ภาควิชาครุศาสตร์วิศวกรรม

คณะครุศาสตร์อุตสาหกรรม

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ใบรับรองปริญญานิพนธ์

ชื่อหัวข้อ ชุดสาธิตการควบคุมอุณหภูมิภายในห้องจำลองแบบวงปิด

Close Loop Temperature Control In Room Model Demonstrating Set

ชื่อนักศึกษา

1. นายทรงพล	มงคลจีรวัฒน์	รหัสประจำตัว	45035422
2. นายธนานุวัฒน์	สุวรรณรุ่งเรือง	รหัสประจำตัว	45035423
3. นางสาววราภรณ์	สุภาพละ	รหัสประจำตัว	45035433
4. นายสุธา	ทรัพย์อนันต์กุล	รหัสประจำตัว	45035446

หลักสูตร ครุศาสตร์อุตสาหกรรมบัณฑิต **สาขาวิชา** เทคโนโลยีการวัดคุมทางอุตสาหกรรม

อาจารย์ที่ปรึกษา อาจารย์อำพล ทองระอา

อาจารย์ที่ปรึกษาร่วม อาจารย์สุรพงษ์ สิริพงศ์ดี

คณะกรรมการสอบปริญญานิพนธ์		ลายมือชื่อ
1. ดร.สุรสิทธิ์	ราตรี	
2. อาจารย์อำพล	ทองระอา	
3. อาจารย์สุรพงษ์	สิริพงษ์ดี	
4. อาจารย์ปิยะ	ศุภวราสุวัฒน์	
5. อาจารย์สุระชัย	พิมพ์สาลี	

วัน/เดือน/ปีที่สอบ วันพฤหัสบดีที่ 20 พฤศจิกายน พ.ศ. 2546 เวลา 9:00 น.

สถานที่สอบ ห้อง ค.311 คณะครุศาสตร์อุตสาหกรรม สจล.

ภาควิชารับรองแล้ว

ลงนาม.....

(นายสุรสิทธิ์ ราตรี)

หัวหน้าภาควิชาครุศาสตร์วิศวกรรม

วันที่ 31 เดือน พ.ย. พ.ศ. 47

<BT4610432>

ชุดสาธิตการควบคุมอุณหภูมิภายในห้องจำลองแบบวงปิด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

ปริญญานิพนธ์

ชุดสาธิตการควบคุมอุณหภูมิภายในห้องจำลองแบบวงปิด

CLOSE LOOP TEMPERATURE CONTROL IN ROOM MODEL DEMONSTRATING SET

T051071

นายทรงพล มงคลจีรวัฒน์

นายธนานุวัฒน์ สุวรรณรุ่งเรือง

นางสาววราภรณ์ สุภาพละ

นายสุธา ทรัพย์อนันต์กุล

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เลขหมู่.....
เลขทะเบียน 51071
วัน,เดือน,ปี 29 มิ.ย. 2547

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

ปริญญานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรครุศาสตร์อุตสาหกรรมบัณฑิต

สาขาวิชาเทคโนโลยีการวัดคุมทางอุตสาหกรรม

ภาควิชาครุศาสตร์วิศวกรรม คณะครุศาสตร์อุตสาหกรรม

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2546

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์

เรื่อง ชุดสาธิตการควบคุมอุณหภูมิภายในห้องจำลองแบบวงปิด

Close Loop Temperature Control in Room Model Demonstrating Set

วัตถุประสงค์

1. เพื่อศึกษากระบวนการการควบคุมอุณหภูมิภายในห้องจำลองด้วย PID
2. เพื่อออกแบบวงจรชุดควบคุมอุณหภูมิภายในห้องจำลองแบบวงปิดด้วย PID
3. เพื่อสร้างชุดสาธิตการควบคุมอุณหภูมิภายในห้องจำลองแบบวงปิดด้วย PID
4. เพื่อทดลองและทดสอบชุดสาธิตการควบคุมอุณหภูมิภายในห้องจำลองแบบวงปิดด้วย PID
5. เพื่อนำชุดสาธิตการควบคุมอุณหภูมิภายในห้องจำลองแบบวงปิดไปใช้ประกอบการเรียนการสอนได้จริงในวิชา วิศวกรรมระบบและระบบควบคุม (รหัสวิชา 03332308) หลักสูตร ปริญญาตรีต่อเนื่อง 2 ปี ภาควิชาครุศาสตร์วิศวกรรม คณะครุศาสตร์อุตสาหกรรม สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ประโยชน์ที่คาดว่าจะได้รับ

1. ได้ความรู้เกี่ยวกับกระบวนการการควบคุมอุณหภูมิภายในห้องจำลองด้วย PID
2. ได้วงจรชุดควบคุมอุณหภูมิภายในห้องจำลองแบบวงปิดด้วย PID
3. ได้ชุดสาธิตการควบคุมอุณหภูมิภายในห้องจำลองแบบวงปิดด้วย PID
4. ได้ผลการทดลองเกี่ยวกับกระบวนการควบคุมอุณหภูมิภายในห้องจำลองแบบวงปิดด้วย PID
5. ได้สื่อการเรียนการสอนเกี่ยวกับกระบวนการควบคุมแบบ PID

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ชื่อหัวข้อ	ชุดสาธิตการควบคุมอุณหภูมิภายในห้องจำลองแบบวงปิด	
นักศึกษา	นายทรงพล	มงคลจีรวัฒน์
	นายธนานุวัฒน์	สุวรรณรุ่งเรือง
	นางสาววราภรณ์	สุภาพละ
	นายสุธา	ทรัพย์อนันต์กุล
อาจารย์ที่ปรึกษา	อาจารย์อำพล	ทองระอา
อาจารย์ที่ปรึกษาร่วม	อาจารย์สุรพงษ์	สิริพงศ์ดี
หลักสูตร	ครุศาสตร์อุตสาหกรรมบัณฑิต	
สาขาวิชา	เทคโนโลยีการวัดคุมทางอุตสาหกรรม	
ปีการศึกษา	2546	

บทคัดย่อ

ปริญญานิพนธ์ฉบับนี้นำเสนอการออกแบบและสร้างชุดสาธิตการควบคุมอุณหภูมิภายในห้องจำลองแบบวงปิด โดยชุดโปรแกรมควบคุมแบ่งออกเป็น 2 ส่วนด้วยกัน คือ ส่วนที่ 1 เป็นส่วนของภาคแสดงผลทางจอคอมพิวเตอร์แบบกราฟฟิก ส่วนที่ 2 เป็นส่วนของการควบคุมกระบวนการความเร็วมอเตอร์ ฮีตเตอร์ และควบคุมอุณหภูมิด้วยตัวคอนโทรลเลอร์แบบพีไอดี ชุดสาธิตนี้จะช่วยให้เข้าใจหลักการทำงานของการควบคุมอุณหภูมิแบบวงปิดได้ง่ายยิ่งขึ้น สามารถนำไปประยุกต์ใช้งานในระบบควบคุมได้ และเหมาะสำหรับนำไปเป็นสื่อประกอบการเรียนการสอนได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Thesis Title	Close Loop Temperature Control in Room Model Demonstrating Set	
Students	Mr.Songphol	Mongkolgheravat
	Mr.Thananuwat	Suwonrungraung
	Miss.Waraporn	Supapala
	Mr.Sutha	Subanankul
Advisor	Mr.Amphon	Thongra-ar
Co-Advisor	Mr.Surapong	Siripongdee
Education Level	Bachelor of Science in Industrial Education	
Program in	Industrial Instrument Technology	
Academic Year	2003	

ABSTRACT

This thesis is present about the design and build the demonstration of control temperature in the close demonstration room. The control program have two part such as first part display about graphic computer, second part is a demonstration of process of control heater motor and temperature by PID controller. This demonstration you can know the technique of work by control temperature which close You can apply to work in control system. This demonstration appropriate for the media learning.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

ปริญญานิพนธ์ฉบับนี้ลุล่วงไปด้วยดี เนื่องจากความร่วมมือของสมาชิกภายในกลุ่มทุกท่าน ขอขอบคุณอาจารย์ภาควิชาครุศาสตร์วิศวกรรมทุกท่าน ที่ช่วยเหลือให้คำแนะนำ ให้คำปรึกษาเกี่ยวกับโครงงาน และขอขอบคุณทุกท่านที่สั่งสอนอบรมให้วิชาความรู้จนปัจจุบัน ขอขอบคุณกำลังใจจากเพื่อนๆ ทุกคนที่ให้กำลังใจในการทำงาน คอยให้ความช่วยเหลือ แนะนำ ข้อมูลต่างๆ และผู้ที่เอื้อเฟื้อสถานที่ในการทำกิจกรรมหลายอย่าง และที่สำคัญที่สุด คือขอขอบคุณบิดามารดา ผู้ให้กำเนิดให้ชีวิต และให้ทุกสิ่งทุกอย่าง ขอบคุณผู้มีอุปการะคุณทุกท่านที่ให้ความสนับสนุน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

เรื่อง	หน้า
บทคัดย่อภาษาไทย	I
บทคัดย่อภาษาอังกฤษ	II
กิตติกรรมประกาศ	III
สารบัญ	IV
สารบัญตาราง	VIII
สารบัญรูป	IX
บทที่ 1 บทนำ	1
1.1 ความเป็นมาและความสำคัญ	1
1.2 ขีดความสามารถของโครงงาน	1
1.3 เนื้อหาโดยสังเขป	2
บทที่ 2 ทฤษฎีและหลักการ	3
2.1 กล่าวนำ	3
2.2 ชนิดของการควบคุม	4
2.2.1 การควบคุมแบบสัดส่วน	4
2.2.2 การควบคุมแบบรวม	8
2.2.3 การควบคุมแบบอัตรา	11
2.2.4 การควบคุมแบบผสม	12
2.3 การปรับค่า PID	21
2.3.1 ผลการตอบสนองที่ดี	21
2.3.2 ความยากง่ายในการควบคุมโปรเซส	23
2.3.3 ผลของ PID ต่อเสถียรภาพของระบบ	24
2.3.4 คุณภาพของกรควบคุม	24
2.3.5 วิธีการปรับค่า	25
2.4 อุปกรณ์ควบคุม	31

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

เรื่อง	หน้า
2.4.1 RS 485	31
2.4.2 PIC16F877	32
2.4.3 ไอซีตรวจจับอุณหภูมิ DS1820	37
2.4.4 คำสั่งเพื่อควบคุมการทำงานของ DS1820	38
2.5 การถ่ายเทความร้อน	39
2.5.1 การนำ	39
2.5.2 การพา	40
2.5.3 การแผ่รังสี	41
2.5.4 การถ่ายเทความร้อนโดยการพาความร้อนและการแผ่รังสีความร้อนรวมกัน	42
2.6 การวัดอุณหภูมิ	43
2.6.1 ชนิดและวิธีการวัดอุณหภูมิ	43
2.6.2 อุณหภูมิและมาตราวัดอุณหภูมิ	44
2.6.3 สเกลอุณหภูมิสัมบูรณ์	5
2.6.4 สเกลอุณหภูมิเทียบเคียง	46
2.6.5 การสอบเทียบ	46
2.7 มอเตอร์กระแสตรง	47
2.8 PWM (Pulse Width Modulation)	48
2.8.1 รูปคลื่น PWM แบบไซน์	48
2.8.2 โครงงานการควบคุมวงจรอิเล็กทรอนิกส์กำลังด้วยคอมพิวเตอร์	50
2.8.3 การสร้างสัญญาณ PWM ด้วยไมโครคอนโทรลเลอร์	51
2.8.4 ลักษณะของสัญญาณ PWM ที่สร้างจากไมโครคอนเลอร์	53
2.9 ฮีตเตอร์	53
2.9.1 หลักการของฮีตเตอร์	53
2.9.2 ชนิดของฮีตเตอร์	54

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

เรื่อง	หน้า
2.9.3 ฮีตเตอร์ครีบและฮีตเตอร์ท่อกลม	55
2.9.4 การคำนวณวัตต์ของฮีตเตอร์	55
บทที่ 3 การออกแบบ การสร้าง และการทำงาน	57
3.1 การออกแบบโครงสร้าง	57
3.1.1 ออกแบบห้องจำลองควบคุมอุณหภูมิ	58
3.2 วงจรตรวจจับอุณหภูมิ	58
3.3 วงจรไดร์เวอร์	59
3.4 วงจรส่งผ่านข้อมูลแบบอนุกรม	60
3.5 วงจรแหล่งจ่ายแรงดัน	61
3.6 วงจรควบคุมการทำงานของชุดสาธิตการควบคุมอุณหภูมิภายในห้องจำลองแบบวงปิด	62
3.7 การออกแบบโปรแกรมควบคุม	62
3.7.1 โปรแกรมตรวจจับอุณหภูมิ	63
3.7.2 โปรแกรมควบคุมมอเตอร์และฮีตเตอร์ด้วยสัญญาณ PWM	64
3.7.3 โปรแกรมหน้าต่างแสดงผลหลัก	64
3.7.4 โปรแกรมย่อยของกิริยาการควบคุม	64
3.7.5 โปรแกรมย่อยของปุ่มกดควบคุม	64
3.7.6 การออกแบบหน้าต่างควบคุมและแสดงผล	68
บทที่ 4 การทดลองและผลการทดลอง	70
4.1 การทดลองห้องจำลองควบคุมอุณหภูมิ	70
4.1.1 ลำดับขั้นการทดลอง	70
4.2 การทดลองวงจรตรวจจับอุณหภูมิ	71
4.3 การทดลองวงจรไดร์เวอร์	72
4.3.1 ลำดับขั้นการทดลอง	72
4.4 การทดลองวงจรส่งผ่านข้อมูลแบบอนุกรม	77

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

เรื่อง	หน้า
4.5 การทดลองวงจรแหล่งจ่ายแรงดัน	78
4.6 วงจรควบคุมการทำงานของชุดสาธิตการควบคุมอุณหภูมิภายในห้องจำลองแบบวงปิด	78
4.6.1 การทดลองชุดควบคุมโดยใช้ตัวสร้างอุณหภูมิภายนอก	78
4.6.2 การทดลองอุณหภูมิโดยกิริยาควบคุมแบบพี	79
4.6.3 การทดลองอุณหภูมิโดยกิริยาควบคุมแบบพีไอ	80
4.6.4 การทดลองอุณหภูมิโดยกิริยาควบคุมแบบพีดี	82
4.6.5 การทดลองอุณหภูมิโดยกิริยาควบคุมแบบพีไอดี	83
บทที่ 5 บทสรุป	85
5.1 สรุป	85
5.2 ปัญหาและแนวทางแก้ไข	85
5.3 แนวทางการพัฒนา	86
บรรณานุกรม	87
ภาคผนวก ก เครื่องต้นแบบ	88
ภาคผนวก ข วงจรและแผ่นวงจรพิมพ์	92
ภาคผนวก ค รายการอุปกรณ์	94
ภาคผนวก ง แผนผังการทำงานและรหัสต้นฉบับของโปรแกรม	97
ภาคผนวก จ คู่มือการใช้งาน	132
ภาคผนวก ฉ รายละเอียดและคุณสมบัติของอุปกรณ์	139
ประวัติผู้แต่ง	182

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

เรื่อง	หน้า
2.1 การเลือกใช้การควบคุมที่เหมาะสมกับโปรเซส	23
2.2 การปรับอัตราของ PID	26
2.2 (ต่อ)การปรับอัตราของ PID	27
2.3 Reaction corve method	28
2.4 Ultimate Sensitivity method	29
2.5 เปรียบเทียบรูปแบบ RS232 RS422 RS485	32
2.6 เปรียบเทียบจุดอ้างอิง (Reference) ของมาตราต่างๆ	44
2.7 จุดปรับเทียบกับสเกลของอุณหภูมิ	45
2.8 บอกเส้นผ่านศูนย์กลาง ความยาว แรงดัน กำลังวัตต์	56
ค.1 รายการอุปกรณ์ของวงจรควบคุมการตรวจวัดอุณหภูมิและวงจรไดร์เวอร์	95
ค.2 รายการอุปกรณ์ของวงจรส่งผ่านข้อมูลแบบอนุกรม	96
ค.3 รายการอุปกรณ์ของวงจรแหล่งจ่ายแรงดัน	96
จ.1 ข้อมูลจำเพาะของชุดสาธิตการควบคุมอุณหภูมิภายในห้องจำลองแบบวงปิด	138

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป

รูปที่	หน้า
2.1 ระบบควบคุมอัตโนมัติชนิดที่มีการป้อนกลับ	3
2.2 การควบคุมเบื้องต้นหรือตัวควบคุมของระบบควบคุมอัตโนมัติ	4
2.3 แผนผังการทำงานตัวควบคุมแบบพี	4
2.4 ความสัมพันธ์ระหว่างเอาต์พุตกับอินพุตของตัวควบคุมแบบพี	5
2.5 แผนผังการทำงานของตัวควบคุมนิวเมติกส์แบบพี	6
2.6 แผนผังการทำงานที่จัดให้กระชับขึ้นจากรูปที่ 2.5	7
2.7 แผนผังการทำงานตัวควบคุมแบบ	8
2.8 ความสัมพันธ์ระหว่างเอาต์พุตกับอินพุตของตัวควบคุมแบบไอ	8
2.9 การเปลี่ยนแปลงค่าของอินพุตและเอาต์พุตของตัวควบคุมแบบไอ	9
2.10 แผนผังการทำงานของไฮดรอลิกแบบรวม	11
2.11 แผนผังการทำงานตัวควบคุมแบบดี	11
2.12 แผนผังการทำงานของตัวควบคุมแบบพีและไอ	14
2.13 การตอบสนองของตัวควบคุมพีและไอต่อค่าความผิดพลาดยูนิตสเต็ป	14
2.14 การตอบสนองของตัวควบคุมพีและไอต่อค่าความผิดพลาดยูนิตแรมป์	15
2.15 แผนผังการทำงานตัวควบคุมพีและดี	16
2.16 การตอบสนองของตัวควบคุมพีและดีที่มีต่อค่าความผิดพลาดยูนิตสเต็ป	16
2.17 การตอบสนองของตัวควบคุมพีและดีที่มีต่อค่าความผิดพลาดยูนิตแรมป์	17
2.18 แผนผังการทำงานของตัวควบคุมแบบพีไอดี	18
2.19 การตอบสนองของตัวควบคุมพีไอดี	18
2.20 ตัวควบคุมนิวเมติกส์แบบพีและไอ	19
2.21 ตัวควบคุมนิวเมติกส์แบบพีและดี	20
2.22 ตัวควบคุมนิวเมติกส์แบบพี (ย่านแคบ)ตัวควบคุมป้อนกลับ W/O	20
2.23 ตัวควบคุมนิวเมติกส์แบบพีป้อนกลับด้วยเบลโล	20
2.24 ตัวควบคุมนิวเมติกส์แบบพีไอดี	21

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป (ต่อ)

รูปที่	หน้า
2.25 ระบบควบคุมแบบป้อนกลับ	21
2.26 ผลตอบสนองของระบบควบคุมต่างๆ	22
(ก) อินพุตที่ต้องการ	22
(ข) โปรเซสไม่มีการรบกวน	22
(ค)โปรเซสมีการรบกวน	22
2.27 25% Damping ratio	25
2.28 พื้นที่การควบคุมน้อยสุด	25
2.29 การหา CS Reaction corve method	28
2.30 การหา KP Reaction corve method	28
2.31 การวัด Pu	29
2.32 ตัวควบคุมแบบดีที่สุด	30
2.33 ตัวควบคุมทั่วๆไป	31
2.34 การต่อ RS485 แบบ 2-WIRE Half Duplex	31
2.35 ตำแหน่งขาสัญญาณต่างๆของ PIC16F877	33
2.36 การจัดสรรพื้นที่หน่วยความจำของ PIC16F877	35
2.37 การจัดวางพื้นที่หน่วยความจำข้อมูล	36
2.38 การจัดขาของไอซี DS1820	37
2.39 โครงสร้างการทำงาน	37
2.40 การจัดสรรหน่วยความจำ	38
2.41 หลักการสร้างสัญญาณ PWM	49
2.42 หลักการของ PWM ที่มีการสุ่มตัวอย่างแบบเอกรูป	50
2.43 สัญญาณ PWM จากโครงการ EE99-61	51
2.44 การเปรียบเทียบสัญญาณ Sample-hold modulating signal กับสัญญาณ Triangular	51
2.45 Tpw (positive) และ Tpw (Negative)	53

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป (ต่อ)

รูปที่	หน้า
2.46 ฮีสเตอร์ครีบ	55
3.1 แผนผังการทำงานของชุดสาธิตการควบคุมอุณหภูมิภายในห้องจำลองแบบวงปิด	57
3.2 โครงสร้างของห้องจำลอง	58
3.3 วงจรตรวจจับอุณหภูมิ	59
3.4 วงจรไดร์เวอร์	60
3.5 วงจรส่งผ่านข้อมูลแบบอนุกรม	61
3.6 วงจรแหล่งจ่ายแรงดัน	61
3.7 วงจรควบคุมการทำงานของชุดสาธิตการควบคุมอุณหภูมิภายในห้องจำลองแบบวงปิด	62
3.8 ผังงานโปรแกรมตรวจจับอุณหภูมิ	63
3.9 ผังงานโปรแกรมควบคุมมอเตอร์และฮีตเตอร์ด้วยสัญญาณ PWM	65
3.10 ผังงานโปรแกรมหน้าต่างแสดงผลหลัก	66
3.11 ผังงานโปรแกรมย่อยกิริยาการควบคุม	67
3.12 ผังงานโปรแกรมย่อยปุ่มกดควบคุม	68
3.13 หน้าต่างควบคุมและแสดงผล	69
4.1 การทดลองห้องจำลอง	71
4.2 ส่ง-รับ อุณหภูมิไปยังไฮเปอร์เทอร์มินอล	72
4.3 การทดลองวงจรตรวจจับอุณหภูมิ	72
4.4 การทดลองวงจรไดร์เวอร์หลอดไฟหลอดที่ 1 ทำงาน	73
4.5 ผลการป้อนข้อมูล 100f1	73
4.6 การทดลองวงจรไดร์เวอร์หลอดไฟหลอดที่ 1 และ 2 ทำงาน	74
4.7 ผลการป้อนข้อมูล 100f1 และ 100f2	74
4.8 การทดลองวงจรไดร์เวอร์หลอดไฟหลอดที่ 1, 2 และ 3 ทำงาน	75
4.9 ผลการป้อนข้อมูล 100f1 , 100f2 และ 100f3	75
4.10 การทดลองวงจรไดร์เวอร์หลอดไฟหลอดที่ 1, 2, 3 และฮีตเตอร์ทำงาน	76

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป (ต่อ)

รูปที่	หน้า
4.11 ผลการป้อนข้อมูล 100f1 , 100f2, 100f3 และ 100h1	76
4.12 วงจรส่งผ่านข้อมูลแบบอนุกรม	77
4.13 ผลการทดลองวงจรวงจรส่งผ่านข้อมูลแบบอนุกรม	77
4.14 การทดลองจากไดเป่าผม	78
4.15 การควบคุมแบบพี ให้เข้าสู่ค่าเป้าหมาย	79
4.16 การควบคุมแบบพี ขณะกำลังรักษาระดับค่าเป้าหมายให้คงที่	80
4.17 การควบคุมแบบพีไอ ให้เข้าสู่ค่าเป้าหมาย	81
4.18 การควบคุมแบบพีไอ ขณะกำลังรักษาระดับค่าเป้าหมายให้คงที่	81
4.19 การควบคุมแบบพีดี ให้เข้าสู่ค่าเป้าหมาย	82
4.20 การควบคุมแบบพีดี ขณะกำลังรักษาระดับค่าเป้าหมายให้คงที่	83
4.21 การควบคุมแบบพีไอดี ให้เข้าสู่ค่าเป้าหมาย	84
4.22 การควบคุมแบบพีไอดี ขณะกำลังรักษาระดับค่าเป้าหมายให้คงที่	84
ก.1 ภาพห้องจำลองที่หนึ่งของชุดสาธิตการควบคุมอุณหภูมิ	89
ก.2 ภาพห้องจำลองที่สองตัวสร้างอุณหภูมิของชุดสาธิตการควบคุมอุณหภูมิ	89
ก.3 ภาพห้องจำลองที่สามที่จะต้องควบคุมอุณหภูมิ	90
ก.4 ภาพห้องจำลองชุดสาธิตการควบคุมอุณหภูมิในวงปิด	90
ก.5 ชุดแหล่งจ่ายไฟ 24 โวลต์	91
ก.6 ชุดควบคุมอุณหภูมิ	91
ข.1 วงจรควบคุมการทำงานของชุดสาธิตการควบคุมอุณหภูมิ	93
ข.2 แผ่นวงจรพิมพ์ควบคุมการทำงานของชุดสาธิตการควบคุมอุณหภูมิ	93
ง.1 ผังงานการควบคุมของชุดสาธิตการควบคุมอุณหภูมิ	98
ง.2 ผังงานโปรแกรม DS1820	99
จ.1 ส่วนควบคุมและแสดงผล	134
จ.2 หน้าต่าง PID Themperature Control	135

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป (ต่อ)

รูปที่	หน้า
จ.3 การเลือก Com Port	136
จ.4 การตั้งค่า Baud Rate	136
จ.5 การตั้งค่าเป้าหมายสูงหรือต่ำกว่าย่านที่กำหนด	137

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

1.1 ความเป็นมาและความสำคัญ

ปัจจุบันโรงงานอุตสาหกรรมส่วนใหญ่ได้ใช้ระบบการควบคุมแบบอัตโนมัติ เพื่อการเพิ่มผลผลิตและทำให้การผลิตมีประสิทธิภาพมากยิ่งขึ้น ทั้งในด้านระยะเวลาการผลิต คุณภาพมาตรฐาน ความน่าเชื่อถือ ความสะอาด และความปลอดภัย ทั้งยังเป็นการลดกำลังคนงาน ค่าใช้จ่ายซึ่งเกี่ยวกับต้นทุนการผลิต ทำให้ผู้ผลิตมีกำไรมากขึ้น ทางด้านเทคนิคแล้วเป้าหมายดังกล่าวจะสามารถบรรลุได้ ต้องอาศัยการตรวจวัดและควบคุมอย่างต่อเนื่องสม่ำเสมอ

การวัดปริมาณต่างๆ ในโรงงานอุตสาหกรรมจะเป็นการวัดปริมาณทางฟิสิกส์ อันได้แก่ การวัดอุณหภูมิ ความดัน ระดับ และอัตราการไหล โดยทั่วไปแล้วเครื่องควบคุมที่ดีจำเป็นจะต้องควบคุมปริมาณทางฟิสิกส์เหล่านี้ให้อยู่คงที่มีค่าใกล้เคียงกับค่าที่เหมาะสมมากที่สุดแม้ว่าสภาวะการทำงานและสภาพแวดล้อมจะมีการเปลี่ยนแปลงอยู่ตลอดเวลาก็ตาม

โรงงานสมัยใหม่ได้นำเอาเครื่องควบคุมอัตโนมัติมาใช้งาน เพื่อทำหน้าที่หลักในการคำนวณหาสัญญาณควบคุมที่เหมาะสม ซึ่งตัวควบคุมอัตโนมัติเหล่านี้ไม่ว่าจะถูกพัฒนาจนสามารถมีประสิทธิภาพสูงและมีการควบคุมที่ซับซ้อนขนาดไหนย่อมมีกิริยาการควบคุมมาจากกิริยาควบคุมพื้นฐานเหล่านี้คือ การควบคุมแบบสัดส่วนหรือแบบพี การควบคุมแบบรวมหรือแบบไอ และการควบคุมแบบอัตราหรือแบบดี จึงสร้างชุดสาธิตการควบคุมอุณหภูมิภายในห้องจำลองแบบวงปิดขึ้นมา เพื่อจะจำลองคุณสมบัติและพฤติกรรมของตัวควบคุม ซึ่งจำเป็นต้องปรับค่าตัวแปรพีไอดีเพื่อให้รักษาเสถียรภาพการควบคุมให้คงที่ ซึ่งย่อมเป็นพื้นฐานสำหรับการทำความเข้าใจในตัวควบคุมที่ซับซ้อนมากยิ่งขึ้น เพื่อใช้ประกอบการเรียนการสอนของนักศึกษาให้มีความรู้ความเข้าใจในเรื่องของการควบคุมมากยิ่งขึ้น

1.2 ขีดความสามารถของโครงงาน

1. สามารถปรับแต่งค่าพี พีไอ พีดี และพีไอดี ได้ด้วยคอมพิวเตอร์
2. สามารถรักษาระดับอุณหภูมิในย่าน 40 – 70 องศาเซลเซียสให้คงที่ได้ โดยมีความผิดพลาด ± 5%
3. สามารถแสดงผลการทำงานด้วยคอมพิวเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. สามารถควบคุมอุณหภูมิภายในห้องจำลองได้

5. ห้องจำลองมีขนาดกว้าง

1.3 เนื้อหาโดยสังเขป

เนื้อหาภายในปริญญานิพนธ์ฉบับนี้ได้แบ่งออกเป็นบทต่างๆ เพื่อสะดวกต่อการศึกษาและทำความเข้าใจ ในแต่ละบทจะประกอบไปด้วยเนื้อหาดังต่อไปนี้

บทที่ 1 กล่าวถึงความเป็นมาและความสำคัญของปริญญานิพนธ์ ขีดความสามารถของโครงงาน และเนื้อหาในบทต่างๆ โดยสังเขป

บทที่ 2 ประกอบด้วยทฤษฎีและหลักการ กล่าวถึงชนิดของระบบควบคุม การปรับแต่งการควบคุมแบบพีไอดี โครงสร้างของไอซี RS 485, RS 232, PIC 16F877 คำสั่งที่ใช้เขียนโปรแกรมสำหรับควบคุม และหลักการทำงานของพีไอดี

บทที่ 3 การออกแบบ การสร้าง และการทำงาน กล่าวถึงการออกแบบการสร้างส่วนประกอบต่างๆ การเขียนโปรแกรมควบคุม วงจรการควบคุมแบบพีไอดี โครงสร้างของชิ้นงาน

บทที่ 4 การทดลองและผลการทดลอง กล่าวถึงการทดลองและการตรวจสอบการทำงานของชุดควบคุมอุณหภูมิ ทดลองส่วนของห้องจำลอง ทดลองวงจรควบคุม ทดลองโปรแกรมการควบคุม ทดลองส่วนของการตรวจจับอุณหภูมิ ทดลองการควบคุมอุณหภูมิให้ได้ค่าตามเป้าหมายและแสดงผลทางจอคอมพิวเตอร์

บทที่ 5 กล่าวถึงสรุปผลของโครงงาน ข้อบกพร่องต่างๆ ปัญหาที่เกิดขึ้น และแนวทางการแก้ปัญหารวมทั้งแนวทางการพัฒนา

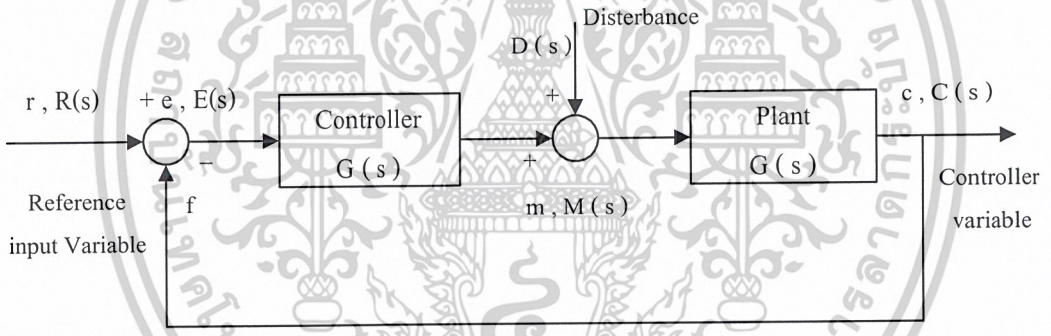
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2
ทฤษฎีและหลักการ

2.1 กล่าวนำ

ข้อมูลที่สำคัญและเป็นประโยชน์มากแก่การวิเคราะห์ วิจัยระบบควบคุมอัตโนมัติ ได้แก่ ชนิดของการควบคุมและแบบของระบบควบคุมเพราะเป็นข้อมูลที่ทำให้ทราบถึงลักษณะการทำงานของระบบโดยทั่วไปทำให้สามารถทำนายคุณสมบัติและการตอบสนองของระบบอย่างประมาณได้

ในระบบควบคุมอัตโนมัติชนิดที่มีการป้อนกลับดังแสดงในรูปที่ 2.1 ค่าความผิดพลาดหรือตัวแปร e



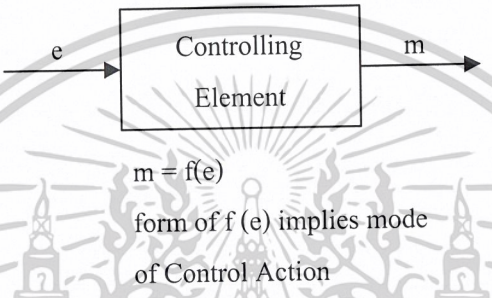
รูปที่ 2.1 ระบบควบคุมอัตโนมัติชนิดที่มีการป้อนกลับ

ซึ่งเป็นสัญญาณที่ได้มาจากการนำเอาสัญญาณป้อนกลับ f ไปเปรียบเทียบกับอินพุตตัวแปร r เป็นสัญญาณที่ถูกป้อนเข้าสู่ส่วนที่เป็นการควบคุมเบื้องต้น $G_1(S)$ เพื่อให้การควบคุมเบื้องต้นตอบสนองให้โยกย้ายตัวแปร m แก่ส่วนที่เป็นต้นกำลังเพื่อแก้ไขให้เอาต์พุตหรือควบคุมตัวแปร c มีลักษณะเป็นไปในทางที่สอดคล้องกับการเปรียบเทียบอินพุตตัวแปร r มากยิ่งขึ้น จะเห็นว่าหัวใจของระบบควบคุม คือส่วนที่เป็นการควบคุมเบื้องต้น $G_1(S)$ เพราะเป็นส่วนที่ทำหน้าที่แก้ไขความคลาดเคลื่อนระหว่างเอาต์พุต c กับอินพุต r ของทั้งระบบกล่าวคือรับเอาความคลาดเคลื่อนคือค่าผิดพลาด e เข้ามาสู่ตัวมันแล้วสั่งให้ส่วนที่เป็นต้นกำลังทำงาน (โดยการโยกย้ายตัวแปร m)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2 ชนิดของการควบคุม

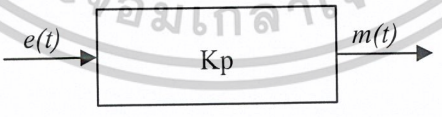
ชนิดของการควบคุมหรือกิริยาการควบคุม หมายถึงชนิดหรือวิธีการทำงานของส่วนที่เป็นตัวควบคุมโดยการควบคุมชนิดของการควบคุมหรือตัวควบคุมเพื่อแก้ไขความคลาดเคลื่อนระหว่างเอาต์พุตตัวแปร c กับอินพุตตัวแปร r ของทั้งระบบซึ่งได้แก่ความสัมพันธ์ระหว่างสัญญาณที่ออกจากการควบคุมเบื้องต้น คือ การโยกย้ายตัวแปร m กับสัญญาณที่เข้าสู่การควบคุมเบื้องต้น คือ ค่าความผิดพลาดตัวแปร e ดังแสดงในรูปที่ 2.2



รูปที่ 2.2 การควบคุมเบื้องต้นหรือตัวควบคุมของระบบควบคุมอัตโนมัติ

2.2.1 การควบคุมแบบสัดส่วน

การควบคุมแบบสัดส่วน (Proportional Control : P - Control)ในกรณีที่ชนิดของการควบคุมเป็นแบบสัดส่วนการทำงานของการควบคุมเบื้องต้นหรือตัวควบคุมจะเป็นไปในลักษณะของสัดส่วนคือการโยกย้ายตัวแปรจะมีค่ามากน้อยตามความมากน้อยของค่าความผิดพลาด e กล่าวคือ



รูปที่ 2.3 แผนผังการทำงานตัวควบคุมแบบพี

$$m(t) = K_P \ \ e(t) \tag{2.1}$$

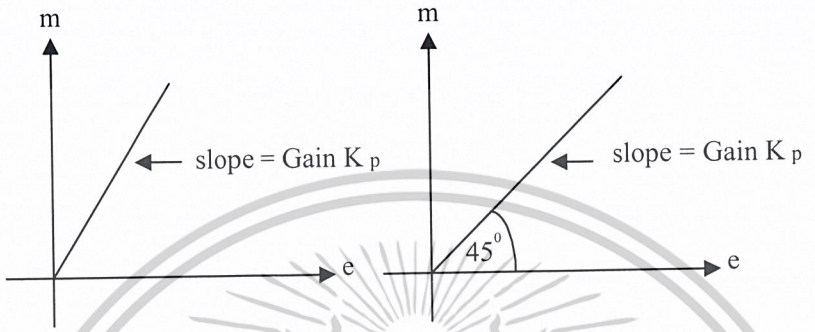
โดยที่ $m(t)$ = การเปลี่ยนแปลงของการโยกย้ายตัวแปร (จาก Ref.Op.Condition)

K_P = อัตราขยายหรือกำลังขยายของตัวควบคุมแบบพี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$e(t)$ = การเปลี่ยนแปลงของค่าความผิดพลาด (จาก Ref.Op.Condition)

อาจแสดงความสัมพันธ์นี้ได้ดังรูปที่ 2.4 ซึ่งจะเห็นได้ว่าเป็นเส้นตรงในบางครั้งจึงเรียกตัวควบคุมชนิดนี้ว่าการขยาย



ก) อัตราขยาย $K_p > 1$ ข) อัตราขยาย $K_p = 1$

รูปที่ 2.4 ความสัมพันธ์ระหว่างเอาต์พุตกับอินพุตของตัวควบคุมแบบพี

ตัวอัดอากาศก็จะโยกที่จุดหมุน O ทำให้ระยะ X มีค่าสูงขึ้นทำให้การอั้นของการไหลมีค่าลดลง อากาศก็ไหลผ่านปลายท่อออกสู่ภายนอกได้สะดวกขึ้นทำให้ความดัน P_b มีค่าลดลงและเบลโล B หดตัว (ระยะ z) การที่เบลโล B หดตัวก็จะมีผลคือทำให้วาล์วบอลเปิดมากขึ้นทำให้ความดัน P_C ซึ่งเป็นการโยกย้ายตัวแปรให้มีค่าสูงขึ้น

กระบวนการต่างๆ ที่เกิดขึ้นทั้งหมดนี้ อาจนำไปหาความสัมพันธ์ระหว่างตัวแปรต่างๆ ที่อุปกรณ์ต่างๆ เพื่อให้ได้มาซึ่งความสัมพันธ์ระหว่างเอาต์พุตกับอินพุตของทั้งตัวควบคุมซึ่งได้แก่ ความดัน P_C กับค่าผิดพลาด E และเมื่อใช้ค่าต่างๆ เป็นค่าที่วัดจากการเปรียบเทียบจะได้ความสัมพันธ์ของการเปลี่ยนแปลงของตัวแปรต่างๆ ดังนี้

ที่ตัวอัดอากาศ

อินพุต คือ e, เอาต์พุต คือ x

ความสัมพันธ์ $$x = \frac{b}{a+b}\ e \qquad (2.2)$$

ที่ด้านหลังของเบลโล B และปลายท่อ

อินพุต คือระยะ x, เอาต์พุต คือความดันหลังปลายท่อ (P_b)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ความสัมพันธ์ $P_b \alpha \pi \ d \ x$

หรือ $P_b = -K_1.x$ (2.3)

โดยที่ K_1 คือตัวคงค่าและเครื่องหมายลบเพราะ P_b จะเพิ่มเมื่อ x ลดค่าดังกล่าวคือการเปลี่ยนแปลงค่าของ P_b กับ x กลับทางกัน

ที่เบลโล B

อินพุต คือความดัน P_b, เอาต์พุต คือการยืดตัวหรือหดตัวของเบลโลความสัมพันธ์หาได้จากแรงสมดุล : (K_2 มีเครื่องหมายลบเพราะการเพิ่มค่าของ P_b ทำให้ z ลดลง)

$$P_b \ A_B = K_B \ z$$

หรือ $$P_b = \frac{K_B}{A_B} \ z \ -K_2 \ z \quad (2.4)$$

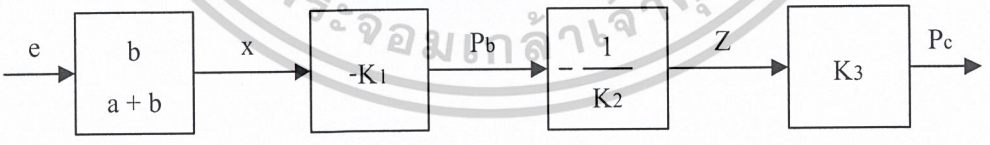
ที่วาล์วบอลหรือรีเลย์ตัวควบคุมนิวแมติกส์

อินพุต คือระยะ z , เอาต์พุต คือช่องการไหลซึ่งมีผลต่อความดัน P_C

ความสัมพันธ์ P_C a z

หรือ $P_C = K_3 \ z$ (2.5)

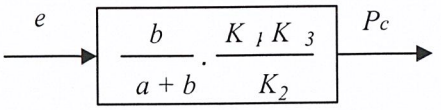
สมการ (2.2) ถึง (2.5) อาจแทนได้ด้วยแผนผังการทำงานดังรูปที่ 2.6 เป็น



รูปที่ 2.5 แผนผังการทำงานของตัวควบคุมนิวแมติกส์แบบพี

ซึ่งสามารถทำให้อยู่ในรูปที่กระชับขึ้นโดยใช้แผนผังการทำงานพีชคณิต ได้ดังรูปที่ 2.5 และได้ความสัมพันธ์ระหว่างการโยกย้ายตัวแปร (m) ซึ่งในระบบนี้ได้แก่ความดัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.6 แผนผังการทำงานที่จัดให้กระชับขึ้นจากรูปที่ 2.5

P_C กับอินพุตของระบบคือค่าความผิดพลาด e เป็น

$$P_C = \frac{b}{a+b} \cdot \frac{K_1 \ K_3}{K_2} \ .e \tag{2.6}$$

ซึ่งจะเห็นว่าการเปลี่ยนแปลงของ P_C เป็นสัดส่วนกับการเปลี่ยนแปลงของ e กล่าวคือเป็นตัวควบคุมแบบพีและเมื่อเทียบกับสมการ (2.1) จะเห็นว่าค่าในวงเล็บของสมการ (2.6) คือตัวควบคุมแบบพีอัตราขยาย K_P ซึ่งก็คือ G_1 (D) ในรูปที่ 2.1

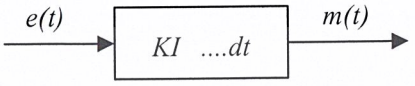
ถึงแม้ว่าตัวควบคุมแบบพีจะทำให้เกิดการสมดุลย์ระหว่างอินพุตกับเอาต์พุตในกระบวนการแต่ตัวควบคุมแบบพีก็มีขีดจำกัดที่สำคัญกล่าวคือ ถ้าสมมติว่าการควบคุมระดับนั้นต้องการที่จะควบคุมระดับให้อยู่ตรงจุดกึ่งกลาง ดังนั้นถ้าลูกลอยอยู่ตรงกึ่งกลางทางด้านไหลเข้าจะต้องเปิด 50% และปริมาณการไหลเข้าจะเท่ากับปริมาณการไหลออกซึ่งเป็นโหลดของระบบถ้าวาล์วทางด้านไหลออกเปิดมากขึ้นจะทำให้โหลดของระบบเพิ่มมากขึ้น นั่นคือลูกลอยจะลดต่ำลงและทำให้วาล์วทางด้านไหลเข้าเปิดมากขึ้น ในที่สุดตำแหน่งของวาล์วทางด้านเข้าก็จะไปอยู่ที่ตำแหน่งใหม่เมื่อระบบเข้าสู่สภาวะสมดุลย์อีกครั้งหนึ่ง แต่อย่างไรก็ตามลูกลอยหรือระดับอาจจะไปอยู่ที่ตำแหน่งใหม่ซึ่งต่างจากตำแหน่งเดิมที่ต้องการควบคุมในตอนแรกค่าความแตกต่างของระดับใหม่กับระดับเดิมนี้จะเรียกว่า “ออฟเซต”

การเกิดออฟเซตนี้เป็นคุณลักษณะของตัวควบคุมแบบพีทั้งหมด เนื่องจากว่าการทำงานของระบบตัวควบคุมแบบพีนั้นไม่สามารถที่จะควบคุมระบบที่มีโหลดที่มีการเปลี่ยนแปลงได้ดีเท่าที่ควรและในกรณีที่โหลดมีค่าคงที่แต่เปลี่ยนค่าของระดับของเหลวที่ถูกควบคุมไปก็เช่นเดียวกัน คือจะเกิดมีออฟเซตขึ้นและเมื่อระบบควบคุมที่มีการควบคุมตัวควบคุมแบบพีเริ่มเข้าสู่สภาวะคงที่นั้นออฟเซตจะต้องมีขนาดไม่เกินครึ่งหนึ่งของย่านตัวควบคุมแบบพี แต่ถ้าออฟเซตมีค่าเกินครึ่งหนึ่งของย่านตัวควบคุมแบบพีแล้ว วาล์วควบคุมก็จะเปิดเต็มที่หรือปิดสนิทและตัวควบคุมก็จะไม่สามารถควบคุมกระบวนการได้อีก ดังนั้นกิริยาควบคุมของตัวควบคุมแบบพีโดยทั่วไปจึงเหมาะสมกับกระบวนการที่ต้องการตัวควบคุมแบบพีในย่านแคบๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.2 การควบคุมแบบรวม

การควบคุมแบบรวม (Integral Control : I - Control) ตัวควบคุมมีลักษณะการทำงานกิริยาการควบคุมเป็นแบบรวมหรือตัวควบคุมแบบไอ คือตัวควบคุมที่ให้การแก้ไขความคลาดเคลื่อน m ในลักษณะที่เป็นสัดส่วนกับช่วงเวลาไต่ขึ้นของค่าความผิดพลาด e



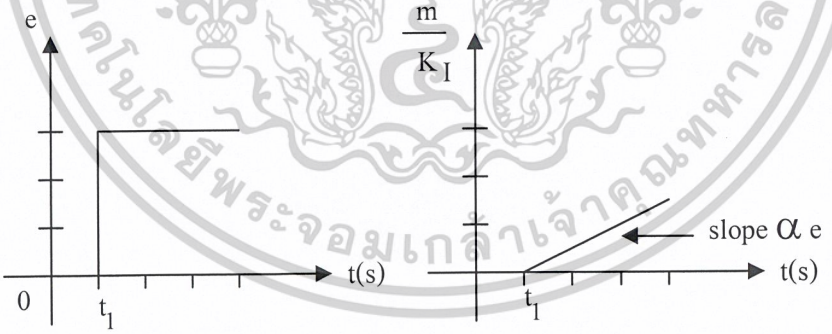
รูปที่ 2.7 แผนผังการทำงานตัวควบคุมแบบไอ

$$m\alpha \int edt$$

หรือ

$$m = K_I \int edt \tag{2.7}$$

โดยที่ K_I คือตัวคงค่าเรียกว่า อัตราขยายหรือกำลังขยายของตัวควบคุมแบบไอและอาจแสดงค่าสัมพันธ์นี้ได้ดังแสดงในรูปที่ 2.8



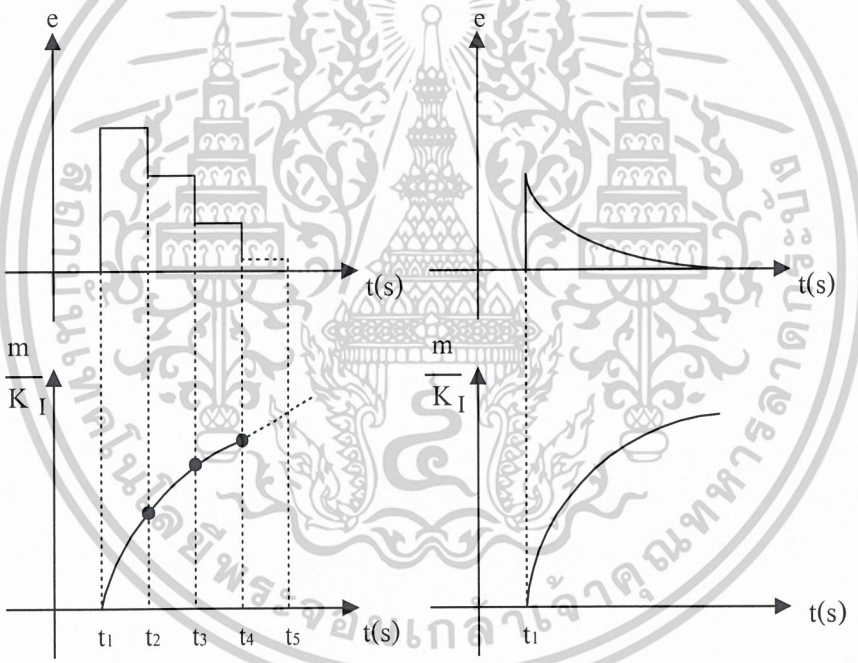
รูปที่ 2.8 ความสัมพันธ์ระหว่างเอาต์พุตกับอินพุตของตัวควบคุมแบบไอ

กล่าวคือหากค่าความผิดพลาดเป็นการเปลี่ยนลำดับขั้นการโยกย้ายอัตราส่วนของตัวแปรจะเป็นฟังก์ชั่นแรมป์ ซึ่งมีความลาดชันขึ้นอยู่กับขนาดของค่าความผิดพลาด e ดังสมการ (2.8) ซึ่งได้จากการดิฟเฟอร์เรนเชียลสมการ (2.7) เทียบกับเวลา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$\frac{1}{K_I}\ \dot{m} = e \tag{2.8}$$

จากความสัมพันธ์ในสมการ (2.8) จะเห็นว่าเมื่อค่าความผิดพลาดมีค่าน้อยลงค่าของ $\frac{\dot{m}}{K_I}$ หรือความชันของ $\frac{\dot{m}}{K_I}$ จะใกล้แนวระดับขึ้นทุกทีและกลับมาอยู่ในระดับดังในรูป 2.9 เมื่อ $e = 0$ ซึ่งในทางคณิตศาสตร์กล่าวได้ว่า e จะเท่ากับศูนย์หรือความชันของ $\frac{1}{K_I}\ m$ จะอยู่ในแนวระดับเมื่อเวลาผ่านไปนานมาก คือ t



(ก) เวลาระหว่างดีฟินิตในรูปแบบการพินิจพิเคราะห์

(ข) เวลาระหว่างอินฟินิตซิมอลในรูปแบบต่อเนื่อง

รูปที่ 2.9 การเปลี่ยนแปลงค่าของอินพุตและเอาต์พุตของตัวควบคุมแบบไอ

ข้อสังเกตุที่เห็นได้ชัดอีกประการก็คือเมื่อ e กลับมา = 0 (เมื่อ $\rightarrow \infty$) แล้วค่าของ $\frac{m}{K_I}$ ไม่จำเป็นว่าจะกลับมา = 0 เลข (เฉพาะความชันของ $\frac{m}{K_I}$ เท่านั้นที่กลับมา = 0) จึงอาจสรุปได้ว่า การโยกย้ายตัว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แปร (m) ของตัวควบคุมแบบไอจะมีค่าที่ลอยตัวผิดกับในตัวควบคุมแบบพี ดังแสดงในสมการ (2.1)

กล่าวคือ m จะ = 0 เสมอทุกครั้งที่ $e = 0$

กระบวนการต่างๆ ที่เกิดขึ้น อาจแทนได้ด้วยความสัมพันธ์ทางคณิตศาสตร์ ได้ดังนี้ที่ช่องหลอด

อินพุต คือ e , เอาต์พุต คือ $\dot{q}_H$

ความสัมพันธ์ $$\dot{Q}_H = C_d \quad A_F \sqrt{\frac{P_S - P_b}{e}}$$

ซึ่งอาจจะเห็นว่าอยู่ในรูปของ $\dot{G}_H = f(A\ ,\ P_b)$ เนื่องจาก C_d , P_S และ e มีค่าคงที่ และโดยที่ A_F ซึ่งได้แก่พื้นที่การไหลมีค่าขึ้นอยู่กับความมากน้อยของ e ซึ่งอาจกล่าวได้ว่า

$$\dot{G}_H = f(e\ \ P_b) \tag{2.9}$$

โดยสมการเชิงเส้นใกล้การเปรียบเทียบสภาวะการทำงาน :

$$\begin{aligned} \dot{q}_H &= K_1.e - K_2.P_b \\ &\cong K_1.e \end{aligned} \tag{2.10}$$

เนื่องจากการเปลี่ยนแปลงของ $\dot{q}_H$ จะขึ้นอยู่กับการเปลี่ยนแปลงของ e มากกว่า P_b มาก

ที่ลูกสูบกระบอกสูบ

อินพุต คือ การไหลของน้ำมันที่เข้าและออกทางส่วนล่างและบนของลูกสูบ

เอาต์พุต คือ การเคลื่อนที่ของลูกสูบ

ความสัมพันธ์ (จากสมการต่อเนื่อง)

$$\dot{q}_H = A.\dot{y} = AD(y) \tag{2.11}$$

ที่วาล์วควบคุมการไหลของเชื้อเพลิง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

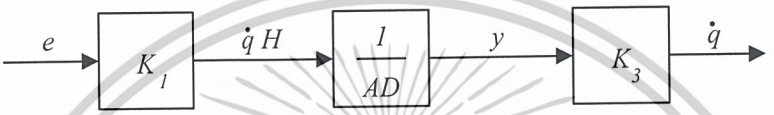
ที่วาล์วควบคุมการไหลของเชื้อเพลิง

อินพุต คือ y , เอาต์พุต คือ q

ความสัมพันธ์ $\dot{q} \; a \; y$

หรือ $\dot{q} = K_3 . y$ (2.12)

สมการ (2.9) ถึง (2.12) อาจแทนได้ด้วยแผนผังการทำงาน ดังในรูปที่ 2.10



รูปที่ 2.10 แผนผังการทำงานของไฮดรอลิกแบบรวม

นั่นคือ

$$q = \frac{K_1 K_3}{AD} \; e$$

$$\cong \frac{K}{D} \; e \qquad (2.13)$$

หรือ

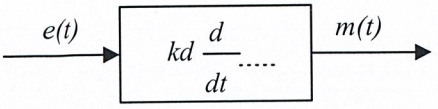
$$q = \frac{K}{D} \; e$$

$$= K \int e dt$$

ซึ่งจะเห็นว่าเอาต์พุตคือการโยกย้ายตัวแปรมีค่าเป็นสัดส่วนกับช่วงเวลาไต่ขึ้นของอินพุตคือค่าความผิดพลาด e

2.2.3 การควบคุมแบบอัตรา

การควบคุมแบบอัตรา (Derivative Control : D - Control) หรือตัวควบคุมแบบดีเป็นการควบคุมซึ่งมีการแก้ไขความค่าคลาดเคลื่อนของระบบในลักษณะที่เป็นสัดส่วนกับอัตราของการเปลี่ยนแปลงของค่าผิดพลาด e กล่าวคือ



รูปที่ 2.11 แผนผังการทำงานตัวควบคุมแบบดี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$m \alpha \frac{de}{dt}$$

หรือ

$$m = K_d \frac{de}{dt} \tag{2.14}$$

โดยที่ K_d คือตัวคงค่าซึ่งเรียกว่าอัตราขยายตัวควบคุมแบบดี

ซึ่งจะเห็นว่าการควบคุมเบื้องต้นหรือตัวควบคุมชนิดนี้จะทำให้การโยกย้ายตัวแปร m ทันทีที่ค่าของความผิดพลาด e มีการเปลี่ยนแปลงผิดกับการควบคุมชนิดอื่นๆ ซึ่งจะให้การแก้ไขความคลาดเคลื่อนจากปริมาณความคลาดเคลื่อน จากปริมาณของความคลาดเคลื่อน e การควบคุมแบบอัตราหรือตัวควบคุมแบบดีนี้จึงมีประโยชน์มากเพราะทำหน้าที่เสมือนเป็นตัวเฝ้าความเคลื่อนไหวของค่าความผิดพลาดโดยจะให้การแก้ไขโดยทันทีที่สภาพของค่าผิดพลาดเปลี่ยนไป จึงเท่ากับเป็นการแก้ไขความคลาดเคลื่อนของระบบล่วงหน้าแทนที่จะให้การแก้ไขหลังจากมีความคลาดเคลื่อนเกิดขึ้นแล้วเหมือนการควบคุมแบบสัดส่วนและแบบรวม ดังที่ได้กล่าวมา

ข้อเสียของการควบคุมแบบอัตราหรือตัวควบคุมแบบดีนี้คือถ้าหากค่าความผิดพลาด e มีค่าคงที่ไม่ว่าจะมากหรือน้อยเท่าใด การควบคุมเบื้องต้นของตัวควบคุมชนิดนี้จะไม่ให้มีการโยกย้ายตัวแปร m กล่าวคือ

$$m = K_d \frac{de}{dt} \tag{2.15}$$

$= 0$ เมื่อ $e =$ ตัวคงค่า

นั่นคือ มันจะไม่ตอบสนองต่อค่าผิดพลาด e_{ss} ใดๆ ในทางปฏิบัติจึงไม่ใช้การควบคุมชนิดนี้เพียงชนิดเดียวในระบบควบคุมอุตโนมัติ แต่จะถูกนำไปใช้ร่วมกับการควบคุมชนิดอื่น ดังที่ได้กล่าวมาแล้วเป็นการควบคุมแบบผสมเสมอ

2.2.4 การควบคุมแบบผสม

การควบคุมแบบผสม (Mixed – Mode of Control Action) เป็นการควบคุมซึ่งมีการแก้ไขความคลาดเคลื่อนหรือของระบบเป็นแบบผสมคือให้การโยกย้ายตัวแปร m ในลักษณะที่เหมือนกับเอาลักษณะการทำงานของการควบคุมชนิดต่างๆ มาผสมกัน เช่น การควบคุมแบบสัดส่วนผสมแบบรวมหรือตัวควบคุมแบบพีและแบบไอ, การควบคุมแบบสัดส่วนผสมแบบอัตราหรือตัวควบคุมแบบพีและดี และการควบคุมแบบสัดส่วนผสมแบบรวมผสมแบบอัตราหรือตัวควบคุมแบบพีไอดี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จุดประสงค์ของการนำเอาการควบคุมแบบต่างๆ มาผสมกันเป็นการควบคุมแบบผสมก็เพื่อแก้ให้การตอบสนองของระบบดีขึ้น กล่าวคือเพื่อปรับปรุงความรวดเร็ว ความแม่นยำ และความเสถียรของระบบให้ดียิ่งขึ้น โดยนำเอาข้อดีของการควบคุมแต่ละแบบมารวมกัน เช่น ระบบที่มีการควบคุมแบบสัดส่วนอาจจะให้การตอบสนองได้รวดเร็วได้จริง แต่ไม่ว่าเวลาจะผ่านไปนานเท่าไร (ที่ค่าสูญเสีย) ย่อมมีความคลาดเคลื่อนเสมอ หากระบบถูกรบกวนจากภายนอกซึ่งปัญหานี้จะแก้ไขไปได้หากใช้การควบคุมแบบสัดส่วนผสมแบบรวมและในทำนองเดียวกันการควบคุมแบบสัดส่วนผสมแบบรวมสำหรับระบบควบคุมบางชนิด เช่น การควบคุมแบบเบาแรงย่อมให้การตอบสนองของระบบสู้การตอบสนองของระบบที่มีการควบคุมแบบสัดส่วนผสมแบบรวมผสมแบบอัตราไม่ได้ ในแง่ที่ว่าขาดระบบล่วงหน้าไป กล่าวคือ ในบางครั้งระบบจะให้การตอบสนองที่มีค่าความคลาดเคลื่อนหรือเอาต์พุตที่แกว่งไปจากอินพุตตัวแปร (r) มากเพราะในระบบที่มีการควบคุมแบบพีและไอนั้นจะไม่ทำงานทันทีที่มีการเปลี่ยนแปลงค่าของค่าความผิดพลาด e ผิดกับแบบหรือในกรณีของการควบคุมแบบเบาแรงระบบที่มีการควบคุมแบบพีและไอ อาจทำงานไม่ทันเพราะการเปรียบเทียบอินพุตตัวแปร (r) เปลี่ยนแปลงอยู่แทบตลอดเวลา

1) การควบคุมแบบสัดส่วนผสมแบบรวม

ความสัมพันธ์ของการโยกย้ายตัวแปร m กับค่าความผิดพลาด e ของการควบคุมแบบผสมชนิดนี้คือ

$$m = K_p \left[e + K_I \int_0^t e\,dt \right]$$

$$K_P \left[1 + \frac{1}{T_I D} \right] e \qquad (2.16)$$

โดยที่ K_P อัตราขยายของตัวควบคุมแบบพี

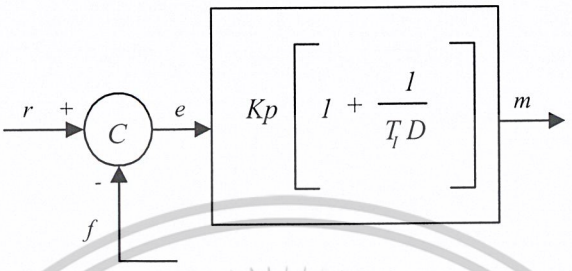
K_I อัตราขยายของตัวควบคุมแบบไอ

T_I ช่วงเวลากิริยาควบคุมแบบไอ $= \frac{K_P}{K_I}$

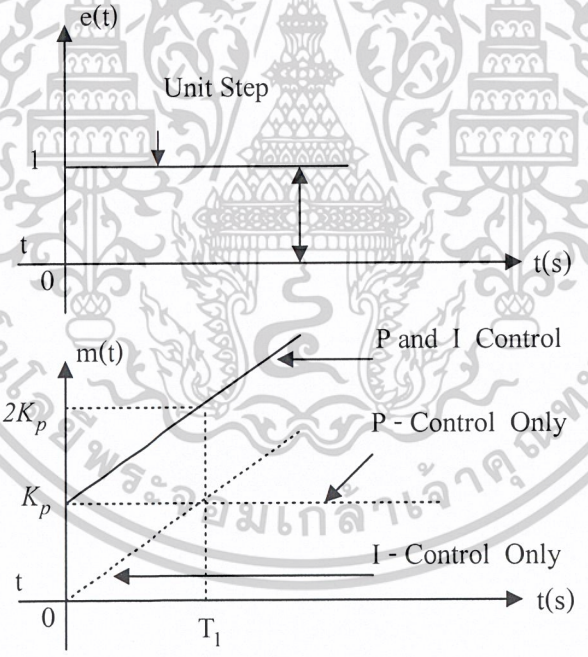
= ระยะเวลาซึ่งตัวควบคุมแบบไอให้การตอบสนองเท่ากับการตอบสนองของตัวควบคุมแบบพี (ส่วนกลับของ T_I คือ อัตราส่วนเริ่มต้น)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ซึ่งแทนได้ด้วยแผนผังการทำงานดังแสดงในรูปที่ 2.12 และการตอบสนองของการโยกย้ายตัวแปร m ที่มีค่าต่อความผิดพลาด e ในกรณีที่ค่าความผิดพลาดเป็นฟังก์ชั่นสเต็ปและฟังก์ชั่นแรมป์อาจแสดงได้ดังรูปที่ 2.13 และ 2.14 ตามลำดับ

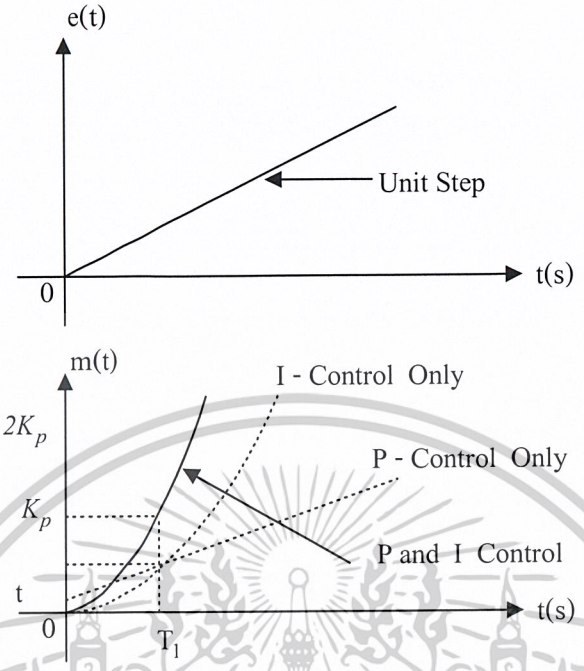


รูปที่ 2.12 แผนผังการทำงานของตัวควบคุมพีและไอ



รูปที่ 2.13 การตอบสนองของตัวควบคุมพีและไอต่อค่าความผิดพลาดยูนิตสเต็ป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.14 การตอบสนองของตัวควบคุมพีและไอต่อค่าผิดพลาดยูนิตแรมป์

2) การควบคุมแบบสัดส่วนผสมแบบอัตรา

ความสัมพันธ์ของการโยกย้ายตัวแปร m กับค่าความผิดพลาด e ของการควบคุมแบบผสมชนิดนี้คือ

$$m = K_P \ e + K_D \ \frac{de}{dt}$$

$$K_P \ e + K_D \ T_d \ \frac{de}{dt}$$

$$= K_P \left[1 + T_d \ D\right] e \qquad (2.17)$$

โดยที่ K_P อัตราขยายของตัวควบคุมแบบพี

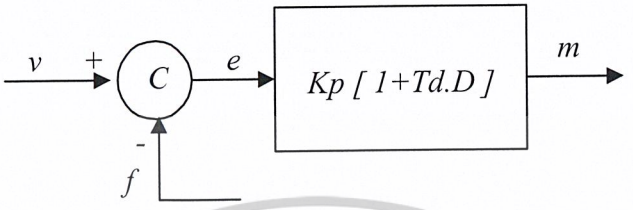
K_D อัตราขยายของตัวควบคุมแบบดี

T_d ช่วงเวลาของตัวควบคุมแบบดี

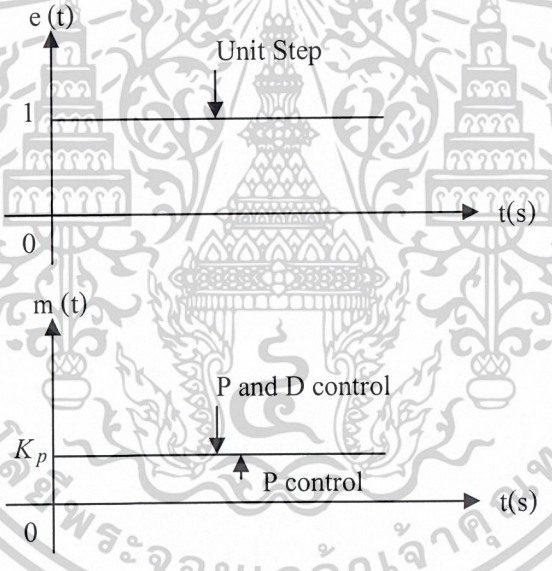
= ระยะเวลาที่ตัวควบคุมแบบดีให้การตอบสนองนำการตอบสนองของตัวควบคุมแบบพี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ซึ่งอาจแทนได้ด้วยแผนผังการทำงาน ดังแสดงในรูปที่ 2.15 และผลการตอบสนองของการโยกย้ายตัวแปร m ที่มีต่อค่าความผิดพลาด e ในกรณีที่ค่าความผิดพลาดเป็นฟังก์ชั่นสเต็ปและฟังก์ชั่นแรมป์อาจแสดงได้ดังรูปที่ 2.16 และ 2.17 ตามลำดับ



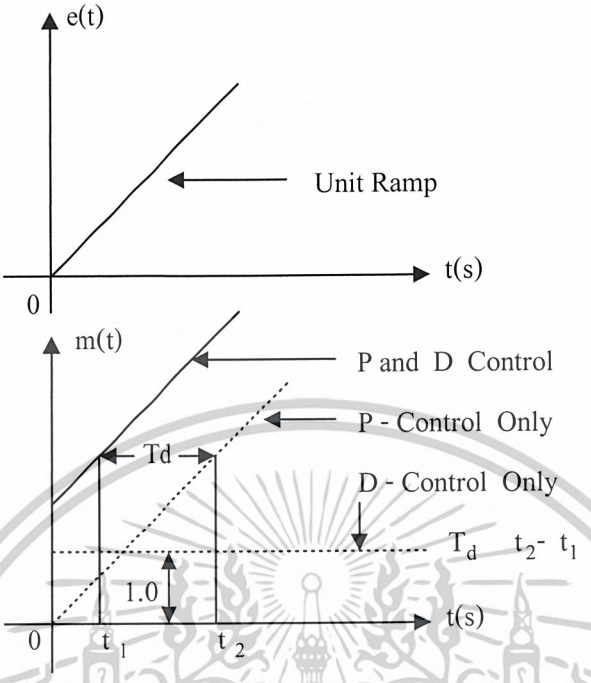
รูปที่ 2.15 แผนผังการทำงานตัวควบคุมพีและดี



รูปที่ 2.16 การตอบสนองของตัวควบคุมพีและดีที่มีต่อค่าความผิดพลาดยูนิตสเต็ป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง



รูปที่ 2.17 การตอบสนองของตัวควบคุมพีและดีที่มีต่อค่าความผิดพลาดยูนิตแรมป์

3) การควบคุมแบบสัดส่วนผสมแบบรวมผสมแบบอัตรา

ความสัมพันธ์ของการโยกย้ายตัวแปร m กับค่าความผิดพลาด e ของการควบคุมแบบผสมชนิดนี้คือ

$$m = K_P\ e + K_I \int_0^t e\,dt + K_D \frac{de}{dt}$$

$$K_P\ e + \frac{K_P}{T_I} \int_0^t e\,dt + K_P\ T_d \frac{de}{dt}$$

$$K_P\ 1 + \frac{1}{T_I D} + T_d\ D\ e \qquad (2.18)$$

โดยที่ K_P อัตราขยายของตัวควบคุมแบบพี

K_I อัตราขยายของตัวควบคุมแบบไอ

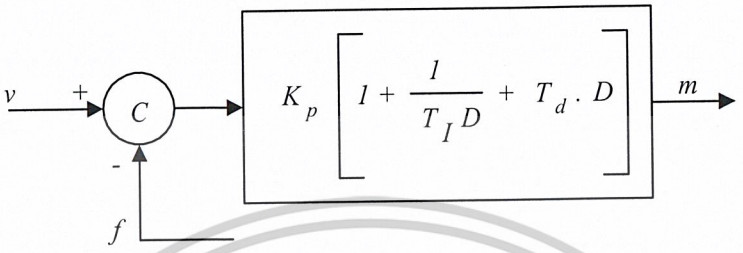
K_D อัตราขยายของตัวควบคุมแบบดี

T_I ช่วงเวลาของตัวควบคุมแบบไอ

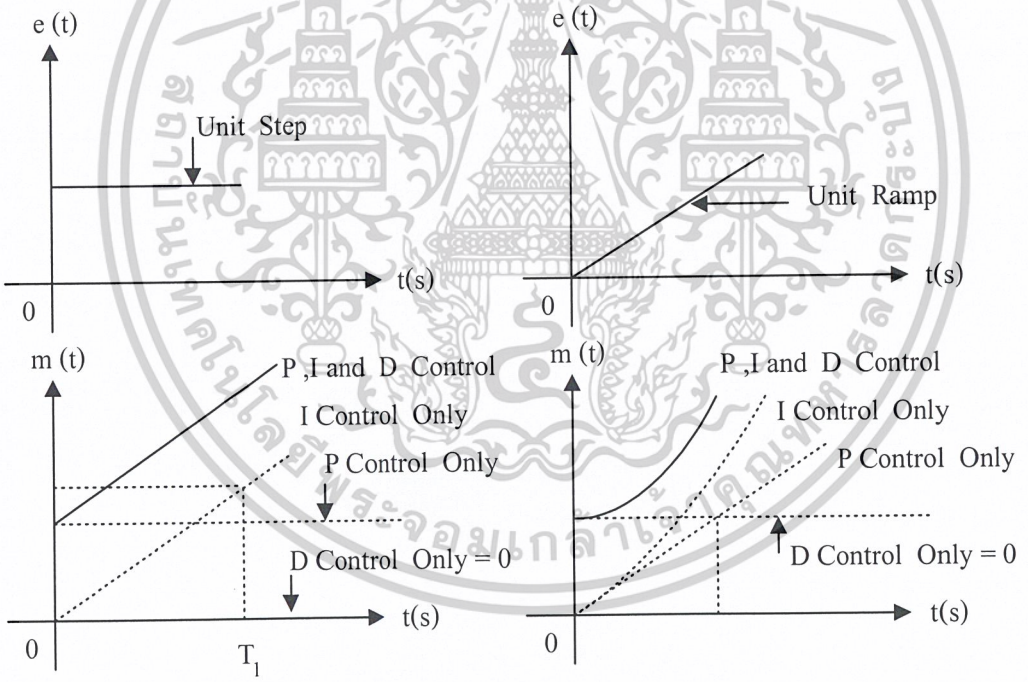
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
51071

T_d ช่วงเวลาของตัวควบคุมแบบดี

ซึ่งอาจแทนได้ด้วยแผนผังการทำงาน ดังแสดงในรูปที่ 2.18 และการตอบสนองของการเปรียบเทียบ



รูปที่ 2.18 แผนผังการทำงานของตัวควบคุมพีไอดี

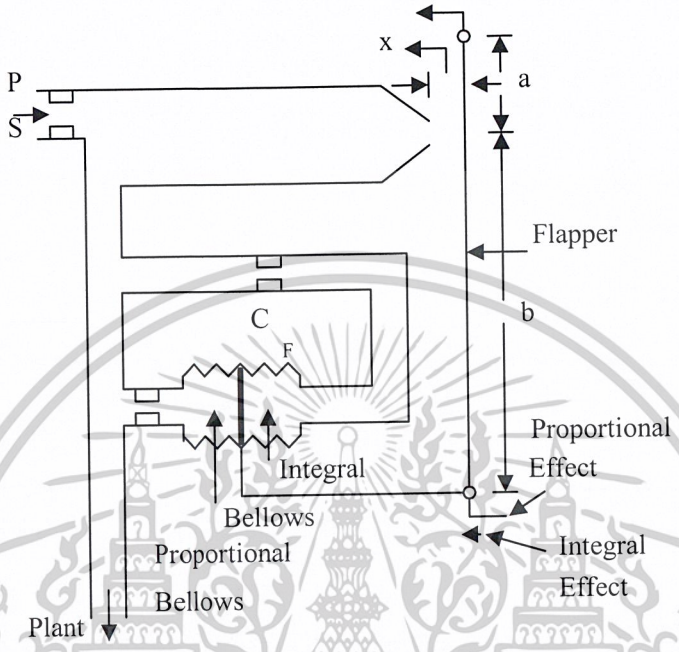


รูปที่ 2.19 การตอบสนองของตัวควบคุมพีไอดี

ตัวแปร m ที่มีต่อค่าความผิดพลาด e ในกรณีที่ค่าความผิดพลาดเป็นฟังก์ชั่นสเต็ปและฟังก์ชั่นแรมป์อาจแสดงได้ดังรูปที่ 2.19

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การควบคุมแบบสัดส่วนผสมแบบรวมหรือตัวควบคุมพีและไอได้แก่ ตัวควบคุมนิวแมติกส์แบบพีและไอ ดังรูปที่ 2.20

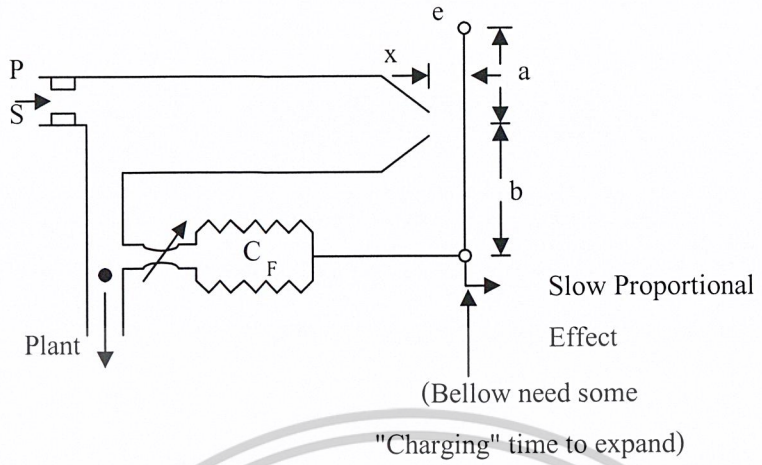


รูปที่ 2.20 ตัวควบคุมนิวแมติกส์แบบพีและไอ

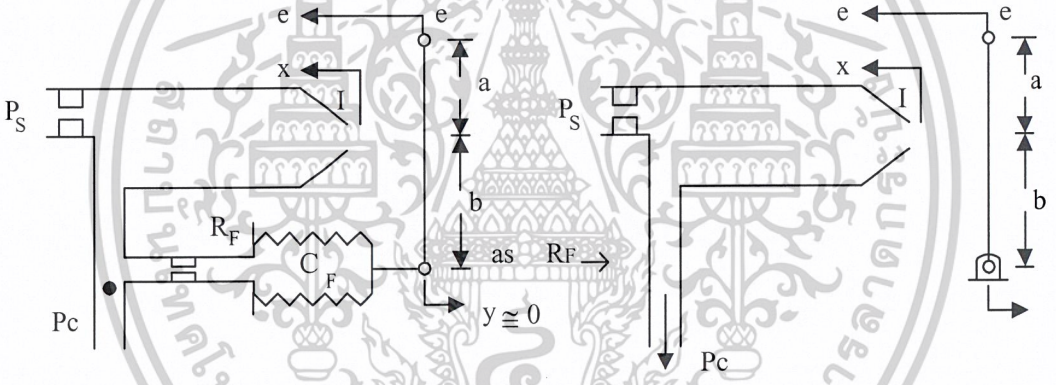
ซึ่งจะเห็นว่ากิริยาควบคุมแบบไอในตัวควบคุมจะเป็นไปในลักษณะที่แก้การป้อนกลับเนื่องมาจากตัวควบคุมแบบพีอย่างช้าๆ ฉะนั้นในช่วงแรกที่ e มีการเปลี่ยนแปลง ระบบก็จะให้ผลตอบสนองเหมือนกับเป็นตัวควบคุมแบบพี ซึ่งดีคือรวดเร็วและเมื่อเวลาผ่านไประบบก็จะให้ผลตอบสนองเหมือนกับเป็นตัวควบคุมแบบไอเมื่อเบลโลแบบรวมมีความดัน = P_c

รูปที่ 2.21 แสดงการควบคุมแบบสัดส่วนผสมแบบอัตราหรือตัวควบคุมแบบพีและดี ซึ่งจะได้เห็นว่าเมื่อ R_F มีค่าสูงมากตัวควบคุมก็จะกลายเป็นตัวควบคุมแบบพี ตัวควบคุมชนิดที่มีการป้อนกลับ ดังในรูปที่ 2.22 และรูปที่ 2.23 ตามลำดับและตัวควบคุมนิวแมติกส์ชนิดตัวควบคุมแบบพีไอดี ดังในรูปที่ 2.24

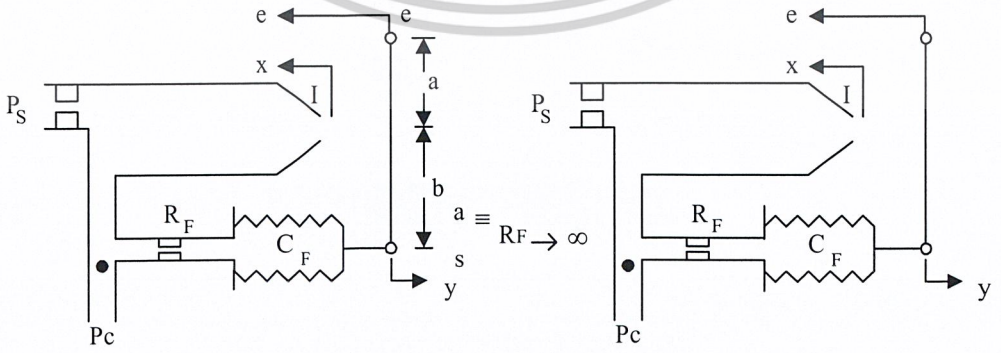
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.21 ตัวควบคุมนิวแมติกส์แบบพีและดี

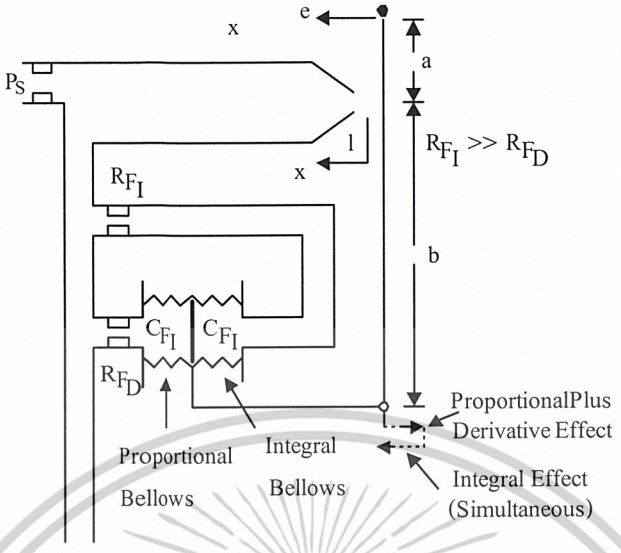


รูปที่ 2.22 ตัวควบคุมนิวแมติกส์แบบพี (ย่านแคบ) ตัวควบคุมป้อนกลับ W / O



รูปที่ 2.23 ตัวควบคุมนิวแมติกส์แบบพีป้อนกลับด้วยเบลโล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

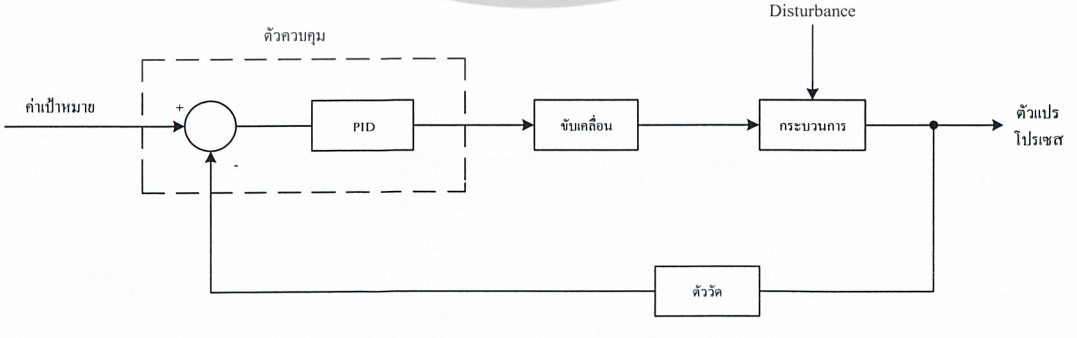


รูปที่ 2.24 ตัวควบคุมนิวแมติกส์แบบพีไอดี

2.3 การปรับค่า PID

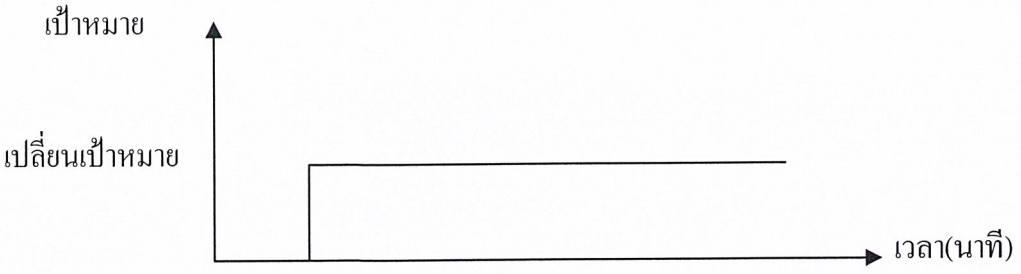
2.3.1 ผลการตอบสนองที่ดี

ในระบบการควบคุมแบบป้อนกลับ ตัวควบคุมจะพยายามรักษาให้ค่าตัวแปรโปรเซสมีค่าเท่ากับค่าเป้าหมายอยู่เสมอ ในกรณีที่เกิดมีการเปลี่ยนค่าเป้าหมายจะทำให้ตัวแปรโปรเซสมีค่าต่างไปจากค่าเป้าหมายไปในขณะหนึ่ง ตัวควบคุมจะพยายามควบคุมให้ตัวแปรโปรเซสมีค่าเท่ากับค่าเป้าหมายนี้ ในที่สุดลักษณะการนำค่าตัวแปรโปรเซสให้เข้าใกล้ค่าเป้าหมายนี้ จะแตกต่างกันตามคุณสมบัติของระบบควบคุม บางระบบควบคุมอาจควบคุมให้ตัวแปรโปรเซสส่งเข้าหาค่าเป้าหมายได้เร็ว แต่บางระบบอาจทำได้ดีกว่า สามารถทดสอบได้โดยดูที่ผลตอบของการควบคุม

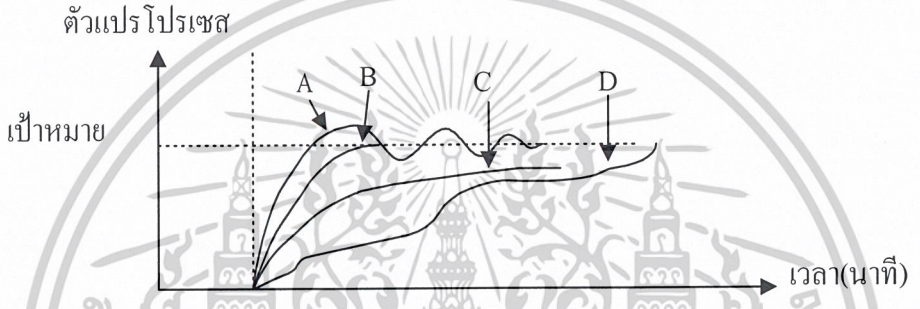


รูปที่ 2.25 ระบบควบคุมแบบป้อนกลับ

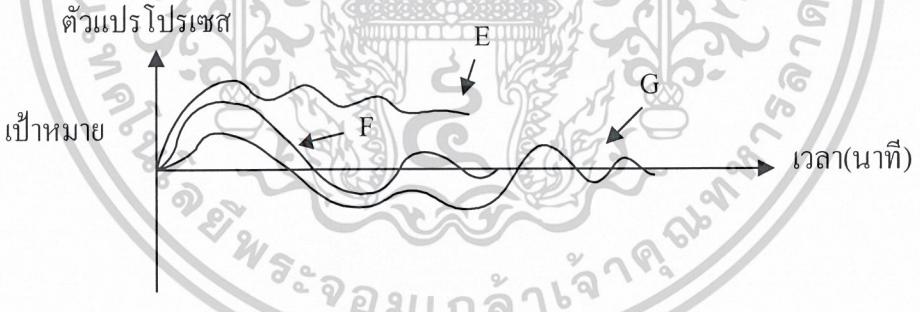
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



(ก) อินพุตที่ต้องการ



(ข) โปรเซสไม่มีการรบกวน



(ค) โปรเซสมีการรบกวน

รูปที่ 2.26 ผลตอบสนองของระบบควบคุมต่างๆ

โดยที่ A มีโอเวอร์ชูตและการแกว่ง

B ตอบรับเร็ว

C ตัวแปรโปรเซสมีค่าไม่เท่ากับค่าเป้าหมายแม้เวลาผ่านพ้นไปนานเรียกว่า ออฟเซต

D ตอบรับช้ามาก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อตัวแปรโปรเซสหยุดนิ่งมีค่าเป้าหมาย ในขณะนั้นเกิดมีการรบกวน (Disturbance) เข้ามารบกวนโปรเซส ได้แก่ การเปลี่ยนของโหลดหรือการเปลี่ยนแปลงของสภาพแวดล้อม เป็นต้น ผลตอบของโปรเซสจะมีหลายแบบดังแสดงในรูป

E เกิดออฟเซตได้ค่าผิดไปจากค่าเป้าหมายเดิม

F เกิดการแกว่งเล็กน้อยก่อนกลับเข้าสู่ค่าเป้าหมายเดิม

G ไม่เกิดออฟเซต

เมื่อพิจารณาผลตอบของการควบคุมชนิดต่างๆ เหล่านี้พอสรุปได้ว่า ระบบการควบคุมที่ดีจะมีคุณสมบัติดังนี้

- มีเสถียรภาพ ไม่เกิดการแกว่ง (Oscillation) เมื่อถูกกระตุ้น
- ตอบรับการเปลี่ยนค่าเป้าหมายได้รวดเร็ว
- ไม่เกิดออฟเซต

2.3.2 ความยากง่ายในการควบคุมโปรเซส

"เดดไทม์" (dead time) เป็นศัตรูตัวร้ายของการควบคุมลักษณะสมบัติของโปรเซสทั่วๆ ไปจะมี ไทม์แล็ก (Time lag) ดังที่ได้กล่าวมาแล้วจากรูปคลื่นผลตอบของโปรเซสต่อ สเต็ปอินพุต เราสามารถหาค่าเดดไทม์ (LE) และค่าคงตัวเวลา (TE) โดยประมาณได้จะเป็นค่าที่ใช้ประเมินความยากง่ายในการควบคุมและใช้เลือกแบบการควบคุม

ตารางที่ 2.1 การเลือกใช้การควบคุมที่เหมาะสมกับโปรเซส

$LE / _{TE}$	แบบการควบคุมที่เหมาะสมกับโปรเซสนั้น
$LE / _{TE} < 0.2$	ON – OFF, P, PI
$0.2 < LE / _{TE} < 0.2$	PI, PID
$0.1 < LE / _{TE}$	Feed forward, Computer Control

$LE / _{TE} < 1.0$ สามารถใช้การควบคุมแบบ PID ได้ตามปกติ $0.5 < LE / _{TE} < 1.0$ การควบคุมแบบ PID ก็ควบคุมได้ยากมาก มีข้อพิจารณาในการเลือกใช้การควบคุมที่เหมาะสมดังตารางที่ 2.1

1. สาเหตุของการขาดเสถียรภาพในระบบ

- ผลของระบบควบคุมข้างเคียวหรืออื่นๆ
- การรบกวนเป็น periodic

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- มี Nonlinear element ในระบบเอง เช่น Blacklash ความผิด Dead band เป็นต้น

- โปรเซสเกน หรือ ลูปเกน ค่าสูงเกินไป

- คุณสมบัติของตัวแปรในระบบไม่คงที่แปรเปลี่ยนตามการรบกวน, ค่าเป้าหมาย, เวลา เป็นต้น

2.3.3 ผลของ PID ต่อเสถียรภาพของระบบ

ในระบบควบคุมแบบป้อนกลับ ซึ่งใช้ตัวควบคุมแบบ PID นั้น ถ้าเราลองแปรค่า PB, T_I และ T_D จะมีผลต่อผลตอบสนองของระบบควบคุมดังนี้

1) ผลของ P action

เมื่อลดค่า PB ลงทำให้อัตราขยายสูงขึ้นจะมีผลทำให้

1.1) ออฟเซตลดลง

1.2) Period ของการแกว่งเล็กลง

1.3) อัตราส่วนของช่วงกว่างการแกว่งเพิ่มขึ้น ระบบขาดเสถียรภาพมากขึ้น

2) ผลของ I action

เมื่อให้ P และ D action คงที่แล้ว ลองลด T_I (Reset time) จะมีผลทำให้

2.1) ออฟเซตจะหายไป

2.2) ผลตอบจะเร็วขึ้น (Fast response)

2.3) อัตราส่วนของช่วงกว้างการแกว่งเพิ่มขึ้นระบบขาดเสถียรภาพมากขึ้น

3) ผลของ D action

เมื่อให้ P และ I action คงที่แล้วเพิ่มเวลา T_D (Derivative time) ให้ยาวขึ้นจะมีผลทำให้

3.1) อัตราส่วนของช่วงกว้างการแกว่งลดลงระบบมีเสถียรภาพมากขึ้น

3.2) period ของการแกว่งสั้นลง

ตามปกติการใช้ค่าเวลา T_D ให้ยาวมีแนวโน้มที่จะทำให้ระบบมีเสถียรภาพมากก็จริง แต่ก็มีจุดอ่อนตรงตอบรับต่อ Noise ได้ง่าย ทำให้ผลตอบของระบบไวเกินไป ผลตอบของการควบคุมแบบต่างๆ ต่อการเปลี่ยนค่าเป้าหมายและการเกิดการรบกวนโดยตั้งค่า PB, T_I และ T_D

2.3.4 คุณภาพของการควบคุม

ดังที่ได้กล่าวมาแล้วว่า ระบบควบคุมที่ดีจะต้องมีคุณสมบัติคือ มีเสถียรภาพ ผลตอบเร็ว และไม่เกิดออฟเซต แต่จะใช้อะไรในการตัดสินว่า การควบคุมไหนจะดีที่สุดนั้น จะต้องมีเกณฑ์ตัดสินที่มีหลักเกณฑ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1) เกณฑ์ในการวัดที่ใช้ตัดสินคุณภาพการควบคุม มีดังนี้

1.1) อัตราส่วนช่วงกว้างการแกว่ง (Damping ratio criterion)

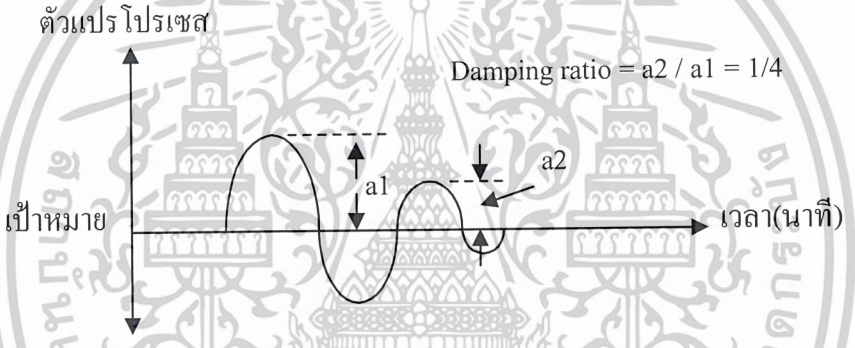
1.2) พื้นที่การควบคุมน้อยสุด (Minimum control area criterion)

1.3) ความสูงของโอเวอร์ชูต แรก (Overshoot amplitude criterion)

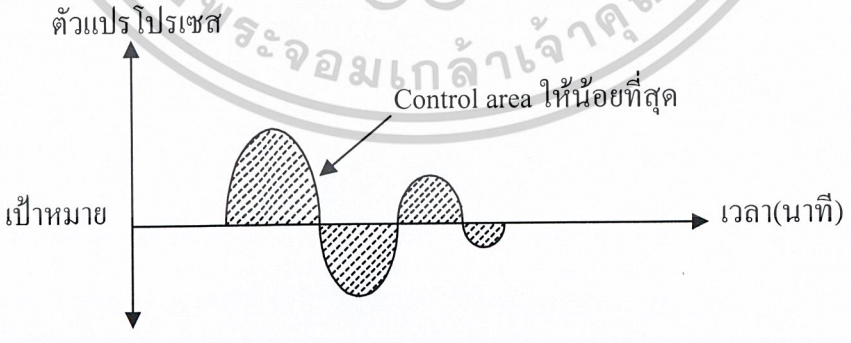
1.4) ผลตอบต่อการรบกวน (Minimum disturbance criterion)

1.5) ผลตอบเชิงความถี่ของระบบ (Frequency response criterion)

ในการควบคุมโปรเซสเกนอัตราส่วนช่วงกว้างการแกว่งเป็นที่นิยมใช้มากสุด เพราะสามารถวัดผลได้ง่าย การควบคุมที่ดีจะให้อัตราส่วนช่วงกว้างการแกว่ง 25% หรือ 1 ต่อ 4 ตามรูปที่ 2.27 และ 2.28



รูปที่ 2.27 25% Damping ratio



รูปที่ 2.28 พื้นที่การควบคุมน้อยสุด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3.5 วิธีการปรับค่า

ได้มีนักคณิตศาสตร์คิดวิธีที่จะหาทางตั้งค่า PID เพื่อให้ได้การควบคุมที่มีคุณภาพดีที่สุด หลายวิธี ดังตัวอย่างที่แสดงในตารางที่ 2.2 อย่างไรก็ตามในทางปฏิบัติก็ยังไม่มีวิธีใดที่ดีพอที่สามารถจะนำไปใช้ในทุกระบบได้

วิธีที่ใช้ในการปรับค่า PID สำหรับการควบคุมเพื่อให้ได้ 25% Damping ratio มี 3 วิธี คือ

1. Reaction Curve method
2. Ultimate Sensitivity method
3. Trial and error

ตารางที่ 2.2 การปรับอัตราของ PID

ชื่อผู้เสนอ	แบบ	แบบการควบคุม	Control Action			เกณฑ์
			PB (%)	T_D (นาที)	T_D (นาที)	
Ziegler Nichols (1942)	A.B.	P	2 PB	_	_	25% Damping Ratio
		PI	2.2 PB	0.83 pu	_	
		PID	1.7 PB	0.5 Pu	0.125 Pu	
lakahashi	A	P	101 KpL / T	_	_	Minimum Control area
		PI	111 KpL / T	3.3 L	_	
		PID	77 KpL / T	2.2 L	0.45 L	
Chien Hrones Reswick	A	P	333 KpL / T	_	_	20% Overshoot
		PI	286 KpL / T	1.2 T	_	
		PID	167 KpL / T	T	0.5 L	
Chien Hrones Reswick	A	P	143 KpL / T	_	_	No Overshoot
		PI	167 KpL / T	T	_	
		PID	105 KpL / T	1.35 T	0.47 L	
Chien Hrones Reswick	B	P	333 KpL / T	_	_	20% Overshoot
		PI	167 KpL / T	4 L	_	
		PID	105 KpL /T	2.4 L	0.4 L	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.2 (ต่อ) การปรับอัตราของ PID

ชื่อผู้เสนอ	แบบ	แบบการควบคุม	Control Action			เกณฑ์
			PB (%)	T_D (นาที)	T_D (นาที)	
Chien	B	P	143 KpL / T	_	_	Minimum
Hrones		PI	143 KpL / T	2.3 L	_	Control
Reswick		PID	83 KpL / T	2 L	0.42 L	area
Fuji	A	P	100 KpL / T	_	_	
Yoshikawa		PI L/T 1	167 KpL /	T+L	_	
		L/T 1	(T+L)	0 L	_	
		PID L/T = 1	250 KpL /	0.5 (T+L)	0.125	
		L/T 1	(T+2L)	0.25 L	(T+L)	
			133 KpL /		0.25 L	
			(T+3L)			
			200 KpL /			
			(T+L)			

1) Reaction Curve method (Transient response method)

วิธีการ

- ให้เปลี่ยนระบบควบคุมเป็นวงรอบเปิด (Open Loop)

- หา Process Characteristic โดยการเปลี่ยนค่าสัญญาณควบคุมไป ΔCS แล้วบันทึกรูปคลื่นของตัวแปรโปรเซส

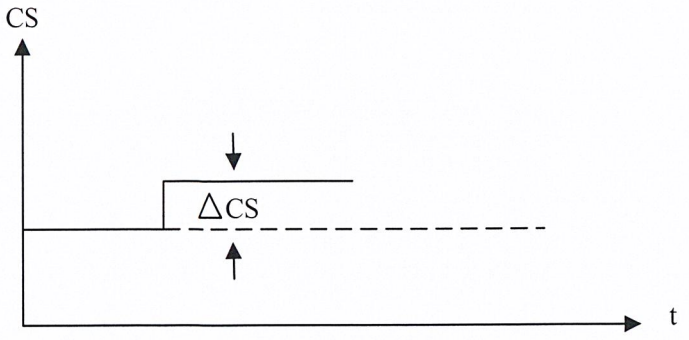
- หา Processd gain (K_p) , dead time (LE) และค่าคงตัวเวลา (TE)

จาก Process Characteristic จากรูปที่ 2.29 และ 2.30

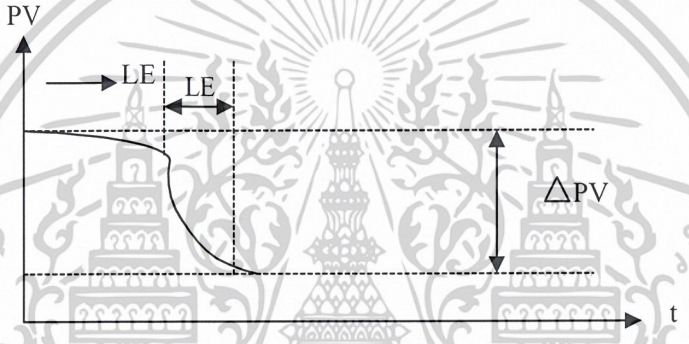
$$KP = \frac{?PV}{?CS} \tag{2.19}$$

- นำค่า K_p, LE, TE ที่หาได้ไปคำนวณหา PB, T_I และ T_D จากตารางที่ 2.2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.29 การหา Δ CS Reaction corve method



รูปที่ 2.30 การหา KP Reaction corve method

ตารางที่ 2.3 Reaction corve method

ชนิดการควบคุม	PB (%)	T_I (นาที)	T_D (นาที)
P	100 Kp. LE/TE	-	0
PI	110 Kp. LE/TE	3.3 LE	0
PID	83 Kp. LE/TE	2 LE	0.5 LE

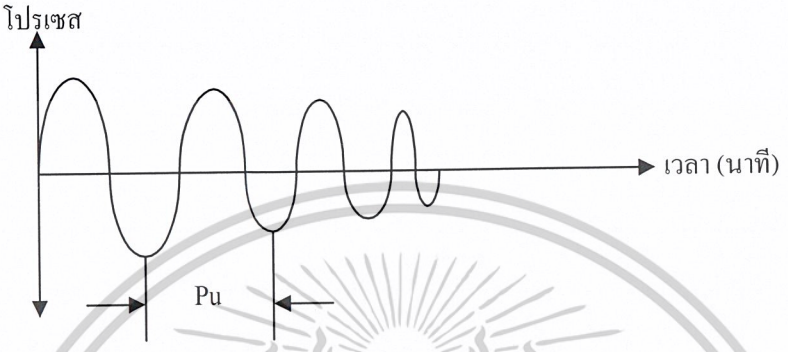
2) Ultimate sensitivity method

วิธีการ

- ให้ระบบควบคุมเป็นแบบวงรอบปิด (Closed loop)
- ตั้ง T_I สูงสุด (Max) และ T_D ต่ำสุด (0 หรือ min) ใช้ P action ในการควบคุมอย่างเดียว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ครั้งแรกตั้งค่า PB ไว้ที่ค่าสูงสุด แล้วลองลดค่า PB ลงมา ลองเปลี่ยนค่าเป้าหมายเพื่อผลตอบ ลดค่า PB ให้ต่ำลงเรื่อยๆ จนถึงค่าที่เมื่อเปลี่ยนค่าเป้าหมายไปเล็กน้อย จะทำให้โปรเซสเกิดการแกว่งต่อเนื่องไปตลอด ค่า PB ขณะนั้นเรียกว่า PB_u (Ultimate proportional band) ดังรูปที่ 2.31



รูปที่ 2.31 การวัด Pu

ตารางที่ 2.4 Ultimate Sensitivity method

ชนิดการควบคุม	PB	T_I	T_D
P	2 PB_u	∞	0
PI	2.2 Pbu	0.83 Pu	0
PID	1.7 PB_u	0.5 Pu	0.125 Pu

- หาคาบเวลาในการแกว่งให้เท่ากับ Pu (ตามรูปที่ 2.31)
- นำค่า PB_u และ Pu ที่หาได้ไปใช้คำนวณหาค่า PB, T_I ,และ T_D จากตารางที่ 2.4

3) Trial and error

เป็นวิธีการที่ใช้หลักการของ Ultimate Sensitivity method เพียงแต่ไม่ต้องคำนวณค่าและใช้การทดลองปรับค่าต่างๆ เพื่อหาผลตอบที่ดีที่สุด

3.1) P Control

1. ปรับตัวควบคุมไปที่ Manual mode
2. ปรับ PB ไปสูงสุด T_I สูงสุด และ T_D ต่ำสุด
3. ปรับค่าเป้าหมาย ไปสู่ค่าที่ต้องการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. ปรับ Manual Control จนตัวแปรโปรเซสหรือค่าที่วัดได้เท่ากับค่าเป้าหมาย

5. ปรับตัวควบคุมไปที่ Automatic mode

6. เปลี่ยนค่าเป้าหมายไปเล็กน้อย เมื่อค่าวัดเริ่มเปลี่ยนจึงลดค่าเป้าหมายมาสู่ที่เดิม

7. ลดค่า PB ลงมาและทำขั้น 6 ใหม่โดยสังเกตผลตอบของค่าวัด

8. ทำขั้น 6 และ 7 หลายๆ ครั้งจนได้อัตราส่วนช่วงกว้างการแกว่งของผลตอบเป็น 25% Damping ratio

3.2) PI Control

1. ทำเหมือน P Control จนขั้น 1 ถึง 8 เพื่อหาค่า PB ที่ดีที่สุด

2. ลดค่า T_I จนออฟเซตหายไป

3. เพิ่มค่า T_D ถ้าเกิดการแกว่งขึ้น

4. ทำขั้น 2 และ 3 จนกว่าจะให้ผลตอบ 25% Damping ratio

3.3) PID Control

1. ทำเหมือน P Control จากขั้น 1 ถึง 6

2. ลดค่า PB ลงมาจนเกิดการแกว่ง

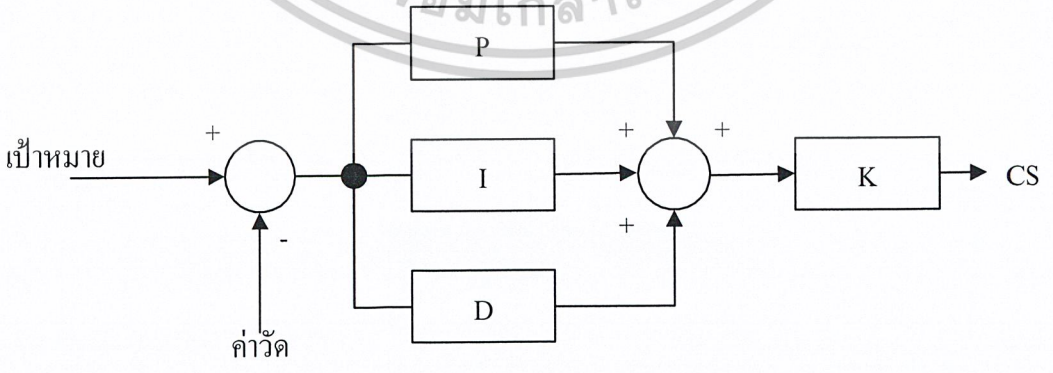
3. เพิ่ม T_D จนการแกว่งหยุด

4. ลดค่า PB จนเกิดการแกว่งใหม่อีก

5. ทำขั้น 2 ถึง 4 หลายครั้งจน T_D ไม่สามารถหยุดการแกว่งได้

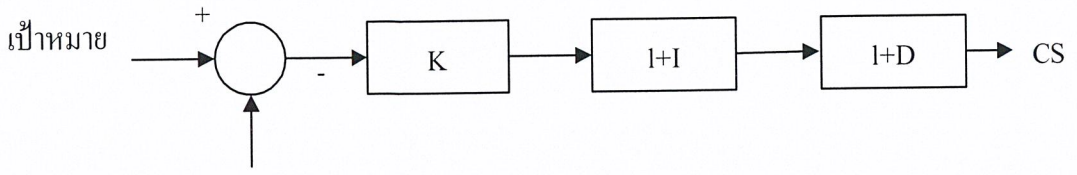
6. เพิ่มค่า PB จนหยุดการแกว่ง

7. ตั้งค่า T_I ให้เท่ากับ T_D ค่าสุดท้าย ($T_I = T_D$ ในกรณีที่ PID interfere Coefficient เท่ากับ 2)



รูปที่ 2.32 ตัวควบคุมแบบดีที่สุด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



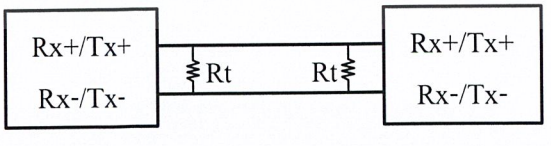
รูปที่ 2.33 ตัวควบคุมทั่วๆ ไป

2.4 อุปกรณ์ควบคุม

2.4.1 RS 485

RS485 เป็นรูปแบบการส่งข้อมูลผ่านสายเคเบิ้ลในลักษณะการสื่อสารแบบอนุกรม ซึ่งได้ปรับปรุงมาจากรูปแบบการสื่อสารแบบ RS232 และ RS422 โดย RS485 นี้มักนิยมใช้เป็นสายเพื่อสื่อสารระบบควบคุมภายในโรงงานอุตสาหกรรม เนื่องจากมีความสะดวกในหลายด้าน ทั้งด้านจำนวนคู่สาย คู่เดียวสามารถควบคุมได้ถึง 32 อุปกรณ์ สามารถส่งสัญญาณไปได้ระยะไกลและความเร็วในการสื่อสารข้อมูลสูงมากด้วย ซึ่งรูปแบบการส่งข้อมูลจำแนกให้เห็นได้ชัดมากขึ้นเมื่อ

รูปแบบRS 485 เป็นรูปแบบที่นำมาจากรูปแบบของ RS422 ทุกประการเพียงแต่มีการดัดแปลงให้ใช้สายสัญญาณเพียง 1 คู่สายบิดเกลียว โดยการใช้สายนี้เป็นทั้งส่งและรับในสายคู่เดียวกัน ซึ่งการที่จะทำงานในลักษณะนี้ได้นั้นที่ขั้วของอุปกรณ์ทุกตัวจะต้องมีการต่อบั๊ฟเฟอร์ที่สามารถปรับให้เป็น Hi Impedance ได้ คือตัดออกจากวงจรชั่วคราวได้นั่นเอง เพื่อใช้ในการบริหารสัญญาณ มักจะเป็นการสื่อสารในลักษณะ Client/Server คือ มีตัวแม่เป็นตัวหลักอยู่ 1ตัวคอยควบคุมดูแลและสั่งการทั้งหมดส่วนตัวลูกที่ต่ออยู่ทุกตัว ก็จะรอรับการสั่งงานจากตัวแม่เท่านั้น ตลอดเวลาตัวลูกแต่ละตัวจะต้องมีชื่อเรียกเพื่อที่ตัวแม่จะใช้ในการสั่งงาน



รูปที่2.34 การต่อ RS 485 แบบ 2-WIRE Half Duplex

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

RS 485 เป็นพอร์ตสำหรับการสื่อสารอนุกรมตัวหนึ่งที่สามารถติดต่อกับอุปกรณ์ตัวอื่นๆ ได้เป็นระยะทางไกลๆ ได้ถึง 1.2 กิโลเมตร หรือ 4,000 ฟุตและมีประโยชน์มากกับงานที่ต้องทำเป็น Network โดยใช้ 75176 เป็น Line Driver RS 485 เป็นการสื่อสารแบบ “Half Duplex”

ตารางที่ 2.5 เปรียบเทียบรูปแบบ RS 232, RS 422 และ RS 485

ข้อมูลของแต่ละรูปแบบ	RS 232	RS 422	RS 485
หมวดการทำงาน	Single Ended	Differential	Differential
จำนวนตัวลูกสูงสุด	1 Drive 1 Recp	1 Drive 10 Recp	1 Drive 32 Recp
ความยาวสูงสุดของสายเคเบิ้ล	50 ฟุต	4,000 ฟุต	4,000 ฟุต
อัตราความเร็วการส่งข้อมูล	20K b/sec	10M b/sec	10M b/sec
แรงดันส่งออกสูงสุด Output	+/-25 โวลต์	-0.25 ถึง +6 โวลต์	-7 ถึง +12 โวลต์
ค่าความต้านทาน Load	3K - 7K โอห์ม	100 โอห์ม	54 โอห์ม
ความไวในการรับ	+/-3 โวลต์	+/-200 มิลลิโวลต์	+/-200 มิลลิโวลต์
ความต้านทาน Input	3K - 7K โอห์ม	มากกว่า 4K โอห์ม	มากกว่า 12K โอห์ม

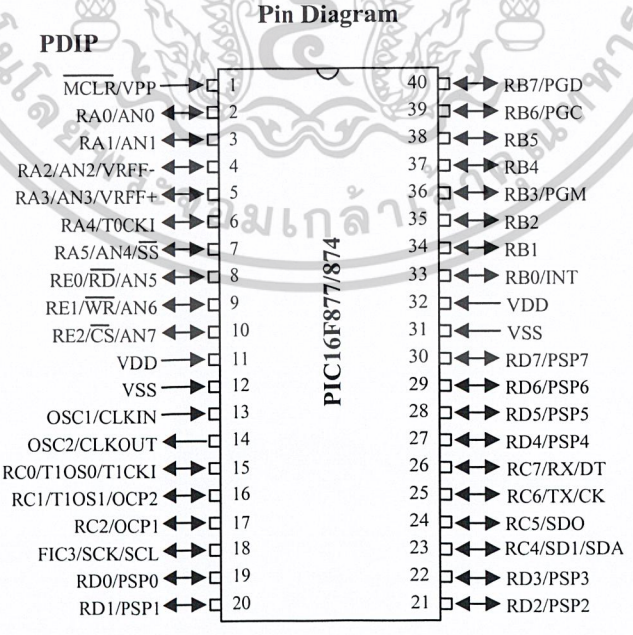
2.4.2 PIC16F877

PIC 16F877 เป็น CPU ที่มีความสามารถสูงพอสมควร ประกอบไปด้วยฟังก์ชั่นการทำงานต่างๆ มากมายสรุปคุณสมบัติคร่าวๆ ได้ดังนี้

- มี 35 Instruction คำสั่ง
- ในการปฏิบัติงานคำสั่งต่างๆ จะใช้ ไซเคิลเดียวและ 2 ไซเคิลในคำสั่งที่เป็นการกระโดด
- ความถี่สูงสุดที่ทำงานได้คือ 20 MHz (16F877-20/P)
- การทำงานจะเป็นลักษณะ Pipeline ทำให้มีการทำงานที่เร็วขึ้น
- หน่วยความจำโปรแกรม FLASH Program Memory มีขนาด 8k (14-Bit Words)
- หน่วยความจำข้อมูล (RAM) 368 Bytes
- หน่วยความจำข้อมูล (EEPROM) 256 Bytes
- สามารถตอบสนองการอินเตอร์รัพได้ถึง 14 แหล่ง
- มี STACK 8 ระดับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- มีเพาเวอร์ออนรีเซต (POR), เพาเวอร์อัปไทเมอร์ (PRRT) และ Oscillator Start-Up Timer
- Watchdog Timer
- สามารถเลือกการป้องกันข้อมูลได้ (Code Protection)
- มีโหมดประหยัดพลังงาน (Sleep Mode)
- เลือกโหมดของสัญญาณนาฬิกาได้หลายโหมด
- สามารถโปรแกรมโดยใช้แรงดัน +5 โวลต์ได้
- มีฟังก์ชั่นการโปรแกรมแบบ ICSP (In-Circuit Serial Programing)
- ทำงานที่ไฟเลี้ยง 2.0 โวลต์ ถึง 5.5 โวลต์
- กระแสทั้งซิงก์และซอร์สของพอร์ตคือ 25 มิลลิแอมป์
- มี Timer/Counter จำนวน 3 ตัว คือ Timer0, Timer1 และTimer2
- มีโมดูล Capture/Compare/PWM จำนวน 2 ชุด
- มี Analog to Digital Converter ความละเอียด 10 บิต 8 แชนแนล ภายในตัว
- มีโมดูลการสื่อสาร USART
- มีโมดูลตรวจจับระดับไฟเลี้ยง Brown-out reset (BOR)
- มีพอร์ต I/O 5 พอร์ตประกอบด้วย A, B, C, D และ E แต่ละพอร์ตจะมีจำนวนบิตไม่เท่ากัน ซึ่งรวมแล้วจะมี I/O จำนวน 33 บิต



รูปที่ 2.35 ตำแหน่งขาสัญญาณต่างๆ ของ PIC 16F877

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การจัดสรรหน่วยความจำของ PIC 16F877 นี้จะแบ่งออกเป็น 3 ส่วนคือ หน่วยความจำโปรแกรม หน่วยความจำข้อมูล (RAM) และหน่วยความจำข้อมูลที่เป็น EEPROM ซึ่งมีดังนี้

1) หน่วยความจำโปรแกรม

PIC 16F877 นี้มีโปรแกรมเคาน์เตอร์ ขนาด 13 บิต ซึ่งสามารถอ้างตำแหน่งของข้อมูลได้ถึง 8 กิโลเวิร์ด โดยจะมีตำแหน่งรีเซตเวคเตอร์ ที่ 0000h และอินเตอร์รัพท์ที่ 0004h ดังนั้นในการเขียนโปรแกรมจึงควรสงวนพื้นที่ส่วนนี้ไว้สำหรับการใช้งานอินเตอร์รัพท์ จากรูปที่ 2.36 จะเห็นได้ว่ามีพื้นที่ของสแต็ก 8 ระดับ และหน่วยความจำแบ่งออกเป็น 4 page (8 kwords) ซึ่งพื้นที่ในส่วนนี้มีไว้สำหรับเก็บข้อมูลคำสั่งทั้งหมดโดยโครงสร้างจะเป็นแบบแฟรช (Flash Memory) ทำให้ลบและเขียนใหม่ได้หลายครั้ง

2) หน่วยความจำข้อมูล

หน่วยความจำข้อมูลจะแบ่งออกเป็นพื้นที่ของ RAM หน่วยความจำใช้งานทั่วไป (Gerneral Purpose Register) ขนาด 360 ไบต์และพื้นที่ของรีจิสเตอร์ฟังก์ชั่นพิเศษ (Special Function Registers) ในการจัดวางพื้นที่จะแบ่งออกเป็น 4 แบงค์ ตั้งแต่แอดเดรส 00h ถึง 1FFh ดังรูปที่ 2.37

3) หน่วยความจำข้อมูล EEPROM

มีหน่วยความจำแบบ EEPROM จำนวน 256 ไบต์ โดยสามารถอ่านและเขียนในขณะทำงานปกติได้แต่ต้องไม่มีการ Enable Code Protect Bit โดยการเข้าถึงนั้นจะต้องทำผ่านรีจิสเตอร์พิเศษ ซึ่งต้องใช้ถึง 4 ตัวดังนี้

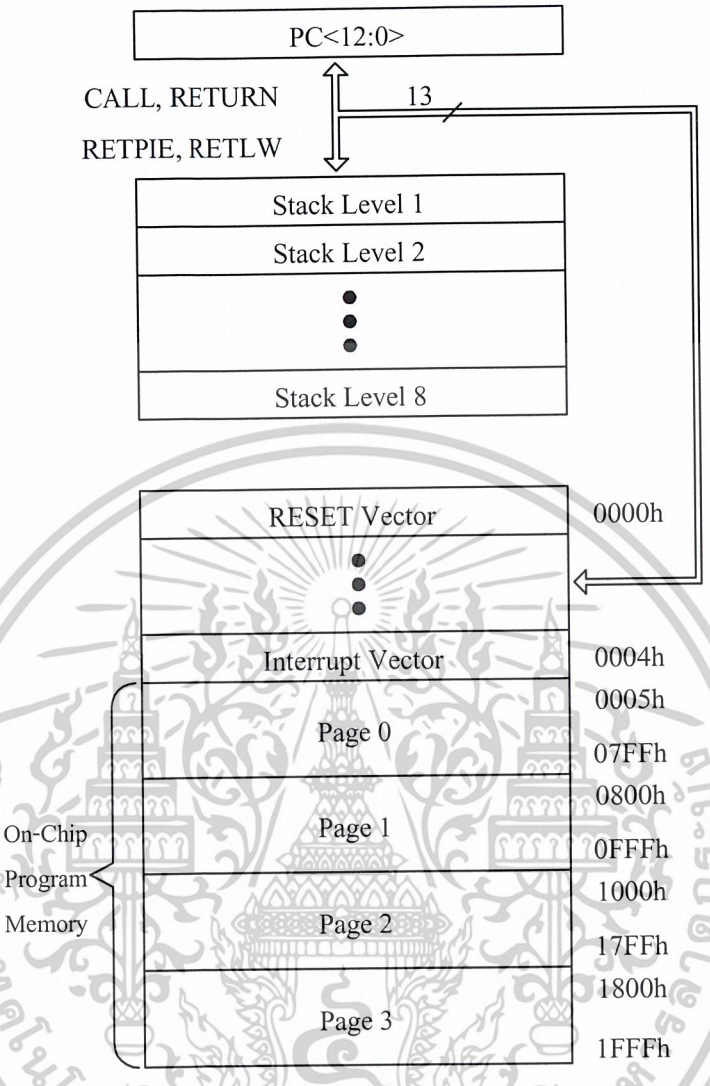
EECON1 : ควบคุมการเข้าถึงหน่วยความจำ

EECON2 : จัดลำดับการเขียนข้อมูล

EEDATA: เป็นบัฟเฟอร์ใช้เก็บข้อมูล 8 บิต สำหรับการอ่านและเขียน

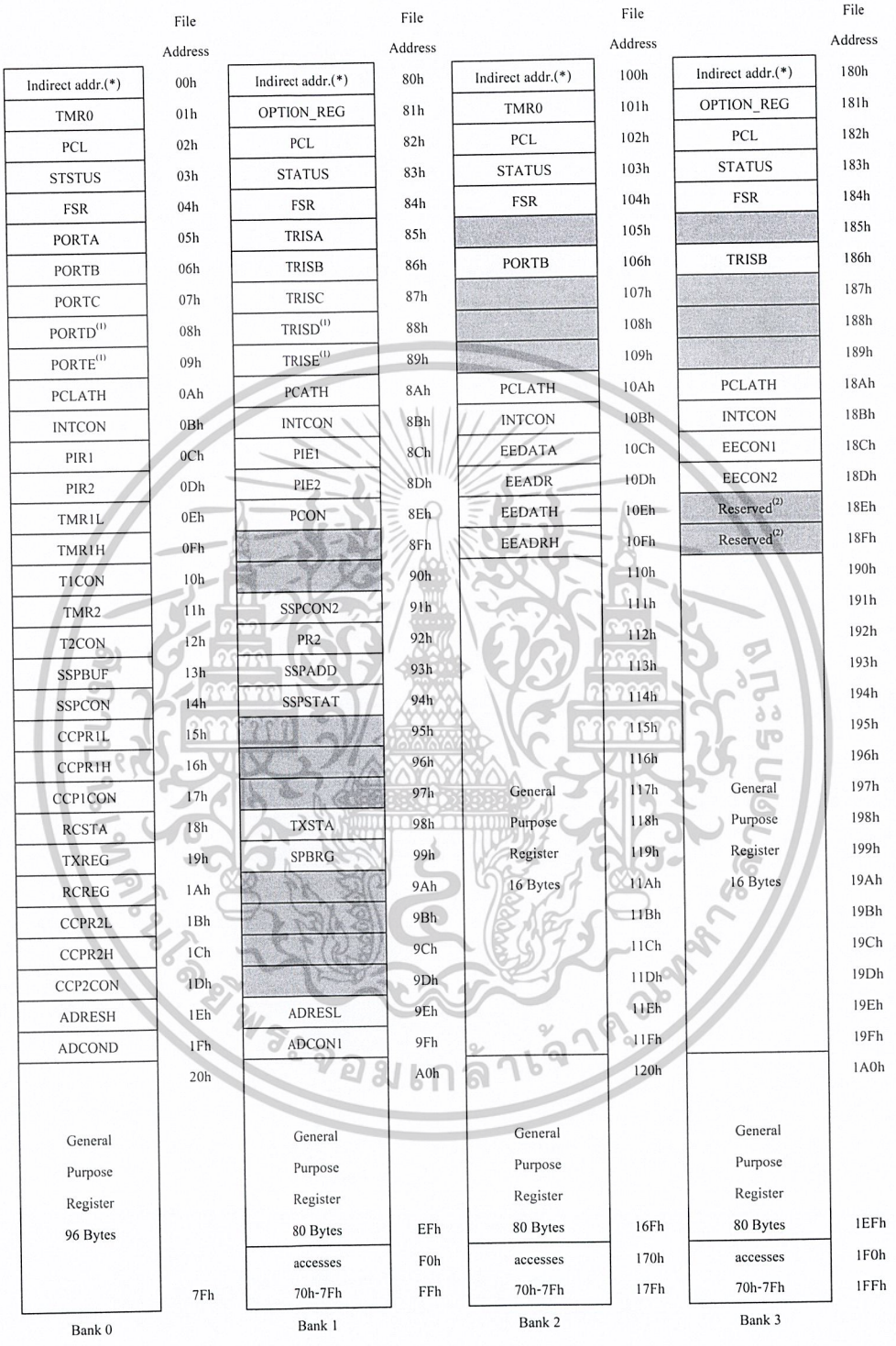
EEADR : รีจิสเตอร์ที่เก็บแอดเดรส 00h – FFh (256 ไบต์)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.36 การจัดสรรพื้นที่หน่วยความจำของ PIC 16F877

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



Unimplemented data memory locations, read as '0'.

* Not a physical register

Note 1: These registers are not implemented on the PIC16F876

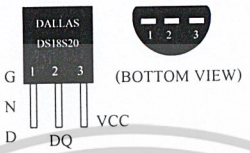
2: these registers are registers are reserved, maintain these registers clear

รูปที่ 2.37 การจัดวางพื้นที่ของหน่วยความจำข้อมูล

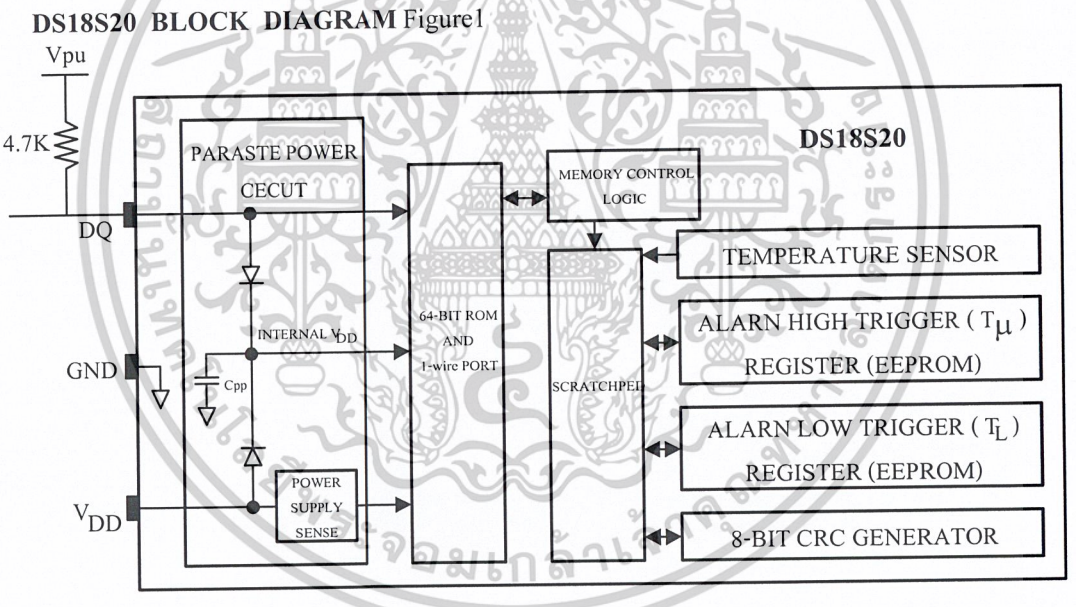
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4.3 ไอซีตรวจจับอุณหภูมิ DS18S20

DS 18S20 เป็นไอซีตรวจจับอุณหภูมิที่ใช้การติดต่อ ระบบบัสแบบหนึ่งสาย มีขาต่อใช้งานเพียง 3 ขา คือ DQ ซึ่งเป็นขาที่เชื่อมต่อกับระบบบัส ขาต่อไฟเลี้ยงภายนอก และขากราวด์ ดังรูปที่ 2.38



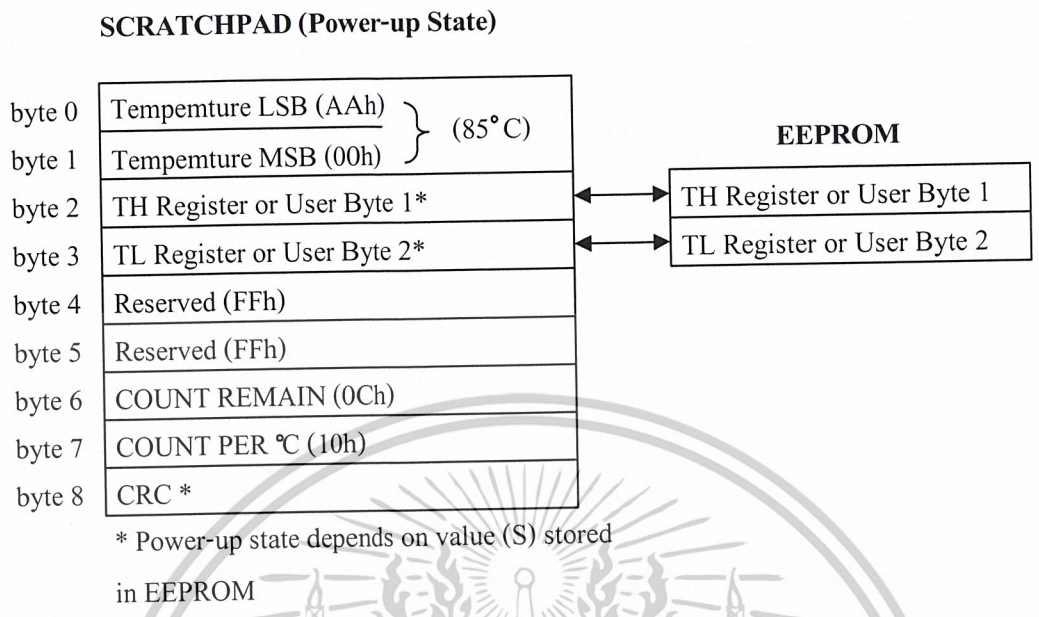
รูปที่ 2.38 การจัดขาของ ไอซี DS 18S20



รูปที่ 2.39 โครงสร้างการทำงาน

หัวใจสำคัญของ DS 18S20 อยู่ที่ตัวตรวจจับอุณหภูมิและหน่วยความจำความเร็วสูงที่เรียกว่า Scratchpad ซึ่งมีขนาด 9 ไบต์ มีการจัดสรรหน่วยความจำ ดังรูปที่ 2.40

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.40 การจัดสรรหน่วยความจำ

เมื่อวัดอุณหภูมิได้ก็จะนำค่าที่วัดได้มาเก็บไว้ใน Scratchpad ที่ไบต์ 0 และ 1 ทั้งนี้เนื่องจากไอซี DS 18S20 สามารถให้ข้อมูลได้ละเอียดถึง 16 บิต เมื่อนำข้อมูลมาแปลงเป็นข้อมูลเลขฐานสิบจึงสามารถแสดงความละเอียดของค่าอุณหภูมิได้ถึง 0.5 องศาเซลเซียส และ 0.9 องศาฟาเรนไฮต์ โดยมีย่านการวัดอุณหภูมิ –55 ถึง +125 องศาเซลเซียส หรือ 67 ถึง +257 องศาฟาเรนไฮต์ โดยค่าขององศาฟาเรนไฮต์ต้องใช้การแปลงหน่วยเข้ามาช่วย ใช้เวลาในการแปลงค่าอุณหภูมิเป็นข้อมูลดิจิตอล ประมาณ 200 มิลลิวินาที สามารถกำหนดขอบเขตของอุณหภูมิที่ทำการวัดได้และให้แจ้งเตือนเมื่อค่าของอุณหภูมิสูงขึ้นหรือลดลงถึงค่าที่กำหนด โดยค่าอุณหภูมิที่กำหนดนี้จะเก็บไว้ที่ Scratchpad ในไบต์ที่ 2 และ 3 ส่วนไบต์ที่ 8 นั้นมีไว้สำหรับบรรจุ Cyclic Redundancy Check (CRC) ซึ่งใช้ในการตรวจสอบความถูกต้องของข้อมูลใช้สมการดังนี้

$$CRC = X^8 + X^5 + X^4 + 1 \tag{2.20}$$

2.4.4 คำสั่งเพื่อควบคุมการทำงานของ DS 18S20

ในการติดต่อกับไอซี DS 18S20 จะมีคำสั่งที่ต้องส่งให้แก่ DS 18S20 เพื่อกำหนดรูปแบบการทำงาน คำสั่งที่ใช้มากที่สุด มีด้วยกัน 3 คำสั่งดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1) คำสั่งไม่ติดต่อกับหน่วยความจำรอม หรือ Skip ROM

เนื่องจากในการใช้งาน DS 18S20 โดยปกติแล้วจะมี DS 18S20 อยู่บนสายสัญญาณเพียงตัวเดียว จึงไม่จำเป็นต้องใช้ข้อมูลกำหนดแอดเดรส ดังนั้นจึงไม่จำเป็นต้องติดต่อกับหน่วยความจำเพื่ออ่านข้อมูล ข้อมูลของคำสั่ง Skip ROM ที่ต้องส่งให้ DS 18S20 คือ OCCH

2) คำสั่งแปลงอุณหภูมิ (Convert T)

มีค่าเท่ากับ 44H เมื่อส่งคำสั่งนี้ให้ DS 18S20 จะต้องทำการวนลูปอย่างน้อย 200 มิลลิวินาที เพื่อให้ DS 18S20 ได้ใช้เวลาในการเปลี่ยนแปลงเป็นข้อมูลดิจิตอลมาเก็บไว้ใน Scartchpad

3) คำสั่งอ่านข้อมูลจาก Scartchpad (Read Scartchpad)

มีค่าเท่ากับ 44H เมื่อส่งคำสั่งนี้ DS 18S20 จะทยอยส่งข้อมูลค่าอุณหภูมิออกมาทั้งหมด 9 ไบต์

2.5 การถ่ายเทความร้อน

ความร้อนสามารถเคลื่อนที่จากจุดๆ หนึ่งไปยังจุดอีกจุดหนึ่งได้ด้วย 3 วิธีด้วยกันคือ การนำ การพา และการแผ่รังสี การเคลื่อนที่ของความร้อนทั้ง 3 วิธีนี้ จะเกิดขึ้นได้ก็ต่อเมื่อมีความแตกต่างของอุณหภูมิเกิดขึ้น แต่กลไกในการที่จะทำให้เกิดการเคลื่อนที่ของความร้อนแต่ละวิธีแตกต่างกันดังนี้

2.5.1 การนำ

การนำความร้อน(Conduction) คือ วิธีการที่ความร้อนเคลื่อนที่จากบริเวณที่มีอุณหภูมิสูงไปยังบริเวณที่มีอุณหภูมิต่ำภายในตัวกลางเดียวกันหรือเป็นการเคลื่อนที่ของความร้อนระหว่างตัวกลางที่ติดกัน แต่มีอุณหภูมิต่างกันในการนำความร้อน ความร้อนจะเคลื่อนที่ผ่านโมเลกุลของสารโดยที่โมเลกุลไม่เคลื่อนที่ (อยู่นิ่ง) การนำความร้อนจะเกิดขึ้นได้ดีมากในตัวกลางที่เป็นของแข็ง การเคลื่อนที่ของความร้อนแบบการนำเกิดขึ้นบ้างในของเหลวและก๊าซ แต่มักจะแยกไม่ออกจากการเคลื่อนที่ของความร้อนแบบการพา ความร้อนเคลื่อนที่โดยการนำได้โดยการเคลื่อนที่ของอิเล็กตรอนจากจุดที่มีอุณหภูมิสูงไปสู่จุดที่มีอุณหภูมิต่ำ นอกจากนี้ความร้อนยังเคลื่อนที่ไปได้โดยการสั่นสะเทือนของโมเลกุลภายในของแข็งในลักษณะของพลังงานของความสั่นสะเทือน (vibrational energy) อีกด้วย

หลักการคำนวณเกี่ยวกับการนำความร้อนถูกตั้งขึ้นโดย โจเซฟ โฟริเออร์ (Joseph Fourier) นักวิทยาศาสตร์ชาวฝรั่งเศส โฟริเออร์ได้เสนอสมการ (2.21) ใช้สำหรับคำนวณอัตราการเคลื่อนที่ของความร้อนโดยการนำ ในปี ค.ศ. 1822 โดยอาศัยข้อมูลจากการทดลองไว้ดังนี้ คือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$Q_x = -kA \frac{dt}{dx} \tag{2.21}$$

โดยที่ k คือ ค่าการนำความร้อน (thermal conductivity) ของสารที่ความร้อนเคลื่อนที่ผ่านมีหน่วยเป็น W / mK ในระบบ SI และ But / ft hr°F ในระบบอังกฤษ A เป็นพื้นที่ตั้งฉากกับการเคลื่อนที่ของความร้อนและ $\frac{dt}{dx}$ เป็นอัตราการเปลี่ยนแปลงของอุณหภูมิกับระยะทาง

สมการนี้เรียกว่า Fourier rate equation ซึ่งเป็นสมการที่ได้มาโดยอาศัยผลการทดลอง จากสมการนี้สามารถหาสมการอื่นๆ เพื่อใช้ให้เป็นประโยชน์ในการคำนวณเกี่ยวกับการเคลื่อนที่ของความร้อนโดยการนำได้

2.5.2 การพา

การพา (Convection) คือวิธีการที่ความร้อนเคลื่อนที่ระหว่างผิวของของแข็งและของไหล ของไหลจะเป็นตัวพาความร้อนมาให้หรือพาความร้อนจากผิวของของแข็ง กลไกที่ทำให้เกิดการเคลื่อนที่ของความร้อนโดยการพาได้นั้น เกิดจากผลรวมของการนำความร้อน การสะสมพลังงานและการเคลื่อนที่ของของไหล การพายังแบ่งออกได้เป็น 2 ชนิดคือ การพาโดยการบังคับ (forced convection) และการพาตามธรรมชาติ (natural หรือ free convection)

การพาโดยการบังคับ คือการเคลื่อนที่ของความร้อนระหว่างผิวของของแข็งและของไหล โดยที่ของไหลถูกบังคับให้เคลื่อนที่ไปสัมผัสกับผิวของของแข็งโดยกลไกภายนอก เช่น พัดลมหรือเครื่องสูบน้ำ

การพาตามธรรมชาติ คือการเคลื่อนที่ของความร้อนระหว่างผิวของของแข็งและของไหล โดยที่ไม่มีกลไกใดๆ ที่ทำให้ของไหลเคลื่อนที่ แต่ของไหลที่อยู่ใกล้กับผิวของของแข็งก็อาจเคลื่อนที่ได้โดยแรงลอยตัวของของไหลเอง แรงลอยตัวนี้เกิดจากความแตกต่างของความหนาแน่นของของไหล เมื่อเกิดความแตกต่างของอุณหภูมิในชั้นของของไหลขึ้น

การคำนวณอัตราการเคลื่อนที่ของความร้อนโดยการพานั้นเป็นสิ่งที่ยุ่งยาก เมื่อพิจารณาแล้วมีหลายสิ่งหลายอย่างที่มีผลต่อการเคลื่อนที่ของความร้อนแบบการพานี้ เป็นต้นว่า คุณสมบัติต่างๆของของไหล เช่น ความหนาแน่น ความร้อนจำเพาะ ความหนืด ฯลฯ ความเร็วของของไหล ความแตกต่างระหว่างอุณหภูมิของพื้นผิวของของแข็งและของไหลเป็นต้น นิวตัน (Newton) ได้ตัดปัญหาความยุ่งยากเหล่านี้ โดยเสนอสมการสำหรับคำนวณอัตราการเคลื่อนที่ของความร้อน โดยการพาดังสมการ (2.22)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$q = h\,(T_h - T_c) \tag{2.22}$$

โดยที่ h คือสัมประสิทธิ์การพาความร้อน (heat transfer coefficient) ซึ่งได้รวมเอาความยุ่งยากทั้งหมดไว้ ถ้าสามารถรู้วิธีการคำนวณค่าสัมประสิทธิของการพาความร้อนก็สามารถคำนวณอัตราการเคลื่อนที่ของความร้อนโดยการพาได้ สมการสำหรับคำนวณค่าสัมประสิทธิของการพาความร้อนส่วนใหญ่จะเป็นสมการชนิดเอมไพริกัล (empirical equation)

h มีหน่วยเป็น W / m^2K ในระบบ SI และ But / ft^2 hr°F q คืออัตราการถ่ายเทความร้อนต่อหน่วยพื้นที่ของของแข็งที่สัมผัสกับของไหล T_h คืออุณหภูมิที่ร้อนกว่า (ของของไหลหรือพื้นผิวของของแข็ง) และ T_c คืออุณหภูมิที่เย็นกว่า (ของของไหลหรือพื้นผิวของของแข็ง)

2.5.3 การแผ่รังสี

ในการแผ่รังสี (Radiation) ความร้อนเคลื่อนที่ได้โดยมิต้องอาศัยตัวกลางดังเช่นในการนำและการพา ในการแผ่รังสีความร้อนจะเคลื่อนที่ได้ดีที่สุดในสูญญากาศ การที่จะอธิบายว่า ความร้อนเคลื่อนที่ได้อย่างไรนั้น เป็นสิ่งที่ยากที่จะสามารถอธิบายให้เห็นพฤติกรรมทางกายภาพได้ มีผู้พยายามเสนอทฤษฎีเกี่ยวกับการแผ่รังสีมาหลายรายแต่ไม่ค่อยเป็นที่น่าพอใจนัก ทฤษฎีที่เป็นที่ยอมรับมากที่สุดในขณะนี้เป็นทฤษฎีที่เสนอโดย ไอสไตน์ (Einstein) ซึ่งกล่าวว่า ในการแผ่รังสีความร้อนเคลื่อนที่โดยอาศัยกลไกของคลื่นแม่เหล็กไฟฟ้า

สำหรับวิศวกรวิธีการที่ความร้อนเคลื่อนที่โดยการแผ่รังสี อาจจะไม่มีความสำคัญเท่ากับการที่จะรู้วิธีคำนวณหาอัตราการเคลื่อนที่ของความร้อนโดยการแผ่รังสี ในปี 1884 สตีแฟนและโบลซ์แมน (Stefan and Boltzmann) ได้เสนอสมการในการคำนวณอัตราการเคลื่อนที่ของความร้อนสูงสุดโดยการแผ่รังสีจากวัตถุที่มีพื้นที่ A และอุณหภูมิ T ดังสมการ (2.23)

$$Q = \mathrm{s}\,AT^4 \tag{2.23}$$

โดยที่ σ คือค่าคงที่ของสตีแฟนและโบลซ์แมน (Stefan – Boltzmann constant) ซึ่งมีค่า 5.67×10^{-8} W / m^2K^4 ในระบบ SI และ 0.1717×10^{-8} Btu / hr ft^2 R^4 ในระบบอังกฤษสมการนี้สามารถที่จะนำไปใช้ในการคำนวณเพื่อประโยชน์ในการออกแบบเครื่องมือต่างๆ ได้

ในปัจจุบันปัญหาของการแผ่รังสีกำลังมีผู้สนใจมากเป็น เนื่องจากการขาดแคลนพลังงาน ซึ่งพลังงานแสงอาทิตย์ซึ่งได้จากการแผ่รังสีจากดวงอาทิตย์มายังโลกเป็นแหล่งพลังงานที่อาจจะนำมาใช้ทดแทนพลังงานที่กำลังใช้อยู่ในปัจจุบันได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

วัตถุที่จะให้ค่าการแผ่รังสีมีความร้อนสูงสุดตามสมการ (2.23) เรียกว่า วัตถุอุดมคติ (ideal body) หรือวัตถุดำ (black body) วัตถุที่มีอยู่ทั่วๆ ไปจะแผ่รังสีได้น้อยกว่าวัตถุอุดมคติการแผ่รังสีความร้อนจากวัตถุโดยทั่วๆ ไปที่มีอุณหภูมิ T และพื้นที่ A จะเขียนได้ดังนี้คือ

$$Q = \text{es}\, AT^4 \tag{2.24}$$

โดยที่ e คือ คุณสมบัติทางการแผ่รังสีความร้อนของวัตถุซึ่งเรียกว่า ค่าการแผ่รังสี (emissivity) ซึ่งเป็นปริมาณที่แสดงถึงประสิทธิภาพของการแผ่รังสีความร้อนของวัตถุ เมื่อเปรียบเทียบกับวัตถุที่สามารถแผ่รังสีความร้อนได้สูงสุด e จะมีค่าน้อยกว่า 1 เสมอ

สมการ (2.24) ใช้สำหรับหาอัตราการแผ่รังสีความร้อนจากวัตถุที่มีพื้นที่ A การที่จะหาอัตราการแลกเปลี่ยนความร้อน โดยการแผ่รังสีความร้อน ระหว่างวัตถุสองชิ้นที่มีอุณหภูมิต่างกันนั้นยุ่งยากมาก หากเราพิจารณากรณีการแลกเปลี่ยนความร้อน ระหว่างวัตถุเล็กที่มีพื้นที่ผิว A และค่าการแผ่รังสีความร้อน e กับวัตถุใหญ่ซึ่งครอบคลุมวัตถุเล็กอยู่ โดยที่วัตถุเล็กมีอุณหภูมิ T_h และวัตถุใหญ่มีอุณหภูมิ T_c เราจะเขียนอัตราการแลกเปลี่ยนความร้อนสุทธิของวัตถุทั้งสองได้ดังนี้คือ

$$Q_n = \text{es}\, A(T_h^4 - T_c^4) \tag{2.25}$$

เราอาจจะเขียนสมการสำหรับอัตราการแลกเปลี่ยนความร้อนสุทธิต่อไปได้ดังนี้คือ

$$Q_n = \text{es}\, A\,(T_h - T_c)\,(T_h + T_c)\,(T_h^2 + T_c^2) \tag{2.26}$$

ในหลายกรณีจะเป็นการสะดวกมากถ้าเราเขียนสมการให้อยู่ในรูปดังนี้คือ

$$Q_n = h_r A\,(T_h - T_c) \tag{2.27}$$

$$h_r = \text{es}\,(T_h + T_c)\,(T_h^2 + T_c^2) \tag{2.28}$$

โดยที่ h_r คือ สัมประสิทธิของการแผ่รังสีความร้อน (radiation heat transfer coefficient)

2.5.4 การถ่ายเทความร้อน โดยการพาความร้อนและการแผ่รังสีความร้อนรวมกัน

เมื่อการถ่ายเทความร้อนเกิดจากการพาและการแผ่รังสีความร้อนพร้อมกันและอัตราการถ่ายเทความร้อนโดยการพาและการแผ่รังสีมีความร้อนไม่ต่างกันมากนัก การวิเคราะห์หาค่าของ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การถ่ายเทความร้อนยุ่งยากมาก แต่ในบางกรณีอาจจะประมาณค่าของการถ่ายเทความร้อนโดยการรวมอัตราการถ่ายเทความร้อนโดยการพาและการแผ่รังสีเข้าด้วยกันได้ เช่น การไหลของก๊าซที่ได้จากการสันดาป ซึ่งประกอบด้วย CO_2 CO และ H_2O ซึ่งมีอุณหภูมิ T_h ในท่อซึ่งมีอุณหภูมิผนัง T_c การถ่ายเทความร้อนจากก๊าซไปสู่ผนังท่อจะเกิดขึ้นทั้งโดยการพาความร้อนและการแผ่รังสีความร้อนหากอัตราการถ่ายเทความร้อนโดยการแผ่รังสีความร้อนมีค่าไม่มากนัก อัตราการถ่ายเทความร้อนรวมจะหาได้โดยประมาณ จากการรวมอัตราการถ่ายเทความร้อนทั้งสองวิธีเข้าด้วยกันดังนี้

$$\begin{aligned} q &= h\,(T_h - T_c) + h_r\,(T_h - T_c) \\ &= (h + h_r)\,(T_h - T_c) \\ &= h_{cr}\,(T_h - T_c) \end{aligned} \tag{2.29}$$

โดยที่ h_{cr} คือสัมประสิทธิของการถ่ายเทความร้อนรวมของการพาและการแผ่รังสีความร้อน

2.6 การวัดอุณหภูมิ

อุณหภูมิเป็นสิ่งที่เกี่ยวข้องภายในชีวิตประจำวันและบ่อยครั้งที่อุณหภูมิเป็นตัวแปรที่ถูกควบคุมในกระบวนการอุตสาหกรรมหลายอย่าง การวัดอุณหภูมิและการรักษาอุณหภูมิ ให้เที่ยงตรงนั้นเป็นสิ่งสำคัญยิ่ง อุณหภูมิเป็นพารามิเตอร์ที่สำคัญในเทอร์โมไดนามิก การส่งถ่ายความร้อนและเป็นส่วนประกอบที่สำคัญของการทำงานหรือปฏิกิริยาเคมี

2.6.1 ชนิดและวิธีการวัดอุณหภูมิ

แม้ในการวัดทุกชนิดสามารถทำได้โดยการเปรียบเทียบค่าที่ต้องการวัดกับมาตรฐานที่ได้มีการนิยามไว้ แต่ก็มีหลายวิธีของการกระทำการเปรียบเทียบดังกล่าว เพื่อให้ได้มาซึ่งค่าที่ต้องการจากการวัด นอกจากนั้นการที่เรานิยามคำเกี่ยวกับวิธีการวัดจะช่วยให้เราสามารถสื่อสารแนวคิดโดยใช้คำที่ยอมรับกันโดยทั่วไปซึ่งการวัดแบ่งออกเป็น 2 ชนิด

1) การวัดโดยตรง (Direct Comparison)

เป็นวิธีการที่เราได้ค่าการวัดโดยตรง เราไม่จำเป็นที่จะต้องคำนวณเพิ่มเติม เพียงแต่อาศัยความสัมพันธ์ระหว่างปริมาณที่ต้องการวัดกับปริมาณอื่นที่เราต้องการวัดจริง ค่าที่ต้องการก็จะได้ทันทีในรูปของข้อมูลเดิม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2) การวัดโดยทางอ้อม (Indirect Comparison)

เป็นการวัดที่เราจะได้ค่าโดยการผ่านตัวกลางที่มีหน่วยแตกต่างกันออกไป ซึ่งเชื่อมต่อกันในบางลักษณะ เช่น วิธีการวัดระยะโดยใช้การเคลื่อนที่ของพัลซ์เป็นวิธีทางอ้อมเนื่องจากต้องมีการคำนวณหาระยะทางจากความสัมพันธ์ระหว่างจำนวนพัลซ์กับการเคลื่อนที่ ซึ่งมีข้อสังเกตเกี่ยวกับการวัดทางอ้อมก็คือ ผลลัพธ์สุดท้ายนั้นจะได้มาจากผลของการวัดโดยตรงหลายๆ ปริมาณ

2.6.2 อุณหภูมิและมาตราวัดอุณหภูมิ

อุณหภูมิเป็นหน่วยมูลฐานที่สำคัญและใช้มากที่สุดค่าหนึ่งในอุตสาหกรรม หน่วยของอุณหภูมิที่ใช้ในปัจจุบันซึ่งอยู่มีหลายหน่วยด้วยกันที่สำคัญคือ เซลเซียส (Celcius) เคลวิน (Kelvin) ฟาเรนไฮต์ (Fahrenheit)

1) เซลเซียส เป็นหน่วยวัดอุณหภูมิที่ตั้งชื่อตามนักวิทยาศาสตร์ชาวสวีเดน Anders Celcius (ค.ศ. 1701 - 1744) โดยเซลเซียสพบว่า ณ ความดันบรรยากาศน้ำบริสุทธิ์จะมีจุดคงที่ทางอุณหภูมิอยู่สองจุดคือ จุดเยือกแข็งของน้ำและจุดที่น้ำเดือด เขาจึงได้กำหนดจุด 0 องศาที่จุดเยือกแข็งของน้ำและ 100 องศาที่จุดเดือดของน้ำ

2) ฟาเรนไฮต์ เป็นหน่วยวัดอุณหภูมิที่ตั้งชื่อตามนักวิทยาศาสตร์ชาวดัตช์ Daniel Gabriel Fahrenheit (ค.ศ. 1686 - 1736) โดยฟาเรนไฮต์ได้พยายามหาจุดต่ำสุดของอุณหภูมิโดยการทดลองกับสารต่างๆ หลายอย่างและพบว่าจุดเยือกแข็งของแอมโมเนียมคลอไรด์เป็นจุดต่ำสุดของอุณหภูมิเท่าที่เขาทดลองได้ จึงกำหนดจุดนี้เป็นจุด 0 องศา ส่วนจุดบนของสเกลนั้นเขาพบว่าอุณหภูมิของร่างกายมนุษย์เป็นจุดที่อุณหภูมิคงที่ จึงกำหนดจุดนี้เป็น 96 องศา สาเหตุที่ไม่กำหนดจุดบนสเกลเป็น 100 องศา เพราะต้องการให้มีค่าเป็นสัดส่วนทวีคูณของ 12 ตามหน่วยอื่นๆ ที่นิยมในสมัยนั้น

ทั้งหน่วยเซลเซียสและฟาเรนไฮต์เป็นมาตราวัดอุณหภูมิที่ได้จากการทดลองหาสภาวะคงที่ทางอุณหภูมิของสาร ณ จุดต่างๆ ที่อยู่ในย่านบรรยากาศของพื้นโลกและแบ่งสเกลให้เป็นไปตามความสะดวกเหมาะสม ต่อมา นักวิทยาศาสตร์ชาวอังกฤษ ลอร์ดเคลวิน เป็นผู้ค้นคิดหน่วยของอุณหภูมิขึ้นในปี ค.ศ. 1851 เรียกว่า มาตราเคลวิน โดยได้กำหนดจุดอุณหภูมิศูนย์สมบูรณ์ (Zero Absolute Temperature) ขึ้น ณ จุดที่เป็นอุดมคติ (Ideal) นี้ อิเล็กตรอนในอะตอมของสารต่างๆ จะหยุดโคจรรอบนิวเคลียสโดยไม่มีพลังงานความร้อนเหลืออยู่ในสารนั้นๆ อีกต่อไป ในทางปฏิบัติไม่สามารถทำให้เย็นจัดถึงจุดนี้ได้ (กำหนดได้จากการคำนวณ) และแบ่งช่วงของมาตราตามมาตราเซลเซียส ระบบหน่วยสากล (The System International d' Unites) ซึ่งเรียกว่า SI ได้กำหนดหน่วยสากลของอุณหภูมิเทอร์โมไดนามิกเป็นมาตราเคลวินและหน่วยของอุณหภูมิทั่วๆ ไปเป็นเซลเซียส

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.6 เปรียบเทียบจุดอ้างอิง (Reference) ของมาตราต่างๆ

จุดอ้างอิง	° C	° F	K
จุดศูนย์สัมบูรณ์	-273.15	-459.67	0
จุดเยือกแข็งของน้ำ	0	32	273.15
จุดเดือดของน้ำ	100	212	373.15

2.6.3 สเกลอุณหภูมิสัมบูรณ์

สเกลอุณหภูมิสัมบูรณ์เป็นสิ่งที่แสดงอุณหภูมิศูนย์หน่วยกับวัสดุที่ไม่มีพลังงานความร้อน นั่นคือไม่มีการสั่นของโมเลกุล สเกลอุณหภูมิสัมบูรณ์ที่นิยมใช้กันโดยทั่วไปมีอยู่ 2 สเกลคือ เคลวิน ในหน่วยของเคลวินและสเกลแรงดันในหน่วยขององศาแรงคิน สเกลเหล่านี้แตกต่างกันเพียงแต่ปริมาณของพลังงานที่ถูกแทนด้วยหนึ่งหน่วยของการวัด ดังนั้น จึงมีความสัมพันธ์ที่เป็นสัดส่วนกันโดยตรงระหว่างอุณหภูมิในหน่วยขององศาแรงดันกับอุณหภูมิในหน่วยของเคลวิน ดังตารางที่ 2.6 แสดงค่าต่างๆ ของอุณหภูมิในหน่วยเคลวินและองศาแรงคินที่แสดงค่าต่างๆ ของอุณหภูมิในหน่วยเคลวินและองศาแรงคินที่จุดปรับเทียบ (Calibration Points) จากตารางเราสามารถแปลงอุณหภูมิในหน่วยขององศาแรงคินเป็นเคลวินได้และแปลงจากเคลวินเป็นองศาแรงคินได้ เพื่อการแปลงหน่วยของอุณหภูมิเราจะสังเกตเห็นความแตกต่างของอุณหภูมิระหว่างจุดที่น้ำมีสถานะเป็นของแข็ง / ของเหลว และจุดที่น้ำมีสถานะเป็นของเหลว / แก๊ส คือ 100 K และ 180 ° R, 1 K = 180 / 100 (1 ° R) = 9 / 5 (1 ° R) ดังนั้น การแปลงหน่วยระหว่างสเกลทั้งสอง จะทำได้โดย

$$T(K) = \frac{5}{9} T(^{\circ}R) \tag{2.30}$$

เมื่อ $T(K)$ คืออุณหภูมิในหน่วยของ K

$T(R)$ คืออุณหภูมิในหน่วยของ (° R)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.7 จุดปรับเทียบกับสเกลของอุณหภูมิ

Calibration Point	อุณหภูมิ			
	K	°R	°F	°C
ไม่มีพลังงานความร้อน	0	0	-459.6	-273.5
ออกซิเจน ของเหลว / แก๊ส	90.18	162.3	-297.3	-182.97
น้ำ ของแข็ง / ของเหลว	273.15	491.6	32	0
น้ำ ของเหลว / แก๊ส	373.15	671.6	212	100
ทองคำ ของแข็ง / ของเหลว	1,336.15	2,405	1,945.50	1,063

2.6.4 สเกลอุณหภูมิเทียบเคียง

สเกลอุณหภูมิเทียบเคียงจากสเกลอุณหภูมิสัมบูรณ์ เพียงแต่มีการเลื่อนแกนอุณหภูมิศูนย์หน่วย ดังนั้นแม้ว่าสเกลเหล่านี้แสดงอุณหภูมิเท่ากับศูนย์หน่วย พลังงานความร้อนของตัวอย่างที่ถูกทดสอบจะไม่เท่ากับศูนย์ สเกลเหล่านี้คือเซลเซียส (มีความสัมพันธ์กับเคลวิน) และฟาเรนไฮต์ (มีความสัมพันธ์กับแรงคิน) โดยอุณหภูมิถูกแสดงในหน่วยของ $^\circ$C และ $^\circ$F ตามลำดับ ตารางที่ 2.6 ได้แสดงจุดปรับเทียบ ต่างๆ ของสเกลเหล่านี้ด้วยปริมาณความร้อนที่แทนที่ด้วย 1 $^\circ$C ก็จะเหมือนกับที่แสดงโดย 1K แต่อุณหภูมิศูนย์ได้มีการเลื่อนไปในสเกลเซลเซียส นั่นคือ

$$T(^\circ C) = T(K) - 273.15 \tag{2.31}$$

ในทำนองเดียวกันขนาดของอุณหภูมิ 1 $^\circ$F ก็จะเหมือนกับขนาดของอุณหภูมิ 1 $^\circ$R แต่สเกลเลื่อนสเกลไปนั่นคือ

$$T(^\circ F) = T(^\circ R) - 459.6 \tag{2.32}$$

ในการแปลงหน่วยของอุณหภูมิจากเซลเซียสไปเป็นฟาเรนไฮต์ เราจะพบว่าสเกลทั้งสองมีความแตกต่างในขนาดขององศาเช่นเดียวกับใน K และ °R และสเกลทั้งสองมีการเลื่อนไป 32 หน่วยดังนั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$T(^{\circ}F)=\frac{9}{5}T(^{\circ}C)+32 \qquad (2.33)$$

2.6.5 การสอบเทียบ

เพื่อนิยามถึงสเกลของอุณหภูมิ ชุดของจุดปรับเทียบจะถูกนำมาใช้ พลังงานความร้อนเฉลี่ยต่อโมเลกุลจะถูกนิยามได้เป็นอย่างดี ด้วยสภาวะสมดุลย์ที่เกิดขึ้นระหว่างสถานะของ ของแข็ง ของเหลว หรือแก๊สของวัสดุบริสุทธิ์หลายๆ ชนิด ดังเช่น สภาวะสมดุลย์ที่เกิดขึ้นระหว่างของแข็งและของเหลวของสารบริสุทธิ์ เมื่ออัตราของการเปลี่ยนแปลงสถานะเหมือนกันทั้งสองทิศทางคือ ของเหลวไปเป็นของแข็งและของแข็งไปเป็นของเหลว ส่วนหนึ่งของจุดปรับเทียบมาตรฐานคือ

1. อ็อกซิเจน สภาวะสมดุลย์ระหว่างของเหลว / แก๊ส
2. น้ำ สภาวะสมดุลย์ระหว่างของแข็ง / ของเหลว
3. น้ำ สภาวะสมดุลย์ระหว่างของเหลว / แก๊ส
4. ทองคำ สภาวะสมดุลย์ระหว่างของแข็ง / ของเหลว

สเกลของอุณหภูมิต่างๆ ถูกนิยามด้วยการกำหนดค่าอุณหภูมิทางตัวเลขโดยการเปรียบเทียบกับจุดปรับเทียบ อย่างไรก็ตามสเกลต่างๆ มีความหมายแตกต่างกันใน 2 ประการคือ

1. ตำแหน่งศูนย์ของอุณหภูมิ
2. ขนาดของหนึ่งหน่วยการวัด นั่นคือ พลังงานความร้อนเฉลี่ยต่อโมเลกุลถูกแทนด้วยหนึ่งหน่วยของสเกล

2.7 มอเตอร์กระแสตรง

มอเตอร์กระแสตรง (DC MOTOR) จะนิยมใช้กับงานเล็กๆ ที่มีระดับแรงดันไฟฟ้าแรงต่ำ เช่น มอเตอร์ในรถแข่งของเด็กเล่น มอเตอร์สตาร์ทเครื่องยนต์ และมอเตอร์ในโรงงานอุตสาหกรรม เป็นต้น

มอเตอร์เป็นอุปกรณ์ไฟฟ้า ที่ใช้ในการแปลงพลังงานไฟฟ้าไปเป็นพลังงานกล โดยการนำพลังงานที่ได้นี้ไปขับเครื่องจักรอื่นๆ ต่อไป ความเร็วมอเตอร์สามารถกำหนดได้โดย

1. แรงบิดของโหลด
2. จำนวนขั้วของมอเตอร์
3. ความถี่ของแหล่งจ่ายไฟที่ใช้กับมอเตอร์
4. แรงดันที่ใช้กับมอเตอร์

ความเร็วของมอเตอร์สามารถหาได้จากสูตร ดังต่อไปนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$N = \frac{120f}{P} \tag{2.34}$$

เมื่อ N = ความเร็วรอบ

f = ความถี่

P = จำนวนขั้ว

จากสูตร จะพบว่าถ้าความถี่ของแหล่งจ่ายไฟเปลี่ยนแปลงไปก็มีผลทำให้มอเตอร์มีความเร็วเปลี่ยนแปลงไปด้วยแต่เมื่อทำการเปลี่ยนความถี่ โดยให้แรงดันคงที่จะมีผลทำให้เกิดฟลักซ์แม่เหล็กเพิ่มขึ้นจนอิ่มตัวซึ่งอาจทำให้มอเตอร์ร้อนจนเกิดความเสียหายได้ ดังนั้นจึงทำการเปลี่ยนแรงดันควบคู่ไปกับความถี่ด้วย และการที่จะเปลี่ยนความถี่ของแหล่งจ่ายสามารถทำได้โดยการใช้อินเวอร์เตอร์

ในการบังคับให้ดีซีมอเตอร์ทำงาน มีอยู่ 2 แนวทางที่นิยมใช้กันคือ การใช้รีเลย์และการใช้ทรานซิสเตอร์ในการควบคุมความเร็วของดีซีมอเตอร์นิยมใช้สัญญาณเป็น PWM โดยหากต้องการให้หมุนก็จะจ่ายที่พัลซ์ที่ความถี่สูงและหากต้องการให้หมุนช้าก็จะจ่ายพัลซ์ที่ความถี่ต่ำ

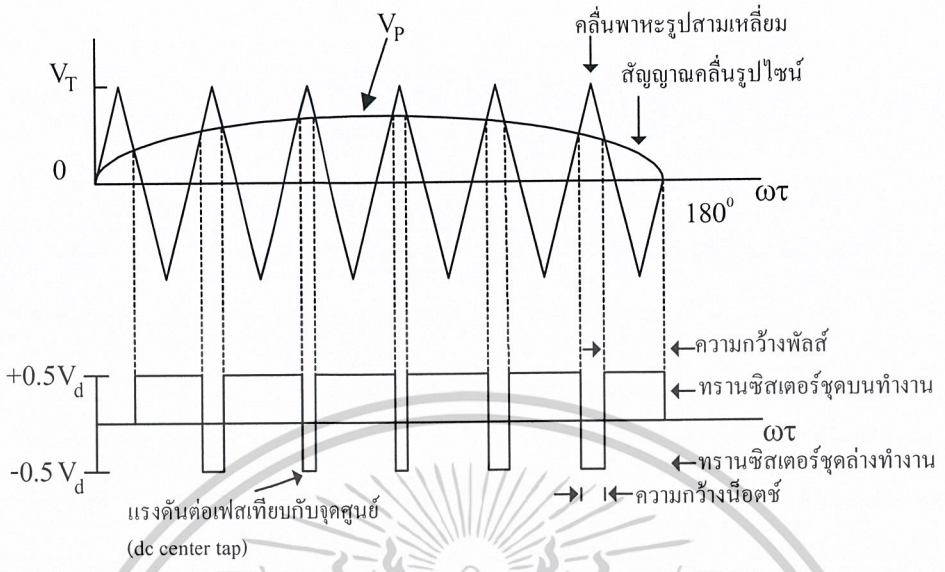
2.8 PWM (Pulse Width Modulation)

2.8.1 รูปคลื่น PWM แบบไซน์

การเอาสัญญาณของรูปคลื่นไซน์มามอดูเลตกับสัญญาณของรูปสามเหลี่ยม ซึ่งสัญญาณรูปคลื่นไซน์นี้ในทางสื่อสารถือว่าเป็นสัญญาณของข้อมูล แต่ในทางอิเล็กทรอนิกส์กำลังเราให้เป็นสัญญาณความถี่มูลฐาน ซึ่งเป็นความถี่เดียวกันกับความถี่มูลฐานที่มอเตอร์จะได้รับ ส่วนสัญญาณคลื่นสามเหลี่ยมในทางสื่อสารเรียกว่าคลื่นพาหะซึ่งในทางอิเล็กทรอนิกส์กำลังก็เรียกสัญญาณนี้ว่าคลื่นพาหะเช่นกัน

สัญญาณทริกเกอร์ที่เกทซึ่งเกิดจากการเปรียบเทียบของสัญญาณอ้างอิง (รูปคลื่นไซน์) กับสัญญาณพาหะ (สามเหลี่ยม) ดังรูปที่ 2.41

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.41 หลักการสร้างสัญญาณ PWM

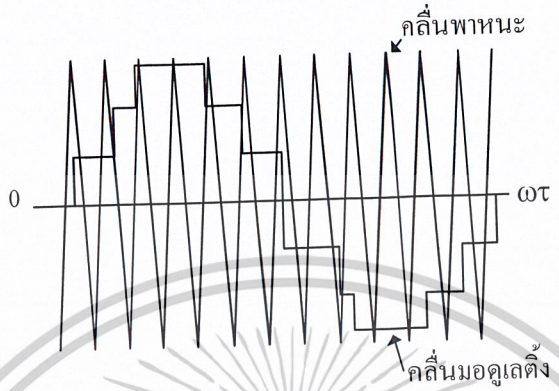
จากรูปจะเห็นว่าสัญญาณ PWM ได้มาจากการที่เอาสัญญาณรูปคลื่นไซน์ความถี่เท่ากับความถี่ที่เราต้องการเอามามอดูเลตกับคลื่นพาหะรูปสามเหลี่ยมซึ่งมีความถี่สูงกว่าคลื่นรูปไซน์หลายเท่า สัญญาณ PWM ที่ได้เกิดจากการเปรียบเทียบจุดตัดของสัญญาณทั้งสอง จะเห็นได้ว่าจุดตัดนี้จะเป็นตัวกำหนดการเปิด-ปิดของอุปกรณ์กำลัง

จากรูปที่ 2.41 เราเรียกความกว้างระหว่างพัลซ์ว่า ความกว้างพัลซ์ (pulse width) และเราเรียกช่องว่างระหว่างพัลซ์ว่า ความกว้างนอตซ์ (notch width) นอกจากนี้เรายังสามารถเปลี่ยนความถี่และแรงดัน rms ของรูปคลื่น PWM นี้ได้ด้วยการเปลี่ยนความถี่และขนาดของสัญญาณรูปไซน์หรือสัญญาณมอดูเลติ้งการเปลี่ยนความถี่ของสัญญาณรูปไซน์จะทำให้ความถี่มูลฐานของรูปคลื่น PWM เปลี่ยน แต่การเปลี่ยนขนาดแอมพลิจูดของสัญญาณรูปไซน์จะทำให้ค่าแรงดัน rms ของ PWM เปลี่ยนซึ่งถ้าเราควบคุมอย่างเป็นสัดส่วนกันผลก็คือทำให้ V/f คงที่

การมอดูเลตด้วยคลื่นรูปไซน์ถือเป็นพื้นฐานที่สำคัญ เข้าใจง่าย และมีประสิทธิภาพ แต่ปัญหาก็คือรูปคลื่น PWM ที่จะมีความกว้างของพัลซ์ไม่สมมาตรตลอดรูปคลื่น ทำให้ยากในกรณีที่จะเขียนโปรแกรมคอมพิวเตอร์เพื่อสร้างรูปคลื่นดังกล่าว ปัจจุบันผู้สร้างมักใช้คอมพิวเตอร์มาเป็นตัวควบคุมสั่งการและการผลิตคลื่น PWM มาจากไมโครโปรเซสเซอร์ ซึ่งจะทำงานตามโปรแกรมที่เขียนไว้ ไม่ได้เกิดจากการมอดูเลตจริงๆ ด้วยเหตุผลดังกล่าวจึงมีผู้คิดค้นนำเอาหลักการของการสุ่ม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวอย่างและคงไว้ (sampling and hold) มาดัดแปลงรูปคลื่นไซน์ซึ่งเป็นเชิงเส้นให้กลายเป็นคลื่นขั้นบันไดแบบไซน์ที่มีการเปลี่ยนระดับแรงดันที่เป็นแบบแผนและแน่นอนดังแสดงในรูปที่ 2.42



รูปที่ 2.42 หลักการของ PWM ที่มีการสุ่มตัวอย่างแบบเอกรูป (Uniform sampling PWM principle)

ผลของการมอดูเลตจะได้คลื่น PWM ที่มีความกว้างพัลซ์เกิดความสมมาตรกันทั้งสองด้าน ซึ่งง่ายต่อการเขียนโปรแกรมให้ไมโครโปรเซสเซอร์ผลิตสัญญาณดังกล่าวออกมา

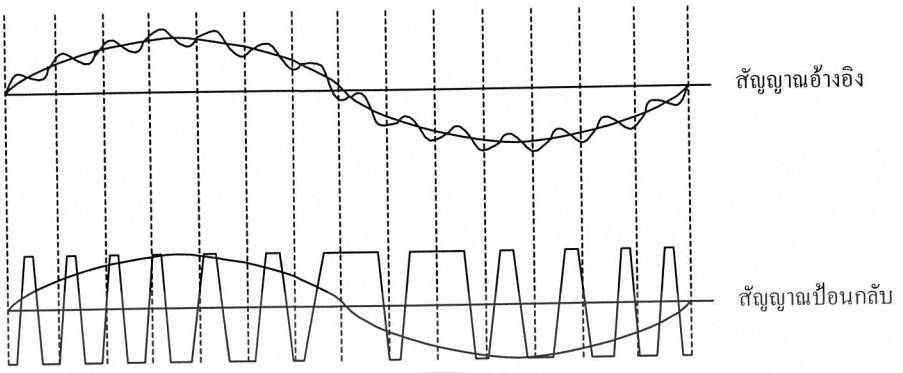
ข้อดีอีกข้อหนึ่งของสัญญาณ PWM ก็คือแก้ข้อเสียในเรื่องของบรเวณช่วงเปลี่ยนจากความเป็นเชิงเส้นสู่ความไม่เป็นเชิงเส้น

2.8.2 โครงงานการควบคุมวงจรอิเล็กกำลังด้วยคอมพิวเตอร์

เทคนิคการทำ PWM (Pulse Width Modulation) มีหลายวิธีที่ใช้กับอินเวอเตอร์ชนิดจ่ายแรงดันโดยควบคุมกระแสโหลดให้เข้าใกล้รูป Sine ได้มากที่สุดจากโครงงานเดิมเลือกใช้เทคนิคเดลต้ามอดดูเลชั่นซึ่งเป็นอีกเทคนิคหนึ่งในการสร้าง PWM

หลักการทำงานของเทคนิคเดลต้ามอดดูเลชั่นที่มีฮิสเตอร์รีซีสคงที่ โดยสัญญาณรูป Sinusoid อ้างอิงจะถูกเปรียบเทียบกับสัญญาณป้อนกลับที่เกิดจากการอินทิเกรต (ใช้วงจรกรองความถี่ต่ำ) สัญญาณเอาท์พุต PWM ซึ่งผลต่างของสัญญาณป้อนกลับและสัญญาณอ้างอิงจะเกิดสัญญาณผิดพลาดและผ่านวงจรฮีสเตอร์รีซีสคอมพาราเตอร์ (Hysteresis Comparator) ทำให้เกิดสัญญาณพัลซ์ PWM แต่รูปสัญญาณ PWM ที่ใช้ในการขับเกตของอุปกรณ์สวิตชิ่งจะสังเกตได้ว่าความกว้างของพัลซ์ PWM มีขนาดไม่เท่ากันตลอดคาบของคลื่นฟันดาเมนทอลเนื่องจากมีการรวมกันของความถี่ซึ่งเป็นของเสียของเทคนิคนี้ ดังรูปที่ 2.43

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

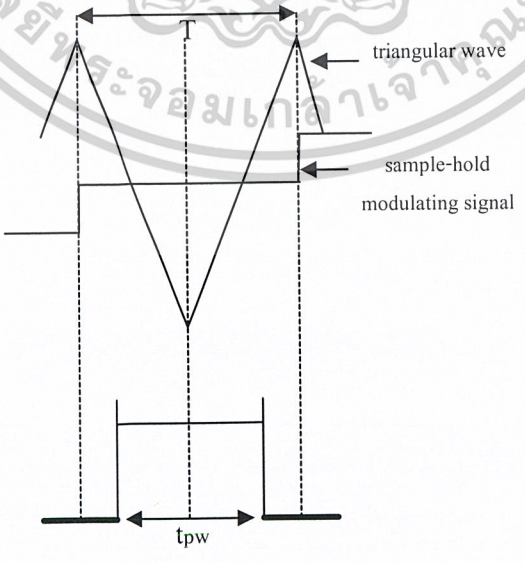


รูปที่ 2.43 สัญญาณ PWM จากโครงการEE99-61

จากรูปจะเห็นว่าสัญญาณป้อนกลับที่ได้จากการอินทริเกรต จะเคลื่อนที่ติดตามสัญญาณรูปไซน์อยู่ระหว่างขอบเขตบนและล่าง (ขอบเขตฮีสเตอรีซีส)

2.8.3 การสร้างสัญญาณPWMด้วยไมโครคอนโทรลเลอร์

การสร้างสัญญาณ PWM ด้วยไมโครคอนโทรลเลอร์(AVR)นั้นทำได้ด้วยวิธีการคำนวณความกว้างของสัญญาณ PWM ซึ่งจะถูกใช้เป็นข้อมูลที่จะใช้เขียนโปรแกรมเพื่อให้ไมโครคอนโทรลเลอร์สร้างสัญญาณ PWM ออกมา โดยในการคำนวณความกว้างของสัญญาณ PWM นี้ได้มาจากการพิจารณาวิธีการสร้างสัญญาณPWM จากการเปรียบเทียบสัญญาณ sample-hold modulating signal กับ triangular wave ดังรูปที่ 2.44



รูปที่ 2.44 การเปรียบเทียบสัญญาณ sample-hold modulating signal กับ สัญญาณtriangular

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในการสร้างสัญญาณ PWM นี้อยู่ภายใต้เงื่อนไขที่ให้อัตราส่วนของแรงดันกับความถี่มีค่าคงที่โดยให้ maximum output voltage อยู่ที่ความถี่ 50 Hz ความกว้างของสัญญาณ PWM นี้แสดงด้วยสมการ

$$t_{PW} = \frac{T}{2} + \frac{K}{2} p[\sin(\omega_m T_1)] \tag{2.35}$$

โดยที่ ω_c = angular frequency of carrier signal

ω_m = angular frequency of modulating signal

$P = \omega_c / \omega_m$

จากสมการ (2.35) จะถูกเปลี่ยนเป็นสมการที่จะนำไปใช้ในการเขียนโปรแกรมซึ่งใช้ระบบตัวเลขฐาน 16 สมการที่ใช้ในการเขียนโปรแกรมนี้แสดงด้วยสมการ

$$T_{PW} = \frac{T}{2} + DE \frac{SIN}{FF} \tag{2.36}$$

จากสมการที่ (2.36) จะพบว่ามีส่วนประกอบอยู่ 2 ส่วนโดยในส่วนแรกของสมการ (2.36) นั้นการกำหนดค่าของTซึ่งเป็นความกว้าง 1 คาบของสัญญาณ PWM 1 พัลซ์ก็คือการกำหนดความถี่ของสัญญาณ sample-hold modulating signal โดยสัญญาณนี้จะมีทั้งหมด 360 sample (ค่าไซน์เปลี่ยนจาก 0 –360 องศา) ซึ่งก็จะได้ พัลซ์ซึ่งเป็นสัญญาณ PWM ออกมาทั้งหมด 360 พัลซ์เช่นเดียวกัน และในส่วนที่ 2 จะเป็นค่าที่เปลี่ยนตามค่ามุมของสัญญาณไซน์ ซึ่งข้อมูลทั้งสองส่วนของสมการ (2.36) จะถูกใช้เป็นข้อมูลในโปรแกรมสร้างสัญญาณ PWM ต่อไปจะเป็นการกล่าวถึงลักษณะของข้อมูลทั้งสองส่วนในสมการ (2.36)

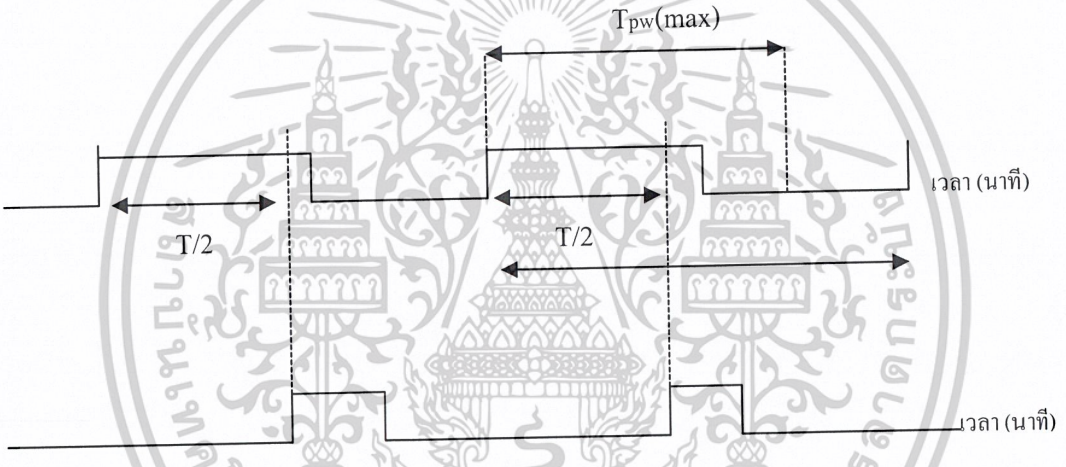
ในส่วนแรกนั้นค่า T/2 จะทำการคำนวณจาก T = T_S/360 (T_S คือคาบเวลาของสัญญาณ sample-hold modulating) ซึ่งจะคำนวณหาค่า T ตั้งแต่ความถี่ 1 – 50 Hz เมื่อได้ค่าT แล้วจะต้องเปลี่ยนเป็น T/2 แล้วเปลี่ยนเป็นค่าจำนวนนับ (N) ซึ่งจะใช้ Timer 0 เป็นตัวนับโดย N=T*CK (CK คือสัญญาณ clock ภายในตัวไมโครคอนโทรลเลอร์ โดย AVR มีค่า 8 MHz) แล้วเปลี่ยนค่า N เป็นระบบเลขฐาน 16

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในส่วนที่สองนั้นค่า SIN เป็นค่าของสัญญาณไซน์(0-360 องศา)ซึ่งถูกเปลี่ยนให้อยู่ในระบบเลขฐาน 16 โดยจะคูณกับค่าคงที่ DE (เลขฐาน16) ซึ่งข้อมูลทั้งสองส่วนนี้จะถูกนำไปใช้ในการประมวลผลจากโปรแกรมสร้าง สัญญาณ PWM

2.8.4 ลักษณะของสัญญาณ PWM ที่สร้างจากไมโครคอนโทรลเลอร์

สัญญาณ PWM ที่อยู่ในช่วงของสัญญาณไซน์ 0-180 องศา;T_{PW}(positive),จะมีค่าเป็นไปตามสมการ (2.36) โดยสัญญาณ PWM จะทำงาน ที่จุดเริ่มต้นของคาบเวลา T ส่วนลักษณะของสัญญาณ PWM ที่อยู่ในช่วงของสัญญาณไซน์ 181-359 องศา; T_{PW}(Negative),จะมีค่าความกว้างของ พัลซ์เป็น $T_{PW}(\text{Negative}) = T_{PW}(\text{max}) - T_{PW}(\text{positive})$:ซึ่งจะเริ่มONที่เวลา T/2



รูปที่ 2.45 T_{PW}(positive) และ T_{PW}(Negative)

โดยสัญญาณ PWM ที่ได้จากไมโครคอนโทรลเลอร์นี้เป็นสัญญาณ PWM 3 ช่องสัญญาณขนาด 5 โวลต์ โดยมีเฟสต่างกัน 120 องศา ซึ่งก่อนที่จะนำสัญญาณ PWM ป้อนให้อินเวอเตอร์ จะต้องเพิ่มระดับของสัญญาณ PWM เป็น 12 โวลต์ ก่อนโดยใช้ไอซี SN74LS07 ซึ่งเป็น open-collector buffer แล้วจึงนำสัญญาณ PWM ป้อนให้กับอินเวอเตอร์

2.9 ฮีตเตอร์

2.9.1 หลักการของฮีตเตอร์

ฮีตเตอร์ เป็นอุปกรณ์ทำความร้อนในอุตสาหกรรมที่มีหลักการพื้นฐานคือ เมื่อมีกระแสไหลผ่านลวดตัวนำที่มีค่าความต้านทานสูงลวดตัวนำจะร้อน ดังนั้นลวดที่ใช้ผลิตฮีตเตอร์จะต้องมี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คุณสมบัติเหนียวและทนอุณหภูมิได้สูง สำหรับลวดฮีตเตอร์ของ TIC เป็นลวด Khantal (นิกเกิ้ล: โครเมียม / 80:20) จากประเทศสวีเดน ทนอุณหภูมิได้ 1250 องศาเซลเซียส เป็นลวดผลิตฮีตเตอร์ที่ดีที่สุดในโลก ส่วนประกอบอื่นๆ ในการผลิตฮีตเตอร์มีดังนี้

- ฉนวนแม็กเซียมออกไซด์ (Mgo) จาก USA มีค่าความนำไฟฟ้าต่ำแต่นำความร้อนดีมาก ทำหน้าที่กั้นกลางระหว่างลวดฮีตเตอร์กับปลอกโลหะ เพื่อป้องกันไม่ให้มีกระแสรั่ว (Leak Current) จากลวดฮีตเตอร์ออกไปยังผิวโลหะ จุดสำคัญคือห้ามมีความชื้นในฉนวนเด็ดขาดเพราะจะทำให้ค่าความนำไฟฟ้าสูงขึ้น หากมีความชื้นแก้ไขได้โดยการอบในเตาอบ

- ท่อสแตนเลส 304 และ 316 จากออสเตรเลีย เลือกที่มีความหนามากเป็นพิเศษ ทำให้ทนอุณภูมิได้สูงกว่าปกติและไม่มีตะเข็บด้านนอกของผิวท่อ จึงไม่มีปัญหาการปริแตกตามแนวตะเข็บท่อฮีตเตอร์เมื่ออุณภูมิสูง

- Insulation Tester เป็นเครื่องทดสอบความเป็นฉนวนของฮีตเตอร์ เพื่อให้แน่ใจว่าในการใช้งานจริงจะไม่มีกระแสรั่วจากลวดฮีตเตอร์สู่ผิวโลหะซึ่งอาจเป็นอันตรายต่อผู้ใช้ได้ มาตรฐาน TIC ได้กำหนดการทดลองแรงดันที่ 1500 VAC และค่าความเป็นฉนวนต้องมากกว่า 500 MΩ

2.9.2 ชนิดของฮีตเตอร์

ฮีตเตอร์แบ่งออกเป็นชนิดต่างๆ ตามลักษณะการใช้งานที่แตกต่างกันดังนี้

ฮีตเตอร์แท่ง คือ Cartridge Heater ใช้ให้ความร้อนกับวัสดุที่เป็นของแข็ง เช่น เหล็กและโลหะต่างๆ ตัวอย่างการใช้งานเช่น งานบรรจุหีบห่อ งานขึ้นรูปพลาสติก

ฮีตเตอร์ครีบหรือ Finned Heater และฮีตเตอร์ท่อกลม ใช้ให้ความร้อนกับอากาศ เช่น ใช้ในห้องอบแห้ง ในเตาอบ

ฮีตเตอร์จุ่มหรือ Immesion Heater หรือบางทีเรียกว่า ฮีตเตอร์ต้มน้ำใช้ให้ความร้อนกับของเหลวทุกชนิด ตัวอย่างการใช้งานเช่น งานต้มน้ำ ต้มน้ำมัน งานผสมสาร

บอบบิ้นฮีตเตอร์ (Bobbin Heater) ใช้ให้ความร้อนของเหลวเหมือนฮีตเตอร์จุ่ม

ฮีตเตอร์อินฟราเรด (Infrared Heater) ใช้ให้ความร้อนกับวัตถุโดยไม่ต้องสัมผัสโดยตรงไม่เหมาะกับวัตถุที่มีลักษณะมันวาว เนื่องจากวัตถุมันวาวจะมีคุณสมบัติสะท้อนแสงทำให้ไม่สามารถดูดซับรังสีอินฟราเรดได้อย่างเต็มที่ที่ติดตั้งในเตาอบหรือเหนือคอนเวเยอร์ได้

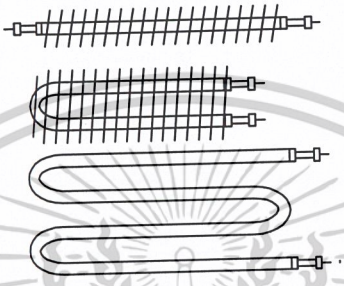
ฮีตเตอร์รัดท่อหรือ Band Heater ใช้ให้ความร้อนกับของเหลวที่อยู่ในท่อ

ฮีตเตอร์แผ่นหรือ Strip Heater ใช้ให้ความร้อนโดยแนบกับวัตถุโดยตรง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.9.3 ฮีตเตอร์ครีบและฮีตเตอร์ท่อกลม

ฮีตเตอร์ครีบ (Finned Heater) ทำจาก Tubular Heater ที่ดัดเป็นรูปต่างๆ และสามารถเพิ่มแผ่นครีบม้วนติดกับท่อฮีตเตอร์อย่างต่อเนื่องจากปลายด้านหนึ่งไปอีกด้านหนึ่ง ส่วนของแผ่นครีบที่เพิ่มขึ้นมาจะทำให้ฮีตเตอร์ สามารถถ่ายเทความร้อนได้เร็วขึ้น ส่วนฮีตเตอร์ท่อกลมคือ Tubular Heater ที่ใช้ให้ความร้อนโดยตรงโดยไม่ติดครีบ



รูปที่ 2.46 ฮีตเตอร์ครีบ

ฮีตเตอร์ครีบและฮีตเตอร์ท่อกลมใช้กับงานต่อไปนี้

- ใช้ในเตาอบ
- ใช้ในท่อ
- ใช้กับเครื่องปรับอากาศ

การติดตั้งสามารถทำได้ 2 วิธี คือติดตั้งแบบให้ความร้อนโดยตรงและแบบส่งผ่านความร้อนจากห้องเผาไปยังห้องอบโดยใช้ลมร้อน เป็นฮีตเตอร์ที่ใช้กับอากาศจึงไม่ควรใช้กับของเหลว เนื่องจากจะเกิดตะกรันจับที่ครีบของฮีตเตอร์ทำให้ความร้อนไม่สามารถถ่ายเทได้ ในกรณีที่ให้ความร้อนกับอากาศที่ไม่หมุนเวียน จึงควรเลือกวัสดุที่จะใช้ทำฮีตเตอร์ เป็นอินโคลอย เนื่องจากมีคุณสมบัติถ่ายเทความร้อนได้ดีและทนอุณหภูมิได้สูงกว่าชนิดอื่น

2.9.4 การคำนวณวัตต์ของฮีตเตอร์

ฮีตเตอร์เป็นอุปกรณ์ทำความร้อนที่มีค่าความต้านทานสูงของลวดตัวนำ ดังนั้นในการผลิตฮีตเตอร์จึงมีการคำนวณหาวัตต์ของฮีตเตอร์เพื่อให้มีคุณสมบัติตามต้องการ ดังสมการที่ (2.37)

$$W = V \quad D \quad S \quad (\Delta_T - 30) \tag{2.37}$$

โดยที่ W = วัตต์ของฮีตเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

D = Spectific Weight

S = Spectific heat capacity

Δ_T = อุณหภูมิที่เราต้องการ

T = วินาที

V = แรงดันไฟฟ้า

ตารางที่ 2.8 บอกเส้นผ่าศูนย์กลาง ความยาว แรงดัน กำลังวัตต์

เส้นผ่าศูนย์กลาง (มิลลิเมตร)	ความยาว (เมตร)	แรงดัน (โวลต์)	กำลังวัตต์ (วัตต์)
6.5	35	24,110,220	50 - 130
	50		50 - 180
	65		50 - 200
	80		50 - 270
	95		50 - 320
	110		50 - 360

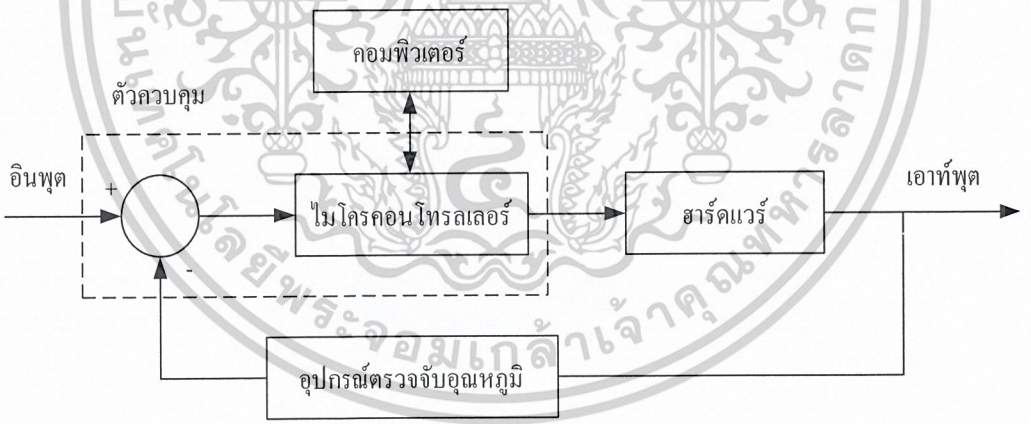
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3
การออกแบบ การสร้าง และการทำงาน

เนื้อหาในบทนี้กล่าวถึงการออกแบบและสร้างชุดสาธิตการควบคุมอุณหภูมิภายในห้องจำลองแบบวงปิด โดยนำทฤษฎีและหลักการที่ได้กล่าวมาแล้วในบทที่ 2 มาประกอบการออกแบบและสร้างวงจรที่ใช้ในชุดสาธิตการควบคุมอุณหภูมิในห้องจำลองแบบวงปิด

การออกแบบ การสร้าง และการทำงานของชุดสาธิตการควบคุมอุณหภูมิภายในห้องจำลองแบบวงปิดอธิบายได้ดังรูปที่ 3.1 จากหลักการทำงานดังกล่าวสามารถแบ่งการออกแบบและสร้างเป็นส่วนดังนี้ การออกแบบโครงสร้าง การออกแบบตัวรับอุณหภูมิ และการออกแบบวงจรควบคุมภายในระบบ และการออกแบบโปรแกรมควบคุมชุดสาธิตการควบคุมอุณหภูมิภายในห้องจำลองแบบวงปิด

3.1 การออกแบบโครงสร้าง



รูปที่ 3.1 แผนผังการทำงานของชุดสาธิตการควบคุมอุณหภูมิภายในห้องจำลองแบบวงปิด

จากรูปที่ 3.1 เมื่อเริ่มต้นการวัดย่านการวัดอุณหภูมิอยู่ที่ 40-70 องศาเซลเซียส สมมุติค่าอุณหภูมิที่จะทำการปรับแต่งที่ 50 องศาเซลเซียส ไมโครคอนโทรลเลอร์จะรับค่าจากคอมพิวเตอร์แล้วทำการควบคุมด้วยค่า P (Proportional) สั่งให้ควบคุมฮาร์ดแวร์ แล้วอุปกรณ์ตรวจจับอุณหภูมิจะทำการวัดอุณหภูมิมาเปรียบเทียบค่าเป้าหมาย เมื่ออุณหภูมิที่ได้ไม่ตรงตามค่าเป้าหมายที่ต้องการไมโครคอนโทรลเลอร์จะทำงานอย่างต่อเนื่อง แต่เนื่องด้วยกิริยาการควบคุมแบบพี ค่อนข้างจะมีค่า

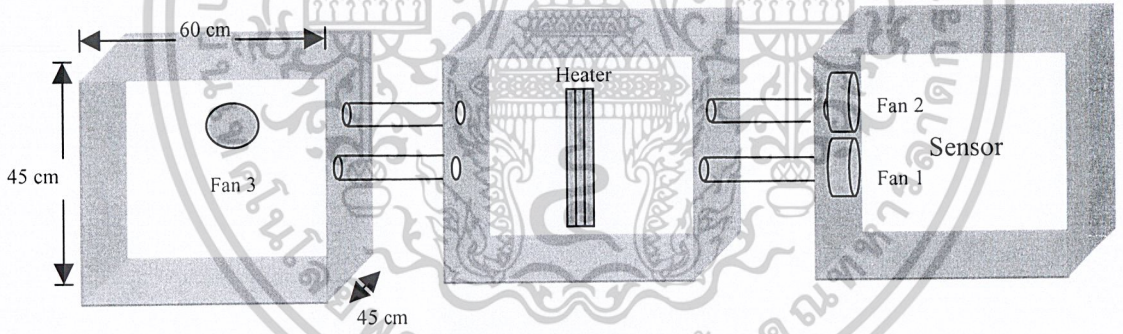
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ผิดพลาดมาก ฉะนั้นผู้ควบคุมการทำงานจึงจำเป็นที่จะต้องป้อนกิริยาตัวอื่นไปเช่น พีไอ พีดี และ พีไอดี เพื่อส่งไปควบคุมฮาร์ดแวร์ และมีการตรวจจับอุณหภูมิมาเปรียบเทียบค่าซ้ำอีก เมื่อยังไม่ได้ค่าเป้าหมายจะทำการควบคุมต่อไปอีกเรื่อยๆจนให้ได้ค่าเป้าหมายที่ต้องการโดยค่าทั้งหมดจะแสดงออกที่คอมพิวเตอร์

3.1.1 ออกแบบห้องจำลองควบคุมอุณหภูมิ

ห้องจำลองที่ต้องการสร้างขึ้นจากแผ่นกันความร้อน ซึ่งสามารถทนความร้อนได้ถึง 90 องศาเซลเซียส มีห้องจำลองทั้งหมด 3 ห้องด้วยกัน โดยมีท่อในการส่งถ่ายของอุณหภูมิในระหว่างห้องแต่ละห้อง ให้เป็นท่อนำพาอุณหภูมิไปยังห้องที่ต้องการวัดอุณหภูมิ 1 ท่อ และท่อระบายความร้อนอีก 1 ท่อ ดังรูปที่ 3.2 ห้องจำลองมีขนาดกว้าง 45 ซม. สูง 45 ซม. และยาว 60 ซม.

ห้องจำลองควบคุมอุณหภูมิมีลักษณะ 3 ลูป ให้ลูปที่ 1 (ซ้ายมือ) คือลูปการระบายลมออกด้วยพัดลมตัวที่ 3 ให้ลูปที่ 2 (ลูปกลาง) คือลูปการสร้างอุณหภูมิจากฮีตเตอร์ ให้ลูปที่ 3 (ขวามือ) คือลูปการควบคุมอุณหภูมิ ภายในลูปประกอบไปด้วย พัดลมตัวที่ 1 ตัวที่ 2 และไอซีตรวจจับอุณหภูมิทั้ง 3 ลูปสามารถควบคุมการไหลเวียนของอากาศด้วยท่อ พีวีซีขนาด 3 นิ้ว จำนวน 4 ท่อ

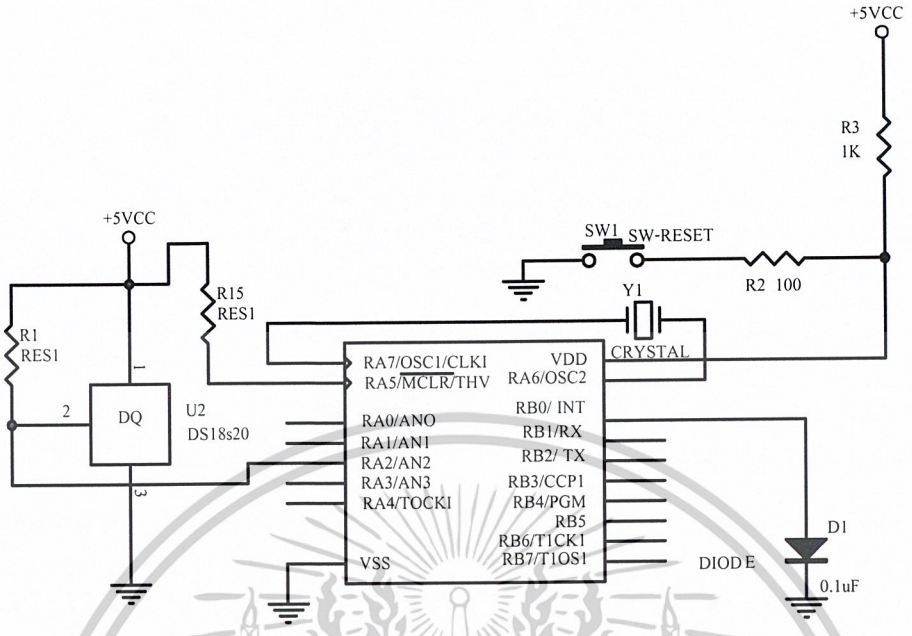


รูปที่ 3.2 โครงสร้างของห้องจำลอง

3.2 วงจรตรวจจับอุณหภูมิ

จากวงจรตรวจจับอุณหภูมิดังรูปที่ 3.3 ทำหน้าที่ในการส่งและรับค่าจากตัวตรวจจับอุณหภูมิ โดยใช้ DS 18S20 เป็นตัวตรวจจับอุณหภูมิแบบบัส 1 สาย DS 18S20 จะทำการส่งค่าไปยัง PIC 16F628 จากนั้นจะทำการส่งสัญญาณซึ่งเป็นสัญญาณแบบดิจิตอลไปยัง PIC 16F877 เพื่อทำการควบคุมอุณหภูมิให้ได้ตามค่าเป้าหมายที่ต้องการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



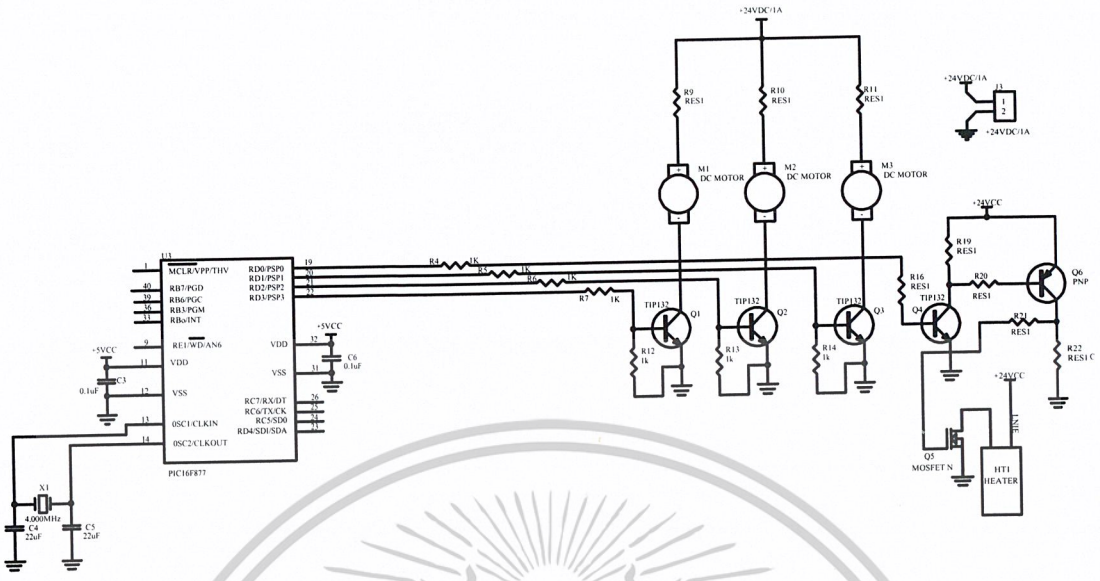
รูปที่ 3.3 วงจรตรวจจับอุณหภูมิ

จากการนำ DS 18S20 มาเป็นตัวส่งสัญญาณ เมื่อเราทำการป้อน Vcc = 5 โวลต์ ให้ DS18S20 แล้วทำการวัดที่ขา 2 จะได้แรงดันออกมาเป็นมิลลิโวลต์ แรงดันที่ออกมาจะแปรผันตามอุณหภูมิ ซึ่งแรงดันที่ขาออกของ DS 18S20 จะส่งไปให้กับไอซี PIC 16F628 ซึ่งจะใช้ DS 18S20 เป็นตัวตรวจวัดอุณหภูมิในห้องจำลองที่ 3

3.3 วงจรไดร์เวอร์

ชุดวงจรไดร์เวอร์ทำหน้าที่ในการต่อขับเคลื่อนมอเตอร์ โดยใช้ให้ทรานซิสเตอร์เป็นตัวขับวงจร แรงดันไฟชุดหนึ่งไหลผ่านทรานซิสเตอร์ทำให้ครบวงจร มอเตอร์ก็จะทำงานตามคำสั่งของโปรแกรม ชุดขับเคลื่อนมีด้วยกัน 3 ชุด ส่วนอีกชุดหนึ่งเป็นการต่อทรานซิสเตอร์ให้กับฮีตเตอร์ โดยใช้สัญญาณการควบคุมจาก PIC 16F877 จะสั่งงานให้ทรานซิสเตอร์เกิดการนำกระแสให้มีแหล่งจ่ายไฟอีกชุดหนึ่งไหลผ่านทรานซิสเตอร์ให้ฮีตเตอร์ทำงาน ดังรูปที่ 3.4 โดยทั้งหมดอาศัยการเขียนโปรแกรมจาก PIC 16F628

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



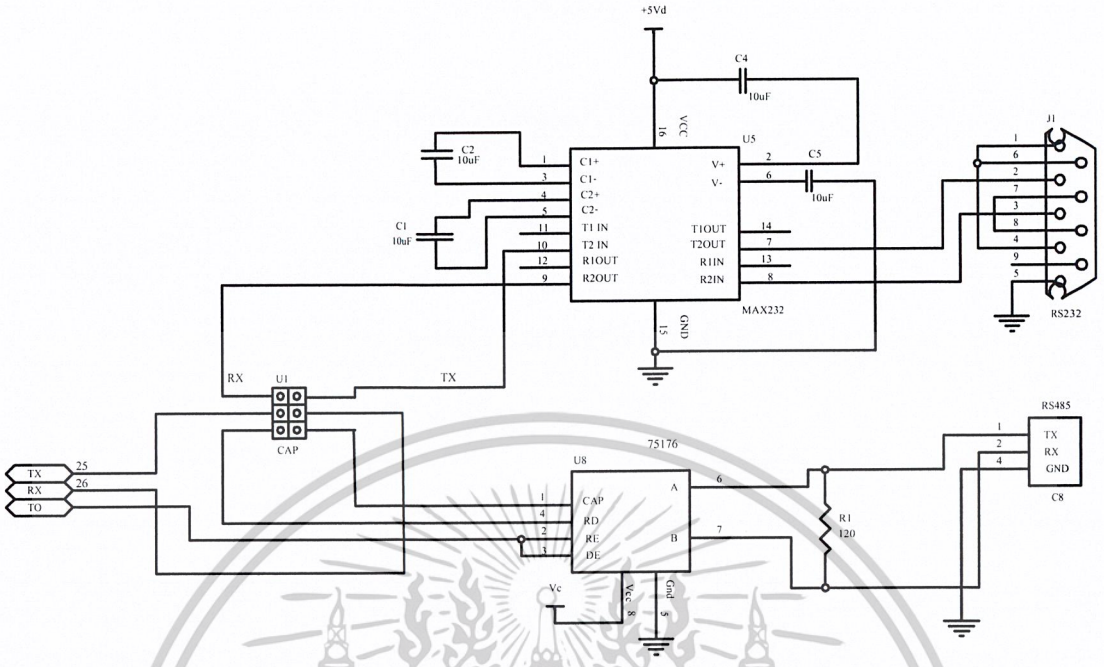
รูปที่3.4 วงจรไดร์เวอร์

3.4 วงจรส่งผ่านข้อมูลแบบอนุกรม

ใช้การส่งข้อมูลผ่านพอร์ตอนุกรม แบบ RS232 และ RS485 ซึ่งสามารถเลือกการใช้งานได้ตัวใดตัวหนึ่ง โดย RS 485 จะเป็นตัวเชื่อมต่อระหว่างชุดควบคุมกับพอร์ตอนุกรมของคอมพิวเตอร์ โดยจะมีการส่งและการรับข้อมูลและจะมีสวิตช์เลือกการทำงานส่งข้อมูลผ่านพอร์ตอนุกรมโดยการทำงานของพอร์ตอนุกรมจะเป็นวงจรสำเร็จ ดังรูปที่ 3.5

จากวงจรดังรูปที่ 3.5 วงจรส่งผ่านข้อมูลแบบอนุกรมสามารถเลือกใช้ ได้ 2 แบบคือ RS 232 และ RS 485 โดยจะเลือกใช้อย่างใดอย่างหนึ่งเท่านั้น การทดลองเราทำได้โดยการ ต่อขา RX และ TX จาก PIC 16F877 ไปในเข้าวงจรส่งผ่านข้อมูลแบบอนุกรม RS 232 และ RS 485 โดยป้อนโปรแกรมส่งผ่านข้อมูลแบบง่ายเพื่อดูผล ทำการเปิดดูในโปรแกรมไฮเปอร์เทอร์มินอล หากสามารถส่งข้อมูลได้แสดงว่าการต่อวงจรถูกต้องสามารถนำวงจรไปใช้ได้

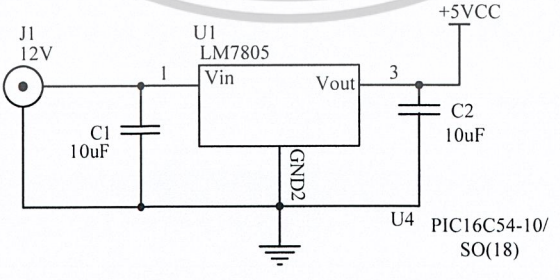
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.5 วงจรส่งผ่านข้อมูลแบบอนุกรม

3.5 วงจรแหล่งจ่ายแรงดัน

วงจรแหล่งจ่ายแรงดันดังรูปที่ 3.6 เป็นชุดแหล่งจ่ายไฟฟ้ากระแสตรง 24 โวลต์ ให้กับวงจรควบคุมต่างๆ ซึ่งวงจรภาคแหล่งจ่ายนี้จะมีแรงดันแบ่งเป็น 3 ชุด คือ Vcc1 จะให้แรงดันไฟฟ้ากระแสตรง 24 โวลต์ เป็นไฟเลี้ยงสำหรับมอเตอร์พัดลม 3 ตัว Vcc2 จะให้แรงดันไฟฟ้ากระแสตรง 12 โวลต์ เป็นไฟเลี้ยงให้รีเลย์ ซึ่งทำหน้าที่ในการตัดและต่อไฟให้ฮีตเตอร์ และ Vcc3 จะให้แรงดันไฟฟ้ากระแสตรง 5 โวลต์ เป็นไฟเลี้ยงให้กับไอซี PIC 16F628 และ DS18S20



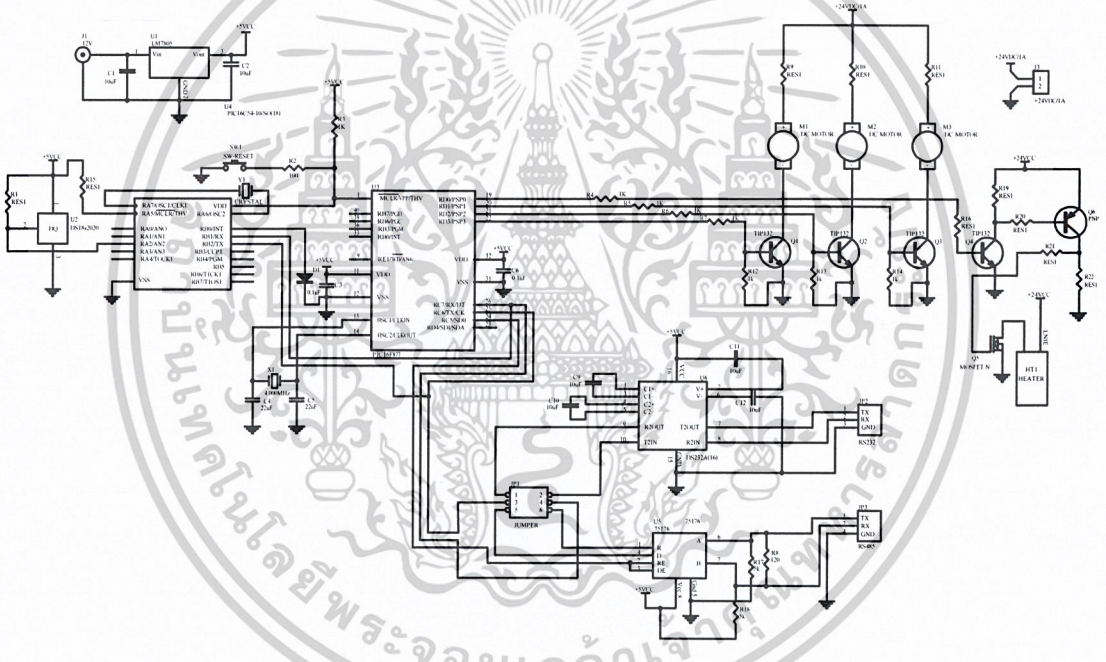
รูปที่ 3.6 วงจรแหล่งจ่ายแรงดัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 3.6 วงจรแหล่งจ่ายแรงดัน จะมีเอาต์พุต 3 เอาต์พุต คือ Vcc1 , Vcc2 และ Vcc3 เมื่อทำการวัดโดยใช้มิเตอร์ วัดแต่ละจุดจะได้ค่าแรงดันดังนี้ Vcc1 จะมีค่าแรงดันเท่ากับ 24 โวลต์ Vcc2 จะมีค่าแรงดันเท่ากับ 12 โวลต์ และที่ Vcc3 จะมีค่าแรงดันเท่ากับ 5 โวลต์

3.6 วงจรการควบคุมการทำงานของชุดสาธิตการควบคุมอุณหภูมิภายในห้องจำลองแบบวงปิด

เป็นการเอาวงจรตรวจวัดอุณหภูมิ วงจรไดร์เวอร์ วงจรส่งผ่านข้อมูลแบบอนุกรม และวงจรจ่ายแรงดัน ต่อรวมกัน ดังแสดงดังรูปที่ 3.7



รูปที่ 3.7 วงจรควบคุมการทำงานของชุดสาธิตการควบคุมอุณหภูมิภายในห้องจำลองแบบวงปิด

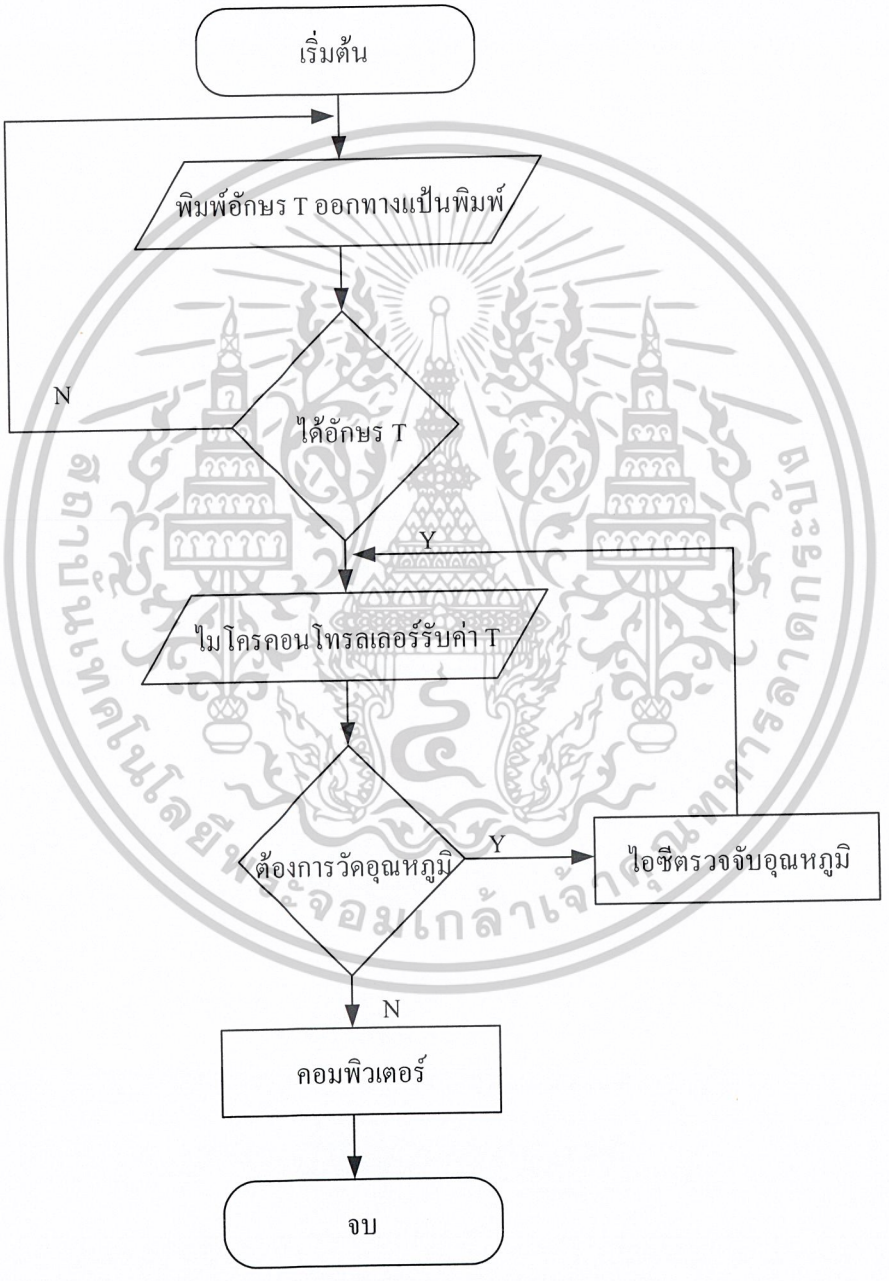
3.7 การออกแบบโปรแกรมควบคุม

การออกแบบโปรแกรมควบคุมสามารถควบคุมการทำงานได้ โดยมีการออกแบบให้ชุดของไมโครคอนโทรลเลอร์สามารถควบคุมการติดต่อของการตรวจจับอุณหภูมิด้วยไอซีตรวจจับอุณหภูมิ อีกทั้งในส่วนของไมโครคอนโทรลเลอร์ยังสามารถควบคุมเปอร์เซ็นต์การมอดูเลตความกว้างของพัลซ์ให้กับการทำงานของมอเตอร์และฮีตเตอร์ และยังมีการอินเตอร์เฟสกับคอมพิวเตอร์ซึ่งเป็นส่วนแสดงผลผ่านทางพอร์ตอนุกรม ซึ่งสามารถออกแบบโปรแกรมต่างๆได้ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.7.1 โปรแกรมตรวจจับอุณหภูมิ

การสร้างโปรแกรมตรวจจับอุณหภูมิ โดยมีไมโครคอนโทรลเลอร์เบอร์ 16F628 ทำหน้าที่ควบคุมไอซีตรวจจับอุณหภูมิเบอร์ DS 18S20 โดยให้มีการส่งและรับข้อมูลผ่านพอร์ตอนุกรมแบบ RS 232 ให้แสดงผลทางคอมพิวเตอร์ผ่านโปรแกรมไฮเปอร์เทอร์มินอล สามารถดูการทำงานได้จากรูปที่ 3.8



รูปที่ 3.8 ผังงานโปรแกรมตรวจจับอุณหภูมิ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.7.2 โปรแกรมควบคุมมอเตอร์และฮีตเตอร์ด้วยสัญญาณ PWM

การสร้างโปรแกรมควบคุมมอเตอร์และฮีตเตอร์ด้วยสัญญาณ PWM อาศัยหลักการของการมอดูเลตความกว้างของสัญญาณพัลซ์ไปควบคุมมอเตอร์และฮีตเตอร์ โดยตัวควบคุมคือไมโครคอนโทรลเลอร์เบอร์ PIC 16F877 ทำหน้าที่คอยสั่งการควบคุมและติดต่อกับคอมพิวเตอร์ผ่านทาง RS 232 แสดงผลที่โปรแกรมไฮเปอร์เทอร์มินอล โดยสามารถเขียนโปรแกรมด้วยภาษาซี สามารถอธิบายหลักการทำงานได้จากรูปที่ 3.9

3.7.3 โปรแกรมหน้าต่างแสดงผลหลัก

การสร้างโปรแกรมหน้าต่างแสดงผลหลักอาศัยหลักการแสดงผลแบบกราฟฟิคด้วยโปรแกรมวิชวล เบสิก ให้ทำการติดต่อคอมพอร์ตผ่านพอร์ตอนุกรม จากนั้นทำการตั้งค่าเป้าหมายตั้งค่ากิริยาควบคุมแล้วให้แสดงผลกราฟที่จอคอมพิวเตอร์ อธิบายหลักการทำงานได้จากรูปที่ 3.10

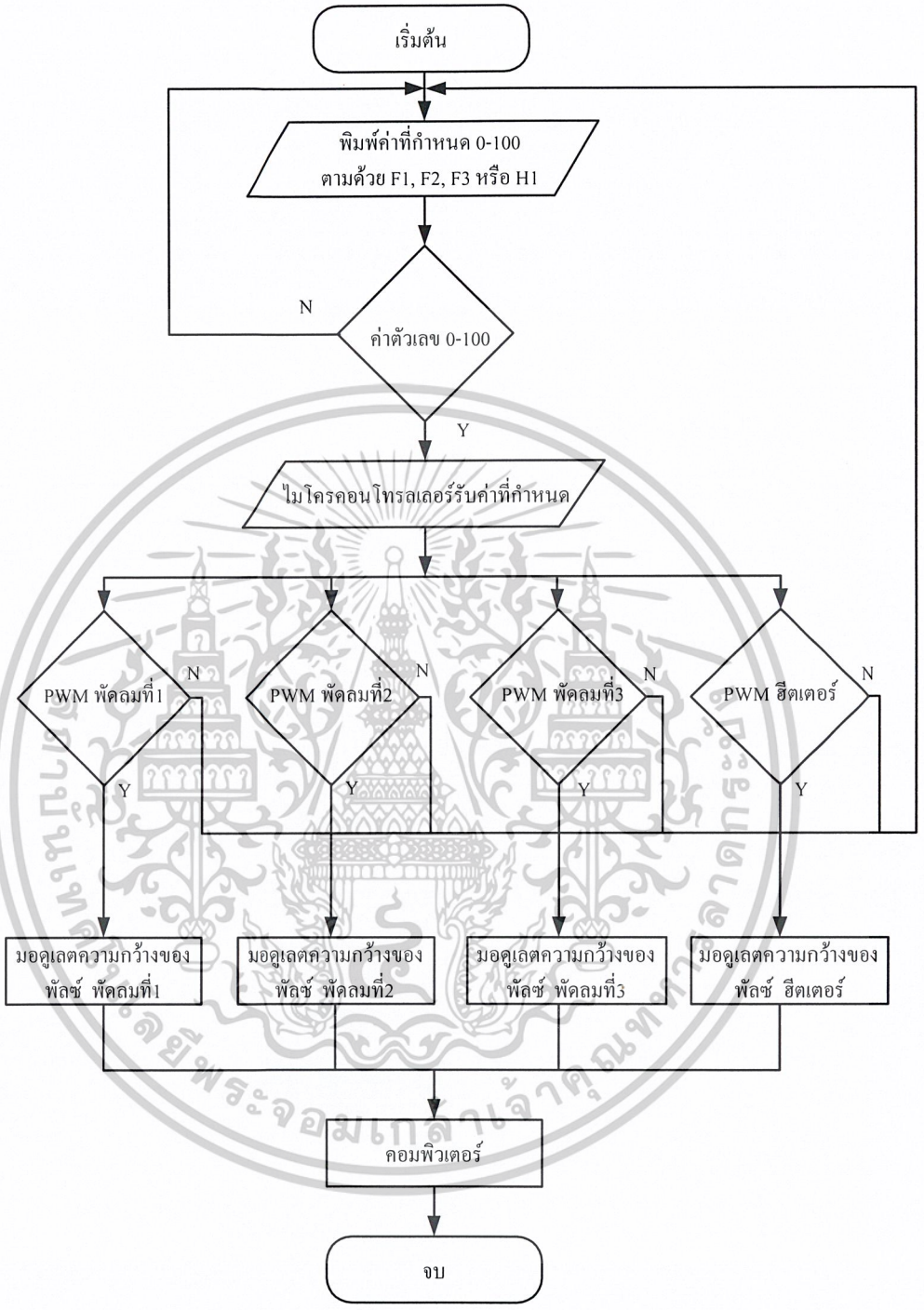
3.7.4 โปรแกรมย่อยของกิริยาการควบคุม

การสร้างโปรแกรมย่อยของกิริยาการควบคุม สามารถเขียนด้วยโปรแกรมการใช้งานของวิชวล เบสิก กำหนดให้มีการตั้งค่าเป้าหมายช่วงอุณหภูมิ 40-70 องศาเซลเซียส จากนั้นกำหนดค่าเกณฑ์การควบคุมและสามารถเลือกกลุ่มกิริยาการควบคุมแบบต่างๆ แสดงกราฟผลการทดลองทางจอคอมพิวเตอร์ สามารถอธิบายการทำงานได้ดังรูปที่ 3.11

3.7.5 โปรแกรมย่อยของปุ่มกดควบคุม

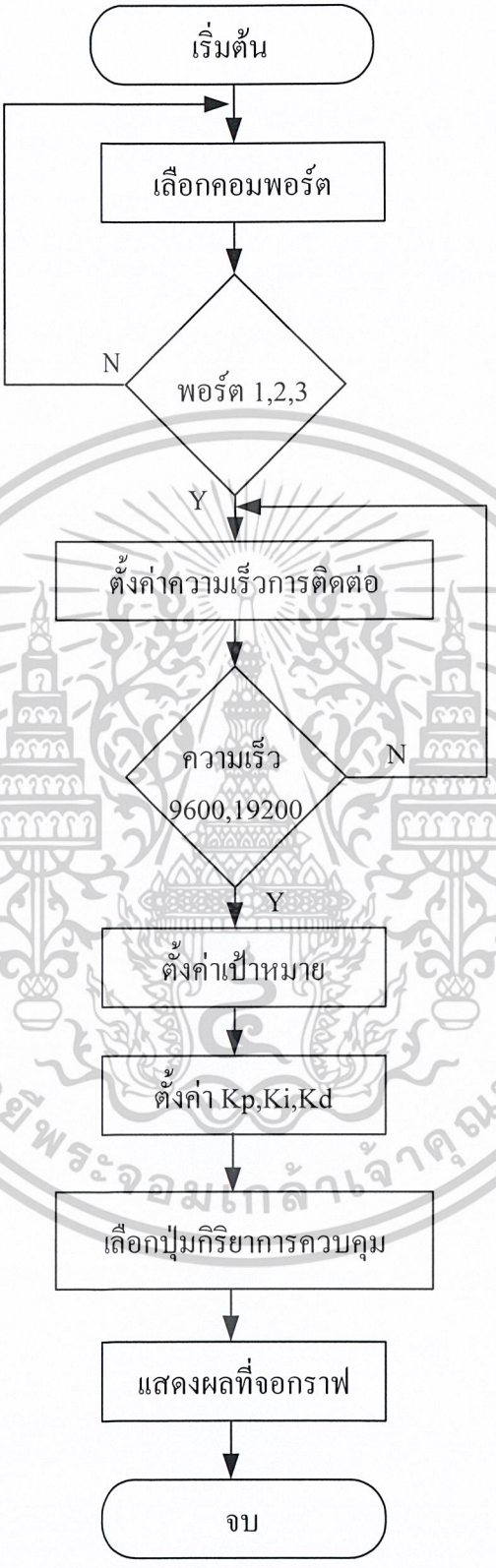
การสร้างโปรแกรมย่อยของปุ่มกดควบคุม เขียนโปรแกรมด้วยโปรแกรมวิชวล เบสิก ให้มีการตั้งค่าเป้าหมายในช่วงอุณหภูมิ 40-70 องศาเซลเซียส จากนั้นกำหนดตั้งค่าเกณฑ์การควบคุมให้เหมาะสม สามารถเลือกปุ่มกดกิริยาการควบคุมแล้วให้แสดงค่ากราฟบนจอคอมพิวเตอร์ สามารถอธิบายการทำงานได้ดังรูป 3.12

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



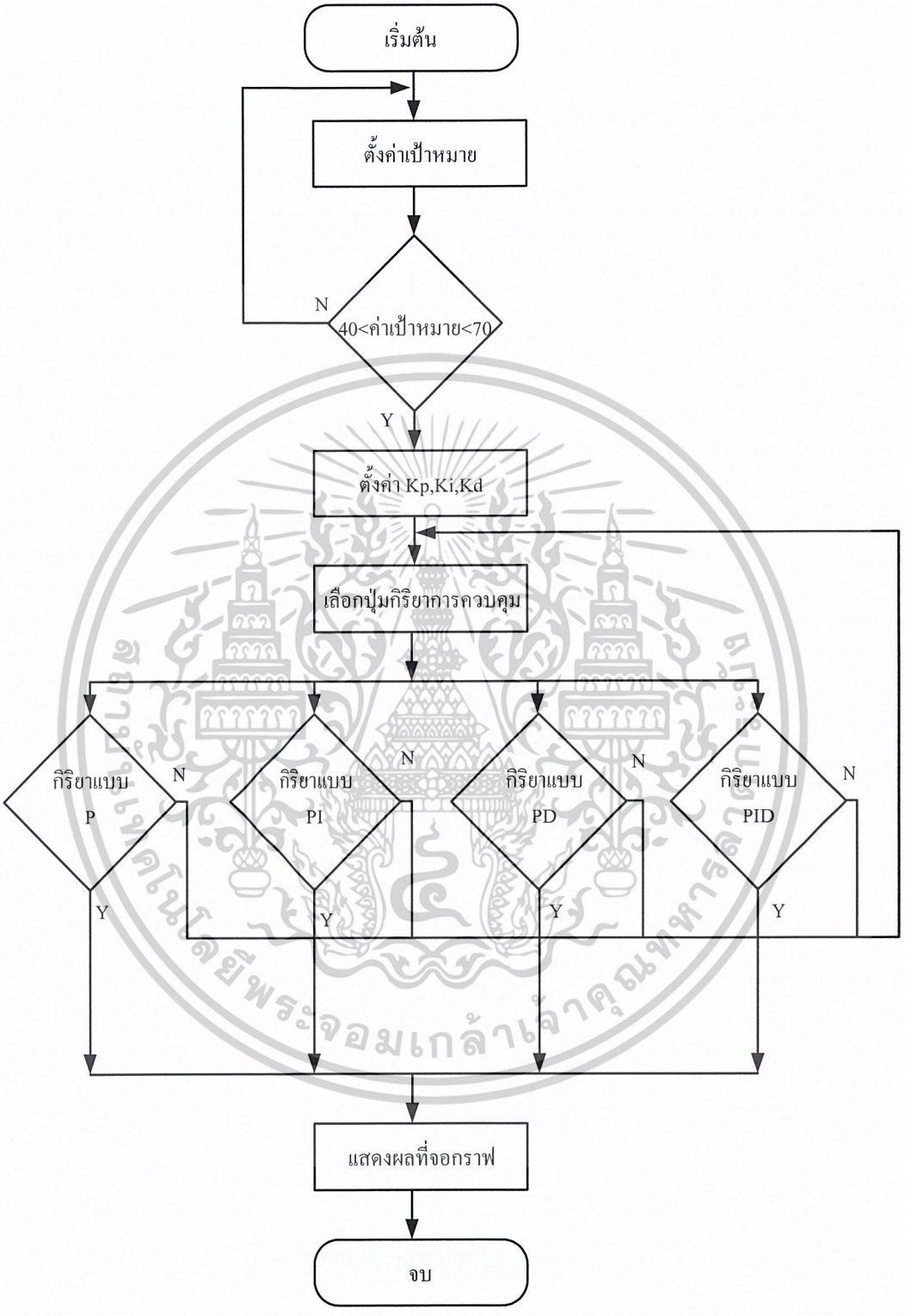
รูปที่ 3.9 ผังงานโปรแกรมควบคุมมอเตอร์และฮีตเตอร์ด้วยสัญญาณ PWM

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.10 ผังงานโปรแกรมหน้าต่างแสดงผลหลัก

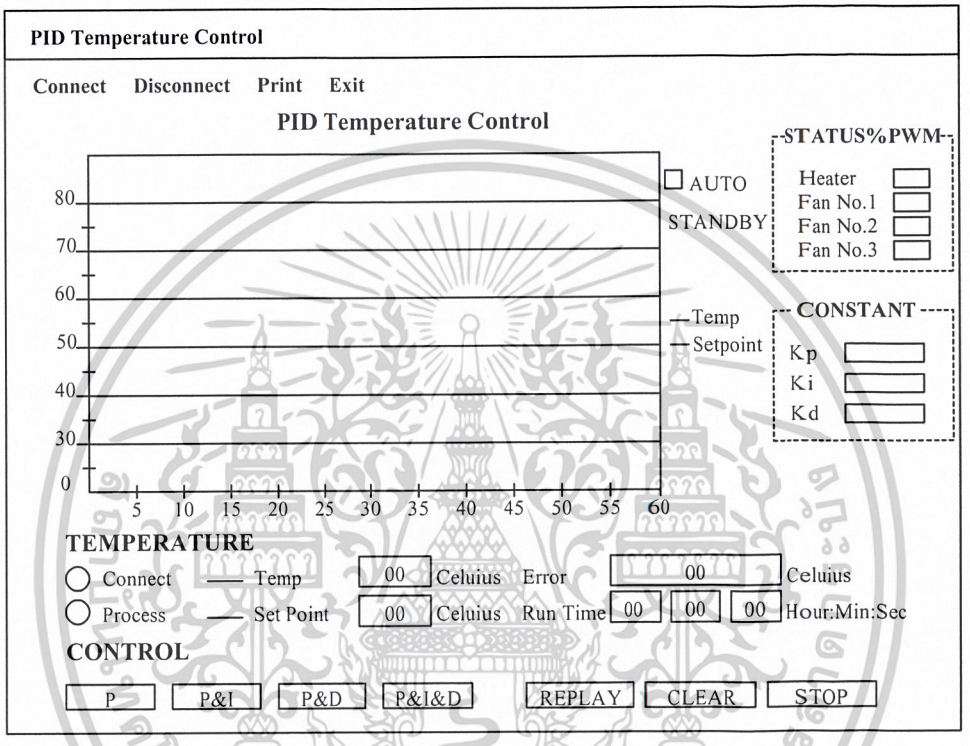
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.11 ผังงานโปรแกรมย่อยกิริยาการควบคุม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

P, PI, PD และ PID ส่วนของการติดผ่านพอร์ตอนุกรม ส่วนของการตั้งค่าเป้าหมาย ส่วนแสดงสถานะการทำงานของฮีตเตอร์และพัดลม ส่วนของการแสดงผลค่าผิดพลาดและระยะเวลาในการทำงาน และส่วนของการควบคุมอื่นๆ เช่น หยุดการทำงาน เคลียร์การทำงาน และย้อนดูผลการทดลอง



รูปที่ 3.13 หน้าต่างควบคุมและแสดงผล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

การทดลองและผลการทดลอง

ในการทดลองชุดสาธิตการควบคุมอุณหภูมิในห้องจำลองแบบวงปิด ใช้โปรแกรมวิชวลเบสิกในการพัฒนาการแสดงผลทางจอคอมพิวเตอร์ โดยแบ่งการทดลองได้แก่ การทดลองผลตอบสนองของกิริยาการควบคุมแบบพีที่มีต่ออินพุตของอุณหภูมิที่ค่า Kp ผลตอบสนองของกิริยาการควบคุมแบบไอที่มีต่ออินพุตของอุณหภูมิที่ค่า Ki ผลตอบสนองของกิริยาการควบคุมแบบดีที่มีต่ออินพุตของอุณหภูมิที่ค่า Kd

4.1 การทดลองห้องจำลองควบคุมอุณหภูมิ

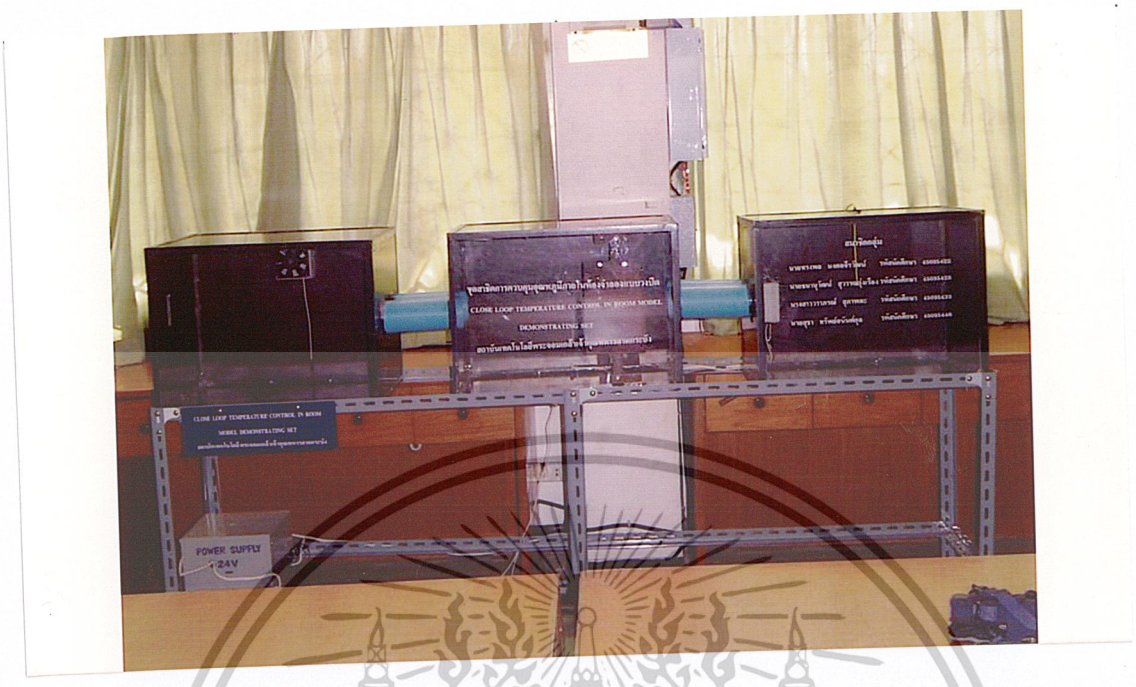
4.1.1 ลำดับขั้นการทดลอง

การทดลองห้องจำลองควบคุมอุณหภูมิมีลักษณะ 3 ลูป สามารถควบคุมการไหลเวียนของอากาศด้วยท่อ พีวีซีขนาด 3 นิ้ว จำนวน 4 ท่อ จากการทดลอง

1) พัดลมตัวที่ 1 ทำหน้าที่ดูดลมร้อนเข้าไปในลูปที่ 3 พัดลมตัวที่ 2 ทำหน้าที่เป่าลมร้อนออกจากลูปที่ 3 พัดลมตัวที่ 3 ทำหน้าที่ระบายลมร้อนออกจากชุดสาธิตการควบคุมอุณหภูมิภายในวงปิด ส่วนฮีตเตอร์ทำหน้าที่สร้างความร้อนให้กับห้องจำลอง

2) ทำการทดสอบกับลูปด้วยควันจากธูป ผลปรากฏได้ว่า พัดลมตัวที่ 1 ทำหน้าที่ดูดควันธูปได้จริง พัดลมตัวที่ 2 สามารถเป่าควันธูปออกได้จริง แต่พัดลมตัวที่ 1 และ 2 จะต้องทำงาน 100 เปอร์เซ็นต์ ถึงจะทำให้ควันธูปเกิดการไหลเวียนได้สะดวก ดังรูปที่ 4.1 ขณะที่พัดลมตัวที่ 1 และ 2 ทำงาน พัดลมตัวที่ 3 จะหยุดทำงานและขณะที่พัดลมตัวที่ 3 ทำงาน พัดลมตัวที่ 1 และ 2 จะหยุดทำงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

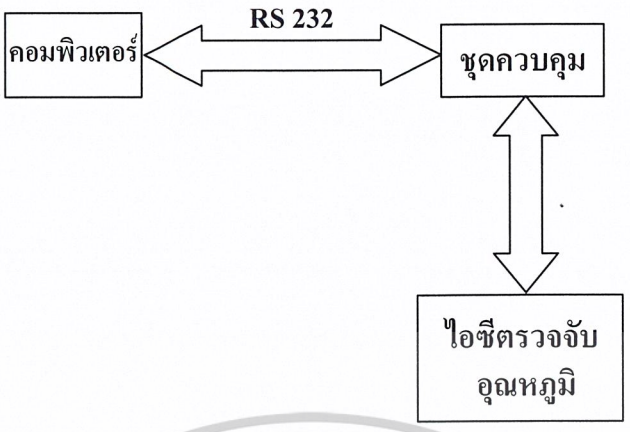


รูปที่ 4.1 การทดลองห้องจำลอง

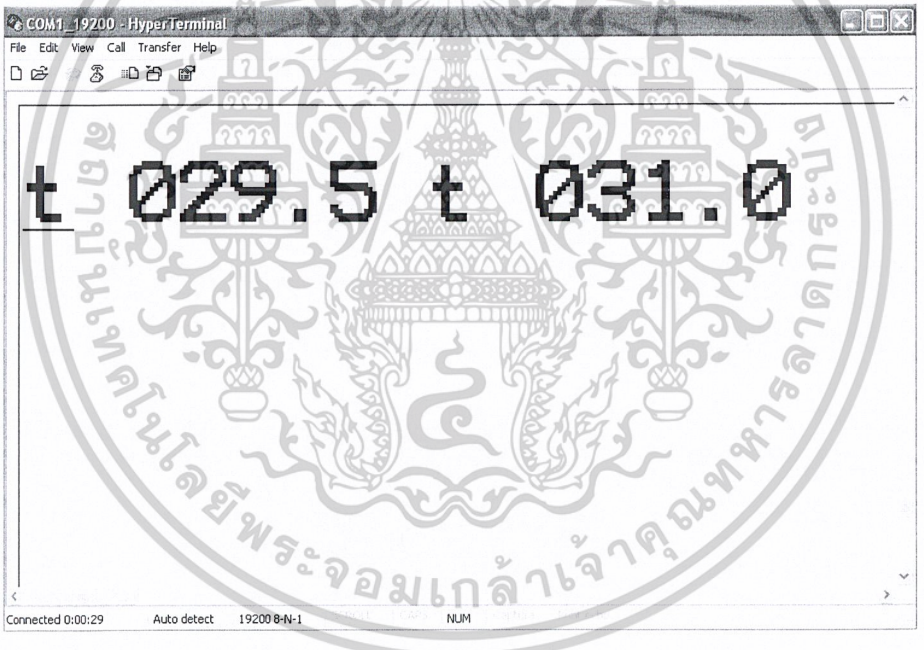
4.2 การทดลองวงจรตรวจจับอุณหภูมิ

การทดลองวงจรตรวจจับอุณหภูมิ โดยการเขียนโปรแกรมให้กับไมโครคอนโทรลเลอร์เบอร์ PIC 16F628 เป็นแบบ 18 ขา เพื่อทำหน้าที่ควบคุมการตรวจจับอุณหภูมิแบบไอซีตรวจจับอุณหภูมิเบอร์ DS 18S20 ซึ่ง DS 18S20 เป็นแบบบัส 1 สาย สามารถตรวจจับอุณหภูมิแล้วส่งข้อมูลให้กับไมโครคอนโทรลเลอร์เป็นสัญญาณดิจิตอล ไมโครคอนโทรลเลอร์ประมวลผลข้อมูลพร้อมที่จะรับการคีย์ข้อมูลที่เป็นพิมพ์ ด้วยอักษร T แล้วกดเว้นวรรคตัวอักษร เพื่อเรียกค่าอุณหภูมิมาแสดงที่หน้าจอโปรแกรมไฮเปอร์เทอร์มินอล ดังรูปที่ 4.3 แล้วไฟที่วงจรตรวจจับอุณหภูมิจะกระพริบตลอดเวลา เมื่อมีการอ่านอุณหภูมิ ดังรูปที่ 4.2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.2 ส่ง-รับ อุณหภูมิไปยังไฮเปอร์เทอร์มินอล



รูปที่ 4.3 การทดลองวงจรตรวจจับอุณหภูมิ

4.3 การทดลองวงจรไดร์เวอร์

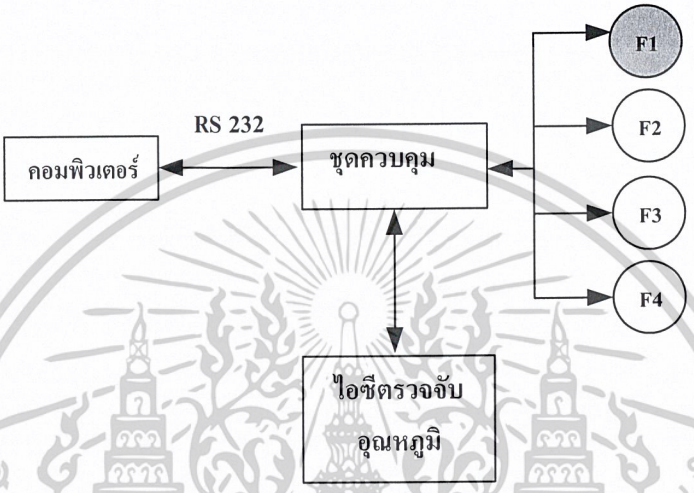
4.3.1 ลำดับขั้นการทดลอง

1) ป้อนค่า ตัวเลขแต่ 0-100 และตามหลังด้วย f1, f2, f3 หรือ h1

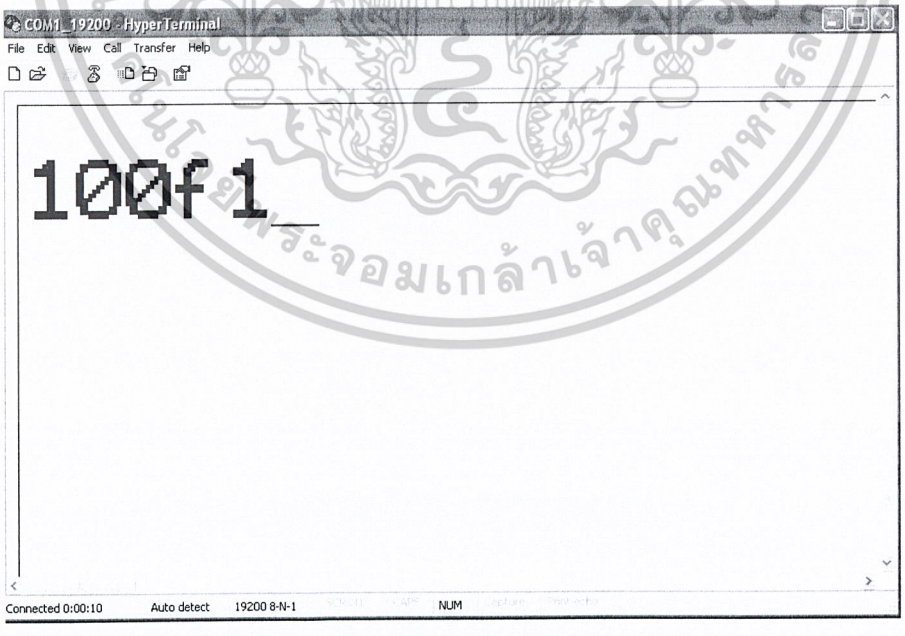
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2) ทำการป้อนข้อมูล 100f1 และจากนั้นตามด้วยการกดเว้นวรรคตัวอักษรที่แป้นพิมพ์เพื่อทำการส่งข้อมูลไปยังไมโครคอนโทรลเลอร์เพื่อให้พัดลมตัวที่ 1 ทำงาน 100 เปอร์เซ็นต์ มีผลดังรูปที่ 4.5

3) สังเกตุผลจาก LED ที่ไมโครคอนโทรลเลอร์สั่งงานไฟหลอดที่ 1 ติดดังรูปที่ 4.4



รูปที่ 4.4 การทดลองวงจรไดร์เวอร์หลอดไฟหลอดที่ 1 ทำงาน

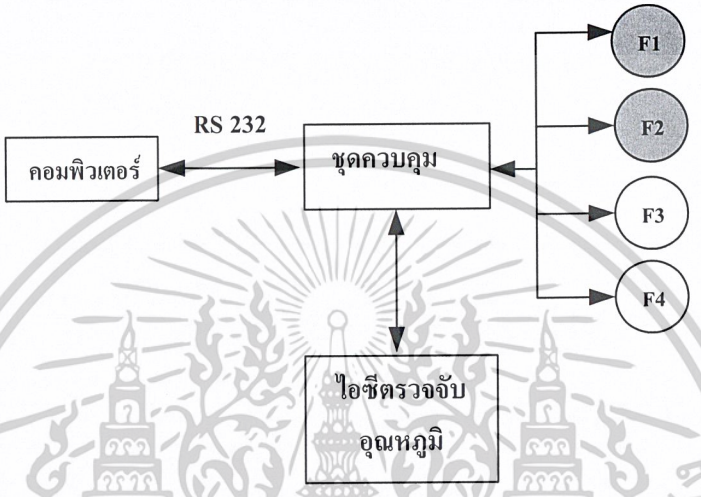


รูปที่ 4.5 ผลการป้อนข้อมูล 100f1

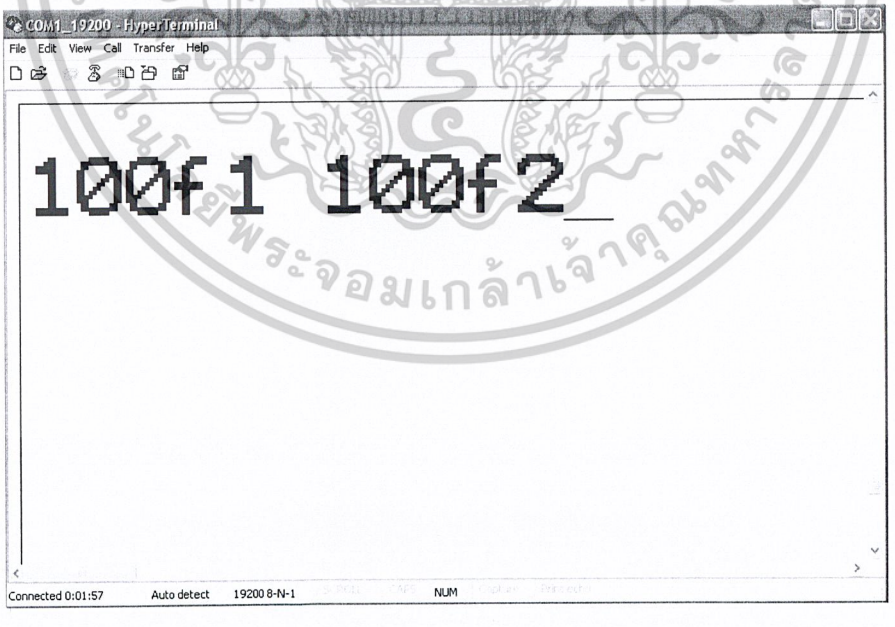
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4) ทำการป้อนข้อมูล 100f1 และ 100f2 จากนั้นตามด้วยการกดเว้นวรรคตัวอักษรที่แป้นพิมพ์เพื่อทำการส่งข้อมูลไปยังไมโครคอนโทรลเลอร์เพื่อให้พัดลมตัวที่ 1 และ 2 ทำงาน 100 เปอร์เซ็นต์ มีผลดังรูปที่ 4.7

5) สังเกตุผลจาก LED ที่ไมโครคอนโทรลเลอร์สั่งงานไฟหลอดที่ 1 และ 2 ติดดังรูปที่ 4.6



รูปที่ 4.6 การทดลองวงจรไดร์เวอร์หลอดไฟหลอดที่ 1 และ 2 ทำงาน

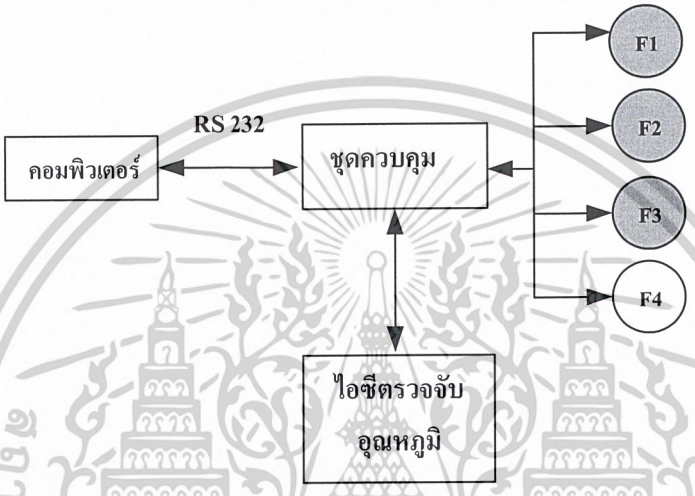


รูปที่ 4.7 ผลการป้อนข้อมูล 100f1 และ 100f2

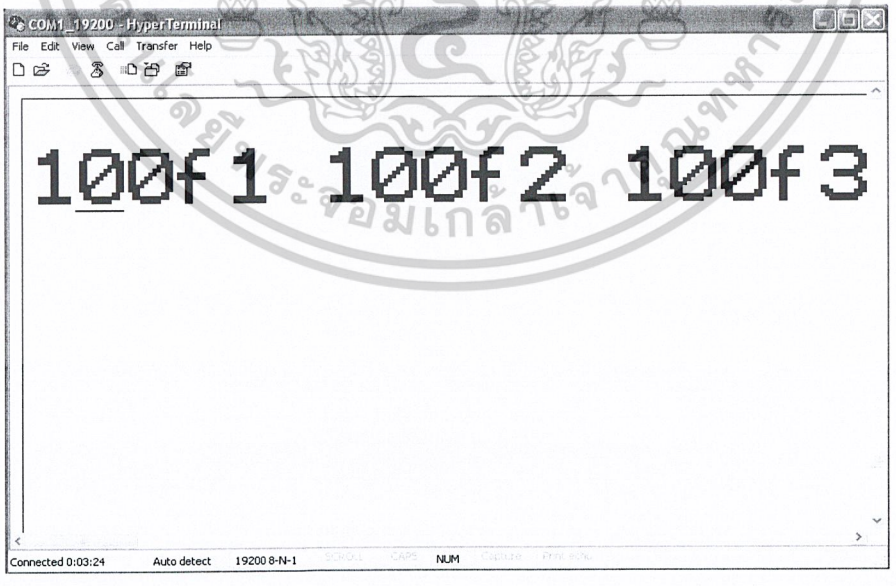
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6) ทำการป้อนข้อมูล 100f1 100f2 และ 100f3 จากนั้นตามด้วยการกดเว้นวรรคตัวอักษรที่แป้นพิมพ์เพื่อทำการส่งข้อมูลไปยังไมโครคอนโทรลเลอร์เพื่อให้พัดลมตัวที่ 1 และ 2 ทำงาน 100 เปอร์เซ็นต์ มีผลดังรูปที่ 4.9

7) สังเกตุผลจาก LED ที่ไมโครคอนโทรลเลอร์สั่งงานไฟหลอดที่ 1, 2 และ 3 ติดดังรูปที่ 4.8



รูปที่ 4.8 การทดลองวงจรไดร์เวอร์หลอดไฟหลอดที่ 1, 2 และ 3 ทำงาน

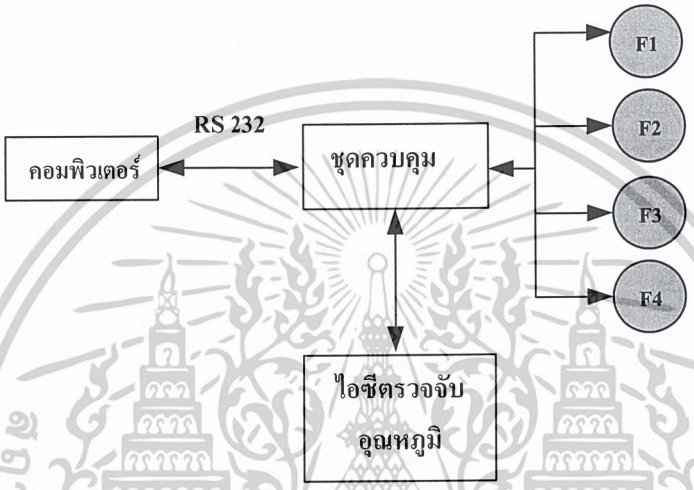


รูปที่ 4.9 ผลการป้อนข้อมูล 100f1 , 100f2 และ 100f3

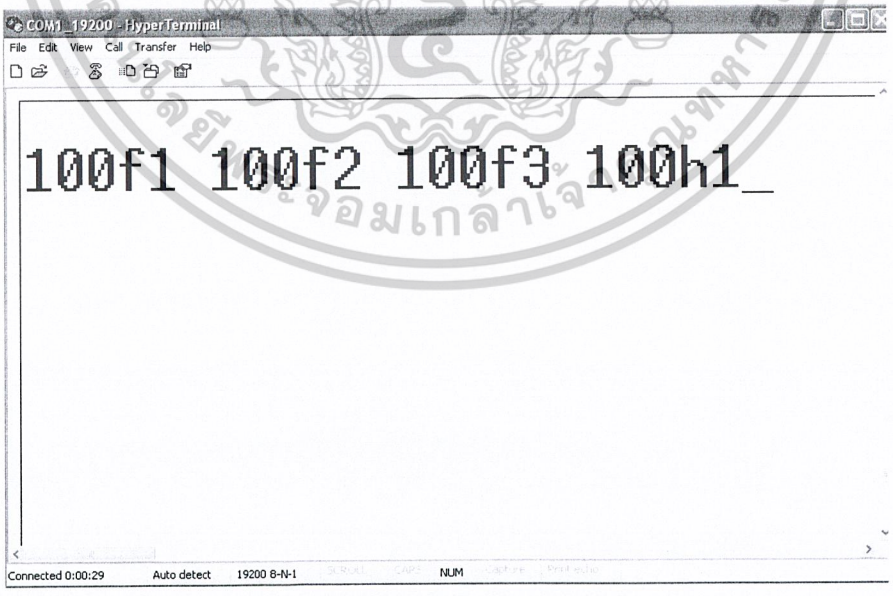
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

8) ทำการป้อนข้อมูล 100f1 100f2 100f3 และ 100h1 จากนั้นตามด้วยการกดเว้นวรรคตัวอักษรที่แป้นพิมพ์เพื่อทำการส่งข้อมูลไปยังไมโครคอนโทรลเลอร์เพื่อให้พัดลมตัวที่ 1, 2, 3 และฮีตเตอร์ตัวที่ 1 ทำงาน 100 เปอร์เซ็นต์ มีผลดังรูปที่ 4.11

9) สังเกตุผลจาก LED ที่ไมโครคอนโทรลเลอร์สั่งงานไฟหลอดที่ 1, 2, 3 และ 4 ติดดังรูปที่ 4.10



รูปที่ 4.10 การทดลองวงจรไดร์เวอร์หลอดไฟหลอดที่ 1, 2, 3 และฮีตเตอร์ทำงาน

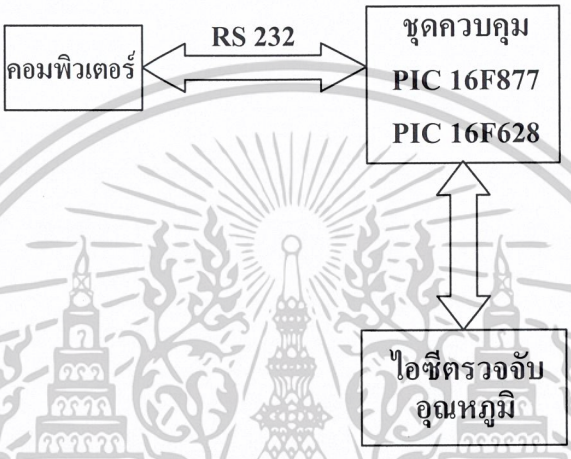


รูปที่ 4.11 ผลการป้อนข้อมูล 100f1 , 100f2, 100f3 และ 100h1

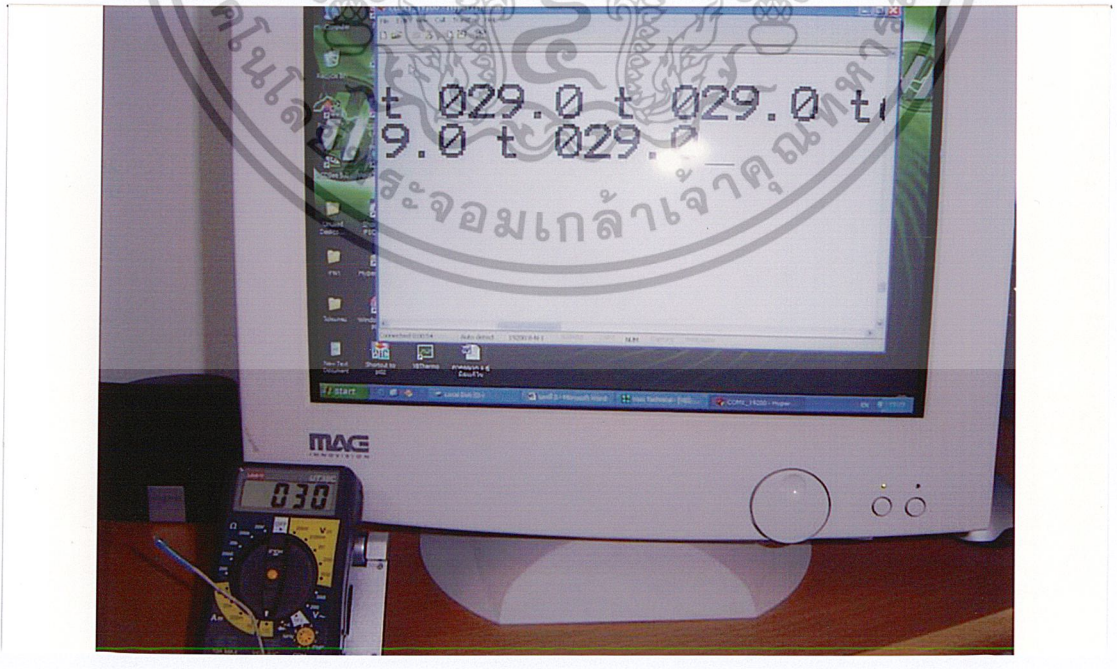
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.4 การทดลองวงจรส่งผ่านข้อมูลแบบอนุกรม

มีการใช้ไมโครคอนโทรลเลอร์ 2 ตัว คือ PIC 16F877 และ PIC 16F628 สามารถส่งผ่านข้อมูลแบบอนุกรม RS 232 โดยต่อขา RX ของไมโครคอนโทรลเลอร์ทั้ง 2 ตัวเข้าไว้ด้วยกันดังรูปที่ 4.12 ดังนั้นจึงสามารถควบคุมไมโครคอนโทรลเลอร์ทั้ง 2 ตัวพร้อมกันได้ สังเกตุการส่งผ่านข้อมูลด้วยการตรวจจับอุณหภูมิ ดังรูปที่ 4.13



รูปที่ 4.12 วงจรส่งผ่านข้อมูลแบบอนุกรม



รูปที่ 4.13 ผลการทดลองวงจรวงจรส่งผ่านข้อมูลแบบอนุกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

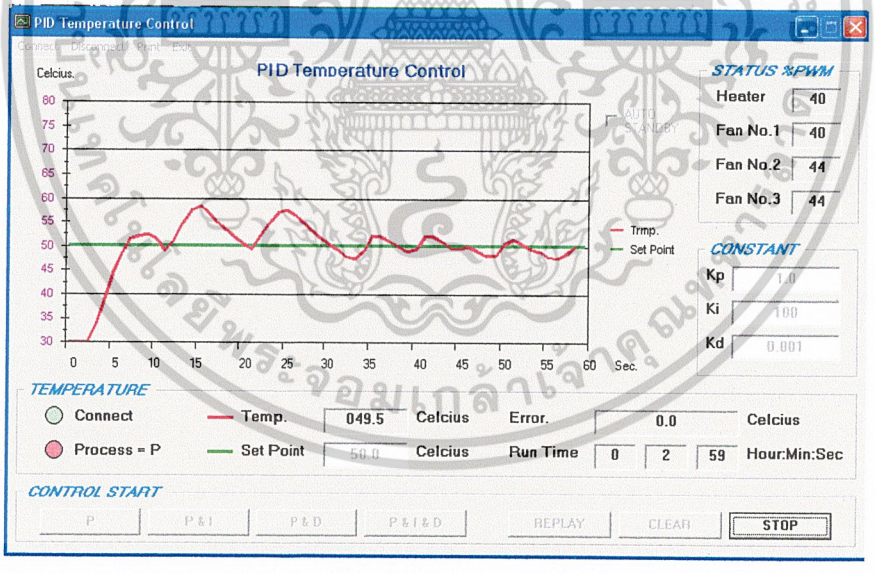
4.5 การทดลองวงจรแหล่งจ่ายแรงดัน

การทดลองกับวงจรเร็คกูเลต โดยใช้ไอซีเร็คกูเลตเบอร์ L7805CV สามารถวัดค่าแรงดันได้ 4.96 โวลต์ ซึ่งใช้เป็นแหล่งจ่ายให้ชุดควบคุม

4.6 วงจรการควบคุมการทำงานของชุดสาธิตการควบคุมอุณหภูมิภายในห้องจำลองแบบวงปิด

4.6.1 การทดลองชุดควบคุมโดยใช้ตัวสร้างอุณหภูมิภายนอก

ทดสอบการทำงานของลูปชุดสาธิตการควบคุมอุณหภูมิภายในห้องจำลองแบบวงปิด โดยการทดสอบการทำงานชุดวงจรควบคุมโดยการใช้เครื่องสร้างอุณหภูมิภายนอกมาทำการปรับระดับให้สามารถมีการพล็อตกราฟที่หน้าต่างของโปรแกรมวิชวล เบสิก มีการแสดงผลการเพิ่มและลดของอุณหภูมิ และการเปลี่ยนสัญญาณความกว้างพัลซ์ให้กับมอเตอร์และฮีตเตอร์ เมื่อมีการเปลี่ยนแปลงของระดับอุณหภูมิที่แตกต่างจากค่าเป้าหมาย สามารถแสดงผลการทดลองได้ดังรูปที่ 4.14



รูปที่ 4.14 การทดลองจากไดเป่าผม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.6.2 การทดลองอุณหภูมิโดยกิริยาควบคุมแบบพี

1) ลำดับขั้นการทดลอง

1.1) เริ่มต้นการทำงานคลิกที่ไอคอน VB Thermoเพื่อเปิดหน้าต่าง PID Themperature Control

1.2) เปิดสวิตช์สตาร์ทชุดควบคุม จากนั้นให้ไปที่คอมพิวเตอร์คลิก Connect ที่หน้าต่าง PID Temperature Control

1.3) ทำการเลือก Com Port โดยการป้อนตัวเลข Com Port เพื่อส่งผ่านข้อมูลแบบอนุกรมให้กับพอร์ตที่เราต้องการติดต่อ ในที่นี้เราเลือกพอร์ตที่ต้องการติดต่อที่ COM 1 หลังจากนั้นคลิก OK

1.4) ตั้งค่า Baud Rate ในที่นี้ตั้งค่าไว้ที่ 19200 หลังจากนั้นคลิก OK

1.5) คลิกปุ่ม AUTO STANDBY ให้ STATUS % PWM ทำงานและแสดงผล

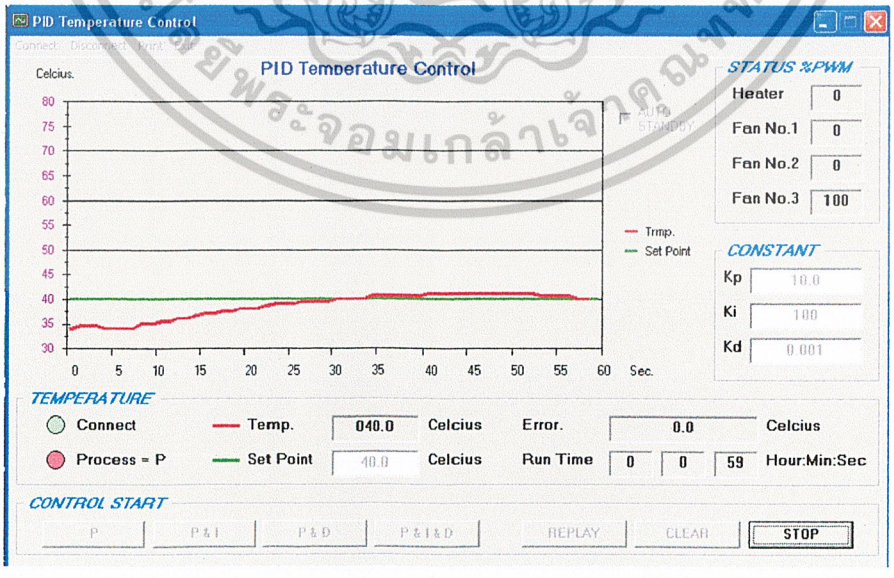
1.6) หน้าต่าง PID Temperature Control พร้อมที่จะรับข้อมูลในการควบคุม

1.7) ตั้งค่ากิริยาในการควบคุมที่เราต้องการ คือ Kp = 10, Ki = 100 หรือ Kd = 0.001

1.8) เลือกปุ่มกิริยาการควบคุมที่ต้องการคือ P

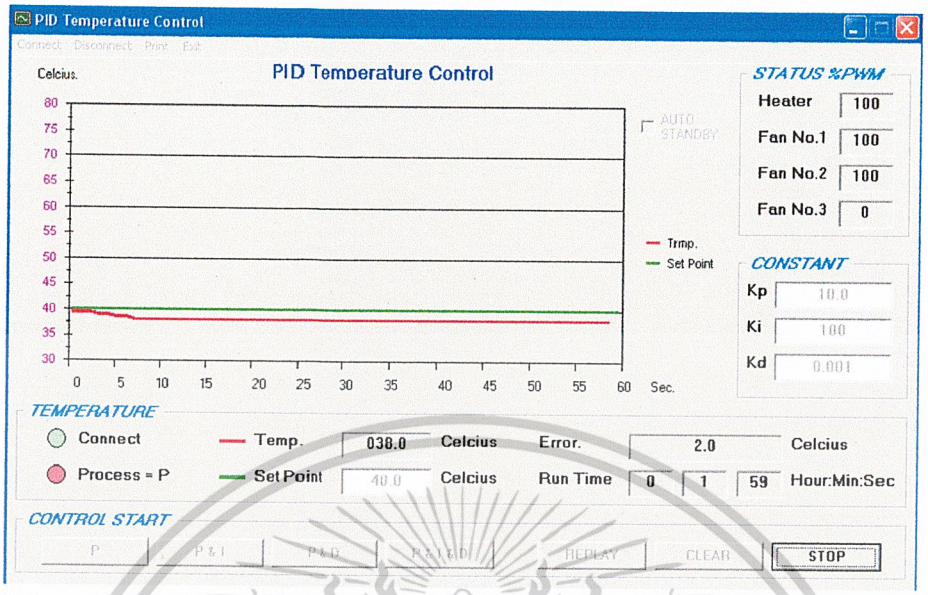
1.9) รอดูผลการทำงานได้จากกราฟที่หน้าต่าง PID Themperature Control ดังแสดงในรูปที่ 4.15 เป็นการควบคุมแบบพีให้เข้าสู่ค่าเป้าหมายที่ตั้งไว้ คือ 40 องศาเซลเซียส

1.10) การควบคุมแบบพีกำลังรักษาระดับค่าเป้าหมายให้คงที่ ดังแสดงดังรูปที่ 4.16



รูปที่ 4.15 การควบคุมแบบพี ให้เข้าสู่ค่าเป้าหมาย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.16 การควบคุมแบบพี ขณะกำลังรักษาระดับค่าเป้าหมายให้คงที่

4.6.3 การทดลองอุณหภูมิโดยกิริยาควบคุมแบบพีไอ

1) ลำดับขั้นการทดลอง

1.1) เริ่มต้นการทำงานคลิกที่ไอคอน VB Thermoเพื่อเปิดหน้าต่าง PID Themperature Control

1.2) เปิดสวิตช์สตาร์ทชุดควบคุมจากนั้นไปที่คอมพิวเตอร์ คลิก Connect ที่หน้าต่าง PID Themperature Control

1.3) ทำการเลือก Com Port โดยการป้อนตัวเลข Com Port เพื่อส่งผ่านข้อมูลแบบอนุกรมให้กับพอร์ตที่เราต้องการติดต่อ ในที่นี้เราเลือกพอร์ตที่ต้องการติดต่อที่ COM 1 หลังจากนั้นคลิก OK

1.4) ตั้งค่า Baud Rate ในที่นี้ตั้งค่าไว้ที่ 19200 หลังจากนั้นคลิก OK

1.5) คลิกปุ่ม AUTO STANDBY ให้ STATUS % PWM ทำงานและแสดงผล

1.6) หน้าต่าง PID Temperature Control พร้อมที่จะรับข้อมูลในการควบคุม

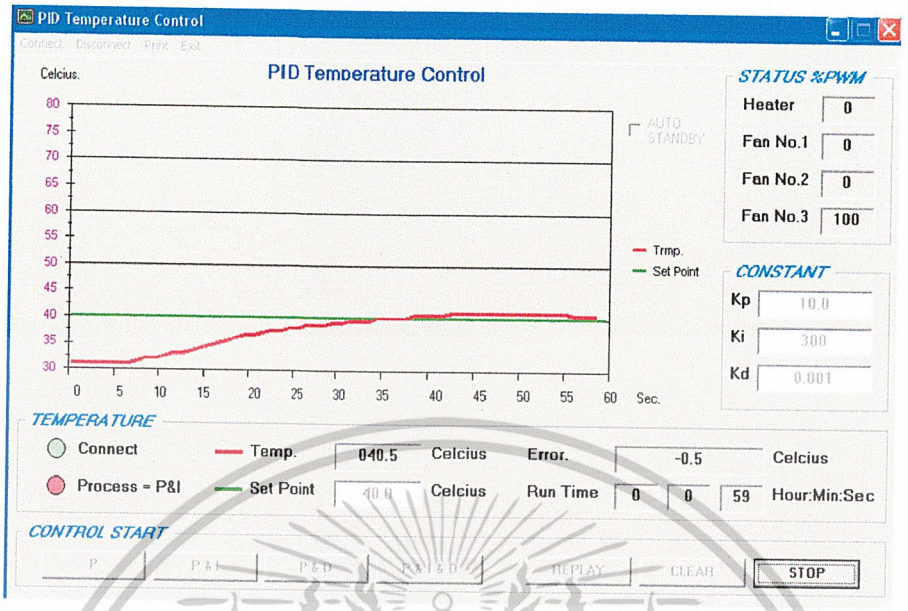
1.7) ตั้งค่ากิริยาในการควบคุมที่เราต้องการ คือ Kp = 10, Ki = 300 หรือ Kd = 0.001

1.8) เลือกปุ่มกิริยาการควบคุมที่ต้องการคือ PI

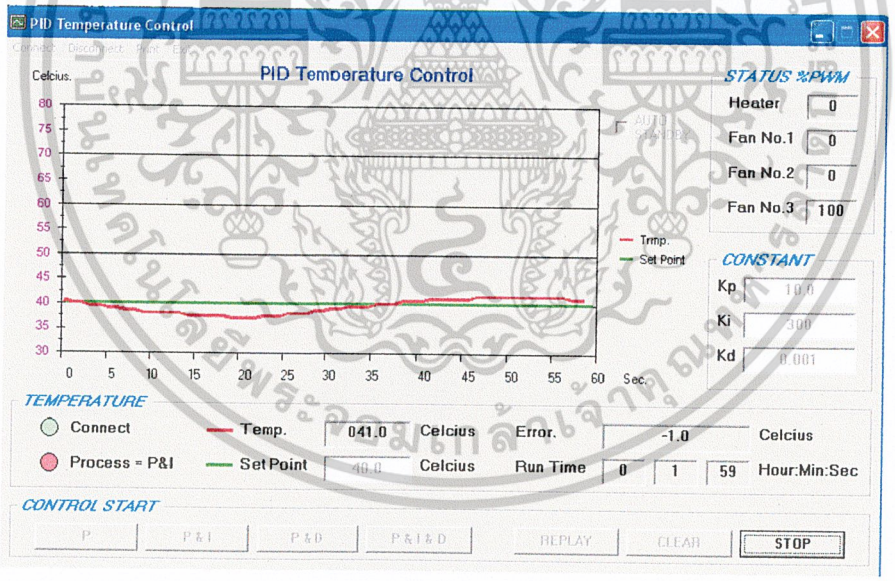
1.9) รอดูผลการทำงานได้จากกราฟที่หน้าต่าง PID Themperature Control ดังแสดงในรูปที่ 4.17 เป็นการควบคุมแบบพีไอให้เข้าสู่ค่าเป้าหมายที่ตั้งไว้ คือ 40 องศาเซลเซียส

1.10) การควบคุมแบบพีไอกำลังรักษาระดับค่าเป้าหมายให้คงที่ ดังแสดงดังรูปที่ 4.18

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.17 การควบคุมแบบพีไอ ให้เข้าสู่ค่าเป้าหมาย



รูปที่ 4.18 การควบคุมแบบพีไอ ขณะกำลังรักษาระดับค่าเป้าหมายให้คงที่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.6.4 การทดลองอุณหภูมิโดยกิริยาควบคุมแบบพีดี

1) ลำดับขั้นการทดลอง

1.1) เริ่มต้นการทำงานคลิกที่ไอคอน VB Thermo เพื่อเปิดหน้าต่าง PID Themperature Control

1.2) เปิดสวิตช์สตาร์ทชุดควบคุมจากนั้นไปที่คอมพิวเตอร์ คลิก Connect ที่หน้าต่าง PID Themperature Control

1.3) ทำการเลือก Com Port โดยการป้อนตัวเลข Com Port เพื่อส่งผ่านข้อมูลแบบอนุกรมให้กับพอร์ตที่เราต้องการติดต่อ ในที่นี้เราเลือกพอร์ตที่ต้องการติดต่อที่ COM 1 หลังจากนั้นคลิก OK

1.4)ตั้งค่า Baud Rate ในที่นี้ตั้งค่าไว้ที่ 19200 หลังจากนั้นคลิก OK

1.5) คลิกปุ่ม AUTO STANDBY ให้ STATUS % PWM ทำงานและแสดงผล

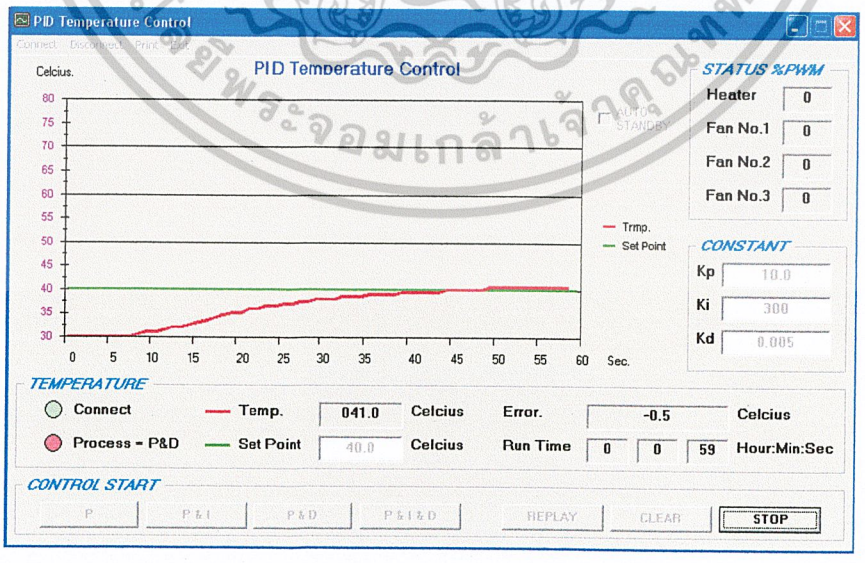
1.6) หน้าต่าง PID Temperature Control พร้อมที่จะรับข้อมูลในการควบคุม

1.7) ตั้งค่ากิริยาในการควบคุมที่เราต้องการ คือ Kp = 10, Ki = 300 หรือ Kd = 0.005

1.8) เลือกปุ่มกิริยาการควบคุมที่ต้องการคือ PD

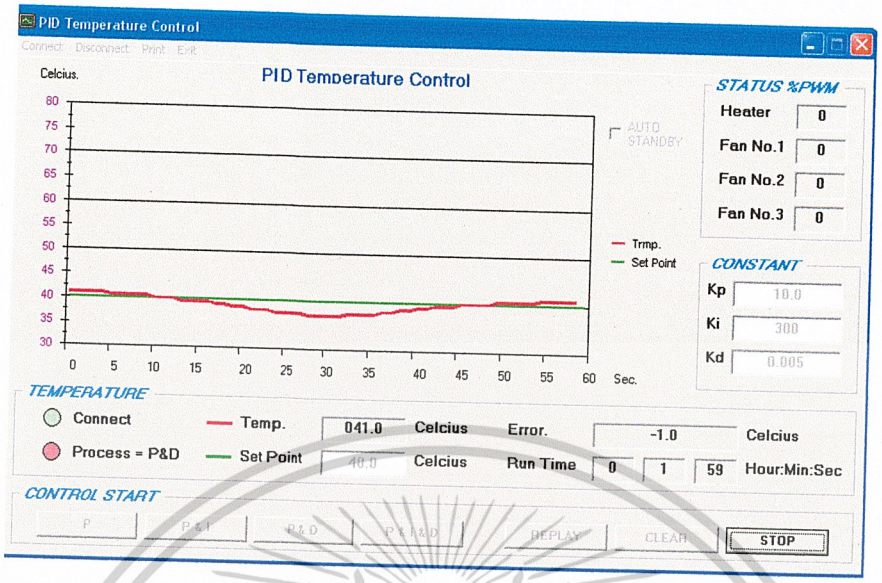
1.9) รอดูผลการทำงานได้จากกราฟที่หน้าต่าง PID Themperature Control ดังแสดงในรูปที่ 4.19 เป็นการควบคุมแบบพีดี ให้เข้าสู่ค่าเป้าหมายที่ตั้งไว้ คือ 40 องศาเซลเซียส

1.10) การควบคุมแบบพีดี กำลังรักษาระดับค่าเป้าหมายให้คงที่ ดังแสดงดังรูปที่ 4.20



รูปที่ 4.19 การควบคุมแบบพีดี ให้เข้าสู่ค่าเป้าหมาย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.20 การควบคุมแบบพีดี ขณะกำลังรักษาระดับค่าเป้าหมายให้คงที่

4.6.5 การทดลองอุณหภูมิโดยกิริยาควบคุมแบบพีไอดี

1) ลำดับขั้นการทดลอง

1.1) เริ่มต้นการทำงานคลิกที่ไอคอน VB Thermoเพื่อเปิดหน้าต่างPID Themperature Control

1.2) เปิดสวิตช์สตาร์ทชุดควบคุมจากนั้นไปที่คอมพิวเตอร์ คลิก Connect ที่หน้าต่าง PID Themperature Control

1.3) ทำการเลือก Com Port โดยการป้อนตัวเลข Com Port เพื่อส่งผ่านข้อมูลแบบอนุกรมให้กับพอร์ตที่เราต้องการติดต่อ ในที่นี้เราเลือกพอร์ตที่ต้องการติดต่อที่ COM 1 หลังจากนั้นคลิก OK

1.4) ตั้งค่า Baud Rate ในที่นี้ตั้งค่าไว้ที่ 19200 หลังจากนั้นคลิก OK

1.5) คลิกปุ่ม AUTO STANDBY ให้ STATUS % PWM ทำงานและแสดงผล

1.6) หน้าต่าง PID Temperature Control พร้อมที่จะรับข้อมูลในการควบคุม

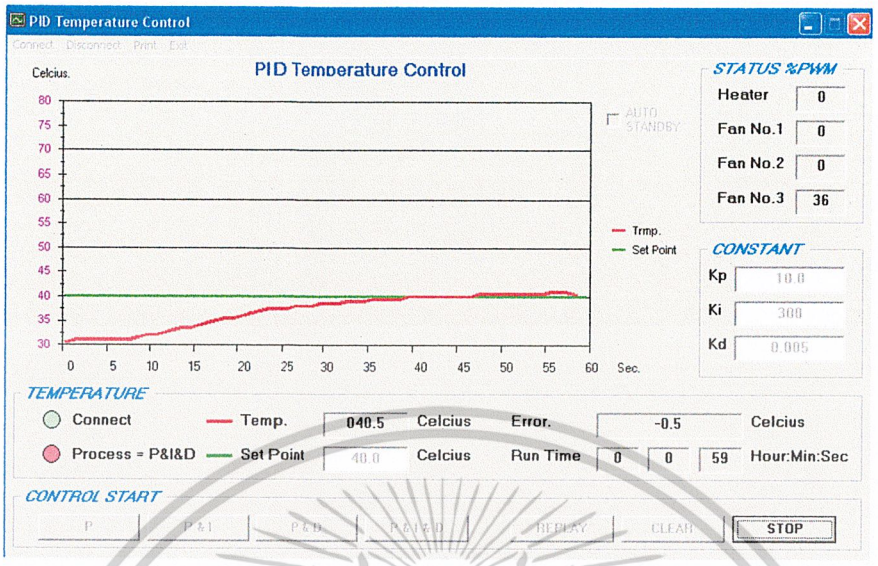
1.7) ตั้งค่ากิริยาในการควบคุมที่เราต้องการ คือ Kp = 10, Ki = 300 หรือ Kd = 0.005

1.8) เลือกปุ่มกิริยาการควบคุมที่ต้องการคือ PID

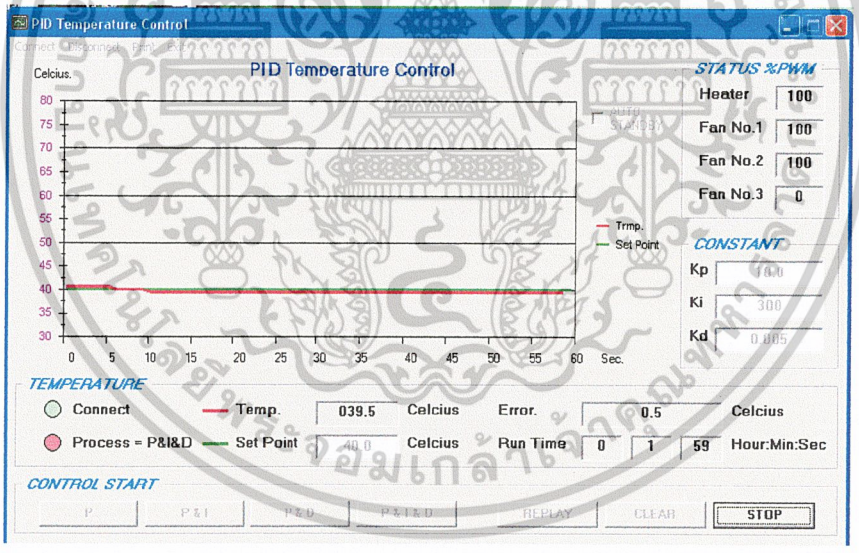
1.9) รอดูผลการทำงานได้จากกราฟที่หน้าต่าง PID Themperature Control ดังแสดงในรูปที่ 4.21 เป็นการควบคุมแบบพีไอดี ให้เข้าสู่ค่าเป้าหมายที่ตั้งไว้ คือ 40 องศาเซลเซียส

1.10) การควบคุมแบบพีไอดี รักษาระดับค่าเป้าหมายให้คงที่ ดังแสดงดังรูปที่ 4.22

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.21 การควบคุมแบบพีไอดี ให้เข้าสู่ค่าเป้าหมาย



รูปที่ 4.22 การควบคุมแบบพีไอดี ขณะกำลังรักษาระดับค่าเป้าหมายให้คงที่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่5
บทสรุป

5.1 สรุป

ในการศึกษาและทดลองสร้างชุดสาธิตการควบคุมอุณหภูมิภายในห้องจำลองแบบวงปิด เป็นแนวคิดที่จะศึกษากระบวนการควบคุมอุณหภูมิด้วย PID โดยใช้ไมโครคอนโทรลเลอร์เป็นตัวควบคุมและแสดงผลทางคอมพิวเตอร์ ได้มีการศึกษารายละเอียดของอุปกรณ์ต่างๆ เป็นอย่างมาก เพื่อให้หลักการทำงานของอุปกรณ์ที่ต้องการมาประยุกต์ใช้ในการสร้างชุดสาธิตการควบคุมอุณหภูมิภายในห้องจำลองแบบวงปิด

การสร้างชุดสาธิตการควบคุมอุณหภูมิภายในห้องจำลองแบบวงปิดได้ออกแบบโครงสร้างของห้องจำลองเป็น 3 ห้องโดยใช้แผ่นทนความร้อนและให้มีด้านที่โปร่งใสเพื่อสามารถมองเห็นการทำงานภายในห้องของพัดลมดูดความร้อนและระบายความร้อน ส่วนด้านของการออกแบบโปรแกรมศึกษาในเรื่องของไมโครคอนโทรลเลอร์ PIC 16F628 และ PIC 16F877 เพื่อนำมาควบคุมการทำงานของวงจรด้านฮาร์ดแวร์และวงจรตรวจจับอุณหภูมิ DS 18S20 ให้ส่งข้อมูลไปให้คอมพิวเตอร์ โดยการติดต่อกับคอมพิวเตอร์นี้จะมีพอร์ตอนุกรม RS 232 เป็นตัวกลางในการส่งผ่านข้อมูลระหว่างไมโครคอนโทรลเลอร์กับการแสดงผลของคอมพิวเตอร์

จากการศึกษาและจัดสร้างชุดสาธิตการควบคุมอุณหภูมิภายในห้องจำลองแบบวงปิดทำให้สามารถเข้าใจหลักการทำงานของกระบวนการควบคุมแบบ PID เข้าใจการเขียนโปรแกรมเพื่อควบคุมมอเตอร์ฮีตเตอร์ และโปรแกรมหน้าต่างที่ใช้แสดงผลทางคอมพิวเตอร์ ความสามารถของชุดสาธิตการควบคุมอุณหภูมิภายในห้องจำลองแบบวงปิดสามารถควบคุมอุณหภูมิด้วยกระบวนการทำงานแบบ PID โดยมีการสั่งงานและแสดงผลที่คอมพิวเตอร์ สามารถตั้งค่าเป้าหมายในย่านที่กำหนดได้ตามต้องการ ดังนั้นโครงงานชุดสาธิตการควบคุมอุณหภูมิภายในห้องจำลองแบบวงปิดจึงเหมาะกับการนำไปประกอบการเรียนการสอนแก่นักศึกษา

5.2 ปัญหาและแนวทางแก้ไข

จากการสร้างและการทดสอบโครงงานพบว่ามีปัญหาเกิดขึ้นซึ่งสามารถสรุปได้ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. ปัญหา การนำค่าตัวแปรต่างๆ มาเข้าสู่สมการของ PID Control ซึ่งกำหนดเขียนในวิชวลเบสิกมีการดึงค่าตัวแปรที่นำมาใส่เกิดการแทนค่าตัวแปรที่ผิด ทำให้สูตรการคำนวณค่าผิดพลาด เกิดการพล็อตกราฟที่ให้ค่าผิดพลาดตามไปด้วย

แนวทางแก้ไข ศึกษาตัวแปรแต่ละค่าให้ถูกต้องจากนั้นนำตัวแปรที่ได้จริงไปเข้าสู่สมการที่ให้เกิดการคำนวณค่าที่ถูกต้องและพล็อตกราฟได้ถูกต้องตามไปด้วย

2. ปัญหา เมื่ออุณหภูมิสูงเลยค่าเป้าหมายทำให้เกิดการลดลงของช่วงอุณหภูมิช้ากว่าขาขึ้น ทำให้มีผลต่อการควบคุมและปรับแต่งค่าอุณหภูมิได้ยากซึ่งจะใช้เวลานานขึ้น

แนวทางแก้ไข เพิ่มขนาดท่อเชื่อมระหว่างห้องจำลองให้ใหญ่ขึ้นและปรับอัตราแรงลมที่ฟีดแบล็คออกให้มอเตอร์หมุนเร็วมากขึ้นจะทำให้ถ่ายเทอากาศได้เร็วขึ้นช่วงอุณหภูมิขาลงจะกลับสู่ค่าเป้าหมายได้เร็วขึ้น

3. ปัญหา เนื่องจากโปรแกรมที่เขียนให้ไมโครคอนโทรลเลอร์แบ่งออกเป็นสองชุด ชุดหนึ่งทำหน้าที่รับข้อมูลเพียงอย่างเดียว ส่วนอีกชุดหนึ่งทำหน้าที่ทั้งรับและส่งข้อมูล จึงมีผลทำให้โปรแกรมที่เขียนให้กับไมโครคอนโทรลเลอร์เกิดการทำงานพร้อมกันไม่ได้ เนื่องจากไอซีตรวจจับอุณหภูมิมีการอินเตอร์รัพต์ตรวจวัดอุณหภูมิช้าประมาณ 2 วินาที ต่อ 1 ครั้ง จึงมีผลทำให้โปรแกรมทำงานพร้อมกันไม่ได้

แนวทางแก้ไข ใช้ไมโครคอนโทรลเลอร์แยกเป็น 2 ตัว โดยเลือกใช้ขนาดเล็กให้เหมาะสมกับขาที่เลือกใช้จากนั้นต่อขารับและส่งข้อมูลเข้าด้วยกัน

5.3 แนวทางการพัฒนา

1. การใช้ ดีซีโซลิดสเตท เป็นตัวขับชุดฮีตเตอร์แทนการใช้รีเลย์ ซึ่งจะทำให้การติดต่อไม่ซับซ้อนการทำงานจะง่ายขึ้น เนื่องจากการใช้รีเลย์จะต้องมีไฟเลี้ยงชุดหน้าสัมผัสของรีเลย์ เมื่อใช้งานนานขึ้นทำให้เกิดการเสื่อมสภาพของหน้าสัมผัส ซึ่งการใช้ ดีซีโซลิดสเตท จะสามารถใช้แทนสวิตช์ได้เลย โดยเมื่อมีสัญญาณทริกเข้ามาก็จะทำหน้าที่ตัดต่อแหล่งจ่ายไฟฟ้ากระแสตรงได้ทันที

2. การใช้บอลวาล์วเป็นตัวควบคุมลมร้อน ซึ่งหากใช้การควบคุมโดยการสร้างพัลส์ให้มอเตอร์จะช่วยการควบคุมลมไม่ต้องใช้บอลวาล์ว แต่จะให้มอเตอร์พัดลมเป็นตัวควบคุมแทนโดยการสร้างพัลซ์ความถี่จาก PIC เพื่อกำหนดความเร็วของมอเตอร์พัดลม

3. ท่อส่งลมร้อนจากที่ใช้ท่อพีวีซีให้เปลี่ยนเป็นท่อโลหะ เพราะถ้าหากฮีตเตอร์ให้ความร้อนมากๆ อาจทำให้ท่อพีวีซีเกิดการหดหรือละลายได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

กิตติ ถีรเชรษฐ. **การวิเคราะห์ระบบควบคุมเชิงเส้น** .กรุงเทพฯ : คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง. 2538

นักสิทธิ์ คูวัฒนาชัย. **การถ่ายเทความร้อน**. พิมพ์ครั้งที่ 3. กรุงเทพฯ : ฟิสิกส์เซนเตอร์. 2536

รศ.ดร.ฟูศักดิ์ ชีวสุวิทย์. **ระบบควบคุมป้อนกลับ**. กรุงเทพฯ : คณะวิศวกรรมศาสตร์สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง. 2530

สมศักดิ์ กีรติวุฒิเศรษฐ์. **หลักการและการใช้งานเครื่องมือวัดอุตสาหกรรม**. กรุงเทพฯ : คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง. 2518

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

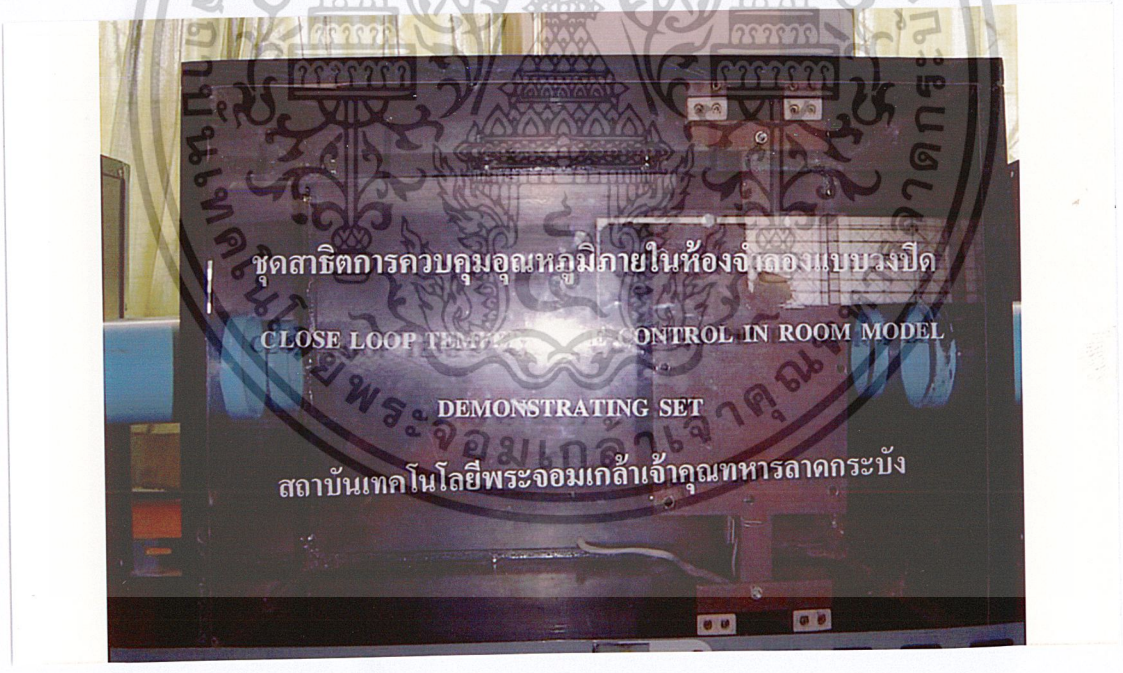


ภาคผนวก ก
เครื่องต้นแบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

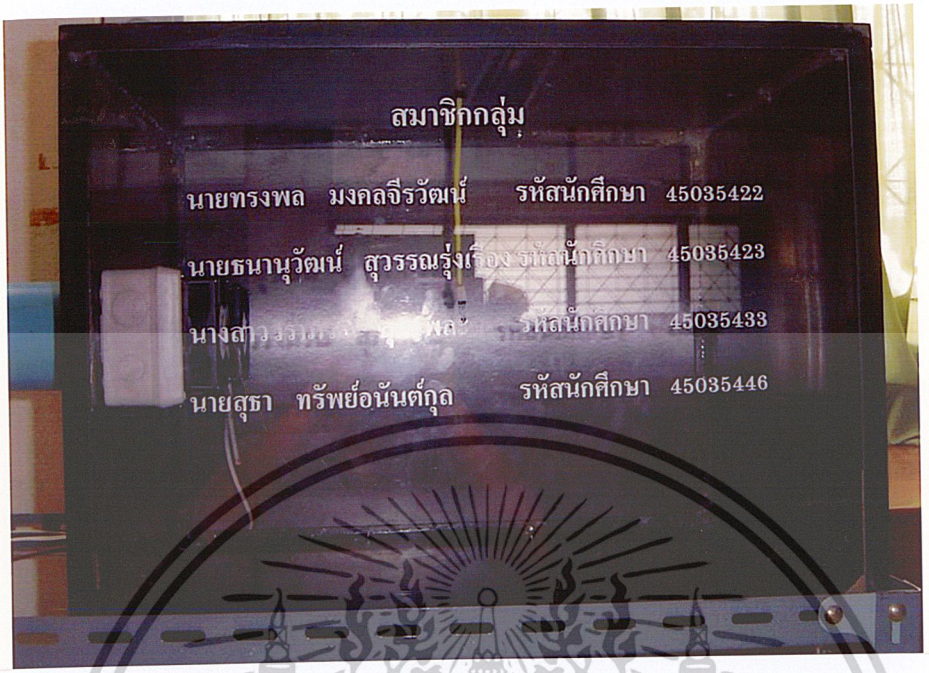


รูปที่ ก.1 ภาพห้องจำลองที่หนึ่งของชุดสาธิตการควบคุมอุณหภูมิ

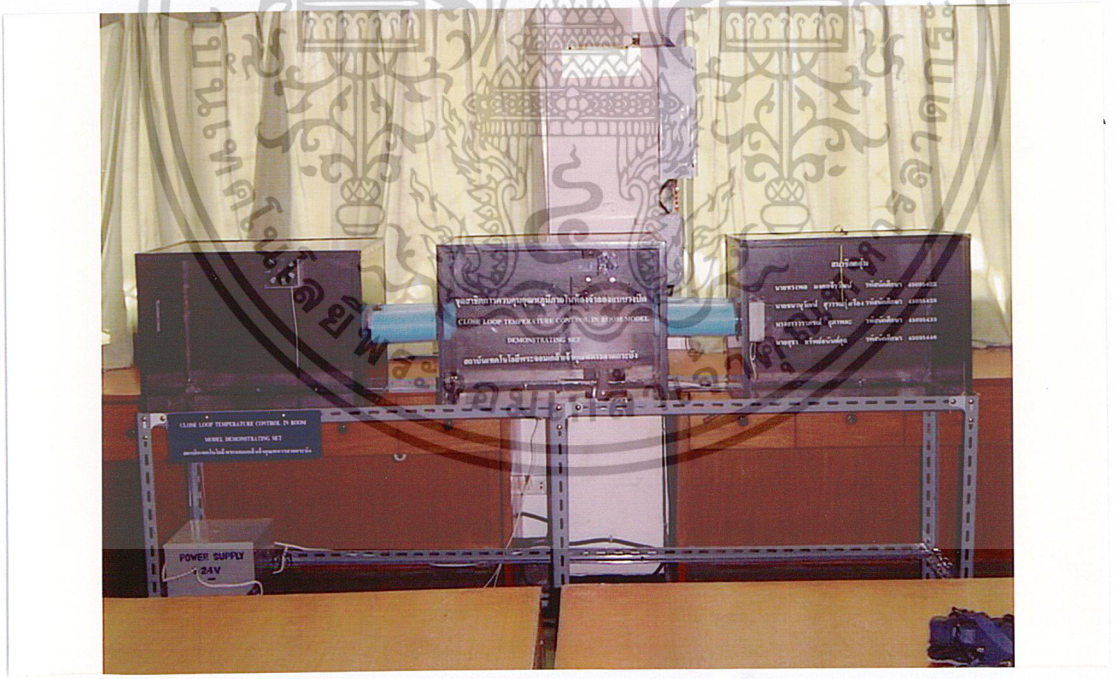


รูปที่ ก.2 ภาพห้องจำลองที่สองตัวสร้างอุณหภูมิของชุดสาธิตการควบคุมอุณหภูมิ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

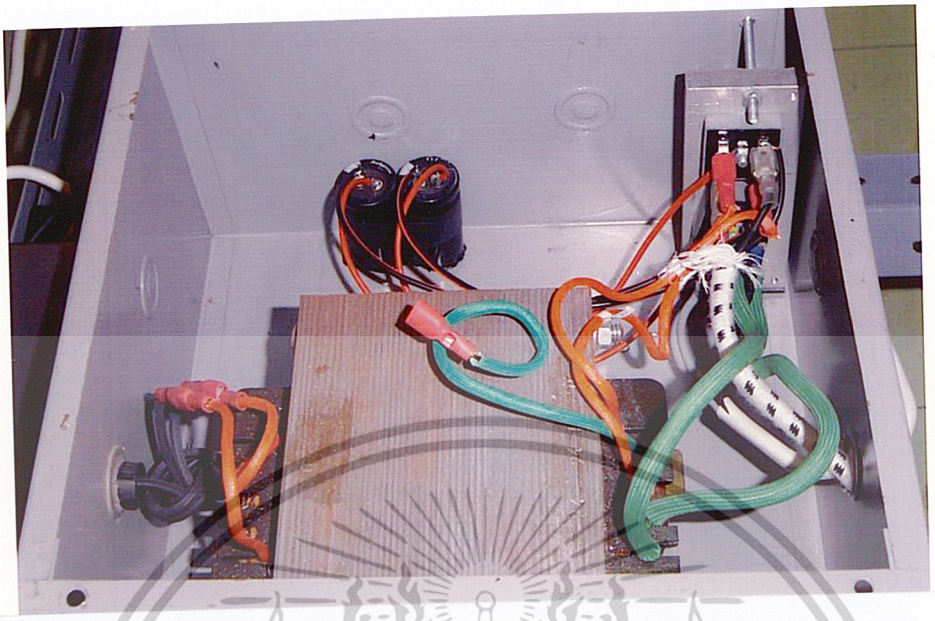


รูปที่ ก.3 ภาพห้องจำลองที่สามที่จะต้องควบคุมอุณหภูมิ

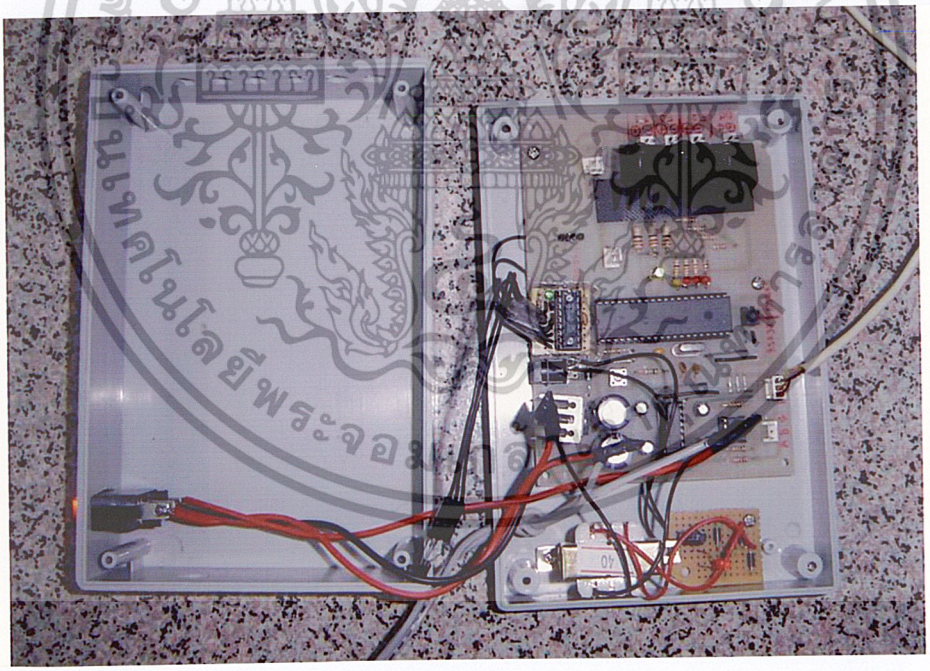


รูปที่ ก.4 ภาพห้องจำลองชุดสาธิตการควบคุมอุณหภูมิในวงปิด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ ก.5 ชุดแหล่งจ่ายไฟ 24 โวลต์



รูปที่ ก.6 ชุดควบคุมอุณหภูมิ

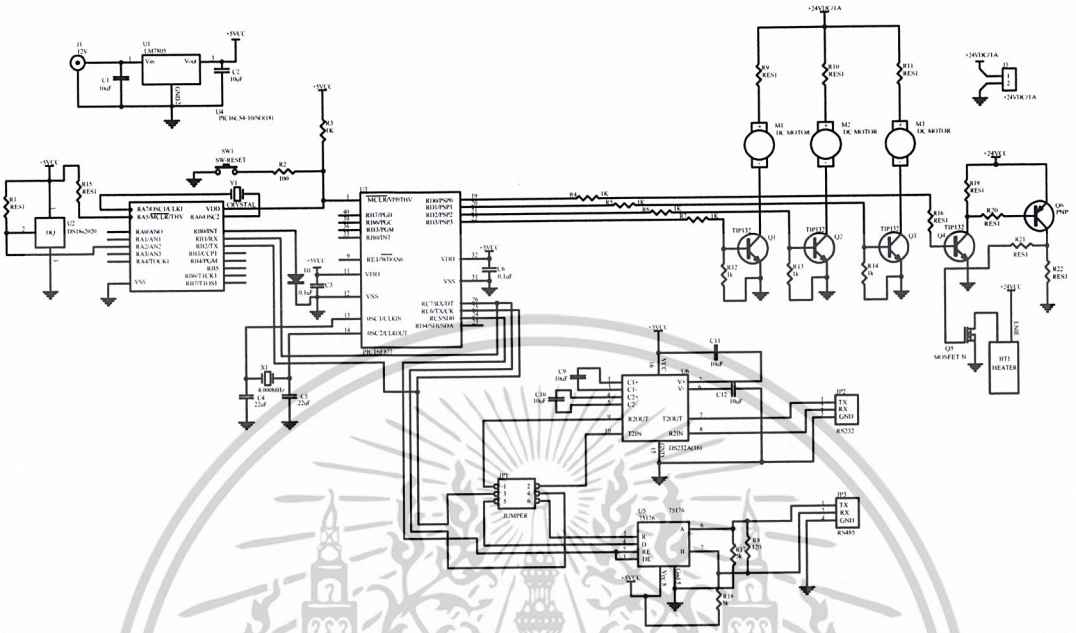
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



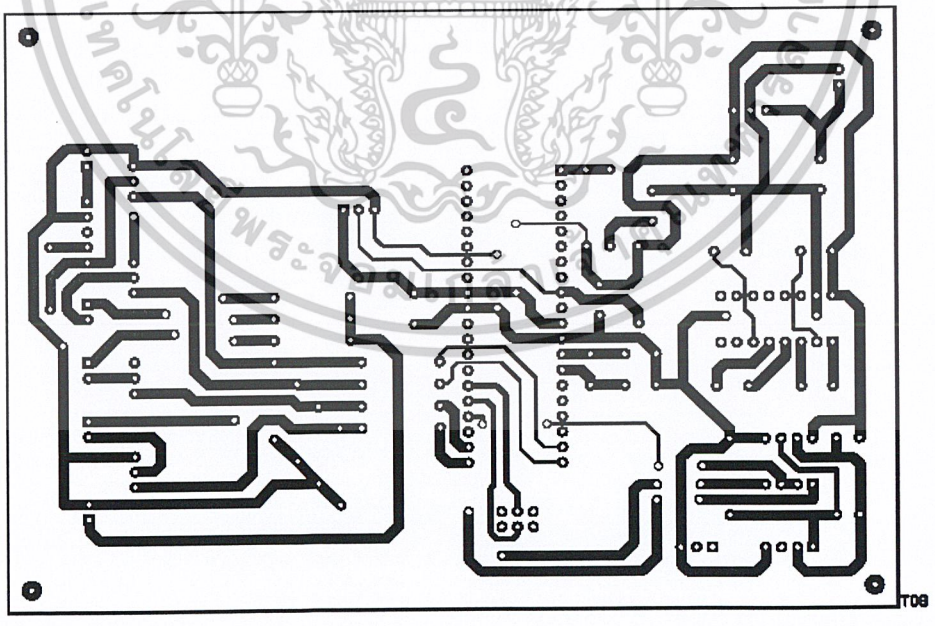
ภาคผนวก ข

วงจรและแผ่นวงจรพิมพ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ ข.1 วงจรควบคุมการทำงานของชุดสาธิตการควบคุมอุณหภูมิ



รูปที่ ข.2 แผ่นวงจรพิมพ์ควบคุมการทำงานของชุดสาธิตการควบคุมอุณหภูมิ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาคผนวก ค
รายการอุปกรณ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ ค.1 รายการอุปกรณ์ของวงจรควบคุมการตรวจวัดอุณหภูมิและวงจรไดร์เวอร์

ชื่ออุปกรณ์	รายละเอียด	จำนวน
วงจรรวม		
IC1	PIC 16F877	1 ตัว
อุปกรณ์สารกึ่งตัวนำ		
Q1, Q2, Q3, Q4	TIP 132	4 ตัว
Q5	DS 18S20	1 ตัว
ตัวเก็บประจุ		
C1, C2	22 pF	2 ตัว
C3, C4	0.1 μF อิเล็กโตรไลท์	2 ตัว
ตัวความต้านทาน		
R1, R2, R3, R4	10 kΩ	4 ตัว
R5	R Pack 9 ขา	1 ตัว
R6, R7	1 kΩ	2 ตัว
R8	100 Ω	1 ตัว
อุปกรณ์อื่นๆ		
M1, M2, M3	มอเตอร์พัดลม 24 V 0.1 A	3ตัว
K1	รีเลย์ 12 V 30 A	1 ตัว
SW1	สวิตช์กดติดปล่อยดับ	1 ตัว
X'T	X'TAL 4.00 MHz	1 ตัว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ ค.2 รายการอุปกรณ์ของวงจรส่งผ่านข้อมูลแบบอนุกรม

ชื่ออุปกรณ์	รายละเอียด	จำนวน
วงจรรวม		
IC1	MAX 232	1 ตัว
IC2	75176	1 ตัว
ตัวเก็บประจุ		
C1, C2, C3, C4	10 μF 25 V อิเล็กโตรไลท์	4 ตัว
ตัวความต้านทาน		
R1	120 Ω	1 ตัว
อุปกรณ์อื่นๆ		
J1, J2	DB 9	2 ตัว
J3	Jumper	3 คู่

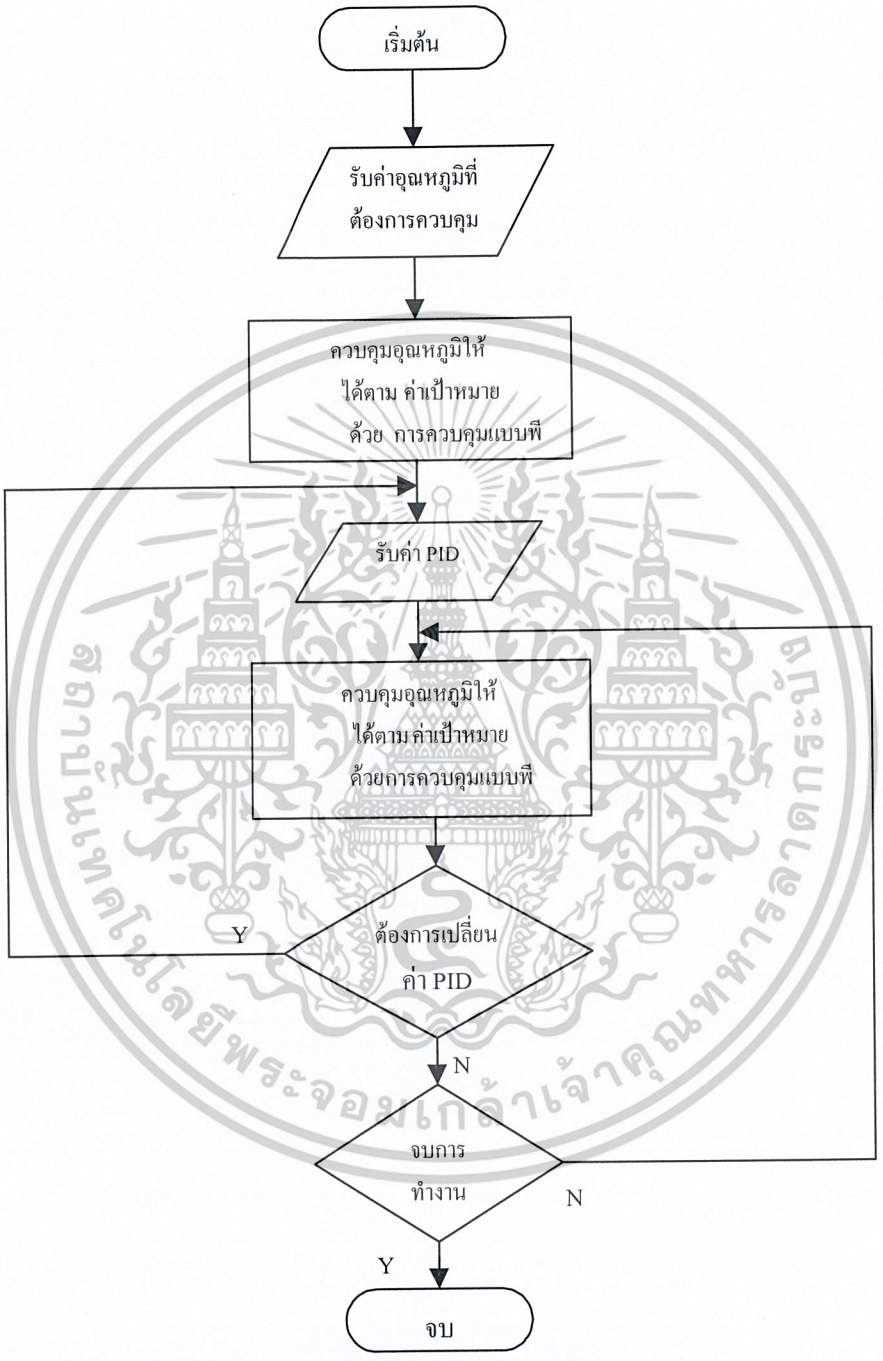
ตารางที่ ค.3 รายการอุปกรณ์ของวงจรแหล่งจ่ายแรงดัน

ชื่ออุปกรณ์	รายละเอียด	จำนวน
วงจรรวม		
IC1	LM7812CT	1 ตัว
IC2	LM7805CT	1 ตัว
ตัวเก็บประจุ		
C1, C2, C3	10 μF 50 V	3 ตัว
อุปกรณ์อื่นๆ		
J1	JACK ตัวผู้	1 ตัว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ง
แผนผังการทำงานและรหัสต้นฉบับของโปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ ง.1 ผังงานการควบคุมของชุดสาธิตการควบคุมอุณหภูมิ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ ง.2 ผังงานโปรแกรม DS18S20

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรมการใช้ดีเลย์หน่วงเวลาในการคอมไพร์

```
/*
 *     Delay functions
 *     See delay.h for details
 *
 *     Make sure this code is compiled with full optimization!!!
 */

#include    "delay.h"

void
DelayMs(unsigned char cnt)
{
#if   XTAL_FREQ <= 2MHZ
      do {
            DelayUs(996);
      } while(--cnt);
#endif

#if    XTAL_FREQ > 2MHZ
      unsigned char     i;
      do {
            i = 4;
            do {
                  DelayUs(250);
            } while(--i);
      } while(--cnt);
#endif
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรมของหน้าต่างแสดงผล

```
Option Explicit
Private CommPort As Integer                          ' Select ComPort
Private BaudRate As String                         ' Setting   BaudRate
Private strRxBuffer As String

Private NProcess As String                  ' P, PI ,PD ,PID
Private SecondCount As Integer              ' between 1 to 60 sec

Private H1Buffer As Integer
Private F1Buffer As Integer
Private F2Buffer As Integer
Private F3Buffer As Integer

Private Uk0 As Integer
Private UK1 As Integer
Private Uk2 As Integer
Private Ek0 As Integer
Private EK1 As Integer
Private Ek2 As Integer


Private Sub chkDefault_Click()
        If Not chkDefault.Value Then
                Call iniPWM
        Else
                Call ClearPWMBuffer
                Call PWMOut
        End If
End Sub

Private Sub cmdClear_Click()
        Call ClearGraphCurrent
        lblProcess.Caption = "Process"
        lblTimeHH.Caption = "0"
        lblTimeMM.Caption = "0"
        lblTimeSS.Caption = "0"

        SecondCount = 0

        chkDefault.Value = False

End Sub

Private Sub cmdProcess_Click(Index As Integer)
' Start Process
Dim strSQL As String

On Error GoTo ErrorHand
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        If rsThermo.State Then rsThermo.Close

        strSQL = "select * from TableThermo"
        If OpenRecordsetFromConnection(cnThermo, rsThermo, strSQL) <>
0 Then

                mnuDisconnect.Enabled = False
                mnuPrint.Enabled = False

                chkDefault.Enabled = False

                cmdClear.Enabled = False
                cmdStop.Enabled = True
                cmdReplay.Enabled = False

                cmdProcess(0).Enabled = False
                cmdProcess(1).Enabled = False
                cmdProcess(2).Enabled = False
                cmdProcess(3).Enabled = False
                txtControl.Enabled = False
                txtKp.Enabled = False
                txtKi.Enabled = False
                txtKd.Enabled = False

                Call ClearPWMBuffer
                Call PWMOut

                Select Case Index
                        Case 0
                                Call iniPWM
                                NProcess = "P"
                                lblProcess = "Process = P"
                        Case 1
                                NProcess = "PI"
                                lblProcess = "Process = P&&I"
                        Case 2
                                NProcess = "PD"
                                lblProcess = "Process = P&&D"
                        Case 3
                                NProcess = "PID"
                                lblProcess = "Process = P&&I&&D"
                        Case Else
                                NProcess = ""
                                lblProcess = "Process"
                End Select

                SecondCount = 1

                lblTimeHH.Caption = "0"
                lblTimeMM.Caption = "0"
                Call ClearGraphCurrent
                Call LoadGraphControl
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                shpProcess.BackColor = &H8080FF

        Else
                MsgBox "???????? ??????????????????.", vbCritical,
"PID Temperature Control"

        End If

Exit Sub
ErrorHand:
        MsgBox Err.Description
End Sub

Private Sub cmdReplay_Click()
Dim strSQL As String

On Error GoTo ErrorHand

        If rsThermo.State Then rsThermo.Close

        strSQL = "select * from TableThermo"
        If OpenRecordsetFromConnection(cnThermo, rsThermo, strSQL) <>
0 Then

                mnuDisconnect.Enabled = False
                mnuPrint.Enabled = False

                chkDefault.Enabled = False

                cmdClear.Enabled = False
                cmdStop.Enabled = True
                cmdReplay.Enabled = False

                cmdProcess(0).Enabled = False
                cmdProcess(1).Enabled = False
                cmdProcess(2).Enabled = False
                cmdProcess(3).Enabled = False
                txtControl.Enabled = False
                txtKp.Enabled = False
                txtKi.Enabled = False
                txtKd.Enabled = False

                lblProcess = "Replay"

                SecondCount = 1

                Call ClearGraphCurrent
                Call LoadGraphControl

                Call ClearPWMBuffer
                Call PWMOut

                tmrReplay.Enabled = True
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                shpProcess.BackColor = &H8080FF

        Else
                MsgBox "???????? ??????????????????.", vbCritical,
"PID Temperature Control"

        End If

Exit Sub
ErrorHand:
        MsgBox Err.Description

End Sub

Private Sub cmdStop_Click()
On Error GoTo ErrorHand

        tmrReplay.Enabled = False

        mnuDisconnect.Enabled = True
        mnuPrint.Enabled = True

        chkDefault.Enabled = True

        cmdStop.Enabled = False
        cmdClear.Enabled = True

        cmdReplay.Enabled = True

        cmdProcess(0).Enabled = True
        cmdProcess(1).Enabled = True
        cmdProcess(2).Enabled = True
        cmdProcess(3).Enabled = True
        txtControl.Enabled = True
        txtKp.Enabled = True
        txtKi.Enabled = True
        txtKd.Enabled = True

        shpProcess.BackColor = Me.BackColor

        NProcess = ""                       ' Normal
        lblProcess.Caption = "Process"

        Call ClearPWMBuffer
        Call PWMOut

        rsThermo.Close

Exit Sub
ErrorHand:
        MsgBox Err.Description & vbCrLf & "- - - Stop Error! - - -"
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
End Sub

Private Sub Form_Load()

    Me.Left = (Screen.Width - Width) \ 2
    Me.Top = (Screen.Height - (Height + 400)) \ 2

        Call iniCommPort        ' initial CommPort

        mnuConnect.Enabled = True
        mnuDisconnect.Enabled = False
        mnuPrint.Enabled = True
        mnuExit.Enabled = True

        cmdStop.Enabled = False
        cmdClear.Enabled = False
        cmdReplay.Enabled = False

        cmdProcess(0).Enabled = False
        cmdProcess(1).Enabled = False
        cmdProcess(2).Enabled = False
        cmdProcess(3).Enabled = False
        txtControl.Enabled = False
        txtKp.Enabled = False
        txtKi.Enabled = False
        txtKd.Enabled = False

    Set cnThermo = New ADODB.Connection
    cnThermo.Open "Provider=Microsoft.Jet.OLEDB.4.0;Persist Security
Info=False;Data Source = " & "dbThermo.mdb"

    Set rsThermo = New ADODB.Recordset

End Sub

Private Sub Form_Unload(Cancel As Integer)

        On Error Resume Next
                rsThermo.Close
                Set cnThermo = Nothing
                cnThermo.Close
                Set cnThermo = Nothing
                End

End Sub

Private Sub mnuConnect_Click()

On Error GoTo ErrorHand

        CommPort = InputBox("??????????? ComPort " & vbCrLf & "1, 2,
3, .....", "SettingPort = Com " & CommPort & ", BaudRate = " &
BaudRate, CommPort)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        BaudRate = InputBox("??????????? BaudRate" & vbCrLf & "9600,
19200, .....", "SettingPort = Com " & CommPort & ", BaudRate = " &
BaudRate, BaudRate)

        If Comm_Open Then

                mnuExit.Enabled = False
                mnuPrint.Enabled = True
                mnuConnect.Enabled = False
                mnuDisconnect.Enabled = True

                chkDefault.Enabled = True

                cmdStop.Enabled = False
                cmdClear.Enabled = True
                cmdReplay.Enabled = True

                cmdProcess(0).Enabled = True
                cmdProcess(1).Enabled = True
                cmdProcess(2).Enabled = True
                cmdProcess(3).Enabled = True
                txtControl.Enabled = True
                txtKp.Enabled = True
                txtKi.Enabled = True
                txtKd.Enabled = True

                shpConnect.BackColor = &HC0FFC0

                Call ClearPWMBuffer
                Call PWMOut

                NProcess = ""
                tmrProcess.Enabled = True

        Else
                MsgBox "???????????? CommPort ???????? BaudRate " &
vbCrLf & " ??? Connect ????"

        End If

Exit Sub
ErrorHand:
        MsgBox "?????????????????"

End Sub

Private Sub mnuDisconnect_Click()

        Call ClearPWMBuffer
        Call PWMOut

        Call Comm_Close
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        mnuDisconnect.Enabled = False
        mnuConnect.Enabled = True
        mnuExit.Enabled = True
        mnuPrint.Enabled = True

        chkDefault.Enabled = False

        cmdStop.Enabled = False
        cmdClear.Enabled = False

        cmdReplay.Enabled = False

        cmdProcess(0).Enabled = False
        cmdProcess(1).Enabled = False
        cmdProcess(2).Enabled = False
        cmdProcess(3).Enabled = False
        txtControl.Enabled = False
        txtKp.Enabled = False
        txtKi.Enabled = False
        txtKd.Enabled = False

        tmrProcess.Enabled = False

        shpConnect.BackColor = Me.BackColor
End Sub

Private Sub mnuExit_Click()
        If MsgBox("?????? ?????????? ????????????? ?", vbQuestion +
vbYesNo, "Temperature Control") = vbYes Then
                Unload Me
        End If
End Sub

Public Sub iniCommPort()
        CommPort = 1                                              '
Default  Port = Com1
        BaudRate = "19200"                                  ' Default
Baud = 9600,19200
        strRxBuffer = ""
End Sub

Public Function Comm_Open() As Boolean
'?????????? ???????????? Port
On Error GoTo ErrorHand

                MSComm1.CommPort = CommPort                       '
Select CommPort
                MSComm1.Settings = BaudRate & ",N,8,1"            '
Setting BaudRate
                If Not MSComm1.PortOpen Then MSComm1.PortOpen = True
                MSComm1.DTREnable = True
                'MSComm1.RTSEnable = True
                MSComm1.RThreshold = 1
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                MSComm1.InputLen = 1

                Comm_Open = True                        ' Open Comm Port =
Complete

Exit Function

ErrorHand:
    MsgBox Err.Description
    MSComm1.PortOpen = False
    Comm_Open = False                           ' Can't Open Comm Port

End Function

Private Sub Comm_Close()                        '??? ??? Reset
???????????????????
On Error Resume Next

                MSComm1.DTREnable = False
                MSComm1.RTSEnable = False
                MSComm1.PortOpen = False

End Sub

Private Sub Comm_Send(strValue As String)               ' ?? ??????
Dim i As Integer, Nx As Integer
Dim strTemp As String

On Error GoTo ErrorHand

                MSComm1.RTSEnable = True
                Nx = Len(strValue)

                Do While Nx > 0
                        strTemp = Right$(strValue, Nx)
                        strTemp = Left$(strTemp, 1)
                        MSComm1.Output = strTemp
' Send String
                        Nx = Nx - 1
                Loop

                MSComm1.RTSEnable = False

Exit Sub

ErrorHand:
    MsgBox Err.Description

End Sub

Private Sub mnuPrint_Click()
On Error GoTo Err_Hand
    CommonDialog1.CancelError = True
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    CommonDialog1.ShowPrinter
    Me.PrintForm
    Exit Sub
Err_Hand:
    If Err.Number <> 32755 Then MsgBox
"?????????????????????? ?????????????", vbInformation,
"???????????"

End Sub

Private Sub MSComm1_OnComm()
Dim RxBuffer As Variant

On Error GoTo ErrorHand

        Select Case MSComm1.CommEvent
              Case comEvReceive
                    RxBuffer = MSComm1.Input
                    If (RxBuffer = " ") Then
                          lblCurrent.Caption =
IIf(IsNumeric(strRxBuffer), strRxBuffer, lblCurrent.Caption)
                          strRxBuffer = ""
                    Else
                          If (RxBuffer = "e") Then
                                strRxBuffer = ""
                                'MsgBox "??????? ??? DS1820.",
vbCritical, "Check Hardware"
                          Else
                                strRxBuffer = strRxBuffer &
RxBuffer
                          End If
                    End If
              Case comEvSend:
'????????????? ?????????????????????????????? ?????????
              Case comEvCTS
              Case comEvDSR
              Case comEvCD
              Case comEvRing
              Case comEvEOF
              Case comBreak
              Case comCDTO
              Case comCTSTO
              Case comDCB
              Case comDSRTO
              Case comFrame
              Case comOverrun
              Case comRxOver
              Case comRxParity
              Case comTxFull
        End Select

Exit Sub
ErrorHand:
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        MsgBox Err.Description

End Sub

Private Sub tmrProcess_Timer()
Dim tempCurrent As Double, tempControl As Double

        tempCurrent = IIf(IsNumeric(lblCurrent.Caption),
lblCurrent.Caption, 40)
        tempControl = IIf(IsNumeric(txtControl.Text), txtControl.Text,
40)
        lblError.Caption = Format((tempControl - tempCurrent),
"###0.0")

        Select Case NProcess
                Case "P"

                        Call LoadGraphCurrent

                        Uk0 = lblError.Caption * txtKp.Text

                        If tempCurrent <> tempControl Then

                                        If tempCurrent < tempControl
Then                    ' Number of Celcius
                                                If H1Buffer < 100 Then
' Number of PWM Max
                                                        H1Buffer =
H1Buffer + Uk0
                                                        F1Buffer =
H1Buffer
                                                Else
                                                        If F2Buffer >
30 Then                 ' Number of PWM Min
F2Buffer = F2Buffer - Uk0
F3Buffer = F2Buffer
                                                        End If
                                                End If
                                        Else
' else CurrentThermo > 40
                                                If H1Buffer > 30 Then
' Number of PWM Max
                                                        H1Buffer =
H1Buffer + Uk0
                                                        F1Buffer =
H1Buffer
                                                Else
                                                        If F2Buffer <
100 Then                ' Number of PWM Min
F2Buffer = F2Buffer - Uk0
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
F3Buffer = F2Buffer
                                                        End If
                                                    End If
                                                End If
                                            End If

                                            If H1Buffer > 100 Then H1Buffer = 100
                                             If F1Buffer > 100 Then F1Buffer = 100
                                              If F2Buffer > 100 Then F2Buffer = 100
                                               If F3Buffer > 100 Then F3Buffer = 100
                                            If H1Buffer < 30 Then H1Buffer = 30
                                             If F1Buffer < 30 Then F1Buffer = 30
                                              If F2Buffer < 30 Then F2Buffer = 30
                                               If F3Buffer < 30 Then F3Buffer = 30

                                    Case "PI"
                                            Call LoadGraphCurrent

                                    Case "PD"
                                            Call LoadGraphCurrent

                                    Case "PID"
                                            Call LoadGraphCurrent

                                    Case Else       ' Normal

                                        If chkDefault.Value Then

                                        If tempCurrent <> tempControl Then

                                                            If tempCurrent < tempControl
Then                        ' Number of Celcius
                                                        If H1Buffer < 100 Then
' Number of PWM Max
                                                            H1Buffer =
H1Buffer + 1
                                                            F1Buffer =
H1Buffer
                                                    Else
                                                        If F2Buffer >
30 Then                     ' Number of PWM Min
F2Buffer = F2Buffer - 1
F3Buffer = F2Buffer
                                                        End If
                                                    End If
                                                Else
' else CurrentThermo > 40
                                                    If H1Buffer > 30 Then
' Number of PWM Max
                                                            H1Buffer =
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
H1Buffer - 1
                                                        F1Buffer =
H1Buffer
                                                Else
                                                        If F2Buffer <
100 Then                        ' Number of PWM Min

F2Buffer = F2Buffer + 1

F3Buffer = F2Buffer
                                                        End If
                                                End If
                                        End If

                        End If

                    Else

                        Call ClearPWMBuffer

                    End If

        End Select

        Call PWMOut
        Call Comm_Send("t ")

End Sub

Private Sub LoadGraphControl()
Dim i As Integer

        With MSChart1

                .Column = 2
                For i = 1 To 60 Step 1
                        .Row = i
                        .Data = txtControl.Text
                Next i

        End With

End Sub

Private Sub LoadGraphCurrent()
Dim i As Integer

On Error GoTo ErrorHand


                MSChart1.Column = 1
                MSChart1.Row = SecondCount
                MSChart1.Data = lblCurrent.Caption
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                rsThermo(1).Value = MSChart1.Data

                rsThermo.MoveNext
                lblTimeSS.Caption = SecondCount
                SecondCount = SecondCount + 1

                If rsThermo.EOF Or (SecondCount > 60) Then

                        Call ClearGraphCurrent
                        Call LoadGraphControl

                        SecondCount = 1
                        lblTimeMM.Caption = (lblTimeMM.Caption + 1)
                        If (lblTimeMM.Caption >= 60) Then
                                lblTimeMM.Caption = 0
                                lblTimeHH.Caption = (lblTimeHH.Caption
+ 1)
                        End If

                        rsThermo.MoveFirst
'                        Call cmdStop_Click
                End If

Exit Sub
ErrorHand:
        MsgBox Err.Description

End Sub

Private Sub LoadGraphReplay()
Dim i As Integer

On Error GoTo ErrorHand

                If Not IsNull(rsThermo(1).Value) Then
                        MSChart1.Column = 1
                        MSChart1.Row = SecondCount
                        MSChart1.Data = rsThermo(1).Value
                        lblCurrent.Caption = Format(MSChart1.Data,
"###0.0")

                        rsThermo.MoveNext
                        SecondCount = SecondCount + 1
                        If rsThermo.EOF Or (SecondCount > 60) Then
                                Call cmdStop_Click
                        End If

                Else
                        Call cmdStop_Click

                End If
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Exit Sub
ErrorHand:
        MsgBox Err.Description

End Sub

Private Sub ClearGraphCurrent()
Dim i As Integer

        MSChart1.ColumnCount = 0
        MSChart1.Refresh
        MSChart1.ColumnCount = 2

        MSChart1.Column = 1
        MSChart1.ColumnLabel = "Trmp."
        MSChart1.Column = 2
        MSChart1.ColumnLabel = "Set Point"

'         lblTimeHH.Caption = "0"
'         lblTimeMM.Caption = "0"
        lblTimeSS.Caption = "0"

End Sub

Private Sub PWMOut()

On Error GoTo ErrorHand

        lblHeater.Caption = H1Buffer
        Comm_Send ("00" & H1Buffer & "h1 ")

        lblFan1.Caption = F1Buffer
        Comm_Send ("00" & F1Buffer & "f1 ")

        lblFan2.Caption = F2Buffer
        Comm_Send ("00" & F2Buffer & "f2 ")

        lblFan3.Caption = F3Buffer
        Comm_Send ("00" & F3Buffer & "f3 ")

Exit Sub
ErrorHand:
        MsgBox Err.Description

End Sub

Private Sub iniPWM()

        H1Buffer = 50
        F1Buffer = 50
        F2Buffer = 50
        F3Buffer = 50
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        Call PWMOut

End Sub

Private Sub ClearPWMBuffer()

        H1Buffer = 0
        F1Buffer = 0
        F2Buffer = 0
        F3Buffer = 0

End Sub

Private Sub tmrReplay_Timer()
Dim tempCurrent As Double, tempControl As Double

        Call LoadGraphReplay

        tempCurrent = IIf(IsNumeric(lblCurrent.Caption),
lblCurrent.Caption, 40)
        tempControl = IIf(IsNumeric(txtControl.Text), txtControl.Text,
40)
        lblError.Caption = Format((lblCurrent.Caption -
txtControl.Text), "###0.0")
        lblTimeSS.Caption = SecondCount - 1

End Sub

Private Sub txtControl_Change()

        If (IsNumeric(txtControl.Text)) Then
                If ((Len(txtControl.Text) >= 2) And ((txtControl.Text
< 40) Or (txtControl.Text > 70))) Then
                        MsgBox "?????????????? 40  ?  70
? ??????????", vbInformation, " Temperature Control"
                        txtControl.Text = "40.0"
                End If
        Else
                If (Len(txtControl.Text) > 0) Then
                        MsgBox "?????????????? ??????????",
vbInformation, " Temperature Control"
                        txtControl.Text = "40.0"
                End If
        End If

End Sub

Private Sub txtKd_Change()
        If (IsNumeric(txtKd.Text)) Then
                ' NOP
        Else
                If (Len(txtKd.Text) > 0) Then
                        MsgBox "?????????????? ??????????",
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
vbInformation, " Temperature Control"
                        txtKd.Text = "0.001"
                End If
        End If
End Sub

Private Sub txtKi_Change()
        If (IsNumeric(txtKi.Text)) Then
                ' NOP
        Else
                If (Len(txtKi.Text) > 0) Then
                        MsgBox "????????????? ??????????",
vbInformation, " Temperature Control"
                        txtKi.Text = "100"
                End If
        End If
End Sub

Private Sub txtKp_Change()
        If (IsNumeric(txtKp.Text)) Then
                ' NOP
        Else
                If (Len(txtKp.Text) > 0) Then
                        MsgBox "????????????? ??????????",
vbInformation, " Temperature Control"
                        txtKp.Text = "1.0"
                End If
        End If
End Sub
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรมการตรวจวัดอุณหภูมิ

```
// tctrl08 read DS1820 ONLY BY PIC16F628

#include <pic.h>
#include <stdlib.h>

#define XTAL_FREQ 4MHZ

#include "delay.h"
#include "delay.c"

/*--PORT-STRUCTURE-------------------------------------------------------
--------*/
//P16F628
//RB1 USART_Rx     I/P
//RB2 USART_Tx     O/P
//RB3 USART_TxEN   O/P
//RA2 LED          O/P
//RA3 DQ DS1820    I/O

/*--PORT-----------------------------------------------------------------
--------*/
#define     PIN(xname,xport,xbit)          static    bit    xname     @
((unsigned)&xport*8+xbit)
PIN (TxEn, PORTB, 3);          // TxEnable RS485

PIN (DQ_LED, PORTA, 2);        // PIN LED BLINK! DQ READ

PIN (DQ, PORTA, 3);            // PIN DQ DS1820
PIN (DQ_DIR, TRISA, 3);        // PIN DQ Direction input/output

/*--ASCIIConvert---------------------------------------------------------
---------*/
#define NumFilter 0b00001111   // AND to use (30-39H) -> (0-9D)
#define NumConvert      0b00110000  // OR to use (0-9D) -> (30-39H)

/*--USART----------------------------------------------------------------
--------*/
static unsigned char    RxBuffer,    // Usart Receive buffer
                  TxBuffer;    // Usart Transmit buffer

static unsigned char    RxTemp0,     // Usart Receive buffer
                  RxTemp1;     // Usart Receive buffer

/*--THERMO DIGIT---------------------------------------------------------
---------*/
static unsigned char  thermo100, thermo10, thermo1, thermo0;      //
Thermo Unitxxx
static unsigned char  ReadStatus;   //  0  ==  out  Process,  1  ==  in
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Process
static unsigned char  ReadError;     // 0 == out NonError, 1 == Error

/*-------------------------------------------------------------------------
---------*/

// ' Delay
void Delay_100Ms(unsigned int dx);

// ' Variable
void iniVariable(void);

// ' iniPort
void iniPort(void);

// ' iniTimer
void iniTimer0(void);

// ' Usart
void iniUsart(void);
void UsartTx(unsigned char TxValue);
void UsartTx_INT(unsigned char TxValue);

// ' iniInterupt
void iniInterupt(void);
static void interrupt isr(void);

// ' SumData
unsigned int SumData(void);

// ' DS1820
void ReadThermometer(void);
unsigned int init1820(void);
void write1820(unsigned char cmd);
unsigned char read1820(void);

// Convert thermo
void ConvertThermo(unsigned char tx);
int num_to_char(int val);
void PrintThermo(void);

/*-------------------------------------------------------------------------
---------*/

// ' main()
void main(void)
{
      iniPort();  // initial port
      iniTimer0();      // initial Timer0
      iniUsart(); // initial Usart
      iniVariable();    // initial Variable
      iniInterupt();    // initial Intrupt
while(1)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
      {
            CLRWDT();   // Idly kick the dog

            ReadThermometer();
      }
}

void Delay_100Ms(unsigned int dx)
{
      do
      {
            DelayMs(100);
      } while(--dx) ;
}

void iniVariable(void)
{
      DQ_LED = 0; // Clear Show DS1820 Connect & Read
      ReadStatus = 0;    // Out Read Process
      ReadError = 0;     // Non Error
}

void iniPort(void)
{
      PORTA = 0x00;              // turn Off all
      PORTB = 0x00;              // turn Off all
//          PORTC = 0x00;              // turn Off all
//    PORTD = 0x00;              // turn Off all
//    PORTE = 0x00;              // turn Off all

      TRISA = 0b00001000;        // 1=Input , 0=Output
      //ADCON1 = 7;              // PORTA all Set to Digital
      CMCON = 0x07;              // (P16F628) all Set to Digital
      TRISB = 0b00000011;        // 1=Input , 0=Output
      RBPU = 1;                  //  OPTION_REG<6>  (ActiveLow)  PORTB  Pull-up
Enable bit
//    TRISC = 0b10000000;        // all bits output
//    TRISD = 0x00;              // all bits output
//    TRISE = 0;           // all bits output
}

void iniTimer0(void)           // Initial (TMR0) Timer0
{
      T0CS = 0;            // OPTION_REG<5> TMR0 Clock Source Select bit
                           // 1 = Transition on RA4/T0CKI pin
                           //  0  =  Internal  instruction  cycle  clock
(CLKOUT)
      T0SE = 1;            // OPTION_REG<4> TMR0 Source Edge Select bit
(Use in Ext.Clock RA4/T0CLK)
                           // 1 = Increment on high-to-low transition on
RA4/T0CKI pin
                           // 0 = Increment on low-to-high transition on
RA4/T0CKI pin
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
PSA = 0;          // OPTION_REG<3> Prescaler Assignment bit
                  // 1 = Prescaler is assigned to the WDT
                  // 0 = Prescaler is assigned to the Timer0
module
      PS2 = 0;
      PS1 = 0;
      PS0 = 0;          // OPTION_REG<2:0> Prescaler Rate Select bits
111=/256

      TMR0 = 0xFF;            // re-load
      TMR0 -= 100;            // re-load
}

void iniUsart(void)           //Async Baud Rate 8 bits, No Parity, 1
Stop Bit
{
      SPBRG = 0x0C;           // 9600=0x19=25 ,19200=0x0C=12
      TXSTA = 0b00100100;     // TXSTA<5> = TX_Enable ,TXSTA<4> BRGH =
high
      RCSTA = 0b10010000;     // RCSTA<7> = SPEN EnableUSART ,RCSTA<4>
= enable continuous mode
}

void UsartTx(unsigned char TxValue) // Rs485 Transmit Value(Command)
{
      TxEn = 1;         // Enable RTS To Send

      TXREG = TxValue;
      while(!TRMT);           //    TXSTA,TRMT(1)    transmission    is
complete if hi

      TxEn = 0;         // Disable RTS To Send
}

void UsartTx_INT(unsigned char TxValue)   //      Rs485      Transmit
Value(Command)
{
      TxEn = 1;         // Enable RTS To Send

      TXREG = TxValue;
      while(!TRMT);           //    TXSTA,TRMT(1)    transmission    is
complete if hi

      TxEn = 0;         // Disable RTS To Send
}

void iniInterupt(void)
{
      //PIE1 = 0b00100000;
      //TMR1IE = 0;           //   PIE1<0>   TMR1   Overflow   Interrupt
Enable bit
      //TMR2IE = 0;           // PIE1<1> TMR2 to PR2 Match Interrupt
Enable bit
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
//CCP1IE = 0;           // PIE1<2> CCP1 Interrupt Enable bit
      //SSPIE = 0;            //  PIE1<3>  SynchroReadus  Serial  Port
Interrupt Enable bit
      //TXIE = 0;       // PIE1<4> USART Transmit Interrupt Enable bit
      RCIE = 1;         // PIE1<5> USART Receive Interrupt Enable bit
      //ADIE = 0;       // PIE1<6> A/D Converter Interrupt Enable bit
      //PSPIE = 0;            //   PIE1<7>   Parallel   Slave   Port
Read/Write Interrupt Enable bit

      //INTEDG = 0;           // OPTION_REG<6> RB0/INT Interrupt Edge
Select bit

      //INTCON = 0b11010000;
      //RBIF = 0;       //  INTCON<0>  RB  Port  Change  Interrupt  Flag
bit(Clear in software)
      //INTF = 0;       // INTCON<1> RB0/INT External Interrupt Flag
bit(Clear in software)
      //T0IF = 0;       //  INTCON<2>  TMR0  Overflow  Interrupt  Flag
bit(Clear in software)
      //RBIE = 0;       // INTCON<3> RB Port Change Interrupt Enable
bit
      //INTE = 0;       //  INTCON<4>  RB0/INT  Enable  the  external
interrupt
//    T0IE = 1;         //  INTCON<5>  TMR0  Overflow  Interrupt  Enable
bit
      PEIE = 1;         // INTCON<6> Peripheral Interrupt Enable bit
      GIE = 1;          // INTCON<7> Global Interrupt Enable bit
}

static void interrupt isr(void)             //  Here  be  interrupt
function - the name is unimportant.
{
      if(RCIF)                      // PIR1<5> USART Receive Interrupt
Flag bit (1 buff. full)
      {
            RxBuffer = RCREG;

            RxTemp1 = RxTemp0;      // Store Data
            RxTemp0 = RxBuffer;

            if(RxTemp0 == 0x20)
            {
                  if(RxTemp1 == 't')      //  Send  Data  Temperature
from DS1820
                  {
                        PrintThermo();          //      Send      Value
232/RS485
                  }
            }
      }
}

void ReadThermometer(void)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
{
unsigned char temp;

      ReadStatus = 1;   // in Read Process

      if(init1820())          // DS1820 Connect
      {
            write1820(0xCC);  // ' Issue Skip ROM command
            write1820(0x44);  // ' Start temperature conversion
            Delay_100Ms(10);        // Pause 2000 ' Wait 2 seconds for
conversion to complete

            if(init1820())          // ' Do another init
            {
                  DQ_LED = 1; // Show DS1820 Connect & Read

                  write1820(0xCC);  // ' Issue Skip ROM command
                  write1820(0xBE);  // ' Read the temperature
                  temp = read1820();

                  ConvertThermo(temp);    //    Convert    Value    To
Thermoxxx Buffer

                  DQ_LED = 0; // Clear Show DS1820 Connect & Read

                  ReadError = ' ';  // Read Complete
            }
            else
            {
                  ReadError = 'e';  // Read Error
                  //UsartTx('n');   // Send Value Not Present
                  //UsartTx(' ');   // Send Value
            }
      }
      else
      {
            ReadError = 'e';  // Read Error
            //UsartTx('n');   // Send Value Not Present
            //UsartTx(' ');   // Send Value
      }

      ReadStatus = 0;   // Out Read Process
}

// ' Initialize DS1820 and check for presence
unsigned int init1820(void)
{
      DQ = 0;           // Low DQ ' Set the data pin low to init
      DQ_DIR = 0; // momentary low
      DelayUs(500);     // Pauseus 500 ' Wait > 480us
      DQ_DIR = 1;      // DQ_DIR = 1 ' Release data pin (set to input
for high)
DelayUs(100);     // Pauseus 100 ' Wait > 60us
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
      if(DQ = 1)  // if DQ = 1 Then
      {           // Lcdout $fe, 1, "DS1820 not present"
            Delay_100Ms(5);   // Pause 500 "0.5Sec"
            return 0;   // ' Try again
      }
      DelayUs(400)      // Pauseus 400 ' Wait for end of presence
pulse
        return(1);
}

// ' Write "command" byte to the DS1820
void write1820(unsigned char cmd)
{
      unsigned int i;

      for(i=0;i<8;i++)  // ' 8 bits to a byte
      {
            if(cmd & 0x01)          // if command.0 = 1 Then
            {
                  DQ = 0;           // Gosub write0 ' Write a 0 bit
                  DQ_DIR = 0; // momentary low
                  asm("NOP"); // ' Delay 1us at 4MHz
                  DQ_DIR = 1; // ' Release data pin (set to input for
high)
                  DelayUs(60);      // ' Low for > 60us for 0
            }
            else                    // Gosub write0 ' Write a 0 bit
            {
                  DQ = 0;           // ' Low for < 15us for 1
                  DQ_DIR = 0; // momentary low
                  DelayUs(60);      // ' Low for > 60us for 0
                  DQ_DIR = 1; // ' Release data pin (set to input for
high)
            }
            cmd = cmd >> 1;   // ' Shift to next bit
      }
}

// ' Read temperature from the DS1820
unsigned char read1820(void)
{
      unsigned int i;
      unsigned char thermo;

      thermo = 0;       // Clear

      for(i=0;i<8;i++)  // ' 8 bits to a byte
      {
            DQ = 0;           // Gosub write0 ' Write a 0 bit
            DQ_DIR = 0; // momentary low
asm("NOP"); // ' Delay 1us at 4MHz
            DQ_DIR = 1; // ' Release data pin (set to input for high)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            if(DQ = 0)
            {
                  thermo = ((thermo >> 1) | 0x80);    // ' Set bit to
1
            }
            else
            {
                  thermo = (thermo >> 1);             // ' Set bit to
0
            }

            DelayUs(60);      // ' Low for > 60us for 0
      }
      return(thermo);
}

// ' Convert to thermoxxx
void ConvertThermo(unsigned char tx)
{
      thermo100 = 0;          // Clear Unit 100
      thermo10 = 0;           // Clear Unit 10
      thermo1 = 0;            // Clear Unit 1
      thermo0 = 0;            // Clear Unit 0

      thermo0 = ((tx & 0x01) * 5);        // Convertion

      tx = (tx / 2);
      if(tx > 99)
      {
            thermo100 = (tx / 100); // Store to thermo100
            tx -= (thermo100*100);  // delete digit100
      }
      if(tx >9)
      {
            thermo10 = (tx / 10);   // Store to thermo10
            tx -= (thermo10*10);          // delete digit10
      }
      thermo1 = tx;                       // Store to thermo1
}

// ' Numerical to Character
int num_to_char(int val)      // converts val to hex character
{
   int ch;
   if (val < 10)
   {
     ch=val+'0';
   }
   else
   {
     val=val-10;
     ch=val + 'A';
   }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
   return(ch);
}

// ' Display Value
void PrintThermo(void)
{
      UsartTx_INT(num_to_char(thermo100));
      UsartTx_INT(num_to_char(thermo10));
      UsartTx_INT(num_to_char(thermo1));
      UsartTx_INT('.');
      UsartTx_INT(num_to_char(thermo0));
      UsartTx_INT(ReadError);
}
```

โปรแกรม PWM ใช้ควบคุมมอเตอร์และฮีตเตอร์

```
#include <pic.h>
#include <stdlib.h>

#define XTAL_FREQ 4MHZ

#include "delay.h"
#include "delay.c"

/*--PORT-STRUCTURE-------------------------------------------------------
--------*/
//P16F877
//RC7 USART_Rx    I/P
//RC6 USART_Tx    O/P
//RC5 USART_TxEN  O/P
//RD0 Heater1          O/P
//RD1 Fan1        O/P
//RD2 Fan2        O/P
//RD3 Fan3        O/P

/*--PORT-----------------------------------------------------------------
--------*/
#define     PIN(xname,xport,xbit)          static     bit     xname     @
((unsigned)&xport*8+xbit)
PIN (Heater1, PORTD, 0);       // Heater
PIN (Fan1, PORTD, 1);          // Fan1
PIN (Fan2, PORTD, 2);          // Fan2
PIN (Fan3, PORTD, 3);          // Fan3

PIN (TxEn, PORTC, 5);          // TxEnable RS485

/*--ASCIIConvert---------------------------------------------------------
---------*/
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
#define NumFilter 0b00001111  // AND to use (30-39H) -> (0-9D)
#define NumConvert      0b00110000  // OR to use (0-9D) -> (30-39H)

/*--USART-------------------------------------------------------------
--------*/
static unsigned char    RxBuffer,   // Usart Receive buffer
                  TxBuffer;   // Usart Transmit buffer

static unsigned char    RxTemp0,    // Usart Receive buffer
                  RxTemp1,    // Usart Receive buffer
                  RxTemp2,    // Usart Receive buffer
                  RxTemp3,    // Usart Receive buffer
                  RxTemp4,    // Usart Receive buffer
                  RxTemp5;    // Usart Receive buffer

/*--BUFFER------------------------------------------------------------
---------*/
static unsigned int     CountBuffer,          // Count PWM Buffer
                  Heater1Buffer,        // Heater1 PWM Buffer
                  Fan1Buffer,       // Fan1 PWM Buffer
                  Fan2Buffer,       // Fan2 PWM Buffer
                  Fan3Buffer;       // Fan3 PWM Buffer

/*--------------------------------------------------------------------
---------*/

// ' Delay
void Delay_100Ms(unsigned int dx);

// ' Variable
void iniVariable(void);

// ' iniPort
void iniPort(void);

// ' iniTimer
void iniTimer0(void);

// ' Usart
void iniUsart(void);
void UsartTx(unsigned char TxValue);

// ' iniInterupt
void iniInterupt(void);
static void interrupt isr(void);

// ' SumData
unsigned int SumData(void);


/*--------------------------------------------------------------------
---------*/
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
// ' main()
void main(void)
{
      iniPort();  // initial port
      iniTimer0();      // initial Timer0
      iniUsart(); // initial Usart
      iniVariable();    // initial Variable
      iniInterupt();    // initial Intrupt

      while(1)
      {
            CLRWDT();   // Idly kick the dog
            // None
      }
}

void Delay_100Ms(unsigned int dx)
{
      do
      {
            DelayMs(100);
      } while(--dx) ;
}

void iniVariable(void)
{
      CountBuffer = 100;          // Count Buffer
      Heater1Buffer = 0;          // Heater1 OFF 0%
      Fan1Buffer = 0;             // Fan1 OFF 0%
      Fan2Buffer = 0;             // Fan1 OFF 0%
      Fan3Buffer = 0;             // Fan1 OFF 0%
}

void iniPort(void)
{
      PORTA = 0x00;               // turn Off all
      PORTB = 0x00;               // turn Off all
      PORTC = 0x00;               // turn Off all
      PORTD = 0x00;               // turn Off all
      PORTE = 0x00;               // turn Off all

      TRISA = 0xFF;               // 1=Input , 0=Output
      ADCON1 = 7;         // PORTA all Set to Digital
      //CMCON = 0x07;              // (P16F628) PORTA all Set to Digital
      TRISB = 0x03;               // RB<7:2>=Output ,RB<1:0>=Input
      RBPU = 1;          //  OPTION_REG<6>  (ActiveLow)  PORTB  Pull-up
Enable bit
      TRISC = 0b10000000;         // all bits output
      TRISD = 0x00;               // all bits output
      TRISE = 0;          // all bits output
}

void iniTimer0(void)              // Initial (TMR0) Timer0
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
{
      T0CS = 0;          // OPTION_REG<5> TMR0 Clock Source Select bit
                         // 1 = Transition on RA4/T0CKI pin
                         //  0  =  Internal  instruction  cycle  clock
(CLKOUT)
      T0SE = 1;          // OPTION_REG<4> TMR0 Source Edge Select bit
(Use in Ext.Clock RA4/T0CLK)
                         // 1 = Increment on high-to-low transition on
RA4/T0CKI pin
                         // 0 = Increment on low-to-high transition on
RA4/T0CKI pin
      PSA = 0;           // OPTION_REG<3> Prescaler Assignment bit
                         // 1 = Prescaler is assigned to the WDT
                         // 0 = Prescaler is assigned to the Timer0
module
      PS2 = 0;
      PS1 = 0;
      PS0 = 0;           // OPTION_REG<2:0> Prescaler Rate Select bits
111=/256

      TMR0 = 0xFF;             // re-load
      TMR0 -= 50;        // re-load
}

void iniUsart(void)              //Async Baud Rate 8 bits, No Parity, 1
Stop Bit
{
      SPBRG = 0x0C;              // 9600=0x19=25 ,19200=0x0C=12
      TXSTA = 0b00100100;        // TXSTA<5> = TX_Enable ,TXSTA<4> BRGH =
high
      RCSTA = 0b10010000;        // RCSTA<7> = SPEN EnableUSART ,RCSTA<4>
= enable continuous mode
}

void UsartTx(unsigned char TxValue) // Rs485 Transmit Value(Command)
{
      TxEn = 1;          // Enable RTS To Send

      TXREG = TxValue;
      while(!TRMT);              //    TXSTA,TRMT(1)    transmission    is
complete if hi

      TxEn = 0;          // Disable RTS To Send
}

void iniInterupt(void)
{
      //PIE1 = 0b00100000;
      //TMR1IE = 0;              //   PIE1<0>   TMR1   Overflow   Interrupt
Enable bit
      //TMR2IE = 0;              // PIE1<1> TMR2 to PR2 Match Interrupt
Enable bit
      //CCP1IE = 0;              // PIE1<2> CCP1 Interrupt Enable bit
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
      //SSPIE = 0;            //  PIE1<3>  SynchroReadus  Serial  Port
Interrupt Enable bit
      //TXIE = 0;       // PIE1<4> USART Transmit Interrupt Enable bit
      RCIE = 1;         // PIE1<5> USART Receive Interrupt Enable bit
      //ADIE = 0;       // PIE1<6> A/D Converter Interrupt Enable bit
      //PSPIE = 0;            //    PIE1<7>    Parallel    Slave    Port
Read/Write Interrupt Enable bit

      //INTEDG = 0;           // OPTION_REG<6> RB0/INT Interrupt Edge
Select bit

      //INTCON = 0b11010000;
      //RBIF = 0;       //  INTCON<0>  RB  Port  Change  Interrupt  Flag
bit(Clear in software)
      //INTF = 0;       //  INTCON<1>  RB0/INT  External  Interrupt  Flag
bit(Clear in software)
      //T0IF = 0;       //  INTCON<2>  TMR0  Overflow  Interrupt  Flag
bit(Clear in software)
      //RBIE = 0;       //  INTCON<3>  RB  Port  Change  Interrupt  Enable
bit
//INTE = 0;       // INTCON<4> RB0/INT Enable the external interrupt
      T0IE = 1;       //  INTCON<5>  TMR0  Overflow  Interrupt  Enable
bit
      PEIE = 1;       // INTCON<6> Peripheral Interrupt Enable bit
      GIE = 1;        // INTCON<7> Global Interrupt Enable bit
}

static void interrupt isr(void)                 // Here be interrupt
function - the name is unimportant.
{
      if(T0IF == 1)                       //  INCON<2>  TMR0  Overflow
Interrupt Flag bit(Clear in software)
      {
            if((CountBuffer) < 100)       //   is   the   Delay   timer
running?
            {
                  if(Heater1Buffer > CountBuffer)     //  Heater1  PWM
ON
                        Heater1 = 1;
                  else
                        Heater1 = 0;

                  if(Fan1Buffer > CountBuffer)  // Fan1 PWM ON
                        Fan1 = 1;
                  else
                        Fan1 = 0;

                  if(Fan2Buffer > CountBuffer)  // Fan2 PWM ON
                        Fan2 = 1;
                  else
                        Fan2 = 0;

                  if(Fan3Buffer > CountBuffer)  // Fan3 PWM ON
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                        Fan3 = 1;
                  else
                        Fan3 = 0;

                  CountBuffer++;
            }
            else
            {
                  CountBuffer = 0;  // Clear Count Buffer For PWM
            }

            TMR0 -= 50;                     // clear the timer0
            T0IF = 0;                 // clear the interrupt flag
      }

      //UsartTx('a');

      if(RCIF)                        // PIR1<5> USART Receive Interrupt
Flag bit (1 buff. full)
      {

            RxBuffer = RCREG;

            RxTemp5 = RxTemp4;              // Store Data
            RxTemp4 = RxTemp3;
            RxTemp3 = RxTemp2;
            RxTemp2 = RxTemp1;
            RxTemp1 = RxTemp0;
            RxTemp0 = RxBuffer;

            if(RxTemp0 == 0x20)
            {
                  switch(RxTemp2)
                  {
                        case 'h': Heater1Buffer = SumData(); break; //
Case Data for Heater
                        case 'f':   // Case Data for Fan
                        {
                              switch(RxTemp1)
                              {
                                    case '1': Fan1Buffer = SumData();
break;
                                    case '2': Fan2Buffer = SumData();
break;
                                    case '3': Fan3Buffer = SumData();
break;
                                    default : /*None*/;
                              }
                        }
                        default : /*None*/;
                  }
            }
      }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
}

unsigned int SumData(void)    // SumData
{
      unsigned int SumTemp;

      SumTemp = (RxTemp3 & NumFilter);
      SumTemp += ((RxTemp4 & NumFilter) * 10);
      SumTemp += ((RxTemp5 & NumFilter) * 100);

      return SumTemp;
}
```



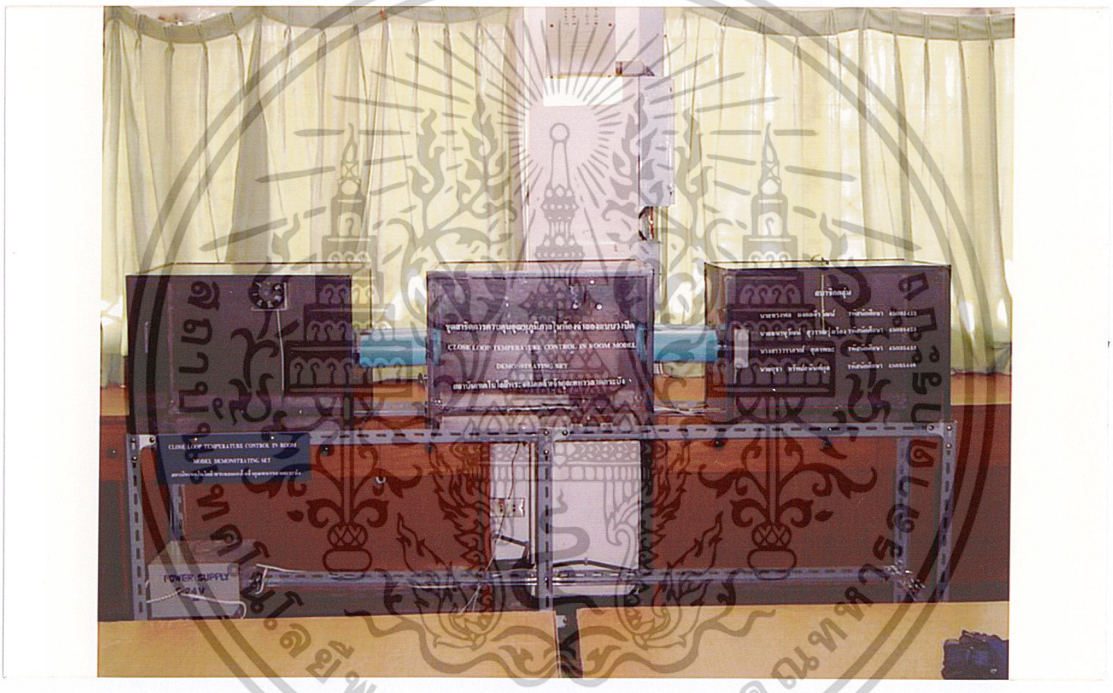
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาคผนวก จ
คู่มือการใช้งาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คู่มือการใช้งาน
ชุดสาธิตการควบคุมอุณหภูมิภายในห้องจำลองแบบวงปิด



ภาควิชาครุศาสตร์วิศวกรรม

คณะครุศาสตร์อุตสาหกรรม

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2546

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

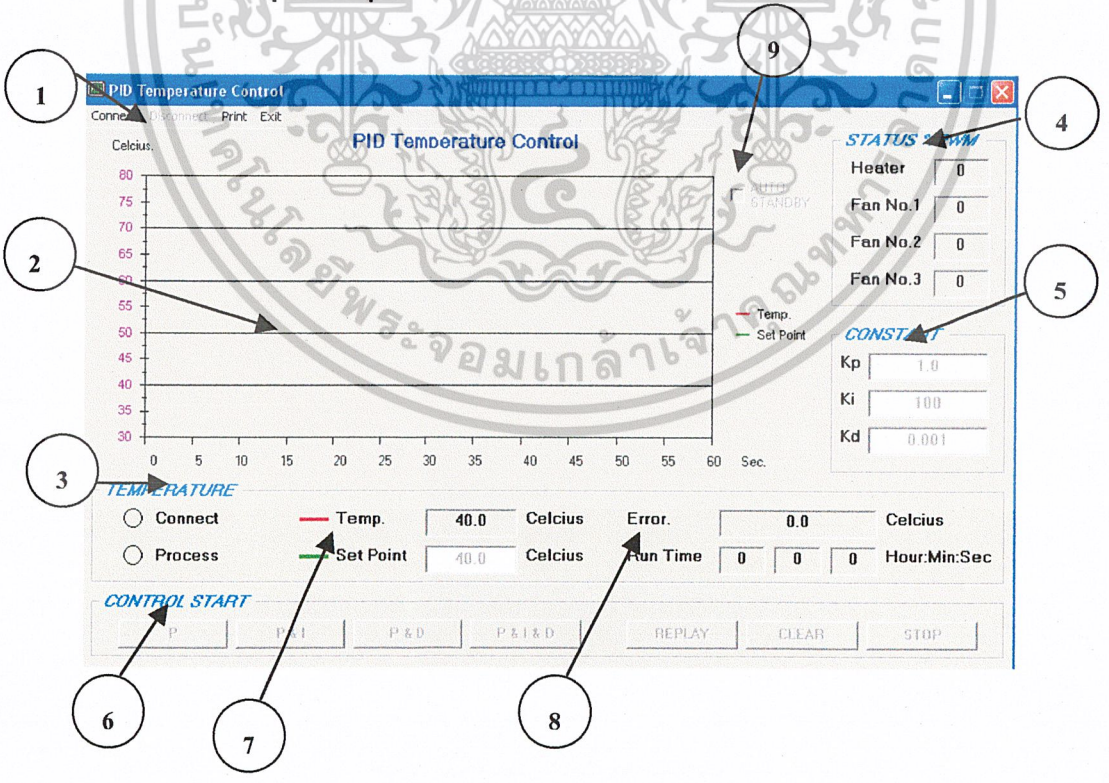
วัตถุประสงค์การทดลอง

1. เพื่อให้นักศึกษาหาค่าคุณลักษณะของโปรเซสจากรูปกราฟผลการทดลองได้
2. บอกความแตกต่างของผลตอบสนองต่ออินพุตแบบขั้นบันไดเมื่อระบบมีลำดับเพิ่มมากขึ้น
3. บอกความหมายของคำว่า Process gain ได้
4. บอกความหมายของคำว่า Process Time Constant ได้
5. บอกความหมายของคำว่า Process Dead Time ได้
6. ต่อวงจรได้ถูกต้องปลอดภัย

1. คำแนะนำเบื้องต้น

ก่อนจะใช้งานชุดสาธิตการควบคุมอุณหภูมิในห้องจำลองแบบวงปิด ควรศึกษาคู่มือการใช้งานให้ถูกต้องเสียก่อน และเป็นการป้องกันการเสียหาย

2. ส่วนประกอบและปุ่มควบคุม



รูปที่ จ.1 ส่วนควบคุมและแสดงผล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ จ.1 มีรายละเอียดต่างๆ ดังนี้

หมายเลข 1 Connect

หมายเลข 2 กราฟแสดงการทำงานเพื่อเข้าสู่ค่าเป้าหมาย

หมายเลข 3 ส่วนแสดงค่าอุณหภูมิ

หมายเลข 4 ส่วนการแสดงผลของ PWM

หมายเลข 5 ส่วนของการป้อนค่ากิริยาควบคุม

หมายเลข 6 ส่วนกิริยาที่ต้องการควบคุมและส่วนในการควบคุม

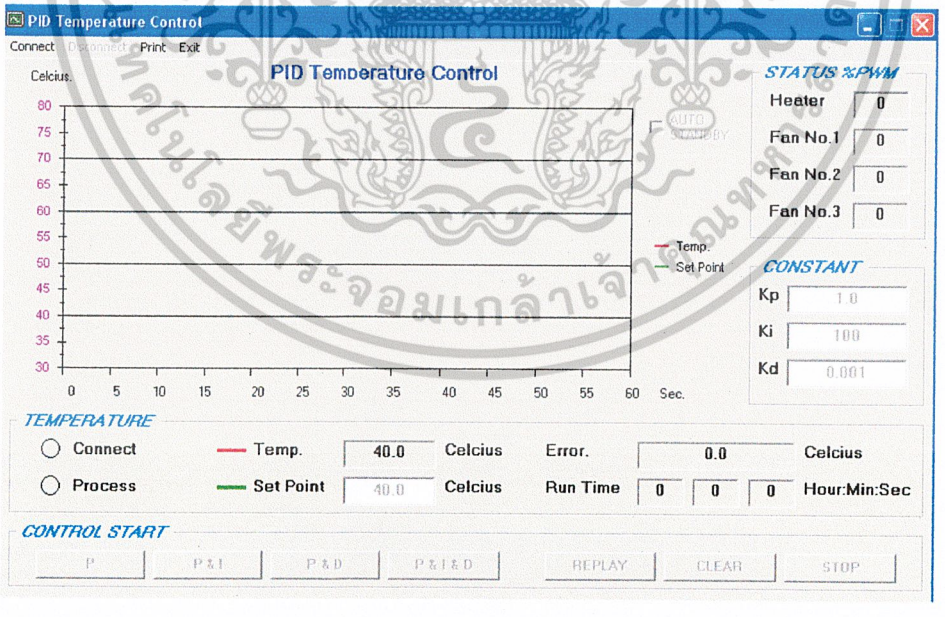
หมายเลข 7 แสดงค่าอุณหภูมิเริ่มต้นกับอุณหภูมิที่ต้องการ

หมายเลข 8 ส่วนการแสดงค่าความผิดพลาดและเวลาในการใช้งาน

หมายเลข 9 ปุ่ม AUTO STANDBY

3. การใช้งาน

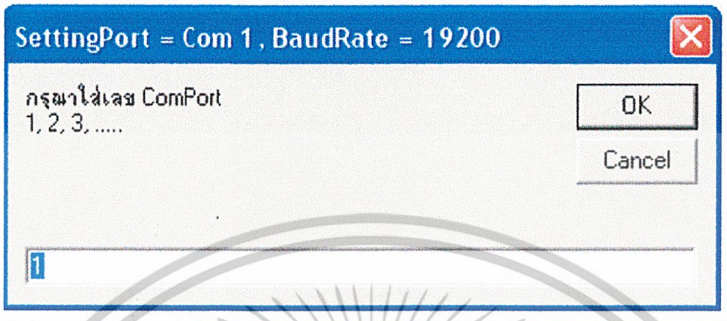
1. เริ่มต้นการใช้งาน คลิกที่ไอคอน VB Thermo เพื่อเปิดหน้าต่าง PID Themperature Control จะเห็นหน้าต่างดังรูปที่ จ.2



รูปที่ จ.2 หน้าต่าง PID Themperature Control

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

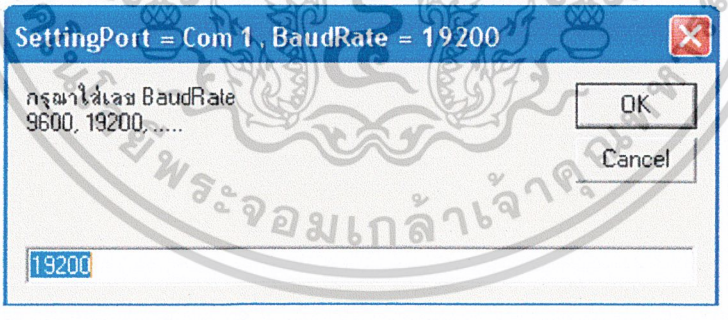
2. เปิดสวิตช์สตาร์ทชุดควบคุม จากนั้นไปที่คอมพิวเตอร์ คลิก Connect ที่หน้าต่าง PID Themperature Control ที่หน้าจอจะแสดงดังรูปที่ จ.3



รูปที่ จ.3 การเลือก Com Port

3. ทำการเลือก Com Port โดยการป้อนตัวเลข Com Port เพื่อส่งผ่านข้อมูลแบบอนุกรมให้กับพอร์ตที่เราต้องการติดต่อ ในที่นี้เราเลือกพอร์ตที่ต้องการติดต่อที่ COM 1 หลังจากนั้นคลิก OK

4. หน้าจอจะแสดงหน้าต่างดังรูปที่ จ.4 เพื่อให้เรา เซตค่า Baud Rate ในที่นี้ตั้งค่าที่ 19200 หลังจากนั้นคลิก OK

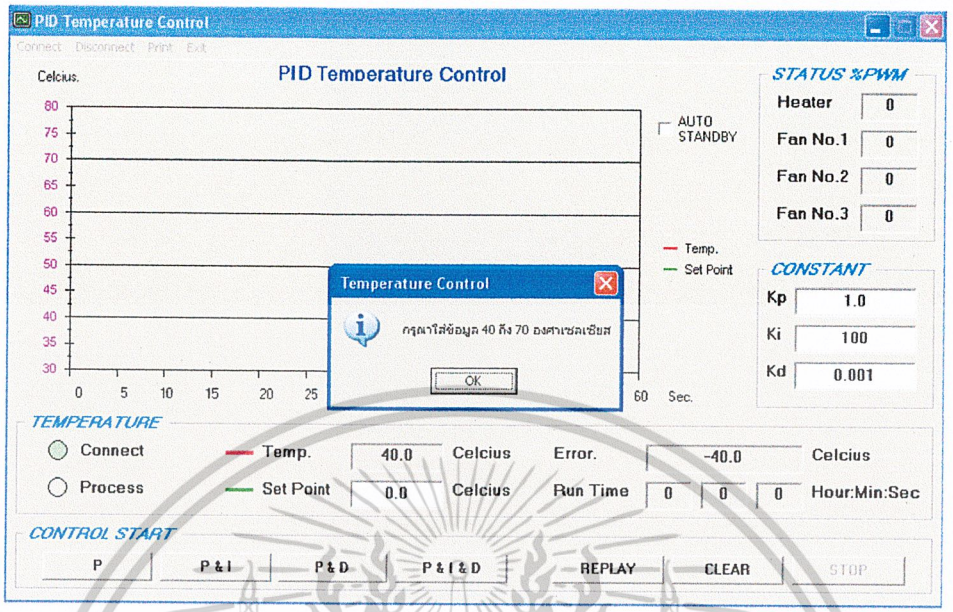


รูปที่ จ.4 การตั้งค่า Baud Rate

5. คลิกปุ่ม AUTO STANDBY ให้ STATUS % PWM ทำงานและแสดงผล

6. หน้าต่าง PID Themperature Control พร้อมที่จะรับข้อมูลในการควบคุมควรตั้งค่าเป้าหมายให้อยู่ในช่วง 40-70 องศาเซลเซียส ถ้าอุณหภูมิน้อยกว่า 40 องศาเซลเซียสหรือมากกว่า 70 องศาเซลเซียส จะแสดงค่าพิดพลาดดังรูปที่ จ.5

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ จ.5 การตั้งค่าเป้าหมายสูงหรือต่ำกว่าย่านที่กำหนด

7. ตั้งค่ากิริยาในการควบคุมที่เราต้องการ คือ Kp, Ki หรือ Kd
8. เลือกปุ่มกิริยาการควบคุมที่ต้องการคือ P, PI, PD หรือ PID
9. รอดูผลการทำงานได้จากกราฟที่หน้าต่าง PID Themperature Control
10. เมื่อต้องการดูผลการทำงานซ้ำให้คลิกที่ปุ่ม REPLAY
11. เมื่อต้องการหยุดการทำงานให้คลิกที่ปุ่ม STOP
12. เมื่อต้องการยกเลิกโปรแกรมให้คลิกที่ปุ่ม CLEAR

4. การดูแลรักษาและข้อควรระวัง

3.1 การดูแลรักษา

- เช็ดทำความสะอาดตัวเครื่องด้วยผ้านุ่ม อย่าใช้สารใดๆ ที่เป็นตัวทำละลายเพราะอาจทำให้ตัวเครื่องเป็นรอยเสียหายได้

- ไม่ควรให้ตู้เกิดการกระแทกเพราะจะทำให้ตู้เสียหายได้

3.2 ข้อควรระวัง

- ในการเคลื่อนย้ายทุกครั้งควรทำการถอดแยกท่อส่งลมร้อนออกก่อน เพราะอาจเกิดการเสียหายได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ขณะที่เครื่องกำลังสร้างอุณหภูมิอยู่ไม่ควรปรับบอลวาล์วเพื่อหรี่ลมหรือปิดท่อส่งลม เพราะจะทำให้ฮีตเตอร์สร้างควรร้อนสูงจนห้องสร้างอุณหภูมิเกิดการเสียหายได้

- ปิดสวิตซ์ และถอดปลั๊กออกทุกครั้งที่เลิกใช้งาน

5. ข้อมูลจำเพาะ

ตารางที่ จ.1 ข้อมูลจำเพาะของชุดสาธิตการควบคุมอุณหภูมิภายในวงปิด

คุณสมบัติ	รายละเอียด
หลักการวัด	จะวัดอุณหภูมิภายในห้องที่ 3 แล้วส่งค่ามายัง PIC 16F877 แล้วแสดงผลที่หน้าจอคอมพิวเตอร์
ย่านการวัด	40°C - 70°C
ส่วนแสดงผล	แสดงผลที่จอคอมพิวเตอร์
ความเที่ยงตรง	(สำหรับการอ่านค่าแบบเต็มสเกล) 40°C ผิดพลาด ±2.0°C 50°C ผิดพลาด ±2.5°C 60°C ผิดพลาด ±3.0°C 70°C ผิดพลาด ±3.5°C
ความผิดพลาด	±0.5 %
แหล่งจ่ายพลังงาน	1. ไฟฟ้ากระแสตรง 24 V 8 A 2. ไฟฟ้ากระแสตรง 24 V 30 A

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาคผนวก ฉ

รายระเอียดและคุณสมบัติของอุปกรณ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

MICROCHIP

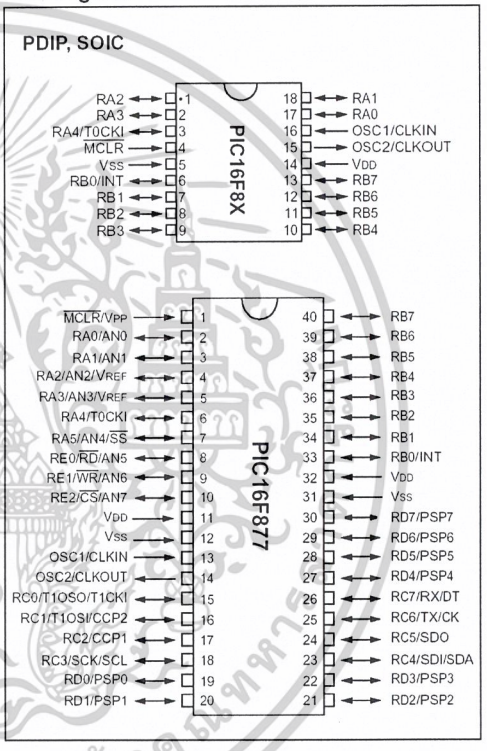
PIC16F8XX

EEPROM Memory Programming Specification

This document includes the programming specifications for the following devices:

- PIC16F83
- PIC16CR83
- PIC16F84
- PIC16CR84
- PIC16F84A
- PIC16F877

Pin Diagram



1.0 PROGRAMMING THE PIC16F8X

The PIC16F8X is programmed using a serial method. The serial mode will allow the PIC16F8X to be programmed while in the users system. This allows for increased design flexibility. This programming specification applies to PIC16F8X devices in all packages.

1.1 Hardware Requirements

The PIC16F8X requires one programmable power supply for VDD (4.5V to 5.5V) and a VPP of 12V to 14V. Both supplies should have a minimum resolution of 0.25V.

1.2 Programming Mode

The programming mode for the PIC16F8X allows programming of user program memory, data memory, special locations used for ID, and the configuration word.

PIN DESCRIPTIONS (DURING PROGRAMMING): PIC16F8X

Pin Name	During Programming		
	Function	Pin Type	Pin Description
RB6	CLOCK	I	Clock input
RB7	DATA	I/O	Data input/output
$\overline{MCLR}$	VTEST MODE	P*	Program Mode Select
VDD	VDD	P	Power Supply
VSS	VSS	P	Ground

Legend: I = Input, O = Output, P = Power

*In the PIC16F8X, the programming high voltage is internally generated. To activate the programming mode, high voltage needs to be applied to $\overline{MCLR}$ input. Since the $\overline{MCLR}$ is used for a level source, this means that $\overline{MCLR}$ does not draw any significant current.

© 1998 Microchip Technology Inc.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

FIGURE 5-1: LOAD DATA COMMAND (PROGRAM/VERIFY)

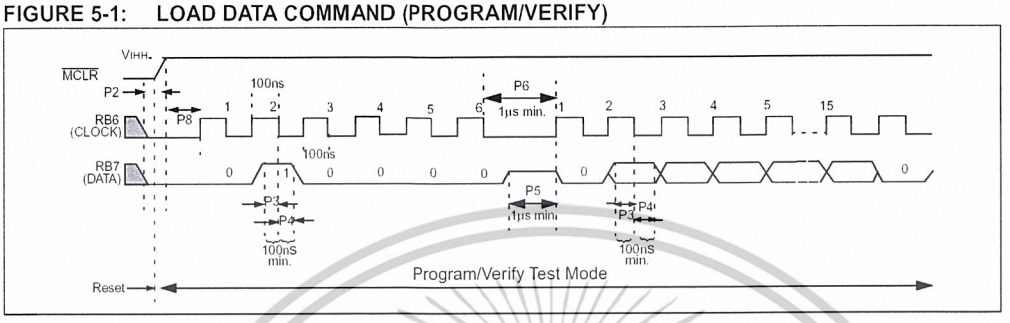


FIGURE 5-2: READ DATA COMMAND (PROGRAM/VERIFY)

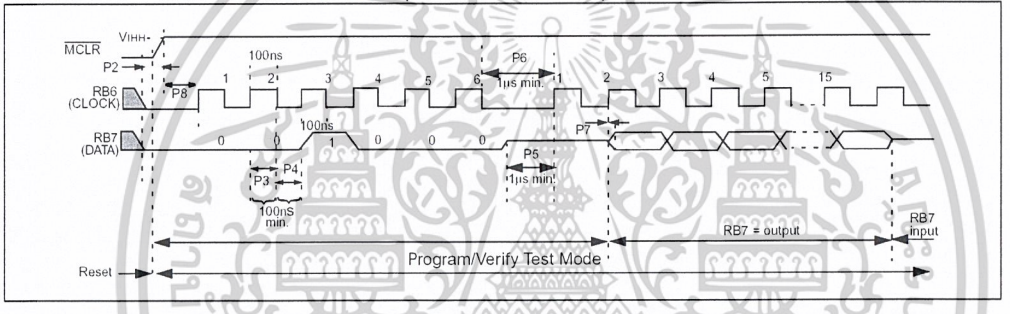
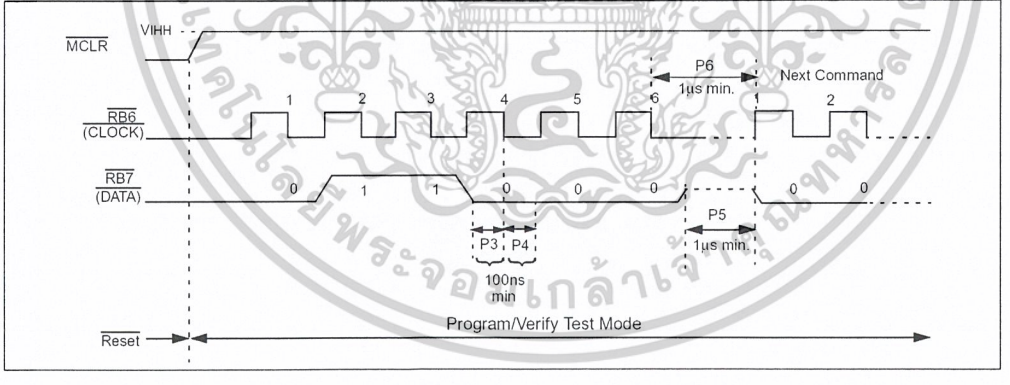


FIGURE 5-3: INCREMENT ADDRESS COMMAND (PROGRAM/VERIFY)



© 1998 Microchip Technology Inc.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

PIC16F8XX

3.0 CONFIGURATION WORD

The PIC16F8X has five configuration bits. These bits can be set (reads '0') or left unchanged (reads '1') to select various device configurations.

3.1 Device ID Word

The device ID word for the PIC16F8XX is located at 2006h.

TABLE 3-1:

Device	Device ID Value	
	Dev	Rev
PIC16F84A	00 0101 010	0 0000
PIC16F877	00 1001 101	0 0000

FIGURE 3-1: CONFIGURATION WORD BIT MAP FOR PIC16F83/CR83/F84/CR84/F84A

Bit Number:	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PIC16F83/ F84/F84A	CP	CP	CP	CP	CP	CP	CP	CP	CP	CP	$\overline{\text{PWRTE}}$	WDTE	FOSC1	FOSC0
PIC16CR83/ CR84	CP	CP	CP	CP	CP	CP	DP	CP	CP	CP	$\overline{\text{PWRTE}}$	WDTE	FOSC1	FOSC0

bit 4-13: **CP**, Code Protection Configuration Bits
1 = code protection off
0 = code protection on

bit 7: **PIC16CR83/CR84 only**
DP, Data Memory Code Protection Bit
1 = code protection off
0 = data memory is code protected

bit 3: **$\overline{\text{PWRTE}}$**, Power Up Timer Enable Configuration Bit
1 = Power up timer disabled
0 = Power up timer enabled

bit 2: **$\overline{\text{WDTE}}$**, WDT Enable Configuration Bits
1 = WDT enabled
0 = WDT disabled

bit 1-0 **FOSC<1:0>**, Oscillator Selection Configuration Bits
11: RC oscillator
10: HS oscillator
01: XT oscillator
00: LP oscillator

© 1998 Microchip Technology Inc.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

MICROCHIP

PIC16F87X

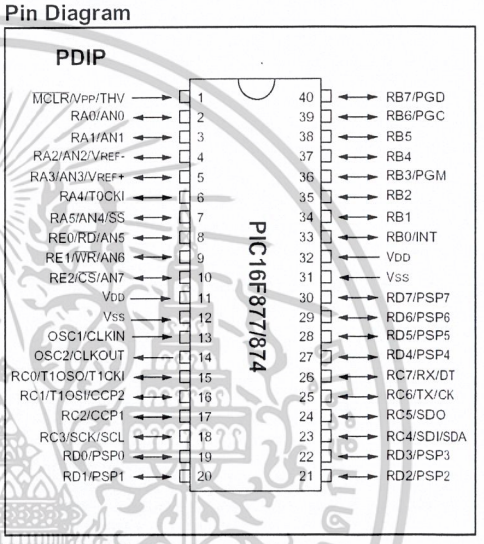
28/40-pin 8-Bit CMOS FLASH Microcontrollers

Microcontroller Core Features:

- High-performance RISC CPU
- Only 35 single word instructions to learn
- All single cycle instructions except for program branches which are two cycle
- Operating speed: DC - 20 MHz clock input
  DC - 200 ns instruction cycle
- Up to 8K x 14 words of FLASH Program Memory,
  Up to 368 x 8 bytes of Data Memory (RAM)
  Up to 256 x 8 bytes of EEPROM data memory
- ★ **Pinout compatible to the PIC16C73/74/76/77**
- Interrupt capability (up to 14 internal/external interrupt sources)
- Eight level deep hardware stack
- Direct, indirect, and relative addressing modes
- Power-on Reset (POR)
- Power-up Timer (PWRT) and Oscillator Start-up Timer (OST)
- Watchdog Timer (WDT) with its own on-chip RC oscillator for reliable operation
- Programmable code-protection
- Power saving SLEEP mode
- Selectable oscillator options
- Low-power, high-speed CMOS FLASH/EEPROM technology
- Fully static design
- In-Circuit Serial Programming™ via two pins
- ★ Only single 5V source needed for programming
- ★ In-Circuit Debugging via two pins
- Processor read/write access to program memory
- Wide operating voltage range: 2.0V to 5.5V
- High Sink/Source Current: 25 mA
- Commercial and Industrial temperature ranges
- Low-power consumption:
  - < 2 mA typical @ 5V, 4 MHz
  - 20 μA typical @ 3V, 32 kHz
  - < 1 μA typical standby current

Pin Diagram

PDIP



Peripheral Features:

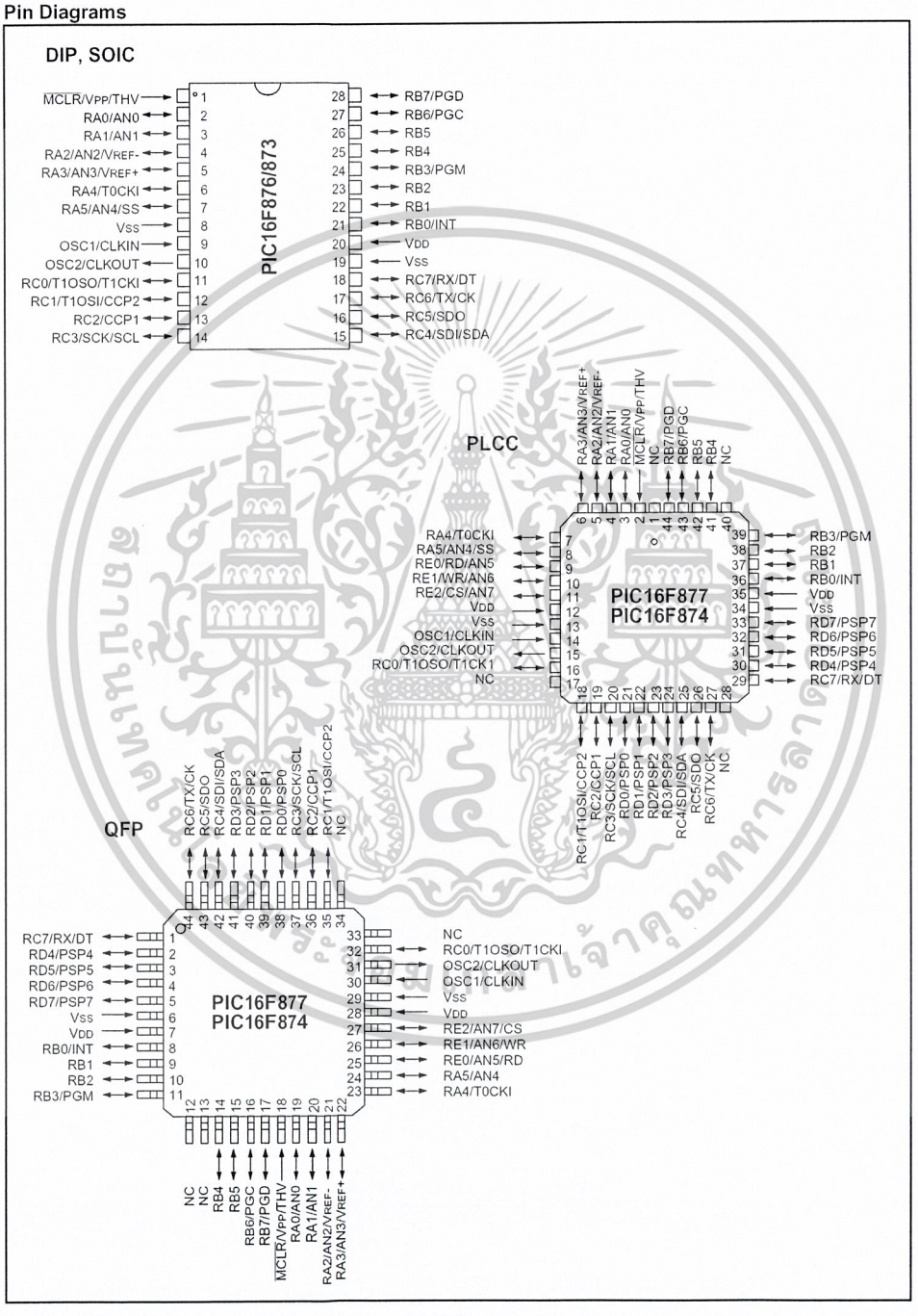
- Timer0: 8-bit timer/counter with 8-bit prescaler
- Timer1: 16-bit timer/counter with prescaler, can be incremented during sleep via external crystal/clock
- Timer2: 8-bit timer/counter with 8-bit period register, prescaler and postscaler
- Two Capture, Compare, PWM modules
- Capture is 16-bit, max. resolution is 12.5 ns, Compare is 16-bit, max. resolution is 200 ns, PWM max. resolution is 10-bit
- ★ 10-bit multi-channel Analog-to-Digital converter
- ★ Synchronous Serial Port (SSP) with SPI™ (Master Mode) and I²C™ (Master/Slave)
- ★ Universal Synchronous Asynchronous Receiver Transmitter (USART/SCI) with 9-bit address detection
- Parallel Slave Port (PSP) 8-bits wide, with external $\overline{RD}$, $\overline{WR}$ and $\overline{CS}$ controls (40/44-pin only)
- Brown-out detection circuitry for Brown-out Reset (BOR)

© 1998 Microchip Technology Inc.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

PIC16F87X

Pin Diagrams



 © 1998 Microchip Technology Inc.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Key Features PICmicro™ Mid-Range Reference Manual (DS33023)	PIC16F873	PIC16F874	PIC16F876	PIC16F877
Operating Frequency	DC - 20 MHz	DC - 20 MHz	DC - 20 MHz	DC - 20 MHz
Resets (and Delays)	POR, BOR (PWRT, OST)	POR, BOR (PWRT, OST)	POR, BOR (PWRT, OST)	POR, BOR (PWRT, OST)
FLASH Program Memory (14-bit words)	4K	4K	8K	8K
Data Memory (bytes)	192	192	368	368
EEPROM Data Memory	128	128	256	256
Interrupts	13	14	13	14
I/O Ports	Ports A,B,C	Ports A,B,C,D,E	Ports A,B,C	Ports A,B,C,D,E
Timers	3	3	3	3
Capture/Compare/PWM modules	2	2	2	2
Serial Communications	MSSP, USART	MSSP, USART	MSSP, USART	MSSP, USART
Parallel Communications	—	PSP	—	PSP
10-bit Analog-to-Digital Module	5 input channels	8 input channels	5 input channels	8 input channels
Instruction Set	35 Instructions	35 Instructions	35 Instructions	35 Instructions

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.0 DEVICE OVERVIEW

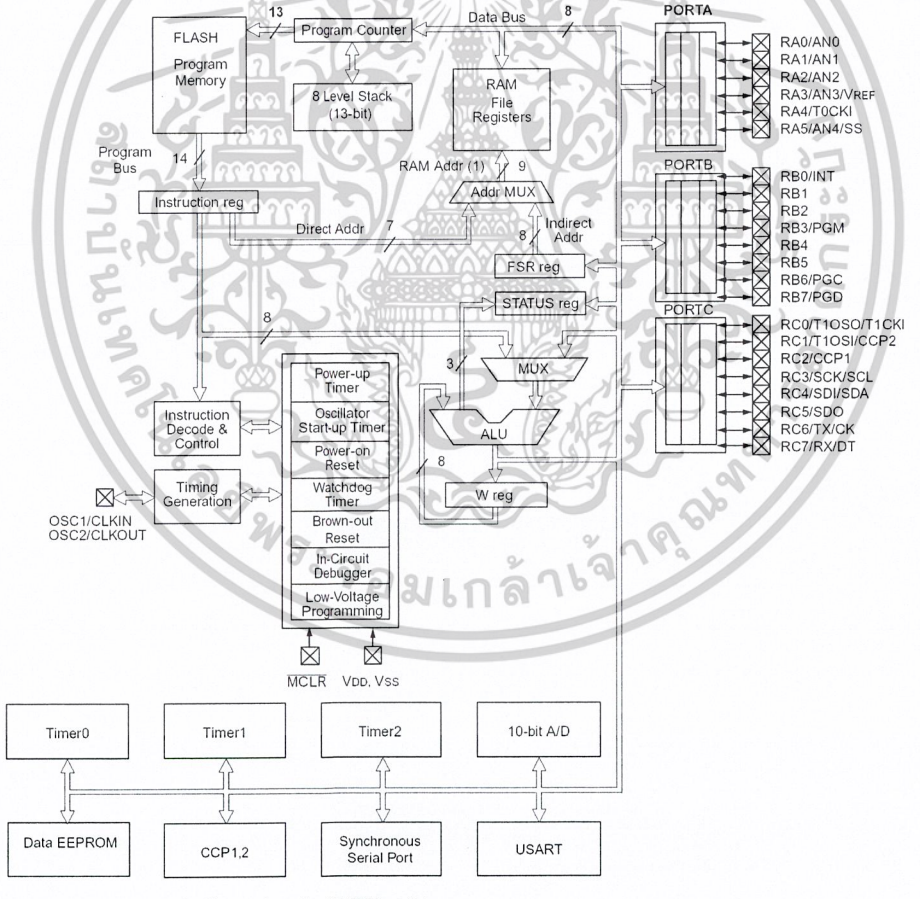
This document contains device-specific information. Additional information may be found in the PICmicro™ Mid-Range Reference Manual, (DS33023), which may be obtained from your local Microchip Sales Representative or downloaded from the Microchip website. The Reference Manual should be considered a complementary document to this data sheet, and is highly recommended reading for a better understanding of the device architecture and operation of the peripheral modules.

There a four devices (PIC16F873, PIC16F874, PIC16F876, and PIC16F877) covered by this data sheet. The PIC16F876/873 devices come in 28-pin packages and the PIC16F877/874 devices come in 40-pin packages. The 28-pin devices do not have a Parallel Slave Port implemented.

The following two figures are device block diagrams sorted by pin number; 28-pin for Figure 1-1 and 40-pin for Figure 1-2. The 28-pin and 40-pin pinouts are listed in Table 1-1 and Table 1-2, respectively.

FIGURE 1-1: PIC16F873 AND PIC16F876 BLOCK DIAGRAM

Device	Program FLASH	Data Memory	Data EEPROM
PIC16F873	4K	192 Bytes	128 Bytes
PIC16F876	8K	368 Bytes	256 Bytes



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

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.0 I/O PORTS

Some pins for these I/O ports are multiplexed with an alternate function for the peripheral features on the device. In general, when a peripheral is enabled, that pin may not be used as a general purpose I/O pin.

Additional information on I/O ports may be found in the PICmicro™ Mid-Range Reference Manual, (DS33023).

3.1 PORTA and the TRISA Register

PORTA is a 6-bit wide bi-directional port. The corresponding data direction register is TRISA. Setting a TRISA bit (=1) will make the corresponding PORTA pin an input, i.e., put the corresponding output driver in a hi-impedance mode. Clearing a TRISA bit (=0) will make the corresponding PORTA pin an output, i.e., put the contents of the output latch on the selected pin.

Reading the PORTA register reads the status of the pins whereas writing to it will write to the port latch. All write operations are read-modify-write operations. Therefore a write to a port implies that the port pins are read, this value is modified, and then written to the port data latch.

Pin RA4 is multiplexed with the Timer0 module clock input to become the RA4/T0CKI pin. The RA4/T0CKI pin is a Schmitt Trigger input and an open drain output. All other RA port pins have TTL input levels and full CMOS output drivers.

Other PORTA pins are multiplexed with analog inputs and analog VREF input. The operation of each pin is selected by clearing/setting the control bits in the ADCON1 register (A/D Control Register1).

Note:	On a Power-on Reset, these pins are configured as analog inputs and read as '0'.

The TRISA register controls the direction of the RA pins, even when they are being used as analog inputs. The user must ensure the bits in the TRISA register are maintained set when using them as analog inputs.

EXAMPLE 3-1: INITIALIZING PORTA

```
BCF     STATUS, RP0     ;
CLRF    PORTA           ; Initialize PORTA by
                        ; clearing output
                        ; data latches
BSF     STATUS, RP0     ; Select Bank 1
MOVLW   0xCF            ; Value used to
                        ; initialize data
                        ; direction
MOVWF   TRISA           ; Set RA<3:0> as inputs
                        ; RA<5:4> as outputs
                        ; TRISA<7:6> are always
                        ; read as '0'.
```

FIGURE 3-1: BLOCK DIAGRAM OF RA3:RA0 AND RA5 PINS

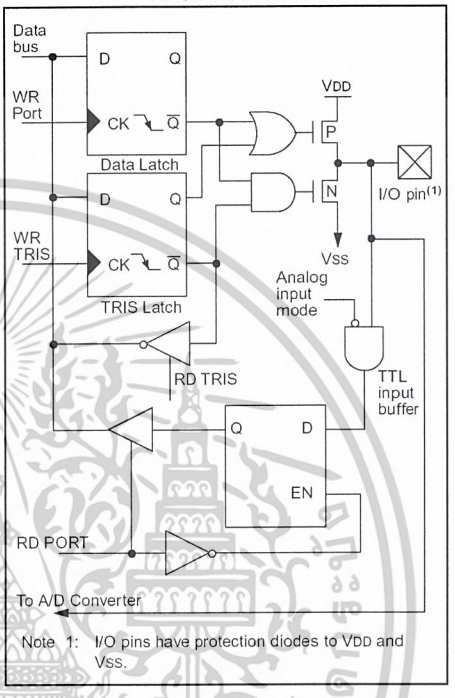
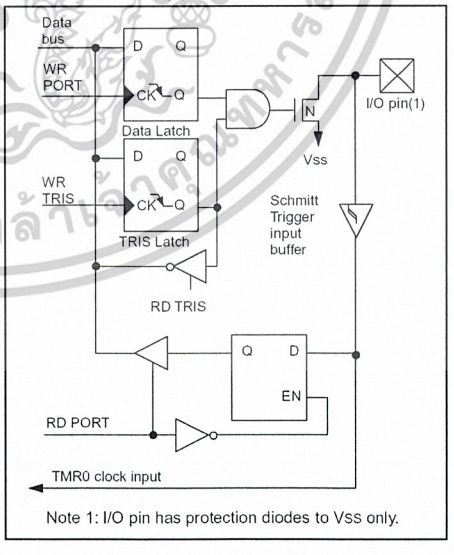


FIGURE 3-2: BLOCK DIAGRAM OF RA4/ T0CKI PIN



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3 PORTC and the TRISC Register

PORTC is an 8-bit wide bi-directional port. The corresponding data direction register is TRISC. Setting a TRISC bit (=1) will make the corresponding PORTC pin an input, i.e., put the corresponding output driver in a hi-impedance mode. Clearing a TRISC bit (=0) will make the corresponding PORTC pin an output, i.e., put the contents of the output latch on the selected pin.

PORTC is multiplexed with several peripheral functions (Table 3-5). PORTC pins have Schmitt Trigger input buffers.

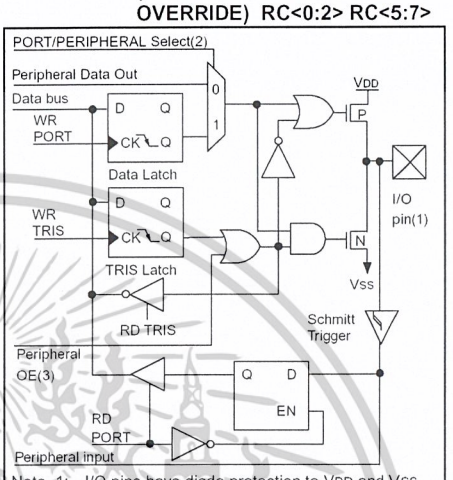
When the I^2C module is enabled, the PORTC (3:4) pins can be configured with normal I^2C levels or with SMBUS levels by using the CKE bit (SSPSTAT <6>).

When enabling peripheral functions, care should be taken in defining TRIS bits for each PORTC pin. Some peripherals override the TRIS bit to make a pin an output, while other peripherals override the TRIS bit to make a pin an input. Since the TRIS bit override is in effect while the peripheral is enabled, read-modify-write instructions (BSF, BCF, XORWF) with TRISC as destination should be avoided. The user should refer to the corresponding peripheral section for the correct TRIS bit settings.

EXAMPLE 3-1: INITIALIZING PORTC

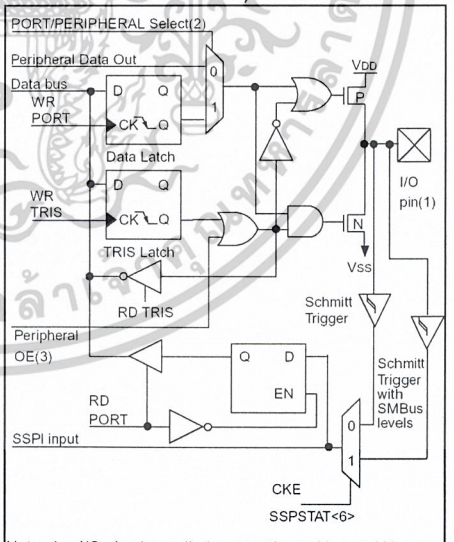
```
BCF     STATUS, RP0   ; Select Bank 0
CLRF    PORTC         ; Initialize PORTC by
                      ; clearing output
                      ; data latches
BSF     STATUS, RP0   ; Select Bank 1
MOVLW   0xCF          ; Value used to
                      ; initialize data
                      ; direction
MOVWF   TRISC         ; Set RC<3:0> as inputs
                      ; RC<5:4> as outputs
                      ; RC<7:6> as inputs
```

FIGURE 3-5: PORTC BLOCK DIAGRAM (PERIPHERAL OUTPUT OVERRIDE) RC<0:2> RC<5:7>



Note 1: I/O pins have diode protection to VDD and VSS.
2: Port/Peripheral select signal selects between port data and peripheral output.
3: Peripheral OE (output enable) is only activated if peripheral select is active.

FIGURE 3-6: PORTC BLOCK DIAGRAM (PERIPHERAL OUTPUT OVERRIDE) RC<3:4>



Note 1: I/O pins have diode protection to VDD and VSS.
2: Port/Peripheral select signal selects between port data and peripheral output.
3: Peripheral OE (output enable) is only activated if peripheral select is active.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

PIC16F87X PRODUCT IDENTIFICATION SYSTEM

To order or obtain information, e.g., on pricing or delivery, refer to the factory or the listed sales office.

PART NO.	-XX	X	/XX	XXX
Device	Frequency Range	Temperature Range	Package	Pattern

Device	PIC16F87X[1], PIC16F87XT[2]: VDD range 4.0V to 5.5V PIC16LF87X[1], PIC16LF87XT[2]: VDD range 2.0V to 5.5V PIC16F87X[1], PIC16F87XT[2]: VDD range 4.0V to 5.5V PIC16LF87X[1], PIC16LF87XT[2]: VDD range 2.0V to 5.5V	
Frequency Range	04 = 4 MHz 20 = 20 MHz	
Temperature Range	b[3] = 0°C to 70°C (Commercial) I = -40°C to +85°C (Industrial)	
Package	PQ = MQFP (Metric PQFP) PT = TQFP (Thin Quad Flatpack) SO = SOIC SP = Skinny plastic dip P = PDIP L = PLCC	
Pattern	QTP, SQTP, Code or Special Requirements (blank otherwise)	

Examples:

g) PIC16F877 -20/P 301 = Commercial temp., PDIP package, 4 MHz, normal VDD limits, QTP pattern #301.

h) PIC16F876 - 04I/SO = Industrial temp., SOIC package, 200 kHz, Extended VDD limits.

i) PIC16F877 - 04I/P = Industrial temp., PDIP package, 10MHz, normal VDD limits.

Note 1: C = CMOS
LC = Low Power CMOS
T = in tape and reel - SOIC, PLCC, MQFP, TQFP packages only.

2: b = blank

* JW Devices are UV erasable and can be programmed to any device configuration. JW Devices meet the electrical requirement of each oscillator type (including LC devices).

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

DALLAS SEMICONDUCTOR

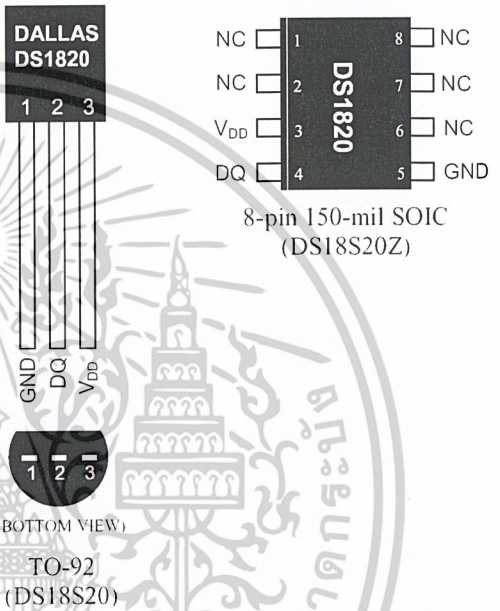
www.dalsemi.com

DS18S20
High Precision 1-Wire® Digital Thermometer

FEATURES

- Unique 1-wire interface requires only one port pin for communication
- Each device has a unique 64-bit serial code stored in an on-board ROM
- Multi-drop capability simplifies distributed temperature sensing applications
- Requires no external components
- Can be powered from data line. Power supply range is 3.0V to 5.5V
- Measures temperatures from –55°C to +125°C (–67°F to +257°F)
- ±0.5°C accuracy from –10°C to +85°C
- 9-bit thermometer resolution
- Converts temperature in 750 ms (max.)
- User-definable nonvolatile alarm settings
- Alarm search command identifies and addresses devices whose temperature is outside of programmed limits (temperature alarm condition)
- Applications include thermostatic controls, industrial systems, consumer products, thermometers, or any thermally sensitive system

PIN ASSIGNMENT



TO-92 (DS18S20)

8-pin 150-mil SOIC (DS18S20Z)

PIN DESCRIPTION

GND - Ground
DQ - Data In/Out
V_{DD} - Power Supply Voltage
NC - No Connect

DESCRIPTION

The DS18S20 Digital Thermometer provides 9–bit centigrade temperature measurements and has an alarm function with nonvolatile user-programmable upper and lower trigger points. The DS18S20 communicates over a 1-wire bus that by definition requires only one data line (and ground) for communication with a central microprocessor. It has an operating temperature range of –55°C to +125°C and is accurate to ±0.5°C over the range of –10°C to +85°C. In addition, the DS18S20 can derive power directly from the data line ("parasite power"), eliminating the need for an external power supply.

Each DS18S20 has a unique 64-bit serial code, which allows multiple DS18S20s to function on the same 1–wire bus; thus, it is simple to use one microprocessor to control many DS18S20s distributed over a large area. Applications that can benefit from this feature include HVAC environmental controls, temperature monitoring systems inside buildings, equipment or machinery, and process monitoring and control systems.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

DETAILED PIN DESCRIPTIONS Table 1

8-PIN SOIC*	TO-92	SYMBOL	DESCRIPTION
5	1	GND	**Ground.**
4	2	DQ	**Data Input/Output pin.** Open-drain 1-wire interface pin. Also provides power to the device when used in parasite power mode (see "Parasite Power" section.)
3	3	V_{DD}	**Optional V_{DD} pin.** V_{DD} must be grounded for operation in parasite power mode.

*All pins not specified in this table are "No Connect" pins.

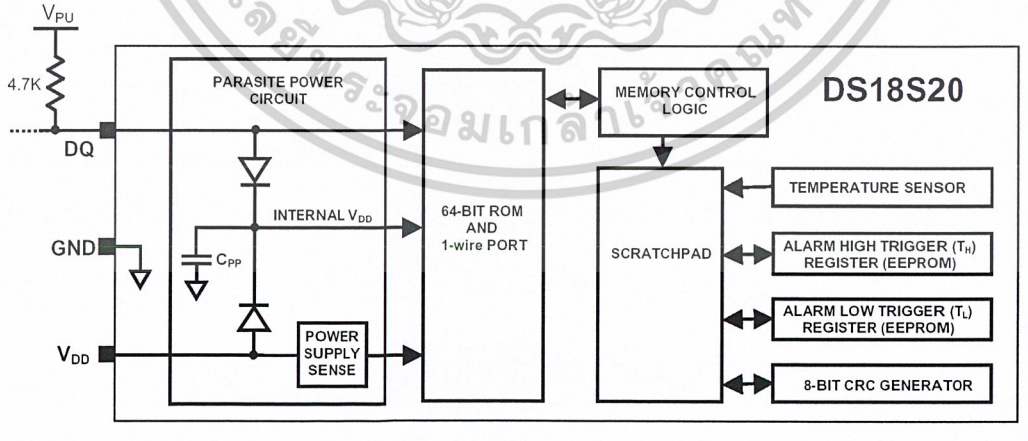
OVERVIEW

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

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

Another feature of the DS18S20 is the ability to operate without an external power supply. Power is instead supplied through the 1-wire pullup resistor via the DQ pin when the bus is high. The high bus signal also charges an internal capacitor (C_{PP}), which then supplies power to the device when the bus is low. This method of deriving power from the 1-wire bus is referred to as "parasite power." As an alternative, the DS18S20 may also be powered by an external supply on V_{DD}.

DS18S20 BLOCK DIAGRAM Figure 1



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

OPERATION – MEASURING TEMPERATURE

The core functionality of the DS18S20 is its direct-to-digital temperature sensor. The temperature sensor output has 9-bit resolution, which corresponds to 0.5°C steps. The DS18S20 powers-up in a low-power idle state; to initiate a temperature measurement and A-to-D conversion, the master must issue a Convert T [44h] command. Following the conversion, the resulting thermal data is stored in the 2-byte temperature register in the scratchpad memory and the DS18S20 returns to its idle state. If the DS18S20 is powered by an external supply, the master can issue "read time slots" (see the 1-WIRE BUS SYSTEM section) after the Convert T command and the DS18S20 will respond by transmitting 0 while the temperature conversion is in progress and 1 when the conversion is done. If the DS18S20 is powered with parasite power, this notification technique cannot be used since the bus must be pulled high by a strong pullup during the entire temperature conversion. The bus requirements for parasite power are explained in detail in the POWERING THE DS18S20 section of this datasheet.

The DS18S20 output data is calibrated in degrees centigrade; for Fahrenheit applications, a lookup table or conversion routine must be used. The temperature data is stored as a 16-bit sign-extended two's complement number in the temperature register (see Figure 2). The sign bits (S) indicate if the temperature is positive or negative: for positive numbers S = 0 and for negative numbers S = 1. Table 2 gives examples of digital output data and the corresponding temperature reading.

Resolutions greater than 9 bits can be calculated using the data from the temperature, COUNT REMAIN and COUNT PER °C registers in the scratchpad. Note that the COUNT PER °C register is hard-wired to 16 (10h). After reading the scratchpad, the TEMP_READ value is obtained by truncating the 0.5°C bit (bit 0) from the temperature data (see Figure 2). The extended resolution temperature can then be calculated using the following equation:

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

Additional information about high-resolution temperature calculations can be found in Application Note 105: "High Resolution Temperature Measurement with Dallas Direct-to-Digital Temperature Sensors".

TEMPERATURE REGISTER FORMAT Figure 2

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

TEMPERATURE/DATA RELATIONSHIP Table 2

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

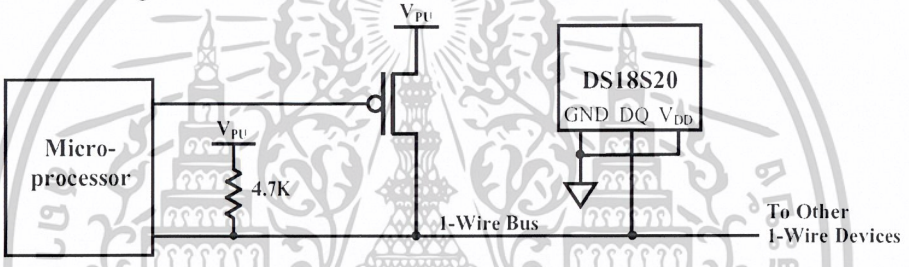
*The power-on reset value of the temperature register is +85°C

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

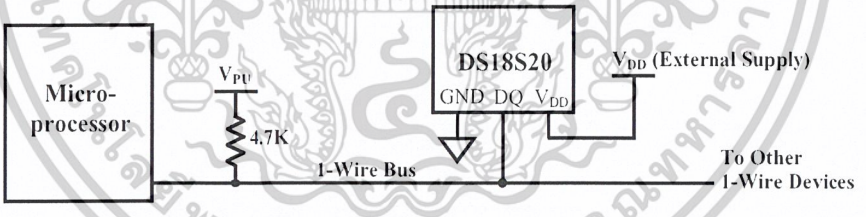
The use of parasite power is not recommended for temperatures above 100°C since the DS18S20 may not be able to sustain communications due to the higher leakage currents that can exist at these temperatures. For applications in which such temperatures are likely, it is strongly recommended that the DS18S20 be powered by an external power supply.

In some situations the bus master may not know whether the DS18S20s on the bus are parasite powered or powered by external supplies. The master needs this information to determine if the strong bus pullup should be used during temperature conversions. To get this information, the master can issue a Skip ROM [CCh] command followed by a Read Power Supply [B4h] command followed by a "read time slot". During the read time slot, parasite powered DS18S20s will pull the bus low, and externally powered DS18S20s will let the bus remain high. If the bus is pulled low, the master knows that it must supply the strong pullup on the 1-wire bus during temperature conversions.

SUPPLYING THE PARASITE-POWERED DS18S20 DURING TEMPERATURE CONVERSIONS Figure 4



POWERING THE DS18S20 WITH AN EXTERNAL SUPPLY Figure 5



64-BIT LASERED ROM CODE

Each DS18S20 contains a unique 64–bit code (see Figure 6) stored in ROM. The least significant 8 bits of the ROM code contain the DS18S20's 1–wire family code: 10h. The next 48 bits contain a unique serial number. The most significant 8 bits contain a cyclic redundancy check (CRC) byte that is calculated from the first 56 bits of the ROM code. A detailed explanation of the CRC bits is provided in the CRC GENERATION section. The 64–bit ROM code and associated ROM function control logic allow the DS18S20 to operate as a 1–wire device using the protocol detailed in the 1-WIRE BUS SYSTEM section of this datasheet.

64-BIT LASERED ROM CODE Figure 6

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

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

DS18S20 FUNCTION COMMAND SET Table 4

Command	Description	Protocol	1-Wire Bus Activity After Command is Issued	Notes
TEMPERATURE CONVERSION COMMANDS				
Convert T	Initiates temperature conversion.	44h	DS18S20 transmits conversion status to master (not applicable for parasite-powered DS18S20s).	1
MEMORY COMMANDS				
Read Scratchpad	Reads the entire scratchpad including the CRC byte.	BEh	DS18S20 transmits up to 9 data bytes to master.	2
Write Scratchpad	Writes data into scratchpad bytes 2 and 3 (T_H and T_L).	4Eh	Master transmits 2 data bytes to DS18S20.	3
Copy Scratchpad	Copies T_H and T_L data from the scratchpad to EEPROM.	48h	None	1
Recall E^2	Recalls T_H and T_L data from EEPROM to the scratchpad.	B8h	DS18S20 transmits recall status to master.	
Read Power Supply	Signals DS18S20 power supply mode to the master.	B4h	DS18S20 transmits supply status to master.	

NOTES:

1. For parasite-powered DS18S20s, the master must enable a strong pullup on the 1-wire bus during temperature conversions and copies from the scratchpad to EEPROM. No other bus activity may take place during this time.
2. The master can interrupt the transmission of data at any time by issuing a reset.
3. Both bytes must be written before a reset is issued.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ABSOLUTE MAXIMUM RATINGS*

Voltage on any pin relative to ground	–0.5V to +6.0V
Operating temperature	–55°C to +125°C
Storage temperature	–55°C to +125°C
Soldering temperature	See J-STD-020A Specification

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

DC ELECTRICAL CHARACTERISTICS (-55°C to +125°C; V_{DD}=3.0V to 5.5V)

PARAMETER	SYMBOL	CONDITION	MIN	TYP	MAX	UNITS	NOTES
Supply Voltage	V_{DD}	Local Power	+3.0		+5.5	V	1
Pullup Supply Voltage	V_{PU}	Parasite Power	+3.0		+5.5	V	1,2
		Local Power	+3.0		V_{DD}		
Thermometer Error	t_{ERR}	-10°C to +85°C			±0.5	°C	3
		-55°C to +125°C			±2		
Input Logic Low	V_{IL}		-0.3		+0.8	V	1,4,5
Input Logic High	V_{IH}	Local Power	+2.2		The lower of 5.5 or V_{DD} + 0.3	V	1, 6
		Parasite Power	+3.0				
Sink Current	I_L	$V_{I/O}$=0.4V	4.0			mA	1
Standby Current	I_{DDS}			750	1000	nA	7,8
Active Current	I_{DD}	V_{DD}=5V		1	1.5	mA	9
DQ Input Current	I_{DQ}			5		µA	10
Drift				±0.2		°C	11

NOTES:

1. All voltages are referenced to ground.
2. The Pullup Supply Voltage specification assumes that the pullup device is ideal, and therefore the high level of the pullup is equal to V_{PU}. In order to meet the V_{IH} spec of the DS18S20, the actual supply rail for the strong pullup transistor must include margin for the voltage drop across the transistor when it is turned on; thus: $V_{PU_ACTUAL} = V_{PU_IDEAL} + V_{TRANSISTOR}$.
3. See typical performance curve in Figure 16
4. Logic low voltages are specified at a sink current of 4 mA.
5. To guarantee a presence pulse under low voltage parasite power conditions, V_{ILMAX} may have to be reduced to as low as 0.5V.
6. Logic high voltages are specified at a source current of 1 mA.
7. Standby current specified up to 70°C. Standby current typically is 3 µA at 125°C.
8. To minimize I_{DDS}, DQ should be within the following ranges: GND ≤ DQ ≤ GND + 0.3V or V_{DD} – 0.3V ≤ DQ ≤ V_{DD}.
9. Active current refers to supply current during active temperature conversions or EEPROM writes.
10. DQ line is high ("hi-Z" state).
11. Drift data is based on a 1000 hour stress test at 125°C with V_{DD} = 5.5V.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

AC ELECTRICAL CHARACTERISTICS: NV MEMORY

(-55°C to +100°C; V_{DD}=3.0V to 5.5V)

PARAMETER	SYMBOL	CONDITION	MIN	TYP	MAX	UNITS
NV Write Cycle Time	t_{WR}			2	10	ms
EEPROM Writes	N_{EEWR}	-55°C to +55°C	50k			writes
EEPROM Data Retention	t_{EEDR}	-55°C to +55°C	10			years

AC ELECTRICAL CHARACTERISTICS (-55°C to +125°C; V_{DD}=3.0V to 5.5V)

PARAMETER	SYMBOL	CONDITION	MIN	TYP	MAX	UNITS	NOTES
Temperature Conversion Time	t_{CONV}				750	ms	1
Time to Strong Pullup On	t_{SPON}	Start Convert T Command Issued			10	µs	
Time Slot	t_{SLOT}		60		120	µs	1
Recovery Time	t_{REC}		1			µs	1
Write 0 Low Time	t_{LOW0}		60		120	µs	1
Write 1 Low Time	t_{LOW1}		1		15	µs	1
Read Data Valid	t_{RDV}				15	µs	1
Reset Time High	t_{RSTH}		480			µs	1
Reset Time Low	t_{RSTL}		480			µs	1,2
Presence Detect High	t_{PDHIGH}		15		60	µs	1
Presence Detect Low	t_{PDLOW}		60		240	µs	1
Capacitance	$C_{IN/OUT}$				25	pF	

NOTES:

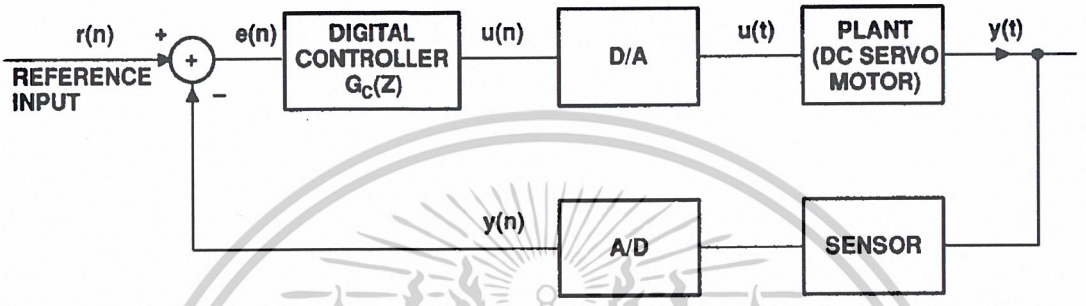
1. Refer to timing diagrams in Figure 17.
2. Under parasite power, if t_{RSTL} > 960 µs, a power on reset may occur.

TYPICAL PERFORMANCE CURVE Figure 16

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

put of the motor, y(t), is measured by a sensor and converted into a digital signal, y(n), by the A/D. The feedback signal is subtracted from the reference signal r(n) to create an error signal e(n). The error signal, e(n), is used by the controller to issue the corresponding control action u(n).

Figure 1. Digital Control System



Analog Versus Digital Controllers

Tradeoffs have to be made in the selection of controllers for a system. Analog controllers provide continous processing of the signal and can be used for very high bandwidth systems. They also give almost infinite resolution of the signal they are measuring, thus providing precise control. Analog controllers have been around for a long time; their behavior is well understood, and this makes them easy to design. They can be implemented with relatively inexpensive components and therefore are sometimes cheaper.

On the negative side, analog controllers suffer from component aging and temperature drifts. Thus, a perfectly designed controller will start to exhibit undesired characteristics after a while. Analog controllers are hardwired solutions; this makes modifications or upgrades in the design difficult. Analog controllers are also limited to simpler algorithms from classical control theory like PID and compensation techniques.

Digital controllers sample the signal at discrete time intervals. This limits the bandwidth (bandwidth is 1/6 to 1/10 sampling rate) that can be handled by the controller. The processing of the signal takes a finite amount of time, adding to phase delay in the system. In addition, the resolution of the signal is limited by the resolution or wordlength of the processor. Digital controllers also require additional components like A/D and D/A; although newer processors include these components on the same chip. Digital controllers are relatively new, and their behavior is not very well understood, thus making design of digital controllers relatively difficult in comparison to analog controllers.

However, digital controllers have some advantages also. They are not affected by component aging and temperature drift, and they provide stable performance. When the design is done in the z-domain, the behavior of digital controllers can be more precisely controlled. They can also be used to implement more sophisticated techniques from modern control theory, such as state controllers, optimal control, and adaptive control. Digital controllers are programmable, thus making them easy to upgrade and maintaining design investment. They can also be time shared to implement different functions in the system, like notch filters and system control, thus reducing system

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

cost. If digital controllers are designed properly, their advantages outweigh their disadvantages. Table 1 compares analog and digital controllers.

Table 1. Comparison of Analog versus Digital Controllers

Controller	Advantages	Disadvantages
Analog	High bandwidth High resolution Ease of design	Component aging Temperature drift Hardwired design Good only for simpler designs
Digital	Programmable solution Insensitive to environment Shows precise behavior Implements advanced algorithms Capable of additional functions	Creates numerical problems Must use high-performance processor Difficult to design

Processor Selection Issues

The choice of processor is critical in determing the performance and behavior of the digital controller. The usual choices are microcontrollers, general purpose microprocessors, and digital signal processors (DSP). RISC processors and bit-slice processors can also be used; however it is not practical to use them in most cases because of their cost.

Digital controllers monitor signals at discrete time intervals or finite sampling rates. If the signal is not sampled fast enough, some of the information may be lost. The processing of the signal takes a finite amount of time. The processing has to be completed before the arrival of the next sample, or preferably as soon as possible. Too much delay in the output can cause loss of information or excessive phase delay in the system, leading to instability. These conditions impose certain minimum performance requirements on the processor. Most of the processors currently used to implement controllers are usually not fast enough to process signals in real time; they rely upon lookup tables with precomputed results.

Digital controllers use discrete steps to represent a signal, which is limited to the wordlength of the processor. Coefficients or gain constants also have to be represented in the limited wordlength. This discretization or loss of resolution is referred to as quantization error. In addition, results of mathematical operations have to fit in a limited wordlength and may lose part of the result due to this limitation. This is referred to as truncation error. Both of these errors cause oscillations or limit cycles and can lead to instability.

Another problem that occurs in digital controllers is overflow of registers. Successive mathematical operations can cause registers to overflow. Registers in most processors wrap around, causing the result of a calculation to go from most positive to least negative, in turn causing the output to reverse directions. Most of these problems occur in microcontrollers and microprocessors because their architectures are not designed for signal processing.

Early microprocessors (μP) and microcontrollers (μC) were designed to replace hardwired logic, TTL gates, etc. Newer microprocessors and microcontrollers have retained most of the old

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Control Algorithms

The next step in designing the controller is to select an appropriate algorithm or controller structure. The processing burden imposed upon the controller is directly dependent upon the complexity and type of controller structure.

Compensation Techniques

Compensation techniques are one of the most commonly used control techniques. In this technique, the controller adds poles and zeros to get a desired system response. For a continuous control system, the controller is designed in the s-domain by using some of the well known methods such as root locus, Bode plots, and Nyquist plots. The analog or s-domain design is then transferred into a discrete form (z- domain) by using a transformation technique. Alternatively, the compensator can be designed directly in the z-domain by using z-domain frequency response methods or the z-domain root locus method. Compensation techniques allow a precise modification to system behavior. Implementation of compensation techniques is described in an application report, Control System Compensation and Implementation with the TMS32010[5].

PID

The PID is a commonly used analog control technique. In a PID controller, terms proportional to the error term, its integral, and its derivative are summed to achieve the controller output. A PID controller may be designed in the s-domain, and then transferred into the z-domain by using one of the transformation methods. Alternatively, the PID algorithm is converted into a discrete form, and the design is carried out entirely in the z-domain. PID is probably the most commonly used algorithm. PID controllers are very robust; although the design of coefficients is somewhat arbitrary[6, 7].

Deadbeat

A deadbeat algorithm is used when a quick settling time is required. Deadbeat design is carried out entirely in the z-domain. A deadbeat controller replaces the poles of the system with poles at the origin of z-domain[8].

State Models

In a state model, a complete representation of the system is made in matrix form. This is accomplished by identifying and developing the relationship between the different states or variables of the plant. An appropriate feedback gain can be chosen to place the poles of the system at any desired location in the z-domain. State controllers are used where multiple variables or states need to be controlled. State controllers are sometimes impractical to implement because it may not be possible to measure all states. They are usually used in conjunction with observers. State controllers allow very precise control of system behavior[6, 9, 10].

Observer Models

Often, in control systems, some of the states of the system are not available for measurement. The measurement of known states in an observer model can be used to estimate unknown states in the control system. The estimated states, along with an appropriate feedback gain, can be used

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

to complete the control loop and place the poles at any desired location. Observers are typically used in conjunction with state controllers, where access to all state variables may not be available[6, 9, 10, 11].

Optimal Control

Optimal control synthesis is used when a specific performance or cost criteria (e.g., time or energy) must be minimized. The given criteria or cost function is used to derive an appropriate control law, which is then implemented with a controller or compensator[7, 12].

Kalman Filter

An observer model can be used in a system where an exact measurement of some states is available. However, in stochastic systems, the presence of noise or uncertainty makes it impossible to make an exact measurement. A Kalman filter is an observer model in a noisy or stochastic system[7, 13, 14].

Adaptive Control

Adaptive control is used in systems in which there is insufficient information about the plant parameters, making it impossible to derive a plant model. It is also used in systems where plant parameters or plant models change over time, making the controller unstable. An adaptive controller tracks changes in the plant by redesigning the controller to give an optimum control system[6, 8, 16].

Design and Implementation of PID Controllers

This section describes design and implementation of a PID controller. PID is a commonly used technique in classical control. In designing controllers, it is often found that just minimizing a term proportional to the error is not sufficient. Including the integral of the error term reduces the steady-state error to zero because it represents the accumulated error. To further improve stability and plant dynamics, a differential of the error term is introduced. This term represents the error rate. A PID controller that includes all three terms can give very good results. This technique is also being used in discrete form with digital control systems.

Two different approaches are used for conversion of PID into discrete form: rectangular and trapezoidal approximations. For the rectangular approximation, the design is done in the analog domain and then converted into z-domain. For the trapezoidal approximation, the design is done directly in the z-domain, using pole placement techniques.

The analog PID algorithm is given by

$$u(t) = K_p e(t) + K_i \int d_t + K_d \frac{de}{dt} \tag{8}$$

where

K_p, K_i, and K_d = PID constants
$u(t)$ = output of controller
$e(t)$ = error signal

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Rectangular Approximation

To discretize, assume that the sampling interval for the system is T. The rectangular approximation is the easiest to use and gives satisfactory results. For the rectangular approximation, assume the integral $\int e dt$ is an accumulation of small rectangles given by $T \times \sum e(i)$ for i = 1 to n (see Figure 4). The differential de/dt (if T is sufficiently small) can be approximated by

$$\frac{e(n) - e(n-1)}{T} \tag{9}$$

where e(n) = the nth sample.

Figure 4. Rectangular Approximation



After conversion into its discrete form and transferring into time domain, the final form is shown in (7). The complete derivation is described in Appendix B.

$$u(n) = u(n-2) + K_1 e(n) + K_2 e(n-1) + K_3 e(n-2) \tag{10}$$

where

K_1 = $K_p + K_d/T + K_i T$
K_2 = $K_i T - 2K_d/T$
K_3 = $K_d/T - K_p$
$u(n)$ = control signal at time interval n
$u(n-2)$ = $(n-2)$th sample
$e(n)$ = error signal at time n
$e(n-1)$ = error signal at time $n-1$
T = sampling interval

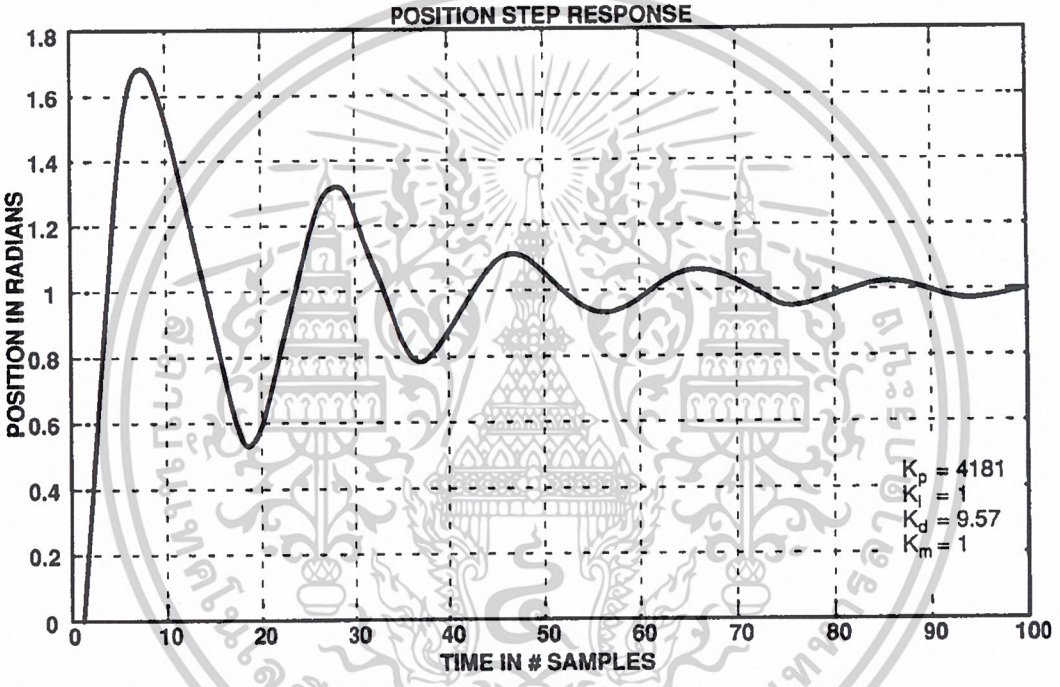
Appendix E shows the code to implement (7) on the PID controller with rectangular approximation. The code takes 11 instructions on the TMS320C15 and executes in 2.6 μs at 20 MHz. The

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

MPY instruction performs a multiply in a single cycle. The LTD instruction loads the T register, implements a data shift or z^{-1} operation, and adds the result of the multiply to the previous value in the accumulator, all in a single cycle.

K_1, K_2, and K_3 are obtained by designing K_p, K_i, and K_d and using conventional techniques from classical control[6, 7] (see Appendix B). Figure 7 through Figure 6 show the step response of the PID controller with different values of K_p, K_i, and K_d.

Figure 5. Step Response of the PID Controller with First Set of K_p, K_i, and K_d Values



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Figure 6. Step Response of the PID Controller with Second Set of K_p, K_i, and K_d Values

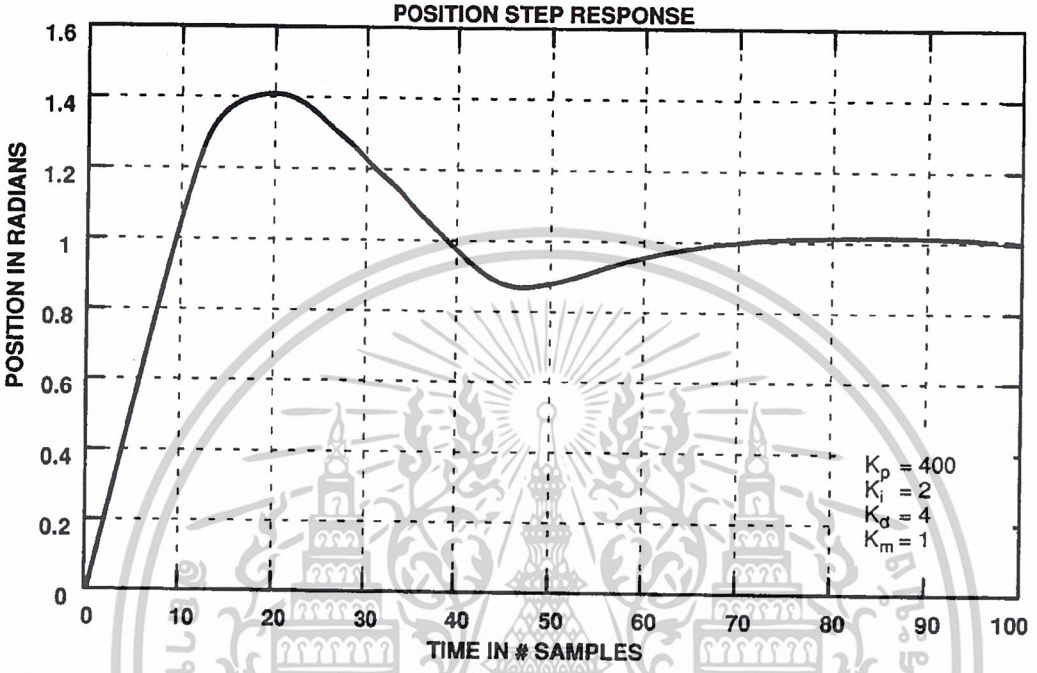
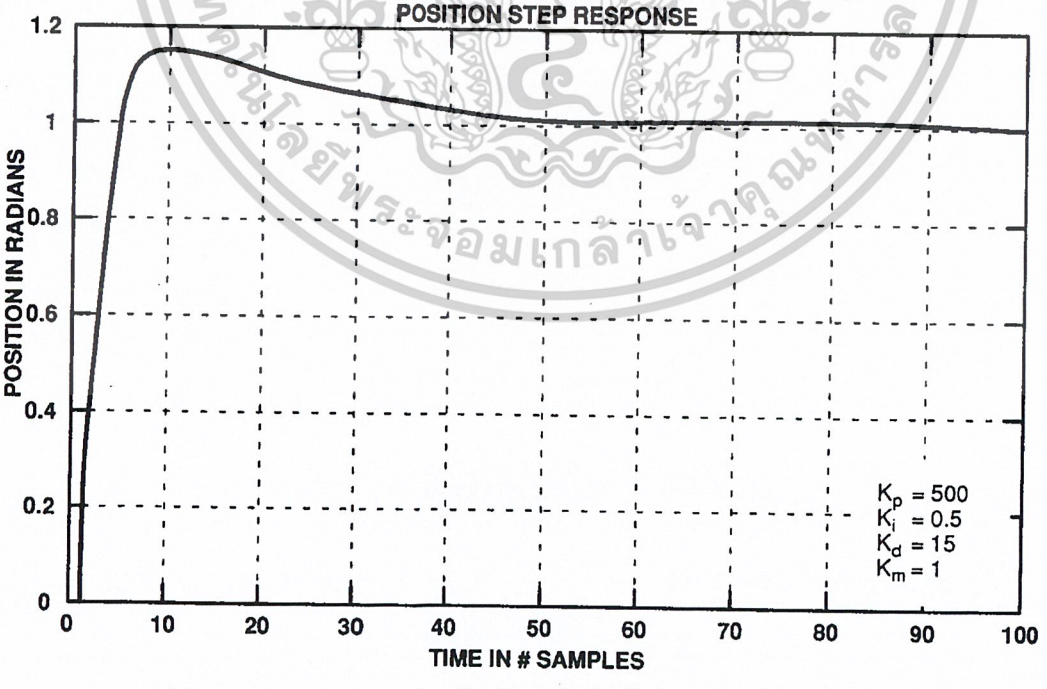


Figure 7. Step Response of the PID Controller with Third Set of K_p, K_i, and K_d Values

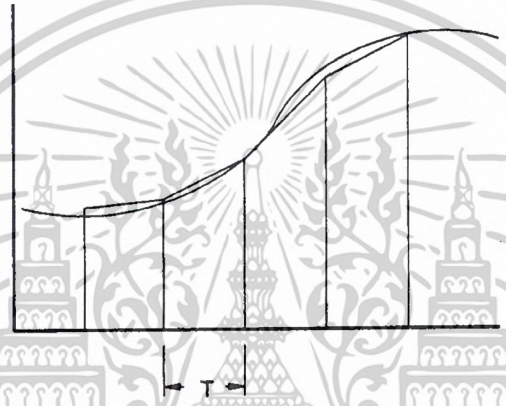


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Trapezoidal Approximation

If a more accurate conversion is required, the trapezoidal approximation or Tustin transformation is used. The area of the integral $\int edt$ is given by the summation of small trapezoids (see Figure 8).

Figure 8. Trapezoidal Approximation



The integral $\int edt$ can also be solved by taking the Laplace transform of (8) and substituting

$$s = 2/T \times (z-1/z+1) \tag{11}$$

After substitution and solving,

$$u(n) = u(n-2) + K_1e(n) + K_2e(n-1) + K_3e(n-2) \tag{12}$$

where

K_1 = $K_p + 2K_d/T + K_iT/2$
K_2 = $K_iT - 4K_d/T$
K_3 = $2K_d/T - K_p + K_iT/2$
$u(n)$ = nth sample of output of controller
$u(n–2)$ = $(n–2)$nd sample of output

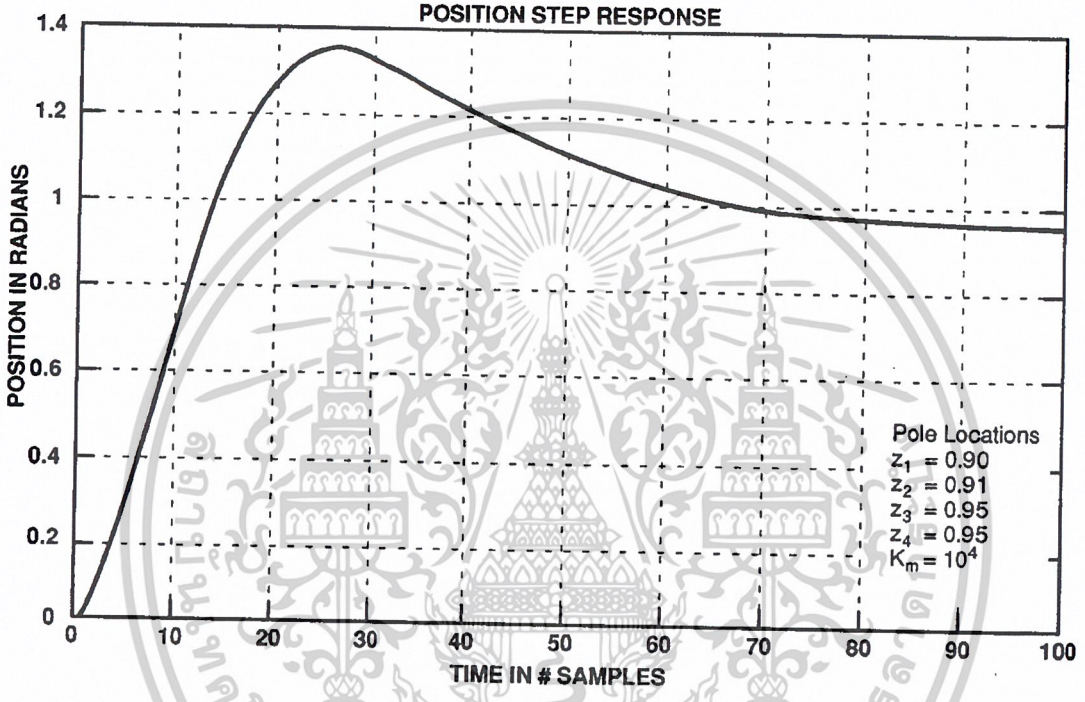
The complete derivation of this equation is described in Appendix B. The code to implement (12) on the PID controller is shown in Appendix E; it takes 12 instructions and executes in 2.8 μs.

The gain constants K_1, K_2, and K_3 are designed by selecting the poles for the system transfer function[15] (see Appendix B). The dominant poles are selected by choosing a desired characteristic equation. The rest of the poles can be selected by placing them near the origin. These polar locations are chosen to ensure system stability and a desired system response. However, some fine tuning may be necessary to achieve an optimum response from the system. As the poles move toward

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

the unit circle, the system response speed decreases while the overshoot decreases; the system may become unstable if the poles are selected too near or outside the unit circle. Figure 9 through Figure 11 show the step response of the PID controller for various pole locations for the system.

Figure 9. PID Controller Step Response for System's First Pole Locations



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Figure 10. PID Controller Step Response for System's Second Pole Locations

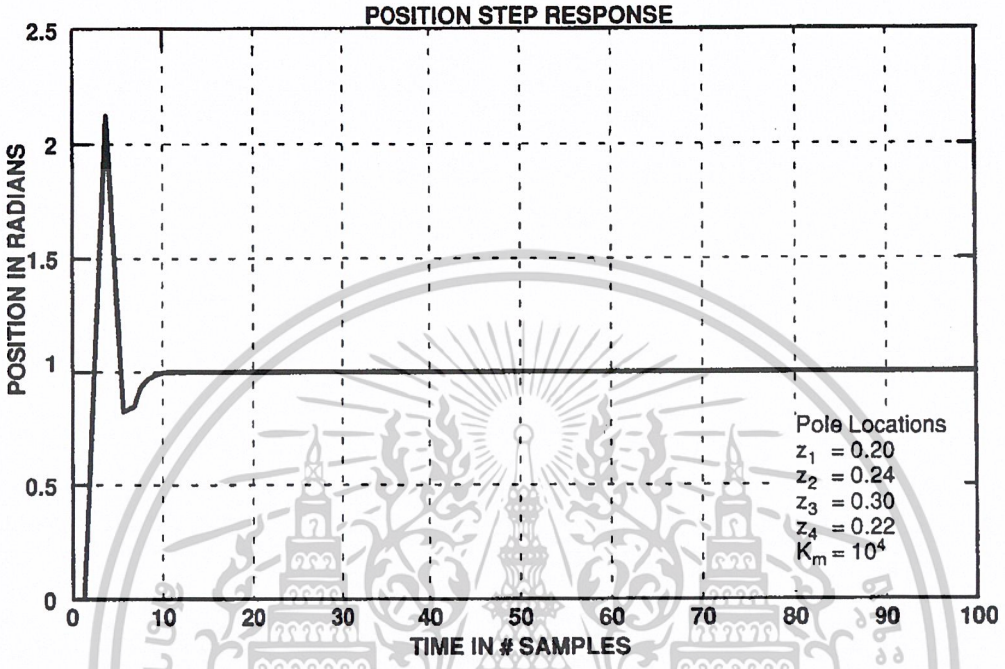
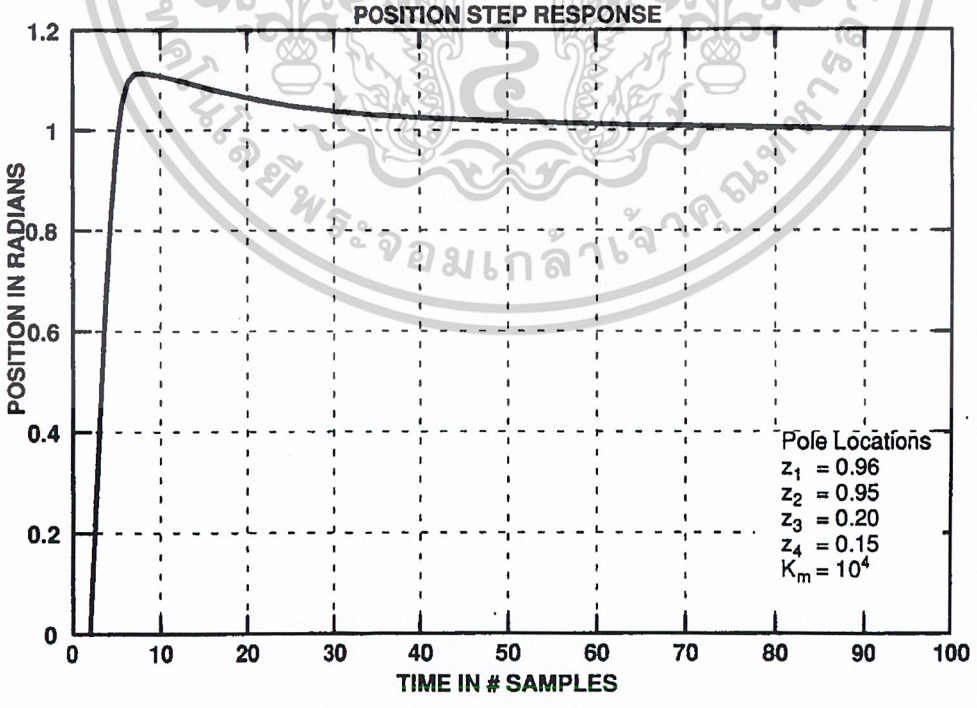


Figure 11. PID Controller Step Response for System's Third Pole Locations



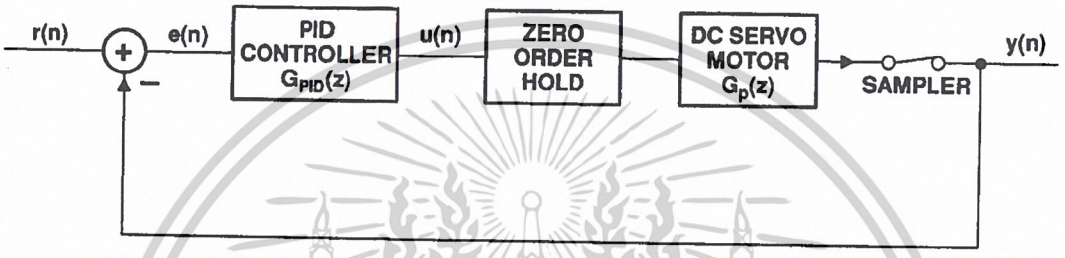
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Our final algorithm comes out to

$$u(n) = u(n-2) + 0.162e(n) - 0.439e(n-1) + 0.3185e(n-2) \quad (13)$$

Figure 12 shows the the block diagram of a system using a PID controller. The zero-order hold represents the function of a D/A, and the sampler represents the function of an A/D.

Figure 12. PID Controller System Block Diagram



Deadbeat

One of the desired characteristics in a control system design is a quick settling time. In an analog controller, it takes the system output an infinite time to settle exactly to the reference input signal. A deadbeat controller is used when a quick or finite settling time is required. A deadbeat controller reaches a steady state in n+1 samples, where n is the order of the controller. Essentially, a deadbeat controller cancels all the poles of the system and replaces them with poles at the origin. Another advantage of deadbeat controllers is that they require few calculations. Therefore, they can be used in systems where synthesis must be repeated frequently (e.g., in adaptive control systems).

Deadbeat controllers compensate for the poles of the system; therefore, they should not be applied to systems with poles outside or in the vicinity of the unit circle in the z-plane. Thus, deadbeat controllers should be used only with stable plants or processes; otherwise they may cause instability. Deadbeat controllers may also require a large amount of gain, thus leading to actuator saturation.

Deadbeat controllers also give a large overshoot. The only design parameter in deadbeat controllers is the sampling period that influences the magnitude of the control signal. When deadbeat control is used, the magnitude of the control signal increases as the sampling period decreases, otherwise, a larger overshoot occurs. Thus, it is important to choose the sampling period carefully when using deadbeat control. Besides increasing the sampling period, there are two other ways to reduce the overshoot.

1) One is to design an extended-order deadbeat controller[8], which allows u(0) or the initial control action to be specified. Since u(0) has the largest magnitude, this allows the overshoot to be controlled.
2) An alternate method is divide the r(t) (the desired final state) into two or three sublevels and reach final steady state in 2(n+1) or 3(n+1) sample times instead of n+1 sample times. This has the same effect as increasing the sample time. However, the final overshoot can be more precisely controlled, depending on how r(t) is subdivided.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

The transfer function of a deadbeat controller is given by

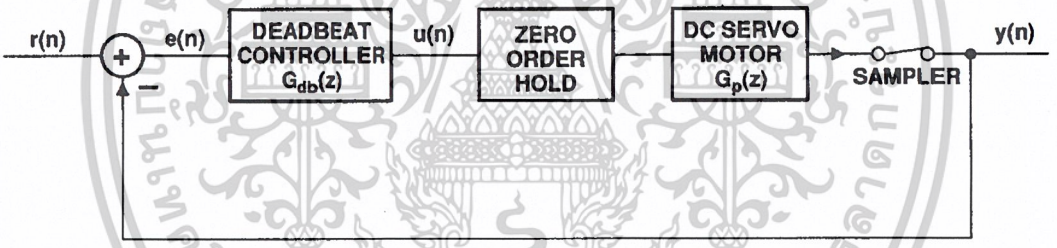
$$G_{db} = \frac{p_0 + p_1 z^{-1} + p_2 z^{-2} + \ldots p_n z^{-n}}{q_0 + q_1 z^{-1} + q_2 z^{-2} + \ldots q_n z^{-n}} \tag{14}$$

The order n of the controller transfer function is the same as the order of the plant transfer function, or n=2. The deadbeat controller will reach final state in n+1 or three sample time intervals. The coefficients p_0, p_1, p_2, q_0, q_1, and q_2 are found from the plant parameters. Appendix C describes the complete equations to find these parameters. Solving for the parameters of the controller, the final transfer function for the controller is

$$G_{db} = \frac{0.1566 - 0.3129z^{-1} + 0.1564z^{-2}}{1 - 0.4218z^{-1} - 0.4216z^{-2}} \tag{15}$$

Figure 13 shows the block diagram of a deadbeat controller and Figure 16 through Figure 15 show the step response of the deadbeat controller. The code to implement the deadbeat controller is given in Appendix E.

Figure 13. Deadbeat Controller Block Diagram



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Figure 14. Step Response of the Deadbeat Controller

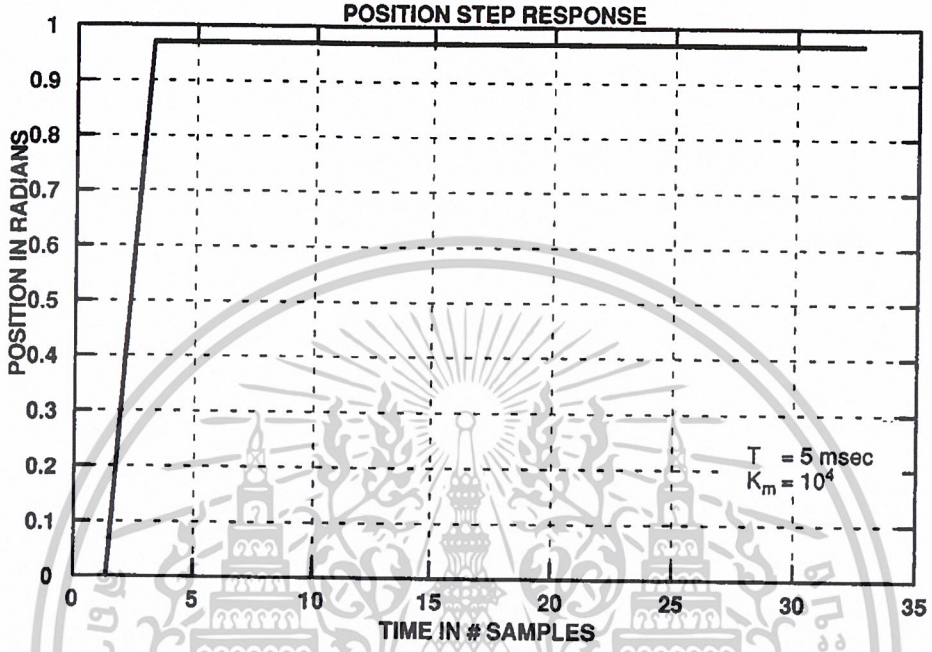
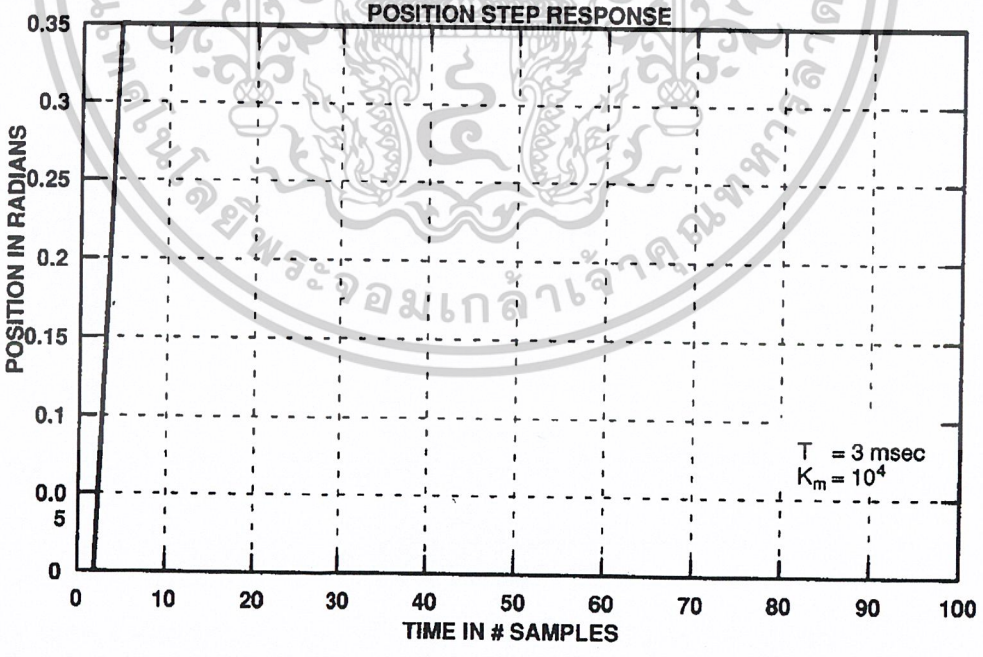
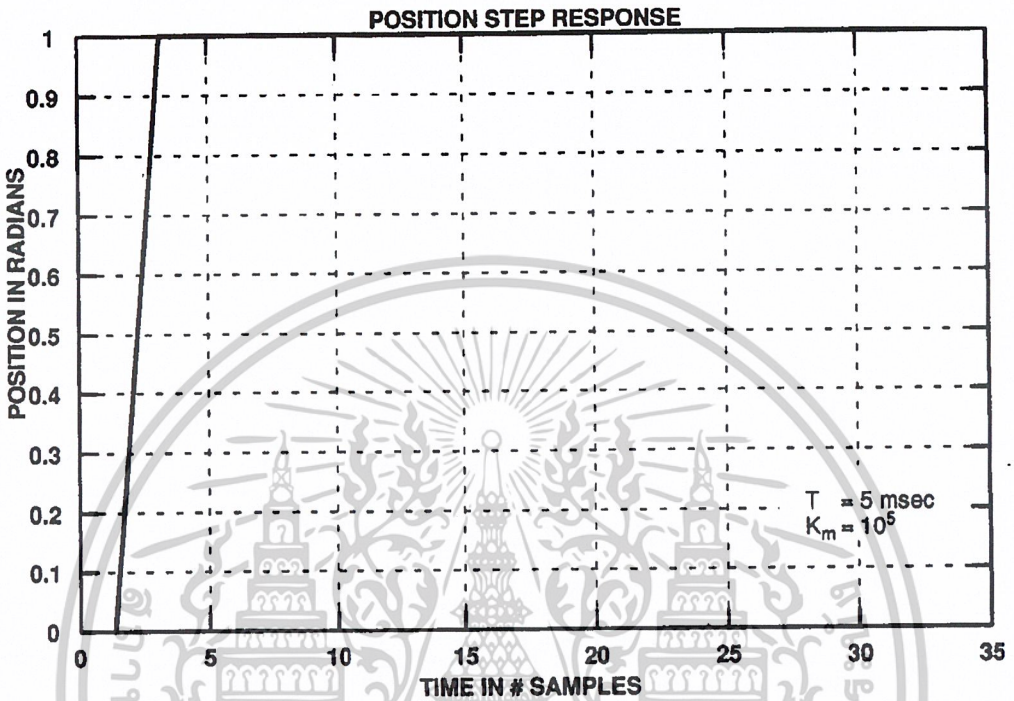


Figure 15. Step Response of the Deadbeat Controller



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Figure 16. Step Response of the Deadbeat Controller



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Appendix B

PID Controller

The PID algorithm is given by

$$u(t) = K_p e(t) + K_i \int edt + K_d \frac{de}{dt} \tag{29}$$

where

K_p, K_i, K_d = PID gain constants
$u(t)$ = control signal
$e(t)$ = error signal

Rectangular Approximation

To convert to discrete form, the integral term edt is approximated by the summation of rectangles $\sum e_i \times T$, where T is the sampling interval and e_i is the value of the error signal at sample time i, written as

$$\int edt = \sum e_i T \tag{30}$$

If the sampling interval T is small enough, the differential term d_e/d_t can be approximated by

$$\frac{de}{dt} = \frac{e\ (n) - e\ (n-1)}{T} \tag{31}$$

where $e(n)$ and $e(n-1)$ are values of the error signal e at time intervals n and $n-1$.

The PID algorithm can now be approximated in discrete form by

$$u(n) = K_p e(n) + K_i \sum e_i T + K_d[e(n) - e(n-1)]/T \tag{32}$$

To reduce (32) into a difference equation, (32) for time interval $n-2$ is written as

$$u(n-2) = K_p e(n-2) + K_i \sum_{i=1}^{n-2} e_i T + K_d[e(n-1) - e(n-2)]/T \tag{33}$$

Subtracting (33) from (32) gives

$$\begin{aligned} u(n) - u(n-2) &= K_p e(n) - e(n-2) + K_i\{e(n) + e(n-1)\}T \\ &\quad + (K_d/T)\{\ e(n) - 2e(n-1) + e(n-2)\ \} \end{aligned} \tag{34}$$

Combining similar terms gives

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$u(n) = u(n-2) + (K_p + K_d/T + K_iT)e(n) - (K_iT - 2K_d/T)e(n-1) + (K_d/T - K_p)e(n-2) \quad (35)$$

or

$$u(n) = u(n-2) + K_1e(n) + K_2e(n-1) + K_3e\ (n-2)$$

where

$K_1 = K_p + K_d/T + K_iT$
$K_2 = K_i\ T - 2K_d/T$
$K_3 = K_d/T - K_p$

K_1, K_2, and K_3 are obtained from the design of K_p, K_i, and K_d, which are designed by using conventional techniques. One way of designing is to use the Ziegler and Nichols ultimative-sensitivity method [6]. With this approach, a proportional controller is first used to control the system. The gain of the controller, Kmax, and the period time, T_p, when the closed loop is on the stability boundary, are measured. The parameters K_p, K_i, and K_d can then obtained as follows:

$K_p = 0.6$ Kmax
$K_i = T_p/2$
$K_d = T_p/8$

Another tuning method is to use the phase margin and critical frequencies [7]. Using this technique, K_p, K_i, and K_d can be computed as follows.

$$\theta = 180 + \phi_m - G_m(j\omega) \mid \omega = \omega_c$$

$$Kp = \frac{\cos(\theta)}{|G_m(j\omega)|\omega = \omega_c} \quad (36)$$

$$K_d\omega_c - K_i/\omega_c = \frac{\sin(\theta)}{\mid G_m(j\omega) \mid \omega = \omega_c}$$

By substituting an arbitrary value of K_i in the above equation, we can obtain K_d. Using this technique and designing with the following parameters

$\phi_m = 55°$ (phase margin)
$\omega_c = 628.315$ radians (critical frequency = 100 hz)

we obtain

$$\mid G_m(j\omega \mid \omega = \omega_c = 0.0001365$$

$$G_m(j\omega)|\omega = \omega_c = -179.8982 \quad (37)$$

The PID constants are then found to be

$K_p = 4181$
$K_d = 9.569$
$K_i = 1$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

K_1, K_2, and K_3 are then obtained as

K_1 = 13751
K_2 = −19138
K_3 = 5387

Both these methods give approximate answers. Further fine tuning of the parameters may be necessary to get the desired response from the system.

The controller is obtained as

$$u(n) = u(n-1) + K_1 e(n) + K_2 e(n-1) + K_3 e(n-2) \tag{38}$$

Trapezoidal Approximation

Another method for converting the analog form of the PID algorithm into a discrete form is to use a trapezoidal approximation, sometimes referred to as the Tustin approximation.

The PID is again given as

$$u(t) = K_p e(t) + K_i \int e dt + K_d \frac{de}{dt} \tag{39}$$

where

$$e(t) = \frac{de}{dt} \tag{40}$$

The Laplace transform of that is expressed as

$$U(s) = K_p E(s) + K_i E(s)/s + K_d s E(s) \tag{41}$$

Combining gives

$$U(s) = (K_p + K_d s + K_i/s) E(s) \tag{42}$$

Conversion into discrete form requires a transfer from s-domain to z-domain by using the Tustin approximation,

$$s = \frac{2}{T} \frac{1 - z^{-1}}{1 + z^{-1}} \tag{43}$$

where T = sampling period.

Substituting (43) in (42) gives

$$\frac{U(z)}{E(z)} = \frac{(2K_pT + 4K_d + T^2K_i) + (2T^2K_i - 8K_d)z^{-1} + (4K_d - 2TK_p + T^2 K_i)z^{-2}}{2T(1 - z^{-2})} \tag{44}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Further computation yields

$$U(z) - U(z)z^{-2} = K_1E(z) + K_2E(z)z^{-1} + K_3E(z)z^{-2} \tag{45}$$

where

$K_1 = K_p + 2K_d/T + K_iT/2$
$K_2 = K_iT - 4K_d/T$
$K_3 = K_iT/2 + 2K_d/T - K_p$

In (45), z^{-1} represents a delay of one sample time. Taking an inverse z-transform of (45) gives

$$u(n) - u(n-2) = K_1e(n) + K_2e(n-1) + K_3e(n-2)$$
or
$$u(n) = u(n-2) + K_1e(n) + K_2e(n-1) + K_3e(n-2) \tag{46}$$

This is the final form of the PID controller. However, before implementing the controller, the constants K_p, K_i, and K_d must be located. Alternatively, constants K_1, K_2, and K_3 have to be determined. These constants can be found by locating the poles of the equivalent system transfer function (i.e., controller + plant).

The transfer function for the controller can be stated as

$$G_c(z) = \frac{K_1 + K_2z^{-1} + K_3z^{-2}}{1 - z^{-2}} \tag{47}$$

The transfer function of the plant is given by

$$G_p(z) = \frac{0.2694z^{-1} + 0.2693z^{-2}}{1 - 1.999z^{-1} + 0.999z^{-2}} \times K_m \tag{48}$$

The overall system transfer function is expressed as

$$G_s(z) = \frac{G_p(z)G_c(z)}{1 + G_p(z)G_c(z)} \tag{49}$$

The denominator of the system transfer function provides the poles of the overall system. The stability and robustness of the system depends upon the location of these poles in the z-domain. Assuming pole locations of 0.96, 0.95, 0.20 and 0.15, a desired characteristic equation for the denominator is obtained. Program 4 for PC-Matlab, in Appendix D, lists the steps to obtain the characteristics equation from pole locations and obtain values of K1, K_2 and K_3. To solve for values of K_1, K_2, K_3, the coefficients of powers of z for the denominator of the system transfer function are compared with the desired polynomial.

Solving for K_1, K_2, and K_3 gives
$K_1 = 1.4795$,
$K_2 = -2.845$,
$K_3 = 1.3636$

The controller is

$$u(n) = u(n-2) + 1.4795e(n) - 2.8405e(n-1) + 1.3636e(n-2) \tag{50}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

19-0175; Rev 3; 5/96

MAXIM

±15kV ESD-Protected, +5V RS-232 Transceivers

General Description

The MAX202E–MAX213E, MAX232E/MAX241E line drivers/receivers are designed for RS-232 and V.28 communications in harsh environments. Each transmitter output and receiver input is protected against ±15kV electrostatic discharge (ESD) shocks, without latchup. The various combinations of features are outlined in the *Selection Guide*. The drivers and receivers for all ten devices meet all EIA/TIA-232E and CCITT V.28 specifications at data rates up to 120kbps, when loaded in accordance with the EIA/TIA-232E specification.

The MAX211E/MAX213E/MAX241E are available in 28-pin SO packages, as well as a 28-pin SSOP that uses 60% less board space. The MAX202E/MAX232E come in 16-pin narrow SO, wide SO, and DIP packages. The MAX203E comes in a 20-pin DIP/SO package, and needs no external charge-pump capacitors. The MAX205E comes in a 24-pin wide DIP package, and also eliminates external charge-pump capacitors. The MAX206E/MAX207E/MAX208E come in 24-pin SO, SSOP, and narrow DIP packages. The MAX232E/MAX241E operate with four 1µF capacitors, while the MAX202E/MAX206E/MAX207E/MAX208E/MAX211E/MAX213E operate with four 0.1µF capacitors, further reducing cost and board space.

Applications

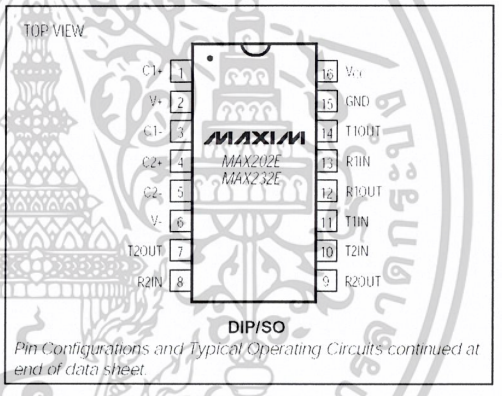
Notebook, Subnotebook, and Palmtop Computers
Battery-Powered Equipment
Hand-Held Equipment

Ordering Information appears at end of data sheet.

Features

- ♦ **ESD Protection for RS-232 I/O Pins:**
  ±15kV—Human Body Model
  ±8kV—IEC1000-4-2, Contact Discharge
  ±15kV—IEC1000-4-2, Air-Gap Discharge
- ♦ **Latchup Free (unlike bipolar equivalents)**
- ♦ **Guaranteed 120kbps Data Rate—LapLink™ Compatible**
- ♦ **Guaranteed 3V/µs Min Slew Rate**
- ♦ **Operate from a Single +5V Power Supply**

Pin Configurations



DIP/SO

Pin Configurations and Typical Operating Circuits continued at end of data sheet.

Selection Guide

PART	No. of RS-232 DRIVERS	No. of RS-232 RECEIVERS	RECEIVERS ACTIVE IN SHUTDOWN	No. of EXTERNAL CAPACITORS	LOW-POWER SHUTDOWN	TTL THREE-STATE
MAX202E	2	2	0	4 (0.1µF)	No	No
MAX203E	2	2	0	None	No	No
MAX205E	5	5	0	None	Yes	Yes
MAX206E	4	3	0	4 (0.1µF)	Yes	Yes
MAX207E	5	3	0	4 (0.1µF)	No	No
MAX208E	4	4	0	4 (0.1µF)	No	No
MAX211E	4	5	0	4 (0.1µF)	Yes	Yes
MAX213E	4	5	2	4 (0.1µF)	Yes	Yes
MAX232E	2	2	0	4 (1µF)	No	No
MAX241E	4	5	0	4 (1µF)	Yes	Yes

LapLink is a registered trademark of Traveling Software, Inc.

For free samples & the latest literature: http://www.maxim-ic.com, or phone 1-800-998-8800

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

±15kV ESD-Protected, +5V RS-232 Transceivers

ABSOLUTE MAXIMUM RATINGS

V_{CC} ..... -0.3V to +6V
V+ ..... (V_{CC} - 0.3V) to +14V
V- ..... -14V to +0.3V
Input Voltages
T_IN ..... -0.3V to (V+ + 0.3V)
R_IN ..... ±30V
Output Voltages
T_OUT ..... (V- - 0.3V) to (V+ + 0.3V)
R_OUT ..... -0.3V to (V_{CC} + 0.3V)
Short-Circuit Duration, T_OUT ..... Continuous
Continuous Power Dissipation (T_A = +70°C)
16-Pin Plastic DIP (derate 10.53mW/°C above +70°C) 842mW
16-Pin Narrow SO (derate 8.70mW/°C above +70°C) 696mW
16-Pin Wide SO (derate 9.52mW/°C above +70°C) 762mW
20-Pin Plastic DIP (derate 11.11mW/°C above +70°C) 889mW
20-Pin SO (derate 10.00mW/°C above +70°C) 800mW
24-Pin Narrow Plastic DIP
(derate 13.33mW/°C above +70°C) 1.07W
24-Pin Wide Plastic DIP
(derate 14.29mW/°C above +70°C) 1.14W
24-Pin SO (derate 11.76mW/°C above +70°C) 941mW
24-Pin SSOP (derate 8.00mW/°C above +70°C) 640mW
28-Pin SO (derate 12.50mW/°C above +70°C) 1W
28-Pin SSOP (derate 9.52mW/°C above +70°C) 762mW
Operating Temperature Ranges
MAX2_ _EC_ _ ..... 0°C to +70°C
MAX2_ _EE_ _ ..... -40°C to +85°C
Storage Temperature Range ..... -65°C to +165°C
Lead Temperature (soldering, 10sec) ..... +300°C

Stresses beyond those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. These are stress ratings only, and functional operation of the device at these or any other conditions beyond those indicated in the operational sections of the specifications is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

ELECTRICAL CHARACTERISTICS

(V_{CC} = +5V ±10% for MAX202E/206E/208E/211E/213E/232E/241E; V_{CC} = +5V ±5% for MAX203E/205E/207E; C1–C4 = 0.1µF for MAX202E/206E/207E/208E/211E/213E; C1–C4 = 1µF for MAX232E/241E; T_A = T_{MIN} to T_{MAX}; unless otherwise noted. Typical values are at T_A = +25°C.)

PARAMETER	SYMBOL	CONDITIONS		MIN	TYP	MAX	UNITS
DC CHARACTERISTICS							
V_{CC} Supply Current	I_{CC}	No load, T_A = +25°C	MAX202E/203E		8	15	mA
			MAX205E–208E		11	20	
			MAX211E/213E		14	20	
			MAX232E		5	10	
			MAX241E		7	15	
Shutdown Supply Current		T_A = +25°C, Figure 1	MAX205E/206E		1	10	µA
			MAX211E/241E		1	10	
			MAX213E		15	50	
LOGIC							
Input Pull-Up Current		T_IN = 0V (MAX205E–208E/211E/213E/241E)			15	200	µA
Input Leakage Current		T_IN = 0V to V_{CC} (MAX202E/203E/232E)				±10	µA
Input Threshold Low	V_{IL}	T_IN; EN, $\overline{SHDN}$ (MAX213E) or $\overline{EN}$, SHDN (MAX205E–208E/211E/241E)				0.8	V
Input Threshold High	V_{IH}	T_IN		2.0			V
		EN, $\overline{SHDN}$ (MAX213E) or $\overline{EN}$, SHDN (MAX205E–208E/211E/241E)		2.4			
Output Voltage Low	V_{OL}	R_OUT; I_{OUT} = 3.2mA (MAX202E/203E/232E) or I_{OUT} = 1.6mA (MAX205E/208E/211E/213E/241E)				0.4	V
Output Voltage High	V_{OH}	R_OUT; I_{OUT} = -1.0mA		3.5	V_{CC} - 0.4		V
Output Leakage Current		$\overline{EN}$ = V_{CC}, EN = 0V, 0V ≤ R_{OUT} ≤ V_{CC}, MAX205E–208E/211E/213E/241E outputs disabled			±0.05	±10	µA

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

±15kV ESD-Protected, +5V RS-232 Transceivers

ELECTRICAL CHARACTERISTICS (continued)

(V_{CC} = +5V ±10% for MAX202E/206E/208E/211E/213E/232E/241E; V_{CC} = +5V ±5% for MAX203E/205E/207E; C1–C4 = 0.1µF for MAX202E/206E/207E/208E/211E/213E; C1–C4 = 1µF for MAX232E/241E; T_A = T_{MIN} to T_{MAX}; unless otherwise noted. Typical values are at T_A = +25°C.)

PARAMETER	SYMBOL	CONDITIONS		MIN	TYP	MAX	UNITS
EIA/TIA-232E RECEIVER INPUTS							
Input Voltage Range				-30		30	V
Input Threshold Low		T_A = +25°C, V_{CC} = 5V	All parts, normal operation	0.8	1.2		V
			MAX213E, $\overline{SHDN}$ = 0V, EN = V_{CC}	0.6	1.5		
Input Threshold High		T_A = +25°C, V_{CC} = 5V	All parts, normal operation		1.7	2.4	V
			MAX213E (R4, R5), $\overline{SHDN}$ = 0V, EN = V_{CC}		1.5	2.4	
Input Hysteresis		V_{CC} = 5V, no hysteresis in shutdown		0.2	0.5	1.0	V
Input Resistance		T_A = +25°C, V_{CC} = 5V		3	5	7	kΩ
EIA/TIA-232E TRANSMITTER OUTPUTS							
Output Voltage Swing		All drivers loaded with 3kΩ to ground (Note 1)		±5	±9		V
Output Resistance		V_{CC} = V+ = V- = 0V, V_{OUT} = ±2V		300			Ω
Output Short-Circuit Current					±10	±60	mA
TIMING CHARACTERISTICS							
Maximum Data Rate		R_L = 3kΩ to 7kΩ, C_L = 50pF to 1000pF, one transmitter switching		120			kbps
Receiver Propagation Delay	t_{PLHR}, t_{PHLR}	C_L = 150pF	All parts, normal operation		0.5	10	µs
			MAX213E (R4, R5), $\overline{SHDN}$ = 0V, EN = V_{CC}		4	40	
Receiver Output Enable Time		MAX205E/206E/211E/213E/241E normal operation, Figure 2			600		ns
Receiver Output Disable Time		MAX205E/206E/211E/213E/241E normal operation, Figure 2			200		ns
Transmitter Propagation Delay	t_{PLHT}, t_{PHLT}	R_L = 3kΩ, C_L = 2500pF, all transmitters loaded			2		µs
Transition-Region Slew Rate		T_A = +25°C, V_{CC} = 5V, R_L = 3kΩ to 7kΩ, C_L = 50pF to 1000pF, measured from -3V to +3V or +3V to -3V, Figure 3		3	6	30	V/µs
ESD PERFORMANCE: TRANSMITTER OUTPUTS, RECEIVER INPUTS							
ESD-Protection Voltage		Human Body Model			±15		kV
		IEC1000-4-2, Contact Discharge			±8		
		IEC1000-4-2, Air-Gap Discharge			±15		

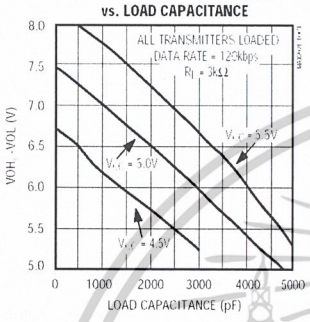
Note 1: MAX211EE_ _ tested with V_{CC} = +5V ±5%.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

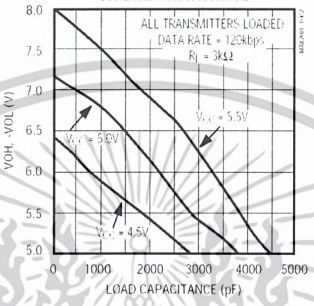
Typical Operating Characteristics

(Typical Operating Circuits, V_{CC} = +5V, T_A = +25°C, unless otherwise noted.)

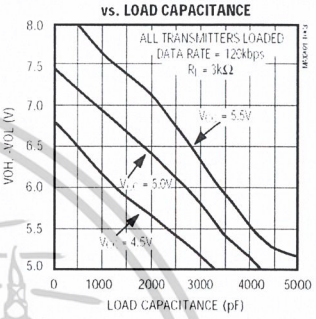
MAX232E
TRANSMITTER OUTPUT VOLTAGE
vs. LOAD CAPACITANCE



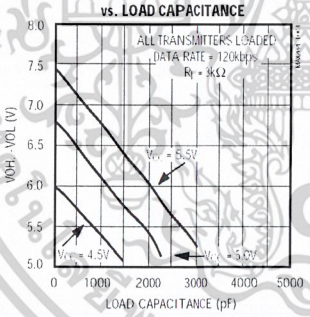
MAX202E/MAX203E
TRANSMITTER OUTPUT VOLTAGE
vs. LOAD CAPACITANCE



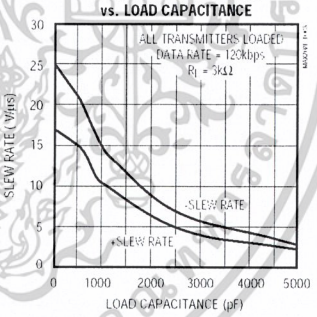
MAX241E
TRANSMITTER OUTPUT VOLTAGE
vs. LOAD CAPACITANCE



MAX211E/MAX213E
TRANSMITTER OUTPUT VOLTAGE
vs. LOAD CAPACITANCE



MAX211E/MAX213E/MAX241E
TRANSMITTER SLEW RATE
vs. LOAD CAPACITANCE



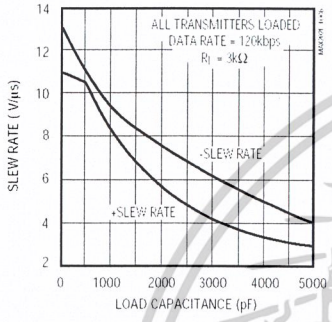
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

±15kV ESD-Protected, +5V RS-232 Transceivers

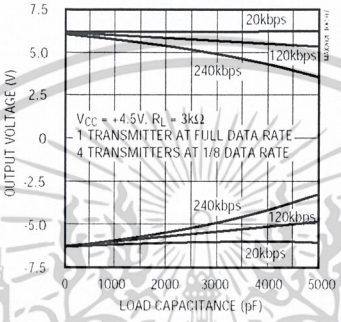
Typical Operating Characteristics (continued)

(Typical Operating Circuits. V_{CC} = +5V, T_A = +25°C, unless otherwise noted.)

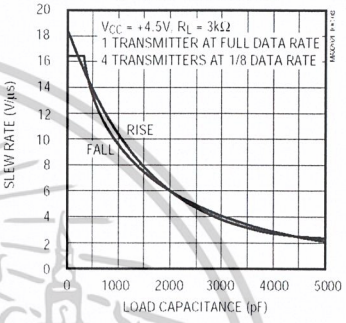
MAX202E/MAX203E/MAX232E TRANSMITTER SLEW RATE vs. LOAD CAPACITANCE



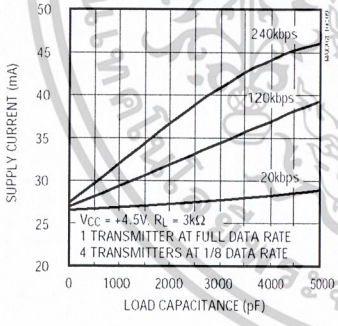
MAX205E–MAX208E TRANSMITTER OUTPUT VOLTAGE vs. LOAD CAPACITANCE



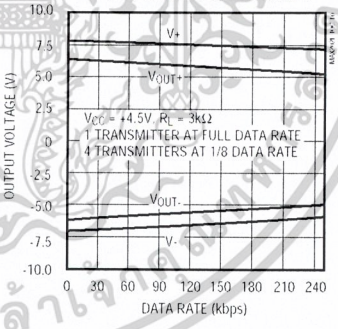
MAX205E–MAX208E TRANSMITTER SLEW RATE vs. LOAD CAPACITANCE



MAX205E–MAX208E SUPPLY CURRENT vs. LOAD CAPACITANCE



MAX205E–MAX208E OUTPUT VOLTAGE vs. DATA RATE



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

±15kV ESD-Protected, +5V RS-232 Transceivers

Pin Descriptions

MAX202E/MAX232E

PIN		NAME	FUNCTION
DIP/SO	LCC		
1, 3	2, 4	C1+, C1-	Terminals for positive charge-pump capacitor
2	3	V+	+2V_{CC} voltage generated by the charge pump
4, 5	5, 7	C2+, C2-	Terminals for negative charge-pump capacitor
6	8	V-	-2V_{CC} voltage generated by the charge pump
7, 14	9, 18	T_OUT	RS-232 Driver Outputs
8, 13	10, 17	R_IN	RS-232 Receiver Outputs
9, 12	12, 15	R_OUT	RS-232 Receiver Outputs
10, 11	13, 14	T_IN	RS-232 Driver Inputs
15	19	GND	Ground
16	20	V_{CC}	+4.5V to +5.5V Supply-Voltage Input
—	1, 6, 11, 16	N.C.	No Connect—not internally connected

MAX203E

PIN		NAME	FUNCTION
DIP	SO		
1, 2	1, 2	T_IN	RS-232 Driver Inputs
3, 20	3, 20	R_OUT	RS-232 Receiver Outputs
4, 19	4, 19	R_IN	RS-232 Receiver Inputs
5, 18	5, 18	T_OUT	RS-232 Transmitter Outputs
6, 9	6, 9	GND	Ground
7	7	V_{CC}	+4.5V to +5.5V Supply-Voltage Input
8	13	C1+	Make no connection to this pin.
10, 16	11, 16	C2-	Connect pins together.
12, 17	10, 17	V-	-2V_{CC} voltage generated by the charge pump. Connect pins together.
13	14	C1-	Make no connection to this pin.
14	8	V+	+2V_{CC} voltage generated by the charge pump
11, 15	12, 15	C2+	Connect pins together.

MAX205E

PIN	NAME	FUNCTION
1–4, 19	T_OUT	RS-232 Driver Outputs
5, 10, 13, 18, 24	R_IN	RS-232 Receiver Inputs
6, 9, 14, 17, 23	R_OUT	TTL/CMOS Receiver Outputs. All receivers are inactive in shutdown.
7, 8, 15, 16, 22	T_IN	TTL/CMOS Driver Inputs. Internal pull-ups to V_{CC}.
11	GND	Ground
12	V_{CC}	+4.75V to +5.25V Supply Voltage
20	$\overline{EN}$	Receiver Enable—active low
21	SHDN	Shutdown Control—active high

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

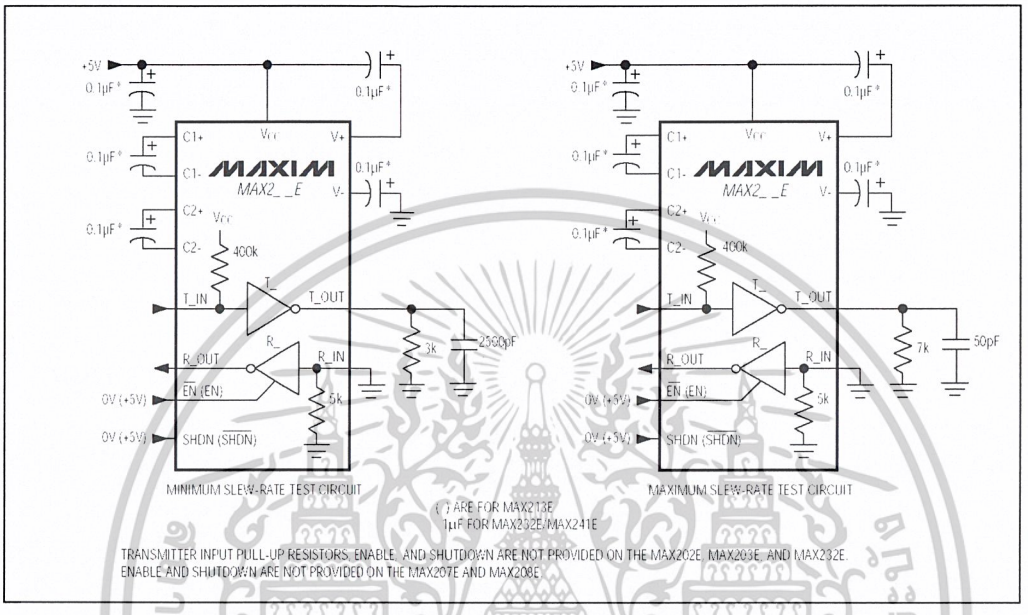


Figure 3. Transition Slew-Rate Circuit

Detailed Description

The MAX202E–MAX213E, MAX232E/MAX241E consist of three sections: charge-pump voltage converters, drivers (transmitters), and receivers. These E versions provide extra protection against ESD. They survive ±15kV discharges to the RS-232 inputs and outputs, tested using the Human Body Model. When tested according to IEC1000-4-2, they survive ±8kV contact-discharges and ±15kV air-gap discharges. The rugged E versions are intended for use in harsh environments or applications where the RS-232 connection is frequently changed (such as notebook computers). The standard (non-"E") MAX202, MAX203, MAX205–MAX208, MAX211, MAX213, MAX232, and MAX241 are recommended for applications where cost is critical.

+5V to ±10V Dual Charge-Pump Voltage Converter

The +5V to ±10V conversion is performed by dual charge-pump voltage converters (Figure 4). The first charge-pump converter uses capacitor C1 to double the +5V into +10V, storing the +10V on the output filter capacitor, C3. The second uses C2 to invert the +10V into -10V, storing the -10V on the V- output filter capacitor, C4.

In shutdown mode, V+ is internally connected to V_{CC} by a 1kΩ pull-down resistor, and V- is internally connected to ground by a 1kΩ pull-up resistor.

RS-232 Drivers

With V_{CC} = 5V, the typical driver output voltage swing is ±8V when loaded with a nominal 5kΩ RS-232 receiver. The output swing is guaranteed to meet EIA/TIA-232E and V.28 specifications that call for ±5V minimum output levels under worst-case conditions. These include a 3kΩ load, minimum V_{CC}, and maximum operating temperature. The open-circuit output voltage swings from (V+ - 0.6V) to V-.

Input thresholds are CMOS/TTL compatible. The unused drivers' inputs on the MAX205E–MAX208E, MAX211E, MAX213E, and MAX241E can be left unconnected because 400kΩ pull-up resistors to V_{CC} are included on-chip. Since all drivers invert, the pull-up resistors force the unused drivers' outputs low. The MAX202E, MAX203E, and MAX232E do not have pull-up resistors on the transmitter inputs.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ประวัติผู้แต่ง



ชื่อ-สกุล	นายทรงพล มงคลจีรวัฒน์
วัน เดือน ปีเกิด	29 มีนาคม พ.ศ. 2525
ภูมิลำเนา	59 หมู่ 8 ตำบลนาป่า อำเภอเมือง
	จังหวัดชลบุรี 20000 โทรศัพท์ 038-274390 ต่อ 5245

ประวัติการศึกษา

ประถมศึกษา	โรงเรียนวัดใหญ่อินทาราม จังหวัดชลบุรี
มัธยมศึกษาตอนต้น	โรงเรียนชลราษฎรอำรุง จังหวัดชลบุรี
ประกาศนียบัตรวิชาชีพ	วิทยาลัยเทคนิคสัตหีบ
ประกาศนียบัตรวิชาชีพชั้นสูง	วิทยาลัยเทคนิคสัตหีบ
ปริญญาตรี	สาขาวิชาเทคโนโลยีการวัดคุมทางอุตสาหกรรม
	ภาควิชาครุศาสตร์วิศวกรรม
	คณะครุศาสตร์อุตสาหกรรม สจล.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ประวัติผู้แต่ง



ชื่อ-สกุล	นายธนานุวัฒน์ สุวรรณรุ่งเรือง
วัน เดือน ปีเกิด	29 กันยายน พ.ศ. 2524
ภูมิลำเนา	61/1 หมู่ 5 ตำบลบ้านค่าย อำเภอบ้านค่าย จังหวัดระยอง 21120 โทรศัพท์ 038-641153

ประวัติการศึกษา

ประถมศึกษา	โรงเรียนอนุบาลระยอง จังหวัดระยอง
มัธยมศึกษาตอนต้น	โรงเรียนบ้านค่าย จังหวัดระยอง
ประกาศนียบัตรวิชาชีพ	วิทยาลัยเทคนิคระยอง
ประกาศนียบัตรวิชาชีพชั้นสูง	วิทยาลัยเทคนิคระยอง
ปริญญาตรี	สาขาวิชาเทคโนโลยีการวัดคุมทางอุตสาหกรรม ภาควิชาครุศาสตร์วิศวกรรม คณะครุศาสตร์อุตสาหกรรม สจล.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ประวัติผู้แต่ง



ชื่อ-สกุล นางสาววราภรณ์ สุภาพละ

วัน เดือน ปีเกิด 23 เมษายน พ.ศ. 2525

ภูมิลำเนา 19/56 หมู่ 8 ตำบลหนองบัว อำเภอบ้านค่าย
จังหวัดระยอง 21120 โทรศัพท์ -

ประวัติการศึกษา

ประถมศึกษา โรงเรียนบ้านคลองขนุน จังหวัดระยอง

มัธยมศึกษาตอนต้น โรงเรียนบ้านค่าย จังหวัดระยอง

ประกาศนียบัตรวิชาชีพ วิทยาลัยเทคนิคระยอง

ประกาศนียบัตรวิชาชีพชั้นสูง วิทยาลัยเทคนิคระยอง

ปริญญาตรี สาขาวิชาเทคโนโลยีการวัดคุมทางอุตสาหกรรม
ภาควิชาครุศาสตร์วิศวกรรม
คณะครุศาสตร์อุตสาหกรรม สจล.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ประวัติผู้แต่ง



ชื่อ-สกุล นายสุธา ทรัพย์อนันต์กุล

วัน เดือน ปีเกิด 18 มีนาคม พ.ศ. 2525

ภูมิลำเนา 81/45 ซอยประชาราษฎร์ 14 หมู่ 3 ถ. ประชาราษฎร์
ตำบลตลาดขวัญ อำเภอเมือง
จังหวัดนนทบุรี 11000 โทรศัพท์ 02-5263910

ประวัติการศึกษา

ประถมศึกษา โรงเรียนอนุบาลนนทบุรี จังหวัดนนทบุรี

มัธยมศึกษาตอนต้น โรงเรียนรัตนาธิเบศร์ จังหวัดนนทบุรี

มัธยมศึกษาตอนปลาย โรงเรียนรัตนาธิเบศร์ จังหวัดนนทบุรี

ประกาศนียบัตรวิชาชีพชั้นสูง สถาบันเทคโนโลยีราชมงคล วิทยาเขตนนทบุรี

ปริญญาตรี สาขาวิชาเทคโนโลยีการวัดคุมทางอุตสาหกรรม
ภาควิชาครุศาสตร์วิศวกรรม
คณะครุศาสตร์อุตสาหกรรม สจล.

คติพจน์ ทำวันนี้ให้ดีที่สุด เพื่อวันหน้าที่ดีกว่า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้