

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

ระบบควบคุมอุณหภูมิโดยใช้ฟัซซี่ลอจิก

TEMPERATURE CONTROL SYSTEM BY FUZZY LOGIC



นาย สมชัย เทียนสุคนธ์

อาจารย์ที่ปรึกษา

อาจารย์ เจริญ วงษ์ขุ่มเย็น

รฟ.

๘๒๓๘ ร

๐๕๔๗

เลขหมู่.....

เลขทะเบียน.....**61720**

วัน,เดือน,ปี **2 1 ก.ค. 2549**

b.....11603008
i.....

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

ภาควิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้ภายในห้องสมุดเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์ปีการศึกษา 2547

ภาควิชา วิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง ระบบควบคุมอุณหภูมิโดยใช้ฟัซซี่ลอจิก

TEMPERATURE CONTROL SYSTEM BY FUZZY LOGIC

ผู้จัดทำ

1. นาย สมชัย เทียนสุคนธ์ รหัสประจำตัว 44010499



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระบบควบคุมอุณหภูมิโดยใช้พีซีลอจิก

นาย สมชัย เทียนสุคนธ์ 44010499

อาจารย์ เจริญ วงษ์ชุ่มเย็น

ปีการศึกษา 2547

บทคัดย่อ

โครงการระบบควบคุมอุณหภูมิโดยใช้พีซีลอจิก จัดทำขึ้นเพื่อเพิ่มความสามารถในการควบคุมระบบต่างๆ ใกล้เคียงความต้องการของตัวมนุษย์เองมากที่สุด โดยเป็นระบบที่มองและคาดเดาการทำงานได้เหมือนกับมนุษย์เองเพื่อเพิ่มประสิทธิภาพของระบบและยืดอายุการทำงานของอุปกรณ์ โดยใช้ไมโครคอนโทรลเลอร์(microcontroller) ตระกูล MCS-51 เบอร์ AT89C51 ในการควบคุมค่าต่างๆ ลักษณะของการควบคุมเป็นการตรวจสอบอุณหภูมิจากเซนเซอร์ แล้วนำค่าที่ได้มาเป็น อินพุตที่ใช้ควบคุม ฮีตเตอร์ และพัดลม การควบคุมโดยทั่วไปจะควบคุมจากอินพุตที่เป็นอุณหภูมิและ อินเทอร์พท์ที่มาจากภายนอกที่เป็นตัวอ้างอิงสัญญาณ

โครงการนี้ได้ใช้ตู้ฟักไข่เป็นตัวสาธิตการทำงาน โดยมีการควบคุม 2 แบบคือ อุณหภูมิเฉลี่ยทั่วทั้งตู้ และ อุณหภูมิเฉพาะจุดซึ่งได้ตั้งเอาไว้ถึง 5 จุดด้วยกัน

TEMPERATURE CONTROL SYSTEM BY FUZZY LOGIC

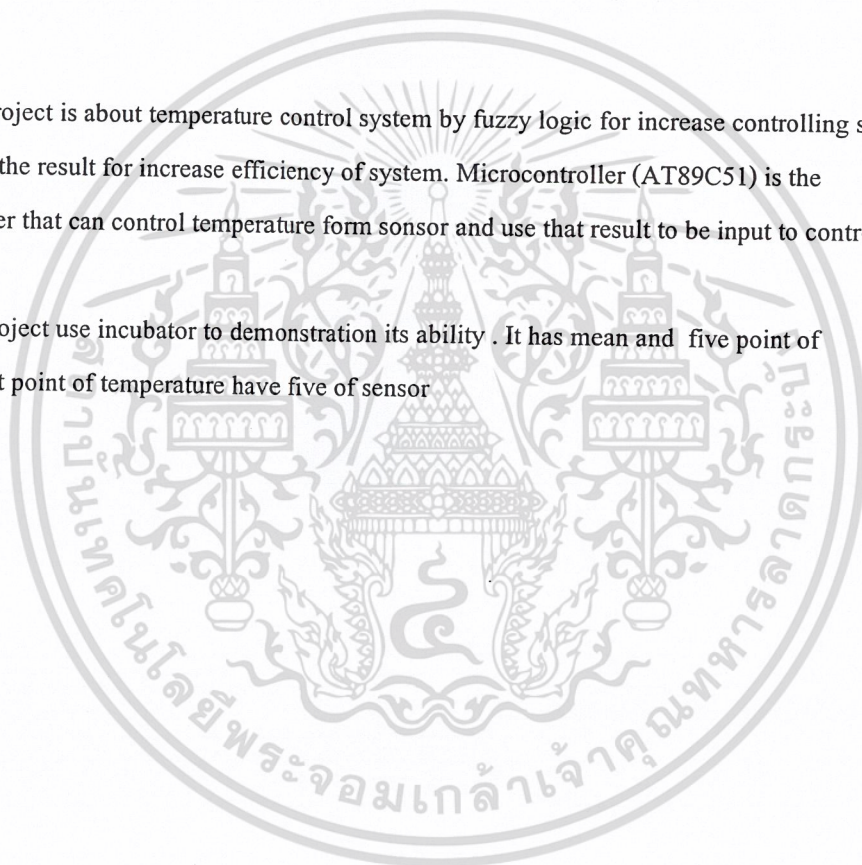
Somchai Teansukont

Charoen Vongchumyen Advisor

ABTRACT

This project is about temperature control system by fuzzy logic for increase controlling system ability. It guess the result for increase efficiency of system. Microcontroller (AT89C51) is the controlling center that can control temperature form sensor and use that result to be input to control heater and fan

This project use incubator to demonstration its ability . It has mean and five point of temperature that point of temperature have five of sensor



กิตติกรรมประกาศ

วิทยานิพนธ์ฉบับนี้คงไม่อาจเสร็จได้ด้วยดี หากไม่ได้รับความช่วยเหลือ และร่วมมือจากหลายๆ ฝ่ายด้วยกัน บุคคลแรกที่ต้องกล่าวถึงเพราะเป็นส่วนสำคัญที่ทำให้วิทยานิพนธ์นี้เสร็จลงได้ก็คืออาจารย์ เจริญ วงษ์หุมนเย็น อาจารย์ที่ปรึกษาวิทยานิพนธ์ ที่ให้ความเอาใจใส่ แนะนำ และช่วยเหลือเสมอมา ซึ่งต้องขอขอบพระคุณเป็นอย่างมาก

ต้องขอขอบพระคุณบุคคลสำคัญที่สุดที่ทำให้ข้าพเจ้ามีวันนี้ ก็คือ บิดา มารดา อันเป็นที่เคารพรักยิ่ง ซึ่งได้เลี้ยงดูผู้เขียนมาเป็นอย่างดี พร้อมทั้งให้โอกาสในการศึกษาอย่างเต็มที่ และยังให้กำลังใจ เอาใจใส่เสมอมา ในทุก ๆ ด้านอันหาที่เปรียบมิได้ ข้าพเจ้าขอระลึกในพระคุณอันสุดประมาณและขอกราบขอบพระคุณมา ณ ที่นี้

สมชัย เทียนสุคนธ์



สารบัญ

	หน้าที่
เรื่อง	I
บทคัดย่อ	II
ABSTRACT	III
กิตติกรรมประกาศ	IV
บทที่ 1 บทนำ	1
1.1 ความสำคัญและที่มา	1
1.2 จุดประสงค์ของงานวิจัย	1
1.3 ขอบเขตของงานวิจัย	1
1.4 วิธีการดำเนินงาน	2
บทที่ 2 ทฤษฎีทางคณิตศาสตร์เกี่ยวกับฟัซซี	3
2.1 ฟัซซีลอจิก (Fuzzy logic)	3
2.1.1 ทฤษฎีฟัซซีลอจิก	3
2.1.2 การหาข้อสรุปตามหลักการเหตุผลโดยใช้ทฤษฎีฟัซซีลอจิก	5
2.2 ความสัมพันธ์ฟัซซี (Fuzzy Relation)	7
2.3 การคอมโพสิชันแบบฟัซซี (Fuzzy Composition)	7
2.4 การอินเฟอร์เรนซ์แบบฟัซซี (Fuzzy Inference)	8
2.5 การฟัซซิฟิเคชัน (Fuzzification)	9
2.6 การดีฟัซซิฟิเคชัน (Defuzzification)	9
2.7 ตัวควบคุมฟัซซี	10
2.7.1 หน่วยฟัซซิฟิเคชัน	10
2.7.2 ฐานกฎการควบคุมของฟัซซี	10
2.7.3 หน่วยอินเฟอร์เรนซ์	11
2.7.4 หน่วยดีฟัซซิฟิเคชัน	11
2.8 รูปแบบการติดต่อสื่อสาร	11
2.9 ไมโครคอนโทรลเลอร์และระบบควบคุม	13
2.9.1 ทฤษฎีของไมโครคอนโทรลเลอร์	13
2.9.2 ไตรแอก	14
2.9.3 ออปแอมป์	15
2.9.4 ออปโตคัปเปิลเลอร์ (Opto-Coupler)	16
2.9.5 เซนเซอร์ DS1820	16
2.9.6 ไครมोटอร์กระแสตรง	17

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3	การออกแบบตัวควบคุมด้วยFuzzy	19
3.1	การกำหนดจำนวนตัวแปรของตัวควบคุม	19
3.2	การนอมอลไลซ์	19
3.3	การกำหนดค่าเกน(Gain)ของตัวแปรอินพุตและเอาต์พุต	20
3.4	การเลือกวิธีและระดับควอนไทซ์(Quantize)	20
3.5	ฟัซซิฟิเคชัน(Fuzzification)	20
3.6	การกำหนดฟังก์ชันการเป็นสมาชิก(Membership function)	21
3.7	ฐานกฎ	24
3.8	ส่วนของอินเฟอร์เรนซ์	24
3.9	แนวทางการออกแบบกฎ	24
3.10	การเลือกวิธีการดีฟัซซิฟิเคชัน	27
3.11	การดีนอมอลไลซ์ (Denormalize)	28
3.12	การออกแบบการทำงานของระบบควบคุมฮาร์ดแวร์	29
3.13	การออกแบบการทำงานของระบบควบคุมพัลลัม	29
3.14	การออกแบบฮาร์ดแวร์ และ ซอฟต์แวร์	33
3.14.1	ส่วนควบคุม	33
3.14.2	ส่วนติดต่อพอร์ตอนุกรม	33
3.14.3	ส่วนแยกกราวด์ของวงจร	34
3.14.4	ส่วนไดรเวอร์	34
3.14.5	วงจรเปรียบเทียบแรงดัน	35
3.14.6	วงจรแหล่งจ่ายไฟ	36
3.14.7	วงจรควบคุมพัลลัม	36
3.14.8	เซนเซอร์	37
บทที่ 4	การทดลองและการทดสอบระบบ	25
4.1	การทดลองการทำงาน	42
4.2	ขั้นตอนการติดตั้งระบบ	43
4.3	การทดลองระบบ	44
บทที่ 5	บทวิจารณ์และสรุป	50
5.1	สรุปและวิจารณ์	50
5.2	ปัญหาที่เกิดขึ้น	51
5.3	แนวทางในการพัฒนาต่อไป	51

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป

รูป	หน้า
รูปที่ 2.1 แสดงการคอมโพสิชันของความสัมพันธ์ฟิซซี	8
รูปที่ 2.2 แสดงการหาข้อสรุปจากหลักฐาน โดยใช้เงื่อนไขความสัมพันธ์ระหว่างเหตุและผล	9
รูปที่ 2.3 แสดงโครงสร้างพื้นฐานของตัวควบคุมฟิซซี	10
รูปที่ 2-4 การทำงานในรูปแบบ 1-wire bus	12
รูปที่ 2-5 การทำงานของการรีเซต และรอการตอบกลับ	12
รูปที่ 2-6 การอ่านและเขียนข้อมูลของไมโครคอนโทรลเลอร์ และ เซนเซอร์	13
รูปที่ 2-7 สัญลักษณ์และวงจรพื้นฐานของไตรแอก	13
รูปที่ 2-8 การเปลี่ยนแปลงค่าของกำลังไฟฟ้าที่ป้อนให้แก่โหลด โดยกำหนดได้จากตำแหน่งเวลาของการทริกที่ให้แก่ไตรแอก	14
รูปที่ 2-9 การทำงานโดยรวมของออปแอมป์	15
รูปที่ 2-10 วงจรเปรียบเทียบแรงดัน 16	
รูปที่ 2-11 วงจรออปโตคัปเปิลอร์พื้นฐาน	16
รูปที่ 2-12 การทำงานภายในของ DS1820	17
รูปที่ 2.13 วงจรไครมोटอร์กระแสตรงพื้นฐาน	18
รูปที่ 3.1 แสดง โครงสร้างของตัวควบคุม	19
รูปที่ 3.2 ฟังก์ชันสามเหลี่ยมซึ่งเป็นฟังก์ชันการเป็นสมาชิกของฟิซซีเซต	21
รูปที่ 3.3 ฟังก์ชันการเป็นสมาชิกของค่าความผิดพลาด (e)	22
รูปที่ 3.4 ฟังก์ชันการเป็นสมาชิกของค่าการเปลี่ยนแปลงของความผิดพลาด (Δe)	23
รูปที่ 3.5 ฟังก์ชันการเป็นสมาชิกของเอาต์พุต	23
รูปที่ 3.6 ผลการตอบสนองของการเปลี่ยนระดับและการแบ่งช่วงของการตอบสนองเพื่อหา ความสัมพันธ์ของค่าผิดพลาด และการเปลี่ยนแปลงค่าผิดพลาด	25
รูปที่ 3.7 แผงผังแสดงการควบคุมฮาร์ดแวร์	30
รูปที่ 3.8 แสดงการควบคุมพัลลวม	31
รูปที่ 3.9 โครงสร้างการทำงานของระบบ	33
รูปที่ 3-10 แสดงวงจรติดต่อสื่อสารแบบอนุกรม RS-232	34
รูปที่ 3-11 แสดงวงจรแยกกราวด์แรงดันต่ำกับแรงดันสูงโดยใช้ออปโตคัปเปิลอร์	34
รูปที่ 3-12 แสดงวงจรควบคุมโหลดทางเอาต์พุตโดยใช้ไตรแอก	35
รูปที่ 3-13 แสดงวงจรเปรียบเทียบแรงดันโดยใช้ออปแอมป์	35
รูปที่ 3.14 การปรับสัญญาณลอจิกสัญญาณที่จะใช้งานเป็นอินเทอร์พท์	36
รูปที่ 3-15 แสดงวงจรเรกกูเลเตอร์ 5 โวลต์ และ 12 โวลต์	36

เอกสารนี้เป็นเอกสารลิขสิทธิ์ของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ไม่อนุญาตให้นำไปใช้ประโยชน์ในการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 3.16 วงจรพัลลวม	37
รูปที่ 3.17การทำงานของDS1820 ในส่วนของRom Code	39
รูปที่ 3.18 การทำงานในการอ่านค่าอุณหภูมิ	40
รูปที่ 3.19 การต่อเซนเซอร์เป็นบัสแบบ 1 –wire bus	41
รูปที่ 3.20 วงจร โดยรวมของระบบควบคุม	41
รูปที่ 4.1 รอมโค้ดที่สามารถอ่านได้จากอุปกรณ์	42
รูปที่ 4.2 การทดสอบการอ่านค่าของเซนเซอร์	42
รูปที่ 4.3 ทดสอบการทำงานของพัลลวมเมื่อกดเลขต่างๆ	42
รูปที่ 4.4 การทดสอบการทำงานของฮีตเตอร์โดยใช้หลอดไฟ	43
รูปที่ 4.5 แสดงการวางตำแหน่งของเซนเซอร์ทั้ง 5 จุด	43
รูปที่ 4.6 แสดงการเชื่อมต่อวงจรสำหรับควบคุม ฮีตเตอร์	44
รูปที่ 4.7 แสดงการติดตั้งพัลลวม	44
รูปที่ 4.8 ทดสอบการทำงานของฮีตเตอร์เมื่ออุณหภูมิต่ำกว่าจุดอ้างอิงมาก	45
รูปที่ 4.9 ทดสอบการทำงานของฮีตเตอร์เมื่ออุณหภูมิต่ำกว่าจุดอ้างอิงไม่มาก	45
รูปที่ 4.10 ทดสอบการทำงานของฮีตเตอร์เมื่ออุณหภูมิอยู่ที่จุดอ้างอิง	46
รูปที่ 4.11 ทดสอบการทำงานของฮีตเตอร์เมื่ออุณหภูมิสูงกว่าจุดอ้างอิงไม่มาก	46
รูปที่ 4.12 ทดสอบการทำงานของฮีตเตอร์เมื่ออุณหภูมิสูงกว่าจุดอ้างอิงมาก	47
รูปที่ 4.13 ตั้งค่าอุณหภูมิที่ต้องการควบคุม	47
รูปที่ 4.14 แสดงการตั้งค่าใหม่ เมื่อกด “r”	47
รูปที่ 4.15 แสดงอุณหภูมิณจุดต่างๆ ค่าอ้างอิง ระดับของไฟ และสถานการณ์ทำงานของพัลลวม	48
รูปที่ 4.16 แสดงการทำงานของพัลลวมเมื่อ อุณหภูมิต่างกันมากกว่า 1 องศาเซลเซียส	48
รูปที่ 4.17 แสดงการปรับระดับของฮีตเตอร์เมื่ออุณหภูมิเข้าใกล้ค่าที่ตั้งไว้	48
รูปที่ 4.18 แสดงการทำงานของพัลลวมทั้ง 2 ตัวเมื่ออุณหภูมิมากกว่าค่าอ้างอิงมากๆ	49

สารบัญตาราง

ตารางที่ 2.1 แสดงตารางความจริงสำหรับตัวปฏิบัติการพีชชีลอจิก ตามนิยามของ Zedeh	4
ตารางที่ 2.2 แสดงตารางความจริงสำหรับตัวปฏิบัติการพีชชีลอจิกตามนิยามของ Baldwin	5
ตารางที่ 2.3 แสดงผลการหาข้อสรุปตามหลักการและเหตุผลโดยใช้ GMP ตามหลักการของ Fukami	7
ตารางที่ 2.4 แสดงผลการหาข้อสรุปตามหลักการและเหตุผลโดยใช้ GMT ตามหลักการของ Fukami	7
ตารางที่ 2.5 แสดงการจ่ายไฟทำงานของมอเตอร์	18
ตารางที่ 3.1 ความสัมพันธ์ของตัวแปรอินพุตและเอาต์พุตของการออกแบบกฎ	25
ตารางที่ 3.2 ช่วงความเป็นสมาชิกในแต่ละ ลิงวิสติกของอินพุตและเอาต์พุต	28
ตารางที่ 3.3 แสดงการควบคุม	29
ตารางที่ 5.1 เปรียบเทียบระบบเดิมกับระบบใหม่	50

บทที่ 1

บทนำ

1.1 ความสำคัญและที่มา

ในปัจจุบัน ในสถานที่หลายๆ แห่งมีความจำเป็นต้องควบคุมสภาพแวดล้อมให้อยู่ในสภาวะคงที่มากที่สุด เช่น เรือนเพาะเลี้ยงอนุบาลสัตว์ โรงเรือนเพาะปลูกพืช ห้องเพาะเชื้อ ฯลฯ โดยจะมีทั้งการควบคุม อุณหภูมิ ความชื้น แสง เป็นต้น เพื่อให้ได้สภาพแวดล้อมตรงตามที่คำนวณเอาไว้เพื่อให้ได้ผลผลิตสูงสุด

ในการควบคุมสภาพแวดล้อมโดยปกติได้มีการใช้อุปกรณ์บางตัวในการควบคุม แต่พบว่าจะไม่สามารถตอบสนองความต้องการหรือควบคุมได้อย่างเต็มที่ เช่น การใช้เทอร์โมสแตต(Thermo state) ในการตัดไฟของฮีตเตอร์(Heater) เมื่ออุณหภูมิเกินค่าที่ตั้งไว้ และเริ่มทำงานเมื่ออุณหภูมิต่ำกว่าที่ตั้งไว้ ซึ่งการใช้เทอร์โมสแตตนี้จะทำให้อุณหภูมิมีการแกว่งมากเกินไป ซึ่งทำให้มีแนวคิดเรื่องการใช้การควบคุมที่ใกล้เคียงกับมนุษย์มาใช้ในการควบคุม

ในบางระบบอุปกรณ์บางชนิดมีความเลื้อยสูง การควบคุมด้วยระบบธรรมดามีความลำบาก อุปกรณ์ที่จะควบคุมจะมีการขึ้น-ลงสูงกว่าจุดที่ตั้งไว้มาก จึงมีแนวคิดที่จะใช้ฟัซซี(Fuzzy) มาใช้ในการควบคุมอุปกรณ์และสภาพแวดล้อม โฟฟัซซีจะเป็นการใช้อินพุตที่มี 1 ค่าหรือ มากกว่า มาเข้ากฎต่างๆ ที่ตั้งเอาไว้เพื่อควบคุมให้ได้เอาต์พุตแบบต่างๆ ที่มีความสามารถในการควบคุมระบบหรือสภาพแวดล้อมให้ตรงหรือมีความใกล้เคียงกับที่ต้องการมากที่สุด

1.2 จุดประสงค์ของงานวิจัย

1.2.1 ศึกษาฟัซซีลอจิก(Fuzzy Logic)เพื่อใช้ในการควบคุมสภาพแวดล้อมเพื่อนำไปใช้งานในรูปแบบต่างๆ

1.2.2 ควบคุมสภาพแวดล้อมที่ต้องการศึกษาอยู่ในช่วงที่ต้องการ ไม่แกว่งออกจากจุดที่ตั้งเอาไว้มากเกินไป

1.2.3 เพิ่มประสิทธิภาพของระบบให้มีความสามารถในการทำงานได้มากยิ่งขึ้น

1.2.4 ยึดอายุการใช้งานของอุปกรณ์

1.3 ขอบเขตของงานวิจัย

ในการใช้ระบบฟัซซีสำหรับควบคุมระบบการทำงานนั้น เราสามารถนำมาใช้ได้หลากหลายแต่ที่จะศึกษาจะเป็นการควบคุมอุณหภูมิ โดยได้ใช้ตู้ฟักไข่เป็นเครื่องสาธิต(Demonstration) โดยปกติตู้ฟักไข่จะใช้เทอร์โมสแตตในการควบคุมการปิด-เปิดฮีตเตอร์ ซึ่งเราจะเอาเทอร์โมสแตตออกและนำระบบของเราเข้าไปแทน

ตัวควบคุมที่เราสนใจมีอยู่ 2 อย่างคือ อุณหภูมิของตัวถังตู้ และอุณหภูมิบางจุด

ตัวแปรที่เป็นอุณหภูมิตัวถังจะแบ่งเป็น 2 แบบคือ อินพุตที่เกิดจากการนำเอาอุณหภูมิมาหาค่าความผิดพลาด e และ ค่าอินพุตที่เกิดจากการเปลี่ยนแปลงค่าความผิดพลาดในแต่ละช่วงเวลา Δe ใช้ในการควบคุม ฮีตเตอร์ให้เพิ่มหรือลด

อุณหภูมิบางจุด จะเป็นอินพุตที่เกิดจากการที่แต่ละตำแหน่งมีค่าไม่เท่ากันเพราะสถานการณ์ต่างๆ เช่น การหมุนของอากาศที่ไม่ดีพอ ใช้ในการควบคุมพัดลมให้มีการไหลเวียนจากจุดที่มีอุณหภูมิสูงไปต่ำ

1.4 วิธีการดำเนินงาน

โครงการนี้จะเริ่มด้วยการศึกษาทฤษฎีต่างๆ เกี่ยวกับงาน ซึ่งมีเรื่องหลักคือ ฟัชชีลอจิก ที่มีรายละเอียดในบทที่ 2 เรื่องการออกแบบงานต่างๆ ในบทที่ 3 การทดลองและผลการทดลองในบทที่ 4 และการสรุปผลการทดลองในบทที่ 5



บทที่ 2

ทฤษฎีทางคณิตศาสตร์เกี่ยวกับฟัซซี่

ระบบควบคุมฟัซซี่ (Fuzzy control) มีพื้นฐานแนวความคิดมาจาก ทฤษฎีฟัซซี่ลอจิก(Fuzzy logic) และทฤษฎีฟัซซี่เซต (Fuzzy set) ฟัซซี่ลอจิกมีพื้นฐานอยู่บนทฤษฎีฟัซซี่เซต ซึ่งจะช่วยให้สามารถอธิบาย ปฏิบัติการและกฎการควบคุมของระบบเป็นคำพูดได้ชัดเจนขึ้น หลักสำคัญของทฤษฎีฟัซซี่เซต คือยอมรับ สมาชิกซึ่งแตกต่างจากทฤษฎีเซตดั้งเดิมที่เน้นชัดเลยว่า เป็นสมาชิกของเซตหรือไม่เท่านั้น ไม่มีการเป็น สมาชิกของเซตเพียงบางส่วน

2.1 ฟัซซี่ลอจิก (Fuzzy logic)

2.1.1 ทฤษฎีฟัซซี่ลอจิก

ฟัซซี่ลอจิกเป็นทฤษฎีทางตรรกศาสตร์ ที่ขยายมาจากทฤษฎีตรรกศาสตร์หลายค่า (multivalued logic) โดยนิยามค่าความจริงของประพจน์ที่เป็นตัวลึงวิสกิด ได้มีการนิยามปฏิบัติการทางตรรกศาสตร์ สำหรับฟัซซี่ลอจิกไว้ต่างๆ กัน ในบทนี้จะอ้างอิงจากนิยามดั้งเดิม Zadeh ดังต่อไปนี้

นิยาม 1 ค่าความจริงของประพจน์ในทางฟัซซี่ลอจิก

กำหนดให้ $V(A)$ เป็น ฟัซซี่เซตในเอกภพสัมพัทธ์ $V = [0,1]$ ซึ่งแทนค่าความจริงของประพจน์ A จะได้ว่า

$$V(A) = \{(V_i, \mu_i) \mid i = 1, 2, \dots, n; V_i \in [0,1]\} \quad (2.1)$$

$$V(\text{not} A) = 1 - V(A) = \{(1 - V_i, \mu_A) \mid i = 1, 2, \dots, n; V_i \in [0,1]\} \quad (2.2)$$

นิยาม 2 การปฏิบัติการทางตรรกศาสตร์ในทางทฤษฎีฟัซซี่ลอจิก

$$\text{กำหนดให้ } V(A) = \{(V, \mu_A(V)) \mid \mu_A(V) \in [0,1]; V \in [0,1]\}$$

$$V(B) = \{(V, \mu_B(V)) \mid \mu_B(V) \in [0,1]; V \in [0,1]\}$$

$$\text{จะได้ว่า } V(A) \wedge V(B) = V(A \text{ and } B) = \{(V, \min\{\mu_A(V), \mu_B(V)\})\} \quad (2.3)$$

$$V(A) \vee V(B) = V(A \text{ or } B) = \{(V, \max\{\mu_A(V), \mu_B(V)\})\} \quad (2.4)$$

$$\neg V(A) = \text{not } (V(A)) = \{(V, 1 - \mu_A(V))\} \quad (2.5)$$

$$\begin{aligned} V(A) \rightarrow V(B) &= V(A \rightarrow B) = \neg V(A) \vee V(B) \\ &= \{(V, \max\{(1 - \mu_A(V)), \mu_B(V)\})\} \end{aligned} \quad (2.6)$$

ในกรณีที่

$$V(A) = \{(V_i, \alpha_i) \mid i = 1, 2, \dots, n; V_i \in [0,1], \alpha_i \in [0,1]\}$$

$$V(A) = \{(V_i, \beta_i) \mid i = 1, 2, \dots, n; V_i \in [0,1], \beta_i \in [0,1]\}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Zedeh ได้เสนอตารางค่าความจริง สำหรับการพิจารณาค่าความจริง ในการปฏิบัติการของ ฟัซซี่ลอจิก โดยขยายจากตรรกศาสตร์บูลีนมาเป็นตรรกะ 3 ค่าคือ True(T) False(F) and unknown (T+F) ได้ดังตาราง 2.1

A	B	$\wedge$	$\vee$	$\neg A$
T	T	T	T	F
T	F	F	T	F
T	T+F	T+F	T	F
F	T	F	T	T
F	F	F	F	T
F	T+F	F	T+F	T
T+F	T	T+F	T	T+F
T+F	F	F	T+F	T+F
T+F	T+F	T+F	T+F	T+F

ตารางที่ 2.1 แสดงตารางความจริงสำหรับตัวปฏิบัติการฟัซซี่ลอจิก ตามนิยามของ Zedeh

Baldwin ได้เสนอการสร้างความจริงสำหรับฟัซซี่ลอจิก โดยทำการนิยามเทอมและฟังก์ชันการเป็นสมาชิก ของตัวแปรลึงวิสตติค " Truth" ซึ่งแสดงไว้ดังนี้

Truth

$= \{(V, \mu_{\text{true}}(V) = V) \mid V \in [0,1]\}$

False

$= \{(V, \mu_{\text{false}}(V) = 1 - \mu_{\text{true}}(V)) \mid V \in [0,1]\}$

very true

$= \{(V, (\mu_{\text{true}}(V))^2) \mid V \in [0,1]\}$

very false

$= \{(V, (1 - \mu_{\text{true}}(V))^2) \mid V \in [0,1]\}$

fairly true

$= \{(V, (\mu_{\text{true}}(V))^{1/2}) \mid V \in [0,1]\}$

fairly false

$= \{(V, (1 - \mu_{\text{true}}(V))^{1/2}) \mid V \in [0,1]\}$

undecided

$= \{(V, 1) \mid V \in [0,1]\}$

absolutely true

$= \{(V, \mu_{\text{at}}(V)) \mid V \in [0,1]\}; \mu_{\text{at}}(V) = \begin{cases} 1; V = 1 \\ 0; V \neq 1 \end{cases}$

absolutely false

$= \{(V, \mu_{\text{af}}(V)) \mid V \in [0,1]\}; \mu_{\text{af}}(V) = \begin{cases} 1; V = 0 \\ 0; V \neq 0 \end{cases}$

ดังนั้น

$(\text{very})^k \text{ true} \rightarrow \text{absolutely true}$

$(\text{very})^k \text{ false} \rightarrow \text{absolutely false}$

$(\text{very})^k \text{ true} \rightarrow \text{undecided}$

$(\text{very})^k \text{ false} \rightarrow \text{undecided}$

เมื่อ $k \rightarrow \infty$

เมื่อ $k \rightarrow \infty$

เมื่อ $k \rightarrow \infty$

เมื่อ $k \rightarrow \infty$

เอกสารนี้เป็นเอกสารที่สงวนไว้เพื่อใช้ในการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

นอกจากนี้ Baldwin ได้นิยามตัวปฏิบัติการ “and”และ ”or” ด้วย minimum และ maximum ตามลำดับ และสร้างตารางค่าความจริงของการปฏิบัติการฟัซซีลอจิกได้ผลลัพธ์ดังตารางที่ 2.2

V(P)	V(Q)	V(P and Q)	V(P or Q)
false	false	false	false
true	false	false	true
true	true	true	true
undecided	false	false	undecided
undecided	true	undecided	true
undecided	undecided	undecided	undecided
true	very true	true	very true
true	fairly true	fairly true	true

ตารางที่ 2.2 แสดงตารางความจริงสำหรับตัวปฏิบัติการฟัซซีลอจิกตามนิยามของ Baldwin

2.1.2 การหาข้อสรุปตามหลักการเหตุผลโดยใช้ทฤษฎีฟัซซีลอจิก

การหาข้อสรุปตามหลักการเหตุผล โดยใช้ทฤษฎีฟัซซีลอจิก จะเป็นการขยายนิยามของสัจนิรันดร์ที่ใช้ในตรรกศาสตร์แบบเดิม เพื่อใช้หาข้อมูลตามหลักการเหตุผลของประพจน์ที่เป็นฟัซซีเซต โดยทำการขยายนิยามของ Modus Ponens และ Modus Tollens สำหรับฟัซซีลอจิก เรียกว่า Generalized Modus Ponens :GMP และ Generalized Modus Tollens :GMT ตามนิยามที่3 และนิยามที่ 4

นิยาม 3 Generalized Modus Ponens :GMP

Generalized Modus Ponens :GMP คือการสรุปหาข้อมูล จากเหตุไปหาผล (forward data - driven)

Premise	u is A'
Implication	if u is A then v is B
Conclusion	v is B'

โดยที่GMP Implication ถูกนิยามดังสมการ (2.7)

$$A \rightarrow B = \max \{c \in [0,1], t(A,c) \leq B\}$$
 (2.7)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

นียาม 4 Generalized Modus Tollens :GMT

Generalized Modus Tollens :GMT คือการหาข้อสรุปจากผลไปหาเหตุ(backup goal driven)

Premise	u is B'
Implication	if u is A then v is B
Conclusion	v is A'

$$A \rightarrow B = \min \{ c \in [0,1], s(B,c) \leq A \}$$
 (2.8)

นียาม 5 ฟัชซีอิมพลิเคชัน (Fuzzy Implication)

ฟัชซีอิมพลิเคชัน คือเงื่อนไขความสัมพันธ์แบบฟัชซีระหว่างส่วนเหตุ(antecedent)และ ส่วนผล (consequence)ที่ใช้ในการหาข้อสรุปตามหลักการและเหตุผล

จากนียามของ GMP Implication ตามนียาม 3 และ GMT Implication ตามนียาม 4 จะเห็นว่าฟัชซีอิมพลิเคชันจะขึ้นอยู่กับการใช้ตัวปฏิบัติการ t-norms และ s-norms ซึ่งมีผู้นำเสนอวิธีการทำ ฟัชซีอิมพลิเคชันแบบต่างๆกันตามตัวปฏิบัติการ t-norms และ s-norms ที่เลือกใช้ดังตัวอย่างต่อไปนี้

Mamdani's Implication

$$\phi[\mu_A(x), \mu_B(y)] = \mu_A(x) \wedge \mu_B(y)$$
 (2.9)

Lasen's Implication

$$\phi[\mu_A(x), \mu_B(y)] = \mu_A(x) \cdot \mu_B(y)$$
 (2.10)

Standard Sequence Fuzzy Implication

$$\{((u,v), \mu_{Rs}(u,v)) | \mu_{Rs}(u,v) = \begin{cases} 1: ?A(u) \text{ } ?B(v) \\ 0: ?A(u) > ?B(v) \end{cases} \}$$
 (2.11)

Fukami et al ได้พิสูจน์หลักการเลือกฟังก์ชันสำหรับการอิมพลิเคชันของ GTM และGMP ไว้ตารางที่ 2.3 และ 2.4

Criteria	Premise(u is A')	Conclusion(v is B')
1	u is A	v is B
2a	u is (very A)	v is (very B)
2b	u is (very A)	v is B
3a	u is (more or less A)	v is (more or less B)
3b	u is (more or less A)	v is B
4a	u is (not A)	v is unknow
4b	u is (not A)	v is (not B)

ตารางที่ 2.3 แสดงผลการหาข้อสรุปตามหลักการและเหตุผล
โดยใช้ GMP ตามหลักการของ Fukami

Criteria	Premise(u is A')	Conclusion(v is B')
1	u is (not B)	v is (not A)
2	u is (not very B)	v is (not very A)
3	u is (more or less B)	v is (not more or less A)
4a	u is B	v is unknow
4b	u is B	v is A

ตารางที่ 2.4 แสดงผลการหาข้อสรุปตามหลักการและเหตุผล
โดยใช้ GMT ตามหลักการของ Fukami

2.2 ความสัมพันธ์ฟัซซี (Fuzzy Relation)

นิยาม 6 ผลคูณคาร์ทีเซียน(Cartesian Product)

กำหนดให้ $A_1, \dots, A_n$ เป็นฟัซซีในเอกภพสัมพัทธ์ $U_1, \dots, U_n$

โดยที่ $u_1, \dots, u_n \in U_1, \dots, U_n$ ตามลำดับ

ผลคูณคาร์ทีเซียนของ $A_1, \dots, A_n$ คือ ฟัซซีเซตในเอกภพสัมพัทธ์ $U_1 \times \dots \times U_n$ ที่มีฟังก์ชันการเป็นสมาชิกดังนี้

$$\mu_{A_1, \dots, A_n}(u_1, \dots, u_n) = \begin{cases} \min \{ \mu_{A_1}(u_1), \dots, \mu_{A_n}(u_n) \} & \text{min - intersection} \\ \mu_{A_1}(u_1), \dots, \mu_{A_n}(u_n) & \text{algebraic product} \end{cases}$$

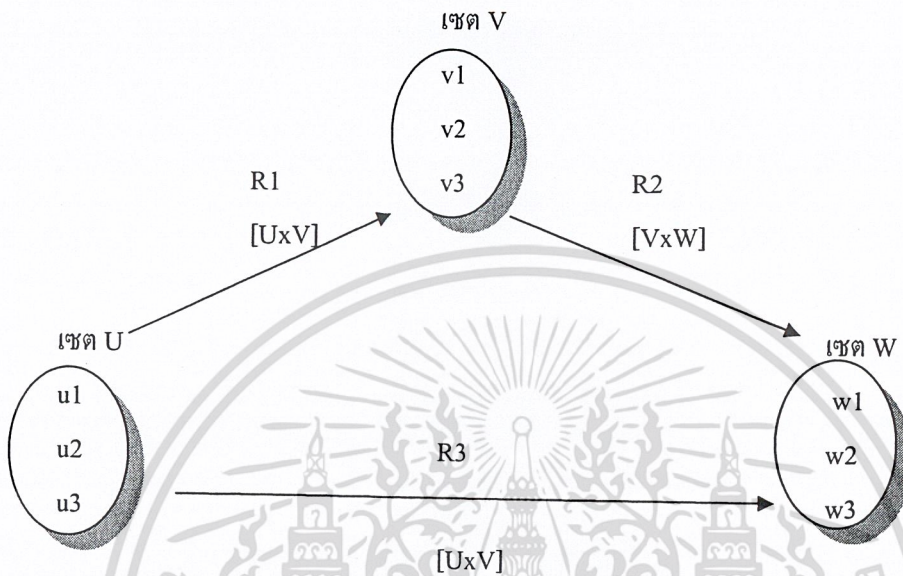
2.3 การคอมโพสิชันแบบฟัซซี (Fuzzy Composition)

กำหนดให้ $R_1(u, v)$ เป็นความสัมพันธ์จากเอกภพสัมพัทธ์ U ไป V โดยที่ $(u, v) \in U \times V$ และ

$R_2(v, w)$ เป็นความสัมพันธ์จากเอกภพสัมพัทธ์ V ไป W โดยที่ $(v, w) \in V \times W$ ใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การคอมโพสิชัน คือ ปฏิบัติการในการหาความสัมพันธ์ $R_3(u,w)$ จากความสัมพันธ์ที่ถ่ายทอดต่อกันเป็นลูกโซ่ $R_1(u,v)$ และ $R_2(v,w)$ ดังรูปที่ 2.1



รูป 2.1 แสดงการคอมโพสิชันของความสัมพันธ์ฟัซซี

กำหนดให้

$R_1(u,v)$ เป็นความสัมพันธ์ฟัซซีบนระนาบ $U \times V$ โดยที่ $(u,v) \in U \times V$

$R_2(v,w)$ เป็นความสัมพันธ์ฟัซซีบนระนาบ $V \times W$ โดยที่ $(v,w) \in V \times W$

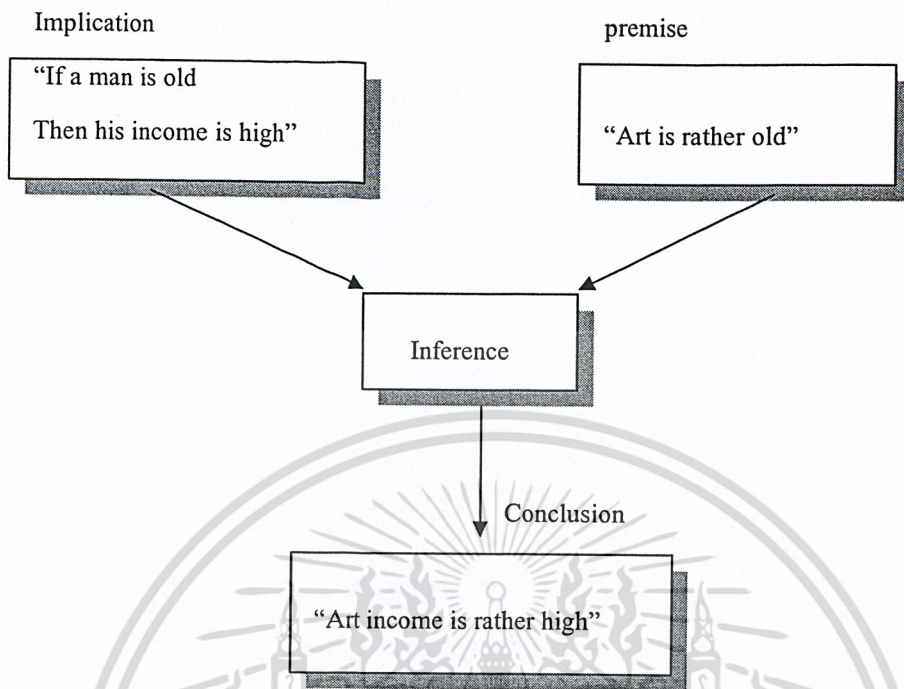
นิยาม 7 MAX-MIN Composition

$$R_3 = R_1 \circ R_2 = \{((u,w), \max\{\min\{\mu_{R_1}(u,v), \mu_{R_2}(v,w)\})\} \mid u \in U, v \in V, w \in W\} \quad (2.12)$$

2.4 การอินเฟอร์เรนซ์แบบฟัซซี (Fuzzy Inference)

นิยาม 8 การอินเฟอร์เรนซ์

การอินเฟอร์เรนซ์ คือ กระบวนการหาข้อสรุป conclusion จากหลักฐาน (premise) โดยใช้เงื่อนไขความสัมพันธ์ระหว่างเหตุและผล(implication)ดังรูป 2.2



รูปที่ 2.2 แสดงการหาข้อสรุปจากหลักฐาน โดยใช้เงื่อนไขความสัมพันธ์ระหว่างเหตุและผล

2.5 การฟัซซิฟิเคชัน (Fuzzification)

นิยาม 9 การฟัซซิฟิเคชัน

การฟัซซิฟิเคชัน คือ กระบวนการหาค่าฟัซซิบนเอกภพสัมพัทธ์ μ ของอินพุต เพื่อเป็นตัวแทนของอินพุต ซึ่งมีค่าเป็นตัวเลข สำหรับการใช้ของระบบฟัซซี

การฟัซซิฟิเคชัน สามารถหาได้ 2 วิธีดังนี้

1. การฟัซซิฟิเคชัน โดยโดยใช้วิธีซิงเกิลตัน จะได้ว่า Fuzzification (u_0) = A

โดยที่ A เป็นฟัซซีซิงเกิลตัน ที่มี $S(A) = \{u_0\}$

$$\text{หรือ} \quad \mu_A(u) = \begin{cases} 1; & u = u_0 \\ 0; & u \end{cases} \quad (2.13)$$

2. การฟัซซิฟิเคชัน โดยใช้ฟัซซีนัมเบอร์ จะได้ว่า Fuzzification (u_0) = A

โดยที่ A เป็นฟัซซีนัมเบอร์

$$\mu_A(u) = 1 \quad \text{เมื่อ } u = u_0$$

และ $\mu_A(u)$ ลดลง เมื่อ u อยู่ห่างออกไปจาก u_0

2.6 การดีฟัซซิฟิเคชัน (Defuzzification)

นิยาม 10 การดีฟัซซิฟิเคชัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

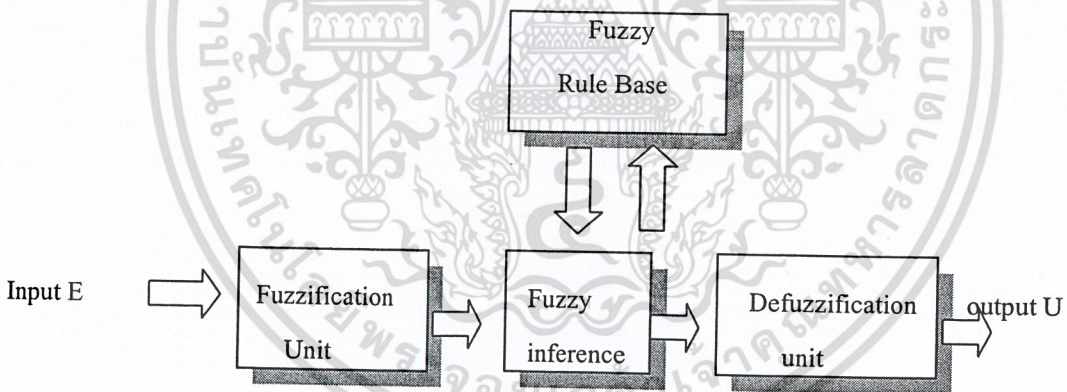
การดีฟัซซิฟิเคชัน คือ กระบวนการหาค่าเอาต์พุตเพียงค่าเดียว(ซึ่งเกิดต้น)ที่เหมาะสมที่สุด เพื่อเป็นสัญญาณเอาต์พุตที่ถูกต้องที่มีค่าความเป็นไปได้ กระจายอยู่บนเอกภพสัมพัทธ์ V ของ เอาต์พุต กำหนดให้ R คือ ความสัมพันธ์ฟัซซีจากเอกภพสัมพัทธ์ U ไป V ที่แสดงถึงความสัมพันธ์ระหว่าง อินพุต u และเอาต์พุต v
 u_0 คือ อินพุตที่วัดได้เป็นซึ่งเกิดต้น

ในการคอมโพซิชั่นระหว่างฟัซซีซึ่งเกิดต้น u_0 กับความสัมพันธ์ฟัซซี R จะได้ผลลัพธ์เป็นฟัซซีเซต B ซึ่งเป็นสับเซตของเอกภพสัมพัทธ์ V และเมื่อทำฟัซซิฟิเคชันจะหาฟัซซีซึ่งเกิดต้น v_0 ซึ่งเป็นตัวแทนของฟัซซีเอาต์พุต B เพื่อใช้เป็นเอาต์พุตที่แท้จริงของระบบได้

2.7 ตัวควบคุมฟัซซี

โครงสร้างของตัวควบคุมฟัซซีประกอบด้วย 4 หน่วยคือ

- 1.หน่วยฟัซซิฟิเคชัน
- 2.หน่วยอินเฟอร์เรนซ์
- 3.หน่วยดีฟัซซิฟิเคชัน
- 4.ฐานการควบคุมฟัซซี(Fuzzy rule base) ดังรูปที่ 2.3



รูป 2.3 แสดงโครงสร้างพื้นฐานของตัวควบคุมฟัซซี

2.7.1 หน่วยฟัซซิฟิเคชัน

หน่วยฟัซซิฟิเคชันมีหน้าที่นอมอไลซ์ตัวแปรสถานะของระบบ $x = [x_1, \dots, x_n]^T$ จะใช้เป็นอินพุตของตัวแปรควบคุมให้อยู่ในเอกภพสัมพัทธ์ที่กำหนด และทำการเปลี่ยนแปลงตัวอินพุต x ให้เป็นตัวแปรฟัซซี X' โดยใช้ ฟัซซีซึ่งเกิดต้น ฟัซซินัมเบอร์ และ ไฮบริดนัมเบอร์

2.7.2 ฐานกฎการควบคุมของฟัซซี

ฐานการควบคุมของฟัซซีเป็นหน่วยที่รวบรวมกฎการควบคุมแบบฟัซซีซึ่งอยู่ในรูปแบบ

IF...THEN...

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กำหนดให้กฎการควบคุมอยู่ในระบบดังนี้

IF x_1 is X_{k1} and...and x_m is X_{km} THEN y is Y_k

หรือ IF x is X_k THEN y is Y_k ; $k = 1, 2, \dots, m$

โดยที่ x คือ ตัวแปรสถานะของกระบวนการซึ่งใช้เป็นอินพุตของตัวควบคุมฟัซซี

$$x = [x_1, \dots, x_m]^T$$

X_k คือ ฟัซซีเซตของตัวแปรอินพุต ในส่วนของกฎการควบคุม

$$X_k = X_{k1} \times \dots \times X_{km}$$

m คือ จำนวนตัวแปรอินพุตของตัวควบคุม

y คือ ตัวแปรเอาต์พุตของตัวควบคุมฟัซซี

Y คือ ฟัซซีเซตของตัวแปรเอาต์พุตในส่วนผลของกฎการควบคุมที่ k

M คือ จำนวนกฎการควบคุมทั้งหมดของฐานกฎการควบคุม

2.7.3 หน่วยอินเฟอเรนซ์

หน่วยอินเฟอเรนซ์ ทำหน้าที่หาค่าเอาต์พุตของระบบฟัซซีจากค่าตัวแปรอินพุตแบบฟัซซี X' โดยวิธี GMP implication หาค่าระดับความเป็นสมาชิก (fire strength) ของแต่ละกฎจากความสัมพันธ์ระหว่างตัวแปรอินพุตแบบฟัซซี X' กับฟัซซีเซตในส่วนเหตุของแต่ละกฎ X_k แล้วจึงคำนวณหาค่าเอาต์พุตแบบฟัซซีจากระดับการเป็นสมาชิกกับฟัซซีเซตในส่วนผลของแต่ละกฎ Y_k

2.7.4 หน่วยดีฟัซซิฟิเคชัน

หน่วยดีฟัซซิฟิเคชัน ทำหน้าที่เปลี่ยนแปลงตัวแปรเอาต์พุตให้เป็นค่าเชิงเกิลตัน และทำการ ดินอโมไลซ์ให้อยู่ในเอกภพสัมพัทธ์ที่ต้องการ เพื่อใช้เป็นค่าเอาต์พุตของตัวควบคุม โดยใช้วิธีการที่ได้กล่าวถึงในเรื่องการดีฟัซซิฟิเคชัน

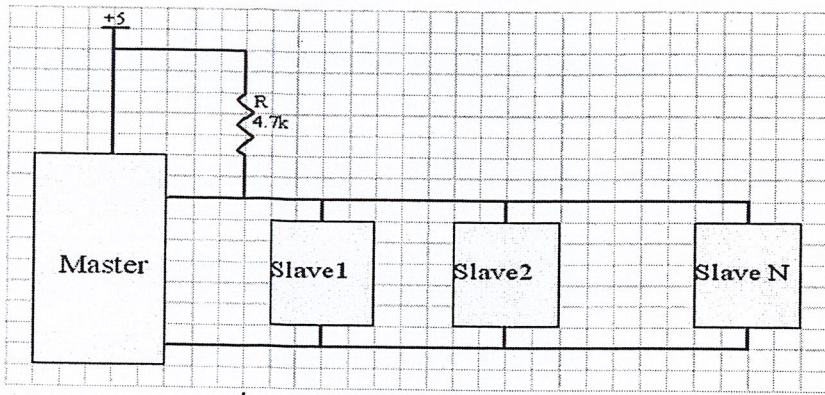
2.8 รูปแบบการติดต่อสื่อสาร

ระบบสื่อสารดัลลาสหนึ่งสาย (The Dallas 1-Wire Bus) เป็นระบบสื่อสารที่มีความฉลาดและใช้จำนวนสายสัญญาณเพียง 1 เส้นเท่านั้น โดยไม่ต้องมีสายสัญญาณนาฬิกาเนื่องจากสายข้อมูลทำหน้าที่เหมือนเป็นสายนาฬิกาในตัว ส่วนค่าของข้อมูลจะพิจารณาจากลักษณะของรูปสัญญาณที่ปรากฏบนสายสัญญาณในแต่ละไทม์สล็อต(Time Slot) โดยมีการกำหนดคาบเวลาสูงสุดต่ำสุดของสถานะต่างๆ ในการสื่อสารไว้อย่างชัดเจน รูปแบบการถ่ายทอดจะเป็นแบบ อะซิงโครนัสในระดับบิต

คุณสมบัติทางเทคนิคของระบบบัสหนึ่งสาย

สายสัญญาณรูปแบบนี้จะเป็นสายสัญญาณแบบสองทิศทาง แต่ข้อมูลจะสามารถเดินทางได้เพียงทิศทางเดียวในช่วงเวลาหนึ่งๆ คือมีลักษณะระบบสื่อสารแบบฮาร์ฟดูเพล็กซ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2-4 การทำงานในรูปแบบ 1-wire bus

ไทม์สล็อต

ฟังก์ชันของไทม์สล็อตที่กำหนดโดยอุปกรณ์มาสเตอร์มีอยู่ด้วยกัน 4 ฟังก์ชันคือ

- ไทม์สล็อตของการรีเซต ใช้ในการเริ่มต้นติดต่อกับอุปกรณ์สเลฟ
- การอ่านข้อมูลสำหรับอ่านข้อมูลที่ถูกส่งมาจากอุปกรณ์สเลฟ
- การเขียนข้อมูล 0, และการเขียนข้อมูล 1 ใช้สำหรับเขียนข้อมูลไปยังอุปกรณ์สเลฟผ่าน

สายสัญญาณของระบบ

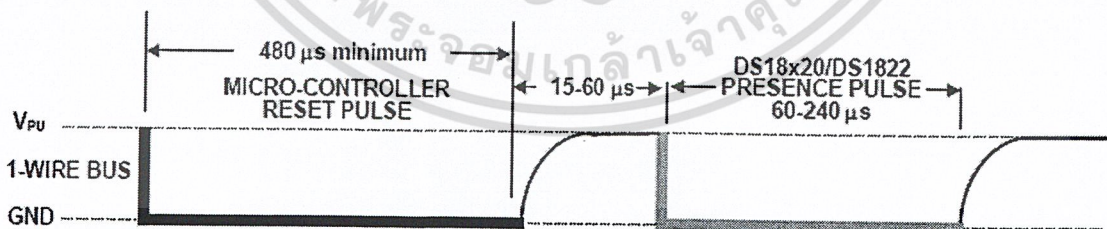
ทางด้านอุปกรณ์สเลฟจะมีฟังก์ชันไทม์สล็อตทั้งสิ้น 3 ฟังก์ชันคือ

- ไทม์สล็อตของการตอบสนอง สำหรับตอบสนองการติดต่อจากอุปกรณ์มาสเตอร์
- การเขียนข้อมูล 0, และการเขียนข้อมูล 1 ใช้สำหรับเขียนข้อมูลไปยังอุปกรณ์มาสเตอร์ผ่าน

สายสัญญาณของระบบซึ่งจะไปสัมผัสกับไทม์สล็อตของการอ่านข้อมูลของอุปกรณ์มาสเตอร์

การแยกแยะฟังก์ชันของแต่ละไทม์สล็อตจะใช้เวลาและความยาวของคาบเวลาและลักษณะของตัว

สัญญาณเป็นตัวกำหนดดังรูป 2-5 และ 2-6

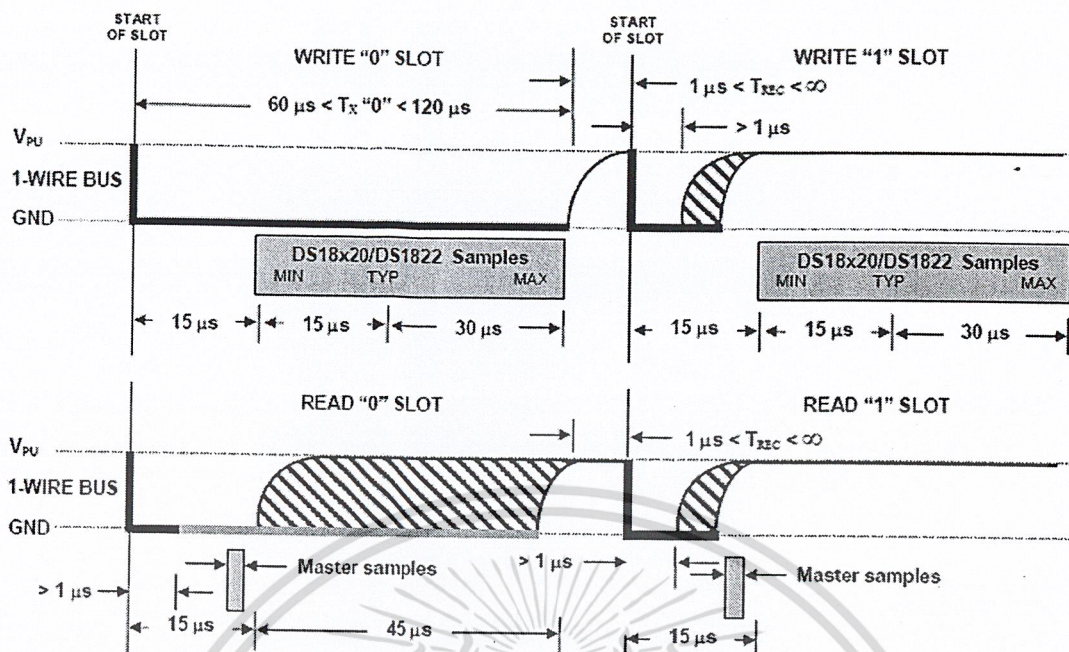


LINE TYPE LEGEND (Figure 2 and Figure 3)

- Bus master pulling low
- DS18x20/DS1822 pulling low
- Resistor pullup

รูปที่ 2-5 การทำงานของการรีเซต และรอการตอบกลับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2-6 การอ่านและเขียนข้อมูลของไมโครคอนโทรลเลอร์และ เซนเซอร์

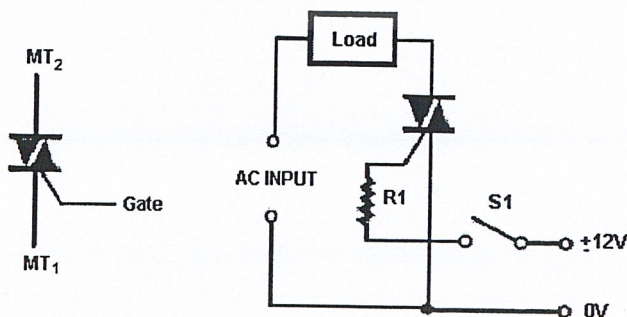
2.9 ไมโครคอนโทรลเลอร์และระบบควบคุม

2.9.1 ทฤษฎีของไมโครคอนโทรลเลอร์

ไมโครคอนโทรลเลอร์(Microcontroller) เป็นชื่อของอุปกรณ์อิเล็กทรอนิกส์แบบหนึ่งซึ่งรวมเอา หน่วยประมวลผล หน่วยคำนวณทางคณิตศาสตร์และลอจิก วงจรรับสัญญาณอินพุต วงจรขับสัญญาณ เอาต์พุต หน่วยความจำ วงจรกำเนิดสัญญาณนาฬิกาไว้ด้วยกัน ทำให้สามารถนำไปใช้งานแทนวงจร อิเล็กทรอนิกส์ที่ซับซ้อนได้เป็นอย่างดี ช่วยลดจำนวนอุปกรณ์และขนาดของระบบ ในขณะที่ขีด ความสามารถนั้นสูงขึ้น โดยที่สามารถเขียนโปรแกรมเพื่อกำหนดรูปแบบการควบคุมได้อย่างอิสระ

2.9.2 ไตรแอก

คุณสมบัติของไตรแอกและหลักการทำงาน



รูปที่ 2-7 สัญลักษณ์และวงจรพื้นฐานของไตรแอก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. โดยปกติ ถ้าไม่มีสัญญาณทริกที่เกิด ไตรแอกจะไม่ทำงาน โดย จะมีลักษณะเหมือนกับสวิตช์ที่ถูกเปิดวงจร

2. ถ้าในกรณีที่ MT2 และ MT1 ถูกป้อนด้วยแรงดันบวกและลบตามลำดับ ไตรแอกจะถูกกระตุ้นให้ทำงานได้โดยการป้อนสัญญาณพัลส์เพียงสั้น ๆ ที่เกิดของมัน โดยจะมีแรงดันตกคร่อมตัวมัน มีค่าประมาณ 1 หรือ 2 โวลต์ เท่านั้น และก็เช่นกันคือเมื่อ ไตรแอกเริ่มทำงานแล้ว ก็จะสามารถคงสภาพการทำงานอยู่เช่นนั้นต่อไปเรื่อย ๆ ตราบเท่าที่ยังมีกระแสไหลผ่านตัวมันอย่างต่อเนื่อง

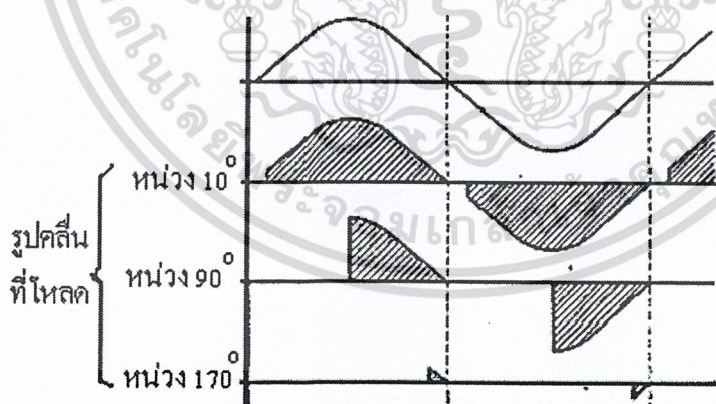
3. หลังจากที่ ไตรแอกคงสภาพการทำงานอยู่นั้น ทางเดียวที่จะหยุดการทำงานลงได้ ก็โดยการลดปริมาณกระแสที่ไหลผ่านตัวมันลง ให้มีค่าต่ำกว่ากระแสโฮลด์คิงของมัน ในกรณีที่ใช้ ไตรแอกในการจ่ายกระแส AC การหยุดทำงานจะเกิดขึ้นอย่างอัตโนมัติ เมื่อแรงดันของไฟสลับเข้าใกล้จุดตัดศูนย์ที่เกิดขึ้นทุก ๆ ครึ่งคลื่น นั่นคือกระแสจะลดลงเป็นศูนย์

4. ไตรแอกถูกกระตุ้นให้ทำงานได้ ทั้งสัญญาณแบบบวกและลบที่ป้อนให้แก่ขาเกต โดยไม่คำนึงถึงขั้วที่ต่ออยู่ที่ MT1 และ MT2

5. ไตรแอกสามารถทนการกระชากของกระแสได้สูง เช่นโดยปกติสำหรับ ไตรแอกที่ทนกระแสปกติได้ 10 แอมแปร์ (rms) สามารถทนการกระชากของกระแสในช่วงหนึ่ง คาบเวลาของไฟ 60 เฮิร์ตซ์ได้สูงถึง 100 แอมแปร์ เป็นต้น

หลักการควบคุมทางเฟส (Phase Trigger)

หลักการควบคุมทางเฟสนี้ โดยทั่วไปจะใช้ ไตรแอกเป็นตัวควบคุมกำลังไฟที่จ่ายให้แก่โหลด โดยแทนที่จะทริกขาเกตด้วยสัญญาณไฟตรงนั้นตรง ๆ ก็จะทริกโดยมีการหน่วงของเฟสด้วยวงจรอีกส่วนหนึ่ง



รูปที่ 2-8 การเปลี่ยนแปลงค่าของกำลังไฟฟ้าที่ป้อนให้แก่โหลด
โดยกำหนดได้จากตำแหน่งเวลาของการทริกที่ให้แก่ไตรแอก

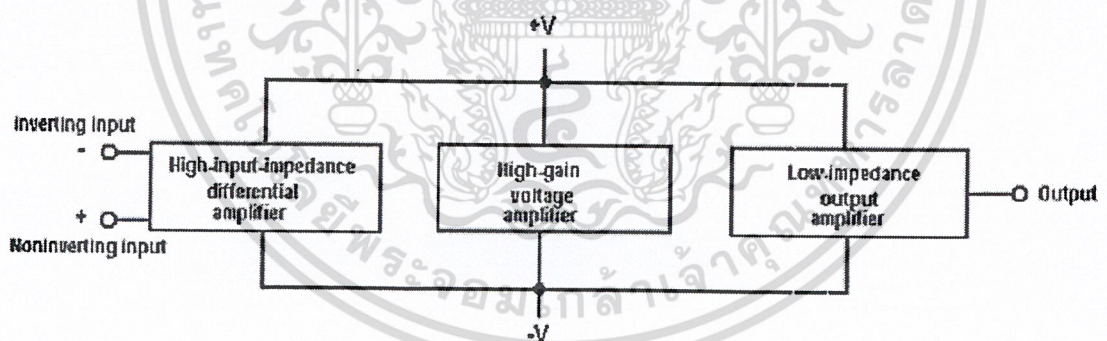
การห้วงเฟสมีผลดังนี้คือ ถ้าไตรแอกถูกทริกที่ตำแหน่งเฟส 10 องศาหลังจากที่ทุก ๆ ครึ่งรูปคลื่นเริ่มเข้ามา กำลังไฟเกือบทั้งหมดก็จะถูกป้อนให้แก่โหลด แต่ถ้าการทริกที่ตำแหน่งเฟส 90 องศาหลังจากทุก ๆ ครึ่งคลื่นเริ่มเข้ามา จะทำให้กำลังไฟที่ป้อนให้แก่โหลดนั้น ลดลงเหลือเพียงครึ่งหนึ่งของกำลังทั้งหมด และถ้าไปทริกที่ตำแหน่งเฟส 170 องศา หลังจากทุก ๆ ครึ่งรูปคลื่นเข้ามาแล้ว จะมีเพียงกำลังไฟส่วนน้อยเท่านั้นที่ป้อนให้แก่โหลด

2.9.3 ออปแอมป์

ออปแอมป์ คือ อุปกรณ์ชนิดหนึ่งซึ่งถูกออกแบบมาให้สามารถทำงานได้หลายรูปแบบ และยังเน้นความสะดวกในการนำไปใช้งานอีกด้วย โดยสามารถประกอบเป็นวงจรได้โดยการต่อรวมกับอุปกรณ์ภายนอกเพียงไม่กี่ตัวเท่านั้น ซึ่งคุณสมบัติของไอซี ออปแอมป์ที่พัฒนาขึ้นทำให้อุปกรณ์ชนิดนี้เป็นที่รู้จักกันแพร่หลาย

คุณสมบัติทั่วไปของออปแอมป์

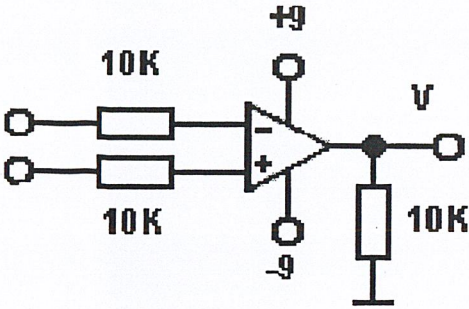
1. เนื่องจากอินพุตอิมพีแดนซ์ของออปแอมป์มีค่าสูงเป็นอนันต์ กระแสเข้าอินพุตจะต่ำจนเกือบเท่าศูนย์หรืออีกนัยหนึ่งไม่มีกระแสอินพุตเข้าสู่ออปแอมป์เลย
2. อัตราขยายขณะเปิดลูปจะมีค่าสูงมาก ซึ่งหมายความว่า แรงดันระหว่างขั้วอินพุตควรจะมีค่าเข้าใกล้ศูนย์
3. เอาต์พุตอิมพีแดนซ์มีค่าต่ำมากจนไม่ทำตัวเป็นโหลดต่อภาคเอาต์พุตของวงจรขยาย



รูปที่ 2-9 การทำงานโดยรวมของออปแอมป์

การเปรียบเทียบแรงดันไฟฟ้า

การทำงานของวงจรคอมพาราเตอร์ในขณะเปิดลูปนั้น ออปแอมป์จะสามารถเปรียบเทียบระดับสัญญาณระหว่างขั้วอินพุตทั้งสองได้อย่างแม่นยำ โดยอาศัยความสัมพันธ์ระหว่างขั้วอินพุต เมื่อแรงดันที่ขั้วลบมีค่าเป็นบวกสูงกว่าแรงดันอินพุตที่ขั้วบวกสัญญาณที่เอาต์พุตจะเป็นลบและมีขนาดเท่ากับ $-V_{cc}$ จากแหล่งจ่ายดังรูปที่ 2-10



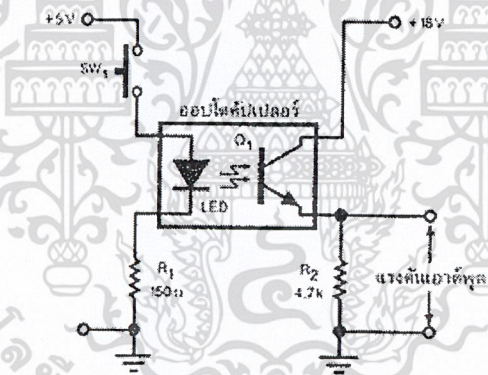
รูปที่ 2-10 วงจรเปรียบเทียบแรงดัน

2.9.4 ออปโตคัปเปิลเลอร์(Opto-Coupler)

ออปโตคัปเปิลเลอร์เป็นอุปกรณ์ที่ประกอบด้วย LED ซึ่งปกติเป็นชนิดอินฟราเรดและโฟโตทรานซิสเตอร์ หรือ โฟโต้ไดโอด ที่ถูกผลิตมาเป็นคู่กัน รวมอยู่ในตัวเดียวกัน

การทำงานของออปโตคัปเปิลเลอร์(Opto-Coupler)

หน้าที่หลักๆของออปโตคัปเปิลเลอร์ คือ แยกกราวด์ทางไฟฟ้า (Ground) ของวงจรออกจากกัน อย่างสิ้นเชิงดังตัวอย่างในรูปที่ 2-11



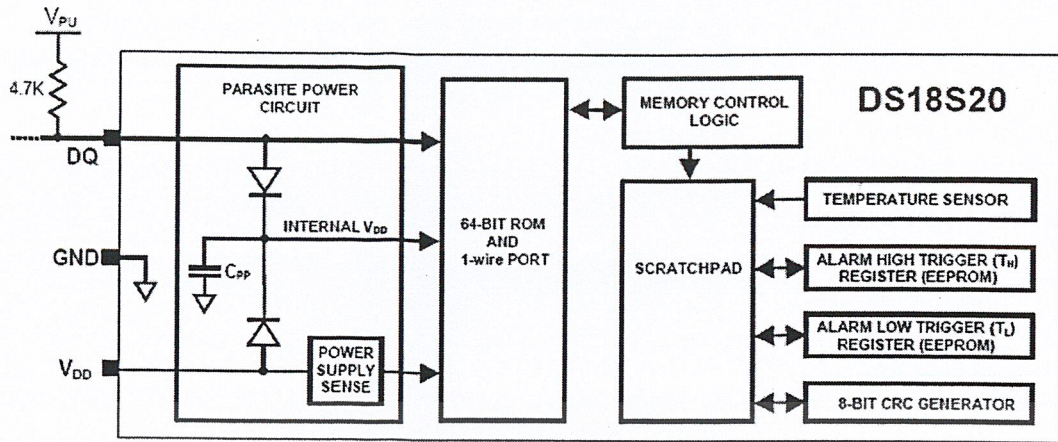
รูปที่ 2-11 วงจรออปโตคัปเปิลเลอร์พื้นฐาน

2.9.5 เซนเซอร์ DS18B20

เป็นอุปกรณ์ตรวจจับอุณหภูมิที่ใช้ในการติดต่อบนระบบบัส 1 สายมีขาต่อใช้งานเพียง 3 ขา คือ DQ เป็น ขาที่เชื่อมต่อกับระบบบัส ขา ต่อไฟเลี้ยง และ ขากราวด์

หัวของDS18B20 อยู่ที่ตัวตรวจจับอุณหภูมิความเร็วสูงที่เรียกว่า สแครตช์แพด(Scratchpad)ซึ่งมีขนาด 9 ไบต์ สามารถให้ข้อมูลอุณหภูมิได้ละเอียดถึง 16 บิต มีย่านวัดอุณหภูมิที่ -55 ถึง +125 องศาเซลเซียส หรือ -67 ถึง +257 องศาฟาเรนไฮต์

ในDS18B20 จะมีหน่วยความจำขนาด 64 บิต สำหรับเก็บข้อมูลต่างๆ ที่สำคัญ ประกอบด้วย รหัสตระกูล 8บิต เลขหมายประจำตัว 48 บิต และ รหัสตรวจสอบความผิดพลาด อีก 8 บิต



รูปที่ 2-12การทำงานภายในของDS1820

รูปแบบคำสั่ง

DS1820 มีรูปแบบคำสั่งที่ใช้สำหรับการควบคุมอยู่หลายคำสั่ง โดยอยู่ในรูปแบบเลขฐาน 16 ซึ่งแต่ละตัวจะมีความหมายอยู่ในตัวเองได้แก่

CCH หมายถึง Skip Rom เพราะมีอุปกรณ์สเลฟอยู่บนสายสัญญาณเพียงตัวเดียว

44H หมายถึง Convert Temperature เป็นคำสั่งที่ใช้แปลงค่าอุณหภูมิเป็นดิจิตอลมาเก็บไว้ในสแควดซ์แพด

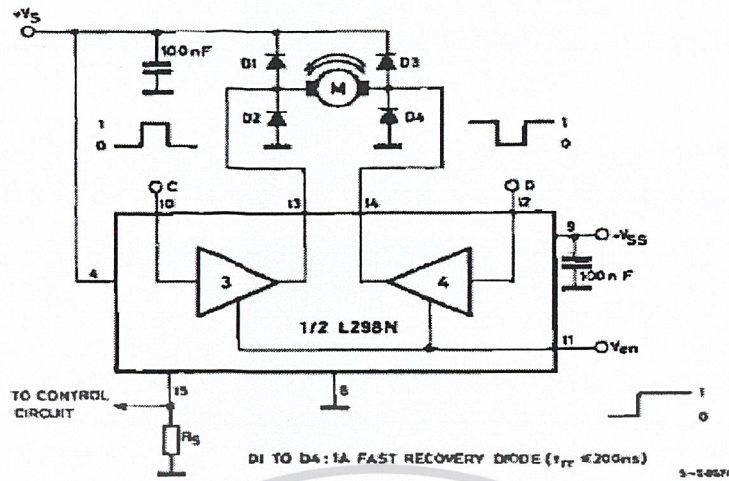
BEH หมายถึง Read Scratchpad เมื่อได้รับคำสั่งนี้ DS1820 จะทยอยส่งข้อมูลค่าอุณหภูมิออกมาทั้งหมด 9 ไบต์

FOH หมายถึง Search Rom เมื่อส่งคำสั่งนี้ออกไปจะเป็นการหาหมายเลขประจำตัวของเซนเซอร์ เพื่อต้องการรู้ว่าอุปกรณ์เซนเซอร์อยู่ที่ตัวในระบบ

55H หมายถึง Math Rom เป็นการตรวจสอบว่าหมายเลขที่ถูกสั่งให้ทำเป็นเซนเซอร์ตัวไหน

2.9.6 ไดรเวอร์กระแสตรง

L293D ทำหน้าที่ในการไดรกระแสที่ออกจาก ไมโครคอนโทรลเลอร์ ให้เพิ่มขึ้น โดยทิศทางการหมุนจะขึ้นอยู่กับที่เรากำหนด โดยจะต้องจ่ายให้ทั้ง ต่างกัน เช่นถ้าจ่ายขาอิน1 เป็น 1 ต้องจ่ายขาอิน2 เป็น 0 และจ่าย ที่ขา อินาเบิล เป็น 1 และหยุดเมื่อเราจ่ายค่าที่เหมือนกันในขาอินพุท เช่น อิน1 และ อิน2 เป็น 1 หรือ 0 ทั้งคู่ หรือจ่ายค่าที่ขาอินาเบิลเป็น 0



รูปที่ 2.13 วงจรไดรเวอร์มอเตอร์กระแสตรงพื้นฐาน

Inputs		Function
$V_{en} = H$	$C = H; D = L$	Forward
	$C = L; D = H$	Reverse
	$C = D$	Fast Motor Stop
$V_{en} = L$	$C = X; D = X$	Free Running Motor Stop

L = Low

H = High

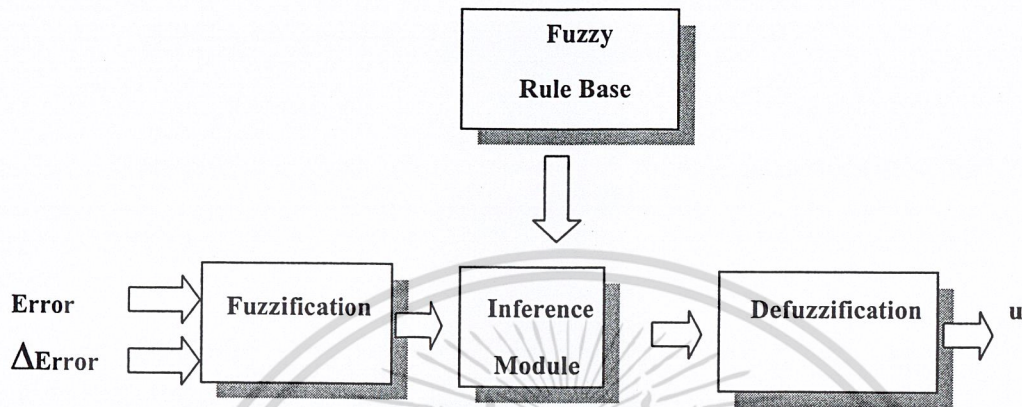
X = Don't care

ตารางที่ 2.5 แสดงการจ่ายไฟทำงานของมอเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3

การออกแบบตัวควบคุมด้วยFuzzy



รูป 3.1 แสดงโครงสร้างของตัวควบคุม

จุดประสงค์หลักของการควบคุม คือ ต้องการให้เอาต์พุตของระบบเข้าสู่ค่าอ้างอิง(setpoint)และไม่ส่งผลให้เกิดค่าพุ่งเกิน (overshoot) และค่าผิดพลาดที่สถานะที่คงตัวสูงเกินไป โดยในปริภูมิงานนี้ต้องการควบคุมค่าแรงดันไฟฟ้าที่จ่ายให้กับขดลวดความร้อนเพื่อให้ได้อุณหภูมิบริเวณที่ควบคุมคงที่ตามค่าอ้างอิงการออกแบบตัวควบคุมมีดังนี้

3.1 การกำหนดจำนวนตัวแปรของตัวควบคุม

ซึ่งประกอบด้วย 2 อินพุตและ 1 เอาต์พุต โดยที่ตัวแปรของตัวอินพุตของตัวควบคุม คือ ค่าความผิดพลาด e (error) และค่าการเปลี่ยนแปลงของค่าความผิดพลาด Δe (change of error) ของระบบควบคุม ส่วนตัวแปรเอาต์พุตของตัวควบคุม คือ สัญญาณควบคุม(control output; u)

3.2 การนอมอลไลซ์

การนอมอลไลซ์ เป็นการแปลงค่าอินพุต (คือ ค่าความผิดพลาดและค่าการเปลี่ยนแปลงของค่าความผิดพลาด) ดังแสดงในสมการ (3.1)ถึง(3.2)ให้อยู่ในช่วงเอกภพสัมพัทธ์ของฟัซซี โดยทั่วไปแล้วการนอมอลไลซ์ จะแปลงค่าเอกภพสัมพัทธ์ให้อยู่ในช่วง $[0,1]$ แต่ในที่นี้ ได้กำหนดเอกภพสัมพัทธ์ของค่าความผิดพลาดให้อยู่ในช่วง $[-5,5]$ และค่าการเปลี่ยนแปลงของค่าความผิดพลาดให้อยู่ในช่วง $[-5,5]$ ซึ่งการเปลี่ยนแปลงนี้จะทำโดยการนำค่าสเกลแฟกเตอร์เป็นตัวคูณค่าความผิดพลาดจริงให้อยู่ในช่วงเอกภพสัมพัทธ์ที่กำหนด การคำนวณหาค่านอมอลไลซ์แฟกเตอร์ หาได้จากสมการ (3.4)

$$e_1 = sp - t_1 \quad (3.1)$$

$$e_2 = sp - t_2 \quad (3.2)$$

$$\Delta e = e_2 - e_1 \quad (3.3)$$

โดยที่ e_1 คือ ค่าความผิดพลาดก่อนหน้านั้น
 e_2 คือ ค่าความผิดพลาดปัจจุบัน
 t_1 คือ อุณหภูมิก่อนหน้านั้น
 t_2 คือ อุณหภูมิปัจจุบัน
 Δe คือ ค่าการเปลี่ยนแปลงความผิดพลาด
 sp คือ ค่าอุณหภูมิอ้างอิง

$$N.F = \frac{x_{\max} - x_{\min}}{N} \quad (3.4)$$

โดยที่ $x_{\max}$ คือ ค่าอุณหภูมิสูงสุดที่ PCDIO CARD อ่านได้ในรูปแรงดันไฟฟ้า
 $x_{\min}$ คือ ค่าอุณหภูมิต่ำสุดที่ PCDIO CARD อ่านได้ในรูปแรงดันไฟฟ้า
 N คือ ค่าสูงสุดของเอกภพสัมพัทธ์ของฟัซซี่
 $N.F$ คือ ค่านอมอลไลซ์แฟกเตอร์

3.3 การกำหนดค่าเกน(Gain)ของตัวแปรอินพุตและเอาต์พุต

ค่าเกนของตัวแปรอินพุตและเอาต์พุต คือ สัมประสิทธิ์ในการถ่วงน้ำหนัก เพื่อกำหนดความสำคัญของตัวแปรสถานะแต่ละตัวของกระบวนการ โดยจะให้กำหนดขอบเขตสูงสุดของตัวแปรสถานะแต่ละตัว โดยปริยายนิพจน์นี้จะใช้ค่าเกน เท่ากับ 1

3.4 การเลือกวิธีและระดับควอนไทซ์(Quantize)

การกำหนดระดับควอนไทซ์ จะขึ้นอยู่กับความต้องการของผู้ออกแบบ และข้อจำกัดทางฮาร์ดแวร์ของอุปกรณ์ที่นำมาใช้ในการสร้างตัวควบคุม กรณีที่ต้องการให้ความถูกต้องสูง จะต้องใช้ระดับควอนไทซ์สูง แต่ในขณะเดียวกันก็ต้องใช้เวลาและหน่วยความจำในการคำนวณมาก ดังนั้น การออกแบบจึงต้องเลือกกระดัดที่เหมาะสมระหว่างความถูกต้องในการควบคุมและข้อจำกัดทางฮาร์ดแวร์

3.5 ฟัซซี่ฟิเคชัน(Fuzzification)

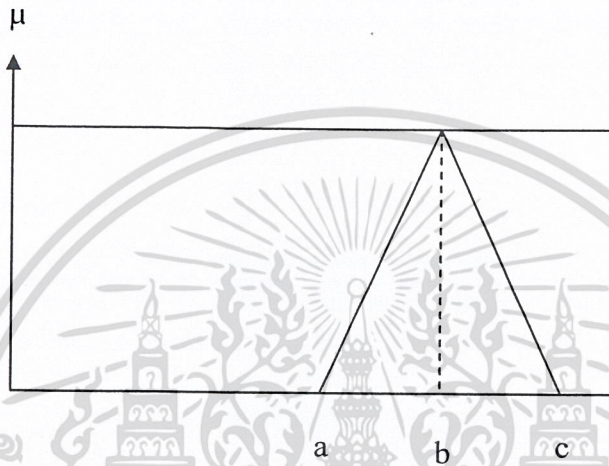
ทำหน้าที่แปลงค่าเอาต์พุต 2 ตัวที่ได้ทำการนอมอลไลซ์ในช่วงเอกภพสัมพัทธ์ที่กำหนดไว้แล้ว ให้อยู่ในรูปของฟัซซี่เซตโดยค่าระดับความเป็นสมาชิกของ e และ Δe แทนด้วย μ_e และ $\mu_{\Delta e}$ ตามลำดับ วิธีฟัซซี่ฟิเคชันที่ใช้ในปริยายนิพจน์นี้ คือ การเปลี่ยนแปลงแบบฟังก์ชัน โดยใช้ฟังก์ชันสามเหลี่ยม ดังแสดงในสมการ(3.5) และลักษณะของฟังก์ชันสามเหลี่ยมแสดงในรูปที่ 3.2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$T(u;a,b,c) = \begin{cases} 0 & u < a \\ (u-a)/(b-a) & a \leq u \leq b \\ (c-u)/(c-b) & b \leq u \leq c \\ 0 & u > c \end{cases} \quad (3.5)$$

โดยที่ $T(u;a,b,c)$ คือ ฟังก์ชันสามเหลี่ยม

a,b,c คือ พารามิเตอร์ของฟังก์ชันสามเหลี่ยม



รูปที่ 3.2 ฟังก์ชันสามเหลี่ยมซึ่งเป็นฟังก์ชันการเป็นสมาชิกของฟัซซีเซต

3.6 การกำหนดฟังก์ชันการเป็นสมาชิก(Membership function)

การกำหนดรูปร่างฟังก์ชันการเป็นสมาชิกของตัวแปรอินพุตจะเป็นการกำหนดความสัมพันธ์ในการแมปปีง(mapping) ค่าตัวแปรอินพุตในเอกภพสัมพัทธ์กับฟัซซีเซตของตัวแปรอินพุต ถ้าต้องการให้ความสัมพันธ์เป็นเชิงเส้น ก็ต้องกำหนดรูปร่างของฟังก์ชันการเป็นสมาชิกแบบเชิงเส้น เช่น รูปสามเหลี่ยมหรือสี่เหลี่ยมคางหมู แต่ถ้าต้องการให้ความสัมพันธ์ไม่เป็นเชิงเส้นควรกำหนดรูปร่างของฟังก์ชันการเป็นสมาชิกแบบไม่เป็นเชิงเส้น เช่น รูปประฆังคว่ำ

การกำหนดการกระจายฟัซซีเซตบนเอกภพสัมพัทธ์ สามารถกำหนดทั้งแบบที่เป็นเชิงเส้นและแบบที่ไม่เป็นเชิงเส้น ขึ้นอยู่กับความต้องการของผู้ออกแบบ โดยบริเวณที่มีความหนาแน่นของฟังก์ชันการเป็นสมาชิกสูง จะแสดงถึงความสัมพันธ์ของตัวแปรอินพุตและเอาต์พุตที่มีความซับซ้อนมากกว่าบริเวณที่มีความหนาแน่นของฟังก์ชันการเป็นสมาชิกต่ำ

การซ้อนทับ(over lap)ของฟังก์ชันการเป็นสมาชิกเป็นคุณสมบัติที่สำคัญอย่างหนึ่งของตัวควบคุมสมาชิกฟัซซี เนื่องจากในการอินเฟอเรนซ์ของฟัซซีต้องการให้เกิดกิริยาระหว่างกันของกฎควบคุมมากกว่า 1 กฎ การกำหนดเปอร์เซ็นต์การซ้อนทับจะขึ้นอยู่กับความคลุมเครือหรือการซ้อนทับของระบบการออกแบบให้ฟังก์ชันการเป็นสมาชิกของตัวแปรอินพุตมีการซ้อนทับมากกว่า 2 เซต จะทำให้ Control hypersurface มีความเรียบมากขึ้น แต่ในกรณีที่มีการเปลี่ยนแปลงการควบคุมใดๆ จะมีผลกระทบต่อบ

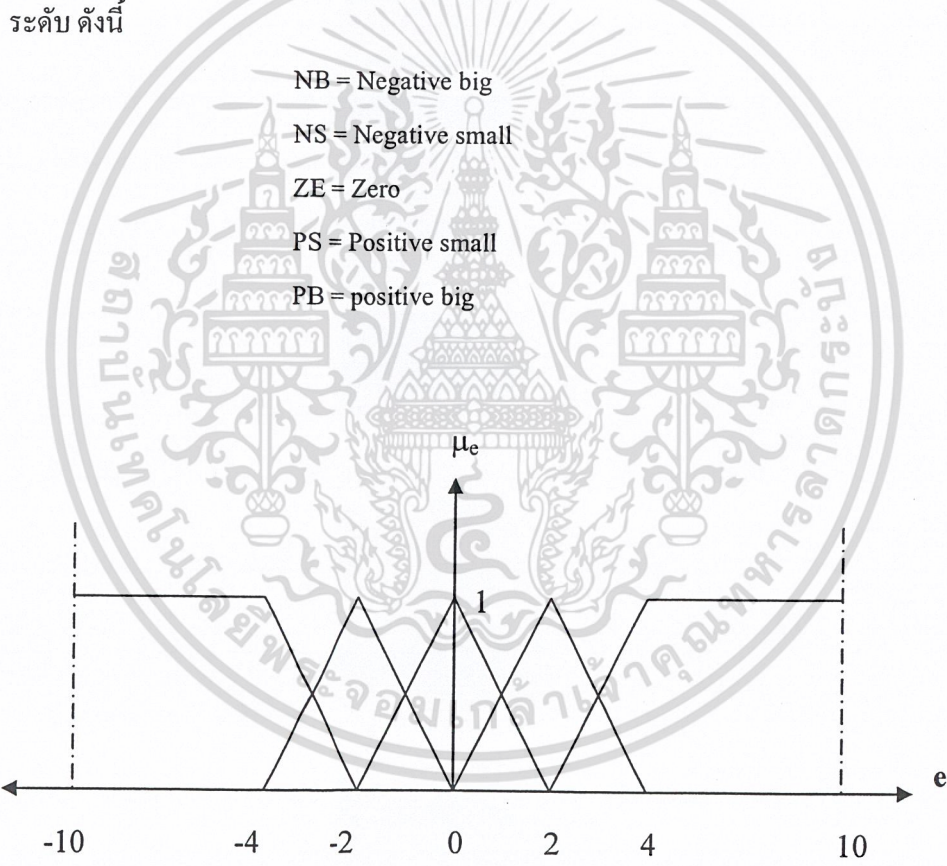
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การควบคุมกฎที่อยู่ติดกันมาก ทำให้เกิดการรบกวนกันของกฎการควบคุม ดังนั้น ในกาออกแบบฟังก์ชันการเป็นสมาชิกของตัวแปรอินพุตจึงไม่ควรให้มีการซ้อนทับกันมากกว่า 2 เซต

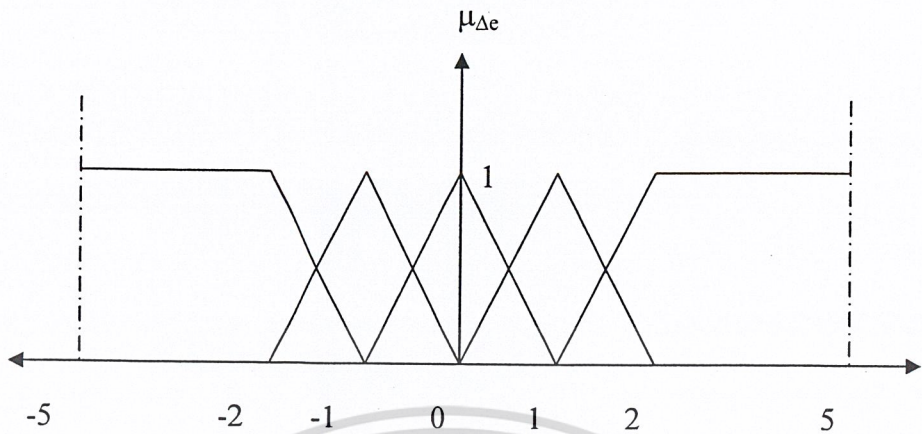
3.6.1 ฟังก์ชันการเป็นสมาชิกของอินพุต

เป็นฟังก์ชันที่ทำหน้าที่แปลงค่าของตัวแปรอินพุต e และ Δe ซึ่งเป็นคลิซปีเซตให้อยู่ในรูปของฟัซซี่เซต โดยมีข้อกำหนดดังนี้

- 1. กำหนดเอกภพสัมพัทธ์ของ e เท่ากับ $[-5,5]$ และ Δe เท่ากับ $[-5,5]$
- 2. รูปร่างฟังก์ชันการเป็นสมาชิกของฟัซซี่เซตเป็นรูปสามเหลี่ยม
- 3. มีการกระจายแบบเส้น และมีการซ้อนทับกับแต่ละฟังก์ชัน 50%
- 4. ระดับจำนวนเทอมของฟัซซี่เซต 5 เทอม ดังแสดงในรูปที่ 3.3 และ 3.4 โดยกำหนดค่าลิงกวิสติก 5 ระดับ ดังนี้



รูปที่ 3.3 ฟังก์ชันการเป็นสมาชิกของค่าความผิดพลาด (e)

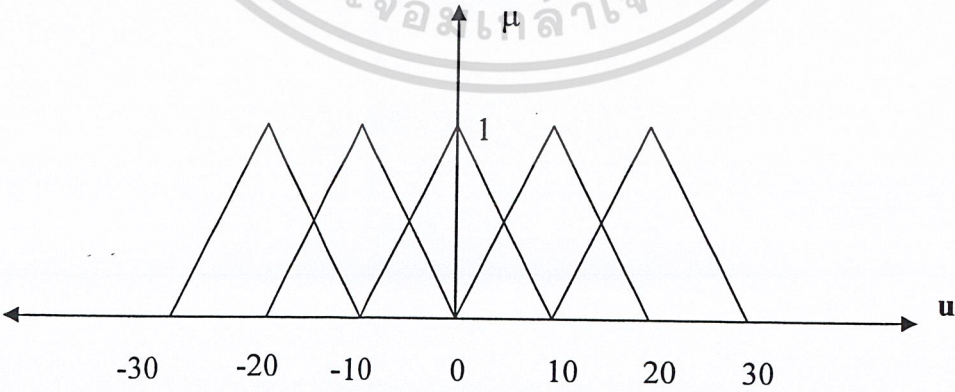


รูปที่ 3.4 ฟังก์ชันการเป็นสมาชิกของค่าการเปลี่ยนแปลงของผลผลิต (Δe)

3.6.2 ฟังก์ชันการเป็นสมาชิกเอาต์พุต

- 1. กำหนดเอกภพสัมพัทธ์ของเอาต์พุต u เท่ากับ $[-30,30]$
- 2. รูปร่างของฟังก์ชันการเป็นสมาชิกของฟัซซี่เป็นรูปสามเหลี่ยม
- 3. มีการกระจายแบบเชิงเส้นและกาซ้อนทับกันแต่ละฟังก์ชัน 50%
- 4. ระดับจำนวนของฟัซซี่เซต 5 เทอม ดังแสดงในรูป 3.5 โดยกำหนดค่าลึงกวิสติก 5 ระดับ ดังนี้

- NB = Negative big
- NS = Negative small
- ZE = Zero
- PS = Positive small
- PB = Positive big



รูปที่ 3.5 ฟังก์ชันการเป็นสมาชิกของเอาต์พุต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.7 ฐานกฎ

ฐานกฎ ประกอบด้วยเซตของกฎในรูป IF-THEN ซึ่งเป็นหัวใจสำคัญของตัวควบคุมแบบฟัซซี่ เพราะว่าถ้าฐานกฎได้รับการออกแบบอย่างเหมาะสม ระบบก็จะทำงานได้อย่างดี แต่ถ้าฐานรากได้รับการออกแบบอย่างไม่เหมาะสม การทำงานของระบบอาจไม่ดีหรืออาจเกิดผลเสียต่อระบบได้ ฐานกฎนั้น อาจจะได้มาจากความรู้หรือประสบการณ์ของผู้เชี่ยวชาญหรือได้มาจากการทดลองซึ่งระบบจะใช้กฎต่างๆ ในการอนุมานหรือเรียกว่าอินเฟอร์เรนซ์ เพื่อหาข้อสรุปต่อไป

3.8 ส่วนของอินเฟอร์เรนซ์

การอินเฟอร์เรนซ์ คือ การหาข้อสรุปจากหลักฐานหรือเหตุผลที่มีอยู่แล้วนำมาสรุปตามเงื่อนไข ความ สัมพันธ์(implication)ในที่นี้ จะเลือกใช้การอนุมานแบบ GMP โดยรูปแบบกฎดังนี้

R1: IF e is A1 AND ce is B1 THEN U is C1

R2: IF e is A2 AND ce is B2 THEN U is C2

...

Rn: IF e is An AND ce is Bn THEN U is Cn

กำหนดให้

R1,R2,...Rnเป็นอันดับของกฎจำนวนกฎจะได้จากการจับคู่(Combination)เทอมของ e และ Δe จะได้จำนวนกฎทั้งหมด 25 กฎ

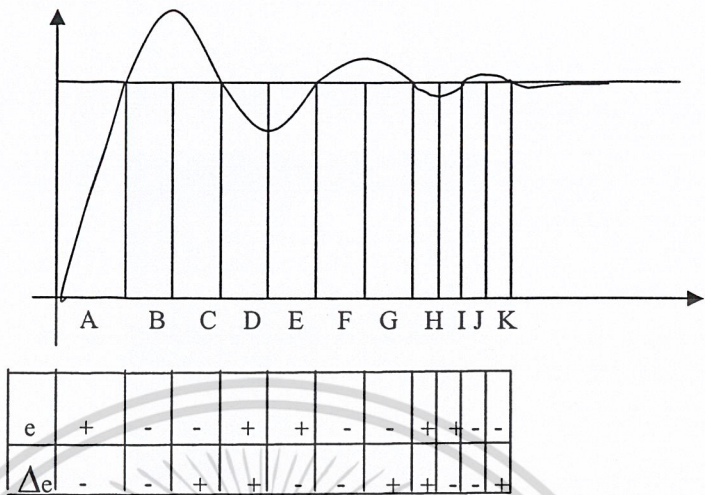
A1,A2,...,An เป็นค่าลึงกวิสติก ของฟังก์ชันการเป็นสมาชิกของ e

B1,B2,...,Bn เป็นค่าลึงกวิสติก ของฟังก์ชันการเป็นสมาชิกของ Δe

C1,C2,...,Cn เป็นค่าลึงกวิสติก ของฟังก์ชันการเป็นสมาชิกของ U

3.9 แนวทางการออกแบบกฎ

แนวทางการออกแบบกฎ โดยวิธีฮิวริสติก(Heuristic)เป็นการออกแบบกฎจากพฤติกรรม การตอบสนองของระบบที่ได้จากประสบการณ์หรือจากการทดสอบจนกระทั่งได้ค่าสมรรถนะที่ต้องการ ซึ่ง การออกแบบกฎโดยประมาณนี้ จะใช้วิธีพิจารณาความสัมพันธ์ของอินพุตและผลของการตอบสนอง ดัง แสดงในรูปที่ 3.6 ซึ่งสามารถเขียนเป็นกฎแบบหยาบได้ดังแสดงในตารางที่ 3.1



รูปที่ 3.6 ผลการตอบสนองของการเปลี่ยนระดับและการแบ่งช่วงของผลการตอบสนอง เพื่อหาความสัมพันธ์ของค่าผิดพลาด และการเปลี่ยนแปลงค่าผิดพลาด

Rule no.	E	Δe	output	reference
1	P	Z	P	Point a,e,i
2	Z	N	N	Point b,f,j
3	N	Z	N	Point c,g,k
4	Z	P	P	Point d,h,l
5	Z	Z	Z	Setpoint
6	P	N	P	Range A,E

ตารางที่ 3.1 ความสัมพันธ์ของตัวแปรอินพุตและเอาต์พุตของการออกแบบกฎ

นอกจากนี้จากความสามารถของอินพุตและผลการตอบสนองทำให้สามารถแบ่งกฎแบบละเอียดได้ 5 กลุ่ม ดังแสดงในรูปที่ 3.7 ดังนี้

กำหนดให้ค่าลึงกวิสติก สำหรับค่าผิดพลาดอัตรา การเปลี่ยนแปลงของค่าผิดพลาด และเอาต์พุตของระบบฟัซซี่ คือ

- NB = Negative big
- NS = Negative small
- ZE = Zero
- PS = Positive small
- PB = Positive big

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กลุ่มที่ 0(Group0) คือกลุ่มของกฎที่มีค่า $e(k)$ และค่า $\Delta e(k)$ เป็น(PositiveหรือNegative) ด้วยขนาด (smallหรือZero)ซึ่งหมายความว่า ค่าปัจจุบันของเอาต์พุตห่างจากค่าที่กำหนดไว้เพียงเล็กน้อย ดังนั้น ค่า $\Delta u(k)$ ควรมีค่าน้อยด้วย (smallหรือZero)สรุปกฎในกลุ่มนี้จะสัมพันธ์กับสภาวะ steady stateของระบบ

กลุ่มที่ 1(Group1) คือกลุ่มของกฎที่มีค่า $e(k)$ เป็น NB,NS ซึ่งหมายความว่า ค่าเอาต์พุตมีค่ามากกว่าค่าที่กำหนดไว้มาก ขณะเดียวกัน ค่า $\Delta e(k)$ เป็นPositiveแสดงว่าเอาต์พุตวิ่งหาค่าที่กำหนดให้ ดังนั้นค่า $\Delta u(k)$ จะเป็นตัวเร่งให้ระบบเข้าสู่ค่าที่กำหนดเร็วขึ้นหรือหน่วงให้ระบบเข้าสู่ค่าที่กำหนดช้าลงนั่นเอง เช่น ถ้าเอาต์พุตของระบบมีค่ามากกว่าค่าที่กำหนดให้มาก ($e(k)$ is NB)และ มันกำลังเคลื่อนที่เข้าหาค่าที่กำหนดอย่างช้าๆ ($\Delta e(k)$ is PS)ดังนั้น ขนาดของการเคลื่อนที่เข้าสู่ค่าที่กำหนดควรจะกระตุ้นให้เร็วขึ้น($\Delta u(k)$ is NS)เป็นต้น

กลุ่มที่ 2(Group2) คือกลุ่มของกฎที่มีค่า $e(k)$ มีค่าเข้าใกล้ค่าที่กำหนด (PS,ZE,NS)หรือมีการต่ำกว่าค่าที่กำหนดมาก (PB)ขณะเดียวกัน ค่า $\Delta e(k)$ เป็นบวก(Position)แสดงว่าค่าเอาต์พุตเคลื่อนที่ออกจากค่าที่กำหนดไว้ ดังนั้นค่า $\Delta u(k)$ ควรมีค่าเป็นบวกเพื่อเปลี่ยนค่าเอาต์พุตให้เคลื่อนที่สู่ค่าที่กำหนดให้แทนที่จะเคลื่อนออกจากค่าที่กำหนด

กลุ่มที่ 3(Group3)คือกลุ่มของกฎที่มีค่า $e(k)$ เป็นPS,PB ซึ่งหมายความว่า เอาต์พุตมีค่าต่ำกว่าที่กำหนดมาก ขณะเดียวกัน ค่า $\Delta e(k)$ เป็นลบ(Negative)แสดงว่าเอาต์พุตเคลื่อนที่เข้าสู่ค่าที่กำหนดให้ $\Delta e(k)$ จะเป็นตัวเร่งให้ระบบเข้าสู่ค่าที่กำหนดเร็วขึ้นหรือหน่วงให้ระบบเข้าสู่ค่าที่กำหนดช้าลงนั่นเอง เช่น ถ้าเอาต์พุตของระบบมีค่ามากกว่าค่าที่กำหนดให้ ($\Delta e(k)$ is PB) และมันกำลังเคลื่อนที่เข้าหาค่าที่กำหนดไว้อย่างรวดเร็ว($\Delta e(k)$ is NS) ดังนั้น ขนาดของการเคลื่อนที่เข้าสู่ค่าที่กำหนดควรจะกระตุ้นให้เร็วขึ้นเพียงเล็กน้อยเท่านั้น($\Delta e(k)$ is PS) เป็นต้น

กลุ่มที่ 4(Group4)คือกลุ่มของกฎที่มีค่า $e(k)$ ใกล้เคียงกับค่าที่กำหนด(ZE,NS)หรือมีค่ามากกว่าที่กำหนดให้ไว้มาก (NB)ขณะเดียวกัน ค่า $\Delta e(k)$ เป็นลบ(Negative) แสดงว่าค่าเอาต์พุตเคลื่อนที่เข้าหาค่าที่กำหนดไว้นั่นเอง

$e \backslash \Delta e$	NB	NS	Z	PS	PB
NB	4			1	
NS					
Z			0		
PS					2
PB		3			

รูปที่ 3.7 กลุ่มของกฎทั้ง 5 ที่ได้จากการออกแบบกฎของตัวควบคุมแบบ อิวิริสติก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากที่กล่าวข้างต้น จะเห็นว่าสิ่งสำคัญสำหรับตัวควบคุมฟัซซี่ คือ ฐานความรู้ ซึ่งจากการนำตัวควบคุมฟัซซี่มาประยุกต์ใช้กับงานด้านต่างๆ พบว่า ปัญหาที่เกิดขึ้น คือ กรณีที่ฐานความรู้ไม่เหมาะสม อาจทำให้ระบบไม่สามารถเข้าสู่สภาวะคงตัวได้ หรือ กรณีที่พารามิเตอร์ของกระบวนการที่การเปลี่ยนแปลง ฐานความรู้เดิมที่มีอยู่อาจจะแก้ปัญหาไม่ได้ ดังนั้น จึงพัฒนาตัวควบคุมฟัซซี่เพื่อให้สามารถมีการเรียนรู้และปรับปรุงฐานความรู้เพื่อให้ระบบมีการตอบสนองที่ดี

3.10 การเลือกวิธีการดีฟัซซิฟิเคชัน

จะเป็นการกำหนดความละเอียดของสัญญาณเอาต์พุต โดยจะต้องสัมพันธ์กับการกำหนดฟังก์ชันการเป็นสมาชิกของตัวแปรอินพุตและตั้งแปรเอาต์พุต การเลือกวิธีการดีฟัซซิฟิเคชัน จะขึ้นอยู่กับความต้องการของผู้ออกแบบ ซึ่งจะมีข้อดี ข้อเสียต่างกันดังนี้

-วิธี Max Procedureเป็นวิธีการสามารถหาได้ง่ายและรวดเร็วที่สุด แต่จะให้สัญญาณการควบคุมที่หยาบมาก จึงไม่เหมาะกับการใช้งาน

-วิธี MOM เป็นวิธีการสามารถหาได้ง่ายและรวดเร็วที่สุด ซึ่งการดีฟัซซิฟิเคชันด้วยวิธีการนี้ จะได้ผลเช่นเดียวกับการกำหนดฟังก์ชันการเป็นสมาชิกของตัวแปรเอาต์พุตเป็นฟัซซี่ซึ่งเกิดขึ้น เนื่องจากค่าตัวแปรอินพุตที่ตำแหน่งอื่นๆ ที่มีค่าระดับการเป็นสมาชิกต่ำกว่าค่าระดับการเป็นสมาชิกสูงสุด จะไม่มีผลต่อค่าเอาต์พุต วิธีนี้จะใช้สัญญาณควบคุมที่หยาบ และผลการควบคุมไม่ดี แต่จะถูกเลือกใช้ในกรณีที่มีการคำนวณน้อยและไม่ต้องการความถูกต้องมากนัก

-วิธีCOG เป็นวิธีที่นิยมกันมากที่สุด เนื่องจากการหาจุดศูนย์กลางของเอาต์พุตที่แท้จริง โดยพิจารณาจากอินพุตทุกตัว ทำให้ได้ค่าที่ถูกต้องกว่าวิธีอื่น แต่ใช้เวลาในการคำนวณมาก

-วิธีFM เป็นวิธีที่ดัดแปลงมาจากวิธี COG โดยใช้ระดับการควอนไตซ์ที่น้อยกว่า เพื่อลดเวลาการคำนวณ ในการดีฟัซซิฟิเคชันด้วยวิธีนี้ จะได้ผลเช่นเดียวกับการกำหนดค่าฟังก์ชันการเป็นสมาชิกของตัวแปรเอาต์พุตเป็นฟัซซี่ซึ่งเกิดขึ้น แต่มีข้อจำกัด คือ ควรใช้ในการออกแบบตัวควบคุมโดยกำหนดค่าฟังก์ชันของการเป็นสมาชิกของตัวแปรเอาต์พุตแบบสมมาตรเท่านั้น ทั้งนี้จากการดีฟัซซิฟิเคชันด้วยวิธีนี้จะใช้ค่าจุดกึ่งกลางของฟัซซี่เซตของตัวแปรเอาต์พุตเพื่อเป็นตัวแทนของแต่ละเซต

ในปริณูณานิพจน์นี้จะใช้แบบCOGเพราะเป็นวิธีที่ละเอียดและถูกต้อง โดยจะมีสมการดังนี้

$$x_0 = \frac{\sum_{j=1}^n \mu(x_j) \cdot x_j}{\sum_{j=1}^n \mu(x_j)} \tag{3.6}$$

ซึ่ง n เป็นจำนวนของระดับการของการควอนไตซ์เอาต์พุต

3.11 การคืนมอดไลซ์ (Denormalize)

ตัวแปรเอาต์พุตให้อยู่ในเอกภพสัมพัทธ์ที่กำหนดโดยจะต้องรู้ช่วงหรือเอกภพสัมพัทธ์ที่แท้จริงของตัวแปรเอาต์พุต

ช่วงของe	ช่วงของ	ช่วงของเอาต์พุต u	ค่าลิงวิสติก
$4 \leq e$	$2 \leq \Delta e$	$10 \text{ u } 30$	PB
$0 \leq e \leq 4$	$0 \leq \Delta e \leq 2$	$0 \text{ u } 20$	PS
$-2 \leq e \leq 2$	$-1 \leq \Delta e \leq 1$	$-10 \text{ u } 10$	Z
$-2 \leq e \leq 0$	$-2 \leq \Delta e \leq 0$	$-20 \text{ u } 0$	NS
$e \leq -4$	$\Delta e \leq -2$	$-30 \text{ u } -10$	NB

ตาราง 3.2 ช่วงความเป็นสมาชิกในแต่ละ ลิงวิสติกของอินพุตและเอาต์พุต

จากแนวทางการออกแบบกฎโดยวิธีวิสติก ดังนี้

1) ที่ค่า $e(k)$ เป็น PB คือค่าอุณหภูมิปัจจุบัน มีค่าต่ำกว่าค่าอ้างอิงมาก ดังนั้น ค่า $u(k)$ จึงมีค่ามากที่สุด คือเป็น PB ที่ $e(k)$ เป็น NB คือค่าอุณหภูมิปัจจุบันมีค่าสูงกว่าค่าอ้างอิงมาก ค่า $u(k)$ จึงมีค่าน้อยที่สุด คือเป็นNB

2) ที่ค่า $e(k)$ เป็น NS และ $\Delta e(k)$ เป็น Negative คือค่าอุณหภูมิปัจจุบันมีค่าสูงกว่าค่าอ้างอิงปานกลาง ซึ่งอุณหภูมิกำลังเพิ่มขึ้น ดังนั้นค่า $u(k)$ จึงเป็น NB

3)ที่ค่า $e(k)$ เป็น NS และ $\Delta e(k)$ เป็น zero กับ position คือค่าอุณหภูมิปัจจุบันมีค่าสูงกว่าค่าอ้างอิงปานกลาง ซึ่งอุณหภูมิกำลังคงที่หรือลดต่ำลง ดังนั้นค่า $u(k)$ จึงมีค่าเพิ่มขึ้นคือเป็น NS

4)ที่ค่า $e(k)$ เป็น PS และ $\Delta e(k)$ เป็น negativeกับ zero คืออุณหภูมิปัจจุบันที่มีค่าต่ำกว่าค่าอ้างอิงปานกลางซึ่งอุณหภูมิกำลังคงที่หรือเพิ่มขึ้น ดังนั้น $u(k)$ เป็น PS

5)ที่ค่า $e(k)$ เป็น PS และ $\Delta e(k)$ เป็น positive คือค่าอุณหภูมิปัจจุบันที่มีค่าต่ำกว่าค่าที่อ้างอิงปานกลาง ซึ่งอุณหภูมิกำลังลดลง ดังนั้น $u(k)$ จึงเป็น PB

6)ที่ค่า $e(k)$ เป็น Z คือค่าอุณหภูมิปัจจุบันมีค่าใกล้เคียงค่าอ้างอิง โดยเมื่อ $\Delta e(k)$ เป็น negative คืออุณหภูมิกำลังขึ้น $u(k)$ เป็น NS เมื่อ $\Delta e(k)$ เป็น zero คืออุณหภูมิคงที่ $u(k)$ เป็น Z และเมื่อ $\Delta e(k)$ เป็น positive คืออุณหภูมิกำลังลดลง $u(k)$ เป็น PS

โดยจะแสดงตารางกฎการควบคุมที่ออกแบบได้ดังตาราง 3.3 ดังนี้

$e \backslash \Delta e$	NB	NS	Z	PS	PB
NB	NB	NB	NB	NB	NB
NS	NB	NB	NS	NS	NS
Z	NS	NS	Z	PS	PS
PS	PS	PS	PS	PB	PB
PB	PB	PB	PB	PB	PB

ตารางที่ 3.3 แสดงการควบคุม

3.12 การออกแบบการทำงานของระบบควบคุมอีทีเตอร์

การทำงานของโปรแกรมส่วนนี้ จะเป็นการควบคุมความร้อนของตู้ให้ได้เท่ากับค่าที่ต้องการ โดยมีรูปแบบการทำงานคือ

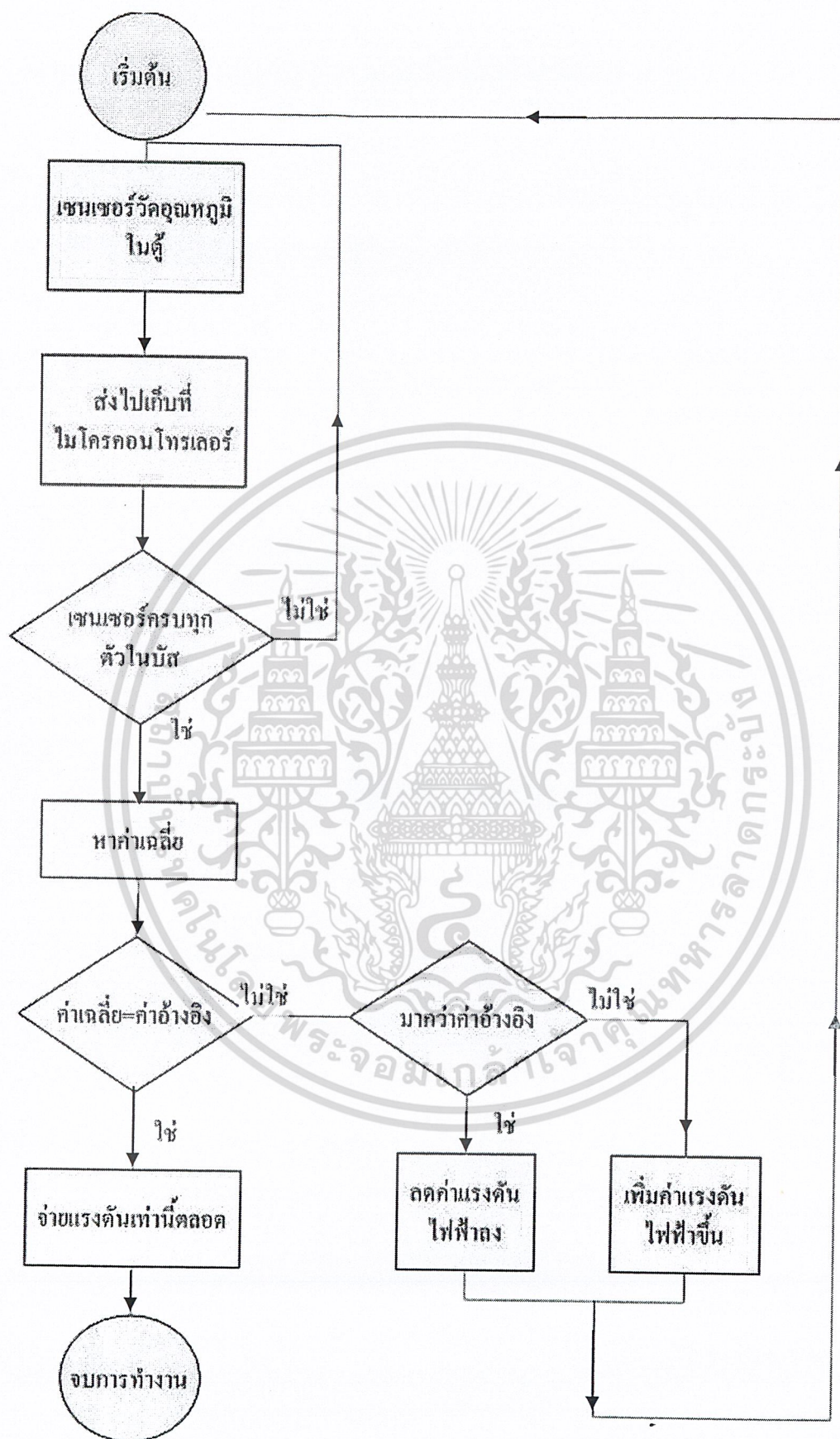
1. ต้องมีการกำหนดค่าอ้างอิงเข้าไปโดยป้อนเป็นเลข 2 หลักทางแป้นพิมพ์
2. ให้มีการอ่านค่าอุณหภูมิ 1 รอบก่อน เพื่อดูว่าอุณหภูมิตอนนี้เท่าไร นำค่าอุณหภูมิที่อ่านได้มา ลบกับค่าอ้างอิง จะได้ค่าความผิดพลาดออกมาตามสมการ 3.1 ซึ่งในขั้นตอนนี้จะยัง ไม่มีการทำงานใดๆ ของทั้งฮาร์ดแวร์ และฟัดลม ในการอ่านค่าครั้งแรกของเซนเซอร์จะมีค่าที่ผิดพลาด เพราะจะอ่านเอาค่าดีฟอลท์ออกมา จึงต้องกำหนดให้มีการอ่านค่าใหม่อีกรอบ
3. หน่วงเวลา ของการทำงานประมาณ 1.5 วินาทีและอ่านค่าอุณหภูมิอีกครั้ง และทำเช่นเดียวกับในข้อ 2 จะได้ค่าความผิดพลาดค่าใหม่ออกมาตามสมการที่ 3.2
4. นำค่าความผิดพลาด ที่เกิดขึ้นมาทำการลบกัน จะได้ค่าความเปลี่ยนแปลงความผิดพลาดที่เกิดขึ้น
5. นำค่าความผิดพลาดและค่าความเปลี่ยนแปลงความผิดพลาดที่เกิดขึ้นมาทำการปรับค่าแรงดันโดยดูว่าควรจะเป็นเพิ่มหรือลด
6. ถ้าค่าความผิดพลาด (err) มีค่าเป็นบวก (+) หมายถึง อุณหภูมิมีค่าน้อยกว่าจุดอ้างอิง ต้องเพิ่มแรงดันไฟฟ้าที่จ่ายให้กับฮีเตอร์เข้าไปอีก ถ้าเป็นลบ(-)หมายถึง อุณหภูมิมีค่ามากกว่าจุดอ้างอิง ต้องลดแรงดันไฟฟ้าที่จ่ายให้กับฮีเตอร์ลง ค่าเป็น ศูนย์ (0) หมายถึง อุณหภูมิมีค่าเท่ากับจุดอ้างอิง ให้คงค่าแรงดันไฟฟ้าในขณะนั้นๆ เอาไว้
7. ถ้าค่าความเปลี่ยนแปลงความผิดพลาด (c_err) มีค่าเป็นบวก (+) หมายถึงอุณหภูมิกำลังลด มีค่าเป็นลบ (-) หมายถึงอุณหภูมิกำลังเพิ่ม มีค่าเป็นศูนย์ (0) หมายถึงอุณหภูมิมีค่าคงที่ การดูค่าความเปลี่ยนแปลงความผิดพลาดจะดูคู่กับค่าความผิดพลาด และนำไปเขียนเป็นกฎในการทำงานแบบฟัซซี่
8. ระบบจะทำงานวนลูปอย่างนี้ตลอดเพื่อทำการปรับค่าอุณหภูมิจนกว่าอุณหภูมิจะเข้าใกล้ค่าอ้างอิง และนิ่งอยู่อย่างนั้นคล้ายรูปที่ 3.6 และเขียนการทำงานได้ดังรูปที่ 3.7

3.13 การออกแบบการทำงานของระบบควบคุมพัฒนา

การทำงานของโปรแกรมส่วนนี้จะเป็นการควบคุมอุณหภูมิของแต่ละบริเวณให้มีค่าเท่ากันโดยตลอดทั้งคู่ โดยมีการทำงานดังนี้

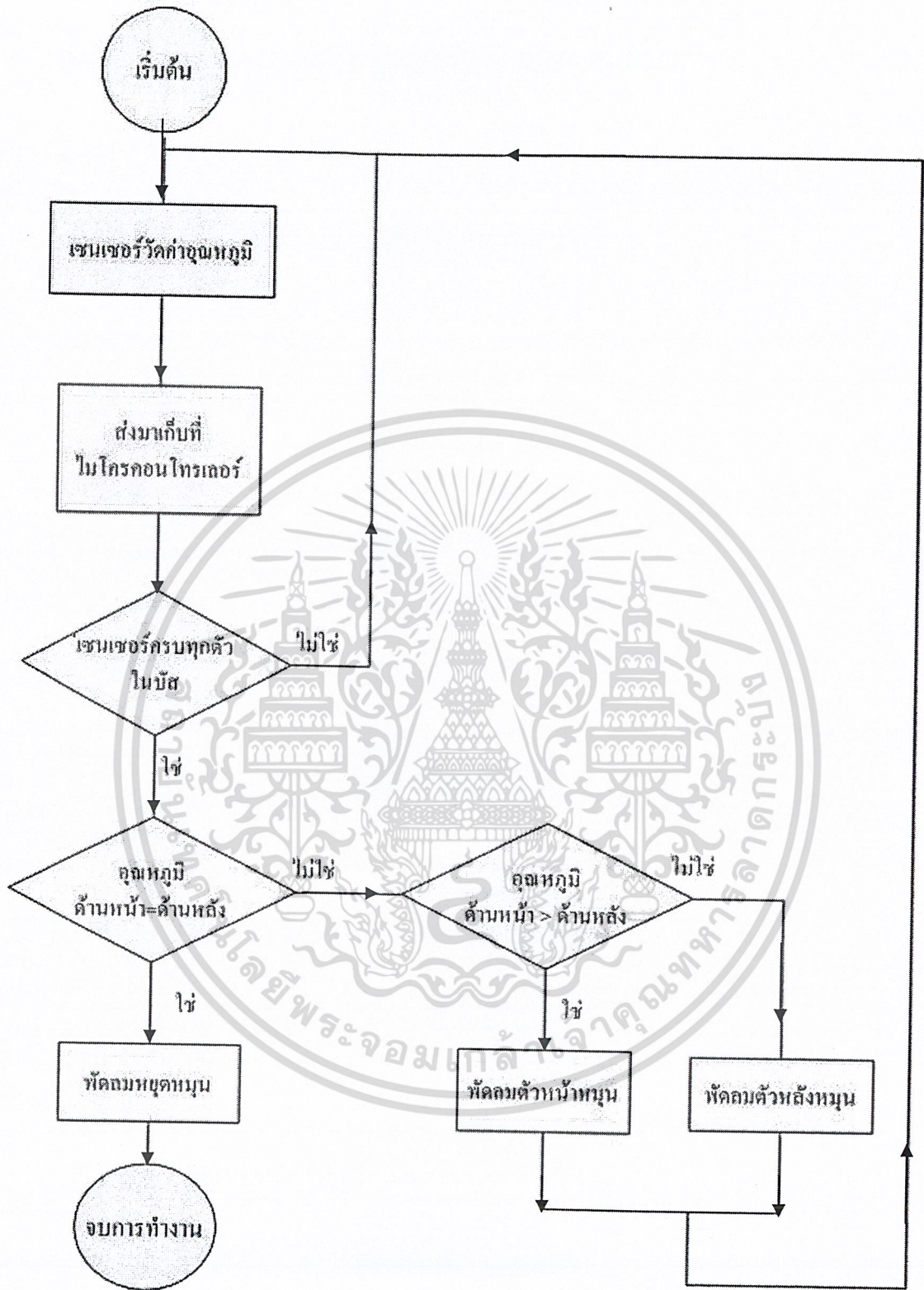
1. ค่าอุณหภูมิของเซนเซอร์ตำแหน่งหน้าตู้ มาเปรียบเทียบกับอุณหภูมิของเซนเซอร์ตำแหน่งหลังตู้ หากค่าความแตกต่างมากกว่า 1 องศาเซลเซียสก็ให้พัดลมทำงาน โดยดูว่าอุณหภูมิของบริเวณใดมีค่ามากกว่ากัน
2. ถ้าอุณหภูมิตำแหน่งหน้าตู้มีค่ามากกว่าก็ให้พัดลมตัวหน้าทำงานเพื่อช่วยเป่าอากาศร้อนไปยังบริเวณหลังตู้เร็วยิ่งขึ้น หรือถ้าอุณหภูมิตำแหน่งหลังตู้มีค่ามากกว่า ก็ให้พัดลมตัวหลังทำงานเพื่อช่วยเป่าอากาศร้อนไปยังบริเวณหน้าตู้เร็วยิ่งขึ้น
3. เมื่ออุณหภูมิมีค่าเท่ากันหรือ ต่างกันไม่เกิน 1 องศาเซลเซียสแล้วก็ให้พัดลมหยุดหมุน
4. ทำการตรวจสอบอย่างนี้ตลอดทุกช่วงเวลาที่ได้รับค่าอุณหภูมิเข้ามา
5. ในกรณีที่เกิดการเปลี่ยนค่าอ้างอิงใหม่ และค่าอ้างอิงมีค่าน้อยกว่าอุณหภูมิปัจจุบัน เกินกว่า 2 องศาเซลเซียส จะให้พัดลมทั้ง 2 ตัวหมุน เพื่อดึงอากาศจากภายนอกเข้ามาช่วยลดอุณหภูมิเร็วยิ่งขึ้น การทำงานแสดงในรูปที่ 3.8





รูปที่ 3.7 แผนผังแสดงการควบคุมอิทธิเตอร์

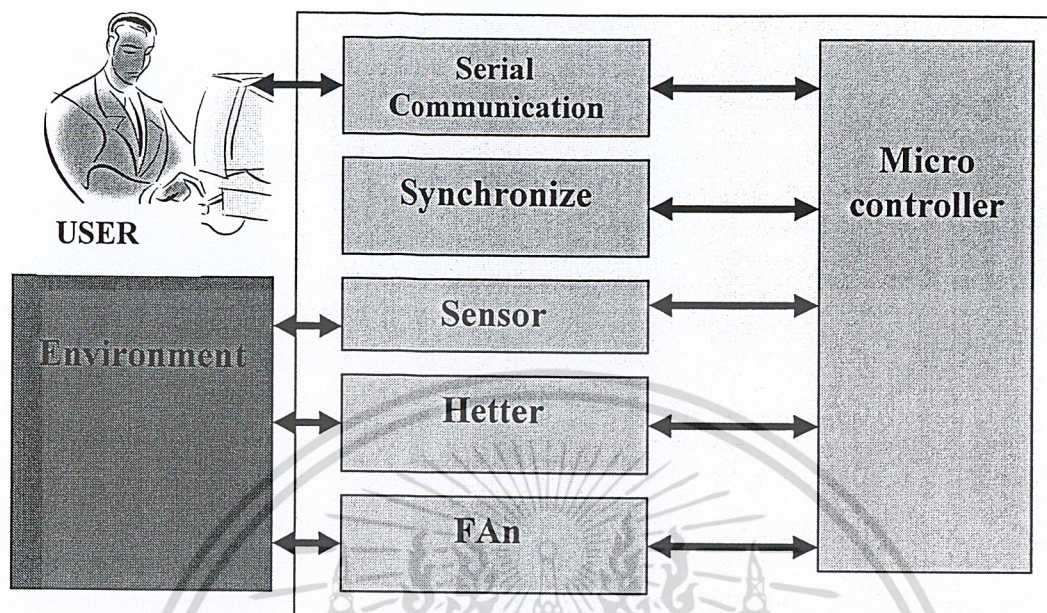
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.8 แสดงการควบคุมพัฒนา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.14 การออกแบบฮาร์ดแวร์ และ ซอฟต์แวร์



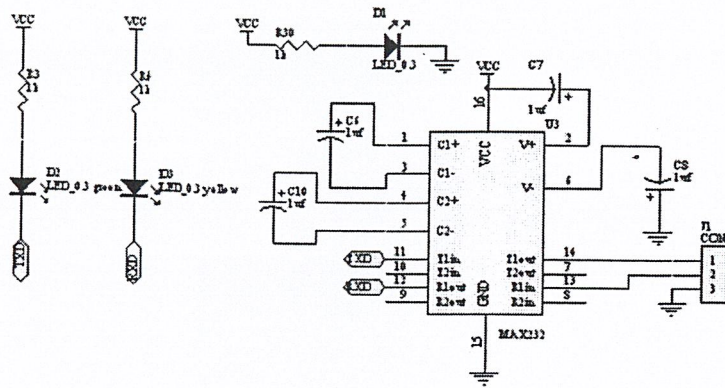
รูปที่ 3.9 โครงสร้างการทำงานของระบบ

3.14.1 ส่วนควบคุม

ใช้ไมโครคอนโทรลเลอร์ MCS-51 เบอร์ AT89C52 1 ตัวในการควบคุมการทำงานโดยมีการติดต่อกับ ส่วนควบคุมพัลลัมที่พอร์ท P2.0 – P2.5 โดย P2.0 – P2.3 เป็น ส่วนควบคุมขาอินพุต 1 – 4 และ P2.4 และ P2.5 เป็นส่วนควบคุมขาอินพุต ติดต่อกับพอร์ทอนุกรมที่พอร์ท P3.1(TxD)และ P3.0(RxD) ติดต่อกับส่วนรับอุณหภูมิที่พอร์ท P2.7 ส่วนอ้างอิงแรงดันไฟฟ้าที่พอร์ท P3.3 ที่เป็น พอร์ทอินเทอร์พท์ภายนอก (INT1)โดยส่วนนี้จะเป็นส่วนที่ทำงาน ร่วมกับอินเทอร์พท์ไทมเมอร์ ส่วนควบคุมสวิตเตอร์ ที่พอร์ท P1.3

3.14.2 ส่วนติดต่อพอร์ทอนุกรม

พอร์ทสื่อสารข้อมูลแบบอนุกรม เป็นพอร์ทภายในของไมโครคอนโทรลเลอร์ ซึ่งติดต่อได้ด้วยซอฟต์แวร์โดยใช้ไอซี MAX232 เป็นการสื่อสารแบบ RS-232 ในส่วนซอฟต์แวร์จะใช้การทำงานเป็น อินเทอร์พท์ การทำงานของซีเรียลพอร์ทจะทำงานโดยใช้ ไทมเมอร์ 1 ของ ไมโครคอนโทรลเลอร์ คือในภาวะปกติจะไม่มีการทำงาน แต่เมื่อมีสัญญาณจากพอร์ทซีเรียลเข้ามา หรือมีการส่งข้อมูลออกไป ก็จะกระโดดไปทำงานในส่วนอินเทอร์พท์นี้ และหยุดทำงาน เมื่อจบกระบวนการทำงาน ไปทำงานในโปรแกรมหลักต่อไป

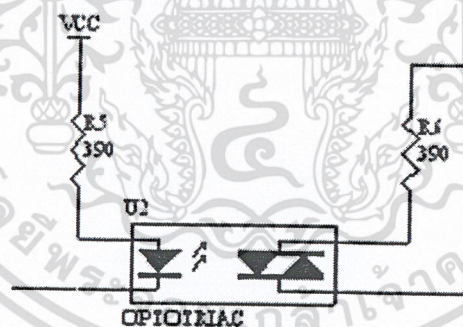


รูปที่ 3-10 แสดงวงจรติดต่อสื่อสารแบบอนุกรม RS-232

3.14.3 ส่วนแยกกราวด์ของวงจร

ออปโตคัปเปลอร์จะทำงานร่วมกับวงจรสับเบอร์และทำการป้องกันการกระโดดข้ามและหลีกเลี่ยงการใช้งานกราวด์ร่วมกันของวงจรแรงดันต่ำกับวงจรแรงดันสูง โดยต่อกับ ไตรแอกเพื่อทำงานร่วมกัน โดยส่วน ออฟโตไดรแอกนี้จะทำงานเมื่อมีการจ่ายค่าที่เป็น 0 ให้ และหยุดทำงานเมื่อ จ่าย 1 ให้กับ ออฟโตไดรแอก

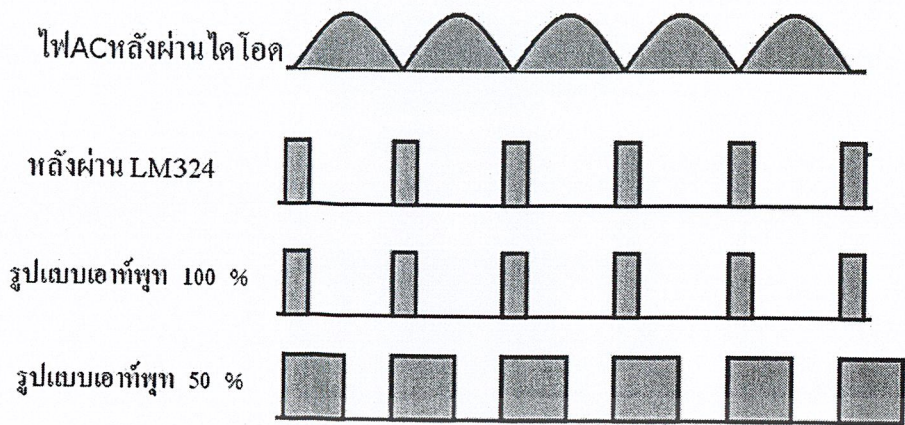
ในส่วนวอลต์แวย์ จะทำงานแบบจ่ายค่าจากไมโครคอนโทรลเลอร์แบบลูกคลื่นซึ่งความยาวของคลื่นจะมากหรือน้อยขึ้นอยู่กับอินพุตที่เข้ามาจากเซนเซอร์ และมาผ่านกฎของฟิซชีลอจิก



รูปที่ 3-11 แสดงวงจรแยกกราวด์แรงดันต่ำกับแรงดันสูงโดยใช้ออปโตคัปเปลอร์

3.14.4 ส่วนไตรแอก

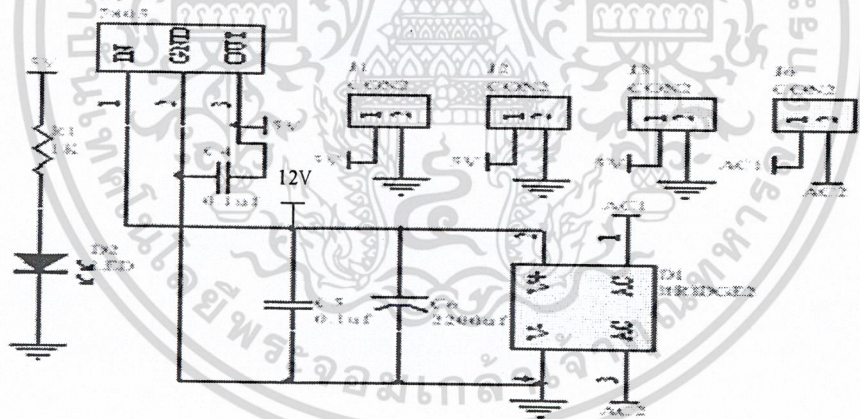
การควบคุมโหลดทางเอาท์พุทของไตรแอก จะมีขาเกต(G) เป็นตัวกระตุ้นกระแสเกตให้กับไตรแอก โดยจะมี R6 และ R7 กำหนดค่ากระแสร่วมกับขาเกต R8 และ C4 ทำหน้าที่เป็นวงจรสับเบอร์ (snubber) ซึ่งจะช่วยให้ไตรแอกทำงานได้อย่างเหมาะสมที่ช่วงครึ่งคลื่นสุดท้ายของวงจรการกระตุ้นเมื่อโหลดที่มาต่อกับเอาท์พุทเป็นพวกตัวเหนี่ยวนำเช่น หลอดไส้ เป็นต้น นอกจากนี้ยังมี C3 ทำหน้าที่เก็บออกเพื่อป้องกันสัญญาณรบกวนต่างๆ ที่จะรบกวนขณะไตรแอกทำงาน



รูปที่ 3.14 การปรับสัญญาณลอจิกสัญญาณที่จะใช้งานเป็นอินเวอร์ต

3.14.6 วงจรแหล่งจ่ายไฟ

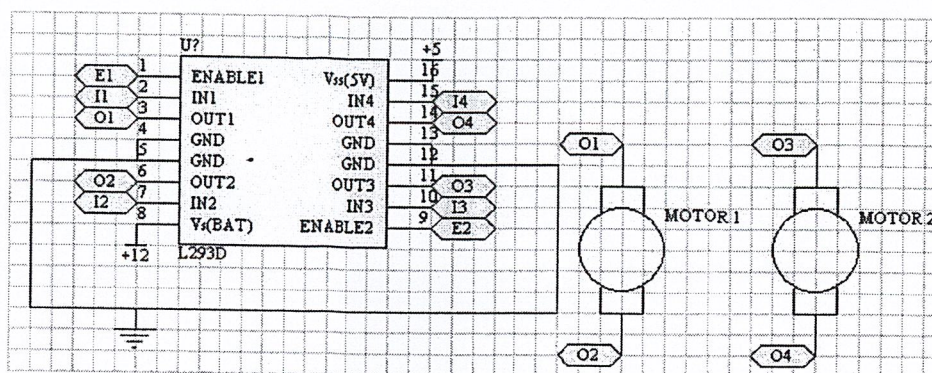
จะทำหน้าที่แปลงไฟกระแสสลับให้เป็นไฟกระแสตรง เพื่อจ่ายไปยังบอร์ดต่างๆ โดยจะมีไดโอดบริดจ์ทำหน้าที่เรียงกระแส แล้วผ่านตัวเก็บประจุขนาด 2,200 μF เพื่อทำให้แรงดันเรียบขึ้น โดยจะมีตัวเก็บประจุขนาดเล็กๆ 0.1 μF เพื่อกำจัดสัญญาณรบกวนทำให้แรงดันมีความคงที่มากขึ้น จากนั้นผ่านไอซี 7805 ซึ่งเป็นไอซีเรกูเลเตอร์ เพื่อให้ได้แรงดันเอาต์พุตเป็น 5 โวลต์ และมีการต่อไฟ 12 โวลต์ เพื่อไปจ่ายให้กับมอเตอร์



รูปที่ 3-15 แสดงวงจรเรกูเลเตอร์ 5 โวลต์ และ12โวลต์

3.14.7 วงจรควบคุมพัดลม

ทำหน้าที่ในการควบคุมพัดลมให้ถ่านเตาอากาศภายในตู้ โดยจะมีพัดลม 2 ตัว บริเวณหน้าตู้และหลังตู้ เมื่ออุณหภูมิบริเวณหน้าตู้และหลังตู้ ต่างกันมากกว่า 1 องศาเซลเซียส จะทำให้พัดลมบริเวณอุณหภูมิที่สูงกว่าทำงานเพื่อพัดอากาศไปบริเวณที่อุณหภูมิต่ำกว่าเพื่อปรับอุณหภูมิภายในตู้ ให้เท่ากันหรือมีความใกล้เคียงกัน การควบคุมจะเป็นจ่าย 5 โวลต์ไปที่ขาอินาเบลและหยุดเมื่อจ่ายค่า 0 โวลต์



รูปที่ 3.16 วงจรพัดลม

3.14.8 เซนเซอร์

ใช้การต่อเป็น 1 wire bus โดยต่อสายสัญญาณทั้งหมดเข้าที่ขาเพียงขาเดียว จะต้องมีการ पुलอัพสายสัญญาณด้วย ตัวต้านทาน ประมาณ 4700 โอห์ม การจ่ายไฟ 5 โวลต์ จะเป็นการจ่ายมาจากจุดเดียวกันหรือคนละจุดก็ได้

ในการติดตั้งเซนเซอร์จะติดตั้งให้มี ความเหมาะสมกับสภาพแวดล้อมนั้นๆ โดยดูที่ความกว้างและความจำเป็น โดยดูต้นแบบที่นำมาทดลองมีความจำเป็นที่ต้องควบคุมอุณหภูมิให้ได้เท่ากันตลอดทั้งตู้ จึงต้องมีการติดเซนเซอร์ไว้ 5 ตำแหน่ง เพื่อให้ทุกจุดมีค่าออกมาเท่ากัน เพื่อให้ไข่ได้รับความร้อนสม่ำเสมอ

ส่วนของซอฟต์แวร์

การทำงานในส่วนที่เป็น การอ่านค่าหมายเลขประจำตัว (ROM Code) ของตัวเซนเซอร์ โดยมีคำสั่งที่ใช้งานอยู่ 5 รูปแบบโดยอยู่ในรูปเลขฐาน 16 โดยคำสั่งที่ใช้งานมีอยู่ 2 แบบ คือ 33 h (Read Rom) และ 55h (Math Rom)

การทำงาน Read Rom

1. อุปกรณ์มาสเตอร์ ส่งสัญญาณ รีเซต โดยการส่ง 0 เป็นเวลา 480 -960 ไมโครวินาที
2. อุปกรณ์สเลฟ ตอบรับโดยการส่ง 0 เป็นเวลา 60- 120 ไมโครวินาที
3. อุปกรณ์มาสเตอร์ ส่งคำสั่ง 33h ออกไป
4. อุปกรณ์สเลฟ จะตอบค่า Rom ออกมาให้เป็นเลข 8 ไบต์ ไบต์แรกเป็น ตระกูล 6 ไบต์ต่อมา เป็น หมายเลขประจำตัว และ ไบต์สุดท้ายเป็น ไบต์ตรวจสอบความถูกต้องตามลำดับ ซึ่งแสดงการทำงานไว้ในรูปที่ 3.16

การทำงานของ Math Rom

1. อุปกรณ์มาสเตอร์ ส่งสัญญาณ รีเซตโดยการส่ง 0 เป็นเวลา 480 -960 ไมโครวินาที
2. อุปกรณ์สเลฟ ตอบรับโดยการส่ง 0 เป็นเวลา 60- 120 ไมโครวินาที
3. อุปกรณ์มาสเตอร์ ส่งคำสั่ง 55h ออกไป
4. อุปกรณ์มาสเตอร์ ส่งไบต์ 0 ไป เปรียบเทียบกับค่าที่อยู่ในอุปกรณ์สเลฟ ถ้าถูกต้องจะส่งไบต์ต่อไป แต่ถ้าผิดก็จะกลับไปเริ่มกระบวนการใหม่ ซึ่งแสดงการทำงานไว้ในรูปที่ 3.16

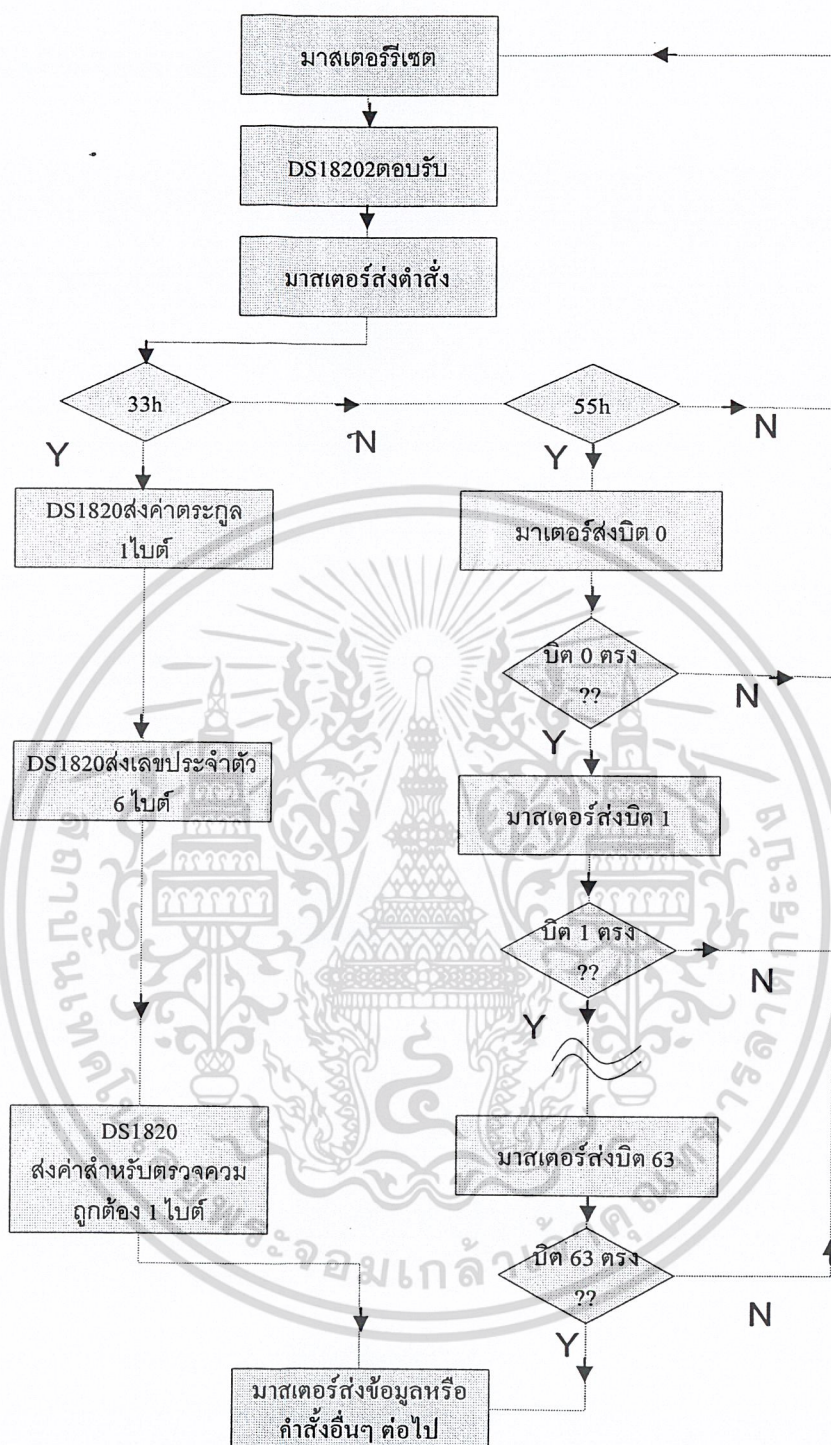
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทำงานของซอฟต์แวร์อีกส่วนหนึ่งคือการอ่านค่าอุณหภูมิ มีคำสั่งที่ใช้งานคือ 44 h คือการแปลงค่าอุณหภูมิมาเป็นข้อมูลดิจิทัล และ BEh คือการอ่านค่าที่ถูกเก็บเอาไว้ในสแควร์แพค มีรูปแบบการทำงาน คือ

1. อุปกรณ์มาสเตอร์ ส่งสัญญาณ รีเซต
2. อุปกรณ์สเลฟ ตอบรับ
3. อุปกรณ์มาสเตอร์ ส่งคำสั่ง 55h ออกไป สั่งให้อุปกรณ์ที่มีหมายเลขตรงกับที่สั่งทำงาน
4. อุปกรณ์มาสเตอร์ ส่งคำสั่ง 44h ออกไป แปลงค่าอุณหภูมิเป็นดิจิทัลแล้วเก็บค่าไว้ในสแควร์แพค
5. หน่วงเวลารอกการเปลี่ยนค่าอุณหภูมิเป็น ดิจิตอล อย่างน้อย 200 นาโนวินาที
6. อุปกรณ์มาสเตอร์ ส่งสัญญาณ รีเซต
7. อุปกรณ์สเลฟ ตอบรับ
8. อุปกรณ์มาสเตอร์ ส่งคำสั่ง 55h ออกไป
9. อุปกรณ์มาสเตอร์ ส่งคำสั่ง BEh ออกไปอ่านค่าจากสแควร์แพค
10. เลื่อนบิตไปทางขวา 1 ครั้ง เพื่อตัดทศนิยมออก โดยการทำงานแสดงดังรูปที่ 3.17

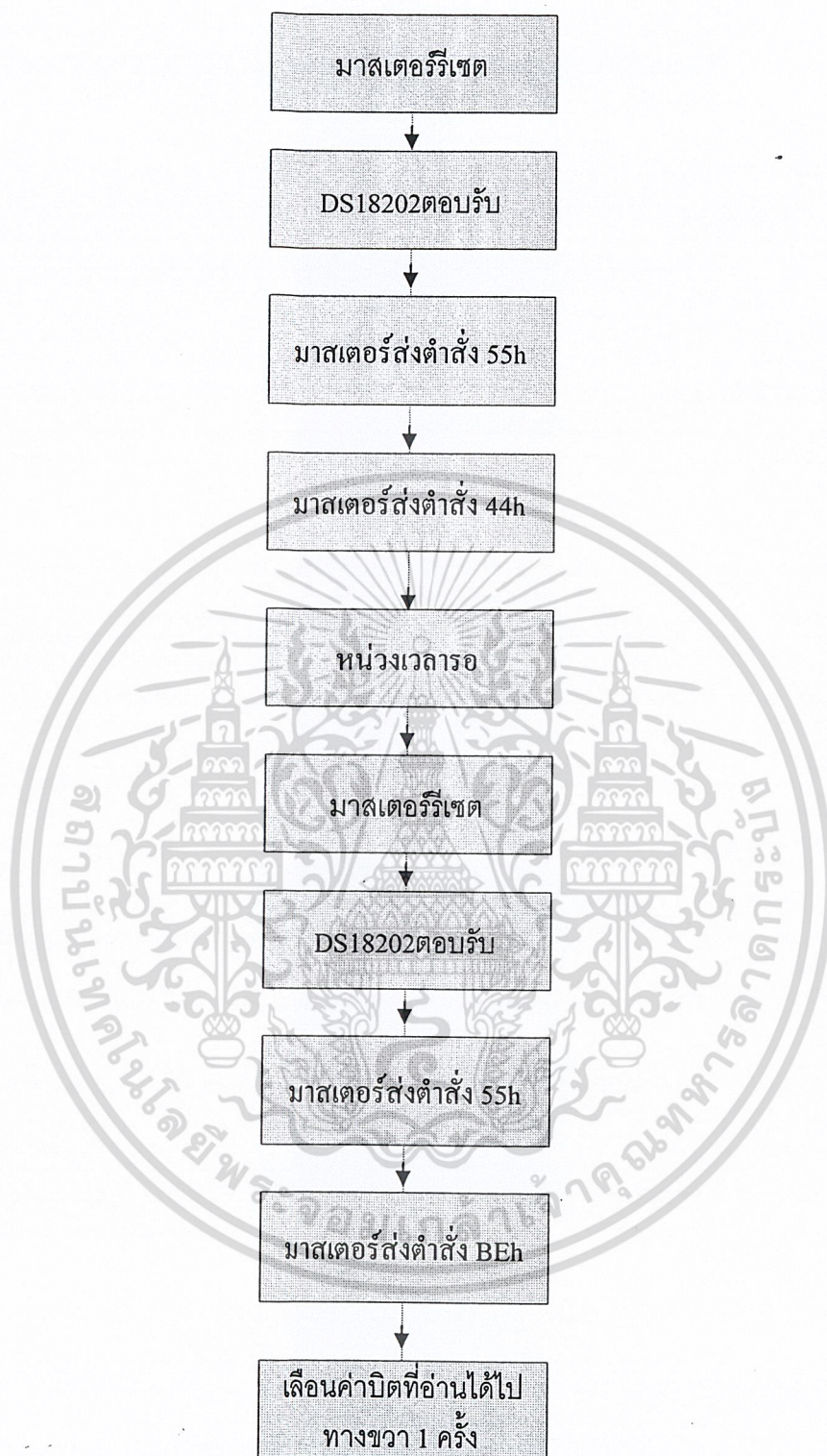


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



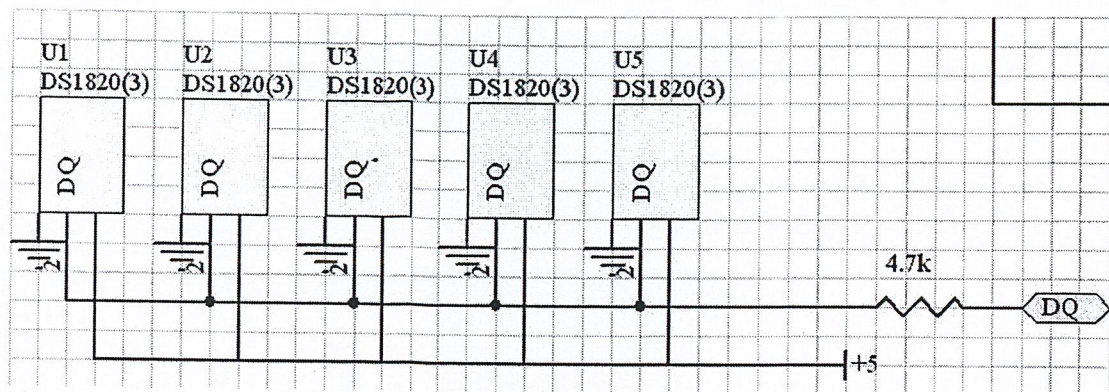
รูปที่ 3.17การทำงานของDS1820 ในส่วนของRom Code

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

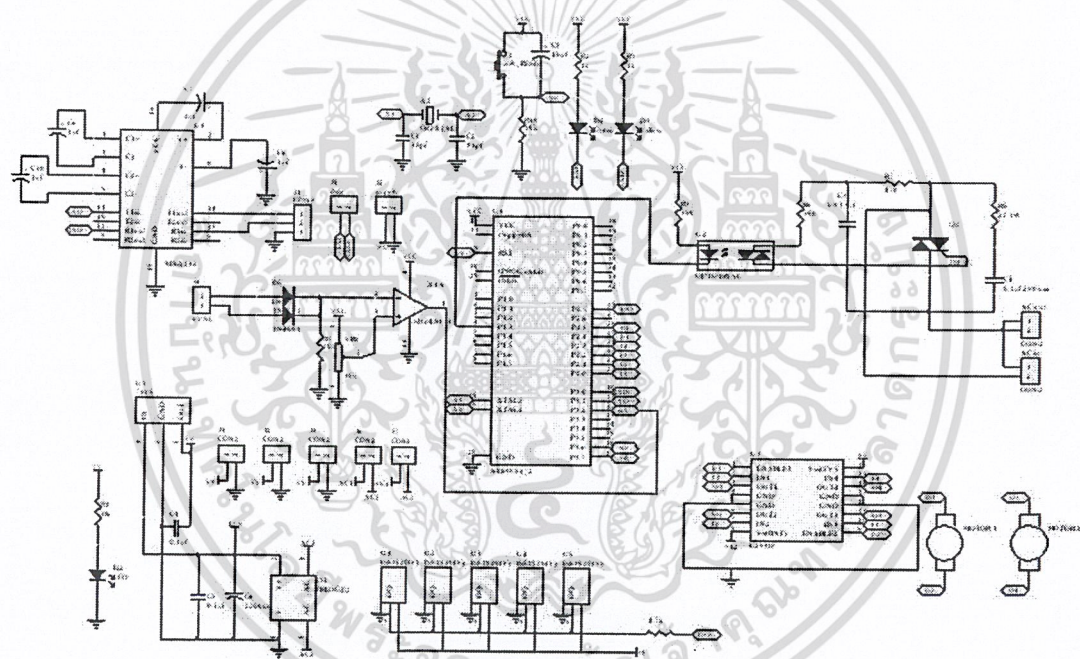


รูปที่ 3.18 การทำงานในการอ่านค่าอุณหภูมิ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.19 การต่อเซนเซอร์เป็นบัสแบบ 1-wire bus



รูปที่ 3.20 วงจรโดยรวมของระบบควบคุม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

การทดลองและการทดสอบระบบ

4.1 การทดลองการทำงาน

การทดสอบการอ่านค่ารอมโค้ดของเซนเซอร์

การทดสอบทำโดยการสั่งให้อ่านเลขประจำตัวออกมาเรื่อยๆ ดังรูปที่ 4.1

```
$ff16 ff00 ff08 ff00 ff92 ff6d ffa4 ff10 @  
$ff16 ff00 ff08 ff00 ff92 ff6d ffa4 ff10 @
```

รูปที่ 4.1 รอมโค้ดที่สามารถอ่านได้จากอุปกรณ์

การทดสอบการอ่านค่าอุณหภูมิของเซนเซอร์

การทดสอบทำโดยให้อ่านค่าอุณหภูมิออกมาเรื่อยๆ และเมื่อ นำสิ่งของที่มีอุณหภูมิสูงกว่า เช่นนี้ว
มือไปแตะที่ตัวเซนเซอร์จะเห็นการทำงานโดยค่าอุณหภูมิจะค่อยๆ เพิ่มขึ้นดังรูป ที่ 4.2

```
$ff16 ff00 ff08 ff00 ff92 ff6d ffa4 ff10 @  
Temp1 22 degrees C  
$ff16 ff00 ff08 ff00 ff92 ff6d ffa4 ff10 @  
Temp1 23 degrees C  
$ff16 ff00 ff08 ff00 ff92 ff6d ffa4 ff10 @  
Temp1 24 degrees C  
$ff16 ff00 ff08 ff00 ff92 ff6d ffa4 ff10 @  
Temp1 25 degrees C
```

รูปที่ 4.2 การทดสอบการอ่านค่าของเซนเซอร์

การทดสอบการทำงานของพัดลม

ทำการทดสอบโดยการ เขียนโปรแกรม ให้พัดลมทำงานเมื่อมีการ ใส่ค่าเข้าไป ดังรูปที่ 4.3

```
Press 2  
Fan 2 on  
Press 4  
Fan off  
Press 1  
Fan 1 on  
Press 2  
Fan 2 on  
Press 4  
Fan off
```

รูปที่ 4.3 ทดสอบการทำงานของพัดลมเมื่อกดเลขต่างๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทดสอบการทำงานของฮีเตอร์

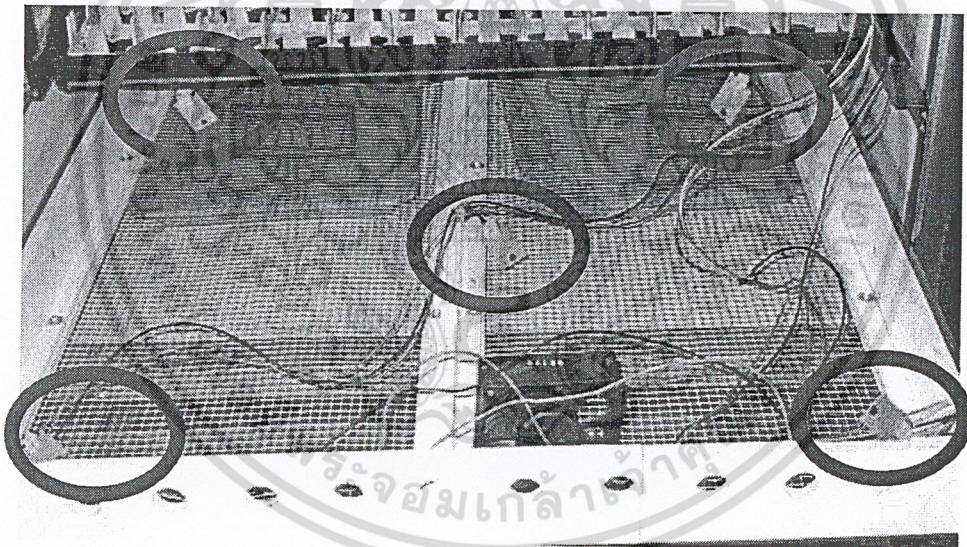
ทำการทดสอบโดยการเขียนโปรแกรม รับค่า ถ้าค่าดังกล่าวตรงกับที่ตั้งเอาไว้ก็ให้แสดงตามนั้น
ดังรูปที่ 4.4

Press 1
Light on
Press 1
Light on
Press 4
light off
Press 1
Light on

รูปที่ 4.4 การทดสอบการทำงานของฮีเตอร์โดยใช้หลอดไฟ

4.2 ขั้นตอนการติดตั้งระบบ

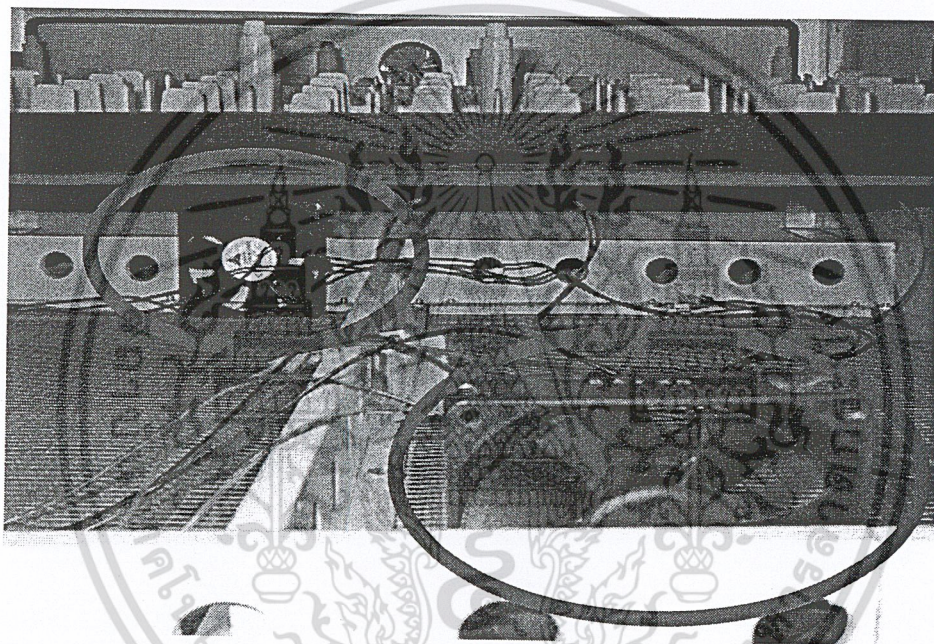
1. ติดตั้งเซนเซอร์ ทั้ง 5 จุดตามที่ออกแบบไว้ดังรูป 4.5
2. ติดตั้งชุดควบคุมเข้ากับฮีเตอร์ ดังรูป 4.6
3. ติดตั้งพัดลมดังรูป 4.7



รูปที่ 4.5 แสดงการวางตำแหน่งของเซนเซอร์ทั้ง 5 จุด



รูปที่ 4.6 แสดงการเชื่อมต่อวงจรสำหรับควบคุม ฮีทเตอร์



รูปที่ 4.7 แสดงการติดตั้งพัดลม

4.3 การทดลองระบบ

ในส่วนนี้ได้มีการใช้โปรแกรมเข้ามาควบคุมอุณหภูมิของห้อง ซึ่งจะแบ่งออกเป็น 5 กรณีคือ

1 อุณหภูมิต่ำกว่าจุดอ้างอิงมาก

ระดับของการทำงานของฮีทเตอร์จะสูงมากเพื่อปรับค่าความร้อนให้มากขึ้น โดยเร็วซึ่งที่ตั้งค่าเอาไว้คือมากกว่าหรือ เท่ากับ -4 ถึง -3 องศาเซลเซียส

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Temp1 24 degrees C
 Temp2 25 degrees C
 Setpoint = 35 degrees C
 CHLevel 7
 Stop

Temp1 24 degrees C
 Temp2 25 degrees C
 Setpoint = 35 degrees C
 CHLevel 7
 Stop

รูปที่ 4.8 ทดสอบการทำงานของฮีตเตอร์เมื่ออุณหภูมิต่ำกว่าจุดอ้างอิงมาก

2 อุณหภูมิต่ำกว่าจุดอ้างอิงไม่มาก

ระดับของการทำงานของฮีตเตอร์จะสูงปานกลางเพื่อปรับค่าความร้อนให้เพิ่มขึ้นเพียงเล็กน้อย ซึ่งที่ตั้งค่าเอาไว้คือ -3 ถึง -1 องศาเซลเซียส

Temp1 24 degrees C
 Temp2 25 degrees C
 Setpoint = 27 degrees C
 CHLevel 5
 Stop

Temp1 24 degrees C
 Temp2 25 degrees C
 Setpoint = 27 degrees C
 CHLevel 5
 Stop

รูปที่ 4.9 ทดสอบการทำงานของฮีตเตอร์เมื่ออุณหภูมิต่ำกว่าจุดอ้างอิงไม่มาก

3 อุณหภูมิอยู่ที่จุดอ้างอิง

ระดับของการทำงานของฮีตเตอร์จะปานกลางเพื่อปรับค่าความร้อนให้คงที่อย่างนั้นซึ่งที่ตั้งค่าเอาไว้คือมากกว่าหรือเท่ากับ -1 ถึง 1 องศาเซลเซียส

Temp1 24 degrees C
 Temp2 25 degrees C
 Setpoint = 26 degrees C
 CHLevel 4
 Stop

Temp1 24 degrees C
 Temp2 25 degrees C
 Setpoint = 26 degrees C
 CHLevel 4
 Stop

รูปที่ 4.10 ทดสอบการทำงานของฮีตเตอร์เมื่ออุณหภูมิอยู่ที่จุดอ้างอิง

4 อุณหภูมิสูงกว่าจุดอ้างอิงไม่มาก

ระดับของการทำงานของฮีตเตอร์จะต่ำปานกลางเพื่อปรับความร้อนให้น้อยลงอย่างช้าๆ ซึ่งที่ตั้งค่าเอาไว้คือมากกว่าหรือ เท่ากับ 1 ถึง 2 องศาเซลเซียส

Fan1 and Fan2

Temp1 25 degrees C
 Temp2 25 degrees C
 Setpoint = 23 degrees C
 CHLevel 1
 Stop

Fan1 and Fan2

Temp1 25 degrees C
 Temp2 25 degrees C
 Setpoint = 23 degrees C
 CHLevel 1
 Stop

Fan1 and Fan2

รูปที่ 4.11 ทดสอบการทำงานของฮีตเตอร์เมื่ออุณหภูมิสูงกว่าจุดอ้างอิงไม่มาก

5 อุณหภูมิสูงกว่าจุดอ้างอิงมาก

ระดับของการทำงานของฮีตเตอร์จะสูงเพื่อปรับค่าความร้อนให้มากขึ้นโดยเร็วซึ่งที่ตั้งค่าเอาไว้คือ 3 องศาเซลเซียส ขึ้นไป และยังได้ตั้งค่าไว้อีกว่า หากอุณหภูมิ สูงกว่าค่าที่ตั้งไว้มาก ก็จะทำให้พัดลมทั้ง 2 ตัวทำงานเพื่อระบายอากาศให้ดีขึ้นอีกด้วย โดยค่าที่ตั้งเอาไว้คือ มากกว่า 2 องศาเซลเซียส


```
Temp1 24 degrees C
Temp2 25 degrees C
Setpoint = 22 degrees C
CHLevel 0
Stop
```

Fan1 and Fan2

```
Temp1 25 degrees C
Temp2 25 degrees C
Setpoint = 22 degrees C
CHLevel 0
Stop
```

Fan1 and Fan2

รูปที่ 4.12 ทดสอบการทำงานของฮีตเตอร์เมื่ออุณหภูมิสูงกว่าจุดอ้างอิงมาก

การตั้งค่าอ้างอิงเพื่อกำหนดระดับของฮีตเตอร์

การกำหนดค่าอ้างอิงสามารถทำในครั้งแรกที่เปิดระบบโดยจะแสดงค่าที่ใส่เข้าไป และหากต้องการเปลี่ยนค่า ก็ทำการกด “r” ก็จะทำการรีเซตใหม่เพื่อป้อนค่าอ้างอิงใหม่ ดังแสดงในรูปที่ 4.4 และรูปที่ 4.5

```
enter setpoint
Setpoint = 40 degrees C
```

รูปที่ 4.13 ตั้งค่าอุณหภูมิที่ต้องการควบคุม

```
Temp1 33 degrees C
Temp2 34 degrees C
Setpoint = 35 degrees C
CHLevel 4
Stop
```

enter setpoint

รูปที่ 4.14 แสดงการตั้งค่าใหม่ เมื่อกด “r”

การทำงานของระบบโดยรวม

ในการทำงาน จะมีการแสดงสถานะต่างๆ ออกมาให้ผู้ใช้งานสามารถดูได้ โดยจะแสดง อุณหภูมิ ณ จุดที่ต่างๆ ค่าที่ติดตั้งเอาไว้ ค่าอ้างอิงที่ตั้งเอาไว้ ระดับของฮีตเตอร์ และสถานะการทำงานของพัดลม ดัง

รูปที่ 4.15 การทำงานของพัดลมจะทำงานเมื่ออุณหภูมิของน้ำตู้กับหลังตู้ต่างกันมากกว่า 1 องศาตามรูปที่เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.16 การปรับระดับไฟ ของฮีทเตอร์ จะปรับจากค่าความต่างของอุณหภูมิกับจุดอ้างอิง ถ้าอุณหภูมิเข้าใกล้ค่าอ้างอิงมาก ระดับไฟก็จะลดลง ดังรูปที่ 4.17 ถ้าหากเรามีการเปลี่ยนค่าอ้างอิง และค่าอุณหภูมิมากกว่าค่าอ้างอิงใหม่ มากกว่า 2 องศาเซลเซียส พัฒมทั้ง 2 ตัวจะทำงานเพื่อปรับค่าอุณหภูมิลงดังรูปที่

4.18

```
Temp1 26 degrees C
Temp2 26 degrees C
Setpoint = 40 degrees C
CHLevel 6
Stop
```

```
Temp1 26 degrees C
Temp2 27 degrees C
Setpoint = 40 degrees C
CHLevel 6
Stop
```

รูปที่ 4.15 แสดงอุณหภูมิณจุดต่างๆ ค่าอ้างอิง ระดับของไฟ และสถานะการทำงานของพัฒม

```
Temp1 32 degrees C
Temp2 35 degrees C
Setpoint = 40 degrees C
CHLevel 6
Fan2
```

```
Temp1 32 degrees C
Temp2 35 degrees C
Setpoint = 40 degrees C
CHLevel 6
Fan2
```

รูปที่ 4.16 แสดงการทำงานของพัฒมเมื่อ อุณหภูมิต่างกันมากกว่า 1 องศาเซลเซียส

```
Temp1 36 degrees C
Temp2 38 degrees C
Setpoint = 40 degrees C
CHLevel 5
Fan2
```

```
Temp1 36 degrees C
Temp2 39 degrees C
Setpoint = 40 degrees C
CHLevel 2
Fan2
```

รูปที่ 4.17 แสดงการปรับระดับของฮีทเตอร์เมื่ออุณหภูมิเข้าใกล้ค่าที่ตั้งไว้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Temp 25 degrees C
 Temp 26 degrees C
 Setpoint = 20 degrees C
 CHLevel 0
 Stop

Fan1 and Fan2

รูปที่ 4.18 แสดงการทำงานของพัดลมทั้ง 2 ตัวเมื่ออุณหภูมิมากกว่าค่าอ้างอิงมากๆ



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

บทวิจารณ์และสรุป

5.1 สรุปและวิจารณ์

ระบบที่นำมาใช้มีการคิดคำนวณอย่างมีแบบแผนเพื่อให้เหมาะสมกับวัตถุประสงค์ของการนำไปใช้งาน สามารถปรับเปลี่ยนอุปกรณ์ต่างๆ ให้เหมาะสมได้ เช่น หากใช้ห้องที่ใหญ่ ก็อาจจะต้องใช้ฮีทเตอร์ ตัวใหญ่ๆ พัดลมหลายๆ ตัว เป็นต้น ซึ่งมีความหลากหลายในการใช้งานกว่าระบบแบบเก่าที่ใช้งานอยู่ อีกทั้งยังสามารถเพิ่มอุปกรณ์เข้าไปได้โดยง่ายโดยการต่อ ส่วนคอนโทรลที่เป็น สเตปเข้ามาอีก ก็จะสามารถเพิ่มการทำงานได้แล้ว

ข้อ	ระบบเดิม	ระบบใหม่
1	การควบคุมระบบยังไม่ดีนักเพราะอาศัยการตัดไฟ จากรีเลย์	ควบคุมไฟได้ดียิ่งขึ้นด้วยการคำนวณค่าอุณหภูมิออกมาแล้วจึงไปคิดเรื่องไฟที่จะต้องจ่ายให้กับระบบทำให้อุณหภูมิมีความแน่นอนมากกว่า
2	การระบายอากาศยังไม่ดีนัก บางจุดอุณหภูมิยังไม่ถึงหรือ ไปถึงไม่สม่ำเสมอ	มีการระบายอุณหภูมิสม่ำเสมอเพราะมีการเพิ่มพัดลมเข้าไปอีก เพื่อช่วยให้การไหลของอากาศดีขึ้น
3	มีเซนเซอร์เพียง 2 จุด ด้านหน้าตู้ ทำให้ไม่สามารถรู้ได้ว่าอุณหภูมิด้านหลังตู้ได้รับความร้อนเพียงพอหรือ มากเกินไปหรือไม่	มีเซนเซอร์หลายจุด สามารถดูได้ว่าที่จุดไหนมีอุณหภูมิเท่าใด เพื่อที่สามารถปรับอุณหภูมิที่จุดนั้นได้อย่างถูกต้อง
4	การควบคุมอุณหภูมิด้วยการเปิด - ปิด บ่อยๆ จะทำให้ฮีทเตอร์มีอายุการใช้งานที่สั้น	ควบคุมฮีทเตอร์ด้วยการเพิ่ม - ลดการจ่ายไฟให้กับฮีทเตอร์ ทำให้ฮีทเตอร์มีอายุการใช้งานนานขึ้น
5	ผลผลิตไม่ได้รับความต่อเนื่องของอุณหภูมิ เพราะ ค่าอุณหภูมิมีการแกว่งที่มากเกินไป	ได้รับอุณหภูมิสม่ำเสมอ เพราะ อุณหภูมิจะถูกรักษาระดับด้วยระบบไม่ให้เปลี่ยนไปมามาก
6	ไม่สามารถปรับเปลี่ยนอุปกรณ์ได้ง่ายนัก	สามารถปรับเปลี่ยนอุปกรณ์ทุกตัวได้อย่างง่ายดายตามเมื่อต้องการเปลี่ยนสภาวะแวดล้อมที่ต้องการควบคุม
7	การเพิ่มอุปกรณ์ทำได้ยาก	การเพิ่มอุปกรณ์ทำได้ง่ายเพียงต่อไมโครคอนโทรลเลอร์ที่เป็น สเตปเพิ่มเข้ามาเมื่อไมโครคอนโทรลเลอร์ที่เป็น มาสเตอร์ทำงานได้ไม่พอกับความต้องการ

ตารางที่ 5.1 เปรียบเทียบระบบเดิมกับระบบใหม่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การเลือกใช้ระบบที่เป็นบัสก์เพื่อ ลดการใช้ สายสัญญาณและจำนวนพอร์ทัลที่ต้องต่อเข้ามา แต่จะมีปัญหาในภายหลังในเรื่องคุณภาพของสายต้องดีมาก ถึงจะสามารถใช้งานในกรณีที่ต้องต่อไกลๆ

การปรับเปลี่ยนสถานที่ที่ต้องการควบคุมต้องมีการปรับเปลี่ยนอุปกรณ์บางอย่าง เช่นถ้าต้องการนำไปใช้ในห้องที่มีขนาดใหญ่ขึ้นก็ควยเปลี่ยนฮีทเตอร์ให้ใหญ่ขึ้นตามไปควย

5.2 ปัญหาที่เกิดขึ้น

1. ปัญหาในการคิดอัลกอร์ทึมสำหรับใช้งานเนื่องจากมีความหลากหลายในการใช้ โดยอัลกอร์ทึมที่เลือกใช้ในตอนแรก ไม่สามารถตอบสนองความต้องการได้ เพราะ ในแบบแรกที่ศึกษาจะเป็น การเลือกใช้อินพุตที่มีหลายค่า แล้วออกเป็นเอาต์พุต 1 ค่า แต่ในระบบที่ออกแบบมีอินพุตเพียง 1 ค่า จึงต้องทำการศึกษาใหม่

2. การใช้อินเทอร์พรัท ในการควบคุมฮีทเตอร์ จะมีผลต่อค่าดีเลย์ของตัวเซนเซอร์ ทำให้ค่าอุณหภูมิที่อ่านได้มีค่าผิดเพี้ยนออกไป ซึ่งทำการปิดอินเทอร์พรัทช่วงนี้ไป มีผลให้การจ่ายไฟ ไปที่ฮีทเตอร์ไม่สม่ำเสมอ เกิดอาการหลอดติดๆ ดับๆ ในช่วงเวลาที่อ่านอุณหภูมิ ซึ่งเราอาจแก้ไขได้โดยเพิ่มไมโครคอนโทรเลอร์ที่เป็น สเตป ไร้วัดควบคุมไฟ ของฮีทเตอร์ต่างหาก โดยให้ไปรับค่าที่ถูกคิดจากไมโครคอนโทรเลอร์ที่เป็นมาสเตอร์แทน

3. การได้ค่าเอาพุตที่ไม่สม่ำเสมอ เพราะไฟกระแสสลับที่ใช้อ้างอิงสัญญาณไม่ได้เท่ากันทุกค่า จำเป็นต้องทำให้เป็นค่าเฉลี่ย ออกมาเสียก่อน

4. ความไม่เชี่ยวชาญในการควบคุมเพราะ พัจฉีเป็นระบบที่ต้องอาศัยความเชี่ยวชาญและการทดสอบหลายๆ ครั้ง โดยฮีทเตอร์เป็นอุปกรณ์ที่ให้ความร้อนช้าทำให้ปัญหาในการเก็บค่าที่เปลี่ยนแปลงหรือการทดสอบอื่นๆ

5. การใช้สายสัญญาณที่ไม่ดีพอจะไม่สามารถทำงานได้เมื่อมีการต่อไปในระยะทางไกลๆ ซึ่งจะเป็นข้อจำกัดในการนำไปใช้งานถ้าต้องการต่อให้ไกลออกไป โดยที่เคยใช้สายทั่วไป สามารถใช้ได้เพียง 15 เมตรเท่านั้น

6. ควรมีการศึกษาฮีทเตอร์ที่นำมาใช้ให้ดีเสียก่อนว่าสามารถให้อุณหภูมิได้ตามช่วงที่ต้องการหรือไม่ เพราะ ในกรณีของระบบนี้ทดสอบที่อุณหภูมิทั่วไป แล้วพบว่าใช้ได้ แต่พอได้ทดสอบในห้องที่มีความเยือกอย่างห้องแอร์ ปรากฏว่าให้อุณหภูมิไม่ถึงค่าที่ต้องการ

5.4 แนวทางในการพัฒนาต่อไป

1. การใช้ชุดควบคุมนี้ไปควบคุมสภาวะแวดล้อมอื่นๆ เพิ่มโดยเพิ่มชุดควบคุมแบบต่างๆเข้ามา เช่น การตรวจสอบความชื้น ความเข้มของแสง ความหนาแน่นของควัน

2. เพิ่มความสามารถของระบบที่ตรวจสอบให้สามารถดูและอย่างใกล้ชิด และดูผลได้ตลอดเวลา

หรือเก็บเป็นไฟล์เพื่อนำไปแสดงผลที่คอมพิวเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3 . การแสดงผลเป็นกราฟ เพื่อดูความต่อเนื่องของอุณหภูมิ หรือความเปลี่ยนแปลงต่างๆ ของอุณหภูมิ ณ สภาวะต่างๆ กัน



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หนังสืออ้างอิง

- [1] MCS-51 Flash Microcontroller ฉบับ AT89C5x ของ Almel
- [2] . การพัฒนาไมโครคอนโทรเลอร์ด้วยภาษาซี ผู้ช่วยศาสตราจารย์ธีรวัฒน์ประภผล



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาพผนวก



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้


```

#include <Atmel/AT89X51.h>
#include <stdio.h>
#include <intrins.h>
#include <math.h>

// GLOBAL VARIABLE - and define
#define BUFFER_SIZE 3
#define MAX_LEVEL 7 // Number of level - 1
#define divide 32

sbit CH = 0x93;
sbit sync=0xB3;
sbit DQ = 0xA7;

data unsigned short PhaseLen; // 'ON' Phase length (in cycles unit)
data unsigned short StepLen; // How long does a level runs (cycles)
data unsigned short MapTable[8]; // Each level cycles
data unsigned short Remain;
data unsigned char CHLevel = 0; // Level of channel
data unsigned char TmpLev = 0;
data unsigned char start=0;
idata unsigned char buffer[BUFFER_SIZE];
//unsigned char ff = 0xff;
int i,j,k;
int loop = 0;
bit loaded = 0;
char set_p[1];

// SERIAL - Get SBUF to buffer
void receive(void) interrupt 4
{
    if(RI)
    {
        RI = 0;
        if (loop < 1)
        {
            buffer[loop] = SBUF;
            loop++;
        }
        else
        {
            buffer[loop] = SBUF;
            loop = 0;
            loaded = 1;
        }
    }
}

// GET_DATA - get data from serial port
int get_data(char* co)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้


```

//unsigned char a[16];

if (loaded==1)
{
    co[0] = ((buffer[0] - 0x30) * 10) + (buffer[1]-0x30);
    loaded=0;
    //TI = 1;
    printf("Setpoint = %d degrees C \n ",(int)co[0]);
    TI = 0;
    return(0);
}
return(1);
}

// DELAY - with an 11.059MHz crystal. Calling 24us, and count 16us.
void delay(int seconds) {
    for (k=0; k<seconds; k++);
}

// OW_RESET - Reset the one-wire bus and returns the presence detect.
unsigned char ow_reset(void) {
    unsigned char presence;

    DQ = 0; //pull DQ line low
    delay(29); // leave it low for 480us
    DQ = 1; // allow line to return high
    delay(3); // wait for presence
    presence = DQ; // get presence signal
    delay(25); // wait for end of timeslot
    return(presence); // presence signal returned
} // 0=presence, 1 = no part

// READ_BIT - reads a bit from the one-wire bus.
unsigned char read_bit(void) {
    bit dat;
    DQ = 0; // pull DQ low to start timeslot
    _nop_ ();
    _nop_ ();
    DQ = 1; // then return high
    _nop_ ();
    _nop_ ();
    _nop_ ();
    _nop_ ();
    dat = DQ;
    for(k=0; k<8; k++) _nop_(); // delay 15us from start of timeslot
    return(dat); // return value of DQ line
}

// WRITE_BIT - writes a bit to the one-wire bus, passed in bitval.
void write_bit(char bitval) {
    DQ = 0; // pull DQ low to start timeslot
    if(bitval==1) DQ =1; // return DQ high if write 1
    delay(5); // hold value for remainder of
timeslot

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้


```

    DQ = 1;
}
// READ_BYTE - reads a byte from the one-wire bus.
unsigned char read_byte(void) {
    unsigned char value = 0;

    for (j=0; j<8; j++) {
        if(read_bit()) value|=0x01<<j; // reads byte in, one byte at a time and
then shifts it left
        delay(6); // wait for rest of timeslot
    }
    return(value);
}
// WRITE_BYTE - writes a byte to the one-wire bus.
void write_byte(char val) {
    unsigned char temp;

    for (j=0; j<8; j++) { // writes byte, one bit at a time
        temp = val>>j; // shifts val right 'j' spaces
        temp &= 0x01; // copy that bit to temp
        write_bit(temp); // write bit in temp into
    }
    delay(5);
}
// READ_TEMP - read temperature by skiprom
char Read_Temp(char* sensor) {
    char get[2]; //secode[8];
    char temp_lsb,temp_msb;

    ow_reset();
    write_byte(0x55);
    // math ROM
    for (i=0;i<8;i++) write_byte(sensor[i]);
    write_byte(0x44);
    // Convert T
    ES = 1;
    delay(20000);
    ES = 0;
    ow_reset();
    write_byte(0x55);
    // math ROM
    for (i=0;i<8;i++) write_byte(sensor[i]);
    write_byte(0xBE);
    // Read Scratch Pad
    for (i=0; i<2; i++) get[i]=read_byte();
    temp_msb = get[1];
    // Sign byte + lsb
    temp_lsb = get[0];
    // Temp data plus lsb

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้


```

        if (temp_msb <= 0x80) temp_lsb = (temp_lsb/2);           // shift to get
whole degree
        temp_msb = temp_msb & 0x80;                             //
mask all but the sign bit
        if (temp_msb >= 0x80) temp_lsb = (~temp_lsb)+1; // twos complement
        if (temp_msb >= 0x80) temp_lsb = (temp_lsb/2);         // shift to get
whole degree
        if (temp_msb >= 0x80) temp_lsb = ((-1)*temp_lsb);       // add sign bit
        return(temp_lsb);
}

```

// READ_ROM - ROM by 1 device

```

/*void Read_ROM(void) {
    char dat[8];

```

```

    ow_reset();
    write_byte(0x33);
    for (i=0; i<8; i++) dat[i]=read_byte();
    printf("$");
    printf("%x",ff,dat[7]);
    printf("%x",ff,dat[6]);
    printf("%x",ff,dat[5]);
    printf("%x",ff,dat[4]);
    printf("%x",ff,dat[3]);
    printf("%x",ff,dat[2]);
    printf("%x",ff,dat[1]);
    printf("%x",ff,dat[0]);
    printf("@n");

```

*/

```

void fan_1() //Front fan
{

```

```

    P2_0 = 1;
    P2_1 = 0;
    P2_2 = 1;
    P2_3 = 0;
    P2_4 = 1;
    P2_5 = 0;

```

}

```

void fan_2() //Back fan
{

```

```

    P2_0 = 0;
    P2_1 = 0;
    P2_2 = 1;
    P2_3 = 0;
    P2_4 = 0;
    P2_5 = 1;

```

}

```

void fan_s() //stop fan
{

```

```

    P2_0 = 1;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้


```

    P2_1 = 0;
    P2_2 = 1;
    P2_3 = 0;
    P2_4 = 0;
    P2_5 = 0;
}
void init_serial(void) {
    TMOD |= 0x20;
    TCON = 0x40;
    SCON = 0x50;
    PCON &= 0x7F;
    TH1 = 0xFD;
    //TL1 = 0xFD;
    TR1 = 1;
    TI = 1;
}
/*void initialize_serial()
{
// Initialize the serial port (9600, 8, N, 1)
    PCON &= 0x7F; // Clear bit
    7 of the PCON register (SMOD1 = 0)
    SCON = 0x50; //
    0101,0000 (Mode 1 and RxD enable)
    TMOD |= 0x20; // Timer #1
    in autoreload 8 bit mode
    TCON = 0x40; // Set
    Timer #1 to run mode
    TH1 = 0xFD; // FA = 4800 FD=9600 // Baud rate is determined
    by
    Timer #1 overflow rate //
    Baud Rate = (Fcpu / 384) / (256 - TH1) //
    Fcpu = 11.0592 MHz //
    TH1 = 253 //
    TR1 = 1; // Turn on
    Timer 1
}*/

void Dimmer() interrupt 1 // timer 0 interrupt TF0
{
    if (start==1 && TmpLev>0)
    {
        start=0;
        CH=0;//turn off light
        Remain=0xFFFF-(MapTable[TmpLev]-PhaseLen);
        TH0 = ((unsigned short)Remain >> 8);//load remain timer
        TL0 = Remain;
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้


```

        TR0 = 1;//enable timer
    }
    else
    {
        CH=1;//turn on light
        TmpLev=CHLevel;
    }
}
void synchronize() interrupt 2    //external interrupt
{
    start=1;
    TH0 = ((unsigned short)MapTable[TmpLev] >> 8);//load timer
    TL0 = MapTable[TmpLev];
    TR0 = 1;//Enable timer 0*/
}
void GetPhaseLen(void)
{
    // Init Timer 0
    TMOD |= 0x01;    // Timer 0, Mode 1, 16-bit timer
    TR0 = 0;    // Disable timer 0
    TH0 = 0;    // Reload timer
    TL0 = 0;

    // Wait for falling edge
    while (!sync) {}
    while (sync) {}
    TR0 = 1;    // Enable timer 0

    // Wait for rising edge
    while (!sync) {}
    TR0 = 0;    // Disable timer 0
}

unsigned short GetAvgPhaseLen(void)
{
    unsigned short avg;
    //unsigned char i;

    GetPhaseLen();
    avg = ((unsigned short)TH0 << 8) | (unsigned short)TL0;    // TH0 or TL0
    for (i=0;i<19;i++) // Use 20 samples
    {
        GetPhaseLen();
        avg = (avg + (((unsigned short)TH0 << 8) | (unsigned short)TL0)) / 2;
    }
    return avg;
}

```

void main(void)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้


```

{
    //unsigned char start;
    int temp1,temp2,err1,err2,c_err;
    char secode1[8],secode2[8];
    secode1[0] = 0x10;
    secode1[1] = 0xA4;
    secode1[2] = 0x6D;
    secode1[3] = 0x92;
    secode1[4] = 0x00;
    secode1[5] = 0x08;
    secode1[6] = 0x00;
    secode1[7] = 0x16;
    //16 00 08 00 92 6D A4 10
    secode2[0] = 0x10;
    secode2[1] = 0x24;
    secode2[2] = 0x99;
    secode2[3] = 0x92;
    secode2[4] = 0x00;
    secode2[5] = 0x08;
    secode2[6] = 0x00;
    secode2[7] = 0x45;
    // 45 00 08 00 92 99 24 10

    // Find phase length
    PhaseLen = GetAvgPhaseLen();
    StepLen = PhaseLen / (MAX_LEVEL+1);
    PhaseLen = 0xFFFF-PhaseLen;
    for (i=0; i<MAX_LEVEL; i++) { // Create map table
        MapTable[i] = 0xFFFF-((MAX_LEVEL-i) * StepLen);
    }
    MapTable[MAX_LEVEL]=MapTable[MAX_LEVEL-1]+(StepLen/2);
    init_serial();
    /*printf("PhaseLen %x \n",(int)PhaseLen);
    printf("MapTable[0] %x \n",(int)MapTable[0]);
    printf("MapTable[1] %x \n",(int)MapTable[1]);
    printf("MapTable[2] %x \n",(int)MapTable[2]);
    printf("MapTable[3] %x \n",(int)MapTable[3]);
    printf("MapTable[4] %x \n",(int)MapTable[4]);
    printf("MapTable[5] %x \n",(int)MapTable[5]);
    printf("MapTable[6] %x \n",(int)MapTable[6]);
    printf("MapTable[7] %x \n",(int)MapTable[7]);*/

    while(1){
        //init_serial();
        TMOD |= 0x01; // Timer 0, Mode 1, 16-bit timer
        TR0 = 0; // Disable timer 0
        IT1 = 0x01; //External interrupt1 active falling edge
        IE = 0x96; //10010110//enable global,External interrupt1 and interrupt

        Timer0

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้


```

printf("enter setpoint \n");
TI =0;
loop = 0;
while(get_data(set_p));
//printf("Setpoint = ",co);
EA = 0;
temp1 = Read_Temp(secode1);
EA = 1;
EA = 0;
temp2 = Read_Temp(secode2);
EA = 0;

```

```

err2 = set_p[0] - temp2;
for (i=0;i<100;i++)
    for (j=0;j<1000;j++);

while(1) {
    //ES = 0;
    EA = 0;
    CH=1;
    temp1 = Read_Temp(secode1);
    EA =1;
    //TI = 1;
    printf("Temp1 %d degrees C \n",(int)temp1);
    TI =0;
    EA =0;
    CH=1;
    temp2 = Read_Temp(secode2);
    EA =1;
    printf("Temp2 %d degrees C \n",(int)temp2);
    TI =0;
    err1 = err2;
    err2 = set_p[0] - temp2;
    c_err = err2 -err1;
    //ES = 1;
    //if (!RI)
    //{

```

C\n",(int)set_p[0]);

```
printf("Setpoint = %d degrees
```

```

TI =0;
//////////
if ((err2 < -4) && (c_err == -2))
{
    CHLevel = 0 ;
    printf("CHLevel 0 \n");
    TI =0;
}
if ((err2 < -4) && (c_err == -1))
{

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้


```

CHLevel = 0 ;
printf("CHLevel 0 \n");
TI =0;
}
if ((err2 < -4) && (c_err == 0))
{
CHLevel = 0 ;
printf("CHLevel 0 \n");
TI =0;
}
if ((err2 < -4) && (c_err == 1))
{
CHLevel = 0 ;
printf("CHLevel 0 \n");
TI =0;
}
if ((err2 < -4) && (c_err == 2))
{
CHLevel = 0 ;
printf("CHLevel 0 \n");
TI =0;
}
////////////////////
if ((err2 == -4) && (c_err == -2))
{
CHLevel = 0 ;
printf("CHLevel 0 \n");
TI =0;
}
if ((err2 == -4) && (c_err == -1))
{
CHLevel = 0 ;
printf("CHLevel 0 \n");
TI =0;
}
if ((err2 == -4) && (c_err == 0))
{
CHLevel = 0 ;
printf("CHLevel 0 \n");
TI =0;
}
if ((err2 == -4) && (c_err == 1))
{
CHLevel = 0 ;
printf("CHLevel 0 \n");
TI =0;
}
if ((err2 == -4) && (c_err == 2))
{
CHLevel = 1 ;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้


```

        printf("CHLevel 1 \n");
        TI =0;
    }
    ///////////////////////////////////
    if ((err2 == -3) && (c_err == -2))
    {
        CHLevel = 0 ;
        printf("CHLevel 0 \n");
        TI =0;
    }
    if ((err2 == -3) && (c_err == -1))
    {
        CHLevel = 0 ;
        printf("CHLevel 0 \n");
        TI =0;
    }
    if ((err2 == -3) && (c_err == 0))
    {
        CHLevel = 0 ;
        printf("CHLevel 0 \n");
        TI =0;
    }
    if ((err2 == -3) && (c_err == 1))
    {
        CHLevel = 0 ;
        printf("CHLevel 0 \n");
        TI =0;
    }
    if ((err2 == -3) && (c_err == 2))
    {
        CHLevel = 1 ;
        printf("CHLevel 1 \n");
        TI =0;
    }
    ///////////////////////////////////
    if ((err2 == -2) && (c_err == -2))
    {
        CHLevel = 0 ;
        printf("CHLevel 0 \n");
        TI =0;
    }
    if ((err2 == -2) && (c_err == -1))
    {
        CHLevel = 1 ;
        printf("CHLevel 1 \n");
        TI =0;
    }
    if ((err2 == -2) && (c_err == 0))
    {
        CHLevel = 1 ;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้


```

        printf("CHLevel 1 \n");
        TI =0;
    }
    if ((err2 == -2) && (c_err == 1))
    {
        CHLevel = 1 ;
        printf("CHLevel 1 \n");
        TI =0;
    }
    if ((err2 == -2) && (c_err == 2))
    {
        CHLevel = 2 ;
        printf("CHLevel 2 \n");
        TI =0;
    }
    ///////////////////////////////////
    if ((err2 == -1) && (c_err == -2))
    {
        CHLevel = 1 ;
        printf("CHLevel 1 \n");
        TI =0;
    }
    if ((err2 == -1) && (c_err == -1))
    {
        CHLevel = 2 ;
        printf("CHLevel 2 \n");
        TI =0;
    }
    if ((err2 == -1) && (c_err == 0))
    {
        CHLevel = 2 ;
        printf("CHLevel 2 \n");
        TI =0;
    }
    if ((err2 == -1) && (c_err == 1))
    {
        CHLevel = 2 ;
        printf("CHLevel 2 \n");
        TI =0;
    }
    if ((err2 == -1) && (c_err == 2))
    {
        CHLevel = 3 ;
        printf("CHLevel 3 \n");
        TI =0;
    }
    ///////////////////////////////////
    if ((err2 == 0) && (c_err == -2))
    {
        CHLevel = 2 ;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้


```

        printf("CHLevel 2 \n");
        TI =0;
    }
    if ((err2 == 0) && (c_err == -1))
    {
        CHLevel = 3 ;
        printf("CHLevel 3 \n");
        TI =0;
    }
    if ((err2 == 0) && (c_err == 0))
    {
        CHLevel = 3 ;
        printf("CHLevel 3 \n");
        TI =0;
    }
    if ((err2 == 0) && (c_err == 1))
    {
        CHLevel = 3 ;
        printf("CHLevel 3 \n");
        TI =0;
    }
    if ((err2 == 0) && (c_err == 2))
    {
        CHLevel = 4 ;
        printf("CHLevel 4 \n");
        TI =0;
    }
    ///////////////////////////////////
    if ((err2 == 1) && (c_err == -2))
    {
        CHLevel = 3 ;
        printf("CHLevel 3 \n");
        TI =0;
    }
    if ((err2 == 1) && (c_err == -1))
    {
        CHLevel = 4 ;
        printf("CHLevel 4 \n");
        TI =0;
    }
    if ((err2 == 1) && (c_err == 0))
    {
        CHLevel = 4 ;
        printf("CHLevel 4 \n");
        TI =0;
    }
    if ((err2 == 1) && (c_err == 1))
    {
        CHLevel = 4 ;
        printf("CHLevel 4 \n");
    }

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ในการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้


```

        TI =0;
    }
    if ((err2 == 1) && (c_err == 2))
    {
        CHLevel = 5 ;
        printf("CHLevel 5 \n");
        TI =0;
    }
    //////////////////////////////////
    if ((err2 == 2) && (c_err == -2))
    {
        CHLevel = 4 ;
        printf("CHLevel 4 \n");
        TI =0;
    }
    if ((err2 == 2) && (c_err == -1))
    {
        CHLevel = 5 ;
        printf("CHLevel 5 \n");
        TI =0;
    }
    if ((err2 == 2) && (c_err == 0))
    {
        CHLevel = 5 ;
        printf("CHLevel 5 \n");
        TI =0;
    }
    if ((err2 == 2) && (c_err == 1))
    {
        CHLevel = 5 ;
        printf("CHLevel 5 \n");
        TI =0;
    }
    if ((err2 == 2) && (c_err == 2))
    {
        CHLevel = 6 ;
        printf("CHLevel 6 \n");
        TI =0;
    }
    //////////////////////////////////
    if ((err2 == 3) && (c_err == -2))
    {
        CHLevel = 5 ;
        printf("CHLevel 5 \n");
        TI =0;
    }
    if ((err2 == 3) && (c_err == -1))
    {
        CHLevel = 6 ;
        printf("CHLevel 6 \n");
    }

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้拿去ใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้


```

        TI =0;
    }
    if ((err2 == 3) && (c_err == 0))
    {
        CHLevel = 6;
        printf("CHLevel 6 \n");
        TI =0;
    }
    if ((err2 == 3) && (c_err == 1))
    {
        CHLevel = 6 ;
        printf("CHLevel 6 \n");
        TI =0;
    }
    if ((err2 == 3) && (c_err == 2))
    {
        CHLevel = 7 ;
        printf("CHLevel 7 \n");
        TI =0;
    }
    ///////////////////////////////////
    if ((err2 == 4) && (c_err == -2))
    {
        CHLevel = 6 ;
        printf("CHLevel 6 \n");
        TI =0;
    }
    if ((err2 == 4) && (c_err == -1))
    {
        CHLevel = 6 ;
        printf("CHLevel 6 \n");
        TI =0;
    }
    if ((err2 == 4) && (c_err == 0))
    {
        CHLevel = 7 ;
        printf("CHLevel 7 \n");
        TI =0;
    }
    if ((err2 == 4) && (c_err == 1))
    {
        CHLevel = 7 ;
        printf("CHLevel 7 \n");
        TI =0;
    }
    if ((err2 == 4) && (c_err == 2))
    {
        CHLevel = 7 ;
        printf("CHLevel 7 \n");
        TI =0;
    }

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้


```

    }
    //////////////////////////////////
    if ((err2 > 4) && (c_err == -2))
    {
        CHLevel = 6 ;
        printf("CHLevel 6 \n");
        TI =0;
    }
    if ((err2 > 4) && (c_err == -1))
    {
        CHLevel = 7 ;
        printf("CHLevel 7 \n");
        TI =0;
    }
    if ((err2 > 4) && (c_err == 0))
    {
        CHLevel = 7 ;
        printf("CHLevel 7 \n");
        TI =0;
    }
    if ((err2 > 4) && (c_err == 1))
    {
        CHLevel = 7 ;
        printf("CHLevel 7 \n");
        TI =0;
    }
    if ((err2 > 4) && (c_err == 2))
    {
        CHLevel = 7 ;
        printf("CHLevel 7 \n");
        TI =0;
    }
    //////////////////////////////////
    if ((temp1 - temp2 ) > (1))
    {
        fan_2();
        printf("Fan2 \n\n");
        TI =0;
    }
    if ((temp2 - temp1 ) > (1))
    {
        fan_1();
        printf("Fan1 \n\n");
        TI =0;
    }
    if (abs(temp1 - temp2 ) <= (1))
    {
        fan_s();
        printf("Stop \n\n");
        TI =0;
    }

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้


```

    }
    if (((temp1-set_p[0]) > (1)) || ((temp2-
set_p[0]) > (1)))
    {
        fan_1();
        fan_2();
        printf("Fan1 and Fan2 \n\n");
        TI =0;
    }
    //Read_ROM();
    for (i=0;i<200;i++)

        for (j=0;j<1000;j++);

    //}

    if (RI && (SBUF == 'r'))
    {
        break;
    }
    else
        RI = 0;
}
}
}

```

