สำนักวิทยบริการฯ พระจอมเกล้าลาดกระบัง

โครงงานชุดควบคุมแผงเซลล์แสงอาทิตย์เคลื่อนตำแหน่งตามดวงอาทิตย์

(SOLAR CELL TRACKING)

T042331

โดย

นายเกรียงศักดิ์ กังวานสกุลทอง

นายจิรวัสส์ ตรีมิ่งมิตร

เลขหมู่.....
เลขทะเบียน 42331
วัน, เดือน, ปี 17 พ.ค. 2545

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

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมระบบควบคุม

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2543

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์ปีการศึกษา 2543

สาขาวิชาวิศวกรรมระบบควบคุม

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง โครงงานชุดควบคุมแผงเซลล์แสงอาทิตย์เคลื่อนตำแหน่งตามดวงอาทิตย์
SOLAR CELL TRACKING

ผู้จัดทำ 1. นายเกรียงศักดิ์ กังวานสกุลทอง
2. นายจิรวัสส์ ตรีมิ่งมิตร

..... อาจารย์ที่ปรึกษา

(อาจารย์ถาวร เบญจนราสุทธิ์)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โครงงานชุดควบคุมแผงเซลล์แสงอาทิตย์เคลื่อนตำแหน่งตามดวงอาทิตย์

นายเกรียงศักดิ์ กังวานสกุลทอง 40010073
นายจิรวัสส์ ตรีมิ่งมิตร 40010133
อาจารย์ถาวร เบญจนราสุทธิ์ อาจารย์ที่ปรึกษา
ปีการศึกษา 2543

บทคัดย่อ

ปัจจุบันมีการพัฒนาเทคโนโลยีเกี่ยวกับแผงเซลล์แสงอาทิตย์ แต่ประสิทธิภาพในการรับแสงยังไม่ดีพอ เนื่องจากการติดตั้งแผงเซลล์แสงอาทิตย์ไว้อยู่กับที่ ดังนั้นโครงงานนี้จะเป็นการควบคุมแผงเซลล์แสงอาทิตย์ให้เคลื่อนที่ติดตามตำแหน่งของดวงอาทิตย์ เพื่อให้ได้รับพลังงานแสงอาทิตย์มากว่าการติดตั้งอยู่กับที่

การควบคุมเริ่มจากภาคตรวจจับแสง ซึ่งมีเซ็นเซอร์ติดตั้งอยู่บนแผงเซลล์แสงอาทิตย์ ส่งสัญญาณผ่านวงจรเปรียบเทียบ เพื่อทำการเปรียบเทียบสัญญาณแล้วส่งผลให้ ไมโครคอนโทรลเลอร์เบอร์ AT89C51 ที่ต่ออยู่กับไอซีเรียลไทม์คล็อก และจอแสดงผล LCD โดยไมโครคอนโทรลเลอร์ถูกโปรแกรมด้วยภาษาแอสเซมบลี เพื่อควบคุมการปิดเปิด และทิศทางการหมุนของมอเตอร์ให้เคลื่อนที่ตามตำแหน่งของดวงอาทิตย์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

SOLAR CELL TRACKING

Mr.Kreangsak Kungwansakultong 40010073

Mr.Jirawat Threemingmid 40010133

Mr.Taworn Benjanarasuth Advisor

Education Year 2000

Abstract

This project presents a solar cell controller for controlling the solar cell to track the sun so that the received solar energy is greater than the one received from the stable position solar cell. The sunlight detector will detect the sunlight from the sensors fixed on the solar cell and sends the signal to the comparator circuit. The compared signal is sent to microcontroller number AT89C51 which is connected to real time clock (RTC) and LCD display. The microcontroller will be program by assembly language for controlling ON-OFF and directive rotation of motors to move the solar cell following the position of the sun.

The experimental results of the controller show that the controller can control the model to track the position of the sun.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

	หน้า
บทคัดย่อภาษาไทย	A
บทคัดย่อภาษาอังกฤษ	B
สารบัญ	C
สารบัญรูป	D
บทที่ 1 บทนำ	1
บทที่ 2 ทฤษฎีพื้นฐาน	2
2.1 ไมโครคอนโทรลเลอร์ (Microcontroller)	2
2.2 ไอซีสร้างฐานเวลาจริงหรือเรียลไทม์คล็อก (Real Time Clock – RTC)	2
บทที่ 3 ลักษณะของโครงงาน	4
3.1 บทนำหลักการออกแบบระบบเซลล์แสงอาทิตย์	4
3.2 ทฤษฎีพื้นฐาน	5
3.3 หลักการออกแบบ	8
3.3.1 ภาคตรวจจับและเปรียบเทียบทิศทาง	8
2.3.2 ภาคควบคุม	12
2.3.3 ภาคต้นกำลัง	12
3.4 โครงสร้างและการออกแบบทางกล	13
3.5 โครงสร้างและการออกแบบส่วนอิเล็กทรอนิกส์	18
3.6 รายการอุปกรณ์	22
บทที่ 4 ผลการทดลอง	26
บทที่ 5 บทสรุปและวิจารณ์	29
ภาคผนวก	
กิตติกรรมประกาศ	
หนังสืออ้างอิง	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูปภาพ

	หน้า
รูปที่ 2.1 แสดงการจัดวางขาต่างๆของชิปไมโครคอนโทรลเลอร์ AT89C51	2
รูปที่ 2.2 การจัดขาของไอซีเรียลไทม์คล็อก DS1307	3
รูปที่ 2.3 แสดงการเชื่อมต่อไมโครคอนโทรลเลอร์กับไอซีเรียลไทม์คล็อก	3
รูปที่ 3.1 บล็อกไดอะแกรมของระบบควบคุมการติดตามดวงอาทิตย์แบบวงรอบปิด	5
รูปที่ 3.2 การเปรียบเทียบทิศทางโดยการบังให้เกิดเงา	6
รูปที่ 3.3 ความแตกต่างที่มุมตกกระทบต่าง ๆ และความสูงของฉากกั้นที่ต่างๆกัน	10
รูปที่ 3.4 แสดงลักษณะของรังสีโดยมีฉากกั้น 2 แผ่น	11
รูปที 3.5 ลักษณะรังสีเปรียบเทียบกับจำนวนแผ่นกั้น	11
รูปที่ 3.6 แสดงโครงสร้างทางกลของโครงงาน	13
รูปที่ 3.7 โครงสร้างส่วนควบคุมการเคลื่อนที่แนวตะวันออก-ตะวันตก	14
รูปที่ 3.8 โครงสร้างส่วนควบคุมการเคลื่อนที่แนวระนาบ	17
รูปที่ 3.9 วงจรทั้งหมดในโครงงานนี้	18
รูปที่ 3.10 วงจรภาคตรวจจับ เปรียบเทียบ และภาคต้นกำลัง	18
รูปที่ 3.11 วงจรภาคควบคุม	20
รูปที่ 3.12 วงจรแหล่งจ่ายไฟ	20
รูปที่ 3.13 การต่อวงจรรวมทั้งหมด	21
รูปที่ 3.14 โฟลว์ชาร์ตการทำงานของโปรแกรม	24
รูปที่ 4.1 แสดงผลที่ได้จากการทดลอง 2 วัน	28

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

1.1 ความสำคัญของโครงงาน

พลังงานแสงอาทิตย์เป็นพลังงานรูปแบบหนึ่ง ที่สามารถแปลงเป็นพลังงานไฟฟ้าเพื่อนำไปใช้งานได้โดยการใช้เซลล์แสงอาทิตย์ (Solar Cell) จะสามารถแปลงพลังงานได้มากน้อยแค่ไหนนั้น ขึ้นอยู่กับปัจจัยหลายอย่าง มุมในการตกกระทบของแสงอาทิตย์บนแผงเซลล์แสงอาทิตย์ก็เป็นปัจจัยหนึ่ง (โดยการตกกระทบในแนวตั้งฉากจะทำให้เกิดการแปลงพลังงานมากที่สุด) ซึ่งการควบคุมให้แผงเซลล์แสงอาทิตย์เคลื่อนที่ไปสู่ตำแหน่งที่ให้พลังงานมากที่สุดจะทำให้เกิดการแปลงพลังงานที่มีประสิทธิภาพมากที่สุดด้วย

1.2 วัตถุประสงค์ของโครงงาน

- เพื่อให้แผงเซลล์แสงอาทิตย์สามารถหมุนหาตำแหน่ง ที่ให้ค่ากำลังไฟฟ้าสูงสุดที่ช่วงเวลาต่างๆ
- ประหยัดพลังงานไฟฟ้าที่ป้อนเข้าแผงวงจรในการหมุนหาตำแหน่ง ที่ให้กำลังไฟฟ้าสูงสุดของแผงเซลล์แสงอาทิตย์

1.3 ขอบเขตของโครงงาน

- สามารถใช้ไมโครคอนโทรลเลอร์ ควบคุมการทำงานของชุดควบคุมการหมุนของแผงเซลล์แสงอาทิตย์เพื่อหาตำแหน่งที่ให้ค่ากำลังไฟฟ้าสูงสุด
- สามารถใช้ไอซีเรียลไทม์คล็อกควบคุมเวลา เพื่อควบคุมการทำงานของชุดควบคุมการหมุนของแผงเซลล์แสงอาทิตย์ผ่านไมโครคอนโทรลเลอร์ได้

1.4 วิธีดำเนินการ

- ศึกษาการทำงานและการใช้งานของไมโครคอนโทรลเลอร์, ไอซีเรียลไทม์คล็อก และอุปกรณ์ที่เกี่ยวกับโครงงานนี้
- ศึกษาและเขียนโปรแกรมสำหรับควบคุมไมโครคอนโทรลเลอร์

1.5 ประโยชน์ที่คาดว่าจะได้รับจากการทำวิจัย

- เข้าใจการใช้งานของไมโครคอนโทรลเลอร์, ไอซีเรียลไทม์คล็อก และอุปกรณ์ที่เกี่ยวกับโครงงานนี้
- สามารถเขียนโปรแกรมเพื่อควบคุมการทำงานของไมโครคอนโทรลเลอร์ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

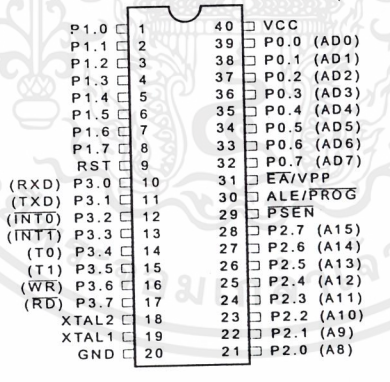
บทที่ 2
ทฤษฎีพื้นฐาน

2.1 ไมโครคอนโทรลเลอร์ (Microcontroller)

ในโครงงานนี้ เราจะใช้ไมโครคอนโทรลเลอร์ เบอร์ AT89C51 ซึ่งมีคุณสมบัติพื้นฐานที่สำคัญ คือ

- หน่วยความจำโปรแกรมภายในขนาด 4 กิโลไบต์
- หน่วยความจำแบบภายใน (Internal Ram) จำนวน 128 ไบต์
- พอร์ตอินพุท/เอาท์พุทแบบขนานจำนวน 32 เส้น ซึ่งสามารถแยกทำงานได้อย่างอิสระ
- มีวงจรนับ/จับเวลาขนาด 16 บิต จำนวนสองวงจร
- มีวงจรควบคุมการอินเตอร์รัพต์จากแหล่งกำเนิดสัญญาณ 6 ประเภท

ไมโครคอนโทรลเลอร์นี้ จะมีขาทั้งหมด 40 ขา ดังแสดงในรูปที่ 2.1 รายละเอียดต่างๆของไมโครคอนโทรลเลอร์นี้ได้แสดงไว้ในภาคผนวกแล้ว



รูปที่ 2.1 แสดงการจัดวางขาต่างๆของชิปไมโครคอนโทรลเลอร์ AT89C51

2.2 ไอซีสร้างฐานเวลาจริงหรือเรียลไทม์คล็อก (Real Time Clock - RTC)

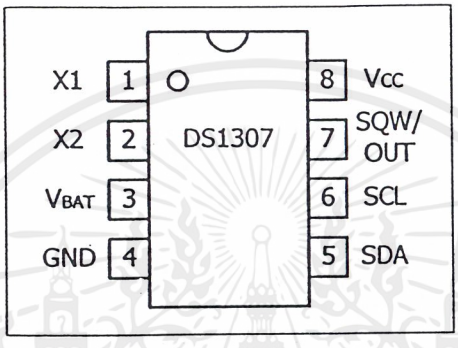
ไอซีเรียลไทม์คล็อก มีหน้าที่สร้างฐานเวลาจริงให้แก่ระบบไมโครคอนโทรลเลอร์จะให้ข้อมูลเกี่ยวกับเวลาทั้งหมด (วินาที นาที ชั่วโมง วันที่ เดือน และปี) ในโครงงานนี้เราจะใช้ไอซีเรียลไทม์คล็อก เบอร์ DS1307 ซึ่งมีคุณสมบัติที่สำคัญ ดังนี้

- เป็นไอซีเรียลไทม์คล็อกให้ข้อมูลตั้งแต่วินาทีจนถึงปี รวมถึงการปรับวันในปีอธิกสุรทินด้วยสามารถให้ข้อมูลเวลาได้อย่างเที่ยงตรงถึงปีคริสตศักราช 2100

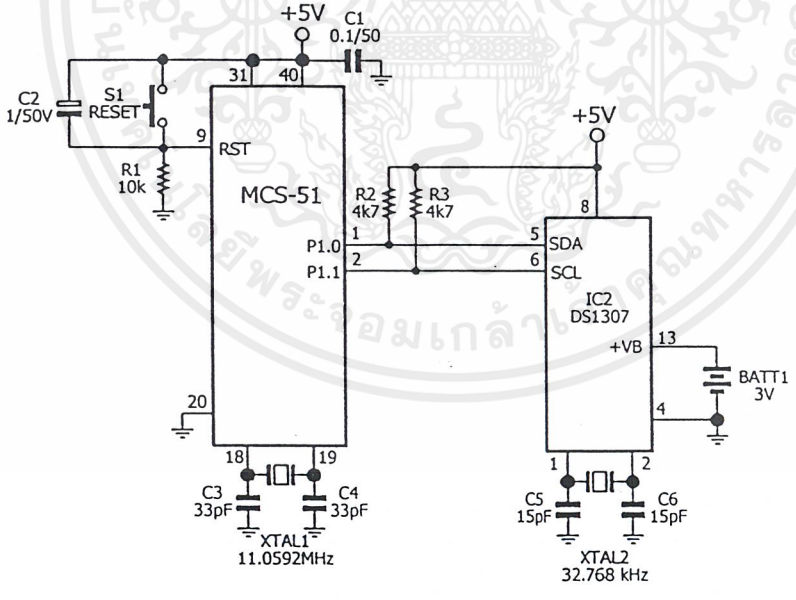
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- มีหน่วยความจำนอนโวลาไทล์แรม 56 ไบต์อยู่ภายใน สามารถเก็บข้อมูลทั่วไปได้
- ใช้การเชื่อมต่อแบบระบบบัส I^2C
- มีวงจรตรวจจับไฟเลี้ยงต่ำหรือหายไปอย่างอัตโนมัติ และสามารถรักษาข้อมูลเวลาไว้ได้แม้ไม่มีไฟเลี้ยงไอซี

ไอซีเรียลไทม์คล็อกนี้ จะมีขา 8 ขา ดังรูปที่ 2.2 ส่วนรายละเอียดการใช้งานได้แสดงไว้ในภาคผนวกแล้ว



รูปที่ 2.2 การจัดขาของไอซีเรียลไทม์คล็อก DS1307



รูปที่ 2.3 แสดงการเชื่อมต่อไมโครคอนโทรลเลอร์กับไอซีเรียลไทม์คล็อก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3
ลักษณะของโครงงาน

3.1 บทนำหลักการออกแบบระบบเซลล์แสงอาทิตย์

การออกแบบระบบเซลล์แสงอาทิตย์ จะประกอบด้วยการเลือกโครงสร้างที่เหมาะสมของระบบ และการเลือกขนาดของส่วนประกอบต่างๆให้เหมาะสมกับสภาพของอากาศ และความต้องการของภาระทางไฟฟ้า โดยจะต้องคำนึงถึงราคาต่อหน่วยพลังงาน ความเชื่อถือได้ของระบบ และปัญหาการบำรุงรักษาระบบเป็นหลัก ในช่วง 10 ปีที่ผ่านมา เซลล์แสงอาทิตย์ได้รับการพัฒนาให้มีราคาต่อวัตต์ลดลงหลายเท่าตัว แต่อย่างไรก็ดี ปัญหาการใช้เซลล์แสงอาทิตย์ในการผลิตกระแสไฟฟ้าในปัจจุบันก็ยังคงอยู่ที่ราคาของแผงเซลล์แสงอาทิตย์เป็นส่วนใหญ่ จากปัญหาดังกล่าว ทำให้มีการคิดหาวิธีที่จะลดจำนวนเซลล์แสงอาทิตย์ที่ใช้ลง เช่น การใช้วิธีการรวมแสงให้มีความเข้มสูง (100 - 200 เท่า) หรือ แบบที่มีความเข้มปานกลาง (20 - 50 เท่า) ซึ่งในทั้งสองกรณีจะต้องใช้เซลล์แสงอาทิตย์ที่ออกแบบเป็นพิเศษให้มีความต้านทานอนุกรมต่ำ และสามารถทำงานได้ที่อุณหภูมิสูง ระบบรวมแสงนี้จะมีความยุ่งยากมากจึงเหมาะสำหรับระบบที่มีขนาดใหญ่เท่านั้น นอกจากจะใช้วิธีที่กล่าวมาแล้ว อาจใช้อีกวิธี คือ การปรับปรุงประสิทธิภาพของระบบ เช่น การปรับปรุงการแมทช์ชิ่งระหว่างเซลล์แสงอาทิตย์กับดวงอาทิตย์ เพื่อให้เซลล์แสงอาทิตย์ได้รับพลังงานแสงอาทิตย์สูงสุดตลอดเวลา อันเป็นการเพิ่มปริมาณของพลังงานออกของระบบให้มากขึ้น

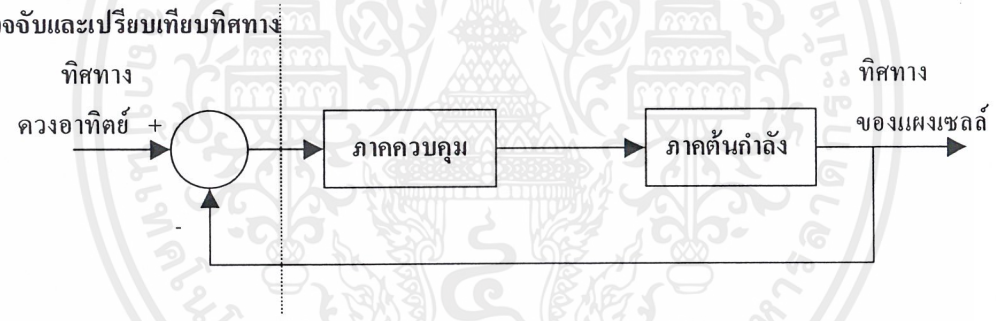
ได้มีการคำนวณเปรียบเทียบปริมาณของแสงอาทิตย์ที่ตกบนระนาบของตัวรับแสง ในกรณีที่มีการติดตามดวงอาทิตย์แบบ 2 แกน (แนวระนาบ และ แนวตะวันออก-ตะวันตก) กรณีที่มีการติดตามดวงอาทิตย์แบบแกนเดียว (ตามแนวตะวันออก-ตะวันตก) และกรณีที่ติดตั้งอยู่กับที่ โดยไม่คิดผลของสภาพอากาศท้องถิ่น พบว่าการใช้ระบบติดตามดวงอาทิตย์แบบ 2 แกน จะได้รับพลังงานเฉลี่ยต่อปีเพิ่มขึ้นจากกรณีที่มีการติดตามดวงอาทิตย์แบบ 1 แกน ประมาณ 5 - 10 % และได้รับพลังงานเพิ่มจากกรณีที่ติดตั้งอยู่กับที่ประมาณ 50 % ในกรณีที่เราคิดผลของสภาพอากาศของท้องถิ่น ความแตกต่างของปริมาณแสงที่ได้รับในทั้งสามกรณีจะลดลง อันเนื่องมาจากการลดลงของปริมาณแสงที่มาจากดวงอาทิตย์โดยตรง (DIRECT RADIATION) สำหรับกรุงเทพฯนั้น ปริมาณของแสงที่มาจากดวงอาทิตย์โดยตรงเฉลี่ยต่อปี จะมีค่าประมาณครึ่งหนึ่งของปริมาณแสงที่ได้รับทั้งหมด งานวิจัยนี้มีวัตถุประสงค์ ที่จะศึกษาความเป็นไปได้ โดยจะศึกษาระบบติดตามดวงอาทิตย์สำหรับเซลล์แสงอาทิตย์แบบแผงราบ และปริมาณที่เพิ่มของพลังงานของระบบจากผลการติดตามดวงอาทิตย์ โดยคำนึงถึงพลังงานที่ต้องใช้ในการหมุนแผงเซลล์แสงอาทิตย์ตามดวงอาทิตย์ด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2 ทฤษฎีพื้นฐาน

<u>หลักการของระบบ</u> ระบบที่ใช้เป็นระบบควบคุมแบบวงรอบปิด ซึ่ง เป็นระบบควบคุมตำแหน่งแบบที่มีการป้อนกลับ (Feedback Position Control System) อย่างหนึ่งซึ่งมีตำแหน่งหรือทิศทางดวงอาทิตย์เป็นตัวควบคุมตำแหน่ง หรือ ทิศทางของแผงเซลล์แสงอาทิตย์ ระบบนี้จะมีการตรวจจับทิศทางของแผงเซลล์แสงอาทิตย์ เพื่อนำมาเปรียบเทียบกับทิศทางจริงของดวงอาทิตย์ตลอดเวลา เมื่อมีความแตกต่างของทิศทางทั้งสองขึ้น ตัวเปรียบเทียบทิศทางจะส่งสัญญาณออกไปควบคุมให้แผงเซลล์แสงอาทิตย์หมุนไปในทิศทางของดวงอาทิตย์เพื่อลดความแตกต่างระหว่างทิศทางทั้งสอง ความผิดพลาดในการติดตามดวงอาทิตย์ของระบบนี้ จะขึ้นอยู่กับความไวของตัวเปรียบเทียบทิศทางของแผงเซลล์แสงอาทิตย์กับทิศทางจริงของดวงอาทิตย์

<u>โครงสร้างของระบบ</u> ระบบควบคุมการติดตามดวงอาทิตย์แบบวงรอบปิด แสดงในรูปที่ 3.1 ประกอบด้วยส่วนต่างๆ ที่สำคัญดังนี้ คือ



รูปที่ 3.1 บล็อกไดอะแกรมของระบบควบคุมการติดตามดวงอาทิตย์แบบวงรอบปิด

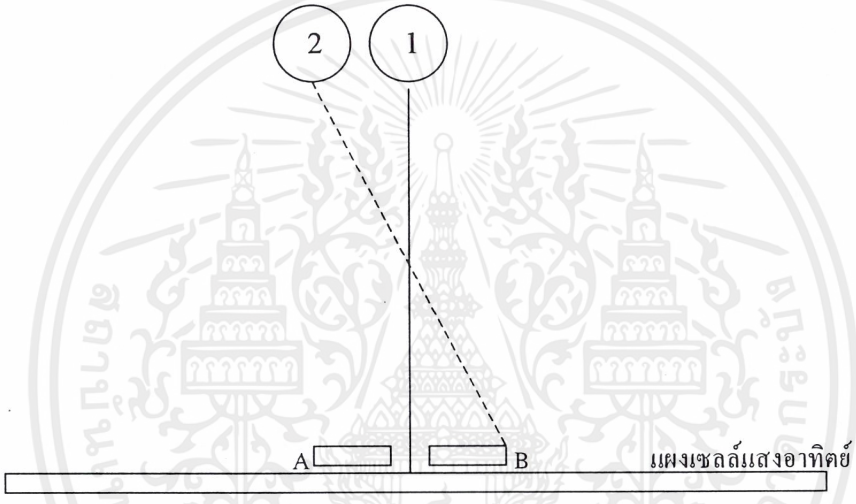
- <u>ภาคตรวจจับและเปรียบเทียบทิศทาง</u>

ภาคตรวจจับ จะประกอบด้วยอุปกรณ์ที่ใช้ในการตรวจจับทิศทางของดวงอาทิตย์ และแผงเซลล์โดยทั่วไปจะเป็นอุปกรณ์ที่มีความไวต่อแสง เช่น โฟโต้ทรานซิสเตอร์ LDR (Light Decreasing Resistance) เป็นต้น แต่ในโครงงานนี้เราจะใช้ LDR สำหรับการเปรียบเทียบทิศทางจะอาศัยการเปรียบเทียบความเข้มของแสงที่ตกกระทบอุปกรณ์ไวแสง โดยใช้วิธีการบังให้เกิดเงา

การบังให้เกิดเงา (Shadow Sensor) วิธีการนี้ เราจะวางอุปกรณ์ไวแสงซึ่งมีลักษณะเหมือนกันไว้สองข้างของวัตถุทึบแสงซึ่งวางตั้งฉากกับแผงเซลล์แสงอาทิตย์ ดังแสดงในรูปที่ 3.2 ในระบบที่มีการติดตามในแนวตะวันออก-ตะวันตก อุปกรณ์ไวแสง A และ B จะวางอยู่ในแนวตะวันออก-ตะวันตก แต่ถ้าต้องการให้มีการติดตามในแนวระนาบด้วย เราก็ต้องเพิ่มอุปกรณ์ที่มีลักษณะเหมือนกันอีก 1 ชุด แต่จะต้องวางอุปกรณ์ไวแสง A และ B ให้อยู่ในแนวระนาบ จากในรูปที่ 3.2 เราจะเห็น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ว่า ถ้าแผงเซลล์แสงอาทิตย์ทำมุมตั้งฉากกับแนวทิศทางของดวงอาทิตย์ดังในตำแหน่ง 1 ทั้งอุปกรณ์ไวแสง A และ B จะได้รับความเข้มแสงเท่ากัน แต่ถ้าเกิดมีความคลาดเคลื่อนดังในตำแหน่ง 2 อุปกรณ์ไวแสง B จะถูกบังไม่ให้ได้รับแสง ซึ่งลักษณะเช่นนี้ จะเป็นการชี้บอกให้ทราบว่าดวงอาทิตย์อยู่เยื้องไปทางทิศใดของแนวตั้งฉากของแผงเซลล์แสงอาทิตย์ การเปรียบเทียบโดยวิธีนี้ความแม่นยำจะขึ้นอยู่กับความยาวของอุปกรณ์ทำให้เกิดเงาโดยความแม่นยำจะสูงขึ้นเมื่อความยาวเพิ่มขึ้น วิธีการนี้จะให้ความแม่นยำได้ถึง 0.1 องศา



รูปที่ 3.2 การเปรียบเทียบทิศทางโดยการบังให้เกิดเงา

ภาคเปรียบเทียบทิศทาง เป็นส่วนที่รับค่าที่วัดได้จากเซนเซอร์มาเปรียบเทียบ ค่าเอาท์พุทที่ได้จะส่งต่อให้กับ ไอซีDRIVE เพื่อควบคุมทิศทางการหมุนของมอเตอร์ ในโครงงานนี้จะใช้ ไอซีเบอร์ LM324 ซึ่งภายในไอซีจะประกอบด้วยออปแอมป์จำนวน 4 ตัว ส่วนรายละเอียดการใช้งานได้แสดงไว้ในภาคผนวกแล้ว

ลักษณะสัญญาณออก ของตัวเปรียบเทียบทิศทางอาจจะเป็น สัญญาณขั้น หรือสัญญาณต่อเนื่องก็ได้ซึ่งแบ่งออกเป็น 3 ลักษณะ คือ

- สัญญาณศูนย์ เป็นสัญญาณที่แสดงว่าความแตกต่าง ของทิศทางของแผงเซลล์กับทิศทาง ของดวงอาทิตย์น้อยกว่าค่าต่ำสุด ที่ตัวเปรียบเทียบทิศทางจะสามารถตรวจจับได้
- สัญญาณบวก เป็นสัญญาณที่แสดงว่าตำแหน่ง ของดวงอาทิตย์อยู่เยื้องไปทางทิศตะวันตกของแนวตั้งฉากของแผง (หรืออาจจะเป็นตรงข้ามก็ได้)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- สัญญาณลบ เป็นสัญญาณที่แสดงว่าตำแหน่ง ของดวงอาทิตย์อยู่เยื้องไปทางทิศตะวันออกของแนวตั้งฉากของแผงเซลล์ (หรืออาจจะเป็นตรงข้ามก็ได้)

ขนาดของสัญญาณออกของตัวเปรียบเทียบทิศทางนั้น นอกจากจะขึ้นอยู่กับความแตกต่างของทิศทางของแผงเซลล์แสงอาทิตย์กับทิศทางของดวงอาทิตย์แล้ว ยังจะขึ้นอยู่กับความเข้มของแสงตรงจากดวงอาทิตย์ ดังนั้น ความแม่นยำของตัวเปรียบเทียบทิศทางจะขึ้นอยู่กับปริมาณของแสงตรงจากดวงอาทิตย์ด้วย สาเหตุนี้จะทำให้อุปกรณ์ตรวจจับและเปรียบเทียบทิศทางของดวงอาทิตย์กับแผงเซลล์แสงอาทิตย์ไม่สามารถจะทำงานได้ในขณะที่มีเมฆมาบังดวงอาทิตย์ หรืออาจจะมีการหลงทางชั่วขณะได้

- ภาคควบคุม

ภาคนี้ส่วนประกอบที่สำคัญ คือ ไมโครคอนโทรลเลอร์ และไอซีเรียลไทม์คล็อก โดยรวมแล้วภาคนี้จะทำหน้าที่ สร้างฐานเวลาจริงให้แก่ระบบ รับสัญญาณจากตัวเปรียบเทียบและส่งสัญญาณไปควบคุมภาคต้นกำลัง เพื่อที่จะขับเคลื่อนแผงเซลล์แสงอาทิตย์ไปในตำแหน่งที่ได้รับพลังงานสูงสุด นอกจากนี้ยังมีหน้าที่ควบคุมภาคต้นกำลังให้หยุดการทำงานเมื่อแผงเซลล์แสงอาทิตย์เจอตำแหน่งที่ได้รับพลังงานสูงสุดแล้วจะเริ่มทำงานอีกครั้งเมื่อเวลาที่หลักนาที เป็น 0 นาที และ 30 นาที หน้าที่อีกอย่างที่สำคัญ คือ จะสั่งให้แผงเซลล์แสงอาทิตย์ส่วนบนหมุนกลับมายังด้านตรงข้ามเมื่อถึงเวลา 6 โมงเย็นเพื่อรอรับแสงอาทิตย์ในตอนเช้า ระบบก็จะหยุดทำงานทุกอย่างจนถึง 8 โมงเช้าระบบจะเริ่มทำงานอีกครั้ง

- ภาคต้นกำลัง

ภาคนี้ประกอบด้วย ไอซีDRIVE เบอร์ L293D และชุดมอเตอร์ (มอเตอร์กระแสตรง เฟืองทดรอบ) มีหน้าที่ ขับเคลื่อนแผงเซลล์แสงอาทิตย์ โดยได้รับการควบคุมจากภาคควบคุม มอเตอร์ที่ใช้โดยทั่วไปจะเป็นมอเตอร์กระแสตรง เนื่องจากควบคุมความเร็วได้ง่าย และเหมาะกับแหล่งที่มีอยู่คือ กระแสไฟตรงจากแบตเตอรี่ ส่วนสาเหตุที่ต้องมีเกียร์ทดรอบ ก็เพื่อลดความเร็วของมอเตอร์และเป็นการเพิ่มแรงบิดในการขับเคลื่อนแผงเซลล์แสงอาทิตย์ด้วย

ข้อดีและข้อเสียของระบบวงรอบปิด

ข้อดี

- มีความแม่นยำสูง
- ไม่ต้องการข้อมูลเกี่ยวกับทางโคจรของดวงอาทิตย์
- สามารถปรับทิศทางได้ตามฤดูกาล
- ระบบสามารถเริ่มต้นได้เอง เนื่องจากสามารถตรวจจับตำแหน่งดวงอาทิตย์ได้เอง
- ไม่มีการสะสมความคลาดเคลื่อน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ข้อเสีย

- ต้องการใช้อุปกรณ์ที่มีความไวต่อแสงที่มีราคาแพง เนื่องจากต้องบรรจุในภาชนะโปร่งแสงที่ต้องปิดมิดชิด เพื่อป้องกันความเสียหายที่อาจจะเกิดจากอากาศ และความชื้น เนื่องจากอุปกรณ์ดังกล่าวต้องอยู่กลางแจ้งตลอดเวลา
- ระบบ 1 ระบบจะต้องใช้ตัวเปรียบเทียบทิศทาง และภาคควบคุมอย่างละชุดเราไม่สามารถใช้ตัวเปรียบเทียบทิศทาง และภาคควบคุมหนึ่งชุดกับชุดแผงเซลล์แสงอาทิตย์หลาย ๆ ชุดได้
- จะต้องใช้มอเตอร์ที่มีกำลังสูง เนื่องจากการเปลี่ยนตำแหน่งในบางครั้งอาจจะต้องทำอย่างรวดเร็ว เช่น ในภาวะที่มีเมฆมาก ๆ จะทำให้อุปกรณ์ไวแสงไม่สามารถตรวจจับตำแหน่งดวงอาทิตย์ได้ ระบบจะหยุดอยู่กับที่ และเมื่อเมฆผ่านไป ระบบจะต้องปรับตำแหน่ง เพื่อให้ระนาบของแผงเซลล์ตั้งฉากรับแนวทิศทางของดวงอาทิตย์อย่างรวดเร็ว
- ระบบนี้อาจจะมีการหลงในบางขณะ เช่น ในภาวะที่มีเมฆมาก ซึ่งทำให้ต้องหมุนแผงเซลล์ไป มา อันเป็นการสิ้นเปลืองพลังงาน

3.3 หลักการออกแบบ

โครงงานการศึกษาตำแหน่ง ของแผงเซลล์แสงอาทิตย์จะเลือกหาตำแหน่ง ที่มีความเข้มของแสงมากที่สุด เพื่อที่จะทำให้แผงเซลล์แสงอาทิตย์จ่ายกำลังไฟฟ้าออกมาได้อย่างเต็มที่และวงจรควบคุมการขับเคลื่อนของแผงเซลล์แสงอาทิตย์นี้ มีบล็อกไดอะแกรมแสดงดังรูปที่ 3.1

3.3.1 <u>ภาคตรวจจับและเปรียบเทียบทิศทาง</u> อุปกรณ์ที่ใช้ในการตรวจจับทิศทางของดวงอาทิตย์กับแผงเซลล์แสงอาทิตย์จะใช้ LDR ซึ่งเป็นอุปกรณ์ที่เปลี่ยนแปลงค่าต้านทานตามความเข้มแสงที่ตกกระทบ โดยจะยึดติดกับแผงเซลล์แสงอาทิตย์

เนื่องจากดวงอาทิตย์อยู่ห่างจากโลกมาก รังสีที่ตกกระทบลงพื้นโลกจึงมีลักษณะเกือบขนานกัน หรือไม่สามารถอ้างว่าแสงเหล่านี้มาจากแหล่งกำเนิดจุดเดียว (Point Source) ดังนั้นสูตรที่อ้างว่า ความเข้มของแสงแปรผันกับ 1 / (ระยะทาง)^2 จึงนำมาใช้ไม่ได้ หรือในทางกลับกันถ้าถือว่าดวงอาทิตย์เป็นแหล่งกำเนิดจุดเดียวระยะทางที่แสงตกลงมาบนชุดเซนเซอร์ เกือบเท่ากันเพราะฉะนั้น เมื่อดวงอาทิตย์ทำมุมเอียงไปเท่าไรค่าความเข้มของแสงบน LDR แต่ละตัวจะมีค่าเท่าๆกัน

ดังนั้นจึงจำเป็นต้องหาวิธีการอื่น ๆ ที่ทำให้ค่าความเข้มของแสงบน LDR แต่ละตัวมีค่าแตกต่างกัน เมื่อรังสีจากดวงอาทิตย์ตกลงบนชุดเซนเซอร์ ทำมุมเอียงออกไปการพิจารณาเกี่ยวกับความ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เข้มของแสงนั้น นิยมแทนความเข้มนั้นด้วย ฟลักซ์ (Flux) (จำนวนเส้นแรงที่ตกกระทบลงบนพื้นที่ 1 หน่วย) เมื่อพิจารณาจากรูป 3.3 จะเห็นได้ว่าเมื่อรังสีตกกระทบทำมุมเอียง จำนวนฟลักซ์ที่ตกกระทบพื้น A หลังฉากกั้น B จะน้อยกว่าด้านหน้า นอกจากนี้แล้วโดยที่รังสีตกกระทบทำมุมเดิม แต่ความสูงของฉากกั้น B เพิ่มขึ้น ความเข้มของแสงจากฉากกั้น B ลงมายังพื้น A จำนวนฟลักซ์ที่สะท้อนกลับลงไปนั้น จะมากหรือน้อยขึ้นอยู่กับลักษณะผิวของฉากกั้น

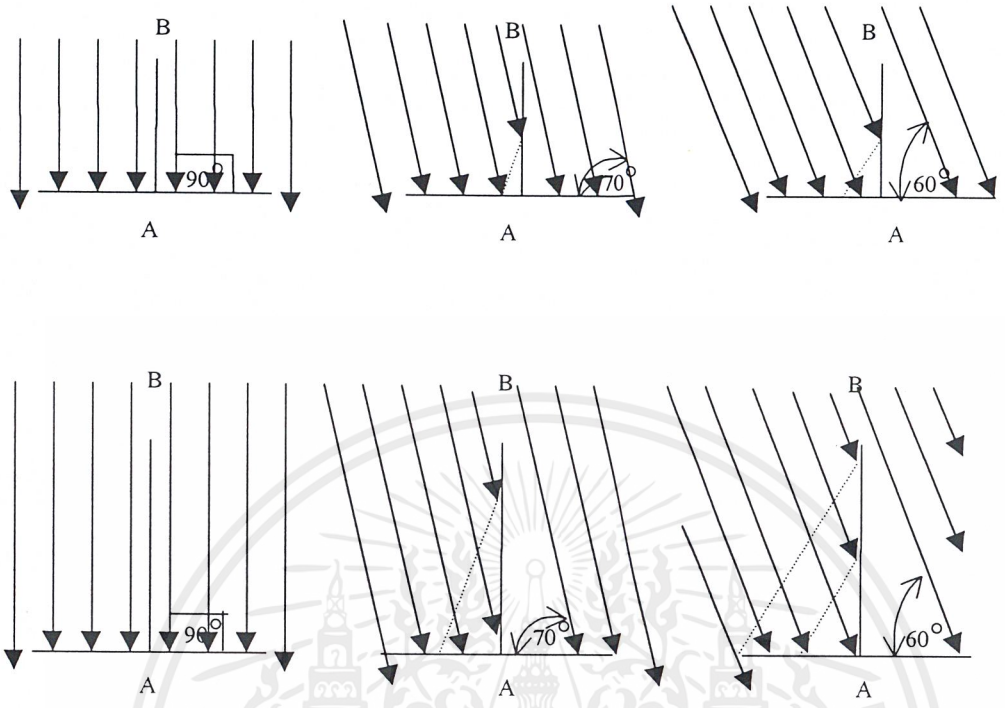
ดังนั้นพอจะกล่าวได้ว่า การใช้ฉากกั้นแสงจะมีผลเกิดขึ้นดังนี้

- ถ้าฉากกั้นขนานกับรังสีจะไม่มีความแตกต่างกัน บนพื้นที่ทั้งสองด้าน
- เมื่อรังสีตกกระทบทำมุมเอียงมากขึ้น ความเข้มของแสงหลังฉากกั้นก็ยิ่งจะน้อยลง
- ในทำนองเดียวกันถ้าเพิ่มความสูงของฉากกั้นความเข้มของแสงหลังฉากกั้น ก็จะยิ่งน้อยลง
- จะมีรังสีส่วนหนึ่งสะท้อนจากฉากกั้น ทำให้พื้นที่ที่ได้รับแสงสะท้อนจะได้รับความเข้มแสงมากกว่าพื้นที่ที่อยู่หลังฉากกั้น

จากประโยชน์ของการมีฉากกั้นดังกล่าวแล้วข้างต้นจึงนำมาออกแบบใช้ประโยชน์ร่วมกับชุดเซ็นเซอร์โดยให้ฉากกั้นสูงกว่าด้านกว้างมาก ๆ เพื่อที่ว่ารังสีตกกระทบทำมุมเพียงเล็กน้อย ความแตกต่างของความเข้มของแสงบนพื้นที่ทั้งด้านหน้า และด้านหลังของฉากกั้น สามารถตรวจจับได้โดยแผงเซลล์แสงอาทิตย์

เนื่องจากว่าชุดเซ็นเซอร์ จะเคลื่อนที่เป็นลักษณะโค้งทั้งในแนวตั้ง และแนวนอน บนแกน 2 แกน การที่ฉากกั้นสูงมาก ๆ จะเป็นอุปสรรคต่อการเคลื่อนที่ในแนวตั้ง เพราะว่าการที่วัตถุที่มีน้ำหนักยื่นห่างออกไปจากจุดหมุนมากขึ้นนั้น จะเป็นการเพิ่มแรงบิด หรือทอร์ค (Torque)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

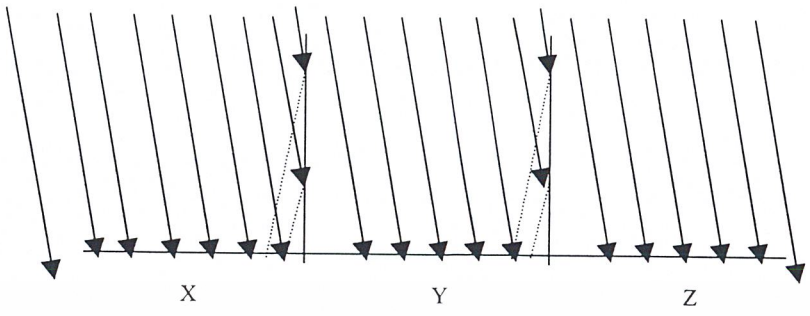


รูปที่ 3.3 ความแตกต่างที่มุมตกกระทบต่าง ๆ และความสูงของฉากกั้นที่ต่าง ๆ กัน

การที่มีข้อจำกัดทางด้านกลศาสตร์ (Mechanics) ทำให้ฉากกั้นมีความยาวจำกัดจึงหันมาพิจารณาถึงความสามารถสะท้อนแสงของผิวฉากกั้น เพราะว่ามีรังสีส่วนหนึ่งสะท้อนกลับลงมาบนพื้นด้านหน้าฉากกั้น จากคุณสมบัตินี้ จึงนำมาใช้ประโยชน์ได้โดยการเพิ่มชุดเซ็นเซอร์ ในแต่ละแนว ดังรูปที่ 3.4 ซึ่งจะเห็นได้ว่าพื้นที่บริเวณ X จะมีความเข้มสูงสุดจากรังสีตกกระทบโดยตรงร่วมกับแสงสะท้อน ส่วนพื้นที่บริเวณ Y จะได้รับรังสีตกกระทบโดยตรงน้อยกว่า แต่ได้รับรังสีสะท้อนเท่าๆ กับบริเวณ X แต่พื้นที่บริเวณ Z นั้น จะได้รับเฉพาะแสงตกกระทบโดยตรงที่ เท่าๆ กับด้าน Y เท่านั้น

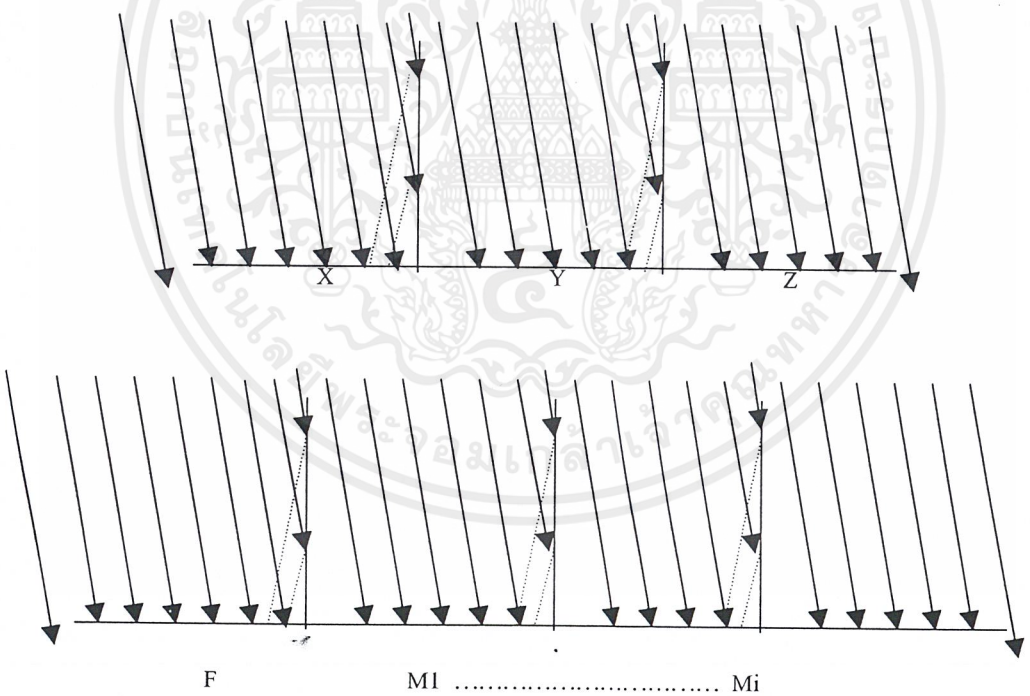
จะเห็นว่าการแบ่งพื้นที่รับแสงมี เซ็กเมนท์ (Segment) มากขึ้นเป็น 3 เซ็กเมนท์ ก็ สามารถช่วยให้พิจารณาการตกกระทบของรังสี ได้ละเอียดขึ้นด้วย แต่การเพิ่มให้มีเซ็กเมนท์มาก ๆ เป็น 4 หรือ 5 เซ็กเมนท์ นั้น ก็ไม่ได้หมายความว่าจะเพิ่มความละเอียดในการพิจารณาได้เพราะว่าเซ็กเมนท์ที่ไม่ได้อยู่ริม จะได้รับรังสีในลักษณะเหมือนกันหมดดังรูปที่ 3.5 ด้าน F จะได้รับรังสีสะท้อนเช่นเดียวกับพื้นที่บริเวณ X ด้าน B จะได้รับรังสีตกกระทบอย่างเดียว ทำนองเดียวกันกับบริเวณ Z และพื้นที่ตรงกลางบริเวณ M1 , M2 , ... , Mi จะได้รับรังสีในทำนองเดียวกันกับบริเวณ Y เหมือนกันหมด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.4 แสดงลักษณะของรังสีโดยมีฉากกั้น 2 แผ่น

การออกแบบของการบังเงา 2 มิติ เราจะไม่แบ่งพื้นที่รับรังสีมากเกินกว่า 3 เซ็กเมนท์ เพราะว่าเป็นการสิ้นเปลือง



รูปที 3.5 ลักษณะรังสีเปรียบเทียบกับจำนวนแผ่นกั้น

ดวงอาทิตย์จะเคลื่อนจากทิศตะวันออกไปยังทิศตะวันตกทุกวัน แต่ไม่ได้ขึ้นและตกลงที่ตำแหน่งเดิมทุกวันจะมีการคลาดเคลื่อนไปในแต่ละวัน ฉะนั้นการตรวจหาตำแหน่งของดวงอาทิตย์จึงเป็น 3 มิติดังนั้นจึงเพิ่มฉากบังแสงขึ้นอีกแนวหนึ่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

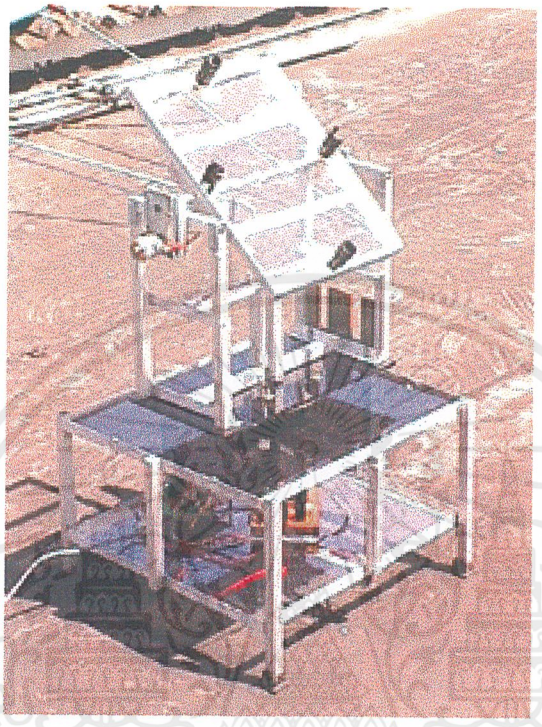
จากการกล่าวมาทั้งหมดข้างต้น เป็นการนำประโยชน์ของการใช้ฉากกั้นมาออกแบบร่วมกับชุดเซ็นเซอร์ นอกจากนั้นแล้วการออกแบบจะต้องพิจารณาปัจจัยทางด้านอื่น ๆ อีก เช่น ด้านกลศาสตร์ตลอดจนถึงด้านสถาปัตยกรรม และอื่น ๆ อีกด้วยจากการบังเงาที่ได้กล่าวมาข้างต้นเราจะนำมาประยุกต์ใช้ในภาคตรวจจับและเปรียบเทียบทิศทาง ซึ่งในรูปที่ 3.2 เมื่อแผงเซลล์แสงอาทิตย์ทำมุมตั้งฉากกับแนวทิศทางของดวงอาทิตย์ ในตำแหน่งที่ 1 อุปกรณ์ไวแสง A และ B จะได้รับความเข้มของแสงเท่ากัน แต่ถ้ามีความคลาดเคลื่อนไปจากแนวตั้งฉากในตำแหน่งที่ 2 อุปกรณ์ไวแสง A และ B ได้รับความเข้มของแสงไม่เท่ากัน (อุปกรณ์ไวแสง B ได้รับความเข้มแสงน้อยกว่า) ซึ่งลักษณะเช่นนี้จะเป็นการชี้บอกให้ทราบว่า ดวงอาทิตย์อยู่เยื้องไปทางทิศใดจากแนวตั้งฉากของแผงเซลล์แสงอาทิตย์

2.3.2 <u>ภาคควบคุม</u> ทำหน้าที่รับผลที่ได้จากการเปรียบเทียบแสงจากแผงเซลล์แสงอาทิตย์ทั้งสอง เพื่อที่จะใช้ในการควบคุมให้แผงเซลล์แสงอาทิตย์ เคลื่อนที่ไปยังตำแหน่งที่ตั้งฉากกับทิศทางของดวงอาทิตย์ซึ่งวงจรควบคุมนี้จะเป็นวงจรอิเล็กทรอนิกส์ ตัวไมโครคอนโทรลเลอร์ที่ใช้ในการควบคุมนี้เราต้องเขียนโปรแกรมให้มันซึ่งตัวโปรแกรมได้แสดงไว้ในภาคผนวกแล้ว

2.3.3 <u>ภาคต้นกำลัง</u> ทำหน้าที่รับค่าจากภาคควบคุมเพื่อหมุนมอเตอร์ไปตามเข็มนาฬิกา หรือทวนเข็มนาฬิกา ภาคต้นกำลังนี้จะมี 2 ส่วน คือ ส่วนที่หมุนในแนวระนาบ กับส่วนที่หมุนในแนวตะวันออก-ตะวันตก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4 โครงสร้างและการออกแบบทางกล

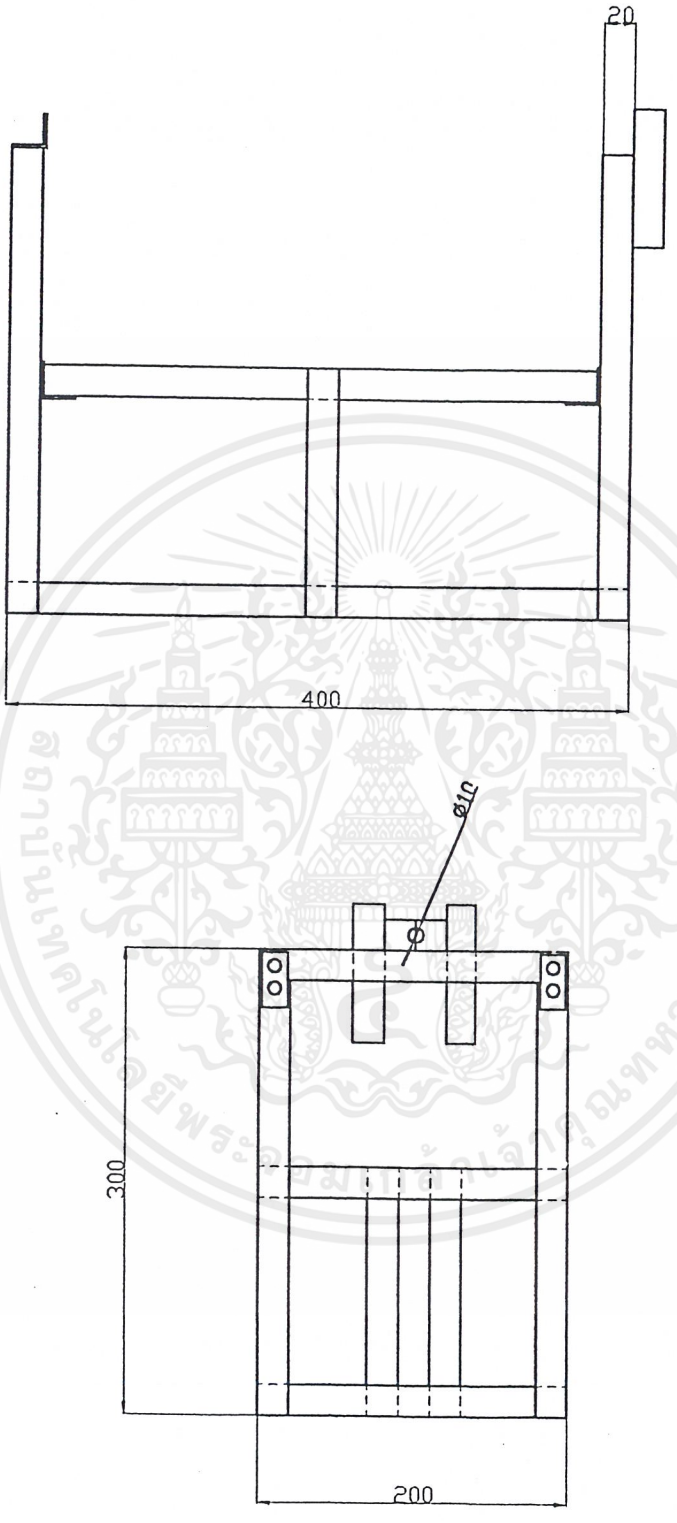


รูปที่ 3.6 แสดงโครงสร้างทางกลของโครงงาน

โครงสร้างทางกลแบ่งเป็น 2 ส่วน คือ

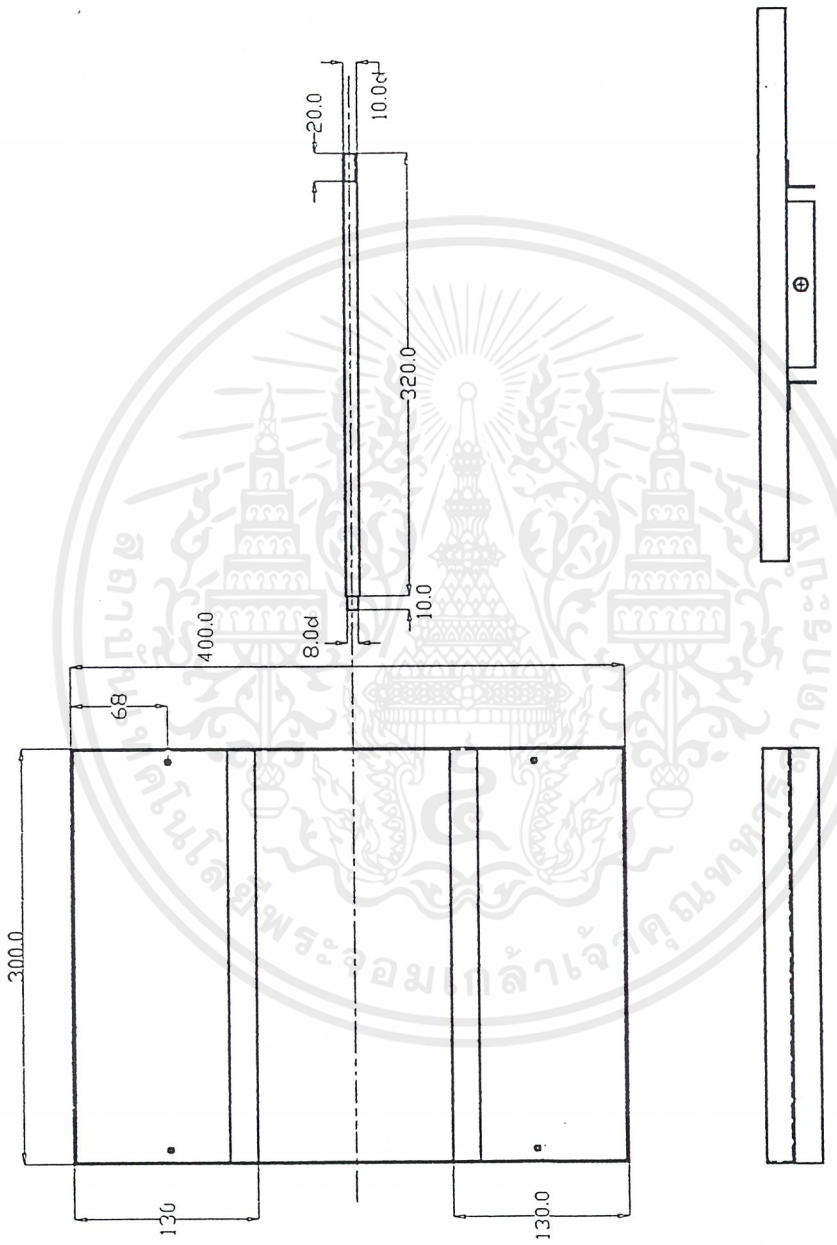
- โครงสร้างส่วนควบคุมการเคลื่อนที่ของแผงเซลล์แสงอาทิตย์ตามแนวตะวันออก-ตะวันตก ดังรูปที่ 3.7
- โครงสร้างส่วนควบคุมการเคลื่อนที่ของแผงเซลล์แสงอาทิตย์ตามแนวระนาบ ดังรูปที่ 3.8

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



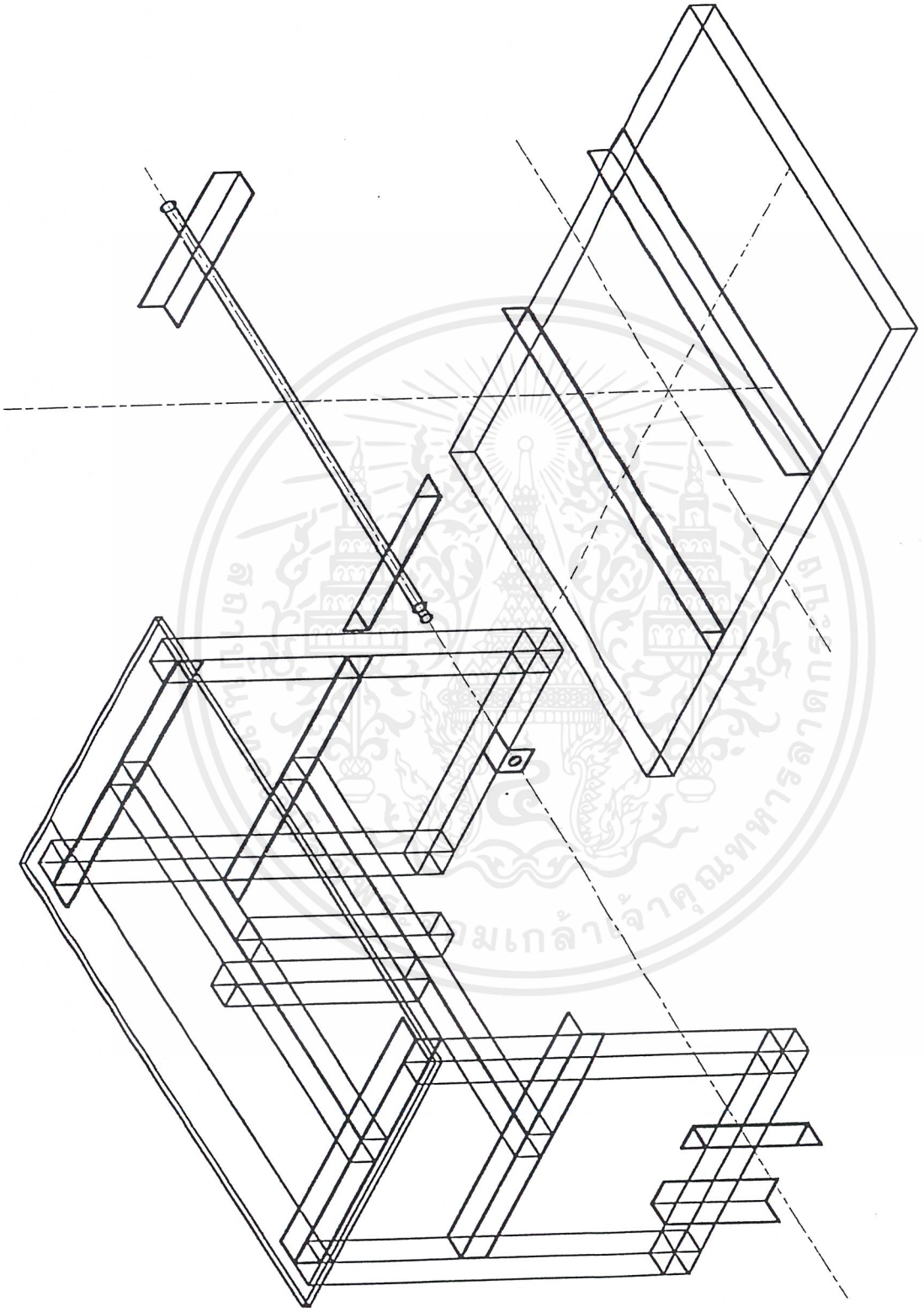
รูปที่ 3.7 โครงสร้างส่วนควบคุมการเคลื่อนที่แนวตะวันออก-ตะวันตก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.7 โครงสร้างส่วนควบคุมการเคลื่อนที่แนวตะวันออก-ตะวันตก (ต่อ)

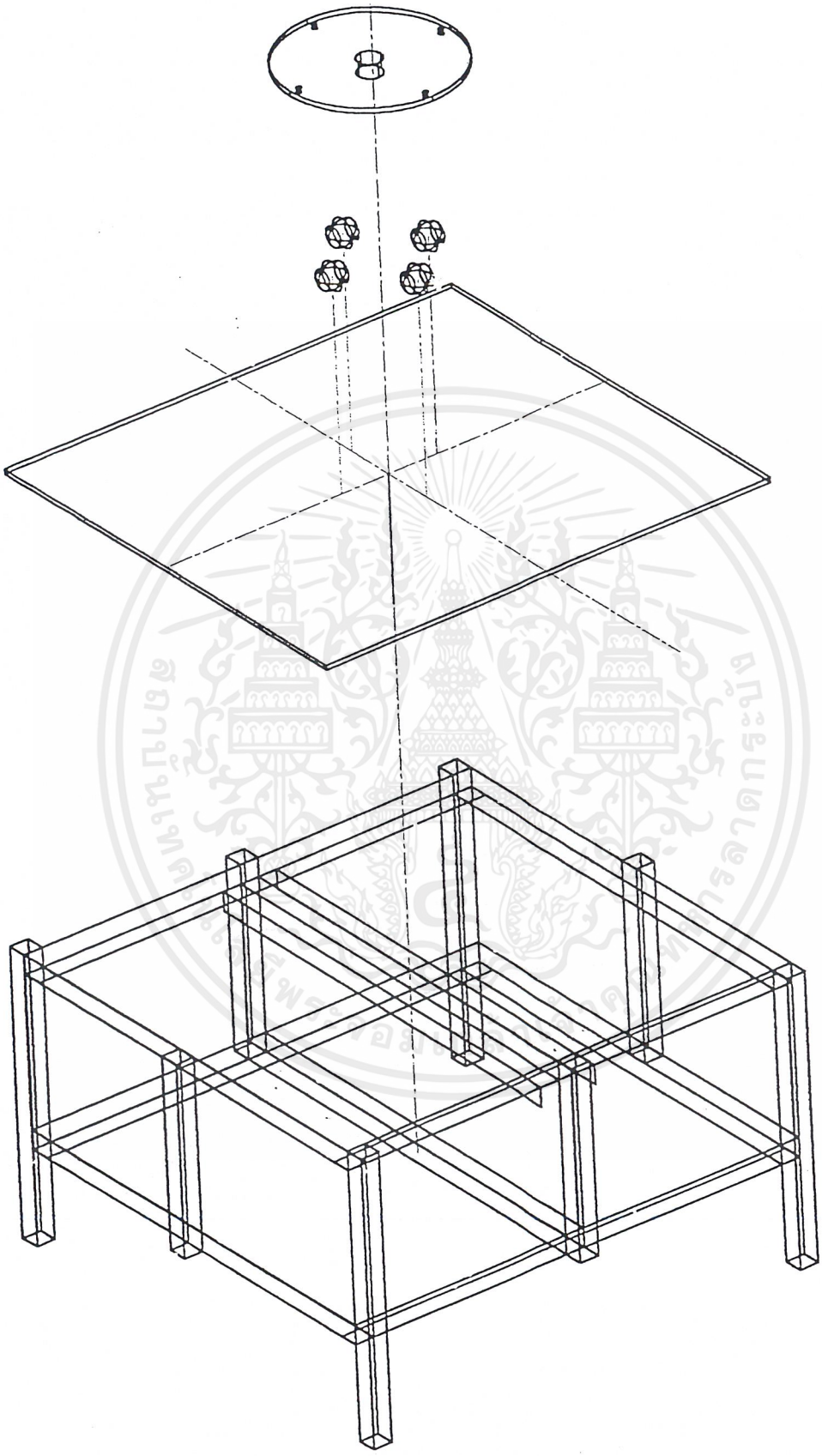
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.7 โครงสร้างส่วนควบคุมการเคลื่อนที่แนวตะวันออก-ตะวันตก (ต่อ)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

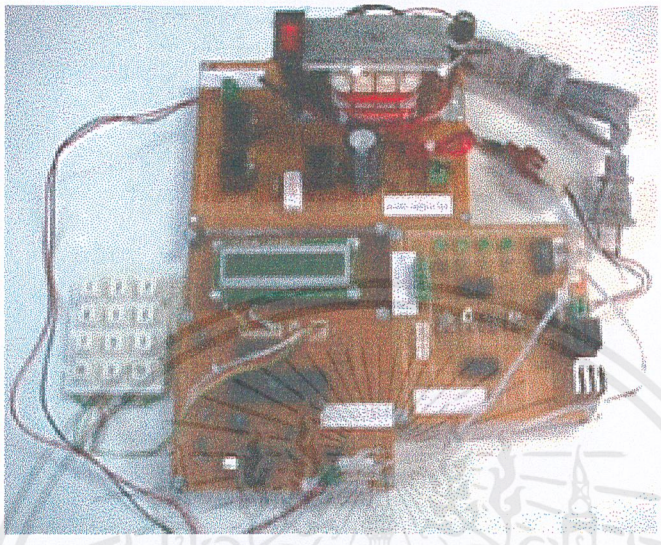


รูปที่ 3.8 โครงสร้างส่วนควบคุมการเคลื่อนที่แนวระนาบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

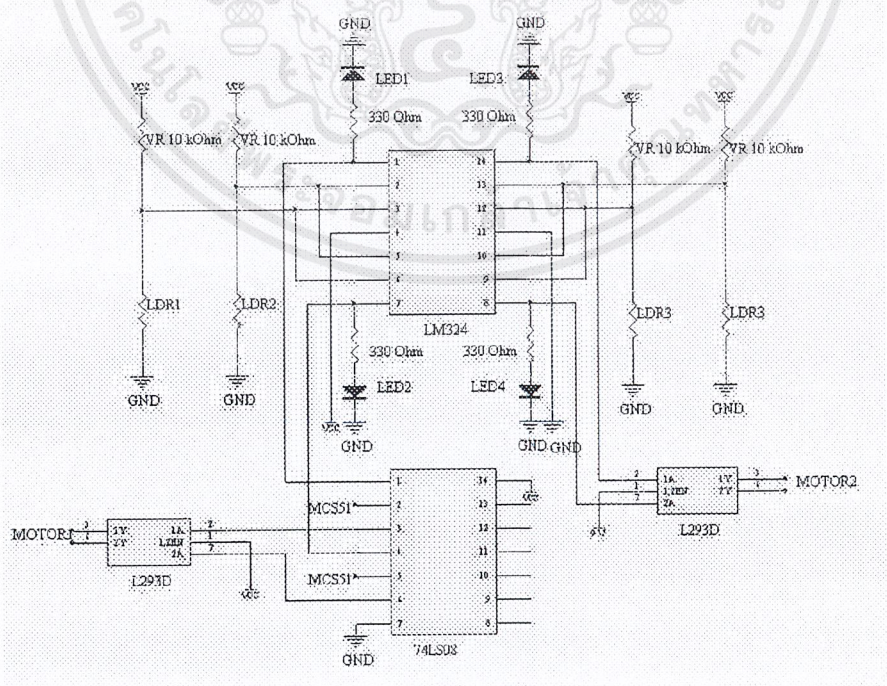
42331

3.5 โครงสร้างและการออกแบบส่วนอิเล็กทรอนิกส์



รูปที่ 3.9 วงจรทั้งหมดในโครงงานนี้

LDR (Light Decreasing Resistance) เป็นอุปกรณ์ที่เปลี่ยนค่าความต้านทานตามความเข้มแสงที่ตกกระทบ



รูปที่ 3.10 วงจรภาคตรวจจับ เปรียบเทียบ และภาคต้นกำลัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ค่า R1, R2, R3 และR4 เราจะปรับค่าความต้านทานให้เท่ากับ LDR1, LDR2, LDR3 และ LDR4 ตามลำดับในกรณีที่ LDR ได้รับความเข้มแสงเต็มที่ (ไม่มีเงามาบัง LDR) เพื่อให้ค่าโวลท์เริ่มต้น V1, V2, V3 และV4 มีค่าเท่ากับ 2.5 โวลท์

<u>หลักการทำงานของวงจรภาคตรวจจับ เปรียบเทียบ และภาคต้นกำลัง</u>

จากรูปที่ 3.10 การเคลื่อนที่ในแนวตะวันออก-ตะวันตก (ส่วนบน) เมื่อมีแสงอาทิตย์ผ่านเข้าตัว LDR 1 และ LDR 2 จะเกิดแรงดัน V1 และ V2 คร่อมตัว LDR1 และ LDR2 ตามลำดับ นำมาเปรียบเทียบที่วงจรเปรียบเทียบ

โดยเมื่อ V1 มากกว่า V2 เอาท์พุต ที่ A1 จะเป็นค่าบวกซึ่งมีค่ามากกว่า 5 โวลท์ แต่เนื่องจากเราป้อนไฟเลี้ยงให้กับออปแอมป์ 5 โวลท์ กับ 0 โวลท์ ดังนั้นค่าเอาท์พุตของออปแอมป์ A1 จะมีค่าเท่ากับ 5 โวลท์ หลอดDIODE1ไฟจะติด ส่วนเอาท์พุตที่ออปแอมป์ A2 จะเป็นค่าลบแต่ไฟเลี้ยงออปแอมป์มีค่าต่ำสุดเท่ากับ 0 โวลท์ ดังนั้นค่าเอาท์พุตของออปแอมป์ A2 จะมีค่าเท่ากับ 0 โวลท์ หลอด DIODE2 ไฟก็จะไม่ติด

เมื่อ V1 น้อยกว่า V2 เอาท์พุตที่ออปแอมป์ A1 จะมีค่าเป็น 0 โวลท์ (เพราะมีค่าติดลบ) หลอดDIODE1ไฟก็จะไม่ติด ส่วนเอาท์พุต ที่ออปแอมป์ A2 จะมีค่าเท่ากับ 5 โวลท์ (เพราะมีค่าเป็นบวก) หลอด DIODE2 ไฟจะติด

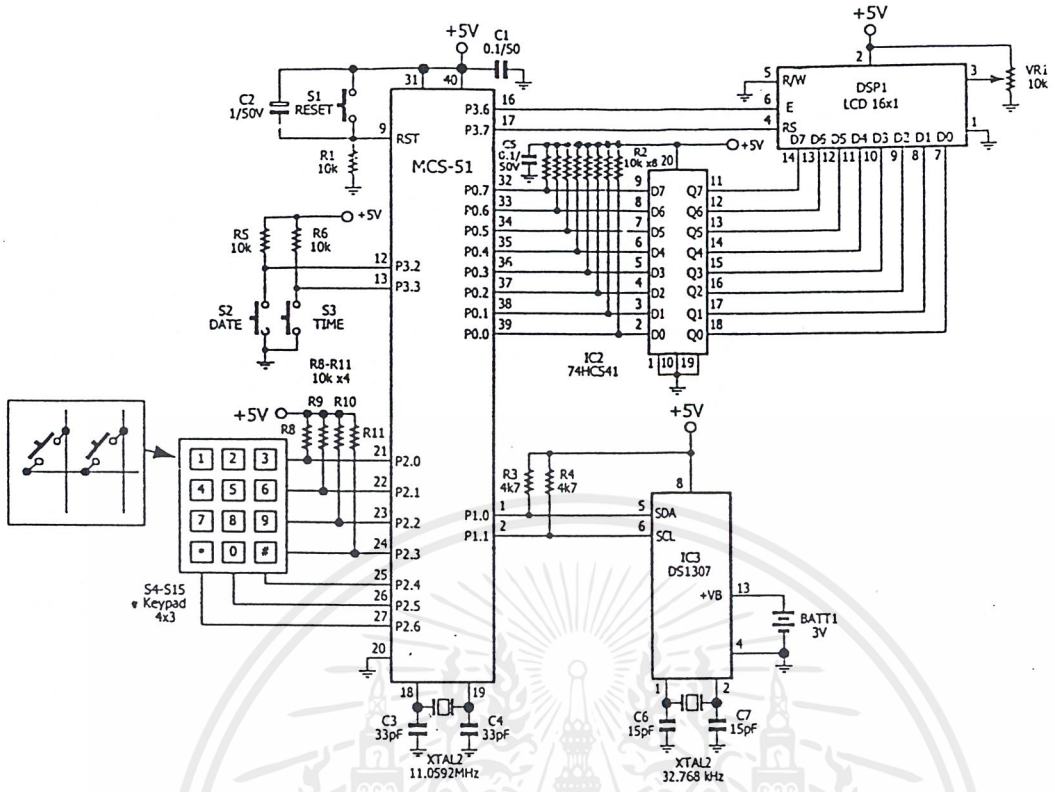
ค่าเอาท์พุตของออปแอมป์ A1 และ A2 จะถูกนำไปป้อนเป็นอินพุต ให้กับ ไอซี AND GATE เบอร์ 74LS08 ซึ่งขาอินพุทอีก 2 ขาจะได้รับสัญญาณมาจากไมโครคอนโทรลเลอร์ไอซีตัวนี้จะใช้ควบคุมกรณีให้แผงเซลล์กลับมาอีกด้านเมื่อถึงเวลา 6 โมงเย็น และเอาท์พุทของไอซีตัวนี้จะต่อเข้ากับ ไอซีDRIVE เบอร์ L293D เพื่อใช้ควบคุมการทำงานของมอเตอร์

การเคลื่อนที่ในแนวระนาบ (ส่วนล่าง) จะมีการทำงานคล้ายกับแนวตะวันออก-ตะวันตก ต่างตรงที่ไม่ต้องต่อกับ ไอซีAND GATE เพราะส่วนล่างนี้เราจะให้มันคงที่เมื่อถึงเวลา 6 โมงเย็น

<u>หลักการทำงานของวงจรภาคควบคุม</u>

จากรูปที่ 3.11 วงจรนี้จะแสดงค่าวันและเวลาที่จอLCD โดยค่าวันและเวลานี้ จะได้รับมาจาก ไมโครคอนโทรลเลอร์ ซึ่งมีการอ้างอิงเวลาจริงมาจาก ไอซีเรียลไทม์คล็อก ค่าเวลานี้จะนำไปใช้ในการควบคุมกรณีที่ เมื่อถึงเวลา 6 โมงเย็น และกรณีที่ต้องการให้ติดตามดวงอาทิตย์ทุก 30 นาที โดยไมโครคอนโทรลเลอร์จะส่งค่าสัญญาณไปควบคุมที่ตัว ไอซี AND GATE และไอซี DRIVE ทั้ง 2 ตัว

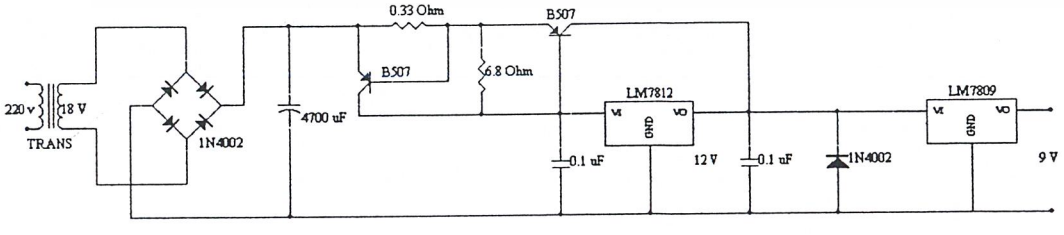
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.11 วงจรภาคควบคุม

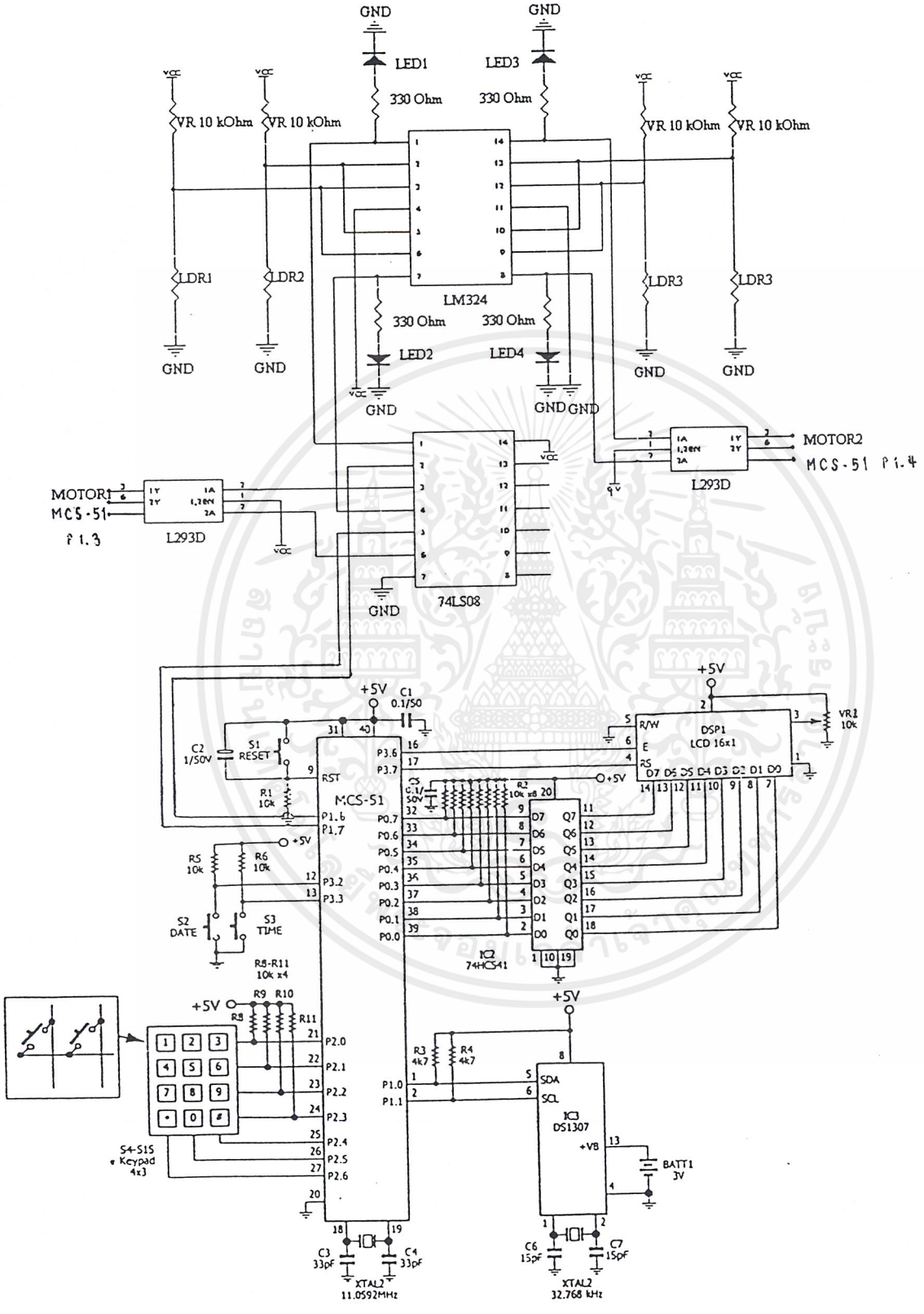
<u>หลักการทำงานวงจรแหล่งจ่ายไฟ</u>

จากรูปที่ 3.12 ไฟกระแสสลับ 18 โวลท์ ที่ได้จากหม้อแปลง (STEP DOWN จาก 220 โวลท์เป็น 18 โวลท์) จะผ่านเข้าวงจรบริดจ์ เพื่อเรียงกระแสให้เป็นไฟกระแสตรง 18 โวลท์ เมื่อผ่านเรกูเลเตอร์ เบอร์ 7812 ก็จะได้แรงดันเอาท์พุท เท่ากับ 12 โวลท์



รูปที่ 3.12 วงจรแหล่งจ่ายไฟ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.13 การต่อวงจรรวมทั้งหมด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.6 รายการอุปกรณ์

วงจรภาคตรวจจับ เปรียบเทียบ และภาคต้นกำลัง

LDR	4	ตัว
VR 10 k	4	ตัว
LED	4	ตัว
R 330 Ω	4	ตัว
OPAMP LM324	1	ตัว (พร้อม SOCKET)
IC AND GATE 74LS08	1	ตัว (พร้อม SOCKET)
IC DRIVE L293D	2	ตัว (พร้อม SOCKET)
REGULATOR 7805	1	ตัว
REGULATOR 7809	1	ตัว
HEATSINK	2	ตัว
TERMINAL 2 ช่อง	4	ตัว
CONNECTOR 8 PIN	1	ตัว
CONNECTOR 2PIN	2	ตัว

วงจรภาคควบคุม

MICROCONTROLLER 89C51	1	ตัว (พร้อม SOCKET)
BUFFER 74HC541	1	ตัว (พร้อม SOCKET)
RTC DS1307	1	ตัว (พร้อม SOCKET)
LCD 16x1	1	ตัว
KEYPAD	1	ตัว
R 10 kΩ	16	ตัว
R 4.7 kΩ	2	ตัว
C 1/50 V	1	ตัว
C 0.1/50 V	2	ตัว
C 33 pF	2	ตัว
C 15 pF	2	ตัว
SWITCH RESET	3	ตัว

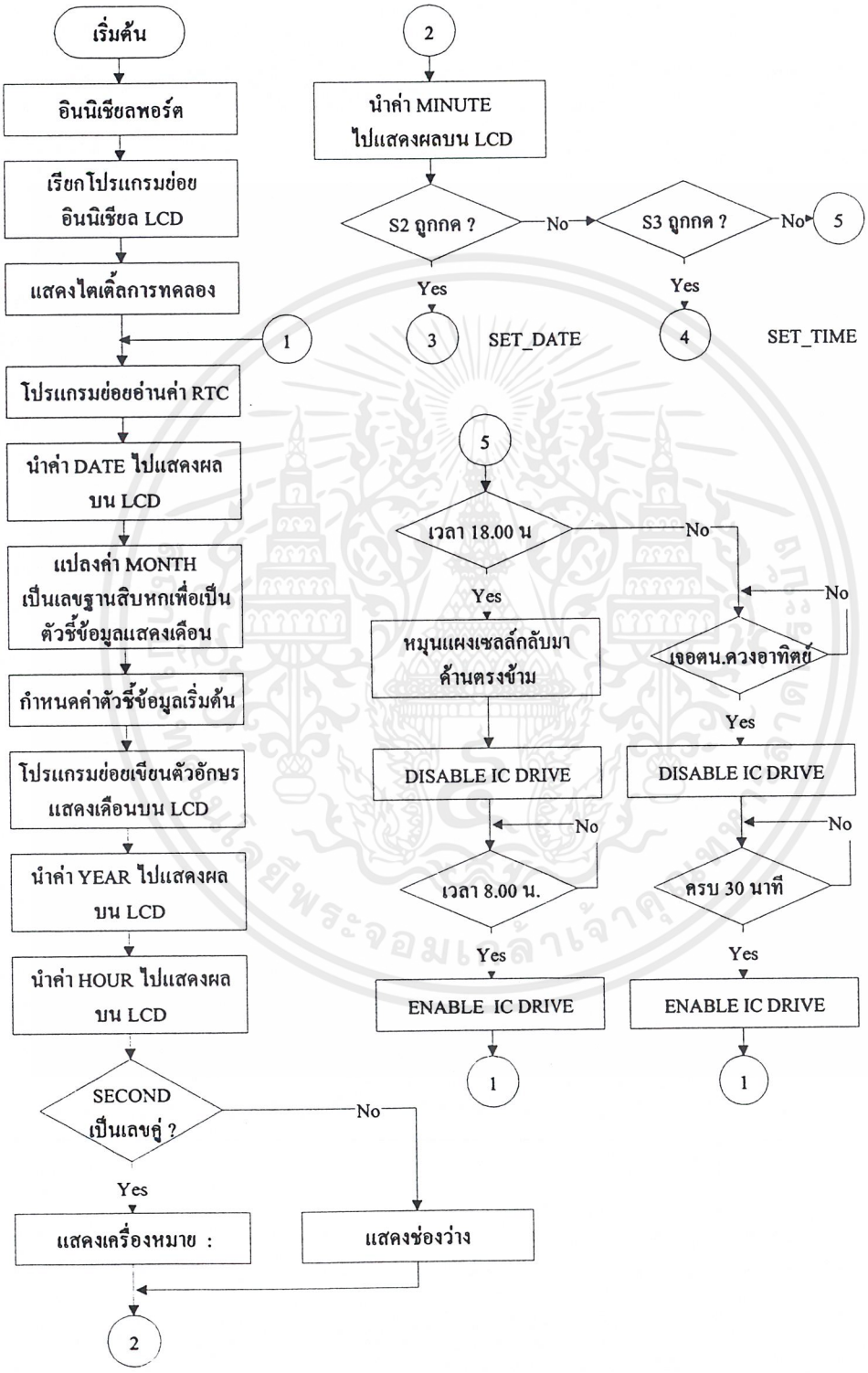
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

CRYSTAL 11.0592 MHz	1	ตัว
CRYSTAL 32.768 kHz	1	ตัว
แบตเตอรี่ 3 V พร้อมลังถ่าน	1	ชุด
REGULATOR 7805	1	ตัว
HEATSINK	1	ตัว
TERMINAL 2 ช่อง	1	ตัว
CONNECTOR 8 PIN	1	ตัว
CONNECTOR 4 PIN	1	ตัว
CONNECTOR 2 PIN	2	ตัว

วงจรแหล่งจ่ายไฟ

TRANSISTOR B507 PNP	2	ตัว
DIODE 1N4002	5	ตัว
R 6.8 R 1W	1	ตัว
R 0.33 R 5W	1	ตัว
C 4700 uF 50 V	1	ตัว
C 0.1 uF	2	ตัว
SWITCH 220 V	1	ตัว
หม้อแปลง 220 V เป็น 18 V	1	ตัว
สายไฟ AC พร้อมปลั๊กตัวผู้	1	ชุด
กระปุกฟิวส์	1	ชุด
ฟิวส์ 0.5 แอมป์	1	ตัว
REGULATOR 7812	1	ตัว
HEATSINK	3	ตัว
TERMINAL 2 ช่อง	2	ตัว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.14 โฟลว์ชาร์ตการทำงานของโปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 3.14 โฟลว์ชาร์ตการทำงานของโปรแกรม (ต่อ)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

ผลการทดลอง

ผลการทดลองการวัดค่ามุมของแผงเซลล์แสงอาทิตย์

การทดลองครั้งนี้ เราจะวัดมุมเฉพาะการเคลื่อนที่ของแผงเซลล์แสงอาทิตย์จำลองที่สร้างจากพลาสติกขนาด 30*40 ซม.ให้ติดดวงอาทิตย์ตามในแนวตะวันออก-ตะวันตก โดยเราจะติดครึ่งวงกลมไว้ที่ตัวอุปกรณ์ ดังรูปที่ 4.1

การทดลองนี้ใช้ระยะเวลา 5 วัน โดยเราจะทำการวัดมุมทุกๆ 1 ชั่วโมง ตั้งแต่ 8.00 น.ถึง 18.00 น. เมื่อถึงเวลาที่เราต้องบันทึกผลแล้วอุปกรณ์เกิดการหลงทางอยู่ เราจะรอบันทึกผลไม่เกิน 10 นาที ถ้าเกิน 10 นาทีแล้วอุปกรณ์ยังหลงทางเช่นเดิม เราจะไม่บันทึกค่า จากการทดลองได้ผลการทดลองแสดงได้ดังนี้

ตารางผลการทดลอง

วันที่ 24 มีนาคม 2544

เวลา	8.00	9.00	10.00	11.00	12.00	13.00
มุม (องศา)	45	57	68	78	89(12.07)	100

เวลา	14.00	15.00	16.00	17.00	18.00	
มุม (องศา)	114	124	142	154(17.02)	158	

วันที่ 25 มีนาคม 2544

เวลา	8.00	9.00	10.00	11.00	12.00	13.00
มุม (องศา)	42	54	68	79	92	104

เวลา	14.00	15.00	16.00	17.00	18.00	
มุม (องศา)	116(14.02)	130	143	155	157	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

วันที่ 28 มีนาคม 2544

เวลา	8.00	9.00	10.00	11.00	12.00	13.00
มุม (องศา)	43	55	65	79	94	103

เวลา	14.00	15.00	16.00	17.00	18.00	
มุม (องศา)	113	125	142	-	156	

วันที่ 29 มีนาคม 2544

เวลา	8.00	9.00	10.00	11.00	12.00	13.00
มุม (องศา)	42	53	69(10.05)	77	93	101

เวลา	14.00	15.00	16.00	17.00	18.00	
มุม (องศา)	114	127	142	154	157	

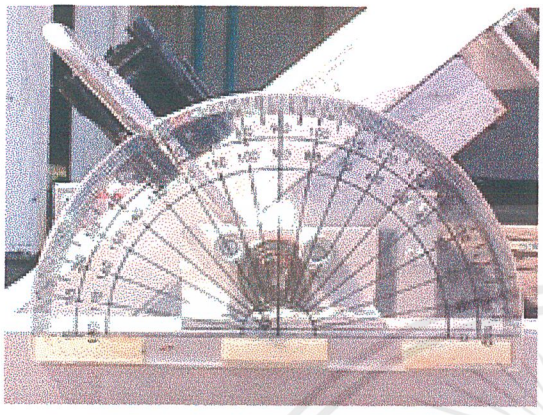
วันที่ 30 มีนาคม 2544

เวลา	8.00	9.00	10.00	11.00	12.00	13.00
มุม (องศา)	41	52	67	78	90	-

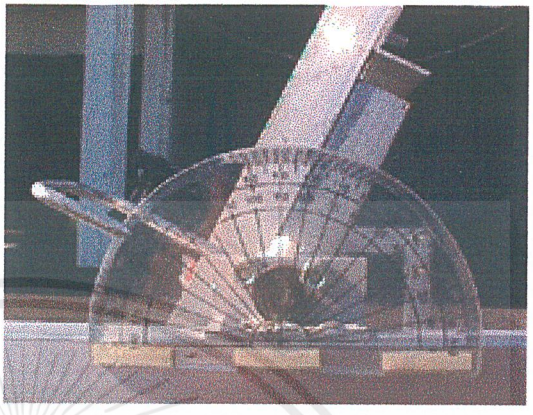
เวลา	14.00	15.00	16.00	17.00	18.00	
มุม (องศา)	113(14.03)	127	141	153	156	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

วันที่ 24 มีนาคม 2544

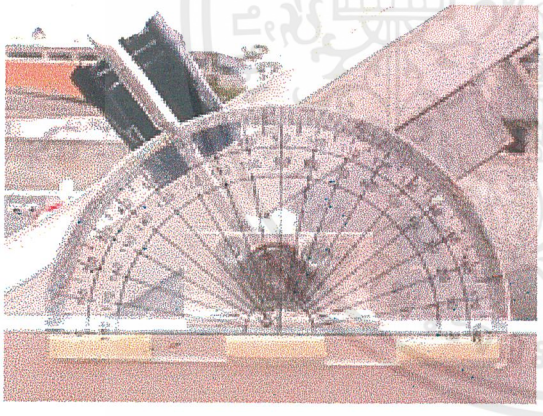


เวลา 15.00 นาฬิกา

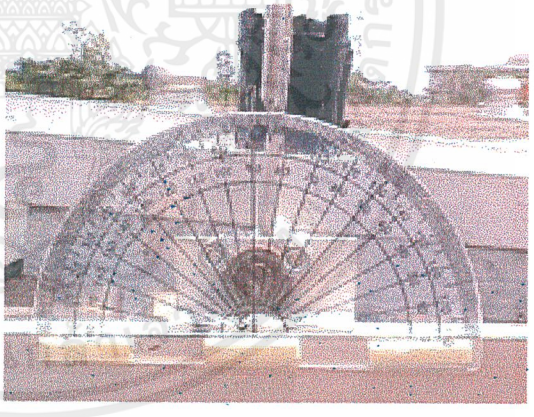


เวลา 16.00 นาฬิกา

วันที่ 28 มีนาคม 2544



เวลา 10.00 นาฬิกา



เวลา 12.00 นาฬิกา

รูปที่ 4.1 แสดงผลที่ได้จากการทดลองของ 2 วัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5
บทสรุปและวิจารณ์

จากผลการทดลองการควบคุมแผงเซลล์แสงอาทิตย์จำลองในแต่ละวัน เราสามารถสรุปได้ว่า ค่ามุมในแต่ละช่วงของแต่ละวันมีค่าใกล้เคียงกัน เป็นการแสดงให้เห็นว่า อุปกรณ์ที่ทดลองมีประสิทธิภาพ ทำงานได้ค่อนข้างดีเป็นไปตามเป้าหมาย แต่ในบางช่วงเวลาที่ไม่ได้มีการบันทึกผลมุม เนื่องจากช่วงนั้นเป็นช่วงที่มีเมฆมาบดบังดวงอาทิตย์ ทำให้อุปกรณ์เกิดการหลงทาง เมื่ออุปกรณ์หลงทางเกิน 10 นาที เราจะไม่สามารถบันทึกค่าได้ อาจจะแก้ไขโดยวิธี ในช่วงที่มีเมฆมาบดบังดวงอาทิตย์ก็ให้อุปกรณ์เคลื่อนที่ตามปฏิทินการเคลื่อนที่ของดวงอาทิตย์แทนโดยต้องทำการโปรแกรมเพิ่มเติม

ปัญหาด้านโครงสร้างทางกล

- ตัว LDR ไม่ได้ยึดติดกับฉากที่บัง ทำให้เวลาเคลื่อนที่ LDR จะสั่น ดังนั้นค่าที่ตรวจจับได้จาก LDR จึงไม่คงที่
- เฟืองทดรอบของชุดมอเตอร์ ที่ฟันเฟืองของแต่ละอันไม่ลงล็อคกันพอดี ทำให้เกิดการสั่นเหมือนกันเวลาที่เคลื่อนที่

จากโครงงานนี้ สามารถนำไปใช้ประยุกต์ได้จริง แต่ต้องทำการปรับปรุงในส่วนของโครงสร้างทางกลให้รับน้ำหนักของแผงเซลล์แสงอาทิตย์ได้จริง ด้านโปรแกรม สามารถนำไปใช้ได้แต่ควรพัฒนาในส่วนที่เกิดการหลงทาง เพื่อให้ได้ประสิทธิภาพดียิ่งขึ้น ส่วนของวงจร เมื่อนำไปใช้กับแผงเซลล์แสงอาทิตย์จริง ก็ต้องมีวงจรชาร์จแบตเตอรี่และแบตเตอรี่ เพื่อใช้เก็บพลังงานไฟฟ้าที่ได้จากแผงเซลล์แสงอาทิตย์พลังงานไฟฟ้าที่ได้นี้ส่วนหนึ่งจะป้อนเป็นไฟเลี้ยงให้กับวงจรทั้งหมดในอุปกรณ์นี้ด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
;*********************************************************************
; Program       : Real Time Clock
; Description   : Real Time Clock , S2 Preset Date , S3 Preset Time
;*********************************************************************
;---------------------------------------------------------------------
; Define Port&Pin Name
;---------------------------------------------------------------------
SDA             BIT     P1.0        ; SDA I2C Bue
SCL             BIT     P1.1        ; SCL I2C Bue
DRIVE_UP        BIT     P1.4        ; Enable Drive UP
DRIVE_DOWN      BIT     P1.5        ; Enable Drive DOWN
DRIVE_IN1       BIT     P1.6        : Drive Input 1
DRIVE_IN2       BIT     P1.7        ; Drive Input 2
KPAD_ROW0       BIT     P2.0        ; Keypad Input Row 0
KPAD_ROW1       BIT     P2.1        ; Keypad Input Row 1
KPAD_ROW2       BIT     P2.2        ; Keypad Input Row 2
KPAD_ROW3       BIT     P2.3        ; Keypad Input Row 3
KPAD_COL2       BIT     P2.4        ; Keypad Output Column 2
KPAD_COL1       BIT     P2.5        ; Keypad Output Column 1
KPAD_COL0       BIT     P2.6        ; Keypad Output Column 0
SET_DATE_SW     BIT     P3.2        ; Interrupt 0 PIN S2
SET_TIME_SW     BIT     P3.3        ; Interrupt 1 PIN S3
LCD_EN          BIT     P3.6        ; LCD Module Enable ( Active High : Level )
LCD_RS          BIT     P3.7        ; LCD Module Register Select
;---------------------------------------------------------------------
; Define User Register
;---------------------------------------------------------------------
FLAG            EQU     02FH        ; User FLAG
;   7   6   5   4   3   2   1   0
;                           I2C_ACK
I2C_ACK         BIT     FLAG.0      ; Define I2C Acknowledge as bit
;---------------------------------------------------------------------
; Define User Register
;---------------------------------------------------------------------
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
LCD_ADDR        EQU     030H        ; For keep LCD Address
LCD_DATA        EQU     031H        ; For keep LCD Data
LCD_PTR         EQU     032H        ; For keep LCD 3 Cha.r Pointer
I2C_ADDR        EQU     033H        ; For keep I2C Address
I2C_DATA        EQU     034H        ; For keep I2C Data
KPAD_DATA       EQU     035H        ; For keep Keypad Data
BUFFER          EQU     036H        ; Buffer
SECONDS         EQU     040H        ; For keep Seconds
MINUTES         EQU     041H        ; For keep Minutes
HOURS           EQU     042H        ; For keep Hours
DAY             EQU     043H        ; For keep Day
DATE            EQU     044H        ; For keep Date
MONTH           EQU     045H        ; For keep Month
YEAR            EQU     046H        ; For keep Year
CONTROL         EQU     047H        ; For keep Control Byte
;-------------------------------------------------------
; Define I2C Slave Address
;-------------------------------------------------------
RTC_ID          EQU     11010000B   ; RTC Slave Address
;-------------------------------------------------------
; Main Program
;-------------------------------------------------------
                ORG     0000H               ; Reset Vector
                MOV     P0,#00000000B       ; Clear Databus
                MOV     P1,#11101111B       ; Clear status all devices
                MOV     P2,#11111111B       ; Clear status keypad and 1-Wire
                MOV     P3,#00011111B       ; Clear status LCD, R-2R, P3.2-3.4
MAIN:           ACALL   INIT_LCD            ; Call LCD Initial subroutine
                MOV     LCD_ADDR,#000H      ; Set Address 00H
                ACALL   SET_ADDR_LCD        ;
                MOV     DPTR,#TITLE_1       ; Index Pointer ROM to Show LCD
                ACALL   WRLINE_LCD          ; 00H-07H, 40H-47H ( Increase automatic )
                ACALL   DELAY_1s            ; Delay
                ACALL   DELAY_1s
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
LOOP:               ACALL  RTC_RD                ; Read RTC
                    MOV    LCD_ADDR,#000H        ; Set Address 00H
                    ACALL  SET_ADDR_LCD          ;
                    MOV    LCD_DATA,#' '         ; Write space to LCD
                    ACALL  WRCHAR_LCD            ;
                    MOV    LCD_DATA,DATE         ; Write Date to LCD
                    ACALL  BCD2LCD               ; Write from RTC BCD Data
                    MOV    LCD_DATA,#' '         ; Write space to LCD
                    ACALL  WRCHAR_LCD            ;
                    MOV    A,MONTH               ; Convert Month from BCD to HEX
                    CJNE   A,#010H,WR_CHK_MONTH_1      ;
                    MOV    A,#00AH               ; October => 0AH
                    AJMP   WRITE_MONTH_NX              ;
WR_CHK_MONTH_1:     CJNE   A,#011H,WR_CHK_MONTH_2      ;
                    MOV    A,#00BH               ; November => 0BH
                    AJMP   WRITE_MONTH_NX        ;
WR_CHK_MONTH_2:     CJNE   A,#012H,WRITE_MONTH_NX      ;
                    MOV    A,#00CH               ; December => 0CH
WRITE_MONTH_NX:     MOV    LCD_PTR,A             ; Set 3 Char. Pointer
                    MOV    DPTR,#MONTH_JAN       ; Set Start Pointer
                    ACALL  WR3CHAR_LCD           ; Write 3 Char. ( Month ) to LCD
                    MOV    LCD_DATA,#' '         ; Write space to LCD
                    ACALL  WRCHAR_LCD            ;
                    MOV    LCD_ADDR,#040H        ; Set Address 40H
                    ACALL  SET_ADDR_LCD          ;
                    MOV    LCD_DATA,YEAR         ; Write Year to LCD
                    ACALL  BCD2LCD               ; Write from RTC BCD Data
                    MOV    LCD_DATA,#' '         ; Write space to LCD
                    ACALL  WRCHAR_LCD            ;
WRITE_TIME:         MOV    A,HOURS               ; Get Hour
                    ANL    A,#00110000B          ; Get x10 Digit
                    JZ     WRITE_TIME_HN         ; Check x10 Digit = 0 ?
                    SWAP   A                     ; If not => Write to LCD
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                ADD     A,#030H                 ; Convert to ASCII
                AJMP    WRITE_TIME_HH           ;
WRITE_TIME_HN:  MOV     A,#' '                  ; x10 = 0 then Write space
WRITE_TIME_HH:  MOV     LCD_DATA,A              ; Write to LCD
                ACALL   WRCHAR_LCD              ;
                MOV     A,HOURS                 ; Write x1 Digit to LCD
                ANL     A,#00001111B            ;
                ADD     A,#030H                 ; Convert to ASCII
                MOV     LCD_DATA,A              ;
                ACALL   WRCHAR_LCD              ;
                MOV     A,SECONDS               ; Check Second = Odd Number ?
                ANL     A,#001H                 ;
                JNZ     WRITE_SPACE             ; Even => Write space
                MOV     LCD_DATA,#':'           ; Odd => Write ':'
                ACALL   WRCHAR_LCD              ;
                AJMP    WRITE_MINUTES           ;
WRITE_SPACE:    MOV     LCD_DATA,#' '           ;
                ACALL   WRCHAR_LCD              ;
WRITE_MINUTES:  MOV     LCD_DATA,MINUTES        ; Write Minute to LCD
                ACALL   BCD2LCD                 ; Write from RTC BCD Data
                JB      SET_DATE_SW,SET_TIME_1  ; Check S2 Pressed? => Set Date
                JMP     SET_DATE
SET_TIME_1:     JB      SET_TIME_SW,CHECK_HOURS ; Check S3 Pressed? => Set Time
                JMP     SET_TIME
CHECK_HOURS:    MOV     A,HOURS                 ;
                CJNE    A,#00011000B,TIME_RECORD ; Check 18.00 o'clock ?
                AJMP    RESET_CELL              ;
TIME_RECORD:    SETB    DRIVE_UP                ; Start Drive Up
                SETB    DRIVE_DOWN              ; Start Drive Down
                SETB    DRIVE_IN1               ;
                SETB    DRIVE_IN2               ;
                AJMP    LOOP                    ; Jump to loop
RESET_CELL:     CLR     DRIVE_DOWN              ; Stop Drive Down
                SETB    DRIVE_UP                ; Start Drive Up
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                CLR     DRIVE_IN1               ;
                CLR     DRIVE_IN2               ;
                ACALL   DELAY_5sec              ; Delay 5 Seconds
CHECK_EIGHT_AM: ACALL   RTC_RD                  ;
                MOV     A,HOURS                 ;
                CJNE    A,#08H,CHECK_EIGHT_AM   ;
                SETB    DRIVE_UP                ; Start Drive Up
                SETB    DRIVE_DOWN              ; Start Drive Down
                AJMP    LOOP                    ; Jump to loop
SET_DATE:       MOV     LCD_ADDR,#000H          ; Set Address 00H
                ACALL   SET_ADDR_LCD            ;
                MOV     DPTR,#SCR_SET_DATE      ; Index Pointer ROM to Show LCD
                ACALL   WRLINE_LCD              ; 00H-07H, 40H-47H ( Increase automatic )
                MOV     LCD_ADDR,#040H          ; Set Address 40H
                ACALL   SET_ADDR_LCD            ;
                ACALL   LCD_BLINK               ; Blink LCD
                ACALL   WAIT_KEYPRESSED         ; Wait until Keypad pressed
                MOV     BUFFER,KPAD_DATA        ; Copy in BUFFER
                MOV     A,KPAD_DATA             ; Get Keypad Data
                ADD     A,#030H                 ; Convert to ASCII
                MOV     LCD_DATA,A              ; Write to LCD
                ACALL   WRCHAR_LCD              ;
                ACALL   LCD_BLINK               ; Blink LCD
                ACALL   WAIT_KEY                ; Wait until Keypad depressed
                ACALL   WAIT_KEYPRESSED         ; Wait until Keypad pressed
                MOV     BUFFER+1,KPAD_DATA      ; Copy in BUFFER+1
                MOV     A,KPAD_DATA             ; Get Keypad Data
                ADD     A,#030H                 ; Convert to ASCII
                MOV     LCD_DATA,A              ; Write to LCD
                ACALL   WRCHAR_LCD              ;
                ACALL   WAIT_KEY                ; Wait until Keypad depressed
                MOV     A,BUFFER                ; Restore BUFFER
                ANL     A,#00FH                 ; Get lower 4 bit
                SWAP    A                       ; Swap nibble
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
MOV     B,A                     ; Move to B
MOV     A,BUFFER+1              ; Restore BUFFER+1
ANL     A,#00FH                 ; Get lower 4 bit
ADD     A,B                     ; Add BUFFER to make RTC BCD
MOV     DATE,A                  ; Keep Date Data
MOV     LCD_ADDR,#043H          ; Set Address 43H
ACALL   SET_ADDR_LCD            ;
ACALL   LCD_BLINK               ; Blink LCD
ACALL   WAIT_KEYPRESSED         ; Wait until Keypad pressed
MOV     BUFFER,KPAD_DATA        ; Copy in BUFFER
MOV     A,KPAD_DATA             ; Get Keypad Data
ADD     A,#030H                 ; Convert to ASCII
MOV     LCD_DATA,A              ; Write to LCD
ACALL   WRCHAR_LCD              ;
ACALL   LCD_BLINK               ; Blink LCD
ACALL   WAIT_KEY                ; Wait until Keypad depressed
ACALL   WAIT_KEYPRESSED         ; Wait until Keypad pressed
MOV     BUFFER+1,KPAD_DATA      ; Copy in BUFFER+1
MOV     A,KPAD_DATA             ; Get Keypad Data
ADD     A,#030H                 ; Convert to ASCII
MOV     LCD_DATA,A              ; Write to LCD
ACALL   WRCHAR_LCD              ;
ACALL   WAIT_KEY                ; Wait until Keypad depressed
MOV     A,BUFFER                ; Restore BUFFER
ANL     A,#00FH                 ; Get lower 4 bit
SWAP    A                       ; Swap nibble
MOV     B,A                     ; Move to B
MOV     A,BUFFER+1              ; Restore BUFFER+1
ANL     A,#00FH                 ; Get lower 4 bit
ADD     A,B                     ; Add BUFFER to make RTC BCD
MOV     MONTH,A                 ; Keep Month Data
MOV     LCD_ADDR,#046H          ; Set Address 46H
ACALL   SET_ADDR_LCD            ;
ACALL   LCD_BLINK               ; Blink LCD
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                ACALL   WAIT_KEYPRESSED     ; Wait until Keypad pressed
                MOV     BUFFER,KPAD_DATA    ; Copy in BUFFER
                MOV     A,KPAD_DATA         ; Get Keypad Data
                ADD     A,#030H             ; Convert to ASCII
                MOV     LCD_DATA,A          ; Write to LCD
                ACALL   WRCHAR_LCD          ;
                ACALL   LCD_BLINK           ; Blink LCD
                ACALL   WAIT_KEY            ; Wait until Keypad depressed
                ACALL   WAIT_KEYPRESSED     ; Wait until Keypad pressed
                MOV     BUFFER+1,KPAD_DATA  ; Copy in BUFFER+1
                MOV     A,KPAD_DATA         ; Get Keypad Data
                ADD     A,#030H             ; Convert to ASCII
                MOV     LCD_DATA,A          ; Write to LCD
                ACALL   WRCHAR_LCD          ;
                ACALL   WAIT_KEY            ; Wait until Keypad depressed
                MOV     A,BUFFER            ; Restore BUFFER
                ANL     A,#00FH             ; Get lower 4 bit
                SWAP    A                   ; Swap nibble
                MOV     B,A                 ; Move to B
                MOV     A,BUFFER+1          ; Restore BUFFER+1
                ANL     A,#00FH             ; Get lower 4 bit
                ADD     A,B                 ; Add BUFFER to make RTC BCD
                MOV     YEAR,A              ; Keep Year Data
                ACALL   RTC_WR              ; Write RTC
                AJMP    LOOP                ; Jump to loop
SET_TIME:       MOV     LCD_ADDR,#000H      ; Set Address 00H
                ACALL   SET_ADDR_LCD        ;
                MOV     DPTR,#SCR_SET_TIME  ; Index Pointer ROM to Show LCD
                ACALL   WRLINE_LCD          ; 00H-07H, 40H-47H ( Increase automatic )
                MOV     LCD_ADDR,#040H      ; Set Address 40H
                ACALL   SET_ADDR_LCD        ;
                ACALL   LCD_BLINK           ; Blink LCD
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
ACALL  WAIT_KEYPRESSED      ; Wait until Keypad pressed
MOV    BUFFER,KPAD_DATA     ; Copy in BUFFER
MOV    A,KPAD_DATA          ; Get Keypad Data
ADD    A,#030H              ; Convert to ASCII
MOV    LCD_DATA,A           ; Write to LCD
ACALL  WRCHAR_LCD           ;
ACALL  LCD_BLINK            ; Blink LCD
ACALL  WAIT_KEY             ; Wait until Keypad depressed
ACALL  WAIT_KEYPRESSED      ; Wait until Keypad pressed
MOV    BUFFER+1,KPAD_DATA   ; Copy in BUFFER+1
MOV    A,KPAD_DATA          ; Get Keypad Data
ADD    A,#030H              ; Convert to ASCII
MOV    LCD_DATA,A           ; Write to LCD
ACALL  WRCHAR_LCD           ;
ACALL  WAIT_KEY             ; Wait until Keypad depressed
MOV    A,BUFFER             ; Restore BUFFER
ANL    A,#00FH              ; Get lower 4 bit
SWAP   A                    ; Swap nibble
MOV    B,A                  ; Move to B
MOV    A,BUFFER+1           ; Restore BUFFER+1
ANL    A,#00FH              ; Get lower 4 bit
ADD    A,B                  ; Add BUFFER to make RTC BCD
MOV    HOURS,A              ; Keep Hours Data
MOV    LCD_ADDR,#043H       ; Set Address 43H
ACALL  SET_ADDR_LCD         ;
ACALL  LCD_BLINK            ; Blink LCD
ACALL  WAIT_KEYPRESSED      ; Wait until Keypad pressed
MOV    BUFFER,KPAD_DATA     ; Copy in BUFFER
MOV    A,KPAD_DATA          ; Get Keypad Data
ADD    A,#030H              ; Convert to ASCII
MOV    LCD_DATA,A           ; Write to LCD
ACALL  WRCHAR_LCD           ;
ACALL  LCD_BLINK            ; Blink LCD
ACALL  WAIT_KEY             ; Wait until Keypad depressed
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
ACALL  WAIT_KEYPRESSED      ; Wait until Keypad pressed
MOV    BUFFER+1,KPAD_DATA   ; Copy in BUFFER+1
MOV    A,KPAD_DATA          ; Get Keypad Data
ADD    A,#030H              ; Convert to ASCII
MOV    LCD_DATA,A           ; Write to LCD
ACALL  WRCHAR_LCD           ;
ACALL  WAIT_KEY             ; Wait until Keypad depressed
MOV    A,BUFFER             ; Restore BUFFER
ANL    A,#00FH              ; Get lower 4 bit
SWAP   A                    ; Swap nibble
MOV    B,A                  ; Move to B
MOV    A,BUFFER+1           ; Restore BUFFER+1
ANL    A,#00FH              ; Get lower 4 bit
ADD    A,B                  ; Add BUFFER to make RTC BCD
MOV    MINUTES,A            ; Keep Minutes Data
MOV    LCD_ADDR,#046H       ; Set Address 46H
ACALL  SET_ADDR_LCD         ;
ACALL  LCD_BLINK            ; Blink LCD
ACALL  WAIT_KEYPRESSED      ; Wait until Keypad pressed
MOV    BUFFER,KPAD_DATA     ; Copy in BUFFER
MOV    A,KPAD_DATA          ; Get Keypad Data
ADD    A,#030H              ; Convert to ASCII
MOV    LCD_DATA,A           ; Write to LCD
ACALL  WRCHAR_LCD           ;
ACALL  LCD_BLINK            ; Blink LCD
ACALL  WAIT_KEY             ; Wait until Keypad depressed
ACALL  WAIT_KEYPRESSED      ; Wait until Keypad pressed
MOV    BUFFER+1,KPAD_DATA   ; Copy in BUFFER+1
MOV    A,KPAD_DATA          ; Get Keypad Data
ADD    A,#030H              ; Convert to ASCII
MOV    LCD_DATA,A           ; Write to LCD
ACALL  WRCHAR_LCD           ;
ACALL  WAIT_KEY             ; Wait until Keypad depressed
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                MOV     A,BUFFER            ; Restore BUFFER
                ANL     A,#00FH             ; Get lower 4 bit
                SWAP    A                   ; Swap nibble
                MOV     B,A                 ; Move to B
                MOV     A,BUFFER+1          ; Restore BUFFER+1
                ANL     A,#00FH             ; Get lower 4 bit
                ADD     A,B                 ; Add BUFFER to make RTC BCD
                MOV     SECONDS,A           ; Keep Seconds Data
                ACALL   RTC_WR              ; Write RTC
                AJMP    LOOP                ; Jump to loop
;---------------------------------------------------------------
; BCD Code to show LCD
; I/P :         LCD_DATA
;---------------------------------------------------------------
BCD2LCD:        PUSH    ACC                 ; Push ACC to Stack
                PUSH    B                   ; Push B to Stack
                MOV     A,LCD_DATA          ; Get input data value
                MOV     B,A                 ; Copy to B
                ANL     A,#11110000B        ; Get higher 4 bit
                SWAP    A                   ; Swap nibble
                ADD     A,#030H             ; Convert to ASCII
                MOV     LCD_DATA,A          ; Write LCD
                ACALL   WRCHAR_LCD          ;
                MOV     A,B                 ; Restore value
                ANL     A,#00001111B        ; Get lower 4 bit
                ADD     A,#030H             ; Convert to ASCII
                MOV     LCD_DATA,A          ; Write LCD
                ACALL   WRCHAR_LCD          ;
                POP     B                   ; Pop B from Stack
                POP     ACC                 ; Pop ACC . from Stack
                RET                         ; Return
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
;-------------------------------------------------------------------
; Wait keypad depressed
;-------------------------------------------------------------------
WAIT_KEY:           MOV   A,P2                  ; Move P2 to ACC
                    ANL   A,#00FH               ; Get only Lower 4 bit
                    CJNE  A,#00FH,WAIT_KEY      ; All bit are high ?
                    RET
;-------------------------------------------------------------------
; Wait keypad pressed 0-9 Only
;-------------------------------------------------------------------
WAIT_KEYPRESSED:    ACALL GET_KPAD              ; Check keypad Data
                    MOV   A,KPAD_DATA           ;
                    CJNE  A,#0,CHK_KEY_NEXT     ; Check keypad pressed ?
                    AJMP  WAIT_KEYPRESSED       ; Loop until keypad pressed
CHK_KEY_NEXT:       CJNE  A,#10,CHK_KEY_0       ; Check key *
                    AJMP  WAIT_KEYPRESSED       ; If key * then loop wait
CHK_KEY_0:          CJNE  A,#11,CHK_VALID_KEY   ; Check key 0
                    MOV   KPAD_DATA,#0          ; Replace KPAD_DATA = 0
                    RET                         ; Return
CHK_VALID_KEY:      JNC   WAIT_KEYPRESSED       ; If KPAD_DATA > 11 then loop wait
                    RET                         ; Return
;-------------------------------------------------------------------
; Keypad Scan key Subroutine
;-------------------------------------------------------------------
GET_KPAD:           MOV   P2,#0FFH              ; Pull P2 to High ( 1-Wire not affect )
                    MOV   KPAD_DATA,#0          ; Clear Keypad Data
CHK_COL0:           CLR   KPAD_COL0             ; Begin Scan Column 0
                    MOV   A,P2                  ; Get Port2 Value
                    ANL   A,#00FH               ; Get only lower 4 bit
                    CJNE  A,#00FH,COL0_DETECT   ; Check All rows ' 1 ' ?
                    AJMP  CHK_COL1              ; All rows ' 1 ' => check next column
COL0_DETECT:        MOV   KPAD_DATA,#01         ; Initial KPAD_DATA = 1
                    AJMP  GET_ROW               ; Jump to get row value
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                    RET                         ; Return
;-------------------------------------------------------------------
; LCD Initialize
;-------------------------------------------------------------------
INIT_LCD:           ACALL  DELAY_100ms          ; Delay
                    CLR    LCD_RS               ; Clear LCD_RS Pin
                    MOV    P0,#00111000B        ; 8 bit Mode
                    ACALL  LCD_CLK              ; Pulse LCD Clock
                    ACALL  DELAY_10ms           ; Delay
                    MOV    P0,#00111000B        ; 8 bit Mode
                    ACALL  LCD_CLK              ; Pulse LCD Clock
                    ACALL  LCD_OFF              ; Display Off
                    ACALL  LCD_CLR              ; Clear Display
                    MOV    P0,#00000110B        ; Entry Mode
                    ACALL  LCD_CLK              ; Pulse LCD Clock
                    ACALL  LCD_HOME             ; Return Home Display
;-------------------------------------------------------------------
; LCD Clear Display
;-------------------------------------------------------------------
LCD_CLR:            CLR    LCD_RS               ; Clear LCD_RS Pin
                    MOV    P0,#00000001B        ; Display Clear
                    ACALL  LCD_CLK              ; Pulse LCD Clock
                    RET
;-------------------------------------------------------------------
; LCD Return Home
;-------------------------------------------------------------------
LCD_HOME:           CLR    LCD_RS               ; Clear LCD_RS Pin
                    MOV    P0,#00000010B        ; Return Home
                    ACALL  LCD_CLK              ; Pulse LCD Clock
                    RET
;-------------------------------------------------------------------
; LCD Display Off
;-------------------------------------------------------------------
LCD_OFF:            CLR    LCD_RS               ; Clear LCD_RS Pin
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                    MOV     P0,#00001000B           ; Display Off
                    ACALL   LCD_CLK                 ; Pulse LCD Clock
                    RET
;-------------------------------------------------------------------
;  LCD Clk
;-------------------------------------------------------------------
LCD_CLK:            SETB    LCD_EN                  ; Pulse Clock to LCD_EN
                    ACALL   LCD_DELAY
                    CLR     LCD_EN
                    ACALL   LCD_DELAY
                    RET
;-------------------------------------------------------------------
;  LCD Display On
;-------------------------------------------------------------------
LCD_ON:             CLR     LCD_RS                  ; Clear LCD_RS Pin
                    MOV     P0,#00001100B           ; Display On
                    ACALL   LCD_CLK
                    RET
;-------------------------------------------------------------------
;  LCD Cursor On
;-------------------------------------------------------------------
LCD_BLINK:          CLR     LCD_RS                  ; Clear LCD_RS Pin
                    MOV     P0,#00001111B           ; Display Cursor and Blink
                    ACALL   LCD_CLK                 ; Pulse LCD Clock
                    RET
;-------------------------------------------------------------------
;  Set  LCD Address
;  I/P :             LCD_ADDR
;-------------------------------------------------------------------
SET_ADDR_LCD:       CLR     LCD_RS                  ; Clear LCD_RS Pin
                    MOV     A,LCD_ADDR              ; Move LCD_ADDR to ACC.
                    SETB    ACC.7                   ; Set bit ACC . 7
                    MOV     P0,A                    ; Move to DATABUS
                    ACALL   LCD_CLK                 ; Pulse LCD Clock
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                    RET
;-------------------------------------------------------------------
; Write Character to show LCD
; I/P :             LCD_DATA
;-------------------------------------------------------------------
WRCHAR_LCD:         SETB    LCD_RS          ; Set LCD_RS Pin
                    MOV     P0, LCD_DATA    ; Move LCD_DATA to DATABUS
                    ACALL   LCD_CLK         ; Pulse LCD Clock
                    ACALL   LCD_ON          ; Display On
                    RET
;-------------------------------------------------------------------
; Write 3 Character from ROM
; I/P :             LCD_PTR
;-------------------------------------------------------------------
WR3CHAR_LCD:        MOV     R0,#0           ; Clear Loop Counter
                    MOV     A,LCD_PTR       ; Get LCD Pointer
                    DEC     A               ; Decrease Value
                    MOV     B,#3            ;
                    MUL     AB              ; Multiply by 3
WR3CHAR_LCD_1:      SETB    LCD_RS          ; Set LCD_RS Pin
                    MOVC    A,@A+DPTR       ; Get Data from ROM with Pointer
                    MOV     P0,A            ; Move ACC . to DATABUS
                    ACALL   LCD_CLK         ; Pulse LCD Clock
                    INC     DPTR            ; Increase Pointer
                    INC     R0              ; Increase Loop Pointer
                    MOV     A,LCD_PTR       ; Restore LCD Pointer
                    DEC     A               ; Decrease Value
                    MOV     B,#3            ;
                    MUL     AB              ; Multiply by 3
                    CJNE    R0,#3,WR3CHAR_LCD_1 ; Do until 3 times
                    LCALL   LCD_ON          ; Show LCD
                    RET                     ; Return
;-------------------------------------------------------------------
; Write Line of 16 Character from ROM
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
; I/P :              DPTR : Locate ROM Address
;-------------------------------------------------------------------------
WRLINE_LCD:        MOV    R0,#0                ; Clear loop counter
WRLINE_LCD_1:      SETB   LCD_RS               ; Set LCD_RS Pin
                   CLR    A                    ; Clear ACC .
                   MOVC   A,@A+DPTR            ; Move data from @DPTR to ACC .
                   MOV    P0,A                 ; Move ACC . to DATABUS
                   ACALL  LCD_CLK              ; Pulse LCD Clock
                   INC    DPTR                 ; Increase Pointer
                   INC    R0                   ; Increase loop counter
                   CJNE   R0,#8,WRLINE_LCD_1   ; Do until 8 times
                   MOV    LCD_ADDR,#040H       ; Set Later 8 Char . Address
                   ACALL  SET_ADDR_LCD
WRLINE_LCD_2:      SETB   LCD_RS               ; Set LCD_RS Pin
                   CLR    A                    ; Clear ACC .
                   MOVC   A,@A+DPTR            ; Move data from @DPTR to ACC .
                   MOV    P0,A                 ; Move ACC . to DATABUS
                   ACALL  LCD_CLK              ; Pulse LCD Clock
                   INC    DPTR                 ; Increase Pointer
                   INC    R0                   ; Increase loop counter
                   CJNE   R0,#16,WRLINE_LCD_2  ; Do until 8+8 times
                   ACALL  LCD_ON               ; Display On
                   RET
;-------------------------------------------------------------------------
; I2C RTC Read
; ------------------------------------------------------------------------
RTC_RD:            MOV    I2C_ADDR,#RTC_ID     ; Set RTC as I2C Write Slave
                   LCALL  I2C_SLAVE            ; Connect Slave
                   MOV    I2C_DATA,#000H       ; Set Slave Address 00H
                   LCALL  I2C_DATA_WR          ; Write Data to Slave
                   LCALL  RTC_RD1              ; Connect RTC Read
                   MOV    SECONDS,I2C_DATA     ; Read Data to SECONDS
                   LCALL  I2C_NACK_BIT         ; Send Not Acknowledge
                   LCALL  RTC_RD1              ; Connect RTC Read
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                MOV     MINUTES,I2C_DATA    ; Read Data to MINUTES
                LCALL   I2C_NACK_BIT        ; Send Not Acknowledge
                LCALL   RTC_RD1             ; Connect RTC Read
                MOV     HOURS,I2C_DATA      ; Read Data to HOURS
                LCALL   I2C_NACK_BIT        ; Send Not Acknowledge
                LCALL   RTC_RD1             ; Connect RTC Read
                MOV     DAY,I2C_DATA        ; Read Data to DAY
                LCALL   I2C_NACK_BIT        ; Send Not Acknowledge
                LCALL   RTC_RD1             ; Connect RTC Read
                MOV     DATE,I2C_DATA       ; Read Data to DATE
                LCALL   I2C_NACK_BIT        ; Send Not Acknowledge
                LCALL   RTC_RD1             ; Connect RTC Read
                MOV     MONTH,I2C_DATA      ; Read Data to MONTH
                LCALL   I2C_NACK_BIT        ; Send Not Acknowledge
                LCALL   RTC_RD1             ; Connect RTC Read
                MOV     YEAR,I2C_DATA       ; Read Data to YEAR
                LCALL   I2C_NACK_BIT        ; Send Not Acknowledge
                LCALL   RTC_RD1             ; Connect RTC Read
                MOV     CONTROL,I2C_DATA    ; Read Data to CONTROL
                LCALL   I2C_NACK_BIT        ; Send Not Acknowledge
                LCALL   I2C_STOP            ; Send Stop Condition
                RET                         ; Return
RTC_RD1:        MOV     I2C_ADDR,#RTC_ID+1  ; Set RTC as I2C Write Slave
                LCALL   I2C_SLAVE           ; Connect Slave
                LCALL   I2C_DATA_RD         ; Read Data from Slave
                RET
;-------------------------------------------------------------------
;  I2C RTC Write
;-------------------------------------------------------------------
RTC_WR:         MOV     I2C_ADDR,#RTC_ID    ; Set RTC as I2C Write Slave
                LCALL   I2C_SLAVE           ; Connect Slave
                MOV     I2C_DATA,#000H      ; Set Slave Address 00H
                LCALL   I2C_DATA_WR         ; Write Data to Slave
                MOV     I2C_DATA,SECONDS    ; Write SECONDS to RTC
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                    LCALL  I2C_DATA_WR          ;
                    MOV    I2C_DATA,MINUTES     ; Write MINUTES to RTC
                    LCALL  I2C_DATA_WR          ;
                    MOV    I2C_DATA,HOURS       ; Write HOURS to RTC
                    LCALL  I2C_DATA_WR          ;
                    MOV    I2C_DATA,DAY         ; Write DAY to RTC
                    LCALL  I2C_DATA_WR          ;
                    MOV    I2C_DATA,DATE        ; Write DATE to RTC
                    LCALL  I2C_DATA_WR          ;
                    MOV    I2C_DATA,MONTH       ; Write MONTH to RTC
                    LCALL  I2C_DATA_WR          ;
                    MOV    I2C_DATA,YEAR        ; Write YEAR to RTC
                    LCALL  I2C_DATA_WR          ;
                    MOV    I2C_DATA,CONTROL     ; Write CONTROL to RTC
                    LCALL  I2C_DATA_WR          ;
                    LCALL  I2C_STOP             ; Send Stop Condition
                    RET                         ; Return
;-------------------------------------------------------------------
; I2C Data Write
; I/P :          I2C_DATA
;-------------------------------------------------------------------
I2C_DATA_WR:        PUSH   ACC                  ; Push ACC .
                    SETB   I2C_ACK              ; Set ACK . bit
                    MOV    A, I2C_DATA          ; Get Data
                    MOV    R5,#008              ; Set loop 8 times
I2C_DATA_WR_1:      RLC    A                    ; Rotatae ACC . to Left with Carry
                    MOV    SDA,C                ; Move Carry Flag to SDA
                    LCALL  I2C_CLK              ; Pulse I2C Clock
                    DJNZ   R5,I2C_DATA_WR_1     ; Do until 8 times
                    SETB   SDA                  ; Set SDA
                    LCALL  I2C_DELAY            ; Delay
                    SETB   SCL                  ; Set SCL
                    LCALL  I2C_DELAY            ; Delay
                    JB     SDA,I2C_DATA_WR_2    ; Check Acknowledge from Slave
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
CHK_COL1:       SETB    KPAD_COL0           ; Stop Scan Column 0
                CLR     KPAD_COL1           ; Begin Scan Column 1
                MOV     A,P2                ; Get Port2 Value
                ANL     A,#00FH             ; Get only lower 4 bit
                CJNE    A,#00FH,COL1_DETECT ; Check All rows ' 1 ' ?
                AJMP    CHK_COL2            ; All rows ' 1 ' => check next column
COL1_DETECT:    MOV     KPAD_DATA,#02       ; Initial KPAD_DATA = 2
                AJMP    GET_ROW             ; Jump to get row value
CHK_COL2:       SETB    KPAD_COL1           ; Stop Scan Column 1
                CLR     KPAD_COL2           ; Begin Scan Column 2
                MOV     A,P2                ; Get Port2 Value
                ANL     A,#00FH             ; Get only lower 4 bit
                CJNE    A,#00FH,COL2_DETECT ; Check All rows ' 1 ' ?
                RET
COL2_DETECT:    MOV     KPAD_DATA,#03       ; Initial KPAD_DATA = 3
GET_ROW:        CLR     KPAD_COL0           ; Enable all Column to find Crosspoint
                CLR     KPAD_COL1           ;
                CLR     KPAD_COL2           ;
                JB      KPAD_ROW0,CHK_ROW1      ; Check Row 0 Detect ?
                RET                         ; Row 0 Detect => return
CHK_ROW1:       JB      KPAD_ROW1,CHK_ROW2      ; Check Row 2 Detect ?
                MOV     A,KPAD_DATA         ; Add 3 with KPAD_DATA
                ADD     A,#3                ;
                MOV     KPAD_DATA,A         ;
                RET                         ; Return
CHK_ROW2:       JB      KPAD_ROW2,CHK_ROW3      ; Check Row 2 Detect ?
                MOV     A,KPAD_DATA         ; Add 6 with KPAD_DATA
                ADD     A,#6                ;
                MOV     KPAD_DATA,A         ;
                RET                         ; Return
CHK_ROW3:       MOV     A,KPAD_DATA         ; Add 9 with KPAD_DATA
                ADD     A,#9                ;
                MOV     KPAD_DATA,A         ;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                    CLR     I2C_ACK             ; Clear ACK . bit
I2C_DATA_WR_2:      CLR     SCL                 ; Clear SCL
                    POP     ACC                 ; Pop ACC .
                    RET                         ; Return
;-------------------------------------------------------------------
; I2C Data Read
; O/P :             I2C_DATA
;-------------------------------------------------------------------
I2C_DATA_RD:        PUSH    ACC                 ; Push ACC .
                    CLR     A                   ; Clear ACC .
                    MOV     R5,#008             ; Set loop 8 times
I2C_DATA_RD_1:      LCALL   I2C_DELAY           ; Delay
                    SETB    SCL                 ; Set SCL
                    LCALL   I2C_DELAY           ; Delay
                    MOV     C,SDA               ; Get SDA to Carry Flag
                    RLC     A                   ; Rotate ACC . to Left with Carry
                    CLR     SCL                 ; Clear SCL
                    DJNZ    R5,I2C_DATA_RD_1    ; Do until 8 times
                    MOV     I2C_DATA,A          ; Move Data to I2C_DATA
                    POP     ACC                 ; Pop ACC .
                    RET                         ; Return
;-------------------------------------------------------------------
; I2C Slave Connect
; I/P :             I2C_ADDR
; O/P :             I2C_ACK
;-------------------------------------------------------------------
I2C_SLAVE:          PUSH    ACC                 ; Push ACC .
                    SETB    I2C_ACK             ; Set ACK . bit
                    MOV     A,I2C_ADDR          ; Get Slave Address
                    LCALL   I2C_START           ; Send Start Condition
                    MOV     R5,#008             ; Set loop 8 times
I2C_SLAVE_1:        RLC     A                   ; Rotate ACC . to Left with Carry
                    MOV     SDA,C               ; Move Carry Flag to SDA
                    LCALL   I2C_CLK             ; Pulse I2C Clock
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                DJNZ    R5,I2C_SLAVE_1      ; Do until 8 times
                SETB    SDA                 ; Set SDA
                LCALL   I2C_DELAY           ; Delay
                SETB    SCL                 ; Set SCL
                LCALL   I2C_DELAY           ; Delay
                JB      SDA,I2C_SLAVE_2     ; Check Acknowledge from Slave
                CLR     I2C_ACK             ; Clear ACK .
I2C_SLAVE_2:    CLR     SCL                 ; Clear SCL
                POP     ACC                 ; Pop ACC .
                RET                         ; Return
;-------------------------------------------------------------
;  I2C Start Condition
;-------------------------------------------------------------
I2C_START:      SETB    SCL                 ; Set SCL
                SETB    SDA                 ; Set SDA
                LCALL   I2C_DELAY           ; Delay
                CLR     SDA                 ; Clear SDA during SCL set
                LCALL   I2C_DELAY           ; Delay
                CLR     SCL                 ; Clear SCL
                RET                         ; Return
;-------------------------------------------------------------
;  I2C Stop Condition
;-------------------------------------------------------------
I2C_STOP:       CLR     SDA                 ; Clear SDA
                LCALL   I2C_DELAY           ; Delay
                SETB    SCL                 ; Set SCL
                LCALL   I2C_DELAY           ; Delay
                SETB    SDA                 ; Set SDA during SCL set
                RET                         ; Return
;-------------------------------------------------------------
;  I2C Clock
;-------------------------------------------------------------
I2C_CLK:        LCALL   I2C_DELAY           ; Pulse SCL
                SETB    SCL                 ;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                    LCALL  I2C_DELAY            ;
                    CLR    SCL                  ;
                    RET                         ; Return
;-------------------------------------------------------------------
; I2C Not Acknowledge
;-------------------------------------------------------------------
I2C_NACK_BIT:       SETB   SDA                  ; Set SDA
                    LCALL  I2C_DELAY            ; Delay
                    LCALL  I2C_CLK              ; Pulse I2C Clock
                    RET                         ; Return
;-------------------------------------------------------------------
; Dummy Delay time I2C_DELAY, LCD_DELAY, 10m, 100m, 1s
;-------------------------------------------------------------------
I2C_DELAY:          MOV    6,#00CH              ; Each loop = 50 us
I2C_DELAY_1:        NOP
                    NOP
                    DJNZ   R6,I2C_DELAY_1
                    RET
LCD_DELAY:          MOV    7,#002               ; Do 2 times
LCD_DELAY_1:        MOV    6,#0E6H              ; Each loop = 1 ms
LCD_DELAY_2:        NOP
                    NOP
                    DJNZ   R6,LCD_DELAY_2
                    DJNZ   R7,LCD_DELAY_1
                    RET
DELAY_10ms:         MOV    7,#010               ; Do 10 times
DELAY_10ms_1:       MOV    6,#0E6H              ; Each loop = 1 ms
DELAY_10ms_2:       NOP
                    NOP
                    DJNZ   R6,DELAY_10ms_2
                    DJNZ   R7,DELAY_10ms_1
                    RET
DELAY_100ms:        MOV    7,#100               ; Do 100 times
DELAY_100ms_1:      MOV    6,#0E6H              ; Each loop = 1 ms
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
DELAY_100ms_2:  NOP
                NOP
                DJNZ    R6,DELAY_100ms_2
                DJNZ    R7,DELAY_100ms_1
                RET
DELAY_1s:       MOV     5,#100              ; Do 100 times
DELAY_1s_1:     ACALL   DELAY_10ms
                DJNZ    R5,DELAY_1s_1
                RET
DELAY_1sec:     MOV     5,#100
                JB      SET_DATE_SW,CHECK_SW_2      ; Check Switch Every 10 ms
                JMP     SET_DATE            ; Jump to Set Date
CHECK_SW_2:     JB      SET_TIME_SW,DELAY_1sec_1;
                JMP     SET_TIME            ; Jump to Set Time
DELAY_1sec_1:   ACALL   DELAY_10ms
                DJNZ    R5,DELAY_1sec_1
                RET
DELAY_1sex:     MOV     4,#001              ; Do 1 times
DELAY_1sex_1:   ACALL   DELAY_1sec
                ACALL   WRITE_TIME_1
                DJNZ    R4,DELAY_1sex_1
                RET
DELAY_5sec:     MOV     4,#005              ; Do 5 times
DELAY_5sec_1:   ACALL   DELAY_1sex
                DJNZ    R4,DELAY_5sec_1
                RET
WRITE_TIME_1:   ACALL   RTC_RD
                MOV     LCD_ADDR,#043H      ; Set Address 43H
                ACALL   SET_ADDR_LCD        ;
                MOV     A,HOURS             ; Get Hour
                ANL     A,#00110000B        ; Get x10 Digit
                JZ      WRITE_TIME_HN_1     ; Check x10 Digit = 0 ?
                SWAP    A                   ; If not => Write to LCD
                ADD     A,#030H             ; Convert to ASCII
```

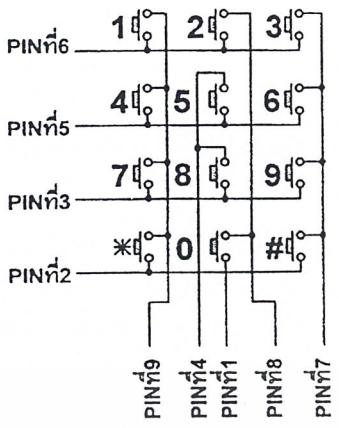
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

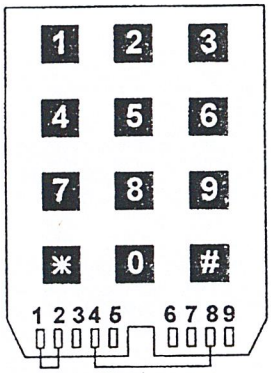
```
                    AJMP    WRITE_TIME_HH_1     ;
WRITE_TIME_HN_1:    MOV     A,#' '              ; x10 = 0 then Write space
WRITE_TIME_HH_1:    MOV     LCD_DATA,A          ; Write to LCD
                    ACALL   WRCHAR_LCD          ;
                    MOV     A,HOURS             ; Write x1 Digit to LCD
                    ANL     A,#00001111B        ;
                    ADD     A,#030H             ; Convert to ASCII
                    MOV     LCD_DATA,A          ;
                    ACALL   WRCHAR_LCD          ;
                    MOV     A,SECONDS           ; Check Second = Odd Number ?
                    ANL     A,#001H             ;
                    JNZ     WRITE_SPACE_1       ; Even => Write space
                    MOV     LCD_DATA,#':'       ; Odd => Write ':'
                    ACALL   WRCHAR_LCD          ;
                    AJMP    WRITE_MINUTES_1     ;
WRITE_SPACE_1:      MOV     LCD_DATA,#' '       ;
                    ACALL   WRCHAR_LCD          ;
WRITE_MINUTES_1:    MOV     LCD_DATA,MINUTES    ; Write Minute to LCD
                    ACALL   BCD2LCD             ; Write from RTC BCD Data
                    RET                         ;
;-------------------------------------------------------------
; Define Constant < Store in Flag EEPROM Program Memory >
;-------------------------------------------------------------
;                   0123456789ABCDEF
TITLE_1:            DB      'Real Time Clock '
SCR_SET_DATE:       DB      ' Date : dd/mm/yy'
SCR_SET_TIME:       DB      ' Time : hh/mm/ss'
MONTH_JAN:          DB      'Jan'
MONTH_FEB:          DB      'Feb'
MONTH_MAR:          DB      'Mar'
MONTH_APR:          DB      'Apr'
MONTH_MAY:          DB      'May'
MONTH_JUN:          DB      'Jun'
MONTH_JUL:          DB      'Jul'
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
MONTH_AUG:      DB    'Aug'
MONTH_SEP:      DB    'Sep'
MONTH_OCT:      DB    'Oct'
MONTH_NOV:      DB    'Nov'
MONTH_DEC:      DB    'Dec'
                END
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

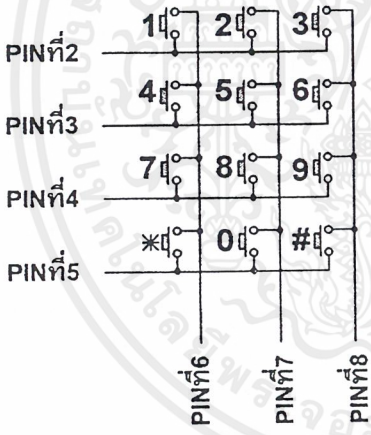


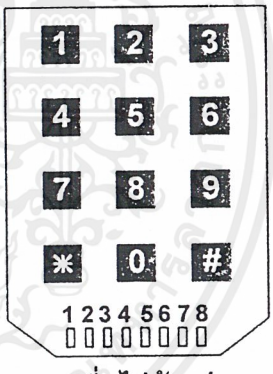


การต่อให้เป็น 4X3
ต้องมีการ JUM

รุ่นที่ 1

KEY 4X3 DECODINGS





PIN ที่1 ไม่ต้องต่อ

รุ่นที่ 2

KEY 4X3 DECODINGS

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

INTEGRATED CIRCUITS
UNITRODE

L293
L293D

Push-Pull Four Channel Driver

FEATURES

- Output Current 1A Per Channel (600mA for L293D)
- Peak Output Current 2A Per Channel (1.2A for L293D)
- Inhibit Facility
- High Noise Immunity
- Separate Logic Supply
- Over-Temperature Protection

DESCRIPTION

The L293 and L293D are quad push-pull drivers capable of delivering output currents to 1A or 600mA per channel respectively. Each channel is controlled by a TTL-compatible logic input and each pair of drivers (a full bridge) is equipped with an inhibit input which turns off all four transistors. A separate supply input is provided for the logic so that it may be run off a lower voltage to reduce dissipation.

Additionally the L293D includes the output clamping diodes within the IC for complete interfacing with inductive loads.

Both devices are available in 16-pin Batwing DIP packages. They are also available in Power SOIC and Hermetic DIL packages.

TRUTH TABLE

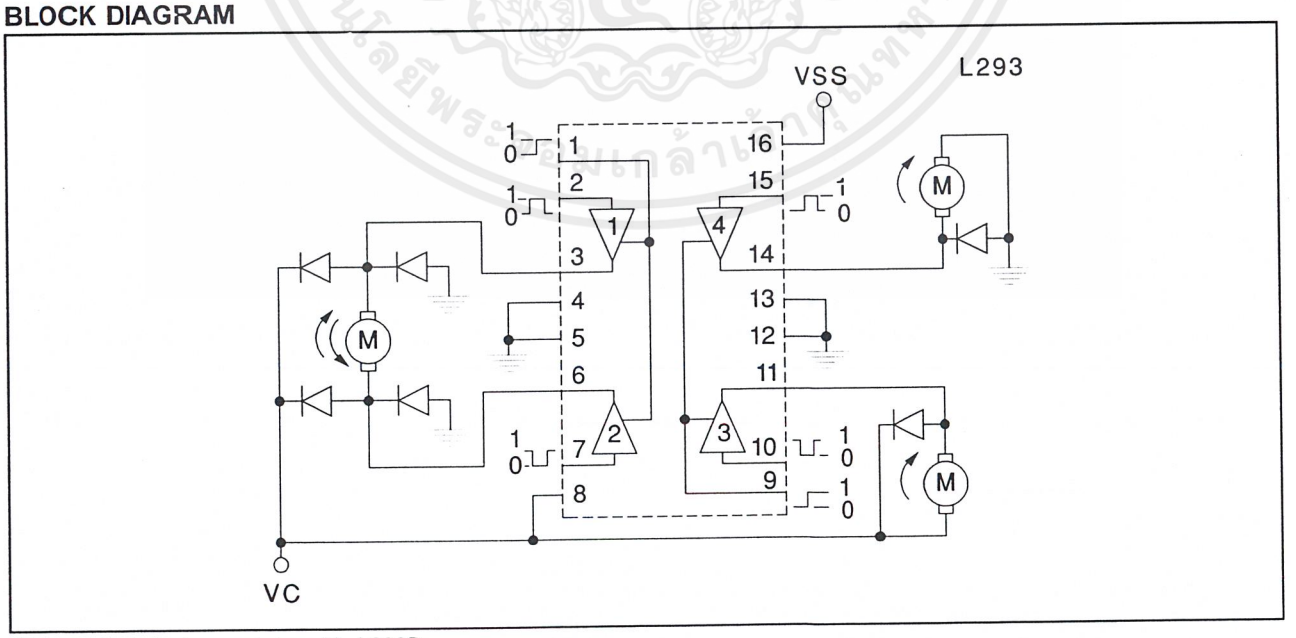
V_I (each channel)	V_{INH}*	V_O
H	H	H
L	H	L
H	L	X**
L	L	X**

**Relative to the considered channel*
***High output impedence*

ABSOLUTE MAXIMUM RATINGS

Collector Supply Voltage, V_C 36V
Logic Supply Voltage, V_{SS} 36V
Input Voltage, V_I 7V
Inhibit Voltage, V_{INH} 7V
Peak Output Current (Non-Repetitive), I_{OUT} (L293) 2A
I_{OUT} (L293D) 1.2A
Total Power Dissipation
at Tground-pins = 80°C, N Batwing pkg, (Note) 5W
Storage and Junction Temperature, T_{stg}, T_J . . . . . -40 to +150°C
Note:Consult packaging section of Databook for thermal limitations and considerations of packages.

BLOCK DIAGRAM

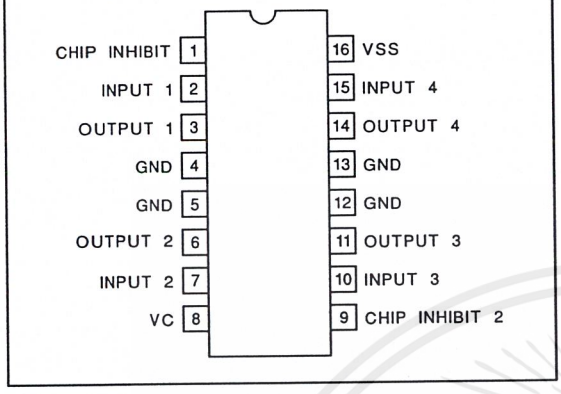


Note: Output diodes are internal in L293D.

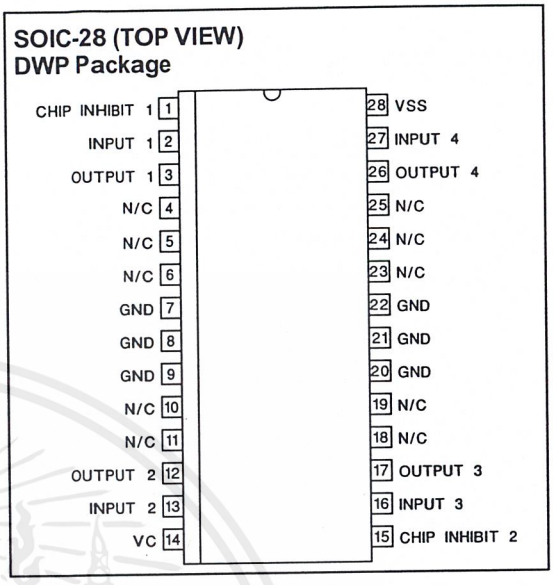
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

CONNECTION DIAGRAMS

DIL-16 (TOP VIEW)
N Package, SP Package



SOIC-28 (TOP VIEW)
DWP Package



ELECTRICAL CHARACTISTICS: (For each channel, V_C = 24V, V_{SS} = 5V, T_{AMB} = 25°C, unless otherwise specified; $T_A = T_J$)

PARAMETER	SYMBOL	TEST CONDITION	MIN.	TYP.	MAX.	UNITS
Collector Supply Voltage	V_C				36	V
Logic Supply Voltage	V_{SS}		4.5		36	V
Collector Supply Current	I_C	V_I = L, I_O = 0, V_{INH} = H		2	6	mA
		V_I = H, I_O = 0, V_{INH} = H		16	24	mA
		V_{INH} = L			4	mA
Total Quiescent Logic Supply Current	I_{SS}	V_I = L, I_O = 0, V_{INH} = H		44	60	mA
		V_I = H, I_O = 0, V_{INH} = H		16	22	mA
		V_{INH} = L		16	24	mA
Input Low Voltage	V_{IL}		-0.3		1.5	V
Input High Voltage	V_{IH}	$V_{SS} \leq 7V$	2.3		Vss	V
		$V_{SS} \geq 7V$	2.3		7	V
Low Voltage Input Current	I_{IL}	V_I = 0V			-10	µA
High Voltage Input Current	I_{IH}	V_I = 4.5V		30	100	µA
Inhibit Low Voltage	$V_{INH, L}$		-0.3		1.5	V
Inhibit High Voltage	$V_{INH, H}$	$V_{SS} \leq 7V$	2.3		Vss	V
		V_{SS} >7V	2.3		7	V
Low Voltage Inhibit Current	$V_{INH, L}$			-30	-100	µA
High Voltage Inhibit Current	$V_{INH, H}$				10	µA
Source Output Saturation Voltage	V_{CEsatH}	I_O = -1A (-0.6A for L293D)		1.4	1.8	V
Sink Output Saturation Voltage	V_{CEsatL}	I_O = 1A (0.6A for L293D)		1.2	1.8	V
Clamp Diode Forward Voltage (L293D only)	V_F	I_F = 0.6A		1.3		V
Rise Time	T_R	0.1 to 0.9 V_O (See Figure 1)		100		ns
Fall Time	T_F	0.9 to 0.1 V_O (See Figure 1)		350		ns
Turn-on Delay	T_{ON}	0. 5 V_I to 0.5 V_O (See Figure 1)		750		ns
Turn-off Delay	T_{OFF}	0. 5 V_I to 0.5 V_O (See Figure 1)		200		ns

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Octal buffer/line driver; 3-state

74HC/HCT541

FEATURES

- Non-inverting outputs
- Output capability: bus driver
- I_{CC} category: MSI

GENERAL DESCRIPTION

The 74HC/HCT541 are high-speed Si-gate CMOS devices and are pin compatible with low power Schottky TTL (LSTTL). They are specified in compliance with JEDEC standard no. 7A.

The 74HC/HCT541 are octal non-inverting buffer/line drivers with 3-state outputs. The 3-state outputs are controlled by the output enable inputs $\overline{OE}_1$ and $\overline{OE}_2$. A HIGH on $\overline{OE}_n$ causes the outputs to assume a high impedance OFF-state.
The "541" is identical to the "540" but has non-inverting outputs.

QUICK REFERENCE DATA

GND = 0 V; T_{amb} = 25 °C; $t_r = t_f$ = 6 ns

SYMBOL	PARAMETER	CONDITIONS	TYPICAL		UNIT
			HC	HCT	
t_{PHL}/ t_{PLH}	propagation delay A_n to Y_n	C_L = 15 pF; V_{CC} = 5 V	10	12	ns
C_I	input capacitance		3.5	3.5	pF
C_{PD}	power dissipation capacitance per buffer	notes 1 and 2	37	39	pF

Notes

1. C_{PD} is used to determine the dynamic power dissipation (P_D in µW):

   $P_D = C_{PD} \times V_{CC}^2 \times f_i + \Sigma (C_L \times V_{CC}^2 \times f_o)$ where:

   f_i = input frequency in MHz

   f_o = output frequency in MHz

   $\Sigma (C_L \times V_{CC}^2 \times f_o)$ = sum of outputs

   C_L = output load capacitance in pF

   V_{CC} = supply voltage in V

2. For HC the condition is V_I = GND to V_{CC}
   For HCT the condition is V_I = GND to V_{CC} – 1.5 V

ORDERING INFORMATION

See *"74HC/HCT/HCU/HCMOS Logic Package Information"*.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

PIN DESCRIPTION

PIN NO.	SYMBOL	NAME AND FUNCTION
1, 19	$\overline{OE}_1$, $\overline{OE}_2$	output enable input (active LOW)
2, 3, 4, 5, 6, 7, 8, 9	A_0 to A_7	data inputs
10	GND	ground (0 V)
18, 17, 16, 15, 14, 13, 12, 11	Y_0 to Y_7	bus outputs
20	V_{CC}	positive supply voltage

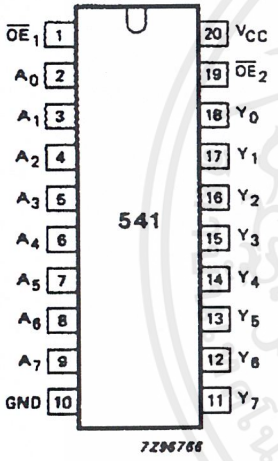


Fig.1 Pin configuration.

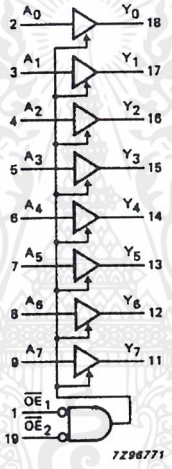


Fig.2 Logic symbol.

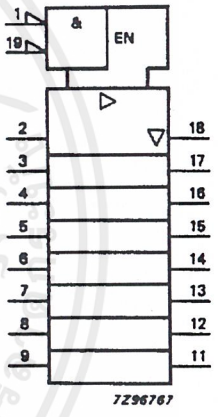


Fig.3 IEC logic symbol.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

FAIRCHILD
SEMICONDUCTOR TM

www.fairchildsemi.com

MC78XX/LM78XX

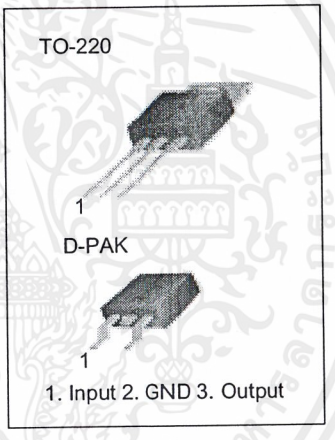
3-terminal 1A positive voltage regulator

Features

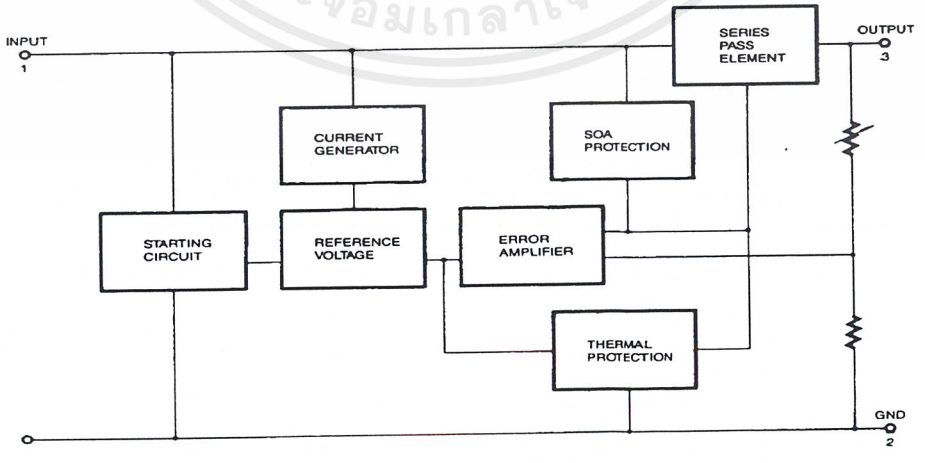
- Output Current up to 1A
- Output Voltages of 5, 6, 8, 9, 10, 11, 12, 15, 18, 24V
- Thermal Overload Protection
- Short Circuit Protection
- Output Transistor Safe Operating area Protection

Description

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



Internal Block Digram



Rev. 5.0

©2000 Fairchild Semiconductor International

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Absolute Maximum Ratings

Parameter	Symbol	Value	Unit
Input Voltage (for V_O = 5V to 18V) (for V_O = 24V)	V_I V_I	35 40	V V
Thermal Resistance Junction-Cases	$R_{\theta JC}$	5	°C/W
Thermal Resistance Junction-Air	$R_{\theta JA}$	65	°C/W
Operating Temperature Range (MC78XXCT/LM78XXCT/MC78XXCDT)	T_{OPR}	0 ~ +125	°C
Storage Temperature Range	T_{STG}	-65 ~ +150	°C

Electrical Characteristics (MC7805/LM7805)

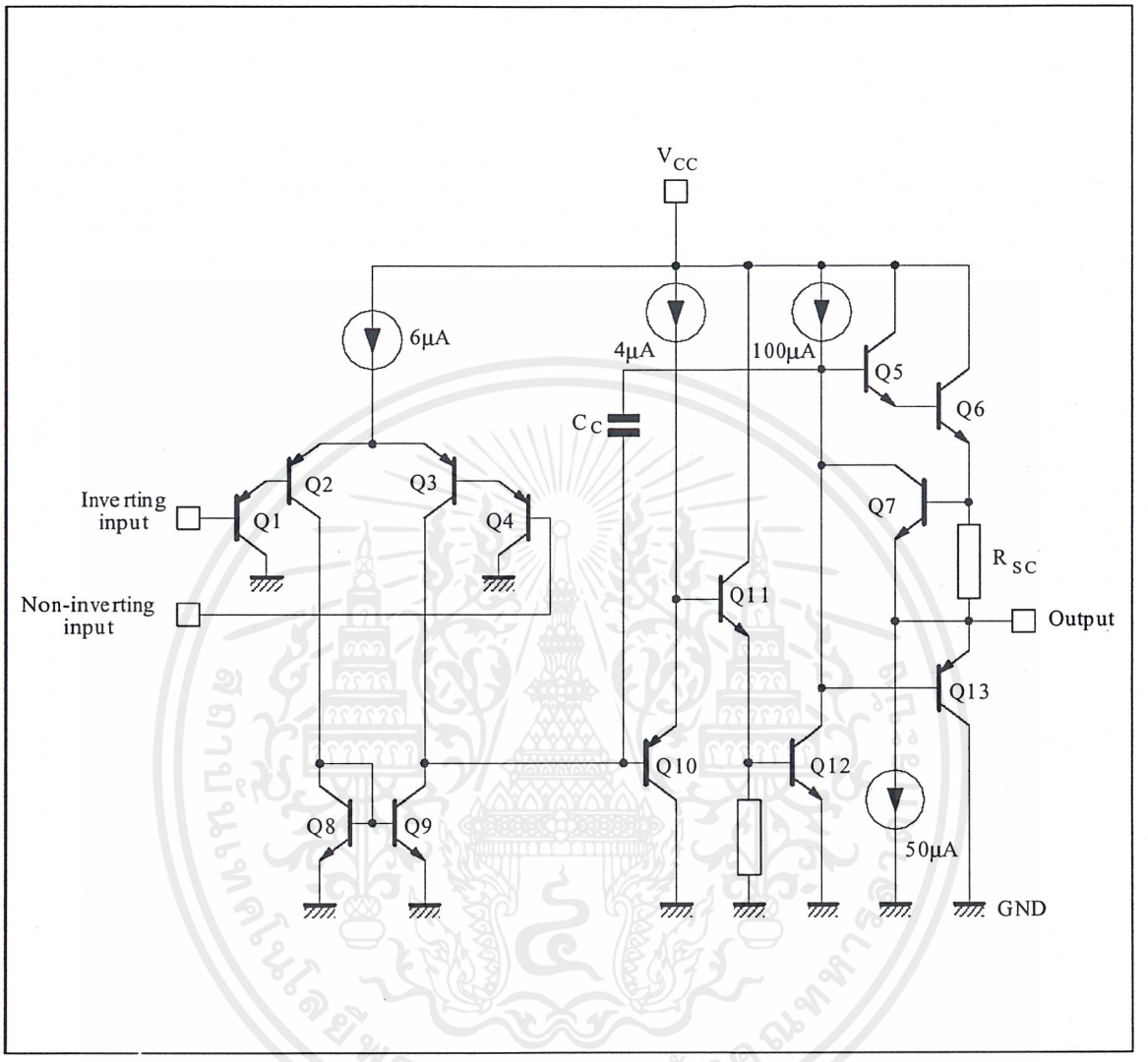
(Refer to test circuit ,0°C < T_J < 125°C, I_O = 500mA, V_I = 10V, C_I= 0.33μF, C_O= 0.1μF, unless otherwise specified)

Parameter	Symbol	Conditions		MC7805/LM7805			Unit
				Min.	Typ.	Max.	
Output Voltage	V_O	T_J =+25 °C		4.8	5.0	5.2	V
		5.0mA ≤ I_O ≤ 1.0A, P_O ≤ 15W V_I = 7V to 20V V_I = 8V to 20V		4.75	5.0	5.25	
Line Regulation	ΔV_O	T_J=+25 °C	V_O = 7V to 25V	-	4.0	100	mV
			V_I = 8V to 12V	-	1.6	50	
Load Regulation	ΔV_O	T_J=+25 °C	I_O = 5.0mA to1.5A	-	9	100	mV
			I_O =250mA to 750mA	-	4	50	
Quiescent Current	I_Q	T_J =+25 °C		-	5.0	8	mA
Quiescent Current Change	ΔI_Q	I_O = 5mA to 1.0A		-	0.03	0.5	mA
		V_I= 7V to 25V		-	0.3	1.3	
Output Voltage Drift	$\Delta V_O/\Delta T$	I_O= 5mA		-	-0.8	-	mV/ °C
Output Noise Voltage	V_N	f = 10Hz to 100KHz, T_A=+25 °C		-	42	-	μV
Ripple Rejection	RR	f = 120Hz V_O = 8V to 18V		62	73	-	dB
Dropout Voltage	V_O	I_O = 1A, T_J =+25 °C		-	2	-	V
Output Resistance	R_O	f = 1KHz		-	15	-	mΩ
Short Circuit Current	I_{SC}	V_I = 35V, T_A =+25 °C		-	230	-	mA
Peak Current	I_{PK}	T_J =+25 °C		-	2.2	-	A

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

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

SCHEMATIC DIAGRAM (1/4 LM124)



ABSOLUTE MAXIMUM RATINGS

Symbol	Parameter	LM124	LM224	LM324	Unit
V_{cc}	Supply Voltage	±16 or 32			V
V_i	Input Voltage	-0.3 to +32			V
V_{id}	Differential Input Voltage - (*)	+32	+32	+32	V
P_{tot}	Power Dissipation N Suffix D Suffix	500 -	500 400	500 400	mW mW
-	Output Short-circuit Duration - (note 1)	Infinite			
I_{in}	Input Current – (note 6)	50	50	50	mA
T_{oper}	Operating Free Air Temperature Range	-55 to +125	-40 to +105	0 to +70	°C
T_{stg}	Storage Temperature Range	-65 to +150	-65 to +150	-65 to +150	°C

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

DALLAS SEMICONDUCTOR

DS1307
64 X 8 Serial Real Time Clock

www.dalsemi.com

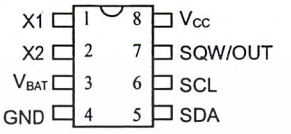
FEATURES

- 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 with oscillator running
- Optional industrial temperature range -40°C to +85°C
- Available in 8-pin DIP or SOIC
- Recognized by Underwriters Laboratory

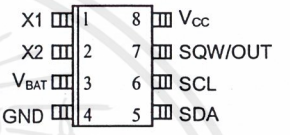
ORDERING INFORMATION

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

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}	- +3V Battery Input
GND	- Ground
SDA	- Serial Data
SCL	- Serial Clock
SQW/OUT	- Square wave/Output Driver

DESCRIPTION

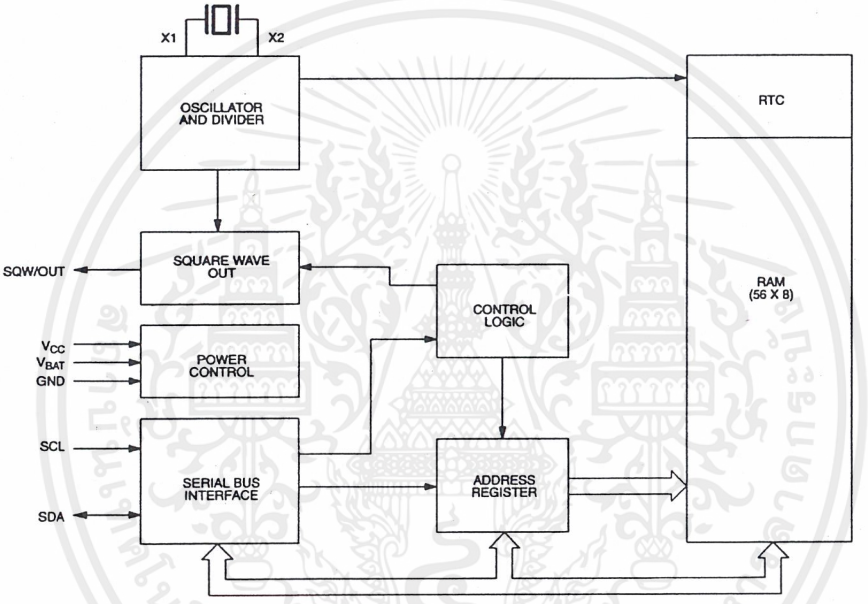
The DS1307 Serial Real Time Clock is a low power, full BCD clock/calendar plus 56 bytes of nonvolatile SRAM. Address and data are transferred serially via a 2-wire bi-directional bus. The clock/calendar provides seconds, minutes, hours, day, date, month, and year information. The end of the month date 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.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

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.

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 is 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 power supply (nominal 3.0V DC) at V_{BAT}.

V_{BAT} - Battery input for any standard 3-volt lithium cell or other energy source. Battery voltage must be held between 2.0 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 48 mAhr or greater will back up the DS1307 for more than 10 years in the absence of power at 25 degrees C.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

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

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

SQW/OUT (Square Wave/ Output Driver) - When enabled, the SQWE bit set to 1, the SQW/OUT pin outputs one of four square wave frequencies (1 Hz, 4 kHz, 8 kHz, 32 kHz). The SQW/OUT pin is open drain which requires an external pullup resistor. SQW/OUT will operate with either Vcc or Vbat applied.

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.

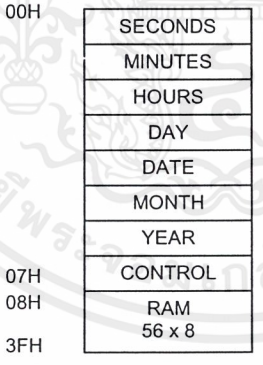
For more information on crystal selection and crystal layout considerations, please consult Application Note 58, "Crystal Considerations with Dallas Real Time Clocks." The DS1307 can also be driven by an external 32.768 kHz oscillator. In this configuration, the X1 pin is connected to the external oscillator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

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

RTC AND RAM ADDRESS MAP

The address map for the RTC and RAM registers of the DS1307 is shown in Figure 2. The 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 multi-byte access, when the address pointer reaches 3Fh, the end of RAM space, it wraps around to location 00h, the beginning of the clock space.

DS1307 ADDRESS MAP Figure 2



CLOCK AND CALENDAR

The time and calendar information is obtained by reading the appropriate register bytes. The 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 1, the oscillator is disabled. When cleared to a 0, the oscillator is enabled.

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

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

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

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

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	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 upon 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	4.096 kHz
1	0	8.192 kHz
1	1	32.768 kHz

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Features

- Compatible with MCS-51™ Products
- 4K Bytes of In-System Reprogrammable Flash Memory
  - Endurance: 1,000 Write/Erase Cycles
- Fully Static Operation: 0 Hz to 24 MHz
- Three-Level Program Memory Lock
- 128 x 8-Bit Internal RAM
- 32 Programmable I/O Lines
- Two 16-Bit Timer/Counters
- Six Interrupt Sources
- Programmable Serial Channel
- Low Power Idle and Power Down Modes

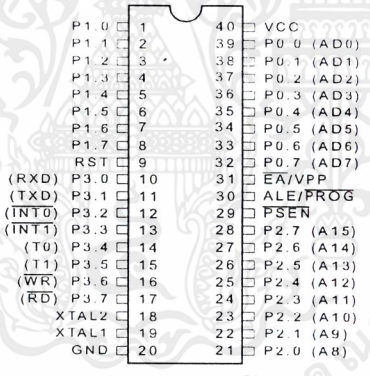
Description

The AT89C51 is a low-power, high-performance CMOS 8-bit microcomputer with 4K bytes of Flash Programmable and Erasable Read Only Memory (PEROM). The device is manufactured using Atmel's high density nonvolatile memory technology and is compatible with the industry standard MCS-51™ instruction set and pinout. The on-chip Flash allows the program memory to be reprogrammed in-system or by a conventional nonvolatile memory programmer. By combining a versatile 8-bit CPU with Flash on a monolithic chip, the Atmel AT89C51 is a powerful microcomputer which provides a highly flexible and cost effective solution to many embedded control applications.

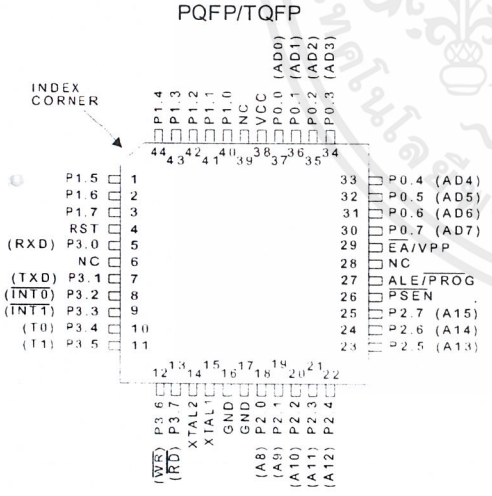
(continued)

Pin Configurations

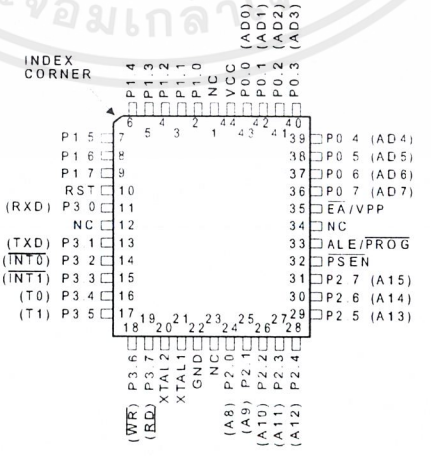
PDIP



PQFP/TQFP



PLCC



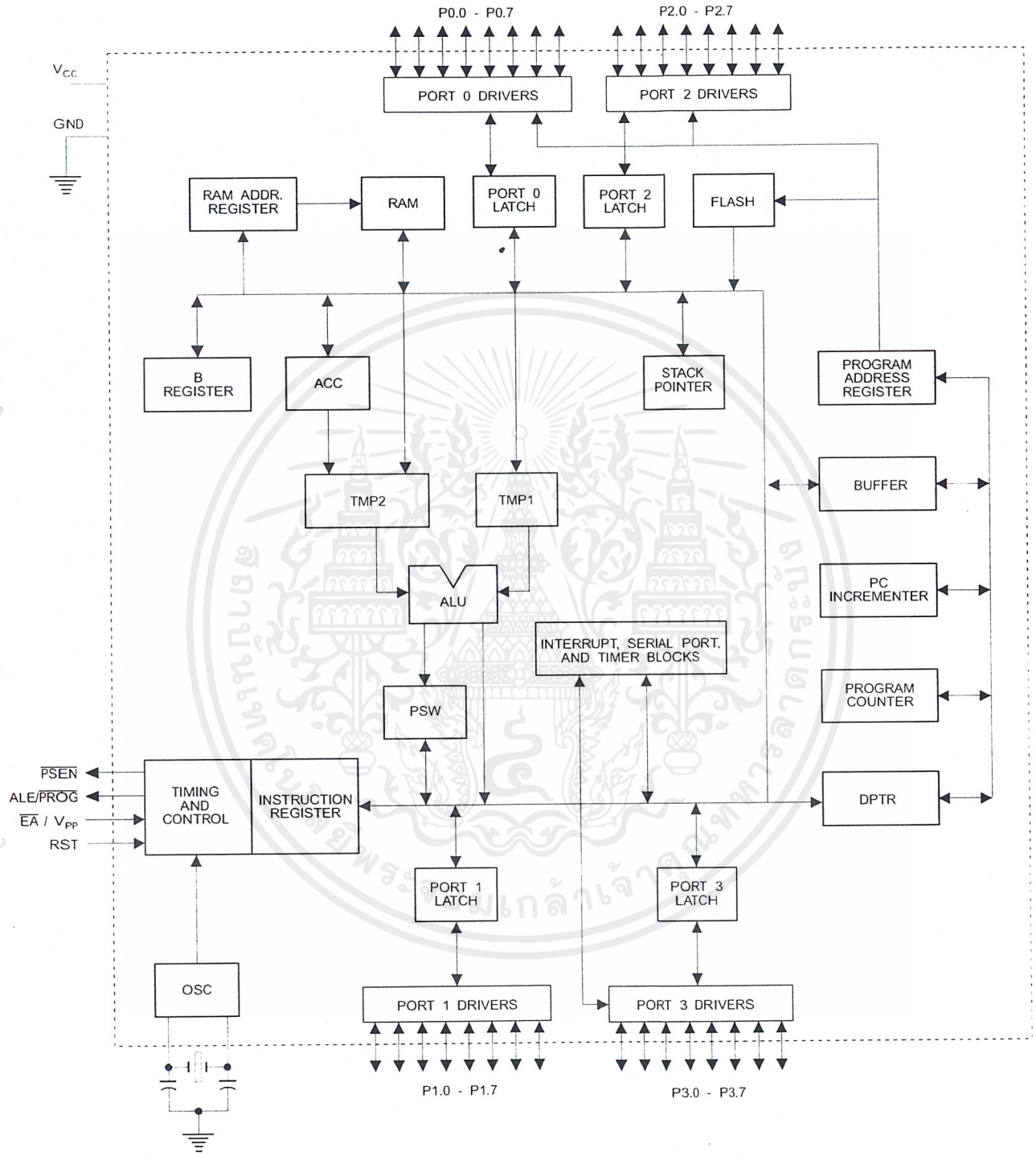
8-Bit Microcontroller with 4K Bytes Flash

AT89C51

0265F-A–12/97

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Block Diagram



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

The AT89C51 provides the following standard features: 4K bytes of Flash, 128 bytes of RAM, 32 I/O lines, two 16-bit timer/counters, a five vector two-level interrupt architecture, a full duplex serial port, on-chip oscillator and clock circuitry. In addition, the AT89C51 is designed with static logic for operation down to zero frequency and supports two software selectable power saving modes. The Idle Mode stops the CPU while allowing the RAM, timer/counters, serial port and interrupt system to continue functioning. The Power Down Mode saves the RAM contents but freezes the oscillator disabling all other chip functions until the next hardware reset.

Pin Description

V_{CC}
Supply voltage.

GND
Ground.

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

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

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

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

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

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

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

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

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

Port 3 also serves the functions of various special features of the AT89C51 as listed below:

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

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

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

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

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

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

$\overline{PSEN}$
Program Store Enable is the read strobe to external program memory.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

When the AT89C51 is executing code from external program memory, $\overline{PSEN}$ is activated twice each machine cycle, except that two $\overline{PSEN}$ activations are skipped during each access to external data memory.

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

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

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

This pin also receives the 12-volt programming enable voltage (V_{PP}) during Flash programming, for parts that require 12-volt V_{PP}.

XTAL1

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

XTAL2

Output from the inverting oscillator amplifier.

Oscillator Characteristics

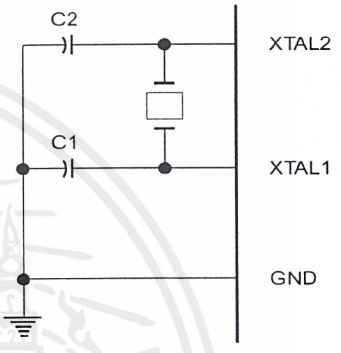
XTAL1 and XTAL2 are the input and output, respectively, of an inverting amplifier which can be configured for use as an on-chip oscillator, as shown in Figure 1. Either a quartz crystal or ceramic resonator may be used. To drive the device from an external clock source, XTAL2 should be left unconnected while XTAL1 is driven as shown in Figure 2. There are no requirements on the duty cycle of the external clock signal, since the input to the internal clocking circuitry is through a divide-by-two flip-flop, but minimum and maximum voltage high and low time specifications must be observed.

Idle Mode

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

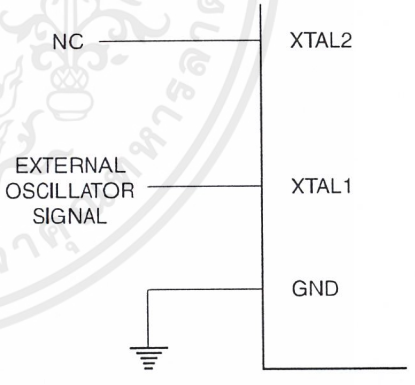
It should be noted that when idle is terminated by a hard ware reset, the device normally resumes program execution, from where it left off, up to two machine cycles before the internal reset algorithm takes control. On-chip hardware inhibits access to internal RAM in this event, but access to the port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

Figure 1. Oscillator Connections



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

Figure 2. External Clock Drive Configuration



Status of External Pins During Idle and Power Down Modes

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

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Power Down Mode

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

Program Memory Lock Bits

On the chip are three lock bits which can be left unprogrammed (U) or can be programmed (P) to obtain the additional features listed in the table below:

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

Lock Bit Protection Modes

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

Programming the Flash

The AT89C51 is normally shipped with the on-chip Flash memory array in the erased state (that is, contents = FFH) and ready to be programmed. The programming interface accepts either a high-voltage (12-volt) or a low-voltage (V_{CC}) program enable signal. The low voltage programming mode provides a convenient way to program the AT89C51 inside the user's system, while the high-voltage programming mode is compatible with conventional third party Flash or EPROM programmers.

The AT89C51 is shipped with either the high-voltage or low-voltage programming mode enabled. The respective top-side marking and device signature codes are listed in the following table.

	V_{PP} = 12V	V_{PP} = 5V
Top-Side Mark	AT89C51 xxxx yyww	AT89C51 xxxx-5 yyww
Signature	(030H)=1EH (031H)=51H (032H)=FFH	(030H)=1EH (031H)=51H (032H)=05H

The AT89C51 code memory array is programmed byte-by-byte in either programming mode. *To program any non-blank byte in the on-chip Flash Memory, the entire memory must be erased using the Chip Erase Mode.*

Programming Algorithm: Before programming the AT89C51, the address, data and control signals should be set up according to the Flash programming mode table and Figures 3 and 4. To program the AT89C51, take the following steps.

1. Input the desired memory location on the address lines.
2. Input the appropriate data byte on the data lines.
3. Activate the correct combination of control signals.
4. Raise $\overline{EA}/V_{PP}$ to 12V for the high-voltage programming mode.
5. Pulse ALE/$\overline{PROG}$ once to program a byte in the Flash array or the lock bits. The byte-write cycle is self-timed and typically takes no more than 1.5 ms. Repeat steps 1 through 5, changing the address and data for the entire array or until the end of the object file is reached.

$\overline{Data}$ Polling: The AT89C51 features $\overline{Data}$ Polling to indicate the end of a write cycle. During a write cycle, an attempted read of the last byte written will result in the complement of the written datum on PO.7. Once the write cycle has been completed, true data are valid on all outputs, and the next cycle may begin. $\overline{Data}$ Polling may begin any time after a write cycle has been initiated.

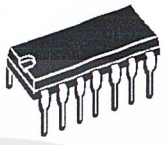
Ready/$\overline{Busy}$: The progress of byte programming can also be monitored by the RDY/$\overline{BSY}$ output signal. P3.4 is pulled low after ALE goes high during programming to indicate BUSY. P3.4 is pulled high again when programming is done to indicate READY.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

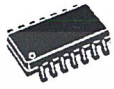
LM124
LM224 - LM324

LOW POWER QUAD OPERATIONAL AMPLIFIERS

- WIDE GAIN BANDWIDTH : 1.3MHz
- INPUT COMMON-MODE VOLTAGE RANGE INCLUDES GROUND
- LARGE VOLTAGE GAIN : 100dB
- VERY LOW SUPPLY CURRENT/AMPLI : 375µA
- LOW INPUT BIAS CURRENT : 20nA
- LOW INPUT OFFSET VOLTAGE : 5mV max.
  (for more accurate applications, use the equivalent parts LM124A-LM224A-LM324A which feature 3mV max)
- LOW INPUT OFFSET CURRENT : 2nA
- WIDE POWER SUPPLY RANGE :
  SINGLE SUPPLY : +3V TO +30V
  DUAL SUPPLIES : ±1.5V TO ±15V



N
DIP14
(Plastic Package)



D
SO14
(Plastic Micropackage)

P
TSSOP14
(Thin Shrink Small Outline Package)

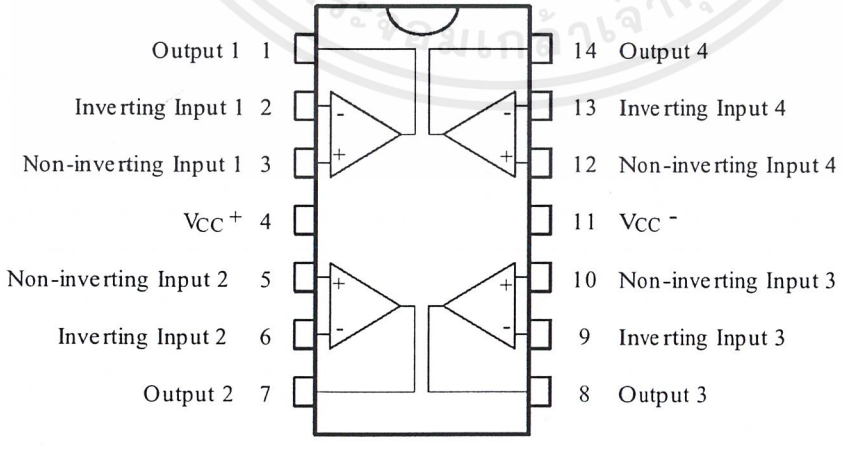
DESCRIPTION

These circuits consist of four independent, high gain, internally frequency compensated operational amplifiers. They operate from a single power supply over a wide range of voltages. Operation from split power supplies is also possible and the low power supply current drain is independent of the magnitude of the power supply voltage.

ORDER CODES

Part Number	Temperature Range	Package		
		N	D	P
LM124	-55°C, +125°C	•	•	•
LM224	-40°C, +105°C	•	•	•
LM324	0°C, +70°C	•	•	•
Example : LM224N				

PIN CONNECTIONS (top view)

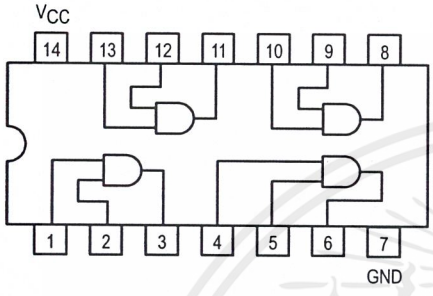


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

MOTOROLA

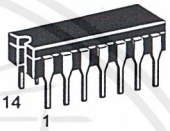
QUAD 2-INPUT AND GATE

SN54/74LS08

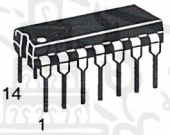


QUAD 2-INPUT AND GATE

LOW POWER SCHOTTKY



J SUFFIX
CERAMIC
CASE 632-08



N SUFFIX
PLASTIC
CASE 646-06

D SUFFIX
SOIC
CASE 751A-02

ORDERING INFORMATION

SN54LSXXJ	Ceramic
SN74LSXXN	Plastic
SN74LSXXD	SOIC

GUARANTEED OPERATING RANGES

Symbol	Parameter		Min	Typ	Max	Unit
V_{CC}	Supply Voltage	54 74	4.5 4.75	5.0 5.0	5.5 5.25	V
T_A	Operating Ambient Temperature Range	54 74	−55 0	25 25	125 70	°C
I_{OH}	Output Current — High	54, 74			−0.4	mA
I_{OL}	Output Current — Low	54 74			4.0 8.0	mA

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

การจัดทำปริญญานิพนธ์นี้สำเร็จลุล่วงได้ด้วยดี โดยได้รับการช่วยเหลือจากบุคคลหลายๆท่าน คณะผู้จัดทำโครงงานรู้สึกซาบซึ้งและขอกราบขอบพระคุณอาจารย์ถาวร เบญจนราสุทธิ์

ขอขอบพระคุณภาควิชาระบบควบคุม ที่เอื้อเฟื้อ อุปกรณ์และเครื่องมือทั้งหมด ตลอดจนสถานที่ๆใช้ในการทำงาน

ขอขอบพระคุณภาคเทคโนโลยีการวัดคุมทางอุตสาหกรรม ที่เอื้อเฟื้อสถานที่ๆใช้ในการทำงาน

ขอขอบพระคุณหอถาวรพฤกษ์ ห้อง 234 ที่ให้ที่พักพิง

ขอขอบพระคุณอาจารย์ทุกท่าน ที่อบรมให้ความรู้แก่คณะผู้จัดทำ จนกระทั่งคณะผู้จัดทำสำเร็จการศึกษา

ขอขอบพระคุณเพื่อนๆทุกคน (เพื่อนๆเด็กเทพ, เพื่อนๆทุกภาค) ซึ่งเป็นที่ปรึกษา เป็นกำลังใจที่ดี และเป็นเรี่ยวแรงที่สำคัญในการทำโครงงานครั้งนี้

และกราบขอขอบพระคุณ คุณพ่อ คุณแม่ ที่เคารพรักยิ่ง ที่ให้การสนับสนุน เป็นกำลังใจที่ยิ่งใหญ่ และกำลังทุนทรัพย์ที่สำคัญตลอดมา

สุดท้ายนี้ หากปริญญานิพนธ์ฉบับนี้มีข้อผิดพลาดและข้อบกพร่องประการใด คณะผู้จัดทำขออภัยไว้ ณ ที่นี้ด้วย

คณะผู้จัดทำ

มีนาคม 2544

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หนังสืออ้างอิง

1. สุนทร วิทูสุรพจน์, " การโปรแกรมภาษาแอสแซมบลีของไมโครคอนโทรลเลอร์ตระกูล 8051 ", บริษัทซีเอ็ดยูเคชั่น จำกัด,198 หน้า, 2530
2. ธีรวัฒน์ ประกอบผล, " การประยุกต์ใช้งานไมโครคอนโทรลเลอร์ ", สมาคมส่งเสริมเทคโนโลยี (ไทย – ญี่ปุ่น), 235 หน้า, 2537
3. ดร. ยุทธนา กุลวิทิต, " ระบบติดตามดวงอาทิตย์สำหรับแผงเซลล์แสงอาทิตย์ ", คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย, 2526

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้