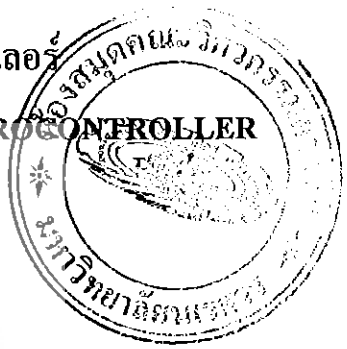
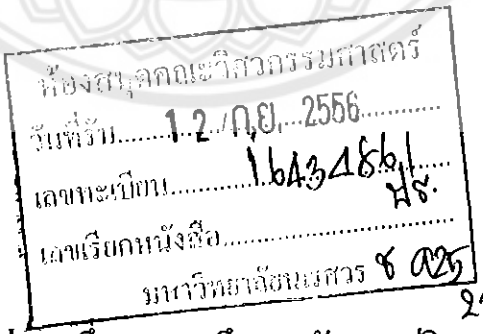
ตู้แช่วัคซีนควบคุมด้วยไมโครคอนโทรลเลอร์

VACCINE REFRIGERATOR CONTROL WITH MICROCONTROLLER



นายไชยศิริ เนตรแก้ว รหัส 49364011

นายศิริชัย ศิรวุฒินานนท์ รหัส 49364295



ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมไฟฟ้า ภาควิชาวิศวกรรมไฟฟ้าและคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร

ปีการศึกษา 2552

ใบรับรองปริญญานิพนธ์

ชื่อหัวข้อโครงงาน ตู้แช่วัคซีนควบคุมด้วยไมโครคอนโทรลเลอร์

ผู้ดำเนินโครงงาน นายไชยศิริ เนตรแก้ว รหัส 49364011

นายศิริชัย ศิรวุฒินานนท์ รหัส 49364295

ที่ปรึกษาโครงงาน ดร.มุฑิตา สงฆ์จันทร์

สาขาวิชา วิศวกรรมไฟฟ้า

ภาควิชา วิศวกรรมไฟฟ้าและคอมพิวเตอร์

ปีการศึกษา 2552

.....

คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร อนุมัติให้ปริญญานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิศวกรรมศาสตรบัณฑิต สาขาวิชาวิศวกรรมไฟฟ้า

.....ที่ปรึกษาโครงงาน
(ดร.มุฑิตา สงฆ์จันทร์)

.....กรรมการ
(ดร.อนันตชัย อยู่แก้ว)

.....กรรมการ
(ดร.ศุภวรรณ พลพิทักษ์ชัย)

ชื่อหัวข้อโครงงาน	**ตู้แช่วัคซีนควบคุมด้วยไมโครคอนโทรลเลอร์**
ผู้ดำเนินโครงงาน	**นายไชยศิริ เนตรแก้ว รหัส 49364011**
	นายศิริชัย ศิรวุฒินานนท์ รหัส 49364295
ที่ปรึกษาโครงงาน	**ดร.มุฑิตา สงฆ์จันทร์**
สาขาวิชา	**วิศวกรรมไฟฟ้า**
ภาควิชา	**วิศวกรรมไฟฟ้าและคอมพิวเตอร์**
ปีการศึกษา	2552

บทคัดย่อ

โครงงานนี้เป็นการประดิษฐ์ตู้แช่วัคซีนควบคุมด้วยไมโครคอนโทรลเลอร์โดยใช้เพลเท็ยร์เป็นตัวทำความเย็นแบบพีไอดี (PID Controller) และศึกษาทฤษฎีของการควบคุมแบบพีไอดีด้วยวิธีการของ Ziegler-Nichols ในการปรับตั้งค่าพารามิเตอร์ โดยวิธีการปรับจูนค่าเกนของตัวควบคุมพีไอดีจะต้องทำการทดลองเพื่อนำผลการทดลองมาคำนวณหาสมการค่าพารามิเตอร์ K_p, K_i และK_d ในการควบคุมอุณหภูมิของอากาศภายในตู้แช่วัคซีน จากผลการทดลองตู้แช่วัคซีนควบคุมด้วยไมโครคอนโทรลเลอร์มีความสามารถในการทำความเย็นให้ได้อุณหภูมิตามที่ตั้งค่าไว้ โดยควบคุมอุณหภูมิได้ในช่วง 5 - 8 องศาเซลเซียส

Project title	Vaccine Refrigerator Control with Microcontroller
Name	Mr. Chaisiri Natkaew ID. 49364011
	Mr. Sirichai Sirawutnanon ID. 49364295
Project advisor	Miss Mutita Songjun, Ph.D.
Major	Electrical Engineering
Department	Electrical and Computer Engineering
Academic year	2009

Abstract

This project aim to invent the vaccine refrigerator control with microcontroller which is PID Controller refrigeration by using peltier and study the PID theory with Ziegler-Nichols mean of parameter setting. The mean of value adjustment of PID Controller has to experiment and calculate the K_p, K_i and K_d parameter equation from the result of the experiment in the temperature control of the air inside the vaccine refrigerator. The result of the project was found that the vaccine refrigerator control with microcontroller can refrigerate as temperature setting of 5 – 8 °C.

กิตติกรรมประกาศ

โครงงานวิศวกรรมเรื่องตู้แช่วัคซีนควบคุมด้วยไมโครคอนโทรลเลอร์ สำเร็จได้ด้วยดี เนื่องจากได้รับคำแนะนำ และความช่วยเหลือรวมทั้งข้อคิดเห็นต่างๆ อันเป็นประโยชน์ในการทำโครงงานนี้จาก ดร. มุฑิตา สงฆ์จันทร์ ขอกราบขอบพระคุณที่ประสิทธิ์ประสาทวิชาความรู้ให้กับคณะผู้ดำเนินงาน

ขอขอบคุณ อาจารย์ประจำภาควิชาวิศวกรรมไฟฟ้าและคอมพิวเตอร์ อาจารย์ประจำภาควิชาวิศวกรรมเครื่องกล และคณะกรรมการ ที่ให้ความรู้ ข้อคิดเห็น และความช่วยเหลือในทุกๆ ด้าน รวมทั้ง ทางคณะวิศวกรรมศาสตร์ที่ให้ความเอื้อเฟื้อเครื่องมือ อุปกรณ์ และเงินทุนสำหรับสนับสนุนบางส่วนในการทำโครงงานครั้งนี้

สุดท้ายนี้ ขอขอบคุณ พี่ๆ เพื่อนๆ และครอบครัวของคณะผู้จัดทำ ที่ให้คำแนะนำ และให้กำลังใจตลอดจนการสนับสนุนในทุกๆ ด้าน เป็นอย่างดี จนโครงงานนี้สำเร็จลุล่วงไปได้ด้วยดี

นายไชยศิริ เนตรแก้ว

นายศิริชัย ศิรวุฒินานนท์

สารบัญ

หน้า

ใบรับรองปริญญานิพนธ์ ..... ก
บทคัดย่อภาษาไทย ..... ข
บทคัดย่อภาษาอังกฤษ ..... ค
กิตติกรรมประกาศ ..... ง
สารบัญ ..... จ
สารบัญตาราง ..... ฉ
สารบัญรูป ..... ญ

บทที่ 1 บทนำ 1
1.1 ที่มาและความสำคัญของโครงงาน 1
1.2 วัตถุประสงค์ของโครงงาน 1
1.3 ขอบเขตของโครงงาน 2
1.4 ขั้นตอนและแผนการดำเนินงาน 2
1.5 ประโยชน์ที่คาดว่าจะได้รับจากโครงงาน 2
1.6 งบประมาณ 3

บทที่ 2 หลักการและทฤษฎี 4
2.1 อุณหภูมิและสเกลอุณหภูมิ 4
2.2 กฎข้อที่หนึ่งและกฎข้อที่สองของเทอร์โมไดนามิกส์สำหรับระบบเปิด 6
2.2.1 กฎข้อที่หนึ่งของเทอร์โมไดนามิกส์สำหรับระบบเปิด 6
2.2.2 กฎอนุรักษ์มวลสำหรับระบบปิด 7
2.2.3 กฎการอนุรักษ์พลังงานสำหรับระบบเปิด 7
2.2.4 กฎที่ข้อที่สองของเทอร์โมไดนามิกส์สำหรับระบบเปิด 8
2.3 การถ่ายโอนความร้อน (Heat transfer) 8
2.4 สัมประสิทธิ์สมรรถนะ 8
2.5 ปรากฏการณ์เพลเทียร์เอฟเฟ็กต์ 9
2.5.1 โครงสร้างเบื้องต้นของอุปกรณ์เพลเทียร์ 9

สารบัญ (ต่อ)

หน้า

2.5.2 เบอร์ของเพลเทียร์ หรือ TEC 11
2.5.3 จุดอ่อนของเพลเทียร์ 13
2.6 ไมโครคอนโทรลเลอร์ 13
2.6.1 สถาปัตยกรรมของไมโครคอนโทรลเลอร์ 15
2.6.2 หน่วยประมวลผลกลางหรือซีพียู 17
2.6.3 หน่วยความจำ 18
2.6.4 หน่วยความจำพิเศษ 20
2.6.5 การทำงานของไมโครคอนโทรลเลอร์ 22
2.7 ไมโครคอนโทรลเลอร์ตระกูล AVR 23
2.7.1 โครงสร้างและสถาปัตยกรรมของ ATmega 64 23
2.7.2 ตัวถังและตำแหน่งขาสัญญาณของ ATmega 64 25
2.8 การทำงานของเซนเซอร์วัดอุณหภูมิ 28
2.8.1 ขาสัญญาณสำหรับสื่อสารข้อมูลของโมดูล DS18S20 29
2.8.2 รูปแบบการสื่อสารข้อมูลของ SHT15 29
2.9 รีเลย์ 31
2.10 มอเตอร์ไฟฟ้ากระแสตรง (DC MOTOR) 31
2.10.1 หลักการทำงานของมอเตอร์กระแสตรง 31
2.10.2 การขับและกลับทิศทางของมอเตอร์กระแสตรง (DC MOTOR) 32
2.10.3 การควบคุมความเร็วของมอเตอร์กระแสตรง 33
2.11 การควบคุมแบบพีไอดี (PID Control) 33
2.11.1 ทฤษฎีการควบคุมแบบพีไอดี 33
2.11.2 การควบคุมแบบสัดส่วน (Proportion Control) 34
2.11.3 การควบคุมแบบอินทิกรัล (Integral Control) 34
2.11.4 การควบคุมแบบอนุพันธ์ (Derivative Control) 35
2.11.5 การตั้งค่าตัวควบคุมผลตอบสนองต่อสัญญาณขั้นบันได โดยใช้วิธี Ziegler-Nichols 36
2.11.6 การแปลงค่าพารามิเตอร์ และค่าคงที่เวลาของตัวควบคุม 37
2.11.7 ตัวควบคุม พีไอดีแบบดิจิตอล 38

สารบัญ (ต่อ)

หน้า

บทที่ 3 ขั้นตอนการดำเนินงานและการออกแบบ 40

3.1 การออกแบบวงจรควบคุมระบบตู้แช่วัคซีนด้วยไมโครคอนโทรลเลอร์ 40

3.1.1 วงจรไฟเลี้ยงไมโครคอนโทรลเลอร์ 42

3.1.2 การต่อใช้งานจอแอลซีดี 16x2 42

3.1.3 การต่อใช้งานสวิตช์ 43

3.1.4 การต่อใช้งานไอซีวัดอุณหภูมิ DS18S20 43

3.1.5 การต่อใช้งาน MAX 232 44

3.1.6 การต่อใช้งานสัญญาณเตือน 45

3.1.7 การต่อใช้งานพัดลม DC 12 โวลต์ 45

3.1.8 การต่อใช้งานเพลเทียร์ 46

3.1.9 การต่อใช้งานไมโครคอนโทรลเลอร์ตระกูล AVR (ATmega 64) 47

3.2 การเขียนโปรแกรมควบคุมการทำงานของไมโครคอนโทรลเลอร์ 48

3.2.1 การทำงานของโปรแกรมการควบคุมแบบพีไอดี 50

3.2.2 การเขียนโปรแกรมนำค่า u(t) ออกไปควบคุมอุปกรณ์ภายนอก 52

3.3 การประกอบบอร์ดควบคุมไมโครคอนโทรลเลอร์เข้ากับตู้แช่วัคซีน 53

บทที่ 4 ผลการทดลองและผลการวิเคราะห์ 56

4.1 การทดลองหาระยะเวลาที่ตู้แช่วัคซีนควบคุมด้วยไมโครคอนโทรเลอร์ทำงาน 56

4.2 การคำนวณหาค่าพารามิเตอร์ K_p, K_i และ K_d 58

4.3 ผลการทดลอง 61

บทที่ 5 สรุปผลของโครงงาน 70

5.1 สรุปผลการทดลอง 70

5.2 ปัญหาและการแก้ไข 70

5.3 ข้อเสนอแนะ 71

เอกสารอ้างอิง 72

สารบัญ (ต่อ)

หน้า

ภาคผนวก ก รายละเอียดของโปรแกรม 73
ภาคผนวก ข รายละเอียดของ ATmaga 64 88
ภาคผนวก ค รายละเอียดของ LCD 16×2 95
ภาคผนวก ง รายละเอียดของ TLP250N, IRFP2907, LM2575, POWER RELAY 97
ภาคผนวก จ รายระเอียดของเซ็นเซอร์ DS18S20 113
ภาคผนวก ฉ รายระเอียดของไอซีเบอร์ MAX 232 120
ประวัติผู้ดำเนินโครงงาน 122

สารบัญตาราง

ตารางที่ หน้า

2.1 ชื่อขา ตำแหน่งขา ชนิดขาและหน้าที่การทำงานของไมโครคอนโทรลเลอร์ ATmega 64..... 26
2.2 การตั้งค่าพารามิเตอร์โดยการวิเคราะห์จากผลตอบสนองต่อสัญญาณขั้นบันได 37
2.3 สัมประสิทธิ์ของตัวควบคุม และค่าคงที่ของเวลา..... 38
4.1 ผลการทดลองหาระยะเวลาที่ตู้แช่วัคซีนควบคุมด้วยไมโครคอนโทรเลอร์ทำงาน เมื่ออุณหภูมิมีการเปลี่ยนแปลงทุก 0.5 องศาเซลเซียส..... 56
4.2 การคำนวณหาคุณสมบัติของระบบโดยการวิเคราะห์จากผลตอบสนองต่อสัญญาณขั้นบันได โดยใช้วิธี Ziegler-Nichols 59
4.3 การคำนวณหาค่าพารามิเตอร์โดยการวิเคราะห์จากผลตอบสนองต่อสัญญาณขั้นบันไดโดยใช้วิธี Ziegler-Nichols 60
4.4 การคำนวณหาค่าสัมประสิทธิ์ของตัวควบคุม และค่าคงที่ของเวลา 60
4.5 ผลการทดลองหาอุณหภูมิในระยะเวลาทุกๆ 1 นาที เมื่อเซตอุณหภูมิ 15.0 องศาเซลเซียส 61
4.6 ผลการทดลองหาอุณหภูมิในระยะเวลาทุกๆ 1 นาที เมื่อเซตอุณหภูมิ 12.0 องศาเซลเซียส 64
4.7 ผลการทดลองหาอุณหภูมิในระยะเวลาทุกๆ 1 นาที เมื่อเซตอุณหภูมิ 10.0 องศาเซลเซียส 67

สารบัญรูป

รูปที่ หน้า

2.1 การเกิดสมดุลระหว่างวัตถุ ก และวัตถุ ข ที่มีอุณหภูมิต่างกัน 4
2.2 เปรียบเทียบสเกลอุณหภูมิในหน่วยต่างๆ 6
2.3 เปรียบเทียบลักษณะของระบบปิดและระบบเปิด 7
2.4 แสดงโครงสร้างเบื้องต้นของอุปกรณ์เพลเทียร์ 10
2.5 แสดงการเกิดเพลเทียร์เอฟเฟ็กต์ 11
2.6 ความหมายของรหัสเบอร์ของเพลเทียร์ 12
2.7 ความสัมพันธ์ระหว่างความสามารถการปั๊มความร้อน (Qc) กับกระแส (I) ที่จ่ายเพลเทียร์ 13
2.8 โครงสร้างและส่วนประกอบหลักเบื้องต้นของไมโครคอนโทรลเลอร์ 15
2.9 โครงสร้างสถาปัตยกรรมของไมโครคอนโทรลเลอร์แบบพรินซ์ตันหรือฟอนนิวแมน 16
2.10 โครงสร้างสถาปัตยกรรมของไมโครคอนโทรลเลอร์แบบฮาร์วาร์ด 17
2.11 ส่วนประกอบหลักของซีพียูในไมโครคอนโทรลเลอร์ 18
2.12 กลไกการทำงานของสแต็กอย่างง่าย 22
2.13 สถาปัตยกรรมภายในของไมโครคอนโทรลเลอร์ ATmega 64 25
2.14 ตัวถังของ ATmega 64 และตำแหน่งขาสัญญาณต่างๆ 26
2.15 Block diagram ของ DS18S20 28
2.16 รูปร่างของโมดูล DS18S20 29
2.17 Temperature Register Format 29
2.18 Memory Map 30
2.19 รีเลย์ 31
2.20 วงจรภายในของรีเลย์ 31
2.21 แสดงโครงสร้างทั่วไปของมอเตอร์กระแสตรง 32
2.22 แสดงแผนภาพบล็อกของการควบคุมแบบพีไอดี 33
2.23 แสดงแผนภาพบล็อกของการควบคุมแบบสัดส่วน 34
2.24 แสดงแผนภาพบล็อกของการควบคุมแบบอินทิกรัล 35
2.25 แสดงแผนภาพบล็อกของการควบคุมแบบอนุพันธ์ 36
2.26 การหาคุณสมบัติของระบบโดยการวิเคราะห์จากผลตอบสนองต่อสัญญาณขั้นบันได 37
2.27 รูปแสดงแผนภาพการไหลของสัญญาณสำหรับตัวควบคุมแบบดิจิตอล 38

สารบัญรูป (ต่อ)

รูปที่ หน้า

3.1 ภาพรวมของระบบทั้งหมดในโครงงาน 40
3.2 รูปวงจรควบคุมอุณหภูมิด้วยไมโครคอนโทรลเลอร์ 41
3.3 วงจรไฟเลี้ยงไมโครคอนโทรลเลอร์ 42
3.4 วงจรจอแอลซีดี 42
3.5 วงจรสวิตช์ 43
3.6 วงจร DS18S20 ตรวจวัดอุณหภูมิ 43
3.7 วงจร MAX 232 ส่งข้อมูลไปยังคอมพิวเตอร์ 44
3.8 วงจร LDR 45
3.9 วงจร BUZZER 45
3.10 วงจรพัดลม DC 12 โวลต์ 45
3.11 วงจรจ่ายกระแสไฟฟ้าเพลเทียร์ 46
3.12 วงจรใช้งานไมโครคอนโทรลเลอร์ 47
3.13 รูปโปรแกรม Code Vision AVR C Compiler 48
3.14 การทำงานของโปรแกรม 49
3.15 แผนภูมิการทำงานของโปรแกรมควบคุมพีไอดีแบบดิจิตอล 51
3.16 แผนภูมิการทำงานของโปรแกรมนำค่า u(t) ออกไปควบคุมอุปกรณ์ภายนอก 52
3.17 รูปการทำงานแบบ PWM ของ u(t) 53
3.18 บอร์ดควบคุมไมโครคอนโทรลเลอร์ 54
3.19 กล่องควบคุมไมโครคอนโทรลเลอร์ 54
3.20 ตู้แช่วัคซีนควบคุมด้วยไมโครคอนโทรลเลอร์ 55
4.1 กราฟแสดงผลระยะเวลาที่อุณหภูมิมีการเปลี่ยนแปลง 58
4.2 การหาคุณสมบัติของระบบโดยการวิเคราะห์จากผลตอบสนองต่อสัญญาณขั้นบันได 59
4.3 กราฟแสดงผลการเปรียบเทียบหาอุณหภูมิที่มีการเปลี่ยนแปลง 63
4.4 กราฟแสดงผลการเปรียบเทียบหาอุณหภูมิที่มีการเปลี่ยนแปลง 66
4.5 กราฟแสดงผลการเปรียบเทียบหาอุณหภูมิที่มีการเปลี่ยนแปลง 69

บทที่ 1
บทนำ

1.1 ที่มาและความสำคัญของโครงงาน

วิทยาศาสตร์และเทคโนโลยีเป็นปัจจัยสำคัญในการพัฒนาประเทศทั้งในด้านวิทยาการความก้าวหน้าทางด้านเครื่องมือการแพทย์ เครื่องมือทางด้านการเกษตร เครื่องมือทางด้านการทำอาหาร เป็นต้น ซึ่งวิทยาการความก้าวหน้าทางด้านเครื่องมือแพทย์มีความสำคัญมากภายในประเทศ เป็นสิ่งที่ช่วยพัฒนาให้คุณภาพชีวิตของประชากรภายในประเทศนั้นๆ ดีขึ้น จากการรักษาพยาบาล การให้บริการสาธารณสุขด้วยเครื่องมือที่ทันสมัยและมีประสิทธิภาพรวมไปถึงยาและเวชภัณฑ์ต่างๆ เนื่องจากจำนวนประชากรที่มีการเจ็บป่วยได้เข้ารับการรักษาพยาบาลเป็นจำนวนมาก เป็นเหตุผลให้มีปริมาณยาและเวชภัณฑ์เพิ่มมากขึ้นเพื่อเป็นการรองรับการใช้รักษาผู้ที่เจ็บป่วย ซึ่งยาและเวชภัณฑ์บางชนิดนั้นต้องการเก็บรักษาในสถานที่ที่มีอุณหภูมิที่พอเหมาะคงที่โดยมีลักษณะเป็นตู้แช่วัคซีนขนาดเล็กกะทัดรัดขนาดบรรจุ 4 ลิตร โดยประมาณ เหมาะแก่การใช้งานใน โรงพยาบาล คลินิก อนามัย ที่มีพื้นที่จำกัด เพื่อเป็นการเก็บรักษายาและเวชภัณฑ์ให้มีคุณภาพคงเดิม มีขนาดบรรจุภัณฑ์ที่เล็ก ทำให้เกิดความเป็นระเบียบ สะดวกในการเก็บรักษาและการนำออกมาใช้ในแต่ละครั้ง

โครงงานนี้จึงเป็นการศึกษาค้นคว้าหลักการทำงานของเพลเทียร์เอฟเฟ็กต์ ในการทำความเย็นภายในตู้แช่วัคซีนให้มีความเย็นอยู่ในภาวะสมดุลตามที่ต้องการ เพื่อเป็นประโยชน์ต่อการแพทย์ในการรักษาคุณภาพของวัคซีนเกี่ยวกับระบบควบคุมความเย็น ตลอดจนสามารถนำไปประยุกต์ใช้เพลเทียร์ในระบบทำความเย็นแบบต่างๆ ได้

1.2 วัตถุประสงค์ของโครงงาน

สร้างตู้แช่วัคซีนโดยใช้ทฤษฎีการควบคุมแบบพีไอดีที่นำไปใช้ควบคุมเพลเทียร์เอฟเฟ็กต์เพื่อรักษาความสมดุลของอุณหภูมิภายในตู้แช่วัคซีน

1.3 ขอบเขตของโครงงาน

1) ศึกษาเพลเทียร์เอฟเฟ็กต์และไมโครคอนโทรลเลอร์

2) สร้างตู้แช่วัคซีนที่ควบคุมอุณหภูมิได้ระหว่าง 5 ถึง 8 องศาเซลเซียส

3) สามารถควบคุมอุณหภูมิกระติกที่มีความจุ 4 ลิตร

4) สามารถเตือนเมื่อกระติกไม่ได้ปิดได้

5) เขียนคำสั่งควบคุมระบบไมโครคอนโทรลเลอร์ด้วยโปรแกรมภาษาซี

1.4 ขั้นตอนและแผนการดำเนินงาน

รายละเอียด	ปี 2555						
	ม.ค.	ก.พ.	มี.ค.	เม.ย.	พ.ค.	มิ.ย.	ก.ค.
1) รวบรวมข้อมูล							
2) ศึกษาการทำงานและออกแบบชิ้นงาน							
3) ติดตั้งและทดสอบ							
4) ปรับปรุงแก้ไขและวิเคราะห์ผล							
5) สรุปผลการดำเนินงาน							
6) จัดทำปริญญานิพนธ์ฉบับสมบูรณ์							

1.5 ประโยชน์ที่คาดว่าจะได้รับจากโครงงาน

1) เข้าใจหลักการทำงานของไมโครคอนโทรลเลอร์

2) เข้าใจถึงการทำงานของโปรแกรมภาษาซี

3) เข้าใจหลักการทำงานของเพลเทียร์เอฟเฟ็กซ์

4) ลดค่าใช้จ่ายในการนำเข้าตู้แช่วัคซีนที่มีราคาแพงจากต่างประเทศ

5) สามารถนำตู้แช่วัคซีนควบคุมด้วยไมโครคอนโทรลเลอร์ประยุกต์ใช้ทางการแพทย์ได้

1.6 งบประมาณในการทำโครงงาน

1) ค่าอุปกรณ์ทำความเย็นและอิเล็กทรอนิกส์	1,500 บาท
2) ค่าเอกสาร	300 บาท
3) ค่าวัสดุอื่นๆ	200 บาท
รวมเป็นเงินทั้งสิ้น (สองพันบาทถ้วน)	2,000 บาท

บทที่ 2
หลักการและทฤษฎี

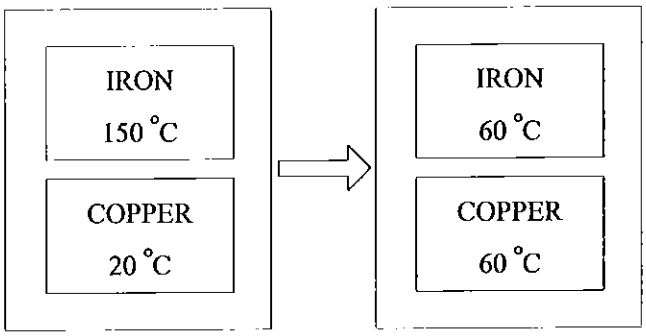
ในบทที่ 2 นี้จะอธิบายถึงหลักการและทฤษฎี ที่รวบรวมเพื่อนำมาใช้ประกอบการทำโครงงานโดยมีใจความสำคัญที่อธิบายถึงองค์ประกอบของอุปกรณ์ต่างๆและการทำงานของโปรแกรมรวมถึงทฤษฎีที่นำมาใช้งานควบคู่กับการเขียนโปรแกรม เพื่อให้ได้ผลงานออกมาตามที่ตั้งเป้าหมายไว้ตามจุดประสงค์ดังที่กล่าวไว้ในบทที่ 1

2.1 อุณหภูมิและสเกลอุณหภูมิ

แม้ว่าเราจะคุ้นเคยกันดีว่าอุณหภูมิ (Temperature) เป็นสมบัติที่ใช้บอกความร้อนหรือเย็นของวัตถุอย่างไรก็ตาม เราไม่สามารถบอกระดับอุณหภูมิจากความรู้สึกเพียงอย่างเดียว ยิ่งไปกว่านั้นบางครั้งการพิจารณาความร้อนหรือเย็นจากความรู้สึกอาจมีความผิดพลาดได้ เช่น เรามีความรู้สึกว่าวัตถุที่เป็นโลหะจะเย็นกว่าวัตถุที่เป็นไม้ แม้ว่าวัตถุทั้งสองจะมีอุณหภูมิเท่ากันก็ตาม

จากการที่ สมบัติของสารหลายอย่าง จะเกิดการเปลี่ยนแปลง ตามอุณหภูมิ โดยมีทิศทางที่มีขนาดของการเปลี่ยนแปลงที่แน่นอนปรากฏการณ์ดังกล่าวได้ถูกนำมาเป็นหลักเกณฑ์ในการวัดอุณหภูมิที่มีความแม่นยำ

เมื่อวัตถุที่มีอุณหภูมิต่างกันมาสัมผัสกันจะทำให้เกิดการถ่ายโอนความร้อนจากวัตถุที่มีอุณหภูมิสูงกว่าไปยังวัตถุที่มีอุณหภูมิต่ำกว่าจนกระทั่งอุณหภูมิของวัตถุทั้งสองเท่ากันหรือวัตถุทั้งสองอยู่ในสมดุลความร้อน ซึ่งที่สภาวะดังกล่าวการถ่ายโอนความร้อนจะสิ้นสุดลง



รูปที่ 2.1 การเกิดสมดุลระหว่างวัตถุ ก และวัตถุ ข ที่มีอุณหภูมิต่างกัน

กฎข้อที่ศูนย์ของเทอร์โมไดนามิกส์ (The zeroth low thermodynamics) กล่าวไว้ว่าวัตถุสองชิ้นอยู่ในสมดุลกับวัตถุชิ้นที่สาม แสดงว่าวัตถุสองชิ้นจะอยู่ในสมดุลของความร้อนต่อกันด้วย กฎข้อนี้เป็นกฎที่ได้จากการทดลองและเป็นหลักการในการวัดอุณหภูมิของวัตถุต่างๆ ด้วยเครื่องมือวัด โดยการเห็นวัตถุชิ้นที่สามเป็นเครื่องมือวัดโดยเมื่อใช้เครื่องมือวัดระดับอุณหภูมิของวัตถุสองชิ้นนั้นแล้วพบว่ามีค่าเท่ากัน จากหลักการ ในกฎข้อที่ศูนย์ของเทอร์โมไดนามิกส์ สามารถระบุได้ว่าวัตถุสองชิ้นอยู่ในสมดุลความร้อนต่อกันได้ แม้ว่าวัตถุสองชิ้นนั้นจะไม่ได้สัมผัสกันโดยตรงก็ตาม

สเกลอุณหภูมิ (Temperature Scale) ที่ใช้ในระบบ SI คือ องศาเซลเซียส ($^{\circ}$C) ซึ่งถูกกำหนดโดย A Celsius นักดาราศาสตร์ชาวสวีเดน (ค.ศ. 1701 -1744) ส่วนสเกลอุณหภูมิที่ใช้ ในระบบอังกฤษคือ องศาฟาเรนไฮต์ ($^{\circ}$F) กำหนดโดย G. Fahrenheit นักประดิษฐ์ชาวเยอรมัน (ค.ศ. 1686 – 1736) สเกลอุณหภูมิมักจะกำหนดอ้างอิงกับจุดอ้างอิงหลัก สองจุดคือจุดเหยือกแข็งและจุดเดือด มีระดับอุณหภูมิที่เป็น 0 และ 100 องศาเซลเซียส ตามลำดับในขณะที่สเกลฟาเรนไฮต์กำหนดให้เป็น 32 และ 212 ฟาเรนไฮต์ ตามลำดับ

ในทางเทอร์โมไดนามิกส์ (Thermodynamics scale of temperature) ซึ่งสเกลนี้ถูกกำหนดขึ้นมาจากพื้นฐานของกฎข้อที่สองของเทอร์โมไดนามิกส์ สเกลอุณหภูมิของเทอร์โมไดนามิกส์ในระบบ SI คือสเกลเคลวิน (Kelvin scale) กำหนดโย Lord Kelvin (ค.ศ. 1824 -1907) โดยใช้สัญลักษณ์เป็น K (ไม่มีเครื่องหมายองศา) ความสัมพันธ์ระหว่างสเกลเคลวินกับองศาเซลเซียสเป็นดังนี้

$$T(K) = T(^{\circ}C) + 273.15 \qquad (2.1)$$

ในระบบอังกฤษ สเกลอุณหภูมิทางเทอร์โมไดนามิกส์ คือสเกลแรงคิน (R) กำหนดโดย William Rankine (ค.ศ. 1829-1872) ความสัมพันธ์ระหว่างสเกลแรงคินกับองศาฟาเรนไฮต์เป็นดังนี้

$$T(R) = T(^{\circ}F) + 459.67 \qquad (2.2)$$

อุณหภูมิในระบบ SI และระบบอังกฤษมีความสัมพันธ์กันดังนี้

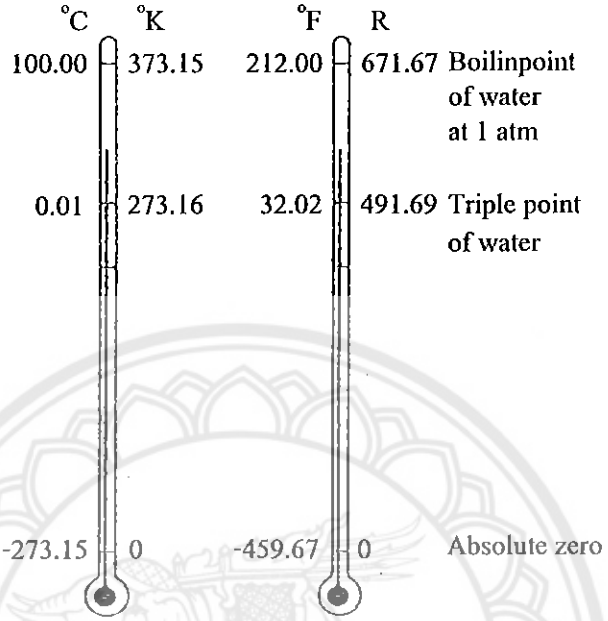
$$T(R) = 1.8\ T(K) \qquad (2.3)$$

$$T(^{\circ}F) = 1.8\ T(^{\circ}C) + 32 \qquad (2.4)$$

สำหรับขนาดของผลต่างของอุณหภูมิ (ΔT) ของอุณหภูมิในหน่วยต่างๆ มีความสัมพันธ์กันดังนี้

$$\Delta T(R) = \Delta(^{\circ}F) = 1.8\ \Delta T(K) = 1.8\ \Delta T(^{\circ}C) \qquad (2.5)$$

จากความสัมพันธ์หมายความว่าเมื่ออุณหภูมิเพิ่มขึ้น 10 องศาเซลเซียส จะมีค่าเท่ากับอุณหภูมิเพิ่มขึ้น 10 เคลวิน และจะเท่ากับอุณหภูมิเพิ่มขึ้น 18 องศาฟาเรนไฮต์ หรือ 18 แรงคิน ความสัมพันธ์ของสเกลอุณหภูมิที่กล่าวมาทั้งหมดเป็นดังแสดงในรูปที่ 2.2



รูปที่ 2.2 เปรียบเทียบสเกลอุณหภูมิในหน่วยต่างๆ

2.2 กฎข้อที่หนึ่งและกฎข้อที่สองของเทอร์โมไดนามิกส์สำหรับระบบเปิด

2.2.1 กฎข้อที่หนึ่งของเทอร์โมไดนามิกส์สำหรับระบบเปิด

ระบบเปิดหรือปริมาตรควบคุม (Open systems หรือ Control volumes) คือระบบที่มีการไหลของมวลผ่านขอบเขตของระบบมีอุปกรณ์ ในงานวิศวกรรมหลายชนิดที่มีลักษณะเป็นระบบเปิด เช่น กังหัน (Radiators) ปั๊ม (Pumps) เครื่องอัด (Compressors) ท่อลำเลียง (Pipes) หัวฉีด (Nozzles) หม้อน้ำรถยนต์ (Radiators) และเครื่องแลกเปลี่ยนความร้อน (heat exchangers) เป็นต้น

ลักษณะของระบบเปิดมีหลายรูปแบบ กล่าวคือ อาจมีมวลเข้าหรือออกจากระบบ เพียงอย่างใดอย่างหนึ่งอาจมีมวลเข้าและออกจากระบบ ในขณะเดียวกัน ก็ได้โดยขึ้นกับลักษณะและการใช้งานระบบเนื่องจากระบบเปิดมีการถ่ายโอนทั้งมวลและพลังงาน ดังนั้น ในการวิเคราะห์ระบบเปิดจึงต้องคำนึงถึงผลที่เกิดจาก การปรากฏการณ์ทั้งสองส่วน สำหรับผลของการถ่ายโอนมวล จะเป็นไปตามกฎของอนุรักษ์มวลและผลิตผลที่เกิดขากการถ่ายโอนพลังงานจะเป็นไปตามกฎอนุรักษ์พลังงาน

2.2.2 กฎอนุรักษ์มวลสำหรับระบบเปิด

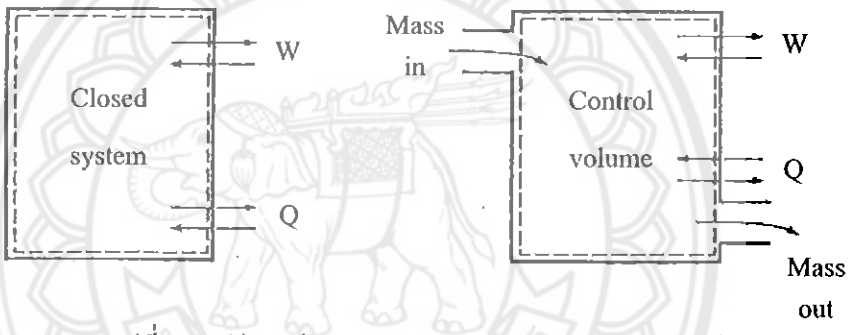
กฎอนุรักษ์ระบบเปิดใดๆ เป็นดังแสดงเป็นไปตามสมการดังนี้

[มวลที่เข้าสู่ระบบ] – [มวลที่ออกจากระบบ] = [มวลในระบบที่เปลี่ยนแปลงไป]

2.2.3 กฎการอนุรักษ์พลังงานสำหรับระบบเปิด

กฎอนุรักษ์พลังงานสำหรับระบบเปิดพบว่าการเปลี่ยนแปลงพลังงานในระบบปิดเกิดจากผลของการถ่ายโอนพลังงานในรูปความร้อนและงาน ตามสมการได้ดังนี้

$$Q - W = \Delta E \tag{2.6}$$



รูปที่ 2.3 เปรียบเทียบลักษณะของระบบปิดและระบบเปิด

สำหรับระบบเปิดซึ่งมีการถ่ายโอนมวลผ่านขอบเขตของระบบในระหว่างกระบวนการนอกจากจะมี การถ่ายโอนพลังงาน ในรูปความร้อนและงานเหมือนกับระบบปิด แล้วพบว่ามีการถ่ายโอนพลังงาน เนื่องจากผลของการโอนมวลด้วย กล่าวคือ ในขณะที่มวลไหลเข้าสู่ระบบ มวลนั้นจะนำเอาพลังงานส่วนหนึ่งเข้าสู่ระบบและในลักษณะเดียวกันสำหรับมวลที่ไหลออกจากระบบจะนำเอาพลังงานส่วนหนึ่งออกจากระบบด้วยเช่นกัน ดังนั้นสมการกฎอนุรักษ์พลังงาน สำหรับระบบเปิดจึงเป็นดังนี้

$$Q - W + \sum E_{fi} - \sum E_{fe} = \Delta E_{CV} \tag{2.7}$$

2.2.4 กฎที่ข้อที่สองของเทอร์โมไดนามิกส์สำหรับระบบเปิด

ถ้อยคำกล่าวในกฎข้อที่สองทางเทอร์โมไดนามิกส์ที่สำคัญมี 2 ถ้อยคำกล่าว คือ ถ้อยคำกล่าวของ Kelvin – Planck ซึ่งเกี่ยวข้องกับกลจักรความร้อนและกล่าวไว้แล้วข้างต้น ส่วนถ้อยคำกล่าวของ Clausius จะเกี่ยวข้องกับ เครื่องทำความเย็น และปั๊มความร้อน ถ้อยคำกล่าวของ Clausius กล่าวไว้ ดังนี้ "เป็นไปไม่ได้ ที่จะสร้างอุปกรณ์ที่สามารถถ่ายโอนความร้อน จากบริเวณที่มีอุณหภูมิต่ำไปยังบริเวณที่มีอุณหภูมิสูงโดยปราศจากการป้อนงานให้อุปกรณ์นั้น"

2.3 การถ่ายโอนความร้อน (Heat transfer)

ความร้อนเป็นพลังงานรูปหนึ่ง ดังนั้นความร้อนมีหน่วยเหมือนกับพลังงานรูปอื่นๆ โดยระบบ SI ความร้อนมีหน่วยมาตรฐานเป็น กิโลจูล (kJ) ส่วนระบบอังกฤษมีหน่วยเป็น บีทียู (Btu)

การถ่ายโอนความร้อน ระหว่างแหล่ง 2 แหล่ง ที่มีความแตกต่างของอุณหภูมิก็ถือได้ว่าเป็นกระบวนการที่ผันกลับไม่ให้ลองพิจารณา กระป๋องน้ำอัดลมที่เย็น และถ้าต้องการให้กระป๋องน้ำอัดลมเย็นตัวลงกระทั่งมีอุณหภูมิกระทั่งมีอุณหภูมิเท่าเดิมอีกครั้งจะทำได้วิธีเดียวคือแช่ในเครื่องทำความเย็น

ถ้ายิ่งความแตกต่างของอุณหภูมิระหว่างแหล่ง 2 แหล่งมีค่าน้อย อัตราเร็วในการถ่ายโอนความร้อนก็จะยิ่งมีค่าน้อยด้วยเช่นกัน

2.4 สัมประสิทธิ์สมรรถนะ

ประสิทธิภาพของเครื่องทำความเย็นถูกแสดงในเทอมของสัมประสิทธิ์สมรรถนะ (Coefficient of Performance; COP) และแทนด้วยสัญลักษณ์ COP_R วัตถุประสงค์ของเครื่องทำความเย็นคือ การกำจัดความร้อน (Q_L) ออกจากเครื่องทำความเย็น ดังนั้นเพื่อให้บรรลุวัตถุประสงค์จึงจำเป็นต้องมีการป้อนงาน $W_{net.in}$ เข้าไปค่า COP ของเครื่องทำความเย็นจึงสามารถแสดงได้ด้วยสมการดังต่อไปนี้

$$COP_R = \frac{Q_L}{W_{net\ in}} \tag{2.8}$$

นอกจากนี้ความสัมพันธ์ดังแสดงในสมการ (2-7) ยังสามารถแสดงในรูปของอัตราโดยแทนที่ Q_L ด้วย $\dot{Q}_L$ และ $W_{net.in}$ และจากกฎอนุรักษ์พลังงานของอุปกรณ์ที่ทำงานแบบวัฏจักร

$$W_{net,in} = Q_H - Q_L \text{ (kJ)} \quad (2.9)$$

แทนสมการ (2.9) ลงในสมการ (2.8) จะได้

$$COP_R = \frac{Q_L}{Q_H - Q_L} = \frac{1}{Q_H/Q_L} \quad (2.10)$$

ข้อสังเกตคือค่า COP_R อาจมีค่ามากกว่าหนึ่งได้นั้นหมายความว่าปริมาณความร้อนที่ถูกกำจัดออกจากช่องทำความเย็นอาจมากกว่าปริมาณที่ป้อนเข้าไปได้ซึ่งจะมีความแตกต่างกับค่าประสิทธิภาพเชิงความร้อนที่มักมีค่าเกินหนึ่งไม่ได้

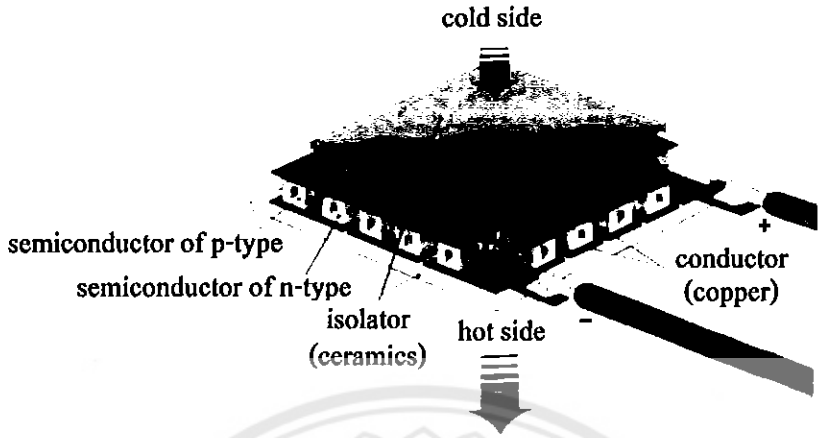
2.5 ปรากฏการณ์เพลเทียร์เอฟเฟ็กต์

ปรากฏการณ์เพลเทียร์เอฟเฟ็กต์ เกิดขึ้นเมื่อนำเอากระแสไฟฟ้า ส่งผ่านโลหะที่มีความแตกต่างกัน 2 ชนิดหรือสารกึ่งตัวนำที่ต่างชนิดซึ่งนำมาประกบกัน (สร้างจังชั่น)ในด้านหนึ่งเกิดความร้อนขึ้นกับจังชั่นอีกด้านที่ตรงข้ามกันจะไล่ความร้อนออก ปรากฏการณ์นี้จึงถูกนำมาทำเป็นตัวโซลิดสเตต ฮีตเตอร์

2.5.1 โครงสร้างเบื้องต้นของอุปกรณ์เพลเทียร์

โครงสร้างเบื้องต้นของอุปกรณ์เพลเทียร์ ประกอบด้วย สาร (N-type) และ (P-type) บิสมัท เทลลูไรท์ (สร้างมาจากสารผสมบิสมัท (Bi) กับเทลลูเรียม (Te) วางลงบนฐานรองเซรามิก) ที่สร้างเป็นรูปลูกเต๋าเล็กๆ จำนวนมาก แล้วเอาแผ่นเซรามิก มาประกบไว้ทั้งสองด้าน ในขณะที่สารชนิด (N-type) และ (P-type) ซึ่งทำเป็นเนื้ออัลลอยส์ ระหว่างบิสมัท (Bi) กับเทลลูเรียม (Te) ทั้งสองมีความแตกต่างในความหนาแน่นอิเล็กตรอนที่อุณหภูมิเท่ากันลูกเต๋าชนิด (P-type) เป็นโลหะที่มีอิเล็กตรอนไม่เพียงพอ (Deficiency of electron) ในขณะที่ลูกเต๋าชนิด (N-type) เป็นโลหะที่มีอิเล็กตรอนหนาแน่นมาก (Excess of electron) เมื่อกระแสไฟฟ้าไหลผ่านตัวโมดูล ตัวโมดูลจะพยายามสร้างดุลภาพ (Attempt to establish) ขึ้นในมวลภายในกระแสไฟฟ้าที่ให้กับสารชนิดพี ทำให้จังชั่นเกิดความร้อนขึ้น และสารชนิดเอ็นเกิดความเย็นขึ้น เมื่ออุปกรณ์นี้ทำงาน ในระดับอุณหภูมิเดียวกัน ทำให้ด้านที่เกิดความร้อน มีความร้อนเพิ่มขึ้น และด้านที่เย็น มีความเย็นมากขึ้นโดยทั่วไป เพลเทียร์ทำงานที่กระแส 4-10 แอมป์ ที่แรงดันไฟฟ้า 12 โวลต์ แต่ในการผลิตเพื่องานอุตสาหกรรม จะใช้เพลเทียร์ให้ทำงานที่กระแส 100 แอมป์ หรือมากกว่านี้ ความต้านทานโหลดใช้เป็นค่าเพื่อคิด

ปริมาณความร้อน (หรือความเย็น) ที่เกิดขึ้นคิดมาจากการเอาค่ายกกำลังสองของกระแสคูณด้วยค่าความต้านทาน ซึ่งได้จากสูตร Watt = I^2R



รูปที่ 2.4 แสดงโครงสร้างเบื้องต้นของอุปกรณ์เพลเทียร์

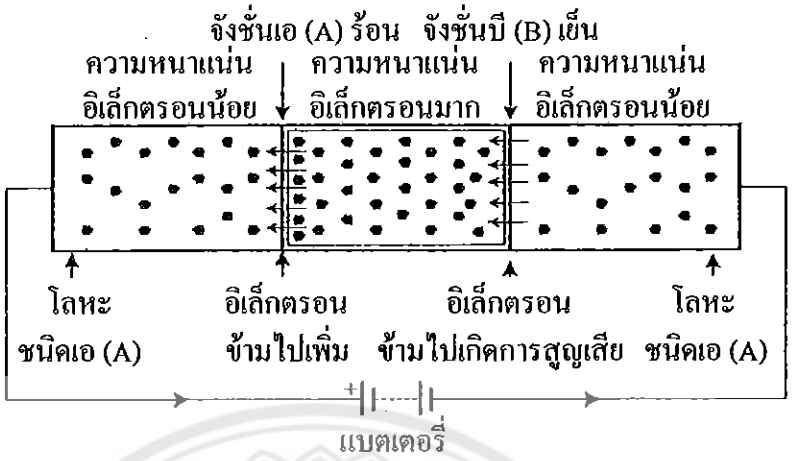
ปกติความต่างเรื่องอุณหภูมิระหว่างด้านร้อนกับด้านเย็นต่างกันไม่น้อยกว่า 65 องศาเซลเซียส และสามารถเพิ่มหรือลดความร้อนได้จากค่ากระแสไฟฟ้าที่จ่ายให้กับเพลเทียร์

เพลเทียร์ไม่มีอุปกรณ์ใดขยับขณะทำงานหรือขณะเคลื่อนย้าย จึงเป็นอุปกรณ์ที่มีความทนทาน (Rugged) และไม่ส่งเสียงรบกวนขณะทำงาน ขนาดมาตรฐานของเพลเทียร์มีขนาด 40×40 ตารางมิลลิเมตร (ขนาดเล็กกว่ามาตรฐานก็มี) ความหนาอยู่ที่ประมาณ 4 มิลลิเมตร อายุการใช้งานเฉลี่ยประมาณ 200,000 ชั่วโมง หรือมากกว่า 20 ปี สำหรับการใช้งานในแบบคูลลิ่งโหมดอย่างเดียว ตัวของเพลเทียร์มีสายไฟฟ้าสีแดงและสีดำ เป็นเครื่องหมายแสดงถึงขั้วการจ่ายไฟบวกและลบ หากสายสีแดงอยู่ทางด้านขวามือ โดยที่ให้ปลายหันเข้าหาตัวเราและแผ่นเพลตด้านล่างของเพลเทียร์จะเป็นเพลตที่ทำให้ร้อน

เมื่อนำเอากระแสไฟฟ้าจ่ายเข้าไปยังอุปกรณ์ดังแสดงการต่อวงจรในรูปที่ 2.5 พบว่าศักย์ไฟลบจากแบตเตอรี่ถูกส่งเข้ากระทำต่อจังชั่นบี (B) ผลักให้เกิดการไหลของอิเล็กตรอนในทิศทางจากขวาไปซ้ายมันเคลื่อนจากทิศทางของเนื้อสารที่มีความหนาแน่นของอิเล็กตรอนเบาบางไปยังด้านหนึ่งของสารที่มีอิเล็กตรอนหนาแน่น กรณีเช่นนี้ย่อมเกิดแรงต้านโดยธรรมชาติมันจึงเกิดงาน (Work) และให้กำลังงานออกมา (Given up energy) มันเคลื่อนตัวได้ช้าลงจังชั่นด้านนี้เกิดความเย็นขึ้น

ในเวลาเดียวกันเมื่อมองจังชั่นเอ (A) เราจ่ายกระแสไฟฟ้าตามวงจรในรูปที่ 2.5 ทำให้อิเล็กตรอนเกิดการไหลจากสารตัวกลางที่มีความหนาแน่นของอิเล็กตรอนมากมายังสารที่มีความ

หนาแน่น น้อยกว่าทั้งนี้เพื่อครบศักย์กับการดึงประจุไฟฟ้าของแบตเตอรี่ศักย์บวกนั่นเอง จังชั่นในส่วนนี้ จังชั่นเอ (A) จึงเกิดความร้อน



รูปที่ 2.5 แสดงการเกิดเพลเทียร์เอฟเฟ็กต์

จากการไหลของอิเล็กตรอนจากด้านที่มีความหนาแน่นน้อยไปยังด้านที่มีความหนาแน่นมากทำให้จังชั่นในด้านนั้นเกิดความเย็น ในขณะที่เมื่ออิเล็กตรอนไหลจากด้านที่มีความหนาแน่นมากไปยังด้านที่มีความหนาแน่นน้อยจังชั่นของด้านนั้นจะเกิดความร้อน

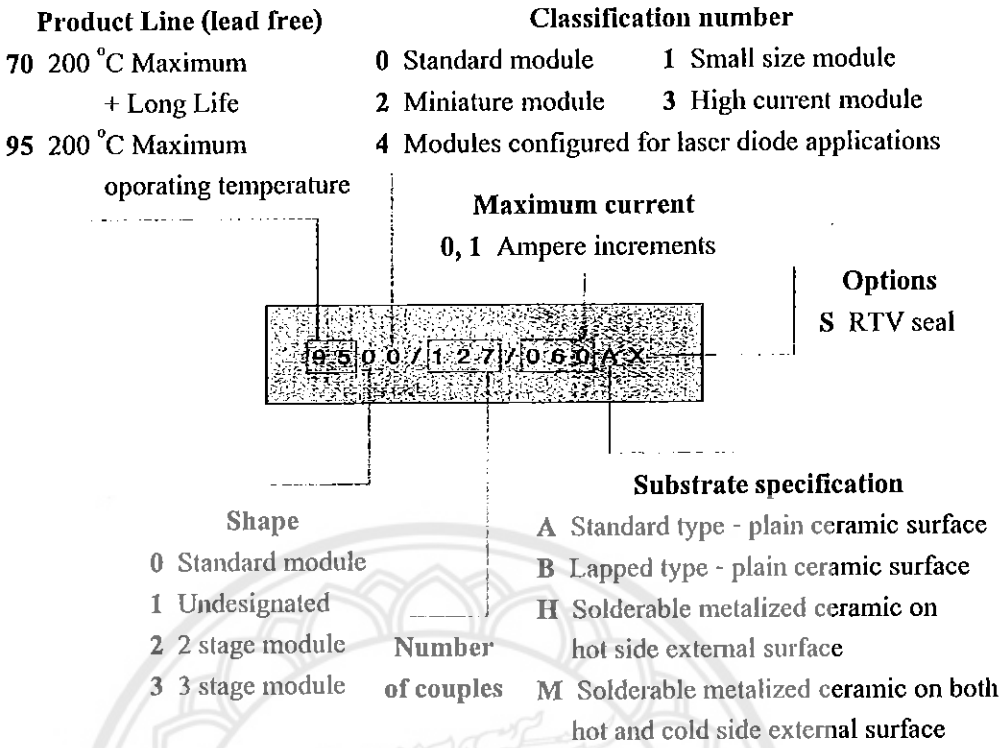
พิจารณาในรูปของวงจร หากใช้การไหลของอิเล็กตรอนเป็นหลัก กระแสอิเล็กตรอนไหลจากศักย์ลบไปหาศักย์บวกจึงไล่วงจรไฟฟ้าได้ว่า แบตเตอรี่คือแหล่งจ่ายและเป็นตัวจ่ายเริ่มต้น จ่ายแรงดันไฟศักย์ลบไปยังโลหะชนิด A ทางด้านขวามือ กระแสไหลผ่านจังชั่นบี (B) ไปยังโลหะชนิด B ที่มีความหนาแน่นสูงไหลผ่านโลหะชนิด B ที่อยู่กลาง ผ่านจังชั่นเอ (A) เพื่อไปยังโลหะชนิดเอ (A) ที่อยู่ทางซ้าย ครบวงจรกับขั้วแบตเตอรี่

จังชั่นเอ (A) เกิดความร้อนขึ้นอย่างรวดเร็วเพราะเกิดแรงเคลื่อนที่ไฟฟ้าเทอร์โมอิเล็กตริกซึ่งการไหลจากสารความหนาแน่นมากไปยังสารที่มีความหนาแน่นน้อยกว่า

จังชั่นบี (B) เกิดความเย็นขึ้นมาอย่างรวดเร็วเพราะเป็นการไหลกลับกับกรณีแรก เป็นการไหลของกระแสจากสารที่มีความหนาแน่นน้อยไปยังสารที่มีความหนาแน่นมากเกิดเป็นปรากฏการณ์เพลเทียร์หรือเพลเทียร์เอฟเฟ็กต์

2.5.2 เบอร์ของเพลเทียร์ หรือ TEC

เบอร์ที่นิยมยกตัวอย่างเช่น TEC ของบริษัท เฟอร์โรเท็ค โดยมีรายละเอียดดังแสดงในรูปที่ 2.6



รูปที่ 2.6 ความหมายของรหัสเบอร์ของเพลเทียร์

โดยรหัสเบอร์ แบ่งออกเป็น 3 กลุ่ม

กลุ่มที่ 1 เป็นการบอกซีรี่ ว่าเป็นซีรี่ส์ 63xx, 70xx หรือ 95xx ตัวเลขที่ตามมา 2 ตัวจะบอกรูปร่างและขนาดว่าเพลเทียร์เป็นชนิดเล็ก-กลาง-กระแสสูง-กระแสต่ำ

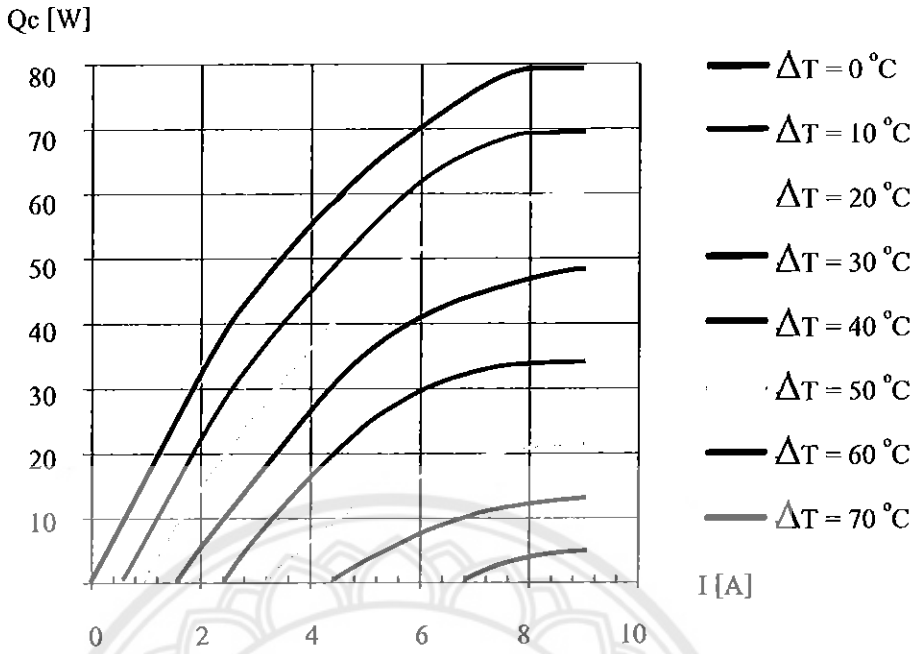
กลุ่มที่ 2 เป็นการระบุถึงจำนวนของคัปเปิ้ลหรือตัวลูกเต๋า เช่น

- ตัวเลข 127 หมายถึง มี 127 คัปเปิ้ล
- ตัวเลข 023 หมายถึง มี 23 คัปเปิ้ล

กลุ่มที่ 3 เป็นการบอกค่ากระแส 10 เท่า จำนวนตัวเลข 3 หลัก ส่วนอักษรตัวท้ายจะมีหรือไม่มีก็ได้ เช่น

- ตัวเลข 060 หมายถึง ต้องการกระแสสูงสุด 6 แอมป์

เพลเทียร์มีค่าความต่างอุณหภูมิเฉลี่ยอยู่ที่ 65 องศาเซลเซียส ค่าความต่างอุณหภูมิ ΔT โดยค่าดังกล่าวนี้มีความสัมพันธ์ระหว่างความสามารถในการปั๊มความร้อน (Qc) ซึ่งมีหน่วยเป็นวัตต์กับกระแส (I) ที่จ่ายด้านอินพุต ความสัมพันธ์นี้แสดงด้วยกราฟดัง รูปที่ 2.7



รูปที่ 2.7 ความสัมพันธ์ระหว่างความสามารถการปั๊มความร้อน (Qc) กับกระแส (I) ที่จ่ายเพลเทียร์

2.5.3 จุดอ่อนของเพลเทียร์

แม้ว่าเพลเทียร์จะเป็นอุปกรณ์ที่ทนทานต่อการใช้งานได้ยาวนาน แต่เพลเทียร์จะเกิดการเสียหายเมื่อทำงานกับแรงดันไฟฟ้ากระโชกหรือเมื่อเกิดภาวะสวิตชิ่ง เพราะเพลเทียร์มีแรงกดจากไฟฟ้ากระแสตรง (DC) เพื่อทำให้เกิดภาวะทางอุณหภูมิ (Thermal stress) จึงทำให้เกิดภาวะช็อตขึ้นระหว่างเพลตของทั้ง 2 ด้านซึ่งมีความต่างของอุณหภูมิมากอยู่แล้ว และหากจำเป็นต้องใช้งานด้วยการควบคุมอุณหภูมิด้วยการจ่ายไฟแบบสวิตชิ่ง ทำได้โดยการให้สวิตชิ่งทำงานด้วยความถี่มากกว่า 2 กิโลเฮิรตซ์ ในการเปลี่ยนดิวตี้ไซเคิล

2.6 ไมโครคอนโทรลเลอร์

ไมโครคอนโทรลเลอร์ (อังกฤษ: Microcontroller) คือ อุปกรณ์ควบคุมขนาดเล็ก ซึ่งบรรจุความสามารถที่คล้ายคลึงกับระบบคอมพิวเตอร์ โดยในไมโครคอนโทรลเลอร์ได้รวมเอาซีพียู หน่วยความจำและพอร์ตซึ่งเป็นส่วนประกอบหลักสำคัญของระบบคอมพิวเตอร์เข้าไว้ด้วยกัน โดยทำการบรรจุเข้าไว้ในตัวถังเดียวกัน

โครงสร้างโดยทั่วไปของไมโครคอนโทรลเลอร์นั้นสามารถแบ่งออกมาได้เป็น 5 ส่วนใหญ่ๆ ดังต่อไปนี้

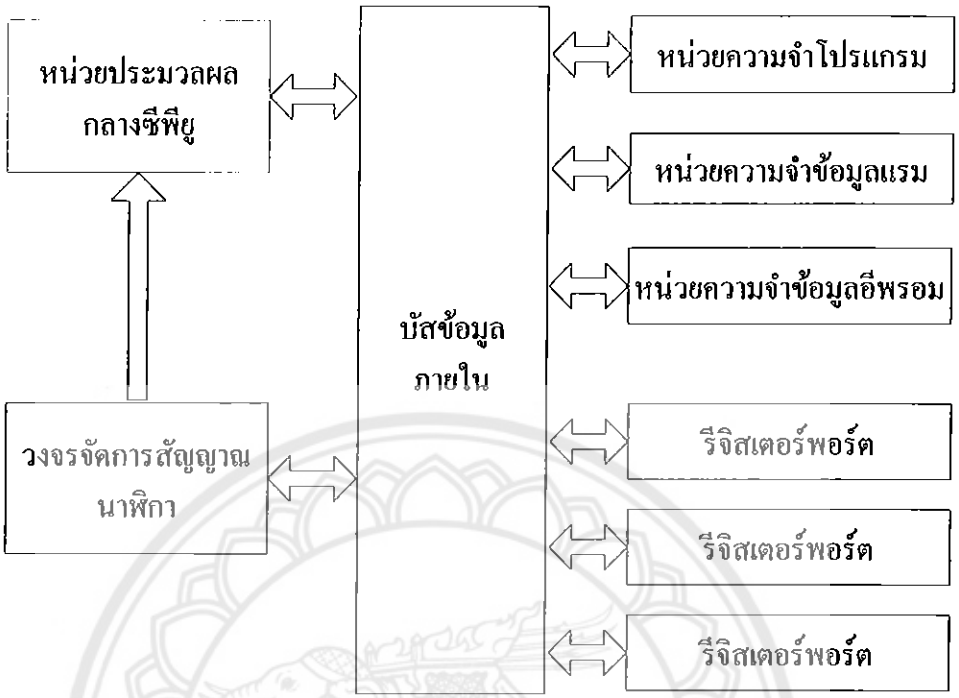
1) หน่วยประมวลผลกลางหรือซีพียู (CPU: Central Processing Unit)

2) หน่วยความจำ (Memory) สามารถแบ่งออกเป็น 2 ส่วน คือ หน่วยความจำที่มีไว้สำหรับเก็บโปรแกรมหลัก (Program Memory) เปรียบเสมือนฮาร์ดดิสก์ของเครื่องคอมพิวเตอร์ตั้งโต๊ะ คือข้อมูลใดๆ ที่ถูกเก็บไว้ในนี้จะไม่สูญหายไปแม้ไม่มีไฟเลี้ยง อีกส่วนหนึ่งคือหน่วยความจำข้อมูล (DataMemory) ใช้เป็นเหมือนกระดาษทดในการคำนวณของซีพียู และเป็นที่พักข้อมูลชั่วคราวขณะทำงาน แต่หากไม่มีไฟเลี้ยง ข้อมูลก็จะหายไปคล้ายกับหน่วยความจำแรม (RAM) ในเครื่องคอมพิวเตอร์ทั่วๆ ไป แต่สำหรับไมโครคอนโทรลเลอร์สมัยใหม่ หน่วยความจำข้อมูลจะมีทั้งที่เป็นหน่วยความจำแรม ซึ่งข้อมูลจะหายไปเมื่อไม่มีไฟเลี้ยง และมีหน่วยความจำรอมเป็นอีอีพรอม (EEPROM : Erasable Electrically Read-Only Memory) ซึ่งสามารถเก็บข้อมูลได้แม้ไม่มีไฟเลี้ยง

3) ส่วนติดต่อกับอุปกรณ์ภายนอก หรือพอร์ต (Port) มี 2 ลักษณะคือ พอร์ตรับสัญญาณหรือพอร์ตอินพุต (Input Port) และพอร์ตส่งสัญญาณหรือพอร์ตเอาท์พุต (Output Port) ส่วนนี้จะใช้ในการเชื่อมต่อกับอุปกรณ์ภายนอก ถือว่าเป็นส่วนที่สำคัญมาก ใช้ร่วมกันระหว่างพอร์ตอินพุต เพื่อรับสัญญาณ อาจจะด้วยการกดสวิตช์ เพื่อนำไปประมวลผลและส่งไปพอร์ตเอาท์พุต เพื่อแสดงผล เช่น การติดดับของหลอดไฟ เป็นต้น

4) ช่องทางเดินของสัญญาณ หรือบัส (Bus) คือเส้นทางการแลกเปลี่ยนสัญญาณข้อมูลระหว่าง ซีพียู หน่วยความจำและพอร์ต เป็นลักษณะของสายสัญญาณ จำนวนมากอยู่ภายในตัวไมโครคอนโทรลเลอร์ โดยแบ่งเป็นบัสข้อมูล (Data Bus), บัสแอดเดรส (Address Bus) และบัสควบคุม (Control Bus)

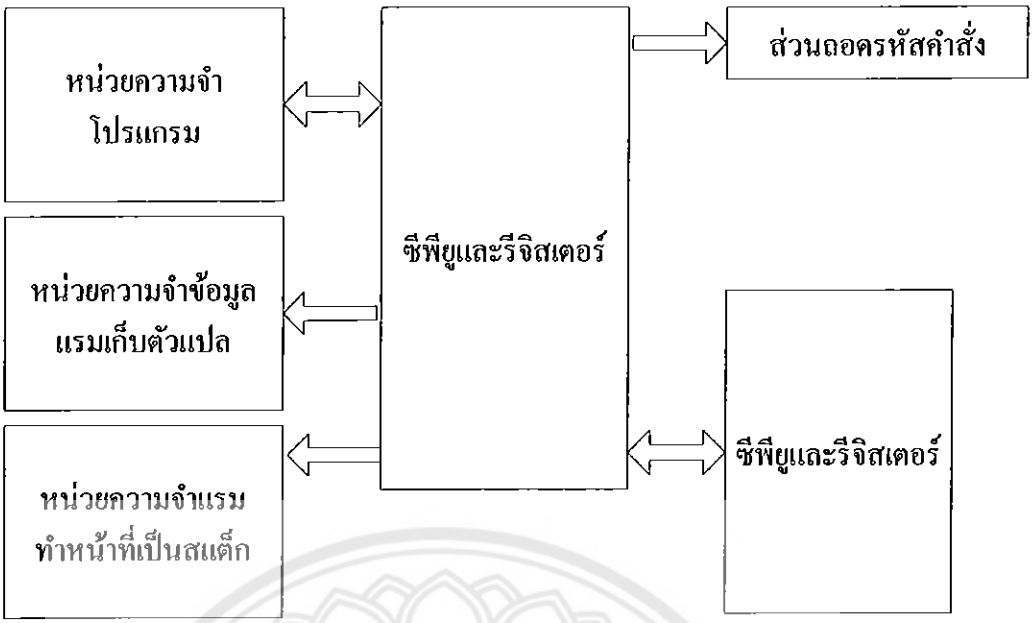
5) วงจรกำเนิดสัญญาณนาฬิกา นับเป็นส่วนประกอบที่สำคัญมากอีกส่วนหนึ่ง เนื่องจากการทำงานที่เกิดขึ้นในตัวไมโครคอนโทรลเลอร์ จะขึ้นอยู่กับการกำหนดจังหวะ หากสัญญาณนาฬิกามีความถี่สูงจังหวะการทำงานก็จะสามารถทำได้ถี่ขึ้นส่งผลให้ไมโครคอนโทรลเลอร์ตัวนั้นมีความเร็วในการประมวลผลสูงตามไปด้วยเนื่องด้วยการควบคุมในโครงงานนี้จะใช้การควบคุมระบบจากไมโครคอนโทรลเลอร์จึงขอกล่าวเกี่ยวกับระบบและทฤษฎีของไมโครคอนโทรลเลอร์ดังนี้



รูปที่ 2.8 โครงสร้างและส่วนประกอบหลักเบื้องต้นของไมโครคอนโทรลเลอร์

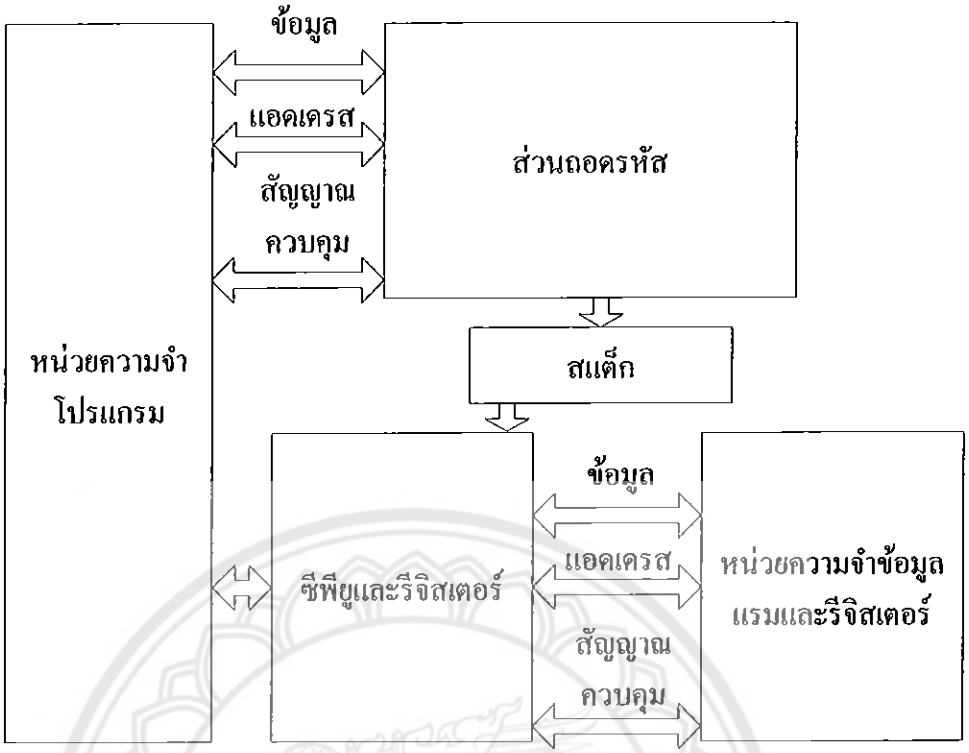
2.6.1 สถาปัตยกรรมของไมโครคอนโทรลเลอร์

เป็นที่ยอมรับกันแพร่หลายว่าสถาปัตยกรรมของไมโครคอนโทรลเลอร์มีด้วยกัน 2 แบบ คือ พรินซ์ตัน (Princeton) หรือฟอนนิวแมน (Von Neumann) และฮาร์วาร์ด (Harvard) ในภาพที่ 2.8 และ 2.9 แสดงการจัดสรรหน่วยความจำและรีจิสเตอร์ในสถาปัตยกรรมของไมโครคอนโทรลเลอร์ทั้งสองแบบพิจารณาในภาพที่ 2.8 เป็นการจัดสรรในสถาปัตยกรรมแบบพรินซ์ตันจะเห็นได้ว่ามีโครงสร้างที่เรียบง่ายไม่ซับซ้อนส่วนของหน่วยความจำโปรแกรมกับหน่วยความจำข้อมูลจะได้รับการจัดสรรให้อยู่รวมกันติดต่อกับซีพียูผ่านส่วนจัดการเชื่อมต่อหน่วยความจำและภายในซีพียูจะมีรีจิสเตอร์บรรจุอยู่ข้อดีของสถาปัตยกรรมแบบนี้คือออกแบบง่ายเพราะหน่วยความจำทั้งหมดอยู่รวมกันสามารถเข้าถึงได้ง่ายหน่วยความจำมีขนาดใหญ่เพียงพอที่จะเก็บได้ทั้งโปรแกรมควบคุมการทำงานและข้อมูลของตัวแปรในการประมวลผลข้อด้อยของสถาปัตยกรรมนี้คือความเร็วในการประมวลผลเนื่องจากหน่วยความจำอยู่รวมกันจึงต้องติดต่อหน่วยความจำโปรแกรมสลับกับหน่วยความจำข้อมูลส่งผลให้ซีพียูต้องใช้จำนวนไซเคิลในการทำงานมาก



รูปที่ 2.9 โครงสร้างสถาปัตยกรรมของไมโครคอนโทรลเลอร์แบบพรินซ์ตัน**หรือฟอนนิวแมน**

ในขณะที่สถาปัตยกรรมแบบฮาร์วาร์ดซึ่งจะแสดงในภาพที่ 2.9 **โดยจะแยกส่วนของหน่วยความจำข้อมูลและรีจิสเตอร์ออกจากหน่วยความจำโปรแกรมทำให้ไซเคิลการทำงานลดลง เนื่องจากสามารถติดต่อหน่วยความจำโปรแกรมและหน่วยความจำข้อมูลได้เร็วกว่านอกจากนั้นในสถาปัตยกรรมนี้ในขณะที่ซีพียูกำลังเอ็กซิคิวต์คำสั่งปัจจุบันอยู่สามารถที่จะเฟตช์คำสั่งถัดไปได้ยิ่งทำให้ไมโครคอนโทรลเลอร์ทำงานได้เร็วขึ้น**



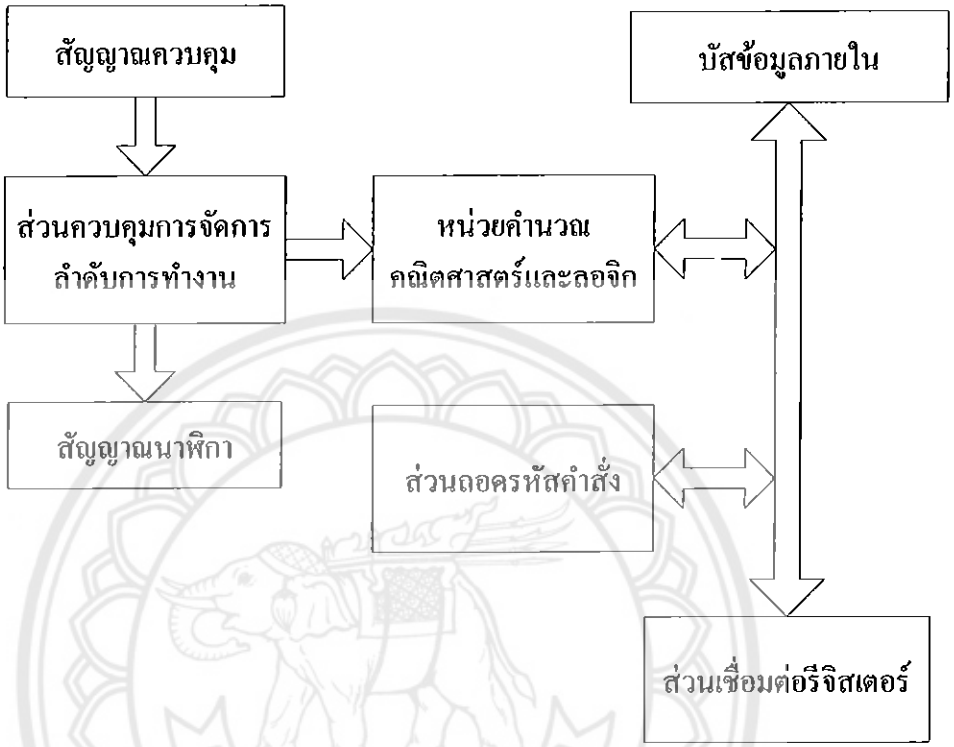
รูปที่ 2.10 โครงสร้างสถาปัตยกรรมของไมโครคอนโทรลเลอร์แบบฮาร์วาร์ด

2.6.2 หน่วยประมวลผลกลางหรือซีพียู

ซีพียูเป็นเสมือนมันสมองของไมโครคอนโทรลเลอร์ซึ่งทำหน้าที่ประมวลผลข้อมูลที่เข้ามาในระบบแล้วทำการส่งต่อไปยังส่วนต่างๆเพื่อควบคุมการทำงานต่อไปในภาพที่ 2.10 แสดงส่วนประกอบพื้นฐานของซีพียูในไมโครคอนโทรลเลอร์ทั่วๆไปจะเห็นได้ว่าหัวใจหลักของซีพียูคือหน่วยคำนวณทางคณิตศาสตร์และลอจิก (ALU: Arithmetic and Logic Unit) ซึ่งได้รับการกำหนดจังหวะการทำงานจากส่วนควบคุมลำดับการทำงานโดยจังหวะการทำงานนั้นจะสัมพันธ์กับสัญญาณนาฬิกาเมื่อซีพียูทำการติดต่อหน่วยความจำสิ่งที่ปรากฏขึ้นบนบัสข้อมูลภายในซีพียูคือรหัสคำสั่ง (Instruction Code) ซึ่งต้องผ่านการทำงานของส่วนถอดรหัสคำสั่ง (Instruction Decoder) เสียก่อนจะได้เป็นข้อมูลคำสั่งที่ซีพียูเข้าใจและสามารถดำเนินการต่อได้หลังจากที่หน่วยคำนวณทางคณิตศาสตร์และลอจิกประมวลผลแล้วก็ส่งข้อมูลมายังส่วนเชื่อมต่อรีจิสเตอร์ภายในซีพียูเพื่อติดต่อกับส่วนอื่นๆต่อไป

การทำงานของซีพียูมีด้วยกัน 2 จังหวะเฟตช์ (Fetch) และเอ็กซิคิวต์ (Executed) โดยการทำงานจะเริ่มจากการเฟตช์คือการเรียกหรือการเข้าถึงคำสั่งแล้วทำการถอดรหัสเป็นภาษาเครื่องเพื่อ

เตรียมการประมวลผลจากนั้นจะเป็นจังหวะของการเอ็กซิคิวต์คือการกระทำตามคำสั่งที่กำหนดให้จนเสร็จสิ้น



รูปที่ 2.11 ส่วนประกอบหลักของซีพียูในไมโครคอนโทรลเลอร์

2.6.3 หน่วยความจำ

ภายในไมโครคอนโทรลเลอร์จะประกอบไปด้วยหน่วยความจำ 3 แบบ คือ หน่วยความจำโปรแกรม (Program Memory) หน่วยความจำข้อมูลแรมและหน่วยความจำข้อมูลอีอีพรอม

1) หน่วยความจำโปรแกรม

หน่วยความจำโปรแกรมเป็นที่สำหรับเก็บข้อมูลคำสั่งของโปรแกรมควบคุมที่ผู้พัฒนาเขียนขึ้นหรือเรียกว่าโปรแกรมมอนิเตอร์ (Monitor Program) ซีพียูจะเข้ามาติดต่อเพื่ออ่านข้อมูลรหัสคำสั่งจากหน่วยความจำในส่วนนี้แล้วนำไปประมวลผลเพื่อควบคุมการทำงานของระบบทั้งหมดต่อไปหน่วยความจำโปรแกรมนี้มักมีขนาดใหญ่ยิ่งมีขนาดมากเท่าใดก็จะสามารถบรรจุ

โปรแกรมที่มีความซับซ้อนหรือสามารถเก็บตารางข้อมูลที่ใช้ในการประมวลผลได้มากตามไปด้วย โดยทั่วไปมีความจุไม่น้อยกว่า 512 ไบต์ขนาดของหน่วยความจำโปรแกรมจะแปรตามความก้าวหน้าทางเทคโนโลยีชนิดของหน่วยความจำโปรแกรมที่ใช้ในไมโครคอนโทรลเลอร์ที่นิยมมีด้วยกันอยู่3 แบบ คือ แบบอีพรอม (EPROM Erasable Programmable Read-Only Memory) แบบอีอีพรอม (EEPROM: Electrically Erasable Programmable Read-Only Memory) และแบบแฟลช (Flash Memory) ความแตกต่างของทั้ง 3 แบบอยู่ที่จำนวนครั้งในการลบและเขียนข้อมูลทับลงไปใหม่โดยสามารถสรุปได้ดังนี้แบบอีพรอมแบ่งออกเป็น 2 แบบคือแบบโปรแกรมได้หลายครั้งและแบบโปรแกรมได้เพียงครั้งเดียว โดยถ้าหากเป็นแบบโปรแกรมได้หลายครั้งนั้นบนตัวถังของไมโครคอนโทรลเลอร์จะมีหน้าต่างกระจกติดอยู่สามารถมองเห็นชิปภายในได้เวลาลบข้อมูลต้องลบด้วยแสงอัลตร้าไวโอเลตจำนวนรอบในการโปรแกรมใหม่อยู่ระหว่าง 10-100 ครั้งแต่ถ้าเป็นแบบโปรแกรมได้ครั้งเดียว (One–Time Programmable, OTP) จะไม่สามารถลบได้ตัวถังของมันจะปิดมิดชิดเหมือนกับไอซีธรรมดาแบบอีอีพรอมหน่วยความจำแบบนี้จะลบและเขียนใหม่ได้ด้วยสัญญาณไฟฟ้าในอดีตเป็นที่นิยมมากเนื่องจากสามารถเขียนใหม่ได้เป็นหลักร้อยรอบขึ้นไปในบางตระกูลถึง 1 ล้านครั้งแต่ในปัจจุบันไม่เป็นที่นิยมใช้ในไมโครคอนโทรลเลอร์เนื่องจากต้นทุนสูง แบบแฟลชหน่วยความจำโปรแกรมชนิดนี้สามารถลบและเขียนได้ด้วยสัญญาณไฟฟ้าแตกต่างกับแบบอีอีพรอมในเชิงการใช้งานตรงที่กระบวนการลบข้อมูลหน่วยความจำโปรแกรมแบบแฟลชจะไม่สามารถเลือกลบเฉพาะเจาะจงบางตำแหน่งได้เมื่อทำการลบข้อมูลจะต้องลบทั้งหมด หน่วยความจำแบบนี้ได้รับความนิยมมากเนื่องจากราคาไม่สูงและสามารถโปรแกรมได้เป็นร้อยครั้งขึ้นไปในบางรุ่นสูงเป็นหมื่นครั้งและเป็นแสนครั้งขึ้นอยู่กับแรงดันที่ใช้ในโปรแกรม

2) หน่วยความจำข้อมูลแรม

เป็นหน่วยความจำที่ต้องมีในไมโครคอนโทรลเลอร์ทุกตัวเพราะใช้เป็นพื้นที่สำหรับเก็บข้อมูลทั้งในระหว่างและหลังการประมวลผลยิ่งมีมากยิ่งช่วยในการทำงานสะดวกขึ้นเพราะหน่วยความจำแรมมีอัตราเร็วในการอ่านเขียนสูงมากและไม่จำกัดจำนวนรอบในการอ่านเขียนในพื้นที่ของหน่วยความจำข้อมูลแรมจะแบ่งออกเป็น 2 ส่วนคือส่วนของข้อมูลทั่วไปสำหรับเก็บค่าตัวแปรและส่วนของรีจิสเตอร์โดยปกติแล้วหน่วยความจำข้อมูลแรมจะมีความจุไม่มากเมื่อเทียบกับหน่วยความจุโปรแกรมในบางตัวอยู่ในหลักสิบไบต์แต่ถ้าไมโครคอนโทรลเลอร์มีความสามารถสูงขึ้นความจุของหน่วยความจำข้อมูลแรมก็เพิ่มมากขึ้นตามทั้งนี้เพราะต้องเพิ่มในส่วนของรีจิสเตอร์ตามความสามารถที่สูงขึ้นของไมโครคอนโทรลเลอร์

3) หน่วยความจำข้อมูลอีอีพรอม

เป็นหน่วยความจำข้อมูลพิเศษที่ในไมโครคอนโทรลเลอร์บางเบอร์บางรุ่นบางตระกูลใช้สำหรับเก็บข้อมูลที่ต้องการรักษาไว้เมื่อไม่มีการจ่ายไฟเลี้ยงให้แก่ไมโครคอนโทรลเลอร์การติดต่อเพื่ออ่านจะมีลักษณะพิเศษขึ้นอยู่กับไมโครคอนโทรลเลอร์แต่ละเบอร์ขนาดของหน่วยความจำแบบนี้มักเท่ากับ 8 บิตส่วนความจุก็จะแตกต่างกันไปตั้งแต่ไม่กี่สิบไบต์จนถึงเป็นกิโลไบต์การอ่านเขียนหน่วยความจำแบบนี้จะใช้สัญญาณไฟฟ้าทั้งหมดและสามารถรักษาข้อมูลล่าสุดไว้แม้ว่าจะไม่มีการจ่ายไฟเลี้ยงให้แก่ไมโครคอนโทรลเลอร์แล้วก็ตามสำหรับจำนวนรอบในการเขียนโดยปกติอยู่ในหลักล้านครั้งขึ้นไป

2.6.4 หน่วยความจำพิเศษ

เป็นหน่วยความจำที่ใช้เก็บข้อมูลพิเศษไม่ได้ใช้สำหรับเก็บชุดคำสั่งของโปรแกรมหรือข้อมูลอื่นๆมีหลายชนิดแตกต่างกันขึ้นอยู่กับตระกูลของไมโครคอนโทรลเลอร์ได้แก่

1) รีจิสเตอร์

เป็นหน่วยความจำพิเศษที่มีบทบาทสูงมากในการทำงานของไมโครคอนโทรลเลอร์เพื่อให้ไมโครคอนโทรลเลอร์สามารถที่จะอ่านและเขียนข้อมูลได้ตลอดเวลาจนกว่าจะหยุดจ่ายไฟเลี้ยงให้แก่ไมโครคอนโทรลเลอร์ ส่วนหน้าที่หลักของรีจิสเตอร์ คือ ใช้เก็บข้อมูลในการทำงานของไมโครคอนโทรลเลอร์โดยข้อมูลที่เก็บนี้มีทั้งข้อมูลแสดงสถานะการทำงาน, ข้อมูลสำหรับควบคุมการทำงานโมดูลย่อยต่างๆภายในไมโครคอนโทรลเลอร์, ข้อมูลที่รับเข้ามาจากพอร์ตอินพุตหรือข้อมูลที่ต้องส่งออกไปยังอุปกรณ์ภายนอกผ่านพอร์ตเอาท์พุตโดยข้อมูลแต่ละประเภทก็จะจัดเก็บลงในรีจิสเตอร์ที่แตกต่างกันตามหน้าที่การทำงานหน่วยความจำที่นำมาใช้ทำรีจิสเตอร์มีด้วยกัน 2 ลักษณะขึ้นอยู่กับสถาปัตยกรรมของไมโครคอนโทรลเลอร์หากเป็นแบบพรินซ์ตันรีจิสเตอร์จะมีอยู่ด้วยกัน 2 ส่วนส่วนแรกจะอยู่ร่วมกันกับซีพียูหรือเรียกว่ารีจิสเตอร์ซีพียูส่วนที่สองจะอยู่แยกต่างหากซึ่งมักเป็นรีจิสเตอร์ควบคุมพอร์ตอินพุตเอาท์พุตและรีจิสเตอร์แสดงสถานะแต่ในสถาปัตยกรรมแบบฮาร์วาร์ดนั้นจะใช้เพียงบางส่วนในหน่วยความจำข้อมูลแรมภายในไมโครคอนโทรลเลอร์

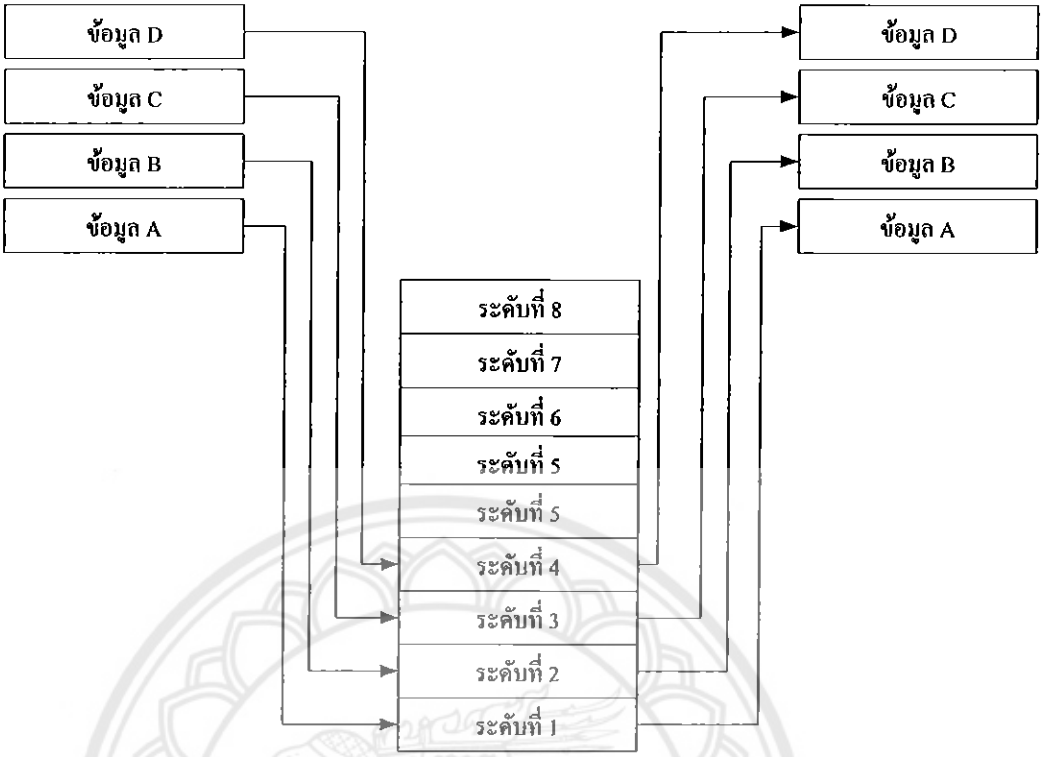
2) โปรแกรมเคาน์เตอร์

การที่ซีพียูสามารถติดต่อกับหน่วยความจำโปรแกรมเพื่ออ่านข้อมูลคำสั่งได้อย่างถูกต้องเป็นผลมาจากรีจิสเตอร์หน้าที่พิเศษตัวหนึ่งคือรีจิสเตอร์ตัวนับโปรแกรมหรือโปรแกรมเคาน์เตอร์ (Program Counter, PC) โดยโปรแกรมเคาน์เตอร์จะเป็นตัวชี้ตำแหน่งแอดเดรสของหน่วยความจำ

โปรแกรมที่ซีพียูจะต้องไปกระทำในลำดับถัดไปโดยปกติแล้วค่าของโปรแกรมเคาน์เตอร์จะเปลี่ยนแปลงโดยอัตโนมัติขึ้นอยู่กับผลการทำงานที่เกิดขึ้นในไมโครคอนโทรลเลอร์บางตระกูลสามารถเข้าถึงโปรแกรมเคาน์เตอร์เพื่อทำการอ่านเขียนได้แต่ในบางตระกูลก็ไม่สามารถทำได้

3) สแต็ก

สแต็ก (Stack) เป็นหน่วยความจำพิเศษที่ไมโครคอนโทรลเลอร์ทุกตัวต้องมีโดยหน้าที่ของมันคือเก็บข้อมูลที่ยังต้องการอยู่ของรีจิสเตอร์และเมื่อข้อมูลนั้นถูกนำมาเก็บไว้ในสแต็กแล้วก็สามารถที่จะเปลี่ยนข้อมูลในรีจิสเตอร์ตัวนั้นๆได้ทันทีหลังจากที่ทำงานเรียบร้อยแล้วจึงกลับมาอ่านข้อมูลเดิมกลับคืนจากสแต็กซึ่งมีกระบวนการทำงานดังในภาพที่ 2.11 การเก็บข้อมูลของสแต็กจะมีลักษณะเป็นระดับหรือเป็นชั้นข้อมูลที่จะเก็บเข้ามาก่อนและจะต้องอ่านออกทีหลังหรือเป็นแบบ FILO (First In Last Out) และจำนวนระดับหรือจำนวนชั้นของสแต็กก็จะมีจำนวนจำกัดในไมโครคอนโทรลเลอร์ส่วนใหญ่จะมีความจุของสแต็กไม่น้อยกว่า 8 ระดับการที่ยิ่งมีขนาดของสแต็กมากหรือมีจำนวนระดับมากก็จะยิ่งช่วยให้การทำงานสะดวกขึ้นเพราะในการประมวลผลมีโอกาสมากที่จะต้องพักข้อมูลในรีจิสเตอร์หลักเพื่อไปทำงานอื่นก่อนหลังจากนั้นจึงกลับมาทำงานต่อโดยเฉพาะอย่างยิ่งกับงานที่มีการอินเทอร์รัพต์หรือขัดจังหวะซีพียูอยู่บ่อยๆรวมถึงการกระโดดไปทำงานที่โปรแกรมย่อยที่มีความต้องการเขียนข้อมูลลงในรีจิสเตอร์ตัวเดียวกันนี้หลังจากทำงานแล้วจึงกลับมาที่โปรแกรมหลักแล้วอ่านค่าเดิมก่อนหน้านี้ และจึงกลับมาทำงานต่อว่างานบางลักษณะการกระโดดไปทำงานยังโปรแกรมย่อยซ้อนกัน 2-3 ชั้นทำให้ต้องมีการเก็บข้อมูลไว้ในสแต็กมากขึ้นหากความจุของสแต็กมีน้อยก็จะไม่สามารถรองรับการทำงานในลักษณะนี้ได้ขนาดของสแต็กโดยปกติจะต้องเท่ากับขนาดของรีจิสเตอร์ตัวนับโปรแกรมหรือ PC เพราะมีโอกาสที่จะต้องเก็บค่าของ PC ไว้ในสแต็กด้วย



รูปที่ 2.12 กลไกการทำงานของสแต็กอย่างง่าย

2.6.5 การทำงานของไมโครคอนโทรลเลอร์

ไมโครคอนโทรลเลอร์จะสามารถทำงานได้เมื่อจ่ายไฟเลี้ยงและต่อวงจรกำเนิดสัญญาณนาฬิกาให้แก่มันจากนั้นซีพียูภายในไมโครคอนโทรลเลอร์จะติดต่อกับหน่วยความจำโปรแกรมเพื่ออ่านข้อมูลคำสั่งทำงานตามคำสั่งที่บรรจุอยู่ในหน่วยความจำโปรแกรม นั่นหมายความว่า ต้องมีการเขียนข้อมูลลงในหน่วยความจำโปรแกรมก่อนโดยไมโครคอนโทรลเลอร์แต่ละเบอร์จะมีรูปแบบของข้อมูลคำสั่งที่แตกต่างกันโดยภาษาที่ใช้เขียนโปรแกรมสามารถแบ่งได้ 2 ระดับ คือ ภาษาระดับสูง (High Level Language) และภาษาแอสเซมบลี (Assembly Language) ซึ่งโดยปกติแล้วไมโครคอนโทรลเลอร์ต้องโปรแกรมด้วยภาษาแอสเซมบลี เนื่องจากสามารถทำงานได้รวดเร็วผ่านกระบวนการแปลงข้อมูลคำสั่งเป็นเลขฐานสิบหกเพื่อทำงานตามคำสั่งเพียง 1 ขั้นตอนคือแปลงจากภาษาแอสเซมบลีเป็นข้อมูลฐานสิบหกที่เรียกว่าออปโค้ด (Opcode) แต่ข้อเสียของการเขียนภาษาแอสเซมบลีคือผู้เขียนต้องทำความเข้าใจในชุดคำสั่งของไมโครคอนโทรลเลอร์เบอร์นั้นๆอย่างถ่องแท้และเมื่อเปลี่ยนเบอร์ไมโครคอนโทรลเลอร์ก็จะต้องทำการเรียนรู้และทำความเข้าใจชุดคำสั่งใหม่ซึ่งอาจทำให้เสียเวลามากรวมทั้งการเขียนโปรแกรมด้วยภาษาแอสเซมบลีผู้เขียนต้องมีทักษะในการเขียนโปรแกรมสูงพอสมควรและเข้าใจถึงสถาปัตยกรรมของไมโครคอนโทรลเลอร์เป็น

อย่างดีในขณะที่การเขียนโปรแกรมด้วยภาษาระดับสูงอาทิเช่นภาษาซีภาษาเบสิกต้องผ่านกระบวนการที่เรียกว่าคอมไพล์ (Compile) เพื่อแปลงภาษาระดับสูงเหล่านั้นเป็นภาษาเครื่องหรือ ออปโค้ดของไมโครคอนโทรลเลอร์เบอร์นั้นๆเสียก่อนเมื่อใช้เครื่องมือทางซอฟต์แวร์ตัวนี้ทำให้ผู้เขียนโปรแกรมอาจไม่จำเป็นต้องศึกษาสถาปัตยกรรมและชุดคำสั่งของไมโครคอนโทรลเลอร์เบอร์นั้นๆอย่างลึกซึ้งเท่ากับการเขียนภาษาแอสเซมบลีทั้งนี้เพราะคอมไพเลอร์จะทำในส่วนนี้แทน ดังนั้นเมื่อผู้ใช้งานเปลี่ยนเบอร์ไมโครคอนโทรลเลอร์ก็เพียงจัดหาโปรแกรมคอมไพเลอร์ที่เหมาะสมมาใช้งานและศึกษาสถาปัตยกรรมของไมโครคอนโทรลเลอร์เบอร์ใหม่อีกเพียงเล็กน้อยก็สามารถใช้งานได้แต่ข้อเสียของการใช้คอมไพเลอร์คือราคาแพงมาก

2.7 ไมโครคอนโทรลเลอร์ตระกูล AVR

ไมโครคอนโทรลเลอร์ AVR ผลิตจากบริษัท ATMEL ซึ่งในบ้านเราเองนิยมนำมาใช้งานกันอย่างกว้างขวางซึ่งส่วนประกอบ หรือสถาปัตยกรรมภาพรวมของไมโครคอนโทรลเลอร์ตระกูลนี้โดยเบอร์ต่างๆของไอซีจะมีสถาปัตยกรรมแต่ละตัวแตกต่างกันออกไป ซึ่งบางตัวมีให้ใช้งานบางตัวไม่มีให้ใช้งาน หรือ ตัวใดมีมากกว่าตัวใด สำหรับไมโครคอนโทรลเลอร์ตัวนั้นๆด้วยคุณสมบัติต่างๆของไมโครคอนโทรลเลอร์ที่เราควรทราบ เช่น ความถี่สูงสุดในการทำงาน หน่วยความจำประเภทต่างๆมีค่าเท่าไร แรงดันในการทำงาน จำนวนพอร์ตอินพุตและเอาท์พุต ตอบสนองการอินเทอร์รัพท์ได้กี่ครั้ง เป็นต้น

2.7.1 โครงสร้างและสถาปัตยกรรมของ ATmega 64

ไมโครคอนโทรลเลอร์ได้มีการคิดค้นและพัฒนามาอย่างต่อเนื่องเพื่อให้มีศักยภาพในการทำงานสูงขึ้นไมโครคอนโทรลเลอร์ตระกูล AVR ของบริษัท ATMEL เป็นไมโครคอนโทรลเลอร์ที่มีฟังก์ชันการใช้งานต่างๆมากมาย เช่น โมดูล Analog to Digital, Timer/Counter, USART, SPI และจัดเป็นไมโครคอนโทรลเลอร์ตระกูลใหม่ที่มีสถาปัตยกรรมแบบ RISC (Advanced RISC architecture) ซึ่งส่วนต่างๆ เหล่านี้จะถูกสร้างรวมอยู่ภายในซีพียูเพียงตัวเดียวทำให้สามารถทำงานได้หลายๆอย่างและสามารถลดในส่วนของฮาร์ดแวร์บางอย่างลงส่วนในเรื่องของความเร็วซีพียูตระกูลนี้จะใช้เวลาในการกระทำคำสั่งต่างๆ เพียง 1 ไซเคิลต่อคำสั่งเท่านั้น ทำให้มีความเร็วในการทำงานมากกว่าซีพียูทั่วไป (ที่ความถี่เดียวกัน) และมีสถาปัตยกรรมภายในดังรูปที่ 2.13 คุณสมบัติต่างๆของไมโครคอนโทรลเลอร์ AVR สามารถสรุปอย่างคร่าวๆ ได้ดังนี้

- สถาปัตยกรรมภายในเป็นแบบ Advanced RISC (Reduce Instruction Set Computer)

- มีคำสั่งควบคุมการทำงานมากกว่า 100 คำสั่ง โดยมีความเร็วในการประมวลผล 1 คำสั่งต่อ 1 สัญญาณนาฬิกา (1 MIP/1MHz)

- มีรีจิสเตอร์ใช้งานทั่วไปขนาด 8 บิต จำนวน 32 ตัว (ทำให้สะดวกต่อการพัฒนาโปรแกรมด้วยภาษา C เป็นอย่างมาก)

- ความเร็วในการทำงาน 1 MIPS ต่อ 1 MHz และมากถึง 16 MIPS เมื่อใช้ความถี่ที่ 16 MHz (ความสามารถในการใช้งานความถี่สัญญาณนาฬิกาขึ้นอยู่กับเบอร์ที่เลือกใช้งาน)

- หน่วยความจำ ROM แบบ Flash (มีโหมดป้องกันหน่วยความจำ) ขนาด 64 กิโลไบต์ (เขียน/ลบ ได้ 10,000 ครั้ง)

- หน่วยความจำข้อมูลแบบ EEPROM (มีโหมดป้องกันหน่วยความจำ) ขนาด 2,048 ไบต์ (เขียน/ลบ) ได้ 100,000 ครั้ง)

- หน่วยความจำข้อมูลแบบ SRAM 4 กิโลไบต์

- ไทเมอร์/เคาน์เตอร์ทั้งแบบ 8 บิตและ 16 บิต พร้อมปรีสเกลเลอร์

- มีระบบตรวจสอบความผิดพลาดในการทำงานของซอฟต์แวร์ (Watchdog Timer with On-Chip Oscillator)

- โมดูลสร้างสัญญาณ PWM (Pulse Width Modulator) มีจำนวน 6 ช่อง

- มีโมดูลแปลงสัญญาณอนาลอกเป็นดิจิตอล (ADC) ขนาด 10 บิต มากถึง 8 ช่อง

- โมดูลเปรียบเทียบแรงดันอนาลอก (Analog Comparator)

- การสื่อสารข้อมูลอนุกรมโดยจะมีทั้งแบบ UART (Universal Asynchronous Receiver Transmitters) หรือแบบ RS232, SPI (Serial Peripheral Interface) และแบบ I^2C เป็นต้น

- พอร์ตอินพุตเอาท์พุตขึ้นอยู่กับเบอร์ AVR ที่เลือกใช้งาน มีตั้งแต่ 8 ขา จนมากกว่า 100 ขาพอร์ต (ATmega 64 มีขาพอร์ตอินพุตเอาท์พุต 53 ขา)

- ทำงานที่ไฟเลี้ยง 4.5 ถึง 5.5 โวลต์

- มีพอร์ต I/O 7 พอร์ตประกอบด้วย A, B, C, D, E, F, G แต่ละพอร์ตจะมีจำนวนบิตดังนี้

- PORTA = PA0–PA7 จำนวน 8 บิต

- PORTB = PB0–PB7 จำนวน 8 บิต

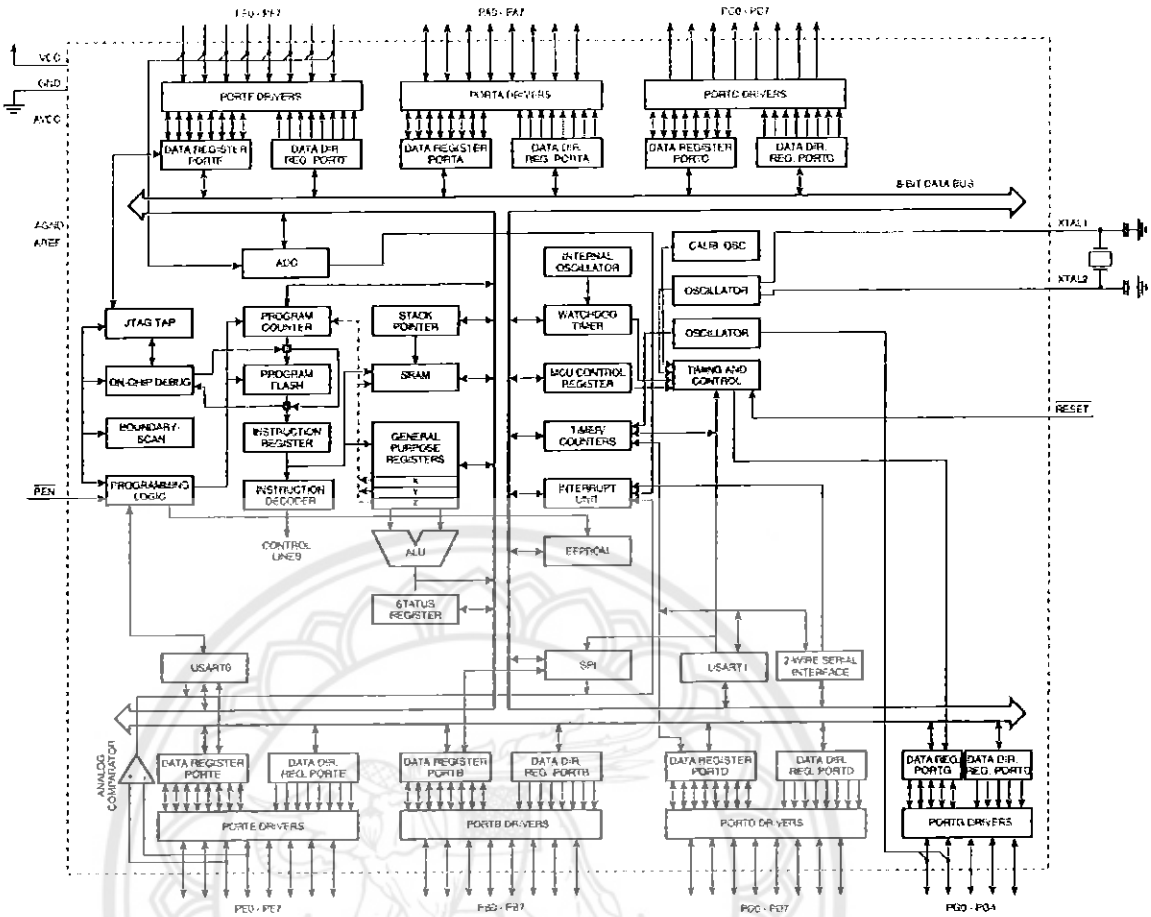
- PORTC = PC0–PC7 จำนวน 8 บิต

- PORTD = PD0–PD7 จำนวน 8 บิต

- PORTE = PE0–PE7 จำนวน 8 บิต

- PORTF = PF0–PF7 จำนวน 8 บิต

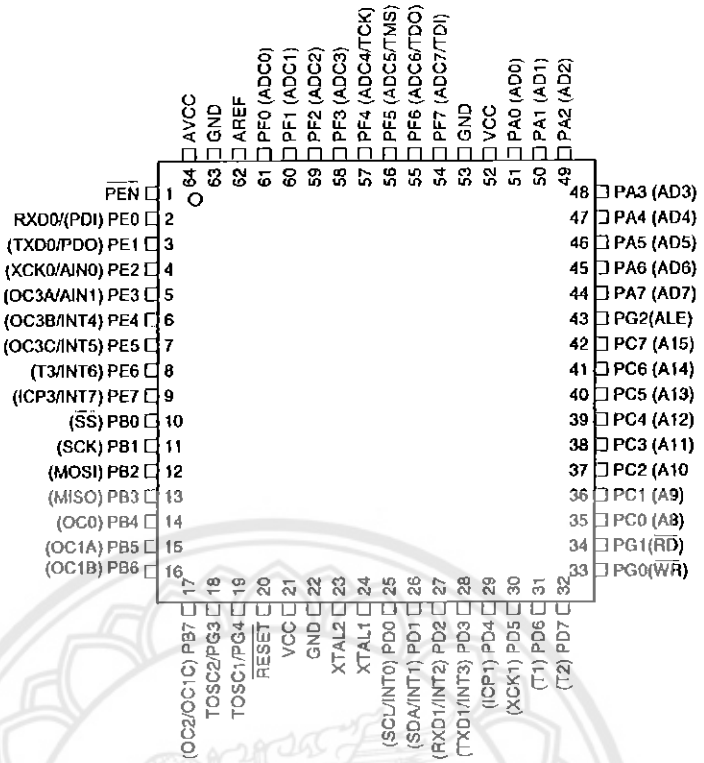
- PORTG = PG0–PG4 จำนวน 5 บิต



รูปที่ 2.13 สถาปัตยกรรมภายในของไมโครคอนโทรลเลอร์ ATmega 64

2.7.2 ตัวถังและตำแหน่งขาสัญญาณของ ATmega 64

ขาพอร์ตอินพุตเอาท์พุตของไมโครคอนโทรลเลอร์ ATmega 64 มีจำนวน 64 ขา โดยแบ่งเป็นขาพอร์ตอินพุตเอาท์พุตอิสระ จำนวน 53 ขา ประกอบไปด้วย PA, PB, PC, PD, PE, PF ขนาด 8 บิต PG ขนาด 5 บิต และขาพอร์ตที่เกี่ยวข้องกับสัญญาณอนาลอกจำนวน 2 ขาพอร์ต คือ AREF และ AVCC รายละเอียดขาพอร์ตทั้งหมดแสดงดังรูปที่ 2.14 ตัวถังของ ATmega 64 และตำแหน่งขาสัญญาณต่างๆ



รูปที่ 2.14 ตัวถังของ ATmega 64 และตำแหน่งขาสัญญาณต่างๆ

ตารางที่ 2.1 ชื่อขา ตำแหน่งขา ชนิดขาและหน้าที่การทำงานของไมโครคอนโทรลเลอร์ ATmega 64

ชื่อขา	ตำแหน่ง	ชนิด	หน้าที่และการทำงาน
VCC	21 , 52	อินพุต	ขาแรงดันไฟตรง
GND	53 , 63	อินพุต	ขากราวด์
PortA (PA0.0-PA0.7)	44-51	อินพุต/เอาท์พุต	ขาพอร์ตเป็นอินพุตเอาท์พุตดิจิตอล กำหนดการพูลอัพภายในขาพอร์ตได้ (internal pull-up register) และสามารถกำหนดใช้งานเป็นพอร์ตอินพุตสัญญาณอนาลอก (A/D Converter) ได้
Port B (PB0.0-PB0.7)	10-17	อินพุต/เอาท์พุต	ขาพอร์ตเป็นอินพุตเอาท์พุตดิจิตอล กำหนดการพูลอัพภายในขาพอร์ตได้ (internal pull-up register) และเป็นขาพอร์ตหน้าที่พิเศษอีกด้วย เช่น ขาสำหรับการโปรแกรมชิพ ขาป้อนสัญญาณนาฬิกาภายนอก

ชื่อขา	ขาที่	ชนิด	หน้าที่และการทำงาน
Port C (PC0.0-PC0.7)	35-42	อินพุต/เอาท์พุต	ขาพอร์ตเป็นอินพุตเอาท์พุตดิจิตอล กำหนดการพูลอัพภายในขาพอร์ตได้ (internal pull-up register) และเป็นขาพอร์ตหน้าที่พิเศษอีกด้วย เช่น ขาเชื่อมต่อกับดีบักและโปรแกรมด้วยการเชื่อมต่อแบบ JTAG
Port D (PD0.0-PD0.7)	25-32	อินพุต/เอาท์พุต	ขาพอร์ตเป็นอินพุตเอาท์พุตดิจิตอล กำหนดการพูลอัพภายในขาพอร์ตได้ (internal pull-up register) และเป็นขาพอร์ตหน้าที่พิเศษอีกด้วย เช่น ขาเชื่อมต่อพอร์ตอนุกรม ขาอินเตอร์รัปต์เนื่องจากสัญญาณภายนอก
Port E (PE0.0-PE0.7)	2-9	อินพุต/เอาท์พุต	ขาพอร์ตเป็นอินพุตเอาท์พุตดิจิตอล กำหนดการพูลอัพภายในขาพอร์ตได้ (internal pull-up register) และเป็นขาพอร์ตหน้าที่พิเศษอีกด้วย เช่น ขาเชื่อมต่อพอร์ตอนุกรม ขาอินเตอร์รัปต์เนื่องจากสัญญาณภายนอก
Port F (PF0.0-PF0.7)	54-61	อินพุต/เอาท์พุต	ขาพอร์ตเป็นอินพุตเอาท์พุตดิจิตอล กำหนดการพูลอัพภายในขาพอร์ตได้ (internal pull-up register) และเป็นขาพอร์ตหน้าที่พิเศษอีกด้วย เช่น ขาเชื่อมต่อพอร์ตอนุกรม ขาอินเตอร์รัปต์เนื่องจากสัญญาณภายนอก
Port G (PG0.0-PG0.4)	18-19 33-34 43	อินพุต/เอาท์พุต	ขาพอร์ตเป็นอินพุตเอาท์พุตดิจิตอล กำหนดการพูลอัพภายในขาพอร์ตได้ (internal pull-up register) และเป็นขาพอร์ตหน้าที่พิเศษอีกด้วย เช่น ขาเชื่อมต่อพอร์ตอนุกรม ขาอินเตอร์รัปต์เนื่องจากสัญญาณภายนอก
$\overline{\text{RESET}}$	20	อินพุต	ขารีเซตวงจร
XTAL1	23	อินพุต	ขาต่อคริสตอลออสซิเลเตอร์ ช่องที่ 1
XTAL2	24	เอาท์พุต	ขาต่อคริสตอลออสซิเลเตอร์ ช่องที่ 2
AVCC	64	เอาท์พุต	ขาแรงดันสำหรับพอร์ต A และโมดูลแปลงสัญญาณอนาลอกเป็นดิจิตอล

ชื่อขา	ขาที่	ชนิด	หน้าที่และการทำงาน
AREF	62	เอาท์พุต	ขาแรงดันอนาลอกอ้างอิงสำหรับโมดูลแปลงสัญญาณอนาลอกเป็นดิจิตอล
$\overline{PEN}$	1	อินพุต	ขาต่อใช้งานสำหรับโหมดการเขียนโปรแกรม SPI แบบอนุกรม

2.8 การทำงานของเซนเซอร์วัดอุณหภูมิ

เซนเซอร์วัดอุณหภูมิ DS18S20 มีฟังก์ชันการทำงานเป็นเซนเซอร์วัดอุณหภูมิแบบดิจิตอล ตัววัดอุณหภูมิแบบ Digital DS18S20 จะวัดค่าอุณหภูมิได้ละเอียดถึง 9-bit (resolution) ในหน่วย centigrade และสามารถกำหนดจุด trigger เมื่อเกินหรือต่ำกว่าค่าที่ผู้ใช้เก็บไว้ในหน่วยความจำได้ (nonvolatile) โดยทำงานในช่วงอุณหภูมิ -55 ถึง +125 องศาเซลเซียส มีความถูกต้องแม่นยำ ±0.5 องศาเซลเซียส ในช่วงการวัดอุณหภูมิระหว่าง -10 ถึง +85 องศาเซลเซียส นอกจากนั้นยังสามารถดึงกำลังไฟฟ้าได้โดยตรงจากสายสัญญาณข้อมูล (data line) โดยมีคุณสมบัติดังนี้

1) เซนเซอร์วัดอุณหภูมิเบอร์ DS18S20 แบบตัวถัง PR35

2) สามารถวัดค่าได้ตั้งแต่ –55 ถึง 125 องศาเซลเซียส หรือ -67 ถึง 257 องศาฟาเรนไฮต์

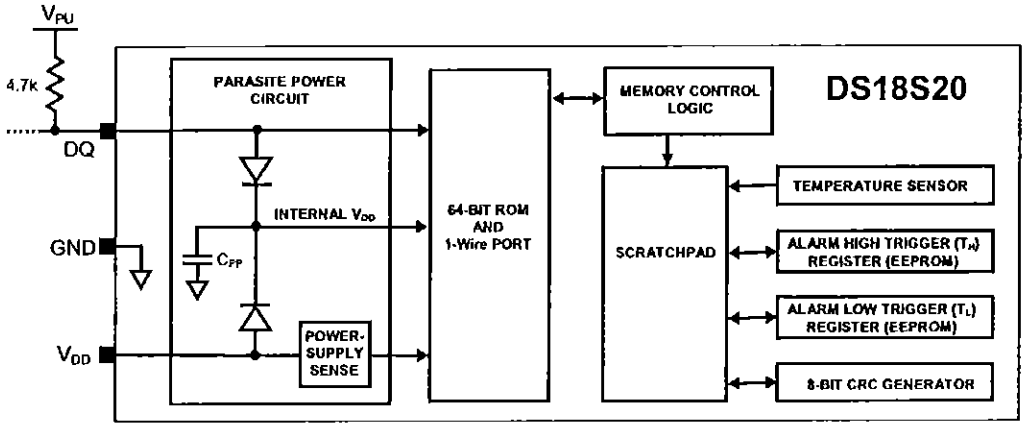
3) มีความละเอียดในการวัดค่าที่ = 0.5 องศาเซลเซียส หรือ 0.9 องศาฟาเรนไฮต์

4) ระยะเวลาการแปลงสัญญาณประมาณ 200 มิลลิวินาที

5) สามารถต่อกับขาสัญญาณไมโครคอนโทรลเลอร์ได้ทันทีโดยใช้การอินเตอร์เฟซแบบสายเส้นเดียว (1 Wire interface)

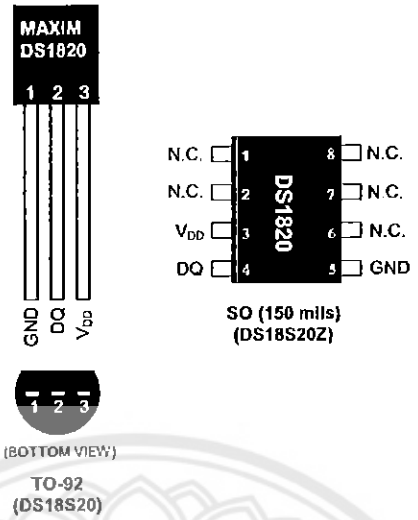
6) สายต่อความยาว 1.2 เมตร พร้อมคอนเน็คเตอร์ขนาด 5 ขา ที่สามารถต่อใช้กับบอร์ดแสดงผลรุ่น DISP4-100 ได้ทันที

7) แรงดันใช้งานที่ 3.0 ถึง 5.0 โวลต์



รูปที่ 2.15 Block diagram ของ DS18S20

2.8.1 ขาสัญญาณสำหรับสื่อสารข้อมูลของโมดูล DS18S20



รูปที่ 2.16 รูปร่างของโมดูล DS18S20

1) ขา VDD เป็นขาแรงดันไฟเลี้ยงเซนเซอร์

2) ขา DQ เป็นขาสัญญาณอินเตอร์เฟซกับไมโครคอนโทรลเลอร์แบบ 1-Wire interface

3) ขา GND เป็นขาแรงดันไฟ 0 โวลต์ หรือ Ground

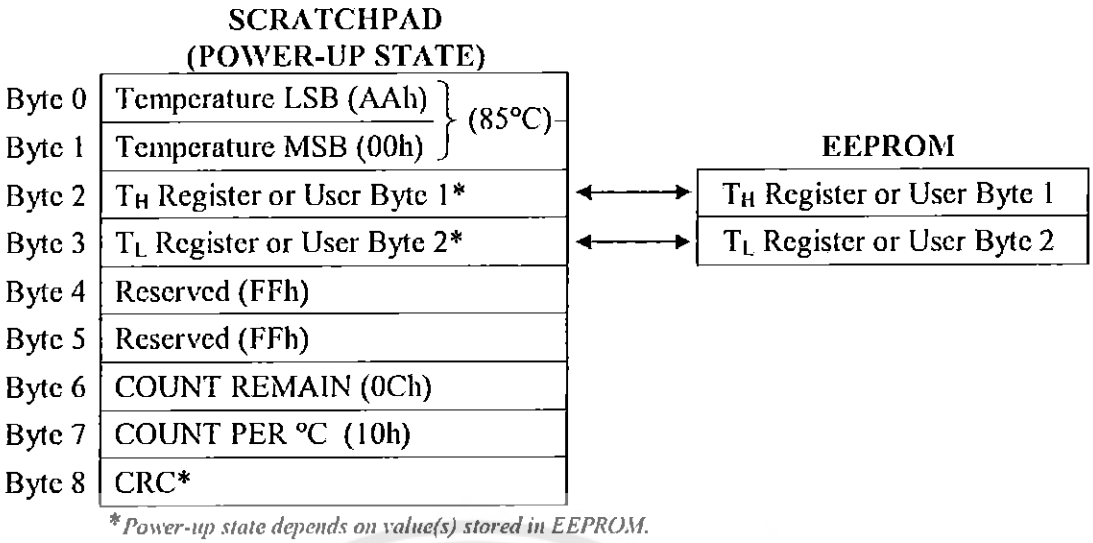
2.8.2 รูปแบบการสื่อสารอ่านข้อมูลอุณหภูมิของ DS18S20

ตัววัดอุณหภูมิ (Sensor) ที่มีความละเอียด 9-bit ซึ่งเทียบได้กับ 0.5 องศาเซลเซียส steps เมื่อต้องการให้มีการวัดค่าอุณหภูมิ และทำการแปลง A-D ตัว master ต้องส่งคำสั่งให้ทำการแปลงค่าอุณหภูมิ (Convert Temperature, 44h) หลังจากที่มีการแปลงค่าอุณหภูมิ (ใช้เวลาไม่เกิน 750 มิลลิวินาที) ค่าอุณหภูมิที่ได้จะเก็บไว้ใน temperature register ที่มีขนาด 2 bytes ใน scratchpad memory แล้วตัว DS18S20 จะกลับเข้าสู่สภาวะ idle

	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
LS BYTE	2^6	2^5	2^4	2^3	2^2	2^1	2^0	2^{-1}
	BIT 15	BIT 14	BIT 13	BIT 12	BIT 11	BIT 10	BIT 9	BIT 8
MS BYTE	S	S	S	S	S	S	S	S

S = SIGN

รูปที่ 2.17 Temperature Register Format



รูปที่ 2.18 Memory Map

ในระบบ 1-Wire bus จะมีตัว bus master อยู่ตัวเดียวที่คอยควบคุมอุปกรณ์ slave ตัว DS18S20 จะเป็นตัว slave เสมอ เมื่อไหร่ก็ตาม ที่มีอุปกรณ์ slave อยู่เพียงตัวเดียวในระบบ bus ก็จะเป็น ระบบ "single drop" และเมื่อมีอุปกรณ์ slave อยู่หลายตัวก็จะเป็น ระบบ "multidrop"

ในการส่งข้อมูลหรือคำสั่ง จะเป็นการส่ง Least significant bit ก่อน

ลำดับขั้นตอนในการเข้าถึงตัว DS18S20 จะเป็นดังนี้

- ลำดับที่ 1: Initialization: การทำงานในระบบ 1-Wire bus จะเริ่มด้วยขั้นตอนการ initialization ที่ประกอบด้วย การส่ง reset pulse จากตัว bus master ที่จะตามด้วย presence pulse ที่ตอบกลับมาจากอุปกรณ์ slave(s)

- ลำดับที่ 2: ชุดคำสั่ง ROM (ROM command): หลังจากที่ตัว bus master ได้รับ presence pulse ตัวมันก็จะสามารถใช้ชุดคำสั่ง ROM เพื่อใช้งาน 64-bit ROM codes ที่เป็นค่าประจำตัวของอุปกรณ์ slave แต่ละตัว

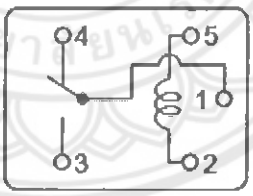
- ลำดับที่ 3: ชุดคำสั่ง Function ของ DS18S20: หลังจากที่ตัว bus master ได้ใช้ชุดคำสั่ง ROM เพื่อเลือกใช้ตัว DS18S20 ที่ต้องการจะติดต่อด้วย ตัวมันก็จะสามารถใช้ชุดคำสั่ง Function ของ DS18S20 ที่ทำให้สามารถอ่าน หรือเขียนค่าจาก Scratchpad Memory, เริ่มกระบวนการแปลงค่าอุณหภูมิ และพิจารณาโหมดการทำงานของแหล่งจ่ายไฟ

2.9 รีเลย์

รีเลย์ (Relay) เป็นอุปกรณ์ใช้สำหรับปิดและเปิดวงจรเช่นเดียวกับสวิตช์แต่การทำงานของรีเลย์ทำงานด้วยการให้กระแสไฟฟ้าไหลผ่านเข้าไปในขดลวดของรีเลย์ทำให้เกิดอำนาจแม่เหล็กดูดขั้วโลหะของรีเลย์ติดหรือขาดออกจากกันทำให้วงจรต่อกันหรือขาดออกจากกันเหมือนการปิดเปิดวงจรด้วยสวิตช์

รีเลย์เป็นสวิตช์แม่เหล็กไฟฟ้าซึ่งทำหน้าที่เชื่อมต่อหรือเปิดวงจรการทำงานจะดึงหน้าสัมผัสเข้าหาหรือให้หนีออกจากอีกข้างหนึ่งดังแสดงในรูปที่ 2.19 และรูปที่ 2.20 โดยแสดงสัญลักษณ์และรูปลักษณะของรีเลย์

รูปที่ 2.19 รีเลย์

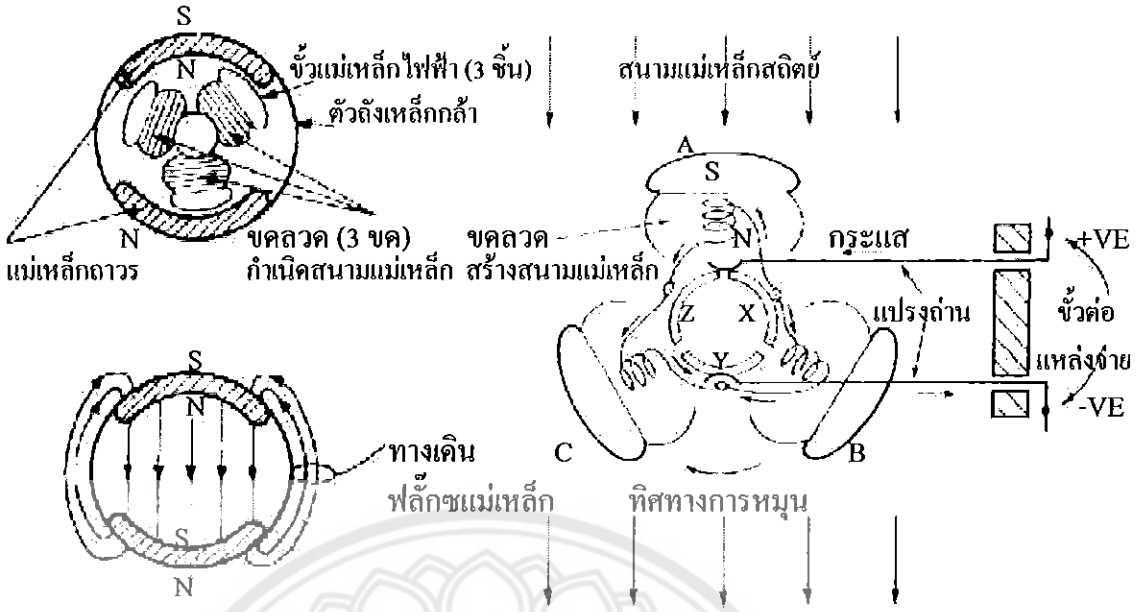


รูปที่ 2.20 วงจรภายในของรีเลย์

2.10 มอเตอร์ไฟฟ้ากระแสตรง (DC MOTOR)

2.10.1 หลักการทำงานของมอเตอร์กระแสตรง

มอเตอร์กระแสตรงจะมีหลักการทำงานโดยวิธีการผ่านกระแสให้กับขดลวดในสนามแม่เหล็ก ซึ่งจะทำให้เกิดแรงแม่เหล็ก โดยส่วนของแรงนี้จะขึ้นอยู่กับกระแสและกำลังของสนามแม่เหล็ก



แม่เหล็กถาวรและทิศทางเดินของฟลักซแม่เหล็ก

รูปที่ 2.21 แสดงโครงสร้างทั่วไปของมอเตอร์กระแสตรง

จากในรูปทางเดินของฟลั๊กซแม่เหล็ก และสนามแม่เหล็กจะเกิดจากแท่งแม่เหล็กเฟอร์ไรต์ 2 ชิ้นที่ขึ้นรูปเป็นแบบโค้งยึดติดกับตัวถังได้พอดี เพื่อที่จะให้เส้นแรงแม่เหล็กวิ่งเข้าสู่ใจกลางของมอเตอร์ได้ ดังนั้นความเข้มของแม่เหล็กจะขึ้นอยู่กับขนาดความหนาของแม่เหล็กด้วย ซึ่งส่งผลให้ฟลั๊กซแม่เหล็กวิ่งไปบนตัวถังโลหะ กระแสไฟฟ้าในขดลวดที่พันกับทุ่นโรเตอร์ก็จะทำให้เกิดสนามแม่เหล็กไฟฟ้า และต้านกับสนามแม่เหล็กถาวร จึงเกิดเป็นแรงบิดเพื่อที่จะหมุนทุ่นโรเตอร์ให้ไปในทิศทางเดียวกันกับทิศทางของสนามแม่เหล็กที่มีแรงมากกว่า กระแสก็จะไหลผ่านไปยังทุ่นโรเตอร์ โดยผ่านแปรงถ่าน ซึ่งจะสัมผัสกับแหวนตัวนำในทุ่นโรเตอร์ และแหวนคอมมิวเตเตอร์ ซึ่งจะถูกแบ่งออกเป็น 3 เซกเมนต์เพื่อที่จะทำหน้าที่นำกระแสเข้าขดลวดนั้นเอง

2.10.2 การขับและกลับทิศทางของมอเตอร์กระแสตรง (DC MOTOR)

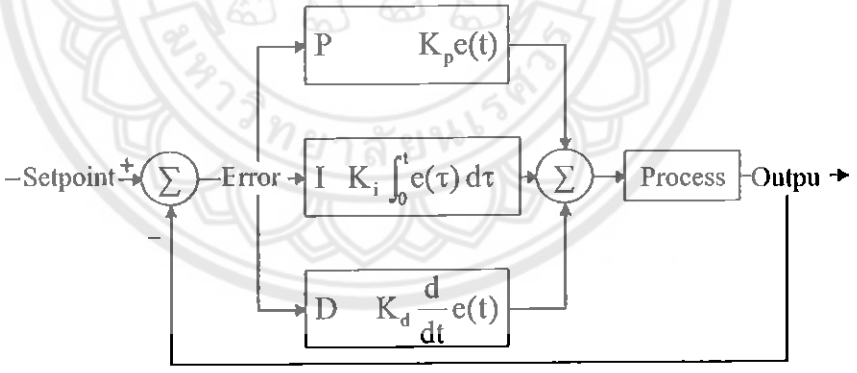
ในการใช้ไอซีไมโครคอนโทรเลอร์เป็นตัวควบคุมการหมุน และทิศทางของมอเตอร์กระแสตรงนั้น เราจะต้องมีส่วนของวงจร ที่เรียกว่าวงจรขับมอเตอร์ (Driver) ในส่วนของวงจรกลับทิศทางของมอเตอร์นั้น สามารถที่จะใช้รีเลย์ต่อวงจร สวิตซ์เพื่อกลับทิศทางของขั้วไฟกระแสตรง หรืออาจใช้อุปกรณ์สารกึ่งตัวนำที่เป็นวงจรขับกำลังเช่น ทรานซิสเตอร์ มอสเฟต แล้วแต่วิธีที่เราจะเลือกใช้งาน

2.10.3 การควบคุมความเร็วของมอเตอร์กระแสตรง

การควบคุมความเร็วของมอเตอร์กระแสตรงมีหลายวิธีด้วยกัน ซึ่งอาจจะใช้วิธีการควบคุมแบบพื้นฐานทั่วไปเช่นการควบคุมด้วยวิธีการใช้ตัวต้านทานปรับค่าโดยต่ออนุกรมกับมอเตอร์ หรือใช้วิธีการการควบคุมโดยการเปลี่ยนค่าของระดับแรงดันที่ป้อนให้กับมอเตอร์ แต่การควบคุมในวิธีดังกล่าวถึงแม้ว่าจะควบคุมความเร็วมอเตอร์ให้คงที่ได้ แต่ที่ความเร็วต่ำจะส่งผลให้แรงบิดต่ำไปด้วย ดังนั้นเราจึงเลือกใช้วิธีการควบคุมโดยการจ่ายกระแสไฟให้กับมอเตอร์เป็นช่วงๆ โดยอาศัยกระแสไฟที่ป้อนให้กับมอเตอร์ให้เป็นค่าเฉลี่ยที่เกิดขึ้นในแต่ละช่วง ซึ่งเราเรียกว่าวิธีการของการมอดูเลชั่นทางความกว้างของพัลส์ PWM (Pulse Width Modulation)

2.11 การควบคุมแบบพีไอดี (PID Control)

การควบคุมแบบพีไอดี เป็นระบบการควบคุมแบบป้อนกลับที่ใช้กันอย่างกว้างขวาง ซึ่งค่าที่จะนำไปใช้ในการคำนวณเป็นค่าความผิดพลาดที่หามาจากความแตกต่างของตัวแปรในกระบวนการและค่าที่ต้องการ ตัวควบคุมจะพยายามลดค่าผิดพลาดให้เหลือน้อยที่สุดด้วยการปรับค่าสัญญาณขาเข้าของกระบวนการ ค่าตัวแปรของพีไอดีที่ใช้จะปรับเปลี่ยนตามธรรมชาติของระบบ



รูปที่ 2.22 แสดงแผนภาพบล็อกของการควบคุมแบบพีไอดี

2.11.1 ทฤษฎีการควบคุมแบบพีไอดี

ทฤษฎีการควบคุมแบบพีไอดี ได้ชื่อตามการรวมกันของเทอมของตัวแปรทั้งสามตามสมการ

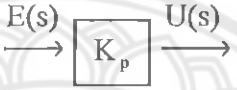
$$u(t) = K_p e(t) + K_i \int_0^t e(\tau)\,d\tau + K_d \frac{d}{dt} e(t) \qquad (2.11)$$

ถ้าเขียนอยู่ในรูปแบบผลการแปลงลาปลาซ จะได้ว่า

$$G(s) = K_p e + K_i + K_d s \qquad (2.12)$$

2.11.2 การควบคุมแบบสัดส่วน (Proportion Control)

การควบคุมระบบแบบป้อนกลับโดยใช้ตัวควบคุมแบบสัดส่วนนั้น สัญญาณควบคุม (u) จะเป็นสัดส่วนโดยตรงกับค่าสัญญาณความผิดพลาด (e) ที่เกิดจากผลต่างระหว่างค่าสัญญาณอ้างอิงกับสัญญาณเอาท์พุตของระบบที่ทำการควบคุมนั้นดังแสดงในแผนภาพบล็อกดังนี้



รูปที่ 2.23 แสดงแผนภาพบล็อกของการควบคุมแบบสัดส่วน

ความสัมพันธ์ระหว่างสัญญาณเอาท์พุตจากตัวควบคุมและสัญญาณผิดพลาดที่ส่งเข้าไปในระบบสามารถเขียนได้ดังนี้

$$u(t) = K_p e(t) \qquad (2.13)$$

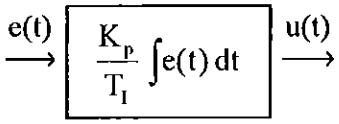
ถ้าเขียนอยู่ในรูปแบบผลการแปลงลาปลาซ จะได้ว่า

$$\frac{U(s)}{E(s)} = K_p \qquad (2.14)$$

เมื่อค่าเกน K_p จะเป็นค่าอัตราขยายของตัวควบคุมนี้หรือจะเรียกว่าเกนสัดส่วน

2.11.3 การควบคุมแบบอินทิกรัล (Integral Control)

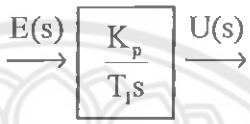
การควบคุมแบบอินทิกรัล นั้นเพื่อต้องการลดค่าความผิดพลาดในสถานะอยู่ตัว ในขณะเดียวกันค่าความมีเสถียรภาพของระบบก็จะลดน้อยลง การควบคุมนั้นดังแสดงในแผนภาพบล็อกดังนี้



รูปที่ 2.24 แสดงแผนภาพบล็อกของการควบคุมแบบอินทิกรัล

$$u(t) = \frac{K_p}{T_I}\int e(t)\,dt \tag{2.15}$$

หรือในรูปแบบของสมการฟังก์ชันถ่ายโอน จะได้ว่า



ถ้าเขียนอยู่ในรูปแบบผลการแปลงลาปลาซ จะได้ว่า

$$\frac{U(s)}{E(s)} = \frac{K_p}{T_I s} \tag{2.16}$$

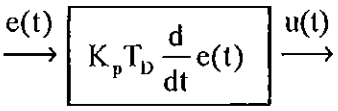
โดยที่ T_I = integral time or reset time

$\frac{1}{T_I}$ = reset rate

การควบคุมแบบอินทิกรัลนี้ จะเห็นว่าสัญญาณควบคุม u(t) อาจจะมีค่าค่อนข้างมากก็ได้ โดยไม่ลดลงทั้งๆ ที่สัญญาณผิดพลาด e(t) มีค่าเป็นศูนย์ในภายหลังหรือเมื่อเวลาผ่านไปแล้วก็ตาม ทั้งนี้ก็เพราะว่าสัญญาณควบคุมในกรณีของการควบคุมแบบอินทิกรัลขึ้นอยู่กับค่าในอดีต ซึ่งจะไม่เหมือนกับตัวควบคุมเชิงสัดส่วน ซึ่งอยู่กับค่าปัจจุบัน

2.11.4 การควบคุมแบบอนุพันธ์ (Derivative Control)

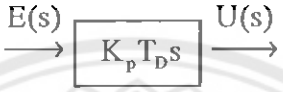
การควบคุมแบบอนุพันธ์นั้นจะช่วยในการเพิ่มค่าความหน่วงให้กับระบบที่ต้องการจะควบคุม นั่นก็คือทำให้ระบบมีเสถียรภาพเพิ่มมากขึ้น แต่โดยทั่วไปแล้วตัวควบคุมเชิงอนุพันธ์นี้จะไม่ทำให้ค่าความผิดพลาดในสถานะอยู่ตัวมีค่าเป็นศูนย์ได้ ในขณะเดียวกันจะเห็นว่าสัญญาณเอาท์พุตที่ออกจากตัวควบคุมเชิงอนุพันธ์นี้เป็นสัญญาณที่เกิดจากการหาอนุพันธ์ของสัญญาณผิดพลาด $\frac{d}{dt}e(t)$ การควบคุมนั้นดังแสดงในแผนภาพบล็อกดังนี้



รูปที่ 2.25 แสดงแผนภาพบล็อกของการควบคุมแบบอนุพันธ์

$$u(t) = K_p T_D \frac{d}{dt} e(t) \tag{2.17}$$

รูปแบบของสมการฟังก์ชันถ่ายโอน จะได้ว่า



ถ้าเขียนอยู่ในรูปแบบผลการแปลงลาปลาซ จะได้ว่า

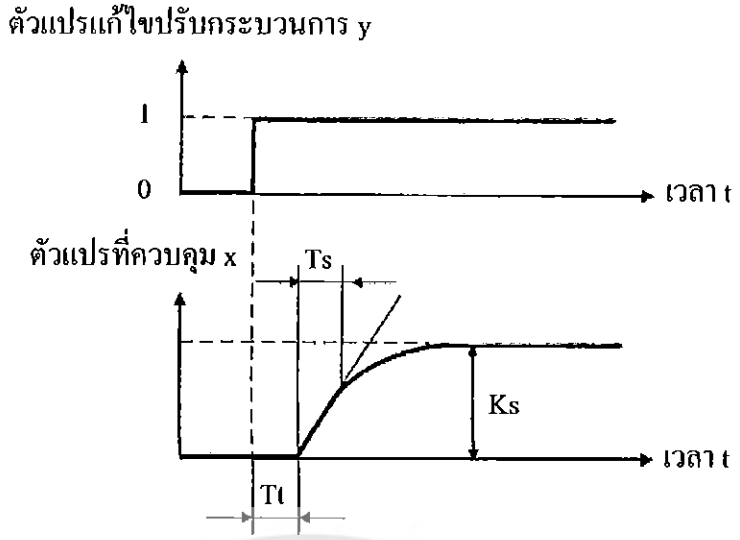
$$\frac{U(s)}{E(s)} = K_p T_D s \tag{2.18}$$

โดยที่ค่า T_D = ช่วงเวลาอนุพันธ์

การควบคุมแบบอนุพันธ์นี้ส่วนมากแล้วจะใช้ร่วมกับตัวควบคุมตัวอื่น หากสัญญาณผิดพลาดนี้มีสัญญาณรบกวนมาก สัญญาณเอาท์พุตที่ออกมาจากตัวควบคุมเชิงอนุพันธ์นี้จะกระเพื่อมค่อนข้างมาก (เนื่องจากค่าความชันของสัญญาณเปลี่ยนแปลงค่อนข้างมาก) ซึ่งจะทำให้ระบบควบคุมของเรามีเสถียรภาพได้

2.11.5 การตั้งค่าตัวควบคุมผลตอบสนองต่อสัญญาณขั้นบันไดโดยใช้วิธี Ziegler-Nichols

การวิเคราะห์จากผลสนองต่อสัญญาณขั้นบันไดของระบบที่ถูกควบคุม รูปที่ 2.26 ซึ่งจากการวิเคราะห์ก็จะทำให้ได้ ค่าเวลาเฉื่อย (Tt), ค่าคงที่การหน่วงเวลา (Ts) และค่าอัตราการขยายของระบบที่ถูกควบคุม (Ks) สำหรับค่าพารามิเตอร์ของตัวควบคุมจะถูกคำนวณได้โดยการใช้สมการที่ปรากฏอยู่ในตารางที่ 2.2



รูปที่ 2.26 การหาคุณสมบัติของระบบโดยการวิเคราะห์จากผลตอบสนองต่อสัญญาณขั้นบันได

ตารางที่ 2.2 การตั้งค่าพารามิเตอร์โดยการวิเคราะห์จากผลตอบสนองต่อสัญญาณขั้นบันได

Controller	K_p	Tn	Tv
P	$\frac{Ts}{Ks \cdot Tt}$	-	-
PI	$0.9\frac{Ts}{Ks \cdot Tt}$	3.3 Tt	-
PID	$1.2\frac{Ts}{Ks \cdot Tt}$	2 Tt	0.5 Tt

วิธีการของ Ziegler-Nichols เป็นการปรับตั้งค่าของตัวควบคุมที่ทำให้ผลตอบสนองของวงจรการควบคุมแบบวงปิดต่อการรบกวนระบบเป็นไปอย่างเหมาะสม เมื่อตัวแปรอ้างอิงเกิดการเปลี่ยนแปลงในลักษณะของสัญญาณขั้นบันไดการปรับตั้งค่าด้วยวิธีการของ Ziegler - Nichols จะทำให้ตัวแปรที่ถูกควบคุมมีการเปลี่ยนแปลงตาม และมีการเกิดผลตอบสนองเกินค่าเป้าหมายของสัญญาณเกิดขึ้น

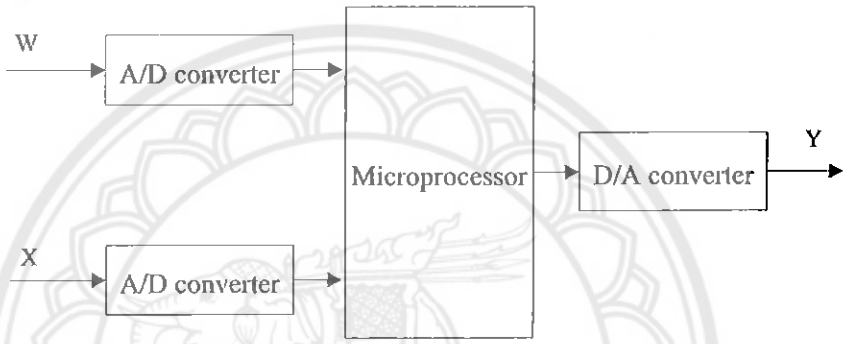
2.11.6 การแปลงค่าพารามิเตอร์ และค่าคงที่เวลาของตัวควบคุม

ตัวควบคุมแบบพีโอดี โดยปกติจะถูกปรับตั้งค่าสัมประสิทธิ์ของการควบคุมแบบอัตราส่วน K_p และค่าที่ของเวลา Tv และ Tn แต่ถึงอย่างไรก็ตามก็มีบ่อยครั้งที่ใช้การกำหนดตัวควบคุมด้วยค่าสัมประสิทธิ์ K_p , K_i และ K_d ซึ่งข้อมูลในตารางที่ 3.5 ได้สรุปถึงสมการที่ใช้สำหรับเปลี่ยนค่าของตัวแปรต่างๆ

ตารางที่ 2.3 สัมประสิทธิ์ของตัวควบคุม และค่าคงที่ของเวลา

การแปลงจากค่า K_p, Tv, Tn ไปเป็น K_i, K_d	การแปลงจากค่า K_p, K_i, K_d ไปเป็น Tn, Tv
$K_i = \frac{K_p}{Tn}$	$Tn = \frac{K_p}{K_i}$
$K_d = K_p \cdot Tv$	$Tv = \frac{K_d}{K_p}$

2.11.7 ตัวควบคุม พีไอดีแบบดิจิตอล



รูปที่ 2.27 รูปแสดงแผนภาพการไหลของสัญญาณสำหรับตัวควบคุมแบบดิจิตอล

ในรูปที่ 2.27 แสดงให้เห็นการไหลของสัญญาณสำหรับตัวควบคุมแบบดิจิตอล โดยที่ค่าของตัวแปลอ้างอิงและตัวแปลที่ถูกควบคุมต่างก็จะถูกอ่านค่าโดยผ่านตัวแปลงสัญญาณอนาลอกเป็นสัญญาณดิจิตอล ซึ่งตัวแปลงสัญญาณจะทำให้เกิดตัวแปรเอาท์พุตในลักษณะของค่าตัวเลข จากนั้นตัวไมโครโปรเซสเซอร์ก็จะปฏิบัติงานตามขั้นตอนดังต่อไปนี้

1) ทำการอ่านค่าของตัวแปรอ้างอิง และตัวแปรที่ถูกควบคุม

2) ทำการเปรียบเทียบค่าของตัวแปรอ้างอิง และตัวแปรที่ถูกควบคุม

3) ทำการประมวลผลสัญญาณ

4) ค่าเอาท์พุตจากการประมวลผลจะถูกส่งไปที่อินพุตของตัวแปลงสัญญาณดิจิตอลเป็นสัญญาณอนาลอก ต่อไป

ตัวแปลงสัญญาณดิจิตอลเป็นสัญญาณอนาลอกจะทำให้ข้อมูลที่ได้จากการประมวลผลของไมโครโปรเซสเซอร์ ซึ่งถูกกำหนดเป็นค่าตัวแปรแก้ไขปรับกระบวนการ ถูกแปลงให้ไปอยู่ในรูปของสัญญาณแรงดันไฟฟ้า

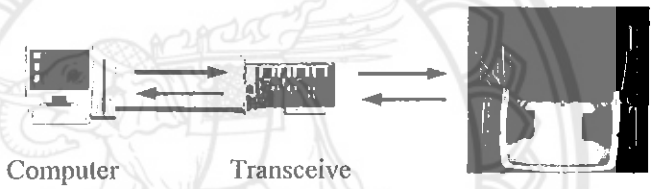
การปฏิบัติการทั้งในส่วนของการอ่านค่าสัญญาณ การประมวลผลและการส่งสัญญาณเอาท์พุตออกไปจะถูกกระทำให้เกิดขึ้นซ้ำๆกันตั้งแต่ 20 ครั้ง ถึงจะเป็นจำนวนหลายพันครั้งต่อวินาที ทั้งนี้ขึ้นอยู่กับชนิดของตัวควบคุมที่ถูกนำมาใช้

จากทฤษฎีที่ได้ศึกษามาแล้วทั้งหมดใน บทที่ 2 ทำให้มีความรู้ในเรื่อง คุณสมบัติ,การทำงาน และการต่อใช้งานของไอซีที่นำมาใช้ในโครงงานคือ ไมโครคอนโทรลเลอร์ตระกูล AVR(ATmega64),เซนเซอร์วัดอุณหภูมิ DS18S20, รีเลย์ และมีความรู้ในเรื่องของทฤษฎีการควบคุมแบบพีไอดี เพื่อนำไปใช้ในการเขียนโปรแกรมภาษาซี เพื่อใช้ควบคุมการทำงาน และสร้างชิ้นงานใน บทที่ 3 ต่อไป

บทที่ 3
ขั้นตอนการดำเนินงานและการออกแบบ

หลังจากการศึกษาหลักการและทฤษฎีที่เกี่ยวข้องกับการทดลองมาแล้วในบทที่ผ่านมา สามารถนำหลักการดังกล่าวมาประยุกต์เพื่อสร้างเป็นโครงงานตู้แช่วัคซีนควบคุมด้วยไมโครคอนโทรลเลอร์ ที่สามารถนำไปใช้งานได้จริง ขั้นตอนการดำเนินงานและการออกแบบมีดังต่อไปนี้

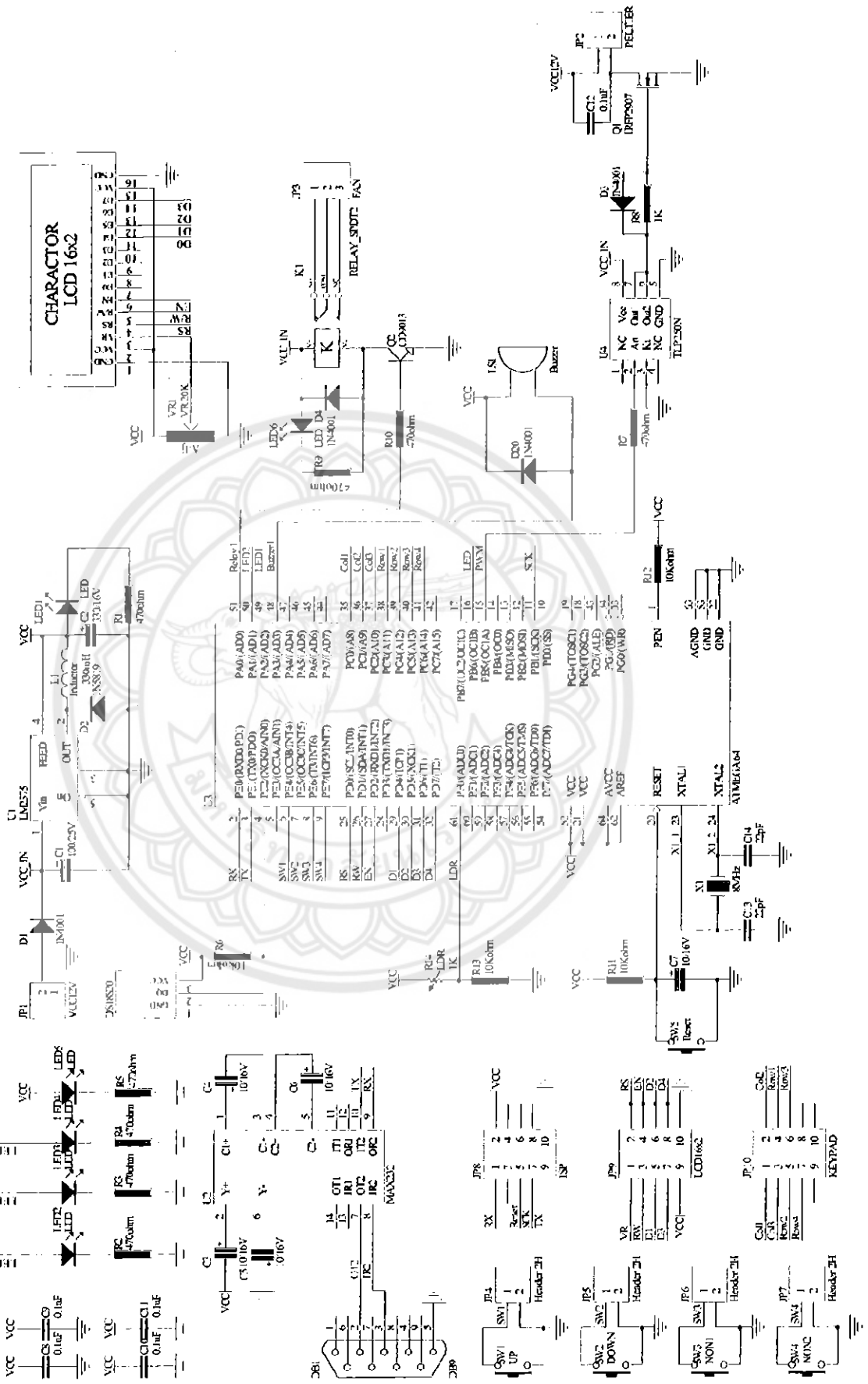
3.1 การออกแบบวงจรควบคุมระบบตู้แช่วัคซีนด้วยไมโครคอนโทรลเลอร์



รูปที่ 3.1 ภาพรวมของระบบทั้งหมดในโครงงาน

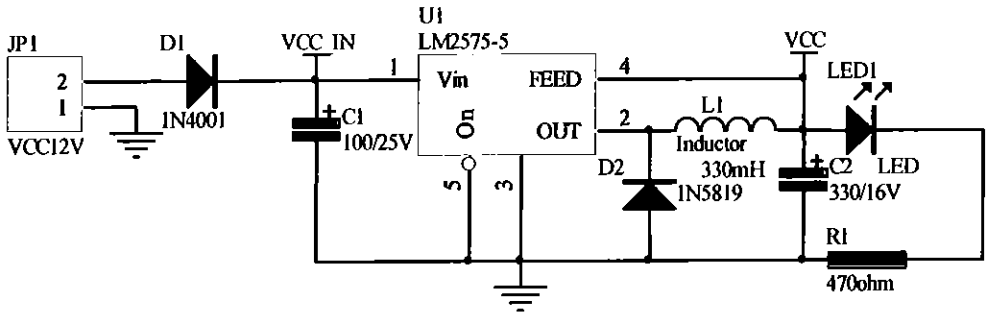
หลักการทำงานของระบบในรูปที่ 3.1 นั้น เริ่มจากการเขียนโปรแกรมข้อมูลสั่งการจากคอมพิวเตอร์ ผ่านโปรแกรม Code Vision AVR C Compiler โดยการเขียนข้อมูลมาอยู่ในรูปแบบของภาษาซี แล้วจึงนำส่งข้อมูลผ่านพอร์ตยูเอสบี (USB 2.0) หลังจากที่คอมพิวเตอร์ได้ส่งข้อมูลออกมาทางพอร์ตยูเอสบีแล้ว ทางด้านฝั่งรับข้อมูลจะเชื่อมต่อเข้ากับไมโครคอนโทรลเลอร์ตระกูล AVR เพื่อรับข้อมูลจากคอมพิวเตอร์เพื่อควบคุมการทำงานของอุปกรณ์ภายนอก คือ การรับค่าอุณหภูมิจากสวิตช์ การวัดค่าอุณหภูมิปัจจุบันด้วยไอซี DS18S20 หลังจากนั้นจึงควบคุมการทำงานของเพลเทียร์ภายในตู้แช่วัคซีนเพื่อทำความเย็นให้ได้ตามอุณหภูมิที่ตั้งค่าไว้

อุปกรณ์รับข้อมูลจากคอมพิวเตอร์ในโครงงานนี้คือ วงจรควบคุมอุณหภูมิด้วยการทำงานของไมโครคอนโทรลเลอร์ตระกูล AVR (ATmega 64) มีการทำงานรับข้อมูลจากคอมพิวเตอร์เป็นข้อมูลอักขระตัวอักษร เพื่อมาแปลงเป็นเลขฐาน 16 ใช้ในการสั่งการและแสดงผลและส่งข้อมูลต่างๆไปยังอุปกรณ์ควบคุมการทำงานในการทำความเย็นตามรูปภาพที่ 3.2



รูปที่ 3.2 รูปวงจรควบคุมอุณหภูมิด้วยไมโครคอนโทรลเลอร์

3.1.1 วงจรไฟเลี้ยงไมโครคอนโทรลเลอร์



รูปที่ 3.3 วงจรไฟเลี้ยงไมโครคอนโทรลเลอร์

1) JP1 เป็นคอนเน็คเตอร์สำหรับต่อไฟเลี้ยงเข้าตั้งแต่ 7-40 โวลต์ DC

2) D1 คือไดโอดมีหน้าที่ป้องกันกระแสย้อนกลับจาก JP1 ที่เป็นคอนเน็คเตอร์

3) C1 และ C2 คือตัวเก็บประจุมีหน้าที่กรองแรงดันไฟให้เรียบ

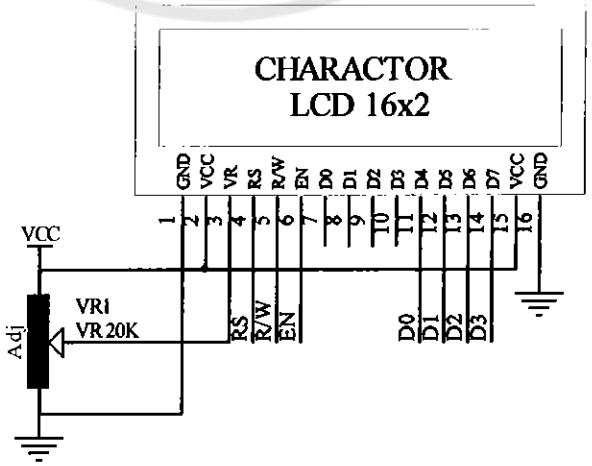
4) LM2575-5 มีหน้าที่พยุงแรงดันไฟฟ้าเมื่อเข้ามา ทำให้เอาท์พุตออก 5 โวลต์

5) D2 คือไดโอดมีหน้าที่ป้องกันกระแสย้อนกลับจาก L1 ที่เป็นตัวเหนี่ยวนำ

6) L1 คือตัวเหนี่ยวนำซึ่งทำหน้าที่เป็นตัวหน่วงกระแสเพื่อไม่ให้แรงดันตกขณะที่ไฟ 5 โวลต์ จ่ายโหลดมากเกินความต้องการ

7) LED1 เป็นแอลอีดีแสดงสถานะเพื่อให้รู้ว่ามีไฟ 5 โวลต์ ออกมาจาก LM2575-5 หรือไม่ส่วน R1 (ตัวต้านทาน) มีหน้าที่เป็นตัวต้านทานกระแสไม่ให้ไหลผ่านแอลอีดีมากเกินไป

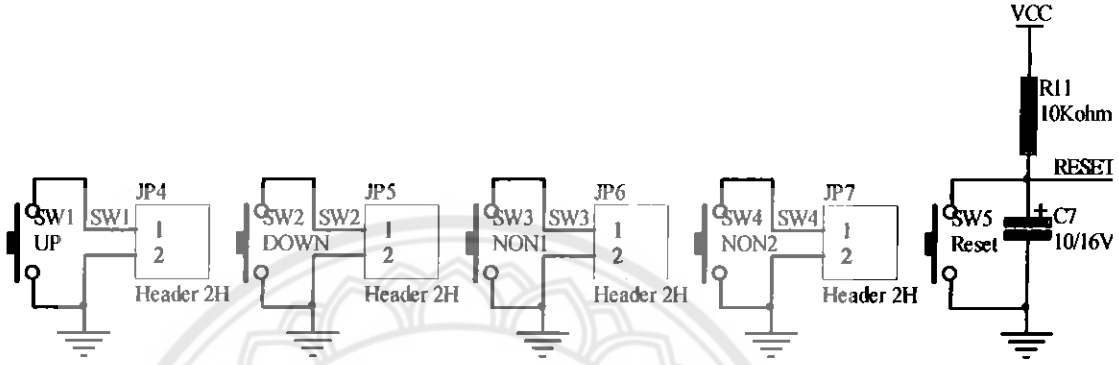
3.1.2 การต่อใช้งานจอแอลซีดี 16x2



รูปที่ 3.4 วงจรจอแอลซีดี

จอแอลซีดีที่นำมาใช้งานเป็นจอแอลซีดีขนาด 16x2 คือ16 ตัวอักษร 2 บรรทัดในการต่อกับไมโครคอนโทรลเลอร์ ที่เป็นการต่อแบบ 4 บิต

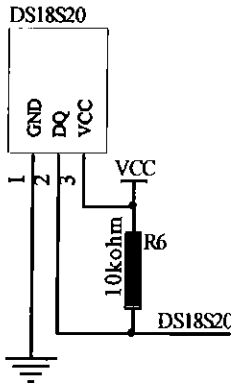
3.1.3 การต่อใช้งานสวิตช์



รูปที่ 3.5 วงจรสวิตช์

สวิตช์ เป็นสวิตช์แบบกดติดปล่อยดับ มี สวิตช์หนึ่ง สวิตช์สอง และสวิตช์ห้า ที่ใช้งาน สวิตช์สาม และสวิตช์สี่ ไม่ใช้งาน หลักการทำงาน คือ เมื่อมีการกดสวิตช์หนึ่ง อุณหภูมิจะเพิ่มขึ้นทีละ 0.5 องศาเซลเซียส ต่อการกดสวิตช์หนึ่งครั้ง เมื่อกดสวิตช์สอง อุณหภูมิจะลดลงทีละ 0.5 องศาเซลเซียสต่อการกดสวิตช์หนึ่งครั้ง และสวิตช์ห้าใช้รีเซ็ต เพื่อเริ่มต้นการทำงานใหม่อีกครั้ง

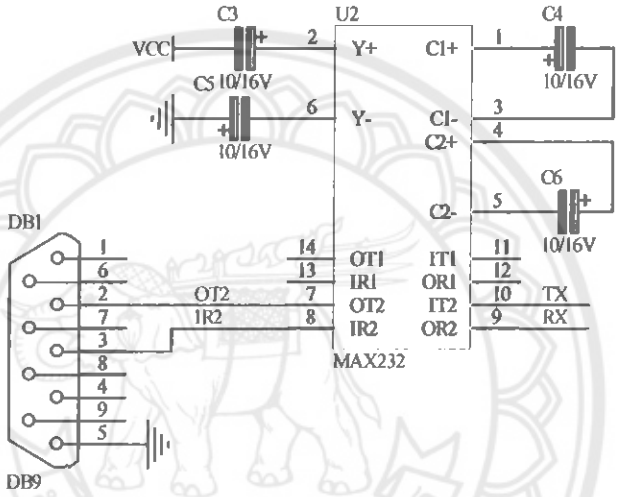
3.1.4 การต่อใช้งานไอซีวัดอุณหภูมิ DS18S20



รูปที่ 3.6 วงจร DS18S20 ตรวจวัดอุณหภูมิ

ไอซี DS18S20 เป็นตัวตรวจวัดอุณหภูมิแบบสายเดียว มีสามขาคือ ขาแหล่งจ่ายไฟ ขากราวด์ และขารับข้อมูลค่าของอุณหภูมิ ซึ่งเป็นขารับข้อมูลสายเดียว ซึ่งทำหน้าที่ตรวจวัดอุณหภูมิของอากาศว่ามีค่ากี่องศาเซลเซียส และนำค่าอุณหภูมิส่งไปยังไมโครคอนโทรลเลอร์เพื่อเก็บไว้คำนวณหาค่าความแตกต่างระหว่างอุณหภูมิของอากาศขณะนั้นกับค่าอุณหภูมิของอากาศที่เราได้กำหนดไว้ เพื่อนำค่าที่ได้ไปใช้ในการควบคุมแบบพีไอดีต่อไป

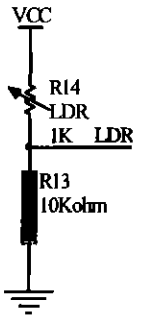
3.1.5 การต่อใช้งาน MAX 232



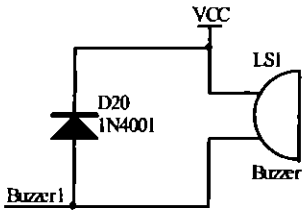
รูปที่ 3.7 วงจร MAX 232 ส่งข้อมูลไปยังคอมพิวเตอร์

วงจร MAX 232 เป็นตัวส่งข้อมูลจากไมโครคอนโทรลเลอร์ไปยังคอมพิวเตอร์ ผ่านทางพอร์ต Hyper Terminal

3.1.6 การต่อใช้งานสัญญาณเตือน



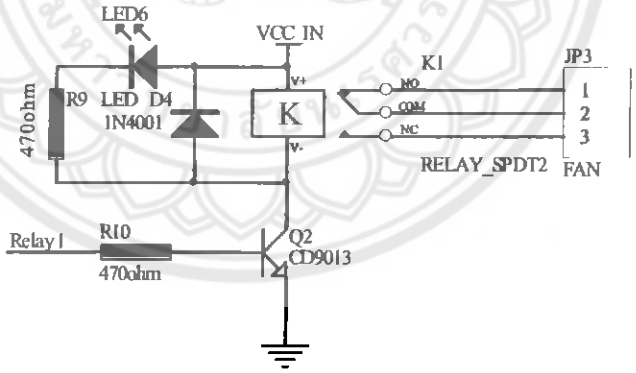
รูปที่ 3.8 วงจร LDR



รูปที่ 3.9 วงจร BUZZER

วงจรสัญญาณเตือน การทำงานจะเริ่มจากวงจร LDR เป็นตัวรับสัญญาณแสงแล้วส่งค่าความเข้มแสงไปยังไมโครคอนโทรลเลอร์ เมื่อเกินค่าที่ได้กำหนดไว้จะส่งจ่ายกระแสไฟไปยังวงจร BUZZER เพื่อส่งสัญญาณเตือน

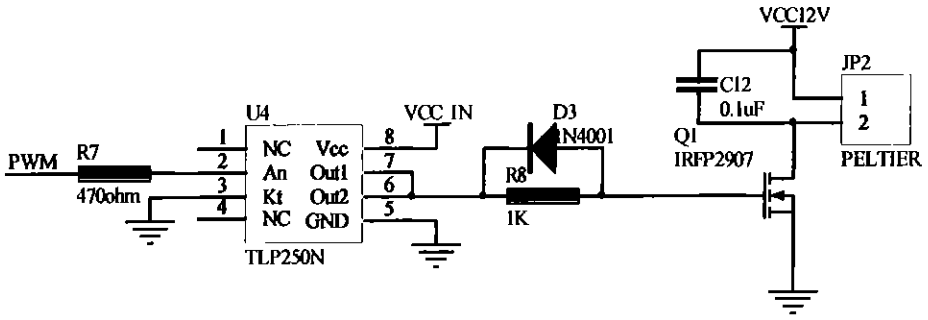
3.1.7 การต่อใช้งานพัดลม DC 12 โวลต์



รูปที่ 3.10 วงจรพัดลม DC 12 โวลต์

วงจรพัดลม DC 12 โวลต์ เป็นวงจรระบายความร้อนให้เพลเทียร์

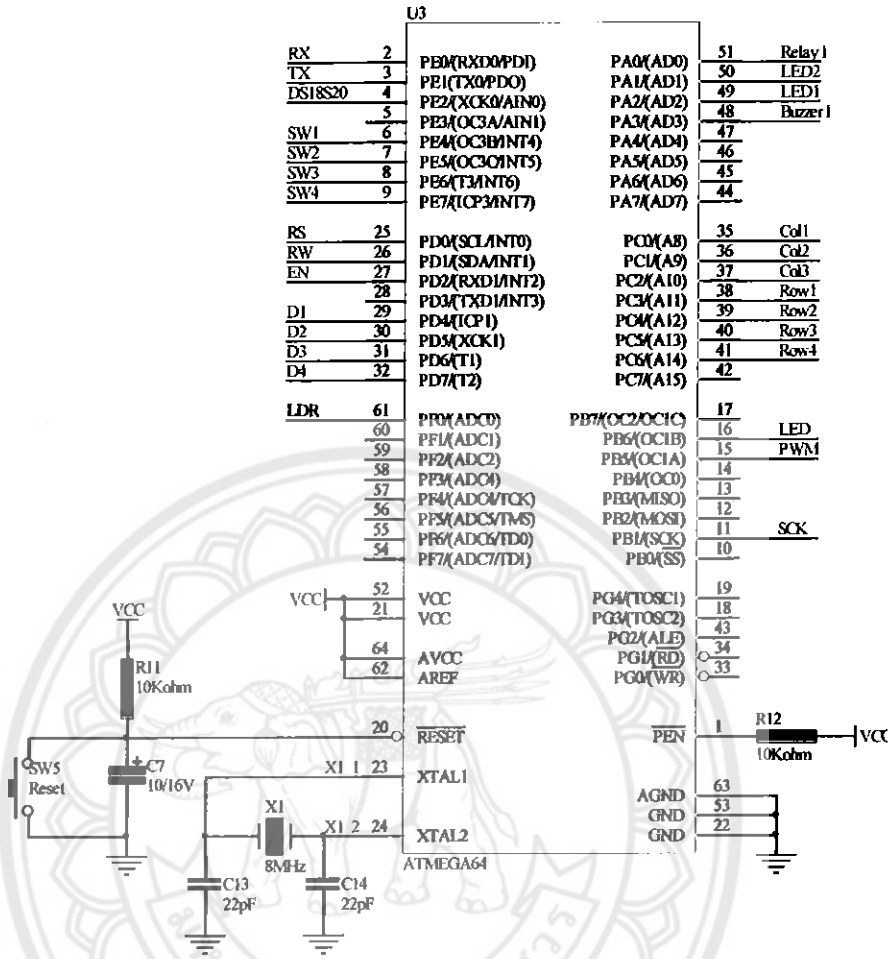
3.1.8 การต่อใช้งานเพลเทียร์



รูปที่ 3.11 วงจรจ่ายกระแสไฟฟ้าเพลเทียร์

วงจรจ่ายกระแสไฟฟ้าเพลเทียร์ การทำงานจะเริ่มจากไมโครคอนโทรลเลอร์ โดยการส่งสัญญาณ PWM ไปยัง ไอซี TLP250N หลังจากนั้นไอซี TLP250N ก็จะส่งสัญญาณเข้ากับมอสเฟต IRFP2907 เพื่อควบคุมการทำงานของ PELTIER

3.1.9 การต่อใช้งานไมโครคอนโทรลเลอร์ตระกูล AVR (ATmega 64)



รูปที่ 3.12 วงจรใช้งานไมโครคอนโทรลเลอร์

U3 เป็นไมโครคอนโทรลเลอร์ตระกูล AVR (ATmega64) ไมโครคอนโทรลเลอร์จะทำงานได้ต้องอาศัย 3 อย่างหลักๆ คือ

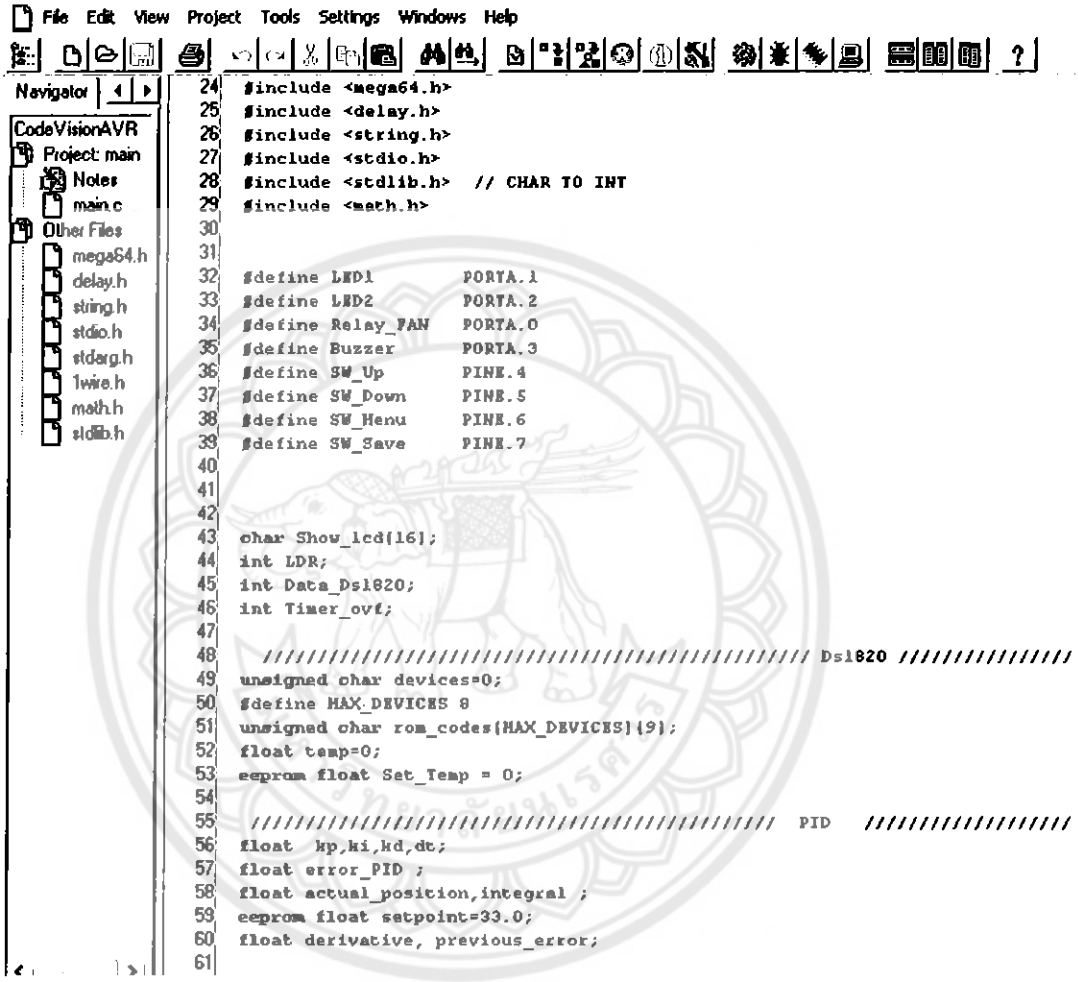
1. ไฟเลี้ยงไมโครคอนโทรลเลอร์

2. X-TAL หรือ คริสตัล มีหน้าที่สร้างสัญญาณนาฬิกาให้กับไมโครคอนโทรเลอร์ C13 (ตัวเก็บประจุ) และ C14 (ตัวเก็บประจุ) มีหน้าที่กรองสัญญาณนาฬิกา (CRYSTAL) ก่อนที่จะเข้าไปยังไมโครคอนโทรลเลอร์ C4 (ตัวเก็บประจุ) มีหน้าที่กรองสัญญาณรับกวนที่จะเกิดขึ้นกับไมโครคอนโทรลเลอร์

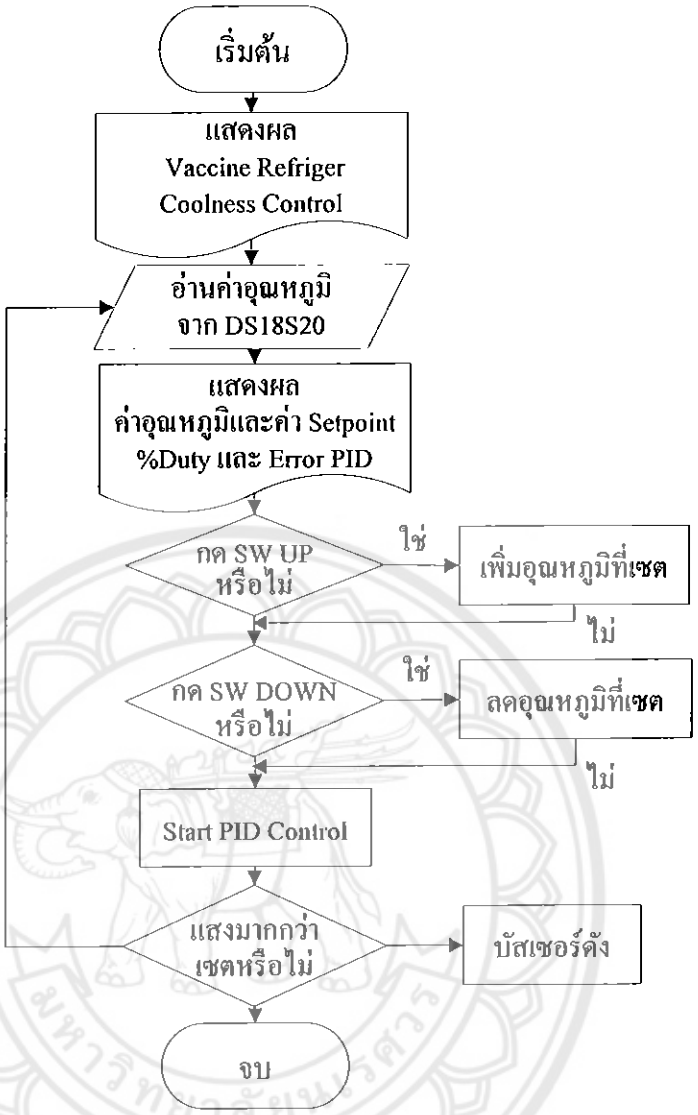
3. R11 (ตัวต้านทาน) และ C7 (ตัวเก็บประจุ) เป็นวงจรรีเซตไมโครคอนโทรลเลอร์ ได้มาจากแผ่นข้อมูลของไมโครคอนโทรลเลอร์ส่วน SW1, SW2, SW5 เป็นการต่อสวิตช์แบบแอคทีฟ GND (กราวด์) คือการรับสัญญาณ GND (กราวด์) และ JP8 เป็นการเชื่อมต่อไว้สำหรับดาวน์โหลดโปรแกรมเข้าไมโครคอนโทรลเลอร์โดยผ่านสาย ISP Down ของบริษัท ETT

3.2 การเขียนโปรแกรมควบคุมการทำงานของไมโครคอนโทรลเลอร์

การทำงานของระบบควบคุมตู้แช่วัคซีนควบคุมด้วยไมโครคอนโทรลเลอร์ซึ่งใช้โปรแกรมภาษาซีในการเขียนคำสั่งควบคุมไมโครคอนโทรลเลอร์ตระกูล AVR มีรูปแบบของโปรแกรมที่ใช้ และ แผนผังการทำงานของโปรแกรม ตามที่แสดงไว้ในรูปที่ 3.13 และ 3.14



รูปที่ 3.13 รูปโปรแกรม Code Vision AVR C Compiler



รูปที่ 3.14 การทำงานของโปรแกรม

เริ่มต้นโปรแกรมจะแสดงผล Vaccine Refriger Coolness Control ที่หน้าจอแอลซีดีเพื่อแสดงว่าโปรแกรมเริ่มทำงานแล้วและจะเริ่มอ่านค่าอุณหภูมิที่ตั้งไว้กับอ่านค่าอุณหภูมิในอากาศที่วัดได้ขณะนั้นเพื่อนำค่าอุณหภูมิไปแสดงผลบนจอแอลซีดี หลังจากนั้นโปรแกรมจะตรวจสอบว่ามีการกดสวิตช์เพิ่มอุณหภูมิหรือไม่ ถ้ามีการกดสวิตช์เพิ่มอุณหภูมิ อุณหภูมิที่ตั้งค่าไว้ก็จะเพิ่มทีละ 0.5 องศาเซลเซียสต่อการกดสวิตช์หนึ่งครั้ง แต่ถ้าไม่มีการกดสวิตช์เพิ่มอุณหภูมิโปรแกรมก็จะทำงานในขั้นตอนต่อไป คือ ตรวจสอบว่ามีการกดสวิตช์ลดอุณหภูมิหรือไม่ ถ้ามีการกดสวิตช์ลดอุณหภูมิ อุณหภูมิที่ตั้งค่าไว้ก็จะถูกลดค่า 0.5 องศาเซลเซียส ต่อการกดสวิตช์หนึ่งครั้ง หลังจากนั้นก็นำค่าอุณหภูมิที่วัดได้จากอากาศ และค่าอุณหภูมิที่ได้ตั้งค่าไว้ เข้าสู่โปรแกรมการคำนวณของการควบคุมแบบพีไอดี ตรวจสอบว่าแสงหากมากกว่าที่เซตไว้บัสเซอร์จะดัง และวนกลับไปเริ่มอ่านค่าอุณหภูมิใหม่อีกครั้ง จนอุณหภูมิที่วัดได้จากอากาศมีค่าเท่ากับอุณหภูมิที่ตั้งค่าไว้ จึงจบการทำงาน

3.2.1 การทำงานของโปรแกรมการควบคุมแบบพีไอดี

หลังจากดำเนินการทำอุปกรณ์รับส่งสัญญาณในระบบเรียบร้อยแล้ว ต่อไปเป็นขั้นตอนการดำเนินการคำนวณโดยใช้สมการ การควบคุมแบบพีไอดี ควบคุมไมโครคอนโทรลเลอร์ เพื่อเข้าไปสั่งการเพลเทียร์เพื่อทำความเย็นให้ได้อุณหภูมิที่ต้องการ

การสร้างตัวควบคุมในที่นี้โดยจะพิจารณาถึงในรูปแบบของการใช้ฮาร์ดแวร์ที่เป็นไมโครคอนโทรลเลอร์ จากเนื้อหาเรื่องทฤษฎีระบบควบคุมที่นำเสนอก่อนหน้านี้ ทำให้ทราบว่าตัวควบคุมพีไอดี สามารถได้รับการเขียนแสดงในรูปความสัมพันธ์ทางคณิตศาสตร์ว่า

$$u(t) = K_p e(t) + K_i \int_0^t e(\tau)\,d\tau + K_d \frac{d}{dt} e(t)$$

(3.1)

เมื่อ $u(t)$ เป็นสัญญาณเอาท์พุตของตัวควบคุมและ $e(t)$ เป็นสัญญาณค่าผิดพลาดที่เกิดขึ้นของระบบควบคุมตัวควบคุม สมการพีไอดีแบบดิจิตอลที่ใช้ควบคุม จึงแทนได้ด้วยความสัมพันธ์ดังสมการนี้

$$u(t) = (K_p \times \text{error}) + (K_i \times \text{integral}) + (K_d \times \text{derivative}) \quad (3.2)$$

error คือ ค่าผิดพลาด ณ เวลาปัจจุบันมีค่าเป็น

$$\text{error} = \text{setpoint} - \text{actual_position} \quad (3.3)$$

setpoint คือ ค่าที่กำหนด

actual_position คือ ค่าความอุณหภูมิจริงขณะนั้น

integral คือ ค่าความผิดพลาดของระบบที่ถูกอินทิเกรท สามารถเขียนแทนด้วยสมการ

$$\text{integral} = \text{integral} - (\text{error} \times dt) \quad (3.4)$$

โดยที่ค่า integral เริ่มต้นจากศูนย์ตามการคำนวณของโปรแกรม และค่า dt คือคาบเวลาการสุ่มสัญญาณ

derivative คือ การหาอนุพันธ์ของค่าความผิดพลาดของระบบ เขียนแทนด้วยสมการ

$$\text{derivative} = \frac{(\text{error} - \text{previous_error})}{dt} \quad (3.5)$$

previous_error คือ ค่าความผิดพลาดก่อนหน้า โดยจะมีค่าเริ่มต้นเท่ากับค่า error ปัจจุบัน

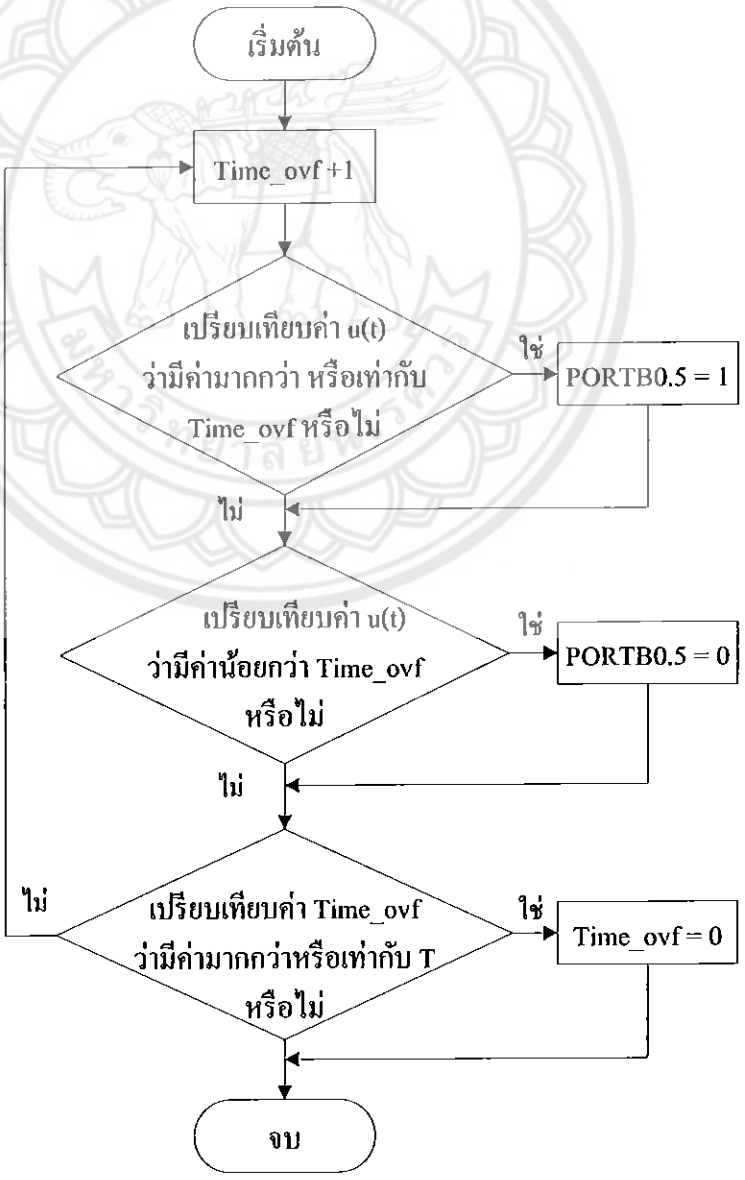
สามารถเขียนแผนภูมิการทำงานของโปรแกรมได้ดังนี้โดยการคำนวณแต่ละรอบจะต้องจำกัดอยู่ในคาบเวลาของการสุ่มสัญญาณ

รูปที่ 3.15 แผนภูมิการทำงานของโปรแกรมควบคุมพีไอดีแบบดิจิตอล

การทำงานของโปรแกรมควบแบบคุมพีไอดี เริ่มจากการอ่านค่าอินพุต สองค่าคือค่าอุณหภูมิที่ตั้งไว้ และอุณหภูมิจริงของอากาศขณะนั้น หลังจากนั้นนำสองค่าที่อ่านได้ไปคำนวณหาค่าความผิดพลาด โดยนำค่าอุณหภูมิที่ตั้งไว้ลบออกด้วยค่าอุณหภูมิที่วัดได้ขณะนั้น และนำค่า

ความผิดพลาดที่คำนวณได้ ไปเข้าสู่การคำนวณของทฤษฎีการควบคุมแบบพีไอดีเพื่อหาค่า u(t) (ตัวแปรแก้ไขระบบ) หลังจากนั้นนำค่า u(t) ที่ได้จากการคำนวณของโปรแกรมไปควบคุมการทำงานของเพลเทียร์ภายในตู้แช่วัคซีนเพื่อทำความเย็นให้ได้อุณหภูมิที่ตั้งค่าไว้เมื่อจบการทำงาน 1 คาบ คือ หนึ่งรอบการทำงานของโปรแกรม โปรแกรมพีไอดีก็จะกลับไปเริ่มทำงานอ่านค่าอินพุตใหม่ คืออุณหภูมิที่ตั้งไว้ และ อุณหภูมิจริงของอากาศขณะนั้น เพื่อใช้ในการทำงานของโปรแกรมอีกครั้งจะทำงานแบบนี้ต่อไปเรื่อยๆจนกว่าจะได้ค่าตัวแปรแก้ไขระบบที่ทำให้ระบบการควบคุมอุณหภูมิของอากาศอยู่ในสภาวะสมดุลคือค่าอุณหภูมิที่ตั้งไว้เท่ากับอุณหภูมิของอากาศที่วัดได้ในขณะนั้น

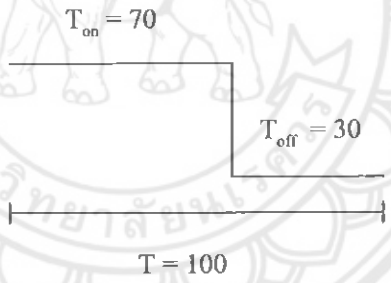
3.2.2 การเขียนโปรแกรมนำค่า u(t) ออกไปควบคุมอุปกรณ์ภายนอก



รูปที่ 3.16 แผนภูมิการทำงานของโปรแกรมนำค่า u(t) ออกไปควบคุมอุปกรณ์ภายนอก

การเขียนโปรแกรมกำหนดการทำงานของค่า u(t) กำหนดค่าตัวแปร T=100, u(t) = 0 และ ตัวแปร Time_ovf ซึ่งชุดโปรแกรมนี้จะทำงานในรูปแบบของการอินเทอร์รัพท์ มีการทำงานดังนี้

ให้ Time_ovf บวกหนึ่ง ต่อไปก็ทำการเปรียบเทียบค่า Time_ovf กับค่า u(t) ว่าค่า u(t) มีค่ามากกว่าหรือเท่ากับค่า Time_ovf หรือไม่ ถ้าใช่ก็ให้ส่งเอาท์พุตเท่ากับ 1 ออกทาง PORTB0.5 แต่ถ้าค่า u(t) มีค่าน้อยกว่าค่า Time_ovf ก็ให้ส่งค่าเอาท์พุตท่ากับ 0 ออกทาง PORTB0.5 ต่อไปก็ทำการเปรียบเทียบค่า Time_ovf กับค่า T ว่าค่า Time_ovf มีค่ามากกว่าหรือเท่ากับค่า T หรือไม่ ถ้าไม่ใช่ก็ให้ กลับไปเริ่มทำงานใหม่คือ บวกเพิ่มค่า Time_ovf ขึ้นจากเดิม +1 จนกว่าค่า Time_ovf จะมีค่ามากกว่าหรือเท่ากับค่า T ก็ให้กำหนดค่า Time_ovf เท่ากับ 0 จบการทำงาน จะได้การทำงานของขดลวดความร้อนอยู่ในรูปแบบของพัลส์วิดท์มอดูเลชั่น (Pulse Width Modulation) หรือ PWM ยกตัวอย่าง เช่น เมื่อเราคำนวณ ค่า u(t) จากสมการพีไอดีได้ค่าเท่ากับ 70 จะได้เอาท์พุตเป็น T_{on} = 70 ครั้ง และT_{off} = 30 ครั้ง ซึ่งการทำงานจะขึ้นอยู่กับค่า u(t) ที่คำนวณได้จากสมการพีไอดี คือค่า $u(t)=T_{on}$ และได้รูป PWM ดังนี้



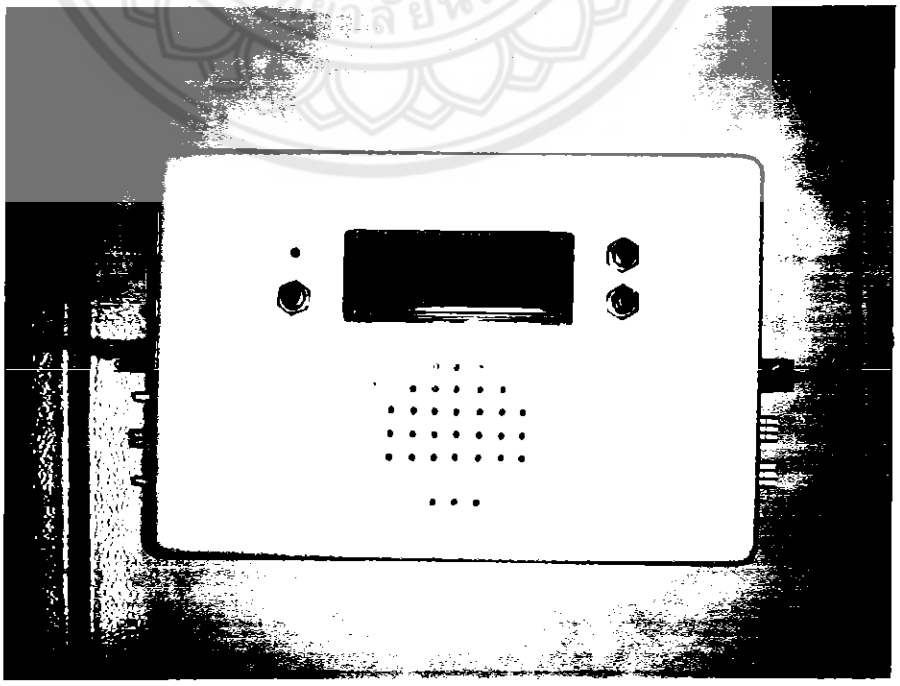
รูปที่ 3.17 รูปการทำงานแบบ PWM ของ u(t)

3.3 การประกอบบอร์ดควบคุมไมโครคอนโทรลเลอร์เข้ากับตู้แช่วัคซีน

หลังจากออกแบบลายวงจรของบอร์ดควบคุมไมโครคอนโทรลเลอร์กับการต่อใช้งานอุปกรณ์ต่างๆเรียบร้อยแล้ว จึงทำการต่ออุปกรณ์ทั้งหมดตามลายวงจรและตามวิธีการต่อใช้งานของไอซีแต่ละแบบลงในบอร์ด โดยต่อเชื่อมด้วยสายไฟ ได้บอร์ดควบคุมดังแสดงในรูปที่ 3.18

รูปที่ 3.18 บอร์ดควบคุมไมโครคอนโทรลเลอร์

เพื่อความเรียบร้อยของชิ้นงาน ชุดวงจรของตัวควบคุมไมโครคอนโทรลเลอร์ ที่ติดตั้งบนแผงวงจร จะถูกประกอบอยู่ในกล่องอุปกรณ์ที่จัดเตรียมไว้ ค่าอุณหภูมิของอากาศที่วัดได้ขณะนั้น ค่าอุณหภูมิของอากาศที่ตั้งค่าไว้ ค่าดิวตี้ไซเคิล และค่าผิดพลาด จะถูกแสดงผลบนหน้าจอแอลซีดี ซึ่งติดตั้งอยู่ด้านหน้าของกล่อง ดังแสดงในรูปที่ 3.19



รูปที่ 3.19 กล่องควบคุมไมโครคอนโทรลเลอร์

เมื่อนำกล่องของบอร์ดควบคุมไมโครคอนโทรลเลอร์ และอุปกรณ์วัดอุณหภูมิมาต่อเข้ากับตู้แช่วัคซีน จะได้ตู้แช่วัคซีนควบคุมด้วยไมโครคอนโทรลเลอร์ ดังแสดงในรูปที่ 3.20

รูปที่ 3.20 ตู้แช่วัคซีนควบคุมด้วยไมโครคอนโทรลเลอร์

จากการดำเนินการสร้างชิ้นงานใน บทที่ 3 โดยเริ่มจากการสร้างบอร์ดควบคุมไมโครคอนโทรลเลอร์เพื่อนำไปใช้ในการควบคุมระบบการทำงานของตู้แช่วัคซีน หลังจากได้บอร์ดควบคุมไมโครคอนโทรลเลอร์แล้ว ก็ทำการเขียนโปรแกรมภาษาซีควบคุมการทำงานของตู้แช่วัคซีนโดยใช้โปรแกรม (Code Vision AVR C Compiler) ช่วยในการเขียนโปรแกรมภาษาซี และใช้ทฤษฎีของการควบคุมแบบพีไอดีมาประกอบการเขียนโปรแกรม และเมื่อได้ค่า U(t) เรียบร้อยแล้วก็ส่งออกไปควบคุมการทำงานของเพลเทียร์ โดยค่าเอาท์พุตที่ส่งออกไปเป็นแบบ PWM ดังที่ได้อธิบายไว้ หลังจากการดำเนินการทั้งหมดจนทำให้ได้ ตู้แช่วัคซีนควบคุมด้วยไมโครคอนโทรลเลอร์ แล้วก็จะนำไปสู่การทดลองใน บทที่ 4 ต่อไป

บทที่ 4
ผลการทดลองและผลการวิเคราะห์

บทนี้เป็นการดำเนินการปฏิบัติงานต่อจากบทที่ 3 เพื่อพิสูจน์ผลการทดลองว่าอุปกรณ์ดังกล่าวที่ทำขึ้นสามารถใช้งานได้จริง

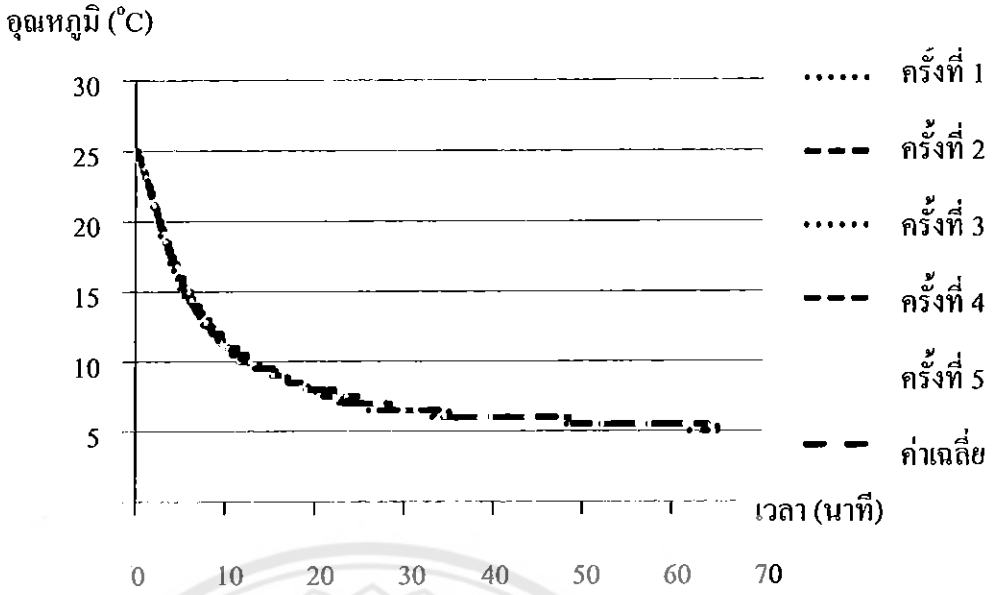
4.1 การทดลองหาระยะเวลาในการลดอุณหภูมิ

การทดลองจับเวลาเพื่อหาระยะเวลาในการลดอุณหภูมิของตู้แช่วัคซีนควบคุมด้วยไมโครคอนโทรเลอร์ที่ทำงานตั้งแต่อุณหภูมิ 25.0 ถึง 5.0 องศาเซลเซียส โดยจะมีการทดลองทั้งหมด 5 ครั้ง ซึ่งผลการทดลองจะแสดงในตารางที่ 4.1

ตารางที่ 4.1 ผลการทดลองหาระยะเวลาในการลดอุณหภูมิทุก ๆ 0.5 องศาเซลเซียส

อุณหภูมิ (°C)	ระยะเวลาในการลดอุณหภูมิ (นาที)					ค่าเฉลี่ย (นาที)
	ครั้งที่ 1	ครั้งที่ 2	ครั้งที่ 3	ครั้งที่ 4	ครั้งที่ 5	
25.0	0.00	0.00	0.00	0.00	0.00	0.00
24.5	0.16	0.16	0.16	0.16	0.17	0.16
24.0	0.33	0.35	0.33	0.34	0.35	0.34
23.5	0.50	0.52	0.49	0.51	0.53	0.51
23.0	1.05	1.08	1.03	1.09	1.09	1.07
22.5	1.20	1.23	1.15	1.21	1.20	1.20
22.0	1.34	1.39	1.27	1.35	1.34	1.34
21.5	1.48	1.53	1.41	1.50	1.49	1.48
21.0	2.03	2.09	1.54	2.04	2.05	2.03
20.5	2.18	2.23	2.07	2.20	2.20	2.18
20.0	2.32	2.39	2.20	2.32	2.31	2.31
19.5	2.47	2.55	2.34	2.47	2.49	2.46
19.0	3.01	3.09	2.47	3.03	3.07	3.01
18.5	3.17	3.27	3.02	3.20	3.22	3.18
18.0	3.34	3.44	3.16	3.36	3.39	3.34

อุณหภูมิ (°C)	ระยะเวลาในการลดอุณหภูมิ (นาที)					ค่าเฉลี่ย (นาที)
	ครั้งที่ 1	ครั้งที่ 2	ครั้งที่ 3	ครั้งที่ 4	ครั้งที่ 5	
17.5	3.51	4.04	3.32	3.55	3.59	3.52
17.0	4.09	4.21	3.47	4.13	4.20	4.10
16.5	4.27	4.40	4.03	4.30	4.40	4.28
16.0	4.46	5.02	4.20	4.52	5.06	4.49
15.5	5.07	5.24	4.39	5.11	5.17	5.08
15.0	5.28	5.47	4.56	5.32	5.36	5.20
14.5	5.51	6.12	5.26	5.55	6.01	5.53
14.0	6.16	6.41	6.00	6.20	6.17	6.19
13.5	6.44	7.11	6.31	6.48	6.43	6.47
13.0	7.15	7.40	7.07	7.27	7.17	7.21
12.5	7.47	8.17	7.31	7.56	7.50	7.52
12.0	8.22	8.54	8.13	8.31	8.21	8.28
11.5	9.02	9.39	8.52	9.16	8.54	9.09
11.0	9.49	10.25	9.29	10.00	9.31	9.51
10.5	10.45	11.23	10.16	10.55	10.20	10.44
10.0	11.46	12.26	11.24	11.41	11.33	11.46
9.5	13.08	13.50	12.38	12.59	12.57	13.06
9.0	15.22	15.09	15.02	15.43	15.19	15.19
8.5	17.01	16.51	16.41	17.10	17.09	16.58
8.0	18.51	19.17	18.33	18.57	18.47	18.53
7.5	20.50	22.13	20.14	21.02	21.03	21.04
7.0	23.01	25.12	22.41	23.11	24.12	23.39
6.5	26.53	28.24	25.59	27.09	26.49	27.03
6.0	33.25	34.01	33.02	35.00	34.31	34.00
5.5	48.21	48.53	48.11	49.31	49.15	48.50
5.0	61.25	64.30	62.51	65.01	63.54	63.32

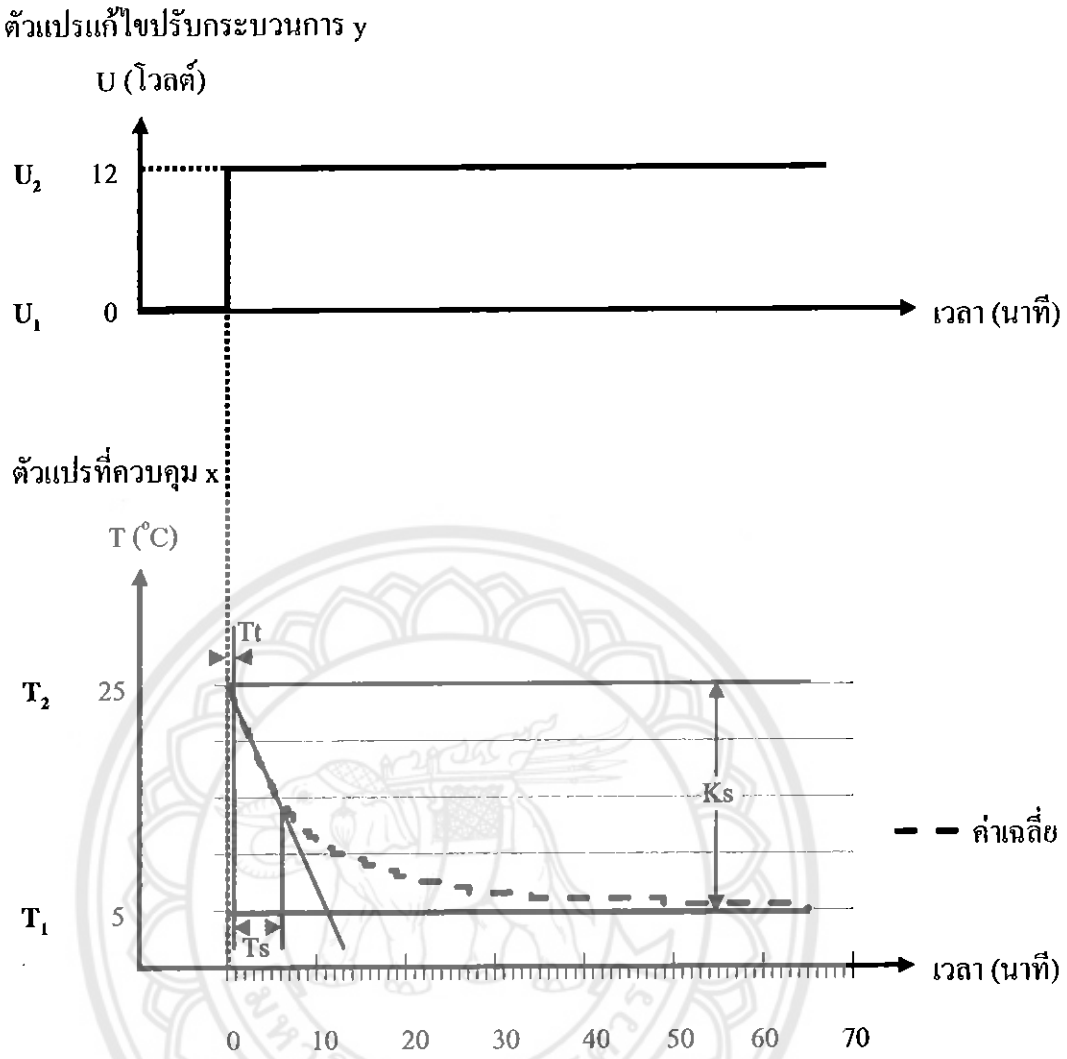


รูปที่ 4.1 กราฟแสดงผลระยะเวลาที่อุณหภูมิมีการเปลี่ยนแปลง

จากผลการทดลองหาระยะเวลาที่ตู้แช่วัคซีนควบคุมด้วยไมโครคอนโทรเลอร์ พบว่าค่าเฉลี่ยของเวลาการเปลี่ยนแปลงไม่แตกต่างกันมากเกินไปเมื่อเปรียบเทียบจากเวลาครั้งที่ 1 ถึง 5 ดังนั้น สามารถที่จะนำค่าเฉลี่ยที่ได้มาใช้ในการคำนวณหาค่าพารามิเตอร์พีไอดีได้

4.2 การคำนวณหาค่าพารามิเตอร์ K_p, K_i และ K_d

การคำนวณหาค่าพารามิเตอร์ K_p, K_i และ K_d โดยจะนำค่าเฉลี่ยที่ได้จากตารางที่ 4.1 มาคำนวณหาค่าพารามิเตอร์โดยการวิเคราะห์จากผลตอบสนองต่อสัญญาณขั้นบันไดโดยใช้วิธี Ziegler-Nichols โดยผลการทดลองจะเป็นตามตารางที่ 4.2 , 4.3 และ 4.4



รูปที่ 4.2 การหาคุณสมบัติของระบบโดยการวิเคราะห์จากผลตอบสนองต่อสัญญาณขั้นบันได

ตารางที่ 4.2 การคำนวณหาคุณสมบัติของระบบโดยการวิเคราะห์จากผลตอบสนองต่อสัญญาณขั้นบันไดโดยใช้วิธี Ziegler-Nichols

Tt	Ts	Ks
$Tt = Time_1 - Time_0$	$Ts = Time_2 - Time_1$	$Ks = \frac{T_2 - T_1}{U_2 - U_1}$
$Tt = 16 - 0$	$Ts = 360 - 16$	$Ks = \frac{25 - 5}{12 - 0}$
$Tt = 16$	$Ts = 344$	$Ks = 1.667$

ตารางที่ 4.3 การคำนวณหาค่าพารามิเตอร์โดยการวิเคราะห์จากผลตอบสนองต่อสัญญาณขั้นบันไดโดยใช้วิธี Ziegler-Nichols

Type of Controller	K_p	Tn	Tv
P	$K_p = \frac{Ts}{Ks \times Tt}$ $K_p = \frac{344}{1.667 \times 16}$ $K_p = 12.900$	-	-
PI	$K_p = 0.9\frac{Ts}{Ks \cdot Tt}$ $K_p = 0.9\frac{344}{1.667 \times 16}$ $K_p = 11.610$	$Tn = 3.3\ Tt$ $Tn = 3.3 \times 16$ $Tn = 52.800$	-
PID	$K_p = 1.2\frac{Ts}{Ks \cdot Tt}$ $K_p = 1.2\frac{344}{1.667 \times 16}$ $K_p = 15.480$	$Tn = 2\ Tt$ $Tn = 2 \times 16$ $Tn = 32$	$Tv = 0.5\ Tt$ $Tv = 0.5 \times 16$ $Tv = 8$

ตารางที่ 4.4 การคำนวณหาค่าสัมประสิทธิ์ของตัวควบคุม และค่าคงที่ของเวลา

การแปลงจากค่า K_p, Tv, Tn ไปเป็น K_i, K_d	การแปลงจากค่า K_p, K_i, K_d ไปเป็น Tn, Tv
$K_i = \frac{K_p}{Tn}$ $K_i = \frac{15.480}{32}$ $K_i = 0.484$	$Tn = \frac{K_p}{K_i}$ $Tn = \frac{15.480}{0.484}$ $Tn = 32$
$K_d = K_p \times Tv$ $K_d = 15.480 \times 8$ $K_d = 123.840$	$Tv = \frac{K_d}{K_p}$ $Tv = \frac{123.840}{15.480}$ $Tv = 8$

จากการคำนวณหาค่าพารามิเตอร์ K_p, K_i และ K_d ที่ได้จากตารางที่ 4.2, 4.3 และ 4.4 พบว่า ค่าพารามิเตอร์ K_p คือ 15.480, K_i คือ 0.484 และ K_d คือ 123.840 โดยค่าพารามิเตอร์ที่ได้จะนำไปเขียนโปรแกรมลงบนบอร์ดไมโครคอนโทรลเลอร์เพื่อควบคุมระบบตู้แช่วัคซีนควบคุมด้วยไมโครคอนโทรเลอร์

4.3 ผลการทดลอง

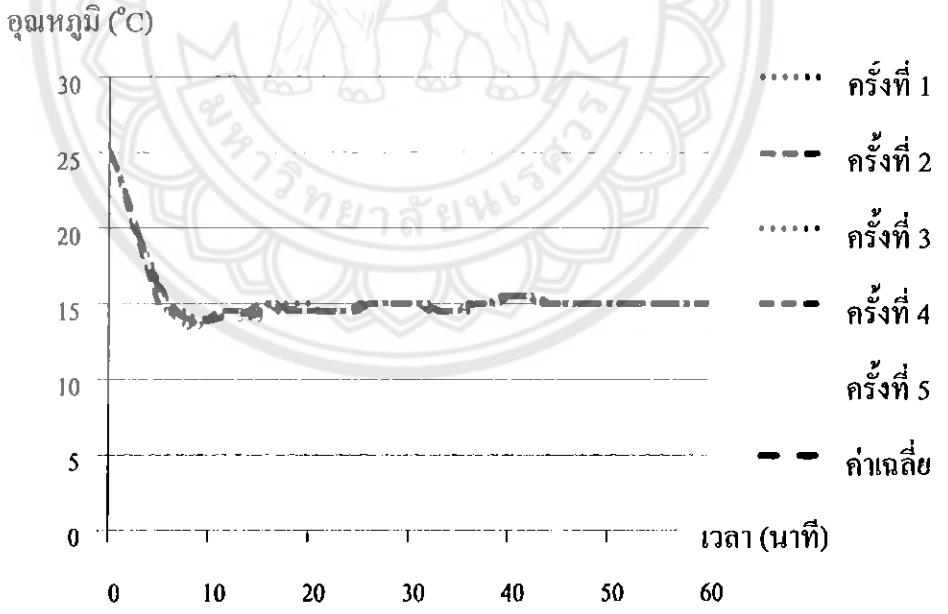
จากการทดลองหาอุณหภูมิในระยะเวลาทุกๆ 1 นาที เมื่อเซตค่าอุณหภูมิไว้ที่ 15.0, 12.0 และ10.0 องศาเซลเซียส โดยจะมีการทดลองทั้งหมด 5 ครั้ง ซึ่งผลการทดลองจะเป็นตามตารางที่ 4.5, 4.6 และ 4.7

ตารางที่ 4.5 ผลการทดลองหาอุณหภูมิในระยะเวลาทุกๆ 1 นาที เมื่อเซตอุณหภูมิ 15.0 องศาเซลเซียส

เวลา (นาที)	อุณหภูมิ (°C)					ค่าเฉลี่ย (°C)
	ครั้งที่ 1	ครั้งที่ 2	ครั้งที่ 3	ครั้งที่ 4	ครั้งที่ 5	
0	25.0	25.0	25.0	25.0	25.0	25.0
1	23.5	23.5	23.5	23.5	23.5	23.5
2	21.5	21.5	21.5	21.0	21.5	21.5
3	19.5	19.5	19.5	19.0	19.5	19.4
4	17.5	17.5	18.0	17.0	17.5	17.5
5	16.0	16.5	16.5	15.0	16.0	16.0
6	14.5	15.0	15.0	14.5	14.5	14.7
7	14.5	14.5	14.5	14.0	14.0	14.3
8	14.0	14.0	14.5	13.5	13.5	13.9
9	13.5	13.5	13.5	14.0	13.5	13.6
10	14.0	14.0	14.0	14.0	13.5	13.9
11	14.0	14.0	14.0	14.5	13.5	14.0
12	14.0	14.5	14.5	14.5	14.0	14.3
13	14.0	14.5	14.5	14.5	14.0	14.3
14	14.0	14.5	14.5	14.5	14.5	14.4
15	14.5	14.0	14.0	15.0	14.0	14.3
16	15.0	15.0	15.0	15.0	15.0	15.0
17	14.5	15.0	15.0	14.5	15.0	14.8
18	14.5	14.5	15.0	14.5	14.5	14.6
19	14.5	14.5	15.0	14.5	14.5	14.6
20	14.5	14.5	15.0	14.5	14.5	14.6

เวลา (นาที)	อุณหภูมิ (°C)					ค่าเฉลี่ย (°C)
	ครั้งที่ 1	ครั้งที่ 2	ครั้งที่ 3	ครั้งที่ 4	ครั้งที่ 5	
21	14.5	14.5	14.5	14.5	14.5	14.5
22	14.5	14.5	14.5	14.5	14.5	14.5
23	14.5	14.5	14.5	14.5	14.5	14.5
24	14.5	14.5	14.5	14.5	14.5	14.5
25	14.5	14.5	14.5	15.0	14.5	14.6
26	15.0	15.0	15.0	15.0	15.0	15.0
27	15.0	15.0	15.0	15.0	15.0	15.0
28	15.0	15.0	15.0	15.0	15.0	15.0
29	15.0	15.0	15.0	15.0	15.0	15.0
30	15.0	15.0	15.0	15.0	15.0	15.0
31	15.0	15.0	15.0	15.0	15.0	15.0
32	14.5	15.0	15.0	14.5	15.0	14.8
33	14.5	14.5	14.5	14.5	14.5	14.5
34	14.5	14.5	14.5	14.5	14.5	14.5
35	14.5	14.5	14.5	14.5	14.5	14.5
36	15.0	15.0	14.5	15.0	15.0	14.9
37	15.0	15.0	15.0	15.0	15.0	15.0
38	15.0	15.0	15.0	15.0	15.0	15.0
39	15.0	15.0	15.0	15.5	15.0	15.1
40	15.5	15.5	15.5	15.5	15.5	15.5
41	15.5	15.5	15.5	15.5	15.5	15.5
42	15.5	15.5	15.5	15.5	15.5	15.5
43	15.0	15.5	15.5	15.0	15.5	15.3
44	15.0	15.0	15.0	15.0	15.0	15.0
45	15.0	15.0	15.0	15.0	15.0	15.0
46	15.0	15.0	15.0	15.0	15.0	15.0
47	15.0	15.0	15.0	15.0	15.0	15.0
48	15.0	15.0	15.0	15.0	15.0	15.0
49	15.0	15.0	15.0	15.0	15.0	15.0

เวลา (นาที)	อุณหภูมิ (°C)					ค่าเฉลี่ย (°C)
	ครั้งที่ 1	ครั้งที่ 2	ครั้งที่ 3	ครั้งที่ 4	ครั้งที่ 5	
50	15.0	15.0	15.0	15.0	15.0	15.0
51	15.0	15.0	15.0	15.0	15.0	15.0
52	15.0	15.0	15.0	15.0	15.0	15.0
53	15.0	15.0	15.0	15.0	15.0	15.0
54	15.0	15.0	15.0	15.0	15.0	15.0
55	15.0	15.0	15.0	15.0	15.0	15.0
56	15.0	15.0	15.0	15.0	15.0	15.0
57	15.0	15.0	15.0	15.0	15.0	15.0
58	15.0	15.0	15.0	15.0	15.0	15.0
59	15.0	15.0	15.0	15.0	15.0	15.0
60	15.0	15.0	15.0	15.0	15.0	15.0



รูปที่ 4.3 กราฟแสดงผลการเปรียบเทียบหาอุณหภูมิที่มีการเปลี่ยนแปลง

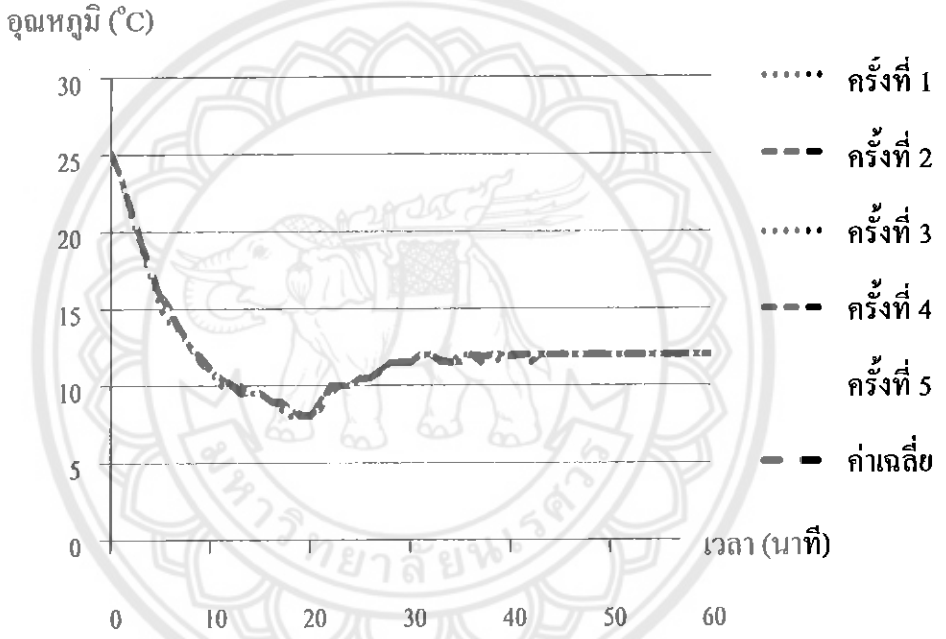
จากรูปที่ 4.3 พบว่า อุณหภูมิมีจะค่อยๆ ลดลง ทำให้เกิดโอเวอร์ชูตที่ไม่มากเกินไปโดยที่โอเวอร์ชูตไม่เกิน 2.0 องศาเซลเซียส และดูจากค่าการแกว่งของอุณหภูมิในสถานะอยู่ตัวที่น้อยที่สุด อยู่ในช่วง 14.5-15.0 องศาเซลเซียส จากการทดลองทั้ง 5 ผลการทดลอง

ตารางที่ 4.6 ผลการทดลองหาอุณหภูมิในระยะเวลาทุกๆ 1 นาที เมื่อเซตอุณหภูมิ 12.0 องศาเซลเซียส

เวลา (นาที)	อุณหภูมิ (°C)					ค่าเฉลี่ย (°C)
	ครั้งที่ 1	ครั้งที่ 2	ครั้งที่ 3	ครั้งที่ 4	ครั้งที่ 5	
0	25.0	25.0	25.0	25.0	25.0	25.0
1	23.5	23.5	23.5	23.5	23.5	23.5
2	21.0	21.5	21.0	21.5	21.5	21.3
3	19.0	19.5	19.0	19.5	19.5	19.3
4	17.0	17.5	17.0	17.5	17.5	17.3
5	15.0	16.0	15.0	16.0	16.0	15.7
6	14.5	15.0	14.0	14.5	14.5	14.6
7	13.5	13.5	13.5	13.5	13.5	13.6
8	12.5	12.5	12.5	12.5	12.5	12.6
9	12.0	11.5	11.5	12.0	12.0	11.8
10	11.0	11.0	11.0	11.0	11.0	11.1
11	10.0	10.5	10.5	10.5	10.5	10.5
12	10.0	10.0	10.0	10.0	10.0	10.1
13	9.5	9.5	9.5	10.0	10.0	9.7
14	9.5	9.5	9.5	9.5	9.5	9.5
15	9.5	9.5	9.5	9.5	9.5	9.5
16	9.0	9.0	9.0	9.0	9.0	9.0
17	8.5	9.0	9.0	9.0	8.5	8.8
18	8.0	8.5	8.5	8.5	8.5	8.4
19	8.0	8.0	8.0	8.0	8.5	8.1
20	8.0	8.0	8.0	8.0	8.0	8.0
21	9.0	8.5	9.0	9.0	8.5	8.8
22	10.0	9.5	10.0	10.0	9.5	9.8
23	10.0	10.0	10.0	10.0	10.0	10.0
24	10.0	10.0	10.0	10.0	10.0	10.0
25	10.5	10.5	10.5	10.5	10.0	10.4
26	10.5	10.5	10.5	10.5	10.5	10.5

เวลา (นาที)	อุณหภูมิ (°C)					ค่าเฉลี่ย (°C)
	ครั้งที่ 1	ครั้งที่ 2	ครั้งที่ 3	ครั้งที่ 4	ครั้งที่ 5	
27	11.0	11.0	11.0	11.0	10.5	10.9
28	11.5	11.5	11.5	11.5	11.0	11.4
29	11.5	11.5	11.5	11.5	11.5	11.5
30	11.5	11.5	11.5	11.5	11.5	11.5
31	12.0	12.0	12.0	12.0	12.0	12.0
32	12.0	12.0	12.0	12.0	12.0	12.0
33	11.5	11.5	11.5	11.5	12.0	11.6
34	11.5	11.5	11.5	11.5	11.5	11.5
35	12.0	11.5	11.5	11.5	11.5	11.6
36	12.0	12.0	12.0	12.0	12.0	12.0
37	12.0	11.5	12.0	11.5	12.0	11.8
38	12.0	12.0	12.0	12.0	12.0	12.0
39	11.5	12.0	12.0	12.0	12.0	11.9
40	12.0	12.0	12.0	12.0	11.5	11.9
41	12.0	12.0	12.0	12.0	12.0	12.0
42	12.0	12.0	12.0	11.5	12.0	12.0
43	12.0	12.0	12.0	12.0	12.0	12.0
44	12.0	12.0	12.0	12.0	12.0	12.0
45	12.0	12.0	12.0	12.0	12.0	12.0
46	12.0	12.0	12.0	12.0	12.0	12.0
47	12.0	12.0	12.0	12.0	12.0	12.0
48	12.0	12.0	12.0	12.0	12.0	12.0
49	12.0	12.0	12.0	12.0	12.0	12.0
50	12.0	12.0	12.0	12.0	12.0	12.0
51	12.0	12.0	12.0	12.0	12.0	12.0
52	12.0	12.0	12.0	12.0	12.0	12.0
53	12.0	12.0	12.0	12.0	12.0	12.0
54	12.0	12.0	12.0	12.0	12.0	12.0
55	12.0	12.0	12.0	12.0	12.0	12.0

เวลา (นาที)	อุณหภูมิ (°C)					ค่าเฉลี่ย (°C)
	ครั้งที่ 1	ครั้งที่ 2	ครั้งที่ 3	ครั้งที่ 4	ครั้งที่ 5	
56	12.0	12.0	12.0	12.0	12.0	12.0
57	12.0	12.0	12.0	12.0	12.0	12.0
58	12.0	12.0	12.0	12.0	12.0	12.0
59	12.0	12.0	12.0	12.0	12.0	12.0
60	12.0	12.0	12.0	12.0	12.0	12.0



รูปที่ 4.4 กราฟแสดงผลการเปรียบเทียบหาอุณหภูมิที่มีการเปลี่ยนแปลง

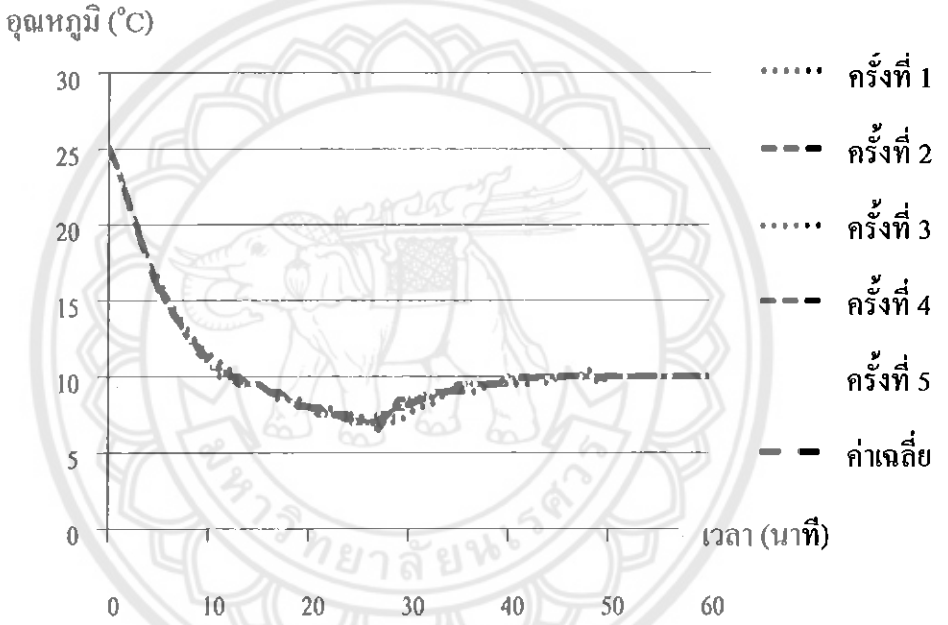
จากรูปที่ 4.4 พบว่า อุณหภูมิมีจะค่อยๆ ลดลง ทำให้เกิดโอเวอร์ชูตที่ไม่มากเกินไปโดยที่โอเวอร์ชูตไม่เกิน 4.0 องศาเซลเซียส และดูจากค่าการแกว่งของอุณหภูมิในสถานะอยู่ตัวที่น้อยที่สุด อยู่ในช่วง 11.5-12.0 องศาเซลเซียส จากการทดลองทั้ง 5 ผลการทดลอง

ตารางที่ 4.7 ผลการทดลองหาอุณหภูมิในระยะเวลาทุกๆ 1 นาที เมื่อเซตอุณหภูมิ 10.0 องศาเซลเซียส

เวลา (นาที)	อุณหภูมิ (°C)					ค่าเฉลี่ย (°C)
	ครั้งที่ 1	ครั้งที่ 2	ครั้งที่ 3	ครั้งที่ 4	ครั้งที่ 5	
0	25.0	25.0	25.0	25.0	25.0	25.0
1	23.5	23.5	23.5	23.5	23.5	23.5
2	21.5	21.5	21.0	21.5	21.0	21.3
3	19.5	19.5	19.0	19.5	19.0	19.3
4	17.5	17.5	17.0	17.5	17.0	17.3
5	16.5	16.0	15.0	16.0	15.0	15.7
6	15.0	14.5	14.5	15.0	14.0	14.6
7	14.0	13.5	13.5	13.5	13.5	13.6
8	13.0	12.5	12.5	12.5	12.5	12.6
9	12.0	12.0	12.0	11.5	11.5	11.8
10	11.5	11.0	11.0	11.0	11.0	11.1
11	11.0	10.5	10.0	10.5	10.5	10.5
12	10.5	10.0	10.0	10.0	10.0	10.1
13	10.0	10.0	9.5	9.5	9.5	9.7
14	9.5	9.5	9.5	9.5	9.5	9.5
15	9.5	9.5	9.5	9.5	9.5	9.5
16	9.0	9.0	9.0	9.0	9.0	9.0
17	8.5	9.0	9.0	9.0	8.5	8.8
18	8.5	8.5	8.5	8.5	8.5	8.5
19	8.5	8.0	8.0	8.0	8.0	8.1
20	8.0	8.0	8.0	8.0	8.0	8.0
21	8.0	7.5	8.0	8.0	7.5	7.8
22	8.0	7.5	7.5	7.5	7.5	7.6
23	7.5	7.5	7.5	7.5	7.0	7.4
24	7.5	7.0	7.0	7.5	7.0	7.2
25	7.5	7.0	7.0	7.0	7.0	7.1
26	7.0	7.0	7.0	7.0	6.5	6.9

เวลา (นาที)	อุณหภูมิ (°C)					ค่าเฉลี่ย (°C)
	ครั้งที่ 1	ครั้งที่ 2	ครั้งที่ 3	ครั้งที่ 4	ครั้งที่ 5	
27	7.0	6.5	7.0	7.5	7.5	7.1
28	7.0	7.5	7.5	7.5	8.0	7.5
29	7.0	8.0	7.5	8.5	7.5	7.7
30	7.5	8.0	8.5	8.5	8.5	8.2
31	8.0	8.5	8.5	8.5	8.5	8.4
32	8.0	8.5	8.5	9.0	9.0	8.6
33	8.5	9.0	9.0	9.0	9.0	8.9
34	9.0	9.0	9.0	9.0	9.0	9.0
35	9.0	9.0	9.0	9.5	9.5	9.2
36	9.0	9.5	9.5	9.5	9.5	9.4
37	9.0	9.5	9.5	9.5	9.5	9.4
38	9.5	9.5	9.5	9.5	9.5	9.5
39	9.5	9.5	9.5	9.5	9.5	9.5
40	9.5	9.5	9.5	10.0	10.0	9.7
41	9.5	10.0	10.0	10.0	10.0	9.9
42	9.5	10.0	10.0	10.0	10.0	9.9
43	10.0	10.0	10.0	10.0	9.5	9.9
44	10.0	10.0	9.5	10.0	10.0	9.9
45	10.0	10.0	10.0	10.0	10.0	10.0
46	10.0	10.0	10.0	10.0	10.0	10.0
47	10.0	10.0	10.0	10.0	10.0	10.0
48	10.5	10.0	10.0	10.0	10.0	10.1
49	9.5	10.0	10.0	10.0	10.0	9.9
50	10.0	10.0	10.0	10.0	10.0	10.0
51	10.0	10.0	10.0	10.0	10.0	10.0
52	10.0	10.0	10.0	10.0	10.0	10.0
53	10.0	10.0	10.0	10.0	10.0	10.0
54	10.0	10.0	10.0	10.0	10.0	10.0
55	10.0	10.0	10.0	10.0	10.0	10.0

เวลา (นาที)	อุณหภูมิ (°C)					ค่าเฉลี่ย (°C)
	ครั้งที่ 1	ครั้งที่ 2	ครั้งที่ 3	ครั้งที่ 4	ครั้งที่ 5	
56	10.0	10.0	10.0	10.0	10.0	10.0
57	10.0	10.0	10.0	10.0	10.0	10.0
58	10.0	10.0	10.0	10.0	10.0	10.0
59	10.0	10.0	10.0	10.0	10.0	10.0
60	10.0	10.0	10.0	10.0	10.0	10.0



รูปที่ 4.5 กราฟแสดงผลการเปรียบเทียบหาอุณหภูมิที่มีการเปลี่ยนแปลง

จากรูปที่ 4.5 พบว่า อุณหภูมิมีจะค่อยๆ ลดลง ทำให้เกิดโอเวอร์ชูตที่ไม่มากเกินไปโดยที่โอเวอร์ชูตไม่เกิน 3.0 องศาเซลเซียส และดูจากค่าการแกว่งของอุณหภูมิในสถานะอยู่ตัวที่น้อยที่สุด อยู่ในช่วง 9.5-10.0 องศาเซลเซียส จากการทดลองทั้ง 5 ผลการทดลอง

บทที่ 5
สรุปผลการทดลอง

จากการศึกษาและทำการทดลอง ในบทนี้จะเป็นเรื่องเกี่ยวกับการสรุปผลการทดลอง รวมถึงปัญหาและการแก้ไขข้อเสนอแนะเพื่อเป็นแนวทางในการพัฒนาโครงงานต่อไป

5.1 สรุปผลการทดลอง

จากการทดลองตู้แช่วัคซีนควบคุมด้วยไมโครคอนโทรลเลอร์ สามารถทำงานได้จริงอย่างมีประสิทธิภาพ โดยมีไมโครคอนโทรลเลอร์ไปควบคุมอุปกรณ์ต่างๆได้ดี การทำงานของตู้แช่วัคซีนควบคุมด้วยไมโครคอนโทรเลอร์ พบว่า ค่าอุณหภูมิจะค่อยๆขึ้นไปสู่จุดเซตพ้อยท์ และไม่เกิดค่าโอเวอร์ชูต เนื่องจากอัตราขยาย K_p ที่มีค่าเหมาะสมดังวิธีการของ Ziegler-Nichols และค่าอุณหภูมิที่ได้ก็จะมีค่าในสถานะอยู่ตัวอยู่ที่ 15.0, 12.0 และ 10.0 องศาเซลเซียส อัตราขยายของ K_d ก็จะชะลออัตราการเปลี่ยนแปลงของสัญญาณขาออกของระบบควบคุม และด้วยผลนี้จะช่วยให้ระบบควบคุมเข้าสู่จุดที่ต้องการได้ ดังนั้นผลของอัตราขยายของค่า K_d จะใช้ในการลดขนาดของโอเวอร์ชูตที่เกิดจากผลอัตราขยายของ K_i และทำให้เสถียรภาพของการรวมกันของระบบควบคุมดีขึ้น

ในโครงงานนี้ตู้แช่วัคซีนควบคุมด้วยไมโครคอนโทรลเลอร์สามารถการควบคุมอุณหภูมิได้ตามเป้าหมายที่วางไว้ในโครงงาน แต่ยังต้องการพัฒนาคุณภาพอีกหลายอย่างเพื่อการควบคุมอุณหภูมิด้วย ระบบทำความเย็นได้ดียิ่งขึ้น

5.2 ปัญหาและการแก้ไข

1. การประกอบอุปกรณ์ เนื่องจากอุปกรณ์บางชิ้นทำขึ้นเอง เวลาประกอบตู้แช่วัคซีนมีปัญหาในการซีลกาวซิลิโคน ทำให้ความเย็นซึมออกมาจากตู้แช่วัคซีน ดังนั้นจึงแก้ไขโดยการใช้กาวอีพ็อกซี่ยึดติดเพื่อป้องกันความเย็นซึมออกจากตู้แช่วัคซีน

2. ระบบทำความเย็นทำงานมากยิ่งขึ้น เนื่องจากปัญหาฝาตู้แช่วัคซีนไม่แน่น ดังนั้นจึงแก้ไขโดยการใช้เทปกาวยึดติดกับฝาตู้แช่วัคซีนเข้ากับตู้แช่วัคซีน ซึ่งทำให้ระบบทำความเย็นทำงานไม่หนักเกินไป

3. การประกอบถ้าใช้แรงกดมากไป จะทำให้เพลเทียร์เสียหายได้ ดังนั้นจึงแก้ไขได้โดยการขยายช่องบรรจุเพลเทียร์ให้มีขนาดใหญ่ขึ้น เพื่อลดแรงกดขณะประกอบเพลเทียร์

4. การคลายความร้อนไม่ดี ส่งผลให้เพลเทียร์ทำงานไม่เต็มประสิทธิภาพ ดังนั้นจึงแก้ไขโดยการใช้ชุดระบายความร้อนชนิดสแตนเลส และพัดลมระบายความร้อนที่มีความเร็วรอบสูง

5. เพลเทียร์มีการใช้กระแสไฟจำนวนมาก ส่งผลให้แหล่งจ่ายไฟไม่เพียงพอ ดังนั้นจึงแก้ไขปัญหาโดยการใช้แหล่งจ่ายไฟขนาด 30 แอมป์ ซึ่งถือว่าเพียงพอกับโครงงานนี้

5.3 ข้อเสนอแนะ

1. ตู้แช่วัคซีนมีขนาดเล็กอาจทำให้แช่วัคซีนได้จำนวนจำกัด หากมีความต้องการพื้นที่แช่วัคซีนมากกว่านี้จำเป็นที่จะต้องเพิ่มขนาดตู้แช่วัคซีน

2. หากต้องการให้ตู้แช่วัคซีนสามารถทำความเย็นอย่างรวดเร็ว ควรเพิ่มจำนวนเพลเทียร์มากกว่านี้ แต่จะส่งผลถึงเรื่องต้นทุนที่สูงขึ้น

3. ควรมีการทำความสะอาดภายในตู้แช่วัคซีนหลังจากใช้งานเสร็จสิ้น เพื่อทำให้ปราศจากเชื้อโรคในการแพร่กระจายสู่วัคซีนที่จะนำมาแช่ต่อไป

4. การใช้งานตู้แช่วัคซีน อุปกรณ์บางอย่างเป็นอุปกรณ์เฉพาะงาน ควรศึกษาการทำงานอย่างละเอียด เพื่อป้องกันไม่เกิดความเสียหายกับตู้แช่วัคซีน

5. การเคลื่อนย้ายตู้แช่วัคซีน ควรระมัดระวังมากที่สุด เนื่องจากเพลเทียร์ภายในตู้แช่วัคซีนอาจแตกหักขึ้นได้

เอกสารอ้างอิง

[1] ประจิน พลังสันติกุล. (2549) . C Programming for AVR Microcontroller and WinAVR (C Compiler).กรุงเทพฯ : จัดพิมพ์โดยบริษัท อินโนเวตีฟ เอ็กเพอริเมนต์ จำกัด

[2] ประจิน พลังสันติกุล. (2551) . Apply C Programming for AVR Microcontroller Book 2. กรุงเทพฯ : จัดพิมพ์โดยบริษัท อินโนเวตีฟ เอ็กเพอริเมนต์ จำกัด

[3] บุญชัย กิ่งรุ้งเพชร. (2543) . Protel99. กรุงเทพฯ : จัดพิมพ์โดยบริษัท แอดคลา เอ็นจิเนียริ่ง จำกัด

[4] ประภาพร ช่างไม้. (2537) . คู่มือการเขียนโปรแกรมภาษา C. นนทบุรี : จัดพิมพ์โดยบริษัท คอมฟอร์ม จำกัด

[5] นิรุธ อำนวยศิลป์. (2550) . คู่มือการเขียนโปรแกรมภาษา C. กรุงเทพฯ : จัดพิมพ์โดยบริษัท โปรวิชั่น จำกัด

[6] วิบูลย์ แสงวีระพันธุ์ศิริ. (2548) . การควบคุมระบบพลศาสตร์.กรุงเทพฯ : จัดพิมพ์โดยสำนักพิมพ์จุฬาลงกรณ์มหาวิทยาลัย

[7] อนุชา หิรัญวัฒน์, นฤพนธ์ พนากุลชัยวิทย์, และสมชัย ตรีรัตนจารุ. (2551) . การควบคุมอัตโนมัติ และการประยุกต์ใช้งานพีแอลซี (ชั้นกลาง). นนทบุรี : จัดพิมพ์โดย ห้างหุ้นส่วนจำกัด ธนินชัย.

ภาคผนวก ก

รายละเอียดของ โปรแกรม

```
#include <mega64.h>
#include <delay.h>
#include <string.h>
#include <stdio.h>
#include <stdlib.h>  // CHAR TO INT
#include <math.h>

#define LED1       PORTA.1
#define LED2       PORTA.2
#define Relay_FAN   PORTA.0
#define Buzzer     PORTA.3
#define SW_Up      PINE.4
#define SW_Down    PINE.5

char Show_lcd[16];
int LDR;
int Data_Ds1820;
int Timer_ovf;

//////////////////////////////////////////// Ds1820 ////////////////////////////////////
unsigned char devices=0;
#define MAX_DEVICES 8
unsigned char rom_codes[MAX_DEVICES][9];
float temp=0;
eeprom float Set_Temp = 0;

//////////////////////////////////////////// PID  ////////////////////////////////////////////
float  kp,ki,kd,dt;
float error_PID ;
float actual_position,integral ;
eeprom float setpoint=5.0;
float derivative, previous_error;
```

```
int j;
int Start_Buzzer=0;
int frequency=100,duty=0;
// 1 Wire Bus functions
#asm
  .equ __w1_port=0x03 ;PORTE
  .equ __w1_bit=2
#endasm
#include <1wire.h>

// DS1820 Temperature Sensor functions
#include <ds1820.h>

// Alphanumeric LCD Module functions
#asm
  .equ __lcd_port=0x12 ;PORTD
#endasm
#include <lcd.h>

#define RXB8 1
#define TXB8 0
#define UPE 2
#define OVR 3
#define FE 4
#define UDRE 5
#define RXC 7
#define FRAMING_ERROR (1<<FE)
#define PARITY_ERROR (1<<UPE)
#define DATA_OVERRUN (1<<OVR)
#define DATA_REGISTER_EMPTY (1<<UDRE)
#define RX_COMPLETE (1<<RXC)
```

```
// USART0 Receiver buffer
#define RX_BUFFER_SIZE0 8
char rx_buffer0[RX_BUFFER_SIZE0];

#if RX_BUFFER_SIZE0<256
unsigned char rx_wr_index0,rx_rd_index0,rx_counter0;
#else
unsigned int rx_wr_index0,rx_rd_index0,rx_counter0;
#endif

// This flag is set on USART0 Receiver buffer overflow
bit rx_buffer_overflow0;

// USART0 Receiver interrupt service routine
interrupt [USART0_RXC] void usart0_rx_isr(void)
{
char status,data;
status=UCSR0A;
data=UDR0;
if ((status & (FRAMING_ERROR | PARITY_ERROR | DATA_OVERRUN))==0)
  {
  rx_buffer0[rx_wr_index0]=data;
  if (++rx_wr_index0 == RX_BUFFER_SIZE0) rx_wr_index0=0;
  if (++rx_counter0 == RX_BUFFER_SIZE0)
    {
    rx_counter0=0;
    rx_buffer_overflow0=1;
    };
  };
}

#ifndef _DEBUG_TERMINAL_IO_
```

```
// Get a character from the USART0 Receiver buffer
#define _ALTERNATE_GETCHAR_
#pragma used+
char getchar(void)
{
char data;
while (rx_counter0==0);
data=rx_buffer0[rx_rd_index0];
if (++rx_rd_index0 == RX_BUFFER_SIZE0) rx_rd_index0=0;
#asm("cli")
--rx_counter0;
#asm("sei")
return data;
}
#pragma used-
#endif

// USART0 Transmitter buffer
#define TX_BUFFER_SIZE0 8
char tx_buffer0[TX_BUFFER_SIZE0];

#if TX_BUFFER_SIZE0<256
unsigned char tx_wr_index0,tx_rd_index0,tx_counter0;
#else
unsigned int tx_wr_index0,tx_rd_index0,tx_counter0;
#endif

// USART0 Transmitter interrupt service routine
interrupt [USART0_TXC] void usart0_tx_isr(void)
{
if (tx_counter0)
  {
```

```
  --tx_counter0;
  UDR0=tx_buffer0[tx_rd_index0];
  if (++tx_rd_index0 == TX_BUFFER_SIZE0) tx_rd_index0=0;
  };
}

#ifndef _DEBUG_TERMINAL_IO_
// Write a character to the USART0 Transmitter buffer
#define _ALTERNATE_PUTCHAR_
#pragma used+
void putchar(char c)
{
while (tx_counter0 == TX_BUFFER_SIZE0);
#asm("cli")
if (tx_counter0 || ((UCSR0A & DATA_REGISTER_EMPTY)==0))
  {
  tx_buffer0[tx_wr_index0]=c;
  if (++tx_wr_index0 == TX_BUFFER_SIZE0) tx_wr_index0=0;
  ++tx_counter0;
  }
else
  UDR0=c;
#asm("sei")
}
#pragma used-
#endif

// Standard Input/Output functions
#include <stdio.h>

// Timer 0 overflow interrupt service routine
interrupt [TIM0_OVF] void timer0_ovf_isr(void)
```

```
{
  Timer_ovf++;
  if(Timer_ovf > 1000)
  {
    if((!SW_Up)   && (setpoint < 100.0))  { setpoint += 0.5; LED2 = !LED2; Timer_ovf=0;}
    if((!SW_Down) && (setpoint > 0.0))    { setpoint -= 0.5; LED2 = !LED2; Timer_ovf=0;}
    if(Timer_ovf > 1000)  Timer_ovf = 1000;
  }
   // if(Start_Buzzer == 1)  Buzzer = !Buzzer;
   // else Buzzer = 0;

}
#define ADC_VREF_TYPE 0x00
// Read the AD conversion result
unsigned int read_adc(unsigned char adc_input)
{
ADMUX=adc_input|ADC_VREF_TYPE;
// Start the AD conversion
ADCSRA|=0x40;
// Wait for the AD conversion to complete
while ((ADCSRA & 0x10)==0);
ADCSRA|=0x10;
return ADCW;
}
void main(void)
{
// Input/Output Ports initialization
// Port A initialization
// Func7=In Func6=In Func5=In Func4=In Func3=In Func2=In Func1=In Func0=In
// State7=T State6=T State5=T State4=T State3=T State2=T State1=T State0=T
PORTA=0x00;
PORTA.3 = 1;
```

```
DDRA=0xFF;

// Port B initialization
// Func7=In Func6=Out Func5=Out Func4=In Func3=In Func2=In Func1=In Func0=In
// State7=T State6=0 State5=0 State4=T State3=T State2=T State1=T State0=T
PORTB=0x00;
DDRB=0x60;

// Port C initialization
// Func7=In Func6=In Func5=In Func4=In Func3=In Func2=In Func1=In Func0=In
// State7=T State6=T State5=T State4=T State3=T State2=T State1=T State0=T
PORTC=0x00;
DDRC=0x00;

// Port D initialization
// Func7=In Func6=In Func5=In Func4=In Func3=In Func2=In Func1=In Func0=In
// State7=T State6=T State5=T State4=T State3=T State2=T State1=T State0=T
PORTD=0x00;
DDRD=0x00;

// Port E initialization
// Func7=In Func6=In Func5=In Func4=In Func3=In Func2=In Func1=In Func0=In
// State7=T State6=T State5=T State4=T State3=T State2=T State1=T State0=T
PORTE=0xF0;
DDRE=0x00;

// Port F initialization
// Func7=In Func6=In Func5=In Func4=In Func3=In Func2=In Func1=In Func0=In
// State7=T State6=T State5=T State4=T State3=T State2=T State1=T State0=T
PORTF=0x00;
DDRF=0x00;
```

```
// Port G initialization
// Func4=In Func3=In Func2=In Func1=In Func0=In
// State4=T State3=T State2=T State1=T State0=T
PORTG=0x00;
DDRG=0x00;

// Timer/Counter 0 initialization
// Clock source: System Clock
// Clock value: 1000.000 kHz
// Mode: Normal top=FFh
// OC0 output: Disconnected
ASSR=0x00;
TCCR0=0x02;
TCNT0=0x00;
OCR0=0x00;

// Timer/Counter 1 initialization
// Clock source: System Clock
// Clock value: 1000.000 kHz
// Mode: Ph. & fr. cor. PWM top=ICR1
// OC1A output: Non-Inv.
// OC1B output: Non-Inv.
// OC1C output: Discon.
// Noise Canceler: Off
// Input Capture on Falling Edge
// Timer 1 Overflow Interrupt: Off
// Input Capture Interrupt: Off
// Compare A Match Interrupt: Off
// Compare B Match Interrupt: Off
// Compare C Match Interrupt: Off
TCCR1A=0xA0;
TCCR1B=0x12;
```

```
TCNT1H=0x00;
TCNT1L=0x00;
// ICR1H=0x00;
// ICR1L=0x00;
// OCR1AH=0x00;
// OCR1AL=0x00;
// OCR1BH=0x00;
// OCR1BL=0x00;
// OCR1CH=0x00;
// OCR1CL=0x00;

// Timer/Counter 2 initialization
// Clock source: System Clock
// Clock value: Timer 2 Stopped
// Mode: Normal top=FFh
// OC2 output: Disconnected
TCCR2=0x00;
TCNT2=0x00;
OCR2=0x00;

// Timer/Counter 3 initialization
// Clock source: System Clock
// Clock value: Timer 3 Stopped
// Mode: Normal top=FFFFh
// Noise Canceler: Off
// Input Capture on Falling Edge
// OC3A output: Discon.
// OC3B output: Discon.
// OC3C output: Discon.
// Timer 3 Overflow Interrupt: Off
// Input Capture Interrupt: Off
// Compare A Match Interrupt: Off
```

```
// Compare B Match Interrupt: Off
// Compare C Match Interrupt: Off
TCCR3A=0x00;
TCCR3B=0x00;
TCNT3H=0x00;
TCNT3L=0x00;
ICR3H=0x00;
ICR3L=0x00;
OCR3AH=0x00;
OCR3AL=0x00;
OCR3BH=0x00;
OCR3BL=0x00;
OCR3CH=0x00;
OCR3CL=0x00;

// External Interrupt(s) initialization
// INT0: Off
// INT1: Off
// INT2: Off
// INT3: Off
// INT4: Off
// INT5: Off
// INT6: Off
// INT7: Off
EICRA=0x00;
EICRB=0x00;
EIMSK=0x00;

// Timer(s)/Counter(s) Interrupt(s) initialization
TIMSK=0x01;
ETIMSK=0x00;
```

```
// USART0 initialization
// Communication Parameters: 8 Data, 1 Stop, No Parity
// USART0 Receiver: On
// USART0 Transmitter: On
// USART0 Mode: Asynchronous
// USART0 Baud rate: 9600
UCSR0A=0x00;
UCSR0B=0xD8;
UCSR0C=0x06;
UBRR0H=0x00;
UBRR0L=0x33;

// Analog Comparator initialization
// Analog Comparator: Off
// Analog Comparator Input Capture by Timer/Counter 1: Off
ACSR=0x80;
SFIOR=0x00;

// ADC initialization
// ADC Clock frequency: 1000.000 kHz
// ADC Voltage Reference: AREF pin
ADMUX=ADC_VREF_TYPE;
ADCSRA=0x83;

// 1 Wire Bus initialization
w1_init();

// LCD module initialization
lcd_init(16);

// Global enable interrupts
#asm("sei")
```

```
Buzzer = 1;

lcd_gotoxy(0,0);
sprintf(Show_lcd, "Vaccine Refriger");
lcd_puts(Show_lcd);
lcd_gotoxy(0,1);
sprintf( Show_lcd,"Coolness Control");
lcd_puts(Show_lcd);
printf("\f\r\nVaccine Refrigerator Control With Microcontroller");
delay_ms(1000);

Buzzer = 0;
delay_ms(50);
Buzzer = 1;
Relay_FAN = 1;

kp = 15.480;
ki = 0.484;
kd = 123.840;
dt = 1;

while (1)
    {
        devices = w1_search(0xf0,rom_codes);
        Data_Ds1820 = ds1820_temperature_10(&rom_codes[0][0]);
        temp = (float)Data_Ds1820/10.0;
        LED1   = !LED1;
        LDR = 0;
        for (j=0;j<20;j++)
        {
          LDR = LDR + read_adc(0);
        }
```

```
LDR = LDR/20;

 if(LDR > 50)  Buzzer = !Buzzer; //Start_Buzzer = 1;
else Buzzer = 1 ;// Start_Buzzer = 0;

actual_position = temp;
error_PID =  actual_position - setpoint;
integral = integral + (error_PID*dt);
derivative = (error_PID - previous_error)/dt;
duty = (int)(kp*error_PID) + (int)(ki*integral) + (int)(kd*derivative);
if( duty >= 100 )  duty = 100;
if( duty < 0 )    duty = 0;

ICR1 = frequency;     // F
OCR1A = duty; //  duty
OCR1B = duty; //  duty

previous_error = error_PID ;

lcd_gotoxy(0,0);
if(temp < -10.0)
{
   sprintf(Show_lcd,"   No  Sensor   ");
   lcd_puts(Show_lcd);
   lcd_gotoxy(0,1);
   sprintf(Show_lcd," Press  Connect ");
   lcd_puts(Show_lcd);
   printf(" No Sensor Press Connect\r\n" );
}
else
{
   sprintf(Show_lcd,"T:%2.1fC  S:%2.1fC ",temp,setpoint);
```

```
            lcd_puts(Show_lcd);
            lcd_gotoxy(0,1);
            sprintf(Show_lcd,"D:%i%%  E:%1.1f  ",duty,error_PID);
            lcd_puts(Show_lcd);
            printf(" Error:%1.1f Integral:%1.1f  Derivative:%1.1f  Duty:%i%%  T:%2.1fC
S:%2.1fC\r\n"
            ,error_PID,integral,derivative,duty,temp,setpoint);
        }
    };
}
```

ภาคผนวก ข

รายละเอียดของ ATmaga 64

Features

- High-performance, Low-power AVR® 8-bit Microcontroller
- Advanced RISC Architecture
  - 130 Powerful Instructions – Most Single Clock Cycle Execution
  - 32 x 8 General Purpose Working Registers + Peripheral Control Registers
  - Fully Static Operation
  - Up to 16 MIPS Throughput at 16 MHz
  - On-chip 2-cycle Multiplier
- Non-volatile Program and Data Memories
  - 64K Bytes of In-System Reprogrammable Flash
    Endurance: 10,000 Write/Erase Cycles
  - Optional Boot Code Section with Independent Lock Bits
    In-System Programming by On-chip Boot Program
    True Read-While-Write Operation
  - 2K Bytes EEPROM
    Endurance: 100,000 Write/Erase Cycles
  - 4K Bytes Internal SRAM
  - Up to 64K Bytes Optional External Memory Space
  - Programming Lock for Software Security
  - SPI Interface for In-System Programming
- JTAG (IEEE std. 1149.1 Compliant) Interface
  - Boundary-scan Capabilities According to the JTAG Standard
  - Extensive On-chip Debug Support
  - Programming of Flash, EEPROM, Fuses, and Lock Bits through the JTAG Interface
- Peripheral Features
  - Two 8-bit Timer/Counters with Separate Prescalers and Compare Modes
  - Two Expanded 16-bit Timer/Counters with Separate Prescaler, Compare Mode, and Capture Mode
  - Real Time Counter with Separate Oscillator
  - Two 8-bit PWM Channels
  - 6 PWM Channels with Programmable Resolution from 1 to 16 Bits
  - 8-channel, 10-bit ADC
    8 Single-ended Channels
    7 Differential Channels
    2 Differential Channels with Programmable Gain (1x, 10x, 200x)
  - Byte-oriented Two-wire Serial Interface
  - Dual Programmable Serial USARTs
  - Master/Slave SPI Serial Interface
  - Programmable Watchdog Timer with On-chip Oscillator
  - On-chip Analog Comparator
- Special Microcontroller Features
  - Power-on Reset and Programmable Brown-out Detection
  - Internal Calibrated RC Oscillator
  - External and Internal Interrupt Sources
  - Six Sleep Modes: Idle, ADC Noise Reduction, Power-save, Power-down, Standby and Extended Standby
  - Software Selectable Clock Frequency
  - ATmega103 Compatibility Mode Selected by a Fuse
  - Global Pull-up Disable
- I/O and Packages
  - 53 Programmable I/O Lines
  - 64-lead TQFP and 64-pad QFN/MLF
- Operating Voltages
  - 2.7 - 5.5V for ATmega64L
  - 4.5 - 5.5V for ATmega64
- Speed Grades
  - 0 - 8 MHz for ATmega64L
  - 0 - 16 MHz for ATmega64

8-bit AVR® Microcontroller with 64K Bytes In-System Programmable Flash

ATmega64
ATmega64L

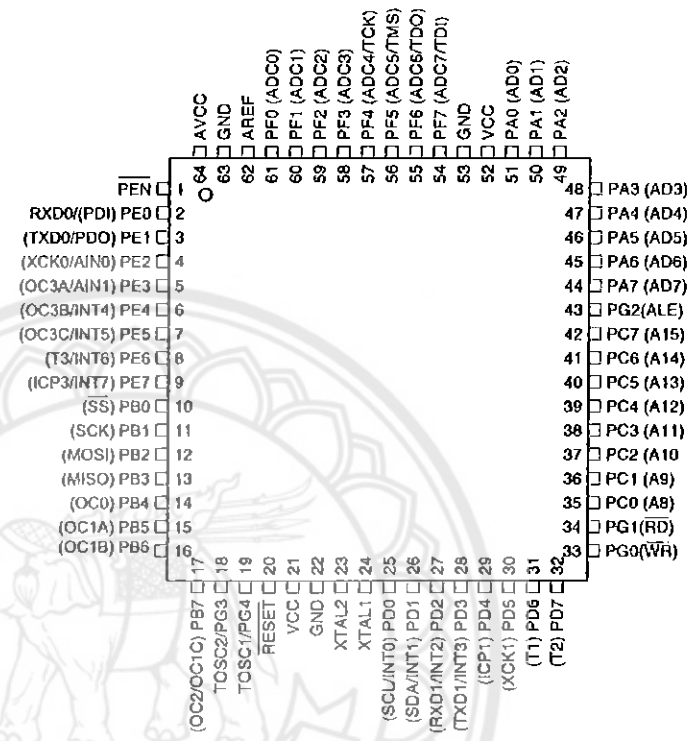
Summary

2490JS–AVR–03/05

Pin Configuration

Figure 1. Pinout ATmega64

TQFP/MLF



Note: The bottom pad under the QFN/MLF package should be soldered to ground.

Disclaimer

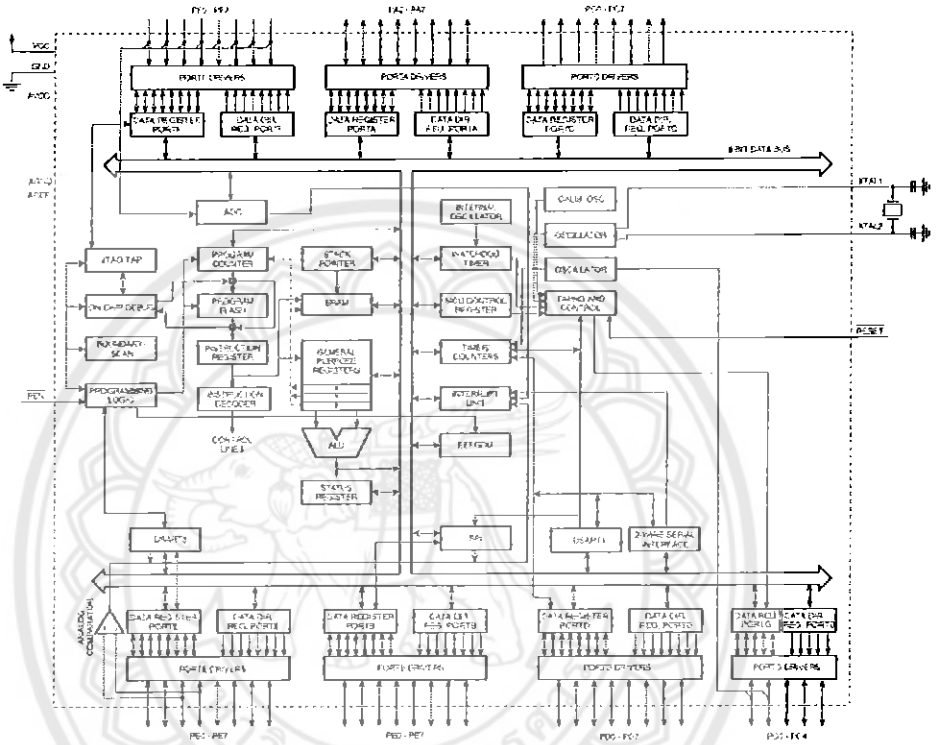
Typical values contained in this data sheet are based on simulations and characterization of other AVR microcontrollers manufactured on the same process technology. Min and Max values will be available after the device is characterized.

Overview

The ATmega64 is a low-power CMOS 8-bit microcontroller based on the AVR enhanced RISC architecture. By executing powerful instructions in a single clock cycle, the ATmega64 achieves throughputs approaching 1 MIPS per MHz, allowing the system designer to optimize power consumption versus processing speed.

Block Diagram

Figure 2. Block Diagram



The AVR core combines a rich instruction set with 32 general purpose working registers. All the 32 registers are directly connected to the Arithmetic Logic Unit (ALU), allowing two independent registers to be accessed in one single instruction executed in one clock cycle. The resulting architecture is more code efficient while achieving throughputs up to ten times faster than conventional CISC microcontrollers.

ATmega103 Compatibility Mode

By programming the M103C Fuse, the ATmega64 will be compatible with the ATmega103 regards to RAM, I/O pins and Interrupt Vectors as described above. However, some new features in ATmega64 are not available in this compatibility mode, these features are listed below:

- One USART instead of two, asynchronous mode only. Only the eight least significant bits of the Baud Rate Register is available.
- One 16 bits Timer/Counter with two compare registers instead of two 16 bits Timer/Counters with three compare registers.
- Two-wire serial interface is not supported.
- Port G serves alternate functions only (not a general I/O port).
- Port F serves as digital input only in addition to analog input to the ADC.
- Boot Loader capabilities is not supported.
- It is not possible to adjust the frequency of the internal calibrated RC Oscillator.
- The External Memory Interface can not release any Address pins for general I/O, neither configure different wait states to different External Memory Address sections.
- Only EXTRF and PORF exist in the MCUCSR Register.
- No timed sequence is required for Watchdog Timeout change.
- Only low-level external interrupts can be used on four of the eight External Interrupt sources.
- Port C is output only.
- USART has no FIFO buffer, so Data OverRun comes earlier.
- The user must have set unused I/O bits to 0 in ATmega103 programs.

Pin Descriptions

VCC

Digital supply voltage.

GND

Ground.

Port A (PA7..PA0)

Port A is an 8-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The Port A output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port A pins that are externally pulled low will source current if the pull-up resistors are activated. The Port A pins are tri-stated when a reset condition becomes active, even if the clock is not running.

Port A also serves the functions of various special features of the ATmega64 as listed on page 72.

Port B (PB7..PB0)

Port B is an 8-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The Port B output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port B pins that are externally pulled low will source current if the pull-up resistors are activated. The Port B pins are tri-stated when a reset condition becomes active, even if the clock is not running.

Port B also serves the functions of various special features of the ATmega64 as listed on page 73.

Port C (PC7..PC0)

Port C is an 8-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The Port C output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port C pins that are externally pulled low will source current if the pull-up resistors are activated. The Port C pins are tri-stated when a reset condition becomes active, even if the clock is not running.

Port C also serves the functions of special features of the ATmega64 as listed on page 76. In ATmega103 compatibility mode, Port C is output only, and the port C pins are not tri-stated when a reset condition becomes active.

Port D (PD7..PD0)

Port D is an 8-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The Port D output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port D pins that are externally pulled low will source current if the pull-up resistors are activated. The Port D pins are tri-stated when a reset condition becomes active, even if the clock is not running.

Port D also serves the functions of various special features of the ATmega64 as listed on page 77.

Port E (PE7..PE0)

Port E is an 8-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The Port E output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port E pins that are externally pulled low will source current if the pull-up resistors are activated. The Port E pins are tri-stated when a reset condition becomes active, even if the clock is not running.

Port E also serves the functions of various special features of the ATmega64 as listed on page 80.

Port F (PF7..PF0)

Port F serves as the analog inputs to the A/D Converter.

Port F also serves as an 8-bit bi-directional I/O port, if the A/D Converter is not used. Port pins can provide internal pull-up resistors (selected for each bit). The Port F output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port F pins that are externally pulled low will source current if the pull-up resistors are activated. The Port F pins are tri-stated when a reset condition becomes active, even if the clock is not running. If the JTAG interface is enabled, the pull-up resistors on pins PF7(TDI), PF5(TMS) and PF4(TCK) will be activated even if a reset occurs.

The TDO pin is tri-stated unless TAP states that shift out data are entered.

Port F also serves the functions of the JTAG interface.

In ATmega103 compatibility mode, Port F is an input port only.

Port G (PG4..PG0)

Port G is a 5-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The Port G output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port G pins that are externally pulled low will source current if the pull-up resistors are activated. The Port G pins are tri-stated when a reset condition becomes active, even if the clock is not running.

Port G also serves the functions of various special features.

In ATmega103 compatibility mode, these pins only serves as strobes signals to the external memory as well as input to the 32 kHz Oscillator, and the pins are initialized to PG0 = 1, PG1 = 1, and PG2 = 0 asynchronously when a reset condition becomes active, even if the clock is not running. PG3 and PG4 are Oscillator pins.

RESET	Reset input. A low level on this pin for longer than the minimum pulse length will generate a reset, even if the clock is not running. The minimum pulse length is given in Table 19 on page 51. Shorter pulses are not guaranteed to generate a reset.
XTAL1	Input to the inverting Oscillator amplifier and input to the internal clock operating circuit.
XTAL2	Output from the inverting Oscillator amplifier.
AVCC	AVCC is the supply voltage pin for Port F and the A/D Converter. It should be externally connected to V_{CC}, even if the ADC is not used. If the ADC is used, it should be connected to V_{CC} through a low-pass filter.
AREF	AREF is the analog reference pin for the A/D Converter.
PEN	This is a programming enable pin for the SPI Serial Programming mode. By holding this pin low during a Power-on Reset, the device will enter the SPI Serial Programming mode. $\overline{\text{PEN}}$ has no function during normal operation.

ภาคผนวก ค

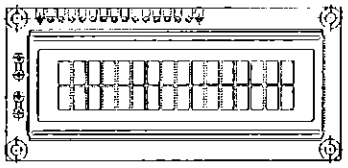
รายละเอียดของ LCD 16×2

VISHAY

LCD-016M002B

Vishay

16 x 2 Character LCD



FEATURES

- 5 x 8 dots with cursor
- Built-in controller (KS 0066 or Equivalent)
- + 5V power supply (Also available for + 3V)
- 1/16 duty cycle
- B/L to be driven by pin 1, pin 2 or pin 15, pin 16 or A.K (LED)
- N.V. optional for + 3V power supply

MECHANICAL DATA

ITEM	STANDARD VALUE	UNIT
Module Dimension	80.0 x 36.0	mm
Viewing Area	66.0 x 16.0	mm
Dot Size	0.56 x 0.66	mm
Character Size	2.96 x 5.56	mm

ABSOLUTE MAXIMUM RATING

ITEM	SYMBOL	STANDARD VALUE			UNIT
		MIN.	TYP.	MAX.	
Power Supply	VDD-VSS	- 0.3	–	7.0	V
Input Voltage	VI	- 0.3	–	VDD	V

NOTE: VSS = 0 Volt, VDD = 5.0 Volt

ELECTRICAL SPECIFICATIONS

ITEM	SYMBOL	CONDITION		STANDARD VALUE			UNIT
				MIN.	TYP.	MAX.	
Input Voltage	VDD	VDD = + 5V		4.7	5.0	5.3	V
		VDD = + 3V		2.7	3.0	5.3	V
Supply Current	IDD	VDD = 5V		–	1.2	3.0	mA
Recommended LC Driving Voltage for Normal Temp Version Module	VDD - V0	- 20 °C		–	–	–	V
		0°C		4.2	4.8	5.1	
		25°C		3.8	4.2	4.6	
		50°C		3.6	4.0	4.4	
		70°C		–	–	–	
LED Forward Voltage	VF	25°C		–	4.2	4.6	V
LED Forward Current	IF	25°C	Array	–	130	260	mA
			Edge	–	20	40	
EL Power Supply Current	IEL	Vel = 110VAC:400Hz		–	–	5.0	mA

DISPLAY CHARACTER ADDRESS CODE:

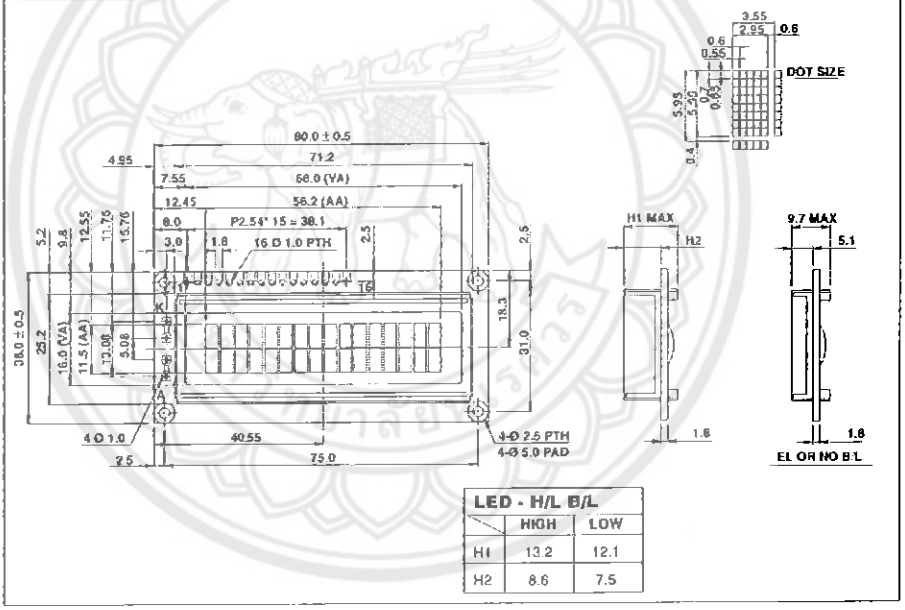
Display Position	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
DD RAM Address	00	01														0F
DD RAM Address	40	41														4F

LCD-016M002B

Vishay 16 x 2 Character LCD

PIN NUMBER	SYMBOL	FUNCTION
1	Vss	GND
2	Vdd	+ 3V or + 5V
3	Vo	Contrast Adjustment
4	RS	H/L Register Select Signal
5	R/$\overline{W}$	H/L Read/Write Signal
6	E	H →L Enable Signal
7	DB0	H/L Data Bus Line
8	DB1	H/L Data Bus Line
9	DB2	H/L Data Bus Line
10	DB3	H/L Data Bus Line
11	DB4	H/L Data Bus Line
12	DB5	H/L Data Bus Line
13	DB6	H/L Data Bus Line
14	DB7	H/L Data Bus Line
15	A/Vee	+ 4.2V for LED/Negative Voltage Output
16	K	Power Supply for B/L (0V)

DIMENSIONS in millimeters



LED - H/L B/L		
	HIGH	LOW
H1	13.2	12.1
H2	8.6	7.5

ภาคผนวก ง

รายละเอียดของ TLP250N, IRFP2907, LM2575, POWER RELAY

TOSHIBA Photocoupler GaAlAs Ired & Photo-IC

TLP250

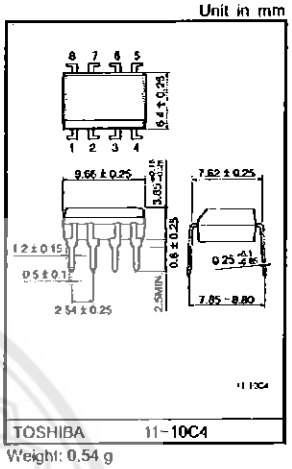
Transistor Inverter
Inverter For Air Conditionor
IGBT Gate Drive
Power MOS FET Gate Drive

The TOSHIBA TLP250 consists of a GaAlAs light emitting diode and a integrated photodetector.
This unit is 8-lead DIP package.
TLP250 is suitable for gate driving circuit of IGBT or power MOS FET.

- Input threshold current: I_F=5mA(max.)
- Supply current (I_{CC}): 11mA(max.)
- Supply voltage (V_{CC}): 10–35V
- Output current (I_O): ±1.5A (max.)
- Switching time (t_{pLH}/t_{pHL}): 1.5μs(max.)
- Isolation voltage: 2500V_{rms}(min.)
- UL recognized: UL1577, file No.E67349
- Option (D4) type
  VDE approved: DIN VDE0884/06.92,certificate No.76823
  Maximum operating insulation voltage: 630V_{PK}
  Highest permissible over voltage: 4000V_{PK}

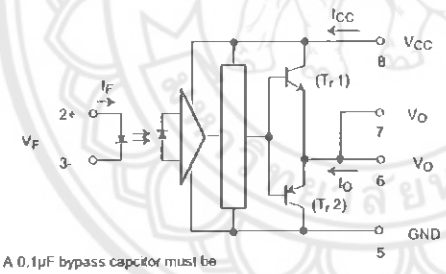
  (Note) When a VDE0884 approved type is needed, please designate the "option (D4)"
- Creepage distance: 6.4mm(min.)
  Clearance: 6.4mm(min.)

Unit in mm



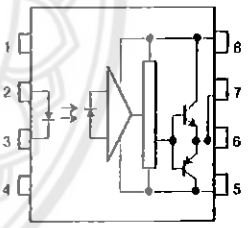
Weight: 0.54 g

Schmatic



A 0.1μF bypass capcitor must be connected between pin 8 and 5 (See Note 5)

Pin Configuration (top view)



1 : N.C.
2 : Anode
3 : Cathode
4 : N.C.
5 : GND
6 : V_O (Output)
7 : V_O
8 : V_{CC}

Truth Table

		Tr1	Tr2
Input LED	On	On	Off
	Off	Off	On

Absolute Maximum Ratings (Ta = 25°C)

	Characteristic			Symbol	Rating	Unit
LED	Forward current			I_F	20	mA
	Forward current derating (Ta ≥ 70°C)			ΔI_F / ΔTa	−0.36	mA / °C
	Peak transient forward current		(Note 1)	I_{FPT}	1	A
	Reverse voltage			V_R	5	V
	Junction temperature			Tj	125	°C
Detector	"H" peak output current (P_W ≤ 2.5μs, f ≤ 15kHz)		(Note 2)	I_{OPH}	−1.5	A
	"L" peak output current (P_W ≤ 2.5μs, f ≤ 15kHz)		(Note 2)	I_{OPL}	+1.5	A
	Output voltage	(Ta ≤ 70°C)		V_O	35	V
		(Ta = 85°C)			24	
	Supply voltage	(Ta ≤ 70°C)		V_{CC}	35	V
		(Ta = 85°C)			24	
	Output voltage derating (Ta ≥ 70°C)			ΔV_O / ΔTa	−0.73	V / °C
	Supply voltage derating (Ta ≥ 70°C)			ΔV_{CC} / ΔTa	−0.73	V / °C
	Junction temperature			Tj	125	°C
Operating frequency			(Note 3)	f	25	kHz
Operating temperature range				T_{opr}	−20~85	°C
Storage temperature range				T_{stg}	−55~125	°C
Lead soldering temperature (10 s)				T_{sol}	260	°C
Isolation voltage (AC, 1 min., R.H.≤ 60%)			(Note 4)	BVS	2500	Vrms

(Note 1) Pulse width P_W ≤ 1μs, 300pps

(Note 2) Exporenential wavefom

(Note 3) Exporenential wavefom, I_{OPH} ≤ −1.0A(≤ 2.5μs), I_{OPL} ≤ +1.0A(≤ 2.5μs)

(Note 4) Device considerd a two terminal device: Pins 1, 2, 3 and 4 shorted together, and pins 5, 6, 7 and 8 shorted together.

(Note 5) A ceramic capacitor(0.1μF) should be connected from pin 8 to pin 5 to stabilize the operation of the high gain linear amplifier. Failure to provide the bypassing may impair the switching property. The total lead length between capacitor and coupler should not exceed 1cm.

Recommended Operating Conditions

Characteristic	Symbol	Min.	Typ.	Max.		Unit
Input current, on	$I_{F(ON)}$	7	8	10		mA
Input voltage, off	$V_{F(OFF)}$	0	—	0.8		V
Supply voltage	V_{CC}	15	—	30	20	V
Peak output current	I_{OPH}/I_{OPL}	—	—	±0.5		A
Operating temperature	T_{opr}	−20	25	70	65	°C

Electrical Characteristics (Ta = −20~70°C, unless otherwise specified)

Characteristic		Symbol	Test Circuit	Test Condition		Min.	Typ.*	Max.	Unit
Input forward voltage		V_F	—	I_F = 10 mA, Ta = 25°C			1.6	1.8	V
Temperature coefficient of forward voltage		$\Delta V_F / \Delta Ta$	—	I_F = 10 mA		—	−2.0	—	mV / °C
Input reverse current		I_R	—	V_R = 5V, Ta = 25°C			—	10	μA
Input capacitance		C_T	—	V = 0, f = 1MHz, Ta = 25°C		—	45	250	pF
Output current	"H" level	I_{OPH}	3	V_{CC} = 30V (*1)	I_F = 10 mA, V_{8-6} = 4V	−0.5	−1.5	—	A
	"L" level	I_{OPL}	2		I_F = 0, V_{6-5} = 2.5V	0.5	2	—	
Output voltage	"H" level	V_{OH}	4	V_{CC1} = +15V, V_{EE1} = −15V, R_L = 200Ω, I_F = 5mA		11	12.8	—	V
	"L" level	V_{OL}	5	V_{CC1} = +15V, V_{EE1} = −15V, R_L = 200Ω, V_F = 0.8V		—	−14.2	−12.5	
Supply current	"H" level	I_{CCH}	—	V_{CC} = 30V, I_F = 10mA, Ta = 25°C		—	7	—	mA
				V_{CC} = 30V, I_F = 10mA		—	—	11	
	"L" level	I_{CCL}	—	V_{CC} = 30V, I_F = 0mA, Ta = 25°C		—	7.5	—	
				V_{CC} = 30V, I_F = 0mA		—	—	11	
Threshold input current	"Output L→H"	I_{FLH}	—	V_{CC1} = +15V, V_{EE1} = −15V, R_L = 200Ω, V_O > 0V		—	1.2	5	mA
Threshold input voltage	"Output H→L"	I_{FHL}	—	V_{CC1} = +15V, V_{EE1} = −15V, R_L = 200Ω, V_O < 0V		0.8	—	—	V
Supply voltage		V_{CC}	—			10	—	35	V
Capacitance (input−output)		C_S	—	V_S = 0, f = 1MHz, Ta = 25°C		—	1.0	2.0	pF
Resistance(input−output)		R_S	—	V_S = 500V, Ta = 25°C, R.H.≤ 60%		1×10^{12}	10^{14}	—	Ω

* All typical values are at Ta = 25°C (*1): Duration of I_O time ≤ 50μs

Switching Characteristics (Ta = –20~70°C , unless otherwise specified)

Characteristic		Symbol	Test Circuit	Test Condition	Min.	Typ.*	Max.	Unit
Propagation delay time	L→H	t_{pLH}	6	I_F = 8mA V_{CC1} = +15V, V_{EE1} = –15V R_L = 200Ω	—	0.15	0.5	μs
	H→L	t_{pHL}			—	0.15	0.5	
Output rise time		t_r			—	—	—	
Output fall time		t_f			—	—	—	
Common mode transient immunity at high level output		C_{MH}	7	V_{CM} = 600V, I_F = 8mA V_{CC} = 30V, Ta = 25°C	–5000	—	—	V / μs
Common mode transient immunity at low level output		C_{ML}	7	V_{CM} = 600V, I_F = 0mA V_{CC} = 30V, Ta = 25°C	5000	—	—	V / μs

* All typical values are at Ta = 25°C

International
IOR Rectifier

AUTOMOTIVE MOSFET

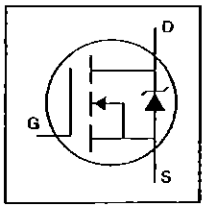
PD-93906D

IRFP2907

HEXFET® Power MOSFET

Typical Applications
- Integrated Starter Alternator
- 42 Volts Automotive Electrical Systems

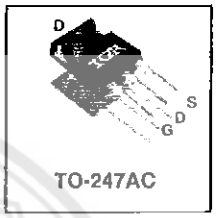
Benefits
- Advanced Process Technology
- Ultra Low On-Resistance
- Dynamic dv/dt Rating
- 175°C Operating Temperature
- Fast Switching
- Repetitive Avalanche Allowed up to Tjmax



V_{DSS} = 75V

$R_{DS(on)}$ = 4.5mΩ

I_D = 209A⑥



TO-247AC

Description
Specifically designed for Automotive applications, this Stripe Planar design of HEXFET® Power MOSFETs utilizes the lastest processing techniques to achieve extremely low on-resistance per silicon area. Additional features of this HEXFET power MOSFET are a 175°C junction operating temperature, fast switching speed and improved repetitive avalanche rating. These benefits combine to make this design an extremely efficient and reliable device for use in Automotive applications and a wide variety of other applications.

G	D	S
Gate	Drain	Source

Absolute Maximum Ratings

	Parameter	Max.	Units
I_D @ T_C = 25°C	Continuous Drain Current, V_{GS} @ 10V	209⑥	A
I_D @ T_C = 100°C	Continuous Drain Current, V_{GS} @ 10V	148⑥	
I_{DM}	Pulsed Drain Current ①	840	
P_D @ T_C = 25°C	Power Dissipation	470	W
	Linear Derating Factor	3.1	W/°C
V_{GS}	Gate-to-Source Voltage	± 20	V
E_{AS}	Single Pulse Avalanche Energy②	1970	mJ
I_{AR}	Avalanche Current	See Fig.12a, 12b, 15, 16	A
E_{AR}	Repetitive Avalanche Energy②		mJ
dv/dt	Peak Diode Recovery dv/dt ③	5.0	V/ns
T_J	Operating Junction and	-55 to + 175	°C
T_{STG}	Storage Temperature Range		
	Soldering Temperature, for 10 seconds	300 (1.6mm from case)	
	Mounting Torque, 6-32 or M3 screw	10 lbf•in (1.1N•m)	

Thermal Resistance

	Parameter	Typ.	Max.	Units
$R_{\theta JC}$	Junction-to-Case	—	0.32	°C/W
$R_{\theta CS}$	Case-to-Sink, Flat, Greased Surface	0.24	—	
$R_{\theta JA}$	Junction-to-Ambient	—	40	

Electrical Characteristics @ T_J = 25°C (unless otherwise specified)

	Parameter	Min.	Typ.	Max.	Units	Conditions
$V_{(BR)DSS}$	Drain-to-Source Breakdown Voltage	75	—	—	V	V_{GS} = 0V, I_D = 250µA
$\Delta V_{(BR)DSS}/\Delta T_J$	Breakdown Voltage Temp. Coefficient	—	0.085	—	V/°C	Reference to 25°C, I_D = 1mA
$R_{DS(on)}$	Static Drain-to-Source On-Resistance	—	3.6	4.5	mΩ	V_{GS} = 10V, I_D = 125A ④
$V_{GS(th)}$	Gate Threshold Voltage	2.0	—	4.0	V	V_{DS} = 10V, I_D = 250µA
g_{fs}	Forward Transconductance	130	—	—	S	V_{DS} = 25V, I_D = 125A
I_{DSS}	Drain-to-Source Leakage Current	—	—	20	µA	V_{DS} = 75V, V_{GS} = 0V
		—	—	250		V_{DS} = 60V, V_{GS} = 0V, T_J = 150°C
I_{GSS}	Gate-to-Source Forward Leakage	—	—	200	nA	V_{GS} = 20V
	Gate-to-Source Reverse Leakage	—	—	-200		V_{GS} = -20V
Q_g	Total Gate Charge	—	410	620	nC	I_D = 125A
Q_{gs}	Gate-to-Source Charge	—	92	140		V_{DS} = 60V
Q_{gd}	Gate-to-Drain ("Miller") Charge	—	140	210		V_{GS} = 10V④
$t_{d(on)}$	Turn-On Delay Time	—	23	—	ns	V_{DD} = 38V
t_r	Rise Time	—	190	—		I_D = 125A
$t_{d(off)}$	Turn-Off Delay Time	—	130	—		R_G = 1.2Ω
t_f	Fall Time	—	130	—		V_{GS} = 10V ④
L_D	Internal Drain Inductance	—	5.0	—	nH	Between lead, 6mm (0.25in.) from package and center of die contact
L_S	Internal Source Inductance	—	13	—		
C_{iss}	Input Capacitance	—	13000	—	pF	V_{GS} = 0V
C_{oss}	Output Capacitance	—	2100	—		V_{DS} = 25V
C_{rss}	Reverse Transfer Capacitance	—	500	—		f = 1.0MHz, See Fig. 6
C_{oss}	Output Capacitance	—	9780	—		V_{GS} = 0V, V_{DS} = 1.0V, f = 1.0MHz
C_{oss}	Output Capacitance	—	1360	—		V_{GS} = 0V, V_{DS} = 60V, f = 1.0MHz
C_{oss} eff.	Effective Output Capacitance ⑤	—	2320	—		V_{GS} = 0V, V_{DS} = 0V to 60V

Source-Drain Ratings and Characteristics

	Parameter	Min.	Typ.	Max.	Units	Conditions
I_S	Continuous Source Current (Body Diode)	—	—	209②	A	MOSFET symbol showing the integral reverse p-n junction diode.
I_{SM}	Pulsed Source Current (Body Diode) ①	—	—	840		
V_{SD}	Diode Forward Voltage	—	—	1.3	V	T_J = 25°C, I_S = 125A, V_{GS} = 0V ④
t_{rr}	Reverse Recovery Time	—	140	210	ns	T_J = 25°C, I_F = 125A
Q_{rr}	Reverse RecoveryCharge	—	880	1320	nC	di/dt = 100A/µs ④
t_{on}	Forward Turn-On Time	Intrinsic turn-on time is negligible (turn-on is dominated by L_S+L_D)				

Notes:

① Repetitive rating; pulse width limited by max. junction temperature. (See fig. 11).

② Starting T_J = 25°C, L = 0.25mH R_G = 25Ω, I_{AS} = 125A. (See Figure 12).

③ I_{SD} ≤ 125A, di/dt ≤ 260A/µs, V_{DD} ≤ $V_{(BR)DSS}$, T_J ≤ 175°C

④ Pulse width ≤ 400µs; duty cycle ≤ 2%.

⑤ C_{oss} eff. is a fixed capacitance that gives the same charging time as C_{oss} while V_{DS} is rising from 0 to 80% V_{DSS}.

⑥ Calculated continuous current based on maximum allowable junction temperature. Package limitation current is 90A.

⑦ Limited by T_{Jmax}, see Fig.12a, 12b, 15, 16 for typical repetitive avalanche performance.

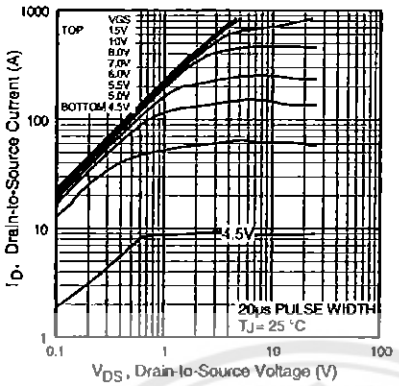


Fig 1. Typical Output Characteristics

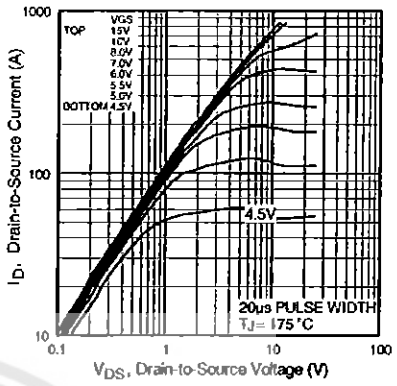


Fig 2. Typical Output Characteristics

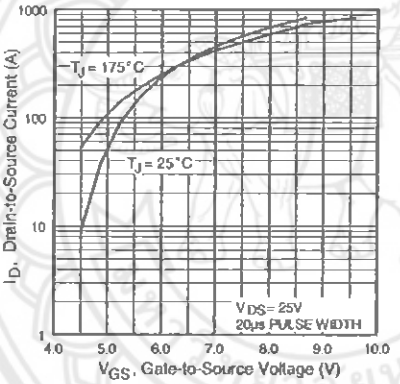


Fig 3. Typical Transfer Characteristics

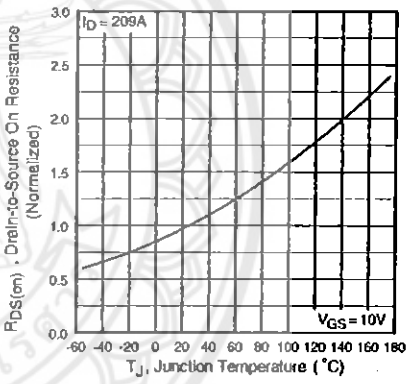


Fig 4. Normalized On-Resistance Vs. Temperature

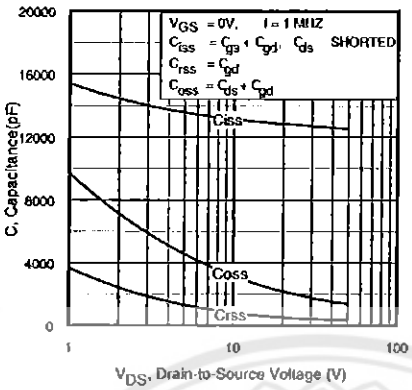


Fig 5. Typical Capacitance Vs. Drain-to-Source Voltage

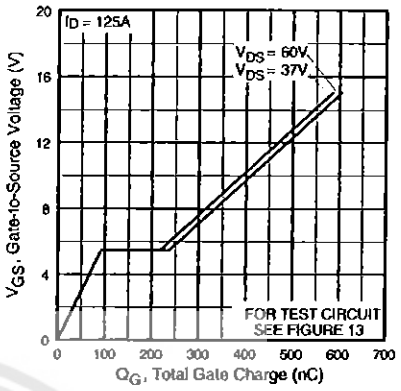


Fig 6. Typical Gate Charge Vs. Gate-to-Source Voltage

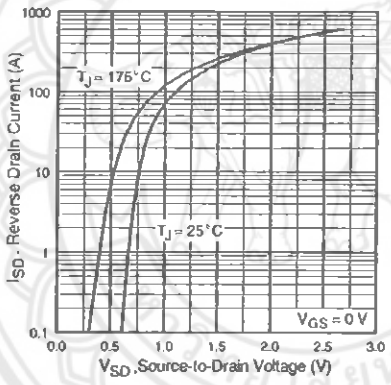


Fig 7. Typical Source-Drain Diode Forward Voltage

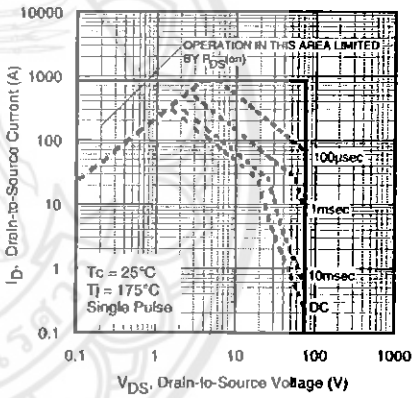


Fig 8. Maximum Safe Operating Area

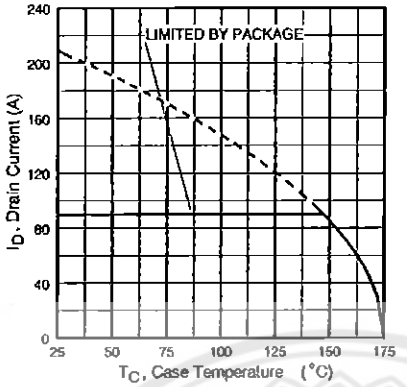


Fig 9. Maximum Drain Current Vs. Case Temperature

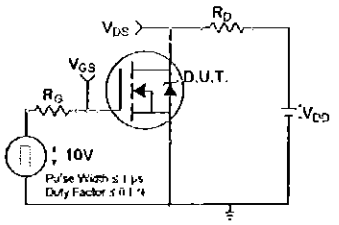


Fig 10a. Switching Time Test Circuit

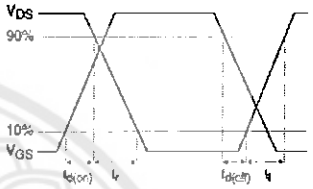


Fig 10b. Switching Time Waveforms

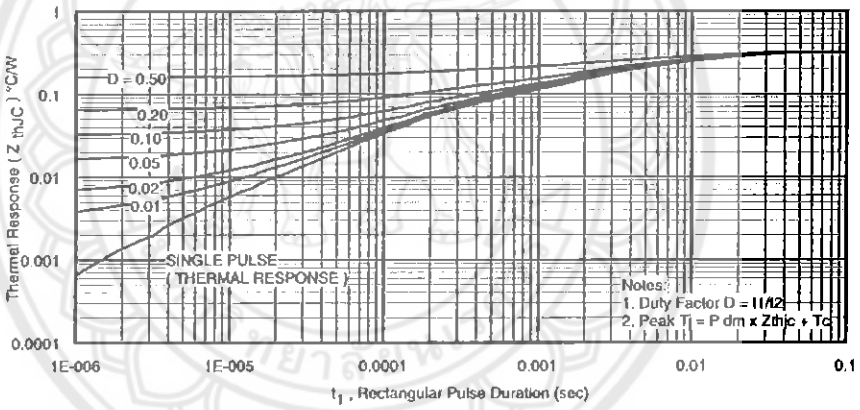


Fig 11. Maximum Effective Transient Thermal Impedance, Junction-to-Case

April 2007

National Semiconductor

LM1575/LM2575/LM2575HV
SIMPLE SWITCHER® 1A Step-Down Voltage Regulator

General Description

The LM2575 series of regulators are monolithic integrated circuits that provide all the active functions for a step-down (buck) switching regulator, capable of driving a 1A load with excellent line and load regulation. These devices are available in fixed output voltages of 3.3V, 5V, 12V, 15V, and an adjustable output version.

Requiring a minimum number of external components, these regulators are simple to use and include internal frequency compensation and a fixed-frequency oscillator.

The LM2575 series offers a high-efficiency replacement for popular three-terminal linear regulators. It substantially reduces the size of the heat sink, and in many cases no heat sink is required.

A standard series of inductors optimized for use with the LM2575 are available from several different manufacturers. This feature greatly simplifies the design of switch-mode power supplies.

Other features include a guaranteed ±4% tolerance on output voltage within specified input voltages and output load conditions, and ±10% on the oscillator frequency. External shutdown is included, featuring 50 µA (typical) standby current. The output switch includes cycle-by-cycle current limiting, as well as thermal shutdown for full protection under fault conditions.

Features

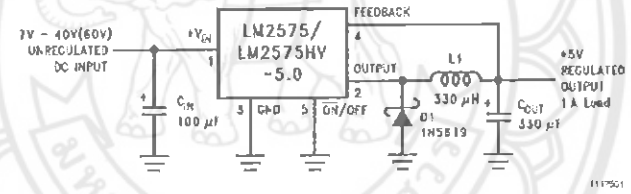
- 3.3V, 5V, 12V, 15V, and adjustable output versions
- Adjustable version output voltage range, 1.23V to 37V (57V for HV version) ±4% max over line and load conditions
- Guaranteed 1A output current
- Wide input voltage range, 40V up to 60V for HV version
- Requires only 4 external components
- 52 kHz fixed frequency internal oscillator
- TTL shutdown capability, low power standby mode
- High efficiency
- Uses readily available standard inductors
- Thermal shutdown and current limit protection
- P+ Product Enhancement tested

Applications

- Simple high-efficiency step-down (buck) regulator
- Efficient pre-regulator for linear regulators
- On-card switching regulators
- Positive to negative converter (Buck-Boost)

Typical Application

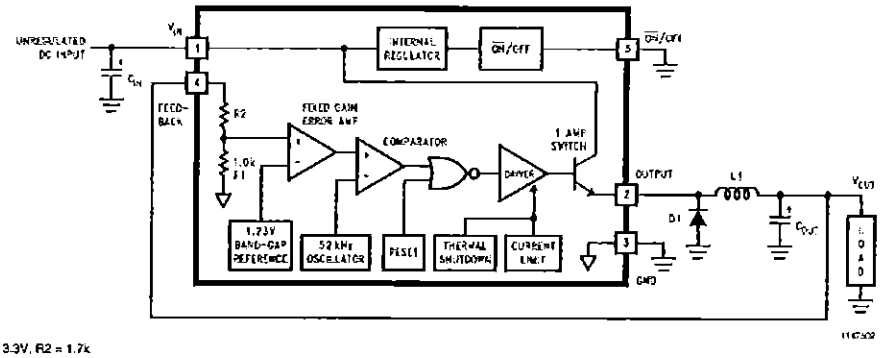
(Fixed Output Voltage Versions)



Note: Pin numbers are for the TO-220 package.

SIMPLE SWITCHER® is a registered trademark of National Semiconductor Corporation

Block Diagram and Typical Application



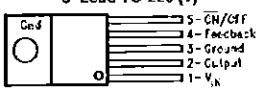
3.3V, R2 = 1.7k
5V, R2 = 3.1k
12V, R2 = 8.84k
15V, R2 = 11.3k
For ADJ. Version
R1 = Open, R2 = 0Ω
Note: Pin numbers are for the TO-220 package.

FIGURE 1.

Connection Diagrams

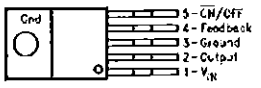
(XX indicates output voltage option. See Ordering Information table for complete part number.)

Straight Leads
5-Lead TO-220 (T)

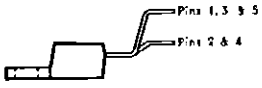


Top View
LM2575T-XX or LM2575HVT-XX
See NS Package Number T05A

Bent, Staggered Leads
5-Lead TO-220 (T)

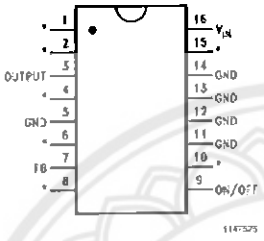


Top View



Side View
LM2575T-XX Flow LB03 or
LM2575HVT-XX Flow LB03
See NS Package Number T05D

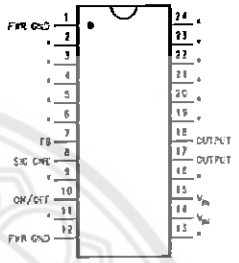
16-Lead DIP (N or J)



*No Internal Connection

Top View
LM2575N-XX or LM2575HVN-XX
See NS Package Number N16A
LM1575J-XX-QML
See NS Package Number J16A

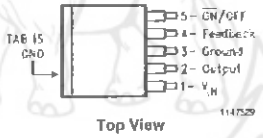
24-Lead Surface Mount (M)



*No Internal Connection

Top View
LM2575M-XX or LM2575HVM-XX
See NS Package Number M24B

TO-263(S)
5-Lead Surface-Mount Package



Top View

Side View
LM2575S-XX or LM2575HVS-XX
See NS Package Number TS5B

FUJITSU THE POSSIBILITIES ARE INFINITE

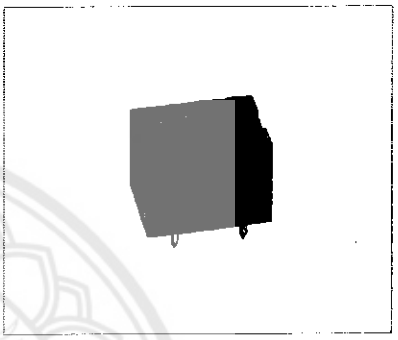
POWER RELAY

1 POLE—3, 5, 10 A

FBR160 SERIES

■ FEATURES

- Compact with high power (3 A to 10 A)
- 6 types of contact materials available for home electronics and automotive applications
- Design conforms to the following safety standards
  - UL114 No. E63615
  - UL508 No. E63614
  - CSA No. LR64026
  - Japan Electric Appliance Control Law (150–300 V)
- For automatic assembly
  Tube packaging suitable for automatic insertion equipment is available



■ ORDERING INFORMATION

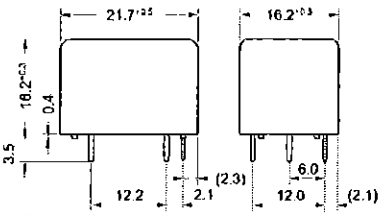
[Example] FBR16 (a) 1 (b) S (c) E (d) D (e) 012 (f) UH (g) –CSA (h) –••• (i) –S (j)

(a)	Series Name	FBR16: FBR160 Series		
(b)	Contact Arrangement	1 : 1 form C (SPDT) 3 : 1 form A (SPST-NO)		
(c)	Enclosure	S : Flux free N : Plastic sealed		
(d)	Coil Rating	E : Nominal power 0.36 W type C : Nominal power 0.5 W type (refer to the SPECIFICATIONS)		
(e)	Coil	D : DC Coil		
(f)	Nominal Voltage	(Example) 012: 12 VDC coil 024: 24 VDC coil (refer to the COIL DATA CHART)		
(g)	UL Standard and Contact Material	UL 114 recognized	UL508 recognized	Material / Rating
		U	R	Silver (3A)
		UK	RK	Silver-cadmium oxide (3 A)
		UH	RH	Silver-cadmium oxide (5 A)
		UW	RW	Silver tin oxide alloy (5 A)
		UHB	RHB	Silver-cadmium oxide (AC 10 A)
		UWB	RWB	Silver tin oxide alloy (DC 10 A

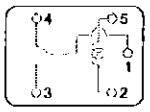
FBR160 SERIES

■ DIMENSIONS

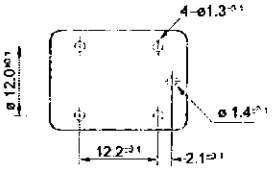
● Dimensions



● Schematic
(BOTTOM VIEW)

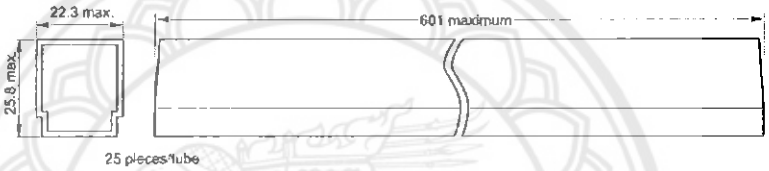


● PC board mounting hole layout
(BOTTOM VIEW)



Note : For 1 form A type, terminal No.4 is removed.

● Tube carrier



25 pieces/tube

Unit: mm

ภาคผนวก จ

รายระเอียดของ เซ็นเซอร์ DS18S20

DALLAS SEMICONDUCTOR MAXIM

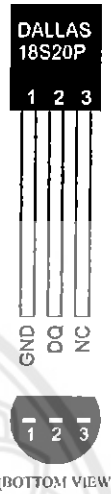
DS18S20-PAR
1-Wire Parasite-Power Digital Thermometer

www.maxim-ic.com

FEATURES

- Unique 1-Wire® interface requires only one port pin for communication
- Derives power from data line ("parasite power")—does not need a local power supply
- Multi-drop capability simplifies distributed temperature sensing applications
- Requires no external components
- ±0.5°C accuracy from –10°C to +85°C
- Measures temperatures from –55°C to +100°C (–67°F to +212°F)
- 9-bit thermometer resolution
- Converts temperature in 750 ms (max.)
- User–definable non-volatile temperature alarm settings
- Alarm search command identifies and addresses devices whose temperature is outside of programmed limits (temperature alarm condition)
- Ideal for use in remote sensing applications (e.g., temperature probes) that do not have a local power source

PIN ASSIGNMENT



(BOTTOM VIEW)

TO-92
(DS18S20-PAR)

PIN DESCRIPTION

GND - Ground
DQ - Data In/Out
NC - No Connect

DESCRIPTION

The DS18S20-PAR digital thermometer provides 9–bit centigrade temperature measurements and has an alarm function with nonvolatile user-programmable upper and lower trigger points. The DS18S20-PAR does not need an external power supply because it derives power directly from the data line ("parasite power"). The DS18S20-PAR communicates over a 1-Wire bus, which by definition requires only one data line (and ground) for communication with a central microprocessor. It has an operating temperature range of –55°C to +100°C and is accurate to ±0.5°C over a range of –10°C to +85°C.

Each DS18S20-PAR has a unique 64-bit identification code, which allows multiple DS18S20-PARs to function on the same 1–wire bus; thus, it is simple to use one microprocessor to control many DS18S20-PARs distributed over a large area. Applications that can benefit from this feature include HVAC environmental controls, temperature monitoring systems inside buildings, equipment or machinery, and process monitoring and control systems.

DETAILED PIN DESCRIPTIONS Table 1

PIN	SYMBOL	DESCRIPTION
1	GND	**Ground.**
2	DQ	**Data Input/Output pin.** Open-drain 1-Wire interface pin. Also provides power to the device when used in parasite power mode (see "Parasite Power" section.)
3	NC	**No Connect.** Doesn't connect to internal circuit.

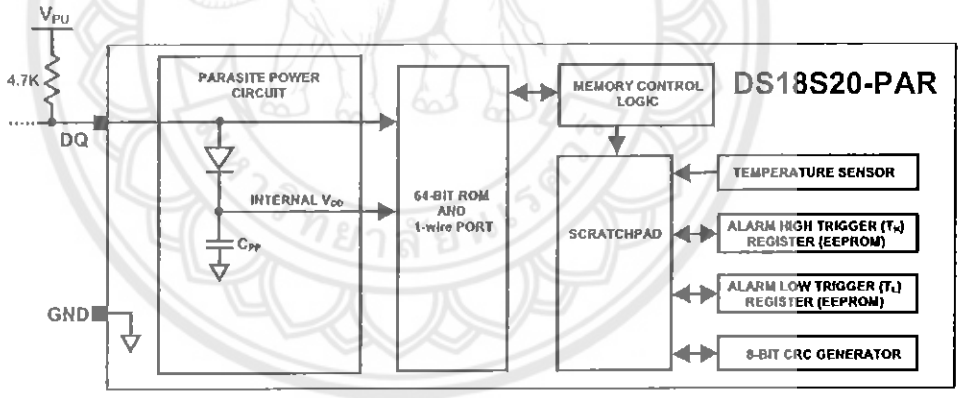
OVERVIEW

The DS18S20-PAR uses Dallas' exclusive 1-Wire bus protocol that implements bus communication using one control signal. The control line requires a weak pullup resistor since all devices are linked to the bus via a 3-state or open-drain port (the DQ pin in the case of the DS18S20-PAR). In this bus system, the microprocessor (the master device) identifies and addresses devices on the bus using each device's unique 64-bit code. Because each device has a unique code, the number of devices that can be addressed on one bus is virtually unlimited. The 1-Wire bus protocol, including detailed explanations of the commands and "time slots," is covered in the 1-WIRE BUS SYSTEM section of this datasheet.

An important feature of the DS18S20-PAR is its ability to operate without an external power supply. Power is instead supplied through the 1-Wire pullup resistor via the DQ pin when the bus is high. The high bus signal also charges an internal capacitor (C_{PP}), which then supplies power to the device when the bus is low. This method of deriving power from the 1-Wire bus is referred to as "parasite power."

Figure 1 shows a block diagram of the DS18S20-PAR, and pin descriptions are given in Table 1. The 64-bit ROM stores the device's unique serial code. The scratchpad memory contains the 2-byte temperature register that stores the digital output from the temperature sensor. In addition, the scratchpad provides access to the 1-byte upper and lower alarm trigger registers (T_H and T_L). The T_H and T_L registers are nonvolatile (EEPROM), so they will retain their data when the device is powered down.

DS18S20-PAR BLOCK DIAGRAM Figure 1



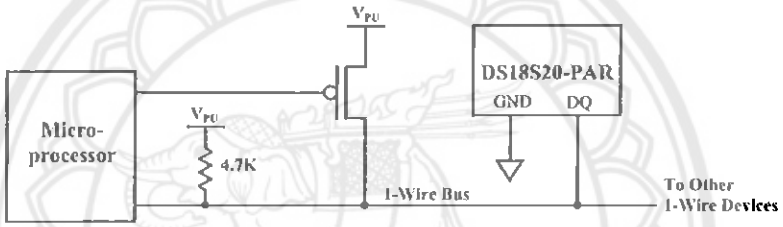
PARASITE POWER

The DS18S20-PAR's parasite power circuit allows the DS18S20-PAR to operate without a local external power supply. This ability is especially useful for applications that require remote temperature sensing or that are very space constrained. Figure 1 shows the DS18S20-PAR's parasite-power control circuitry, which "steals" power from the 1-Wire bus via the DQ pin when the bus is high. The stolen charge powers the DS18S20-PAR while the bus is high, and some of the charge is stored on the parasite power capacitor (C_{PP}) to provide power when the bus is low.

The 1-Wire bus and C_{PP} can provide sufficient parasite power to the DS18S20-PAR for most operations as long as the specified timing and voltage requirements are met (refer to the DC ELECTRICAL CHARACTERISTICS and the AC ELECTRICAL CHARACTERISTICS sections of this data sheet). However, when the DS18S20-PAR is performing temperature conversions or copying data from the scratchpad memory to EEPROM, the operating current can be as high as 1.5 mA. This current can cause an unacceptable voltage drop across the weak 1-Wire pullup resistor and is more current than can be supplied by C_{PP}. To assure that the DS18S20-PAR has sufficient supply current, it is necessary to provide a strong pullup on the 1-Wire bus whenever temperature conversions are taking place or data is being copied from the scratchpad to EEPROM. This can be accomplished by using a MOSFET to pull the bus directly to the rail as shown in Figure 2. The 1-Wire bus must be switched to the strong pullup within 10 µs (max) after a Convert T [44h] or Copy Scratchpad [48h] command is issued, and the bus must be held high by the pullup for the duration of the conversion (t_{conv}) or data transfer (t_{WT} = 10 ms). No other activity can take place on the 1-Wire bus while the pullup is enabled.

SUPPLYING THE DS18S20-PAR DURING TEMPERATURE CONVERSIONS

Figure 2



OPERATION – MEASURING TEMPERATURE

The core functionality of the DS18S20-PAR is its direct-to-digital temperature sensor. The temperature sensor output has 9-bit resolution, which corresponds to 0.5°C steps. The DS18S20-PAR powers-up in a low-power idle state; to initiate a temperature measurement and A-to-D conversion, the master must issue a Convert T [44h] command. Following the conversion, the resulting thermal data is stored in the 2-byte temperature register in the scratchpad memory and the DS18S20-PAR returns to its idle state. The DS18S20-PAR output data is calibrated in degrees centigrade; for Fahrenheit applications, a lookup table or conversion routine must be used. The temperature data is stored as a 16-bit sign-extended two's complement number in the temperature register (see Figure 3). The sign bits (S) indicate if the temperature is positive or negative: for positive numbers S = 0 and for negative numbers S = 1. Table 2 gives examples of digital output data and the corresponding temperature reading.

Resolutions greater than 9 bits can be calculated using the data from the temperature, COUNT REMAIN and COUNT PER °C registers in the scratchpad. Note that the COUNT PER °C register is hard-wired to

16 (10h). After reading the scratchpad, the TEMP_READ value is obtained by truncating the 0.5°C bit (bit 0) from the temperature data (see Figure 3). The extended resolution temperature can then be calculated using the following equation:

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

TEMPERATURE REGISTER FORMAT Figure 3

	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
LS Byte	2^6	2^5	2^4	2^3	2^2	2^1	2^0	2^{-1}
	bit 15	bit 14	bit 13	bit 12	bit 11	bit 10	bit 9	bit 8
MS Byte	S	S	S	S	S	S	S	S

TEMPERATURE/DATA RELATIONSHIP Table 2

TEMPERATURE	DIGITAL OUTPUT (Binary)	DIGITAL OUTPUT (Hex)
+85.0°C*	0000 0000 1010 1010	00AAh
+25.0°C	0000 0000 0011 0010	0032h
+0.5°C	0000 0000 0000 0001	0001h
0°C	0000 0000 0000 0000	0000h
-0.5°C	1111 1111 1111 1111	FFFFh
-25.0°C	1111 1111 1100 1110	FFCEh
-55.0°C	1111 1111 1001 0010	FF92h

*The power-on reset value of the temperature register is +85°C

OPERATION – ALARM SIGNALING

After the DS18S20-PAR performs a temperature conversion, the temperature value is compared to the user-defined two's complement alarm trigger values stored in the 1-byte T_H and T_L registers (see Figure 4). The sign bit (S) indicates if the value is positive or negative: for positive numbers S = 0 and for negative numbers S = 1. The T_H and T_L registers are nonvolatile (EEPROM) so they will retain data when the device is powered down. T_H and T_L can be accessed through bytes 2 and 3 of the scratchpad as explained in the MEMORY section of this datasheet.

T_H AND T_L REGISTER FORMAT Figure 4

bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
S	2^6	2^5	2^4	2^3	2^2	2^1	2^0

Only bits 8 through 1 of the temperature register are used in the T_H and T_L comparison since T_H and T_L are 8-bit registers. If the result of a temperature measurement is higher than T_H or lower than T_L, an alarm condition exists and an alarm flag is set inside the DS18S20-PAR. This flag is updated after every temperature measurement; therefore, if the alarm condition goes away, the flag will be turned off after the next temperature conversion.

The master device can check the alarm flag status of all DS DS18S20-PARs on the bus by issuing an Alarm Search [ECh] command. Any DS18S20-PARs with a set alarm flag will respond to the command,

so the master can determine exactly which DS18S20-PARs have experienced an alarm condition. If an alarm condition exists and the T_H or T_L settings have changed, another temperature conversion should be done to validate the alarm condition.

64-BIT LASERED ROM CODE

Each DS18S20-PAR contains a unique 64–bit code (see Figure 5) stored in ROM. The least significant 8 bits of the ROM code contain the DS18S20-PAR's 1- wire family code: 10h. The next 48 bits contain a unique serial number. The most significant 8 bits contain a cyclic redundancy check (CRC) byte that is calculated from the first 56 bits of the ROM code. A detailed explanation of the CRC bits is provided in the CRC GENERATION section. The 64–bit ROM code and associated ROM function control logic allow the DS18S20-PAR to operate as a 1–wire device using the protocol detailed in the 1-WIRE BUS SYSTEM section of this datasheet.

64-BIT LASERED ROM CODE Figure 5

8-BIT CRC		48-BIT SERIAL NUMBER		8-BIT FAMILY CODE (10h)	
MSB	LSB	MSB	LSB	MSB	LSB

MEMORY

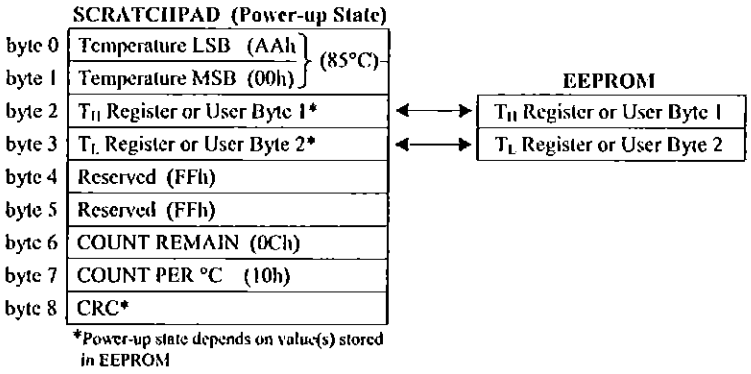
The DS18S20-PAR's memory is organized as shown in Figure 6. The memory consists of an SRAM scratchpad with nonvolatile EEPROM storage for the high and low alarm trigger registers (T_H and T_L). Note that if the DS18S20-PAR alarm function is not used, the T_H and T_L registers can serve as general-purpose memory. All memory commands are described in detail in the DS18S20-PAR FUNCTION COMMANDS section.

Byte 0 and byte 1 of the scratchpad contain the LSB and the MSB of the temperature register, respectively. These bytes are read-only. Bytes 2 and 3 provide access to T_H and T_L registers. Bytes 4 and 5 are reserved for internal use by the device and cannot be overwritten; these bytes will return all 1s when read. Bytes 6 and 7 contain the COUNT REMAIN and COUNT PER °C registers, which can be used to calculate extended resolution results as explained in the OPERATION – MEASURING TEMPERATURE section. Byte 8 of the scratchpad is read-only and contains the cyclic redundancy check (CRC) code for bytes 0 through 7 of the scratchpad. The DS18S20-PAR generates this CRC using the method described in the CRC GENERATION section.

Data is written to bytes 2 and 3 of the scratchpad using the Write Scratchpad [4Eh] command; the data must be transmitted to the DS18S20-PAR starting with the least significant bit of byte 2. To verify data integrity, the scratchpad can be read (using the Read Scratchpad [BEh] command) after the data is written. When reading the scratchpad, data is transferred over the 1-Wire bus starting with the least significant bit of byte 0. To transfer the T_H and T_L data from the scratchpad to EEPROM, the master must issue the Copy Scratchpad [48h] command.

Data in the EEPROM registers is retained when the device is powered down; at power-up the EEPROM data is reloaded into the corresponding scratchpad locations. Data can also be reloaded from EEPROM to the scratchpad at any time using the Recall E^2 [B8h] command. The master can issue "read time slots" (see the 1-WIRE BUS SYSTEM section) following the Recall E^2 command and the DS18S20-PAR will indicate the status of the recall by transmitting 0 while the recall is in progress and 1 when the recall is done.

DS18S20-PAR MEMORY MAP Figure 6



ภาคผนวก ฉ

รายระเอียดของ ไอซีเบอร์ MAX 232

+5V-Powered, Multichannel RS-232 Drivers/Receivers

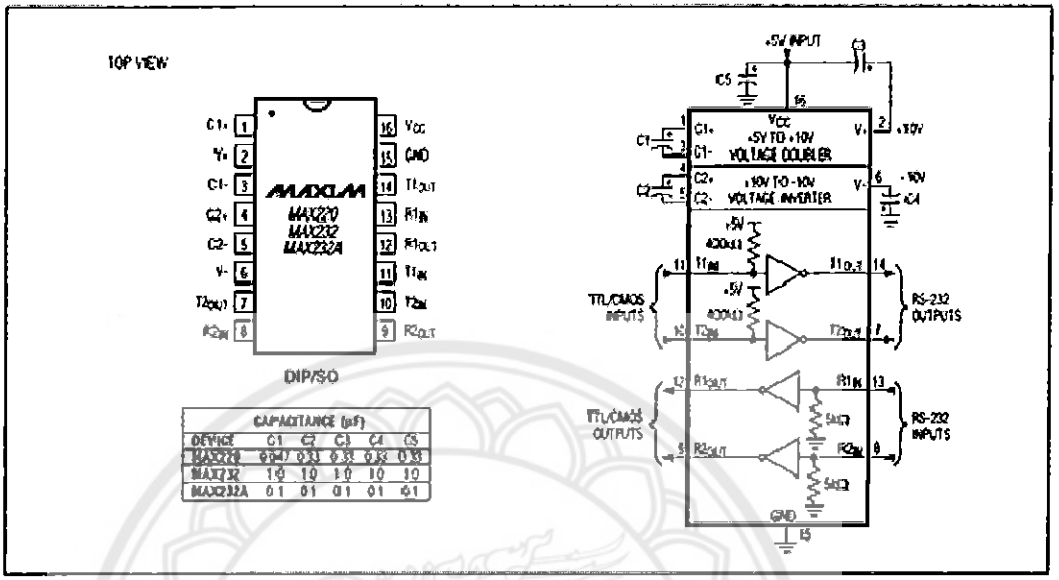


Figure 5. MAX220/MAX232/MAX232A Pin Configuration and Typical Operating Circuit

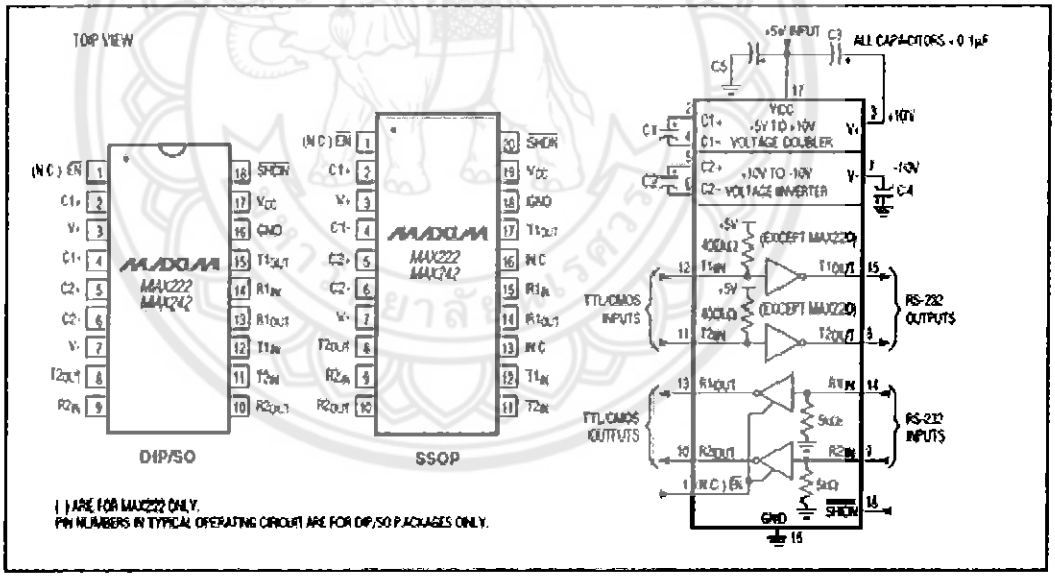


Figure 6. MAX222/MAX242 Pin Configurations and Typical Operating Circuit