

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

เครื่องควบคุมตำแหน่งแผงโฟโตโวลตาอิก



นายณัฐศัย เมฆจำเริญ
นายสรายทธร เหลืองเรืองโรจน์

โครงการพิเศษนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิทยาศาสตรบัณฑิต

ภาควิชาฟิสิกส์ประยุกต์

คณะวิทยาศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

๒๖๗.

๐๒๓๖๘ ค

๑๕๔๗

ปีการศึกษา ๒๕๔๗

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

เลขที่..... 58697

เลขทะเบียน..... 31 ส.ค. 2549

วันที่.....

๓๑ ส.ค. ๒๕๔๗
b.....
i.....

Photovoltaic Frame Positioning Controller



Mr. Nuttasak Mekchunroen

Mr. Sarayut Luengruengroj

A special Project Submitted in Partial Fulfillment of the Requirement for the Degree of Bachelor
of science

Department of Applied Physics

Faculty of Science





King Mongkut 's Institute of Technology Ladkrabang

Academic Year 2004

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โครงการพิเศษเรื่อง เครื่องควบคุมตำแหน่งแผงโฟโตโวลตาอิก
นักศึกษา นายณัฐศักดิ์ เมฆจำเริญ
นายสรายุทธ เหลืองเรืองโรจน์
ภาควิชา ฟิสิกส์ประยุกต์
สาขาวิชา ฟิสิกส์ประยุกต์
อาจารย์ที่ปรึกษา รศ.วิจิต ศิริโชติ

ภาควิชาฟิสิกส์ประยุกต์ คณะวิทยาศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
อนุมัติให้โครงการพิเศษนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิทยาศาสตรบัณฑิต

คณะกรรมการตรวจสอบ	ลายมือชื่อ
ประธานกรรมการ อ.สุรชาติ กมลดีดก	
กรรมการ ผศ.สาทราย คุณิพงษ์	
กรรมการ อ.เบญจพล ต้นฮู้	
กรรมการที่ปรึกษา รศ.วิจิต ศิริโชติ	



(รองศาสตราจารย์ วิชาญ เตชิตธีระ)

หัวหน้าภาควิชาฟิสิกส์ประยุกต์

ลิขสิทธิ์ของภาควิชาฟิสิกส์ประยุกต์ คณะวิทยาศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โครงการพิเศษ เครื่องควบคุมตำแหน่งแผงโฟโตโวลตาอิก

นักศึกษา นายณัฐศักดิ์ เมฆจำเริญ
 นายสรายุทธ เหลืองเรืองโรจน์
 ภาควิชา ฟิสิกส์ประยุกต์ คณะวิทยาศาสตร์
 สาขาวิชา ฟิสิกส์ประยุกต์
 ปีการศึกษา 2547
 อาจารย์ที่ปรึกษา รศ. วิชิต ศิริโชติ

บทคัดย่อ

เครื่องควบคุมตำแหน่งแผงโฟโตโวลตาอิกในทิศเหนือ-ใต้ได้รับการพัฒนาขึ้น โดยใช้บอร์ดไมโครคอนโทรลเลอร์ MCS51 มุม declination ถูกคำนวณ โดยใช้เวลาที่องศาและตำแหน่งละติจูดของเขตภาคกระบัง มุมจะมีค่าเปลี่ยนแปลงอยู่ในช่วง ± 23.5 องศา ตลอดทั้งปี ผลจากการทดลองในเดือนตุลาคม 2548 แสดงให้เห็นว่า การปรับมุมที่ได้จากการทดลองมีค่าความผิดพลาด ± 2 องศา จากค่ามุม declination ที่คำนวณได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Special Project Title	Photovoltaic Frame positioning controller
Name	Mr. Nuttasak Mekchunroen Mr. Sarayut Luengruengroj
Department	Applied Physics
Program	Applied Physics
Academic Year	2004
Special Project Advisor	Assoc.Prof. Wichit Sirichote



ABSTRACT

A controller for adjusting North-South tilt angle of photovoltaic frame has been developed. The controller was built with MCS51 microcontroller board. Declination angle was computed using local time and was corrected for Ladkrabang's latitude. The tilt angle spans ± 23.5 degrees along the period of one year. The test on October 2005 has shown frame's angle was deviated by ± 2 degree from declination angle.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

ในการทำปฏิญานิพนธ์ฉบับนี้ต้องขอขอบพระคุณ รศ.วิจิต ศิริโชติ อาจารย์ประจำ
ภาควิชาฟิสิกส์ประยุกต์ คณะวิทยาศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหาร
ลาดกระบัง อาจารย์ที่ปรึกษาปฏิญานิพนธ์ที่ได้ให้คำปรึกษาและแนะแนวทางในการดำเนินการ
รวมทั้งสนับสนุนเรื่องทุนทรัพย์ที่ใช้ในการดำเนินการทำปฏิญานิพนธ์นี้ ตลอดจนถึงบิดา-มารดา
และเพื่อนๆ ในภาคที่คอยช่วยเหลือและให้กำลังใจมาโดยตลอดจนทำให้ปฏิญานิพนธ์ฉบับนี้สำเร็จ
ลงได้ด้วยดี



นายณัฐศักดิ์ เมฆจำเริญ
นายสรายุทธ เหลืองเรืองโรจน์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

	หน้า
บทคัดย่อภาษาไทย	ก
บทคัดย่อภาษาอังกฤษ	ข
กิตติกรรมประกาศ	ค
สารบัญ	ง
สารบัญตาราง	ฉ
สารบัญรูป	ช
บทที่ 1 บทนำ	1
1.1 ความเป็นมาของโครงการพิเศษ/ปัญหาพิเศษ	1
1.2 วัตถุประสงค์	1
1.3 ขอบเขตของการวิจัย	1
1.4 ขั้นตอนการวิจัยและวิธีการดำเนินงาน	2
1.5 ผลที่คาดว่าจะได้รับ	2
บทที่ 2 ทฤษฎีและหลักการ	3
2.1 เซลล์แสงอาทิตย์	3
2.1.1 ชนิดของเซลล์แสงอาทิตย์	3
2.1.2 หลักการทำงานของเซลล์แสงอาทิตย์	4
2.2 ชุดปรับมุมของแผงเซลล์แสงอาทิตย์	5
2.3 มอเตอร์กระแสสลับ	7
2.4 ส่วนของวงจรที่ใช้ควบคุมมอเตอร์	13
2.5 Solar Declination	14
2.6 การวัดปริมาณกระแสไฟฟ้าที่แผงเซลล์แสงอาทิตย์ผลิตได้ในแต่ละวัน	17
บทที่ 3 วิธีดำเนินการวิจัย	18
3.1 ส่วนของการติดตั้งอุปกรณ์	18
3.1.1 การติดตั้งมอเตอร์	18
3.1.2 การเชื่อมต่อวงจรที่ใช้ในการควบคุมมอเตอร์	19
3.2 ส่วนของการคำนวณค่ามุมตกกระทบของแนวรังสีของแสงอาทิตย์	20
3.3 ส่วนของการเขียนโปรแกรม	22
บทที่ 4 ผลการทดลองและการอภิปรายผล	24
4.1 ผลจากชุดปรับมุมของแผงเซลล์แสงอาทิตย์	24

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

	หน้า
4.2 สรุปผลการทดลอง	27
บทที่ 5 สรุปและข้อเสนอแนะ	28
5.1 สรุปผลการทดลอง	28
5.2 ปัญหาที่พบและข้อเสนอแนะ	28
บรรณานุกรม	29
ภาคผนวก	30



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

	หน้า
ตารางที่ 3.1 แสดงค่ามุม Declination ในแต่ละวันตลอดทั้งปีของประเทศไทย	20
ตารางที่ 4.1 แสดงผลการทดลองที่ได้ตลอดเดือนตุลาคม	26



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป

	หน้า
รูปที่ 2.1 เซลล์แสงอาทิตย์	3
รูปที่ 2.2 แสดงถึงหลักการทำงานของเซลล์แสงอาทิตย์	4
รูปที่ 2.3 ชุดปรับมุมของแผงเซลล์แสงอาทิตย์	5
รูปที่ 2.4 กลไกที่ใช้ในการปรับมุมของชุดปรับมุมของแผงเซลล์แสงอาทิตย์	5
รูปที่ 2.5 แสดงถึงกลไกการที่ใช้ในการปรับมุมของชุดปรับมุมของแผงเซลล์แสงอาทิตย์	6
รูปที่ 2.6 อุปกรณ์ทางไฟฟ้าพื้นฐานของ มอเตอร์กระแสสลับ	7
รูปที่ 2.7 สนามแม่เหล็กที่หมุนไปของมอเตอร์กระแสสลับ	8
รูปที่ 2.8 ลักษณะของแต่ละเฟสของไฟฟ้า 3 เฟส	9
รูปที่ 2.9 วิธีการเชื่อมต่อกระแสไฟฟ้า 3 เฟสเข้ากับ 6 ขั้วของสเตเตอร์	10
รูปที่ 2.10 แสดงว่ากระแสไฟฟ้า 3 เฟสทำให้เกิดการหมุนของสนามแม่เหล็กได้อย่างไร	10
รูปที่ 2.11 โครงสร้างของโรเตอร์ของอินดักชันมอเตอร์กระแสสลับ	11
รูปที่ 2.12 แสดงการเหนี่ยวนำของศักย์ไฟฟ้าในโรเตอร์ก่อให้เกิดกระแสไหลในตัวนำโรเตอร์	12
รูปที่ 2.13 วงจรที่ใช้เชื่อมต่อกับตู้ควบคุมมอเตอร์เพื่อสามารถใช้งานได้แบบกึ่งอัตโนมัติ	13
รูปที่ 2.14 ตู้ควบคุมมอเตอร์	13
รูปที่ 2.15 แสดงการเกิดฤดูกาลของโลก	14
รูปที่ 2.16 แสดงค่ามุม Declination ที่เส้นศูนย์สูตร	14
รูปที่ 2.17 แสดงค่า Solar Declination ของแต่ละวันตลอดทั้งปี	15
รูปที่ 2.18 แสดงรูปวงกลมหนึ่งหน่วย	16
รูปที่ 2.19 แสดงบอร์ดที่ทำการวัดกระแสไฟฟ้าที่ผลิตได้	17
รูปที่ 3.1 แสดงชุดปรับมุมของแผงเซลล์แสงอาทิตย์	18
รูปที่ 3.2 แสดงตำแหน่งที่รีเลย์เชื่อมต่อเข้ากับวงจรของตู้ควบคุมมอเตอร์	19
รูปที่ 3.3 แสดงวงจรรีเลย์ที่เชื่อมต่อกับบอร์ดไมโครคอนโทรลเลอร์	20
รูปที่ 3.4 โพลีวัชาร์ทแสดงการทำงานของโปรแกรมที่ใช้ควบคุมชุดปรับมุมของแผงเซลล์แสงอาทิตย์	22
รูปที่ 3.5 แสดงวงจรของ DS 1307	23
รูปที่ 4.1 แผงเซลล์แสงอาทิตย์ที่ควบคุมด้วยชุดปรับมุม	24
รูปที่ 4.2 แสดงมอเตอร์ที่ควบคุมด้วยบอร์ดไมโครคอนโทรลเลอร์	25
รูปที่ 4.3 แสดงตัววัดมุมที่ติดตั้งไว้	25

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้ในการเรียนการสอนเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1

บทนำ

1.1 ความเป็นมาของโครงการพิเศษ/ปัญหาพิเศษ

ปัจจุบันพลังงานไฟฟ้าเป็นสิ่งจำเป็นในการดำเนินชีวิต ทุกกิจกรรมในชีวิตประจำวันย่อมต้องมีไฟฟ้าเข้ามาเกี่ยวข้องเสียเป็นส่วนใหญ่ อีกทั้งความเจริญได้ขยายออกไปยังพื้นที่ห่างไกลทำให้มีไฟฟ้าใช้กันอย่างทั่วถึง ดังนั้นความต้องการพลังงานไฟฟ้าจึงเพิ่มมากขึ้นเป็นผลให้ต้องเพิ่มปริมาณการผลิตไฟฟ้าตามไปด้วย

การผลิตกระแสไฟฟ้าในปัจจุบันส่วนใหญ่นั้นมาจากการใช้ทรัพยากรธรรมชาติ เช่น น้ำมัน ถ่านหิน ก๊าซธรรมชาติ มาเป็นเชื้อเพลิงในการผลิตกระแสไฟฟ้าซึ่งทรัพยากรเหล่านี้เป็นสิ่งที่ใช้แล้วหมดไป จึงจำเป็นต้องหาพลังงานอย่างอื่นมาทดแทน เช่น พลังงานน้ำ พลังงานลม พลังงานแสงอาทิตย์ซึ่งเป็นพลังงานที่เกิดขึ้นเองตามธรรมชาติ ซึ่งพลังงานแสงอาทิตย์ก็เป็นอีกทางเลือกหนึ่งที่มีความนิยม โดยสามารถเปลี่ยนพลังงานแสงอาทิตย์เป็นกระแสไฟฟ้าโดยใช้เซลล์แสงอาทิตย์ และยังสามารถผลิตกระแสไฟฟ้ายังพื้นที่ห่างไกลที่แม้ไฟฟ้ายังเข้าไม่ถึงได้

ในการผลิตกระแสไฟฟ้าด้วยเซลล์แสงอาทิตย์ให้ได้ประสิทธิภาพสูงสุดนั้น จำเป็นที่จะต้องคำนึงถึงมุมตกกระทบของแสงที่ตกกระทบบนเซลล์แสงอาทิตย์ด้วย ซึ่งในการผลิตกระแสไฟฟ้าด้วยเซลล์แสงอาทิตย์จะนั้นสามารถผลิตกระแสไฟฟ้าได้มากที่สุดเมื่อแสงอาทิตย์ตกกระทบเป็นมุมตั้งฉากกับเซลล์แสงอาทิตย์ แต่ในความเป็นจริงแล้วเนื่องจากการที่แกนโลกเอียงและดวงอาทิตย์มีการโคจรตลอดเวลา ทำให้แสงที่ตกกระทบบนพื้นโลกมีมุมเปลี่ยนแปลงตลอดเวลา ดังนั้นในงานวิจัยนี้จึงได้นำเอาแนวคิดนี้มาสร้างเป็นระบบควบคุมตำแหน่งมุมของเซลล์แสงอาทิตย์เพื่อให้เซลล์แสงอาทิตย์สามารถผลิตกระแสไฟฟ้าได้มากที่สุด

1.2 วัตถุประสงค์

- เพื่อสร้างชุดควบคุมตำแหน่งมุมของแผงโฟโตโวลตาอิก

1.3 ขอบเขตของการวิจัย

ในการวิจัยนี้มีเป้าหมายเพื่อสร้างโมดูลที่มีหน้าที่สำหรับการปรับตำแหน่งมุมเอียงของแผงเซลล์แสงอาทิตย์แนวทิศทางทางเหนือ – ใต้ในแต่ละวัน เพื่อที่จะสามารถรับแสงอาทิตย์ได้ในทิศทางตั้งฉากกับเซลล์แสงอาทิตย์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.4 ขั้นตอนการวิจัยและวิธีการดำเนินงาน

1.4.1 ศึกษาเซลล์แสงอาทิตย์

1.4.2 ศึกษาและสร้างชุดควบคุมมอเตอร์แบบกึ่งอัตโนมัติและทำการติดตั้งกับตู้ควบคุมมอเตอร์

1.4.3 ทำการติดตั้งมอเตอร์เข้ากับส่วนกลไกของการปรับแผงเซลล์แสงอาทิตย์

1.4.4 ศึกษาการคำนวณหาค่า Solar declination

1.4.5 ทำการเขียน โปรแกรมเพื่อคำนวณค่าและควบคุมชุดมอเตอร์ในการปรับแผงเซลล์แสงอาทิตย์

1.4.6 ทดลองและวิเคราะห์ผลที่ได้จากการทดลอง

1.4.7 สรุปผลของการวิจัยและแนวทางในการพัฒนา

1.5 ผลที่คาดว่าจะได้รับ

สามารถสร้างชุดควบคุมตำแหน่งมุมของเซลล์แสงอาทิตย์ตามคำนวณที่ได้จากการคำนวณในแต่ละวัน

บทที่ 2

ทฤษฎีและหลักการ

2.1 เซลล์แสงอาทิตย์



รูปที่ 2.1 เซลล์แสงอาทิตย์

2.1.1 ชนิดของเซลล์แสงอาทิตย์

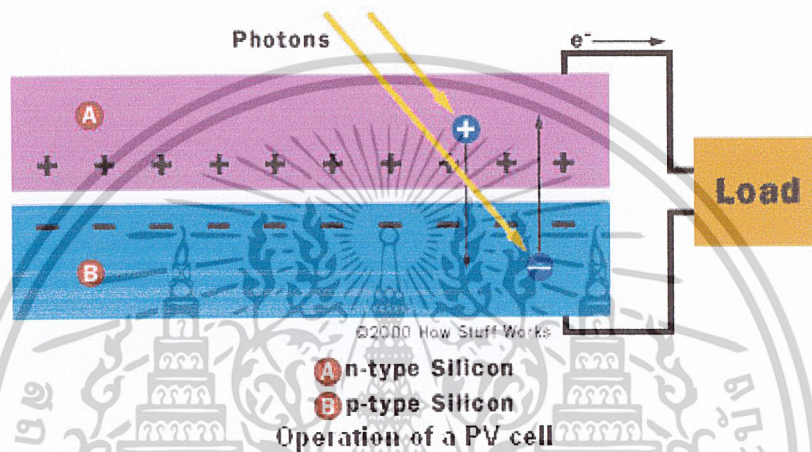
เซลล์แสงอาทิตย์แบ่งได้เป็น 3 ชนิดใหญ่ๆตามลักษณะ crystal structure ได้แก่

1. Monocrystalline สร้างโดยนำซิลิกอนที่หลอมมาตัดให้เป็นแผ่นบาง ซึ่งเซลล์แสงอาทิตย์ชนิดนี้จะให้ประสิทธิภาพที่สูง
2. Polycrystalline สร้างโดยเทซิลิกอนเหลวลงในบล็อกขึ้นรูปให้เป็นแผ่น ระหว่างการแข็งตัว crystal structure จะก่อตัวขึ้นมา แต่ที่ขอบจะเกิดความบกพร่องขึ้น ซึ่งผลจากความบกพร่องนี้จะทำให้ประสิทธิภาพลดลง แต่ราคาจะน้อยกว่าชนิดแรก
3. Amorphus(thin layer cell) สร้างโดยการปลูกซิลิกอนลงบนฐานรอง เช่น แก้วเป็นชั้น ซึ่ง layer จะมีความหนาน้อยกว่า $1 \mu\text{m}$ ซึ่งทำให้เซลล์แสงอาทิตย์ชนิดนี้มีราคาถูกเนื่องจากใช้วัสดุน้อย แต่ก็ให้ประสิทธิภาพน้อยกว่าอีก 2 ชนิดมาก ซึ่งทำให้มันถูกนำไปใช้ในอุปกรณ์ที่ต้องการพลังงานไม่มากเช่น นาฬิกา เครื่องคิดเลขขนาดเล็ก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.2 หลักการทำงานของเซลล์แสงอาทิตย์

สารกึ่งตัวนำที่เป็น p-n junction เมื่อแสงตกกระทบ พลังงานโฟตอนของแสงจะถูกดูดกลืนโดยสารกึ่งตัวนำซึ่งถ้าโฟตอนมีพลังงานเพียงพอจะทำให้เกิดคู่อิเล็กตรอน-โฮลอิสระขึ้น ซึ่งอิเล็กตรอนอิสระจะถูกเร่งด้วยสนามไฟฟ้าของเซลล์ให้ข้ามผ่าน p-n junction ไปที่ขั้วไฟฟ้าซึ่งจะทำให้เกิดกระแสไฟฟ้าขึ้น

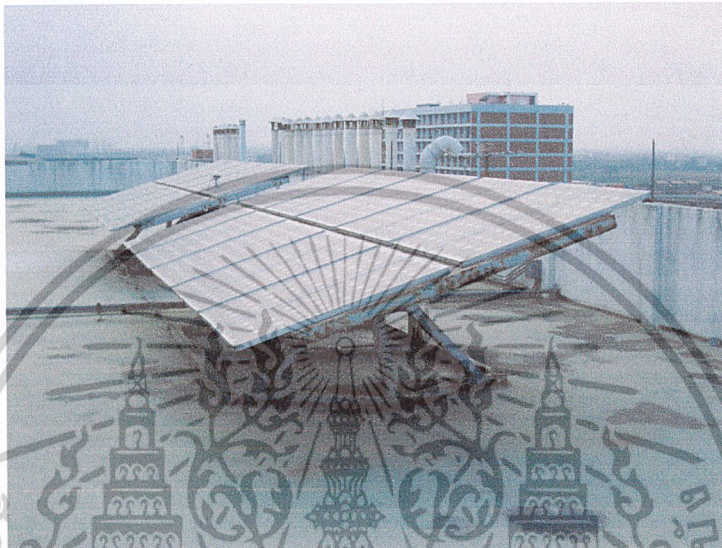


รูปที่ 2.2 แสดงถึงหลักการทำงานของเซลล์แสงอาทิตย์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2 ชุดปรับมุมของแผงเซลล์แสงอาทิตย์

ในการปรับมุมของแผงเซลล์แสงอาทิตย์นั้น เราจำเป็นต้องอาศัยกลไกอย่างอื่นเพื่อเข้ามาช่วยในการปรับมุมของเซลล์แสงอาทิตย์ ซึ่งในที่นี้เราจะใช้ชุดปรับมุมของแผงเซลล์แสงอาทิตย์ เพื่อใช้ในการปรับเปลี่ยนมุม



รูปที่ 2.3 ชุดปรับมุมของแผงเซลล์แสงอาทิตย์

หลักการทำงานของชุดปรับมุมของแผงเซลล์แสงอาทิตย์ คือ การใช้มอเตอร์ในการหมุนเฟืองที่เชื่อมต่อกับแท่งเหล็กที่เป็นเกลิยว ซึ่งเมื่อหมุนจะทำให้แขนที่จับอยู่กับเกลิยวสามารถเคลื่อนที่เข้าออกทำให้เราปรับมุมแผงเซลล์แสงอาทิตย์ได้



รูปที่ 2.4 กลไกที่ใช้ในการปรับมุมของชุดปรับมุมของแผงเซลล์แสงอาทิตย์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยที่เราหมุนเฟืองไปด้านหนึ่งซึ่งให้แขนเลื่อนเข้าจะทำให้แผงเซลล์แสงอาทิตย์ที่ติดอยู่กับแขนก็จะเลื่อนลง แต่ถ้าเราให้เฟืองหมุนไปอีกด้านหนึ่งจะทำให้แขนเลื่อนออกและแผงเซลล์แสงอาทิตย์ที่ติดอยู่ที่แขนจะเลื่อนขึ้น



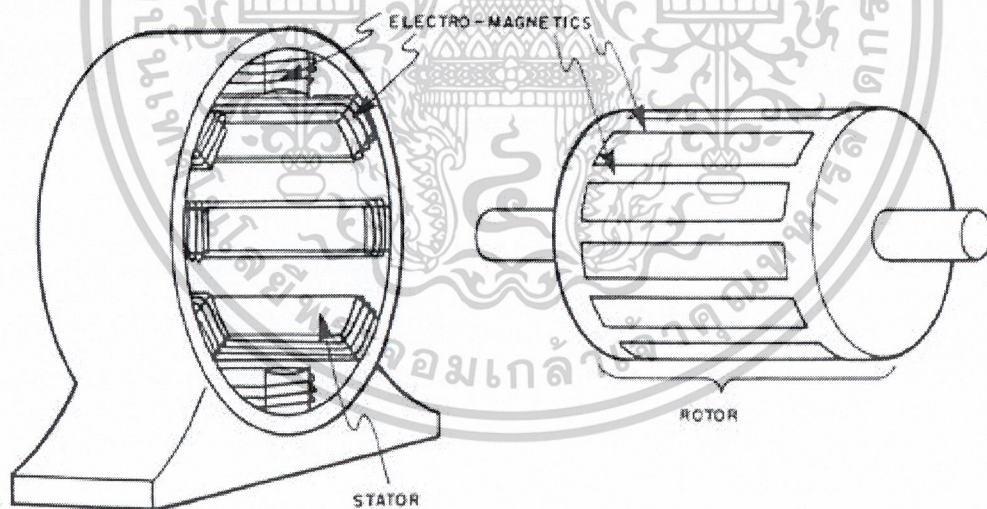
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3 มอเตอร์กระแสสลับ

มอเตอร์กระแสสลับใช้ในการหมุนเฟืองที่เชื่อมต่อกับแท่งเหล็กที่เป็นเกลียวเพื่อปรับมุมของแผงเซลล์แสงอาทิตย์

พื้นฐานการทำงานของมอเตอร์กระแสสลับ

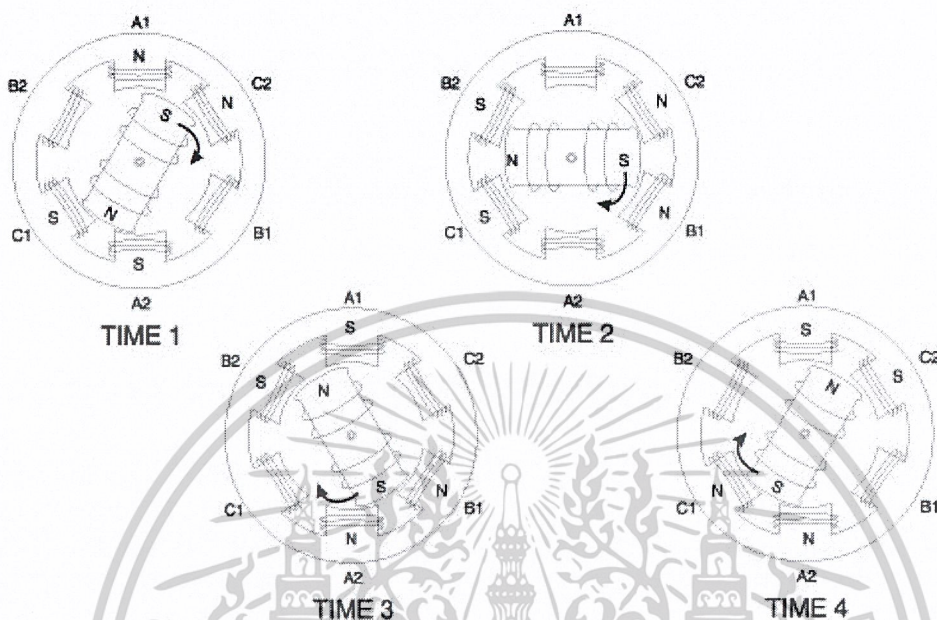
มอเตอร์กระแสสลับมีชิ้นส่วนทางไฟฟ้า 2 ส่วน คือ ส่วน สเตเตอร์ และส่วน โรเตอร์ ดังแสดงในรูป ส่วนสเตเตอร์เป็นส่วนที่อยู่ในส่วนที่อยู่หนึ่งของมอเตอร์ ซึ่งจะประกอบไปด้วยแม่เหล็กไฟฟ้าเรียงอยู่ภายในโดยรอบทรงกระบอกโดยหันขั้วเข้าสู่ศูนย์กลาง ส่วน โรเตอร์เป็นส่วนที่หมุนของมอเตอร์ ซึ่งจะประกอบไปด้วยแม่เหล็กไฟฟ้าเรียงอยู่โดยรอบทรงกระบอก โดยหันขั้วเข้าหาขั้วของสเตเตอร์ โรเตอร์จะอยู่ในของสเตเตอร์และเชื่อมต่อกับแกนของมอเตอร์ จุดประสงค์ของอุปกรณ์มอเตอร์นี้ก็คือ ทำให้โรเตอร์หมุนซึ่งจะทำให้แกนของมอเตอร์หมุนไปด้วย การหมุนของมอเตอร์นี้เกิดจากปรากฏการณ์แม่เหล็กที่ขั้วแม่เหล็กเดียวกันกันจะผลักกัน ในขณะที่ขั้วตรงข้ามจะดึงดูดกัน ถ้าเราเปลี่ยนขั้วของสเตเตอร์ตามลำดับในทิศทางที่สนามแม่เหล็กรวมมีทิศทางหมุนไป โรเตอร์จะเคลื่อนที่ตาม และหมุนด้วยสนามแม่เหล็กของสเตเตอร์



รูปที่ 2.6 อุปกรณ์ทางไฟฟ้าพื้นฐานของ มอเตอร์ไฟฟ้ากระแสสลับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สนามแม่เหล็กที่หมุนของสเตเตอร์อธิบายได้ดังรูป

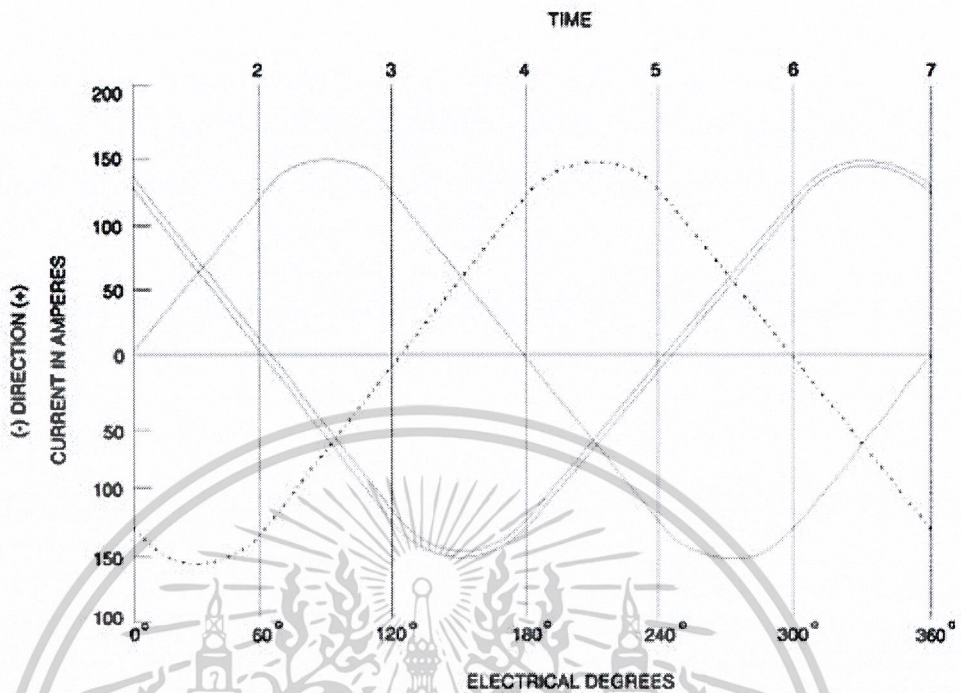


รูปที่ 2.7 สนามแม่เหล็กที่หมุนไปของมอเตอร์กระแสลับ

สเตเตอร์มีขั้วแม่เหล็ก 6 ขั้วและ โรเตอร์มี 2 ขั้ว ที่ Time1 ขั้วสเตเตอร์ A-1และC-2 เป็นขั้วเหนือ ในขณะที่ขั้วตรงข้าม A-2และC-1 เป็นขั้วใต้ S-poleของโรเตอร์จะถูกดึงดูดโดยขั้ว N-pole 2 ขั้วของสเตเตอร์ และขั้ว N-poleของโรเตอร์จะถูกดึงดูดโดยขั้ว S-pole 2ขั้วของสเตเตอร์ ที่ Time2 สภาพขั้วของขั้วของสเตเตอร์จะเปลี่ยนไป C-2และB-1 เป็น N-pole ในขณะที่ C-1และB-2 เป็น S-pole โรเตอร์จะถูกบังคับให้หมุนไป 60 องศาตั้งแฉกกับขั้วสเตเตอร์ดังรูป ที่ Time3 B-1และA-2 เป็น N-pole ที่ Time4 A-2และC-1 เป็น N-pole เมื่อมีการเปลี่ยนแปลงแต่ละครั้งเกิดขึ้น ขั้วของโรเตอร์จะถูกดึงดูดโดยขั้วตรงข้ามของสเตเตอร์ ดังนั้นเมื่อสนามแม่เหล็กของสเตเตอร์หมุนไป โรเตอร์จะถูกบังคับให้หมุนโดยมัน

วิธีหนึ่งที่จะสร้างการหมุนของสนามแม่เหล็กในสเตเตอร์ของมอเตอร์กระแสลับ คือใช้ พาวเวอร์ซัพพลาย 3เฟส ในการจ่ายไฟให้กับขดลวดสเตเตอร์ ซึ่งโดยทั่วไปแล้วสำหรับ แหล่งจ่ายไฟกระแสสลับชนิดเฟสเดียวจะผลิตกระแส โดยกระแสจะมีค่าเป็นบวกช่วงเวลาหนึ่งและเป็นลบช่วงเวลาหนึ่งสลับไปมา ซึ่ง ณ เวลาหนึ่งกระแสจะมีค่าเป็นบวกหรือลบค่าใดค่าหนึ่งเท่านั้น แต่สำหรับแหล่งจ่ายไฟบางตัวจะผลิตกระแสที่ต่างกัน 3เส้น(เฟส) อย่างไรก็ตาม ซีกของกระแสและความเข้มของกระแสแต่ละเฟสจะไม่เท่ากัน ดังรูป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

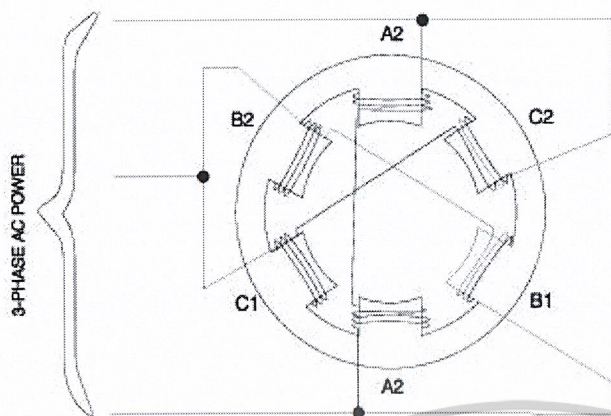


รูปที่ 2.8 ลักษณะของแต่ละเฟสของไฟฟ้า 3 เฟส

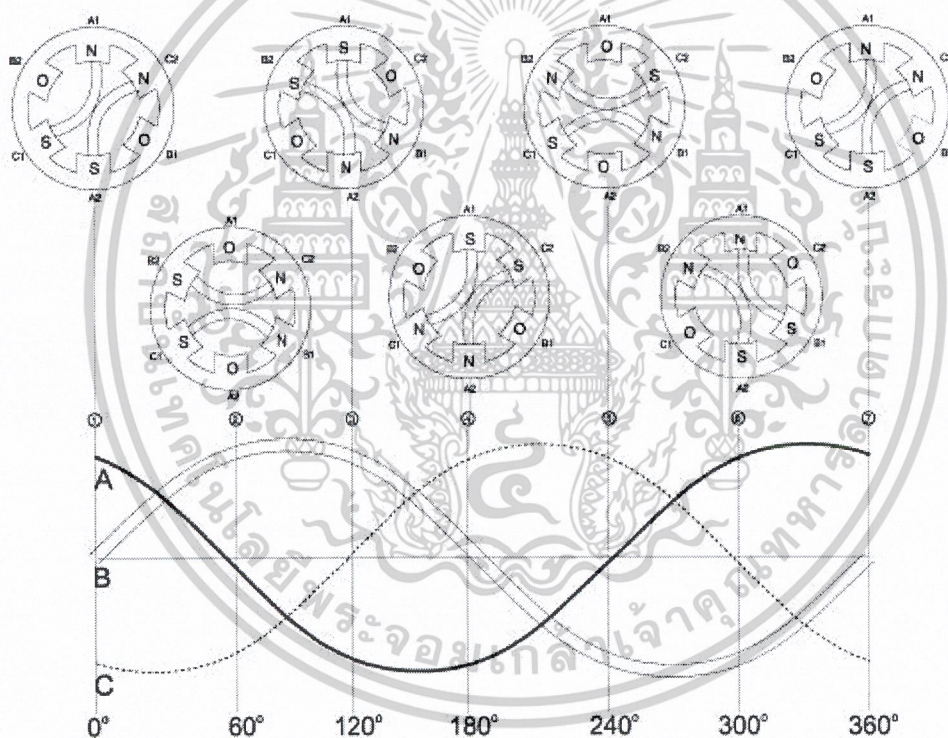
ให้กระแส 3 เฟสเป็น เฟส A, B, C ที่ Time1 เฟส A จะมีค่าเป็นศูนย์แอมป์ เฟส B จะมีค่าเพิ่มขึ้นเข้าใกล้ค่าสูงสุดและอยู่ในซีกลบ เฟส C จะมีค่าใกล้ค่าสูงสุดแต่กำลังลดลงและจะอยู่ในซีกบวก ที่ Time2 ค่าของเฟส A จะเพิ่มขึ้นและอยู่ในซีกบวก ค่าของเฟส B จะลดลงและยังคงอยู่ในซีกลบ ค่าของเฟส C จะลดลงจนเป็นศูนย์ กระแสที่สมบูรณ์ (ศูนย์ไปยังค่าสูงสุดในซีกหนึ่งกลับไปยังศูนย์ไปยังค่าสูงสุดในอีกทิศหนึ่งและกลับมายังศูนย์) แหล่งกำเนิดกระแสจะต้องหมุนขดลวดหนึ่งรอบ ดังนั้น กระแสที่สมบูรณ์จะมี 360 องศา จากรูปจะเห็นว่าแต่ละเฟสจะต่างจากเฟสอื่น 120 องศา

เพื่อสร้างการหมุนของสนามแม่เหล็กของสเตเตอร์ของมอเตอร์กระแสสลับ 3 เฟส การเชื่อมต่อขั้ว 6 ขั้วของสเตเตอร์แสดงดังรูป แต่ละเฟสของพาวเวอร์ซัพพลาย 3 เฟส จะต่อกับขั้วตรงข้ามและขดลวดจะพันในทิศทางเดียวกัน ซึ่งสภาพขั้วของแม่เหล็กไฟฟ้าจะขึ้นอยู่กับทิศของกระแสที่ไหลผ่านขดลวด ดังนั้นถ้าหากขั้วตรงข้ามของแม่เหล็กไฟฟ้ามีการพันขดลวดในทิศทางเดียวกัน สภาพขั้วของขั้วของสเตเตอร์ที่หันเข้าหากันจะตรงข้ามกัน ดังนั้นเมื่อขั้ว A_1 เป็น N-pole A_2 จะเป็น S-pole เมื่อ B_1 เป็น N-pole B_2 จะเป็น S-pole

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.9 วิธีการเชื่อมต่อกระแสไฟ 3 เฟสเข้ากับ 6 ขั้วของสเตเตอร์

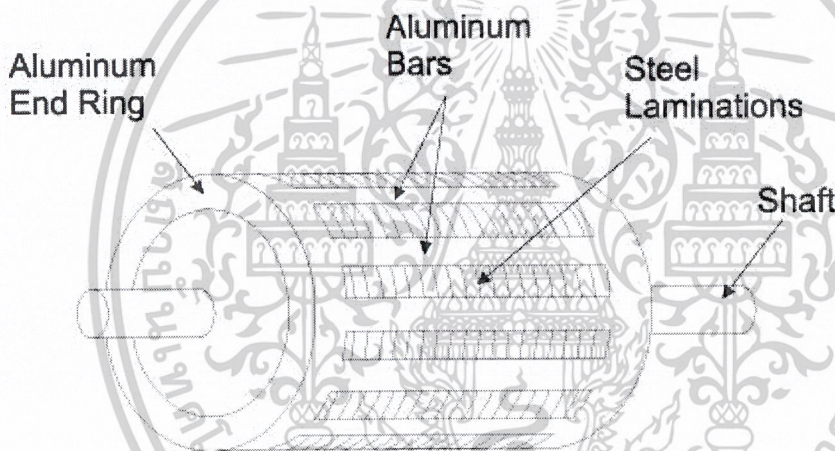


รูปที่ 2.10 แสดงว่ากระแสไฟ 3 เฟสทำให้เกิดการหมุนของสนามแม่เหล็กได้อย่างไร

จากรูป แสดงว่าสนามแม่เหล็กเกิดการหมุนได้อย่างไร ที่ Time1 กระแสที่ไหลในขั้วเฟส A เป็นบวก และขั้ว A-1 จะเป็น N กระแสที่ไหลในขั้วเฟส C เป็นลบทำให้ C-2 เป็น N และ C-1 เป็น S ไม่มีกระแสไหลในขั้วเฟส B ทำให้ขั้วนี้ไม่มีความเป็นแม่เหล็ก ที่ Time2 เฟสจะเลื่อนไป 60 องศา ทำให้ขั้ว C-2 และ B-1 เป็น N C-1 และ B-2 เป็น S ดังนั้นเมื่อกระแสที่ไหลเลื่อนเฟสไป จะทำให้ขั้ว N และ S เคลื่อนไปในทิศตามเข็มนาฬิการอบๆ สเตเตอร์ ทำให้เกิดการหมุนของสนามแม่เหล็ก โรเตอร์จะ

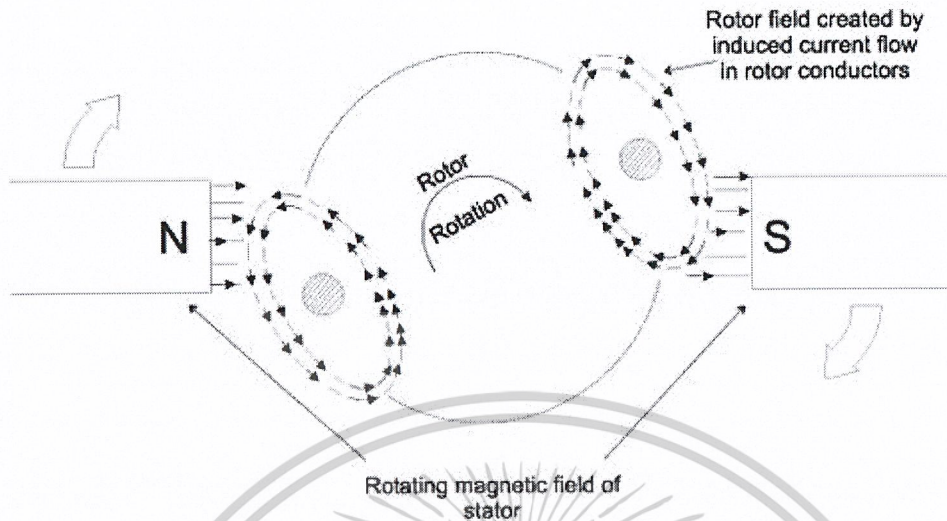
ประพฤติตัวเหมือนเป็นแท่งแม่เหล็ก ซึ่งจะถูกดึงดูดให้หมุนไปตามสนามแม่เหล็กที่หมุนไป เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในส่วนของโรเตอร์ ปกติแล้วจะให้แท่งโรเตอร์พันด้วยขดลวดเหมือนกับสเตเตอร์ และให้กระแสไฟตรงเข้าไปเพื่อทำให้เกิดสนามแม่เหล็กในทิศทางหนึ่ง นี่คือวิธีการทำงานของซิงโครนัสมอเตอร์ แต่ในปัจจุบันมอเตอร์ที่ใช้ในงานอุตสาหกรรมส่วนใหญ่ไม่ใช่ซิงโครนัสมอเตอร์แต่เป็นอินดักชันมอเตอร์ ซึ่งข้อแตกต่างของอินดักชันมอเตอร์กับซิงโครนัสมอเตอร์ที่สำคัญคือ การจ่ายกระแสให้กับโรเตอร์ ซึ่งอินดักชันมอเตอร์ไม่ต้องใช้แหล่งจ่ายไฟภายนอกเหมือนซิงโครนัสมอเตอร์ในการจ่ายไฟให้กับโรเตอร์ โดยอินดักชันมอเตอร์จะใช้วิธีการเหนี่ยวนำแทน การเหนี่ยวนำเป็นหนึ่งในลักษณะเฉพาะของแม่เหล็ก ซึ่งเป็นปรากฏการณ์ธรรมชาติซึ่งจะเกิดขึ้นเมื่อตัวนำ(แท่งอะลูมิเนียมของโรเตอร์ ตามรูป)เคลื่อนที่ตัดผ่านสนามแม่เหล็ก หรือเมื่อสนามแม่เหล็กเคลื่อนที่ผ่านตัวนำซึ่งจะทำให้เกิดกระแสไฟฟ้าไหลในตัวนำ นี่คือการเหนี่ยวนำให้เกิดกระแสไฟฟ้าไหล



รูปที่ 2.11 โครงสร้างของโรเตอร์ของอินดักชันมอเตอร์กระแสสลับ

ในอินดักชันมอเตอร์กระแสที่ไหลใน โรเตอร์ไม่ได้เกิดจากการต่อตัวนำเข้ากับแหล่งกำเนิดไฟฟ้าโดยตรง แต่เกิดจากผลของการที่ตัวนำโรเตอร์ตัดผ่านเส้นฟลักซ์ของสนามแม่เหล็ก ซึ่งสร้างโดยสนามแม่เหล็กของสเตเตอร์ กระแสเหนี่ยวนำที่เกิดขึ้นใน โรเตอร์นี้จะก่อให้เกิดสนามแม่เหล็กรอบๆตัวนำโรเตอร์ดังแสดงในรูป

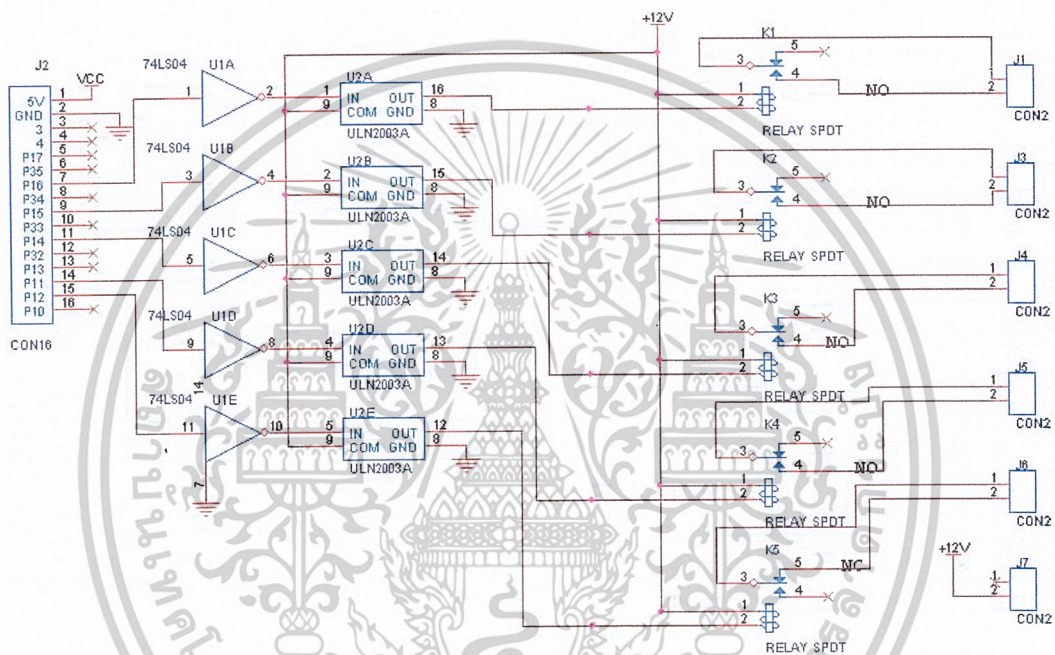


รูปที่ 2.12 แสดงการเหนี่ยวนำของศักย์ไฟฟ้าในโรเตอร์ก่อให้เกิดกระแสไหลในตัวนำโรเตอร์

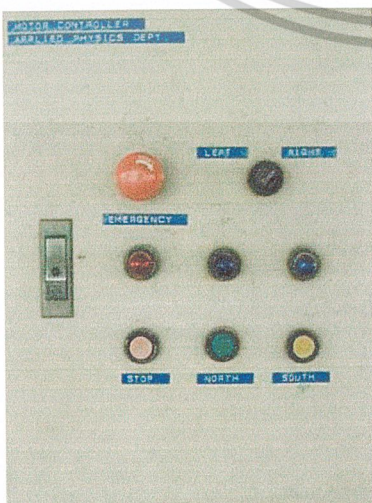
สนามแม่เหล็กรอบๆแต่ละตัวนำโรเตอร์นี้จะทำให้ตัวนำโรเตอร์ประจุติดกันเสมือนเป็นแม่เหล็กถาวร เมื่อสนามแม่เหล็กที่เกิดจากสเตเตอร์หมุนไปเนื่องจากผลของกระแสไฟสลับ 3 เฟส สนามแม่เหล็กเหนี่ยวนำของโรเตอร์จะถูกดึงดูดและหมุนตามไปทำให้โรเตอร์หมุนไปด้วย และเนื่องจากโรเตอร์จะเชื่อมต่อกับแกนของมอเตอร์ เมื่อโรเตอร์หมุนจะทำให้แกนของมอเตอร์หมุนไปด้วย

2.4 ส่วนของวงจรที่ใช้ควบคุมมอเตอร์

สำหรับในส่วนของการควบคุมและสั่งงานมอเตอร์ที่ใช้ในการปรับมุมของเซลล์แสงอาทิตย์ ในที่นี่เราจะใช้เป็นแบบกึ่งอัตโนมัติ (Semi-Automatic) คือ สามารถควบคุมและสั่งการด้วยคน (Manual) หรือ แบบอัตโนมัติ (Automatic) ที่ใช้การเขียนโปรแกรมเข้ามาควบคุม โดยเราจะออกแบบวงจร โดยใช้รีเลย์เพื่อเชื่อมต่อกับตู้ควบคุมมอเตอร์เพื่อทำให้สามารถควบคุมได้แบบกึ่งอัตโนมัติ



รูปที่ 2.13 วงจรที่ใช้เชื่อมต่อกับตู้ควบคุมมอเตอร์เพื่อสามารถใช้งานได้แบบกึ่งอัตโนมัติ

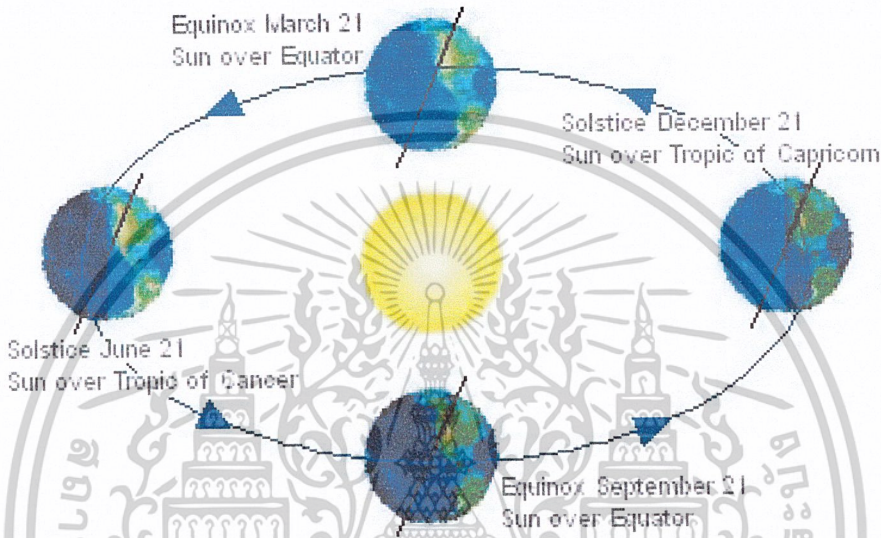


รูปที่ 2.14 ตู้ควบคุมมอเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

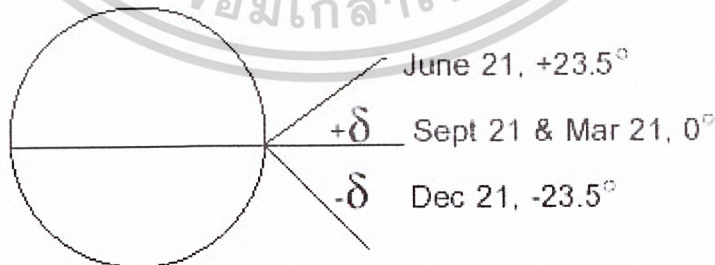
2.5 Solar Declination

ระนาบการหมุนของแกนโลกเอียง 23.5 องศาจากระนาบของวงโคจรของโลกรอบดวงอาทิตย์ โดยจะเอียงแบบนี้ตลอดทั้งปี ทำให้เกิดฤดูกาลขณะที่ซีกโลกเหนือและซีกโลกใต้หันหน้าเข้าและออกดวงอาทิตย์



รูป 2.15 แสดงการเกิดฤดูกาลของโลก

จุดที่กำหนดให้ในวงโคจรของโลก ดวงอาทิตย์จะปรากฏข้างบน โดยตรงที่ Latitude โดยเป็นแนวจาก 23.5 องศาเหนือ ถึง 23.5 องศาใต้ ความลาดเอียงของดวงอาทิตย์ (Declination of the sun) เป็นมุมที่เหมือนกับ Latitude



รูป 2.16 แสดงค่ามุม Declination ที่เส้นศูนย์สูตร

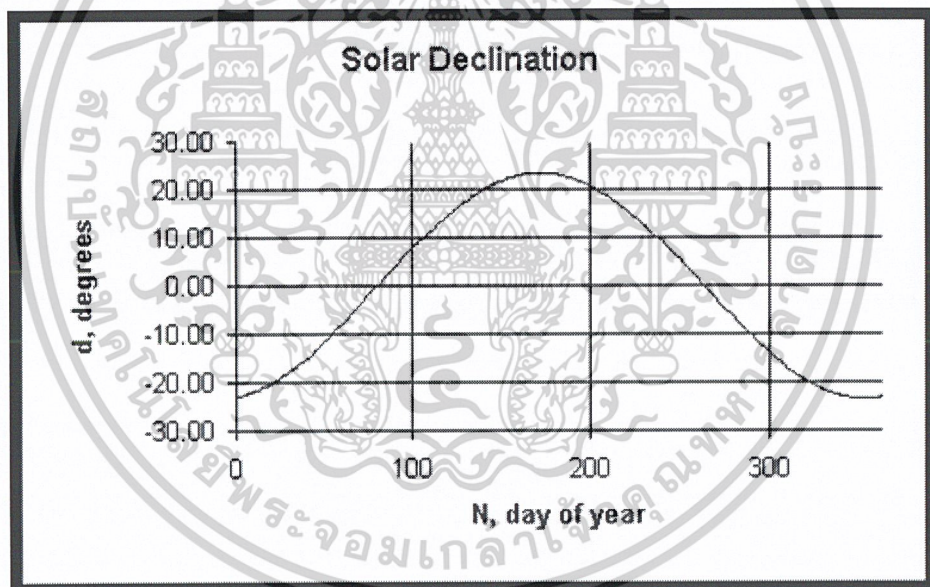
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Solar Declination นี้จะแปรผันแบบ Sinusoidal ตลอดทั้งปี จุดที่มี Solar Declination มากที่สุด ซีกโลกเหนือจะเป็นหน้าร้อน และซีกโลกใต้จะเป็นหน้าหนาว และในทำนองเดียวกัน สำหรับจุดที่มี Solar Declination ทางใต้มีค่ามากที่สุด ฤดูใบไม้ผลิและฤดูใบไม้ร่วง Solar Declination ของดวงอาทิตย์จะเป็น 0

Solar Declination สามารถหาได้จากสมการ

$$\delta = 23.45 \times \sin \left[\frac{360}{365} \times (284 + d) \right]$$

โดย d คือ ค่าวันของปี d=1 คือวันที่ 1 มกราคมของปี



รูปที่ 2.17 แสดงค่า Solar Declination ของแต่ละวันตลอดทั้งปี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยสมการ Declination มาจากการที่โลกเคลื่อนที่รอบดวงอาทิตย์ใช้เวลา 365 วันกวางดุม
ไปได้ 360 องศา

365	วัน ได้มุม 360	องศา
1	วัน ได้มุม $360/365$	องศา
d	วัน ได้มุม $(360 \times d)/365$	องศา



จากรูป 2.16 และ 2.17 จะเห็นว่าค่า Declination เปลี่ยนแปลงอยู่ในช่วง -23.45 ถึง 23.45
และมีค่าแปรตามฟังก์ชัน Sinusoidal จะได้ว่า

$$\delta = 23.45 \times \sin \left[\frac{360}{365} \times d \right]$$

สำหรับสมการข้างบนค่าแรกของ Declination จะเริ่มที่ตำแหน่ง 0 องศาจะมีค่าเป็นค่าศูนย์
แต่ที่จริงแล้วค่าแรกของ Declination จะเริ่มที่ตำแหน่ง 270 องศาจะมีค่าเป็นลบ เราจะต้องบวก
ค่าคงที่เพิ่มเข้าไปเพื่อให้เริ่มต้นค่าเป็นลบ

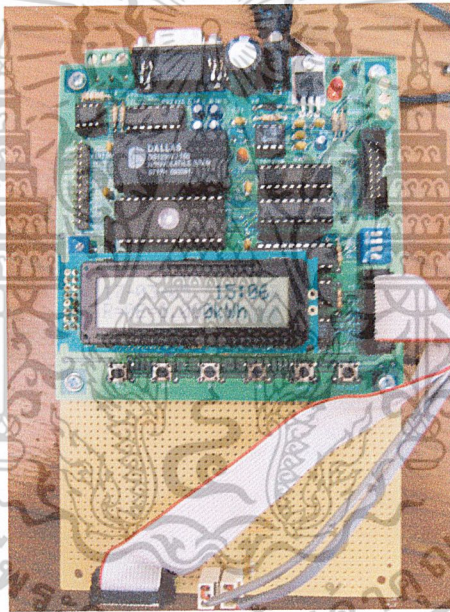
$$\delta = 23.45 \times \sin \left[\frac{360}{365} \times (284 + d) \right]$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

2.6 การวัดปริมาณกระแสไฟฟ้าที่แผงเซลล์แสงอาทิตย์ผลิตได้ในแต่ละวัน

ในการวัดปริมาณกระแสไฟฟ้าที่แผงเซลล์แสงอาทิตย์ผลิตได้ จะใช้บอร์ดไมโครคอนโทรลเลอร์ในการวัดปริมาณกระแสไฟฟ้า โดยจะต่อดีเทคเตอร์เข้ากับมิเตอร์วัดกระแสไฟฟ้าและต่อขาสัญญาณจากดีเทคเตอร์เข้ากับขาอินเทอร์รัพของไมโครคอนโทรลเลอร์ โดยทุกๆการหมุนหนึ่งรอบของมิเตอร์ ดีเทคเตอร์จะส่งสัญญาณพัลส์มาให้กับ ไมโครคอนโทรลเลอร์ และไมโครคอนโทรลเลอร์จะนำค่าจำนวนรอบการหมุนของมิเตอร์มาคำนวณเป็นค่าปริมาณกระแสไฟฟ้าอีกทีหนึ่ง



รูปที่ 2.19 แสดงบอร์ดที่ทำการวัดกระแสไฟฟ้าที่ผลิตได้

บทที่ 3

วิธีดำเนินการวิจัย

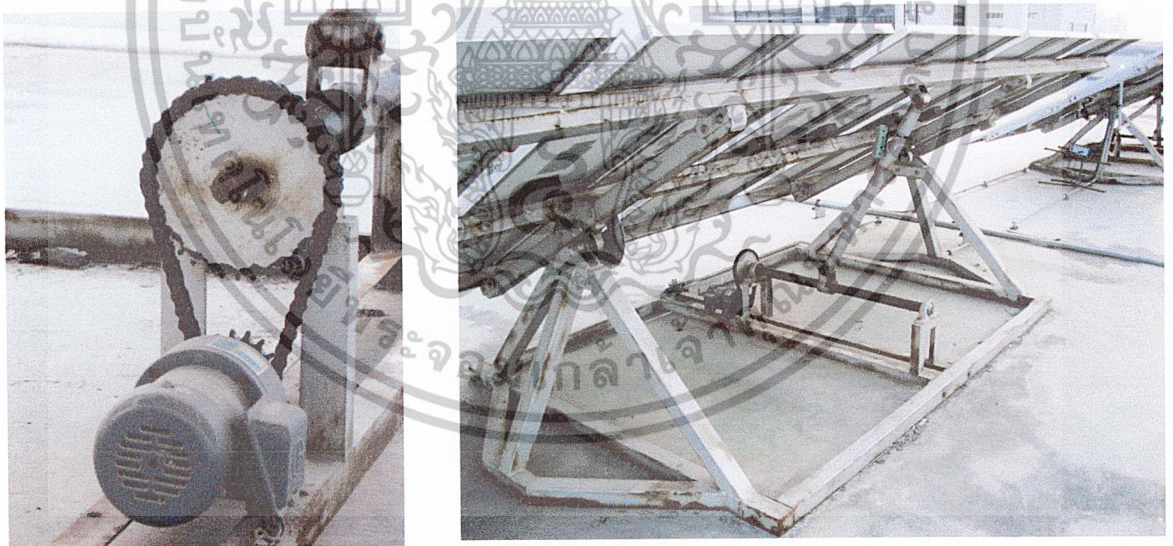
วิธีการดำเนินงานและการรวบรวมข้อมูลต่างๆ

3.1 ส่วนของการติดตั้งอุปกรณ์

ในขั้นแรกของการดำเนินงานเราจะทำการติดตั้งอุปกรณ์ต่างๆเข้ากับชุดทดลองเพื่อที่จะพร้อมสำหรับการดำเนินงานในส่วนต่อไป โดยเราจะเริ่มจาก

3.1.1 การติดตั้งมอเตอร์

ในส่วนแรกเราจะทำการติดตั้งมอเตอร์เข้ากับชุดปรับมุมของแผงเซลล์แสงอาทิตย์ เพื่อใช้มอเตอร์ในการจับชุดปรับมุมของแผงเซลล์แสงอาทิตย์ให้สามารถปรับมุมของเซลล์แสงอาทิตย์ได้ตามแนวทิศเหนือ - ใต้

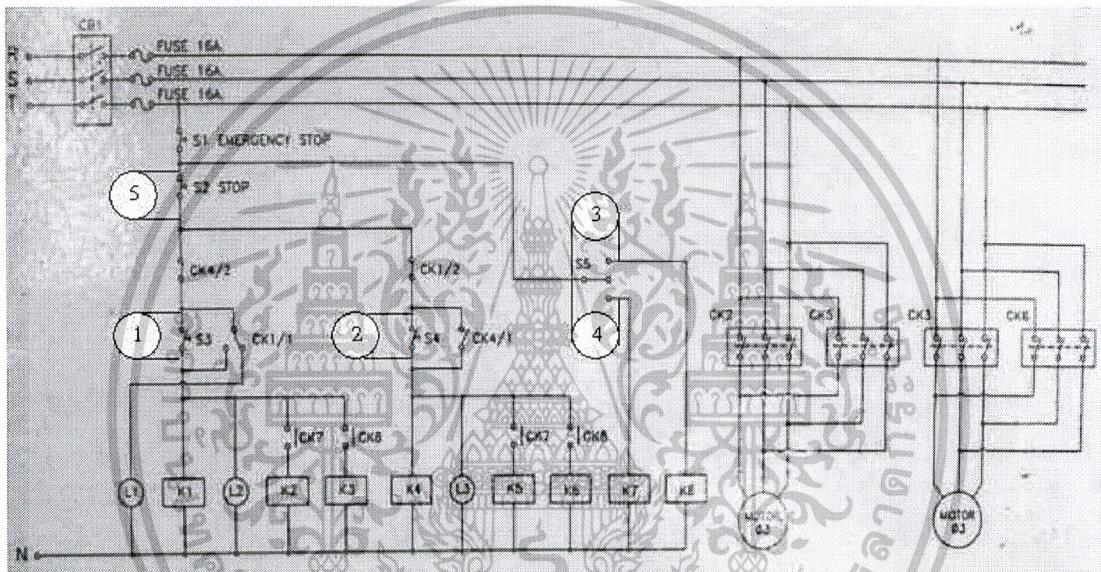


รูปที่ 3.1 แสดงชุดปรับมุมของแผงเซลล์แสงอาทิตย์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

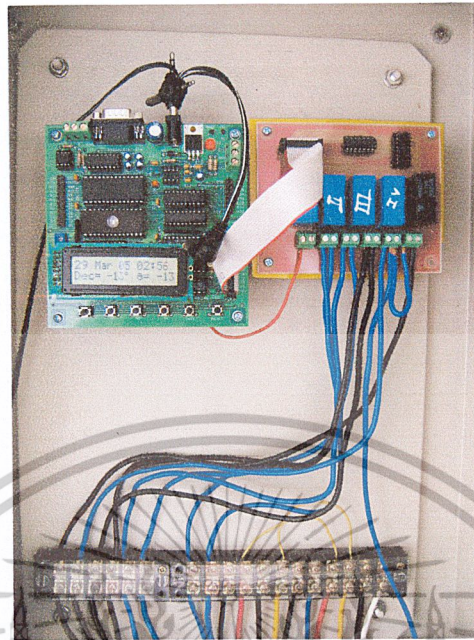
3.1.2 การเชื่อมต่อวงจรที่ใช้ในการควบคุมมอเตอร์

ในส่วนของภาควงจรที่ออกแบบให้เชื่อมต่อกับตู้ควบคุมมอเตอร์และบอร์ดไมโครคอนโทรลเลอร์เพื่อที่จะทำให้สามารถควบคุมมอเตอร์ได้แบบกึ่งอัตโนมัตินั้น เราจะใช้รีเลย์เข้ามาช่วยเป็นอุปกรณ์ในการตัดต่อทำหน้าที่คล้ายกับสวิตช์ที่เราใช้รีเลย์ 5 ตัวในการทำหน้าที่ต่างๆ คือ การเลือกมอเตอร์ซ้าย-ขวา ปรับทิศทาง - ได้ และทำการตัดต่อกระแสไฟฟ้าที่เข้ามาในวงจร ซึ่งใช้บอร์ดไมโครคอนโทรลเลอร์เป็นตัวคำนวณและสั่งการผ่านวงจรรีเลย์เพื่อควบคุมมอเตอร์ในการปรับมุมของแผงเซลล์แสงอาทิตย์ตามค่ามุมที่คำนวณได้



รูปที่ 3.2 แสดงตำแหน่งรีเลย์ที่เชื่อมต่อเข้ากับวงจรของตู้ควบคุมมอเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.3 แสดงวงจรรีเลย์ที่เชื่อมต่อกับบอร์ดไมโครคอนโทรลเลอร์

3.2 ส่วนของการคำนวณค่ามุมตกกระทบของแนวรังสีของแสงอาทิตย์

ในส่วนต่อไปนี้เป็นการศึกษาวิธีการหาค่ามุมค่ามุมของแนวรังสีของแสงอาทิตย์ที่ทำกับพื้นผิวของโลกเพื่อที่จะสามารถปรับแผงเซลล์แสงอาทิตย์ให้แสงตกกระทบแผงเซลล์แสงอาทิตย์ในแนวตั้งฉาก สำหรับประเทศไทยที่ตั้งอยู่สูงจากเส้นศูนย์สูตร 15 องศา ค่ามุมตกกระทบของแนวรังสีของแสงอาทิตย์ที่ทำต่อผิวโลกสามารถคำนวณได้ตามตารางข้างล่างนี้

ตารางที่ 3.1 แสดงค่ามุม Declination ในแต่ละวันตลอดทั้งปีของประเทศไทย

Date	Jan	Feb	Mar	April	May	Jun	July	Aug	Sep	Oct	Nov	Dec
1	38S	32S	23S	10S	0	7N	8N	2N	7S	18S	30S	37S
2	37S	32S	22S	10S	0	7N	8N	2N	7S	18S	30S	37S
3	37S	31S	22S	10S	0	7N	8N	2N	8S	19S	31S	37S
4	37S	31S	22S	9S	0	7N	7N	2N	8S	19S	31S	37S
5	37S	31S	21S	9S	1N	7N	7N	1N	8S	20S	31S	37S
6	37S	30S	21S	8S	1N	7N	7N	1N	9S	20S	31S	37S
7	37S	30S	20S	8S	1N	7N	7N	1N	9S	21S	32S	37S
8	37S	30S	20S	8S	2N	7N	7N	0	10S	21S	32S	37S
9	37S	29S	20S	7S	2N	7N	7N	0	10S	21S	32S	38S
10	37S	29S	19S	7S	2N	8N	7N	0	10S	22S	33S	38S

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่ออนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

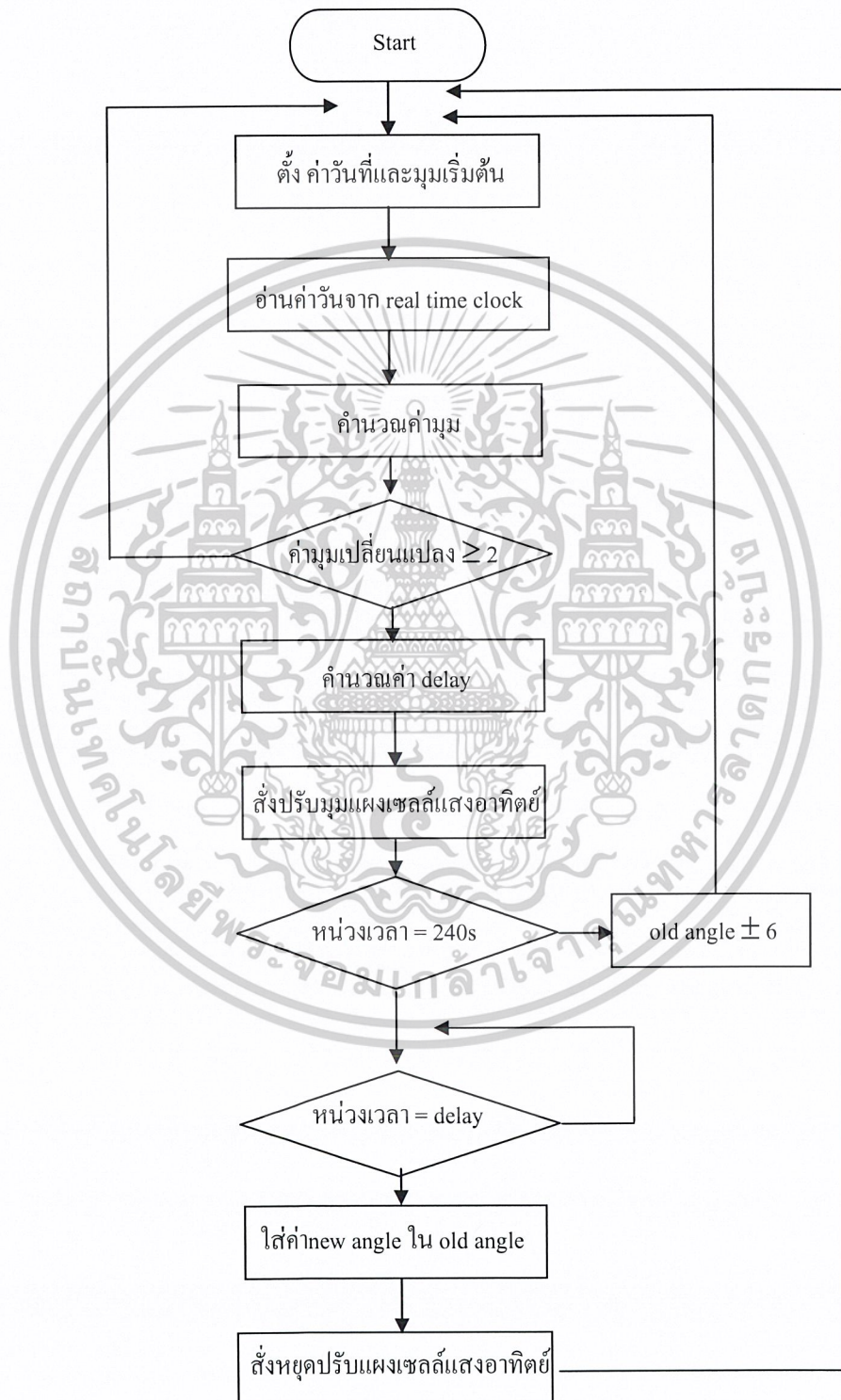
ตารางที่ 3.1(ต่อ) แสดงค่ามุม Declination ในแต่ละวันตลอดทั้งปีของประเทศไทย

Date	Jan	Feb	Mar	April	May	Jun	July	Aug	Sep	Oct	Nov	Dec
11	36S	29S	19S	7S	2N	8N	7N	0	11S	23S	33S	38S
12	36S	28S	18S	6S	3N	8N	6N	0	11S	23S	33S	38S
13	36S	28S	18S	6S	3N	8N	6N	0	12S	23S	33S	38S
14	36S	28S	18S	5S	3N	8N	6N	0	12S	24S	34S	38S
15	36S	27S	17S	5S	3N	8N	6N	1S	12S	24S	34S	38S
16	36S	27S	17S	5S	4N	8N	6N	1S	13S	25S	34S	38S
17	35S	27S	16S	4S	4N	8N	6N	1S	13S	25S	34S	38S
18	35S	26S	16S	4S	4N	8N	6N	2S	14S	25S	34S	38S
19	35S	26S	16S	4S	4N	8N	5N	2S	14S	26S	35S	38S
20	35S	26S	15S	3S	5N	8N	5N	2S	14S	26S	35S	38S
21	35S	25S	15S	3S	5N	8N	5N	3S	15S	26S	35S	38S
22	34S	25S	14S	2S	5N	8N	5N	3S	15S	27S	35S	38S
23	34S	25S	14S	2S	5N	8N	5N	4S	16S	27S	35S	38S
24	34S	24S	14S	2S	5N	8N	4N	4S	16S	27S	36S	38S
25	34S	24S	13S	1S	5N	8N	4N	4S	16S	28S	36S	38S
26	34S	24S	13S	1S	6N	8N	4N	5S	17S	28S	36S	38S
27	33S	23S	12S	1S	6N	8N	4N	5S	17S	28S	36S	38S
28	33S	23S	12S	0	6N	8N	3N	5S	18S	29S	36S	38S
29	33S	-	12S	0	6N	8N	3N	6S	18S	29S	36S	38S
30	33S	-	11S	0	6N	8N	3N	6S	18S	29S	37S	38S
31	32S	-	11S	-	6N	-	3N	6S	-	30S	-	38S

*หมายเหตุ S = south , N = north

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

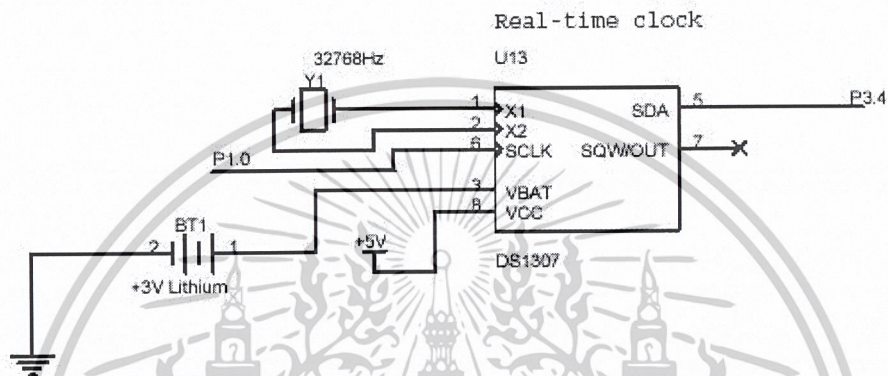
3.3 ส่วนของการเขียนโปรแกรม



รูปที่ 3.4 โฟลว์ชาร์ทแสดงการทำงานของโปรแกรมที่ใช้ควบคุมชุดปรับมุมของแผงเซลล์

แสงอาทิตย์ เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำหรับส่วนของการเขียน โปรแกรมเราจะใช้ภาษาซีในการเขียนโปรแกรมร่วมกับบอร์ด ไมโครคอนโทรลเลอร์เพื่อควบคุมชุดปรับมุมของแผงเซลล์แสงอาทิตย์โดยแสดงขั้นตอนต่างๆใน โพลชาร์ทข้างต้น เราจะใช้ไอซีเบอร์ DS 1307 (real time clock) ซึ่งทำหน้าที่เป็นนาฬิกาเพื่อที่จะ เรียกค่าวันและเดือนเข้ามาคำนวณหาค่ามุมตกกระทบของแนวรังสีของแสงอาทิตย์ที่ทำกับพื้นผิว ของโลกในแต่ละวัน โดยใช้วิธีการห้วงเวลาในการปรับมุมของแผงเซลล์แสงอาทิตย์



รูปที่ 3.5 แสดงวงจรของ DS 1307

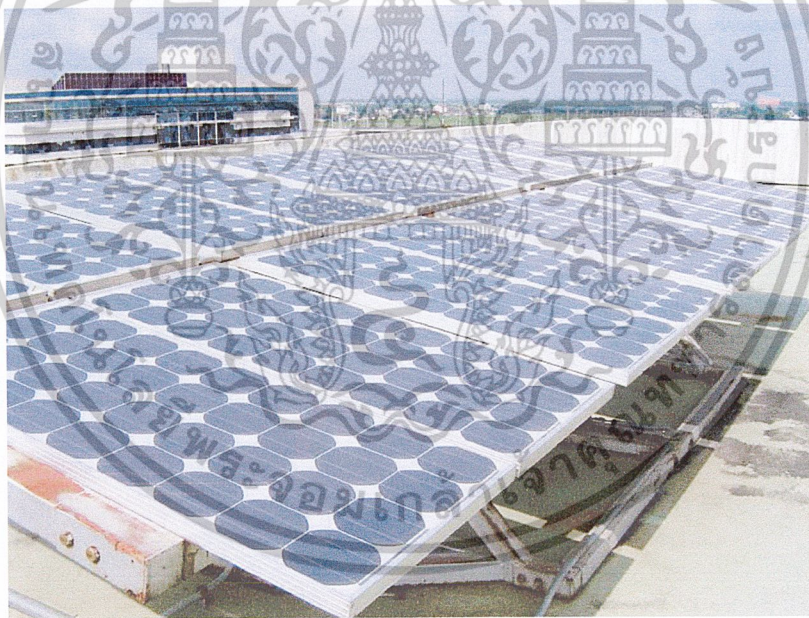
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

ผลการทดลองและการอภิปรายผล

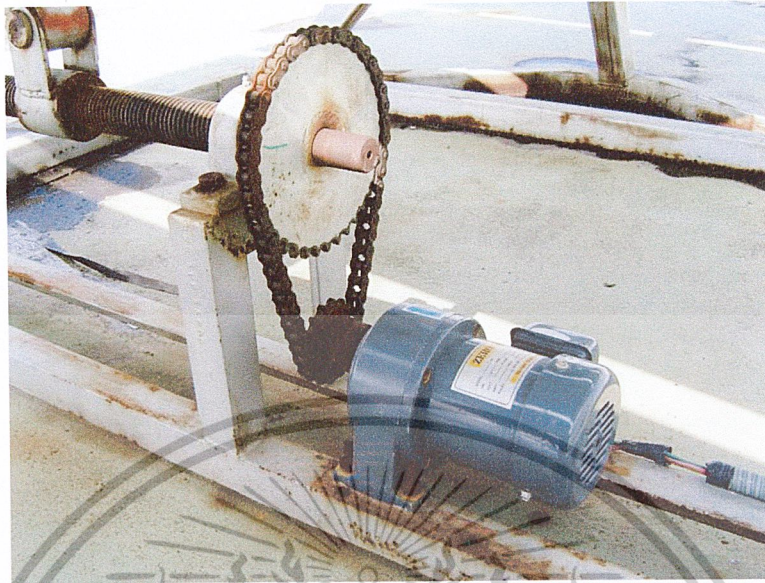
4.1 ผลจากชุดปรับมุมของแผงเซลล์แสงอาทิตย์

การทดลองในส่วนของชุดปรับมุมของแผงเซลล์แสงอาทิตย์ เราจะให้บอร์ดไมโครคอนโทรลเลอร์ทำงานอย่างต่อเนื่องด้วยระบบอัตโนมัติตลอด 1 เดือน โดยบอร์ดไมโครคอนโทรลเลอร์เป็นตัวคำนวณและตั้งปรับมุมของแผงเซลล์แสงอาทิตย์ ซึ่งจะทำการปรับมุมเมื่อมุมมีการเปลี่ยนแปลงมากกว่า 2 องศา และทำการเก็บผลในแต่ละวันสำหรับมุมของเซลล์แสงอาทิตย์อ่านได้จากตัววัดมุมที่ติดตั้งไว้ที่แผงเซลล์แสงอาทิตย์ จะได้ผลตามที่แสดงในตาราง



รูปที่ 4.1 แผงเซลล์แสงอาทิตย์ที่ควบคุมด้วยชุดปรับมุม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.2 แสดงมอเตอร์ที่ควบคุมด้วยบอร์ดไมโครคอนโทรลเลอร์



รูปที่ 4.3 แสดงตัววัดมุมที่ติดตั้งไว้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.1 แสดงผลการทดลองที่ได้ตลอดเดือนตุลาคม

วัน / เดือน	Declination	ค่าจากการทดลอง	ค่าความคาดเคลื่อน	กระแสที่ผลิตได้(kWh)
1-Oct	18	18.0	0.0	7.343
2-Oct	18	18.0	0.0	6.670
3-Oct	19	18.0	1.0	10.410
4-Oct	19	18.0	1.0	9.523
5-Oct	20	20.0	0.0	5.968
6-Oct	20	20.0	0.0	2.845
7-Oct	21	20.0	1.0	4.265
8-Oct	21	20.0	1.0	4.823
9-Oct	21	20.0	1.0	4.633
10-Oct	22	22.0	0.0	0.000
11-Oct	22	22.0	0.0	3.130
12-Oct	22	22.0	0.0	3.953
13-Oct	22	22.0	0.0	9.580
14-Oct	23	22.0	1.0	9.172
15-Oct	23	22.0	1.0	9.257
16-Oct	24	23.5	0.5	9.187
17-Oct	24	23.5	0.5	9.645
18-Oct	25	23.5	1.5	9.685
19-Oct	25	23.5	1.5	9.205
20-Oct	25	23.5	1.5	9.335
21-Oct	26	25.0	1.0	8.997
22-Oct	26	25.0	1.0	8.748
23-Oct	26	25.0	1.0	9.220
24-Oct	27	25.0	2.0	9.780
25-Oct	27	25.0	2.0	9.898
26-Oct	27	25.0	2.0	8.845
27-Oct	28	26.5	1.5	8.683
28-Oct	28	26.5	1.5	9.620
29-Oct	28	26.5	1.5	8.610
30-Oct	29	26.5	2.5	9.720
31-Oct	29	26.5	2.5	0.000

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ทางการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่ใช้

4.2 สรุปผลการทดลอง

จากการทดลองตลอด 1 เดือนในเดือนตุลาคมโดยการเก็บผลทุกวัน จะเห็นว่าชุดปรับมุมของเซลล์แสงอาทิตย์ที่ใช้การห้วงเวลาในการปรับมุม ค่ามุมที่ได้จากการทดลองมีค่าผิดพลาดจากค่าที่คำนวณได้ประมาณ ± 2 องศา



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

สรุปและข้อเสนอแนะ

ในบทนี้เป็นการสรุปผลการทดลองที่ได้จากชุดปรับมุมของแผงเซลล์แสงอาทิตย์ โดยมีบอร์ดไมโครคอนโทรลเลอร์เป็นตัวคำนวณและตั้งปรับมุมของแผงเซลล์แสงอาทิตย์ซึ่งทำงานอย่างต่อเนื่องตลอด 1 เดือน โดยจะกล่าวถึงผลการทดลอง และปัญหาที่เกิดขึ้นระหว่างการดำเนินงาน

5.1 สรุปผลการทดลอง

สำหรับชุดปรับมุมของแผงเซลล์แสงอาทิตย์ เราสามารถที่จะปรับมุมให้รับแสงได้มากที่สุด โดยคำนวณจากค่า Solar Declination ซึ่งเป็นค่ามุมตกกระทบของแนวรังสีของดวงอาทิตย์ที่ทำกับพื้นผิวของโลก และสามารถปรับมุมของแผงเซลล์แสงอาทิตย์โดยใช้ชุดปรับมุมแผงเซลล์แสงอาทิตย์ที่สามารถทำงานได้แบบกึ่งอัตโนมัติ ซึ่งค่าที่ได้จากการทดลองมีค่าใกล้เคียงกับค่าจริงที่คำนวณได้

5.2 ปัญหาที่พบและข้อเสนอแนะ

จากการดำเนินงานสร้างชุดปรับมุมของแผงเซลล์แสงอาทิตย์ในงานวิจัยนี้พบว่ามีปัญหาบางประการที่ควรได้รับการแก้ไขและปรับปรุงต่อไปได้แก่

1. ในส่วนกลไกการปรับมุมของชุดปรับมุมของแผงเซลล์แสงอาทิตย์ในการวิจัยนี้พบว่าชิ้นส่วนของชุดปรับมุมมีความไม่สมบูรณ์เนื่องจากเกิดสนิม และมีความฝืดขึ้นทำให้อัตราเร็วในการปรับมุมมีค่าไม่คงที่
2. ในส่วนของการปรับมุมในการวิจัยนี้จะใช้การหน่วงเวลาในการปรับค่ามุม ซึ่งค่าที่ได้ นั้นอาจจะไม่ตรงกับค่ามุมจริงที่ได้ ในส่วนนี้อาจจะแก้ไขโดยการใช้อุปกรณ์เพื่อตรวจสอบค่ามุมที่ปรับได้จริงจากชุดปรับมุมของแผงเซลล์แสงอาทิตย์
3. ในส่วนของการทดลองนี้เราไม่สามารถเปรียบเทียบค่าของไฟฟ้าที่ผลิตได้จากเซลล์แสงอาทิตย์ที่มีการปรับมุมกับเซลล์แสงอาทิตย์ส่วนที่มีมุมคงที่ เนื่องจากไฟฟ้าที่ผลิตได้นั้นต่อเข้ากับมาตรวัดไฟเพียงแค่อันหนึ่งตัวเท่านั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

Larry D. Partain “Solar cells and their applications” wiley interscience (wiley series in microwave and optical engineering). 1995. ISBN 0-471-57420-1

“How Solar Cells work” available at: <http://science.howstuffworks.com/solar-cell.htm>

“Photovoltaics: Solar Electricity and Solar cells in Theory and practice” available at: www.solarserver.de/wisser/photovoltaik-e.html

“Basic electrical ac-motor rotor operation” available at : www.yoursitetop10.com/basic_ac_motor_operation.html



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาคผนวก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Source Code

โปรแกรมที่ใช้ปรับมุมแผงเซลล์แสงอาทิตย์

CONTROL.C

```
#include<stdio.h>
#include<reg52.h>
#include<math.h>
#include<intrins.h>
#include<absacc.h>
#include"ds1307m.c"
#include"LCDdrv.c"
#include"setDT.c"

sfr WDT = 0xA6;

sbit  LED = P1^7;

void set_angle();

int day[] = {0,0,31,59,90,120,151,181,212,243,273,304,334};
int *month[] = {"0","Jan","Feb","Mar","Apr","May","Jun","Jul","Aug","Sep","Oct","Nov","Dec"};
int dec,a,b,angle,old_angle,direct,delay_time,total_time,d;
int run_1s, tick, angle_flag, reset_time, flag_chk, run, set_degree, act_key, chk_key ;
float Solar_Dec;

#####
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

timer() interrupt 1
{
    TH0 = 0xdc;
    tick++;
    if(tick==100)
    {
        tick = 0;
        delay_time++;
        LED ^= 1;
        run_1s = 1;
    }
    WDT = 0x1e;
    WDT = 0xe1;
}

//#####

int H2D(int hex)
{
    int val,val_I;
    val_I = hex&0x0f;
    val = hex>>=4;
    val = val*10;
    dec = val + val_I;
}

//#####

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

void calculate()
{
    float DOY,cal,degree,rad;

    DOY    = ( day[H2D(time[5])] + H2D(time[4]) )+284;
    degree  = 0.9863 * DOY;
    rad     = degree*0.0174;
    cal     = sin(rad);
    Solar_Dec = (23.45*cal)-15;
    angle   = Solar_Dec;
}

#####

void chk_time()
{
    if(time[2]==0x05 && time[1]==0x00 && time[0]==0x00)
    {
        calculate();
        if(angle >= -28)
        {
            direct = old_angle-angle;
            if(abs(direct) >= 2)
            {
                total_time = (abs(direct)*80)/2;
                printf("\ntotal = %d \n",total_time);
                angle_flag = 1;
            }
        }
    }
}
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

if(time[2]==0x05 && time[1]==0x04 && time[0]==0x00)
{
    P1 = 0xfb;
    printf("\n delay_time = %d d =%d P1 = %x\n",delay_time,d,P1);
    if(angle_flag == 1)
    {
        if(direct < 0 )
        {
            old_angle = old_angle + 6;
        }
        if(direct > 0 )
        {
            old_angle = old_angle - 6;
        }
        angle_flag = 0;
        reset_time = 1;
    }
}
if(time[2]==0x05 && time[1]==0x04 && time[0]==0x05)
{
    P1 = 0xff;
    printf("\n delay_time = %d d =%d P1 = %x\n",delay_time,d,P1);
}
}
#####

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

void show()
{
    char line[16],line1[16];

    sprintf(line,"%02x %s %02x %02x:%02x ",time[4],month[ H2D(time[5]) ],time[6],time[2],time[1]);
    goto_xy(0,0);
    Puts(line);
    sprintf(line1,"Dec= %d%c %c= %d%c      ",angle,0xdf,0xf2,old_angle,0xdf);
    goto_xy(0,1);
    Puts(line1);
}

#####

void setting(void)
{
    char line[16],line1[16];

    if(run == 0)
    {
        sprintf(line," Set %c = %d%c      ",0xf2,old_angle,0xdf);
        goto_xy(0,0);
        Puts(line);
        sprintf(line1," Press S3      ");
        goto_xy(0,1);
        Puts(line1);
    }
}

#####

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

void change_angle()
{
  if(angle_flag == 1)
  {
    if(direct < 0 )
    {
      if(reset_time == 1)
      {
        P1 = 0x9f;
        delay_time = 0;
        reset_time = 0;
      }
      if(delay_time == 5)
      {
        P1 = 0xff;
      }
      d = delay_time;
      if(delay_time == total_time)
      {
        P1 = 0xfb;
        old_angle = angle;
      }
      if(delay_time == (total_time+5))
      {
        P1 = 0xff;
        reset_time = 1;
        angle_flag = 0;
      }
    }
  }

  //#####

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

if(direct > 0 )
{
    if(reset_time == 1)
    {
        P1 = 0xdd;
        delay_time = 0;
        reset_time = 0;
    }
    if(delay_time == 5)
    {
        P1 = 0xff;
    }
    d = delay_time;
    if(delay_time == total_time)
    {
        P1 = 0xfb;
        old_angle = angle;
    }
    if(delay_time == (total_time+5))
    {
        P1 = 0xff;
        reset_time = 1;
        angle_flag = 0;
    }
}
}
}

#####

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

void update_angle()
{
    if (!(inp&0x02))
    {
        calculate();
    }
}

```

#####

```

void set_angle()
{
    if (!(inp&0x04))
    {
        set_degree = 1;
    }

    if (inp&0x04)
    {
        set_degree = 0;
    }
}

```

```

if( (!(inp&0x20)) && set_degree && act_key)
{
    old_angle = old_angle * -1;
    act_key = 0;
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

if( !(inp&0x40) && set_degree && act_key)
{
    old_angle--;
    if(old_angle < -38)
    {
        old_angle = 38;
    }
    act_key = 0;
}
if( !(inp&0x80) && set_degree && act_key)
{
    old_angle++;
    if(old_angle > 38)
    {
        old_angle = 0;
    }
    act_key = 0;
}
printf("\r\nold_angle = %d ",old_angle);
}

#####

void set_time()
{
    if (inp&0x10)
        { check=0; }
    if( !(inp&0x10)&&(check==0)&&!(inp&0x08))
        {
            check=chk=chd=cht=ch=1;
            setDT();
        }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    }
}

//#####

```

```

void chk_process()

```

```

{
    if(run_1s == 1)
    {
        RTCread();
        update_angle();
        chk_time();
        show();
        change_angle();
        run_1s = 0;
    }
}

```

```

//#####

```

```

void main()

```

```

{
    P1 = 0xff;

    run_1s = tick = a = delay_time = total_time = angle_flag = flag_chk = run = 0;
    b = reset_time = 1;

    WDT = 0x1e;
    WDT = 0xe1;
    TH0 = 0xdc;
    TMOD |= 1;
    IE = 0x82;
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

TR0 =1;

initRTC();
InitLcd();

old_angle = -38;
calculate();

while(1)
{
    inp=XBYTE[0x200];

    chk_key = inp & 0xf0;
    if( chk_key == 0xffff )
        { act_key = 1; }
    if (!(inp&0x10))
//        { flag_chk = 1; }
//    if (!(inp&0x20) && (flag_chk == 1 ))
        { run =1; }

    setting();
    set_angle();
    set_time();

    if(run==1)
        {
            chk_process();
        }
}
}
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรมของ Solar energy meter

PULSE.C

```
#include<stdio.h>
#include<reg52.h>
#include<intrins.h>
#include<absacc.h>
#include<string.h>
#include"lcddrv.c"
#include"ds1307.c"
#include"setDT.c"
//#include"setTO.c"//extension;can set time on and time off, but it's not complete yet.
#include"showdata.c"
#include"sds1820.c"
#include"login.c"

float energy,round,old_energy;
bit check,check2,work,plcd;
xdata float res[31][12];
sbit on=P1^7;
int delay_led,delay_time,old_min,key;

sfr WDT = 0xA6;

void cal();
void lcd();
void checkwork();
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

void count() interrupt 0
{
    if (work==1)
    {
        WDT = 0x1e;
        WDT = 0xe1;
        round++;
        on=delay_led=0;
        cal();
    }
}

void function() interrupt 1
{
    TH0=0xdc;
    TL0=0;
    if (delay_led++>=10) on=1;
    if (delay_time++==10)
    {
        delay_time=0;
        RTCread();
        WDT = 0x1e;
        WDT = 0xe1;
        ds1820();
        WDT = 0x1e;
        WDT = 0xe1;
        if (plcd) lcd();
        checkwork();
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
}
```

```
void watchdog() interrupt 5
```

```
{
```

```
    TF2=0;
```

```
    WDT = 0x1e;
```

```
    WDT = 0xe1;
```

```
}
```

```
void online()
```

```
{
```

```
    plcd=0;
```

```
    if /*!(old_energy!=energy)//&&(log==1)//use &&(log==1)) to work with login
```

```
    {
```

```
        printf("\n%02x-%02x-%02x %02x:%02x:%02x R=%.0f E=%.3fkWh T=%.1f C
```

```
Tmax=%.1f C Tav=%.1f
```

```
C",time[4],time[5],time[6],time[2],time[1],time[0],round,energy,rtemp,temppmax,tempav);
```

```
        //printf("\n%02x-%02x-%02x
```

```
%02x:%02x:%02x",time[4],time[5],time[6],time[2],time[1],time[0]);
```

```
        old_energy=energy;
```

```
    }
```

```
    plcd=1;
```

```
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

/*void logintimeout()
{
    if (old_min!=time[1])
    {
        old_min=time[1];
        key++;
        plcd=0;
        if ((key==3)&&log)
        {
            log=key=0;
            printf("\nconnection time out\n");
        }
        plcd=1;
    }
}*/

void printdata()
{
    int i,j,k;
    printf("\nDate-Month,Energy(kWh),Tmax(C),Tav(C)\n");
    for(i=1;i<=12;i++)
    {
        if ((i==1)||(i==3)||(i==5)||(i==7)||(i==8)||(i==10)||(i==12))
            { k=31; }
        if ((i==4)||(i==6)||(i==9)||(i==11))
            { k=30; }
        if (i==2)
            { k=29; }
        for(j=1;j<=k;j++)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        {
            if (res[j][i]>=0&&res[j][i]<=20.0)
                printf("%02d-%02d,%.3f,%.1f,%.1f\n",j,i,res[j][i],tmax[j][i],tav[j][i]);

            else printf("%02d-%02d,n/a,%.1f,%.1f\n",j,i,tmax[j][i],tav[j][i]);
        }
    }
}

printEN()
{
    printf("\nENERGY = %.3f kWh T=%.1f C Tmax=%.1f C Tav=%.1f
C",energy,rtemp,tempmax,tempav);
}

void lcd()
{
    char line0[16],line1[16];
    goto_xy(0,0);
    sprintf(line0,"%02x-%02x-%02x
%02x:%02x"/*:%02x"*/,time[4],time[5],time[6],time[2],time[1]/*,*time[0]*/);
    Puts(line0);
    goto_xy(0,1);
    sprintf(line1,"R=%.0fE=%.3fkWh ",round,energy);
    Puts(line1);
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

void cal()
{
    energy = round/600;
}

int htoc (int x)
{
    int a,b,c,d,e;
    a=x&0xf0;
    b=x&0x0f;
    c=a>>4;
    d=c*10;
    e=d+b;
    return(e);
}

void checkwork()
{
    if ((time[2]>=ctoh(5))&&(time[2]<ctoh(18)))        { work=1; }
    else { work=0; }
    if ((time[2]==ctoh(18))&&(time[1]==0)&&(time[0]==0))
    {
        res[htoc(time[4])][htoc(time[5])]=energy;
        //printf("%f\n",res[htoc(time[4])][htoc(time[5])]);
        tmax[htoc(time[4])][htoc(time[5])]=tempmax;
        tav[htoc(time[4])][htoc(time[5])]=tempav;
    }
    if ((time[2]==ctoh(18))&&(time[1]==0)&&(time[0]==1))        { work=0; }
    if ((time[2]==4)&&(time[1]==ctoh(59))&&(time[0]==ctoh(59)))

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

{
    WDT = 0xe;
    WDT = 0xe;
    round=energy=0;
    tempmax=tempav=rtemp;
}
}

```

```

void checkkey()

```

```

{
    if (RI==1)
    {
        RI=key=0;
        /*if (log==0)
        {
            plcd=0;
            login();//work fine while connected with hyperterminal but still have a
            problem with telnet
            plcd=1;
        }
        if (log==1)
        {*/
            switch(SBUF)
            {
                //case(0x45,0x65): log=0; plcd=0; printf("\nlogout..."); plcd=1;
                break;
                case(0x52,0x72): plcd=0; printEN(); plcd=1; break;
                case(0x41,0x61): plcd=0; printdata(); plcd=1; break;
                case(0x3F): plcd=0;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

printf("\n HELP\n");
printf("\n?- This help list\n");
printf("R- Read Energy\n");
printf("A- Read All data\n");
//printf("E- Logout and Exit");

default: plcd=0; printf("\nSolar Energy Meter (? for HELP)\n>>");

plcd=1;

    }
    /*}
    if (log==2)
    {
        switch(SBUF)
        {
            case(0x45,0x65): log=0; plcd=0; printf("\nlogout..."); plcd=1;
break;
            case(0x52,0x72): plcd=0; printEN(); plcd=1; break;
            case(0x3F): plcd=0;
                printf("\n HELP\n");
                printf("\n?- This help list\n");
                printf("R- Read Energy\n");
                printf("E- Logout and Exit");
                default: plcd=0; printf("\nSolar Energy Meter (? for HELP)\n>>");

plcd=1;

        }
    }
    /*/
}
}
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

void main()
{
    //round=energy=work=0;
    delay_time=log=key=0;
    on=plcd=1;
    IE |= 0x83;
    IP |= 0x01;
    TCON |= 0x01;
    TMOD |= 0x01;
    TH0=0xdc;
    TL0=0;
    ET2=1;
    RCAP2L=0xff;
    RCAP2H=0xef;
    TH2=0xef;
    TL2=0xff;
    TCON |= 0x10;
    TR2=1;
    WDT = 0x1e;
    WDT = 0xe1;
    InitLcd();
    initRTC();
    Puts(" Loading... ");
    dlay(30000);
    while(1)
    {

        inp=XBYTE[0x200];
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

if (inp&0x10)
    {    check=0;    }
if ((!(inp&0x10))&&(check==0)&&(!(inp&0x08)))
{
    check=chk=chd=cht=ch=1;
    plcd=0;
    setDT();
    plcd=1;
}
if (!(inp&0x80))
{
    plcd=0;
    showdata();
    plcd=1;
}
checkkey();
online();
//logintimeout();//if need to use login function,enable this and every disable
function in checkkey.
}
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

LCDDRV.C

// below addresses are lcd's registers mapped to external data memory space

// see the schematic of 8051sbc for memory mapped i/o

```
#define BUSY 0x80 // busy flag location
```

```
#define LCD_command_write (*((unsigned char xdata*)0x0000))
```

```
#define LCD_data_write (*((unsigned char xdata*)0x0001))
```

```
#define LCD_command_read (*((unsigned char xdata*)0x0002))
```

```
#define LCD_data_read (*((unsigned char xdata*)0x0003))
```

```
#define LCD 0x0A
```

```
goto_xy(char x,char y);
```

```
char LcdReady(void)
```

```
{  
    unsigned char flag;  
    while( (flag=LCD_command_read)&BUSY)  
    {  
        if(flag == 0xFF) return -1;  
    }  
    return 1;  
}
```

```
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

void clr_screen(void)
{
    LcdReady();
    LCD_command_write=0x01;
}

```

```

char InitLcd(void)
{
    if(-1 ==LcdReady()) return -1;
    LCD_command_write=0x38;

    LcdReady();
    LCD_command_write=0x0c;
    clr_screen();
    goto_xy(0,0);
}

```

```

goto_xy(char x,char y)
{
    LcdReady();
    switch(y){
        case 0 : LCD_command_write=0x80+x; break;
        case 1 : LCD_command_write=0xC0+x; break;
        case 2 : LCD_command_write=0x94+x; break;
        case 3 : LCD_command_write=0xd4+x; break;
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

char *Puts(char* str)
{
    unsigned char i;
    for (i=0; str[i] != '\0'; i++){
        LcdReady();
        LCD_data_write=str[i];

        // for(l=0;l<4000;l++)
        //     _nop_;
        // LCD_command_write=0x18;
        // for(l=0;l<8000;l++)
        //     _nop_;
    }
    return str;
}

void putch_lcd(char ch)
{
    LcdReady();
    LCD_data_write=ch;
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

SETDT.C

```
xdata int secs,mins,hous,days,mons,yers;
char line0[16],line1[16],inp;
bit check,chk,chd,cht,ch;

void dlay(int n)
{
    int j;
    for(j=0;j<=n;j++);
}

void setDATE()
{
    sprintf(line0,"DATE=%02d MONTH=%02d",days,mons);
    goto_xy(0,0);
    Puts(line0);
    sprintf(line1," YEAR=%02d ",yers);
    goto_xy(0,1);
    Puts(line1);
    inp=XBYTE[0x200];
    if (!(inp&0x20))
    {
        dlay(10000);
        days++;
        if (days==32)
            { days=1; }
    }
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

if (!(inp&0x40))
{
    dlay(10000);
    mons++;
    if (mons==13)
        {    mons=1;    }
}
if (!(inp&0x80))
{
    dlay(10000);
    yers++;
    if (yers==100)
        {    yers=1;    }
}
if (!(inp&0x10))
{
    check=1;
    chd=0;
}
}

void setTime()
{
    sprintf(line0,"hour=%02d min=%02d",hous,mins);
    goto_xy(0,0);
    Puts(line0);
    sprintf(line1,"  sec=%02d  ",secs);
    goto_xy(0,1);
    Puts(line1);
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

inp=XBYTE[0x200];
if (!(inp&0x20))
{
    dlay(10000);
    hous++;
    if (hous==24)
        { hous=0; }
}
if (!(inp&0x40))
{
    dlay(10000);
    mins++;
    if (mins==60)
        { mins=0; }
}
if (!(inp&0x80))
{
    dlay(10000);
    secs++;
    if (secs==60)
        { secs=0; }
}
if (inp&0x10)
    { ch=0; }
if (!(inp&0x10)&&(ch==0))
{
    check=1;
    cht=0;
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
}
```

```
int ctoh(int z)
```

```
{
```

```
    int a,b,c;
```

```
    a=z/10;
```

```
    b=z%10;
```

```
    a=a*16;
```

```
    c=a+b;
```

```
    return(c);
```

```
}
```

```
void convert()
```

```
{
```

```
    secs=ctoh(secs);
```

```
    mins=ctoh(mins);
```

```
    hous=ctoh(hous);
```

```
    days=ctoh(days);
```

```
    mons=ctoh(mons);
```

```
    yers=ctoh(yers);
```

```
}
```

```
void RTCwrite()
```

```
{
```

```
    start();
```

```
    write(0xD0);
```

```
    write(0x00);
```

```
    write(secs);
```

```
    write(mins);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

write(hous);
write(0x11);
write(days);
write(mons);
write(yers);
stop();
}

```

```

void setDT()
{
days=mons=1;
yers=secs=mins=hous=0;
InitLcd();
while (chd==1)
{
inp=XBYTE[0x200];
if (inp&0x10)
{
chk=0; }
if ((inp&0x10)&&(chk==0))
{
setDATE(); }
}
InitLcd();
while (cht==1)
{
setTIME(); }
convert();
RTCwrite();
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

SDS1820.C

```
/*#include<stdio.h>
#include<reg52.h>*/

/***** I/O PORT *****/

sbit TMDAT = P3^5;

/*RESULT*/

unsigned char mtemp;
float temp,rtemp,tempmax,tempav,old_temp;
bit work;
xdata float tmax[31][12],tav[31][12];

/***** FUNCTION *****/

void dmsec (unsigned int count) { // mSec Delay 11.0592 Mhz
    unsigned int i; // Keil v5.2
    while (count) {
        i = 115; while (i>0) i--;
        count--;
    }
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

void tmreset (void)
{
    // Reset TX
    unsigned int i;
    TMDAT = 0;
    i = 103; while (i>0) i--;    // Approx 900 uS
    TMDAT = 1;
    i = 4; while (i>0) i--;
}

void tmpr (void) {
    // Wait for Presence RX
    unsigned int i;
    while (TMDAT);
    while (~TMDAT);
    i = 4; while (i>0) i--;
}

bit tmrbit (void) {
    // read one bit
    unsigned int i;
    bit dat;
    TMDAT = 0; i++;
    TMDAT = 1; i++; i++;
    dat = TMDAT;
    i = 8; while (i>0) i--;    // Approx 65 uS
    return (dat);
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

unsigned char tmrbyte (void) {      // read one byte
unsigned char i,j,dat;
    dat = 0;
    for (i=1;i<=8;i++)
    {
        j = tmrbit ();
        dat = (j << 7) | (dat >> 1);
    }
    return (dat);
}

void tmwbyte (unsigned char dat)
{
    unsigned int i;
    unsigned char j;
    bit testb;
    for (j=1;j<=8;j++) {
        testb = dat & 0x01;
        dat = dat >> 1;
        if (testb) {
            TMDAT = 0;          // Write 1
            i++; i++;           // Approx 4 uS
            TMDAT = 1;
            i = 8; while (i>0) i--;          // Approx 65 uS
        }
        else {
            TMDAT = 0;          // Write 0
            i = 8; while (i>0) i--;          // Approx 65 uS
            TMDAT = 1;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        i++; i++;                // Approx 4 uS
    }
}
}

```

```

void tmstart (void) {          // ds1820 start convert
    tmreset ();
    tmpre ();
    dmsec (1);
    tmwbyte (0xcc);           // skip rom
    tmwbyte (0x44);           // convert
}

```

```

void tmrtemp (void) {         // read temp
    tmreset ();
    tmpre ();
    dmsec (1);
    tmwbyte (0xcc);           // skip rom
    tmwbyte (0xbe);           // convert
    temp = tmrbyte ();        // LSB
    mtemp = tmrbyte ();      // MSB
}

```

```

void tmp() //check for negative temp
{
    if (mtemp) return; //don't care negative temp
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

void comtemp()
{
    if ((rtemp>tempmax)&&work)
    {
        //if ((rtemp-tempmax)<50)
        if (rtemp<70)
            tempmax=rtemp;
    }
}

void avtemp()
{
    if ((old_temp!=rtemp)&&work)
    {
        /*if ((rtemp>old_temp)&&(rtemp-old_temp<30))
        {
            old_temp=rtemp;
            tempav=(tempav+rtemp)/2;
        }
        if ((rtemp<old_temp)&&(old_temp-rtemp<30))
        {
            old_temp=rtemp;
            tempav=(tempav+rtemp)/2;
        }
        */
        if (rtemp<70)
        {
            old_temp=rtemp;
            tempav=(tempav+rtemp)/2;
        }
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    }
}

/***** MAIN *****/

void ds1820 (void) {
    //unsigned int i;
    //temp=mtemp=0;
    //while (1) {
    tmstart ();           // ds1820 start convert
    dmsec (1);
    tmrtemp ();          // read temperature
    tmp();
    rtemp=temp/2;//printf("\n%.1f",temp/2);
    comtemp();
    avtemp();
    //}
}
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

SHOWDATA.C

```
char line0[16],line1[16];
int daym,moum;
xdata float res[31][12],tav[31][12];
```

```
void setshow()
{
    goto_xy(0,0);
    Puts(" DATE AND MONTH ");
    sprintf(line1," %d-%d ",daym,moum);
    goto_xy(0,1);
    Puts(line1);
    inp=XBYTE[0x200];
    if (!(inp&0x20))
    {
        dlay(10000);
        daym++;
        if (daym==32)
            { daym=1; }
    }
    if (!(inp&0x40))
    {
        dlay(10000);
        moum++;
        if (moum==13)
            { moum=1; }
    }
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

void showvalue()
{
    char deg;
    deg=0xdf;
    sprintf(line0,"%d-%d Tav=%.1f%cC  ",daym,moum,tav[daym][moum],deg);
    goto_xy(0,0);
    Puts(line0);
    sprintf(line1,"ENERGY=%.3fkWh  ",res[daym][moum]);
    goto_xy(0,1);
    Puts(line1);
}

void showdata()
{
    daym=moum=1;
    InitLcd();
    while (inp&0x10)
    {
        inp=XBYTE[0x200];
        setshow();
    }
    while (!(inp&0x10))
    {
        inp=XBYTE[0x200];
    }
    while (inp&0x10)
    {
        inp=XBYTE[0x200];
        showvalue();
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

check=1;

}



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

char flag;
int time[8];

sbit  SCL = P1^0;
sbit  SDA = P3^4;

void delay()
{
    _nop_();
    _nop_();
    _nop_();
    _nop_();
}

void start()
{
    SDA = 1;
    SCL = 1;
    SDA = 0;
    delay();
    SCL = 0;
}

```



```
void stop()
```

```
{
```

```
    SDA = 0;
```

```
    SCL = 1;
```

```
    SDA = 1;
```

```
    delay();
```

```
}
```

```
void clow()
```

```
{
```

```
    SCL = 0;
```

```
    delay();
```

```
}
```

```
void chigh()
```

```
{
```

```
    SCL = 1;
```

```
    delay();
```

```
}
```

```
void write (unsigned char dat)
```

```
{
```

```
    unsigned char i;
```

```
    bit outbit;
```

```
    for (i=1;i<=8;i++)
```

```
    {
```

```
        outbit = dat & 0x80;
```

```
        SDA = outbit;
```

```
        dat = dat << 1;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        chigh();
        clow();
    }
    SDA = 1;
    chigh();
    outbit = SDA;
    clow();
//    return (outbit);
}

char read ()
{
    int i,dat;
    bit inbit;
    dat = 0;
    for (i=1;i<=8;i++)
    {
        chigh();
        inbit = SDA;
        dat = dat << 1;
        dat = dat | inbit;
        clow();
    }
    SDA = 0;
    if(flag ==1)
    {
        SDA =1;
    }
    chigh ();
    clow ();
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
SDA = 1;
return (dat);
}
```

```
void RTCread()
{
```

```
char a;
```

```
start();
```

```
write(0xD0);
```

```
write(0x00);
```

```
start();
```

```
write(0xD1);
```

```
flag = 0;
```

```
for(a=0;a<6;a++)
```

```
{ time[a] = read(); }
```

```
flag = 1;
```

```
time[a] = read();
```

```
stop();
```

```
}
```

```
void initRTC()
```

```
{
```

```
start();
```

```
write(0xD0);
```

```
write(0x07);
```

```
write(0x90);
```

```
stop();
```

```
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

LOGIN.C

```
xdata char user[11],pass[31];
char log;
char root1[]="sarayut",root2[]="wichit",root3[]="guest";
char word[]="solar";

void login()
{
    int a,b,c;
    char i,j,k,l;
    printf("\n\nSolarOS 6.2\n");
    //printf("\nPlease LOGIN\nuse 'guest' as LOGIN name and your email address as
password\n");
    do
    {
        log=0;
        a=b=-1;
        printf("\nlogin: ");
        do
        {
            if (a++<10) { user[a]=_getkey();*putchar(user[a]);*/}
            else { user[a]=0x0d; }
        }
        while(user[a]!=0x0d);user[a]=0;
        printf("\nPassword: ");
        do
        {
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

if (b++<30)
{
    pass[b]=_getkey();//putchar(pass[b]);
    putchar(0x08);
    putchar(0x20);
}
else { pass[b]=0x0d; }
}
while(pass[b]!=0x0d);pass[b]=0;
i=strcmp(user,root1);
j=strcmp(user,root2);
l=strcmp(user,root3);
if ((i&&j)==0)
{
    k=strcmp(pass,word);
    if (k==0) { log=1; }
    else { printf("\nLogin incorrect\n"); }
}
else
{
    if (l==0)
    {
        for(c=0;c<=b;c++)
        {
            if ((pass[c]!='@')||(pass[c]!='.')) { log=2; }
        }
        if (log==0) { printf("\nLogin incorrect\n"); }
    }
    else{ printf("\nLogin incorrect\n"); }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    }  
  }  
  while(log==0);  
}
```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

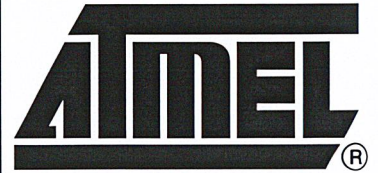
Features

- Compatible with MCS-51® Products
- 8K Bytes of In-System Programmable (ISP) Flash Memory
 - Endurance: 1000 Write/Erase Cycles
- 4.0V to 5.5V Operating Range
- Fully Static Operation: 0 Hz to 33 MHz
- Three-level Program Memory Lock
- 256 x 8-bit Internal RAM
- 32 Programmable I/O Lines
- Three 16-bit Timer/Counters
- Eight Interrupt Sources
- Full Duplex UART Serial Channel
- Low-power Idle and Power-down Modes
- Interrupt Recovery from Power-down Mode
- Watchdog Timer
- Dual Data Pointer
- Power-off Flag
- Fast Programming Time
- Flexible ISP Programming (Byte and Page Mode)

Description

The AT89S52 is a low-power, high-performance CMOS 8-bit microcontroller with 8K bytes of in-system programmable Flash memory. The device is manufactured using Atmel's high-density nonvolatile memory technology and is compatible with the industry-standard 80C51 instruction set and pinout. The on-chip 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 in-system programmable Flash on a monolithic chip, the Atmel AT89S52 is a powerful microcontroller which provides a highly-flexible and cost-effective solution to many embedded control applications.

The AT89S52 provides the following standard features: 8K bytes of Flash, 256 bytes of RAM, 32 I/O lines, Watchdog timer, two data pointers, three 16-bit timer/counters, a six-vector two-level interrupt architecture, a full duplex serial port, on-chip oscillator, and clock circuitry. In addition, the AT89S52 is designed with static logic for operation down to zero frequency and supports two software selectable power saving modes. The Idle Mode stops the CPU while allowing the RAM, timer/counters, serial port, and interrupt system to continue functioning. The Power-down mode saves the RAM contents but freezes the oscillator, disabling all other chip functions until the next interrupt or hardware reset.



**8-bit
Microcontroller
with 8K Bytes
In-System
Programmable
Flash**

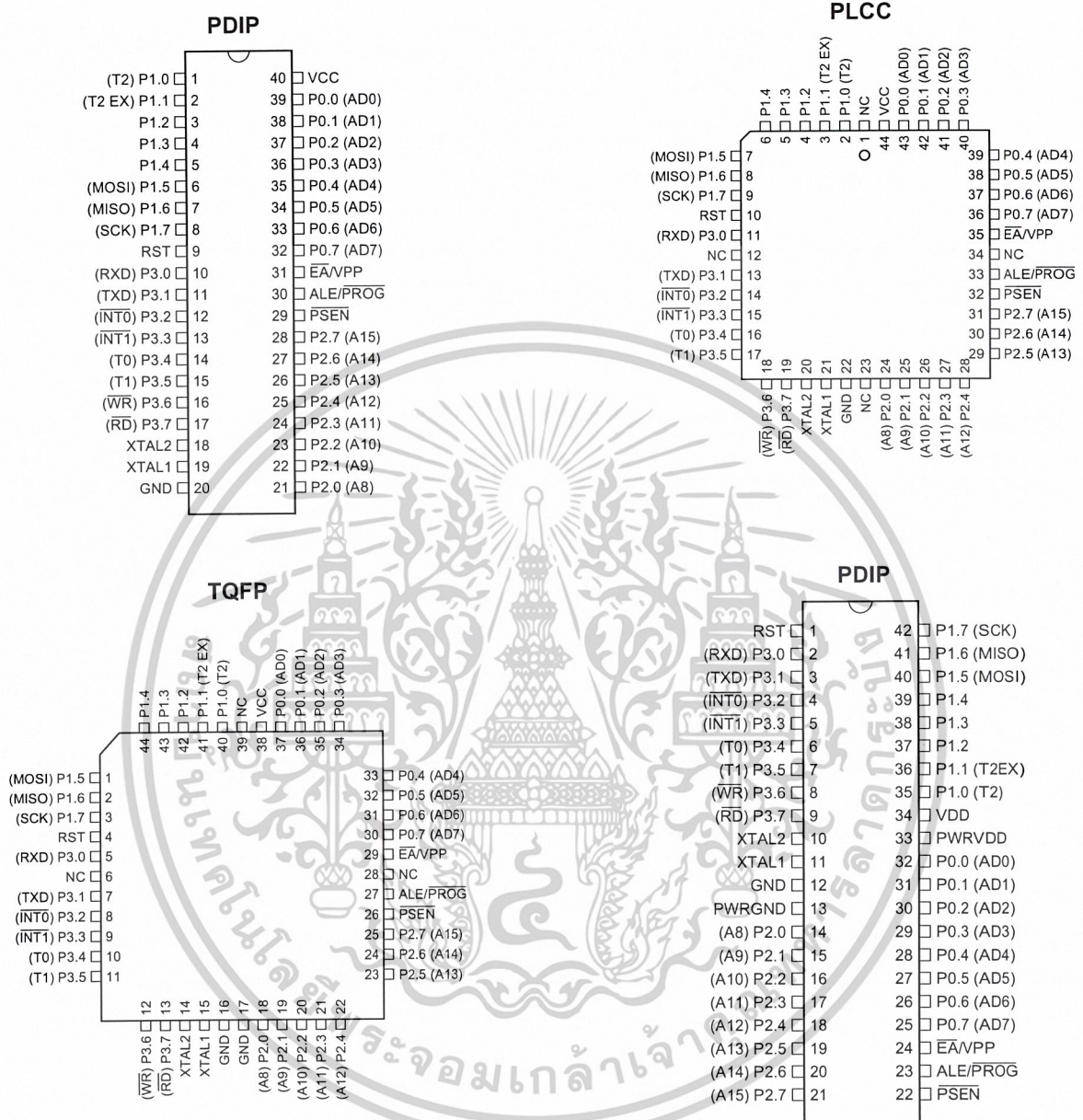
AT89S52

1919B-MICRO-11/03



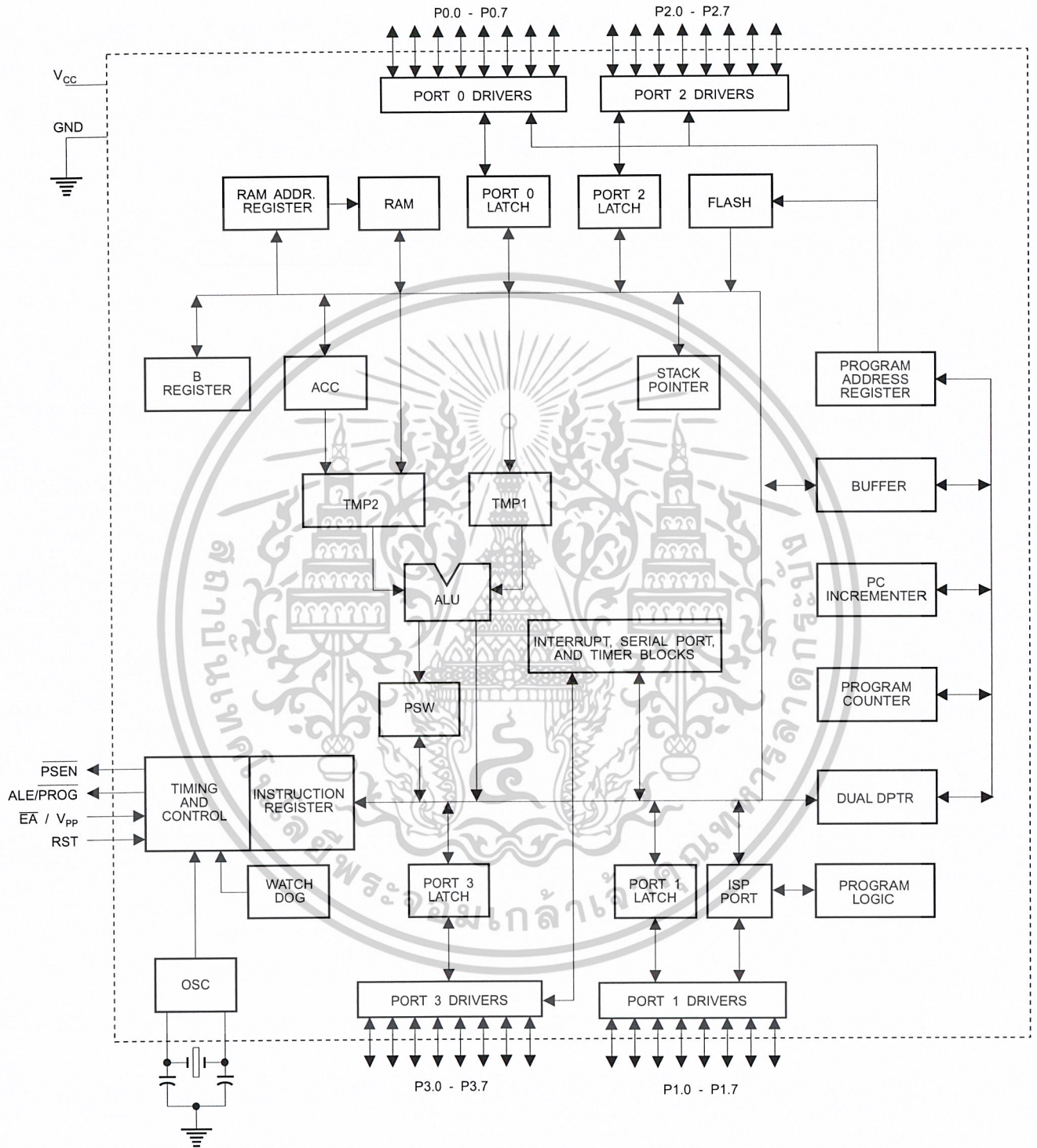
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Pin Configurations



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Block Diagram



Pin Description

- VCC** Supply voltage.
- GND** Ground.
- Port 0** Port 0 is an 8-bit open drain bidirectional I/O port. As an output port, each pin can sink eight TTL inputs. When 1s are written to port 0 pins, the pins can be used as high-impedance inputs.
- Port 0 can also be configured to be the multiplexed low-order address/data bus during accesses to external program and data memory. In this mode, P0 has internal pull-ups.
- Port 0 also receives the code bytes during Flash programming and outputs the code bytes during program verification. **External pull-ups are required during program verification.**
- Port 1** Port 1 is an 8-bit bidirectional I/O port with internal pull-ups. 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 pull-ups 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 pull-ups.
- In addition, P1.0 and P1.1 can be configured to be the timer/counter 2 external count input (P1.0/T2) and the timer/counter 2 trigger input (P1.1/T2EX), respectively, as shown in the following table.
- Port 1 also receives the low-order address bytes during Flash programming and verification.
- | Port Pin | Alternate Functions |
|----------|---|
| P1.0 | T2 (external count input to Timer/Counter 2), clock-out |
| P1.1 | T2EX (Timer/Counter 2 capture/reload trigger and direction control) |
| P1.5 | MOSI (used for In-System Programming) |
| P1.6 | MISO (used for In-System Programming) |
| P1.7 | SCK (used for In-System Programming) |
- Port 2** Port 2 is an 8-bit bidirectional I/O port with internal pull-ups. 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 pull-ups 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 pull-ups.
- Port 2 emits the high-order address byte during fetches from external program memory and during accesses to external data memory that use 16-bit addresses (MOVX @ DPTR). In this application, Port 2 uses strong internal pull-ups 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 pull-ups. 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 pull-ups 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 pull-ups.

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

Port 3 also serves the functions of various special features of the AT89S52, as shown in the following table.

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

- RST** Reset input. A high on this pin for two machine cycles while the oscillator is running resets the device. This pin drives high for 98 oscillator periods after the Watchdog times out. The DISRTO bit in SFR AUXR (address 8EH) can be used to disable this feature. In the default state of bit DISRTO, the RESET HIGH out feature is enabled.
- ALE/ $\overline{\text{PROG}}$** Address Latch Enable (ALE) is an output pulse for latching the low byte of the address during accesses to external memory. This pin is also the program pulse input ($\overline{\text{PROG}}$) 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.
- $\overline{\text{PSEN}}$** Program Store Enable ($\overline{\text{PSEN}}$) is the read strobe to external program memory.
When the AT89S52 is executing code from external program memory, $\overline{\text{PSEN}}$ is activated twice each machine cycle, except that two $\overline{\text{PSEN}}$ activations are skipped during each access to external data memory.
- $\overline{\text{EA/VPP}}$** External Access Enable. $\overline{\text{EA}}$ 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, $\overline{\text{EA}}$ will be internally latched on reset.
 $\overline{\text{EA}}$ 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.
- XTAL1** Input to the inverting oscillator amplifier and input to the internal clock operating circuit.
- XTAL2** Output from the inverting oscillator amplifier.



Special Function Registers

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

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

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

Timer 2 Registers: Control and status bits are contained in registers T2CON (shown in Table 2) and T2MOD (shown in Table 6) for Timer 2. The register pair (RCAP2H, RCAP2L) are the Capture/Reload registers for Timer 2 in 16-bit capture mode or 16-bit auto-reload mode.

Interrupt Registers: The individual interrupt enable bits are in the IE register. Two priorities can be set for each of the six interrupt sources in the IP register.

Table 1. AT89S52 SFR Map and Reset Values

0F8H								0FFH
0F0H	B 00000000							0F7H
0E8H								0EFH
0E0H	ACC 00000000							0E7H
0D8H								0DFH
0D0H	PSW 00000000							0D7H
0C8H	T2CON 00000000	T2MOD XXXXXX00	RCAP2L 00000000	RCAP2H 00000000	TL2 00000000	TH2 00000000		0CFH
0C0H								0C7H
0B8H	IP XX000000							0BFH
0B0H	P3 11111111							0B7H
0A8H	IE 0X000000							0AFH
0A0H	P2 11111111		AUXR1 XXXXXXXX0				WDRST XXXXXXXX	0A7H
98H	SCON 00000000	SBUF XXXXXXXX						9FH
90H	P1 11111111							97H
88H	TCON 00000000	TMOD 00000000	TL0 00000000	TL1 00000000	TH0 00000000	TH1 00000000	AUXR XXX0XX0	8FH
80H	P0 11111111	SP 0000111	DP0L 00000000	DP0H 00000000	DP1L 00000000	DP1H 00000000	PCON 0XXX0000	87H

Table 2. T2CON – Timer/Counter 2 Control Register

T2CON Address = 0C8H						Reset Value = 0000 0000B		
Bit Addressable								
Bit	TF2	EXF2	RCLK	TCLK	EXEN2	TR2	C/T $\bar{2}$	CP/RL $\bar{2}$
	7	6	5	4	3	2	1	0

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

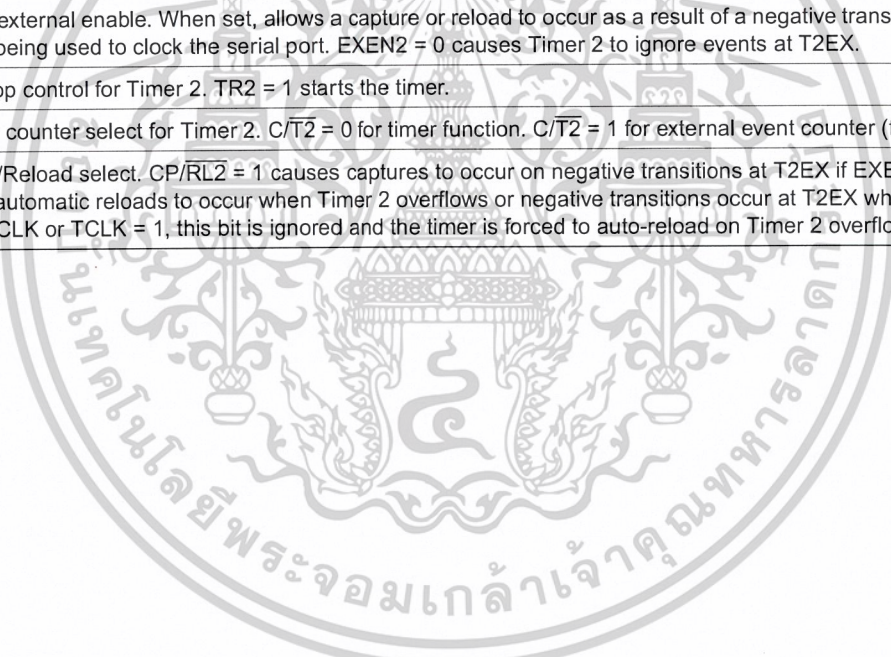


Table 3. AUXR: Auxiliary Register

AUXR	Address = 8EH	Reset Value = XXX00XX0B							
	Not Bit Addressable								
		–	–	–	WDIDLE	DISRTO	–	–	DISALE
Bit		7	6	5	4	3	2	1	0
–	Reserved for future expansion								
DISALE	Disable/Enable ALE								
	DISALE	Operating Mode							
	0	ALE is emitted at a constant rate of 1/6 the oscillator frequency							
	1	ALE is active only during a MOVX or MOVC instruction							
DISRTO	Disable/Enable Reset out								
	DISRTO								
	0	Reset pin is driven High after WDT times out							
	1	Reset pin is input only							
WDIDLE	Disable/Enable WDT in IDLE mode								
	WDIDLE								
	0	WDT continues to count in IDLE mode							
	1	WDT halts counting in IDLE mode							

Dual Data Pointer Registers: To facilitate accessing both internal and external data memory, two banks of 16-bit Data Pointer Registers are provided: DP0 at SFR address locations 82H-83H and DP1 at 84H-85H. Bit DPS = 0 in SFR AUXR1 selects DP0 and DPS = 1 selects DP1. The user should **ALWAYS** initialize the DPS bit to the appropriate value before accessing the respective Data Pointer Register.

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

Table 4. AUXR1: Auxiliary Register 1

AUXR1	Address = A2H	Reset Value = XXXXXXX0B							
	Not Bit Addressable								
		–	–	–	–	–	–	DPS	
Bit		7	6	5	4	3	2	1	0
–	Reserved for future expansion								
DPS	Data Pointer Register Select								
	DPS								
	0	Selects DPTR Registers DP0L, DP0H							
	1	Selects DPTR Registers DP1L, DP1H							

Memory Organization MCS-51 devices have a separate address space for Program and Data Memory. Up to 64K bytes each of external Program and Data Memory can be addressed.

Program Memory If the \overline{EA} pin is connected to GND, all program fetches are directed to external memory. On the AT89S52, if \overline{EA} is connected to V_{CC} , program fetches to addresses 0000H through 1FFFH are directed to internal memory and fetches to addresses 2000H through FFFFH are to external memory.

Data Memory The AT89S52 implements 256 bytes of on-chip RAM. The upper 128 bytes occupy a parallel address space to the Special Function Registers. This means that the upper 128 bytes have the same addresses as the SFR space but are physically separate from SFR space.

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

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

```
MOV 0A0H, #data
```

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

```
MOV @R0, #data
```

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

Watchdog Timer (One-time Enabled with Reset-out) The WDT is intended as a recovery method in situations where the CPU may be subjected to software upsets. The WDT consists of a 14-bit counter and the Watchdog Timer Reset (WDTRST) SFR. The WDT is defaulted to disable from exiting reset. To enable the WDT, a user must write 01EH and 0E1H in sequence to the WDTRST register (SFR location 0A6H). When the WDT is enabled, it will increment every machine cycle while the oscillator is running. The WDT timeout period is dependent on the external clock frequency. There is no way to disable the WDT except through reset (either hardware reset or WDT overflow reset). When WDT overflows, it will drive an output RESET HIGH pulse at the RST pin.

Using the WDT To enable the WDT, a user must write 01EH and 0E1H in sequence to the WDTRST register (SFR location 0A6H). When the WDT is enabled, the user needs to service it by writing 01EH and 0E1H to WDTRST to avoid a WDT overflow. The 14-bit counter overflows when it reaches 16383 (3FFFH), and this will reset the device. When the WDT is enabled, it will increment every machine cycle while the oscillator is running. This means the user must reset the WDT at least every 16383 machine cycles. To reset the WDT the user must write 01EH and 0E1H to WDTRST. WDTRST is a write-only register. The WDT counter cannot be read or written. When WDT overflows, it will generate an output RESET pulse at the RST pin. The RESET pulse duration is $98 \times T_{OSC}$, where $T_{OSC} = 1/F_{OSC}$. To make the best use of the WDT, it should be serviced in those sections of code that will periodically be executed within the time required to prevent a WDT reset.



WDT During Power-down and Idle

In Power-down mode the oscillator stops, which means the WDT also stops. While in Power-down mode, the user does not need to service the WDT. There are two methods of exiting Power-down mode: by a hardware reset or via a level-activated external interrupt which is enabled prior to entering Power-down mode. When Power-down is exited with hardware reset, servicing the WDT should occur as it normally does whenever the AT89S52 is reset. Exiting Power-down with an interrupt is significantly different. The interrupt is held low long enough for the oscillator to stabilize. When the interrupt is brought high, the interrupt is serviced. To prevent the WDT from resetting the device while the interrupt pin is held low, the WDT is not started until the interrupt is pulled high. It is suggested that the WDT be reset during the interrupt service for the interrupt used to exit Power-down mode.

To ensure that the WDT does not overflow within a few states of exiting Power-down, it is best to reset the WDT just before entering Power-down mode.

Before going into the IDLE mode, the WDIDLE bit in SFR AUXR is used to determine whether the WDT continues to count if enabled. The WDT keeps counting during IDLE (WDIDLE bit = 0) as the default state. To prevent the WDT from resetting the AT89S52 while in IDLE mode, the user should always set up a timer that will periodically exit IDLE, service the WDT, and reenter IDLE mode.

With WDIDLE bit enabled, the WDT will stop to count in IDLE mode and resumes the count upon exit from IDLE.

UART

The UART in the AT89S52 operates the same way as the UART in the AT89C51 and AT89C52. For further information on the UART operation, refer to the ATMEL Web site (<http://www.atmel.com>). From the home page, select "Products", then "8051-Architecture Flash Microcontroller", then "Product Overview".

Timer 0 and 1

Timer 0 and Timer 1 in the AT89S52 operate the same way as Timer 0 and Timer 1 in the AT89C51 and AT89C52. For further information on the timers' operation, refer to the ATMEL Web site (<http://www.atmel.com>). From the home page, select "Products", then "8051-Architecture Flash Microcontroller", then "Product Overview".

Timer 2

Timer 2 is a 16-bit Timer/Counter that can operate as either a timer or an event counter. The type of operation is selected by bit C/T2 in the SFR T2CON (shown in Table 2). Timer 2 has three operating modes: capture, auto-reload (up or down counting), and baud rate generator. The modes are selected by bits in T2CON, as shown in Table 5. Timer 2 consists of two 8-bit registers, TH2 and TL2. In the Timer function, the TL2 register is incremented every machine cycle. Since a machine cycle consists of 12 oscillator periods, the count rate is 1/12 of the oscillator frequency.

Table 5. Timer 2 Operating Modes

RCLK +TCLK	CP/RL2	TR2	MODE
0	0	1	16-bit Auto-reload
0	1	1	16-bit Capture
1	X	1	Baud Rate Generator
X	X	0	(Off)

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

Capture Mode

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

Auto-reload (Up or Down Counter)

Timer 2 can be programmed to count up or down when configured in its 16-bit auto-reload mode. This feature is invoked by the DCEN (Down Counter Enable) bit located in the SFR T2MOD (see Table 6). Upon reset, the DCEN bit is set to 0 so that timer 2 will default to count up. When DCEN is set, Timer 2 can count up or down, depending on the value of the T2EX pin.

Figure 1. Timer in Capture Mode

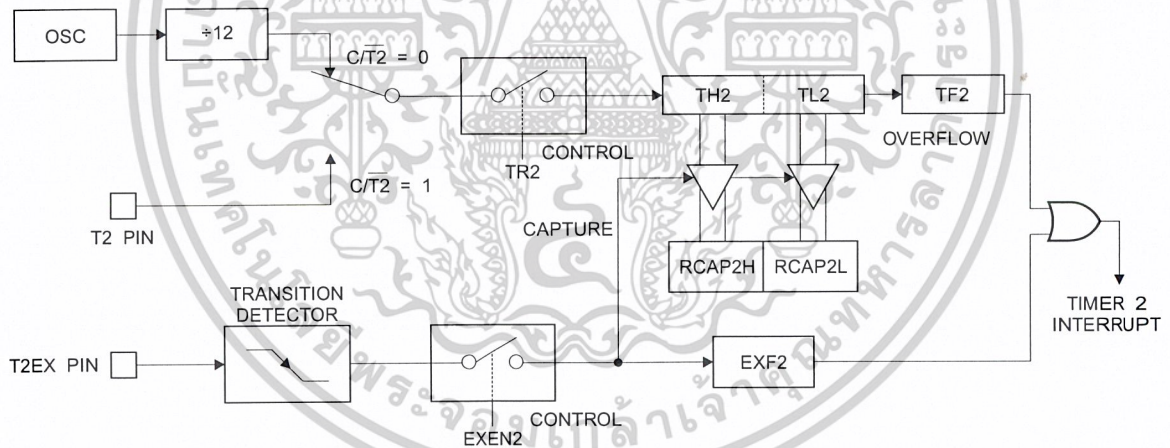


Figure 2 shows Timer 2 automatically counting up when DCEN = 0. In this mode, two options are selected by bit EXEN2 in T2CON. If EXEN2 = 0, Timer 2 counts up to 0FFFFH and then sets the TF2 bit upon overflow. The overflow also causes the timer registers to be reloaded with the 16-bit value in RCAP2H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

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

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

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

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

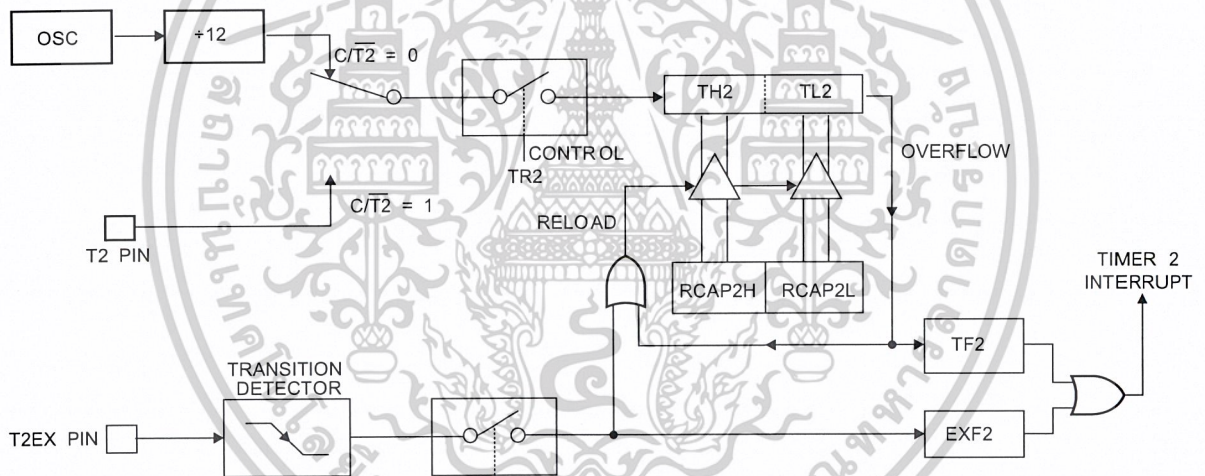
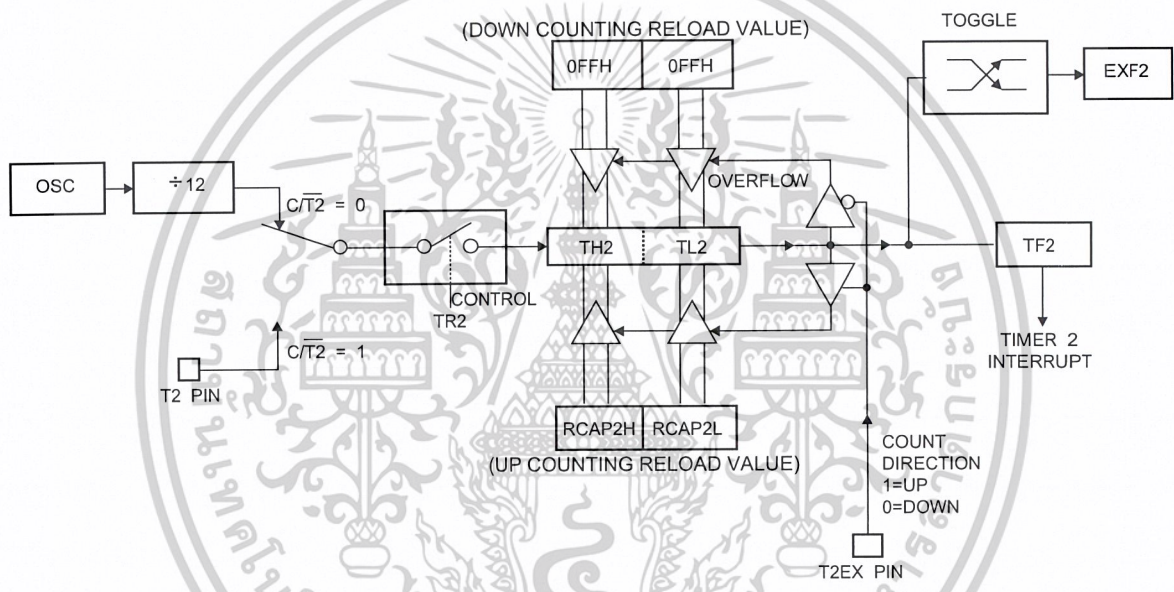


Table 6. T2MOD – Timer 2 Mode Control Register

T2MOD Address = 0C9H						Reset Value = XXXX XX00B		
Not Bit Addressable								
Bit	7	6	5	4	3	2	T2OE	DCEN
	-	-	-	-	-	-	1	0

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

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





Baud Rate Generator

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

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

The baud rates in Modes 1 and 3 are determined by Timer 2's overflow rate according to the following equation.

$$\text{Modes 1 and 3 Baud Rates} = \frac{\text{Timer 2 Overflow Rate}}{16}$$

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

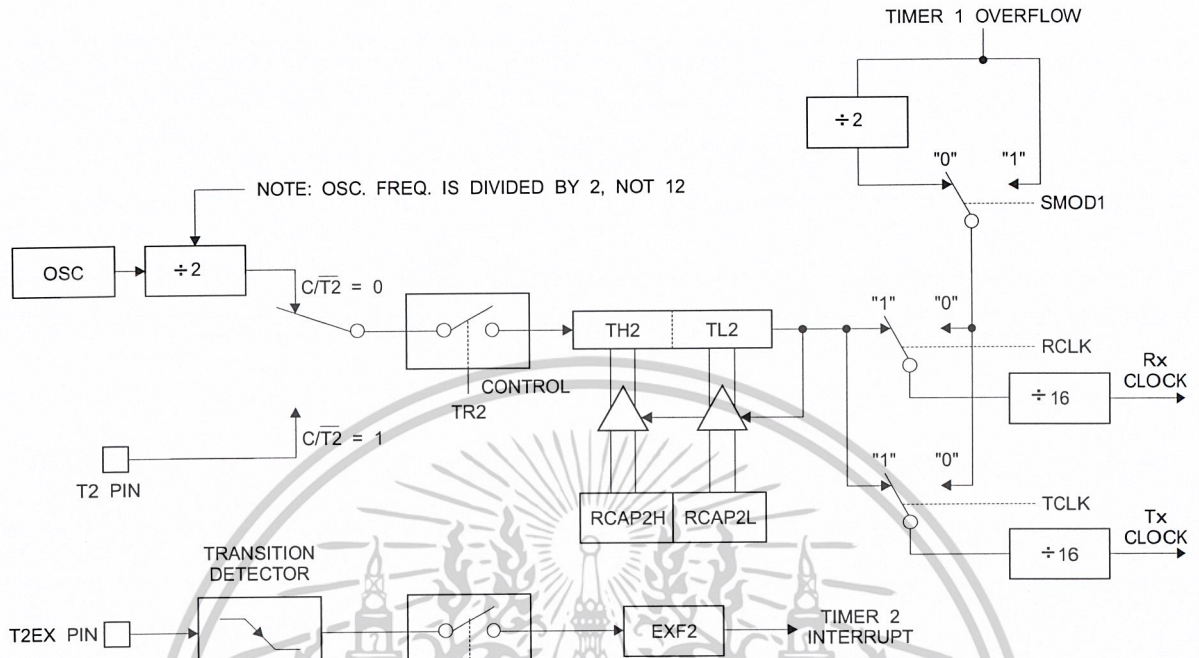
$$\frac{\text{Modes 1 and 3 Baud Rate}}{\text{Baud Rate}} = \frac{\text{Oscillator Frequency}}{32 \times [65536 - \text{RCAP2H}, \text{RCAP2L}]}$$

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

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

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

Figure 4. Timer 2 in Baud Rate Generator Mode



Programmable Clock Out

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

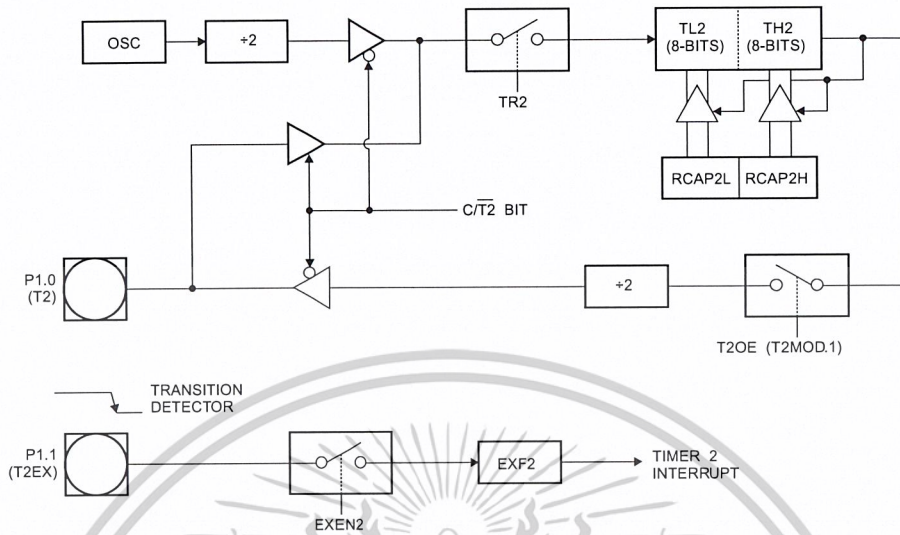
To configure the Timer/Counter 2 as a clock generator, bit C/T2 (T2CON.1) must be cleared and bit T2OE (T2MOD.1) must be set. Bit TR2 (T2CON.2) starts and stops the timer.

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

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

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

Figure 5. Timer 2 in Clock-Out Mode



Interrupts

The AT89S52 has a total of six interrupt vectors: two external interrupts ($\overline{INT0}$ and $\overline{INT1}$), three timer interrupts (Timers 0, 1, and 2), and the serial port interrupt. These interrupts are all shown in Figure 6.

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

Note that Table 5 shows that bit position IE.6 is unimplemented. User software should not write a 1 to this bit position, since it may be used in future AT89 products.

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

The Timer 0 and Timer 1 flags, TF0 and TF1, are set at S5P2 of the cycle in which the timers overflow. The values are then polled by the circuitry in the next cycle. However, the Timer 2 flag, TF2, is set at S2P2 and is polled in the same cycle in which the timer overflows.

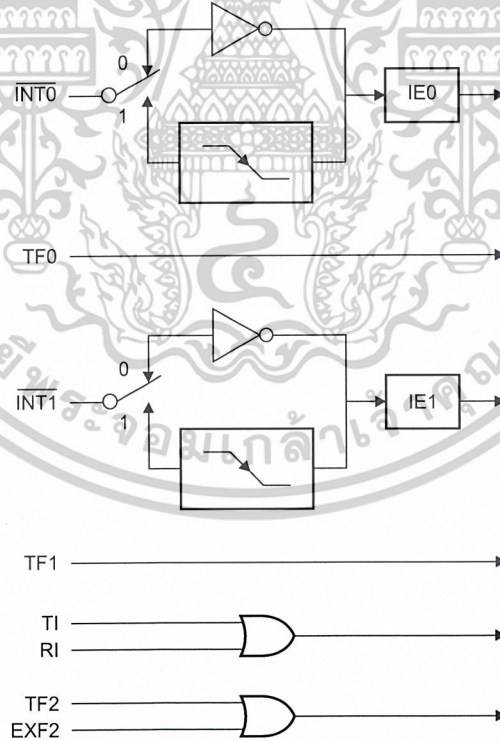
Table 7. Interrupt Enable (IE) Register

(MSB)		(LSB)					
EA	-	ET2	ES	ET1	EX1	ET0	EX0
Enable Bit = 1 enables the interrupt.							
Enable Bit = 0 disables the interrupt.							

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

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

Figure 6. Interrupt Sources



Oscillator Characteristics

XTAL1 and XTAL2 are the input and output, respectively, of an inverting amplifier that can be configured for use as an on-chip oscillator, as shown in Figure 7. 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 8. 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

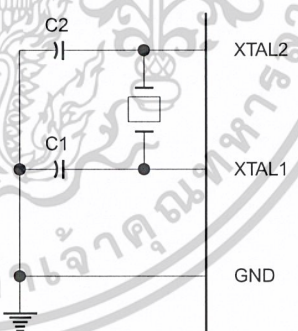
In idle mode, the CPU puts itself to sleep while all the on-chip peripherals remain active. The mode is invoked by software. The content of the on-chip RAM and all the special functions registers remain unchanged during this mode. The idle mode can be terminated by any enabled interrupt or by a hardware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

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

Power-down Mode

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

Figure 7. Oscillator Connections



- Note: 1. C1, C2 = 30 pF \pm 10 pF for Crystals
 = 40 pF \pm 10 pF for Ceramic Resonators

Figure 8. External Clock Drive Configuration

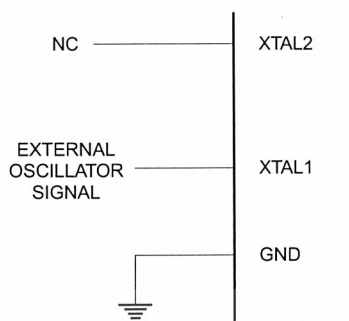


Table 8. Status of External Pins During Idle and Power-down Modes

Mode	Program Memory	ALE	$\overline{\text{PSEN}}$	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

Program Memory Lock Bits

The AT89S52 has three lock bits that can be left unprogrammed (U) or can be programmed (P) to obtain the additional features listed in the following table.

Table 9. Lock Bit Protection Modes

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

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



Programming the Flash – Parallel Mode

The AT89S52 is shipped with the on-chip Flash memory array ready to be programmed. The programming interface needs a high-voltage (12-volt) program enable signal and is compatible with conventional third-party Flash or EPROM programmers.

The AT89S52 code memory array is programmed byte-by-byte.

Programming Algorithm: Before programming the AT89S52, the address, data, and control signals should be set up according to the Flash programming mode table and Figures 13 and 14. To program the AT89S52, 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.
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 50 μ s. Repeat steps 1 through 5, changing the address and data for the entire array or until the end of the object file is reached.

Data Polling: The AT89S52 features Data Polling to indicate the end of a byte write cycle. During a write cycle, an attempted read of the last byte written will result in the complement of the written data on P0.7. Once the write cycle has been completed, true data is valid on all outputs, and the next cycle may begin. Data Polling may begin any time after a write cycle has been initiated.

Ready/Busy: The progress of byte programming can also be monitored by the RDY/BSY output signal. P3.0 is pulled low after ALE goes high during programming to indicate BUSY. P3.0 is pulled high again when programming is done to indicate READY.

Program Verify: If lock bits LB1 and LB2 have not been programmed, the programmed code data can be read back via the address and data lines for verification. The status of the individual lock bits can be verified directly by reading them back.

Reading the Signature Bytes: The signature bytes are read by the same procedure as a normal verification of locations 000H, 100H, and 200H, except that P3.6 and P3.7 must be pulled to a logic low. The values returned are as follows.

- (000H) = 1EH indicates manufactured by Atmel
- (100H) = 52H indicates AT89S52
- (200H) = 06H

Chip Erase: In the parallel programming mode, a chip erase operation is initiated by using the proper combination of control signals and by pulsing ALE/\overline{PROG} low for a duration of 200 ns - 500 ns.

In the serial programming mode, a chip erase operation is initiated by issuing the Chip Erase instruction. In this mode, chip erase is self-timed and takes about 500 ms.

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

Programming the Flash – Serial Mode

The Code memory array can be programmed using the serial ISP interface while RST is pulled to V_{CC} . The serial interface consists of pins SCK, MOSI (input) and MISO (output). After RST is set high, the Programming Enable instruction needs to be executed first before other operations can be executed. Before a reprogramming sequence can occur, a Chip Erase operation is required.

The Chip Erase operation turns the content of every memory location in the Code array into FFH.

Either an external system clock can be supplied at pin XTAL1 or a crystal needs to be connected across pins XTAL1 and XTAL2. The maximum serial clock (SCK) frequency should be less than 1/16 of the crystal frequency. With a 33 MHz oscillator clock, the maximum SCK frequency is 2 MHz.

Serial Programming Algorithm

To program and verify the AT89S52 in the serial programming mode, the following sequence is recommended:

1. Power-up sequence:
 - Apply power between V_{CC} and GND pins.
 - Set RST pin to "H".
 - If a crystal is not connected across pins XTAL1 and XTAL2, apply a 3 MHz to 33 MHz clock to XTAL1 pin and wait for at least 10 milliseconds.
2. Enable serial programming by sending the Programming Enable serial instruction to pin MOSI/P1.5. The frequency of the shift clock supplied at pin SCK/P1.7 needs to be less than the CPU clock at XTAL1 divided by 16.
3. The Code array is programmed one byte at a time in either the Byte or Page mode. The write cycle is self-timed and typically takes less than 0.5 ms at 5V.
4. Any memory location can be verified by using the Read instruction which returns the content at the selected address at serial output MISO/P1.6.
5. At the end of a programming session, RST can be set low to commence normal device operation.

Power-off sequence (if needed):

- Set XTAL1 to "L" (if a crystal is not used).
- Set RST to "L".
- Turn V_{CC} power off.

Data Polling: The Data Polling feature is also available in the serial mode. In this mode, during a write cycle an attempted read of the last byte written will result in the complement of the MSB of the serial output byte on MISO.



Serial Programming Instruction Set

The Instruction Set for Serial Programming follows a 4-byte protocol and is shown in Table 11.

Programming Interface – Parallel Mode

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

Most worldwide major programming vendors offer support for the Atmel AT89 microcontroller series. Please contact your local programming vendor for the appropriate software revision.

Table 10. Flash Programming Modes

Mode	V _{CC}	RST	PSEN	ALE/ PROG	EA/ V _{PP}	P2.6	P2.7	P3.3	P3.6	P3.7	P0.7-0 Data	P2.4-0	P1.7-0
												Address	
Write Code Data	5V	H	L		12V	L	H	H	H	H	D _{IN}	A12-8	A7-0
Read Code Data	5V	H	L	H	H	L	L	L	H	H	D _{OUT}	A12-8	A7-0
Write Lock Bit 1	5V	H	L		12V	H	H	H	H	H	X	X	X
Write Lock Bit 2	5V	H	L		12V	H	H	H	L	L	X	X	X
Write Lock Bit 3	5V	H	L		12V	H	L	H	H	L	X	X	X
Read Lock Bits 1, 2, 3	5V	H	L	H	H	H	H	L	H	L	P0.2, P0.3, P0.4	X	X
Chip Erase	5V	H	L		12V	H	L	H	L	L	X	X	X
Read Atmel ID	5V	H	L	H	H	L	L	L	L	L	1EH	X 0000	00H
Read Device ID	5V	H	L	H	H	L	L	L	L	L	52H	X 0001	00H
Read Device ID	5V	H	L	H	H	L	L	L	L	L	06H	X 0010	00H

- Notes:
1. Each PROG pulse is 200 ns - 500 ns for Chip Erase.
 2. Each PROG pulse is 200 ns - 500 ns for Write Code Data.
 3. Each PROG pulse is 200 ns - 500 ns for Write Lock Bits.
 4. RDY/BSY signal is output on P3.0 during programming.
 5. X = don't care.

Figure 9. Programming the Flash Memory (Parallel Mode)

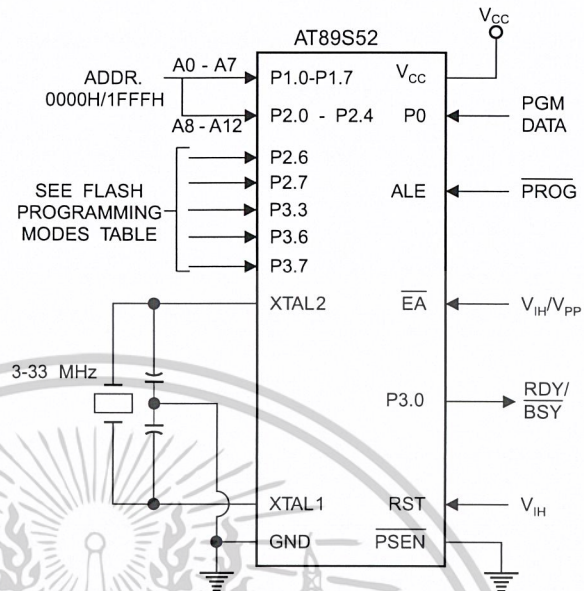
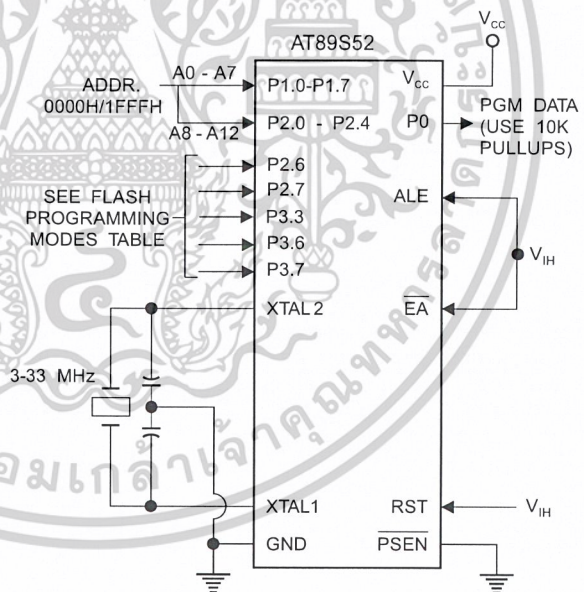


Figure 10. Verifying the Flash Memory (Parallel Mode)



Flash Programming and Verification Characteristics (Parallel Mode)

$T_A = 20^\circ\text{C}$ to 30°C , $V_{CC} = 4.5$ to 5.5V

Symbol	Parameter	Min	Max	Units
V_{PP}	Programming Supply Voltage	11.5	12.5	V
I_{PP}	Programming Supply Current		10	mA
I_{CC}	V_{CC} Supply Current		30	mA
$1/t_{CLCL}$	Oscillator Frequency	3	33	MHz
t_{AVGL}	Address Setup to $\overline{\text{PROG}}$ Low	$48t_{CLCL}$		
t_{GHAX}	Address Hold After $\overline{\text{PROG}}$	$48t_{CLCL}$		
t_{DVGL}	Data Setup to $\overline{\text{PROG}}$ Low	$48t_{CLCL}$		
t_{GHDX}	Data Hold After $\overline{\text{PROG}}$	$48t_{CLCL}$		
t_{EHS}	P2.7 ($\overline{\text{ENABLE}}$) High to V_{PP}	$48t_{CLCL}$		
t_{SHGL}	V_{PP} Setup to $\overline{\text{PROG}}$ Low	10		μs
t_{GHSL}	V_{PP} Hold After $\overline{\text{PROG}}$	10		μs
t_{GLGH}	$\overline{\text{PROG}}$ Width	0.2	1	μs
t_{AVQV}	Address to Data Valid		$48t_{CLCL}$	
t_{ELQV}	$\overline{\text{ENABLE}}$ Low to Data Valid		$48t_{CLCL}$	
t_{EHQZ}	Data Float After $\overline{\text{ENABLE}}$	0	$48t_{CLCL}$	
t_{GHBL}	$\overline{\text{PROG}}$ High to $\overline{\text{BUSY}}$ Low		1.0	μs
t_{WC}	Byte Write Cycle Time		50	μs

Figure 11. Flash Programming and Verification Waveforms – Parallel Mode

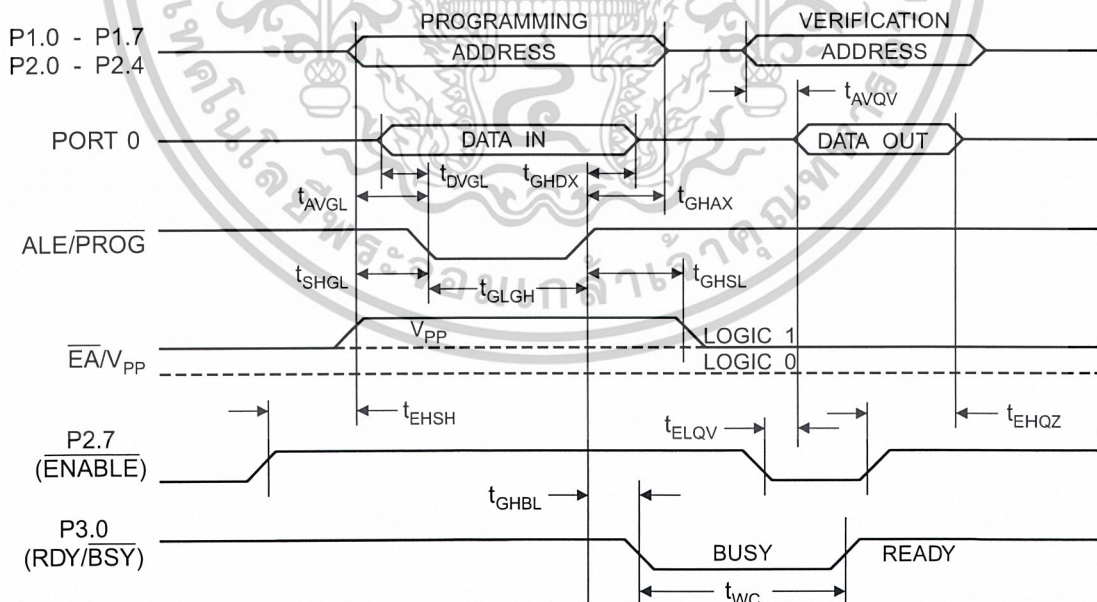


Figure 12. Flash Memory Serial Downloading

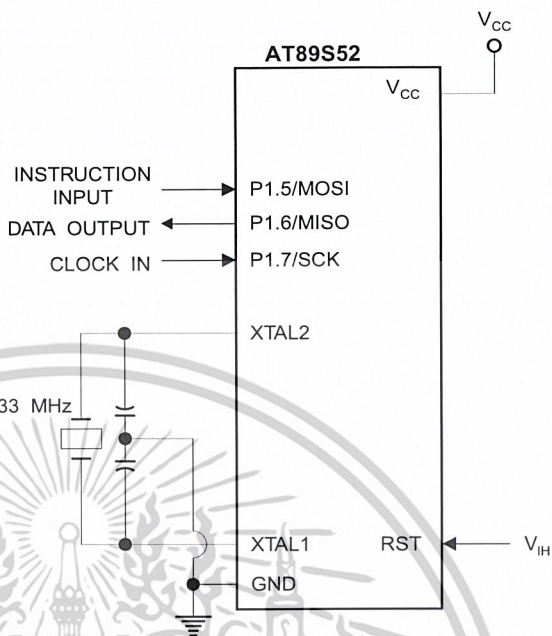
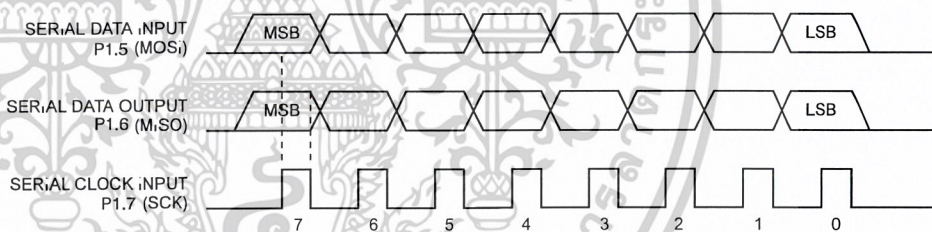


Figure 13. Serial Programming Waveforms



Flash Programming and Verification Waveforms – Serial Mode



Table 11. Serial Programming Instruction Set

Instruction	Instruction Format				Operation
	Byte 1	Byte 2	Byte 3	Byte 4	
Programming Enable	1010 1100	0101 0011	xxxx xxxx	xxxx xxxx 0110 1001 (Output on MISO)	Enable Serial Programming while RST is high
Chip Erase	1010 1100	100x xxxx	xxxx xxxx	xxxx xxxx	Chip Erase Flash memory array
Read Program Memory (Byte Mode)	0010 0000	xxx A12 A11 A10 A9 A8	A7 A6 A5 A4 A3 A2 A1 A0	D7 D6 D5 D4 D3 D2 D1 D0	Read data from Program memory in the byte mode
Write Program Memory (Byte Mode)	0100 0000	xxx A12 A11 A10 A9 A8	A7 A6 A5 A4 A3 A2 A1 A0	D7 D6 D5 D4 D3 D2 D1 D0	Write data to Program memory in the byte mode
Write Lock Bits ⁽¹⁾	1010 1100	1110 00 B1 B2	xxxx xxxx	xxxx xxxx	Write Lock bits. See Note (1).
Read Lock Bits	0010 0100	xxxx xxxx	xxxx xxxx	xxx LB3 LB2 LB1 xx	Read back current status of the lock bits (a programmed lock bit reads back as a "1")
Read Signature Bytes	0010 1000	xxx A12 A11 A10 A9 A8	A7 xxx xxx0	Signature Byte	Read Signature Byte
Read Program Memory (Page Mode)	0011 0000	xxx A12 A11 A10 A9 A8	Byte 0	Byte 1... Byte 255	Read data from Program memory in the Page Mode (256 bytes)
Write Program Memory (Page Mode)	0101 0000	xxx A12 A11 A10 A9 A8	Byte 0	Byte 1... Byte 255	Write data to Program memory in the Page Mode (256 bytes)

Note: 1. B1 = 0, B2 = 0 ---> Mode 1, no lock protection
 B1 = 0, B2 = 1 ---> Mode 2, lock bit 1 activated
 B1 = 1, B2 = 0 ---> Mode 3, lock bit 2 activated
 B1 = 1, B2 = 1 ---> Mode 4, lock bit 3 activated

Each of the lock bit modes needs to be activated sequentially before Mode 4 can be executed.

After Reset signal is high, SCK should be low for at least 64 system clocks before it goes high to clock in the enable data bytes. No pulsing of Reset signal is necessary. SCK should be no faster than 1/16 of the system clock at XTAL1.

For Page Read/Write, the data always starts from byte 0 to 255. After the command byte and upper address byte are latched, each byte thereafter is treated as data until all 256 bytes are shifted in/out. Then the next instruction will be ready to be decoded.

Serial Programming Characteristics

Figure 14. Serial Programming Timing

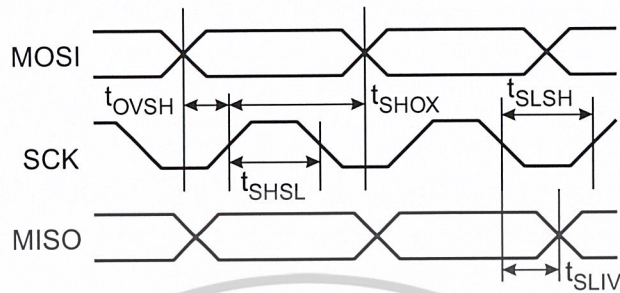


Table 12. Serial Programming Characteristics, $T_A = -40^\circ\text{C}$ to 85°C , $V_{CC} = 4.0 - 5.5\text{V}$ (Unless Otherwise Noted)

Symbol	Parameter	Min	Typ	Max	Units
$1/t_{CLCL}$	Oscillator Frequency	3		33	MHz
t_{CLCL}	Oscillator Period	30			ns
t_{SHSL}	SCK Pulse Width High	$8 t_{CLCL}$			ns
t_{SLSH}	SCK Pulse Width Low	$8 t_{CLCL}$			ns
t_{OVSH}	MOSI Setup to SCK High	t_{CLCL}			ns
t_{SHOX}	MOSI Hold after SCK High	$2 t_{CLCL}$			ns
t_{SLIV}	SCK Low to MISO Valid	10	16	32	ns
t_{ERASE}	Chip Erase Instruction Cycle Time			500	ms
t_{SWC}	Serial Byte Write Cycle Time			$64 t_{CLCL} + 400$	μs





Absolute Maximum Ratings*

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

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

DC Characteristics

The values shown in this table are valid for $T_A = -40^\circ\text{C}$ to 85°C and $V_{CC} = 4.0\text{V}$ to 5.5V , unless otherwise noted.

Symbol	Parameter	Condition	Min	Max	Units
V_{IL}	Input Low Voltage	(Except \overline{EA})	-0.5	$0.2 V_{CC} - 0.1$	V
V_{IL1}	Input Low Voltage (\overline{EA})		-0.5	$0.2 V_{CC} - 0.3$	V
V_{IH}	Input High Voltage	(Except XTAL1, RST)	$0.2 V_{CC} + 0.9$	$V_{CC} + 0.5$	V
V_{IH1}	Input High Voltage	(XTAL1, RST)	$0.7 V_{CC}$	$V_{CC} + 0.5$	V
V_{OL}	Output Low Voltage ⁽¹⁾ (Ports 1,2,3)	$I_{OL} = 1.6 \text{ mA}$		0.45	V
V_{OL1}	Output Low Voltage ⁽¹⁾ (Port 0, ALE, PSEN)	$I_{OL} = 3.2 \text{ mA}$		0.45	V
V_{OH}	Output High Voltage (Ports 1,2,3, ALE, PSEN)	$I_{OH} = -60 \mu\text{A}, V_{CC} = 5\text{V} \pm 10\%$	2.4		V
		$I_{OH} = -25 \mu\text{A}$	$0.75 V_{CC}$		V
		$I_{OH} = -10 \mu\text{A}$	$0.9 V_{CC}$		V
V_{OH1}	Output High Voltage (Port 0 in External Bus Mode)	$I_{OH} = -800 \mu\text{A}, V_{CC} = 5\text{V} \pm 10\%$	2.4		V
		$I_{OH} = -300 \mu\text{A}$	$0.75 V_{CC}$		V
		$I_{OH} = -80 \mu\text{A}$	$0.9 V_{CC}$		V
I_{IL}	Logical 0 Input Current (Ports 1,2,3)	$V_{IN} = 0.45\text{V}$		-50	μA
I_{TL}	Logical 1 to 0 Transition Current (Ports 1,2,3)	$V_{IN} = 2\text{V}, V_{CC} = 5\text{V} \pm 10\%$		-650	μA
I_{LI}	Input Leakage Current (Port 0, \overline{EA})	$0.45 < V_{IN} < V_{CC}$		± 10	μA
RRST	Reset Pulldown Resistor		50	300	$\text{K}\Omega$
C_{IO}	Pin Capacitance	Test Freq. = 1 MHz, $T_A = 25^\circ\text{C}$		10	pF
I_{CC}	Power Supply Current	Active Mode, 12 MHz		25	mA
		Idle Mode, 12 MHz		6.5	mA
	Power-down Mode ⁽¹⁾	$V_{CC} = 5.5\text{V}$		50	μA

Notes: 1. Under steady state (non-transient) conditions, I_{OL} must be externally limited as follows:

Maximum I_{OL} per port pin: 10 mA

Maximum I_{OL} per 8-bit port:

Port 0: 26 mA Ports 1, 2, 3: 15 mA

Maximum total I_{OL} for all output pins: 71 mA

If I_{OL} exceeds the test condition, V_{OL} may exceed the related specification. Pins are not guaranteed to sink current greater than the listed test conditions.

2. Minimum V_{CC} for Power-down is 2V.

AC Characteristics

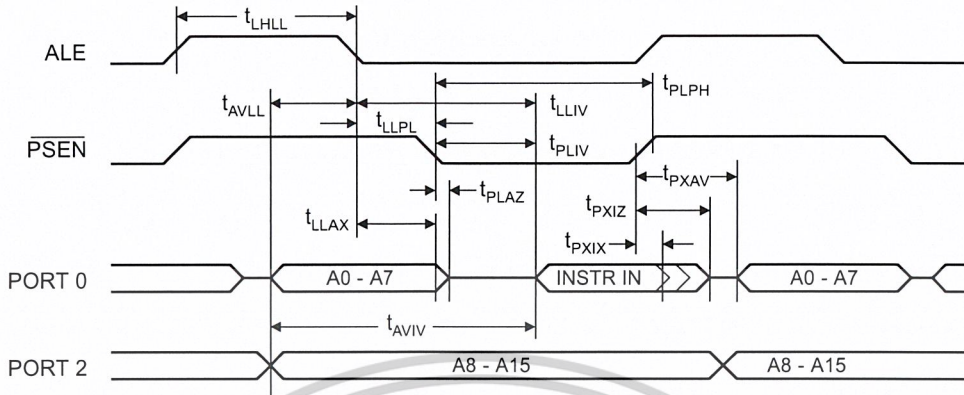
Under operating conditions, load capacitance for Port 0, ALE/ $\overline{\text{PROG}}$, and $\overline{\text{PSEN}}$ = 100 pF; load capacitance for all other outputs = 80 pF.

External Program and Data Memory Characteristics

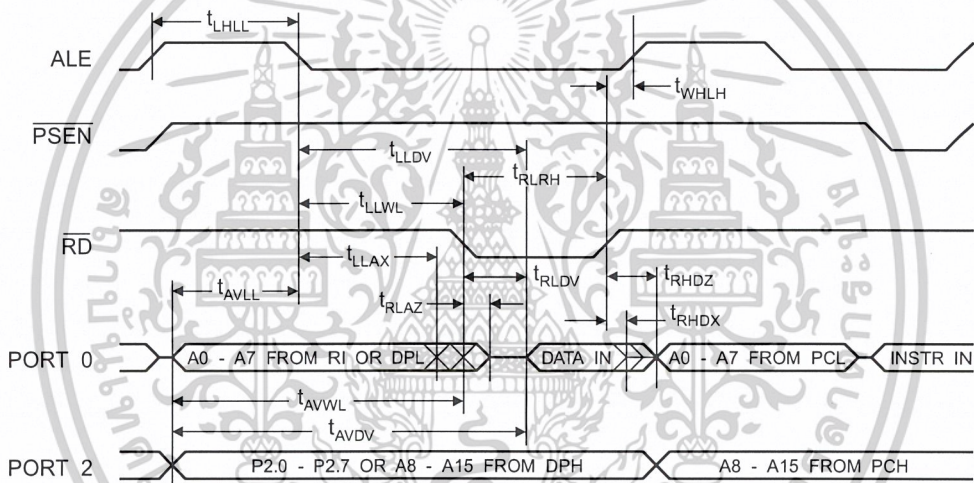
Symbol	Parameter	12 MHz Oscillator		Variable Oscillator		Units
		Min	Max	Min	Max	
$1/t_{\text{CLCL}}$	Oscillator Frequency			0	33	MHz
t_{LHLL}	ALE Pulse Width	127		$2t_{\text{CLCL}}-40$		ns
t_{AVLL}	Address Valid to ALE Low	43		$t_{\text{CLCL}}-25$		ns
t_{LLAX}	Address Hold After ALE Low	48		$t_{\text{CLCL}}-25$		ns
t_{LLIV}	ALE Low to Valid Instruction In		233		$4t_{\text{CLCL}}-65$	ns
t_{LLPL}	ALE Low to $\overline{\text{PSEN}}$ Low	43		$t_{\text{CLCL}}-25$		ns
t_{PLPH}	$\overline{\text{PSEN}}$ Pulse Width	205		$3t_{\text{CLCL}}-45$		ns
t_{PLIV}	$\overline{\text{PSEN}}$ Low to Valid Instruction In		145		$3t_{\text{CLCL}}-60$	ns
t_{PXIX}	Input Instruction Hold After $\overline{\text{PSEN}}$	0		0		ns
t_{PXIZ}	Input Instruction Float After $\overline{\text{PSEN}}$		59		$t_{\text{CLCL}}-25$	ns
t_{PXAV}	$\overline{\text{PSEN}}$ to Address Valid	75		$t_{\text{CLCL}}-8$		ns
t_{AVIV}	Address to Valid Instruction In		312		$5t_{\text{CLCL}}-80$	ns
t_{PLAZ}	$\overline{\text{PSEN}}$ Low to Address Float		10		10	ns
t_{RLRH}	$\overline{\text{RD}}$ Pulse Width	400		$6t_{\text{CLCL}}-100$		ns
t_{WLWH}	$\overline{\text{WR}}$ Pulse Width	400		$6t_{\text{CLCL}}-100$		ns
t_{RLDV}	$\overline{\text{RD}}$ Low to Valid Data In		252		$5t_{\text{CLCL}}-90$	ns
t_{RHDX}	Data Hold After $\overline{\text{RD}}$	0		0		ns
t_{RHDZ}	Data Float After $\overline{\text{RD}}$		97		$2t_{\text{CLCL}}-28$	ns
t_{LLDV}	ALE Low to Valid Data In		517		$8t_{\text{CLCL}}-150$	ns
t_{AVDV}	Address to Valid Data In		585		$9t_{\text{CLCL}}-165$	ns
t_{LLWL}	ALE Low to $\overline{\text{RD}}$ or $\overline{\text{WR}}$ Low	200	300	$3t_{\text{CLCL}}-50$	$3t_{\text{CLCL}}+50$	ns
t_{AVWL}	Address to $\overline{\text{RD}}$ or $\overline{\text{WR}}$ Low	203		$4t_{\text{CLCL}}-75$		ns
t_{QVWX}	Data Valid to $\overline{\text{WR}}$ Transition	23		$t_{\text{CLCL}}-30$		ns
t_{QVWH}	Data Valid to $\overline{\text{WR}}$ High	433		$7t_{\text{CLCL}}-130$		ns
t_{WHQX}	Data Hold After $\overline{\text{WR}}$	33		$t_{\text{CLCL}}-25$		ns
t_{RLAZ}	$\overline{\text{RD}}$ Low to Address Float		0		0	ns
t_{WHLH}	$\overline{\text{RD}}$ or $\overline{\text{WR}}$ High to ALE High	43	123	$t_{\text{CLCL}}-25$	$t_{\text{CLCL}}+25$	ns



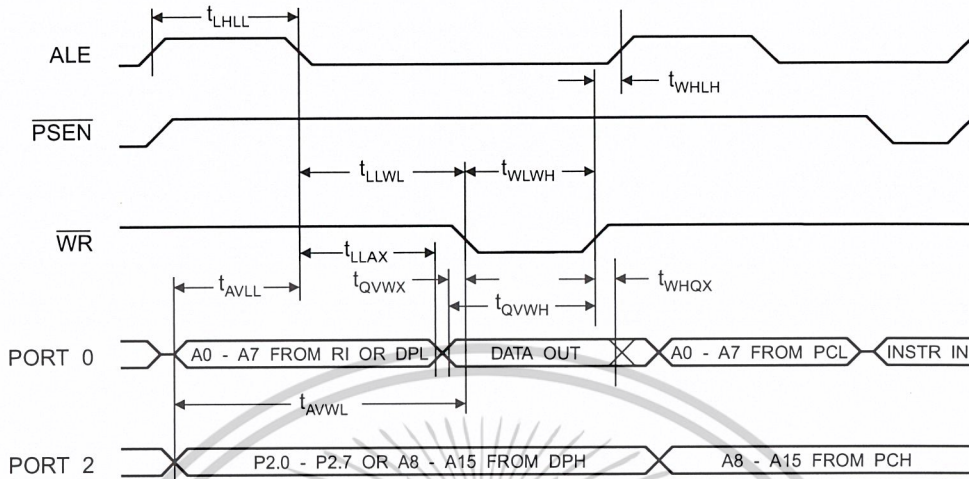
External Program Memory Read Cycle



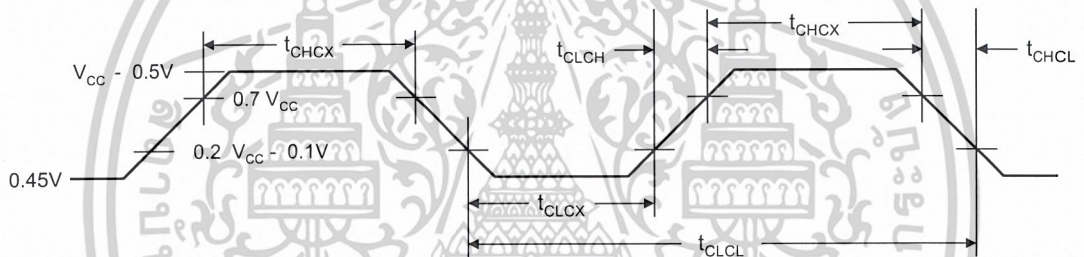
External Data Memory Read Cycle



External Data Memory Write Cycle



External Clock Drive Waveforms



External Clock Drive

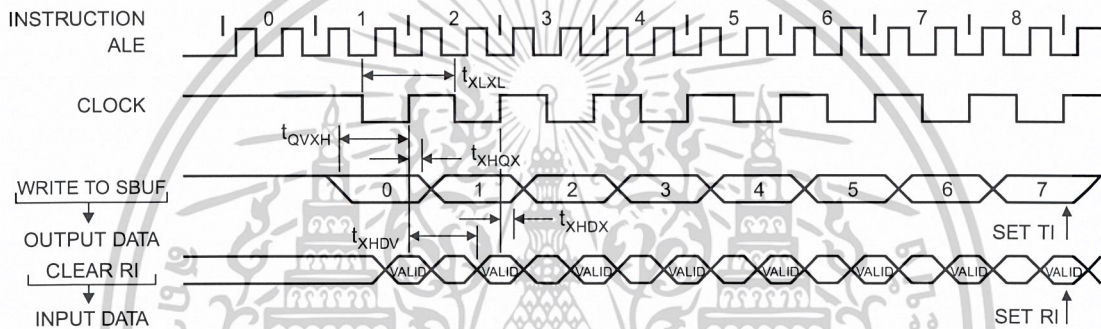
Symbol	Parameter	Min	Max	Units
$1/t_{CLCL}$	Oscillator Frequency	0	33	MHz
t_{CLCL}	Clock Period	30		ns
t_{CHCX}	High Time	12		ns
t_{CLCX}	Low Time	12		ns
t_{CLCH}	Rise Time		5	ns
t_{CHCL}	Fall Time		5	ns

Serial Port Timing: Shift Register Mode Test Conditions

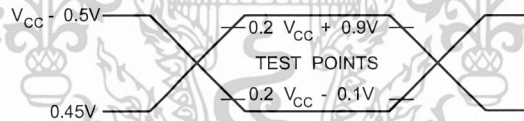
The values in this table are valid for $V_{CC} = 4.0V$ to $5.5V$ and Load Capacitance = 80 pF .

Symbol	Parameter	12 MHz Osc		Variable Oscillator		Units
		Min	Max	Min	Max	
t_{XLXL}	Serial Port Clock Cycle Time	1.0		$12t_{CLCL}$		μs
t_{QVXH}	Output Data Setup to Clock Rising Edge	700		$10t_{CLCL}-133$		ns
t_{XHQX}	Output Data Hold After Clock Rising Edge	50		$2t_{CLCL}-80$		ns
t_{XHDX}	Input Data Hold After Clock Rising Edge	0		0		ns
t_{XHDV}	Clock Rising Edge to Input Data Valid		700		$10t_{CLCL}-133$	ns

Shift Register Mode Timing Waveforms

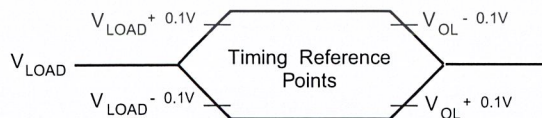


AC Testing Input/Output Waveforms⁽¹⁾



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

Float Waveforms⁽¹⁾



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

Ordering Information

Speed (MHz)	Power Supply	Ordering Code	Package	Operation Range
24	4.0V to 5.5V	AT89S52-24AC	44A	Commercial (0° C to 70° C)
		AT89S52-24JC	44J	
		AT89S52-24PC	40P6	
		AT89S52-24SC	42PS6	
		AT89S52-24AI	44A	Industrial (-40° C to 85° C)
		AT89S52-24JI	44J	
		AT89S52-24PI	40P6	
		AT89S52-24SI	42PS6	
33	4.5V to 5.5V	AT89S52-33AC	44A	Commercial (0° C to 70° C)
		AT89S52-33JC	44J	
		AT89S52-33PC	40P6	
		AT89S52-33SC	42PS6	

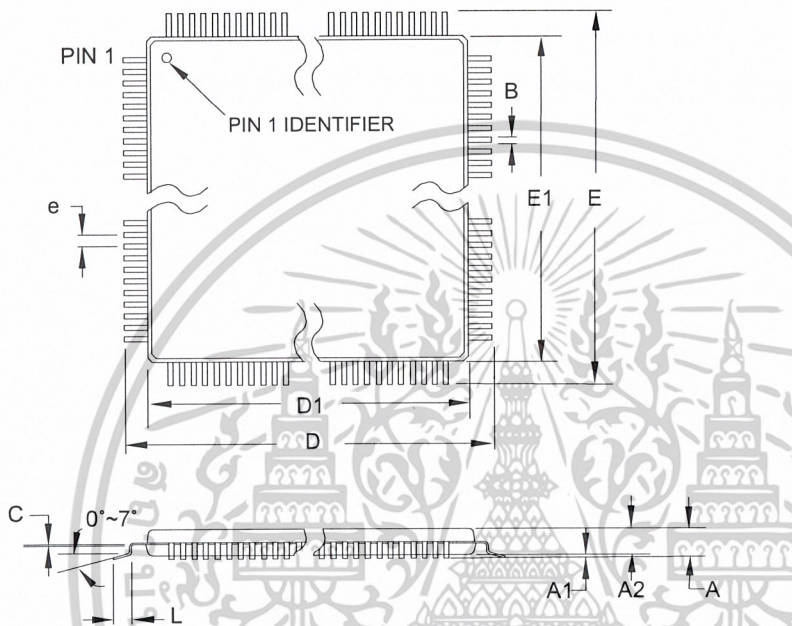


Package Type	
44A	44-lead, Thin Plastic Gull Wing Quad Flatpack (TQFP)
44J	44-lead, Plastic J-leaded Chip Carrier (PLCC)
40P6	40-pin, 0.600" Wide, Plastic Dual Inline Package (PDIP)
42PS6	42-pin, 0.600" Wide, Plastic Dual Inline Package (PDIP)



Packaging Information

44A – TQFP



COMMON DIMENSIONS
(Unit of Measure = mm)

SYMBOL	MIN	NOM	MAX	NOTE
A	–	–	1.20	
A1	0.05	–	0.15	
A2	0.95	1.00	1.05	
D	11.75	12.00	12.25	
D1	9.90	10.00	10.10	Note 2
E	11.75	12.00	12.25	
E1	9.90	10.00	10.10	Note 2
B	0.30	–	0.45	
C	0.09	–	0.20	
L	0.45	–	0.75	
e	0.80 TYP			

- Notes:
1. This package conforms to JEDEC reference MS-026, Variation ACB.
 2. Dimensions D1 and E1 do not include mold protrusion. Allowable protrusion is 0.25 mm per side. Dimensions D1 and E1 are maximum plastic body size dimensions including mold mismatch.
 3. Lead coplanarity is 0.10 mm maximum.

10/5/2001



2325 Orchard Parkway
San Jose, CA 95131

TITLE

**44A, 44-lead, 10 x 10 mm Body Size, 1.0 mm Body Thickness,
0.8 mm Lead Pitch, Thin Profile Plastic Quad Flat Package (TQFP)**

DRAWING NO.

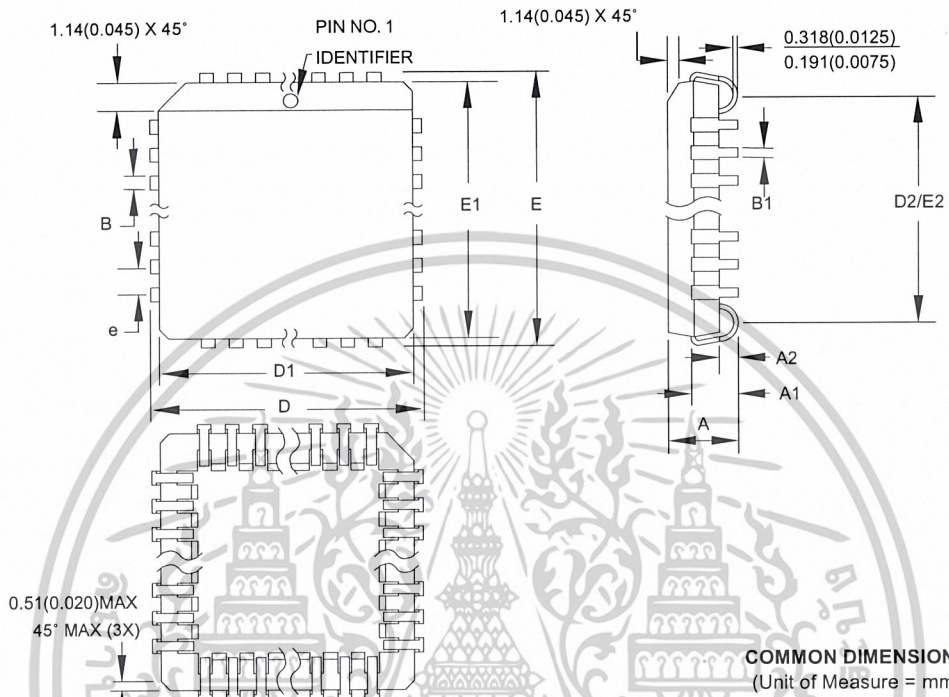
44A

REV.

B

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

44J – PLCC



- Notes:
1. This package conforms to JEDEC reference MS-018, Variation AC.
 2. Dimensions D1 and E1 do not include mold protrusion. Allowable protrusion is .010" (0.254 mm) per side. Dimension D1 and E1 include mold mismatch and are measured at the extreme material condition at the upper or lower parting line.
 3. Lead coplanarity is 0.004" (0.102 mm) maximum.

10/04/01



2325 Orchard Parkway
San Jose, CA 95131

TITLE

44J, 44-lead, Plastic J-leaded Chip Carrier (PLCC)

DRAWING NO.

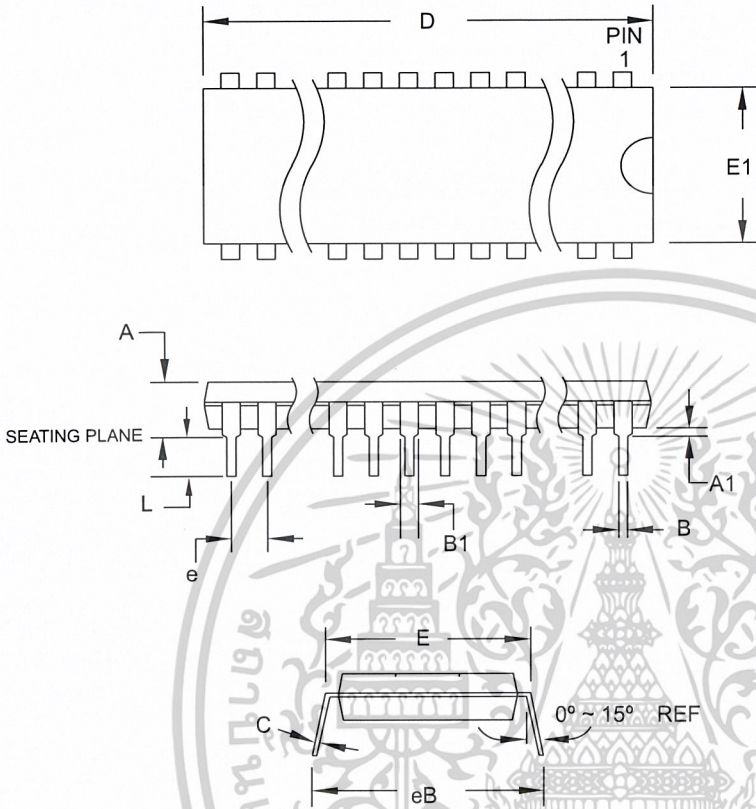
44J

REV.

B



40P6 – PDIP



COMMON DIMENSIONS
(Unit of Measure = mm)

SYMBOL	MIN	NOM	MAX	NOTE
A	–	–	4.826	
A1	0.381	–	–	
D	52.070	–	52.578	Note 2
E	15.240	–	15.875	
E1	13.462	–	13.970	Note 2
B	0.356	–	0.559	
B1	1.041	–	1.651	
L	3.048	–	3.556	
C	0.203	–	0.381	
eB	15.494	–	17.526	
e	2.540 TYP			

- Notes:
1. This package conforms to JEDEC reference MS-011, Variation AC.
 2. Dimensions D and E1 do not include mold Flash or Protrusion. Mold Flash or Protrusion shall not exceed 0.25 mm (0.010").

09/28/01

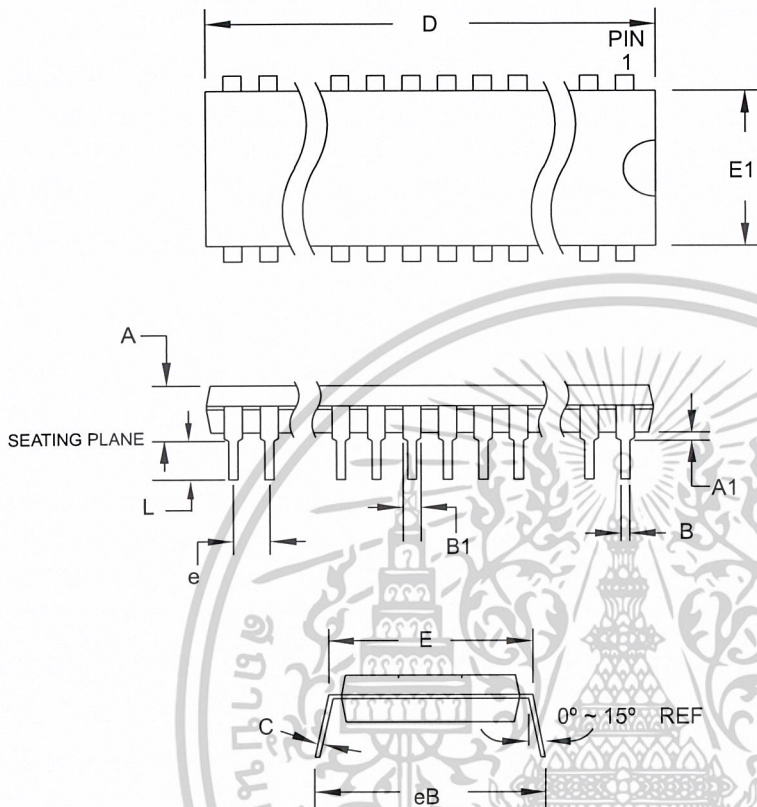
2325 Orchard Parkway
San Jose, CA 95131

TITLE
40P6, 40-lead (0.600"/15.24 mm Wide) Plastic Dual In-line Package (PDIP)

DRAWING NO.
40P6

REV.
B

42PS6 – PDIP




COMMON DIMENSIONS
(Unit of Measure = mm)

SYMBOL	MIN	NOM	MAX	NOTE
A	–	–	4.83	
A1	0.51	–	–	
D	36.70	–	36.96	Note 2
E	15.24	–	15.88	
E1	13.46	–	13.97	Note 2
B	0.38	–	0.56	
B1	0.76	–	1.27	
L	3.05	–	3.43	
C	0.20	–	0.30	
eB	–	–	18.55	
e	1.78 TYP			

- Notes:
1. This package conforms to JEDEC reference MS-011, Variation AC.
 2. Dimensions D and E1 do not include mold Flash or Protrusion. Mold Flash or Protrusion shall not exceed 0.25 mm (0.010").

11/6/03

 2325 Orchard Parkway San Jose, CA 95131	TITLE 42PS6, 42-lead (0.600"/15.24 mm Wide) Plastic Dual Inline Package (PDIP)	DRAWING NO.	REV.
		42PS6	A



Atmel Corporation

2325 Orchard Parkway
San Jose, CA 95131, USA
Tel: 1(408) 441-0311
Fax: 1(408) 487-2600

Regional Headquarters

Europe

Atmel Sarl
Route des Arsenalux 41
Case Postale 80
CH-1705 Fribourg
Switzerland
Tel: (41) 26-426-5555
Fax: (41) 26-426-5500

Asia

Room 1219
Chinachem Golden Plaza
77 Mody Road Tsimshatsui
East Kowloon
Hong Kong
Tel: (852) 2721-9778
Fax: (852) 2722-1369

Japan

9F, Tonetsu Shinkawa Bldg.
1-24-8 Shinkawa
Chuo-ku, Tokyo 104-0033
Japan
Tel: (81) 3-3523-3551
Fax: (81) 3-3523-7581

Atmel Operations

Memory

2325 Orchard Parkway
San Jose, CA 95131, USA
Tel: 1(408) 441-0311
Fax: 1(408) 436-4314

Microcontrollers

2325 Orchard Parkway
San Jose, CA 95131, USA
Tel: 1(408) 441-0311
Fax: 1(408) 436-4314

La Chantrerie
BP 70602
44306 Nantes Cedex 3, France
Tel: (33) 2-40-18-18-18
Fax: (33) 2-40-18-19-60

ASIC/ASSP/Smart Cards

Zone Industrielle
13106 Rousset Cedex, France
Tel: (33) 4-42-53-60-00
Fax: (33) 4-42-53-60-01

1150 East Cheyenne Mtn. Blvd.
Colorado Springs, CO 80906, USA
Tel: 1(719) 576-3300
Fax: 1(719) 540-1759

Scottish Enterprise Technology Park
Maxwell Building
East Kilbride G75 0QR, Scotland
Tel: (44) 1355-803-000
Fax: (44) 1355-242-743

RF/Automotive

Theresienstrasse 2
Postfach 3535
74025 Heilbronn, Germany
Tel: (49) 71-31-67-0
Fax: (49) 71-31-67-2340

1150 East Cheyenne Mtn. Blvd.
Colorado Springs, CO 80906, USA
Tel: 1(719) 576-3300
Fax: 1(719) 540-1759

Biometrics/Imaging/Hi-Rel MPU/ High Speed Converters/RF Datacom

Avenue de Rochepleine
BP 123
38521 Saint-Egreve Cedex, France
Tel: (33) 4-76-58-30-00
Fax: (33) 4-76-58-34-80

Literature Requests

www.atmel.com/literature

Disclaimer: Atmel Corporation makes no warranty for the use of its products, other than those expressly contained in the Company's standard warranty which is detailed in Atmel's Terms and Conditions located on the Company's web site. The Company assumes no responsibility for any errors which may appear in this document, reserves the right to change devices or specifications detailed herein at any time without notice, and does not make any commitment to update the information contained herein. No licenses to patents or other intellectual property of Atmel are granted by the Company in connection with the sale of Atmel products, expressly or by implication. Atmel's products are not authorized for use as critical components in life support devices or systems.

© Atmel Corporation 2003. All rights reserved. Atmel® and combinations thereof are the registered trademarks of Atmel Corporation or its subsidiaries. MCS® is a registered trademark of Intel Corporation. Other terms and product names may be the trademarks of others.

Printed on recycled paper.

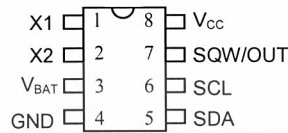
1919B-MICRO-11/03

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

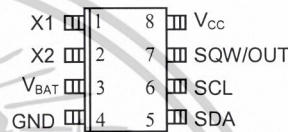
FEATURES

- 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

PIN ASSIGNMENT



DS1307 8-Pin DIP (300 mil)



DS1307Z 8-Pin SOIC (150 mil)

ORDERING INFORMATION

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

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

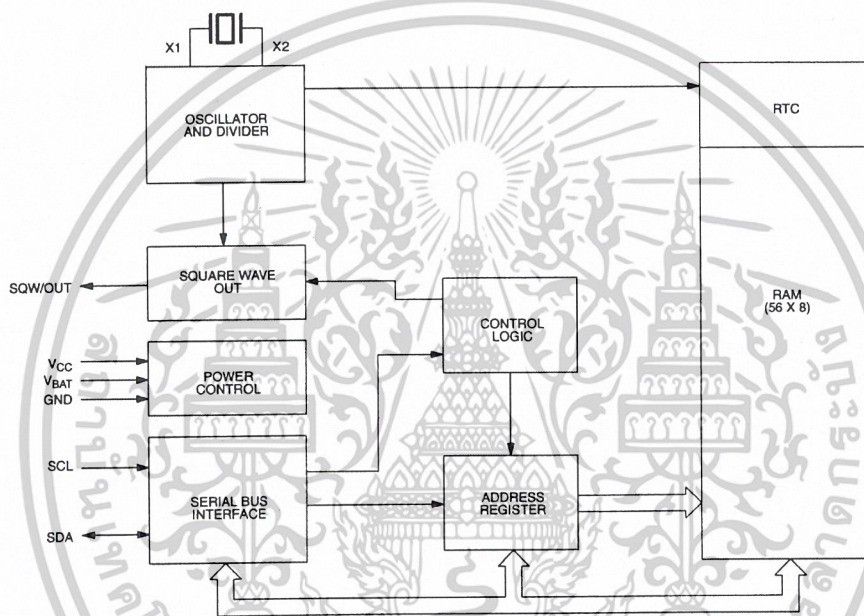
DESCRIPTION

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.

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 $1.25 \times V_{BAT}$ 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 $1.25 \times V_{BAT}$. 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 $1.25 \times V_{BAT}$, 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 $1.25 \times V_{BAT}$ 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.

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

00H	SECONDS
	MINUTES
	HOURS
	DAY
	DATE
	MONTH
	YEAR
07H	CONTROL
08H	RAM
3FH	56 x 8

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.

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.

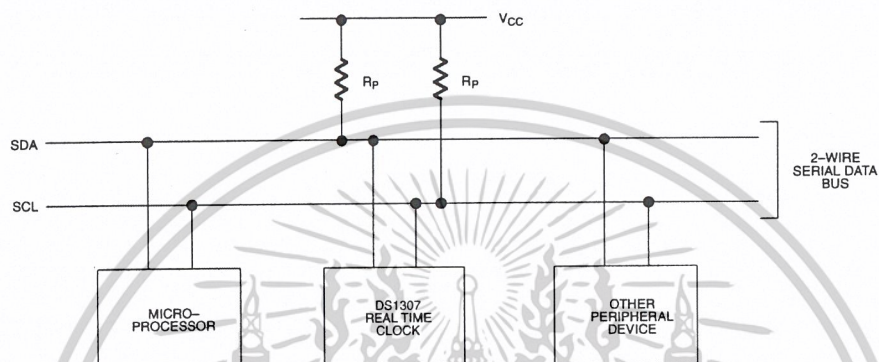
SQUAREWAVE OUTPUT FREQUENCY Table 1

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

2-WIRE SERIAL DATA BUS

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

TYPICAL 2-WIRE BUS CONFIGURATION Figure 4



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

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

Accordingly, the following bus conditions have been defined:

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

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

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

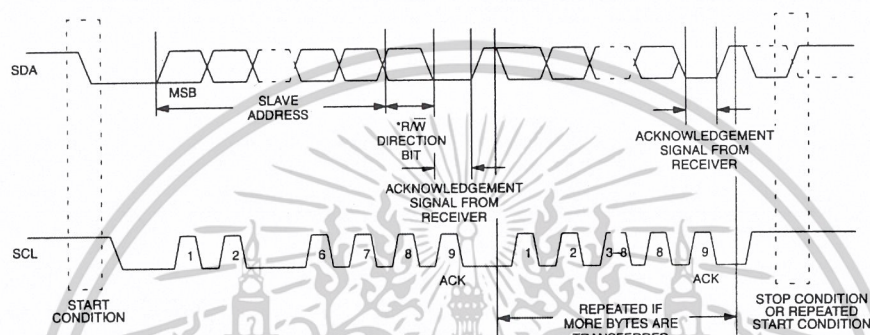
Data valid: The state of the data line represents valid data when, after a START condition, the data line is stable for the duration of the HIGH period of the clock signal. The data on the line must be changed during the LOW period of the clock signal. There is one clock pulse per bit of data.

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

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

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

DATA TRANSFER ON 2-WIRE SERIAL BUS Figure 5



Depending upon the state of the R/W bit, two types of data transfer are possible:

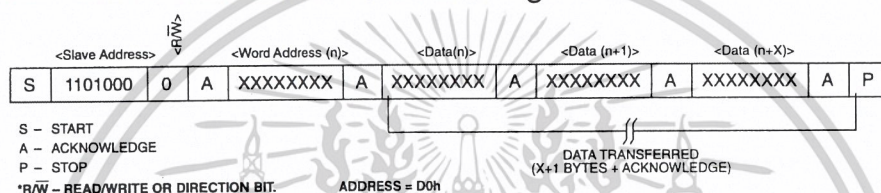
1. **Data transfer from a master transmitter to a slave receiver.** The first byte transmitted by the master is the slave address. Next follows a number of data bytes. The slave returns an acknowledge bit after each received byte. Data is transferred with the most significant bit (MSB) first.
2. **Data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

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

The DS1307 may operate in the following two modes:

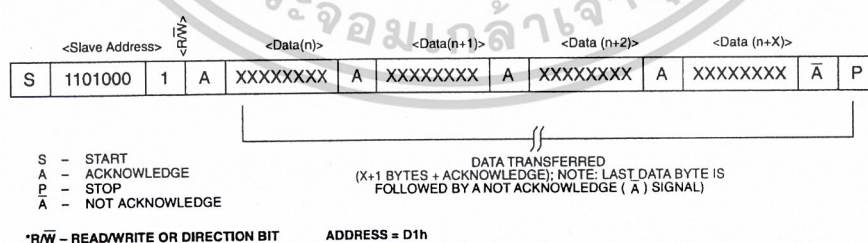
1. **Slave receiver mode (DS1307 write mode):** Serial data and clock are received through SDA and SCL. After each byte is received an acknowledge bit is transmitted. START and STOP conditions are recognized as the beginning and end of a serial transfer. Address recognition is performed by hardware after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

DATA WRITE - SLAVE RECEIVER MODE Figure 6



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

DATA READ - SLAVE TRANSMITTER MODE Figure 7



ABSOLUTE MAXIMUM RATINGS*

Voltage on Any Pin Relative to Ground	-0.5V to +7.0V
Operating Temperature	0°C to 70°C (-40°C to 85°C for industrial)
Storage Temperature	-55°C to +125°C
Soldering Temperature	260°C for 10 seconds DIP See JPC/JEDEC Standard J-STD-020A for Surface Mount Devices

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

RECOMMENDED DC OPERATING CONDITIONS

(0°C to 70°C or -40°C to +85°C)

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

DC ELECTRICAL CHARACTERISTICS(0°C to 70°C or -40°C to +85°C; V_{CC} = 4.5V to 5.5V)

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

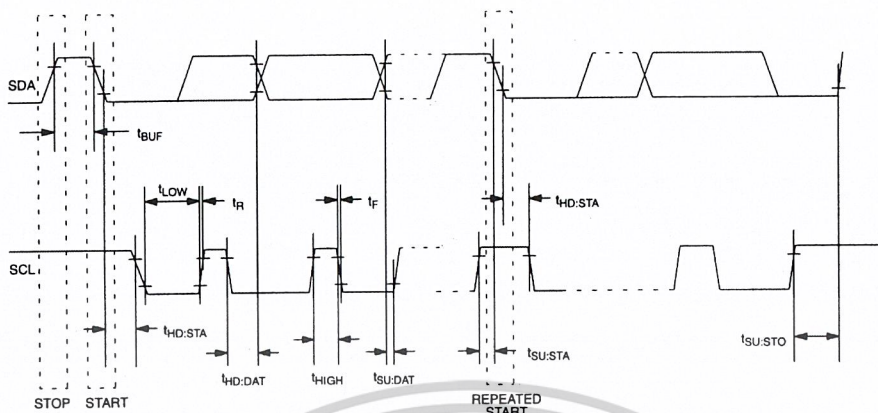
AC ELECTRICAL CHARACTERISTICS(0°C to 70°C or -40°C to +85°C; $V_{CC}=4.5V$ to 5.5V)

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

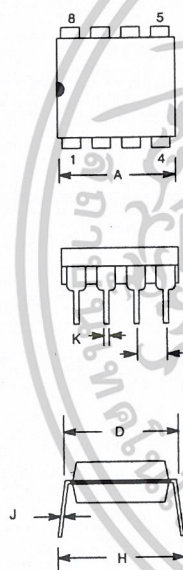
NOTES:

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

TIMING DIAGRAM Figure 8



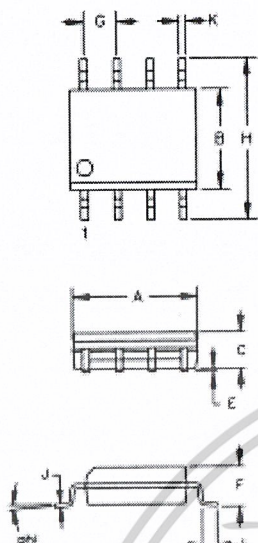
**DS1307 64 X 8 SERIAL REAL TIME CLOCK
8-PIN DIP MECHANICAL DIMENSIONS**



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

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

DS1307Z 64 X 8 SERIAL REAL TIME CLOCK 8-PIN SOIC (150-MIL) MECHANICAL DIMENSIONS



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

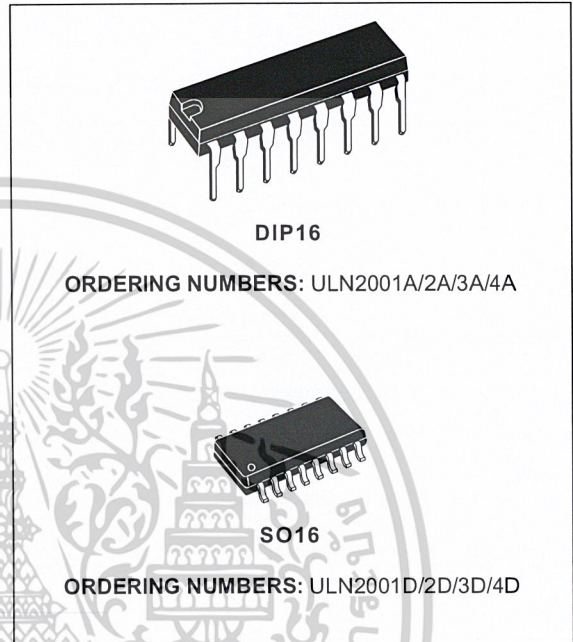
56-G2008-001



ULN2001A-ULN2002A ULN2003A-ULN2004A

SEVEN DARLINGTON ARRAYS

- SEVEN DARLINGTONS PER PACKAGE
- OUTPUT CURRENT 500mA PER DRIVER (600mA PEAK)
- OUTPUT VOLTAGE 50V
- INTEGRATED SUPPRESSION DIODES FOR INDUCTIVE LOADS
- OUTPUTS CAN BE PARALLELED FOR HIGHER CURRENT
- TTL/CMOS/PMOS/DTL COMPATIBLE INPUTS
- INPUTS PINNED OPPOSITE OUTPUTS TO SIMPLIFY LAYOUT



DESCRIPTION

The ULN2001A, ULN2002A, ULN2003 and ULN2004A are high voltage, high current darlington arrays each containing seven open collector darlington pairs with common emitters. Each channel rated at 500mA and can withstand peak currents of 600mA. Suppression diodes are included for inductive load driving and the inputs are pinned opposite the outputs to simplify board layout.

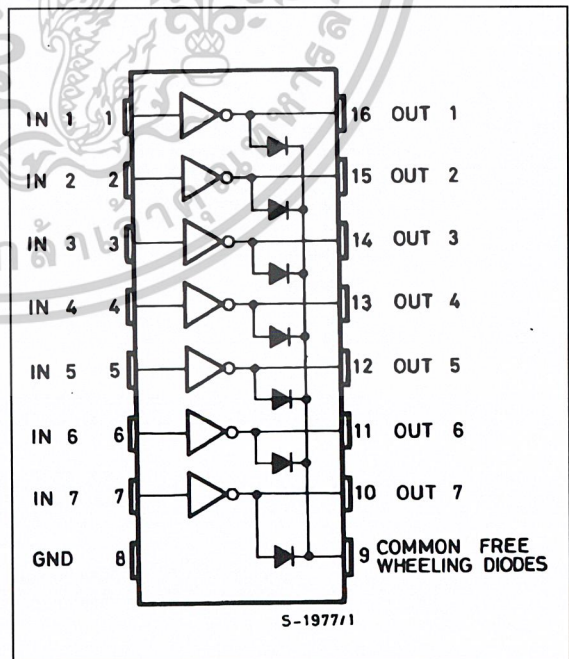
The four versions interface to all common logic families :

ULN2001A	General Purpose, DTL, TTL, PMOS, CMOS
ULN2002A	14-25V PMOS
ULN2003A	5V TTL, CMOS
ULN2004A	6-15V CMOS, PMOS

These versatile devices are useful for driving a wide range of loads including solenoids, relays DC motors, LED displays filament lamps, thermal print-heads and high power buffers.

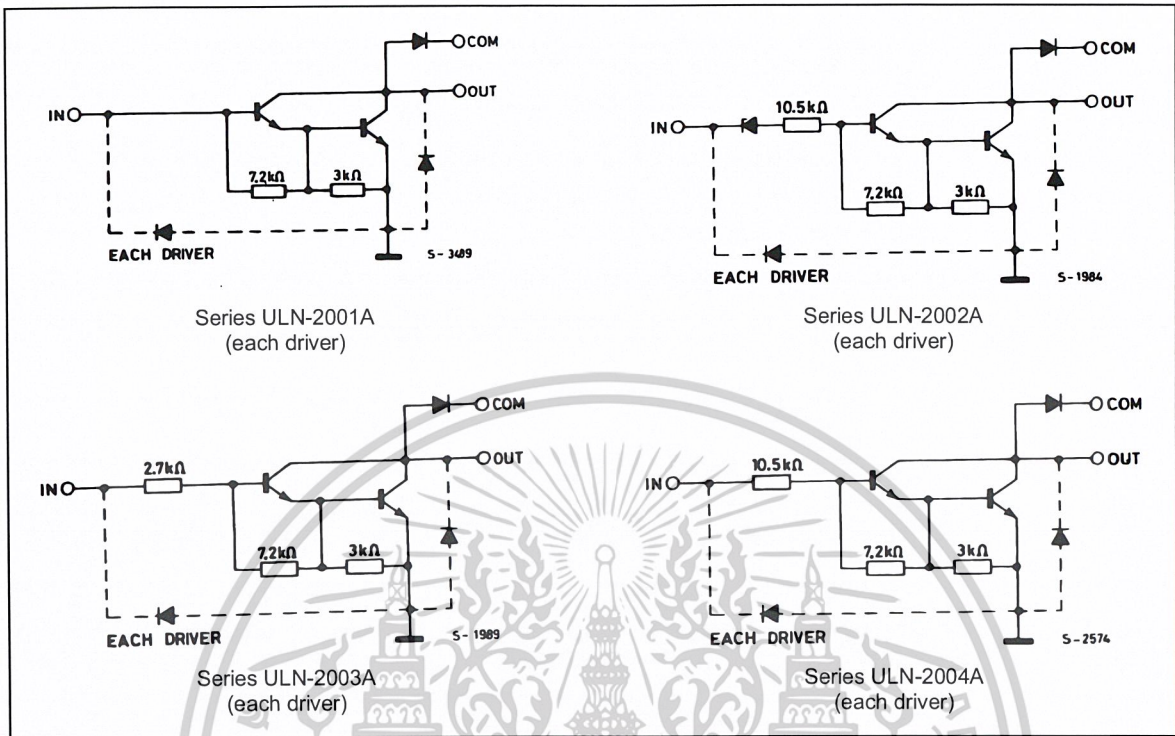
The ULN2001A/2002A/2003A and 2004A are supplied in 16 pin plastic DIP packages with a copper leadframe to reduce thermal resistance. They are available also in small outline package (SO-16) as ULN2001D/2002D/2003D/2004D.

PIN CONNECTION



ULN2001A - ULN2002A - ULN2003A - ULN2004A

SCHEMATIC DIAGRAM



ABSOLUTE MAXIMUM RATINGS

Symbol	Parameter	Value	Unit
V_o	Output Voltage	50	V
V_{in}	Input Voltage (for ULN2002A/D - 2003A/D - 2004A/D)	30	V
I_c	Continuous Collector Current	500	mA
I_b	Continuous Base Current	25	mA
T_{amb}	Operating Ambient Temperature Range	- 20 to 85	°C
T_{stg}	Storage Temperature Range	- 55 to 150	°C
T_j	Junction Temperature	150	°C

THERMAL DATA

Symbol	Parameter	DIP16	SO16	Unit
$R_{th\ j-amb}$	Thermal Resistance Junction-ambient	Max. 70	120	°C/W

ELECTRICAL CHARACTERISTICS ($T_{amb} = 25^{\circ}\text{C}$ unless otherwise specified)

Symbol	Parameter	Test Conditions	Min.	Typ.	Max.	Unit	Fig.
I_{CEX}	Output Leakage Current	$V_{CE} = 50\text{V}$ $T_{amb} = 70^{\circ}\text{C}$, $V_{CE} = 50\text{V}$			50 100	μA μA	1a 1a
		$T_{amb} = 70^{\circ}\text{C}$ for ULN2002A $V_{CE} = 50\text{V}$, $V_i = 6\text{V}$			500	μA	1b
		for ULN2004A $V_{CE} = 50\text{V}$, $V_i = 1\text{V}$			500	μA	1b
$V_{CE(sat)}$	Collector-emitter Saturation Voltage	$I_C = 100\text{mA}$, $I_B = 250\mu\text{A}$		0.9	1.1	V	2
		$I_C = 200\text{mA}$, $I_B = 350\mu\text{A}$		1.1	1.3	V	2
		$I_C = 350\text{mA}$, $I_B = 500\mu\text{A}$		1.3	1.6	V	2
$I_{i(on)}$	Input Current	for ULN2002A, $V_i = 17\text{V}$		0.82	1.25	mA	3
		for ULN2003A, $V_i = 3.85\text{V}$		0.93	1.35	mA	3
		for ULN2004A, $V_i = 5\text{V}$		0.35	0.5	mA	3
		$V_i = 12\text{V}$		1	1.45	mA	3
$I_{i(off)}$	Input Current	$T_{amb} = 70^{\circ}\text{C}$, $I_C = 500\mu\text{A}$	50	65		μA	4
$V_{i(on)}$	Input Voltage	$V_{CE} = 2\text{V}$ for ULN2002A $I_C = 300\text{mA}$			13	V	5
		for ULN2003A $I_C = 200\text{mA}$			2.4		
		$I_C = 250\text{mA}$			2.7		
		$I_C = 300\text{mA}$			3		
		for ULN2004A $I_C = 125\text{mA}$			5		
		$I_C = 200\text{mA}$			6		
		$I_C = 275\text{mA}$			7		
		$I_C = 350\text{mA}$			8		
h_{FE}	DC Forward Current Gain	for ULN2001A $V_{CE} = 2\text{V}$, $I_C = 350\text{mA}$	1000				2
C_i	Input Capacitance			15	25	pF	
t_{PLH}	Turn-on Delay Time	$0.5 V_i$ to $0.5 V_o$		0.25	1	μs	
t_{PHL}	Turn-off Delay Time	$0.5 V_i$ to $0.5 V_o$		0.25	1	μs	
I_R	Clamp Diode Leakage Current	$V_R = 50\text{V}$ $T_{amb} = 70^{\circ}\text{C}$, $V_R = 50\text{V}$			50 100	μA μA	6 6
V_F	Clamp Diode Forward Voltage	$I_F = 350\text{mA}$		1.7	2	V	7

TEST CIRCUITS

Figure 1a.

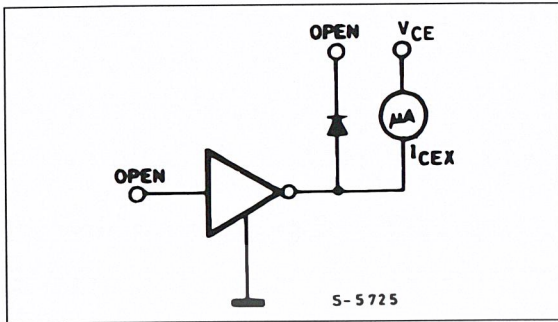


Figure 1b.

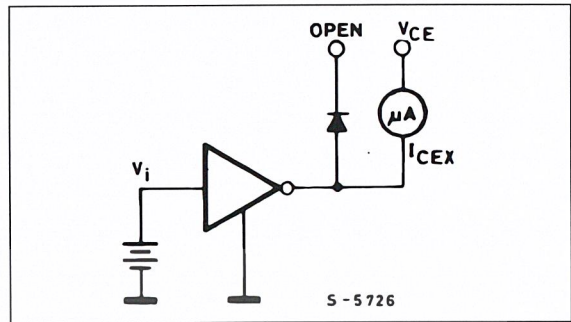


Figure 2.

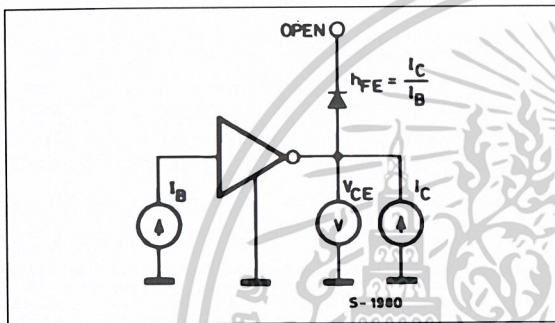


Figure 3.

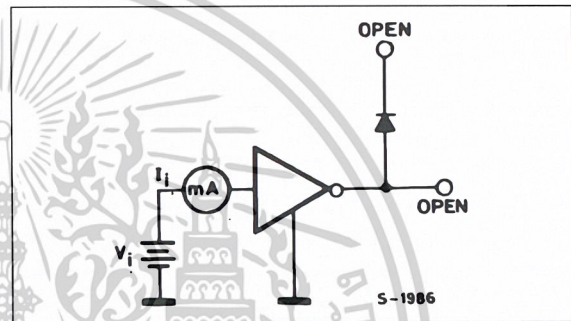


Figure 4.

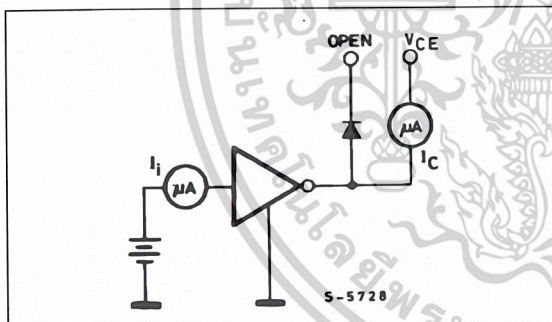


Figure 5.

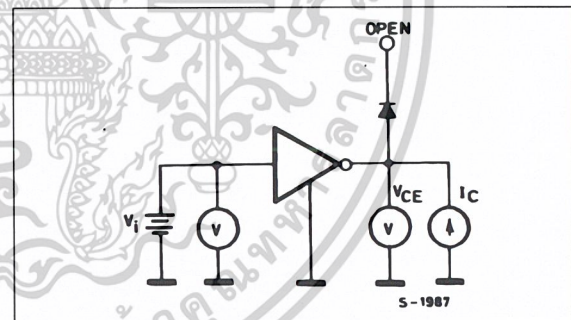


Figure 6.

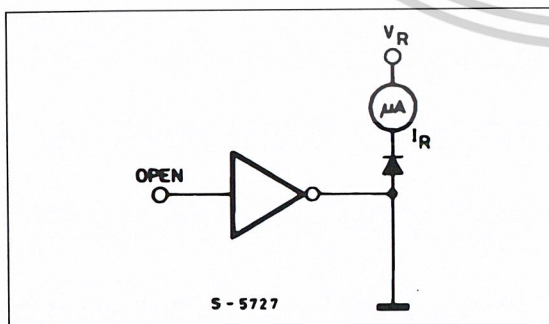


Figure 7.

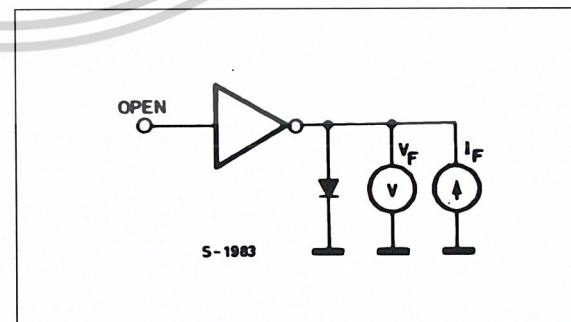


Figure 8: Collector Current versus Input Current

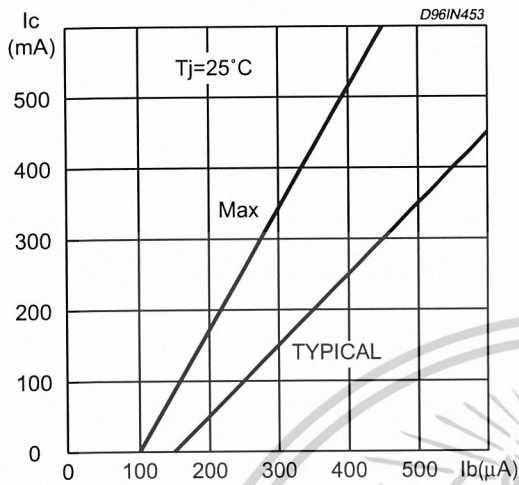


Figure 9: Collector Current versus Saturation Voltage

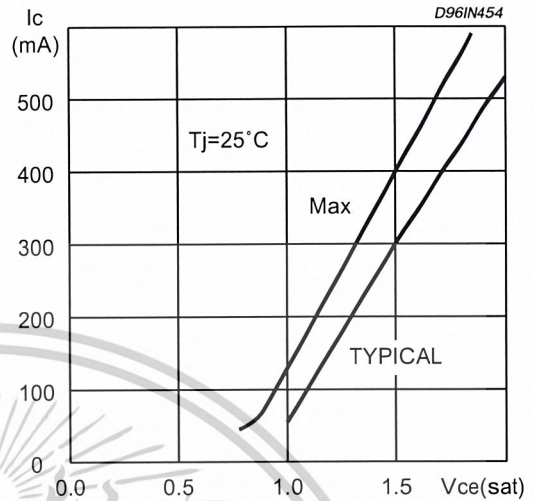


Figure 10: Peak Collector Current versus Duty Cycle

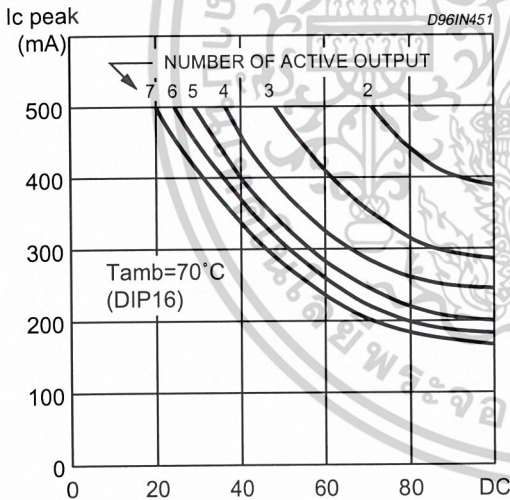
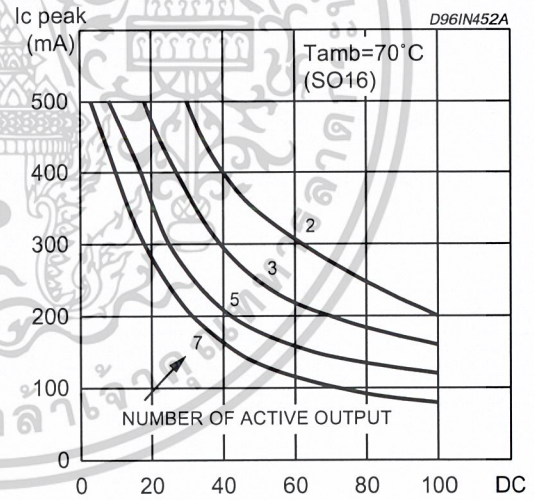


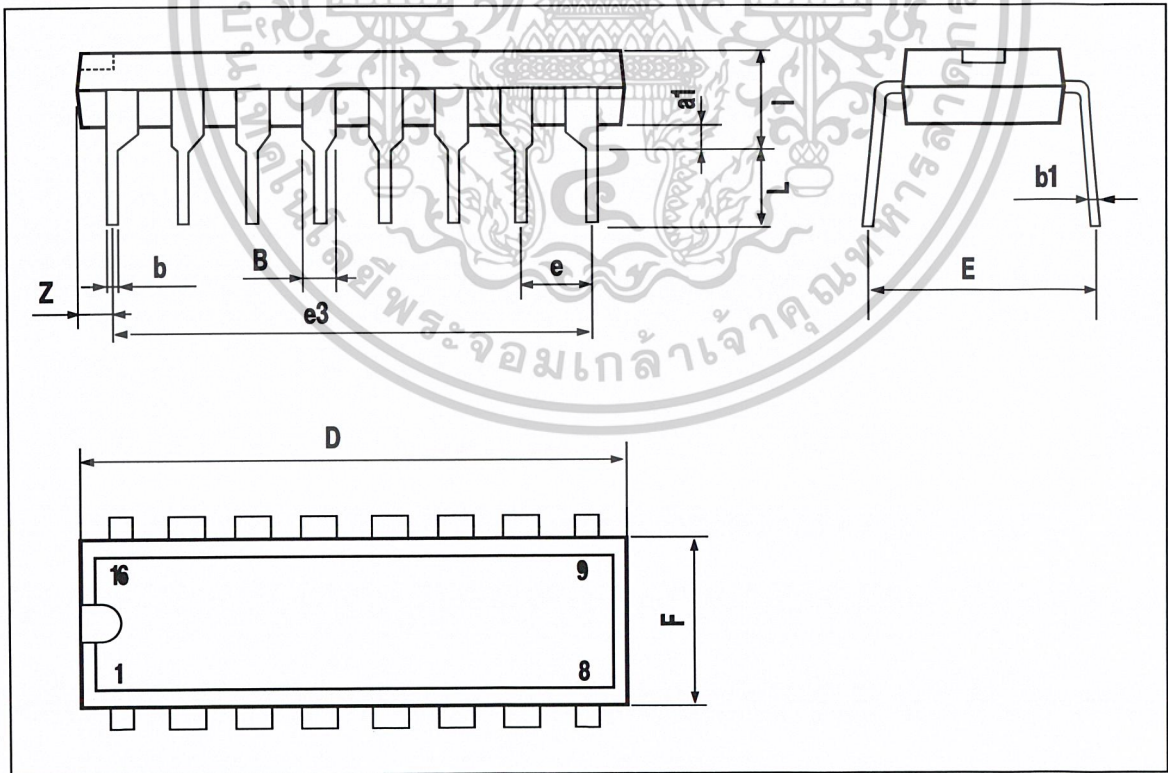
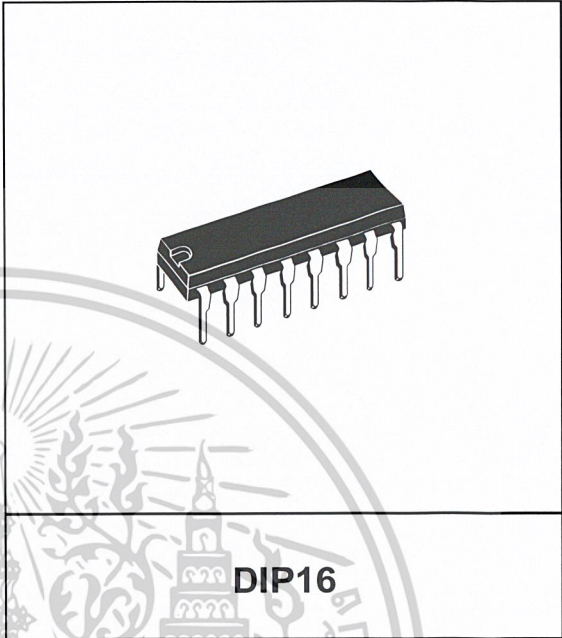
Figure 11: Peak Collector Current versus Duty Cycle



ULN2001A - ULN2002A - ULN2003A - ULN2004A

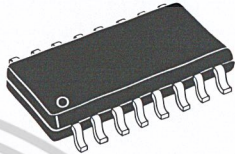
DIM.	mm			inch		
	MIN.	TYP.	MAX.	MIN.	TYP.	MAX.
a1	0.51			0.020		
B	0.77		1.65	0.030		0.065
b		0.5			0.020	
b1		0.25			0.010	
D			20			0.787
E		8.5			0.335	
e		2.54			0.100	
e3		17.78			0.700	
F			7.1			0.280
I			5.1			0.201
L		3.3			0.130	
Z			1.27			0.050

OUTLINE AND MECHANICAL DATA



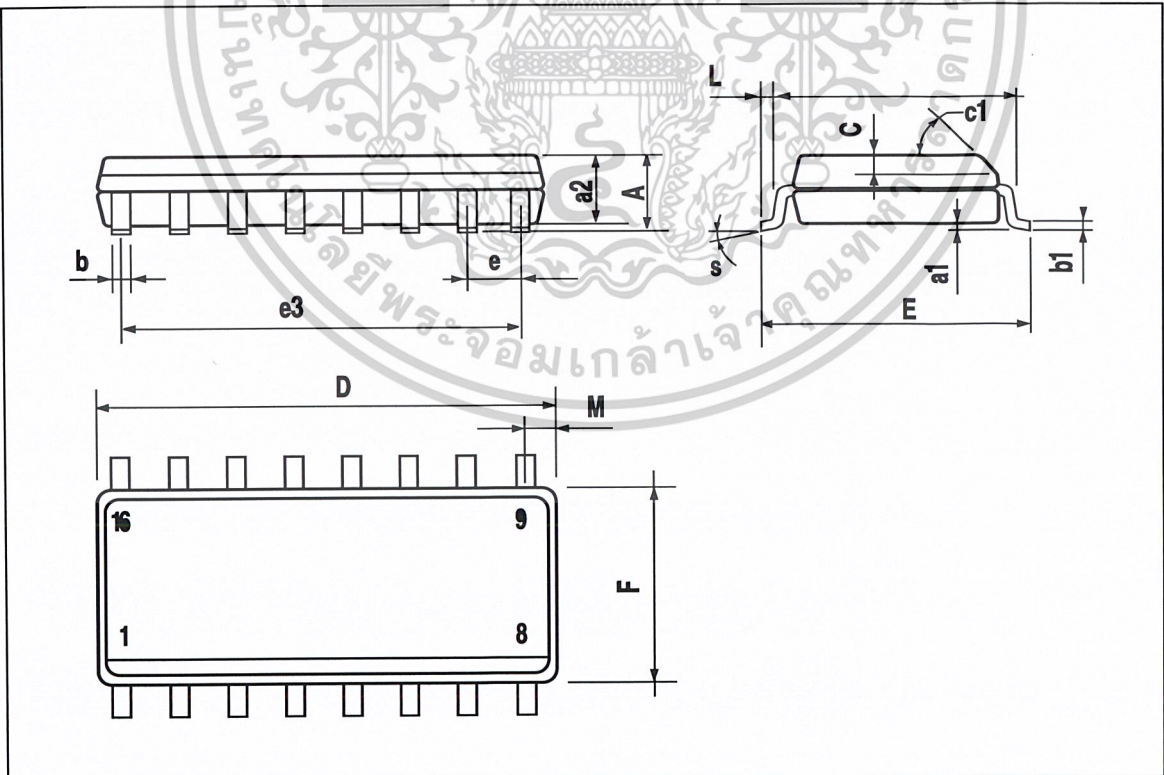
DIM.	mm			inch		
	MIN.	TYP.	MAX.	MIN.	TYP.	MAX.
A			1.75			0.069
a1	0.1		0.25	0.004		0.009
a2			1.6			0.063
b	0.35		0.46	0.014		0.018
b1	0.19		0.25	0.007		0.010
C		0.5			0.020	
c1	45° (typ.)					
D (1)	9.8		10	0.386		0.394
E	5.8		6.2	0.228		0.244
e		1.27			0.050	
e3		8.89			0.350	
F (1)	3.8		4	0.150		0.157
G	4.6		5.3	0.181		0.209
L	0.4		1.27	0.016		0.050
M			0.62			0.024
S	8°(max.)					

OUTLINE AND MECHANICAL DATA



SO16 Narrow

(1) D and F do not include mold flash or protrusions. Mold flash or protrusions shall not exceed 0.15mm (.006inch).



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

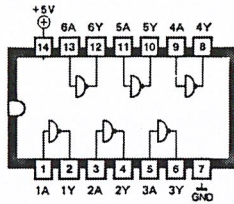


Information furnished is believed to be accurate and reliable. However, STMicroelectronics assumes no responsibility for the consequences of use of such information nor for any infringement of patents or other rights of third parties which may result from its use. No license is granted by implication or otherwise under any patent or patent rights of STMicroelectronics. Specification mentioned in this publication are subject to change without notice. This publication supersedes and replaces all information previously supplied. STMicroelectronics products are not authorized for use as critical components in life support devices or systems without express written approval of STMicroelectronics.

The ST logo is a registered trademark of STMicroelectronics
© 2002 STMicroelectronics – Printed in Italy – All Rights Reserved
STMicroelectronics GROUP OF COMPANIES

Australia - Brazil - Canada - China - Finland - France - Germany - Hong Kong - India - Israel - Italy - Japan - Malaysia - Malta - Morocco - Singapore - Spain - Sweden - Switzerland - United Kingdom - United States.

<http://www.st.com>



Description:
This package contains six inverters.

Mode of operation:
All six inverters can be used independently of one another.
For each inverter, a low input gives a high output and vice versa.
The 7414 contains six inverters with Schmitt-trigger inputs and has the same pinout.

Application:
Logical inversion, pulse shaping, oscillators

Data:									
Propagation delay	ns	10	6	2.75	3.5	6	33	9.5	3
Supply current	mA	12	2	8	6.9	26	1.2	3	23
Families:		Std	ALS	AS	F	H	L	LS	S
		●	●	●	●	●	●	●	●

Hex INVERTER

7404

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้