กับคีย์ที่กด

โดยในแต่ละเมนูจะมีโปรแกรมย่อยในส่วนของการตรวจสอบการกดคีย์และการหน่วงเวลาด้วย คือ ถ้าไม่มีการกดคีย์ใดๆภายในเวลาประมาณ 10 วินาที โปรแกรมจะทำการกระโดดออกไปยังส่วนการอ่านค่าจาก RTC ใหม่อีกครั้ง หรือถ้าคีย์ที่กดเป็นคีย์เพื่อกลับไปยังหน้าจอหลักโปรแกรมก็จะทำการกระโดดออกไปยังการอ่านค่าจาก RTC ใหม่อีกครั้งเช่นเดียวกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4
การทดลองและผลการทดลอง

ในบทนี้จะกล่าวถึงการทดลองใช้งานเครื่องควบคุมอุปกรณ์ไฟฟ้าภายในบ้านแบบโปรแกรมได้ว่ามีการใช้งานอย่างไร ผู้ใช้สามารถกำหนดการใช้งานอย่างไร และจะได้ผลอย่างไร ดังต่อไปนี้

เครื่องควบคุมอุปกรณ์ไฟฟ้าภายในบ้านแบบโปรแกรมได้ ถูกออกแบบมาให้ทำงานในลักษณะของสวิตช์ที่สามารถต่อคร่อมสวิตช์เดิมได้ทันที โดยมี Relay Output ทำหน้าที่เป็นสวิตช์ดังกล่าว ซึ่งจากการเชื่อมต่อดังที่กล่าวมานี้ จะเห็นได้ว่าการเปิด-ปิดอุปกรณ์ไฟฟ้าสามารถทำได้ 2 ทางคือ การเปิดสวิตช์ที่มีอยู่เดิมซึ่งสามารถทำได้ด้วยมือ และการเปิดสวิตช์ที่ต่อเพิ่มเข้ามาในที่นี้คือ Relay Output ซึ่งสามารถทำได้โดยการโปรแกรมเท่านั้น จากลักษณะดังข้างต้น เครื่องควบคุมอุปกรณ์ไฟฟ้าภายในบ้านแบบโปรแกรมได้นี้ จึงมีโหมดการทำงานให้เลือกใช้ด้วยกัน 2 โหมดคือ

(1) โหมดการทำงานตามโปรแกรมหรือแบบอัตโนมัติ (Program Mode or Auto Mode) ซึ่งเราสามารถกำหนดการเปิด-ปิดอุปกรณ์ไฟฟ้าได้จากการตั้งโปรแกรมให้กับอุปกรณ์ ในแต่ละโปรแกรมจะประกอบด้วย เวลา (ชั่วโมง+นาที) และลักษณะการเปิด-ปิดของอุปกรณ์ ณ เวลาที่กำหนด ซึ่งเราสามารถควบคุมการทำงานของอุปกรณ์ไฟฟ้าได้ทั้งสิ้น 8 อุปกรณ์ แต่ละอุปกรณ์สามารถโปรแกรมได้ 30 โปรแกรมต่อวัน

การทำงานในโหมดการทำงานแบบอัตโนมัตินี้จะถูกต้องตามโปรแกรมที่เรากำหนดก็ต่อเมื่อสวิตช์ที่มีอยู่เดิมจะต้องอยู่ในตำแหน่งปิด (OFF) เท่านั้น

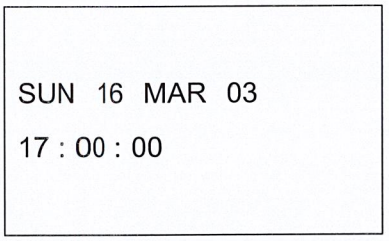
(2) โหมดการทำงานด้วยมือ (Manual Mode) ซึ่งก็คือการเปิด-ปิดเครื่องใช้ไฟฟ้าโดยใช้สวิตช์ที่มีอยู่เดิม ดังนั้นในโหมดการทำงานนี้ โปรแกรมจะต้องทำให้ Relay Output มีสถานะปิด (OFF) ทั้งหมด

การเลือกโหมดการทำงานของอุปกรณ์ไฟฟ้าแบบโปรแกรมได้นี้ ทำได้โดยการเลื่อนสวิตช์ไปอยู่ในตำแหน่งของโหมดการทำงานที่ต้องการ

ต่อจากนี้จะกล่าวถึงการใช้งานในโหมดการทำงานแบบอัตโนมัติหรือตามโปรแกรม ซึ่งจะอธิบายให้ผู้ใช้งานสามารถใช้งานเครื่องควบคุมอุปกรณ์ไฟฟ้าภายในบ้านแบบโปรแกรมได้นี้อย่างมีประสิทธิภาพสูงสุด

หลังจากจ่ายไฟให้กับอุปกรณ์แล้ว LCD จะแสดงหน้าจอดังรูปที่ 4.1 ซึ่งเป็นหน้าจอหลักของเครื่องควบคุมอุปกรณ์ไฟฟ้าแบบโปรแกรมได้นี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



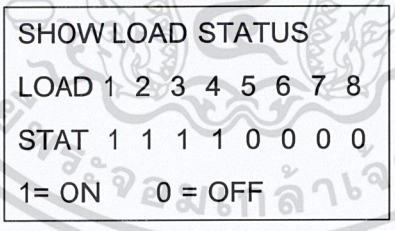
รูปที่ 4.1 ตัวอย่างการแสดงผลที่ LCD Display

จากรูปที่ 4.1 เป็นตัวอย่างการแสดงผลที่ LCD Display ในขณะที่ใช้งานปกติ (ยังไม่มีการกดคีย์ใด ๆ) โดยที่บรรทัดแรกของหน้าจอจะแสดงค่าวัน, วันที่, เดือน และปี ส่วนในบรรทัดถัดมาจะแสดงค่าเวลาเป็น ชั่วโมง, นาที และวินาที ซึ่งค่าต่าง ๆ เหล่านี้จะเป็นค่าจริงอยู่ตลอดเวลา นอกจากนี้ยังเป็นการบอกสภาวะว่าเครื่องพร้อมใช้งาน

*ในขั้นตอนนี้จะไม่มีการเปิดไฟ Black Light ของ LCD

เมื่อมีการกดคีย์ต่าง ๆ ก็จะเปิดไฟ Black Light ของ LCD และเข้าสู่เมนูการตั้งวัน, ตั้งเวลา, ตั้งค่ารหัสผ่าน, ตรวจสอบสถานะของเครื่องใช้ไฟฟ้าที่ต่ออยู่ที่เอาท์พุตสามารถทำงานได้ตามเวลาที่กำหนดดังจะอธิบายดังต่อไปนี้

เมื่อทำการกดคีย์ “HELP” ที่ LCD Display จะแสดงดังรูปที่ 4.2



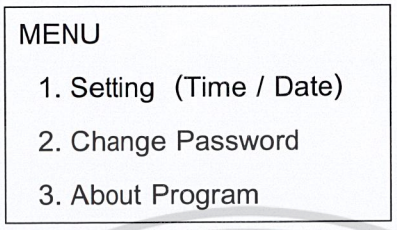
รูปที่ 4.2 ตัวอย่างการแสดงผลที่ LCD Display เมื่อทำการกดคีย์ HELP เพื่อตรวจสอบสถานะของเครื่องใช้ไฟฟ้า

จากรูปที่ 4.2 เป็นตัวอย่างการแสดงผลที่ LCD Display เมื่อกดคีย์ “HELP” โดยบรรทัดที่ 2 จะบอกว่าเป็นเครื่องใช้ไฟฟ้าเครื่องที่เท่าไหร่ ส่วนบรรทัดที่ 3 จะแสดงสถานะของเครื่องใช้ไฟฟ้าแต่ละเครื่องโดยถ้าเป็น “1” หมายถึง เครื่องใช้ไฟฟ้านั้นกำลังเปิด (On) หรือ กำลังทำงานอยู่ แต่ถ้าเป็น “0” หมายถึง เครื่องใช้ไฟฟ้านั้นปิดอยู่ (Off) หรือ เครื่องใช้ไฟฟ้านั้น ๆ ไม่ได้ทำงาน จากขั้นตอนนี้ถ้าหากมี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การกดคีย์ “BACK” แล้ว จะทำการปิดไฟ Black Light ของ LCD และจะแสดงหน้าจอดังรูปที่ 4.1 ต่อไป

เมื่อทำการคีย์ใด ๆ ยกเว้นคีย์ “HELP” จะแสดงดังรูปที่ 4.3



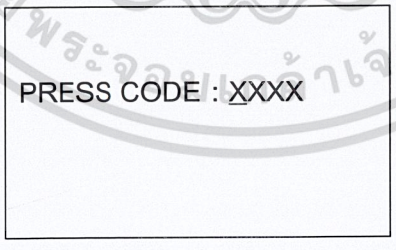
รูปที่ 4.3 แสดงหน้าจอเมื่อทำการกดคีย์ใด ๆ (ยกเว้นคีย์ HELP)

จากรูปที่ 4.3 แสดงหน้าจอ LCD ซึ่งเป็นเมนู (Menu) หลักของเครื่องควบคุมนี้ จะเห็นได้ว่ามีเมนูย่อยให้เลือกทั้งหมด 3 เมนู ซึ่งจะกล่าวถึงรายละเอียดของแต่ละเมนูต่อไป

จากขั้นตอนนี้ถ้าหากมีการกดคีย์ “BACK” แล้ว โปรแกรมจะทำการปิดไฟ Black Light และจะแสดงหน้าจอดังรูปที่ 4.1 ต่อไป

4.1 การตั้งค่าวันและตั้งค่าเวลา

จากหน้าจอ LCD ดังรูปที่ 4.3 หากมีการกดคีย์ “1” ซึ่งจะเป็นการเข้าไปยังเมนูย่อยที่ 1 ซึ่งเป็นเมนูเกี่ยวกับการตั้งค่าวันและเวลา ซึ่งจะแสดงหน้าจอดังรูปที่ 4.4



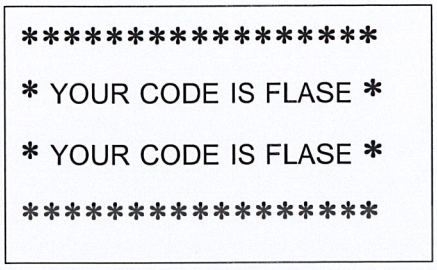
รูปที่ 4.4 แสดงหน้าจอเพื่อให้กดรหัสผ่าน

จากรูปที่ 4.4 แสดงหน้าจอ LCD เพื่อให้กดรหัสผ่านโดยถ้าหากเป็นการใช้งานครั้งแรกหรือมีการ Reset เครื่องควบคุมจะกำหนดให้รหัสผ่านมีค่าเป็น “0000”

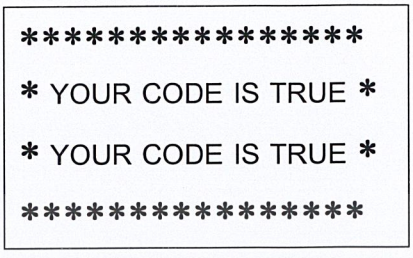
จากขั้นตอนนี้ถ้าหากมีการกดคีย์ “BACK” แล้ว จะแสดงหน้าจอ LCD ดังรูปที่ 4.3 เพื่อรอรับการกดคีย์ใด ๆ ต่อไป

เอกสารนี้เป็นลิขสิทธิ์ของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ห้ามมิให้ทำซ้ำ ดัดแปลง หรือจำหน่ายจ่ายแจก ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 4.4 ในขั้นตอนนี้ หากมีการกดรหัสผ่านผิด จะแสดงหน้าจอ LCD ดังรูปที่ 4.5 เป็นเวลา 1 วินาที แล้วจะแสดงหน้าจอ LCD ดังรูปที่ 4.4 เพื่อรอการกดรหัสผ่านอีกครั้ง

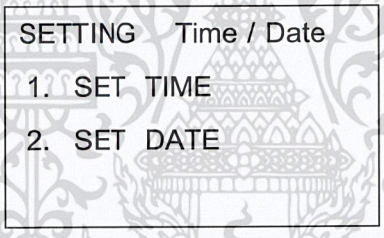


รูปที่ 4.5 แสดงหน้าจอ LCD เมื่อกดรหัสผ่านผิด



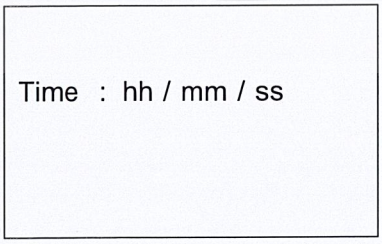
รูปที่ 4.6 แสดงหน้าจอ LCD เมื่อกดรหัสผ่านถูกต้อง

แต่ถ้าหากกดรหัสผ่านได้ถูกต้อง จะแสดงหน้าจอ LCD ดังรูปที่ 4.6 เป็นเวลา 1 วินาที แล้วจะแสดงหน้าจอดังรูปที่ 4.7



รูปที่ 4.7 แสดงหน้าจอ LCD ซึ่งเป็นเมนูการตั้งค่าวันและเวลา

จากรูปที่ 4.7 ซึ่งเป็นเมนูเกี่ยวกับการตั้งค่าวันและเวลาจากขั้นตอนนี้หากมีการกดคีย์ “BACK” แล้วจะแสดงหน้าจอ LCD ดังรูปที่ 4.3 เพื่อรอรับการกดคีย์ใด ๆ ต่อไป แต่หากกดคีย์ “1” ก็จะเข้าสู่เมนูของการตั้งค่าเวลา ดังแสดงในรูปที่ 4.8



รูปที่ 4.8 แสดงหน้าจอ LCD ซึ่งเป็นเมนูการตั้งเวลา

จากขั้นตอนนี้หากมีการกดคีย์ “BACK” แล้วจะแสดงหน้าจอดังรูปที่ 4.7 เพื่อรอการกดคีย์ใด ๆ ต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 4.8 แสดงหน้าจอ LCD เกี่ยวกับเมนูการตั้งเวลาโดยจะมีรายละเอียดของสัญลักษณ์ต่าง ๆ ดังนี้

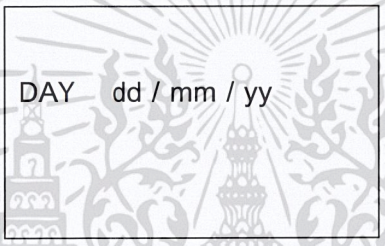
hh = ให้ป้อนค่าชั่วโมง โดยมีค่าอยู่ระหว่าง 00-23

mm = ให้ป้อนค่านาที โดยมีค่าอยู่ระหว่าง 00-59

ss = ให้ป้อนค่าวินาที โดยมีค่าอยู่ระหว่าง 00-59

เมื่อทำการป้อนค่าต่าง ๆ ครบหมดแล้ว ให้กดคีย์ "ENTER" ซึ่งจะถือว่าตั้งค่าเวลาเสร็จสมบูรณ์แล้ว และเวลาอ้างอิงก็จะเปลี่ยนจากค่าเดิมไปเป็นค่าที่เราได้ตั้งเอาไว้ หลังจากนั้นหน้าจอ LCD จะแสดงดังรูปที่ 4.7 อีกครั้ง

จากรูปที่ 4.7หากมีการกดคีย์ "2" จะเข้าสู่เมนูของการตั้งค่าวัน ดังแสดงในรูปที่ 4.9



รูปที่ 4.9 แสดงหน้าจอ LCD ที่เป็นเมนูการตั้งค่าวัน

เช่นเดียวกับรูปที่ 4.8 หากมีการกดคีย์ "BACK" แล้วจะแสดงห้าจอ LCD ดังรูปที่ 4.7 อีกครั้งเพื่อรอการกดคีย์ใด ๆ ต่อไป

จากรูปที่ 4.9 แสดงหน้าจอ LCD เกี่ยวกับเมนูการตั้งวันโดยจะมีรายละเอียดของสัญลักษณ์ต่าง ๆ ดังนี้

DAY = ให้ป้อนวัน SUN-SAT โดยกด 1-7

dd = ให้ป้อนค่าวันที่ โดยมีค่าอยู่ระหว่าง 01-31

mm = ให้ป้อนค่าเดือน โดยมีค่าอยู่ระหว่าง 01-12

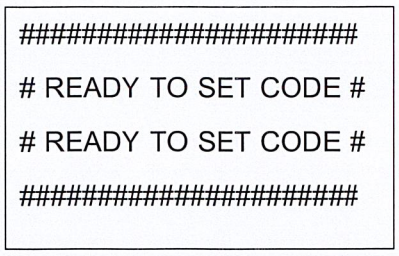
yy = ให้ป้อนค่าปี ค.ศ. โดยมีค่าอยู่ระหว่าง 00-99

เช่นเดียวกับการตั้งเวลา คือเมื่อป้อนค่าต่าง ๆ ครบแล้วให้กดคีย์ "ENTER" จึงจะถือว่าการตั้งค่าเสร็จสมบูรณ์ แล้ว LCD จะแสดงหน้าจอดังรูปที 4.7 อีกครั้ง

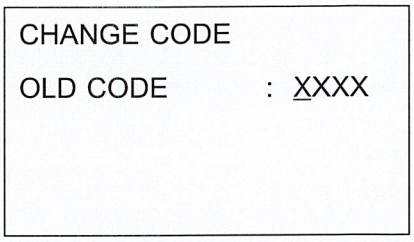
4.2 การเปลี่ยนรหัสผ่าน

จากหน้าจอ LCD ดังรูปที่ 4.3 (จากเมนูแรก) หากมีการกดคีย์ "2" จะเป็นการเข้าไปยังเมนูย่อยที่ 2 (Change Password) ซึ่งเป็นเมนูเกี่ยวกับการเปลี่ยนรหัสผ่าน โดย LCD จะแสดงหน้าจอดังรูปที่ 4.10 ก่อนเป็นเวลา 1 วินาที แล้วจึงแสดงหน้าจอดังรูปที่ 4.11

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



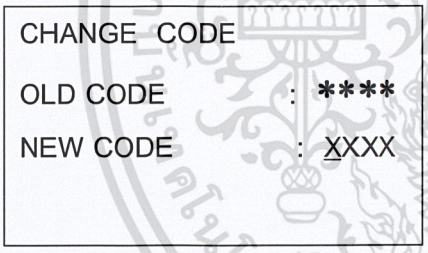
รูปที่ 4.10 แสดงหน้าจอ LCDแจ้งให้ทราบว่าจะทำการเปลี่ยนรหัสผ่าน



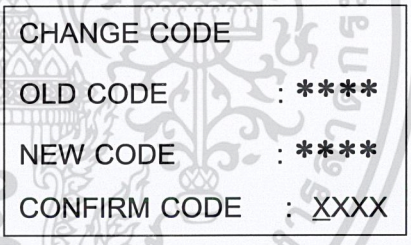
รูปที่ 4.11 แสดงหน้าจอ LCD รับค่ารหัสผ่านเก่า

จากรูปที่ 4.11 จะเป็นการรับค่ารหัสผ่านเก่าก่อนที่จะทำการเปลี่ยนรหัสผ่าน เพื่อเป็นการป้องกันการเปลี่ยนรหัสผ่านจากบุคคลภายนอก โดยถ้าป้อนรหัสผ่านผิด LCD จะแสดงหน้าจอดังรูปที่ 4.5 เป็นเวลา 1 วินาที แล้วจะแสดงหน้าจอดังรูปที่ 4.11 อีกครั้ง จากขั้นตอนนี้หากมีการกดคีย์ “BACK” LCD จะแสดงหน้าจอดังรูปที่ 4.7 เพื่อรับค่าคีย์ใด ๆ ต่อไป

แต่ถ้าหากว่ารหัสผ่านที่ป้อนเข้าไปถูกต้อง LCD จะแสดงหน้าจอดังรูปที่ 4.12



รูปที่ 4.12 แสดงหน้าจอLCDรับค่ารหัสผ่านใหม่ (เมื่อกดรหัสผ่านเก่าถูกต้อง)

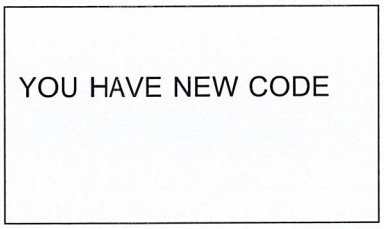


รูปที่ 4.13 แสดงหน้าจอLCDรับค่ารหัสผ่านเก่า

จากรูปที่ 4.12 แสดงหน้าจอ LCD เพื่อรับค่ารหัสผ่านใหม่ ซึ่งถ้าหากว่ากดรหัสผ่านใหม่ครบทั้ง 4 ตัวแล้วจะแสดงหน้าจอดังรูปที่ 4.13 ซึ่งเป็นการแสดงหน้าจอ LCD ที่รอรับการยืนยันรหัสผ่านใหม่อีกครั้ง จากขั้นตอนนี้หากมีการกดคีย์ “BACK” จะแสดงหน้าจอ LCD ดังรูปที่ 4.12 เพื่อรับการกดรหัสผ่านใหม่อีกครั้ง

จากรูปที่ 4.13 หากมีการยืนยันรหัสผ่านไม่ตรงกับรหัสผ่านใหม่ที่ได้ป้อนเอาไว้ จะแสดงหน้าจอดังรูปที่ 4.5 เป็นเวลา 1 วินาที แล้ว LCD จะแสดงหน้าจอดังรูปที่ 4.11 อีกครั้ง แต่ถ้าการยืนยันรหัสผ่านถูกต้อง LCD จะแสดงหน้าจอ ดังรูปที่ 4.14 เป็นเวลา 1 วินาที แล้วกลับไปแสดงหน้าจอดังรูปที่ 4.3 (กลับไปยังเมนูแรก) ซึ่งจะถือว่าการเปลี่ยนรหัสผ่านเสร็จสมบูรณ์แล้ว

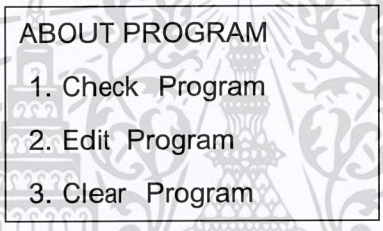
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.14 แสดงหน้าจอ LCD แจ้งให้ทราบว่าการเปลี่ยนรหัสผ่านเสร็จสมบูรณ์

4.3 การป้อนโปรแกรมเปิด-ปิดอุปกรณ์ไฟฟ้าภายในบ้าน

ถ้าในขณะที่ LCD แสดงหน้าจอดังรูปที่ 4.3 แล้วมีการกดคีย์ "3" LCD จะเปลี่ยนมาแสดงหน้าจอดังรูปที่ 4.15 ซึ่งเป็นหน้าจอที่แสดงถึงการตั้งค่าเวลาเพื่อทำงานเกี่ยวกับการควบคุมการเปิด-ปิดอุปกรณ์ไฟฟ้าภายในบ้านโดยจะเห็นได้ว่ามีเมนูย่อยอีก 3 เมนูคือ



รูปที่ 4.15 แสดงหน้าจอ LCD ซึ่งแสดงเมนูการทำงานเกี่ยวกับการเปิด-ปิดอุปกรณ์ไฟฟ้าภายในบ้าน

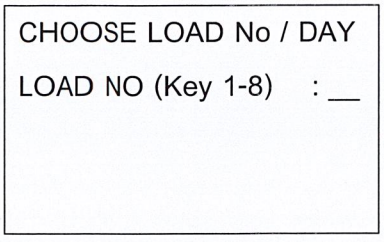
1. Check Program เป็นเมนูที่ใช้ในการตรวจสอบหรือดูว่าโปรแกรม (ซึ่งประกอบด้วยค่าเวลาและลักษณะการทำงานของอุปกรณ์ไฟฟ้า ณ เวลานั้นว่าเปิดหรือปิด) ที่เราได้กำหนดให้แก่อุปกรณ์ไฟฟ้าแต่ละตัว ในแต่ละวันนั้นมีอะไรบ้าง
2. Edit Program เป็นเมนูที่ใช้ในการป้อนค่าหรือกำหนดโปรแกรมการทำงานให้แก่อุปกรณ์ไฟฟ้าแต่ละตัว ในแต่ละวัน
3. Clear Program เป็นเมนูที่ใช้ในการลบโปรแกรมการทำงานของอุปกรณ์ไฟฟ้าที่เราไม่ต้องการ

ซึ่งจากขั้นตอนนี้หากมีการกดคีย์ "BACK" แล้ว LCD จะกลับไปแสดหน้าจอดังรูปที่ 4.3

4.3.1 การตรวจสอบโปรแกรม(Check Program)

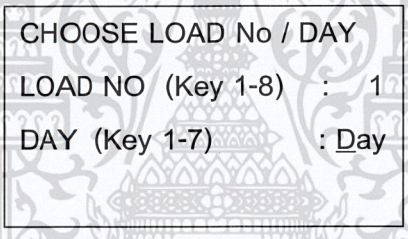
จากรูปที่ 4.15 หากมีการกดคีย์ "1" ซึ่งเป็นการเข้าไปทำงานในเมนู Check Program หน้าจอ LCD จะแสดงดังรูปที่ 4.16

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.16 แสดงหน้าจอ LCD ซึ่งใช้เพื่อเลือกอุปกรณ์ไฟฟ้าที่ต้องการติดต่อ (ตรวจสอบหรือตั้งค่าโปรแกรม)

จากรูปที่ 4.16 เป็นหน้าจอ LCD ซึ่งใช้เลือกอุปกรณ์ไฟฟ้าที่เราต้องการติดต่อ (ตรวจสอบหรือตั้งค่าโปรแกรม) การเลือกอุปกรณ์ไฟฟ้านั้นสามารถทำได้โดยการกดคีย์ “1-8” ซึ่งหากมีการกดคีย์ “1-8” คีย์ใดคีย์หนึ่งแล้ว LCD จะแสดงหน้าจอดังรูปที่ 4.17 (ในที่นี้สมมติให้กดคีย์ 1) แต่หากเป็นการกดคีย์ “BACK” LCD จะแสดงหน้าจอดังรูปที่ 4.15 เพื่อเข้าสู่เมนูต่าง ๆ ต่อไป



รูปที่ 4.17 แสดงหน้าจอ LCD ใช้เพื่อเลือกวัน (SUN – SAT) หลังจากที่เลือกอุปกรณ์ไฟฟ้าแล้ว

จากรูปที่ 4.17 แสดงหน้าจอ LCD เพื่อใช้เลือกวัน หลังจากที่ได้ทำการเลือกอุปกรณ์ไฟฟ้าเรียบร้อยแล้ว

หากมีการกดคีย์ “BACK” LCD จะแสดงหน้าจอดังรูปที่ 4.16 เพื่อทำการเลือกอุปกรณ์ไฟ-ฟ้าอีกครั้ง แต่ถ้าเป็นการกดคีย์ “1-7” LCD จะแสดงวัน (SUN-SAT) ที่ตำแหน่งของ Curser แทนคำว่า Day ถ้าหากเราป้อนวันผิดพลาดสามารถแก้ไขใหม่ได้โดยการกดคีย์ “BACK” แล้วทำการป้อนวันใหม่อีกครั้ง (กดคีย์ 1-7) ซึ่งถ้าหากถูกต้องแล้ว สามารถเข้าไปตรวจสอบโปรแกรมได้โดยการกดคีย์ “ENTER“ ซึ่งจะแสดงดังรูปที่ 4.18 (ในที่นี้สมมติให้เลือกวันอาทิตย์ คือกดคีย์ “1”)

<u>สมมติว่า</u> เราได้ทำการป้อนโปรแกรมสำหรับวันอาทิตย์ (SUN) ของอุปกรณ์ไฟฟ้าชิ้นที่ 1 ไว้ดังนี้

01	08 : 00	ON
02	12 : 00	OFF
03	13 : 00	ON

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

PROGRAM LOAD No.1 SUN		
01	08 : 00	ON
02	12 : 00	OFF
03	13 : 00	ON

รูปที่ 4.18 แสดงตัวอย่างหน้าจอ LCD ที่แสดงโปรแกรมสำหรับควบคุม
การเปิด-ปิดอุปกรณ์ไฟฟ้าที่ได้ป้อนเอาไว้

จากรูปที่ 4.18 คือหน้าจอของ LCD ที่แสดงโปรแกรมของวันอาทิตย์ควบคุมอุปกรณ์ไฟฟ้าที่ 1 จะเห็นว่าหน้าจอ LCD สามารถแสดงโปรแกรมได้เพียง 3 โปรแกรมนั้น ซึ่งตามการออกแบบแล้วจะมีทั้งสิ้น 30 โปรแกรม เราสามารถดูโปรแกรมที่เหลือได้โดยการกดคีย์ " ▽ " และคีย์ " △ " ซึ่งเป็นการเลื่อนการแสดงโปรแกรมทีละบรรทัด

จากรูปที่ 4.18 หากกดคีย์ " ▽ " 1 ครั้ง LCD จะแสดงหน้าจอดังรูปที่ 4.19และถ้ากดอีกครั้งจะได้หน้าจอดังรูปที่ 4.20

PROGRAM LOAD No.1 SUN		
02	12 : 00	OFF
03	13 : 00	ON
END		

รูปที่ 4.19 แสดงหน้าจอ LCD เมื่อกดคีย์ " ▽ " 1 ครั้ง

PROGRAM LOAD No.1 SUN		
03	13 : 00	ON
END		
END		

รูปที่ 4.20 แสดงหน้าจอ LCD เมื่อกดคีย์ " ▽ " 2 ครั้ง

ซึ่งจากหน้าจอ LCD ดังรูปที่ 4.20 หากกดคีย์ "△ " LCD จะแสดงหน้าจอดังรูปที่ 4.19 และหากกดคีย์ "△ " อีกครั้งจะแสดงหน้าจอดังรูปที่ 4.18 ซึ่งในรูปที่ 4.18 หากกดคีย์ " △ " จะไม่มีการเปลี่ยนแปลงใด ๆ ทั้งสิ้น ซึ่งในขั้นตอนเหล่านี้หากคีย์ที่กดเป็นคีย์ " BACK" LCD จะแสดงหน้าจอดังรูปที่ 4.16 และรอการทำงานในขั้นตอนต่อไป

*** จากรูปที่ 4.19 ละ 4.20 จะเห็นได้ว่ามีข้อความ END ปรากฏขึ้นนั่นหมายความว่าได้มีการตั้งค่าโปรแกรมไว้เพียงเท่านี้ จากนี้ไปไม่มีโปรแกรมใด ๆ ทั้งสิ้น

4.3.2 การตั้งค่าหรือการกำหนดโปรแกรม (Edit Program)

ในขณะที่ LCD แสดงหน้าจอดังรูปที่ 4.15 หากมีการกดคีย์ "2" จะเป็นการเข้าไปทำงานในเมนู Edit Program ซึ่ง LCD จะแสดงหน้าจอรับการกดรหัสผ่านดังรูปที่ 4.4 หากการป้อนรหัสผ่านนั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ถูกต้อง LCD แสดงหน้าจอดังรูปที่ 4.16 เพื่อรอการเลือกอุปกรณ์ไฟฟ้าที่ต้องการตั้งค่าโปรแกรมต่อไป

จากขั้นตอนการรับค่ารหัสผ่านและจากรูปที่ 4.16 หากมีการกดคีย์ “BACK” LCD จะแสดงหน้าจอดังรูปที่ 4.15 เพื่อทำเมนูต่างๆต่อไป

จากรูปที่ 4.16 จะทำการเลือกอุปกรณ์ไฟฟ้าและทำการเลือกวัน เพื่อตั้งค่าโปรแกรมหรือกำหนดโปรแกรมมีขั้นตอนและวิธีการเช่นเดียวกับที่ได้อธิบายเอาไว้แล้ว ในเรื่องการตรวจสอบโปรแกรม (หัวข้อ 4.3.1)

สมมติว่า ต้องการแก้ไขเวลาเปิดจาก 8.00น. เป็น 8.30น. และทำการเพิ่มโปรแกรมการปิดอุปกรณ์ไฟฟ้านี้ที่เวลา 17.00น.

เมื่อเลือกอุปกรณ์ไฟฟ้าและวันแล้ว (ในที่นี้เลือกอุปกรณ์ไฟฟ้าที่1และเลือกวันอาทิตย์) LCD จะแสดงหน้าจอดังรูปที่ 4.18 ซึ่งเราต้องการแก้ไขโปรแกรมที่ 01 ให้เปิด (ON) ที่เวลา 8.30น. สามารถทำได้โดย กดคีย์ “0” ตามด้วยคีย์ “1” ซึ่ง LCD จะแสดงหน้าจอดังรูปที่ 4.21

PROGRAM LOAD No.1 SUN		
01	hh : mm	0/1
02	12 : 00	OFF
03	13 : 00	ON

รูปที่ 4.21 แสดงหน้าจอ LCD เมื่อต้องการแก้ไขโปรแกรม (ในที่นี้คือโปรแกรมที่ 1)

จากรูปที่ 4.21 ให้ทำการป้อนค่าเวลาที่ต้องการจะกำหนดให้กับอุปกรณ์และในส่วนของการกำหนดลักษณะการทำงาน(เปิดหรือปิด) ทำได้โดยการกดคีย์ “0” หมายถึง ให้ปิดที่เวลาที่กำหนดก่อนหน้านี้ ส่วนการกดคีย์ “1” หมายถึง ให้เปิดที่เวลาที่กำหนดก่อนหน้านี้ หลังจากนั้นให้กดคีย์ “ENTER” ซึ่งจะถือว่าการป้อนโปรแกรมนั้นเสร็จสมบูรณ์ ซึ่งจะได้ดังรูปที่ 4.22

PROGRAM LOAD No.1 SUN		
01	08 : 30	ON
02	12 : 00	OFF
03	13 : 00	ON

รูปที่ 4.22 แสดงหน้าจอ LCD เมื่อได้ทำการแก้ไขโปรแกรมเรียบร้อยแล้ว

จากรูปที่ 4.22 ถ้าต้องการเพิ่มโปรแกรมการควบคุมการเปิด-ปิดให้กับอุปกรณ์ไฟฟ้า โดยไม่ต้องการแก้ไขโปรแกรมที่มีอยู่แล้วสามารถทำได้โดย ป้อนค่าลงไปแทนที่ตำแหน่งโปรแกรมที่เป็นข้อความ “END” ซึ่งทำได้โดยการกดคีย์ “▽ “ จนกว่าจะมีข้อความ “END” ปรากฏขึ้นที่หน้าจอ (ในกรณี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ที่หน้าจอขณะนั้นไม่มีข้อความ “END” เท่านั้น แต่ถ้าหน้าจอในขณะนั้นมีข้อความ “END” อยู่แล้ว สามารถป้อนโปรแกรมได้เลย) จากรูปที่ 4.22 หากทำการกดคีย์ “ ▽ “ 1 ครั้งจะแสดงหน้าจอดังรูปที่ 4.19 ซึ่งจากรูปนี้จะเห็นได้ว่าโปรแกรมเดิมที่ได้กำหนดไว้มีถึง 03 เพราะฉะนั้นโปรแกรมใหม่ที่ป้อนเข้าไปจะต้องเป็นโปรแกรมที่ 04 ดังนั้นจากรูปที่ 4.19 ถ้ากดคีย์ “0” ตามด้วยคีย์ “4” แล้ว LCD จะแสดงหน้าจอดังรูปที่ 4.23

PROGRAM LOAD No.1 SUN		
02	12 : 00	OFF
03	13 : 00	ON
04	hh : mm	0/1

รูปที่ 4.23 แสดงหน้าจอ LCD เมื่อต้องการเพิ่มโปรแกรมที่ 4 ให้กับอุปกรณ์ไฟฟ้า

จากรูปที่ 4.23 เมื่อป้อนเวลาและลักษณะการทำงานแล้ว(ในที่นี้คือ 17.00น. ให้ OFF) ให้กดคีย์ “ENTER” แล้ว LCD จะแสดงหน้าจอดังรูปที่ 4.24

PROGRAM LOAD No.1 SUN		
03	13 : 00	ON
04	17 : 00	OFF
END		

รูปที่ 4.24 แสดงหน้าจอ LCD เมื่อตั้งค่าโปรแกรมเสร็จสมบูรณ์และรอการตั้งค่าครั้งต่อไป

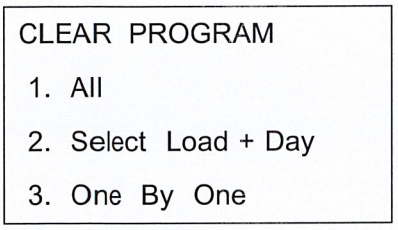
จากรูปที่ 4.24 เมื่อเปรียบเทียบกับรูปที่ 4.23 จะเห็นได้ว่าเมื่อตั้งค่าโปรแกรมเสร็จแล้วจะทำการเลื่อนโปรแกรมให้ 1 บรรทัดโดยอัตโนมัติ (ในกรณีที่ตั้งค่าโปรแกรมในบรรทัดสุดท้ายเท่านั้น) ทั้งนี้เพื่อความสะดวกในการตั้งค่าโปรแกรมต่อๆไป

การออกจากเมนูนี้ทำได้โดยการกดคีย์ “BACK” ซึ่ง LCD จะแสดงหน้าจอดังรูปที่ 4.16 เพื่อการทำงานต่างๆ ตามที่ต้องการต่อไป

4.3.3 **การลบโปรแกรม** (Clear Program)

ในขณะที่ LCD แสดงหน้าจอดังรูปที่ 4.15 หากมีการกดคีย์ “3” จะเป็นการเข้าไปทำงานในเมนู Clear Program โดย LCD จะแสดงหน้าจอรับการกดรหัสผ่านดังรูปที่ 4.4 ซึ่งหากการกด รหัสผ่านถูกต้อง LCD จะแสดงหน้าจอดังรูปที่ 4.25

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.25 แสดงหน้าจอ LCD ที่แสดงเมนูย่อยการลบโปรแกรมของอุปกรณ์ไฟฟ้า

จากรูปที่ 4.25 จะเป็นเมนูย่อยการลบโปรแกรมซึ่งมีรายละเอียดดังนี้

(1) All เป็นการลบโปรแกรมทั้งหมด คือการลบโปรแกรมของอุปกรณ์ไฟฟ้าทั้ง 8 ตัวและลบทุกวันตั้งแต่วันอาทิตย์ถึงวันเสาร์

(2) Select Load+Day เป็นการลบโปรแกรมของอุปกรณ์ไฟฟ้าตัวใดตัวหนึ่ง ภายในวันใดวันหนึ่งเท่านั้น โดยลบหมดทั้ง 30 โปรแกรมของวันนั้นๆ

(3) One By One เป็นการลบโปรแกรมของอุปกรณ์ไฟฟ้า โดยทำการลบทีละ 1 โปรแกรมเท่านั้น

จากรูปที่ 4.25 หากมีการกดคีย์ "1" LCD จะแสดงหน้าจอดังรูปที่ 4.26 เป็นเวลา 1 วินาที เพื่อเป็นการแจ้งให้ทราบว่าได้ทำการลบโปรแกรมเรียบร้อยแล้ว จากนั้นหน้าจอ LCD จะแสดงหน้าจอดังรูปที่ 4.25 อีกครั้ง

รูปที่ 4.26 แสดงหน้าจอ LCD เพื่อแจ้งให้ทราบว่าได้ทำการลบโปรแกรมแล้ว

จากรูปที่ 4.25 หากมีการกดคีย์ "2" LCD จะแสดงหน้าจอดังรูปที่ 4.16 เพื่อทำการเลือกอุปกรณ์ไฟฟ้าที่ต้องการทำการลบโปรแกรม ซึ่งขั้นตอนและวิธีการในการเลือกอุปกรณ์ไฟฟ้าและการเลือกวันได้อธิบายแล้วในเรื่องการตรวจสอบโปรแกรม (หัวข้อที่ 4.3.1) ซึ่งเมื่อได้เลือกค่าต่างๆแล้ว LCD จะแสดงหน้าจอดังรูปที่ 4.26 เป็นเวลา 1 วินาที เพื่อแจ้งให้ทราบว่าได้ทำการลบโปรแกรมของอุปกรณ์ไฟฟ้านั้นภายในวันที่ถูกเรียบร้อยแล้ว จากนั้น LCD จะแสดงหน้าจอดังรูปที่ 4.25 อีกครั้ง

เอกสารนี้สงวนการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 4.25 หากมีการกดคีย์ “3” LCD จะแสดงหน้าจอดังรูปที่ 4.16 เพื่อทำการเลือกอุปกรณ์ไฟฟ้าและเลือกวันดังที่ได้อธิบายไปแล้ว ซึ่งถ้าหากทำการเลือกเสร็จแล้ว LCD จะแสดงหน้าจอดังรูปที่ 4.22 (สมมติ เลือกวันอาทิตย์และอุปกรณ์ไฟฟ้าที่ 1)

จากรูปที่ 4.22 หากเราต้องการทำการลบโปรแกรมที่ 2 และ 3 ออกเพื่อให้อุปกรณ์ไฟฟ้าเปิดที่เวลา 8.30น. และปิดที่เวลา 17.00น. เท่านั้น สามารถทำได้ดังนี้คือ ให้กดคีย์ “0” ตามด้วยคีย์ “2” แล้ว LCD จะแสดงหน้าจอดังรูปที่ 4.27

PROGRAM LOAD No.1 SUN		
01	08 : 30	ON
02	13 : 00	ON
03	17 : 00	OFF

รูปที่ 4.27 แสดงหน้าจอ LCD เมื่อทำการลบโปรแกรมที่ 2 เรียบร้อยแล้ว

จากรูปที่ 4.27 จะเห็นได้ว่าโปรแกรมที่ 2 (เดิม) ได้ถูกลบทิ้งเป็นที่เรียบร้อยแล้ว และโปรแกรมที่ 3 จะเลื่อนมาเป็นโปรแกรมที่ 2 ส่วนโปรแกรมที่ 4 ก็จะเลื่อนมาเป็นโปรแกรมที่ 3 เช่นกัน และจะกระทำการเลื่อนแบบนี้เรื่อยๆไปจนถึงโปรแกรมที่ 30

และจากรูปที่ 4.27 ต้องทำการลบโปรแกรมที่ 2 (โปรแกรมที่ 3 เดิม) สามารถทำได้โดยการกดคีย์ “0” ตามด้วยคีย์ “2” ซึ่ง LCD จะแสดงหน้าจอดังรูปที่ 4.28

PROGRAM LOAD No.1 SUN		
01	08 : 30	ON
02	17 : 00	OFF
END		

รูปที่ 4.28 แสดงหน้าจอ LCD เมื่อทำการลบโปรแกรมที่ 2 (โปรแกรมที่ 3 เดิม) เรียบร้อยแล้ว

จากรูปที่ 4.28 จะเห็นได้ว่าโปรแกรมที่ 2 (โปรแกรม 3 เดิม) ถูกลบทิ้งแล้ว และเหลือเพียงโปรแกรมที่เราต้องการเท่านั้น

เราสามารถออกจากเมนูนี้ได้โดยการกดคีย์ “BACK” ซึ่ง LCD จะแสดงหน้าจอดังรูปที่ 4.25 เพื่อการทำงานต่างๆต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หากมีการตั้งค่าโปรแกรมโดยที่เวลาซ้ำกัน แต่ลักษณะการเปิด-ปิด (ON-OFF) แตกต่างกัน การทำงานของอุปกรณ์นั้นจะขึ้นอยู่กับลักษณะการเปิด-ปิดของโปรแกรมที่อยู่ในตำแหน่งที่มากกว่าหรือมากที่สุด (โปรแกรมล่าสุด) ดังนี้

PROGRAM LOAD No.1 SUN		
01	08 : 00	ON
02	08 : 00	OFF
END		

(ก)

PROGRAM LOAD No.1 SUN		
01	08 : 00	OFF
02	08 : 00	ON
END		

(ข)

รูปที่ 4.29 แสดงการตั้งค่าเวลาของโปรแกรมซ้ำกัน แต่ลักษณะการเปิด-ปิดแตกต่างกัน

จากรูปที่ 4.29 ถ้าหากทำการตั้งค่าโปรแกรมดังรูปที่ 4.29 (ก) ที่เวลา 8.00 น.อุปกรณ์ไฟฟ้านั้นจะปิด (OFF) แต่ถ้าหากตั้งค่าโปรแกรมดังรูปที่ 4.29 (ข) อุปกรณ์ไฟฟ้านั้นจะเปิด (ON)

*** **หมายเหตุ** ในทุกๆขั้นตอนการทำงานหากไม่มีการกดคีย์ใดๆ ภายในเวลา 10 วินาที โปรแกรมจะทำการปิดไฟ Black Light ของ LCD และ LCD จะแสดงหน้าจอหลักดังรูปที่ 4.3 เสมอ

ผลการทดลอง หลังจากทำการตั้งโปรแกรมให้กับอุปกรณ์แล้ว เมื่อเวลาที่ได้ตั้งเอาไว้ตรงกับฐานเวลาอ้างอิง การทำงานของอุปกรณ์ไฟฟ้าที่ทำการควบคุมนั้นสามารถทำงานได้ตามต้องการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5
บทวิจารณ์และสรุป

5.1 สรุปผลการทดลอง

จากการทดลองเครื่องควบคุมอุปกรณ์ไฟฟ้าภายในบ้านแบบโปรแกรมได้สามารถสรุปผลการทดลองได้ว่า ในการทำงานที่ต้องอาศัยการอ้างอิงเวลา (Real Time Clock) โดยฐานเวลาอ้างอิงนี้จะต้องมีความเที่ยงตรงอย่างมาก เพราะเป็นตัวกำหนดการทำงานของเอาท์พุตให้ทำงานได้ตรงตามค่าเวลาที่แท้จริง ซึ่งถ้าหากเราใช้ไทเมอร์ที่มีอยู่ภายในตัวไมโครคอนโทรลเลอร์นั้นอาจจะไม่ได้ค่าเวลาที่แท้จริงเท่ากับการใช้ Real Time Clock แต่ในการใช้ Real Time Clock ก็มีข้อจำกัดที่ว่าจะต้องทำการใส่แบตเตอร์รี่แบ็กอัพให้ฐานเวลาสามารถผลิตสัญญาณนาฬิกาออกมาได้ตรงตามความเป็นจริงมากที่สุด ส่วนในการเขียนโปรแกรมให้ทำงานนั้นก็ใช้ความรู้ในการประยุกต์ใช้ไมโครคอนโทรลเลอร์ตระกูล MCS-51มาเป็นส่วนประมวลผลให้โปรแกรมสามารถไปควบคุมวงจรที่นำมาต่อร่วมกับไมโครคอนโทรลเลอร์ได้อย่างมีประสิทธิภาพมากที่สุด

ส่วนในการตั้งโปรแกรมให้เอาท์พุททำงานนั้นสามารถทำให้เอาท์พุทปิดหรือเปิดได้ในรอบวัน โดยแต่ละอุปกรณ์จะสามารถโปรแกรมได้สูงสุด 30 โปรแกรมต่อวัน ถ้าต้องการให้สามารถปิดหรือเปิดได้มากกว่านี้จะต้องมีการเพิ่มหน่วยความจำให้กับระบบมากขึ้นเพื่อเป็นการเพิ่มประสิทธิภาพให้มีฟังก์ชั่นการทำงานที่หลากหลายมากยิ่งขึ้น ซึ่งถ้าทำการเพิ่มหน่วยความจำเข้าไปแล้วก็สามารถเพิ่มจำนวนเอาท์พุทให้มากกว่า 8 ได้ซึ่งอาจจะเป็น 16 หรือมากกว่านี้ก็ได้

ส่วนในวงจรภาครีเลย์เอาท์พุทถ้าต้องการให้มีการใช้งานกับเครื่องใช้ไฟฟ้าที่มีกำลังไฟเพิ่มขึ้น เช่น เครื่องปรับอากาศ จะต้องทำการเปลี่ยนรีเลย์ใหม่ให้หน้าสัมผัสสามารถทนได้มากกว่านี้

5.2 ปัญหาที่เกิดจากการทดลอง

เนื่องจากได้นำไมโครคอนโทรลเลอร์ไปต่อกับอุปกรณ์ภายนอกหลายอย่าง เช่น LCD, Relay Output, คีย์แพดขนาด 4X4, สวิตช์เลือกโหมดการทำงาน, หน่วยความจำภายนอก และ RTC จึงเป็นผลให้พอร์ตของไมโครคอนโทรลเลอร์ไม่เพียงพอกับความต้องการ จึงต้องทำการศึกษาหาวิธีที่สามารถลดจำนวนเส้นสัญญาณต่างๆลงให้น้อยที่สุดเท่าที่จะทำได้ ซึ่งจากการศึกษาเราได้เลือกใช้ PCF8574 เนื่องจากมีการส่งสัญญาณด้วยระบบบัสแบบ I^2C ซึ่งใช้เส้นสัญญาณเพียง 2 เส้นเท่านั้น

การแบ็คอัพข้อมูลของไมโครคอนโทรลเลอร์และหน่วยความจำภายนอก เนื่องจากไฟฟ้าดับหรือเกิดแหล่งจ่ายไฟขัดข้องนั้นสามารถทำได้เพียง 6 ชั่วโมงเท่านั้น ซึ่งถ้าเกินเวลาที่กำหนดนี้จะเกิดความสูญเสียต่อข้อมูลโปรแกรมที่เก็บไว้ในหน่วยความจำโปรแกรมภายนอก เป็นผลให้เราต้องเสียเวลาในการโปรแกรมใหม่อีกครั้ง การจะทำให้การแบ็คอัพข้อมูลของไมโครคอนโทรลเลอร์และหน่วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ความจำภายนอกสามารถทำได้มากกว่า 6 ชั่วโมงนั้น อาจต้องใช้แบตเตอร์รี่แบ็คอัพที่มีขนาดใหญ่กว่าเดิม เป็นผลให้เครื่องควบคุมเครื่องใช้ไฟฟ้าภายในบ้านแบบโปรแกรมได้นี้มีขนาดใหญ่ ซึ่งในเรื่องนี้หากต้องการนำไปใช้งานได้ตามต้องการจะต้องทำการศึกษาและพัฒนาต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
;*****************************************************************
; Program    :  Programable Electric Appliance Controller
; Description: For control 8 electrics appliance on or off in real time
; For        :  MCS-51 Microcontroller, choose AT89S53
; Filename   :  Project4.asm
; Assembler  :  SXA51
; *****************************************************************
;*******************************
;  Define Port&Pin Name
;*******************************
SDA          BIT    P3.2  ;  SDA I2C BUS
SCL          BIT    P3.3  ;  SCL I2C BUS
SW_1         BIT    P3.4  ;  Switch for set to Manual mode
LCD_EN       BIT    P1.6  ;  LCD Module Enable (Active High : Level)
LCD_RS       BIT    P1.5  ;  LCD Module Register select
KEY_DA       BIT    P1.4

LCD_BLK      BIT    P1.7  ;  LED Black Light of LCD Module
BEEP         BIT    P3.0

;*******************************
;  Define User Register
;*******************************
FLAG         EQU    02FH  ;  User Flag
I2C_ACK      BIT    FLAG.0      ;  Define I2C Acknowledge as bit
;*******************************
;  Define User Register
;*******************************
LCD_ADDR     EQU    030H  ;  For keep LCD Address
LCD_PTR      EQU    032H  ;  For keep LCD 3 Char Pointer
KPAD_DATA    EQU    033H  ;  For keep Keypad Data
LCD_DATA     EQU    034H  ;  For keep LCD Data

I2C_ADDR     EQU    035H  ;  For keep I2C Address
I2C_DATA     EQU    036H  ;  For keep I2C Data
IO_DATA      EQU    037H  ;  For keep I2C 8 bit I/O Data

BUFFER       EQU    038H  ;  Buffer For keep Data from Keypad

COD1         EQU    03AH  ;  For keep Real Code
COD2         EQU    03BH
COD3         EQU    03CH
COD4         EQU    03DH

BUFF1        EQU    03EH  ;  For keep Code that User keyed
BUFF2        EQU    03FH
BUFF3        EQU    040H
BUFF4        EQU    041H
ERR_TRUE     EQU    042H  ;For keep Data that Code is keyed True or False

REST1        EQU    043H  ;  For keep Old Code when want to Change Code
REST2        EQU    044H
REST3        EQU    045H
REST4        EQU    046H

BUFF_LOAD    EQU    047H  ;For keep Number of Load which want to contact
BUFF_DAY     EQU    048H  ;For keep Day (Sun-Sat) which want to contact
BACK_CODE    EQU    049H  ;  For keep Code of Menu Function
LOW_DP       EQU    04AH  ;  For keep DPTR Low bit
HIGH_DP      EQU    04BH  ;  For keep DPTR High bit
RAM_PTR      EQU    04CH  ;  For keep Ram Pointer
EDIT_PTR     EQU    04DH

BUFF_DPL     EQU    04EH
BUFF_DPH     EQU    04FH
SLIDE_DPL    EQU    050H
SLIDE_DPH    EQU    051H
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งไม่อนุญาตให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
BUFF_L1     EQU   052H  ;  For keep Status of each Load ( On-Off )
BUFF_L2     EQU   053H
BUFF_L3     EQU   054H
BUFF_L4     EQU   055H
BUFF_L5     EQU   056H
BUFF_L6     EQU   057H
BUFF_L7     EQU   058H
BUFF_L8     EQU   059H
BUFF_OUT    EQU   05AH  ;  For keep Data of Relay Output

SECONDS     EQU   060H  ;  For keep Seconds
MINUTES     EQU   061H  ;  For keep Minutes
HOURS EQU   062H        ;  For keep Hours
DAY         EQU   063H  ;  For keep Day
DATE        EQU   064H  ;  For keep Date
MONTH EQU   065H        ;  For keep Month
YEAR        EQU   066H  ;  For keep Year
CONTROL     EQU   067H  ;  For keep Control Data off RTC

WDOG        EQU   096H  ;  Internal RAM Address off Watchdog Timer

;*******************************
;  define I2C Slave Address
;*******************************
PCF8574_ID  EQU   031H        ;  For keep I2C Slave Address
PCF8574_1   EQU   01110010B   ;  Address of The First PCF8574
PCF8574_2   EQU   01110100B   ;  Address of The Seconde PCF8574
RTC_ID      EQU   11010000B   ;  Address of RTC

;*************************************************************
; Main Program
; *************************************************************
            ORG   0000H             ; Reset Vecter
            MOV   P0,#00000000B     ; Clear Data Bus
            MOV   P1,#01111111B
            MOV   P2,#00000000B     ; Clear Status Lcd
            MOV   P3,#11111110B
            MOV   WDOG,#11000011B

MAIN:       LCALL INIT_LCD          ;  Call LCD Initial Subroutine
            MOV   COD1,#0           ;  Set Code to 0000
            MOV   COD2,#0
            MOV   COD3,#0
            MOV   COD4,#0
            MOV   BACK_CODE,#000H
            SETB  LCD_BLK           ;  On LCD Black Light
            SETB  BEEP              ;  Off Sound "Beep"

LOOP:       LCALL LCD_CLR
            LCALL SHOW_CLOCK        ;  Call Subroutine Show Real Time Clock
            MOV   WDOG,#11000011B
            LCALL DELAY_1s
            MOV   WDOG,#11000011B
            CLR   LCD_BLK           ;  Off LCD Black Light

            MOV   A,BACK_CODE
            CJNE  A,#002H,LOOP1
            LCALL SWAP_C2R
            MOV   BACK_CODE,#000H

LOOP1:      MOV   WDOG,#11000011B
            LCALL SHOW_CLOCK ;  Call Subroutine Show Real Time Clock
            LCALL GETOUT_DATA ;  Call Subroutine Get Status of each Load
            LCALL GETBUFF_OUT ;  Call Subroutine Get Output to Relay Output
            MOV   IO_DATA,BUFF_OUT  ;  Move Output to Relay Output though PCF8574
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            MOV    PCF8574_ID,#PCF8574_2
            LCALL  PCF8574_WR

            JNB    SW_1,PRESS_SW1     ;  Jump to set to Manual Mode if SW1 is seted
            JB     KEY_DA,GOTO_FIRST_MENU  ; Jump to Menu Function if pressed key
            LJMP   LOOP1

;*****************************************************************
; CHECK  KEY IS PRESSED WAS KEY "HELP" OR NOT ?
; *****************************************************************
GOTO_FIRST_MENU:  MOV    WDOG,#11000011B
            SETB   LCD_BLK
            LCALL  GETKEY_DATA
            MOV    A,KPAD_DATA
            CJNE   A,#00EH,GOTO_FIRST_MENU1 ;  Check key is pressed is "HELP" ?
            LCALL  SHOW_STAT                ;  Call Subroutine Show Load Status



GOTO_1ST_MENU0_4: MOV    R5,#90
GOTO_1ST_MENU0_3: MOV    R6,#100
GOTO_1ST_MENU0_2: MOV    R7,#0E6H
GOTO_1ST_MENU0:   MOV    WDOG,#11000011B
                  DJNZ   R7,GOTO_1ST_MENU0_1
                  DJNZ   R6,GOTO_1ST_MENU0_2
                  DJNZ   R5,GOTO_1ST_MENU0_3
                  LJMP   LOOP  ;  if don't press any key in 10s return to main
GOTO_1ST_MENU0_1: JNB    KEY_DA,GOTO_1ST_MENU0

            LCALL  GETKEY_DATA
            MOV    A,KPAD_DATA
            CJNE   A,#00DH,GOTO_1ST_MENU0_5 ; Check key is pressed is "BACK" ?
            LCALL  KEY_RELEASE
            LJMP   LOOP

GOTO_1ST_MENU0_5: CLR    BEEP
            LCALL  DELAY_100ms
            SETB   BEEP
            LJMP   GOTO_1ST_MENU0_4

GOTO_FIRST_MENU1:       LCALL FIRST_MENU    ;  Call Subroutine Show 1st Menu
            LJMP   LOOP                             ;  if key is pressed is not
"HELP"


;*****************************************************************
; FUNCTION SET TO MANUAL MODE BY USING SWITCH
; *****************************************************************
PRESS_SW1:  MOV    WDOG,#11000011B
            LCALL  LCD_CLR
            SETB   LCD_BLK
            MOV    LCD_ADDR,#040H     ;  Show LCD set to Manual Mode
            LCALL  SET_ADDR_LCD
            MOV    DPTR,#TITLE_1
            LCALL  WRLINE_LCD
            MOV    WDOG,#11000011B
            LCALL  DELAY_1s
            MOV    WDOG,#11000011B
            LCALL  DELAY_1s
            MOV    WDOG,#11000011B
            LCALL  LCD_CLR
            MOV    IO_DATA,#000H      ;  Off All Relay Output
            MOV    PCF8574_ID,#PCF8574_2
            LCALL  PCF8574_WR
PRESS_SW1_0:      MOV    WDOG,#11000011B
            LCALL  SHOW_CLOCK          ;  Call Subroutine Show Real Time Clock
            LCALL  DELAY_1s
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
PRESS_SW1_1:      MOV   WDOG,#11000011B
            CLR   LCD_BLK
            LCALL SHOW_CLOCK  ;  Call Subroutine Show Real Time Clock

            LCALL GETOUT_DATA ;  Call Subroutine Get Status of each Load
            LCALL GETBUFF_OUT ;  Call Subroutine Get Output to Relay Output

            JB    KEY_DA,PRESS_SW1_2 ;  Check Key Pad is pressed or not ?
            JNB   SW_1,PRESS_SW1_1

            MOV   WDOG,#11000011B
            LCALL LCD_CLR
            MOV   LCD_ADDR,#040H     ;  Show LCD that set to Program Mode

            LCALL SET_ADDR_LCD
            MOV   DPTR,#TITLE_23
            LCALL WRLINE_LCD
            SETB  LCD_BLK
            LCALL DELAY_1s
            MOV   WDOG,#11000011B
            LCALL DELAY_1s
            LJMP  LOOP
PRESS_SW1_2:      MOV   WDOG,#11000011B
            LCALL GETKEY_DATA
            MOV   A,KPAD_DATA
            CJNE  A,#00EH,PRESS_SW1_4   ;  Check key is pressed is "HELP" ?
            SETB  LCD_BLK
            LCALL SHOW_STAT             ;  Call Subroutine Show Load Status

            MOV   LCD_DATA,#'0'
            MOV   LCD_ADDR,#019H
            LCALL SET_ADDR_LCD
            LCALL WRCHAR_LCD

            MOV   LCD_ADDR,#01BH
            LCALL SET_ADDR_LCD
            LCALL WRCHAR_LCD

            MOV   LCD_ADDR,#01DH
            LCALL SET_ADDR_LCD
            LCALL WRCHAR_LCD

            MOV   LCD_ADDR,#01FH
            LCALL SET_ADDR_LCD
            LCALL WRCHAR_LCD

            MOV   LCD_ADDR,#021H
            LCALL SET_ADDR_LCD
            LCALL WRCHAR_LCD

            MOV   LCD_ADDR,#023H
            LCALL SET_ADDR_LCD
            LCALL WRCHAR_LCD

            MOV   LCD_ADDR,#025H
            LCALL SET_ADDR_LCD
            LCALL WRCHAR_LCD

            MOV   LCD_ADDR,#027H
            LCALL SET_ADDR_LCD
            LCALL WRCHAR_LCD
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
PRESS_SW1_34:     MOV   R5,#90
PRESS_SW1_33:     MOV   R6,#100
PRESS_SW1_32:     MOV   R7,#0E6H
PRESS_SW1_30:     MOV   WDOG,#11000011B
```

```
                DJNZ  R7,PRESS_SW1_31
                DJNZ  R6,PRESS_SW1_32
                DJNZ  R5,PRESS_SW1_33
                LCALL LCD_CLR
                LJMP  PRESS_SW1_1
PRESS_SW1_31:   JNB   KEY_DA,PRESS_SW1_30

          LCALL GETKEY_DATA
          MOV   A,KPAD_DATA
          CJNE  A,#00DH,PRESS_SW1_35
          LCALL KEY_RELEASE
          LCALL LCD_CLR
          LJMP  PRESS_SW1_0

PRESS_SW1_4:    SETB  LCD_BLK
          LCALL FIRST_MENU

PRESS_SW1_35:   CLR   BEEP
          LCALL       DELAY_100ms
          SETB  BEEP
          LJMP  PRESS_SW1_34

;*************************************************************
;FUNCTION SHOW FIRST MENU OF PROGRAM
;*************************************************************
FIRST_MENU: LCALL LCD_CLR
FIRST_MENU1:    MOV   WDOG,#11000011B
          MOV   LCD_ADDR,#000H
          LCALL SET_ADDR_LCD
          MOV   DPTR,#TITLE_3
          LCALL WRLINE_LCD

          MOV   LCD_ADDR,#040H
          LCALL SET_ADDR_LCD
          LCALL WRLINE_LCD

          MOV   LCD_ADDR,#014H
          LCALL SET_ADDR_LCD
          LCALL WRLINE_LCD

          MOV   LCD_ADDR,#054H
          LCALL SET_ADDR_LCD
          LCALL WRLINE_LCD
          MOV   WDOG,#11000011B
          LCALL DELAY_1s
          LCALL KEY_RELEASE


FIRST_MENU2_4:  MOV   R5,#90
FIRST_MENU2_3:  MOV   R6,#100
FIRST_MENU2_2:  MOV   R7,#0E6H
FIRST_MENU2:    MOV   WDOG,#11000011B
                DJNZ  R7,FIRST_MENU2_1
                DJNZ  R6,FIRST_MENU2_2
                DJNZ  R5,FIRST_MENU2_3
                LJMP  LOOP
FIRST_MENU2_1:  JNB   KEY_DA,FIRST_MENU2

          LCALL GETKEY_DATA
          MOV   A,KPAD_DATA
          CJNE  A,#01H,FIRST_MENU3
          LCALL PRESS_SETTING
          LJMP  FIRST_MENU1
FIRST_MENU3:    CJNE  A,#02H,FIRST_MENU4
          LCALL CHANGE_CODE
          LCALL LCD_CLR
          MOV   LCD_ADDR,#040H
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            LCALL SET_ADDR_LCD
            MOV   DPTR,#TITLE_12
            LCALL WRLINE_LCD
            MOV   WDOG,#11000011B
            LCALL DELAY_1s
            LJMP  FIRST_MENU1

FIRST_MENU4:      CJNE  A,#03H,FIRST_MENU5
            LCALL ABOUT_PRO
            LJMP  FIRST_MENU1

FIRST_MENU5:      CJNE  A,#00DH,FIRST_MENU6
            LJMP  LOOP

FIRST_MENU6:      CLR   BEEP
            LCALL DELAY_100ms
            SETB  BEEP
            LJMP  FIRST_MENU2_4

;*************************************************
;SETTING Menu Function ( When press key 1 from 1st Menu )
;*************************************************
PRESS_SETTING:    MOV   WDOG,#11000011B
            MOV   BACK_CODE,#001H
            LCALL SHOW_CODE
            LCALL KEY_RELEASE
            LCALL PRESSED_CODE
            LCALL CH_COD
            MOV   A,ERR_TRUE
            CJNE  A,#0,TRUE
            MOV   WDOG,#11000011B
            LCALL CODE_ERR
            AJMP  PRESS_SETTING

TRUE:       MOV   WDOG,#11000011B
            LCALL CODE_TRUE
            LCALL LCD_CLR
TIME_DATE:  MOV   WDOG,#11000011B
            MOV   LCD_ADDR,#000H
            LCALL SET_ADDR_LCD
            MOV   DPTR,#TITLE_13
            LCALL WRLINE_LCD

            MOV   LCD_ADDR,#040H
            LCALL SET_ADDR_LCD
            LCALL WRLINE_LCD

            MOV   LCD_ADDR,#014H
            LCALL SET_ADDR_LCD
            LCALL WRLINE_LCD
            LCALL KEY_RELEASE

TIME_DATE1_4:     MOV   R5,#90
TIME_DATE1_3:     MOV   R6,#100
TIME_DATE1_2:     MOV   R7,#0E6H
TIME_DATE1_0:     MOV   WDOG,#11000011B
                  DJNZ  R7,TIME_DATE1_1
                  DJNZ  R6,TIME_DATE1_2
                  DJNZ  R5,TIME_DATE1_3
                  LJMP  LOOP
TIME_DATE1_1:     JNB   KEY_DA,TIME_DATE1_0
            LCALL GETKEY_DATA
            MOV   A,KPAD_DATA
            CJNE  A,#01H,TIME_DATE2
            LCALL SET_TIME
            LJMP  TIME_DATE
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งไม่อนุญาตให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
TIME_DATE2: MOV   A,KPAD_DATA
            CJNE  A,#02H,TIME_DATE3
            LCALL SET_DATE
            LJMP  TIME_DATE

TIME_DATE3: MOV   A,KPAD_DATA
            CJNE  A,#00DH,TIME_DATE4
            RET
TIME_DATE4: CLR   BEEP
            LCALL DELAY_100ms
            SETB  BEEP
            LJMP  TIME_DATE1_4

;*************************************************
;CHANGE CODE Menu Function ( When press key 2 from 1st Menu )
;*************************************************
CHANGE_CODE:      MOV   WDOG,#11000011B
            MOV   BACK_CODE,#001H
            LCALL LCD_CLR
            MOV   LCD_ADDR,#000H
            LCALL SET_ADDR_LCD
            MOV   DPTR,#TITLE_5
            LCALL WRLINE_LCD

            MOV   LCD_ADDR,#040H
            LCALL SET_ADDR_LCD
            LCALL WRLINE_LCD

            MOV   LCD_ADDR,#014H
            LCALL SET_ADDR_LCD
            LCALL WRLINE_LCD

            MOV   LCD_ADDR,#054H
            LCALL SET_ADDR_LCD
            LCALL WRLINE_LCD
            MOV   WDOG,#11000011B
            LCALL DELAY_1s

CLEAR_LCD:  MOV   WDOG,#11000011B
            LCALL LCD_CLR
            MOV   LCD_ADDR,#00H
            LCALL SET_ADDR_LCD
            MOV   DPTR,#TITLE_8
            LCALL WRLINE_LCD

            MOV   LCD_ADDR,#040H
            LCALL SET_ADDR_LCD
            LCALL WRLINE_LCD
            LCALL KEY_RELEASE


OLD_CODE:   MOV   WDOG,#11000011B
            MOV   LCD_ADDR,#050H
            LCALL SET_ADDR_LCD
            LCALL LCD_BLINK

            LCALL PRESSED_CODE

            LCALL CH_COD
            MOV   A,ERR_TRUE
            CJNE  A,#0,NEW_COD
            LCALL CODE_ERR
            MOV   WDOG,#11000011B
            LCALL DELAY_1s
            LJMP  CLEAR_LCD

NEW_COD:    MOV   WDOG,#11000011B
            MOV   BACK_CODE,#00H
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            MOV   LCD_ADDR,#014H
            LCALL SET_ADDR_LCD
            MOV   DPTR,#TITLE_10
            LCALL WRLINE_LCD

            MOV   LCD_ADDR,#024H
            LCALL SET_ADDR_LCD
            LCALL LCD_BLINK
            LCALL PRESSED_CODE
            LCALL SWAP_C2R
            LCALL SWAP_C2B

CONFIRM_COD:      MOV   WDOG,#11000011B
            MOV   BACK_CODE,#002H
            MOV   LCD_ADDR,#054H
            LCALL SET_ADDR_LCD
            MOV   DPTR,#TITLE_11
            LCALL WRLINE_LCD

            MOV   LCD_ADDR,#064H
            LCALL SET_ADDR_LCD
            LCALL LCD_BLINK

            LCALL PRESSED_CODE
            LCALL CH_COD
            MOV   A,ERR_TRUE
            CJNE  A,#1,VALID_CODE
            RET

VALID_CODE: MOV   WDOG,#11000011B
            LCALL SWAP_C2R
            LCALL CODE_ERR
            MOV   WDOG,#11000011B
            CLR   BEEP
            LCALL DELAY_1s
            SETB  BEEP
            MOV   WDOG,#11000011B
            LJMP  CLEAR_LCD


;*************************************************
;ABOUT PROGRAM Menu Function ( When press key 3 from 1st Menu )
;*************************************************
ABOUT_PRO:  MOV   WDOG,#11000011B
            MOV   BACK_CODE,#03H
            LCALL LCD_CLR
            MOV   LCD_ADDR,#000H
            LCALL SET_ADDR_LCD
            MOV   DPTR,#TITLE_14
            LCALL WRLINE_LCD

            MOV   LCD_ADDR,#040H
            LCALL SET_ADDR_LCD
            LCALL WRLINE_LCD

            MOV   LCD_ADDR,#014H
            LCALL SET_ADDR_LCD
            LCALL WRLINE_LCD

            MOV   LCD_ADDR,#054H
            LCALL SET_ADDR_LCD
            LCALL WRLINE_LCD
            MOV   WDOG,#11000011B
            LCALL DELAY_1s
            LCALL KEY_RELEASE

ABOUT_PRO1_4:     MOV   R5,#90
ABOUT_PRO1_3:     MOV   R6,#100
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งไม่อนุญาตให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
ABOUT_PRO1_2:      MOV    R7,#0E6H
ABOUT_PRO1_0:      MOV    WDOG,#11000011B
                   DJNZ   R7,ABOUT_PRO1_1
                   DJNZ   R6,ABOUT_PRO1_2
                   DJNZ   R5,ABOUT_PRO1_3
                   LJMP   LOOP
ABOUT_PRO1_1:      JNB    KEY_DA,ABOUT_PRO1_0

             LCALL GETKEY_DATA
             MOV   A,KPAD_DATA
             CJNE  A,#01H,ABOUT_PRO2
             LCALL CHECK_PRO
             LJMP  ABOUT_PRO
ABOUT_PRO2:  CJNE  A,#02H,ABOUT_PRO3
             LCALL EDIT_PRO
             LJMP  ABOUT_PRO
ABOUT_PRO3:  CJNE  A,#03H,ABOUT_PRO4
             LCALL CLEAR_PRO
             LJMP  ABOUT_PRO
ABOUT_PRO4:  CJNE  A,#00DH,ABOUT_PRO1_5
             RET
ABOUT_PRO1_5:      CLR    BEEP
             LCALL DELAY_100ms
             SETB  BEEP
             LJMP  ABOUT_PRO1_4

;*************************************************
;FUNCTION SHOW PROGRAM WHICH IS SETTED
;*************************************************
CHECK_PRO:   MOV    WDOG,#11000011B
             LCALL  GET_LOAD_DAY
             MOV    A,BACK_CODE
             CJNE   A,#04H,CHECK_PRO0
             RET
CHECK_PRO0:  MOV    WDOG,#11000011B
             LCALL  TITLE_LOAD_DAY

CHECK_PRO1:  MOV    RAM_PTR,#000H
             LCALL  GET_ADDR_RAM
             LCALL  SAVE_DP
CHECK_PRO2:  MOV    WDOG,#11000011B
             MOV    LCD_ADDR,#040H
             LCALL  SET_ADDR_LCD
             MOV    DPTR,#TITLE_19
             LCALL  WRLINE_LCD
             MOV    LCD_ADDR,#040H
             LCALL  SET_ADDR_LCD
             LCALL  SHOW_RAM

             MOV    LCD_ADDR,#014H
             LCALL  SET_ADDR_LCD
             MOV    DPTR,#TITLE_19
             LCALL  WRLINE_LCD
             MOV    LCD_ADDR,#014H
             LCALL  SET_ADDR_LCD
             LCALL  SHOW_RAM

             MOV    WDOG,#11000011B
             MOV    LCD_ADDR,#054H
             LCALL  SET_ADDR_LCD
             MOV    DPTR,#TITLE_19
             LCALL  WRLINE_LCD
             MOV    LCD_ADDR,#054H
             LCALL  SET_ADDR_LCD
             LCALL  SHOW_RAM
             LCALL  KEY_RELEASE

CHECK_PRO_14:      MOV    R5,#90
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
CHECK_PRO_13:      MOV   R6,#100
CHECK_PRO_12:      MOV   R7,#0E6H
CHECK_PRO_10:      MOV   WDOG,#11000011B
                   DJNZ  R7,CHECK_PRO_11
                   DJNZ  R6,CHECK_PRO_12
                   DJNZ  R5,CHECK_PRO_13
                   LJMP  LOOP
CHECK_PRO_11:      JNB   KEY_DA,CHECK_PRO_10

            LCALL GETKEY_DATA
            MOV   A,KPAD_DATA

CHECK_P_UP: CJNE  A,#00AH,CHECK_P_DW
            MOV   A,RAM_PTR
            CJNE  A,#000H,CHECK_P_UP1
            CLR   BEEP
            LCALL DELAY_100ms
            SETB  BEEP
            LJMP  CHECK_PRO_14
CHECK_P_UP1:       DEC   RAM_PTR
            MOV   R5,#010H
CHECK_P_UP2:       LCALL UPLOAD_DP
            MOV   A,DPL
            CJNE  A,#000H,CHECK_P_UP3
            MOV   DPL,#0FFH
            DEC   DPH
            LCALL SAVE_DP
            DJNZ  R5,CHECK_P_UP2
            LJMP  CHECK_PRO2

CHECK_P_UP3:       DEC   DPL
CHECK_P_UP4:       LCALL SAVE_DP
            DJNZ  R5,CHECK_P_UP2
            LJMP  CHECK_PRO2

CHECK_P_DW: CJNE  A,#00BH,CHECK_PRO_2
            MOV   A,RAM_PTR
            CJNE  A,#01BH,CHECK_P_DW1
            LJMP  CHECK_PRO1
CHECK_P_DW1:       INC   RAM_PTR
            MOV   R5,#008H
CHECK_P_DW2:       LCALL UPLOAD_DP
            MOV   A,DPL
            CJNE  A,#000H,CHECK_P_DW3
            MOV   DPL,#0FFH
            DEC   DPH
            LCALL SAVE_DP
            DJNZ  R5,CHECK_P_DW2
            LJMP  CHECK_PRO2

CHECK_P_DW3:       DEC   DPL
CHECK_P_DW4:       LCALL SAVE_DP
            DJNZ  R5,CHECK_P_DW2
            LJMP  CHECK_PRO2

CHECK_PRO_2:       CJNE  A,#00DH,CHECK_PRO_15
            LJMP  CHECK_PRO
CHECK_PRO_15:      CLR   BEEP
                   LCALL DELAY_100ms
                   SETB  BEEP
                   LJMP  CHECK_PRO_14
;*****************************************************
;FUNCTION EDIT PROGRAM WHICH CHOOSE LOAD+DAY
;*****************************************************
EDIT_PRO:   LCALL LCD_CLR
            LCALL SHOW_CODE
            LCALL KEY_RELEASE
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            LCALL PRESSED_CODE
            MOV   A,BACK_CODE
            CJNE  A,#04H,EDIT_PRO00
            RET

EDIT_PRO00: LCALL CH_COD
            MOV   A,ERR_TRUE
            CJNE  A,#0,EDIT_TRUE
            LCALL CODE_ERR
            LJMP  EDIT_PRO

EDIT_TRUE:  LCALL CODE_TRUE

EDIT_PRO1:  LCALL GET_LOAD_DAY
            MOV   A,BACK_CODE
            CJNE  A,#04H,EDIT_PRO0
            RET

EDIT_PRO0:  LCALL TITLE_LOAD_DAY

EDIT_PRO2:  MOV   RAM_PTR,#000H
            LCALL GET_ADDR_RAM
            LCALL SAVE_DP
EDIT_PRO3:  MOV   LCD_ADDR,#040H
            LCALL SET_ADDR_LCD
            MOV   DPTR,#TITLE_19
            LCALL WRLINE_LCD
            MOV   LCD_ADDR,#040H
            LCALL SET_ADDR_LCD
            LCALL SHOW_RAM

            MOV   LCD_ADDR,#014H
            LCALL SET_ADDR_LCD
            MOV   DPTR,#TITLE_19
            LCALL WRLINE_LCD
            MOV   LCD_ADDR,#014H
            LCALL SET_ADDR_LCD
            LCALL SHOW_RAM

            MOV   LCD_ADDR,#054H
            LCALL SET_ADDR_LCD
            MOV   DPTR,#TITLE_19
            LCALL WRLINE_LCD
            MOV   LCD_ADDR,#054H
            LCALL SET_ADDR_LCD
            LCALL SHOW_RAM
            LCALL KEY_RELEASE

EDIT_PRO_14:      MOV   R5,#90
EDIT_PRO_13:      MOV   R6,#100
EDIT_PRO_12:      MOV   R7,#0E6H
EDIT_PRO_10:            MOV   WDOG,#11000011B
                  DJNZ  R7,EDIT_PRO_11
                  DJNZ  R6,EDIT_PRO_12
                  DJNZ  R5,EDIT_PRO_13
                  LJMP  LOOP
EDIT_PRO_11:      JNB   KEY_DA,EDIT_PRO_10

            LCALL GETKEY_DATA
            MOV   A,KPAD_DATA

EDIT_P_UP:  CJNE  A,#00AH,EDIT_P_DW
            MOV   A,RAM_PTR
            CJNE  A,#000H,EDIT_P_UP1
            CLR   BEEP
            LCALL DELAY_100ms
            SETB  BEEP
            LJMP  EDIT_PRO_14
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
EDIT_P_UP1: DEC    RAM_PTR
            MOV    R5,#010H
EDIT_P_UP2: LCALL  UPLOAD_DP
            MOV    A,DPL
            CJNE   A,#000H,EDIT_P_UP3
            MOV    DPL,#0FFH
            DEC    DPH
            LCALL  SAVE_DP
            DJNZ   R5,EDIT_P_UP2
            LJMP   EDIT_PRO3

EDIT_P_UP3: DEC    DPL
EDIT_P_UP4: LCALL  SAVE_DP
            DJNZ   R5,EDIT_P_UP2
            LJMP   EDIT_PRO3

EDIT_P_DW:  CJNE   A,#00BH,EDIT_PRO_2
EDIT_P_DW0: MOV    A,RAM_PTR
            CJNE   A,#01BH,EDIT_P_DW1
            LJMP   EDIT_PRO2
EDIT_P_DW1: INC    RAM_PTR
            MOV    R5,#008H
EDIT_P_DW2: LCALL  UPLOAD_DP
            MOV    A,DPL
            CJNE   A,#000H,EDIT_P_DW3
            MOV    DPL,#0FFH
            DEC    DPH
            LCALL  SAVE_DP
            DJNZ   R5,EDIT_P_DW2
            LJMP   EDIT_PRO3

EDIT_P_DW3: DEC    DPL
EDIT_P_DW4: LCALL  SAVE_DP
            DJNZ   R5,EDIT_P_DW2
            LJMP   EDIT_PRO3

EDIT_PRO_2: SUBB   A,#004H
            JNC    EDIT_PRO_4

EDIT_B:     MOV    A,KPAD_DATA
            CJNE   A,#004H,EDIT_P_NUM0
EDIT_PRO_4: LJMP   EDIT_PRO_3

EDIT_P_NUM0:       MOV    BUFFER,KPAD_DATA
            LCALL  KEY_RELEASE
EDIT_P_NUM1:       MOV    WDOG,#11000011B
            JNB    KEY_DA,EDIT_P_NUM1
            LCALL  GETKEY_DATA
            MOV    A,KPAD_DATA
EDIT_P_NUM2:       CJNE   A,#00DH,EDIT_P_NUM3
            LJMP   EDIT_PRO1

EDIT_P_NUM3:             CJNE   A,#00AH,EDIT_P_NUM4
EDIT_P_NUMB:       CLR    BEEP
            LCALL  DELAY_100ms
            SETB   BEEP
            LJMP   EDIT_P_NUM1

EDIT_P_NUM4:             SUBB   A,#00AH
            JNC    EDIT_P_NUMB
            MOV    BUFFER+1,KPAD_DATA
            LCALL  KEY_RELEASE
            LCALL  CONV2BCD
            LCALL  GET_ADDR_RAM
            MOV    A,EDIT_PTR
            DEC    A
            MOV    B,#004H
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            MUL   AB
            CJNE  A,#000H,EDIT_ADDR_RAM
            MOV   BUFF_DPL,DPL
            MOV   BUFF_DPH,DPH
            LJMP  EDIT_TITLE22_0

EDIT_ADDR_RAM:    MOV   R5,A
EDIT_ADDR_RAM0:   INC   DPTR
                  DJNZ  R5,EDIT_ADDR_RAM0
            MOV   BUFF_DPL,DPL
            MOV   BUFF_DPH,DPH

EDIT_TITLE22_0:   MOV   A,EDIT_PTR
            SUBB  A,RAM_PTR
            JNC   EDIT_TITLE22_1
            LJMP  EDIT_PRO_33
EDIT_TITLE22_1:   CJNE  A,#001H,EDIT_TITLE22_21
                  LJMP  EDIT_TITLE22_11

EDIT_TITLE22_21:  LJMP  EDIT_TITLE22_2

EDIT_TITLE22_11:  MOV   LCD_ADDR,#040H
            LCALL SET_ADDR_LCD
            MOV   DPTR,#TITLE_22
            LCALL WRLINE_LCD

            MOV   LCD_ADDR,#040H
            LCALL SET_ADDR_LCD
            LCALL BUFFER2ACC
            MOV   LCD_DATA,A
            LCALL BCD2LCD
            MOV   DPL,BUFF_DPL
            MOV   DPH,BUFF_DPH
            MOVX  @DPTR,A
            INC   DPTR
            MOV   BUFF_DPL,DPL
            MOV   BUFF_DPH,DPH

EDIT_HH1:   MOV   LCD_ADDR,#043H
            LCALL SET_ADDR_LCD
            MOV   LCD_DATA,#'h'
            LCALL WRCHAR_LCD
            MOV   LCD_DATA,#'h'
            LCALL WRCHAR_LCD
            MOV   LCD_ADDR,#043H
            LCALL SET_ADDR_LCD
            LCALL LCD_BLINK

EDIT_HH1_1: LCALL PRESS_BCD
            LCALL BUFFER2ACC
            MOV   B,A
            SUBB  A,#024H
            JC    EDIT_HH1_2
            CLR   BEEP
            LCALL DELAY_100ms
            SETB  BEEP
            LJMP  EDIT_HH1

EDIT_HH1_2: MOV   A,B
            MOV   DPL,BUFF_DPL
            MOV   DPH,BUFF_DPH
            MOVX  @DPTR,A
            INC   DPTR
            MOV   BUFF_DPL,DPL
            MOV   BUFF_DPH,DPH

EDIT_MM1:   MOV   LCD_ADDR,#046H
            LCALL SET_ADDR_LCD
```

เอกสารนี้เป็นเอกสารสงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            MOV   LCD_DATA,#'m'
            LCALL WRCHAR_LCD
            MOV   LCD_DATA,#'m'
            LCALL WRCHAR_LCD
            MOV   LCD_ADDR,#046H
            LCALL SET_ADDR_LCD
            LCALL LCD_BLINK

EDIT_MM1_1: LCALL PRESS_BCD
            LCALL BUFFER2ACC
            MOV   B,A
            SUBB  A,#060H
            JC    EDIT_MM1_2
            CLR   BEEP
            LCALL DELAY_100ms
            SETB  BEEP
            LJMP  EDIT_MM1

EDIT_MM1_2: MOV   A,B
            MOV   DPL,BUFF_DPL
            MOV   DPH,BUFF_DPH
            MOVX  @DPTR,A
            INC   DPTR
            MOV   BUFF_DPL,DPL
            MOV   BUFF_DPH,DPH

EDIT_NF1_0: MOV   LCD_ADDR,#049H
            LCALL SET_ADDR_LCD
            LCALL LCD_BLINK
            LCALL KEY_RELEASE

EDIT_NF1:   MOV   WDOG,#11000011B
            JNB   KEY_DA,EDIT_NF1
            LCALL GETKEY_DATA
            MOV   A,KPAD_DATA

            CJNE  A,#00H,EDIT_NF1_1
            MOV   A,#002H
            MOV   LCD_PTR,A
            MOV   DPTR,#LOAD_ON
            LCALL WR3CHAR_LCD
            MOV   A,#00H
            MOV   DPL,BUFF_DPL
            MOV   DPH,BUFF_DPH
            MOVX  @DPTR,A
            LJMP  EDIT_NF1_0

EDIT_NF1_1: CJNE  A,#001H,EDIT_NF1_2
            MOV   A,#001H
            MOV   LCD_PTR,A
            MOV   DPTR,#LOAD_ON
            LCALL WR3CHAR_LCD
            MOV   A,#01H
            MOV   DPL,BUFF_DPL
            MOV   DPH,BUFF_DPH
            MOVX  @DPTR,A
            LJMP  EDIT_NF1_0

EDIT_NF1_2: CJNE  A,#00FH,EDIT_NF1_B
            LCALL KEY_RELEASE
            LCALL LCD_ON
            LJMP  EDIT_PRO_14
EDIT_NF1_B: CLR   BEEP
            LCALL DELAY_100ms
            SETB  BEEP
            LJMP  EDIT_NF1

EDIT_TITLE22_2:   CJNE  A,#002H,EDIT_TITLE22_32
```

เอกสารนี้เป็นเอกสารสงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                LJMP  EDIT_TITLE22_22
EDIT_TITLE22_32:  LJMP  EDIT_TITLE22_3

EDIT_TITLE22_22:  MOV   LCD_ADDR,#014H
            LCALL SET_ADDR_LCD
            MOV   DPTR,#TITLE_22
            LCALL WRLINE_LCD

            MOV   LCD_ADDR,#014H
            LCALL SET_ADDR_LCD
            LCALL BUFFER2ACC
            MOV   LCD_DATA,A
            LCALL BCD2LCD
            MOV   DPL,BUFF_DPL
            MOV   DPH,BUFF_DPH
            MOVX  @DPTR,A
            INC   DPTR
            MOV   BUFF_DPL,DPL
            MOV   BUFF_DPH,DPH

EDIT_HH2:   MOV   LCD_ADDR,#017H
            LCALL SET_ADDR_LCD
            MOV   LCD_DATA,#'h'
            LCALL WRCHAR_LCD
            MOV   LCD_DATA,#'h'
            LCALL WRCHAR_LCD
            MOV   LCD_ADDR,#017H
            LCALL SET_ADDR_LCD
            LCALL LCD_BLINK

EDIT_HH2_1: LCALL PRESS_BCD
            LCALL BUFFER2ACC
            MOV   B,A
            SUBB  A,#024H
            JC    EDIT_HH2_2
            CLR   BEEP
            LCALL DELAY_100ms
            SETB  BEEP
            LJMP  EDIT_HH2

EDIT_HH2_2: MOV   A,B
            MOV   DPL,BUFF_DPL
            MOV   DPH,BUFF_DPH
            MOVX  @DPTR,A
            INC   DPTR
            MOV   BUFF_DPL,DPL
            MOV   BUFF_DPH,DPH

EDIT_MM2:   MOV   LCD_ADDR,#01AH
            LCALL SET_ADDR_LCD
            MOV   LCD_DATA,#'m'
            LCALL WRCHAR_LCD
            MOV   LCD_DATA,#'m'
            LCALL WRCHAR_LCD
            MOV   LCD_ADDR,#01AH
            LCALL SET_ADDR_LCD
            LCALL LCD_BLINK

EDIT_MM2_1: LCALL PRESS_BCD
            LCALL BUFFER2ACC
            MOV   B,A
            SUBB  A,#060H
            JC    EDIT_MM2_2
            CLR   BEEP
            LCALL DELAY_100ms
            SETB  BEEP
            LJMP  EDIT_MM2
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
EDIT_MM2_2: MOV   A,B
            MOV   DPL,BUFF_DPL
            MOV   DPH,BUFF_DPH
            MOVX  @DPTR,A
            INC   DPTR
            MOV   BUFF_DPL,DPL
            MOV   BUFF_DPH,DPH

EDIT_NF2_0: MOV   LCD_ADDR,#01DH
            LCALL SET_ADDR_LCD
            LCALL LCD_BLINK
            LCALL KEY_RELEASE

EDIT_NF2:   MOV   WDOG,#11000011B
            JNB   KEY_DA,EDIT_NF2
            LCALL GETKEY_DATA
            MOV   A,KPAD_DATA

            CJNE  A,#00H,EDIT_NF2_1
            MOV   A,#002H
            MOV   LCD_PTR,A
            MOV   DPTR,#LOAD_ON
            LCALL WR3CHAR_LCD
            MOV   A,#00H
            MOV   DPL,BUFF_DPL
            MOV   DPH,BUFF_DPH
            MOVX  @DPTR,A
            LJMP  EDIT_NF2_0

EDIT_NF2_1: CJNE  A,#001H,EDIT_NF2_2
            MOV   A,#001H
            MOV   LCD_PTR,A
            MOV   DPTR,#LOAD_ON
            LCALL WR3CHAR_LCD
            MOV   A,#01H
            MOV   DPL,BUFF_DPL
            MOV   DPH,BUFF_DPH
            MOVX  @DPTR,A
            LJMP  EDIT_NF2_0

EDIT_NF2_2: CJNE  A,#00FH,EDIT_NF2_B
            LCALL KEY_RELEASE
            LCALL LCD_ON
            LJMP  EDIT_PRO_14
EDIT_NF2_B: CLR   BEEP
            LCALL DELAY_100ms
            SETB  BEEP
            LJMP  EDIT_NF2

EDIT_TITLE22_3:   CJNE  A,#003H,RET2EDIT
            LJMP  EDIT_TITLE22_4
RET2EDIT:   LJMP  EDIT_PRO_33
EDIT_TITLE22_4:   MOV   LCD_ADDR,#054H
            LCALL SET_ADDR_LCD
            MOV   DPTR,#TITLE_22
            LCALL WRLINE_LCD

            MOV   LCD_ADDR,#054H
            LCALL SET_ADDR_LCD
            LCALL BUFFER2ACC
            MOV   LCD_DATA,A
            LCALL BCD2LCD
            MOV   DPL,BUFF_DPL
            MOV   DPH,BUFF_DPH
            MOVX  @DPTR,A
            INC   DPTR
            MOV   BUFF_DPL,DPL
            MOV   BUFF_DPH,DPH
```

เอกสารนี้เป็นเอกสารสงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
EDIT_HH3:    MOV   LCD_ADDR,#057H
             LCALL SET_ADDR_LCD
             MOV   LCD_DATA,#'h'
             LCALL WRCHAR_LCD
             MOV   LCD_DATA,#'h'
             LCALL WRCHAR_LCD
             MOV   LCD_ADDR,#057H
             LCALL SET_ADDR_LCD
             LCALL LCD_BLINK

EDIT_HH3_1:  LCALL PRESS_BCD
             LCALL BUFFER2ACC
             MOV   B,A
             SUBB  A,#024H
             JC    EDIT_HH3_2
             CLR   BEEP
             LCALL DELAY_100ms
             SETB  BEEP
             LJMP  EDIT_HH3

EDIT_HH3_2:  MOV   A,B
             MOV   DPL,BUFF_DPL
             MOV   DPH,BUFF_DPH
             MOVX  @DPTR,A
             INC   DPTR
             MOV   BUFF_DPL,DPL
             MOV   BUFF_DPH,DPH

EDIT_MM3:    MOV   LCD_ADDR,#05AH
             LCALL SET_ADDR_LCD
             MOV   LCD_DATA,#'m'
             LCALL WRCHAR_LCD
             MOV   LCD_DATA,#'m'
             LCALL WRCHAR_LCD
             MOV   LCD_ADDR,#05AH
             LCALL SET_ADDR_LCD
             LCALL LCD_BLINK

EDIT_MM3_1:  LCALL PRESS_BCD
             LCALL BUFFER2ACC
             MOV   B,A
             SUBB  A,#060H
             JC    EDIT_MM3_2
             CLR   BEEP
             LCALL DELAY_100ms
             SETB  BEEP
             LJMP  EDIT_MM3

EDIT_MM3_2:  MOV   A,B
             MOV   DPL,BUFF_DPL
             MOV   DPH,BUFF_DPH
             MOVX  @DPTR,A
             INC   DPTR
             MOV   BUFF_DPL,DPL
             MOV   BUFF_DPH,DPH

EDIT_NF3_0:  MOV   LCD_ADDR,#05DH
             LCALL SET_ADDR_LCD
             LCALL LCD_BLINK
             LCALL KEY_RELEASE

EDIT_NF3:    MOV   WDOG,#11000011B
             JNB   KEY_DA,EDIT_NF3
             LCALL GETKEY_DATA
             MOV   A,KPAD_DATA

             CJNE  A,#00H,EDIT_NF3_1
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งไม่อนุญาตให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            MOV   A,#002H
            MOV   LCD_PTR,A
            MOV   DPTR,#LOAD_ON
            LCALL WR3CHAR_LCD
            MOV   A,#00H
            MOV   DPL,BUFF_DPL
            MOV   DPH,BUFF_DPH
            MOVX  @DPTR,A
            LJMP  EDIT_NF3_0

EDIT_NF3_1: CJNE  A,#001H,EDIT_NF3_2
            MOV   A,#001H
            MOV   LCD_PTR,A
            MOV   DPTR,#LOAD_ON
            LCALL WR3CHAR_LCD
            MOV   A,#01H
            MOV   DPL,BUFF_DPL
            MOV   DPH,BUFF_DPH
            MOVX  @DPTR,A
            LJMP  EDIT_NF3_0

EDIT_NF3_2: CJNE  A,#00FH,EDIT_NF3_B
            LCALL KEY_RELEASE
            LCALL LCD_ON
            LJMP  EDIT_P_DW0
EDIT_NF3_B: CLR   BEEP
            LCALL DELAY_100ms
            SETB  BEEP
            LJMP  EDIT_NF3

EDIT_PRO_3: MOV   A,KPAD_DATA
            CJNE  A,#00DH,EDIT_PRO_33
            LJMP  EDIT_PRO1

EDIT_PRO_33:      CLR   BEEP
            LCALL DELAY_100ms
            SETB  BEEP
            LJMP  EDIT_PRO_14


;************************************************
;FUNCTION CLEAR PROGRAM   SHOW TITLE 2 SELECT TYPE 2 CLEAR RAM
;************************************************
CLEAR_PRO:  LCALL LCD_CLR
            LCALL SHOW_CODE
            LCALL KEY_RELEASE
            LCALL PRESSED_CODE
            MOV   A,BACK_CODE
            CJNE  A,#04H,CLEAR_PRO0
            RET

CLEAR_PRO0: LCALL CH_COD
            MOV   A,ERR_TRUE
            CJNE  A,#0,CLEAR_TRUE
            LCALL CODE_ERR
            AJMP  CLEAR_PRO

CLEAR_TRUE: MOV   WDOG,#11000011B
            LCALL CODE_TRUE

CLEAR_PRO1: MOV   WDOG,#11000011B
            MOV   LCD_ADDR,#000H
            LCALL SET_ADDR_LCD
            MOV   DPTR,#TITLE_20
            LCALL WRLINE_LCD

            MOV   LCD_ADDR,#040H
            LCALL SET_ADDR_LCD
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            LCALL WRLINE_LCD

            MOV   LCD_ADDR,#014H
            LCALL SET_ADDR_LCD
            LCALL WRLINE_LCD

            MOV   LCD_ADDR,#054H
            LCALL SET_ADDR_LCD
            LCALL WRLINE_LCD

CLEAR_PRO_14:     MOV   R5,#90
CLEAR_PRO_13:     MOV   R6,#100
CLEAR_PRO_12:     MOV   R7,#0E6H
CLEAR_PRO_10:     MOV   WDOG,#11000011B
                  DJNZ  R7,CLEAR_PRO_11
                  DJNZ  R6,CLEAR_PRO_12
                  DJNZ  R5,CLEAR_PRO_13
                  LJMP  LOOP
CLEAR_PRO_11:     JNB   KEY_DA,CLEAR_PRO_10

            LCALL GETKEY_DATA
            MOV   A,KPAD_DATA

P_CLR_ALL:  CJNE  A,#01H,CLEAR_PRO_2
            LCALL CLEAR_ALL
            LJMP  CLEAR_PRO1

CLEAR_PRO_2:      CJNE  A,#002H,CLEAR_PRO_3
            LCALL CLEAR_S_LD
            LJMP  CLEAR_PRO1

CLEAR_PRO_3:      CJNE  A,#003H,CLEAR_PRO_4
            LCALL CLEAR_1BY1
            LJMP  CLEAR_PRO1

CLEAR_PRO_4:      CJNE  A,#00DH,CLEAR_PRO_15
            RET
CLEAR_PRO_15:     CLR   BEEP
                  LCALL DELAY_100ms
                  SETB  BEEP
                  LJMP  CLEAR_PRO_14

;*****************************************************
;FUNTION SHOW LCD TO RECEIVE LOAD + DAY TO GET ADDRESS RAM
;*****************************************************
GET_LOAD_DAY:     MOV   WDOG,#11000011B
            LCALL LCD_CLR
            MOV   LCD_ADDR,#000H
            LCALL SET_ADDR_LCD
            MOV   DPTR,#TITLE_15
            LCALL WRLINE_LCD

            MOV   LCD_ADDR,#040H
            LCALL SET_ADDR_LCD
            LCALL WRLINE_LCD

            MOV   LCD_ADDR,#051H
            LCALL SET_ADDR_LCD
            LCALL LCD_BLINK
            LCALL KEY_RELEASE

CHECK_LOAD_4:     MOV   R5,#90
CHECK_LOAD_3:     MOV   R6,#100
CHECK_LOAD_2:     MOV   R7,#0E6H
CHECK_LOAD_0:     MOV   WDOG,#11000011B
                  DJNZ  R7,CHECK_LOAD_1
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                DJNZ  R6,CHECK_LOAD_2
                DJNZ  R5,CHECK_LOAD_3
                LJMP  LOOP
CHECK_LOAD_1:   JNB   KEY_DA,CHECK_LOAD_0

          LCALL GETKEY_DATA
          MOV   A,KPAD_DATA

          CJNE  A,#00H,CHECK_CON
          LJMP  CHECK_LOAD_5
CHECK_CON:CJNE  A,#00DH,CHECK_CON1
          MOV   BACK_CODE,#04H
          RET

CHECK_CON1:MOV  A,KPAD_DATA
          CJNE  A,#009H,CHECK_LOAD1
CHECK_LOAD_5:   CLR   BEEP
          LCALL DELAY_100ms
          SETB  BEEP
          LJMP  CHECK_LOAD_4

CHECK_LOAD1:    SUBB  A,#009H
          JNC   CHECK_LOAD_5
          MOV   A,KPAD_DATA
          MOV   BUFF_LOAD,A
          LCALL KPAD2LCD

          MOV   LCD_ADDR,#014H
          LCALL SET_ADDR_LCD
          MOV   DPTR,#TITLE_17
          LCALL WRLINE_LCD

CHECK_DAY0:MOV  LCD_ADDR,#025H
          LCALL SET_ADDR_LCD
          LCALL LCD_BLINK
          LCALL KEY_RELEASE

CHECK_DAY_4:    MOV   R5,#90
CHECK_DAY_3:    MOV   R6,#100
CHECK_DAY_2:    MOV   R7,#0E6H
CHECK_DAY_0:    MOV   WDOG,#11000011B
                DJNZ  R7,CHECK_DAY_1
                DJNZ  R6,CHECK_DAY_2
                DJNZ  R5,CHECK_DAY_3
                LJMP  LOOP
CHECK_DAY_1:    JNB   KEY_DA,CHECK_DAY_0

          LCALL GETKEY_DATA
          MOV   A,KPAD_DATA
          CJNE  A,#00H,CH_D_CON
CHECK_DAY_5:    CLR   BEEP
          LCALL DELAY_100ms
          SETB  BEEP
          LJMP  CHECK_DAY_4
CH_D_CON:CJNE   A,#00DH,CH_D_CON1
          LJMP  GET_LOAD_DAY

CH_D_CON1:MOV   A,KPAD_DATA
          CJNE  A,#008H,CHECK_DAY2
          LJMP  CHECK_DAY_5

CHECK_DAY2:SUBB A,#008H
          JNC   CHECK_DAY_5
          MOV   A,KPAD_DATA
          MOV   LCD_PTR,A
          MOV   BUFF_DAY,A
          MOV   DPTR,#DAY_SUN
          LCALL WR3CHAR_LCD
```

เอกสารนี้เป็นเอกสารสงวนลิขสิทธิ์ สำหรับใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            LCALL KEY_RELEASE

CHECK_DAY3_4:     MOV   R5,#90
CHECK_DAY3_3:     MOV   R6,#100
CHECK_DAY3_2:     MOV   R7,#0E6H
CHECK_DAY3_0:     MOV   WDOG,#11000011B
                  DJNZ  R7,CHECK_DAY3_1
                  DJNZ  R6,CHECK_DAY3_2
                  DJNZ  R5,CHECK_DAY3_3
                  LJMP  LOOP
CHECK_DAY3_1:     JNB   KEY_DA,CHECK_DAY3_0

            LCALL GETKEY_DATA
            MOV   A,KPAD_DATA
            CJNE  A,#00DH,CHECK_DAY4
            MOV   LCD_ADDR,#025H
            LCALL SET_ADDR_LCD
            MOV   A,#01H
            MOV   LCD_PTR,A
            MOV   DPTR,#DAY_DAY
            LCALL WR3CHAR_LCD
            LJMP  CHECK_DAY0

CHECK_DAY4: CJNE  A,#00FH,CHECK_DAY3_5
            MOV   BACK_CODE,#03H
            RET
CHECK_DAY3_5:     CLR   BEEP
            LCALL DELAY_100ms
            SETB  BEEP
            LJMP  CHECK_DAY3_4

;*************************************************
;FUNCTION SHOW THAT  " PROGRAM LOAD NO X DAY" AFTER KEY LOAD NO+DAY
;*************************************************
TITLE_LOAD_DAY:   LCALL LCD_CLR
            MOV   LCD_ADDR,#000H
            LCALL SET_ADDR_LCD
            MOV   DPTR,#TITLE_18
            LCALL WRLINE_LCD

            MOV   LCD_ADDR,#00FH
            LCALL SET_ADDR_LCD
            MOV   A,BUFF_LOAD
            MOV   KPAD_DATA,A
            LCALL KPAD2LCD

            MOV   LCD_ADDR,#011H
            LCALL SET_ADDR_LCD
            MOV   A,BUFF_DAY
            MOV   LCD_PTR,A
            MOV   DPTR,#DAY_SUN
            LCALL WR3CHAR_LCD
            RET

;*************************************************
;FUNCTION SHOW LCD WHICH DATA FROM RAM
;*************************************************
SHOW_RAM:   LCALL UPLOAD_DP
            MOVX  A,@DPTR

            CJNE  A,#000H,SHOW_RAM0
            MOV   B,A
            INC   DPTR
            INC   DPTR
            INC   DPTR
            INC   DPTR
            LCALL SAVE_DP
            MOV   A,#001H
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            MOV   LCD_PTR,A
            MOV   DPTR,#RAM_END
            LCALL WR3CHAR_LCD
            RET

SHOW_RAM0:  MOV   LCD_DATA,A
            LCALL BCD2LCD

            MOV   LCD_DATA,#' '
            LCALL WRCHAR_LCD

            INC   DPTR
            MOVX  A,@DPTR
            MOV   LCD_DATA,A
            LCALL BCD2LCD

            MOV   LCD_DATA,#':'
            LCALL WRCHAR_LCD

            INC   DPTR
            MOVX  A,@DPTR
            MOV   LCD_DATA,A
            LCALL BCD2LCD

            MOV   LCD_DATA,#' '
            LCALL WRCHAR_LCD

            INC   DPTR
            MOVX  A,@DPTR
            MOV   B,A
            INC   DPTR
            LCALL SAVE_DP
            MOV   A,B

SHOW_ON:    CJNE  A,#01H,SHOW_OFF
            LJMP  SHOW_RAM1
SHOW_OFF:   MOV   A,#02H

SHOW_RAM1:  MOV   LCD_PTR,A
            MOV   DPTR,#LOAD_ON
            LCALL WR3CHAR_LCD
            MOV   WDOG,#11000011B
            RET


;*****************************************************
;FUNCTION GET ADDRESS OF RAM TO CONNECT FROM DAY AND LOAD
;*****************************************************
GET_ADDR_RAM:     MOV   WDOG,#11000011B
            MOV   A,BUFF_DAY
ADDR_SUN:   CJNE  A,#01H,ADDR_MON
            MOV   A,BUFF_LOAD
SUN_LOAD1:  CJNE  A,#01H,SUN_LOAD2
            MOV   DPTR,#0000H
            RET
SUN_LOAD2:  CJNE  A,#02H,SUN_LOAD3
            MOV   DPTR,#0078H
            RET
SUN_LOAD3:  CJNE  A,#03H,SUN_LOAD4
            MOV   DPTR,#00F0H
            RET
SUN_LOAD4:  CJNE  A,#04H,SUN_LOAD5
            MOV   DPTR,#0168H
            RET
SUN_LOAD5:  CJNE  A,#05H,SUN_LOAD6
            MOV   DPTR,#01E0H
            RET
SUN_LOAD6:  CJNE  A,#06H,SUN_LOAD7
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            MOV   DPTR,#0258H
            RET
SUN_LOAD7:  CJNE  A,#07H,SUN_LOAD8
            MOV   DPTR,#02D0H
            RET
SUN_LOAD8:  MOV   DPTR,#0348H
            RET

ADDR_MON:   CJNE  A,#02H,ADDR_TUE
            MOV   A,BUFF_LOAD
MON_LOAD1:  CJNE  A,#01H,MON_LOAD2
            MOV   DPTR,#03E8H
            RET
MON_LOAD2:  CJNE  A,#02H,MON_LOAD3
            MOV   DPTR,#0460H
            RET
MON_LOAD3:  CJNE  A,#03H,MON_LOAD4
            MOV   DPTR,#04D8H
            RET
MON_LOAD4:  CJNE  A,#04H,MON_LOAD5
            MOV   DPTR,#0550H
            RET
MON_LOAD5:  CJNE  A,#05H,MON_LOAD6
            MOV   DPTR,#05C8H
            RET
MON_LOAD6:  CJNE  A,#06H,MON_LOAD7
            MOV   DPTR,#00640H
            RET
MON_LOAD7:  CJNE  A,#07H,MON_LOAD8
            MOV   DPTR,#06B8H
            RET
MON_LOAD8:  MOV   DPTR,#0730H
            RET

ADDR_TUE:   CJNE  A,#03H,ADDR_WED
            MOV   A,BUFF_LOAD
TUE_LOAD1:  CJNE  A,#01H,TUE_LOAD2
            MOV   DPTR,#07D0H
            RET
TUE_LOAD2:  CJNE  A,#02H,TUE_LOAD3
            MOV   DPTR,#0848H
            RET
TUE_LOAD3:  CJNE  A,#03H,TUE_LOAD4
            MOV   DPTR,#08C0H
            RET
TUE_LOAD4:  CJNE  A,#04H,TUE_LOAD5
            MOV   DPTR,#0938H
            RET
TUE_LOAD5:  CJNE  A,#05H,TUE_LOAD6
            MOV   DPTR,#09B0H
            RET
TUE_LOAD6:  CJNE  A,#06H,TUE_LOAD7
            MOV   DPTR,#0A28H
            RET
TUE_LOAD7:  CJNE  A,#07H,TUE_LOAD8
            MOV   DPTR,#0AA0H
            RET
TUE_LOAD8:  MOV   DPTR,#0B18H
            RET

ADDR_WED:   CJNE  A,#04H,ADDR_THU
            MOV   A,BUFF_LOAD
WED_LOAD1:  CJNE  A,#01H,WED_LOAD2
            MOV   DPTR,#0BB8H
            RET
WED_LOAD2:  CJNE  A,#02H,WED_LOAD3
            MOV   DPTR,#0C30H
            RET
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
WED_LOAD3:  CJNE  A,#03H,WED_LOAD4
            MOV   DPTR,#0CA8H
            RET
WED_LOAD4:  CJNE  A,#04H,WED_LOAD5
            MOV   DPTR,#0D20H
            RET
WED_LOAD5:  CJNE  A,#05H,WED_LOAD6
            MOV   DPTR,#0D98H
            RET
WED_LOAD6:  CJNE  A,#06H,WED_LOAD7
            MOV   DPTR,#0E10H
            RET
WED_LOAD7:  CJNE  A,#07H,WED_LOAD8
            MOV   DPTR,#0E88H
            RET
WED_LOAD8:  MOV   DPTR,#0F00H
            RET

ADDR_THU:   CJNE  A,#05H,ADDR_FRI
            MOV   A,BUFF_LOAD
THU_LOAD1:  CJNE  A,#01H,THU_LOAD2
            MOV   DPTR,#0FA0H
            RET
THU_LOAD2:  CJNE  A,#02H,THU_LOAD3
            MOV   DPTR,#1018H
            RET
THU_LOAD3:  CJNE  A,#03H,THU_LOAD4
            MOV   DPTR,#1090H
            RET
THU_LOAD4:  CJNE  A,#04H,THU_LOAD5
            MOV   DPTR,#1108H
            RET
THU_LOAD5:  CJNE  A,#05H,THU_LOAD6
            MOV   DPTR,#1180H
            RET
THU_LOAD6:  CJNE  A,#06H,THU_LOAD7
            MOV   DPTR,#11F8H
            RET
THU_LOAD7:  CJNE  A,#07H,THU_LOAD8
            MOV   DPTR,#1270H
            RET
THU_LOAD8:  MOV   DPTR,#12E8H
            RET

ADDR_FRI:   CJNE  A,#06H,ADDR_SAT
            MOV   A,BUFF_LOAD
FRI_LOAD1:  CJNE  A,#01H,FRI_LOAD2
            MOV   DPTR,#1388H
            RET
FRI_LOAD2:  CJNE  A,#02H,FRI_LOAD3
            MOV   DPTR,#1400H
            RET
FRI_LOAD3:  CJNE  A,#03H,FRI_LOAD4
            MOV   DPTR,#1478H
            RET
FRI_LOAD4:  CJNE  A,#04H,FRI_LOAD5
            MOV   DPTR,#14F0H
            RET
FRI_LOAD5:  CJNE  A,#05H,FRI_LOAD6
            MOV   DPTR,#1568H
            RET
FRI_LOAD6:  CJNE  A,#06H,FRI_LOAD7
            MOV   DPTR,#15E0H
            RET
FRI_LOAD7:  CJNE  A,#07H,FRI_LOAD8
            MOV   DPTR,#1658H
            RET
FRI_LOAD8:  MOV   DPTR,#17D0H
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            RET

ADDR_SAT:   MOV   A,BUFF_LOAD
SAT_LOAD1:  CJNE  A,#01H,SAT_LOAD2
            MOV   DPTR,#1770H
            RET
SAT_LOAD2:  CJNE  A,#02H,SAT_LOAD3
            MOV   DPTR,#17E8H
            RET
SAT_LOAD3:  CJNE  A,#03H,SAT_LOAD4
            MOV   DPTR,#1860H
            RET
SAT_LOAD4:  CJNE  A,#04H,SAT_LOAD5
            MOV   DPTR,#18D8H
            RET
SAT_LOAD5:  CJNE  A,#05H,SAT_LOAD6
            MOV   DPTR,#1950H
            RET
SAT_LOAD6:  CJNE  A,#06H,SAT_LOAD7
            MOV   DPTR,#19C8H
            RET
SAT_LOAD7:  CJNE  A,#07H,SAT_LOAD8
            MOV   DPTR,#1A40H
            RET
SAT_LOAD8:  MOV   DPTR,#1AB8H
            RET

;************************************************
;FUNCTION SAVE ADDRESS OF DPTR FOR CONNECT RAM
;************************************************
SAVE_DP:    MOV   A,DPL
            MOV   LOW_DP,A
            MOV   A,DPH
            MOV   HIGH_DP,A
            RET

;************************************************
;FUNCTION LOAD ADDRESS OF DPTR FOR CONNECT RAM
;************************************************
UPLOAD_DP:  MOV   A,LOW_DP
            MOV   DPL,A
            MOV   A,HIGH_DP
            MOV   DPH,A
            RET

;************************************************
;FUNCTION SHOW LCD WAIT FOR PRESS CODE
;************************************************
SHOW_CODE:  LCALL LCD_CLR
            MOV   LCD_ADDR,#040H
            LCALL SET_ADDR_LCD
            MOV   DPTR,#TITLE_4
            LCALL WRLINE_LCD

            MOV   LCD_ADDR,#04EH
            LCALL SET_ADDR_LCD
            LCALL LCD_BLINK
            RET

;************************************************
;FUNCTION GET KEYPAD_DATA AFTER KEY PRESSED
;************************************************
GETKEY_DATA:MOV   A,P1
            ANL   A,#00001111B
            MOV   DPTR,#CONV_KEY
            MOVC  A,@A+DPTR
            ANL   A,#00001111B
            MOV   KPAD_DATA,A
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            RET

;*****************************************************
;FUNCTION CHECK RELEASE KEY OR NOT ?
;*****************************************************
KEY_RELEASE:      MOV   WDOG,#11100011B
            JB    KEY_DA,KEY_RELEASE
            RET

;*****************************************************
;FUNCTION SET DAY DATE MONTH YEAR
;*****************************************************

SET_DATE:   MOV   WDOG,#11000011B
            LCALL LCD_CLR
            MOV   LCD_ADDR,#040H
            LCALL SET_ADDR_LCD
            MOV   DPTR,#SCR_SET_DATE
            LCALL WRLINE_LCD
            LCALL DELAY_1s
            LCALL KEY_RELEASE

INT_P_DAY:  MOV   WDOG,#11000011B
            MOV   LCD_ADDR,#044H
            LCALL SET_ADDR_LCD
            MOV   A,#01H
            MOV   LCD_PTR,A
            MOV   DPTR,#DAY_DAY
            LCALL WR3CHAR_LCD
            MOV   LCD_ADDR,#044H
            LCALL SET_ADDR_LCD
            LCALL LCD_BLINK
            LCALL KEY_RELEASE

PRESS_DAY:  MOV   WDOG,#11000011B
            JNB   KEY_DA,PRESS_DAY
            LCALL GETKEY_DATA
            MOV   A,KPAD_DATA
            CJNE  A,#00H,DAY_CON
            LJMP  DAY_CON11

DAY_CON:    CJNE  A,#00DH,DAY_CON1
            LJMP  TIME_DATE

DAY_CON1:   MOV   A,KPAD_DATA

            CJNE  A,#008H,PRESS_DAY2
DAY_CON11:  CLR   BEEP
            LCALL DELAY_100ms
            SETB  BEEP
            LJMP  PRESS_DAY

PRESS_DAY2: SUBB  A,#008H
            JNC   DAY_CON11

            MOV   A,KPAD_DATA
            MOV   LCD_PTR,A
            MOV   DPTR,#DAY_SUN
            LCALL WR3CHAR_LCD
            MOV   DAY,KPAD_DATA

INT_P_DATE: MOV   WDOG,#11000011B
            MOV   LCD_ADDR,#049H
            LCALL SET_ADDR_LCD
            MOV   LCD_DATA,#'d'
            LCALL WRCHAR_LCD
            MOV   LCD_DATA,#'d'
            LCALL WRCHAR_LCD
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            MOV   LCD_ADDR,#049H
            LCALL SET_ADDR_LCD
            LCALL LCD_BLINK
            LCALL KEY_RELEASE

PRESS_DATE: MOV   WDOG,#11000011B
            JNB   KEY_DA,PRESS_DATE
            LCALL GETKEY_DATA
            MOV   A,KPAD_DATA
            CJNE  A,#00DH,DATE_CON
            LJMP  INT_P_DAY
DATE_CON:   MOV   A,KPAD_DATA

            CJNE  A,#004H,PRESS_DATE1
DATE_CON_B: CLR   BEEP
            LCALL DELAY_100ms
            SETB  BEEP
            LJMP  INT_P_DATE

PRESS_DATE1:      SUBB  A,#004H
            JNC   DATE_CON_B

            MOV   BUFFER,KPAD_DATA
            LCALL KPAD2LCD
            LCALL LCD_BLINK
            LCALL KEY_RELEASE

PRESS_DATE_1:     MOV   WDOG,#11000011B
            JNB   KEY_DA,PRESS_DATE_1
            LCALL GETKEY_DATA
            MOV   A,KPAD_DATA

            CJNE  A,#00AH,PRESS_DATE2
DATE_CON_B1:      CLR   BEEP
            LCALL DELAY_100ms
            SETB  BEEP
            LJMP  PRESS_DATE_1

PRESS_DATE2:      SUBB  A,#00AH
            JNC   DATE_CON_B1

            MOV   BUFFER+1,KPAD_DATA
            LCALL KPAD2LCD
            LCALL KEY_RELEASE

            LCALL BUFFER2ACC
            MOV   B,A
            CJNE  A,#000H,PRESS_DATE3
            LJMP  INT_P_DATE
PRESS_DATE3:      SUBB  A,#032H
            JNC   DATE_CON_B

            MOV   A,B
            MOV   DATE,A

INT_P_MONTH:      MOV   WDOG,#11000011B
            MOV   LCD_ADDR,#04CH
            LCALL SET_ADDR_LCD
            MOV   LCD_DATA,#'m'
            LCALL WRCHAR_LCD
            MOV   LCD_DATA,#'m'
            LCALL WRCHAR_LCD
            MOV   LCD_ADDR,#04CH
            LCALL SET_ADDR_LCD
            LCALL LCD_BLINK
            LCALL KEY_RELEASE

PRESS_MONTH:      MOV   WDOG,#11000011B
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            JNB   KEY_DA,PRESS_MONTH
            LCALL GETKEY_DATA
            MOV   A,KPAD_DATA
            CJNE  A,#00DH,MONTH_CON
            LJMP  INT_P_DATE

MONTH_CON:  LCALL PRESS_BCD
            MOV   WDOG,#11000011B
            LCALL BUFFER2ACC
            MOV   B,A

            CJNE  A,#000H,PRESS_MONTH3
MONTH_CON_B:      CLR   BEEP
            LCALL DELAY_100ms
            SETB  BEEP
            LJMP  INT_P_MONTH

PRESS_MONTH3:     SUBB  A,#013H
            JNC   MONTH_CON_B

            MOV   A,B
            MOV   MONTH,A

INT_P_YEAR: MOV   LCD_ADDR,#04FH
            LCALL SET_ADDR_LCD
            LCALL LCD_BLINK
            LCALL KEY_RELEASE

PRESS_YEAR: MOV   WDOG,#11000011B
            JNB   KEY_DA,PRESS_YEAR
            LCALL GETKEY_DATA
            MOV   A,KPAD_DATA
            CJNE  A,#00DH,YEAR_CON
            LJMP  INT_P_MONTH

YEAR_CON:   LCALL PRESS_BCD
            LCALL BUFFER2ACC
            MOV   YEAR,A


DATE_P_ENTER:     MOV   WDOG,#11000011B
            JNB   KEY_DA,DATE_P_ENTER
            LCALL GETKEY_DATA
            MOV   A,KPAD_DATA

            CJNE  A,#00DH,DATE_P_ENTER1
            LJMP  INT_P_YEAR

DATE_P_ENTER1:    CJNE  A,#00FH,DATE_P_ENTER2
            LCALL RTC_WR
            LCALL KEY_RELEASE
            RET
DATE_P_ENTER2:    CLR   BEEP
            LCALL DELAY_100ms
            SETB  BEEP
            LJMP  DATE_P_ENTER

;*****************************************************
;FUNCTION SET TIME ( HOURS MINUTES SECONDS )
;*****************************************************
SET_TIME:   MOV   WDOG,#11000011B
SET_TIME0:  LCALL LCD_CLR
            MOV   LCD_ADDR,#040H
            LCALL SET_ADDR_LCD
            MOV   DPTR,#SCR_SET_TIME
            LCALL WRLINE_LCD

INT_P_HOURS:MOV   WDOG,#11000011B
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            MOV   LCD_ADDR,#04AH
            LCALL SET_ADDR_LCD
            MOV   LCD_DATA,#'h'
            LCALL WRCHAR_LCD
            MOV   LCD_DATA,#'h'
            LCALL WRCHAR_LCD
            MOV   LCD_ADDR,#04AH
            LCALL SET_ADDR_LCD
            LCALL LCD_BLINK
            LCALL KEY_RELEASE

PRESS_HOURS:      MOV   WDOG,#11000011B
            JNB   KEY_DA,PRESS_HOURS
            LCALL GETKEY_DATA
            MOV   A,KPAD_DATA
            CJNE  A,#00DH,HOURS_CON
            LJMP  TIME_DATE
HOURS_CON:  MOV   A,KPAD_DATA

            CJNE  A,#003H,PRESS_HOURS1
HOURS_CON_B:      CLR   BEEP
            LCALL DELAY_100ms
            SETB  BEEP
            LJMP  INT_P_HOURS

PRESS_HOURS1:     SUBB  A,#003H
            JNC   HOURS_CON_B

            MOV   BUFFER,KPAD_DATA
            LCALL KPAD2LCD
            LCALL LCD_BLINK
            LCALL KEY_RELEASE

PRESS_HOURS_1:    MOV   WDOG,#11000011B
            JNB   KEY_DA,PRESS_HOURS_1
            LCALL GETKEY_DATA
            MOV   A,KPAD_DATA

            CJNE  A,#00AH,PRESS_HOURS2
HOURS_CON_B1:     CLR   BEEP
            LCALL DELAY_100ms
            SETB  BEEP
            LJMP  PRESS_HOURS_1

PRESS_HOURS2:     SUBB  A,#00AH
            JNC   HOURS_CON_B1

            MOV   BUFFER+1,KPAD_DATA
            LCALL KPAD2LCD
            LCALL LCD_BLINK
            LCALL KEY_RELEASE

            LCALL BUFFER2ACC
            MOV   B,A
            SUBB  A,#024H
            JNC   HOURS_CON_B

            MOV   A,B
            MOV   HOURS,A

INT_P_MIN:  MOV   WDOG,#11000011B
            MOV   LCD_ADDR,#04DH
            LCALL SET_ADDR_LCD
            MOV   LCD_DATA,#'m'
            LCALL WRCHAR_LCD
            MOV   LCD_DATA,#'m'
            LCALL WRCHAR_LCD
            MOV   LCD_ADDR,#04DH
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งไม่อนุญาตให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            LCALL SET_ADDR_LCD
            LCALL LCD_BLINK
            LCALL KEY_RELEASE

PRESS_MIN:  MOV   WDOG,#11000011B
            LCALL PRESS_BCD

            LCALL BUFFER2ACC
            MOV   B,A
            SUBB  A,#060H
            JC    PRESS_MIN11
MIN_CON:    CLR   BEEP
            LCALL DELAY_100ms
            SETB  BEEP
            LJMP  INT_P_MIN

PRESS_MIN11:      MOV   A,B
            MOV   MINUTES,A

INT_P_SEC:  MOV   WDOG,#11000011B
            MOV   LCD_ADDR,#050H
            LCALL SET_ADDR_LCD
            MOV   LCD_DATA,#'s'
            LCALL WRCHAR_LCD
            MOV   LCD_DATA,#'s'
            LCALL WRCHAR_LCD
            MOV   LCD_ADDR,#050H
            LCALL SET_ADDR_LCD
            LCALL LCD_BLINK
            LCALL KEY_RELEASE

PRESS_SEC:  MOV   WDOG,#11000011B
            LCALL PRESS_BCD
            LCALL BUFFER2ACC
            MOV   B,A
            SUBB  A,#060H
            JC    PRESS_SEC11
            CLR   BEEP
            LCALL DELAY_100ms
            SETB  BEEP
            LJMP  INT_P_SEC

PRESS_SEC11:      MOV   A,B
            MOV   SECONDS,A


TIME_P_ENTER:     MOV   WDOG,#11000011B
            JNB   KEY_DA,TIME_P_ENTER
            LCALL GETKEY_DATA
            MOV   A,KPAD_DATA

            CJNE  A,#00DH,TIME_P_ENTER1
            LJMP  SET_TIME0

TIME_P_ENTER1:    CJNE  A,#00FH,TIME_P_ENTER2
            LCALL RTC_WR
            LCALL KEY_RELEASE
            RET
TIME_P_ENTER2:    CLR   BEEP
            LCALL DELAY_100ms
            SETB  BEEP
            LJMP  TIME_P_ENTER
;*****************************************************
;FUNCTION SHOW LCD DAY+DATE AND CLOCK
;*****************************************************
SHOW_CLOCK: LCALL RTC_RD
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            MOV   LCD_ADDR,#043H
            LCALL SET_ADDR_LCD
            MOV   A,DAY
            MOV   LCD_PTR,A
            MOV   DPTR,#DAY_SUN
            LCALL WR3CHAR_LCD

             MOV  LCD_DATA,#' '
            LCALL WRCHAR_LCD

            MOV   LCD_DATA,DATE
            LCALL BCD2LCD

            MOV   LCD_DATA,#' '
            LCALL WRCHAR_LCD

            MOV   A,MONTH
            CJNE  A,#010H,WR_CHK_MONTH_1
            MOV   A,#00AH
            LJMP  WRITE_MONTH_NX
WR_CHK_MONTH_1:   CJNE  A,#011H,WR_CHK_MONTH_2
            MOV   A,#00BH
            LJMP  WRITE_MONTH_NX
WR_CHK_MONTH_2:   CJNE  A,#012H,WRITE_MONTH_NX
            MOV   A,#00CH
WRITE_MONTH_NX:   MOV   LCD_PTR,A
            MOV   DPTR,#MONTH_JAN
            LCALL WR3CHAR_LCD

            MOV   LCD_DATA,#' '
            LCALL WRCHAR_LCD

            MOV   LCD_DATA,YEAR
            LCALL BCD2LCD

            MOV   LCD_ADDR,#01AH
            LCALL SET_ADDR_LCD
            MOV   A,HOURS
            ANL   A,#00110000B
            JZ    WRITE_TIME_HN
            SWAP  A
            ADD   A,#030H
            LJMP  WRITE_TIME_HH
WRITE_TIME_HN:    MOV   A,#' '
WRITE_TIME_HH:    MOV   LCD_DATA,A
            LCALL WRCHAR_LCD

            MOV   A,HOURS
            ANL   A,#00001111B
            ADD   A,#030H
            MOV   LCD_DATA,A
            LCALL WRCHAR_LCD

            MOV   LCD_DATA,#':'
            LCALL WRCHAR_LCD

WRITE_MINUTES:    MOV   LCD_DATA,MINUTES
            LCALL       BCD2LCD

            MOV   LCD_DATA,#':'
            LCALL WRCHAR_LCD

            MOV   LCD_DATA,SECONDS
            LCALL BCD2LCD
            RET

;*************************************************
;PF8574 READ
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
;*****************************************************
PCF8574_RD: MOV   A,PCF8574_ID
            INC   A
            MOV   I2C_ADDR,A
            LCALL I2C_SLAVE

            LCALL I2C_DATA_RD
            MOV   IO_DATA,I2C_DATA
            LCALL I2C_NACK_BIT

            LCALL I2C_STOP
            RET

;*****************************************************
;I2C PCF8574 WRITE
;*****************************************************
PCF8574_WR: MOV   I2C_ADDR,PCF8574_ID
            LCALL I2C_SLAVE

            MOV   I2C_DATA,IO_DATA
            LCALL I2C_DATA_WR

            LCALL I2C_STOP
            RET

;*****************************************************
;I2C DATA WRITE
;*****************************************************
I2C_DATA_WR:      PUSH  ACC
            SETB  I2C_ACK
            MOV   A,I2C_DATA
            MOV   R5,#008
I2C_DATA_WR_1:    RLC   A
            MOV   SDA,C
            LCALL I2C_CLK
            DJNZ  R5,I2C_DATA_WR_1
            SETB  SDA
            LCALL I2C_DELAY
            SETB  SCL
            LCALL I2C_DELAY
            JB    SDA,I2C_DATA_WR_2
            CLR   I2C_ACK
I2C_DATA_WR_2:    CLR   SCL
            POP   ACC
            RET

;*****************************************************
;I2C SLAVE CONNECT
;*****************************************************
I2C_SLAVE:  PUSH  ACC
            SETB  I2C_ACK
            MOV   A,I2C_ADDR
            LCALL I2C_START
            MOV   R5,#008
I2C_SLAVE_1:      RLC   A
            MOV   SDA,C
            LCALL I2C_CLK
            DJNZ  R5,I2C_SLAVE_1
            SETB  SDA
            LCALL I2C_DELAY
            SETB  SCL
            LCALL I2C_DELAY
            JB    SDA,I2C_SLAVE_2
            CLR   I2C_ACK
I2C_SLAVE_2:      CLR   SCL
            POP   ACC
            RET
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
;*************************************************
;I2C DATA READ
;*************************************************
I2C_DATA_RD:      PUSH  ACC
            CLR   A
            MOV   R5,#008
I2C_DATA_RD_1:    LCALL I2C_DELAY
            SETB  SCL
            LCALL I2C_DELAY
            MOV   C,SDA
            RLC   A
            CLR   SCL
            DJNZ  R5,I2C_DATA_RD_1
            MOV   I2C_DATA,A
            POP   ACC
            RET

;*************************************************
;I2C START CONDITION
;*************************************************
I2C_START:  SETB  SCL
            SETB  SDA
            LCALL I2C_DELAY
            CLR   SDA
            LCALL I2C_DELAY
            CLR   SCL
            RET

;*************************************************
;I2C STOP CONDITION
;*************************************************
I2C_STOP:   CLR   SDA
            LCALL I2C_DELAY
            SETB  SCL
            LCALL I2C_DELAY
            SETB  SDA
            RET

;*************************************************
;I2C CLOCK
;*************************************************
I2C_CLK:    LCALL I2C_DELAY
            SETB  SCL
            LCALL I2C_DELAY
            CLR   SCL
            RET

;*************************************************
;I2C NOT ACKNOWLEDGE
;*************************************************
I2C_NACK_BIT:     SETB  SDA
            LCALL I2C_DELAY
            LCALL I2C_CLK
            RET

;*************************************************
;I2C ACKNOWLEDGE
;*************************************************
I2C_ACK_BIT:      CLR   SDA
            LCALL I2C_DELAY
            LCALL I2C_CLK
            SETB  SDA
            RET

;*************************************************
;  LCD  INITIALIZE
;*************************************************
INIT_LCD:   LCALL DELAY_100ms
```

เอกสารนี้เป็นเอกสารสงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            CLR   LCD_RS

            MOV   IO_DATA,#00111000B            ;  FUNCTION SET
            MOV   PCF8574_ID,#PCF8574_1
            LCALL PCF8574_WR
            LCALL LCD_CLK
            LCALL DELAY_10ms

            MOV   IO_DATA,#00111000B            ;  FUNCTION SET AGAIN
            LCALL PCF8574_WR
            LCALL LCD_CLK

            LCALL LCD_OFF                   ;  DISPLAY OFF

            LCALL LCD_CLR                   ;  CLEAR DISPLAY

            MOV   IO_DATA,#00000110B            ;  ENTRY MODE
            LCALL PCF8574_WR
            LCALL LCD_CLK

            LCALL LCD_HOME                  ;  RETURN HOME
            RET

;*********************************
;LCD CLEAR DISPLAY
;*********************************
LCD_CLR:    CLR   LCD_RS
            MOV   IO_DATA,#00000001B
            MOV   PCF8574_ID,#PCF8574_1
            LCALL PCF8574_WR
            LCALL LCD_CLK
            RET

;*********************************
;LCD RETURN HOME
;*********************************
LCD_HOME:   CLR   LCD_RS
            MOV   IO_DATA,#00000010B
            MOV   PCF8574_ID,#PCF8574_1
            LCALL PCF8574_WR
            LCALL LCD_CLK
            RET

;*********************************
;LCD DISPLAY OFF
;*********************************
LCD_OFF:    CLR   LCD_RS
            MOV   IO_DATA,#00001000B
            MOV   PCF8574_ID,#PCF8574_1
            LCALL PCF8574_WR
            LCALL LCD_CLK
            RET

;*********************************
;LCD CLOCK
;*********************************
LCD_CLK:    SETB  LCD_EN
            LCALL LCD_DELAY
            CLR   LCD_EN
            LCALL LCD_DELAY
            RET

;*********************************
;LCD  DISPLAY ON
;*********************************
LCD_ON:     CLR   LCD_RS
            MOV   IO_DATA,#00001100B
            MOV   PCF8574_ID,#PCF8574_1
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ธนาคารจุฬาลงกรณ์ ขอสงวนสิทธิ์ในการฟ้องร้องดำเนินคดีตามกฎหมายกับผู้ละเมิดเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            LCALL PCF8574_WR
            LCALL LCD_CLK
            RET

;*******************************
;LCD CURSOR ON
;*******************************
LCD_BLINK:  CLR   LCD_RS
            MOV   IO_DATA,#00001111B
            MOV   PCF8574_ID,#PCF8574_1
            LCALL PCF8574_WR
            LCALL LCD_CLK
            RET

;*******************************
;SET LCD ADDRESS
;I/P: LCD_ADDR
;*******************************
SET_ADDR_LCD:     CLR   LCD_RS
            MOV   A,LCD_ADDR
            SETB  ACC.7
            MOV   IO_DATA,A
            MOV   PCF8574_ID,#PCF8574_1
            LCALL PCF8574_WR
            LCALL       LCD_CLK
            RET

;*************************************************
;WRITE 1 LINE OF 20 CHARACTER FROM ROM
;I/P:       DPTR: LOCATE ROM ADDRESS
;*************************************************
WRLINE_LCD: MOV   R0,#0                ;  CLEAR COUNTER
WRLINE_LCD_1:     SETB  LCD_RS
            CLR   A
            MOVC  A,@A+DPTR
            MOV   IO_DATA,A
            MOV   PCF8574_ID,#PCF8574_1
            LCALL PCF8574_WR
            LCALL LCD_CLK
            INC   DPTR
            INC   R0
            CJNE  R0,#20,WRLINE_LCD_1
            LCALL LCD_ON
            RET

;************************************************
;WRITE CHARACTER TO SHOW LCD
;************************************************
WRCHAR_LCD: SETB  LCD_RS
            MOV   IO_DATA,LCD_DATA
            MOV   PCF8574_ID,#PCF8574_1
            LCALL PCF8574_WR
            LCALL LCD_CLK
            LCALL LCD_ON
            RET

;************************************************
;WRITE 3 CHARACTER FROM ROM
;************************************************
WR3CHAR_LCD:      MOV   R0,#0
            MOV   A,LCD_PTR
            DEC   A
            MOV   B,#3
            MUL   AB
WR3CHAR_LCD_1:    SETB  LCD_RS
            MOVC  A,@A+DPTR
            MOV   IO_DATA,A
            MOV   PCF8574_ID,#PCF8574_1
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งไม่อนุญาตให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            LCALL PCF8574_WR
            LCALL LCD_CLK
            INC   DPTR
            INC   R0
            MOV   A,LCD_PTR
            DEC   A
            MOV   B,#3
            MUL   AB
            CJNE  R0,#3,WR3CHAR_LCD_1
            LCALL LCD_ON
            RET

;*************************************************
;BCD CODE TO SHOW LCD
;*************************************************
BCD2LCD:    PUSH  ACC
            PUSH  B
            MOV   A,LCD_DATA
            MOV   B,A
            ANL   A,#11110000B
            SWAP  A
            ADD   A,#030H
            MOV   LCD_DATA,A
            LCALL WRCHAR_LCD
            MOV   A,B
            ANL   A,#00001111B
            ADD   A,#030H
            MOV   LCD_DATA,A
            LCALL WRCHAR_LCD
            POP   B
            POP   ACC
            RET

;*************************************************
;KEYPAD TO SHOW LCD
;*************************************************
KPAD2LCD:   MOV   A,KPAD_DATA
            ADD   A,#030H
            MOV   LCD_DATA,A
            LCALL WRCHAR_LCD
            RET

;*************************************************
;I2C RTC READ
;*************************************************
RTC_RD:     MOV   I2C_ADDR,#RTC_ID
            LCALL I2C_SLAVE

            MOV   I2C_DATA,#000H
            LCALL I2C_DATA_WR

            MOV   I2C_ADDR,#RTC_ID+1
            LCALL I2C_SLAVE
            LCALL I2C_DATA_RD

            MOV   SECONDS,I2C_DATA
            LCALL       I2C_ACK_BIT

            LCALL       I2C_DATA_RD
            MOV   MINUTES,I2C_DATA
            LCALL I2C_ACK_BIT

            LCALL       I2C_DATA_RD
            MOV   HOURS,I2C_DATA
            LCALL I2C_ACK_BIT

            LCALL       I2C_DATA_RD
            MOV   DAY,I2C_DATA
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            LCALL I2C_ACK_BIT

            LCALL       I2C_DATA_RD
            MOV   DATE,I2C_DATA
            LCALL I2C_ACK_BIT

            LCALL       I2C_DATA_RD
            MOV   MONTH,I2C_DATA
            LCALL I2C_ACK_BIT

            LCALL       I2C_DATA_RD
            MOV   YEAR,I2C_DATA
            LCALL I2C_ACK_BIT

            LCALL       I2C_DATA_RD
            MOV   CONTROL,I2C_DATA
            LCALL I2C_NACK_BIT

            LCALL I2C_STOP
            RET

;*****************************************************
;I2C RTC WRITE
;*****************************************************
RTC_WR:     MOV   I2C_ADDR,#RTC_ID
            LCALL I2C_SLAVE

            MOV   I2C_DATA,#000H
            LCALL I2C_DATA_WR

            MOV   I2C_DATA,SECONDS
            LCALL I2C_DATA_WR

            MOV   I2C_DATA,MINUTES
            LCALL I2C_DATA_WR

            MOV   I2C_DATA,HOURS
            LCALL I2C_DATA_WR

            MOV   I2C_DATA,DAY
            LCALL I2C_DATA_WR

            MOV   I2C_DATA,DATE
            LCALL I2C_DATA_WR

            MOV   I2C_DATA,MONTH
            LCALL I2C_DATA_WR

            MOV   I2C_DATA,YEAR
            LCALL I2C_DATA_WR

            MOV   I2C_DATA,CONTROL
            LCALL I2C_DATA_WR

            LCALL I2C_STOP
            RET

;*****************************************************
;PRESSED CODE AND STORE IN BUFFER
;*****************************************************

PRESSED_CODE:     MOV   WDOG,#11000011B
                  NOP
PRESSED_CODE0_4:  MOV   R5,#90
PRESSED_CODE0_3:  MOV   R6,#100
PRESSED_CODE0_2:  MOV   R7,#0E6H
PRESSED_CODE0:    MOV   WDOG,#11000011B
                  DJNZ  R7,PRESSED_CODE0_1
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่อนุญาตให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                DJNZ  R6,PRESSED_CODE0_2
                DJNZ  R5,PRESSED_CODE0_3
                LJMP  LOOP
PRESSED_CODE0_1:  JNB   KEY_DA,PRESSED_CODE0

            LCALL GETKEY_DATA
            MOV   A,KPAD_DATA
            CJNE  A,#00DH,PRESSED_CODE1
            MOV   A,BACK_CODE
            CJNE  A,#001H,CHECK_BC_1
            LJMP  FIRST_MENU1
CHECK_BC_1: MOV   WDOG,#11000011B
            CJNE  A,#002H,CHECK_BC_2
            MOV   LCD_ADDR,#054H
            LCALL SET_ADDR_LCD
            MOV   DPTR,#TITLE_19
            LCALL WRLINE_LCD
            LCALL SWAP_C2R
            LJMP  NEW_COD
CHECK_BC_2: MOV   WDOG,#11000011B
            CJNE  A,#003H,PRESSED_CODE
            MOV   BACK_CODE,#04H
            RET

PRESSED_CODE1:    MOV   WDOG,#11000011B
            CJNE  A,#00AH,P_C_NO1
P_C_NO1_B:  CLR   BEEP
            LCALL DELAY_100ms
            SETB  BEEP
            LJMP  PRESSED_CODE
P_C_NO1:    SUBB  A,#00AH
            JNC   P_C_NO1_B

PRESSED_CODE11:   MOV   BUFF1,KPAD_DATA
            MOV   LCD_DATA,#00101010B
            LCALL WRCHAR_LCD
            LCALL LCD_BLINK
            LCALL KEY_RELEASE
            LCALL LCD_BLINK

PRESSED_CODE2:    MOV   WDOG,#11000011B
            JNB   KEY_DA,PRESSED_CODE2
            LCALL GETKEY_DATA
            MOV   A,KPAD_DATA

            CJNE  A,#00AH,P_C_NO2
P_C_NO2_B:  CLR   BEEP
            LCALL DELAY_100ms
            SETB  BEEP
            LJMP  PRESSED_CODE2

P_C_NO2:    SUBB  A,#00AH
            JNC   P_C_NO2_B

PRESSED_CODE22:   MOV   BUFF2,KPAD_DATA
            MOV   LCD_DATA,#00101010B
            LCALL WRCHAR_LCD
            LCALL LCD_BLINK
            LCALL KEY_RELEASE
            LCALL LCD_BLINK

PRESSED_CODE3:    MOV   WDOG,#11000011B
            JNB   KEY_DA,PRESSED_CODE3
            LCALL GETKEY_DATA
            MOV   A,KPAD_DATA

            CJNE  A,#00AH,P_C_NO3
P_C_NO3_B:  CLR   BEEP
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            LCALL DELAY_100ms
            SETB  BEEP
            LJMP  PRESSED_CODE3

P_C_NO3:    SUBB  A,#00AH
            JNC   P_C_NO3_B

PRESSED_CODE33:   MOV   BUFF3,KPAD_DATA
            MOV   LCD_DATA,#00101010B
            LCALL WRCHAR_LCD
            LCALL LCD_BLINK
            LCALL KEY_RELEASE
            LCALL LCD_BLINK

PRESSED_CODE4:    MOV   WDOG,#11000011B
            JNB   KEY_DA,PRESSED_CODE4
            LCALL GETKEY_DATA
            MOV   A,KPAD_DATA

            CJNE  A,#00AH,P_C_NO4
P_C_NO4_B:  CLR   BEEP
            LCALL DELAY_100ms
            SETB  BEEP
            LJMP  PRESSED_CODE4

P_C_NO4:    SUBB  A,#00AH
            JNC   P_C_NO4_B

PRESSED_CODE44:   MOV   BUFF4,KPAD_DATA
            MOV   LCD_DATA,#00101010B
            LCALL WRCHAR_LCD
            LCALL LCD_BLINK
            LCALL KEY_RELEASE
            LCALL LCD_BLINK
            RET

;*************************************************
;CHECK CODE BETWEEN BUFFER(BUFF1-BUFF4) AND REAL CODE ( COD1-COD4 )
;*************************************************
CH_COD:           MOV   A,BUFF1
            CJNE  A,COD1,CODE_FALSE
CH_COD2:    MOV   A,BUFF2
            CJNE  A,COD2,CODE_FALSE
CH_COD3:    MOV   A,BUFF3
            CJNE  A,COD3,CODE_FALSE
CH_COD4:    MOV   A,BUFF4
            CJNE  A,COD4,CODE_FALSE
CODE_CORRECT:     MOV   ERR_TRUE,#1
            RET
CODE_FALSE: MOV   ERR_TRUE,#0
            RET

;*********************************************
;BUFFER TO ACC
;*********************************************
BUFFER2ACC: MOV   A,BUFFER
            ANL   A,#0FH
            SWAP  A
            MOV   B,A
            MOV   A,BUFFER+1
            ANL   A,#0FH
            ADD   A,B
            RET
;*********************************************
;SWAP BETWEET COD1-4 AND REST1-4
;*********************************************
SWAP_C2R:   MOV   A,COD1
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            XCH   A,REST1
            XCH   A,COD1
            MOV   A,COD2
            XCH   A,REST2
            XCH   A,COD2
            MOV   A,COD3
            XCH   A,REST3
            XCH   A,COD3
            MOV   A,COD4
            XCH   A,REST4
            XCH   A,COD4
            RET

;*********************************************
;SWAP BETWEET COD1-4 AND BUFF1-4
;*********************************************
SWAP_C2B:   MOV   A,COD1
            XCH   A,BUFF1
            XCH   A,COD1
            MOV   A,COD2
            XCH   A,BUFF2
            XCH   A,COD2
            MOV   A,COD3
            XCH   A,BUFF3
            XCH   A,COD3
            MOV   A,COD4
            XCH   A,BUFF4
            XCH   A,COD4
            RET

;*********************************************
;FUNCTION SHOW LCD WHEN PRESSED FALSE CODE
;*********************************************
CODE_ERR:   MOV   LCD_ADDR,#000H
            LCALL SET_ADDR_LCD
            MOV   DPTR,#TITLE_7
            LCALL WRLINE_LCD

            MOV   LCD_ADDR,#040H
            LCALL SET_ADDR_LCD
            LCALL WRLINE_LCD

            MOV   LCD_ADDR,#014H
            LCALL SET_ADDR_LCD
            LCALL WRLINE_LCD

            MOV   LCD_ADDR,#054H
            LCALL SET_ADDR_LCD
            LCALL WRLINE_LCD
            MOV   WDOG,#11000011B
            CLR   BEEP
            LCALL DELAY_1s
            SETB  BEEP
            RET

;*********************************************
;FUNCTION SHOW LCD WHEN PRESSED TRUE CODE
;*********************************************
CODE_TRUE:  MOV   LCD_ADDR,#000H
            LCALL SET_ADDR_LCD
            MOV   DPTR,#TITLE_6
            LCALL WRLINE_LCD
            MOV   LCD_ADDR,#040H
            LCALL SET_ADDR_LCD
            LCALL WRLINE_LCD

            MOV   LCD_ADDR,#014H
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งไม่อนุญาตให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            LCALL SET_ADDR_LCD
            LCALL WRLINE_LCD

            MOV   LCD_ADDR,#054H
            LCALL SET_ADDR_LCD
            LCALL WRLINE_LCD
            LCALL DELAY_1s
            RET

;*********************************************
;FUNCTION CLEAR ALL RAM TO 00H
;*********************************************
CLEAR_ALL:  MOV   DPTR,#0000H
            MOV   R7,#01BH
CLEAR_ALL1: MOV   R6,#0FFH                  ;  EACH LOOP = 1mS
CLEAR_ALL2: MOV   A,#000H
            MOVX  @DPTR,A
            INC   DPTR
            DJNZ  R6,CLEAR_ALL2
            DJNZ  R7,CLEAR_ALL1
            LCALL LCD_CLR
            MOV   LCD_ADDR,#040H
            LCALL SET_ADDR_LCD
            MOV   DPTR,#TITLE_21
            LCALL WRLINE_LCD
            MOV   WDOG,#11000011B
            LCALL DELAY_1s
            RET

;*********************************************
;FUNCTION CLEAR  RAM TO 00H WHICH SELECT LOAD AND DAY
;*********************************************
CLEAR_S_LD: MOV   WDOG,#11000011B
            LCALL GET_LOAD_DAY
            MOV   A,BACK_CODE
            CJNE  A,#04H,CLEAR_S_LD0
            RET
CLEAR_S_LD0:LCALL GET_ADDR_RAM
            MOV   R7,#078H
CLEAR_S_LD1:      MOV   A,#000H
            MOVX  @DPTR,A
            INC   DPTR
            DJNZ  R7,CLEAR_S_LD1
            LCALL LCD_CLR
            MOV   LCD_ADDR,#040H
            LCALL SET_ADDR_LCD
            MOV   DPTR,#TITLE_21
            LCALL WRLINE_LCD
            MOV   WDOG,#11000011B
            LCALL DELAY_1s
            RET

;*********************************************
;FUNCTION CLEAR  RAM TO 00H WHICH ONE BY ONE
;*********************************************
CLEAR_1BY1: MOV   WDOG,#11000011B
            LCALL GET_LOAD_DAY
            MOV   A,BACK_CODE
            CJNE  A,#04H,CLEAR_1BY1_0
            RET
CLEAR_1BY1_0:     LCALL TITLE_LOAD_DAY

CLEAR_1BY11:      MOV   RAM_PTR,#000H
            LCALL GET_ADDR_RAM
            LCALL SAVE_DP
CLEAR_1BY12:      MOV   LCD_ADDR,#040H
            LCALL SET_ADDR_LCD
            MOV   DPTR,#TITLE_19
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            LCALL WRLINE_LCD
            MOV   LCD_ADDR,#040H
            LCALL SET_ADDR_LCD
            LCALL SHOW_RAM

            MOV   LCD_ADDR,#014H
            LCALL SET_ADDR_LCD
            MOV   DPTR,#TITLE_19
            LCALL WRLINE_LCD
            MOV   LCD_ADDR,#014H
            LCALL SET_ADDR_LCD
            LCALL SHOW_RAM

            MOV   LCD_ADDR,#054H
            LCALL SET_ADDR_LCD
            MOV   DPTR,#TITLE_19
            LCALL WRLINE_LCD
            MOV   LCD_ADDR,#054H
            LCALL SET_ADDR_LCD
            LCALL SHOW_RAM
            LCALL KEY_RELEASE


CLEAR_1BY1_1_4:   MOV   R5,#90
CLEAR_1BY1_1_3:   MOV   R6,#100
CLEAR_1BY1_1_2:   MOV   R7,#0E6H
CLEAR_1BY1_1_0:   MOV   WDOG,#11000011B
                  DJNZ  R7,CLEAR_1BY1_1_1
                  DJNZ  R6,CLEAR_1BY1_1_2
                  DJNZ  R5,CLEAR_1BY1_1_3
                  LJMP  LOOP
CLEAR_1BY1_1_1:   JNB   KEY_DA,CLEAR_1BY1_1_0

            LCALL GETKEY_DATA
            MOV   A,KPAD_DATA

CLEAR_P_UP: CJNE  A,#00AH,CLEAR_P_DW
            MOV   A,RAM_PTR
            CJNE  A,#000H,CLEAR_P_UP1
            CLR   BEEP
            LCALL DELAY_100ms
            SETB  BEEP
            LJMP  CLEAR_1BY1_1_4
CLEAR_P_UP1:      DEC   RAM_PTR
            MOV   R5,#010H
CLEAR_P_UP2:      LCALL UPLOAD_DP
            MOV   A,DPL
            CJNE  A,#000H,CLEAR_P_UP3
            MOV   DPL,#0FFH
            DEC   DPH
            LCALL SAVE_DP
            DJNZ  R5,CLEAR_P_UP2
            LJMP  CLEAR_1BY12

CLEAR_P_UP3:      DEC   DPL
CLEAR_P_UP4:      LCALL SAVE_DP
            DJNZ  R5,CLEAR_P_UP2
            LJMP  CLEAR_1BY12

CLEAR_P_DW: CJNE  A,#00BH,CLEAR_1BY1_2
            MOV   A,RAM_PTR
            CJNE  A,#01BH,CLEAR_P_DW1
            LJMP  CLEAR_1BY11
CLEAR_P_DW1:      INC   RAM_PTR
            MOV   R5,#008H
CLEAR_P_DW2:      LCALL UPLOAD_DP
            MOV   A,DPL
            CJNE  A,#000H,CLEAR_P_DW3
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            MOV   DPL,#0FFH
            DEC   DPH
            LCALL SAVE_DP
            DJNZ  R5,CLEAR_P_DW2
            LJMP  CLEAR_1BY12

CLEAR_P_DW3:      DEC   DPL
CLEAR_P_DW4:      LCALL SAVE_DP
            DJNZ  R5,CLEAR_P_DW2
            LJMP  CLEAR_1BY12

CLEAR_1BY1_2:     SUBB  A,#004H
            JNC   CLEAR_1BY1_4
            MOV   A,KPAD_DATA
            CJNE  A,#004H,CLEAR_P_NUM0
CLEAR_1BY1_4:     LJMP  CLEAR_1BY1_3

CLEAR_P_NUM0:     MOV   BUFFER,KPAD_DATA
            LCALL KEY_RELEASE
CLEAR_P_NUM1:     MOV   WDOG,#11000011B
                  JNB   KEY_DA,CLEAR_P_NUM1
            LCALL GETKEY_DATA
            MOV   A,KPAD_DATA
CLEAR_P_NUM2:     CJNE  A,#00DH,CLEAR_P_NUM3
            LJMP  CLEAR_1BY1

CLEAR_P_NUM3:     CJNE  A,#00AH,CLEAR_P_NUM4
CLEAR_P_NUMB:     CLR   BEEP
            LCALL DELAY_100ms
            SETB  BEEP
            LJMP  CLEAR_P_NUM1

CLEAR_P_NUM4:     SUBB  A,#00AH
            JNC   CLEAR_P_NUMB
            MOV   BUFFER+1,KPAD_DATA
            LCALL KEY_RELEASE

            LCALL CONV2BCD
            LCALL GET_ADDR_RAM
            MOV   A,EDIT_PTR
            DEC   A
            MOV   B,#004H
            MUL   AB
            CJNE  A,#000H,CLEAR_ADDR_RAM
            MOV   BUFF_DPL,DPL
            MOV   BUFF_DPH,DPH
            LJMP  CLEAR_1BY1_5

CLEAR_ADDR_RAM:   MOV   R5,A
CLEAR_ADDR_RAM0:  INC   DPTR
                  DJNZ  R5,CLEAR_ADDR_RAM0
            MOV   BUFF_DPL,DPL
            MOV   BUFF_DPH,DPH

CLEAR_1BY1_5:     MOV   A,EDIT_PTR
            SUBB  A,RAM_PTR
            JNC   CLEAR_1BY1_6
            LJMP  CLEAR_1BY1_14

CLEAR_1BY1_6:     MOV   WDOG,#11000011B
            CJNE  A,#001H,CLEAR_1BY1_7
            LJMP  CLEAR_1BY1_10
CLEAR_1BY1_7:     CJNE  A,#002H,CLEAR_1BY1_8
            LJMP  CLEAR_1BY1_10
CLEAR_1BY1_8:     CJNE  A,#003H,CLEAR_1BY1_9
            LJMP  CLEAR_1BY1_10
CLEAR_1BY1_9:     LJMP  CLEAR_1BY1_14
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งยังมีสงวนสิทธิ์ในการดำเนินคดีตามกฎหมายกับผู้ที่นำเอกสารนี้ไปใช้ในทางที่ผิดกฎหมายทุกกรณี

```
CLEAR_1BY1_10:    MOV   DPL,BUFF_DPL
                  MOV   DPH,BUFF_DPH

            INC   DPTR
            INC   DPTR
            INC   DPTR
            INC   DPTR
            MOV   SLIDE_DPL,DPL
            MOV   SLIDE_DPH,DPH

CLEAR_SL_1: MOV   A,EDIT_PTR
            CJNE  A,#01EH,CLEAR_SL_2
CLEAR_SL_0: MOV   DPL,BUFF_DPL
            MOV   DPH,BUFF_DPH
            MOV   A,#000H
            MOVX  @DPTR,A
            INC   DPTR
            MOVX  @DPTR,A
            INC   DPTR
            MOVX  @DPTR,A
            INC   DPTR
            MOVX  @DPTR,A
            LJMP  CLEAR_1BY1_110

CLEAR_SL_2: MOV   DPL,SLIDE_DPL
            MOV   DPH,SLIDE_DPH
            MOVX  A,@DPTR
            CJNE  A,#000H,CLEAR_SL_3
            LJMP  CLEAR_SL_0

CLEAR_SL_3: MOV   WDOG,#11000011B
            MOV   DPL,BUFF_DPL
            MOV   DPH,BUFF_DPH
            INC   DPTR
            MOV   BUFF_DPL,DPL                  ;HH OF BUFF
            MOV   BUFF_DPH,DPH

            MOV   DPL,SLIDE_DPL
            MOV   DPH,SLIDE_DPH
            INC   DPTR                          ;HH OF SLIDE
            MOVX  A,@DPTR
            INC   DPTR                          ;MM OF      SLIDE
            MOV   SLIDE_DPL,DPL
            MOV   SLIDE_DPH,DPH

            MOV   DPL,BUFF_DPL
            MOV   DPH,BUFF_DPH
            MOVX  @DPTR,A
            INC   DPTR                          ;MM OF BUFF
            MOV   BUFF_DPL,DPL
            MOV   BUFF_DPH,DPH

            MOV   DPL,SLIDE_DPL
            MOV   DPH,SLIDE_DPH
            MOVX  A,@DPTR
            INC   DPTR                          ;ON/OFF OF SLIDE
            MOV   SLIDE_DPL,DPL
            MOV   SLIDE_DPH,DPH

            MOV   DPL,BUFF_DPL
            MOV   DPH,BUFF_DPH
            MOVX  @DPTR,A
            INC   DPTR                          ;ON/OFF OF BUFF
            MOV   BUFF_DPL,DPL
            MOV   BUFF_DPH,DPH

            MOV   DPL,SLIDE_DPL
            MOV   DPH,SLIDE_DPH
```

เอกสารนี้เป็นเอกสารสงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            MOVX  A,@DPTR
            INC   DPTR                              ;NEW PROGRAM OF SLIDE
            MOV   SLIDE_DPL,DPL
            MOV   SLIDE_DPH,DPH

            MOV   DPL,BUFF_DPL
            MOV   DPH,BUFF_DPH
            MOVX  @DPTR,A
            INC   DPTR                              ;NEW PROGRAM OF BUFF
            MOV   BUFF_DPL,DPL
            MOV   BUFF_DPH,DPH

            INC   EDIT_PTR
            LJMP  CLEAR_SL_1


CLEAR_1BY1_110:   MOV   WDOG,#11000011B
                  MOV   R5,#00CH
CLEAR_1BY1_11:    LCALL UPLOAD_DP
            MOV   A,DPL
            CJNE  A,#000H,CLEAR_1BY1_12
            MOV   DPL,#0FFH
            DEC   DPH
            LCALL SAVE_DP
            DJNZ  R5,CLEAR_1BY1_11
            LJMP  CLEAR_1BY12

CLEAR_1BY1_12:    DEC   DPL
CLEAR_1BY1_13:    LCALL SAVE_DP
            DJNZ  R5,CLEAR_1BY1_11
            LJMP  CLEAR_1BY12

CLEAR_1BY1_3:     MOV   A,KPAD_DATA
            CJNE  A,#00DH,CLEAR_1BY1_14
            LJMP  CLEAR_1BY1
CLEAR_1BY1_14:    CLR   BEEP
            LCALL DELAY_100ms
            SETB  BEEP
            LJMP  CLEAR_1BY1_1_4

;*******************************************
;FUNCTION CONV BUFFER 2 BCD 2 CHECK WITH RAM_PTR
;*******************************************
CONV2BCD:   LCALL BUFFER2ACC
            MOV   B,A
            CJNE  A,#030H,CONV2BCD_1
            MOV   EDIT_PTR,#01EH
            RET

CONV2BCD_1: CJNE  A,#020H,CONV2BCD_2
            MOV   EDIT_PTR,#014H
            RET
CONV2BCD_2: SUBB  A,#020H
            JC    CONV2BCD_3
            MOV   EDIT_PTR,#014H
            MOV   A,B
            ANL   A,#00FH
            ADD   A,EDIT_PTR
            MOV   EDIT_PTR,A
            RET

CONV2BCD_3: MOV   A,B
            CJNE  A,#010H,CONV2BCD_4
            MOV   EDIT_PTR,#00AH
            RET
CONV2BCD_4: SUBB  A,#010H
            JC    CONV2BCD_5
            MOV   EDIT_PTR,#00AH
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            MOV   A,B
            ANL   A,#00FH
            ADD   A,EDIT_PTR
            MOV   EDIT_PTR,A
            RET

CONV2BCD_5: MOV   A,B
            MOV   EDIT_PTR,A
            RET

;*********************************************
;FUNCTION RECEIVE  2  KEY(NUMBER) AND STORAGE IN BUFFER
;*********************************************
PRESS_BCD:  MOV   WDOG,#11000011B
            JNB   KEY_DA,PRESS_BCD
            LCALL GETKEY_DATA
            MOV   A,KPAD_DATA

            CJNE  A,#00AH,PRESS_BCD1
BCD_B:            CLR   BEEP
            LCALL DELAY_100ms
            SETB  BEEP
            LJMP  PRESS_BCD

PRESS_BCD1: SUBB  A,#00AH
            JNC   BCD_B

PRESS_BCD11:MOV   BUFFER,KPAD_DATA
            LCALL KPAD2LCD
            LCALL LCD_BLINK
            LCALL KEY_RELEASE

PRESS_BCD_1:      MOV   WDOG,#11000011B
            JNB   KEY_DA,PRESS_BCD_1
            LCALL GETKEY_DATA
            MOV   A,KPAD_DATA

            CJNE  A,#00AH,PRESS_BCD2
BCD_B1:     CLR   BEEP
            LCALL DELAY_100ms
            SETB  BEEP
            LJMP  PRESS_BCD_1

PRESS_BCD2: SUBB  A,#00AH
            JNC   BCD_B1

PRESS_BCD22:      MOV   BUFFER+1,KPAD_DATA
            LCALL KPAD2LCD
            LCALL LCD_BLINK
            LCALL KEY_RELEASE
            RET

;*********************************************
;FUNCTION FIND DATA OF EACH LOAD (BUFF_L_) THAT 0 OR 1
;*********************************************
GETOUT_DATA:      MOV   BACK_CODE,#000H
                  MOV   BUFF_DAY,DAY
                  MOV   BUFF_LOAD,#001H

GETOUT_DATA1:     MOV   A,BUFF_LOAD
            CJNE  A,#009H,GETOUT_DATA2
            RET
GETOUT_DATA2:     LCALL GET_ADDR_RAM
            LCALL SAVE_DP
            MOV   R5,#01EH
GETOUT_DATA3:     LCALL UPLOAD_DP
            MOVX  A,@DPTR                     ;  PROGRAM NUMBER
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            CJNE  A,#000H,GETOUT_DATA4
            INC   BUFF_LOAD
            LJMP  GETOUT_DATA1

GETOUT_DATA4:     INC   DPTR                        ;  HH
            MOVX  A,@DPTR
            CJNE  A,HOURS,GETOUT_DATA5
            LJMP  GETOUT_DATA6
GETOUT_DATA5:     INC   DPTR                        ;  MM
GETOUT_DATA7:     INC   DPTR                        ;  ON/OFF
                  INC   DPTR                        ;  NEW PROGRAM
                  LCALL SAVE_DP
                  DJNZ  R5,GETOUT_DATA3
                  INC   BUFF_LOAD
                  LJMP  GETOUT_DATA1
GETOUT_DATA6:     INC   DPTR                        ;  MM
            MOVX  A,@DPTR
            CJNE  A,MINUTES,GETOUT_DATA7
            MOV   BACK_CODE,#005H
            INC   DPTR                            ;  ON OFF
            MOVX  A,@DPTR
            MOV   B,A
            INC   DPTR                            ;  NEW PROGRAM
            LCALL SAVE_DP
            MOV   A,B
            CJNE  A,#000H,GETOUT_DATA8
            MOV   A,BUFF_LOAD
GETOUT_DATA_1:    CJNE  A,#001H,GETOUT_DATA_2
            MOV   BUFF_L1,#000H
            DJNZ  R5,GETOUT_DATA3
            INC   BUFF_LOAD
            LJMP  GETOUT_DATA1

GETOUT_DATA_2:    CJNE  A,#002H,GETOUT_DATA_3
            MOV   BUFF_L2,#000H
            DJNZ  R5,GETOUT_DATA3
            INC   BUFF_LOAD
            LJMP  GETOUT_DATA1

GETOUT_DATA_3:    CJNE  A,#003H,GETOUT_DATA_4
            MOV   BUFF_L3,#000H
            DJNZ  R5,GETOUT_DATA3
            INC   BUFF_LOAD
            LJMP  GETOUT_DATA1

GETOUT_DATA_4:    CJNE  A,#004H,GETOUT_DATA_5
            MOV   BUFF_L4,#000H
            DJNZ  R5,GETOUT_DATA3
            INC   BUFF_LOAD
            LJMP  GETOUT_DATA1

GETOUT_DATA_5:    CJNE  A,#005H,GETOUT_DATA_6
            MOV   BUFF_L5,#000H
            DJNZ  R5,GETOUT_DATA3
            INC   BUFF_LOAD
            LJMP  GETOUT_DATA1

GETOUT_DATA_6:    CJNE  A,#006H,GETOUT_DATA_7
            MOV   BUFF_L6,#000H
            DJNZ  R5,GETOUT_DATA_66
            INC   BUFF_LOAD
            LJMP  GETOUT_DATA1
GETOUT_DATA_66:   LJMP  GETOUT_DATA3

GETOUT_DATA_7:    CJNE  A,#007H,GETOUT_DATA_8
            MOV   BUFF_L7,#000H
            DJNZ  R5,GETOUT_DATA_66
            INC   BUFF_LOAD
```

เอกสารนี้เป็นเอกสารสงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
ATA1

FF_L8,#000H
T_DATA_66
D
ATA1


BUFF_LOAD
#001H,GETOUT_DATA_21
#001H
T_DATA_66
D
ATA1

#002H,GETOUT_DATA_31
#001H
T_DATA_77
D
ATA1

#003H,GETOUT_DATA_41
#001H
T_DATA_77
D
ATA1

#004H,GETOUT_DATA_51
#001H
T_DATA_77
D
ATA1
TOUT_DATA3

#005H,GETOUT_DATA_61
#001H
T_DATA_77
D
ATA1

#006H,GETOUT_DATA_71
#001H
T_DATA_77
D
ATA1

#007H,GETOUT_DATA_81
#001H
T_DATA_77
D
ATA1

FF_L8,#001H
T_DATA_77
D
ATA1

**********
CF8574_2 ( RELAY OUTPUT )
**********
BUFF_L1
ETBUFF_OUT1
UT
110B
,A
OUT2
BUFF_OUT
001B
```

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
โอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            LJMP

GETOUT_DATA_8:
            DJNZ
            INC
            LJMP


GETOUT_DATA8:
GETOUT_DATA_11:
            MOV
            DJNZ
            INC
            LJMP

GETOUT_DATA_21:
            MOV
            DJNZ
            INC
            LJMP

GETOUT_DATA_31:
            MOV
            DJNZ
            INC
            LJMP

GETOUT_DATA_41:
            MOV
            DJNZ
            INC
            LJMP
GETOUT_DATA_77:

GETOUT_DATA_51:
            MOV
            DJNZ
            INC
            LJMP

GETOUT_DATA_61:
            MOV
            DJNZ
            INC
            LJMP

GETOUT_DATA_71:
            MOV
            DJNZ
            INC
            LJMP

GETOUT_DATA_81:
            DJNZ
            INC
            LJMP

;*****************
;FUNCTION FIND OUT
;*****************
GETBUFF_OUT:
            CJNE
            MOV
            ANL
            MOV
            LJMP
GETBUFF_OUT1:
            ORL
```

เอกสารนี้เป็นเอกสารสงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            MOV   BUFF_OUT,A

GETBUFF_OUT2:     MOV   A,BUFF_L2
            CJNE  A,#00H,GETBUFF_OUT3
            MOV   A,BUFF_OUT
            ANL   A,#11111101B
            MOV   BUFF_OUT,A
            LJMP  GETBUFF_OUT4
GETBUFF_OUT3:     MOV   A,BUFF_OUT
            ORL   A,#00000010B
            MOV   BUFF_OUT,A

GETBUFF_OUT4:     MOV   A,BUFF_L3
            CJNE  A,#00H,GETBUFF_OUT5
            MOV   A,BUFF_OUT
            ANL   A,#11111011B
            MOV   BUFF_OUT,A
            LJMP  GETBUFF_OUT6
GETBUFF_OUT5:     MOV   A,BUFF_OUT
            ORL   A,#00000100B
            MOV   BUFF_OUT,A

GETBUFF_OUT6:     MOV   A,BUFF_L4
            CJNE  A,#00H,GETBUFF_OUT7
            MOV   A,BUFF_OUT
            ANL   A,#11110111B
            MOV   BUFF_OUT,A
            LJMP  GETBUFF_OUT8
GETBUFF_OUT7:     MOV   A,BUFF_OUT
            ORL   A,#00001000B
            MOV   BUFF_OUT,A

GETBUFF_OUT8:     MOV   A,BUFF_L5
            CJNE  A,#00H,GETBUFF_OUT9
            MOV   A,BUFF_OUT
            ANL   A,#11101111B
            MOV   BUFF_OUT,A
            LJMP  GETBUFF_OUT10
GETBUFF_OUT9:     MOV   A,BUFF_OUT
            ORL   A,#00010000B
            MOV   BUFF_OUT,A

GETBUFF_OUT10:    MOV   A,BUFF_L6
            CJNE  A,#00H,GETBUFF_OUT11
            MOV   A,BUFF_OUT
            ANL   A,#11011111B
            MOV   BUFF_OUT,A
            LJMP  GETBUFF_OUT12
GETBUFF_OUT11:    MOV   A,BUFF_OUT
            ORL   A,#00100000B
            MOV   BUFF_OUT,A

GETBUFF_OUT12:    MOV   A,BUFF_L7
            CJNE  A,#00H,GETBUFF_OUT13
            MOV   A,BUFF_OUT
            ANL   A,#10111111B
            MOV   BUFF_OUT,A
            LJMP  GETBUFF_OUT14
GETBUFF_OUT13:    MOV   A,BUFF_OUT
            ORL   A,#01000000B
            MOV   BUFF_OUT,A

GETBUFF_OUT14:    MOV   A,BUFF_L8
            CJNE  A,#00H,GETBUFF_OUT15
            MOV   A,BUFF_OUT
            ANL   A,#01111111B
            MOV   BUFF_OUT,A
            LJMP  GETBUFF_OUT16
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
GETBUFF_OUT15:    MOV   A,BUFF_OUT
            ORL   A,#10000000B
            MOV   BUFF_OUT,A
GETBUFF_OUT16:    RET

;*******************************************
;FUNCTION LCD SHOW STATUS OF EACH LOAD
;*******************************************
SHOW_STAT:  LCALL LCD_CLR
            MOV   LCD_ADDR,#000H
            LCALL SET_ADDR_LCD
            MOV   DPTR,#TITLE_24
            LCALL WRLINE_LCD

            MOV   LCD_ADDR,#040H
            LCALL SET_ADDR_LCD
            LCALL WRLINE_LCD

            MOV   LCD_ADDR,#014H
            LCALL SET_ADDR_LCD
            LCALL WRLINE_LCD

            MOV   LCD_ADDR,#054H
            LCALL SET_ADDR_LCD
            LCALL WRLINE_LCD

SHOW_STAT1: MOV   A,BUFF_L1
            CJNE  A,#000H,SHOW_STAT1_1
            MOV   LCD_DATA,#'0'
            LJMP  SHOW_STAT1_2
SHOW_STAT1_1:     MOV   LCD_DATA,#'1'
SHOW_STAT1_2:     MOV   LCD_ADDR,#019H
            LCALL SET_ADDR_LCD
            LCALL WRCHAR_LCD

SHOW_STAT2: MOV   A,BUFF_L2
            CJNE  A,#000H,SHOW_STAT2_1
            MOV   LCD_DATA,#'0'
            LJMP  SHOW_STAT2_2
SHOW_STAT2_1:     MOV   LCD_DATA,#'1'
SHOW_STAT2_2:     MOV   LCD_ADDR,#01BH
            LCALL SET_ADDR_LCD
            LCALL WRCHAR_LCD

SHOW_STAT3: MOV   A,BUFF_L3
            CJNE  A,#000H,SHOW_STAT3_1
            MOV   LCD_DATA,#'0'
            LJMP  SHOW_STAT3_2
SHOW_STAT3_1:     MOV   LCD_DATA,#'1'
SHOW_STAT3_2:     MOV   LCD_ADDR,#01DH
            LCALL SET_ADDR_LCD
            LCALL WRCHAR_LCD

SHOW_STAT4: MOV   A,BUFF_L4
            CJNE  A,#000H,SHOW_STAT4_1
            MOV   LCD_DATA,#'0'
            LJMP  SHOW_STAT4_2
SHOW_STAT4_1:     MOV   LCD_DATA,#'1'
SHOW_STAT4_2:     MOV   LCD_ADDR,#01FH
            LCALL SET_ADDR_LCD
            LCALL WRCHAR_LCD

SHOW_STAT5: MOV   A,BUFF_L5
            CJNE  A,#000H,SHOW_STAT5_1
            MOV   LCD_DATA,#'0'
            LJMP  SHOW_STAT5_2
SHOW_STAT5_1:     MOV   LCD_DATA,#'1'
SHOW_STAT5_2:     MOV   LCD_ADDR,#021H
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            LCALL SET_ADDR_LCD
            LCALL WRCHAR_LCD

SHOW_STAT6: MOV   A,BUFF_L6
            CJNE  A,#000H,SHOW_STAT6_1
            MOV   LCD_DATA,#'0'
            LJMP  SHOW_STAT6_2
SHOW_STAT6_1:     MOV   LCD_DATA,#'1'
SHOW_STAT6_2:     MOV   LCD_ADDR,#023H
            LCALL SET_ADDR_LCD
            LCALL WRCHAR_LCD

SHOW_STAT7: MOV   A,BUFF_L7
            CJNE  A,#000H,SHOW_STAT7_1
            MOV   LCD_DATA,#'0'
            LJMP  SHOW_STAT7_2
SHOW_STAT7_1:     MOV   LCD_DATA,#'1'
SHOW_STAT7_2:     MOV   LCD_ADDR,#025H
            LCALL SET_ADDR_LCD
            LCALL WRCHAR_LCD

SHOW_STAT8: MOV   A,BUFF_L8
            CJNE  A,#000H,SHOW_STAT8_1
            MOV   LCD_DATA,#'0'
            LJMP  SHOW_STAT8_2
SHOW_STAT8_1:     MOV   LCD_DATA,#'1'
SHOW_STAT8_2:     MOV   LCD_ADDR,#027H
            LCALL SET_ADDR_LCD
            LCALL WRCHAR_LCD

            MOV   WDOG,#11100011B

            LCALL KEY_RELEASE

            RET

;************************************************
;DELAY TIME LCD_DELAY 10ms , 100ms
;************************************************

I2C_DELAY:  MOV   R6,#00CH
I2C_DELAY_1:      NOP
            NOP
            DJNZ  R6,I2C_DELAY_1
            RET

LCD_DELAY:  MOV   R7,#002
LCD_DELAY_1:      MOV   R6,#0E6H                ;  EACH LOOP = 1mS
LCD_DELAY_2:      NOP
            NOP
            DJNZ  R6,LCD_DELAY_2
            DJNZ  R7,LCD_DELAY_1
            RET

DELAY_10ms: MOV   R7,#010
DELAY_10ms_1:     MOV   R6,#0E6H          ;  EACH LOOP = 1mS
DELAY_10ms_2:     NOP
            NOP
            DJNZ  R6,DELAY_10ms_2
            DJNZ  R7,DELAY_10ms_1
            RET

DELAY_100ms:      MOV   R7,#100
DELAY_100ms_1:    MOV   R6,#0E6H          ;  EACH LOOP = 1mS
DELAY_100ms_2:    NOP
            NOP
            DJNZ  R6,DELAY_100ms_2
            DJNZ  R7,DELAY_100ms_1
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            RET

DELAY_1s:   MOV   R5,#100
DELAY_1s_1: LCALL DELAY_10ms
            DJNZ  R5,DELAY_1s_1
            RET

;*************************************************************
;DEFINE CONSTANT
;*************************************************************

TITLE_1:    DB    ' SET TO MANUAL MODE '

TITLE_3:    DB    '        MENU        '
            DB    '1.Setting(Time/Date)'
            DB    '2.Change Password   '
            DB    '3.About Program     '

TITLE_4:    DB    ' PRESS CODE : XXXX  '

TITLE_5:    DB    '####################'
            DB    '# READY 2 SET CODE #'
            DB    '# READY 2 SET CODE #'
            DB    '####################'

TITLE_6:    DB    '*********************'
            DB    '* YOUR CODE IS TRUE*'
            DB    '* YOUR CODE IS TRUE*'
            DB    '*********************'

TITLE_7:    DB    '*********************'
            DB    '*YOUR CODE IS FALSE*'
            DB    '*YOUR CODE IS FALSE*'
            DB    '*********************'

TITLE_8:    DB    '    CHANGE CODE     '
TITLE_9:    DB    'OLD CODE      : XXXX'
TITLE_10:   DB    'NEW CODE      : XXXX'
TITLE_11:   DB    'CONFIRM CODE  : XXXX'

TITLE_12:   DB    ' YOU HAVE NEW CODE  '

TITLE_13:   DB    '  SETTING  Time/Date'
            DB    ' 1. SET TIME        '
            DB    ' 2. SET DATE        '

TITLE_14:   DB    '    ABOUT PROGRAM   '
            DB    ' 1. Check Program   '
            DB    ' 2. Edit Program    '
            DB    ' 3. Clear Program   '

TITLE_15:   DB    'CHOOSE LOAD No / DAY'
TITLE_16:   DB    'LOAD No(Key1-8):    '
TITLE_17:   DB    'DAY (Key 1-7)  : Day'

TITLE_18:   DB    'PROGRAM LOAD No     '
TITLE_19:   DB    '                    '
TITLE_20:   DB    '    CLEAR PROGRAM   '
            DB    ' 1. All             '
            DB    ' 2. Select Load+Day '
            DB    ' 3. One By One      '

TITLE_21:   DB    ' CLEAR RAM COMPLETED'
TITLE_22:   DB    '    hh:mm 0/1       '

TITLE_23:   DB    '   SET TO PROGRAM   '
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
TITLE_24:   DB     '  SHOW LOAD STATUS  '
            DB     'LOAD 1 2 3 4 5 6 7 8'
            DB     'STAT                '
            DB     '    1=ON   0=OFF    '

SCR_SET_DATE:      DB     '    Day  dd/mm/yy   '
SCR_SET_TIME:      DB     '   Time : hh/mm/ss  '


MONTH_JAN:  DB     'JAN'
MONTH_FEB:  DB     'FEB'
MONTH_MAR:  DB     'MAR'
MONTH_APR:  DB     'APR'
MONTH_MAY:  DB     'MAY'
MONTH_JUN:  DB     'JUN'
MONTH_JUL:  DB     'JUL'
MONTH_AUG:  DB     'AUG'
MONTH_SEP:  DB     'SEP'
MONTH_OCT:  DB     'OCT'
MONTH_NOV:  DB     'NOV'
MONTH_DEC:  DB     'DEC'

DAY_SUN:    DB     'SUN'
DAY_MON:    DB     'MON'
DAY_TUE:    DB     'TUE'
DAY_WED:    DB     'WED'
DAY_THU:    DB     'THU'
DAY_FRI:    DB     'FRI'
DAY_SAT:    DB     'SAT'
DAY_DAY:    DB     'Day'
LOAD_ON:    DB     'ON '
LOAD_OFF:   DB     'OFF'
RAM_END:    DB     'END'

CONV_KEY:   DB     '123:456;789<=0>?'

            END
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

DALLAS SEMICONDUCTOR

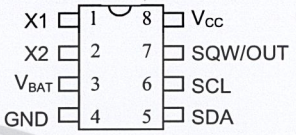
DS1307
64 x 8 Serial Real-Time Clock

www.maxim-ic.com

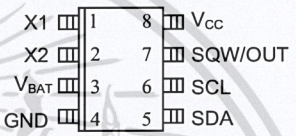
FEATURES

- Real-time clock (RTC) 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, battery-backed, nonvolatile (NV) RAM for data storage
- Two-wire serial interface
- Programmable squarewave output signal
- Automatic power-fail detect and switch circuitry
- Consumes less than 500nA in battery backup mode with oscillator running
- Optional industrial temperature range: -40°C to +85°C
- Available in 8-pin DIP or SOIC
- Underwriters Laboratory (UL) recognized

PIN ASSIGNMENT



DS1307 8-Pin DIP (300-mil)



DS1307 8-Pin SOIC (150-mil)

PIN DESCRIPTION

V_{CC}	- Primary Power Supply
X1, X2	- 32.768kHz Crystal Connection
V_{BAT}	- +3V Battery Input
GND	- Ground
SDA	- Serial Data
SCL	- Serial Clock
SQW/OUT	- Square Wave/Output Driver

ORDERING INFORMATION

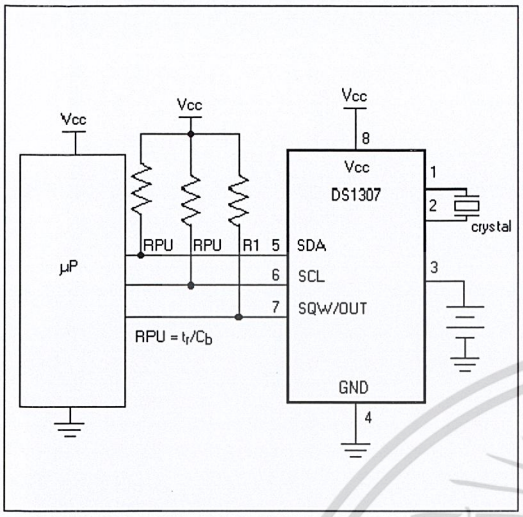
DS1307	8-Pin DIP (300-mil)
DS1307Z	8-Pin SOIC (150-mil)
DS1307N	8-Pin DIP (Industrial)
DS1307ZN	8-Pin SOIC (Industrial)

DESCRIPTION

The DS1307 Serial Real-Time Clock is a low-power, full binary-coded decimal (BCD) clock/calendar plus 56 bytes of NV 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 fewer 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 that detects power failures and automatically switches to the battery supply.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

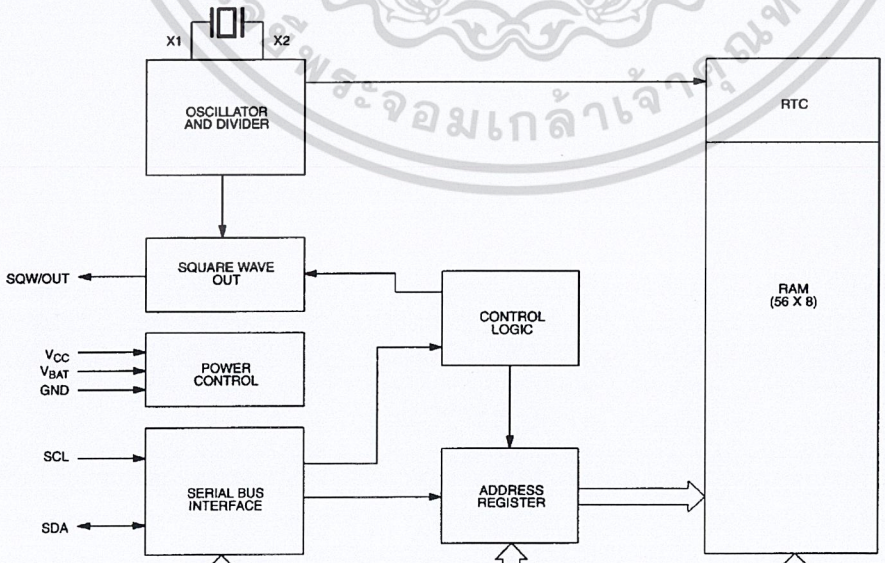
TYPICAL OPERATING CIRCUIT



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 RTC.

DS1307 BLOCK DIAGRAM Figure 1



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

SIGNAL DESCRIPTIONS

V_{CC}, GND – DC power is provided to the device on these pins. V_{CC} is the +5V input. When 5V is applied within normal limits, the device is fully accessible and data can be written and read. When a 3V 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 power supply (nominal 3.0V DC) at V_{BAT}.

V_{BAT} – Battery input for any standard 3V lithium cell or other energy source. Battery voltage must be held between 2.0V and 3.5V for proper operation. The nominal write protect trip point voltage at which access to the RTC and user RAM is denied is set by the internal circuitry as 1.25 x V_{BAT} nominal. A lithium battery with 48mAhr or greater will back up the DS1307 for more than 10 years in the absence of power at 25°C. UL recognized to ensure against reverse charging current when used in conjunction with a lithium battery.

See "Conditions of Acceptability" at http://www.maxim-ic.com/TechSupport/QA/ntrl.htm.

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

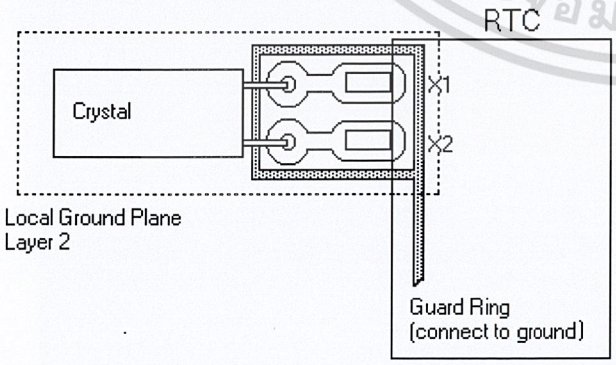
SDA (Serial Data Input/Output) – SDA is the input/output pin for the 2-wire serial interface. The SDA pin is open drain which requires an external pullup resistor.

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 (1Hz, 4kHz, 8kHz, 32kHz). The SQW/OUT pin is open drain and requires an external pull-up resistor. SQW/OUT will operate with either Vcc or Vbat applied.

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

For more information on crystal selection and crystal layout considerations, please consult Application Note 58, "Crystal Considerations with Dallas Real-Time Clocks." The DS1307 can also be driven by an external 32.768kHz oscillator. In this configuration, the X1 pin is connected to the external oscillator signal and the X2 pin is floated.

RECOMMENDED LAYOUT FOR CRYSTAL



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

CLOCK ACCURACY

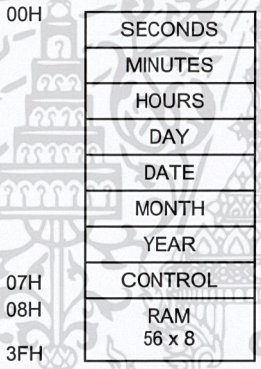
The accuracy of the clock is dependent upon the accuracy of the crystal and the accuracy of the match between the capacitive load of the oscillator circuit and the capacitive load for which the crystal was trimmed. Additional error will be added by crystal frequency drift caused by temperature shifts. External circuit noise coupled into the oscillator circuit may result in the clock running fast. See Application Note 58, "Crystal Considerations with Dallas Real-Time Clocks" for detailed information.

Please review Application Note 95, "Interfacing the DS1307 with a 8051-Compatible Microcontroller" for additional information.

RTC AND RAM ADDRESS MAP

The address map for the RTC and RAM registers of the DS1307 is shown in Figure 2. The RTC registers are located in address locations 00h to 07h. The RAM registers are located in address locations 08h to 3Fh. During a multi-byte 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 RTC 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 BCD format. Bit 7 of register 0 is the clock halt (CH) bit. When this bit is set to a 1, the oscillator is disabled. When cleared to a 0, the oscillator is enabled.

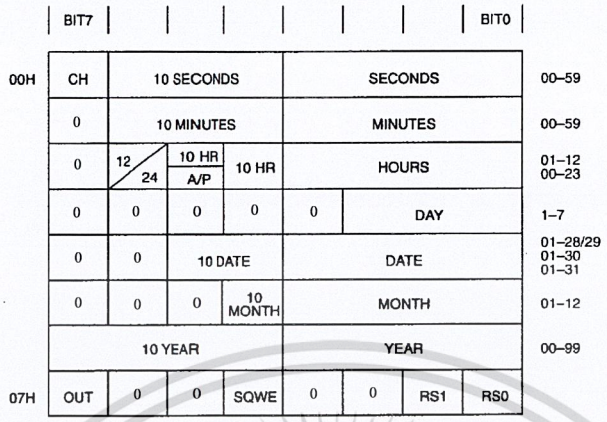
Please note that the initial power-on state of all registers is not defined. Therefore, it is important to enable the oscillator (CH bit = 0) during initial configuration.

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).

On a 2-wire START, the current time is transferred to a second set of registers. The time information is read from these secondary registers, while the clock may continue to run. This eliminates the need to re-read the registers in case of an update of the main registers during a read.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

DS1307 TIMEKEEPER REGISTERS Figure 3



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	0	0	SQWE	0	0	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 upon the value of the RS0 and RS1 bits. With the square wave output set to 1Hz, the clock registers update on the falling edge of the square wave.

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

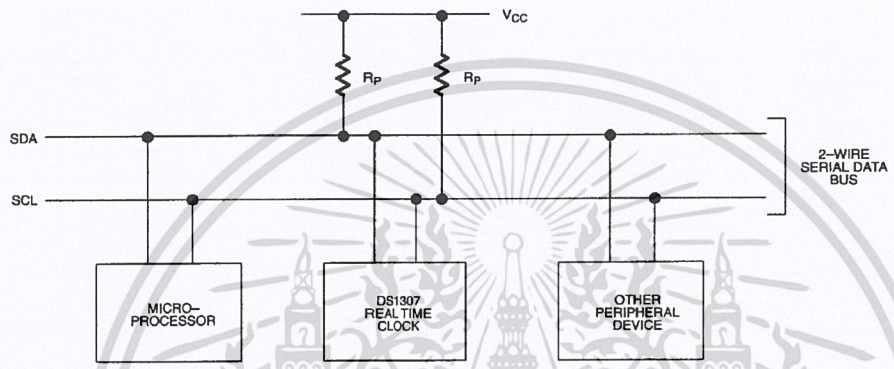
RS1	RS0	SQW OUTPUT FREQUENCY
0	0	1Hz
0	1	4.096kHz
1	0	8.192kHz
1	1	32.768kHz

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

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 referred to as slaves. The bus must be controlled by a master device that 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.

TYPICAL 2-WIRE BUS CONFIGURATION Figure 4



Figures 5, 6, and 7 detail how data is transferred on the 2-wire bus.

- 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 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.

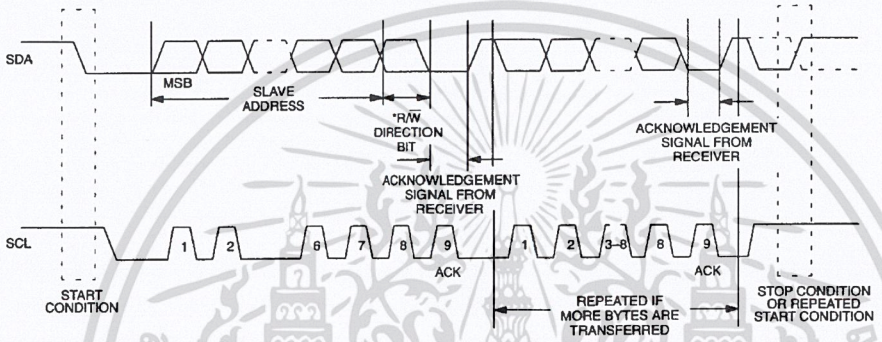
Each data transfer is initiated with a START condition and terminated with a STOP condition. The number of data bytes transferred between START and STOP conditions is not limited, and is determined by the master device. The information is transferred byte-wise and each receiver acknowledges with a ninth bit. Within the 2-wire bus specifications a regular mode (100kHz clock rate) and a fast mode (400kHz clock rate) are defined. The DS1307 operates in the regular mode (100kHz) only.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลง และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

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

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

DATA TRANSFER ON 2-WIRE SERIAL BUS Figure 5



Depending upon the state of the R/$\overline{W}$ bit, two types of data transfer are possible:

1. **Data transfer from a master transmitter to a slave receiver.** The first byte transmitted by the master is the slave address. Next follows a number of data bytes. The slave returns an acknowledge bit after each received byte. Data is transferred with the most significant bit (MSB) first.

2. **Data transfer from a slave transmitter to a master receiver.** The first byte (the slave address) is transmitted by the master. The slave then returns an acknowledge bit. This is followed by the slave transmitting a number of data bytes. The master returns an acknowledge bit after all received bytes other than the last byte. At the end of the last received byte, a "not acknowledge" is returned.

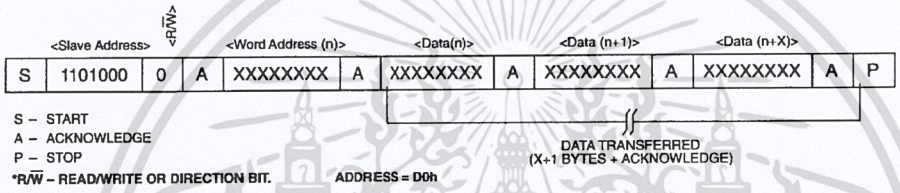
The master device generates all of the serial clock pulses and the START and STOP conditions. A transfer is ended with a STOP condition or with a repeated START condition. Since a repeated START condition is also the beginning of the next serial transfer, the bus will not be released. Data is transferred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

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

The DS1307 may operate in the following two modes:

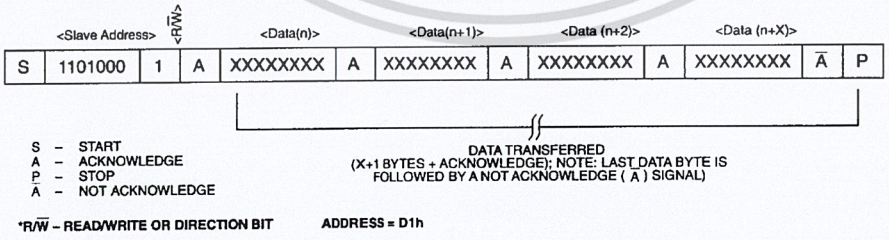
1. **Slave receiver mode (DS1307 write mode):** Serial data and clock are received through SDA and SCL. After each byte is received an acknowledge bit is transmitted. START and STOP conditions are recognized as the beginning and end of a serial transfer. Address recognition is performed by hardware after 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 ($R/\overline{W}$) which, for a write, is a 0. After receiving and decoding the address byte the device 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



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 ($R/\overline{W}$) which, for a read, is a 1. After receiving and decoding the address byte the device 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



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ABSOLUTE MAXIMUM RATINGS*

Voltage on Any Pin Relative to Ground	-0.5V to +7.0V
Storage Temperature	-55°C to +125°C
Soldering Temperature	260°C for 10 seconds DIP See JPC/JEDEC Standard J-STD-020A for Surface Mount Devices

* 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.

Range	Temperature	V_{CC}
Commercial	0°C to +70°C	4.5V to 5.5V V_{CC1}
Industrial	-40°C to +85°C	4.5V to 5.5V V_{CC1}

RECOMMENDED DC OPERATING CONDITIONS

(Over the operating range*)

PARAMETER	SYMBOL	MIN	TYP	MAX	UNITS	NOTES
Supply Voltage	V_{CC}	4.5	5.0	5.5	V	
Logic 1	V_{IH}	2.2		V_{CC} + 0.3	V	
Logic 0	V_{IL}	-0.5		+0.8	V	
V_{BAT} Battery Voltage	V_{BAT}	2.0		3.5	V	

*Unless otherwise specified.

DC ELECTRICAL CHARACTERISTICS

(Over the operating range*)

PARAMETER	SYMBOL	MIN	TYP	MAX	UNITS	NOTES
Input Leakage (SCL)	I_{LI}			1	µA	
I/O Leakage (SDA & SQW/OUT)	I_{LO}			1	µA	
Logic 0 Output (I_{OL} = 5mA)	V_{OL}			0.4	V	
Active Supply Current	I_{CCA}			1.5	mA	7
Standby Current	I_{CCS}			200	µA	1
Battery Current (OSC ON); SQW/OUT OFF	I_{BAT1}		300	500	nA	2
Battery Current (OSC ON); SQW/OUT ON (32kHz)	I_{BAT2}		480	800	nA	
Power-Fail Voltage	V_{PF}	1.216 x V_{BAT}	1.25 x V_{BAT}	1.284 x V_{BAT}	V	8

*Unless otherwise specified.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

AC ELECTRICAL CHARACTERISTICS

(Over the operating range*)

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	3
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	4,5
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	6
I/O Capacitance (T_A = 25°C)	$C_{I/O}$		10		pF	
Crystal Specified Load Capacitance (T_A = 25°C)			12.5		pF	

*Unless otherwise specified.

NOTES:

1. I_{CCS} specified with V_{CC} = 5.0V and SDA, SCL = 5.0V.
2. V_{CC} = 0V, V_{BAT} = 3V.
3. After this period, the first clock pulse is generated.
4. A device must internally provide a hold time of at least 300ns 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.
5. 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.
6. C_B – Total capacitance of one bus line in pF.
7. I_{CCA} – SCL clocking at max frequency = 100kHz.
8. V_{PF} measured at V_{BAT} = 3.0V.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

CONTENTS

1 FEATURES
2 GENERAL DESCRIPTION
3 ORDERING INFORMATION
4 BLOCK DIAGRAM
5 PINNING
5.1 DIP16 and SO16 packages
5.2 SSOP20 package
6 CHARACTERISTICS OF THE I²C-BUS
6.1 Bit transfer
6.2 Start and stop conditions
6.3 System configuration
6.4 Acknowledge
7 FUNCTIONAL DESCRIPTION
7.1 Addressing
7.2 Interrupt output
7.3 Quasi-bidirectional I/Os
8 LIMITING VALUES
9 HANDLING
10 DC CHARACTERISTICS
11 I²C-BUS TIMING CHARACTERISTICS
12 PACKAGE OUTLINES
13 SOLDERING
13.1 Introduction
13.2 Through-hole mount packages
13.2.1 Soldering by dipping or by solder wave
13.2.2 Manual soldering
13.3 Surface mount packages
13.3.1 Reflow soldering
13.3.2 Wave soldering
13.3.3 Manual soldering
13.4 Suitability of IC packages for wave, reflow and dipping soldering methods
14 DATA SHEET STATUS
15 DEFINITIONS
16 DISCLAIMERS
17 PURCHASE OF PHILIPS I²C COMPONENTS

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Remote 8-bit I/O expander for I²C-bus PCF8574

1 FEATURES

- Operating supply voltage 2.5 to 6 V
- Low standby current consumption of 10 μA maximum
- I²C-bus to parallel port expander
- Open-drain interrupt output
- 8-bit remote I/O port for the I²C-bus
- Compatible with most microcontrollers
- Latched outputs with high current drive capability for directly driving LEDs
- Address by 3 hardware address pins for use of up to 8 devices (up to 16 with PCF8574A)
- DIP16, or space-saving SO16 or SSOP20 packages.

2 GENERAL DESCRIPTION

The PCF8574 is a silicon CMOS circuit. It provides general purpose remote I/O expansion for most microcontroller families via the two-line bidirectional bus (I²C-bus).

The device consists of an 8-bit quasi-bidirectional port and an I²C-bus interface. The PCF8574 has a low current consumption and includes latched outputs with high current drive capability for directly driving LEDs. It also possesses an interrupt line ($\overline{INT}$) which can be connected to the interrupt logic of the microcontroller. By sending an interrupt signal on this line, the remote I/O can inform the microcontroller if there is incoming data on its ports without having to communicate via the I²C-bus. This means that the PCF8574 can remain a simple slave device.

The PCF8574 and PCF8574A versions differ only in their slave address as shown in Fig.10.

3 ORDERING INFORMATION

TYPE NUMBER	PACKAGE		
	NAME	DESCRIPTION	VERSION
PCF8574P; PCF8574AP	DIP16	plastic dual in-line package; 16 leads (300 mil); long body	SOT38-1
PCF8574T; PCF8574AT	SO16	plastic small outline package; 16 leads; body width 7.5 mm	SOT162-1
PCF8574TS; PCF8574ATS	SSOP20	plastic shrink small outline package; 20 leads; body width 4.4 mm	SOT266-1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4 BLOCK DIAGRAM

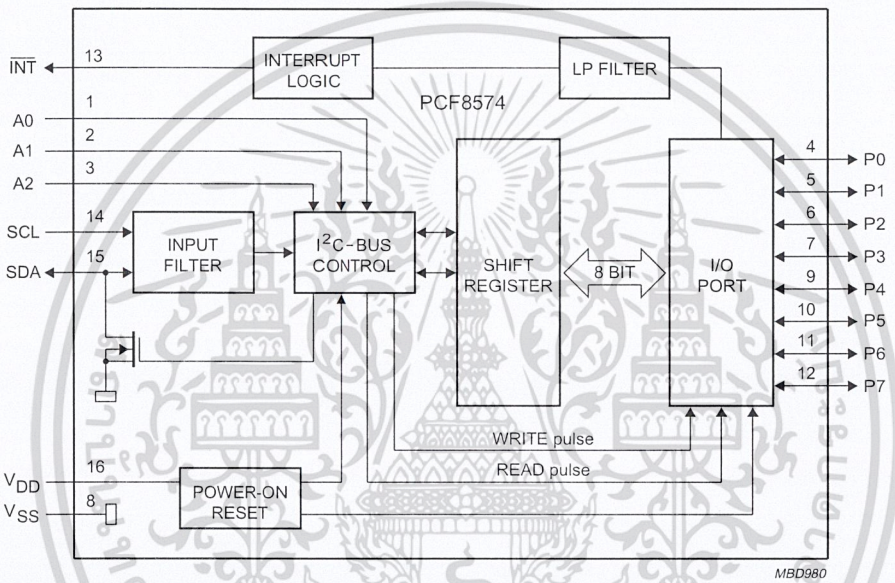


Fig.1 Block diagram (pin numbers apply to DIP16 and SO16 packages).

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5 PINNING

5.1 DIP16 and SO16 packages

SYMBOL	PIN	DESCRIPTION
A0	1	address input 0
A1	2	address input 1
A2	3	address input 2
P0	4	quasi-bidirectional I/O 0
P1	5	quasi-bidirectional I/O 1
P2	6	quasi-bidirectional I/O 2
P3	7	quasi-bidirectional I/O 3
V_{SS}	8	supply ground
P4	9	quasi-bidirectional I/O 4
P5	10	quasi-bidirectional I/O 5
P6	11	quasi-bidirectional I/O 6
P7	12	quasi-bidirectional I/O 7
$\overline{INT}$	13	interrupt output (active LOW)
SCL	14	serial clock line
SDA	15	serial data line
V_{DD}	16	supply voltage

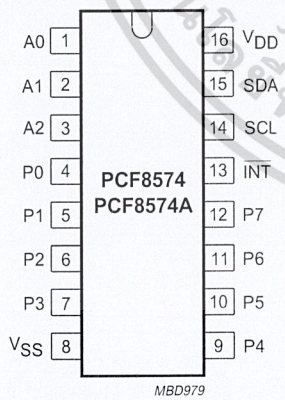


Fig.2 Pin configuration (DIP16).

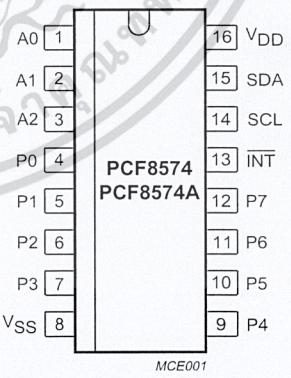


Fig.3 Pin configuration (SO16).

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.2 SSOP20 package

SYMBOL	PIN	DESCRIPTION
$\overline{INT}$	1	interrupt output (active LOW)
SCL	2	serial clock line
n.c.	3	not connected
SDA	4	serial data line
V_{DD}	5	supply voltage
A0	6	address input 0
A1	7	address input 1
n.c.	8	not connected
A2	9	address input 2
P0	10	quasi-bidirectional I/O 0
P1	11	quasi-bidirectional I/O 1
P2	12	quasi-bidirectional I/O 2
n.c.	13	not connected
P3	14	quasi-bidirectional I/O 3
V_{SS}	15	supply ground
P4	16	quasi-bidirectional I/O 4
P5	17	quasi-bidirectional I/O 5
n.c.	18	not connected
P6	19	quasi-bidirectional I/O 6
P7	20	quasi-bidirectional I/O 7

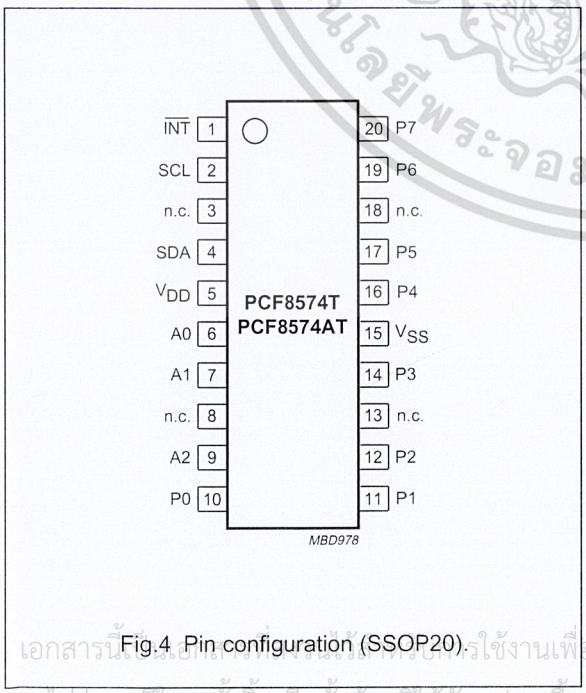


Fig.4 Pin configuration (SSOP20).

เอกสารนี้จัดทำขึ้นเพื่อใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6 CHARACTERISTICS OF THE I²C-BUS

The I²C-bus is for 2-way, 2-line communication between different ICs or modules. The two lines are a serial data line (SDA) and a serial clock line (SCL). Both lines must be connected to a positive supply via a pull-up resistor when connected to the output stages of a device. Data transfer may be initiated only when the bus is not busy.

6.1 Bit transfer

One data bit is transferred during each clock pulse. The data on the SDA line must remain stable during the HIGH period of the clock pulse as changes in the data line at this time will be interpreted as control signals (see Fig.5).

6.2 Start and stop conditions

Both data and clock lines remain HIGH when the bus is not busy. A HIGH-to-LOW transition of the data line, while the clock is HIGH is defined as the start condition (S).
A LOW-to-HIGH transition of the data line while the clock is HIGH is defined as the stop condition (P) (see Fig.6).

6.3 System configuration

A device generating a message is a 'transmitter', a device receiving is the 'receiver'. The device that controls the message is the 'master' and the devices which are controlled by the master are the 'slaves' (see Fig.7).

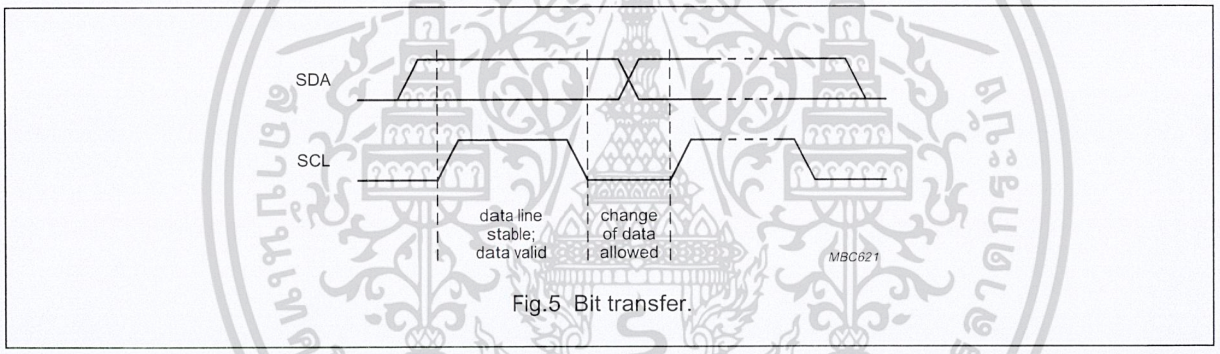


Fig.5 Bit transfer.

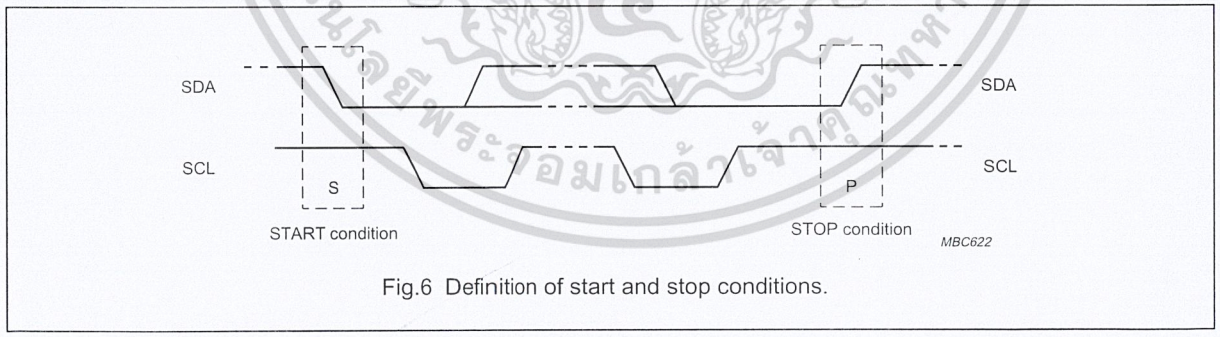


Fig.6 Definition of start and stop conditions.

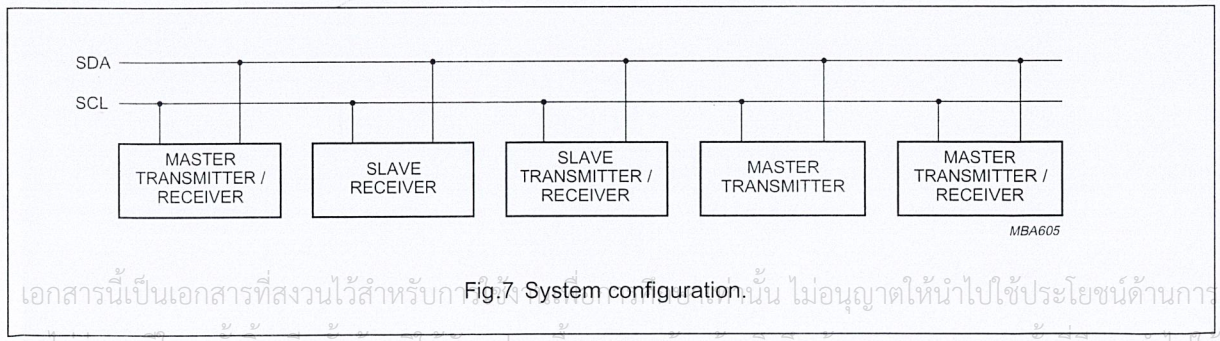


Fig.7 System configuration.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6.4 Acknowledge

The number of data bytes transferred between the start and the stop conditions from transmitter to receiver is not limited. Each byte of eight bits is followed by one acknowledge bit (see Fig.8). The acknowledge bit is a HIGH level put on the bus by the transmitter whereas the master generates an extra acknowledge related clock pulse.

A slave receiver which is addressed must generate an acknowledge after the reception of each byte. Also a master must generate an acknowledge after the reception of each byte that has been clocked out of the slave transmitter. The device that acknowledges has to pull down the SDA line during the acknowledge clock pulse, so that the SDA line is stable LOW during the HIGH period of the acknowledge related clock pulse, set-up and hold times must be taken into account.

A master receiver must signal an end of data to the transmitter by **not** generating an acknowledge on the last byte that has been clocked out of the slave. In this event the transmitter must leave the data line HIGH to enable the master to generate a stop condition.

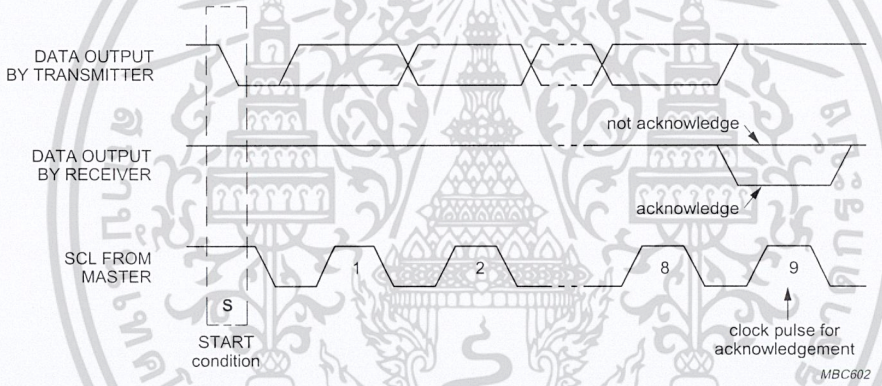


Fig.8 Acknowledgment on the I²C-bus.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

7 FUNCTIONAL DESCRIPTION

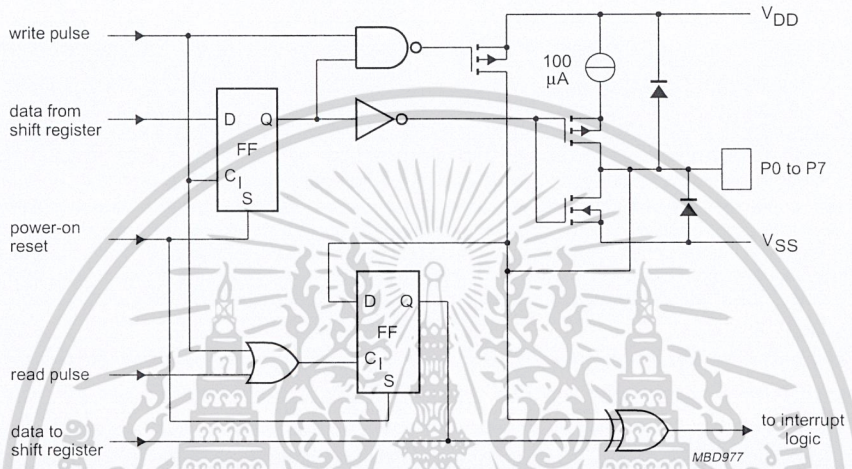
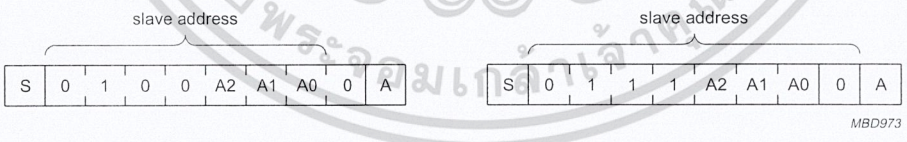


Fig.9 Simplified schematic diagram of each I/O.

7.1 Addressing

For addressing see Figs 10, 11 and 12.



a. PCF8574. b. PCF8574A.

Fig.10 PCF8574 and PCF8574A slave addresses.

Each of the PCF8574's eight I/Os can be independently used as an input or output. Input data is transferred from the port to the microcontroller by the READ mode (see Fig.12). Output data is transmitted to the port by the WRITE mode (see Fig.11).

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

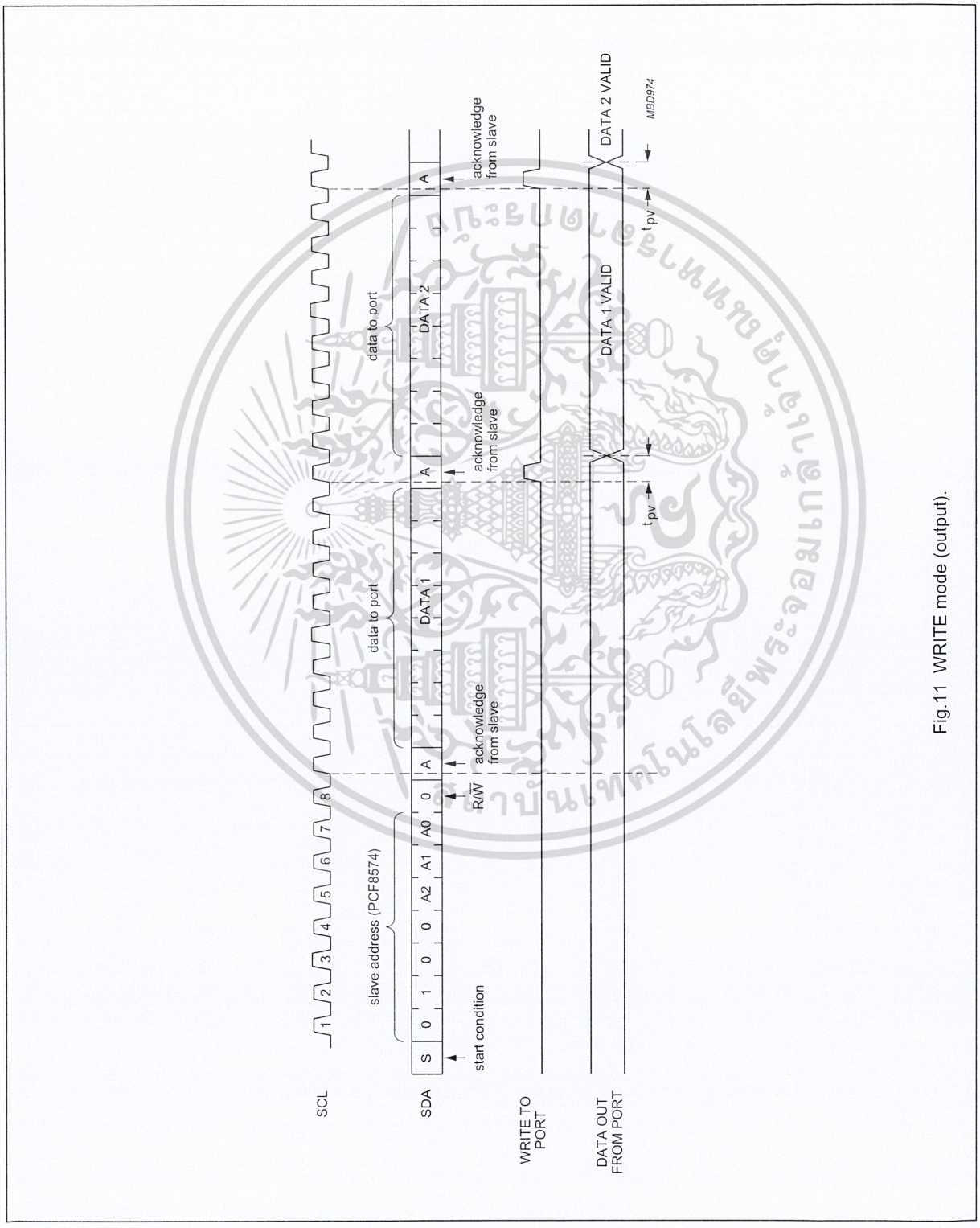
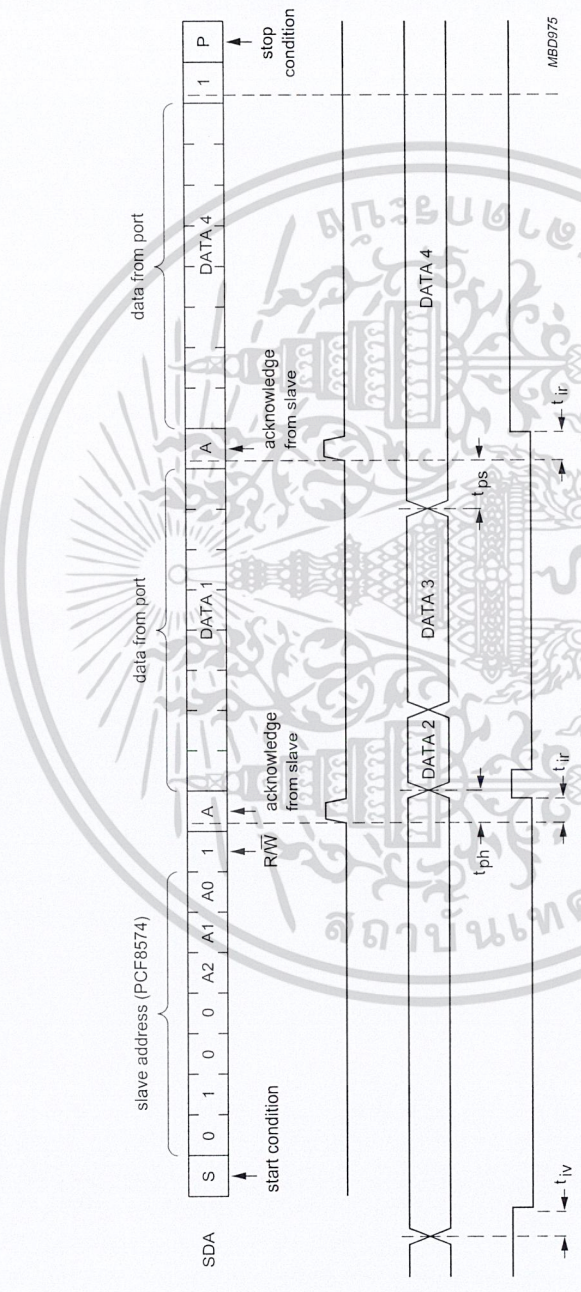


Fig.11 WRITE mode (output).



A LOW-to-HIGH transition of SDA, while SCL is HIGH is defined as the stop condition (P). Transfer of data can be stopped at any moment by a stop condition. When this occurs, data present at the last acknowledge phase is valid (output mode). Input data is lost.

Fig.12 READ mode (input).

7.2 Interrupt output

The PCF8574 provides an open-drain output ($\overline{INT}$) which can be fed to a corresponding input of the microcontroller (see Figs 13 and 14). This gives these chips a type of master function which can initiate an action elsewhere in the system.

An interrupt is generated by any rising or falling edge of the port inputs in the input mode. After time t_{iv} the signal $\overline{INT}$ is valid.

Resetting and reactivating the interrupt circuit is achieved when data on the port is changed to the original setting or data is read from or written to the port which has generated the interrupt.

Resetting occurs as follows:

- In the READ mode at the acknowledge bit after the rising edge of the SCL signal
- In the WRITE mode at the acknowledge bit after the HIGH-to-LOW transition of the SCL signal
- Interrupts which occur during the acknowledge clock pulse may be lost (or very short) due to the resetting of the interrupt during this pulse.

Each change of the I/Os after resetting will be detected and, after the next rising clock edge, will be transmitted as $\overline{INT}$. Reading from or writing to another device does not affect the interrupt circuit.

7.3 Quasi-bidirectional I/Os

A quasi-bidirectional I/O can be used as an input or output without the use of a control signal for data direction (see Fig.15). At power-on the I/Os are HIGH. In this mode only a current source to V_{DD} is active. An additional strong pull-up to V_{DD} allows fast rising edges into heavily loaded outputs. These devices turn on when an output is written HIGH, and are switched off by the negative edge of SCL. The I/Os should be HIGH before being used as inputs.

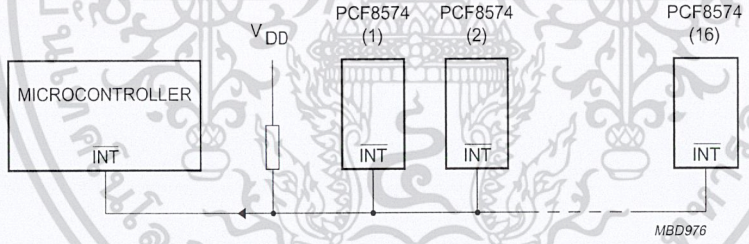


Fig.13 Application of multiple PCF8574s with interrupt.

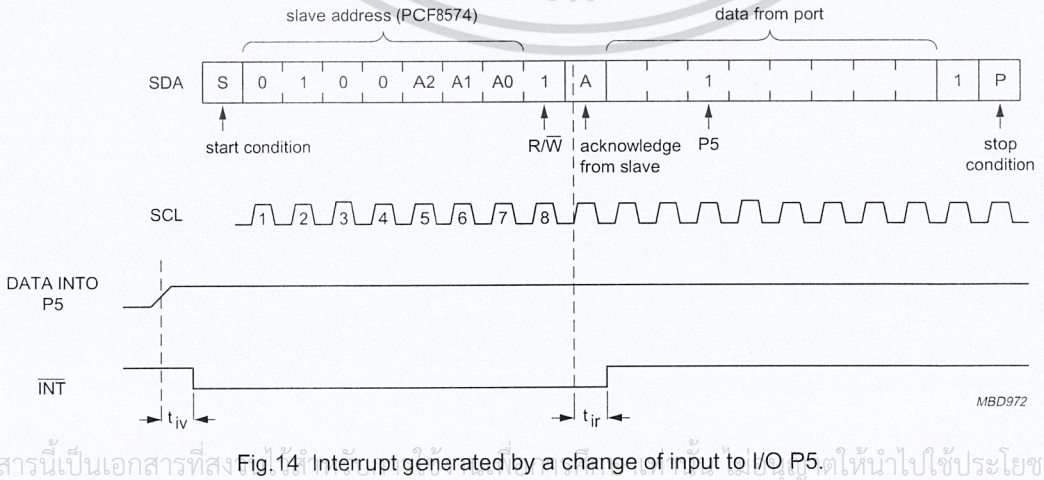


Fig.14 Interrupt generated by a change of input to I/O P5.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

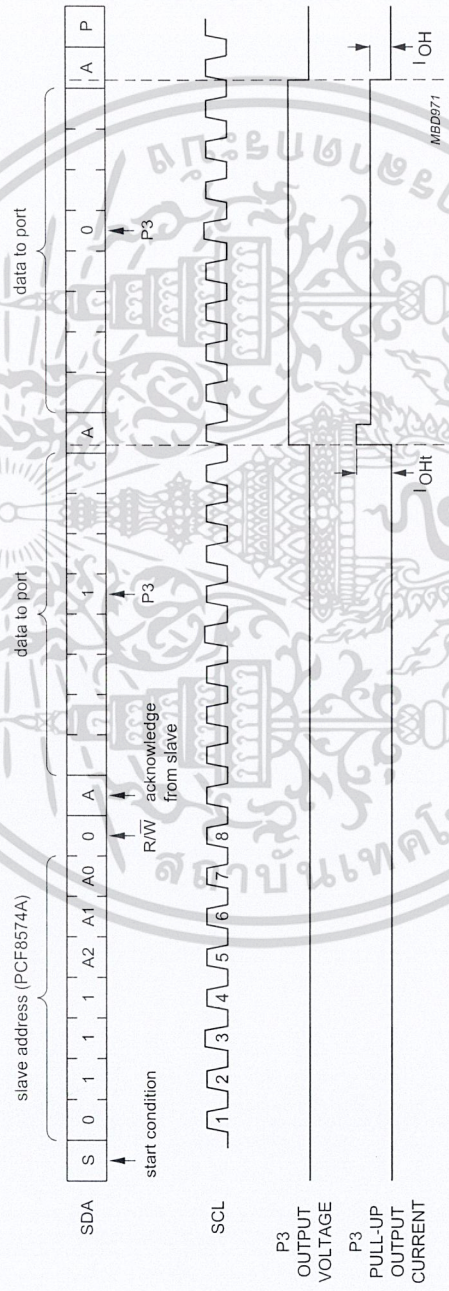


Fig.15 Transient pull-up current I_{OHt} while P3 changes from LOW-to-HIGH and back to LOW.

8 LIMITING VALUES

In accordance with the Absolute Maximum Rating System (IEC 60134).

SYMBOL	PARAMETER	MIN.	MAX.	UNIT
V_{DD}	supply voltage	−0.5	+7.0	V
V_I	input voltage	V_{SS} − 0.5	V_{DD} + 0.5	V
I_I	DC input current	−	±20	mA
I_O	DC output current	−	±25	mA
I_{DD}	supply current	−	±100	mA
I_{SS}	supply current	−	±100	mA
P_{tot}	total power dissipation	−	400	mW
P_O	power dissipation per output	−	100	mW
T_{stg}	storage temperature	−65	+150	°C
T_{amb}	ambient temperature	−40	+85	°C

9 HANDLING

Inputs and outputs are protected against electrostatic discharge in normal handling. However, to be totally safe, it is desirable to take precautions appropriate to handling MOS devices, see *"Handling MOS Devices"*.

10 DC CHARACTERISTICS

V_{DD} = 2.5 to 6 V; V_{SS} = 0 V; T_{amb} = −40 to +85 °C; unless otherwise specified.

SYMBOL	PARAMETER	CONDITIONS	MIN.	TYP.	MAX.	UNIT
Supply						
V_{DD}	supply voltage		2.5	−	6.0	V
I_{DD}	supply current	operating mode; V_{DD} = 6 V; no load; V_I = V_{DD} or V_{SS}; f_{SCL} = 100 kHz	−	40	100	µA
I_{stb}	standby current	standby mode; V_{DD} = 6 V; no load; V_I = V_{DD} or V_{SS}	−	2.5	10	µA
V_{POR}	Power-on reset voltage	V_{DD} = 6 V; no load; V_I = V_{DD} or V_{SS}; note 1	−	1.3	2.4	V
Input SCL; input/output SDA						
V_{IL}	LOW level input voltage		−0.5	−	+0.3V_{DD}	V
V_{IH}	HIGH level input voltage		0.7V_{DD}	−	V_{DD} + 0.5	V
I_{OL}	LOW level output current	V_{OL} = 0.4 V	3	−	−	mA
I_L	leakage current	V_I = V_{DD} or V_{SS}	−1	−	+1	µA
C_i	input capacitance	V_I = V_{SS}	−	−	7	pF

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

SYMBOL	PARAMETER	CONDITIONS	MIN.	TYP.	MAX.	UNIT
I/Os						
V_{IL}	LOW level input voltage		−0.5	−	$+0.3V_{DD}$	V
V_{IH}	HIGH level input voltage		$0.7V_{DD}$	−	V_{DD} + 0.5	V
$I_{IHL(max)}$	maximum allowed input current through protection diode	$V_I \geq V_{DD}$ or $V_I \leq V_{SS}$	−	−	±400	μA
I_{OL}	LOW level output current	V_{OL} = 1 V; V_{DD} = 5 V	10	25	−	mA
I_{OH}	HIGH level output current	$V_{OH} = V_{SS}$	30	−	300	μA
I_{OHt}	transient pull-up current	HIGH during acknowledge (see Fig.15); $V_{OH} = V_{SS}$; V_{DD} = 2.5 V	−	−1	−	mA
C_i	input capacitance		−	−	10	pF
C_o	output capacitance		−	−	10	pF
Port timing; $C_L \leq$ 100 pF (see Figs 11 and 12)						
t_{pv}	output data valid		−	−	4	μs
t_{su}	input data set-up time		0	−	−	μs
t_h	input data hold time		4	−	−	μs
Interrupt $\overline{INT}$ (see Fig.14)						
I_{OL}	LOW level output current	V_{OL} = 0.4 V	1.6	−	−	mA
I_L	leakage current	$V_I = V_{DD}$ or V_{SS}	−1	−	+1	μA
TIMING; $C_L \leq$ 100 pF						
t_{iv}	input data valid time		−	−	4	μs
t_{ir}	reset delay time		−	−	4	μs
Select inputs A0 to A2						
V_{IL}	LOW level input voltage		−0.5	−	$+0.3V_{DD}$	V
V_{IH}	HIGH level input voltage		$0.7V_{DD}$	−	V_{DD} + 0.5	V
I_{LI}	input leakage current	pin at V_{DD} or V_{SS}	−250	−	+250	nA

Note

1. The Power-on reset circuit resets the I²C-bus logic at $V_{DD} < V_{POR}$ and sets all I/Os to logic 1 (with current source to V_{DD}).

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

11 I²C-BUS TIMING CHARACTERISTICS

SYMBOL	PARAMETER	MIN.	TYP.	MAX.	UNIT
I²C-bus timing (see Fig.16; note 1)					
f_{SCL}	SCL clock frequency	–	–	100	kHz
t_{SW}	tolerable spike width on bus	–	–	100	ns
t_{BUF}	bus free time	4.7	–	–	µs
$t_{SU;STA}$	START condition set-up time	4.7	–	–	µs
$t_{HD;STA}$	START condition hold time	4.0	–	–	µs
t_{LOW}	SCL LOW time	4.7	–	–	µs
t_{HIGH}	SCL HIGH time	4.0	–	–	µs
t_r	SCL and SDA rise time	–	–	1.0	µs
t_f	SCL and SDA fall time	–	–	0.3	µs
$t_{SU;DAT}$	data set-up time	250	–	–	ns
$t_{HD;DAT}$	data hold time	0	–	–	ns
$t_{VD;DAT}$	SCL LOW to data out valid	–	–	3.4	µs
$t_{SU;STO}$	STOP condition set-up time	4.0	–	–	µs

Note

1. All the timing values are valid within the operating supply voltage and ambient temperature range and refer to V_{IL} and V_{IH} with an input voltage swing of V_{SS} to V_{DD}.

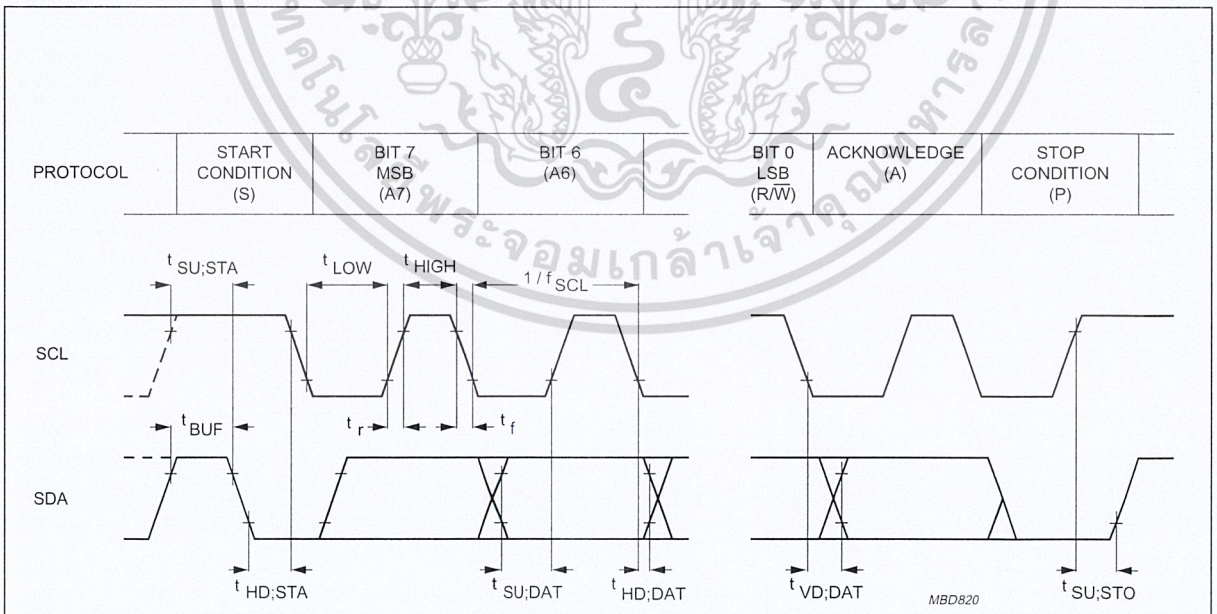
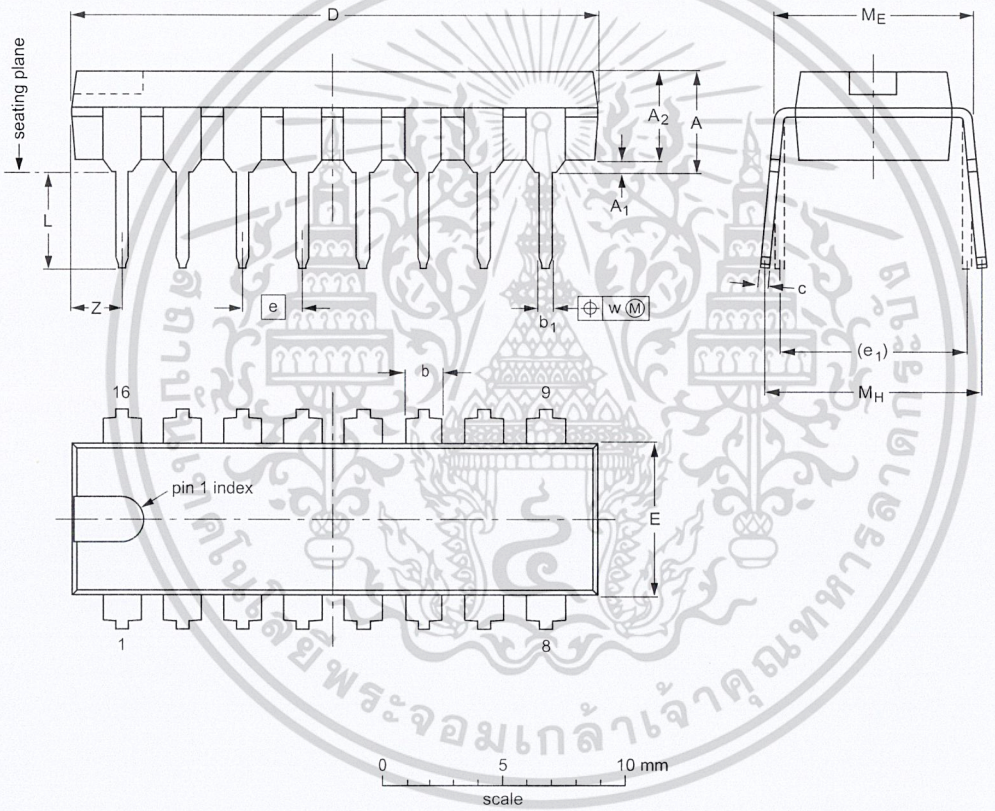


Fig.16 I²C-bus timing diagram.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

12 PACKAGE OUTLINES

DIP16: plastic dual in-line package; 16 leads (300 mil); long body **SOT38-1**



DIMENSIONS (inch dimensions are derived from the original mm dimensions)

UNIT	A max.	A_1 min.	A_2 max.	b	b_1	c	D (1)	E (1)	e	e_1	L	M_E	M_H	w	Z (1) max.
mm	4.7	0.51	3.7	1.40 1.14	0.53 0.38	0.32 0.23	21.8 21.4	6.48 6.20	2.54	7.62	3.9 3.4	8.25 7.80	9.5 8.3	0.254	2.2
inches	0.19	0.020	0.15	0.055 0.045	0.021 0.015	0.013 0.009	0.86 0.84	0.26 0.24	0.10	0.30	0.15 0.13	0.32 0.31	0.37 0.33	0.01	0.087

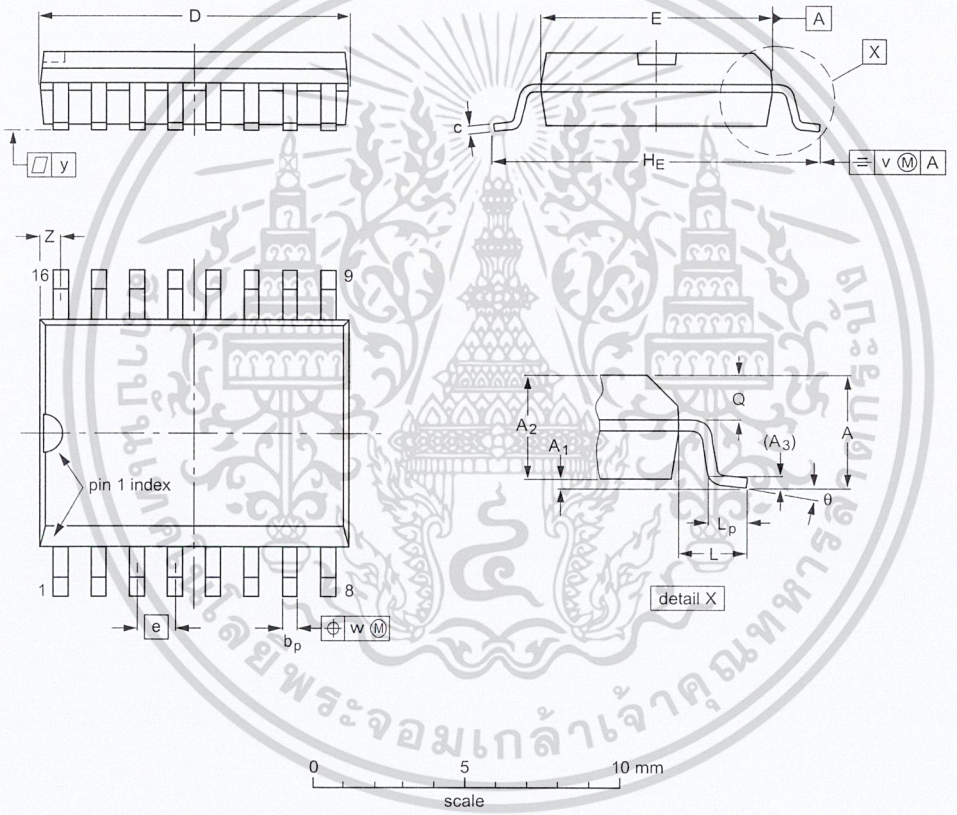
Note

1. Plastic or metal protrusions of 0.25 mm maximum per side are not included.

OUTLINE VERSION	REFERENCES				EUROPEAN PROJECTION	ISSUE DATE
	IEC	JEDEC	EIAJ			
SOT38-1	050G09	MO-001	SC-503-16			~~95-01-19~~ 99-12-27

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

SO16: plastic small outline package; 16 leads; body width 7.5 mm **SOT162-1**



DIMENSIONS (inch dimensions are derived from the original mm dimensions)

UNIT	A max.	A_1	A_2	A_3	b_p	c	$D^{(1)}$	$E^{(1)}$	e	H_E	L	L_p	Q	v	w	y	$Z^{(1)}$	θ
mm	2.65	0.30 0.10	2.45 2.25	0.25	0.49 0.36	0.32 0.23	10.5 10.1	7.6 7.4	1.27	10.65 10.00	1.4	1.1 0.4	1.1 1.0	0.25	0.25	0.1	0.9 0.4	8° 0°
inches	0.10	0.012 0.004	0.096 0.089	0.01	0.019 0.014	0.013 0.009	0.41 0.40	0.30 0.29	0.050	0.419 0.394	0.055	0.043 0.016	0.043 0.039	0.01	0.01	0.004	0.035 0.016	

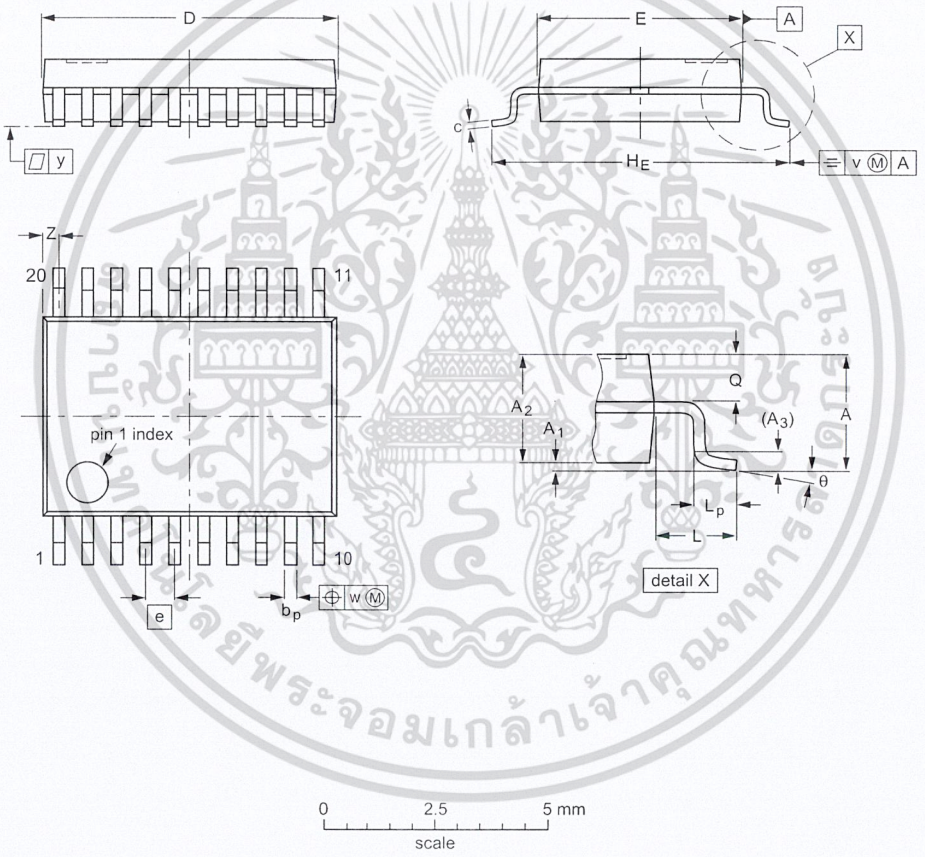
Note

1. Plastic or metal protrusions of 0.15 mm maximum per side are not included.

OUTLINE VERSION	REFERENCES				EUROPEAN PROJECTION	ISSUE DATE
	IEC	JEDEC	EIAJ			
SOT162-1	075E03	MS-013				~~97-05-22~~ 99-12-27

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

SSOP20: plastic shrink small outline package; 20 leads; body width 4.4 mm **SOT266-1**



DIMENSIONS (mm are the original dimensions)

UNIT	A max.	A_1	A_2	A_3	b_p	c	$D^{(1)}$	$E^{(1)}$	e	H_E	L	L_p	Q	v	w	y	$Z^{(1)}$	θ
mm	1.5	0.15 0	1.4 1.2	0.25	0.32 0.20	0.20 0.13	6.6 6.4	4.5 4.3	0.65	6.6 6.2	1.0	0.75 0.45	0.65 0.45	0.2	0.13	0.1	0.48 0.18	10^o 0^o

Note

1. Plastic or metal protrusions of 0.20 mm maximum per side are not included.

OUTLINE VERSION	REFERENCES				EUROPEAN PROJECTION	ISSUE DATE
	IEC	JEDEC	EIAJ			
SOT266-1		MO-152				~~95-02-22~~ 99-12-27

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

13 SOLDERING

13.1 Introduction

This text gives a very brief insight to a complex technology. A more in-depth account of soldering ICs can be found in our *"Data Handbook IC26; Integrated Circuit Packages"* (document order number 9398 652 90011).

There is no soldering method that is ideal for all IC packages. Wave soldering is often preferred when through-hole and surface mount components are mixed on one printed-circuit board. Wave soldering can still be used for certain surface mount ICs, but it is not suitable for fine pitch SMDs. In these situations reflow soldering is recommended.

13.2 Through-hole mount packages

13.2.1 Soldering by dipping or by solder wave

The maximum permissible temperature of the solder is 260 °C; solder at this temperature must not be in contact with the joints for more than 5 seconds. The total contact time of successive solder waves must not exceed 5 seconds.

The device may be mounted up to the seating plane, but the temperature of the plastic body must not exceed the specified maximum storage temperature ($T_{stg(max)}$). If the printed-circuit board has been pre-heated, forced cooling may be necessary immediately after soldering to keep the temperature within the permissible limit.

13.2.2 Manual soldering

Apply the soldering iron (24 V or less) to the lead(s) of the package, either below the seating plane or not more than 2 mm above it. If the temperature of the soldering iron bit is less than 300 °C it may remain in contact for up to 10 seconds. If the bit temperature is between 300 and 400 °C, contact may be up to 5 seconds.

13.3 Surface mount packages

13.3.1 Reflow soldering

Reflow soldering requires solder paste (a suspension of fine solder particles, flux and binding agent) to be applied to the printed-circuit board by screen printing, stencilling or pressure-syringe dispensing before package placement.

Several methods exist for reflowing; for example, convection or convection/infrared heating in a conveyor type oven. Throughput times (preheating, soldering and cooling) vary between 100 and 200 seconds depending on heating method.

Typical reflow peak temperatures range from 215 to 250 °C. The top-surface temperature of the packages should preferable be kept below 220 °C for thick/large packages, and below 235 °C for small/thin packages.

13.3.2 Wave soldering

Conventional single wave soldering is not recommended for surface mount devices (SMDs) or printed-circuit boards with a high component density, as solder bridging and non-wetting can present major problems.

To overcome these problems the double-wave soldering method was specifically developed.

If wave soldering is used the following conditions must be observed for optimal results:

- Use a double-wave soldering method comprising a turbulent wave with high upward pressure followed by a smooth laminar wave.
- For packages with leads on two sides and a pitch (e):
  - larger than or equal to 1.27 mm, the footprint longitudinal axis is **preferred** to be parallel to the transport direction of the printed-circuit board;
  - smaller than 1.27 mm, the footprint longitudinal axis **must** be parallel to the transport direction of the printed-circuit board.

  The footprint must incorporate solder thieves at the downstream end.
- For packages with leads on four sides, the footprint must be placed at a 45° angle to the transport direction of the printed-circuit board. The footprint must incorporate solder thieves downstream and at the side corners.

During placement and before soldering, the package must be fixed with a droplet of adhesive. The adhesive can be applied by screen printing, pin transfer or syringe dispensing. The package can be soldered after the adhesive is cured.

Typical dwell time is 4 seconds at 250 °C.
A mildly-activated flux will eliminate the need for removal of corrosive residues in most applications.

13.3.3 Manual soldering

Fix the component by first soldering two diagonally-opposite end leads. Use a low voltage (24 V or less) soldering iron applied to the flat part of the lead. Contact time must be limited to 10 seconds at up to 300 °C. When using a dedicated tool, all other leads can be soldered in one operation within 2 to 5 seconds between 270 and 320 °C.

เอกสารนี้สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

13.4 Suitability of IC packages for wave, reflow and dipping soldering methods

MOUNTING	PACKAGE[1]	SOLDERING METHOD		
		WAVE	REFLOW[2]	DIPPING
Through-hole mount	DBS, DIP, HDIP, SDIP, SIL	suitable[3]	–	suitable
Surface mount	BGA, LBGA, LFBGA, SQFP, TFBGA, VFBGA	not suitable	suitable	–
	HBCC, HBGA, HLQFP, HSQFP, HSOP, HTQFP, HTSSOP, HVQFN, HVSON, SMS	not suitable[4]	suitable	–
	PLCC[5], SO, SOJ	suitable	suitable	–
	LQFP, QFP, TQFP	not recommended[5][6]	suitable	–
	SSOP, TSSOP, VSO	not recommended[7]	suitable	–

Notes

1. For more detailed information on the BGA packages refer to the *"(LF)BGA Application Note"* (AN01026); order a copy from your Philips Semiconductors sales office.
2. All surface mount (SMD) packages are moisture sensitive. Depending upon the moisture content, the maximum temperature (with respect to time) and body size of the package, there is a risk that internal or external package cracks may occur due to vaporization of the moisture in them (the so called popcorn effect). For details, refer to the Drypack information in the *"Data Handbook IC26; Integrated Circuit Packages; Section: Packing Methods"*.
3. For SDIP packages, the longitudinal axis must be parallel to the transport direction of the printed-circuit board.
4. These packages are not suitable for wave soldering. On versions with the heatsink on the bottom side, the solder cannot penetrate between the printed-circuit board and the heatsink. On versions with the heatsink on the top side, the solder might be deposited on the heatsink surface.
5. If wave soldering is considered, then the package must be placed at a 45° angle to the solder wave direction. The package footprint must incorporate solder thieves downstream and at the side corners.
6. Wave soldering is suitable for LQFP, QFP and TQFP packages with a pitch (e) larger than 0.8 mm; it is definitely not suitable for packages with a pitch (e) equal to or smaller than 0.65 mm.
7. Wave soldering is suitable for SSOP and TSSOP packages with a pitch (e) equal to or larger than 0.65 mm; it is definitely not suitable for packages with a pitch (e) equal to or smaller than 0.5 mm.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

14 DATA SHEET STATUS

DATA SHEET STATUS(1)	PRODUCT STATUS(2)	DEFINITIONS
Objective data	Development	This data sheet contains data from the objective specification for product development. Philips Semiconductors reserves the right to change the specification in any manner without notice.
Preliminary data	Qualification	This data sheet contains data from the preliminary specification. Supplementary data will be published at a later date. Philips Semiconductors reserves the right to change the specification without notice, in order to improve the design and supply the best possible product.
Product data	Production	This data sheet contains data from the product specification. Philips Semiconductors reserves the right to make changes at any time in order to improve the design, manufacturing and supply. Changes will be communicated according to the Customer Product/Process Change Notification (CPCN) procedure SNW-SQ-650A.

Notes

1. Please consult the most recently issued data sheet before initiating or completing a design.
2. The product status of the device(s) described in this data sheet may have changed since this data sheet was published. The latest information is available on the Internet at URL http://www.semiconductors.philips.com.

15 DEFINITIONS

Short-form specification — The data in a short-form specification is extracted from a full data sheet with the same type number and title. For detailed information see the relevant data sheet or data handbook.

Limiting values definition — Limiting values given are in accordance with the Absolute Maximum Rating System (IEC 60134). Stress above one or more of the limiting values may cause permanent damage to the device. These are stress ratings only and operation of the device at these or at any other conditions above those given in the Characteristics sections of the specification is not implied. Exposure to limiting values for extended periods may affect device reliability.

Application information — Applications that are described herein for any of these products are for illustrative purposes only. Philips Semiconductors make no representation or warranty that such applications will be suitable for the specified use without further testing or modification.

16 DISCLAIMERS

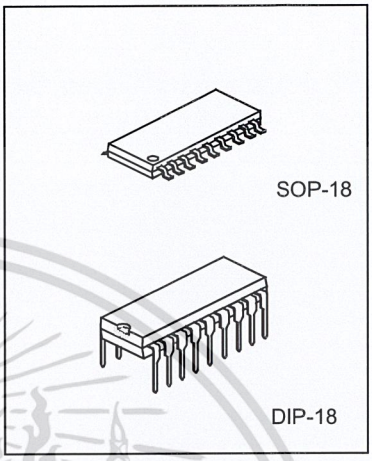
Life support applications — These products are not designed for use in life support appliances, devices, or systems where malfunction of these products can reasonably be expected to result in personal injury. Philips Semiconductors customers using or selling these products for use in such applications do so at their own risk and agree to fully indemnify Philips Semiconductors for any damages resulting from such application.

Right to make changes — Philips Semiconductors reserves the right to make changes, without notice, in the products, including circuits, standard cells, and/or software, described or contained herein in order to improve design and/or performance. Philips Semiconductors assumes no responsibility or liability for the use of any of these products, conveys no licence or title under any patent, copyright, or mask work right to these products, and makes no representations or warranties that these products are free from patent, copyright, or mask work right infringement, unless otherwise specified.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

UTC ULN2803 LINEAR INTEGRATED CIRCUIT

EIGHT DARLINGTON ARRAYS



DESCRIPTION

The UTC ULN2803 is high-voltage, high-current Darlington drivers comprised of eight NPN Darlington pairs.

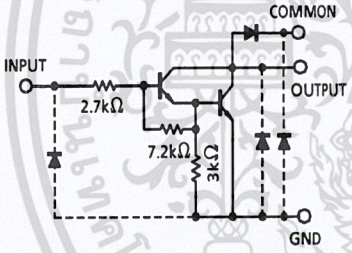
FEATURES

*Output current (single output) 500mA MAX.
*High sustaining voltage output 50V MIN.
*Output clamp diodes
*Inputs compatible with various types of logic

APPLICATIONS

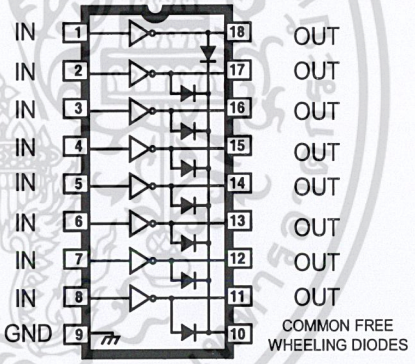
*Relay, hammer, lamp and display (LED) drivers.

SCHEMATICS (EACH DRIVER)



Note: The input and output parasitic diodes cannot be used as clamp diodes.

PIN CONFIGURATIONS



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ABSOLUTE MAXIMUM RATINGS

PARAMETER	SYMBOL	VALUE	UNIT
Input Voltage	V_{IN}	-0.5~30	V
Output Sustaining Voltage	$V_{CE\ (SUS)}$	-0.5~50	V
Output Current	I_{OUT}	500	mA/ch
Clamp Diode Reverse Voltage	VR	50	V
Clamp Diode Forward Current	IF	500	mA
Power Dissipation	P_D	DIP: 1.47 SOP:0.54/0.625(Note)	W
Operating Ambient Temperature Range	T_{opr}	-40 to +85	°C
Storage Temperature Range	T_{Stg}	-55 to +150	°C

Note: On glass epoxy PCB (30x30x1.6mm Cu 50%)

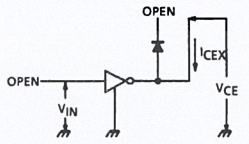
ELECTRICAL CHARACTERISTICS (Ta = 25 °C, Unless otherwise specified)

PARAMETER	SYMBOL	TEST CONDITIONS	MIN	TYP	MAX	UNIT	FIG
Output Leakage Current	I_{CEX}	V_{CE}=50V,T_A=25°C V_{CE}=50V,T_A=85°C			50 100	μA	1
Collector-Emitter Saturation Voltage	$V_{CE(SAT)}$	I_{OUT}=350mA,I_{IN}=500μA I_{OUT}=200mA,I_{IN}=350μA I_{OUT}=100mA,I_{IN}=250μA		1.3 1.1 0.9	1.6 1.3 1.1	V	2
Input Current (output on)	$I_{IN(ON)}$	V_{IN} =3.85V,I_{OUT}=350mA		0.93	1.35	mA	3
Input Current (output off)	$I_{IN(OFF)}$	I_{OUT}=500μA,T_A=85°C	50	65		μA	4
Input Voltage (output on)	$V_{IN(ON)}$	V_{CE}=2.0V I_{OUT}=200mA I_{OUT}=250mA I_{OUT}=300mA			 2.4 2.7 3.0	V	5
Clamp Diode Reverse Current	IR	VR=50V, T_A=25°C VR=50V, T_A=85°C			50 100	μA	6
Clamp Diode Forward Voltage	VF	IF=350mA			2.0	V	7
Input Capacitance	C_{IN}			15	25	pF	-
Turn-On Delay	t_{ON}	V_{OUT}=50V,RL=125Ω,CL=15pF		0.1	1	μS	8
Turn-Off Delay	t_{OFF}	V_{OUT}=50V,RL=125Ω,CL=15pF		0.2	1	μS	8

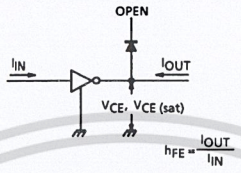
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

TEST CIRCUIT

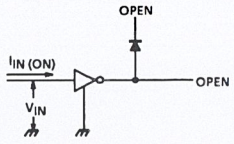
1. I_{CEX}



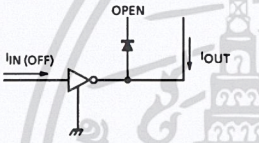
2. $V_{CE\,(sat)}$, h_{FE}



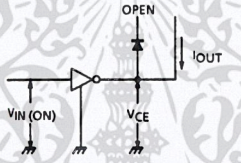
3. $I_{IN\,(ON)}$



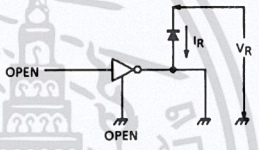
4. $I_{IN\,(OFF)}$



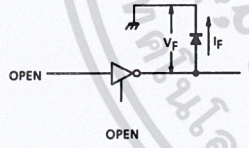
5. $V_{IN\,(ON)}$



6. I_R

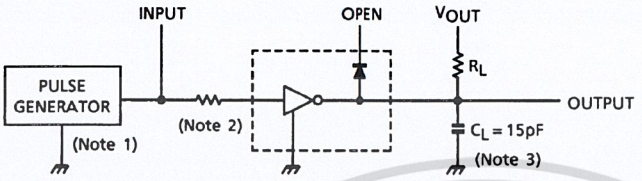


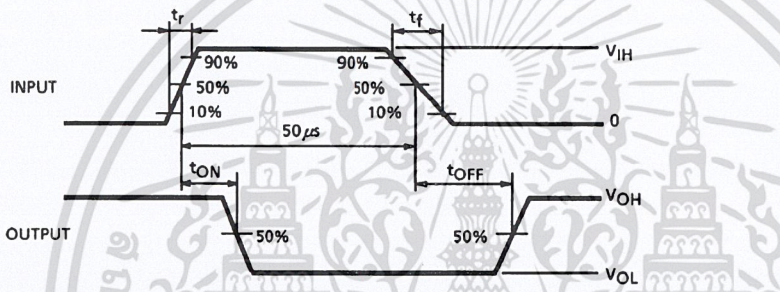
7. V_F



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

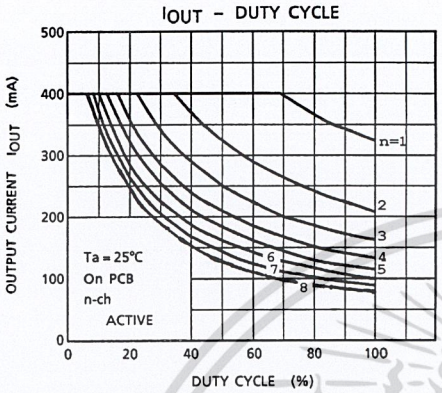
8. tON, tOFF

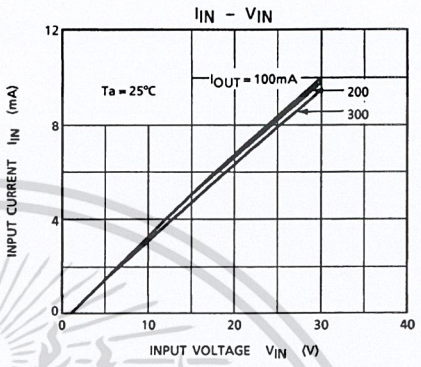


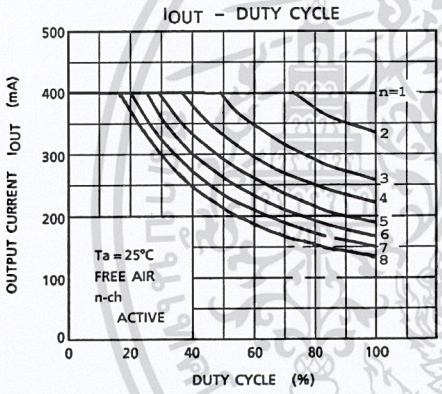


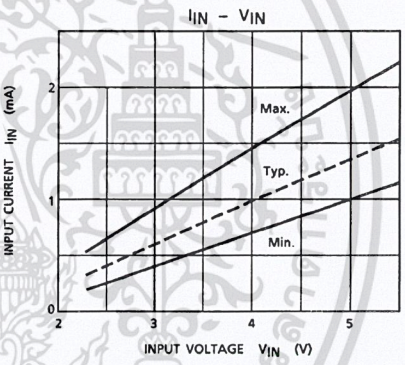
Note1: Pulse width 50μs, duty cycle 10%
Output impedance 50Ω, tr<=5ns, tf<=10ns
Note2: R_1: 0, V_{IH} : 3V
Note3: CL includes probe and jig capacitance.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

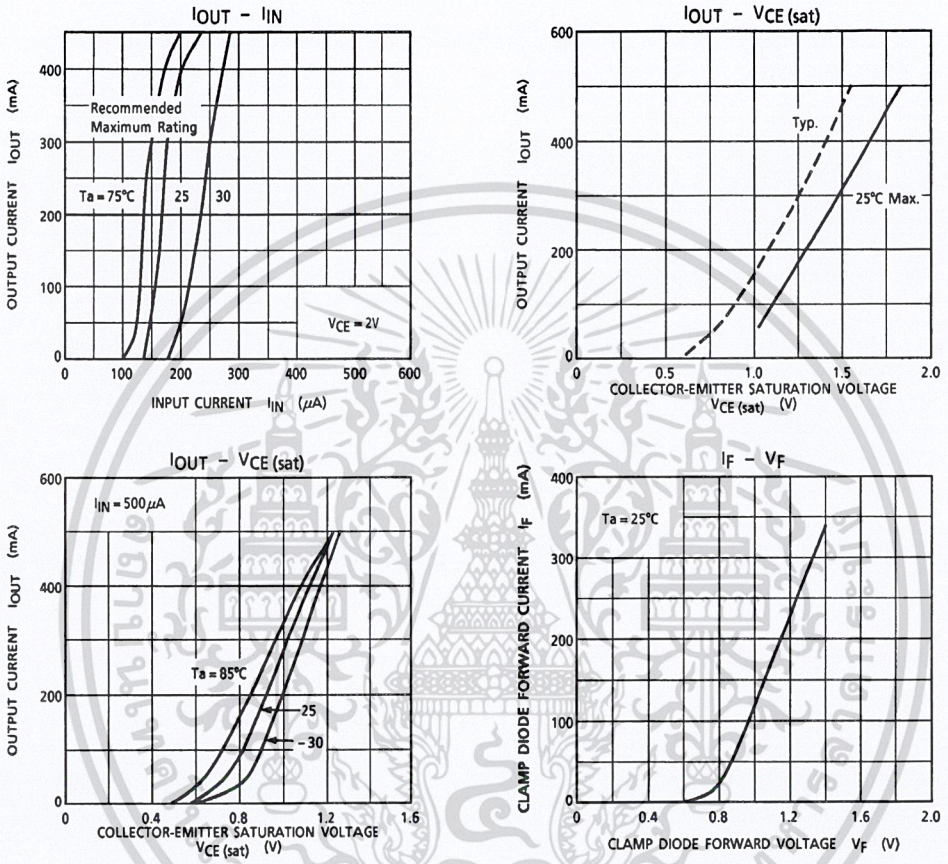




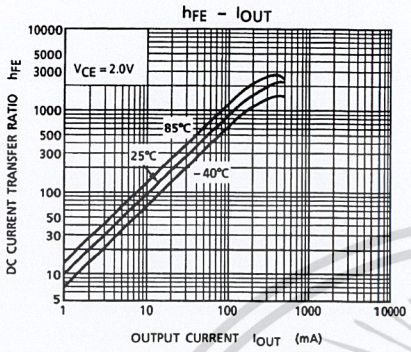


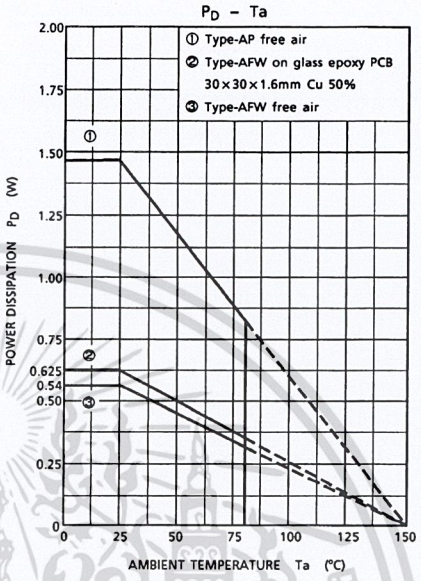


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้





เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

FAIRCHILD SEMICONDUCTOR™

October 1987
Revised January 1999

MM74C922 • MM74C923
16-Key Encoder • 20-Key Encoder

General Description

The MM74C922 and MM74C923 CMOS key encoders provide all the necessary logic to fully encode an array of SPST switches. The keyboard scan can be implemented by either an external clock or external capacitor. These encoders also have on-chip pull-up devices which permit switches with up to 50 kΩ on resistance to be used. No diodes in the switch array are needed to eliminate ghost switches. The internal debounce circuit needs only a single external capacitor and can be defeated by omitting the capacitor. A Data Available output goes to a high level when a valid keyboard entry has been made. The Data Available output returns to a low level when the entered key is released, even if another key is depressed. The Data Available will return high to indicate acceptance of the new key after a normal debounce period; this two-key roll-over is provided between any two switches.

An internal register remembers the last key pressed even after the key is released. The 3-STATE outputs provide for easy expansion and bus operation and are LPTTL compatible.

Features

- 50 kΩ maximum switch on resistance
- On or off chip clock
- On-chip row pull-up devices
- 2 key roll-over
- Keybounce elimination with single capacitor
- Last key register at outputs
- 3-STATE output LPTTL compatible
- Wide supply range: 3V to 15V
- Low power consumption

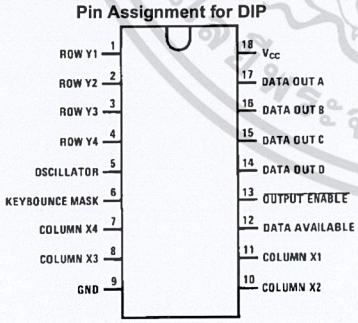
Ordering Code:

Order Number	Package Number	Package Description
MM74C922N	N18A	18-Lead Plastic Dual-In-Line Package (PDIP), JEDEC MS-001, 0.300" Wide
MM74C922WM	M20B	20-Lead Small Outline Integrated Circuit (SOIC), JEDEC MS-013, 0.300" Wide
MM74C923WM	M20B	20-Lead Small Outline Integrated Circuit (SOIC), JEDEC MS-013, 0.300" Wide
MM74C923N	N20A	20-Lead Plastic Dual-In-Line Package (PDIP), JEDEC MS-001, 0.300" Wide

Device also available in Tape and Reel. Specify by appending suffix letter "X" to the ordering code.

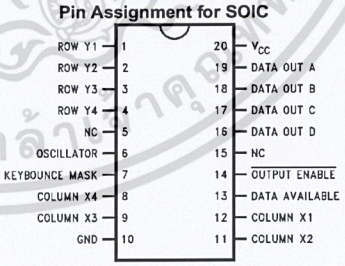
Connection Diagrams

Pin Assignment for DIP



Top View
MM94C922

Pin Assignment for SOIC



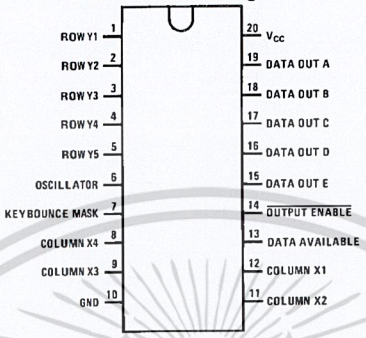
Top View
MM74C922

© 1999 Fairchild Semiconductor Corporation

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Connection Diagrams (Continued)

Pin Assignment for
DIP and SOIC Package



Top View
MM74C923

Truth Tables

(Pins 0 through 11)

Switch Position		0 Y1,X1	1 Y1,X2	2 Y1,X3	3 Y1,X4	4 Y2,X1	5 Y2,X2	6 Y2,X3	7 Y2,X4	8 Y3,X1	9 Y3,X2	10 Y3,X3	11 Y3,X4
DATA OUT	A	0	1	0	1	0	1	0	1	0	1	0	1
	B	0	0	1	1	0	0	1	1	0	0	1	1
	C	0	0	0	0	1	1	1	1	0	0	0	0
	D	0	0	0	0	0	0	0	0	1	1	1	1
	E (Note 1)	0	0	0	0	0	0	0	0	0	0	0	0

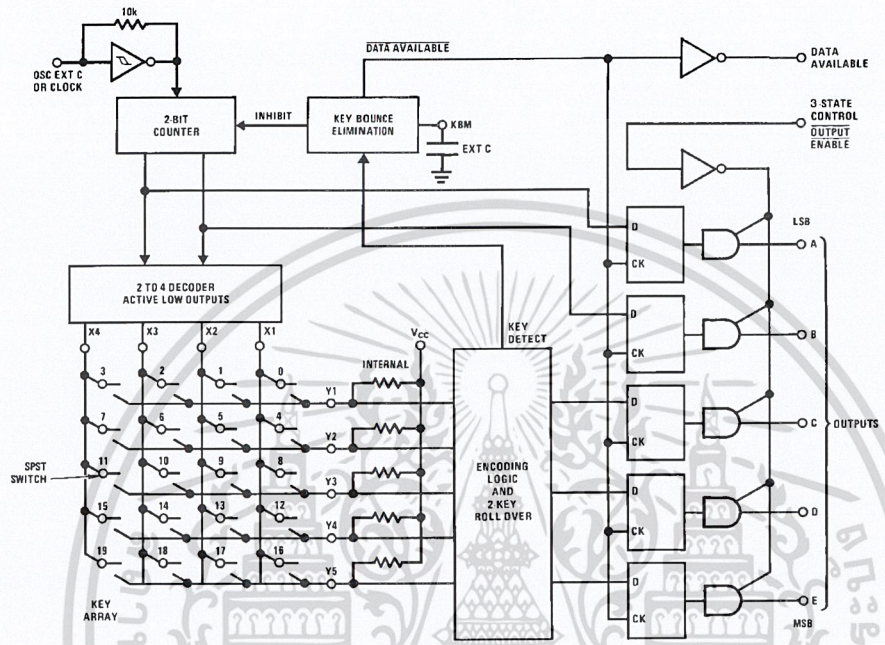
(Pins 12 through 19)

Switch Position		12 Y4,X1	13 Y4,X2	14 Y4,X3	15 Y4,X4	16 Y5(Note 1), X1	17 Y5 (Note 1), X2	18 Y5 (Note 1), X3	19 Y5 (Note 1), X4
DATA OUT	A	0	1	0	1	0	1	0	1
	B	0	0	1	1	0	0	1	1
	C	1	1	1	1	0	0	0	0
	D	1	1	1	1	0	0	0	0
	E (Note 1)	0	0	0	0	1	1	1	1

Note 1: Omit for MM74C922

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Block Diagram



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Absolute Maximum Ratings (Note 2)

Voltage at Any Pin	V_{CC} − 0.3V to V_{CC} + 0.3V
Operating Temperature Range	
MM74C922, MM74C923	−40°C to +85°C
Storage Temperature Range	−65°C to +150°C
Power Dissipation (P_D)	
Dual-In-Line	700 mW
Small Outline	500 mW
Operating V_{CC} Range	3V to 15V
V_{CC}	18V
Lead Temperature	
(Soldering, 10 seconds)	260°C

Note 2: "Absolute Maximum Ratings" are those values beyond which the safety of the device cannot be guaranteed. Except for "Operating Temperature Range" they are not meant to imply that the devices should be operated at these limits. The table of "Electrical Characteristics" provides conditions for actual device operation.

DC Electrical Characteristics

Min/Max limits apply across temperature range unless otherwise specified

Symbol	Parameter	Conditions	Min	Typ	Max	Units
CMOS TO CMOS						
V_{T+}	Positive-Going Threshold Voltage at Osc and KBM Inputs	V_{CC} = 5V, $I_{IN} \geq$ 0.7 mA	3.0	3.6	4.3	V
		V_{CC} = 10V, $I_{IN} \geq$ 1.4 mA	6.0	6.8	8.6	V
		V_{CC} = 15V, $I_{IN} \geq$ 2.1 mA	9.0	10	12.9	V
V_{T-}	Negative-Going Threshold Voltage at Osc and KBM Inputs	V_{CC} = 5V, $I_{IN} \geq$ 0.7 mA	0.7	1.4	2.0	V
		V_{CC} = 10V, $I_{IN} \geq$ 1.4 mA	1.4	3.2	4.0	V
		V_{CC} = 15V, $I_{IN} \geq$ 2.1 mA	2.1	5	6.0	V
$V_{IN(1)}$	Logical "1" Input Voltage, Except Osc and KBM Inputs	V_{CC} = 5V	3.5	4.5		V
		V_{CC} = 10V	8.0	9		V
		V_{CC} = 15V	12.5	13.5		V
$V_{IN(0)}$	Logical "0" Input Voltage, Except Osc and KBM Inputs	V_{CC} = 5V		0.5	1.5	V
		V_{CC} = 10V		1	2	V
		V_{CC} = 15V		1.5	2.5	V
I_{rp}	Row Pull-Up Current at Y1, Y2, Y3, Y4 and Y5 Inputs	V_{CC} = 5V, V_{IN} = 0.1 V_{CC}		−2	−5	μA
		V_{CC} = 10V		−10	−20	μA
		V_{CC} = 15V		−22	−45	μA
$V_{OUT(1)}$	Logical "1" Output Voltage	V_{CC} = 5V, I_O = −10 μA	4.5			V
		V_{CC} = 10V, I_O = −10 μA	9			V
		V_{CC} = 15V, I_O = −10 μA	13.5			V
$V_{OUT(0)}$	Logical "0" Output Voltage	V_{CC} = 5V, I_O = 10 μA			0.5	V
		V_{CC} = 10V, I_O = 10 μA			1	V
		V_{CC} = 15V, I_O = 10 μA			1.5	V
R_{on}	Column "ON" Resistance at X1, X2, X3 and X4 Outputs	V_{CC} = 5V, V_O = 0.5V		500	1400	Ω
		V_{CC} = 10V, V_O = 1V		300	700	Ω
		V_{CC} = 15V, V_O = 1.5V		200	500	Ω
I_{CC}	Supply Current Osc at 0V, (one Y low)	V_{CC} = 5V		0.55	1.1	mA
		V_{CC} = 10V		1.1	1.9	mA
		V_{CC} = 15V		1.7	2.6	mA
$I_{IN(1)}$	Logical "1" Input Current at $\overline{\text{Output Enable}}$	V_{CC} = 15V, V_{IN} = 15V		0.005	1.0	μA
$I_{IN(0)}$	Logical "0" Input Current at $\overline{\text{Output Enable}}$	V_{CC} = 15V, V_{IN} = 0V	−1.0	−0.005		μA
CMOS/LPTTL INTERFACE						
$V_{IN(1)}$	Except Osc and KBM Inputs	V_{CC} = 4.75V	V_{CC} − 1.5			V
$V_{IN(0)}$	Except Osc and KBM Inputs	V_{CC} = 4.75V			0.8	V
$V_{OUT(1)}$	Logical "1" Output Voltage	I_O = −360 μA, V_{CC} = 4.75V, I_O = −360 μA	2.4			V
$V_{OUT(0)}$	Logical "0" Output Voltage	I_O = −360 μA, V_{CC} = 4.75V, I_O = −360 μA			0.4	V

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

DC Electrical Characteristics (Continued)

Symbol	Parameter	Conditions	Min	Typ	Max	Units
OUTPUT DRIVE (See Family Characteristics Data Sheet) (Short Circuit Current)						
I_{SOURCE}	Output Source Current (P-Channel)	V_{CC} = 5V, V_{OUT} = 0V, T_A = 25°C	–1.75	–3.3		mA
I_{SOURCE}	Output Source Current (P-Channel)	V_{CC} = 10V, V_{OUT} = 0V, T_A = 25°C	–8	–15		mA
I_{SINK}	Output Sink Current (N-Channel)	V_{CC} = 5V, V_{OUT} = V_{CC}, T_A = 25°C	1.75	3.6		mA
I_{SINK}	Output Sink Current (N-Channel)	V_{CC} = 10V, V_{OUT} = V_{CC}, T_A = 25°C	8	16		mA

AC Electrical Characteristics (Note 3)

T_A = 25°C, C_L = 50 pF, unless otherwise noted

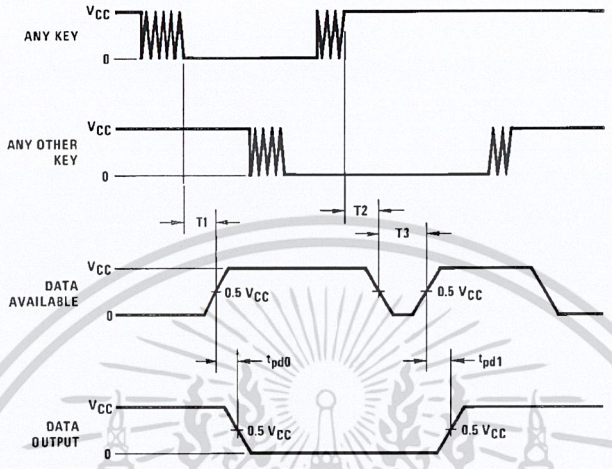
Symbol	Parameter	Conditions	Min	Typ	Max	Units
t_{pd0}, t_{pd1}	Propagation Delay Time to Logical "0" or Logical "1" from D.A.	C_L = 50 pF (Figure 1)				
		V_{CC} = 5V		60	150	ns
		V_{CC} = 10V		35	80	ns
		V_{CC} = 15V		25	60	ns
t_{0H}, t_{1H}	Propagation Delay Time from Logical "0" or Logical "1" into High Impedance State	R_L = 10k, C_L = 10 pF (Figure 2)				
		V_{CC} = 5V, R_L = 10k		80	200	ns
		V_{CC} = 10V, C_L = 10 pF		65	150	ns
		V_{CC} = 15V		50	110	ns
t_{H0}, t_{H1}	Propagation Delay Time from High Impedance State to a Logical "0" or Logical "1"	R_L = 10k, C_L = 50 pF (Figure 2)				
		V_{CC} = 5V, R_L = 10k		100	250	ns
		V_{CC} = 10V, C_L = 50 pF		55	125	ns
		V_{CC} = 15V		40	90	ns
C_{IN}	Input Capacitance	Any Input (Note 4)		5	7.5	pF
C_{OUT}	3-STATE Output Capacitance	Any Output (Note 4)		10		pF

Note 3: AC Parameters are guaranteed by DC correlated testing.

Note 4: Capacitance is guaranteed by periodic testing.

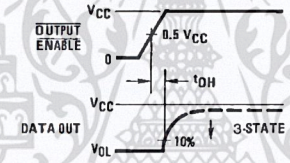
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Switching Time Waveforms



T1 = T2 = RC, T3 = 0.7 RC, where R = 10k and C is external capacitor at KBM input.

FIGURE 1.



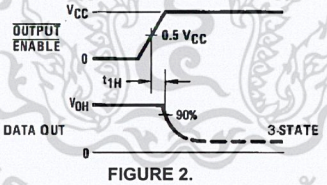
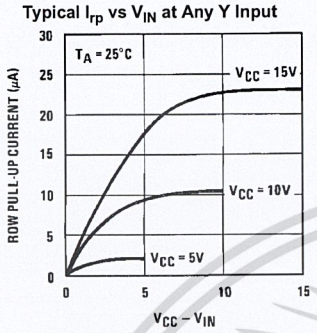


FIGURE 2.

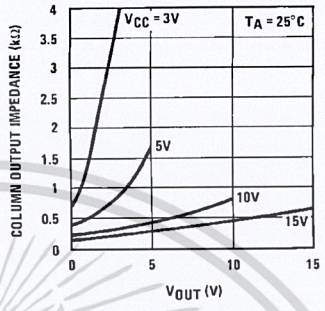
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Typical Performance Characteristics

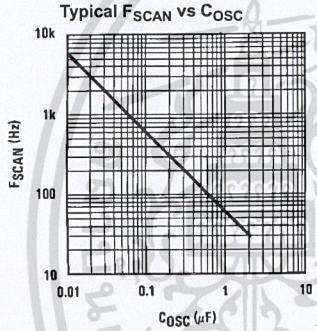
Typical I_{rp} vs V_{IN} at Any Y Input



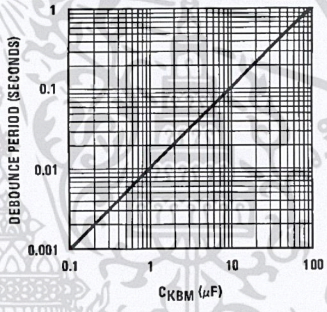
Typical R_{on} vs V_{OUT} at Any X Output



Typical F_{SCAN} vs C_{OSC}

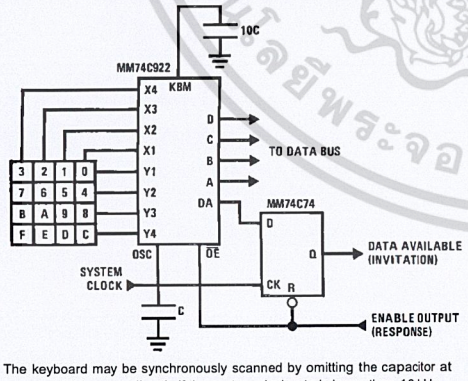


Typical Debounce Period vs C_{KBM}



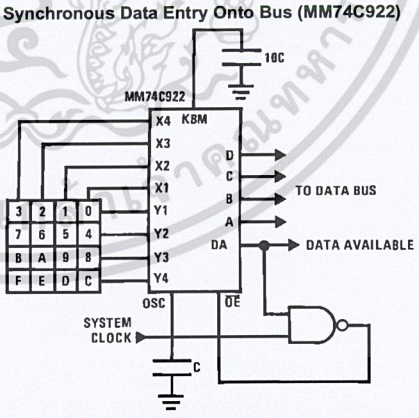
Typical Applications

Synchronous Handshake (MM74C922)



The keyboard may be synchronously scanned by omitting the capacitor at osc. and driving osc. directly if the system clock rate is lower than 10 kHz

Synchronous Data Entry Onto Bus (MM74C922)

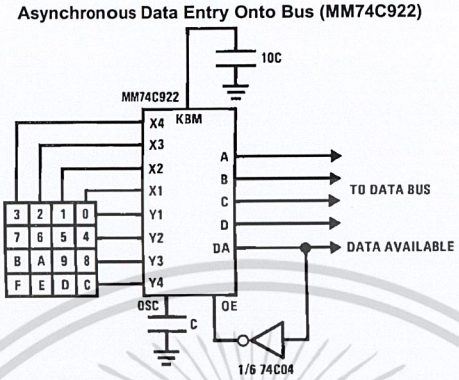


Outputs are enabled when valid entry is made and go into 3-STATE when key is released.

The keyboard may be synchronously scanned by omitting the capacitor at osc. and driving osc. directly if the system clock rate is lower than 10 kHz

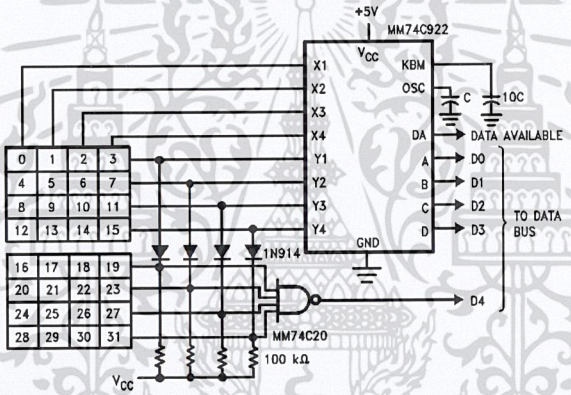
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Asynchronous Data Entry Onto Bus (MM74C922)



Outputs are in 3-STATE until key is pressed, then data is placed on bus. When key is released, outputs return to 3-STATE.

Expansion to 32 Key Encoder (MM74C922)



Theory of Operation

The MM74C922/MM74C923 Keyboard Encoders implement all the logic necessary to interface a 16 or 20 SPST key switch matrix to a digital system. The encoder will convert a key switch closer to a 4(MM74C922) or 5(MM74C923) bit nibble. The designer can control both the keyboard scan rate and the key debounce period by altering the oscillator capacitor, C_{OSE}, and the key bounce mask capacitor, C_{MSK}. Thus, the MM74C922/MM74C923's performance can be optimized for many keyboards.

The keyboard encoders connect to a switch matrix that is 4 rows by 4 columns (MM74C922) or 5 rows by 4 columns (MM74C923). When no keys are depressed, the row inputs are pulled high by internal pull-ups and the column outputs sequentially output a logic "0". These outputs are open drain and are therefore low for 25% of the time and otherwise off. The column scan rate is controlled by the oscillator input, which consists of a Schmitt trigger oscillator, a 2-bit counter, and a 2–4-bit decoder.

When a key is depressed, key 0, for example, nothing will happen when the X1 input is off, since Y1 will remain high. When the X1 column is scanned, X1 goes low and Y1 will go low. This disables the counter and keeps X1 low. Y1 going low also initiates the key bounce circuit timing and locks out the other Y inputs. The key code to be output is a combination of the frozen counter value and the decoded Y inputs. Once the key bounce circuit times out, the data is latched, and the Data Available (DAV) output goes high.

If, during the key closure the switch bounces, Y1 input will go high again, restarting the scan and resetting the key bounce circuitry. The key may bounce several times, but as soon as the switch stays low for a debounce period, the closure is assumed valid and the data is latched.

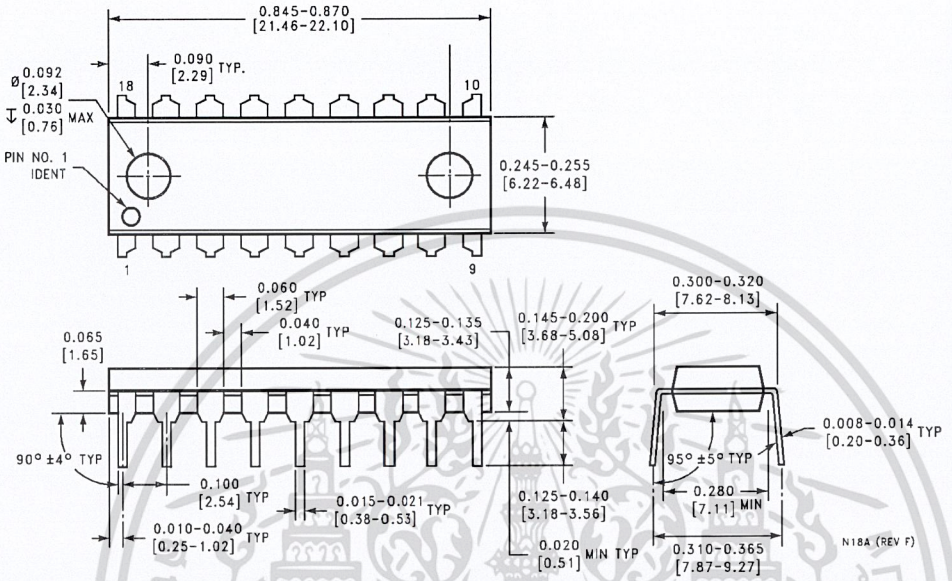
A key may also bounce when it is released. To ensure that the encoder does not recognize this bounce as another key closure, the debounce circuit must time out before another closure is recognized.

The two-key roll-over feature can be illustrated by assuming a key is depressed, and then a second key is depressed. Since all scanning has stopped, and all other Y inputs are disabled, the second key is not recognized until the first key is lifted and the key bounce circuitry has reset.

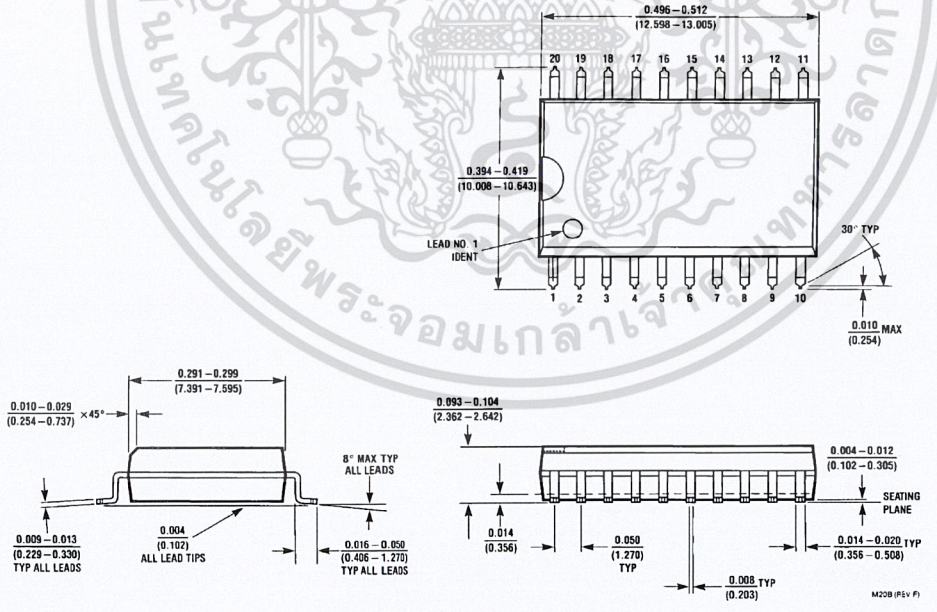
The output latches feed 3-STATE, which is enabled when the Output Enable ($\overline{OE}$) input is taken low.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Physical Dimensions inches (millimeters) unless otherwise noted



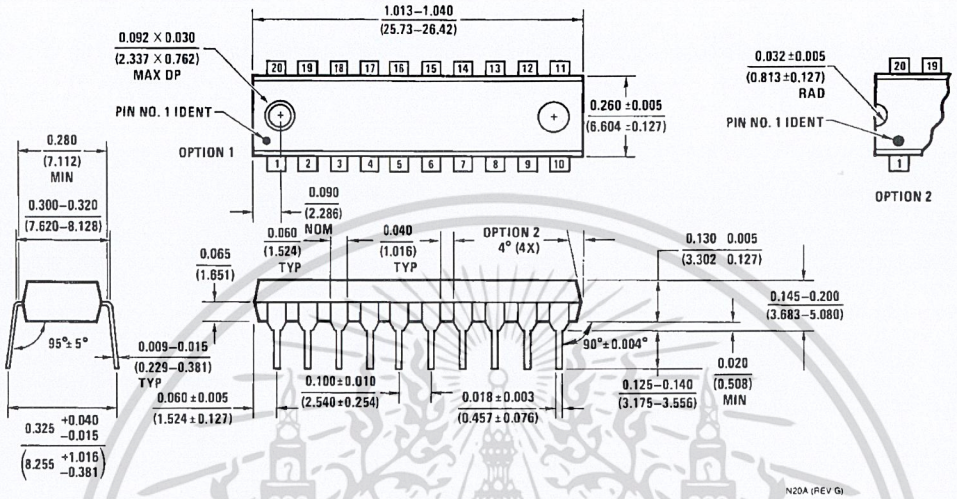
18-Lead Plastic Dual-In-Line Package (PDIP), JEDEC MS-001, 0.300" Wide
Package Number N18A



20-Lead Plastic Small Outline I.C. Package (M)
Package Number M20B

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Physical Dimensions inches (millimeters) unless otherwise noted (Continued)



20-Lead Plastic Dual-In-Line Package (PDIP), JEDEC MS-001, 0.300" Wide
Package Number N20A

LIFE SUPPORT POLICY

FAIRCHILD'S PRODUCTS ARE NOT AUTHORIZED FOR USE AS CRITICAL COMPONENTS IN LIFE SUPPORT DEVICES OR SYSTEMS WITHOUT THE EXPRESS WRITTEN APPROVAL OF THE PRESIDENT OF FAIRCHILD SEMICONDUCTOR CORPORATION. As used herein:

1. Life support devices or systems are devices or systems which, (a) are intended for surgical implant into the body, or (b) support or sustain life, and (c) whose failure to perform when properly used in accordance with instructions for use provided in the labeling, can be reasonably expected to result in a significant injury to the user.
2. A critical component in any component of a life support device or system whose failure to perform can be reasonably expected to cause the failure of the life support device or system, or to affect its safety or effectiveness.

www.fairchildsemi.com

Fairchild does not assume any responsibility for use of any circuitry described, no circuit patent licenses are implied and Fairchild reserves the right at any time without notice to change said circuitry and specifications.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

FEATURES of

LIQUID CRYSTAL DISPLAY MODULE

MODEL NO. : MTC-S16100XRGHS

Character Type Dot Matrix LCD Module
Easy interface of 4-bit or 8-bit
Display character pattern : 5 × 7 font with cursor(208 kinds)
The special character pattern can be programmable by Character Generator RAM directly.
A customer character pattern can be programmable by mask option.
Automatically power on reset.
Internal Memory:
- Character Generator ROM (CGROM) : 10,080 bits (204 characters × 5 × 8 dot)
- Character Generator RAM (CGRAM) : 64 × 8 bits (8 characters × 5 × 8 dot)

Low power operation:
- Power supply voltage range : 2.7 ~ 5.5 V (V_{DD})
- LCD drive voltage range : 3.0 ~ 13.0 V (V_{DD} - Vo)

Low power consumption
CMOS process
Duty-cycle : 1/16

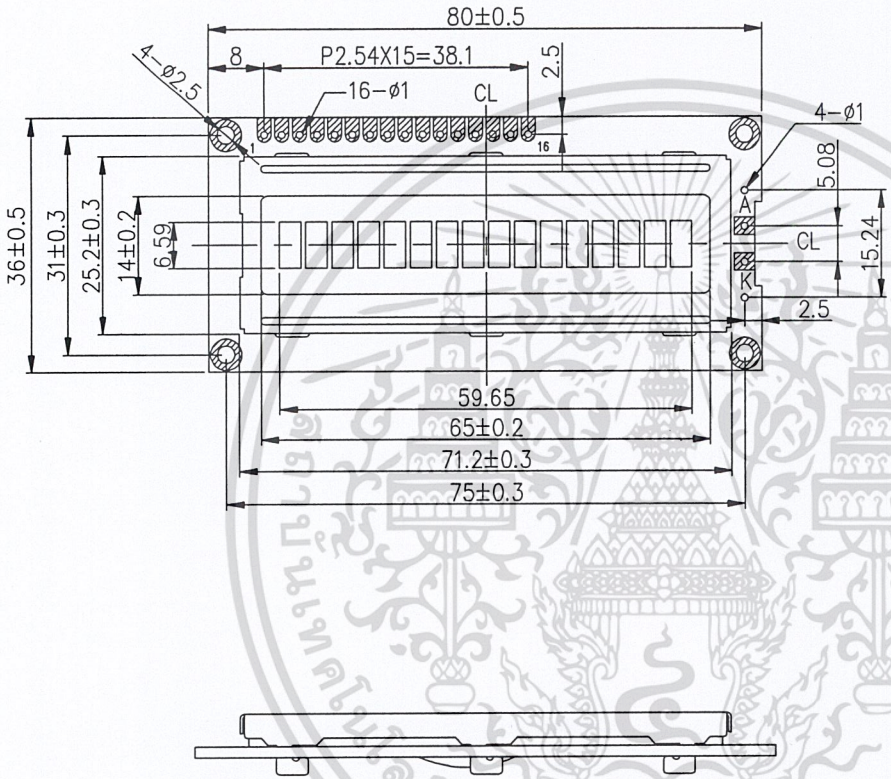
View Direction	☑ 6 O'clock		☐ 12 O'clock	
LCD Type	☐ FSTN Positive		☐ FSTN Negative	
	☑ STN Gray	☐ STN Yellow Green	☐ STN Blue	
Rear Polarizer	☑ Reflective	☐ Transflective	☐ Transmissive	
Backlight Type	☐ LED	☐ Internal Power	☐ EL	☐ 5V input
		☐ External Power	☐ CCFL	☐ 12V input
Backlight Color	☐ White	☐ Amber	☐ Blue Green	☐ Yellow Green / ☐ Other
Temperature Range	☐ Normal	☑ Wide	☐ Super Wide	
CCFL Inverter	☐ Build-in	☑ Not Build-in		
EL Driver IC	☐ Build-in	☑ Not Build-in		
Touch Screen	☐ With	☑ Without		

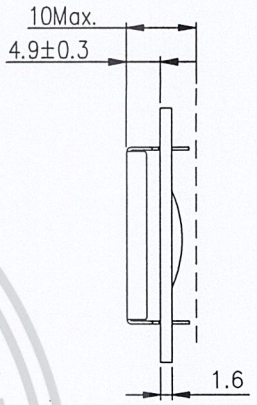
TO BE VERY CAREFUL !

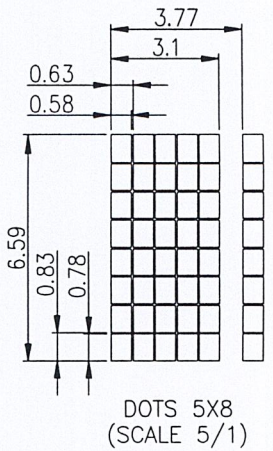
The LCD driver ICs are made of CMOS process, which is very easy to be damaged by static charge, make sure the user is grounded when handling the LCM.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

REV.	DESCRIPTION	DRAWN	CHECK	APP'D	DATE







DOTS 5X8
(SCALE 5/1)

PIN ASSIGNMENT							
1.	Vss	5.	R/W	9.	DB2	13.	DB6
2.	Vdd	6.	E	10.	DB3	14.	DB7
3.	Vee	7.	DB0	11.	DB4	15.	LEDA
4.	RS	8.	DB1	12.	DB5	16.	LEDK

NO.	PART NAME	Q'TY	DATE	MATERIAL	FINISH
	M-16100X		07/22/99'		

DRAWN	CHECK	APP'D	UNIT: MM	SCALE: 1/1	PRODUCT: MTC-16100X
Henry	Chris	Suny		UNLESS OTHERWISE SPECIFIED TOLERANCE ON FRACTIONS DECIMALS ANGLES 0.10 0.5°	DWG NO.: C16100X0
					FILE: DWG\MTC16\C16100X
					SHEET 1 / 1 Rev: A

Microtips Technology

GENERAL SPECIFICATION

Item	Content
Display Resolution	16 Charactersx 1 line
Dimensional Outline(mm)	80.0(W)x 36.0(H)x 10.0 max(D)
Display mode	Reflective/ Positive Type
Circuit	Controller IC, Segment-driver IC
Interface	Data (DB0～DB7), RS, R/W, E

ABSOLUTE MAXIMUN RATING

(1) Electrical Absolute Ratings

Item	Symbol	Min.	Max.	Unit	Note
Power Supply for Logic	V_{DD}-V_{SS}	-0.3	7.0	Volt	
Power Supply for LCD	V_{DD}-Vo	0	15.0	Volt	
Input Voltage	V_{IN}	-0.3	V_{DD}+0.3	Volt	

Note: Operator should be grounded during handling LCM.

(2) Environmental Absolute Maximum Ratings

Item	Normal Temperature				Wide Temperature			
	Operating		Storage		Operating		Storage	
	Max.	Min.	Max.	Min.	Max,	Min.	Max,	Min.
Ambient Temperature	0℃	+50℃	-20℃	+70℃	-20℃	+60℃	-30℃	+70℃
Humidity(without condensation)	Note 2,4		Note 3,5		Note 4,5		Note 4,6	

Note 2 Ta≦50℃: 80% RH max
Ta>50℃: Absolute humidity must be lower than the humidity of 85%RH at 50℃

Note 3 Ta at -20℃ will be<48hrs at 70℃will be <120hrs when humidity is higher than 70%.

Note 4 Background color changes slightly depending on ambient temperature. This phenomenon is reversible.

Note 5 Ta≦70℃: 75RH max
Ta>70℃: absolute humidity must be lower than the humidity of 75%RH at 70℃

Note 6 Ta at -30℃ will be <48hrs, at 80 ℃ will be <120hrs when humidity is higher than 70%.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ELECTRICAL CHARACTERISTICS

Item	Symbol	Condition	Min.	Typ.	Max.	Unit	note
Power Supply for Logic	V_{DD}-V_{SS}	-	2.7	4.5	5.5	Volt	
Input Voltage	V_{IL}	L level	V_{SS}	0.2 V_{DD}	-	Volt	
	V_{IH}	H level	0.8 V_{DD}	V_{DD}	-	Volt	
LCM Recommend LCD Module Driving Voltage	V_{DD}=5.0V V_{DD}-Vo= 3.8V	Ta=-20℃	-	-	-	Volt	
		Ta=0℃	-	-	-		
		Ta=25℃	3.3	3.8	4.3		
		Ta=50℃	-	-	-		
Power Supply Current for LCM	I_{DD} (B/L OFF)	V_{DD}=5.0V V_{DD}-Vo= 3.8V Ta=25℃	-	1.5	3.0	mA	
	I_{LED}	V_{LED}=4.2V	-	110	180		

OPTICAL CHARACTERISTICS

Item	Symbol	Condition	Min.	Typ	Max.	Unit	note
Viewing angle range	Φf(12 o'clock)	When Cr≧2	-	40	-	Degree	9,10
	Φb(6 o'clock)		-	50	-		
	Φl(9 o'clock)		-	40	-		
	Φr(3 o'clock)		-	40	-		
Rise Time	T_r	V_{DD}-Vo =3.8V Ta=25℃		67	80	mS	
Fall Time	T_f			150	200		
Contrast	Cr		-	6.32	-		7

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

MECHANICAL SPECIFICATION

Product No.		MTC-S16100XRGHS
Module Size		80.0(W)mm× 36.0(H)mm× 10.0(D)mm Max
Display Format		16 characters× 1 lines
Character Size		3.1 mm× 6.59 mm
Character Pitch		3.77 mm
Dot Size		0.58(W)mm× 0.78(H)mm
Dot Pitch		0.63(W)mm× 0.83(H)mm
Duty Ratio		1/16 Duty
Character Font Table		☑English ☐European ☐Other_____
LCD Display Mode	STN	☑Gray Mode ☐Yellow Mode ☐Blue Mode
	FSTN	☐ Black & White(Normally White/Positive Image) ☐ Black & White(Normally White/Negative Image)
	Rear Polarizer:	☑ Reflective ☐ Transflective ☐ Transmissive
Viewing Direction		☑6 O'clock ☐12 O'clock ☐3 O'clock ☐9 O'clock
Backlight		☑Without ☐CCFL ☐EL ☐LED
Controller		KS0066 or compatible
DC/DC Converter		Without
CCFL Inverter		Without
EL Driver		Without

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

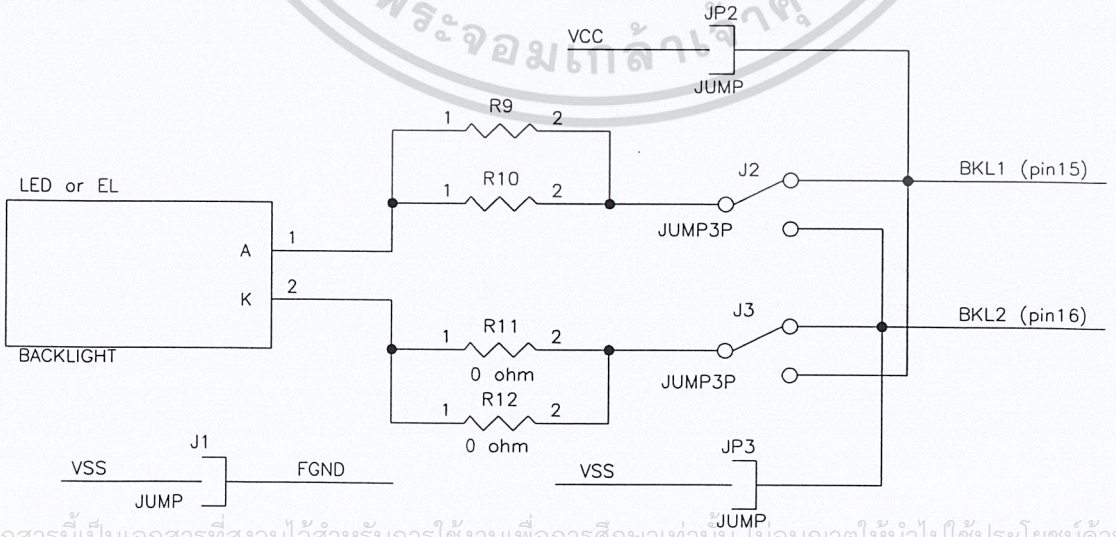
INTERFACE PIN ASSIGNMENT

PIN NO.	PIN OUT	DESCRIPTION
1	V_{SS}	Ground
2	V_{DD}	Logic Circuit Power Supply
3	Vo	Power Supply For LCD Panel
4	RS	Data/ Instruction Register Select
5	R/W	Read/ Write Select
6	E	Enable Signal
7	DB0	3-State I/O Data Bus
|	|	
14	DB7	
15	BKL1	Power Supply for Backlight. See JUMPER EXPLANATION below. 100V/400Hz AC for EL, 4.2V or 120~180mA DC for LED backlight Don't care if no backlight
16	BKL2	

JUMPER EXPLANATION

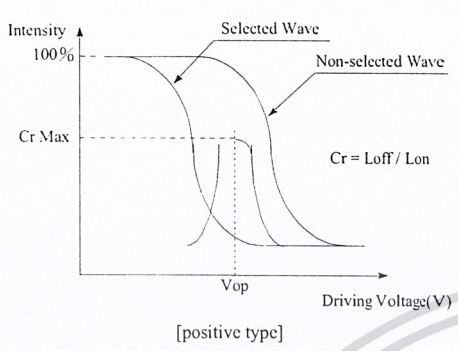
1. JP2 and JP3 are both short: Pin15 is short with pin2 and pin16 is short with pin1. Backlight powered up via V_{DD} (pin2) and V_{SS}(pin1). Customer does not need to supply power to pin15 and pin16.
2. JP2 short and JP3 open: Pin 15 is short with Pin2 and pin16 is NOT short with pin1. Backlight is powered with V_{DD} (same as logic circuit) and customer should keep pin 15 floating or connected to V_{DD}. Customer could control the backlight independently ON or OFF with pin16 L or H.
3. JP2 open and JP3 short: Pin 15 is NOT short with pin 2 and pin16 is short with pin1. Backlight is common VSS with logic and customer should keep pin16 floating or connected to GND (V_{SS}). Customer could control the backlight ON or OFF with pin15 H or L
4. JP2 and JP3 are both open: The backlight is fully independent with the logic, control the backlight via pin15/pin16 or A/K.
5. J1 short: Bezel and screw holes connected to GND. J1 open: Bezel and screw holes floating
6. Never change the J2 and J3, it may burn off your system.

Note: J1, JP2 and JP3 are #0805 0 ohm resistors on the rear side of the PCB.

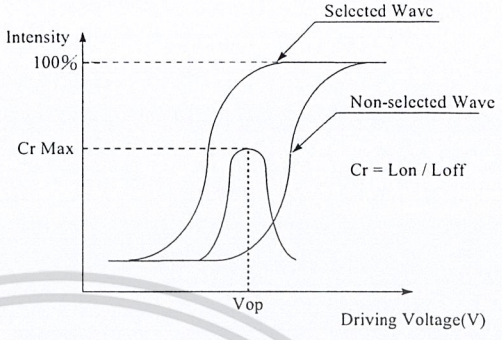


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

[Note 7] Definition of Operation Voltage (Vop)

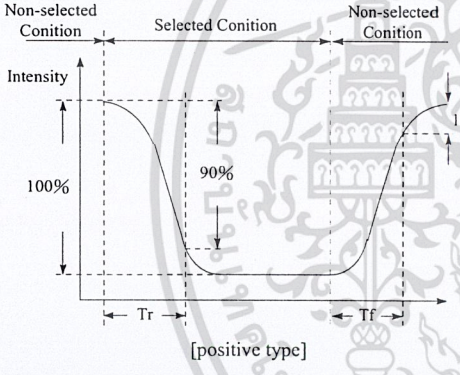


[positive type]

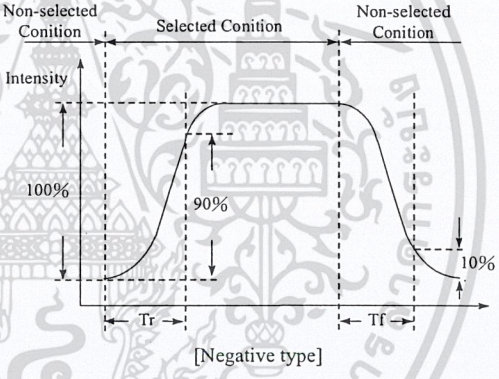


[Negative type]

[Note 8] Definition of Response Time (Tr, Tf)



[positive type]



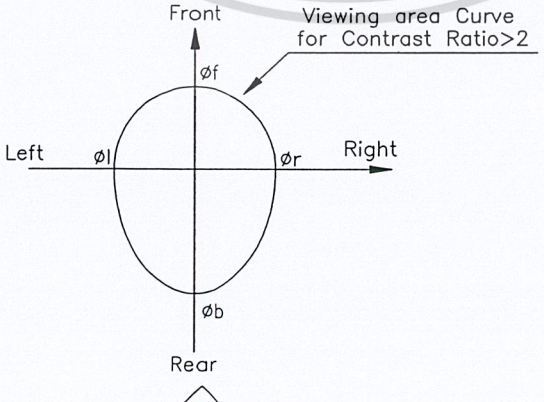
[Negative type]

Conditions:

Operating Voltage : Vop **Viewing Angle(θ, φ): 0° , 0°**

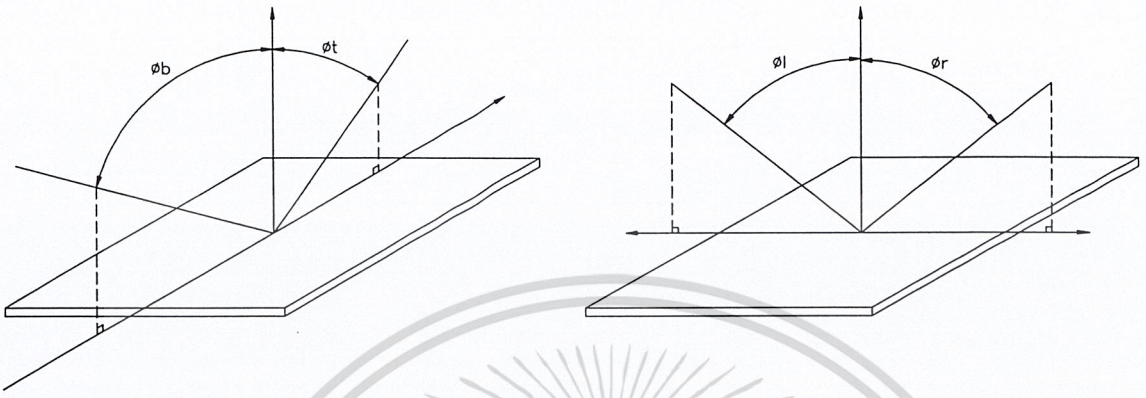
Frame Frequency : 64 Hz **Driving Wave form : 1/N duty, 1/a bias**

[Note 9] Definition of Viewing Direction

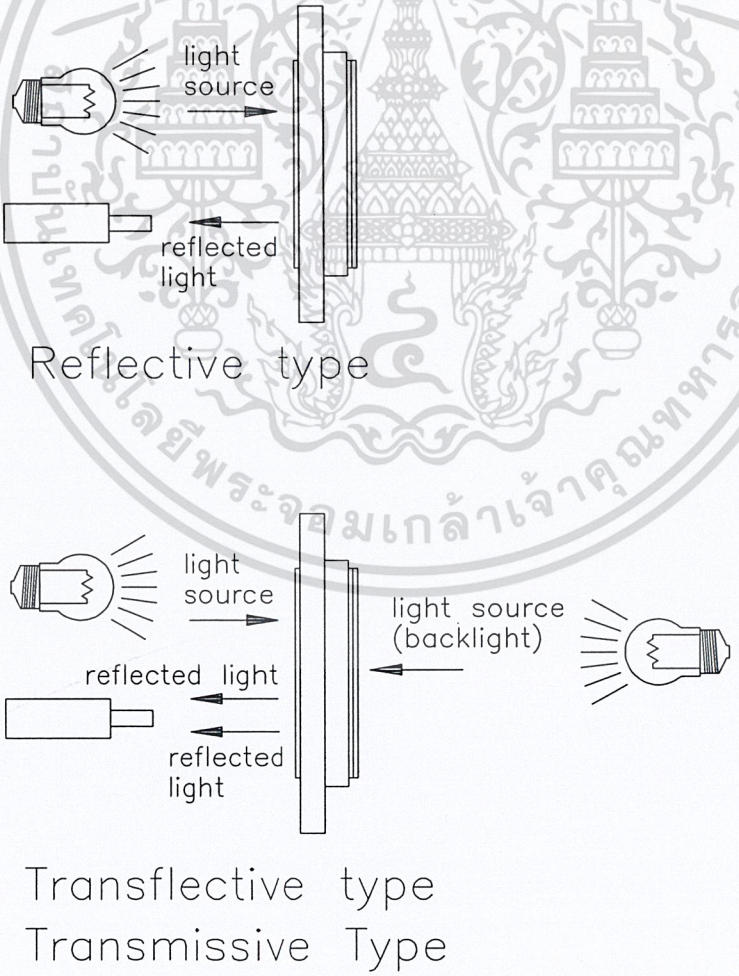


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

[Note 10] Definition of viewing angle

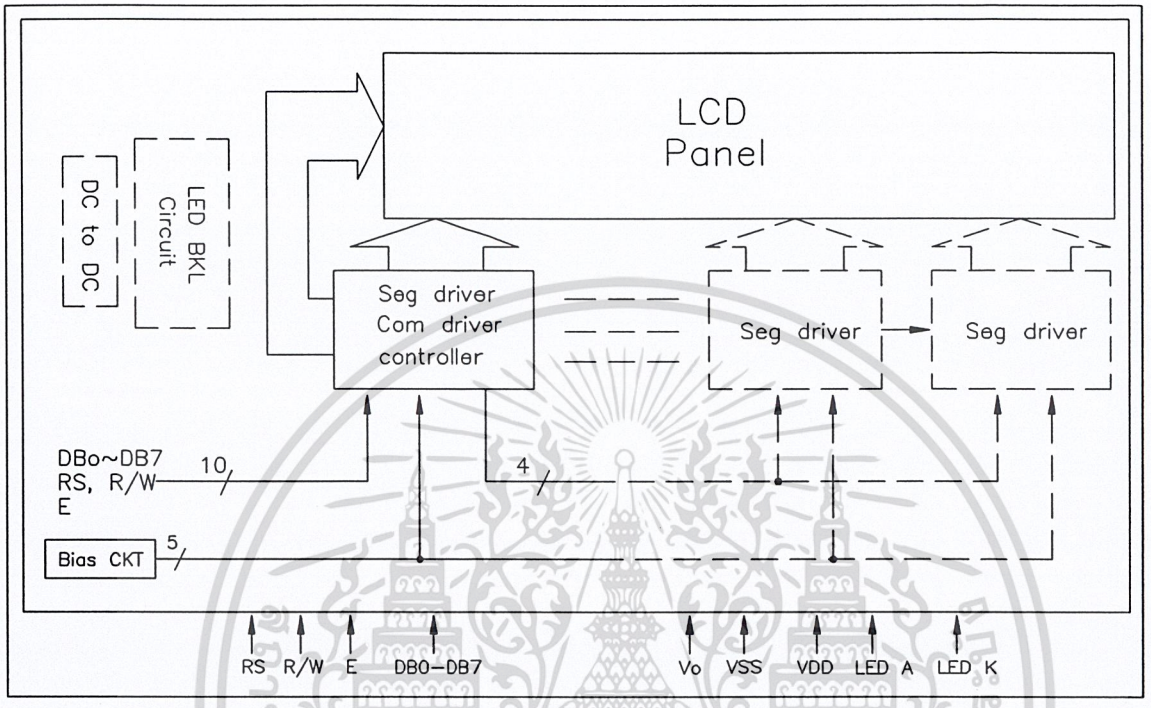


[Note 11] Description of Measuring Equipment

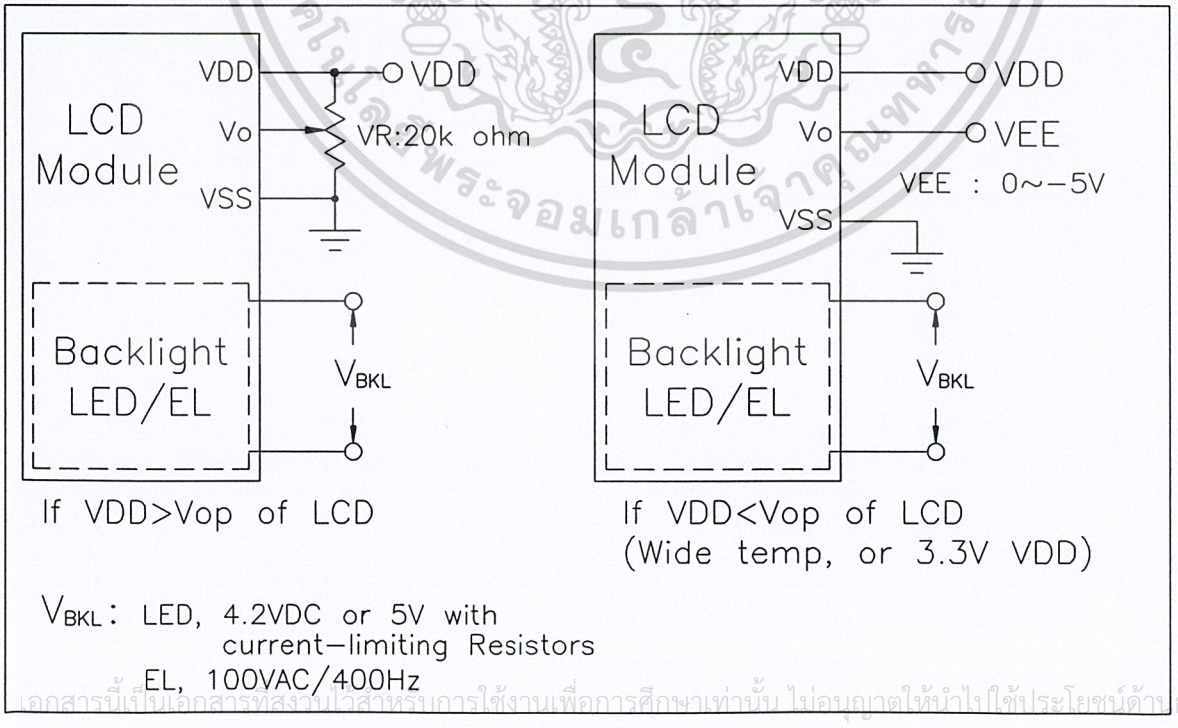


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

BLOCK DIAGRAM



POWER SUPPLY



V_{BKL}: LED, 4.2VDC or 5V with current-limiting Resistors
EL, 100VAC/400Hz

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

TIMING CHARACTERISTICS

AC Characteristics (V_{SS}= 0V, V_{DD}=4.5V to 5.0V, Ta=0 to 50℃)

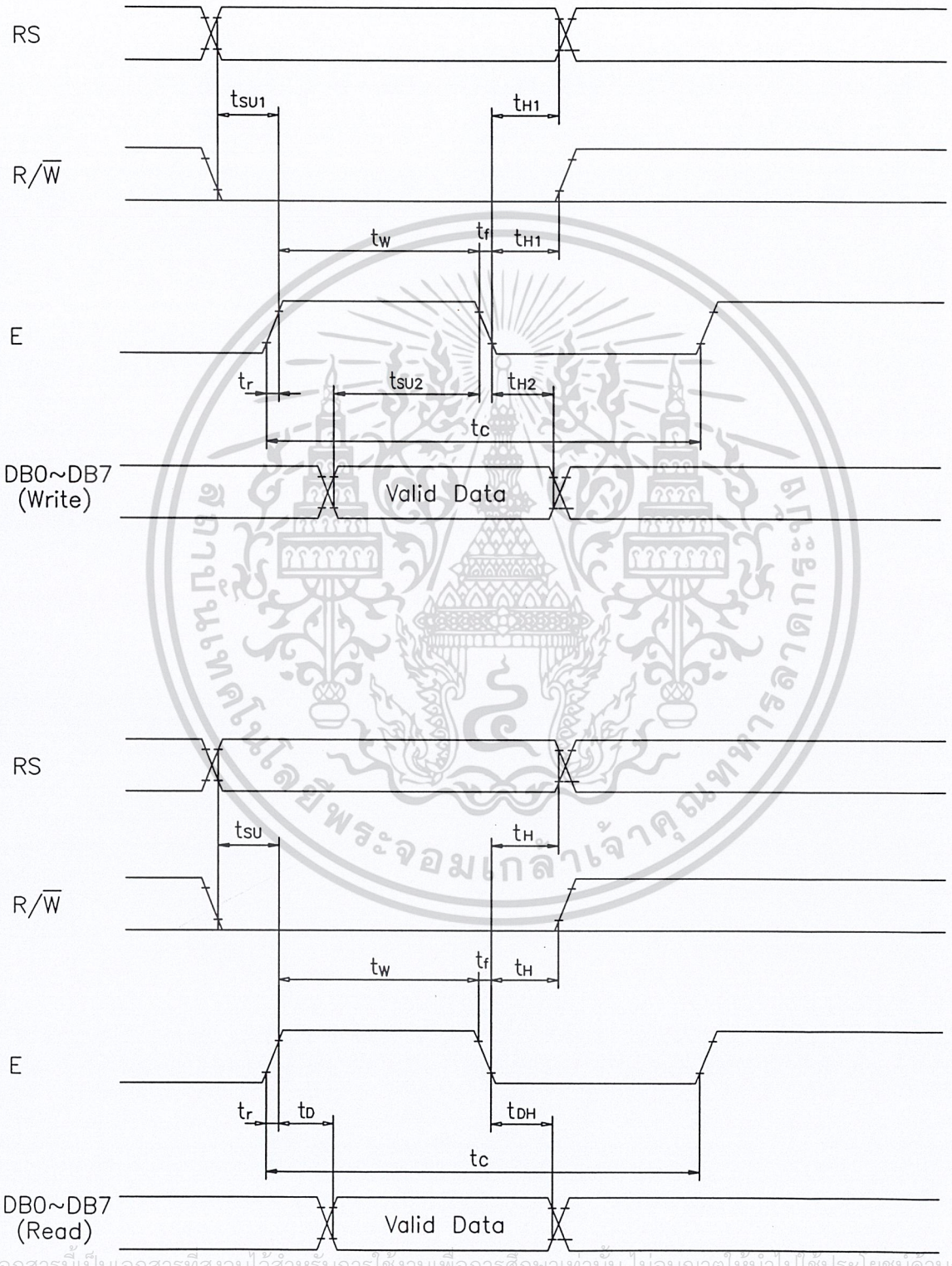
Mode	Characteristic	Symbol	Min.	Typ.	Max.	Unit
Write Mode	E Cycle Time	t_C	500	-	-	ns
	E Rise/Fall Time	t_R, t_F	-	-	20	ns
	E Pulse Width(High, Low)	t_W	230	-	-	ns
	R/W And RS Setup Time	t_{SU1}	40	-	-	ns
	R/W And RS Hold Time	t_{H1}	10	-	-	ns
	Data Setup Time	t_{SU2}	80	-	-	ns
	Data Hold Time	t_{H2}	10	-	-	ns
Read Mode	E Cycle Time	t_C	500	-	-	ns
	E Rise/Fall Time	t_R, t_F	-	-	20	ns
	E Pulse Width(High, Low)	t_W	230	-	-	ns
	R/W And RS Setup Time	t_{SU}	40	-	-	ns
	R/W And RS Hold Time	t_H	10	-	-	ns
	Data Setup Time	t_D	-	-	120	ns
	Data Hold Time	t_{DH}	5	-	-	ns

AC Characteristics (V_{SS}= 0V, V_{DD}=2.7V to 4.5V, Ta=0 to 50℃)

Mode	Characteristic	Symbol	Min.	Typ.	Max.	Unit
Write Mode	E Cycle Time	t_C	1000	-	-	ns
	E Rise/Fall Time	t_R, t_F	-	-	25	ns
	E Pulse Width(High, Low)	t_W	450	-	-	ns
	R/W And RS Setup Time	t_{SU1}	60	-	-	ns
	R/W And RS Hold Time	t_{H1}	20	-	-	ns
	Data Setup Time	t_{SU2}	195	-	-	ns
	Data Hold Time	t_{H2}	10	-	-	ns
Read Mode	E Cycle Time	t_C	1000	-	-	ns
	E Rise/Fall Time	t_R, t_F	-	-	25	ns
	E Pulse Width(High, Low)	t_W	450	-	-	ns
	R/W And RS Setup Time	t_{SU}	60	-	-	ns
	R/W And RS Hold Time	t_H	20	-	-	ns
	Data Setup Time	t_D	-	-	360	ns
	Data Hold Time	t_{DH}	5	-	-	ns

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Read/Write Timing Chart



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Commands

Instruction	Instruction Code										Description	Execution time(f_{osc} is 270kHz)
	RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0		
Clear Display	0	0	0	0	0	0	0	0	0	1	Write "20H" to DDRAM. and set DDRAM address to "00H" from AC	1.53mS
Return Home	0	0	0	0	0	0	0	0	1	*	Set DDRAM address to "00H" from AC and return cursor to its original position if shifted. The contents of DDRAM are not changed.	1.53mS
Entry Mode	0	0	0	0	0	0	0	1	I/D	SH	Assign cursor moving direction and make shift of entire display enable.	39 µS
Display ON/OFF	0	0	0	0	0	0	1	D	C	B	Set display(D), cursor(C), and blinking of cursor(B) on/off control bit.	39 µS
Cursor or Display Shift	0	0	0	0	0	1	S/C	R/L	*	*	Set cursor moving and display shift control bit, and the direction, without changing DDRAM data.	39 µS
Function Set	0	0	0	0	1	DL	N	F	*	*	Set interface data length (DL : 4-bit/8-bit), numbers of display line (N : 1-line/2-line), display font type(F : 5 X 8 dots/ 5 X 11 dots)	39 µS
Set CG RAM Address	0	0	0	1	AC5	AC4	AC3	AC2	AC1	AC0	Set CGRAM address in address counter.	39 µS
Set DD RAM Address	0	0	1	AC6	AC5	AC4	AC3	AC2	AC1	AC0	Set DDRAM address in address counter.	39 µS
Read Busy Flag and Address	0	1	BF	AC6	AC5	AC4	AC3	AC2	AC1	AC0	Whether during internal operation or not can be known by reading BF. The contents of address counter can also be read.	0 µS
Write Data to RAM	1	0	D7	D6	D5	D4	D3	D2	D1	D0	Write data into internal RAM (DDRAM/CGRAM).	43 µS
Read Data from RAM	1	1	D7	D6	D5	D4	D3	D2	D1	D0	Read data from internal RAM (DDRAM/CGRAM).	43 µS

* means don't care

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

COMMANDS DESCRIPTION

Clear Display

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

Clear all the display data by writing "20H" (space code) to all DDRAM address, and set DDRAM address to "00H" into AC (address counter). Return cursor to the original status. namely, bring the cursor to the left edge on first line of the display. Make entry mode increment (I/D = "1").

Return Home

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

Return Home is cursor return home instruction. Set DDRAM address to "00H" into the address counter. Return cursor to its original site and return display to its original status, if shifted. Content of DDRAM is not changed.

Entry Mode Set

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

Set the moving direction of cursor and display.
I/D : Increment / decrement of DDRAM address (cursor or blink)
When I/D = "High", cursor/blink moves to right and DDRAM address is increased by 1.
When I/D = "Low", cursor/blink moves to left and DDRAM address is decreased by 1.
* CGRAM operates the same as DDRAM, when read from or write to CGRAM.
SH: Shift of entire display
When DDRAM read (CGRAM read/write) operation or SH = "Low", shift of entire display is not performed. If SH = "High" and DDRAM write operation, shift of entire display is performed according to I/D value (I/D = "1" , shift left, I/D = "0" : shift right).

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Display ON/OFF Control

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

Control display/cursor/blink ON/OFF 1 bit register.
D : Display ON/OFF control bit
When D = "High", entire display is turned on.
When D = "Low", display is turned off, but display data is remained in DDRAM.
C : Cursor ON/OFF control bit
When C = "High", cursor is turned on.
When C = "Low", cursor is disappeared in current display, but I/D register remains its data.
B : Cursor Blink ON/OFF control bit
When B = "High", cursor blink is on, that performs alternate between all the high data and display character at the cursor position.
When B = "Low", blink is off.

Cursor or Display Shift

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

Without writing or reading of display data, shift right/left cursor position or display.
This instruction is used to correct or search display data. (Refer to Table 4)
During 2-line mode display, cursor moves to the 2nd line after 40th digit of 1st line.
Note that display shift is performed simultaneously in all the line.
When displayed data is shifted repeatedly, each line shifted individually.
When display shift is performed, the contents of address counter are not changed.

S/C	R/L	Operation
0	0	Shift cursor to the left, AC is decreased by 1.
0	1	Shift cursor to the right, AC is increased by 1.
1	0	Shift all of the display to the left, cursor moves according to the display.
1	1	Shift all of the display to the right, cursor moves according to the display.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Function Set

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
00	0	0	0	1	DL	N	F	*	*

DL : Interface data length control bit
When DL = "High", it means 8-bit bus mode with MPU.
When DL = "Low", it means 4-bit bus mode with MPU. So to speak, DL is a signal to select 8-bit or 4-bit bus mode. When 4-bit bus mode, it needs to transfer 4-bit data by two times.
N : Display line number control bit
When N = "Low", it means 1-line display mode.
When N = "High", 2-line display mode is set.
F : Display font type control bit
When F = "Low", it means 5 X 8 dots format display mode
When F = "High", 5 x11 dots format display mode.

Set CG RAM Address

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	1	AC5	AC4	AC3	AC2	AC1	AC0

Set CGRAM address to AC.
This instruction makes CGRAM data available from MPU.

Set DD RAM Address

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	1	AC6	AC5	AC4	AC3	AC2	AC1	AC0

Set DDRAM address to AC.
This instruction makes DDRAM data available from MPU.
When 1-line display mode (N = 0), DDRAM address is from "00H" to "4FH".
In 2-line display mode (N = 1), DDRAM address in the 1st line is from "00H" to "27H", and DDRAM address in the 2nd line is from "40H" to "67H".

Read Busy Flag and Address

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	1	BF	AC6	AC5	AC4	AC3	AC2	AC1	AC0

This instruction shows whether KS0066U is in internal operation or not. If the resultant BF is High, it means the internal operation is in progress and you have to wait until BF to be Low, and then the next instruction can be performed. In this instruction you can read also the value of address counter.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Write Data to RAM

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
1	0	D7	D6	D5	D4	D3	D2	D1	D0

Write binary 8-bit data to DDRAM/CGRAM.
The selection of RAM from DDRAM, CGRAM, is set by the previous address set instruction : DDRAM address set, CGRAM address set. RAM set instruction can also determine the AC direction to RAM. After write operation, the address is automatically increased/decreased by 1, according to the entry mode.

Read Data to RAM

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
1	0	D7	D6	D5	D4	D3	D2	D1	D0

Read binary 8-bit data from DDRAM/CGRAM.
The selection of RAM is set by the previous address set instruction. If address set instruction of RAM is not performed before this instruction, the data that read first is invalid, because the direction of AC is not determined. If you read RAM data several times without RAM address set instruction before read operation, you can get correct RAM data from the second, but the first data would be incorrect, because there is no time margin to transfer RAM data.
In case of DDRAM read operation, cursor shift instruction plays the same role as DDRAM address set instruction : it also transfers RAM data to output data register. After read operation address counter is automatically increased/decreased by 1 according to the entry mode. After CGRAM read operation, display shift may not be executed correctly.

NOTE: In case of RAM write operation, after this AC is increased/decreased by 1 like read operation. In this time, AC indicates the next address position, but you can read only the previous data by read instruction.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

DD RAM ADDRESSING

For 16×1 or 8×1 Display

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

For 16×2 or 8×2 Display

Character	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
DD RAM Address	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

For 16×4 Display

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

For 20×2 Display

Character	1	2	3	4	5	6	7	8	9	10	…	…	17	18	19	20
DD RAM Address	00	01	02	03	04	05	06	07	08	09	…	…	10	11	12	13
	40	41	42	43	44	45	46	47	48	49	…	…	50	51	52	53

For 20×4 Display

Character	1	2	3	4	5	6	7	8	9	10	…	…	17	18	19	20
	00	01	02	03	04	05	06	07	08	09	…	…	10	11	12	13
DD RAM Address	40	41	42	43	44	45	46	47	48	49	…	…	50	51	52	53
	14	15	16	17	18	19	1A	1B	1C	1D	…	…	24	25	26	27
	54	55	56	57	58	59	5A	5B	5C	5D	…	…	64	65	66	67

For 40×2 Display

Character	1	2	3	4	5	6	7	8	9	10	…	…	37	38	39	40
DD RAM Address	00	01	02	03	04	05	06	07	08	09	…	…	24	25	26	27
	40	41	42	43	44	45	46	47	48	49	…	…	64	65	66	67

For 40×4 Display

Character	E	1	2	3	4	5	6	7	8	9	10	…	…	37	38	39	40
DD RAM Address	E1	00	01	02	03	04	05	06	07	08	09	…	…	24	25	26	27
		40	41	42	43	44	45	46	47	48	49	…	…	64	65	66	67
	E2	00	01	02	03	04	05	06	07	08	09	…	…	24	25	26	27
		40	41	42	43	44	45	46	47	48	49	…	…	64	65	66	67

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

CG RAM MAPPING

Character Code (DD RAM data)								CG RAM Address						Character Patterns (CG RAM data)								
7	6	5	4	3	2	1	0	5	4	3	2	1	0	7	6	5	4	3	2	1	0	
High							Low	High					Low	High							Low	
											0	0	0				0	1	1	0	0	
											0	0	1				1	0	0	1	0	
											0	1	0				0	0	1	0	0	←Character
											0	1	1				0	1	0	0	0	
0	0	0	0	*	0	0	0	0	0	0	1	0	0	*	*	*	1	1	1	1	0	Pattern
											1	0	1				0	0	0	0	0	
											1	1	0				0	0	0	0	0	
											1	1	1				0	0	0	0	0	←Cursor
											0	0	0				1	1	1	1	1	
											0	0	1				1	0	0	0	1	
											0	1	0				1	0	1	0	1	←Character
											0	1	1				1	0	1	1	1	
0	0	0	0	*	0	0	1	0	0	1	1	0	0	*	*	*	1	0	1	0	1	Pattern
											1	0	1				1	0	0	0	1	
											1	1	0				1	1	1	1	1	
											1	1	1				0	0	0	0	0	←Cursor
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	
											0	0	0				1	1	1	1	1	
											0	0	1				1	0	0	0	1	
											0	1	0				1	1	1	0	1	←Character
											0	1	1				1	0	0	0	1	
0	0	0	0	*	1	1	1	1	1	1	1	0	0	*	*	*	1	0	1	1	1	Pattern
											1	0	1				1	0	0	0	1	
											1	1	0				1	1	1	1	1	
											1	1	1				0	0	0	0	0	←Cursor

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

CHARACTER FONT TABLE

Upper 4 bits / Lower 4 bits	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
0000	CG RAM (1)															
0001	CG RAM (2)															
0010	CG RAM (3)															
0011	CG RAM (4)															
0100	CG RAM (5)															
0101	CG RAM (6)															
0110	CG RAM (7)															
0111	CG RAM (8)															
1000	CG RAM (1)															
1001	CG RAM (2)															
1010	CG RAM (3)															
1011	CG RAM (4)															
1100	CG RAM (5)															
1101	CG RAM (6)															
1110	CG RAM (7)															
1111	CG RAM (8)															

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้ ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้ประโยชน์ด้านการค้า

RELIABILITY TEST

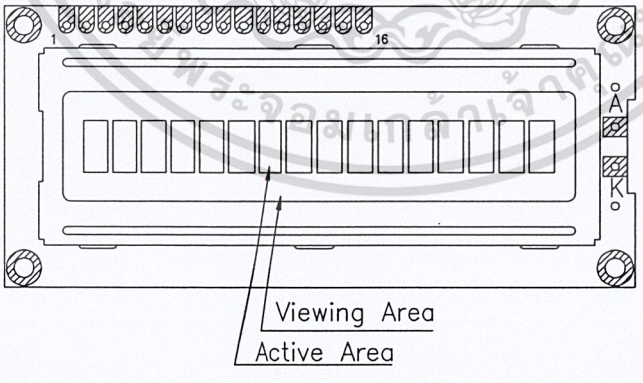
No	Item	Conditions		Note
1	High Temp. Operation	70℃	240 HR	
2	High Temp. Storage	80℃	240 HR	
3	Low Temp. Operation	-20℃	240 HR	
4	Low Temp. Storage	-30℃	240 HR	
5	High Temp./Humid Storage	60℃ 90%RH	240 HR	
6	Thermal Shock	-20℃ ,30min +60℃ ,30min	10 cycles	
7	Vibration Test (IEC-68-2-6)	Frequency : 10~55 Hz Duration : 20 times, 6 min/time Amplitude : 0.75 mm	-	
8	Shock (IEC 68-2-27)	Duration : 11 mS Acceleration : 100g	-	X, Y, Z direction

APPEARANCE CHECK

CONDIITON OF APPEARANCE CHECK:

(1)Specimen shall be checked by eyes in distance of 30cm under 40w-fluorescence lamp.

(2)Checking direction shall be in 45 degree from perpendicular line op specimen surface.



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

HANDLING PRECAUTIONS

(1)Treat polarizer very carefully since it is easy to be damaged.

(2)When cleaning the display surface, use soft cloth (e.g. gauss) with a solvent (recommended below) and wipe lightly.

- ethyl alcohol
- iso-prcolol

Do not wipe the display surface with dry or hard materials that will damage the polarizer surface.

Do not use the following solvents:

- water
- ketone
- aromatics

(3)Direct current causes electro-chemical reaction with remarkable degradation of the display quality. Give careful consideration to prevent direct current at ON/OFF timing and during operation.

(4)Avoid strong shock and drop from the height.

(5)To prevent LCD panels from degradation, do not operate or store them exposed directly to sunshine or high temperature/humidity.

(6)Give careful consideration to avoid electrical static discharge with causes uneven contrast.

(7)Even a small condensation on the contact pads (terminals) causes electro-chemical reaction which makes missing row and column. Give careful attention to avoid condensation. When assembling with zebra connector, clean the surface of the pads with alcohol and keep the air very clean.

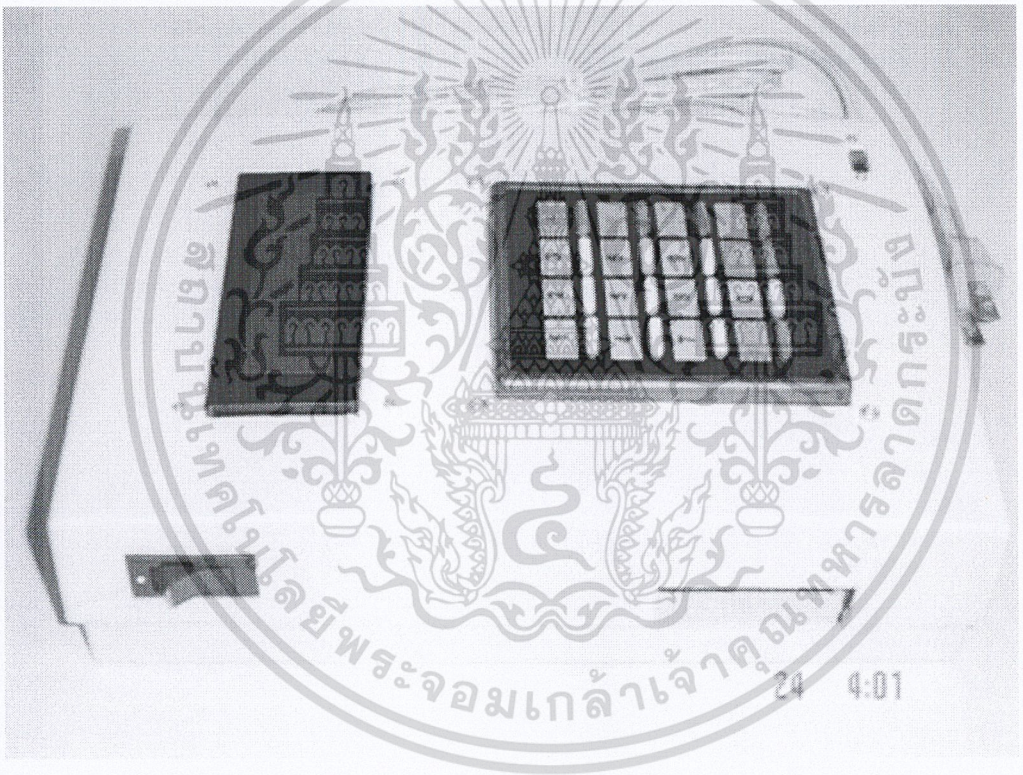
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

LCD PRODUCT QUALITY STANDARD

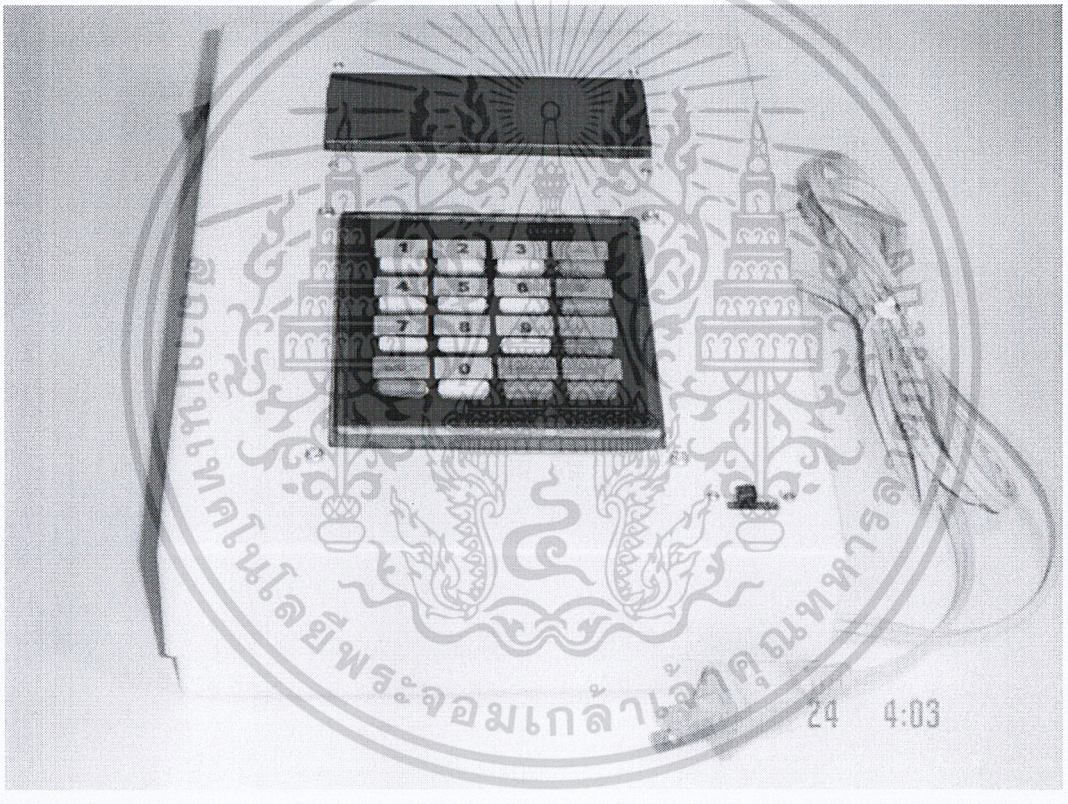
DISPLAY APPEARANCE

No	Item	Criteria
1	inclusions (black spot, white spot, dust)	(1)round type diameter mm(a*) — no of defect* a≦0.20 — neglect 0.20＜a≦0.35 — 5max 0.35＜a — none (2)linear type length mm(l) — width mm(W) — no. of defect na — W≦0.03 — neglect l≦3 — 0.03＜W≦0.08 — 6 3＜l — 0.08＜W — none
2	scratch	1.scratch on protective film is permitted. 2.scratch on polarizer shall be as follow: (1)round type diameter mm(a*) — no of defect a≦0.15 — neglect 0.15＜a≦0.20 — 2 max 0.20＜a — none (2)linear type be judged bye 1.-(2) linear type
3	dent	diameter ＜ 1.5mm
4	bubble	not exceeding 0.5mm average diameter is acceptable between glass and polarizing film
5	pin hole	(a+b)/2 ≦ 0.15mm maximum number: ignored 0.15＜(a+b)/2≦0.20mm maximum number:10
6	dot defect	(a+b)/2≦0.20mm maximum number: ignored 0.20＜(a+b)/2≦0.30mm maximum number:5 x=width
7	contrast irregularity(spot)	diameter spec — no of defect a≦0.50mm — neglect 0.50＜a≦0.75 — 5 0.75＜a≦1.00 — 3 1.00＜a — none
8	dot width	design width ±15%
9	color tone and uniformity	obvious uneven color is not permitted

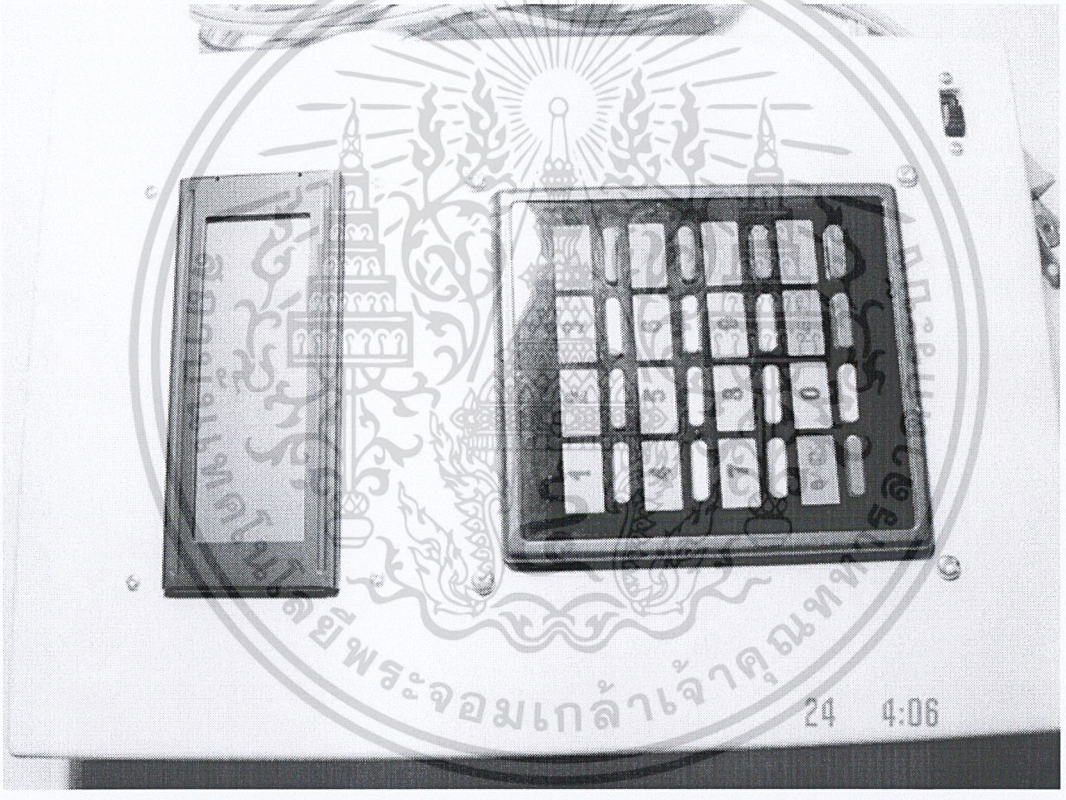
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



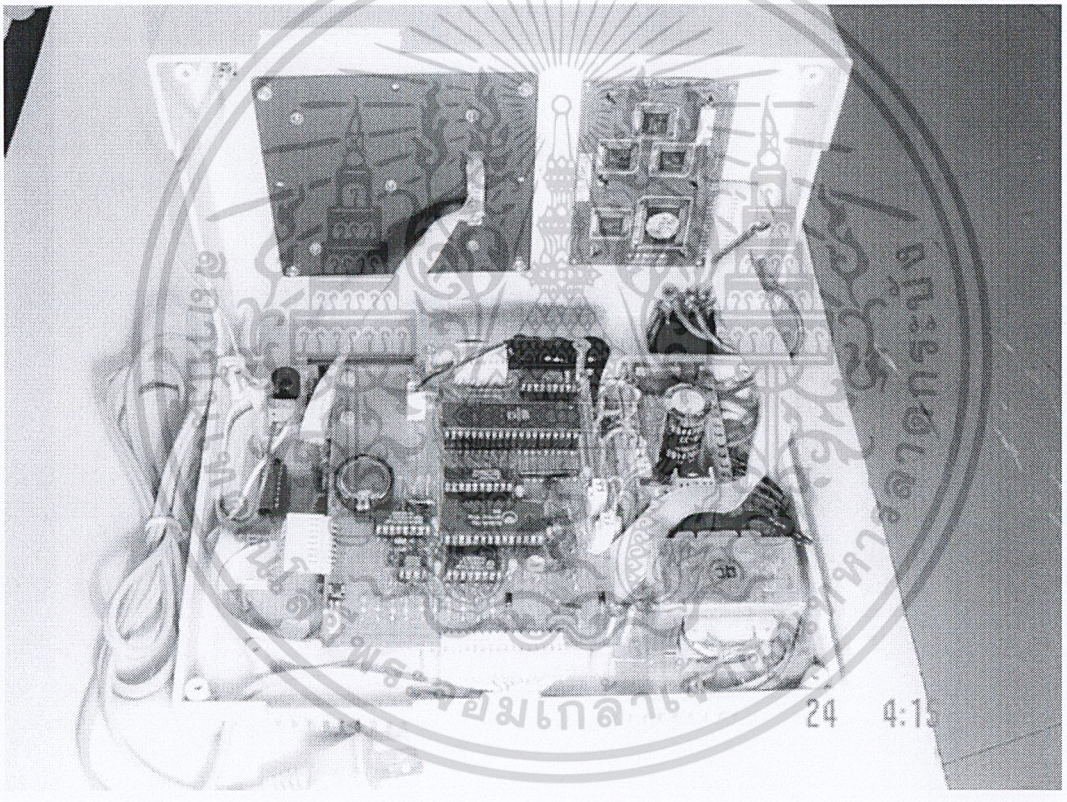
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



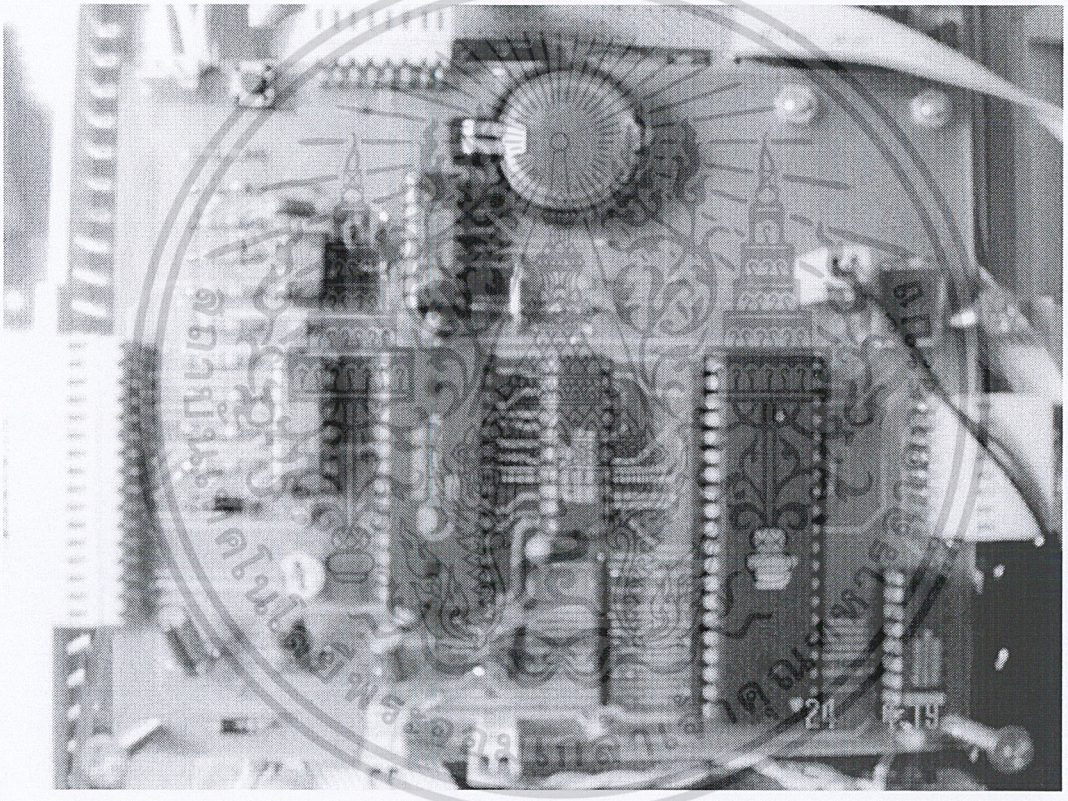
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



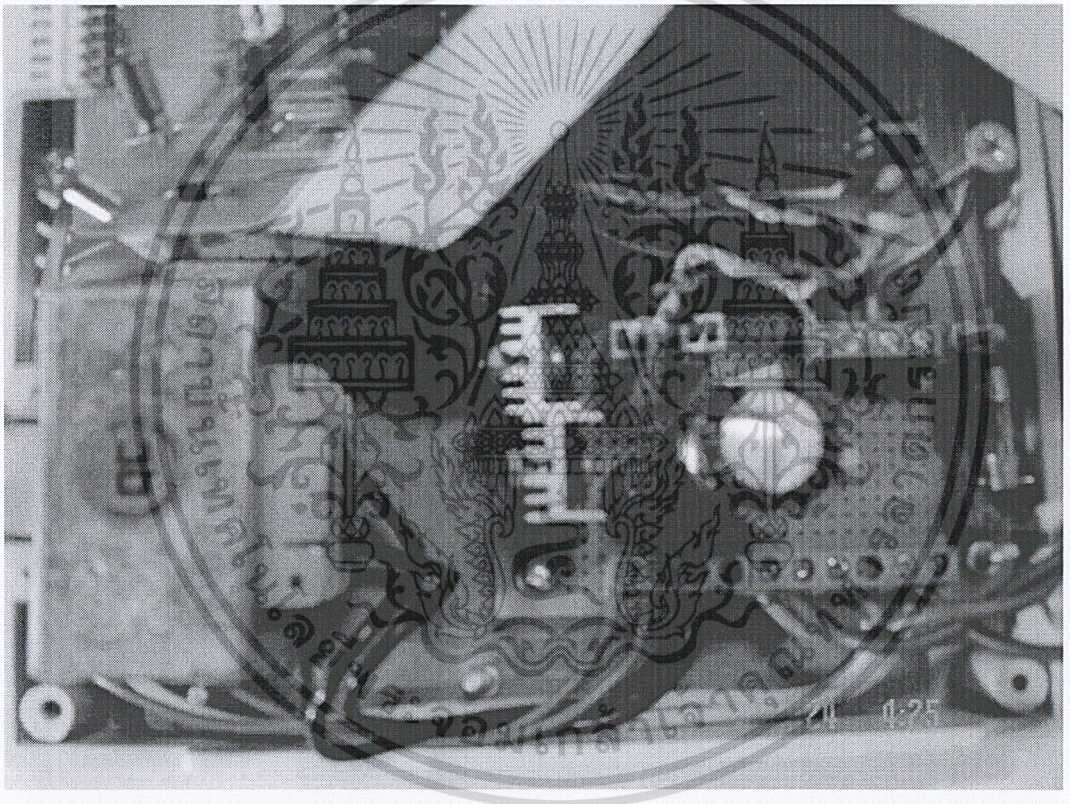
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

โครงงานและปริญญานิพนธ์ฉบับนี้สำเร็จลุล่วงได้ด้วยดี เนื่องจากได้รับความช่วยเหลืออย่างดียิ่งของผู้ช่วยศาสตร์ ดร.เกียรติศักดิ์ คมวัชระ อาจารย์ที่ปรึกษาโครงงาน ซึ่งได้ให้คำแนะนำและข้อคิดเห็นต่างๆที่เป็นประโยชน์อย่างสูงต่อการทำโครงงานด้วยดีเสมอมา ผู้จัดทำโครงงานขอขอบพระคุณอาจารย์ภาควิชาวิศวกรรมระบบควบคุมที่คอยผลักดันและกระตุ้นให้ทำโครงงานจนสำเร็จลุล่วงด้วยดี

สุดท้ายนี้ผู้จัดทำโครงงานใคร่กราบขอบพระคุณ บิดา มารดา และขอบคุณพี่ๆน้องๆและเพื่อนทุกคนที่ให้กำลังใจและความช่วยเหลือ แก่ผู้จัดทำโครงงานเสมอมาจนสำเร็จการศึกษา

เฉลิมพล บวรกิติวงศ์

ธีรวัฒน์ คุณโชติ

มีนาคม 2546

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หนังสืออ้างอิง

ชัยวัฒน์ ลิ้มพรจิตรวิไล และวรพจน์ กรแก้ววัฒนากุล, “เรียนรู้และปฏิบัติการไมโครคอนโทรลเลอร์ MS-51 Flash Microcontroller”, อินโนเวทีฟ เอ็กซ์เพอริเมนต์, 475 หน้า, 2542

สมยศ จุณณะปิยะ “การใช้ไมโครคอนโทรลเลอร์ MS-51”, ภาควิชาวิศวกรรมโทรคมนาคม คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง, 358 หน้า, 2543

สุนทร วิทูรสุรพจน์ “การโปรแกรมภาษาแอสเซมบลีของไมโครคอนโทรลเลอร์ตระกูล 8051”, ซีเอ็ดยูเคชัน, 198 หน้า, 2537.

สุเจตน์ จันทรังษ์ “ไมโครคอนโทรลเลอร์ซีพเดี่ยว 8051”, วิทยาลัยมหานคร, 178 หน้า, 2535.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้