



การควบคุมอุณหภูมิในเตาอบไฟฟ้าโดยใช้ฟัซซี่ลอจิก
FUZZY LOGIC TEMPERATURE CONTROLLER FOR OVEN

นางสาวฐิติรัตน์ พัฒนศรีศักดิ์ รหัส 43362425
นายบัณฑิต ศรีคง รหัส 43362532
นายสุทธิชาติ สุชัยมา รหัส 43362748

ห้องสมุดคณะวิศวกรรมศาสตร์
วันที่รับ..... 25 / พ.ค. 2553 /
เลขทะเบียน..... 5021285
เลขเรียกหนังสือ..... ๒/๖
มหาวิทยาลัยนเรศวร ๖367 ๓

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต²⁵⁴⁶
สาขาวิชาวิศวกรรมไฟฟ้า ภาควิชาวิศวกรรมไฟฟ้าและคอมพิวเตอร์
คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร
ปีการศึกษา 2546

หัวข้อโครงการ	การควบคุมอุณหภูมิในเตาอบไฟฟ้าโดยใช้ฟuzzyลอจิก
ผู้ดำเนินโครงการ	นางสาววิจิรัตน์ พัฒนศรีศักดา รหัส 43362425 นายบัณฑิต ศรีคง รหัส 43362532 นายสุทธิชาติ สุรัมย์มา รหัส 43362748
อาจารย์ที่ปรึกษา	อาจารย์พรพิศุทธิ์ วรจิรันตัน
อาจารย์ที่ปรึกษาร่วม	ดร.ขงยุทธ ชนบดีเฉลิมรุ่ง
สาขาวิชา	วิศวกรรมไฟฟ้า
ภาควิชา	วิศวกรรมไฟฟ้าและคอมพิวเตอร์
ปีการศึกษา	2546

บทคัดย่อ

โครงการนี้เป็นการศึกษาและนำทฤษฎี Fuzzy Logic มาใช้ควบคุมอุณหภูมิในเตาอบไฟฟ้าจำลองเพื่อเป็นแนวทางในการควบคุมอุณหภูมิในอุตสาหกรรมขั้นสูงต่อไป

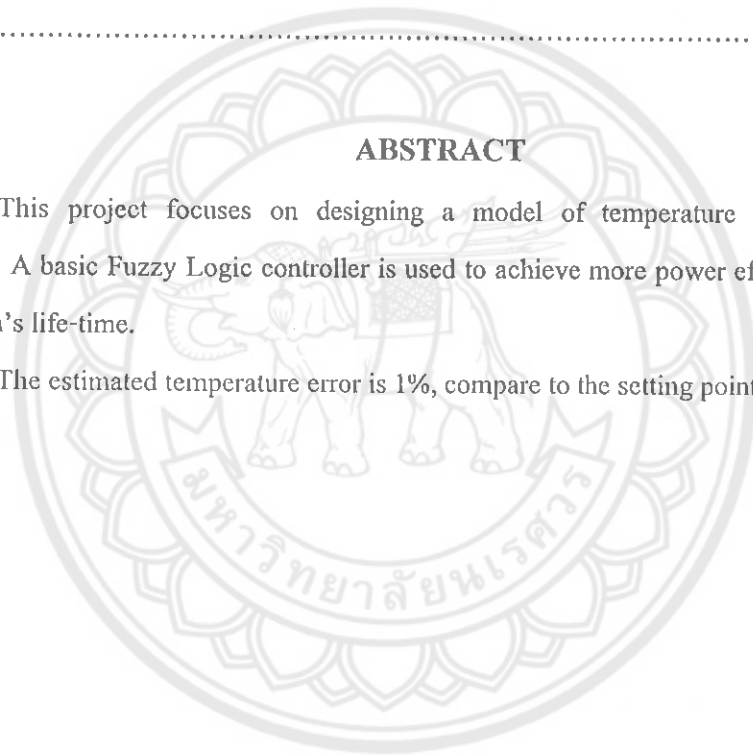
ผลที่ได้รับจากการทำโครงการนี้คือสามารถควบคุมอุณหภูมิในเตาอบได้ตามค่าอุณหภูมิที่ตั้งเอาไว้เมื่อเทียบกับอุณหภูมิที่วัดได้จริงจากเทอร์โมคัปเปิลซึ่งอาจจะมีคลาดเคลื่อนประมาณ 1%

Project Title Fuzzy Logic Temperature Controller For Oven
Name Miss Titirut Pattanasrisukda ID 43362425
Mr. Bunbit Srikong ID 43362532
Mr. Sootthichat Soothamma ID 43362478
Project Advisor Miss Pornpisoot Worajiran
CO-Project Advisor Dr.Yongyoot Chanabodychaloemrung
Major Electrical Engineering
Department Electrical and Computer Engineering
Academic Year 2003

ABSTRACT

This project focuses on designing a model of temperature controlled oven for industries. A basic Fuzzy Logic controller is used to achieve more power efficiency and prolong the system's life-time.

The estimated temperature error is 1%, compare to the setting point.



กิตติกรรมประกาศ

โครงการวิศวกรรมเรื่องการควบคุมอุณหภูมิในเตาอบไฟฟ้าโดยใช้ฟิวส์ล่อจิก ที่ทำสำเร็จเป็นรูปเล่มได้เนื่องจากได้รับความกรุณาจาก อาจารย์ พรพิศุทธิ์ วรจิรันตน์ ภาควิชาวิศวกรรมไฟฟ้า และคอมพิวเตอร์ มหาวิทยาลัยนเรศวร ผู้เป็นอาจารย์ที่ปรึกษา ได้วางรากฐานและประสิทธิ์วิชาความรู้ทางด้านนี้ ทั้งยังให้ความช่วยเหลือเป็นอย่างดีให้แก่ผู้ร่วมงาน อีกสองท่านคือ อาจารย์ยงยุทธ ชนบดีเฉลิมรุ่ง และอาจารย์สุชาติ แย้มเม่น ภาควิชาไฟฟ้าและคอมพิวเตอร์ที่กรุณาให้คำชี้แนะต่าง ๆ และ สุดท้ายนี้ที่จะขาดไม่ได้คือภาควิชาอุตสาหกรรมเกษตร คณะเกษตรที่ได้เอื้อเพื่อให้ใช้เตาอบ

ขอแสดงความขอบคุณผู้ที่ช่วยให้โครงการนี้สำเร็จได้ด้วยดี

อาจารย์ พรพิศุทธิ์ วรจิรันตน์ ที่ปรึกษาโครงการ

นางสาวฐิติรัตน์ พัฒนศรีศักดิ์

นายบัณฑิต ศรีคง

นายสุทธิชาติ สุรัมย์มา



สารบัญ

	หน้า
บทคัดย่อภาษาไทย.....	ก
บทคัดย่อภาษาอังกฤษ.....	ข
กิตติกรรมประกาศ.....	ค
สารบัญ.....	ง
สารบัญตาราง.....	ฉ
สารบัญรูป.....	ช
บทที่ 1 บทนำ	
1.1 ที่มาและความสำคัญของ โครงการ.....	1
1.2 วัตถุประสงค์ของโครงการ.....	1
1.3 ขอบข่ายของงาน.....	1
1.4 กิจกรรมการดำเนินการ.....	2
1.5 ผลที่คาดว่าจะได้รับ.....	2
1.6 งบประมาณ.....	2
บทที่ 2 ความรู้และทฤษฎีพื้นฐาน	
2.1 Thermocoupleและการใช้งาน.....	3
2.2 ความรู้พื้นฐานเกี่ยวกับฮอปแอมป์.....	7
2.3 ไอซี A/D converter.....	12
2.4 เอสซีอาร์.....	15
2.5 ทฤษฎี Fuzzy Logic.....	19
บทที่ 3 การออกแบบการควบคุมอุณหภูมิในเตาอบไฟฟ้าโดยใช้ฟuzzyลอจิก	
3.1 ภาพรวมของระบบการควบคุมอุณหภูมิในเตาอบไฟฟ้าโดยใช้ฟuzzyลอจิก.....	30
3.2 อุปกรณ์ที่ใช้.....	30
3.3 แนวคิดและขั้นตอนการออกแบบ.....	30

สารบัญ(ต่อ)

	หน้า
บทที่ 4 ผลการออกแบบระบบการควบคุมอุณหภูมิของเตาอบ โดยใช้ Fuzzy logic ผลการดำเนิน โครงการ.....	47
บทที่ 5 สรุปผลการปฏิบัติโครงการ	
5.1 สรุปผลการปฏิบัติงาน	49
5.2 ประโยชน์ที่ได้รับจากการทำงาน.....	49
5.3 แนวทางพัฒนาระบบต่อไปในอนาคต.....	49
เอกสารอ้างอิง.....	50
ภาคผนวก.....	51
ประวัติผู้เขียน โครงการ.....	72



สารบัญตาราง

ตารางที่	หน้า
2.1 เทอร์โมคัปเปิลชนิดต่างๆ ที่นิยมใช้กัน	4
2.2 ขาและคำอธิบายของ ADC 0804	12
3.1 แสดงตัวอย่างกฎ.....	36
3.2 การเปรียบเทียบระหว่างค่า $e(t)$ และ $y(t)$	37
3.3 แสดงค่า Output	38



สารบัญรูป

รูปที่	หน้า
2.1 วงจรพื้นฐานของเทอร์โมคัปเปิล.....	3
2.2 ความสัมพันธ์ระหว่างค่าอุณหภูมิและซีเบคที่เกิดขึ้นไม่เป็นเชิงเส้นแต่เป็นแบบพาราโบลา....	4
2.3 โครงสร้างพื้นฐานของเทอร์โมคัปเปิลแบบง่ายที่เหมาะสมสำหรับการวัดอุณหภูมิในเตาอบไฟฟ้า....	5
2.4 วิธีที่นิยมใช้ในการป้องกันเทอร์โมคัปเปิลในสภาพแวดล้อมที่มีการก่กร่อนสูงหรือเกี่ยวข้องกับปฏิกิริยาเคมี.....	5
2.5 โครงสร้างที่มีแร่เป็นฉนวนและเปลือกหุ้มเป็น โลหะหรือ MIMS ของเทอร์โมคัปเปิลซึ่งสามารถดึงลดขนาดจนมีเส้นผ่านศูนย์กลางเล็กถึง 0.5 มิลลิเมตร	6
2.6 แสดงรอยต่อของเทอร์โมคัปเปิล	7
2.7 แสดงอัตราขยายและวงจรสมมูลของออปแอมป์.....	7
2.8 แรงดันออปเซทเปลี่ยนจาก 1 mv เป็น 10 mv ที่เอาต์พุท เมื่อออปแอมป์มีอัตราขยาย 10 เท่า...8	8
2.9 การกำจัดแรงดันออปเซท โดยการให้ขยายเฉพาะแรงดันที่เป็น ไฟสลับ.....	9
2.10 แสดงวิธีการปรับออปเซท	10
2.11 แสดงการใช้อนาล็อกสวิตช์เพื่อเปิดปิดวงจรทางอินพุท.....	11
2.12 แสดงขาต่าง ๆ ของ ADC 0804.....	12
2.13 แสดงการต่อใช้งานของ ADC 0804	14
2.14 แสดงสัญลักษณ์และ โครงสร้างพื้นฐานของ เอสซีอาร์.....	15
2.15 แสดงการทำงานของเอสซีอาร์.....	16
2.16 แสดงการปรับมุมทริกเกอร์.....	17
2.17 ขั้นตอนการออกแบบ Fuzzy logic controller.....	20
2.18 block diagram ของระบบตัวอย่าง.....	21
2.19 ลักษณะทั่วไปของ Fuzzy set.....	22
2.20a Triangular Membership Function.....	23
2.20b Trapezoidal Membership Function.....	23
2.20c Gaussian Membership Function.....	23
2.20d Generalized bell Membership Function.....	24
2.20e Sigmoidal Membership Function.....	24
2.20f Left-right Membership Function.....	24

สารบัญรูป(ต่อ)

รูปที่	หน้า
2.21 Fuzzy set จากระบบตัวอย่าง และ input ที่วัดได้.....	25
2.22 ตัวอย่าง Rules Structure และ Rules matrix.....	27
2.23 Output.....	29
3.1 วงจร KEYPAD, LCD และเทอร์โมคัปเปิล.....	30
3.2 วงจร SCR	31
3.3 ภาพเตาอบ	31
3.4 ระบบควบคุมในเตาอบ.....	32
3.5 รูปของ $e(t)$	34
3.6 รูปของ $y(t)$	34
3.7 รูปของ Output sub-domains	35
3.8 Block Diagram ของระบบ.....	45
3.9 รูปวงจรรวมของระบบ	46
4.1 กราฟการที่ไม่มีการ delay	47
4.2 รูปการ delay ที่ 0.5s	48
4.3 รูปการ delay ที่ 1s	48

บทที่ 1

บทนำ

1.1 ที่มาและความสำคัญของโครงการ

เนื่องจากในปัจจุบันมีอุตสาหกรรมที่ต้องใช้เตาอบเพื่อควบคุมอุณหภูมิอยู่มาก เช่น อุตสาหกรรมเซรามิก อุตสาหกรรมทางการเกษตร ยังมีข้อผิดพลาดในเรื่องของอุณหภูมิที่ต้องการ หรือระยะเวลาที่ใช้ในการเพิ่มหรือลดอุณหภูมิ จึงทำให้เกิดแนวความคิดที่จะใช้ทฤษฎีของ Fuzzy Logic มาควบคุมแทนเทอร์โมสตัทที่ปิดเปิดวงจรอย่างทันทีทันใดซึ่งอาจจะส่งผลเสียต่ออุปกรณ์ภายในได้ และในระบบควบคุมที่มีสมการทางคณิตศาสตร์ที่ยุ่งยากหรือบางครั้งไม่สามารถหาสมการได้ เราสามารถใช้ทฤษฎีของ Fuzzy Logic มาควบคุมแทน ซึ่งมีลักษณะเด่น คือ

- ไม่ใช่แบบจำลองทางคณิตศาสตร์ในการควบคุม
- สามารถพัฒนาให้การควบคุมคล้ายกับพฤติกรรมของมนุษย์ได้
- สามารถเข้าถึงปัญหาระบบได้เร็ว จึงสามารถทำการควบคุมได้อย่างทันเวลา

1.2 วัตถุประสงค์ของโครงการ

- 1.2.1 ศึกษาการทำงานของ Fuzzy Logic Controller
- 1.2.2 ศึกษาการออกแบบ Fuzzy Logic Controller ทางด้านการเขียนโปรแกรม
- 1.2.3 เปรียบเทียบข้อดีและข้อเสียกับเตาอบไฟฟ้าทั่วไป
- 1.2.4 เพื่อเป็นแบบจำลองของการออกแบบเตาอบไฟฟ้าในงานอุตสาหกรรม
- 1.2.5 ศึกษาเกี่ยวกับการนำเทอร์โมคัปเปิลมาใช้ในการวัดอุณหภูมิ

1.3 ขอบข่ายของงาน

ใช้ Thermocouple สร้างแบบจำลองระบบควบคุมอุณหภูมิของเตาอบในอุตสาหกรรม โดยใช้เตาอบขนาดเล็กและนำข้อมูลที่วัดได้มาแปลง AD เพื่อให้เป็นไปตามเงื่อนไขเกี่ยวกับระบบ Fuzzy จากนั้นข้อมูลจะถูกส่งไปยังกฎพื้นฐานของ Fuzzy แล้วนำผลที่ได้มาแปลง DA เพื่อไปควบคุมอุณหภูมิของเตาอบเพื่อให้ได้อุณหภูมิตามที่ตั้งไว้

1.4 กิจกรรมการดำเนินการ

กิจกรรม	ปี 2545		ปี 2546									
	พ. ย.	ธ. ค.	ม. ค.	ก. พ.	มี. ค.	เม. ย.	พ. ค.	มิ. ย.	ก. ค.	ส. ค.	ก. ย.	ต. ค.
รวบรวมข้อมูลและ ศึกษาFuzzy logic												
จัดหาอุปกรณ์ และเริ่มลงมือทำชิ้นงาน												
ทดสอบ ปรับปรุงและ แก้ไข												
เตรียมนำเสนอชิ้นงาน และสรุปผลที่ได้รับ												

1.5 ผลที่คาดว่าจะได้รับ

- 1.5.1 เข้าใจการทำงานของ Fuzzy Logic Controller
- 1.5.2 สามารถออกแบบ Fuzzy Logic Controller ได้
- 1.5.3 สามารถนำไปประยุกต์ใช้กับการควบคุมอุณหภูมิของเตาอบในโรงงานอุตสาหกรรมได้
- 1.5.4 สามารถสร้างวงจรวัดอุณหภูมิโดยใช้เทอร์โมคัปเปิลได้

1.6 งบประมาณ

- | | |
|----------------------|-----------|
| 1.6.1 เตาไฟฟ้า | 1,500 บาท |
| 1.6.2 วงจรไฟฟ้าต่างๆ | 1,000 บาท |
| 1.6.3 Thermocouple | 500 บาท |
| 1.6.4 เอกสาร | 1,000 บาท |

บทที่ 2

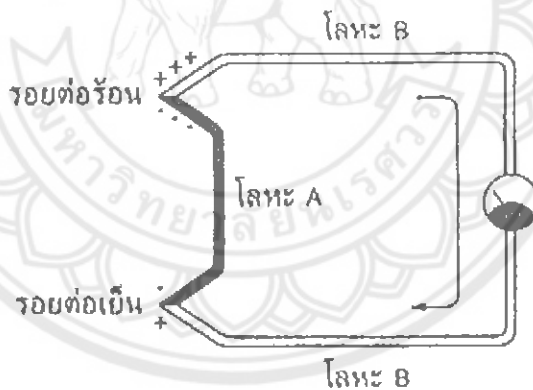
ความรู้และทฤษฎีพื้นฐาน

2.1 Thermocouple และการใช้งาน

เทอร์โมคัปเปิลเป็นอุปกรณ์ที่วัดอุณหภูมิที่สูง ซึ่งงานอุตสาหกรรมส่วนใหญ่มักต้องเกี่ยวกับอุณหภูมิสูงๆ เหล่านี้

2.1.1 หลักการทำงาน

เมื่อต่อโลหะ 2 ชนิดเข้าด้วยกัน ตามรูปที่ 2.1 ให้มีรอยต่อ ระหว่างโลหะ 2 ชนิดนี้ 2 แห่ง แล้วทำให้รอยต่อทั้งสองมีอุณหภูมิต่างกัน จะเกิดกระแสไฟฟ้าขนาดอ่อนๆ ไหลภายในวงจรการ ที่เกิดกระแสไฟฟ้าไหลได้นี้ เนื่องจากมีความแตกต่างศักย์เกิดขึ้นที่รอยต่อแต่ละแห่ง และมีขั้วตรงข้ามกัน. โดยที่แรงดันขั้ว รอยต่อร้อนจะสูงกว่า แรงดันที่รอยต่อเย็น ความต่างศักย์ที่ เกิดขึ้นบนรอยต่อของโลหะนี้เราเรียกว่า " แรงดันไฟฟ้าซี เบค " (Seebeck EMF) ผลต่างของแรงดันที่เกิดขึ้นนี้ทำให้เกิดเป็นแรงดันจำนวนหนึ่งที่ทำให้มีกระแสไหลก็ได้.



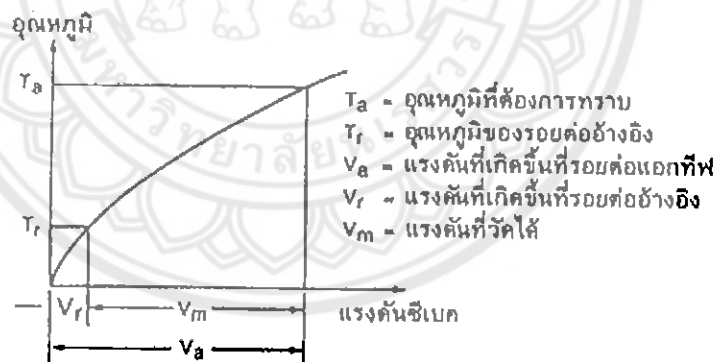
รูปที่ 2.1 วงจรพื้นฐานของเทอร์โมคัปเปิล

แรงดันที่เกิดขึ้นมีค่าน้อยมาก โดยปกติจะมีค่าประมาณ 0.2 - 0.3 มิลลิโวลต์เท่านั้น ซึ่งขึ้นอยู่กับชนิดของโลหะที่นำมา จับคู่ด้วย ตัวอักษรภาษาอังกฤษตารางที่ 1 แสดงถึงคุณสมบัติของเทอร์โมคัปเปิลบางชนิดที่นิยมใช้กัน

ตารางที่ 2.1 เทอร์โมคัปเปิลชนิดต่าง ๆ ที่นิยมใช้กัน

ชนิด	โลหะที่ใช้	ย่านอุณหภูมิ	ข้อดี-ข้อเสีย
S	โรเดียม (+) VS ททาลัม 10%	0-1400°C	มีเสถียรภาพสูง แหะระคาย
R	โรเดียม (+) VS ททาลัม 13%	0-1400°C	เช่นเดียวกับชนิด S
J	เหล็ก (+) VS ทองแดง-นิกเกิล (คอนสแตนตัน)	0-800°C	จะเกิดสนิมเหล็กขึ้น ถ้าไม่มีสารป้องกัน
K	นิกเกิล-โครเมียม (โครเมล) (+) VS นิกเกิล-ซิลิเนียม (ซิลอมล)	0-1100°C	เหมาะสำหรับบรรยากาศที่มีการออกซิไดซ์
T	ทองแดง (+) VS ทองแดง-นิกเกิล (คอนสแตนตัน)	-200 - 400°C	นิยมใช้กับอุณหภูมิต่ำกว่าอุณหภูมิห้อง จนถึงอุณหภูมิค่า
E	นิกเกิล-โครเมียม (โครเมล) (-) VS ทองแดง -นิกเกิล	0-800°C	เพียงตรงและมีเสถียรภาพ ตลอดจน ราคาถูก
N	นิกเกิล-โครเมียม-ซิลิกอน (นิโครซิล) (+) VS นิกเกิล-ซิลิกอน (นิกซิล)	0-1250°C	มีเสถียรภาพสูงที่ค่าอุณหภูมิสูงๆ

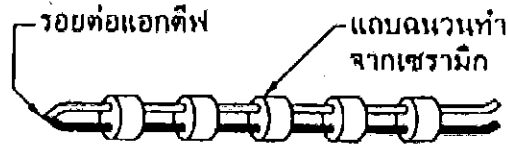
ความสัมพันธ์ระหว่างการเปลี่ยนแปลงอุณหภูมิที่รอยต่อของเทอร์โมคัปเปิล และแรงดันผลลัพธ์ที่ได้นั้นไม่ได้เปลี่ยนแปลงแบบเชิงเส้น แต่จะคล้ายกราฟพาราโบลา (parabolic curve) ดังแสดงในรูปที่ 2.2



รูปที่ 2.2 ความสัมพันธ์ระหว่างค่าอุณหภูมิ และแรงดันซีเบคที่เกิดขึ้นนั้น ไม่เป็นเชิงเส้น แต่เป็นแบบพาราโบลา

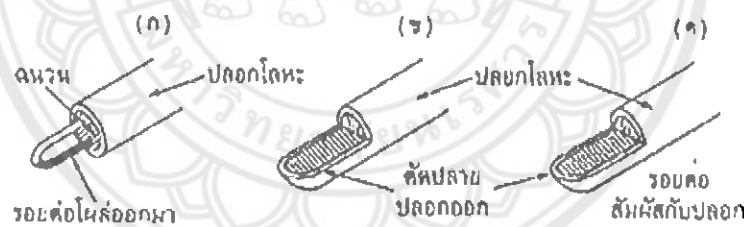
เทอร์โมคัปเปิลสำหรับอุณหภูมิสูงๆ ที่ใช้งานจริงนั้น มีรูปร่างต่างๆ กัน แบบที่ง่ายที่สุดนั้น อาจจะเป็นลวดเส้นเล็ก ๆ สอง เส้นร้อยผ่านฉนวนที่ทำจากเซรามิก แล้วเชื่อมปลายให้ติดกัน ดังรูปที่ 2.3 เทอร์โมคัปเปิลแบบนี้ใช้ในเตาเผาเครื่องเคลือบดินเผาขนาดเล็ก ๆ ได้ แต่สำหรับการวัดอุณหภูมิในสภาพแวดล้อมที่มีการกักความร้อนสูง หรือ เกี่ยวข้องกับปฏิกิริยาเคมี เช่น ไซของสาร เคมี

โลหะที่หลอมละลาย หรือ เปลวไฟแล้ว จำเป็นที่จะต้องหุ้มเทอร์โมคัปเปิล ด้วยปลอกโลหะ แล้วบรรจุสารที่เป็นฉนวนไฟฟ้า แต่นำความร้อนได้ดีไว้ระหว่างกลาง ให้เทอร์โมคัปเปิลติดตั้งอยู่แน่นอน หนา กับปลอกโลหะนั้น ดังแสดงในรูปที่ 2.4



รูปที่ 2.3 โครงสร้างพื้นฐานของเทอร์โมคัปเปิลแบบง่าย ๆ ที่เหมาะสำหรับการวัดอุณหภูมิในเตาอบไฟฟ้า

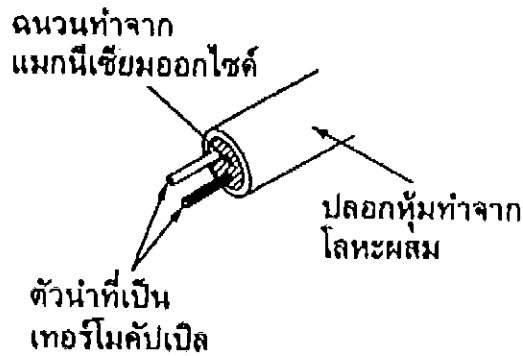
จะเห็นว่าวิธีติดตั้งอยู่ 3 แบบ คือ ในรูปที่ 2.4(ก) จะใส่ส่วนปลาย ที่เป็นรอยต่อระหว่างโลหะทั้งสองใส่ออกมาแบบนี้ จะให้ผลตอบสนองต่ออุณหภูมิ เร็วแต่ไม่สามารถทนการกัดกร่อนจากปฏิกิริยาเคมีได้



รูปที่ 2.4 วิธีที่นิยมกันในการป้องกันเทอร์โมคัปเปิล ในสภาพแวดล้อมที่มีการกัดกร่อนสูงหรือที่เกี่ยวข้องกับปฏิกิริยาเคมี

รูปที่ 2.4 (ก) ปล่อยให้รอยต่อใส่ออกมา (ข) และ (ค) บรรจุรอยต่ออยู่ในเปลือก รูปที่ 2.4 (ข) คือ หุ้มรอยต่อทั้งหมดไว้ในปลอก ซึ่งป้องกันการเสียหายได้ดี แต่การตอบสนองต่ออุณหภูมิ จะช้าลง เพราะปลอกหุ้ม และฉนวนจะนำความร้อนได้ช้า ซึ่งแก้ไขด้วยการติดตั้ง ในรูปที่ 4 (ค) คือ รอยต่อถูกยึดติดกับด้าน ในของปลอกหุ้มจึงได้ทั้งความไวต่อการเปลี่ยนแปลงอุณหภูมิ และป้องกันการเสียหายได้ดี.

โครงสร้างยอดนิยม สำหรับเทอร์โมคัปเปิลอุณหภูมิสูงทั้งหลายในปัจจุบัน



รูปที่ 2.5 โครงสร้างที่มีแร่เป็นฉนวนและเปลือกหุ้มเป็นโลหะหรือ MIMS ของเทอร์โมคัปเปิล ซึ่งสามารถดึงลดขนาดจนมีเส้นผ่านศูนย์กลางเล็กลงถึง 0.5 มม

2.1.2 การใช้งานเทอร์โมคัปเปิล

ในการวัดอุณหภูมิด้วยเทอร์โมคัปเปิลนั้น จะต้องควบคุมอุณหภูมิที่รอยต่อด้านหนึ่งไว้ในอุณหภูมิคงที่ ในขณะที่รอยต่อ อีกด้านหนึ่งอยู่ในอุณหภูมิที่ต้องการวัด ถ้าต้องการความเที่ยงตรงสูง จะต้องแช่รอยต่อด้านหนึ่งไว้ในกระติกน้ำแข็ง ที่ 0 องศาเซลเซียส แต่ถ้าไม่ต้องการความเที่ยงตรงมากนักก็ปล่อยให้รอยต่อด้านเย็นไว้ที่อุณหภูมิปกติก็ได้ จากนั้นทำการวัดแรงดันที่ได้จากเทอร์โมคัปเปิล แล้วนำมาหาค่าอุณหภูมิที่แท้จริงจากสมการ

$$V_a = V_m + V_r$$

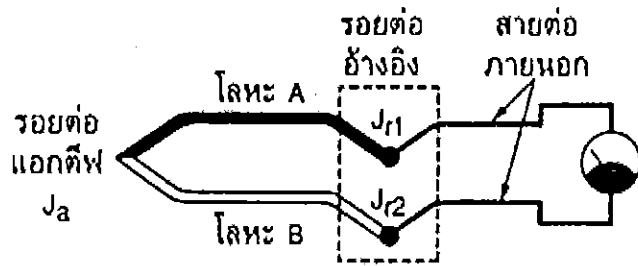
เมื่อ V_a คือ แรงดันซีเบคที่เกิดขึ้นที่รอยต่อแอลทีฟ (ในอุณหภูมิที่ต้องการทราบ)

V_m คือ แรงดันที่วัดได้

V_r คือ แรงดันซีเบคที่เกิดขึ้นที่รอยต่ออ้างอิง. (ในอุณหภูมิกงที่)

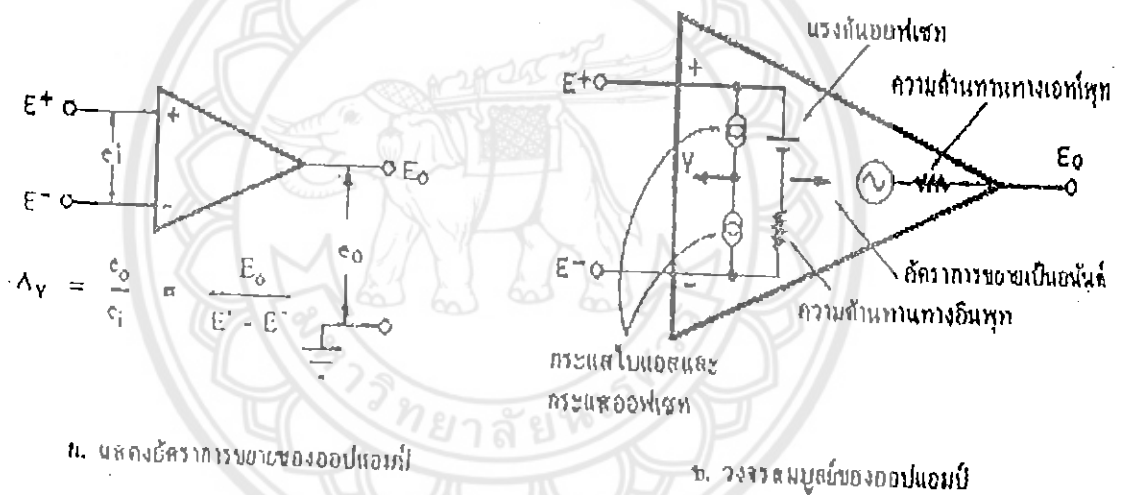
การหาค่าแรงดัน V_r จากตารางความสัมพันธ์ระหว่างแรงดันและอุณหภูมิ เมื่อทราบค่าอุณหภูมิอ้างอิง (T_r) แล้วนำไปบวกกับแรงดันที่วัดได้ ได้ผลลัพธ์เป็นแรงดัน V_a แล้วค่อยหาอุณหภูมิที่ ต้องการทราบ (T_a) จากตารางเดิมอีกครั้งหนึ่ง.

หากพิจารณาจริง ๆ แล้วจุดเชื่อมต่อสองจุดระหว่างเทอร์โมคัปเปิลกับสายต่อภายนอกที่ต่อไปยังโวลต์มิเตอร์ ก็มีคุณสมบัติเป็นเทอร์โมคัปเปิลด้วยเหมือนกัน คือ ส่วนที่อยู่ในเส้นประในรูปที่ 2.6 จึงควรให้สายต่อภายนอกทั้งสองเส้นนี้เป็นโลหะ ชนิดเดียวกัน และให้อุณหภูมิที่รอยต่อทั้งสองเท่ากัน, ก็จะไม่เกิดผลกระทบกับความถูกต้องของการวัดอุณหภูมิ รูปที่ 2.6 ในทางปฏิบัติ รอยต่อครั้งหนึ่งสองจุด คือ J_{r1} และ J_{r2}



รูปที่ 2.6 แสดงรอยต่อของเทอร์โมคัปเปิล

2.2 ความรู้พื้นฐานเกี่ยวกับออปแอมป์



รูปที่ 2.7 แสดงอัตราขยายและวงจรสมมูลของออปแอมป์

ออปแอมป์มีสัญลักษณ์และสัญญาณเข้าออก ตามที่แสดงในรูปที่ 2.7 ความสัมพันธ์ของสัญญาณเข้าและออก แสดงด้วยสูตรง่าย ๆ คือ

$$E_o = A_v(E^+ - E^-)$$

A_v เป็นอัตราขยายแรงดันซึ่งมีค่าสูงมาก จึงเป็นคุณสมบัติที่พิเศษสุดของออปแอมป์ ออปแอมป์แบบอุดมคติ จะมีคุณสมบัติตามเงื่อนไขดังนี้

1. อัตราขยายแรงดัน สูงมากจนเป็นอนันต์
2. ความต้านทานทางอินพุต สูงมากจนเป็นอนันต์
3. ความต้านทานทางเอาต์พุต ต่ำมากจนเป็นศูนย์

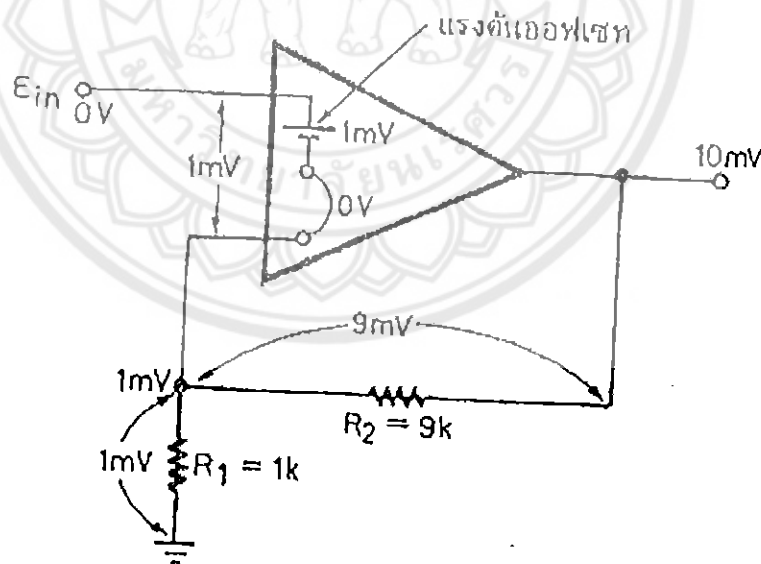
4. แรงดันออฟเซตทางอินพุตเป็นศูนย์
5. กระแสออฟเซตทางอินพุตเป็นศูนย์
6. ลักษณะสมบัติเชิงความถี่ ขยายได้ดีตั้งแต่ไฟตรงจนความถี่สูงมากเป็นอนันต์
7. ไม่มีข้อบกพร่องอื่น ๆ

แต่ในความเป็นจริงแล้ว ออปแอมป์ก็ไม่ได้มีคุณสมบัติอย่างในอุดมคติเลยทีเดียว แต่อย่างไรก็ตาม เพื่อความสะดวกในการทำความเข้าใจ สำหรับผู้เริ่มต้นในช่วงแรกให้ถือว่าออปแอมป์ที่ใช้ศึกษากันต่อไปนี้มีคุณสมบัติใกล้เคียงกับอุดมคติมาก

การใช้ออปแอมป์ที่เป็นอุดมคติในการออกแบบวงจรนั้น เป็นวิธีการที่นิยมใช้กันมาก หลังจากออกแบบเสร็จแล้ว จึงค่อยมาพิจารณาเงื่อนไข หรือคุณสมบัติต่าง ๆ ที่ไม่เป็นอุดมคติ แล้วค่อยแก้ไขและปรับปรุงวงจรให้ดีขึ้นอีกทีหนึ่ง

แรงดันออฟเซตทางอินพุต

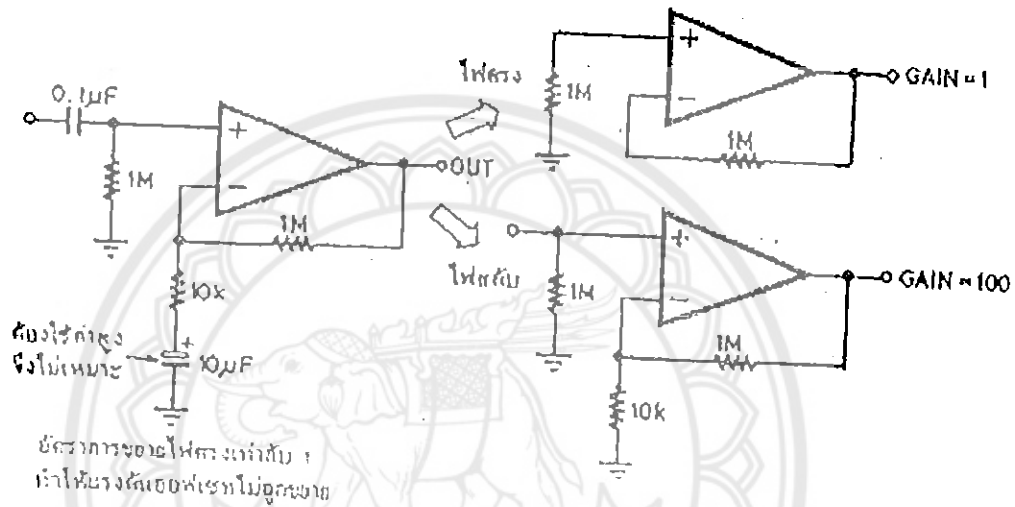
แรงดันออฟเซตทางอินพุตเป็นเรื่องสำคัญมากเรื่องหนึ่ง เมื่อเราศึกษาเรื่องออปแอมป์ แรงดันขนาดเล็กที่ปรากฏระหว่างอินพุตบวกลบ ของออปแอมป์ในขณะที่แรงดันอินพุตเป็นศูนย์ ลองดูรูปที่ 2.8 ประกอบ



แรงดันออฟเซตจะถูก
ขยายไปออกที่เอาต์พุต

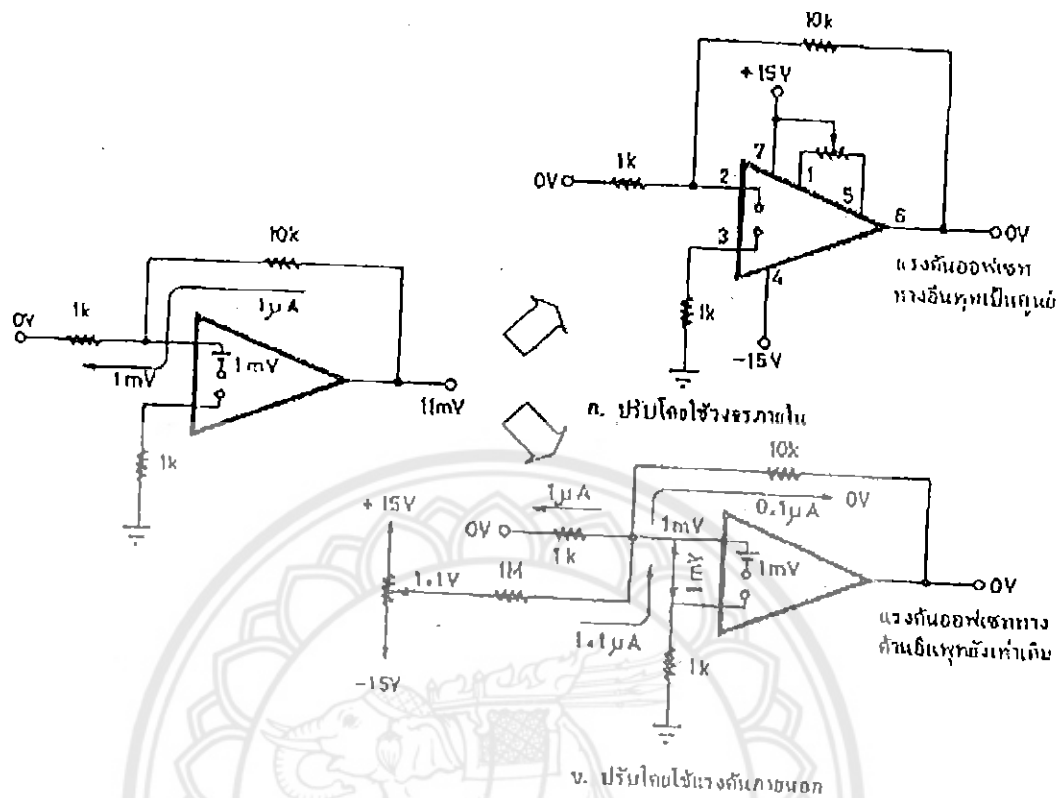
รูปที่ 2.8 แรงดันออฟเซตเปลี่ยนจาก 1 mV เป็น 10 mV ที่เอาต์พุตเมื่อออปแอมป์มีอัตราขยาย 10 เท่า

สำหรับวงจรที่ไม่ต้องการขยายแรงดันไฟตรง มักพยายามกำจัดผลของแรงดันออฟเซตนี้ไม่ให้ไปปรากฏที่เอาต์พุตอย่างเช่น วิธีการในรูปที่ 9 ซึ่งเป็นวงจรขยายแรงดัน ไฟสลับด้วยออปแอมป์สังเกตุจากรูปทางซ้ายมีตัวเก็บประจุต่ออยู่ ซึ่งคุณสมบัติของตัวเก็บประจุคือจะกันไฟตรงเอาไว้ และยอมให้ผ่านเฉพาะไฟสลับ ดังนั้นเมื่อมีแรงดันไฟตรงป้อนกลับมาที่อินพุตลบ ตัวเก็บประจุจะทำให้สัญญาณไฟตรงทั้งหมดไหลผ่านเข้าอินพุตลบหักล้างกับแรงดันไฟตรง ที่จะออกทางเอาต์พุตได้อัตราการขยายเท่ากับ 1



รูปที่ 2.9 การกำจัดแรงดันออฟเซต โดยการให้ขยายเฉพาะแรงดันที่เป็นไฟสลับ

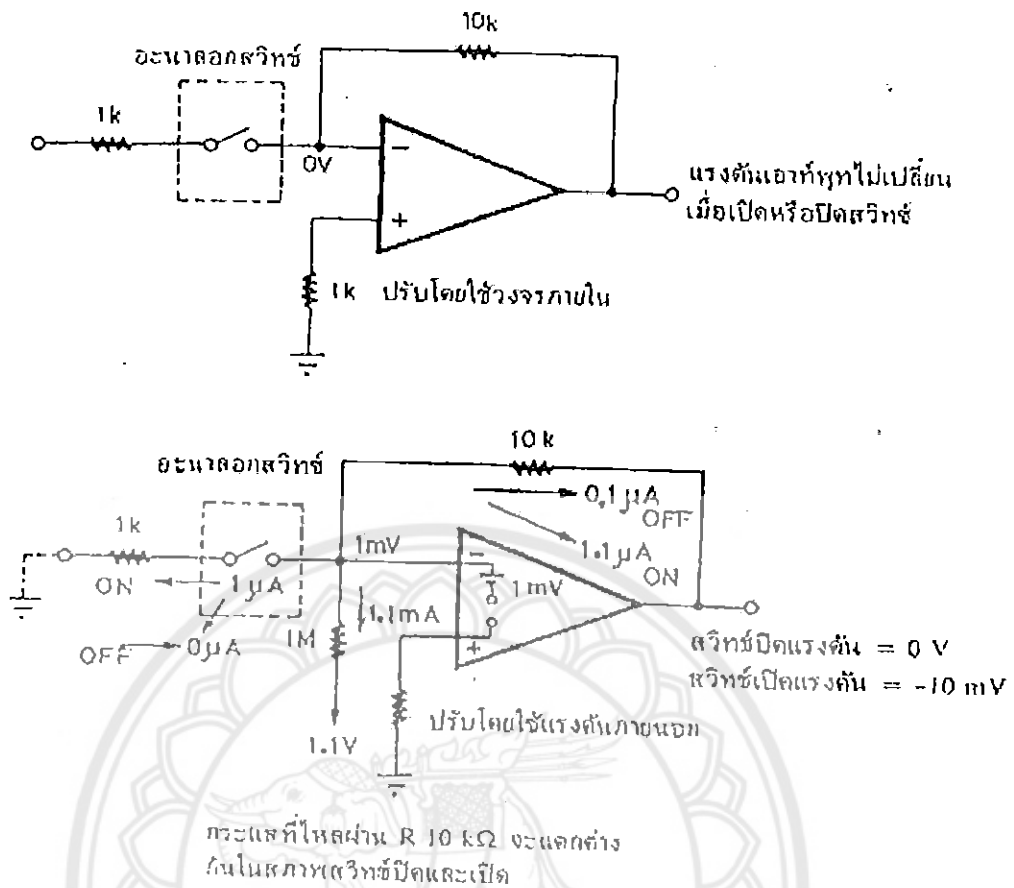
เมื่อมีสัญญาณไฟสลับผ่านเข้ามาจากเอาต์พุต มันจะเลือกไหลผ่านตัวเก็บประจุลงดินไปเสมือนกับว่าตัวต้านทานค่า 10 กิโลโอห์ม ในวงจรถูกต่อลงดินกลายเป็นรูปทางขวามือล่าง เมื่อคำนวณอัตราขยายจากค่าของอุปกรณ์แล้วได้เท่ากับ 100 เท่า



รูปที่ 2.10 แสดงวิธีการปรับออฟเซต

การปรับเพื่อลดขนาดแรงดันออฟเซตที่อินพุตสามารถทำได้ 2 วิธีคือ ใช้วงจรรภายในของออปแอมป์ โดยต่ออุปกรณ์ปรับเข้ากับขาของออปแอมป์และอีกวิธีหนึ่งคือ ป้อนแรงดันจากภายนอกเข้าไปลบล้างผลของแรงดันออฟเซตนี้การป้อนแรงดันจากภายนอกแม้จะลดผลของออฟเซตไม่ให้ไปปรากฏที่เอาต์พุตได้จริง แต่อย่าลืมว่า แรงดันออฟเซตทางอินพุตของออปแอมป์ ก็ยังไม่ได้หายไปไหน ดูรูปที่ 2.10 ประกอบ

ซึ่งเรื่องนี้อาจจะเป็นปัญหาได้เมื่อวงจรออปแอมป์ซับซ้อนขึ้น หรือในวงจรที่ใช้อนาล็อกสวิตช์ (Analogueswitch) อนาล็อกสวิตช์เป็นสวิตช์อิเล็กทรอนิกส์ ชนิดหนึ่งมีการทำงานคล้ายคลึงกับรีเลย์ เพียงแต่ไม่มีหน้าสัมผัสเชิงกลแบบรีเลย์ ใช้ในการเปิดปิดสัญญาณไฟฟ้าภายใน มักจะทำด้วยทรานซิสเตอร์หรือเฟ็ท อนาล็อกสวิตช์ที่มีขายตามท้องตลาดในปัจจุบัน มักจะเป็นชนิดที่ทำจากเฟ็ททั้งนั้น

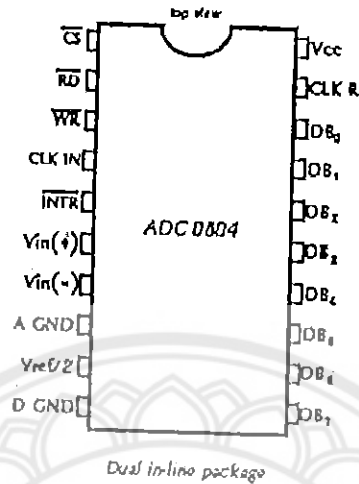


รูปที่ 2.11 แสดงการใช้อนาล็อกสวิตช์เพื่อเปิดปิดวงจรทางอินพุต

ตัวอย่างในรูปที่ 2.11 จะเห็นว่าการปรับออฟเซตโดยใช้แรงดันภายนอกจะทำให้เอาต์พุตมีแรงดันต่างกันเมื่อสวิตช์อยู่ในสภาวะเปิดหรือปิด

2.3 ไอซี A/D converter

โดยจะแสดงถึงโคอะแกรมรายละเอียดของขาต่าง ๆ ของไอซี A/D converter เบอร์ ADC 0804



รูปที่ 2.12 แสดงขาต่างๆ ของ ADC 0804

ตารางที่ 2.2 ขาและคำอธิบายของ ADC 0804

Pin No.	Symbol	Input / Output or Power	Description
1	\overline{CS}	Input	Chip select line from μP - control
2	\overline{RD}	Input	Read line from μP - control
3	\overline{WR}	Input	Write line from μP - control
4	CLK IN	Input	Clock
5	\overline{INTR}	Output	Interrupt line goes to μP interrupt input
6	Vin(+)	Input	Analog voltage (positive input)
7	Vin(-)	Input	Analog voltage (positive input)
8	A GND	Power	Analog ground
9	Vref/2	Input	Alternative voltage reference (+)
10	D GND	Power	Digital ground
11	DB7	Output	MSB data out put
12	DB6	Output	Data output

Pin No.	Symbol	Input / Output or Power	Description
13	DB5	Output	Data output
14	DB4	Output	Data output
15	DB3	Output	Data output
16	DB2	Output	Data output
17	DB1	Output	Data output
18	DB0	Output	LSB data output
19	CLK R	Input	Connect external resistor for clock
20	Vcc	Power	Positive of 5V power supply and primary reference voltage

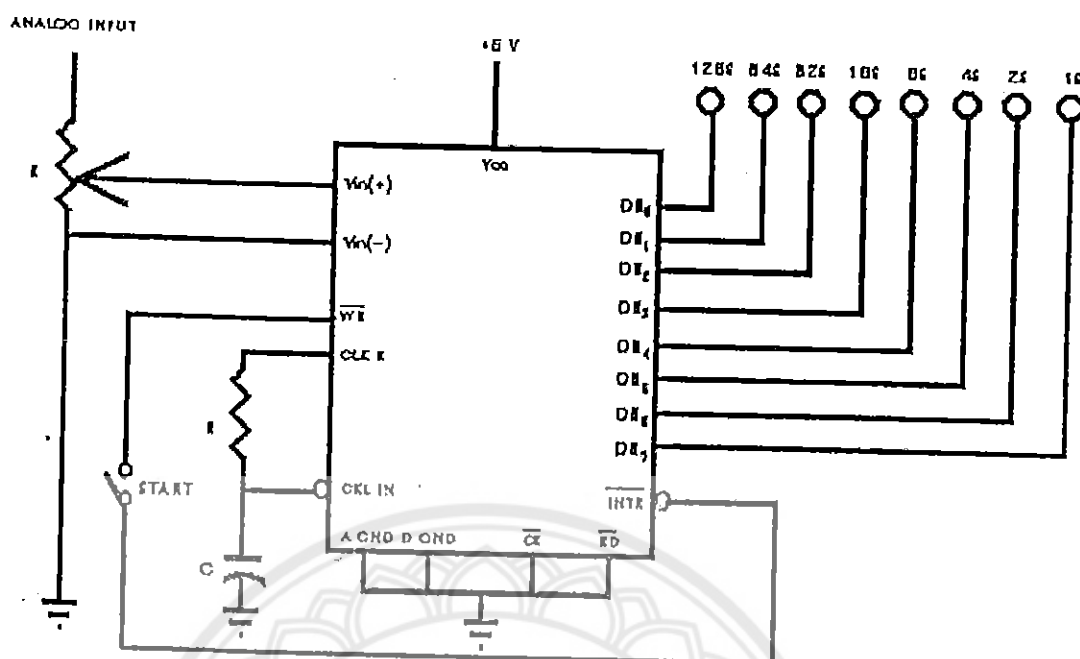
A/D Converter เบอร์ 0804 ถูกออกแบบมาให้สามารถเชื่อมต่อเข้าโดยตรงกับไมโครโปรเซสเซอร์หลาย ๆ เบอร์ เช่น Z80 8085 8080

บางขาของไอซี ADC 0804 อาจเหมือนขาของไมโครโปรเซสเซอร์ ที่เราใช้งานกันทั่วไป ไอซี ADC 0804 สามารถเชื่อมต่อกับไมโครโปรเซสเซอร์แบบ 8 บิต ที่นิยมใช้กันทั่วไปชนิดอื่น ๆ ได้อีกด้วยเช่น 6800 และ 6502

ขา \overline{CS} Control Input ใช้สำหรับสัญญาณ (Chip Select) จากวงจรถอดรหัสค่าแอดเดรสในไมโครโปรเซสเซอร์

ADC 0804 เป็น A/D แบบประมาณค่า (Successive – Approximation A/D Converter) แบบ 8 บิต CMOS มีเอาต์พุตสามสถานะ (Three State Output) ดังนั้นจึงสามารถที่จะเชื่อมต่อเข้าโดยตรงกับระบบไมโครโปรเซสเซอร์ฐานระบบ (Base System) ทางบัสข้อมูลของไมโครโปรเซสเซอร์ได้ (Data Bus)

ADC 0804 มีเอาต์พุตเป็นเลขฐานสองและมีการแปลงเวลา (Conversion time) เพียง 10μ S เท่านั้น อินพุตและเอาต์พุตของมันเข้ากันได้ทั้ง CMOS และ TTL มีตัวกำเนิดสัญญาณนาฬิกา (Clock) รวมอยู่ในชิปสำเร็จรูปอยู่แล้ว โดยจะต้องต่อกับอุปกรณ์ภายนอก เช่น ตัวต้านทาน ตัวเก็บประจุ เพิ่มเติมเพื่อให้ทำงานได้



รูปที่ 2.13 แสดงการต่อใช้งานของ ADC 0804

ไอซี ADC 0804 ทำงานด้วยไฟ DC 5V จากพาวเวอร์ซัพพลายและสามารถใส่รหัสความต่างศักย์อนาล็อกทางอินพุตได้ตั้งแต่ 0 ถึง 5V ไอซี ADC 0804 Converter สามารถนำมาทดสอบโดยใช้วงจรการต่อใช้งานของไอซี A/D 0804

หน้าที่ของวงจรคือใส่รหัสความแตกต่างของศักย์ไฟฟ้าระหว่าง $V_{in (+)}$ และ $V_{in (-)}$ เปรียบเทียบกับระดับแรงดันอ้างอิงใช้ 5.2 V เพื่อให้สัมพันธ์กับเลขฐานสอง

ตัวอย่างเช่น ADC 0804 แบบ 8 บิต หรือ 0.39 เฟอร์เซ็นต์ หมายความว่า ในแต่ละ 0.02 V (5.12 x 0.39) ที่เพิ่มศักย์ไฟฟ้าที่อินพุตเป็นอนาล็อก (Analog Input) ตัวเลขฐานสองจะนับเพิ่ม 1 สวิตช์เริ่มต้น (Start) ถูกปิด (Close) เป็นครั้งแรกและถูกเปิด (Open) เพื่อเริ่มต้นการทำงานอิสระ (Free Running) ของ A/D

\overline{WR} Input ในที่นี้จะถือเป็นอินพุตสัญญาณนาฬิกา (Clock Input) ที่มีเอาต์พุตอินเทอร์รัปต์ (INTR) สัญญาณ \overline{WR} ที่จุดสิ้นสุดการเปลี่ยนอนาล็อกเป็นดิจิตอลในแต่ละครั้ง

การเปลี่ยนจาก Low เป็น High ของสัญญาณที่ \overline{WR} อินพุตเป็นการเริ่มกระบวนการทำงานในการเปลี่ยนแปลงอนาล็อกเป็นดิจิตอลของ A/D

เมื่อการแปลงเสร็จสิ้นลง ส่วนที่แสดงผลเลขฐานสองก็จะเปลี่ยนแปลงและเอาต์พุต \overline{INTR} จะให้พัลส์ลบ พัลส์อินเทอร์รัปต์ที่เป็นลบจะถูกป้อนกลับสัญญาณอินพุต \overline{WR} และจะถูกนำไปยังการเปลี่ยน A/D ครั้งต่อไป

อัตราการเปลี่ยนแปลง A/D ของ ADC 0804 นับว่าค่อนข้างสูง เพราะใช้เทคนิคการประมาณค่า (Successive – Approximation) ในกระบวนการเปลี่ยนตัวด้านทานและตัวเก็บประจุต่อเข้ากับ CLK R และ CLK IN อินพุตให้กับไอซี ทำให้สัญญาณนาฬิกาภายในเกิดการต้านข้อมูลเอาท์พุต (DB0 – DB7) จะผลัดกันให้แอลอีดีแสดงผลเป็นเลขฐานสอง

2.4 เอสซีอาร์ (Silicon Controller Rectifier ; SCR)

เอสซีอาร์ เป็นอุปกรณ์ที่นำกระแสได้ทิศทางเดียว (จากขั้วแอโนดไปยังขั้วแคโทด) ที่ประกอบด้วยสารกึ่งตัวนำชนิด p และ n วางตัวซ้อนสลับกัน 4 ชั้น ซึ่งทำให้เกิดบริเวณรอยต่อ 3 แห่งคือ J_1 , J_2 , และ J_3 ดังรูป 14

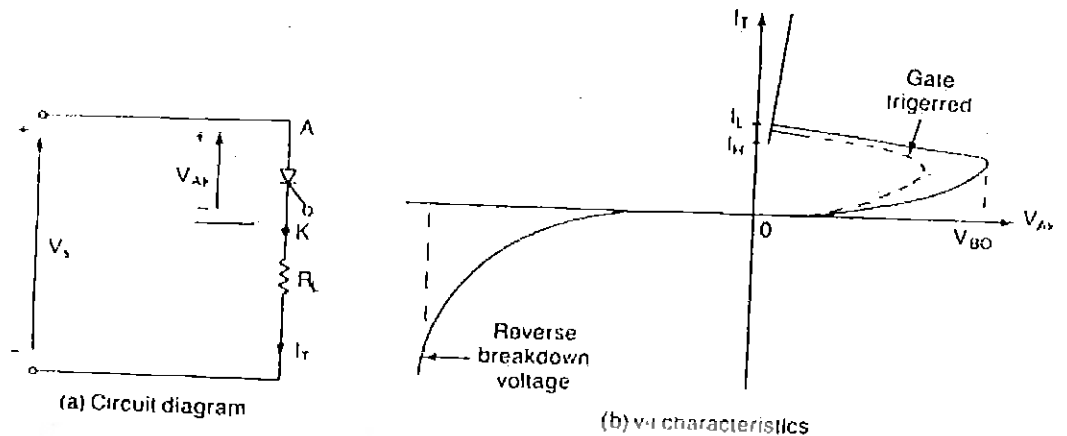


รูปที่ 2.14 แสดงสัญลักษณ์และโครงสร้างพื้นฐานของเอสซีอาร์

จากรูปที่ 2.14 แสดงสัญลักษณ์และโครงสร้างพื้นฐานของเอสซีอาร์ สังเกตว่าเอสซีอาร์แตกต่างกับไดโอดคือ มีขั้วที่สาม นอกเหนือจากขั้วแอโนด(Anode;A) และขั้วแคโทด(Cathode;K) คือขั้วเกต (Gate ; G) ซึ่งทำหน้าที่เป็นสวิตช์เปิด (on) และเปิด (off) เอสซีอาร์

2.4.1 หลักการทำงานของเอสซีอาร์

ตามปกติการเปิดเอสซีอาร์ให้นำกระแส ทำได้โดยป้อนแรงดันไฟฟ้าที่สัมพันธ์กับ 2 ส่วน คือป้อนแรงดันไบอัสตรง[ป้อนแรงดันบวกให้แก่ขั้วแอโนดเมื่อเทียบกับแคโทด(V_{AK})] และป้อนแรงดันเกต (V_G) ที่เป็นบวกที่ขั้วเกต [ป้อนแรงดันบวกให้แก่ขั้วเกต] ในทางปฏิบัติจะเป็นสัญญาณพัลส์หรือสัญญาณจุดชนวนหรือสัญญาณทรigger (ซึ่งเกิดจาก IC หรือวงจรภายนอก)



รูปที่ 2.15 แสดงการทำงานของเอสซีอาร์

ขณะ SCR นำกระแสอยู่นั้น ทรานซิสเตอร์ที่กระแสไหลผ่าน SCA มีค่ามากกว่าระดับกระแสแลตชิ่ง (Latching Current ; I_L) ของ SCR SCR นี้จะยังคงรักษาสถานะนำกระแสอย่างต่อเนื่อง

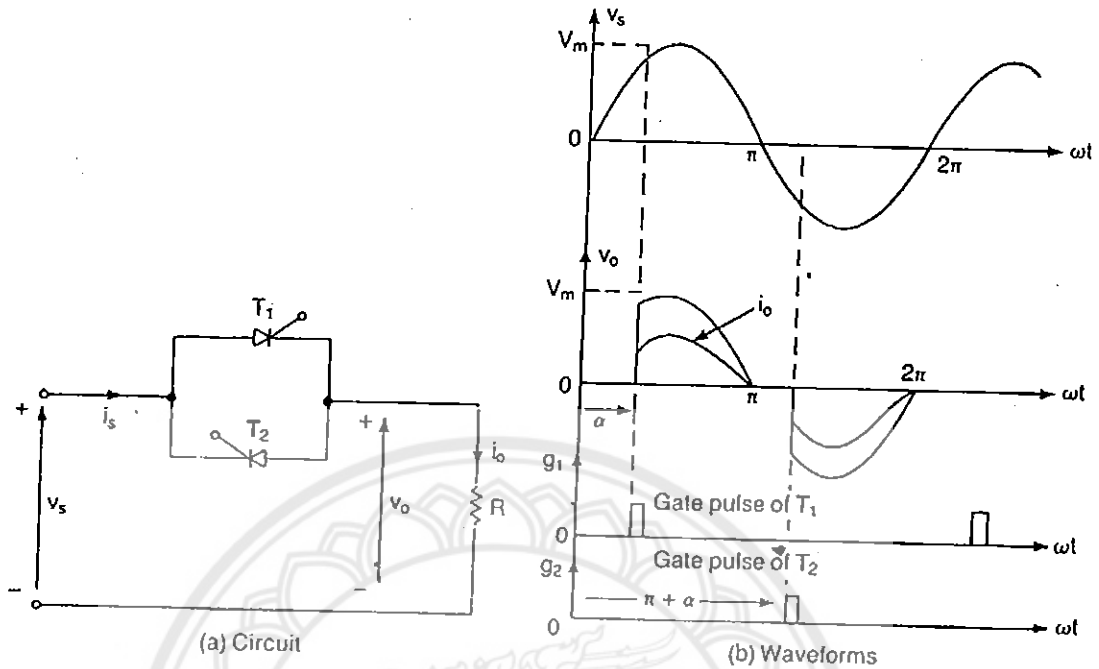
ถ้าต้องการให้ SCR ปิดหรือหยุดนำกระแส จะต้องทำให้กระแสที่ไหลผ่านมีปริมาณน้อยกว่าระดับกระแสโฮลดิ้ง (Holding Current ; I_H) ของ SCR

ถ้าเราป้อนแรงดันไบอัสตรง (V_{AK}) ให้กับ SCR เพียงส่วนเดียวโดยปราศจาก V_G ขณะ V_{AK} มีค่าน้อย จะมีเพียงกระแสรั่วไหล (Leakage Current) จำนวนเล็กน้อยเท่านั้นที่ไหลผ่าน SCR แต่เมื่อเพิ่ม V_{AK} มากขึ้น จนถึงค่าแรงดันพังทลายทีศรีเวิร์ด (Forward Breakdown Voltage ; V_{BO}) กระแสรั่วไหลจะมีปริมาณมากกว่า I_L เอสซีอาร์จึงนำกระแสได้ แต่ถ้าปริมาณของกระแสรั่วไหลมีค่ามากกว่ากระแสที่พิกัด (Rated Current) ของ SCR ก็จะทำให้ SCR เสียหาย ดังนั้นจึงไม่ควรเปิด S_{CR} โดย ป้อนแรงดันไบอัสตรง (V_{AK}) เพียงส่วนเดียวโดยไม่มี V_G

ถ้าเราป้อนแรงดันไบอัสกลับ [ป้อนแรงดันที่เป็นลบให้กับขั้วแอนโอดเมื่อเทียบกับขั้วแคโทด] เพียงส่วนเดียวหรือป้อน V_G ให้กับ SCR ด้วยก็ตาม เป็นผลให้ SCR ไม่นำกระแส ถ้าระดับแรงดันไบอัสกลับนี้มีค่าสูงกว่าแรงดันพังทลายทีศรีเวิร์ด (Reverse Breakdown Voltage) จะทำให้ SCR เสียหาย

2.4.2 การควบคุมแบบสองทิศทางเฟสเดียวที่มีโหลดความต้านทาน

การป้องกันปัญหาเกี่ยวกับองค์ประกอบ dc ที่ปนอยู่กับกระแสเอาต์พุตทำได้โดยการควบคุม 2 ทิศทาง (หรือการควบคุมแบบเต็มคลื่น) ดังรูป 2.16



รูปที่ 2.16 แสดงการปรับมุมทริกเกอร์

พัลส์ที่เกตของ T_1 และ T_2 อยู่ห่างกัน 180° ดังนั้น ในช่วงครึ่งไซเคิลด้านบวกของแรงดันอินพุตจะควบคุมการไหลของกำลังไฟฟ้าได้โดยปรับมุมทริกเกอร์ของ T_1 และควบคุมการไหลของกำลังไฟฟ้าในช่วงครึ่งไซเคิลด้านลบของแรงดันอินพุตได้โดยปรับมุมทริกเกอร์ของ T_2 ดังรูป 16

ถ้า $v_s = \sqrt{2} V_s \sin \omega t$ เป็นแรงดันอินพุตชั่วขณะ และมุมทริกเกอร์ของ T_1 และ T_2 มีค่าเท่ากัน โดย $\alpha_1 = \alpha_2 = \alpha$ จะหาค่าแรงดันเอาต์พุต rms ได้จาก

$$\begin{aligned}
 V_o &= \left[\frac{2}{2\pi} \int_{\alpha}^{\pi} 2V_s \sin^2 \omega t \, d(\omega t) \right]^{1/2} \\
 &= \left[\frac{4V_s^2}{4\pi} \int_{\alpha}^{\pi} (1 - \cos 2\omega t) \, d(\omega t) \right]^{1/2} \\
 &= V_s \left[\frac{2}{2\pi} \int_{\alpha}^{\pi} 2V_s \sin^2 \omega t \, d(\omega t) \right]^{1/2} \dots\dots(1)
 \end{aligned}$$

จากสมการ 1 เมื่อปรับค่า α จาก 0 ไปยัง π , ค่า V_o ก็จะเปลี่ยนแปลงค่าจาก V_s ไปเป็น 0 นอกจากนี้ Power Factor ของอินพุต ($I_s = I_o$) มีค่าเป็น

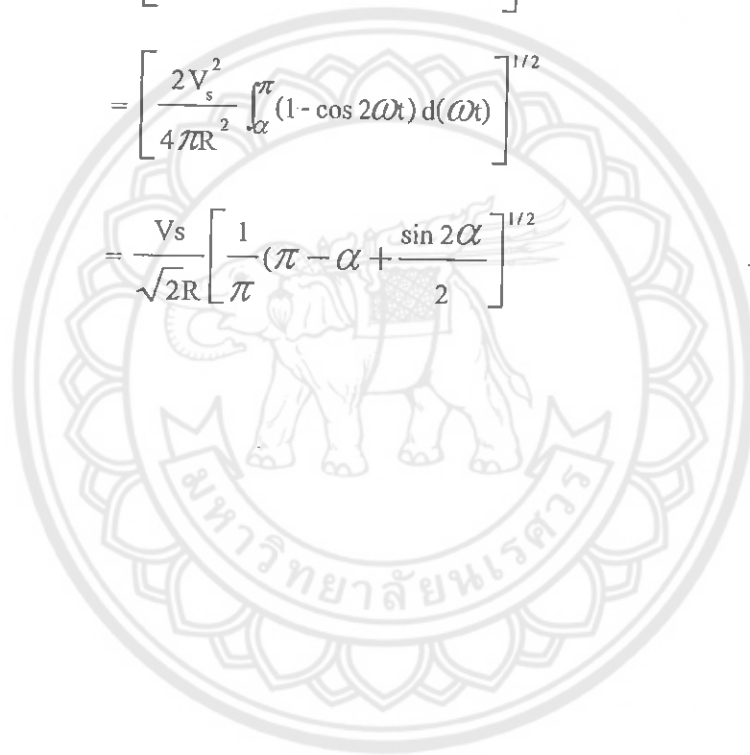
$$PF = \frac{P_o}{VA} = \frac{V_o}{V_s} = \left[\frac{1}{\pi} \left(\pi - \alpha + \frac{\sin 2\alpha}{2} \right) \right]^{1/2} \dots\dots(2)$$

ค่าเฉลี่ยของกระแสที่เอสซีอาร์ (IA) มีค่าเป็น

$$\begin{aligned}
 IA &= \frac{1}{2\pi R} \int_{\alpha}^{\pi} 2V_s^2 \sin^2 \omega t d(\omega t) \\
 &= \frac{\sqrt{2} V_s}{2\pi R^2} (\cos \alpha + 1)
 \end{aligned}$$

ค่า rms ของกระแสที่เอสซีอาร์ (I_R) มีค่าเป็น

$$\begin{aligned}
 IR &= \left[\frac{1}{2\pi R^2} \int_{\alpha}^{\pi} 2V_s^2 \sin^2 \omega t d(\omega t) \right]^{1/2} \\
 &= \left[\frac{2V_s^2}{4\pi R^2} \int_{\alpha}^{\pi} (1 - \cos 2\omega t) d(\omega t) \right]^{1/2} \\
 &= \frac{V_s}{\sqrt{2}R} \left[\frac{1}{\pi} (\pi - \alpha) + \frac{\sin 2\alpha}{2} \right]^{1/2}
 \end{aligned}$$



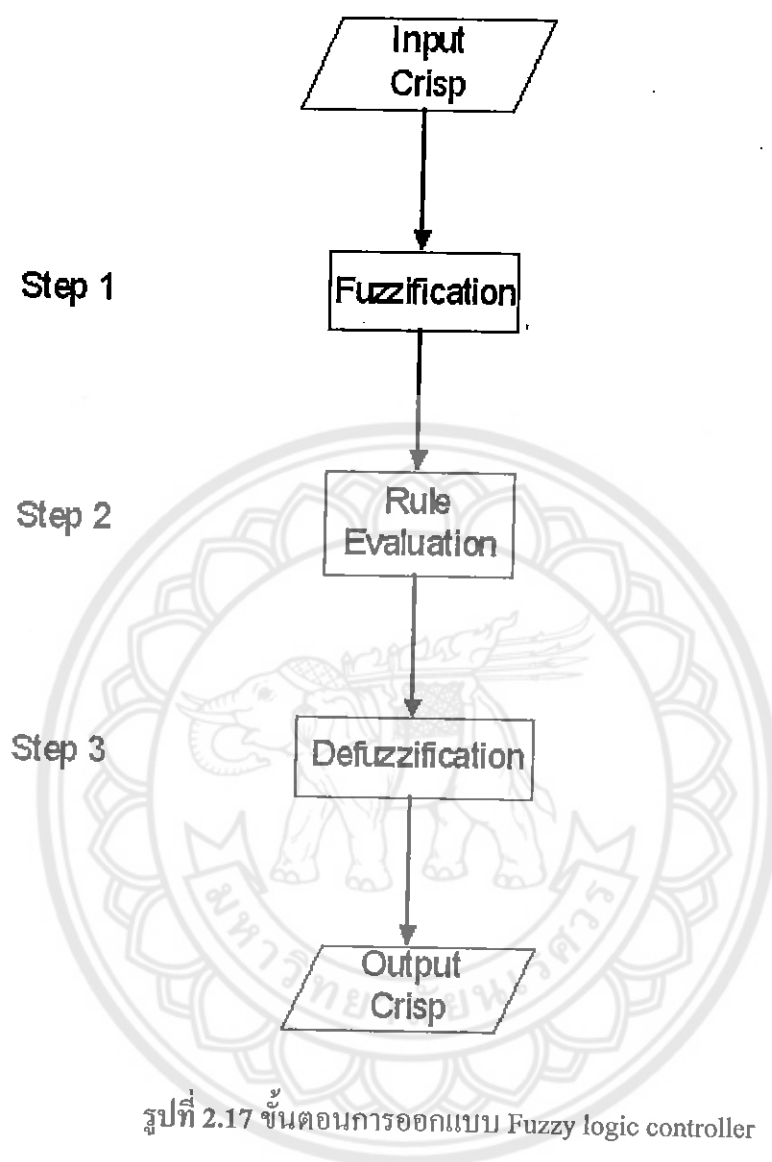
2.5 ทฤษฎี Fuzzy Logic

Fuzzy Logic ระบบ Fuzzy control เป็นระบบที่เรียกว่า Rules-base system ซึ่งหมายความว่า จะมี Fuzzy rules หรือกฎของ Fuzzy เป็นตัวตัดสินใจกลไกการทำงานของระบบควบคุมนั้นให้มีการปรับเปลี่ยนไปตามผลกระทบที่เกิดขึ้นในระบบ จุดมุ่งหมายของระบบ Fuzzy control ก็คือทำให้ Fuzzy Rules-base system นี้เข้าไปแทนการควบคุมแบบเก่า ซึ่งใช้มนุษย์เป็นผู้ควบคุม (Skilled human operator)

ลักษณะเด่นของการควบคุมแบบฟัซซี่

- ไม่ใช่แบบจำลองทางคณิตศาสตร์ในการควบคุม (Model-Free Estimator)
- สามารถพัฒนาให้การควบคุมคล้ายกับพฤติกรรมของมนุษย์ได้
- ไม่ใช่พื้นฐานทางคณิตศาสตร์ที่ยุ่งยากในการคำนวณ
- สามารถเข้าถึงปัญหาระบบได้เร็ว จึงสามารถทำการควบคุมได้อย่างทันเวลา
- ถ้าสามารถสร้างฟังก์ชันความเป็นสมาชิกได้อย่างถูกต้องจะใช้กฎในการควบคุมเพียงไม่กี่ข้อ
- สามารถทำการคำนวณค่าแต่ละค่าของแต่ละอินพุต-เอาต์พุตไว้ล่วงหน้าแล้วจัดเก็บไว้ในหน่วยความจำประเภทรอมใดเพื่อให้การทำงานเร็วขึ้น เนื่องจากการคำนวณไว้ล่วงหน้าแล้ว จึงสามารถควบคุมระบบได้โดยการใช้ข้อมูลจากรอม ซึ่งอยู่ในรูปของตารางค้นหา
- สามารถประยุกต์ใช้ได้เป็นอย่างดีในสาขาระบบผู้เชี่ยวชาญ การทำการตัดสินใจและการประมวลผลสารสนเทศ
- เป็นการควบคุมโดยการวิเคราะห์เชิงคุณภาพ

ขั้นตอนการออกแบบ Fuzzy logic controller

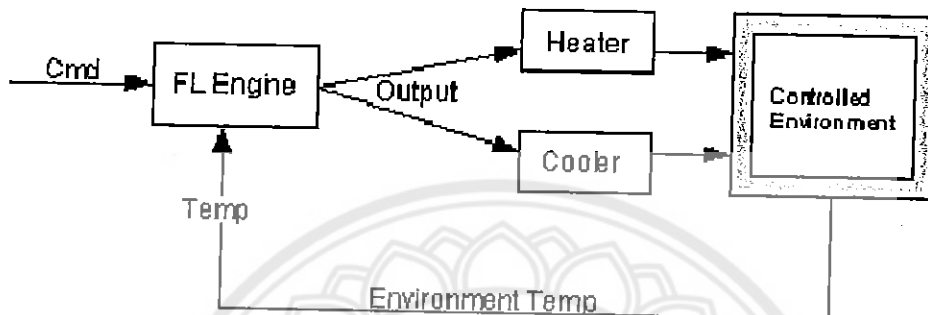


รูปที่ 2.17 ขั้นตอนการออกแบบ Fuzzy logic controller

Input Crisp

ตัวแปรในลักษณะของคำพูด (Linguistic variables)

SIMPLE FL CONTROL SYSTEM



รูปที่ 2.18 block diagram ของระบบตัวอย่าง

Cmd: อุณหภูมิที่ตั้งไว้

Temp: อุณหภูมิที่วัดได้ในเตาอบ

Error: อุณหภูมิที่ตั้งไว้ - อุณหภูมิที่วัดได้ในเตาอบ

Error-dot: อัตราการเปลี่ยนแปลงของอุณหภูมิ

Output: ค่าที่ได้ออกมาจาก Fuzzy Logic

จากรูป 2.18 Cmd คือค่าอุณหภูมิที่เราเซตขึ้นมาจากระบบจะเช็คอุณหภูมิเข้าทาง Temp แล้วทำการเทียบโดยผ่านขบวนการของ Fuzzy Logicจากระบบจะทำการสั่ง Microcontroller

ทำงานเช่นจากรูปนี้จะทำการปรับแรงดันตามที่ Microcontroller สั่งมา ซึ่งเราจะแบ่ง Input Set ได้ดังนี้

"N" = "negative" error or error-dot input level

"Z" = "zero" error or error-dot input level

"P" = "positive" error or error-dot input level

และแบ่ง Output Set ได้ดังนี้

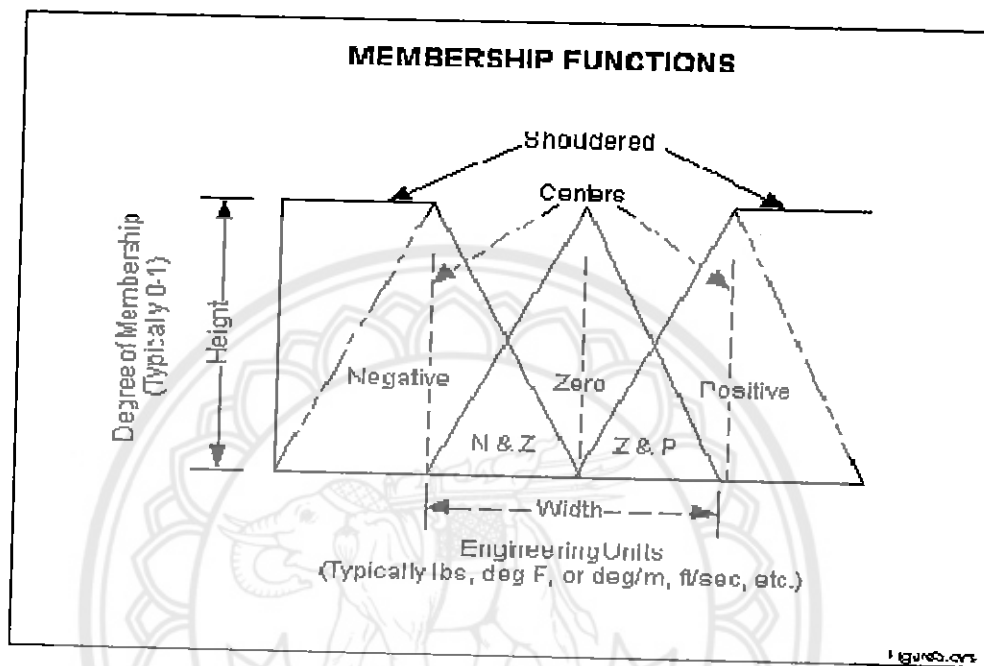
"H" = "ร้อน"

"-" = "ไม่เปลี่ยนแปลง"

"C" = "เย็น"

ซึ่งในระบบนี้กำหนดให้มีเพียง 3 กลุ่ม ซึ่งถ้าจะให้ทำงานในระบบมีความละเอียดมากขึ้น ควรแบ่งให้มีกลุ่มของสมาชิกมากขึ้น

Fuzzification คือกลุ่มของจำนวนสมาชิกที่อธิบายด้วยกราฟที่ความแตกต่างของระดับ output ต่างๆ เช่น นุ่ม, ความเร็ว แล้วระบุจำนวนสมาชิกในกลุ่ม

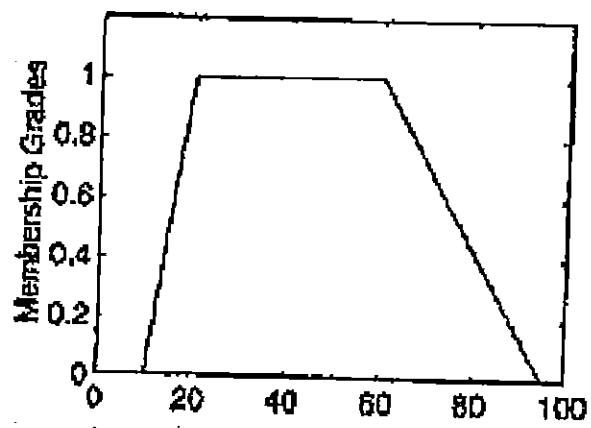


รูปที่ 2.19 ลักษณะทั่วไปของ Fuzzy set

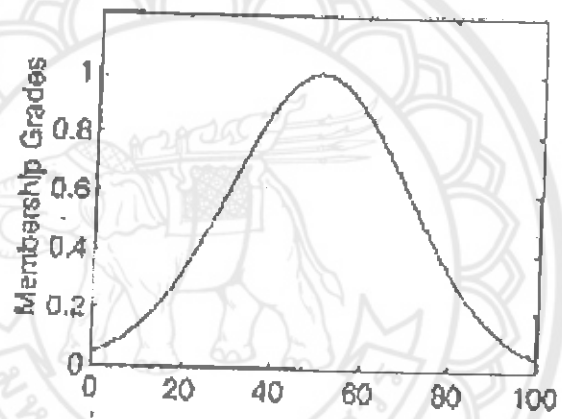
Types of Membership Function

1. Triangular Membership Function
2. Trapezoidal Membership Function
3. Gaussian Membership Function
4. Generalized bell Membership Function
5. Sigmoidal Membership Function
6. Left-right Membership Function

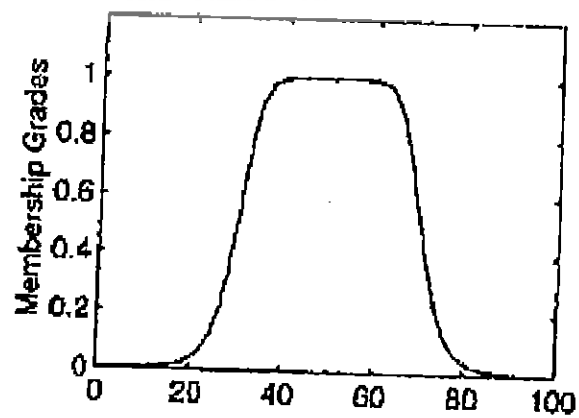
ดังรูปแสดง



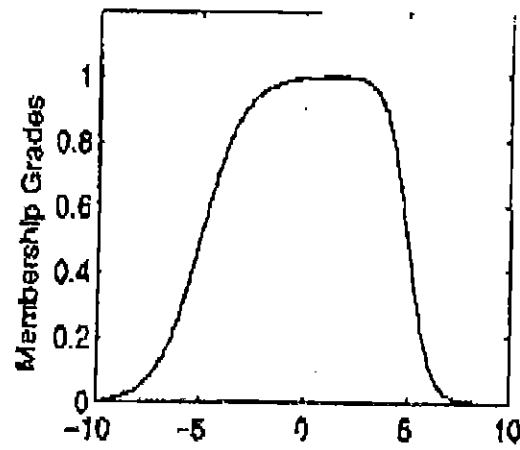
รูปที่ 2.20a Triangular Membership Function



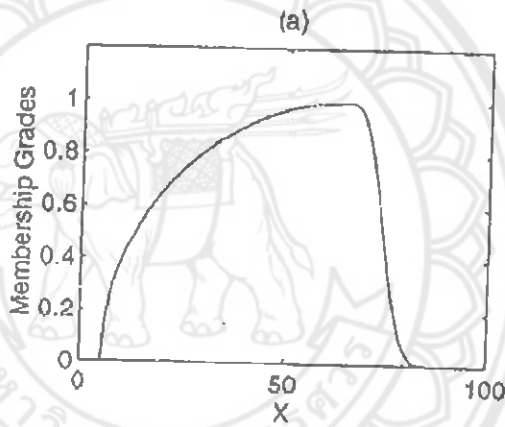
รูปที่ 2.20b Trapezoidal Membership Function



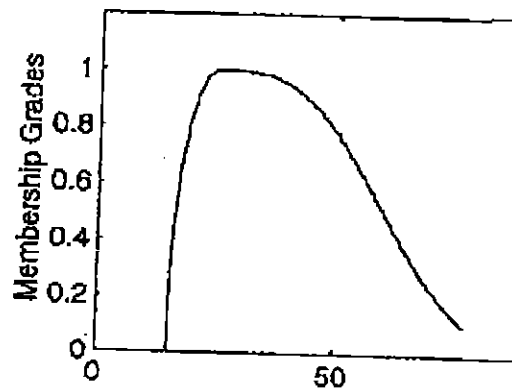
รูปที่ 2.20c Gaussian Membership Function



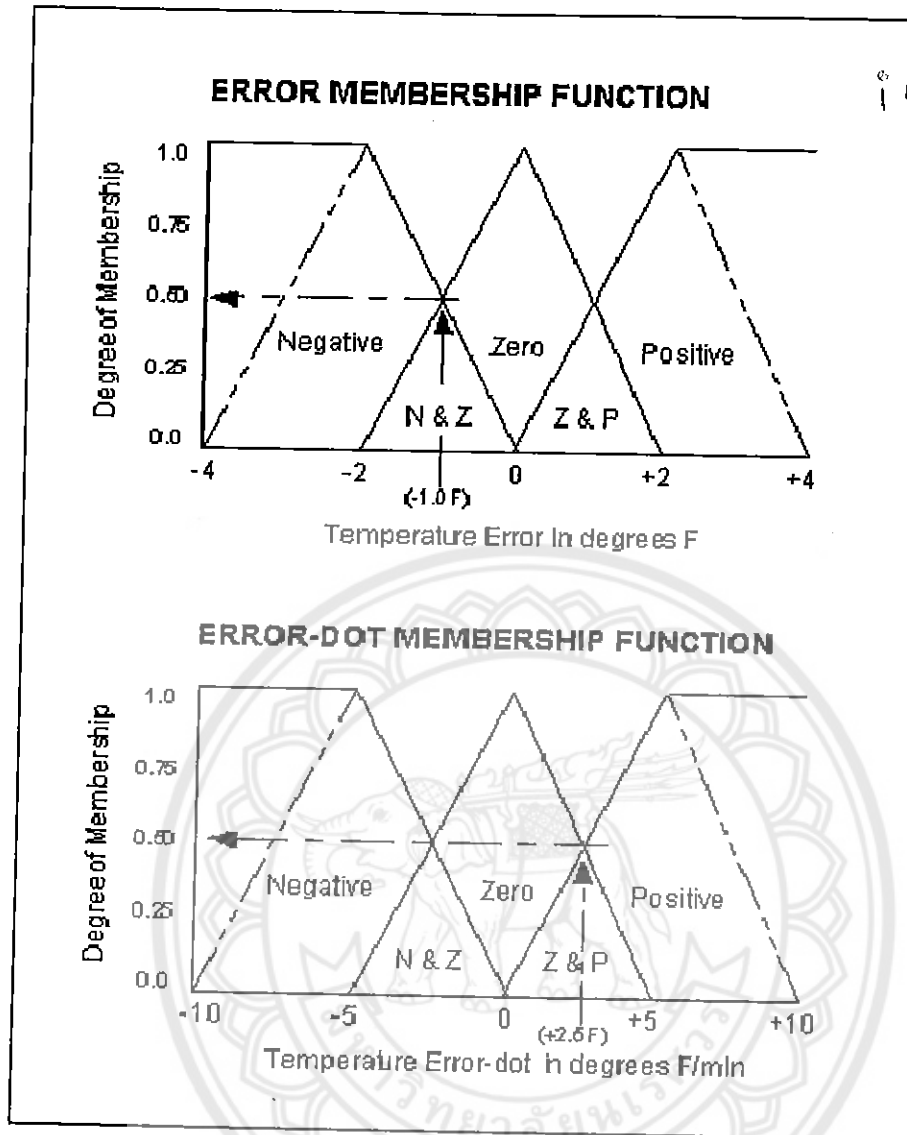
รูปที่ 2.20d Generalized bell Membership Function



รูปที่ 2.20e Sigmoidal Membership Function



รูปที่ 2.20f Left-right Membership Function



15021285

ส.ร.
ศ. 9611
2546

รูปที่ 2.21 Fuzzy set จากระบบตัวอย่าง และ input ที่วัดได้

ในรูปที่ 2.21 มีค่า input Error = -1.0 F และ Error-dot = 2.5 F/min

Rules Evaluation

จากรูปที่ 2.21

Fuzzy set ของ INPUT#1 คือ ("Error", positive (P), zero (Z), negative (N))

Fuzzy set ของ INPUT#2 คือ ("Error-dot", positive (P), zero (Z), negative (N))

Fuzzy set ของ Output คือ (Heat (H), No Change (-), Cool (C))

INPUT#1 แสดงสถานะของระบบคือ

Error = Command-Feedback

P=Too cold, Z=Just right, N=Too hot

INPUT#2 แสดงสถานะของระบบคือ

Error-dot = $d(\text{Error})/dt$

P=Getting hotter Z=Not changing N=Getting colder

OUTPUT ผลการตอบสนองต่อ input ของระบบคือ

Output H = Call for heating = Don't change anything

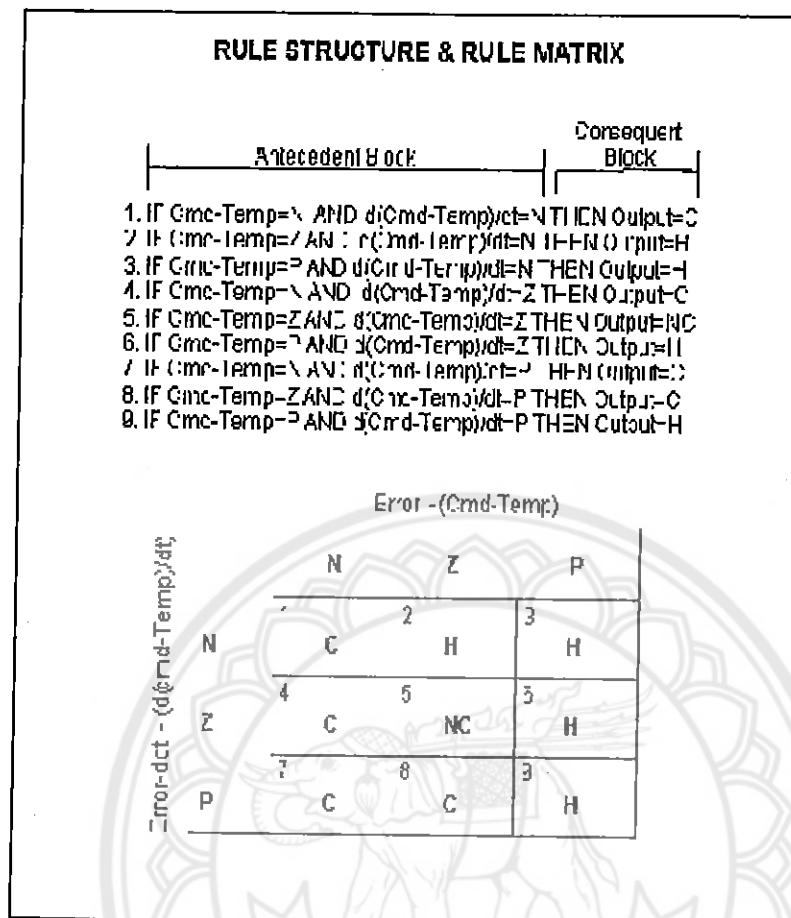
C = Call for cooling

System operating rules

ทำการแบ่งค่าตามช่วงของแต่ละกลุ่มทั้งของค่า Error และของ Error-dot เพื่อรอค่า Error ที่เข้ามาในระบบของกราฟ Error และทำการบันทึกค่าของแต่ละกลุ่มทั้ง Error และของ Error-dot เพื่อที่ใช้ในขบวนการต่อไป

Rules

คือลำดับอนุกรมที่ใกล้เคียงกันโดยใช้ If-Then ในการอธิบายเช่น IF มุมเป็นศูนย์และอัตราเร็วเป็นศูนย์ Then ความเร็วก็เป็นศูนย์ด้วยเหมือนกัน ดังตัวอย่างต่อไปนี้



รูปที่ 2.22 ตัวอย่าง Rules Structure และ Rules matrix

Inferencing

ขั้นตอนสุดท้ายคือการหา firing strength ของแต่ละกฎ และผลลัพธ์ทางลอจิก (logical products) จากแต่ละกฎ จะต้องถูกนำมาคำนวณ (max-min'd, max-dot'd, averaged, root-sum-squared, etc.) ก่อนที่จะส่งต่อไปยังขั้นตอนการ defuzzification เพื่อหา crisp output วิธีการ inference มีหลายวิธี เช่น

1. The MAX-MIN method

สมการ
$$r_{ij} = \frac{\sum_{k=1}^m \min(x_{ik}, x_{jk})}{\sum_{k=1}^m \max(x_{ik}, x_{jk})}$$

2. The AVERAGING method

สมการ

$$r_{ij} = \frac{\sum_{k=1}^m \min(x_{ik}, x_{jk})}{\frac{1}{2} \sum_{k=1}^m (x_{ik} + x_{jk})}$$

3. The ROOT-SUM-SQUARE (RSS) method

สมการ

$$r_{ij} = \sum_{k=1}^m [\min(x, y)]^{\frac{1}{2}}$$

r_{ij} คือ firing strength ของแต่ละกฎ

x คือ ค่า error

y คือ ค่า error-dot

จากตัวอย่างแสดงการ inference ด้วยวิธี ROOT-SUM-SQUARE ได้ผลลัพธ์คือพื้นที่ส่วนที่แรเงาในรูปที่ 2.23

Defuzzification

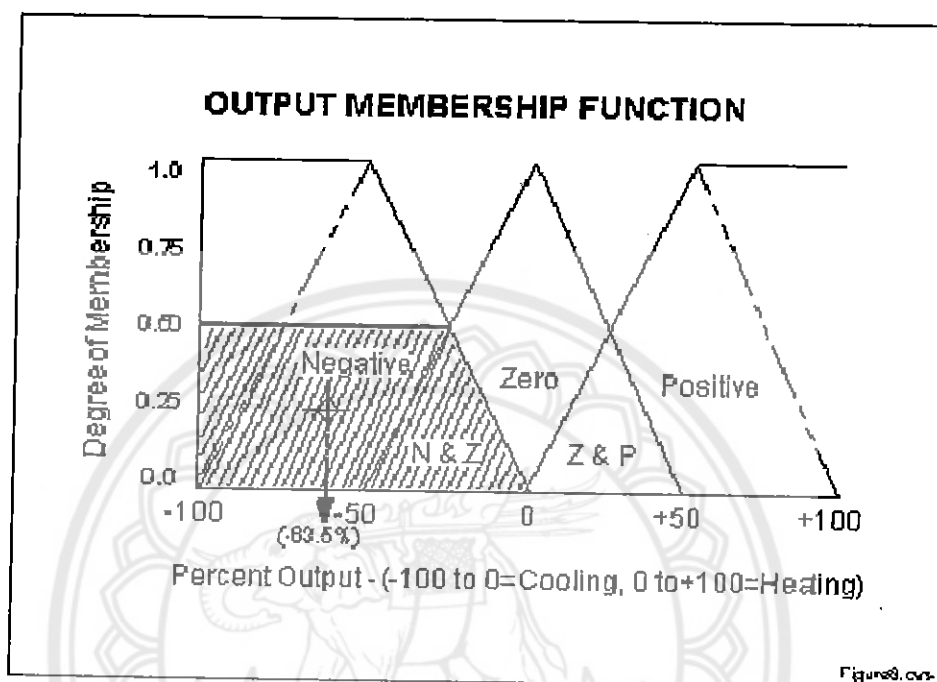
การ Defuzzification คือ การนำค่าที่ได้จากการ inference มาคำนวณหาค่าที่เป็นผลลัพธ์เพื่อนำไปควบคุมระบบต่อไป

การ Defuzzification มีหลายวิธี

1. Max-membership principle
2. Centroid method
3. Weighted average method
4. Mean-max membership
5. Centre of sums
6. Centre of largest area
7. First (or last) of maxima

Output Crisp

หลังจากการ defuzzification โดย วิธี MAX-MIN จะได้ Set ของ Output ดังส่วนที่แรเงา และหลังจากการคำนวณหาจุด Centroid และจึงได้ Output เป็นค่า -63.5% ซึ่งจะนำไปใช้ในการ ปรับอุณหภูมิของระบบต่อไป



รูปที่ 2.23 Output

บทที่ 3

การออกแบบการควบคุมอุณหภูมิในเตาอบไฟฟ้าโดยใช้พีซีลอจิก

3.1 ภาพรวมของระบบการควบคุมอุณหภูมิในเตาอบไฟฟ้าโดยใช้พีซีลอจิก

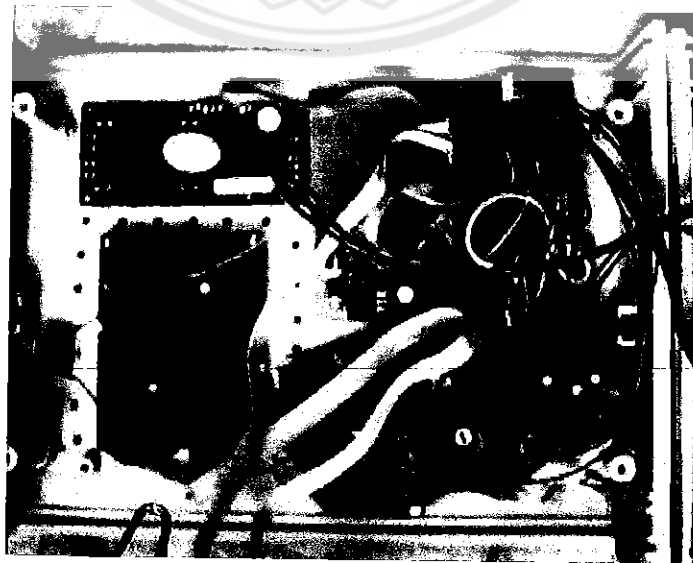
เมื่อตั้งค่าอุณหภูมิที่ต้องการผ่านทาง keypad แล้วนั้นทางด้านเทอร์โมคัปเปิลจะทำการวัดค่าอุณหภูมิของเตาอบเพื่อนำมาเปรียบเทียบกับอุณหภูมิที่ตั้งไว้ โดยผ่านขบวนการทาง พีซีลอจิก เพื่อที่จะไปสั่งให้ Microcontroller ไปสั่งให้วงจรปรับมุมทริกของแรงดันไฟฟ้าโดยใช้ SCR และให้ความร้อนแก่เตาอบตามที่ต้องการ

3.2 อุปกรณ์ที่ใช้

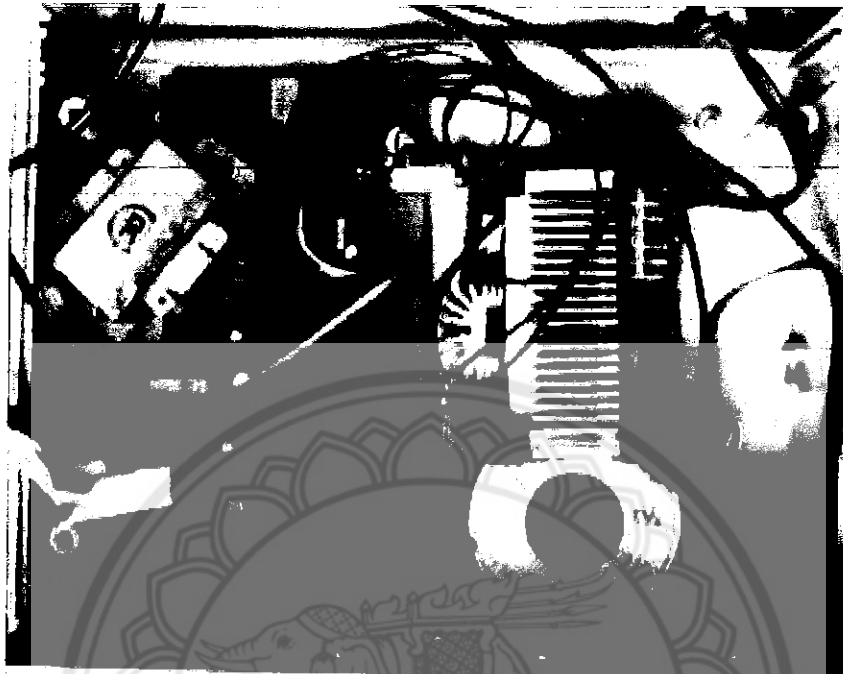
- CPU ตระกูล MCS-51 เบอร์ AT89C52
- LCD
- KEYPAD
- โปรแกรมภาษา C
- โปรแกรม RAD51 V1.07beta
- เทอร์โมคัปเปิล
- อื่น

3.3 แนวคิดและขั้นตอนการออกแบบ

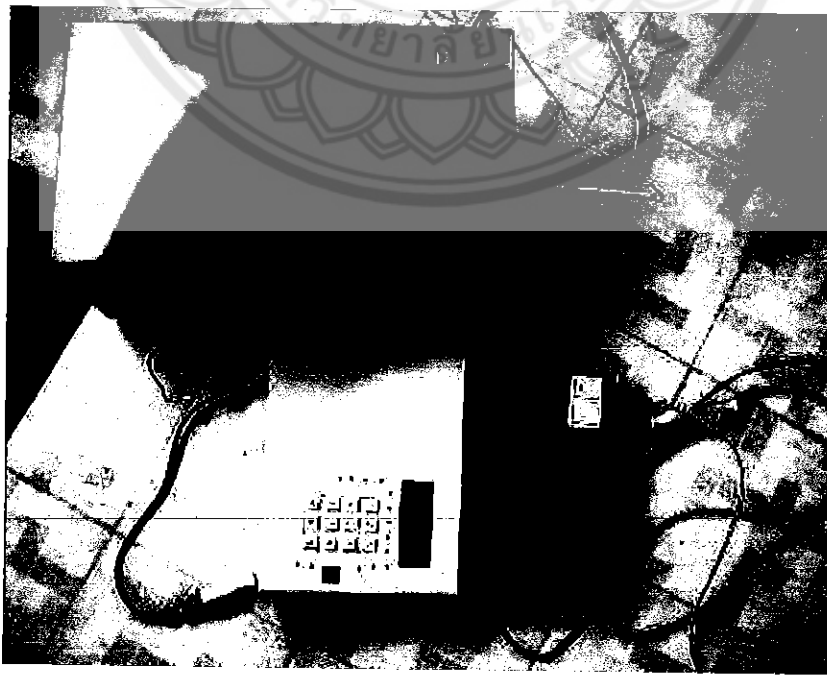
1. ทางด้าน Software และ Hardware



รูปที่ 3.1 วงจร KEYPAD , LCD และเทอร์โมคัปเปิล



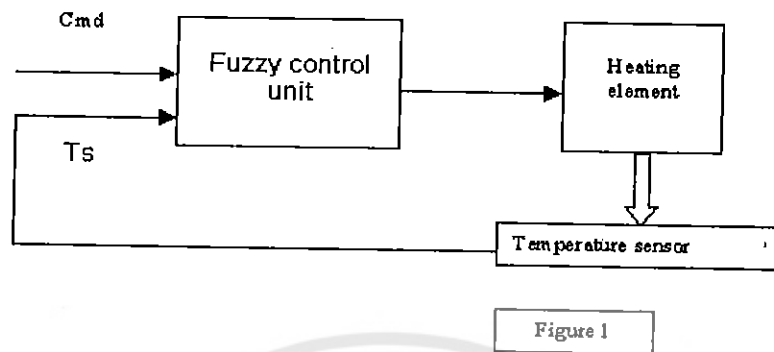
รูปที่ 3.2 วงจร SCR



รูปที่ 3.3 ภาพเตาอบ

2. การออกแบบ

รูปที่ 3.4 แสดงระบบที่ต้องการควบคุมประกอบด้วย ขดลวดทำความร้อน, เซ็นเซอร์อุณหภูมิ และ หน่วยควบคุมแบบ Fuzzy



รูปที่ 3.4 ระบบควบคุมในเตาอบ

Setpoint คือ อุณหภูมิที่ตั้งไว้

Ts คือ ค่าอุณหภูมิที่วัดได้จากเตาอบ

กระบวนการหาค่าสัญญาณ Input

ก่อนที่จะมีการวิเคราะห์ระบบด้วยวิธีการแบบฟัซซี่ (fuzzy analysis) จะต้องมีการคำนวณหาค่าสัญญาณอินพุท โดย $e(t) = (\text{Set point} - Ts)$ และ $\text{error-rate } y(t) = de(t)/dt$.

$e(t)$ และ $y(t)$ จะเป็นอินพุทของฟัซซี่ลอจิกคอนโทรลเลอร์ $e(t)$ โดยกำหนดช่วงของ $e(t)$ คือ -210 ถึง 210 องศาเซลเซียสและ คำนวณหาค่า $y(t)$ ได้จากการเขียนโปรแกรมอย่างง่าย ดังที่แสดงไว้บางส่วน ช่วงของ $y(t)$ ที่วัดได้จากโปรแกรมคือ -4 ถึง 4 องศาเซลเซียส/วินาที

```

void main(void)
{
    unsigned char x,e1=0,e2,i=0,x1=39,Num;
    lcdset();

    strcpy(LCDBUF,"Naresuan Nu. ");
    lcdout();
    dmsec(2000);
  
```

```
strcpy(LCDBUF,"Temp C Set C");
lcdout();
clear_dis();
dmsec(2000);
while(1)
{
x=getkey();

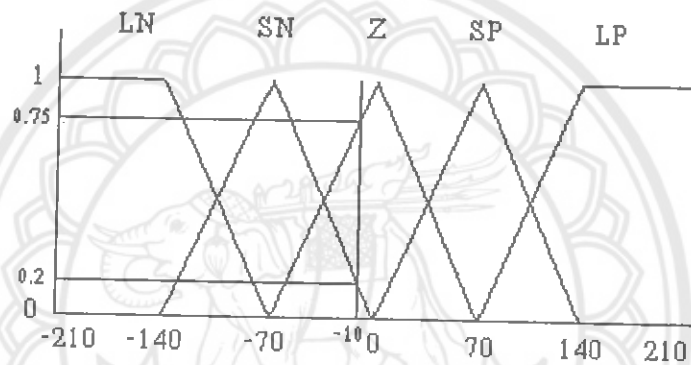
    if(x!=0xff)
    {
        if(x==14)
        {
            INPUT_NUM();
        }
        Temp=P1;
        Temp=Temp+39;
        SHOW_NUM(0x84,Temp);
i=Temp;
        e2=abs(i-x1);
        x1=i;
        if(e2>e1)
        {
            Num=e2;
        }
        else
        {
            Num=e1;
        }
        e1=Num;
```

จากโปรแกรมข้างต้นเป็นการคำนวณหาค่า Error-dot คือ เมื่อมีการรับค่าอุณหภูมิเข้ามาแล้วมีการนำมาลบอุณหภูมิเดิม แล้วทำการเก็บค่าอุณหภูมิที่เปลี่ยนแปลงไว้ เพื่อนำมาเปรียบเทียบกับค่าอุณหภูมิที่เปลี่ยนแปลงค่าใหม่ ค่าอุณหภูมิที่เปลี่ยนแปลงมากที่สุดคือช่วงของ Error-dot

กำหนด fuzzy sub-sets สำหรับ inputs

ขั้นตอนแรกการวิเคราะห์ระบบด้วยวิธีการแบบฟัซซี่ คือการแบ่ง domain ของ $e(t)$ and $y(t)$ เป็นห้า sub-domains ได้แก่: Large Negative (LN), Small Negative (SN), Zero (Z), Small Positive (SP) และ Large Positive (LP)

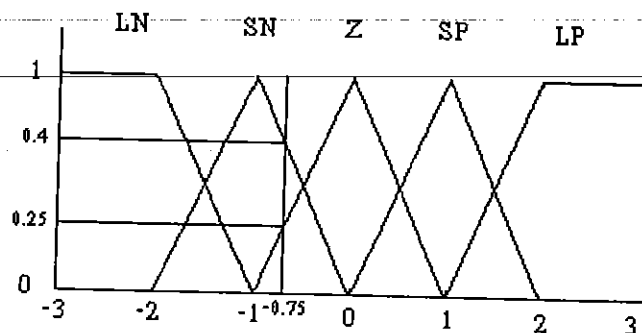
กำหนด membership functions สำหรับแต่ละ fuzzy อินพุต



รูปที่ 3.5 รูปของ $e(t)$

จากรูปที่ 3.5 membership function แสดงให้เห็นถึงค่าความเป็นสมาชิก (degree of membership) ของแต่ละอินพุตในแต่ละ fuzzy sub-set.

รูปที่ 3.6 เป็นกราฟของ Error-dot ซึ่งเป็นแบบ the triangular ประกอบด้วยห้า sub-domains คือ: Large Negative (LN), Small Negative (SN), Zero (Z), Small Positive (SP) และ Large Positive (LP)



รูปที่ 3.6 รูปของ $y(t)$

กำหนด fuzzy sub-sets สำหรับแต่ละ fuzzy output

สำหรับโครงการนี้ domain ของเอาต์พุตเป็น sub-domains: Large Negative (LN), Small Negative (SN), Zero (Z), Small Positive (SP) และ Large Positive (LP)

กำหนด membership functions สำหรับ output.

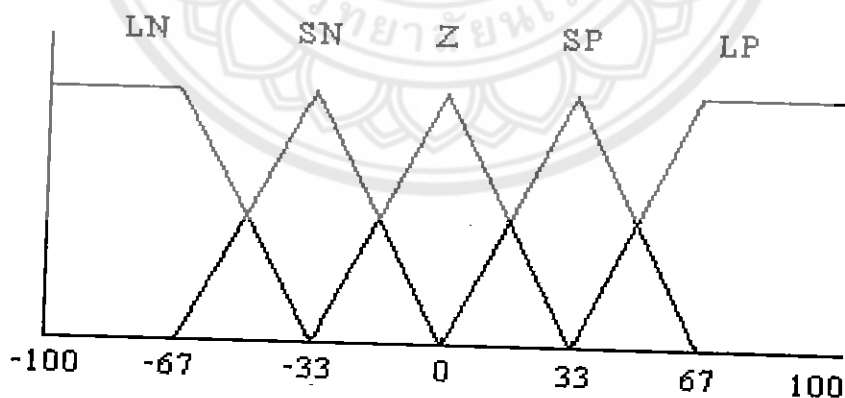
Membership function ที่ออกแบบไว้จะมีส่วนหนึ่งที่เรียกว่า "singleton".

The singleton เป็น impulse function ซึ่งเป็นค่าตัวเลข (numeric value) ค่าหนึ่งในแต่ละ sub-domain ของ output. ในโครงการนี้ ถ้าช่วงของ crisp output คือ -100-100 singleton ของ Large Negative (LN) = -67, Small Negative (SN) = -33, Zero (Z) = 0, Small Positive (SP) = 33 และ Large Positive (LP) = 67

ซึ่งเราได้กำหนดค่า output สูงสุดเท่ากับ 100% และต่ำสุดเท่ากับ -100% โดยที่ค่า output เป็นเปอร์เซ็นต์ที่ต้องเพิ่มหรือลด คือ ถ้า output = 100% หมายถึง ระบบมีอุณหภูมิต่ำกว่าค่าที่ตั้งค่าไว้และอัตราการเปลี่ยนแปลงของอุณหภูมิน้อยมาก ให้เพิ่มมุมมองของ SCR ไปอีก 100% จากของเดิมแต่ต้องไม่เกิน 360 องศา

ในทางกลับกันถ้า output = -100% หมายถึง ระบบมีอุณหภูมิสูงกว่าค่าที่ตั้งค่าไว้และอัตราการเปลี่ยนแปลงของอุณหภูมิสูงมาก ให้ลดมุมมองของ SCR ไปอีก 100% จากของเดิมแต่ต้องไม่ต่ำกว่า 0 องศา

รูปที่ 3.7 เป็น membership functions ของ Output



รูปที่ 3.7 รูปของ Output sub-domains

การสร้าง rule matrix

ผลตอบสนองของระบบถูกกำหนดโดยกฎ (rules) ธรรมดา คือ:

if (A and B) then C

ส่วน if เป็นเงื่อนไข และส่วน then เป็นผล ของกฎนั้นๆ ส่วนที่ตามหลัง then เรียกว่า "the consequent" อธิบายสถานะของ fuzzy output ของ system

A, B, C คือ ประโยค logic ซึ่งใน fuzzy logic มีค่าความจริงหรือค่าความเป็นสมาชิกอยู่ระหว่าง 0 ถึง 1

จำนวนของกฎส่วนใหญ่แล้วจะมีจำนวนเท่ากับ $N \times M$ เมื่อ N คือจำนวนของ $e(t)$: the error input subsets และ M คือ จำนวนของ $y(t)$: the error rate input subsets ขณะนี้ค่าของ $N=M=5$ ทำให้มีกฎทั้งหมด 25 กฎ

ตัวอย่างกฎ

if Error is Large Negative and Error rate is Large Negative Then output is Large Positive.

if Error is Small Positive and Error rate is Large Positive Then output is Small Negative.

สามารถแสดงกฎทั้งหมดได้ดังตาราง:

ตารางที่ 3.1 แสดงตัวอย่างกฎ

$e(t)$	LN	SN	Z	SP	LP
$y=de(t)/dt$					
LN	x=LP	x=LP	x=LP	x=LP	x=LN
SN	x=LP	x=SP	x=SP	x=SN	x=LN
Z	x=LP	x=SP	x=Z	x=SN	x=LN
SP	x=LP	x=SP	x=SN	x=SN	x=LN
LP	x=LP	x=LN	x=LN	x=LN	x=LN

x=fuzzy output

ค่าความเป็นสมาชิกของ output set จะขึ้นอยู่กับ membership functions ของ input ที่กำหนดไว้ดังรูปที่ 3.5 และ 3.6

ตัวอย่างการคำนวณ สมมติว่า $e(t)=-10$ C และ $y(t)=-0.75$ C/min.

จาก membership functions ของ input ที่กำหนดไว้ดังรูปที่ 3.5 และ 3.6

จะได้ค่าความเป็นสมาชิกของ output set

ตารางที่ 3.2 การเปรียบเทียบระหว่างค่า $e(t)$ และ $y(t)$

$e(t)$ $y(t) = de(t)/dt$	LN	SN	Z	SP	P
LN	$e=0.0$ $y=0.00$	$e=0.2$ $y=0.0$	$e=0.75$ $y=0.0$	$e=0.0$ $y=0.0$	$e=0.0$ $y=0.00$
SN	$e=0.0$ $y=0.25$	$e=0.2$ $y=0.25$	$e=0.75$ $y=0.25$	$e=0.0$ $y=0.25$	$e=0.0$ $y=0.25$
Z	$e=0.0$ $y=0.4$	$e=0.2$ $y=0.4$	$e=0.75$ $y=0.4$	$e=0.0$ $y=0.4$	$e=0.0$ $y=0.4$
SP	$e=0.0$ $y=0.00$	$e=0.2$ $y=0.0$	$e=0.75$ $y=0.0$	$e=0.0$ $y=0.0$	$e=0.0$ $y=0.00$
LP	$e=0.0$ $y=0.00$	$e=0.2$ $y=0.0$	$e=0.75$ $y=0.0$	$e=0.0$ $y=0.0$	$e=0.0$ $y=0.00$

ขบวนการ Inference

เป็นการนำค่าความเป็นสมาชิกของ subset ต่างๆ ใน membership function ของทุก input มาเปรียบเทียบกับกฎที่กำหนดขึ้น แล้วนำค่าความเป็นสมาชิกซึ่งเป็นผลลัพธ์จากกฎทุกกฎมาทำการ defuzzification เพื่อหาค่าจำนวน crisp ของ output.

ในที่นี้เราใช้วิธีที่เรียกว่า ROOT-SUM-SQUARE (RSS)

ขั้นตอนการหาค่าความเป็นสมาชิก

หาค่าความเป็นสมาชิก (degree of membership) ของเอาท์พุทในแต่ละ sub-domain ด้วยวิธี Root-sum-square โดยคำนวณจาก $x_i = \sqrt{\sum_{i=1}^N (\min(A_i, B_i))^2}$ ของ กฎข้อที่เอาท์พุทอยู่ใน sub-domain เดียวกัน A_i และ B_i คือค่าความเป็นสมาชิก (degree of membership) ของ A and B ตามกฎที่ตั้งขึ้น ตามตารางที่ 3.2 ซึ่งเป็นค่าที่มาจากค่าความเป็นสมาชิกใน sub-domain ของแต่ละตัว

การคำนวณหาจำนวน crisp ของ output ใช้วิธีหา weighted average ดังสมการ:

$$X = \frac{\sum_{i=1}^N x_i \times S_i}{\sum_{i=1}^N x_i}$$

โดย $x_i = \sqrt{\sum_{i=1}^N (\min(A_i, B_i))^2}$ และ S_i คือค่า singleton ของแต่ละ fuzzy sub-domains ของ output.

ตารางที่ 3.3 แสดงค่าความเป็นสมาชิก

e(t) y=de(t)/dt	LN	SN	Z	SP	P
LN	x=min(e,y)=0.0	x=min(e,y)=0.0	x=min(e,y)=0.0	x=min(e,y)=0.0	x=min(e,y)=0.0
SN	x=min(e,y)=0.0	x=min(e,y)=0.2	x=min(e,y)=0.25	x=min(e,y)=0.0	x=min(e,y)=0.0
Z	x=min(e,y)=0.0	x=min(e,y)=0.2	x=min(e,y)=0.4	x=min(e,y)=0.0	x=min(e,y)=0.0
SP	x=min(e,y)=0.0	x=min(e,y)=0.0	x=min(e,y)=0.0	x=min(e,y)=0.0	x=min(e,y)=0.0
LP	x=min(e,y)=0.0	x=min(e,y)=0.0	x=min(e,y)=0.0	x=min(e,y)=0.0	x=min(e,y)=0.0

x=fuzzy output

จากสมการ, จะสามารถคำนวณค่า output ด้วยวิธีหา weighted average ได้ดังนี้

$$x = \frac{33 * 0.377 + 0 * 0.47}{0.377 + 0.47} = 14.688$$

การเขียนโปรแกรม

● Fuzzification

ส่วนของ error $e(t) = (\text{Set point} - Ts)$ ในส่วนนี้จะยกมาเฉพาะ Small Positive (SP) และ Large Positive (LP)

```

/*Member Error_SPositive {0 70 140}*/
float Error_SPositive (int _CRISP)
{
    if (_CRISP <= 0) return(0);
    else
    {
        if (_CRISP <= 70) return(_CRISP);
        else
        {
            if (_CRISP <= 140) return(140-_CRISP);
            else return(0);
        }
    }
}

```

โปรแกรมข้างต้นเป็นส่วนหนึ่งของ error $e(t)$ Small Positive (SP) ซึ่งเมื่อรับค่า CRISP input เข้ามาก็จะพิจารณาตามเงื่อนไขว่า

ถ้า CRISP <= 0	ให้ return ค่า	0	ไปที่ Error_Spositive
ถ้า CRISP <= 70	ให้ return ค่า	CRISP	ไปที่ Error_Spositive
ถ้า CRISP <= 140	ให้ return ค่า	140-CRISP	ไปที่ Error_Spositive
ถ้า CRISP > 140	ให้ return ค่า	0	ไปที่ Error_Spositive

```

/*Member Error_LPositive {70 140 210 }*/
float Error_LPositive (int _CRISP)
{
    if (_CRISP <= 70) return(0);
    else
    {
        if (_CRISP <= 140) return(_CRISP-70);
        else return(70); //test
    }
}

```

โปรแกรมข้างต้นเป็นส่วนของ error e(t) Large Positive (LP) ซึ่งเมื่อรับค่า CRISP input เข้ามา ก็จะพิจารณาตามเงื่อนไขว่า

ถ้า CRISP \leq 70 ให้ return ค่า 0 ไปที่ Error_LPositive

ถ้า CRISP \leq 140 ให้ return ค่า CRISP-70 ไปที่ Error_Lpositive

ถ้า CRISP > 140 ให้ return ค่า 70 ไปที่ Error_LPositive

ส่วนของ error-rate $y(t)=de(t)/dt$. ในส่วนนี้จะยกมาเฉพาะ Large Negative (LN), Small Negative(SN)

```

/*Fuzzy set for linguistic delta error Min -3 Max 3*/
/*Member Delta_Error_LNegative {-3 -2 -1 0 }*/
float Delta_Error_LNegative (int _CRISP)
{
    if (_CRISP <= -2) return(70);
    else
    {
        if (_CRISP <= -1) return((-_CRISP*70)-70);
        else return(0);
    }
}

```

โปรแกรมข้างต้นเป็นส่วนของ error-rate $y(t)$ Large Negative (LN) ซึ่งเมื่อรับค่า CRISP input เข้ามาก็จะพิจารณาตามเงื่อนไขว่า

ถ้า CRISP ≤ -2 ให้ return ค่า 70 ไปที่ Delta_Error_LNegative

ถ้า CRISP ≤ -1 ให้ return ค่า $(CRISP*70)-70$ ไปที่ Delta_Error_LNegative

ถ้า CRISP > -1 ให้ return ค่า 0 ไปที่ Delta_Error_LNegative

```

/*Member Delta_Error_SNegative {0 -2 -1 0 0 }*/
float Delta_Error_SNegative (int _CRISP)
{
    if (_CRISP <= -2) return(0);
    else
    {
        if (_CRISP <= -1) return(140-(-_CRISP*70));
        else
        {
            if (_CRISP <= 0) return(-_CRISP*70);
            else return(0);
        }
    }
}

```

โปรแกรมข้างต้นเป็นส่วนของ error-rate $y(t)$ Small Negative(SN) ซึ่งเมื่อรับค่า CRISP input เข้ามาก็จะพิจารณาตามเงื่อนไขว่า

ถ้า CRISP <= -2	ให้ return	ค่า 0	ไปที่ Delta_Error_SNegative
ถ้า CRISP <= -1	ให้ return	ค่า 140(CRISP*70)	ไปที่ Delta_Error_Snegative
ถ้า CRISP <= 0	ให้ return	ค่า (-CRISP*70)	ไปที่ Delta_Error_SNegative
ถ้า CRISP > 0	ให้ return	ค่า 0	ไปที่ Delta_Error_SNegative

- ส่วนการสร้าง rule matrix

สามารถสร้างได้เป็น matrix แบบ 5x5 มีอยู่ทั้ง 25 กรณี ซึ่งกำหนดมาจากความสัมพันธ์ของ input ทั้งสอง เช่น

- ถ้าอุณหภูมิต่ำกว่าค่าที่ตั้งไว้และอัตราการเปลี่ยนแปลงของอุณหภูมิต่ำมาก(LN and LN) ให้ output เป็นการเปลี่ยนแปลงที่เพิ่มขึ้นอย่างมาก (LP)
- ถ้าอุณหภูมิเท่ากับค่าที่ตั้งไว้และไม่มีอัตราการเปลี่ยนแปลงของอุณหภูมิ (Z and Z) ให้ output ไม่ต้องมีการเปลี่ยนแปลง (Z)
- ถ้าอุณหภูมิสูงกว่าค่าที่ตั้งไว้และอัตราการเปลี่ยนแปลงของอุณหภูมิสูงมาก (LP and LP) ให้ output เป็นการเปลี่ยนแปลงที่ลดลงอย่างมาก (LN)

ซึ่งสามารถกำหนดความสัมพันธ์ได้ดังตารางที่ 3.1

จากการที่เรามี input 2 ตัว ซึ่งแต่ละตัวมี 5 sub-domains ทำให้มี input เกิดขึ้นได้ 25 กรณี ดังนั้นในการกำหนดค่าความเป็นสมาชิกของแต่ละ sub-domains ของ output จะต้องนำ input ทั้งสองมาเปรียบเทียบกันแล้วเลือกค่าน้อยกว่าเพื่อนำไปคำนวณค่าความเป็นสมาชิกต่อไป การเปรียบเทียบเลือกค่าน้อยกว่ามีโปรแกรมดังนี้ซึ่งจะไม่ยกมาทั้ง 25 ตัว

```

/*#define fuzzy_and(a,b) ((a)<(b) ?(a) : (b)) /* if a<b then return a else return b */
float fuzzy_and(float a,float b)
{
    if(a>b)
        return b;
    else
        return a;
}

```

```

float Defuzz(void)
/* if Error is LNegative and Delta_Error is LNegative Then output is LPositive*/
R1 = fuzzy_and(Error_LNegative(Error),Delta_Error_LNegative(DeltaError));
.
/* if Error is LPositive and Delta_Error is LPositive Then output is LNegative*/
R25 = fuzzy_and(Error_LPositive(Error),Delta_Error_LPositive(DeltaError));

```

- ส่วน Inferencing

เป็นส่วนที่แสดงค่าความเป็นสมาชิกของแต่ละ sub-domains ของ output โดยนำค่าที่เกิดจาก input ทั้งสอง มาจัดความสัมพันธ์กันว่าจะให้ค่า output ออกมาอยู่ใน sub-domains ใดของ output ดังตาราง 3.4 จากนั้นนำค่า input ทั้งสองมาเปรียบเทียบกันแล้วเลือกค่าน้อยกว่ามากำลังสองแล้วถอดราก จะได้ค่าความเป็นสมาชิกของแต่ละ sub-domains ของ output ออกมา แต่จากโปรแกรมจะเห็นว่ามีการหารด้วย 70 เนื่องมาจากการกำหนดค่าความเป็นสมาชิกของ input สูงสุดเท่ากับ 70 เพื่อความสะดวกในการหาสมการเส้นตรงและการประกาศตัวแปร

$$\text{Output_LPositive} = \sqrt{\left(\frac{R1 \cdot R1}{70 \cdot 70}\right) + \left(\frac{R2 \cdot R2}{70 \cdot 70}\right) + \left(\frac{R3 \cdot R3}{70 \cdot 70}\right) + \left(\frac{R4 \cdot R4}{70 \cdot 70}\right) + \left(\frac{R5 \cdot R5}{70 \cdot 70}\right) + \left(\frac{R6 \cdot R6}{70 \cdot 70}\right) + \left(\frac{R11 \cdot R11}{70 \cdot 70}\right) + \left(\frac{R16 \cdot R16}{70 \cdot 70}\right)};$$

$$\text{Output_SPositive} = \sqrt{\left(\frac{R7 \cdot R7}{70 \cdot 70}\right) + \left(\frac{R8 \cdot R8}{70 \cdot 70}\right) + \left(\frac{R9 \cdot R9}{70 \cdot 70}\right) + \left(\frac{R12 \cdot R12}{70 \cdot 70}\right)};$$

$$\text{Output_Normal} = \sqrt{\left(\frac{R13 \cdot R13}{70 \cdot 70}\right)};$$

$$\text{Output_SNegative} = \sqrt{\left(\frac{R17 \cdot R17}{70 \cdot 70}\right) + \left(\frac{R18 \cdot R18}{70 \cdot 70}\right) + \left(\frac{R19 \cdot R19}{70 \cdot 70}\right) + \left(\frac{R14 \cdot R14}{70 \cdot 70}\right)};$$

$$\text{Output_LNegative} = \sqrt{\left(\frac{R21 \cdot R21}{70 \cdot 70}\right) + \left(\frac{R22 \cdot R22}{70 \cdot 70}\right) + \left(\frac{R23 \cdot R23}{70 \cdot 70}\right) + \left(\frac{R24 \cdot R24}{70 \cdot 70}\right) + \left(\frac{R25 \cdot R25}{70 \cdot 70}\right) + \left(\frac{R10 \cdot R10}{70 \cdot 70}\right) + \left(\frac{R15 \cdot R15}{70 \cdot 70}\right) + \left(\frac{R20 \cdot R20}{70 \cdot 70}\right)};$$

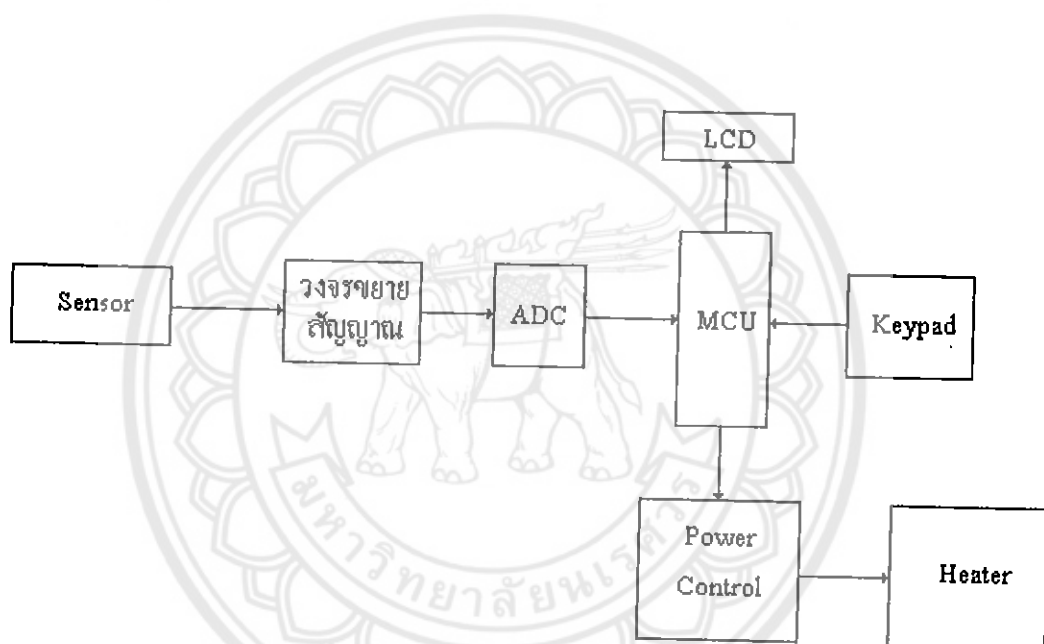
- ส่วน Defuzzification

เมื่อได้ค่าความเป็นสมาชิกของแต่ละ sub-domains ของ output แล้ว ก็จะนำมาคำนวณหาค่า output ของ fuzzy ด้วยวิธี weight average โดยนำค่าความเป็นสมาชิกมาคูณกับ singleton ของแต่ละ sub-domains แล้วนำมารวมกันจากนั้นก็หารด้วยผลรวมของค่าความเป็นสมาชิกทั้งหมด ซึ่งจะได้อค่า fuzzy output ออกมาเป็นเปอร์เซ็นต์เพื่อนำไปคำนวณมุมที่จะต้องเพิ่มหรือลดในการทริกของ SCR ต่อไป

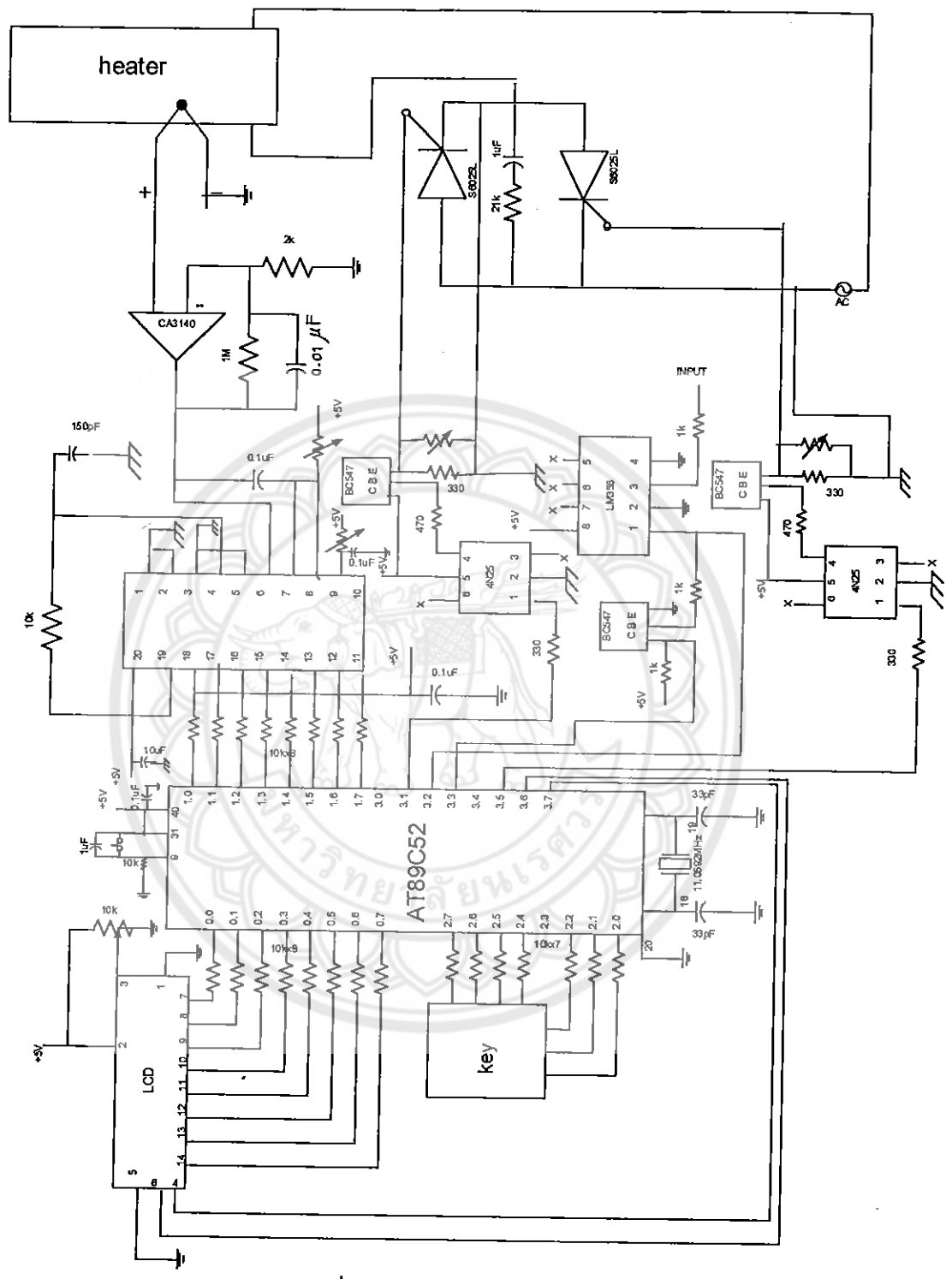
$$\begin{aligned} & \text{return} \left(\frac{\left(\text{Output_LPositive} \cdot \text{LPositive_Centre} \right) + \left(\text{Output_SPositive} \cdot \text{SPositive_Centre} \right) + \right. \\ & \left. \left(\text{Output_Normal} \cdot \text{Normal_Centre} \right) + \left(\text{Output_SNegative} \cdot \text{SNegative_Centre} \right) + \left(\text{Output_LNegative} \cdot \right. \right. \\ & \left. \left. \text{LNegative_Centre} \right) \right) / \\ & \left(\text{Output_LPositive} + \text{Output_SPositive} + \text{Output_Normal} + \text{Output_SNegative} + \text{Output_LNegative} \right); \end{aligned}$$

4. เริ่มต้นการทำงาน

เมื่อเทอร์โมคัปเปิลวัดอุณหภูมิเข้ามาจะเปลี่ยนค่าจากอุณหภูมิเป็นแรงดันไฟฟ้า แต่แรงดันที่ได้มีค่าน้อยมากจึงต้องใช้โอปแอมป์ขยายแรงดันเพื่อให้สามารถส่งไปที่วงจรแปลงสัญญาณอนาลอกเป็นดิจิตอลได้ เมื่อแปลงสัญญาณอนาลอกเป็นดิจิตอลแล้วก็จะส่งไปให้กับ Microprocessor โดยเราสามารถตั้งค่าอุณหภูมิที่ต้องการได้โดยกดที่ KEYPAD จากนั้นจะนำค่าอุณหภูมิที่วัดได้มาเปรียบเทียบกับค่าอุณหภูมิที่ตั้งไว้มาหาค่า Error เพื่อส่งไปให้ Fuzzy Logic ประมวลผลในการควบคุมพลังงานที่จ่ายให้กับเตาอบไฟฟ้า โดยจะควบคุมพลังงานจากการกำหนดมุมทริกให้กับ SCR จ่ายพลังงานออกมาต่าง ๆ กัน จนได้อุณหภูมิที่ต้องการ



รูปที่ 3.8 Block Diagram ของระบบ



รูปที่ 3.15 รูปวงจรรวมของระบบ

บทที่ 4

ผลการออกแบบระบบการควบคุมอุณหภูมิของเตาอบ โดยใช้ Fuzzy logic

ผลการดำเนินโครงการ

หลังจากที่ได้ทำโครงการเกี่ยวกับการควบคุมอุณหภูมิในเตาอบโดยใช้ Fuzzy logic แล้ว ได้ทำการทดลองได้ผลดังนี้

- สามารถควบคุมอุณหภูมิในเตาอบได้ตามที่ set ไว้
- เตาอบสามารถวัดอุณหภูมิได้ถึง 250 องศาเซลเซียส
- มีหน้าจอ LCD แสดงผลของอุณหภูมิ

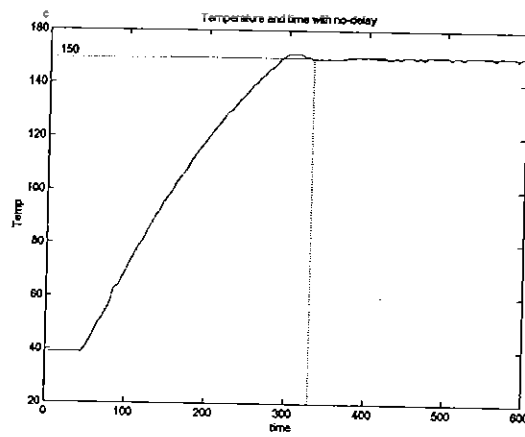
วิธีการใช้งานและการทดลองระบบควบคุม

เมื่อเปิดเครื่องจะพบจอ LCD ให้สำหรับกรอกค่าอุณหภูมิที่เราต้องการและจะมีค่าอุณหภูมิที่ได้รับจากเทอร์โมคัปเปิลแสดงอยู่

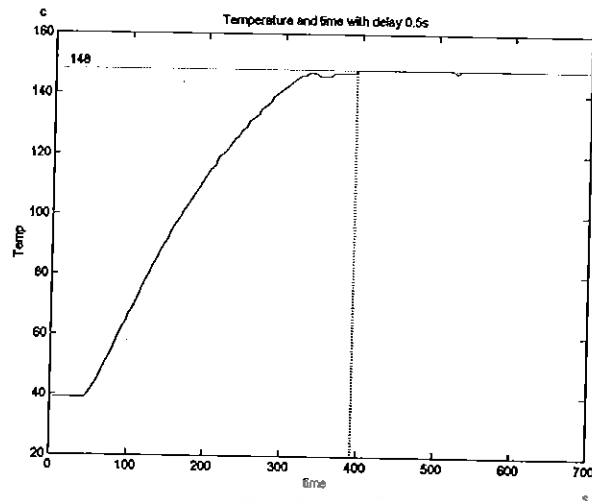
จากการทดสอบ

- ในการใช้เทอร์โมคัปเปิลวัดขณะยังไม่เปิดเครื่องจะพบว่าอุณหภูมิเริ่มที่ 39 องศา
- ทำการ set ค่าอุณหภูมิที่ต้องการ
- รอผลจากการทำงานของระบบ
- แล้วทำการทดสอบซ้ำ

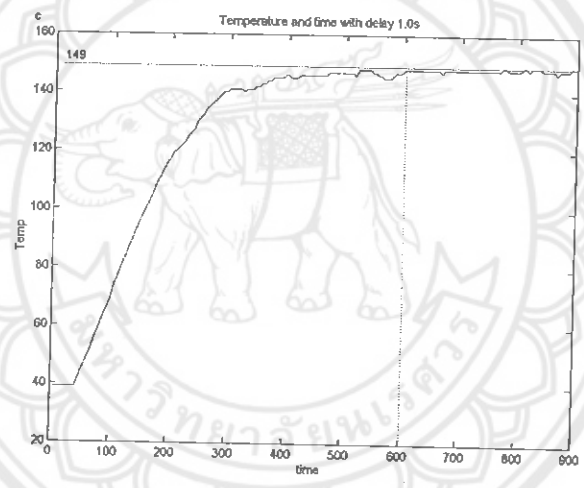
กราฟจากการทดลองจากการ delay ที่ค่าต่างๆ



รูปที่ 4.1 กราฟการที่ไม่มี delay



รูปที่ 4.2 รูปการ delay ที่ 0.5s



รูปที่ 4.3 รูปการ delay ที่ 1s

สรุปผลจากกราฟ

จากกราฟจะเห็นได้ว่าถ้าโปรแกรมไม่มีการ delay ดังรูปที่ (4.1) จะทำให้อุณหภูมิเข้าสู่ค่าอุณหภูมิที่ตั้งเอาไว้เร็วที่สุด รองลงมาคือโปรแกรมที่มีการ delay ที่ 0.5 s ดังรูปที่ (4.2) และจะมีการเข้าสู่ค่าที่ช้าที่สุดเมื่อมีการ delay 1 s ดังรูปที่ (4.3)

สรุป

โครงการงานชิ้นนี้ ได้มีการสร้างเป็นลักษณะแบบจำลองการทำงานของระบบควบคุมอุณหภูมิเตาอบในโรงงานอุตสาหกรรมโดยใช้ Fuzzy logic ซึ่งระบบควบคุมนี้สามารถใช้กับเตาอบที่ทำการฆ่าเชื้อจุลินทรีย์ต่างๆ ได้ โดยมีตัวไมโครคอนโทรลเลอร์ MCS - 51 มาเป็นตัวเชื่อมระหว่างฮาร์ดแวร์และซอฟต์แวร์ที่ใช้ในการควบคุม

บทที่ 5

สรุปผลการปฏิบัติโครงการ

5.1 สรุปผลการปฏิบัติงาน

สามารถควบคุมอุณหภูมิได้ตามที่เราต้องการรวมถึงการได้เรียนรู้และใช้งาน ทฤษฎี Fuzzy logic ซึ่งสามารถนำมาประยุกต์ใช้งานในโรงงานอุตสาหกรรมและศูนย์วิจัยต่างได้ และได้เรียนรู้ทฤษฎีและการใช้งาน ออปแอมป์ SCR อุปกรณ์ A/D และ D/A

5.2 ประโยชน์ที่ได้รับจากการทำงาน

1. สามารถควบคุมอุณหภูมิในเตาอบโดยใช้ ทฤษฎี Fuzzy logic ได้ตามต้องการ
2. ได้เรียนรู้และเข้าใจในการนำ ทฤษฎี Fuzzy logic มาประยุกต์มาใช้งาน
3. ได้เรียนรู้การทำงานและใช้งาน ไมโครชิป MCS51
4. ได้เรียนรู้การทำงานและใช้งาน ออปแอมป์
5. ได้เรียนรู้การทำงานและใช้งาน SCR
6. ได้เรียนรู้การทำงานใช้งาน อุปกรณ์ A/D และ D/A
7. รู้จักการแก้ปัญหาในการทำงาน
8. รู้จักการทำงานร่วมกันเป็นกลุ่ม

5.3 แนวทางพัฒนาระบบต่อไปในอนาคต

สามารถนำโครงการนี้ไปประยุกต์ใช้งานในระบบควบคุมอุณหภูมิอื่นๆได้เช่น การควบคุมอุณหภูมิทางการเกษตรกรรม ทางการแพทย์ และอื่นๆ ที่ต้องการควบคุมอุณหภูมิที่คลาดเคลื่อนเล็กน้อย และยังสามารถนำหลักการของ Fuzzy logic ไปประยุกต์ใช้งานในการควบคุมระบบการทำงานอื่นๆได้

เอกสารอ้างอิง

- [1] "Europe Gets into Fuzzy Logic" (Electronics Engineering Times, Nov. 11, 1991).
- [2] "Fuzzy Sets and Applications: Selected Papers by L.A. Zadeh", ed. R.R. Yager et al. (John Wiley, New York, 1987).
- [3] "U.S. Loses Focus on Fuzzy Logic" (Machine Design, June 21, 1990).
- [4] "Why the Japanese are Going in for this 'Fuzzy Logic'" by Emily T. Smith (Business Week, Feb. 20, 1993, pp. 39).
- [5] "Fuzzy Logic-An Introduction" by Steven D.Kachler.
- [6] "A practical implementation of a fuzzy logic controller with Motorola 68HC11" by Loan Susnea, University "DUNAREA DE JOS" of Galati, Romania.
- [7] ผศ.ธีรวัฒน์ ประกอบผล. การพัฒนาไมโครคอนโทรลเลอร์ด้วยภาษาซี . พิมพ์ครั้งที่ 4. กรุงเทพฯ : สำนักพิมพ์ ส.ส.ท.
- [8] George J. Klir, Ute H. St. Clair, Mo yuan. Fuzzy Set Theory Foundations and Applications. International Edition. Prentice – Hall International, Inc
- [9] วรพจน์ กรแก้ววัฒนกุล, ชัยวัฒน์ ลิ้มพรจิตรวิไล. MCS-51 Flash Microcontroller ฉบับ AT89c5x ของ ATMEL. พิมพ์ครั้งที่ 1. กรุงเทพฯ : บริษัท อินโนเวทีฟ เอ็กเพอริเมนต์ จำกัด.
- [10] J.-S.R.Jang, C.-T.Sun, E. Mizutani. Neuro-Fuzzy and Soft computing. Prentic, Hall of India Private Limited.
- [11] มงคล ทองสงคราม. อิเล็กทรอนิกส์กำลัง. พิมพ์ครั้งที่ 4. กรุงเทพฯ : ห้างหุ้นส่วนจำกัด วี. เจ. พรินติ้ง.
- [12] วิโรจน์ อัสวรังสี, ชัชวาล เต็มฤทธิวงศ์, กรชูลิ ใช้สถิตย์. การใช้ออปแอมป์และลิเนียร์ไอซี. กรุงเทพฯ : บริษัท ซีอีคยูเคชั่น จำกัด.



```

#include <reg51.h>
#include <absacc.h>
#include <ctype.h>
#include <intrins.h>
#include <math.h>
#include <stdarg.h>
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

int a,b;
bit chk_up_lo;
bit stop_t0=0;
#define value_TL0 0xfe
/*#define fuzzy_one 0xff
#define fuzzy_zero 0*00
#define fuzzy_or(a,b) ((a)>(b) ?(a) : (b)) /* if a>b then return a else return b */
/*#define fuzzy_and(a,b) ((a)<(b) ?(a) : (b)) /* if a<b then return a else return b */
/*#define fuzzy_not(a) (fuzzy_one+fuzzy_zero-a)
#define setpoint=P2 */
sbit P3_0=P3^1;
sbit P3_1=P3^5;
sbit sw_clear=P3^4;

unsigned int tick;

int e1=0,chk_e2_buf1=0,t,t_old=0;

unsigned int Temp,temp_set;
int OldError=0,DeltaError, Error;
int Output,Output_Buf=0;
char LNegative_Centre=-100,SNegative_Centre=-
33,Normal_Centre=0,SPositive_Centre=33,LPositive_Centre=100;
float idata
R1,R2,R3,R4,R5,R6,R7,R8,R9,R10,R11,R12,R13,R14,R15,R16,R17,R18,R19,R20,R21,R22,R23,R24,R25;
float idata Output_LPositive,Output_SPositive,Output_Normal,Output_SNegative,Output_LNegative;

```

```

//*****
sfr LCDDA = 0x80;    // V-S82 P0 = lcd data
sbit LCDRS = P3^7;  // V-S82 lcd rs
sbit LCDRW = P3^5;  // V-S82 lcd rw
sbit LCDEN = P3^6;  // V-S82 lcd en

bit keypress;

unsigned char KEYCODE[16]={
    0x7d,0xee,0xed,0xeb,
    0xde,0xdd,0xdb,0xbe,
    0xbd,0xbb,0xe7,0xd7,
    0xb7,0x77,0x7e,0x7b);

unsigned char LCDBUF[16+1]; // lcd buffer 16x2
unsigned char DISBUF[3];
unsigned char DISBUF1[3];
void dmsec (unsigned int count) { // mSec Delay 11.0592 Mhz
    unsigned int i; // Keil v5.2/v5.5
    while (count) {
        i = 6; while (i>0) i--;
        count--;
    }
}

void dmsec2 (unsigned int count2) { // mSec Delay 11.0592 Mhz
    unsigned int i2; // Keil v5.2/v5.5
    while (count2) {
        i2 = 115; while (i2>0) i2--;
        count2--;
    }
}

```

```

/*void time(unsigned int cir)
{
    unsigned int x;
    for(x=0;x<cir;x++)
    {
        TH0=0xff;
        TLO=0xdb;
        TF0=0;
        TR0=1;
        while(TF0==0);
        TR0=0;
    }
}*/
/*Fuzzy set for linguistic error Min -90 Max 90*/
/*Member Error_LNegative {-90 -90 -45 0 }*/
float Error_LNegative (int _CRISP)
{
    if (_CRISP <= -140) return(70); //test
    else
    {
        if (_CRISP <= -70) return(-_CRISP-70);
        else return(0);
    }
}

/*Member Error_SNegative {-45 -20 0 0 }*/
float Error_SNegative (int _CRISP)
{
    if (_CRISP <= -140) return(0);
    else
    {

```



```

        if (_CRISP <= -70) return(140+_CRISP);
        else
        {
            if (_CRISP <= 0) return((-_CRISP));
            else return(0);
        }
    }
}

```

/*Member Error_Normal {-20 0 20 0}*/

float Error_Normal (int _CRISP)

```

{
    if (_CRISP <= -70) return(0);
    else
    {
        if (_CRISP <= 0) return(70+_CRISP);
        else
        {
            if (_CRISP <= 70) return(70-_CRISP);
            else return(0);
        }
    }
}

```

/*Member Error_SPositive {0 20 45 0}*/

float Error_SPositive (int _CRISP)

```

{
    if (_CRISP <= 0) return(0);
    else
    {
        if (_CRISP <= 70) return(_CRISP);
        else
        {
            if (_CRISP <= 140) return(140-_CRISP);

```

```

else return(0);
    }
}

/*Member Error_LPositive {0 20 90 90 }*/
float Error_LPositive (int _CRISP)
{
    if (_CRISP <= 70) return(0);
    else
    {
        if (_CRISP <= 140) return(_CRISP-70);
        else return(70); //test
    }
}

/*Fuzzy set for linguistic delta error Min -9 Max 9*/

/*Member Delta_Error_LNegative {-9 -9 -4.5 0 }*/
float Delta_Error_LNegative (int _CRISP)
{
    if (_CRISP <= -2) return(70); //test
    else
    {
        if (_CRISP <= -1) return((-_CRISP*70)-70);
        else return(0);
    }
}

/*Member Delta_Error_SNegative {-4.5 -2 0 0 }*/
float Delta_Error_SNegative (int _CRISP)
{
    if (_CRISP <= -2) return(0);
    else

```

```

    {
        if (_CRISP <= -1) return(140-(-_CRISP*70));
        else
        {
            if (_CRISP <= 0) return(-_CRISP*70);
            else return(0);
        }
    }
}

```

/*Member Delta_Error_Normal {-2 0 2 0 }*/

float Delta_Error_Normal (int _CRISP)

```

{
    if (_CRISP <= -1) return(0);
    else
    {
        if (_CRISP <= 0) return(70+(_CRISP*70));
        else
        {
            if (_CRISP <= 1) return(70+(-_CRISP*70));
            else return(0);
        }
    }
}

```

/*Member Delta_Error_SPositive {0 2 4.5 0 }*/

float Delta_Error_SPositive (int _CRISP)

```

{
    if (_CRISP <= 0) return(0);
    else
    {
        if (_CRISP <= 1) return(_CRISP*70);
        else
        {
            if (_CRISP <= 2) return(140-(_CRISP*70));

```

```

        else return(0);
    }
}

/*Member Delta_Error_LPositive {0 2 9 9}*/
float Delta_Error_LPositive (int _CRISP)
{
    if (_CRISP <= 1) return(0);
    else
    {
        if (_CRISP <= 2) return((_CRISP*70)-70);
        else return(70);//test
    }
}

#define fuzzy_one 0*ff
#define fuzzy_zero 0*00
#define fuzzy_or(a,b) ((a)>(b) ?(a) : (b)) /* if a>b then return a else return b */
#define fuzzy_and(a,b) ((a)<(b) ?(a) : (b)) /* if a<b then return a else return b */
#define fuzzy_not(a) (fuzzy_one+fuzzy_zero-a)
#define setpoint=P2 */
/* Fuzzy Calculation */

float fuzzy_and(float a,float b)
{
    if(a>b)
        return b;
    else
        return a;
}

float Defuzz(void)
{

```

```
/* if Error is LNegative and Delta_Error is LNegative Then output is LPositive*/
R1 = fuzzy_and(Error_LNegative(Error),Delta_Error_LNegative(DeltaError));

/* if Error is LNegative and Delta_Error is SNegative Then output is LPositive*/
R2 = fuzzy_and(Error_LNegative(Error),Delta_Error_SNegative(DeltaError));

/* if Error is LNegative and Delta_Error is Normal Then output is LPositive*/
R3 = fuzzy_and(Error_LNegative(Error),Delta_Error_Normal(DeltaError));

/* if Error is LNegative and Delta_Error is SPositive Then output is LPositive*/
R4 = fuzzy_and(Error_LNegative(Error),Delta_Error_SPositive(DeltaError));

/* if Error is LNegative and Delta_Error is LPositive Then output is LPositive*/
R5 = fuzzy_and(Error_LNegative(Error),Delta_Error_LPositive(DeltaError));

/* if Error is SNegative and Delta_Error is LNegative Then output is SPositive*/
R6 = fuzzy_and(Error_SNegative(Error),Delta_Error_LNegative(DeltaError));

/* if Error is SNegative and Delta_Error is SNegative Then output is SPositive*/
R7 = fuzzy_and(Error_SNegative(Error),Delta_Error_SNegative(DeltaError));

/* if Error is SNegative and Delta_Error is Normal Then output is SPositive*/
R8 = fuzzy_and(Error_SNegative(Error),Delta_Error_Normal(DeltaError));

/* if Error is SNegative and Delta_Error is SPositive Then output is SPositive*/
R9 = fuzzy_and(Error_SNegative(Error),Delta_Error_SPositive(DeltaError));

/* if Error is SNegative and Delta_Error is LPositive Then output is SPositive*/
R10 = fuzzy_and(Error_SNegative(Error),Delta_Error_LPositive(DeltaError));

/* if Error is Normal and Delta_Error is SNegative Then output is Normal*/
R11 = fuzzy_and(Error_Normal(Error),Delta_Error_LNegative(DeltaError));

/* if Error is Normal and Delta_Error is LNegative Then output is Normal*/
R12 = fuzzy_and(Error_Normal(Error),Delta_Error_SNegative(DeltaError));
```

/ if Error is Normal and Delta_Error is Normal Then output is Normal*/*
R13 = fuzzy_and(Error_Normal(Error),Delta_Error_Normal(DeltaError));

/ if Error is Normal and Delta_Error is SPositive Then output is Normal*/*
R14 = fuzzy_and(Error_Normal(Error),Delta_Error_SPositive(DeltaError));

/ if Error is Normal and Delta_Error is LPositive Then output is Normal*/*
R15 = fuzzy_and(Error_Normal(Error),Delta_Error_LPositive(DeltaError));

/ if Error is SPositive and Delta_Error is LNegative Then output is SNegative*/*
R16 = fuzzy_and(Error_SPositive(Error),Delta_Error_LNegative(DeltaError));

/ if Error is SPositive and Delta_Error is SNegative Then output is SNegative*/*
R17 = fuzzy_and(Error_SPositive(Error),Delta_Error_SNegative(DeltaError));

/ if Error is SPositive and Delta_Error is Normal Then output is SNegative*/*
R18 = fuzzy_and(Error_SPositive(Error),Delta_Error_Normal(DeltaError));

/ if Error is SPositive and Delta_Error is SPositive Then output is SNegative*/*
R19 = fuzzy_and(Error_SPositive(Error),Delta_Error_SPositive(DeltaError));

/ if Error is SPositive and Delta_Error is LPositive Then output is SNegative*/*
R20 = fuzzy_and(Error_SPositive(Error),Delta_Error_LPositive(DeltaError));

/ if Error is LPositive and Delta_Error is LNegative Then output is LNegative*/*
R21 = fuzzy_and(Error_LPositive(Error),Delta_Error_LNegative(DeltaError));

/ if Error is LPositive and Delta_Error is SNegative Then output is LNegative*/*
R22 = fuzzy_and(Error_LPositive(Error),Delta_Error_SNegative(DeltaError));

/ if Error is LPositive and Delta_Error is Normal Then output is LNegative*/*
R23 = fuzzy_and(Error_LPositive(Error),Delta_Error_Normal(DeltaError));

/ if Error is LPositive and Delta_Error is SPositive Then output is LNegative*/*
R24 = fuzzy_and(Error_LPositive(Error),Delta_Error_SPositive(DeltaError));

```

/* if Error is LPositive and Delta_Error is LPositive Then output is LNegative*/
R25 = fuzzy_and(Error_LPositive(Error),Delta_Error_LPositive(DeltaError));

/* Output set LNegative_Centre = 150'(n delay) SNegative_Centre = 120'(n delay)
   Normal_Centre = 90'(n delay) SPositive_Centre = 60'(n delay)
   LPositive_Centre = 30'(n delay)
*/

/*Output_LPositive = sqrt(((R1*R1)/(70*70))+((R2*R2)/(70*70))+((R3*R3)/(70*70))+((R4*R4)/(70*70))+
((R5*R5)/(70*70)));
Output_SPositive = sqrt(((R6*R6)/(70*70))+((R7*R7)/(70*70))+((R8*R8)/(70*70))+((R9*R9)/(70*70))+
((R10*R10)/(70*70)));
Output_Normal = sqrt(((R11*R11)/(70*70))+((R12*R12)/(70*70))+((R13*R13)/(70*70))+((R14*R14)/
(70*70))+((R15*R15)/(70*70)));
Output_SNegative = sqrt(((R16*R16)/(70*70))+((R17*R17)/(70*70))+((R18*R18)/(70*70))+((R19*R19)/
(70*70))+((R20*R20)/(70*70)));
Output_LNegative = sqrt(((R21*R21)/(70*70))+((R22*R22)/(70*70))+((R23*R23)/(70*70))+((R24*R24)/
(70*70))+((R25*R25)/(70*70)));
*/

Output_LPositive = sqrt(((R1*R1)/(70*70))+((R2*R2)/(70*70))+((R3*R3)/(70*70))+((R4*R4)/(70*70))+
((R5*R5)/(70*70))+((R6*R6)/(70*70))+((R11*R11)/(70*70))+((R16*R16)/(70*70)));
Output_SPositive = sqrt(((R7*R7)/(70*70))+((R8*R8)/(70*70))+((R9*R9)/(70*70))+((R12*R12)/
(70*70)));
Output_Normal = sqrt(((R13*R13)/(70*70)));
Output_SNegative = sqrt(((R17*R17)/(70*70))+((R18*R18)/(70*70))+((R19*R19)/(70*70))+((R14*R14)/
(70*70)));
Output_LNegative = sqrt(((R21*R21)/(70*70))+((R22*R22)/(70*70))+((R23*R23)/(70*70))+((R24*R24)/
(70*70))+((R25*R25)/(70*70))+((R10*R10)/(70*70))+((R15*R15)/(70*70))+((R20*R20)/(70*70)));

/* Centroid */
return(((Output_LPositive*LPositive_Centre)+(Output_SPositive*SPositive_Centre)+(Output_Normal*N
ormal_Centre)
+(Output_SNegative*SNegative_Centre)+(Output_LNegative*LNegative_Centre)) /

```

```

(Output_LPositive+Output_SPositive+Output_Normal+Output_SNegative+Output_LNegative));

}

/*end fuzzy*/

void lcdwi (unsigned char com) { // LCD write instruction
    LCDEN = 0; LCDRS = 0; LCDRW = 0;
    LCDDA = com;
    LCDEN = 0; _nop_ (); _nop_ ();
    LCDEN = 1; _nop_ (); _nop_ ();
    LCDEN = 0;
    dmsec2 (2);
}

void lcdwd (unsigned char dat) { // LCD write data
    LCDEN = 0; LCDRS = 1; LCDRW = 0;
    LCDDA = dat;
    LCDEN = 0; _nop_ (); _nop_ ();
    LCDEN = 1; _nop_ (); _nop_ ();
    LCDEN = 0;
    dmsec2 (2);
}

void lcdset (void) { // LCD start process
    LCDEN = 0;
    LCDRS = 0;
    LCDRW = 0;
    dmsec2 (100);
    lcdwi (0x38); // function set
    lcdwi (0x0c); // display on
}

/*(void lcdclear (void) { // clear LCD & LCDBUF
    unsigned char i;
    lcdwi (0x01);

```



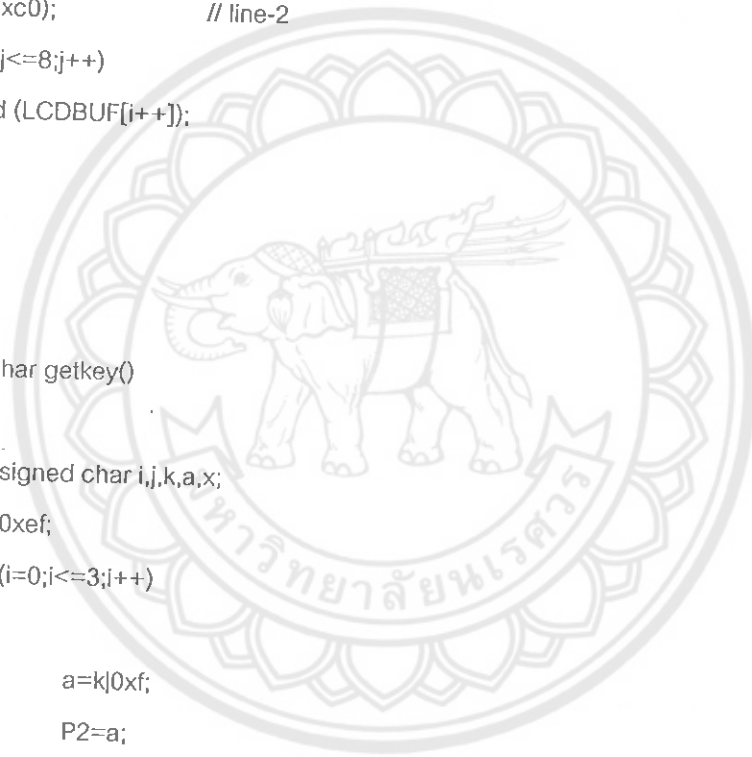
```

    for (i=0;i<=31;i++)
        LCDBUF[i] = '';
    }*/

void lcdout (void) {          // load LCDBUF to LCD
    unsigned char i,j;
    i = 0;
    lcdwi (0x80);             // line-1
    for (j=1;j<=8;j++)
        lcdwd (LCDBUF[i++]);
    lcdwi (0xc0);            // line-2
    for (j=1;j<=8;j++)
        lcdwd (LCDBUF[i++]);
}

unsigned char getkey()
{
    unsigned char i,j,k,a,x;
    k=0xef;
    for(i=0;i<=3;i++)
    {
        a=k|0xf;
        P2=a;
        x=P2;
        x=x&0xf;
        dmsec2(1);
        if(x!=0xf)
        {
            if(~keypress)
            {
                keypress=1;
                a=k&0xf0;
                x=x|a;
            }
        }
    }
}

```



```

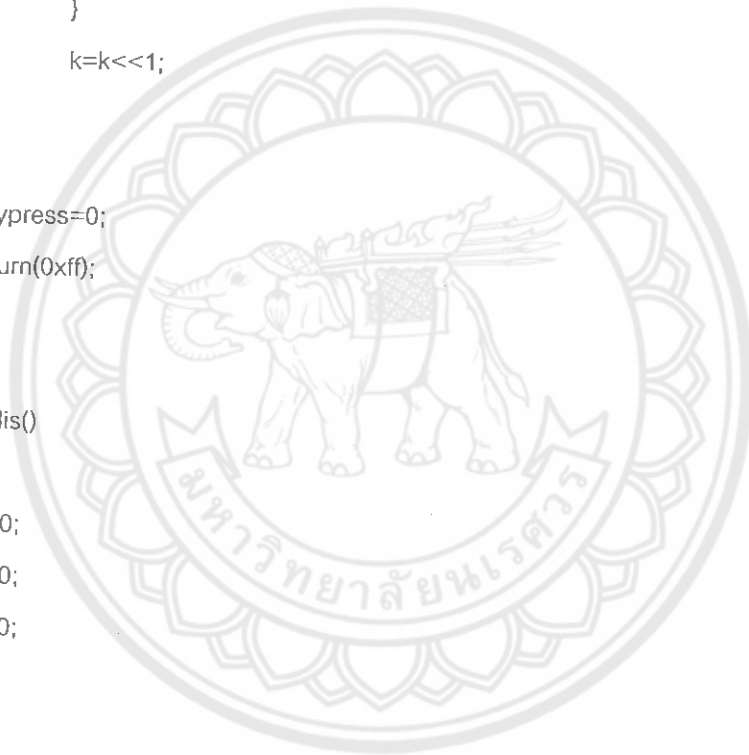
        for(j=0;j<=15;j++)
        if(x==KEYCODE[j])
        {
            return(j);
        }
        return(0xff);
    }
    return(0xff);
}
k=k<<1;
}
keypress=0;
return(0xff);
}

void clear_dis()
{
    DISBUF[0]=0;
    DISBUF[1]=0;
    DISBUF[2]=0;
}

void MOV_DIGI()
{
    DISBUF[0]=DISBUF[1];
    DISBUF[1]=DISBUF[2];
}

void SHOW_NUM(unsigned char addr,int Temp)
{
    char buf;

```



```

int Temp_signed;
Temp_signed=Temp;
Temp=abs(Temp);
DISBUF1[0]=Temp/100;
buf=Temp%100;
DISBUF1[1]=buf/10;
DISBUF1[2]=buf%10;
lcdwi(addr);
if(Temp_signed<0)
    lcdwd('-');
else
{
    if(DISBUF1[0]>0)
        lcdwd(DISBUF1[0]+0x30);
}
lcdwd(DISBUF1[1]+0x30);
lcdwd(DISBUF1[2]+0x30);
}

```

```
void INPUT_NUM(void)
```

```

{
    unsigned char x;
    unsigned int Num;
    Num=0;
    x=0;
    // clear_dis();
    lcdwi(0xc6);

    lcdwi(0x0f);

    while(x!=15)

```

```

    {
        x=getkey();

        if(x<10)
        {
            MOV_DIGI();
            DISBUF[2]=x;
            lcdwi(0xc3);
            lcdwd(DISBUF[0]+0x30);
            lcdwd(DISBUF[1]+0x30);
            lcdwd(DISBUF[2]+0x30);
            temp_set=temp_set*10+x; //temp_set
            lcdwi(0xc6);
            lcdwi(0x0f);

            // SHOW_NUM(0xc4,Num) ;
        }
        if(x==14)
        {
            clear_dis();
            temp_set=0;
            lcdwi(0xc3);
            lcdwd(DISBUF[0]+0x30);
            lcdwd(DISBUF[1]+0x30);
            lcdwd(DISBUF[2]+0x30);
            lcdwi(0xc6);
            lcdwi(0x0f);
        }

        /* if(x==10)
        {
            return Num;
        }

```

```
    return 0;*/
}
    lcdwi(0x0c);
}
```

```
void main(void)
```

```
{
```

```
    // unsigned char x;
```

```
    lcdset();
```

```
    strcpy(LCDBUF,"Naresuan Nu.  ");
```

```
    lcdout();
```

```
    dmsec2(2000);
```

```
    strcpy(LCDBUF,"t set C");
```

```
    lcdout();
```

```
    lcdwi(0xC6);
```

```
    lcdwd(0xdf);
```

```
    clear_dis();
```

```
//IE=0x81;
```

```
//IT0=0x01;
```

```
sw_clear=1;
```

```
    P3_0=0;
```

```
    P3_1=0;
```

```
    EA=1;
```

```
    EX0=1;
```

```
    EX1=1;
```

```
    IT0=1;
```

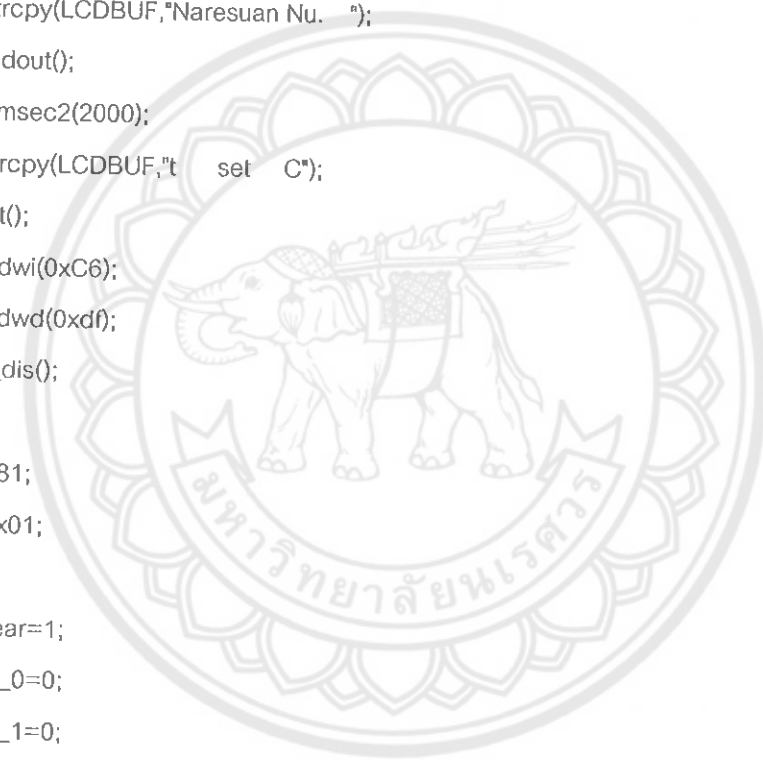
```
    IT1=1;
```

```
    TMOD=0x21;
```

```
    //TH1=0xfd;
```

```
    //ES=1;
```

```
    //TR1=1;
```



```
//P3_6=1;
//P3_7=1;
//TMOD=0x01;
//TH0=0xFF;
//TL0=value_TL0;
ET0=1;
tick=0;
//TR0=1;
```

```
while(1)
{
```

```
/* x=getkey();
if(x!=0xff)
{
if(x==14)
{
INPUT_NUM();
}
}
}*/
```

```
if(sw_clear==0)
{
dmsec(20);
if(sw_clear==0)
{
INPUT_NUM();
}
}
Temp=P1;
Temp=Temp+39;
//Temp=50;
```

```
//temp_set=100;  
SHOW_NUM(0x82,Temp);
```

```
Error = temp_set-Temp;  
DeltaError = Error-OldError;  
OldError = Error;  
Output=Defuzz();  
//SHOW_NUM(0x84,Output);  
//P1=Output;
```

```
//if(Output_Buf!=Output)  
//(  
    Output_Buf=Output;  
    t_old=t_old-(Output);  
//)  
if(t_old>370)  
{  
    t_old=370;  
}  
if(t_old<=202)  
{  
    t_old=202;  
}  
t=400-t_old;  
//SHOW_NUM(0x84,Output);  
//SHOW_NUM(0xC0,t);
```

```
dmsec2(1000);
```

```
}
```

```
}
```

```
void int0_service() interrupt 0
```

```
{
```

```
    TH0=0xFF;
```

```
    TL0=value_TL0;
```

```
    TR0=1;//start timer0
```

```
    chk_up_lo=0;
```

```
    stop_t0=1;
```

```
    // t=1;
```

```
}
```

```
void int1_service() interrupt 2
```

```
{
```

```
    TH0=0xFF;
```

```
    TL0=value_TL0;
```

```
    TR0=1;//start timer0
```

```
    chk_up_lo=1;
```

```
    stop_t0=1;
```

```
    // t=1;
```

```
}
```

```
void timer0_service() interrupt 1
```

```
{
```

```
    TR0=0;
```

```
    tick++;
```

```
    if(tick==t)
```

```
    {
```

```
        //t=t+1;
```

```
        //if(t==100){t=1;}
```



```
TR0=0;
tick=0;
stop_t0=0;
if(chk_up_lo==0)
{
    P3_0=1;
    dmsec(1);
    P3_0=0;
}
else
{
    P3_1=1;
    dmsec(1);
    P3_1=0;
}
}
if(stop_t0)
{
    TH0=0xFF;
    TL0=value_TL0;
    TR0=1;
}
}
```



ประวัติผู้จัดทำโครงการ

ชื่อ นางสาว จูติรัตน์ พัฒนศรีศักดิ์
ภูมิลำเนา 90 หมู่ 1 ตำบลบ้านบุง อำเภอเมือง จังหวัดพิจิตร
ประวัติการศึกษา

- จบชั้นมัธยมศึกษาตอนปลาย โรงเรียนพิจิตรพิทยาคม
- ปัจจุบันกำลังศึกษาในระดับปริญญาตรี ชั้นปีที่ 4
- สาขาวิชาวิศวกรรมไฟฟ้า คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร

e-mail : joyintell@hotmail.com



ชื่อ นายบัณฑิต ศรีคง
ภูมิลำเนา 185/8 ม. 7 ต. แก่งไสภา อ.วังทอง จังหวัดพิษณุโลก
ประวัติการศึกษา

- จบชั้นมัธยมศึกษาตอนปลาย โรงเรียนพิษณุโลกพิทยาคม
- ปัจจุบันกำลังศึกษาในระดับปริญญาตรี ชั้นปีที่ 4
- สาขาวิชาวิศวกรรมไฟฟ้า คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร

e-mail : bunditsrikong@hotmail.com



ชื่อ นาย สุทธิชาติ สุขัมมา
ภูมิลำเนา 81/3 ม. 3 ตำบล ผาจุก อำเภอเมือง จังหวัด อุตรดิตถ์
ประวัติการศึกษา

- จบชั้นมัธยมศึกษาตอนปลาย โรงเรียนอุตรดิตถ์
- ปัจจุบันกำลังศึกษาในระดับปริญญาตรี ชั้นปีที่ 4
- สาขาวิชาวิศวกรรมไฟฟ้า คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร

e-mail : t_soothamma@hotmail.com

