สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

เครื่องควบคุมอุปกรณ์ไฟฟ้าภายในบ้านแบบโปรแกรมได้

PROGRAMMABLE ELECTRIC APPLIANCE CONTROLLER

โดย

นายธีรวัฒน์ คุ้มทรัพย์

นายสิทธิโชค ถิ่นถลาง

เลขหมู่.....
เลขทะเบียน 50186
วัน,เดือน,ปี 27 เม.ย. 2547

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

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิศวกรรมระบบควบคุม

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2545

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์ปีการศึกษา 2545

ภาควิชาวิศวกรรมระบบควบคุม

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง เครื่องควบคุมอุปกรณ์ไฟฟ้าภายในบ้านแบบโปรแกรมได้

ผู้จัดทำ

1. นายธีรวัฒน์ คุ้มทรัพย์
2. นายสิทธิโชค ถิ่นฉลาง

.....อาจารย์ที่ปรึกษา

(เกียรติศักดิ์ ดมวิธระ)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เครื่องควบคุมอุปกรณ์ไฟฟ้าภายในบ้านแบบโปรแกรมได้

นายธีรวัฒน์ คุ้มทรัพย์
นายสิทธิโชค ถิ่นถลาง
ผศ.ดร.เกียรติศักดิ์ คมวัชระ อาจารย์ที่ปรึกษา
ปีการศึกษา 2545

บทคัดย่อ

ปริญญานิพนธ์ฉบับนี้แสดงรายละเอียดของเครื่องควบคุมอุปกรณ์ไฟฟ้าภายในบ้านแบบโปรแกรมได้ เป้าหมายของโครงงานนี้คือการควบคุมอุปกรณ์ไฟฟ้าเพื่อจะเปิดหรือปิดตามเวลาที่ถูกตั้งไว้ และสามารถเลือกโปรแกรมสำหรับการหน่วงเวลาเปิด/ปิดในแต่ละอุปกรณ์ไฟฟ้าได้ โดยจุดหลักของระบบนี้คือการใช้ไมโครคอนโทรลเลอร์ (MCS-51) เป็นตัวประมวลผลกลางและใช้ไอซีฐานเวลา (DS1307) เป็นฐานเวลาอ้างอิง ผู้ใช้สามารถเลือกอุปกรณ์ไฟฟ้า เลือกโปรแกรม และตั้งค่าเวลาโดยใช้คีย์แพด (4 x 4) และแสดงผลด้วย LCD ขนาด(4 แถว x 20 ตัวอักษร) หลังจากที่ผู้ใช้ตั้งค่าโปรแกรมการทำงานแล้ว ในแต่ละอุปกรณ์ไฟฟ้าก็จะ เปิด/ปิด ตามโปรแกรมที่ถูกตั้งไว้โดยเปรียบเทียบกับฐานเวลาอ้างอิง

Abstract

This paper presents the detail of programmable electric appliance controller .The object of this project is control electric devices to on or off follow the time set and can be selected program for delay on/off in each electric device. The main point of this system is used microcontroller (MCS-51) to be central processing unit and used the real time IC (DS1307) to be a referent time base of the system. The user can be selected electric device, select program, set time by use keypad (4 x4) and display to LCD size (4 row x 20 character). After the user set program all of electric device. In each electric device will on/off follow the program set by compare the referent time base.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

	หน้า
บทที่ 1 บทนำ	1
1.1 กล่าวนำ	1
1.2 วัตถุประสงค์หลักของโครงงาน	1
1.3 ขอบเขตของโครงงาน	1
1.4 ขั้นตอนการดำเนินงาน	2
1.5 ประโยชน์ที่ได้รับจากโครงงาน	2
บทที่ 2 ทฤษฎี	3
2.1 ไมโครคอนโทรลเลอร์ตระกูล MCS-51	3
2.2 ไอซีสร้างฐานเวลาจริงเบอร์ DS1307	26
2.3 หลักการของระบบบัส I^2C	31
2.4 ไอซีขยายพอร์ตเบอร์ PCF8574	37
2.5 LCD (Liquid Crystal Diaplays)	40
บทที่ 3 การคำนวณและการสร้าง	48
3.1 แนวคิดและหลักการทำงาน	48
3.2 วงจรภาคอินพุตของ MCS-51	49
3.3 วงจรภาคเอาต์พุต MCS-51	51
3.4 วงจรส่วนขับโหลด	53
3.5 โฟลวชาร์ตของโปรแกรมหลัก	54
บทที่ 4 การทดลองและผลการทดลอง	56
บทที่ 5 บทวิจารณ์และสรุป	65
5.1 สรุปผลการทดลอง	65
5.2 ปัญหาที่เกิดขึ้นจากการทดลอง	65
ภาคผนวก	
กิตติกรรมประกาศ	
หนังสืออ้างอิง	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูปภาพ

หน้า

รูปที่ 2.1 โครงสร้างพื้นฐานของไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลชในอนุกรมAT89Cxx 4

รูปที่ 2.2 โครงสร้างของไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลชในอนุกรม AT89Sxx 5

รูปที่ 2.3 รายละเอียดโครงสร้างหลักของไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลช 6

รูปที่ 2.4 การจัดขามาตรฐานของไมโครคอนโทรลเลอร์ MCS-51 8

รูปที่ 2.5 วงจรภายในของพอร์ตทุกพอร์ตในไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลช 10

รูปที่ 2.6 วงจรพูลอัปภายในพอร์ตไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลช 11

รูปที่ 2.7 ไซเกิลการทำงานของไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลช 13

รูปที่ 2.8 การจัดสรรหน่วยความจำโปรแกรมของไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลช 16

รูปที่ 2.9 การเชื่อมต่อหน่วยความจำโปรแกรมภายนอกของไมโครคอนโทรลเลอร์ MCS-51 17

รูปที่ 2.10 การเชื่อมต่อหน่วยความจำข้อมูลภายนอกของไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลช 18

รูปที่ 2.11 การจัดสรรพื้นที่ของหน่วยความจำข้อมูลภายในไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลช 18

รูปที่ 2.12 การจัดสรรพื้นที่ของหน่วยความจำข้อมูลภายในส่วนล่างของไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลช 19

รูปที่ 2.13 โครงสร้างของหน่วยความจำข้อมูลภายในในส่วนบนของไมโครคอนโทรลเลอร์ MCS-51 20

รูปที่ 2.14 การจัดสรรพื้นที่ของรีจิสเตอร์ฟังก์ชั่นพิเศษ SFR 20

รูปที่ 2.15 การจัดขาไอซี DS 1307 26

รูปที่ 2.16โครงสร้างภายในของไอซีรีลไทม์คล็อกเบอร์ DS1307 27

รูปที่ 2.17 การจัดสรรหน่วยความจำแรมภายใน DS1307 29

รูปที่ 2.18 รูปแบบของข้อมูลสำหรับติดต่อกับ DS1307 ในโหมดการเขียนข้อมูล 30

รูปที่ 2.19 รูปแบบของข้อมูลสำหรับติดต่อกับ DS1307 ในโหมดการอ่านข้อมูล 30

รูปที่ 2.20 ผังแสดงการเชื่อมต่อของอุปกรณ์ต่าง ๆ บนบัส I^2C 31

รูปที่ 2.21 แสดงวงจรเอาท์พุตของอุปกรณ์ในระบบบัส I^2C 32

รูปที่ 2.22 การต่อตัวต้านทานพูลอัปบนสายสัญญาณในระบบบัส I^2C 33

รูปที่ 2.23 การต่อตัวต้านทาน R_S เพื่อลดสัญญาณรบกวนขนาดใหญ่ที่อาจเข้ามาในบัส I^2C 33

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูปภาพ (ต่อ)

หน้า

รูปที่ 2.24 ไดอะแกรมเวลาแสดงสถานะต่าง ๆ ในบัส I^2C 35

รูปที่ 2.25 รูปแบบของข้อมูลกำหนดแอดเดรสที่ใช้ในการอ้างถึงแบบ 7 บิต 36

รูปที่ 2.26 รูปแบบของข้อมูลอนุกรมที่ใช้ติดต่อกับอุปกรณ์บัส I^2C เมื่อใช้การอ้างถึงแบบ 7 บิต 37

รูปที่ 2.27 รูปแบบของข้อมูลอนุกรมที่ใช้ติดต่อกับอุปกรณ์บัส I^2C เมื่อใช้การอ้างถึงแบบ 10 บิต 37

รูปที่ 2.28 การจัดขาไอซีขยายพอร์ตอินพุตเอาต์พุต PCF8574/ PCF8574A 38

รูปที่ 2.29 วงจรภายในขาพอร์ตของไอซี PCF8574/PCF8574A 40

รูปที่ 2.30 ไดอะแกรมการทำงานของโมดูล LCD แบบอักขระ 41

รปที่ 3.1 บล็อกไดอะแกรมของวงจร 48

รูปที่ 3.2 บล็อกไดอะแกรมภาคอินพุต 49

รูปที่ 3.3 วงจรการเชื่อมต่อภาคอินพุต 50

รูปที่ 3.4 บล็อกไดอะแกรมภาคเอาต์พุต 51

รูปที่ 3.5 วงจรการเชื่อมต่อภาคเอาต์พุต 52

รูปที่ 3.6 วงจรส่วนขับโหลด 53

รูปที่ 3.7 โฟลวชาร์ตของโปรแกรมหลัก 54

รูปที่ 4.1 สภาวะเริ่มต้นการทำงานบนหน้าจอ LCD 56

รูปที่ 4.2 การแสดงในสถาวะการทำงานปกติ 56

รูปที่ 4.3 ผังการตั้งค่าโปรแกรม 57

รูปที่ 4.4 แสดงการใส่ค่าช่วงเวลาที่ต้องการสั่งงานโหลด 57

รูปที่ 4.5 แสดงการใส่ค่าสถานะของโหลด 58

รูปที่ 4.6 แสดงเคอร์เซอร์ที่รอการใส่ค่าตัวต่อไป 58

รูปที่ 4.7 แสดงการใช้ปุ่ม CLEAR 58

รูปที่ 4.8 แสดงการใส่ค่าข้อมูลใหม่หลังจากกดปุ่ม CLEAR 59

รูปที่ 4.9 แสดงการใช้ปุ่ม ENTER 59

รูปที่ 4.10 แสดงการจบการใส่ค่าในแต่ละโหลด 59

รูปที่ 4.11 หน้าจอของโปรแกรมหน่วงเวลา 60

รูปที่ 4.12 หน้าจอกรณีที่มีการสั่งงานให้ปิดโหลดทั้งหมด 60

รูปที่ 4.13 ผังการใช้งานโปรแกรมแบบสั่งงานโดยตรง 61

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูปภาพ (ต่อ)

	หน้า
รูปที่ 4.14 ตัวอย่างการตั้งค่าแบบสั่งงานโดยตรง	61
รูปที่ 4.15 ผังการตั้งค่าเวลา	62
รูปที่ 4.16 ตัวอย่างการตั้งค่าเวลา	62
รูปที่ 4.17 ผังการตั้งค่าวัน,วันที่,เดือน,ปี	63
รูปที่ 4.18 ตัวอย่างการตั้งค่าวัน,วันที่,เดือน,ปี	63

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

หน้า

ตารางที่ 1.1 ขอบเขตของโครงงาน (รูปแบบการพัฒนาและปรับปรุง) 1

ตารางที่ 2.1 รายละเอียดโดยสรุปบางส่วนของไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลช ที่ Atmel ผลิตขึ้นและใช้ในการอ้างอิง 5

ตารางที่ 2.2 หน้าที่พิเศษของพอร์ต 1 ในไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลช 9

ตารางที่ 2.3 การเลือกแบงก์ของหน่วยความจำส่วนล่างเพื่อติดต่อกับรีจิสเตอร์แบงก์ R0-R7 22

ตารางที่ 2.4 รายละเอียดหน้าที่การทำงานของแต่ละขาของไอซี PCF8574/PCF8574A 39

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

1.1 กล่าวนำ

เครื่องควบคุมอุปกรณ์ไฟฟ้าภายในบ้านแบบโปรแกรมได้ (Programmable Electric Appliance Controller) เป็นเครื่องควบคุมที่ทำหน้าที่ควบคุมการเปิดหรือปิด อุปกรณ์เครื่องใช้ไฟฟ้าแบบอัตโนมัติ ตามช่วงเวลาการทำงานที่ผู้ใช้ต้องการ และสามารถแก้ไขปรับเปลี่ยนช่วงเวลาได้ เพื่อที่จะสามารถนำไปแทนที่ระบบการควบคุมแบบเดิมที่ผู้ใช้ต้องคอย เปิด/ปิด การทำงานของเครื่องใช้ไฟฟ้าอยู่เป็นประจำ ให้เป็นการทำงานแบบอัตโนมัติ และสามารถช่วยประหยัดพลังงานไฟฟ้าในช่วงเวลาที่อุปกรณ์นั้นๆ ไม่ได้ใช้งาน

1.2 วัตถุประสงค์หลักของการพัฒนาโครงงาน

1. แก้ไขข้อเสียและข้อบกพร่องต่างๆ ของ เครื่องควบคุมอุปกรณ์ไฟฟ้าภายในบ้านแบบโปรแกรมได้เดิม (โครงงานปี 2544)

2. ปรับปรุงรูปแบบของซอฟต์แวร์ การใช้งานให้มีรูปแบบหลากหลายและเอื้อประโยชน์ต่อผู้ใช้มากขึ้น

3. ปรับปรุงรูปแบบของฮาร์ดแวร์ ให้มีความเหมาะสมต่อการนำไปใช้งานจริง

1.3 ขอบเขตของโครงงาน

รูปแบบการพัฒนาและปรับปรุง

โครงงานปี 2544	โครงงานปี 2545
ทำงานได้ 8 ช่วงเวลา /โหลด	ทำงานได้ 98 ช่วงเวลา /โหลด
มีโปรแกรมสั่งเปิดและปิดเท่านั้น	มีโปรแกรมให้เลือกในทุกโหลดประกอบด้วย 1. โปรแกรมทำงานตามเวลาที่ตั้ง 2. โปรแกรมทำงานหน่วงเวลาเปิด-ปิด
จอ LCD แสดงผลมีขนาดเล็ก(2 บรรทัด 16 ตัวอักษร) ดูสภาวะโหลดได้ครั้งละ 1 โหลด	ใช้จอขนาด (4 บรรทัด 20 ตัวอักษร) ซึ่งช่วยแสดงสภาวะการทำงานของโหลดทั้ง 8 โหลดในเวลาเดียวกันได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จอ LCD แสดงผลในที่มืดไม่ได้	ใช้การอินเตอร์รัปต์จากคีย์แพดควบคุมการ เปิด/ปิด LED Blacklight เพื่อแสดงผลในที่มืด
ทำงานได้เฉพาะตามช่วงเวลาที่ผู้ใช้โปรแกรมไว้เท่านั้น	เพิ่มโหมดการทำงานเป็น 3 โหมด 1. ทำงานช่วงเวลาที่ผู้ใช้ใส่ไว้ (Auto) 2. เลือก เปิด/ปิดโหลดได้โดยตรง (Manual) 3. ปิดโหลดทั้งหมดโดยทันที (Disable Loads)
เมื่อไฟดับข้อมูลจะหาย	เพิ่มวงจร ไฟเลี้ยงสำรอง (Backup) เพื่อรักษาข้อมูลกรณีไฟดับ
มีขนาดตัวเครื่องที่ใหญ่เกินไป	ปรับลดขนาดทั้ง ตัวเครื่อง และ แผงวงจร(PCB) เพื่อให้เหมาะสมกับการใช้งาน

ตารางที่ 1.1 ขอบเขตของโครงงาน (รูปแบบการพัฒนาและปรับปรุง)

1.4 ขั้นตอนการดำเนินงาน

1. ศึกษารายละเอียดของ โครงงาน
2. ออกแบบ และ ทดลองวงจร พร้อมทั้งเขียน โปรแกรมควบคุมการทำงาน
3. จัดทำแผ่นวงจรพิมพ์ (PCB) และติดตั้งลงกล่อง
4. จัดทำปริญญานิพนธ์

1.5 ประโยชน์ที่ได้รับจากโครงงาน

ประโยชน์ที่ได้จากการทำโครงงาน คือ ความรู้ที่ได้รับจากการศึกษาและการออกแบบวงจร รวมถึงโปรแกรมต่างๆ ที่นำมาใช้กับโครงงานนี้ เป็นการนำความรู้ที่ได้มาใช้ในการปฏิบัติงานจริง อีกทั้งยังช่วยให้มีประสบการณ์ในการทำงานร่วมกัน การจัดการและการแก้ไขกับปัญหาต่างๆ ที่เกิดขึ้น ซึ่งเป็นประโยชน์อย่างมากในการทำงานและการใช้ชีวิตประจำวัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2
ทฤษฎี

2.1 ไมโครคอนโทรลเลอร์ตระกูล MCS-51

โครงสร้างและสถาปัตยกรรมของไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลช

ไมโครคอนโทรลเลอร์ตระกูล MCS-51 ที่กล่าวถึงนี้จะอ้างถึงไมโครคอนโทรลเลอร์ตระกูล MCS-51 ซึ่งมีหน่วยความจำภายในเป็นแบบแฟลช (flash memory) ของ Atmel Corporation มีเบอร์ขึ้นต้นด้วย AT89 เหตุผลที่ใช้ไมโครคอนโทรลเลอร์แบบนี้ในการกล่าวถึงเพื่อใช้งานไมโครคอนโทรลเลอร์ตระกูล MCS-51 มีด้วยกันหลายประการดังนี้

1. หน่วยความจำโปรแกรมภายในตัวไมโครคอนโทรลเลอร์เป็นแบบแฟลช ทำให้สามารถลบและเขียนใหม่ได้นับพันครั้ง จึงสามารถใช้งานในรูปแบบของไมโครคอนโทรลเลอร์ชิปเดี่ยวไม่ต้องใช้หน่วยความจำภายนอกส่งผลให้สามารถใช้งานพอร์ตอินพุตเอาต์พุตของไมโครคอนโทรลเลอร์ได้อย่างเต็มประสิทธิภาพ

2. ต้นทุนและเวลาในการพัฒนาระบบไมโครคอนโทรลเลอร์ลดลงอย่างมาก เนื่องจากไม่ต้องใช้เครื่องมือพัฒนาจำพวกอีมูเลเตอร์และเครื่องโปรแกรมอีพรอม

3. บริษัทผู้ผลิตได้ทำการผลิตไมโครคอลโทรลเลอร์ตระกูลนี้ออกมาหลายเบอร์และมีความสามารถแตกต่างกันไป ทำให้มีทางเลือกในการใช้งานสูง

4. ด้วยการใช้หน่วยความจำภายในตัวไมโครคอนโทรลเลอร์ ทำให้สามารถป้องกันการคัดลอกข้อมูลของหน่วยความจำโปรแกรมได้เป็นอย่างดี

5. ในบางเบอร์ของไมโครคอนโทรลเลอร์ที่ผลิตโดย Atmel สามารถทำการโปรแกรมข้อมูลในหน่วยความจำโปรแกรมได้โดยที่ไม่ต้องถอดตัวไมโครคอนโทรลเลอร์ออกมาทำการโปรแกรมใหม่หรือเรียกว่า การโปรแกรมในวงจร หรือในระบบ (In-system programming) โดยใช้ลักษณะการติดต่อแบบ SPI (Serial Peripheral Interface) ทำให้การพัฒนาหรือการซ่อมบำรุง ตลอดจนการปรับปรุงหรืออัพเกรดข้อมูลในหน่วยความจำโปรแกรมทำได้อย่างสะดวกภายใต้งบประมาณที่ไม่สูงมากนัก

6. ชุดคำสั่งและสถาปัตยกรรมพื้นฐานเหมือนกับไมโครคอนโทรลเลอร์ MCS-51 ของผู้ผลิตไม่ว่าจะเป็นอินเทล, ซีเมนส์, หรือดัลลัส

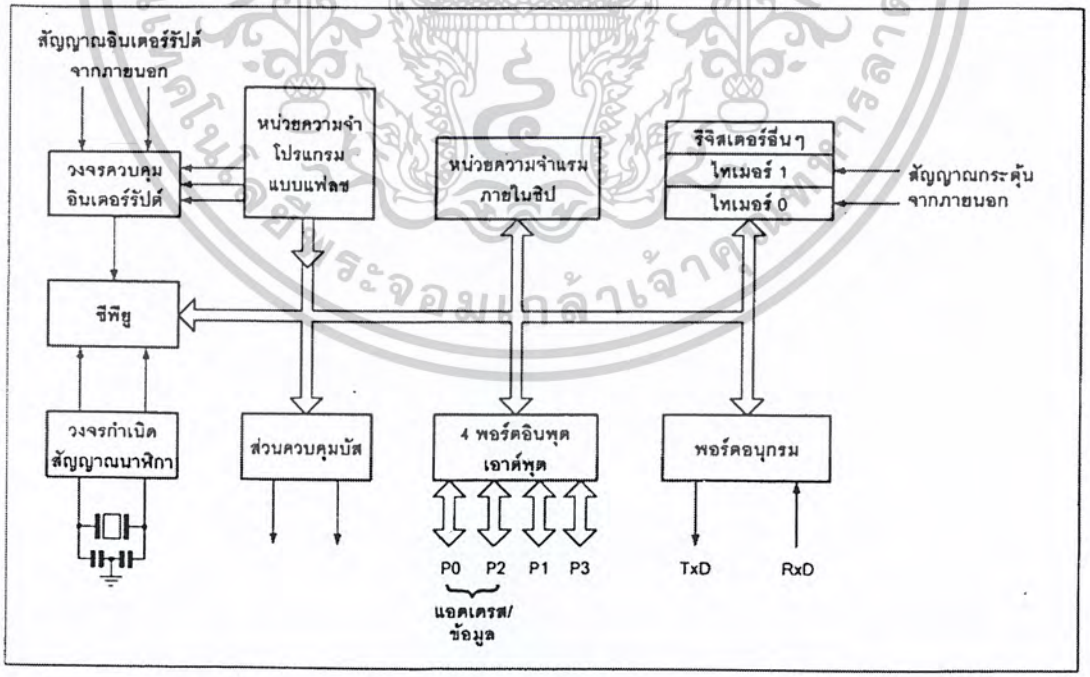
คุณสมบัติทางเทคนิคของไมโครคอนโทรลเลอร์ตระกูล MCS-51 อนุกรม AT89xx

- เป็นไมโครคอนโทรลเลอร์ที่ใช้ซีพียูขนาด 8 บิต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ภายในมีหน่วยความจำโปรแกรมเป็นแบบแฟลชสามารถลบและเขียนใหม่ได้พันครั้ง
- หน่วยความจำข้อมูลพื้นฐานเป็นหน่วยความจำแบบแรม ในบางเบอร์จะมีหน่วยความจำแบบอีอีพรอมเพิ่มเติม
- ขาพอร์ตเป็นแบบสองทิศทาง สามารถใช้งานเป็นได้ทั้งอินพุตและเอาต์พุต
- มีวงจรสื่อสารอนุกรมแบบฟูลดูเพล็กซ์
- ไทเมอร์/เคาน์เตอร์ขนาด 16 บิตอย่างน้อย 2 ตัว
- สามารถรองรับแหล่งกำเนิดอินเตอร์รัปต์ได้ 6 ประเภท
- สามารถขยายหน่วยความจำภายนอกเพิ่มเติมได้สูงสุด 64 กิโลไบต์
- มีวงจรกำเนิดสัญญาณนาฬิกาอยู่ภายในชิป
- มีวงจรสื่อสารอนุกรมแบบ SPI สำหรับในอนุกรม AT89Sxx
- มีวอตช์ด็อกไทเมอร์ในตัว สำหรับในอนุกรม AT89Sxx

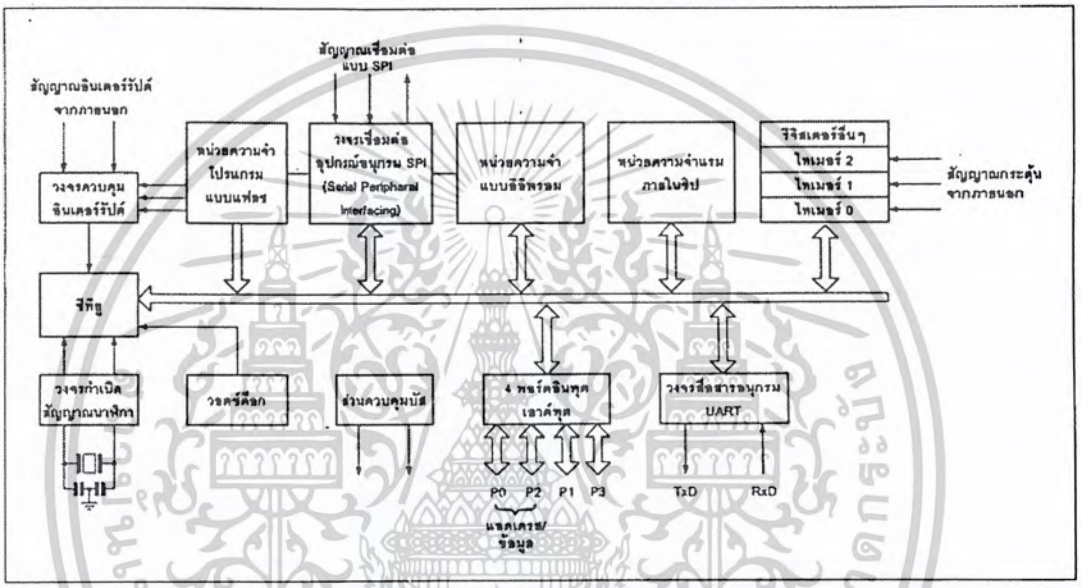
ในรูปที่ 2.1 เป็นโครงสร้างพื้นฐานของไมโครคอนโทรลเลอร์ MCS-51 ในอนุกรม AT89Cxx จะเห็นได้ว่า โครงสร้าง AT89Cxx จะเหมือนกับไมโครคอนโทรลเลอร์ตระกูล MCS-51 พื้นฐานหากแต่แตกต่างกันเฉพาะหน่วยความจำโปรแกรมแบบแฟลชที่เพิ่มเติมเข้ามา หากเป็น ไมโครคอนโทรลเลอร์ในอนุกรม 87xx หน่วยความจำโปรแกรมภายในจะเป็นแบบอีพรอม และบางเบอร์สามารถโปรแกรมได้เพียงครั้งเดียว



รูปที่ 2.1 โครงสร้างพื้นฐานของไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลชในอนุกรม AT89Cxx

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำหรับในรูปที่ 2.2 เป็นโครงสร้างพื้นฐานของอนุกรม AT89Sxx จะเห็นได้ว่ามีส่วนประกอบที่เพิ่มเติมแตกต่างจาก AT89Cxx อยู่หลายส่วน อาทิ วงจรเชื่อมต่ออนุกรมแบบ SPI ซึ่งในไมโครคอนโทรลเลอร์อนุกรมนี้ใช้ในการเขียนข้อมูลลงในหน่วยความจำโปรแกรมโดยไม่ต้องถอดตัวชิปออกไปจากระบบหรือเรียกว่า การโปรแกรมในวงจร ไทเมอร์/เคาน์เตอร์ขนาด 16 บิตที่เพิ่มเติมเข้ามาอีกหนึ่งตัวเป็นไทเมอร์ 2 และวงจรวอตช์ด็อกที่ใช้ในการตรวจสอบการทำงานผิดพลาดของซีพียู



รูปที่ **2.2** โครงสร้างของไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลชในอนุกรม AT89SXX

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

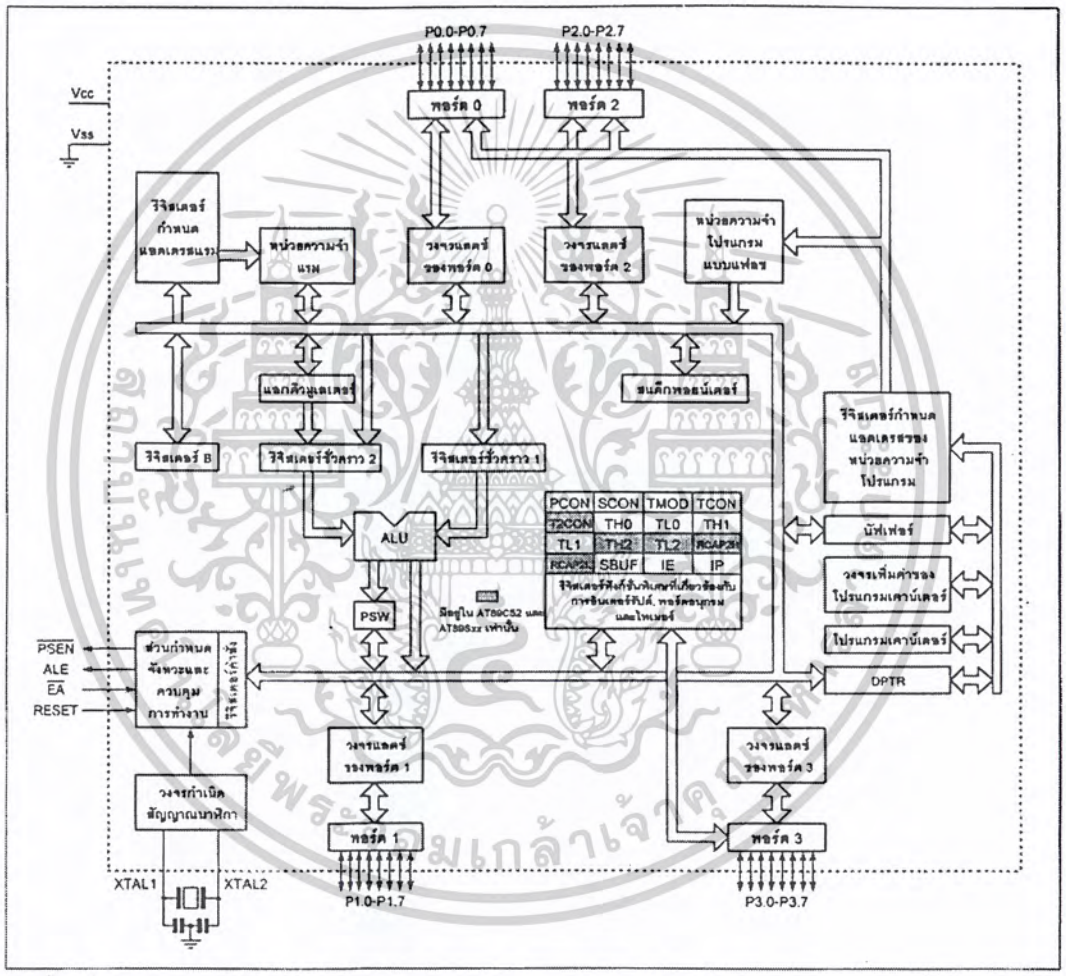
ตารางที่ 2.1 รายละเอียดโดยสรุปบางส่วนของไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลชที่ Atmel ผลิตขึ้นและใช้ในการอ้างอิง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในตารางที่ 2.1 แสดงรายละเอียดบางส่วนของไมโครคอนโทรลเลอร์ ตระกูล MCS-51 แต่ละเบอร์ที่ Atmel ผลิตขึ้น และมีใช้งานอยู่ในปัจจุบัน

การจัดขาของไมโครคอนโทรลเลอร์ MCS-51

ไมโครคอนโทรลเลอร์ MCS-51 ทุกเบอร์จะมีสถาปัตยกรรมและขาใช้งานพื้นฐานเหมือนกันดังแสดงในรูปที่ 2.3 และ 2.4 โดยมีรายละเอียดขั้นต้น ดังนี้



รูปที่ 2.3 รายละเอียดโครงสร้างหลักของไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลช

ขา Vcc ใช้สำหรับต่อไฟเลี้ยง +5V

ขา GND เป็นขากราวด์ สำหรับต่อกับกราวด์ของระบบ

ขาพอร์ต (P0.0 - P0.7) มี 8 ขา แต่ละขาสามารถกำหนดให้เป็นได้ทั้งอินพุตและเอาต์พุต สำหรับใช้งานทั่วไป ถ้าหากต้องการกำหนดให้ขาพอร์ต 0 ขาใดขาหนึ่งเป็นอินพุตสามารถทำได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยการเขียน ข้อมูล "1" ไปยังแต่ละบิตของพอร์ตที่ต้องการติดต่อด้วย ส่งผลให้ขาพอร์ตนั้นมีสถานะปล่อยลอย (float) จึงมีอินพุตอิมพีแดนซ์สูง สามารถใช้งานเป็นขาพอร์ตอินพุตได้ นอกจากนั้นขาพอร์ตนี้ยังถูกใช้งานในการติดต่อกับขาแอดเดรสไบต์ต่ำของหน่วยความจำภายนอก (A0-A7) และขาข้อมูล (D0-D7) โดยใช้กระบวนการมัลติเพล็กซ์เข้าช่วย เพื่อสลับการทำงานให้เป็นได้ทั้งขาติดต่อแอดเดรสและขาข้อมูล

ขาพอร์ต1 (P1.0 - P1.1) มี 8 ขา แต่ละขาสามารถกำหนดให้เป็นได้ทั้งอินพุตและเอาต์พุต สำหรับใช้งานทั่วไป ถ้าหากต้องการกำหนดให้ขาพอร์ต 0 ขาใดขาหนึ่งเป็นอินพุตสามารถทำได้โดยการเขียนข้อมูล "1" ไปยังแต่ละบิตของพอร์ตที่ต้องการติดต่อด้วย นอกจากนั้นในอนุกรม AT89SXX จะใช้ขา P1.0 เป็นขาอินพุตสำหรับนับค่าของไทเมอร์ 2 และ P1.1 เป็นขาอินพุตทริกเกอร์ของไทเมอร์ 2 ในขณะที่ขา P1.4 ถึง P1.7 เป็นขาสำหรับเชื่อมต่อแบบ SPI เพื่อทำการโปรแกรมข้อมูลในระบบ

ขาพอร์ต2 (P2.0 – P2.7) มี 8 ขา แต่ละขาสามารถให้เป็นได้ทั้งอินพุตและเอาต์พุตสำหรับใช้งานทั่วไป ถ้าหากต้องการกำหนดให้ขาพอร์ต 0 ขาใดขาหนึ่งเป็นอินพุตสามารถทำได้โดยการเขียนข้อมูล "1" ไปยังแต่ละบิตของพอร์ตที่ต้องการติดต่อด้วย ส่งผลให้ขาพอร์ตนั้นมีสถานะปล่อยลอย (float) จึงมีอินพุตอิมพีแดนซ์สูง สามารถใช้งานเป็นขาพอร์ตอินพุตได้ นอกจากนั้นาพอร์ตนี้ยังถูกใช้งานในการติดต่อกับขาแอดเดรสไบต์สูงของหน่วยความจำภายนอก (A8 – A15)

ขาพอร์ต3 (P3.0 - P3.7) มี 8 ขา แต่ละขาสามารถกำหนดให้เป็นได้ทั้งอินพุตและเอาต์พุต สำหรับใช้งานทั่วไป ถ้าหากต้องการกำหนดให้ขาพอร์ต 0 ขาใดขาหนึ่งเป็นอินพุตสามารถทำได้โดยการเขียนข้อมูล "1" ไปยังแต่ละบิตของพอร์ตที่ต้องการติดต่อด้วย ส่งผลให้ขาพอร์ตนั้นมีสถานะปล่อยลอย (float) จึงมีอินพุตอิมพีแดนซ์สูง สามารถใช้งานเป็นขาพอร์ตอินพุตได้ นอกจากนั้นขาพอร์ต 3 ยังเป็นขาที่มีหน้าที่การใช้งานพิเศษ ดังมีรายละเอียดขั้นต้นดังต่อไปนี้

P3.0 ใช้เป็นขาอินพุตสำหรับรับข้อมูลจากการสื่อสารแบบอนุกรม หรือขา RxD

P3.1 ใช้เป็นขาอินพุตสำหรับส่งข้อมูลจากการสื่อสารแบบอนุกรม หรือขา TxD

P3.2 ใช้เป็นขาอินพุตสำหรับรับสัญญาณอินเตร์รัปต์จากภายนอกช่องที่ 0 หรือขา $\overline{INT0}$

P3.3 ใช้เป็นขาอินพุตสำหรับรับสัญญาณอินเตร์รัปต์จากภายนอกช่องที่ 1 หรือขา $\overline{INT1}$

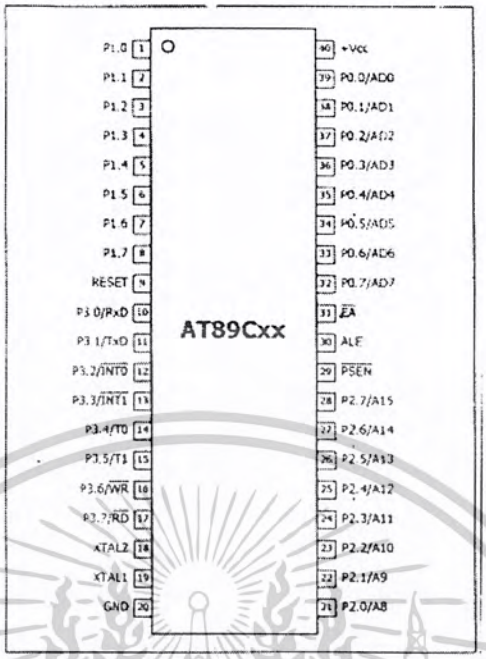
P3.4 ใช้เป็นขาอินพุตสำหรับรับสัญญาณไทเมอร์จากภายนอกช่องที่ 0 หรือขา T0

P3.5 ใช้เป็นขาอินพุตสำหรับรับสัญญาณอินเตอร์รัปต์จากภายนอกช่องที่ 1 หรือขา T1

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

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

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.4 การจัดขามาตรฐานของไมโครคอนโทรลเลอร์ MCS-51

ขารีเซต ใช้ในการรีเซตการทำงานของไมโครคอนโทรลเลอร์ โดยในการป้อนสัญญาณเพื่อรีเซต สถานะที่ขานี้ต้องอยู่ในระดับรีเซตอย่างน้อย 2 แมชชีนไซเกิล โดยที่วงจรกำเนิดสัญญาณนาฬิกายังคงทำงานต่อเนื่องไปอย่างเป็นปกติ

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

$\overline{PSEN}$ (Program Store Enable) ขานี้ใช้ในการส่งสัญญาณเพื่อร้องขอติดต่อกับหน่วยความจำโปรแกรมภายนอก เมื่อไมโครคอนโทรลเลอร์ต้องการอ่านข้อมูลจากหน่วยความจำโปรแกรมภายนอกตัวไมโครคอนโทรลเลอร์ ต้องการอ่านข้อมูลจากหน่วยความจำโปรแกรมภายนอก ตัวไมโครคอนโทรลเลอร์จะส่งสัญญาณออกมาที่ขานี้ 2 ครั้งในแต่ละแมชชีนไซเกิล แต่ถ้าหากติดต่อกับหน่วยความจำข้อมูลภายนอก ขานี้จะไม่มีการส่งสัญญาณใด ๆ ออกมา

ขา $\overline{EA}$/Vpp (External Access enable/Programming voltage input) ใช้สำหรับเลือกการติดต่อหน่วยความจำโปรแกรมจากภายนอกหรือภายในตัวไมโครคอนโทรลเลอร์ ถ้าหากขานี้เป็น "0" เป็นการเลือกให้ไมโครคอนโทรลเลอร์ติดต่อกับหน่วยความจำโปรแกรมภายนอก แต่ถ้าหากขานี้เป็น"1"เป็นการเลือกให้ไมโครคอนโทรลเลอร์ติดต่อกับหน่วยความจำภายในตัวไมโคร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คอนโทรลเลอร์ นอกจากนี้ ที่ขานี้นยังใช้เป็นขาอินพุตสำหรับรับแรงดันไฟสูงสำหรับการโปรแกรมหน่วยความจำภายในไมโครคอนโทรลเลอร์ สำหรับในไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลชต้องการแรงดันสำหรับการโปรแกรม +12V

ขา XTAL1 และ XTAL2 เป็นขาสำหรับต่อคริสตอลเพื่อสร้างสัญญาณนาฬิกาในการกำหนดจังหวะการทำงานของไมโครคอนโทรลเลอร์

โครงสร้างและการทำงานของพอร์ต

ไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลชมีพอร์ตให้ใช้งานทั้งสิ้น 4 พอร์ตคือ พอร์ต 0 ถึงพอร์ต 3 แต่ละพอร์ตมีขนาด 8 บิต เป็นพอร์ตแบบ 2 ทิศทาง กล่าวคือ สามารถเป็นได้ทั้งอินพุตสำหรับรับสัญญาณข้อมูลเข้าและเอาต์พุตสำหรับส่งสัญญาณข้อมูลออก ทุกพอร์ตของไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลชมีวงจรแลตช์และวงจรขับตลอดจนบัฟเฟอร์อินพุต ดังแสดงให้เห็นในสถาปัตยกรรมในรูปที่ 2.3

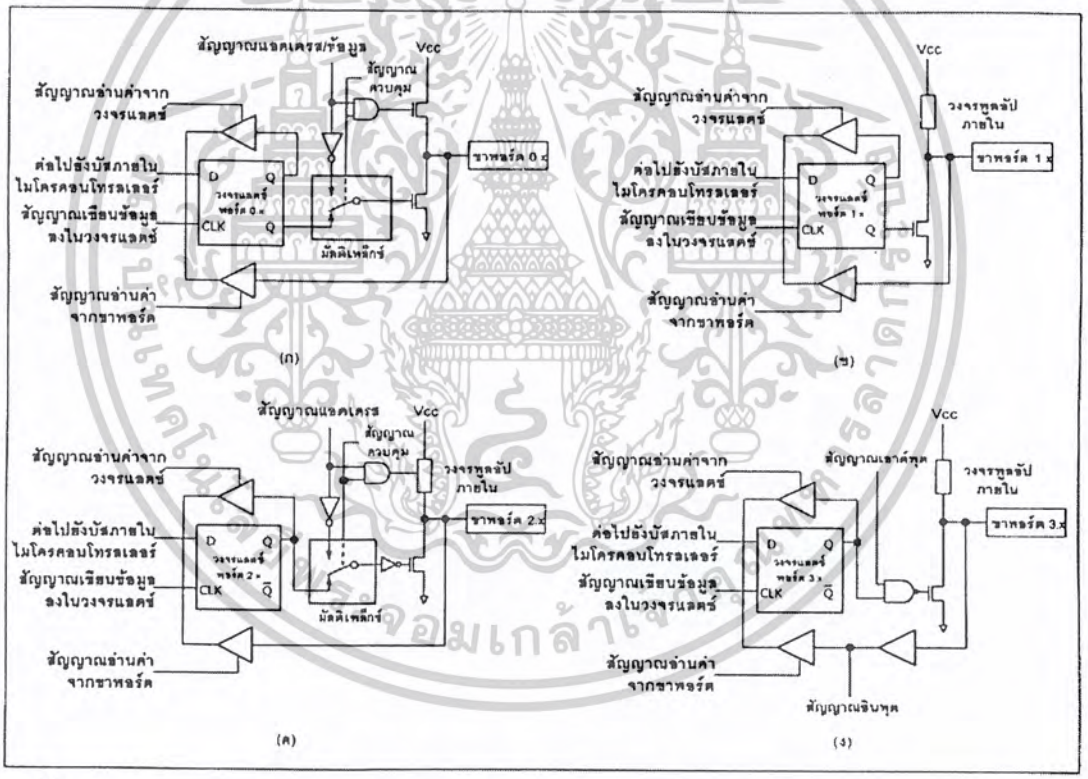
ขา	เบอร์ของไมโครคอนโทรลเลอร์	หน้าที่พิเศษ
P1.0	AT89C52/AT89SXX	ขา T2 เป็นขาอินพุตนับค่าของไทเมอร์/เคาน์เตอร์ 2 และเป็นขาเอาต์พุตของการกำเนิดสัญญาณนาฬิกาโดยไทเมอร์ 2 (clock out)
P1.1	AT89C52/AT89SXX	ขา T2EX เป็นขาอินพุตทริกเกอร์สำหรับการแคปเจอร์/รีโหลดและควบคุมทิศทางของสัญญาณ
P1.4	AT89SXX	ขา $\overline{SS}$ (Slave Select) เป็นขาเลือกการติดต่อในกรณีที่ไมโครคอนโทรลเลอร์เป็นอุปกรณ์สเลฟ ในระบบการติดต่อแบบ SPI
P1.5	AT89SXX	ขา MOSI (Master data output, Slave data input) ใช้ในการติดต่อกับพอร์ต SPI
P1.6	AT89SXX	ขา MOSI (Master data output, Slave data input) ใช้ในการติดต่อกับพอร์ต SPI
P1.7	AT89SXX	ขา SCK (Master clock output) เป็นขาสัญญาณนาฬิกาของการติดต่อกับพอร์ต SPI

ตารางที่ 2.2 หน้าที่พิเศษของพอร์ต 1 ในไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลช

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ที่พอร์ต 0 และพอร์ต 2 จะใช้งานเป็นพอร์ตอินพุตและเอาต์พุตสำหรับงานทั่วไป และใช้ในการติดต่อกับหน่วยความจำภายนอก สำหรับพอร์ต 3 ทั้งพอร์ตและพอร์ต 1 บางขานอกจากจะใช้เป็นขาพอร์ตอินพุตเอาต์พุตตามปกติแล้ว ยังสามารถใช้งานในหน้าที่พิเศษได้อีก ขึ้นอยู่กับว่าเป็นไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลชเบอร์ใด ดังสรุปได้ในตารางที่ 2.2

ในรูปที่ 2.5 แสดงวงจรภายในของแต่ละพอร์ตของไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลชโดยในรูปที่ 2.5 (ก) เป็นวงจรของพอร์ต 0 วงจรแลตช์ของแต่ละบิตในแต่ละพอร์ตก็คือวงจรดีฟลิปฟลอปนั่นเอง การอ่านค่าสถานะของพอร์ตและสถานะของวงจรแลตช์สามารถกระทำได้อย่างอิสระต่อกันด้วยสัญญาณที่แยกจากกัน นั่นคือ สัญญาณอ่านข้อมูลจากขาพอร์ต และสัญญาณอ่านข้อมูลจากวงจรแลตช์ส่วนการเขียนข้อมูลมายังพอร์ตต้องส่งสัญญาณมายังขา CLK ของดีฟลิปฟลอปในขณะที่ข้อมูลจะส่งผ่านมาทางขาบัสข้อมูลภายในเข้าสู่ขา D ของดีฟลิปฟลอป

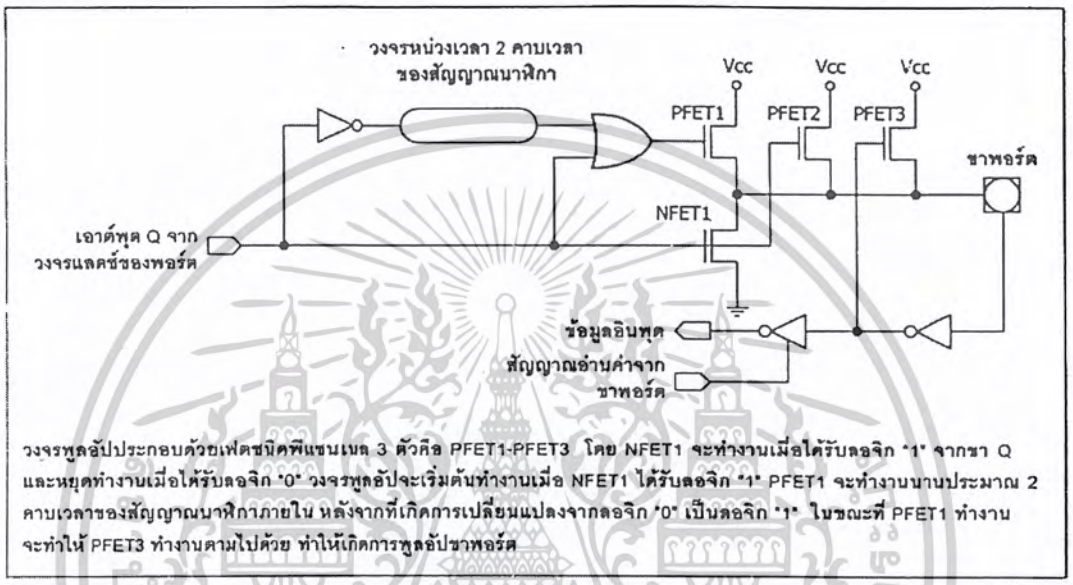


รูปที่ 2.5 วงจรภายในของพอร์ตทุกพอร์ตในไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลช

ในรูปที่ 2.5 (ข) เป็นวงจรของพอร์ต 1 ซึ่งมีลักษณะโดยทั่วไปคล้ายกับพอร์ต 0 หากแต่ไม่มีวงจรมัลติเพล็กซ์ เนื่องจากพอร์ตนี้จะไม่ใช้ในการติดต่อกับหน่วยความจำภายนอก แต่จะมีวงจรพูลอัปภายในที่แต่ละบิตของพอร์ตนี้แทน สำหรับรายละเอียดของวงจรพูลอัปแสดงในรูปที่ 2.6

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในรูปที่ 2.5 (ค) เป็นวงจรภายในของพอร์ต 2 จะคล้ายกับพอร์ต 0 มาก ต่างกันเพียงมีวงจรพูลอัปเพิ่มเติมเข้ามา ส่วนในรูปที่ 2.5 (ง) เป็นวงจรภายในของพอร์ต 3 จะเห็นได้ว่าคล้ายกับพอร์ต 1 มีการเพิ่มเติมวงจรบัฟเฟอร์ และวงจรอินพุตเอาต์พุตเมื่อทำงานในฟังก์ชั่นพิเศษเข้ามา เนื่องจากพอร์ต 3 สามารถนำไปใช้งานในหน้าที่พิเศษได้ทุกขา



รูปที่ 2.6 วงจรพูลอัปภายในพอร์ตของไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลช

การใช้งานเป็นพอร์ตอินพุต

เนื่องจากพอร์ตทั้งหมดของไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลชสามารถเป็นได้ทั้งอินพุตเอาต์พุต ดังนั้นจึงมีความจำเป็นอย่างยิ่งต้องทำความเข้าใจถึงการกำหนดลักษณะการทำงานให้แก่พอร์ตของไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลช

ในการกำหนดให้เป็นพอร์ตอินพุต ต้องเริ่มต้นด้วยการเขียนข้อมูล "1" มาที่แต่ละบิตของพอร์ตที่ต้องการใช้งานเป็นอินพุต เพื่อหยุดการทำงานของเฟตที่ใช้ในการขับสัญญาณเอาต์พุตของบิตนั้น ๆ ทำให้ขาสัญญาณของพอร์ตเชื่อมต่อกับวงจรพูลอัปภายในโดยตรง ส่งผลให้ขาพอร์ตนั้นมีลอจิกเป็น "1" สามารถรับสัญญาณลอจิก "0" จากอุปกรณ์ภายนอกได้ง่าย สัญญาณข้อมูลจากอุปกรณ์ภายนอกจะถูกส่งเข้ามาเก็บไว้ในวงจรบัฟเฟอร์ภายในพอร์ต แล้วรอให้ซีพียูมาอ่านค่าเข้าไปเมื่อเป็นเช่นนี้ อุปกรณ์ภายนอกที่เชื่อมต่อกับพอร์ตอินพุตของไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลชควรกำหนดให้ทำงานในสภาวะลอจิก "0" จะดีและสะดวกที่สุด (ซึ่งในปัจจุบันอุปกรณ์อินพุตที่เชื่อมต่อไมโครคอนโทรลเลอร์แทบทั้งหมดทำงานที่ลอจิก "0" แล้ว)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การใช้งานเป็นพอร์ตเอาต์พุต

โดยปกติแล้ว ขาพอร์ตจะกำหนดให้มีลักษณะเป็นเอาต์พุตอยู่แล้ว ดังนั้นจึงสามารถส่งข้อมูลออกไปได้อย่างง่ายดายและตรงไปตรงมา กล่าวคือ เมื่อต้องการส่งข้อมูล "0" ออกไปทางเอาต์พุตก็ให้เขียนข้อมูล "0" ไปยังวงจรแลตช์ ซึ่งก็จะส่งผลต่อไปขับเฟต ทำให้เฟตทำงานที่ขาพอร์ตที่กำหนดให้ทำงานเกิดลอจิก "0" ขึ้น ในทางตรงกันข้ามหากต้องการส่งข้อมูล "1" ออกไป ก็ให้เขียนข้อมูล "1" ไปยังวงจรแลตช์ วงจรขับจะหยุดทำงาน ทำให้ที่ขาพอร์ตเชื่อมต่อกับวงจรพูลอัปภายในเกิดเป็นลอจิก "1" ที่ขาพอร์ตนั้น ซึ่งจะคล้ายกับการกำหนดให้เป็นขาอินพุตมาก เพียงแต่แตกต่างกันที่กระบวนการในการเคลื่อนย้ายข้อมูล โดยถ้าเป็นอินพุตจะมีสัญญาณมาอ่านข้อมูลที่บัฟเฟอร์ แต่ถ้าเป็นเอาต์พุตจะไม่มีการอ่านข้อมูลที่บัฟเฟอร์แต่อย่างใด เว้นแต่ในกรณีที่ต้องการตรวจสอบข้อมูลที่ส่งออกมาทางเอาต์พุต

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

การอ่านค่าลอจิกจากพอร์ต

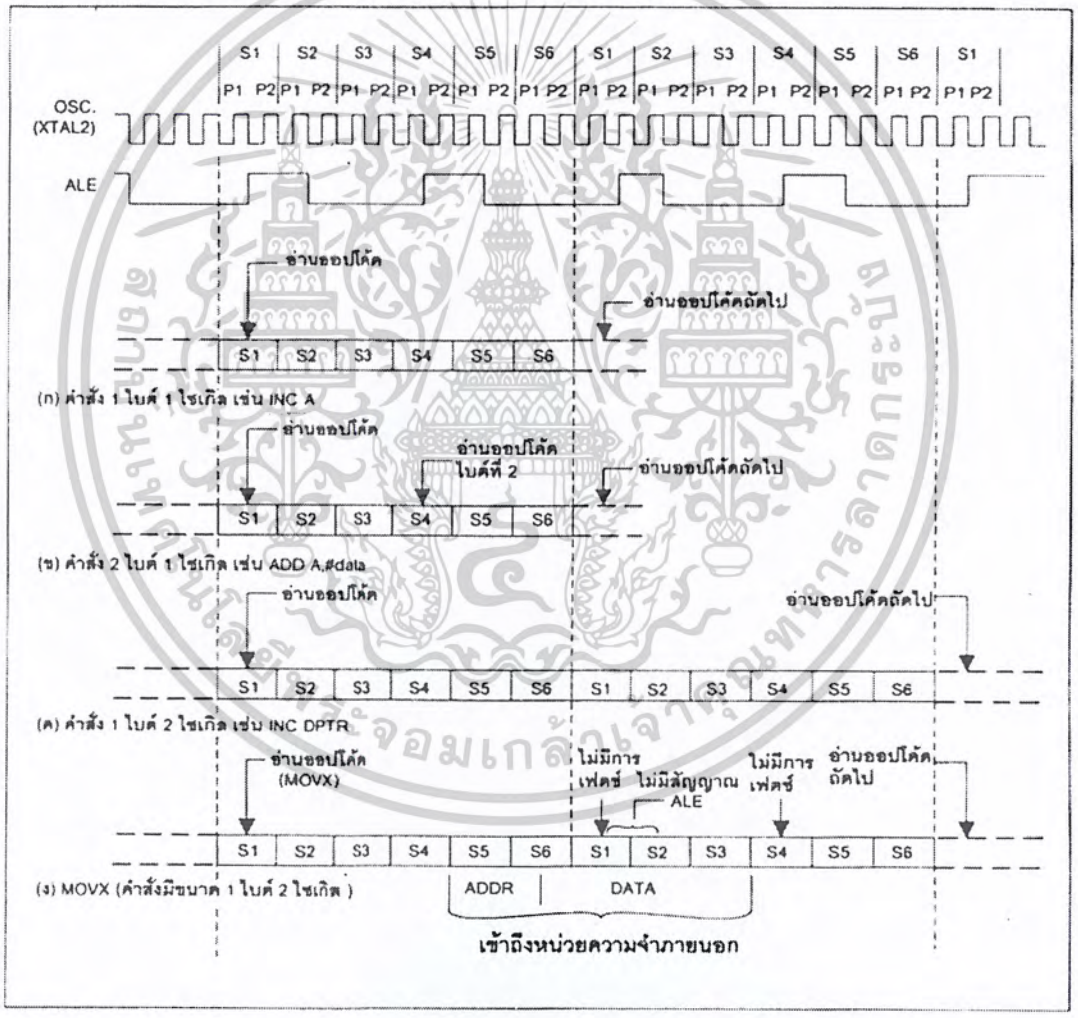
ในไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลชสามารถอ่านค่าลอจิกจากพอร์ตได้ 2 ลักษณะ คือ อ่านจากขาพอร์ต โดยตรง และอ่านจากวงจรแลตช์ของแต่ละพอร์ต

ในกรณีที่พอร์ตต่อกับขาเบสทรานซิสเตอร์ชนิด NPN และขาอิมิตเตอร์ของทรานซิสเตอร์ตัวนั้นต่อลงกราวด์ หากมีการส่งข้อมูล "1" ไปยังทรานซิสเตอร์ จะทำให้ทรานซิสเตอร์ทำงาน สถานะลอจิกที่ขาพอร์ตจะเป็น "0" เนื่องจากเมื่อทรานซิสเตอร์ทำงาน จะเสมือนว่าพอร์ตนั้นถูกต่อลงกราวด์ทำให้หากอ่านค่าลอจิกที่ขาพอร์ตจะได้ผลตรงข้ามกับที่ส่งออกมา แต่ถ้าหากทำงานอ่านค่าลอจิกที่วงจรแลตช์ จะได้ค่าที่ตรงกับค่าที่ต้องการส่งจริง ดังนั้น ในการอ่านค่าลอจิกพอร์ตจึงต้องเลือกวิธีการให้เหมาะสมกับอุปกรณ์ที่นำมาต่อด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จังหวะการทำงานของไมโครคอนโทรลเลอร์ MCS-51

ในการใช้งานไมโครคอนโทรลเลอร์ MCS-51 จะต้องทำความเข้าใจถึงจังหวะการทำงานของซีพียูและลำดับขั้นตอนการประมวลผลคำสั่ง ในการประมวลผลคำสั่งของของซีพียูจะมีขั้นตอนหลัก ๆ 2 ขั้นตอนคือ **กระบวนการเฟตช์ (fetch)** เป็นการเรียกคำสั่งออกจากหน่วยความจำโปรแกรมแล้วทำการแปลงรหัสคำสั่งนั้นเป็นภาษาเครื่องเพื่อเตรียมการประมวลผล ขั้นตอนต่อมาคือ **กระบวนการเอ็กซิคิวท์ (execute)** เป็นการกระทำตามคำสั่งที่กำหนดหรือตามที่เฟตช์ขึ้นมาโดยกระบวนการก่อนหน้านี้ เมื่อทำการเอ็กซิคิวท์คำสั่งเรียบร้อยแล้ว ก็จะไปเริ่มกระบวนการเฟตช์คำสั่งต่อไป



รูปที่ 2.7 ไซเกิลการทำงานของไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลช

เมื่อเริ่มจ่ายไฟให้แก่ไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลชจะเกิดการรีเซตในลักษณะที่เรียกว่า **เพาเวอร์ออนรีเซต (power-on reset)** ทำให้ซีพียูไปเริ่มต้นการทำงานที่แอดเดรส 0000H

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ของหน่วยความจำโปรแกรมจังหวะการทำงานของซีพียูจะเป็นไปตามรูปแบบ โดยได้รับการกำหนดมาจากกรอบการทำงานหรือแมชชีนไซเกิล (machinecycle) ในรูปที่ 2.7 เป็นไดอะแกรมเวลาแสดงจังหวะการทำงานของไมโครคอนโทรลเลอร์ MCS-51 โดยใน 1 รอบการทำงานหรือแมชชีนไซเกิลจะแบ่งย่อยออกเป็น 6 เสตท (state) กำหนดชื่อเป็น S1-S6 ในแต่ละสเตทมีค่าเวลาเท่ากับ 2 คาบเวลาของสัญญาณนาฬิกา ถ้าสัญญาณนาฬิกามีความถี่ 12 MHz จะมีคาบเวลาเท่ากับ 1 ms คาบเวลาทั้งสองภายในหนึ่งสเตตจะเรียกว่า เฟส 1 (phase1) และเฟส 2 (phase2)

ในรูปที่ 2.7 (ก) และ (ข) จะเป็นการเอ็กซิคิวต์คำสั่งที่ใช้เวลา 1 ไซเกิล เริ่มต้นที่สเตท 1 จะเป็นการอ่านค่าออปโค้ดอันเป็นกระบวนการแลตช์ค่าของออปโค้ดส่งไปให้รีจิสเตอร์คำสั่ง (Instruction Register : IR) การเฟตช์ครั้งที่สองจะเกิดขึ้นที่สเตท 4 ภายในแมชชีนไซเกิลเดียวกัน ในกรณีที่เป็นคำสั่งไบต์เดียว การเฟตช์ครั้งที่ 2 ภายในแมชชีนไซเกิลเดียวกันจะถูกตัดทิ้งไป ในคำสั่งที่มีใช้เวลา 1 ไซเกิลจะสิ้นสุดการทำงานลงในสเตท 6 ของแมชชีนไซเกิลเดียวกัน

ในกรณีที่คำสั่งใช้เวลา 2 ไซเกิล การทำงานของคำสั่งนั้นจะสิ้นสุดลงในสเตท 6 ของแมชชีนไซเกิลที่สอง ดังในไดอะแกรมรูปที่ 2.7 (ค) สำหรับในการกระทำคำสั่ง MOVX ซึ่งเป็นคำสั่งขนาด 1 ไบต์ 2 ไซเกิล จะไม่มีการเฟตช์เกิดขึ้นในไซเกิลที่สองของคำสั่ง MOVX นี้ เนื่องจากซีพียูจะไปทำการติดต่อกับหน่วยความจำภายนอกดังแสดงในไดอะแกรมรูปที่ 2.7 (ง) จะเห็นได้ว่า เวลาในการเอ็กซิคิวท์จะไม่ได้ขึ้นอยู่กับว่าการติดต่อหน่วยความจำโปรแกรมภายในหรือภายนอก

ในรูปที่ 2.7 แสดงสัญญาณและไดอะแกรมเวลาของการเข้าถึงหน่วยความจำโปรแกรมภายนอก โดยในรูปที่ 2.7 (ก) เป็นไดอะแกรมในขณะที่ยังไม่มีการกระทำคำสั่ง MOVX สัญญาณที่ขา ALE และ $\overline{PSEN}$ จะเกิดการแอกทีฟ 2 ครั้งภายในหนึ่งแมชชีนไซเกิล ในทุกครั้งที่ ALE เกิดการแอกตีฟที่พอร์ต 0 (P0) จะมีค่าของรีจิสเตอร์ PC ในไบต์ต่ำออกมา ในขณะที่พอร์ต 2 (P2) ก็จะมีค่าของ PC ในไบต์สูงเพื่อชี้ไปยังแอดเดรสต่อไปที่ต้องไปดำเนินการ สำหรับขา $\overline{PSEN}$ ก็จะเกิดการแอกตีฟเมื่อมีการติดต่อกับหน่วยความจำโปรแกรมภายนอกในกรณีที่กระทำคำสั่ง MOVX เพื่อเข้าถึงหน่วยความจำข้อมูลภายนอกที่ขา $\overline{PSEN}$ จะไม่เกิดการแอกตีฟ 2 ครั้งภายใน 1 แมชชีนไซเกิล เนื่องจากบัสแอดเดรสและบัสข้อมูลจะถูกใช้ในการติดต่อหน่วยความจำข้อมูลภายนอกแทน แต่สำหรับสัญญาณ ALE ยังคงแอกทีฟตามจังหวะการทำงานเหมือนเดิม

จากไดอะแกรมเวลาทั้งหมดสามารถสรุปได้ว่า ในการทำงาน 1 รอบหรือ 1 แมชชีนไซเกิล ซีพียูในไมโครคอนโทรลเลอร์ MCS-51 จะใช้เวลา 12 คาบเวลาของสัญญาณนาฬิกา นั่นคือ เวลาในการทำงาน 1 ไซเกิลมีค่าเท่ากับ 1 ms หรือมีความเร็วในการทำงานภายใน 1MHz ในกรณีที่ใช้ความถี่สัญญาณนาฬิกา 12 MHz ดังนั้นถ้าต้องการทราบความเร็วของการทำงานภายในของไมโครคอนโทรลเลอร์ MCS-51 สามารถหาได้จาก ค่าความถี่สัญญาณนาฬิกาหารด้วย 12 และถ้าต้องการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หาค่าเวลาของ 1 รอบการทำงานหรือ 1 แมชชีนไซเกิล สามารถทำได้โดยการหาส่วนกลับของความเร็วในการทำงานภายในของไมโครคอนโทรลเลอร์ MCS-51 สามารถสรุปเป็นสูตรทางคณิตศาสตร์ได้ดังนี้

ความเร็วในการทำงานภายในของไมโครคอนโทรลเลอร์เท่ากับ

ความถี่ของสัญญาณนาฬิกา (ค่าของคริสตอลที่ต่ออยู่ที่ขา XTAL1 และ XTAL2) / 12

เวลา 1 แมชชีนไซเกิล = 1 / ความเร็วในการทำงานภายในของไมโครคอนโทรลเลอร์

การจัดหน่วยความจำของไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลช

ในไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลชมีหน่วยความจำภายในหลัก ๆ อยู่ 2 ส่วน คือ หน่วยความจำโปรแกรมและหน่วยความจำข้อมูล ซึ่งก็มีขนาดและการจัดสรรแตกต่างกันไปในแต่ละเบอร์ซึ่งจะกล่าวถึงรายละเอียดของการจัดสรรหน่วยความจำภายในไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลช การเชื่อมต่อกับหน่วยความจำภายนอก และข้อมูลเบื้องต้นของรีจิสเตอร์ฟังก์ชันพิเศษที่ใช้ควบคุมการทำงานของไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลช

หน่วยความจำโปรแกรม (Program memory)

ในรูปที่ 2.8 แสดงการจัดหน่วยความจำโปรแกรมของไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลชในเบอร์ต่าง ๆ ที่นิยมใช้งาน อันประกอบด้วยเบอร์ AT89C51 และ AT89C52 จะเห็นได้ว่า ทั้งสองเบอร์สามารถติดต่อหน่วยความจำโปรแกรมได้สูงสุด 64 กิโลไบต์ โดยสามารถเลือกใช้หน่วยความจำโปรแกรมภายในอย่างเดียวหรือรวมกับภายนอกหรือเลือกใช้หน่วยความจำภายนอกอย่างเดียวก็ได้ ดังในรูปที่ 2.8 (ก) โดยภายใน AT89C51 จะมีหน่วยความจำโปรแกรมภายใน 4 กิโลไบต์ ในขณะที่ AT89C52 จะมีขนาด 8 กิโลไบต์

ในกรณีที่ใช้หน่วยความจำภายในและภายนอกรวมกัน หากใช้ AT89C51 ก็จะสามารถติดต่อกับหน่วยความจำภายนอกได้ 60 กิโลไบต์ และถ้าใช้เบอร์ AT89C52 จะสามารถติดต่อกับหน่วยความจำโปรแกรมภายนอกได้ 56 กิโลไบต์

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

หน่วยความจำโปรแกรมมีแอดเดรสเริ่มต้นที่ 0000H เมื่อซีพียูได้รับการรีเซตให้เริ่มต้นการทำงาน จะต้องมาเริ่มต้นที่แอดเดรส 0000H นี้เสมอ อย่างไรก็ตาม ในพื้นที่ของหน่วยความจำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรมไม่ว่าจะใช้งานภายในหรือภายนอกก็ตาม ต้องมีการสงวนพื้นที่บางตำแหน่งเอาไว้สำหรับการบริการอินเตอร์รัปต์ 6 ประเภท ประเภทละ 8 ไบต์ ประกอบด้วย

พื้นที่สำหรับบริการอินเตอร์รัปต์ 0 จากภายนอก กำหนดไว้ที่แอดเดรส 0003H

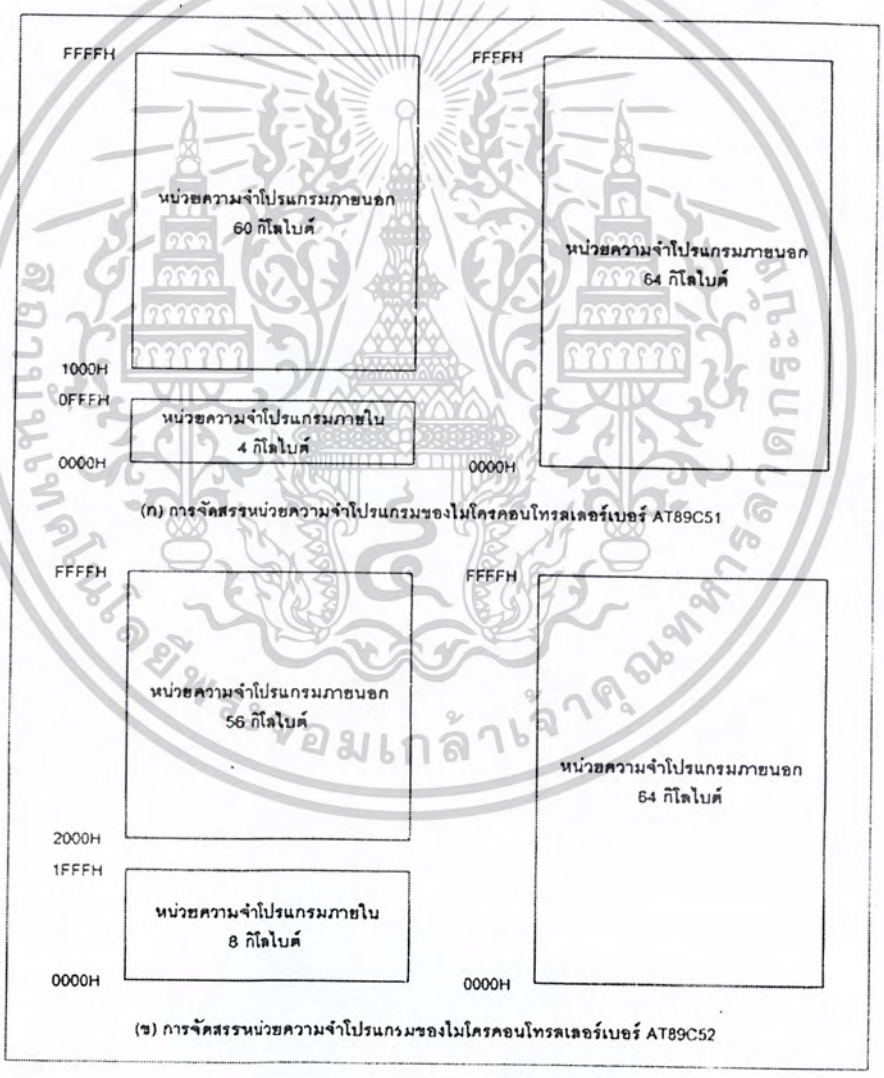
พื้นที่สำหรับบริการอินเตอร์รัปต์ 0 จากไทเมอร์ 0 กำหนดไว้ที่แอดเดรส 000BH

พื้นที่สำหรับบริการอินเตอร์รัปต์ 1 จากภายนอก กำหนดไว้ที่แอดเดรส 0013H

พื้นที่สำหรับบริการอินเตอร์รัปต์ 0 จากไทเมอร์ 1 กำหนดไว้ที่แอดเดรส 001BH

พื้นที่สำหรับบริการอินเตอร์รัปต์ของการสื่อสารอนุกรม กำหนดไว้ที่แอดเดรส 0023H

พื้นที่สำหรับบริการอินเตอร์รัปต์จากไทเมอร์ 2 กำหนดไว้ที่แอดเดรส 002BH



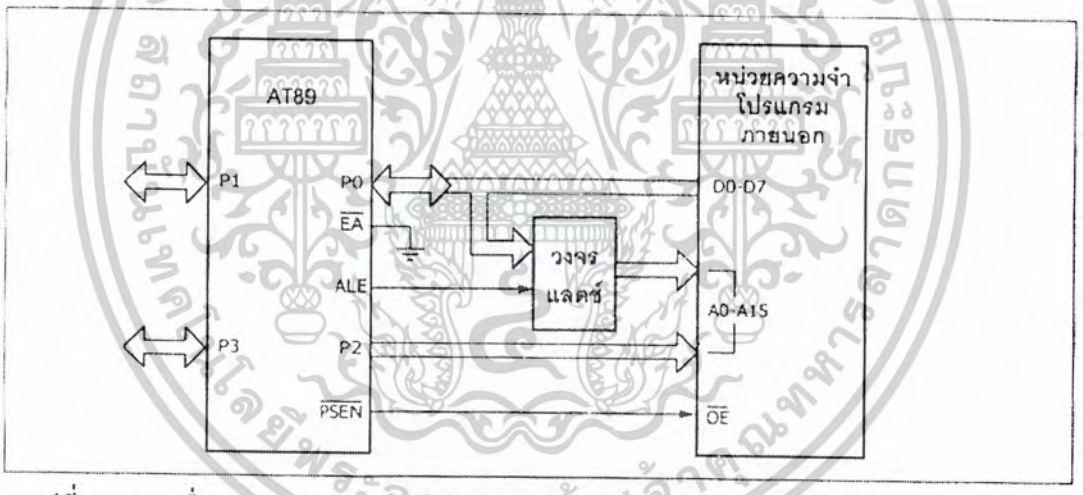
รูปที่ 2.8 การจัดสรรหน่วยความจำโปรแกรมของไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลช

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

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

การต่อหน่วยความจำภายนอกแสดงดังในรูปที่ 2.9 จะเห็นได้ว่าขาพอร์ต P0.0-P0.7 ถูกใช้เป็นขาข้อมูล D0-D7 และขาแอดเดรสไบต์ต่ำ โดยผ่านวงจรแลทช์ ซึ่งปกติใช้ไอซีเบอร์ 74HC573 และใช้สัญญาณ ALE และ $\overline{PSEN}$ ในการเลือกว่า ต้องการใช้งานขา P0.0-P0.7 เพื่อเป็นขาข้อมูลหรือขาแอดเดรสในขณะที่ขา P2.0-P2.7 ใช้ในการเชื่อมต่อกับขาแอดเดรสไบต์สูง A8-A15 ดังนั้นเมื่อมีการติดต่อกับหน่วยความจำโปรแกรมภายนอกไมโครคอนโทรลเลอร์จะเหลือขาพอร์ตใช้งานเพียง 16 บิต คือที่ขาพอร์ต P1.0-P1.7 และ P3.0-P3.7



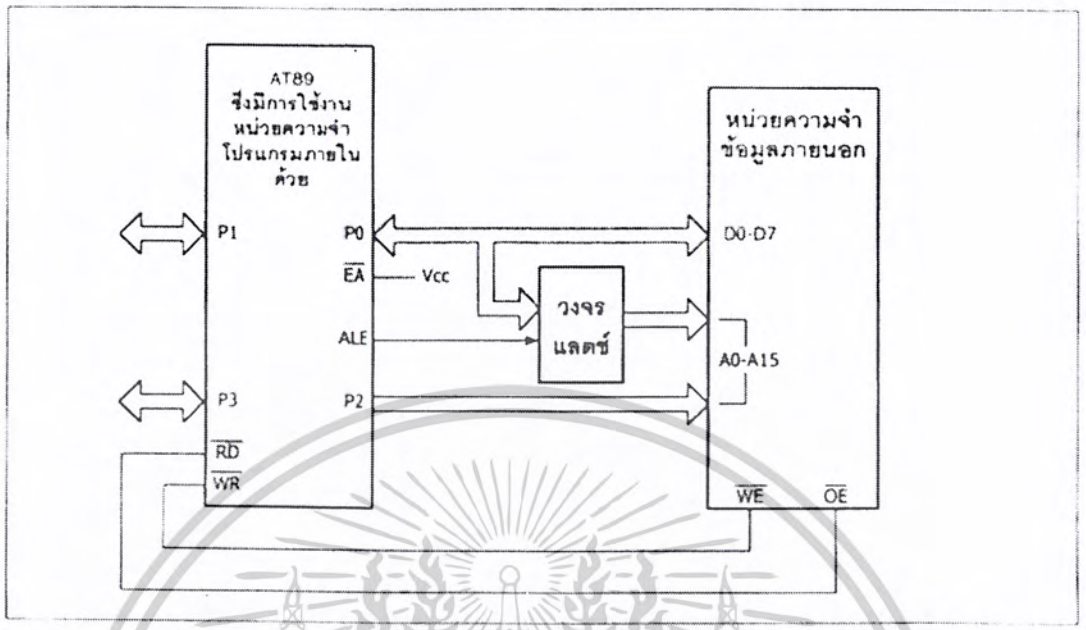
รูปที่ 2.9 การเชื่อมต่อหน่วยความจำโปรแกรมภายนอกของไมโครคอนโทรลเลอร์ MCS-51

หน่วยความจำข้อมูล (Data Memory)

มีด้วยกัน 2 แบบคือ หน่วยความจำข้อมูลภายนอกและภายในโดยไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลชในอนุกรม AT89 สามารถติดต่อกับหน่วยความจำข้อมูลภายนอกได้สูงสุด 64 กิโลไบต์โดยการใช้คำสั่ง MOVX ในการติดต่อกับหน่วยความจำข้อมูลภายนอก การติดต่อหน่วยความจำข้อมูลภายนอกของไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลชแสดงดังในรูปที่ 2.10 จะเห็นได้ว่า มีลักษณะคล้ายกับการติดต่อกับหน่วยความจำโปรแกรมภายนอกแตกต่างกันที่มีสัญญาณที่ใช้สำหรับการอ่านและเขียนหน่วยความจำข้อมูลภายนอก นั่นคือ ขา $\overline{RD}$ และ $\overline{WR}$

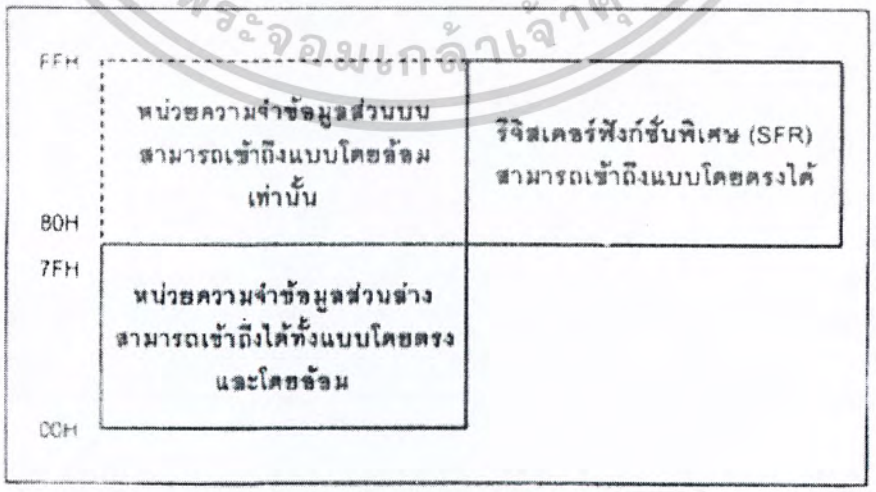
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

50186



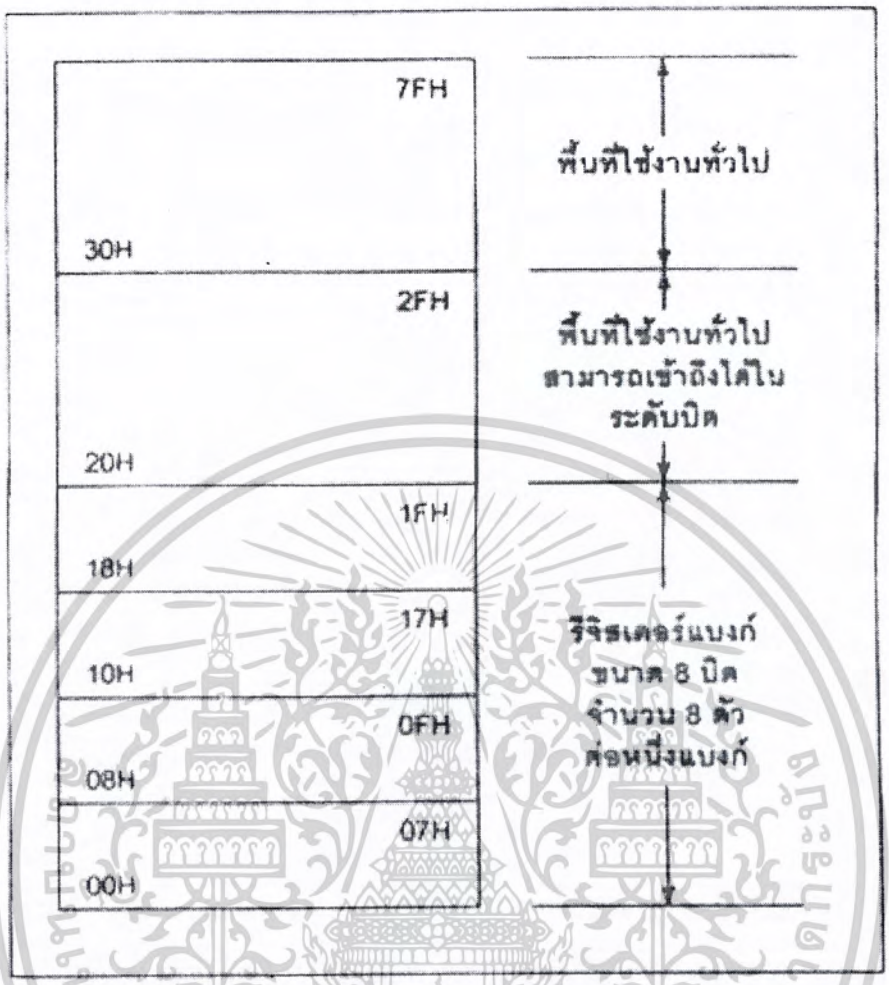
รูปที่ 2.10 การเชื่อมต่อหน่วยความจำข้อมูลภายนอกของไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลช

สำหรับไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลชในอนุกรม AT89 ทุกเบอร์จะมีหน่วยความจำข้อมูลภายในเป็นแบบแรม (RAM : Random Access Memory) โดยแต่ละเบอร์จะมีขนาดแตกต่างกันไป ในเบอร์ AT89C51 มีหน่วยความจำข้อมูลภายในขนาด 128 ไบต์ ในขณะที่เบอร์ AT89C52 มีขนาด 256 ไบต์ สำหรับการจัดสรรหน่วยความจำข้อมูลภายในแบ่งเป็น 3 ส่วน คือ หน่วยความจำข้อมูลส่วนกลาง (lower), ส่วนบน (upper) และรีจิสเตอร์ฟังก์ชั่นพิเศษ (SFR : Special Function Register) แต่ละส่วนมีขนาด 128 ไบต์ ดังแสดงการจัดสรรในรูปที่ 2.11



รูปที่ 2.11 การจัดสรรพื้นที่ของหน่วยความจำข้อมูลภายในไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลช

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.12 การจัดสรรพื้นที่ของหน่วยความจำข้อมูลภายในส่วนล่างของไมโครคอนโทรลเลอร์MCS-51 แบบแฟลช

จะเห็นได้ว่า หน่วยความจำข้อมูลส่วนบนและรีจิสเตอร์ฟังก์ชั่นพิเศษมีตำแหน่งทับซ้อนกัน แต่จะใช้การติดต่อที่แตกต่างกัน และในไมโครคอนโทรลเลอร์ MCS-51 บางเบอร์จะไม่มีหน่วยความจำข้อมูลส่วนบน

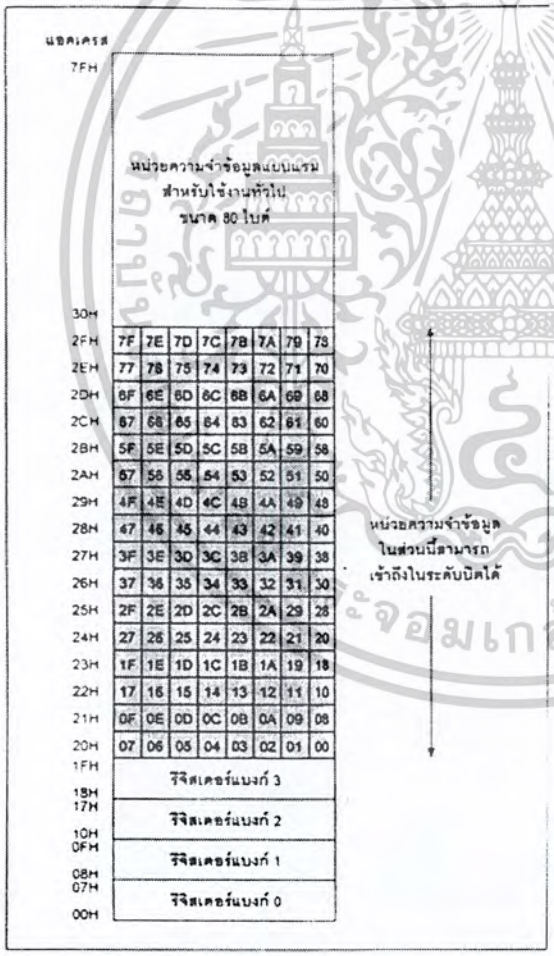
ขนาดของหน่วยความจำข้อมูลของไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลชโดยแท้จริงแล้วมีเพียง 256 ไบต์ แต่ด้วยการจัดเข้าถึงแตกต่างกัน จึงดูเสมือนว่า ไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลชมีหน่วยความจำข้อมูลภายในสูงถึง 384 ไบต์ โดยในหน่วยความจำข้อมูลส่วนล่างขนาด 128 ไบต์ มีแอดเดรสอยู่ที่ 00H-7FH สามารถเข้าถึงได้โดยตรงและโดยอ้อม สำหรับหน่วยความจำข้อมูลส่วนบนมีขนาด 128 ไบต์เช่นกัน มีแอดเดรสอยู่ที่ 80H-FFH สามารถเข้าถึงแบบโดยอ้อมเท่านั้น ในขณะที่รีจิสเตอร์ SFR มีแอดเดรสอยู่ที่ 80H-FFH เช่นเดียวกับหน่วยความจำข้อมูลส่วนบนแต่สำหรับรีจิสเตอร์ SFR ใช้การเข้าถึงแบบโดยตรง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

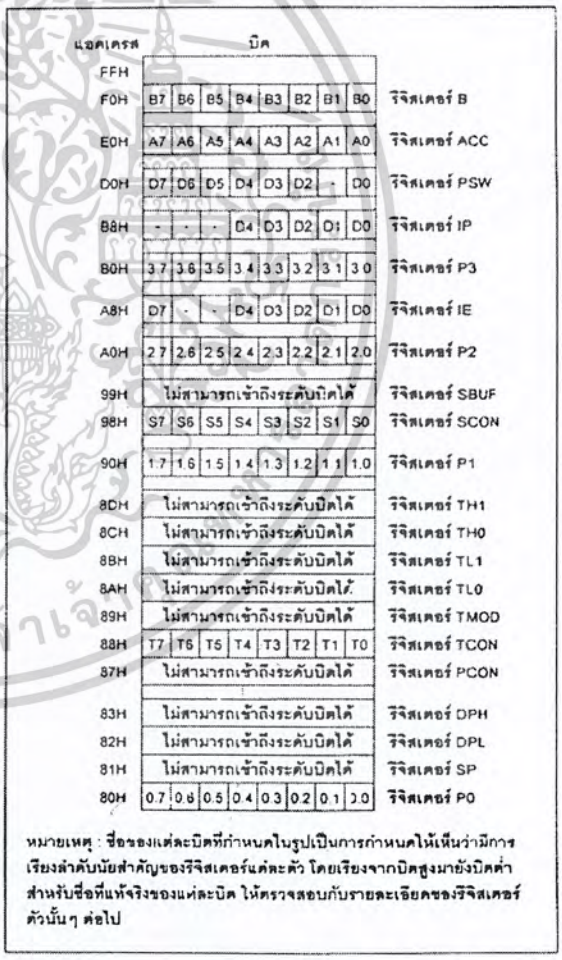
ดังนั้นเพื่อความสะดวกและง่ายตลอดจนป้องกันความสับสนในการเขียนโปรแกรมสำหรับผู้เริ่มต้นจึงควรใช้หน่วยความจำข้อมูลภายในเพียง 128 ไบต์จากหน่วยความจำข้อมูลส่วนล่างร่วมกับรีจิสเตอร์ SFR

ในรูปที่ 2.12 แสดงการจัดสรรหน่วยความจำข้อมูลส่วนล่าง หน่วยความจำ 32 ไบต์ต่ำสุดที่แอดเดรส 00H-1FH แบ่งเป็น 4 กลุ่ม เรียกว่า 4 แบงก์ (bank) แต่ละแบงก์มีรีจิสเตอร์ 8 ตัวคือ R0-R7 การติดต่อกับหน่วยความจำในแบงก์ใดให้กำหนดที่รีจิสเตอร์ PSW (Program Status Word register)

หน่วยความจำข้อมูล 16 ไบต์ถัดมาที่แอดเดรส 20H-2FH เป็นพื้นที่สำหรับใช้งานทั่วไปสามารถเข้าถึงได้ในระดับบิต (Bit addressable) และหน่วยความจำข้อมูลที่เหลือ 81 ไบต์จะต้องแบ่งส่วนหนึ่งสำรองไว้เป็นพื้นที่ขอสแต็ก (stack : ที่พักข้อมูลชั่วคราวในกรณีที่ซีพียูมีการกระโดดไปงานในโปรแกรมย่อย) การเข้าถึงหน่วยความจำส่วนนี้ต้องใช้การเข้าถึงในระดับไบต์



รูปที่ 2.13 โครงสร้างของหน่วยความจำข้อมูลภายในในส่วนบนของไมโครคอนโทรลเลอร์ MCS-51



รูปที่ 2.14 การจัดสรรพื้นที่ของรีจิสเตอร์ฟังก์ชั่นพิเศษ (SFR)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในรูปที่ 2.13 แสดงโครงสร้างของหน่วยความจำข้อมูลส่วนบนซึ่งจะมีลักษณะที่คล้ายกับหน่วยความจำข้อมูลส่วนล่าง หากแต่ใน 80 ไบต์บนไม่จำเป็นต้องสำรองไว้สำหรับสแต็ก และต้องใช้การเข้าถึงในลักษณะโดยอ้อมเท่านั้น

รีจิสเตอร์ฟังก์ชั่นพิเศษ (Special Function Register : SFR)

เป็นรีจิสเตอร์ที่ใช้ควบคุมการทำงานของไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลชทั้งหมดมีด้วยกัน 22 ตัว สำหรับในไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลชเบอร์ AT89C51 และ 28 ตัวในไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลชเบอร์ AT89C52 และอนุกรม AT89SXX ทั้งนี้เนื่องจากใน AT89C52 และ AT89SXX มีจำนวนไทเมอร์/เคาน์เตอร์มากกว่า AT89C51

รีจิสเตอร์ SFR มีแอดเดรสอยู่ระหว่าง 80H-FFH ในพื้นที่ของหน่วยความจำข้อมูลส่วนบน สามารถเข้าถึงได้โดยตรง (direct addressing) ในรูปที่ 2.14 แสดงการจัดสรรพื้นที่ของรีจิสเตอร์ SFR แต่ละตัวในหน่วยความจำข้อมูลส่วนบน สำหรับรายละเอียดเบื้องต้นของรีจิสเตอร์ SFR มีดังนี้

บิต 7	บิต 6	บิต 5	บิต 4	บิต 3	บิต 2	บิต 1	บิต 0
CY	AC	F0	RS1	RS0	OV	-	P

CY : แฟลกทด (Carry flag) เป็น "1" เมื่อมีการกระทำคำสั่งทางคณิตศาสตร์และลอจิกแล้วค่าของแอกคิวมูเลเตอร์เกิน 255 (ฐานสิบ) หรือ FFH

AC : แฟลกทดเสริม (Auxiliary Carry flag) เป็น "1" เมื่อมีการกระทำคำสั่งทางคณิตศาสตร์แล้วทำให้เกิดการทดข้ามจากบิต 3 มายังบิต 4 มักใช้ในการแปลงค่าเลขฐานสิบ (BCD operation)

F0 : แฟลกใช้งานทั่วไป เมื่อผู้เขียนโปรแกรมกำหนดค่าที่บิตนี้แล้ว ไม่ว่าจะกระทำคำสั่งใด ๆ ที่บิตนี้จะไม่มีการเปลี่ยนแปลง

RS1 : บิตเลือกรีจิสเตอร์แบงก์ (Register Select 1) ใช้งานร่วมกับบิต RS0 เพื่อเลือกแบงก์ของรีจิสเตอร์ R0-R7

RS0 : บิตเลือกรีจิสเตอร์แบงก์ (Register Select0) ใช้งานร่วมกับบิต RS1 เพื่อเลือกแบงก์ของรีจิสเตอร์ R0-R7

OV : บิตเกิน (Overflow) เป็น "1" เมื่อมีการกระทำคำสั่งทางคณิตศาสตร์และลอจิกแล้ว ทำให้เกิดการทดเข้ามาจากบิต 6 มายังบิต 7 ของแอกคิวมูเลเตอร์ หรือแอกคิวมูเลเตอร์มีค่า เกิน 127 (ฐานสิบ) นอกจากนั้นยังใช้เป็นการแสดงค่าลบด้วย

\- : บิตนี้ผู้ใช้งานสามารถกำหนดใช้งานได้อย่างอิสระ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

P : บิตพาริตี้ (Parity) ใช้ในการตรวจสอบจำนวนค่า "1" ภายในแอกคิวมูเลเตอร์ ถ้าหากในแอกคิวมูเลเตอร์มีจำนวนบิตที่เป็น"1"รวมกันเป็นเลขคู่ บิตนี้จะเป็น "0" รวมกันเป็นเลขคี่ บิตนี้จะเป็น "1"

RS1	RS0	แบงก์ของรีจิสเตอร์	ช่วงแอดเดรส
0	0	แบงก์ 0	00H-07H
0	1	แบงก์ 1	08H-0FH
1	0	แบงก์ 2	10H-17H
1	1	แบงก์ 3	18H-1FH

ตารางที่ 2.3 การเลือกแบงก์ของหน่วยความจำส่วนล่างเพื่อติดต่อกับรีจิสเตอร์แบงก์ R0-R7

รีจิสเตอร์แสดงสถานะของโปรแกรม (Program Status Word : PSW)

เป็นรีจิสเตอร์ขนาด 8 บิต สามารถเข้าถึงได้ในระดับบิต นั่นหมายความว่า สามารถกระทำคำสั่งหรือกำหนดค่าในแต่ละบิตของรีจิสเตอร์ตัวนี้ได้อย่างอิสระ มีแอดเดรสอยู่ที่ D0H เป็นรีจิสเตอร์ที่เก็บสถานะของการทำงานของโปรแกรมในขณะนั้น จะเรียกสถานะต่าง ๆ ของโปรแกรมว่า แฟลก (flag) เมื่อซีพียูกระทำคำสั่งทางคณิตศาสตร์และลอจิกแล้วเกิดการเปลี่ยนแปลงสถานะขึ้น ผลของการเปลี่ยนแปลงนั้นจะมาปรากฏที่บิตต่าง ๆ ของรีจิสเตอร์ PSW ดังแสดงข้างต้น

จะเห็นได้ว่า นอกจากรีจิสเตอร์ PSW ถูกใช้ในการเก็บสถานะของโปรแกรมแล้ว ที่บิต RS0 และ RS1 ยังใช้ในการเลือกแบงก์ของหน่วยความจำส่วนล่าง ซึ่งเป็นพื้นที่ของรีจิสเตอร์ R0-R7 ด้วย ดังมีรายละเอียดแสดงในตารางที่ 2.3 โดยปกติแล้วในการใช้งานรีจิสเตอร์ R0-R7 มักนิยมเลือกใช้แบงก์ 0 เป็นลำดับแรก หากไม่เพียงพอจึงเลือกในแบงก์อื่น ๆ มาใช้ แต่ต้องระมัดระวังในการกำหนดค่าและลำดับการติดต่อให้ดี มิเช่นนั้น อาจทำให้การเขียนโปรแกรมเกิดความสับสน จึงควรเลือกใช้รีจิสเตอร์ R0-R7 ในแบงก์ 0 เพียงแบงก์เดียวให้ชำนาญเสียก่อน

การกำหนดค่าของรีจิสเตอร์ PSW เพื่อเลือกใช้งานรีจิสเตอร์ R0-R7 ควรกำหนดไว้ที่ตอนต้นของโปรแกรมเสมอ เพื่อจะได้เขียนโปรแกรมติดต่อกับรีจิสเตอร์ R0-R7 ได้อย่างสะดวกและไม่เกิดความผิดพลาด

แอกคิวมูเลเตอร์ (Accumulator : ACC)

มีขนาด 8 บิตมีแอดเดรสอยู่ที่ตำแหน่ง E0H เป็นรีจิสเตอร์ที่ใช้สำหรับเก็บข้อมูลหรือผลลัพธ์ที่ได้จากการทำงานของไมโครคอนโทรลเลอร์ โดยเฉพาะอย่างยิ่งในการคำนวณทาง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คณิตศาสตร์และลอจิก ก่อนที่จะส่งข้อมูลหรือผลลัพธ์ที่ได้นั้นให้แก่ซีพียูเพื่อทำการประมวลผลต่อไป อาจเรียกรีจิสเตอร์แอกคิวมูเลเตอร์อย่างสั้น ๆ ว่า รีจิสเตอร์ A หรือ ACC

รีจิสเตอร์ A นี้สามารถเข้าถึงระดับบิตได้ นั่นหมายความว่า สามารถกระทำคำสั่งหรือกำหนดค่าในแต่ละบิตของรีจิสเตอร์ตัวนี้ได้อย่างอิสระ

รีจิสเตอร์ B

มีขนาด 8 บิต มีแอดเดรสอยู่ที่ F0H มีหน้าที่พิเศษคือ หากมีความต้องการคูณหรือหารทางคณิตศาสตร์ จะต้องนำข้อมูลที่ต้องการหารหรือคูณนั้น มาเก็บไว้ในรีจิสเตอร์ B นี้ แล้วจึงกระทำคำสั่งการคูณหรือหารกับค่าในรีจิสเตอร์ A ต่อไป

ในกรณีที่ไม่ได้มีความต้องการคูณหรือหารข้อมูล สามารถใช้รีจิสเตอร์ B นี้ในการเก็บข้อมูลทั่วไปได้ เหมือนกับรีจิสเตอร์ปกติ และสามารถเข้าถึงในระดับบิตได้เช่นเดียวกับรีจิสเตอร์ A

โปรแกรมเคาน์เตอร์ (Program Counter : PC)

มีขนาด 16 บิต มีหน้าที่แจ้งแอดเดรสของหน่วยความจำโปรแกรมในตำแหน่งถัดไปที่ซีพียูจะต้องไปทำงาน รีจิสเตอร์ PC เป็นรีจิสเตอร์ตัวเดียวที่ไม่ได้จัดสรรไว้รวมกับรีจิสเตอร์ SFR ตัวอื่นๆ การเปลี่ยนแปลงค่าของรีจิสเตอร์ PC จะขึ้นอยู่กับผลของการกระทำคำสั่งแต่ละคำสั่งภายในหน่วยความจำโปรแกรม

รีจิสเตอร์ PC มีความสำคัญมาก โดยเฉพาะอย่างยิ่งในการตรวจสอบการทำงานของโปรแกรมว่า ดำเนินไปตามลำดับขั้นตอนตามที่กำหนดไว้หรือไม่

สแต็กพอยน์เตอร์ (Stack Pointer : SP)

หรือรีจิสเตอร์ตัวชี้สแต็ก มีขนาด 8 บิต มีแอดเดรสอยู่ที่ 81H ใช้ในการเก็บค่าตำแหน่งของตัวชี้สแต็ก ซึ่งสามารถเปลี่ยนแปลงได้เมื่อซีพียูมีการกระโดดไปทำงานที่โปรแกรมย่อย หรือกระโดดโปรแกรมย่อยกลับมายังโปรแกรมหลัก เมื่อมีการรีเซตเกิดขึ้น (รีเซต : การกระทำที่ส่งผลให้ซีพียูต้องเริ่มต้นการทำงานใหม่ตั้งแต่ต้น) ค่าของรีจิสเตอร์ SP จะเท่ากับ 07H นั่นหมายความว่าตัวชี้สแต็กมีค่า 07H แอดเดรสแรกของพื้นที่ที่สำรองไว้ทำหน้าที่เป็นสแต็กจะเท่ากับ 08H

รีจิสเตอร์ชี้ข้อมูลหรือดาต้าพอยน์เตอร์ (Data Pointer : DTPR)

มีขนาด 16 บิต โดยแบ่งเป็นรีจิสเตอร์ชี้ข้อมูลไบต์สูง (DPH) และรีจิสเตอร์ชี้ข้อมูลไบต์ต่ำ (DPL) แต่ละตัวมีขนาด 8 บิต มีแอดเดรสอยู่ที่ 82H สำหรับ DPL และ 83H สำหรับ DPH รีจิสเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

DPTR นี้ใช้ในการเก็บค่าแอดเดรสของหน่วยความจำหรืออุปกรณ์ภายนอกที่ไมโครคอนโทรลเลอร์ต้องการติดต่อด้วย

รีจิสเตอร์พอร์ต (Port register)

เป็นรีจิสเตอร์ขนาด 8 บิต ที่ใช้เก็บข้อมูลของแต่ละพอร์ตของไมโครคอนโทรลเลอร์ MCS-51 มีด้วยกันทั้งสิ้น 4 ตัวคือ รีจิสเตอร์พอร์ต 0 หรือ P0 มีแอดเดรสอยู่ที่ 80H, รีจิสเตอร์พอร์ต 1 หรือ P1 มีแอดเดรสอยู่ที่ 90H, รีจิสเตอร์พอร์ต 2 หรือ P2 มีแอดเดรสอยู่ที่ A0H และรีจิสเตอร์พอร์ต 3 หรือ P3 มีแอดเดรสอยู่ที B0H รีจิสเตอร์ทุกตัวสามารถเข้าถึงได้ในระดับบิต เมื่อต้องการอ่านหรือเขียนข้อมูลออกไปยังพอร์ตของไมโครคอนโทรลเลอร์ จะต้องกระทำผ่านรีจิสเตอร์นี้ทุกครั้ง

รีจิสเตอร์บัฟเฟอร์ข้อมูลอนุกรม (Serial Data Buffer : SBUF)

เป็นรีจิสเตอร์ขนาด 8 บิต มีแอดเดรสอยู่ที่ 99H ใช้ในการเก็บข้อมูลที่ทำการส่งออกหรือรับเข้าของวงจรสื่อสารอนุกรมที่มีอยู่ในไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลชโดยภายในรีจิสเตอร์ SBUF นี้จะแบ่งออกเป็น 2 ส่วนคือ รีจิสเตอร์บัฟเฟอร์สำหรับส่งข้อมูล (transmit buffer register) และรีจิสเตอร์บัฟเฟอร์สำหรับรับข้อมูล (receive buffer register) เมื่อมีการเขียนข้อมูลมายังรีจิสเตอร์ SBUF ข้อมูลนั้นจะถูกส่งต่อไปยังบัฟเฟอร์สำหรับส่งข้อมูล เพื่อส่งออกจากไมโครคอนโทรลเลอร์ผ่านทางขา TxD หรือขา P3.1 ในกรณีที่มีการอ่านข้อมูลจากรีจิสเตอร์ SBUF ข้อมูลจะถูกส่งผ่านไปยังรีจิสเตอร์บัฟเฟอร์สำหรับรับข้อมูลเพื่อส่งต่อไปยังไมโครคอนโทรลเลอร์ต่อไป สำหรับการรับข้อมูลอนุกรมจากภายนอกนั้นจะผ่านมาทางขา RxD หรือ P3.0 ของไมโครคอนโทรลเลอร์ MCS-51

รีจิสเตอร์ไทเมอร์ (Timer register)

เป็นรีจิสเตอร์ขนาด 16 บิต แต่จะจัดแบ่งเป็นไบต์สูงและไบต์ต่ำ เช่นเดียวกับรีจิสเตอร์ DPTR รีจิสเตอร์ไทเมอร์ใช้ในการเก็บค่าของตัวนับหรือเคาน์เตอร์ (counter) ภายในไมโครคอนโทรลเลอร์เพื่อใช้ในการสร้างฐานเวลา, จับเวลา หรือนับจำนวนพัลส์สัญญาณนาฬิกาภายในบางทีเรียกรีจิสเตอร์ตัวนี้ว่า รีจิสเตอร์ไทเมอร์/เคาน์เตอร์

ในไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลชเบอร์ AT89C51 จะมีรีจิสเตอร์ไทเมอร์/เคาน์เตอร์ 2 ตัว แบ่งเป็น T0 หรือ Timer 0 และ T1 หรือ Timer 1 ในรีจิสเตอร์ยังแบ่งเป็นรีจิสเตอร์ไทเมอร์ไบต์ต่ำ (TL) และรีจิสเตอร์ไทเมอร์ไบต์สูง (TH) เหมือนกัน โดยรีจิสเตอร์ TL0 มีแอดเดรส

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อยู่ที่ 8AH รีจิสเตอร์ TH0 มีแอดเดรสอยู่ที่ 8BH ในขณะที่ TL1 และ TH1 มีแอดเดรสอยู่ที่ 8CH และ 8DH ตามลำดับ

สำหรับในไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลชเบอร์ AT89C52 และในอนุกรม AT89SXX จะมีรีจิสเตอร์ไทเมอร์/เคาน์เตอร์ถึง 3 ตัว โดยมีรีจิสเตอร์ TL2 และ TH2 ซึ่งมีแอดเดรสอยู่ที่ CCH และ CDH ตามลำดับเพิ่มเติมเข้ามา

รีจิสเตอร์แคปเจอร์ (Capture register)

เป็นรีจิสเตอร์ขนาด 16 บิต มีเฉพาะในไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลชเบอร์ AT89C52 และในอนุกรม AT89SXX เท่านั้น เนื่องจากต้องใช้ร่วมกับไทเมอร์/เคาน์เตอร์2 (Timer2) ซึ่งมีอยู่ในไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลชเบอร์ AT89C52 และในอนุกรม AT89SXX โดยรีจิสเตอร์แคปเจอร์นี้มีชื่อเรียกอย่างย่อว่า รีจิสเตอร์ RCAP2 ซึ่งแบ่งออกเป็นไบต์ต่ำคือ RCAP2L มีแอดเดรสอยู่ที่ CAH และไบต์สูงคือ RCAP2H มีแอดเดรสอยู่ที่ CBH

รีจิสเตอร์แคปเจอร์จะถูกใช้งานเมื่อกำหนดให้ไทเมอร์ 2 ทำงานในโหมดแคปเจอร์ ซึ่งเป็นโหมดที่กำหนดให้ไมโครคอนโทรลเลอร์ทำการตรวจจับการเปลี่ยนแปลงสถานะทางลอจิกที่ขา T2EX ทั้งนี้เพื่อใช้ประโยชน์ในการวัดคาบเวลา ความถี่ ตลอดจนการเปลี่ยนแปลงของสัญญาณพัลส์ที่ขา T2EX นี้

รีจิสเตอร์ควบคุม (Control register)

รีจิสเตอร์ SFR ที่ใช้การควบคุมการทำงานในส่วนต่าง ๆ ไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลช ยังมีอีกหลายตัวประกอบด้วย

รีจิสเตอร์ PCON เป็นรีจิสเตอร์ที่เกี่ยวข้องกับการกำหนดอัตราการรับส่งข้อมูลของวงจรสื่อสารอนุกรมและกำหนดการทำงานในโหมดประหยัดพลังงานของไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลช

รีจิสเตอร์ SCON เป็นรีจิสเตอร์ที่ใช้ในการควบคุมการทำงานของวงจรสื่อสารอนุกรมภายในไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลช

รีจิสเตอร์ TCON และ T2CON เป็นรีจิสเตอร์ที่ใช้ในการควบคุมการทำงานของไทเมอร์/เคาน์เตอร์ภายในไมโครคอนโทรลเลอร์ MCS51 แบบแฟลช โดย T2CON ใช้สำหรับไทเมอร์/เคาน์เตอร์ 2 ของไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลชเบอร์ AT789C52

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รีจิสเตอร์ TMOD และ T2MOD เป็นรีจิสเตอร์ที่ใช้กำหนดโหมดหรือลักษณะในการทำงานของไทเมอร์/เคาน์เตอร์ภายในไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลช โดย T2MOD ใช้สำหรับไทเมอร์/เคาน์เตอร์2 ของไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลชเบอร์ AT89C52

รีจิสเตอร์ IE และ IP เป็นรีจิสเตอร์ที่เกี่ยวข้องกับการตอบสนองการอินเตอร์รัปต์ (interrupt : การขัดจังหวะการทำงานปกติของซีพียู) โดย IE เป็นรีจิสเตอร์สำหรับอีนาเบิลหรือใช้ในการกำหนดลักษณะของการตอบสนองการอินเตอร์รัปต์ ในขณะที่ IP เป็นรีจิสเตอร์สำหรับกำหนดลำดับความสำคัญของการตอบสนองการอินเตอร์รัปต์ว่า จะให้ซีพียูตอบสนองการเกิดอินเตอร์รัปต์ในลักษณะใดก่อนหรือหลัง

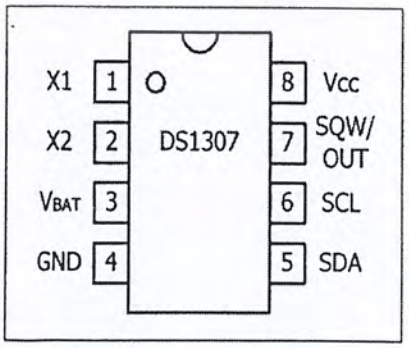
2.2 ไอซีสร้างฐานเวลาจริงเบอร์ DS1307 (RTC : Real Time Clock)

การทำงานคือจะทำหน้าที่สร้างฐานเวลาจริงให้แก่ระบบไมโครคอนโทรลเลอร์ โดย DS1307 จะให้ข้อมูลเกี่ยวกับเวลาทั้งหมดไม่ว่าจะเป็นค่าของเวลาที่ละเอียดถึงหลักวินาที , นาที , ชั่วโมง , วันที่ (date) , วันในสัปดาห์ (day) , เดือน และปี โดยสามารถปรับวันเดือนปีให้ตรงตามปฏิทินได้อย่างถูกต้อง รวมถึงการกำหนดวันในปีอธิกสุรทินด้วย คุณสมบัติทางเทคนิคที่สำคัญมีดังนี้

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

รายละเอียดการต่อใช้งานของ DS1307

ในรูปที่ 2.15 แสดงการจัดขาของ DS1307 แต่ละขามีหน้าที่และการใช้งานดังนี้



รูปที่ 2.15 การจัดขาของไอซี DS1307

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

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

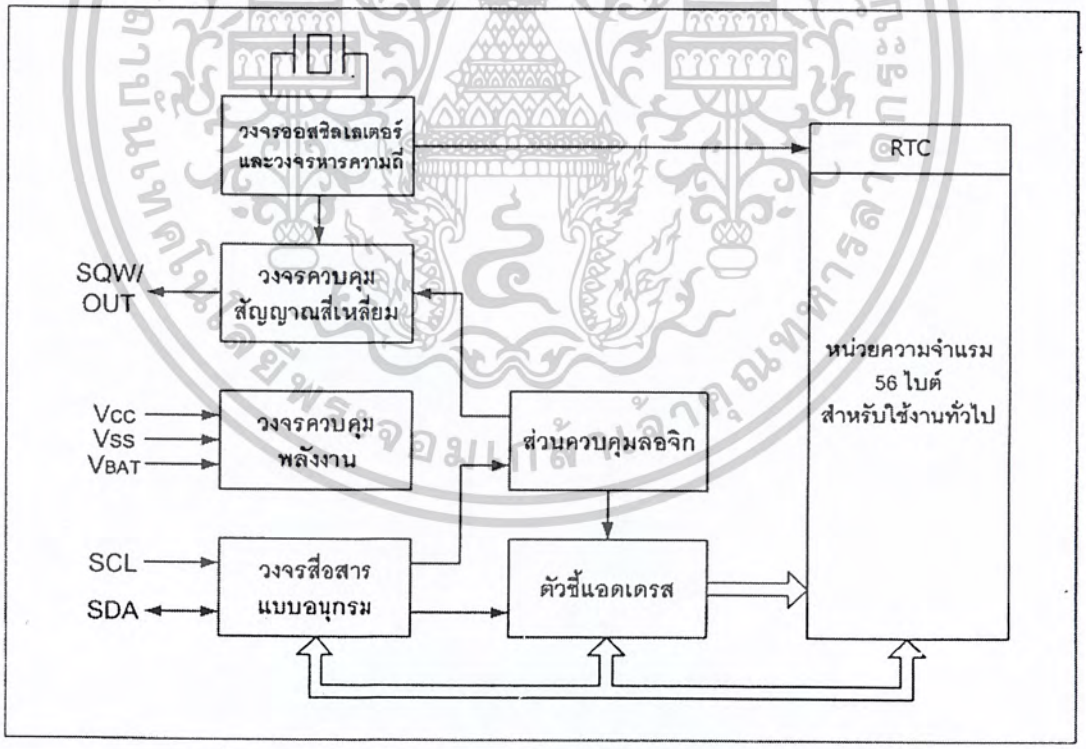
V_{BAT} (ขา 3) ใช้ต่อกับแบตเตอรี่ 3V เพื่อรักษาการทำงานของวงจรสร้างฐานเวลาของ DS1307 ให้คงอยู่ต่อไป แม้ว่าไม่มีไฟเลี้ยงจ่ายให้แก่ DS1307 ชนิดของแบตเตอรี่ที่เหมาะสมคือ แบตเตอรี่แบบลิเธียม ซึ่งมีความจุ 40 mAhr หรือมากกว่า จะสามารถรักษาข้อมูลได้นาน 10 ปีที่อุณหภูมิ 25 องศาเซลเซียส

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

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

X1 , X2 (ขา 1 , 2) ใช้ต่อกับคริสตอลความถี่มาตรฐาน 32.768 kHz เพื่อใช้เป็นฐานเวลาในการสร้างค่าเวลาจริง ในการใช้งานต้องต่อคริสตอลเข้ากับขาทั้งสองนี้และที่แต่ละขาต้องต่อตัวเก็บประจุค่าต่ำ ๆ ประมาณ 15 pF ร่อมกับขากราวด์ด้วย

การทำงานของ DS1307



รูปที่ 2.16 โครงสร้างภายในของไอซีรีลไทม์คล็อกเบอร์ DS1307

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ไอซี DS1307 จัดการเชื่อมต่อในแบบบัส I^2C โดยจะทำงานเป็นอุปกรณ์สเลฟเสมอ ดังนั้น การติดต่อเพื่อใช้งานจึงต้องกำหนดรูปแบบตามที่กำหนดไว้ในการติดต่อแบบ I^2C ในรูปที่ 2.16 แสดงส่วนประกอบหลักที่สำคัญและไดอะแกรมการทำงานของ DS1307 วงจรออสซิลเลเตอร์ถือเป็นหัวใจหลักของไอซี เนื่องจากเป็นจุดเริ่มต้นของการสร้างข้อมูลเวลาจริง ในขณะที่ DS1307 ทำงานที่ขา SQW/OUT จะมีสัญญาณพัลส์สี่เหลี่ยมส่งออกมาตลอดเวลาในกรณีที่มีการอีนาเบิลวงจรกำเนิดสัญญาณพัลส์ที่รีจิสเตอร์ควบคุม 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

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

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

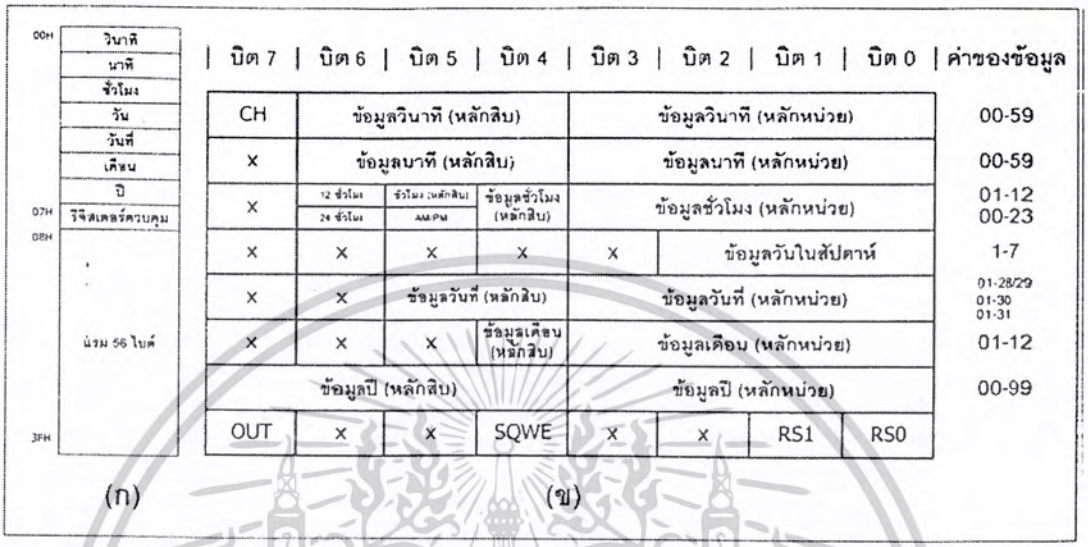
การจัดสรรหน่วยความจำใน DS1307

ในรูปที่ 2.17 (ก) แสดงการจัดสรรพื้นที่ของหน่วยความจำภายใน DS1307 พื้นที่ 7 ไบต์แรกตั้งแต่แอดเดรส 00H-06H เป็นพื้นที่ของรีจิสเตอร์ค่าเวลาใช้ในการเก็บข้อมูลเกี่ยวกับเวลาไบต์ต่อมาที่แอดเดรส 07H เป็นพื้นที่ของรีจิสเตอร์ควบคุมการทำงานของ DS1307 ในรูปที่ 2.17 (ข) แสดงรายละเอียดของรีจิสเตอร์ค่าเวลาและรีจิสเตอร์ควบคุมของ DS1307

ด้วยการจัดสรรพื้นที่แบบนี้ ทำให้ผู้ใช้งานสามารถเรียกข้อมูลเวลาออกมาได้ตามที่ต้องการโดยไม่จำเป็นต้องอ่านออกมาทั้งหมดก็ได้ ค่าของเวลาทั้งหมดจะอยู่ในรูปของเลขฐานสิบ สำหรับการแสดงเวลาในรูปของชั่วโมง สามารถเลือกได้ว่าต้องการแบบ 12 หรือ 24 ชั่วโมง โดยกำหนดที่บิต 6 ของแอดเดรส 02H และเมื่อเลือกแบบ 12 ชั่วโมง ที่บิต 5 ในแอดเดรสเดียวกันจะใช้ในการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แสดงค่า AM/PM โดยถ้าบิตนี้เป็น "1" หมายถึง ค่าชั่วโมงในขณะนี้เป็นช่วงเวลาหลังเที่ยงวัน ในกรณีที่เป็นแบบ 24 ชั่วโมง บิตนี้จะใช้ในการแสดงค่า 2 ของหลักสิบในหน่วยชั่วโมง



รูปที่ 2.17 (ก) การจัดสรรหน่วยความจำแรมภายใน DS1307

(ข) รายละเอียดของรีจิสเตอร์เก็บค่าเวลาและรีจิสเตอร์ควบคุมของ DS1307

รีจิสเตอร์ควบคุม

มีแอดเดรสอยู่ที่ 07H มีรายละเอียดของแต่ละบิตดังนี้

OUT (Output control) : ใช้ในการควบคุมระดับลอจิกที่ขา SQW/OUT ในกรณีที่ดิสเอเบิลการกำเนิดสัญญาณสี่เหลี่ยม โดยถ้าบิตนี้เป็น "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

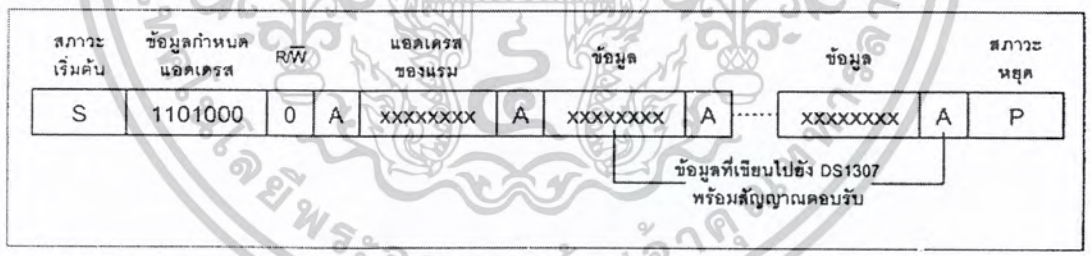
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โหมดการทำงานของ DS1307

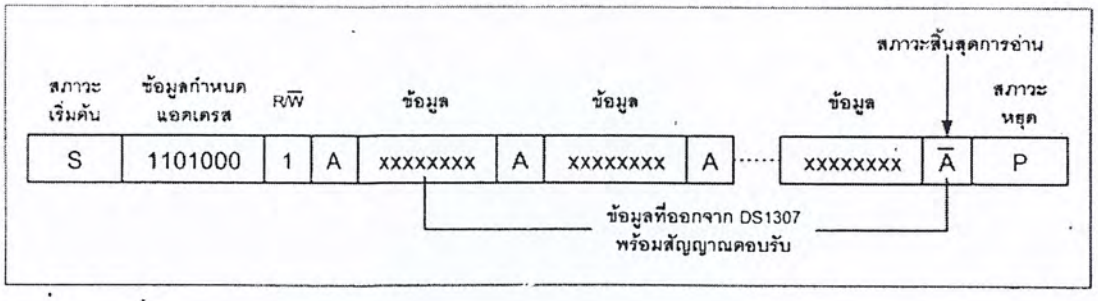
มีด้วยกัน 2 โหมด คือ โหมดเขียนข้อมูลและโหมดอ่านข้อมูล ในการใช้งาน DS1307 ตามปกติจะใช้งานเฉพาะโหมดอ่านข้อมูลเท่านั้น เนื่องจากไมโครคอนโทรลเลอร์จะติดต่อกับ DS1307 เพื่ออ่านข้อมูลของเวลาไปใช้งาน โหมดการเขียนข้อมูลจะถูกใช้งานก็ต่อเมื่อต้องการตั้งค่าเวลาใหม่และต้องการเขียนข้อมูลลงในหน่วยความจำใช้งานทั่วไปอย่างไรก็ตามเมื่อเริ่มต้นติดต่อกับ DS1307 จำเป็นอย่างยิ่งที่จะต้องเข้าสู่โหมดการเขียนข้อมูลก่อนเพื่อกำหนดแอดเดรสที่ต้องการอ่านข้อมูล จากนั้นจึงเปลี่ยนโหมดการทำงานมาเป็นโหมดการอ่านข้อมูลต่อไป

(1) โหมดการเขียนข้อมูล

มีรูปแบบดังในรูปที่ 2.18 เริ่มต้นเมื่อไมโครคอนโทรลเลอร์ทำการกำหนดสภาวะเริ่มต้น (START : S) จากนั้นส่งข้อมูลกำหนดแอดเดรส 1101000 ตามด้วยข้อมูลเลือกการเขียน นั่นคือค่า 0 จากนั้นจะรอการตอบรับจาก DS1307 ขั้นตอนต่อมา คือ ส่งข้อมูลเพื่อเลือกแอดเดรสที่ต้องการเขียน จากนั้นรอการตอบรับจาก DS1307 เมื่อมีการตอบรับมาเรียบร้อย ก็เริ่มทยอยเขียนข้อมูลลงไปครั้งละแอดเดรส หลังจากเขียนข้อมูลในแต่ละแอดเดรส จะต้องหยุดรอการตอบรับจาก DS1307 ทุกครั้ง จึงจะสามารถเขียนข้อมูลต่อไปได้ เมื่อเขียนเรียบร้อยแล้วให้ส่งสภาวะหยุด (STOP : P) เป็นอันสิ้นสุดกระบวนการเขียนข้อมูล



รูปที่ 2.18 รูปแบบของข้อมูลสำหรับติดต่อกับ DS1307 ในโหมดการเขียนข้อมูล



รูปที่ 2.19 รูปแบบของข้อมูลสำหรับติดต่อกับ DS1307 ในโหมดการอ่านข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

(2) โหมดการอ่านข้อมูล

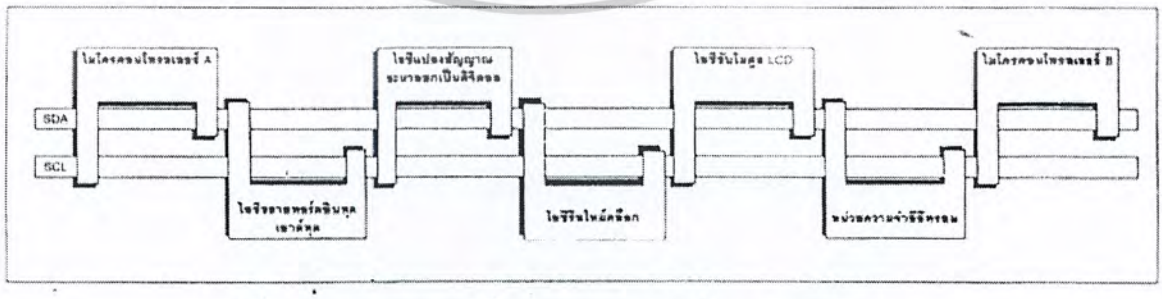
มีรูปแบบแสดงในรูปที่ 2.19 เริ่มต้นการทำงานเหมือนกับโหมดการเขียนข้อมูล คือ ไมโครคอนโทรลเลอร์กำหนดสภาวะเริ่มต้นแล้วส่งข้อมูลกำหนดแอดเดรสตามด้วยข้อมูลเลือกการอ่านซึ่งเท่ากับ 1 จากนั้นรอการตอบรับจาก DS1307 เมื่อตอบรับเรียบร้อย DS1307 จะทยอยส่งข้อมูลออกมาให้ไมโครคอนโทรลเลอร์คราวละ 1 แอดเดรสหรือ 1 ไบต์ โดยแอดเดรสที่เลือกอ่านข้อมูลจะต้องมีการกำหนดมาก่อนล่วงหน้าด้วยโหมดการเขียนข้อมูล วิธีการง่าย ๆ คือ เข้าสู่โหมดการเขียนข้อมูลก่อน เมื่อถึงจังหวะที่ต้องเขียนข้อมูลให้ทำการสร้างสภาวะเริ่มต้นและส่งข้อมูลกำหนดแอดเดรสใหม่อีกครั้ง ตามด้วยเลือกโหมดการอ่านข้อมูล ข้อมูลที่ออกมาจาก DS1307 ก็จะเป็นข้อมูลจากแอดเดรสที่กำหนดไว้ก่อนหน้านี้

2.3 หลักการของระบบบัส I^2C

ความรู้เบื้องต้นเกี่ยวกับ I^2C

I^2C ย่อมาจาก Inter-IC Communication หมายถึง การติดต่อสื่อสารระหว่างไอซีโดยบัส I^2C ได้รับการพัฒนาขึ้นโดยฟิลิปส์ (Philips) ด้วยจุดมุ่งหมายหลัก คือ ต้องการให้ไอซีหรือโมดูลสามารถติดต่อ สั่งงาน และควบคุมภายใต้สัญญาณเพียง 2 เส้น เส้นหนึ่ง คือ สายข้อมูล อีกเส้นหนึ่ง คือ สายสัญญาณนาฬิกาที่ใช้ในการกำหนดจังหวะการทำงาน การต่อร่วมกันของอุปกรณ์บนบัส I^2C ทำได้ง่ายมาก เพียงต่อสายข้อมูลและสายสัญญาณนาฬิกาของอุปกรณ์แต่ละตัวขนานหรือพ่วงกันไป ส่วนการกำหนดแอดเดรสหรือตำแหน่งสำหรับติดต่ออุปกรณ์แต่ละตัว จะใช้รหัสข้อมูลและการกำหนดสภาวะลอจิกที่ขาแอดเดรสของอุปกรณ์แต่ละตัว

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



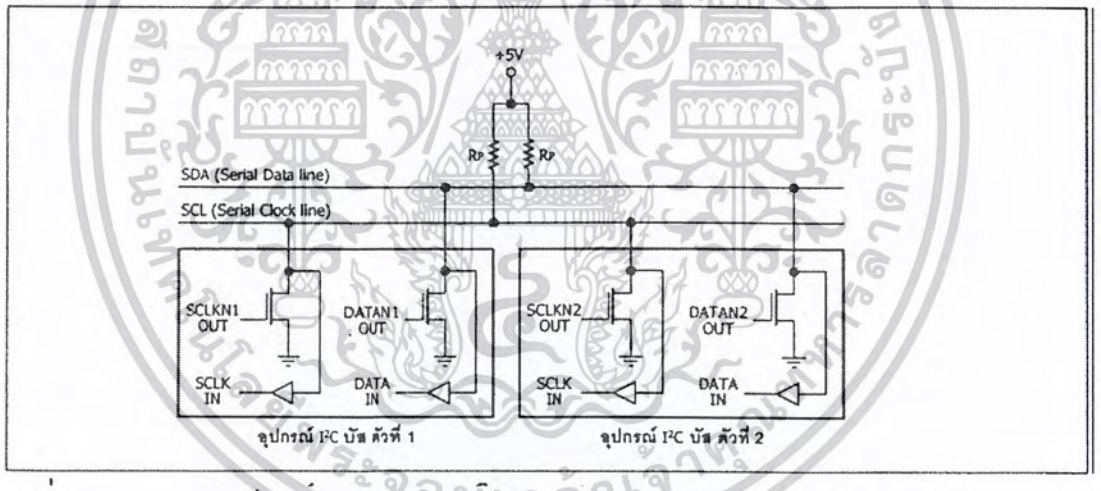
รูปที่ 2.20 ผังแสดงการเชื่อมต่อของอุปกรณ์ต่าง ๆ บนบัส I^2C

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในรูปที่ 2.20 แสดงผังของการเชื่อมต่ออุปกรณ์ต่าง ๆ บนบัส I^2C จะเห็นได้ว่า อุปกรณ์ที่ทำการเชื่อมต่อบนบัส I^2C มีหลากหลาย ไม่ว่าจะเป็นไอซีขยายพอร์ตอินพุตเอาต์พุต (I/O Expander), ไอซีแปลงสัญญาณอนาลอกเป็นดิจิตอล (ADC) และแปลงสัญญาณดิจิตอลเป็นอนาลอก (DAC), ไอซีรีลไทม์คล็อก (RTC), ไอซีขับโมดูล LCD, หน่วยความจำอีอีพรอม และไมโครคอนโทรลเลอร์

คุณสมบัติโดยทั่วไปของบัส I^2C

สาย SDA และ SCL เป็นสายสัญญาณ 2 ทิศทาง (bi-directional line) ต้องมีการต่อตัวต้านทานพูลอัพกับแรงดัน +5V ไว้ตลอดเวลาเพื่อให้สายมีสถานะลอจิกสูงในขณะที่ไม่มีการติดต่อใช้งานทั้งยังช่วยในการป้องกันสัญญาณรบกวนที่อาจมีเข้ามาในสายสัญญาณทั้งสอง วงจรเอาต์พุตของอุปกรณ์ที่ต่ออยู่บนบัส I^2C ต้องมีลักษณะเป็นวงจรเดรนเปิด (open-drain) หรือคอลเล็กเตอร์เปิด (open-collector) ดังแสดงรายละเอียดในรูปที่ 2.21



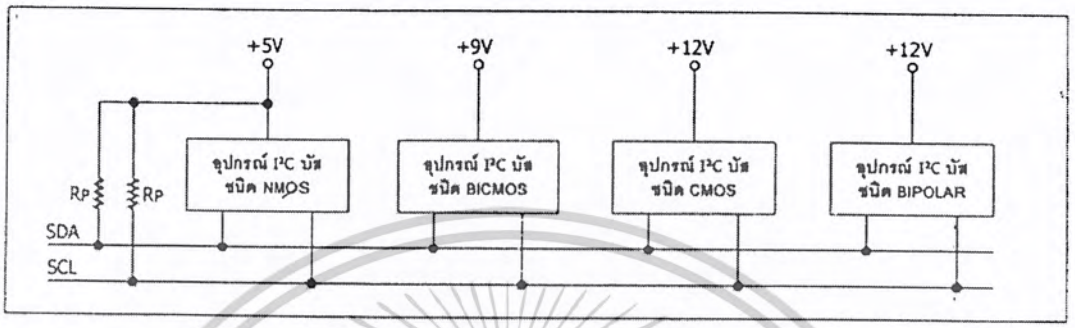
รูปที่ 2.21 แสดงวงจรเอาต์พุตของอุปกรณ์ในระบบบัส I^2C

อัตราการถ่ายทอดข้อมูลบนบัส I^2C สูงถึง 100 กิโลบิตต่อวินาทีในโหมดปกติ (standard mode) และสูงถึง 400 กิโลบิตต่อวินาทีในโหมดความเร็วสูง (fast mode) อุปกรณ์ที่ต่อร่วมอยู่บนบัส I^2C จะต้องมีค่าความจุไฟฟ้ารวมที่เกิดขึ้นระหว่างสาย SDA และ SCL ไม่เกิน 400 pF การเข้าถึงอุปกรณ์บนบัส 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 ในขณะที่อีกตัว

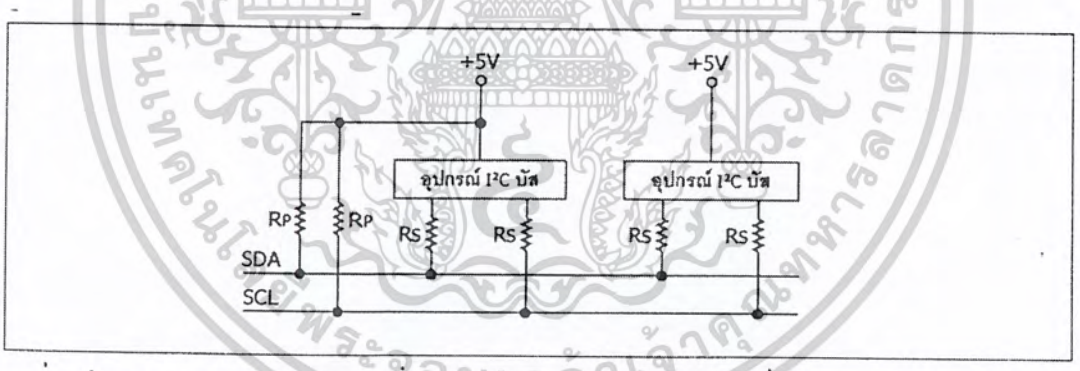
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หนึ่งใช้ไฟเลี้ยง +12V การต่อร่วมกันบนบัส I^2C สามารถกระทำได้ในลักษณะเดียวกับกรณีที่อุปกรณ์ทั้งสองใช้ไฟเลี้ยงเท่ากัน กล่าวคือ ให้ต่อสาย SDA และ SCL ของอุปกรณ์แต่ละตัวเข้าด้วยกัน และต้องต่อตัวต้านทานพูลอัพ (R_p) เข้ากับแรงดัน +5V ไว้ด้วยเสมอ ดังแสดงในรูปที่ 2.22



รูปที่ 2.22 การต่อตัวต้านทานพูลอัพบนสายสัญญาณในระบบบัส I^2C

ในกรณีที่อาจมีแรงดันไฟกระชากขนาดใหญ่ปะปนเข้ามาในบัส I^2C ที่ขา SDA และ SCL ของอุปกรณ์แต่ละตัวต้องต่อตัวต้านทานอนุกรมกับขา SDA และ SCL เรียกว่า R_S ก่อนต่อเข้าสู่บัส I^2C ดังแสดงในรูปที่2.23



รูปที่ 2.23 การต่อตัวต้านทาน R_S เพื่อลดสัญญาณรบกวนขนาดใหญ่ที่อาจเข้ามาในบัส I^2C

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

อุปกรณ์ที่เป็นผู้สร้างข้อมูลหรือส่งข้อมูล เรียกว่า ตัวส่ง (transmitter)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

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

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 มีการเปลี่ยนแปลงระดับลอจิกจากสูงไปต่ำ ในขณะที่สาย SCL มีสถานะลอจิกสูง เรียกสภาวะที่เกิดขึ้นนี้ว่า "สภาวะเริ่มต้น (START)"

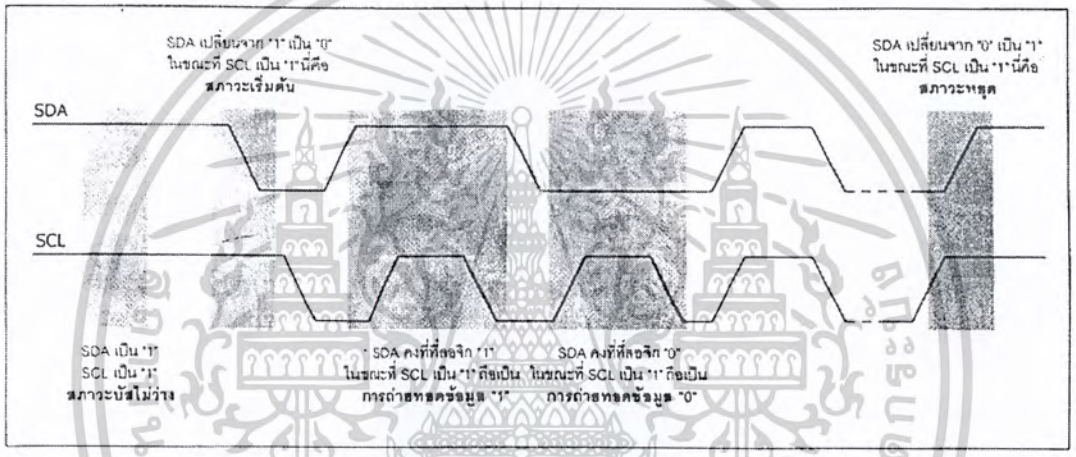
(3) **หยุดการถ่ายทอดข้อมูล (stop data transfer)** เกิดขึ้นเมื่อสาย SDA มีการเปลี่ยนแปลงระดับลอจิกจากต่ำไปสูง ในขณะที่สาย SCL มีสถานะลอจิกสูง เรียกสภาวะที่เกิดขึ้นนี้ว่า "สภาวะหยุด (STOP)"

(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 ต้องคงที่ตลอดช่วงเวลาที่สาย 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 ทางด้านตัวรับจะส่งบิตรับรู้ที่มีสถานะลอจิกต่ำลงบนบัส อุปกรณ์สเลฟที่ถูกอ้างถึงในการติดต่อหรือกำลังติดต่ออยู่ในขณะนั้นก็จะกำเนิดบิตรับรู้เพื่อตอบสนองให้ทราบว่าได้รับข้อมูลในแต่ละไบต์เรียบร้อยแล้ว

ในรูปที่ 2.24 เป็นไดอะแกรมเวลาที่แสดงถึงการเกิดสภาวะต่าง ๆ บนบัส I^2C ไม่ว่าจะเป็นสภาวะบัสว่าง, เริ่มต้น, ถ่ายทอดข้อมูล, รับรู้ และหยุดการถ่ายทอดข้อมูล



รูปที่ 2.24 ไดอะแกรมเวลาแสดงสถานะต่าง ๆ ในบัส I^2C

การทำงานบนบัส I^2C

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

ดังนั้นหัวใจสำคัญในอันดับแรกของการทำงานบนบัส I^2C คือ การอ้างถึงอุปกรณ์แต่ละตัว ซึ่งในที่นี้จะอธิบายรายละเอียดของการอ้างถึงทั้ง 2 รูปแบบ

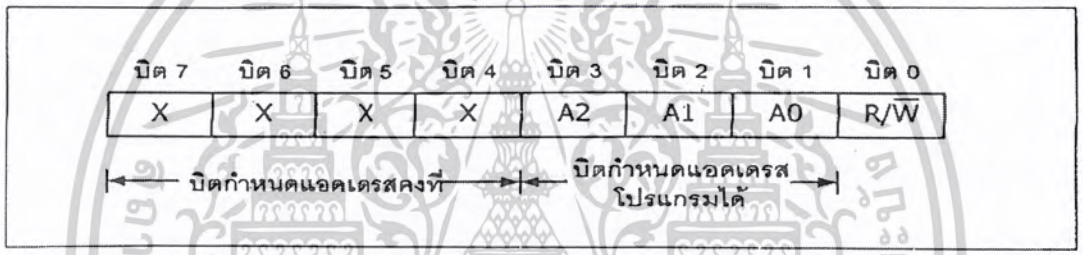
การอ้างถึงแบบ 7 บิต (7-bit addressing)

ข้อมูลไบต์แรกที่เกิดขึ้นหลังจากสภาวะเริ่มต้นคือข้อมูลที่ใช้ในการอ้างถึงอุปกรณ์ที่ต้องการติดต่อ หรือ ข้อมูลกำหนดแอดเดรส โดยมีรูปแบบแสดงในรูปที่ 2.2 ใน 7 บิตบนรวมทั้ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บิต MSB ด้วยจะเป็นข้อมูลแอดเดรสของอุปกรณ์สเลฟที่ต้องการติดต่อ โดยแบ่งเป็น บิตกำหนดแอดเดรสคงที่ (fixed address bit) จำนวน 4 บิต ซึ่งข้อมูลนี้อุปกรณ์แต่ละตัวจะถูกกำหนดมาจากผู้ผลิต ไม่สามารถเปลี่ยนแปลงแก้ไขได้ ถัดมาอีก 3 บิตเป็นบิตกำหนดแอดเดรสที่สามารถโปรแกรมได้ (programmable address bit) โดยผู้ใช้งานต้องกำหนดสถานะลอจิกในบิต LSB เป็นบิตที่ใช้กำหนดการอ่านหรือเขียนข้อมูลกับอุปกรณ์สเลฟตัวนั้น ๆ หากบิต LSB เป็น "0" หมายถึงต้องการเขียนข้อมูลไปยังอุปกรณ์นั้น ถ้าเป็น "1" จะเป็นการอ่านข้อมูลจากอุปกรณ์เสลฟ

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



รูปที่ 2.25 รูปแบบของข้อมูลกำหนดแอดเดรสที่ใช้ในการอ้างถึงแบบ 7 บิต

ข้อมูลในไบต์ต่อมา คือ ข้อมูลที่ทำการถ่ายทอดจริง (data)

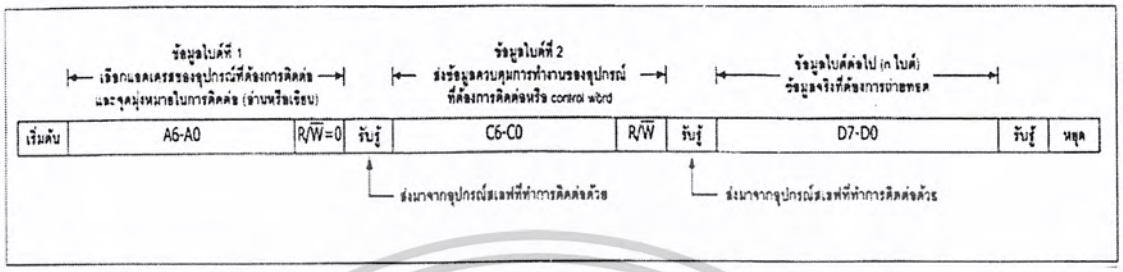
หลังจากที่มีการถ่ายทอดข้อมูลในไบต์ อุปกรณ์สเลฟที่ได้รับการติดต่อต้องส่งสัญญาณรับรู้ตอบกลับมาด้วยทุกครั้ง เพื่อให้กระบวนการถ่ายทอดข้อมูลสามารถดำเนินต่อไปได้ ในรูปที่ 2.21 แสดงรูปแบบข้อมูลอนุกรมที่เกิดขึ้นในการติดต่อบนบัส I^2C ของการอ้างถึงแบบ 7 บิต

การอ้างถึงแบบ 10 บิต

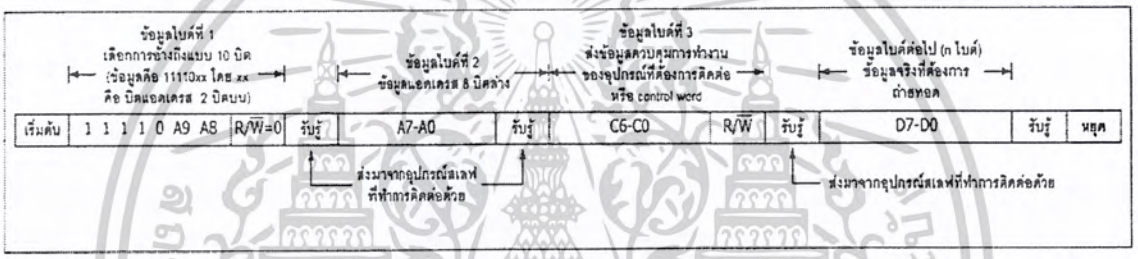
ในการอ้างถึงแบบนี้ยังคงใช้รูปแบบข้อมูลอนุกรมที่เหมือนกันแบบ 7 บิต หากแต่จะมีข้อมูลเพิ่มเติมขึ้นมาเล็กน้อย โดยในข้อมูลไบต์แรกหลังจากเกิดสภาวะเริ่มต้น ต้องกำหนดให้ 5 บิตบนมีข้อมูลเป็น 11110 ส่วนอีก 2 บิตถัดมาเป็นบิตแอดเดรสของอุปกรณ์ที่ต้องการติดต่อ ในบิต LSB ของข้อมูลไบต์แรกยังคงเป็นการกำหนดว่าต้องการอ่านหรือเขียนข้อมูลกับอุปกรณ์สเลฟตัวที่ต้องการติดต่อด้วย ข้อมูลไบต์ต่อมาเป็นข้อมูลแอดเดรสในไบต์ที่ 2 ของอุปกรณ์ที่ต้องการติดต่อด้วย ข้อมูลไบต์ถัดไปจึงเป็นข้อมูลควบคุม ข้อมูลหลังจากนั้นก็จะเป็นข้อมูลจริงที่ใช้ในการติดต่อเช่นเดียวกับการอ้างถึงแบบ 7 บิต หลังจากถ่ายทอดข้อมูลครบทุกไบต์ ต้องมีสภาวะรับรู้เกิดขึ้น เพื่อ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ให้กระบวนการถ่ายทอดข้อมูลสามารถดำเนินต่อไปได้ ในรูปที่ 2.22 แสดงรูปแบบข้อมูลอนุกรมของการอ้างถึงแบบ 10 บิต



รูปที่ 2.26 รูปแบบของข้อมูลอนุกรมที่ใช้ติดต่อกับอุปกรณ์บัส I^2C เมื่อใช้การอ้างถึงแบบ 7 บิต



รูปที่ 2.27 รูปแบบของข้อมูลอนุกรมที่ใช้ติดต่อกับอุปกรณ์บัส I^2C เมื่อใช้การอ้างถึงแบบ 10 บิต

2.4 ไอซีขยายพอร์ตอินพุตเอาต์พุตเบอร์ PCF8574A

ข้อมูลเบื้องต้นของ PCF8574A

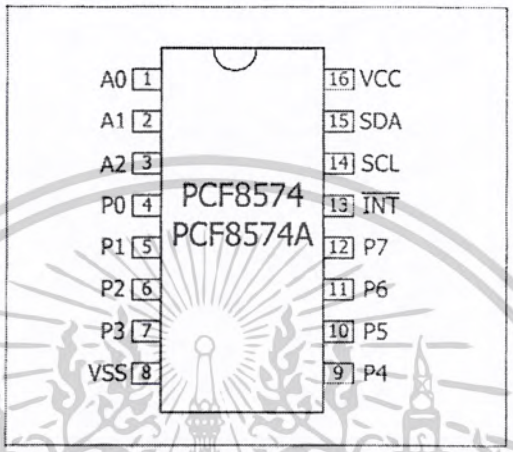
คุณสมบัติ

- ทำงานที่ระดับแรงดันตั้งแต่ 2.5V ถึง 6V
- กินกระแสในสถาวะสแตนด์บายต่ำเพียง 10 uA
- ใช้การเชื่อมต่อแบบบัส I^2C
- มีเอาต์พุตอินเตอร์รัปต์แบบเดรนปิด
- เอาต์พุตสามมารถขับกระแสสูง โดยสามารถนำไปขับ LED ได้โดยตรง และสามารถแลตช์ค่าได้
- สามารถกำหนดตำแหน่งแอดเดรสของไอซีแต่ละตัวได้ทางฮาร์ดแวร์ ด้วยขา A0-A2 ทำให้สามารถต่อพ่วงกันได้ถึง 8 ตัว

การจัดการขาของไอซี PCF8574A แสดงในรูปที่ 2.28 การทำงานของแต่ละขาแสดงในตารางที่ 2.4 ขาพอร์ตทั้ง 8 ขาของ PCF8574A สามารถกำหนดให้อิมพุตหรือเอาต์พุตได้โดยไม่จำเป็นต้องใช้คำสั่งควบคุมเพื่อเลือกให้เป็นขาเอาต์พุตหรือขาอินพุต ลักษณะวงจรภายในของพอร์ต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อินพุตเอาต์พุตแสดงในรูปที่ 2.29 เมื่อจ่ายไฟให้กับ PCF8574A ครั้งแรก ขาพอร์ตทั้ง 8 ขาจะมีลอจิกเป็น 1 ซึ่งจะเป็นการจ่ายกระแสมาจากแหล่งจ่ายกระแสคงที่ภายในตัวไอซี ทำให้มีกระแสในขณะลอจิก 1 นี้เพียง 100 uA เท่านั้น ในกรณีที่ต้องการให้มีแหล่งจ่ายกระแสสูงๆ จำเป็นต้องต่อตัวต้านทานพูลอัปเอาไว้ที่ขาพอร์ตเหล่านี้ด้วย



รูปที่ 2.28 การจัดขาไอซีขยายพอร์ตอินพุตเอาต์พุต PCF8574/ PCF8574A

เมื่อต้องการให้ขาพอร์ตเหล่านี้ทำหน้าที่เป็นอินพุตจะต้องส่งสัญญาณให้ขาเหล่านี้มีลอจิก "1" เสียก่อน เมื่อขาอินพุตได้รับสัญญาณจากภายนอกเข้ามาไอซี PCF8574A จะสร้างสัญญาณอินเตอร์รัปต์ (INT) ป้อนให้ไมโครคอนโทรลเลอร์หรือคอมพิวเตอร์รับรู้การต้องคอยตรวจสอบขาอินพุตอยู่ตลอดเวลา สัญญาณอินเตอร์รัปต์จะถูกรีเซตเมื่อมีการอ่านข้อมูลหรือมีการเปลี่ยนค่าของอิมพุตไปสู่ค่าเดิม

การเขียนโปรแกรมควบคุม PCF8574A

เนื่องจาก PCF8574A มีการเชื่อมต่อแบบบัส I^2C ดังนั้นการติดต่อจึงสามารถใช้โปรแกรมย่อยการติดต่อกับอุปกรณ์สเลฟ การสร้างสภาวะเริ่มต้น,สภาวะหยุด,การอ่านและเขียนข้อมูล ส่วนที่ต้องเปลี่ยนแปลงสำหรับ PCF8574A คือข้อมูลกำหนดแอดเดรส โดยข้อมูลของ PCF8574A มีรูปแบบดังนี้

บิต 7	บิต 6	บิต 5	บิต 4	บิต 3	บิต 2	บิต 1	บิต 0
0	1	1	1	A2	A1	A0	R/W

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

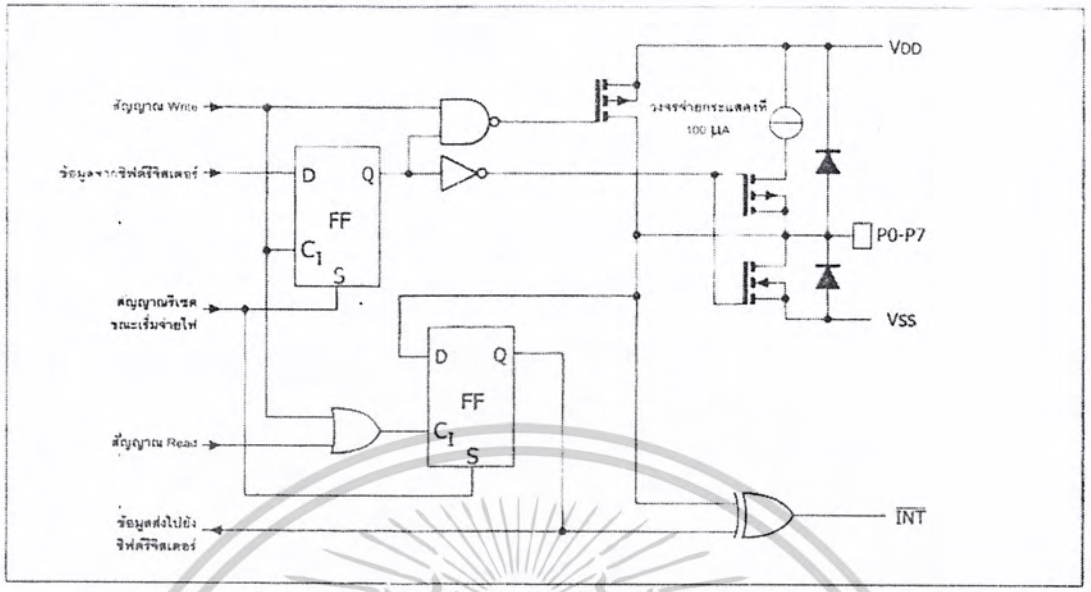
บิต A0 , A1 , A2 ใช้ในการระบุ PCF8574A ที่ใช้บนบอร์ดในกรณีที่มีการต่อ PCF8574 มากกว่า 1 ตัว โดยค่าของ A0-A2 จะมีความแตกต่างกันไปในแต่ละตัว สามารถกำหนดได้ทางฮาร์ดแวร์ โดยการต่อขา A0-A2 เข้ากับไฟเลี้ยง + 5V เพื่อกำหนดเป็นลอจิก "1" หรือกราวด์เพื่อกำหนดเป็นลอจิก "0" ดังนั้นค่าแอดเดรสของ PCF8574A 1 ตัว จึงต้องนำสถานะที่กำหนดทางฮาร์ดแวร์ที่ขา A0-A2 มารวมจึงจะสมบูรณ์ ส่วนบิต P/W ใช้กำหนดว่าต้องการอ่านหรือเขียนข้อมูลกับไอซี PCF8574A ยกตัวอย่าง ถ้าหากกำหนดขา A0-A2 ลงกราวด์ทั้งหมดและต้องการอ่านข้อมูลจาก PCF8574A ข้อมูลกำหนดแอดเดรส ที่ส่งให้แก่ PCF8574A คือ 01110001B เป็นต้น

ไอซี PCF8574A ยังมีอีกเบอร์หนึ่งในอนุกรมเดียวกัน คือ PCF8574 ซึ่งมีข้อมูลกำหนดแอดเดรสที่แตกต่างกับ PCF8574A แต่ฟังก์ชั่นการทำงานเหมือนกันทุกประการ โดยข้อมูลของ PCF8574 มีดังนี้

ชื่อ	ตำแหน่งขา	หน้าที่
A0	1	อินพุตแอดเดรสตัวที่ 1
A1	2	อินพุตแอดเดรสตัวที่ 2
A2	3	อินพุตแอดเดรสตัวที่ 3
PO	4	พอร์ตอิมพุตเอาต์พุต 2 ทิศทางบิต 0
P1	5	พอร์ตอิมพุตเอาต์พุต 2 ทิศทางบิต 1
P2	6	พอร์ตอิมพุตเอาต์พุต 2 ทิศทางบิต 2
P3	7	พอร์ตอิมพุตเอาต์พุต 2 ทิศทางบิต 3
Vss	8	กราวด์
P4	9	พอร์ตอิมพุตเอาต์พุต 2 ทิศทางบิต 4
P5	10	พอร์ตอิมพุตเอาต์พุต 2 ทิศทางบิต 5
P6	11	พอร์ตอิมพุตเอาต์พุต 2 ทิศทางบิต 6
P7	12	พอร์ตอิมพุตเอาต์พุต 2 ทิศทางบิต 7
INT	13	ขาเอาต์พุตอินเตอร์รัปต์ (ทำงานที่ลอจิก 0)
SCL	14	ขาสัญญาณนาฬิกาสำหรับ I^2C บัส
SCA	15	ขาข้อมูลสำหรับ I^2C บัส
Vdd	16	ไฟเลี้ยง

ตารางที่ 2.4 รายละเอียดหน้าที่การทำงานของแต่ละขาของไอซี PCF8574/PCF8574A

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.29 วงจรภายในขาพอร์ตของไอซี PCF8574/PCF8574A

2.5 LCD (Liquid Crystal Displays)

รายละเอียดเกี่ยวกับโมดูล LCD

ในโมดูล LCD จะมีส่วนประกอบหลัก ๆ 3 ส่วน ดังนี้

ตัวแสดงผล (display) ภายในเป็นผลึกเหลวที่สามารถแสดงผลให้เห็นโดยอาศัยแสงจากภายนอก ดังนั้นจึงต้องมีมุมในการมองข้อมูลที่แสดงผลบนจอ LCD

ตัวควบคุม (controller) เป็นตัวรับข้อมูลจากอุปกรณ์ภายนอกมาควบคุมการทำงานของโมดูล LCD เช่น ลบจอภาพ แสดงตัวอักษร หรือเลื่อนเคอร์เซอร์ เป็นต้น ตัวควบคุมนี้ใช้ชิปควบคุมโดยเฉพาะชิปที่นิยมใช้ คือ เบอร์ HD44780 และ HD61830 โดย HD44780 จะใช้ควบคุม LCD แบบอักขระ ส่วน HD61830 ใช้ควบคุม LCD แบบกราฟิก

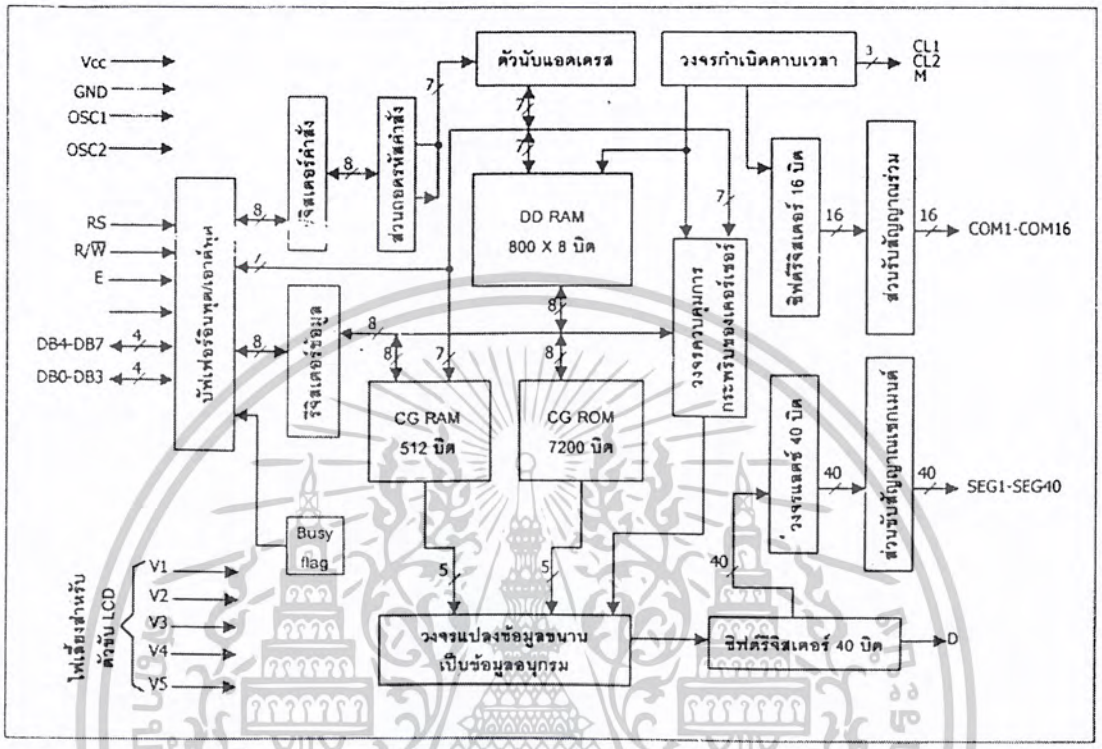
ตัวขับ (driver) เป็นตัวรับสัญญาณจากตัวควบคุมมาขับให้ตัวแสดงผลแสดงข้อมูลตามที่กำหนด ชิปที่ใช้ทำหน้าที่เป็นตัวขับนี้ได้แก่ เบอร์ HD44100H และ MSM5259 เป็นต้น

โครงสร้างภายในของตัวควบคุมโมดูล LCD

ในการใช้งานโมดูล LCD จำเป็นต้องทำความเข้าใจเกี่ยวกับโครงสร้างและคำสั่งที่ใช้ในการควบคุมให้ดีเสียก่อน ในที่นี้ขอยกตัวอย่างโมดูล LCD แบบอักขระ ซึ่งเป็น LCD ที่ใช้ในตัว project จริง ๆ เพราะสามารถเข้าใจได้ง่าย ให้รูปที่ 2.30 เป็นบล็อกไดอะแกรมภายในของชิปควบคุม LCD เบอร์ HD44780 ซึ่งใช้ใน โมดูล LCD แบบอักขระ ประกอบด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บัฟเฟอร์อินพุตเอาต์พุต เป็นส่วนที่ใช้ในการติดต่อรับส่งข้อมูลกับอุปกรณ์ภายนอก เพื่อที่จะถ่ายทอดข้อมูลเข้าออกภายในตัวควบคุม



รูปที่ 2.30 ไดอะแกรมการทำงานของโมดูล LCD แบบอักขระ

รีจิสเตอร์คำสั่ง (Instruction 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

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

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

แฟลก BUSY เป็นส่วนที่ทำหน้าที่แจ้งสถานะการทำงานของตัวควบคุมให้อุปกรณ์ภายนอกทราบว่า ตัวควบคุมพร้อมที่จะรับข้อมูลหรือคำสั่งหรือไม่ ดังนั้นก่อนการส่งข้อมูลหรือคำสั่งมายังตัวควบคุมต้องตรวจสอบสถานะของแฟลก BUSY นี้เสียก่อน

โมดูล LCD ขนาด 20 ตัวอักษร 4 บรรทัด(LCD 20×4) แบบมี LED Backlight

โมดูล LCD ขนาด 20×4 มีขาต่อใช้งานทั้งสิ้น 16 ขา สำหรับรายละเอียดการทำงานของแต่ละขามีดังนี้

A (ขา 1) : ขา แอโนด ของ LED Backlight

K (ขา 2) : ขา คาโทด ของ LED Backlight

V_{SS} (ขา 3) : ต่อกราวด์

V_{DD} (ขา 4) : ต่อไฟเลี้ยง +5 โวลต์

V_O (ขา 5) : เป็นขาอินพุทรับแรงดันเพื่อปรับความเข้มการแสดงผล

RS (ขา 4) : เป็นขาอินพุทใช้ในการแยกชนิดของข้อมูลที่ทำการประมวลผลในขณะนั้นว่าเป็นคำสั่งสำหรับรีจิสเตอร์ IR หรือเป็นข้อมูลสำหรับรีจิสเตอร์ DR โดยถ้าขานี้เป็น "0" ข้อมูลที่ส่งมาจะเป็นคำสั่ง แต่ถ้าขานี้เป็น "1" ข้อมูลที่ส่งมาจะเป็นข้อมูลสำหรับการแสดงผล

R/$\overline{W}$ (ขา 5) : เป็นขาที่ใช้เลือกการอ่านหรือเขียนข้อมูลกับ LCD ถ้าเป็น "0" เป็นการกำหนดให้เขียนข้อมูล แต่ถ้าเป็น "1" จะเป็นการอ่านข้อมูล

E (ขา 6) : เป็นขาอีนาเบิล LCD ให้ทำงาน

D0-D7 (ขา 7-14) : เป็นขาที่ใช้เป็นทางผ่านของข้อมูลระหว่าง LCD กับอุปกรณ์ภายนอกขนาด 8 บิต

คำสั่งควบคุมโมดูล LCD

ในการเขียนคำสั่งลงในตัวควบคุม แน่นอนว่าต้องกำหนดให้ขา RS และ R/W เป็น "0" แล้วเขียนคำสั่งตามไป คำสั่งควบคุมโมดูล LCD ของชิปควบคุม HD44780 ที่สำคัญมี 10 คำสั่งดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. คำสั่งเคลียร์ตัวแสดงผล (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" เป็นคำสั่งให้เคอร์เซอร์เคลื่อนที่กลับไปยังตำแหน่งซ้ายสุดของจอแสดงผล แต่ข้อมูลบนจอแสดงผลไม่เปลี่ยนแปลงนั่นคือ ข้อมูลคำสั่งของคำสั่งนี้จะเป็น 02H หรือ 03H ก็ได้

3. คำสั่งเลือกโหมดการป้อนข้อมูล (Entry mode Set)

มีรายละเอียดของรูปแบบข้อมูลคำสั่งดังนี้

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" เมื่อเกิดข้อมูลใหม่ตัวเคอร์เซอร์จะเลื่อนไปทางขวามือ

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

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

4. คำสั่งควบคุมการแสดงผล

มีรายละเอียดของรูปแบบข้อมูลคำสั่งดังนี้

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

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บิต C ใช้ควบคุมการแสดงตัวเคอร์เซอร์บนจอแสดงผล ถ้าต้องการให้มีเคอร์เซอร์แสดงผลบนจอแสดงผล ต้องกำหนดให้บิตนี้เป็น "1" ถ้ากำหนดให้เป็น "0" จะเป็นการปิดเคอร์เซอร์ หรือไม่แสดงเคอร์เซอร์

บิต B ใช้ควบคุมการกะพริบของเคอร์เซอร์ ถ้าบิตนี้เป็น "1" เคอร์เซอร์จะกะพริบ

ดังนั้นจะมีข้อมูลคำสั่งได้ตั้งแต่ 08H – 0FH (8 รูปแบบคำสั่ง) ที่ใช้บ่อยคือ 0CH เป็นการสั่งให้เปิดจอแสดงผล แต่ไม่แสดงเคอร์เซอร์ และ 0FH เป็นการสั่งให้เปิดจอแสดงผล แสดงเคอร์เซอร์และสั่งให้เคอร์เซอร์กะพริบ

5. คำสั่งควบคุมการเลื่อนเคอร์เซอร์และข้อมูลตัวอักษร

มีรายละเอียดของรูปแบบข้อมูลคำสั่งดังนี้

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 – 13H
0	1	เลื่อนเคอร์เซอร์ไปทางขวา	14H – 17H
1	0	เลื่อนตัวอักษรใหม่ไปทางซ้าย	18H – 1BH
1	1	เลื่อนตัวอักษรใหม่ไปทางขวา	1CH – 1FH

6. คำสั่งกำหนดฟังก์ชันการทำงาน

มีรายละเอียดของรูปแบบข้อมูลคำสั่งดังนี้

บิต7	บิต6	บิต5	บิต4	บิต3	บิต2	บิต1	บิต0
0	0	1	DL	N	F	*	*

บิต DL 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"

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บิต F ใช้เลือกความละเอียดของตัวอักษรให้การแสดงผล ถ้าบิตนี้เป็น “0” จะเป็นการแสดงผลแบบ 5×7 จุด และถ้าเป็น “1” จะแสดงผลเป็นแบบ 5×10 จุด

ข้อมูลคำสั่งที่ใช้บ่อยคือ 38H เป็นการกำหนดให้โมดูล LCD ทำงานในแบบ 8 บิต แสดงผล 2 บรรทัด และเลือกความละเอียดเป็น 5×7 จุด

7. คำสั่งเลือกแอดเดรสของ CGRAM

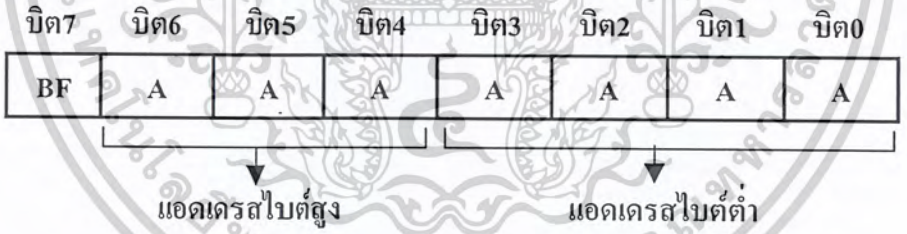
เมื่อต้องการกำหนดแอดเดรสของ CGRAM ต้องกำหนดให้บิต 7 เป็น “0” บิต 6 เป็น “1” ส่วนอีก 6 บิตที่เหลือจะแทนด้วยค่าแอดเดรสของ CGRAM จะต้องทำการกำหนดแอดเดรสด้วยคำสั่งนี้ก่อนที่จะอ่านหรือเขียนข้อมูลให้ CGRAM โดยแอดเดรสของ CGRAM อยู่ระหว่าง 00H-3FH

8. คำสั่งเลือกแอดเดรสของ DDRAM

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

9. คำสั่งอ่านแฟลก BUSY และแอดเดรส

มีรายละเอียดของรูปแบบข้อมูลคำสั่งดังนี้



เป็นคำสั่งที่ใช้อ่านแฟลก BUSY (BF) โดยแฟลกนี้จะเป็นตัวบอกสถานะของตัวควบคุม LCD ว่าพร้อมจะรับข้อมูลอยู่หรือไม่ ถ้าหากบิต BF เป็น “0” แสดงว่าตัวควบคุม LCD พร้อมรับข้อมูลหรือคำสั่ง แต่ถ้าเป็น “1” แสดงว่า ขณะนี้ตัวควบคุม LCD ยังอยู่ในกระบวนการทำงานภายในหรือกำลังประมวลผลข้อมูลอยู่ ยังไม่พร้อมรับข้อมูลหรือคำสั่ง

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

นอกจากนี้ยังใช้เป็นคำสั่งอ่านข้อมูลแอดเดรสของ CGRAM และ DDRAM ด้วย โดยบิต0–บิต6 เป็นค่าข้อมูลของ แอดเดรสที่ต้องการอ่าน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การเขียนคำสั่งและข้อมูลให้แก่โมดูล LCD

ในการเขียนข้อมูลเพื่อควบคุมให้โมดูล LCD แสดงผลตามที่ผู้ใช้งานต้องการ ต้องส่งคำสั่ง (instruction) แล้วกำหนดโหมดการทำงานให้แก่โมดูล LCD ก่อน จากนั้นจึงค่อยส่งข้อมูล (data) ที่ต้องการแสดงผลเนื่องจากบัสข้อมูลของโมดูล LCD มี 8 เส้น คือ D0-D7 และใช้เป็นทางผ่านของทั้งคำสั่งและข้อมูล ดังนั้นในการส่งคำสั่งและข้อมูลจึงต้องอาศัยการกำหนดสัญญาณลอจิกที่ขา RS ถ้าหากที่ขา RS ได้ลอจิก "0" หมายความว่า ข้อมูลที่ป้อนให้แก่โมดูล LCD ขณะนั้นเป็นคำสั่งในทางตรงข้าม หากขา RS ได้รับลอจิก "1" ข้อมูลที่ป้อนให้ขณะนั้นเป็นข้อมูลที่ใช้ในการแสดงผล

เมื่อต้องการเขียนหรืออ่านข้อมูลใน CGRAM และ DDRAM เริ่มต้นต้องกำหนดแอดเดรสที่ต้องการอ่านหรือเขียนก่อน โดยใช้คำสั่งเลือกแอดเดรส จากนั้นกำหนดให้ขา RS เป็น "1" เพื่อแจ้งให้ตัวควบคุมภายในโมดูล LCD ทราบว่าข้อมูลที่ปรากฏต่อไปนี้เป็นข้อมูลปกติไม่ใช่คำสั่ง

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

ในกรณีที่ต้องการเขียนข้อมูล เมื่อกำหนดแอดเดรสและป้อนลอจิก "1" ให้ขา RS แล้ว แล้วต้องกำหนดให้ขา R/$\overline{W}$ เป็น "0" ข้อมูลที่อยู่บนบัสข้อมูลจะถูกเขียนลงในรีจิสเตอร์ DR จากนั้นจึงถ่ายทอดลงใน DDRAM ต่อไป

จังหวะการทำงานของ LCD โมดูล

ในการติดต่อกับโมดูล LCD จะต้องมีการหน่วงเวลาหลังจากที่ทำการส่งรหัสคำสั่งหรือข้อมูลเนื่องจากต้องรอให้คอนโทรลเลอร์ภายใน LCD โมดูล แปลความหมายของรหัสคำสั่งและทำงานตามคำสั่งให้เรียบร้อยก่อน จากนั้นจึงจะรับข้อมูลหรือดำเนินการต่อไป

ดังนั้น ในการใช้งานโมดูล LCD ผู้เขียนโปรแกรมต้องมีโปรแกรมเพื่อหน่วงเวลารอให้โมดูล LCD พร้อมทำงานด้วย โดยเมื่อเริ่มจ่ายไฟให้แก่โมดูล LCD ต้องรอประมาณ 10 มิลลิวินาที เพื่อให้โมดูล LCD ทำการเตรียมความพร้อมหรืออินิเชียล (initial) หลังจากนั้นก็จะกำหนดลอจิกให้แก่ขา RS ของโมดูล LCD แล้วต้องหน่วงเวลาอีกประมาณ 2 มิลลิวินาทีเพื่อให้คอนโทรลเลอร์ใน LCD โมดูลแปลความหมายของลอจิกที่ขา RS ว่าข้อมูลต่อไปที่จะได้รับนั้นเป็นรหัสคำสั่งหรือเป็นข้อมูลที่ต้องการแสดงผล จากนั้นจะเป็นการส่งข้อมูลมารอที่บัสข้อมูล D0-D7 (กรณีทำงานในโหมด 8 บิต) ขั้นตอนต่อไปจะเป็นการส่งสัญญาณพัลส์ไปที่ขา E เพื่ออีนาเบิลโมดูล LCD ให้รับข้อมูลจากบัสข้อมูลเข้าไป โดยพัลส์ที่ป้อนเข้าที่ขา E ของโมดูล LCD ต้องเป็นพัลส์ขอบขาขึ้น จากนั้นทำการหน่วงเวลา 2 มิลลิวินาที

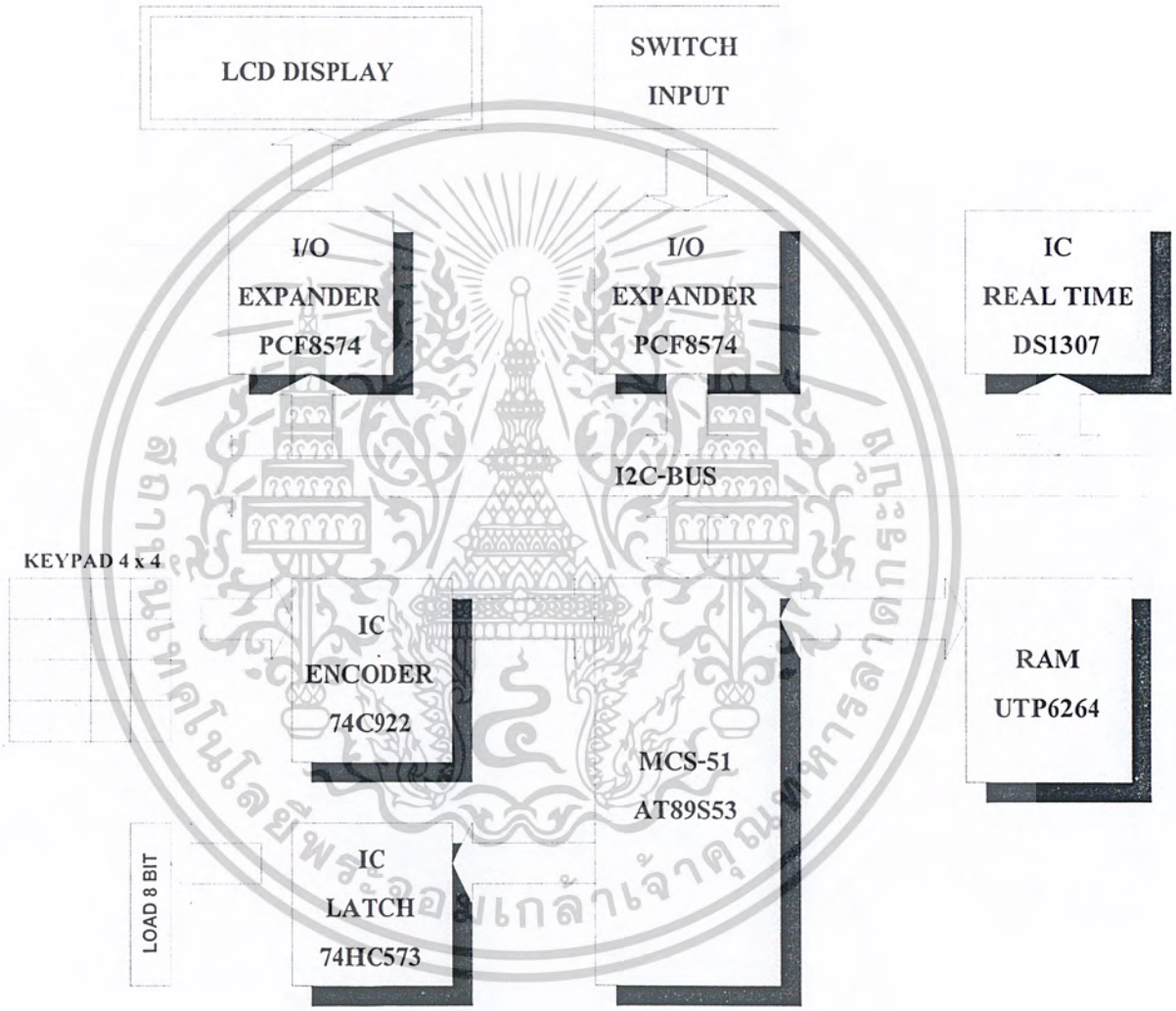
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ทั้งหมดที่กล่าวมาคือขั้นตอนและจังหวะในการทำงาน 1 รอบของโมดูล LCD จะเห็นได้ว่ามีโปรแกรมย่อยที่สำคัญอยู่ 3 โปรแกรมย่อยคือ โปรแกรมอินนิเชียล LCD , โปรแกรมหน่วงเวลา และ โปรแกรมย่อยการส่งพัลส์เพื่ออีนาเบิลโมดูล LCD

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3
การออกแบบและการสร้าง

3.1 แนวคิดและหลักการทำงาน



รูปที่ 3.1 บล็อกไดอะแกรมของวงจร

การทำงานโดยรวมของวงจร

จากบล็อกไดอะแกรมการทำงานของวงจร รูปที่ 3.1 จะใช้ไมโครคอนโทรเลอร์ตระกูล MCS-51 เบอร์ AT89S53 เป็นตัวประมวลผลหลักของระบบ โดยใช้การต่อเชื่อม แรมภายนอก (External RAM) เบอร์ UTP6264 เพื่อเป็นการขยายหน่วยความจำให้สามารถรองรับจำนวนข้อมูลที่ต้องการใช้งานจากผู้ใช้ให้ได้มากขึ้น และจะใช้ IC DS1307 เป็นตัวฐานเวลาหลักของระบบ โดยใช้

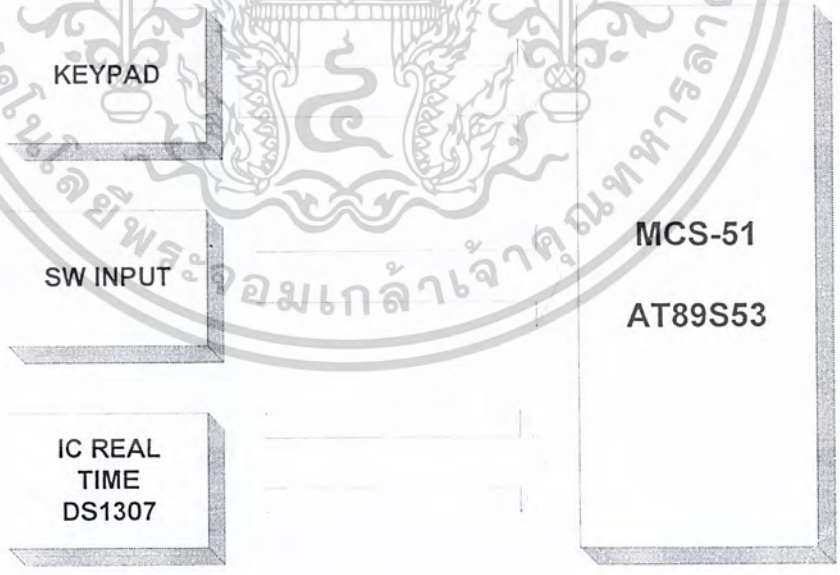
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การเชื่อมต่อในระบบบัส I^2C เพื่อใช้ในการรับส่งข้อมูลฐานเวลากับตัวไมโครคอนโทรลเลอร์เพื่อนำไปประมวลผล และในขณะเดียวกันการส่งผลลการทำงานที่ได้ไปแสดงผลยังจอแสดงผล(LCD) นั้นจะใช้การส่งข้อมูลผ่านระบบบัส I^2C เช่นกัน โดยส่งผ่านไอซีขยายพอร์ตเบอร์ PCF8574 ก่อนจะส่งไปยังจอแสดงผล รวมทั้งในส่วนของสวิตซ์อินพุต ต่างๆ จะใช้การรับส่งข้อมูลการทำงานผ่าน ไอซีขยายพอร์ตเบอร์ PCF8574 ทางระบบบัส I^2C เช่นเดียวกัน

และในส่วนของคีย์แพด จะใช้ ไอซีเข้ารหัส (IC Encoder) เบอร์ MM 74C922 เป็นตัวเข้ารหัสข้อมูลเพื่อส่งไปยังไมโครคอนโทรลเลอร์ โดยการใช้ไอซีเข้ารหัสเบอร์ MM 74C922 นี้จะช่วยประหยัดสายข้อมูลที่เชื่อมต่อกับไมโครคอนโทรลเลอร์ และง่ายต่อการเขียนโปรแกรมตรวจสอบหาคีย์ที่ถูกกดได้ง่ายขึ้น

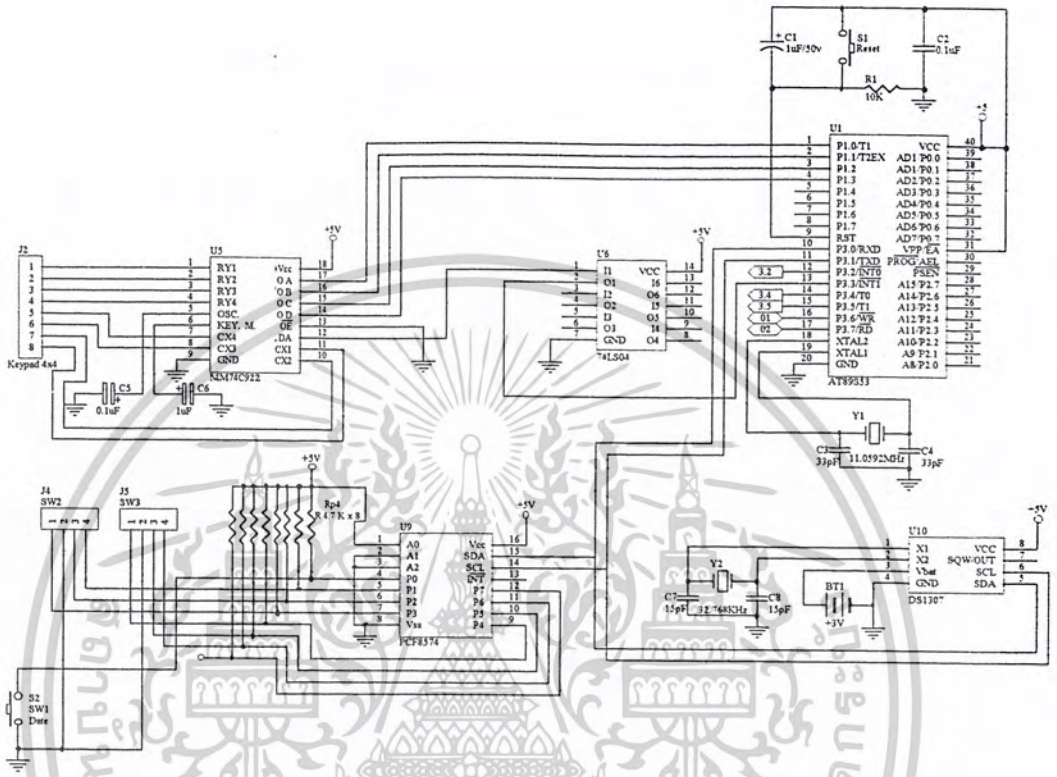
โดยในส่วนของภาคเอาต์พุตนั้น จะไม่สามารถใช้ไมโครคอนโทรลเลอร์ขับรีเลย์ ได้โดยตรง เนื่องจากกระแสที่จ่ายในแต่ละขาอยู่ที่ประมาณ 10 mA ซึ่งไม่เพียงพอต่อการขับรีเลย์ให้ทำงาน ดั้งนั้นจึงต้องนำ ไอซีบัฟเฟอร์เบอร์ MM 74HC573 มาต่อเพิ่มเพื่อยกระดับกระแสขึ้น โดยไอซีนี้สามารถจ่ายกระแสได้ถึง 35 mA ซึ่งเพียงพอต่อการขับรีเลย์ให้ทำงาน

2.2 วงจรภาคอินพุตของ MCS-51



รูปที่ 3.2 บล็อกไดอะแกรมภาคอินพุต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.3วงจรการเชื่อมต่อภาคอินพุต

จากในรูปที่ 3.3 ในส่วนของอินพุตตัวแรกคือคีย์แพด โดยคีย์แพดชุดนี้มีขนาด 4 x 4 ถ้าหากว่านำไปต่อเข้ากับไมโครคอนโทรลเลอร์โดยตรง จะเป็นการสิ้นเปลืองพอร์ตของไมโครคอนโทรลเลอร์ อีกทั้งยังเป็นการยุ่งยากต่อการเขียนโปรแกรมเพื่อตรวจสอบการหาคีย์ที่ถูกกด ดังนั้นด้วยเหตุผลนี้ จึงได้นำเอา IC MM74C922 ซึ่งเป็นไอซีเข้ารหัสคีย์แพดขนาด 4 x4 โดยตัวไอซีนี้จะสามารถให้รหัสในรูปแบบเลขฐานสอง เพียง 4 บิตซึ่งทำให้ง่ายต่อการเขียนโปรแกรมตรวจสอบมากขึ้นในขณะเดียวกันที่ขา DA ของไอซีถอดรหัสนี้จะเกิดค่าลอจิกเป็น “1” ทุกครั้งที่มีการกดคีย์ใดๆเกิดขึ้นโดยจะนำไปต่อเข้ากับขาอินเตอร์รัปต์ของไมโครคอนโทรลเลอร์ (INT1) เพื่อนำไปใช้ในการควบคุมการติดของ LED Blacklight และหน่วงเวลาให้ดับโดยอัตโนมัติแต่เนื่องจากการทำงานของไมโครคอนโทรลเลอร์ในการอินเตอร์รัปต์นั้นในสภาวะปกติจะต้องรักษาสภาวะลอจิก “1” ไว้ และจะเกิดการอินเตอร์รัปต์โดยสมบูรณ์เมื่อมีการเปลี่ยนจากลอจิก “1” มาเป็นลอจิก “0” โดยเป็นลอจิก “0” อยู่ในช่วงเวลาไม่ต่ำกว่า 1 ไซเกิล และกลับคืนเป็นลอจิก “1” ตามเดิม แต่เนื่อง

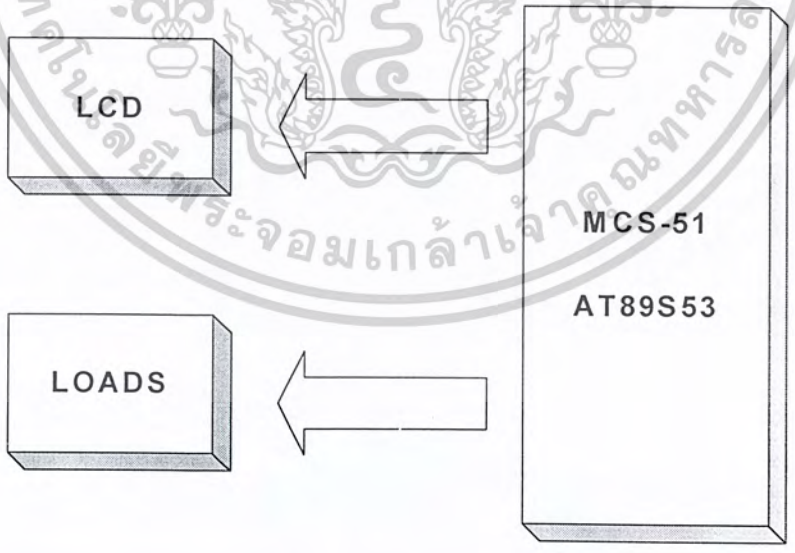
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากว่าค่าลอจิกที่ได้จากขา DA ของไอซีเข้ารหัสนั้นในสภาวะที่ไม่มีการกดใดๆจะให้เป็นลอจิก "0" จะให้ลอจิก "1" ก็ต่อเมื่อมีการกดคีย์ใดๆจึงจะให้ลอจิก "1" ออกมา ซึ่งจะเห็นได้ว่าตรงกันข้ามกับการทำงานในโหมดการอินเตอร์รัพท์ของไมโครคอนโทรลเลอร์ ดังนั้นจึงนำเอาไอซีเบอร์ 74LS04 มาใช้เป็นตัวกลับสภาวะของ สัญญาณนี้ก่อนไปเข้าไมโครคอนโทรเลอร์ และในส่วนของค่า C ที่ต่ออยู่กับขา 5 และ 6 ของไอซีเข้ารหัสคีย์แพดซึ่งจะเป็นตัวกำหนดความไวในการสแกนคีย์ โดยทั่วไปจะกำหนดให้ C6 มีค่าเป็น 10 เท่าของ C5

ในส่วนของอินพุทตัวที่สอง ซึ่งเป็นส่วนที่ทำหน้าที่รับข้อมูลการเลือกโหมดการทำงานโดยจะใช้ไอซีขยายพอร์ตเบอร์ PCF8574 ต่อวงจรใช้ตรวจสอบหาการเปลี่ยนแปลงของสวิตซ์เลือกโหมด โดยใช้การรับส่งข้อมูลผ่านระบบบัส I^2C โดยในส่วนที่เป็นขาอินพุทที่เข้ามายัง PCF8574 ที่ใช้ในการต่อสวิตซ์นั้น จำเป็นจะต้องต่อตัวต้านทานเข้ากับ +VCC เพื่อรักษาสภาวะ "1" ในกรณีที่ไม่มีการเลือกสวิตซ์ ณ บิตนั้น

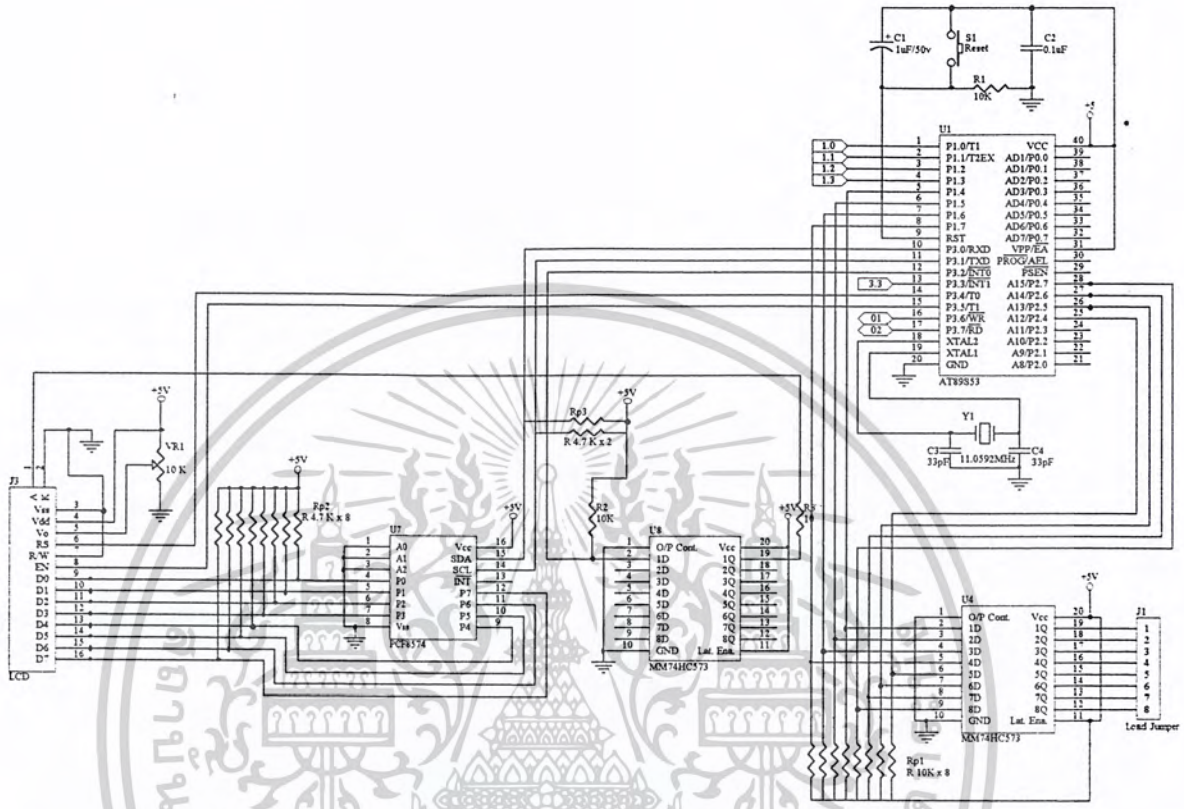
ในส่วนของอินพุทถัดมาซึ่งเป็นฐานเวลาหลักของการทำงานในวงจรโดยใช้ไอซีเบอร์ DS1307 และใช้การรับส่งข้อมูลผ่านระบบบัส I^2C เช่นเดียวกับ ตัว PCF8574 โดยไอซีฐานเวลานี้จะมีแบตเตอร์รี่ลิเธียมไอออนขนาด 3 V เป็นตัวคอยรักษาค่าฐานเวลาในขณะไฟฟ้าดับ

2.2 วงจรภาคเอาต์พุตของ MCS-51



รูปที่ 3.4 บล็อกไดอะแกรมภาคเอาต์พุต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



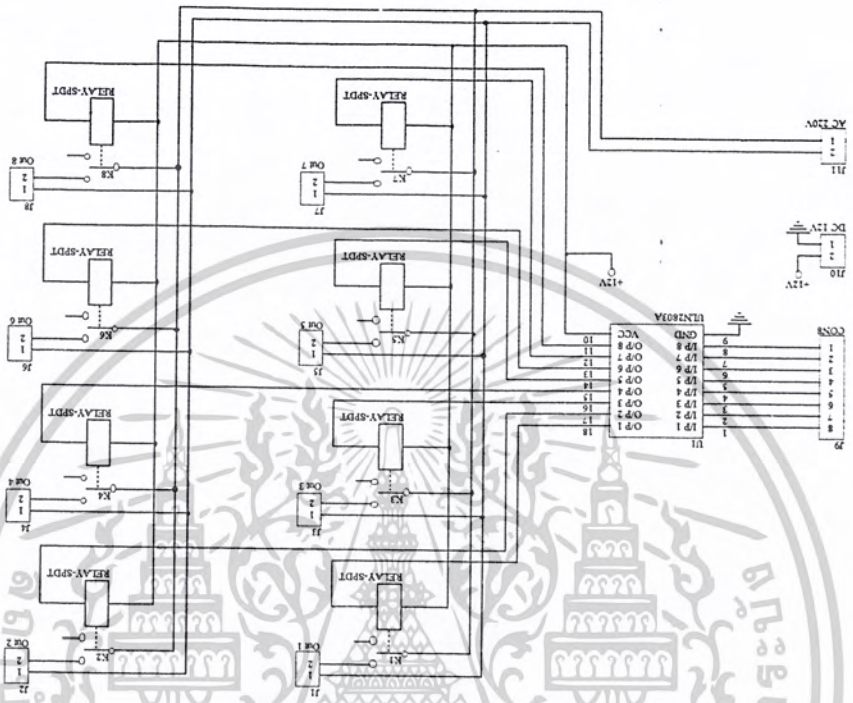
รูปที่ 3.5 วงจรการเชื่อมต่อภาคเอาต์พุต

จากรูปที่ 3.5 ในส่วนของเอาต์พุตตัวแรก ซึ่งเป็นจอแสดงผล (LCD) โดยได้รับข้อมูลจากไมโครคอนโทรลเลอร์ในโหมด 8 บิต ซึ่งใช้การส่งข้อมูลผ่านระบบบัส I^2C ด้วยไอซีขยายพอร์ดเบอร์ PCF8574 และในขณะเดียวกันการควบคุมหลอด LED Blacklight จะใช้การควบคุมบิต 3.2 ของ ไมโครคอนโทรลเลอร์เป็นตัวสั่ง เปิด/ปิด ตัว LED Blacklight โดยจะรับค่าช่วงการหน่วงเวลาการติดสว่างของ LED Blacklight จากโปรแกรมการอินเตอร์รัปต์ทางอินพุต

และในส่วนของเอาต์พุตตัวที่ 2 ซึ่งเป็นการควบคุมโหลดของระบบ จะใช้การสั่งงานจากไมโครคอนโทรลเลอร์ไปควบคุมการทำงานของรีเลย์โดยตรงไม่ได้ เนื่องจากกระดับกระแสที่จ่ายได้ในแต่ละบิตของไมโครคอนโทรลเลอร์อยู่ที่ระดับ 10 mA ซึ่งไม่เพียงพอต่อสั่งให้ตัวรีเลย์ทำงาน ดังนั้นจึงแก้ไขโดยการนำเอาไอซีบัฟเฟอร์เบอร์ 74HC573 มาต่อร่วมเข้าไปโดยจะสามารถยกระดับกระแสได้เพิ่มมากขึ้น และมีระดับการจ่ายกระแส อยู่ที่ 35 mA ของแต่ละบิตของไอซีบัฟเฟอร์นี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3 วงจรส่วนขับโหลด

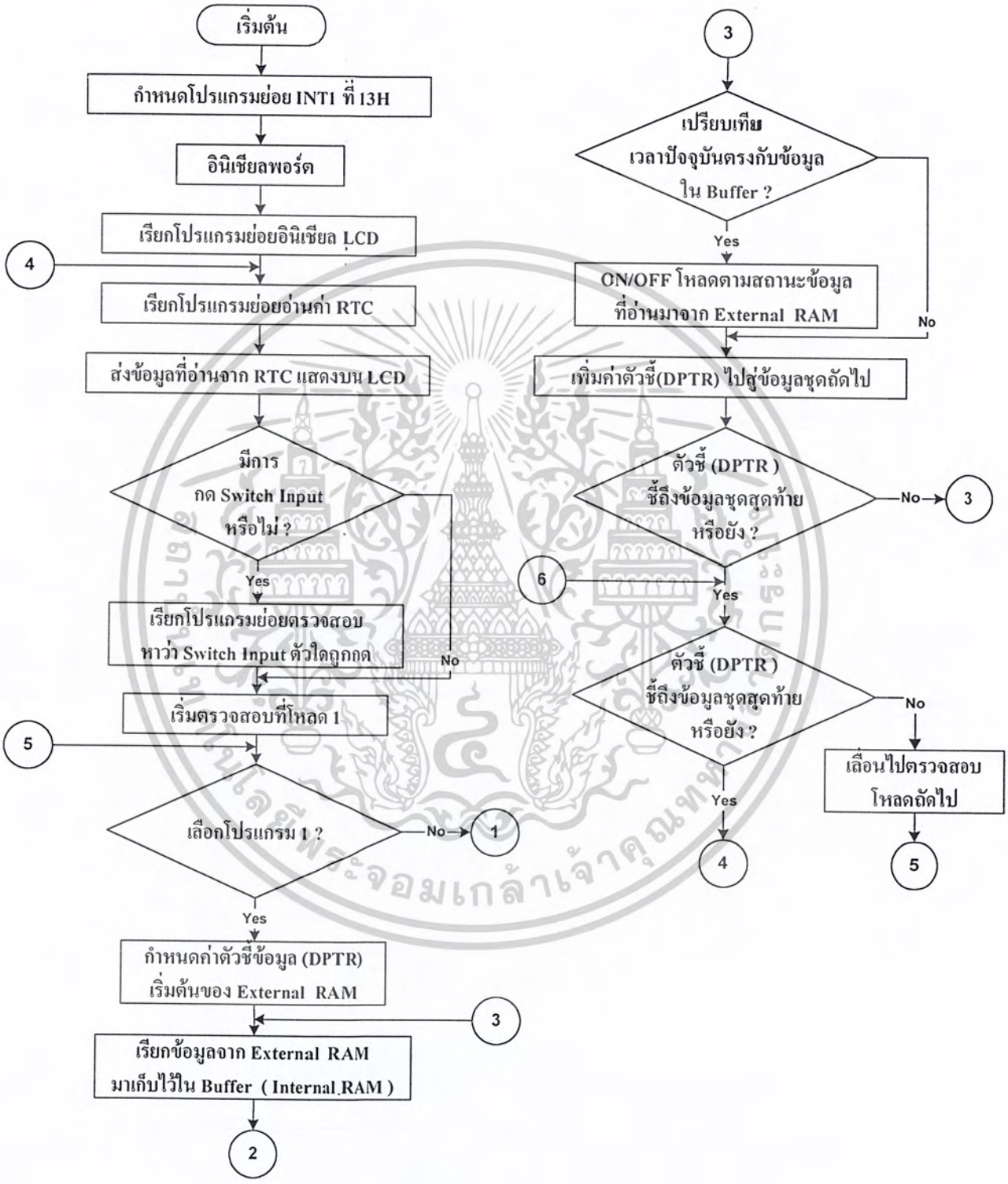


รูปที่ 3.6 วงจรส่วนขับโหลด

จากรูปที่ 3.6 เป็นวงจรในส่วนการขับโหลดเอาต์พุต โดยจะใช้การสั่งงานจากไมโครคอนโทรลเลอร์ไปยังรีเลย์โดยผ่านไอซีไดรเวอร์เบอร์ ULN2803A ซึ่งภายในบรรจุอินเวอร์เตอร์เกตแบบคอลเลคเตอร์เปิด 8 ตัวสามารถใช้กับแรงดันได้สูงสุด +50 V กระแสเอาต์พุตสูงสุดในแต่ละขาเท่ากับ 500 mA นอกจากนั้นวงจรภายในของไอซีไดรเวอร์ตัวนี้ยังต่อไดโอดป้องกันกระแสไหลย้อนกลับเนื่องจากสนามแม่เหล็กเกิดการยุบตัวไว้ที่ทุกขาเอาต์พุต ทำให้สามารถต่อโหลดที่เป็นขดลวดได้ทันที

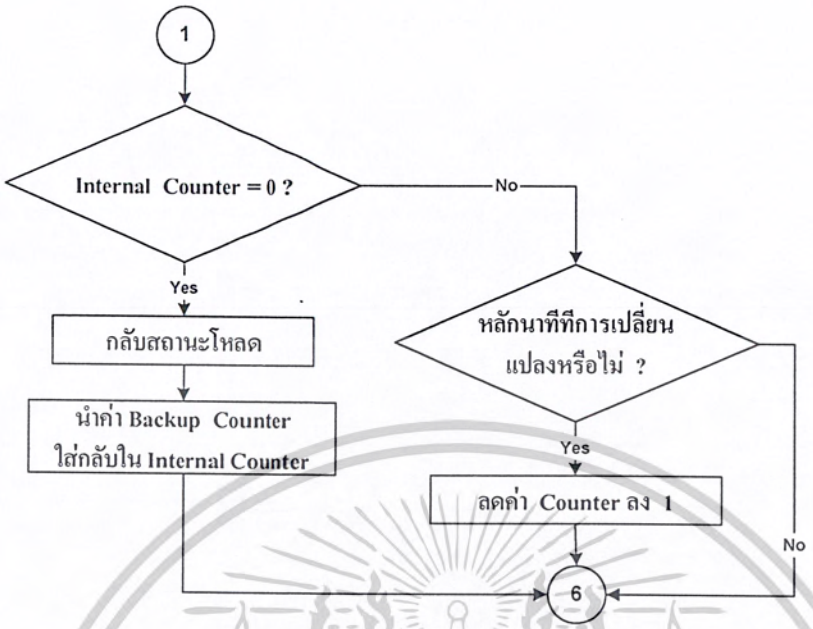
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.5 โฟลวชาร์ตของโปรแกรมหลัก



(ก)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



(ข)

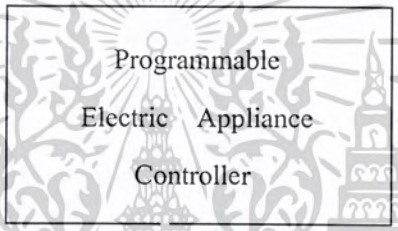
รูปที่ 3.7 โฟลวชาร์ตของโปรแกรมหลัก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4
การทดลองและผลการทดลอง

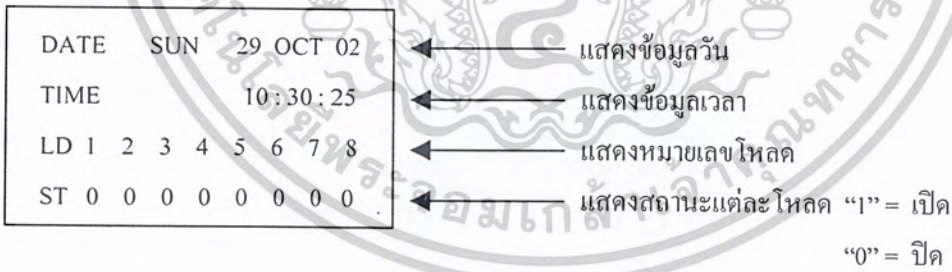
ในหัวข้อนี้จะเป็นการกล่าวถึงการใช้งานและผลการทำงานที่ได้ (คู่มือการใช้งาน)

1. ในสภาวะก่อนการใช้งาน จะต้องตรวจสอบสวิตซ์ ด้านหน้าเครื่องให้อยู่ตรงตำแหน่ง AUTO และ ตำแหน่ง RUN
2. เสียบปลั๊ก และกดสวิตซ์เปิดเครื่อง
3. หน้าจอจะแสดงสภาวะเริ่มต้นการทำงานดังรูปที่ 4.1



รูปที่ 4.1 สภาวะเริ่มต้นการทำงานบนหน้าจอ LCD

4. การแสดงผลในสภาวะการทำงานปกติ

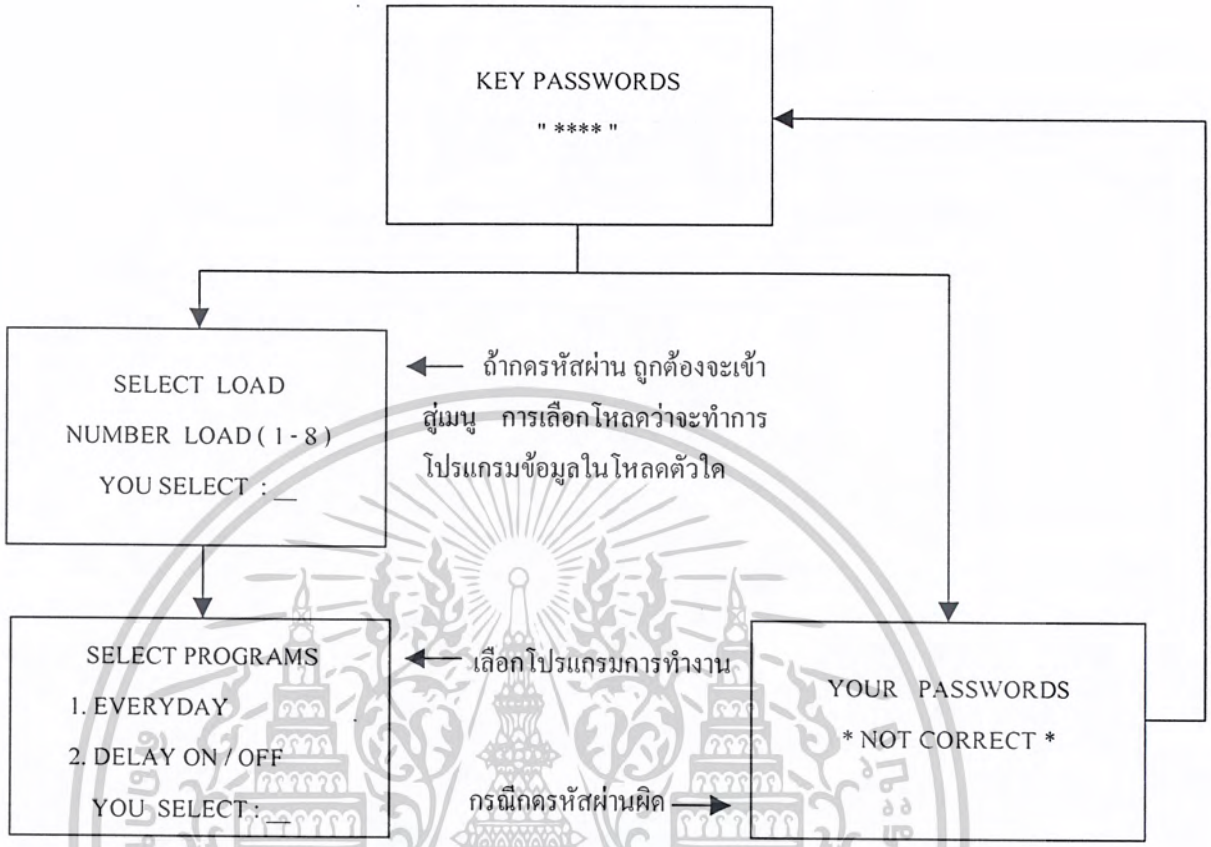


รูปที่ 4.2 การแสดงผลในสภาวะการทำงานปกติ

5. เมื่อต้องการตั้งค่าโปรแกรมการใช้งานสามารถทำได้ดังนี้

เลื่อนสวิตซ์ไปยังตำแหน่งPRO(โปรแกรมอัตโนมัติ) จะเข้าสู่การใส่รหัสผ่าน (Passwords)* ถ้าหากค่าไม่ถูกต้องหน้าจอจะฟ้อง และกลับไปสู่การใส่ค่ารหัสผ่าน* อีกครั้ง เมื่อใส่รหัสผ่าน* ที่ถูกต้องแล้วโปรแกรมจะทำงานต่อไปโดยจะเข้าสู่เมนูการเลือกโหลด และ โปรแกรมที่ต้องการ โดยสามารถเลือกโปรแกรมได้ 2 แบบ คือ โปรแกรมทำงานตามเวลาที่ตั้งของทุกวัน (Everyday) และ โปรแกรมหน่วงเวลา (Delay) ดังรูปที่ 4.3

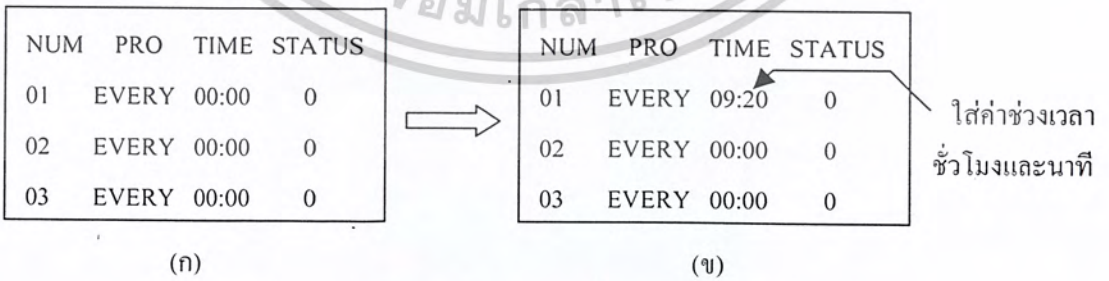
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.3 ผังการตั้งค่าโปรแกรม

1. กรณีเลือก โปรแกรม Everyday (กด 1)

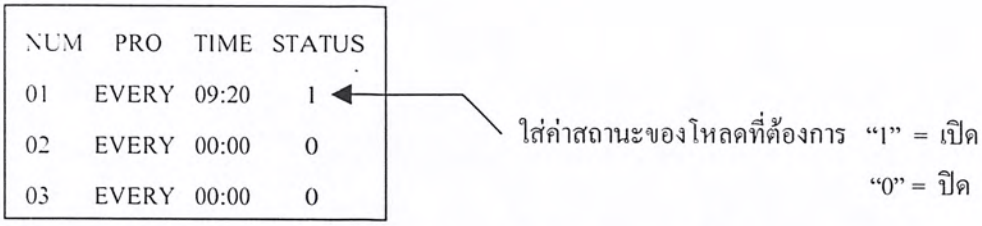
หน้าจอจะแสดงดังรูปที่ 4.4 (ก) จากนั้นใส่ค่าช่วงเวลาที่ต้องการสั่งงานโหลดเป็น ชั่วโมง และนาที เช่น ต้องการให้ โหลดเริ่มทำงานที่เวลา 09.20 ก็จะใส่ได้ตามรูปที่ 4.4 (ข)



รูปที่ 4.4 แสดงการใส่ค่าช่วงเวลาที่ต้องการสั่งงานโหลด

หลังจากนั้นตามด้วยการใส่ค่าสถานะของโหลดที่ต้องการ "1" = เปิด , "0" = ปิด ดังรูปที่ 4.5 (สถานะ : เปิด = จ่ายไฟ , ปิด = หยุดจ่ายไฟ)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.5 แสดงการใส่ค่าสถานะของโหลด

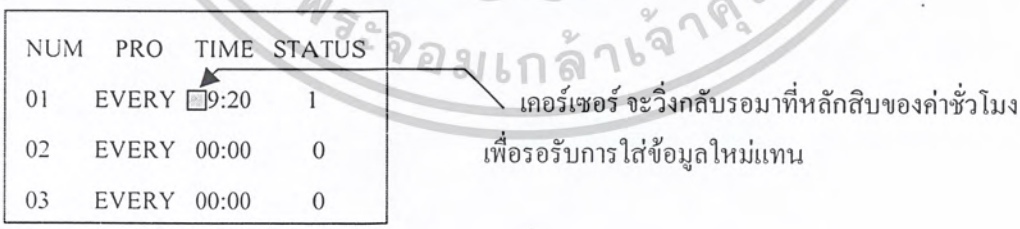
หลังจากใส่สถานะของโหลดแล้ว ตัวเคอร์เซอร์จะเลื่อนมากระพริบรออยู่หลังตัวเลขสถานะ ที่ได้ใส่เข้าไปดังรูปที่ 4.6 เพื่อรอการใส่โปรแกรมช่วงถัดไป

รูปที่ 4.6 แสดงเคอร์เซอร์ที่รอการใส่ค่าตัวต่อไป

การโปรแกรมในช่วงเวลาถัดไปนั้นจะใช้การเลือกปุ่มใดปุ่มหนึ่งใน 3 ปุ่มบนคีย์แพด

- ปุ่ม CLEAR

เป็นปุ่มที่ใช้สำหรับการแก้ไขค่าช่วงเวลาที่ได้ใส่ไว้ในช่วงเวลานั้นๆ เมื่อกดปุ่มนี้เคอเซอร์จะวิ่งกลับรอมาที่หลักสิบของค่าชั่วโมง เพื่อรอการใส่ข้อมูลใหม่เข้าไปแทนที่ข้อมูลเดิม ดังรูปที่ 4.7



รูปที่ 4.7 แสดงการใช้ปุ่ม CLEAR

เช่น หากเราตั้งให้ช่วงเวลาแรก เปิดเวลา 9.20 น. แล้ว ต้องการจะเปลี่ยนไปให้เปิดที่ เวลา 9.30 น. ก็ทำการกดปุ่ม CLEAR เคอร์เซอร์จะวิ่งมาที่ตำแหน่งชั่วโมง จากนั้นก็สามารถใส่ค่า 9.30 และสถานะ 1 ลงไปใหม่ได้เลย ดังรูปที่ 4.8

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

NUM	PRO	TIME	STATUS
01	EVERY	09:30	1
02	EVERY	00:00	0
03	EVERY	00:00	0

รูปที่ 4.8 แสดงการใส่ค่าข้อมูลใหม่หลังจาก กดปุ่ม CLEAR

- ปุ่ม ENTER

เป็นปุ่มที่ใช้สำหรับการใส่ค่าช่วงเวลาการทำงานในรอบถัดไป เมื่อกดปุ่มนี้เคอร์เซอร์จะมาเริ่มที่ ค่าชั่วโมงของช่วงเวลาถัดไป รูปที่ 4.9 (ก) เช่น เมื่อต้องการเปิดโหลดเวลา 09.30 และปิดเวลา 10.30 สามารถใส่ค่าเวลา ดังรูปที่ 4.9 (ข)

NUM	PRO	TIME	STATUS
01	EVERY	09:30	1
02	EVERY	0:00	0
03	EVERY	00:00	0

(ก)

NUM	PRO	TIME	STATUS
01	EVERY	09:30	1
02	EVERY	10:30	0
03	EVERY	00:00	0

(ข)

รูปที่ 4.9 แสดงการใช้ปุ่ม ENTER

- ปุ่ม 2ND FUN

เป็นปุ่มที่ใช้สำหรับการออกจากการตั้งค่าเวลาในแต่ละโหลด เมื่อกดปุ่มนี้แล้วจะออกจากการตั้งค่าโหลดนั้นๆ แต่ก่อนที่จะกดปุ่มนี้จะต้องใส่ค่า ชั่วโมง ,นาที และ สถานะ เป็น (99:99 9) เพื่อแสดงว่าจบการตั้งค่า แล้วจึงกดปุ่มนี้ เช่น จากรูปที่ 4.10 ต้องการตั้งค่าเพียง 2 ช่วงเวลาในช่วงเวลาที่ 3 ให้ใส่ 9 ลงในค่าชั่วโมง ,นาที และสถานะ แล้วจึงกดปุ่ม 2ND FUN

NUM	PRO	TIME	STATUS
01	EVERY	09:30	1
02	EVERY	10:30	0
03	EVERY	99:99	9

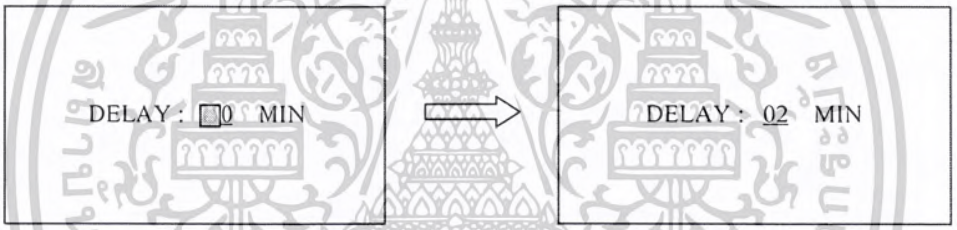
รูปที่ 4.10 แสดงการจบการใส่ค่าในแต่ละโหลด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อออกจากการตั้งค่าแล้วให้เลื่อนสวิตซ์กลับมาที่ตำแหน่ง RUN หน้าจอจะกลับสู่สถาวะการทำงานปกติ และเครื่องก็จะทำงานตามเวลาที่ได้ตั้งไว้ในโหลดนั้นๆที่ได้ตั้งค่าไว้ เป็นการจบการตั้งค่าในหนึ่งโหลด หากต้องการตั้งค่าโหลดอื่นอีกให้ทำตามขั้นตอนเดิม (เริ่มจากข้อ 5)

2. กรณี เลือกโปรแกรม Delay (กด 2)

หน้าจอจะแสดงผลดังรูปที่ 4.11 (ก) เพื่อรอรับค่า Delay ที่ต้องการจากผู้ใช้ได้ตั้งแต่ 01 นาทีจนถึง 99 นาที โดยค่า Delay ทีใส่เข้าไปจะมีการทำงานทำงานตามที่ได้ตั้งไว้ หลังจากที่ผู้ใช้ออกจากโปรแกรม Delay แล้ว เช่น ใส่ DELAY 02 MIN ดังรูปที่ 4.11 (ข) เมื่อออกจากโปรแกรมหลังจากนั้น 2 นาที (หมายเหตุ : ดูการเปลี่ยนแปลงในหลักนาที) โหลดก็จะ เปิด เป็นเวลา 2 นาที ตามด้วยการ ปิด อีก 2 นาทีโดยจะทำงานสลับกัน เปิด/ปิด เช่นนี้ไปเรื่อย ๆ จนกว่าจะมีการใส่ค่าใหม่เข้าไปแทนที่ หรือเปลี่ยนไปทำงานในแบบ Everyday

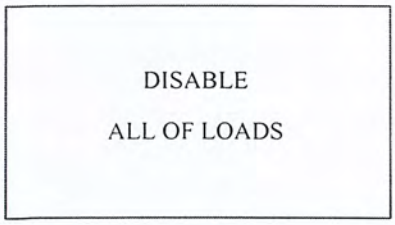


รูปที่ 4.11 หน้าจอของโปรแกรมหน่วงเวลา

6. การควบคุมโหลดแบบฉุกเฉินทำได้ 2 แบบ

1 แบบปิดโหลดทั้งหมด (Dissble All Of Loads)

ทำได้โดยการเลื่อนสวิตซ์จากตำแหน่ง AUTO ไปยังตำแหน่ง DIS โดยโหลดทั้งหมดจะปิดการทำงานทั้งหมดทันทีและแสดงผลบนหน้าจอ ดังรูปที่ 4.12 และ เมื่อเลื่อนสวิตซ์กลับมาที่ตำแหน่ง AUTO โหลดทั้งหมดก็กลับมาทำงานตามสภาวะเวลาเดิมที่ได้ตั้งไว้

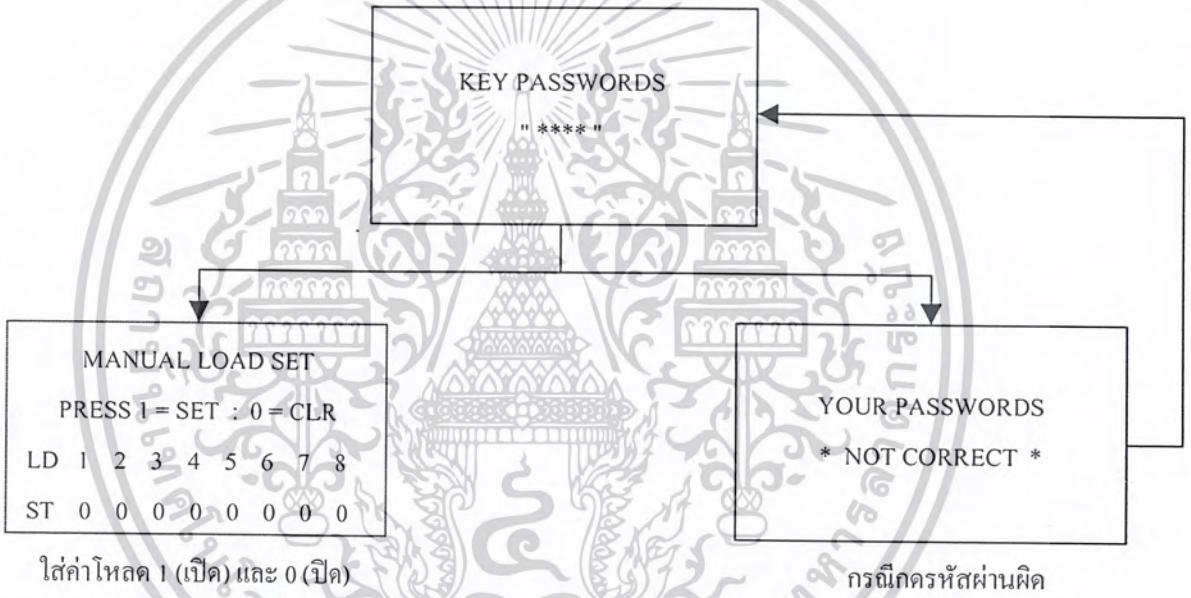


รูปที่ 4.12 หน้าจอกรณีที่มีการสั่งงานให้ปิดโหลดทั้งหมด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. แบบสั่งงานโดยตรง (Manual)

ทำได้โดยการเลื่อนสวิตซ์จากตำแหน่ง AUTO ไปยังตำแหน่ง MAN แล้วจะเข้าสู่การใส่รหัสผ่าน* ถ้าหากค่าไม่ถูกต้องหน้าจอจะฟ้อง และกลับไปสู่การใส่รหัสผ่าน* อีกครั้ง เมื่อใส่รหัสผ่าน* ที่ถูกต้องแล้วหน้าจอจะแสดง ดังรูปที่ 4.13 โดยที่ผู้ใช้สามารถ เปิด/ปิดโหลดนั้นได้ทันที โดยกด 1 (เปิด) และกด 0 (เปิด) และค่าสถานะที่โหลดบนจอ(บรรทัดสุดท้ายบนจอ)จะแสดงสภาวะโหลดปัจจุบันด้วย การใส่ค่าจะต้องใส่เรียงลำดับจากโหลด 1 ถึง โหลด 8 และเมื่อต้องการออกจากโปรแกรมนี้เลื่อนสวิตซ์กลับมาที่ตำแหน่ง AUTO (ก่อนเลื่อนสวิตซ์มาที่ตำแหน่ง AUTO จะต้องใส่สถานะครบทั้ง 8 โหลด ก่อนจึงสามารถออกจากโปรแกรมได้) หน้าจอจะกลับมาที่สถาวะการทำงานปกติ

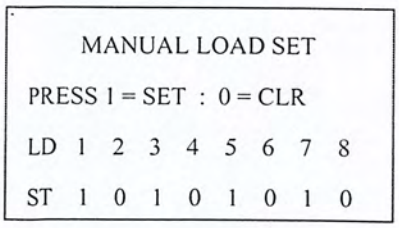


ใส่ค่าโหลด 1 (เปิด) และ 0 (ปิด)

กรณีกดรหัสผ่านผิด

รูปที่ 4.13 ผังการใช้งานโปรแกรมแบบสั่งงานโดยตรง

ตัวอย่างการใช้งานของโปรแกรมนี้ คือหากเราต้องการให้โหลด 1,3,5,7 เปิดก็จะใส่ค่าโดยการใส่ค่าเรียงลำดับจากโหลด 1- 8 ดังนี้ 1 0 1 0 1 0 1 0 หน้าจอจะแสดงดังรูปที่ 4.14 และเครื่องก็จะทำงานตามสถานะที่ตั้งไว้

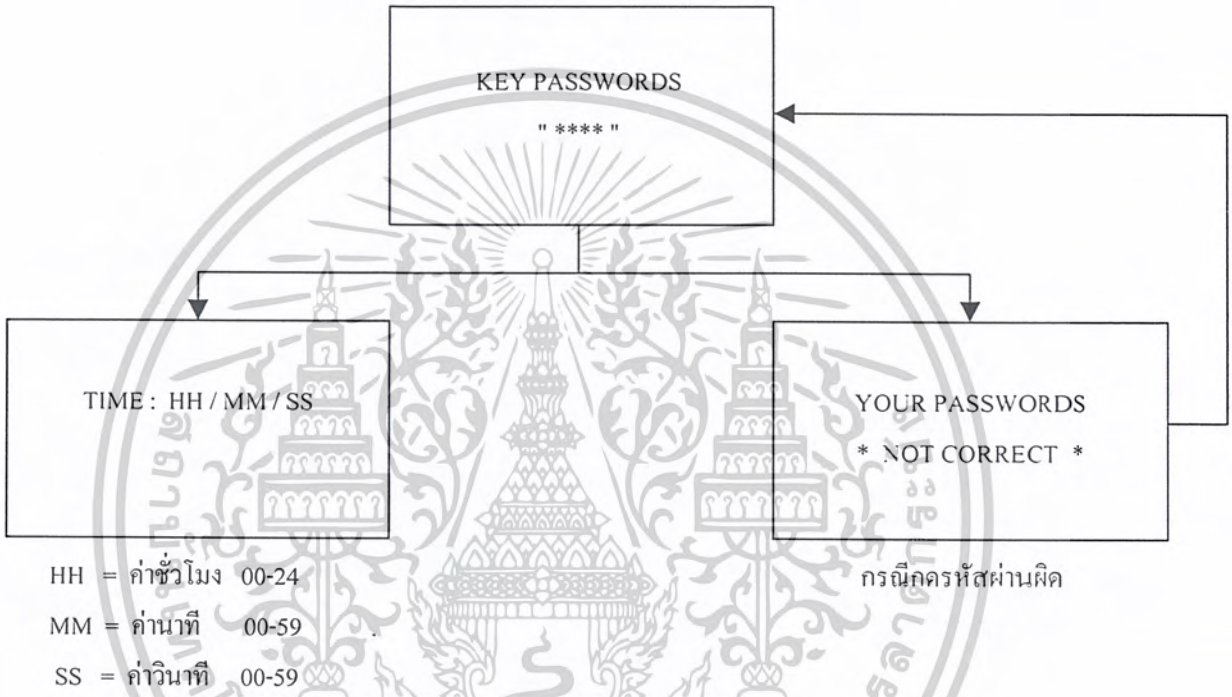


รูปที่ 4.14 ตัวอย่างการตั้งค่าแบบสั่งงานโดยตรง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

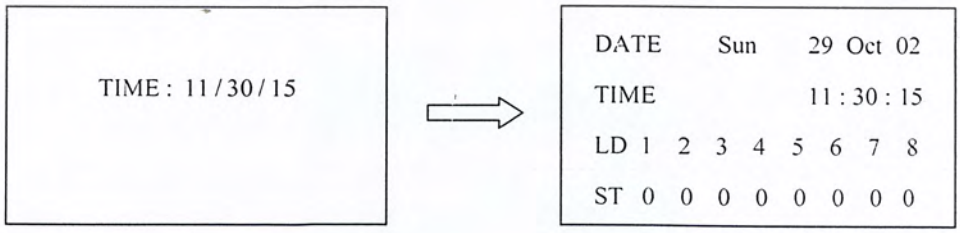
7. การตั้งค่าเวลา (เลื่อนสวิตซ์ไปที่ตำแหน่ง TIME)

เมื่อเลื่อนสวิตซ์ไปที่แหน่ง TIME แล้ว จะเข้าสู่การใส่รหัสผ่าน* ถ้าหากใส่ไม่ถูกต้องหน้าจอจะฟ้อง และกลับไปสู่การใส่รหัสผ่าน* อีกครั้ง เมื่อใส่ รหัสผ่าน* ที่ถูกต้องแล้ว สามารถใส่ค่าชั่วโมง(HH) , นาที(MM) , วินาที(SS) ดังรูปที่ 4.15 แล้วให้เลื่อนสวิตซ์กลับมาที่ตำแหน่ง RUN ก็จะกลับมาสู่การทำงาน เดิมหน้าจอจะแสดงเวลาตามเวลาที่ตั้งเข้าไปใหม่



รูปที่ 4.15 ผังการตั้งค่าเวลา

เช่น ต้องการตั้งเวลาที่ 11 นาฬิกา 30 นาที 15 วินาที ใส่ดังรูปที่ 4.16 (ก) ในกรณีที่ใส่ค่าผิดให้กดสวิตซ์ 2ND FUN เพื่อเข้าสู่การตั้งค่าเวลาใหม่ เมื่อได้ค่าเวลาที่ต้องการแล้วก็เลื่อนสวิตซ์กลับมาที่ตำแหน่ง RUN แล้ว หน้าจอจะแสดงค่าที่ตั้งเข้าไปใหม่โดยทันที รูปที่ 4.16 (ข)



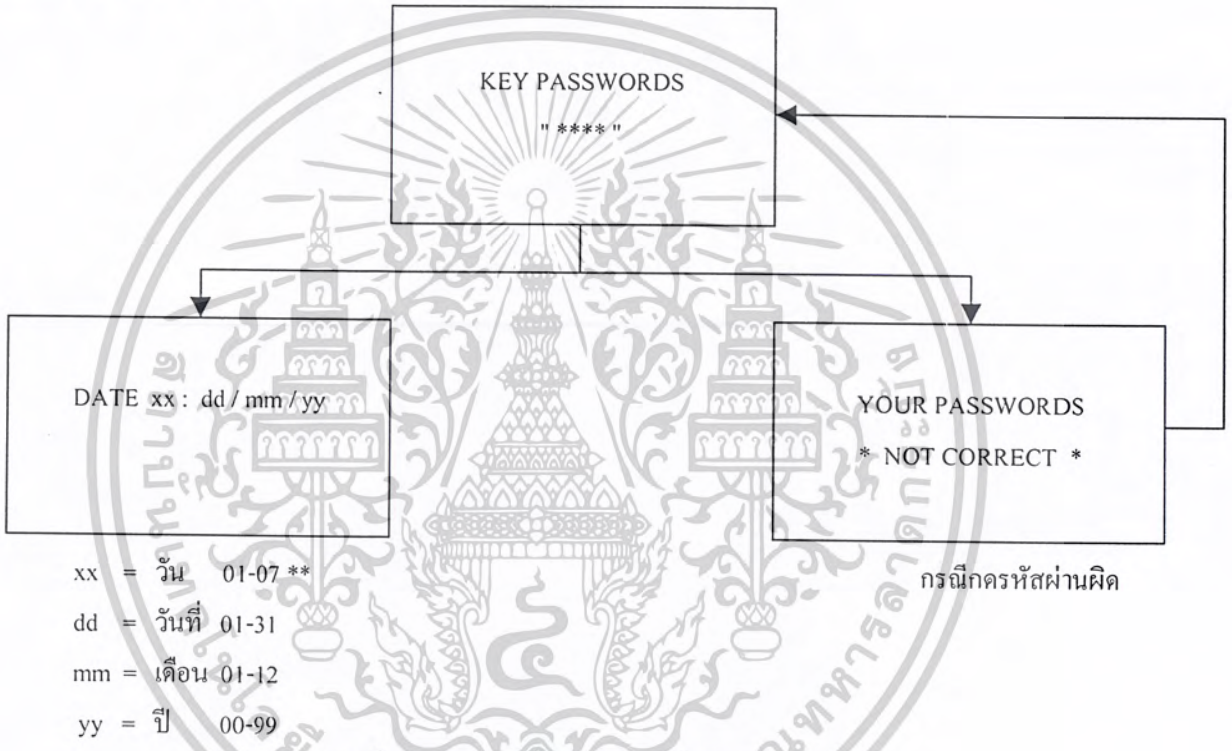
(ก) (ข)

รูปที่ 4.16 ตัวอย่างการตั้งค่าเวลา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

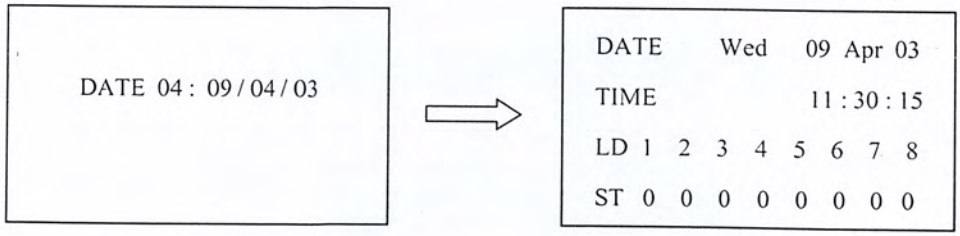
8. เมื่อต้องการตั้งค่า วัน,วันที่,เดือน,ปี (กดสวิตซ์ DATE)

เมื่อกดสวิตซ์ DATE ค้างไว้สักครู่ จะเข้าสู่การใส่รหัสผ่าน* ถ้าใส่ไม่ถูกต้องหน้าจอจะฟ้องและกลับไปสู่การใส่รหัสผ่าน* อีกครั้ง เมื่อใส่รหัสผ่าน* ที่ถูกต้องแล้ว สามารถใส่ค่า วัน(xx), วันที่(dd) ,เดือน(mm), ปี(yy) ดังรูปที่ 4.17 เมื่อใส่ค่าครบแล้วหน้าจอก็จะกลับเข้าสู่การทำงานปกติ และจะแสดงค่าที่ตั้งเข้าไปใหม่ทันที กรณีเกิดความผิดพลาดในการตั้งค่าก่อนจะใส่ค่าครบทั้งหมด ให้กดปุ่ม 2ND FUN เพื่อออกจากการตั้งค่า หากจะทำการแก้ไขใหม่ก็เพียงกดสวิตซ์ DATE อีกครั้งแล้วทำตามขั้นตอนเดิม



รูปที่ 4.17 ผังการ ตั้งค่า วัน,วันที่,เดือน,ปี

เช่น ต้องการตั้งค่าเป็น วันพุธที่ 9 เมษายน 2003 ใส่ค่าดังรูปที่ 4.18 (ก) เมื่อใส่ค่าครบแล้ว หน้าจอจะกลับไปที่สถาวะการทำงานเดิมและจะแสดงค่าที่ตั้งเข้าไปใหม่ทันที ดังรูปที่ 4.18 (ข)



(ก) (ข)

รูปที่ 4.18 ตัวอย่างการตั้งค่า วัน,วันที่,เดือน,ปี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หมายเหตุ 1.เมื่อไม่การกดคีย์แพด เป็นเวลา 10 วินาที LED Blacklight บนหน้าจอจะดับ

2. * รหัสผ่าน (Passwords) ของเครื่องฯ ที่ได้โปรแกรมไว้คือ 1 2 3 4

3. ** ค่าตัวแปร xx ของเครื่องฯ นี้คือ

01 = Sun

02 = Mon

03 = Tue

04 = Wed

05 = Thu

06 = Fri

07 = Set

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5
บทวิจารณ์และสรุป

5.1 สรุปผลการทดลอง

เครื่องควบคุมอุปกรณ์ไฟฟ้าภายในบ้านแบบโปรแกรมได้ที่สร้างขึ้นมานี้ ได้ทำการปรับปรุงแก้ไขข้อเสียและข้อบกพร่องของเครื่องควบคุมอุปกรณ์ไฟฟ้าภายในบ้านแบบโปรแกรมได้ตัวเดิม(ปีการศึกษา 2544) โดยเริ่มต้นตั้งแต่การศึกษาข้อเสียหรือข้อบกพร่องของเครื่องควบคุมอุปกรณ์ไฟฟ้าภายในบ้านแบบโปรแกรมได้ตัวเดิม แล้วจึงทำการออกแบบระบบทั้งทางฮาร์ดแวร์และซอฟต์แวร์ใหม่ทั้งหมด ในส่วนของฮาร์ดแวร์นั้นซึ่งได้ใช้ตัวไมโครคอนโทรลเลอร์ตระกูล MCS-51 เป็นตัวประมวลผลหลัก ซึ่งจำเป็นอย่างยิ่งที่ผู้ออกแบบวงจรและผู้เขียนโปรแกรมจะต้องเข้าใจการใช้งานในการติดต่อส่งข้อมูลอย่างลึกซึ้ง เนื่องจากการทำงานของเครื่องควบคุมอุปกรณ์ไฟฟ้าภายในบ้านแบบโปรแกรมได้ตัวนี้ ใช้การเชื่อมต่อระหว่างตัวประมวลผลหลักกับอุปกรณ์อื่น เป็นจำนวนมากไม่ว่าจะเป็นการเชื่อมต่อกับจอแสดงผล (LCD) ,ไอซี DS1307 , AD , สวิตซ์อินพุต , คีย์แพด ซึ่งจะต้องมีการทำงานที่สัมพันธ์กันเป็นอย่างดีกับโปรแกรมที่ได้เขียนไว้ โดยการออกแบบทั้งหมดนั้นควรจัดทำบนแผงทดลองโปรโตบอร์ดก่อน เพราะถ้าหากมีการแก้ไขเปลี่ยนแปลงทางฮาร์ดแวร์จะสามารถทำได้ง่าย และเมื่อทำการเขียนซอฟต์แวร์ควบคุมการทำงานทางฮาร์ดแวร์ ได้ถูกต้องตามเงื่อนไขที่ได้ออกแบบไว้แล้ว จึงจะนำไปทำเป็นแผ่นวงจรพิมพ์(PCB)เพื่อติดตั้งลงกล่อง จากนั้นจึงเป็นการทดสอบการทำงานอีกครั้ง โดยถ้าหากการทำงานของวงจรมีปัญหา ควรจะแยกตรวจสอบเป็นส่วนๆ เพื่อง่ายต่อการหาจุดผิดพลาดที่เกิดขึ้น

5.2 ปัญหาที่เกิดขึ้นจากการทดลอง

ปัญหาที่เกิดขึ้นจะมีทั้งทางด้าน ฮาร์ดแวร์ และ ซอฟต์แวร์ ในส่วนของฮาร์ดแวร์นั้น ผู้ออกแบบวงจรควรจะวางแผนการจัดการเชื่อมต่ออุปกรณ์ต่างๆเข้ากับตัวประมวลผลให้เหมาะสมตั้งแต่แรก เพราะถ้าหากเกิดความผิดพลาดในส่วนนี้ขึ้นแล้ว จะเป็นการยากลำบากในการแก้ไขหลังจากได้ทำแผ่นวงจรพิมพ์ (PCB)ไปแล้ว

และในส่วนของซอฟต์แวร์นั้น ผู้เขียนโปรแกรมจะต้องคำนึงถึง รอบ(loop) การทำงานภายในโปรแกรมต่างๆให้ดี เพราะว่าอาจจะเกิดปัญหาเกี่ยวกับการซ้อนทับกันของสเต็ก จนอาจทำให้เกิดการค้าง(hang) ของระบบการทำงานของไมโครคอนโทรลเลอร์ขึ้นได้ จึงควรจัดสรรพื้นที่การใช้งานในหน่วยความจำของ รอม และ แรม ให้เหมาะสมตั้งแต่เริ่มแรกเช่นกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

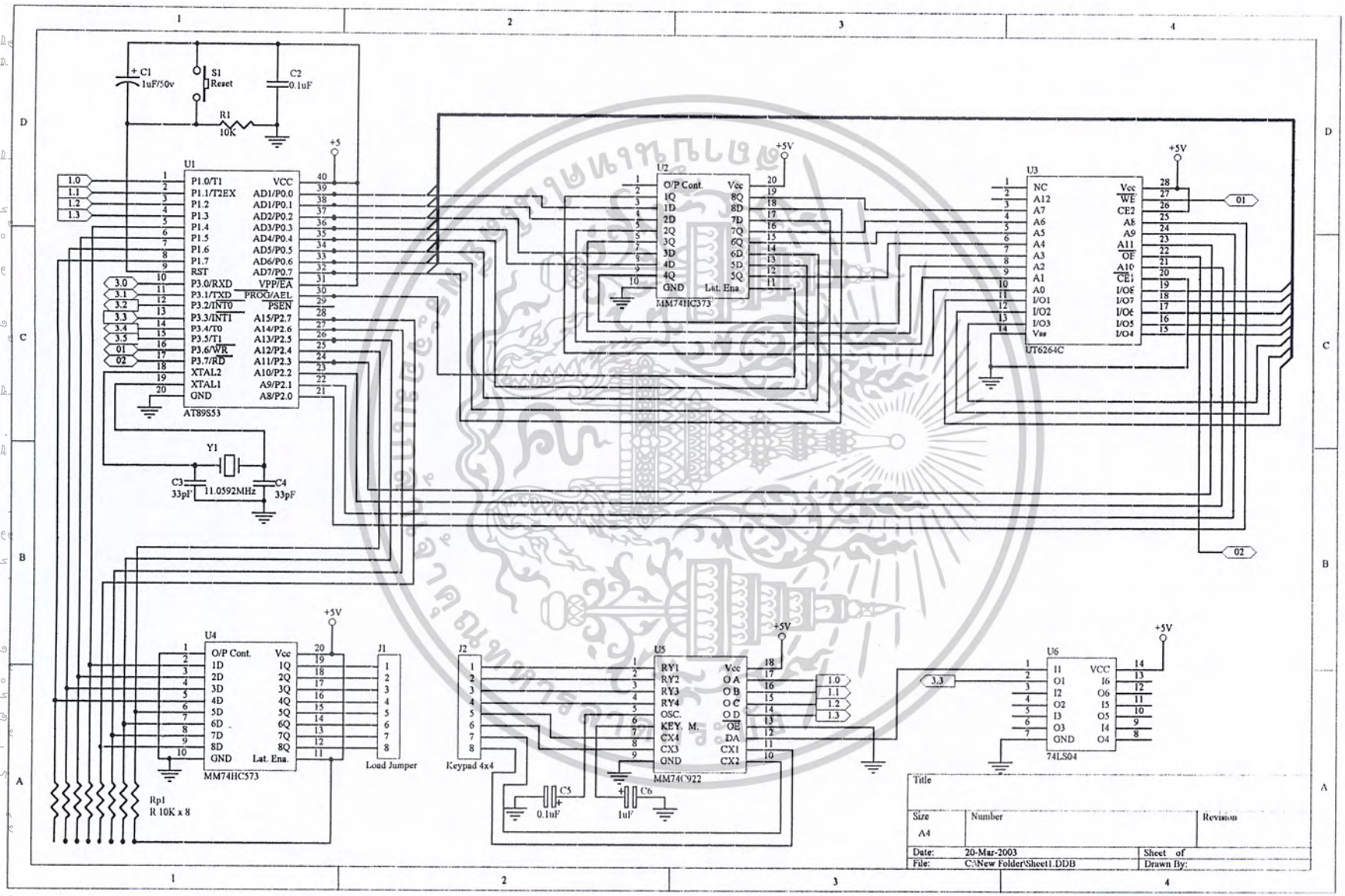
ภาคผนวก

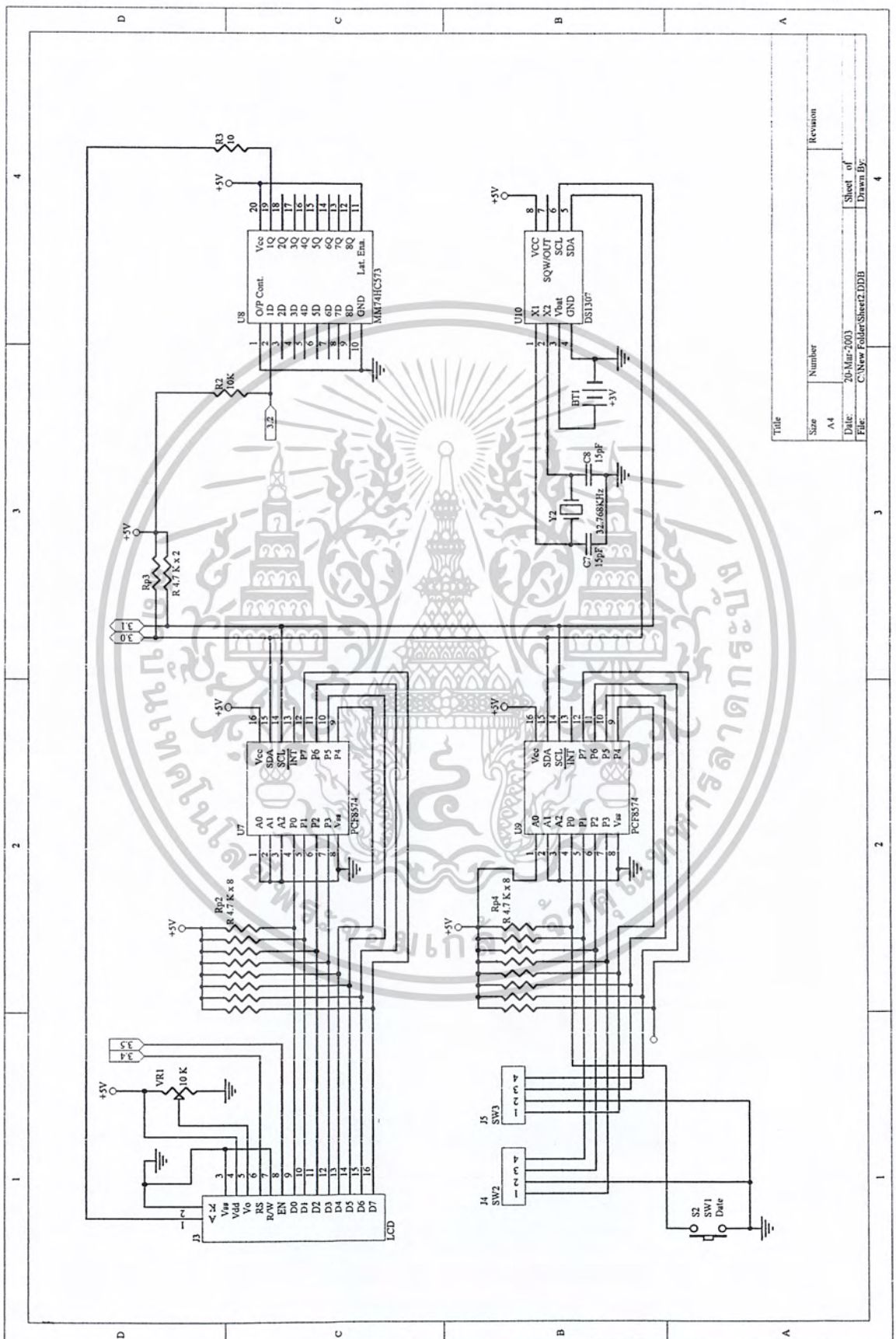
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ฮาร์ดแวร์

- วงจรและลายวงจรของแผงวงจรควบคุมหลัก
- วงจรและลายวงจรของแหล่งจ่าย
- วงจรและลายวงจรของส่วนขับโหลด
- รูปรายละเอียดของชิ้นงาน (เครื่องฯ)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้


C1
1uF/50v
S1
Reset
C2
0.1uF
R1
10K
+5
U1
AT89S53
P1.0/T1
P1.1/T2EX
P1.2
P1.3
P1.4
P1.5
P1.6
P1.7
RST
P3.0/RXD
P3.1/TXD
P3.2/INT0
P3.3/INT1
P3.4/T0
P3.5/T1
P3.6/WR
P3.7/RD
XTAL2
XTAL1
GND
VCC
AD1/P0.0
AD1/P0.1
AD2/P0.2
AD3/P0.3
AD4/P0.4
AD5/P0.5
AD6/P0.6
AD7/P0.7
VPP/EA
PROG/AEL
PSEN
A15/P2.7
A14/P2.6
A13/P2.5
A12/P2.4
A11/P2.3
A10/P2.2
A9/P2.1
A8/P2.0
Y1
C3
33pF
11.0592MHz
C4
33pF
U2
MM74HC373
O/P Cont.
Lat. Ena.
U3
UT6264C
01
02
U4
MM74HC573
Load Jumper
J1
J2
Keypad 4x4
U5
MM74C922
KEY. M.
OSC.
U6
74LS04
3.3
Rp1
R 10K x 8
C5
0.1uF
C6
1uF
Title
Size
A4
Number
Revision
Date:
20-Mar-2003
Sheet of
File:
C:\New Folder\Sheet1.DDB
Drawn By:

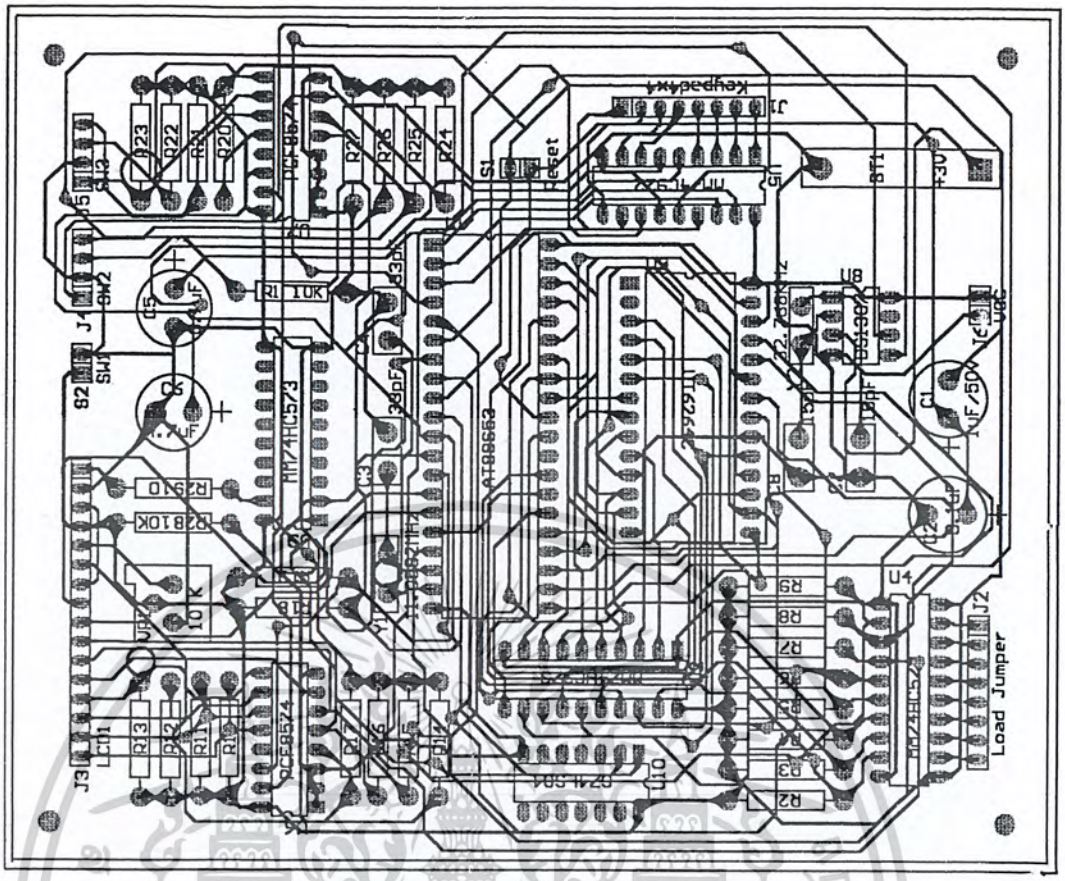


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

(ก)

(ข)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



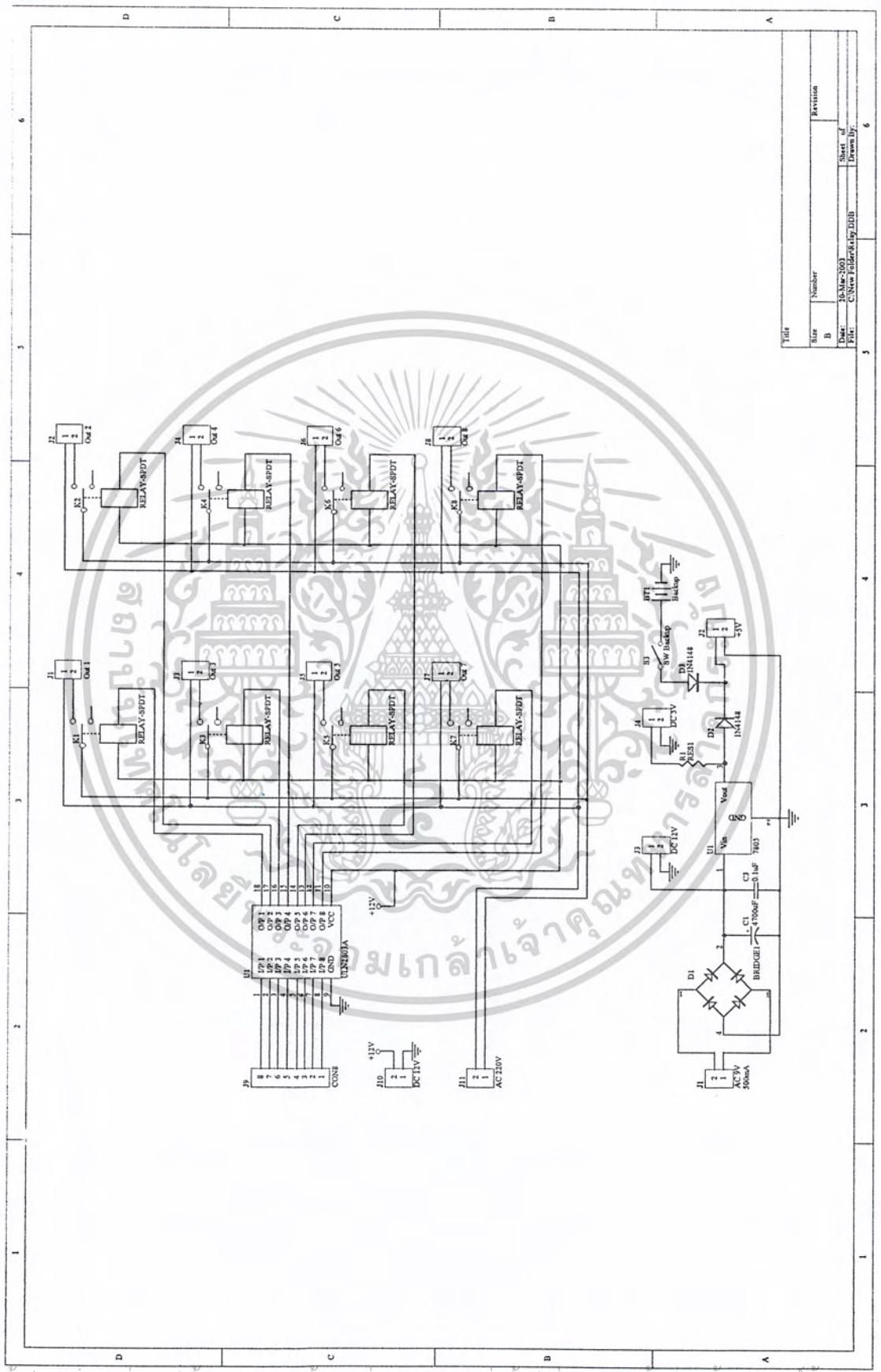
(ค)

แผงวงจรควบคุมหลัก (ก) ลายวงจรด้านบน

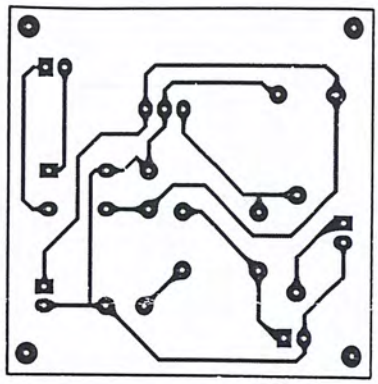
(ข) ลายวงจรด้านล่าง

(ค) การลงอุปกรณ์

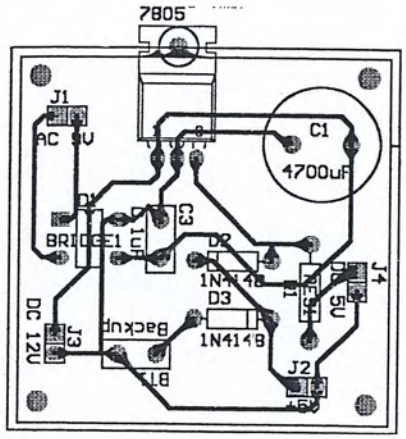
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



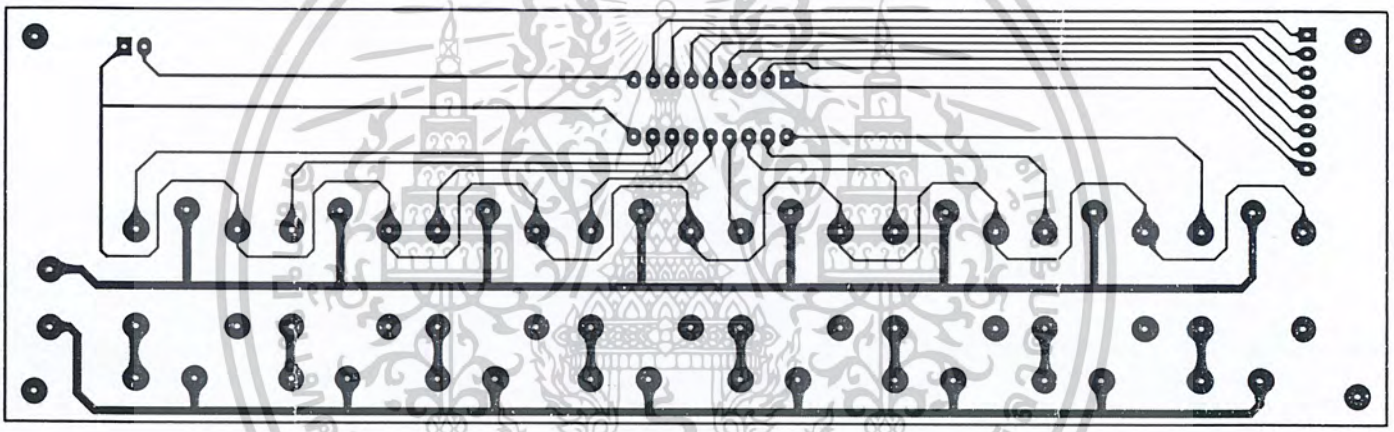
(ก)



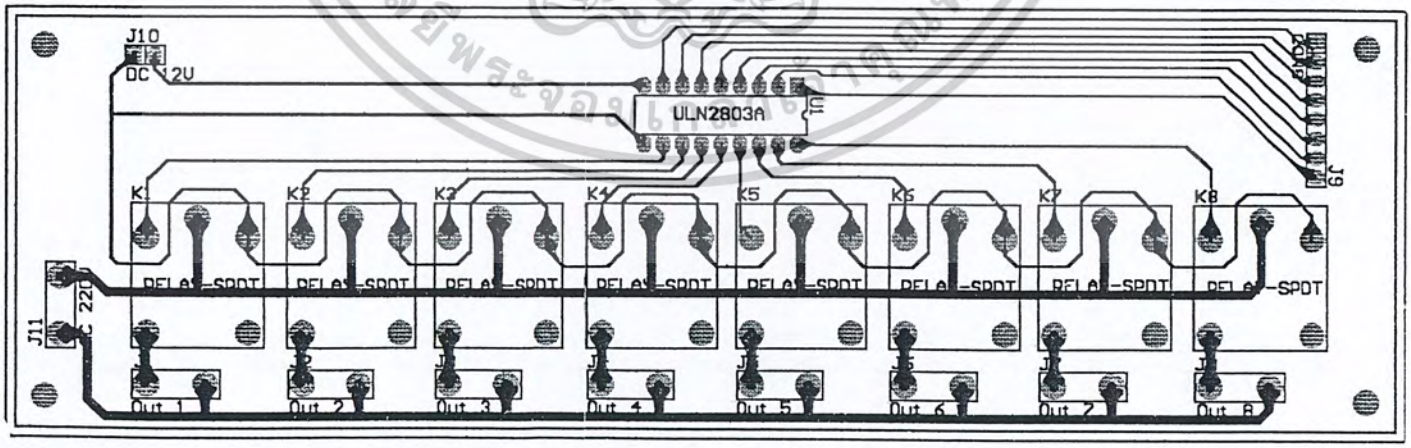
(ข)

แหล่งจ่ายไฟ (ก) ลายวงจร

(ข) การลงอุปกรณ์



(ก)



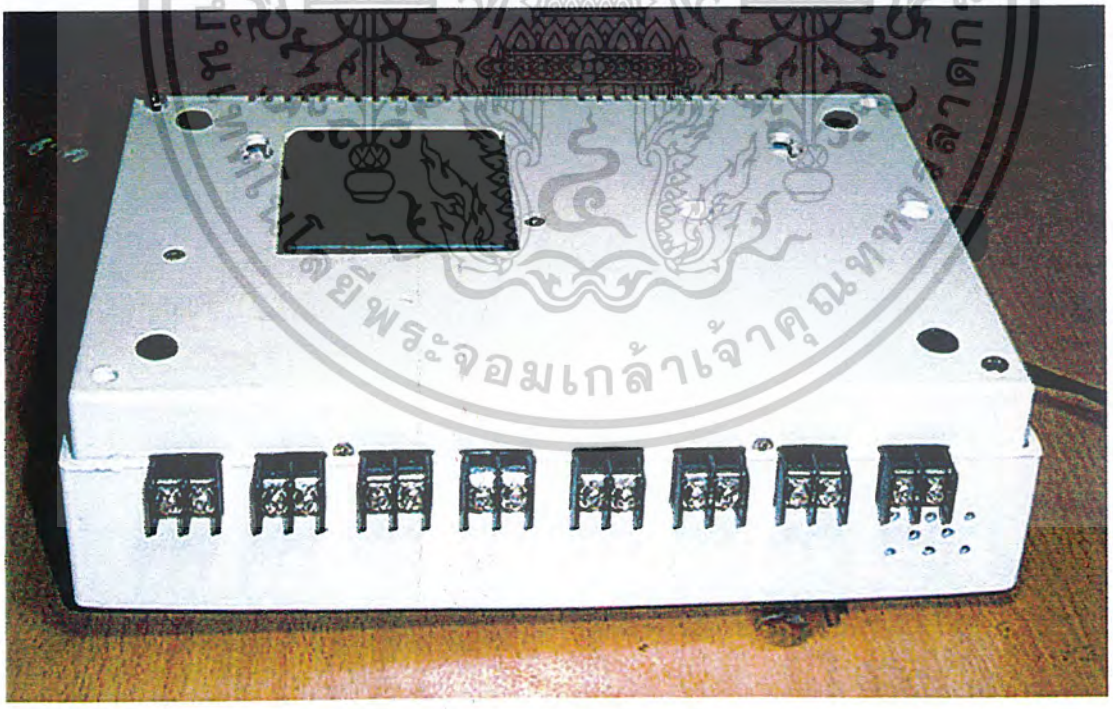
(ข)

ภาคเอาต์พุต (ก) ลายวงจร

(ข) การลงอุปกรณ์

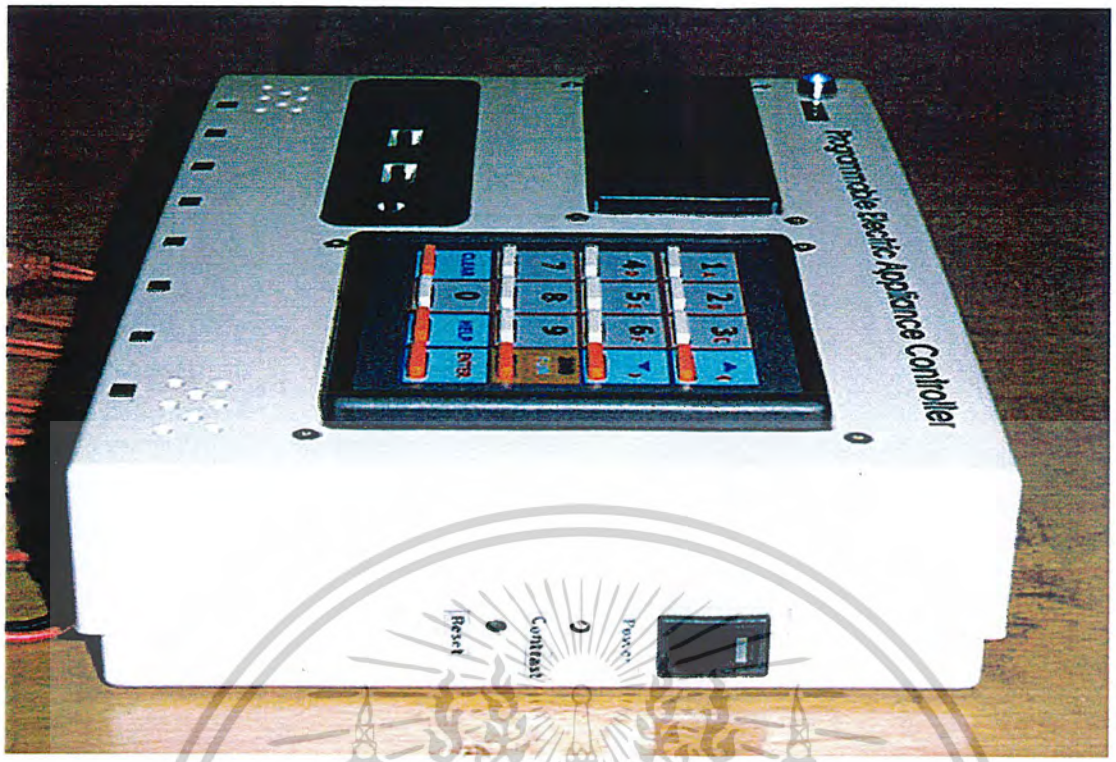
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ด้านบน สวิตซ์ควบคุม , Keypad , จอแสดงผล และ ด้านหน้า จุดต่อโหลด

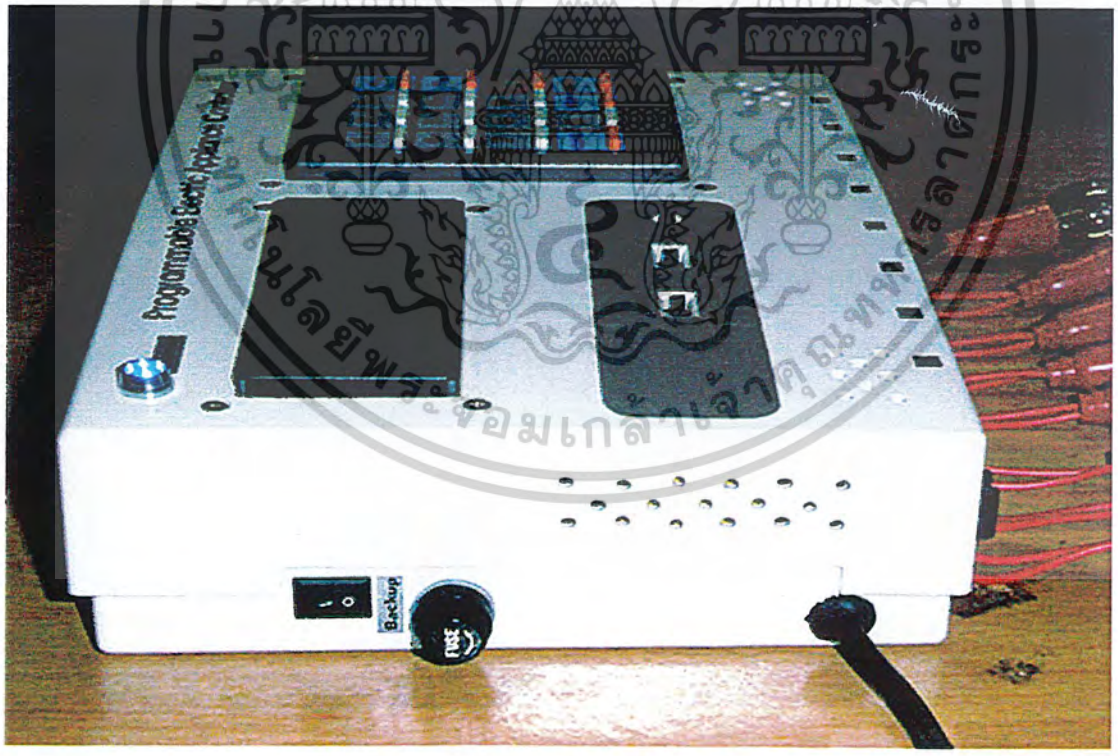


ด้านล่าง จุดใส่ Battery Backup

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

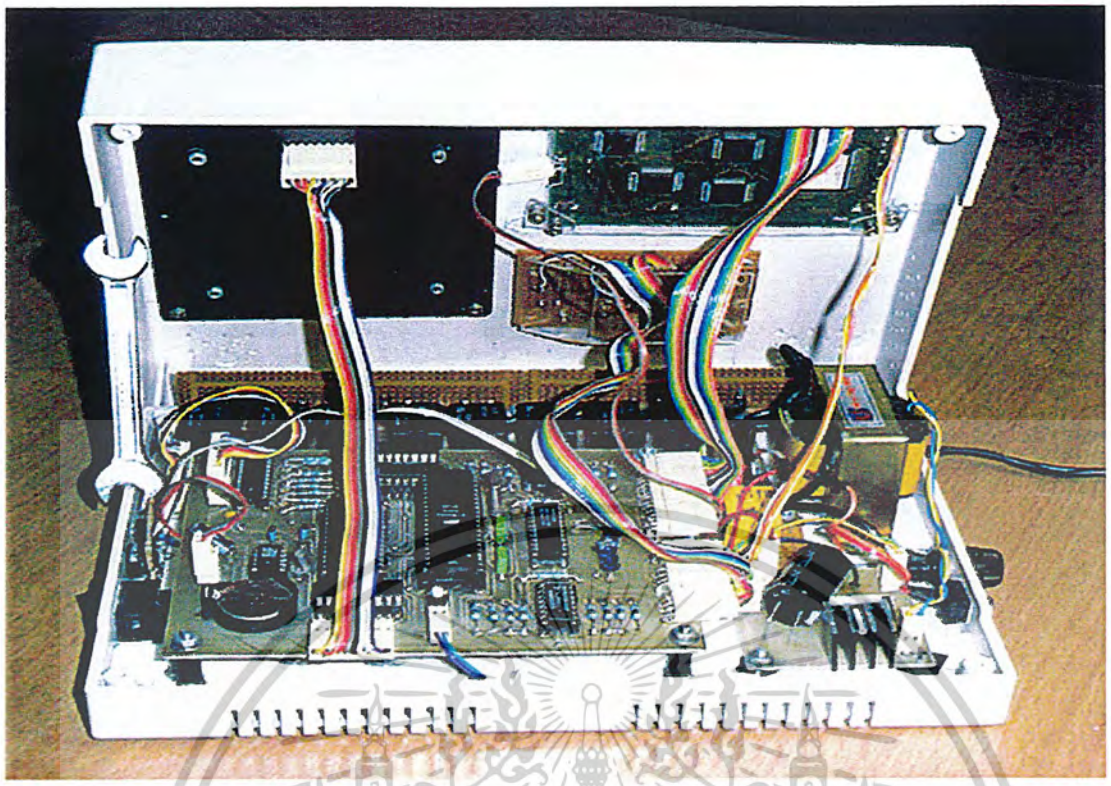


ด้านซ้าย สวิตซ์ ปิด/เปิด Power , ปรับความเข้มหน้าจอ และ ปุ่ม Reset

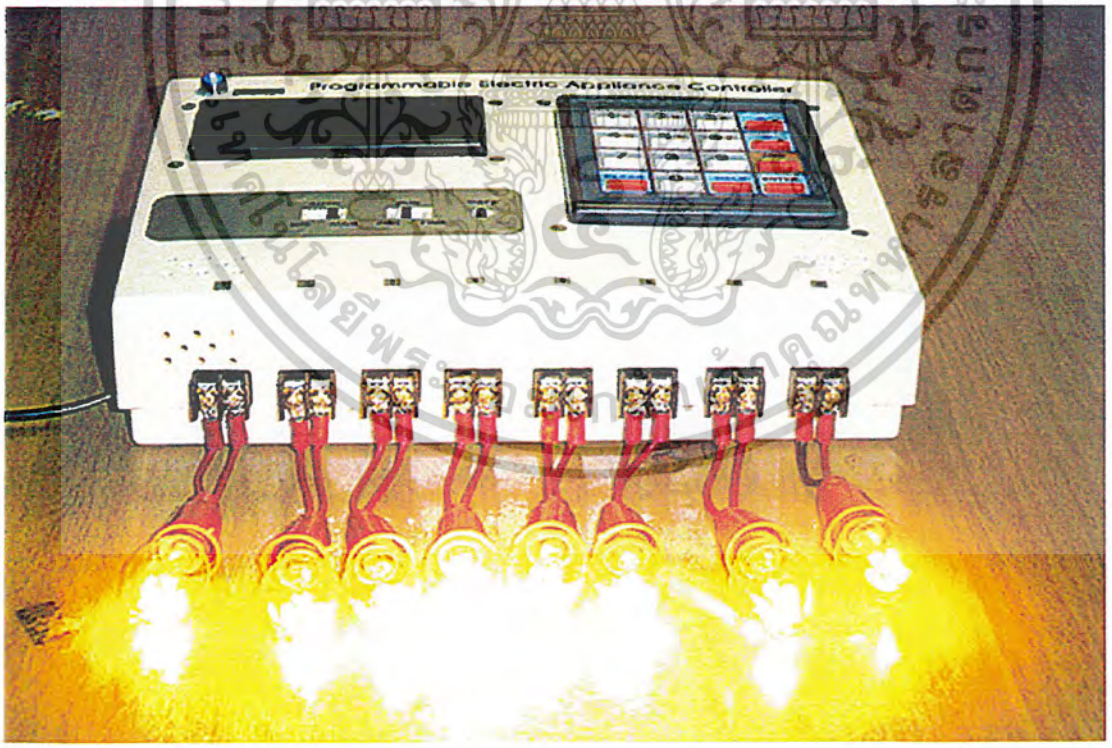


ด้านขวา สวิตซ์ ปิด/เปิด Backup , ฟิวส์ และสายไฟ AC

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภายใน การติดตั้งอุปกรณ์ต่างๆ



การต่อโหลด (จำลอง) จำนวน 8 โหลด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ซอฟต์แวร์

- โปรแกรมการทำงานของโครงงานนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
;*****************************************************
:PROGRAMMABLE ELECTRIC APPLIANCE CONTROLLER
;*****************************************************
SDA             BIT     P3.0
SCL             BIT     P3.1
CHKEY           BIT     P3.3
LCD_EN          BIT     P3.4
LCD_RS          BIT     P3.5
;*****************************************************
FLAG            EQU     02FH
I2C_ACK         BIT     FLAG.0
FLAG2           EQU     02EH
;*****************************************************
LCD_ADDR        EQU     030H
LCD_DATA        EQU     031H
LCD_PTR         EQU     032H
I2C_ADDR        EQU     033H
I2C_DATA        EQU     034H
KPAD_DATA       EQU     035H
BUFFER          EQU     036H
SECONDS         EQU     040H
MINUTES         EQU     041H
HOURS           EQU     042H
DAY             EQU     043H
DATE            QU      044H
MONTH           EQU     045H
YEAR            EQU     046H
CONTROL         EQU     047H
PL              EQU     048H
BUFHOUR         EQU     049H
BUFMINUTE       EQU     04AH
BUFSTATUS       EQU     04BH
TIME            EQU     050H
OUT             EQU     060H
HIGH1           EQU     061H
LOW1            EQU     062H
HIGH_NUM        EQU     063H
LOW_NUM         EQU     064H
BUFFER1         EQU     065H
SELECT_LO       EQU     066H
SELECT_PR       EQU     067H
START_H         EQU     068H
START_M         EQU     069H
DELAY_TIME      EQU     06AH
STOP_H          EQU     06BH
STOP_M          EQU     06CH
DP_H            EQU     06FH
DP_L            EQU     070H
IO_DATA         EQU     071H
COUNT1          EQU     072H
ADD_LCD1        EQU     073H
BUFLOW          EQU     074H
BUFHIGH         EQU     075H
BUFLOW2         EQU     076H
BUFHIGH2        EQU     077H
BUFLOW3         EQU     078H
BUFHIGH3        EQU     079H
COUNT2          EQU     07AH
PASSHIGH        EQU     07BH
PASSLOW         EQU     07CH
BUF_LOAD        EQU     07DH
ADD_ST          EQU     07EH
OLD_TIME        EQU     07FH
;*****************************************************
RTC_ID          EQU     11010000B
PCF8574_ID      EQU     01110000B
PCF_ID          EQU     01110010B
;*****************************************************
;               Main Program.
;*****************************************************
                ORG     0000H
                AJMP    INTI

                ORG     0013H
                AJMP    ON_LED

INTI:           LCALL   PCF_RD
                MOV     A,IO_DATA
                ANL     A,#0FH
                CJNE    A,#0BH,INTI

                MOV     PL,#00000000B
                LCALL   PCF8574_WR
                SETB    SDA
                SETB    SCL
                MOV     P1,#00001111B
                MOV     P3,#11001011B
                CLR     P2.4
                CLR     P2.5
                CLR     P2.6
                CLR     P2.7
                MOV     IE,#10000100B
                MOV     COUNT1,#00H
                MOV     FLAG2,#00H

MAIN:           LCALL   INIT_LCD

                MOV     LCD_ADDR,#040H
                LCALL   SET_ADDR_LCD
                MOV     DPTR,#TITLE_1
                LCALL   WRLINE_LCD
                MOV     LCD_ADDR,#014H
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                LCALL   SET_ADDR_LCD
                MOV     DPTR,#TITLE_2
                LCALL   WRLINE_LCD
                MOV     LCD_ADDR,#54H
                LCALL   SET_ADDR_LCD
                MOV     DPTR,#TITLE_3
                LCALL   WRLINE_LCD

                LCALL   DELAY_1s
                LCALL   DELAY_1s

                LCALL   LCD_CLR
                LCALL   KEEP_ST
                LCALL   BACK_ST

LOOP:           LCALL   RTC_RD

                MOV     LCD_ADDR,#000H
                LCALL   SET_ADDR_LCD
                MOV     DPTR,#DATE1
                LCALL   WRLINE_LCD2

                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_MON
                MOV     A,#00AH
                AJMP    WRITE_MONTH_NX
WR_CHK_MONTH_1: CJNE    A,#011H,WR_CHK_MONTH_2
                MOV     A,#00BH
                AJMP    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,#040H
                LCALL   SET_ADDR_LCD
                MOV     DPTR,#TIME1
                LCALL   WRLINE_LCD2

                MOV     LCD_ADDR,#04AH
                LCALL   SET_ADDR_LCD

                MOV     LCD_DATA,#' '
                LCALL   WRCHAR_LCD
                MOV     A,HOURS
                ANL     A,#11110000B
                JZ      WRITE_TIME_HN
                SWAP    A
                ADD     A,#030H
                AJMP    WRITE_TI
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
                MOV     LCD_DATA,MINUTES
                LCALL   BCD2LCD
                MOV     LCD_DATA,#':'
                LCALL   WRCHAR_LCD
                MOV     LCD_DATA,SECONDS
                LCALL   BCD2LCD

                MOV     LCD_ADDR,#014H
                LCALL   SET_ADDR_LCD
                MOV     DPTR,#LOAD1
                LCALL   WRLINE_LCD

                MOV     A,IO_DATA
                ANL     A,#0FH
                CJNE    A,#0BH,CH_SW1
                MOV     A,IO_DATA
                ANL     A,#0F0H
                CJNE    A,#0D0H,CH_SW2
;*************************************************

                MOV     DPTR,#01EFH
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        MOVX    A,@DPTR
        CJNE    A,#02H,X1
        MOV     DPTR,#01F0H
        MOVX    A,@DPTR
        CJNE    A,#00H,DE_C
        JNB     P1.4,SE_T
        ACALL   REFUSE
        CLR     P1.4
        MOV     LCD_ADDR,#058H
        LCALL   SET_ADDR_LCD
        MOV     LCD_DATA,#'0'
        LCALL   WRCHAR_LCD
        AJMP    Y1
SE_T:   ACALL   REFUSE
        SETB    P1.4
        MOV     LCD_ADDR,#058H
        LCALL   SET_ADDR_LCD
        MOV     LCD_DATA,#'1'
        LCALL   WRCHAR_LCD
        AJMP    Y1
REFUSE: MOV     DPTR,#01F8H
        MOVX    A,@DPTR
        MOV     DPTR,#01F0H
        MOVX    @DPTR,A
        RET
DE_C:   MOV     A,MINUTES
        ANL     A,#0FH
        CJNE    A,07FH,YZ
        AJMP    Y1
YZ:     MOV     DPTR,#01F0H
        MOVX    A,@DPTR
        DEC     A
        MOVX    @DPTR,A
        AJMP    Y1
;*******************************************************
CH_SW1: AJMP    CH_SW
;*******************************************************
CH_SW2: AJMP    CH_SW_AUTO
;*******************************************************
X1:     MOV     DPTR,#0000H
        MOV     R1,#01H
        ACALL   MAKE
;*******************************************************
Y1:     MOV     DPTR,#03EFH
        MOVX    A,@DPTR
        CJNE    A,#02H,X2
        MOV     DPTR,#01F1H
        MOVX    A,@DPTR
        CJNE    A,#00H,DE_C2
        JNB     P1.5,SE_T2
        ACALL   REFUSE2
        CLR     P1.5
        MOV     LCD_ADDR,#05AH
        LCALL   SET_ADDR_LCD
        MOV     LCD_DATA,#'0'
        LCALL   WRCHAR_LCD
        MP      Y2
SE_T2:  CALL    REFUSE2
        SETB    P1.5
        MOV     LCD_ADDR,#05AH
        LCALL   SET_ADDR_LCD
        MOV     LCD_DATA,#'1'
        LCALL   WRCHAR_LCD
        AJMP    Y2
REFUSE2: MOV    DPTR,#01F9H
        MOVX    A,@DPTR
        MOV     DPTR,#01F1H
        MOVX    @DPTR,A
        RET
DE_C2:  MOV     A,MINUTES
        ANL     A,#0FH
        CJNE    A,07FH,YZ2
        AJMP    Y2
YZ2:    MOV     DPTR,#01F1H
        MOVX    A,@DPTR
        DEC     A
        MOVX    @DPTR,A
        AJMP    Y2
X2:     MOV     DPTR,#0200H
        MOV     R1,#02H
        ACALL   MAKE
;*******************************************************
Y2:     MOV     DPTR,#05EFH
        MOVX    A,@DPTR
        CJNE    A,#02H,X3
        MOV     DPTR,#01F2H
        MOVX    A,@DPTR
        CJNE    A,#00H,DE_C3
        JNB     P1.6,SE_T3
        ACALL   REFUSE3
        CLR     P1.6
        MOV     LCD_ADDR,#05CH
        LCALL   SET_ADDR_LCD
        MOV     LCD_DATA,#'0'
        LCALL   WRCHAR_LCD
        AJMP    Y3
SE_T3:  ACALL   REFUSE3
        SETB    P1.6
        MOV     LCD_ADDR,#05CH
        LCALL   SET_ADDR_LCD
        MOV     LCD_DATA,#'1'
        LCALL   WRCHAR_LCD
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        AJMP    Y3
REFUSE3:MOV     DPTR,#01FAH
        MOVX    A,@DPTR
        MOV     DPTR,#01F2H
        MOVX    @DPTR,A
        RET
DE_C3:  MOV     A,MINUTES
        ANL     A,#0FH
        CJNE    A,07FH,YZ3
        AJMP    Y3
YZ3:    MOV     DPTR,#01F2H
        MOVX    A,@DPTR
        DEC     A
        MOVX    @DPTR,A
        AJMP    Y3
X3:     MOV     DPTR,#0400H
        MOV     R1,#03H
        ACALL   MAKE
;*************************************************
Y3:     MOV     DPTR,#07EFH
        MOVX    A,@DPTR
        CJNE    A,#02H,X4
        MOV     DPTR,#01F3H
        MOVX    A,@DPTR
        CJNE    A,#00H,DE_C4
        JNB     P1.7,SE_T4
        ACALL   REFUSE4
        CLR     P1.7
        MOV     LCD_ADDR,#05EH
        LCALL   SET_ADDR_LCD
        MOV     LCD_DATA,#'0'
        LCALL   WRCHAR_LCD
        AJMP    Y4
SE_T4:  ACALL   REFUSE4
        SETB    P1.7
        MOV     LCD_ADDR,#05EH
        LCALL   SET_ADDR_LCD
        MOV     LCD_DATA,#'1'
        LCALL   WRCHAR_LCD
        AJMP    Y4
REFUSE4:MOV     DPTR,#01FBH
        MOVX    A,@DPTR
        MOV     DPTR,#01F3H
        MOVX    @DPTR,A
        RET
DE_C4:  MOV     A,MINUTES
        ANL     A,#0FH
        CJNE    A,07FH,YZ4
        AJMP    Y4
YZ4:    MOV     DPTR,#01F3H
        MOVX    A,@DPTR
        DEC     A
        MOVX    @DPTR,A
        AJMP    Y4
X4:     MOV     DPTR,#0600H
        MOV     R1,#04H
        ACALL   MAKE
;*************************************************
Y4:     MOV     DPTR,#09EFH
        MOVX    A,@DPTR
        CJNE    A,#02H,X5
        MOV     DPTR,#01F4H
        MOVX    A,@DPTR
        CJNE    A,#00H,DE_C5
        JNB     P2.4,SE_T5
        ACALL   REFUSE5
        CLR     P2.4
        MOV     LCD_ADDR,#060H
        LCALL   SET_ADDR_LCD
        MOV     LCD_DATA,#'0'
        LCALL   WRCHAR_LCD
        AJMP    Y5
SE_T5:  ACALL   REFUSE5
        SETB    P2.4
        MOV     LCD_ADDR,#060H
        LCALL   SET_ADDR_LCD
        MOV     LCD_DATA,#'1'
        LCALL   WRCHAR_LCD
        JMP     Y5
REFUSE5:MOV     DPTR,#01FCH
        MOVX    A,@DPTR
        MOV     DPTR,#01F4H
        MOVX    @DPTR,A
        RET
DE_C5:  MOV     A,MINUTES
        ANL     A,#0FH
        CJNE    A,07FH,YZ5
        AJMP    Y5
YZ5:    MOV     DPTR,#01F4H\
        MOVX    A,@DPTR
        DEC     A
        MOVX    @DPTR,A
        AJMP    Y5
X5:     MOV     DPTR,#0800H
        MOV     R1,#05H
        ACALL   MAKE
;*************************************************
Y5:     MOV     DPTR,#0BEFH
        MOVX    A,@DPTR
        CJNE    A,#02H,X6
        MOV     DPTR,#01F5H
        MOVX    A,@DPTR
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        CJNE    A,#00H,DE_C6
        JNB     P2.5,SE_T6
        ACALL   REFUSE6
        CLR     P2.5
        MOV     LCD_ADDR,#062H
        LCALL   SET_ADDR_LCD
        MOV     LCD_DATA,#'0'
        LCALL   WRCHAR_LCD
        AJMP    Y6
SE_T6:  ACALL   REFUSE6
        SETB    P2.5
        MOV     LCD_ADDR,#062H
        LCALL   SET_ADDR_LCD
        MOV     LCD_DATA,#'1'
        LCALL   WRCHAR_LCD
        AJMP    Y6
REFUSE6: MOV    DPTR,#01FDH
        MOVX    A,@DPTR
        MOV     DPTR,#01F5H
        MOVX    @DPTR,A
        RET
DE_C6:  MOV     A,MINUTES
        ANL     A,#0FH
        CJNE    A,07FH,YZ6
        AJMP    Y6
YZ6:    MOV     DPTR,#01F5H
        MOVX    A,@DPTR
        DEC     A
        MOVX    @DPTR,A
        AJMP    Y6
X6:     MOV     DPTR,#0A00H
        MOV     R1,#06H
        ACALL   MAKE
;*****************************************************
Y6:     MOV     DPTR,#0DEFH
        MOVX    A,@DPTR
        CJNE    A,#02H,X7
        MOV     DPTR,#01F6H
        MOVX    A,@DPTR
        CJNE    A,#00H,DE_C7
        JNB     P2.6,SE_T7
        ACALL   REFUSE7
        CLR     P2.6
        MOV     LCD_ADDR,#064H
        LCALL   SET_ADDR_LCD
        MOV     LCD_DATA,#'0'
        LCALL   WRCHAR_LCD
        AJMP    Y7
SE_T7:  ACALL   REFUSE7
        SETB    P2.6
        MOV     LCD_ADDR,#064H
        LCALL   SET_ADDR_LCD
        MOV     LCD_DATA,#'1'
        LCALL   WRCHAR_LCD
        AJMP    Y7
REFUSE7: MOV    DPTR,#01FEH
        MOVX    A,@DPTR
        MOV     DPTR,#01F6H
        MOVX    @DPTR,A
        RET
DE_C7:  MOV     A,MINUTES
        ANL     A,#0FH
        CJNE    A,07FH,YZ7
        AJMP    Y7
YZ7:    MOV     DPTR,#01F6H
        MOVX    A,@DPTR
        DEC     A
        MOVX    @DPTR,A
        AJMP    Y7
X7:     MOV     DPTR,#0C00H
        MOV     R1,#07H
        ACALL   MAKE
;*****************************************************
Y7:     MOV     DPTR,#0FEFH
        MOVX    A,@DPTR
        CJNE    A,#02H,X8
        MOV     DPTR,#01F7H
        MOVX    A,@DPTR
        CJNE    A,#00H,DE_C8
        JNB     P2.7,SE_T8
        ACALL   REFUSE8
        CLR     P2.7
        MOV     LCD_ADDR,#066H
        LCALL   SET_ADDR_LCD
        MOV     LCD_DATA,#'0'
        LCALL   WRCHAR_LCD
        AJMP    Y8
SE_T8:  ACALL   REFUSE8
        SETB    P2.7
        MOV     LCD_ADDR,#066H
        LCALL   SET_ADDR_LCD
        MOV     LCD_DATA,#'1'
        LCALL   WRCHAR_LCD
        AJMP    Y8
REFUSE8: MOV    DPTR,#01FFH
        MOVX    A,@DPTR
        MOV     DPTR,#01F7H
        MOVX    @DPTR,A
        RET
DE_C8:  MOV     A,MINUTES
        ANL     A,#0FH
        CJNE    A,07FH,YZ8
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                AJMP    Y8
YZ8:            MOV     DPTR,#01F7H
                MOVX    A,@DPTR
                DEC     A
                MOVX    @DPTR,A
                AJMP    Y8
X8:             MOV     DPTR,#0E00H
                MOV     R1,#08H
                ACALL   MAKE
;*****************************************************
Y8:             MOV     A,MINUTES
                ANL     A,#0FH
                MOV     07FH,A

                MOV     A,FLAG2
                ANL     A,#01H
                CJNE    A,#01H,MI
                AJMP    DE
MI:             MOV     A,COUNT1
                CJNE    A,#00H,BX
                AJMP    LOOP
BX:             SETB    FLAG2.0
DE:             DEC     COUNT1
                MOV     A,COUNT1
                CJNE    A,#00H,OUTL
                MOV     COUNT1,#00H
                CLR     P3.2
                CLR     FLAG2.0
OUTL:           AJMP    LOOP
;END MAIN PROGRAM
;*****************************************************
ON_LED:         PUSH    ACC
                SETB    P3.2
                MOV     COUNT1,#13H
                POP     ACC
                RETI
;*****************************************************
CH_SW:          MOV     B,A
                ANL     A,#0AH
                CJNE    A,#0AH,SW2
                AJMP    SET_DATE
SW2:            MOV     A,B
                ANL     A,#0DH
                CJNE    A,#0DH,SW3
                AJMP    SET_TIME
SW3:            MOV     A,B
                ANL     A,#07H
                CJNE    A,#07H,SW_4
                LCALL   SET_PRO
SW_4:           RET

CH_SW_AUTO:     MOV     B,A
                ANL     A,#0E0H
                CJNE    A,#0E0H,MANUAL
                AJMP    DISABLE_LOAD
MANUAL:         MOV     A,B
                ANL     A,#0B0H
                CJNE    A,#0B0H,NO_SW
                LCALL   MANUAL_SYS
NO_SW:          RET
;*****************************************
MAKE:           MOV     R3,#01EH  ;**
PRONEXT:        DEC     R3
                INC     DPTR
                MOVX    A,@DPTR
                CJNE    A,HOURS,OUT1
                INC     DPTR
                MOVX    A,@DPTR
                CJNE    A,MINUTES,OUT2
                ACALL   SETLOAD1
                CJNE    R3,#0,PRONEXT
                RET
OUT1:           INC     DPTR
                INC     DPTR
                INC     DPTR
                INC     DPTR
                CJNE    R3,#0,PRONEXT
                RET
OUT2:           INC     DPTR
                INC     DPTR
                INC     DPTR
                CJNE    R3,#0,PRONEXT
                RET
;*****************************************************
SETLOAD1:       INC     DPTR
                MOVX    A,@DPTR
                CJNE    A,#01H,CHK_OFF0
                CJNE    R1,#01H,CP1
                SETB    P1.4
                MOV     LCD_ADDR,#058H
                LCALL   SET_ADDR_LCD
                MOV     LCD_DATA,#'1'
                LCALL   WRCHAR_LCD
                AJMP    CP8
CHK_OFF0:       AJMP    CHK_OFF
CP1:            CJNE    R1,#02H,CP2
                SETB    P1.5
                MOV     LCD_ADDR,#05AH
                LCALL   SET_ADDR_LCD
                MOV     LCD_DATA,#'1'

                LCALL   WRCHAR_LCD
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        AJMP    CP3
CP2:    CJNE    R1,#03H,CP3
        SETB    P1.6
        MOV     LCD_ADDR,#05CH
        LCALL   SET_ADDR_LCD
        MOV     LCD_DATA,#'1'
        LCALL   WRCHAR_LCD
        AJMP    CP8
CP3:    CJNE    R1,#04H,CP4
        SETB    P1.7
        MOV     LCD_ADDR,#05EH
        LCALL   SET_ADDR_LCD
        MOV     LCD_DATA,#'1'
        LCALL   WRCHAR_LCD
        AJMP    CP8
CP4:    CJNE    R1,#05H,CP5
        SETB    P2.4
        MOV     LCD_ADDR,#060H
        LCALL   SET_ADDR_LCD
        MOV     LCD_DATA,#'1'
        LCALL   WRCHAR_LCD
        AJMP    CP8
CP5:    JNE     R1,#06H,CP6
        SETB    P2.5
        MOV     LCD_ADDR,#062H
        LCALL   SET_ADDR_LCD
        MOV     LCD_DATA,#'1'
        LCALL   WRCHAR_LCD
        AJMP    CP8
CP6:    CJNE    R1,#07H,CP7
        SETB    P2.6
        MOV     LCD_ADDR,#064H
        LCALL   SET_ADDR_LCD
        MOV     LCD_DATA,#'1'
        LCALL   WRCHAR_LCD
        AJMP    CP8
CP7:    JNE     R1,#08H,CP8
        SETB    P2.7
        MOV     LCD_ADDR,#066H
        LCALL   SET_ADDR_LCD
        MOV     LCD_DATA,#'1'
        LCALL   WRCHAR_LCD
CP8:    INC     DPTR
        INC     DPTR
        RET
CHK_OFF: CJNE   A,#00H,EDL
        CJNE    R1,#01H,CB1
        CLR     P1.4
        MOV     LCD_ADDR,#058H
        LCALL   SET_ADDR_LCD
        MOV     LCD_DATA,#'0'
        LCALL   WRCHAR_LCD
        AJMP    CB8
EDL:    AJMP    ED
CB1:    CJNE    R1,#02H,CB2
        CLR     P1.5
        MOV     LCD_ADDR,#05AH
        LCALL   SET_ADDR_LCD
        MOV     LCD_DATA,#'0'
        LCALL   WRCHAR_LCD
        AJMP    CB8
CB2:    CJNE    R1,#03H,CB3
        CLR     P1.6
        MOV     LCD_ADDR,#05CH
        LCALL   SET_ADDR_LCD
        MOV     LCD_DATA,#'0'
        LCALL   WRCHAR_LCD
        AJMP    CB8
CB3:    CJNE    R1,#04H,CB4
        CLR     P1.7
        MOV     LCD_ADDR,#05EH
        LCALL   SET_ADDR_LCD
        MOV     LCD_DATA,#'0'
        LCALL   WRCHAR_LCD
        AJMP    CB8
CB4:    CJNE    R1,#05H,CB5
        CLR     P2.4
        MOV     LCD_ADDR,#060H
        LCALL   SET_ADDR_LCD
        MOV     LCD_DATA,#'0'
        LCALL   WRCHAR_LCD
        AJMP    CB8
CB5:    CJNE    R1,#06H,CB6
        CLR     P2.5
        MOV     LCD_ADDR,#062H
        LCALL   SET_ADDR_LCD
        MOV     LCD_DATA,#'0'
        LCALL   WRCHAR_LCD
        AJMP    CB8
CB6:    CJNE    R1,#07H,CB7
        CLR     P2.6
        MOV     LCD_ADDR,#064H
        LCALL   SET_ADDR_LCD
        MOV     LCD_DATA,#'0'
        LCALL   WRCHAR_LCD
        AJMP    CB8
CB7:    CJNE    R1,#08H,CB8
        CLR     P2.7
        MOV     CD_ADDR,#066H
        LCALL   SET_ADDR_LCD
        MOV     LCD_DATA,#'0'
        LCALL   WRCHAR_LCD
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
CB8:            INC     DPTR
                INC     DPTR
ED:             RET
;*************************************************
CHK_PASSW:      LCALL   KEEP_ST
                LCALL   LCD_CLR
                MOV     LCD_ADDR,#040H
                CALL    SET_ADDR_LCD
                MOV     DPTR,#KEY_PASSW
                LCALL   WRLINE_LCD

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

                MOV     LCD_ADDR,#01CH
                LCALL   SET_ADDR_LCD
                LCALL   LCD_BLINK

                LCALL   WAIT_KEYPRESSED
                MOV     BUFFER,KPAD_DATA
                MOV     LCD_DATA,#'*'
                LCALL   WRCHAR_LCD
                LCALL   LCD_BLINK
                LCALL   WAIT_KEY

                LCALL   WAIT_KEYPRESSED
                MOV     BUFFER+1,KPAD_DATA
                MOV     LCD_DATA,#'*'
                LCALL   WRCHAR_LCD
                LCALL   LCD_BLINK
                LCALL   WAIT_KEY
                LCALL   BUFFER2ACC
                MOV     PASSHIGH,A

                LCALL   WAIT_KEYPRESSED
                MOV     BUFFER,KPAD_DATA
                MOV     LCD_DATA,#'*'
                LCALL   WRCHAR_LCD
                LCALL   LCD_BLINK
                LCALL   WAIT_KEY

                LCALL   WAIT_KEYPRESSED
                MOV     BUFFER+1,KPAD_DATA
                MOV     LCD_DATA,#'*'
                LCALL   WRCHAR_LCD

                LCALL   BUFFER2ACC
                MOV     PASSLOW,A
                MOV     A,PASSHIGH
                CJNE    A,#12H,NOT_TRUE
                MOV     A,PASSLOW
                CJNE    A,#34H,NOT_TRUE
                LCALL   LCD_CLR
                RET
NOT_TRUE:       LCALL   LCD_CLR
                MOV     LCD_ADDR,#040H
                LCALL   SET_ADDR_LCD
                MOV     DPTR,#FALSE1
                LCALL   WRLINE_LCD
                MOV     LCD_ADDR,#014H
                LCALL   SET_ADDR_LCD
                MOV     DPTR,#FALSE2
                LCALL   WRLINE_LCD
                LCALL   DELAY_1s
                LCALL   DELAY_1s
                LCALL   LCD_CLR
                LCALL   BACK_ST
                AJMP    LOOP
;*************************************************
REST:           LCALL   LCD_CLR
                LCALL   BACK_ST
                AJMP    LOOP
;*************************************************
DISABLE_LOAD:   LCALL   LCD_CLR
                MOV     LCD_ADDR,#040H
                LCALL   SET_ADDR_LCD
                MOV     DPTR,#DISABLE_SHOW
                LCALL   WRLINE_LCD
                MOV     LCD_ADDR,#014H
                LCALL   SET_ADDR_LCD
                MOV     DPTR,#DISABLE_SHOW2
                LCALL   WRLINE_LCD
                CLR     P1.4
                CLR     P1.5
                CLR     P1.6
                CLR     P1.7
                CLR     P2.4
                CLR     P2.5
                CLR     P2.6
                CLR     P2.7
WAD:            LCALL   PCF_RD
                MOV     A,IO_DATA
                ANL     A,#0F0H
                CJNE    A,#0E0H,NQD
                AJMP    WAD
NQD             LCALL   LCD_CLR
                MOV     LCD_ADDR,#054H
                LCALL   SET_ADDR_LCD
                MOV     DPTR,#ST
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                LCALL   WRLINE_LCD
                AJMP    LOOP
;*************************************************
SET_DATE:       LCALL   KEEP_ST
                LCALL   CHK_PASSW
                LCALL   LCD_CLR
                MOV     LCD_ADDR,#000H
                LCALL   SET_ADDR_LCD
                MOV     DPTR,#SCR_SET_DATE
                LCALL   WRLINE_LCD

                MOV     LCD_ADDR,#06H
                LCALL   SET_ADDR_LCD
                LCALL   LCD_BLINK

                LCALL   WAIT_KEYPRESSED
                MOV     BUFFER,KPAD_DATA
                LCALL   KPAD2LCD
                LCALL   LCD_BLINK
                LCALL   WAIT_KEY

                LCALL   WAIT_KEYPRESSED
                MOV     BUFFER+1,KPAD_DATA
                LCALL   KPAD2LCD

                LCALL   BUFFER2ACC\
                MOV     DAY,A

                MOV     LCD_ADDR,#00AH
                LCALL   SET_ADDR_LCD
                LCALL   LCD_BLINK

                LCALL   WAIT_KEYPRESSED
                MOV     BUFFER,KPAD_DATA
                LCALL   KPAD2LCD
                LCALL   LCD_BLINK
                LCALL   WAIT_KEY

                LCALL   WAIT_KEYPRESSED
                MOV     BUFFER+1,KPAD_DATA
                LCALL   KPAD2LCD
                LCALL   WAIT_KEY

                LCALL   BUFFER2ACC
                MOV     DATE,A

                MOV     LCD_ADDR,#00DH
                LCALL   SET_ADDR_LCD
                LCALL   LCD_BLINK

                LCALL   WAIT_KEYPRESSED
                MOV     BUFFER,KPAD_DATA
                LCALL   KPAD2LCD
                LCALL   LCD_BLINK
                LCALL   WAIT_KEY

                LCALL   WAIT_KEYPRESSED
                MOV     BUFFER+1,KPAD_DATA
                LCALL   KPAD2LCD

                LCALL   BUFFER2ACC
                MOV     MONTH,A

                MOV     LCD_ADDR,#010H
                LCALL   SET_ADDR_LCD
                LCALL   LCD_BLINK

                LCALL   WAIT_KEYPRESSED
                MOV     BUFFER,KPAD_DATA
                LCALL   KPAD2LCD
                LCALL   LCD_BLINK
                LCALL   WAIT_KEY

                LCALL   WAIT_KEYPRESSED
                MOV     BUFFER+1,KPAD_DATA
                LCALL   KPAD2LCD

                LCALL   BUFFER2ACC
                MOV     YEAR,A
                LCALL   RTC_WR

                LCALL   LCD_CLR
                LCALL   BACK_ST
OO:             AJMP    LOOP
;*************************************************
SET_TIME:       LCALL   KEEP_ST
                LCALL   CHK_PASSW
                LCALL   LCD_CLR
                MOV     LCD_ADDR,#040H
                LCALL   SET_ADDR_LCD
                MOV     DPTR,#SCR_SET_TIME
                LCALL   WRLINE_LCD

                MOV     LCD_ADDR,#04AH
                LCALL   SET_ADDR_LCD
                LCALL   LCD_BLINK

                LCALL   WAIT_KEYPRESSED
                MOV     BUFFER,KPAD_DATA
                LCALL   KPAD2LCD
                LCALL   LCD_BLINK
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        LCALL   WAIT_KEY
        LCALL   WAIT_KEYPRESSED
        MOV     BUFFER+1,KPAD_DATA
        LCALL   KPAD2LCD
        LCALL   WAIT_KEY

        LCALL   BUFFER2ACC
        MOV     HOURS,A

        MOV     LCD_ADDR,#04DH
        LCALL   SET_ADDR_LCD
        LCALL   LCD_BLINK

        LCALL   WAIT_KEYPRESSED
        MOV     BUFFER,KPAD_DATA
        LCALL   KPAD2LCD
        LCALL   LCD_BLINK
        LCALL   WAIT_KEY

        LCALL   WAIT_KEYPRESSED
        MOV     BUFFER+1,KPAD_DATA
        LCALL   KPAD2LCD

        LCALL   BUFFER2ACC
        MOV     MINUTES,A

        MOV     LCD_ADDR,#050H
        LCALL   SET_ADDR_LCD
        LCALL   LCD_BLINK

        LCALL   WAIT_KEYPRESSED
        MOV     BUFFER,KPAD_DATA
        LCALL   KPAD2LCD
        LCALL   LCD_BLINK
        LCALL   WAIT_KEY

        LCALL   WAIT_KEYPRESSED
        MOV     BUFFER+1,KPAD_DATA
        LCALL   KPAD2LCD

        LCALL   BUFFER2ACC
        MOV     SECONDS,A

        LCALL   RTC_WR

WA1:    LCALL   PCF_RD
        MOV     A,IO_DATA
        ANL     A,#0FH
        CJNE    A,#0DH,NQ
        AJMP    WA1
NQ:     LCALL   LCD_CLR
        LCALL   BACK_ST
NQX:    LCALL   OO
;*****************************************************
MANUAL_SYS: LCALL   KEEP_ST
        LCALL   LCD_CLR
        MOV     LCD_ADDR,#000H
        LCALL   SET_ADDR_LCD
        MOV     DPTR,#MANUAL_SHOW
        LCALL   WRLINE_LCD
        MOV     LCD_ADDR,#040H
        LCALL   SET_ADDR_LCD
        MOV     DPTR,#KEY
        LCALL   WRLINE_LCD
        MOV     LCD_ADDR,#014H
        LCALL   SET_ADDR_LCD
        MOV     DPTR,#LOAD1
        LCALL   WRLINE_LCD
        MOV     LCD_ADDR,#054H
        LCALL   SET_ADDR_LCD
        MOV     DPTR,#ST
        LCALL   WRLINE_LCD
        LCALL   BACK_ST

        MOV     LCD_ADDR,#058H
        LCALL   SET_ADDR_LCD
        LCALL   LCD_BLINK
        LCALL   WAIT_KEYPRESSED_ST
        MOV     BUFSTATUS,KPAD_DATA
        LCALL   KPAD2LCD
        LCALL   LCD_BLINK
        MOV     A,BUFSTATUS
        ANL     A,#0FH
        CJNE    A,#01H,C_LR
        MOV     LCD_ADDR,#058H
        LCALL   SET_ADDR_LCD
        SETB    P1.4
        MOV     LCD_ADDR,#058H
        LCALL   SET_ADDR_LCD
        MOV     LCD_DATA,#'1'
        LCALL   WRCHAR_LCD
        AJMP    SHIFT2
C_LR:   CLR     P1.4
        MOV     LCD_ADDR,#058H
        LCALL   SET_ADDR_LCD
        MOV     LCD_DATA,#'0'
        LCALL   WRCHAR_LCD

SHIFT2: MOV     LCD_ADDR,#05AH
        LCALL   SET_ADDR_LCD
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        LCALL   LCD_BLINK
        LCALL   WAIT_KEYPRESSED_ST
        MOV     BUFSTATUS,KPAD_DATA
        LCALL   KPAD2LCD
        LCALL   LCD_BLINK
        MOV     A,BUFSTATUS
        ANL     A,#0FH
        CJNE    A,#01H,C_LR2
        MOV     LCD_ADDR,#05AH
        LCALL   SET_ADDR_LCD
        SETB    P1.5
        MOV     LCD_ADDR,#05AH
        LCALL   SET_ADDR_LCD
        MOV     LCD_DATA,#'1'
        LCALL   WRCHAR_LCD
        AJMP    SHIFT3
C_LR2:  CLR     P1.5
        MOV     LCD_ADDR,#05AH
        LCALL   SET_ADDR_LCD
        MOV     LCD_DATA,#'0'
        LCALL   WRCHAR_LCD

SHIFT3: MOV     LCD_ADDR,#05CH
        LCALL   SET_ADDR_LCD
        LCALL   LCD_BLINK
        LCALL   WAIT_KEYPRESSED_ST
        MOV     BUFSTATUS,KPAD_DATA
        LCALL   KPAD2LCD
        LCALL   LCD_BLINK
        MOV     A,BUFSTATUS
        ANL     A,#0FH
        CJNE    A,#01H,C_LR3
        MOV     LCD_ADDR,#05CH
        LCALL   SET_ADDR_LCD
        SETB    P1.6
        MOV     LCD_ADDR,#05CH
        LCALL   SET_ADDR_LCD
        MOV     LCD_DATA,#'1'
        LCALL   WRCHAR_LCD
        AJMP    SHIFT4
C_LR3:  CLR     P1.6
        MOV     LCD_ADDR,#05CH
        LCALL   SET_ADDR_LCD
        MOV     LCD_DATA,#'0'
        LCALL   WRCHAR_LCD

SHIFT4: MOV     LCD_ADDR,#05EH
        LCALL   SET_ADDR_LCD
        LCALL   LCD_BLINK
        LCALL   WAIT_KEYPRESSED_ST
        MOV     BUFSTATUS,KPAD_DATA
        LCALL   KPAD2LCD
        LCALL   LCD_BLINK
        MOV     A,BUFSTATUS
        ANL     A,#0FH
        CJNE    A,#01H,C_LR4
        MOV     LCD_ADDR,#05EH
        LCALL   SET_ADDR_LCD
        SETB    P1.7
        MOV     LCD_ADDR,#05EH
        LCALL   SET_ADDR_LCD
        MOV     LCD_DATA,#'1'
        LCALL   WRCHAR_LCD
        AJMP    SHIFT5
C_LR4:  CLR     P1.7
        MOV     LCD_ADDR,#05EH
        LCALL   SET_ADDR_LCD
        MOV     LCD_DATA,#'0'
        LCALL   WRCHAR_LCD

SHIFT5: MOV     LCD_ADDR,#060H
        LCALL   SET_ADDR_LCD
        LCALL   LCD_BLINK
        LCALL   WAIT_KEYPRESSED_S
        MOV     BUFSTATUS,KPAD_DATA
        LCALL   KPAD2LCD
        LCALL   LCD_BLINK
        MOV     A,BUFSTATUS
        ANL     A,#0FH
        CJNE    A,#01H,C_LR5
        MOV     LCD_ADDR,#060H
        LCALL   SET_ADDR_LCD
        SETB    P2.4
        MOV     LCD_ADDR,#060H
        LCALL   SET_ADDR_LCD
        MOV     LCD_DATA,#'1'
        LCALL   WRCHAR_LCD
        AJMP    SHIFT6
C_LR5:  CLR     P2.4
        MOV     LCD_ADDR,#060H
        LCALL   SET_ADDR_LCD
        MOV     LCD_DATA,#'0'
        LCALL   WRCHAR_LCD

SHIFT6: MOV     LCD_ADDR,#062H
        LCALL   SET_ADDR_LCD
        LCALL   LCD_BLINK
        LCALL   WAIT_KEYPRESSED_ST
        MOV     BUFSTATUS,KPAD_DATA
        LCALL   KPAD2LCD
        LCALL   LCD_BLINK
        MOV     A,BUFSTATUS
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                ANL     A,#0FH
                CJNE    A,#01H,C_LR6
                MOV     LCD_ADDR,#062H
                LCALL   SET_ADDR_LCD
                SETB    P2.5
                MOV     LCD_ADDR,#062H
                LCALL   SET_ADDR_LCD
                MOV     LCD_DATA,#'1'
                LCALL   WRCHAR_LCD
                AJMP    SHIFT7
C_LR6:          CLR     P2.5
                MOV     LCD_ADDR,#062H
                LCALL   SET_ADDR_LCD
                MOV     LCD_DATA,#'0'
                LCALL   WRCHAR_LCD

SHIFT7:         MOV     LCD_ADDR,#064H
                LCALL   SET_ADDR_LCD
                LCALL   LCD_BLINK
                LCALL   WAIT_KEYPRESSED_ST
                MOV     BUFSTATUS,KPAD_DATA
                LCALL   KPAD2LCD
                LCALL   LCD_BLINK
                MOV     A,BUFSTATUS
                ANL     A,#0FH
                CJNE    A,#01H,C_LR7
                MOV     LCD_ADDR,#064H
                LCALL   SET_ADDR_LCD
                SETB    P2.6
                MOV     LCD_ADDR,#064H
                LCALL   SET_ADDR_LCD
                MOV     LCD_DATA,#'1'
                LCALL   WRCHAR_LCD
                AJMP    SHIFT8
C_LR7:          CLR     P2.6
                MOV     LCD_ADDR,#064H
                LCALL   SET_ADDR_LCD
                MOV     LCD_DATA,#'0'
                LCALL   WRCHAR_LCD

SHIFT8:         MOV     LCD_ADDR,#066H
                LCALL   SET_ADDR_LCD
                LCALL   LCD_BLINK
                LCALL   WAIT_KEYPRESSED_ST
                MOV     BUFSTATUS,KPAD_DATA
                LCALL   KPAD2LCD
                LCALL   LCD_BLINK
                MOV     A,BUFSTATUS
                ANL     A,#0FH
                CJNE    A,#01H,C_LR8
                MOV     LCD_ADDR,#066H
                LCALL   SET_ADDR_LCD
                SETB    P2.7
                MOV     LCD_ADDR,#066H
                LCALL   SET_ADDR_LCD
                MOV     LCD_DATA,#'1'
                LCALL   WRCHAR_LCD
                AJMP    WAM
C_LR8:          CLR     P2.7
                MOV     LCD_ADDR,#066H
                LCALL   SET_ADDR_LCD
                MOV     LCD_DATA,#'0'
                LCALL   WRCHAR_LCD

WAM:            LCALL   PCF_RD
                MOV     A,IO_DATA
                ANL     A,#0F0H
                CJNE    A,#0B0H,NQM
                AJMP    WAM
NQM:            LCALL   LCD_CLR
                MOV     LCD_ADDR,#054H
                LCALL   SET_ADDR_LCD
                LCALL   CLR_LOAD
                MOV     DPTR,#ST
                LCALL   WRLINE_LCD
                LCALL   NQX
;***************************************************
CLR_LOAD:       CLR     P1.4
                CLR     P1.5
                CLR     P1.6
                CLR     P1.7
                CLR     P2.4
                CLR     P2.5
                CLR     P2.6
                CLR     P2.7
                RET
;***************************************************
KEEP_ST:        MOV     A,P1
                ANL     A,#0F0H
                MOV     B,A
                MOV     A,P2
                ANL     A,#0F0H
                SWAP    A
                ADD     A,B
                SWAP    A
                MOV     BUF_LOAD,A
                RET

BACK_ST:        MOV     ADD_ST,#068H
                MOV     R3,#8
                MOV     A,BUF_LOAD
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
NL:     CLR     C
        RLC     A
        JNC     ZERO
        PUSH    ACC
        MOV     A,ADD_ST
        CLR     C
        SUBB    A,#02H
        MOV     ADD_ST,A
        MOV     LCD_ADDR,ADD_ST
        LCALL   SET_ADDR_LCD
        MOV     LCD_DATA,#'1'
        LCALL   WRCHAR_LCD
        POP     ACC
        DJNZ    R3,NL
        MOV     LCD_ADDR,#054H
        LCALL   SET_ADDR_LCD
        MOV     LCD_DATA,#'S'
        LCALL   WRCHAR_LCD
        MOV     LCD_ADDR,#055H
        LCALL   SET_ADDR_LCD
        MOV     LCD_DATA,#'T'
        LCALL   WRCHAR_LCD
        MOV     LCD_ADDR,#056H
        LCALL   SET_ADDR_LCD
        MOV     LCD_DATA,#':'
        LCALL   WRCHAR_LCD
        RET

ZERO:   PUSH    ACC
        MOV     A,ADD_ST
        CLR     C
        SUBB    A,#02H
        MOV     ADD_ST,A
        MOV     LCD_ADDR,ADD_ST
        LCALL   SET_ADDR_LCD
        MOV     LCD_DATA,#'0'
        LCALL   WRCHAR_LCD
        POP     ACC
        DJNZ    R3,NL
        MOV     LCD_ADDR,#054H
        LCALL   SET_ADDR_LCD
        MOV     LCD_DATA,#'S'
        LCALL   WRCHAR_LCD
        MOV     LCD_ADDR,#055H
        LCALL   SET_ADDR_LCD
        MOV     LCD_DATA,#'T'
        \LCALL  WRCHAR_LCD
        RET
;*****************************************************
SET_PRO: LCALL  KEEP_ST
        LCALL   CHK_PASSW
        LCALL   LCD_CLR
        MOV     LCD_ADDR,#000H
        LCALL   SET_ADDR_LCD
        MOV     DPTR,#SELECT_L1
        LCALL   WRLINE_LCD
        MOV     LCD_ADDR,#040H
        LCALL   SET_ADDR_LCD
        MOV     DPTR,#SELECT_L2
        LCALL   WRLINE_LCD
        MOV     LCD_ADDR,#014H
        LCALL   SET_ADDR_LCD
        MOV     DPTR,#SELECT_LOAD
        LCALL   WRLINE_LCD

        MOV     LCD_ADDR,#022H
        LCALL   SET_ADDR_LCD
        LCALL   LCD_BLINK
        LCALL   WAIT_KEYPRESSED3
        MOV     BUFSTATUS,KPAD_DATA
        LCALL   KPAD2LCD
        LCALL   LCD_BLINK
        MOV     A,BUFSTATUS
        MOV     SELECT_LO,A

        LCALL   LCD_CLR
        MOV     LCD_ADDR,#000H
        LCALL   SET_ADDR_LCD
        MOV     DPTR,#SELECT_P
        LCALL   WRLINE_LCD
        MOV     LCD_ADDR,#040H
        LCALL   SET_ADDR_LCD
        MOV     DPTR,#CHOICE1
        LCALL   WRLINE_LCD
        MOV     LCD_ADDR,#014H
        LCALL   SET_ADDR_LCD
        MOV     DPTR,#CHOICE2
        LCALL   WRLINE_LCD
        MOV     LCD_ADDR,#054H
        LCALL   SET_ADDR_LCD
        MOV     DPTR,#SELECT_PRO
        LCALL   WRLINE_LCD

        MOV     LCD_ADDR,#063H
        LCALL   SET_ADDR_LCD
        LCALL   LCD_BLINK
        LCALL   WAIT_KEYPRESSED2
        MOV     BUFSTATUS,KPAD_DATA
        LCALL   KPAD2LCD
        LCALL   LCD_BLINK
        MOV     A,BUFSTATUS
        MOV     SELECT_PR,A
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                CJNE    A,#02H,EVERY1
                AJMP    PRO_DELAY
EVERY1:         LCALL   EVERY

PRO_DELAY:      LCALL   LCD_CLR
                MOV     LCD_ADDR,#040H
                LCALL   SET_ADDR_LCD
                MOV     DPTR,#TIMEDELAY
                LCALL   WRLINE_LCD
                MOV     LCD_ADDR,#04BH
                LCALL   SET_ADDR_LCD
                LCALL   LCD_BLINK

                LCALL   WAIT_KEYPRESSED
                MOV     BUFFER,KPAD_DATA
                LCALL   KPAD2LCD
                LCALL   LCD_BLINK
                LCALL   WAIT_KEY

                LCALL   WAIT_KEYPRESSED
                MOV     BUFFER+1,KPAD_DATA
                LCALL   KPAD2LCD

                LCALL   BUFFER2ACC
                MOV     TIME,A

                MOV     A,SELECT_LO
                CJNE    A,#01H,TE1
                MOV     A,TIME
                MOV     DPTR,#01F0H
                MOVX    @DPTR,A
                MOV     DPTR,#01F8H
                MOVX    @DPTR,A
                MOV     A,#02H
                MOV     DPTR,#01EFH
                LCALL   PUTTIME
TE1:            CJNE    A,#02H,TE2
                MOV     A,TIME
                MOV     DPTR,#01F1H
                MOVX    @DPTR,A
                MOV     DPTR,#01F9H
                MOVX    @DPTR,A
                MOV     A,#02H
                MOV     DPTR,#03EFH
                LCALL   PUTTIME
TE2:            CJNE    A,#03H,TE3
                MOV     A,TIME
                MOV     DPTR,#01F2H
                MOVX    @DPTR,A
                MOV     DPTR,#01FAH
                MOVX    @DPTR,A
                MOV     A,#02H
                MOV     DPTR,#05EFH
                LCALL   PUTTIME
TE3:            CJNE    A,#04H,TE4
                MOV     A,TIME
                MOV     DPTR,#01F3H
                MOVX    @DPTR,A
                MOV     DPTR,#01FBH
                MOVX    @DPTR,A
                MOV     A,#02H
                MOV     DPTR,#07EFH
                LCALL   PUTTIME
TE4:            CJNE    A,#05H,TE5
                MOV     A,TIME
                MOV     DPTR,#01F4H
                MOVX    @DPTR,A
                MOV     DPTR,#01FCH
                MOVX    @DPTR,A
                MOV     A,#02H
                MOV     DPTR,#09EFH
                LCALL   PUTTIME
TE5:            CJNE    A,#06H,TE6
                MOV     A,TIME
                MOV     DPTR,#01F5H
                MOVX    @DPTR,A
                MOV     DPTR,#01FDH
                MOVX    @DPTR,A
                MOV     A,#02H
                MOV     DPTR,#0BEFH
                LCALL   PUTTIME
TE6:            CJNE    A,#07H,TE7
                MOV     A,TIME
                MOV     DPTR,#01F6H
                MOVX    @DPTR,A
                MOV     DPTR,#01FEH
                MOVX    @DPTR,A
                MOV     A,#02H
                MOV     DPTR,#0DEFH
                LCALL   PUTTIME
TE7:            CJNE    A,#08H,WT2
                MOV     A,TIME
                MOV     DPTR,#01F7H
                MOVX    @DPTR,A
                MOV     DPTR,#01FFH
                MOVX    @DPTR,A
                MOV     A,#02H
                MOV     DPTR,#0FEFH
                LCALL   PUTTIME

PUTTIME:        MOVX    @DPTR,A
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        MOV     A,TIME
        MOV     96H,#01H

WT2:    \LCALL  PCF_RD
        MOV     A,IO_DATA
        ANL     A,#0FH
        CJNE    A,#07H,NT2
        AJMP    WT2

NT2:    LCALL   LCD_CLR
        LCALL   BACK_ST
        LCALL   NQX
;*****************************************************
EVERY:  MOV     A,SELECT_LO
        CJNE    A,#01H,EE1
        MOV     DPTR,#01EFH
        MOV     A,#00H
        MOVX    @DPTR,A
        CLR     P1.4
        MOV     DPTR,#0000H
        LCALL   PUTVALUE
EE1:    CJNE    A,#02H,EE2
        MOV     DPTR,#03EFH
        MOV     A,#00H
        MOVX    @DPTR,A
        CLR     P1.5
        MOV     DPTR,#0200H
        LCALL   PUTVALUE
EE2:    CJNE    A,#03H,EE3
        MOV     DPTR,#05EFH
        MOV     A,#00H
        MOVX    @DPTR,A
        CLR     P1.6
        MOV     DPTR,#0400H
        LCALL   PUTVALUE
EE3:    CJNE    A,#04H,EE4
        MOV     DPTR,#07EFH
        MOV     A,#00H
        MOVX    @DPTR,A
        CLR     P1.7
        MOV     DPTR,#0600H
        LCALL   PUTVALUE
EE4:    CJNE    A,#05H,EE5
        MOV     DPTR,#09EFH
        MOV     A,#00H
        MOVX    @DPTR,A
        CLR     P2.4
        MOV     DPTR,#0800H
        LCALL   PUTVALUE
EE5:    CJNE    A,#06H,EE6
        MOV     DPTR,#0BEFH
        MOV     A,#00H
        MOVX    @DPTR,A
        CLR     P2.5
        MOV     DPTR,#0A00H
        LCALL   PUTVALUE
EE6:    CJNE    A,#07H,EE7
        MOV     DPTR,#0DEFH
        MOV     A,#00H
        MOVX    @DPTR,A
        CLR     P2.6
        MOV     DPTR,#0C00H
        LCALL   PUTVALUE
EE7:    CJNE    A,#08H,EE8
        MOV     DPTR,#0FEFH
        MOV     A,#00H
        MOVX    @DPTR,A
        CLR     P2.7
        MOV     DPTR,#0E00H
        LCALL   PUTVALUE
EE8:    LCALL   NT2
;*****************************************************
PUTVALUE:
        MOV     BUFHIGH,83H
        MOV     BUFLOW,82H
        MOV     HIGH1,83H
        MOV     LOW1,82H
        MOV     LOW_NUM,#00H
        MOV     HIGH_NUM,#00H

        MOV     R4,#01EH
NEXT_PUT:
        LCALL   PUT_NUMBER
        MOV     A,BUFFER1+1
        ANL     A,#00FH
        SWAP    A
        MOV     B,A
        MOV     A,BUFFER1
        ANL     A,#00FH
        ADD     A,B
        MOV     82H,LOW1
        MOV     83H,HIGH1
        MOVX    @DPTR,A
        LCALL   INCDPTR3
        DJNZ    R4,NEXT_PUT

        MOV     BUFLOW2,BUFLOW
        MOV     BUFHIGH2,BUFHIGH
        MOV     BUFLOW3,BUFLOW
        MOV     BUFHIGH3,BUFHIGH

        MOV     COUNT2,#0AH
PUT:    LCALL   LCD_CLR
        MOV     LCD_ADDR,#000H
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        LCALL   SET_ADDR_LCD
        MOV     DPTR,#PRO1
        LCALL   WRLINE_LCD

        MOV     LOW1,BUFLOW2
        MOV     HIGH1,BUFHIGH2
        MOV     82H,LOW1
        MOV     83H,HIGH1

        MOV     LCD_ADDR,#040H
        LCALL   SET_ADDR_LCD
        MOVX    A,@DPTR
        MOV     LCD_DATA,A
        LCALL   HEX2LCD

        MOV     LCD_ADDR,#044H
        LCALL   SET_ADDR_LCD
        MOV     DPTR,#PRO2
        LCALL   WRLINE_LCD3

        LCALL   INCDPTR3
        MOV     82H,LOW1
        MOV     83H,HIGH1

        MOV     LCD_ADDR,#014H
        LCALL   SET_ADDR_LCD
        MOVX    A,@DPTR
        MOV     LCD_DATA,A
        LCALL   HEX2LCD

        MOV     LCD_ADDR,#018H
        LCALL   SET_ADDR_LCD
        MOV     DPTR,#PRO2
        LCALL   WRLINE_LCD3

        LCALL   INCDPTR3
        MOV     82H,LOW1
        MOV     83H,HIGH1

        MOV     LCD_ADDR,#054H
        LCALL   SET_ADDR_LCD
        MOVX    A,@DPTR
        MOV     LCD_DATA,A
        LCALL   HEX2LCD

        MOV     LCD_ADDR,#058H
        LCALL   SET_ADDR_LCD
        MOV     DPTR,#PRO2
        LCALL   WRLINE_LCD3

        LCALL   INCDPTR3

PO:     MOV     LOW1,BUFLOW3
        MOV     HIGH1,BUFHIGH3
        MOV     ADD_LCD1,#040H
        LCALL   RECIVE
        LCALL   INCDPTR
        MOV     82H,LOW1
        MOV     83H,HIGH1
        LCALL   WR_EXRAM
        MOV     LOW1,82H
        MOV     HIGH1,83H
        LCALL   INCDPTR
        LCALL   INCDPTR

        MOV     ADD_LCD1,#014H
        LCALL   RECIVE
        MOV     82H,LOW1
        MOV     83H,HIGH1
        LCALL   WR_EXRAM
        MOV     LOW1,82H
        MOV     HIGH1,83H
        LCALL   INCDPTR
        LCALL   INCDPTR

        MOV     ADD_LCD1,#054H
        LCALL   RECIVE
        MOV     82H,LOW1
        MOV     83H,HIGH1
        LCALL   WR_EXRAM

        MOV     LOW1,82H
        MOV     HIGH1,83H
        LCALL   INCDPTR
        MOV     BUFLOW3,LOW1
        MOV     BUFHIGH3,HIGH1
        DEC     COUNT2
        MOV     A,COUNT2
        CJNE    A,#00H,PUT2

WA2:    LCALL   PCF_RD
        MOV     A,IO_DATA
        ANL     A,#0FH
        CJNE    A,#07H,NQ2
        AJMP    WA2

NQ2:    LCALL   NT2
PUT2:   AJMP    PUT
;************************************ manual select **********
WAIT_KEYPRESSED_ST:JNB  CHKEY,READ_ST
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                AJMP    WAIT_KEYPRESSED_ST
READ_ST:        MOV     KPAD_DATA,#0
                MOV     A,P1
                ANL     A,#00FH
                CJNE    A,#00EH,C_HK_ST
                MOV     KPAD_DATA,#00H
                RET
C_HK_ST:        CJNE    A,#03H,WAIT_KEYPRESSED_ST
                MOV     KPAD_DATA,#01H
                RET
;************************************ select programs*********
WAIT_KEYPRESSED2: JNB   CHKEY,READ2
                AJMP    WAIT_KEYPRESSED2
READ2:          MOV     KPAD_DATA,#0
                MOV     A,P1
                ANL     A,#00FH
                CJNE    A,#003H,C_HK2
                MOV     KPAD_DATA,#01H
                RET
C_HK2:          CJNE    A,#002H,WAIT_KEYPRESSED2
                MOV     KPAD_DATA,#02H
                RET
;************************************ select load*************
WAIT_KEYPRESSED3: JNB   CHKEY,READ3
                JMP     WAIT_KEYPRESSED3
READ3:          MOV     KPAD_DATA,#0
                MOV     A,P1
                ANL     A,#00FH
                CJNE    A,#001H,CH3
                MOV     KPAD_DATA,#03H
                RET
CH3:            CJNE    A,#002H,CH4
                MOV     KPAD_DATA,#02H
                RET
CH4:            CJNE    A,#003H,CH5
                MOV     KPAD_DATA,#01H
                RET
CH5:            CJNE    A,#005H,CH6
                MOV     KPAD_DATA,#06H
                RET
CH6:            CJNE    A,#006H,CH7
                MOV     KPAD_DATA,#05H
                RET
CH7:            CJNE    A,#007H,CH8
                MOV     KPAD_DATA,#04H
                RET
CH8:            CJNE    A,#00AH,CH9
                MOV     KPAD_DATA,#08H
                RET
CH9:            CJNE    ,#00BH,WAIT_KEYPRESSED3
                MOV     KPAD_DATA,#07H
                RET
;*************************************************************
RECIVE:         MOV     A,#0BH
                ADD     A,ADD_LCD1
                MOV     LCD_ADDR,A
                LCALL   SET_ADDR_LCD
                LCALL   LCD_BLINK

                LCALL   WAIT_KEYPRESSED
                MOV     BUFFER,KPAD_DATA
                LCALL   KPAD2LCD
                LCALL   LCD_BLINK
                LCALL   WAIT_KEY

                LCALL   WAIT_KEYPRESSED
                MOV     BUFFER+1,KPAD_DATA
                LCALL   KPAD2LCD

                LCALL   BUFFER2ACC
                MOV     BUFHOUR,A

                MOV     A,#0EH
                ADD     A,ADD_LCD1
                MOV     LCD_ADDR,A
                LCALL   SET_ADDR_LCD
                LCALL   LCD_BLINK

                LCALL   WAIT_KEYPRESSED
                MOV     BUFFER,KPAD_DATA
                LCALL   KPAD2LCD
                LCALL   LCD_BLINK
                LCALL   WAIT_KEY
                LCALL   WAIT_KEYPRESSED
                MOV     BUFFER+1,KPAD_DATA
                LCALL   KPAD2LCD
                LCALL   WAIT_KEY

                LCALL   BUFFER2ACC
                MOV     BUFMINUTE,A

                MOV     A,#012H
                ADD     A,ADD_LCD1
                MOV     LCD_ADDR,A
                LCALL   SET_ADDR_LCD
                LCALL   LCD_BLINK

                LCALL   WAIT_KEYPRESSED
                MOV     BUFSTATUS,KPAD_DATA
                LCALL   KPAD2LCD
                LCALL   LCD_BLINK
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
EL3:        LCALL   WAIT_KEYPRESSED_S
            MOV     A,KPAD_DATA
            CJNE    A,#0FH,EL1
            LCALL   RECIVE
EL1:        CJNE    A,#08H,EL2
            MOV     96H,#01H
            LCALL   LCD_CLR
            LCALL   BACK_ST
            LCALL   NQX
EL2:        CJNE    A,#0CH,EL3
            RET
;*****************************************************
INCDPTR:    MOV     A,LOW1
            CJNE    A,#0FFH,THEN
            MOV     LOW1,#00H
            INC     HIGH1
            MOV     \A,HIGH1
            CJNE    A,#0FH,THEN2
            RET
THEN2:      INC     LOW1
            RET
THEN:       INC     LOW1
            RET
;*****************************************************
INCDPTR3:   MOV     A,LOW1
            CJNE    A,#0FFH,THEN3
            MOV     LOW1,#00H
            INC     HIGH1
            MOV     A,HIGH1
            CJNE    A,#0FH,THEN4
            RET
THEN4:      INC     LOW1
            INC     LOW1
            INC     LOW1
            INC     LOW1
            INC     LOW1
            RET
THEN3:      INC     LOW1
            INC     LOW1
            INC     LOW1
            INC     LOW1
            INC     LOW1
            RET
;*****************************************************
PUT_NUMBER: INC     LOW_NUM
            MOV     A,LOW_NUM
            CJNE    A,#00AH,ED2
            INC     HIGH_NUM
            MOV     A,HIGH_NUM
            CJNE    A,#00AH,ED3
            RET
ED2:        MOV     BUFFER1,A
            MOV     BUFFER1+1,HIGH_NUM
            RET
ED3:        MOV     LOW_NUM,#00H
            MOV     BUFFER1,LOW_NUM
            MOV     BUFFER1+1,HIGH_NUM
            RET
;*****************************************************
INCDPTR2:   MOV     LOW1,82H
            MOV     HIGH1,83H
            MOV     A,82H
            CJNE    A,#0FFH,THEN6
            MOV     82H,#00H
            INC     83H
            MOV     HIGH1,83H
            MOV     A,83H
            CJNE    A,#1FH,THEN5
            RET
THEN5:      INC     LOW1
            INC     LOW1
            RET
THEN6:      INC     LOW1
            INC     LOW1
            RET
;*****************************************************
HEX2LCD:    PUSH    ACC
            MOV     A,LCD_DATA
            MOV     B,#16
            DIV     AB
            ADD     A,#030H
            MOV     LCD_DATA,A
            LCALL   HEX_CHK
            LCALL   WRCHAR_LCD
            MOV     A,B
            ADD     A,#030H
            MOV     LCD_DATA,A
            ACALL   HEX_CHK
            LCALL   WRCHAR_LCD
            POP     ACC
            RET

HEX_CHK:    MOV     A,LCD_DATA
            CJNE    A,#03AH,CHK_OTHER
CHK_OTHER:  JNC     CONV_2_ALPHA
            RET

CONV_2_ALPHA: ADD   A,#7
            MOV     LCD_DATA,A
            RET
;*****************************************************
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
HEX1LCD:    PUSH    ACC
            MOV     A,LCD_DATA
            MOV     B,#16
            DIV     AB
            ADD     A,#030H
            MOV     LCD_DATA,A
            LCALL   HEX_CHK2
            MOV     A,B
            ADD     A,#030H
            MOV     LCD_DATA,A
            LCALL   HEX_CHK2
            LCALL   WRCHAR_LCD
            POP     ACC
            RET
.
HEX_CHK2:   MOV     A,LCD_DATA
.           CJNE    A,#03AH,CHK_OTHER2
CHK_OTHER2: JNC     CONV_2_ALPHA2
            RET

CONV_2_ALPHA2: ADD  ,#7
            MOV     LCD_DATA,A
            RET
;*************************************************
WR_EXRAM:   MOV     R0,#49H
            MOV     R1,#0
WR_NEXT:    MOV     A,@R0
            MOVX    @DPTR,A
            INC     DPTR
            INC     R0
            INC     R1
            CJNE    R1,#3,WR_NEXT
            MOV     R1,#0
            RET
;*************************************************
PCF8574_WR: MOV     I2C_ADDR,#PCF8574_ID
            LCALL   I2C_SLAVE

            MOV     I2C_DATA,PL
            LCALL   I2C_DATA_WR

            LCALL   I2C_STOP
            RET
;*************************************************
PCF_RD:     MOV     I2C_ADDR,#PCF_ID+1
            LCALL   I2C_SLAVE
            ACALL   I2C_DATA_RD
            MOV     IO_DATA,I2C_DATA
            LCALL   2C_NACK_BIT
            LCALL   I2C_STOP
            RET
;*************************************************
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
            R   SCL
            NZ      R5,I2C_DATA_RD_1
            MOV     I2C_DATA,A
            POP     ACC
            RET
;*************************************************
BUFFER2ACC: MOV     A,BUFFER
            ANL     A,#00FH
            SWAP    A
            MOV     B,A
            MOV     A,BUFFER+1
            ANL     A,#00FH
            ADD     A,B
            RET
;*************************************************
KPAD2LCD:   MOV     A,KPAD_DATA
            ADD     A,#030H
            MOV     LCD_DATA,A
            LCALL   WRCHAR_LCD
            RET
;*************************************************
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
;*************************************************
WAIT_KEY:   JNB     CHKEY,RELEASE
            AJMP    WAIT_KEY
RELEASE:    RET
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
;********************************** WAIT ***************
WAIT_KEYPRESSED_S:JNB     CHKEY,READS
                  AJMP    WAIT_KEYPRESSED_S
READS:            MOV     KPAD_DATA,#0
                  MOV     A,P1
                  ANL     A,#00FH
                  CJNE    A,#00FH,KS2
                  MOV     KPAD_DATA,#0FH
                  RET
KS2:              CJNE    A,#00CH,KS3
                  MOV     KPAD_DATA,#0CH
                  RET
KS3:              CJNE    A,#008H,WAIT_KEYPRESSED_S
                  MOV     KPAD_DATA,#08H
                  RET
;********************************** WAIT 0-9*************
WAIT_KEYPRESSED:  JNB     CHKEY,READ
                  AJMP    WAIT_KEYPRESSED
READ:             MOV     KPAD_DATA,#0
                  MOV     A,P1
                  ANL     A,#00FH
                  CJNE    A,#001H,CHK3
                  MOV     KPAD_DATA,#03H
                  RET
CHK3:             CJNE    A,#002H,CHK4
                  MOV     KPAD_DATA,#02H
                  RET
CHK4:             CJNE    A,#003H,CHK5
                  MOV     KPAD_DATA,#01H
                  RET
CHK5:             CJNE    A,#005H,CHK6
                  MOV     KPAD_DATA,#06H
                  RET
CHK6:             CJNE    A,#006H,CHK7
                  MOV     KPAD_DATA,#05H
                  RET
CHK7:             CJNE    A,#007H,CHK8
                  MOV     KPAD_DATA,#04H
                  RET
CHK8:             CJNE    A,#009H,CHK9
                  MOV     KPAD_DATA,#09H
                  RET
CHK9:             CJNE    A,#00AH,CHK10
                  MOV     KPAD_DATA,#08H
                  RET
CHK10:            CJNE    A,#00BH,CHK11
                  MOV     KPAD_DATA,#07H
                  RET
CHK11:            CJNE    A,#008H,CHK12
                  LCALL   KEEP_ST
                  LCALL   REST
CHK12:            CJNE    A,#00EH,WAIT_KEYPRESSED
                  MOV     KPAD_DATA,#00H
                  RET
;*****************************************************
INIT_LCD:         LCALL   DELAY_100ms
                  CLR     LCD_RS

                  MOV     PL,#00111000B
                  LCALL   PCF8574_WR
                  ACALL   LCD_CLK
                  LCALL   DELAY_10ms

                  MOV     PL,#00111000B
                  LCALL   PCF8574_WR
                  ACALL   LCD_CLK

                  ACALL   LCD_OFF

                  ACALL   LCD_CLR

                  MOV     PL,#00000110B
                  LCALL   PCF8574_WR
                  ACALL   LCD_CLK

                  ACALL   LCD_HOME
;*****************************************************
LCD_CLR:          CLR     LCD_RS
                  MOV     PL,#00000001B
                  LCALL   PCF8574_WR
                  ACALL   LCD_CLK
                  RET
;*****************************************************
LCD_HOME:         CLR     LCD_RS
                  MOV     PL,#00000010B
                  LCALL   PCF8574_WR
                  ACALL   LCD_CLK
                  RET
;*****************************************************
LCD_OFF:          CLR     LCD_RS
                  MOV     PL,#00001000B
                  LCALL   PCF8574_WR
                  ACALL   LCD_CLK
                  RET
;*****************************************************
LCD_CLK:          SETB    LCD_EN
                  LCALL   LCD_DELAY
                  CLR     LCD_EN
                  LCALL   LCD_DELAY
                  RET
;*****************************************************
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
LCD_ON:         CLR     LCD_RS
                MOV     PL,#00001100B
                LCALL   PCF8574_WR
                ACALL   LCD_CLK
                RET
;*****************************************************
LCD_BLINK:      CLR     LCD_RS
                MOV     PL,#00001111B
                LCALL   PCF8574_WR
                LCALL   LCD_CLK
                RET
;*****************************************************
SET_ADDR_LCD:   CLR     LCD_RS
                MOV     A,LCD_ADDR
                SETB    ACC.7
                MOV     PL,A
                LCALL   PCF8574_WR
                LCALL   LCD_CLK
                RET
;*****************************************************
WRCHAR_LCD:     SETB    LCD_RS
                MOV     L,LCD_DATA
                LCALL   PCF8574_WR
                LCALL   LCD_CLK
                LCALL   LCD_ON
                RET
;*****************************************************
WRLINE_LCD:     MOV     R0,#0
WRLINE_LCD_1:   SETB    LCD_RS
                CLR     A
                MOVC    A,@A+DPTR
                MOV     PL,A
                LCALL   PCF8574_WR
                LCALL   LCD_CLK
                INC     DPTR
                INC     R0
                CJNE    R0,#20,WRLINE_LCD_1
                LCALL   LCD_ON
                RET
;*****************************************************
WRLINE_LCD2:    MOV     R0,#0
WRLINE_LCD_3:   SETB    LCD_RS
                CLR     A
                MOVC    A,@A+DPTR
                MOV     PL,A
                LCALL   PCF8574_WR
                LCALL   LCD_CLK
                INC     DPTR
                INC     R0
                CJNE    R0,#6,WRLINE_LCD_3
                LCALL   LCD_ON
                RET
;*****************************************************
WRLINE_LCD3:    MOV     R0,#0
WRLINE_LCD_4:   SETB    LCD_RS
                CLR     A
                MOVC    A,@A+DPTR
                MOV     PL,A
                LCALL   PCF8574_WR
                LCALL   LCD_CLK
                INC     DPTR
                INC     R0
                CJNE    R0,#16,WRLINE_LCD_4
                LCALL   LCD_ON
                RET
;*****************************************************
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     PL,A
                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
;*****************************************************
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
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        LCALL   I2C_ACK_BIT

        LCALL   I2C_DATA_RD
        MOV     HOURS,I2C_DATA
        LCALL   I2C_ACK_BIT

        LCALL   I2C_DATA_RD
        MOV     DAY,I2C_DATA
        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
;***************************************************
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
;***************************************************
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:      PUSH    ACC
                SETB    I2C_ACK
                MOV     A,I2C_ADD
                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
                CALL    I2C_DELAY
                B       SDA,I2C_SLAVE_2
                LR      I2C_ACK
I2C_SLAVE_2:    CLR     SCL
                OP      ACC
                RET
;***************************************************
I2C_START:      SETB    SCL
                SETB    SDA
                CALL    I2C_DELAY
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                LR      SDA
                CALL    I2C_DELAY
                CLR     SCL
                RET
;*****************************************************
I2C_STOP:       CLR     SDA
                LCALL   I2C_DELAY
                SETB    SCL
                LCALL   I2C_DELAY
                SETB    SDA
                RET
;*****************************************************
I2C_CLK:        LCALL   I2C_DELAY
                SETB    SCL
                LCALL   I2C_DELAY
                CLR     SCL
                RET
;*****************************************************
I2C_ACK_BIT:    CLR     SDA
                LCALL   I2C_DELAY
                LCALL   I2C_CLK
                SETB    SDA
                \RET
;*****************************************************
I2C_NACK_BIT:   SETB    SDA
                LCALL   I2C_DELAY
                LCALL   I2C_CLK
                SETB    SCL
                RET
;*****************************************************
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
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
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
DELAY_100ms_2:  NOP
                NOP
                DJNZ    R6,DELAY_100ms_2
                DJNZ    R7,DELAY_100ms_1
                RET

DELAY_1s:       MOV     R5,#100
DELAY_1s_1:     LCALL   DELAY_10ms
                DJNZ    R5,DELAY_1s_1
                RET
;*****************************************************
;                Define Constant
;*****************************************************
;               0123456789ABCDEF
TITLE_1:        DB      '  Programmable  '
TITLE_2:        DB      ' Electric Applince '
TITLE_3:        DB      '   Controller   '
DATE1:          DB      'DATE:          '
TIME1:          DB      'TIME:          '
LOAD1:          DB      'LD: 1 2 3 4 5 6 7 8 '
SCR_SET_DATE:   DB      ' Date xx: 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      'Mar'
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_SET:        DB      'Set'
SCR_SET_PRO:    DB      'PRO:   :hh/mm:s  '
SETVALUE:       DB      'PR   EV:hh/mm:s  '
SELECT_L1:      DB      '  SELECT LOAD   '
SELECT_L2:      DB      ' NUMBER LOAD (1-8) '
SELECT_LOAD:    DB      '  YOU SELECT:   '
SELECT_P:       DB      '  SELECT PROGRAM  '
CHOICE1:        DB      '  1. EVERY DAY   '
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
CHOICE2:          DB      ' 2. DELAY ON/OFF '
SELECT_PRO:       DB      '  YOU SELECT:  '
CLEAR_LINE:       DB      '               '
PRO1:             DB      'NUM  DATE  TIME  ST'
PRO2:             DB      'EVERY: 00/00: 0 '
KEY_PASSW:        DB      '  KEY PASSWORDS  '
PASSW:            DB      '    "  "    '
FALSE1:           DB      '  YOUR PASSWORDS  '
FALSE2:           DB      ' * NOT CORRECT * '
TIMEDELAY:        DB      '  DELAY : 00 MIN  '
DISABLE_SHOW:     DB      '    DISABLE     '
DISABLE_SHOW2:    DB      '  ALL OF LOADS  '
MANUAL_SHOW:      DB      ' MANUAL LOAD SET  '
ST:               DB      'ST: 0 0 0 0 0 0 0 0'
KEY:              DB      'PRESS 1=SET : 0=CLR '
;************************************ END*****************
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Data Sheet

- **DS1307** 64 x 8 Serial Real Time Clock
- **PCF8574** Remote 8-bit I/O expander for I^2C-bus

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

DALLAS SEMICONDUCTOR

DS1307
64 X 8 Serial Real Time Clock

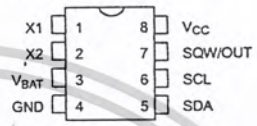
FEATURES

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

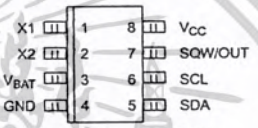
ORDERING INFORMATION

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

PIN ASSIGNMENT



DS1307 8–PIN DIP (300 MIL)



DS1307Z 8–PIN SOIC (150 MIL)

PIN DESCRIPTION

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

DESCRIPTION

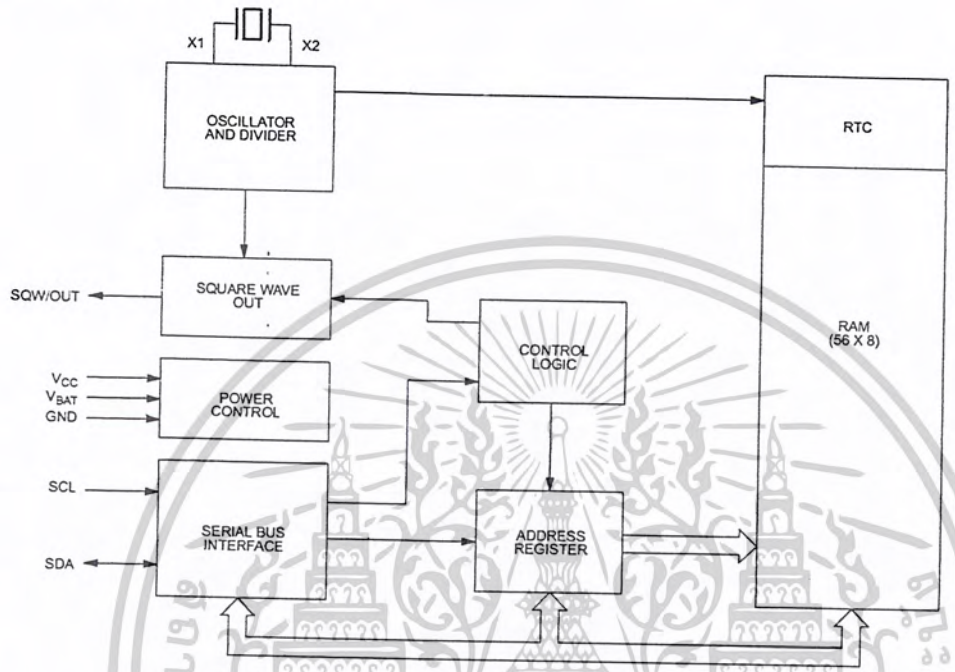
The DS1307 Serial Real Time Clock is a low power full BCD clock calendar plus 56 bytes of nonvolatile SRAM. Address and data are transferred serially via a 2–wire bi–directional bus. The clock/calendar provides seconds, minutes, hours, day, date, month, and year information. The end of the month date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

OPERATION

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

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

DS1307 BLOCK DIAGRAM Figure 1



SIGNAL DESCRIPTIONS

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

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

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

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

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

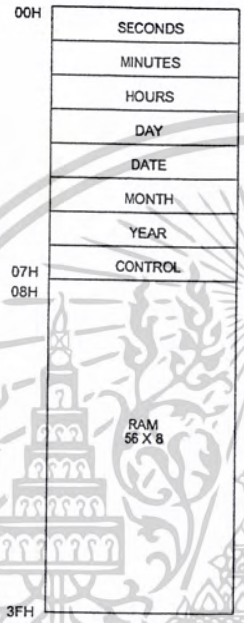
X1, X2 – Connections for a standard 32.768 KHz quartz crystal. The internal oscillator circuitry is designed for operation with a crystal having a specified load capacitance (CL) of 12.5 pF.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

RTC AND RAM ADDRESS MAP

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

DS1307 ADDRESS MAP Figure 2



CLOCK AND CALENDAR

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

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

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

DS1307 TIMEKEEPER REGISTERS Figure 3

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

CONTROL REGISTER

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

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

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

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

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

SQUAREWAVE OUTPUT FREQUENCY Table 1

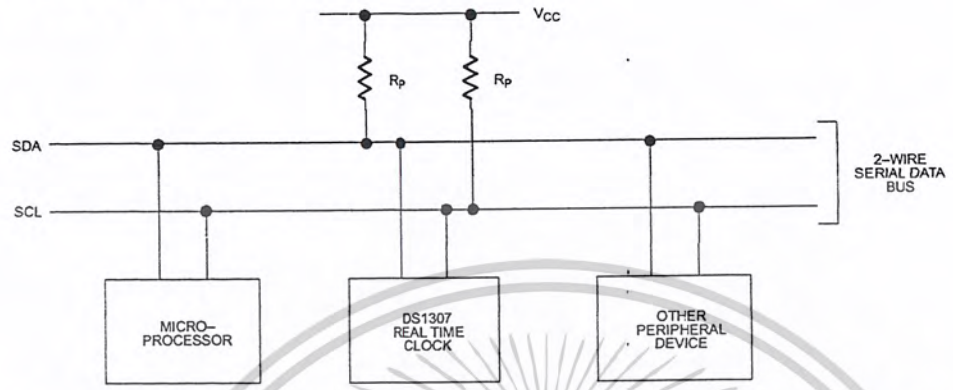
RS1	RS0	SQW OUTPUT FREQUENCY
0	0	1 Hz
0	1	4 KHz
1	0	8 KHz
1	1	32 KHz

2–WIRE SERIAL DATA BUS

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

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

TYPICAL 2–WIRE BUS CONFIGURATION Figure 4



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

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

Accordingly, the following bus conditions have been defined:

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

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

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

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

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

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

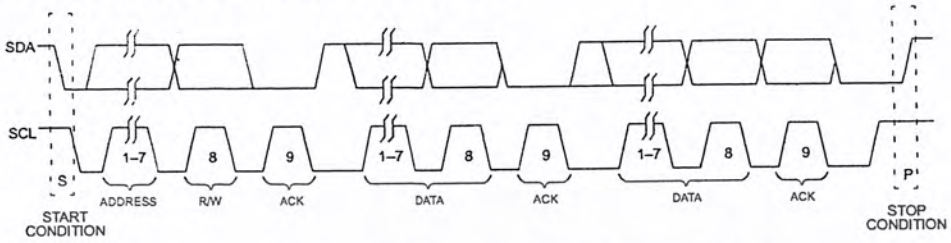
A device that acknowledges must pull down the SDA line during the acknowledge clock pulse in such a way that the SDA line is stable low during the high period of the acknowledge related clock pulse. Of course, setup and hold times must be taken into account. When receiving data from a slave a master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

DATA TRANSFER

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

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

DATA TRANSFER ON 2–WIRE SERIAL BUS Figure 5



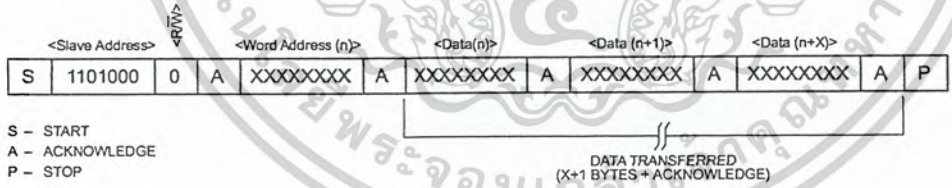
1. Data transfer from a master transmitter to a slave receiver. The first byte transmitted by the master is the slave address. Next follows a number of data bytes. The slave returns an acknowledge bit after each received byte. Data is transfered with the most significant bit (MSB) first.
2. Data transfer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

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

The DS1307 may operate in the following two modes:

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

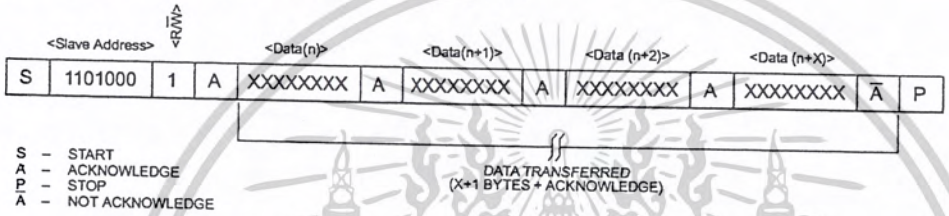
DATA WRITE – SLAVE RECEIVER MODE Figure 6



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

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

DATA READ – SLAVE TRANSMITTER MODE Figure 7



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

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ABSOLUTE MAXIMUM RATINGS*

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

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

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

RECOMMENDED DC OPERATING CONDITIONS

(0°C to 70°C)

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

DC ELECTRICAL CHARACTERISTICS

(0°C to 70°C; V_{CC}=4.5V to 5.5V)

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

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

AC ELECTRICAL CHARACTERISTICS

(0°C to 70°C; V_{CC}=4.5V to 5.5V)

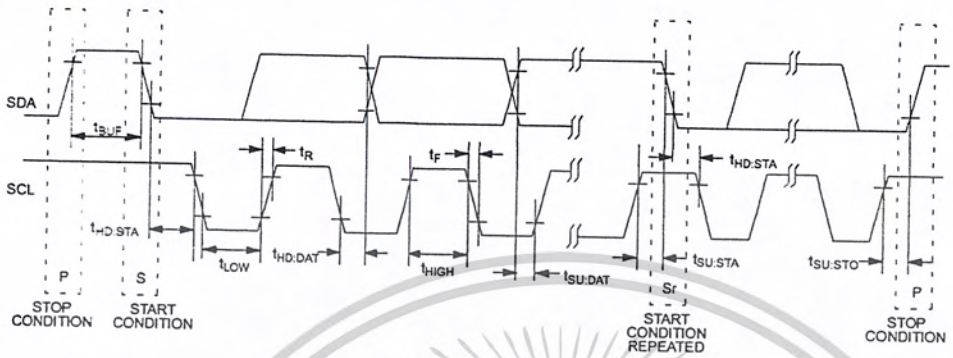
PARAMETER	SYMBOL	MIN	TYP	MAX	UNITS	NOTES
SCL Clock Frequency	f_{SCL}	0		100	KHz	
Bus Free Time Between a STOP and START Condition	t_{BUF}	4.7			µs	
Hold Time (Repeated) START Condition	$t_{HD:STA}$	4.0			µs	5
LOW Period of SCL Clock	t_{LOW}	4.7			µs	
HIGH Period of SCL Clock	t_{HIGH}	4.0			µs	
Set–up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

NOTES:

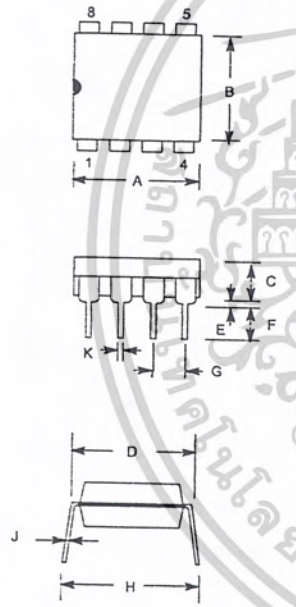
1. All voltages are referenced to ground.
2. Logic zero voltages are specified at a sink current of 5 mA at V_{CC}=4.5V, V_{OL}=GND for capacitive loads.
3. I_{CCS} specified with V_{CC}=5.0V and SDA, SCL=5.0V.
4. V_{CC}=0V, V_{BAT}=3V.
5. After this period, the first clock pulse is generated.
6. A device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

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

TIMING DIAGRAM



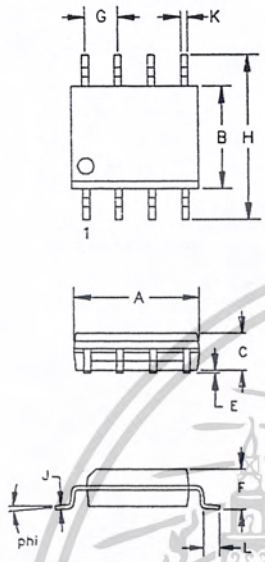
DS1307 64 X 8 SERIAL REAL TIME CLOCK 8–PIN DIP



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

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

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



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

56–G2008–001

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

DATA SHEET

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

Product specification
Supersedes data of September 1994
File under Integrated Circuits, IC12

1997 Apr 02

Philips Semiconductors

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

CONTENTS

1 FEATURES
2 GENERAL DESCRIPTION
3 ORDERING INFORMATION
4 BLOCK DIAGRAM
5 PINNING
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
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 DIP
13.2.1 Soldering by dipping or by wave
13.2.2 Repairing soldered joints
13.3 SO and SSOP
13.3.1 Reflow soldering
13.3.2 Wave soldering
13.3.3 Repairing soldered joints
14 DEFINITIONS
15 LIFE SUPPORT APPLICATIONS
16 PURCHASE OF PHILIPS I²C COMPONENTS

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1 FEATURES

- Operating supply voltage 2.5 to 6 V
- Low standby current consumption of 10 μA maximum
- I²C 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).

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.9.

3 ORDERING INFORMATION

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

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4 BLOCK DIAGRAM

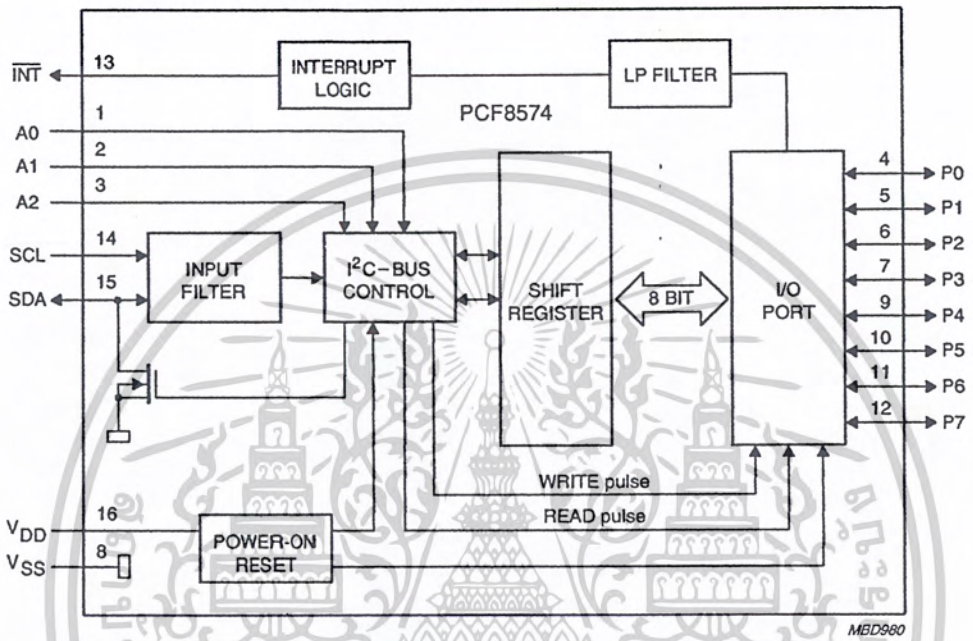


Fig.1 Block diagram (SOT38-1 and SOT162-1).

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5 PINNING

SYMBOL	PIN		DESCRIPTION
	DIP16; SO16	SSOP20	
A0	1	6	address input 0
A1	2	7	address input 1
A2	3	9	address input 2
P0	4	10	quasi-bidirectional I/O 0
P1	5	11	quasi-bidirectional I/O 1
P2	6	12	quasi-bidirectional I/O 2
P3	7	14	quasi-bidirectional I/O 3
V_{SS}	8	15	supply ground
P4	9	16	quasi-bidirectional I/O 4
P5	10	17	quasi-bidirectional I/O 5
P6	11	19	quasi-bidirectional I/O 6
P7	12	20	quasi-bidirectional I/O 7
$\overline{INT}$	13	1	interrupt output (active LOW)
SCL	14	2	serial clock line
SDA	15	4	serial data line
V_{DD}	16	5	supply voltage
n.c.	–	3	not connected
n.c.	–	8	not connected
n.c.	–	13	not connected
n.c.	–	18	not connected

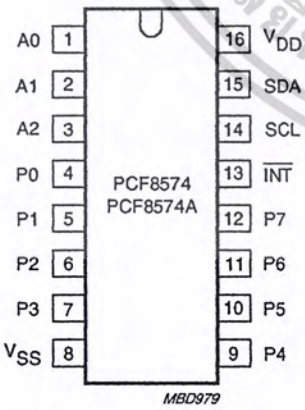


Fig.2 Pin configuration (DIP16; SO16).

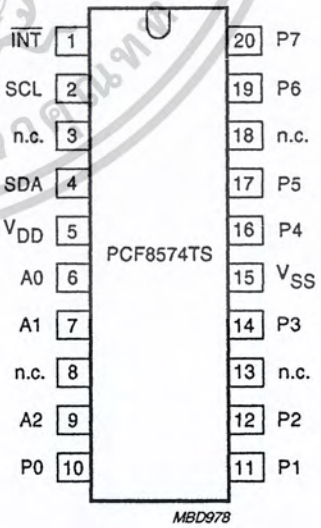


Fig.3 Pin configuration (SSOP20).

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น 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.4).

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.5).

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.6).

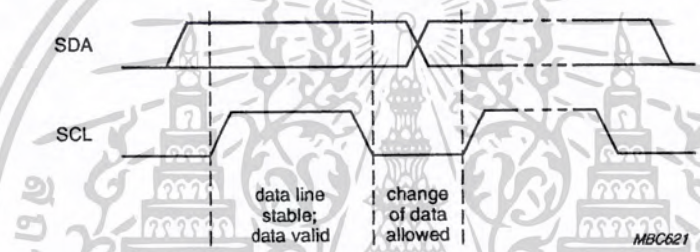


Fig.4 Bit transfer.

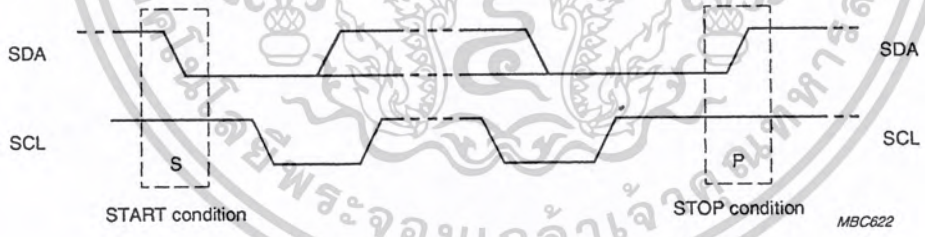


Fig.5 Definition of start and stop conditions.

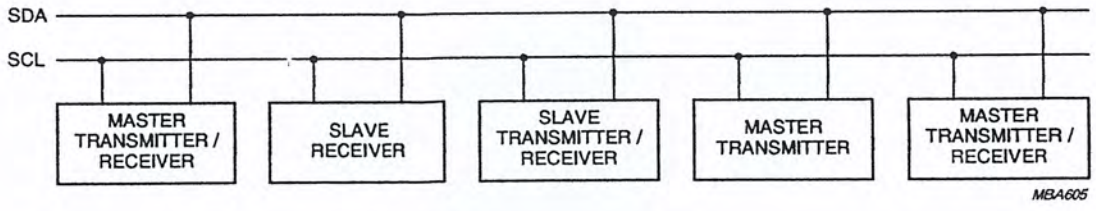


Fig.6 System configuration.

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. 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.

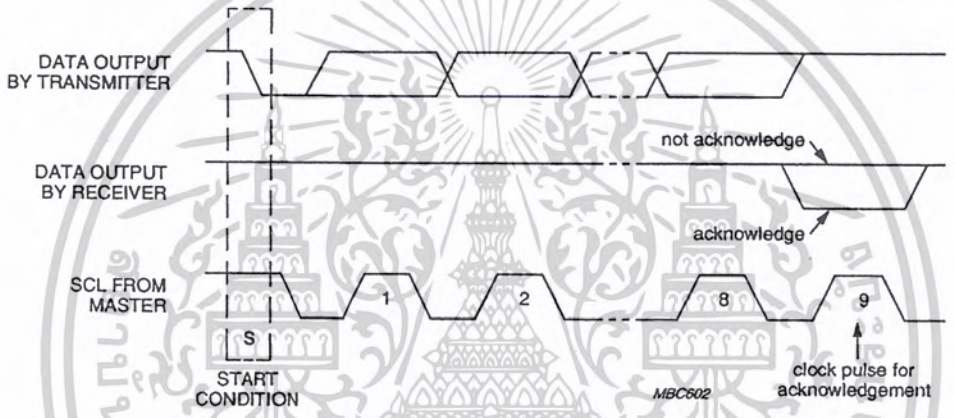


Fig.7 Acknowledgment on the I²C-bus.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

7 FUNCTIONAL DESCRIPTION

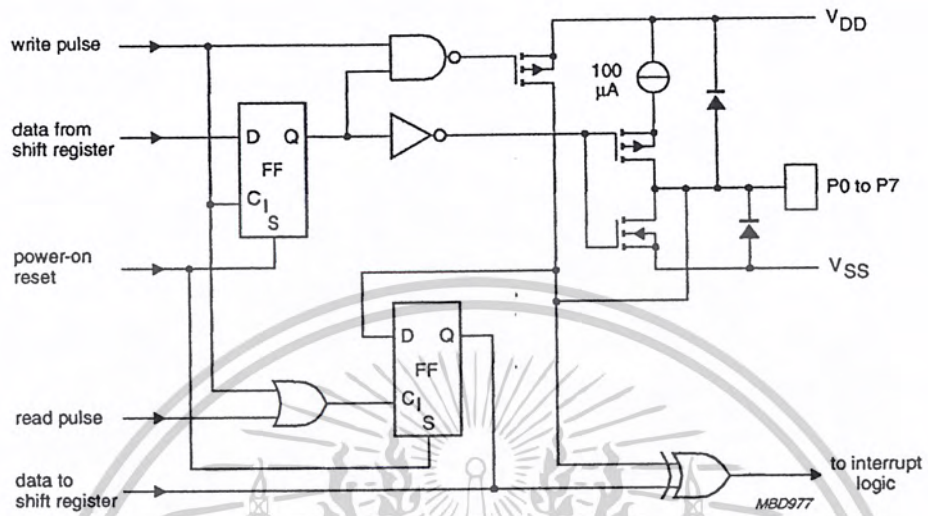
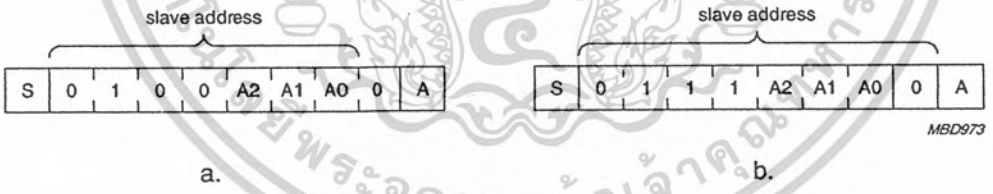


Fig.8 Simplified schematic diagram of each I/O.

7.1 Addressing

For addressing see Figs 9, 10 and 11.



(a) PCF8574.
(b) PCF8574A.

Fig.9 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.11). Output data is transmitted to the port by the WRITE mode (see Fig.10).

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

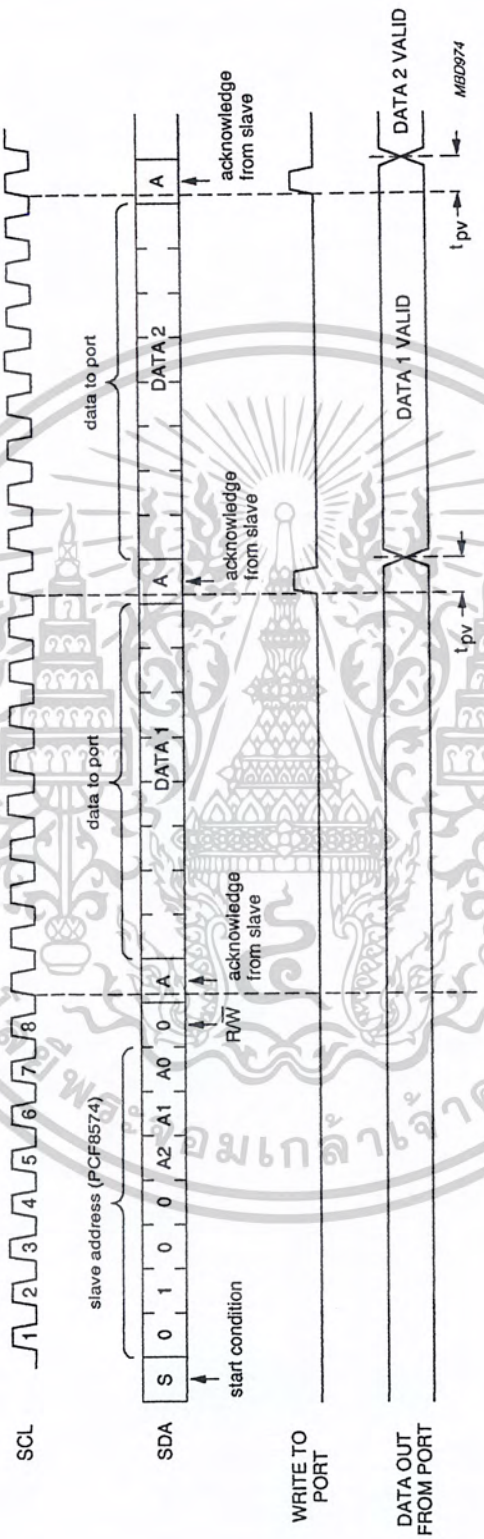
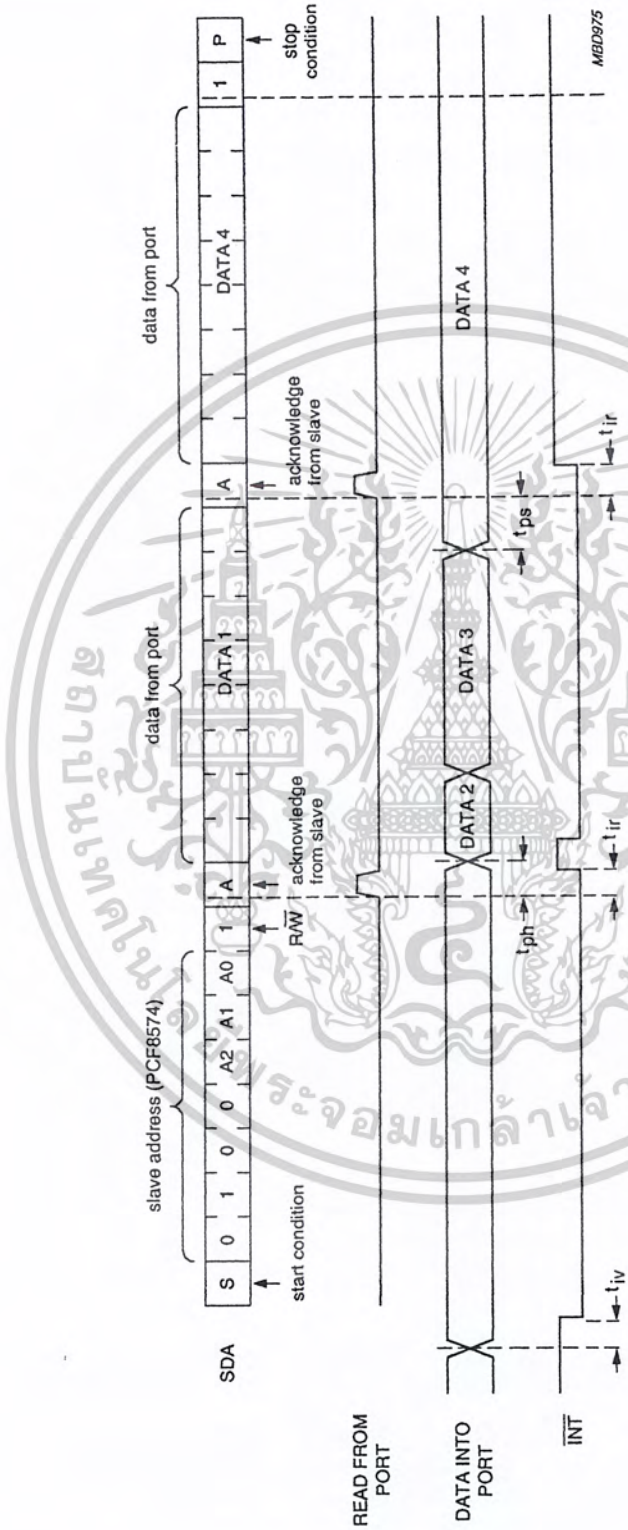


Fig.10 WRITE mode (output).

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



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.11 READ mode (input).

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

7.2 Interrupt (see Figs 12 and 13)

The PCF8574 provides an open drain output ($\overline{INT}$) which can be fed to a corresponding input of the microcontroller. 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 (see Fig.14)

A quasi-bidirectional I/O can be used as an input or output without the use of a control signal for data direction. 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.

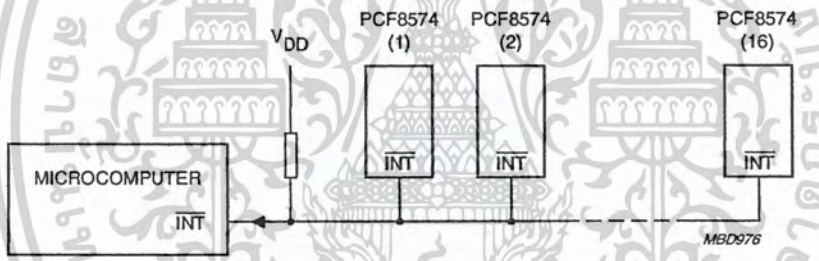


Fig.12 Application of multiple PCF8574s with interrupt.

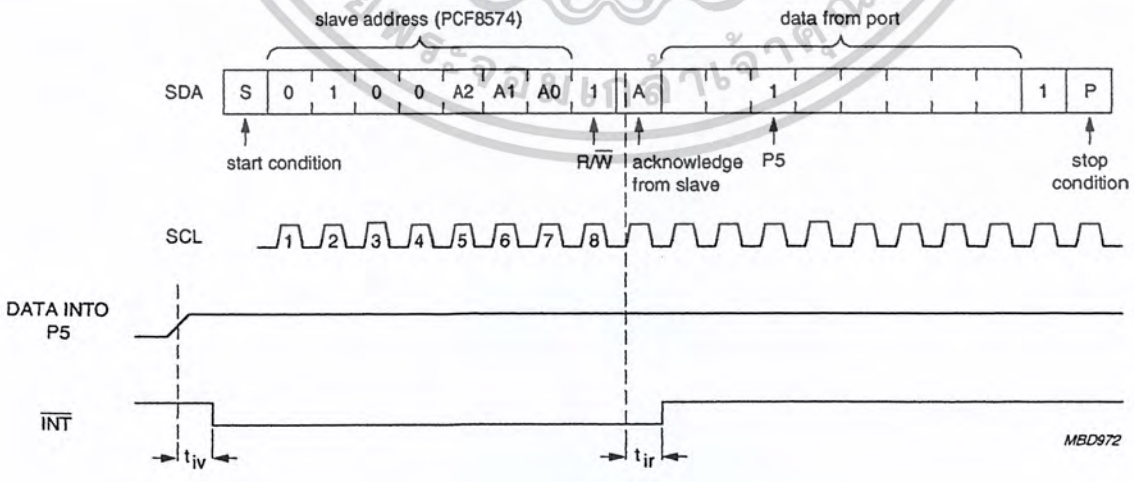


Fig.13 Interrupt generated by a change of input to I/O P5.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

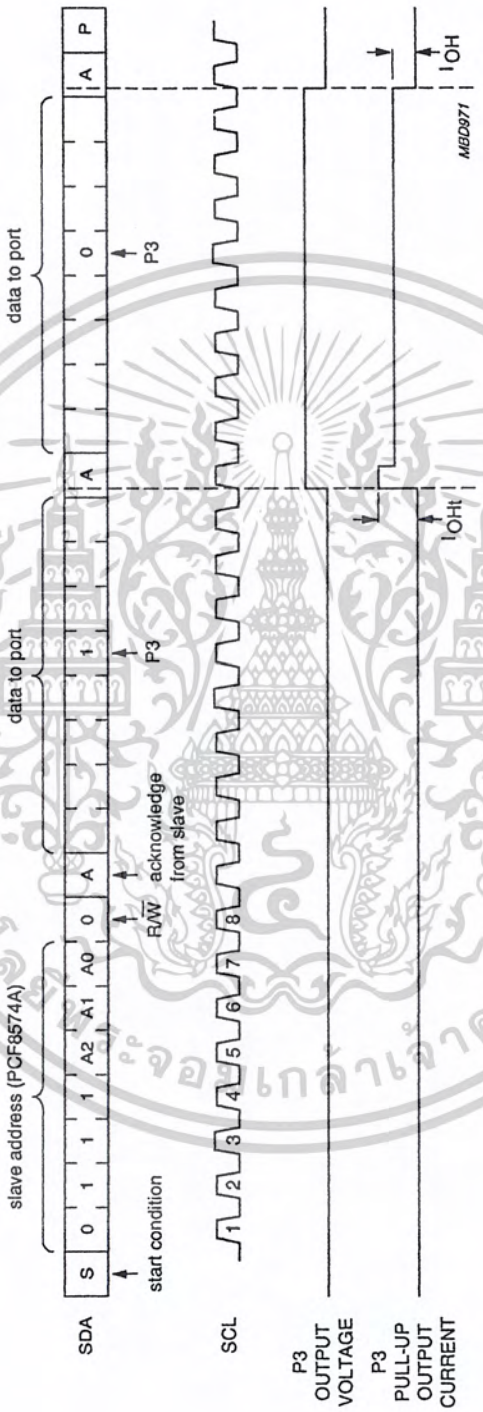


Fig.14 Transient pull-up current I_{OH} while P3 changes from LOW-to-HIGH and back to LOW.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

8 LIMITING VALUES

In accordance with the Absolute Maximum Rating System (IEC 134).

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}	operating 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. Advice can be found in Data Handbook IC12 under *"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.14); $V_{OH} = V_{SS}$; $V_{DD} = 2.5$ V	−	−1	−	mA
C_i	input capacitance		−	−	10	pF
C_o	output capacitance		−	−	10	pF
Port timing; $C_L \leq 100$ pF (see Figs 10 and 11)						
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.13)						
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 with $V_{DD} < V_{POR}$ 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.15; 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}.

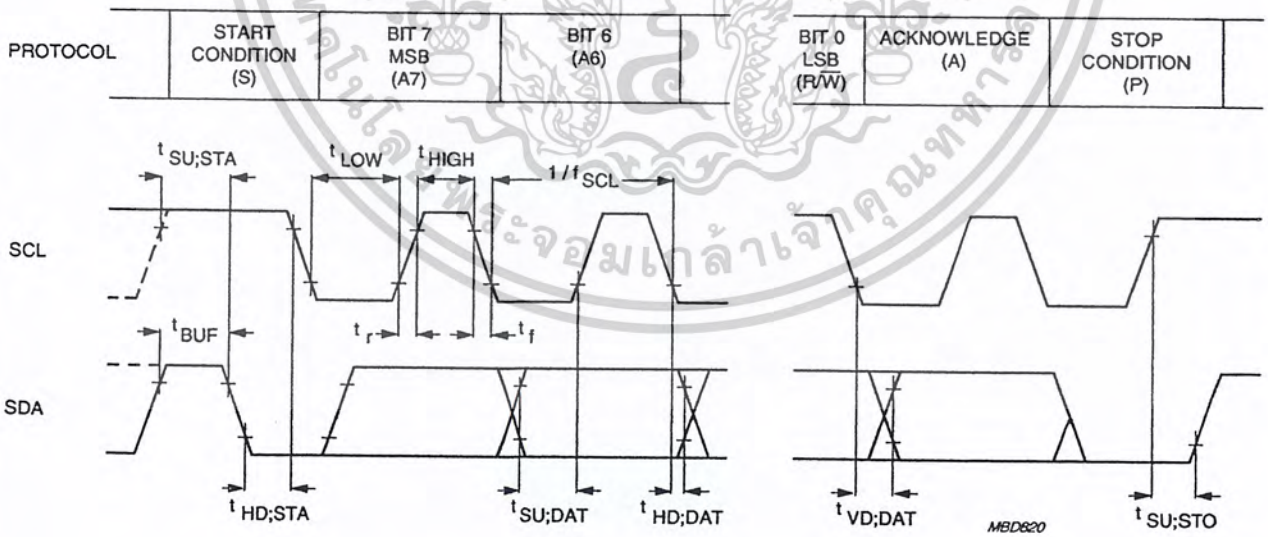
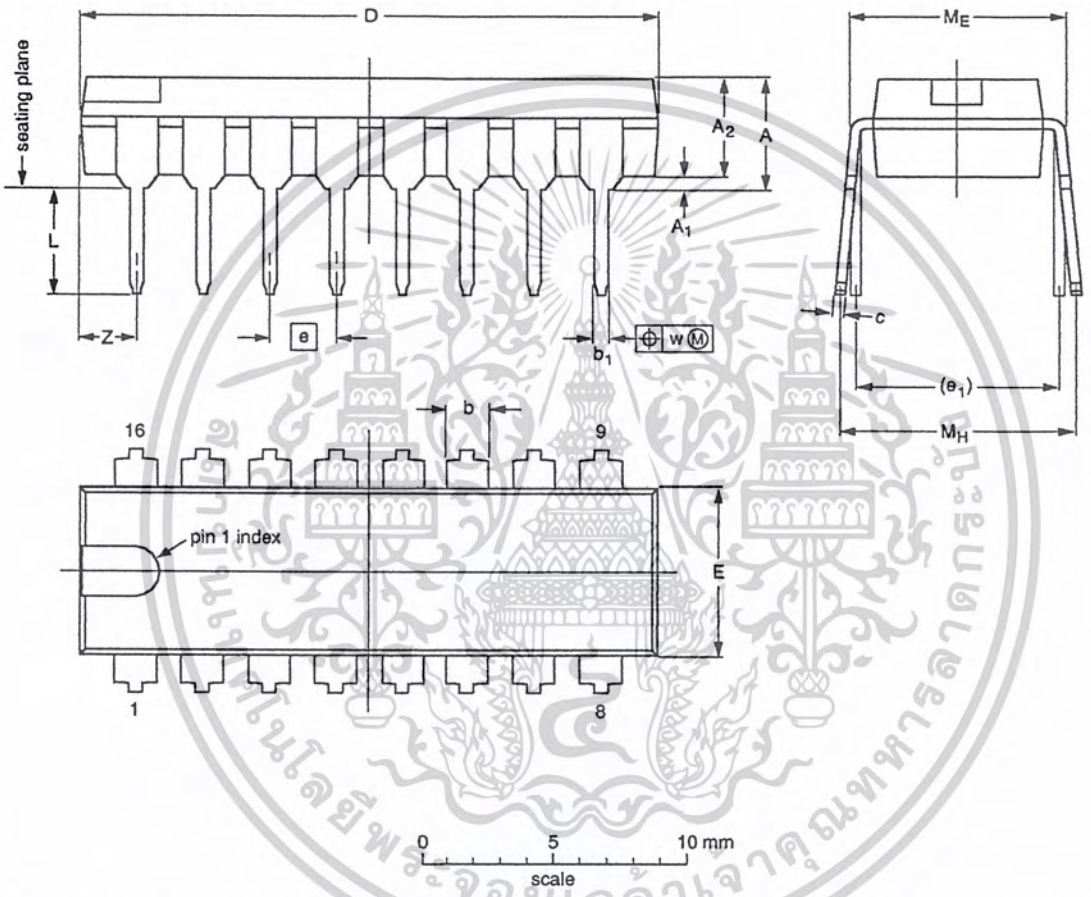


Fig.15 I²C-bus timing diagram.

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

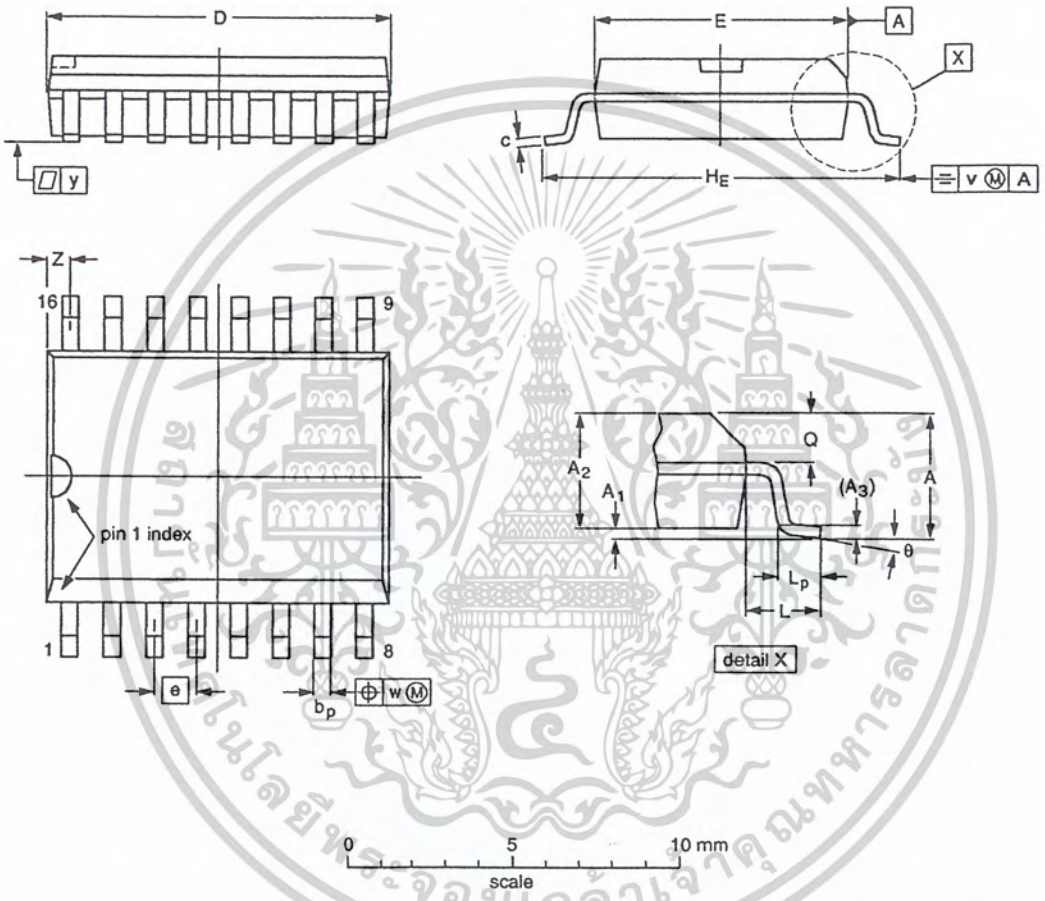
Note

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-001AE				~~92-10-02~~ 95-01-19

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	

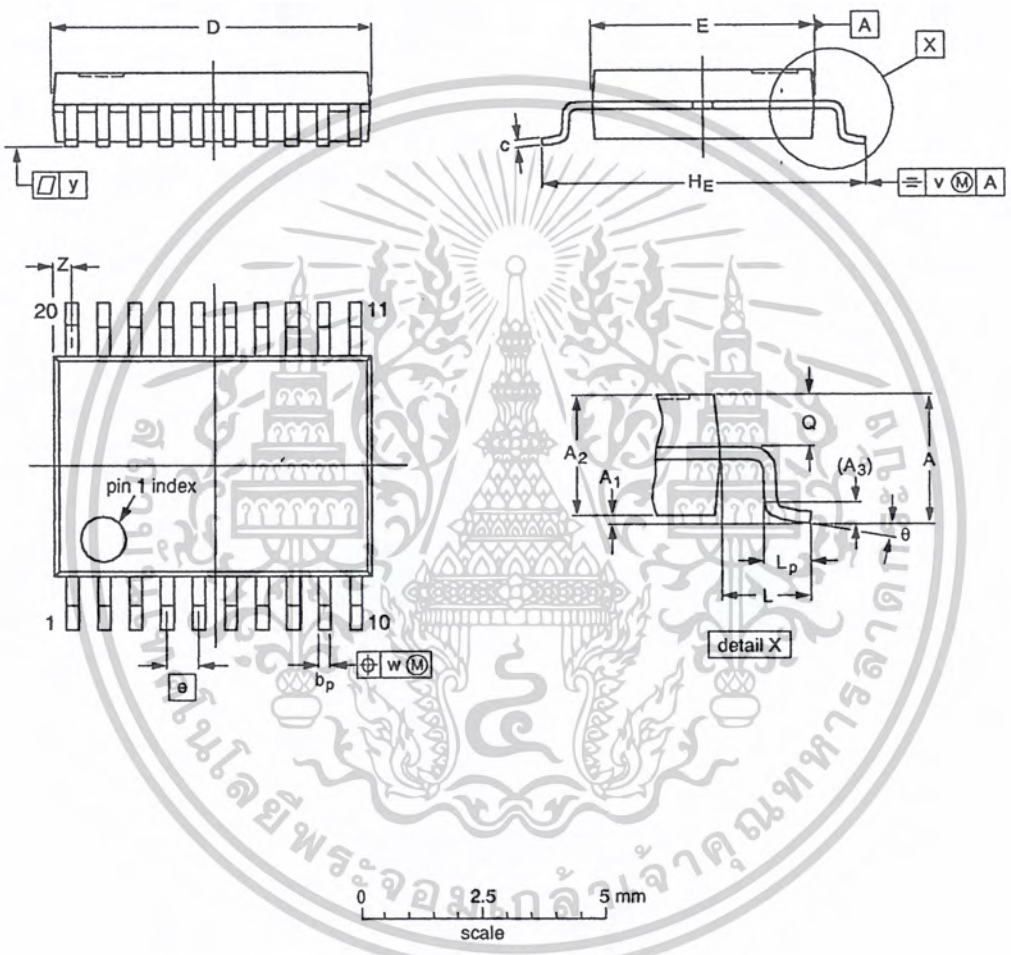
Note

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-013AA				~~95-01-24~~ 97-05-22

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° 0°

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							~~90-04-05~~ 95-02-25

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

13 SOLDERING

13.1 Introduction

There is no soldering method that is ideal for all IC packages. Wave soldering is often preferred when through-hole and surface mounted components are mixed on one printed-circuit board. However, wave soldering is not always suitable for surface mounted ICs, or for printed-circuits with high population densities. In these situations reflow soldering is often used.

This text gives a very brief insight to a complex technology. A more in-depth account of soldering ICs can be found in our *"IC Package Databook"* (order code 9398 652 90011).

13.2 DIP

13.2.1 SOLDERING BY DIPPING OR BY WAVE

The maximum permissible temperature of the solder is 260 °C; solder at this temperature must not be in contact with the joint 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 REPAIRING SOLDERED JOINTS

Apply a low voltage soldering iron (less than 24 V) to the lead(s) of the package, 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 SO and SSOP

13.3.1 REFLOW SOLDERING

Reflow soldering techniques are suitable for all SO and SSOP packages.

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 techniques exist for reflowing; for example, thermal conduction by heated belt. Dwell times vary between 50 and 300 seconds depending on heating method. Typical reflow temperatures range from 215 to 250 °C.

Preheating is necessary to dry the paste and evaporate the binding agent. Preheating duration: 45 minutes at 45 °C.

13.3.2 WAVE SOLDERING

Wave soldering is **not** recommended for SSOP packages. This is because of the likelihood of solder bridging due to closely-spaced leads and the possibility of incomplete solder penetration in multi-lead devices.

If wave soldering cannot be avoided, the following conditions must be observed:

- **A double-wave (a turbulent wave with high upward pressure followed by a smooth laminar wave) soldering technique should be used.**
- **The longitudinal axis of the package footprint must be parallel to the solder flow and must incorporate solder thieves at the downstream end.**

Even with these conditions, only consider wave soldering SSOP packages that have a body width of 4.4 mm, that is SSOP16 (SOT369-1) or SSOP20 (SOT266-1).

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.

Maximum permissible solder temperature is 260 °C, and maximum duration of package immersion in solder is 10 seconds, if cooled to less than 150 °C within 6 seconds. 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 REPAIRING SOLDERED JOINTS

Fix the component by first soldering two diagonally-opposite end leads. Use only a low voltage soldering iron (less than 24 V) 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.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

14 DEFINITIONS

Data sheet status	
Objective specification	This data sheet contains target or goal specifications for product development.
Preliminary specification	This data sheet contains preliminary data; supplementary data may be published later.
Product specification	This data sheet contains final product specifications.
Limiting values	
Limiting values given are in accordance with the Absolute Maximum Rating System (IEC 134). 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	
Where application information is given, it is advisory and does not form part of the specification.	

15 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 customers using or selling these products for use in such applications do so at their own risk and agree to fully indemnify Philips for any damages resulting from such improper use or sale.

16 PURCHASE OF PHILIPS I²C COMPONENTS

Purchase of Philips I²C components conveys a license under the Philips' I²C patent to use the components in the I²C system provided the system conforms to the I²C specification defined by Philips. This specification can be ordered using the code 9398 393 40011.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Philips Semiconductors – a worldwide company

Argentina: see South America

Australia: 34 Waterloo Road, NORTH RYDE, NSW 2113,
Tel. +61 2 9805 4455, Fax. +61 2 9805 4466

Austria: Computerstr. 6, A-1101 WIEN, P.O. Box 213,
Tel. +43 1 60 101, Fax. +43 1 60 101 1210

Belarus: Hotel Minsk Business Center, Bld. 3, r. 1211, Volodarski Str. 6,
220050 MINSK, Tel. +375 172 200 733, Fax. +375 172 200 773

Belgium: see The Netherlands

Brazil: see South America

Bulgaria: Philips Bulgaria Ltd., Energoproject, 15th floor,
51 James Bourchier Blvd., 1407 SOFIA,
Tel. +359 2 689 211, Fax. +359 2 689 102

Canada: PHILIPS SEMICONDUCTORS/COMPONENTS,
Tel. +1 800 234 7381

China/Hong Kong: 501 Hong Kong Industrial Technology Centre,
72 Tat Chee Avenue, Kowloon Tong, HONG KONG,
Tel. +852 2319 7888, Fax. +852 2319 7700

Colombia: see South America

Czech Republic: see Austria

Denmark: Prags Boulevard 80, PB 1919, DK-2300 COPENHAGEN S,
Tel. +45 32 88 2636, Fax. +45 31 57 1949

Finland: Sinikalliontie 3, FIN-02630 ESPOO,
Tel. +358 9 615800, Fax. +358 9 61580/xxx

France: 4 Rue du Port-aux-Vins, BP317, 92156 SURESNES Cedex,
Tel. +33 1 40 99 6161, Fax. +33 1 40 99 6427

Germany: Hammerbrookstraße 69, D-20097 HAMBURG,
Tel. +49 40 23 53 60, Fax. +49 40 23 536 300

Greece: No. 15, 25th March Street, GR 17778 TAVROS/ATHENS,
Tel. +30 1 4894 339/239, Fax. +30 1 4814 240

Hungary: see Austria

India: Philips INDIA Ltd, Shivsagar Estate, A Block, Dr. Annie Besant Rd.
Worli, MUMBAI 400 018, Tel. +91 22 4938 541, Fax. +91 22 4938 722

Indonesia: see Singapore

Ireland: Newstead, Clonskeagh, DUBLIN 14,
Tel. +353 1 7640 000, Fax. +353 1 7640 200

Israel: RAPAC Electronics, 7 Kehilat Saloniki St, TEL AVIV 61180,
Tel. +972 3 645 0444, Fax. +972 3 649 1007

Italy: PHILIPS SEMICONDUCTORS, Piazza IV Novembre 3,
20124 MILANO, Tel. +39 2 6752 2531, Fax. +39 2 6752 2557

Japan: Philips Bldg 13-37, Kohnan 2-chome, Minato-ku, TOKYO 108,
Tel. +81 3 3740 5130, Fax. +81 3 3740 5077

Korea: Philips House, 260-199 Itaewon-dong, Yongsan-ku, SEOUL,
Tel. +82 2 709 1412, Fax. +82 2 709 1415

Malaysia: No. 76 Jalan Universiti, 46200 PETALING JAYA, SELANGOR,
Tel. +60 3 750 5214, Fax. +60 3 757 4880

Mexico: 5900 Gateway East, Suite 200, EL PASO, TEXAS 79905,
Tel. +9-5 800 234 7381

Middle East: see Italy

Netherlands: Postbus 90050, 5600 PB EINDHOVEN, Bldg. VB,
Tel. +31 40 27 82785, Fax. +31 40 27 88399

New Zealand: 2 Wagener Place, C.P.O. Box 1041, AUCKLAND,
Tel. +64 9 849 4160, Fax. +64 9 849 7811

Norway: Box 1, Manglerud 0612, OSLO,
Tel. +47 22 74 8000, Fax. +47 22 74 8341

Philippines: Philips Semiconductors Philippines Inc.,
106 Valero St. Salcedo Village, P.O. Box 2108 MCC. MAKATI,
Metro MANILA, Tel. +63 2 816 6380, Fax. +63 2 817 3474

Poland: Ul. Lukiska 10, PL 04-123 WARSZAWA,
Tel. +48 22 612 2831, Fax. +48 22 612 2327

Portugal: see Spain

Romania: see Italy

Russia: Philips Russia, Ul. Usatcheva 35A, 119048 MOSCOW,
Tel. +7 095 755 6918, Fax. +7 095 755 6919

Singapore: Lorong 1, Toa Payoh, SINGAPORE 1231,
Tel. +65 350 2538, Fax. +65 251 6500

Slovakia: see Austria

Slovenia: see Italy

South Africa: S.A. PHILIPS Pty Ltd., 195-215 Main Road Martindale,
2092 JOHANNESBURG, P.O. Box 7430 Johannesburg 2000,
Tel. +27 11 470 5911, Fax. +27 11 470 5494

South America: Rua do Rocio 220, 5th floor, Suite 51,
04552-903 São Paulo, SÃO PAULO - SP, Brazil,
Tel. +55 11 821 2333, Fax. +55 11 829 1849

Spain: Balmes 22, 08007 BARCELONA,
Tel. +34 3 301 6312, Fax. +34 3 301 4107

Sweden: Kottbygatan 7, Akalla, S-16485 STOCKHOLM,
Tel. +46 8 632 2000, Fax. +46 8 632 2745

Switzerland: Allmendstrasse 140, CH-8027 ZÜRICH,
Tel. +41 1 488 2686, Fax. +41 1 481 7730

Taiwan: Philips Semiconductors, 6F, No. 96, Chien Kuo N. Rd., Sec. 1,
TAIPEI, Taiwan Tel. +886 2 2134 2870, Fax. +886 2 2134 2874

Thailand: PHILIPS ELECTRONICS (THAILAND) Ltd.,
209/2 Sanpavuth-Bangna Road Prakanong, BANGKOK 10260,
Tel. +66 2 745 4090, Fax. +66 2 398 0793

Turkey: Talatpasa Cad. No. 5, 80640 GÜLTEPE/ISTANBUL,
Tel. +90 212 279 2770, Fax. +90 212 282 6707

Ukraine: PHILIPS UKRAINE, 4 Patrice Lumumba str., Building B, Floor 7,
252042 KIEV, Tel. +380 44 264 2776, Fax. +380 44 268 0461

United Kingdom: Philips Semiconductors Ltd., 276 Bath Road, Hayes,
MIDDLESEX UB3 5BX, Tel. +44 181 730 5000, Fax. +44 181 754 8421

United States: 811 East Arques Avenue, SUNNYVALE, CA 94088-3409,
Tel. +1 800 234 7381

Uruguay: see South America

Vietnam: see Singapore

Yugoslavia: PHILIPS, Trg N. Pasica 5/v, 11000 BEOGRAD,
Tel. +381 11 625 344, Fax.+381 11 635 777

For all other countries apply to: Philips Semiconductors, Marketing & Sales Communications,
Building BE-p, P.O. Box 218, 5600 MD EINDHOVEN, The Netherlands, Fax. +31 40 27 24825

Internet: http://www.semiconductors.philips.com

© Philips Electronics N.V. 1997

SCA53

All rights are reserved. Reproduction in whole or in part is prohibited without the prior written consent of the copyright owner.

The information presented in this document does not form part of any quotation or contract, is believed to be accurate and reliable and may be changed without notice. No liability will be accepted by the publisher for any consequence of its use. Publication thereof does not convey nor imply any license under patent- or other industrial or intellectual property rights.

Printed in The Netherlands 417067/1200/02/pp24 Date of release: 1997 Apr 02 Document order number: 9397 750 01758

Let's make things better.

Philips Semiconductors

PHILIPS

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

ขอขอบคุณ ผศ.ดร.เกียรติศักดิ์ คมวัชระ อาจารย์ที่ปรึกษา และอาจารย์ทุกท่านที่ได้ประสิทธิ์ประศาสน์ความรู้ในแขนงต่างๆ รวมทั้งให้คำปรึกษา คำแนะนำ และ แนวทางการแก้ไขปัญหาต่างๆที่เกิดขึ้น จนโครงงานนี้สำเร็จลุล่วงไปด้วยดี

ขอขอบพระคุณ บิดา มารดา และครอบครัว ที่ได้ให้การสนับสนุนในทุกๆด้าน

ขอขอบคุณ พี่ๆ น้องๆ และ เพื่อนๆ ทุกคนที่ได้ช่วยเหลือในด้านต่างๆ ด้วยดีเสมอมา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หนังสืออ้างอิง

1. ชัยวัฒน์ ลิ้มพรจิตรวิไล, วรพจน์ กรแก้ววัฒนกุล, "เรียนรู้และปฏิบัติการไมโครคอนโทรลเลอร์MCS-51". บริษัท อินโนเวตีฟ เอ็กเพอริเมนต์ จำกัด, 475 หน้า
2. ธีรวัฒน์ ประกอบผล, "การประยุกต์ใช้งานไมโครคอนโทรลเลอร์", สมาคมส่งเสริมเทคโนโลยี (ไทย – ญี่ปุ่น), 155 หน้า, 2540
3. สุนทร วิทูสุรพจน์, "การใช้งานไมโครคอนโทรลเลอร์ตระกูล 8051", บริษัท ซีเอ็ดยูเคชั่น จำกัด (มหาชน). 179 หน้า, 2537

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้