สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

เตาอบควบคุมอุณหภูมิ

Oven Temperature Control

โดย

นาย รชต เตชะเวชเจริญ

นาย สุรวีร์ ลือลีลาโรจน์

นาย อติวุฒิ เพชรธนสาร

อาจารย์ที่ปรึกษา

อาจารย์ เฉลิมพันธ์ หวังวิวัฒนา

เลขหมู่.....
เลขทะเบียน 119449
วัน,เดือน,ปี - 7 S.A. 2554

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

รายงานนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตร์บัณฑิต

สาขาอิเล็กทรอนิกส์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ประจำภาคเรียนที่ 2 ปีการศึกษา 2553

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เตาอบควบคุมอุณหภูมิ

Oven Temperature Control

โดย

นาย รชต เตชะเวชเจริญ รหัส 50011263

นาย สุรวีร์ ลือลีลาโรจน์ รหัส 50011760

นาย อติวุฒิ เพชรธนสาร รหัส 50011823

อาจารย์ที่ปรึกษา

อาจารย์ เฉลิมพันธ์ หวังวิวัฒนา

รายงานนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตร์บัณฑิต

สาขาอิเล็กทรอนิกส์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ประจำภาคเรียนที่ 2 ปีการศึกษา 2553

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์ปีการศึกษา 2553

สาขาวิชาวิศวกรรม อิเล็กทรอนิกส์

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง เตาอบควบคุมอุณหภูมิ

OVEN TEMPARATURE CONTROL

ผู้จัดทำ

1. นาย รชต เตชะเวชเจริญ รหัส 50011263
2. นาย สุรวีร์ ลือลีลาโรจน์ รหัส 50011760
3. นาย อติวุฒิ เพชรธนสาร รหัส 50011823

.....อาจารย์ที่ปรึกษา

(อาจารย์เฉลิมพันธ์ หวังวิวัฒนา)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เตาอบควบคุมอุณหภูมิ

นาย รชต เตชะเวชเจริญ รหัส 50011263
นาย สุรวีร์ ลือลีลาโรจน์ รหัส 50011760
นาย อติวุฒิ เพชรธนสาร รหัส 50011823
อาจารย์เฉลิมพันธ์ หวังวิวัฒนา อาจารย์ที่ปรึกษา
ปีการศึกษา 2553

บทคัดย่อ

โครงงานนี้เป็นการออกแบบ และสร้างระบบควบคุมอุณหภูมิในเตาอบ โดยมีส่วนประกอบสำคัญได้แก่ ส่วนตรวจจับอุณหภูมิ ส่วนควบคุมอัตโนมัติ และส่วนแสดงผล โดยการใช้หลักการของPID Control ในการกำหนดค่า duty cycle ของสัญญาณ เมื่อทำการตรวจจับอุณหภูมิได้แล้ว จะส่งค่าไปยังไมโครคอนโทรลเลอร์เพื่อประมวลผลเปรียบเทียบอุณหภูมิที่วัดได้กับอุณหภูมิที่ตั้งไว้ หากยังไม่ได้อุณหภูมิตามต้องการ ระบบจะส่งสัญญาณโดยใช้หลักการของ Pulse Width Modulation ในการควบคุมการทำงานของฮีตเตอร์ จนกว่าจะถึงอุณหภูมิที่ต้องการซึ่งสามารถแสดงผลของอุณหภูมิได้ทางจอ LCD

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Oven Temperature Control

Mr. Ratchata	Techawetcharoen	ID 50011263
Mr. Suravee	Lurleelaroj	ID 50011760
Mr. Atiwut	Phetthanasarn	ID 50011823
Mr. Charoemphun	Wungwiwattana	Adviser

Academic Year 2010

Abstract

This project is designed and used microcontroller technology for control the temperature in Oven. There are three parts of the important in this project. It consists of temperature control machine, auto controls and display part .It can be set the time ,temperature and can also keep the temperature in the computer. Using Triac and pulse width to control the operation of Heater. The automatic control part composes the temperature detector which get the value of the temperature and send to microcontroller for evaluate the result. Then microcontroller will compare the measured value and set point value by PID Control by adjusting the constants to fit the system that we designed. If it isn't the desired temperature. System will send a signal to control the heater and can be display with LCD display for indication set point value and measured value.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

การทำโครงงานเรื่อง เตาอบควบคุมอุณหภูมิ ได้เรียนรู้ขั้นตอนการทำงาน การทำงานของเซนเซอร์วัดอุณหภูมิ และไมโครคอนโทรเลอร์ โดยได้อาจารย์ เฉลิมพันธุ์ หวังวิวัฒนา ซึ่งเป็นอาจารย์ที่ปรึกษาในโครงงานชิ้นนี้ ที่ได้ให้ความรู้และคำแนะนำต่างๆเกี่ยวกับโครงงาน ทำให้โครงงานนี้มีความคืบหน้าในการทำงานเป็นอย่างมาก จึงขอขอบคุณอาจารย์ เฉลิมพันธุ์ หวังวิวัฒนา ที่ให้ความกรุณาเป็นอย่างสูงครับ

รชต เตชะเวชเจริญ
สุรวีร์ ลือลีลาโรจน์
อติวุฒิ เพชรธนสาร

คณะผู้จัดทำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

หน้า

บทคัดย่อภาษาไทย.....I

บทคัดย่อภาษาอังกฤษ.....II

กิตติกรรมประกาศ.....III

สารบัญ.....IV

สารบัญรูปภาพ.....VI

สารบัญตาราง.....X

บทที่ 1 บทนำ

1.1 ความสำคัญและความเป็นมา.....1

1.2 วัตถุประสงค์ของโครงงาน.....2

1.3 ขอบเขตของโครงงาน.....2

1.4 ประโยชน์ที่ได้รับจากโครงงาน.....2

บทที่ 2 ทฤษฎี

2.1 ฮีตเตอร์.....3

2.2 ไดแอก ไทรแอกและการใช้งาน.....9

2.3 Sensor DS1820.....20

2.4 ไมโครคอนโทรลเลอร์ (PIC16F877A).....26

2.5 LCD.....48

บทที่ 3 การออกแบบ

3.1 Block Diagram.....54

3.2 ส่วนประกอบของเตาอบควบคุมอุณหภูมิ.....54

3.3 PROCESS CONTROL.....57

3.4 การใช้โมดูล CCP1 และ TIMER2 กำเนิดสัญญาณ.....69

3.5 RS-232.....71

3.6 การทำงานของเตาอบควบคุมอุณหภูมิ.....80

3.7 รูปวงจรส่วนประกอบของเตาอบควบคุมอุณหภูมิ.....80

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

หน้า

3.8 รูปส่วนประกอบของวงจรจับเวลา.....81
3.9 รูปส่วนประกอบของวงจรไดร์ฟพัดลม 12V.....81
3.10 รูปส่วนประกอบของวงจรไดร์ฟฮีตเตอร์.....81
3.11 รูปเครื่องควบคุมอุณหภูมิและตาอบ.....82
บทที่ 4 ผลการทดลอง.....83
บทที่ 5 สรุปผลการทดลองและข้อเสนอแนะ.....96

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูปภาพ

หน้า

รูปที่ 2.1 ฮีตเตอร์แท่ง หรือ Cartridge Heater.....4
รูปที่ 2.2 ฮีตเตอร์แบบ High Density.....4
รูปที่ 2.3 ฮีตเตอร์แท่งแบบ Low Density.....5
รูปที่ 2.4 ฮีตเตอร์ครีบ หรือ Finned Heater.....5
รูปที่ 2.5 ฮีตเตอร์จุ่ม หรือ Immersion Heater.....6
รูปที่ 2.6 บอบบิ้นฮีตเตอร์ หรือ Bobbin Heater..... 6
รูปที่ 2.7 ฮีตเตอร์อินฟาเรด หรือ Infrared Heater.....7
รูปที่ 2.8 ฮีตเตอร์รัดท่อ หรือ Band Heater.....8
รูปที่ 2.9 ฮีตเตอร์แผ่น หรือ Strip.....8
รูปที่ 2.10 โครงสร้างและสัญลักษณ์.....9
รูปที่ 2.11 วงจรสมมูลของไดแอกและการไบอัสไดแอก.....10
รูปที่ 2.12 กราฟคุณสมบัติไดแอก.....10
รูปที่ 2.13 โครงสร้าง สัญลักษณ์และวงจรสมมูลเปรียบเทียบระหว่างไตรแอกกับเอสซีอาร์.....11
รูปที่ 2.14 สภาวะที่ 1 หรือควอนแดรนด์ที่ 1.....12
รูปที่ 2.15 สภาวะที่ 2 หรือควอนแดรนด์ที่ 2.....13
รูปที่ 2.16 สภาวะที่ 3 หรือควอนแดรนด์ที่ 3.....13
รูปที่ 2.17 สภาวะที่ 4 หรือควอนแดรนด์ที่ 4.....14
รูปที่ 2.18 กราฟลักษณะสมบัติของไตรแอก.....14
รูปที่ 2.19 วงจรหรี่ไฟ.....16
รูปที่ 2.20 วงจรหรี่ขด.....16
รูปที่ 2.21 วงจรหรี่ไฟแสงสว่างที่มีชุดป้องกันสัญญาณรบกวน.....17
รูปที่ 2.22 วงจรปรับความเร็วมอเตอร์แบบยูนิเวอร์แซล.....17
รูปที่ 2.23 วงจรควบคุมแรงดันไฟสลับ 1 เฟส ด้วยไตรแอค และSCR.....18
รูปที่ 2.24 การต่อไตรแอคแบบแยกแต่ละไลน์.....18
รูปที่ 2.25 การต่อไตรแอคแบบอันดับร่วมกับโหลด.....19
รูปที่ 2.26 การต่อไตรแอคแบบแยกแต่ละไลน์.....19

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูปภาพ (ต่อ)

หน้า

รูปที่ 2.27 การต่อไตรแอกแบบรวมกัน.....20
รูปที่ 2.28 โครงสร้าง และขาของ DS18B20 ตัวถังแบบ TO-92.....20
รูปที่ 2.29 โครงสร้างรีจิสเตอร์ภายในของ DS18B20.....21
รูปที่ 2.30 โครงสร้างภายในรีจิสเตอร์ Temperature LSB และ MSB.....21
รูปที่ 2.31 การต่อแบบใช้ไฟเลี้ยง R Pull-up.....22
รูปที่ 2.32 การต่อแบบจ่ายไฟเลี้ยงให้กับขา VDD 22
รูปที่ 2.33 การต่อใช้งาน DS18B20.....23
รูปที่ 2.34 การเริ่มการติดต่อสื่อสารแบบ 1 - wire ด้วย Reset Pulse และ Presence Pulse.....23
รูปที่ 2.35 การเขียนข้อมูลลง DS18B20.....24
รูปที่ 2.36 การอ่านข้อมูลจาก DS18B20.....24
รูปที่ 2.37 โครงสร้างและส่วนประกอบหลักเบื้องต้นของไมโครคอนโทรลเลอร์.....26
รูปที่ 2.38 โครงสร้างสถาปัตยกรรมของไมโครคอนโทรลเลอร์แบบพรินซ์ตัน 27
รูปที่ 2.39 โครงสร้างสถาปัตยกรรมของไมโครคอนโทรลเลอร์แบบฮาร์วาร์ด.....28
รูปที่ 2.40 ส่วนประกอบหลักของซีพียูในไมโครคอนโทรลเลอร์.....29
รูปที่ 2.41 กลไกการทำงานของสแต็กอย่างง่าย.....33
รูปที่ 2.42 สถาปัตยกรรมภายในของไมโครคอนโทรลเลอร์ PIC16F877A.....36
รูปที่ 2.43 ตัวถังของ PIC16F877A และตำแหน่งขาสัญญาณต่าง ๆ.....37
รูปที่ 2.44 การวางพื้นที่ของหน่วยความจำโปรแกรมของ PIC16F877A.....40
รูปที่ 2.45 การจัดสรรพื้นที่หน่วยความจำข้อมูลของ PIC16F877A.....41
รูปที่ 2.46 โครงสร้างของคำสั่งแบบไบต์.....44
รูปที่ 2.47 โครงสร้างของคำสั่งแบบบิต.....45
รูปที่ 2.48โครงสร้างของคำสั่งแบบ Literal and Control.....47
รูปที่ 3.1 บล็อกไดอะแกรมแสดงส่วนประกอบต่างๆ.....54
รูปที่ 3.2 Sensor DS18S20.....54
รูปที่ 3.3 แสดงFlow chart การทำงานเซนเซอร์DS18S20.....55
รูปที่ 3.4 keypad 4*3.....56
รูปที่ 3.5 จอแสดงผล LCD 16*2.....56
รูปที่ 3.6 แสดง Transient ในระบบควบคุมแบบป้อนกลับ.....59

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูปภาพ (ต่อ)

หน้า

รูปที่ 3.7 แสดง Feedback control.....64

รูปที่ 3.8 แสดง Feed forward Control.....65

รูปที่ 3.9 แสดง Feedback control loop.....66

รูปที่ 3.10 แสดง basic control loop.....67

รูปที่ 3.11 แสดง time constant.....67

รูปที่ 3.12 แสดง accuracy profile.....68

รูปที่ 3.13 การทำงานของโมดูล CCP1/TIMER2 ในการกำเนิดสัญญาณ.....69

รูปที่ 3.14 การกำหนดค่าให้กับรีจิสเตอร์ CCPR1L:CCP1CON<5:4>.....70

รูปที่ 3.15 ระดับสัญญาณแบบ TTL.....72

รูปที่ 3.16 แสดงการส่งข้อมูลแบบขนานและแบบอนุกรม.....74

รูปที่ 3.17 แสดงการส่งข้อมูลแบบอนุกรมด้วยความเร็ว 9600 บิตต่อวินาที.....75

รูปที่ 3.18 แสดงการส่งข้อมูลขนาด 8 บิตแบบอนุกรมพร้อมกับบิตเริ่มต้น,บิตพาริตี้,บิตหยุด ด้วยความเร็ว9600 บิตต่อวินาที.....76

รูปที่ 3.19 แสดงระดับแรงดันสัญญาณของพอร์ตอนุกรม RS-232 กับ TTL ในสถานะลอจิก "1" และ "0".....77

รูปที่ 3.20 ไอซี MAX232.....77

รูปที่ 3.21 การต่อMAX232 กับ connecter DB9.....78

รูปที่ 3.22 ลักษณะคอนเน็กเตอร์แบบDB9.....78

รูปที่ 3.23 Flow Chart การทำงานของเตาอบควบคุมอุณหภูมิ.....80

รูปที่ 3.24 ส่วนประกอบวงจรที่ต่อกับไมโครคอนโทรลเลอร์.....80

รูปที่ 3.25 วงจรจับเวลา.....81

รูปที่ 3.26 วงจรพัดลม 12 v.....81

รูปที่ 3.27 วงจร ไดร์ฟฮีตเตอร์.....81

รูปที่ 3.28 เครื่องควบคุมอุณหภูมิ.....82

รูปที่ 3.29 เครื่องควบคุมอุณหภูมิ.....82

รูปที่ 3.30 เครื่องควบคุมอุณหภูมิและเตา.....82

รูปที่ 4.1 แสดงTemperature = 40 C , 15 min ,Kp=20, Ki=10, Kd=0.....83

รูปที่ 4.2 แสดงTemperature = 50 C , 15 min ,Kp=20, Ki=10, Kd=0.....87

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 4.3 แสดงTemperature = 60 C , 15 min ,Kp=20, Ki=10, Kd=0……..90
รูปที่ 4.4 แสดงTemperature = 40 C , 15 min ,Kp=10, Ki=0, Kd=0……...94
รูปที่ 4.5 แสดงTemperature = 50 C , 15 min ,Kp=10, Ki=0, Kd=0……...94
รูปที่ 4.6 แสดงTemperature = 60 C , 15 min ,Kp=10, Ki=0, Kd=0……...95

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

หน้า

ตารางที่ 1 การเลือกแบงก์ข้อมูลของหน่วยความจำ.....42
ตารางที่ 2 คำสั่งที่มีโครงสร้างแบบไบต์.....45
ตารางที่ 3 คำสั่งที่มีโครงสร้างแบบ Literal and Control.....47
ตารางที่ 4 ความสัมพันธ์ของขา RS , R/W และ E.....53

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

1.1 ความสำคัญและความเป็นมา

เตาอบไฟฟ้าได้รับการออกแบบและผลิตขึ้นเพื่อสามารถใช้แทนเตาอบแบบ ดั้งเดิม คือเตาอบที่ใช้ถ่านฟืนหรือแก๊สธรรมชาติเป็นเชื้อเพลิง เตาอบไฟฟ้าได้รับความนิยมใน หมู่ชาวตะวันตก การใช้ไฟฟ้าเป็นแหล่งพลังงานทำให้ใช้งานได้ง่าย และสะดวกเนื่องจาก สามารถควบคุมอุณหภูมิภายในเตาอบ และตั้งเวลาการเปิดปิดโดยอัตโนมัติหลักการทำงานของ เตาอบไฟฟ้า คือการทำให้อาหารสุกโดยแปลงพลังงานไฟฟ้าเป็นพลังงานความร้อน และใช้วิธีการแผ่รังสีความร้อน (Convection) ที่เกิดจากขดลวดความร้อนให้กับโหลดหรืออาหารจน อุณหภูมิที่ชิ้นอาหารปรับสูงขึ้นทำให้อาหารสุก โดยเป็นการแผ่ความร้อนจากพื้นผิวของอาหารเข้าสู่ศูนย์กลาง ซึ่งนับเป็นวิธีที่มนุษย์ใช้ในการประกอบอาหารหรือขนมมาเป็นเวลาช้านานตั้งแต่ใช้ฟืนเป็นเชื้อเพลิง ในปัจจุบันเตาอบไฟฟ้าไม่ได้รับการพัฒนาทางเทคโนโลยีมากนักเนื่องจากผู้ผลิตได้หันมาพัฒนาเตาอบไมโครเวฟกันมากขึ้นแต่อาหารบางชนิดก็ยังจำเป็นต้อง ใช้เตาอบไฟฟ้าในการประกอบอาหารอยู่ เช่น การทำขนมเค้ก การอบพิซซ่า เป็นต้น โดยปกติ แล้ว เตาอบไฟฟ้าจะใช้พลังงานมากกว่าเตาอบไมโครเวฟ ส่วนปริมาณการใช้พลังงานจะมากหรือน้อยขึ้นอยู่ที่ขนาดความจุของตู้เป็นสำคัญ เตาอบไฟฟ้าส่วนมากจะใช้พลังงานอยู่ระหว่าง 650 - 1,500 วัตต์ และองค์ประกอบที่ทำให้อาหารร้อนเร็วหรือช้าขึ้นอยู่กับชนิดรูปร่าง ปริมาณ อาหารที่นำเข้าตู้อบในการเปลี่ยนพลังงาน ไฟฟ้าเป็นพลังงานความร้อนนั้นจะผ่านอุปกรณ์ที่ใช้เปลี่ยนพลังงานไฟฟ้าเป็นพลังงานความร้อน โดยใช้หลักการง่ายๆไม่ซับซ้อนวัสดุที่ทำจะมีคุณสมบัติเป็นตัวต้านทานทางไฟฟ้า รูปแบบมักจะเป็นขดลวดความร้อนหรือแท่งฉนวนหุ้มปิดด้วยวัสดุพาความร้อนหรือเป็นลักษณะคล้ายแท่งแก้วเซรามิกเมื่อผ่านกระแสไฟฟ้าเข้าไปจะเกิดความร้อนมีอุณหภูมิสูงขึ้นอาจมีพัดลมช่วยระบายความร้อนให้ทั่วถึงทั้งเตาอบหรือไม่มีก็ได้ เตาอบไฟฟ้าจะต่างจากเตาอบไมโครเวฟในเรื่องของการเปิดปิดประตูของเตาเนื่องจากเตาไฟฟ้าไม่ก่อกำเนิดคลื่นความถี่สูงใดๆ จึงไม่มีการติดตั้งสวิทช์ที่ประตูของเตาขณะเปิดประตูเตาไฟฟ้าจึงยังคงทำงานอยู่ต่อไปจนกว่าจะตัดการจ่ายกระแสไฟฟ้า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. ขนาดความจุของเตาอบไฟฟ้า

ตามนิยามของสหภาพยุโรปซึ่งได้กำหนดกฎเกณฑ์การใช้พลังงานของเตาอบไฟฟ้า(EU Commission Directive 2002/40/EC) ได้กำหนดขนาดความจุของเตาอบไฟฟ้าไว้ดังนี้
ขนาดเล็ก ปริมาตรความจุมากกว่าหรือเท่ากับ 12 ลิตรแต่ไม่เกินกว่า 35 ลิตร ขนาดกลาง ปริมาตรความจุมากกว่าหรือเท่ากับ 35 ลิตรแต่ไม่เกินกว่า 65 ลิตร ขนาดใหญ่ ปริมาตรความจุมากกว่าหรือเท่ากับ 65 ลิตรขึ้นไปจากการศึกษาพบว่าเตาอบไฟฟ้าที่จำหน่ายในประเทศไทยส่วนมากจะเป็นขนาดเล็กและขนาดกลางตามข้อกำหนดของสหภาพยุโรปสำหรับขนาดใหญ่ที่มีความจุมากกว่า 65ลิตรมักจะเป็นเตาอบเชิงพาณิชย์ที่ใช้ในร้านทำอาหารหรือขนมอบและจะมีรูปแบบใหญ่ที่เคลื่อนย้ายได้ยากต่างจากเตาอบแบบตั้งโต๊ะ เตาอบที่มีจำหน่ายในประเทศไทยมีความจุเริ่มต้นที่ 7 ลิตรซึ่งมีความจุที่น้อยกว่าที่กำหนดไว้ในข้อกำหนดของสหภาพยุโรปอาจเนื่องมาจากเตาอบขนาดเล็กมีราคาที่ประหยัดและสามารถใช้งานที่เหมาะสมกับความต้องการของคนไทย

1.2 วัตถุประสงค์ของโครงงาน

1.2.1 เพื่อศึกษาวิธีการควบคุมแบบระบบปิด (Close Loop)มาประยุกต์ใช้งานกับอุปกรณ์อื่นๆได้
1.2.2 เพื่อศึกษาหลักการทำงานของเซนเซอร์วัดอุณหภูมิ
1.2.3 เพื่อศึกษาและประยุกต์ใช้งานไมโครคอนโทรเลอร์กับอุปกรณ์อื่นๆได้
1.2.4 ประยุกต์ใช้ความรู้การออกแบบวงจรเพื่อให้วงจรทำงานได้อย่างมีประสิทธิภาพ

1.3ขอบเขตของโครงงาน

การออกแบบและการสร้างวงจรควบคุมเตาอบนี้ อาศัยการเปลี่ยนแปลงพลังงานความร้อนที่ผ่านขดลวด โดยมีเทอร์โมคับเปิลเป็นตัวรับความร้อนและเปลี่ยนเป็นสัญญาณ ไฟฟ้า จากนั้นADC จะทำหน้าที่แปลงสัญญาณอนาลอกเป็นสัญญาณดิจิตอลเพื่อส่งให้ตัวควบคุม CPU ประมวลผล แล้วสั่งการออกไปผ่าน DAC เพื่อแปลงสัญญาณดิจิตอลเป็นอนาลอก ไปยังแผงวงจรไฟฟ้าที่จะคอยสวิทซ์ให้เตาอบกำเนิดความร้อนตามอุณหภูมิที่ต้องการ โดยมีช่วงอุณหภูมิระหว่าง -55 C ถึง125 C โดยมีช่วงห่าง0.5C และสามารถทำงานได้อย่างมีประสิทธิภาพ

1.4 ประโยชน์ที่ได้รับจากโครงงาน

1.3.1 สามารถพัฒนาความรู้ที่ได้จากการเรียนไปประยุกต์ใช้งานจริงในการประดิษฐ์เตาอบ
1.3.2 สามารถใช้งานบอร์ดแสดงผล LCD ในการแสดงค่าได้จริง
1.3.3 เข้าใจการเขียนโปรแกรม C ในการออกแบบ PIC16F877A

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2
ทฤษฎี

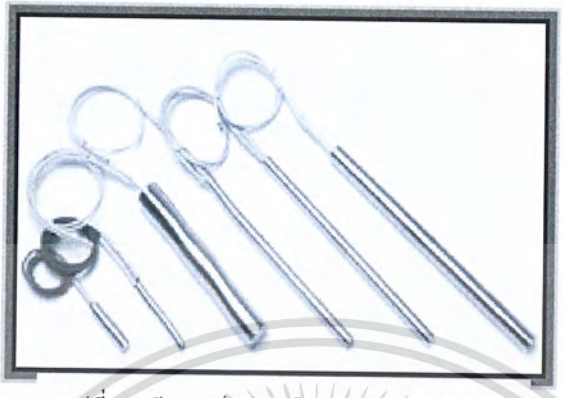
2.1 ฮีทเตอร์

Heater ถูกแบ่งออกเป็นชนิดต่างๆ 7 ชนิด ตามลักษณะการใช้งานที่แตกต่างกันดังนี้

1. ฮีตเตอร์แท่ง หรือ Cartridge Heater
   - ใช้ความร้อนกับวัสดุที่เป็นของแข็ง เช่น เหล็ก และโลหะต่างๆ
   - ตัวอย่างการใช้งาน เช่น งานบรรจุหีบห่อ งานขึ้นรูปพลาสติก
2. ฮีตเตอร์ครีบ หรือ Finned Heater
   - ใช้ให้ความร้อนกับอากาศ เช่น ใช้ในห้องอบแห้ง ในเตาอบ
3. ฮีตเตอร์จุ่ม หรือ Immersion Heater หรือบางทีเรียกว่า ฮีตเตอร์ต้มน้ำ
   - ใช้ให้ความร้อนกับของเหลวทุกชนิด
   - ตัวอย่างการใช้งาน เช่น งานต้มน้ำ ต้มน้ำมัน งานผสมสาร
4. บอบบิ้นฮีตเตอร์ หรือ Bobbin Heater
   - ใช้ให้ความร้อนกับของเหลวทุกชนิด
5. ฮีตเตอร์อินฟาเรด หรือ Infrared Heater
   - ใช้ให้ความร้อนกับวัตถุโดยไม่ต้องสัมผัสโดยตรง ไม่เหมาะกับวัตถุที่มีลักษณะมันวาว เนื่องจากวัตถุมันวาวจะมีสมบัติสะท้อนแสง ทำให้ไม่สามารถดูดซับแสงอินฟาเรดได้อย่างเต็มที่
   - ตัวอย่างการใช้งาน เช่น ติดตั้งในตู้อบ
6. ฮีตเตอร์รัดท่อ หรือ Band Heater
   - ใช้ให้ความร้อนกับของเหลวที่อยู่ในท่อหรือทรงกระบอกโดยรัดจากด้านนอก
7. ฮีตเตอร์แผ่น หรือ Strip Heater
   - ใช้ให้ความร้อนโดยแนบกับวัตถุโดยตรง สามารถออกแบบให้เป็นรูปทรงใดๆก็ได้

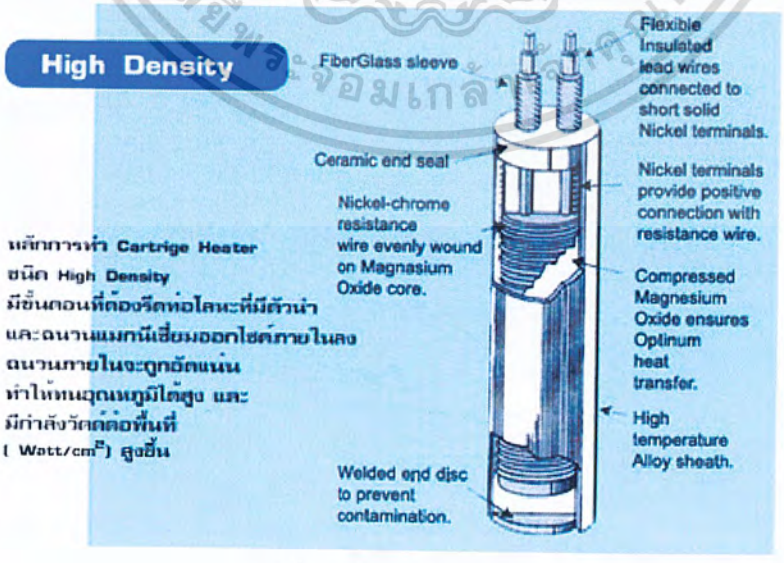
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ฮีตเตอร์แท่ง หรือ Cartridge Heater



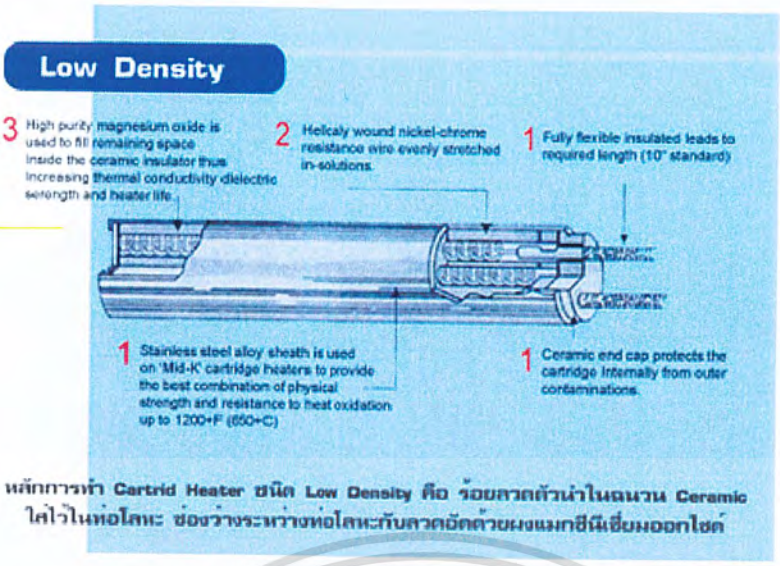
รูปที่ 2.1 ฮีตเตอร์แท่ง หรือ **Cartridge Heater**

ลักษณะการใช้งานทั่วไปของ Cartridge Heater คือ ใส่ไว้ในช่องบนวัตถุ ความร้อนจะถูกส่งผ่านจากฮีตเตอร์ ไปยังวัตถุที่ต้องการให้ความร้อน ตัวอย่างงาน เช่น ให้ความร้อนแม่พิมพ์ของเครื่องบรรจุหีบห่อ Cartridge Heater แบ่งออกเป็น 2 ชนิด คือ High Density และ Low Density (บางครั้งเรียกว่า High Temperature Low Temperature) การกำหนดว่า Cartridge Heater ตัวใดเป็น High Density และ Low Density จะพิจารณาจากค่า Watt Density ซึ่งเป็นหน้าที่ของทางผู้ผลิต ทาง TIC สามารถผลิตฮีตเตอร์ให้ตามที่ต้องการได้



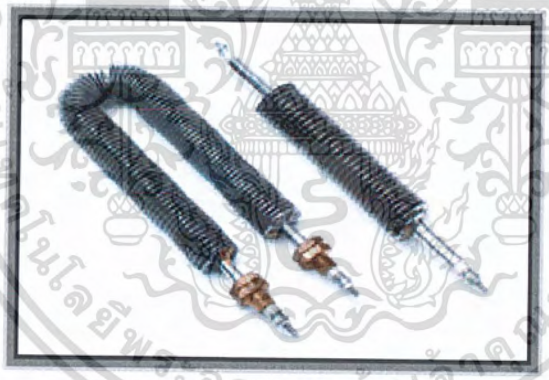
รูปที่ 2.2 ฮีตเตอร์แบบ High Density

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.3 ฮีตเตอร์แท่งแบบ Low Density

ฮีตเตอร์ครีบ หรือ Finned Heater

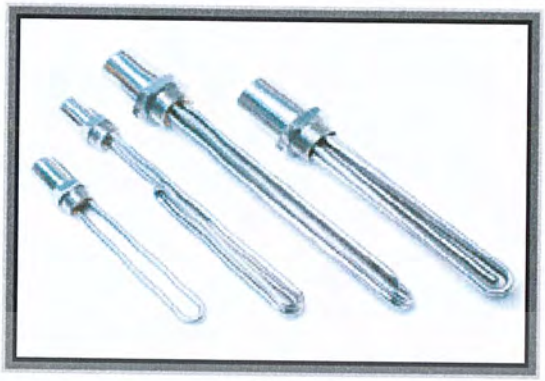


รูปที่ 2.4 ฮีตเตอร์ครีบ หรือ **Finned Heater**

ฮีตเตอร์ครีบและฮีตเตอร์ท่อกลม ใช้กับงานต่อไปนี้ ใช้ในเตาอบ ,ใช้ในท่อ DUCT , ใช้กับเครื่องปรับอากาศ การติดตั้งสามารถทำได้ 2 วิธีคือ ติดตั้งแบบให้ความร้อนโดยตรง และ แบบส่งผ่านความร้อนจากห้องเผาไปยังห้องอบโดยใช้ลมร้อน เป็นฮีตเตอร์ที่ใช้กับอากาศ ไม่ควรใช้กับของเหลว เนื่องจากจะเกิดตะกอนจับที่ครีบของฮีตเตอร์ทำให้ความร้อนไม่สามารถถ่ายเทได้ ในกรณีที่ให้ความร้อนกับอากาศที่ไม่หมุนเวียน ควรเลือกวัสดุที่ใช้ทำฮีตเตอร์เป็นอินโคลอย เนื่องจากมีคุณสมบัติถ่ายเทความร้อนได้ และทนอุณหภูมิได้กว่าชนิดอื่น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ฮีตเตอร์จุ่ม หรือ Immersion Heater



รูปที่ 2.5 ฮีตเตอร์จุ่ม หรือ **Immersion Heater**

ทำจาก Tubular Heater ที่ดัดเป็นรูปตัวยู และเชื่อมติดกับเกลียว ซึ่งมีขนาดเกลียวตั้งแต่ 1" 1¼ 1½ 2½ ขนาดของเกลียวจะขึ้นอยู่กับจำนวนเส้นของฮีตเตอร์ ตามความเหมาะสมของกำลังวัตต์ และความยาวของฮีตเตอร์ ฮีตเตอร์แบบจุ่ม เหมาะสำหรับใช้กับของเหลว เช่น ต้มน้ำหรืออุ่นน้ำมัน การติดตั้งสามารถทำได้โดยเชื่องเกลียวตัวเมียติดกับถังแล้วใส่ฮีตเตอร์ แบบเกลียวเข้าไป โดยตัวฮีตเตอร์ขนานกับพื้นถัง ควรระวังไม่ให้ส่วนของฮีตเตอร์โผล่พ้นของเหลวเนื่องจากจะทำให้ส่วนที่อยู่เหนือของเหลวร้อนจัดเกินไปทำให้อายุการใช้งานสั้น และเพื่อให้ความร้อนกระจายทั่วถึง ควรติดตั้งใบพัดกวนของเหลวด้วย

บอบบิ้นฮีตเตอร์ หรือ Bobbin Heater

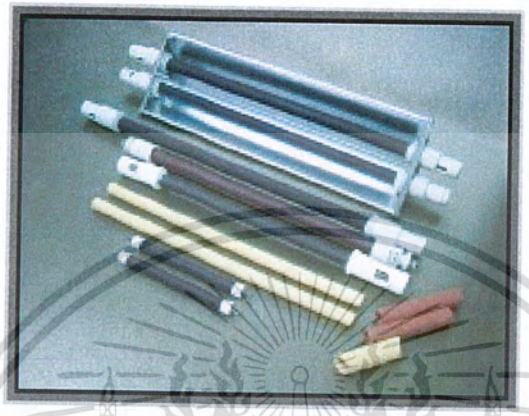
รูปที่ 2.6 บอบบิ้นฮีตเตอร์ หรือ **Bobbin Heater**

เป็นฮีตเตอร์แบบจุ่มชนิดหนึ่ง ถูกออกแบบสำหรับให้ความร้อนกับของเหลว สามารถเคลื่อนย้ายได้ง่าย ปลอดฮีตเตอร์สามารถเลือกให้เหมาะสมกับการใช้งานมีให้เลือกทั้งสแตน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เลส 304, สแตนเลส316 และควอทซ์ โดยแบบสแตนเลสมีข้อดี คือ เมื่อฮีตเตอร์เสียสามารถซ่อมได้ แบบควอกซ์ ใช้สำหรับงานชุบโดยใช้ไฟฟ้า, แช่ในกรด หรือ สารละลาย

ฮีตเตอร์อินฟาเรด หรือ Infrared Heater

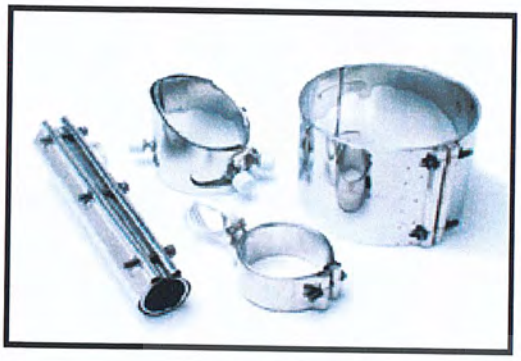


รูปที่ 2.7 ฮีตเตอร์อินฟาเรด หรือ **Infrared Heater**

ลักษณะการทำงานของ Infrared Heater เป็นการส่งผ่านความร้อนแบบแผ่รังสี จึงมีประสิทธิภาพสูง ความสูญเสียต่ำ ประหยัดไฟได้ 30 – 50 %สามารถให้ความร้อนวัตถุได้ถึงเนื้อใน จึงทำให้ประหยัดเวลาได้ 1 – 10 เท่า (การให้ความร้อนการพา และการนำความร้อนจะทำให้วัตถุร้อนเฉพาะที่ผิว แล้วค่อยๆ ซึมเข้าๆไปเนื้อในจึงใช้เวลามาก)มีขนาดเล็กกว่าฮีตเตอร์แบบทั่วๆไป ทำให้ประหยัดเนื้อที่ การติดตั้ง และการถอดเปลี่ยนเพื่อซ่อมบำรุงง่ายมีความปลอดภัยสูง เนื่องจากไม่มีเปลวไฟ ตัวเรือนมีความเป็นฉนวนสูง ไฟไม่รั่วให้รังสี 3 – 10 ไมโครเมตร ซึ่งเป็นช่วงที่วัสดุเกือบทุกชนิดสามารถดูดซับรังสีได้ดี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ฮีตเตอร์รัดท่อ หรือ Band Heater



ได้รับการออกแบบสำหรับท่อ หรือถังรูปทรงกระบอก ฉนวนของฮีตเตอร์ทำจากแผ่น Micea และลวดฮีตเตอร์เป็นแบบแบน (Ribbon Wire Heating Element) จึงทำให้ฮีตเตอร์ชนิดนี้มีเส้นผ่าศูนย์กลางเล็กขนาด 25 มม. หรืออาจใหญ่ถึง 600 มม. ก็ได้ ส่วนความกว้างอยู่ระหว่าง 20 – 300 มม. ตัวถังด้านนอกเป็นแผ่นเหล็กหรือสแตนเลส เหมาะสมสำหรับให้ความร้อนกับเครื่องฉีดพลาสติกมีอีกชื่อหนึ่งว่า ฮีตเตอร์กระบอก

รูปที่ 2.8 ฮีตเตอร์รัดท่อ หรือ **Band Heater**

ฮีตเตอร์แผ่น หรือ Strip Heater

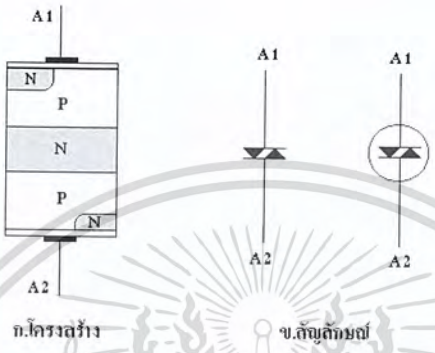
รูปที่ 2.9 ฮีตเตอร์แผ่น หรือ **Strip**

โครงสร้างจะเป็นฮีตเตอร์แบบเดียวกับฮีตเตอร์ รัดท่อแต่รูปทรงจะเป็นแบบสี่เหลี่ยมจตุรัส หรือสี่เหลี่ยมพืนผ้า เหมาะสมสำหรับให้ความร้อนกับแม่พิมพ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2 ไดแอค ไทรแอคและการใช้งาน

ไดแอค(Triac) เป็นอุปกรณ์สารกึ่งตัวนำที่อยู่ในกลุ่มของของไทริสเตอร์ มี 2 ขั้วคือ ขั้วแอโนด 1 (A_1) และขั้วแอโนด 2 (A_2) เพราะไดแอคสามารถนำกระแสได้สองด้าน ไดแอคสามารถนำไปใช้กับแรงดันไฟฟ้าสลับและแรงดันไฟฟ้ากระแสตรงได้



รูปที่ 2.10 โครงสร้างและสัญลักษณ์

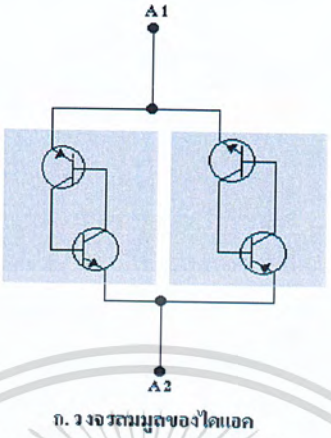
จากรูปที่ 2.1 เป็นโครงสร้างและสัญลักษณ์ของไดแอค จากโครงสร้างเป็นอุปกรณ์สารกึ่งตัวนำมี 3 ตอนใหญ่ชนิดสาร PNP และยังประกอบด้วยสารกึ่งตัวนำ 2 ตอนย่อยชนิด N ต่อร่วมในสารกึ่งตัวนำชนิด P ทั้ง 2 ตอนด้านนอก มีขาต่อออกมาใช้งานเพียง 2 ขา แต่ละขาที่ต่อใช้งานจะต่อร่วมกับสารกึ่งตัวนำทั้งชนิด N และชนิด P จึงทำให้ไดแอคสามารถทำงานได้ทั้งแรงดันไฟบวกและแรงดันไฟลบ ขาแอโนด1 (A_1) เรียกว่า ขาเทอมินอล 1 (Main Terminal 1) ใช้ตัวย่อ MT_1 และขาแอโนด2 (A_2) เรียกว่า ขาเทอมินอล 2 (Main Terminal 2) ใช้ตัวย่อ MT_2 แต่ละขาสามารถต่อสลับกันได้

การทำงานของไดแอค

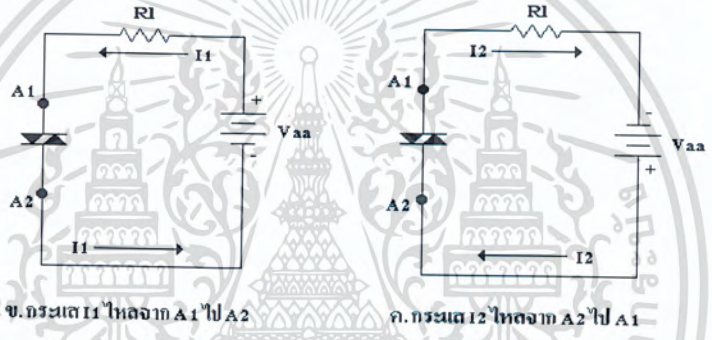
ไดแอคมี 2 ขา แต่มีคุณสมบัติสามารถทำงานได้กับแรงดันช่วงบวก และแรงดันช่วงลบคือกระแสได้ 2 ทิศทาง ดังนั้นในการใช้งานจึงไม่จำเพาะเจาะจงในการต่อวงจร ใช้ขาด้านใดด้านหนึ่งต่อเข้าวงจรก็จะได้คุณสมบัติเหมือนกัน การทำงานของไดแอคเปรียบเหมือนกับชอคเลย์ไดโอด 2 ตัวต่อกลับหัวกัน มีวงจรสมมูลของไดแอคแทนด้วยทรานซิสเตอร์ 4 ตัว ดังรูปที่ 2.2 ก ทรานซิสเตอร์ Q_1 และ Q_2 แทนชอคเลย์ไดโอดตัวที่ 1 จะเป็นตัวนำกระแสแอโนดมาจากขั้ว A_1 ไปสู่ขั้ว A_2 ทรานซิสเตอร์ Q_3 และ Q_4 แทนแทนชอคเลย์ไดโอดตัวที่ 2 จะเป็นตัวนำกระแสแอโนดมาจากขั้ว A_2 ไปสู่ขั้ว A_1 ไดแอคเมื่อไบอัสตรงให้ขา A_1 กับ A_2 ไดแอคทำงานเมื่อแรงดันไบอัสตรงที่จ่ายให้ตัวไดแอคถึงค่าแรงดันเบรคโอเวอร์ของตัวไดแอค $V_{BR(F)}$ จะทำให้กระแสไหลผ่านขั้ว A_1 ไปยังขั้ว A_2 และ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อไดแอคทำงานเมื่อไบอัสกลับจ่ายให้ตัวไดแอคถึงค่าแรงดันเบรคโอเวอร์ของตัวไดแอค $-V_{BR(R)}$ จะทำให้กระแสไหลผ่านขั้ว A_2 ไปยังขั้ว A_1 ดังรูปที่ 2.2 ข และ 2.2 ค



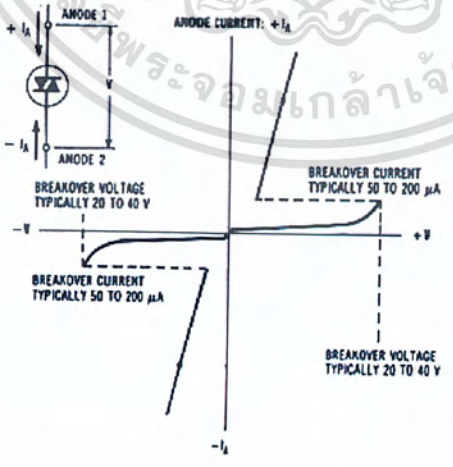
ก. วงจรสมมูลของไดแอค



ข. กระแส I1 ไหลจาก A1 ไป A2 ค. กระแส I2 ไหลจาก A2 ไป A1

รูปที่ 2.11 วงจรสมมูลของไดแอคและการไบอัสไดแอค

กราฟคุณสมบัติของไดแอค



รูปที่ 2.12 กราฟคุณสมบัติไดแอค

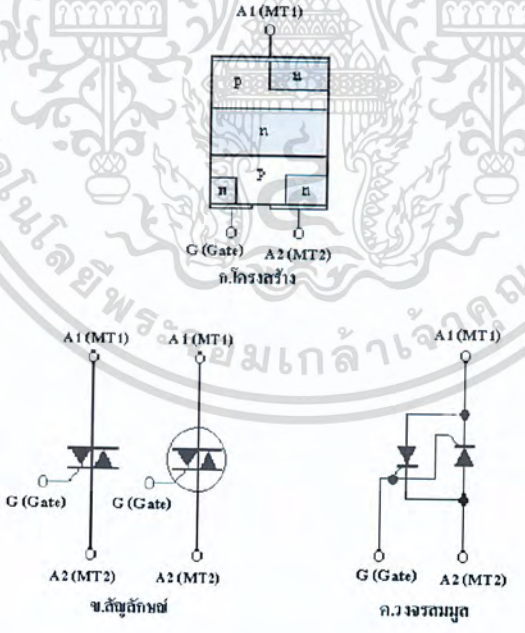
จากรูปที่2.3เป็นกราฟแสดงคุณสมบัติของตัวไดแอคโดยไดแอคสามรถนำกระแสได้ทั้งช่วงบวกและช่วงลบของแรงดันไฟสลับ การจ่ายแรงดันให้ไดแอคในครั้งแรกจะมีกระแสไหลผ่านตัวไดแอค

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

น้อยมาก เรียกว่ากระแสรั่วซึมเมื่อเพิ่มแรงดันให้ตัวไดแอคมากขึ้นจนถึงค่าแรงดันเบรคโอเวอร์ V_{BO} จะทำให้มีกระแสไหลผ่านตัวไดแอคถึงค่ากระแสเบรคโอเวอร์ I_{BO} ตัวไดแอคจะยอมให้กระแสไหลผ่านได้ทำให้ความต้านทานในตัวไดแอคลดลง ยิ่งจ่ายแรงดันไบอัสให้ตัวไดแอคมากขึ้นจะยิ่งทำให้ความต้านทานในตัวไดแอคยิ่งลดลง กระแสจะยิ่งไหลเพิ่มมากขึ้น

ค่าความต้านทานในตัวไดแอคที่ลดลงเมื่อมันนำกระแสจะลดลงน้อยและยังคงเหลือค่าความต้านทานในตัวไดแอค คุณสมบัตินี้จึงนิยมนำไดแอคไปใช้เป็นตัวป้อนกระแสกระตุ้นขา G ของไตรแอค เพราะช่วยป้องกันกระแสกระชากจากแหล่งจ่ายไปทำให้ขา G ของไตรแอคชำรุดเสียหายได้

ไตรแอค(Triac) เป็นอุปกรณ์จำพวกสารกึ่งตัวนำในกลุ่มของไทริสเตอร์ มีลักษณะโครงสร้างภายในคล้ายกับไดแอค แต่มีขาเกตเพิ่มขึ้นมาอีก 1 ขา ไตรแอตถูกสร้างขึ้นเพื่อแก้ไขข้อบกพร่องของ SCR ซึ่งไม่สามารถนำกระแสในซีกลบของไฟฟ้าสลับได้ การนำไตรแอคไปใช้งานส่วนใหญ่จะใช้ทำเป็นวงจรควบคุมการทำงานเป็นสวิตซ์ต่อแรงดันไฟสลับ ไตรแอคถูกสร้างขึ้นมาให้ใช้งานกับกระแสสูง ๆ ดังนั้นต้องระวังเรื่องของการระบายความร้อน สัญลักษณ์,โครงสร้างและวงจรสมมูลของไตรแอคดังรูปที่ 2.4 ก,ขและค



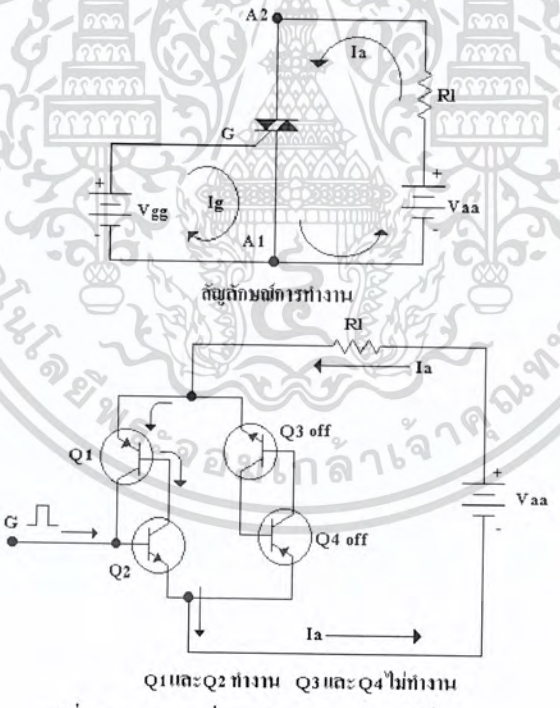
รูปที่ 2.13 โครงสร้าง สัญลักษณ์และวงจรสมมูลเปรียบเทียบระหว่างไตรแอคกับเอสซีอาร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โครงสร้างของไตรแอคจะประกอบด้วยสารกึ่งตัวนำตอนใหย่ 3 ตอน คือ PNP และในสารกึ่งตัวนำตอนใหญ่จะมีสารกึ่งตัวนำตอนย่อยชนิด N อีก 3 ตอน ต่อร่วมในสารกึ่งตัวนำ P ทั้ง 2 ตอน มีขาต่อออกมาใช้งาน 3 ขา เหมือน SCR ขาแอโนด 1 (A1) เรียกว่า ขาเทอร์มินอล 1(Main terminal 1) MT_1 ขาแอโนด 2 (A2) เรียกว่า ขาเทอร์มินอล 2(Main terminal 2) MT_2 และขาเกท(Gate) G การทำงานของไตรแอค

ไตรแอคมีคุณสมบัติ ทำงานได้ทั้งแรงดันช่วงบวกและแรงดันช่วงลบ การนำกระแสของไตรแอคจะขึ้นอยู่กับแรงดันที่ป้อนกระตุ้นขา G และแรงดันที่จ่ายให้ขา A_2 และ A_1 การจ่ายไบอัสให้ตัวไตรแอคสามารถแบ่งได้เป็น 4 สภาวะคือ

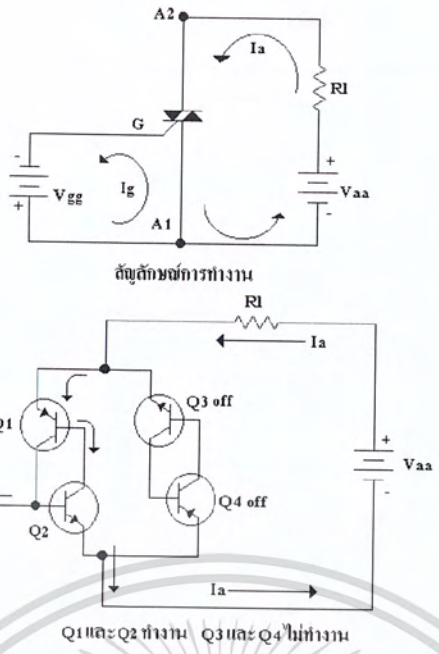
1.สภาวะที่ 1 หรือควอนแดรนด์ที่ 1 จ่ายแรงดันบวกให้ขา A_2 จ่ายแรงดันลบให้ขา A_1 และจ่ายแรงดันบวกกระตุ้นขา G จะเกิดการนำกระแสในตัวไตรแอคดังรูปที่ 2.5 ทิศทางการไหลของกระแสทั้งสองจะไหลในทิศทางเดียวกัน หรือกระแสไหลเสริมกัน ทำให้ I_A ไหลมากขึ้น



รูปที่ 2.14 สภาวะที่ 1 หรือควอนแดรนด์ที่ 1

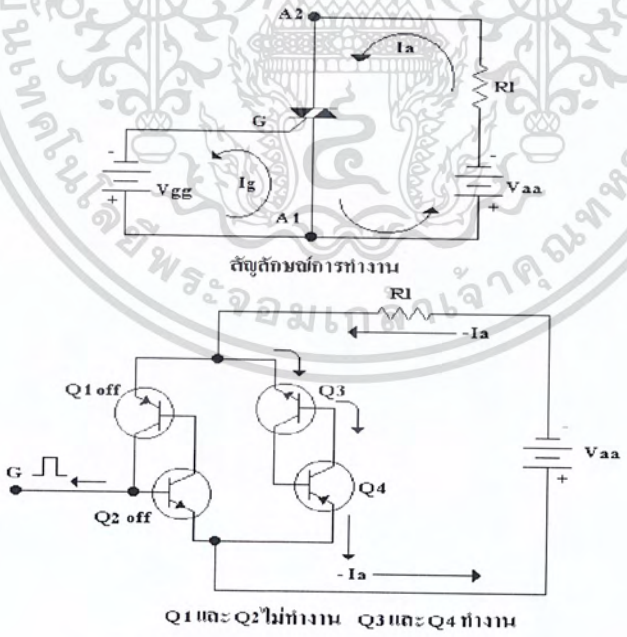
2.สภาวะที่ 2 หรือควอนแดรนด์ที่ 2 จ่ายแรงดันบวกให้ขา A_2 จ่ายแรงดันลบให้ขา A_1 แต่จ่ายแรงดันลบกระตุ้นขา G จะเกิดการนำกระแสในตัวไตรแอคดังรูปที่ 2.6 ทิศทางการไหลของกระแสทั้งสองจะไหลในทิศทางสวนทาง หรือกระแสไหลหักล้างกัน ทำให้ I_A ไหลน้อยลง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.15 สภาวะที่ 2 หรือควอนแดรนด์ที่ 2

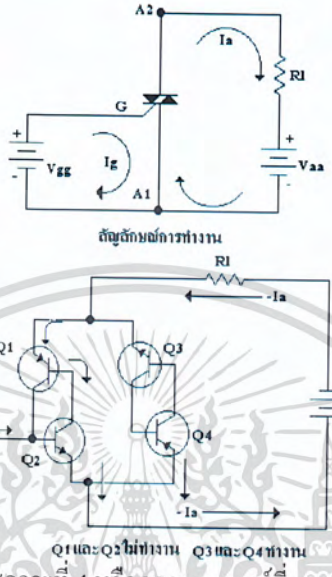
3.สภาวะที่ 3 หรือควอนแดรนด์ที่ 3 จ่ายแรงดันลบให้ขา A_2 จ่ายแรงดันบวกให้ขา A_1 แต่จ่ายแรงดันลบกระตุ้นขา G จะเกิดการนำกระแสในตัวไตรแอคดังรูปที่ 2.7 ทิศทางการไหลของกระแสทั้งสองจะไหลในทิศทางเดียวกัน หรือกระแสไหลเสริมกัน ทำให้ I_A ไหลมากขึ้น



รูปที่ 2.16 สภาวะที่ 3 หรือควอนแดรนด์ที่ 3

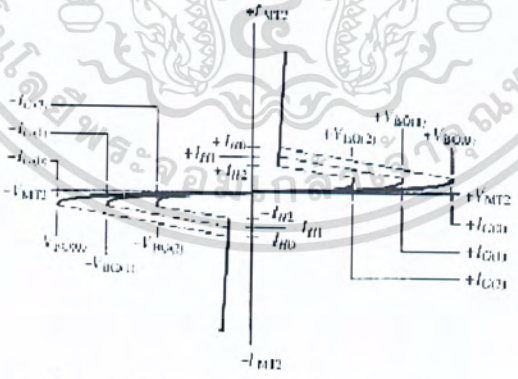
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.สภาวะที่ 4 หรือควอนแดรนด์ที่ 4 จ่ายแรงดันลบให้ขา A_2 จ่ายแรงดันบวกให้ขา A_1 แต่จ่ายแรงดันบวกกระตุ้นขา G จะเกิดการนำกระแสในตัวไตรแอคดังรูปที่ 2.7 ทิศทางการไหลของกระแสทั้งสองจะไหลในทิศทางสวนทาง หรือกระแสไหลหักล้างกัน ทำให้ I_A ไหลน้อยลง



รูปที่ 2.17 สภาวะที่ 4 หรือควอนแดรนด์ที่ 4

กราฟคุณสมบัติของไตรแอค



รูปที่ 2.18 กราฟลักษณะสมบัติของไตรแอค

จากกราฟลักษณะสมบัติของไตรแอค แรงดันไบอัสตรงกับไตรแอค($V_{BO(0)}$) และจ่ายกระแสเกตบวก(I_G) ให้กับเกตของไตรแอค จะทำให้ไตรแอคนำกระแสได้โดยกระแสจะไหลจากขั้ว A_1 ไปยังขั้ว A_2 และเมื่อป้อนแรงดันไบอัสลบให้กับไตรแอค($-V_{BO(0)}$) โดยไม่ให้เกินกว่าค่าแรงดันพังทลาย สามารถจ่ายกระแสเกตลบ($-I_G$) กระแสจะไหลจากขั้ว A_2 ไปยังขั้ว A_1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.1การหยุดการนำกระแสของไตรแอค

ไตรแอคเมื่อนำกระแสแล้วจะไม่จำเป็นต้องคงค้างแรงดันที่จ่ายกระตุ้นขาGเพราะไตรแอคจะนำกระแสต่อเนื่องได้เหมือนกับ SCR จะหยุดนำกระแสทำได้ 2 วิธีเหมือน SCR คือ

1.ตัดแหล่งจ่ายแรงดัน V_{AA} ที่ป้อนให้ขา A_2 และขา A_1 ของไตรแอคออกชั่วขณะ

2.ลดแรงดันไบอัสตรงที่จ่ายให้ขา A_2 และ A_1 ลง จนทำให้มีกระแสไหลผ่านตัวไตรแอคต่ำกว่ากระแสโฮลดิ้ง(holding current) ของไตรแอค

รายละเอียดและขีดจำกัดของไตรแอค

1.แรงดันบล็อกกิ้งสูงสุดซ้ำ (Repetitive Peak Blocking Voltage) ใช้ตัวย่อเป็น V_{DRM} หรือ V_{RPM} คือแรงดันสูงสุดที่ป้อนเฉพาะขาA_2 และ A_1 ที่ตัวไตรแอคยังไม่นำกระแสโดยไม่มีแรงดันกระตุ้นที่ขา G

2.กระแสไบอัสตรงสูงสุดเป็น rms (RMS Max Forward Current) ใช้ตัวย่อ $I_{T(MAX)}$ คือค่ากระแสสูงสุดเป็น rms ที่ไตรแอคทนได้ เมื่อมีกระแสไบอัสตรงไหลผ่าน ถ้ากระแสไหลผ่านไตรแอคเกินกว่าค่านี้ไตรแอคจะชำรุดเสียหาย

3.กระแสโฮลดิ้ง (Holding Current) ใช้ตัวย่อ I_H คือกระแสต่ำสุดที่ไหลผ่านตัวไตรแอคระหว่างขา A_2 และ A_1 แล้วไตรแอคยังคงนำกระแสได้

4.กระแสกระตุ้นเกท(Gate Trigger Current) ใช้ตัวย่อ I_{GT} คือกระแสที่ใช้กระตุ้นขา G ของไตรแอคเพื่อทำให้ไตรแอคทำงาน มักบอกค่ากระแสไว้ที่ต่ำสุด

5.แรงดันกระตุ้นเกท(Gate Trigger Voltage) ใช้ตัวย่อ V_{GT} คือแรงดันที่ป้อนให้ขา G เทียบกับขา A_1 เพื่อกระตุ้นให้ไตรแอคนำกระแส มักบอกค่าสูงสุด

6.กระแสกระตุ้นเกทสูงสุด(Peak Gate Trigger Current) ใช้ตัวย่อ I_{GTM} คือค่ากระแสกระตุ้นที่ไหลผ่านเกทค่าสูงสุดยังคงทำให้ไตรแอคทำงานได้

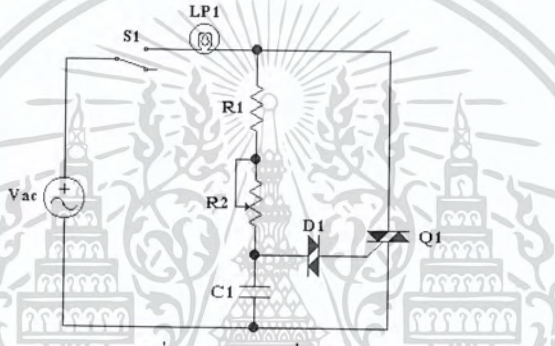
7.ย่านอุณหภูมิในการทำงาน(Operating Junction Temperature) ใช้ตัวย่อ T_j คือย่านอุณหภูมิตรงรอยต่อที่ไตรแอค ทำงานได้โดยไม่ชำรุดเสียหาย บอกค่าตั้งแต่ต่ำสุด(-) จนถึงสูงสุดเป็นค่าบวก(+)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.2 การประยุกต์ไดแอคและไตรแอคไปใช้งาน

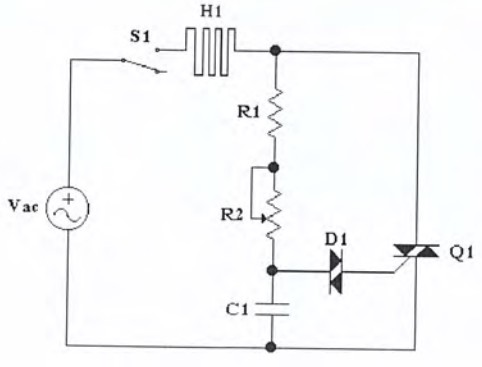
การนำไตรแอคไปใช้งานนิยมใช้มากสุด คือ การนำไปใช้ควบคุมกำลังไฟฟ้ากระแสสลับที่โหลด เช่น ควบคุมความสว่างของหลอดไฟฟ้า หรือวงจรหรี่ไฟ ควบคุมความเร็วมอเตอร์ไฟฟ้ากระแสสลับ ควบคุมความร้อนของลวดความร้อน เช่น เตาอบไฟฟ้า เตาหลอดโลหะ เป็นต้น

1.วงจรหรี่ไฟหรือขดลวดความร้อน โดยใช้ไตรแอคและไดแอคเป็นตัวทำงานวงจรหรี่ไฟหรือขดลวดความร้อน โดยการทำงานของไตรแอคทำหน้าที่เป็นสวิตช์ตัดต่อวงจรตามการควบคุมแรงดันกระตุ้นที่ขา G ของไตรแอค วงจรควบคุมและหน่วงเวลาแรงดันกระตุ้นขา G ใช้ตัวต้านทานและตัวเก็บประจุ มีไดแอคเป็นตัวป้องกันกระแสกระชากเข้าขา Gของไตรแอค ดังรูปที่ 2.10



รูปที่ 2.19 วงจรหรี่ไฟ

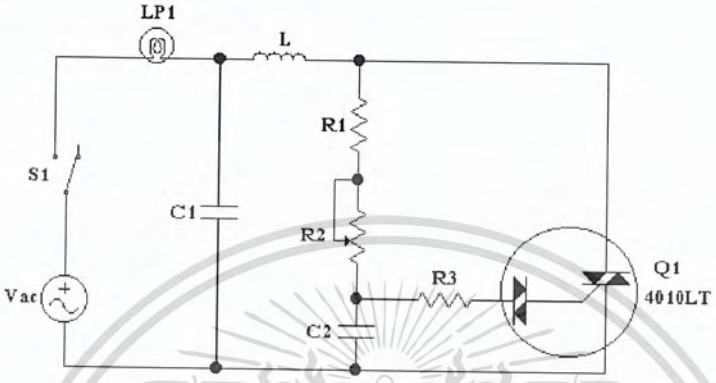
การทำงานดังกล่าวทำให้สามารถปรับเปลี่ยนความสว่างของหลอดไฟ LP ได้ โดยอาศัยการปรับเปลี่ยนเวลาและมุมเฟสของแรงดันที่จะทำให้ไดแอคนำกระแส และไปทำให้ไตรแอคนำกระแสกำลังไฟฟ้าที่ตกคร่อมหลอดไฟ LP จะเปลี่ยนแปลงตามไปด้วย ถ้านำวงจรดังกล่าวไปหรี่ความร้อนของเตาไฟฟ้า กระทะไฟฟ้า กาต้มน้ำไฟฟ้า เตารีดไฟฟ้าและอื่น ๆ ดังรูปที่ 2.11



รูปที่ 2.20 วงจรหรี่ขดลวดความร้อน

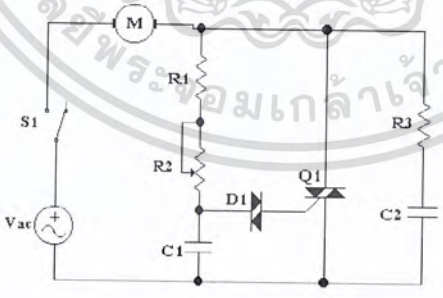
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.วงจรหรี่ไฟแสงสว่างที่มีชุดป้องกันสัญญาณรบกวน วงจรจะใช้ควอดแดรกเป็นตัวทำงาน วงจรหรี่ไฟแสงสว่างที่มีชุดป้องกันสัญญาณรบกวน จะมีวงจรคล้ายกับวงจรหรี่ไฟในรูปที่ 2.10 แต่เพิ่มวงจรป้องกันสัญญาณรบกวนเข้าไป ช่วยป้องกันกระแสกระชากผ่านเข้าวงจร และป้องกันฮาร์โมนิกส์สูง ๆ ออกไปรบกวนระบบไฟฟ้าภายนอก ดังรูปที่ 2.12



รูปที่ 2.21 วงจรหรี่ไฟแสงสว่างที่มีชุดป้องกันสัญญาณรบกวน

3.วงจรปรับความเร็วมอเตอร์แบบยูนิเวอร์แซล มอเตอร์แบบยูนิเวอร์แซล(Universal Motor) เป็นมอเตอร์ที่สามารถนำไปใช้งานได้ทั้งแรงดันไฟตรง DC และแรงดันไฟสลับ AC จึงเรียกว่า ยูนิเวอร์แซล มอเตอร์ ในการควบคุมความเร็วของยูนิเวอร์แซลมอเตอร์ สามารถทำได้โดยใช้การควบคุมเวลาและเฟสของอุปกรณ์ทริกเกอร์ ร่วมกับ SCR ไตรแอค หรือควอดแตรก อย่างใดอย่างหนึ่ง ดังรูปที่ 2.22



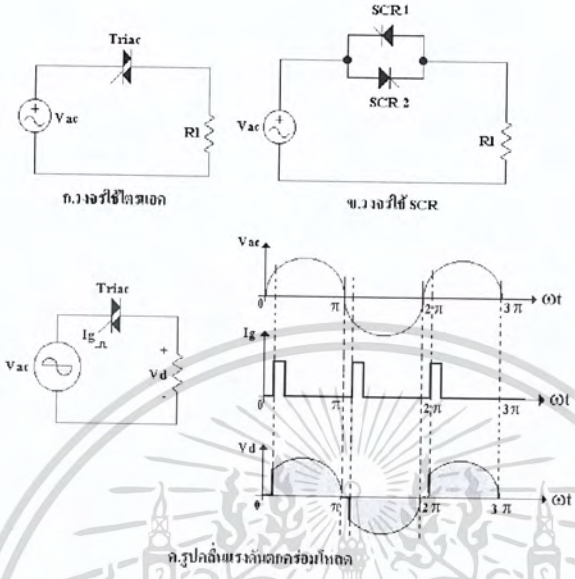
รูปที่ 2.22 วงจรปรับความเร็วมอเตอร์แบบยูนิเวอร์แซล

2.2.3การใช้ไตรแอคควบคุมเฟสของแรงดันไฟสลับ

ไตรแอคสามารถนำกระแสในไฟสลับได้2ซีกคือ ซีกบวกและซีกลบ ในการควบคุมไฟสลับสามารถทำได้กับ ไฟสลับ 1 เฟส และ 3 เฟส ดังนี้คือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
119449

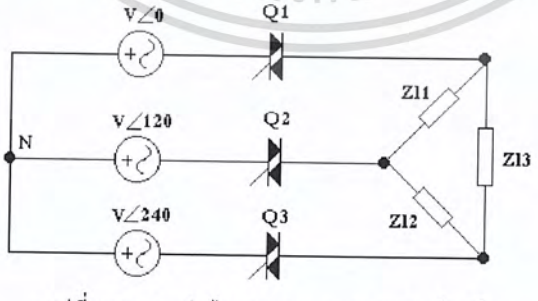
1.ควบคุมแรงดันไฟสลับชนิดเฟสเดียว ถ้าใช้ไตรแอคจะใช้เพียงตัวเดียว แต่ถ้าใช้ SCR จะต้องใช้ 2 ตัว ดังรูปที่ 2.23



รูปที่ 2.23 วงจรควบคุมแรงดันไฟสลับ 1 เฟส ด้วยไตรแอค และSCR

2.ควบคุมแรงดันไฟสลับ ชนิด 3 เฟส แรงดันไฟสลับ 3 เฟส แต่ละเฟสมีมุมต่างกัน 120 ^{0} องศา ในแต่ละอินพุตป้อนผ่านเข้าแต่ละโหลดโดยเฉพาะ ในแต่ละชุดของการจ่ายแรงดันไปยังโหลด จะถูกควบคุมแรงดันด้วยไตรแอคแต่ละตัว โดยใช้ไตรแอคหนึ่งต่อกับโหลดหนึ่งชุด สามารถนำไปใช้กับโหลดที่ต่อวงจรทั้งแบบเดลต้าและแบบสตาร์ ดังนี้คือ

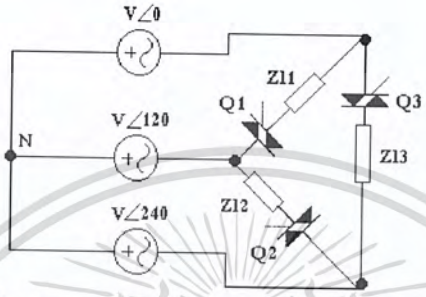
2.1 แบบเดลต้า การต่อวงจรแบบนี้ จะใช้ไตรแอคเป็นตัวควบคุมเฟสในการทำงานจะสามารถต่อไตรแอคร่วมในวงจรได้ทั้งแบบต่อไตรแอคแยกในแต่ละไลน์ และต่อไตรแอกอันดับร่วมกับดหลดแต่ละชุด ดังรูปที่ 2.24



รูปที่ 2.24 การต่อไตรแอคแบบแยกแต่ละไลน์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

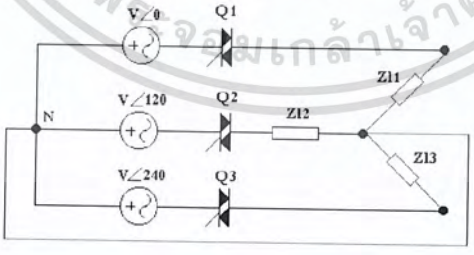
เป็นวงจรใช้ไตรแอคควบคุมเฟสของแรงดันไฟสลับที่จ่ายโหลด โหลด Z_{L1} จะทำงานเมื่อไตรแอค Q_1 และ Q_2 เกิดการนำกระแส โหลด Z_{L2} จะทำงานเมื่อไตรแอค Q_2 และ Q_3 เกิดการนำกระแส และโหลด Z_{L3} จะทำงานเมื่อไตรแอค Q_3 และ Q_1 เกิดการนำกระแส



รูปที่ 2.25 การต่อไตรแอคแบบอันดับร่วมกับโหลด

จากรูปที่ 2.25 เป็นวงจรที่ใช้ไตรแอคควบคุมต่อแบบอันดับ สภาวะการทำงานของโหลดจะขึ้นอยู่กับไตรแอคแต่ละไลน์ เมื่อไตรแอค Q_1 นำกระแส โหลด Z_{L1} จะทำงาน ไตรแอค Q_2 นำกระแส โหลด Z_{L2} จะทำงาน และไตรแอค Q_3 นำกระแส โหลด Z_{L3} จะทำงาน

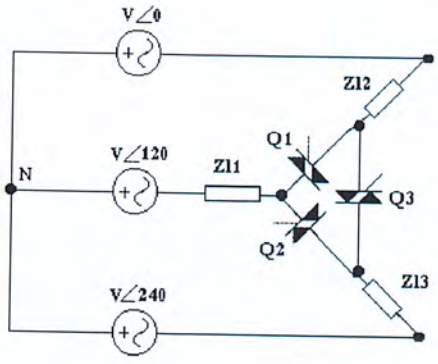
2.2 แบบสตาร์ การต่อวงจรแบบสตาร์โดยใช้ไตรแอคเป็นตัวควบคุมเฟสในการทำงานสามารถต่อไตรแอคเข้าวงจรได้ทั้งแบบต่อไตรแอคแยกแต่ละไลน์ และแบบไตรแอคต่อรวมกับแล้วจึงต่อไปโหลดแต่ละโหลด ดังรูปที่ 2.26



รูปที่ 2.26 การต่อไตรแอคแบบแยกแต่ละไลน์

จากรูปที่ 2.26 เป็นวงจรใช้ไตรแอคควบคุมเฟสของแรงดันไฟสลับ โดยโหลดถูกต่อวงจรแบบสตาร์ แหล่งจ่ายแรงดันแต่ละชุด V_a, V_b และ V_C เมื่อไตรแอค Q_1 นำกระแสโหลด Z_{L1} จะทำงาน ไตรแอค Q_2 นำกระแสโหลด Z_{L2} จะทำงาน และไตรแอค Q_3 นำกระแสโหลด Z_{L3} จะทำงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.27 การต่อไตรแอคแบบรวมกัน

จากรูปที่ 2.27 เป็นวงจรใช้ไตรแอคควบคุมเฟสของแรงดันไฟสลับ โหลดต่อแบบสตาร์ ส่วนตัวไตรแอคต่อร่วมกันแบบสตาร์ ไตรแอค Q_1 นำกระแส โหลด Z_{L1} และ Z_{L2} จะทำงาน ไตรแอค Q_2 นำกระแส โหลด Z_{L1} และ Z_{L3} จะทำงาน และไตรแอค Q_3 นำกระแส โหลด Z_{L2} และ Z_{L3} จะทำงาน

2.3 Sensor DS1820

DS18B20 เป็น IC วัดอุณหภูมิแบบดิจิตอล ของ Dallas Semiconductor สามารถวัดอุณหภูมิเป็นหน่วยองศา C ในช่วง -55C ถึง 125C ที่ความละเอียด 9 - 12 บิต และมีความแม่นยำอยู่ที่ 0.5C ในช่วง -10C ถึง 85C ในกรณีที่เป็นตัวถังแบบ TO - 92 นั้นจะมีโครงสร้าง และขาดังแสดงในรูปที่ 1

PIN	SYMBOL	Description
1	GND	Ground
2	DQ	Data Input/ Output pin
3	Vdd	Optional Vdd pin

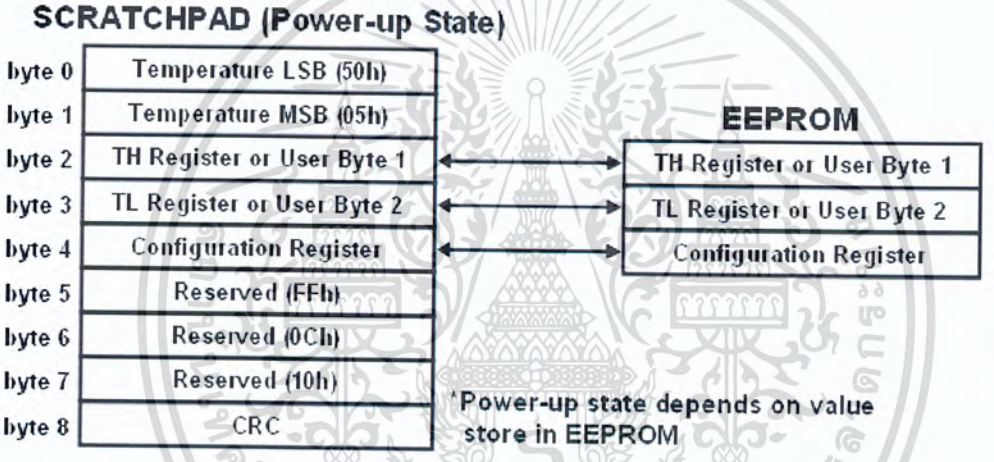
รูปที่ 2.28 โครงสร้าง และขาของ DS18B20 ตัวถังแบบ TO-92

การสื่อสารและควบคุม DS18B20 นั้นสามารถทำได้โดยใช้บัสข้อมูลแบบ 1 - wire ของ Dallas Semiconductor ซึ่งใช้สายสัญญาณเพียงแค่เส้นเดียวเท่านั้น ภายใน DS18B20 แต่ละตัวมีโค้ดประจำตัวขนาด 64 บิต ทำให้สามารถใช้งาน DS18B20 หลายตัวทำงานบนบัสแบบ 1 wire พร้อมกันได้ นอกจากนี้ DS18B20 ยังสามารถทำงานในโหมดพาราสิต (Parasite Power Mode) ซึ่งเป็นการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ทำงานโดยไม่ใช้ไฟเลี้ยง แต่ใช้พลังงานจากสายสัญญาณ 1 - wire ซึ่งมีประโยชน์มากสำหรับการวัดอุณหภูมิระยะไกล หรือในการใช้งานในที่ ๆ มีเนื้อที่จำกัด แต่ในบทความนี้จะกล่าวถึงรายละเอียดการใช้งานขั้นพื้นฐานในโหมดธรรมดาเท่า นั้น สำหรับผู้ที่สนใจสามารถหาข้อมูลเพิ่มเติมได้จาก Datasheet ของDS18B20

โครงสร้างรีจิสเตอร์ภายในของ DS18B20 มีลักษณะดังแสดงในรูปที่ 2 จะเห็นได้ว่าประกอบไปด้วย SRAM Scratchpad ขนาด 9 ไบต์ และ EEPROM ขนาด 3 ไบต์ ซึ่งใช้เก็บค่าอุณหภูมิสูงสุด (TH) ต่ำสุด (TL) สำหรับเปรียบเทียบการเกิดสัญญาณเตือน และรีจิสเตอร์ควบคุม (Configuration Register)



รูปที่ 2.29 โครงสร้างรีจิสเตอร์ภายในของ DS18B20

ข้อมูลอุณหภูมิที่วัดได้จะถูกเก็บอยู่ในรีจิสเตอร์ Temperature ซึ่งมีขนาด 16 บิต ดังแสดงในรูปที่ 2.30 ถ้าข้อมูลอุณหภูมิเป็นบวก S จะเป็น “1” แต่ถ้าข้อมูลอุณหภูมิเป็นลบ S จะเป็น “0” ในกรณีที่ DS18B20 ทำงานในโหมดความละเอียด 12 บิต บิตทุกบิตในรีจิสเตอร์ Temperature จะถูกใช้หมด แต่ในกรณีที่ทำงานในโหมด 9 - 11 บิต บิตล่าง (บิต 0 – บิต 2) จะไม่ถูกใช้งาน ซึ่งในการกำหนดโหมดความละเอียดการทำงานของ DS18B20 นั้นสามารถกำหนดได้ที่รีจิสเตอร์ Configuration ซึ่งโดยปกติเริ่มต้น DS18B20 จะทำงานในโหมด 12 บิต

	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
LS Byte	2^3	2^2	2^1	2^0	2^{-1}	2^{-2}	2^{-3}	2^{-4}
	bit15	bit14	bit13	bit12	bit11	bit10	bit9	bit8
MS Byte	S	S	S	S	S	2^6	2^5	2^4

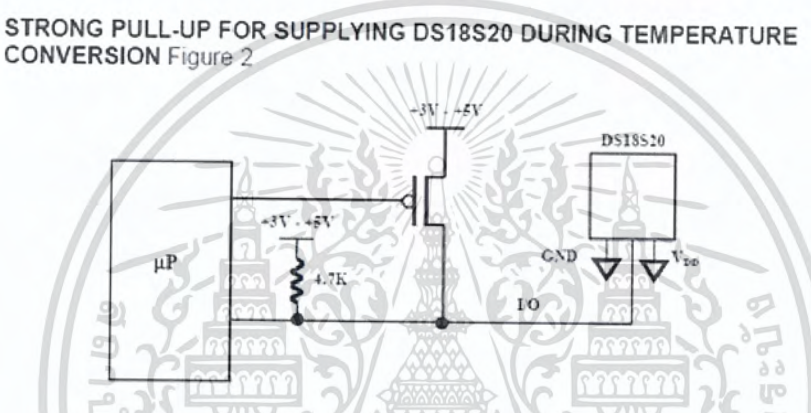
รูปที่ 2.30 โครงสร้างภายในรีจิสเตอร์ Temperature LSB และ MSB

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การสื่อสารแบบ 1 - wire เป็นระบบบัสข้อมูลแบบ Half - duplex นั้นคือสามารถสื่อสารได้ 2 ทิศทาง แต่ไม่สามารถรับ และส่งข้อมูลพร้อมกันในช่วงเวลาเดียวกันได้ ระบบบัสมีการทำงานเป็นแบบ Master/Slave โดยอุปกรณ์ Master จะเป็นตัวควบคุมสถานะ และจังหวะการรับส่งของบัสข้อมูล ในขณะที่อุปกรณ์ Slave จะทำงานตามการควบคุมของอุปกรณ์ Master เท่านั้น

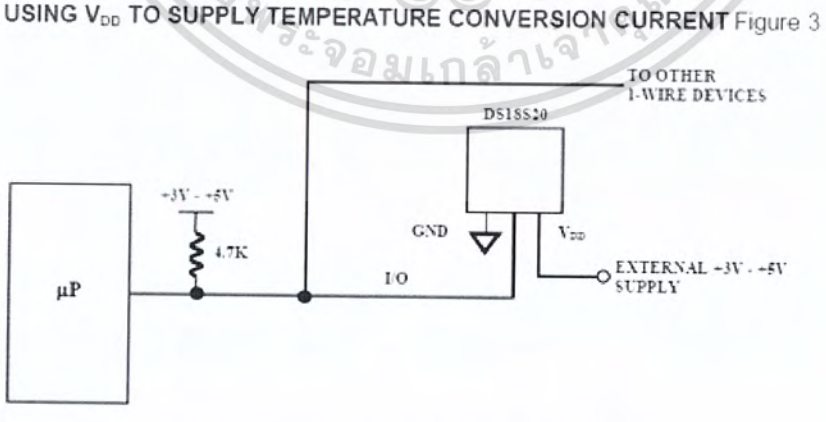
ในการต่อใช้งานจะมีอยู่ 2 วิธีด้วยกัน

1. ใช้ไฟเลี้ยงจาก R Pull-up (PARASITE POWER) วิธีนี้ขา VDD จะต้องต่อลง GND ทำให้ต่อสายเพียง 2 เส้นเท่านั้น



รูปที่ 2.31 การต่อแบบใช้ไฟเลี้ยง R Pull-up

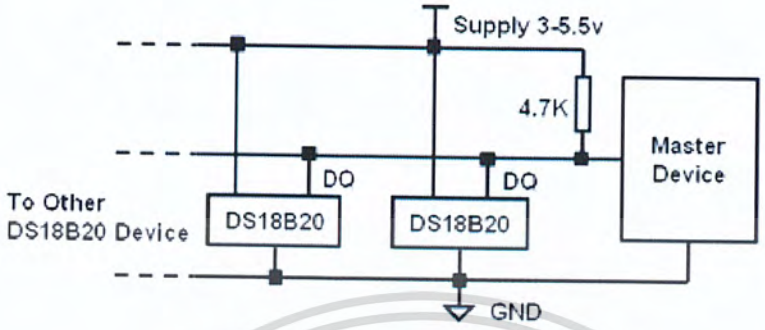
2. ต่อไฟเลี้ยงให้กับขา VDD (External power supply) วิธีนี้จะเป็นที่นิยมใช้กันมากกว่า



รูปที่ 2.32 การต่อแบบจ่ายไฟเลี้ยงให้กับขา VDD

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

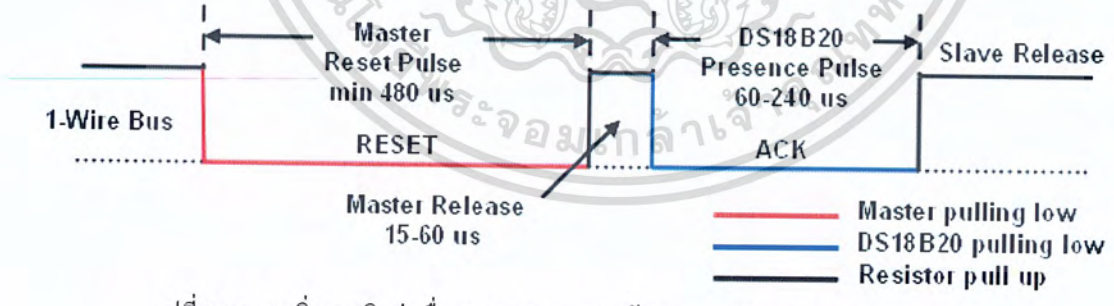
ในการใช้งานบัสแบบ 1 wire นี้ สายสัญญาณข้อมูล DQ จะต้องมีสภาวะปกติที่ลอจิกสูง สามารถทำได้โดยการต่อตัวต้านทานประมาณ 5 กิโลโอห์มพูลอัพไว้กับไฟเลี้ยง หรือในกรณีที่ใช้บัสแบบ 1 wire ต่อร่วมกับอุปกรณ์ DS18B20 หลายตัว ก็สามารถทำได้ดังแสดงในรูปที่ 2.23



รูปที่ 2.33 การต่อใช้งาน DS18B20

รูปแบบของสัญญาณบนบัส 1-wire สามารถแบ่งออกได้เป็น 6 รูปแบบ คือ Reset pulse, Presence pulse, write 0, write 1, read 0, read 1

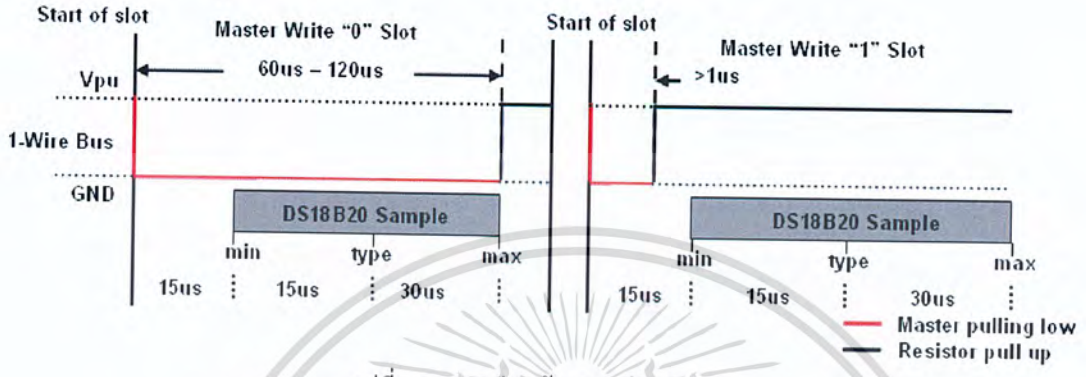
ในกระบวนการเริ่มต้นการสื่อสารแบบ 1 - wire ทั้งหมดนั้น อุปกรณ์ Master ต้องขอเริ่มการสื่อสารด้วยการสร้าง Reset pulse ก่อน เมื่ออุปกรณ์ Slave ได้รับ Reset pulse ก็จะสร้าง Presence pulse เพื่อตอบรับการขอเริ่มการสื่อสารนั้น ซึ่งมีรายละเอียดของช่วงเวลาต่าง ๆ ดังแสดงในรูปที่ 2.24



รูปที่ 2.34 การเริ่มการติดต่อสื่อสารแบบ 1 - wire ด้วย Reset Pulse และ Presence Pulse

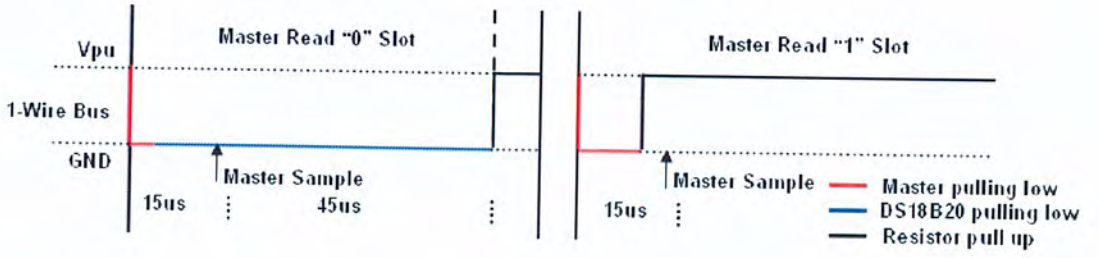
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในการเขียนข้อมูลแบ่งออกเป็น 2 ชนิดคือการเขียนข้อมูล “1” และการเขียนข้อมูล “0” ดังแสดงในรูปที่ 6 การเขียนข้อมูลลง DS18B20 ต้องใช้ช่วงเวลาของไทม์สล็อตอย่างต่ำ 60 µsec และต้องมีช่วงเวลาระหว่างไทม์สล๊อตอย่างต่ำ 1 µsec



รูปที่ 2.35 การเขียนข้อมูลลง DS18B20

การเขียนข้อมูลทั้ง 2 ชนิด เริ่มแรกอุปกรณ์ Master ต้องดึงสัญญาณบนบัส 1-wire ลงมาให้อยู่ในสถานะลอจิกต่ำก่อน ในกรณีที่ต้องการเขียนข้อมูล “0” ลงใน DS18B20 อุปกรณ์ Master ต้องดึงสัญญาณบนบัสให้เป็นลอจิกต่ำต่อ จนกว่าจะครบช่วงเวลาไทม์สล็อต (อย่างต่ำ 60 µsec) ส่วนในกรณีที่ต้องการเขียนข้อมูล “1” ลง DS18B20 อุปกรณ์ Master ต้องปล่อยบัส เพื่อให้บัสกลับไปอยู่ในสถานะลอจิกสูงก่อนการ Sampling ของ DS18B20 ซึ่งจะอยู่ในช่วง 15 µsec - 60 µsec หลังจากที่อุปกรณ์ Master ดึงสัญญาณบัส 1 - wire ลงมา ในการอ่านค่าภายใน SRAM ของ DS18B20 สามารถทำได้ก็ต่อเมื่ออุปกรณ์ Master ได้เขียนข้อมูลเพื่อขอทำการอ่านค่าใน SRAM (Read Scratchpad) ซึ่งมีค่าเป็น 0xBE ลงไปที่ DS18B20 เสียก่อน จากนั้นจึงเริ่มอ่านข้อมูลจากบัส 1 - wire โดยไทม์สล็อตของการอ่านต้องมีช่วงเวลาอย่างต่ำ 60 µsec และต้องมีช่วงเวลาระหว่างไทม์สล็อตอย่างต่ำ 1 µsec ดังแสดงในรูปที่ 7



รูปที่ 2.36 การอ่านข้อมูลจาก DS18B20

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การอ่านข้อมูลจากบัส 1 - wire เริ่มแรกอุปกรณ์ Master จะต้องดึงบัส 1 - wire ลงให้อยู่ในสถานะลอจิกต่ำเป็นช่วงเวลาอย่างน้อย 1 μsec จากนั้นจึงค่อยปล่อยบัส ในกรณีที่ DS18B20 ส่งข้อมูล "0" DS18B20 จะดึงบัสให้เป็นลอจิกต่ำจนจนกว่าจะสิ้นสุดไทม์สล็อตถึงจึงจะปล่อยบัสให้กลับ ไปอยู่ในสถานะลอจิกสูง ส่วนในกรณีที่ DS18B20 ส่งข้อมูล "1" DS18B20 จะปล่อยบัสให้อยู่ในสภานะลอจิกสูงตลอด ในการ Sample เพื่อรับข้อมูลจาก DS18B20 ควรทำภายใน 15 μsec หลังจากจุดเริ่มของไทม์สล็อต

ขั้นตอนการเข้าใช้งาน DS18B20 มี 3 ขั้นตอนด้วยกันคือ

1. Initialization

2. ROM Command

3. DS18B20 Function Command

การ Initialization ประกอบไปด้วยการส่ง Reset Pulse จากอุปกรณ์ Master ตามด้วย Presence Pulse ซึ่งตอบรับโดย DS18B20 เพื่อบ่งบอกว่าอุปกรณ์พร้อมทำงาน

หลังจากการทำ Initialization เสร็จเรียบร้อยแล้ว อุปกรณ์ Master ต้องส่ง ROM Command ไปยัง DS18B20 ROM Command นั้นแบ่งออกได้เป็น 5 คำสั่งด้วยกันคือ SEARCH ROM [F0h], READ ROM [33h], MATCH ROM [55h], SKIP ROM [CCh], ALARM SEARCH [ECh] ซึ่งในกรณีที่ต่อใช้งาน DS18B20 เพียงตัวเดียวนั้นจะใช้ ROM Command ได้แค่ 2 คำสั่งนั้นคือ READ ROM ซึ่งเป็นการอ่านค่า ROM Code ขนาด 64 บิต บนตัว DS18B20 อีกคำสั่งคือ SKIP ROM ซึ่งเป็นคำสั่งที่ใช้ในกรณีที่อุปกรณ์ Master ต้องการส่งคำสั่งควบคุม DS18B20 ทุกตัว ซึ่งไม่จำเป็นต้องระบุ ROM Code

หลังจากที่อุปกรณ์ Master ส่ง Rom Command ไปยัง DS18B20 แล้ว อุปกรณ์ Master จะสามารถใช้ Function Command เพื่อเข้าไปควบคุมการทำงานของ DS18B20 ได้ Function Command ประกอบไปด้วย CONVERT T [44h], WRITE SCRATCHPAD[4Eh], READ SCRATCHPAD[BEh], COPY SCRATCHPAD [48h], RECALL E2[B8h], READ POWER SUPPLY [B4h]

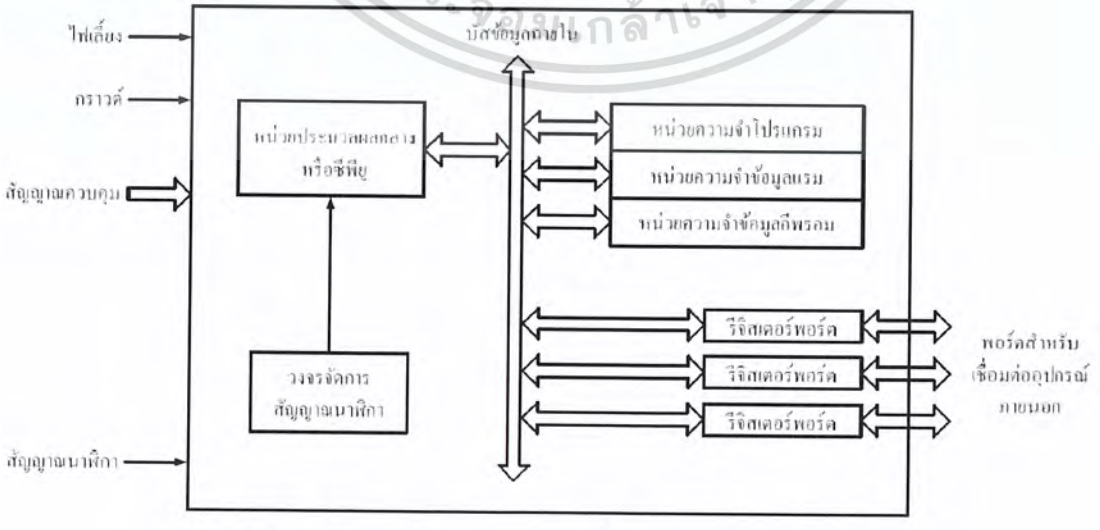
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4 ไมโครคอนโทรลเลอร์ (16F877A)

เนื่องด้วยการควบคุมในโครงงานนี้จะใช้การควบคุมระบบจากไมโครคอนโทรลเลอร์จึงขอกล่าวเกี่ยว กับระบบและทฤษฎีของไมโครคอนโทรลเลอร์ ดังนี้

2.4.1 ความรู้เบื้องต้นเกี่ยวกับไมโครคอนโทรลเลอร์

ไมโครคอนโทรลเลอร์ (Microcontroller) มาจากคำ 2 คำ คำหนึ่งคือไมโคร(Micro) หมายถึงขนาดเล็ก และคำว่า คอนโทรลเลอร์ (Controller) หมายถึง ตัวควบคุมหรืออุปกรณ์ควบคุม ดังนั้นไมโครคอนโทรลเลอร์ จึงหมายถึง อุปกรณ์ควบคุมขนาดเล็ก แต่ในตัวอุปกรณ์ควบคุมขนาดเล็กนี้ได้บรรจุความสามารถที่คล้ายคลึงกับระบบคอมพิวเตอร์ที่คนโดยส่วนใหญ่คุ้นเคย กล่าวคือ ภายในไมโครคอนโทรลเลอร์ได้รวมเอาซีพียู หน่วยความจำ และพอร์ต ซึ่งเป็นส่วนประกอบหลักสำคัญของระบบคอมพิวเตอร์เข้าไว้ด้วยกัน โดยบรรจุรวมกันอยู่ภายใต้ตัวถังเดียวกัน ในภาพที่ 2-12 แสดงส่วนประกอบหลักที่สำคัญและกลไกการทำงานเบื้องต้นของไมโครคอนโทรลเลอร์ ซีพียูจะติดต่อกับหน่วยความจำโปรแกรมเพื่ออ่านคำสั่งที่ระบุไว้ โดยต้องทำการอ้างตำแหน่งของหน่วยความจำผ่านสายสัญญาณที่เรียกว่า บัสตำแหน่ง (Address bus) แล้วทำการอ่านข้อมูลคำสั่งออกมาจากหน่วยความจำโปรแกรมในแอดเดรสนั้นๆ จากนั้นทำการประมวลผล โดยมีหน่วยความจำข้อมูลแรมเป็นที่พักของข้อมูล ที่อยู่ในระหว่างการประมวลผลข้อมูลในการประมวลผล จะส่งผ่านสายสัญญาณที่เรียกว่า บัสข้อมูล (Data bus) แล้วส่งต่อไปยังอุปกรณ์ภายนอกผ่านทางพอร์ตเอาต์พุต

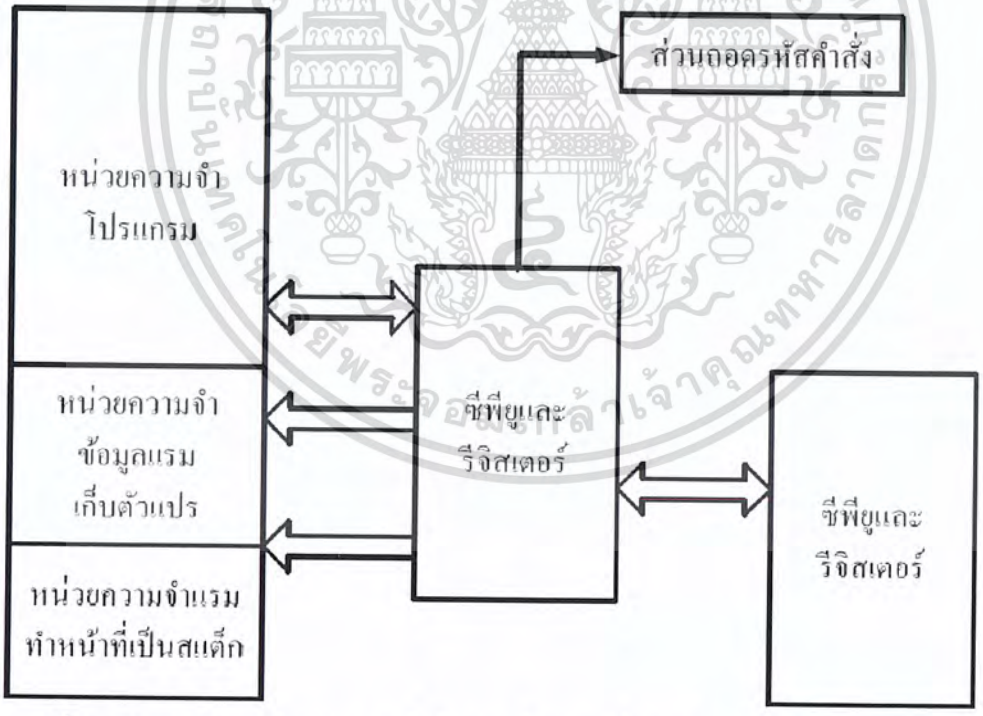


รูปที่ 2.37 โครงสร้างและส่วนประกอบหลักเบื้องต้นของไมโครคอนโทรลเลอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4.1.1 สถาปัตยกรรมของไมโครคอนโทรลเลอร์

เป็นที่ยอมรับกันว่าสถาปัตยกรรมของไมโครคอนโทรลเลอร์มีด้วยกัน 2 แบบคือพรินซ์ตัน (Princeton) หรือ ฟอนนิวแมน (Von Neumann) และ ฮาร์วาร์ด (Harvard) ในภาพที่ 2-13และ 2-14 แสดงการจัดสรรหน่วยความจำและรีจิสเตอร์ในสถาปัตยกรรมของไมโครคอนโทรลเลอร์ทั้งสองแบบ พิจารณาในภาพที่ 2-13 เป็นการจัดสรรในสถาปัตยกรรมแบบพรินซ์ตัน จะเห็นได้ว่ามีโครงสร้างที่เรียบง่ายไม่ซับซ้อน ส่วนของหน่วยความจำโปรแกรมกับหน่วยความจำข้อมูลจะได้รับการจัดสรรให้อยู่รวมกัน ติดต่อกับซีพียูผ่านส่วนจัดการเชื่อมต่อหน่วยความจำ และภายในซีพียูจะมีรีจิสเตอร์บรรจุอยู่ ข้อดีของสถาปัตยกรรมแบบนี้คือ ออกแบบง่าย เพราะหน่วยความจำทั้งหมดอยู่รวมกัน สามารถเข้าถึงได้ง่าย หน่วยความจำมีขนาดใหญ่เพียงพอที่จะเก็บได้ทั้งโปรแกรมควบคุมการทำงาน และข้อมูลของตัวแปรในการประมวลผล ข้อด้อยของสถาปัตยกรรมนี้คือ ความเร็วในการประมวลผล เนื่องจากหน่วยความจำอยู่รวมกัน จึงต้องติดต่อหน่วยความจำโปรแกรมสลับกับหน่วยความจำข้อมูล ส่งผลให้ซีพียูต้องใช้จำนวนไซเคิลในการทำงานมาก

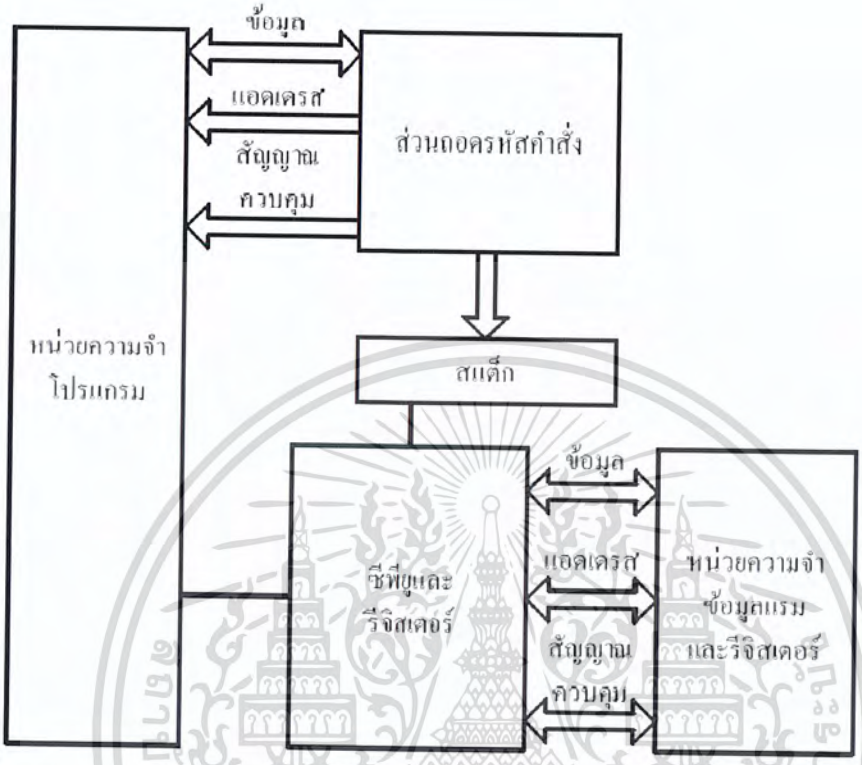


รูปที่ 2.38 โครงสร้างสถาปัตยกรรมของไมโครคอนโทรลเลอร์แบบพรินซ์ตันหรือ ฟอน นิวแมน

ในขณะที่สถาปัตยกรรมแบบฮาร์วาร์ด ซึ่งแสดงในภาพที่ 2-14 จะแยกส่วนของหน่วยความจำ ข้อมูล และรีจิสเตอร์ออกจากหน่วยความจำ โปรแกรมทำให้ไซเคิลการทำงานลดลง เนื่องจากสามารถติดต่อหน่วยความจำโปรแกรม และหน่วยความจำข้อมูลได้เร็วกว่า นอกจากนั้นใน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สถาปัตยกรรมนี้ ในขณะที่ซีพียูกำลังเอ็กซิคิวต์คำสั่งปัจจุบันอยู่สามารถที่จะเฟตช์คำสั่งถัดไปได้ยิ่งทำให้ไมโครคอนโทรลเลอร์ทำงานได้เร็วขึ้น



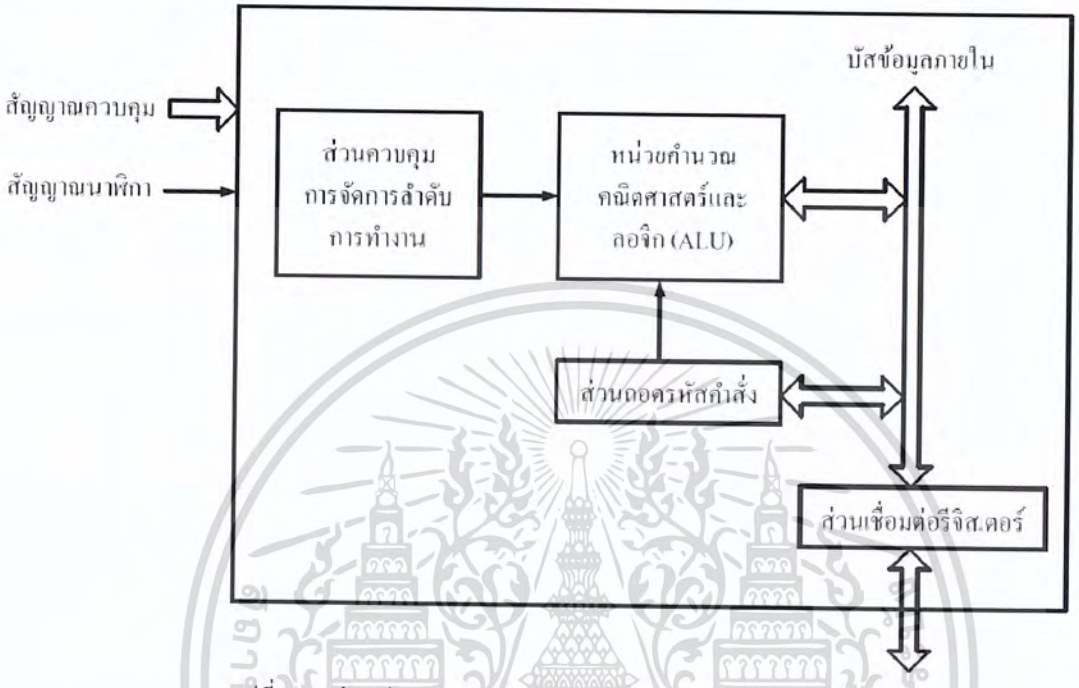
รูปที่ 2.39 โครงสร้างสถาปัตยกรรมของไมโครคอนโทรลเลอร์แบบฮาร์วาร์ด

2.4.1.2 หน่วยประมวลผลกลาง หรือซีพียู

ซีพียูเป็นเสมือนมันสมองของไมโครคอนโทรลเลอร์ ซึ่งทำหน้าที่ประมวลผลข้อมูลที่เข้ามาในระบบ แล้วทำการส่งต่อไปยังส่วนต่าง ๆ เพื่อควบคุมการทำงานต่อไป ในภาพที่ 2-15แสดงส่วนประกอบพื้นฐานของซีพียูในไมโครคอนโทรลเลอร์ทั่ว ๆ ไป จะเห็นได้ว่าหัวใจหลักของซีพียูคือหน่วยคำนวณทางคณิตศาสตร์และลอจิก (ALU: Arithmetic and Logic Unit)ซึ่งได้รับการกำหนดจังหวะการทำงานจากส่วนควบคุมลำดับการทำงาน โดยจังหวะการทำงานนั้นจะสัมพันธ์กับสัญญาณนาฬิกา เมื่อซีพียูทำการติดต่อหน่วยความจำ สิ่งที่ปรากฏขึ้นบนบัสข้อมูลภายในซีพียู คือ รหัสคำสั่ง (Instruction Code) ซึ่งต้องผ่านการทำงานของส่วนถอดรหัสคำสั่ง (Instruction Decoder) เสียก่อนจะได้เป็นข้อมูลคำสั่งที่ซีพียูเข้าใจ และสามารถดำเนินการต่อได้ หลังจากที่หน่วยคำนวณทางคณิตศาสตร์และลอจิกประมวลผลแล้วก็ส่งข้อมูลมายังส่วนเชื่อมต่อรีจิสเตอร์ ภายในซีพียู เพื่อที่จะติดต่อกับส่วนอื่น ๆ ต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทำงานของซีพียูมีด้วยกัน 2 จังหวะ เฟตช์ (Fetch) และเอ็กซิคิวต์ (Executed) โดยการทำงานจะเริ่มจากการเฟตช์ คือการเรียกหรือการเข้าถึงคำสั่ง แล้วทำการถอดรหัสเป็นภาษาเครื่องเพื่อเตรียมการประมวลผล จากนั้นจะเป็นจังหวะของการเอ็กซิคิวต์ คือการกระทำตามคำสั่งที่กำหนดให้จนเสร็จสิ้น



รูปที่ 2.40 ส่วนประกอบหลักของซีพียูในไมโครคอนโทรลเลอร์

2.4.1.3 หน่วยความจำ

ในไมโครคอนโทรลเลอร์จะประกอบด้วยหน่วยความจำ 3 แบบคือหน่วยความจำโปรแกรม (Program Memory), หน่วยความจำข้อมูลแรม (RAM Data Memory) และหน่วยความจำข้อมูลอีอีพรอม (EEPROM Data Memory)

- หน่วยความจำโปรแกรม

หน่วยความจำโปรแกรม เป็นที่สำหรับเก็บข้อมูลคำสั่งของโปรแกรมควบคุมที่ผู้พัฒนาเขียนขึ้นหรือเรียกว่า โปรแกรมมอนิเตอร์ (Monitor Program) ซีพียูจะเข้ามาติดต่อเพื่ออ่านข้อมูลรหัสคำสั่งจากหน่วยความจำในส่วนนี้ แล้วนำไปประมวลผลเพื่อควบคุมการทำงานของระบบทั้งหมดต่อไปหน่วยความจำโปรแกรมนี้มักมีขนาดใหญ่ ยิ่งมีขนาดมากเท่าใดก็จะสามารถบรรจุโปรแกรมที่มีความซับซ้อน หรือสามารถเก็บตารางข้อมูลที่ใช้ในการประมวลผลได้มากตามไปด้วย โดยทั่วไปมีความจุไม่น้อยกว่า 512 ไบต์ ขนาดของหน่วยความจำโปรแกรมจะแปรตามความก้าวหน้าทางเทคโนโลยี ชนิดของหน่วยความจำโปรแกรมที่ใช้ในไมโครคอนโทรลเลอร์ มีอยู่ 3 แบบที่นิยมกันคือแบบอีพรอม (EPROM: Erasable Programmable Read - Only Memory), แบบอีอีพรอม(EEPROM: Electrically Erasable Programmable Read - Only Memory) และแบบแฟลช

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

(Flash Memory) ความแตกต่างของทั้ง 3 แบบอยู่ที่จำนวนครั้งในการลบและเขียนข้อมูลทับลงไปใหม่ โดยสามารถสรุปได้ดังนี้

แบบอีพรอม แบ่งออกเป็น 2 แบบคือ แบบโปรแกรมได้หลายครั้ง และแบบโปรแกรมได้ครั้งเดียว ถ้าหากเป็นแบบโปรแกรมได้หลายครั้งบนตัวถังของไมโครคอนโทรลเลอร์ จะมีหน้าต่างกระจกติดอยู่สามารถมองเห็นชิปภายในได้ เวลาลบข้อมูลต้องลบด้วยแสงอุลตร้าไวโอเลต จำนวนรอบในการโปรแกรมใหม่อยู่ระหว่าง 10-100 ครั้ง แต่ถ้าเป็นแบบโปรแกรมได้ครั้งเดียว (One -Time Programmable, OTP) จะไม่สามารถลบได้ ตัวถังของมันจะปิดมิดชิดเหมือนกับไอซีธรรมดา

แบบอีอีพรอม หน่วยความจำแบบนี้จะลบและเขียนใหม่ได้ด้วยสัญญาณไฟฟ้า ในอดีตเป็นที่นิยมมากเนื่องจากสามารถเขียนใหม่ได้เป็นหลักร้อยรอบขึ้นไป ในบางตระกูลถึง 1 ล้านครั้งแต่ในปัจจุบันไม่เป็นที่นิยมใช้ในไมโครคอนโทรลเลอร์ เนื่องจากต้นทุนสูง

แบบแฟลช หน่วยความจำโปรแกรมชนิดนี้สามารถลบและเขียนได้ด้วยสัญญาณไฟฟ้าแตกต่างกับแบบอีอีพรอมในเชิงการใช้งานตรงที่กระบวนการลบข้อมูล หน่วยความจำโปรแกรมแบบแฟลชจะไม่สามารถเลือกลบเฉพาะเจาะจงบางตำแหน่งได้ เมื่อทำการลบข้อมูลจะต้องลบทั้งหมด หน่วยความจำแบบนี้ได้รับความนิยมมาก เนื่องจากราคาไม่สูง และสามารถโปรแกรมได้เป็นร้อยครั้งขึ้นไป ในบางรุ่นสูงเป็นหมื่นครั้ง และเป็นแสนครั้ง ขึ้นอยู่กับแรงดันที่ใช้ในการโปรแกรม

ขนาดข้อมูลของหน่วยความจำโปรแกรมขึ้นอยู่กับผู้ผลิตไมโครคอนโทรลเลอร์ ยกตัวอย่างในไมโครคอนโทรลเลอร์ตระกูล MCS-51, 68HC05, 68HC08, 68HC11, ขนาดของหน่วยความจำโปรแกรมคือ 8 บิต ถ้าเป็นตระกูล PIC จะเป็น 12 และ 14 บิต ถ้าเป็นตระกูล AVR, 68HC12 จะเป็นขนาด 16 บิต ทั้งนี้ขนาดของหน่วยความจำโปรแกรมไม่ได้เป็นตัวระบุความสามารถในการประมวลผลของไมโครคอนโทรลเลอร์

- หน่วยความจำข้อมูลแรม

เป็นหน่วยความจำที่ต้องมีในไมโครคอนโทรลเลอร์ทุกตัว เพราะใช้เป็นพื้นที่สำหรับเก็บข้อมูลทั้งในระหว่างและหลังการประมวลผล ยิ่งมีมากยิ่งช่วยในการทำงานสะดวกขึ้น เพราะหน่วยความจำแรมมีอัตราเร็วในการอ่านเขียนสูงมาก และไม่จำกัดจำนวนรอบในการอ่านเขียน ในพื้นที่ของหน่วยความจำข้อมูลแรมจะแบ่งออกเป็น 2 ส่วนคือ ส่วนของข้อมูลทั่วไปสำหรับเก็บค่าตัวแปร และส่วนของรีจิสเตอร์ โดยปกติแล้ว หน่วยความจำข้อมูลแรมจะมีความจุไม่มากเมื่อเทียบกับหน่วยความจุโปรแกรมในบางตัวอยู่ในหลักสิบไบต์ แต่ถ้าไมโครคอนโทรลเลอร์มีความสามารถสูงขึ้น ความจุของหน่วยความจำข้อมูลแรมก็เพิ่มมากขึ้นตาม ทั้งนี้เพราะต้องเพิ่มในส่วนของรีจิสเตอร์ตามความสามารถที่สูงขึ้นของไมโครคอนโทรลเลอร์

ขนาดของหน่วยความจำข้อมูลแรมโดยส่วนใหญ่จะมีขนาด 8 บิต แต่สามารถต่อร่วมกันเป็น 16 บิตได้ ส่วนการจัดสรรตำแหน่งแอดเดรสของหน่วยความจำข้อมูลแรมจะขึ้นอยู่กับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สถาปัตยกรรมของไมโครคอนโทรลเลอร์ หากเป็นแบบฮาร์วาร์ด (Harvard) จะได้รับการจัดสรรให้อยู่แยกจากหน่วยความจำโปรแกรม จึงทำให้มีค่าแอดเดรสเหมือนกัน ทั้งในหน่วยความจำโปรแกรมและหน่วยความจำข้อมูล แต่จริงๆ แล้วอยู่ต่างที่กัน ซึ่งจะพบในไมโครคอนโทรลเลอร์ MCS-51, PIC, AVR เป็นต้น แต่ถ้าเป็นแบบพรินซ์ตัน (Princeton) จะจัดสรรให้อยู่ในบริเวณเดียวกัน ดังนั้นค่าแอดเดรสจะไม่มีทางตรงกัน จะพบในไมโครคอนโทรลเลอร์ 68HCxxx ของ Motorola

- หน่วยความจำข้อมูลอีอีพรอม

เป็นหน่วยความจำข้อมูลพิเศษที่ในไมโครคอนโทรลเลอร์บางเบอร์ บางรุ่น บางตระกูลใช้สำหรับเก็บข้อมูลที่ต้องการรักษาไว้เมื่อไม่มีการจ่ายไฟเลี้ยงให้แก่ไมโครคอนโทรลเลอร์ การติดต่อเพื่ออ่านจะมีลักษณะพิเศษขึ้นอยู่กับไมโครคอนโทรลเลอร์แต่ละเบอร์ ขนาดของหน่วยความจำแบบนี้มักเท่ากับ 8 บิตส่วนความจุก็จะแตกต่างกันไปตั้งแต่ไม่กี่สิบไบต์จนถึงเป็นกิโลไบต์ การอ่านเขียนหน่วยความจำแบบนี้จะใช้สัญญาณไฟฟ้าทั้งหมด และสามารถรักษาข้อมูลล่าสุดไว้แม้ว่าจะไม่มีการจ่ายไฟเลี้ยงให้แก่ไมโครคอนโทรลเลอร์แล้วก็ตามสำหรับจำนวนรอบในการเขียนโดยปกติจะอยู่ในหลักล้านครั้งขึ้นไป

2.4.1.4 หน่วยความจำพิเศษ

เป็นหน่วยความจำที่ใช้เก็บข้อมูลพิเศษ ไม่ได้ใช้สำหรับเก็บชุดคำสั่งของโปรแกรม หรือข้อมูลอื่น ๆ มีหลายชนิดแตกต่างกันขึ้นอยู่กับตระกูลของไมโครคอนโทรลเลอร์

- รีจิสเตอร์

เป็นหน่วยความจำพิเศษที่มีบทบาทสูงมาก ในการทำงานของไมโครคอนโทรลเลอร์สามารถที่จะอ่านและเขียนข้อมูลได้ตลอดเวลาจนกว่าจะหยุดจ่ายไฟเลี้ยง ให้แก่ไมโครคอนโทรล-เลอร์ หน้าที่หลักก็คือใช้เก็บข้อมูลในการทำงานของไมโครคอนโทรลเลอร์ โดยข้อมูลที่เก็บนี้มีทั้งข้อมูลแสดงสถานะการทำงาน, ข้อมูลสำหรับควบคุมการทำงานโมดูลย่อยต่าง ๆ ภายในไมโครคอนโทรลเลอร์, ข้อมูลที่รับเข้ามาจากพอร์ตอินพุต, หรือข้อมูลที่ต้องส่งออกไปยังอุปกรณ์ภายนอกผ่านพอร์ตเอาต์พุต โดยข้อมูลแต่ละประเภทก็จะจัดเก็บลงในรีจิสเตอร์ที่แตกต่างกันตามหน้าที่การทำงานหน่วยความจำที่นำมาใช้ทำรีจิสเตอร์มีด้วยกัน 2 ลักษณะขึ้นอยู่กับสถาปัตยกรรมของไมโครคอนโทรลเลอร์ หากเป็นแบบพรินซ์ตัน รีจิสเตอร์จะมีอยู่ด้วยกัน 2 ส่วน ส่วนแรกจะอยู่ร่วมกันกับซีพียู หรือเรียกว่า รีจิสเตอร์ซีพียู ส่วนที่สองจะอยู่แยกต่างหาก ซึ่งมักเป็นรีจิสเตอร์ควบคุมพอร์ตอินพุตเอาต์พุตและรีจิสเตอร์แสดงสถานะ แต่ในสถาปัตยกรรมแบบฮาร์วาร์ด จะใช้บางส่วนในหน่วยความจำข้อมูลแรมภายในไมโครคอนโทรลเลอร์

- โปรแกรมเคาน์เตอร์

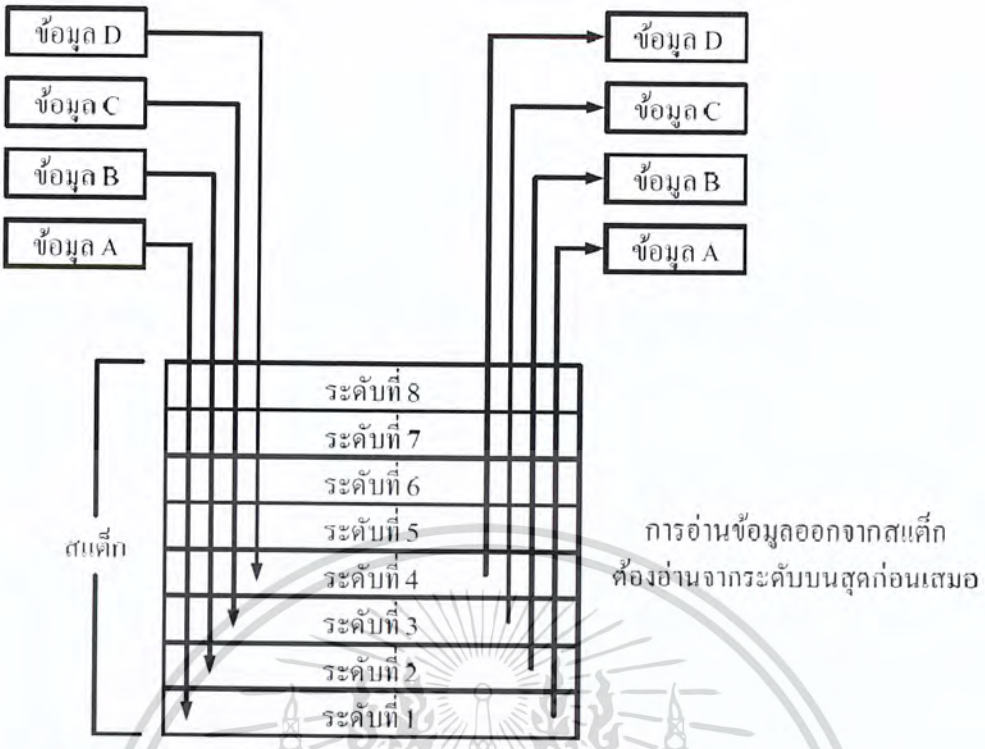
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การที่ซีพียูสามารถติดต่อกับหน่วยความจำโปรแกรม เพื่ออ่านข้อมูลคำสั่งได้อย่างถูกต้อง เป็นผลมาจากรีจิสเตอร์หน้าที่พิเศษตัวหนึ่งคือ รีจิสเตอร์ตัวนับโปรแกรม หรือโปรแกรมเคาน์เตอร์ (Program Counter, PC) โดยโปรแกรมเคาน์เตอร์จะเป็นตัวชี้ตำแหน่งแอดเดรสของหน่วยความจำโปรแกรมที่ซีพียูจะต้องไปกระทำในลำดับถัดไป โดยปกติแล้วค่าของโปรแกรมเคาน์เตอร์จะเปลี่ยนแปลงโดยอัตโนมัติขึ้นอยู่กับผลการทำ งานที่เกิดขึ้น ในไมโครคอนโทรลเลอร์บางตระกูลสามารถเข้าถึงโปรแกรมเคาน์เตอร์เพื่อทำการอ่านเขียนได้ แต่ในบางตระกูลก็ไม่สามารถทำได้

- สแต็ก

สแต็ก (Stack) เป็นหน่วยความจำพิเศษที่ไมโครคอนโทรลเลอร์ทุกตัวต้องมี โดยหน้าที่ของมันคือ เก็บข้อมูลที่ยังต้องการอยู่ของรีจิสเตอร์ และเมื่อข้อมูลนั้นถูกนำมาเก็บไว้ในสแต็กแล้วก็สามารถที่จะเปลี่ยนข้อมูลในรีจิสเตอร์ตัวนั้น ๆ ได้ทันที หลังจากที่ทำงานเรียบร้อยแล้วจึงกลับมาอ่านข้อมูลเดิมกลับคืนจากสแต็ก ซึ่งมีกระบวนการทำงานดังในภาพที่2-16 การเก็บข้อมูลของสแต็กจะมีลักษณะเป็นระดับหรือเป็นชั้น ข้อมูลที่เก็บเข้ามาก่อนจะต้องอ่านออกทีหลังหรือแบบ FILO (First In Last Out) และจำนวนระดับหรือจำนวนชั้นของสแต็กก็มีจำกัด ในไมโครคอนโทรลเลอร์ส่วนใหญ่จะมีความจุของสแต็กไม่น้อยกว่า 8 ระดับ การที่ยิ่งมีขนาดของสแต็กมากหรือมีจำนวนระดับมาก ก็จะยิ่งช่วยให้การทำงานสะดวกขึ้นเพราะ ในการประมวลผลมีโอกาสมากที่จะต้องพักข้อมูลในรีจิสเตอร์หลักเพื่อไปทำงานอื่นก่อน หลังจากนั้นจึงกลับมาทำงานต่อ โดยเฉพาะอย่างยิ่งกับงานที่มีการอินเตอร์รัพต์ หรือขัดจังหวะซีพียูอยู่บ่อย ๆรวมถึงการกระโดดไปทำงานที่โปรแกรมย่อย ที่มีความต้องการเขียนข้อมูลลงในรีจิสเตอร์ตัวเดียวกันนี้ หลังจากทำงานแล้วจึงกลับมาที่โปรแกรมหลัก แล้วอ่านค่าเดิมก่อนหน้านี้กลับมาทำงานต่อ ทว่างานบางลักษณะการกระโดดไปทำงานยังโปรแกรมย่อยซ้อนกัน 2-3 ชั้น ทำให้ต้องมีการเก็บข้อมูลไว้ในสแต็กมาขึ้น หากความจุของสแต็กมีน้อย ก็จะไม่สามารถรองรับการทำงานในลักษณะนี้ได้ ขนาดของสแต็กโดยปกติจะต้องเท่ากับขนาดของรีจิสเตอร์ตัวนับโปรแกรมหรือ PC เพราะมีโอกาสที่จะต้องเก็บค่าของ PC ไว้ในสแต็กด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.41 กลไกการทำงานของสแต็กอย่างง่าย

2.4.1.5 การทำงานของไมโครคอนโทรลเลอร์

ไมโครคอนโทรลเลอร์จะสามารถทำงานได้เมื่อจ่ายไฟเลี้ยง และต่อวงจรกำเนิดสัญญาณนาฬิกาให้แก่มัน จากนั้นซีพียูภายในไมโครคอนโทรลเลอร์จะติดต่อกับหน่วยความจำโปรแกรมเพื่ออ่านข้อมูลคำสั่งทำงานตามคำสั่งที่บรรจุอยู่ในหน่วยความจำโปรแกรมนั่นหมายความว่า ต้องมีการเขียนข้อมูลลงในหน่วยความจำโปรแกรมก่อน โดยไมโครคอนโทรลเลอร์แต่ละเบอร์จะมีรูปแบบของข้อมูลคำสั่งที่แตกต่างกัน ภาษาที่ใช้เขียนโปรแกรมสามารถแบ่งได้ 2 ระดับ คือภาษาระดับสูง (High Level Language) และภาษาแอสเซมบลี (Assembly Language) โดยปกติไมโครคอนโทรลเลอร์ต้องโปรแกรมด้วยภาษาแอสเซมบลี เนื่องจากสามารถทำงานได้รวดเร็วผ่านกระบวนการแปลงข้อมูลคำสั่งเป็นเลขฐานสิบหกเพื่อทำงานตามคำสั่งเพียง 1 ขั้นตอนคือแปลงจากภาษาแอสเซมบลีเป็นข้อมูลฐานสิบหกที่เรียกว่า ออปโค้ด (Opcode) แต่ข้อเสียของการเขียนภาษาแอสเซมบลีคือผู้เขียนต้องทำความเข้าใจในชุดคำสั่งของไมโครคอนโทรลเลอร์เบอร์นั้นๆอย่างถ่องแท้ และเมื่อเปลี่ยนเบอร์ไมโครคอนโทรลเลอร์ก็จะต้องทำการเรียนรู้ และทำความเข้าใจชุดคำสั่งใหม่ซึ่งอาจทำให้เสียเวลามาก รวมทั้งการเขียนโปรแกรมด้วยภาษาแอสเซมบลี ผู้เขียนต้องมีทักษะในการเขียนโปรแกรมสูงพอสมควร และเข้าใจถึงสถาปัตยกรรมของไมโครคอนโทรลเลอร์เป็นอย่างดี ในขณะที่การเขียนโปรแกรมด้วยภาษาระดับสูง อาทิเช่น ภาษาซี ภาษาเบสิก ต้องผ่านกระบวนการที่เรียกว่า คอมไพล์ (Compile) เพื่อแปลงภาษาระดับสูงเหล่านั้นเป็นภาษาเครื่องหรือออปโค้ดของไมโครคอนโทรลเลอร์เบอร์นั้น ๆ เสียก่อน เมื่อใช้เครื่องมือทางซอฟต์แวร์ตัวนี้ ทำให้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ผู้เขียนโปรแกรมอาจไม่จำเป็นต้องศึกษาสถาปัตยกรรม และชุดคำสั่งของไมโครคอนโทรลเลอร์เบอร์นั้น ๆ อย่างลึกซึ้งเท่ากับการเขียนภาษาแอสเซมบลี ทั้งนี้เพราะคอมไพเลอร์จะทำในส่วนนี้แทน ดังนั้นเมื่อผู้ใช้งานเปลี่ยนเบอร์ไมโครคอนโทรลเลอร์ ก็เพียงจัดหาโปรแกรมคอมไพเลอร์ที่เหมาะสมมาใช้งาน และศึกษาสถาปัตยกรรมของไมโครคอนโทรลเลอร์เบอร์ใหม่อีกเพียงเล็กน้อยก็สามารถใช้งานได้ แต่ข้อเสียของการใช้คอมไพเลอร์คือ ราคาแพงมาก

2.4.2 โครงสร้างและสถาปัตยกรรมของ PIC16F877A

ไมโครคอนโทรลเลอร์ได้มีการคิดค้นและพัฒนามาอย่างต่อเนื่อง เพื่อให้มีศักยภาพในการทำงานสูงขึ้น ไมโครคอนโทรลเลอร์ตระกูล PIC ของทางบริษัท Microchip เป็นไมโครคอนโทรล-เลอร์ที่มีฟังก์ชันการใช้งานต่างๆมากมาย เช่น โมดูล Analog to Digital, Timer/Counter, USART,SPI และอื่น ๆ ซึ่งส่วนต่างๆ เหล่านี้จะถูกสร้างรวมอยู่ภายในซีพียูเพียงตัวเดียว ทำให้สามารถทำงานได้หลาย ๆ อย่าง และสามารถลดในส่วนของฮาร์ดแวร์บางอย่างลงส่วนในเรื่องของความเร็วซีพียูตระกูลนี้จะใช้เวลาในการกระทำคำสั่งต่าง ๆ เพียง 1 หรือ 2 ไซเคิลต่อคำสั่งเท่านั้น โดยการทำงานนี้เป็นลักษณะไปป์ ไลน์ (Pipe Line) ทำให้มีความเร็วในการทำงานมากกว่าซีพียูทั่วไป (ที่ความถี่เดียวกัน) และมีสถาปัตยกรรมภายในดังภาพที่ 2-17 คุณสมบัติต่าง ๆ ของไมโครคอนโทรลเลอร์ PIC16F877A สามารถสรุปอย่างคร่าว ๆ ได้ดังนี้

- 35 ชุดคำสั่ง
- ในการปฏิบัติงานคำสั่งต่าง ๆ จะใช้ไซเคิลเดียวและ 2 ไซเคิลในคำสั่งที่เป็นการกระโดด
- ความถี่สูงสุดที่ทำงานได้คือ 20 MHz
- การทำงานจะเป็นลักษณะไปป์ ไลน์ทำให้มีการทำงานที่เร็วขึ้น
- หน่วยความจำโปรแกรมแบบแฟลชขนาด 8k (14-Bit Word)
- หน่วยความจำข้อมูลแบบแรม 368 ไบต์
- หน่วยความจำข้อมูลแบบอีอีพรอม 256 ไบต์
- สามารถตอบสนองการอินเตอร์รัพต์ได้ 14 แหล่ง
- สแต็ก 8 ระดับ
- เพาเวอร์ออนรีเซต (POR), เพาเวอร์อัพไทเมอร์ (PWRT) และ Oscillator Start-Up Time
- Watchdog Timer
- สามารถเลือกการป้องกันข้อมูลได้ (Code Protection)
- โหมดประหยัดพลังงาน (Sleep Mode)
- เลือกโหมดของ สัญญาณนาฬิกาได้หลายโหมด
- สามารถโปรแกรมโดยใช้แรงดัน +5 โวลต์ได้
- ฟังก์ชั่นการโปรแกรมแบบ ICSP (In – Circuit Serial Programming)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ทำงานที่ไฟเลี้ยง 2.0 ถึง 5.5 โวลต์
- Timer/Counter จำนวน 3 ตัวคือ Timer0, Timer1 และTimer2
- โมดูล Capture/Compare/PWM จำนวน 2 ชุด
- Analog to Digital Converter ความละเอียด 10 บิต 8 แชนแนล ภายในตัว
- มีโมดูลการสื่อสาร USART
- มีโมดูลตรวจจับระดับแรงดันไฟเลี้ยง Brow – Out Reset (BOR)
- มีพอร์ต I/O 5 พอร์ต ประกอบด้วย A, B, C, D, E แต่ละพอร์ตจะมีจำนวนบิตไม่เท่ากัน

PORTA = RA0 – RA5 จำนวน 6 บิต

PORTB = RB0 – RB7 จำนวน 8 บิต

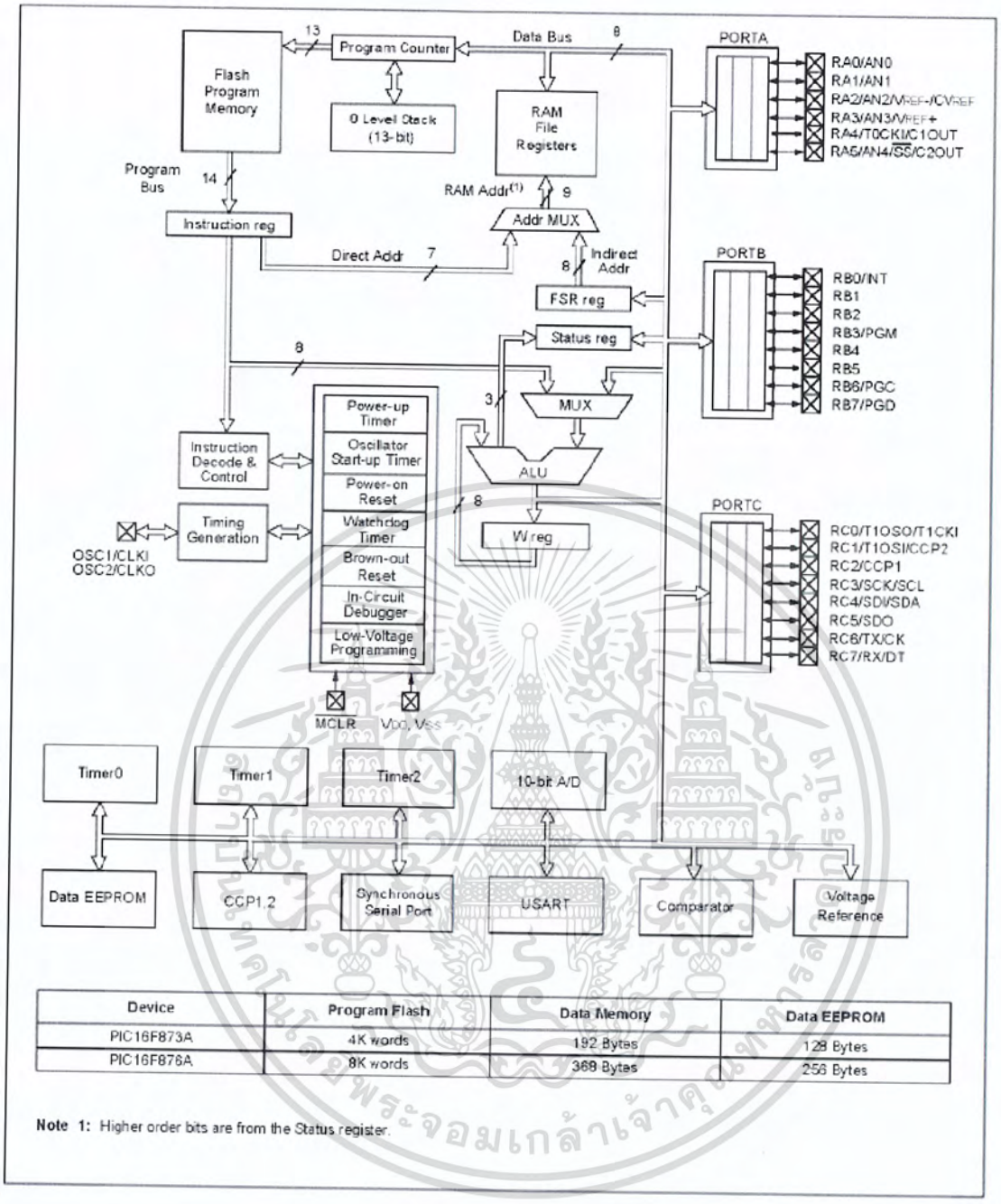
PORTC = RC0 – RC7 จำนวน 8 บิต

PORTD = RD0 – RD7 จำนวน 8 บิต

PORTE = RE0 – RE2 จำนวน 3 บิต

ขาสัญญาณของ PIC16F877A นี้จะมีทั้งหมด 40 ขาดังแสดงในภาพที่ 2-17 ประกอบไปด้วยขาที่ทำหน้าที่ต่าง ๆ โดยจะมีขาสัญญาณ I/O Ports ทั้งหมดจำนวน 33 ขา สามารถนำไปใช้เป็นอินพุต/เอาต์พุตได้ทั้งหมดทุกขา ยกเว้นขา RA4 ซึ่งโครงสร้างภายในเป็นแบบ Open Drain ดังนั้นหากต้องการนำไปใช้เป็นขาสัญญาณเอาต์พุต จะต้องต่อตัวต้านทานพูลอัพ (Pull – Up) ไว้ด้วย ส่วนขาที่เหลือสามารถใช้งานได้ตามปกติ นอกจากขาสัญญาณ I/O แล้ว ยังประกอบไปด้วยขาสัญญาณอื่น ๆ อีกคือ ขาไฟเลี้ยง, กราวนด์, ขารีเซ็ต และขาออสซิลเลเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

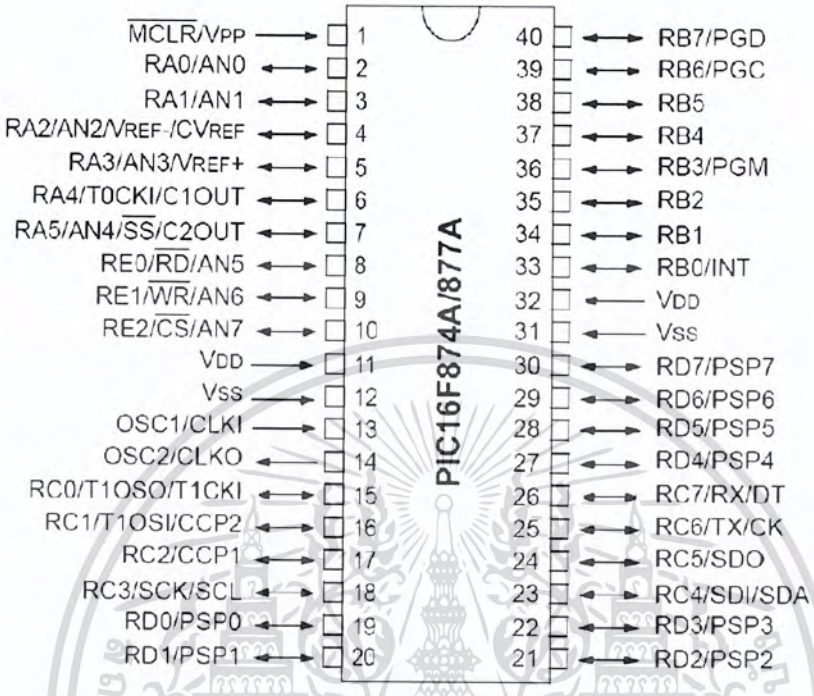


Device	Program Flash	Data Memory	Data EEPROM
PIC16F873A	4K words	192 Bytes	128 Bytes
PIC16F876A	8K words	368 Bytes	256 Bytes

Note 1: Higher order bits are from the Status register.

รูปที่ 2.42 สถาปัตยกรรมภายในของไมโครคอนโทรลเลอร์ PIC16F877A

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.43 ตัวถังของ PIC16F877A และตำแหน่งขาสัญญาณต่าง ๆ

หน้าที่การทำงานของขาสัญญาณต่าง ๆ สามารถสรุปได้ดังนี้

- ขา MCLR/VPP (ขา 1) ใช้ต่อสัญญาณรีเซ็ตแอคทีฟ 0 หรือใช้สำหรับป้อนแรงดันสำหรับการโปรแกรม

- ขา OSC1/CLKI (ขา 9) ใช้ต่อสัญญาณอินพุตสัญญาณนาฬิกาของซีพียู

- ขา OSC2/CLKO (ขา 14) เป็นขาเอาต์พุตของสัญญาณนาฬิกา (1/4 ของCLKI) และใช้ต่อร่วมกับขา OSC1 เพื่อกำเนิดสัญญาณนาฬิกา ในกรณีที่ใช้กับคริสตอล หรือวงจรเรโซเนเตอร์

- ขา VSS (ขา 12, 31) ใช้สำหรับต่อกราวนด์

- ขา VDD (ขา 11, 32) ใช้สำหรับต่อแรงดันไฟเลี้ยง

- ขา RA0 – RA5 (ขา 2 – 7) ขาสัญญาณอินพุต/เอาต์พุตพอร์ต A การกำหนดว่าเป็นอินพุตหรือเอาต์พุตขึ้นอยู่กับการกำหนดข้อมูลให้รีจิสเตอร์ TRISA ถ้าให้บิตใดเป็น 1 บิตนั้นจะเป็นอินพุต และถ้าให้เป็น 0 บิตนั้นจะเป็นเอาต์พุต

- ขา AN0 – AN7 (ขา 2 – 5, 7 – 10) ใช้สำหรับรับสัญญาณแอนะลอก

- ขา VREF– (ขา 4) ขาสัญญาณแรงดันอ้างอิงลบของ A/D

- ขา VREF+ (ขา 5) ขาสัญญาณแรงดันอ้างอิงบวกของ A/D

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ขา T0CKI (ขา 6) ใช้สำหรับป้อนอินพุตสัญญาณนาฬิกาของ Timer 0

- ขา SS (ขา 7) เป็นขาสัญญาณ Slave Select ในโหมด Synchronous Serial Port

- ขา RB0 – RB7 (ขา 33 – 40) ขาสัญญาณอินพุต/เอาต์พุตพอร์ต B การกำหนดว่าเป็นอินพุตหรือเอาต์พุตขึ้นอยู่กับการกำหนดข้อมูลให้รีจิสเตอร์ TRISB ถ้าให้บิตใดเป็น 1 บิตนั้นจะเป็นอินพุต และถ้าให้เป็น 0 บิตนั้นจะเป็นเอาต์พุต

- ขา INT (ขา 33) ใช้สำหรับรับสัญญาณอินเตอร์รัพต์

- ขา PGM (ขา 36) ใช้สำหรับการโปรแกรมแบบแรงดันต่ำ

- ขา PGC (ขา 39) ใช้เป็นขาสัญญาณนาฬิกาในโหมดการโปรแกรม

- ขา PGD (ขา 40) ใช้เป็นขาสัญญาณข้อมูลในการโปรแกรม

- ขา RC0 – RC7 (ขา 33, 16 – 18, 23 – 16) ขาสัญญาณอินพุต/เอาต์พุตพอร์ต C การกำหนดว่าเป็นอินพุตหรือเอาต์พุตขึ้นอยู่กับการกำหนดข้อมูลให้รีจิสเตอร์ TRISC ถ้าให้บิตใดเป็น 1 บิตนั้นจะเป็นอินพุต และถ้าให้เป็น 0 บิตนั้นจะเป็นเอาต์พุต

- ขา T1OSO/T1CKI (ขา 33) เป็นขาเอาต์พุตสัญญาณออสซิลเลเตอร์ และขาอินพุตสัญญาณนาฬิกาของ Timer1

- ขา T1OSI/CCP2 (ขา 16) เป็นขาอินพุตสัญญาณออสซิลเลเตอร์ของTimer1 และขาสัญญาณ Capture 2 input/Compare 2 output/PWM 2 output

- ขา CCP1 (ขา 17) ขาสัญญาณ Capture 1 input/Compare 1 output/PWM 1output

- ขา SCK/SCL (ขา 18) ใช้เป็นขาอินพุตสัญญาณนาฬิกาในการสื่อสารแบบSynchronous และขาสัญญาณนาฬิกาในโหมด I2C และ SPI

- ขา SDI/SDA (ขา 23) เป็นขาอินพุตสัญญาณข้อมูลในโหมด SPI และขาอินพุต/เอาต์พุตสัญญาณข้อมูลในโหมด I2C

- ขา SDO (ขา 24) ขาเอาต์พุตสัญญาณข้อมูลในโหมด SDI

- ขา TX/CX (ขา 25) ขาเอาต์พุตสัญญาณด้านส่งในการสื่อสารแบบ USART และขาสัญญาณนาฬิกาในโหมด การสื่อสารแบบ Synchronous

- ขา RX/DT (ขา 26) ขาอินพุตสัญญาณด้านรับของการสื่อสารแบบUSART และขาสัญญาณข้อมูลในการสื่อสารแบบ Synchronous

- ขา RD0/PSP0 – RD7/PSP7 (ขา 19 – 22, 27 – 30) ขาสัญญาณอินพุต/เอาต์พุตพอร์ต D การกำหนดว่าเป็นอินพุตหรือเอาต์พุตขึ้นอยู่กับการกำหนดข้อมูลให้รีจิสเตอร์ TRISD ถ้าให้บิตใดเป็น 1 บิตนั้นจะเป็นอินพุตและถ้าเป็น 0 บิตนั้นจะเป็นเอาต์พุต และสามารถใช้เป็น Slave Port กรณีติดต่อกับระบบบัสของไมโครโปรเซสเซอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ขา RE0 – RE2 (ขา 8 – 10) ขาสัญญาณอินพุต/เอาต์พุตพอร์ต E การกำหนดว่าเป็นอินพุตหรือเอาต์พุตขึ้นอยู่กับการกำหนดข้อมูลให้รีจิสเตอร์ TRISE บิตที่0 ถึง 2 ถ้าให้บิตใดเป็น 1 บิตนั้นจะเป็นอินพุต และถ้าเป็น 0 บิตนั้นจะเป็นเอาต์พุต

- ขา RD (ขา 8) ใช้เป็นขาสัญญาณควบคุมการอ่านในโหมด Parallel Slave Port

- ขา WR (ขา 9) ใช้เป็นขาสัญญาณควบคุมการเขียนในโหมด Parallel Slave Port

- ขา CS (ขา 10) ใช้เป็นขาสัญญาณควบคุม ช่องip Select ในโหมด Parallel Slave Port

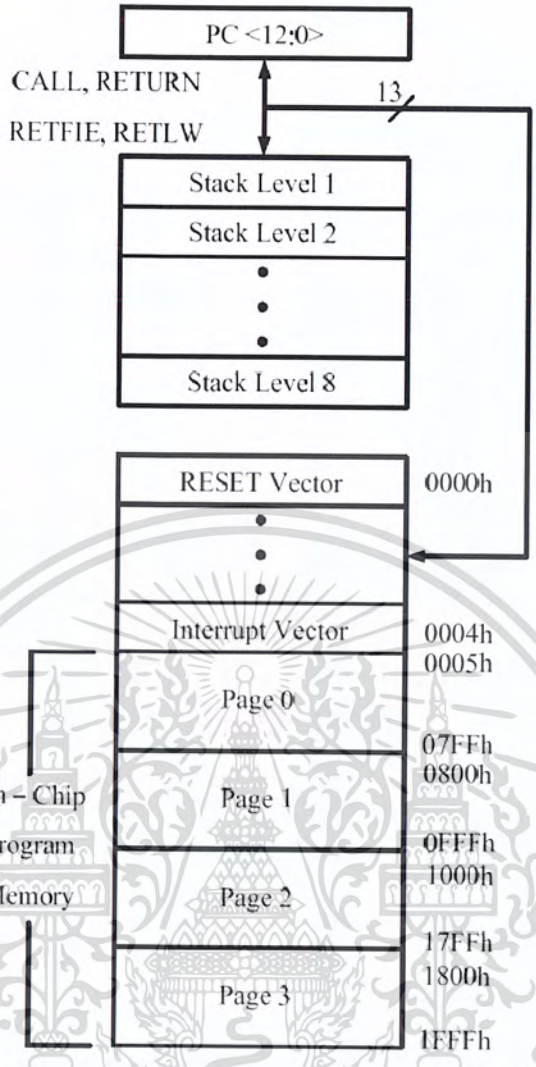
2.4.3 หน่วยความจำภายในไมโครคอนโทรลเลอร์ PIC16F877A

หน่วยความจำ เป็นส่วนประกอบที่สำคัญสำหรับไมโครคอนโทรลเลอร์ เป็นพื้นที่สำหรับเก็บออปโค้ดโปรแกรมและข้อมูลอื่น ๆ ไมโครคอนโทรลเลอร์ตระกูล PIC16F87X จะมีหน่วยความจำภายใน แบ่งออกเป็น 3 ส่วนด้วยกันคือ หน่วยความจำโปรแกรม, หน่วยความจำข้อมูล และหน่วยความจำข้อมูลอีอีพรอม

2.4.3.1 หน่วยความจำโปรแกรม

หน่วยความจำโปรแกรมเป็นพื้นที่สำหรับใช้ในการเก็บซอร์สโค้ดโปรแกรม โดยจะเป็นหน่วยความจำแบบแฟลช จึงสามารถทำการเขียนและลบได้หลายครั้ง ทำให้สะดวกต่อการทดลองพัฒนาโปรแกรม โดยมีโปรแกรมเคาน์เตอร์ขนาด 13 บิต ซึ่งสามารถอ้างอิงตำแหน่งข้อมูลถึง 8 กิโลเวิร์ดตั้งแต่แอดเดรส 0000h ถึง 1FFFh ดังภาพที่ 2-20 หน่วยความจำโปรแกรม ของ PIC16F877A นี้จะมีขนาด 8 k×14 บิตเวิร์ด (8 กิโลเวิร์ด) แบ่งออกเป็น 4 Page จำนวนPage ละ 2 กิโลเวิร์ด ดังจะเห็นได้จากภาพที่ 2-20 จะมีรีเซตเวกเตอร์อยู่ที่ตำแหน่ง 0000h และอินเตอร์รัพท์เวกเตอร์อยู่ที่ 0004h จะเห็นได้ว่ามีแอดเดรสเวกเตอร์ของการอินเตอร์รัพท์อยู่ที่ตำแหน่งเดียว แต่ไมโครคอนโทรลเลอร์PIC16F87X นั้นมีแหล่งกำเนิดสัญญาณอินเตอร์รัพท์ได้ถึง 14 แหล่ง ซึ่งเมื่อเกิดอินเตอร์รัพต์จากแหล่งใดก็ตาม ซีพียูจะกระโดดไปทำงานในตำแหน่งแอดเดรสเดียวกันนั้น คือแอดเดรส 0004hดังนั้นเราจึงไม่สามารถลำดับความสำคัญของการอินเตอร์รัพท์ได้ ส่วนหน่วยความจำสแต็กจะมีขนาดความลึก 8 ระดับและไม่สามารถเข้าถึงได้โดยตรงจากการเขียนโปรแกรมไม่มีคำสั่ง PUSH –POP เหมือนกับซีพียูตระกูลอื่น ๆ ค่าของสแต็กจะมีการเปลี่ยนแปลงอัตโนมัติ เมื่อมีการใช้คำสั่งที่เป็นการกระโดด, การเรียกใช้โปรแกรมย่อย หรือ เมื่อมีการอินเตอร์รัพต์ขึ้นเท่านั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

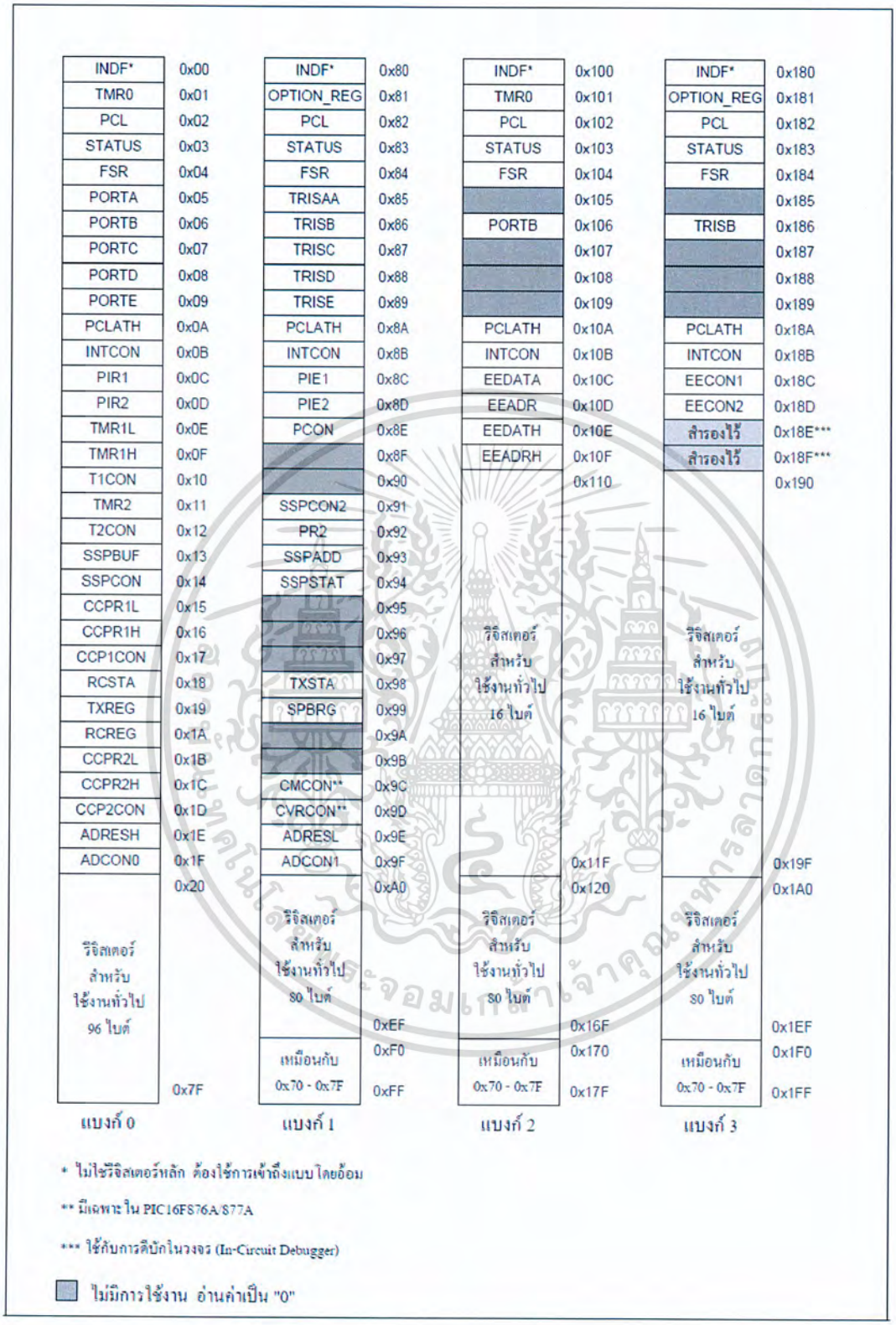


รูปที่ 2.44 การวางพื้นที่ของหน่วยความจำโปรแกรมของ PIC16F877A

2.4.3.2 หน่วยความจำข้อมูล

หน่วยความจำข้อมูลนี้ จะประกอบไปด้วยพื้นที่ของรีจิสเตอร์ใช้งานทั่วไป (General Purpose Register) ขนาด 368 ไบต์ และพื้นที่ของรีจิสเตอร์ฟังก์ชั่นพิเศษ (Special Function Register) ซึ่งพื้นที่ของหน่วยความจำเหล่านี้จะถูกแบ่งออกเป็น 4 แบงก์ดังในภาพที่ 2-20 การเข้าถึงข้อมูลในแต่ละส่วนต้องกำหนด แบงก์ข้อมูลที่เราต้องการเข้าถึง โดยการกำหนดค่าในรีจิสเตอร์ที่ทำหน้าที่เลือกแบงก์ คือ RP0 และ RP1 โดยจะอยู่ในรีจิสเตอร์ STATUS บิตที่ 5 และ ตามลำดับซึ่งค่าต่าง ๆสามารถกำหนดได้ตามตารางที่ 2-9

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.45 การจัดสรรพื้นที่หน่วยความจำข้อมูลของ PIC16F877A

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 1 การเลือกแบงก์ข้อมูลของหน่วยความจำ

RP1:RP0	Bank
00	0
01	1
10	2
11	3

- รีจิสเตอร์ใช้งานทั่วไป

รีจิสเตอร์ใช้งานทั่วไป (General Purpose Register) เป็นหน่วยความจำใช้งานทั่วไปโดยโครงสร้างจะเป็นหน่วยความจำชนิด Static RAM สามารถเปลี่ยนแปลงข้อมูลได้ตลอดเวลาและข้อมูลต่างๆ จะหายไปเมื่อไม่มีไฟเลี้ยงให้กับ ซีพียู หน่วยความจำ ซีพียู PIC16F87Xจะมีขนาด368 ไบต์ ข้อมูลจะเป็นแบบ 8 บิต เหมาะสำหรับการใช้เก็บข้อมูล หรือใช้เป็นตัวแปลต่างๆ ในการเขียนโปรแกรมโดยจะกระจายอยู่ในแบงก์ต่างๆ ทั้ง 4 แบงก์

- รีจิสเตอร์ฟังก์ชันพิเศษ

รีจิสเตอร์ฟังก์ชั่นพิเศษ (Special Purpose Register) เป็นรีจิสเตอร์ที่ใช้ในการควบคุมการทำงานในฟังก์ชั่นต่าง ๆ ของ CPU โดยจะจัดวางลักษณะเดียวกันกับรีจิสเตอร์ใช้งานทั่วไป คือจะอยู่ภายในแต่ละแบงก์ทั้ง 4 แบงก์ เช่นรีจิสเตอร์แสดงสถานะ (STATUS Register), รีจิสเตอร์การอินเตอร์รัพต์ (INTCON Register), รีจิสเตอร์แสดงการทำงานของฟังก์ชันต่าง ๆ (OPTION_REG Register) เป็นต้น

2.4.3.3 หน่วยความจำข้อมูลอีอีพรอม

PIC16F877A มีหน่วยความจำอีอีพรอมจำนวน 256 ไบต์ โดยสามารถอ่านและเขียนในขณะทำงานปกติได้ แต่ต้องไม่มีการ Enable Code Protect Bit โดยการเข้าถึงนั้นจะต้องทำผ่านรีจิสเตอร์พิเศษ 4 ตัวคือ EECON1 ทำหน้าที่ควบคุมการเข้าถึงหน่วยความจำ, EECON2 ทำหน้าที่จัดลำดับการเขียนข้อมูล, EEDATA เป็นบัฟเฟอร์ใช้เก็บข้อมูล 8 บิต สำหรับการอ่านและเขียน และ EEADR เป็นรีจิสเตอร์ที่เก็บแอดเดรส 00h – FFh (256 ไบต์) ข้อมูลที่ถูกเขียนลงในหน่วยความจำแบบอีอีพรอมจะคงสถานะเดิมอยู่ตลอด แม้จะไม่มีการจ่ายไฟเลี้ยงให้กับซีพียูแล้วก็ตาม ซึ่งในการอ่านและเขียนข้อมูลของอีอีพรอมนี้ ไม่สามารถใช้คำสั่งโอนย้ายข้อมูลแบบปกติเหมือนที่ใช้กับหน่วยความจำประเภทแรม แต่จะต้องใช้กระบวนการพิเศษผ่านรีจิสเตอร์พิเศษทั้ง 4 ตัว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การอ่านข้อมูลจากหน่วยความจำอีอีพรอมมีขั้นตอนดังนี้

1. กำหนดแอดเดรสที่ต้องการอ่านลงในรีจิสเตอร์ EEADR ซึ่งค่าที่ใส่จะอยู่ในช่วง 00h – FFh

2. เคลียร์บิต EEPGD ที่อยู่ในรีจิสเตอร์ EECON1 บิตที่ 7 เพื่อเลือกการติดต่อกับหน่วยความจำอีอีพรอม

3. เซ็ตบิต RD เป็น 1 เพื่อเริ่มต้นการอ่านข้อมูล

4. ข้อมูลที่อ่านออกมาได้จะอยู่ในรีจิสเตอร์ EEDATA การเขียนข้อมูลลงหน่วยความจำอีอีพรอมจะยุ่งยากกว่าการอ่านข้อมูล ซึ่งมีลำดับขั้นตอนต่าง ๆ ดังนี้

4.1 กำหนดตำแหน่งแอดเดรสที่ต้องการเขียนข้อมูล ในรีจิสเตอร์ EEADRโดยจะมีค่าอยู่ในช่วง 00h – FFh

4.2 กำหนดข้อมูล 8 บิตที่ต้องการเขียนลงในรีจิสเตอร์ EEDATA

4.3 เคลียร์บิต EEPGD เป็น 0 เพื่อกำหนดให้เป็นการติดต่อกับหน่วยความจำอีอีพรอม

4.4 เซ็ตบิต WREN เพื่อ Enable การเขียนข้อมูล

4.5 ทำการ Disable การอินเตอร์รัพต์ทั้งหมด (หากมีการใช้งานอินเตอร์รัพต์อยู่) เพื่อไม่ให้การอินเตอร์รัพต์มาขัดจังหวะในขณะที่มีการเขียนข้อมูลลงอีอีพรอม

4.6 ทำขั้นตอนพิเศษดังต่อไปนี้ ซึ่งต้องทำทุกครั้งที่มีการเขียนข้อมูลลงอีอีพรอม และต้องระวังไม่ให้มีการอินเตอร์รัพต์เกิดขึ้นในระหว่างนี้

- เขียนข้อมูล 55h ลงในรีจิสเตอร์ EECON2 ขั้นตอนนี้มี 2 เสต็ปคือ โหลดข้อมูล 55h ให้รีจิสเตอร์ W และโหลดข้อมูลจากรีจิสเตอร์ Wไปยังรีจิสเตอร์ EECON2

- เขียนข้อมูล AAh ลงในรีจิสเตอร์ EECON2 ขั้นตอนนี้มี 2 เสต็ป คือ โหลดข้อมูล AAh ให้รีจิสเตอร์ W และโหลดข้อมูลจากรีจิสเตอร์ W ไปยังรีจิสเตอร์ EECON2

- เซ็ตบิต WR เป็น 1 โดยบิตนี้จะกลายเป็น 0 เองเมื่อการเขียนข้อมูลเสร็จสิ้น

4.7 ทำการ Enable การอินเตอร์รัพต์ เพื่ออนุญาตให้มีการอินเตอร์รัพต์ (เฉพาะในกรณีที่มีการใช้งานอินเตอร์รัพต์อยู่ก่อนแล้วเท่านั้น) หากไม่ได้ใช้งานอินเตอร์รัพต์ก็ควรจะ Disable การอินเตอร์รัพต์ไว้

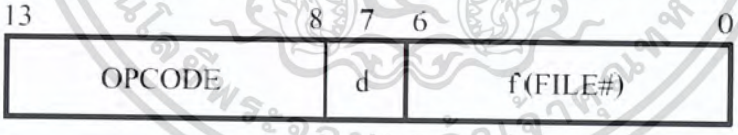
4.8 เคลียร์บิต WREN เป็น 0 เพื่อ Disable การเขียนข้อมูลของอีอีพรอม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.9 เมื่อกระบวนการเขียนข้อมูลเสร็จสิ้นบิต WR จะถูกเคลียร์โดยอัตโนมัติและแฟลกบิต EEIF ซึ่งเป็นแฟลกแสดงสถานการณ์อินเตอร์รัพต์ของอีอีพรอมจะเซ็ตเป็น 1 และถ้าหากมีการ Enableอินเตอร์รัพต์ชนิดนี้ไว้ก็จะมีการอินเตอร์รัพต์เกิดขึ้น

2.4.4 ชุดคำสั่งในไมโครคอนโทรลเลอร์ PIC16F877A

ชุดคำสั่งในภาษาแอสเซมบลีของ PIC16F877A จะมีอยู่ทั้งหมด 35 คำสั่ง ซึ่งมีโครงสร้างเป็นแบบ 14 บิต โดยโครงสร้างของคำสั่งจะแบ่งออกเป็น 3 ลักษณะคือ Byte – oriented operation เป็นโครงสร้างของคำสั่งในการประมวลผลข้อมูลระดับไบต์, Bit – oriented operation เป็นโครงสร้างของคำสั่งในการประมวลผลข้อมูลระดับบิต และ Literal and Control operation เป็นโครงสร้างของคำสั่งในการประมวลผลข้อมูลกับค่าคงที่ k (literal) โครงสร้างของคำสั่งแบบไบต์นี้ดังแสดงในภาพที่ 2-22 จะใช้ตัวอักษร f เป็นตัวแปรแทนตำแหน่งของรีจิสเตอร์ต่าง ๆ ซึ่งจะมีขนาด 7 บิต ดังนั้นจึงสามารถอ้างถึงตำแหน่งของรีจิสเตอร์ได้ตั้งแต่ 00h – 7Fh ส่วนตัวอักษร d (Destination) จะเป็นค่าของตำแหน่งปลายทางของผลลัพธ์จากการประมวลผล โดยหากตัวอักษร d แทนด้วย 1 จะหมายถึงการนำผลลัพธ์ที่ได้จากการประมวลผลคำสั่งเก็บไว้ที่รีจิสเตอร์ไฟล์ (f) และหากตัวอักษร d เป็น 0 จะหมายถึงการนำผลลัพธ์ที่ได้เก็บไว้ในรีจิสเตอร์ W แต่ในการใช้งานจริงสามารถกำหนดเป็นตัวอักษร W หรือ F แทนก็ได้เพื่อให้เข้าใจได้ง่ายขึ้น โดยถ้าแทน d ด้วย W หมายถึงนำผลลัพธ์เก็บที่รีจิสเตอร์ W และถ้า d แทนด้วย F ผลลัพธ์จะถูกเก็บในรีจิสเตอร์ไฟล์คำสั่งที่มีโครงสร้างแบบนี้ อยู่ในตารางที่ 2-10



d = 0 for destination W

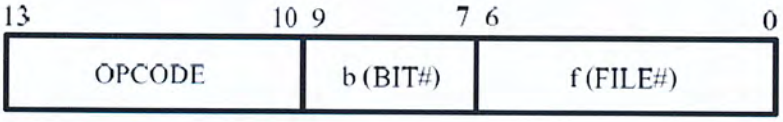
d = 1 for destination f

F = 7-bit file register address

รูปที่ 2.46 โครงสร้างของคำสั่งแบบไบต์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำหรับโครงสร้างของคำสั่งแบบบิต ตัวอักษร b จะแทนตำแหน่งของบิตข้อมูลในรีจิสเตอร์ไฟล์ (f) ซึ่งจะใช้ในการอ้างตำแหน่งแอดเดรสของบิตข้อมูลในรีจิสเตอร์ต่าง ๆ โดยโครงสร้างของ b จะมีขนาด 3 บิต ดังนั้นจึงสามารถอ้างถึงตำแหน่งของบิตข้อมูลได้ 8 ตำแหน่ง“000 – 111” (บิต 0 – บิต 7) ดังในภาพที่ 2-23 คำสั่งที่มีโครงสร้างแบบนี้อยู่ในตารางที่ 2-8



b = 3-bit address

f = 7-bit file register address

รูปที่ 2.47 โครงสร้างของคำสั่งแบบบิต

ตารางที่ 2 คำสั่งที่มีโครงสร้างแบบไบต์

รหัสนีโมนิค		คำอธิบาย	ออปโค้ด 14 บิต			
			MSB			LSB
ADDWF	f, d	Add W and f	00	0111	dfff	ffff
ANDWF	f, d	AND W with f	00	0101	dfff	ffff
CLRF	f	Clear f	00	0001	1fff	ffff
CLRW		Clear W	00	0001	0xxx	xxxx
COMF	f, d	Component f	00	1001	dfff	ffff
DECF	f, d	Decrement f	00	0011	dfff	ffff
DECFSZ	f, d	Decrement f, Skip if 0	00	1011	dfff	ffff
INCF	f, d	Increment f	00	1010	dfff	ffff
INCFSZ	f, d	Increment f, Skip if 0	00	1111	dfff	ffff
IORWF	f, d	Inclusive OR W with f	00	0100	dfff	ffff
MOVF	f, d	Move f	00	1000	dfff	ffff
MOVWF	f	Move W to f	00	0000	1fff	ffff

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2-10 คำสั่งที่มีโครงสร้างแบบไบต์ (ต่อ)

รหัสนีโมนิค		คำอธิบาย	ออปโค้ด 14 บิต			
			MSB			LSB
NOP		No Operation	00	0000	0xx0	0000
RLF	f, d	Rotate Left f through Carry	00	1101	dfff	ffff
RRF	f, d	Rotate Right f through Carry	00	1100	dfff	ffff
SUBWF	f, d	Subtract W from f	00	0010	dfff	ffff
SWAPF	f, d	Swap nibbles in f	00	1110	dfff	ffff
XORWF	f, d	Exclusive OR W with F	00	0110	dfff	ffff

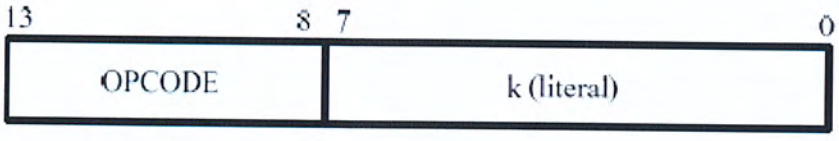
ตารางที่ 2-11 คำสั่งที่มีโครงสร้างแบบบิต

รหัสนีโมนิค		คำอธิบาย	ออปโค้ด 14 บิต			
			MSB			LSB
BCF	f, d	Bit Clear f	01	00bb	bfff	ffff
BSF	f, d	Bit Set f	01	01bb	bfff	ffff
BTFSC	f, d	Bit Test f, Skip if Clear	01	10bb	bfff	ffff
BTFSS	f, d	Bit Test f, Skip if Set	01	11bb	bfff	ffff

ส่วนโครงสร้างของคำสั่งแบบ Literal and Control นั้นค่า k จะใช้แทนค่าคงที่ ซึ่งมีทั้งแบบ 8 บิต และแบบ 11 บิต ดังแสดงในภาพที่ 2- 24 ส่วนคำสั่งอยู่ในตารางที่ 2-10

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Genaral



k = 8-bit immediate value

CALL and GOTO instruction only

k = 11-bit immediate value

รูปที่ 2.48 โครงสร้างของคำสั่งแบบ Literal and Control

ตารางที่ 3 คำสั่งที่มีโครงสร้างแบบ Literal and Control

รหัสมีโมนิค		คำอธิบาย	ออปโค้ด 14 บิต			
			MSB			LSB
ADDLW	k	Add literal and W	11	111x	kkkk	kkkk
ANDLW	k	AND literal with W	11	1001	kkkk	kkkk
CALL	k	Call subroutine	10	0kkk	kkkk	kkkk
CLRWDT		Clear Watchdog Timer	00	0000	0110	0100
GOTO	k	Go to address	10	1kkk	kkkk	kkkk
IORLW	k	Inclusive OR literal with W	11	1000	kkkk	kkkk
MOVLW	k	Move literal to W	11	00xx	kkkk	kkkk
RETFIE		Return from interrupt	00	0000	0000	1001
RETLW	k	return with literal in W	11	01xx	kkkk	kkkk

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รหัสนีโมนิค		คำอธิบาย	ออปโค้ด 14 บิต			
			MSB			LSB
RETURN		Return from Subroutine	00	0000	0000	1000
SLEEP		Go into standby mode	00	0000	0110	0011
SUBLW	k	Subtract W from literal	11	110x	kkkk	kkkk
XORLW	k	Exclusive OR literal with W	11	1010	kkkk	kkkk

2.5 LCD (Liquid Crystal Display)

อุปกรณ์สำหรับแสดงผลในปัจจุบันมีหลายแบบด้วยกัน LCD ก็เป็นอุปกรณ์แสดงผลประเภทหนึ่งที่ได้รับความนิยมอย่างแพร่หลาย ในส่วนนี้เป็นการยกตัวอย่างการใช้งาน FPGA เพื่อควบคุมจอแสดงผลLCD แบบ 16 ตัวอักษร 1 แถว

สำหรับบทความนี้จะแบ่งออกเป็น 2 ตอน ซึ่งตอนแรกจะอธิบายถึงหลักการทั่วไปและคำสั่งที่ใช้ควบคุมโมดูล LCD และในตอนที่ 2 จะเป็นการเชื่อมต่อบอร์ด WIZARD FLEX-A01 เข้ากับโมดูล LCD และ การเขียนภาษา VHDL สำหรับควบคุมการแสดงผลของโมดูล LCD

ส่วนประกอบหลักในโมดูล LCD

1.ตัวควบคุม (Controller) เป็นอุปกรณ์สำหรับรับข้อมูลที่จะส่งมาจากอุปกรณ์ที่อยู่ภายนอกเพื่อควบคุมการทำงานภายในโมดูล LCD เช่นการลบจอภาพ , การแสดงตัวอักษร หรือการเลื่อนเคอร์เซอร์ เป็นต้น

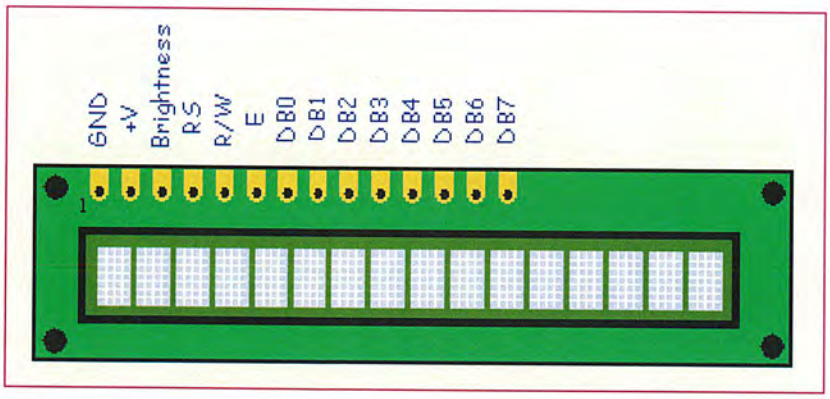
2.ตัวขับ (Driver) เป็นอุปกรณ์รับข้อมูลจากตัวควบคุม (Controller) เพื่อขับให้ตัวแสดงผลแสดงข้อมูลตามที่กำหนด

3.ตัวแสดงผล (Dot Matrix Display) เป็นอุปกรณ์แสดงผลให้สามารถมองเห็นเป็นตัวอักษรหรืออักขระ ซึ่งภายในชุดแสดงผลจะเป็นผลึกเหลวที่สามารถแสดงผลให้เห็นโดยอาศัยการเปิดและปิดตัวเองกับแสงจากภายนอก

โมดูล LCD แบบ 16 ตัวอักษร 1 บรรทัด

สำหรับ LCD ที่อยู่บนบอร์ด WIZARD FLEX-A01 จะเป็น LCD แบบ 16 ตัวอักษร 1 บรรทัด จะมีขาให้ต่อใช้งานทั้งหมด 14 ขา สามารถแสดงตำแหน่งการจัดขาของโมดูล LCD ได้ดังรูปที่ 3.49

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.49 LCD แบบ 16 ตัวอักษร 1 แถวและการจัดตำแหน่งขา

โดยแต่ละขาจะมีหน้าที่การทำงานที่แตกต่างกันดังนี้

ขาที่ 1 (Vss) : กราวด์

ขาที่ 2 (Vdd) : ไฟเลี้ยง +5V

ขาที่ 3 (Vo) : ขาอินพุทสำหรับรับแรงดันจากภายนอกเพื่อปรับความเข้มของการแสดงผล

ขาที่ 4 (RS) : ขาอินพุทใช้สำหรับบอกตัวควบคุมภายใน LCD ว่าในขณะนั้นเป็นข้อมูลสำหรับแสดงผลหรือเป็นคำสั่งสำหรับเซตค่าต่างๆภายใน LCD โดยที่หากกำหนดให้ RS เป็นลอจิก "0" ข้อมูลที่ส่งมานั้นจะเป็นคำสั่งและถ้ากำหนดให้ RS เป็นลอจิก "1" ข้อมูลที่ส่งมาจะเป็นข้อมูลสำหรับการแสดงผล

ขาที่ 5 (R/W) : ขาอินพุทสำหรับเลือกว่าจะทำการอ่านหรือเขียนข้อมูลแก่ LCD หากกำหนดให้ขา R/W เป็นลอจิก "0" จะเป็นการเขียนข้อมูลให้แก่ LCD และหากกำหนดกำหนดให้ขา R/W เป็นลอจิก "1" จะเป็นการอ่านข้อมูลจาก LCD

ขาที่ 6 (E) : ขาอินพุทสำหรับแอนนาเบิลให้คอนโทรลเลอร์ใน LCD ทำงานตามข้อมูลที่ส่งไป

ขาที่ 7 ถึงขาที่ 14 (DB0-DB7) : ขาข้อมูลขนาด 8 บิตใช้สำหรับส่งผ่านข้อมูลระหว่าง LCD กับอุปกรณ์ภายนอก

คำสั่งควบคุมโมดูล LCD

ในการเขียนคำสั่งให้แก่โมดูล LCD จะต้องกำหนดให้สัญญาณ R และ R/W เป็นลอจิก "0" จากนั้นจึงส่งคำสั่งออกไปที่บัสข้อมูลและทำการหน่วงเวลาเพื่อให้ตัวคอนโทรลใน LCD ตอบสนองกับคำสั่งที่จะทำการส่งไป หลังจากนั้นจึงกำหนดให้สัญญาณ E เป็นลอจิก "1" เพื่อแอนนาเบิลให้ LCD ตอบสนองกับคำสั่งและคำสั่งที่ใช้ควบคุม LCD ที่สำคัญมีดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.คำสั่งเคลียร์ตัวแสดงผล (Clear Display)

เป็นคำสั่งใช้สำหรับเขียนข้อมูลช่องว่างหรือ SPACE (ASCII A0H) เข้าไปใน DDRAM ทั้งหมดและเมื่อตัวควบคุมทำคำสั่งนี้ ตัวควบคุมจะทำการกำหนดแอดแดรสของ DDRAM เป็น 0 และเคอร์เซอร์จะกลับไปอยู่ตำแหน่งบนสุดซ้ายมือของจอแสดงผลจะมีข้อมูลคำสั่งเป็น 01H โดยมีรูปแบบคำสั่งดังนี้

2.คำสั่ง Return Home

เป็นคำสั่งสำหรับให้เคอร์เซอร์เคลื่อนที่กลับไปยังตำแหน่งซ้ายสุดบนจอแสดงผล โดยที่ข้อมูลในจอภาพไม่เปลี่ยนแปลง ข้อมูลคำสั่งจะเป็น 02H หรือ 03H ก็ได้ โดยมีรูปแบบคำสั่งดังนี้

3.คำสั่งเลือกโหมดการป้อนข้อมูล (Entry Mode Set)

จะเป็นคำสั่งการเลือกโหมดการป้อนข้อมูล ซึ่งข้อมูลคำสั่งสำหรับการเลือกโหมดการป้อนข้อมูลจะอยู่ในช่วง 04H ถึง 07H โดยมีรูปแบบคำสั่งดังนี้

โดยที่บิต I/D เป็นบิตที่ใช้กำหนดว่า เมื่อเขียนหรืออ่านข้อมูลแล้วจะทำให้แอดแดรสของ DDRAM เพิ่มขึ้นหรือลดลงหนึ่งแอดแดรส หากกำหนดให้ I/D เป็นลอจิก "1" แอดแดรสของ DDRAM จะเพิ่มขึ้นและหากกำหนดให้เป็นลอจิก "0" แอดแดรส DDRAM จะลดลง

บิต S เป็นบิตที่ใช้สำหรับกำหนดลักษณะของการแสดงผลเมื่อมีการป้อนข้อมูล หากกำหนดให้บิต S เป็นลอจิก "1" เมื่อมีข้อมูลใหม่ปรากฏบนจอแสดงผลแล้วตัวเคอร์เซอร์จะอยู่กับที่ตัวอักษรข้อมูลเดิมจะถูกเลื่อนไปทางซ้ายหากกำหนดให้บิต S เป็นลอจิก "0" ข้อมูลตัวอักษรจะอยู่กับที่และตัวเคอร์เซอร์จะเลื่อนไปทางขวา

4.คำสั่งควบคุมการแสดงผล (Display ON/OFF Control)

เป็นคำสั่งสำหรับควบคุมการเปิด-ปิดจอแสดงผลและควบคุมเคอร์เซอร์บนจอแสดงผล โดยมีค่าข้อมูลคำสั่งสำหรับควบคุมการแสดงผลจะมีค่าตั้งแต่ 08H ถึง 0FH (8 รูปแบบคำสั่ง) มีรูปแบบคำสั่งดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยที่

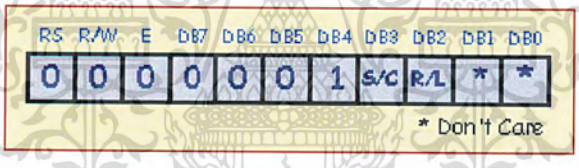
บิต B เป็นบิตที่ใช้สำหรับควบคุมการกระพริบของเคอร์เซอร์ถ้ากำหนดให้ B เป็นลอจิก "1" เคอร์เซอร์จะกระพริบและหากกำหนดให้ปีนลอจิก "0" ตัวเคอร์เซอร์จะไม่กระพริบ

บิต C จะใช้สำหรับควบคุมการแสดงตัวเคอร์เซอร์บนจอแสดงผล หากกำหนดให้บิต C มีค่าเป็นลอจิก "1" จะเป็นการกำหนดให้แสดงตัวเคอร์เซอร์และหากกำหนดให้เป็นลอจิก "0" จะไม่แสดงตัวเคอร์เซอร์หรือปิดเคอร์เซอร์นั้นเอง

บิต D ใช้สำหรับควบคุมการเปิดปิดจอแสดงผลซึ่งหากกำหนดให้บิต D เป็นลอจิก "0" จะเป็นการปิดจอแสดงผลและหากกำหนดให้เป็นลอจิก "1" จะเป็นการเปิดจอแสดงผล

5.คำสั่งควบคุมการเลื่อนเคอร์เซอร์และข้อมูลตัวอักษร (Cursor or Display Shift)

เป็นคำสั่งสำหรับกำหนดให้ตำแหน่งเคอร์เซอร์หรือข้อมูลเกิดขึ้นทางซ้ายหรือขวาสำหรับคำสั่งนี้จะมีรหัสคำสั่งตังแต่ 10H ถึง 1FH โดยมีรูปแบบคำสั่งดังนี้



โดยที่

S/C	R/L	ลักษณะการเลื่อน
0	0	เลื่อนเคอร์เซอร์จากตำแหน่งเดิมไปทางซ้าย 1 ตำแหน่ง
0	1	เลื่อนเคอร์เซอร์จากตำแหน่งเดิมไปทางขวา 1 ตำแหน่ง
1	0	เลื่อนตัวอักษรที่เกิดขึ้นใหม่ไปทางซ้าย
1	1	เลื่อนตัวอักษรที่เกิดขึ้นใหม่ไปทางขวา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6.คำสั่งกำหนดฟังก์ชันการทำงาน (Function Set)

เป็นคำสั่งสำหรับกำหนดลักษณะการส่งผ่านข้อมูลเป็นแบบ 8 บิตหรือ 4 บิต กำหนดจำนวณบรรทัดการแสดงผลและความละเอียดของตัวอักษรในการแสดงผล โดยมีรูปแบบคำสั่งดังนี้

โดยที่

บิต F เป็นบิตสำหรับเลือกขนาดความละเอียดของตัวอักษรที่จะแสดงผล หากกำหนดให้บิต F มีค่าเป็นลอจิก "0" จะเป็นการแสดงผลแบบ 5x7 จุด และหากกำหนดค่าเป็นลอจิก "1" การแสดงผลจะเป็นแบบ 5x10 จุด

บิต N เป็นบิตสำหรับกำหนดจำนวนบรรทัดในการแสดงผลหากกำหนดบิต N ให้มีค่าเป็นลอจิก "0" จะเป็นการแสดงผลแบบ 1 บรรทัด และหากกำหนดให้เป็นลอจิก "1" การแสดงผลจะได้มากว่า 2 บรรทัด บิต DL เป็นบิตสำหรับกำหนดจำนวนบิตที่จะส่งผ่านข้อมูล หากกำหนดให้บิต DL มีค่าเป็นลอจิก "0" จะเป็นการกำหนดให้ส่งผ่านข้อมูลแบบ 4 บิต และหากกำหนดให้เป็นลอจิก "1" เป็นการกำหนดให้การส่งผ่านข้อมูลเป็นแบบ 8 บิต

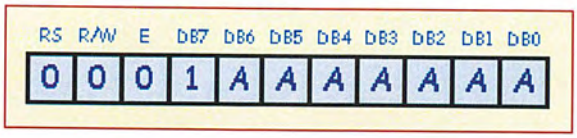
7.คำสั่งเลือกแอดแดรสของ CGRAM (Set CGRAM Address)

จะเป็นการกำหนดแอดแดรสใน CGRAM โดยจะต้องทำการกำหนดแอดแดรสก่อนเขียนหรืออ่านข้อมูลจาก CGRAM โดยที่แอดแดรสของ CGRAM จะอยู่ระหว่าง 00H ถึง 3FH โดยมีรูปแบบคำสั่งดังนี้

8.คำสั่งเลือกแอดแดรสของ DDRAM (Set DDRAM Address)

เป็นคำสั่งสำหรับกำหนดค่าแอดแดรสใน DDRAM ก่อนที่จะทำการอ่านข้อมูลหรือเขียนข้อมูลจาก DDRAM โดยที่แอดแดรสของ DDRAM จะขึ้นอยู่กับการกำหนดค่า N ในคำสั่งกำหนดฟังก์ชันการทำงานด้วย หากกำหนดค่า N เป็นลอจิก "0" (1 บรรทัด) แอดแดรสของ DDRAM จะอยู่ระหว่าง 80H ถึง 0CFH และหากกำหนดค่า N เป็นลอจิก "1" (2 บรรทัด) ค่าแอดแดรสของ DDRAM จะมี 2 ช่วงคืออยู่ระหว่าง 80H ถึง 87H และ C0H ถึง C7H โดยมีรูปแบบคำสั่งดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



การเชื่อมต่อและควบคุมจอแสดงผลแบบ LCD

เมื่อเริ่มต้นใช้งาน LCD จะต้องกำหนดค่าให้แก่ขา RS ของโมดูล LCD เพื่อให้คอนโทรลเลอร์ภายใน LCD รู้ว่าข้อมูลต่อไปที่จะได้รับจากอุปกรณ์ภายนอกนั้นเป็นรหัสคำสั่งหรือข้อมูลที่จะแสดงผล จากนั้นจะเป็นจะเป็นการส่งข้อมูลมาที่ขาข้อมูลของ LCD ขั้นตอนต่อไปจะต้องส่งสัญญาณพัลซ์ให้แก่ขา E เพื่อแอนนาเบิลให้ LCD รับข้อมูลจากบัสข้อมูลเข้าไป โดยพัลซ์ที่ป้อนให้แก่ขา E ต้องเป็นพัลซ์ขอบขาลง แสดงความสัมพันธ์ของขา RS , R/W และ E เพื่อการอ่านและเขียนข้อมูลกับโมดูล LCD ดังตารางที่ 1

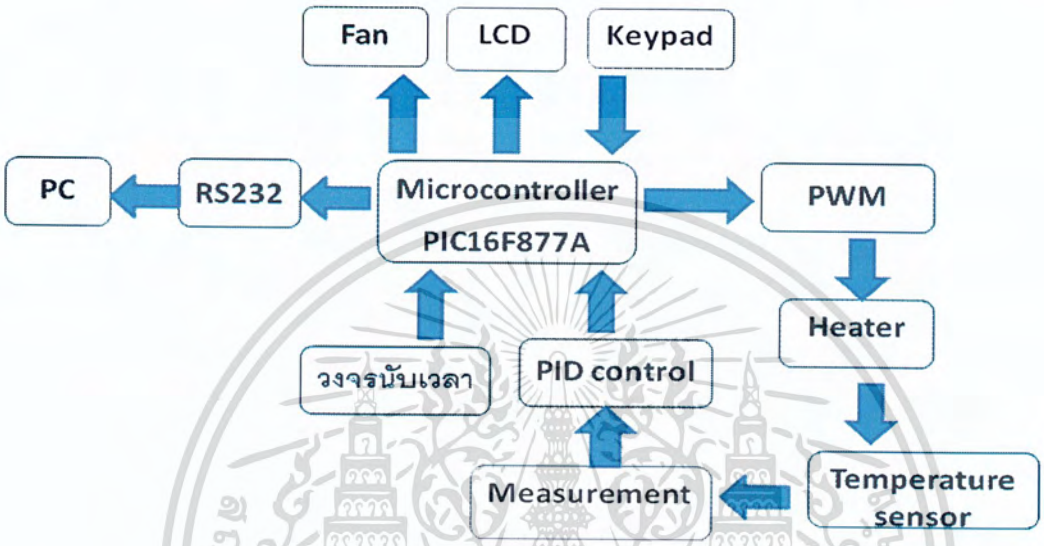
ตารางที่ 4 ความสัมพันธ์ของขา RS , R/W และ E

RS	R/W	E	การทำงาน
0	0		เขียนคำสั่งให้แก่ LCD
0	1		อ่านสถานะจาก LCD
1	0		เขียนข้อมูลให้แก่ LCD
1	1		อ่านข้อมูลจากแก่ LCD

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3
หลักการและการออกแบบ

3.1 Block Diagram



รูปที่ 3.1 บล็อกไดอะแกรมแสดงส่วนประกอบต่างๆ

3.2 ส่วนประกอบของเตาอบควบคุมอุณหภูมิ

เตาอบควบคุมอุณหภูมิประกอบด้วย 4 ส่วนคือ

1.) ส่วนsensor ตรวจวัดอุณหภูมิภายในเตาอบ
2.) ส่วนรับค่าการตั้งเวลาและอุณหภูมิเพื่อที่จะควบคุมอุณหภูมิ
3.) ส่วนแสดงผลค่าของอุณหภูมิ, การตั้งค่าอุณหภูมิที่ต้องการควบคุมและการตั้งเวลาในการควบคุม
4.) ส่วนของHeaterในการให้ความร้อนภายในเตาอบควบคุมอุณหภูมิ

3.2.1 ส่วนSensor ตรวจวัดอุณหภูมิภายในเตาอบ

รูปที่3.2 Sensor DS18S20

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เซนเซอร์วัดอุณหภูมิที่ใช้คือเบอร์DS18S20 มีคำสั่งที่ใช้ในการเขียนโปรแกรมเพื่อควบคุมการทำงานเซนเซอร์ 3 คำสั่งหลักๆคือ

1.) Skip Rom (0CCH)

ใช้กรณี DS1820 ต่ออยู่บนสายสัญญาณเพียงตัวเดียว ก็ ไม่จำเป็นต้องทำการอ่านค่า Address ของ อุปกรณ์ เพื่อประหยัดเวลา

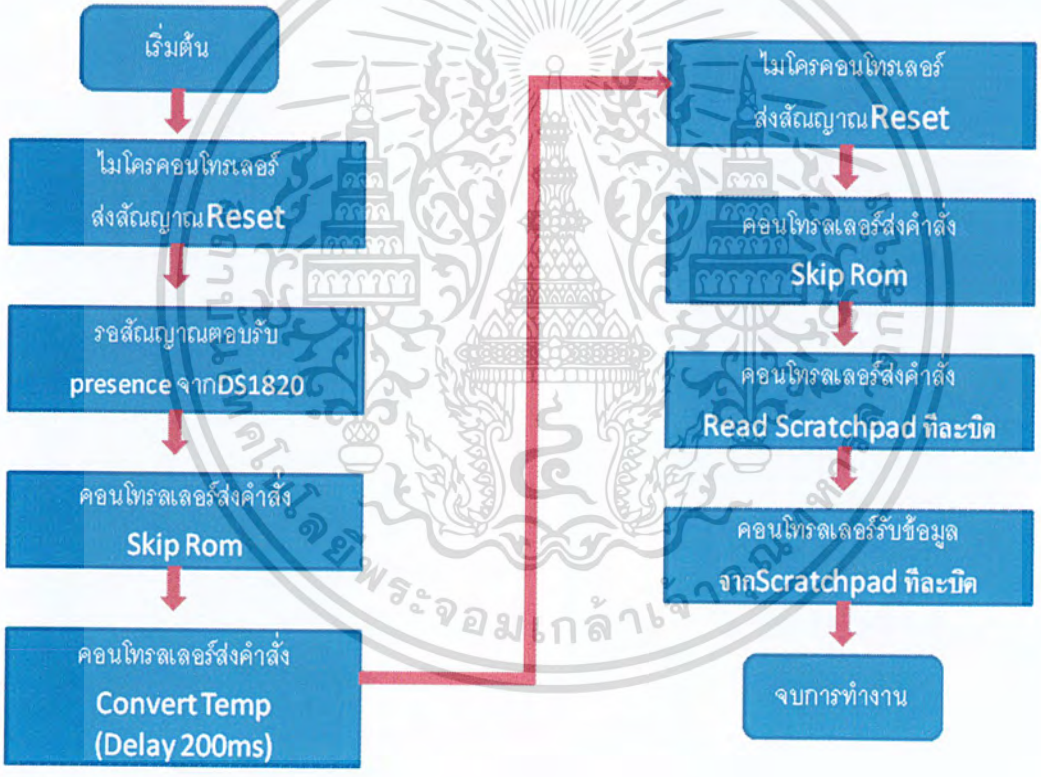
2.) Convert Temperature (044H)

DS1820 จะทำการแปลงค่าอุณหภูมิมาเก็บที่ Scratchpad

3.) Read Scratchpad (0BEH)

DS1820 จะทยอยส่งข้อมูลค่าอุณหภูมิจาก Scratchpad ทั้ง 9 ไบต์ออกมา

3.2.2 ขั้นตอนการทำงานของ DS18S20



รูปที่3.3 แสดงFlow chart การทำงานเซนเซอร์DS18S20

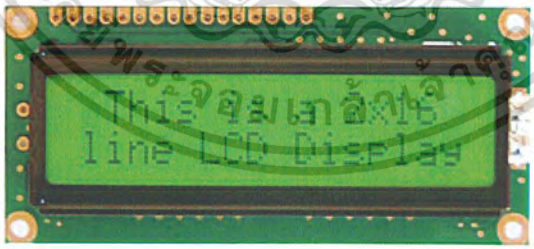
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.3 ส่วนรับค่าการตั้งเวลาและอุณหภูมิเพื่อที่จะควบคุมอุณหภูมิ

รูปที่3.4 keypad 4*3

ใช้ Matrix switch หรือ keypad ในการรับค่าโดยมีการต่อR-pull up ไว้ที่สายของ Keypad สามารถตรวจสอบการกดswitch ได้โดยใช้เทคนิคในการสแกนคีย์เข้ามาช่วย ซึ่งถูกเขียนและควบคุมจาก Software โดยหลักการจะทำการสแกนในแนวแถวหรือคอลัมน์ เช่น หากให้สแกนในแนว แถว ก็จะส่ง Logic '0' ไปที่ละแถวจนครบ 4 แถวหากมีการกด Key ตำแหน่งในแนวคอลัมน์ก็จะเปลี่ยนเป็น "0" เราก็จะเช็คค่าจาก Port ทั้ง Port เข้ามาว่ามีค่าเป็นเท่าไรแล้วนำไปกำหนดตามตำแหน่งจริงของ Keypad ก็จะได้เป็นค่าการกดตัวเลขนั้นออกมา

3.2.4 ส่วนแสดงผลค่าของอุณหภูมิ, การตั้งค่าอุณหภูมิที่ต้องการควบคุมและการตั้งเวลาในการควบคุม



รูปที่3.5 จอแสดงผล LCD 16*2

จะใช้เป็น LCD ขนาด 16 ตัวอักษร 2 บรรทัดแบบมี Back light ซึ่งจะมีการควบคุมการทำงานของ Back light โดยใช้ทรานซิสเตอร์ที่ทำหน้าที่เป็นสวิตช์ การควบคุม LCD จะเป็นการควบคุมการทำงานแบบ 4 bits ซึ่งเป็นการประหยัด Port ในการใช้งานสามารถปรับแสงสว่างของหน้าจอ LCD ได้ด้วยตัวต้านทานปรับค่าได้ 10 K Ω

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3 PROCESS CONTROL

ในขบวนการผลิตโดยเฉพาะการผลิตแบบต่อเนื่อง และการผลิตที่มีขนาดใหญ่ เช่น ขบวนการกลั่นน้ำมัน ขบวนการผลิตไฟฟ้า ขบวนการถลุงเหล็ก นั้น ล้วนแล้วแต่เป็นขบวนการที่มีความซับซ้อนค่อนข้างมาก การควบคุมขบวนการผลิตเพื่อให้ได้ผลิตผลที่ดีมีคุณภาพตามต้องการ (desired productivity) รวมทั้งต้องประหยัดด้วย เหล่านี้นับเป็นสิ่งสำคัญ อันเป็นที่ต้องการสำหรับเจ้าของกิจการ

การนำเทคโนโลยี่การวัดคุมและระบบควบคุม(instrumentation and control system) เข้ามาใช้งานจึงเป็นสิ่งสำคัญที่สุด เพื่อให้บรรลุวัตถุประสงค์ของการผลิตที่ดีตามต้องการ

ในปัจจุบัน เทคโนโลยีระบบควบคุม ได้รับการพัฒนาอย่างรวดเร็ว อุปกรณ์แบบใหม่ ๆ ได้ถูกนำมาใช้ในงาน เพื่อเพิ่มประสิทธิภาพและประสิทธิผลในการผลิต

ตัวควบคุมแบบ Analog PID Controller ซึ่งเคยถูกนำมาใช้แทนที่ Pneumatic PID Controller กำลังจะล้าสมัยไปแบบสิ้นเชิง การนำเอาไมโครคอมพิวเตอร์มาประยุกต์ใช้ในงานควบคุมได้เข้ามาแทนที่เกือบจะทั้งหมด

อย่างไรก็ตาม ถ้าพิจารณากันให้ลึกซึ้งจะพบว่าหลักการควบคุม (Control Regulatory) ของตัวควบคุมไม่ว่าจะเป็นแบบ Pneumatic , Analog , Digital , Computer control หรือ อะไรก็แล้วแต่ยังคงใช้หลักการเหมือนเดิมไม่เปลี่ยนแปลง เพียงแต่พัฒนาการของระบบควบคุมจะทำให้มีประสิทธิภาพมากขึ้น โดยที่มีขนาดเล็กลง

ระบบควบคุมส่วนใหญ่ ในโรงงานอุตสาหกรรมก็ยังคงเป็นแบบ P-only, PI ,PD หรือ PID นั่นเองไม่เปลี่ยนแปลง

เนื้อหาในส่วนที่จะกล่าวนี้ เป็นการสร้างความเข้าใจพื้นฐานที่ดี โดยเน้นให้ผู้อ่านสามารถนำไปใช้งานได้ทันที (ready to application) มากกว่าการนำเสนอแบบตำราเรียนทั่วไป

อย่างไรก็ตาม สิ่งที่จำเป็นอย่างยิ่งที่จะสร้างทักษะได้นั้น คือการปฏิบัติควบคู่ไปด้วย จะทำให้เข้าใจปัญหาและสามารถแก้ไขได้เป็นอย่างดี ท่านควรจะต้องหาโปรแกรม PID simulator มาทดลองใช้งานด้วยจะเป็นการดีและทำให้เข้าใจtheory แบบแตกฉานได้

วิธีการควบคุมแบบ PID control ซึ่งเป็นส่วนหนึ่งของการควบคุมในระบบวงปิดหรือระบบควบคุมแบบป้อนกลับ (Closed-loop Control Systems, Feedback Control Systems) โดย การควบคุมแบบ PID จะประกอบไปด้วยส่วนการควบคุมที่สำคัญด้วยกันคือ

Proportinal control action (P - Action)

Integral control action (I-Action)

Derivative control action (D-Action)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

P Action

เป็นการกำหนดการทำงานของ ouput ให้เป็นสัดส่วนเปอร์เซ็นกับค่า error หรือการเปลี่ยนแปลงของค่าที่วัดได้

Output = (error X 100) / Pb ; error=(ค่าset point) - (ค่าที่วัดได้)

ในทางปฏิบัติ P Action จะเข้าใกล้ค่าหนึ่ง ซึ่งไม่ใช่ค่า set point จริง ซึ่งเรียกว่าค่า offset

I Action

จะใช้ในการแก้ปัญหา offset ระบบควบคุม I Action จะเข้าไปช่วยกำจัดค่า offset ที่ยังคงมีอยู่ให้ระบบเข้าสู่ set point โดยค่า output ที่ออกมาจะขึ้นอยู่กับ Integral Time ที่กำหนดขึ้นมาตั้งแต่ต้น หากกำหนดให้ integral time น้อย ระบบจะเข้าสู่ set point ได้อย่างรวดเร็วแต่จะเกิดการกระเพื่อม hunting ของ process มากด้วย และหากกำหนดให้ Integral time มากจะเกิด hunting น้อย แต่จะใช้เวลานานกว่าระบบจะเข้าสู่ set point

D Action

ในกรณีที่มีการรบกวนระบบจากภายนอก disturbance เป็นผลให้ process ของระบบมีการเปลี่ยนแปลงอย่างทันที่ทันใด เราควรจะใช้การควบคุมแบบ D Action Derivative ซึ่งจะมีการตอบสนองที่รวดเร็ว เป็นผลให้ระบบเข้าสู่ set point ได้รวดเร็วขึ้น

การปรับจูน

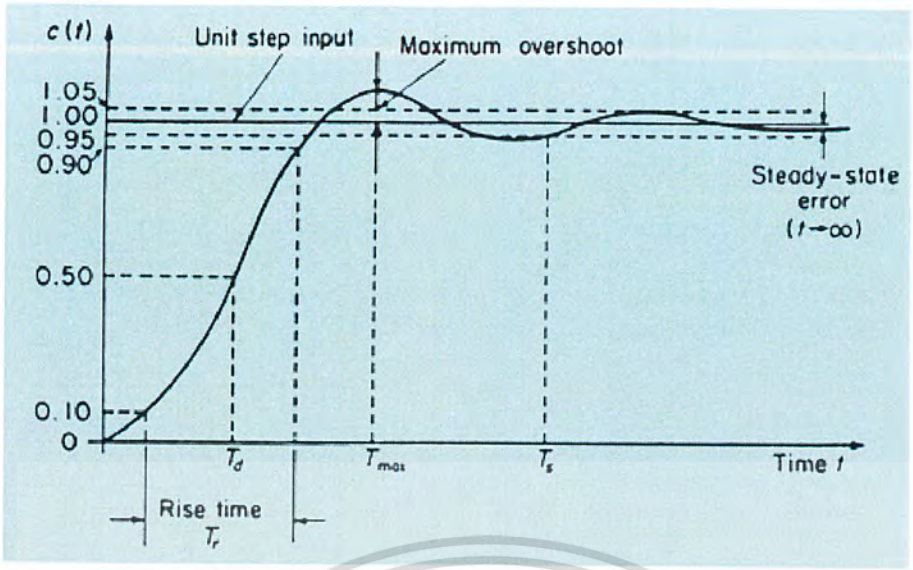
วิธีการ Ziegler–Nichols โดยวิธีการนี้นำเสนอโดย John G. Ziegler และ Nathaniel B. Nichols ในคริสต์ทศวรรษที่ 1940 ขั้นแรกให้ตั้งค่า Ki และ Kd เป็นศูนย์ เพิ่มอัตราขยาย P สูงที่สุด, Ku, จนกระทั่งเริ่มเกิดการแกว่ง นำค่า Ku และค่าช่วงการแกว่ง Pu มาหาค่าตัวแปรที่เหลือดังตาราง:

Ziegler–Nichols method

Control Type	Kp	Ki	Kd
P	0.50Ku	-	-
PI	0.45Ku	1.2Kp / Pu	-
PID	0.60Ku	2Kp / Pu	KpPu / 8

การทดสอบประสิทธิภาพของระบบ FeedBack เราจะใช้สัญญาณอินพุทแบบ Unit Step ใส่เข้าไปในระบบเพื่อใช้วิเคราะห์ transient ในระบบควบคุมแบบป้อนกลับโดยสามารถตรวจจากลักษณะดังรูปนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.6 แสดง Transient ในระบบควบคุมแบบป้อนกลับ

1. Steady State Performance

ค่านี้จะได้จากการหาค่า steady-state error ของการตอบสนองต่อฟังก์ชันอินพุทพื้นฐานที่กล่าวมาแล้ว

2. Transient Performance

ค่านี้จะวิเคราะห์โดยการใส่ unit-step function เป็นอินพุทอ้างอิงและจะได้ผลตอบสนองดังรูป ลักษณะที่สำคัญของผลตอบสนองนี้ได้แก่

Overshoot

เป็นค่า error ที่มากที่สุดระหว่างอินพุทและเอาท์พุท ค่านี้จะใช้ในการประมาณความเสถียรของระบบ ค่า overshoot จะวัดเป็นสัดส่วนเทียบกับค่าสุดท้ายหรือค่าอินพุทอ้างอิงดังนี้

Percent overshoot = Maximum overshoot / Final desired value x 100

Time delay

ค่า time delay td เป็นช่วงเวลาที่ใช้ในการตอบสนองของระบบตั้งแต่เริ่มต้นจนกระทั่งเอาท์พุทมีค่าเป็น 50 % ของค่าอินพุทอ้างอิง

Rise time

ค่า rise time tr เป็นช่วงเวลาตั้งแต่เอาท์พุทมีค่าเป็น 10 % จนถึง 90 % ของค่าอินพุทอ้างอิง

Setting time

ค่า setting time ts เป็นช่วงเวลาตั้งแต่เริ่มต้นจนกระทั่งการแกว่งของเอาท์พุทลดลงอยู่ในขอบเขตที่กำหนด โดยปกติแล้วขอบเขตนี้จะอยู่ในช่วง 5 % ของอินพุทอ้างอิง นอกจากนี้ยังมีลักษณะที่สำคัญอื่นๆอีก เช่น damping ratio, damping factor และ undamped natural frequency ซึ่งไม่ได้แสดงในรูป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. Process definition

คือ การเปลี่ยนแปลงสภาพทาง physics หรือทางเคมีของสสารหรือการเปลี่ยนรูปของพลังงาน เช่น

- ขบวนการทำความเย็น
- ขบวนการกลั่นน้ำมัน
- ขบวนการผลิตไฟฟ้า เป็นต้น

2. Instrument

เป็นอุปกรณ์ที่นำมาใช้เพื่อวัดค่าของ Process เพื่อให้รับรู้ถึงสภาพของ Process โดยมีวัตถุประสงค์เพื่อการปรับแต่งควบคุม (Control) ให้ Process เป็นไปตามที่ต้องการ (Set Point)ในการวัดและการควบคุม เพื่อให้มีคุณสมบัติตามที่ต้องการมี ตัวแปร ที่เกี่ยวข้องที่ต้องการวัดค่า เช่น

- อุณหภูมิ (Temperature)
- ความดัน (Pressure)
- ระดับ (Level)
- ปริมาณการไหล (Flow Rate)
- น้ำหนัก (Weight)
- แรงดันไฟฟ้า (Voltage)
- กระแสไฟฟ้า (Current)
- ความหนืด (Viscosity)
- ความเร็ว (Velocity or Speed)
- ความชื้น (Moisture)
- ความต้านทาน (Resistance)
- อัตราเร่ง (Acceleration)
- การสั่นสะเทือน (Vibration)
- ค่า pH

เป็นต้น

2.1 รูปแบบของ Instrument ที่นำมาใช้งานมีหลายแบบ ตามลักษณะความต้องการใช้งาน ได้แก่

- Indicator เพื่อแสดงค่าของตัวแปร ไม่สามารถส่งสัญญาณไฟฟ้าได้ เช่น pressure indicator
- Transmitter (Tx) เพื่อแสดงค่าของตัวแปร โดยสามารถส่งเป็นสัญญาณไฟฟ้าได้ด้วย มีทั้ง Analog และ Digital Signal เช่น pressure transmitter
- Switch เพื่อแสดงค่าของตัวแปร โดยส่งเป็นสัญญาณไฟฟ้าได้ด้วย เป็นแบบ On-Off (Digital Signal) เช่น pressure switch
- Recorder Device เพื่อเก็บบันทึกของตัวแปร ปัจจุบันนิยมใช้ในรูปแบบของ Electronic Format เช่น เก็บไว้ใน Hard Disk
- Controller เป็นอุปกรณ์ที่ใช้ควบคุมขบวนการ เป็นลักษณะของการ programming เช่น PLC , PID control
- Alarm Device เป็นอุปกรณ์ที่มีไว้เพื่อเตือนค่าของตัวแปรที่อาจเป็นอันตรายต่อระบบหรือ Process

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

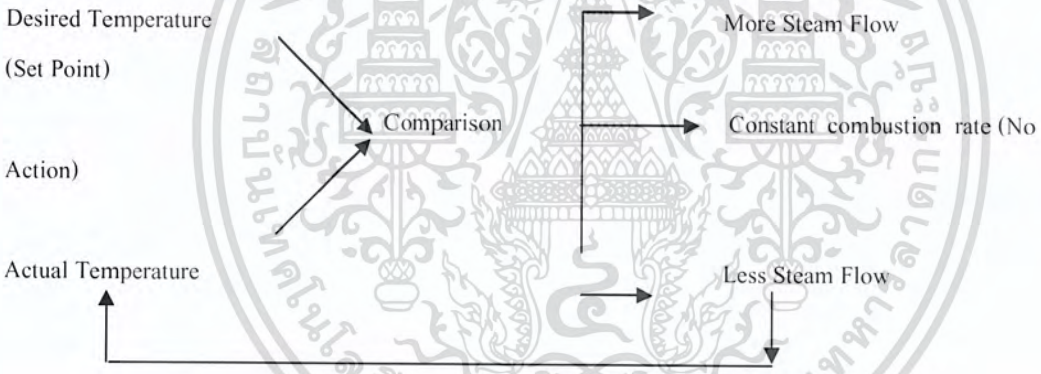
- Interlocking Device เป็นอุปกรณ์สำหรับป้องกันอันตรายของระบบหรือ process โดยมีรูปแบบต่าง เช่น ต้องมีสัญญาณของการจ่ายน้ำ (Flow Rate) เกิดขึ้นก่อนจึงจะสามารถจ่ายสารเคมีบางส่วนได้
- Transducer มีคุณสมบัติคล้าย ๆ กับ Transmitter แต่ไม่ได้ Generate สัญญาณไฟฟ้ามาตรฐานออกมา (Standard Signal : 4 -20 mA ,1-5 V)

3. Control Function

หน้าที่หลักของการควบคุมสามารถจำแนกได้ 3 ประการ คือ

- Gathering Information เป็นการรับรวบรวมข้อมูลเกี่ยวกับตัวแปรที่เราต้องการควบคุม และเกี่ยวข้อง
- Decision ประมวลผลและตัดสินใจในข้อมูลที่รับเข้ามาในขั้นตอน Gathering
- Take Action ส่งค่า Output ออกไปสั่งการ Final Drive จากผลของ Decision

ตัวอย่างเช่น



3.1 Type of Control

การควบคุมสามารถกระทำได้ด้วยการควบคุมแบบ Manual และ Automatic ในการควบคุมแบบ Manual การตัดสินใจสั่งการมนุษย์จะเป็นผู้กระทำ ส่วนในการควบคุมแบบอัตโนมัติ (Automatic Control) การตัดสินใจสั่งการจะกระทำด้วยอุปกรณ์ หรือที่รู้จักกันคือ Controller เช่นในการควบคุมความเร็วของรถยนต์แบบ Manual ผู้ที่จะควบคุมตัดสินใจก็คือคนขับรถ

ความแตกต่างของการควบคุมแบบ Manual และแบบอัตโนมัติ (Automatically) พิจารณาจากการตัดสินใจ (Decision) ว่าได้กระทำโดยอะไร มนุษย์หรือเครื่องจักร (Man or Machine)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.1 วัตถุประสงค์ของ process control

- เพิ่มประสิทธิภาพการผลิต
- ป้องกันความเสียหายของระบบ และผู้ปฏิบัติงาน
- ลดค่าใช้จ่ายในการผลิต เช่นใช้ operator น้อยลง
- ใช้กับขบวนการผลิตแบบต่อเนื่องและจำนวนมากๆ

3.2 Process Control Terms (ควรทำความเข้าใจ และจำให้ได้)

Terms หรือคำจำกัดความที่พบเห็นบ่อย ๆ ใน Process Control มีดังนี้

3.2.1 Controlled Variable

คือตัวแปรของ Process ที่เราต้องการควบคุม ค่าตัวแปรควบคุมที่ใช้ส่วนมาก ได้แก่ อุณหภูมิ, ความดัน, อัตราการไหล, ระดับ เช่นในการควบคุมอุณหภูมิของไอน้ำ Controlled Variable ก็คือ อุณหภูมิ

3.2.2 Measured Variable

เป็นการวัดค่าของตัวแปรที่เราต้องการควบคุม อาจจะเป็นตัวเดียวกับ Controlled Variable หรือไม่ก็ได้

3.2.3 Set Point

เป็นค่าเป้าหมายในการควบคุม เช่น Steam Temp. Set Point = 560°C เป็นต้น ในการควบคุมตัวแปรที่เราวัดซึ่งก็คือ Measured Variable และ Controlled Variable โดยทั่วไปจะเป็นตัวเดียวกัน เช่นการควบคุมอุณหภูมิของน้ำ Measure Variable และ Controlled Variable คืออุณหภูมิ

ในบางกรณี เช่นการควบคุมระดับน้ำในถัง Controlled Variable ก็คือระดับน้ำ แต่ Measured Variable สามารถวัดในรูปของความดันหรือวัดความดันแตกต่างจากนั้นจึงแปลงค่าความดันไปเป็นระดับของน้ำได้ หรือการวัด Flow Rate ค่า Measured Variable คือ Differential Pressure

3.2.4 Deviation or Error

เมื่อค่า Set Point และค่า Controlled Variable ถูกนำมาเปรียบเทียบกันถ้าเกิดความแตกต่างระหว่างค่า 2 ค่านี้เราเรียกว่า Deviation หรือ Error

Error เป็นคำที่ใช้เรียกเมื่อค่าทั้งสองแตกต่างกัน มิใช่หมายความว่า ค่าความผิดพลาด ส่วนมากใช้กับ instrument คือมีค่าเท่ากับ ค่าที่วัดได้ -ค่ามาตรฐานหรือค่าที่วัดได้ที่เที่ยงตรงกว่า

Deviation จะเน้นช่วง Error ที่เกิดในช่วงเริ่มต้น และเป็นคำที่ใช้กันในระบบ control

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.5 Manipulated Variable

เมื่อมี Deviation เกิดขึ้นในการควบคุมตัวควบคุม (Controller) ก็จะส่ง Control Output ออกไป เพื่อขจัดค่า Deviation นี้ Control Output นี้จะไปทำการปรับแต่งค่าตัวแปรที่เรียกว่า "Manipulated Variale" ซึ่งเป็นตัวที่ทำให้เกิดการเปลี่ยนแปลงของ "Controlled Variable"

เช่นในการควบคุมอุณหภูมิของน้ำด้วยการผ่าน Heat Exchanger ด้วยการใช้ไอน้ำ อุณหภูมิของน้ำจะถูกวัดและนำมาเปรียบเทียบกับค่า Set Point ถ้าค่าทั้งสองนี้ไม่เท่ากัน ก็ต้องมีการปรับค่าปริมาณของไอน้ำ เพื่อทำให้อุณหภูมิเข้าสู่ค่า set point ในกรณีนี้ไอน้ำก็คือ Manipulated Variable ซึ่งก็คือตัวแปรปรับแต่ง process นั่นเอง

3.2.6 Disturbances or Upset

การควบคุมอุณหภูมิของน้ำที่ผ่าน Heat Exchanger ที่กล่าวมาแล้ว การควบคุมอุณหภูมิจะขึ้นอยู่กับการปรับปริมาณ steam flow โดยตรง แต่ยังมีปัจจัยอื่นที่ทำให้อุณหภูมิของน้ำที่เราควบคุม (water flow out temperature) เปลี่ยนไปทั้ง ๆ ที่ปริมาณ flow rate ไอน้ำเท่าเดิมเราเรียกว่า "Disturbance" หรือ "Upset" เช่น มีการเปลี่ยนปริมาณการใช้น้ำที่ปลายทาง , อุณหภูมิของไอน้ำเปลี่ยนแปลง , ความสะอาดของ Heat Exchanger , Ambient Temperature เป็นต้น

3.2.7 Closed Loop Control

เป็นการควบคุมชนิดหนึ่งซึ่ง Control Action จะขึ้นอยู่กับ Process Output

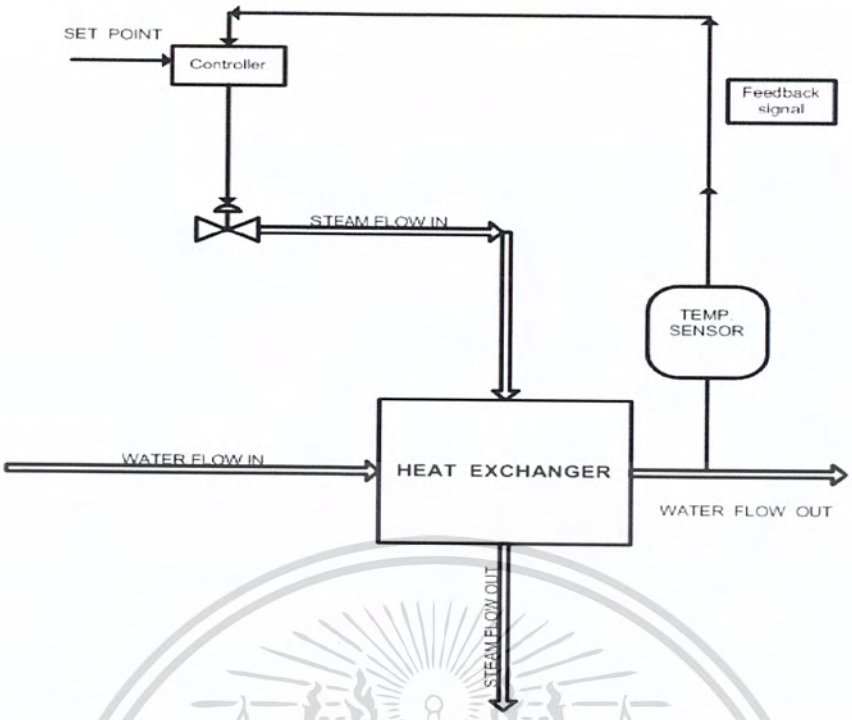
3.2.8 Feedback Control

เป็นส่วนหนึ่งของ Closed Loop Control โดยที่ Control Action จะขึ้นอยู่กับ Process Output โดยการวัดค่าตัวแปรที่เราต้องการควบคุมแล้วนำกลับเข้ามาเปรียบเทียบกับค่า Set Point

การควบคุมแบบ Feedback Control แบ่งได้เป็น 2 อย่าง คือ

- Negative Feedback เป็นการควบคุมที่นำค่าสัญญาณที่ต้องการควบคุมป้อนกลับเปรียบเทียบกับค่า Setpoint โดยนำมาลบกับค่า Set point
- Positive Feedback เป็นการควบคุมที่นำค่าสัญญาณที่วัดได้ป้อนกลับเข้ามาบวกกับค่า Set point

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.7 แสดง Feedback control

3.2.9 Feed Forward Control

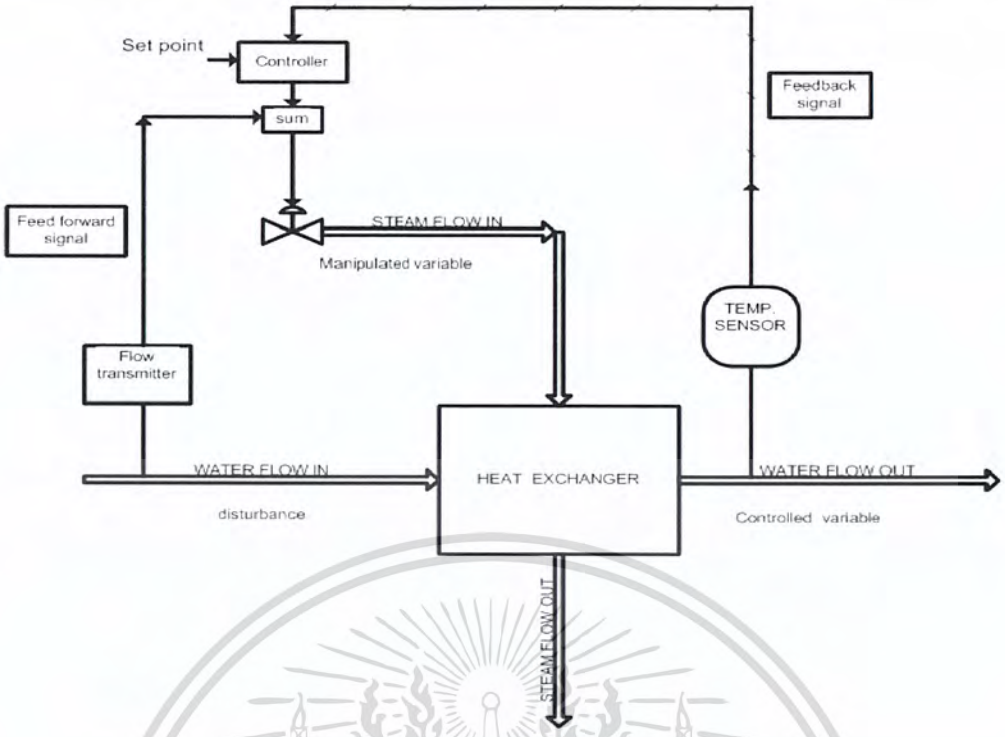
เป็นการควบคุมอีกแบบหนึ่งที่นำเอาค่าของสัญญาณของ Disturbance เข้ามาเพื่อทำการปรับแต่ง Manipulated Variable ก่อนที่ Controlled Variable จะเปลี่ยนแปลงไปมาก นิยมใช้กับ process ที่มีการเปลี่ยนแปลงของ load บ่อย ๆ เมื่อนำการควบคุมชนิดนี้เข้ามาใช้ร่วมใน Feedback Loop จะทำให้ Controlled Variable เกิดเปลี่ยนแปลงหรือมี Deviation น้อยที่สุด จากรูป

water temperature outlet = controlled variable

steam flow rate(inlet) = manipulated variable

water flow rate(inlet) = disturbance

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



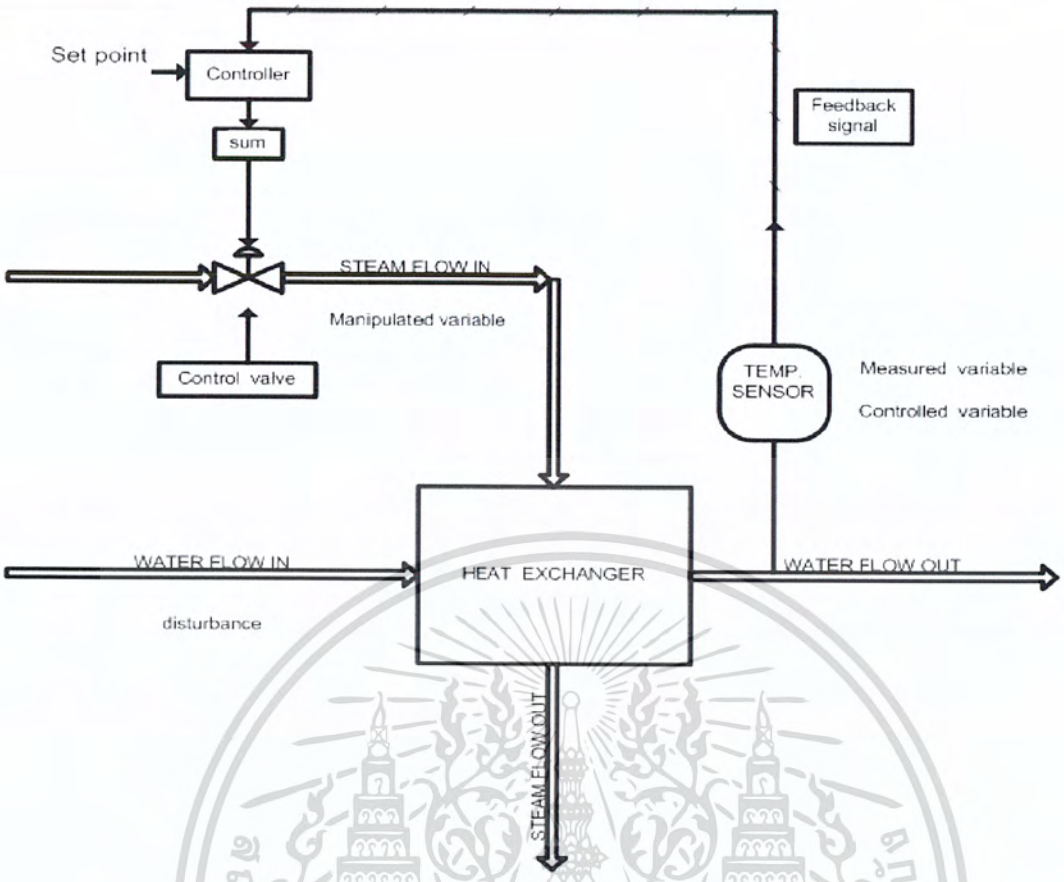
รูปที่ 3.8 แสดง Feedforward Control

4. Feedback Control Loop

ระบบควบคุมส่วนใหญ่จะอาศัยหลักการของ feedback control loop ในการควบคุมแบบนี้ เราจะใช้ sensor เป็นตัววัดค่า controlled variable และส่งข้อมูลให้กับ controller โดยมี comparator ซึ่งเป็นส่วนหนึ่งใน controller จะทำการเปรียบเทียบสัญญาณ controlled variable นี้กับ Set point ความแตกต่างที่ได้จากค่าทั้งสองจะเรียกว่า error หรือ deviation จากนั้น controller จะนำค่า error signal ไปเป็นตัวกำหนดขนาด และทิศทางการเปลี่ยนแปลงของ final control element เช่น control valve , damper เป็นต้น เพื่อทำการเปลี่ยนค่า manipulated variable

ค่า controlled variable หรือ measured variable (water temperature :out) จะถูกวัดโดย Sensor แล้วส่งค่ากลับมาที่ controller จากนั้น controller จะส่งสัญญาณที่ผ่านการประมวลผลแล้วตาม control mode เพื่อทำการปรับแต่ง manipulated variable(steam flow : in) เพื่อรักษาค่า controlled variable ให้ได้ค่าตามที่ต้องการ (Set point) อยู่ตลอดเวลา จากการทำงานลักษณะนี้เราจึงเรียบระบบนี้ว่า feedback control loop หรือ closed loop control ตามรูป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

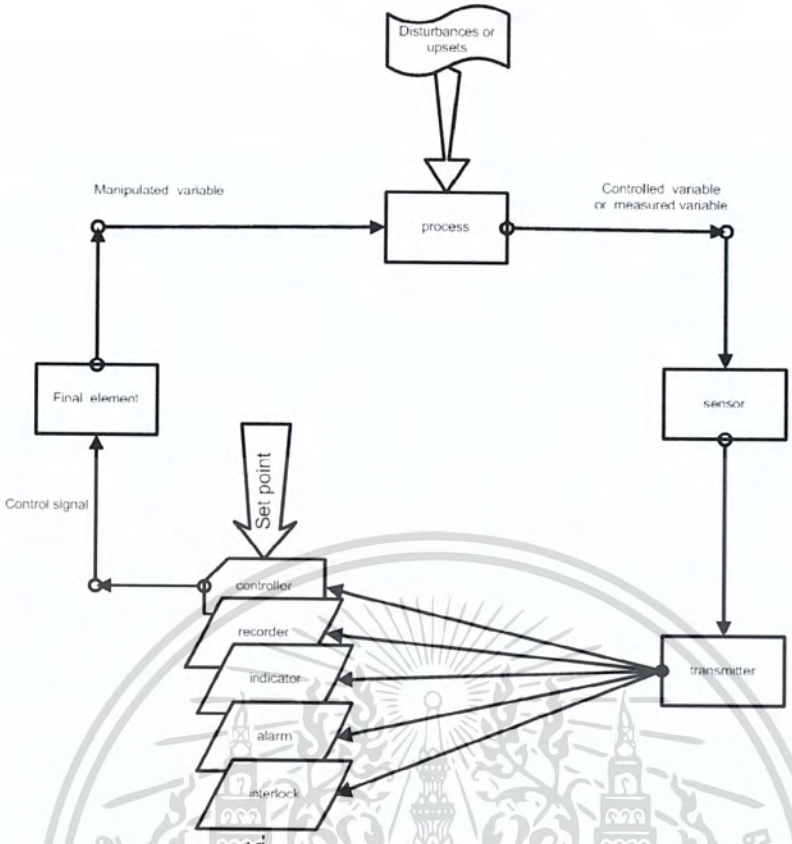


รูปที่ 3.9 แสดง Feedback control loop

จากรูป เป็นตัวอย่างของ feedback control loop ถ้าอุณหภูมิของน้ำร้อนที่วัดมาสูงกว่าค่า set point , controller จะส่งสัญญาณไปหรี่ control lvalve ลด steam flow rate ทำให้อุณหภูมิของน้ำร้อนลดลง กลับเข้าหา set point

ในทางกลับกัน ถ้าอุณหภูมิของน้ำร้อนที่วัดมาได้ต่ำกว่าค่า set point controller ก็จะส่งสัญญาณไปเปิดวาล์วเพิ่มขึ้น เพื่อเพิ่มปริมาณไอน้ำ ทำให้อุณหภูมิของน้ำร้อนเพิ่มขึ้น เป็นเช่นนี้สลับไปมา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.10 แสดง basic control loop

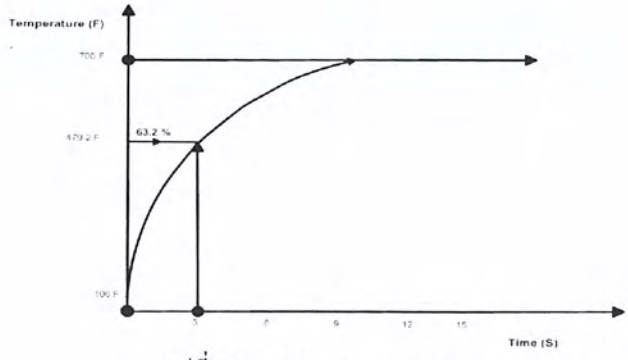
5. SENSOR

เป็นอุปกรณ์วัดค่าตัวแปร (PV) ได้แก่ temperature sensor , pressure sensor , level sensor เป็นต้น ในส่วนของรายละเอียด ควรศึกษาในเรื่องของ sensor and transducer อีกครั้ง ในที่นี้จะกล่าวถึงเฉพาะในเรื่องที่เกี่ยวข้องกับ control system เท่านั้น โดยมี term ที่เกี่ยวข้องดังนี้

5.1 Response Time

เป็นเวลาในการตอบสนองของ process ของตัว Sensor เมื่อมีการเปลี่ยนแปลงค่าของ input. โดยปกติ Response Time มักถูกใช้วัดในรูปของ Time Constant หนึ่งๆ

Time Constant หมายถึง การเปลี่ยนแปลงค่าของ Output ที่เปลี่ยนแปลงตามเวลาขึ้นกับค่า input โดยคิดเมื่อ Output เริ่มเปลี่ยนแปลงไปจนถึง 63.2% ของ Output ที่เปลี่ยนแปลงทั้งหมด ตามรูป



รูปที่ 3.11 แสดง time constant

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากกราฟเป็น Response ของเครื่องมือวัดอุณหภูมิ ปกติวัดอุณหภูมิอยู่ที่ 100 °F ถ้าอุณหภูมิเกิดเปลี่ยนเพิ่มขึ้นกระทันหันเป็น Step Change ถึง 700 °F แต่ผลตอบสนองของเครื่องวัดจะมีค่าค่อย ๆ เพิ่มขึ้นตามจนถึง 63.2% ของการเปลี่ยนแปลง 600 °F (700-100)

จากกราฟ จะเห็นว่าผลการตอบสนองการวัดของตัว Sensor จนชี้ถึงค่า 63.2% หรือที่ 479.2 °F (600*63.2%) นั้นใช้เวลา 3 วินาที ซึ่งก็คือ 1 Time Constant ดังนั้นจึงกล่าวได้ว่า 1 time constant มีค่าเท่ากับ 3 วินาที

การคำณวนเป็นดังนี้

2 Time Constant (6 วินาที)

700 °F - 479.2 °F = 220.8 °F

(220.8°F x 0.632) + 479.2 °F = 618.7 °F

3 Time Constant (9 วินาที)

700 °F – 618.7 °F =81.3 °F

(81.3 °F *0.632) + 618.7 °F = 670.5 °F

5.2 Accuracy

เป็นความสามารถของตัว Sensor ที่จะชี้แสดงค่าที่แท้จริง เครื่องมือวัดที่มี Accuracy ดีก็คือเครื่องมือที่สามารถชี้แสดงค่าได้เท่ากับค่าจริงหรือใกล้เคียงของจริงมากที่สุด แต่ไม่จำเป็นว่าทุก ๆ ครั้งของการวัดค่าเดิมจะแสดงค่าได้เท่ากันทุกครั้ง

รูปที่ 3.12 แสดง accuracy profile

5.3 Precision

เป็นความสามารถของตัว Sensor ที่จะแสดงค่าในการวัดที่ใกล้เคียงหรือเท่ากันทุกครั้ง เมื่อถูกนำไปวัดค่าตัวเดิมในสภาวะเดิมในแต่ละครั้ง ซึ่งค่าที่ได้จะเท่ากับหรือใกล้เคียงกับค่าจริงหรือไม่ก็ได้ หากใกล้เคียงกับค่าจริงก็เรียกว่า accuracy ได้ด้วย

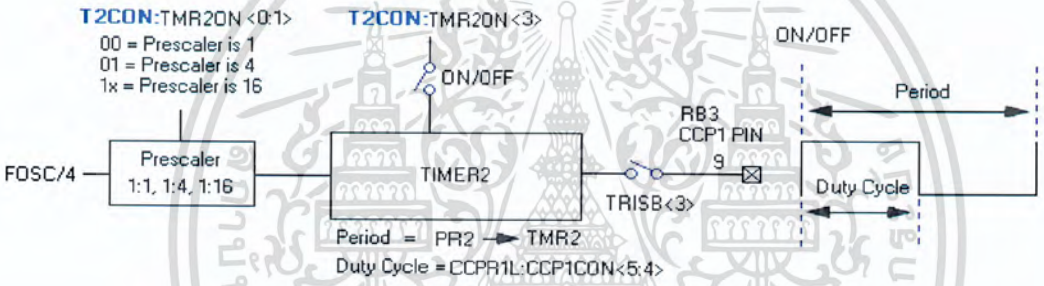
โดยปกติ sensor ที่ดีจะมีค่า accuracy and precision อยู่ใกล้เคียงกับค่าที่แท้จริงเสมอ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4 การใช้โมดูล CCP1 และ TIMER2 กำเนิดสัญญาณ (Pulse Generate)

โมดูล CCP(Capture/Compare/PWM) สามารถใช้กำเนิดสัญญาณ และมอดูเลชั่นทางความกว้างของพัลส์ความละเอียดขนาด 10 บิท PWM (Pulse Width Modulator) ซึ่งจะทำงานร่วมกันกับ TIMER2

CCP1 เป็นโมดูลที่ เมื่อกำหนดค่าเริ่มต้นในการทำงานให้แล้ว โมดูลนี้จะทำงานต่อไปได้เองโดยอิสระ ทำให้ MCU ไม่ต้องเสียเวลา วนลูปในการ จัดการ กับเอาต์พุต ในขณะที่โมดูล นี้ทำงาน MCU สามารถที่จะควบคุมการเปิด-ปิดสัญญาณ ผ่านทาง รีจิสเตอร์ควบคุมได้ - จึงทำให้สามารถใช้งานในการกำเนิดสัญญาณ อย่างเช่นการสร้างความถี่ 40KHz เพื่อเป็นความถี่พาหะ(CARRIER FREQUENCY) ของ Infrared Remote control (IR)



รูปที่ 3.13 การทำงานของโมดูล CCP1/TIMER2 ในการกำเนิดสัญญาณ

ส่วนกำเนิดสัญญาณ TIMER2 ได้รับสัญญาณนาฬิกา FOSC/4 จากระบบ โดยกำหนดตัวหารความถี่ (Prescaler) ได้จาก T2CON
รีจิสเตอร์ PR2 เป็นตัวกำหนดคาบเวลา

CCPR1 มีขนาด 16 บิทใช้ในการควบคุม duty cycle ประกอบด้วย CCPR1H และ CCPR1L

รีจิสเตอรควบคุมคือ CCP1CON CCP1CON<3:0>จะต้องกำหนดเป็น 11xx = PWM mode

ขั้นตอนการใช้งาน CCP1 กำเนิดสัญญาณ

1). หาคาบเวลาของ PWM (PWM PERIOD) เพื่อกำหนดค่าให้กับ PR2

1.1). กำหนดความถี่ (จะได้คาบเวลาของพัลส์ช=1/f) และ ดิวตี้ไซเคิลของพัลส์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.2).คำนวนหาค่ารีจิสเตอร์ PR2 เพื่อกำหนดคาบเวลาของ TIMER2 เพื่อใช้ผลิตความถี่ หาได้จากสูตร

$$PR2 = \frac{T}{4 \times T_{osc} \times P} - 1$$

เมื่อ T = คาบเวลา (1/f) ของพัลส์ , Tosc = คาบเวลา (1/f) ของ XTal , P = ค่าของ ปรีเสกลเลอร์ 1,4,16 โดยที่ค่า PR2 เป็นเลขจำนวนเต็มและมีค่าน้อยกว่า 2

2). หาดิวตี้ไซเคิลของ PWM (PWM DUTY CYCLE) เพื่อกำหนดค่าให้กับ CCPR1(CCPR1L:CCP1CON<5:4>)

2.1). หาดิวตี้ไซเคิลของ PWM

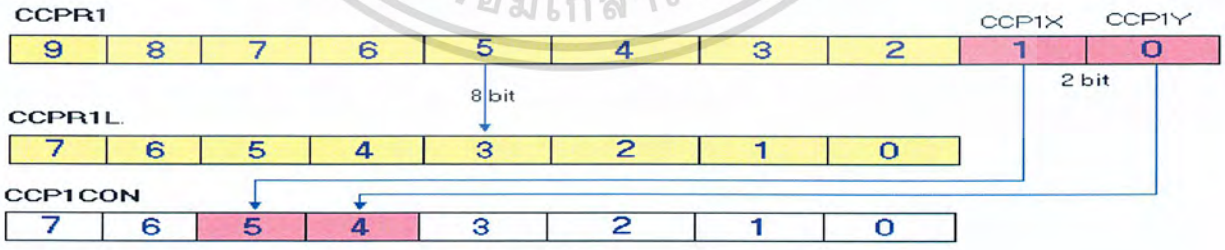
$$D_{PWM} = \frac{d \times T}{100}$$

d= percent-duty cycle , T = = คาบเวลาของพัลส์

2.2).หาค่าที่จะโปรแกรมลงในรีจิสเตอร์ **CCPR1L:CCP1CON<5:4>** จาก

$$(CCPR1L:CCP1CON<5:4>) = \frac{D_{PWM}}{T_{osc} \times P}$$

2.3) ค่าที่คำนวนได้ CCPR1L:CCP1CON<5:4> ซึ่งมีขนาด 10 บิท โดยจะต้องเก็บไว้ใน CCPR1L= 8 บิท และ CCP1X: CCP1Y 2 บิทขวาสุดจะอยู่ในรีจิสเตอร์ CCP1CON<5:4> (บิทที่ 5 และ4 ตามลำดับ)



รูปที่ 3.14 การกำนดค่าให้กับรีจิสเตอร์ CCPR1L:CCP1CON<5:4>

3). กำนดค่าให้กับ <u>T2CON</u>

- TMR2 = 0 ,Clear ค่า TMR2
- T2CON<7> ไม่ใช้ให้=0

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- T2CON<6:3> เป็นการกำหนดค่า POSTScaler ไม่มีผลกำหนดให้เป็น 1111
- T2CON<2> = On/Off Timer2
- T2CON<0:1> กำนด Prescaler

จากข้อ 3 หาก T2CON<2> =1 Timer2 ก็จะเริ่มต้นทำงานทันที ตัวอย่างที่1 การคำนวน เมื่อใช้ XTAL= 4MHz และกำเนิดสัญญาณความถี่ 1KHz ดิวตี้ไซเคิล = 50% ใช้ Prescaler =4

- คาบเวลาของสัญญาณ T = 1/1e3 = 0.001 sec
- คาบเวลาของ XTAL Tosc = 1/4e6 = 2.5e-7 sec
- PR= (0.001 / (4*2.5e-7 * 4)) - 1 = 249
- Dpwm = (50*1e-3)/100 = 5e-4 sec
- CCP1CON CCP1CON<3:0> = (5e-4)/(2.5e-7 * 4) = 500 แปลงเป็นไบนารี 10 บิท = 01111101 00

ตัวอย่างที่2 การคำนวน เมื่อใช้ XTAL= 4MHz และกำเนิดสัญญาณความถี่ 40KHz ดิวตี้ไซเคิล = 50% ใช้ Prescaler =1

- คาบเวลาของสัญญาณ T = 1/40e3 = 2.5e-5 sec
- คาบเวลาของ XTAL Tosc = 1/4e6 = 2.5e-7 sec
- PR= (2.5e-5 / (4*2.5e-7 * 1)) - 1 = 24
- Dpwm = (50* 2.5e-5) / 100 = 1.25e-5 sec
- CCP1CON CCP1CON<3:0> = (1.25e-5) / (2.5e-7 * 1) = 50 แปลงเป็นไบนารี 10 บิท = 00001100 10

ตัวอย่างการกำหนดค่าเริ่มต้นในฟังชั่น

3.5 RS-232

เป็นมาตรฐานการเชื่อมต่อข้อมูลแบบอนุกรมที่มีคนนิยมใช้มากที่สุด กำหนดโดย EIA (Electronics Industry Association) หรือสมาคมผู้ประกอบการอุตสาหกรรมอิเล็กทรอนิกส์ ของอเมริกา ตั้งแต่ปี 1969 โดยมีจุดเริ่มต้นจากความต้องการที่จะกำหนดมาตรฐานการเชื่อมต่อระหว่าง คอมพิวเตอร์กับโมเด็มในสมัยนั้น ตัวมาตรฐานจะกำหนดสิ่งที่เกี่ยวข้องกับการเชื่อมต่อนี้ด้วยกันทั้งหมด 4 หัวข้อหลักๆ ด้วยกันคือ

1.)คุณสมบัติทางไฟฟ้าของสัญญาณ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.) คุณสมบัติทางกลของการเชื่อมต่อ ซึ่งหมายถึงตัวคอนเน็กเตอร์นั่นเอง

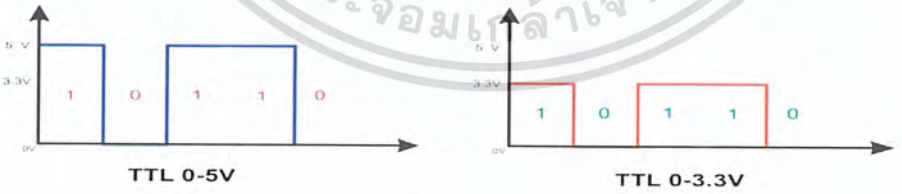
3.)หน้าที่การทำงานของวงจรสำหรับแลกเปลี่ยนข้อมูล

4.) มาตรฐานการเชื่อมต่อสำหรับระบบสื่อสารเฉพาะอย่าง

มาตรฐาน RS-232-C เป็นมาตรฐาน RS-232 ที่มีการปรับปรุงแก้ไขจากมาตรฐานเดิม ซึ่งเราอาจจะคุ้นเคยกับชื่อนี้มากกว่า RS-232-A หรือ RS-232-B อันที่จริงแล้วยังมีมาตรฐาน RS-232-D ที่ใหม่กว่า RS-232-C โดยที่มีการเพิ่มข้อกำหนดของคอนเน็กเตอร์แบบ DB เข้าไปด้วย เช่น DB-25 ซึ่งในขณะนั้นสิทธิบัตรของตัวคอนเน็กเตอร์แบบนี้ได้หมดอายุลงพอดี จึงสามารถรวมข้อกำหนดเข้าไว้ได้ ลักษณะโดยทั่วไปของการเชื่อมต่อข้อมูลแบบอนุกรมตามมาตรฐาน RS-232 คือเป็นการสื่อสารข้อมูลแบบจุดต่อจุด ซึ่งเดิมทีเป็นการสื่อสารข้อมูลระหว่างคอมพิวเตอร์กับโมเด็ม ซึ่งจริงๆแล้วทั้งสองฝั่งจะเป็นอะไรก็ได้ การสื่อสารเป็นแบบสองทางพร้อมกัน (Full-duplex) โดยอาจใช้สายสัญญาณอื่นร่วมเพื่อทำแฮนด์เชค (Hand-shake) หรือไม่ก็ได้ มาตรฐาน RS-232 จำกัดความยาวสายไว้ที่ 50 ฟุต (หรือประมาณ 15 เมตร) สำหรับการส่งสัญญาณที่ความเร็ว 19,200 บิทต่อวินาที โดยที่ความยาวสายจะต้องสั้นลงถ้าต้องการสื่อสารที่ความเร็วสูงขึ้น และถ้ามีสัญญาณรบกวนมากๆ เช่นในโรงงาน หรือบริเวณใกล้เครื่องจักรที่เป็นแบบมีการสวิทช์สัญญาณไฟฟ้าที่กระแสสูงๆ ก็จะทำให้ต้องมีการลดความเร็วในการส่งสัญญาณลงหรือใช้สายที่สั้นลง

TTL (Transistor-Transistor Logic)

TTL เป็นระดับแรงดันที่ถูกกำหนดขึ้นในยุคแรกๆเพื่อใช้ระหว่าง Transistor กับ Transistor ภายในวงจรรวม(IC) ดังนั้น TTL จะใช้ระดับแรงดัน อยู่ที่ 0 – 5 V แต่ในปัจจุบันมีอุปกรณ์หลายเบอร์ที่ทำงานในช่วง 0 – 3.3 V ซึ่งผู้ใช้การตรวจสอบจาก Datasheet ของอุปกรณ์ที่ใช้เสียก่อนว่าเป็นระดับแรงดันแบบใด เพราะหากใช้ผิดประเภทจะทำให้อุปกรณ์เสียหาย



รูปที่ 3.15 ระดับสัญญาณแบบ TTL

UART

UART ย่อมาจากคำว่า Universal Asynchronous Receiver Transmitter ซึ่งหมายถึงอุปกรณ์ที่ทำหน้าที่รับและส่งข้อมูลแบบอะซิงโครนัสนั่นเอง

สำหรับการสื่อสารอนุกรมบนคอมพิวเตอร์แล้ว UART ถือว่าเป็นหัวใจสำคัญของการสื่อสารแบบอนุกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หน้าที่หลักของ UART คือ ทำหน้าที่แปลงข้อมูลที่อยู่ในรูปแบบขนานจากคอมพิวเตอร์ให้อยู่ในรูปแบบอนุ กรมแบบอะซิงโครนัสแล้วส่งออกไป
และทำหน้าที่แปลงสัญญาณอนุกรมแบบอะซิงโครนัสที่ป้อนเข้ามายัง UART ให้เป็นแบบขนาน ก่อนที่จะส่งเข้าคอมพิวเตอร์ ซึ่งนอกจาก UART จะส่งข้อมูล ไปยังคอมพิวเตอร์แล้ว ยังทำการแจ้งข้อมูลอื่นๆ ให้คอมพิวเตอร์ทราบด้วย เช่น อัตราความเร็วในการรับส่งข้อมูล (บอดเรต), รูปแบบการส่งข้อมูล, ความผิดพลาดที่เกิดขึ้นระหว่างการถ่ายทอดข้อมูล (ผิดพลาดจากพาริตี้, เฟรมข้อมูล, โอเวอร์รัน) เป็นต้น

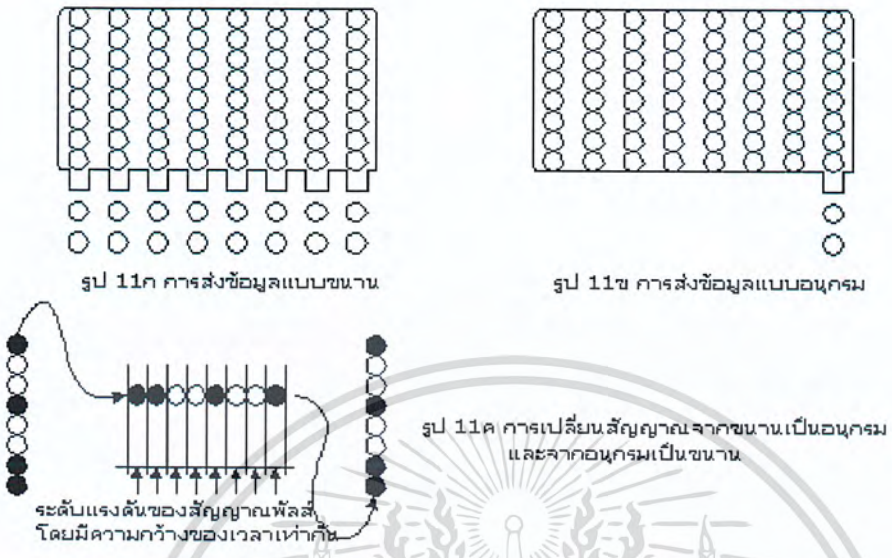
ภายในUARTจะมีส่วนของวงจรสร้างอัตราการถ่ายทอดข้อมูลแบบโปรแกรมได้ (Programmable Buadrate Generator) โดยการกำหนดค่า
ตัวหารให้กับสัญญาณนาฬิกาของ UART โดยตัวหารนี้มีขนาด 16 บิตดังนั้นจึงกำหนดตัวหารให้อยู่ในช่วง 10 – 65,535

UART สามารถรับส่งข้อมูลได้ทั้งแบบฮาล์ฟดูเพล็กซ์ (Half Duplex) และฟูลดูเพล็ก (Full Doplex) โดยการส่งแบบฮาล์ฟดูเพล็กเป็นการส่ง
แบบทิศทางเดียว ส่วนการส่งแบบฟูลดูเพล็กนั้นสามารถรับและส่งข้อมูลได้ในคราวเดียวกัน

ในการสื่อสารผ่านโมเดม จะเป็นการสื่อสารแบบอนุกรม (Serial) คือ โมเดมจัดส่งข้อมูลออกไปทีละบิต หัวใจสำคัญอยู่ที่ชิป UART ซึ่งจะแปลง
ข้อมูลขนาด 8 บิต ที่มีอยู่ ให้ส่งออกไปทีละบิต (อนุกรม) และสามารถส่งผ่านสายโทรศัพท์ได้ โดยอาศัยการแปลง สัญญาณ ด้วยโมเดมนั่นเอง
ใน คอมพิวเตอร์ จะมีพอร์ต อนุกรม ที่เรียกว่าRS-232 ก็จะมีชิป UART คอยควบคุมการรับส่งข้อมูลอยู่แล้ว เช่นเม้าส์ ที่เราใช้กันอยู่ในอดีตก็เชื่อม
กับคอมพิวเตอร์ ทางพอร์ตอนุกรม โดย มี UART คอยทำกับดูแลอยู่ แต่ UART ที่อยู่กับ I/O card มักจะเป็น UART เบอร์ 8250 หรือ 16540 ซึ่งไม่
เหมาะ ที่จะใช้กับโมเดมที่มีความเร็วสูง ๆ เนื่องจากไม่มีบัฟเฟอร์ ของข้อมูล แต่ถ้าคุณใช้โม เดมแบบติดตั้งภายใน (Internal Modem) จะไม่มีปัญหา
เรื่องชิป UART เพราะโมเดมแบบติดตั้งภาย ใน จะมีชิป UART ของตัวมันเองอยู่แล้ว ข้อมูลในไมโครคอนโทรลเลอร์ที่เราใช้ศึกษาอยู่นี้ จะเป็นข้อมูลที่มีความยาวขนาด 1 ไบต์ หรือ 8 บิตซึ่งโดยปกติถ้าเราจะให้ส่งข้อมูลพร้อมๆกันไป 8 บิตจะเป็นวิธีการส่งข้อมูลแบบขนาน แสดง**ได้ดังด้านล่าง**
จะเป็นการส่งข้อมูลขนาด 8 บิตพร้อมกันไปยังอุปกรณ์ภายนอก และจะต้องมีจำนวนของสายสัญญาณจำนวน 8 เส้น เพื่อให้พอดีกับจำนวนของบิตที่ต้องการจะส่ง การส่งข้อมูลแบบขนาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จึงทำให้มีการส่งข้อมูลที่มีความรวดเร็ว แต่ถ้าหากมีการสื่อสารข้อมูลในระยะไกล ก็จะต้องใช้จำนวนของสาย และระยะทางของสายมากขึ้นจึงทำให้มีการสิ้นเปลืองค่าใช้จ่ายสูง



รูปที่ 3.16 แสดงการส่งข้อมูลแบบขนานและแบบอนุกรม

ดังนั้นการสื่อสารข้อมูลแบบอนุกรมจึงถูกนำมาใช้ ในการสื่อสาร โดยจะใช้สายเพียงเส้นเดียวในการส่งข้อมูล หรือรับข้อมูล (คำว่าเส้นเดียวหมายความว่าสายส่ง(TxD) 1 เส้น สายรับ(RxD) 1 เส้น และสายกราวด์ร่วม(Ground) 1 เส้น) นำมาใช้สื่อสารข้อมูลกับอุปกรณ์ภายนอกในระยะทางที่ไกล ดังในรูป 1ข ถ้าหากต้องการส่งข้อมูลขนาด 8 บิต ก็จะทำการส่งข้อมูลออกไปทีละบิตเป็นลำดับไป จนกว่าจะครบจำนวนทั้ง 8 บิต ดังในรูปที่ 3.22 จะแสดงการเปลี่ยนข้อมูลแบบขนานให้เป็นแบบอนุกรม ข้อมูลจะถูกส่งไปตามสายสัญญาณทีละบิตตามจังหวะเวลาที่กำหนดเป็นความกว้างของพัลส์ โดยจังหวะเวลาที่กล่าวนี้จะต้องมีมาตรฐาน ของฝ่ายส่ง และฝ่ายรับด้วย ในการรับสัญญาณที่ส่งมาทีละบิต จะทำการตรวจสอบระดับแรงดันของสัญญาณที่เข้ามาเพื่อแปลงเป็นลอจิก "1" หรือ "0" เมื่อรับข้อมูลเข้ามาครบใน 1 ไบต์ที่กำหนดไว้ ก็จะถูกเปลี่ยนให้อยู่ในรูปแบบของข้อมูลแบบขนานเหมือนเดิม

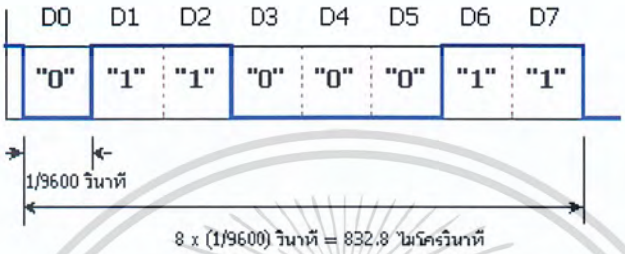
จังหวะเวลาของการสื่อสารข้อมูลอนุกรม

ในการสื่อสารข้อมูลแบบอนุกรม เพื่อรับหรือส่งข้อมูล จะเป็นลักษณะของกลุ่มข้อมูล ดังนั้นอัตราความเร็วจะต้องมีค่าเท่ากันระหว่างการรับและการส่งโดยทั่วไปเราจะระบุความเร็วของจำนวนบิตในการรับและส่งข้อมูล เป็นจำนวนของบิตที่จะส่งใน 1 วินาที โดยเรียกความเร็วในการส่งข้อมูลว่า อัตราบอด(Baud Rate) ซึ่งมีหน่วยเป็นบิตต่อวินาที เช่น 300, 1,200, 2,400, 4,800 และ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

9.600 บิตต่อวินาที ในรูปที่ 3.23 ถ้าหากมีการส่งข้อมูลด้วยความเร็ว 9600 บิตต่อวินาที จะใช้เวลาในการรับส่งข้อมูลหนึ่งบิตมีค่าเท่ากับ 1/9600 หรือ 104.1 ไมโครวินาที และเวลาในการรับส่งข้อมูลทั้ง 8 บิตจะมีค่าเท่ากับ 8 x 104.1 หรือ832.8 ไมโครวินาที

D7	D6	D5	D4	D3	D2	D1	D0
1	1	0	0	0	1	1	0



รูปที่ 3.17 แสดงการส่งข้อมูลแบบอนุกรมด้วยความเร็ว 9600 บิตต่อวินาที

รูปแบบของการสื่อสารข้อมูลอนุกรม

การสื่อสารข้อมูลอนุกรมแบบอะซิงโครนัส เป็นวิธีการรับและส่งข้อมูลโดยไม่ต้องอาศัยสัญญาณนาฬิกาส่งร่วมไปด้วย แต่จะใช้อัตราความเร็วของจำนวนข้อมูลต่อวินาที และจะทำการเพิ่มบิตข้อมูลบางอย่างร่วมไปกับการส่งข้อมูลจริง เพื่อจะได้ทำการตรวจสอบข้อมูลได้อย่างถูกต้องมากยิ่งขึ้นแสดงดังรูปที่ 13 ซึ่งประกอบด้วยกัน 4 ส่วนคือ

1.) บิตเริ่มต้น (Start bit) จะมีขนาด 1 บิต จะเป็นระดับลอจิกตรงกันข้ามกับระดับลอจิกของสภาวะสายสื่อสาร ขณะที่ยังไม่มีการส่งข้อมูล

2.) บิตข้อมูล (Data bit) จะเริ่มจากบิตที่มีนัยสำคัญต่ำสุดก่อนหรือ บิต LSB ก่อน โดยข้อมูลที่จะส่งอาจจะมีขนาด 5 ,6, 7 หรือ 8 บิตก็ได้

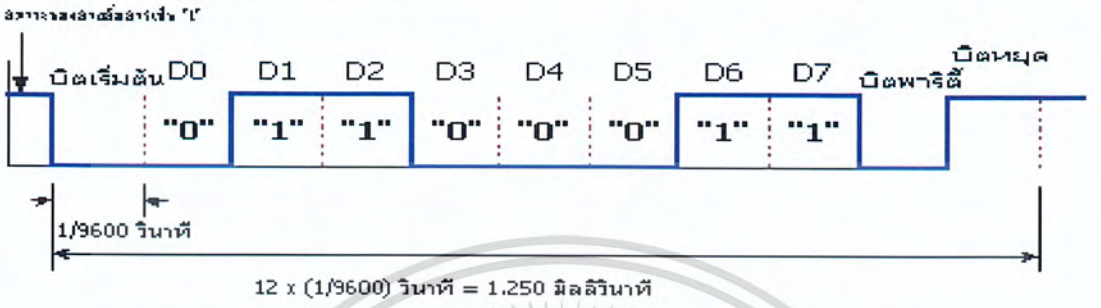
3.) บิตแสดงสภาวะเลขคู่หรือเลขคี่ (Parity bit) มีขนาด 1 บิตโดยบิตนี้จะนำไปต่อท้ายกับบิตข้อมูล ค่าของบิตนี้ขึ้นอยู่กับจำนวนค่าของข้อมูลที่เป็น "1"
โดยเลือกการส่งข้อมูลเป็นแบบ พาริตี้คู่ หรือ พาริตี้คี่ ตัวอย่าง ถ้ากำหนดให้มีการส่งข้อมูลแบบพาริตี้คู่ แต่ข้อมูลมีเลข 1 เป็นจำนวนคี่ ก็จะให้บิตพาริตี้นี้เป็น "1"เพื่อจะได้จำนวนเลข "1" เป็นคู่นั้นเอง ทำนองเดียวกันทางด้านรับเองก็ต้องมีการตรวจสอบจำนวนข้อมูลที่รับเข้ามาเป็น "1" รวมทั้งบิตพาริตี้ 1 บิต ถ้ามีค่า "1"เป็นจำนวนคู่ แสดงว่าข้อมูลที่รับเข้ามาถูกต้อง สามารถกำหนดการรับและส่งข้อมูลเป็นแบบ NONE โดยไม่ต้องมีการตรวจสอบพาริตี้บิตก็ได้

4.) บิตสุดท้ายหรือบิตหยุด (Stop bit) เป็นการระบุถึงขอบเขตของการสิ้นสุดข้อมูล โดยจะทำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ให้ขาข้อมูลมีสถานะ ลอจิกเป็น "1" ซึ่งอาจมีจำนวนมากกว่า หนึ่งบิตก็ได้ เช่น 1 บิต 1.5 บิต หรือ 2 บิต

D7	D6	D5	D4	D3	D2	D1	D0
1	1	0	0	0	1	1	0



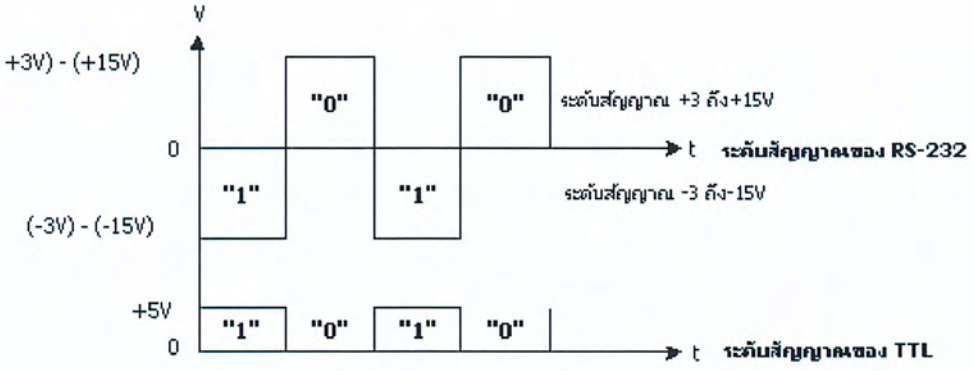
รูปที่ 3.18 **แสดง**การส่งข้อมูลขนาด 8 บิตแบบอนุกรมพร้อมกับบิตเริ่มต้น,บิตพาริตี้,บิตหยุด ด้วยความเร็ว9600 บิตต่อวินาที

การเชื่อมต่อพอร์ตอนุกรมมาตรฐาน RS-232

การกำหนดมาตรฐานการเชื่อมต่อแบบอนุกรม EIA RS-232 (x) เป็นมาตรฐานภายในอุตสาหกรรม โดยคณะกรรมการสมาคมอุตสาหกรรมอิเล็กทรอนิกส์ (Electronic Industries Association) ออกแบบมาเพื่อใช้ในการส่งข้อมูลอนุกรมแบบ อะซิงโครนัส 2 ทิศทางเพื่อให้มีการใช้งาน ในการเชื่อมต่อที่สอดคล้องกัน ระหว่างอุปกรณ์คอมพิวเตอร์ต่างๆ การรับส่งสัญญาณจะกำหนดความยาวสูงสุดไว้ที่ไม่เกิน 50 ฟุตโดยมีระดับ สัญญาณตั้งแต่ 3 โวลท์ จนถึง 15 โวลท์ สำหรับลอจิก "0" และมีระดับแรงดันที่ -3 โวลท์ จนถึง -15 โวลท์ สำหรับลอจิก "1" ดังแสดงในรูป 3.25

ดังนั้นสังเกตได้ว่าจะมีระดับแรงดันที่ใช้ในสถานะลอจิก "0" และ ลอจิก "1" แตกต่างออกไปจากระบบไอซีดิจิตอลทั่วๆไป การต่อใช้งานจึงต้อง
มีอุปกรณ์ที่ทำหน้าที่เปลี่ยนระดับแรงดันจาก 0 - 5 โวลท์ จากไมโครคอนโทรลเลอร์ ให้เป็นระดับแรงดันที่สูงกว่า +3 หรือต่ำกว่า - 3 โดยจะมีไอซีสำเร็จรูปพร้อมใช้งาน หรืออาจจะต่อวงจรจากทรานซิสเตอร์ได้

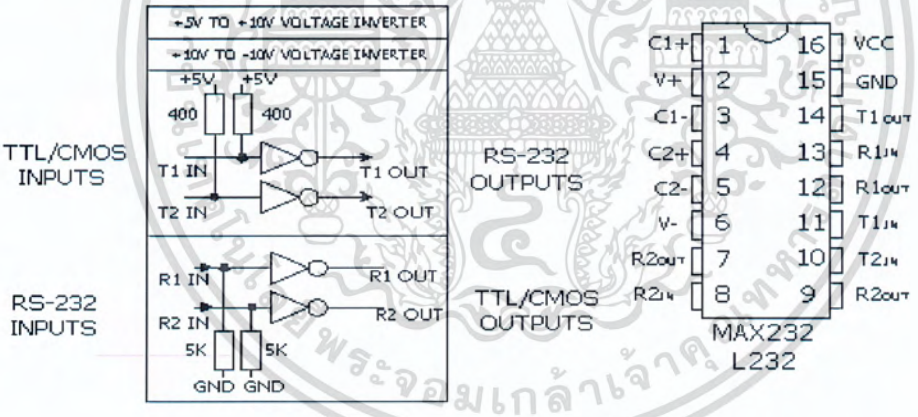
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.19 แสดงระดับแรงดันสัญญาณของพอร์ตอนุกรม RS-232 กับ TTL ในสถานะลอจิก "1" และ "0"

IC MAX232

ไอซี MAX232 เป็นไอซีที่แปลงระดับสัญญาณจากระดับ TTL ไปเป็นระดับของ RS-232 และในทำนองเดียวกันก็รับระดับสัญญาณจาก RS-232เพื่อแปลงเป็นระดับสัญญาณจากระดับ TTL ให้กับไมโครคอนโทรลเลอร์ได้

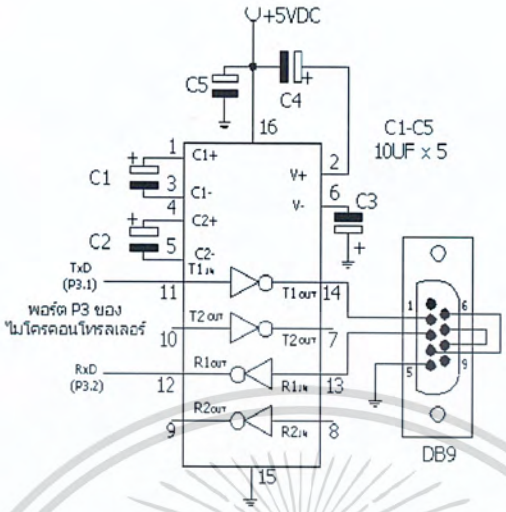


รูปที่ 3.20 ไอซี MAX232

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

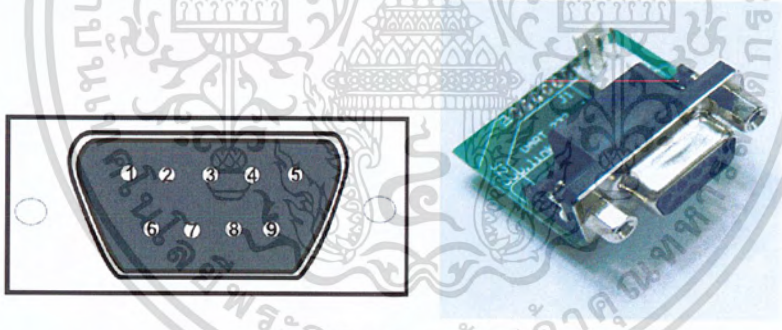
วิธีต่อใช้งาน MAX 3232

วงจรนี้ใช้กับ TTL 0-3.3V เป็น RS 232



รูปที่ 3.21 การต่อMAX232 กับ connecter DB9

ขาของ คอนเน็กเตอร์ DB9



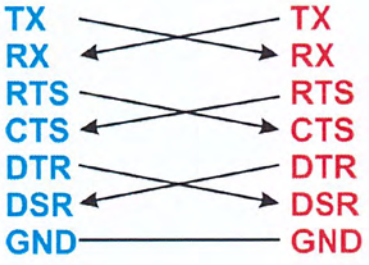
รูปที่ 3.22 ลักษณะคอนเน็กเตอร์แบบDB9

DB-9M	Function	Abbreviation
Pin #1	Data Carrier Detect	CD
Pin #2	Receive Data	RD or RX or RXD
Pin #3	Transmitted Data	TD or TX or TXD
Pin #4	Data Terminal Ready	DTR
Pin #5	Signal Ground	GND
Pin #6	Data Set Ready	DSR
Pin #7	Requests To Send	RTS
Pin #8	Clear To Send	CTS
Pin #9	Ring Indicator	RI

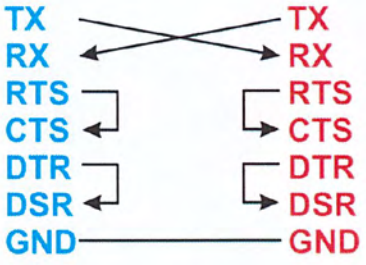
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การเชื่อมต่อสาย DB9

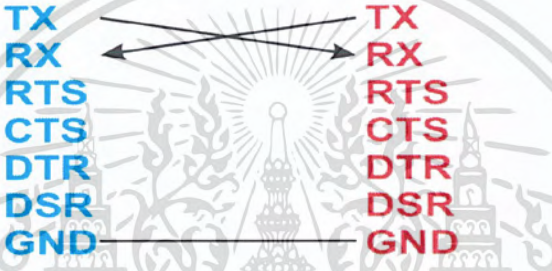
การเชื่อมต่อสาย DB9 โดยทั่วไปแบ่งได้เป็น 3 แบบดังรูป



NULL Modem



3 Wire



3Wire None Flow control

TX = เป็นขาส่งข้อมูล

RX = เป็นขารับข้อมูล

RTS = เป็นขาที่ส่งสถานะไปยังตัวรับ ว่าต้องการส่งข้อมูล เมื่อต้องการส่งข้อมูล จะ ON จนกระทั่งส่ง Data ออกทางขา TX จนเสร็จจึงจะ OFF

CTS = เป็นขาที่รอรับสถานะ จาก RTS ของอุปกรณ์ที่ต่ออยู่ด้วย

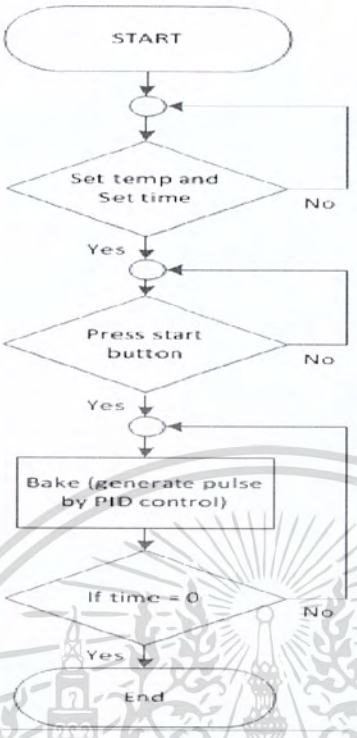
DTR = เป็นขาที่แสดงสถานะว่า Port นั้นเปิดอยู่หรือไม่

DSR = เป็นขาที่ใช้ตรวจเช็ค สถานะ DTR ของอุปกรณ์ที่เชื่อมต่ออยู่ด้วย

GND = Signal Ground

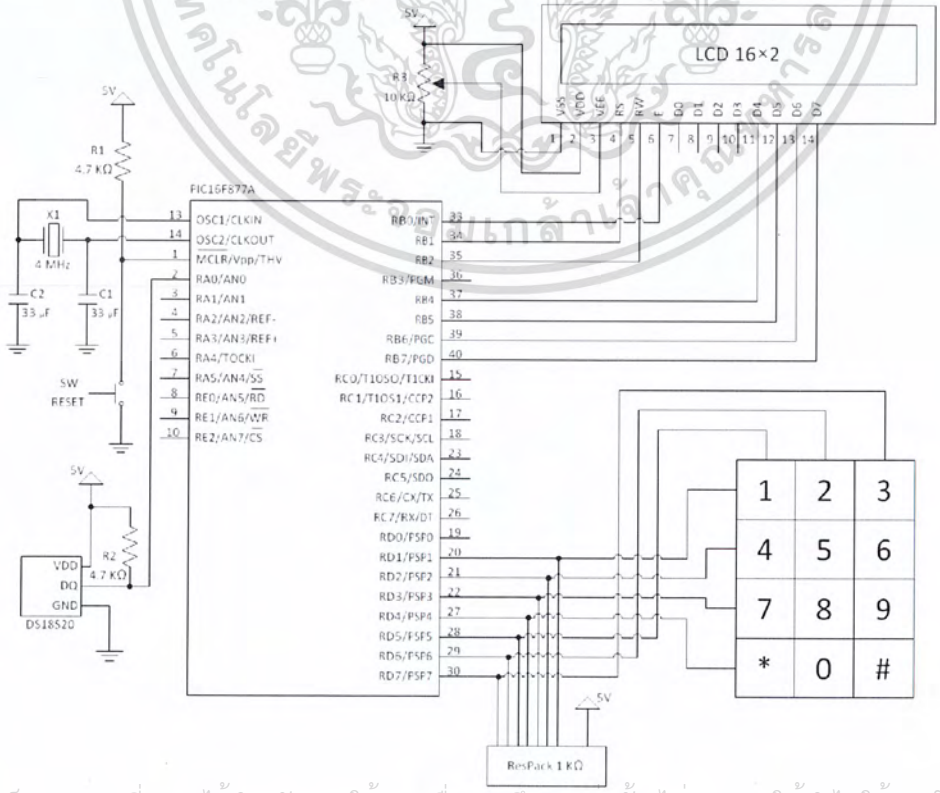
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.6 การทำงานของเตาอบควบคุมอุณหภูมิ



รูปที่ 3.23 Flow Chart การทำงานของเตาอบควบคุมอุณหภูมิ

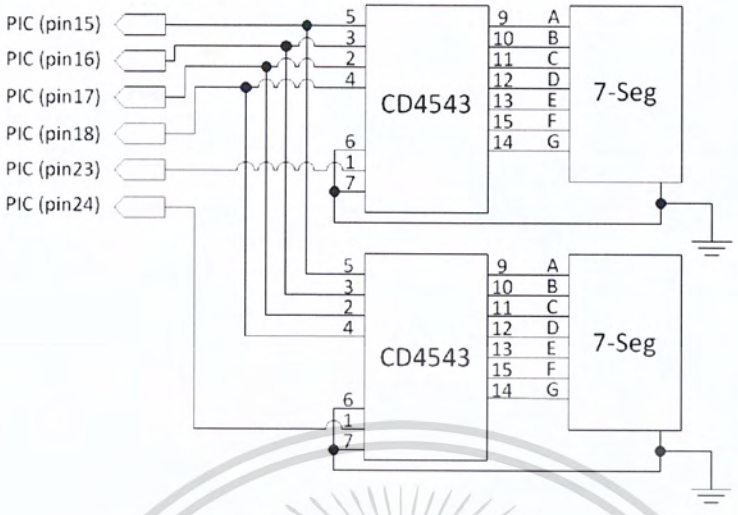
3.7 รูปวงจรส่วนประกอบของเตาอบควบคุมอุณหภูมิ



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

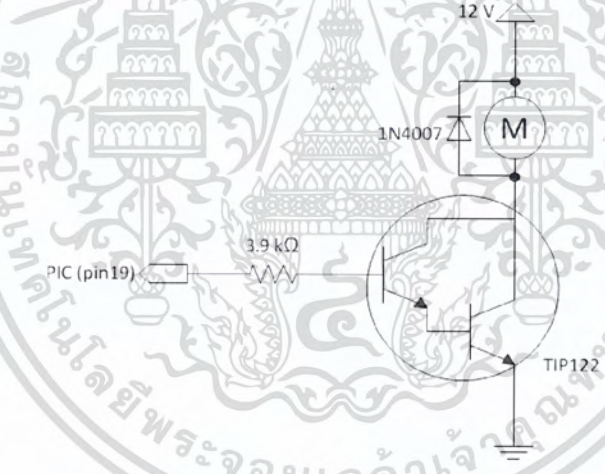
รูปที่ 3.24 ส่วนประกอบวงจรที่ต่อกับไมโครคอนโทรลเลอร์

3.8 รูปส่วนประกอบของวงจรจับเวลา



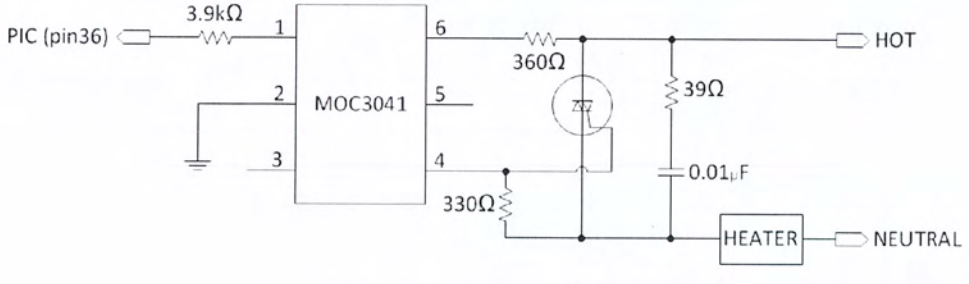
รูปที่ 3.25 วงจรจับเวลา

3.9 รูปส่วนประกอบของวงจรไดรฟ์พัดลม 12V



รูปที่ 3.26 วงจรพัดลม 12 V

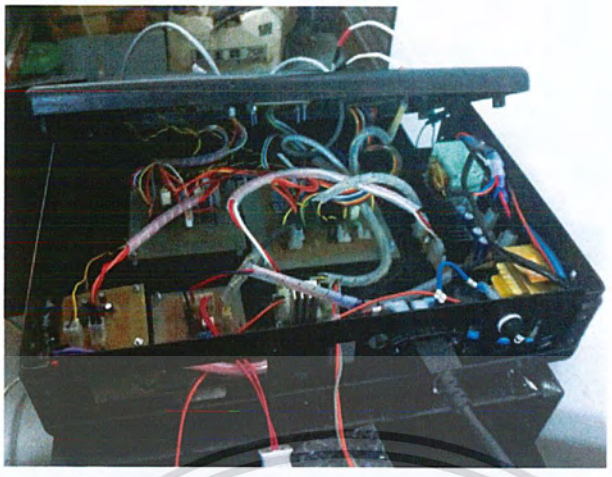
3.10 รูปส่วนประกอบของวงจรไดร์ฟอีตเตอร์



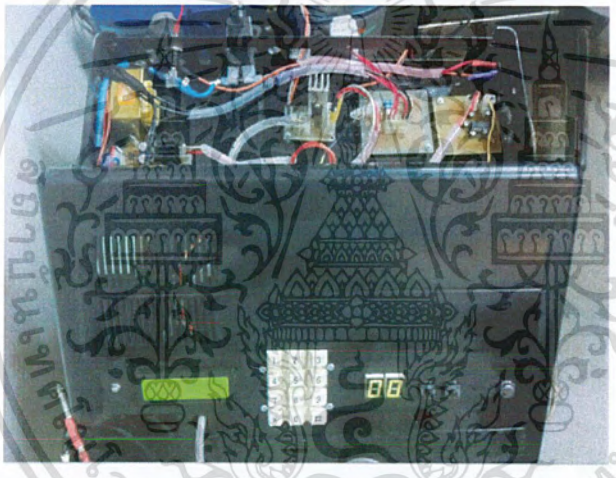
รูปที่ 3.27 วงจรไดร์ฟอีตเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.11 รูปเครื่องควบคุมอุณหภูมิ และเตาอบ



รูปที่ 3.28 เครื่องควบคุมอุณหภูมิ



รูปที่ 3.29 เครื่องควบคุมอุณหภูมิ

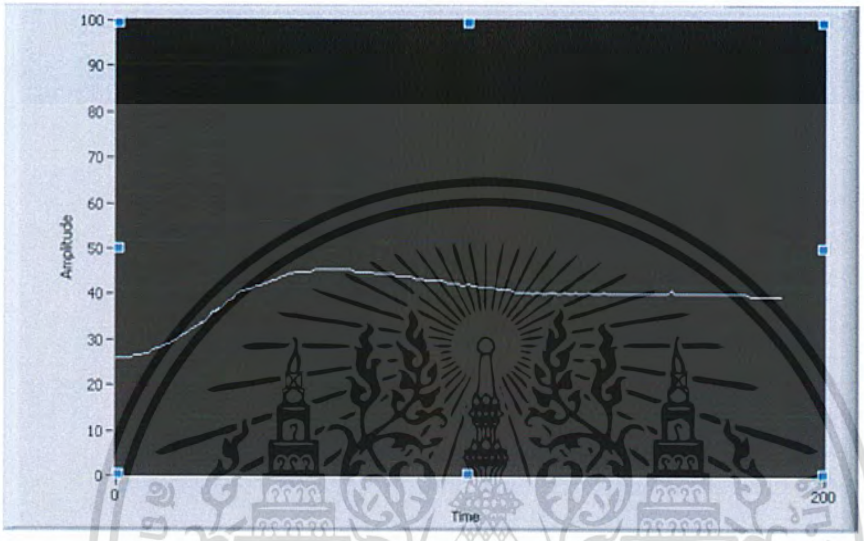
รูปที่ 3.30 เครื่องควบคุมอุณหภูมิและเตาอบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่4

ผลการทดลอง

จากการทดลองเพื่อหาค่าความสัมพันธ์ของกราฟที่อุณหภูมิต่างๆ ดังนี้



รูปที่ 4.1 แสดงTemperature = 40 C , 15 min ,Kp=20, Ki=10, Kd=0

14/3/2011 17:03:30	temp 26.000000	14/3/2011 17:04:40	temp 29.000000
14/3/2011 17:03:35	temp 26.000000	14/3/2011 17:04:45	temp 29.000000
14/3/2011 17:03:40	temp 26.000000	14/3/2011 17:04:50	temp 29.500000
14/3/2011 17:03:45	temp 26.000000	14/3/2011 17:04:55	temp 30.000000
14/3/2011 17:03:50	temp 26.000000	14/3/2011 17:05:00	temp 30.500000
14/3/2011 17:03:55	temp 26.500000	14/3/2011 17:05:05	temp 31.000000
14/3/2011 17:04:00	temp 26.500000	14/3/2011 17:05:10	temp 31.500000
14/3/2011 17:04:05	temp 26.500000	14/3/2011 17:05:15	temp 32.000000
14/3/2011 17:04:10	temp 27.000000	14/3/2011 17:05:20	temp 32.500000
14/3/2011 17:04:15	temp 27.000000	14/3/2011 17:05:25	temp 32.000000
14/3/2011 17:04:20	temp 27.500000	14/3/2011 17:05:30	temp 33.500000
14/3/2011 17:04:25	temp 28.000000	14/3/2011 17:05:35	temp 34.000000
14/3/2011 17:04:30	temp 28.000000	14/3/2011 17:05:40	temp 35.000000
14/3/2011 17:04:35	temp 28.500000	14/3/2011 17:05:45	temp 35.500000

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

14/3/2011 17:05:50	temp 36.000000	14/3/2011 17:08:30	temp 45.500000
14/3/2011 17:05:55	temp 36.500000	14/3/2011 17:08:35	temp 45.500000
14/3/2011 17:06:00	temp 37.000000	14/3/2011 17:08:40	temp 45.500000
14/3/2011 17:06:05	temp 38.000000	14/3/2011 17:08:45	temp 45.500000
14/3/2011 17:06:10	temp 38.500000	14/3/2011 17:08:50	temp 45.500000
14/3/2011 17:06:15	temp 39.000000	14/3/2011 17:08:55	temp 45.500000
14/3/2011 17:06:20	temp 40.000000	14/3/2011 17:09:00	temp 45.500000
14/3/2011 17:06:25	temp 44.500000	14/3/2011 17:09:05	temp 45.000000
14/3/2011 17:06:30	temp 40.500000	14/3/2011 17:09:10	temp 45.000000
14/3/2011 17:06:35	temp 41.000000	14/3/2011 17:09:15	temp 45.000000
14/3/2011 17:06:40	temp 41.000000	14/3/2011 17:09:20	temp 45.000000
14/3/2011 17:06:45	temp 41.500000	14/3/2011 17:09:25	temp 45.000000
14/3/2011 17:06:50	temp 42.000000	14/3/2011 17:09:30	temp 45.000000
14/3/2011 17:06:55	temp 42.000000	14/3/2011 17:09:35	temp 44.500000
14/3/2011 17:07:00	temp 42.500000	14/3/2011 17:09:40	temp 44.500000
14/3/2011 17:07:05	temp 42.500000	14/3/2011 17:09:45	temp 44.500000
14/3/2011 17:07:10	temp 43.000000	14/3/2011 17:09:50	temp 44.500000
14/3/2011 17:07:15	temp 43.000000	14/3/2011 17:09:55	temp 44.500000
14/3/2011 17:07:20	temp 43.500000	14/3/2011 17:10:00	temp 44.500000
14/3/2011 17:07:25	temp 44.000000	14/3/2011 17:10:05	temp 44.000000
14/3/2011 17:07:30	temp 44.000000	14/3/2011 17:10:10	temp 44.000000
14/3/2011 17:07:35	temp 44.500000	14/3/2011 17:10:15	temp 44.000000
14/3/2011 17:07:40	temp 44.500000	14/3/2011 17:10:20	temp 44.000000
14/3/2011 17:07:45	temp 45.000000	14/3/2011 17:10:25	temp 44.000000
14/3/2011 17:07:50	temp 45.000000	14/3/2011 17:10:30	temp 44.000000
14/3/2011 17:07:55	temp 45.000000	14/3/2011 17:10:35	temp 43.500000
14/3/2011 17:08:00	temp 45.000000	14/3/2011 17:10:40	temp 43.500000
14/3/2011 17:08:05	temp 45.000000	14/3/2011 17:10:45	temp 43.500000
14/3/2011 17:08:10	temp 45.500000	14/3/2011 17:10:50	temp 43.000000
14/3/2011 17:08:15	temp 45.500000	14/3/2011 17:10:55	temp 43.000000
14/3/2011 17:08:20	temp 45.500000	14/3/2011 17:11:00	temp 43.000000
14/3/2011 17:08:25	temp 45.500000	14/3/2011 17:11:05	temp 43.000000

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

14/3/2011 17:11:10	temp 43.000000	14/3/2011 17:13:50	temp 40.500000
14/3/2011 17:11:15	temp 43.000000	14/3/2011 17:13:55	temp 40.000000
14/3/2011 17:11:20	temp 42.500000	14/3/2011 17:14:00	temp 40.000000
14/3/2011 17:11:25	temp 42.500000	14/3/2011 17:14:05	temp 40.500000
14/3/2011 17:11:30	temp 42.500000	14/3/2011 17:14:10	temp 40.000000
14/3/2011 17:11:35	temp 42.500000	14/3/2011 17:14:15	temp 40.000000
14/3/2011 17:11:40	temp 42.000000	14/3/2011 17:14:20	temp 40.500000
14/3/2011 17:11:45	temp 42.000000	14/3/2011 17:14:25	temp 40.000000
14/3/2011 17:11:50	temp 42.500000	14/3/2011 17:14:30	temp 40.000000
14/3/2011 17:11:55	temp 42.000000	14/3/2011 17:14:35	temp 40.000000
14/3/2011 17:12:00	temp 42.000000	14/3/2011 17:14:40	temp 40.500000
14/3/2011 17:12:05	temp 41.500000	14/3/2011 17:14:45	temp 40.000000
14/3/2011 17:12:10	temp 41.500000	14/3/2011 17:14:50	temp 40.000000
14/3/2011 17:12:15	temp 41.500000	14/3/2011 17:14:55	temp 40.000000
14/3/2011 17:12:20	temp 41.500000	14/3/2011 17:15:00	temp 40.500000
14/3/2011 17:12:25	temp 41.500000	14/3/2011 17:15:05	temp 40.000000
14/3/2011 17:12:30	temp 41.000000	14/3/2011 17:15:10	temp 40.000000
14/3/2011 17:12:35	temp 41.000000	14/3/2011 17:15:15	temp 40.000000
14/3/2011 17:12:40	temp 41.000000	14/3/2011 17:15:20	temp 40.000000
14/3/2011 17:12:45	temp 41.000000	14/3/2011 17:15:25	temp 40.000000
14/3/2011 17:12:50	temp 41.000000	14/3/2011 17:15:30	temp 40.000000
14/3/2011 17:12:55	temp 40.500000	14/3/2011 17:15:35	temp 40.000000
14/3/2011 17:13:00	temp 40.500000	14/3/2011 17:15:40	temp 40.000000
14/3/2011 17:13:05	temp 40.500000	14/3/2011 17:15:45	temp 40.000000
14/3/2011 17:13:10	temp 40.500000	14/3/2011 17:15:50	temp 40.000000
14/3/2011 17:13:15	temp 40.500000	14/3/2011 17:15:55	temp 40.000000
14/3/2011 17:13:20	temp 40.000000	14/3/2011 17:16:00	temp 40.000000
14/3/2011 17:13:25	temp 40.500000	14/3/2011 17:16:05	temp 40.000000
14/3/2011 17:13:30	temp 40.500000	14/3/2011 17:16:10	temp 40.000000
14/3/2011 17:13:35	temp 40.000000	14/3/2011 17:16:15	temp 40.000000
14/3/2011 17:13:40	temp 40.000000	14/3/2011 17:16:20	temp 40.000000
14/3/2011 17:13:45	temp 40.500000	14/3/2011 17:16:25	temp 40.000000

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

14/3/2011 17:16:30	temp 40.000000	14/3/2011 17:17:55	temp 40.000000
14/3/2011 17:16:35	temp 41.000000	14/3/2011 17:18:00	temp 40.000000
14/3/2011 17:16:40	temp 40.000000	14/3/2011 17:18:06	temp 40.000000
14/3/2011 17:16:45	temp 40.000000	14/3/2011 17:18:11	temp 40.000000
14/3/2011 17:16:50	temp 40.000000	14/3/2011 17:18:16	temp 40.000000
14/3/2011 17:16:55	temp 40.000000	14/3/2011 17:18:21	temp 40.000000
14/3/2011 17:17:00	temp 40.000000	14/3/2011 17:18:26	temp 39.500000
14/3/2011 17:17:05	temp 40.000000	14/3/2011 17:18:31	temp 39.500000
14/3/2011 17:17:10	temp 40.000000	14/3/2011 17:18:36	temp 39.500000
14/3/2011 17:17:15	temp 40.000000	14/3/2011 17:18:41	temp 39.500000
14/3/2011 17:17:20	temp 40.000000	14/3/2011 17:18:46	temp 39.500000
14/3/2011 17:17:25	temp 40.000000	14/3/2011 17:18:51	temp 39.500000
14/3/2011 17:17:30	temp 40.000000	14/3/2011 17:18:56	temp 39.500000
14/3/2011 17:17:35	temp 40.000000	14/3/2011 17:19:01	temp 39.500000
14/3/2011 17:17:40	temp 40.000000	14/3/2011 17:19:06	temp 39.500000
14/3/2011 17:17:45	temp 40.000000	14/3/2011 17:19:11	temp 39.500000
14/3/2011 17:17:50	temp 40.000000		

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 4.2 แสดงTemperature = 50 C , 15 min ,Kp=20, Ki=10, Kd=0

14/3/2011 18:36:06	temp 30.500000	14/3/2011 18:37:41	temp 35.500000
14/3/2011 18:36:11	temp 30.500000	14/3/2011 18:37:46	temp 36.000000
14/3/2011 18:36:16	temp 30.500000	14/3/2011 18:37:51	temp 36.500000
14/3/2011 18:36:21	temp 30.500000	14/3/2011 18:37:56	temp 37.000000
14/3/2011 18:36:26	temp 30.500000	14/3/2011 18:38:01	temp 37.500000
14/3/2011 18:36:31	temp 30.500000	14/3/2011 18:38:06	temp 38.000000
14/3/2011 18:36:36	temp 31.000000	14/3/2011 18:38:11	temp 39.000000
14/3/2011 18:36:41	temp 31.000000	14/3/2011 18:38:16	temp 39.500000
14/3/2011 18:36:46	temp 31.500000	14/3/2011 18:38:21	temp 40.000000
14/3/2011 18:36:51	temp 31.500000	14/3/2011 18:38:26	temp 40.500000
14/3/2011 18:36:56	temp 32.000000	14/3/2011 18:38:31	temp 41.000000
14/3/2011 18:37:01	temp 32.000000	14/3/2011 18:38:36	temp 41.500000
14/3/2011 18:37:06	temp 32.500000	14/3/2011 18:38:41	temp 42.500000
14/3/2011 18:37:11	temp 33.000000	14/3/2011 18:38:46	temp 43.000000
14/3/2011 18:37:16	temp 33.000000	14/3/2011 18:38:51	temp 43.500000
14/3/2011 18:37:21	temp 33.500000	14/3/2011 18:38:56	temp 44.500000
14/3/2011 18:37:26	temp 34.000000	14/3/2011 18:39:01	temp 45.000000
14/3/2011 18:37:31	temp 34.500000	14/3/2011 18:39:06	temp 46.000000
14/3/2011 18:37:36	temp 35.000000	14/3/2011 18:39:11	temp 46.500000

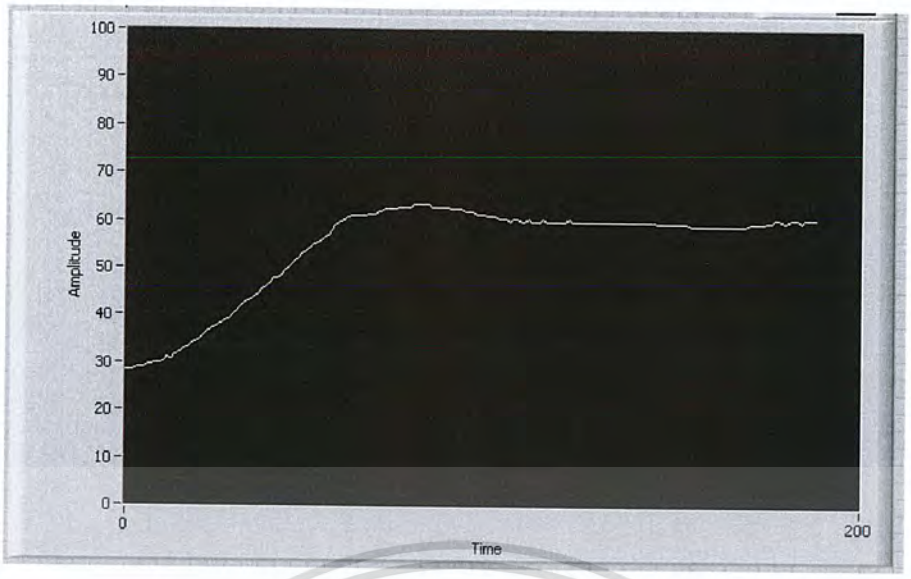
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

14/3/2011 18:39:16	temp 47.000000	14/3/2011 18:41:56	temp 55.000000
14/3/2011 18:39:21	temp 48.000000	14/3/2011 18:42:01	temp 55.000000
14/3/2011 18:39:26	temp 48.500000	14/3/2011 18:42:06	temp 55.000000
14/3/2011 18:39:31	temp 49.000000	14/3/2011 18:42:11	temp 54.500000
14/3/2011 18:39:36	temp 50.000000	14/3/2011 18:42:16	temp 54.500000
14/3/2011 18:39:41	temp 51.000000	14/3/2011 18:42:21	temp 54.500000
14/3/2011 18:39:46	temp 51.000000	14/3/2011 18:42:26	temp 54.500000
14/3/2011 18:39:51	temp 51.500000	14/3/2011 18:42:31	temp 54.000000
14/3/2011 18:39:56	temp 52.500000	14/3/2011 18:42:36	temp 54.000000
14/3/2011 18:40:01	temp 53.000000	14/3/2011 18:42:41	temp 53.500000
14/3/2011 18:40:06	temp 54.000000	14/3/2011 18:42:46	temp 53.500000
14/3/2011 18:40:11	temp 54.500000	14/3/2011 18:42:51	temp 53.500000
14/3/2011 18:40:16	temp 55.500000	14/3/2011 18:42:56	temp 53.000000
14/3/2011 18:40:21	temp 56.000000	14/3/2011 18:43:01	temp 53.000000
14/3/2011 18:40:26	temp 56.000000	14/3/2011 18:43:06	temp 53.000000
14/3/2011 18:40:31	temp 56.000000	14/3/2011 18:43:11	temp 53.000000
14/3/2011 18:40:36	temp 56.000000	14/3/2011 18:43:16	temp 52.500000
14/3/2011 18:40:41	temp 56.000000	14/3/2011 18:43:21	temp 52.500000
14/3/2011 18:40:46	temp 56.000000	14/3/2011 18:43:26	temp 52.000000
14/3/2011 18:40:51	temp 56.000000	14/3/2011 18:43:31	temp 52.000000
14/3/2011 18:40:56	temp 56.000000	14/3/2011 18:43:36	temp 52.000000
14/3/2011 18:41:01	temp 56.000000	14/3/2011 18:43:41	temp 52.000000
14/3/2011 18:41:06	temp 56.000000	14/3/2011 18:43:46	temp 51.500000
14/3/2011 18:41:11	temp 56.000000	14/3/2011 18:43:51	temp 51.500000
14/3/2011 18:41:16	temp 56.000000	14/3/2011 18:43:56	temp 51.500000
14/3/2011 18:41:21	temp 56.000000	14/3/2011 18:44:01	temp 51.000000
14/3/2011 18:41:26	temp 56.000000	14/3/2011 18:44:06	temp 51.000000
14/3/2011 18:41:31	temp 55.500000	14/3/2011 18:44:11	temp 50.500000
14/3/2011 18:41:36	temp 55.500000	14/3/2011 18:44:16	temp 50.500000
14/3/2011 18:41:41	temp 55.500000	14/3/2011 18:44:21	temp 50.500000
14/3/2011 18:41:46	temp 55.500000	14/3/2011 18:44:26	temp 50.500000
14/3/2011 18:41:51	temp 55.500000	14/3/2011 18:44:31	temp 50.000000

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

14/3/2011 18:44:36	temp 50.500000	14/3/2011 18:46:11	temp 50.000000
14/3/2011 18:44:41	temp 50.500000	14/3/2011 18:46:16	temp 50.000000
14/3/2011 18:44:46	temp 50.000000	14/3/2011 18:46:21	temp 50.000000
14/3/2011 18:44:51	temp 50.000000	14/3/2011 18:46:26	temp 50.000000
14/3/2011 18:44:56	temp 50.500000	14/3/2011 18:46:31	temp 50.000000
14/3/2011 18:45:01	temp 50.000000	14/3/2011 18:46:36	temp 50.000000
14/3/2011 18:45:06	temp 50.000000	14/3/2011 18:46:41	temp 50.500000
14/3/2011 18:45:11	temp 50.000000	14/3/2011 18:46:46	temp 50.500000
14/3/2011 18:45:16	temp 50.500000	14/3/2011 18:46:51	temp 50.000000
14/3/2011 18:45:21	temp 50.000000	14/3/2011 18:46:56	temp 50.000000
14/3/2011 18:45:26	temp 50.000000	14/3/2011 18:47:01	temp 50.000000
14/3/2011 18:45:31	temp 50.000000	14/3/2011 18:47:06	temp 50.000000
14/3/2011 18:45:36	temp 50.000000	14/3/2011 18:47:11	temp 50.000000
14/3/2011 18:45:41	temp 50.500000	14/3/2011 18:47:16	temp 50.000000
14/3/2011 18:45:46	temp 50.000000	14/3/2011 18:47:21	temp 50.000000
14/3/2011 18:45:51	temp 50.000000	14/3/2011 18:47:26	temp 50.000000
14/3/2011 18:45:56	temp 50.000000	14/3/2011 18:47:31	temp 50.000000
14/3/2011 18:46:01	temp 50.000000		
14/3/2011 18:46:06	temp 50.000000		

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.3 แสดงTemperature = 60 C , 15 min ,Kp=20, Ki=10, Kd=0

14/3/2011 19:09:36	temp 28.500000	14/3/2011 19:11:16	temp 35.000000
14/3/2011 19:09:41	temp 28.500000	14/3/2011 19:11:21	temp 35.500000
14/3/2011 19:09:46	temp 28.500000	14/3/2011 19:11:26	temp 36.500000
14/3/2011 19:09:51	temp 29.000000	14/3/2011 19:11:31	temp 37.000000
14/3/2011 19:09:56	temp 29.000000	14/3/2011 19:11:36	temp 37.500000
14/3/2011 19:10:01	temp 29.000000	14/3/2011 19:11:41	temp 38.000000
14/3/2011 19:10:06	temp 29.500000	14/3/2011 19:11:46	temp 38.500000
14/3/2011 19:10:11	temp 29.500000	14/3/2011 19:11:51	temp 39.000000
14/3/2011 19:10:16	temp 30.000000	14/3/2011 19:11:56	temp 39.500000
14/3/2011 19:10:21	temp 30.000000	14/3/2011 19:12:01	temp 40.000000
14/3/2011 19:10:26	temp 30.500000	14/3/2011 19:12:06	temp 41.000000
14/3/2011 19:10:31	temp 31.000000	14/3/2011 19:12:11	temp 42.000000
14/3/2011 19:10:36	temp 31.000000	14/3/2011 19:12:16	temp 42.500000
14/3/2011 19:10:41	temp 31.500000	14/3/2011 19:12:21	temp 43.000000
14/3/2011 19:10:46	temp 32.000000	14/3/2011 19:12:26	temp 43.500000
14/3/2011 19:10:51	temp 32.500000	14/3/2011 19:12:31	temp 44.000000
14/3/2011 19:10:56	temp 33.000000	14/3/2011 19:12:36	temp 44.500000
14/3/2011 19:11:01	temp 33.500000	14/3/2011 19:12:41	temp 45.500000
14/3/2011 19:11:06	temp 34.000000	14/3/2011 19:12:46	temp 46.000000
14/3/2011 19:11:11	temp 34.500000	14/3/2011 19:12:51	temp 47.000000

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

14/3/2011 19:12:56	temp 48.000000	14/3/2011 19:15:37	temp 62.500000
14/3/2011 19:13:01	temp 48.000000	14/3/2011 19:15:42	temp 62.500000
14/3/2011 19:13:06	temp 48.500000	14/3/2011 19:15:47	temp 62.500000
14/3/2011 19:13:11	temp 49.500000	14/3/2011 19:15:52	temp 63.000000
14/3/2011 19:13:16	temp 50.000000	14/3/2011 19:15:57	temp 63.000000
14/3/2011 19:13:21	temp 51.000000	14/3/2011 19:16:02	temp 63.000000
14/3/2011 19:13:26	temp 51.500000	14/3/2011 19:16:07	temp 63.000000
14/3/2011 19:13:31	temp 52.000000	14/3/2011 19:16:12	temp 63.500000
14/3/2011 19:13:36	temp 53.000000	14/3/2011 19:16:17	temp 63.500000
14/3/2011 19:13:41	temp 53.500000	14/3/2011 19:16:22	temp 63.500000
14/3/2011 19:13:46	temp 54.500000	14/3/2011 19:16:27	temp 63.500000
14/3/2011 19:13:51	temp 55.000000	14/3/2011 19:16:32	temp 63.500000
14/3/2011 19:13:56	temp 55.500000	14/3/2011 19:16:37	temp 63.000000
14/3/2011 19:14:01	temp 56.000000	14/3/2011 19:16:42	temp 63.000000
14/3/2011 19:14:06	temp 56.500000	14/3/2011 19:16:47	temp 63.000000
14/3/2011 19:14:11	temp 57.000000	14/3/2011 19:16:52	temp 63.000000
14/3/2011 19:14:16	temp 58.000000	14/3/2011 19:16:57	temp 63.000000
14/3/2011 19:14:21	temp 59.000000	14/3/2011 19:17:02	temp 62.500000
14/3/2011 19:14:26	temp 59.500000	14/3/2011 19:17:07	temp 62.500000
14/3/2011 19:14:31	temp 60.000000	14/3/2011 19:17:12	temp 62.500000
14/3/2011 19:14:36	temp 60.500000	14/3/2011 19:17:17	temp 62.500000
14/3/2011 19:14:41	temp 61.000000	14/3/2011 19:17:22	temp 62.000000
14/3/2011 19:14:46	temp 61.000000	14/3/2011 19:17:27	temp 62.000000
14/3/2011 19:14:51	temp 61.000000	14/3/2011 19:17:32	temp 62.000000
14/3/2011 19:14:56	temp 61.000000	14/3/2011 19:17:37	temp 61.500000
14/3/2011 19:15:02	temp 61.000000	14/3/2011 19:17:42	temp 61.500000
14/3/2011 19:15:07	temp 61.500000	14/3/2011 19:17:47	temp 61.500000
14/3/2011 19:15:12	temp 61.500000	14/3/2011 19:17:52	temp 61.000000
14/3/2011 19:15:17	temp 61.500000	14/3/2011 19:17:57	temp 61.000000
14/3/2011 19:15:22	temp 62.000000	14/3/2011 19:18:02	temp 61.000000
14/3/2011 19:15:27	temp 62.000000	14/3/2011 19:18:07	temp 60.500000
14/3/2011 19:15:32	temp 62.500000	14/3/2011 19:18:12	temp 60.500000

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

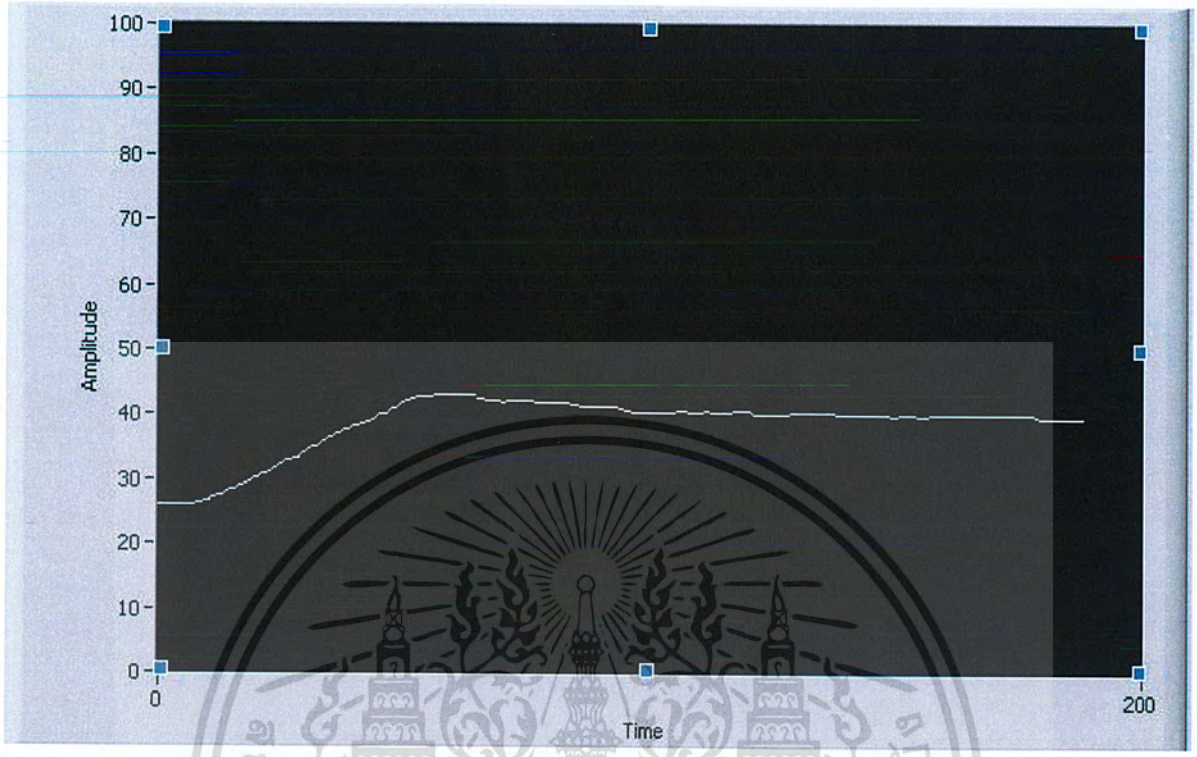
14/3/2011 19:18:17	temp 60.500000	14/3/2011 19:20:57	temp 60.000000
14/3/2011 19:18:22	temp 60.000000	14/3/2011 19:21:02	temp 60.000000
14/3/2011 19:18:27	temp 60.500000	14/3/2011 19:21:07	temp 60.000000
14/3/2011 19:18:32	temp 60.500000	14/3/2011 19:21:12	temp 60.000000
14/3/2011 19:18:37	temp 60.000000	14/3/2011 19:21:17	temp 60.000000
14/3/2011 19:18:42	temp 60.000000	14/3/2011 19:21:22	temp 60.000000
14/3/2011 19:18:47	temp 60.500000	14/3/2011 19:21:27	temp 60.000000
14/3/2011 19:18:52	temp 60.000000	14/3/2011 19:21:32	temp 60.000000
14/3/2011 19:18:57	temp 60.000000	14/3/2011 19:21:37	temp 60.000000
14/3/2011 19:19:02	temp 60.000000	14/3/2011 19:21:42	temp 59.500000
14/3/2011 19:19:07	temp 60.500000	14/3/2011 19:21:47	temp 59.500000
14/3/2011 19:19:12	temp 60.000000	14/3/2011 19:21:52	temp 59.500000
14/3/2011 19:19:17	temp 60.000000	14/3/2011 19:21:57	temp 59.500000
14/3/2011 19:19:22	temp 60.000000	14/3/2011 19:22:02	temp 59.500000
14/3/2011 19:19:27	temp 60.000000	14/3/2011 19:22:07	temp 59.500000
14/3/2011 19:19:32	temp 60.000000	14/3/2011 19:22:12	temp 59.500000
14/3/2011 19:19:37	temp 60.000000	14/3/2011 19:22:17	temp 59.500000
14/3/2011 19:19:42	temp 60.500000	14/3/2011 19:22:22	temp 59.500000
14/3/2011 19:19:47	temp 60.000000	14/3/2011 19:22:27	temp 59.000000
14/3/2011 19:19:52	temp 60.000000	14/3/2011 19:22:32	temp 59.000000
14/3/2011 19:19:57	temp 60.000000	14/3/2011 19:22:37	temp 59.000000
14/3/2011 19:20:02	temp 60.000000	14/3/2011 19:22:42	temp 59.000000
14/3/2011 19:20:07	temp 60.000000	14/3/2011 19:22:47	temp 59.000000
14/3/2011 19:20:12	temp 60.000000	14/3/2011 19:22:52	temp 59.000000
14/3/2011 19:20:17	temp 60.000000	14/3/2011 19:22:57	temp 59.000000
14/3/2011 19:20:22	temp 60.000000	14/3/2011 19:23:02	temp 59.000000
14/3/2011 19:20:27	temp 60.000000	14/3/2011 19:23:07	temp 59.000000
14/3/2011 19:20:32	temp 60.000000	14/3/2011 19:23:12	temp 59.000000
14/3/2011 19:20:37	temp 60.000000	14/3/2011 19:23:17	temp 59.000000
14/3/2011 19:20:42	temp 60.000000	14/3/2011 19:23:22	temp 59.000000
14/3/2011 19:20:47	temp 60.000000	14/3/2011 19:23:27	temp 59.000000
14/3/2011 19:20:52	temp 60.000000	14/3/2011 19:23:32	temp 59.000000

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

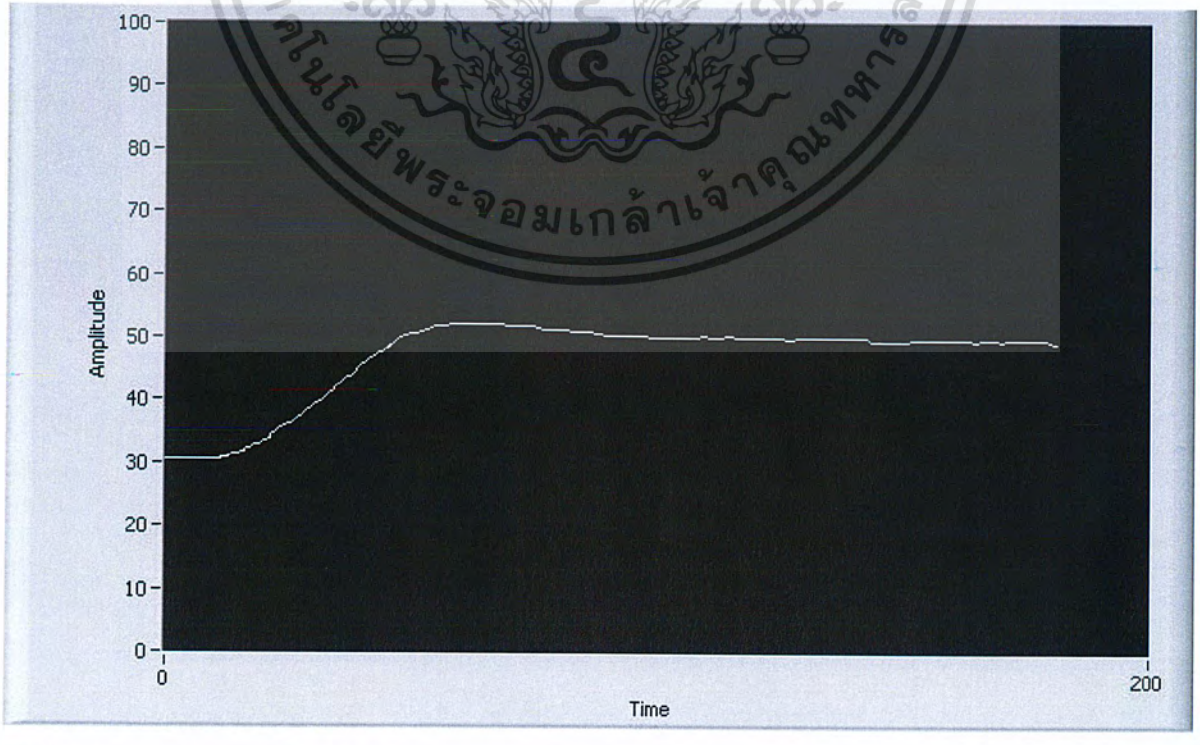
14/3/2011 19:23:37	temp 59.000000	14/3/2011 19:24:32	temp 60.000000
14/3/2011 19:23:42	temp 59.000000	14/3/2011 19:24:37	temp 60.000000
14/3/2011 19:23:47	temp 59.500000	14/3/2011 19:24:42	temp 60.500000
14/3/2011 19:23:52	temp 59.500000	14/3/2011 19:24:47	temp 60.500000
14/3/2011 19:23:57	temp 59.500000	14/3/2011 19:24:52	temp 60.500000
14/3/2011 19:24:02	temp 59.500000	14/3/2011 19:24:57	temp 60.000000
14/3/2011 19:24:07	temp 59.500000	14/3/2011 19:25:02	temp 60.500000
14/3/2011 19:24:12	temp 60.000000	14/3/2011 19:25:07	temp 60.500000
14/3/2011 19:24:17	temp 60.000000	14/3/2011 19:25:12	temp 60.500000
14/3/2011 19:24:22	temp 60.500000	14/3/2011 19:25:17	temp 60.500000
14/3/2011 19:24:27	temp 60.500000		

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ลักษณะของกราฟเมื่อเริ่มทำการจูน (ให้ค่า Kp = 10, Ki =0) ที่อุณหภูมิต่างๆ

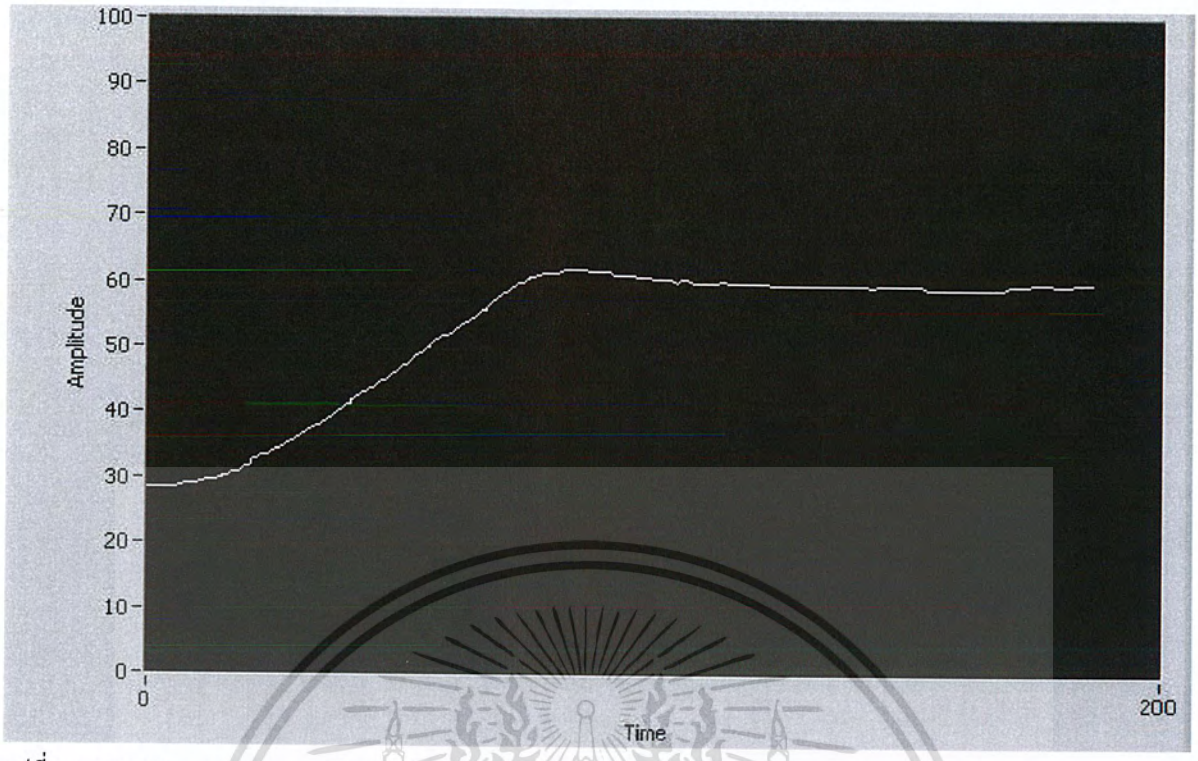


รูปที่ 4.4 แสดง Temperature = 40 C , 15 min ,Kp=10, Ki=0, Kd=0



รูปที่ 4.5 แสดงTemperature = 50 C , 15 min ,Kp=10, Ki=0, Kd=0

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.6 แสดงTemperature = 60 C , 15 min ,Kp=10, Ki=0, Kd=0

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

สรุปผลการทดลองและข้อเสนอแนะ

1.)จากการทดลองเตาอบควยคุมอุณหภูมิสามารถควบคุมอุณหภูมิให้อยู่ในช่วงที่ต้องการได้โดยมีค่าที่Steady State เปลี่ยนแปลงอยู่ในช่วง+-2% ซึ่งเป็นค่าที่สามารถยอมรับได้ โดยการค่าความเร็วในการเข้าสู่Steady State , Overshoot , การแกว่งเพราะค่าError ที่ Steady State และRise time มีค่าขึ้นอยู่กับค่า K_p ,K_i , K_d ในส่วนข้องสมการPID

2.) จากการทดลองเมื่อนำวิธีการควบคุมระบบแบบ PID Control ทำให้สามารถควบคุมอุณหภูมิได้ตามต้องการ แต่ต้องมีการปรับจูนค่าต่างๆ ได้แก่ K_p ,K_i , K_d เพื่อให้เหมาะสมกับระบบของตู้อบที่ออกแบบไว้ โดยหลังจากเพิ่มค่า Ki แล้วทำให้ระบบเข้าสู่ setpoint ได้เร็วแต่ผลที่ได้รับจะทำให้เกิดค่าovershootที่มากขึ้น

3.) จากการทดลองเราสามารถควบคุมอุณหภูมิได้โดยการจ่ายไฟให้แก่ Heater ปริมาณเฉลี่ยของไฟที่จ่ายให้กับ Heater ขึ้นอยู่กับความกว้างของPWMที่เราได้มาจากการคำณวนค่าสมการPID

4.) การทดลองนี้เป็นการทดลองในระบบปิดทำให้ควบคุมอุณหภูมิให้ลดลงได้ยากเนื่องจากการสะสมของพลังงานความร้อนในระบบปิด จึงต้องมีการนำพัดลมมาช่วยในการระบายอาการร้อนออกจากเตาอบเมื่ออุณหภูมิที่วัดได้มีค่าสูงก่อนกว่าค่าที่ต้องการ

5.) จากการทดลองได้นำการสื่อสารข้อมูลแบบ Serial Portโดยใช้รูปแบบการส่งสัญญาณแบบ RS232 เพื่อส่งข้อมูลที่ได้จากการวัดอุณหภูมิของไมโครคอนโทรลเลอร์ ไปยังคอมพิวเตอร์โดยใช้โปรแกรม Labviewในการเก็บค่าและแสดงผลออกจากกราฟ ทำให้สามารถที่จะตรวจสอบอุณหภูมิของเตาอบได้อย่างมีประสิทธิภาพ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก

source code

```
#include <16F877.H>
#fuses HS,NOWDT,NOPUT,NOPROTECT
#use delay(clock =20000000)
#define PWM_PIN  PIN_B3
#define FAN_PIN  PIN_D0
#define LOOPCNT 1000
#define pid pid1
#define use_portb_lcd
#include <lcd.c>
#ifndef TOUCH_PIN
#define TOUCH_PIN  PIN_A0
#endif
#use rs232 (baud=9600, parity=N, xmit=PIN_C6,rcv=PIN_C7, bits=8)
#bit  row1 = 0x08.1
#bit  row2 = 0x08.2
#bit  row3 = 0x08.3
#bit  row4 = 0x08.4
#bit  col1 = 0x08.5
#bit  col2 = 0x08.6
#bit  col3 = 0x08.7
//*****************************Variable*****************************/
signed int digit0,digit1,min=0;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
long int loop = LOOPCNT,pulse,width;
unsigned char temp_lsb,half,temp_buff;
float DS18S20_temp,
    DS18S20_temp_old ;
float error, old_error, sumerror;

float kp = 20.0,          // PID parameter  20.9,30
    ki = 10.0,          // Characteristic of pid parameter is [pid = kp*error + ki*sumerror
+ kd*d_error]   0.37,8.0
    kd = 0;

float d_error;
float pid_max = 1000,
    pid_min = 5  ;
float pid,ki_error,oldki_error;
long int aaa;
int  sp ;
int16 overflow=0;
void chkdigit();
void show7seg();
int data;
long int j=0;
//*****************************SCANKEYPAD*****************************/
void kbd_init()                  // scan column
{
  col1 = 1; col2 = 1; col3 = 1;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
}
int kbd_getc()                  // get parameter from keypad
{
  col1 = 0;
  if(row1 == 0)
  { while(row1==0);
    col1 = 1;
    return('1');   }
  if(row2 == 0)
  { while(row2==0);
    col1 = 1;
    return('4');   }
  if(row3 == 0)
  { while(row3==0);
    col1 = 1;
    return('7');   }
  if(row4 == 0)
  { while(row4==0);
    col1 = 1;
    return('*');   }
  col1 = 1; col2 = 0;
  if(row1 == 0)
  { while(row1==0);
    col2 = 1;
    return('2');   }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
if(row2 == 0)
{ while(row2==0);
  col2 = 1;
  return('5');  }
if(row3 == 0)
{ while(row3==0);
  col2 = 1;
  return('8');  }
if(row4 == 0)
{ while(row4==0);
  col2 = 1;
  return('0');  }
col2 = 1; col3 = 0;
if(row1 == 0)
{ while(row1==0);
  col3 = 1;
  return('3');  }
if(row2 == 0)
{ while(row2==0);
  col3 = 1;
  return('6');  }
if(row3 == 0)
{ while(row3==0);
  col3 = 1;
  return('9');  }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  if(row4 == 0)
  { while(row4==0);
    col3 = 1;
    return('#');   }
  col3 = 1;
  return(0);
}
//*****************************1-wire*****************************/
BYTE touch_read_byte()
{
  BYTE i,data;

  for(i=1;i<=8;++i)
  {
    output_low(TOUCH_PIN);
    delay_us(14);
    output_float(TOUCH_PIN);
    delay_us(5);
    shift_right(&data,1,input(TOUCH_PIN));
    delay_us(100);
  }
  return(data);
}

BYTE touch_write_byte(BYTE data) {
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  BYTE i;

  for(i=1;i<=8;++i) {
   output_low(TOUCH_PIN);
   delay_us(10);
   if(shift_right(&data,1,0)) {
    output_high(TOUCH_PIN);
    delay_us(10);
    if(!input_state(TOUCH_PIN))
     return(0);
   } else {
    output_low(TOUCH_PIN);
    delay_us(10);
    if(input_state(TOUCH_PIN))
       return(0);
   }
   delay_us(50);
   output_high(TOUCH_PIN);
   delay_us(50);
  }
  return(TRUE);
}
BYTE touch_present()
{
   BOOLEAN present;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    output_low(TOUCH_PIN);
    delay_us(500);
    output_float(TOUCH_PIN);

    delay_us(5);

    if(!input(TOUCH_PIN))
     return(FALSE);
    delay_us(65);
    present=!input(TOUCH_PIN);
    delay_us(240);
    if(present)
     return(TRUE);
    else
     return(FALSE);
}

//*************READ&CONVERT TEMPERATURE FROM DS18S20 *************/

byte ReadTemp_DS1820()
{
  byte i,buffer[9];
  if(touch_present())
  {
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    touch_write_byte(0xcc);
    touch_write_byte(0x44);
    delay_ms(300);
    touch_present();
    touch_write_byte(0xcc);
    touch_write_byte(0xbe);
    for(i=0;i<9;i++) buffer[i] = touch_read_byte();
    temp_lsb = buffer[0];
    half = temp_lsb & 0x01;
    DS18S20_temp = (float)buffer[0]/2;
    return(TRUE);
  }
  return(FALSE);
}
//************************SHOW TEMPERATURE************************/
void temp_ds(){

  lcd_send_byte(0,0x80);
  if(ReadTemp_DS1820())
  {
    if(temp_buff!=temp_lsb)
    {
      temp_buff = temp_lsb;
      lcd_send_byte(0,0x80);
      if(j==0)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
      {
        printf(lcd_putc,"Temp: %3.1f %cC ",DS18S20_temp,0xDF);
      }
      if(j!=0)
      {
        if ( DS18S20_temp
<=DS18S20_temp_old+3&&DS18S20_temp>=DS18S20_temp_old-3)    // Protect
code from error of interrupt sensor
          {
            printf(lcd_putc,"Temp: %3.1f %cC ",DS18S20_temp,0xDF);
          }
      }
    }
  }
  //else lcd_putc("Error");
}
//****************SHOW TEMPERATURE SET VALUE**********************/
void ShowTempSet()
{
  int data;
  lcd_send_byte(0,0xc0);
  lcd_putc("Temp set : ");
  lcd_send_byte(0,0xcb);
  data = read_eeprom(0);
  printf(lcd_putc,"%d%cC ",data,0xDF);
  sp = data ;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
}
//* ************MODE  SETTING TEMPERATURE ***************************/
void set_temp()
{
unsigned char addres,temp_num[5],A,key;
unsigned int16 delay;
lcd_send_byte(0,0xc0);
lcd_putc("SET-->  ");
A = 0;
addres = 0xcb;
delay = 1000;
do
{
key=kbd_getc();
  if((key!='*')&(key!=0))
  {
     temp_ds();
     lcd_send_byte(0,0xc0);
     temp_num[A]=key;
     lcd_send_byte(0,addres);
     lcd_putc(key);
     addres++;
     A++;
     delay--;
     delay_ms(10);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    delay=1000;
  }
  delay_ms(10);
  delay--;
}
  while((key!='#')&(A<4)&(delay!=0));
  lcd_send_byte(0,0xcb);
  if((key=='#')&(A<5))
  {
  switch (A)
  {
      case 2 : temp_num[3] = temp_num[0] & 0x0f;
               break;
      case 3 : temp_num[0] = (temp_num[0] & 0x0f) * 10;
               temp_num[1] = temp_num[1] & 0x0f;
               temp_num[3] = temp_num[0] + temp_num[1];
               break;
      case 4 : temp_num[0] = (temp_num[0] & 0x0f) * 100;
               temp_num[1] = (temp_num[1] & 0x0f) * 10;
               temp_num[2] = temp_num[2] & 0x0f;
               temp_num[3] = temp_num[0] + temp_num[1] + temp_num[2];
               break;
    }
    if(temp_num[3]<=125)
    {
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        write_eeprom(0,temp_num[3]);
        lcd_putc("save");
        delay_ms(2000);
      }
      else
      {
        lcd_putc("ERROR");
        delay_ms(2000);
      }
    }
    else
    {
    lcd_putc("OUT");
    delay_ms(2000);
    }
    lcd_send_byte(0,0xc0);
    lcd_putc("              ");
    lcd_send_byte(0,0xc0);
    temp_ds();
}
//*****************************FAN ******************************/
void fan()
{
  if ( DS18S20_temp
<=DS18S20_temp_old+3&&DS18S20_temp>=DS18S20_temp_old-3)
  {
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    if ( sp < DS18S20_temp )
    {
      output_high(FAN_PIN);
    }
    else
    {
      output_low(FAN_PIN);
    }
  }
}
//*****************************Define Digit*****************************/
void chkdigit()
{
  digit0 = min%10;
  digit1 = min/10;
}
//*****************************Show 7-seg *****************************/
void show7seg()
{
  output_c(0x00+digit0);
    output_high(PIN_C5);
    output_low(PIN_C5);
  output_c(0x10+digit1);
    output_high(PIN_C4);
    output_low(PIN_C4);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
}
//*****************************PID calculartr*****************************/
void pid_cal ()
{
        error = sp - DS18S20_temp ;// Start the calculate pid control
        ki_error = ki*error+oldki_error ;

        pid = kp*error + ki_error ;
        if( pid>pid_max)
        {
           ki_error = oldki_error;
           pid = pid_max ;
        }
        else if (pid<pid_min)
        {
          ki_error = oldki_error;
          pid = pid_min ;
        }
        oldki_error = ki_error;
        error = old_error;
}
//*****************************TIMER_PWM*****************************/
#INT_TIMER1
void IntTMR1_isr()      //SET over flow 1 state = 1 ms
{
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
   overflow++;
}
#INT_TIMER2
void isr_timer2()          // SET over flow 4000 state , time total 1 process = 4 sec
{ aaa = pid ;
  if(--loop == 0)
  {
    loop = LOOPCNT;
    pulse = width;
    output_high(PWM_PIN);
    width = aaa;
  }

  if(--pulse == 0)       // SET parameter of PWM from PID parameter
  {
    output_low(PWM_PIN);
  }
}
//************************ MAIN PROGRAM ************************/
void main()
{
  int keypad,i;
  disable_interrupts(INT_timer2);              // disable interrupt timer 2
  enable_interrupts(GLOBAL);
  //setup_timer_2(T2_DIV_BY_4,250,5);          // when use clock osc = 4MHz
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
setup_timer_2(T2_DIV_BY_16,125,10);          // when use clock osc = 20MHZ
disable_interrupts(INT_TIMER1);
setup_timer_1(T1_INTERNAL | T1_DIV_BY_4);
set_timer1(3036);

SET_TRIS_C(0x00);
SET_TRIS_D(0x1F);
SET_TRIS_E(0xFF);

kbd_init();
lcd_init();
ShowTempSet();

for(i=0;i<10;i++)
temp_ds();
start:
SET_TRIS_A(0b000110);
output_low(PWM_PIN);

while(true)
{
  keypad=kbd_getc();
  if(keypad=='*')
  {
    set_temp();
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
}
DS18S20_temp_old = DS18S20_temp ;
ShowTempSet();
temp_ds();
chkdigit();
show7seg();
  if(!input(pin_E0))
  {
     min=min+10;
     if(min>=99)
     {
       min=0;
     }
  }
if(!input(pin_E1))
{
  min=min+1;
   if(min>=99)
     {
       min=0;
     }
}
if(min!=0 && !input(pin_E2))
{
  enable_interrupts(INT_TIMER1);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
enable_interrupts(INT_TIMER2);
output_high(PIN_A2);          // Set LED when put the start buttom
while(min>=0)
{
if(j%5==0)   // Delay to calculate PID
  {
    if ( DS18S20_temp
<=DS18S20_temp_old+3&&DS18S20_temp>=DS18S20_temp_old-3)
      {
        pid_cal() ;
        printf(" %3.1f  \r\n",DS18S20_temp);

        DS18S20_temp_old = DS18S20_temp ;
      }
  }
  J++ ;
  temp_ds();
  ShowTempSet();
  keypad=kbd_getc();
  fan();
  if(keypad=='*')
  {
    set_temp();
  }
////////////////////////////////////////////////////////////////////////////////
  chkdigit();
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        show7seg();
        if(overflow>1200)
        {
          min--;
          overflow=0;
          if(min==0)
          {
            disable_interrupts(INT_timer2);
            output_low(PWM_PIN);
            chkdigit();
            show7seg();
            output_low(PIN_A2);
            output_high(PIN_A1);
            delay_ms(3000);
            overflow=0;
            goto start;
          }
        }
      }
    }

  }
}
```

//

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้