

เครื่องอบอัตโนมัติ

AUTOMATICS OVEN



เลขหมู่.....
เลขทะเบียน.....50387
วัน,เดือน,ปี.....13 พ.ศ. 2547

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

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาอิเล็กทรอนิกส์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานปีการศึกษา 2545 ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เครื่องอบอัตโนมัติ
AUTOMATICS OVEN



ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาอิเล็กทรอนิกส์

คณะวิศวกรรมศาสตร์


สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานปีการศึกษา 2545 ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญาโทปีการศึกษา 2545
ภาควิชา อิเล็กทรอนิกส์
คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
เรื่อง เครื่องอบอัตโนมัติ
Automatics Oven

ผู้จัดทำ

1. นายกิตติ โสทธิบำเพ็ญ เลขประจำตัว 42010019
2. นายคมสิทธิ์ คุณวัฒน์ เลขประจำตัว 42010042
3. นางสาวจรรุรินทร์ สิทธิโชคคณานนท์ เลขประจำตัว 42010055


.....อาจารย์ที่ปรึกษา
(อาจารย์พลศาสตร์ เลิศประเสริฐ)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เครื่องอบอัตโนมัติ

นายกิตติ โสทธิบำเพ็ญ
นายคมสิทธิ์ คุณวัฒน์
นางสาวจารุรินทร์ สิทธิโชคคณานนท์
อาจารย์พลศาสตร์ เลิศประเสริฐ (อาจารย์ที่ปรึกษา)
ภาคการศึกษาที่ 2 ปีการศึกษา 2545

บทคัดย่อ

รายงานฉบับนี้เป็นส่วนหนึ่งของวิชา 01044509 Project II โดยเป็นการศึกษาการออกแบบวงจรของเครื่องอบอัตโนมัติ ซึ่งประกอบด้วยส่วนที่เป็นเซนเซอร์วัดอุณหภูมิ โดยใช้ไอซีสำเร็จรูปเบอร์ DS1620 ส่วนที่เป็นวงจรสำหรับขับเครื่องกำเนิดความร้อน โดยสามารถควบคุมกำลังงานไฟฟ้าด้วยการสร้างสัญญาณทริกให้แก่ไทรแอก และส่วนประมวลผลโดยไมโครโปรเซสเซอร์ และแสดงผลโดยจะแสดงผลชนิดLCD รวมทั้ง การเขียน โปรแกรมภาษาแอสเซมบลี ควบคุมการทำงานของไมโครคอนโทรลเลอร์ด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Automatics Oven

Mr.Kitti Sothibumphen

Mr. Khomsit Kunawat

Miss. Jarurin Sittichokkananon

Mr. Polsart Lertprasert (Adviser)

2nd Semester, Educational Year 2002

Abstract

This project is a part of subject 01044509 Project II. Presents a circuit designed of automatics oven. The system includes of temperature sensor ,package IC number DS1620 , the drive circuit for heater that can control power by generate triacker's signal to trig triac and we process the data by microprocessor and display part we use module LCD . Another knowledge of this project is how to write assembly language of microcontroller.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

โครงการชิ้นนี้สำเร็จลงได้ด้วยดี เนื่องจากความช่วยเหลือ และคำแนะนำของอาจารย์ พลศาสตร์ เลิศประเสริฐ อาจารย์ที่ปรึกษา รวมทั้งอาจารย์ภาควิชาต่างๆ ท่าน รุ่นพี่และเพื่อนๆ ที่คอย ให้ความช่วยเหลือและให้คำแนะนำต่างๆ ที่เป็นประโยชน์ในการทำโครงการชิ้นนี้ ทางผู้จัดทำจึงขอขอบคุณทุกท่านมา ณ ที่นี้ด้วย



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

	หน้า
บทคัดย่อ	I
Abstract	II
กิตติกรรมประกาศ	III
สารบัญ	IV
บทที่ 1 บทนำ	1
บทที่ 2 ทฤษฎี และหลักการ	2
2.1 การวัดอุณหภูมิ	2
2.1.1 ลักษณะสำคัญ	2
2.1.2 การอ่านอุณหภูมิ	3
2.1.3 การควบคุมการทำงานแบบเทอร์โมสแตท	4
2.1.4 การทำงานและควบคุม	5
2.1.5 การสื่อสารแบบสามสาย	6
2.2 การควบคุมกำลังงานไฟฟ้า	8
2.2.1 วงจรควบคุมเฟสชิฟต์เฟด	8
2.3 วงจรควบคุม	11
2.3.1 การควบคุมไฟฟ้าแบบตัดต่อ	13
2.4 ไมโครคอนโทรลเลอร์ MCS-51	14
2.4.1 การจัดขาของไมโครคอนโทรลเลอร์ MCS-51	14
2.4.2 โครงสร้างและการทำงานของพอร์ต	16
2.4.3 การใช้งานเป็นพอร์ตอินพุต	18
2.4.4 การใช้งานเป็นพอร์ตเอาต์พุต	19
2.4.5 การอ่านค่าลอจิกจากพอร์ต	19
2.5 รายละเอียดเกี่ยวกับโมดูล LCD	20
2.5.1 โครงสร้างภายในโมดูล LCD	20
2.5.2 โมดูล LCD ขนาด 16 ตัวอักษร 1 บรรทัด	22
2.5.3 คำสั่งควบคุม LCD โมดูล	22

เอกสารนี้เป็นเอกสารที่ 2.5.4 การเขียนคำสั่งและข้อมูลให้แก่โมดูล LCD อนุญาตให้นำไปใช้ประโยชน์ได้ 25 การค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

	2.5.5 จังหวะการทำงานของโมดูล LCD	26
	2.6 การเชื่อมต่อกับคีย์แพดหรือสวิตช์เมตริก 4x4 จุด	27
	2.6.1 การเชื่อมต่อกับคีย์แพดเข้ากับ ไมโครคอนโทรลเลอร์	28
	2.7 ไทเมอร์/เคาน์เตอร์	30
	2.7.1 การทำงานในโหมดไทเมอร์	30
	2.7.2 การทำงานในโหมดเคาน์เตอร์	30
	2.7.3 รีจิสเตอร์ที่เกี่ยวข้องกับการทำงานของไทเมอร์/เคาน์เตอร์ 0 และ 1	31
	2.7.4 โหมดการทำงานของไทเมอร์/เคาน์เตอร์ 0 และ 1	34
บทที่ 3	การออกแบบและหลักการทำงาน	38
	3.1 วงจรแหล่งจ่ายไฟกระแสตรง	40
	3.2 การวัดอุณหภูมิ โดยใช้ IC เบอร์ DS1620	40
	3.3 วงจรควบคุม	42
	3.4 วงจรขับเครื่องกำเนิดความร้อน	43
บทที่ 4	ผลการทดลอง	44
	4.1 ความถูกต้องของตัวจิจิตอลเทอร์โมมิเตอร์	57
	4.2 ผลการควบคุมความร้อน	58
	4.3 ผลการควบคุมเวลา	59
	4.4 การกระจายความร้อนรอบบริเวณที่ใช้งาน	59
บทที่ 5	สรุปและวิจารณ์ผลการทดลอง	61
	เอกสารอ้างอิง	63
	ภาคผนวก	64

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1

บทนำ

ในโรงงานอุตสาหกรรมแปรรูปพืชผลทางการเกษตรส่วนใหญ่ เครื่องอบถูกนำมาใช้ในกระบวนการอบ เพื่อให้ได้ผลิตภัณฑ์ตามที่ต้องการและรวดเร็วต่อการส่งออกจำหน่าย หรือ การถนอมอาหารต่างๆ เพื่อให้ผลิตภัณฑ์คงอยู่ได้นานขึ้น จากนั้นทำการบรรจุผลิตภัณฑ์สำเร็จเพื่อการส่งออกสินค้าทางการเกษตรที่แปรรูปแล้ว โดยเครื่องอบอัตโนมัติที่ให้ความละเอียด ความเที่ยงตรงที่ดีและสะดวกรวดเร็วในการใช้งาน เหล่านี้ต้องสั่งนำเข้ามาจากต่างประเทศ ซึ่งมีราคาแพงมาก

ด้วยเทคโนโลยีทางด้านอิเล็กทรอนิกส์และคอมพิวเตอร์ในปัจจุบันได้เข้ามามีบทบาทเป็นอย่างมากในวงการอุตสาหกรรม และด้วยความสามารถมากมายของไมโครโปรเซสเซอร์ หรือ ไมโครคอนโทรลเลอร์ ที่สามารถนำมาใช้ในระบบควบคุมการทำงานต่าง ๆ ตั้งแต่ระบบที่เล็ก ๆ ที่มีใช้ในอุปกรณ์เครื่องใช้ในชีวิตประจำวัน ไปจนถึงระดับใหญ่ ๆ ในงานอุตสาหกรรมได้เป็นอย่างดี โครงการงานนี้จึงได้นำเอาไมโครคอนโทรลเลอร์ตระกูล MCS-51 ซึ่งเป็นไมโครคอนโทรลเลอร์ที่นิยมใช้ในการควบคุมระบบขนาดกลาง มาควบคุมระบบและประมวลผลการทำงานในส่วนต่าง ๆ ของโครงการ โดยได้เลือกใช้ไมโครคอนโทรลเลอร์เบอร์ AT89C51

วัตถุประสงค์

ออกแบบและสร้างเครื่องอบอัตโนมัติที่สามารถควบคุมอุณหภูมิได้ สามารถทำการกำหนดอุณหภูมิให้ความร้อนตามที่ต้องการได้ สามารถทำการตั้งเวลาในการใช้งานตามที่ต้องการได้

ขอบเขต

โครงการนี้ได้ออกแบบสร้างเครื่องอบอัตโนมัติที่ให้ความร้อนตามที่เรากำหนดอุณหภูมิและสามารถกำหนดเวลาในการใช้งานตามที่ต้องการได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2

ทฤษฎี และหลักการ

2.1 การวัดอุณหภูมิ

การวัดอุณหภูมิในโครงงานนี้ จะใช้วงจรรวม(IC : Integrated Circuit) เพื่อทำการแปลงค่าอุณหภูมิให้เป็นข้อมูลดิจิทัล โดยใช้ IC หมายเลข DS1620 ซึ่งผลิตโดยบริษัท “DALLAS SEMICONDUCTOR” ซึ่งเป็น Digital Thermometer and Thermostat ซึ่งมีลักษณะต่าง ๆ และการใช้งานดังนี้

2.1.1 ลักษณะสำคัญ (Features)

ไม่ต้องการอุปกรณ์ต่อรวมภายนอก

วัดอุณหภูมิได้ตั้งแต่ -55°C ถึง 125°C โดยเพิ่มขึ้นละ 0.5°C

(-67°F ถึง 257°F เพิ่มขึ้นละ 0.9°F)

ให้ค่าอุณหภูมิที่แปลงแล้วเป็นดิจิทัล 9 บิต

แปลงอุณหภูมิหนึ่งค่าใช้เวลา 200 ms โดยประมาณ

สามารถตั้งอุณหภูมิในการใช้เป็นเทอร์โมสแตทได้และยังคงค่าไว้แม้จะตัดไฟเลี้ยงรับและส่งข้อมูลโดยใช้การติดต่อแบบสามสาย (CLK, DQ, RST)

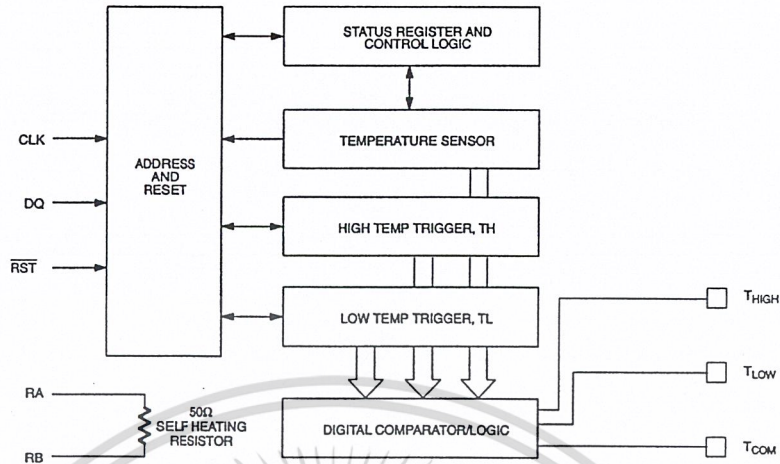


DS1620 8-Pin DIP (300-mil)
See Mech Drawings Section

PIN DESCRIPTION

DQ	- 3-Wire Input/Output
CLK/CONV	- 3-Wire Clock Input and Stand-alone Convert Input
RST	- 3-Wire Reset Input
GND	- Ground
T _{HIGH}	- High Temperature Trigger
T _{LOW}	- Low Temperature Trigger
T _{COM}	- High/Low Combination Trigger
V _{DD}	- Power Supply Voltage (3V - 5V)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับรูปที่ 2.1 ตำแหน่งขาของ DS1620 นุญาติให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.2 BLOCK DIAGRAM

2.1.2 การอ่านอุณหภูมิ

ในการอ่านอุณหภูมินั้นจะได้เป็นข้อมูล 9 บิต ซึ่งเป็นอุณหภูมิของตัวถึงขณะนั้นและเป็นแบบ 2's complement ซึ่งตารางที่ 1 เป็นตัวอย่างความสัมพันธ์ระหว่างข้อมูลกับอุณหภูมิ ข้อมูลจะถูกส่งโดยทางการติดต่อแบบสามสาย โดยจะมี LSB เป็นบิตแรก DS1620 จะให้ความต่างอุณหภูมิขั้นละ 0.5°C หรือขั้นละ 1 LSB

TEMPERATURE/DATA RELATIONSHIPS Table 1

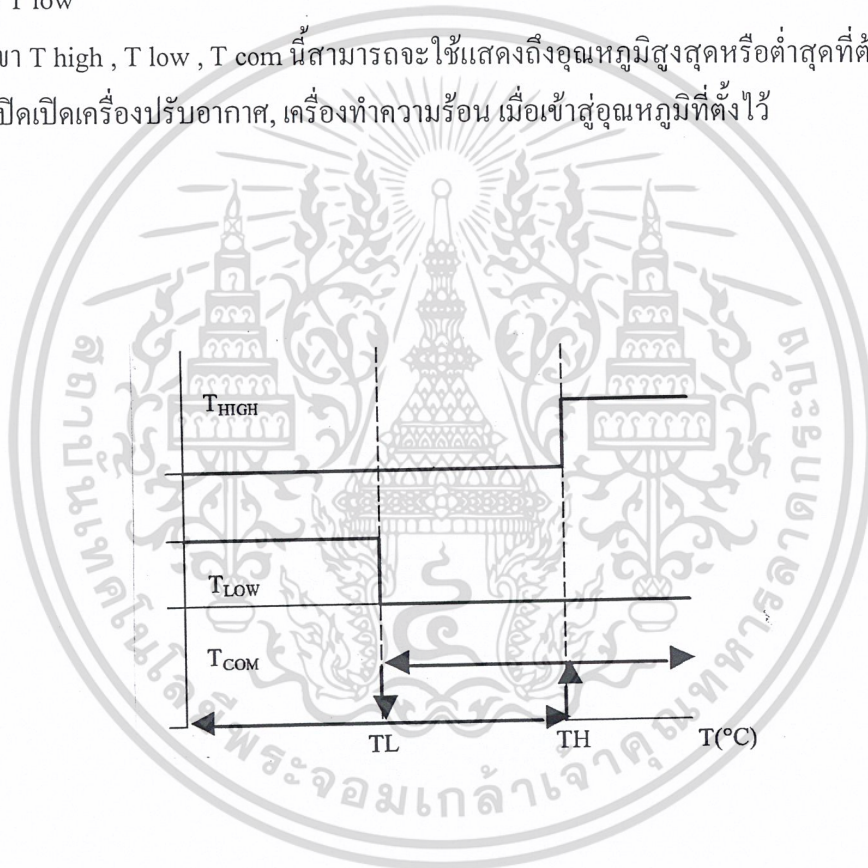
TEMP	DIGITAL OUTPUT (Binary)	DIGITAL OUTPUT (Hex)
+125°C	0 11111010	00FA
+25°C	0 00110010	0032h
+½°C	0 00000001	0001h
+0°C	0 00000000	0000h
-½°C	1 11111111	01FFh
-25°C	1 11001110	01CEh
-55°C	1 10010010	0192h

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.3 การควบคุมการทำงานแบบเทอร์โมสแตท

ขา T high , T low , T com จะจำเป็นมากเมื่อ DS1620 ถูกใช้งานในโหมดของเทอร์โมสแตท ดังแสดงในรูปที่ 2.3 เมื่ออุณหภูมิสูงกว่าหรือเท่ากับอุณหภูมิที่ตั้งไว้ในรีจิสเตอร์ที่เก็บอุณหภูมิสูงสุด (T high) ขา T high และ T com จะให้เอาต์พุตเป็น '1' และจะคงสภาวะไว้จนกว่าอุณหภูมิจริงจะลดลงต่ำกว่าค่าในรีจิสเตอร์ T high ที่ขา T high จะให้เอาต์พุตเป็น '0' แต่ขา T com ยังคงเป็น '1' อยู่จนกว่าอุณหภูมิจริงจะลดลงต่ำกว่าอุณหภูมิที่ตั้งไว้ในรีจิสเตอร์ที่เก็บอุณหภูมิต่ำสุด ขา T com จะกลายเป็น '0' แต่ขา T low จะกลายเป็น '1' และจะคงสภาวะไว้จนกว่าอุณหภูมิจริงจะสูงกว่าค่าที่ตั้งไว้ในรีจิสเตอร์ T low

ขา T high , T low , T com นี้สามารถจะใช้แสดงถึงอุณหภูมิสูงสุดหรือต่ำสุดที่ต้องการ หรือนำไปใช้ปิดเปิดเครื่องปรับอากาศ, เครื่องทำความร้อน เมื่อเข้าสู่อุณหภูมิที่ตั้งไว้



รูปที่ 2.3 THERMOSTAT OUTPUT OPERATION

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.4 การทำงานและควบคุม

DS1620 ควรมีข้อมูลรีจิสเตอร์ T high และ T low สำหรับการทำงานในโหมดเทอร์โมสเตท Configuration/Status Register จะใช้เก็บค่าที่แสดงถึงการทำงานของ DS1620 ซึ่งจะอธิบายได้ดังนี้

DONE	THF	TLF	NVB	1	0	CPU	1 SHOT
------	-----	-----	-----	---	---	-----	--------

DONE	: Conversion Done Bit ถ้าเป็น '1' คือแปลงอุณหภูมิเสร็จแล้ว ถ้าเป็น '0' คือกำลังแปลงอุณหภูมิอยู่
THF	: Temperature High Flag จะเป็น '1' เมื่ออุณหภูมิจริงมากกว่าหรือเท่ากับอุณหภูมิในรีจิสเตอร์ T high และจะเป็น '0' ได้เมื่อตัดไฟเลี้ยงหรือเขียน '0' ทับลงไป ตำแหน่งนี้
TLF	: Temperature Low Flag จะเป็น '1' เมื่ออุณหภูมิจริงน้อยกว่าหรือเท่ากับอุณหภูมิในรีจิสเตอร์ T low และจะเป็น '0' ได้เมื่อตัดไฟเลี้ยงหรือเขียน '0' ทับลงไป ตำแหน่งนี้
NVB	: Nonvolatile Memory Busy Flag จะเป็น '1' เมื่อ E ² Memory กำลังถูกเขียนอยู่ และจะเป็น '0' เมื่อไม่ถูกใช้งาน
CPU	: CPU Use Bit ถ้าเป็น '0' แสดงว่า CLK/CONV จะถูกใช้เป็นขาที่สั่งให้เริ่มแปลงอุณหภูมิ ถ้าเป็น '1' แสดงว่า CLK/CONV จะใช้เป็นขา Clock ธรรมดาโดยใช้งานร่วมกับขา DQ และ RST ในการสื่อสารแบบสามสาย สำหรับบิตนี้สามารถเก็บแบบไม่ต้องการแรงดันในการ รักษาข้อมูลไว้
1 SHOT	: One-Shot Mode ถ้าเป็น '1' จะเป็นการแปลงอุณหภูมิครั้งเดียวเมื่อได้รับคำสั่ง (Protocol) ถ้าเป็น '0' จะเป็นการแปลงอุณหภูมิแบบต่อเนื่อง

ในการใช้ DS1620 เป็นเทอร์โมสเตท จะเป็นการแปลงอุณหภูมิแบบต่อเนื่อง อย่างไรก็ตาม สำหรับการใช้งานที่ต้องอ่านทีละครั้งในขณะที่ต้องการ และยังต้องการที่จะประหยัดพลังงานควรจะใช้ใช้งานในโหมด 1 SHOT

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.5 การสื่อสารแบบสามสาย (3 – WIRE COMMUNICATION)

BUS 3 สายนี้จะป็นสัญญาณสามลักษณะ คือ RST (reset) signal, CLK (clock) signal, DQ (data) signal การสื่อสารทั้งหมดจะเริ่มขึ้นได้เมื่อขา RST ได้รับลอจิก '1' ถ้าเป็น '0' จะเป็นการตัดการสื่อสาร สัญญาณ CLK คือ สัญญาณนาฬิกาที่มีขอบขาลงแล้วตามด้วยขอบขาขึ้น และสำหรับสัญญาณ Data ที่จะสื่อสารนั้น ควรจะมีระดับแรงดันของแต่ละบิตตรงกับขอบขาขึ้นของสัญญาณ Clock และในขณะที่ขา CLK เป็น '1' จะทำให้ขา DQ เป็นสภาวะ High Impedance เมื่อขา RST เป็น '0' จะเป็นการตัดการสื่อสารทั้งหมดและขา DQ จะมีสภาวะเป็น High Impedance ในการสื่อสารแบบสามสายนี้จะใช้ LSB เป็นบิตแรก คำสั่งที่ใช้ในการติดต่อแบบสามสายจะให้มาในตารางที่ 3 ซึ่งควรจะใช้คำสั่งเหล่านี้เท่านั้นที่จะเขียนลงไปใน DS1620 ถ้าเป็นคำสั่งอื่นๆ อาจจะทำให้ DS1620 เสียหายได้ ซึ่งคำสั่งต่างๆ จะอธิบายได้ดังนี้

Read Temperature [AAh]

เป็นคำสั่งที่จะอ่านข้อมูลอุณหภูมิที่แปลงแล้ว โดยหลังจากใส่คำสั่งนี้ จะได้ค่าอุณหภูมิ 9 บิต จากรีจิสเตอร์ออกมา

Write TH [01h]

เป็นการสั่งให้เขียนค่าลงใน TH register โดยหลังจากใส่คำสั่งนี้แล้ว 9 บิตต่อมา จะถูกตั้งเป็นค่าที่ใช้จำกัดอุณหภูมิสูงสุดของ O/P T high

Write TL [02h]

เป็นการสั่งให้เขียนค่าลงใน TL register โดยหลังจากใส่คำสั่งนี้แล้ว 9 บิตต่อมา จะถูกตั้งเป็นค่าที่ใช้จำกัดอุณหภูมิสูงสุดของ O/P T low

Read TH [A1h]

เป็นคำสั่งที่ใช้อ่านค่าใน TH register โดยหลังจากใส่คำสั่งนี้เข้าไป จะได้ข้อมูล 9 บิต ที่เป็นค่าอุณหภูมิสูงสุดในการทำงานของ T high O/P

Read TL [A2h]

เป็นคำสั่งที่ใช้อ่านค่าใน TL register โดยหลังจากใส่คำสั่งนี้เข้าไป จะได้ข้อมูล 9 บิต ที่เป็นค่าอุณหภูมิต่ำสุดในการทำงานของ T low O/P

Start Convert [EEh]

เป็นคำสั่งที่ใช้ให้เริ่มการแปลงอุณหภูมิ โดยในโหมด 1 Shot นั้น DS1620 จะทำการแปลงอุณหภูมิค่าเดียวและเข้าสู่สภาวะนิ่งส่วนในโหมดต่อเนื่องจะเป็นการสั่งให้เริ่มแปลงอุณหภูมิไปเรื่อยๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Stop Convert [22h]

เป็นคำสั่งให้หยุดการแปลงอุณหภูมิในโหมดต่อเนื่อง หลังจากใส่คำสั่งนี้จะแปลงอุณหภูมิค่าสุดท้ายในขณะนั้นให้เสร็จแล้วจะเข้าสู่สภาวะนิ่ง จนกว่าจะได้รับคำสั่ง Start Convert จึงจะเริ่มแปลงค่าอุณหภูมิอย่างต่อเนื่องต่อไป

Write Config [0Ch]

เป็นคำสั่งที่ใช้เพื่อจะเขียนค่าเงื่อนไขการทำงานลงใน Config register หลังจากใส่คำสั่งนี้แล้ว 8 บิตต่อจากคำสั่งจะต้องเป็นค่า Config ที่ต้องการเขียนลงไป

Read Config [ACh]

เป็นคำสั่งที่ใช้อ่านค่าจาก Config register โดยหลังจากใส่คำสั่งนี้แล้วจะได้ข้อมูล 8 บิตออกมา ซึ่งเป็นใน Config register

INSTRUCTION	DESCRIPTION	PROTOCOL	3-WIRE BUS DATA AFTER ISSUING PROTOCOL	NOTES
TEMPERATURE CONVERSION COMMANDS				
Read Temperature	Reads last converted temperature value from temperature register.	AAh	<read data>	
Read Counter	Reads value of count remaining from counter.	A0h	<read data>	
Read Slope	Reads value of the slope accumulator.	A9h	<read data>	
Start Convert T	Initiates temperature conversion.	EEh	Idle	1
Stop Convert T	Halts temperature conversion.	22h	Idle	1
THERMOSTAT COMMANDS				
Write TH	Writes high temperature limit value into TH register.	01h	<write data>	2
Write TL	Writes low temperature limit value into TL register.	02h	<write data>	2
Read TH	Reads stored value of high temperature limit from TH register.	A1h	<read data>	2
Read TL	Reads stored value of low temperature limit from TL register.	A2h	<read data>	2
Write Config	Writes configuration data to configuration register.	0Ch	<write data>	2
Read Config	Reads configuration data from configuration register.	ACh	<read data>	2

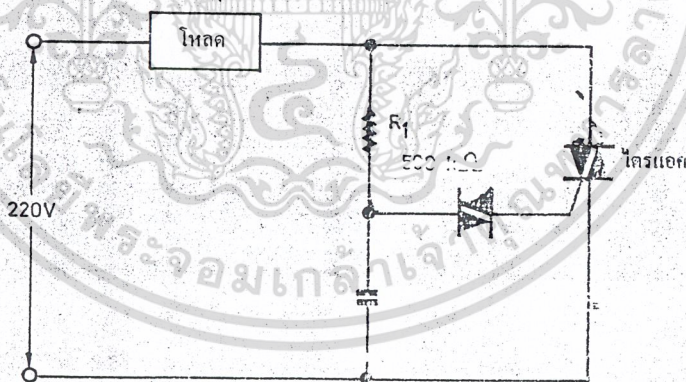
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
 รูปที่ 2.4 ตารางคำสั่ง

2.2 การควบคุมกำลังงานไฟฟ้า

ระบบไฟฟ้ากำลังที่ใช้ในโรงงานอุตสาหกรรมส่วนใหญ่เป็นระบบ 380 โวลต์ สามเฟส หรือระบบเฟสเดียว 200 โวลต์ อุปกรณ์ไทรสเตอร์และอิเล็กทรอนิกส์กำลังในปัจจุบันหลายชนิดได้รับการออกแบบให้มีประสิทธิภาพในการทนกระแสและแรงดันได้สูงขึ้น การใช้อุปกรณ์สารกึ่งตัวนำเป็นสวิตช์ในการเปิดปิดระบบไฟฟ้ากำลังอย่างรวดเร็วและได้ผลจึงทำให้มีผู้หันมานิยมใช้กันมากขึ้น การควบคุมกำลังงานไฟฟ้าเพื่อให้เหมาะสมกับการใช้งานเป็นสิ่งจำเป็นอย่างยิ่งต่อการควบคุมการทำงาน เพื่อให้ผลที่ต้องการ เช่น การควบคุมพลังงานจ่ายให้กับเครื่องกำเนิดความร้อนเพื่อควบคุมอุณหภูมิ ควบคุมกำลังสำหรับจ่ายให้ระบบแสงสว่างเพื่อควบคุมปริมาณแสง ควบคุมกำลังงานจ่ายให้มอเตอร์เพื่อควบคุมความเร็วของการหมุน เป็นต้น

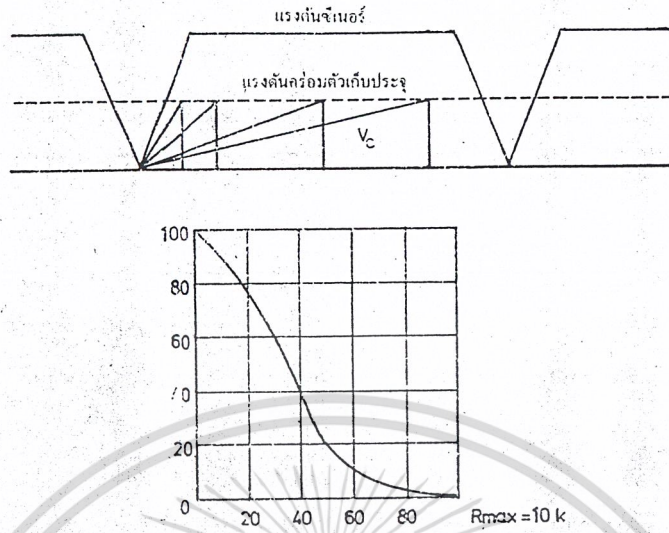
2.2.1 วงจรควบคุมเฟสชนิดฟูลเวฟ

การควบคุมเฟสแบบฟูลเวฟแบบง่ายและใช้ในกรณีกระแสไม่สูงมากนัก มักใช้ไทรแอก การควบคุมเฟสในกรณีนี้ ถ้าเป็นกรณีพื้นฐานเบื้องต้นที่ไม่ต้องการความนุ่มนวลในการควบคุมกำลังงาน เรามักใช้วิธีการของ RC เฟสชิฟท์เป็นตัวสร้างทริกให้แก่ไทรแอก ดังรูปที่ 2.6



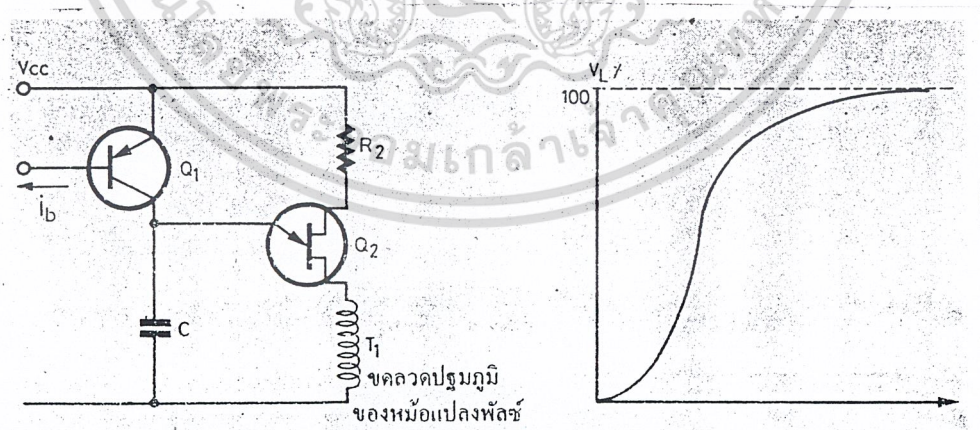
รูปที่ 2.6 วงจรควบคุมเฟสใช้ไทรแอก

วงจรในรูปที่ 2.6 สามารถควบคุมกำลังงานไฟฟ้าจากค่าน้อยที่สุดไปยังค่ามากที่สุดได้ แต่ช่วงการควบคุมไม่เป็นเชิงเส้น และการปรับค่า R จะอยู่ในช่วงที่กว้างมาก จึงไม่สะดวกต่อการใช้งานในการควบคุมอัตโนมัติ เช่น นำสัญญาณควบคุมมาจากแสงด้วยอุปกรณ์ LDR จากอุณหภูมิด้วยเทอร์มิสเตอร์ (thermistor) ฯลฯ เพราะอุปกรณ์ตรวจจับสัญญาณเหล่านี้มักจะมีช่วงการแปรค่าไว้มากนัก การควบคุมในกรณีนี้จำเป็นต้องสร้างวงจรควบคุมการจุดทริกเกทเสียใหม่



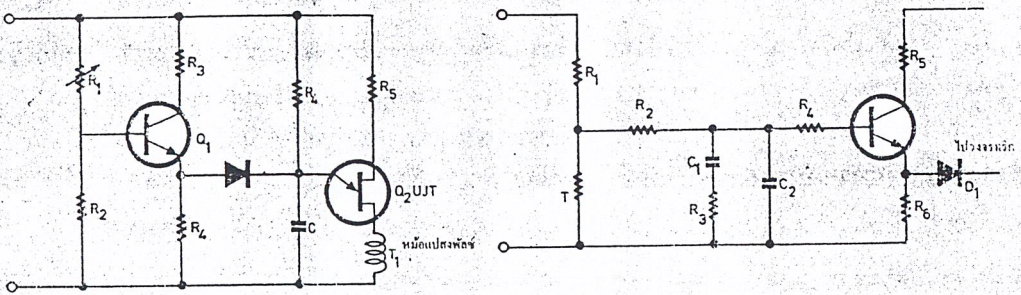
รูปที่ 2.7 ลักษณะของแรงดันคร่อมตัวเก็บประจุและ % V_L

การใช้ BJT เป็นวงจรถริกทำให้การทรिकค่อนข้างจะราบเรียบขึ้น และสามารถควบคุมได้ในกรณีนี้ เราสามารถควบคุมแรงดันคร่อมตัวเก็บประจุด้วยการควบคุมกระแส และการควบคุมนี้ถ้าใช้วงจรถักกระแสที่ที่แปรค่าได้ ผลของการควบคุมจะเป็นเชิงเส้นได้ และการควบคุมก็สามารถนำมาจากสัญญาณควบคุมอื่นใดก็ได้ ดังรูปที่ 2.8



รูปที่ 2.8 แสดงการใช้กระแส i_b ควบคุมกำลังงาน

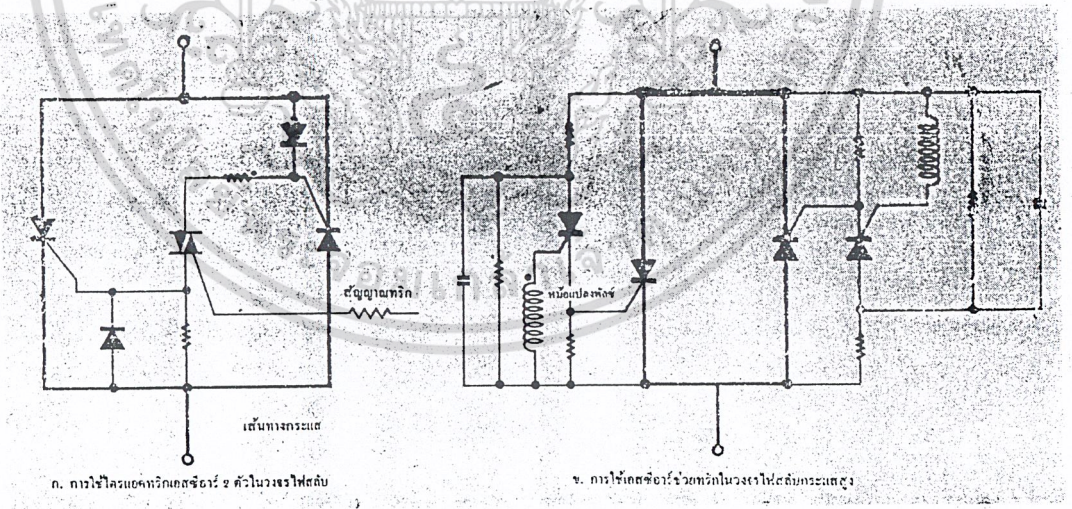
ด้วยกรณีเช่นนี้เราจึงใช้สัญญาณการควบคุมด้วยการใช้ทรานซิสเตอร์เป็นตัวควบคุมการทำงานของ BJT อีกต่อ จึงได้ดังตัวอย่างวงจรในรูปที่ 2.9



รูปที่ 2.9 การใช้เทอร์มิสเตอร์ควบคุมกำลังงานไฟฟ้า

จากกรณีนี้เราสามารถนำหลักการควบคุมกำลังงานมาควบคุมอุณหภูมิได้ โดยใช้เทอร์มิสเตอร์เป็นอุปกรณ์ตรวจวัดค่าอุณหภูมิ กล่าวคือ เมื่ออุณหภูมิสูงก็ควบคุมให้กำลังงานน้อยลงและถ้าอุณหภูมิต่ำก็ควบคุมให้กำลังงานเพิ่มขึ้น

ในกรณีที่ใช้กระแสสูง ๆ เราไม่สามารถที่จะใช้ไดรแอกได้เพราะไดรแอกกระแสสูง ๆ หาได้ยากและมีราคาแพงมาก ดังนั้นจึงจำเป็นต้องนำเอาเอสซีอาร์ 2 ตัวมาต่อแทนไดรแอกแทนลักษณะการต่อเอสซีอาร์ในวงจรควบคุมไฟสลับเป็นดังรูปที่ 2.10

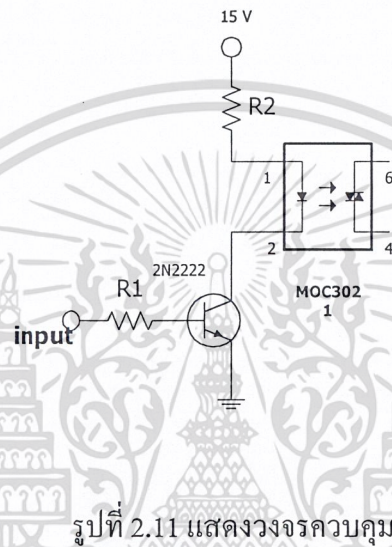


รูปที่ 2.10 การใช้เอสซีอาร์ แทนไดรแอก

การทำงานของทรานซิสเตอร์สำหรับเอสซีอาร์กระแสสูงมักต้องมีการขยายกระแสด้วยการใช้ไดรแอกหรือเอสซีอาร์อีกตัวเสียก่อน ทั้งนี้เพราะในกรณีเอสซีอาร์กระแสสูงจำเป็นต้องใช้กระแสในการทรานซิสเตอร์สูงด้วยเช่นกัน อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3 วงจรควบคุม

วงจรควบคุม (Control Circuit) เป็นวงจรที่รับสัญญาณที่ผ่านการประมวลผลมาจากไมโครคอนโทรลเลอร์ ซึ่งจะมีลักษณะ เป็น สัญญาณ Pulse แบบ Rectangular wave โดยจะนำสัญญาณที่ได้ไปทำการควบคุมการทำงานในส่วนของวงจรขับกำลังงานตัวนำความร้อน (Drive Heater Circuit) เพื่อเป็นการควบคุมกำลังงานเฉลี่ยให้กับ Heater ในการทำความร้อนให้ได้ตามต้องการ



รูปที่ 2.11 แสดงวงจรควบคุม

โดยการทำงานของวงจรควบคุม จะเริ่มต้นจากการรับสัญญาณมาจากไมโครคอนโทรลเลอร์ จากนั้นจะส่งผ่านสัญญาณ โดยใช้อุปกรณ์ส่งสัญญาณทางแสงที่อยู่ในรูปของไอซี เบอร์ MOC 3021 ซึ่งจะทำการแยกส่วนของวงจรควบคุม(Control Circuit) และส่วนของวงจรขับกำลังงานตัวนำความร้อน(Drive Heater Circuit) ส่งสัญญาณผ่านทางแสงภายในตัวไอซี เพื่อทำการหลีกเลี่ยงและป้องกันโอกาสที่จะเกิดความเสียหายแก่วงจรเนื่องจากในส่วนของ Drive Heater Circuit จะเป็นส่วนที่กระแส และแรงดันอยู่ในระดับที่สูง (high current and high voltage) ซึ่งหากไม่มีการแยกวงจร อาจจะทำให้เกิดความเสียหายขึ้นกับ Control Circuit และวงจรในส่วนของDigital ได้

ในการส่งสัญญาณ โดยใช้ MOC 3021 จะต้องนำ Transistor มาเป็นตัวรับสัญญาณ Pulse จากไมโครคอนโทรลเลอร์ เนื่องจาก 2 ประการด้วยกัน

1. เพื่อเป็นการขับให้กระแสเพียงพอในการทำให้ MOC 3021 สามารถทำงานได้เพราะหากไม่มี Transistor มาช่วยในการขับกระแส อาจจะทำให้กระแสไม่เพียงพอที่จะทำให้ MOC 3021 เกิดการทำงาน ทั้งนี้เนื่องจากสัญญาณที่ออกมาจากวงจรประมวลผล ที่เป็นไมโครคอนโทรลเลอร์ซึ่งเป็น

อุปกรณ์ทาง Digital ที่มีการไหลของกระแสน้อย

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์การเชิงงานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

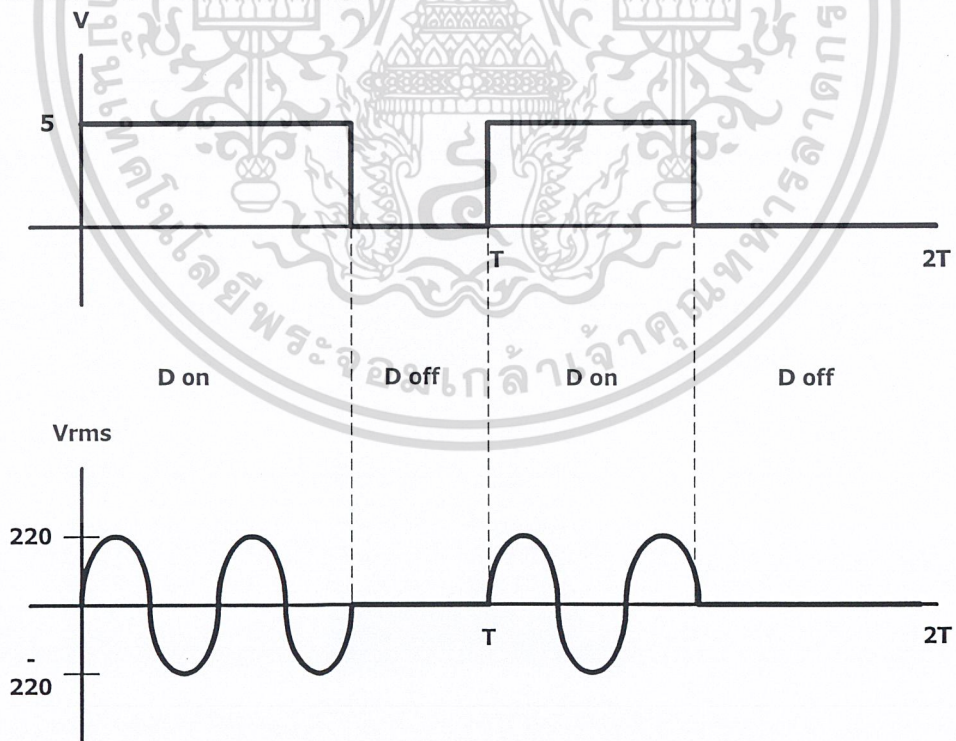
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แต่การนำ Transistor มาใช้นั้นจะมีการนำตัวความต้านทาน(Resistor) มาเป็นตัวควบคุมปริมาณของกระแสให้เหมาะสม โดยมี

R1 : เพื่อควบคุมปริมาณกระแสทางด้าน input เป็นตัวกำหนดกระแสเบสเพื่อให้ได้กระแสคอลเลกเตอร์ให้เพียงพอตามต้องการในการขับส่วนของ MOC 3021 และเพื่อให้วงจรประมวลผลเกิดความเสียหาย เนื่องจากกระแสสูงเกินไปเมื่อ Transistor เกิดความเสียหายและเกิด Short circuit

R2 : เพื่อควบคุมปริมาณกระแสที่จะขับให้ MOC 3021 ทำงาน และในกรณีที่ Transistor เกิดความเสียหาย และ Short circuit จะเป็นตัวควบคุมไม่ให้กระแสสูงเกินไป จนถึงระดับที่ MOC 3021 ไม่สามารถที่จะรับได้

- เพื่อให้สัญญาณที่ออกจาก MOC 3021 สัมพันธ์กับสัญญาณ Pulse ที่ได้รับมาจากส่วน วงจรประมวลผล เนื่องจากภายใน MOC 3021 ที่ต่อกับ Transistor เป็น ไดโอดเปล่งแสง เพราะฉะนั้นเมื่อมีสัญญาณ Pulse เข้ามาจะทำให้ Transistor ทำงาน (มีการไหลกระแส) เป็นผลให้ไดโอดภายใน MOC 3021 เปล่งแสง (ซึ่งเป็นการส่งสัญญาณทางแสง) ทำให้เกิดสัญญาณ output ที่สัมพันธ์กับสัญญาณ input สัญญาณ output ที่ได้จะส่งไปที่ส่วนของ Drive Heater Circuit เพื่อให้เกิดการทำงานได้ จะสามารถสังเกตได้จากกราฟ



จากกราฟจะเห็นได้ถึงลักษณะการทำงานในส่วนของ Drive Heater Circuit โดยการนำ

MOC 3021 มาช่วยในการส่งผ่านสัญญาณ และทำการแยกส่วนของวงจร Drive Heater Circuit ออก

จากส่วนอื่นๆ

เอกสารนี้เป็นเอกสารลิขสิทธิ์ของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี หากมีข้อสงสัยหรือต้องการข้อมูลเพิ่มเติม กรุณาติดต่อฝ่ายวิชาการ
 ไม่ว่าที่พิมพ์ขึ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

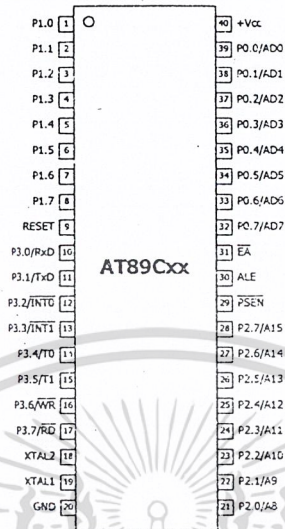
2.3.1 การควบคุมไฟฟ้าแบบตัดต่อ (mark – space)

การควบคุมแบบตัดต่อเหมาะสมสำหรับโหลดที่มีความเฉื่อยสูง หรือระบบที่ให้ความร้อนที่มีค่าคงตัวเวลาสูง การควบคุมเช่นนี้มีข้อดี คือการตัดต่อสวิตช์กระทำที่กระแสเป็น 0 จึงสามารถลดปัญหาในการเกิดฮาร์มอนิกส์ หรือการแทรกสอดทางแม่เหล็กไฟฟ้า (EMI) จึงนำวิธีการควบคุมแบบนี้มาใช้เพราะหากใช้วิธีการควบคุมกำลังงานแบบเฟสอาจจะทำให้เกิดการรบกวนสัญญาณต่างๆได้ โดยการแทรกสอดทางแม่เหล็กไฟฟ้า ทั้งนี้เนื่องจาก Heater ที่ใช้มีกระแสสูง ซึ่งวิธีการควบคุมแบบเฟสจะทำให้เกิดความถี่สูง เมื่อใช้กับอุปกรณ์ที่มีกระแสสูงจะทำให้การสัญญาณรบกวน แบบ EMI ที่รุนแรง อย่างไรก็ตาม การควบคุมจะขาดความแม่นยำ ถ้าค่าคงตัวของระบบมีค่าไม่สูงนักเมื่อเปรียบเทียบกับคาบเวลาของแรงดันไฟฟ้ากระแสสลับ แต่สำหรับในที่นี่ได้นำ ไมโครคอนโทรลเลอร์ มาเป็นอุปกรณ์ประมวลผล เพื่อสร้างสัญญาณ Pulse ในการควบคุมกำลังงาน จึงสามารถช่วยให้เกิดความแม่นยำที่สูงขึ้นได้ อีกทั้ง ไอซีเบอร์ MOC 3021 ก็เป็นไอซีส่งสัญญาณทางแสง แบบ Zero crossing ด้วย จึงช่วยให้สัญญาณควบคุม output จาก MOC 3021 จะเป็นศูนย์ เมื่อแรงดันไฟฟ้ากระแสสลับตัดที่จุด Zero crossing

สัญญาณควบคุมที่ใช้ในการควบคุมกำลังงานนั้น จะได้มาจากการประมวลผลของ ไมโครคอนโทรลเลอร์ โดยจะรับค่าอุณหภูมิที่ต้องการ และค่าอุณหภูมิที่วัดได้ เพื่อให้ได้ระดับอุณหภูมิที่ต้องการ และมีความเสถียรให้ได้มากที่สุด

2.4 ไมโครคอนโทรลเลอร์ MCS-51

2.4.1 การจัดขาของไมโครคอนโทรลเลอร์ MCS-51



รูปที่ 2.12 การจัดขาของไมโครคอนโทรลเลอร์ MCS-51

ทุกเบอร์จะมีการจัดเรียงขาเหมือนกันดังรูปที่ 2.12 ดังนี้

ขา Vcc ใช้สำหรับต่อไฟเลี้ยง
ขา GND เป็นสายดิน

ขาพอร์ต 0 (P0.0-P0.7) แต่ละขาสามารถเป็นได้ทั้งอินพุตและเอาต์พุตสำหรับใช้งานทั่วไป ถ้าต้องการให้ขาพอร์ต 0 ขาใดขาหนึ่งเป็นอินพุต ทำได้โดยเขียนข้อมูล 1 ไปยังแต่ละพอร์ตที่ต้องการติดต่อด้วย ทำให้ขามีสถานะปล่อยลอย(float) นอกจากนั้นยังใช้งานในการติดต่อกับขาแอดเดรสไบต์ต่ำของหน่วยความจำภายนอก (A0-A7) และขาข้อมูล (D0-D7) โดยใช้กระบวนการ multiplex เข้าช่วยเพื่อสลับการทำงานให้เป็นที่ทั้งขาติดต่อกับขาแอดเดรสไบต์ และขาข้อมูล

ขาพอร์ต 1 (P1.0-P1.7) แต่ละขาสามารถเป็นได้ทั้งอินพุตและเอาต์พุตสำหรับใช้งานทั่วไป ถ้าต้องการให้ขาพอร์ต 1 ขาใดขาหนึ่งเป็นอินพุต ทำได้โดยเขียนข้อมูล 1 ไปยังแต่ละพอร์ตที่ต้องการติดต่อด้วย นอกจากนั้นในอนุกรม AT89SXX ยังใช้ขา P1.0 เป็นขาอินพุตสำหรับนับค่าของไทมเมอร์ 2 (T2) และ P1.1 เป็นขาอินพุต ทรigger ของไทมเมอร์ 2 (T2) ส่วนขา P1.4-P1.7 เป็นขาสำหรับเชื่อมต่อแบบ SPI เพื่อทำงานโปรแกรมข้อมูลในระบบ

ขาพอร์ต 2 (P2.0-P2.7) แต่ละขาสามารถเป็นได้ทั้งอินพุตและเอาต์พุตสำหรับใช้งานทั่วไป ถ้าต้องการให้ขาพอร์ต 2 ขาใดขาหนึ่งเป็นอินพุต ทำได้โดยเขียนข้อมูล 1 ไปยังแต่ละพอร์ตที่ต้องการติดต่อด้วยทำให้ขามีสถานะปล่อยลอย(float) นอกจากนั้นยังใช้งานในการติดต่อกับขาแอดเดรสไบต์

สูงของหน่วยความจำภายนอก (A8-A15)
เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้สำหรับอาจารย์งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขาพอร์ต 3 (P3.0-P3.7) แต่ละขาสามารถเป็นได้ทั้งอินพุตและเอาต์พุตสำหรับใช้งานทั่วไป ถ้าต้องการให้ขาพอร์ต 3 ขาใดขาหนึ่งเป็นอินพุต ทำได้โดยเขียนข้อมูล 1 ไปยังแต่ละพอร์ตที่ต้องการ ติดต่อด้วยทำให้ขาที่มีสถานะปล่อยลอย(float)) นอกจากนั้นยังใช้งานในหน้าที่พิเศษดังนี้

P3.0 สามารถเป็นขาอินพุต สำหรับรับข้อมูลจากการสื่อสารแบบอนุกรม หรือขา RxD

P3.1 สามารถเป็นขาอินพุต สำหรับส่งข้อมูลจากการสื่อสารแบบอนุกรม หรือขา TxD

P3.2 สามารถเป็นขาอินพุต สำหรับรับสัญญาณอินเตอร์รัปต์จากภายนอกช่องที่ 0 หรือ ขา INTO

P3.2 สามารถเป็นขาอินพุต สำหรับรับสัญญาณอินเตอร์รัปต์จากภายนอกช่องที่ 1 หรือ ขา INT1

P3.3 สามารถเป็นขาอินพุต สำหรับรับสัญญาณไทมเมอร์จากภายนอกช่องที่ 0 หรือขา TO

P3.4 สามารถเป็นขาอินพุต สำหรับรับสัญญาณไทมเมอร์จากภายนอกช่องที่ 1 หรือขา T1

P3.5 สามารถเป็นขาสัญญาณ WR ในกรณีที่ใช้เชื่อมต่อกับหน่วยความจำภายนอก

P3.6 สามารถเป็นขาสัญญาณ RD ในกรณีที่ใช้เชื่อมต่อกับหน่วยความจำภายนอก

ขารีเซต ใช้ในการรีเซตการทำงาน โดยการป้อนสัญญาณ 1 อย่างน้อย 2 เมกซ์ซินไซเคล โดยที่ clk ยังทำงานเป็นปกติ

ขา ALE/PROG (Address Latch Enable / Program Pulse Input) เป็นขาที่ใช้ควบคุมการแลตซ์ของขาพอร์ต 0 ในกรณีที่ใช้งานกับหน่วยความจำภายนอก นอกจากนั้นยังเป็นขา รับ Pulse ของการโปรแกรมข้อมูลลงใน MCS-51 รุ่นที่มีหน่วยความจำแบบ EPROM

ขา PSEN (Program Store Enable) ใช้ในการส่งสัญญาณเพื่อร้องขอติดต่อกับหน่วยความจำภายนอก เมื่อ MCS-51 ต้องการอ่านข้อมูลจากหน่วยความจำภายนอก โดย MCS-51 จะต้องส่งสัญญาณออกที่ขานี้ 2 ครั้งในแต่ละเมกซ์ซินไซเคล หากต้องการอ่านข้อมูลจากหน่วยความจำภายในจะไม่มี การส่งสัญญาณใดๆออกมาเลย

ขา EA/Vpp (External Access Enable / Programming Voltage Input) ใช้ในการเลือกติดต่อกับหน่วยความจำภายนอกหรือหน่วยความจำภายในของ MCS-51 หากขานี้เป็น 1 เป็นการเลือกติดต่อกับหน่วยความจำภายนอก แต่หากขานี้เป็น 0 เป็นการเลือกติดต่อกับหน่วยความจำภายใน นอกจากนี้ยังเป็นขา รับแรงดันไฟสูงสำหรับการโปรแกรมหน่วยความจำภายในของ MCS-51

ขา XTAL1, XTAL2 เป็นขาสำหรับต่อขาคริสตอลเพื่อสร้าง clk ในการกำหนดจังหวะการทำงาน ของ MCS-51

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

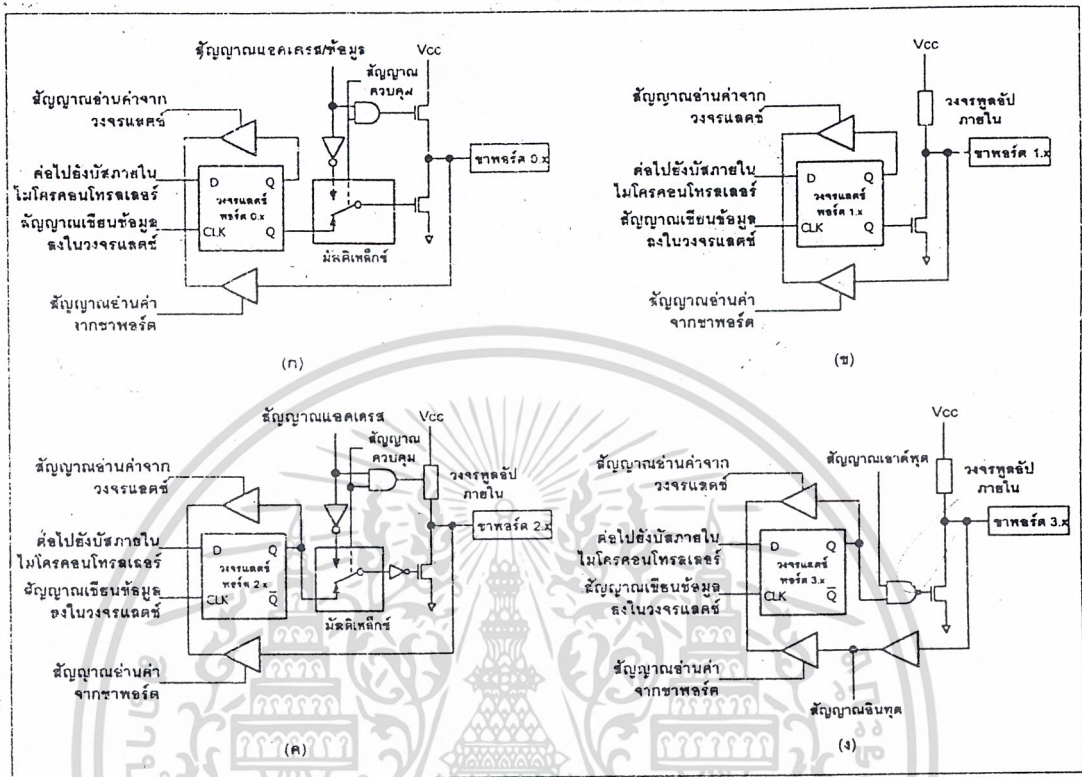
2.4.2 โครงสร้างและการทำงานของพอร์ต

พอร์ตใช้งานมีทั้งหมด 4 พอร์ต คือ พอร์ต0 พอร์ต1 พอร์ต2 พอร์ต3 โดยแต่ละพอร์ตจะมีขนาด 8 บิตและเป็นพอร์ต 2ทิศทาง คือสามารถเป็นทั้งอินพุตและเอาต์พุต ทุกพอร์ตจะมีวงจรถ่ายและวงจรถับ บัฟเฟอร์อินพุต

ที่พอร์ต 0 , 2 สามารถเป็นทั้งพอร์ตอินพุตและพอร์ตเอาต์พุตสำหรับใช้งานทั่วไป และใช้ติดต่อกับหน่วยความจำภายนอก สำหรับพอร์ต 3 และพอร์ต 1 บางขานนอกจากจะเป็นทั้งพอร์ตอินพุตและพอร์ตเอาต์พุตแล้วยังทำงานในหน้าที่พิเศษดังตาราง

ขา	เบอร์ของไมโครคอนโทรลเลอร์	หน้าที่พิเศษ
P1.0	AT89C52/AT89SXX	ขา T2 เป็นขาอินพุตนับค่าไทมเมอร์/เคาน์เตอร์ 2 และเป็นขาเอาต์พุตของแหล่งกำเนิดสัญญาณนาฬิกาโดยไทมเมอร์ 2
P1.1	AT89C52/AT89SXX	ขา T2EX เป็นขาอินพุตทริกเกอร์ สำหรับการแคปเจอร์/รีโหลด/ควบคุมทิศทางของสัญญาณ
P1.2	AT89SXX	ขา SS เป็นขาเลือกการติดต่อในกรณีที่ไม่โครคอนโทรลเลอร์เป็นอุปกรณ์สเลฟในระบบการติดต่อแบบ SPI
P1.3	AT89SXX	ขาMOSI (Master data output) ใช้ในการติดต่อกับพอร์ต SPI
P1.6	AT89SXX	ขาMOSO (Master data input) ใช้ในการติดต่อกับพอร์ต SPI
P1.7	AT89SXX	ขา SCK (Master clock output) เป็นขาสัญญาณนาฬิกาของการติดต่อกับพอร์ต SPI

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

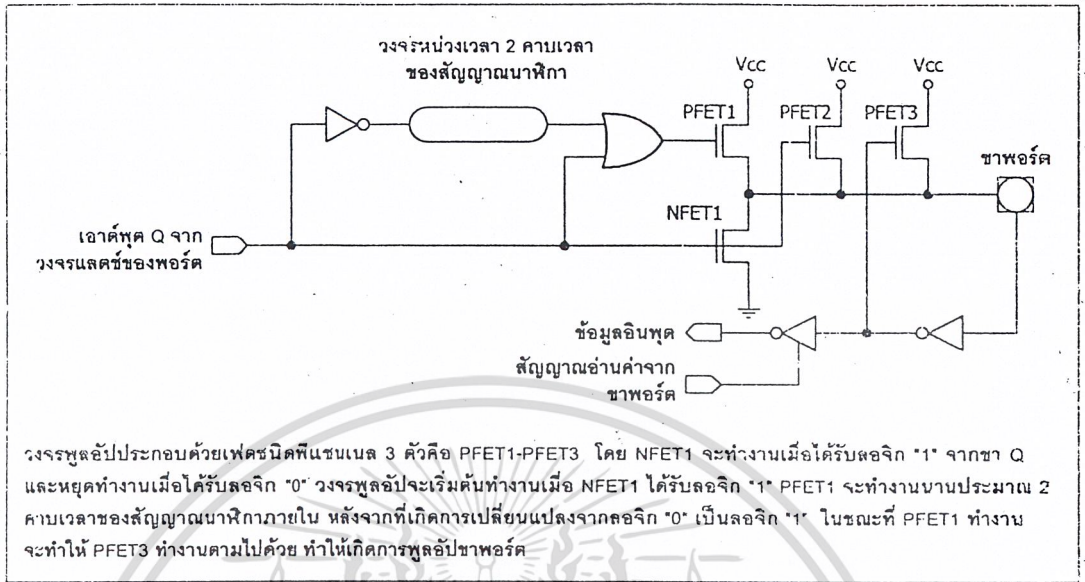


รูปที่ 2.13 แสดงวงจรภายในของแต่ละพอร์ตในไมโครคอนโทรลเลอร์

โดยรูป ก เป็นวงจรพอร์ต 0 วงจรแลตซ์แต่ละบิต คือ วงจรดีฟลิปฟลอป การอ่านสถานะของพอร์ตและสถานะของแลตซ์กระทำได้อย่างอิสระต่อกันด้วยสัญญาณที่แยกจากกัน ส่วนการเขียนข้อมูลยังพอร์ตที่ต้องการส่งสัญญาณมายังขา clk ของวงจรดีฟลิปฟลอป โดยขณะที่ข้อมูลจะผ่านทางขา บัสของข้อมูลภายในเข้าสู่ขา D ของวงจรดีฟลิปฟลอป ที่พอร์ตนี้มีวงจรมัลติเพล็กซ์สำหรับกำหนดลักษณะการทำงานของพอร์ตว่าต้องการใช้งานเป็นพอร์ตอินพุตปกติหรือใช้ติดต่อกับหน่วยความจำภายนอกของไมโครคอนโทรลเลอร์

เนื่องจากขาพอร์ต 0 ไม่มีวงจรพูลอัพภายใน หากมีการนำพอร์ต 0 ไปใช้งานเป็นพอร์ตอินพุตจะต้องต่อตัวต้านทานพูลอัพภายนอกเข้าที่พอร์ต 0 ทุกขาด้วย

รูป ข เป็นวงจรของพอร์ต 1 มีลักษณะคล้ายพอร์ต 0 หากแต่ไม่มีวงจรมัลติเพล็กซ์ เนื่องจากพอร์ตนี้จะไม่ใช่ติดต่อกับหน่วยความจำภายนอกแต่จะมีวงจรพูลอัพภายในที่แต่ละบิตของพอร์ตนี้ไม่แทน ส่วนรูปของวงจรพูลอัพภายในแสดงดังรูป



รูปที่ 2.14 วงจรพัลส์ภายใน

รูป ก เป็นวงจรภายในของพอร์ต 2 จะมีลักษณะคล้ายพอร์ต 0 มากๆ ต่างเพียงมีวงจรพัลส์เพิ่มเติมมา

รูป ง เป็นวงจรภายในของพอร์ต 3 มีลักษณะคล้ายพอร์ต 1 มีการเพิ่มเติมวงจรบัฟเฟอร์และวงจรอินพุตเอาต์พุตเมื่อทำงานในฟังก์ชันพิเศษ

2.4.3 การใช้งานเป็นพอร์ตอินพุต

การกำหนดเป็นพอร์ตอินพุตต้องเริ่มต้นด้วยการเขียนข้อมูล 1 มาที่แต่ละบิตของพอร์ตที่ต้องการใช้งานเป็นอินพุต เพื่อหยุดการทำงานของเฟตที่ใช้ในการขับสัญญาณเอาต์พุตของบิตนั้นๆ ทำให้ขาสัญญาณของพอร์ตที่บิตนั้นๆ เชื่อมต่อกับวงจรพัลส์ภายใน โดยตรงส่งผลให้ขาพอร์ตนั้นๆ มีลอจิก 1 สามารถรับสัญญาณลอจิก 0 จากอุปกรณ์ภายนอกได้ง่าย สัญญาณจากข้อมูลจากอุปกรณ์ภายนอกจะถูกส่งเข้ามาเก็บไว้ในวงจรบัฟเฟอร์ภายในพอร์ต แล้วรอให้ cpu มาอ่านค่าเข้าไป เมื่อเป็นเช่นนี้ อุปกรณ์ภายนอกที่เชื่อมต่อกับพอร์ตอินพุตของไมโครคอนโทรลเลอร์ ควรกำหนดให้สถานะการทำงานที่ลอจิก 0 จะสะดวกที่สุด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4.4 การใช้งานเป็นพอร์ตเอาต์พุต

โดยปกติขาพอร์ตถูกกำหนดเป็นพอร์ตเอาต์พุตอยู่แล้วจึงส่งข้อมูลออกไปได้อย่างง่ายดาย คือเมื่อต้องการส่งข้อมูล 0 ก็เขียนข้อมูล 0 ไปยังวงจรแลตซ์ซึ่งจะส่งค่าไปขับเพดให้ทำงาน หากส่งข้อมูล 1 ไปยังวงจรแลตซ์ซึ่งจะส่งค่าไปขับเพดให้หยุดทำงาน ทำให้ขาพอร์ตเชื่อมต่อกับวงจรพูลอัพภายในโดยตรงส่งผลให้ขาพอร์ตนั้นๆมีลอจิก 1 ซึ่งคล้ายกับการกำหนดให้เป็นขาอินพุต แต่ต่างกันที่กระบวนการย้ายข้อมูล โดยถ้าเป็นขาอินพุตจะมีสัญญาณมาอ่านข้อมูลที่วงจรับัพเฟอร์ ถ้าเป็นเอาต์พุตจะไม่มีสัญญาณมาอ่านข้อมูลที่วงจรับัพเฟอร์แต่อย่างใด

2.4.5 การอ่านค่าลอจิกจากพอร์ต

สามารถอ่านค่าลอจิกจากพอร์ตได้ 2 ลักษณะ คือ อ่านโดยจากพอร์ตโดยตรง และอ่านจากวงจรแลตซ์ของแต่ละพอร์ต ที่สำคัญการอ่านลอจิกจากพอร์ตควรเลือกวิธีการให้เหมาะสมกับอุปกรณ์ที่นำมาต่อด้วย



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.5 รายละเอียดเกี่ยวกับโมดูล LCD

ในโมดูล LCD จะประกอบด้วยส่วนหลักๆ 3 ส่วน ดังนี้

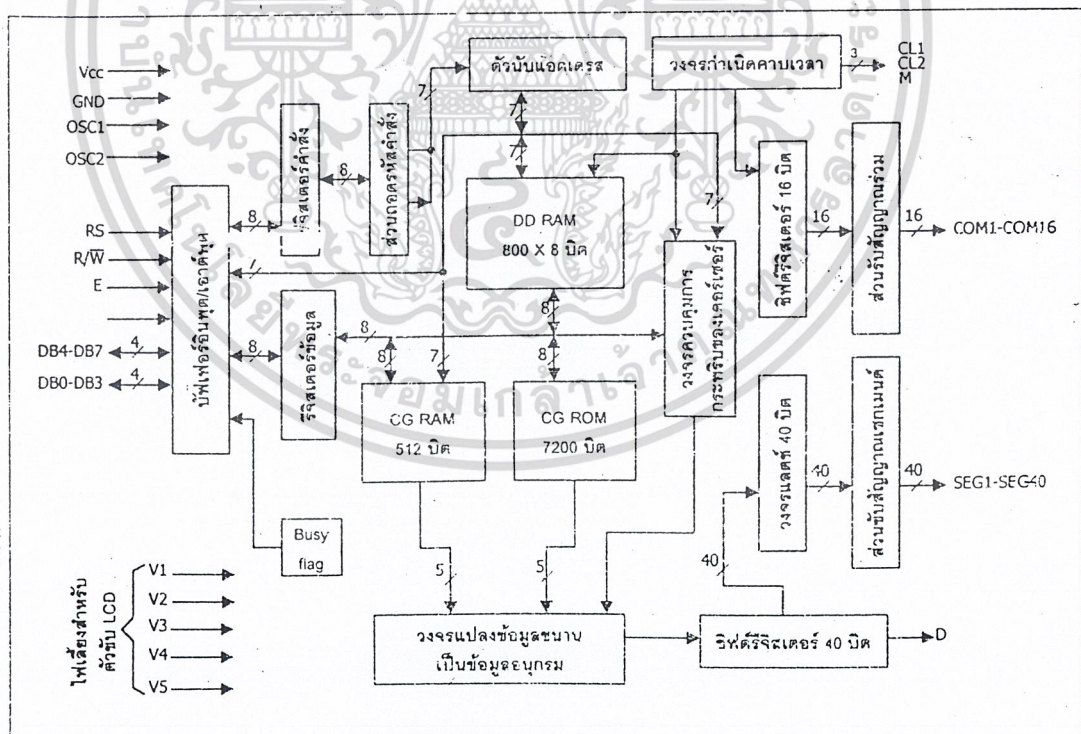
1. ตัวแสดงผล (display) ภายในเป็นผลึกเหลวแสดงผลโดยอาศัยแสงจากภายนอก ดังนั้นจึงต้องมีมุมในการมองข้อมูลที่แสดงบนจอ LCD

2. ตัวควบคุม (controller) เป็นตัวรับข้อมูลจากอุปกรณ์ภายนอกมาควบคุมการทำงานของ LCD เช่น ลบจอภาพ แสดงอักษร เป็นต้น ใช้ชิปควบคุมเฉพาะ คือ เบอร์ 44780 ควบคุม LCD แบบอักษร ส่วน HD61830 ควบคุม LCD แบบกราฟฟิก

3. ตัวขับ (driver) รับสัญญาณจากตัวควบคุมมาขับตัวแสดงผลให้แสดงข้อมูลตามกำหนด ใช้ชิปควบคุม คือ เบอร์ HD 44100H MSM5259

2.5.1 โครงสร้างภายในโมดูล LCD

การใช้งานโมดูล LCD ต้องเข้าใจโครงสร้างและคำสั่งที่ใช้ควบคุมที่ชิปตัวอย่างของโมดูล LCD แบบอักษร โดยในรูปที่ 2.15 เป็นเบอร์ 44780 ประกอบด้วย



เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ที่ 2.15 ใต้อะแกรมการทำงานของโมดูล LCD แบบอักษรที่ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บัฟเฟอร์อินพุตเอาทพุท ใช้ติดต่อรับส่งข้อมูลกับอุปกรณ์ภายนอก เพื่อจะถ่ายทอดข้อมูลเข้าออกภายในตัวควบคุม

รีจิสเตอร์คำสั่ง (Instruction Register : IR) เป็นรีจิสเตอร์ใช้รับข้อมูลคำสั่งจากอุปกรณ์ภายนอก เพื่อไปควบคุมตัวแสดงผล

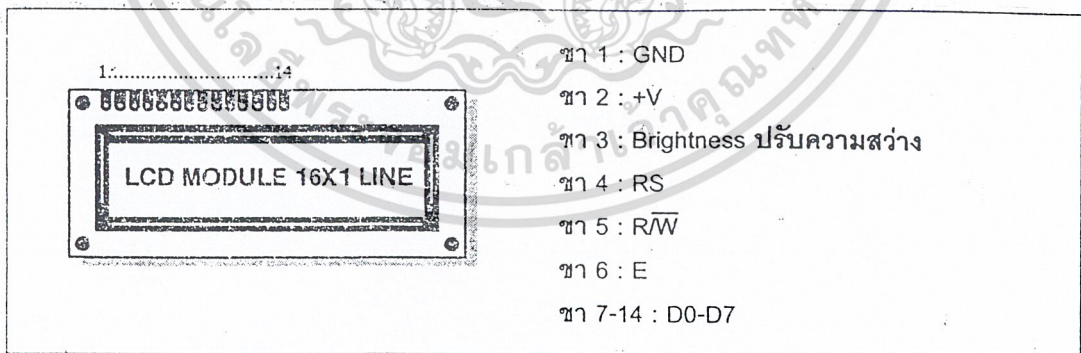
รีจิสเตอร์ข้อมูล (Data Register : DR) เป็นรีจิสเตอร์ใช้รับข้อมูลจากอุปกรณ์ภายนอกเพื่อถ่ายทอดไปยังหน่วยความจำที่ทำหน้าที่เก็บข้อมูลแสดงผล หรือนำข้อมูลไปสร้างตัวอักษรเพิ่มเติมในแรมเก็บตัวอักษร

แรมเก็บข้อมูลการแสดงผล (Display Data Ram : DDRAM) ทำหน้าที่เก็บข้อมูลจากรีจิสเตอร์ DR ตัวควบคุมจะนำข้อมูลใน DDRAM นี้ไปเปิดตาราง (look up-table) ของตัวอักษรที่เก็บในหน่วยความจำรวมและแรมเก็บตัวอักษร เพื่อนำไปแสดงที่ตัวแสดงผล

รอมเก็บตัวอักษร (Character Generator ROM : CGROM) ใช้เก็บข้อมูลตัวอักษร หรือสัญลักษณ์ที่อ่านออกไปแสดงที่ตัวแสดงผล ได้ ถูกอ่านด้วยค่า DDRAM

แรมเก็บตัวอักษร (Character Generator RAM : CGRAM) ใช้เก็บข้อมูลตัวอักษรที่มีการสร้างเพิ่มเติมในกรณีตัวอักษรใน DDRAM ไม่เพียงพอ การอ่านทำเช่นเดียวกับ DDRAM

แฟลค busy แจ้งสถานะการทำงานของตัวควบคุมให้อุปกรณ์ภายนอกทราบว่า ตัวรับข้อมูลพร้อมรับคำสั่งหรือข้อมูลหรือไม่ ดังนั้นก่อนส่งข้อมูลต้องตรวจสอบสถานะแฟลค busy ว่าว่างหรือไม่



รูปที่ 2.16 รูปร่างและการจัดขาของโมดูล LCD แบบอักขระ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.5.2 โมดูล LCD ขนาด 16 ตัวอักษร 1 บรรทัด (LCD 16*1)

ที่ยกโมดูลตัวนี้เพราะถูก หาง่าย มีโครงสร้างที่มีมาตรฐาน มีผู้ผลิตหลายบริษัท ซึ่งมีระบ
เบอร์ที่แตกต่างกัน ที่นี้ใช้เบอร์ HD44750 ของ HITASHI

โมดูล LCD ขนาด 16 ตัวอักษร 1 บรรทัด มีขาทั้งหมด 14ขา มีการจัดขาตามรูปที่ 2.16
สำหรับรายละเอียดของแต่ละขามีดังนี้

- Vss (ขา1) : ต่อกราวด์
- Vdd(ขา2) : ต่อไฟเลี้ยง +5 โวลต์
- Vo(ขา3) : เป็นขาอินพุต รับแรงดันเพื่อปรับความเข้มการแสดงผล
- RS(ขา4) : เป็นขาอินพุตใช้แยกข้อมูลที่ใช้ในการประมวลผลในขณะนั้นว่าเป็นคำ
สั่งของรีจิสเตอร์ IR หรือ DR ถ้าขาเป็น 0 ข้อมูลที่ส่งมาจะเป็นคำสั่ง
แต่ถ้าขาเป็น 1 จะเป็นข้อมูลสำหรับการแสดงผล
- R/W(ขา5) : เป็นขาที่ใช้เลือกการอ่านหรือเขียนข้อมูลกับ LCD ถ้าเป็น 0 เป็นการ
กำหนดให้เขียนข้อมูล แต่ถ้าเป็น 1 จะเป็นการอ่านข้อมูล
- E(ขา6) : เป็นขา enable LCD ให้ทำงาน
- D0-D7(ขา7-14) : ใช้เป็นทางผ่านระหว่าง LCD กับอุปกรณ์ภายนอกขนาด 8 บิต

2.5.3 คำสั่งควบคุม LCD โมดูล

การเขียนคำสั่งลงในตัวควบคุมต้องกำหนดขา RS และ R/W เป็น 0 แล้วเขียนคำสั่งตามไป คำ
สั่งที่สำคัญมี 10 คำสั่ง ดังนี้

1.คำสั่ง Clear Display

มีข้อมูลคำสั่งเป็น 01H เป็นคำสั่งที่ใช้เขียนข้อมูลช่องว่าง(Space) เข้าไปในDDRAM ทั้งหมด เมื่อตัว
ควบคุมเอ็กซิกิวส์คำสั่งนี้ จะกำหนดแอสแตรคของ DDRAM เป็น 0 เคอร์เซอร์เลื่อนมาที่ซ้ายสุดของ
จอภาพ แล้วเซตบิต I/D ให้เป็น 1

2.คำสั่ง Return Home

ต้องกำหนดให้บิต 1 เป็น “1” เป็นคำสั่งเลื่อนให้เคอร์เซอร์ไปด้านซ้ายสุดของจอแสดงผล แต่ข้อมูล
บนจอแสดงผลไม่เปลี่ยนแปลง นั่นคือ ข้อมูลของคำสั่งนี้จะเป็น 02H หรือ 03H ก็ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. คำสั่ง Enter Mode Set

มีรายละเอียดดังนี้

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

บิต S เป็นบิตกำหนดลักษณะการแสดงผล เมื่อมีการป้อนข้อมูล หากบิต S เป็น 1 เมื่อเกิดข้อมูลใหม่บนจอแสดงผล ตัวเคอร์เซอร์ จะอยู่กับที่ แต่ตัวอักษรข้อมูลเดิมจะถูกดันไปทางซ้ายมือ หากบิต S เป็น 0 เมื่อเกิดข้อมูลใหม่บนจอแสดงผล ตัวเคอร์เซอร์จะถูกดันไปทางขวามือ
บิต I/D ใช้ในการกำหนดว่า เมื่อเขียนหรืออ่านข้อมูลแล้ว ทำให้แอดเดรสของ DDRAM เพิ่มขึ้นหรือลดลงไป 1 แอดเดรส โดยถ้าบิตนี้เป็น 1 แอดเดรสของ DDRAM จะเพิ่มขึ้น แต่ถ้าเป็น 0 แอดเดรสของ DDRAM จะลดลง

4. คำสั่งควบคุมการแสดงผล

มีรายละเอียดดังนี้

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

บิต D ใช้ควบคุมการเปิดปิดจอแสดงผล ถ้าบิตนี้เป็น 1 เป็นการเปิดจอแสดงผล ถ้าบิตนี้เป็น 0 เป็นการปิดจอแสดงผล

บิต C ใช้ควบคุมตัวเคอร์เซอร์บนจอแสดงผล ถ้าบิตนี้เป็น 1 จะมีเคอร์เซอร์แสดงผลบนจอแสดงผล ถ้าบิตนี้เป็น 0 จะไม่มีเคอร์เซอร์แสดงผลบนจอแสดงผล หรือไม่มีการแสดงเคอร์เซอร์

บิต B ควบคุมการกระพริบของตัวเคอร์เซอร์ ถ้าบิตนี้เป็น 1 เคอร์เซอร์จะกระพริบ ถ้าบิตนี้เป็น 0 เคอร์เซอร์จะไม่กระพริบ

5. คำสั่งควบคุมการเลื่อนเคอร์เซอร์และข้อมูลตัวอักษร

มีรายละเอียดดังนี้

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

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านกรค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การควบคุมการเลื่อนเคอร์เซอร์และตัวอักษรบนจอแสดงผลขึ้นอยู่กับกำหนดบิต S/C และ R/L สามารถสรุปได้ดังนี้

S/C	R/L	ลักษณะการเลื่อน	ข้อมูลคำสั่ง
0	0	เลื่อนเคอร์เซอร์ไปทางซ้าย	10H-13H
1	1	เลื่อนเคอร์เซอร์ไปทางขวา	14H-17H
1	0	เลื่อนตัวอักษรใหม่ไปทางซ้าย	18H-1BH
1	1	เลื่อนตัวอักษรใหม่ไปทางขวา	1CH-1FH

6. คำสั่งกำหนดฟังก์ชันการทำงาน

มีรายละเอียดดังนี้

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

บิต DL ใช้กำหนดจำนวนบิตที่ใช้ติดต่อส่งผ่านข้อมูล ถ้าบิตนี้เป็น 0 จะเป็นการติดต่อแบบ 4 บิต แต่ถ้าบิตนี้เป็น 1 จะเป็นการติดต่อแบบ 8 บิต

บิต N ใช้กำหนดจำนวนบรรทัดของการแสดงผล ถ้าบิตนี้เป็น 0 จะแสดงผล 1 บรรทัด ถ้าบิตนี้เป็น 1 จะแสดงผล 2 บรรทัด กรณีต้องการให้แสดงผลมากกว่า 2 บรรทัด และสามารถที่จะแสดงผลมากกว่า 2 บรรทัด ต้องกำหนดบิต N ให้เป็น 1

บิต F เลือกลักษณะของตัวอักษร ถ้าบิตนี้เป็น 0 จะแสดงผลแบบ 5*7 จุด ถ้าบิตนี้เป็น 1 จะแสดงผลแบบ 5*10 จุด

จุดนำสังเกต คือ โมดูล LCD ขนาด 16 ตัวอักษร 1 บรรทัด แม้จะมีบรรทัดการแสดงผลแค่ 1 บรรทัด แต่จะกำหนด N ให้เป็น 1 เนื่องจากแอดเดรสของ DDRAM แบ่งเป็น 2 ช่อง คือ 00H และ 40H

7. คำสั่งเลือกแอดเดรสของ CGRAM

เมื่อต้องการกำหนดแอดเดรสของ CGRAM ต้องกำหนดให้บิต 7 เป็น 0 บิต 6 เป็น 1 ส่วนอีก 6 บิตที่เหลือจะแทนด้วยค่าแอดเดรสของ CGRAM จะต้องกำหนดแอดเดรสด้วยคำสั่งนี้ก่อนที่จะอ่านหรือเขียนข้อมูลให้ CGRAM ซึ่งแอดเดรสของ CGRAM จะอยู่ระหว่าง 00H-3FH

8. คำสั่งเลือกแอดเดรสของDDRAM

ใช้ในการเลือกแอดเดรสของDDRAMก่อนที่จะอ่านหรือเขียนข้อมูล ต้องกำหนดให้บิต 7 เป็น 1 ส่วนอีก 7 บิตที่เหลือจะเป็นค่าแอดเดรสของ DDRAM ซึ่งแอดเดรสของ CGRAM จะอยู่ระหว่าง 8CH-FFH ทั้งนี้จำนวนแอดเดรสขึ้นอยู่กับข้อกำหนดสถานะที่บิต N หากแทนบิต N เป็น 0 ซึ่งแอดเดรสของ CGRAM จะอยู่ระหว่าง 80H-0CFH บิต N หากแทนบิต N เป็น 1 ซึ่งแอดเดรสของ CGRAM จะมีอยู่ 2 ช่วง คือ อยู่ระหว่าง 80H-0CFH และ 0C0H-0C7H

9. คำสั่งการแฟลค Busy และแอดเดรส

มีรายละเอียดดังนี้

บิต7	บิต6	บิต5	บิต4	บิต3	บิต2	บิต1	บิต0
BF	A	A	A	A	A	A	A

เป็นคำสั่งที่ใช้อ่านแฟลค Busy (BF) โดยแฟลคนี้จะเป็นตัวบอกสถานะของตัวควบคุม LED ว่าพร้อมจะรับข้อมูลหรือไม่ ถ้าบิต BF เป็น 0 แสดงว่าตัวควบคุม LED พร้อมจะรับข้อมูลหรือคำสั่ง แต่ถ้าบิต BF เป็น 1 แสดงว่าขณะนี้ตัวควบคุม LED ยังอยู่ในกระบวนการทำงานภายในหรือกำลังประมวลผลข้อมูลอยู่ ยังไม่พร้อมจะรับข้อมูลหรือคำสั่ง

เมื่อต้องการอ่านแฟลคต้องกำหนดให้ขา R/W เป็น 1 ด้วย แต่สัญญาณที่ RS ยังคงเป็น 0 อยู่ เพราะข้อมูลนี้เป็นข้อมูลคำสั่ง

นอกจากนี้ยังใช้เป็นข้อมูลคำสั่งอ่านข้อมูลแอดเดรสของ CGRAM และ DDRAM โดยบิต 0-6 เป็นค่าของข้อมูลของแอดเดรสที่ต้องการอ่าน

2.5.4 การเขียนคำสั่งและข้อมูลให้แก่โมดูล LCD

ในการเขียนข้อมูลเพื่อควบคุมให้แก่โมดูล LCD แสดงผลตามที่ผู้ใช้งานต้องการต้องส่งคำสั่ง (instruction) แล้วกำหนดโหมดของการทำงานให้แก่โมดูล LCD ก่อน จากนั้นค่อยส่งข้อมูล (data) ที่ต้องการแสดงผล เนื่องจากข้อมูลของโมดูล LCD มี 8 เส้น คือ D0-D7 และใช้เป็นเส้นทางผ่านของทั้งคำสั่งและข้อมูล ดังนั้นในการส่งคำสั่งและข้อมูลจึงต้องอาศัยการกำหนดสัญญาณลอจิกที่ขา RS ถ้าหากที่ขา RS ได้รับลอจิก 0 หมายความว่า ข้อมูลที่ป้อนให้แก่โมดูล LCD ขณะนั้นเป็นคำสั่ง แต่ถ้าหากที่ขา RS ได้รับลอจิก 1 หมายความว่า ข้อมูลที่ป้อนให้แก่โมดูล LCD ขณะนั้นเป็นข้อมูลที่ใช้ในการ

เอกสารนี้เป็นเอกสารลิขสิทธิ์ของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ ห้ามเผยแพร่โดยไม่ได้รับอนุญาต
ไม่ว่าโดยทางใดก็ตาม หากมีให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อต้องการเขียนหรืออ่านข้อมูลใน CGRAM และ DDRAM เริ่มต้นต้องกำหนดแอดเดรสที่ต้องการอ่านหรือเขียนก่อน โดยใช้คำสั่งเลือกแอดเดรส จากนั้นกำหนดให้ขา RS เป็น 1 เพื่อแจ้งให้ตัวควบคุมภายใน โมดูล LCD ทราบว่าข้อมูลที่ปรากฏต่อไปนี้เป็นข้อมูลปกติไม่ใช่คำสั่ง

ในกรณีที่ต้องการอ่านข้อมูลต้องกำหนดให้ขา R/W เป็น 1 ข้อมูลขนาด 8 บิต (หรือ 4 บิต) ก็จะปรากฏบนบัสข้อมูล โดยข้อมูลที่อ่านมาได้จะเป็นข้อมูลจากแอดเดรส CGRAM หรือ DDRAM ตามที่ต้องการ

ในกรณีที่ต้องการเขียนข้อมูล เมื่อกำหนดแอดเดรสและป้อนลอจิก 1 ให้ขา RS แล้วแล้วต้องกำหนดให้ขา R/W เป็น 0 ข้อมูลที่อยู่บนบัสข้อมูลจะถูกเขียนลงในรีจิสเตอร์ DR จากนั้นค่อยถ่ายทอดไปยัง DDRAM ต่อไป

2.5.5 จังหวะการทำงานของโมดูล LCD

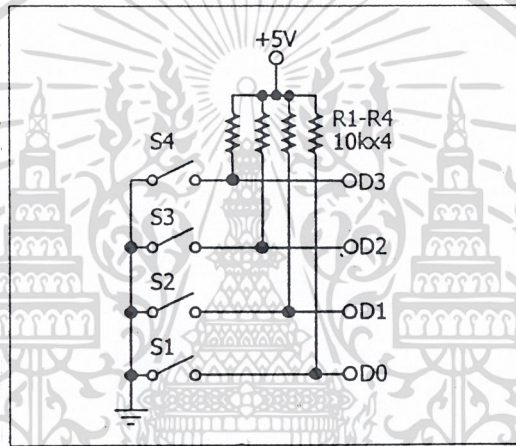
ในการติดต่อกับโมดูล LCD จะต้องมีการหน่วงเวลาหลังจากที่ทำการส่งรหัสคำสั่งหรือข้อมูล เนื่องจากต้องรอให้ไมโครคอนโทรลเลอร์ภายในโมดูล LCD แปลความหมายของรหัสคำสั่งและทำงานตามคำสั่งให้เรียบร้อยก่อน จากนั้นค่อยรับข้อมูลหรือดำเนินการต่อไป

ดังนั้นการใช้งาน โมดูล LCD ผู้เขียนโปรแกรมต้องมีโปรแกรมเพื่อหน่วงเวลาทำให้โมดูล LCD พร้อมทำงานด้วย โดยเริ่มจ่ายไฟให้แก่โมดูล LCD ต้องรอประมาณ 10 ms เพื่อให้โมดูล LCD ทำการเตรียมความพร้อมหรืออินทิเชี่ยล (initial) หลังจากนั้นก็กำหนดลอจิกให้แก่ขา RS ของโมดูล LCD แล้วต้องหน่วงเวลาอีกประมาณ 2 ms เพื่อให้ไมโครคอนโทรลเลอร์ในโมดูล LCD แปลความหมายของลอจิกที่ขา RS ว่าข้อมูลต่อไปนี้จะได้รับเป็นรหัสคำสั่งหรือข้อมูลที่ต้องแสดงผล จากนั้นจะเป็นการส่งข้อมูลมารอบที่บัสข้อมูล D0-D7 (ในกรณีข้อมูล 8 บิต) ขั้นตอนต่อไปนี้จะเป็นการส่งสัญญาณพัลส์ ไปที่ขา E เพื่อ enable โมดูล LCD ให้รับข้อมูลจากบัสข้อมูลเข้าไป โดยสัญญาณพัลส์ ที่ป้อนเข้าขา E ของโมดูล LCD ต้องเป็นพัลส์ขอบขาขึ้น จากนั้นหน่วงเวลาอีกประมาณ 2 ms

ทั้งหมดที่กล่าวมาคือขั้นตอนและจังหวะในการทำงาน 1 รอบของโมดูล LCD จะเห็นได้ว่าจะมีโปรแกรมน้อยที่ล้าคัญอยู่ 3 โปรแกรมย่อย คือ โปรแกรมอินทิเชี่ยล LCD โปรแกรมหน่วงเวลา โปรแกรมการส่งพัลส์เพื่อ enable โมดูล LCD

2.6 การเชื่อมต่อกับคีย์แพดหรือสวิตช์เมตริก 4×4จุด

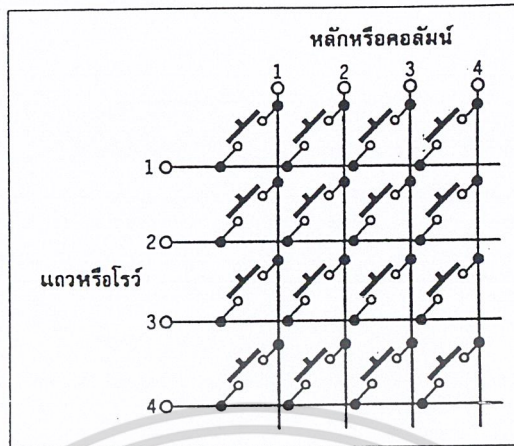
การอ่านค่าหรือรับค่าการกดสวิตช์เป็นอีกงานหนึ่งที่ไม่โครคอนโทรลเลอร์ต้องสามารถรองรับและเชื่อมต่อใช้งานร่วมด้วยวงจรของสวิตช์มีด้วยกัน 2 ลักษณะใหญ่ๆคือต่อเข้ากับไฟเลี้ยงหรือกราวด์โดยตรงเมื่อสวิตช์ตัวใดต่อวงจรสามารถอ่านค่าได้โดยตรง ดังในรูปที่ 2.17 วงจรในลักษณะนี้ไม่มีความซับซ้อน สามารถอ่านค่าของสวิตช์ได้ง่ายและรวดเร็ว แต่มีข้อเสียคือถ้าหากจำนวนของสวิตช์มีมากๆจำนวนของสายข้อมูลจะมีมากตาม ทำให้ระบบหรือวงจรโดยรวมมีขนาดใหญ่และสิ้นเปลือง



รูปที่ 2.17 วงจรของสวิตช์แบบต่อเข้ากับไฟเลี้ยงและกราวด์

วงจรของสวิตช์อีกลักษณะหนึ่งคือ การต่อวงจรแบบเมตริกซ์ (matrix switch) ดังในรูปที่ 2.18 สวิตช์จะถูกต่อในแนวแกนตั้งและแนวนอน จะเรียกแนวตั้งว่า หลักหรือคอลัมน์ (column) ในขณะที่แนวนอนจะเรียกว่า แถวหรือโรว์ (row) ดังนั้นค่าของสวิตช์จะต้องประกอบด้วย ตำแหน่งในแนวหลักและแถว กระบวนการที่จะทำได้มาซึ่งค่าของสวิตช์มีขั้นตอนซับซ้อนพอสมควร แต่วงจรของสวิตช์แบบนี้มีข้อดีคือสามารถรองรับการเพิ่มของสวิตช์ได้อย่างสะดวก เพียงเพิ่มเติมจำนวนสวิตช์และแก้ไขซอฟต์แวร์อีกเล็กน้อยเท่านั้น ทำให้วงจรสวิตช์แบบเมตริกเป็นที่นิยมใช้มากในระบบควบคุมอัตโนมัติหรือกึ่งอัตโนมัติที่มีจำนวนสวิตช์มากกว่า 8 ตัว ในการใช้งานทั่วไปจะเรียกสวิตช์แบบเมตริกซ์นี้ว่า คีย์แพด(keypad)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

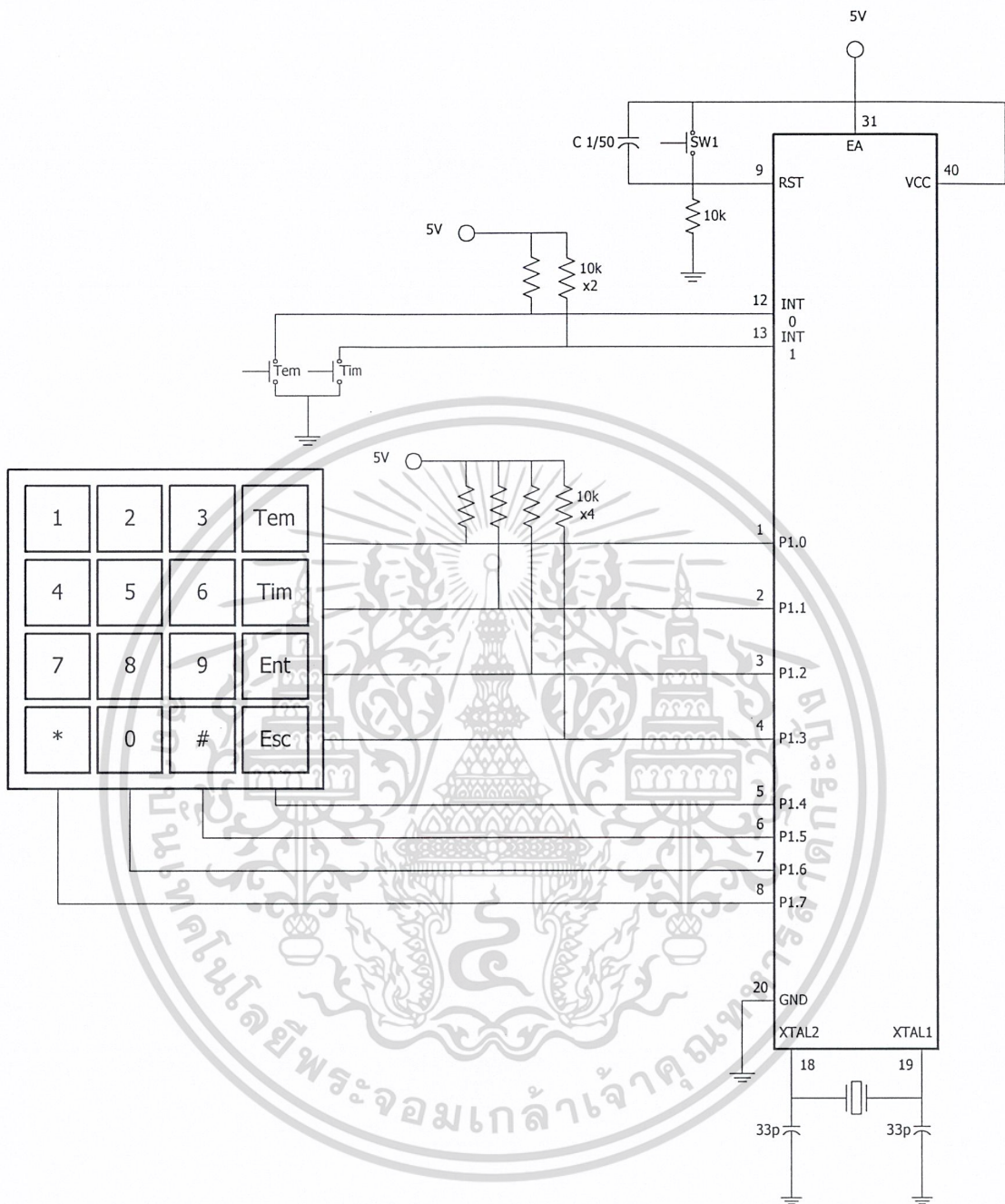


รูปที่ 2.18 วงจรของสวิตช์แบบเมตริกซ์หรือคีย์แพด

2.6.1 การเชื่อมต่อคีย์แพดเข้ากับไมโครคอนโทรลเลอร์ MCS-51

ตัวอย่างวงจรดังรูปที่ 2.19 จะใช้พอร์ต 2 ของไมโครคอนโทรลเลอร์เชื่อมต่อเข้ากับคีย์แพดทั้ง 8 เส้นคือสายของคอลัมน์ 4 สาย C0-C3 และสายทางโรวอีก 4 สายคือ R0-R3 โดยเฉพาะที่ขาพอร์ต P2.0-P2.3 จะต้องต่อตัวต้านทานพูลอัปไว้เพื่อกำหนดสถานะเริ่มต้นที่ไม่มีกรกดคีย์ โดยไมโครคอนโทรลเลอร์จะทำการส่งข้อมูล 0 ไปยัง P2.6, P2.5 และ P2.4 ตามลำดับ ในทุกครั้งที่มีการส่งข้อมูลไปยังสายคอลัมน์ของคีย์แพด ไมโครคอนโทรลเลอร์จะทำการอ่านค่าที่ P2.0-P2.3 เข้ามาด้วย หากไม่มีการกดค่าของ P2.0-P2.3 ก็จะเป็น 1 ทั้งหมด ถ้าหากมีการกดคีย์ ค่าของ P2.0-P2.3 ก็จะไม่เป็น 1111 อีกต่อไป เป็นการแจ้งให้ทราบว่า มีการกดคีย์แพดขึ้นแล้ว จากนั้นไมโครคอนโทรลเลอร์ก็จะทำการค้นหาตำแหน่งต่อไป โดยการค้นหาตำแหน่งสิ่งที่จะได้มาอย่างแรกคือ ค่าตำแหน่งของคีย์นั้น จากนั้นก็จะนำตำแหน่งนั้น ไปเปิดตารางข้อมูล เพื่อจะได้หมายเลขของคีย์ที่กดอย่างแท้จริง ยกตัวอย่างจากวงจรในรูปที่ 2.18 หากคีย์ 0 ถูกกด ไมโครคอนโทรลเลอร์จะได้ค่าตำแหน่งของคีย์ 0 เป็น OBH จากนั้นนำค่า OBH นี้ไปเปิดตารางก็จะได้ข้อมูลเป็น 0 หมายถึงคีย์ 0 ซึ่งกระบวนการหลังนี้จะป็นหน้าที่ของซอฟต์แวร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.19 การเชื่อมต่อคีย์แพดเข้ากับไมโครคอนโทรลเลอร์ MCS-51

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.7 ไทเมอร์/เคาน์เตอร์

ไทเมอร์/เคาน์เตอร์ (Timer/Counter) เป็นอีกหนึ่งส่วนประกอบที่สำคัญของไมโครคอนโทรลเลอร์ เนื่องจากในการทำงานของไมโครคอนโทรลเลอร์จะต้องมีการเก็บและตรวจสอบค่าของเวลาและจำนวนของสัญญาณนาฬิกาอยู่ตลอดเวลาทั้งนี้เพื่อประโยชน์ในการสร้างฐานเวลา สร้างสัญญาณพัลส์ เปรียบเทียบค่าเวลาหรือเปรียบเทียบค่าของการนับ รวมไปถึงการกำหนดอัตราเร็วในการสื่อสารข้อมูลของพอร์ตอนุกรมด้วย

ในไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลชเบอร์ AT89C51 มีวงจรไทเมอร์/เคาน์เตอร์ขนาด 16 บิต 2 ตัว โดยค่าของวงจรไทเมอร์/เคาน์เตอร์นี้จะเก็บไว้ในรีจิสเตอร์ขนาด 16 บิตที่ชื่อ ไทเมอร์ 0 (Timer 0) และ ไทเมอร์ 1 (Timer 1) เรียกสั้นๆ ว่า T0 และ T1 ในไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลชเบอร์ AT89C52 และอนุกรม AT89Sxx จะมีไทเมอร์/เคาน์เตอร์ถึง 3 ตัวคือ มี ไทเมอร์ 2 (Timer 2 : T2) เพิ่มเติมเข้ามา โดยรีจิสเตอร์ไทเมอร์/เคาน์เตอร์ทั้งสามตัวสามารถกำหนดให้ทำงานเป็นตัวตั้งเวลาหรือไทเมอร์ และตัวนับหรือเคาน์เตอร์ได้อย่างอิสระต่อกัน การนับค่าของ ไทเมอร์/เคาน์เตอร์ในไมโครคอนโทรลเลอร์ MCS-51 จะเป็นการนับขึ้นหรือเพิ่มค่าเพียงทางเดียว

2.7.1 การทำงานในโหมดไทเมอร์

เมื่อกำหนดให้ทำงานเป็นตัวตั้งเวลาหรือไทเมอร์ ค่าของรีจิสเตอร์จะเพิ่มขึ้นในทุกๆเมซินไซเคิล ดังนั้นเมื่อทำงานเป็นไทเมอร์ รีจิสเตอร์จะทำการนับค่าของเมซินไซเคิลนั่นเอง และเนื่องจากเมซินไซเคิลประกอบด้วยคาบเวลาของวงจรกำเนิดสัญญาณนาฬิกา 12 คาบเวลา ดังนั้นอัตราในการนับของรีจิสเตอร์จึงเท่ากับ $1/12$ ของความถี่สัญญาณนาฬิกา

2.7.2 การทำงานในโหมดเคาน์เตอร์

เมื่อการทำงานเป็นตัวนับหรือเคาน์เตอร์ ค่าของรีจิสเตอร์จะเพิ่มขึ้นก็ต่อเมื่อมีการเปลี่ยนแปลงของระดับลอจิกจาก “1” เป็น “0” เกิดขึ้นที่ขาอินพุตทางฮาร์ดแวร์ของวงจรไทเมอร์/เคาน์เตอร์ ซึ่งก็คือขา T0 (P1.0) ในไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลชเบอร์ AT89C52 และอนุกรม AT89Sxx โดยจะมีการสุ่มรับสัญญาณจากขาอินพุตในทุกๆ คาบเวลาที่ 2 ของสเตตที่ 5 (S5P2) ในแต่ละเมซินไซเคิล

เมื่อสัญญาณอินพุตเปลี่ยนแปลงจาก “1” เป็น “0” เป็นเวลาหนึ่งไซเคิล ในไซเคิลต่อมาค่าของการนับจะเพิ่มขึ้นหนึ่งค่า และจะไปปรากฏในรีจิสเตอร์ภายในคาบเวลาที่ 1 ของสเตตที่ 3 (S3P1) ของเมซินไซเคิลต่อไป หลังจากที่ตรวจจับพบการเปลี่ยนแปลงที่ขาไทเมอร์อินพุตแล้ว เมื่อเป็นเช่นนี้ในกระบวนการตรวจจับการเปลี่ยนแปลงของสัญญาณอินพุตที่ขาไทเมอร์จะต้องใช้ 2 เมซินไซเคิลไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อัตราการบินของเคาน์เตอร์จึงเท่ากับ $1/24$ ของความถี่สัญญาณพิก้า ดังนั้นความถี่สูงสุดของสัญญาณอินพุตที่ไทเมอร์/เคาน์เตอร์ภายในไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลชสามารถตรวจจับได้จึงเท่ากับความถี่ของสัญญาณพิก้าหารด้วย 24 ยกตัวอย่าง ในไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลชเบอร์ AT89C51 สามารถใช้สัญญาณพิก้าได้สูงสุด 24 MHz ดังนั้นความถี่สูงสุดของสัญญาณอินพุตที่ไทเมอร์/เคาน์เตอร์สามารถตรวจจับได้คือ 1 MHz

2.7.3 รีจิสเตอร์ที่เกี่ยวข้องกับการทำงานของไทเมอร์/เคาน์เตอร์ 0 และ 1

ในการทำงานของไทเมอร์/เคาน์เตอร์ 0 และ 1 ในไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลช มีรีจิสเตอร์ที่ต้องเกี่ยวข้องเป็นพื้นฐานอยู่ 6 ตัว ดังมีรายละเอียดต่อไปนี้

1. รีจิสเตอร์ไทเมอร์

มีด้วยกัน 4 ตัวคือ TL0 มีแอดเดรสอยู่ที่ 8AH, TH0 มีแอดเดรสอยู่ที่ 8CH, TL1 มีแอดเดรสอยู่ที่ 8BH และ TH1 มีแอดเดรสอยู่ที่ 8DH รีจิสเตอร์ทั้ง 4 ตัวจะอยู่ในพื้นที่ของรีจิสเตอร์ฟังก์ชันพิเศษคือ SFR รีจิสเตอร์แต่ละตัวมีขนาด 8 บิต แต่ในการใช้งานโดยทั่วไปมักใช้ร่วมกันโดยจัดเป็นคู่คือ TL0 กับ TH0 รวมกันเป็นรีจิสเตอร์ไทเมอร์ 0 ขนาด 16 บิต และ TL1 กับ TH1 รวมกันเป็นรีจิสเตอร์ไทเมอร์ 1 ขนาด 16 บิต โดยใน TL0 กับ TL1 เก็บข้อมูล 8 บิตล่าง ส่วน TH0 กับ TH1 เก็บข้อมูล 8 บิตบน

2. รีจิสเตอร์ควบคุมการทำงานของไทเมอร์/เคาน์เตอร์หรือ TCON (Timer/Counter Control Register)

เป็นรีจิสเตอร์ขนาด 8 บิต มีแอดเดรสอยู่ที่ 88H ในพื้นที่ของรีจิสเตอร์ฟังก์ชันพิเศษ SFR สามารถเข้าถึงได้ในระดับบิตมีรายละเอียดการทำงานดังนี้

บิต 7	บิต 6	บิต 5	บิต 4	บิต 3	บิต 2	บิต 1	บิต 0
TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

TF1(Timer 1 overflow flag): เซ็ตด้วยกระบวนการทางฮาร์ดแวร์ เมื่อค่าของรีจิสเตอร์ ไทเมอร์ 1 เกิดการนับเกินหรือเกิดโอเวอร์โฟลว การเคลียร์บิตนี้ทำได้ด้วยเช่นกัน โดยบิตนี้จะเคลียร์เมื่อมีการอินเทอร์รัปต์เกิดขึ้น

TR1(Timer1 runcontrol bit) : ใช้ในการเปิดปิดการทำงานของไทเมอร์ 1 ทำการเซตและเคลียร์ด้วยกระบวนการทางซอฟต์แวร์ ถ้าต้องการให้ไทเมอร์ 1 ทำงานต้องเซตให้บิตนี้เป็น 1

TF0(Timer 0 overflow flag): เซ็ตด้วยกระบวนการทางฮาร์ดแวร์ เมื่อค่าของรีจิสเตอร์ ไทเมอร์ 0 เกิดการนับเกินหรือเกิดโอเวอร์โฟลว การเคลียร์บิตนี้ทำได้ด้วยเช่นกัน โดยบิตนี้จะเคลียร์เมื่อมีการอินเทอร์รัปต์เกิดขึ้น

TR0(Timer0 runcontrol bit) : ใช้ในการเปิดปิดการทำงานของไทเมอร์ 0 ทำการเซตและเคลียร์ด้วยกระบวนการทางซอฟต์แวร์ ถ้าต้องการให้ไทเมอร์ 0 ทำงานต้องเซตให้บิตนี้เป็น 1

IE1(External Interrupt 1 edge flag): ใช้ในกระบวนการอินเทอร์รัปต์ สามารถเซตได้ด้วยกระบวนการทางฮาร์ดแวร์ เมื่อสามารถตรวจจับขอบขาของสัญญาณอินเทอร์รัปต์จากภายนอกที่ขาอินพุตอินเทอร์รัปต์ 1 (INT1) ได้ และจะทำการเคลียร์เมื่อมีการอินเทอร์รัปต์เกิดขึ้น

IT1(Interrupt 1 type control 1): ใช้ในกระบวนการอินเทอร์รัปต์ โดยใช้ในการเลือกลักษณะของสัญญาณอินเทอร์รัปต์จากภายนอกที่ต้องการให้ทำการตอบสนองสำหรับขาอินพุตอินเทอร์รัปต์ 1 (INT1) ทำการเซตและเคลียร์ด้วยกระบวนการทางซอฟต์แวร์

“0” เลือกขอบขาลงของสัญญาณ(falling edge)

“1” เลือกระดับลอจิกต่ำ(low level triggered)

IE0(External Interrupt 0 edge flag): ใช้ในกระบวนการอินเทอร์รัปต์ สามารถเซตได้ด้วยกระบวนการทางฮาร์ดแวร์ เมื่อสามารถตรวจจับขอบขาของสัญญาณอินเทอร์รัปต์จากภายนอกที่ขาอินพุตอินเทอร์รัปต์ 0 (INT0) ได้ และจะทำการเคลียร์เมื่อมีการอินเทอร์รัปต์เกิดขึ้น

IT0(Interrupt 1 type control 0): ใช้ในกระบวนการอินเทอร์รัปต์ โดยใช้ในการเลือกลักษณะของสัญญาณอินเทอร์รัปต์จากภายนอกที่ต้องการให้ทำการตอบสนองสำหรับขาอินพุตอินเทอร์รัปต์ 0 (INT0) ทำการเซตและเคลียร์ด้วยกระบวนการทางซอฟต์แวร์

“0” เลือกขอบขาลงของสัญญาณ(falling edge)

“1” เลือกระดับลอจิกต่ำ(low level triggered)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. รีจิสเตอร์เลือกโหมดการทำงานของไทเมอร์/เคาน์เตอร์หรือ

TMOD (Timer/Counter Mode Control Register)

เป็นรีจิสเตอร์ขนาด 8 บิต มีแอดเดรสอยู่ที่ 89H ในพื้นที่ของรีจิสเตอร์ฟังก์ชันพิเศษ SFR ไม่สามารถเข้าถึงได้ในระดับบิตแบ่งการทำงานออกเป็น 2 ส่วนคือ 4 บิตล่างใช้ในการเลือกโหมดการทำงานของไทเมอร์ 0 และ 4 บิตบนใช้ในการเลือกโหมดการทำงานของไทเมอร์ 1

บิต 7	บิต 6	บิต 5	บิต 4	บิต 3	บิต 2	บิต 1	บิต 0
GATE	C/T	M1	M0	GATE	C/T	M1	M0
ไทเมอร์ 1				ไทเมอร์ 0			

GATE: ใช้เลือกลักษณะการควบคุมการทำงานของไทเมอร์/เคาน์เตอร์

“0” ไทเมอร์/เคาน์เตอร์จะทำงานเมื่อบิต TRx ในรีจิสเตอร์ TCON เป็น “1” เรียกการควบคุมแบบนี้ว่าการควบคุมทางซอฟต์แวร์

“1” ไทเมอร์/เคาน์เตอร์จะทำงานเมื่อบิต TRx ในรีจิสเตอร์ TCON เป็น “1” และสถานะลอจิกที่ขาอินพุตอินเตอร์รัปต์ INTO และ INT1 เป็น “1” เรียกการควบคุมแบบนี้ว่าการควบคุมทางฮาร์ดแวร์

C/T (Timer or Counter selector): ใช้เลือกลักษณะการทำงานของไทเมอร์/เคาน์เตอร์

“0” เลือกให้ทำงานเป็นไทเมอร์โดยใช้สัญญาณอินพุตจากสัญญาณนาฬิกาภายในไมโครคอนโทรลเลอร์

“1” เลือกให้ทำงานเป็นเคาน์เตอร์โดยรับสัญญาณอินพุตทางขา T0 หรือ T1

M1,M0 (Mode selector bit) : ใช้เลือกโหมดการทำงานของไทเมอร์/เคาน์เตอร์

“00” เลือกให้ทำงานในโหมดไทเมอร์/เคาน์เตอร์ 13 บิต

“01” เลือกให้ทำงานในโหมดไทเมอร์/เคาน์เตอร์ 16 บิต

“10” เลือกให้ทำงานในโหมดไทเมอร์/เคาน์เตอร์ ขนาด 8 บิตแบบตั้งค่าอัตโนมัติ

“11” สำหรับ ไทเมอร์ 0 เลือกให้ทำงานในโหมดไทเมอร์/เคาน์เตอร์ แยกส่วนโดยแยกออก

เป็น ไทเมอร์/เคาน์เตอร์ 8 บิต 2 ตัว

เอกสารนี้เป็นทรัพย์สินทางปัญญาของสถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

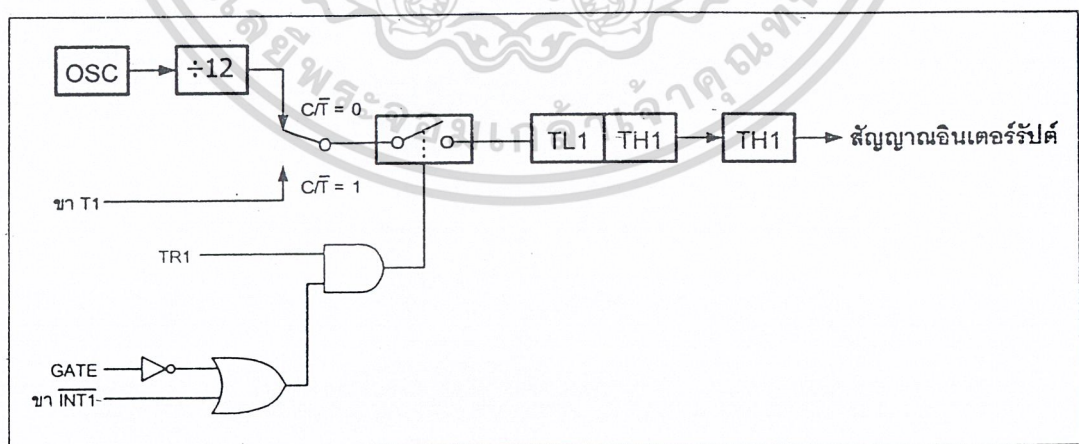
2.7.4 โหมดการทำงานของไทเมอร์/เคาน์เตอร์ 0 และ 1

การเลือกโหมดการทำงานของไทเมอร์/เคาน์เตอร์ 0 และ 1 สามารถทำได้ที่รีจิสเตอร์ TCON และ TMOD ร่วมกัน โดย TCON ใช้ในการเอ็นเอเบิลหรือดิสเอเบิล ไทเมอร์/เคาน์เตอร์ ส่วน TMOD ใช้ในการเลือกโหมดและลักษณะการทำงาน

1. การทำงานในโหมด 0 : ไทเมอร์/เคาน์เตอร์ 13 บิต

มีไดอะแกรมการทำงานแสดงดังรูปที่ 2.20 ในที่นี้จะใช้ ไทเมอร์/เคาน์เตอร์ 1 ในการอธิบาย โหมดนี้จะเป็นการกำหนดให้ใช้งานรีจิสเตอร์ TL1 เพียง 5 บิตและ TH1 ครบ 8 บิต โดย TL0 จะทำหน้าที่คล้ายกับเป็นปริสเกลเลอร์หาร 32 สัญญาณอินพุตสำหรับการนับจะเลือกจากสัญญาณนาฬิกาภายในหรือภายนอกผ่านทางขา T1 ขึ้นอยู่กับการควบคุมของบิต C/T และ GATE ในรีจิสเตอร์ TMOD บิต TR1 ในรีจิสเตอร์ TCON และสถานะลอจิกที่ขาอินพุต INT1 เมื่อ TL1 นับครบ 32 คือจาก 0-31 ก็จะส่งสัญญาณไปยัง TH1 เพื่อทำการเพิ่มค่า ดังนั้น โหมดนี้ค่าของการนับจะมีขนาด 13 บิต เมื่อการนับครบรอบก็จะทำการเซตบิต TF1 ในรีจิสเตอร์ TCON

ส่วนการทำงานในโหมดนี้ของ ไทเมอร์/เคาน์เตอร์ 0 มีลักษณะเหมือนกันทุกประการเพียงแต่เปลี่ยนรีจิสเตอร์และขาสัญญาณที่เกี่ยวข้องให้เป็นของ ไทเมอร์/เคาน์เตอร์ 0



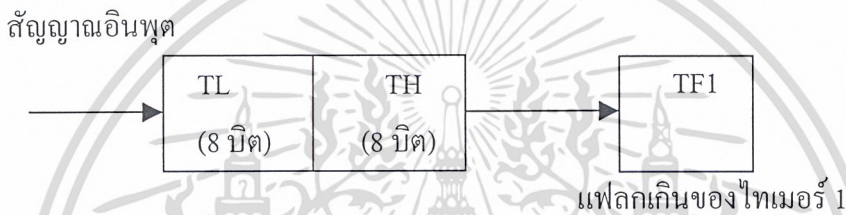
รูปที่ 2.20 ไดอะแกรมการทำงานในโหมด 0 ของ ไทเมอร์/เคาน์เตอร์ 1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. การทำงานในโหมด 1 : ไทเมอร์/เคาน์เตอร์ 16 บิต

มีไดอะแกรมการทำงานแสดงดังรูปที่ 2.21 ในที่นี้จะใช้ไทเมอร์/เคาน์เตอร์ 1 ในการอธิบายการทำงานในโหมดนี้จะคล้ายกับโหมด 0 แต่จะใช้งานรีจิสเตอร์ TL1 และ TH1 ครบ 8 บิต ดังนั้นในโหมดนี้ค่าของการนับจะมีขนาด 16 บิตคือ 0000H-FFFFH เมื่อการนับครบรอบค่าของการนับเปลี่ยนจาก FFFFH เป็น 0000H และจะเซตบิต TF1 ในรีจิสเตอร์ TCON

ส่วนการทำงานในโหมดนี้ของไทเมอร์/เคาน์เตอร์ 0 มีลักษณะเหมือนกันทุกประการเพียงแต่เปลี่ยนรีจิสเตอร์และขาสัญญาณที่เกี่ยวข้องให้เป็นของไทเมอร์/เคาน์เตอร์ 0

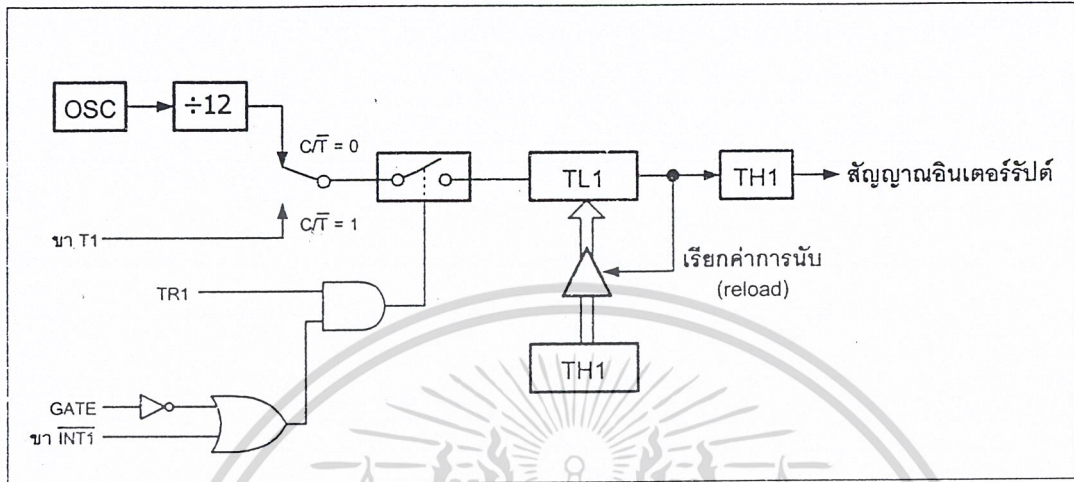


รูปที่ 2.21 ไดอะแกรมการทำงานในโหมด 1 ของไทเมอร์/เคาน์เตอร์ 1

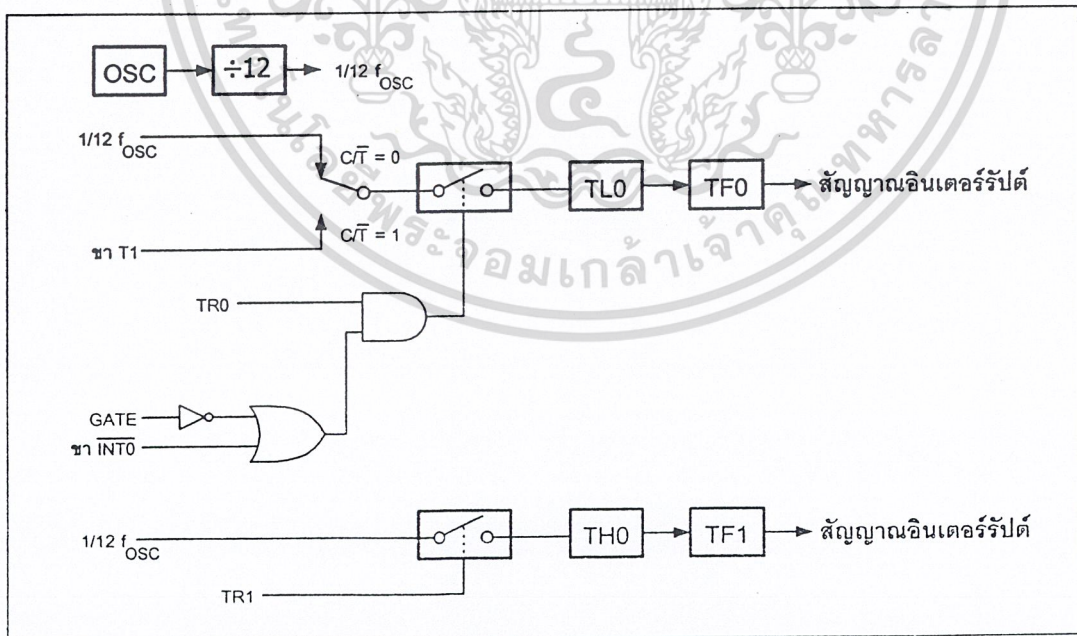
3. การทำงานในโหมด 2 : ไทเมอร์/เคาน์เตอร์ 8 บิตแบบตั้งค่าอัตโนมัติ

มีไดอะแกรมการทำงานแสดงดังรูปที่ 2.22 ในที่นี้จะใช้ไทเมอร์/เคาน์เตอร์ 1 ในการอธิบายการทำงานในโหมดนี้จะแยกรีจิสเตอร์ไทเมอร์ออกเป็น 2 ตัว ตัวละ 8 บิต โดยรีจิสเตอร์ TL1 ทำหน้าที่เป็นตัวนับค่า ส่วน TH1 ใช้ในการเก็บค่าเริ่มต้นของการนับ เมื่อเริ่มต้นการทำงานค่าของรีจิสเตอร์ TH1 จะถูกส่งไปยังรีจิสเตอร์ TL1 ทำให้เมื่อเริ่มต้นการทำงานค่าของรีจิสเตอร์ TL1 และ TH1 จะเหมือนกันเมื่อ TL1 นับถึง FFH และจะเริ่มต้นการนับรอบใหม่ จะทำการเซตบิต TF1 พร้อมกับทำการรับค่าการนับเริ่มต้นจาก TH1 ใหม่โดยอัตโนมัติ หรือเรียกกระบวนการนี้ว่า (reload) แม้ว่าจะมีการส่งค่าเริ่มต้นไปยัง TL1 แล้วก็ตาม ค่าของข้อมูลในรีจิสเตอร์ TH1 ก็ยังคงเป็นค่าเดิม ไม่มีการเปลี่ยนแปลง จนกว่าจะมีการกำหนดค่าใหม่ ด้วยกระบวนการทางซอฟต์แวร์

ส่วนการทำงานในโหมดนี้ของไทเมอร์/เคาน์เตอร์ 0 มีลักษณะเหมือนกันทุกประการเพียงแต่เปลี่ยนรีจิสเตอร์และขาสัญญาณที่เกี่ยวข้องให้เป็นของไทเมอร์/เคาน์เตอร์ 0



รูปที่ 2.22 ไดอะแกรมการทำงานในโหมด 2 ของไทเมอร์/เคาน์เตอร์ 1



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น รูปที่ 2.23 ไดอะแกรมการทำงานในโหมด 3 ของไทเมอร์/เคาน์เตอร์ 1 ที่มีการนำไปใช้

4. การทำงานในโหมด 3 : ไทเมอร์/เคาน์เตอร์ แยกส่วนหรือไทเมอร์/เคาน์เตอร์ 8 บิต

ในโหมดนี้เป็นโหมดเดียวที่การทำงานของไทเมอร์ 0 และไทเมอร์ 1 ไม่เหมือนกัน ของไทเมอร์ 1 ก่อนเมื่อเข้าสู่โหมดนี้เป็นการสั่งให้ไทเมอร์/เคาน์เตอร์หยุดนับ ค่าของการนับก่อนหน้านี้จะถูกเก็บไว้ในรีจิสเตอร์ไทเมอร์ 1 มีลักษณะการทำงานเหมือนกับการติสเอเบิลไทเมอร์/เคาน์เตอร์ 1 ด้วยการเคลียร์บิต TR1 ในรีจิสเตอร์ TCON ส่วนการทำงานของไทเมอร์ 0 ในโหมดนี้มีไคอะแกรมการทำงานแสดงดังรูปที่ 2.23 การทำงานในโหมดนี้จะแยกรีจิสเตอร์ไทเมอร์ 0 ออกเป็น 2 ตัวตัวละ 8 บิตคือรีจิสเตอร์ TH0 และ TH1 โดยแยกการทำงานออกจากกันรีจิสเตอร์ TLO สามารถเลือกการทำงานได้เหมือนกับไทเมอร์/เคาน์เตอร์ตามปกติ ส่วนรีจิสเตอร์ TH0 สามารถทำงานในโหมดไทเมอร์ได้เพียงอย่างเดียว สามารถรับสัญญาณอินพุตจากสัญญาณนาฬิกาภายในเพียงทางเดียวเท่านั้นแต่การแจ้งเตือนยังคงเหมือนเดิมหากแต่ TLO แจ้งผ่านบิต TFO ในขณะที่ TH1 จะแจ้งผ่านบิต TF1



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3

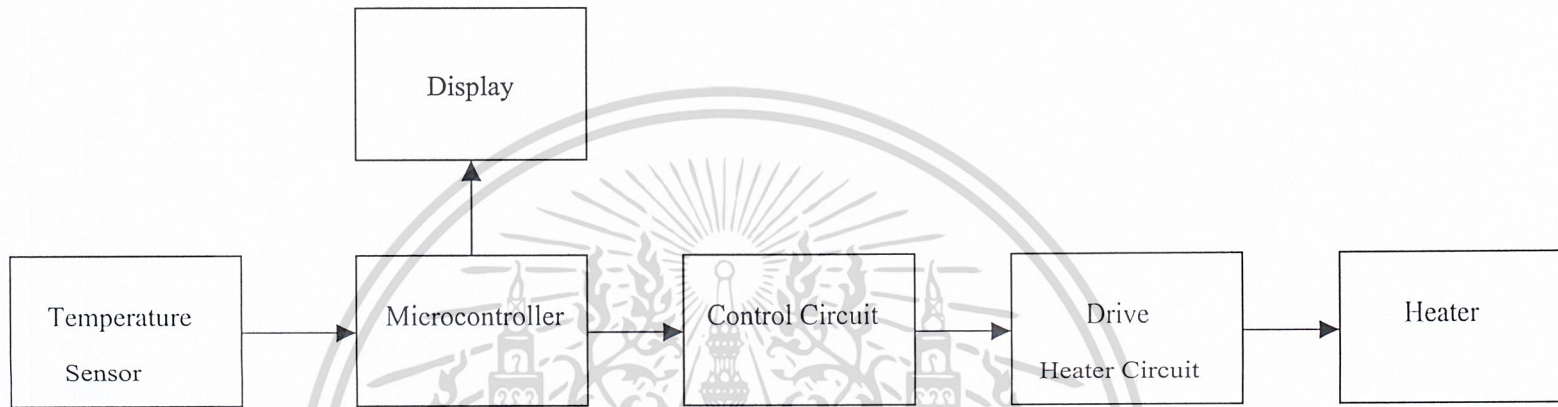
การออกแบบและหลักการทำงาน

โครงการเครื่องอบอัตโนมัติ โดยมีหลักการพื้นฐานและส่วนประกอบหลักดังนี้ คือ

1. ส่วนประมวลผล เราใช้ไมโครคอนโทรลเลอร์ในการประมวลผลและควบคุมการทำงานของส่วนต่าง ๆ
2. ส่วนกำเนิดความร้อน
3. ส่วนวัดอุณหภูมิ ใช้ไอซีดิจิตอลเทอร์โมมิเตอร์ เบอร์ DS1620 ที่มีความละเอียดของการวัดอุณหภูมิในระดับ 0.5 องศาเซลเซียส และสามารถทำงานได้ในช่วงอุณหภูมิตั้งแต่ -55 องศาเซลเซียส จนถึง 125 องศาเซลเซียส ซึ่งรับข้อมูลในรูปแบบอนุกรม
4. ส่วนรับข้อมูล
5. ส่วนแสดงผล ใช้ โมดูล LCD ขนาด 16 ตัวอักษร 1 บรรทัด
6. ส่วนจ่ายไฟเลี้ยง ใช้ไอซีเรกกูเรเตอร์ เบอร์ 7815 จ่ายไฟเลี้ยง +15V และเบอร์ 7805 จ่ายไฟเลี้ยง +5V

ซึ่งมี Block Diagram แสดงการทำงานดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.1 แสดงขั้นตอนการทำงานของเครื่องอบอัตโนมัติ

3.1 วงจรแหล่งจ่ายไฟกระแสตรง

ในการออกแบบแหล่งจ่ายไฟกระแสตรงสำหรับวงจรที่จะใช้นั้นเพื่อนำไปใช้เป็นไฟเลี้ยงให้กับไอซีตัวต่างๆที่เราจะต้องนำมาใช้ โดยในวงจรจะต้องการไฟเลี้ยงสำหรับตัวไอซี MOC 3021 จะประมาณ 15 โวลต์ซึ่งต้องการเป็นแรงดันไฟบวก +15 โวลต์ และไฟเลี้ยงสำหรับ ไอซีที่ใช้งานทางดิจิทัล คือ DS1620 , MCS-51 และ โมดูล LCD ซึ่งจะต้องการไฟเลี้ยงที่แรงดันไฟบวก +5 โวลต์

แหล่งจ่ายไฟกระแสตรงขนาด +15 โวลต์ สามารถทำได้โดยการนำแรงดันไฟกระแสสลับ (ไฟบ้าน 220 โวลต์) มาทำการผ่านหม้อแปลง STEPDOWN เพื่อให้ได้แรงดันอยู่ในระดับที่ต้องการแล้วใช้วงจร RECTIFIER ช่วยให้ได้สัญญาณเฉพาะทางซีกบวก และซีกลบ จากนั้นใช้ตัวเก็บประจุช่วยในการกรองสัญญาณให้เป็นแรงดันไฟกระแสตรงโดยใช้ไอซี LM7815 ซึ่งเป็นไอซีที่ทำงานในการควบคุมให้ได้แรงดันคงที่อยู่ที่ +15 โวลต์

แหล่งจ่ายไฟกระแสตรงขนาด +5 โวลต์ สามารถสร้างได้โดยการนำเอาแรงดันไฟกระแสตรงขนาด +15 โวลต์ที่ทำการสร้างมาควบคุมให้อยู่ในระดับแรงดันคงที่ +5 โวลต์ โดยใช้ไอซี LM7805 ซึ่งเป็นไอซีที่ทำงานในการควบคุมให้ได้แรงดันคงที่อยู่ที่ +5 โวลต์

3.2 การวัดอุณหภูมิ โดยใช้ IC เบอร์ DS1620

จากการใช้งานต่างๆ ที่ได้กล่าวมาแล้วนั้น โหมดที่จะนำมาใช้ในโครงงานนี้จะเป็นโหมด 1 Shot เพื่อง่ายต่อการนำข้อมูลมาเปรียบเทียบตามเงื่อนไขที่ตั้งไว้โดยจะมีขั้นตอนต่าง ๆ ดังนี้ เขียนค่า Config ลงใน Config register เพื่อกำหนดให้เป็นการทำงานในโหมด 1 Shot และให้ขา CLK/CONV เป็นขาที่ต้องการสัญญาณนาฬิกาธรรมดาโดยข้อมูลที่ใส่ใน Config register คือ 0Bh โดยในข้อนี้จะทำเพียงครั้งแรกในการเริ่มใช้งานแค่ครั้งเดียวเท่านั้น เพราะ Config register จะเป็น E² PROM และจำกัดการเขียนไว้ประมาณ 50,000 ครั้ง จึงไม่จำเป็นต้องเขียนทุกครั้งที่ใช้งาน

เมื่อต้องการค่าอุณหภูมิมาใช้งาน ก็ต้องใส่คำสั่งให้ DS1620 เริ่มแปลงอุณหภูมิ การที่จะทราบว่าการแปลงอุณหภูมิเสร็จหรือยังนั้น เราจะตรวจสอบได้จาก DONE BIT ของ Configuration/Status Register ว่าเป็น '1' หรือยังโดยใส่คำสั่งในการอ่าน Config ป้อนเข้าไปแล้วรอรับข้อมูลที่ได้ แล้วตรวจสอบ DONE BIT ว่าเป็น '1' หรือยัง ถ้ายัง ก็ทำซ้ำข้อ 3 อีกครั้ง ถ้าเป็น '1' แล้วก็ให้ทำขั้นตอนต่อไป

เมื่อแปลงอุณหภูมิเสร็จแล้วให้ใส่คำสั่งอ่านอุณหภูมิเข้าไป และรอรับข้อมูลอุณหภูมิที่ DS1620 ส่งออกมา เมื่อได้ค่าอุณหภูมิแล้ว DS1620 จะเข้าสู่สถานะนิ่ง และถ้าต้องการอ่านอุณหภูมิอีก ก็ย้อนไปเริ่มทำตั้งแต่การการแปลงอุณหภูมิ ในการใช้งานจริงนั้นจะต้องใช้ชุดคำสั่งดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. เขียนค่า Config ลงใน Config Register [0Bh] การเขียน Config ทำเพียงครั้งเดียว
2. การอ่านค่าอุณหภูมิ
 - 2.1 ใช้คำสั่ง Start convert : EEh
 - 2.2 การแปลง อุณหภูมิต้องใช้เวลา การดูว่าเสร็จหรือยัง โดยดูจาก DONE BIT โดยใช้คำสั่ง Read Config : ACh
 - 2.3 เมื่อแปลงเสร็จ ใช้คำสั่งอ่านค่าอุณหภูมิ Read Temp : AAh

CPU Mode	DS1620 Mode	DATA	Comments
1. T _x	R _x	0Ch	Write Config
T _x	R _x	0Bh	1 SHOT Mode
2.1 T _x	R _x	EEh	Start convert
2.2 T _x	R _x	ACh	Read Register
R _x	T _x	Data	CPUรับข้อมูล Register
2.3 T _x	R _x	AAh	Read Temp
R _x	T _x	Data	CPU รับข้อมูล Temp

ข้อมูลดิจิทัลที่ออกมาจาก IC เบอร์ DS1620 จะเข้า MCS-51 ในวงจรนี้ใช้ MCS-51 เบอร์ AT89C52 และส่วนที่แสดงผลเลือกใช้โมดูล LCD ขนาด 16 ตัวอักษร 1 บรรทัด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3 วงจรควบคุม

การออกแบบส่วนของวงจรควบคุม ควรคำนึงถึงตัวความต้านทานเพื่อให้มีการไบอัสที่เหมาะสม และการจำกัดกระแสไม่ให้เกิดความเสียหาย

การหาค่าความต้านทาน R_1 ซึ่งเป็น ตัวต้านทานไบอัสให้กับ ทรานซิสเตอร์

$$I_C = \beta I_B$$

โดย β พิจารณาที่ค่าต่ำสุดของพารามิเตอร์ ทรานซิสเตอร์ หาได้จาก Data sheet
 I_C พิจารณาที่ค่าต่ำสุดของกระแสไบอัสสำหรับ MOC 3021 จะได้

$$I_B = \frac{I_C}{\beta}$$

$$I_B = 100 \mu A$$

สามารถหา R_1 ได้จากการ วนลูป ที่บริเวณ ขา Base และ Emitter ของตัวทรานซิสเตอร์

$$V_B = I_B R_1 + V_{BE}$$

$$5 = I_B R_1 + 0.7$$

$$R_1 = \frac{5 - 0.7}{I_B}$$

$$R_1 = 43 K\Omega$$

ในที่นี้เลือกใช้ 10 Kohm กระแสเบส เพื่อใช้กระแสมากกว่ากระแสที่คำนวณได้ที่ใช้ในการไบอัส ทำให้แน่นอนว่าจะสามารถทำงานได้

ส่วน R_2 จะเป็นตัวความต้านทานที่จำกัดกระแส หากเกิดการเสียหายขึ้นกับทรานซิสเตอร์ เพื่อไม่ให้ MOC 3021 เกิดความเสียหาย

$$\frac{V}{I} = R$$

โดย $V = 15 \text{ Volt}$

I_{\max} คือ กระแสที่ MOC 3021 สามารถทนได้ ซึ่งหาได้จากใน Data Sheet (50 mA)

$$R_2 = \frac{15}{I_{\max}}$$

$$R_2 = 300 \Omega$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากการคำนวณแสดงว่าต้องใช้ R2 อย่างน้อย 300 ohm แต่ในที่นี้เลือกใช้ 1 Kohm เพื่อให้สามารถป้องกันไม่ให้เกิดความเสียหาย การเลือกเช่นนี้จะทำให้กระแสสูงสุดต่ำกว่า กระแสสูงสุดใน Data Sheet

3.4 วงจรขับเคลื่อนเครื่องกำเนิดความร้อน

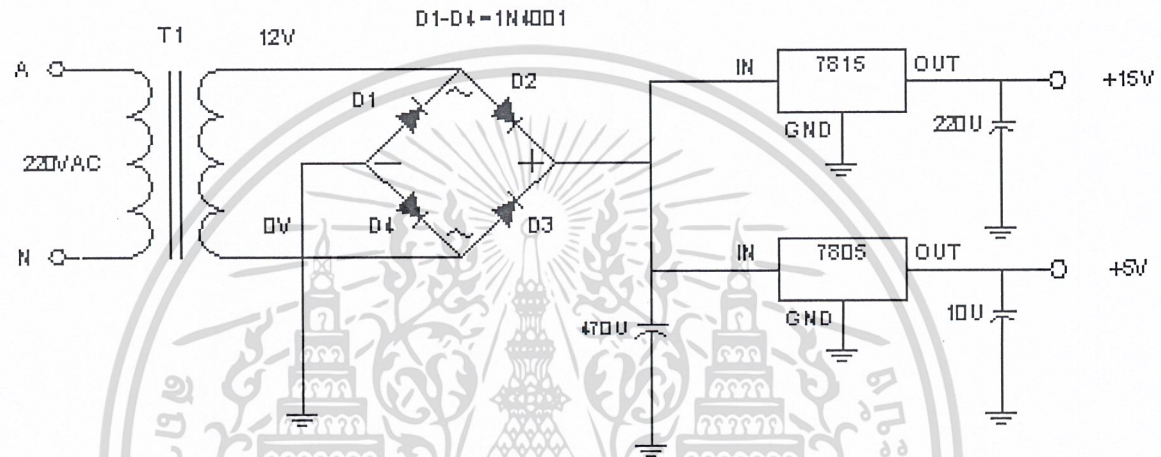
วงจรที่ใช้เป็นวงจรที่สามารถควบคุมกำลังงานไฟฟ้าโดยใช้สัญญาณดังที่กล่าวมาแล้วเป็นสัญญาณที่ได้มาจากการเขียนโปรแกรม MCS-51 ที่เกี่ยวข้องกับอุณหภูมิที่สามารถวัดได้

วงจรถูกนำมาใช้นั้นคือวงจรที่ใช้ SCR 2 ตัวมาเป็นตัวขับเคลื่อนเครื่องกำเนิดความร้อน เพื่อให้สามารถขับเคลื่อนกระแสไฟสลับได้แบบ Full Wave การที่นำ SCR มาใช้เนื่องจากกระแสที่ใช้ในการขับเคลื่อนเครื่องกำเนิดความร้อนเป็นกระแสที่สูง ซึ่งในหากใช้ตัวไดรเอดเป็นตัวขับนั้นการหาไดรเอดที่เหมาะสมมาใช้จะมีราคาแพง และหายาก แต่ในการใช้ SCR ที่สามารถใช้ในการทำงานในกระแสสูงได้นั้น การทริกเพื่อให้ตัว SCR ทำงานได้นั้นจะต้องขยายกระแสด้วยการใช้ไดรเอดก่อน เพราะ SCR กระแสสูงนั้นต้องใช้กระแสในการทริกให้เกิดการทำงานสูงด้วย

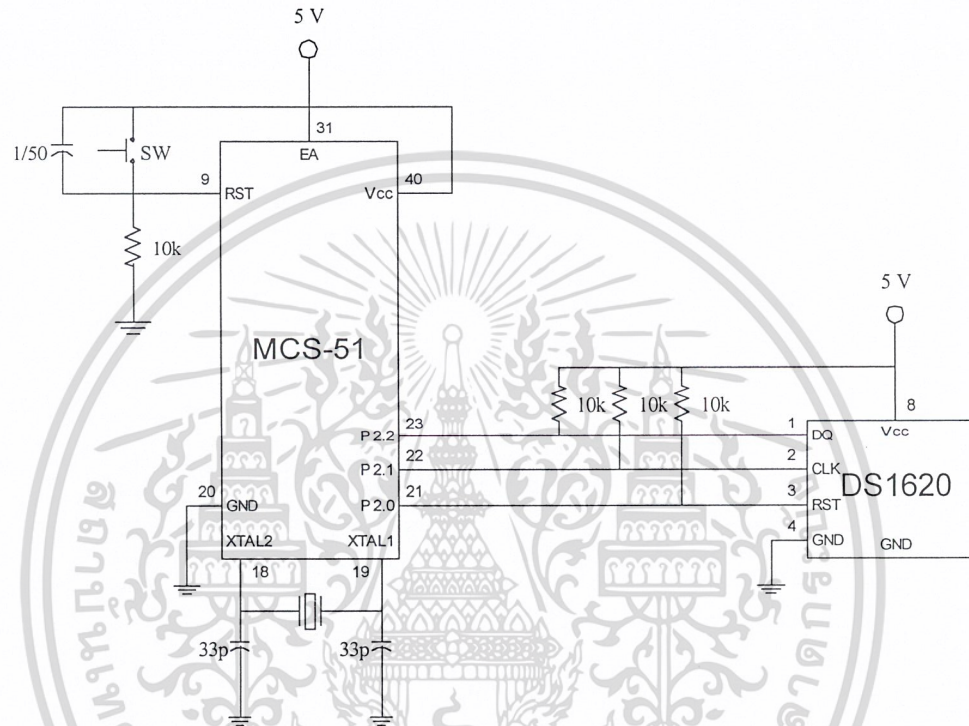
การเลือกใช้ SCR และ ไดรเอด นั้นต้องคำนึงถึง กระแส และแรงดันที่ต้องใช้ในการขับตัวอุปกรณ์ต่างๆ ซึ่งในที่นี้เป็นเครื่องกำเนิดความร้อนที่ใช้แรงดันกระแสไฟสลับจากไฟบ้านที่มีแรงดันสูงถึง $220 V_{rms}$ หรือ $311 V_p$ เพราะฉะนั้นอุปกรณ์ที่จะเลือกใช้ต้องทนแรงดันได้มากกว่า $220 V_{rms}$ หรือ $311 V_p$ และต้องทนกระแสกระชากสูงสุด หรือได้มากกว่า 15 เท่าของกระแสโหลด

ในที่นี้ได้้นำเครื่องกำเนิดความร้อนขนาดใหญ่ ซึ่งกระแสโหลดเป็น 22 แอมป์ และต้องทนแรงดันได้มากกว่า $311 V_p$

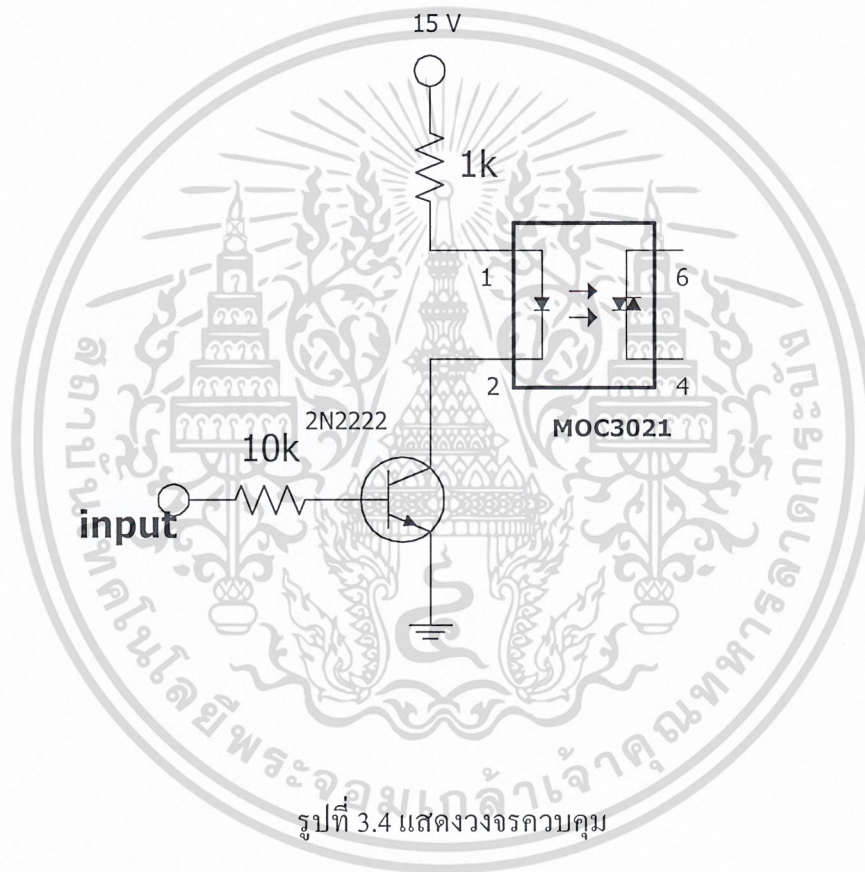
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



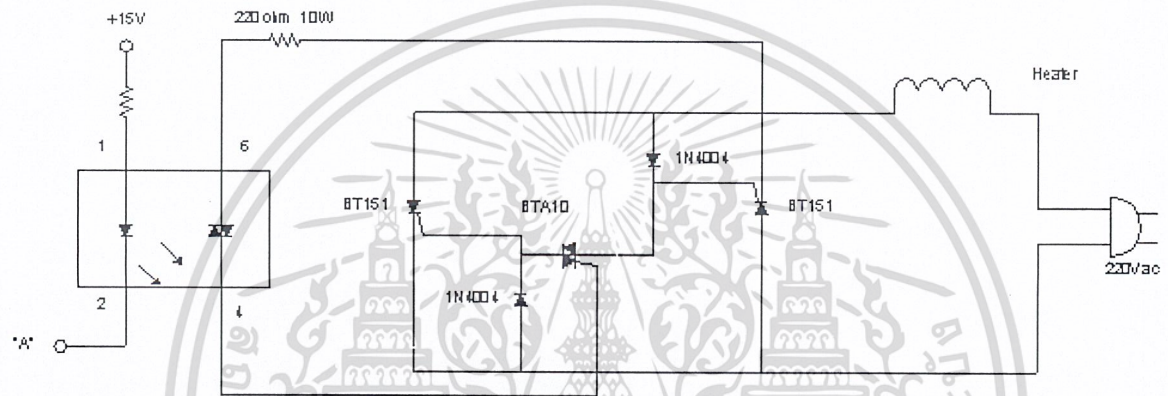
รูปที่ 3.2 แสดงแหล่งจ่ายแรงดันกระแสตรง



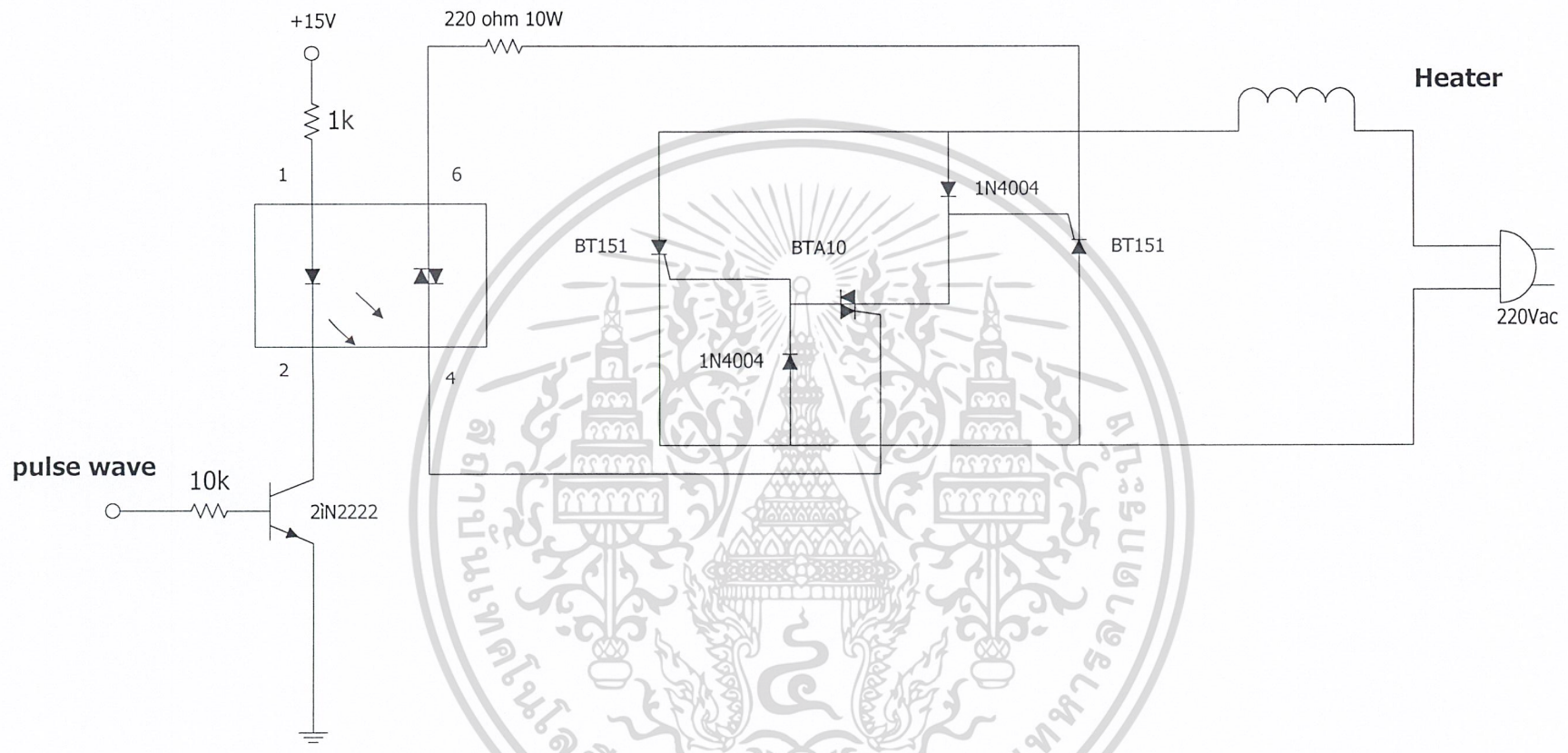
รูปที่ 3.3 แสดงวงจร MCS-51 ต่อเข้ากับ DS-1620



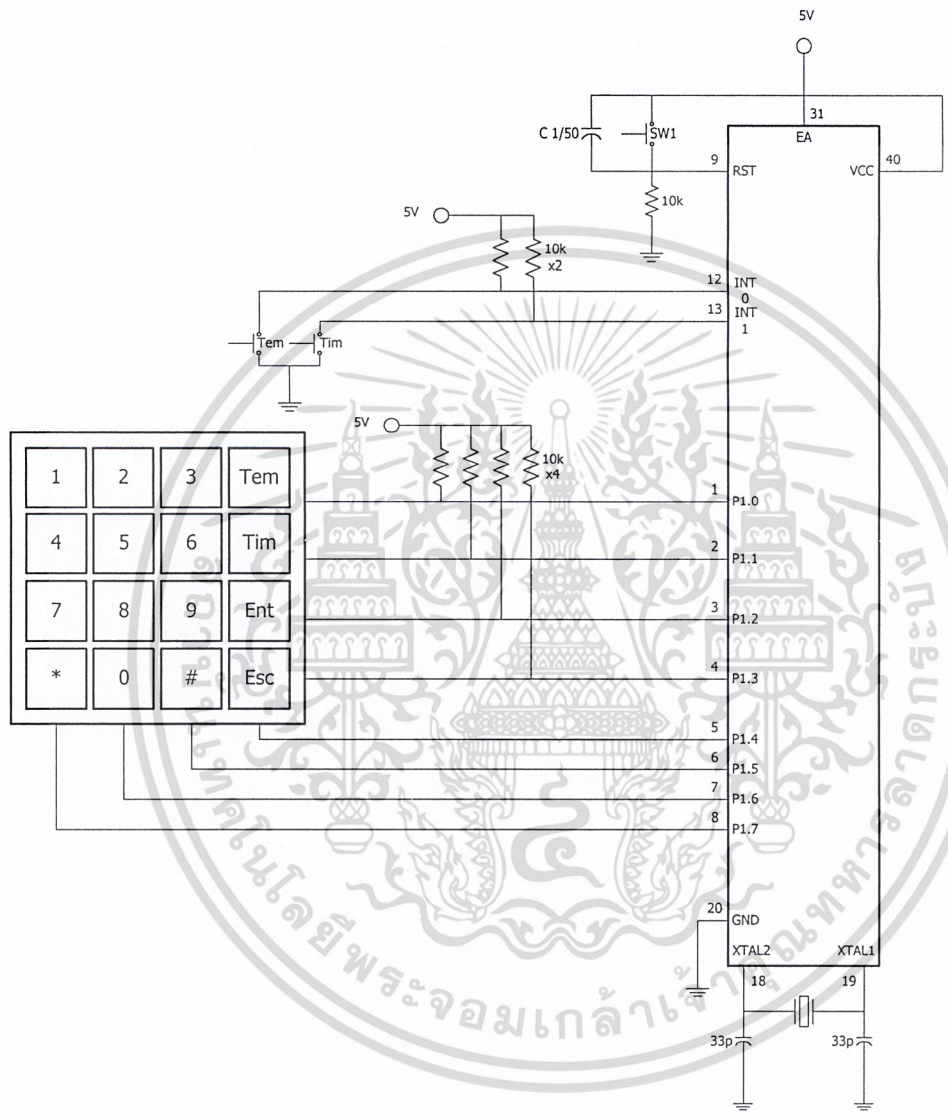
รูปที่ 3.4 แสดงวงจรควบคุม



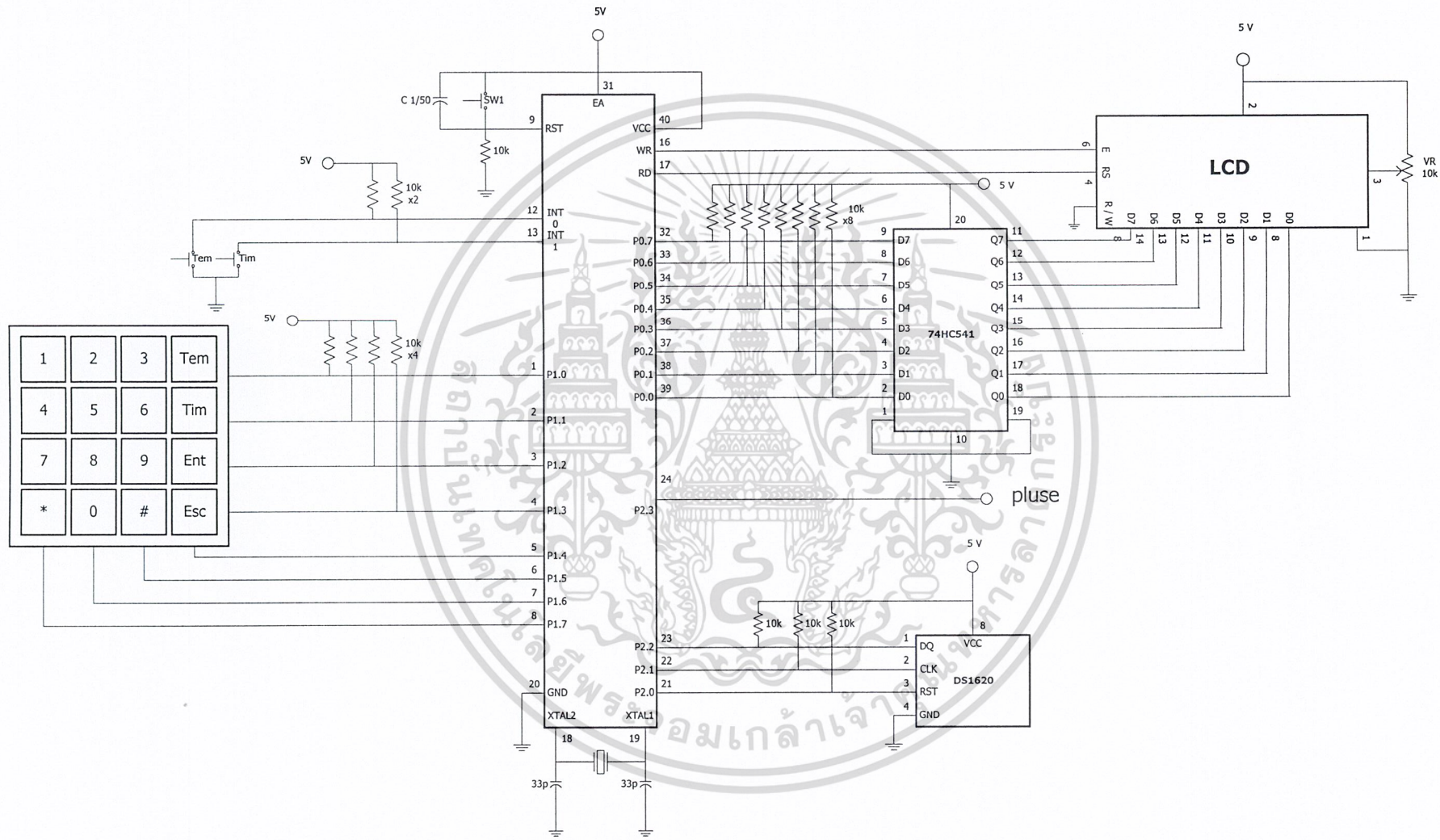
รูปที่ 3.5 แสดงวงจร Drive Heater



รูปที่ 3.6 แสดงวงจรควบคุมกำลังงานไฟฟ้า

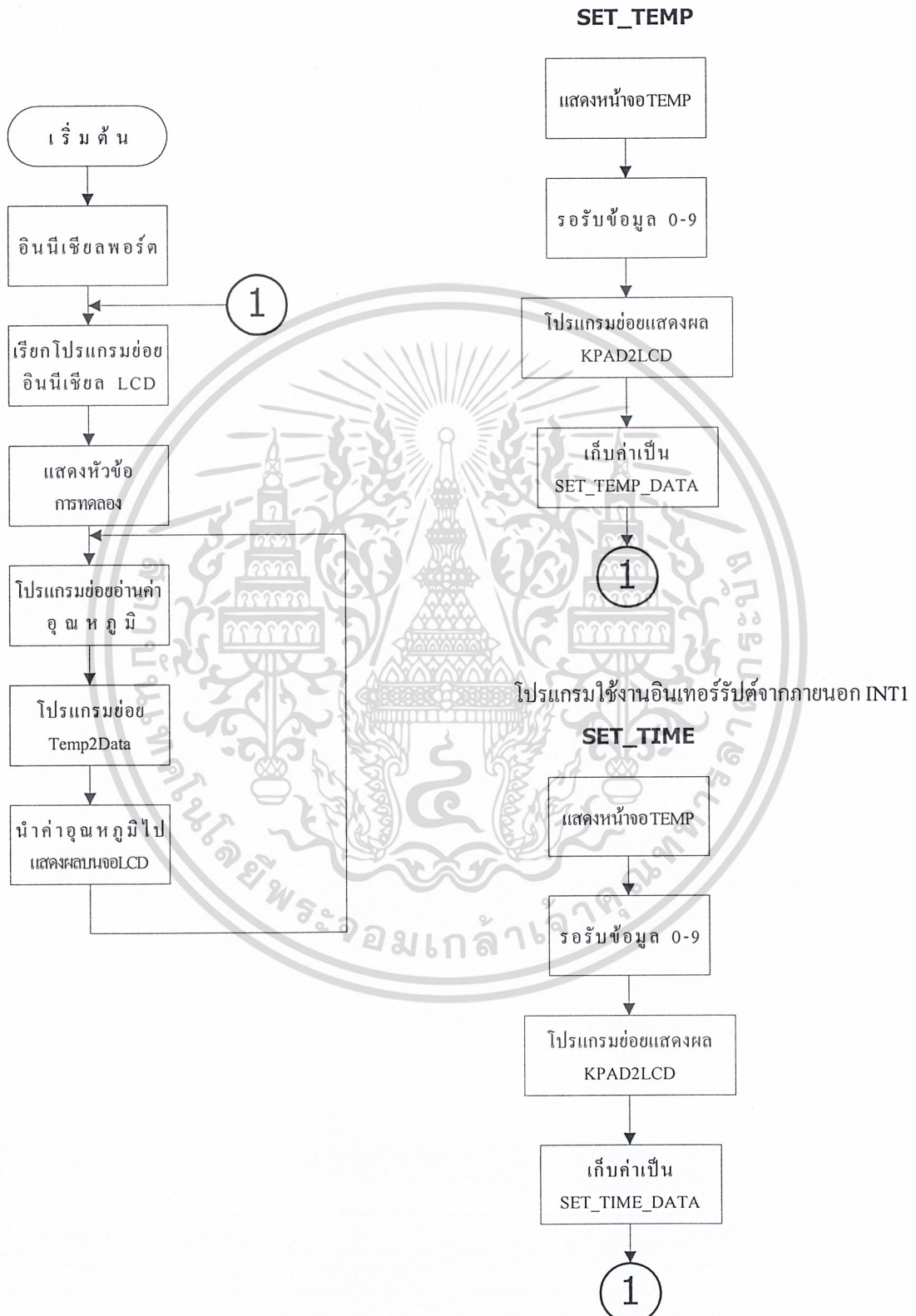


รูปที่ 3.7 แสดงวงจรเชื่อมต่อคีย์แพดเข้ากับไมโครคอนโทรลเลอร์ MCS-51



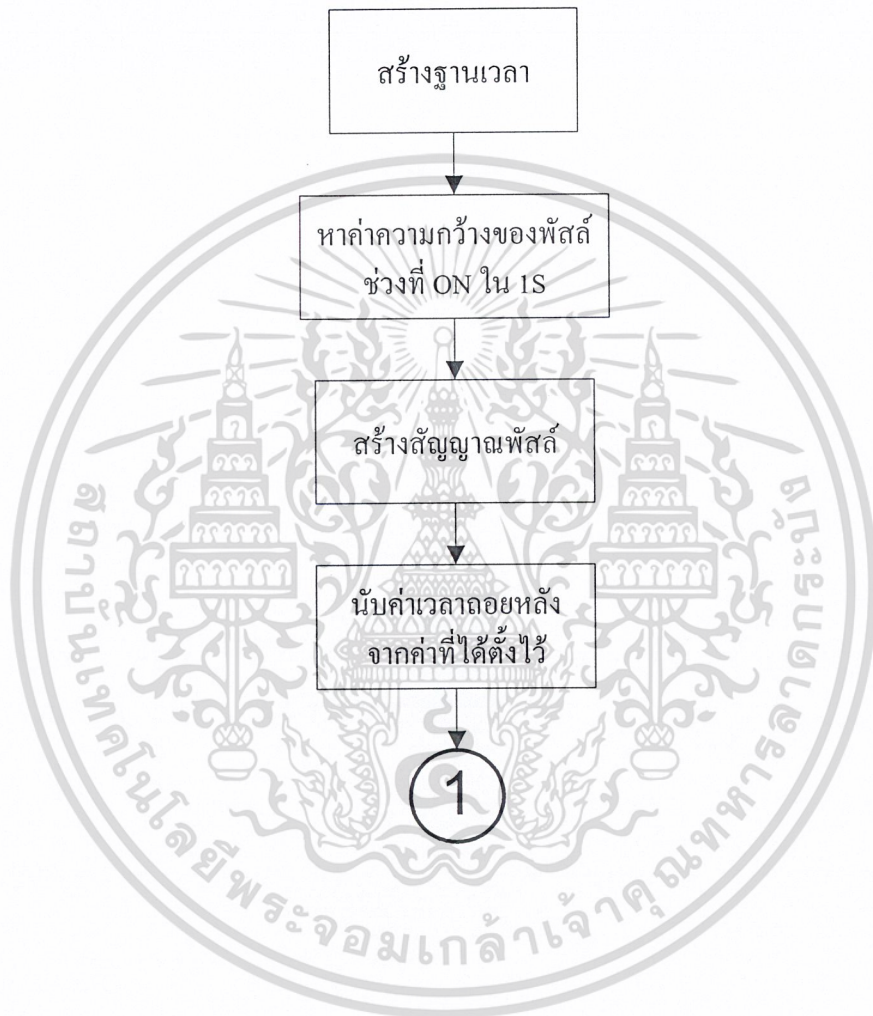
รูปที่ 3.8 แสดงวงจรรวมของเครื่องอบอัตโนมัติ

โปรแกรมใช้งานอินเทอร์รับต์จากภายนอก INTO



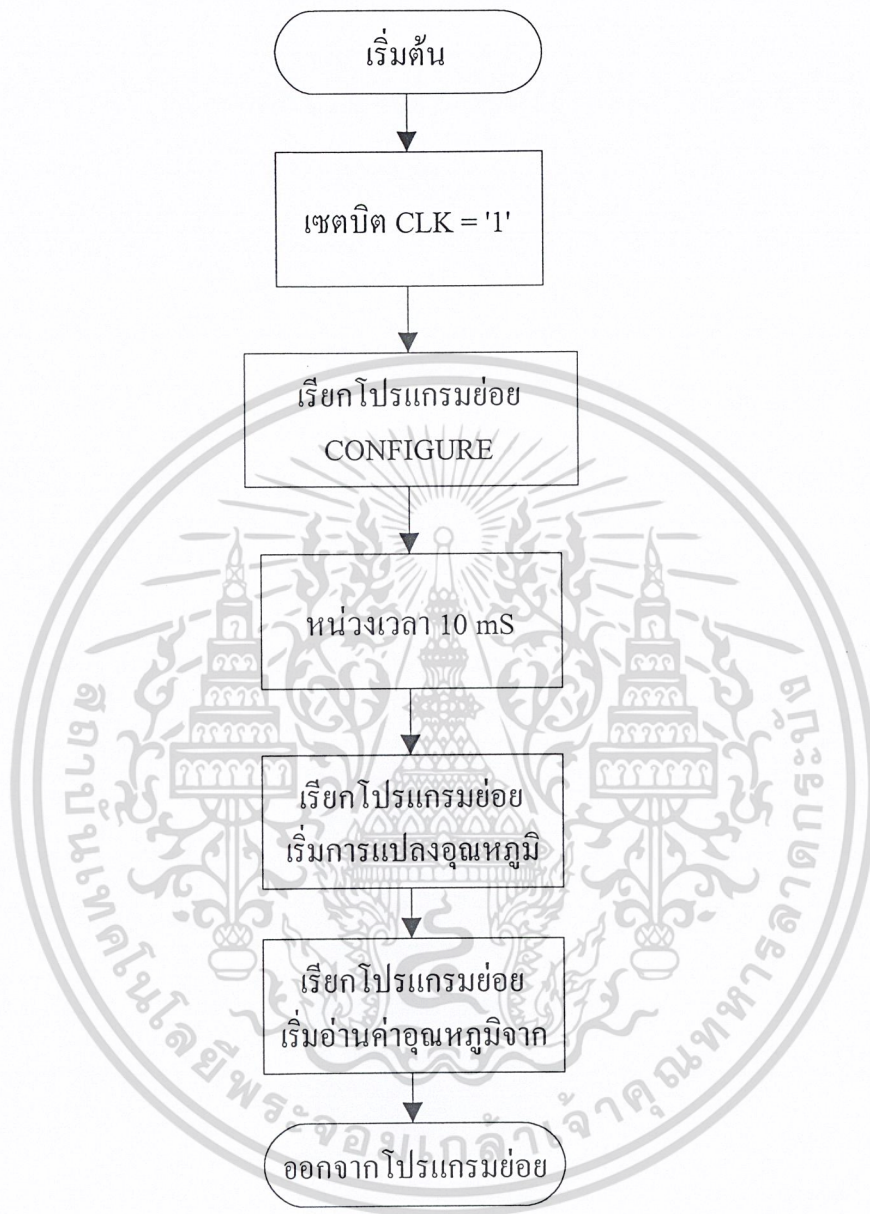
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น รูปที่ 3.8 แสดงโฟลวชาร์ตของโปรแกรมควบคุมเครื่องอบอัดโนมัต

อินเทอร์รัปต์จากไทเมอร์/เคาน์เตอร์ 0



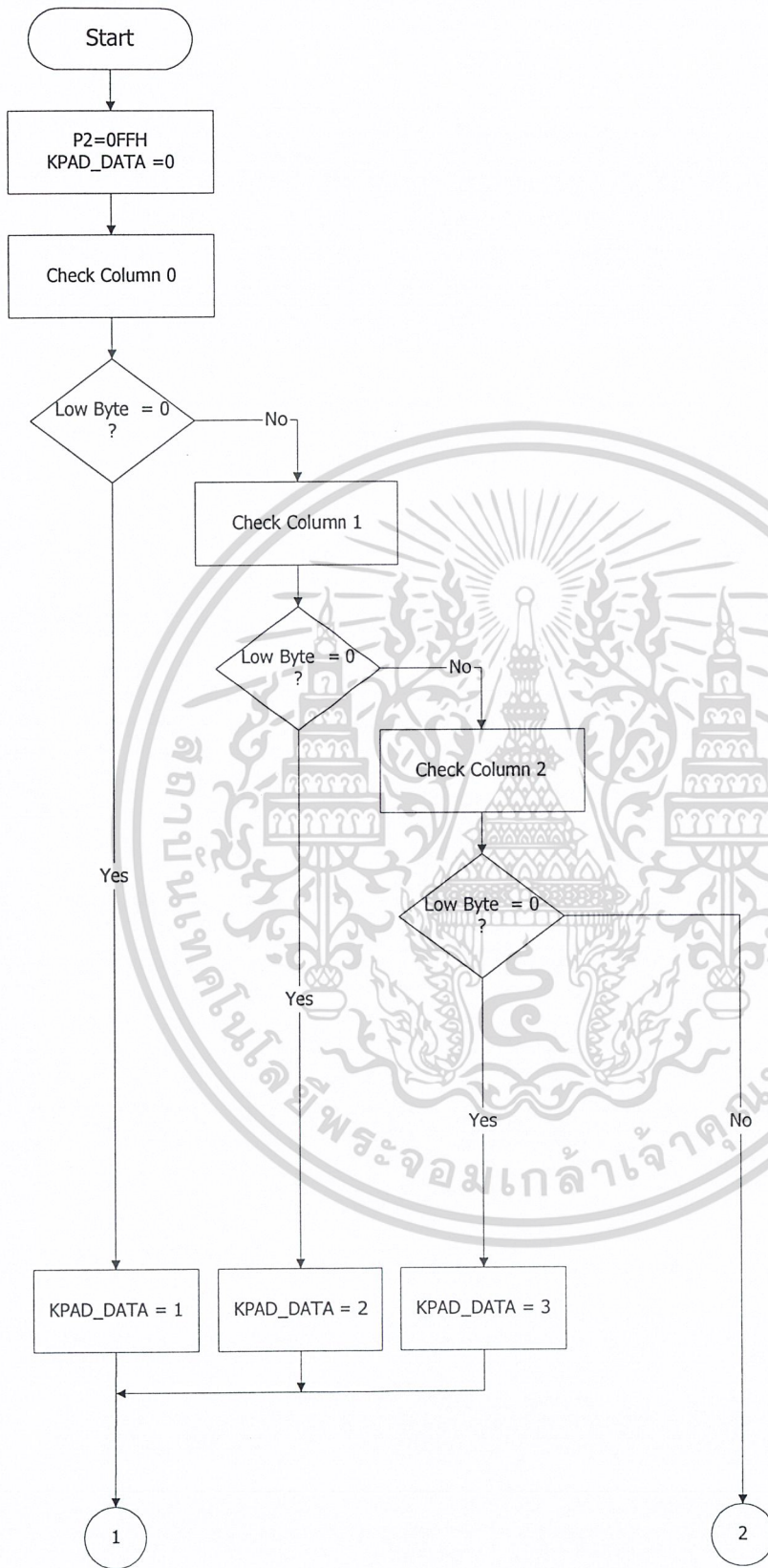
รูปที่ 3.9 แสดงโฟลวชาร์ตการอินเทอร์รัปต์ของไทเมอร์/เคาน์เตอร์ 0 การทำงานในโหมด 2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

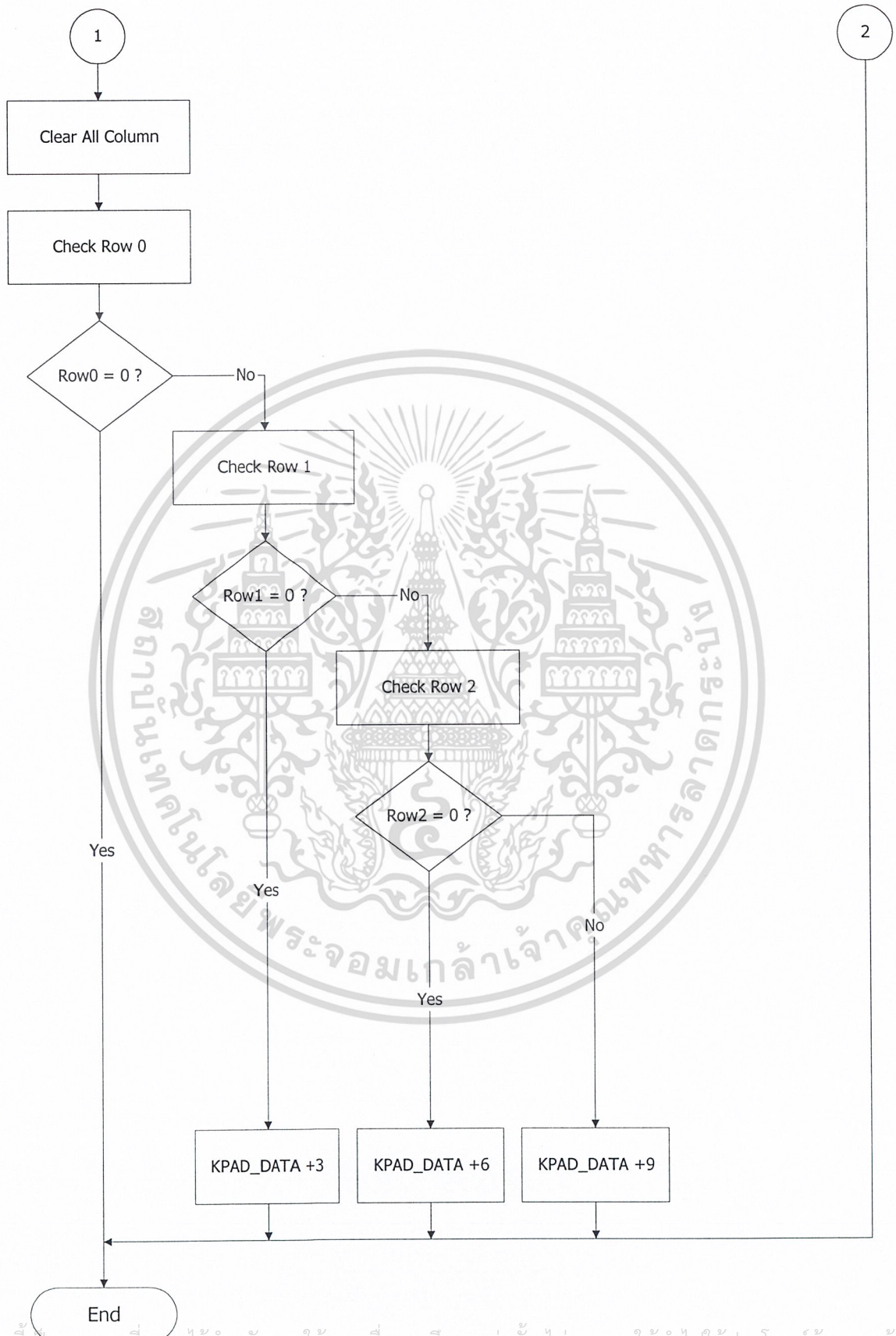


รูปที่ 3.10 แสดงโฟลวชาร์ต โปรแกรมย่อยการอ่านค่าอุณหภูมิ IC DS1620

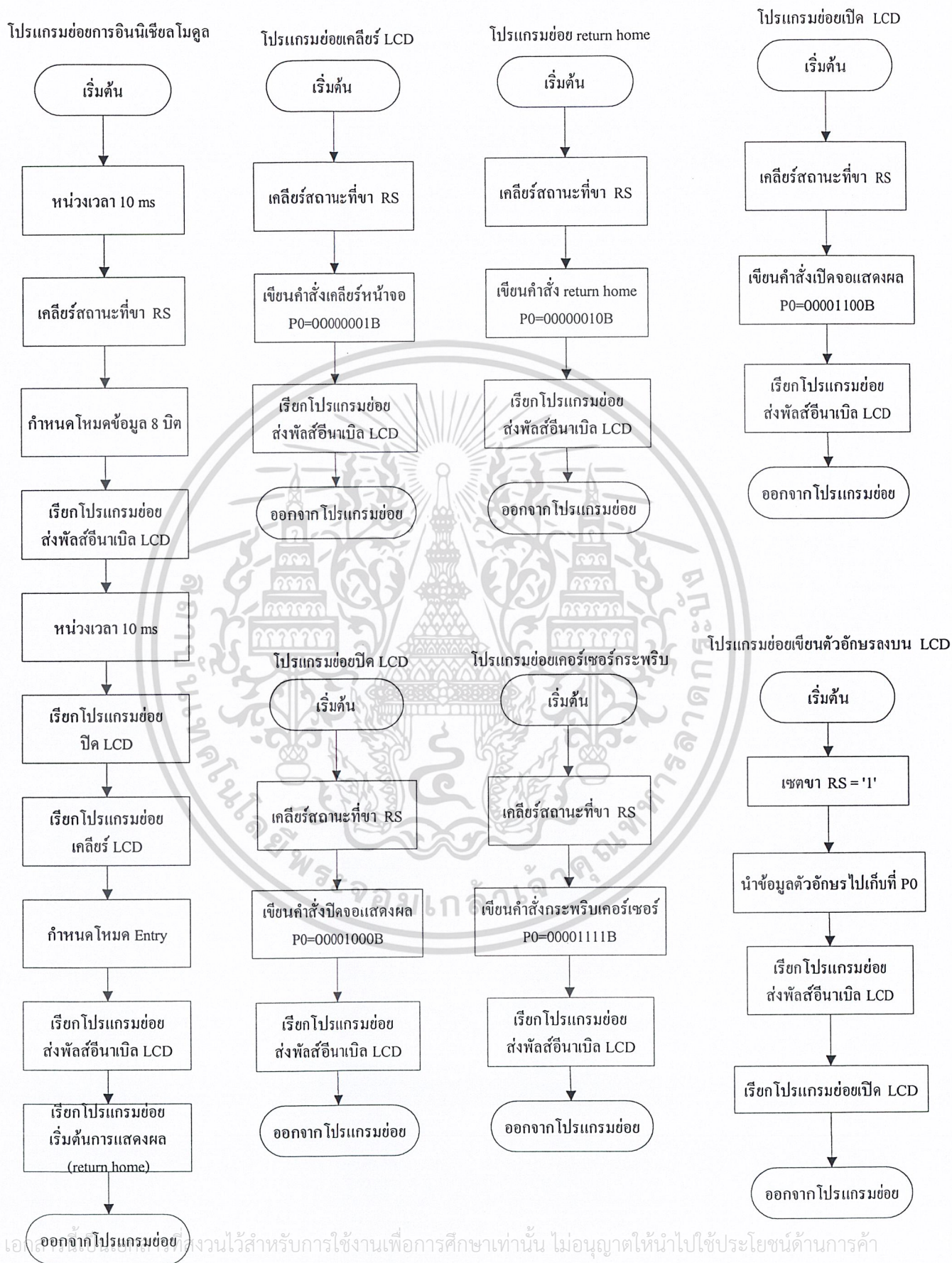
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเฉพาะในวงจำกัดเท่านั้น ไม่ควรเผยแพร่หรือใช้ประโยชน์ด้านการค้า
 รูปที่ 3.11 ไฟลวชาร์ตโปรแกรมย่อยการอ่านค่าคีย์แพด
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 รูปที่ 3.12 ไฟลวซาร์ตโปรแกรมย่อยการอ่านค่าคีย์แพด
 ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมีเหตุดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นทรัพย์สินทางปัญญาของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี ซึ่งสงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 3.13 โฟลวชาร์ตโปรแกรมย่อยการแสดงผลข้อความบน โมดูล LCD 16 ตัว 1 บรรทัด

บทที่ 4

ผลการทดลอง

4.1 ความถูกต้องของตัวดิจิตอลเทอร์โมมิเตอร์

ในการทดลองความถูกต้องของดิจิตอลเทอร์โมมิเตอร์นั้น เราได้นำเทอร์โมมิเตอร์แบบปรอท มาเป็นอุณหภูมิอ้างอิง โดยการนำเทอร์โมมิเตอร์ทั้งสองแบบไปวัดอุณหภูมิพร้อมกันในสถานะต่างๆ ดังแสดงผลในตารางที่ 4.1

ตารางที่ 4.1 แสดงการเปรียบเทียบอุณหภูมิระหว่างเทอร์โมมิเตอร์แบบปรอทกับ
ดิจิตอลเทอร์โมมิเตอร์

สถานะ	เทอร์โมมิเตอร์แบบปรอท	ดิจิตอลเทอร์โมมิเตอร์
ในห้องที่ไม่ได้เปิดเครื่อง ปรับอากาศตอนเช้า	26°C	26 °C
ในห้องที่ไม่ได้เปิดเครื่อง ปรับอากาศตอนเที่ยง	30°C	30.5°C
ในห้องเปิดเครื่องปรับอากาศ	25°C	25°C
ในห้องเก็บของตอนเที่ยง	36°C	36.5°C

จากตารางที่ 4.1 จะเห็นว่าอุณหภูมิที่ได้จากดิจิตอลเทอร์โมมิเตอร์มีค่าใกล้เคียงกับเทอร์โมมิเตอร์แบบปรอท ซึ่งถือว่ามีความถูกต้องใช้ได้ นอกจากนี้เรายังได้สังเกตเห็นว่าดิจิตอลเทอร์โมมิเตอร์ยังมีความไวต่อการเปลี่ยนแปลงของอุณหภูมิสูงกว่าเทอร์โมมิเตอร์แบบปรอทมาก

4.2 ผลการควบคุมความร้อน

จากการทดลองการใช้งานของเครื่องอบอัตโนมัติให้ความร้อนนั้น เราจะทำการตั้งอุณหภูมิไว้ที่ค่าๆหนึ่ง แล้วทดลองควบคุมความร้อน โดยบันทึกผลเวลาที่ช่วงของอุณหภูมิเปลี่ยน ดังแสดงได้ดังผลในตารางที่ 4.2

ตารางที่ 4.2 แสดงผลการควบคุมความร้อนกับเวลา

อุณหภูมิเริ่มต้น 40°C

อุณหภูมิตั้งค่า 50°C

อุณหภูมิ	เวลา
40	0
41	0.20
42	0.35
43	0.47
44	0.58
45	1.08
46	1.15
47	1.25
48	1.32
49	1.40
50	1.05
51	2.05
50	3.30
49	4.10
48	4.40

จากผลการทดลอง จะเห็นได้ว่าเครื่องจะใช้เวลาในการเพิ่มค่าความร้อนขึ้นจนถึงอุณหภูมิที่ต้องการประมาณ 5 นาที โดยในช่วงแรกนั้นอุณหภูมิจะมีการเพิ่มขึ้นเรื่อย จนเมื่อใกล้ถึงอุณหภูมิที่ต้องการ อุณหภูมิจะแกว่งขึ้นลงอยู่ในช่วงประมาณ 1-2 องศาเซลเซียส

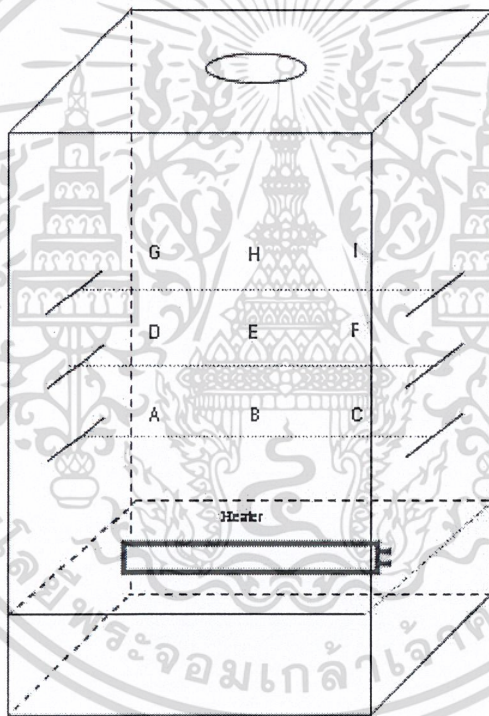
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3 ผลการควบคุมเวลา

สำหรับการตั้งค่าเวลานั้นสามารถทำงานได้อย่างถูกต้อง เพราะว่าไทม์เทอร์มในตัวไมโครคอนโทรลเลอร์นั้นมีคาบเวลาในการนับที่แน่นอนและแม่นยำ จึงทำให้แถบจะไม่มี ความผิดพลาดเกิดขึ้นเลย ในส่วนนี้

4.4 การกระจายความร้อนรอบบริเวณที่ใช้งาน

สำหรับผลการกระจายความร้อนที่ได้จากตัวฮีตเตอร์นั้น เราทดสอบโดยการควบคุมให้ฮีตเตอร์แผ่ความร้อนในระดับหนึ่ง แล้วปล่อยให้ฮีตเตอร์แผ่ความร้อนออกมาจนไม่ทำให้อุณหภูมิเพิ่มขึ้นได้แล้ว โดยนำเทอร์โมมิเตอร์ไปวัดอุณหภูมิที่จุดต่างๆ ดังรูปที่ 4.1



รูปที่ 4.1 แสดงการทดสอบการกระจายความร้อน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คั้งแสดงผลการทดลองไว้ในตารางที่ 4.3

ตารางที่ 4.3 แสดงผลการวัดอุณหภูมิบริเวณรอบๆพื้นที่การใช้งาน

ตั้งอุณหภูมิไว้ที่ 34°C

วัดที่จุด	อุณหภูมิ (°C)
A	35
B	35
C	35
D	34.5
E	35
F	34.5
G	34.5
H	35
I	34.5

จากผลการทดลอง จะเห็นได้ว่าการแผ่กระจายความร้อนนั้นจะไม่เท่ากันทั่วทั้งบริเวณที่ใช้ งาน ซึ่งจะขึ้นอยู่กับอุณหภูมิของสภาพแวดล้อมด้วยว่ามีผลในการเอื้ออำนวยต่อการแผ่ความร้อนมาก น้อยเพียงใด แต่สังเกตได้ว่าจะมีความแตกต่างกันไม่มาก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

สรุปและวิจารณ์ผลการทดลอง

จากผลการทดลอง จะเห็นได้ว่าเราสามารถรักษาอุณหภูมิตามที่กำหนดเอาไว้ได้ โดยจะมีการแกว่งของอุณหภูมิอยู่ในช่วง ± 2 องศาเซลเซียสจากอุณหภูมิที่ตั้งเอาไว้ ซึ่งความผิดพลาดที่เกิดขึ้นนี้อาจเกิดมาจากหลายสาเหตุด้วยกันดังนี้

1. การเขียนโปรแกรมควบคุมจากไมโครคอนโทรลเลอร์ ในส่วนนี้เป็นส่วนที่ค่อนข้างซับซ้อน จึงควรตรวจสอบความผิดพลาดที่เกิดขึ้นได้ยาก แต่สามารถที่จะควบคุมความผิดพลาดให้น้อยที่สุด โดยขึ้นอยู่กับตัวผู้เขียนเองว่าจะมีความสามารถแค่ไหน
2. จากการทำการทดลองหาตำแหน่งการติดตั้งอุณหภูมิ ว่าตำแหน่งไหนเหมาะสมที่สุดสำหรับการติดตั้งเซ็นเซอร์นั้น เราควรที่จะติดตั้งพร้อมกันแล้วทำการวัดอุณหภูมิพร้อมกันทั้ง 9 ตำแหน่ง แต่เนื่องจากตัวเซ็นเซอร์ที่ทำการวัดมีราคาแพง จึงต้องมีการวัดทีละจุด ทำให้แต่ละครั้งที่ทำการวัด ค่าอุณหภูมิที่อ่านได้อาจคลาดเคลื่อน ทำให้หาตำแหน่งที่ติดตั้งเซ็นเซอร์ได้ไม่เหมาะสมเท่าที่ควร
3. จากการทดลอง เราไม่ได้มีการคำนวณขนาดพัดลมที่นำมาใช้งาน ไม่มีการคำนวณเรื่องความชื้น และช่องระบายอากาศภายในตู้ว่าเพียงพอต่อการถ่ายเทความร้อนหรือไม่ ซึ่งถ้ามีการพัฒนาต่อในเรื่องนี้ คงต้องมีการศึกษาเรื่องการออกแบบตู้ที่เหมาะสม รวมทั้งตัวอุปกรณ์ต่างๆ ที่ใช้ด้วย

นอกจากนี้สภาพแวดล้อมที่ใช้ทำการใช้งานก็มีผลต่อความเร็วในการเพิ่มอุณหภูมิด้วย โดยถ้านำเครื่องอบไปใช้งานที่สภาพแวดล้อมที่มีอุณหภูมิต่ำ จะทำให้การเพิ่มความร้อนเป็นไปได้ยากและช้า แต่ถ้านำไปใช้งานที่สภาพแวดล้อมที่มีอุณหภูมิสูง จะทำให้การเพิ่มความร้อนทำได้ง่ายและเร็วกว่า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขั้นตอนการใช้งานเครื่องอบอัตโนมัติ

1. เสียบปลั๊กอิทเตอร์ และปลั๊กส่วนควบคุม
2. สามารถตั้งอุณหภูมิที่ต้องการได้โดยกดปุ่ม Tem โดยอุณหภูมิที่ตั้งจะจำกัดอยู่ในช่วงประมาณ 35 ถึง 99 องศาเซลเซียส โดยจะไม่สามารถตั้งอุณหภูมิได้ในระดับ 0.5 องศาเซลเซียส
3. สามารถตั้งเวลาในการทำงานของตู้อบได้ โดยกดปุ่ม Tim โดยสามารถตั้งเวลาได้สูงสุด 99 นาที โดยต้องตั้งเวลารอบ 2 หลัก เช่น เวลา 5 นาที ต้องกดตัวเลข 05
4. เปิดสวิตช์ POWER ปิดลม เมื่อต้องให้พัดลมทำงาน

1	2	3	Tem
4	5	6	Tim
7	8	9	Ent
*	0	#	Esc

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เอกสารอ้างอิง

ชัยวัฒน์ ลิ้มพรจิตรวิไล , วรพจน์ กรแก้ววัฒนกุล . “เรียนรู้และปฏิบัติการไมโครคอนโทรลเลอร์ MCS-51” , บริษัท อินโนเวตีฟ เอ็กเพอริเมนต์ จำกัด

ยีน ภู่วรรณ . “อิเล็กทรอนิกส์อุตสาหกรรม” ,บริษัท ซีเอ็ดยูเคชั่น จำกัด

www.silaresearch.com

www.lycos.com

www.kmitl.ac.th/electronics

www.guestlink.com



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

;*****
; Program      : AUTOMATICS OVEN
; For         : MCS-51 Microcontroller System
;*****

;-----
; Define Port&Pin Name
;-----

KROW0          BIT        P1.0
KROW1          BIT        P1.1
KROW2          BIT        P1.2
KROW3          BIT        P1.3
KCOL3          BIT        P1.4
KCOL2          BIT        P1.5
KCOL1          BIT        P1.6
KCOL0          BIT        P1.7
RST            BIT        P2.0
CLK            BIT        P2.1
DQ             BIT        P2.2
PULSE_BIT     BIT        P2.3
FAN            BIT        P2.4
LCD_EN        BIT        P3.6
LCD_RS        BIT        P3.7

;-----
; Define User Register
;-----

LCD_ADDR      EQU        030H
LCD_DATA      EQU        031H
KDATA         EQU        032H
TEMP_DATA1    EQU        033H
TEMP_DATA2    EQU        034H
DIGIT         EQU        035H
TEN           EQU        036H
HUNDRED       EQU        037H
TEMP_READ     EQU        03BH
TEMP_SET      EQU        03CH
TEMP_OLD      EQU        03DH
TEMP_DIFF     EQU        03EH
T200u         EQU        040H
T10m          EQU        041H
T1s           EQU        042H
T1MIN         EQU        043H
TOFF          EQU        045H
TON           EQU        046H
TIME_SET      EQU        047H
TIME_DATA1    EQU        048H
TIME_DATA2    EQU        049H
TIME_DATA3    EQU        04AH
TIME1         EQU        04BH
TIME2         EQU        04CH
TIME3         EQU        04DH
DATA          EQU        021H

;-----
; Main Program.
;-----

                ORG        0000H
                LJMP       MAIN
                ORG        0003H
INT0_SERV:     ACALL       SUB_INT0
                RETI
                ORG        000BH
TIMER0_SERV:   ACALL       SUB_TIMER
                RETI
                ORG        0013H
INT1_SERV:     ACALL       SUB_INT1
                RETI
MAIN:          MOV         TMOD,#02H
                MOV         TLO,#38H
                MOV         TH0,#38H
                MOV         IE,#10000111B

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

MOV      P0,#00000000B
MOV      P1,#11111111B
MOV      P3,#00111111B
MOV      P2,#00000000B
MOV      TOFF,#OFFH
MOV      TEMP_SET,#00H
TITLE:  ACALL  INIT_LCD
        MOV   LCD_ADDR,#000H
        ACALL SET_ADDR_LCD
        MOV   DPTR,#TITLE1
        ACALL WRLINE_LCD
        ACALL DELAY_1s
        MOV   LCD_ADDR,#000H
        ACALL SET_ADDR_LCD
        MOV   DPTR,#TITLE2
        ACALL WRLINE_LCD
        ACALL DELAY_1s
        MOV   TEMP_DATA1,#30H
        MOV   TEMP_DATA2,#30H
        MOV   TIME_DATA1,#0
        MOV   TIME_DATA2,#0
        MOV   TIME1,#30H
        MOV   TIME2,#30H
L_READTEMP: CLR  PULSE_BIT
            SETB CLK
            ACALL START_CONVERT
            ACALL DELAY_1s
            ACALL READ_TEMPERATURE
            MOV   A, R1
            CLR  C
            RRC  A
            MOV   DATA.1,C
            MOV   TEMP_READ,A
            MOV   B,#100
            DIV  AB
            ADD  A,#30H
            MOV   HUNDRED,A
            MOV   TEN,B
            MOV   A,TEN
            MOV   B,#10
            DIV  AB
            ADD  A,#30H
            MOV   TEN,A
            MOV   A,B
            ADD  A,#30H
            MOV   DIGIT,A
            ACALL LCD_CLR
            MOV   LCD_ADDR,#000H
            ACALL SET_ADDR_LCD
            MOV   DPTR,#DISPLAYSHOW
            ACALL WRLINE_LCD
            MOV   LCD_ADDR,#002H
            ACALL SET_ADDR_LCD
            MOV   LCD_DATA,TEN
            ACALL WRCHAR_LCD
            MOV   LCD_DATA,DIGIT
            ACALL WRCHAR_LCD
            MOV   LCD_ADDR,#005H
            ACALL SET_ADDR_LCD
            MOV   C,DATA.1
            ACALL DOT
            MOV   LCD_ADDR,#042H
            ACALL SET_ADDR_LCD
            MOV   LCD_DATA,TIME1
            ACALL WRCHAR_LCD
            MOV   LCD_DATA,TIME2
            ACALL WRCHAR_LCD
            MOV   A,TIME_SET
            CJNE A,#00H,KLOOP
            MOV   TIME_SET,#00H
            CLR  TRO
            AJMP EXIT_MAIN
KLOOP:  MOV   A,TEMP_READ
        SUBB  A,TEMP_SET
        JZ   HALF
        JC  PULSE_INC

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

PULSE_OFF:      MOV      TOFF,#100
                SJMP      EXIT_MAIN
PULSE_INC:      MOV      A,TEMP_SET
                SUBB      A,TEMP_READ
                MOV      TEMP_DIFF,A
                SUBB      A,#3
                JNC      PULSE_ON
                MOV      A,TEMP_DIFF
                MOV      B,#20
                MUL      AB
                MOV      TON,A
                MOV      A,#100
                SUBB      A,TON
                MOV      TOFF,A
                SJMP      EXIT_MAIN
PULSE_ON:       MOV      TOFF,#1
                SJMP      EXIT_MAIN
HALF:           MOV      TOFF,#90
                SJMP      EXIT_MAIN
SETPLUSEUP:     MOV      A,TOFF
                SUBB      A,#10
                MOV      TOFF,A
EXIT_MAIN:      AJMP      L_READTEMP

```

```

;-----
; CHECK TEMP=.5 OR .0
;-----

```

```

DOT:            JNC      DOT0
                MOV      LCD_DATA,#'5'
                ACALL     WRCHAR_LCD
                RET
DOT0:           MOV      LCD_DATA,#'0'
                ACALL     WRCHAR_LCD
                RET

```

```

;-----
; Interrupt0 Set TEMPERATURE Subroutine
;-----

```

```

SUB_INT0:      PUSH     ACC
                PUSH     PSW
                ACALL     LCD_CLR
                MOV      LCD_ADDR,#000H
                ACALL     SET_ADDR_LCD
                MOV      DPTR,#TITLETEMP
                ACALL     WRLINE_LCD
                ACALL     DELAY_1s
                ACALL     LCD_CLR
                MOV      LCD_ADDR,#000H
                ACALL     SET_ADDR_LCD
                MOV      DPTR,#TEMPWAIT
                ACALL     WRLINE_LCD
                MOV      LCD_ADDR,#041H
                ACALL     SET_ADDR_LCD
                ACALL     LCD_BLINK
                ACALL     GET_KEY
                MOV      A,KDATA
                MOV      DPTR,#ASCIITABLE
                MOVC     A,@A+DPTR
                CJNE     A,#1BH,FANON
FANOFF:        ACALL     LCD_CLR
                MOV      LCD_ADDR,#000H
                ACALL     SET_ADDR_LCD
                MOV      DPTR,#TITLEFANOFF
                ACALL     WRLINE_LCD
                CLR      FAN
                ACALL     DELAY_1s
                AJMP     EXITTEMP
FANON:         CJNE     A,#0DH,NEXTTEMP
                ACALL     LCD_CLR
                MOV      LCD_ADDR,#000H
                ACALL     SET_ADDR_LCD
                MOV      DPTR,#TITLEFANON
                ACALL     WRLINE_LCD
                SETB     FAN
                ACALL     DELAY_1s

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

NEXTTEMP:      AJMP      EXITTEMP
                MOV       TEMP_DATA1,A
                MOV       LCD_DATA,A
                ACALL    WRCHAR_LCD
                ACALL    WAIT_KEY
                ACALL    LCD_BLINK
                ACALL    GET_KEY
                MOV       A,KDATA
                MOV       DPTR,#ASCIITABLE
                MOVC     A,@A+DPTR
                MOV       TEMP_DATA2,A
                MOV       LCD_DATA,A
                ACALL    WRCHAR_LCD
                ACALL    WAIT_KEY
                MOV       A,TEMP_DATA1
                CLR      C
                SUBB    A,#30H
                MOV       B,#10
                MUL     AB
                MOV       TEMP_SET,A
                MOV       A,TEMP_DATA2
                CLR      C
                SUBB    A,#30H
                ADD     A,TEMP_SET
                MOV       TEMP_SET,A
                ACALL    DELAY_10ms
                SUBB    A,TEMP_READ
                JC       OFF_PULSE
                MOV       TOFF,#1
                SETB    FAN
                SETB    TRO
                SJMP    EXITTEMP
OFF_PULSE:     MOV       TOFF,#100
EXITTEMP:      POP      PSW
                POP      ACC
                RET

```

```

;-----
; Interrupt1 Set TIME Subroutine
;-----

```

```

SUB_INT1:     PUSH     ACC
                PUSH     PSW
                MOV      TIME_DATA1,#0
                MOV      TIME_DATA2,#0
                MOV      TIME1,#30H
                MOV      TIME2,#30H
                ACALL    LCD_CLR
                MOV      LCD_ADDR,#000H
                ACALL    SET_ADDR_LCD
                MOV      DPTR,#TITLTIME
                ACALL    WRLINE_LCD
                ACALL    DELAY_1s
                ACALL    LCD_CLR
                MOV      LCD_ADDR,#000H
                ACALL    SET_ADDR_LCD
                MOV      DPTR,#TIMEWAIT
                ACALL    WRLINE_LCD
                MOV      LCD_ADDR,#041H
                ACALL    SET_ADDR_LCD
                ACALL    LCD_BLINK
                ACALL    GET_KEY
                MOV      A,KDATA
                MOV      DPTR,#ASCIITABLE
                MOVC     A,@A+DPTR
                MOV      TIME1,A
                MOV      LCD_DATA,A
                ACALL    WRCHAR_LCD
                CLR      C
                SUBB    A,#30H
                MOV      TIME_DATA1,A
                ACALL    WAIT_KEY
                ACALL    LCD_BLINK
                ACALL    GET_KEY
                MOV      A,KDATA
                MOV      DPTR,#ASCIITABLE
                MOVC     A,@A+DPTR

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

MOV     TIME2,A
MOV     LCD_DATA,A
ACALL   WRCHAR_LCD
CLR     C
SUBB    A,#30H
MOV     TIME_DATA2,A
ACALL   WAIT_KEY
MOV     A,TIME_DATA1
SWAP    A
ORL     A,TIME_DATA2
MOV     TIME_SET,A
SETB    TRO
POP     PSW
POP     ACC
RET

```

```

;-----
; Timer0 Subroutine
;-----

```

```

SUB_TIMER:    PUSH    ACC
              PUSH    PSW
              INC     T200u
              MOV     A,T200u
              CJNE   A,#50,END_TIME
              MOV     T200u,#0
              INC     T10m
PULSE_GEN:    MOV     A,T10m
              CJNE   A,TOFF,TIME_LOOP
              SETB    PULSE_BIT
TIME_LOOP:    CJNE   A,#100,END_TIME
              CLR     PULSE_BIT
              INC     T1S
              MOV     T10m,#0
              MOV     A,T1S
              CJNE   A,#60,END_TIME
              MOV     T1S,#0
              MOV     A,TIME_SET
              DEC     A
              MOV     R4,A
              ANL    A,#0FH
              CJNE   A,#0FH,NEXTT
              MOV     A,R4
              CLR     C
              SUBB   A,#06H
              SJMP   NEXTT2
NEXTT:        MOV     A,R4
NEXTT2:       ADD     A,#0
              DA     A
              MOV     TIME_SET,A
              ANL    A,#0FH
              ADD    A,#30H
              MOV     TIME2,A
              MOV     A,TIME_SET
              SWAP   A
              ANL    A,#0FH
              ADD    A,#30H
              MOV     TIME1,A
END_TIME:    POP     PSW
              POP     ACC
              RET

```

```

;-----
; Scan Key Subroutine
;-----

```

```

GET_KEY:     MOV     P1,#0FFH
              MOV     KDATA,#00H
CHKC0:       CLR     KCOL0
              MOV     A,P1
              ANL    A,#00FH
              CJNE   A,#00FH,CODATA
              AJMP   CHKC1
CODATA:     MOV     KDATA,#00H
              AJMP   GET_ROW
CHKC1:       SETB    KCOL0
              CLR     KCOL1
              MOV     A,P1

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

ANL      A,#00FH
CJNE    A,#00FH,C1DATA
AJMP    CHK2
C1DATA: MOV    KDATA,#01H
AJMP    GET_ROW
CHKC2:  SETB   KCOL1
CLR     KCOL2
MOV     A,P1
ANL     A,#00FH
CJNE    A,#00FH,C2DATA
AJMP    CHK3
C2DATA: MOV    KDATA,#02H
AJMP    GET_ROW
CHKC3:  SETB   KCOL2
CLR     KCOL3
MOV     A,P1
ANL     A,#00FH
CJNE    A,#00FH,C3DATA
AJMP    GET_KEY
C3DATA: MOV    KDATA,#03H
GET_ROW: CLR   KCOL0
CLR     KCOL1
CLR     KCOL2
CLR     KCOL3
CHKR0:  JB     KROW0,CHKR1
RET
CHKR1:  JB     KROW1,CHKR2
MOV     A,KDATA
ADD     A,#04H
MOV     KDATA,A
RET
CHKR2:  JB     KROW2,CHKR3
MOV     A,KDATA
ADD     A,#08H
MOV     KDATA,A
RET
CHKR3:  JB     KROW3,CHKR0
MOV     A,KDATA
ADD     A,#0CH
MOV     KDATA,A
RET
;-----
; Wait keypad depressed
;-----
WAIT_KEY: MOV   A,P1
ANL     A,#00FH
CJNE    A,#00FH,WAIT_KEY
RET
;-----
; LCD Initialize
;-----
INIT_LCD: ACALL  DELAY_100ms
CLR     LCD_RS
MOV     P0,#00111000B
ACALL  LCD_CLK
ACALL  DELAY_10ms
MOV     P0,#00111000B
ACALL  LCD_CLK
ACALL  LCD_OFF
ACALL  LCD_CLR
MOV     P0,#00000110B
ACALL  LCD_CLK
ACALL  LCD_HOME
;-----
; LCD Clear Display
;-----
LCD_CLR: CLR     LCD_RS
MOV     P0,#00000001B
ACALL  LCD_CLK
RET

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

;-----
; LCD Return Home
;-----
LCD_HOME:      CLR      LCD_RS
               MOV      PO,#00000010B
               ACALL    LCD_CLK
               RET

;-----
; LCD Display Off
;-----
LCD_OFF:       CLR      LCD_RS
               MOV      PO,#00001000B
               ACALL    LCD_CLK
               RET

;-----
; LCD Clk
;-----
LCD_CLK:       SETB     LCD_EN
               ACALL    LCD_DELAY
               CLR      LCD_EN
               ACALL    LCD_DELAY
               RET

;-----
; LCD Display On
;-----
LCD_ON:        CLR      LCD_RS
               MOV      PO,#00001100B
               ACALL    LCD_CLK
               RET

;-----
; LCD Cursor On
;-----
LCD_BLINK:     CLR      LCD_RS
               MOV      PO,#00001111B
               ACALL    LCD_CLK
               RET

;-----
; Set LCD Address
;-----
SET_ADDR_LCD:  CLR      LCD_RS
               MOV      A,LCD_ADDR
               SETB     ACC.7
               MOV      PO,A
               ACALL    LCD_CLK
               RET

;-----
; Write Character to show LCD
;-----
WRCHAR_LCD:   SETB     LCD_RS
               MOV      PO,LCD_DATA
               ACALL    LCD_CLK
               ACALL    LCD_ON
               RET

;-----
; Write Line of 16 Character from ROM
;-----
WRLINE_LCD:   MOV      R0,#0
WRLINE_LCD_1: SETB     LCD_RS
               CLR      A
               MOVC    A,@A+DPTR
               MOV      PO,A
               ACALL    LCD_CLK
               INC      DPTR
               INC      R0
               CJNE    R0,#8,WRLINE_LCD_1
               MOV      LCD_ADDR,#040H
               ACALL    SET_ADDR_LCD
WRLINE_LCD_2: SETB     LCD_RS
               CLR      A

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

MOV      A,@A+DPTR
MOV      P0,A
ACALL   LCD_CLK
INC      DPTR
INC      R0
CJNE    R0,#16,WRLINE_LCD_2
ACALL   LCD_ON
RET

```

```

;-----
;Writes the value in A to the DS1620
;-----

```

```

WRITE1620:   CLR      C
MOV        R0, #08H
NEXTBITWRITE: CLR      CLK
RRC        A
MOV        DQ, C
SETB       CLK
DJNZ      R0, NEXTBITWRITE
RET

```

```

;-----
;Reads the value from the DS1620 and puts it in A
;-----

```

```

READ1620:    CLR      C
MOV        R0, #08H
NEXTBITREAD: SETB     DQ
CLR        CLK
MOV        C, DQ
SETB       CLK
RRC        A
DJNZ      R0, NEXTBITREAD
CLR        DQ
RET

```

```

;-----
;Routine to Configure DS1620
;-----

```

```

CONFIGURE:   SETB     RST
MOV         A, #0CH
ACALL      WRITE1620
MOV        A, #00001011B
ACALL      WRITE1620
CLR        RST
RET

```

```

;-----
;Start Temperature Conversion on 1620
;-----

```

```

START_CONVERT: SETB     RST
MOV         A, #0EEH
ACALL      WRITE1620
CLR        RST
RET

```

```

;-----
;Read Temperature from 1620
;-----

```

```

READ_TEMPERATURE: SETB     RST
MOV         A, #0AAH
ACALL      WRITE1620
ACALL      READ1620
MOV        R1, A
ACALL      READ1620
MOV        R2, A
CLR        RST
RET

```

```

;-----
;Read Config. from 1620
;-----

```

```

READ_CONFIG: SETB     RST
MOV         A, #0ACH
ACALL      WRITE1620
ACALL      READ1620

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

                CLR        RST
                RET

;-----
;Delay 1ms
;-----
DELAYMS:      MOV        R7,#00H
LOOPA:        INC        R7
                MOV        A,R7
                CJNE       A,#0FFH,LOOPA
                RET

;-----
;Delay time LCD_DELAY, 10ms, 100m, 1s
;-----
LCD_DELAY:    MOV        R7,#002
LCD_DELAY_1:  MOV        R6,#0E6H
LCD_DELAY_2:  NOP
                NOP
                DJNZ       R6,LCD_DELAY_2
                DJNZ       R7,LCD_DELAY_1
                RET

DELAY_10ms:   MOV        R7,#010
DELAY_10ms_1: MOV        R6,#0E6H
DELAY_10ms_2: NOP
                NOP
                DJNZ       R6,DELAY_10ms_2
                DJNZ       R7,DELAY_10ms_1
                RET

DELAY_100ms:  MOV        R7,#100
DL100ms_1:   MOV        R6,#0E6H
DL100ms_2:   NOP
                NOP
                DJNZ       R6,DL100ms_2
                DJNZ       R7,DL100ms_1
                RET

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

;-----
;Define Constant < Store in Flash EEPROM Program Memory >
;-----
;
; 0123456789ABCDEF
TITLE1:       DB        ' Reading Temp. '
TITLE2:       DB        ' Please wait  '

DISPLAYSHOW:  DB        ' . C MIN'
TITLETEMP:    DB        'SET TEMPERATURE '
TEMPWAIT:     DB        ' TEMP. : C '
TITLETIME:    DB        ' SET TIME '
TIMEWAIT:     DB        ' TIME. : MIN'
TITLEFANON:   DB        ' FAN ON '
TITLEFANOFF:  DB        ' FAN OFF '

ASCIITABLE:   DB        '1','2','3','A'
                DB        '4','5','6','B'
                DB        '7','8','9','0DH'
                DB        '*','0','#','1BH'

END

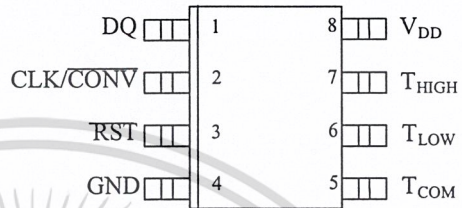
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

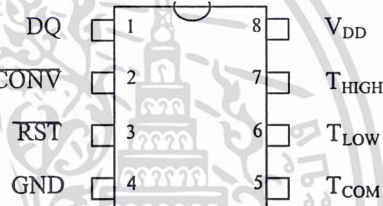
FEATURES

- Requires no external components
- Supply voltage range covers from 2.7V to 5.5V
- Measures temperatures from -55°C to +125°C in 0.5°C increments; Fahrenheit equivalent is -67°F to +257°F in 0.9°F increments
- Temperature is read as a 9-bit value
- Converts temperature to digital word in 1 second (max)
- Thermostatic settings are user-definable and nonvolatile
- Data is read from/written via a 3-wire serial interface (CLK, DQ, $\overline{\text{RST}}$)
- Applications include thermostatic controls, industrial systems, consumer products, thermometers, or any thermally sensitive system
- 8-pin DIP or SOIC (208-mil) packages

PIN ASSIGNMENT



DS1620S 8-Pin SOIC (208-mil)
See Mech Drawings Section



DS1620 8-Pin DIP (300-mil)
See Mech Drawings Section

PIN DESCRIPTION

- DQ - 3-Wire Input/Output
- CLK/CONV - 3-Wire Clock Input and Stand-alone Convert Input
- $\overline{\text{RST}}$ - 3-Wire Reset Input
- GND - Ground
- T_{HIGH} - High Temperature Trigger
- T_{LOW} - Low Temperature Trigger
- T_{COM} - High/Low Combination Trigger
- V_{DD} - Power Supply Voltage (3V - 5V)

DESCRIPTION

The DS1620 Digital Thermometer and Thermostat provides 9-bit temperature readings which indicate the temperature of the device. With three thermal alarm outputs, the DS1620 can also act as a thermostat. T_{HIGH} is driven high if the DS1620's temperature is greater than or equal to a user-defined temperature TH. T_{LOW} is driven high if the DS1620's temperature is less than or equal to a user-defined temperature TL. T_{COM} is driven high when the temperature exceeds TH and stays high until the temperature falls below that of TL.

User-defined temperature settings are stored in nonvolatile memory, so parts can be programmed prior to insertion in a system, as well as used in standalone applications without a CPU. Temperature settings and temperature readings are all communicated to/from the DS1620 over a simple 3-wire interface.

OPERATION-MEASURING TEMPERATURE

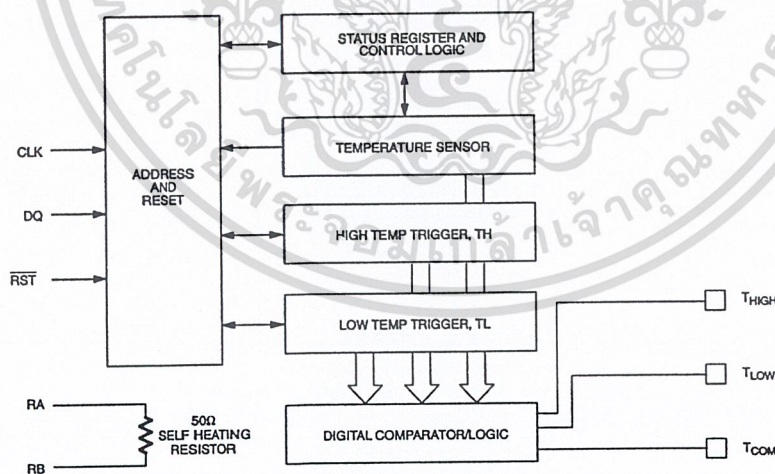
A block diagram of the DS1620 is shown in Figure 1. The DS1620 measures temperatures through the use of an onboard proprietary temperature measurement technique. A block diagram of the temperature measurement circuitry is shown in Figure 2.

The DS1620 measures temperature by counting the number of clock cycles that an oscillator with a low temperature coefficient goes through during a gate period determined by a high temperature coefficient oscillator. The counter is preset with a base count that corresponds to -55°C . If the counter reaches 0 before the gate period is over, the temperature register, which is also preset to the -55°C value, is incremented, indicating that the temperature is higher than -55°C .

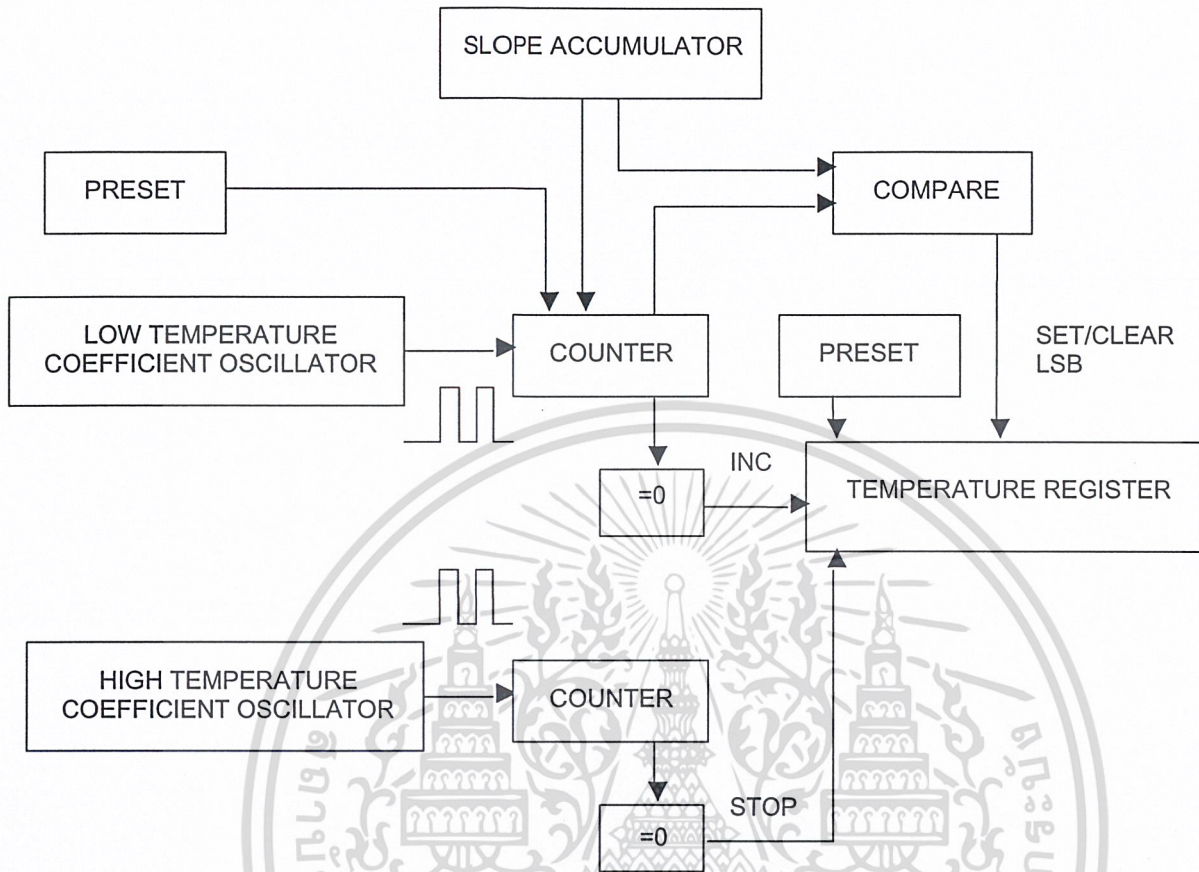
At the same time, the counter is then preset with a value determined by the slope accumulator circuitry. This circuitry is needed to compensate for the parabolic behavior of the oscillators over temperature. The counter is then clocked again until it reaches 0. If the gate period is still not finished, then this process repeats.

The slope accumulator is used to compensate for the nonlinear behavior of the oscillators over temperature, yielding a high-resolution temperature measurement. This is done by changing the number of counts necessary for the counter to go through for each incremental degree in temperature. To obtain the desired resolution, therefore, both the value of the counter and the number of counts per degree C (the value of the slope accumulator) at a given temperature must be known.

DS1620 FUNCTIONAL BLOCK DIAGRAM Figure 1



TEMPERATURE MEASURING CIRCUITRY Figure 2



This calculation is done inside the DS1620 to provide 0.5°C resolution. The temperature reading is provided in a 9-bit, two's complement reading by issuing a READ TEMPERATURE command. Table 1 describes the exact relationship of output data to measured temperature. The data is transmitted serially through the 3-wire serial interface, LSB first. The DS1620 can measure temperature over the range of -55°C to +125°C in 0.5°C increments. For Fahrenheit usage, a lookup table or conversion factor must be used.

TEMPERATURE/DATA RELATIONSHIPS Table 1

TEMP	DIGITAL OUTPUT (Binary)	DIGITAL OUTPUT (Hex)
+125°C	0 11111010	00FA
+25°C	0 00110010	0032h
+½°C	0 00000001	0001h
+0°C	0 00000000	0000h
-½°C	1 11111111	01FFh
-25°C	1 11001110	01CEh
-55°C	1 10010010	0192h

Since data is transmitted over the 3-wire bus LSB first, temperature data can be written to/read from the DS1620 as either a 9-bit word (taking $\overline{\text{RST}}$ low after the 9th (MSB) bit), or as two transfers of 8-bit

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

words, with the most significant 7 bits being ignored or set to 0, as illustrated in Table 1. After the MSB, the DS1620 will output 0s.

Note that temperature is represented in the DS1620 in terms of a $\frac{1}{2}^{\circ}\text{C}$ LSB, yielding the following 9-bit format:

MSB								LSB								
X	X	X	X	X	X	X	X	1	1	1	0	0	1	1	1	0

T = -25°C

Higher resolutions may be obtained by reading the temperature, and truncating the 0.5°C bit (the LSB) from the read value. This value is TEMP_READ. The value left in the counter may then be read by issuing a READ COUNTER command. This value is the count remaining (COUNT_REMAIN) after the gate period has ceased. By loading the value of the slope accumulator into the count register (using the READ SLOPE command), this value may then be read, yielding the number of counts per degree C (COUNT_PER_C) at that temperature. The actual temperature may then be calculated by the user using the following:

$$\text{TEMPERATURE} = \text{TEMP_READ} - 0.25 + \frac{(\text{COUNT_PER_C} - \text{COUNT_REMAIN})}{\text{COUNT_PER_C}}$$

DETAILED PIN DESCRIPTION Table 2

PIN	SYMBOL	DESCRIPTION
1	DQ	Data Input/Output pin for 3-wire communication port.
2	CLK/ <u>CONV</u>	Clock input pin for 3-wire communication port. When the DS1620 is used in a stand-alone application with no 3-wire port, this pin can be used as a convert pin. Temperature conversion will begin on the falling edge of <u>CONV</u> .
3	<u>RST</u>	Reset input pin for 3-wire communication port.
4	GND	Ground pin.
5	T _{COM}	High/Low Combination Trigger. Goes high when temperature exceeds TH; will reset to low when temperature falls below TL.
6	T _{LOW}	Low Temperature Trigger. Goes high when temperature falls below TL.
7	T _{HIGH}	High Temperature Trigger. Goes high when temperature exceeds TH.
8	V _{DD}	Supply Voltage. 2.7V – 5.5V input power pin.

OPERATION–THERMOSTAT CONTROLS

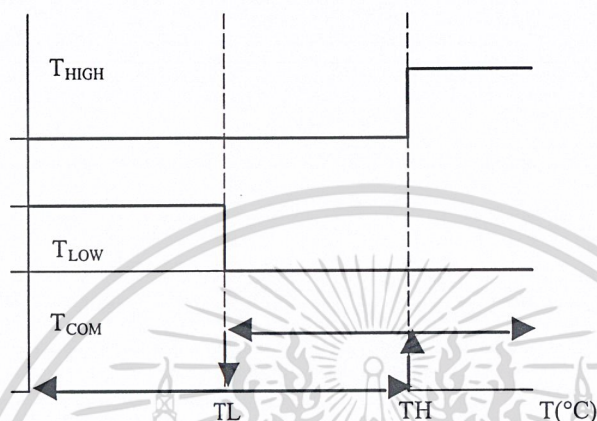
Three thermally triggered outputs, T_{HIGH}, T_{LOW}, and T_{COM}, are provided to allow the DS1620 to be used as a thermostat, as shown in Figure 3. When the DS1620's temperature meets or exceeds the value stored in the high temperature trip register, the output T_{HIGH} becomes active (high) and remains active until the DS1620's measured temperature becomes less than the stored value in the high temperature register, TH. The T_{HIGH} output can be used to indicate that a high temperature tolerance boundary has been met or exceeded, or it can be used as part of a closed loop system to activate a cooling system and deactivate it when the system temperature returns to tolerance.

The T_{LOW} output functions similarly to the T_{HIGH} output. When the DS1620's measured temperature equals or falls below the value stored in the low temperature register, the T_{LOW} output becomes active. T_{LOW} remains active until the DS1620's temperature becomes greater than the value stored in the low temperature register, TL. The T_{LOW} output can be used to indicate that a low temperature tolerance

boundary has been met or exceeded, or as part of a closed loop system it can be used to activate a heating system and deactivate it when the system temperature returns to tolerance.

The T_{COM} output goes high when the measured temperature meets or exceeds T_H , and will stay high until the temperature equals or falls below T_L . In this way, any amount of hysteresis can be obtained.

THERMOSTAT OUTPUT OPERATION Figure 3



OPERATION AND CONTROL

The DS1620 must have temperature settings resident in the T_H and T_L registers for thermostatic operation. A configuration/status register also determines the method of operation that the DS1620 will use in a particular application and indicates the status of the temperature conversion operation. The configuration register is defined as follows:

CONFIGURATION/STATUS REGISTER

DONE	THF	TLF	NVB	1	0	CPU	1SHOT
------	-----	-----	-----	---	---	-----	-------

where

DONE = Conversion Done Bit. 1=conversion complete, 0=conversion in progress.

THF = Temperature High Flag. This bit will be set to 1 when the temperature is greater than or equal to the value of T_H . It will remain 1 until reset by writing 0 into this location or by removing power from the device. This feature provides a method of determining if the DS1620 has ever been subjected to temperatures above T_H while power has been applied.

TLF = Temperature Low Flag. This bit will be set to 1 when the temperature is less than or equal to the value of T_L . It will remain 1 until reset by writing 0 into this location or by removing power from the device. This feature provides a method of determining if the DS1620 has ever been subjected to temperatures below T_L while power has been applied.

NVB = Nonvolatile Memory Busy Flag. 1=write to an E^2 memory cell in progress. 0=nonvolatile memory is not busy. A copy to E^2 may take up to 10 ms.

CPU = CPU Use Bit. If $CPU=0$, the $\overline{CLK/CONV}$ pin acts as a conversion start control, when \overline{RST} is low. If CPU is 1, the DS1620 will be used with a CPU communicating to it over the 3-wire port, and the

operation of the CLK/CONV pin is as a normal clock in concert with DQ and RST. This bit is stored in nonvolatile E² memory, capable of at least 50,000 writes. The DS1620 is shipped with CPU=0.

1SHOT = One-Shot Mode. If 1SHOT is 1, the DS1620 will perform one temperature conversion upon reception of the Start Convert T protocol. If 1SHOT is 0, the DS1620 will continuously perform temperature conversion. This bit is stored in nonvolatile E² memory, capable of at least 50,000 writes. The DS1620 is shipped with 1SHOT=0.

For typical thermostat operation, the DS1620 will operate in continuous mode. However, for applications where only one reading is needed at certain times or to conserve power, the one-shot mode may be used. Note that the thermostat outputs (T_{HIGH}, T_{LOW}, T_{COM}) will remain in the state they were in after the last valid temperature conversion cycle when operating in one-shot mode.

OPERATION IN STAND-ALONE MODE

In applications where the DS1620 is used as a simple thermostat, no CPU is required. Since the temperature limits are nonvolatile, the DS1620 can be programmed prior to insertion in the system. In order to facilitate operation without a CPU, the CLK/CONV pin (pin 2) can be used to initiate conversions. Note that the CPU bit must be set to 0 in the configuration register to use this mode of operation. Whether CPU=0 or 1, the 3-wire port is active. Setting CPU=1 disables the stand-alone mode.

To use the CLK/CONV pin to initiate conversions, RST must be low and CLK/CONV must be high. If CLK/CONV is driven low and then brought high in less than 10 ms, one temperature conversion will be performed and then the DS1620 will return to an idle state. If CLK/CONV is driven low and remains low, continuous conversions will take place until CLK/CONV is brought high again. With the CPU bit set to 0, the CLK/CONV will override the 1SHOT bit if it is equal to 1. This means that even if the part is set for one-shot mode, driving CLK/CONV low will initiate conversions.

3-WIRE COMMUNICATIONS

The 3-wire bus is comprised of three signals. These are the RST (reset) signal, the CLK (clock) signal, and the DQ (data) signal. All data transfers are initiated by driving the RST input high. Driving the RST input low terminates communication. (See Figures 4 and 5.) A clock cycle is a sequence of a falling edge followed by a rising edge. For data inputs, the data must be valid during the rising edge of a clock cycle. Data bits are output on the falling edge of the clock and remain valid through the rising edge.

When reading data from the DS1620, the DQ pin goes to a high-impedance state while the clock is high. Taking RST low will terminate any communication and cause the DQ pin to go to a high-impedance state.

Data over the 3-wire interface is communicated LSB first. The command set for the 3-wire interface as shown in Table 3 is as follows.

Read Temperature [AAh]

This command reads the contents of the register which contains the last temperature conversion result. The next nine clock cycles will output the contents of this register.

Write TH [01h]

This command writes to the TH (HIGH TEMPERATURE) register. After issuing this command the next nine clock cycles clock in the 9-bit temperature limit which will set the threshold for operation of the T_{HIGH} output.

Write TL [02h]

This command writes to the TL (LOW TEMPERATURE) register. After issuing this command the next nine clock cycles clock in the 9-bit temperature limit which will set the threshold for operation of the T_{LOW} output.

Read TH [A1h]

This command reads the value of the TH (HIGH TEMPERATURE) register. After issuing this command the next nine clock cycles clock out the 9-bit temperature limit which sets the threshold for operation of the T_{HIGH} output.

Read TL [A2h]

This command reads the value of the TL (LOW TEMPERATURE) register. After issuing this command the next nine clock cycles clock out the 9-bit temperature limit which sets the threshold for operation of the T_{LOW} output.

Read Counter [A0h]

This command reads the value of the counter byte. The next nine clock cycles will output the contents of this register.

Read Slope [A9h]

This command reads the value of the slope counter byte from the DS1620. The next nine clock cycles will output the contents of this register.

Start Convert T [EEh]

This command begins a temperature conversion. No further data is required. In one-shot mode the temperature conversion will be performed and then the DS1620 will remain idle. In continuous mode this command will initiate continuous conversions.

Stop Convert T [22h]

This command stops temperature conversion. No further data is required. This command may be used to halt a DS1620 in continuous conversion mode. After issuing this command the current temperature measurement will be completed and then the DS1620 will remain idle until a Start Convert T is issued to resume continuous operation.

Write Config [0Ch]

This command writes to the configuration register. After issuing this command the next eight clock cycles clock in the value of the configuration register.

Read Config [ACh]

This command reads the value in the configuration register. After issuing this command the next eight clock cycles output the value of the configuration register.

DS1620 COMMAND SET Table 3

INSTRUCTION	DESCRIPTION	PROTOCOL	3-WIRE BUS DATA AFTER ISSUING PROTOCOL	NOTES
TEMPERATURE CONVERSION COMMANDS				
Read Temperature	Reads last converted temperature value from temperature register.	AAh	<read data>	
Read Counter	Reads value of count remaining from counter.	A0h	<read data>	
Read Slope	Reads value of the slope accumulator.	A9h	<read data>	
Start Convert T	Initiates temperature conversion.	EEh	Idle	1
Stop Convert T	Halts temperature conversion.	22h	Idle	1
THERMOSTAT COMMANDS				
Write TH	Writes high temperature limit value into TH register.	01h	<write data>	2
Write TL	Writes low temperature limit value into TL register.	02h	<write data>	2
Read TH	Reads stored value of high temperature limit from TH register.	A1h	<read data>	2
Read TL	Reads stored value of low temperature limit from TL register.	A2h	<read data>	2
Write Config	Writes configuration data to configuration register.	0Ch	<write data>	2
Read Config	Reads configuration data from configuration register.	ACh	<read data>	2

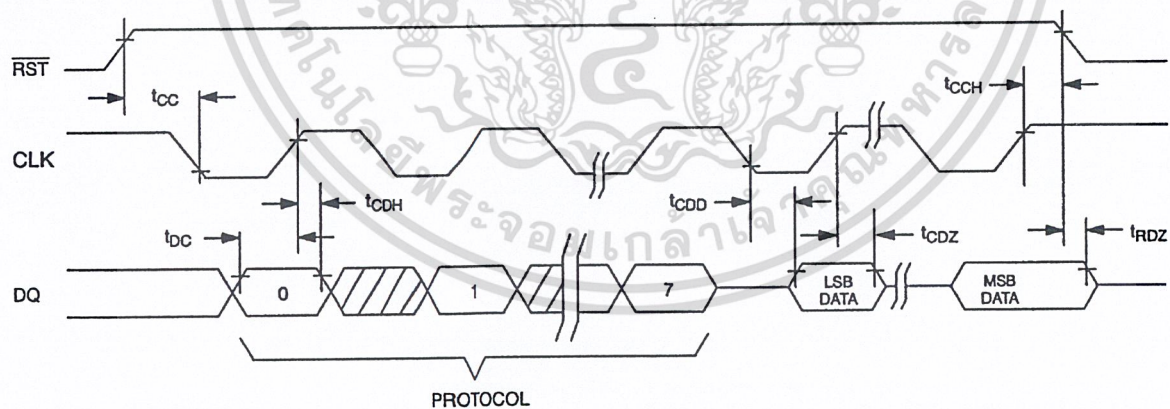
NOTES:

1. In continuous conversion mode, a Stop Convert T command will halt continuous conversion. To restart, the Start Convert T command must be issued. In one-shot mode, a Start Convert T command must be issued for every temperature reading desired.
2. Writing to the E² typically requires 10 ms at room temperature. After issuing a write command no further writes should be requested for at least 10 ms.

FUNCTION EXAMPLE

Example: CPU sets up DS1620 for continuous conversion and thermostatic function.

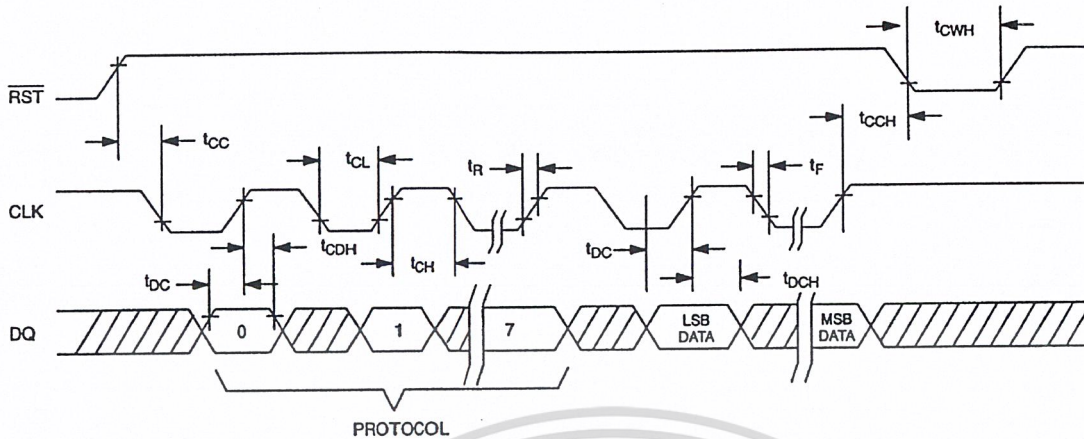
CPU MODE	DS1620 MODE (3-WIRE)	DATA (LSB FIRST)	COMMENTS
TX	RX	0Ch	CPU issues Write Config command
TX	RX	00h	CPU sets DS1620 up for continuous conversion
TX	RX	Toggle $\overline{\text{RST}}$	CPU issues Reset to DS1620
TX	RX	01h	CPU issues Write TH command
TX	RX	0050h	CPU sends data for TH limit of +40°C
TX	RX	Toggle $\overline{\text{RST}}$	CPU issues Reset to DS1620
TX	RX	02h	CPU issues Write TL command
TX	RX	0014h	CPU sends data for TL limit of +10°C
TX	RX	Toggle $\overline{\text{RST}}$	CPU issues Reset to DS1620
TX	RX	A1h	CPU issues Read TH command
RX	TX	0050h	DS1620 sends back stored value of TH for CPU to verify
TX	RX	Toggle $\overline{\text{RST}}$	CPU issues Reset to DS1620
TX	RX	A2h	CPU issues Read TL command
RX	TX	0014h	DS1620 sends back stored value of TL for CPU to verify
TX	RX	Toggle $\overline{\text{RST}}$	CPU issues Reset to DS1620
TX	RX	EEh	CPU issues Start Convert T command
TX	RX	Drop $\overline{\text{RST}}$	CPU issues Reset to DS1620

READ DATA TRANSFER Figure 4

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

WRITE DATA TRANSFER Figure 5



NOTE: t_{CL} , t_{CH} , t_R , and t_F apply to both read and write data transfer.

RELATED APPLICATION NOTES

The following Application Notes can be applied to the DS1620. These notes can be obtained from the Dallas Semiconductor "Application Note Book", via our Web site at <http://www.dalsemi.com/>, or through our faxback service at (972) 371-4441.

Application Note 67: "Applying and Using the DS1620 in Temperature Control Applications"

Application Note 85: "Interfacing the DS1620 to the Motorola SPI Bus"

Application Note 105: "High Resolution Temperature Measurement with Dallas Direct-to-Digital Temperature Sensors"

Sample DS1620 subroutines that can be used in conjunction with AN105 can be downloaded from the web-site or our Anonymous FTP Site.

ABSOLUTE MAXIMUM RATINGS*

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

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

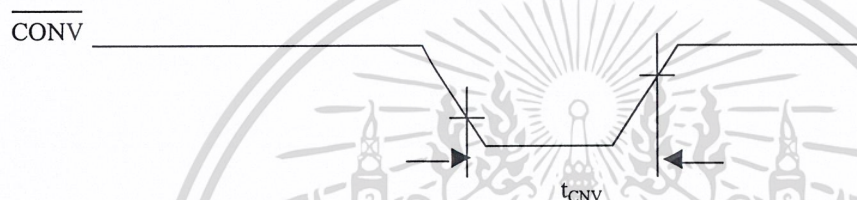
RECOMMENDED DC OPERATING CONDITIONS

PARAMETER	SYMBOL	MIN	TYP	MAX	UNITS	NOTES
Supply	V_{DD}	2.7		5.5	V	1
Logic 1	V_{IH}	2.0		$V_{CC}+0.3$	V	1
Logic 0	V_{IL}	-0.3		+0.6	V	1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงแก้ไขหรือทำซ้ำโดยไม่ได้รับอนุญาตจากเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

DC ELECTRICAL CHARACTERISTICS (-55°C to +125°C; $V_{DD}=2.7V$ to 5.5V)

PARAMETER	SYMBOL	CONDITION	MIN	MAX	UNITS	NOTES
Thermometer Error	T_{ERR}	0°C to +70°C -55°C to +0°C and 70°C to 125°C		$\pm\frac{1}{2}$	°C	10, 11
				See Typical Curve		
Logic 0 Output	V_{OL}			0.4	V	3
Logic 1 Output	V_{OH}		2.4		V	2
Input Resistance	R_I	\overline{RST} to GND DQ, CLK to V_{DD}	1 1		M Ω M Ω	
Active Supply Current	I_{CC}	0°C to +70°C		1	mA	4, 5
Standby Supply Current	I_{STBY}	0°C to +70°C		1	μA	4, 5

SINGLE CONVERT TIMING DIAGRAM (STAND-ALONE MODE)**AC ELECTRICAL CHARACTERISTICS** (-55°C to +125°C; $V_{DD}=2.7V$ to 5.5V)

PARAMETERS	SYMBOL	MIN	TYP	MAX	UNITS	NOTES
Temperature Conversion Time	T_{TC}		400	1000	ms	
Data to CLK Setup	t_{DC}	35			ns	6
CLK to Data Hold	t_{CDH}	40			ns	6
CLK to Data Delay	t_{CDD}			100	ns	6, 7, 8
CLK Low Time	t_{CL}	285			ns	6
CLK High Time	t_{CH}	285			ns	6
CLK Frequency	f_{CLK}	DC		1.75	MHz	6
CLK Rise and Fall	t_R, t_F			500	ns	
\overline{RST} to CLK Setup	t_{CC}	100			ns	6
CLK to \overline{RST} Hold	t_{CCH}	40			ns	6
\overline{RST} Inactive Time	t_{CWH}	125			ns	6, 9
CLK High to I/O High-Z	t_{CDZ}			50	ns	6
\overline{RST} Low to I/O High-Z	t_{RDZ}			50	ns	6
Convert Pulse Width	t_{CNV}	250 ns		500 ms		
NV Write Cycle Time	t_{WR}		10	50	ms	12

AC ELECTRICAL CHARACTERISTICS (-55°C to +125°C; $V_{DD}=2.7V$ to 5.5V)

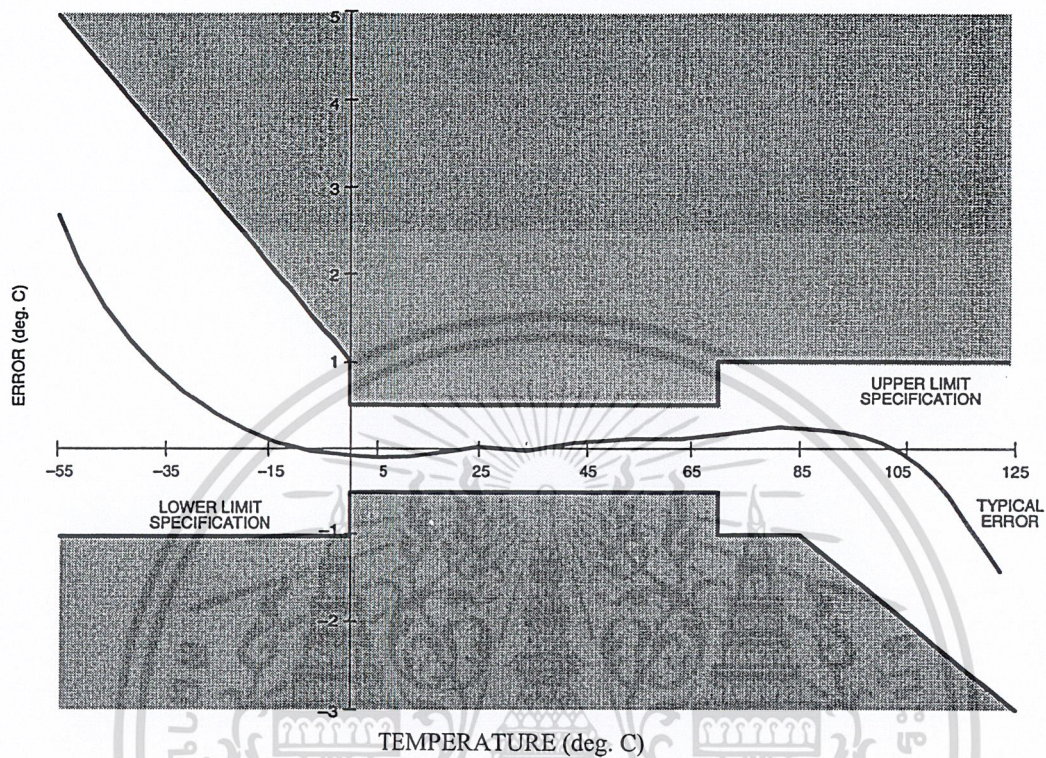
PARAMETER	SYMBOL	MIN	TYP	MAX	UNITS	NOTES
Input Capacitance	C_I		5		pF	
I/O Capacitance	$C_{I/O}$		10		pF	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

NOTES:

1. All voltages are referenced to ground.
2. Logic 1 voltages are specified at a source current of 1 mA.
3. Logic 0 voltages are specified at a sink current of 4 mA.
4. I_{CC} specified with DQ pin open and CLK pin at V_{DD} .
5. I_{CC} specified with V_{CC} at 5.0V and $\overline{RST} = GND$.
6. Measured at $V_{IH} = 2.0V$ or $V_{IL} = 0.8V$.
7. Measured at $V_{OH} = 2.4V$ or $V_{OL} = 0.4V$.
8. Load capacitance = 50 pF.
9. t_{CWH} must be 10 ms minimum following any write command that involves the E^2 memory.
10. See typical curve for specification limits outside $0^\circ C$ to $70^\circ C$ range.
11. Thermometer error reflects temperature accuracy as tested during calibration.
12. Writing to the nonvolatile memory should only take place in the $0^\circ C$ to $70^\circ C$ temperature range.
13. Valid for design revisions D1 and above. The supply range for Rev. C2 and below is $4.5V \leq 5.5V$.

TYPICAL PERFORMANCE CURVE

DS1620 DIGITAL THERMOMETER AND THERMOSTAT
TEMPERATURE READING ERROR

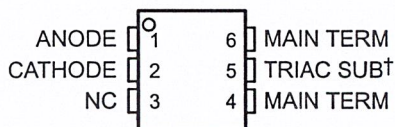
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

MOC3020 THRU MOC3023 OPTOCOUPLED/OPTOISOLATORS

SOES025A – OCTOBER 1986 – REVISED APRIL 1998

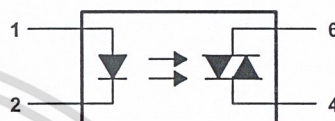
- 400 V Phototriac Driver Output
- Gallium-Arsenide-Diode Infrared Source and Optically-Coupled Silicon Triac Driver (Bilateral Switch)
- UL Recognized . . . File Number E65085
- High Isolation . . . 7500 V Peak
- Output Driver Designed for 220 Vac
- Standard 6-Terminal Plastic DIP
- Directly Interchangeable with Motorola MOC3020, MOC3021, MOC3022, and MOC3023

MOC3020 – MOC3023 . . . PACKAGE
(TOP VIEW)



† Do not connect this terminal
NC – No internal connection

logic diagram



typical 115/240 Vac(rms) applications

- Solenoid/Valve Controls
- Lamp Ballasts
- Interfacing Microprocessors to 115/240 Vac Peripherals
- Motor Controls
- Incandescent Lamp Dimmers

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

Input-to-output peak voltage, 5 s maximum duration, 60 Hz (see Note 1)	7.5 kV
Input diode reverse voltage	3 V
Input diode forward current, continuous	50 mA
Output repetitive peak off-state voltage	400 V
Output on-state current, total rms value (50-60 Hz, full sine wave): $T_A = 25^\circ\text{C}$	100 mA
$T_A = 70^\circ\text{C}$	50 mA
Output driver nonrepetitive peak on-state current ($t_W = 10$ ms, duty cycle = 10%, see Figure 7)	1.2 A
Continuous power dissipation at (or below) 25°C free-air temperature:	
Infrared-emitting diode (see Note 2)	100 mW
Phototriac (see Note 3)	300 mW
Total device (see Note 4)	330 mW
Operating junction temperature range, T_J	-40°C to 100°C
Storage temperature range, T_{stg}	-40°C to 150°C
Lead temperature 1,6 (1/16 inch) from case for 10 seconds	260°C

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

- NOTES:
1. Input-to-output peak voltage is the internal device dielectric breakdown rating.
 2. Derate linearly to 100°C free-air temperature at the rate of 1.33 mW/°C.
 3. Derate linearly to 100°C free-air temperature at the rate of 4 mW/°C.
 4. Derate linearly to 100°C free-air temperature at the rate of 4.4 mW/°C.

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

Copyright © 1998, Texas Instruments Incorporated



TEXAS
INSTRUMENTS

POST OFFICE BOX 655303 • DALLAS, TEXAS 75265

การใช้ผลิตภัณฑ์ Texas Instruments นั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปะหรือลอกเลียนแบบข้อมูลใดๆ ของเอกสารทุกครั้งที่มีการนำไปใช้

MOC3020 THRU MOC3023 OPTOCOUPLEDERS/OPTOISOLATORS

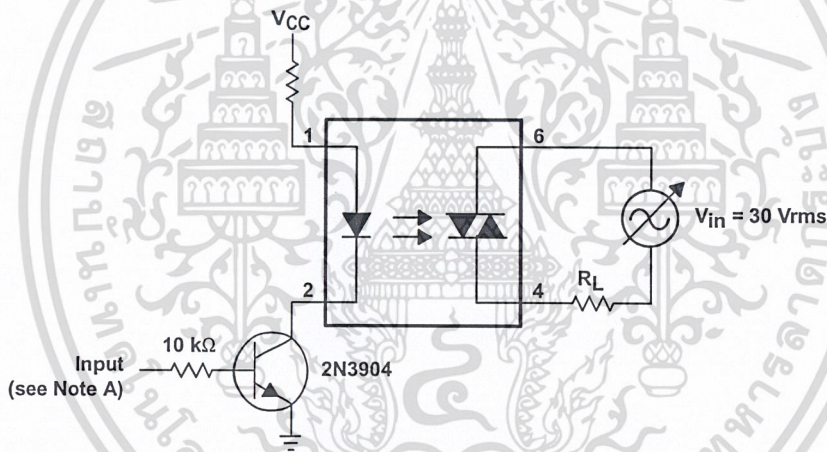
SOES025A – OCTOBER 1986 – REVISED APRIL 1998

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

PARAMETER		TEST CONDITIONS	MIN	TYP	MAX	UNIT	
I_R	Static reverse current	$V_R = 3\text{ V}$		0.05	100	μA	
V_F	Static forward voltage	$I_F = 10\text{ mA}$		1.2	1.5	V	
$I_{(DRM)}$	Repetitive off-state current, either direction	$V_{(DRM)} = 400\text{ V}$, See Note 5		10	100	nA	
dv/dt	Critical rate of rise of off-state voltage	See Figure 1		100		V/ μs	
$dv/dt(c)$	Critical rate of rise of commutating voltage	$I_O = 15\text{ mA}$, See Figure 1		0.15		V/ μs	
I_{FT}	Input trigger current, either direction	Output supply voltage = 3 V	MOC3020		15	30	mA
			MOC3021		8	15	
			MOC3022		5	10	
			MOC3023		3	5	
V_{TM}	Peak on-state voltage, either direction	$I_{TM} = 100\text{ mA}$		1.4	3	V	
I_H	Holding current, either direction			100		μA	

NOTE 5: Test voltage must be applied at a rate no higher than 12 V/ μs .

PARAMETER MEASUREMENT INFORMATION



NOTE A. The critical rate of rise of off-state voltage, dv/dt , is measured with the input at 0 V. The frequency of V_{in} is increased until the phototriac turns on. This frequency is then used to calculate the dv/dt according to the formula:

$$dv/dt = 2\sqrt{2} \pi f V_{in}$$

The critical rate of rise of commutating voltage, $dv/dt(c)$, is measured by applying occasional 5-V pulses to the input and increasing the frequency of V_{in} until the phototriac stays on (latches) after the input pulse has ceased. With no further input pulses, the frequency of V_{in} is then gradually decreased until the phototriac turns off. The frequency at which turn-off occurs may then be used to calculate the $dv/dt(c)$ according to the formula shown above.

Figure 1. Critical Rate of Rise Test Circuit

TYPICAL CHARACTERISTICS

EMITTING-DIODE TRIGGER CURRENT (NORMALIZED)

vs
FREE-AIR TEMPERATURE

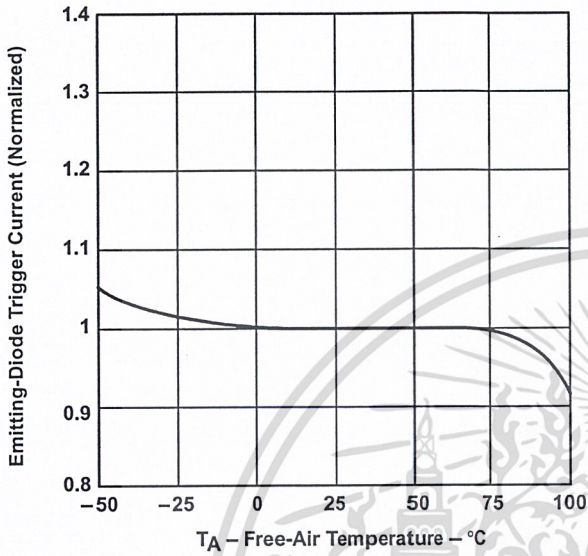


Figure 2

ON-STATE CHARACTERISTICS

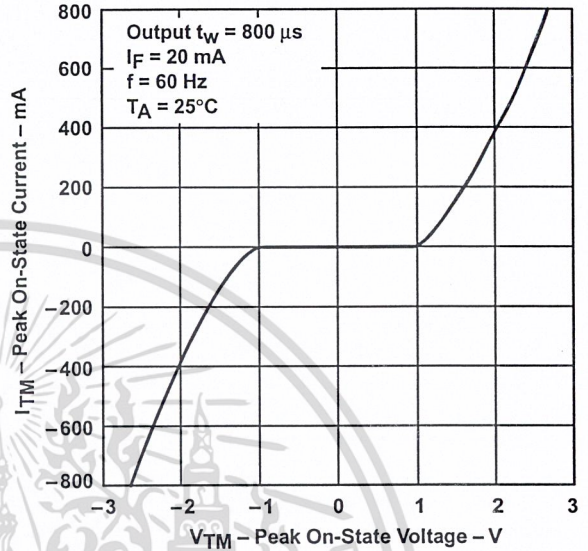


Figure 3

NONREPETITIVE PEAK ON-STATE CURRENT
vs
PULSE DURATION

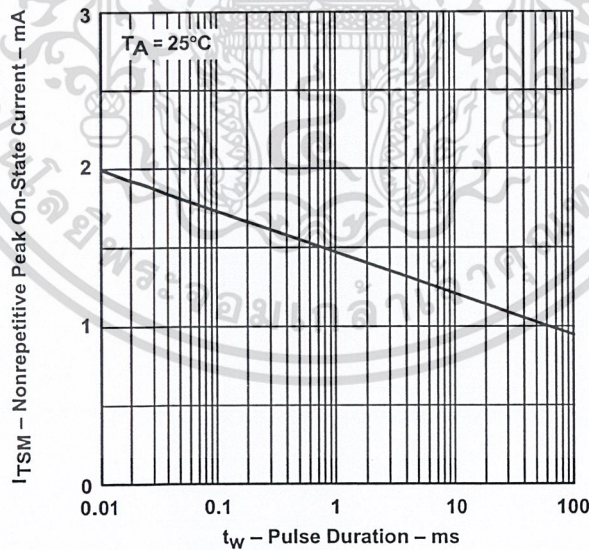


Figure 4

MOC3020 THRU MOC3023 OPTOCOUPPLERS/OPTOISOLATORS

SOES025A – OCTOBER 1986 – REVISED APRIL 1998

APPLICATIONS INFORMATION

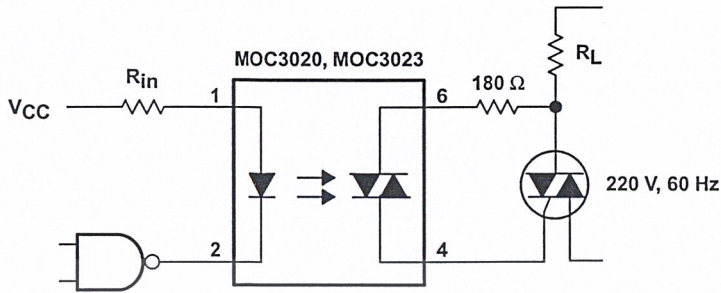


Figure 5. Resistive Load

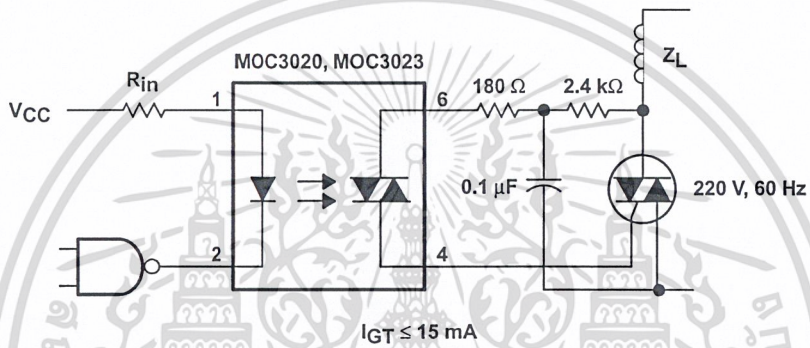


Figure 6. Inductive Load With Sensitive-Gate Triac

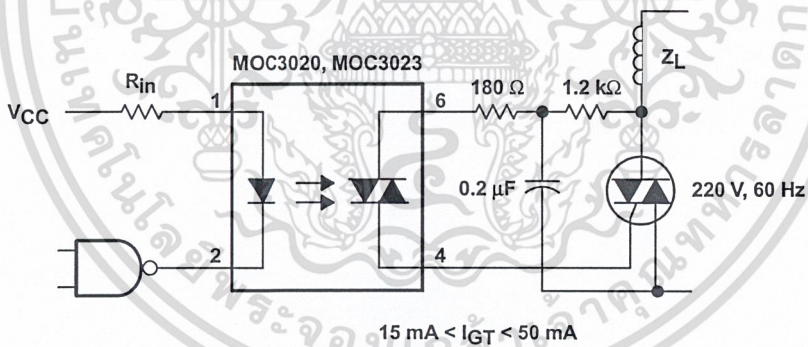


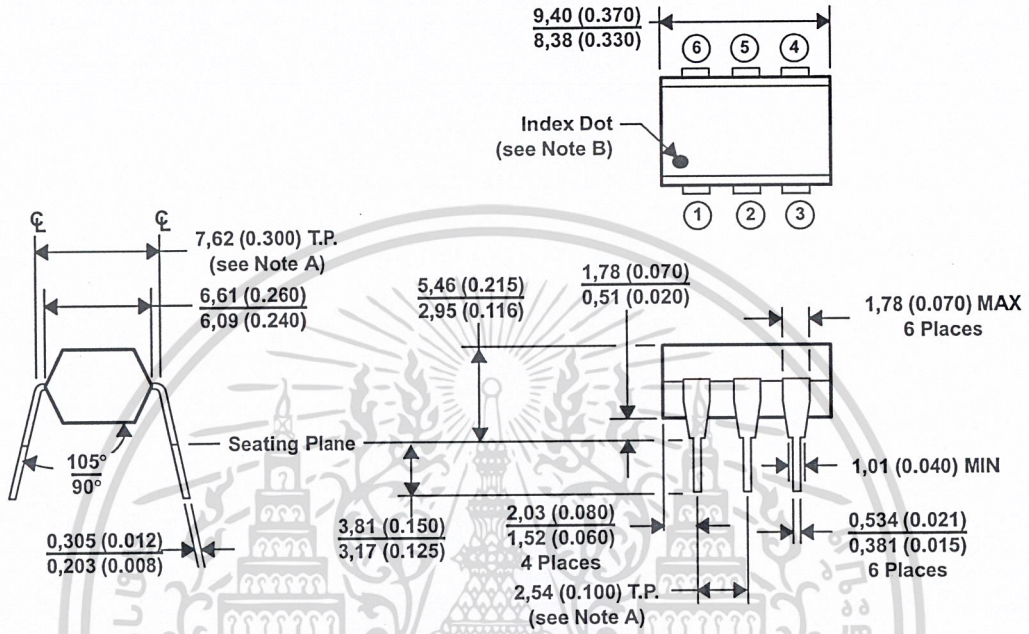
Figure 7. Inductive Load With Nonsensitive-Gate Triac

MOC3020 THRU MOC3023 OPTOCOUPLEDERS/OPTOISOLATORS

SOES025A – OCTOBER 1986 – REVISED APRIL 1998

MECHANICAL INFORMATION

Each device consists of a gallium-arsenide infrared-emitting diode optically coupled to a silicon phototriac mounted on a 6-terminal lead frame encapsulated within an electrically nonconductive plastic compound. The case can withstand soldering temperature with no deformation and device performance characteristics remain stable when operated in high-humidity conditions.



- NOTES: A. Leads are within 0.13 (0.005) radius of true position (T.P.) with maximum material condition and unit installed.
 B. Pin 1 identified by index dot.
 C. The dimensions given fall within JEDEC MO-001 AM dimensions.
 D. All linear dimensions are given in millimeters and parenthetically given in inches.

Figure 8. Packaging Specifications