



การออกแบบตัวควบคุมแบบ พีไอดี-ฟัซซี่ สำหรับระบบควบคุม  
และตรวจสอบขบวนการโดยใช้ไมโครคอมพิวเตอร์

DESIGN OF PID-FUZZY CONTROLLER AND MONITORING  
PROCESS SYSTEM USING MICROCOMPUTER

โดย

นาย ชีรวิทย์ รัตนวิสุทธิ

นาย เอกชัย นิตระเนตร

นาย กิติพงษ์ นพกรวิเศษ

นาย วัชรินทร์ อินทฤทธิ์

วัน เดือน ปี... 1 ก.ค. 2540  
เลขทะเบียน... 037146  
เลขเรียกหนังสือ... T 38239 ๗๒๑ ก.

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชา วิศวกรรมการวัดคุม

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2538

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

037146

# ปริญญาานิพนธ์ ปีการศึกษา 2538

ภาควิชา เทคโนโลยีการวัดคุมทางอุตสาหกรรม  
สาขาวิชา วิศวกรรมการวัดคุมทางอุตสาหกรรม  
คณะวิศวกรรมศาสตร์  
สถาบันเทคโนโลยีพระจอมเกล้า เจ้าคุณทหารลาดกระบัง

เรื่อง DESIGN OF PID-FUZZY CONTROLLER AND  
MONITORING PROCESS SYSTEM USING  
MICROCOMPUTER

ผู้จัดทำ

1. นาย ชีรวุฒิ รัตนวิสุทธิ 36012103
2. นาย เอกชัย นัตระเนตร 36012131
3. นาย กิติพงษ์ นพกรวิเศษ 36013276
4. นาย วัชรินทร์ อินทฤทธิ 36013304

.....อาจารย์ที่ปรึกษา

( อาจารย์ สักกริยา ชิตวงศ์ )

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## หัวข้อปริญญาานิพนธ์

การออกแบบตัวควบคุมแบบ พีไอดี-ฟัซซี่ สำหรับระบบควบคุม และตรวจสอบขบวนการโดยใช้ไมโครคอมพิวเตอร์

## จัดทำโดย

นาย ธีรวุฒิ รัตนวิสุทธิ เลขประจำตัว 36012103  
นาย เอกชัย ฉัตรเนตร เลขประจำตัว 36012131  
นาย กิติพงษ์ นพกรวิเศษ เลขประจำตัว 36013276  
นาย วัชรินทร์ อินทฤทธิ์ เลขประจำตัว 36013304

## ระดับการศึกษา

ปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิศวกรรมการวัดคุมทางอุตสาหกรรม

## ภาควิชา

เทคโนโลยีการวัดคุมทางอุตสาหกรรม

## ปีการศึกษา

2538

## อาจารย์ผู้ควบคุมปริญญาานิพนธ์

อ. สักกรียา ชิตวงศ์

## บทคัดย่อ

โครงการ การออกแบบตัวควบคุมแบบ พีไอดี-ฟัซซี่ สำหรับระบบควบคุมและตรวจสอบขบวนการโดยใช้ไมโครคอมพิวเตอร์ ฉบับนี้เป็นการศึกษาทฤษฎี FUZZY CONTROL เพื่อใช้หาค่าพารามิเตอร์ของเครื่องควบคุมพีไอดี (PID CONTROLLER) โดยนำเอาไมโครคอมพิวเตอร์ที่ใช้งานทั่วไปมาควบคุมกระบวนการให้มีการทำงานตามที่ต้องการ โดยใช้โปรแกรมที่เขียนขึ้นซึ่งสามารถใช้งานได้สะดวก ง่ายต่อการใช้งานและมีความสามารถในการควบคุมสูง พร้อมทั้งมีการแสดงผลพร้อมออกมาในรูปของตัวเลขและแบบกราฟ ซึ่งโครงการนี้สามารถนำไปใช้งานกับกระบวนการทั่วไปที่ใช้ในโรงงาน

ลักษณะของการควบคุมกระบวนการประกอบไปด้วยโหมดการควบคุมแบบ PID Control , PID-FUZZY Control และ Manual Control โดยนำสัญญาณอินพุตที่เป็นแรงดันหรือกระแสมาตรฐาน แปลงเป็นสัญญาณดิจิทัลมาคำนวณแล้วได้ผลลัพธ์และส่งสัญญาณควบคุมกระบวนการเป็นสัญญาณแรงดันหรือกระแสมาตรฐาน ซึ่งการแปลงสัญญาณสำหรับโครงการนี้จะใช้อุปกรณ์ที่ออกแบบสร้างขึ้นเอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**THESIS** DESIGN OF PID-FUZZY CONTROLLER AND MONITORING  
PROCESS SYSTEM USING MICROCOMPUTER

**BY** MR. TEERAWUT RATTANAVISUT  
MR. AKEKACHAI CHATTRANET  
MR. KITIPONG NOPPAKORNVISATE  
MR. WACHARIN INTARIT

**DEGREE** BACHELOR OF ENGINEERING  
INSTRUMENTATION ENGINEERING

**DEPARTMENT** INDUSTRIAL INSTRUMENTATION TECHNOLOGY

**ACADEMIC YEAR** 1995

**THESIS ADVISOR** MR. SAKREYA CHITWONG



## ABSTRACT

Project - **DESIGN OF PID-FUZZY CONTROLLER AND MONITORING PROCESS SYSTEM USING MICROCOMPUTER** is to apply fuzzy control theory in finding PID Controller parameter and using general micro computer to control process. The program written for the project is designed for ease of use and ability to control and can display output in both numeric and graphic format which can be seen in general in factory process.

There are three types of control in the project : PID Control , PID-FUZZY Control and Manual Control which can be either voltage or current input from sensor will be convert to digital signal for calculation. The result signal will be converted back to either voltage or current to control the process. The hardware for A/D and D/A was part of the design in this project.

# สารบัญ

## บทนำ

วัตถุประสงค์ของปริญญาโท	ก
ประวัติย่อของ FUZZY CONTROL	ข
CONCEPTS OF FUZZY LOGIC	ข
FUZZY MODEL	จ

## บทที่ 1 ชนิดของเครื่องควบคุมแบบป้อนกลับ

เครื่องควบคุมแบบสัดส่วน Proportional Controller (P)	1-1
เครื่องควบคุมแบบสัดส่วนร่วมกับอินทิกรัล Proportional - Integral Controller (PI)	1-6
เครื่องควบคุมแบบสัดส่วนร่วมกับอินทิกรัลและเดอริเวทีฟ Proportional - Integral Derivative Controller (PID)	1-9
เครื่องควบคุมแบบสัดส่วนและเดอริเวทีฟ Proportional - Derivative Controller (PD)	1-11
เครื่องควบคุมแบบดิจิตอล Digital Controller	1-12
รีเซ็ต - วินด์อัป	1-13

## บทที่ 2 PID - FUZZY CONTROLLER

ทฤษฎี FUZZY CONTROL	2-2
คลิซเซต (CRISP SET)	2-3
ฟัซซีเซต (FUZZY SET)	2-4
ฟัซซีลอจิก (FUZZY LOGIC)	2-6
กฎการวินิจฉัย (INFERENCE)	2-7
วิธีการ DEFUZZIFIER	2-9
PID CONTROLLER	2-12
FUZZY GAIN SCHEDULING	2-13
ขั้นตอนการออกแบบ SOFTWARE	2-19

## บทที่ 3 การอินเตอร์เฟส

การอ้างแอดเดรสของพอร์ต I/O	3-1
เทคนิคในการดีโค๊ดแอดเดรสสำหรับพอร์ต I/O	3-6
ลักษณะต่างๆ ไปของ CARD I/O	3-18
การ DECODE PORT	3-19

#### บทที่ 4 โปรแกรมควบคุมกระบวนการและการแสดงผล

ความสามารถของโปรแกรม	4-1
การทำงานของโปรแกรม	4-4
การใช้งานโปรแกรม	4-9

#### บทที่ 5 ผลการทดลอง

โหมด PID	5-1
โหมด PID - FUZZY	5-4

#### บทที่ 6 บทสรุป

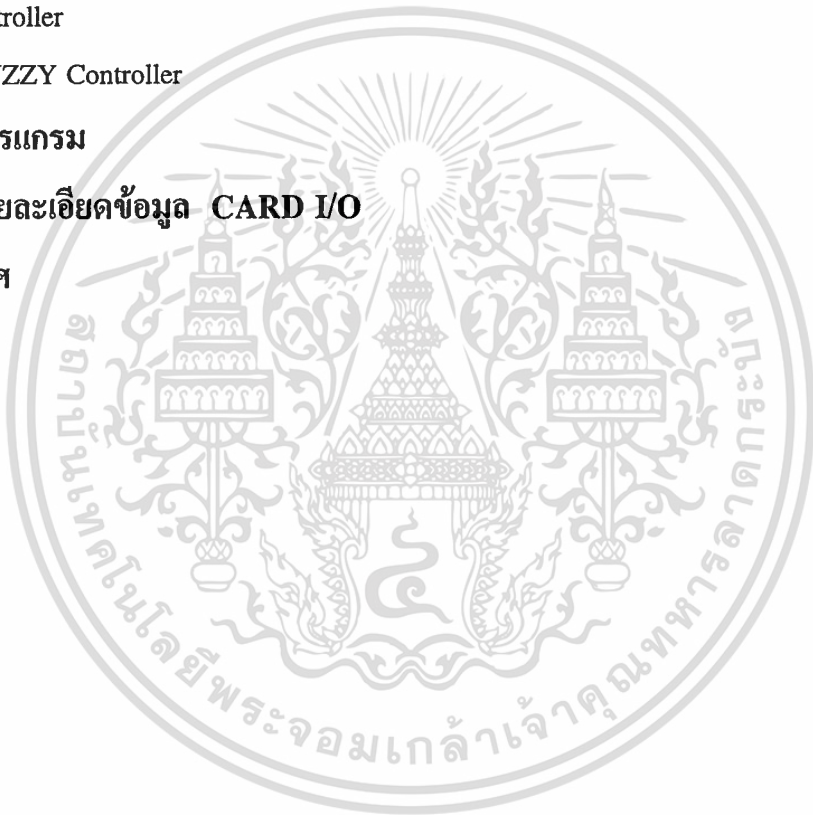
PID Controller	6-1
PID - FUZZY Controller	6-2

ภาคผนวก ก. โปรแกรม

ภาคผนวก ข. รายละเอียดข้อมูล CARD I/O

กิตติกรรมประกาศ

หนังสืออ้างอิง



## บทนำ

FUZZY LOGIC เป็นศาสตร์ที่มีมากกว่า 30 ปี แล้วแต่เพิ่งจะมาเป็นที่ยอมรับอย่างแพร่หลายเมื่อไม่นานมานี้ สิ่งที่ทำให้การควบคุมด้วยระบบ FUZZY LOGIC เป็นที่สนใจของชาวโลกคือ ในปี 1987 ที่ประเทศญี่ปุ่น ได้ประสบความสำเร็จในการควบคุมการจราจรทางรถไฟด้วยทฤษฎี FUZZY LOGIC ที่ Sendai นับจากจุดนั้นมาบริษัทและมหาวิทยาลัยหลายแห่งในประเทศญี่ปุ่น ได้มีการศึกษาอย่างจริงจังในศาสตร์แขนงนี้ จนสามารถทำให้นำทฤษฎีนี้มาใช้ในอุปกรณ์เครื่องใช้ไฟฟ้าต่าง ๆ มากมาย เช่น ใช้ในเครื่องปรับอากาศ กล้องวิดีโอ หม้อหุงข้าว และกล้องถ่ายภาพ เป็นต้น และปัจจุบันนี้ยังได้นำไปประยุกต์ใช้ใน Expert System เช่น ควบคุมอุณหภูมิของหม้อกลั่นน้ำมัน ในโรงกลั่น การควบคุมระบบ Automatic Transmission ในรถยนต์ เป็นต้น ซึ่งการควบคุมเหล่านี้ สามารถทำให้ได้ผลลัพธ์ใกล้เคียงผลลัพธ์ซึ่งเกิดจากการตัดสินใจของมนุษย์ ทำให้ผลการควบคุมออกมาเป็นที่น่าสนใจเป็นอย่างยิ่ง

ในการควบคุมกระบวนการทางอุตสาหกรรม มีวิธีการในการควบคุมหลายวิธีซึ่งโดยปกติแล้วเรามักใช้ PID Controller ในการควบคุม แต่ในปัจจุบันกระบวนการทางอุตสาหกรรมมีความซับซ้อนขึ้น ซึ่งทำให้การควบคุมโดยใช้ PID Control มีขีดจำกัดอยู่ระดับหนึ่ง และ เนื่องจากความซับซ้อนของกระบวนการ ทำให้การหารูปแบบของกระบวนการออกมาเป็นแบบจำลองทางคณิตศาสตร์ทำได้ลำบากจึงเริ่มมีการคิดวิธีการควบคุมแบบใหม่ ๆ เพื่อแก้ปัญหาดังกล่าว ซึ่งปัจจุบันมีวิธีการในการควบคุมกระบวนการที่ได้รับการคิดค้นและพัฒนาขึ้นมาหลายวิธี ซึ่งวิธีการที่ได้รับความนิยมอย่างแพร่หลาย คือ การควบคุมโดยการเลียนแบบความคิดของมนุษย์ ได้แก่ การควบคุมโดยใช้ทฤษฎี Fuzzy Set และ Neural Network

สำหรับ Project ที่นำเสนอนี้เป็นการควบคุมกระบวนการโดยใช้ PID-Fuzzy Controller ซึ่งอาศัยหลักการของ Fuzzy Set มาใช้ในการออกแบบค่าพารามิเตอร์ของ Controller การควบคุมกระบวนการใน Project ที่นำเสนอนี้ทุกขั้นตอนจะทำงานด้วย Computer โดยระบบนั้นเป็นระบบ input เดียว output เดียว โดยการนำ PID-Fuzzy Controller ซึ่งเป็น Software ที่ได้ทำการสร้างขึ้นร่วมกับ Computer Hardware ไปใช้ในระบบควบคุมกระบวนการ (Process)

### วัตถุประสงค์ของปริญญาานิพนธ์

- 1.ศึกษาหลักการของการควบคุมกระบวนการด้วย Fuzzy Controller ซึ่งอาศัยหลักการของทฤษฎี Fuzzy set และ Fuzzy logic
- 2.ศึกษาการเลือกใช้กฎพื้นฐาน(Rules) และ ฟังก์ชันระดับการเป็นสมาชิก(Membership Function) ให้เหมาะสมกับกระบวนการ(Process)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.ทำการออกแบบ PID-Fuzzy Controller ไปประยุกต์ใช้งานจริงโดยการนำ PID-Fuzzy Controller ซึ่งเป็น Software ที่ได้ทำการสร้างขึ้น ร่วมกับ PC Hardware ไปใช้ในระบบควบคุมป้อนกลับ (Feedback Control System)

4.ทำการทดลองการทำงานของ PID-Fuzzy Controller ด้วย PC computer

5.เปรียบเทียบระบบควบคุมป้อนกลับ (Feedback Control System) ที่ทำการควบคุมด้วย PID-Fuzzy Controller และ PID Controller

## ประวัติย่อของ FUZZY CONTROL

การวิจัยทางด้าน FUZZY CONTROL และ FUZZY MODELLING มีการเจริญเติบโตขึ้นอย่างรวดเร็วนับตั้งแต่ Professor Lotfy A.Zadeh แห่ง University of Berkeley ได้เสนอหลักการทางคณิตศาสตร์ของทฤษฎี FUZZY SET ขึ้นมาเป็นครั้งแรกในปี ค.ศ.1965 ทฤษฎี Fuzzy Set และ FUZZY ALGORITHM ได้นำไปประยุกต์ใช้ในการควบคุมเป็นครั้งแรกโดยใช้ควบคุมเครื่องทำความร้อน (Steam engine) โดย Professor Mamdani แห่ง London University ในปี ค.ศ. 1974 มีการอภิปรายถึงสถานะปัจจุบันของผลงานวิจัยที่เกี่ยวกับ FUZZY CONTROLLER โดย Mamdani และ Sembi ในปี ค.ศ.1980 Willaelys, Malvache และ Hammad ได้เสนอวิธีในการสร้าง Fuzzy controller ขึ้นมาจากพื้นฐานของ FUZZY MODEL ของ CONTROL PROCESS(1977) แนวคิดของ FUZZY ALGORITHM ได้ถูกนำมาใช้ในการควบคุมความเร็วของ DCMotor โดย Willaelys, Malvache และ Mangin(1977) Willaelys, Malvache(1978,1979) , Braae, Rutherford (1979) ได้นำเสนอ Linguistic analysis ของ FUZZY CONTROLLER MODEL Application ของ FUZZY CONTROL สำหรับระบบซึ่งเป็น Multidimensional, nonlinear และมี deadtime ได้ถูกสร้างขึ้นโดย Mamdani และ Procyk(1979) Algorithm สำหรับตรวจสอบ FUZZY ได้รับการเสนอจาก Czogala และ Pedrycz(1981) และ Tong(1979) ได้ทำการเสนอแนวทางสำหรับการสร้าง FUZZY MODEL ซึ่งประกอบด้วย Verballisation, Fuzzyfication และ Identification

ทฤษฎี FUZZY ได้นำไปประยุกต์ใช้ในการควบคุมกระบวนการอุตสาหกรรมอย่างแพร่หลายในทศวรรษที่ 80 อย่างแพร่หลายและได้รับการพัฒนาอย่างต่อเนื่องมาจนถึงปัจจุบัน

## CONCEPTS OF FUZZY LOGIC

คำว่า "FUZZY" คืออะไร?

คำนี้มาจากคำว่า "Fuzz" ซึ่งหมายถึงขนอ่อนที่ปกคลุมตัวลูกไก่ คำว่า "Fuzz" ในภาษาอังกฤษมีความหมายว่า "ไม่ชัดเจน มัว" คำนี้ตรงกับภาษาฝรั่งเศสว่า "Flou" ส่วนญี่ปุ่นนั้นใช้เอกสารเป็นเอกสารที่ส่งวันไวสำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นับญาติให้ไปใช้ประโยชน์ด้านการค้า

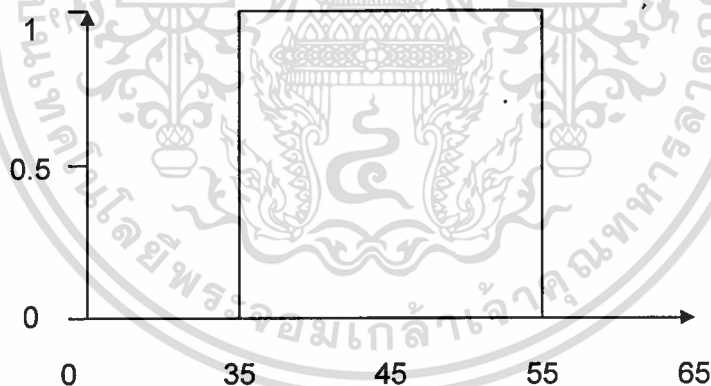
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คำว่า “Aimai” อย่างไรก็ดี ในเชิงวิชาการและเทคโนโลยี คำว่า “Fuzzy” นี้เป็นศัพท์เทคนิคโดยเฉพาะ และหมายความถึงสภาพแห่งความกำกวมหรือคลุมเคลือที่มีฐานมาจากสัญชาตญาณมนุษย์มากเสียยิ่งกว่าจากเรื่องของ Probability (ความเป็นไปได้หรือความน่าจะเป็น)

ประมาณ 30 ปีมาแล้วที่ศาสตราจารย์ ลอตฟี เอ ซาเดห์ (L.A. Zadeh) ได้คิดค้น Fuzzy Sets เพื่อต้องการปรับแนวความคิดเรื่องขอบเขตของ Fuzzy เข้ากับวิทยาศาสตร์ ทฤษฎี Fuzzy จึงได้รับการสร้างขึ้นบนรากฐานของ Fuzzy Sets นี้ และแล้ววิศวกรรมศาสตร์แขนงใหม่ภายใต้ชื่อว่า “Fuzzy Engineering” ก็เกิดขึ้น แม้ Fuzzy Sets อาจจะฟังดูว่าเกี่ยวกับคณิตศาสตร์มากไปหน่อยก็ตาม แต่แนวความคิดพื้นฐานสามารถอธิบายได้โดยไม่ยากเลยแม้แต่น้อย

### FUZZY ทำงานอย่างไร?

Fuzzy Sets เป็นทฤษฎีทางคณิตศาสตร์ที่ใช้เป็นเครื่องมือในการตัดสินใจ ซึ่งในการตัดสินใจที่ใช้ในระบบนี้จะเลียนแบบการตัดสินใจของมนุษย์ก็จะเป็นการประมาณ (Approximation) จากปัจจัยหรือตัวแปรต่าง ๆ ของระบบเพื่อทำการสรุปหาเหตุผล ดังนั้นฟัซซีลอจิกจึงไม่ใช่ลอจิกที่มีเพียง 0 กับ 1 หรือเราเรียกว่า Crisp Sets ดังแสดงในรูปที่ 1 เท่านั้น แต่เป็นลอจิกที่แทนความหมายของตัวแปร โดยบอกถึงระดับความเป็นสมาชิกของตัวแปรนั้น ๆ ดังแสดงในรูปที่ 2

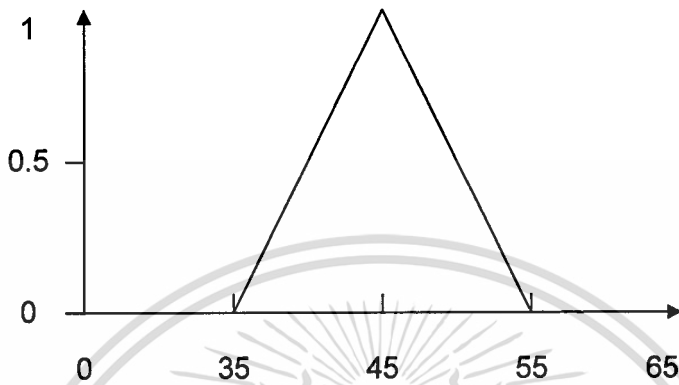


รูปที่ 1 แสดงกราฟ Crisp Sets ของช่วงอายุวัยกลางคน

เราลองมาดูตัวอย่างจากการนิยามโนภาพจากคำว่า “วัยกลางคน” ดู คำดังกล่าวนี้ชวนให้นึกถึงภาพ ๆ หนึ่งขึ้นมาในสมอง แต่ก็ยังเป็นภาพที่มีขอบเขตคลุมเคลือ คือไม่อาจทราบค่าที่แน่นอนได้ จึงไม่อาจจะใช้ระบบไบนารีในเครื่องคอมพิวเตอร์ธรรมดาแก้ปัญหานี้ได้ จุดนี้เองที่ทฤษฎี Fuzzy เข้ามาช่วยในการคำนวณหาขอบเขต โดยสมมติว่าเราสรุปว่า วัยกลางคนหมายถึงอายุ 45 ปี แต่คนอายุ 35 หรือ 55 ก็ไม่อาจจะจัดอยู่เข้า “ไม่ใช่วัยกลางคนได้” ดังนั้นความหมายของคำว่า “วัยกลางคน” นั้นค่อนข้างจะแตกต่างกันภายในกรอบแห่งขอบเขตนั้น ในทางตรงกันข้าม ผู้ที่มีอายุน้อยกว่า 30 หรือมากกว่า 60 ก็เรียกได้ว่า “ไม่อยู่ในข่ายวัยกลางคนแน่” ความคิดนี้แสดงออกมาได้ตาม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาก็เท่านั้น เมื่ออนุญาตให้เผยแพร่โดยไม่เสียประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กราฟรูปที่ 2 หรือตามฟังก์ชันที่เรียกว่า “ฟังก์ชันความเป็นสมาชิก” (Membership Function) ที่มีค่าเกรดระหว่าง 0 และ 1 โดย Fuzz Sets นี้ก็แสดงออกในรูปของฟังก์ชันความเป็นสมาชิกนี้ อย่างไรก็ตามสิ่งที่สังเกตได้คือ ค่าเกรดระหว่างฟังก์ชันความเป็นสมาชิก สามารถจะแปรเปลี่ยนได้ระหว่าง 0 และ 1 นี้เองทำให้การแสดงออกมาเป็นรูปธรรมที่เป็นตัวเลขได้ดังแสดงในรูปที่ 2



รูปที่ 2 แสดงกราฟ Fuzzy Sets ของช่วงอายุวัยกลางคน

จากรูปที่ 1 คนอายุ 34 และ 56 จะไม่ถูกจัดเป็น “วัยกลางคน” แต่ใครก็ตามที่ตอนนี้อยู่ที่คิดว่าตัวเองหนุ่มด้วยอายุแค่ 34 ก็จะเข้า “วัยกลางคน” ทันทีเมื่อถึงวันเกิดในปีหน้า ลักษณะก่อนข้างยี่ดหุ่ยนี้ เกิดขึ้นก็เป็นเพราะ การกำหนดค่าที่ไม่มีควมยี่ดหุ่ย แนวความคิดที่มีการกำหนดค่าชัดเจนในรูป 0 หรือ 1 เช่นเดียวกับระบบไบนารีที่ใช้กันใคอมพิวเตอร์ธรรมดาทั่วไปนั้น ทำงานโดยกำหนดช่วงตายตัว (Fixed Range) โดยที่เราที่กำหนดไว้ว่า “วัยกลางคน” คือช่วงระหว่าง 35 ถึง 55 ปี ลักษณะเช่นนี้เรียกว่า Crisp Sets ซึ่งจะมีลักษณะที่ตรงกันข้ามกับ Fuzzy Sets อย่างสิ้นเชิง

### ข้อได้เปรียบของ Fuzzy Logic เหนือ Traditional Logic

Fuzzy Logic theory มีข้อได้เปรียบเหนือ Traditional Logic หรือลอจิกแบบทั่วไปอยู่ 2 ประการ คือ

ประการแรก Fuzzy Sets สามารถให้คำจำกัดความของคุณภาพทางภาษา (Qualitatively Using Linguistic Terms) ได้ดี เช่น คำว่า ค่อนข้าง หรือ เกือบจะ ประกอบกับค่าแสดงคุณลักษณะทั่ว ๆ ไป เช่น สูง เตี้ย ดี เลว ร้อน เย็น และอื่น ๆ และยังสามารถให้ค่าความสำคัญของแต่ละสมาชิกใน Fuzzy Sets ได้ (Degrees of Membership)

ประการที่ 2 เอาท์พุทที่ได้จาก Fuzzy System จะมีการเปลี่ยนแปลงแบบค่อยเป็นค่อยไปอย่างต่อเนื่อง ถึงแม้ว่า อินพุทของ Fuzzy System จะเปลี่ยนไปอย่างทันทีทันใดก็ตาม

## Fuzzy Model

องค์ประกอบของระบบควบคุมแบบฟัซซี่ลอจิก และระบบควบคุมที่ใช้กันอยู่ในปัจจุบัน จะมีลักษณะแตกต่างกันไม่มาก

ส่วนหลัก ๆ ที่ได้เพิ่มเข้ามาใช้สำหรับระบบควบคุมที่ใช้ระบบฟัซซี่ลอจิก คือ

1 Fuzzification of Inputs คือ ขั้นตอนที่ทำหน้าที่แปลงข้อมูลที่เข้ามา ไปเป็นอินพุตของระบบฟัซซี่ ซึ่งจะอยู่ในรูปของค่าความเป็นสมาชิก (Degree of Membership) ในฟัซซี่เซต ดังที่ได้กล่าวมาแล้วในตอนต้น

2 Rule Evaluation คือ ขั้นตอนการตีความหรือวิเคราะห์ซึ่งใช้เทคนิคของ 'and' Minimum หรือ 'or' โดยใช้ Maximum ของตัวแปรส่วนเงื่อนไข (Condition) ในการคำนวณผลลัพธ์รวมจากกฎ IF/THEN RULES ของ Fuzzy System ซึ่งโดยปกติจะใช้เทคนิคของกฎ Minimum หรือกฎ Smallest เพื่อที่จะทำให้ความแข็งแกร่งของกฎที่ได้เลือกอยู่นั้นขึ้นกับค่าของตัวแปรส่วนเงื่อนไข (Condition) ที่มีค่าความเป็นสมาชิกน้อยที่สุด

โดยลักษณะของ IF/THEN RULES มีดังนี้

IF เงื่อนไข THEN ผลลัพธ์

หรือ IF เงื่อนไข(1) และ เงื่อนไข(2) THEN ผลลัพธ์

3 Defuzzification of Outputs คือขั้นตอนที่ทำการเปรียบเทียบ Fuzzy Output ทั้งหมดเข้ากับผลลัพธ์รวมซึ่งในยุคแรก ๆ ของระบบ Fuzzy จะใช้ Fuzzy Output ที่มีค่ามากที่สุดในระบบเป็นผลลัพธ์ของระบบ (Maximum Defuzzification) ซึ่งในปัจจุบันวิธีการนี้ไม่ได้นิยม เพราะว่าจะมีการละทิ้งผลลัพธ์ของกฎบางกฎ จึงเปลี่ยนมาใช้วิธีหา Centroid (Center of Gravity) เพราะสามารถถ่วงน้ำหนักของ Fuzzy Outputs ได้อย่างทั่วถึง โดยวิธี Centroid ทำได้โดย

$$Z = abz_1 + a(1-b)z_2 + (1-a)bz_3 + (1-a)(1-b)z_4$$

$Z = \text{Result of centroid method}$

$a = \mu_{e(k)}$  ; Degree of membership  $e_{(k)}$

$b = \mu_{\Delta e(k)}$  ; Degree of membership  $\Delta e_{(k)}$

# บทที่ 1

## ชนิดของเครื่องควบคุมแบบป้อนกลับ

หนทางหนึ่งที่เครื่องควบคุมแบบป้อนกลับจะทำการตัดสินใจเพื่อที่จะรักษาผลของการควบคุมไว้ที่จุดเป้าหมายโดยการคำนวณค่าทางออกบนพื้นฐานของความแตกต่างระหว่างตัวแปรที่ถูกรักษาและค่าเป้าหมาย ในส่วนนี้เราจะมองถึงเครื่องควบคุมที่พบกันทั่วไปโดยดูที่สมการซึ่งเป็นการอธิบายการทำงานของเครื่องควบคุมเหล่านี้

**เครื่องควบคุมแบบสัดส่วน Proportional Controller (P)** เครื่องควบคุมแบบสัดส่วน คือเป็นชนิดที่ง่ายที่สุดของเครื่องควบคุม โดยยกเว้นเครื่องควบคุมแบบเปิด - ปิด ซึ่งจะไม่สามารถกล่าวถึงในที่นี้ สมการที่อธิบายการทำงานของเครื่องควบคุมแบบสัดส่วนเป็นดังนี้

$$m(t) = \bar{m} + Kc[r(t) - c(t)] \quad (1.1)$$

หรือ

$$m(t) = \bar{m} + Kc.e(t) \quad (1.2)$$

เมื่อ

$m(t)$  = ทางออกของเครื่องควบคุม , *psig* หรือ *mA*

$r(t)$  = ค่าเป้าหมาย , *psig* หรือ *mA*

$c(t)$  = ตัวแปรที่ถูกรักษา , *psig* หรือ *mA* เป็นสัญญาณที่ได้จากอุปกรณ์ส่งสัญญาณ

$e(t)$  = สัญญาณผิดพลาด , *psig* หรือ *mA* เป็นความแตกต่างระหว่างค่าเป้าหมายกับตัวแปรที่ถูกรักษา

$Kc$  = อัตราการขยายของเครื่องควบคุม , *psi / psi*, *mA / mA*

$\bar{m}$  = ค่าไบอัส , *psig* หรือ *mA* เป็นค่าทางออกของเครื่องควบคุมเมื่อสัญญาณผิดพลาด

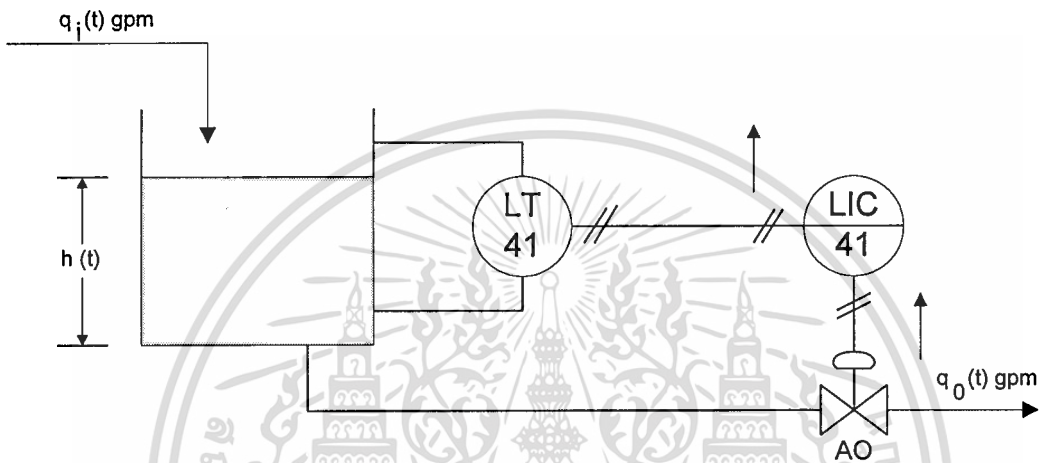
เป็นศูนย์ค่านี้ปกติจะถูกตั้งไว้ในระหว่างการปรับเทียบเครื่องควบคุมอยู่ที่กึ่งกลางสเกล คือ 9 *psig* หรือ 12 *mA*

โดยที่พิกัดทางเข้าและทางออกนั้นเหมือนกัน (3-15 *psig* หรือ 4-20 *mA*) และสัญญาณทางเข้าและสัญญาณทางออกตลอดจนค่าเป้าหมายเหมือนกัน ซึ่งในบางครั้งอาจจะเป็นเศษส่วนหรือเปอร์เซ็นต์ของพิกัด มันเป็นเรื่องที่น่าสนใจในสมการที่ (1.1) ที่ถูกเขียนขึ้นมาสำหรับเครื่องควบคุมที่เป็นกริยาทางออกแบบกลับ คือถ้าตัวแปรที่ถูกรรักษามีค่าเพิ่มมากขึ้นเหนือค่าเป้าหมายและค่าผิดพลาดกลายเป็นค่าลบ จากสมการแสดงให้เห็นว่าทางออกของเครื่องควบคุมนั้นได้ลดลงในทางที่จะแสดงให้เห็นถึงเครื่องควบคุมที่เป็นกริยาทางออกแบบตรง ในทางคณิตศาสตร์สามารถทำได้โดยการให้อัตราการขยายของเครื่องควบคุมที่เป็นลบ แต่เราจะต้องจำไว้ว่าหนึ่งในเครื่องควบคุมในอุตสาหกรรมจะไม่มีอัตราขยายที่เป็นค่าลบแต่จะมีเฉพาะค่าบวกอย่างเดียว สวิตซ์เลือก

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

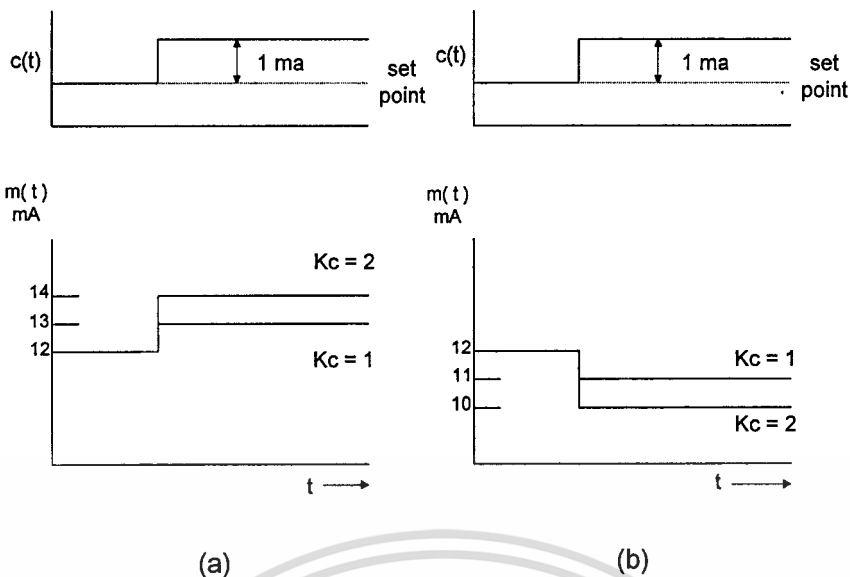
แบบกลับ/แบบตรง จึงได้ถูกนำมาใช้สำหรับอัตราการขยายที่เป็นลบถูกใช้เมื่อมีการกระทำทางคณิตศาสตร์ของระบบควบคุมเพื่อที่จะวิเคราะห์เครื่องควบคุมกริยาทางออกแบบตรง

สมการ (1.1) และ (1.2) แสดงให้เห็นทางออกของเครื่องควบคุมที่เป็นสัดส่วนกับค่าผิดพลาดระหว่างค่าเป้าหมายกับตัวแปรที่ถูกควบคุม ความเป็นสัดส่วนนี้ถูกกำหนดโดยค่าอัตราการขยายของเครื่องควบคุม อัตราการขยายหรือความไวของเครื่องควบคุมนี้จะบอกให้รู้ว่าทางออกจะมีขนาดมากเท่าใดจากการเปลี่ยนแปลงของค่าผิดพลาดที่เกิดขึ้นอันนี้จะได้แสดงในรูปที่ 1-2

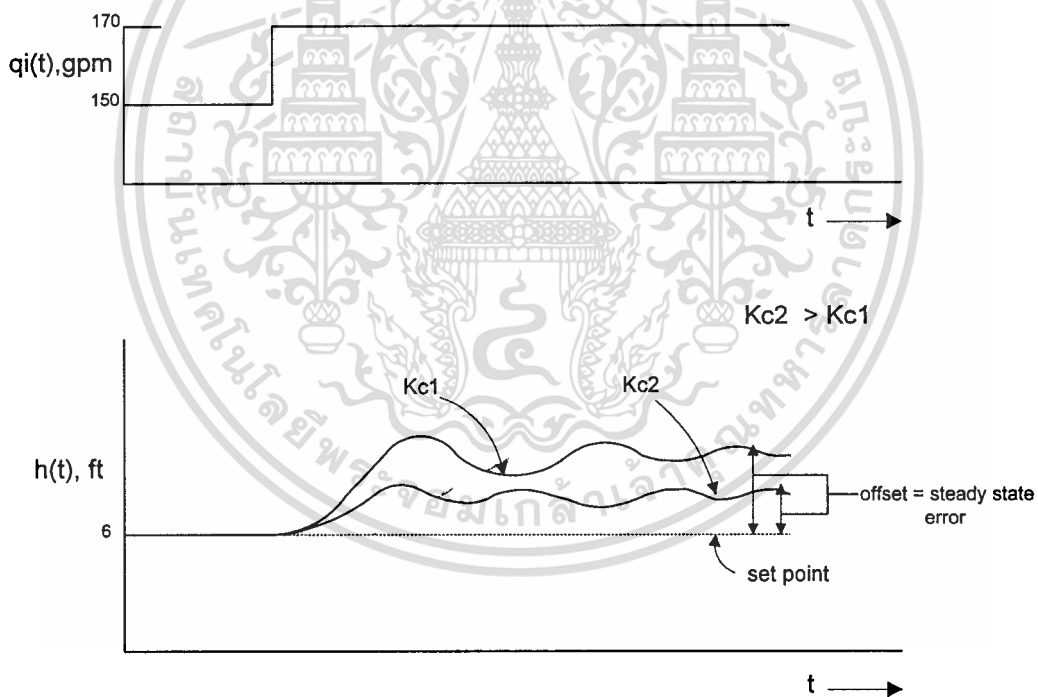


รูปที่ 1-1 วงรอบการควบคุมของระดับของเหลว

เครื่องควบคุมแบบสัดส่วนอย่างเดียวมีข้อดีคือมีตัวปรับค่าอัตราการขยายเพียงตัวเดียว อย่างไรก็ตามข้อเสียของเครื่องควบคุมแบบสัดส่วนนั้น ในการทำงานจะเกิดค่าผิดพลาดที่สภาวะคงที่ (Steady - State Error) OFFSET เพื่อที่จะแสดงถึงค่าผิดพลาดที่สภาวะคงที่เป็นกราฟพิจารณวงรอบการควบคุมระดับของเหลวในรูปที่ 1-1 กำหนดให้เงื่อนไขการทำงานที่  $\bar{q}_i = \bar{q}_o = 150 \text{ gpm}$  และ  $\bar{h} = 6 \text{ ft}$  เพื่อที่จะให้ค่าที่ทางออกของวาล์วเป็น  $150 \text{ gpm}$  จะต้องให้แรงดันลมแก่วาล์ว  $9 \text{ psig}$  แต่ถ้าอัตราไหลทางเข้าเพิ่มขึ้นผลตอบสนองของระบบร่วมกับเครื่องควบคุมแบบสัดส่วนแสดงดังรูปที่ 1-3 เครื่องควบคุมได้นำค่าของตัวแปรที่ถูกควบคุมกลับเข้าสู่สภาวะคงที่แต่ค่าที่สภาวะคงที่นี้จะไม่ใช่ค่าที่จุดเป้าหมาย ความแตกต่างระหว่างค่าเป้าหมายและค่าที่สภาวะคงที่ของตัวแปรที่ถูกควบคุมคือออฟเซ็ท รูปที่ 1-3 แสดงให้เห็นผลตอบสนองของค่าตัวแปรที่ถูกควบคุมทั้งสองค่าที่แตกต่างกันของค่าอัตราการขยายที่ถูกปรับเปลี่ยน จะพบว่าค่าอัตราการขยายของเครื่องควบคุมที่มากจะให้ค่าออฟเซ็ทน้อยแต่จะเกิดการแกว่งมากกว่า อย่างไรก็ตามจะมีค่าที่มากที่สุดของอัตราการขยายของเครื่องควบคุมที่ทำให้กระบวนการเข้าสู่ความไม่เสถียรภาพการคำนวณค่าที่มากที่สุดของอัตราการขยายจะนี้เรียกว่า **Ultimate gain**



รูปที่ 1-2 ผลของอัตราขยายในเครื่องควบคุมต่อทางออกของเครื่องควบคุม  
 a) เครื่องควบคุมกรียาทางออกแบบตรง b) เครื่องควบคุมกรียาทางออกแบบกลับ



รูปที่ 1-3 ผลตอบสนองของกระบวนการควบคุมระดับของเหลว

การพิจารณาระบบควบคุมระดับของเหลวเดียวกันนี้ในรูปที่ 1-1 ด้วยเงื่อนไขของการทำงานเหมือนเดิม และจำไว้ว่าอย่างหนึ่งว่าเครื่องควบคุมแบบสัดส่วนอย่างเดียวที่กรียาทางออกแบบตรงนั้นอัตราขยายเป็นลบ ( $-K_c$ ) ดูสมการต่อไปนี้

$$m(t) = 9 + (-K_c).e(t) \tag{1.3}$$

ตอนนี้กำหนดให้อัตราไหลทางเข้าเพิ่มขึ้นเป็น 170 gpm เมื่อเป็นเช่นนี้แล้วระดับของเหลวจะต้อง

เพิ่มขึ้นจากนั้นเครื่องควบคุมจะส่งสัญญาณทางออกไปเปิดวาล์วเพื่อที่จะทำให้ระดับของเหลวกลับ

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เข้าสู่สภาวะคงที่ในตำแหน่งเดิม อัตราไหลทางออกตอนนี้จะต้องเป็น  $170 \text{ gpm}$  เพื่อที่จะให้วาล์วผ่านอัตราไหลค่าใหม่ที่เพิ่มขึ้นก็จะต้องเปิดให้มากขึ้นกว่าเดิม วาล์วที่ใช้เป็นแบบแรงดันเพื่อเปิดสมมติให้ค่าแรงดันค่าใหม่ที่กระทำกับวาล์วนี้เป็น  $10 \text{ psig}$  นั่นคือเครื่องควบคุมจะต้องให้ทางออกเป็น  $10 \text{ psig}$  แก่วาล์ว มองกลับไปที่สมการ (1.3) จะเห็นว่ามียอดเดียวกันเท่านั้นที่ทางออกของเครื่องควบคุมเท่ากับ  $10 \text{ psig}$  ได้ก็ต่อเมื่อเทอมที่สองทางขวามือจะต้องเป็น  $+1 \text{ psig}$  และในเทอมค่าผิดพลาดไม่สามารถที่จะเป็นศูนย์ที่สภาวะคงที่ได้ และค่าผิดพลาดที่สภาวะคงที่ก็คือออฟเซต ค่าผิดพลาดที่เป็นลบอันเนื่องมาจากค่าของตัวแปรที่ถูกควบคุมมีค่ามากกว่าค่าเป้าหมาย

มีอยู่สองจุดที่น่าสนใจในตัวอย่างนี้ อันแรกคือขนาดของเทอมความผิดพลาดขึ้นอยู่กับค่าอัตราขยายของเครื่องควบคุมโดยที่ทั้งเทอมรวมกันแล้วมีค่าเท่ากับ  $+1 \text{ psig}$  ดังนี้

$-Kc$	$-e(\infty)(\text{offset})$
1	1
2	0.5
4	0.25

จากที่กล่าวมาแล้วว่าอัตราขยายที่มากจะเกิดออฟเซตน้อย ผู้อ่านจะต้องจำไว้ว่าอัตราขยายที่มากกว่าขอบเขตอันหนึ่งแล้วจะทำให้ขบวนการเข้าสู่ความไม่เสถียรภาพ

อันดับที่สองเพื่อเป็นการสรุปในตัวอย่างนี้ จะเห็นว่าเครื่องควบคุมแบบสัดส่วนอย่างเดียวนี้จะพยายามทำหน้าที่โดยให้การเปลี่ยนแปลงนั้นเข้าสู่สภาวะคงที่ และจำนวนของความเบี่ยงเบนจากค่าเป้าหมายหรือออฟเซต ขึ้นอยู่กับอัตราขยายของเครื่องควบคุม

ผู้ผลิตเครื่องควบคุมส่วนมากจะไม่ใช้เทอมของอัตราขยายเพื่อที่จะบ่งบอกความไวของเครื่องควบคุม แต่จะใช้เทอมของ **Proportional Band PB** โดยจะเป็นความสัมพันธ์กับอัตราขยาย

$$\text{คือ} \quad PB = \frac{100}{K_c} \quad (1.4)$$

ดังนั้น สมการที่ได้อธิบายสำหรับเครื่องควบคุมแบบสัดส่วนเขียนใหม่ดังนี้

$$m(t) = \bar{m} + \frac{100}{PB} (r(t) - c(t)) \quad (1.5)$$

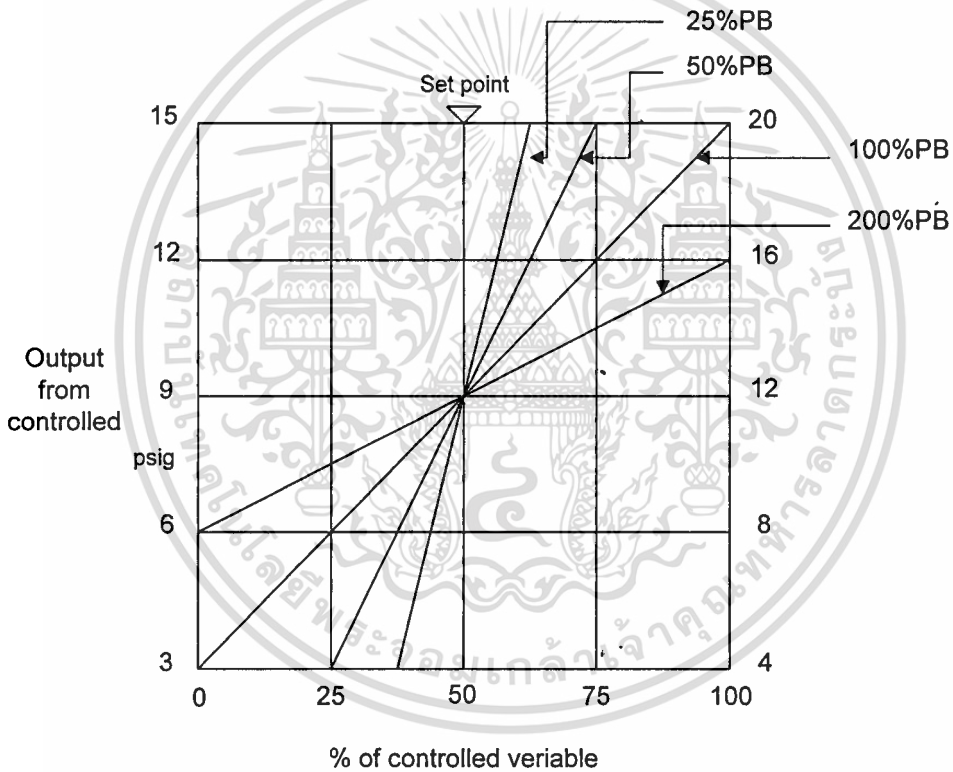
หรือ

$$m(t) = \bar{m} + \frac{100}{PB} e(t) \quad (1.6)$$

เทอม 100 ถูกนำมาใช้เพราะว่า PB โดยปกติจะอ้างถึง Percent Proportional Band

สมการ (1.6) แสดงให้เห็นจริงว่าอัตราขยายมากก็เหมือนกับมี Proportional Band ต่ำ หรือแคบ และถ้าอัตราขยายต่ำแล้วจะเหมือนกับมี Proportional Band มากหรือกว้าง นั่นก็หมายความว่าก่อนที่จะเริ่มใช้งานเครื่องควบคุมอย่างน้อยที่สุดก็ต้องรู้ว่าเครื่องควบคุมนี้มีการปรับแบบอัตราขยายหรือปรับ Proportional band

คำจำกัดความอีกอย่างหนึ่งของ Proportional Band ก็คือจะมองที่ผิดพลาด (ที่กระจายอยู่ในเทอมของเปอร์เซ็นต์พิกัดของตัวแปรที่ถูกควบคุม) ที่ต้องการที่จะเคลื่อนค่าที่ทางออกของเครื่องควบคุมจากค่าที่ต่ำสุดไปสู่สูงสุด พิจารณาวงรอบการควบคุมกระบวนการแลกเปลี่ยนความร้อน อุปกรณ์ส่งสัญญาณมีพิกัดจาก 100-300 องศาเซลเซียส และค่าเป้าหมายของเครื่องควบคุมอยู่ที่ 200 องศาเซลเซียส รูปที่ 1-4 จะแสดงกราฟที่เป็นคำจำกัดความของ PB ในรูปนี้แสดงค่า PB 100% ก็แสดงว่าค่าตัวแปรที่ถูกควบคุมสามารถที่จะแปรค่าได้ 100% ของพิกัด ถ้า PB 50% เช่นเดียวกันก็แสดงว่าค่าตัวแปรที่ถูกควบคุมสามารถที่จะแปรค่าได้ 50% ของพิกัด และถ้า PB 200% ถูกต้องค่าตัวแปรที่ถูกควบคุมสามารถที่จะแปรค่าได้ 200% ของพิกัด แต่อย่างไรก็ตามเครื่องควบคุมแบบสัดส่วนอย่างเดียวเครื่องนี้ไม่สามารถที่จะให้ค่าทางออกเกินกว่าพิกัดที่เป็นอยู่ได้ PB 200% จะหมายถึงว่าความไว้น้อยมากต่อความผิดพลาดหรืออัตราการขยายน้อยมากนั่นเอง



	Controller output		
	3 psig	9 psig	5 psig
	4 mA	2 mA	20 mA
PB = 100%	100 C	200 C	300 C
PB = 50%	150 C	200 C	300 C
PB = 25%	175 C	200 C	300 C
PB = 200%		200 C	300 C

รูปที่ 1-4 คำจำกัดความของ Proportional Band

เพื่อที่จะหาฟังก์ชันถ่ายโอนของเครื่องควบคุมแบบสัดส่วนอย่างเดียว สมการ 5-27 สามารถจะเขียนได้ดังนี้

$$m(t) - \bar{m} = Kc.(e(t) - 0) \quad (1.7)$$

กำหนดตัวแปรเบี่ยงเบนได้ดังนี้

$$M(t) = m(t) - \bar{m} \quad (1.8)$$

$$E(t) = e(t) - 0 \quad (1.9)$$

ดังนั้น

$$M(t) = Kc.E(t) \quad (1.10)$$

แปลงลาปลาซจะได้ฟังก์ชันถ่ายโอนดังนี้

$$\frac{M(s)}{E(s)} = Kc \quad (1.11)$$

สรุปได้ว่าเครื่องควบคุมแบบสัดส่วนอย่างเดียวเป็นนั่นเป็นชนิดที่ง่ายมากด้วยข้อดีที่มีพารามิเตอร์เพียงตัวเดียวในการปรับแต่งอัตราขยายหรือ PB ส่วนข้อเสียก็คือว่าการทำงานของเครื่องควบคุมแบบนี้จะเกิดออฟเซ็ทแก่ตัวแปรที่ถูกควบคุมในกรณีที่เครื่องควบคุมแบบสัดส่วนอย่างเดียวสามารถจะควบคุมได้ก็เป็นกระบวนการที่ตัวแปรที่ถูกควบคุมอยู่ในขอบเขตของค่าเป้าหมายอย่างไรก็ตามกระบวนการที่การควบคุมจะต้องอยู่ที่ค่าเป้าหมายเพียงค่าเดียวนั้นเครื่องควบคุมดังกล่าวไม่สามารถนำมาใช้งานได้

**เครื่องควบคุมแบบสัดส่วนร่วมกับอินทิกรัล Proportional-Integral Controller (PI)** กระบวนการส่วนใหญ่ที่ไม่สามารถที่จะถูกควบคุมให้เกิดค่าออฟเซ็ทได้โดยค่าตัวแปรที่ถูกควบคุมจะต้องอยู่ที่ค่าเป้าหมายเลยจำเป็นจะต้องมีเทอมบางเทอมที่ชาญฉลาดกระทำร่วมกับการทำงานแบบสัดส่วนอย่างเดียวของเครื่องควบคุมเพื่อที่จะขจัดค่าออฟเซ็ทออกไป และเทอมที่ชาญฉลาดอันนี้เองคือ กริยาอินทิกรัล หรือ รีเซ็ท เพราะฉะนั้นเครื่องควบคุมก็จะกลายเป็นแบบสัดส่วนร่วมกับอินทิกรัล โดยได้ถูกจำกัดความไว้ในสมการดังนี้

$$m(t) = \bar{m} + Kc[r(t) - c(t)] + \frac{Kc}{\tau_i} \int [r(t) - c(t)] dt \quad (1.12)$$

หรือ

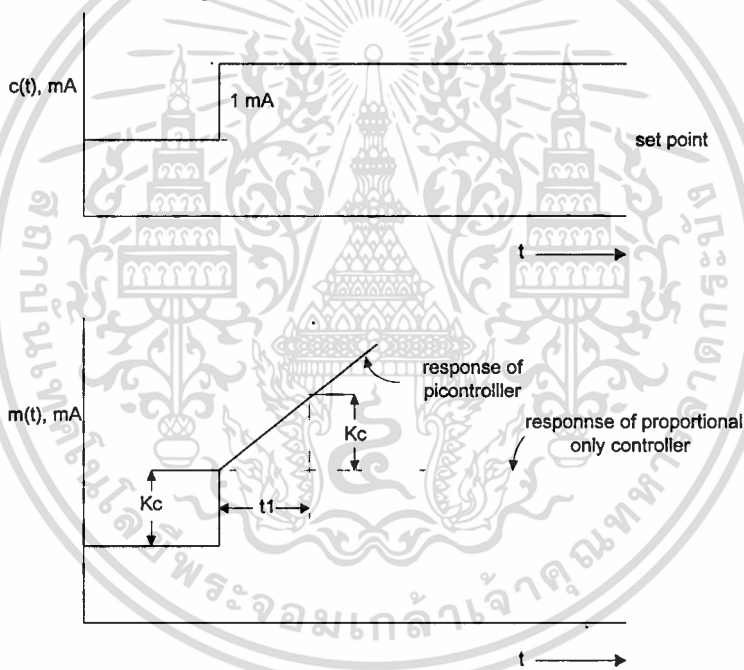
$$m(t) = \bar{m} + Kc[r(t) - c(t)] + \frac{Kc}{\tau_i} \int e(t) dt \quad (1.13)$$

เมื่อ  $\tau_i$  = เวลาในการอินทิเกรต หรือ เวลารีเซ็ท นาที/ จำนวนครั้ง

เพราะฉะนั้นเครื่องควบคุมแบบ PI จะมีสองพารามิเตอร์คืออัตราการขยายและเวลาในการอินทิเกรตซึ่งจะต้องถูกปรับแต่งเพื่อให้การควบคุมนั้นเหมาะสมที่สุด เพื่อที่จะให้เข้าใจคุณลักษณะของเวลารีเซ็ท  $\tau_i$  โดยพิจารณาตัวอย่างสมมติที่แสดงในรูปที่ 1-5  $\tau_i$  คือเวลาที่จะทำให้เครื่อง  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ควบคุมกระทำกริยาแบบสัดส่วนที่ซ้ำกันโดยหน่วยของเวลาอันนี้เป็นนาที/จำนวนครั้ง ค่าเวลาที่เล็กของ  $\tau_i$  จะทำให้เส้นโค้งของผลตอบสนองนั้นชันมาก ซึ่งจะหมายความว่า เป็นผลตอบสนองที่เร็ว อีกทางหนึ่งที่จะใช้ขยายความโดยดูที่สมการ (1.13) ค่าของ  $\tau_i$  ที่เล็กจะเป็นค่าที่มากในเทอมอยู่หน้าอินทิเกรต และจะเป็นตัวคูณของกริยาอินทิเกรตหรือรีเซ็ท

ต่อไปเราจะมาดูระบบควบคุมระดับของเหลวอีกครั้งที่จะถูกใช้สำหรับอธิบายว่าทำไมออฟเซ็ทจึงเกิดขึ้น เมื่ออัตราไหลทางเข้าเพิ่มขึ้นเป็น 170 *gpm* อัตราการไหลทางออกก็จำเป็นจะต้องพยายามให้เป็น 170 *gpm* ด้วย เพื่อให้มีอัตราไหลผ่านที่วาล์ว 170 *gpm* วาล์วจะต้องเปิดกว้างโดยมีแรงดันลมเพื่อเปิดเป็น 10 *psig* ด้วยเครื่องควบคุมแบบ PI ไม่ว่าค่าผิดพลาดจะเกิดด้วยเวลาที่ยาวนานเท่าใดเครื่องควบคุมก็จะเก็บค่าผิดพลาดนี้รวมกันไว้เพื่อชดเชยให้ที่ทางออกของเครื่องควบคุมในการแก้ความผิดพลาดและความผิดพลาดนั้นก็จะมีหมดไป เมื่อค่าผิดพลาดหมดไปแล้วทางออกของเครื่องควบคุมจะอยู่ในสภาวะ



รูปที่ 1-5 แสดงผลตอบสนองของเครื่องควบคุมแบบ PI (แบบกริยาทางออกตรง)

ต่อการเปลี่ยนแปลงของความผิดพลาดแบบเป็นขั้น

$$m(t) = \bar{m} + \frac{Kc}{\tau_i} (\int 0 dt) \quad (1.14)$$

ในความเป็นจริงแล้วถ้าความผิดพลาดเป็นศูนย์ไม่ได้หมายความว่าเทอมอินทิเกรตจะเป็นศูนย์ด้วย แต่จะหมายถึงเครื่องควบคุมกำลังอินทิเกรตค่าผิดพลาดที่เป็นศูนย์โดยการบวกด้วยศูนย์เข้าไปที่ทางออกซึ่งยังคงที่อยู่ สำหรับกระบวนการระดับของเหลวนี้อินทิเกรตเทอมจะมีค่าเท่า 1 *psig* เพราะฉะนั้นทางออกของเครื่องควบคุมจะเป็น 10 *psig* โดยปราศจากค่าผิดพลาดและที่เกิดขึ้นนี้ได้เป็นการขยายความว่าทำไมกริยารีเซ็ทจึงได้ขจัดออฟเซ็ทออกไปได้

เอกสารนี้เป็นเอกสารของกรมส่งเสริมการค้าระหว่างประเทศ กระทรวงพาณิชย์ อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$\frac{Kc}{\tau_i} \int_0^t dt \quad (1.15)$$

บางผู้ผลิตจะไม่ใช้เทอมของเวลารีเซ็ต  $\tau_i$  สำหรับเป็นค่าปรับแต่งแต่จะใช้อัตราการรีเซ็ต reset rate

$$\frac{R}{I} = \frac{1}{\tau_i}, \text{ จำนวนครั้ง/นาที} \quad (1.16)$$

เพราะฉะนั้นก่อนที่จะทำการปรับพารามิเตอร์เหล่านี้จะต้องรู้ว่าเครื่องควบคุมใช้ค่าของเวลารีเซ็ตหรืออัตราการรีเซ็ต ต่อไปจะขอแสดงสมการสำหรับเครื่องควบคุมแบบ PI ในอุตสาหกรรมที่ผู้ผลิตได้กำหนดไว้

Foxboro Co.

$$m(t) = \bar{m} + \frac{100}{PB} e(t) + \frac{100}{PB \cdot \tau_i} \int e(t) dt \quad (1.17)$$

Fisher Controls

$$m(t) = \bar{m} + \frac{100}{PB} e(t) + \frac{100\tau_i R}{PB I} \int e(t) dt \quad (1.18)$$

Taylor Co. Honeywell, Inc.

$$m(t) = \bar{m} + Kc \cdot e(t) + Kc \cdot \tau_i \frac{R}{I} \int e(t) dt \quad (1.19)$$

เป็นสิ่งที่น่าสนใจคือ Honeywell ได้ทำการพัฒนาเครื่องควบคุมที่ใช้ไมโครโปรเซสเซอร์ รุ่น TDC 2000 จะใช้สมการ (1.13) ส่วน Fisher Controls ได้ใช้ไมโครโปรเซสเซอร์ รุ่น PROVOX ใช้สมการ (1.16) และ (1.17) สมการที่ (1.13) ก็เป็นอันหนึ่งที่มีจุดประสงค์เพื่อเป็นมาตรฐานโดยสมาคมเครื่องมือวัดแห่งอเมริกา Instrument Society of America (ISA) สิ่งที่สำคัญที่จะต้องจดจำก็คือความสัมพันธ์ระหว่างอัตราขยายกับ Proportional Band และความสัมพันธ์ระหว่างค่ารีเซ็ตกับอัตรารีเซ็ต

เพื่อที่จะหาฟังก์ชันถ่ายโอนสำหรับเครื่องควบคุมแบบ PI สมการ (1.13) เขียนได้ดังต่อไปนี้

$$m(t) - \bar{m} = Kc(e(t) - 0) + \frac{Kc}{\tau_i} \int (e(t) - 0) dt \quad (1.20)$$

โดยใช้ข้อกำหนดเดิมของตัวแปรที่เบี่ยงเบนในสมการ (1.9) และ (1.10) แปลงลาปลาซจากรูปใหม่จะได้

$$\frac{M(s)}{E(s)} = Kc \left( 1 + \frac{1}{\tau_i s} \right) \quad (1.21)$$

สรุปเครื่องควบคุมแบบสัดส่วนร่วมกับอินทิเกรตจะมีสองพารามิเตอร์ที่จะต้องปรับแต่งได้แก่ อัตราขยายหรือ Proportional Band และค่าเวลารีเซ็ต ข้อดีของเครื่องควบคุมแบบนี้ก็คือ กริยาอินทิกรัลหรือรีเซ็ตจะทำการขจัดออฟเซตให้หมดไป

## เครื่องควบคุมแบบสัดส่วนร่วมกับอินทิกรัลและเดอริเวทีฟ Proportional-

**Integral-Derivative Controller (PID)** โหมดการควบคุมอันใหม่ที่ได้เพิ่มเข้าไปกับเครื่อง

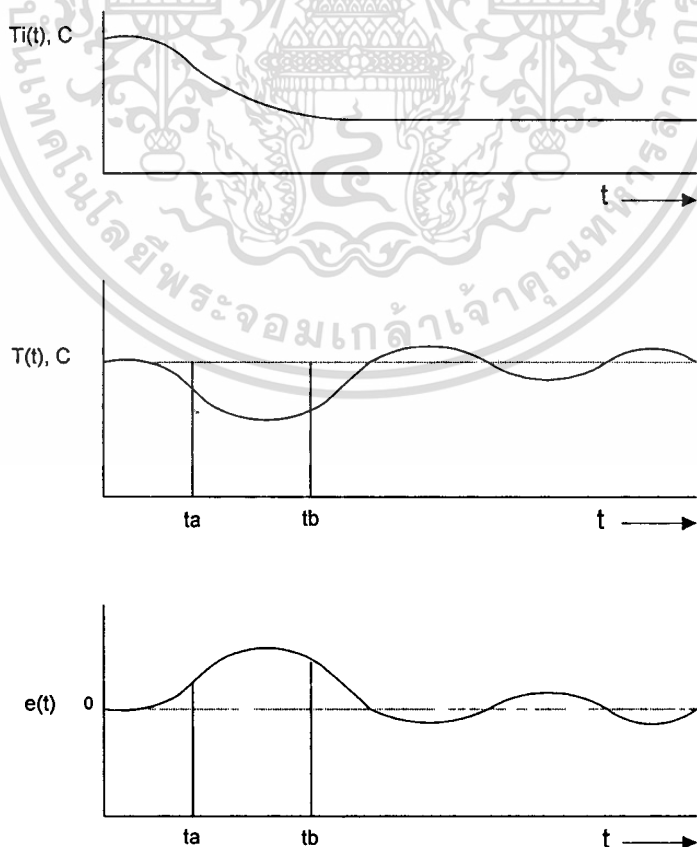
ควบคุมแบบ PI ก็คือ เดอริเวทีฟ หรืออาจเรียกกริยาของอัตราการเปลี่ยนแปลง rate หรือ peract จุดประสงค์ก็เพื่อที่จะเป็นการคาดการณ์ล่วงหน้าเมื่อกระบวนการทำงาน โดยมองที่อัตราการเปลี่ยนแปลงต่อเวลาของความผิดพลาด ซึ่งมันเป็นค่าของอนุพันธ์ สมการที่ใช้อธิบายเป็นดังต่อไปนี้

$$m(t) = \bar{m} + K_c e(t) + \frac{K_c}{\tau_i} \int e(t) dt + K_c \tau_D \frac{de(t)}{dt} \quad (1.22)$$

เมื่อ  $\tau_D$  = เดอริเวทีฟ หรือ อัตราหน่วยเป็นเวลา, นาที

เพราะฉะนั้น เครื่องควบคุมแบบ PID จะมีสามพารามิเตอร์ที่จะต้องปรับแต่งเพื่อให้การควบคุมนั้นเหมาะสม ส่วนค่าเวลาเดอริเวทีฟจะเป็นหน่วยของเวลาที่ทุกผู้ผลิตต่างใช้เป็นแบบเดียวกันและจากที่กล่าวมากริยาเดอริเวทีฟจะทำให้เครื่องควบคุมมีความสามารถในการคาดเดาเหตุการณ์ล่วงหน้าได้ด้วยการมองเหตุการณ์ไปข้างหน้า โดยการคำนวณค่าอนุพันธ์ของความผิดพลาด จำนวนของค่าคาดการณ์จะมากน้อยเท่าใดก็จะขึ้นอยู่กับพารามิเตอร์ของเวลาเดอริเวทีฟที่ได้ถูกปรับแต่ง

ต่อไปจะพิจารณากระบวนการแลกเปลี่ยนความร้อนและใช้เพื่อให้ความหมายของค่าว่าการคาดเดาเหตุการณ์ล่วงหน้า สมมุติว่าอุณหภูมิทางเข้าของของเหลวในกระบวนการลดลงไปจำนวนหนึ่งและอุณหภูมิทางออกเริ่มที่จะลดลงที่จะได้แสดงในรูปที่ 1-6



รูปที่ 1-6 การควบคุมในกระบวนการแลกเปลี่ยนความร้อน

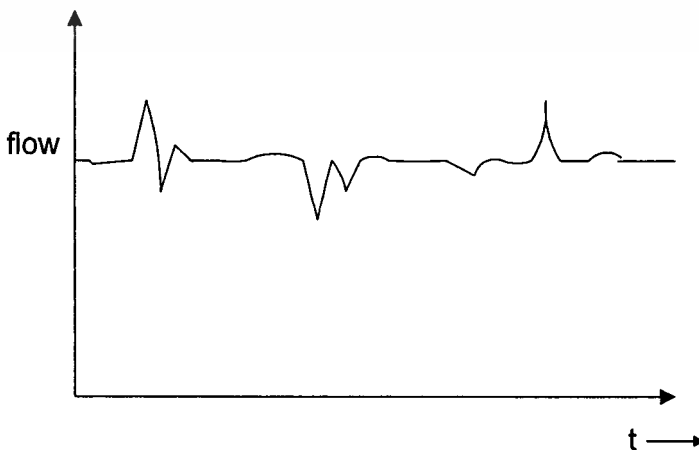
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นิยามให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ที่เวลา  $t_a$  จำนวนของค่าผิดพลาดเป็นบวกแต่อาจจะเล็กน้อย จำนวนของการแก้ไขค่าผิดพลาดของระบบควบคุมโดยเฉพาะค่าการแก้ไขแบบสัดส่วนและอินทิกรัลจะน้อย อย่างไรก็ตามค่าเดริเวทีฟของความผิดพลาดอันนี้โดยสังเกตจากความชันของความผิดพลาดจะมีค่ามากและเป็นบวกจะเป็นผลให้ค่าการแก้ไขเฉพาะเดริเวทีฟนั้นมีค่ามาก เครื่องควบคุมจะรู้ว่าตัวแปรที่ถูกควบคุมกำลังมุ่งหน้าออกจากเป้าหมายด้วยความเร็ว และที่เวลา  $t_b$  ค่าผิดพลาดก็ยังคงเป็นบวกและมากขึ้นกว่าเดิม จำนวนของค่าแก้ไขของแบบสัดส่วนและแบบอินทิกรัลนั้นก็จะมีมากกว่าแต่ก่อนและก็ยังคงรวมเข้าไปกับค่าแก้ไขเพื่อที่จะเปิดวาล์วให้กว้างขึ้น แต่ค่าเดริเวทีฟของความผิดพลาดที่เวลานี้จะเป็นลบ เป็นผลให้ค่าที่แก้ไขก่อนหน้านี้ถูกลดลง นั่นคือตัวแปรที่ถูกควบคุมได้เริ่มที่จะกลับมาสู่ค่าเป้าหมาย ที่จุดนี้เองค่าเดริเวทีฟก็เริ่มที่หักค่าแก้ไขของตัวมันเองออกจากค่าแก้ไขทั้งหมด เมื่อเป็นเช่นนี้ก็จะต้องใช้เวลาที่ยาวนานขึ้นเพื่อที่จะทำให้ค่าตัวแปรที่ถูกควบคุมกลับเข้าหาเป้าหมายอย่างไรก็ตามการพุ่งเลขขอบเขตและการแกว่งรอบค่าเป้าหมายจะถูกลดลงไป

เครื่องควบคุมแบบ PID ถูกใช้ในกระบวนการที่มีค่าเวลาคงตัวที่ยาวนาน ยกตัวอย่างเช่นวงรอบการควบคุมของอุณหภูมิและความเข้มข้น กระบวนการที่มีค่าเวลาคงตัวที่สั้น (ค่าความจุที่น้อย) จะเร็วและอ่อนไหวต่อสัญญาณรบกวนในกระบวนการมาก กระบวนการเหล่านี้ได้แก่วงรอบการควบคุมของอัตราไหลและการควบคุมความดันของกระแสของเหลว (ของเหลวที่พุ่งเป็นลำ เช่น น้ำพุ) พิจารณาผลการบันทึกของอัตราการไหลดังรูปที่ 1-7 เมื่อนำโหมดเดริเวทีฟมาใช้โดยจะเห็นว่าเป็นการขยายผลต่อสัญญาณรบกวนเท่า นั้นเพราะว่าค่าเดริเวทีฟจะไวต่อการเปลี่ยนแปลงของสัญญาณรบกวนซึ่งจะมีค่ามาก กระบวนการที่มีเวลาคงตัวมาก (ค่าความจุที่มาก) โดยปกติจะเป็นแบบถูกหน่วงหรือช้าและไม่ไวต่อสัญญาณรบกวน อย่างไรก็ตามถ้าจะใช้เครื่องควบคุมแบบ PID กับกระบวนการการอุณหภูมิแล้วก็ต้องตรวจสอบอุปกรณ์ส่งสัญญาณที่ไม่เป็นแหล่งกำเนิดสัญญาณรบกวนเสียเอง

ฟังก์ชันถ่ายโอนของเครื่องควบคุมแบบ PID ในอุดมคติจะเป็นในสมการ (1.19) ดังต่อไปนี้



เอกสารนี้เป็นเอกสารที่รูปที่ 1-7 แสดงผลการบันทึกอัตราการไหล ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$m(t) - \bar{m} = Kc(e(t) - 0) + \frac{Kc}{\tau_i} \int (e(t) - 0) dt + Kc\tau_D \frac{d(e(t) - 0)}{dt} \quad (1.23)$$

โดยใช้ข้อกำหนดที่แล้วมาของตัวแปรที่ถูกเบี่ยงเบนโดยสมการ (1.9) และ (1.10) แปลงลาปลาซ และจัดรูปจะได้

$$\frac{M(s)}{E(s)} = Kc \left( 1 + \frac{1}{\tau_i s} + \tau_D s \right) \quad (1.24)$$

ฟังก์ชันถ่ายโอนนี้ถูกเรียกว่าเป็นอุดมคติเพราะว่าในทางปฏิบัติในค่าของเดอริเวทีฟเป็นไปไม่ได้ที่จะทำการคำนวณหา ดังนั้นค่าเดอริเวทีฟจะถูกประมาณโดยใช้เทอมของ นำหน้า/ล่าหลัง ผลที่ได้จะเป็นฟังก์ชันถ่ายโอนที่ใกล้เคียงและแม่นยำกว่า

$$\frac{M(s)}{E(s)} = Kc \left( 1 + \frac{1}{\tau_i s} \right) \left( \frac{\tau_D s + 1}{\alpha \tau_D s + 1} \right) \quad (1.25)$$

ค่าของ  $\tau_{DS}$  อยู่ในพิสัยระหว่าง 0.05 ถึง 0.1 รูปเครื่องควบคุมแบบ PID จะมีสามพารามิเตอร์ที่ต้องถูกปรับแต่ง ได้แก่ อัตราขยายหรือ Proportional Band เวลารีเซ็ทหรืออัตรารีเซ็ท อัตราต่อเวลา อัตราต่อเวลาหรือเดอริเวทีฟนี้ค่าเวลาจะเป็นหน่วยนาที่ เครื่องควบคุมแบบ PID จะถูกใช้กับกระบวนการที่มีเวลาคงที่ยาวนานและโดยไม่มีสัญญาณรบกวน ข้อดีของเดอริเวทีฟก็คือจะมีความสามารถในการคาดการณ์ข้างหน้าขณะที่กระบวนการหรือตัวแปรที่ถูกควบคุมกำลังมุ่งหน้าไป

### เครื่องควบคุมแบบสัดส่วนและเดอริเวทีฟ Proportional-Derivative

**Controller (PD)** เครื่องควบคุมแบบนี้จะถูกใช้ในกระบวนการที่ใช้เครื่องควบคุมแบบสัดส่วนอย่างเดียวแต่ต้องการค่าของการคาดการณ์ล่วงหน้าอีกน้อย แสดงสมการได้ดังนี้

$$m(t) = \bar{m} + Kc.e(t) + Kc.\tau_D \frac{de(t)}{dt} \quad (1.26)$$

และฟังก์ชันถ่ายโอนในอุดมคติคือ

$$\frac{M(s)}{E(s)} = Kc(1 + \tau_D s) \quad (1.27)$$

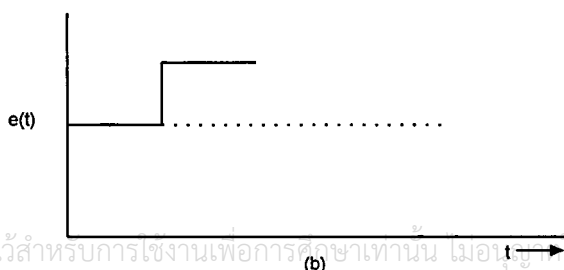
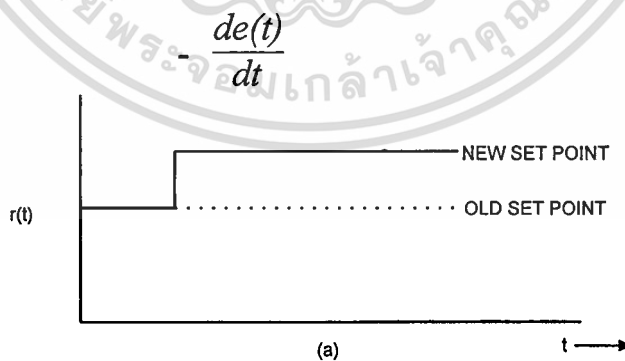
ข้อเสียของเครื่องควบคุมแบบ PD ไม่สามารถที่จะแก้ค่าผิดพลาดที่สภาวะคงที่ได้ จะเฉพาะกริยาอินทิกรัลเท่านั้นที่สามารถจะขจัดออฟเซ็ทให้หมดไปได้ อย่างไรก็ตามเครื่องควบคุมแบบ PD สามารถที่จะมีอัตราขยายที่สูงขึ้นกว่าแบบสัดส่วนอย่างเดียวจึงเป็นผลให้ออฟเซ็ทมีค่าน้อยไปได้ในวงรอบการควบคุมแบบง่าย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**เครื่องควบคุมแบบดิจิทัล Digital Controller** จากที่ได้กล่าวมาแล้วในสมการ (1.25) เป็นฟังก์ชันถ่ายโอนของเครื่องควบคุมในอุตสาหกรรมแบบอนาลอก อย่างไรก็ตามสมการของเครื่องควบคุมแบบดิจิทัลที่เป็นแบบที่ไม่ต่อเนื่องในสมการ (1.19) วิธีการในการปรับแต่งจะไม่ได้แตกต่างจากการปรับแต่งเครื่องควบคุมแบบอนาลอก

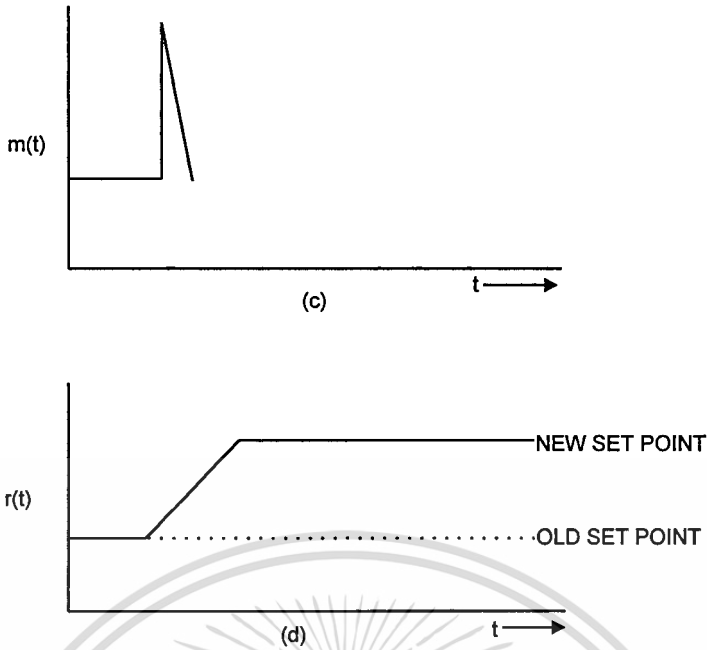
มีเรื่องราวของเครื่องควบคุมอีกเล็กน้อย ที่จะได้อธิบายต่อไปก่อนที่จะจบในส่วนนี้ได้แก่สมการ (1.19) แสดงให้เห็นว่าทุกเวลาที่มีการเปลี่ยนแปลงค่าอัตราขยายแล้วก็จะมีผลกระทบต่อกริยาอินทิกรัลและเดอริเวทีฟเพราะ  $\tau_i, \tau_d$  ถูกคูณและหารโดยค่าอัตราขยายหมายความว่าถ้าต้องเปลี่ยนแปลงเฉพาะอัตราขยายอย่างเดียวโดยไม่ต้องให้มีผลกับอินทิกรัลและเดอริเวทีฟแล้ว ก็จะต้องเปลี่ยนแปลงค่าทั้งอินทิกรัลและเดอริเวทีฟทุกค่าของอัตราขยายที่ถูกเปลี่ยนไปซึ่งเครื่องควบคุมแบบอนาลอกจะเป็นแบบนี้ทั้งหมด บางครั้งเครื่องควบคุมดังกล่าวนี้ถูกเรียกว่า **Interacting Controller** เครื่องควบคุมที่ใช้ไมโครโปรเซสเซอร์โดยทั่วไปจะเป็นแบบนี้ด้วยอย่างไรก็ตามจะมีเครื่องควบคุมบางแบบที่แก้ปัญหานี้โดยการแทน  $Kc/\tau_i$  ด้วยเทอม  $Ki$  และ  $Kc \cdot \tau_d$  ด้วยเทอม  $Kd$  และพารามิเตอร์ทั้งสามคือ  $Kc, Ki, Kd$

สุดท้ายเป็นความสัมพันธ์ของกริยาเดอริเวทีฟ ในทางหนึ่งที่จะเปลี่ยนแปลงค่าเป้าหมายของเครื่องควบคุมโดยการเปลี่ยนแปลงในรูปที่ 1-8a เมื่อเป็นแบบการเปลี่ยนแปลงแบบเป็นขั้นในค่าผิดพลาดแสดงในรูปที่ 1-8b ดังนั้นเครื่องควบคุมจะต้องให้ค่าเดอริเวทีฟต่อความผิดพลาดนี้ และกริยาเดอริเวทีฟนี้จะมีการเปลี่ยนแปลงในทันทีที่ทางออกของเครื่องควบคุมที่แสดงในรูปที่ 1-8c การเปลี่ยนแปลงที่ทางออกนี้ไม่เป็นที่ต้องการและอาจก่อให้เกิดความเสียหายต่อการทำงานของกระบวนการ เพื่อที่จะแก้ปัญหานี้จะใช้เพียงค่าลบของอนุพันธ์ในตัวแปรที่ถูกควบคุมเป็นค่าเดอริเวทีฟต่อความผิดพลาดดังกล่าว



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษเท่านั้น ไม่นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 1-8 แสดงผลกระทบท่อการเปลี่ยนแปลงของค่าเป้าหมาย

ค่าอนุพันธ์ของทั้งสองจะเท่ากันเมื่อค่าเป้าหมายยังคงที่สามารถแสดงได้ดังต่อไปนี้

$$\frac{de(t)}{dt} = \frac{d[r(t) - c(t)]}{dt} = \frac{dr(t)}{dt} - \frac{dc(t)}{dt} \quad (1.28)$$

ขณะนี้ค่าเป้าหมายได้เปลี่ยนเป็นค่าใหม่แล้วค่าเดอริเวทีฟไม่ได้สร้างกรียาทันทีได้ออกมา ในขณะที่เดียวกันพฤติกรรมต่างๆ ก็ยังคงเหมือนแต่ก่อน การเลือกที่จะใช้เครื่องควบคุมแบบอนาล็อกหรือแบบไมโครโปรเซสเซอร์จะต้องคำนึงถึงกรียาเดอริเวทีฟที่จะเป็นผลต่อตัวแปรที่ถูกควบคุมด้วย

### รีเซ็ต-วินด์อัฟ

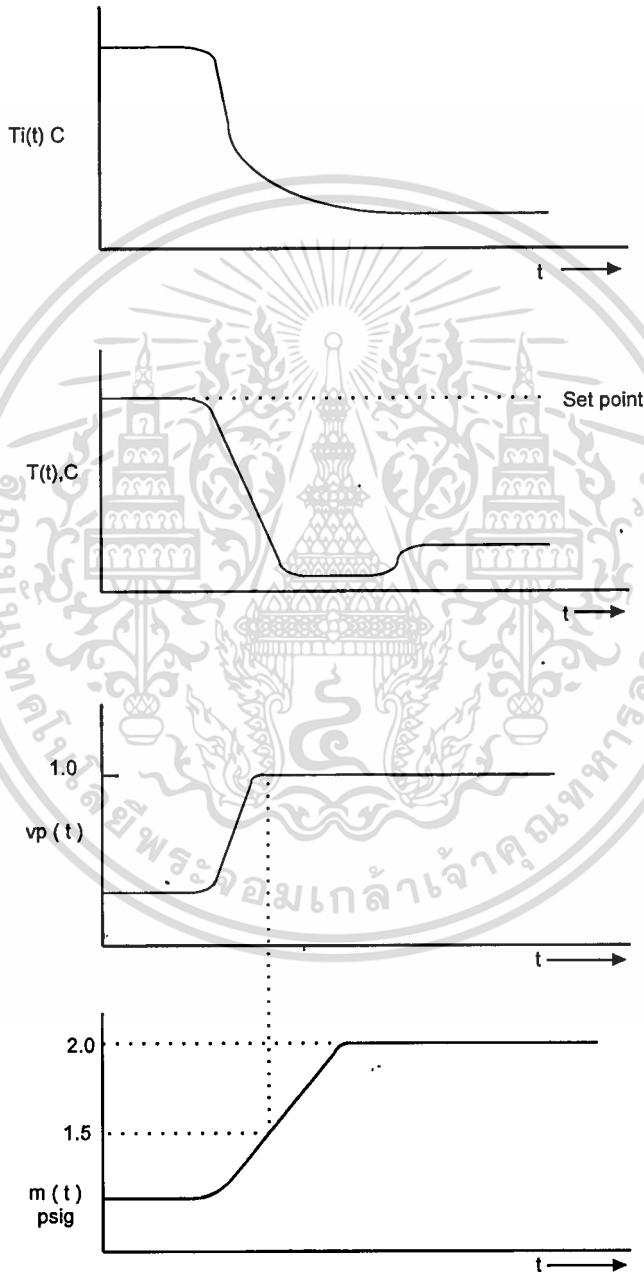
ปัญหา รีเซ็ต-อัฟนี้เป็นปัญหาที่สำคัญและเกิดขึ้นจริงในกระบวนการ โดยเกิดได้ทุกเวลาที่เครื่องควบคุมมีการใช้โหมดควบคุมแบบอินทิกรัลร่วมอยู่ด้วย วงรอบการควบคุมของกระบวนการแลกละเปลี่ยนความร้อนที่แสดงในรูปที่ 5-8 ถูกใช้เพื่อแสดงให้เห็นปัญหาอันนี้

สมมติว่าอุณหภูมิของเหลวที่ทางเข้าของกระบวนการลดลงไปจำนวนหนึ่ง สิ่งรบกวนอันนี้จะกระทบไปจนถึงอุณหภูมิที่ทางออกของกระบวนการด้วย เครื่องควบคุม PI และ PID ก็จะทำกรเปิดวาล์วไอน้ำให้กว้างขึ้น โดยเพิ่มสัญญาณแรงดันลมให้กับวาล์วที่เป็นแบบใช้แรงดันเพื่อเปิด จนอุณหภูมิทางออกเพิ่มขึ้นจนเท่ากับค่าเป้าหมายที่ต้องการ แต่ถ้าสมมติว่าถ้าตัวแปรที่ถูกควบคุมยังไม่เพิ่มขึ้นเท่าใด เครื่องควบคุมก็จะต้องอินทิเกรตค่าผิดพลาดไปจนที่ทางออกเป็น 15 psig ที่จุดนี้วาล์วไอน้ำก็จะเปิดกว้างสุดเพราะฉะนั้นวงรอบการควบคุมไม่สามารถที่จะทำอะไรได้อีกแล้ว กระบวนการอยู่นอกเหนือจากการควบคุมดังแสดงเป็นกราฟในรูปที่ 1-9 ในรูปนี้แสดงให้เห็นว่าเมื่อวาล์วถูกเปิดสุด ตัวแปรที่ถูกควบคุมหรืออุณหภูมิทางออกก็ยังคงไม่เข้าหาค่าเป้าหมาย ก็ยังคง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่ออนุญาตให้เผยแพร่ไปยังกระดานการคำ

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

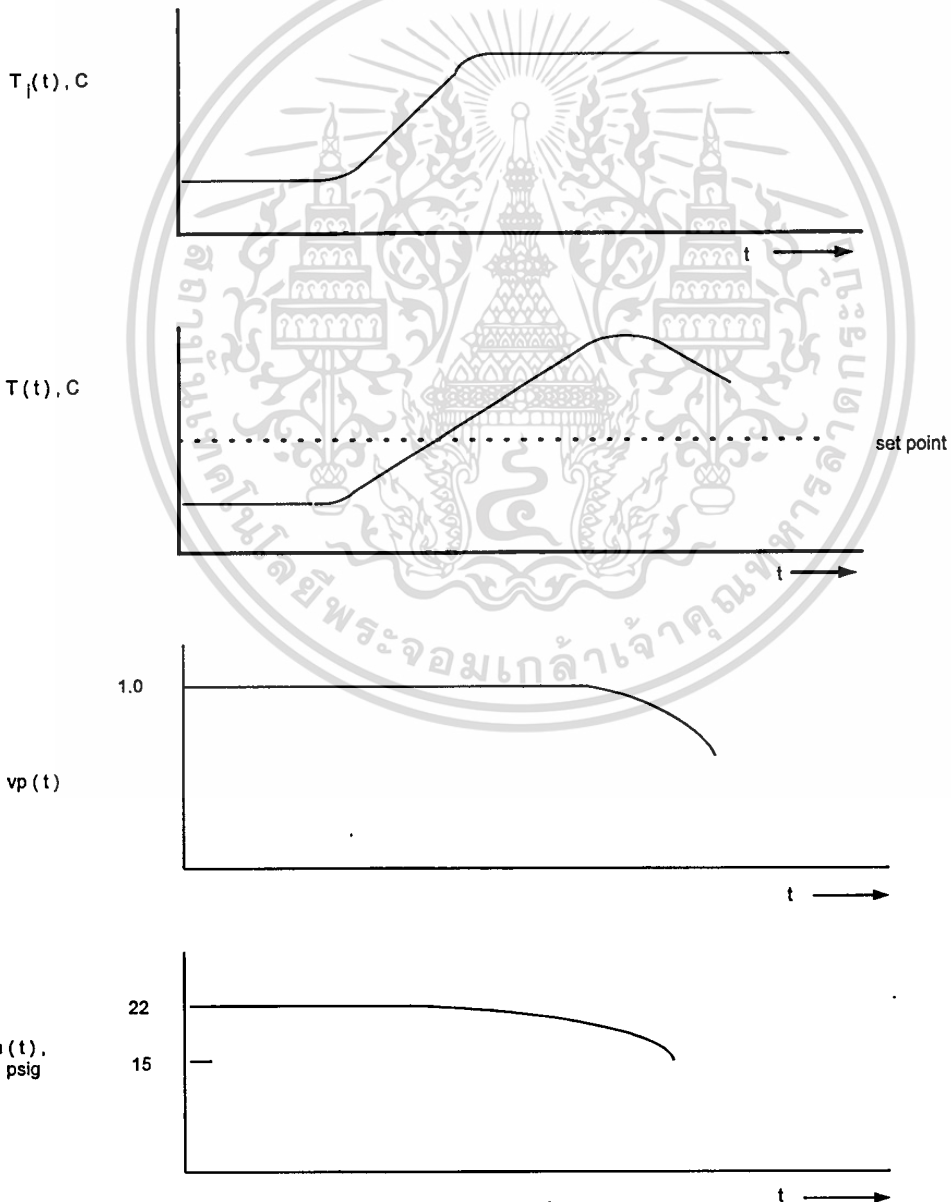
มีค่าผิดพลาดอยู่ตลอด เครื่องควบคุมก็พยายามที่จะแก้ (โดยการอินทิเกรตค่าผิดพลาด) ออกที่ทางออกมากขึ้นแต่ว่าวาล์วไม่สามารถที่จะเปิดกว้างกว่านี้ได้แล้ว และที่ทางเครื่องควบคุมจะอินทิเกรตมากขึ้นจนไปถึงที่แรงดันแหล่งจ่ายซึ่งโดยปกติจะประมาณ 20 psig ที่จุดนี้เองเครื่องควบคุมไม่สามารถที่จะเพิ่มค่าที่ทางออกได้มากกว่านี้ทางออกที่จุดอิมิต์วนี้ได้แสดงไว้ในรูปที่ 1-9 ด้วย จุดอิมิต์วจะเป็นผลมาจากกรียาอินทิกรัลของเครื่องควบคุมที่ผิดพลาดได้เกิดขึ้นเป็นเวลานาน และสภาวะอิมิต์วนี้จึงถูกเรียกว่า รีเซ็ท - วินด์อัฟ



รูปที่ 1-9 แสดงผลตอบสนองในวงรอบการควบคุมของกระบวนการแลกเปลี่ยนความร้อน และในขณะนี้นสมมติว่าอุณหภูมิทางเข้าได้กลับมาตามเดิมอุณหภูมิทางออกของกระบวนการก็เริ่มต้นที่จะเพิ่มขึ้นแสดงในรูปที่ 1-10 รูปนี้แสดงอุณหภูมิทางออกเพิ่มขึ้นเข้าสู่ค่าเป้าหมายและผ่านเลยไป

เอาไว้แล้ววาล์วก็ยังคงเปิดกว้างสุดอยู่ ซึ่งในความเป็นจริงแล้วน่าจะเปิดไปได้แล้ว แต่เหตุผลที่มันไม่ปิดก็ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เพราะว่าเครื่องควบคุมจะต้องอินทรีเกรดค่าที่ค้างอยู่เดิมออกเสียก่อนจาก 20 *psig* ไปเป็น 15 *psig* จากนั้นถึงจะเริ่มถึงที่จะปิดวาล์วได้ ในเวลานี้เองทำให้อุณหภูมิทางออกเกิดการพุ่งเลขขอบเขตของค่าเป้าหมายไปจำนวนหนึ่ง การแก้ไขในกรณีรีเซ็ต-วินด์อัฟ โดยการเปลี่ยนโหมดการควบคุมไปที่ควบคุมด้วยมือเครื่องควบคุมก็จะหยุดการอินทรีเกรดจากนั้นก็เปลี่ยนโหมดการควบคุมกลับไปใช้อัตโนมัตเพื่อเริ่มที่จะลดค่าที่ทางออกต่อไป ข้อเสียของการกระทำแบบนี้ก็คือไม่มีการเตือนว่ากรณีนี้จะเกิดขึ้นเมื่อใด อย่างไรก็ตามเครื่องควบคุมส่วนมากจะมีอุปกรณ์ป้องกันในกรณีเกิดรีเซ็ต-วินด์อัฟ ไว้ด้วย การป้องกันโดยอัตโนมัติจะหยุดการอินทรีเกรดของเครื่องควบคุมถ้าค่าที่ทางออกเข้าสู่ 15 *psig* (20*mA*) หรือ 3 *psig* (4*mA*) รีเซ็ต-วินด์อัฟ มักจะเกิดกับกระบวนการแบบแบทช์ (batch process) หรือในการควบคุมแบบแคสเคด (cascade control) และเมื่ออุปกรณ์ควบคุมขั้นสุดท้ายถูกควบคุมจากเครื่องควบคุมที่มากกว่าหนึ่ง



เอกสารนี้รูปที่ 1-10 แสดงผลตอบสนองในวงรอบการควบคุมของกระบวนการแลกเปลี่ยนความร้อน  
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทที่ 2

# PID-FUZZY CONTROLLER

## FUZZY CONTROL คืออะไร

ระบบ Fuzzy Control เป็นระบบที่เรียกว่า rule-based system ซึ่งหมายความว่า จะมี Fuzzy rules หรือ กฎของ Fuzzy เป็นตัวตัดสินใจกลไกการทำงานของระบบควบคุมนั้นให้มีการปรับเปลี่ยนไปตามผลกระทบที่เกิดขึ้นในระบบ จุดมุ่งหมายของระบบ Fuzzy Control ก็คือการให้ Fuzzy rule-based system นี้เข้าไปแทนการควบคุมแบบเก่าซึ่งใช้มนุษย์เป็นผู้ควบคุม (skilled human operator) plant นั้น ๆ Fuzzy Control จะมีประโยชน์ในสถานการณ์ที่

- ไม่มี mathematical model ที่ยอมรับได้สำหรับ Plant นั้น ๆ
- เดิมมีการใช้คนที่มีความชำนาญเป็นผู้ควบคุมกระบวนการ ซึ่งต้องอาศัยกฎของการควบคุมที่ไม่ชัดเจนและคลุมเครือ

## ทำไมจึงต้องใช้ FUZZY CONTROL ?

เหตุผลที่ต้องใช้ Fuzzy Control นั้นแบ่งได้เป็น 2 ส่วนใหญ่ ๆ คือ เหตุผลทางทฤษฎีและเหตุผลทางปฏิบัติ

### 1.เหตุผลทางทฤษฎีสำหรับ Fuzzy Control

- โดยทั่วไปแล้วหลักการทางวิศวกรรมที่คตินั้น ควรจะต้องใช้ข้อมูลที่มีอยู่อย่างมีประสิทธิภาพสูงสุดเมื่อ Mathematical model ของระบบนั้นยากเกินไปที่จะหามาได้ (ซึ่งเป็นไปได้มากในระบบที่ใช้ในการปฏิบัติจริงทั่วไป) ข้อมูลที่สำคัญที่สุดก็จะมาจาก

1. Sensor ซึ่งจะวัดค่าตัวแปรที่สำคัญออกมาเป็นตัวเลข

2. ผู้เชี่ยวชาญ ซึ่งจะอธิบายระบบออกมาเป็นภาษาของมนุษย์ (linguistic

description) และ Control Instruction

-Fuzzy Control นั้นสามารถออกแบบโดยนำข้อมูลจากผู้เชี่ยวชาญ ซึ่งมีลักษณะไม่ชัดเจนนั้น มาใช้ได้อย่างมีระบบและมีประสิทธิภาพ ซึ่ง Controller แบบเก่านั้นไม่สามารถนำข้อมูลแบบภาษาของมนุษย์ (linguistic description) มาใช้ได้ จึงทำให้ในสถานการณ์ที่ข้อมูลที่สำคัญที่สุดได้มาจากผู้เชี่ยวชาญนั้น Fuzzy Controller จะเป็นทางเลือกที่ดีที่สุด

-Fuzzy Control เป็น model-free approach กล่าวคือไม่ต้องการ mathematical model ของระบบเหล่านั้นก็ได้มาขึ้นตามไปด้วย ดังนั้น model-free approach จึงมีความสำคัญมากขึ้น การ Control แบบเก่านั้นก็มีบ้างเหมือนกันที่เป็น model-free approach เช่น PID Control

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

-Fuzzy Control นั้นเป็น nonlinear controller ซึ่งถูกปรับแต่งโดยทฤษฎี universal approximation Fuzzy logic controller นั้น สามารถปรับแต่งให้ทำงานเป็น nonlinear control action ใด ๆ ก็ได้ ดังนั้นถ้าเลือก parameter ของ Fuzzy controller ได้เหมาะสม ก็จะ สามารถควบคุมระบบ nonlinear ได้

## 2.เหตุผลทางการปฏิบัติสำหรับ FUZZY CONTROL

-เข้าใจได้ง่าย เนื่องจาก Fuzzy control เป็นการเลียนแบบวิธีการควบคุมของ มนุษย์ซึ่งหลักการของ Fuzzy control ก็สามารที่จะเข้าใจได้ง่ายดายแม้แต่โดยบุคคลซึ่งไม่ใช่ผู้ เชี่ยวชาญ ใน 2 ทศวรรษที่ผ่านมา ทฤษฎีการควบคุมแบบเก่าได้มีการใช้ mathematical tools ที่ซับซ้อนมากขึ้นเรื่อย ๆ ซึ่งก็เป็นสิ่งจำเป็นในการแก้ปัญหาที่ยาก อย่างไรก็ตามสิ่งเหล่านี้ก็ทำให้จำนวน ของวิศวกรปฏิบัติงานผู้เข้าใจในทฤษฎีเหล่านี้ลดน้อยลงไปด้วย ดังนั้นวิศวกรปฏิบัติงานที่ทำงานใน การออกแบบสินค้าอุปโภคบริโภค ก็มีแนวโน้มที่จะใช้วิธีซึ่งใช้งานและเข้าใจได้ง่าย ซึ่ง Fuzzy control ก็นับว่าเป็นวิธีการหนึ่งนั่นเอง

-Implement ได้ง่าย ระบบ Fuzzy logic ซึ่งเป็นหัวใจของ Fuzzy Control สามารถ Implement ได้ไม่ยากนัก ปัจจุบันมี Fuzzy VLSI Chip เกิดขึ้นมากมาย ซึ่งก็จะทำให้ใน การสร้าง Fuzzy Controller ทำได้ง่ายและรวดเร็ว

-ใช้ค่าใช้จ่ายไม่มากนักในการพัฒนา ในทฤษฎีทางการปฏิบัตินั้น ค่าใช้จ่ายใน การพัฒนานับว่าเป็นปัจจัยสำคัญที่จะบ่งชี้ความสำเร็จของผลิตภัณฑ์เนื่องจาก Fuzzy Control นั้น สามารถทำความเข้าใจได้ง่าย ใช้เวลาในการเรียนรู้ไม่มากนัก ทำให้ software cost ต่ำและจากการที่ Fuzzy control สามารถ Implement ได้ง่าย จึงทำให้ hardware cost ต่ำด้วยเช่นกัน นอกจากนี้ยังมี software tools สำหรับการออกแบบ Fuzzy controller อีกด้วย ดังนั้น Fuzzy control จึงเป็นวิธีการ ที่มีอัตราส่วนระหว่างประสิทธิภาพการทำงานต่อราคาสูงมากทีเดียว

## ทฤษฎี FUZZY CONTROL

ระบบควบคุมฟัซซี่ (FuzzyControl) มีพื้นฐานแนวความคิดมาจากทฤษฎีฟัซซี่ลอ จิก (Fuzzy Logic) และทฤษฎีฟัซซี่เซต (Fuzzy Set) ฟัซซี่ลอจิกมีพื้นฐานอยู่บนทฤษฎีฟัซซี่เซต ซึ่งจะช่วยให้สามารถอธิบายการปฏิบัติและกฎการควบคุมของระบบเป็นคำพูดได้อย่างชัดเจนขึ้น หลักสำคัญของทฤษฎีฟัซซี่เซต(Fuzzy Set) คือ ขอมรับสมาชิกที่มีลักษณะตามเซตเพียงบางส่วนเข้า มาเป็นสมาชิก ซึ่งแตกต่างจากทฤษฎีเซตดั้งเดิม (CRISP SET) ทฤษฎีเซตดั้งเดิมจะเน้นชัดเจนเลย ว่าเป็นสมาชิกของเซตหรือไม่เท่านั้น ไม่มีการเป็นสมาชิกของเซตเพียงบางส่วน ต่อไปเราจะกล่าว ถึงรายละเอียดของทฤษฎีข้างต้นและการนำทฤษฎีดังกล่าวมาประยุกต์ใช้ในการควบคุมระบบตาม ลำดับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## คลิซเซต (CRISP SET)

ทฤษฎีคลิซเซต (CRISP SET) เป็นทฤษฎีเซตที่เราคุ้นเคยกันดี โดยหลักการพื้นฐานของคลิซเซต (CRISP SET) มีดังนี้

### 1. การเป็นสมาชิกของเซต

“ถ้า  $x$  เป็นสมาชิกของเซต  $A$ ” เราจะใช้สัญลักษณ์

$$x \in A$$

“ถ้า  $x$  ไม่เป็นสมาชิกของเซต  $A$ ” เราจะใช้สัญลักษณ์

$$x \notin A$$

ซึ่งในคลิซเซต (CRISP SET) นี้ เราจะระบุชัดเจนว่าสมาชิกของ Universe เป็นสมาชิกของเซต  $A$  หรือไม่ เราสามารถแทนค่าระดับการเป็นสมาชิก (Membership function) ซึ่งแทนด้วยสัญลักษณ์  $\mu_A$  ดังนี้

$$\mu_A(x) = 1 \quad \text{ก็ต่อเมื่อ} \quad x \in A$$

$$\mu_A(x) = 0 \quad \text{ก็ต่อเมื่อ} \quad x \notin A$$

### 2. การเป็นสับเซต

“ถ้าสมาชิกทุกตัวของเซต  $A$  เป็นสมาชิกของเซต  $B$  จะเรียกว่า เซต  $A$  เป็นสับเซตของเซต  $B$ ”  
เราใช้สัญลักษณ์การเป็นสับเซตดังนี้

$$A \subseteq B$$

### 3. การเท่ากันของเซต

“ถ้าสมาชิกทุกตัวของเซต  $A$  เท่ากับสมาชิกทุกตัวของเซต  $B$  จะเรียกว่า เซต  $A$  เท่ากับเซต  $B$ ”  
เราใช้สัญลักษณ์การเท่ากันดังนี้

$$A = B$$

### 4. การไม่เท่ากันของเซต

“ถ้าสมาชิกทุกตัวของเซต  $A$  ไม่เท่ากับสมาชิกทุกตัวของเซต  $B$  จะเรียกว่า เซต  $A$  ไม่เท่ากับเซต  $B$ ”

เราใช้สัญลักษณ์การไม่เท่ากันดังนี้

$$A \neq B$$

### 5. การเป็นสับเซตแท้

“ถ้าสมาชิกทุกตัวของเซต  $A$  เป็นสับเซตของเซต  $B$  และเซต  $A$  ไม่เท่ากับเซต  $B$  จะเรียกว่า เซต  $A$  เป็นสับเซตแท้ของเซต  $B$ ”

เราใช้สัญลักษณ์การเป็นสับเซตแก้ดังนี้

$$A \subset B$$

6. การปฏิบัติการเบื้องต้นของเซต ได้แก่

• **Complement** เป็นการสร้างเซตใหม่โดยสมาชิกของเซตใหม่จะไม่เป็นสมาชิกของเซต A เลย

$$\text{Complement } A = A'$$

• **Union** เป็นการสร้างเซตใหม่โดยสมาชิกของเซตใหม่จะมีสมาชิกของทุกเซตรวมอยู่

• **Inter section** เป็นการสร้างเซตใหม่โดยสมาชิกของเซตใหม่จะมีสมาชิกซึ่งเป็นสมาชิกของทุกเซต

## ฟัซซีเซต (FUZZY SET)

ทฤษฎีฟัซซีเซต (Fuzzy Set) เป็นการรวมสมาชิกของหลาย ๆ คลิชเซต (Crisp Set) ที่มีอยู่จริงและพอจะมีลักษณะเข้ารวมกลุ่มได้ ฟัซซีเซต(Fuzzy Set)ยอมรับการเป็นสมาชิกของสมาชิกที่มีลักษณะตามเซตเพียงบางส่วนซึ่งจะมีการเปลี่ยนแปลงทีละน้อยระหว่างการมีคุณสมบัติของการเป็นสมาชิกอย่างครบถ้วน กับ ไม่มีคุณสมบัติของการเป็นสมาชิกเลย แม้ว่าจะไม่พร้อมกันก็ตาม จึงสามารถใช้ฟัซซีเซต(Fuzzy Set)ไปทำงานกับระบบคลิชเซตได้

### 1. การเป็นสมาชิกของฟัซซีเซต

ในระบบคลิชเซต (Crisp Set) จะกำหนดเพียงว่า  $x$  เป็นสมาชิกของเซต A หรือไม่เป็นสมาชิกของเซต A โดยแทนด้วยฟังก์ชันระดับการเป็นสมาชิก (Membership function) เป็น 0 หรือ 1 แต่ฟัซซีเซต (Fuzzy Set) จะยอมรับสมาชิกที่มีลักษณะที่ถูกเพียงบางส่วนและผิดเพียงบางส่วนไม่มีขอบเขตแน่นอน การประยุกต์ใช้ทฤษฎีฟัซซีเซตจะต้องแสดงค่าระดับ (DEGREE) ซึ่งเป็นค่าที่เป็นไปได้ที่จะเป็นสมาชิกของเซตหรือฟังก์ชันระดับการเป็นสมาชิก (Membership Function) โดยเราใช้สัญลักษณ์  $\mu$  แทนค่าระดับความเป็นสมาชิกซึ่งมีค่าระหว่าง 0 ถึง 1 แสดงดังนี้

$$\mu_A(x) \longrightarrow [0,1]$$

หมายความว่าระดับการเป็นสมาชิก (Grade of membership หรือ Degree of Membership) ของ  $x$  ใน Fuzzy เซต A อยู่ในช่วงทั้งหมดจาก 0 ถึง 1 เมื่อประยุกต์เข้ากับฟัซซีลอจิก (Fuzzy Logic) ค่า  $\mu$  จะถูกเรียกว่า ค่าความจริงที่แสดงค่าระดับขอบเขตของเซตคือ

$$0 \leq X \leq 1$$

โดยถ้าค่าสูงแสดงว่ามีความเป็นสมาชิกมาก, ถ้าค่าต่ำแสดงว่ามีความเป็นสมาชิกน้อย, ค่า 0 จะหมายถึงไม่เป็นสมาชิกเลยและ 1 คือเป็นสมาชิกอย่างสมบูรณ์ ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขอเน้นว่าฟังก์ชันระดับการเป็นสมาชิก (Membership function) ของฟัซซีเซต ถึงแม้จะมีค่าระหว่าง 0 ถึง 1 แต่ย่นำไปสับสนกับความน่าจะเป็น (Probability) ฟัซซีเซตเป็นรูปแบบหนึ่งของความไม่แน่นอน (Uncertainty) และโดยธรรมชาติ ฟัซซีเซต(Fuzzy Set)เป็นศาสตร์ที่ไม่มีความเกี่ยวข้องกับสถิติ

## 2.นิยามที่ฟัซซีเซต (Fuzzy Set) ขยายมาจากกลิตเซต(Crisp Set)

-ฟัซซีเซต(Fuzzy Set) จะว่างก็ต่อเมื่อฟังก์ชันระดับการเป็นสมาชิก(Membership Function) ของฟัซซีเซต(Fuzzy Set)นั้นเป็น 0 ตลอดทั้ง X

-ฟัซซีเซต(Fuzzy Set) A และ B จะเท่ากันก็ต่อเมื่อ  $\mu_A(x) = \mu_B(x)$  สำหรับทุก  $x$  ใน X เขียนแทนด้วย

$$A = B$$

-Subset หรือ Containment นิยามโดย

$$A \subset B \iff \mu_A(x) \leq \mu_B(x)$$

-Complementของฟัซซีเซต(Fuzzy Set) Aเขียนแทนด้วย  $A'$ นิยามโดย

$$\mu_{A'}(x) = 1 - \mu_A(x)$$

-Unionของฟัซซีเซต (Fuzzy set) A และ B ซึ่งมีฟังก์ชันระดับการเป็นสมาชิก (Membership Function)  $\mu_A(x)$  และ  $\mu_B(x)$  ตามลำดับสามารถแทนด้วย ฟัซซีเซต(Fuzzy Set) C กำหนดโดย

$$\mu_C(x) = \text{Max}[\mu_A(x), \mu_B(x)] \quad \text{หรืออาจเขียนย่อเป็น}$$

$$\mu_C = \mu_A \vee \mu_B$$

-Intersectionของฟัซซีเซต(Fuzzy Set) A และ B ซึ่งมีฟังก์ชันระดับการเป็นสมาชิก (Membership Function)  $\mu_A(x)$  และ  $\mu_B(x)$  ตามลำดับสามารถแทนด้วย ฟัซซีเซต(Fuzzy Set)Cกำหนดโดยฟังก์ชันระดับการเป็นสมาชิก(Membership Function)ของฟัซซีเซต(Fuzzy Set)C กำหนดโดย

$$\mu_C(x) = \text{Min}[\mu_A(x), \mu_B(x)] \quad \text{หรืออาจเขียนย่อเป็น}$$

$$\mu_C = \mu_A \wedge \mu_B$$

อย่างไรก็ตามไม่มีหลักตายตัวว่า การUnionต้องใช้Max Operation, การIntersection ต้องใช้Min Operationหรือการ Complement ต้องใช้  $\mu_{A'}(x) = 1 - \mu_A(x)$  ดังนั้นเพื่อให้Generalมากขึ้นจึงมีนิยามสัจพจน์(Axiom)ปฏิบัติการแต่ละชนิดดังนี้

-Complement C :  $[0,1] \longrightarrow [1,0]$

Axiom C1 :  $C(0)=1$  และ  $C(1)=0$  หมายความว่าสามารถครอบคลุมคลิซเซต (Crisp Set)

Axiom C2 : สำหรับทุกๆ  $a, b \in [0,1]$  ถ้า  $a < b$  แล้ว  $C(a) \geq C(b)$

ทุกๆการปฏิบัติที่จะถือว่าเป็นการComplement ได้อย่างน้อยที่สุดจะต้องสอดคล้องกับสังพจน์ทั้งสอง

**-Union U :  $[0,1] \times [0,1] \rightarrow [0,1]$**

Axiom U1:  $U(0,0)=0; U(0,1)=U(1,0)=U(1,1)=1$  นั้นหมายความว่า จะต้องสามารถครอบคลุมคลิซเซต(Crisp Set)

Axiom U2:  $U(a,b)=U(b,a)$ ; มีคุณสมบัติการสลับที่

Axiom U3: ถ้า  $a \leq a'$  และ  $b \leq b'$  แล้ว  $U(a,b) \leq U(a',b')$

Axiom U4:  $(U(a,b),c) = U(a,U(b,c))$ ; มีคุณสมบัติการจัดหมู่

ทุกๆการปฏิบัติที่จะถือว่าเป็นการUnion ได้อย่างน้อยที่สุดจะต้องสอดคล้องกับสังพจน์ทั้งสอง

**-Intersection I :  $[0,1] \times [0,1] \rightarrow [0,1]$**

Axiom I1:  $I(1,1)=1; I(0,1)=I(1,0)=I(0,0)=0$  นั้นหมายความว่า จะต้องสามารถครอบคลุมคลิซเซต(Crisp Set)

Axiom I2:  $I(a,b)=I(b,a)$ ; มีคุณสมบัติการสลับที่

Axiom I3: ถ้า  $a \leq a'$  และ  $b \leq b'$  แล้ว  $I(a,b) \geq I(a',b')$  หมายความว่า I เป็น

Monotonic

Axiom I4:  $I(I(a,b),c)=I(a,I(b,c))$ ; มีคุณสมบัติการจัดหมู่

ทุกๆการปฏิบัติที่จะถือว่าเป็นการ Intersection ได้อย่างน้อยที่สุดจะต้องสอดคล้องกับสังพจน์ทั้งสอง

## ฟัซซีลอจิก (FUZZY LOGIC)

ฟัซซีลอจิกจะมีการกระทำทางลอจิกเพื่อรวมค่าลอจิกต่าง ๆ ให้เป็นค่าฟัซซีลอจิก (Fuzzy Logic) คล้าย ตัวแปรในระบบลอจิก 2 ระดับ โดยจะต้องมีกำหนดลักษณะความหมายให้แตกต่างกันแต่ใช้คำสั่งทางลอจิกที่เหมือนกันคือ AND, OR และ NOT ความหมายต่าง ๆ ในระบบฟัซซี (Fuzzy System) ถูกกำหนดโดย L.A.Zadeh ผู้คิดค้นระบบฟัซซีลอจิก(Fuzzy Logic)

**การ AND ของฟัซซีลอจิก**

ตามคำจำกัดความของZadehคือค่าความจริงที่น้อยที่สุด(Minimum)นั่นคือสำหรับค่าฟัซซี

A และ B

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$\mu(A \text{ and } B) = \min(\mu A, \mu B)$$

### การ OR ของฟัซซี่ลอจิก

ตามคำจำกัดความของ Zadeh คือค่าความจริงเป็นค่าที่มากที่สุด (Maximum)

$$\mu(A \text{ or } B) = \max(\mu A, \mu B)$$

### การ NOT ของค่าฟัซซี่ลอจิก

$$\mu(\text{not } A) = 1 - \mu A$$

ซึ่งการกระทำทั้งสามนี้เป็นสมมูลของการกระทำในลอจิก 2 ระดับ สำหรับค่า  $\mu$  มีค่าอยู่ระหว่าง 0 ถึง 1

## การประยุกต์ใช้งานของฟัซซี่ลอจิก (FUZZY LOGIC APPLICATION)

โดยทั่วไปจะมีลักษณะ โครงสร้างพื้นฐานแบ่งได้ 3 ส่วนหลัก คือ

- การเปลี่ยนคลิซเซตเป็นฟัซซี่เซต (FUZZIER)
- กฎของการวินิจฉัย (INFERENCE MECHANISM THAT EMPLOYS RULES)
- การเปลี่ยนฟัซซี่เป็นคลิซ (DEFUZZIER)

ในการใช้กับระบบจะต้องเปลี่ยนให้อยู่ในรูปของฟัซซี่โดเมน (FUZZY DOMAIN) เคลื่อนย้ายประมวลผลข้อมูลแล้วเปลี่ยนกลับให้อยู่ในรูป คลิซโดเมน (CRISP DOMAIN) ตามเดิมซึ่งเหมือนกับการกระทำทางอนาลอก (ANALOG) คือ เปลี่ยนให้อยู่ในรูปของโดเมนความถี่ (FREQUENCY DOMAIN) จะง่ายกว่าโดเมนเวลา (TIME DOMAIN) ในระบบฟัซซี่ (FUZZY SYSTEM) กฎพื้นฐานสามารถอธิบายการทำงานของระบบในรูปของฟัซซี่ (FUZZY) ได้ง่าย ดังนั้นเราจะเปลี่ยนค่าอินพุตในรูปของคลิซ (CRISP) ไปอยู่ในฟัซซี่โดเมนมากกว่าจะเปลี่ยนจากฟัซซี่ไปอยู่ในรูปคลิซโดเมน (CRISP DOMAIN)

### กฎการวินิจฉัย (INFERENCE)

สำหรับกฎการวินิจฉัยค่าอินพุตและค่าความจริง จะสนับสนุนเงื่อนไขสำหรับการสร้างส่วนกำหนดรูปแบบฟัซซี่ที่ช่วงปกติฟัซซี่คอมไพเลอร์ จะสุ่มค่าของอินพุตและนำมาวินิจฉัยค่า เพื่อให้ได้ผลออกมาทางเอาต์พุตของระบบตามทฤษฎี ระบบจะรวมเอาค่าที่เป็นไปได้ทั้งหมดของอินพุต เพื่อนำมาวินิจฉัยและประมวลผล แต่จริง ๆ แล้วการครอบคลุมค่าเหล่านี้ไม่จำเป็นในการใช้งานปกติ

กฎการวินิจฉัยนั้นได้มาจากการสร้างความสัมพันธ์ระหว่าง Input และ Output ขึ้นมา เพื่อนำไปสู่ output ที่ต้องการ ซึ่งความสัมพันธ์ระหว่าง input และ output ก็คือ **RULE** นั้นเอง เช่น เมื่อให้ input ของ controller คือ error(er) และ change of error(ce) ของระบบป้อนกลับและให้ output ของ controller เป็น input ของ process หรือ control input (ci) เราจะสามารถสร้างความสัมพันธ์ได้ดังนี้คือ

*IF er = LN AND ce = LN THEN ci = LP*

*OR*

*IF er = SN AND ce = SN THEN ci = SP*

*OR*

- 
- 
- 

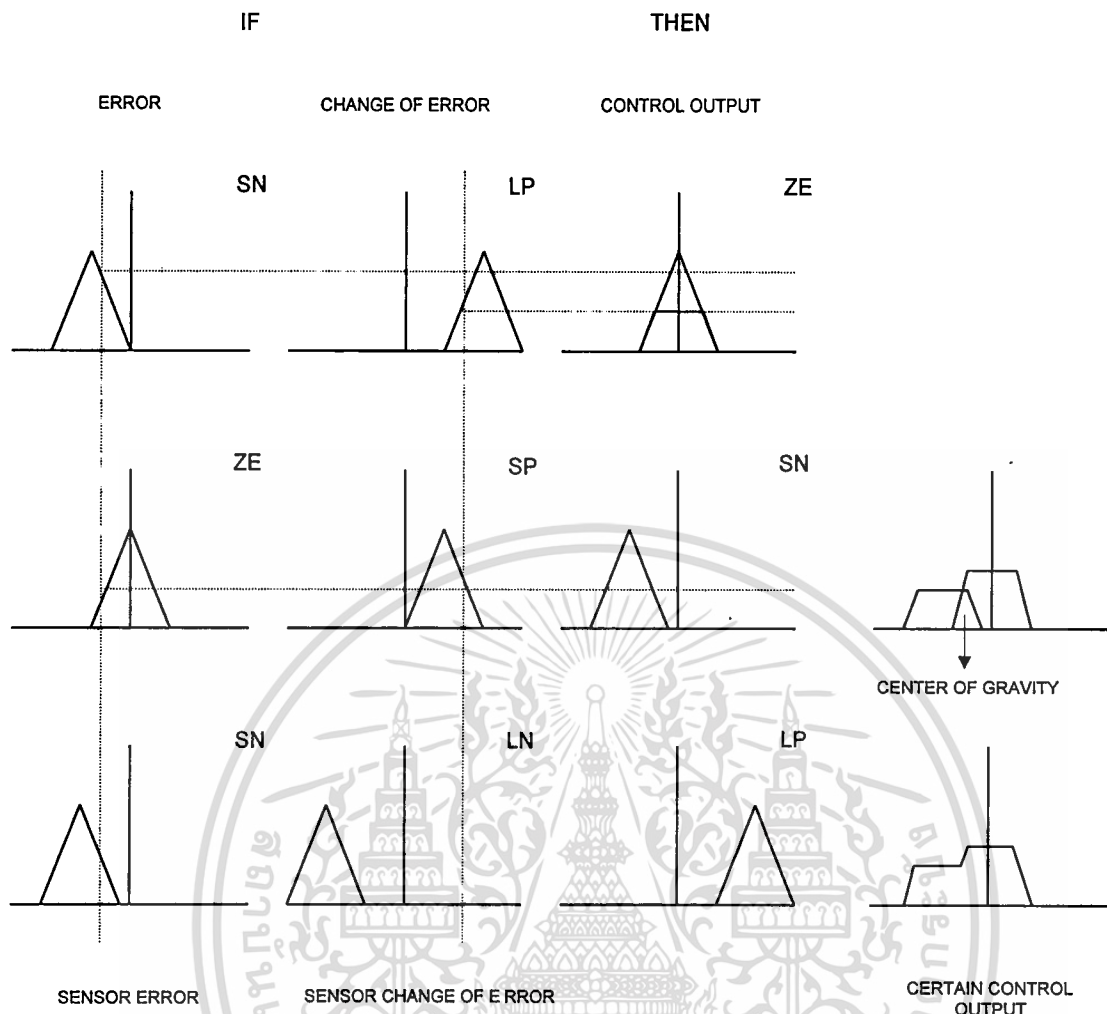
หมายเหตุ: LN = Large Negative

SN = Small Negative

LP = Large Positive

SP = Small Positive





รูปแสดงการ INFERENCE

ซึ่งเมื่อนำ rule ทั้งหมดมารวมเข้าด้วยกัน ก็จะได้รูป Membership Function ของ control input โดยจะนำไปหาค่าของ control input ที่ต้องการได้ต่อไป

### วิธีการ DEFUZZIFIER

มีเทคนิคและวิธีการในการเปลี่ยนฟัซซี่เป็นคลิช (DEFUZZIER) มีอยู่หลายเทคนิค ซึ่งจะกล่าวถึงเป็นบางเทคนิคดังนี้

#### 1. เทคนิค Maximizer เลือกค่าสูงสุดจากหลาย ๆ แบบมาเพียงหนึ่ง

เป็นการใช้ค่าสูงสุดของค่าระดับการเป็นสมาชิก จากการกระทำหลาย ๆ แบบ แล้วเลือกกระทำเพียง 1 รูปแบบ ถ้าหากเกิดการกระทำที่มีค่า  $\mu$  สูงสุดเท่ากัน 2 อย่าง จะต้องใช้รูปแบบการแก้ปัญหาอีกลักษณะหนึ่ง คือ ใช้ค่าเฉลี่ยของค่าเข้าพุท หรือเลือกการกระทำที่สัมพันธ์ กับค่าระดับของระบบพื้นฐาน ถึงแม้เทคนิค Maximizer จะเป็นวิธีที่ง่ายที่สุด แต่ก็ไม่มีประสิทธิภาพเท่าที่ควร

## 2. ทฤษฎีค่าน้ำหนักเฉลี่ย

จะใช้ค่าเฉลี่ยของการกระทำหลังจากการกำหนดค่าระดับของระดับการเป็นสมาชิกไว้ล่วงหน้าแล้ว เป็นวิธีที่ง่ายและใช้การคำนวณเพียงเล็กน้อย แต่ก็ยังให้ค่าที่ไม่ค่อยชัดเจน เช่นเดียวกับเทคนิค Maximizer ที่เกิดความไม่ชัดเจนก็เพราะว่าค่าของเข้าพุทของฟังก์ชันระดับการเป็นสมาชิก (MEMBERSHIP FUNCTION) มีค่าเข้าพุทมากกว่า 1 ค่า ต่อค่า  $\mu$  ที่กำหนดให้ ค่าเข้าพุทฟังก์ชันของการเป็นสมาชิก มีลักษณะคล้ายเป็นรูปปิรามิดหรือปิรามิดตัดยอด ถ้า  $\mu = 0.5$  ค่าเข้าพุทจากค่าฟังก์ชันของขอบสัญญาณทั้งด้านขาขึ้นและขาลง ถ้า  $\mu = 1$  จะมีค่าตรงกับช่วงของสัญญาณ ที่เกิดขึ้นทั้งหมด

วิธีการกำจัดความไม่ชัดเจนสามารถทำได้ด้วยกระบวนการแฉมบั้งค่าในฟังก์ชันเข้าพุทด้วยค่าที่แน่นอนในให้กลับไปอยู่ในฟังก์ชันอินพุท ซึ่งเป็นวิธีที่น่าเบื่อ และไม่สามารถใช้ค่าที่เป็นค่าตรงข้ามของฟังก์ชันอินพุทในการวิเคราะห์ด้วย

## 3. กรรมวิธีค่าศูนย์กลาง

เป็นการแสดงค่าเข้าพุทที่สัมพันธ์กับค่าจุดศูนย์กลางมวลของเข้าพุทในระดับที่ทำงาน เพราะเราไม่ใช้ค่าขอบของฟังก์ชันและระดับการเป็นสมาชิก และจะไม่เกิดความไม่ชัดเจนอีกต่อไปกรรมวิธีค่าศูนย์กลางเป็นการคำนวณที่แน่นอนและเป็นการแก้ไขข้อบกพร่องที่เกิดขึ้นในวิธีอื่นๆ

ค่าศูนย์กลางมักจะสัมพันธ์กับค่าเข้าพุท 1 ค่า ผลที่ได้จะเป็นการกระทำอย่างหยาบ ๆ ภายในช่วงของเข้าพุท กฎทั่ว ๆ ไป จะต้องมีการปฏิบัติเพียงจุดเดียวของแต่ละระบบ ในกรณีที่ต้องทำพร้อม ๆ กันหลายกฎ ในการกระทำหนึ่งครั้ง จะต้องเกิดการซ้อนทับกัน (OVERLAP) ของค่าฟังก์ชันอินพุท ของระดับการเป็นสมาชิก เพื่อแก้ไขความไม่ต่อเนื่องของเข้าพุท ถึงแม้ว่าจะมีข้อบกพร่องแต่ก็เป็นวิธีที่ดีที่สุดในการรวมกัน (COMBINATION) และการแปลงค่าฟัซซี่ กลับคืน(DEFUZZIFICATION)

กรรมวิธีนี้จะรวมค่าเข้าพุทของการกระทำหลาย ๆ ค่า เป็นค่าค่าเดียวสำหรับใช้ในระบบและค่าเข้าพุทค่าเดียนี้เป็นค่าน้ำหนักเฉลี่ยของศูนย์กลาง (CENTROID) ของแต่ละฟังก์ชันระดับการเป็นสมาชิก

## 4. กรรมวิธีซิงเกิลตัน : สังเคราะห์เอาเข้าพุทเดียว

เป็นเทคนิคพิเศษของแบบกรรมวิธีค่าศูนย์กลาง หรือจะเรียกอีกอย่างหนึ่งคือ กรรมวิธี REMAINING COMBINATION/DEFUZZIFICATION วิธีนี้เป็นการนำค่าเข้าพุทของและฟัซซี่ที่ชุดมาใช้ใหม่ เป็นค่าเข้าพุทค่าเดียวโดยใช้ค่าน้ำหนักเฉลี่ยจากการกระทำรวมกันหลาย ๆ อย่าง วิธีนี้ได้ค่าความถูกต้องน้อยกว่าค่ากรรมวิธีค่าศูนย์กลาง และยังคงต้องการการซ้อนทับของอินพุทฟังก์ชัน

เอกสารนี้เป็นเอกสารทศวันวิสาห์สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นิยญาติให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เพื่อเป็นการหลีกเลี่ยงความไม่ต่อเนื่องของเอาพุท ด้วยหลักการและการคำนวณที่ไม่ยุ่งยาก ประกอบกับยังไม่มีใครคิดค้นวิธีใหม่และดีกว่านี้ กรรมวิธีนี้จึงน่าที่จะใช้แทนกรรมวิธีค่าศูนย์กลางได้ดีที่สุด

## หลักการหา Fuzzy Control Rule

ในการออกแบบ fuzzy controller นั้น ปัญหาสำคัญอันหนึ่งคือการหา fuzzy control rule ที่เหมาะสม ซึ่งวิธีที่สามารถใช้ได้จริงในงาน process control อยู่ 3 วิธีด้วยกัน คือ

### 1. จากความรู้ และ ประสบการณ์ของผู้เชี่ยวชาญ

Fuzzy Controller ส่วนมากจะได้รับการออกแบบขึ้น โดยอ้างอิงกับ ความรู้และประสบการณ์ของวิศวกรควบคุมซึ่งอันที่จริงแล้วก็สามารถกล่าวได้ว่า Fuzzy Control นั้นเป็น Application ที่ใช้ได้จริงอันแรกทางด้านระบบผู้เชี่ยวชาญ (Expert System) เลยทีเดียว

การออกแบบด้วยวิธีนี้เหมาะกับระบบที่ผู้ควบคุม ควบคุมเป็นหลักที่สำคัญ Control Rule นั้นก็จะได้มาจากการ List วิธีการควบคุมต่าง ๆ โดยวิศวกรควบคุม แต่ข้อเสียของวิธีนี้ก็คือ ส่วนมากผู้ควบคุมจะไม่สามารถถ่ายทอดความรู้ออกมาเป็นสายลักษณะอักษร โดยเฉพาะเมื่อระบบมีความซับซ้อนมาก

### 2. จากการจำลองลักษณะการทำงานของผู้ควบคุม

เมื่อทักษะของผู้ควบคุมเป็นสิ่งสำคัญจึงจำเป็นมากที่จะหา Fuzzy Control Rule มาจากการจำลองลักษณะ การทำงานของผู้ควบคุม ซึ่งวิธีนี้สามารถเป็นไปได้ ที่จะแปลงการทำงานของผู้ควบคุมมาให้เป็น Input และ Output ของ Controller

วิธีนี้ค่อนข้างคล้ายกับการจำลองกระบวนการ แต่ว่าการจำลองการทำงาน ของผู้ควบคุมนั้น จะง่ายกว่าการจำลองกระบวนการ เพราะ Input ของระบบนั้นจะหาได้ง่ายกว่า แต่ในสถานการณ์จริงนั้นก็ควรจะรวมวิธีที่ 1 และ วิธีที่ 2 เข้าด้วยกัน

### 3. จากการจำลองกระบวนการ

ในวิธีแรกนั้นจะมีพื้นฐานมาจากความคิดคร่าว ๆ ที่เกี่ยวกับคุณลักษณะของกระบวนการ เช่น Output เพิ่มขึ้นเมื่อ Input เพิ่มขึ้น , กระบวนการที่มี Timelag ฯลฯ ส่วนวิธีที่ 2 จะใช้เฉพาะตัวแปรที่มีให้แก่ผู้ควบคุมกระบวนการ ซึ่งทั้ง 2 วิธี นี้จะใช้ดีเฉพาะในกรณีที่ผู้ควบคุมมีบทบาทสำคัญ ในการควบคุมกระบวนการเท่านั้น

แต่ถ้าไม่ต้องการขึ้นอยู่กับผู้ควบคุมและให้ผลการควบคุมดีกว่าการควบคุมด้วยผู้ควบคุม ก็จะมีการออกแบบอีกวิธีหนึ่งคือ การจำลองกระบวนการ ซึ่งจะเป็นวิธีที่ซับซ้อนกว่า ซึ่งการออกแบบด้วยวิธีนี้ได้มีการศึกษาวิจัยมาหลายครั้งด้วยกัน การจำลองกระบวนการในที่นี้จะเป็นการแสดง

คุณลักษณะของ process ออกมาด้วย fuzzy set โดยพิจารณาจาก input , ตัวแปรสถานะ และ output

มีแนวความคิดสองทางด้วยกันในการออกแบบ fuzzy controller จากแบบจำลอง fuzzy

- ทางแรก คือ การออกแบบให้ control rule นั้น ทำการ compensate ลักษณะที่ไม่ต้องการของกระบวนการเพื่อให้ได้ได้ตามเป้าหมายที่ต้องการ
- ทางที่สอง คือ เป็นไปตามทฤษฎีของ Optimal Control ซึ่งจะให้โครงการและ parameter ของ control rule ทำให้ระบบที่มี Fuzzy controller ควบคุมอยู่นั้นเป็นไปตามเป้าหมายที่ต้องการ.

## PID Controller

transfer function ของ controller แบบ PID มีรูปแบบดังต่อไปนี้

$$G_c(S) = K_p + \frac{K_i}{S} + K_d \cdot S \quad (2.1)$$

ซึ่ง  $K_p, K_i, K_d$  คือ proportional , integral และ derivative gains ตามลำดับรูปแบบอื่นที่ใช้กันอยู่ซึ่งเท่ากับสมการบน คือ

$$G_c(S) = K_p \left( 1 + \frac{1}{T_i \cdot S} + T_d \cdot S \right) \quad (2.2)$$

โดยที่  $T_i = K_p / K_i$  และ  $T_d = K_d / K_p$   $T_i$  และ  $T_d$  ถูกเรียกว่า integral time และ derivation time constants ตามลำดับ

ในรูปของ discrete-time ซึ่งเทียบเคียงกับเงื่อนไขสำหรับ PID control ที่ใช้ยู่นี้คือ

$$CS(k) = K_p \cdot e(k) + K_i \cdot T_s \sum_{i=1}^n e(i) + \frac{K_d}{T_s} \cdot \Delta e(k) + CS(0)$$

ในที่นี้  $CS(k)$  คือ สัญญาณควบคุม (control signal)

$e(k)$  คือ ค่าผิดพลาดระหว่างจุดอ้างอิงและค่าทางออกของขบวนการ (process output )

$T_s$  คือ คาบเวลา Sampling สำหรับเครื่องควบคุม (control)

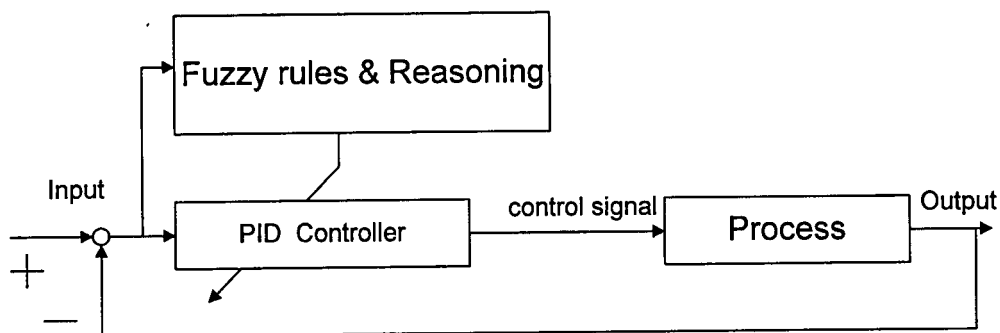
$CS(0)$  คือ สัญญาณควบคุมเมื่อค่าความผิดพลาดเป็นศูนย์ โดยปกติจะกำหนดไว้ที่กึ่งกลางสเกล

$$\text{และ } \Delta e(k) = e(k) - e(k-1)$$

ค่าตัวแปร parameters ของเครื่องควบคุม PID  $K_p, K_i, K_d$  หรือ  $K_p, T_i, T_d$  สามารถจะถูกใช้ที่จะทำให้เกิด response curve ต่างๆจากที่ให้กับ process การหาที่ดีที่สุด (optimum) ในการปรับแต่งเครื่องควบคุม (controller) สำหรับ process ไม่เป็นการง่าย

ในส่วนต่อไปนี้หลักของ on-line gain scheduling ของเครื่องควบคุม PID FUZZY RULES

## FUZZY GAIN SCHEDULING



รูปที่ 2-1 แสดงระบบของการควบคุมแบบ PID ด้วย FUZZY GAIN SCHEDULER

โดยจะเป็นการนำ FUZZY RULES และ reasoning ไปกำหนดค่า parameter สำหรับ controller โดยซึ่ง  $K_p, K_d$  จะถูกกำหนดในย่านของ  $|K_{pmin}, K_{pmax}|$  และ  $|K_{dmin}, K_{dmax}|$  ตามลำดับย่านที่เหมาะสมถูกต้องขึ้นมาโดยการทดลองและจะเป็นไปตามสมการ ต่อไปนี้

$$K_{p, min} = 0.32K_u \quad , \quad K_{p, max} = 0.6 K_u$$

$$K_{d, min} = 0.08K_u.T_u \quad , \quad K_{d, max} = 0.15K_u.T_u$$

เพื่อความสะดวกค่า  $K_p$  และ  $K_d$  จะถูกกำหนดให้อยู่ในย่านระหว่าง 0 และ 1 โดยเป็น linear transformation ดังต่อไปนี้

$$K'p = (K_p - K_{p, min}) / (K_{p, max} - K_{p, min}) \quad (2.3)$$

$$K'd = (K_d - K_{d, min}) / (K_{d, max} - K_{d, min})$$

ในโครงสร้างวิธีของ PID parameter จะถูกกำหนดด้วยหลักของ current error  $e(k)$  และค่าแตกต่างอันดับแรก  $\Delta e(k)$

ค่า integral time constant จะถูกกำหนดในเทอมของ derivative time constant คือ

$$T_i = \alpha . T_d \quad (2.4)$$

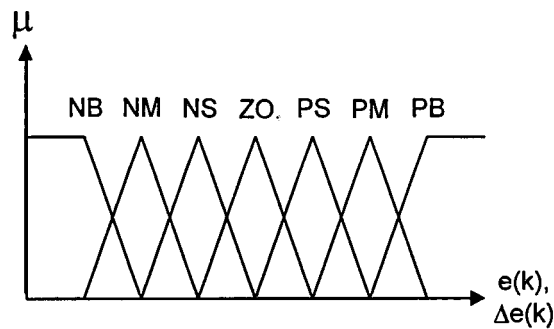
และค่า integral gain จะหาได้ดังนี้

$$K_i = K_p / (\alpha . T_d) = K^2 p / (\alpha . K_d) \quad (2.5)$$

ค่า parameter  $K'p, K'd$  และ  $\alpha$  จะถูกกำหนดโดยเซตของ fuzzy rules รูปแบบคือ If  $e(k)$  is  $A_i$  and  $\Delta e(k)$  is  $B_i$ , then  $K'p$  is  $C_i$ ,  $K'd$  is  $D_i$  and  $\alpha = \alpha_i$

$$i = 1, 2, 3, \dots, m \quad (2.6)$$

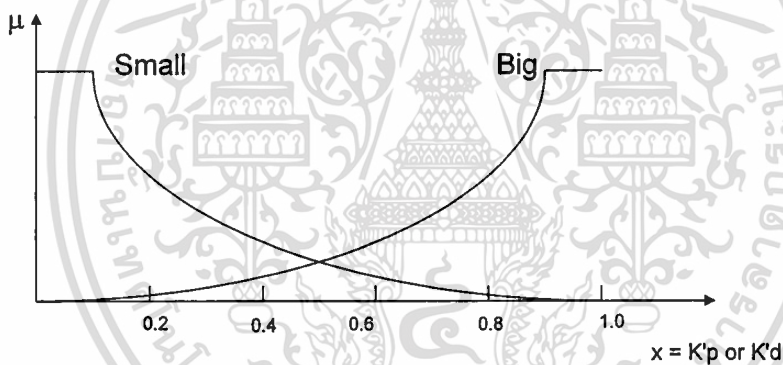
โดยที่  $A_i, B_i, C_i, D_i$  คือ fuzzy set ตรงกับส่วนของ sets  $i$  คือ ค่าคงที่ membership function (MF) ของ fuzzy sets เหล่านี้สำหรับ  $e(k)$  และ  $\Delta e(k)$  ถูกแสดงไว้ในรูปที่ 2-2



รูปที่ 2-2

ในรูปนี้ N แทนค่า negative , P แทนค่า Positive , ZO ประมาณเท่ากับ 0, S แทนค่า Small , M คือ Medium , B คือ Big ดังนั้น NM ใช้แทนสำหรับ negative -Medium, PB สำหรับ Positive Big เป็นดังเช่นนี้ ทุกค่า

FUZZY SET Ci และ Di อาจจะไม่ Big ก็ Small และ คุณลักษณะของ membership function แสดงดังรูปที่ 2-3



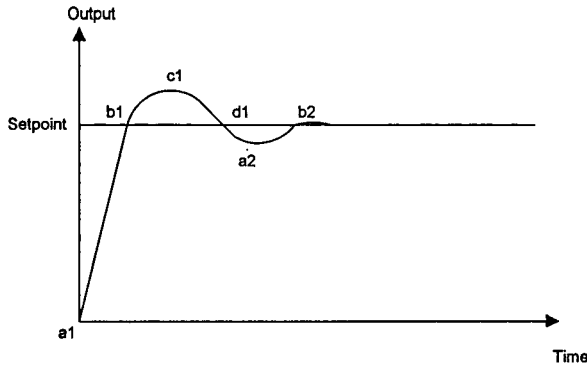
รูปที่ 2-3

ที่ซึ่ง grade ของ membership function  $\mu$  และค่าตัวแปร  $X = K'p$  หรือ  $K'd$  มีความสัมพันธ์กันดังต่อไปนี้

$$\mu_{small}(x) = -1/4 \ln x \text{ or } x_{small}(\mu) = e^{-4\mu} \text{ For small}$$

$$\mu_{big}(x) = -1/4 \ln(1-x) \text{ or } x_{big}(\mu) = 1 - e^{-4\mu} \text{ For big} \quad (2.7)$$

fuzzy rules ในสมการที่ 2.6 อาจจะได้จากความรู้ความชำนาญของผู้ปฏิบัติงานในที่นี้เราตั้ง rule ด้วยการทดลองหลักการของ step response ของ Process



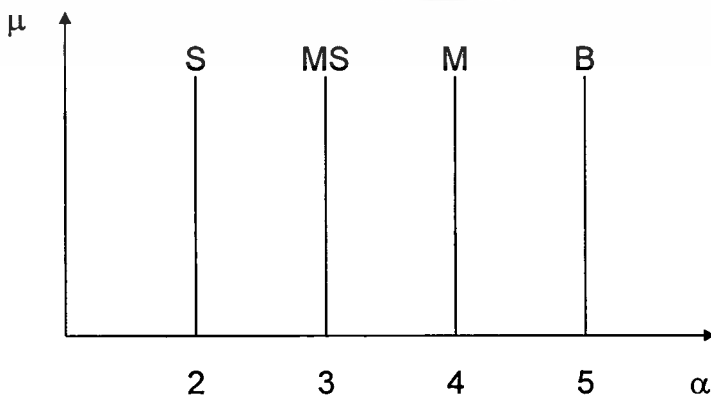
จากรูป ที่ 2-4 จะแสดงตัวอย่างของ time response ที่ต้องการ

ที่จุดเริ่มต้น  $a_1$  เราจำเป็นต้องใช้สัญญาณ control ที่ big เพื่อที่จะให้มีช่วง rise time ที่รวดเร็ว ในการที่จะผลิตสัญญาณ control ที่ big นั้น เครื่องควบคุมแบบ PID ควรมีค่า proportional gain ( $K'p$ ) สามารถที่จะแทนด้วย fuzzy set ที่big และ derivative gain  $K'd$  ด้วย fuzzy set ที่ small ค่า กิริยาของintegral จะถูกกำหนดด้วยความสัมพันธ์จากค่า กิริยา derivative ในสมการที่2-4 สำหรับ เครื่องควบคุมแบบPID การนำเอาค่า ที่ small หรือค่า integral time constant  $T_i$  ที่small จะเป็น ผลให้กิริยา integral มากอย่างไรก็ตามกิริยา integral ควรจะมากหรือน้อยนั้นจะถูกกำหนดใน โครงสร้างโดยการเปรียบเทียบกับวิธีที่รู้จักกันดีคือ กฎ tuning rule PID ด้วย Ziegler - Nichols

ในZiegler - Nichols rule ค่า integral time constant  $T_i$  จะถูกนำค่า 4ค่า ค่าซึ่งใหญ่เท่ากับ derivative time constant เสมอ นั่นคือ $\alpha$ เท่ากับที่ 4 ในโครงสร้าง $\alpha$ จะนำค่าที่ ต่ำกว่า 4 (2) ที่จะ กำหนดค่าที่มากของกิริยา integral ดังนั้น rule รอบจุด  $a_1$  อ่านได้ว่า

if  $e(k)$  is PB and  $\Delta e(k)$  is Zo, then  $K'p$  is Big ,  $K'd$  is small ,and  $\alpha = 2$

ข้อสังเกตซึ่ง  $\alpha$  อาจจะพิจารณาด้วย fuzzy number ซึ่งมี membership function เดียว ดังแสดง ในรูปที่ 2-5 สำหรับตัวอย่าง  $\alpha$  เป็น 2 เมื่อ  $\alpha$  เป็น small



รูปที่ 2-5

รอบจุด b1 ในรูปที่ 2-4 เราคาดว่าสัญญาณควบคุม (control Signal) ควรจะน้อยเพื่อจะหลีกเลี่ยง overshoot ที่มาก นั่นคือ เครื่องควบคุมแบบ PID ควรจะมี proportional gain ที่ต่ำ derivative gain ที่สูงและ integral gain ที่ต่ำ ดังเช่น fuzzy rule ที่นำมาต่อไปนี้

if  $e(k)$  is z0 and  $\Delta e(k)$  is NB , then  $K'p$  is small ,  $K'd$  is Big , and  $\alpha = 5$   
 ดังนั้นเซตของ rule จะแสดงในตารางที่ 2-1 อาจใช้ที่จะปรับค่า proportional gain ( $K'P$ ) Tuning rules สำหรับ  $K'd$  และ ให้ค่าโดยตารางที่ 2-2 และตารางที่ 2-3 ตามลำดับในตารางให้ B แทน Big และ S แทน small

$Kp'$

		$\Delta e(k)$						
		NB	NM	NS	ZO	PS	PM	PB
$e(k)$	NB	B	B	B	B	B	B	B
	NM	S	B	B	B	B	B	S
	NS	S	S	B	B	B	S	S
	ZO	S	S	S	B	S	S	S
	PS	S	S	B	B	B	S	S
	PM	S	B	B	B	B	B	S
	PB	B	B	B	B	B	B	B

ตารางที่ 2-1

$Kd'$

		$\Delta e(k)$						
		NB	NM	NS	ZO	PS	PM	PB
$e(k)$	NB	S	S	S	S	S	S	S
	NM	B	B	S	S	S	B	B
	NS	B	B	B	S	B	B	B
	ZO	B	B	B	B	B	B	B
	PS	B	B	B	S	B	B	B
	PM	B	B	S	S	S	B	B
	PB	S	S	S	S	S	S	S

ตารางที่ 2-2

เมื่อ B = BIG

S = SMALL

NB = Negative Big

NM = Negative Medium

NS = Negative Small

ZO = Zero

PS = Positive Small

PM = Positive Medium

PB = Positive Big

$\alpha$

	$\Delta e(k)$							
	NB	NM	NS	ZO	PS	PM	PB	
NB	2	2	2	2	2	2	2	
NM	3	3	2	2	2	3	3	
NS	4	3	3	2	3	3	4	
ZO	5	4	3	3	3	4	5	
PS	4	3	3	2	3	3	4	
PM	3	3	2	2	2	3	3	
PB	2	2	2	2	2	2	2	

ตารางที่ 2-3

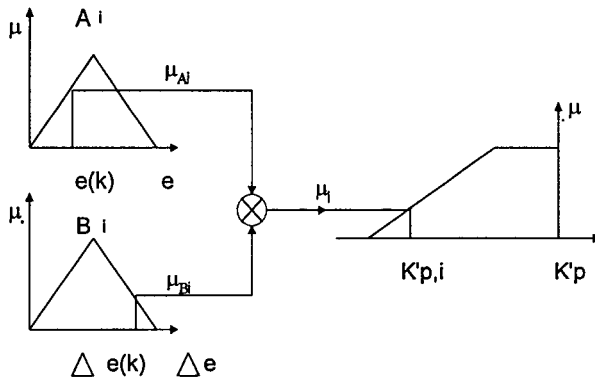
ค่าจริงของ ith rules ในสมการที่ 2-6  $\mu_i$  จะถูกหาค่าโดย product ของค่า MF ในส่วนก่อนของ rules

$$\mu_i = \mu_{Ai} [e(k)] * \mu_{Bi} [\Delta e(k)] \tag{2.8}$$

โดยที่  $\mu_{Ai}$  คือ ค่า MF ของ fuzzy set  $A_i$  จะถูกให้ค่าโดย  $e(k)$  และ  $\mu_{Bi}$  คือค่า MF ของ fuzzy set  $B_i$  จะถูกให้ค่าโดย  $\Delta e(k)$

หลักการของ  $\mu_i$  ค่าของ  $K'p$  และ  $K'd$  สำหรับแต่ละ rule จะถูกกำหนดค่าที่ตรงกับ membership function ความสัมพันธ์ของ process ของ fuzzy rule ถูกแสดงไว้ในรูปที่ 2-6

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2-6

โดยการใช้ membership function ในรูปที่ 2-2 เรามีเงื่อนไขดังต่อไปนี้

$$\sum_{i=1}^m \mu_i = 1 \tag{2.9}$$

แล้วการ defuzzification เป็นผลดังต่อไปนี้

$$K'p = \sum_{i=1}^m \mu_i \cdot K'p,i \tag{2.10a}$$

$$K'd = \sum_{i=1}^m \mu_i \cdot K'd,i \tag{2.10b}$$

$$\alpha = \sum_{i=1}^m \mu_i \cdot \alpha_i \tag{2.10c}$$

ในที่นี้  $K'p,i$  คือค่าของ  $K'p$  ที่ตรงกับ grade  $\mu_i$  สำหรับ  $i$ th rule (ดูจากรูปที่ 2-6)  $K'd,i$  ก็ถูกหาค่าในวิธีเดียวกัน เมื่อ  $K'p, K'd$  และ  $\alpha$  ถูกหาค่าได้ เครื่องควบคุมแบบ PID จะหาค่า parameter จากการคำนวณจากสมการดังต่อไปนี้ นั้นตรงกับสมการที่ 2.3 และสมการที่ 2.5

$$Kp = (Kp,max - Kp,min)K'p + Kp,min \tag{2.11a}$$

$$Kd = (Kd,max - Kd,min)K'd + Kd,min \tag{2.11b}$$

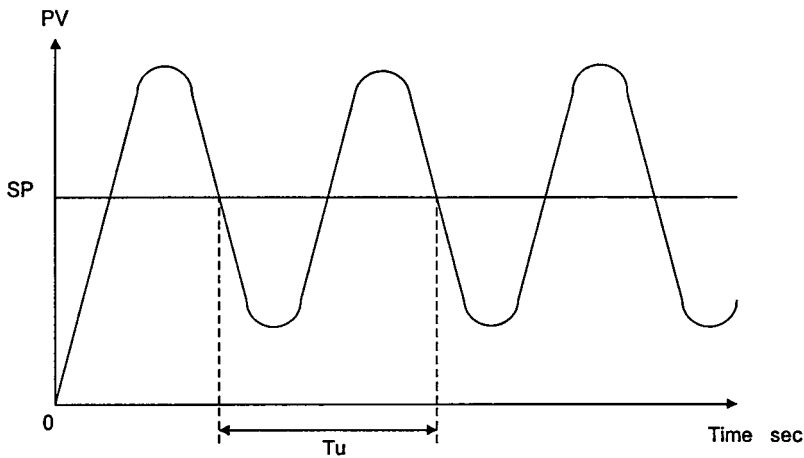
$$Ki = K^2p / (\alpha \cdot Kd) \tag{2.11c}$$

หลักการของการ simulation ที่ยาวนาน ด้วย process ต่างๆ rule ของกฎห้วม่มือสำหรับการกำหนดย่านของ  $Kp$  และย่านของ  $Kd$  จะถูกให้ค่าดังนี้

$$Kp,min = 0.32Ku, \quad Kp,max = 0.6Ku$$

$$Kd,max = 0.08Ku.Tu, \quad Kd,min = 0.15Ku.Tu$$

โดยที่  $Ku$  และ  $Tu$  คือ gain และ period ของการ oscillation ที่ stability ตามลำดับ



รูปที่ 2-7

## ขั้นตอนการออกแบบ Software

โดยเรานำเอาทฤษฎี FUZZY ของ IEEE มาประยุกต์ทำได้ดังนี้

### 1. FUZZIFICATION OF INPUT

เราจะนำเอา input (PV) ของ Process ที่ได้มาหาค่า Degree of membership โดยเราจะแปลง input (PV) ของ Process เป็น input ของ FUZZY LOGIC SYSTEM ได้ 2 ค่าคือ

$$1.1 \text{ ERROR } e(k) = SP - PV$$

$$1.2 \text{ CHANGE ERROR } = \Delta e(k) = e(k) - e(k-1)$$

SP (SET POINT) = ค่าเป้าหมายที่เราตั้งหรือกำหนด

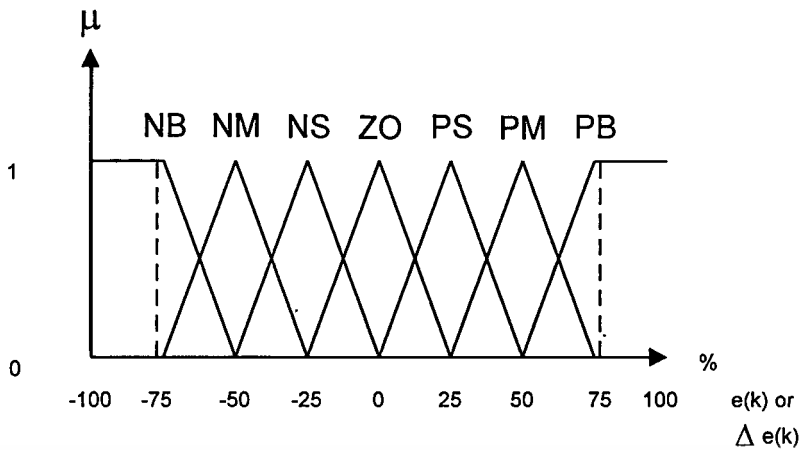
PV (PROCESS VARIABLE) = ค่า input ที่วัดได้ ณ. เวลาใดๆ

$e(k)$  = ค่า ERROR ปัจจุบัน

$\Delta e(k)$  = ค่า CHANGE ERROR

$e(k-1)$  = ค่า ERROR ก่อนหน้าปัจจุบัน

โดยจากตัวแปรอินพุตทั้ง 2 ตัวนี้  $e(k)$  และ  $\Delta e(k-1)$  จะต้องนำมาผ่านขั้นตอนของการ FUZZIFICATION ให้อยู่ในรูปของ FUZZY INPUT ได้โดยการเทียบค่าตัวแปรอินพุตกับกราฟความเป็นสมาชิก (Membership function) ของรูปที่ 2-1 และรูป 2- 2 ซึ่งรูปที่ 2-1 จะเป็นกราฟความเป็นสมาชิกของ ERROR ( $e(k)$ ) และรูปที่ 2-2 จะเป็นกราฟความเป็นสมาชิกของ CHANGE ERROR  $\Delta e(k)$  ซึ่งแต่ละกราฟจะมีจำนวนเทอมหรือ LABEL เท่ากับ 7 LABEL



โดยที่ NB = Negative Big

NM = Negative Medium

NS = Negative Small

ZO = Zero

PS = Positive Small

PM = Positive Medium

PB = Positive Big

## 2. RULE EVALUATION

การคำนวณค่าตามกฎที่ตั้งไว้ โดยอาศัยค่าความเป็นสมาชิกของตัวแปรที่ได้จากการเปรียบเทียบ ค่า FUZZY INPUT กับกราฟความเป็นสมาชิก เป็นตัวกำหนดความสำคัญของกฎแต่ละกฎ โดยกฎต่างๆ นี้จะแสดงได้ดังตารางที่ 2-1 , ตารางที่ 2-2 , ตารางที่ 2-3 ระหว่างตัวแปร FUZZY INPUT ทั้ง 2 ตัวในส่วนเงื่อนไขคือ ERROR และ CHANGE ERROR ตาม FUZZY IF/THEN RULES

## 3. DEFUZZIFICATION OF OUTPUTS

เป็นขั้นตอนที่นำเอาผลลัพธ์ที่ได้จากขั้นตอน RULES EVALUATION เพื่อหาค่าของ FUZZY OUTPUT ซึ่งเมื่อทำการเปรียบเทียบตามกฎต่างๆ แล้ว จะเห็นได้ว่าถ้ามีการคำนวณตามกฎการควบคุม แล้วนำเพียงกฎใดกฎหนึ่งมาใช้จะทำให้ผลการควบคุมระบบผิดพลาดได้จึงนำค่าเอาท์พุทของกฎต่างๆ มาหาค่าเอาท์พุทที่จะใช้ในการควบคุมระบบ โดยใช้การคำนวณแบบ CENTER OF GRAVITY คำนวณผลลัพธ์ออกมาเป็นค่าจุดศูนย์กลางสำหรับตัวแปรเอาท์พุทซึ่งได้แก่ค่า  $K'_p, K'_d$  และ

หลังจากที่ได้ทำ DEFUZZIFICATION แล้วผลลัพธ์ของการคำนวณในรูปของ FUZZY OUTPUT ซึ่งค่าของ FUZZY OUTPUT นี้ยังไม่สามารถนำไปใช้ในการควบคุม Process ได้โดยเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตรงจึงต้องนำมาผ่านขบวนการที่จะทำการเปลี่ยนให้ค่าของ FUZZY OUTPUT นั้นสามารถนำไปใช้ในการควบคุม Process ได้ดังนี้

จากการคำนวณ FUZZY OUTPUT จะได้ค่า  $K'_p, K'_d$  และ จะนำค่า 3 ค่านี้เข้าไปแทนดังสมการต่อไปนี้จะได้ค่า OUTPUT ที่จะเป็นพารามิเตอร์นำไปควบคุมได้โดยตรง

$$K_p = (K_{p,max} - K_{p,min})K' p + K_{p,min}$$

$$K_d = (K_{d,max} - K_{d,min})K' d + K_{d,min}$$

$$K_i = K^2 p / (\alpha \cdot K_d)$$

โดยที่ค่า  $K_{p,max}$  ,  $K_{p,min}$  ,  $K_{d,max}$  ,  $K_{d,min}$  หาค่าได้ดังนี้

$$K_{p,max} = 0.6 \text{ KU} \quad K_{p,min} = 0.32 \text{ Ku}$$

$$K_{d,max} = 0.15 \text{ KU TU} \quad K_{d,min} = 0.08 \text{ Ku .Tu}$$

โดยค่า KU และ TU คือ Gain และ period ของการ Oscillation ที่ Stability ตามลำดับ เมื่อได้ค่าพารามิเตอร์  $K_p, K_i, K_d$  ออกมาก็จะแปลงค่า  $K_p$  เป็น PB แปลง  $K_i$  เป็น  $T_i$  และ  $K_d$  เป็น  $T_d$  จาก

$$PB = 100/K_p$$

$$T_i = K_p/K_i = 100/ (PB)(K_i)$$

$$T_d = K_d/K_p = (PB)(K_d)/ 100$$

เมื่อได้ค่าพารามิเตอร์ข้างบนทั้ง 3 ค่าแล้วก็จะนำแทนค่าในสมการ PID คือ

$$CS(k) = K_p \cdot e(k) + K_i \cdot T_s \sum_{i=1}^n e(i) + \frac{K_d}{T_s} \cdot \Delta e(k) + CS(0)$$

ค่า  $CS(k)$  คือค่าสัญญาณเอาต์พุตที่ใช้ในการควบคุม

## บทที่ 3

### การอินเทอร์เฟส

วิธีอ้างและใช้งานแอดเดรสต่าง ๆ ของพอร์ท I/O ที่ใช้งานอยู่ใน IBM / PC

#### การอ้างแอดเดรสของพอร์ท I/O

ในการควบคุมและตรวจสอบสภาวะการทำงาน รวมทั้งการอ่านข้อมูลจากอุปกรณ์ที่เป็นชิพซ์พอร์ทหรือการ์ดต่าง ๆ ที่ใช้ในระบบของ IBM / PC นั้น จะกระทำโดยผ่านทางพอร์ท I/O ของระบบ ดังนั้นในการที่จะใช้งานหรือควบคุมการทำงานของอุปกรณ์เหล่านี้ จึงจำเป็นต้องศึกษาถึงวิธีการควบคุมพอร์ท I/O ต่าง ๆ ของระบบด้วย และเนื่องจากการควบคุมหรือติดต่อกับพอร์ทเหล่านี้ ต้องกระทำโดยการอ้างถึงแอดเดรสของพอร์ท I/O เหล่านี้โดยตรง เราจึงจำเป็นต้องศึกษาถึงหลักการอ้างแอดเดรสของ 8088 ใน IBM / PC ด้วย

สำหรับแอดเดรสของพอร์ท I/O ต่าง ๆ นั้น จะเป็นแอดเดรสที่ถูกสร้างขึ้น โดย 8088 ซึ่งแอดเดรสเหล่านี้เป็นแอดเดรสที่จัดไว้สำหรับพอร์ท I/O โดยเฉพาะ คือแยกจากแอดเดรสของหน่วยความจำโดยเด็ดขาด ส่วนการส่งข้อมูลให้กับพอร์ทเหล่านี้จะทำได้โดยการใช้คำสั่ง OUT ของ 8088 ส่งข้อมูลนั้น ไปยังแอดเดรสของพอร์ทที่ต้องการ และ สำหรับการตรวจสอบหรือการอ่านข้อมูลจากพอร์ท ก็จะทำให้ได้โดยการใช้คำสั่ง IN ของ 8088 อ่านข้อมูลจากแอดเดรสของพอร์ทที่ต้องการเช่นกัน

ภายในไมโครโปรเซสเซอร์เบอร์ 8088 นี้จะมีแอดเดรสสำหรับใช้กับพอร์ท I/O อยู่ทั้งสิ้น 65,536 หรือ 64K แอดเดรส (ในขณะที่มีแอดเดรสสำหรับหน่วยความจำอยู่ 1Mbyte) ซึ่งทำให้การอ้างแอดเดรสของพอร์ท I/O ที่ทำงานร่วมกับ 8088 นั้น ต้องใช้จำนวนเส้นแอดเดรสในบัสแอดเดรสทั้งสิ้น 16 เส้น คือ A0-A15 แต่สำหรับใน IBM / PC นี้ถูกออกแบบมาให้ใช้เส้นแอดเดรสเฉพาะ 10 เส้นล่าง คือ A0-A9 เท่านั้นดังนั้นในการอ้างถึงแอดเดรสของพอร์ทของอุปกรณ์หรือชิพซ์พอร์ทใด ๆ ที่ใช้ร่วมกับ IBM / PC จึงใช้จำนวนเส้นแอดเดรสเพียง 10 เส้นด้วย โดยเส้นแอดเดรสที่เหลือคือ A10-A15 นั้นจะไม่ถูกนำไปใช้งาน อย่างไรก็ตามถึงแม้ว่าเส้นแอดเดรส A10-A15 นี้จะไม่ถูกนำไปใช้งานแต่ค่าแอดเดรสบนเส้นแอดเดรสเหล่านี้ยังคงเปลี่ยนแปลงตามค่าแอดเดรสของพอร์ทที่กำหนดไว้ในคำสั่ง OUT หรือ IN อยู่ด้วย เพียงแต่ไม่ได้ถูกนำมาตีคู่ร่วมกับแอดเดรส A0-A9 เท่านั้น ตัวอย่างเช่นในการใช้คำสั่ง OUT ส่งข้อมูลไปยังพอร์ทที่ตรงกับแอดเดรส 0010H นั้นจะให้ผลเหมือนกับการส่งข้อมูลไปยังพอร์ทที่ตรงกับแอดเดรส 0410H, 0810, 0C10H, ทั้งนี้เนื่องจากแอดเดรส 6 บิตบนไม่ได้ถูกใช้งาน จึงทำให้การเปลี่ยนแปลงค่าแอดเดรสบนเส้นแอดเดรส A10-A15 นั้น ไม่ทำให้เกิดความแตกต่างใด ๆ ขึ้น

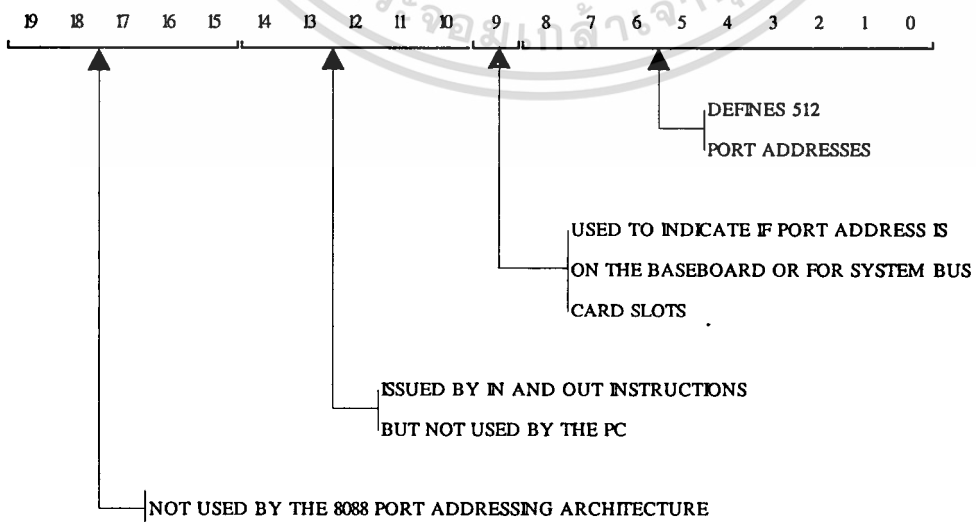
เนื่องจากใน IBM / PC ได้ใช้งานเส้นแอดเดรสเพียง 10 เส้น (คือ A0-A9) ดังนั้นจึงสามารถที่จะอ้างแอดเดรสของพอร์ทได้สูงสุดเพียง 1024 พอร์ท (จากจำนวน 64K พอร์ท) เท่านั้น นอกจากนี้ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในกรณีที่เป็นการอ่านข้อมูลจากพอร์ทของ IBM / PC ข้อมูลในบิต A9 จะถูกจัดให้มีหน้าที่ในการแบ่งพอร์ททั้ง 1024 พอร์ทออกเป็น 2 ส่วน (ส่วนละ 512 พอร์ท) อีกด้วย กล่าวคือถ้าข้อมูลในบิต A9 เป็น “0” แล้ว เราจะทำการอ่านข้อมูลได้เฉพาะจากพอร์ทของอุปกรณ์หรือชิพพอร์ทต่าง ๆ ที่อยู่บนเมนบอร์ด (Main Board) ของ IBM / PC เช่น 8253-5, 8237-5 หรือ 8259A เท่านั้น แต่ถ้าข้อมูลในบิต A9 นี้เป็น “1” ก็จะทำให้การอ่านข้อมูลได้เฉพาะจากพอร์ทที่อยู่บนการ์ดต่าง ๆ เท่านั้น

จากที่ได้กล่าวมานั้นจะสรุปได้ว่าพอร์ทบน IBM / PC ทั้ง 1024 พอร์ทถูกแบ่งออกเป็น 2 กลุ่ม โดยที่กลุ่มแรกเป็นกลุ่มของพอร์ทที่อยู่บนเมนบอร์ด และกลุ่มที่จัดเตรียมไว้สำหรับพอร์ทที่อยู่บนการ์ดต่าง ๆ

สำหรับในกรณีของการส่งข้อมูลให้กับพอร์ททั้ง 1024 พอร์ท เราสามารถที่จะเลือกส่งไปยังพอร์ทใด ๆ ใน IBM / PC ได้ ดังนั้นการเลือกแอดเดรสสำหรับพอร์ทที่อยู่บนการ์ดจึงสามารถทำได้โดยสะดวก แต่อย่างไรก็ตามสิ่งที่จะต้องคำนึงถึงก็คือถ้าแอดเดรสที่เราเลือกให้กับพอร์ทนี้ตรงกับค่าแอดเดรสเดิมที่มีอยู่บนเมนบอร์ดแล้ว เมื่อเราทำการส่งข้อมูลให้กับพอร์ทที่อยู่ในตำแหน่งแอดเดรสนี้ ก็จะเท่ากับเป็นการส่งข้อมูลให้กับทั้งพอร์ทที่อยู่บนเมนบอร์ดและพอร์ทที่อยู่บนการ์ดด้วย ซึ่งในกรณีเช่นนี้อาจจะก่อให้เกิดความผิดพลาดขึ้นได้เช่นกัน ดังนั้นในการกำหนดค่าแอดเดรสให้กับพอร์ทที่ถูกสร้างขึ้นบนการ์ดต่าง ๆ จึงควรจะให้ค่าแอดเดรสที่แอดเดรสบิต A9 มีค่าเป็น “1” คือ แอดเดรส OFE00H จนถึง OFFFFH เท่านั้น (แอดเดรสบิต A10-A15 ไม่ถูกใช้ในการตีโค้ด แต่เพื่อความสะดวกจึงกำหนดให้มีค่าเป็น “1” ในฐานสองทั้งหมด แต่ในการใช้งานจริงอาจเปลี่ยนให้แอดเดรส A10-A15 แต่ละบิตมีค่าเป็น “1” หรือ “0” ก็ได้)

สำหรับรูปที่ 3-1 นี้ จะแสดงถึงการใช้งานแอดเดรสบิตต่าง ๆ ในการอ้างแอดเดรสของพอร์ทใน IBM / PC



เอกสารนี้เป็นเอกสารที่รูปที่ 3-1 การใช้งานแอดเดรสสำหรับพอร์ท I/O ใน IBM / PC ซึ่งประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากที่ได้กล่าวไว้ในหัวข้อที่ผ่านมา นั้น พอร์ต I/O ทั้ง 1024 พอร์ตใน IBM / PC จะถูกแบ่งออกเป็น 2 กลุ่ม ๆ ละ 512 พอร์ต สำหรับในหัวข้อนี้จะกล่าวถึงการใช้งานพอร์ตต่าง ๆ เหล่านี้ โดยจะแบ่งออกเป็น 2 กลุ่มตามที่ได้อธิบายไว้ในหัวข้อที่ผ่านมาดังนี้

0000H	512	USED ON BASE SYSTEM LOGIC BOARD
01FFH		
0200H	512	'AVAILABLE IN SYSTEM BUS CARD SLOTS
03FFH		
0400H	64512	NOT USED IN PC DESIGN
FFFFH		

รูปที่ 3-2 การใช้งานแอดเดรสของพอร์ตบน IBM/PC

1. ในกลุ่มแรกนี้เป็นกลุ่มของพอร์ต I/O ที่อยู่บนเมนบอร์ดของ IBM / PC ซึ่งจะมีแอดเดรสอยู่ในตำแหน่ง 0000H จนถึง 01FFH (ขอให้ระลึกอยู่เสมอว่า A10-A15 นั้นไม่ถูกใช้งาน) หรือ แอดเดรสที่มีบิต A9 เป็น “0” นั่นเอง

สำหรับแอดเดรสของพอร์ต I/O ในกลุ่มนี้จะถูกใช้ในการอ้างแอดเดรสของชิพซ์พอร์ทัลและอุปกรณ์ที่เป็น I/O ต่าง ๆ บนเมนบอร์ดของ IBM / PC เช่น แอดเดรส 0000H จนถึง 0000FH จะถูกใช้เป็นแอดเดรสสำหรับ B237-5 DMA Controller เป็นต้น ในรูปที่ 3-3 จะแสดงถึงการใช้งานแอดเดรสต่าง ๆ ตั้งแต่ 0000H จนถึง 01FFH ในการอ้างแอดเดรสของชิพซ์พอร์ทัลและอุปกรณ์ต่าง ๆ ที่ทำหน้าที่เป็น I/O บนเมนบอร์ดของ IBM / PC



อย่างไรก็ตามการใช้งานแอดเดรสในกลุ่มนี้อาจจะเปลี่ยนแปลงไปได้ ทั้งนี้ขึ้นอยู่กับการใช้งานการ์ดต่าง ๆ ร่วมกับ IBM / PC โดยการ์ดที่ถูกออกแบบผลิตขึ้นมาใหม่นั้นอาจจะใช้ค่าแอดเดรสต่าง ๆ ที่เหลืออยู่นั้นได้ ดังนั้นก่อนที่จะทำการออกแบบวงจรอินเทอร์เฟสที่จำเป็นต้องใช้ค่าแอดเดรสสำหรับพอร์ต I/O จึงควรจะตรวจสอบดูก่อนว่าการ์ดต่าง ๆ ที่ใช้อยู่ในระบบของ IBM / PC ที่เราใช้งานอยู่นั้นมีคาร์ดใดบ้าง และ คาร์ดเหล่านั้นใช้งานแอดเดรสใดบ้าง จากนั้นจึงทำการออกแบบวงจรอินเทอร์เฟสโดยเลือกใช้เฉพาะแอดเดรสที่ยังไม่ถูกใช้งาน

		HEX ADDRESS	USE
0200H	1	0200H	(NOT USED)
0201H	1	0201H	GAME CONTROL ADAPTER
0202H	118	0202H - 0277H	(NOT USED)
0277H			
0278H	8	0278 - 027FH	SECOND PRINTER PORT ADAPTER
027FH			
0280H	120	0280H - 02F7H	(NOT USED)
02F7H			
02F8H	8	02F8H - 02FFH	SECOND SERIAL PORT ADAPTER CARD
02FFH			
0300H	120	0300H - 0377H	(NOT USED)
0377H			
0378H	8	0378H - 037FH	PRINTER PORT ADAPTER CARD
037FH			
0380H	48	0380H - 03AFH	(NOT USED)
03AFH			
03B0H	16	03B0H - 03BFH	MONOCHROME AND PRINTER ADAPTER
03BFH			
03C0H	16	03C0H - 03CFH	(NOT USED)
03CFH			
03D0H	16	03D0H - 03DFH	COLOR GRAPHICS ADAPTER
03DFH			
03E0H	16	03E0H - 03EFH	(NOT USED)
03EFH			
03F0H	8	03F0H - 03F7H	5 1/4 INCH DISKETTE DRIV ADAPTER CARD
03F7H			
03F8H	8	03F8H - 03FFH	SERIAL PORT ADAPTER CARD
03FFH			

NOTE NEW FEATURES BY IBM AND OTHER MANUFACTURES MAY USE SOME OF THE SPARE

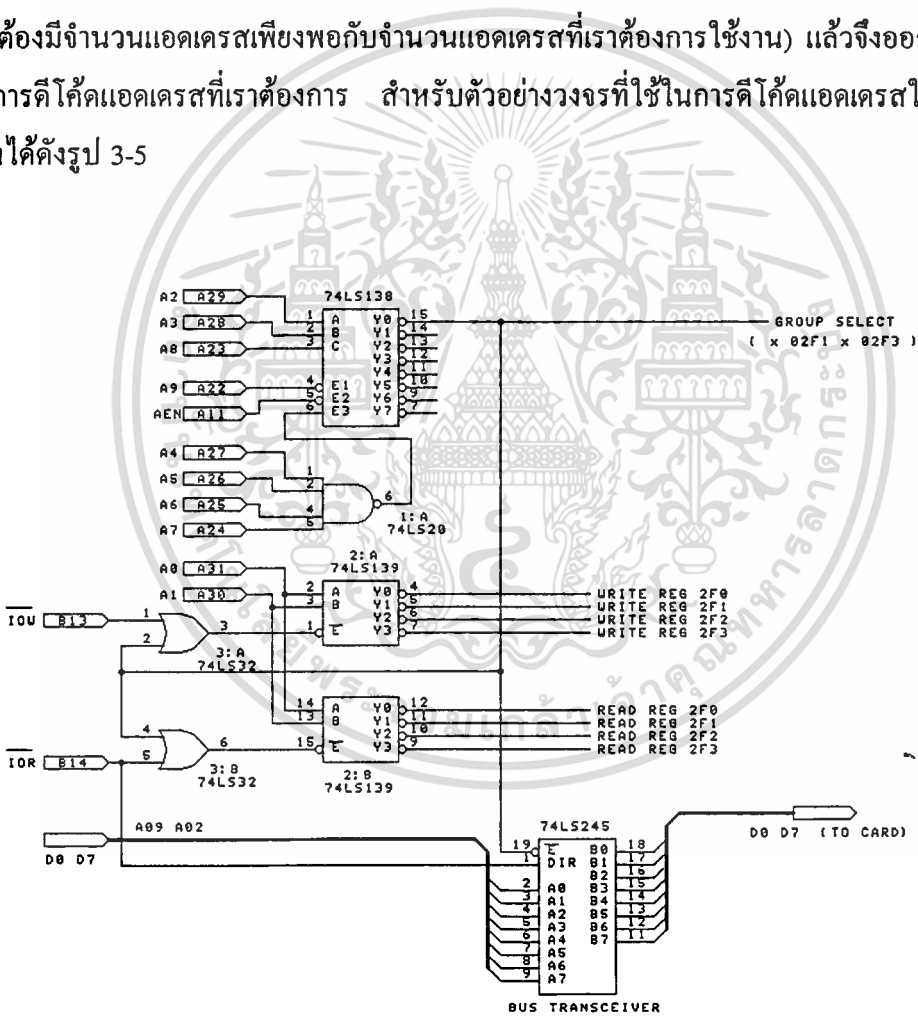
I/O ADDRESS DECODES

**เทคนิคในการดีโค้ดแอดเดรสสำหรับพอร์ท I/O**

ในหัวข้อต่าง ๆ ที่ผ่านมาข้างต้นนั้น ได้กล่าวถึงการอ้างแอดเดรสและการใช้งานแอดเดรสต่าง ๆ ของพอร์ท I/O ใน IBM / PC สำหรับในหัวข้อนี้จะกล่าวถึงวิธีการต่าง ๆ ที่ใช้ในการดีโค้ดแอดเดรสต่าง ๆ ให้เป็นไปตามที่ต้องการ

**การดีโค้ดแบบ Fixed**

วิธีการดีโค้ดแบบนี้เป็นวิธีที่ง่ายและสะดวกในการดีโค้ดแอดเดรสหรือกลุ่มของแอดเดรสของพอร์ท I/O ซึ่งวิธีนี้เป็นการกำหนดจำนวนของแอดเดรสที่เราต้องการใช้ จากนั้นจึงทำการเลือกบล็อกของแอดเดรสที่ยังไม่ถูกใช้งานโดยคาร์ดหรือวงจรมินิโพลาริซอื่น ๆ (บล็อกของแอดเดรสที่เลือกต้องมีจำนวนแอดเดรสเพียงพอกับจำนวนแอดเดรสที่เราต้องการใช้งาน) แล้วจึงออกแบบวงจรที่ทำการดีโค้ดแอดเดรสที่เราต้องการ สำหรับตัวอย่างวงจรที่ใช้ในการดีโค้ดแอดเดรสในแบบนี้จะแสดงได้ดังรูป 3-5



รูปที่ 3-5 ตัวอย่างวงจรดีโค้ดแอดเดรสแบบ Fixed

จากรูปจะเห็นได้ว่า วงจรที่ใช้เป็นวงจรที่สามารถทำการดีโค้ดแอดเดรสได้ 8 กลุ่ม โดยแต่ละกลุ่มจะมีจำนวนแอดเดรส 4 แอดเดรส ซึ่งแอดเดรสทั้ง 8 กลุ่มจะแสดงได้ดังตารางข้างล่าง

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กลุ่ม	แอดเดรส
0 (Y0)	02F0H-02F3H
1 (Y1)	02F4H-02F7H
2 (Y2)	02F8H-02FBH
3 (Y3)	02FCH-02FFH
4 (Y4)	03F0H-03F3H
5 (Y5)	03F4H-03F7H
6 (Y6)	03F8H-03FBH
7 (Y7)	03FCH-03FFH

สำหรับในตัวอย่างนี้จะเลือกใช้การตีโค้ดแอดเดรสในกลุ่ม 0 (เริ่มจากแอดเดรส 02F0H จนถึง 02F3H) คือ ใช้สัญญาณเอาต์พุต (สัญญาณ GROUPSELECT) จากขา YO (ขา15) ของ 74LS138 ไปทำการ OR กับสัญญาณ IOR และ IOW เพื่อสร้างเป็นสัญญาณอีนาเบิลวงจรถีโค้ด (74LS139) แอดเดรสอีก 4 แอดเดรส ซึ่งแบ่งเป็น 2 ชุดคือ ชุดที่เป็น WRITE REG ซึ่งจะแอกทีฟ (ลอจิก "0") เมื่อ CPU ต้องการจะส่งข้อมูลให้กับวงจรรภายนอก (สัญญาณ IOW แอกทีฟ) และ ชุดที่เป็น READ REG ซึ่งจะแอกทีฟเมื่อ CPU ต้องการจะอ่านข้อมูลจากวงจรรภายนอก (สัญญาณ IOR แอกทีฟ) สัญญาณ WRITE REG และ READ REG นี้โดยทั่วไปจะนำไปเป็นสัญญาณสโตรบ (Strobe) ให้กับวงจรรภายนอกที่เกี่ยวข้อง เพื่อให้สามารถส่งหรือรับข้อมูลจาก CPU ได้ในช่วงเวลาที่เหมาะสม นอกจากนี้สัญญาณ GROUPSELECT ยังถูกนำไปใช้ในการอีนาเบิลบัฟเฟอร์ 74LS245 ด้วย เพื่อให้ CPU สามารถส่งหรือรับข้อมูลจากอุปกรณ์ภายนอกได้เมื่อแอดเดรสในกลุ่มนี้ถูกเลือก สำหรับทิศทางของข้อมูลจะถูกควบคุมโดยสัญญาณ IOR ส่วนสัญญาณ AEN จะถูกนำมาใช้ในการคิสเอเบิลวงจรถีโค้ดโดยถ้าสัญญาณ AEN เป็น 1 ซึ่งเป็นช่วงเวลาของขบวนการ DMA นั้น 74LS138 จะถูกคิสเอเบิลทันที ทั้งนี้ก็เพื่อป้องกันความผิดพลาดที่อาจเกิดขึ้น เนื่องจากการตีโค้ดแอดเดรสของพอร์ตในระหว่างขบวนการ DMA นั้นเอง (ในระหว่างนี้แอดเดรสบนบัสแอดเดรสจะเป็นแอดเดรสของหน่วยความจำ คือ สัญญาณ MEMW หรือ MEMR จะแอกทีฟ แต่ในขณะที่เดียวกันสัญญาณ IOR หรือ IOW ก็จะไม่แอกทีฟด้วย ดังนั้นถ้าไม่คิสเอเบิลวงจรถีโค้ดไว้แล้ว อาจจะทำให้วงจรถีโค้ดคิดว่าแอดเดรสบนบัสแอดเดรสเป็นแอดเดรสของพอร์ต I/O ก็ได้) ในการตีโค้ดแอดเดรสของพอร์ต I/O เราจะต้องคำนึงถึงช่วงเวลาของสัญญาณที่เกิดขึ้นในขบวนการอ่านหรือเขียนข้อมูลลงบนพอร์ต I/O ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. ในช่วงเริ่มต้นของบัสไซเคิลที่เกี่ยวกับพอร์ท I/O นั้น ถ้าสัญญาณจากวงจรตีโค้ดมีการหน่วงเวลา (DELAY) มากเกินไป อาจจะทำให้สัญญาณตีโค้ดนี้เกิดขึ้นหลังจากที่สัญญาณ IOR หรือ IOW แอคทีฟ และเนื่องจากค่าแอดเดรสบนบัสแอดเดรสนั้นเปลี่ยนแปลงได้ตลอดเวลา ดังนั้นก่อนที่ค่าแอดเดรสที่ถูกต้องจะถูกส่งออกมาบนบัสแอดเดรสนั้น วงจรตีโค้ดจะได้รับค่าแอดเดรสอื่นอยู่ ซึ่งถ้าหากวงจรตีโค้ดมีการหน่วงเวลามากเกินไปแล้ว สัญญาณตีโค้ดแอดเดรสที่ไม่ถูกต้องนี้อาจจะถูกหน่วงเวลาจนเกิดขึ้นในช่วงเวลาที่สัญญาณ IOR หรือ IOW เกิดขึ้นแล้วก็ได้ ทำให้ข้อมูลบนบัสข้อมูลนั้นถูกส่งไปยังพอร์ทที่ไม่ถูกต้อง สำหรับใน IBM/PC จะถูกออกแบบให้การหน่วงเวลาในวงจรตีโค้ดนั้นมีค่าไม่เกิน 92 nanosec.

2. ในช่วงท้ายของบัสไซเคิลในการเขียนข้อมูลลงบนพอร์ท I/O นั้น ถ้าสัญญาณ IOW มีการหน่วงเวลาออกไป และวงจรตีโค้ดมีความเร็วในการทำงานสูงแล้ว อาจจะทำให้ข้อมูลในบัสไซเคิลนี้ถูกส่งให้กับพอร์ท I/O ที่มีแอดเดรสตรงกับค่าแอดเดรสในบัสไซเคิลต่อไปก็ได้ สำหรับใน IBM / PC สัญญาณ IOW จะมีการหน่วงเวลาไปไม่เกิน 200 nanosec.

อย่างไรก็ตามช่วงเวลาที่ต้องสนใจมากอีกช่วงเวลาหนึ่ง ก็คือ ช่วงเวลาระหว่างขอบขาขึ้นของสัญญาณ IOW กับช่วงเวลาที่ข้อมูลที่ต้องการถูกส่งออกมาบนบัสข้อมูล ถ้าสัญญาณ IOW ถูกหน่วงเวลาไปเกินกว่า 120 nanosec. แล้ว อาจจะทำให้พอร์ท I/O ใ้รับข้อมูลที่ไม่ถูกต้องก็ได้ และสำหรับสัญญาณ IOR นั้นถ้ามีการหน่วงเวลาเกิดขึ้นแล้ว ก็จะทำให้ความเร็วในการอ่านข้อมูลถูกลดลง

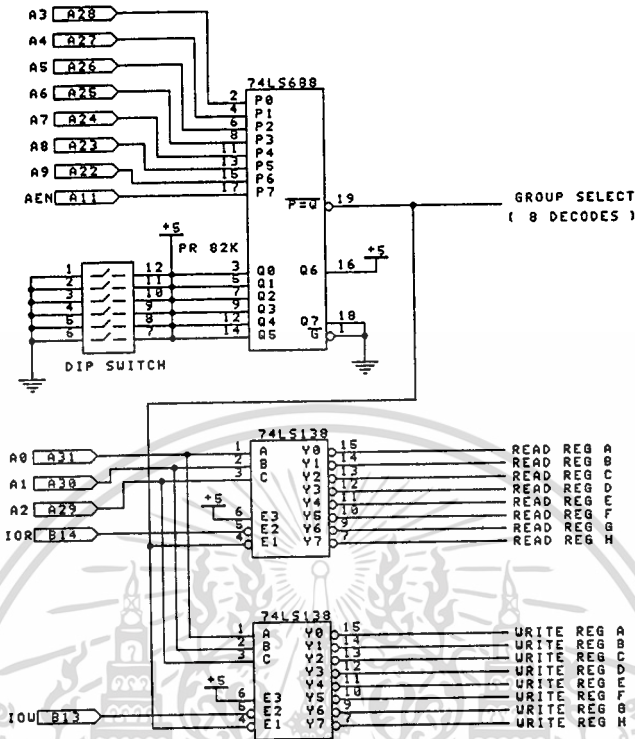
### การตีโค้ดโดยใช้สวิทช์เลือก

การตีโค้ดในแบบ fixed ที่ได้กล่าวไว้ในหัวข้อที่ผ่านมา มีข้อเสียอยู่บางประการคือ แอดเดรสที่เราเลือกใช้งาน ใ้วันนั้นอาจจะซ้ำกับแอดเดรสของคาร์ดอื่นที่เรานำมาเพิ่มเข้าไปในระบบ ในภายหลังก็ได้ ซึ่งกรณีเช่นนี้เราต้องแก้ไขวงจรเพื่อหลีกเลี่ยงไปใช้แอดเดรสอื่นที่ยังว่างอยู่ และไม่ถูกใช้งานโดยคาร์ดที่จะเพิ่มเข้าไปใหม่ ซึ่งยุ่งยากและต้องเสียเวลามากขึ้น ปัญหาเช่นนี้เราสามารถแก้ไขได้โดยใช้วงจรตีโค้ดที่สามารถเปลี่ยนแปลงค่าแอดเดรสได้ โดยเพียงแค่เปลี่ยนตำแหน่งของสวิทช์ (ในที่นี้คือ DIP Switch) ที่เซ็ทไว้ในวงจรเท่านั้น ดังรูปที่ 3-6

จากรูปเป็นวงจรที่ทำการตีโค้ดกลุ่มแอดเดรสขนาด 8 แอดเดรส ซึ่งการเลือกกลุ่มแอดเดรสที่จะทำการตีโค้ดนี้จะทำได้โดยการเซ็ท DIP Switch ที่ขา Q0-Q5 ของ 74LS688

สำหรับหน้าที่ของ 74LS688 นี้จะทำการเปรียบเทียบค่าของอินพุต 2 ชุดที่ถูกส่งเข้ามาทางขา P0-P7 และขา Q0-Q7 ถ้าอินพุตทั้ง 2 ชุดนี้เท่ากันแล้ว เอาท์พุทที่ขา P=Q จะให้อาท์พุทเป็นลอจิก "0" จากในวงจรขา P0-P6ของ 74LS688 ต่อกับแอดเดรสบิต A3-A9 ในขณะที่ขา Q0-Q5 ต่อกับความต้านทานที่ทำหน้าที่เป็น Pull Up (รักษาระดับแรงดันให้เป็นลอจิก "1" ไว้ในกรณีที่ไม่มีอินพุตใด ๆ เข้ามา) และขา Q0-Q5 นี้จะต่อกับปลายข้างหนึ่งของ DIP Switch ด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3-6 ตัวอย่างวงจรดีโค้ดโดยใช้สวิตช์เลือก

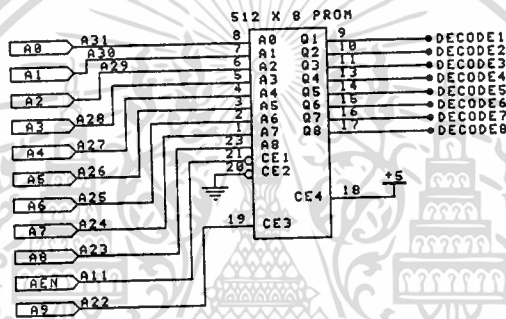
ส่วนปลายอีกข้างหนึ่งของ DIP Switch นั้นจะต่อลง Ground (ลอจิก “0”) ไว้ ดังนั้นถ้าเราทำการ “ON” DIP Switch ที่ต่อกับขาใดถูก “OFF” ขานั้นก็จะได้รับลอจิก “1” และเนื่องจากอินพุตที่ขา P0-P5 นั้นต้องเปลี่ยนแปลงตามไปด้วยจึงจะทำให้เอาต์พุตของ 74LS688 แอคทีฟได้ ทำให้เราสามารถเปลี่ยนค่าแอดเดรสที่ต้องการดีโค้ดได้ง่ายกว่าวิธีการดีโค้ดแบบ Fixed สำหรับขา Q6 นั้นจะต่อกับลอจิก “1” (+5 V) และขา P6 ต่อกับแอดเดรสบิต A6 ในกรณีเช่นนี้จึงเท่ากับเป็นการบังคับให้แอดเดรสที่จะทำการดีโค้ดได้นั้น จะต้องมีแอดเดรสบิต A9 เป็น “1” เท่านั้น ส่วนขา P7 จะต่อกับสัญญาณ AEN โดยมีขา Q7 ต่อกับลอจิก “0” การต่อในลักษณะนี้ก็เพื่อป้องกันไม่ให้ 74LS688 ทำการดีโค้ดในระหว่างขบวนการ DMA นั้นเอง เอาต์พุตจากขา P=Q ของ 74LS688 นี้จะถูกนำไปใช้ในการอินาเบิล 74LS138 ซึ่งทำหน้าที่ในการดีโค้ดแอดเดรส 8 แอดเดรสของกลุ่มแอดเดรสที่เราเลือก (โดยใช้ DIP Switch ดังที่ได้กล่าวในตอนต้น)

วงจรในลักษณะนี้เราสามารถจะนำไปใช้เป็นการดีโค้ดในแบบ Fixed ได้โดยการนำเอา DIP Switch ออก จากนั้นถ้าอินพุตใดต้องการลอจิก “0” จึงจะใช้ตัวนำเชื่อมต่อระหว่างขั้วทั้งสองแทนการเสียบ DIP Switch ให้ “ON” แต่ถ้าอินพุตใดต้องการลอจิก “1” ก็ปล่อยขั้วทั้งสองนั้นไว้

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## การตีโค้ดโดยใช้ PROM

การตีโค้ดในแบบต่างที่กล่าวมาแล้วนั้น เป็นการตีโค้ดในลักษณะที่แอดเดรสของพอร์ตต่าง ๆ อยู่รวมกันเป็นกลุ่ม แต่ในบางกรณีพอร์ตที่เราใช้งานนั้นมีแอดเดรสแยกกันอย่างเป็นอิสระเช่นในการนำเอาหน้าที่การทำงานที่อยู่บนการ์ดต่าง ๆ มารวมไว้บนการ์ดเพียงการ์ดเดียว และมีความจำเป็นที่ต้องคงค่าแอดเดรสของพอร์ตเดิม (ที่อยู่บนการ์ดเดิม) ไว้ด้วย ทำให้ไม่สามารถใช้การตีโค้ดในแบบต่าง ๆ ที่ผ่านมาได้ เนื่องจากการใช้วิธีการตีโค้ดในแบบที่ผ่านมานั้นจะทำให้ต้องใช้อุปกรณ์ที่ทำการตีโค้ดนั้นมากเกินไป ในกรณีเช่นนี้เราจำเป็นต้องใช้การตีโค้ดอีกแบบหนึ่งซึ่งจะได้กล่าวถึงในหัวข้อนี้ คือการตีโค้ดโดยใช้ PROM (Programmable Read Only Memory) ดังในรูปที่ 3-7



รูปที่ 3-7 ตัวอย่างวงจรตีโค้ดโดยใช้ PROM

จากรูปข้างต้นเป็นวิธีการง่าย ๆ แบบหนึ่งในการตีโค้ดโดยใช้ PROM ซึ่งจะเห็นได้ว่าเราใช้เส้นแอดเดรส A0-A8 ของระบบต่อเข้ากับเส้นแอดเดรส A0-A8 ของ PROM และใช้บิตข้อมูลทั้ง 8 ของ PROM คือ Q1-Q8 เป็นเอาต์พุต สำหรับใช้เป็นสัญญาณตีโค้ดให้กับกับพอร์ตต่าง ๆ 8 พอร์ต อย่างไรก็ตามสัญญาณตีโค้ดทั้ง 8 เส้นคือ DECODE1-DECODE8 นี้ยังคงต้องนำไป OR กับสัญญาณ IOR หรือ IOW ก่อนที่จะนำไปอินพุตที่มีแอดเดรสตรงกับแอดเดรสที่ป้อนให้กับ PROM นั้น

จากที่ได้กล่าวมานั้นจะเห็นได้ว่าส่วนของวงจรตีโค้ดนั้น จะมี PROM เพียงตัวเดียวเท่านั้น ซึ่ง PROM ที่จะนำมาใช้งานนี้จะต้องถูกโปรแกรมมาก่อนแล้ว โดยข้อมูลที่โปรแกรมให้กับแอดเดรสต่าง ๆ ของ PROM นั้น จะต้องสัมพันธ์กับสัญญาณตีโค้ดที่เราต้องการ กล่าวคือเราจะต้องทราบเสียก่อนว่าค่าแอดเดรสของพอร์ตทั้ง 8 ที่เราต้องการจะตีโค้ดนั้น มีแอดเดรสใดบ้าง แล้วจึง

กำหนดว่าพอร์ตใดจะใช้สัญญาณตีโค้ดเส้นใด จากนั้นจึงโปรแกรมข้อมูลให้กับ PROM โดยแอดเดรสการคำนวณว่ากรณิใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เครื่องคอมพิวเตอร์ต้องการให้สัญญาณติโค้ดไคแอกทีฟ (ในที่นี้จะกำหนดให้สัญญาณติโค้ดแอกทีฟที่ลอจิก “0”) ก็กำหนดให้ข้อมูลในบิตที่ตรงกับสัญญาณติโค้ดนั้นเป็น “0” เช่นถ้าเรากำหนดให้แอดเดรสของพอร์ทที่เราต้องการจะติโค้ดเป็น 0393H และเลือกใช้สัญญาณ DECODE5 เราก็จะต้องทำการโปรแกรมให้แอดเดรส 0193H ของ PROM (เหตุที่แอดเดรสของ PROM เป็น 0193H แทนที่จะเป็นแอดเดรส 0393H เหมือนกับแอดเดรสของพอร์ท ก็เพราะแอดเดรสของ PROM มีเพียง 9 บิต คือ A0-A8 เท่านั้น ส่วนบิต 9 จะถูกต่อเข้ากับ PROM ในภายหลัง เพื่ออีนาเบิล PROM เมื่อข้อมูลในบิต A9 นี้เป็น “1” เท่านั้น) มีข้อมูลในบิต Q5 (ถ้านับเริ่มจากบิต D0 ก็คือบิต D4) เป็น “0” ส่วนบิตอื่น ๆ นั้นมีค่าเป็น “1” ทั้งหมด ดังนั้นการโปรแกรมแอดเดรส 0193H ของ PROM จึงต้องโปรแกรมด้วยข้อมูล 0EFH เป็นต้น สำหรับข้อมูลในแอดเดรสอื่น ๆ ที่นอกเหนือจากแอดเดรสทั้ง 8 ที่กำหนดแล้วจะต้องโปรแกรมให้ข้อมูลทุกบิตเป็น “1” ทั้งหมด ซึ่งก็คือโปรแกรมด้วยข้อมูลที่ตำแหน่ง 0FFH นั่นเอง

### ตัวอย่างเช่น

ถ้าแอดเดรสของพอร์ททั้ง 8 ที่เราต้องการจะติโค้ดเป็น 024A, 02B5, 0317, 0361, 0382, 03A8, 03C4 และ 03DB ในฐานสิบหกตามลำดับ โดยกำหนดให้สัญญาณจากการติโค้ดแอดเดรสเหล่านี้เป็นสัญญาณ DECODE1 จนถึง DECODE8 ตามลำดับแล้ว (เช่นสัญญาณจากการติโค้ด 024AH ก็คือสัญญาณ DECODE1 และสัญญาณจากการติโค้ด 02B5H ก็คือสัญญาณ DECODE2 เป็นต้น) เราจะต้องทำการโปรแกรม PROM ให้มีข้อมูลที่สัมพันธ์กับเอาต์พุตที่เราต้องการ ดังนี้

1. แอดเดรสของพอร์ทเป็นแอดเดรสที่บิต A9 ถูกใช้งานร่วมด้วย โดยในบิตนี้จะต้องมีข้อมูลเป็น “1” ในขณะที่แอดเดรสของ PROM จะมีเพียง 9 บิต คือ A0-A8 เท่านั้น เราจึงจัดแอดเดรสของ PROM เมื่อเทียบกับแอดเดรสของพอร์ทดังนี้

แอดเดรสของพอร์ท (บิต A9 ถูกใช้งาน)	แอดเดรสของ (PROM) (เฉพาะบิต A0-A8)
024A	04A
02B5	0B5
0317	117
0361	161
0382	182
03A8	1A8
03C4	1C4
03D4	1DB

2. ข้อมูลที่จะโปรแกรมให้กับแอดเดรสทั้ง 8 ของ PROM จะต้องสัมพันธ์กับเอาต์พุตที่ต้องการ เช่น ถ้ามีการอ้างถึงแอดเดรสของพอร์ท 02B5H แล้ว PROM จะต้องให้เอาต์พุตที่มีลอจิก "0" ที่ขา Q2 (DECODE 2) ส่วนเอาต์พุตที่ขาอื่นต้องเป็น "1" ดังนั้นจึงต้องโปรแกรมให้แอดเดรส 00B5H ของ PROM มีข้อมูลเป็น 1111 1101 (ฐานสอง) หรือ ORDH เป็นต้น สำหรับแอดเดรสอื่นๆ นอกเหนือจากแอดเดรสทั้ง 8 นี้แล้วจะต้องถูกโปรแกรมให้มีข้อมูลเป็น OFF (ฐานสิบหก) ทั้งหมด ดังนี้

แอดเดรสของ PROM(ฐานสิบหก)	ข้อมูล(ฐานสิบหก)
000-049	OFF
04A	0FE
04B-0B4	OFF
0B5	0FD
0B6-116	OFF
117	0FB
118-160	OFF
161	0F7
162-181	OFF
182	0EF
183-1A7	OFF
1A8	0DF
1A9-1C3	OFF
1C4	0BF
1C5-1DA	OFF
1DB	07F
1DC-1FF	OFF

อย่างไรก็ตามสิ่งสำคัญอีกสิ่งหนึ่งที่จะต้องคำนึงถึงเสมอเมื่อใช้วิธีการดีโค้ดในแบบนี้ก็คือ PROM ที่ใช้นั้นจะต้องใช้เวลาในการทำงานน้อยกว่า 92 nanosec. ด้วย

#### การเพิ่มจำนวนแอดเดรสของพอร์ทบน IBM / PC

จากที่ได้กล่าวมาแล้วว่า ภายใน IBM / PC นั้นได้ออกแบบให้มีการอ้างแอดเดรสของพอร์ท (โดยวงจรมินิเทอร์เฟสภายนอก) ไม่รวมพอร์ทที่อ้างถึงโดยระบบบนเมนบอร์ดของ IBM / PC อีก 512 พอร์ท) ได้เพียง 512 พอร์ท จากจำนวนที่ 8088 สามารถอ้างได้สูงสุดถึง 64K พอร์ท

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่อนำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

และในจำนวน 512 พอร์ต ดังกล่าวนี้อาจได้ถูกใช้งานโดยอุปกรณ์หรือการ์ดต่าง ๆ ไปไม่น้อย ซึ่งอาจจะทำให้การออกแบบใช้งานวงจรอินเทอร์เฟซที่จำเป็นต้องใช้แอดเดรสของพอร์ตเป็นจำนวนมากนั้นเกิดปัญหาขึ้นได้ ดังนั้นในหัวข้อนี้จะได้กล่าวถึงวิธีการง่าย ๆ ในการเพิ่มจำนวนแอดเดรสของพอร์ตต่าง ๆ บน IBM / PC

### การใช้เส้นแอดเดรส 6 บิตบน

เราได้ทราบมาแล้วว่า การออกแบบวิธีการดีโค้ดแอดเดรสของพอร์ตต่าง ๆ ใน IBM / PC นั้น จะใช้เส้นแอดเดรสเพียง 10 บิต คือ A0-A9 เท่านั้น จากจำนวนทั้งสิ้น 16 บิต (A0-A15) จะเห็นได้ว่ามีจำนวนเส้นแอดเดรสที่ไม่ได้ถูกใช้งานอยู่อีกถึง 6 บิตคือ A10-A15 ในหัวข้อนี้จึงจะกล่าวถึงวิธีการที่จะใช้งานเส้นแอดเดรสเหล่านี้ในการอ้างแอดเดรสของพอร์ตบนวงจรอินเทอร์เฟซที่เราออกแบบขึ้น

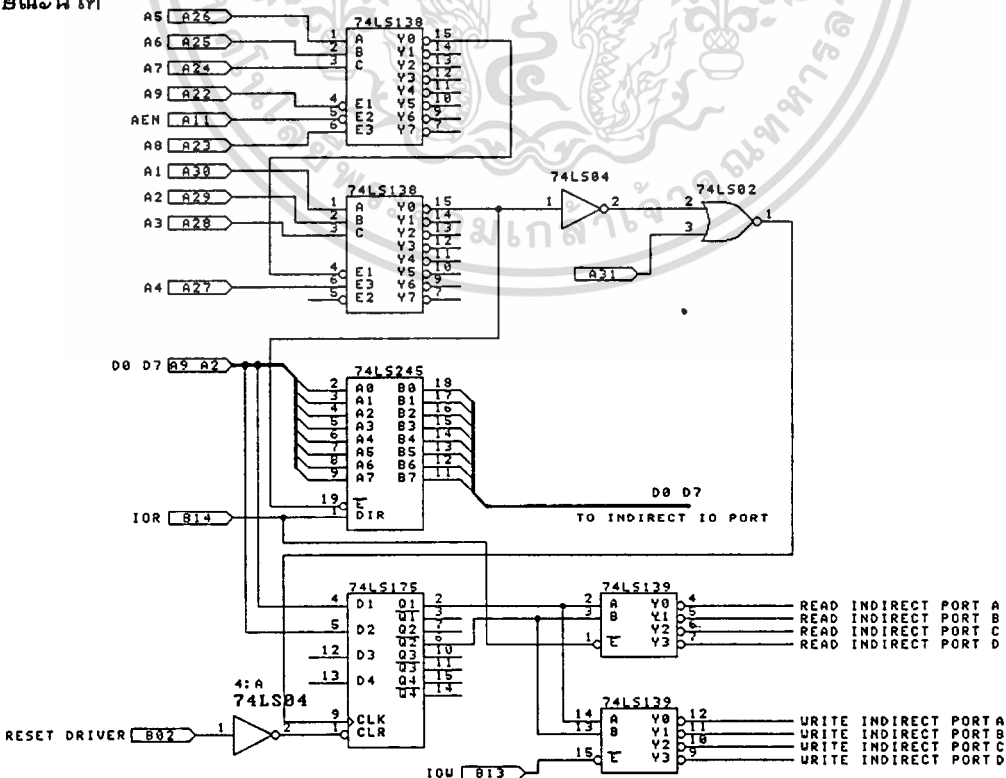
ในการที่จะใช้งานเส้นแอดเดรสเหล่านี้ เราจะต้องเริ่มต้นจากการตรวจสอบว่าแอดเดรสใดบ้างที่ถูกใช้งานโดยการ์ดหรือวงจรอินเทอร์เฟซอื่น ๆ ที่เราเพิ่มเข้าไปในระบบจากนั้นจึงเลือกแอดเดรสที่ไม่ถูกใช้งานโดยวงจรหรือการ์ดเหล่านี้มาใช้งาน ทั้งนี้เพื่อให้แน่ใจว่าในขณะที่เราใช้งานแอดเดรส 5 บิตบนอยู่นั้น จะไม่มีผลกับแอดเดรสของพอร์ตเดิมที่มีอยู่บนการ์ดต่าง ๆ ตัวอย่างเช่น ถ้าในระบบของเรามีการใช้งานการ์ดเพียง 2 แผ่นคือ CoIO / Graphics Card ซึ่งใช้แอดเดรสของพอร์ตตั้งแต่ 03D0H จนถึง 03DFH และ 5 1/4 Disk Drive Adapter Card ซึ่งใช้แอดเดรสของพอร์ตตั้งแต่ 03F0H-03F7H แล้ว เราก็สามารถจะใช้แอดเดรส 6 บิตบนนี้ กับแอดเดรสใดก็ได้ที่แอดเดรส 10 บิตล่าง ไม่ได้อยู่ในช่วงของ 0300H-03DFH และ 03F0H-03F7H ทั้งนี้ก็เพราะแอดเดรสของพอร์ตเหล่านี้ใช้งานเฉพาะเส้นแอดเดรส 10 บิตล่างเท่านั้น ดังนั้นไม่ว่าแอดเดรส 6 บิตบนที่เหลือจะเปลี่ยนแปลงไปอย่างไรก็ตาม โดยที่แอดเดรส 10 บิตล่างยังคงเดิมอยู่พอร์ตเหล่านี้ก็จะถูกทำให้แอกทีฟได้เสมอ เช่น แอดเดรสของพอร์ต 0300H นั้น ไม่ว่าเราจะอ้างแอดเดรสเป็น 0300H, 0B00H, 3B00H, หรือค่าแอดเดรสอื่นใดที่ยังคงมีแอดเดรสบิต A3-A0 เป็น 11 0000 0000 ในฐานสองอยู่ พอร์ต ๆ นี้ก็จะยังคงถูกทำให้แอกทีฟคือสามารถที่จะรับส่งข้อมูลให้กับระบบได้อยู่ ดังนั้นเราจึงต้องเลือกใช้เฉพาะแอดเดรสของพอร์ตที่แอดเดรส 10 บิตล่างยังไม่ถูกใช้งานโดยวงจรหรือการ์ดอื่นใดในระบบ

หลังจากได้แอดเดรส 10 บิตล่างนี้แล้ว จึงใช้สัญญาณที่ได้จากการดีโค้ดแอดเดรส 10 บิตล่างนี้ไปอินเวิลวงจรดีโค้ดแอดเดรส 6 บิตบนที่เหลือ จากลักษณะเช่นนี้จะเห็นได้ว่าสำหรับแอดเดรส 10 บิตล่างที่เราเลือกไว้ 1 แอดเดรส เมื่อนำมาดีโค้ดร่วมกับแอดเดรส 6 บิตบนที่เหลือ (A10-A15) แล้ว จะทำให้ได้แอดเดรสเพิ่มขึ้นเป็น 64 แอดเดรส เช่นถ้าแอดเดรส 10 บิตล่างที่เราเลือกเป็น 0213H เมื่อนำมาดีโค้ดร่วมกับแอดเดรส 6 บิตบนจะทำให้ได้แอดเดรสเพิ่มขึ้นเป็น 64 แอดเดรสคือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แอดเดรส(ฐานสิบหก)	แอดเดรส(ฐานสอง)
0213	0000 00 100011
0613	0000 01 100011
0A13	0000 10 100011
0E13	0000 11 100011
1213	0001 00 100011
::	::
::	::
::	::
0FA13	1111 10 100011
0FE13	1111 11 100011

จากตัวอย่างข้างต้นจะเห็นได้ว่าแอดเดรส 10 บิตล่างเพียงแอดเดรสเดียว เมื่อนำมาดีโค้ดร่วมกับแอดเดรส 6 บิตบนก็จะทำให้ได้แอดเดรสเพิ่มขึ้นอีก 63 แอดเดรส ( 64 - 1 ) ดังนั้นถ้าเราเลือกกลุ่มของแอดเดรส 10 บิตล่างที่มีจำนวน 8 แอดเดรส ก็จะทำให้เราได้แอดเดรสเพิ่มขึ้นอีกเป็นจำนวนถึง 502 แอดเดรส ( 63\*8=502) เมื่อรวมกับแอดเดรสเดิมอีก 8 แอดเดรส ก็จะได้จำนวนแอดเดรสทั้งสิ้นถึง 512 แอดเดรส สำหรับรูปข้างล่างนี้แสดงวงจรที่สามารถจะใช้ในการดีโค้ดในลักษณะนี้ได้

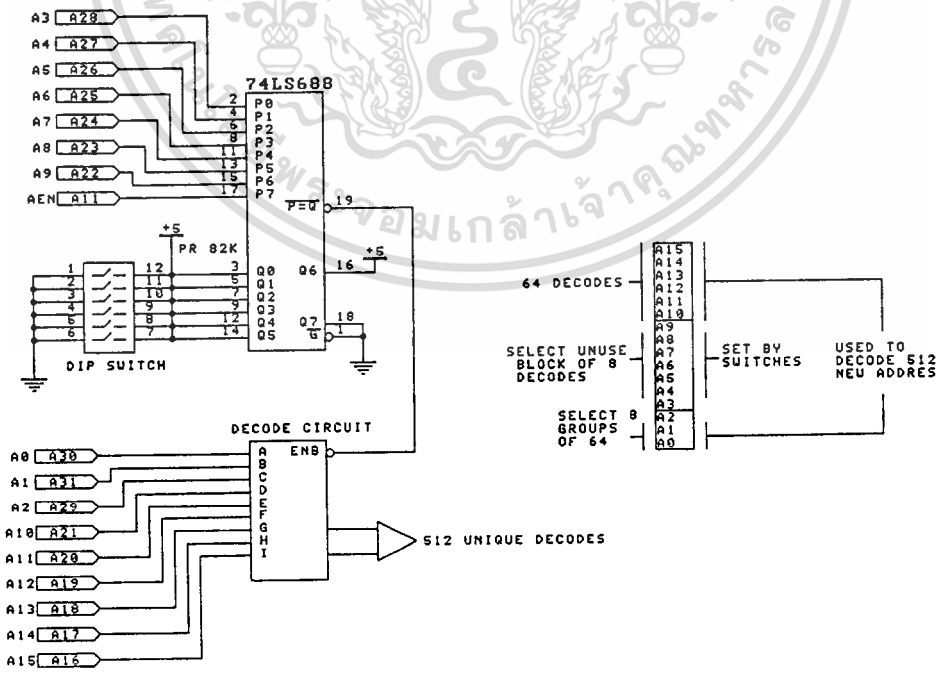


เอกสารนี้เป็นเอกสารที่สงวนไว้รูปที่ 3-8 ตัวอย่างวงจรดีโค้ดแอดเดรส 6 บิตบนให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปข้างต้นนั้นเราจะเลือกกลุ่มของแอดเดรสทั้ง 8 โดยการเซ็ท DIP Switch ที่ขาอินพุท Q0-Q5 ของ IC SN74LS688 สำหรับเอาต์พุท P=Q ซึ่งเป็นสัญญาณที่แอกทีฟเมื่อค่าแอดเดรสบิต A3-A9 บนบัสแอดเดรสของระบบนั้นตรงกับค่าที่เซ็ทไว้ที่อินพุท Q0-Q6 ของ 74LS688 นั้นจะถูกนำไปใช้ในการอินเวิลวงจรดีโค้ดแอดเดรส A0-A2 และ A10-A15 ซึ่งจะทำได้สัญญาณจากการดีโค้ดเป็นจำนวนถึง 512 เส้น อย่างไรก็ตามสัญญาณดีโค้ดเหล่านี้ยังจำเป็นต้องนำไป OR หรือ AND กับสัญญาณ IOR หรือ IOW ของระบบก่อนที่จะนำไปใช้งานจริงด้วย (วงจรมีลักษณะเป็นเพียงบล็อกไดอะแกรมอย่างง่าย ๆ เพื่อเป็นแนวทางในการออกแบบวงจรถ่านั้น ในทางปฏิบัติผู้อ่านอาจจะดัดแปลงแก้ไขเพื่อให้เหมาะสมกับสภาพของงานได้ในภายหลัง)

**การใช้วิธีอ้างแอดเดรสแบบ Indirect**

วิธีการอ้างแอดเดรสในแบบนี้เป็นวิธีที่นิยมใช้กันโดยทั่วไป ซึ่งหลักการของวิธีการอ้างแอดเดรสในแบบนี้ ก็คือ ใช้ข้อมูลของพอร์ทหนึ่งในการอ้างแอดเดรสของอีกพอร์ทหนึ่งกล่าวคือเมื่อเราส่งข้อมูลไปยังแอดเดรสของพอร์ทที่เรากำหนดให้ใช้สำหรับการอ้างแอดเดรสแบบ Indirect นี้แล้ว ข้อมูลที่เราส่งไปให้กับพอร์ทนั้น ๆ ก็จะถูกนำไปเข้าวงจรดีโค้ดอีกครั้งหนึ่ง จึงจะได้สัญญาณดีโค้ดเพื่อส่งให้กับพอร์ทต่าง ๆ ที่ต้องใช้การอ้างแอดเดรสแบบ Indirect นี้ ดังนั้นในกรณีที่บัสข้อมูลของระบบเป็นขนาด 8 บิต เมื่อใช้วิธีการอ้างแอดเดรสแบบ Indirect นี้ ทำให้เราสามารถขยายแอดเดรสของพอร์ทขึ้นได้เป็นจำนวนถึง 256 แอดเดรส



สำหรับจำนวนพอร์ทที่ใช้ในการอ้างแอดเดรสแบบ Indirect นี้เราจำเป็นต้องใช้ 2 พอร์ท คือ พอร์ทที่ถูกกำหนดให้ใช้สำหรับการอ้างแอดเดรสแบบ Indirect ซึ่งข้อมูลที่ส่งให้กับพอร์ทนี้ จะเป็นข้อมูลที่ถูกนำไปใช้ในการดีโค้ดเพื่อให้ได้สัญญาณดีโค้ดแอดเดรสของพอร์ทที่เป็นแบบ Indirect ส่วนอีกพอร์ทที่ใช้สำหรับรับหรือส่งข้อมูลให้กับพอร์ทที่เป็นแบบ Indirect นี้โดยเฉพาะ

สำหรับตัวอย่างวงจรที่ใช้สำหรับการดีโค้ดแอดเดรสแบบ Indirect นี้จะแสดงดังในรูป 3 - 9 จากตัวอย่างวงจรข้างต้นนั้น เป็นวงจรที่ใช้สำหรับการดีโค้ดแอดเดรสของพอร์ทที่เป็นแบบ Indirect 4 พอร์ท โดยแอดเดรสของพอร์ทที่ใช้ในการอ้างแอดเดรสแบบ Indirect นี้คือ แอดเดรส 0211H และ แอดเดรสพอร์ทที่ใช้ในการรับและส่งข้อมูลให้กับพอร์ทแบบ Indirect ทั้ง 4 พอร์ทก็คือแอดเดรส 0210H

จากวงจรนั้นเมื่อเราต้องการจะอ้างถึงแอดเดรสของพอร์ทแบบ Indirect นี้ ก็จะทำให้ได้โดยใช้คำสั่ง OUT ส่งข้อมูลให้กับพอร์ท 0211H โดยข้อมูลที่ส่งให้กับพอร์ท 0211H นี้จะขึ้นอยู่กับพอร์ทที่เราต้องการติดต่อด้วย เช่น ถ้าเราต้องการจะติดต่อกับพอร์ทที่ 1 ก็ส่งข้อมูลเป็น 01H ให้กับพอร์ท 0211H และ เนื่องจากในวงจรนี้เราทำการดีโค้ดเพียง 4 พอร์ท ดังนั้นข้อมูลจึงมีเพียง 4 ลักษณะคือ 01H , 02H, 03H และ 04H เมื่อพอร์ท 0211H นี้ได้รับข้อมูลที่เราส่งไปให้แล้ว ก็จะนำเอาข้อมูลบิต DO และ D1 ไปทำการดีโค้ด ซึ่งจะทำให้ได้สัญญาณดีโค้ดของทั้ง 4 พอร์ทตามที่เราต้องการ ส่วนการอ่านหรือเขียนข้อมูลลงบนพอร์ททั้ง 4 นี้ จะทำได้โดยใช้คำสั่ง IN หรือ OUT รับหรือส่งข้อมูลไปยังพอร์ท 0210H

### การอ้างแอดเดรสของพอร์ทโดยใช้วิธี Memory-Mapped

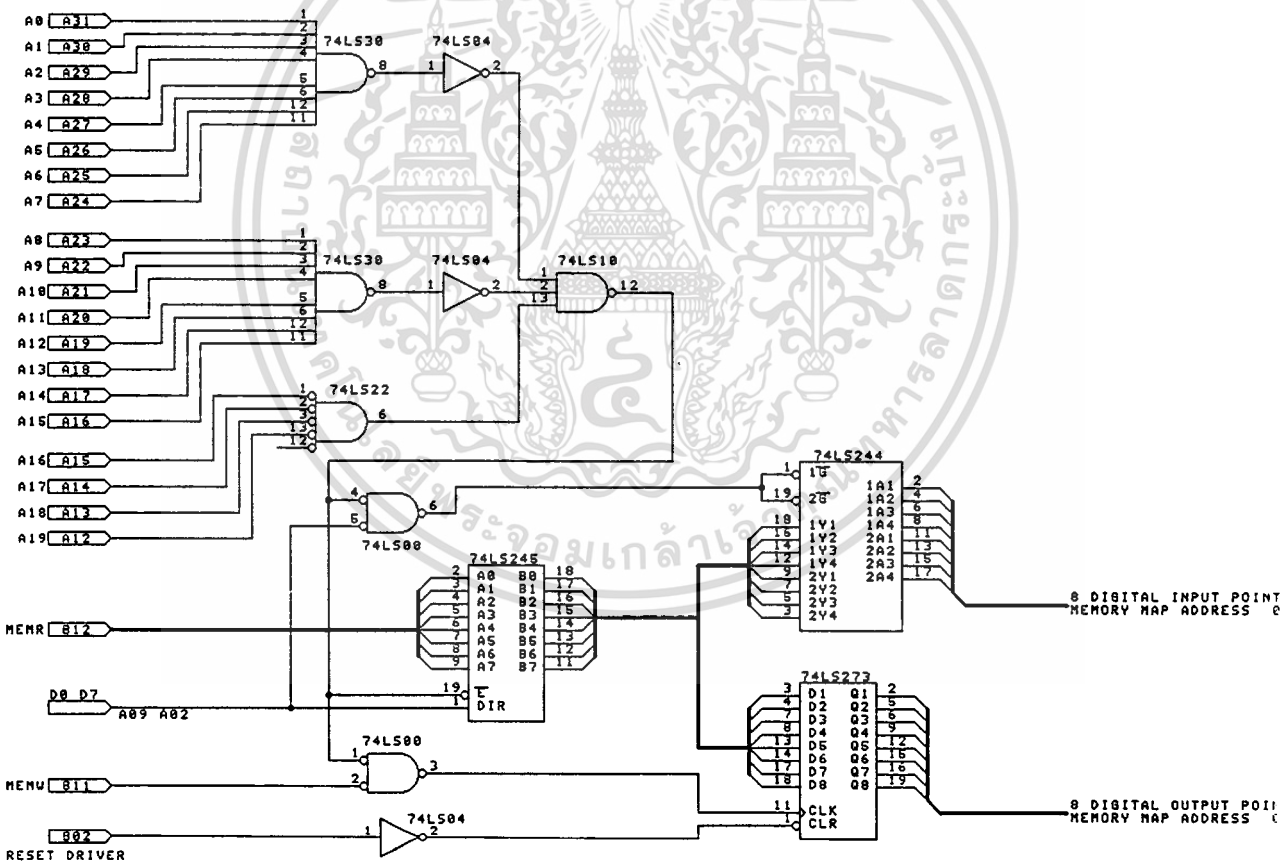
การอ้างแอดเดรสโดยวิธีต่าง ๆ ที่กล่าวถึงข้างต้นนั้น เป็นวิธีการอ้างในแบบที่เรียกว่า I/O Mapped กล่าวคือเป็นวิธีการอ้างแอดเดรสของพอร์ท I/O โดยตรง คือใช้คำสั่ง IN หรือ OUT ในการรับหรือส่งข้อมูล และ ใช้สัญญาณ IOR หรือ IOW เป็นสัญญาณควบคุมการรับส่งข้อมูล ซึ่งจะเป็นการแยกออกจากการอ้างแอดเดรสของหน่วยความจำโดยสิ้นเชิง และภายใน IBM / PC ก็ใช้วิธีการอ้างแอดเดรสของพอร์ทแบบ I/O Mapped นี้

แต่ในหัวข้อนี้เราจำกัดการอ้างแอดเดรสของพอร์ท I/O ในอีกแบบหนึ่ง ซึ่งเป็นแบบที่เรียกว่า Memory-Mapped ในการอ้างแอดเดรสของพอร์ท I/O ในแบบนี้จะเป็นการใช้แอดเดรสของหน่วยความจำเป็นแอดเดรสของพอร์ท I/O คือ การอ้างถึงแอดเดรสของพอร์ทเหล่านี้จะใช้วิธีเดียวกันกับการอ้างแอดเดรสของหน่วยความจำ เช่น ใช้คำสั่ง MOV แทนคำสั่ง IN หรือ OUT และใช้สัญญาณ MEMR และ MEMW ในการควบคุมการรับส่งข้อมูลแทนที่จะใช้ IOR หรือ IOW สำหรับวิธีการอ้างแอดเดรสแบบ Memory-Mapped นี้ ถึงแม้จะไม่มีการใช้งานใน IBM / PC ก็ตาม แต่ใน CPU บางเบอร์เช่น 6502 นั้นจะใช้ได้เฉพาะวิธีการแบบ Memory-Mapped นี้ในการอ้างแอดเดรสของพอร์ท I/O เท่านั้น ทั้งนี้เนื่องจากภายใน 6502 ไม่มีการแยกคำสั่งที่ใช้ในการจัดการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เกี่ยวกับพอร์ท I/O ไว้โดยเฉพาะ และไม่มีขาสัญญาณที่จัดไว้สำหรับควบคุมพอร์ท I/O โดยเฉพาะด้วย

การใช้วิธีการอ้างแอดเดรสแบบ Memory-Mapped นี้ จะช่วยให้เราสามารถขยายจำนวนแอดเดรสของพอร์ท I/O ออกไปได้อีกเป็นจำนวนมาก ทั้งนี้เนื่องจาก CPU 8088 สามารถจะอ้างแอดเดรสสำหรับหน่วยความจำได้เป็นจำนวนถึง 1Mbyte ในขณะที่อ้างแอดเดรสของพอร์ท I/O ได้เพียง 64K พอร์ทเท่านั้น นอกจากนี้การใช้วิธีการอ้างแอดเดรสแบบ Memory-Mapped นี้ยังทำให้เราแน่ใจได้ว่า แอดเดรสของพอร์ท I/O ที่เรากำหนดขึ้นนั้นจะไม่ตรงกับแอดเดรสของพอร์ทใด ๆ ที่ถูกใช้งานโดยวงจรหรือคาร์ตอื่น ๆ ในระบบ ทำให้การกำหนดแอดเดรสให้กับพอร์ท I/O นั้นสามารถทำได้สะดวกขึ้น สำหรับข้อดีอีกประการหนึ่งของการอ้างแอดเดรสแบบนี้ก็คือ เราสามารถที่จะนำเอาชุดคำสั่งต่าง ๆ ที่เกี่ยวกับการจัดการข้อมูลของหน่วยความจำมาใช้ในการเกี่ยวกับข้อมูลของพอร์ทที่มีการอ้างแอดเดรสแบบนี้ได้ ซึ่งจะทำการจัดการเกี่ยวกับข้อมูลของพอร์ทเหล่านี้มีประสิทธิภาพสูงขึ้น



รูปที่ 3-10 ตัวอย่างวงจรถ่ายใช้ในการตีโค้ดแอดเดรสแบบ Memory-Mapped

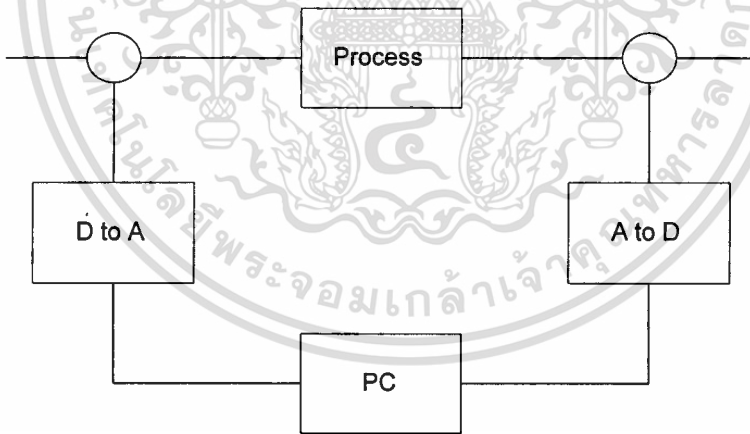
สำหรับข้อเสียของการใช้วิธีการอ้างแอดเดรสในรูปแบบเดียวกับหน่วยความจำ ดังนั้นในการตีโค้ดที่ยุ่งยากกว่าการอ้างแอดเดรสแบบ I/O Mapped และ ในการที่จะอ้างแอดเดรสของพอร์ทได้นั้น เราอาจจะจำเป็นต้องทำการเปลี่ยนแปลงค่าในรีจิสเตอร์ Segment ก่อน โดยเฉพาะในการคำนวณว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กรณีที่ว่าเซกเมนต์ของพอร์ทที่เราต้องการอ้างถึงนั้นไม่ตรงกับค่าเซกเมนต์ของโปรแกรมที่ CPU กำลังทำอยู่ ซึ่งจะทำให้ขาดความคล่องตัวในการอ้างถึงแอดเดรสของพอร์ทต่าง ๆ เหล่านี้ไปได้ นอกจากนั้นยังมีปัญหาเกี่ยวกับช่วงเวลาของบัสไซเคิลเข้ามาเกี่ยวข้องด้วย กล่าวคือช่วงเวลาในบัสไซเคิลที่เกี่ยวกับหน่วยความจำนั้น จะใช้ช่วงเวลานานเท่ากับช่วงเวลาของคล็อก 4 ลูก ในขณะที่บัสไซเคิลที่เกี่ยวกับพอร์ท I / O จะใช้ช่วงเวลานานเท่ากับช่วงเวลาของคล็อกถึง 5 ลูก ดังนั้นจึงทำให้การทำงานของพอร์ท I / O ที่มีการอ้างแอดเดรสแบบ Memory - Mapped ต้องมีความเร็วสูงกว่าอุปกรณ์ที่ใช้ในการอ้างแอดเดรสแบบ I / O Mapped

สำหรับรูปที่ 3 - 10 จะแสดงตัวอย่างวงจรที่ใช้ในการดีโค้ดแอดเดรสแบบ Memory Mapped ให้กับพอร์ท I / O

### ลักษณะทั่ว ๆ ไปของ CARD IO

IO Card เป็น Card ที่ใช้เชื่อมต่อกับเครื่อง PC เพื่อทำการแปลงสัญญาณ Analog เป็น Digital และ Digital เป็น Analog โดยจะรับสัญญาณ Analog จาก Process มาแปลงเป็นสัญญาณ Digital เครื่อง PC จะนำสัญญาณที่ได้ไปคำนวณเพื่อควบคุม Process เมื่อได้ผลจากการคำนวณก็จะทำการแปลงสัญญาณ Digital ที่ได้ไปเป็นสัญญาณ Analog แล้วจึงส่งสัญญาณที่ได้ไปควบคุม Process ดังรูป



ซึ่งใน Card IO จะมีอุปกรณ์ร่วมและหน้าที่ของตัวมันดังนี้

- IC ADC574A (Analog To Digital Converter)

ทำหน้าที่รับสัญญาณ Input ในรูปของ Analog แล้วเปลี่ยนไปเป็น Digital เพื่อส่งให้ CPU ประมวลผล ซึ่งสามารถที่จะประยุกต์ใช้งานในการเชื่อมต่อ (Interface) กับอุปกรณ์ภายนอกที่ให้สัญญาณเป็น Analog สำหรับ ADC574A สามารถรับสัญญาณ Analog ได้โดยตรง 10 Volts หรือ Range 20 Volts และเปลี่ยนไปเป็นสัญญาณ Digital ขนาด 12 Bits

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาดูเท่านั้น เมื่ออนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



- IC DAC1210 (Digital To Analog Converter)

ทำหน้าที่รับสัญญาณ Digital จาก CPU ที่คำนวณเสร็จแล้วมาแปลงเป็นสัญญาณ Analog เพื่อส่งไปควบคุม Process สำหรับ ADC1210 จะรับสัญญาณ Input ขนาด 12 Bits แล้วทำการแปลงเป็นกระแสจากนั้นจะผ่านวงจร OP-AMP เพื่อเปลี่ยนกระแสเป็น Volts แล้วนำไปควบคุม Process

- วงจร DECODER / PORT

ทำหน้าที่ถอดรหัสตำแหน่งของ Port ที่จะใช้งาน และสะดวกในการเปลี่ยนตำแหน่งของ Port โดยการ Set Dip - Switch ซึ่งจะทำให้ Card IO สามารถเชื่อมกับเครื่อง PC ได้มากกว่า 1 Card โดยกำหนดให้ Port ที่ใช้งานแตกต่างกัน

การ DECODE PORT

ตำแหน่ง Port บน Card IO จะใช้ IC TTL 74LS688 , 74LS139 , 74LS32 และ Switch DIP - 12 ต่อร่วมกันเป็นวงจร DECODE PORT โดยใช้ DIP-Switch เป็นตัวกำหนด Port ตามความต้องการของผู้ใช้งาน

สำหรับ Card IO จะใช้ตำแหน่งของ Port ดังนี้

X308 = Port Control ของ DAC

X309 = Port Control ของ DAC

X30A = Port Control ของ ADC

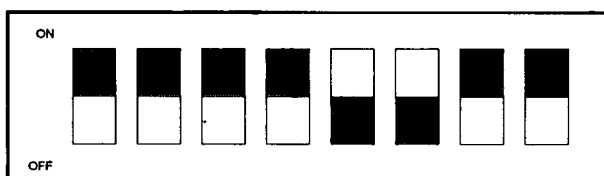
X30B = Port Control ของ ADC

โดยเราจะกำหนดระดับ Logic ให้กับตำแหน่ง Address นั้น ๆ ตามความต้องการ ซึ่งบน Card IO สามารถที่จะกำหนดระดับของ Logic ให้กับตำแหน่งของ Address โดยใช้ DIP Switch

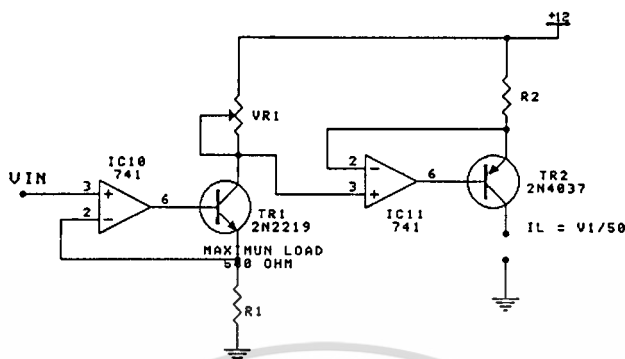
หาก ON Switch จะได้รับระดับ Logic "0" หาก OFF Switch จะได้รับระดับ Logic "1" ซึ่งตำแหน่ง Address ของ Card IO จะเป็น X30X จะได้

A11	A10	A9	A8		A7	A6	A5	A4		A3	A2	A1	A0
0	0	1	1		0	0	0	0		X	X	X	X
3					0					0			

เพราะฉะนั้นสามารถ Set DIP Switch ได้ดังรูป



## - วงจรแปลงแรงดันเป็นกระแส



## วงจรแปลงแรงดันเป็นกระแส

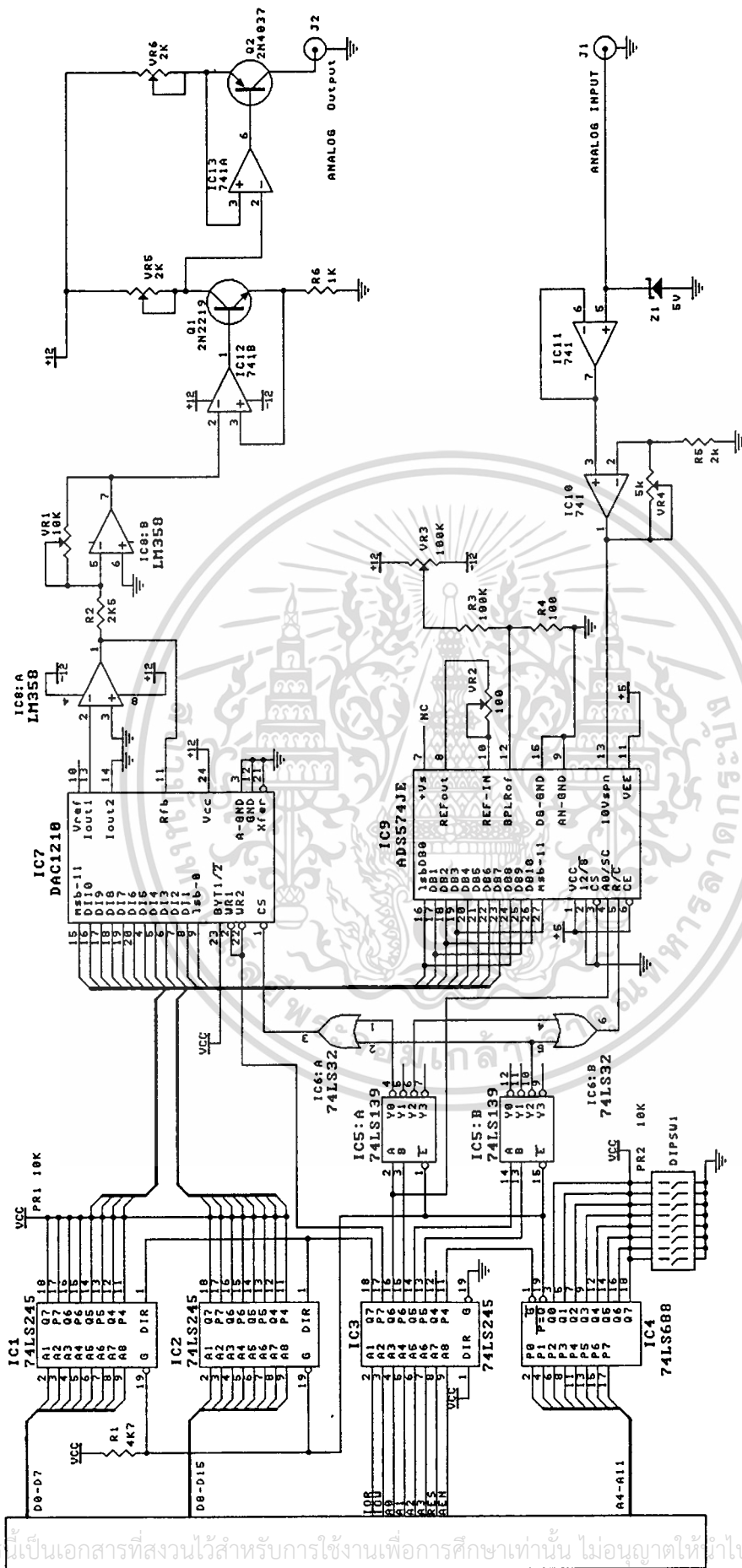
คุณสมบัติทางอุดมคติของ OP-AMP ซึ่งมีคุณสมบัติ 2 ประการ คือ

- 1) แรงดันที่ตกคร่อมระหว่างขั้ว Inverting และ Noninverting มีค่าประมาณ 10 Volts
  - 2) กระแสที่ไหลเข้า OP-AMP มีค่าประมาณ 0 AMP เนื่องจากคุณสมบัติทั้ง 2 ประการ
- แรงดันที่ตกคร่อมความต้านทาน 1 กิโลโอห์ม มีค่าเท่ากับ  $V_{in}/1000$  ส่วนแรงดัน  $V_1$  ที่ตกคร่อม  $VR_1$  มีค่าเท่ากับ  $VR_1$  คูณ กระแสที่ไหลผ่านความต้านทานบวกกับกระแสที่ไหลผ่านทรานซิสเตอร์เข้าออปแอมป์ด้วย

ทรานซิสเตอร์ 2N2219 ต่อเพื่อให้สามารถจ่ายกระแสได้มากขึ้น เนื่องจาก เอ๊าท์พุท ของออปแอมป์ไม่สามารถจ่ายกระแสได้มาก

ทางด้านเอ๊าท์พุทความต้านทานมีค่า 50 โอห์ม ต่ออนุกรมกับทรานซิสเตอร์ 2N4037 เป็นการกำหนดภาระที่นำมาต่อใช้งานได้สูงสุดไม่เกิน 500 โอห์ม โดยสมมุติ ให้แรงดันตกคร่อม  $V_{ce}$  เท่ากับ 1 V. และที่ค่ากระแสสูงสุดคือ 20 มิลลิแอมป์ ซึ่งตกคร่อมความต้านทาน 50 โอห์ม เท่ากับ 1 V. ทรานซิสเตอร์ 2N4037 สามารถจ่ายกระแสได้มากขึ้น

การปรับ  $VR_1$  มีวิธีการดังนี้ ขั้นแรกป้อนแรงดัน  $V_{in}$  เท่ากับ 5 โวลต์ แล้วปรับให้ได้แรงดันตกคร่อม  $VR_1$  เท่ากับ 1 โวลต์ นั่นคือแรงดัน  $V_1$  และแรงดันที่ตกคร่อมความต้านทาน 50 โอห์ม เท่ากับ 1 โวลต์ ด้วย ซึ่งขณะนี้สามารถจ่ายกระแสให้กับ โหลด ได้ถึง 20 มิลลิแอมป์ และ โหลดต้องมีค่าความต้านทานไม่เกิน 500 โอห์ม



INPUT/OUTPUT CARD

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้ไปใช้ประโยชน์ด้านการค้า  
 ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทที่ 4

### โปรแกรมควบคุมกระบวนการและการแสดงผล

โปรแกรมควบคุมกระบวนการและแสดงผล ที่ได้เขียนขึ้นนี้จะเป็นโปรแกรมที่ทำงานบน Microsoft Windows เท่านั้น จะไม่สามารถนำมาทำงานบน DOS (Disk Operating System) ได้เลย ซึ่งการนำโปรแกรมควบคุมกระบวนการและแสดงผลมาทำงานบน Microsoft Windows นี้มีข้อดีกว่าการทำงานบน DOS เนื่องจาก Microsoft Windows เป็นระบบที่สามารถทำงานได้หลายงานพร้อมกัน แต่เป็นระบบหลายงานในลักษณะที่เรียกว่า Non - Preemptive นั่นคือ ความสามารถในการทำงานหลายงานต้องได้จากความร่วมมือของทุก ๆ Application ที่ทำงานในขณะนั้น นั่นคือทุก ๆ Application ที่ทำงานในขณะนั้นต้องไม่มีการยึดครองทรัพยากรของระบบเพียง Application เดียว (ทรัพยากรหมายถึง CPU, Memory, Printer, Keyboard เป็นต้น) จะต้องแบ่งทรัพยากรมาใช้ร่วมกัน โดยเมื่อ Application ใช้ทรัพยากรจนเสร็จเรียบร้อยแล้ว ต้องคืนทรัพยากรให้ระบบกลับคืนไปเพื่อให้ Application อื่นนำไปใช้งานได้

#### ความสามารถของโปรแกรม

โปรแกรมควบคุมกระบวนการและแสดงผลเป็นโปรแกรมที่สร้างขึ้นสำหรับการควบคุมกระบวนการ โดยสามารถเลือกโหมดการควบคุมได้ 3 Mode คือ

- PID Control
- PID-FUZZY Control
- Manual Control

ซึ่งทั้ง 3 Mode การควบคุมนี้จะมีการคำนวณที่ต่างกันอย่างออกไปตามโหมดนั้น ๆ ส่วนการแสดงผลสามารถแบ่งออกเป็น 2 ส่วน ซึ่งจะแสดงผลพร้อมกันทั้ง 2 ส่วนพร้อมกันคือ

- ส่วนแสดงผลเป็นตัวเลข
- ส่วนแสดงผลเป็นกราฟ

ในส่วนของการแสดงผลเป็นตัวเลข จะแสดงพารามิเตอร์ ต่าง ๆ ของกระบวนการตามโหมดการควบคุม และจะมีการ Update ข้อมูลทุก 1 วินาที ซึ่งค่าที่แสดงผลนี้ สามารถแสดงผลได้ทศนิยมสูงสุด 3 ตำแหน่ง โดยการแสดงผลเป็นตัวเลขจะแสดงค่าของพารามิเตอร์ ณ เวลานั้น ๆ เท่านั้น โดยมีพารามิเตอร์ต่าง ๆ ดังนี้

## โหมด PID Control

P I D	SP :	D/R :	PB :	H-range :
	PV :	√ :	Ti :	L-range :
	E :	TS :	Td :	H-alarm :
	CS :			L-alarm :

SP	Set Point
PV	Process Variable
E	Error ( $E = SP - PV$ )
CS	Control Signal
D/R	Direct / Reverse of Output
√	Square root of PV
Ts	Sampling Period Time
PB	Proportional Band
Ti	Integral Time
Td	Derivative Time
H Range	High Range
L Range	Low Range
H Alarm	High Alarm
L Alarm	Low Alarm

## โหมด PID - FUZZY Control

F U Z Z Y P & I D	PU :	D/R :	PB :	H-range :
	TU :	√ :	Ti :	L-range :
	SP :	E :	Td :	H-alarm :
	PV :	CS :	TS :	L-alarm :

PU	Ultimate Gain
TU	Ultimate Period Time
SP	Set Point
PV	Process Variable
D/R	Direct / Reverse of Output
√	Square root of PV
E	Error ( $E = SP - PV$ )

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์การใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

CS	Control Signal
PB	Proportional Band
Ti	Integral Time
Td	Derivative Time
Ts	Sampling Period Time
H Range	High Range
L Range	Low Range
H Alarm	High Alarm
L Alarm	Low Alarm

### โหมด Manual Control



SP	Set Point
PV	Process Variable
D/R	Direct / Reverse of Output

ในส่วนของการแสดงผลเป็นกราฟ จะแสดงค่าของพารามิเตอร์ด้วยกัน 3 ค่า คือ

SP	Set Point
PV	Process Value
CS	Control Signal

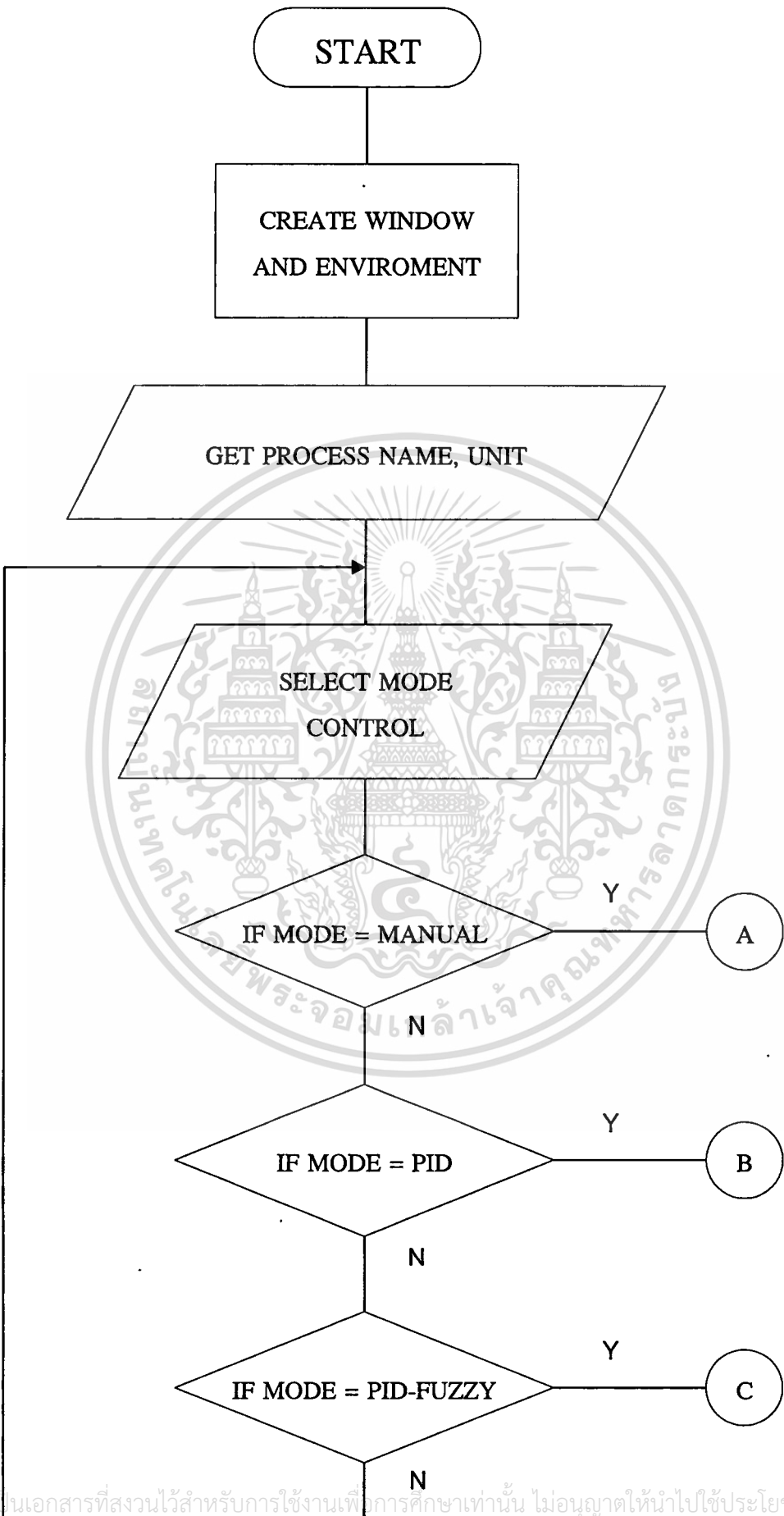
ซึ่งส่วนของการแสดงผลเป็นกราฟนี้จะไม่ขึ้นกับโหมดการควบคุม แม้ว่าจะมีการเปลี่ยนโหมดการควบคุมก็ยังคงแสดงผลต่อไป การแสดงผลเป็นกราฟนี้จะมีการ Update ข้อมูลทุก ๆ 1 วินาที และแสดงผลได้สูงสุดถึง 250 ตำแหน่ง นั่นคือ จะสามารถแสดงผล ณ เวลานั้น ๆ และเวลาก่อนหน้านั้นอีก 249 ตำแหน่ง (249 วินาที)

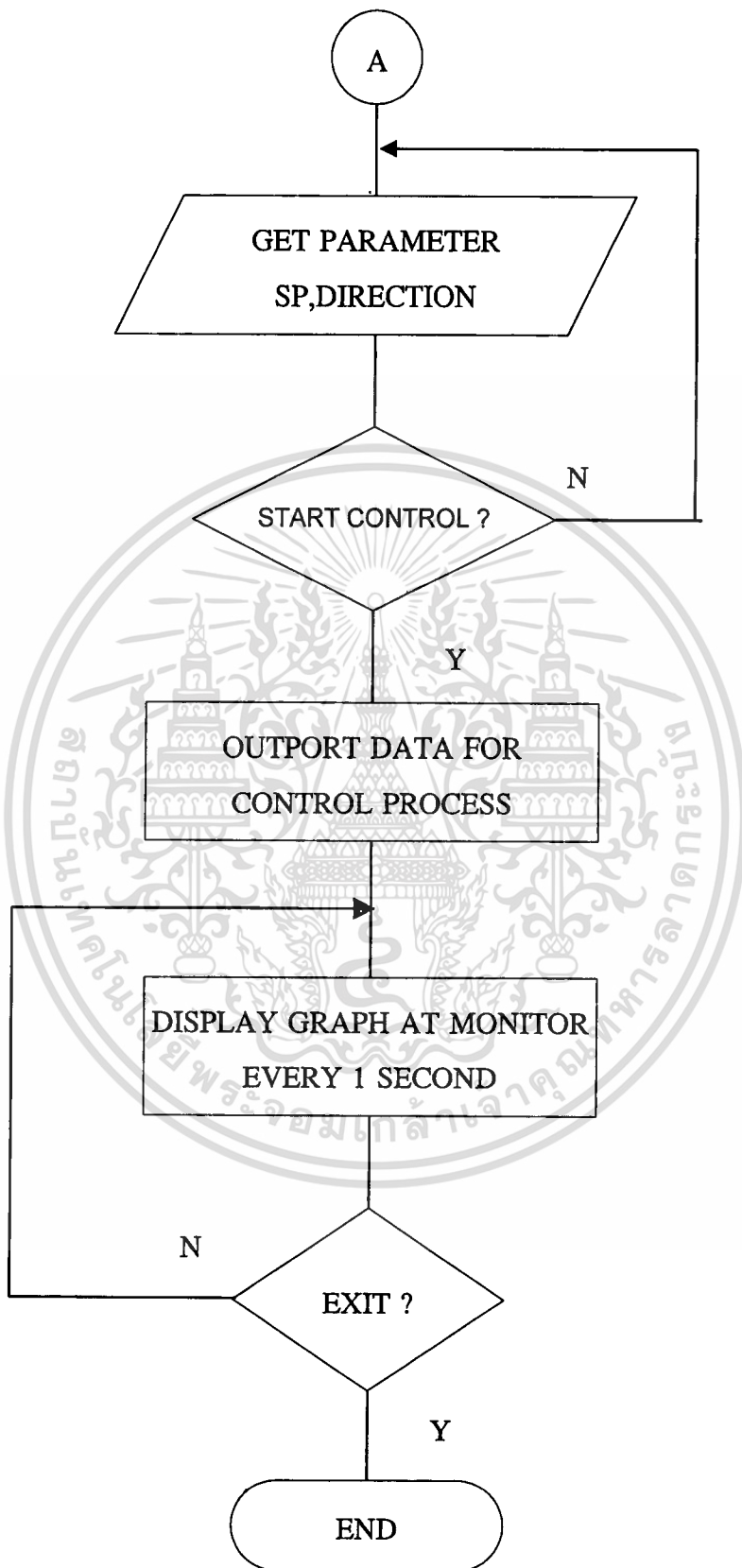


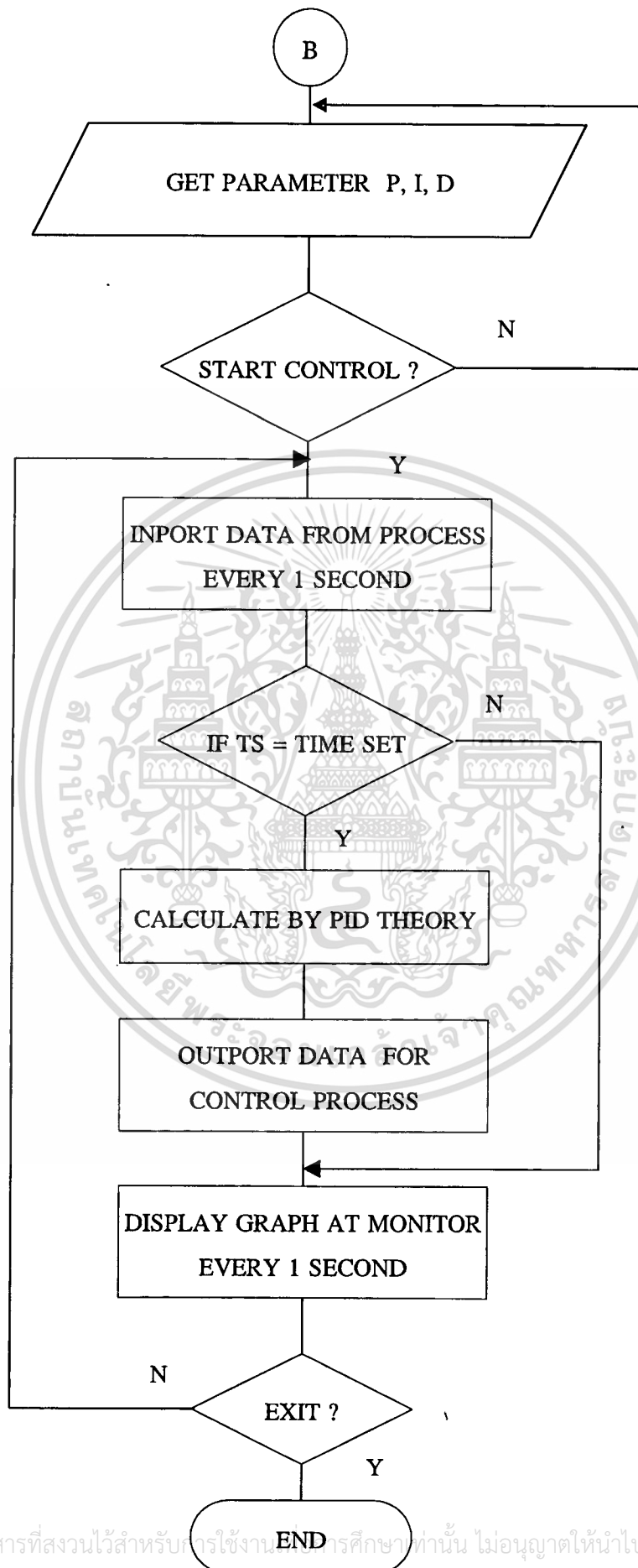
### การทำงานของโปรแกรม

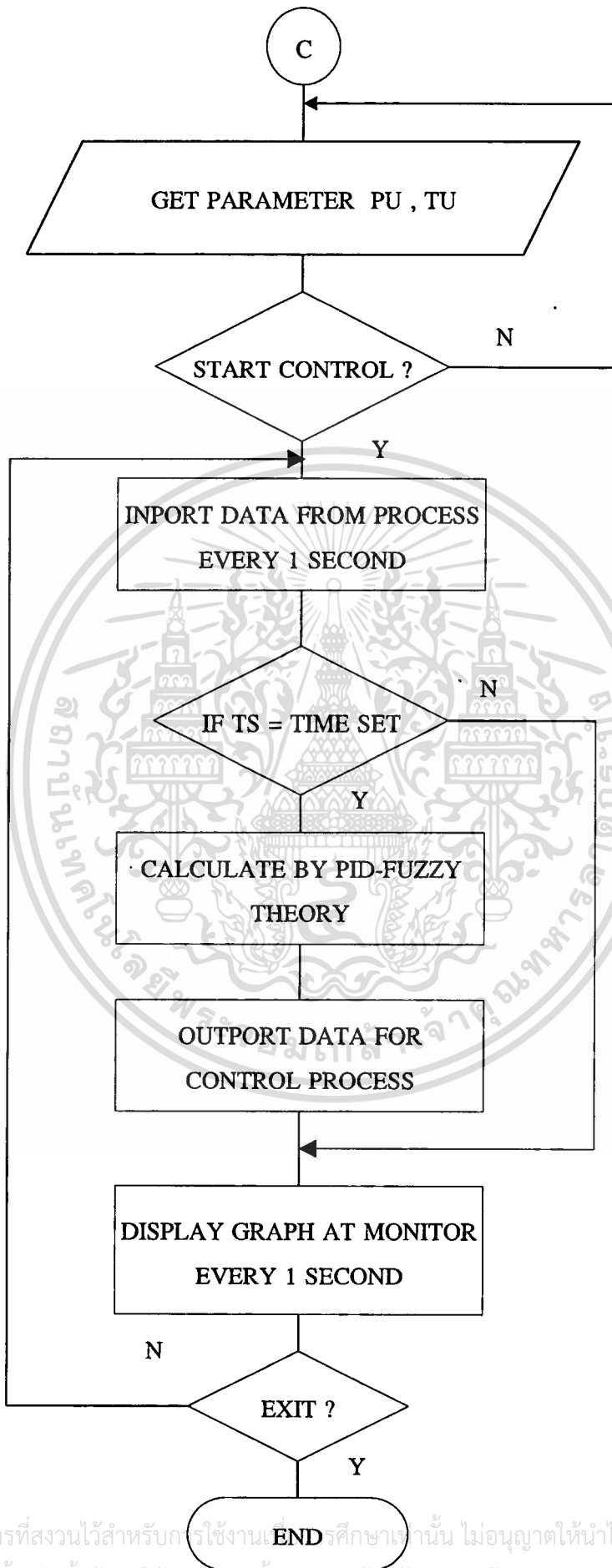
โปรแกรมควบคุมกระบวนการและแสดงผล เป็น โปรแกรมที่เขียนขึ้นโดยใช้โปรแกรม Borlandc C++ Version 3.1 For Windows ซึ่งโปรแกรมควบคุมกระบวนการและแสดงผลนี้ จะประกอบด้วยส่วนต่าง ๆ ดังนี้

- Monitor.C** เป็นไฟล์ที่ใช้กำหนดเงื่อนไขต่าง ๆ ของโปรแกรม พร้อมทั้งส่วนในการคำนวณ สมการการควบคุม PID และ PID-FUZZY ด้วย
- Monitor.RES** เป็นไฟล์ใช้สำหรับเก็บ Enviroment ต่าง ๆ ของโปรแกรมโดยจะเก็บ Menu, Icon, Bitmap, Dialog และ Cursor เพื่อใช้ในส่วนการสนับสนุนการทำงานของโปรแกรม
- Monitor.DEF** เป็นไฟล์ที่ใช้สำหรับ กำหนดสิ่งต่าง ๆ ที่ใช้ร่วมกับโปรแกรม รวมถึงการจัดการ Memory ด้วย
- Monitor.H** เป็นไฟล์ที่ใช้สำหรับกำหนดฟังก์ชันย่อยต่าง ๆ ในการทำงานของโปรแกรม รวมถึงการกำหนดตัวแปรที่ใช้สำหรับการอ้างถึง File Monitor.RES ด้วย
- Monitor.PRJ** เป็นไฟล์ที่ใช้สำหรับ การรวบรวมไฟล์ต่าง ๆ เข้ารวมกัน เพื่อใช้ในการ Compile และ Link เพื่อให้เกิด Execute File
- Monitor.EXE** เป็นไฟล์ ได้จากการ Compile และ Link ทุก ๆ ไฟล์ข้างต้นเข้าด้วยกัน แล้วและจะเป็นไฟล์ที่ใช้สำหรับกรนำไปใช้งานการทำงานของโปรแกรม สามารถแสดงเป็น Flowchart ได้ดังนี้



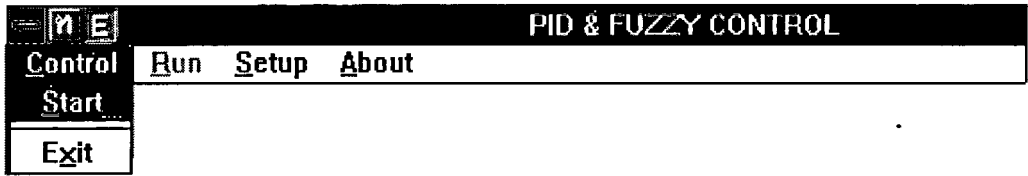




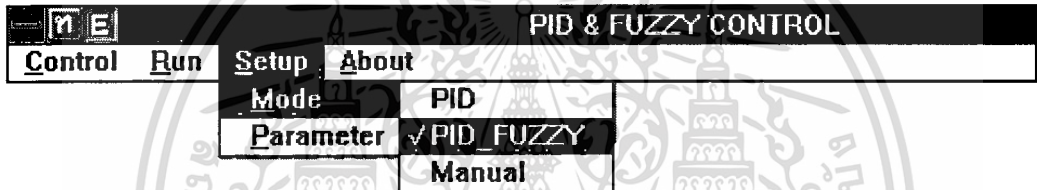


## การใช้งาน โปรแกรม

ในส่วนนี้ จะกล่าวถึงการใช้งานของโปรแกรมว่าใช้งานได้อย่างไร โดยมี Menu ดังนี้



- File - New ใช้สำหรับการกำหนดชื่อ และ หน่วยของ Process โดยการเริ่มต้นโปรแกรม ทุก ๆ ครั้งต้องกำหนดชื่อ และ หน่วยของ Process ก่อนมิฉะนั้นจะไม่สามารถเลือก Menu Setup-Parameter ได้
- File - Exit ใช้ เมื่อต้องการออกจากโปรแกรม

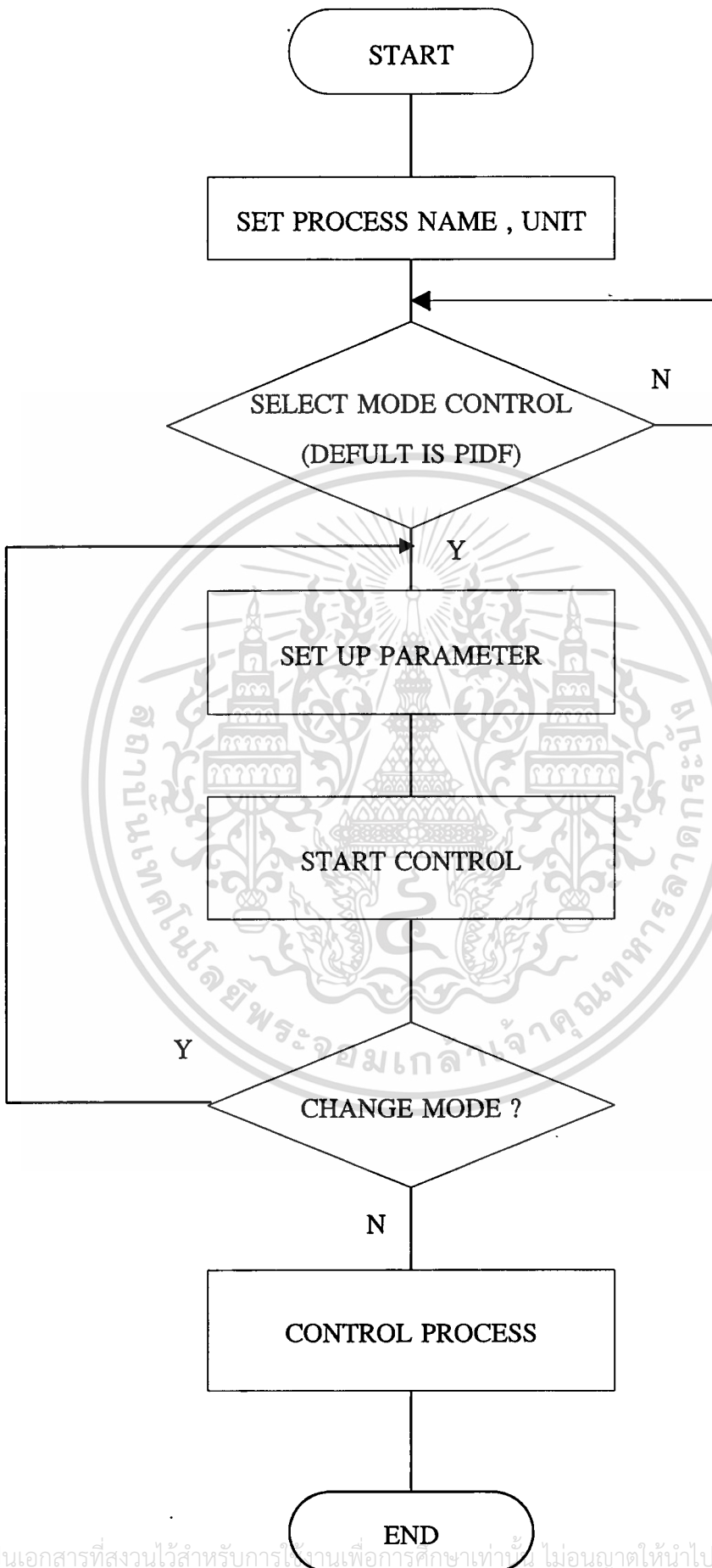


- RUN ใช้ สำหรับ เริ่มต้นการควบคุมกระบวนการ โดย Menu Run จะสามารถทำงานได้ต่อเมื่อ มีการ Setup Parameter แล้ว
- Set - Mode - PID ใช้สำหรับการเลือกโหมดการควบคุมกระบวนการให้เป็นแบบ PID
- Set - Mode - PID\_FUZZY ใช้สำหรับการเลือกโหมด การควบคุมกระบวนการให้เป็นแบบ PID - FUZZY
- Setup - Mode - Manual ใช้สำหรับการเลือกโหมด การควบคุมกระบวนการใช้เป็นแบบ Manual
- Setup - Parameter ใช้สำหรับการ Setup Parameter ต่าง ๆ ของกระบวนการซึ่งจะเป็น Parameter ของโหมดการควบคุม ณ ขณะนั้น
- Help ใช้สำหรับดูส่วนที่เกี่ยวข้องกับ โปรแกรม

ซึ่งการใช้งาน โปรแกรมสามารถแสดงลำดับขั้นตอนการดำเนินการได้ดัง Flow Chart

ต่อไปนี้เป็น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



# บทที่ 5

## ผลการทดลอง

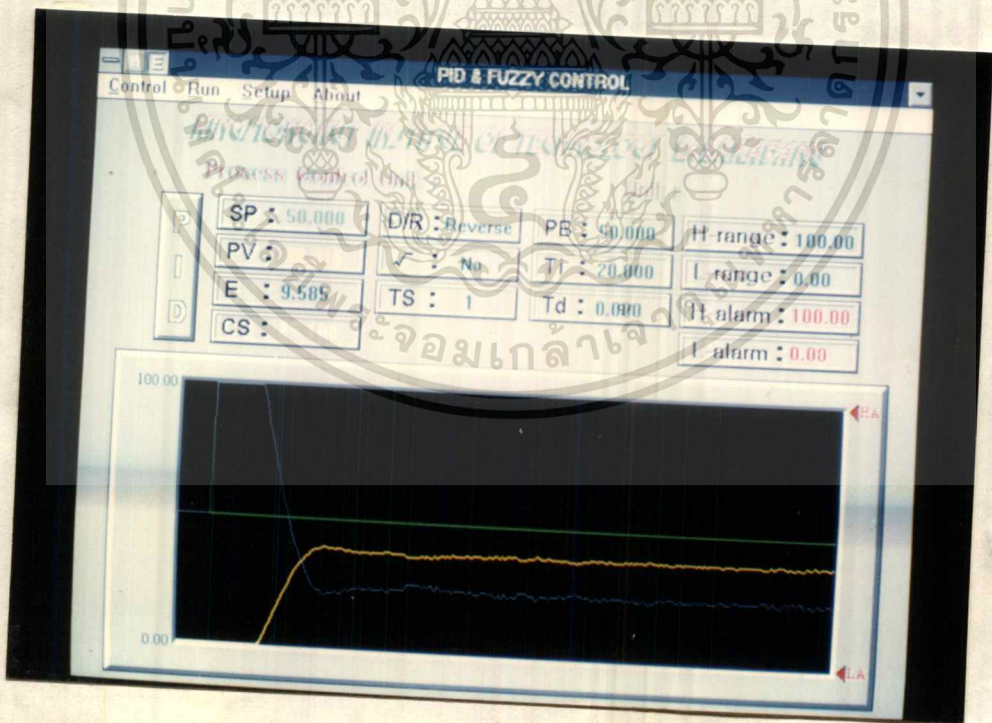
### โหมด PID

ในโหมด PID Control นี้เราจะทำการ Tuning Parameter PID 2 วิธีด้วยกัน คือ

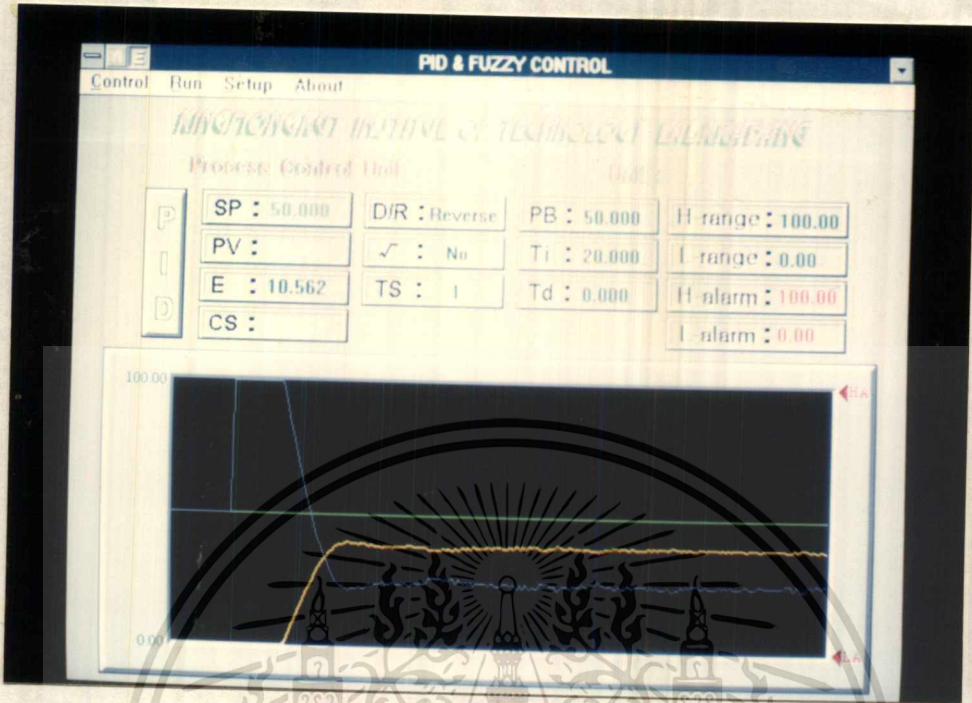
1. วิธีการสุ่มทางปฏิบัติ
2. วิธี Zeigler Nichols

#### 1. วิธีการสุ่มทางปฏิบัติ

- 1.1 ทดลองป้อนค่า PB อย่างเดียว ซึ่งจะทำให้เกิดค่าความผิดพลาดออฟเซต ดังรูป 5.1 และรูป 5.2

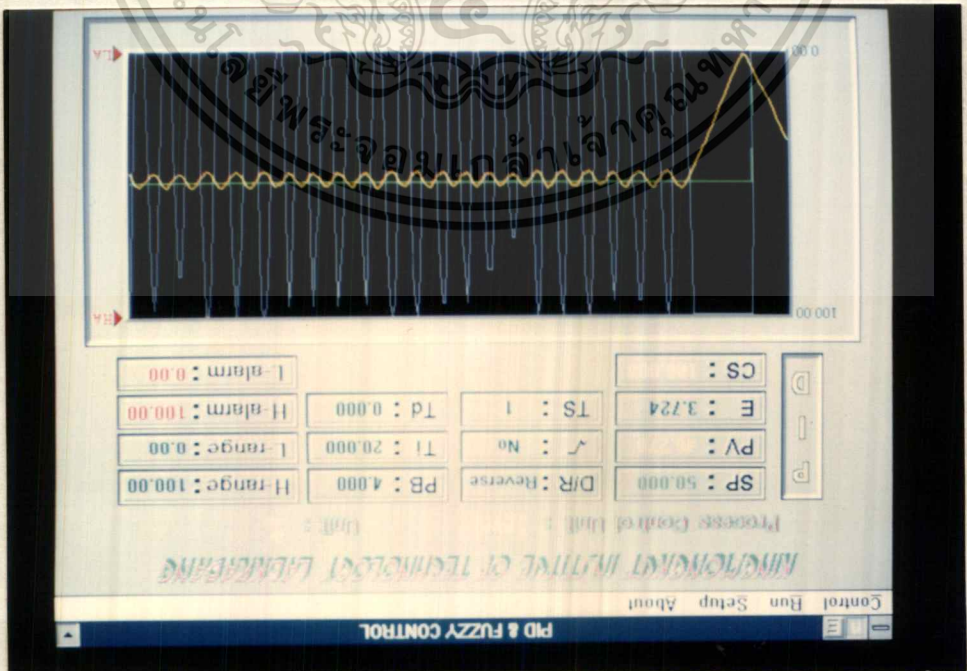


รูปที่ 5.1



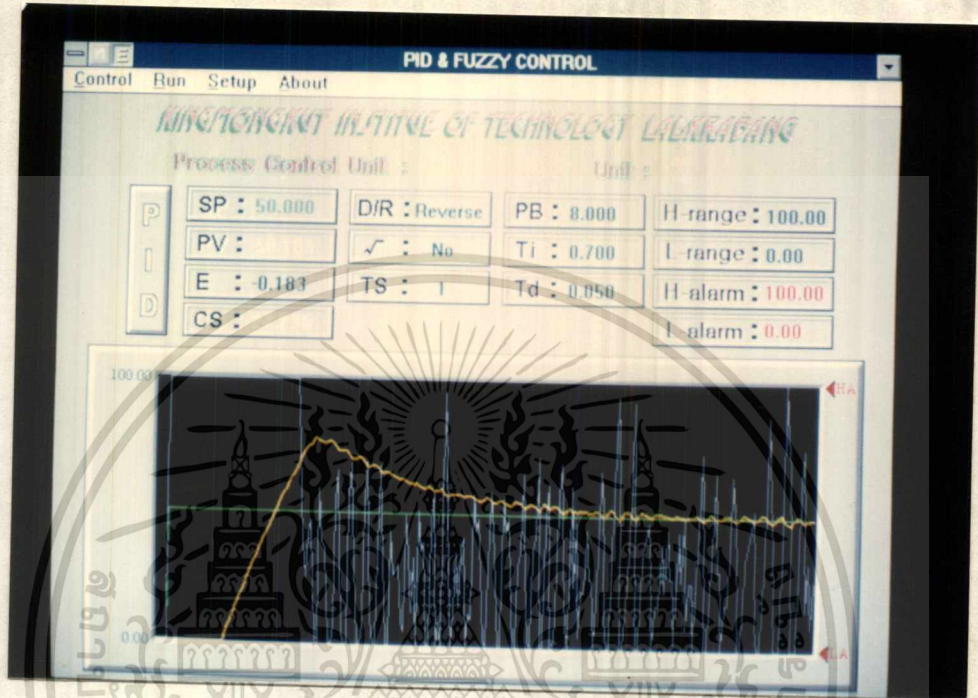
รูปที่ 5.2

1.2 ทำการลดค่า PB ลง จนทำให้กระบวนการเกิดเปลี่ยนแปลงแบบ ออสซิลเลท ซึ่งค่า PB ที่ได้ก็คือค่า KU ( $KU=100/PB$ ) ดังรูป 5.3



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานรูปที่ 5.3 ศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.3 ควบคุมกระบวนการด้วยผู้ควบคุม โดยป้อนค่า PB,Ti,Td ตามความเหมาะสม  
ของผู้ควบคุมเอง ดังรูป 5.4



รูปที่ 5.4

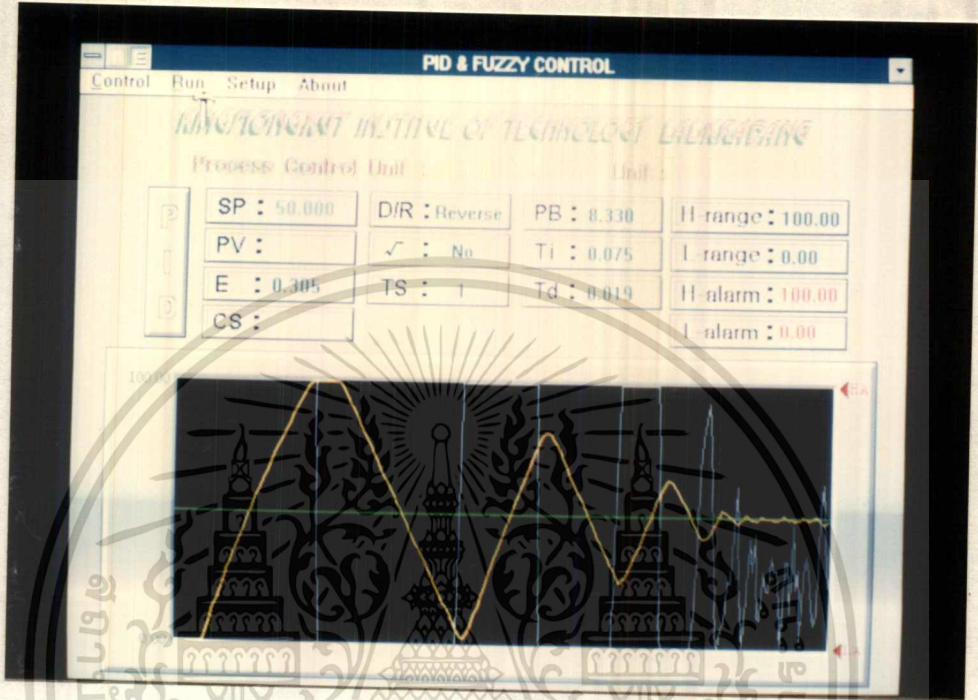
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. วิธี สูตร Zeigler Nichols ซึ่งใช้  $PU=5, TU=1.2$  ดังรูป 5.5, 5.6, 5.7, 5.8

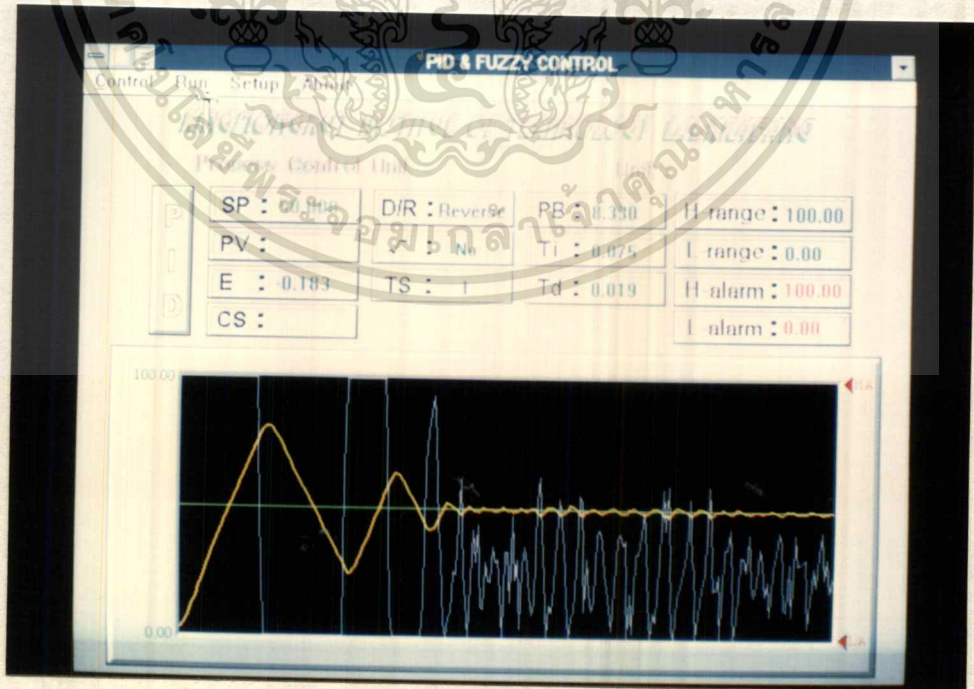
$$K_p = 0.6 K_u \tag{5.1}$$

$$T_i = 0.5 T_u \tag{5.2}$$

$$T_d = 0.125 T_u \tag{5.3}$$



รูปที่ 5.5



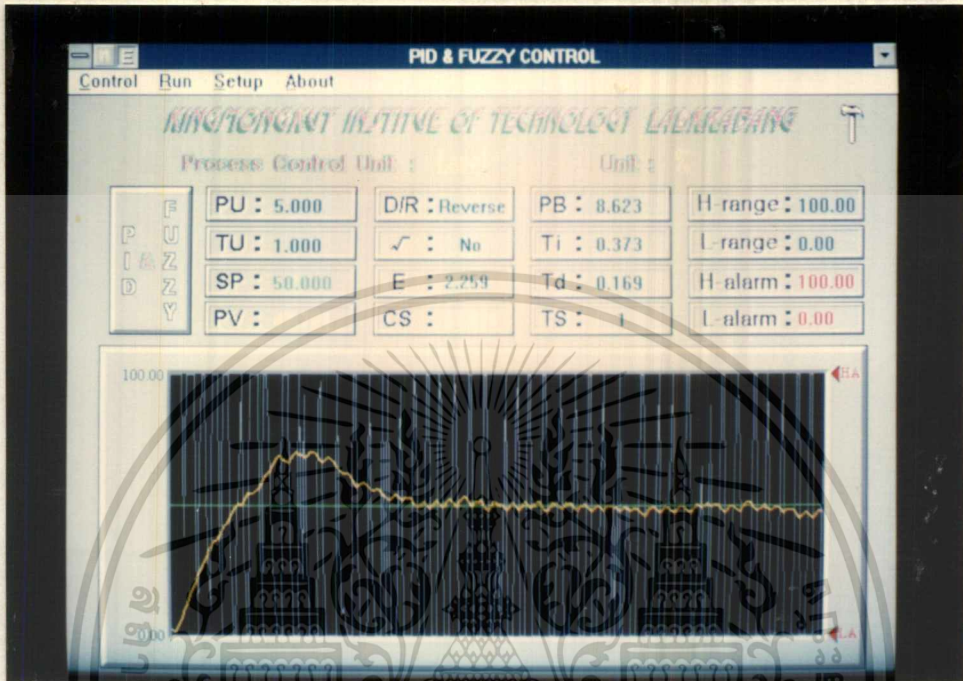
รูปที่ 5.6

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



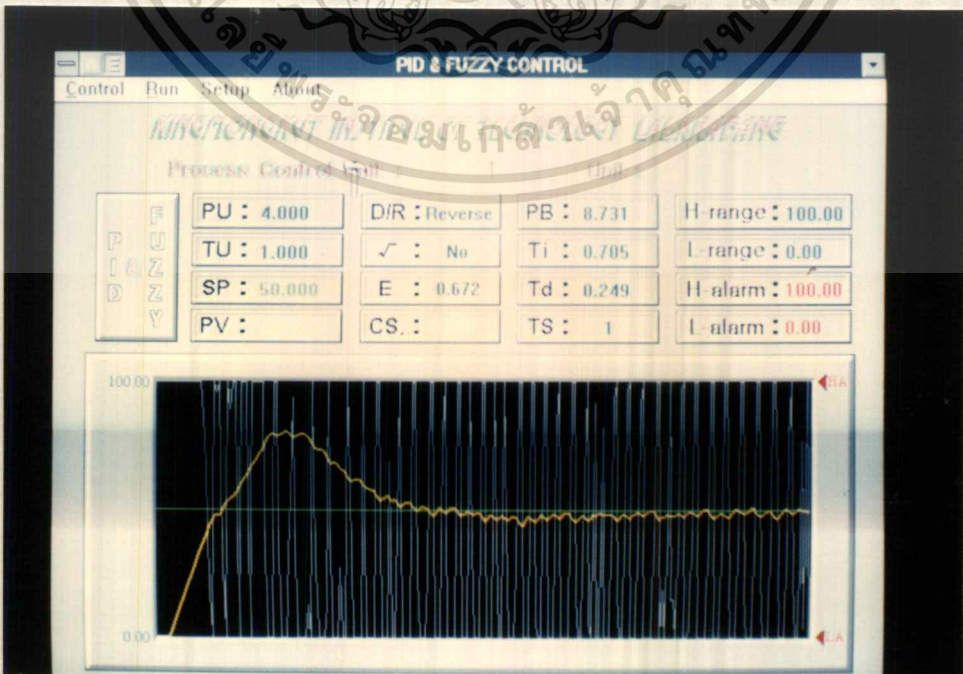
## โหมด PID-FUZZY

ในการควบคุมในโหมดนี้ ค่าที่เราป้อนมีเพียง 2 ค่า คือ  $PU=5$ ,  $TU=1$  ดังรูป 5.9



รูปที่ 5.9

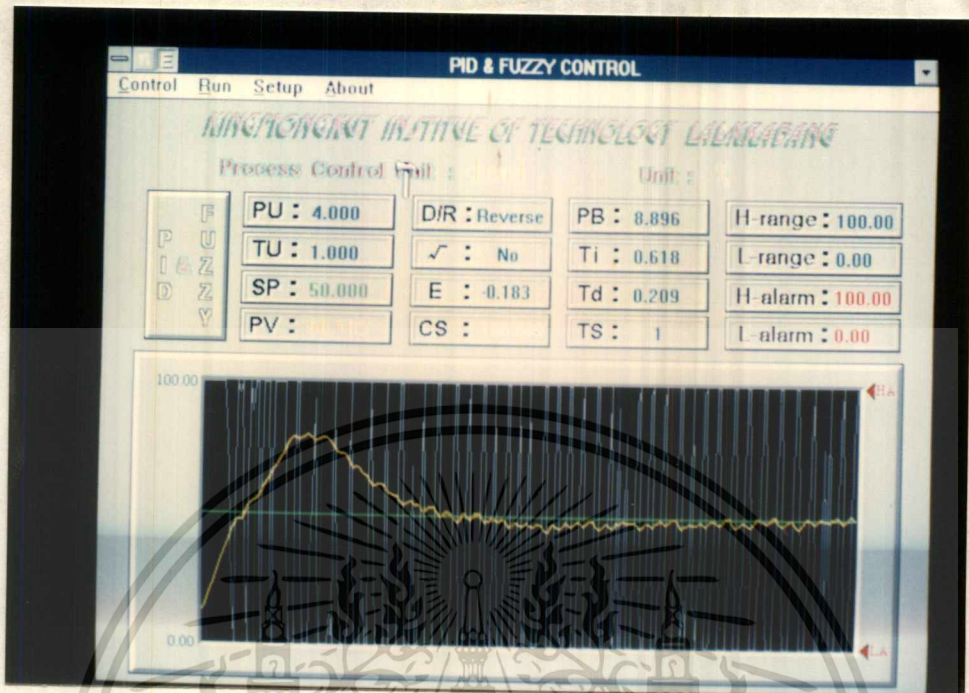
ทดลองป้อนค่า  $PU=4$ ,  $TU=1$  ได้ผลการทดลองดังรูป 5.10



รูปที่ 5.10

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

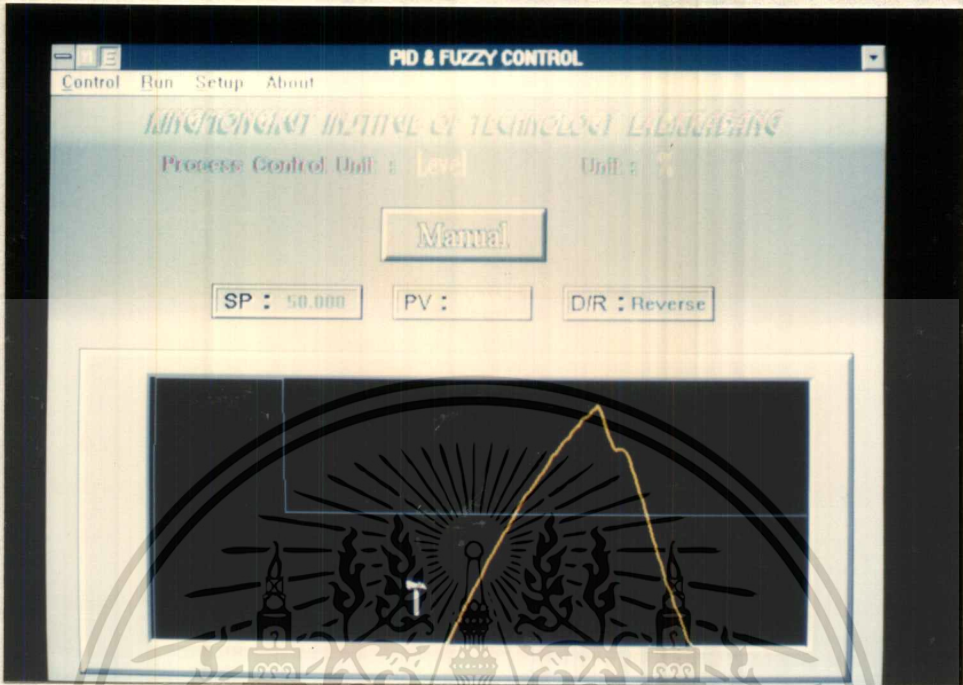
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



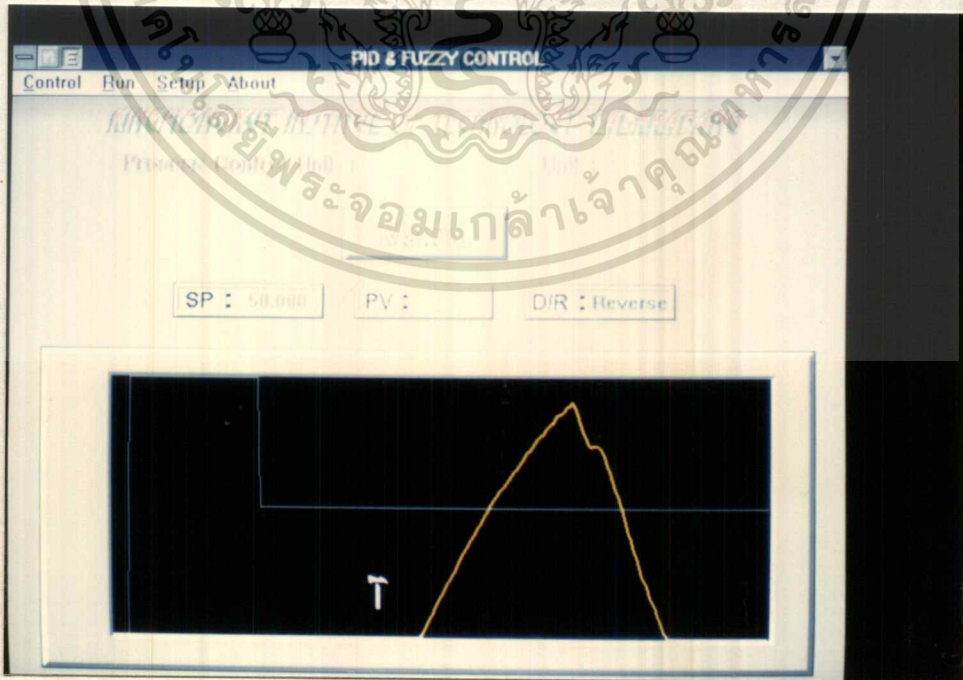
รูปที่ 5.11

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โหมด Manual



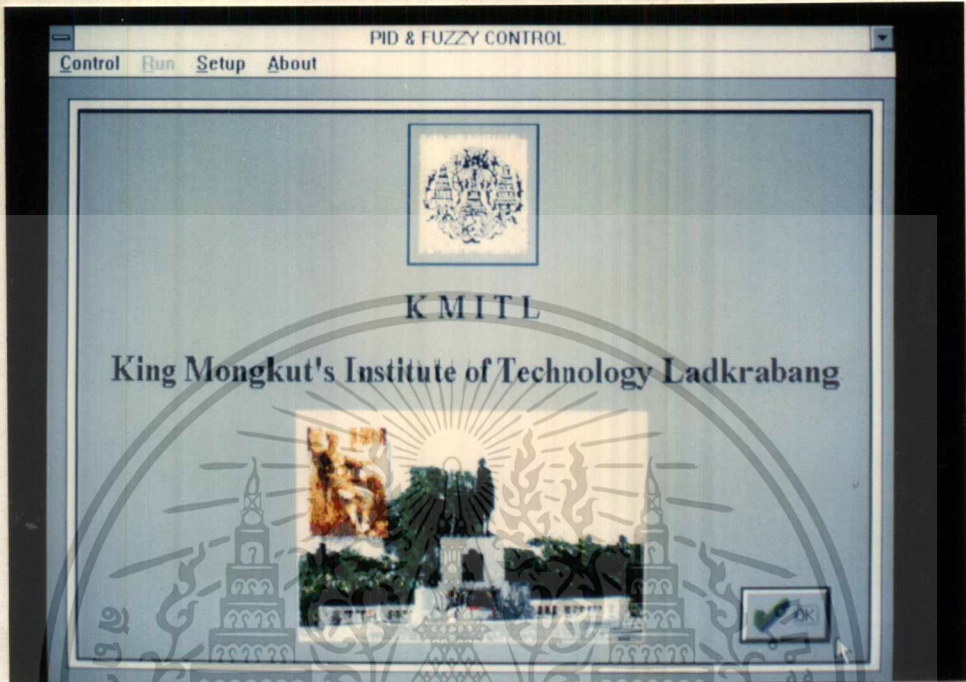
รูปที่ 5.12



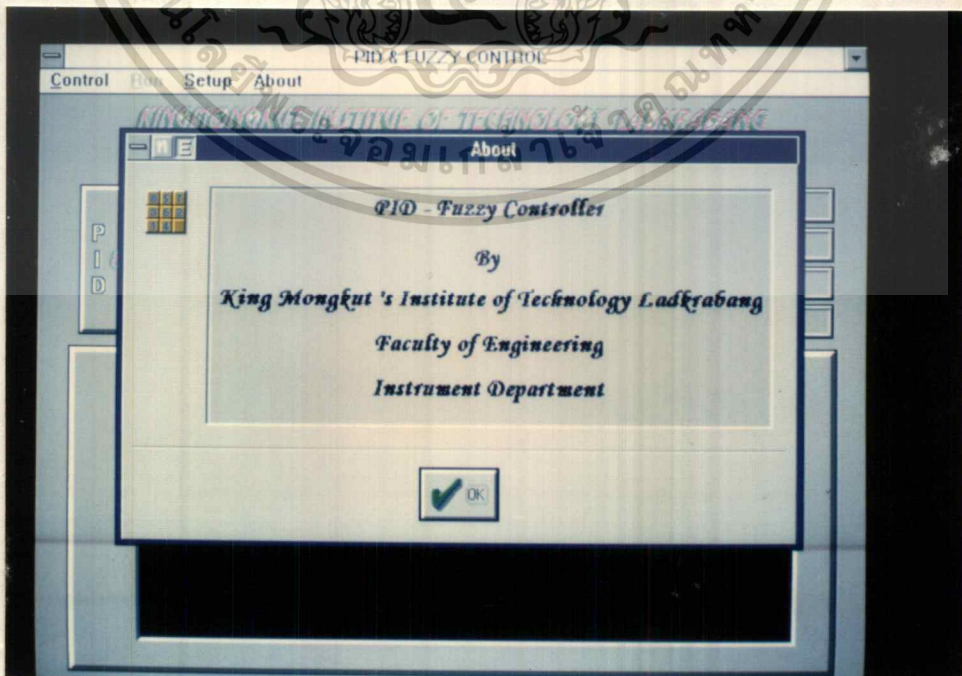
รูปที่ 5.13

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### รูปภาพแสดงหน้าจอแบบต่างๆ

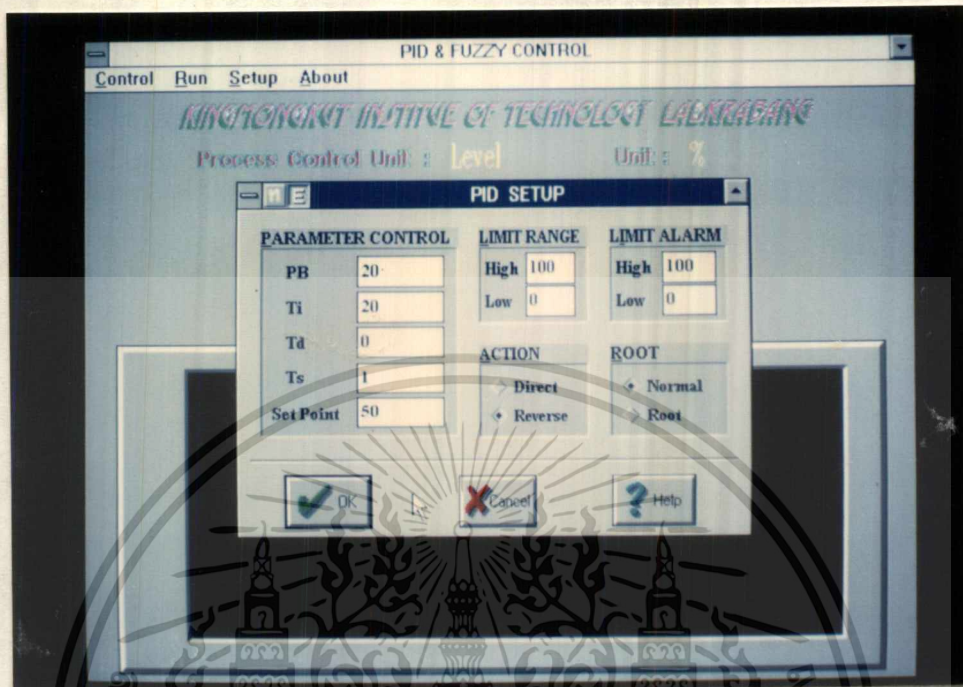


รูปที่ 5.14

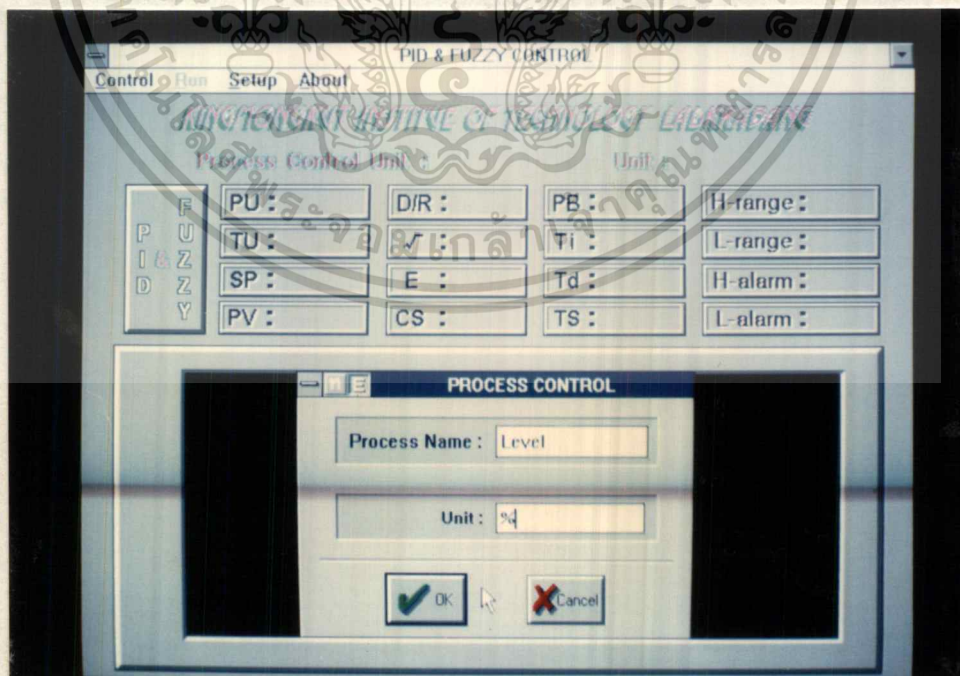


รูปที่ 5.15

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

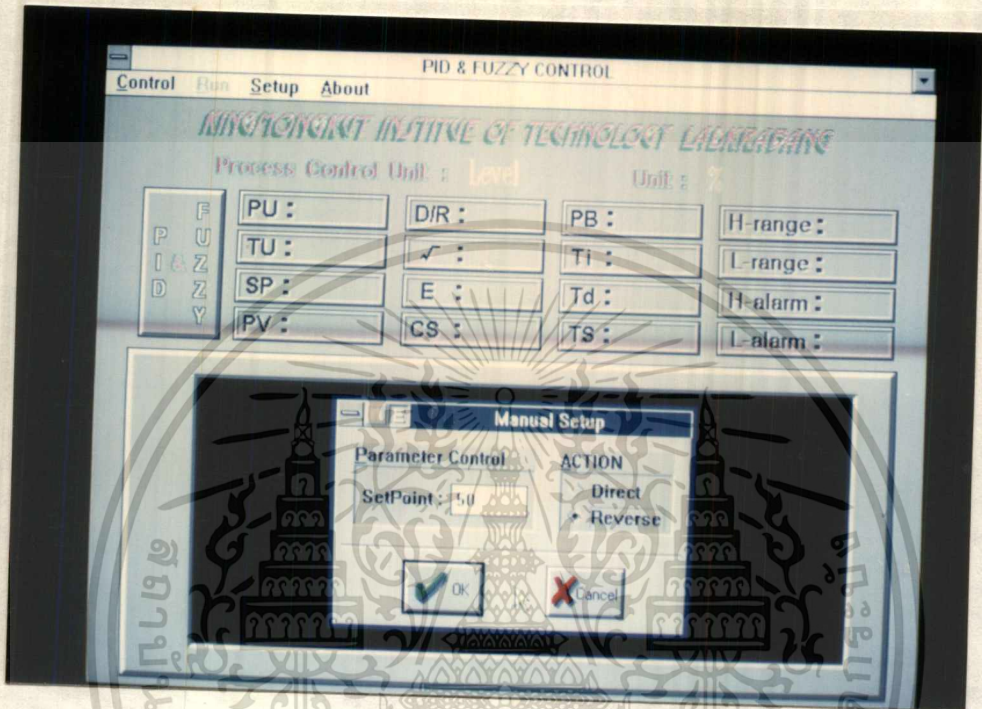


รูปที่ 5.16



รูปที่ 5.17

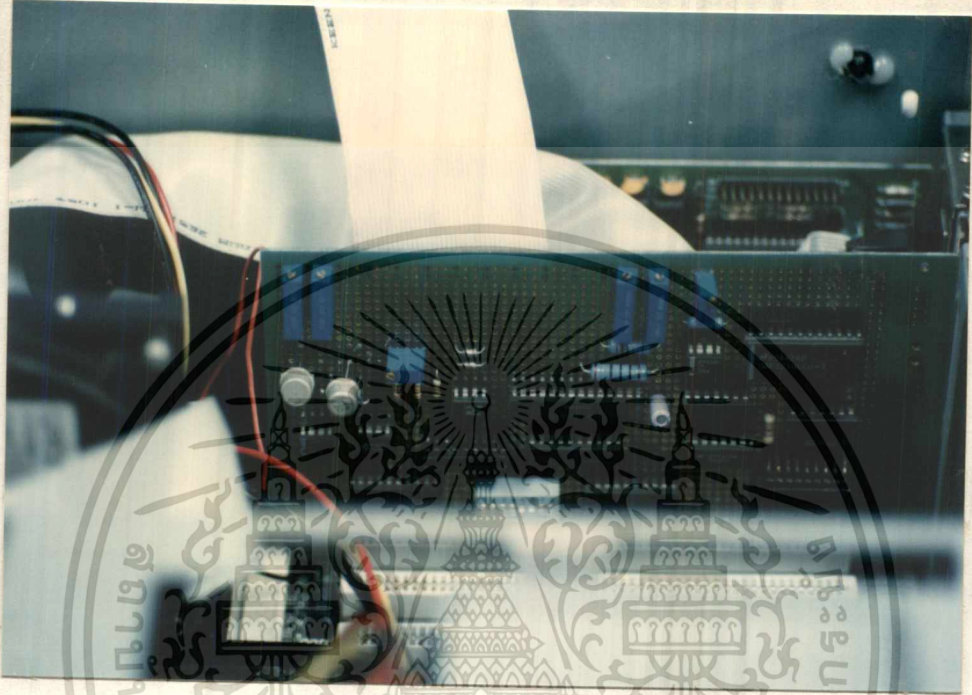
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



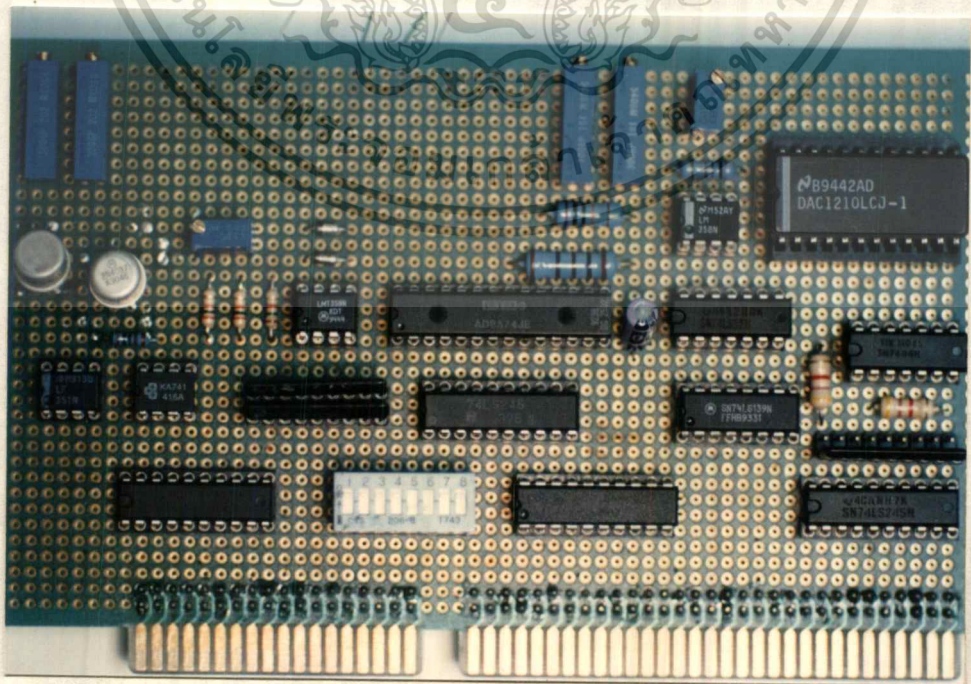
รูปที่ 5.18

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

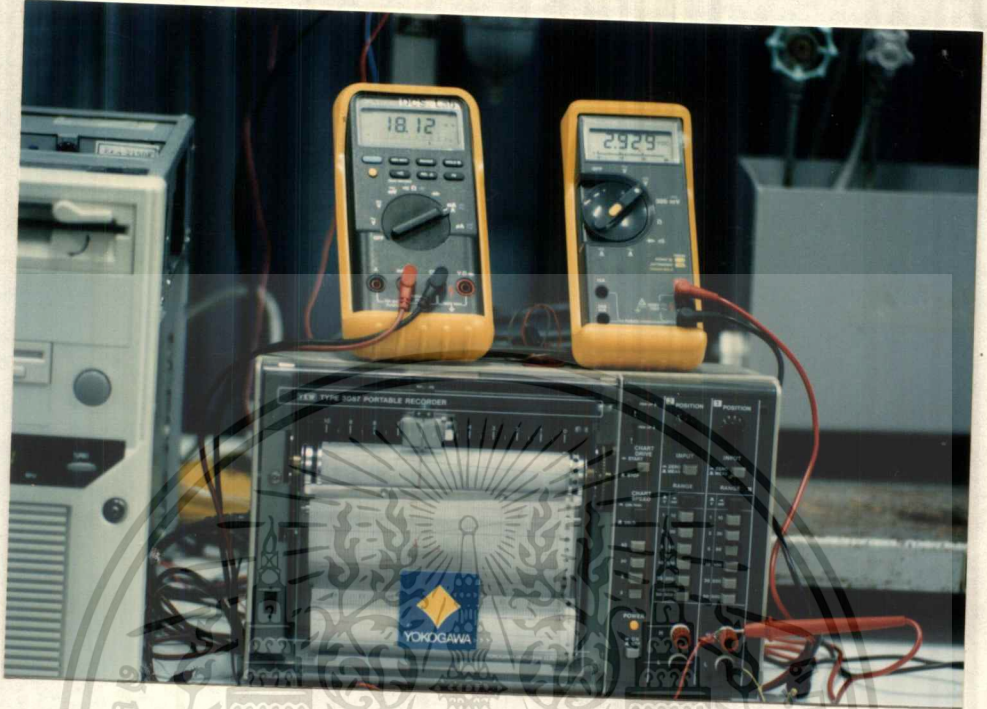
### รูปภาพแสดงฮาร์ดแวร์และกระบวนการ



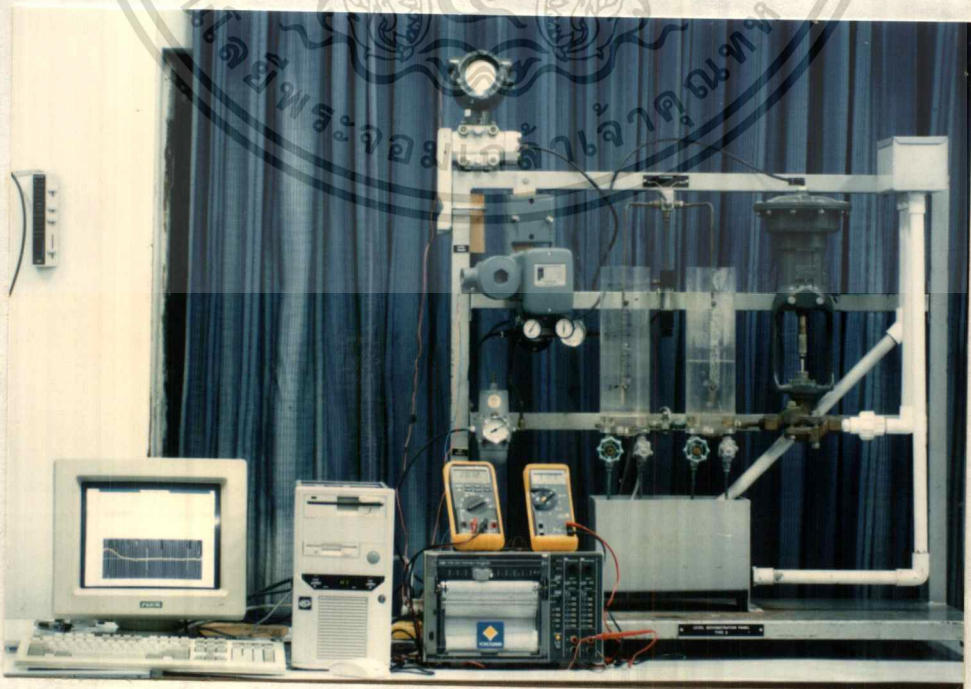
รูปที่ 5.19 การ์ด อินเทอร์เน็ต



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานรูปที่ 5.20 วิชาการเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5.21



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# บทที่ 6

## บทสรุป

### PID Controller

ตัวควบคุมกระบวนการในทางอุตสาหกรรมส่วนใหญ่ใช้การควบคุมแบบ PID Controller (Proportional - Integral and Derivative Controller) ซึ่งสัญญาณเอาต์พุตควบคุมได้มาจากการรวมของเทอม  $P + I + D$  ดังนี้

$$CS(k) = Kp \left( e(k) + \left( \frac{Ts}{Ti} \right) \sum_{i=1}^n e(i) + \left( \frac{Td}{Ts} \right) \Delta e(k) \right) + CS(0) \quad (6.1)$$

ไม่ว่ากระบวนการใด ๆ ก็ตามการควบคุมที่ดีเพื่อให้ค่าเป้าหมายตามที่ต้องการนั้น จะต้องมี การปรับแต่งตัวแปร ( Tuning Parameter ) ให้ได้ค่าเหมาะสมกับกระบวนการนั้น ๆ ซึ่งวิธีการปรับแต่งตัวแปรควบคุมสามารถแบ่งออกได้อย่างกว้าง ๆ ได้ 2 วิธี คือ การสุ่มทางปฏิบัติ และ การใช้สูตรทางทฤษฎี

การสุ่มทางปฏิบัติทำได้โดย

1. ป้อน  $PB(Kp)$  อย่างเดียวแล้วให้  $Ti$  มีค่าสูงสุดและ  $Td$  มีค่าเป็น 0 ต้องป้อน  $PB$  ให้เกิดค่า Offset น้อยที่สุด และในขณะเดียวกันก็ยังคงเป็นค่าที่ยังคงรักษาไม่ให้เกิดกระบวนการเกิดการเปลี่ยนแปลงแบบ Oscillate แล้วลองเปลี่ยนค่าเป้าหมายใหม่ สังเกตดูว่ากระบวนการเปลี่ยนแปลงตามค่าเป้าหมายหรือไม่

2. ต่อไป แก้อัปเดตค่า offset ที่เกิดขึ้น โดยการป้อน  $Ti$  ต้องค่อย ๆ ลดค่า  $Ti$  โดยให้เกิด Over Shoot มี Amplitude ต่ำที่สุด

3. สุดท้ายป้อนค่า  $Td$  ต้องค่อย ๆ เพิ่มจากศูนย์ขึ้นไปตามลำดับ ซึ่งเทอมของ Derivative นี้เป็นการคาดคะเนความเปลี่ยนแปลงของกระบวนการล่วงหน้าซึ่งอยู่ในรูปของความชัน การป้อน  $Td$  นั้น จะต้องระมัดระวัง เพราะจะมีผลต่อการควบคุมเป็นอย่างมาก โดยจะทำให้เกิดความไวในการควบคุมอย่างรวดเร็ว จนอาจทำให้การควบคุมเกิดการเปลี่ยนแปลงแบบ Oscillate หรือ ขาดเสถียรภาพในการควบคุมได้

การใช้สูตรทางทฤษฎี

สูตรที่ใช้กันอย่างแพร่หลายและนิยมคือ Ziegler Nichols

$$Kp = 0.6Ku \quad (6.2)$$

$$Ti = 0.5Tu \quad (6.3)$$

$$Td = 0.125Tu \quad (6.4)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จะเห็นว่าตัวแปรที่เราต้องทราบคือ  $Ku$  (Ultimate Gain) และ  $Tu$  (Ultimate Period Time) ซึ่งวิธีการหา นั้นได้จากการทดลองป้อน  $PB$  อย่างเดียว จนทำให้กระบวนการเกิดการเปลี่ยนแปลงแบบ Oscillate ซึ่งมีค่า Gain ที่เราเพื่อนนั้นก็คือ  $Ku$  และคาบเวลาของการ Oscillate ก็คือ  $Tu$  ซึ่งทั้ง  $Ku$  และ  $Tu$  นั้นขึ้นอยู่กับระบบของกระบวนการในแต่ละกระบวนการก็จะมีค่าแตกต่างกัน เมื่อได้  $Ku$  และ  $Tu$  แล้วก็นำไปแทนในสมการที่ 6.2, 6.3 และ 6.4 ก็จะได้ค่า  $Kp$ ,  $Ti$  และ  $Td$  แล้วนำไปป้อนในสมการที่ 6.1 ก็จะได้ตัวแปรที่เหมาะสมด้วยทฤษฎีของกระบวนการนั้นๆ

## PID - FUZZZY

การควบคุมโดยใช้ทฤษฎีของ Fuzzy เข้ามาช่วยนั้นสิ่งที่จะต้องทราบจากกระบวนการก็คือ  $Ku$  และ  $Tu$  เช่นเดียวกับสูตร Zeigler Nichols ส่วนทฤษฎี Fuzzy นั้น ได้กล่าวอธิบายแล้ว ดังนั้นต่อไปจะเปรียบเทียบวิธีการควบคุมแบบต่างๆ ดังที่กล่าวมา

การเปรียบเทียบ การควบคุมแบบ PID และ PID-FUZZZY Controller สิ่งที่ได้ชัดเจนที่สุดก็คือ

- การควบคุมแบบ PID ตัวแปรการควบคุม  $Kp, Ti, Td$  ผู้ควบคุมจะต้องป้อนให้ตัวควบคุมเอง และค่า  $Kp, Ti, Td$  นั้นจะคงที่ตลอดของการควบคุม แม้ว่าการเปลี่ยนแปลงของค่าความผิดพลาดของกระบวนการจะมากหรือน้อยอย่างไรก็ตาม ค่า  $Kp, Ti, Td$  ก็ยังคงที่ตลอด ดังนั้นการควบคุมด้วยวิธีนี้ ผู้ควบคุมต้องมีความชำนาญ และมีประสบการณ์ในการสังเกตพฤติกรรมของกระบวนการมากพอ ส่วนการควบคุมด้วยวิธีใช้สูตร Zeigler Nichols นั้น เมื่อหาค่า  $Ku$  และ  $Tu$  ได้แล้วก็ป้อน  $Kp, Ti, Td$  ที่หาได้เข้าสมการ PID ได้เลย แต่ทั้งนี้การควบคุมด้วยการใช้สูตรนี้ บางครั้งอาจจะไม่ดีเท่าที่ควรนัก เพราะยังเป็นค่าที่ไม่ละเอียดมากพอ ดังนั้น ผู้ควบคุมก็จำเป็นต้องปรับตัวแปรให้เหมาะสมต่อไปด้วยการสังเกตพฤติกรรมของกระบวนการเช่นกัน

- การควบคุมแบบ PID-FUZZZY ค่าตัวแปรการควบคุม  $Kp, Ti, Td$  ผู้ควบคุมไม่ต้องป้อนให้ เพียงแต่ป้อนค่า  $Ku$ ,  $Tu$  ให้เท่านั้น PID-FUZZZY ก็จะหาค่า  $Kp, Ti, Td$  ให้เลย โดยเอาค่าความผิดพลาดของกระบวนการในแต่ละรอบการคำนวณนำมาหาค่า  $Kp, Ti, Td$  ดังนั้น จะเห็นได้ว่าค่า  $Ku, Ti, Td$  จะเปลี่ยนแปลงตามค่าความผิดพลาดของกระบวนการในทุกๆ รอบการคำนวณ ซึ่งความละเอียดและความเหมาะสมของค่า นั้นก็ขึ้นอยู่กับที่ตั้งกฎและการใช้ทฤษฎีของ Fuzzy ว่า ได้สร้างรายละเอียดมากน้อยเพียงไร



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# โปรแกรม MONITOR .C

```
/* -----< Include files >----- */
#include <windows.h>
#include <stdlib.h>
#include <stdio.h>
#include <math.h>
#include "c:\borlandc\project\monitor.h"

/* -----< Global variables >----- */
HANDLE      hInst;          // Instance of application
HWND        hWndMain;      // Main window
WNDCLASS    WC;
static HMENU hMenu;
static HBITMAP hTitle,hPID,hPIDF,hManual,hSCOPE,hKMITL;
static HFONT  hLFont,hMFont,hSFont;
int         P=1;
int         MODE, C_Action=ID_Dir, C_Root=ID_Norm;
char        T_PB[8],T_Ti[8],T_Td[8],T_Ts[8],T_SP[8];
char        T_HR[8],T_LR[8],T_HA[8],T_LA[8],T_PU[8],T_TU[8];
char        T_Name[30],T_Unit[10];
float       Temp;
static int   I_Ts,Ts_C,RUNN,SET,New;
static int   DisplayPV[3600],DisplayCS[3600],DisplaySP[3600],DisplayHA,DisplayLA;
static float DataPV[3600],DataCS[3600];
static float PB,Ti,Td,Ts,SP,HR,LR,HA,LA,PU,TU,PV,CS,E;
static float KP,KD,KI,Ku,Kp_max,Kp_min,Kd_max,Kd_min,Kp,Kd,L,E1,E2;
static float e,d_e,a,b,ui,big,small;

/* -----< WinMain FUNCTION >----- */
#pragma argsused
int PASCAL WinMain(hInstance, hPrevInstance, lpCmdLine, nCmdShow)
HANDLE hInstance; // current instance
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้ภายในเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

HANDLE hPrevInstance;      // previous instance
LPSTR lpCmdLine;          // command line
int nCmdShow;              // show-window type (open/icon)
{
    MSG msg;                // message
    if (!hPrevInstance)     // Other instances of app running?
        if (!InitApplication(hInstance)) // Initialize shared things
            return (FALSE); // Exits if unable to initialize
    /* Perform initializations that apply to a specific instance */
    if (!InitInstance(hInstance, nCmdShow))
        return (FALSE);
    /* Acquire and dispatch messages until a WM_QUIT message is received. */
    while (GetMessage(&msg, // message structure
        NULL, // handle of window receiving the message
        NULL, // lowest message to examine
        NULL)) // highest message to examine
    {
        TranslateMessage(&msg); // Translates virtual key codes
        DispatchMessage(&msg); // Dispatches message to window
    }
    return (msg.wParam); // Returns the value from PostQuitMessage
}

```

/\* -----< InitApplication FUNCTION >----- \*/

```
int InitApplication (HANDLE hInstance)
```

```

{
    /* -----< Specify window class information >----- */
    WC.lpszClassName = "PID & FUZZY"; // Window class
    WC.hInstance = hInstance; // Instance
    WC.lpfnWndProc = (long (FAR PASCAL*)())MainWndProc; // Window function
    WC.style = CS_HREDRAWICS_VREDRAW; // Class style

```

```
    WC.lpszMenuName = "Menu_1"; // Load menu
```

เอกสารนี้เป็นทรัพย์สินทางปัญญาของสำนักงานเพื่อการศึกษาเท่านั้น ไม่นำไปเผยแพร่โดยไม่ได้รับอนุญาต

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

WC.hCursor = LoadCursor(hInstance,"Cursor_1"); // Load Mouse cursor
WC.hIcon = LoadIcon(hInstance,"Icon_1"); // Load icon
WC.hbrBackground = GetStockObject(LTGRAY_BRUSH); // Gray background
WC.cbClsExtra = 0; // No extra bytes
WC.cbWndExtra = 0; // No extra bytes
return (RegisterClass(&WC));
}

```

```

/* -----< InitInstance FUNCTION >----- */

```

```

BOOL InitInstance(hInstance, nCmdShow)

```

```

HANDLE hInstance; // Current instance identifier.
int nCmdShow; // Param for first ShowWindow() call.
{
hInst = hInstance;
/* Create a main window for this application instance. */
hWndMain = CreateWindow(
"PID & FUZZY", // See RegisterClass() call.
"PID & FUZZY CONTROL", // Text for window title bar.
WS_TILED | WS_SYSMENU | // Window style.
WS_CAPTION | WS_MINIMIZEBOX, // Window style.
0, // Default horizontal position.
0, // Default vertical position.
640, // Default width.
480, // Default height.
NULL, // Overlapped windows have no parent.
NULL, // Use the window class menu.
hInstance, // This instance owns this window.
NULL // Pointer not needed.
);
/* If window could not be created, return "failure" */
if (!hWndMain)
return (FALSE);

```

เอกสารนี้เป็นเอกสารสำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

ShowWindow(hWndMain, nCmdShow); // Show the window
UpdateWindow(hWndMain);        // Sends WM_PAINT message
return (TRUE);                  // Returns the value from PostQuitMessage
}

```

```

/* -----< KMITL DIALOG >----- */

```

```

#pragma argsused

```

```

BOOL FAR PASCAL About1 (HWND hDlg, unsigned dMsg,
                        WORD wParam, LONG lParam)

```

```

{
HDC      hDC,hMemDC;
HBITMAP  hOldBitmap;

hDC = GetDC(hDlg);
switch (dMsg)
{
case WM_INITDIALOG:
    return TRUE;
case WM_COMMAND:
    if (wParam == IDOK || wParam == IDCANCEL)
    {
        EndDialog(hDlg,TRUE);
        return TRUE;
    }
    break;
case WM_PAINT:
    hMemDC = CreateCompatibleDC(hDC);
    hOldBitmap = SelectObject(hMemDC,hKMITL);
    BitBlt(hDC,5,5,600,400,hMemDC,0,0,SRCCOPY);
    SelectObject(hMemDC,hOldBitmap);
    DeleteDC(hMemDC);

```

```

    break;

```

เอกสารนี้เป็นเอกสารที่สร้างขึ้นไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    }

    ReleaseDC(hDlg,hDC);

    return FALSE;
}

/* -----< About2 DIALOG >----- */
#pragma argsused
BOOL FAR PASCAL About2 (HWND hDlg, unsigned dMsg,
                        WORD wParam, LONG lParam)
{
    switch (dMsg)
    {
        case WM_INITDIALOG:
            return TRUE;
        case WM_COMMAND:
            if (wParam == IDOK || wParam == IDCANCEL)
            {
                EndDialog(hDlg,TRUE);
                return TRUE;
            }
            break;
    }

    return FALSE;
}

/* -----< Process Name DIALOG >----- */
#pragma argsused
BOOL FAR PASCAL Pro_Name (HWND hDlg, unsigned dMsg,
                          WORD wParam, LONG lParam)
{
    switch (dMsg)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

case WM_INITDIALOG:
    return TRUE;

case WM_COMMAND:
    switch (wParam)
    {
        case IDOK:
            GetDlgItemText(hDlg,ID_Name,T_Name,30);
            GetDlgItemText(hDlg,ID_Unit,T_Unit,10);
            EnableMenuItem(hMenu,S_Mode_PID,MF_ENABLED);
            EnableMenuItem(hMenu,S_Mode_PIDF,MF_ENABLED);
            EnableMenuItem(hMenu,S_Mode_Manual,MF_ENABLED);
            EnableMenuItem(hMenu,S_Parameter,MF_ENABLED);
            EnableMenuItem(hMenu,S_Save,MF_ENABLED);
            DrawMenuBar(hWndMain);
            New = 1;
            EndDialog(hDlg,TRUE);
            break;
        case IDCANCEL:
            EndDialog(hDlg,FALSE);
            break;
    }
}

return FALSE;
}

/* -----< PID Setup DIALOG >----- */
#pragma argsused
BOOL FAR PASCAL PID (HWND hDlg, unsigned dMsg,
                    WORD wParam, LONG lParam)
{
    switch (dMsg)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

case WM\_INITDIALOG:

```
CheckRadioButton(hDlg, ID_Dir, ID_Rev, C_Action);
```

```
CheckRadioButton(hDlg, ID_Norm, ID_Root, C_Root);
```

```
if (SET == 1)
```

```
{
```

```
SetDlgItemText(hDlg, ID_PB, T_PB);
```

```
SetDlgItemText(hDlg, ID_Ti, T_Ti);
```

```
SetDlgItemText(hDlg, ID_Td, T_Td);
```

```
SetDlgItemText(hDlg, ID_Ts, T_Ts);
```

```
SetDlgItemText(hDlg, ID_SP, T_SP);
```

```
SetDlgItemText(hDlg, ID_HR, T_HR);
```

```
SetDlgItemText(hDlg, ID_LR, T_LR);
```

```
SetDlgItemText(hDlg, ID_HA, T_HA);
```

```
SetDlgItemText(hDlg, ID_LA, T_LA);
```

```
}
```

```
else
```

```
{
```

```
SetDlgItemText(hDlg, ID_HR, "100");
```

```
SetDlgItemText(hDlg, ID_LR, "0");
```

```
SetDlgItemText(hDlg, ID_HA, "100");
```

```
SetDlgItemText(hDlg, ID_LA, "0");
```

```
}
```

```
return TRUE;
```

case WM\_COMMAND:

```
switch (wParam)
```

```
{
```

```
case ID_Dir:
```

```
case ID_Rev: C_Action = wParam; break;
```

```
case ID_Root:
```

```
case ID_Norm: C_Root = wParam; break;
```

```
case IDOK:
```

```
GetDlgItemText(hDlg, ID_PB, T_PB, 8);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

GetDlgItemText(hDlg,ID_Ti,T_Ti,8);
GetDlgItemText(hDlg,ID_Td,T_Td,8);
GetDlgItemText(hDlg,ID_Ts,T_Ts,8);
GetDlgItemText(hDlg,ID_SP,T_SP,8);
GetDlgItemText(hDlg,ID_HR,T_HR,8);
GetDlgItemText(hDlg,ID_LR,T_LR,8);
GetDlgItemText(hDlg,ID_HA,T_HA,8);
GetDlgItemText(hDlg,ID_LA,T_LA,8);
PB = atof(T_PB);
Ti = atof(T_Ti);
Td = atof(T_Td);
Ts = atof(T_Ts);
SP = atof(T_SP);
HR = atof(T_HR);
LR = atof(T_LR);
HA = atof(T_HA);
LA = atof(T_LA);
I_Ts = Ts;
E = E1 = E2 = 0;
SET = 1;
MODE = 0;
CheckMenuItem(hMenu,S_Mode_PID,MF_BYCOMMAND|MF_CHECKED);
CheckMenuItem(hMenu,S_Mode_PIDF,MF_BYCOMMAND|MF_UNCHECKED);
CheckMenuItem(hMenu,S_Mode_Manual,MF_BYCOMMAND|MF_UNCHECKED);
EnableMenuItem(hMenu,RUN,MF_ENABLED);
DrawMenuBar(hWndMain);
EndDialog(hDlg,TRUE);
break;
case IDCANCEL:
    EndDialog(hDlg,FALSE);
    break;
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    }

return FALSE;

}

/* -----< PID_FUZZY Setup DIALOG >----- */
#pragma argsused
BOOL FAR PASCAL PID_FUZZY (HWND hDlg, unsigned dMsg,
                           WORD wParam, LONG lParam)
{
switch (dMsg)
{
case WM_INITDIALOG:
    CheckRadioButton(hDlg, ID_Dir, ID_Rev, C_Action);
    CheckRadioButton(hDlg, ID_Norm, ID_Root, C_Root);
    if (SET == 1)
    {
        SetDlgItemText(hDlg, ID_PU, T_PU);
        SetDlgItemText(hDlg, ID_TU, T_TU);
        SetDlgItemText(hDlg, ID_Ts, T_Ts);
        SetDlgItemText(hDlg, ID_SP, T_SP);
        SetDlgItemText(hDlg, ID_HR, T_HR);
        SetDlgItemText(hDlg, ID_LR, T_LR);
        SetDlgItemText(hDlg, ID_HA, T_HA);
        SetDlgItemText(hDlg, ID_LA, T_LA);
    }
else
    {
        SetDlgItemText(hDlg, ID_HR, "100");
        SetDlgItemText(hDlg, ID_LR, "0");
        SetDlgItemText(hDlg, ID_HA, "100");
        SetDlgItemText(hDlg, ID_LA, "0");
    }
}
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

return TRUE;

case WM_COMMAND:
    switch (wParam)
    {
        case ID_Dir:

        case ID_Rev: C_Action = wParam; break;

        case ID_Root:

        case ID_Norm: C_Root = wParam; break;

        case IDOK:

            GetDlgItemText(hDlg,ID_PU,T_PU,8);
            GetDlgItemText(hDlg,ID_TU,T_TU,8);
            GetDlgItemText(hDlg,ID_Ts,T_Ts,8);
            GetDlgItemText(hDlg,ID_SP,T_SP,8);
            GetDlgItemText(hDlg,ID_HR,T_HR,8);
            GetDlgItemText(hDlg,ID_LR,T_LR,8);
            GetDlgItemText(hDlg,ID_HA,T_HA,8);
            GetDlgItemText(hDlg,ID_LA,T_LA,8);

            PU = atof(T_PU);
            TU = atof(T_TU);
            Ts = atof(T_Ts);
            SP = atof(T_SP);
            HR = atof(T_HR);
            LR = atof(T_LR);
            HA = atof(T_HA);
            LA = atof(T_LA);

            I_Ts = Ts;

            E= E1 = E2 = 0;

            SET = 1;

            MODE = 1;

            CheckMenuItem(hMenu,S_Mode_PID,MF_BYCOMMAND|MF_UNCHECKED);
            CheckMenuItem(hMenu,S_Mode_PIDF,MF_BYCOMMAND|MF_CHECKED);
            CheckMenuItem(hMenu,S_Mode_Manual,MF_BYCOMMAND|MF_UNCHECKED);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        EnableMenuItem(hMenu,RUN,MF_ENABLED);

        DrawMenuBar(hWndMain);

        EndDialog(hDlg,TRUE);

        break;

    case IDCANCEL:

        EndDialog(hDlg,FALSE);

        break;

    }

    break;

}

return FALSE;

}

/* -----< Manual Setup DIALOG >----- */
#pragma argsused
BOOL FAR PASCAL Manual (HWND hDlg, unsigned dMsg,
                        WORD wParam, LONG lParam)
{
    switch (dMsg)
    {
        case WM_INITDIALOG:
            CheckRadioButton(hDlg,ID_Dir,ID_Rev,C_Action);
            if (SET == 1)
                SetDlgItemText(hDlg,ID_SP,T_SP);
            return TRUE;

        case WM_COMMAND:
            switch (wParam)
            {
                case ID_Dir:
                case ID_Rev: C_Action = wParam; break;

                case IDOK:

                    GetDlgItemText(hDlg,ID_SP,T_SP,8);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        SP = atof(T_SP);

        SET = 1;

        MODE = 2;

        CheckMenuItem(hMenu,S_Mode_PID,MF_BYCOMMAND|MF_UNCHECKED);
        CheckMenuItem(hMenu,S_Mode_PIDF,MF_BYCOMMAND|MF_UNCHECKED);
        CheckMenuItem(hMenu,S_Mode_Manual,MF_BYCOMMAND|MF_CHECKED);
        EnableMenuItem(hMenu,RUN,MF_ENABLED);

        DrawMenuBar(hWndMain);

        EndDialog(hDlg,TRUE);

        break;

    case IDCANCEL:
        EndDialog(hDlg,FALSE);

        break;

    }

    break;

}

return FALSE;
}

/* -----< MainWinProc FUNCTION >----- */
LONG FAR PASCAL MainWndProc (HWND hWnd, unsigned msg,
                             WORD wParam, LONG lParam)
{
    HDC          hDC,hDCT,hMemDC;

    HBITMAP      hOldBitmap;

    HFONT        hOldFont;

    static HBRUSH hRedBrush,hOldBrush;

    static HPEN  hBlackPen,hGrayPen,hYellowPen,hGreenPen,hOldPen;

    PAINTSTRUCT  ps;

    static FARPROC lpfnProcName,lpfnProcPID,lpfnProcPIDF,lpfnProcManual;

    static FARPROC lpfnProcKMITL,lpfnProcINS,lpfnProcContent;

    RECT         RECT1;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

POINT          Polygpts[4];

static LOGFONT lf;

int            IDM;

unsigned       Px,Py;

static int     t=582,i=1,m=1,k=2,z=1,count=1;

static int     First = 1;

switch (msg) {
    case WM_CREATE:
        MODE = 1;  RUNN = 0;  New = 0;  Ts_C = 0;  SET = 0;

        hMenu = GetMenu(hWnd);

        SetTimer(hWnd,1,1000,NULL);

        hKMITL = LoadBitmap(hInst,"KMITL");
        hTitle = LoadBitmap(hInst,"TITLE");
        hPID = LoadBitmap(hInst,"PID");
        hPIDF = LoadBitmap(hInst,"PIDFUZZY");
        hManual = LoadBitmap(hInst,"MANUAL");
        hSCOPE = LoadBitmap(hInst,"SCOPE");
        hRedBrush = CreateSolidBrush(RGB(255,0,0));
        hYellowPen = CreatePen(PS_SOLID,1,RGB(255,255,0));
        hGrayPen = CreatePen(PS_SOLID,1,RGB(128,128,128));
        hGreenPen = CreatePen(PS_SOLID,1,RGB(0,255,0));
        hBlackPen = CreatePen(PS_SOLID,1,RGB(0,0,0));

        lf.lfWeight = FW_NORMAL;

        lf.lfCharSet = DEFAULT_CHARSET;

        lf.lfPitchAndFamily = FF_ROMAN;

        lf.lfHeight = 14;

        hSFont = CreateFontIndirect(&lf);

        lf.lfWeight = FW_NORMAL;

        lf.lfCharSet = OEM_CHARSET;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

lf.lfPitchAndFamily = FF_ROMAN;

lf.lfHeight = 27;

hMFont = CreateFontIndirect(&lf);

lf.lfWeight = FW_NORMAL;

lf.lfCharSet = OEM_CHARSET;

lf.lfPitchAndFamily = FF_ROMAN;

lf.lfHeight = 35;

hLFont = CreateFontIndirect(&lf);

break;

case WM_COMMAND:

switch (wParam) {

case F_New :

lpfnProcName = MakeProcInstance(Pro_Name,hInst);

DialogBox(hInst,"DLG_Name",hWnd,lpfnProcName);

FreeProcInstance(lpfnProcName);

InvalidateRect(hWnd,NULL,TRUE);

UpdateWindow(hWnd);

break;

case F_Exit:

IDM = MessageBox (hWnd," End Work", "End",

MB_OKCANCEL | MB_ICONQUESTION);

switch (IDM) {

case IDOK: PostQuitMessage(0);

break;

case IDCANCEL:

break; }

break;

case RUN:

IDM = MessageBox (hWnd,"Start Control PROCESS",

"CONTROL PROCESS",

MB_OKCANCEL | MB_ICONQUESTION);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

switch (IDM) {
    case IDOK: RUNN = 1; P=0;
                break;
    case IDCANCEL:
                break; }

break;

case S_Mode_PID:
    lpfnProcPID = MakeProcInstance(PID,hInst);
    DialogBox (hInst,"DLG_PID",hWnd,lpfnProcPID);
    FreeProcInstance(lpfnProcPID);
    InvalidateRect(hWnd,NULL,TRUE);
    UpdateWindow(hWnd);
    break;

case S_Mode_PIDF:
    lpfnProcPIDF = MakeProcInstance(PID_FUZZY,hInst);
    DialogBox (hInst,"DLG_PIDF",hWnd,lpfnProcPIDF);
    FreeProcInstance(lpfnProcPIDF);
    InvalidateRect(hWnd,NULL,TRUE);
    UpdateWindow(hWnd);
    break;

case S_Mode_Manual:
    lpfnProcManual = MakeProcInstance(Manual,hInst);
    DialogBox (hInst,"DLG_Manual",hWnd,lpfnProcManual);
    FreeProcInstance(lpfnProcPIDF);
    InvalidateRect(hWnd,NULL,TRUE);
    UpdateWindow(hWnd);
    break;

case S_Parameter:
    if (MODE == 0)
    {
        lpfnProcPID = MakeProcInstance(PID,hInst);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

DialogBox (hInst,"DLG_PID",hWnd,lpfnProcPID);

FreeProcInstance(lpfnProcPID);

}

else if (MODE == 1)

{

lpfnProcPIDF = MakeProcInstance(PID_FUZZY,hInst);

DialogBox (hInst,"DLG_PIDF",hWnd,lpfnProcPIDF);

FreeProcInstance(lpfnProcPIDF);

}

else if (MODE == 2)

{

lpfnProcManual = MakeProcInstance(Manual,hInst);

DialogBox (hInst,"DLG_Manual",hWnd,lpfnProcManual);

FreeProcInstance(lpfnProcPIDF);

}

if (RUNN == 1)

EnableMenuItem(hMenu,RUN,MF_ENABLED);

InvalidateRect(hWnd,NULL,TRUE);

UpdateWindow(hWnd);

break;

case H_Content:

lpfnProcContent = MakeProcInstance(Content,hInst);

DialogBox(hInst,"DLG_Content",hWnd,lpfnProcContent);

FreeProcInstance(lpfnProcContent);

break;

case H_About:

lpfnProcKMITL = MakeProcInstance(About1,hInst);

DialogBox(hInst,"DLG_KMITL",hWnd,lpfnProcKMITL);

FreeProcInstance(lpfnProcKMITL);

lpfnProcINS = MakeProcInstance(About2,hInst);

DialogBox(hInst,"DLG_INS",hWnd,lpfnProcINS);

FreeProcInstance(lpfnProcINS);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        break;
    }
    break;
case WM_PAINT:
    hDC = BeginPaint(hWnd,&ps);
    if (First == 1)
    {
        lpfnProcKMITL = MakeProcInstance(About1,hInst);
        DialogBox(hInst,"DLG_KMITL",hWnd,lpfnProcKMITL);
        FreeProcInstance(lpfnProcKMITL);
        hMemDC = CreateCompatibleDC(hDC);
        hOldBitmap = SelectObject(hMemDC,hTitle);
        BitBlt(hDC,50,0,540,64,hMemDC,0,0,SRCCOPY);
        if (New == 1)
        {
            SetTextColor(hDC,RGB(255,255,0));
            SetBkMode(hDC,TRANSPARENT);
            PrintText(hDC,280,40,"%s",T_Name);
            PrintText(hDC,470,40,"%s",T_Unit);
        }
        if (MODE == 0)
        {
            SelectObject(hMemDC,hPID);
            BitBlt(hDC,45,64,550,113,hMemDC,0,0,SRCCOPY);
            if (SET == 1 || RUNN == 1)
            {
                SetTextColor(hDC,RGB(0,0,255));
                SetBkMode(hDC,TRANSPARENT);
                PrintText(hDC,395,72,"%0.3f",PB);
                PrintText(hDC,395,100,"%0.3f",Ti);
                PrintText(hDC,395,128,"%0.3f",Td);
                PrintText(hDC,290,128,"%d",I_Ts);
            }
        }
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

PrintText(hDC,540,72,"%0.2f",HR);
PrintText(hDC,540,100,"%0.2f",LR);
if (C_Action == ID_Dir)
    PrintText(hDC,280,72,"%s","Direct");
else PrintText(hDC,271,72,"%s","Reverse");
if (C_Root == ID_Norm)
    PrintText(hDC,285,100,"%s","No");
else PrintText(hDC,280,100,"%s","Yes");
SetTextColor(hDC,RGB(0,255,0));
PrintText(hDC,151,71,"%0.3f",SP);
SetTextColor(hDC,RGB(255,0,0));
PrintText(hDC,540,127,"%0.2f",HA);
PrintText(hDC,540,155,"%0.2f",LA);
}
if (RUNN == 1)
{
    SetBkMode(hDC,TRANSPARENT);
    SetTextColor(hDC,RGB(0,0,255));
    PrintText(hDC,151,127,"%0.3f",E);
    SetTextColor(hDC,RGB(128,255,255));
    PrintText(hDC,151,155,"%0.3f",CS);
    SetTextColor(hDC,RGB(255,255,0));
    PrintText(hDC,151,100,"%0.3f",PV);
}
}
else if (MODE == 1)
{
    SelectObject(hMemDC,hPIDF);
    BitBlt(hDC,25,64,590,113,hMemDC,0,0,SRCCOPY);
    if (SET == 1 || RUNN == 1)
    {
        SetBkMode(hDC,TRANSPARENT);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

SetTextColor(hDC,RGB(0,0,255));
PrintText(hDC,158,72,"%0.3f",PU);
PrintText(hDC,158,101,"%0.3f",TU);
PrintText(hDC,426,156,"%d",I_Ts);
PrintText(hDC,561,72,"%0.2f",HR);
PrintText(hDC,561,100,"%0.2f",LR);
if (C_Action == ID_Dir)
    PrintText(hDC,290,72,"%s","Direct");
else PrintText(hDC,283,72,"%s","Reverse");
if (C_Root == ID_Norm)
    PrintText(hDC,300,100,"%s","No");
else PrintText(hDC,298,100,"%s","Yes");
if (RUNN == 1)
{
    SetTextColor(hDC,RGB(0,255,0));
    PrintText(hDC,158,128,"%0.3f",SP);
    SetTextColor(hDC,RGB(255,0,0));
    PrintText(hDC,561,128,"%0.2f",HA);
    PrintText(hDC,561,156,"%0.2f",LA);
    SetBkMode(hDC,TRANSPARENT);
    SetTextColor(hDC,RGB(0,0,255));
    PrintText(hDC,288,127,"%0.3f",E);
    PrintText(hDC,410,72,"%0.3f",PB);
    PrintText(hDC,410,100,"%0.3f",Ti);
    PrintText(hDC,410,128,"%0.3f",Td);
    SetTextColor(hDC,RGB(128,255,255));
    PrintText(hDC,288,155,"%0.3f",CS);
    SetTextColor(hDC,RGB(255,255,0));
    PrintText(hDC,158,156,"%0.3f",PV);
}
}

```

else if (MODE == 2)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

{
    SelectObject(hMemDC,hManual);
    BitBlt(hDC,120,64,400,113,hMemDC,0,0,SRCCOPY);
    if (SET == 1 || RUNN == 1)
    {
        SetTextColor(hDC,RGB(0,255,0));
        SetBkMode(hDC,TRANSPARENT);
        PrintText(hDC,184,140,"%0.3f",SP);
        SetTextColor(hDC,RGB(0,0,255));
        if (C_Action == ID_Dir)
            PrintText(hDC,459,140,"%s","Direct");
        else PrintText(hDC,451,140,"%s","Reverse");
    }
    if (RUNN == 1)
    {
        SetTextColor(hDC,RGB(255,255,0));
        SetBkMode(hDC,TRANSPARENT);
        PrintText(hDC,315,140,"%0.3f",PV);
    }
    SelectObject(hMemDC,hSCOPE);
    BitBlt(hDC,25,184,590,250,hMemDC,0,0,SRCCOPY);
    SelectObject(hMemDC,hOldBitmap);
    if (SET == 1)
    {
        hOldBrush = SelectObject(hDC,hRedBrush);
        hOldFont = SelectObject(hDC,hSFont);
        DisplayHA = HA;
        Temp = HA - DisplayHA;
        DisplayHA = DisplayHA * 2;
        if (Temp > 0.5) DisplayHA++;
        Polygpts[0].x = 586;
        Polygpts[0].y = 405-DisplayHA;
    }
}

```

```

Polygpts[1].x = 586+6;
Polygpts[1].y = 405-DisplayHA-6;
Polygpts[2].x = 586+6;
Polygpts[2].y = 405-DisplayHA+6;
Polygpts[3].x = 586;
Polygpts[3].y = 405-DisplayHA;
Polygon(hDC,Polygpts,4);
SetTextColor(hDC,RGB(255,0,0));
SetBkMode(hDC,TRANSPARENT);
PrintText(hDC,594,405-DisplayHA-7,"HA");
DisplayLA = LA;
Temp = LA - DisplayLA;
DisplayLA = DisplayLA * 2;
if (Temp > 0.5) DisplayLA++;
Polygpts[0].x = 586;
Polygpts[0].y = 405-DisplayLA;
Polygpts[1].x = 586+6;
Polygpts[1].y = 405-DisplayLA-6;
Polygpts[2].x = 586+6;
Polygpts[2].y = 405-DisplayLA+6;
Polygpts[3].x = 586;
Polygpts[3].y = 405-DisplayLA;
Polygon(hDC,Polygpts,4);
SetTextColor(hDC,RGB(255,0,0));
SetBkMode(hDC,TRANSPARENT);
PrintText(hDC,594,405-DisplayLA-7,"LA");
SetTextColor(hDC,RGB(0,0,255));
SetTextAlign(hDC,TA_RIGHT);
PrintText(hDC,77,199,"%0.2f",HA);
PrintText(hDC,77,399,"%0.2f",LA);
SelectObject(hDC,hOldFont);
SelectObject(hDC,hOldBrush);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    }

DeleteDC(hMemDC);

EndPaint(hWnd,&ps);

break;

case WM_TIMER:

SetRect(&RECT1,25,64,615,177);

hDCT = GetDC(hWnd);

PrintText(hDCT,40,50,"%d",P++);

if (RUNN == 1)

{

Process(count);

InvalidateRect(hWnd,&RECT1,0);

UpdateWindow(hWnd);

hOldPen = SelectObject(hDCT,hBlackPen);

MoveTo(hDCT,t,405-DisplayCS[m]);

z=0;

for(i=m; i<k-1; i++,z+=2)

LineTo(hDCT,t+z,405-DisplayCS[i]);

MoveTo(hDCT,t,405-DisplayPV[m]);

z=0;

for(i=m; i<k-1; i++,z+=2)

LineTo(hDCT,t+z,405-DisplayPV[i]);

MoveTo(hDCT,t,405-DisplaySP[m]);

z=0;

for(i=m; i<k-1; i++,z+=2)

LineTo(hDCT,t+z,405-DisplaySP[i]);

t-=2;

if(t<82){ t=82; m++; }

z=0;

SelectObject(hDCT,hGreenPen);

MoveTo(hDCT,t,405-DisplaySP[m]);

for(i=m; i<k; i++,z+=2)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

LineTo(hDCT,t+z,405-DisplaySP[i]);

z=0;

SelectObject(hDCT,hGrayPen);

MoveTo(hDCT,t,405-DisplayCS[m]);

for(i=m; i<k; i++,z+=2)

    LineTo(hDCT,t+z,405-DisplayCS[i]);

z=0;

SelectObject(hDCT,hYellowPen);

MoveTo(hDCT,t,405-DisplayPV[m]);

for(i=m; i<k; i++,z+=2)

    LineTo(hDCT,t+z,405-DisplayPV[i]);

k++; count++;
}

SelectObject(hDCT,hOldPen);
ReleaseDC(hWnd,hDCT);
break;
case WM_DESTROY:
DeleteObject(hKMITL);
DeleteObject(hTitle);
DeleteObject(hPID);
DeleteObject(hPIDF);
DeleteObject(hManual);
DeleteObject(hSCOPE);
DeleteObject(hSFont);
DeleteObject(hMFont);
DeleteObject(hLFont);
DeleteObject(hRedBrush);
DeleteObject(hYellowPen);
DeleteObject(hGrayPen);
DeleteObject(hGreenPen);
DeleteObject(hBlackPen);

PostQuitMessage(0);    // Send WM_QUIT if window

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

KillTimer(hWnd,1);

break;

// Pass other messages to the default window function
default: return (DefWindowProc(hWnd, msg, wParam, lParam));
}

return 0L;
}

/* -----< PrintText FUNCTION >----- */
void PrintText (HDC hDC, short x, short y, char *szFormat, ...)
{
char szBuffer[256];
char *pArguments ;
pArguments = (char *) &szFormat + sizeof szFormat;
vsprintf (szBuffer,szFormat,pArguments);
TextOut (hDC,x,y,szBuffer, strlen(szBuffer));
}

/* -----< Process FUNCTION >----- */
void Process(int Index)
{
unsigned char In_value,Out_value;
float inv,outv;

//-----READ DATA-----//

In_value = inportb(0x30A);
inv = (In_value-51.0)*(100.0/(255.0-51.0));
if(inv<0) inv = 0;
if(inv>100) inv = 100;
if (C_Root == ID_Norm)
PV = inv;
if (C_Root == ID_Root)
PV = sqrt(inv);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Ts_C++;
if (Ts_C == I_Ts)
{
    Ts_C = 0;
//-----PID MODE-----//
    if (MODE == 0)
    {
        E1 = E;
        E = SP - PV;
        E2 = E2 + E;
        CS = (100/PB) * ( E + ((Ts*E2)/(Ti*60)) + ( (Td*60)*(E-E1)/Ts ) );
    }
//-----PID & FUZZY MODE-----//
    if (MODE == 1)
    {
        E1 = E;
        E = SP - PV;
        E2 = E2 + E;
//-----Value of error-----
        e = E;
        if((e)>=(-100)&&(e)<(-6))
            a=1;
        else if((e)>=(-6)&&(e)<(-4))
            a=((-4)-(e))/2;
        else if((e)>=(-4)&&(e)<(-2))
            a=((-2)-(e))/2;
        else if((e)>=(-2)&&(e)<0)
            a=(0-(e))/2;
        else if(e>=0 && e<2)
            a=(2-e)/2;
        else if(e>=2 && e<4)
            a=(4-e)/2;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

else if(e>=4 && e<6)
    a=(6-e)/2;
else if(e>=6 && e<=100)
    a=1;
//-----Value of delta error-----
d_e = (E-E1);
if((d_e)>=(-100)&&(d_e)<(-6))
    b=1;
else if((d_e)>=(-6)&&(d_e)<(-4))
    b=((-4)-(d_e))/2;
else if((d_e)>=(-4)&&(d_e)<(-2))
    b((-2)-(d_e))/2;
else if((d_e)>=(-2)&&(d_e)<0)
    b=(0-(d_e))/2;
else if(d_e>=0 && d_e<2)
    b=(2-d_e)/2;
else if(d_e>=2 && d_e<4)
    b=(4-d_e)/2;
else if(d_e>=4 && d_e<6)
    b=(6-d_e)/2;
else if(d_e>=6 && d_e<=100)
    b=1;
ui=a*b;
big=1-exp((-4)*ui);
small=exp((-4)*ui);
Kp=kp();
Kd=kd();
L=lfa();
Ku = 100/PU;
Kp_min = 0.32*Ku;
Kp_max = 0.6*Ku;
Kd_min = 0.08*Ku*TU;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Kd_max = 0.15*Ku* $TU$ ;

KP = ((Kp_max-Kp_min)*Kp) + Kp_min;

KD = ((Kd_max-Kd_min)*Kd) + Kd_min;

KI = (KP*KP)/(L*KD);

Ti = KP/KI;

Td = KD/KP;

PB = 100/KP;

CS = (100/PB) * ( E + ((Ts*E2)/(Ti*60)) + ( (Td*60)*(E-E1)/Ts ) );

}

//-----MANUAL MODE-----//

if (MODE == 2)

{

CS = SP;

}

//-----OUT DATA-----//

if(CS>100.0) CS = 100.0;

if(CS<0.0) CS = 0.0;

outv = CS;

if (C_Action == ID_Dir)

Out_value = (outv * ((255.0-51.0)/100.0)) + 51.0;

if (C_Action == ID_Rev)

Out_value = (outv * ((51.0-255.0)/100.0)) + 255.0;

outportb(0x308,Out_value);

} // end of Time sampling Count

//-----OUT DATA TO DISPLAY & SAVE-----//

DataPV[Index] = PV;

DataCS[Index] = CS;

DisplayPV[Index] = DataPV[Index];

Temp = DataPV[Index] - DisplayPV[Index];

DisplayPV[Index] = DisplayPV[Index] * 2;

if (Temp > 0.5) DisplayPV[Index]++;

DisplayCS[Index] = DataCS[Index];

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Temp = DataCS[Index] - DisplayCS[Index];
DisplayCS[Index] = DisplayCS[Index] * 2;
if (Temp > 0.5) DisplayCS[Index]++;
DisplaySP[Index] = SP;
Temp = SP - DisplaySP[Index];
DisplaySP[Index] = DisplaySP[Index] * 2;
if (Temp > 0.5) DisplaySP[Index]++;
}
// ----- FUZZY RULE OF Kp^ -----
float kp( )
{
float p;
if(((e)>=(-100)&&(e)<(-4))&&((d_e)>=(-100)&&(d_e)<(-4)))
    p=u3();
if(((e)>=(-100)&&(e)<(-4))&&((d_e)>=(-4)&&(d_e)<(-2)))
    p=u1();
if(((e)>=(-100)&&(e)<(-4))&&((d_e)>=(-2)&&(d_e)<0))
    p=u1();
if(((e)>=(-100)&&(e)<(-4))&&((d_e)>=(0)&&(d_e)<2))
    p=u1();
if(((e)>=(-100)&&(e)<(-4))&&((d_e)>=2&&(d_e)<4))
    p=u1();
if(((e)>=(-100)&&(e)<(-4))&&((d_e)>=4&&(d_e)<=100))
    p=u2();
if(((e)>=(-4)&&(e)<(-2))&&((d_e)>=(-100)&&(d_e)<(-4)))
    p=u12();
if(((e)>=(-4)&&(e)<(-2))&&((d_e)>=(-4)&&(d_e)<(-2)))
    p=u3();
if(((e)>=(-4)&&(e)<(-2))&&((d_e)>=(-2)&&(d_e)<0))
    p=u1();
if(((e)>=(-4)&&(e)<(-2))&&((d_e)>=(0)&&(d_e)<2))
    p=u1();

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

if(((e)>=(-4)&&(e)<(-2))&&((d_e)>=2&&(d_e)<4))
    p=u2();
if(((e)>=(-4)&&(e)<(-2))&&((d_e)>=4&&(d_e)<=100))
    p=u8();
if(((e)>=(-2)&&(e)<(0))&&((d_e)>=(-100)&&(d_e)<(-4)))
    p=u16();
if(((e)>=(-2)&&(e)<(0))&&((d_e)>=(-4)&&(d_e)<(-2)))
    p=u12();
if(((e)>=(-2)&&(e)<(0))&&((d_e)>=(-2)&&(d_e)<(0)))
    p=u3();
if(((e)>=(-2)&&(e)<(0))&&((d_e)>=(0)&&(d_e)<2))
    p=u2();
if(((e)>=(-2)&&(e)<(0))&&((d_e)>=2&&(d_e)<4))
    p=u8();
if(((e)>=(-2)&&(e)<(0))&&((d_e)>=4&&(d_e)<=100))
    p=u16();
if(((e)>=(0)&&(e)<2)&&((d_e)>=(-100)&&(d_e)<(-4)))
    p=u16();
if(((e)>=(0)&&(e)<2)&&((d_e)>=(-4)&&(d_e)<(-2)))
    p=u15();
if(((e)>=(0)&&(e)<2)&&((d_e)>=(-2)&&(d_e)<(0)))
    p=u9();
if(((e)>=(0)&&(e)<2)&&((d_e)>=(0)&&(d_e)<2))
    p=u5();
if(((e)>=(0)&&(e)<2)&&((d_e)>=2&&(d_e)<4))
    p=u14();
if(((e)>=(0)&&(e)<2)&&((d_e)>=4&&(d_e)<=100))
    p=u16();
if(((e)>=2&&(e)<4)&&((d_e)>=(-100)&&(d_e)<(-4)))
    p=u15();
if(((e)>=2&&(e)<4)&&((d_e)>=(-4)&&(d_e)<(-2)))
    p=u9();

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

if(((e)>=2&&(e)<4)&&((d_e)>=(-2)&&(d_e)<0))
    p=u10);
if(((e)>=2&&(e)<4)&&((d_e)>=(0)&&(d_e)<2))
    p=u10);
if(((e)>=2&&(e)<4)&&((d_e)>=2&&(d_e)<4))
    p=u50);
if(((e)>=2&&(e)<4)&&((d_e)>=4&&(d_e)<=100))
    p=u140);
if(((e)>=4&&(e)<=100)&&((d_e)>=(-100)&&(d_e)<(-4)))
    p=u90);
if(((e)>=4&&(e)<=100)&&((d_e)>=(-4)&&(d_e)<(-2)))
    p=u10);
if(((e)>=4&&(e)<=100)&&((d_e)>=(-2)&&(d_e)<0))
    p=u10);
if(((e)>=4&&(e)<=100)&&((d_e)>=(0)&&(d_e)<2))
    p=u10);
if(((e)>=4&&(e)<=100)&&((d_e)>=2&&(d_e)<4))
    p=u10);
if(((e)>=4&&(e)<=100)&&((d_e)>=4&&(d_e)<=100))
    p=u50);
return(p);
}
// ----- FUZZY RULE OF kd^-----
float kd( )
{
    float d;
    if(((e)>=(-100)&&(e)<(-4))&&((d_e)>=(-100)&&(d_e)<(-4)))
        d=u130);
    if(((e)>=(-100)&&(e)<(-4))&&((d_e)>=(-4)&&(d_e)<(-2)))
        d=u140);
    if(((e)>=(-100)&&(e)<(-4))&&((d_e)>=(-2)&&(d_e)<0))
        d=u160);
}

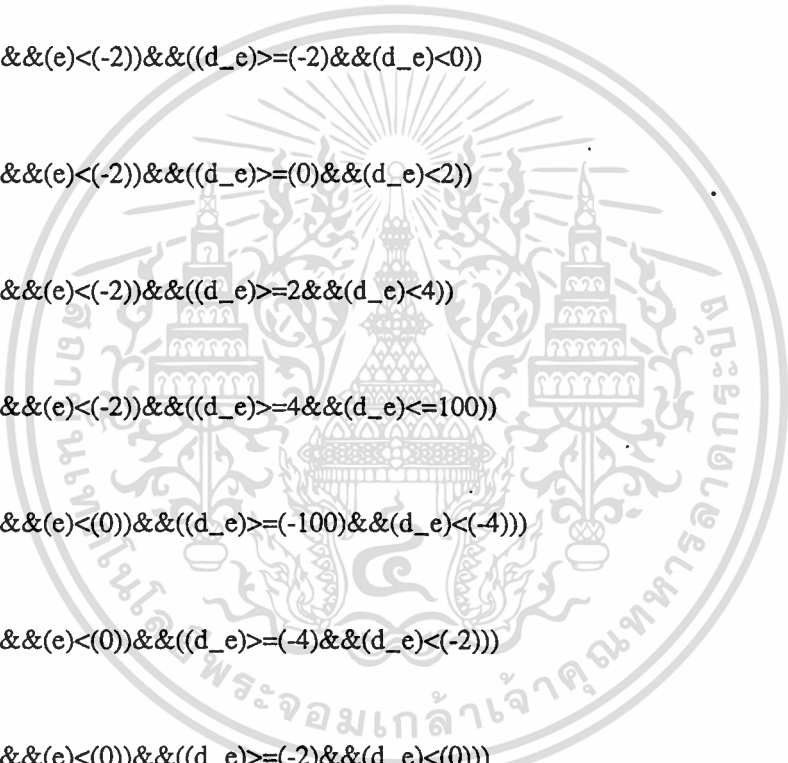
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

if(((e)>=(-100)&&(e)<(-4))&&((d_e)>=(0)&&(d_e)<2))
    d=u16();
if(((e)>=(-100)&&(e)<(-4))&&((d_e)>=2&&(d_e)<4))
    d=u15();
if(((e)>=(-100)&&(e)<(-4))&&((d_e)>=4&&(d_e)<=100))
    d=u13();
if(((e)>=(-4)&&(e)<(-2))&&((d_e)>=(-100)&&(d_e)<(-4)))
    d=u1();
if(((e)>=(-4)&&(e)<(-2))&&((d_e)>=(-4)&&(d_e)<(-2)))
    d=u5();
if(((e)>=(-4)&&(e)<(-2))&&((d_e)>=(-2)&&(d_e)<0))
    d=u14();
if(((e)>=(-4)&&(e)<(-2))&&((d_e)>=(0)&&(d_e)<2))
    d=u15();
if(((e)>=(-4)&&(e)<(-2))&&((d_e)>=2&&(d_e)<4))
    d=u9();
if(((e)>=(-4)&&(e)<(-2))&&((d_e)>=4&&(d_e)<=100))
    d=u1();
if(((e)>=(-2)&&(e)<(0))&&((d_e)>=(-100)&&(d_e)<(-4)))
    d=u1();
if(((e)>=(-2)&&(e)<(0))&&((d_e)>=(-4)&&(d_e)<(-2)))
    d=u1();
if(((e)>=(-2)&&(e)<(0))&&((d_e)>=(-2)&&(d_e)<(0)))
    d=u5();
if(((e)>=(-2)&&(e)<(0))&&((d_e)>=(0)&&(d_e)<2))
    d=u9();
if(((e)>=(-2)&&(e)<(0))&&((d_e)>=2&&(d_e)<4))
    d=u1();
if(((e)>=(-2)&&(e)<(0))&&((d_e)>=4&&(d_e)<=100))
    d=u1();
if(((e)>=(0)&&(e)<2)&&((d_e)>=(-100)&&(d_e)<(-4)))
    d=u1();

```



if(((e)>=0)&&(e)<2)&&((d\_e)>=-4)&&(d\_e)<(-2)))

d=u1();

if(((e)>=0)&&(e)<2)&&((d\_e)>=-2)&&(d\_e)<0))

d=u2();

if(((e)>=0)&&(e)<2)&&((d\_e)>=0)&&(d\_e)<2))

d=u3();

if(((e)>=0)&&(e)<2)&&((d\_e)>=2)&&(d\_e)<4))

d=u1();

if(((e)>=0)&&(e)<2)&&((d\_e)>=4)&&(d\_e)<=100))

d=u1();

if(((e)>=2)&&(e)<4)&&((d\_e)>=-100)&&(d\_e)<(-4)))

d=u1();

if(((e)>=2)&&(e)<4)&&((d\_e)>=-4)&&(d\_e)<(-2)))

d=u2();

if(((e)>=2)&&(e)<4)&&((d\_e)>=-2)&&(d\_e)<0))

d=u8();

if(((e)>=2)&&(e)<4)&&((d\_e)>=0)&&(d\_e)<2))

d=u12();

if(((e)>=2)&&(e)<4)&&((d\_e)>=2)&&(d\_e)<4))

d=u3();

if(((e)>=2)&&(e)<4)&&((d\_e)>=4)&&(d\_e)<=100))

d=u1();

if(((e)>=4)&&(e)<=100)&&((d\_e)>=-100)&&(d\_e)<(-4)))

d=u4();

if(((e)>=4)&&(e)<=100)&&((d\_e)>=-4)&&(d\_e)<(-2)))

d=u8();

if(((e)>=4)&&(e)<=100)&&((d\_e)>=-2)&&(d\_e)<0))

d=u16();

if(((e)>=4)&&(e)<=100)&&((d\_e)>=0)&&(d\_e)<2))

d=u16();

if(((e)>=4)&&(e)<=100)&&((d\_e)>=2)&&(d\_e)<4))

d=u12();

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

if(((e)>=4&&(e)<=100)&&((d_e)>=4&&(d_e)<=100))
    d=u4();
return(d);
}
// ----- FUZZY RULE OF ALFA -----
float lfa( )
{
float l;
if(((e)>=(-100)&&(e)<(-4))&&((d_e)>=(-100)&&(d_e)<(-4)))
    l=(a*b*2)+(a*(1-b)*2)+((1-a)*b*3)+((1-a)*(1-b)*3);
if(((e)>=(-100)&&(e)<(-4))&&((d_e)>=(-4)&&(d_e)<(-2)))
    l=(a*b*2)+(a*(1-b)*2)+((1-a)*b*3)+((1-a)*(1-b)*2);
if(((e)>=(-100)&&(e)<(-4))&&((d_e)>=(-2)&&(d_e)<0))
    l=(a*b*2)+(a*(1-b)*2)+((1-a)*b*2)+((1-a)*(1-b)*2);
if(((e)>=(-100)&&(e)<(-4))&&((d_e)>=0)&&(d_e)<2))
    l=(a*b*2)+(a*(1-b)*2)+((1-a)*b*2)+((1-a)*(1-b)*2);
if(((e)>=(-100)&&(e)<(-4))&&((d_e)>=2&&(d_e)<4))
    l=(a*b*2)+(a*(1-b)*2)+((1-a)*b*2)+((1-a)*(1-b)*3);
if(((e)>=(-100)&&(e)<(-4))&&((d_e)>=4&&(d_e)<=100))
    l=(a*b*2)+(a*(1-b)*2)+((1-a)*b*3)+((1-a)*(1-b)*3);
if(((e)>=(-4)&&(e)<(-2))&&((d_e)>=(-100)&&(d_e)<(-4)))
    l=(a*b*3)+(a*(1-b)*3)+((1-a)*b*4)+((1-a)*(1-b)*3);
if(((e)>=(-4)&&(e)<(-2))&&((d_e)>=(-4)&&(d_e)<(-2)))
    l=(a*b*3)+(a*(1-b)*2)+((1-a)*b*3)+((1-a)*(1-b)*3);
if(((e)>=(-4)&&(e)<(-2))&&((d_e)>=(-2)&&(d_e)<0))
    l=(a*b*2)+(a*(1-b)*2)+((1-a)*b*3)+((1-a)*(1-b)*2);
if(((e)>=(-4)&&(e)<(-2))&&((d_e)>=0)&&(d_e)<2))
    l=(a*b*2)+(a*(1-b)*2)+((1-a)*b*2)+((1-a)*(1-b)*3);
if(((e)>=(-4)&&(e)<(-2))&&((d_e)>=2&&(d_e)<4))
    l=(a*b*2)+(a*(1-b)*3)+((1-a)*b*3)+((1-a)*(1-b)*3);
if(((e)>=(-4)&&(e)<(-2))&&((d_e)>=4&&(d_e)<=100))
    l=(a*b*3)+(a*(1-b)*3)+((1-a)*b*3)+((1-a)*(1-b)*4);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

if(((e)>=(-2)&&(e)<(0))&&((d\_e)>=(-100)&&(d\_e)<(-4)))

l=(a\*b\*4)+(a\*(1-b)\*3)+((1-a)\*b\*5)+((1-a)\*(1-b)\*4);

if(((e)>=(-2)&&(e)<(0))&&((d\_e)>=(-4)&&(d\_e)<(-2)))

l=(a\*b\*3)+(a\*(1-b)\*3)+((1-a)\*b\*4)+((1-a)\*(1-b)\*3);

if(((e)>=(-2)&&(e)<(0))&&((d\_e)>=(-2)&&(d\_e)<(0)))

l=(a\*b\*3)+(a\*(1-b)\*2)+((1-a)\*b\*3)+((1-a)\*(1-b)\*3);

if(((e)>=(-2)&&(e)<(0))&&((d\_e)>=(0)&&(d\_e)<2))

l=(a\*b\*2)+(a\*(1-b)\*3)+((1-a)\*b\*3)+((1-a)\*(1-b)\*3);

if(((e)>=(-2)&&(e)<(0))&&((d\_e)>=2&&(d\_e)<4))

l=(a\*b\*3)+(a\*(1-b)\*3)+((1-a)\*b\*3)+((1-a)\*(1-b)\*4);

if(((e)>=(-2)&&(e)<(0))&&((d\_e)>=4&&(d\_e)<=100))

l=(a\*b\*3)+(a\*(1-b)\*4)+((1-a)\*b\*4)+((1-a)\*(1-b)\*5);

if(((e)>=(0)&&(e)<2)&&((d\_e)>=(-100)&&(d\_e)<(-4)))

l=(a\*b\*5)+(a\*(1-b)\*4)+((1-a)\*b\*4)+((1-a)\*(1-b)\*3);

if(((e)>=(0)&&(e)<2)&&((d\_e)>=(-4)&&(d\_e)<(-2)))

l=(a\*b\*4)+(a\*(1-b)\*3)+((1-a)\*b\*3)+((1-a)\*(1-b)\*3);

if(((e)>=(0)&&(e)<2)&&((d\_e)>=(-2)&&(d\_e)<0))

l=(a\*b\*3)+(a\*(1-b)\*3)+((1-a)\*b\*3)+((1-a)\*(1-b)\*2);

if(((e)>=(0)&&(e)<2)&&((d\_e)>=(0)&&(d\_e)<2))

l=(a\*b\*3)+(a\*(1-b)\*3)+((1-a)\*b\*2)+((1-a)\*(1-b)\*3);

if(((e)>=(0)&&(e)<2)&&((d\_e)>=2&&(d\_e)<4))

l=(a\*b\*3)+(a\*(1-b)\*4)+((1-a)\*b\*3)+((1-a)\*(1-b)\*3);

if(((e)>=(0)&&(e)<2)&&((d\_e)>=4&&(d\_e)<=100))

l=(a\*b\*4)+(a\*(1-b)\*5)+((1-a)\*b\*3)+((1-a)\*(1-b)\*4);

if(((e)>=2&&(e)<4)&&((d\_e)>=(-100)&&(d\_e)<(-4)))

l=(a\*b\*4)+(a\*(1-b)\*3)+((1-a)\*b\*3)+((1-a)\*(1-b)\*3);

if(((e)>=2&&(e)<4)&&((d\_e)>=(-4)&&(d\_e)<(-2)))

l=(a\*b\*3)+(a\*(1-b)\*3)+((1-a)\*b\*3)+((1-a)\*(1-b)\*2);

if(((e)>=2&&(e)<4)&&((d\_e)>=(-2)&&(d\_e)<0))

l=(a\*b\*3)+(a\*(1-b)\*2)+((1-a)\*b\*2)+((1-a)\*(1-b)\*2);

if(((e)>=2&&(e)<4)&&((d\_e)>=(0)&&(d\_e)<2))

l=(a\*b\*2)+(a\*(1-b)\*3)+((1-a)\*b\*2)+((1-a)\*(1-b)\*2);

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

if(((e)>=2&&(e)<4)&&((d_e)>=2&&(d_e)<4))
    l=(a*b*3)+(a*(1-b)*3)+((1-a)*b*2)+((1-a)*(1-b)*3);
if(((e)>=2&&(e)<4)&&((d_e)>=4&&(d_e)<=100))
    l=(a*b*3)+(a*(1-b)*4)+((1-a)*b*3)+((1-a)*(1-b)*3);
if(((e)>=4&&(e)<=100)&&((d_e)>=(-100)&&(d_e)<(-4)))
    l=(a*b*3)+(a*(1-b)*3)+((1-a)*b*2)+((1-a)*(1-b)*2);
if(((e)>=4&&(e)<=100)&&((d_e)>=(-4)&&(d_e)<(-2)))
    l=(a*b*3)+(a*(1-b)*2)+((1-a)*b*2)+((1-a)*(1-b)*2);
if(((e)>=4&&(e)<=100)&&((d_e)>=(-2)&&(d_e)<0))
    l=(a*b*2)+(a*(1-b)*2)+((1-a)*b*2)+((1-a)*(1-b)*2);
if(((e)>=4&&(e)<=100)&&((d_e)>=(0)&&(d_e)<2))
    l=(a*b*2)+(a*(1-b)*2)+((1-a)*b*2)+((1-a)*(1-b)*2);
if(((e)>=4&&(e)<=100)&&((d_e)>=2&&(d_e)<4))
    l=(a*b*2)+(a*(1-b)*3)+((1-a)*b*2)+((1-a)*(1-b)*2);
if(((e)>=4&&(e)<=100)&&((d_e)>=4&&(d_e)<=100))
    l=(a*b*3)+(a*(1-b)*3)+((1-a)*b*2)+((1-a)*(1-b)*2);
return(l);
}
// ----- VALUE OF FUZZY RULE Kp^ AND Kd^ -----
float u1( )
{
    float m1;
    m1=(a*b*big)+(a*(1-b)*big)+((1-a)*b*big)+((1-a)*(1-b)*big);
    return(m1);
}
float u2( )
{
    float m2;
    m2=(a*b*big)+(a*(1-b)*big)+((1-a)*b*big)+((1-a)*(1-b)*small);
    return(m2);
}
float u3( )

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

{
    float m3;
    m3=(a*b*big)+(a*(1-b)*big)+((1-a)*b*small)+((1-a)*(1-b)*big);
    return(m3);
}
float u4( )
{
    float m4;
    m4=(a*b*big)+(a*(1-b)*big)+((1-a)*b*small)+((1-a)*(1-b)*small);
    return(m4);
}
float u5( )
{
    float m5;
    m5=(a*b*big)+(a*(1-b)*small)+((1-a)*b*big)+((1-a)*(1-b)*big);
    return(m5);
}
float u8( )
{
    float m8;
    m8=(a*b*big)+(a*(1-b)*small)+((1-a)*b*small)+((1-a)*(1-b)*small);
    return(m8);
}
float u9( )
{
    float m9;
    m9=(a*b*small)+(a*(1-b)*big)+((1-a)*b*big)+((1-a)*(1-b)*big);
    return(m9);
}
float u12( )
{
    float m12;

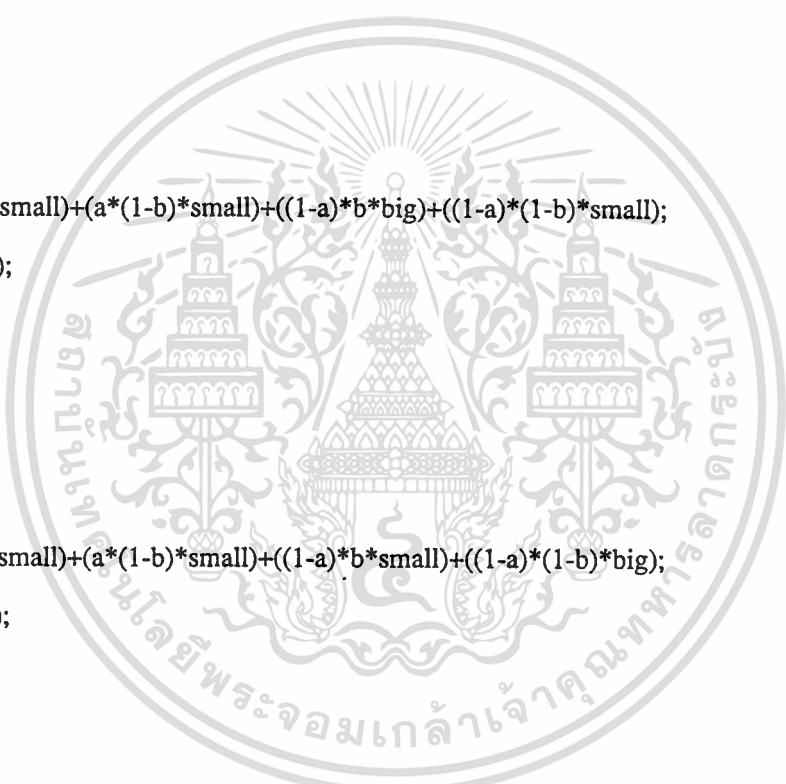
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    m12=(a*b*small)+(a*(1-b)*big)+((1-a)*b*small)+((1-a)*(1-b)*small);
    return(m12);
}
float u13()
{
    float m13;
    m13=(a*b*small)+(a*(1-b)*small)+((1-a)*b*big)+((1-a)*(1-b)*big);
    return(m13);
}
float u14()
{
    float m14;
    m14=(a*b*small)+(a*(1-b)*small)+((1-a)*b*big)+((1-a)*(1-b)*small);
    return(m14);
}
float u15()
{
    float m15;
    m15=(a*b*small)+(a*(1-b)*small)+((1-a)*b*small)+((1-a)*(1-b)*big);
    return(m15);
}
float u16()
{
    float m16;
    m16=(a*b*small)+(a*(1-b)*small)+((1-a)*b*small)+((1-a)*(1-b)*small);
    return(m16);
}

```



## โปรแกรม MONITOR.H

```
#define F_New          510
#define F_Exit        520
#define RUN           600
#define S_Mode_PID    711
#define S_Mode_PIDF   712
#define S_Mode_Manual 713
#define S_Parameter    710
#define H_Help         810
```

```
#define ID_PB          102
#define ID_Ti          103
#define ID_Td          104
#define ID_Ts          105
#define ID_SP          106
#define ID_HR          108
#define ID_LR          109
#define ID_HA          111
#define ID_LA          112
#define ID_Dir         114
#define ID_Rev         115
#define ID_Norm        117
#define ID_Root        118
#define ID_PU          119
#define ID_TU          120
#define ID_Name        121
#define ID_Unit        122
```

```
int PASCAL WinMain (HANDLE, HANDLE, LPSTR, int);
```

```
BOOL InitApplication (HANDLE);
```

```
BOOL InitInstance (HANDLE, int);
```

เอกสาร LONG FAR PASCAL MainWndProc (HWND,unsigned,WORD,LONG); นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

BOOL FAR PASCAL About1 (HWND,unsigned,WORD,LONG);  
BOOL FAR PASCAL About2 (HWND,unsigned,WORD,LONG);  
BOOL FAR PASCAL Pro\_Name (HWND,unsigned,WORD,LONG);  
BOOL FAR PASCAL PID (HWND,unsigned,WORD,LONG);  
BOOL FAR PASCAL PID\_FUZZY (HWND,unsigned,WORD,LONG);  
BOOL FAR PASCAL Manual (HWND,unsigned,WORD,LONG);

void PrintText(HDC hDC,short x, short y,char \*szFormat, ...);

void Process (int);

float u1(void);

float u2(void);

float u3(void);

float u4(void);

float u5(void);

float u8(void);

float u9(void);

float u12(void);

float u13(void);

float u14(void);

float u15(void);

float u16(void);

float kp(void);

float kd(void);

float lfa(void);



## MONITOR .DEF

NAME PID & FUZZY

DESCRIPTION 'This is software of PID & FUZZY CONTROLLER'

EXETYPE WINDOWS

STUB 'MONSTUB.EXE'

CODE PRELOAD MOVEABLE DISCARDABLE

DATA PRELOAD MOVEABLE MULTIPLE

HEAPSIZE 4096

STACKSIZE 8192

### EXPORTS

MainWndProc @1

About1 @2

About2 @3

Pro\_Name @4

PID @5

PID\_FUZZY @6

Manual @7



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## Microprocessor-Compatible ANALOG-TO-DIGITAL CONVERTER

### FEATURES

- COMPLETE 12-BIT A/D CONVERTER WITH REFERENCE, CLOCK, AND 3-, 12-, OR 16-BIT MICROPROCESSOR BUS INTERFACE
- IMPROVED PERFORMANCE SECOND SOURCE FOR 574A-TYPE A/D CONVERTERS
  - 25 $\mu$ s Maximum Conversion Time
  - 150ns Bus Access Time
  - A<sub>0</sub> Input Bus Contention During Read Operation Eliminated
- FULLY SPECIFIED FOR OPERATION ON  $\pm 12V$  OR  $\pm 15V$  SUPPLIES
- NO MISSING CODES OVER TEMPERATURE
  - 0°C to -75°C: ADC574AJH, XH Grades
  - 55°C to -125°C: ADC574ASH, TH Grades

contained  $\pm 10V$  reference, internal clock, digital interface for microprocessor control, and three-state outputs.

The reference circuit, containing a buried zener, is laser-trimmed for minimum temperature coefficient. The clock oscillator is current-controlled for excellent stability over temperature. Full-scale and offset errors may be externally-trimmed to zero. Internal scaling resistors are provided for the selection of analog input signal ranges of 0V to  $\pm 10V$ , 0V to  $\pm 20V$ ,  $\pm 5V$ , and  $\pm 10V$ .

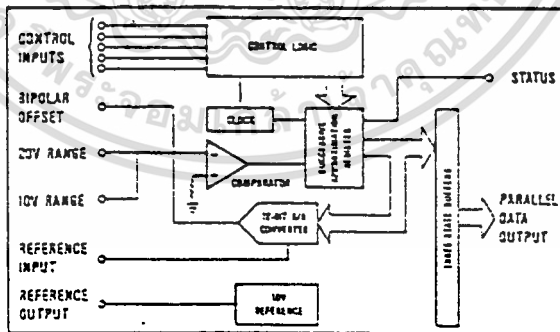
The converter may be externally programmed to provide 3- or 12-bit resolution. The conversion time for 12 bits is factory set for 20 $\mu$ s typical.

Output data are available in a parallel format from TTL-compatible three-state output buffers. Output data are coded in straight binary for unipolar input signals and bipolar offset binary for bipolar input signals.

The ADC574A, available in both industrial and military temperature ranges, requires supply voltages of  $\pm 5V$  and  $\pm 12V$  or  $\pm 15V$ . It is packaged in a hermetic 28-pin side-braced ceramic DIP.

### DESCRIPTION

The ADC574A is a 12-bit successive approximation analog-to-digital converter, utilizing state-of-the-art CMOS and laser-trimmed bipolar die custom-designed for freedom from latch-up and for optimum AC performance. It is complete with a self-



# SPECIFICATIONS

ELECTRICAL  
 $T_c = -25^\circ\text{C}$ ,  $V_{cc} = -12\text{V}$  or  $-15\text{V}$ ,  $V_{EE} = -12\text{V}$  or  $-15\text{V}$ ,  $V_{LDRIVE} = +5\text{V}$  unless otherwise specified.

MODEL	ADC374AJM, ADC374ABM			ADC374AKH, ADC374ATH			UNITS
	MIN	TYP	MAX	MIN	TYP	MAX	
RESOLUTION			12				Bits
INPUT							
ANALOG							V
Voltage Ranges: Unipolar			0 to +10, 0 to +20				V
Bipolar			$\pm 5, \pm 10$				V
Impedance: 0 to +10V, $\pm 5\text{V}$	3.7	5	6.3	.	.	.	$\times \Omega$
+10V 0V to -20V	7.5	10	12.5	.	.	.	$\times \Omega$
DIGITAL (CE, CS, R/C, A <sub>n</sub> 12/8)							V
Over Temperature Range	+2.4 <sup>(1)</sup>		+5.5	.	.	.	V
Voltages: Logic 1	-0.5		-0.8	.	.	.	V
Logic 0	-5	0.1	-5	.	.	.	$\mu\text{A}$
Current: 2.0V $\leq V_{in} \leq 5.0\text{V}$		5					pF
Capacitance							
TRANSFER CHARACTERISTICS							
ACCURACY							
At $-25^\circ\text{C}$						$\pm 1/2$	LSB
Linearity Error			$\pm 1$			.	LSB
Unipolar Offset Error (adjustable to zero)			$\pm 2$			$\pm 4$	LSB
Bipolar Offset Error (adjustable to zero)			$\pm 10$			.	LSB
Full-Scale Calibration Error <sup>(2)</sup> (adjustable to zero)			$\pm 0.3$			.	% of FS <sup>(3)</sup>
No Missing Codes Resolution	11			12		.	Bits
Inherent Quantization Error		$\pm 1/2$				.	LSB
From $T_{min}$ to $T_{max}$							
Linearity Error: J, K Grades			$\pm 1$			$\pm 1/2$	% of FS
S, T Grades			$\pm 1$			.	% of FS
Full-Scale Calibration Error			$\pm 0.5$			$\pm 0.4$	% of FS
Without initial adjustment <sup>(2)</sup> : J, K Grades			$\pm 0.8$			$\pm 0.6$	% of FS
S, T Grades			$\pm 0.22$			$\pm 0.12$	% of FS
Adjusted to zero at $-25^\circ\text{C}$ : J, K Grades			$\pm 0.5$			$\pm 0.25$	% of FS
S, T Grades							Bits
No Missing Codes Resolution	11			12			Bits
POWER SUPPLY SENSITIVITY							
Change in Full-Scale Calibration						$\pm 1$	LSB
$-13.5\text{V} < V_{cc} < -16.5\text{V}$ or $-11.4\text{V} < V_{EE} < -12.6\text{V}$			$\pm 2$			.	LSB
$-18.5\text{V} < V_{cc} < -13.5\text{V}$ or $-12.6\text{V} < V_{EE} < -11.4\text{V}$			$\pm 2$			$\pm 1$	LSB
$-4.5\text{V} < V_{LDRIVE} < -5.5\text{V}$			$\pm 1/2$			.	LSB
CONVERSION TIME <sup>(4)</sup>							$\mu\text{s}$
3-Bit Cycle	10	13	17	.	.	.	$\mu\text{s}$
12-Bit Cycle	15	20	25	.	.	.	$\mu\text{s}$
DRIFT							ppm/ $^\circ\text{C}$
Unipolar Offset: J, K Grades			$\pm 10$			$\pm 5$	ppm/ $^\circ\text{C}$
S, T Grades			$\pm 5$			$\pm 2.5$	LSB
Change over Temperature Range, All Grades			$\pm 2$			$\pm 1$	ppm/ $^\circ\text{C}$
Bipolar Offset, All Grades			$\pm 10$			$\pm 5$	LSB
Change over Temperature Range: J, K Grades			$\pm 2$			$\pm 1$	ppm/ $^\circ\text{C}$
S, T Grades			$\pm 4$			$\pm 2$	LSB
Full-Scale Calibration: J, K Grades			$\pm 45$			$\pm 25$	ppm/ $^\circ\text{C}$
S, T Grades			$\pm 50$			$\pm 25$	ppm/ $^\circ\text{C}$
Change over Temperature Range: J, K Grades			$\pm 9$			$\pm 5$	LSB
S, T Grades			$\pm 20$			$\pm 10$	LSB
OUTPUT							
DIGITAL DR <sub>in</sub> - OB <sub>in</sub> STATUS)							
Over Temperature Range							
Output Codes: Unipolar				Unipolar Straight Binary (USB)			
Bipolar				Bipolar Offset Binary (BOB)			
Logic Levels: Logic 0 ( $I_{OL} = 1.5\text{mA}$ )	-2.4		-0.4	.	.	.	V
Logic 1 ( $I_{OH} = 500\mu\text{A}$ )	-5	0.1	-5	.	.	.	V
Leakage, Data Bus Only, High-Z State		5		.	.	.	$\mu\text{A}$
Capacitance							pF
INTERNAL REFERENCE VOLTAGE							V
Voltage	+9.9	+10.0	+10.1	.	.	.	mV
Source Current Available for External Loads <sup>(5)</sup>	2.0						mA

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

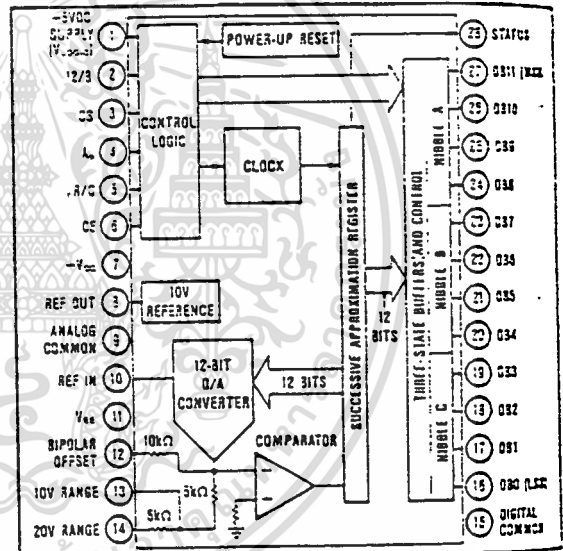
## ELECTRICAL (CONT)

$T_a = -25^\circ\text{C}$ ,  $V_{cc} = -12\text{V}$  or  $-15\text{V}$ ,  $V_{ee} = -12\text{V}$  or  $-15\text{V}$ ,  $V_{load} = -5\text{V}$  unless otherwise specified.

MODEL	ADC374AJM, ADC374AM			ADC374AKM, ADC374ATH			UNITS
	MIN	TYP	MAX	MIN	TYP	MAX	
<b>POWER SUPPLY REQUIREMENTS</b>							
Voltage: $V_{cc}$	+11.4		-16.5	.	.	.	V
$V_{ee}$	-11.4		-16.5	.	.	.	V
$V_{load}$	-4.5		-5.5	.	.	.	V
Current: $I_{cc}$		11	15	.	.	.	mA
$I_{ee}$		21	28	.	.	.	mA
$I_{load}$		7	15	.	.	.	mA
Power Dissipation ( $\pm 15\text{V}$ Supplies)		515	720	.	.	.	mW
<b>TEMPERATURE RANGE (Ambient)</b>							
Specification: J, K Grades	0		-75	.	.	.	$^\circ\text{C}$
S, T Grades	-55		-125	.	.	.	$^\circ\text{C}$
Storage	-55		-150	.	.	.	$^\circ\text{C}$

\*Same specification as grade to the immediate left.  
 NOTES: (1) Although this guaranteed threshold is higher than the standard TTL guaranteed level ( $-2.0\text{V}$ ), bus loading is much less. Typical input current is 0.25% of a standard TTL load. (2) With fixed  $50\Omega$  resistor from REF OUT to REF IN. This parameter is also adjustable to zero at  $-25^\circ\text{C}$  (see "Optional External Scale and Offset Adjustments" section). (3) FS in this specification table means Full-Scale Range. That is, for a  $\pm 10\text{V}$  input range, FS means 20V, for a  $\pm 2\text{V}$  range, FS means 4V. Use of the term Full Scale for these specifications instead of Full-Scale Range is consistent with other vendors' 574 and 574A specification tables. (4) See "Controlling the ADC374A" section for detailed information concerning digital timing. (5) External loading must be constant during conversion. When supplying an external load and operating on  $\pm 12\text{V}$  supplies, a buffer amplifier must be provided for the reference output.

## CONNECTION DIAGRAM



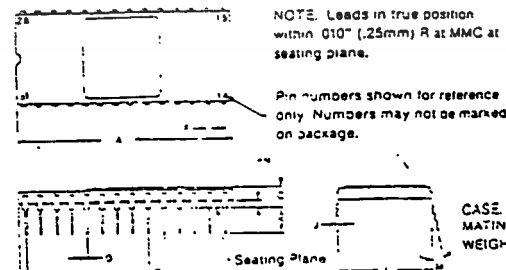
## ABSOLUTE MAXIMUM RATINGS

$V_{cc}$ to Digital Common	0 to $-16.5\text{V}$
$V_{ee}$ to Digital Common	0 to $-16.5\text{V}$
$V_{load}$ to Digital Common	0 to $+7\text{V}$
Analog Common to Digital Common	$\pm 1\text{V}$
Control Inputs (CE, CS, A <sub>n</sub> , 12/3, R/C) to Digital Common	$-0.5\text{V}$ to $V_{load} - 0.5\text{V}$
Analog Inputs (REF IN, BIP OFF, 10V <sub>W</sub> ) to Analog Common	$\pm 16.5\text{V}$
20V <sub>W</sub> to Analog Common	$\pm 24\text{V}$
REF OUT	Indefinite Short to Common, Momentary Short to $V_{cc}$
Chip Temperature: J, K, L Grades	$+100^\circ\text{C}$
S, T, U Grades	$+150^\circ\text{C}$
Power Dissipation	1000mW
Lead Temperature, Soldering	$-300^\circ\text{C}$ , 10sec
Thermal Resistance, $\theta_{JA}$	$48^\circ\text{C/W}$

**CAUTION:** These devices are sensitive to electrostatic discharge. Appropriate I.C. handling procedures should be followed.

## MECHANICAL

DIM	INCHES		MILLIMETERS	
	MIN	MAX	MIN	MAX
A	1.388	1.414	35.20	36.92
C	.108	.168	2.74	4.22
D	.016	.021	0.38	0.53
F	.035	.060	0.89	1.52
G	.100 BASIC		2.54 BASIC	
H	.036	.064	0.91	1.63
J	.008	.012	0.20	0.30
K	.120	.240	3.05	6.10
L	.800 BASIC		16.24 BASIC	
M	-	10°	-	10°
N	.025	.060	0.64	1.52



CASE: Ceramic, hermetic  
 MATING CONNECTOR: 28C3J2  
 WEIGHT: 4.8 grams (IC 172)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# DISCUSSION OF SPECIFICATIONS

## LINEARITY ERROR

Linearity error is defined as the deviation of actual code transition values from the ideal transition values. Ideal transition values lie on a line drawn through zero (or minus full scale for bipolar operation) and plus full scale. The zero value is located at an analog input value  $1/2\text{LSB}$  before the first code transition ( $000_H$  to  $001_H$ ). The full-scale value is located at an analog value  $3/2\text{LSB}$  beyond the last code transition ( $FFF_H$  to  $FFF_H$ ) (see Figure 1).

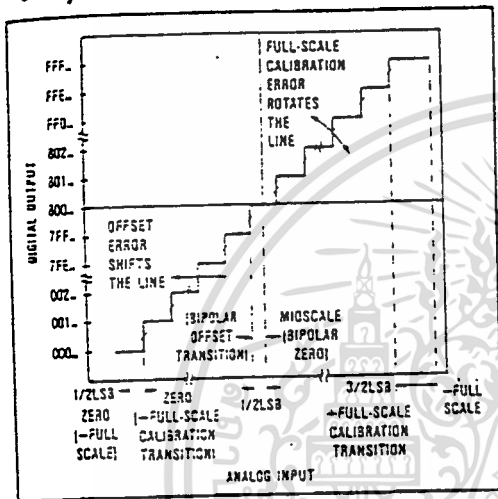


FIGURE 1. ADC574A Transfer Characteristic Terminology.

Thus, for a converter connected for bipolar operation and with a full-scale range (or span) of  $20\text{V}$  ( $\pm 10\text{V}$ ), the zero value of  $-10\text{V}$  is  $2.44\text{mV}$  below the first code transition ( $000_H$  to  $001_H$  at  $-9.99756\text{V}$ ) and the plus full-scale value of  $+10\text{V}$  is  $7.32\text{mV}$  above the last code transition ( $FFF_H$  to  $FFF_H$  at  $+9.99268\text{V}$ ) (see Table I).

## NO MISSING CODES (DIFFERENTIAL LINEARITY ERROR)

A specification which guarantees no missing codes requires that every code combination appear in a monotonically-increasing sequence as the analog input is

increased throughout the range. Thus, every input code width (quantum) must have a finite width. If an input quantum has a value of zero (a differential linearity error of  $-1\text{LSB}$ ), a missing code will occur.

ADC574A KH and TH grades are guaranteed to have no missing codes to 12-bit resolution over their respective specification temperature ranges.

## UNIPOLAR OFFSET ERROR

An ADC574A connected for unipolar operation has an analog input range of  $0\text{V}$  to plus full scale. The first output code transition should occur at an analog input value  $1/2\text{LSB}$  above  $0\text{V}$ . Unipolar offset error is defined as the deviation of the actual transition value from the ideal value. The unipolar offset temperature coefficient specifies the change of this transition value versus a change in ambient temperature.

## BIPOLAR OFFSET ERROR

A/D converter specifications have historically defined bipolar offset as the first transition value above the minus full-scale value. The ADC574A specification, however, follows the terminology defined for the 574 converter several years ago. Thus, bipolar offset is located near the mid-scale value of  $0\text{V}$  (bipolar zero) at the output code transition  $7FF_H$  to  $800_H$ .

Bipolar offset error for the ADC574A is defined as the deviation of the actual transition value from the ideal transition value located  $1/2\text{LSB}$  below  $0\text{V}$ . The bipolar offset temperature coefficient specifies the maximum change of the code transition value versus a change in ambient temperature.

## FULL SCALE CALIBRATION ERROR

The last output code transition ( $FFF_H$  to  $FFF_H$ ) occurs for an analog input value  $3/2\text{LSB}$  below the nominal full-scale value. The full scale calibration error is the deviation of the actual analog value at the last transition point from the ideal value. The full-scale calibration temperature coefficient specifies the maximum change of the code transition value versus a change in ambient temperature.

## POWER SUPPLY SENSITIVITY

Electrical specifications for the ADC574A assume the application of the rated power supply voltages of  $\pm 5\text{V}$  and  $\pm 12\text{V}$  or  $\pm 15\text{V}$ . The major effect of power supply

TABLE I. Input Voltages, Transition Values, and LSB Values.

Binary (BIN) Output	Input Voltage Range and LSB Values				
	Defined As:	$\pm 10\text{V}$	$\pm 5\text{V}$	$0$ to $-10\text{V}$	$0$ to $-20\text{V}$
Analog Input Voltage Range		$20\text{V}$	$10\text{V}$	$10\text{V}$	$20\text{V}$
One Least Significant Bit (LSB)	$\frac{\text{FSR}}{2^n}$ $n = 8$ $n = 12$	$7.32\text{mV}$ $4.88\text{mV}$	$3.906\text{mV}$ $2.44\text{mV}$	$3.906\text{mV}$ $2.44\text{mV}$	$7.32\text{mV}$ $4.88\text{mV}$
Output Transition Values	$\rightarrow$ Full-Scale Calibration Mid-Scale Calibration (Bipolar Offset) Zero Calibration ( $\rightarrow$ Full-Scale Calibration)	$-10\text{V} - 3/2\text{LSB}$ $0 - 1/2\text{LSB}$ $-10\text{V} - 1/2\text{LSB}$	$-5\text{V} - 3/2\text{LSB}$ $0 - 1/2\text{LSB}$ $-5\text{V} - 1/2\text{LSB}$	$-10\text{V} - 1/2\text{LSB}$ $-5\text{V} - 1/2\text{LSB}$ $0 - 1/2\text{LSB}$	$-20\text{V} - 3/2\text{LSB}$ $-10\text{V} - 1/2\text{LSB}$ $0 - 1/2\text{LSB}$

voltage deviations from the rated values will be a small change in the full-scale calibration value. This change, of course, results in a proportional change in all code transition values (i.e. a gain error). The specification describes the maximum change in the full-scale calibration value from the initial value for a change in each power supply voltage.

### TEMPERATURE COEFFICIENTS

The temperature coefficients for full-scale calibration, unipolar offset and bipolar offset specify the maximum change from the  $-25^{\circ}\text{C}$  value to the value at  $T_{\text{MIN}}$  or  $T_{\text{MAX}}$ .

### QUANTIZATION UNCERTAINTY

Analog-to-digital converters have an inherent quantization error of  $\pm 1$  LSB. This error is a fundamental property of the quantization process and cannot be eliminated.

### CODE WIDTH (QUANTUM)

Code width, or quantum, is defined as the range of analog input values for which a given output code will occur. The ideal code width is 1 LSB.

## INSTALLATION

### LAYOUT PRECAUTIONS

Analog (pin 9) and digital (pin 15) commons are not connected together internally in the ADC574A, but should be connected together as close to the unit as possible and to an analog common ground plane beneath the converter on the component side of the board. In addition, a wide conductor pattern should run directly from pin 9 to the analog supply common, and a separate wide conductor pattern from pin 15 to the digital supply common.

If the single-point system common cannot be established directly at the converter, pin 9 and pin 15 should still be connected together at the converter; a single wide conductor pattern then connects these two pins to the system common. This single common path will typically carry about 1.5mA of current out of the converter. Code-dependent currents do not flow in analog (pin 9) or digital (pin 15) commons. DC currents that flow are typically  $-7\text{mA}$  in pin 9 and  $-5.5\text{mA}$  in pin 15.

Coupling between analog input and digital lines should be minimized by careful layout. For instance, if the lines must cross, they should do so at right angles. Parallel analog and digital lines should be separated from each other by a pattern connected to common.

If external full scale and offset potentiometers are used, the potentiometers and associated resistors should be located as close to the ADC574A as possible. If no trim adjustments are used, the fixed resistors should likewise be as close as possible.

### POWER SUPPLY DECOUPLING

The power supplies should be bypassed with  $10\mu\text{F}$  tantalum

bypass capacitors located close to the converter to obtain noise-free operation. Noise on the power supply lines can degrade the converter's performance. Noise and spikes from a switching power supply are especially troublesome.

### ANALOG SIGNAL SOURCE IMPEDANCE

The signal source supplying the analog input signal to the ADC574A will be driving into a nominal DC input impedance of either  $5\text{k}\Omega$  or  $10\text{k}\Omega$ . However, the output impedance of the driving source should be very low, such as the output impedance provided by a wideband, fast-settling operational amplifier. Transients in A/D input current are caused by the changes in output current of the internal D/A converter as it tests the various bits. The output voltage of the driving source must remain constant while furnishing these fast current changes. If the application requires a sample hold, select a sample hold with sufficient bandwidth to preserve the accuracy or use a separate wideband buffer amplifier to lower the output impedance.

### RANGE CONNECTIONS

The ADC574A offers four standard input ranges:  $0\text{V}$  to  $-10\text{V}$ ,  $0\text{V}$  to  $-20\text{V}$ ,  $\pm 5\text{V}$ , and  $\pm 10\text{V}$ . If a  $10\text{V}$  input range is required, the analog input signal should be connected to pin 13 of the converter. A signal requiring a  $20\text{V}$  range is connected to pin 14. In either case the other pin of the two is left unconnected. Full-scale and offset adjustments are described below.

To operate the converter with a  $10.24\text{V}$  ( $2.5\text{mV}$  LSB) or  $20.48\text{V}$  ( $5\text{mV}$  LSB) input range, insert a  $200\Omega$  potentiometer in series with pin 13 for the  $10.24\text{V}$  range, or a  $500\Omega$  potentiometer in series with pin 14 for the  $20.48\text{V}$  range. Use a fixed  $50\Omega$ ,  $1\%$  resistor for  $R_2$  (Figures 2 and 3). Offset adjustment is still performed as described below. Full-scale adjustment is performed as described below but with adjustment performed using the input potentiometer instead of  $R_2$ .

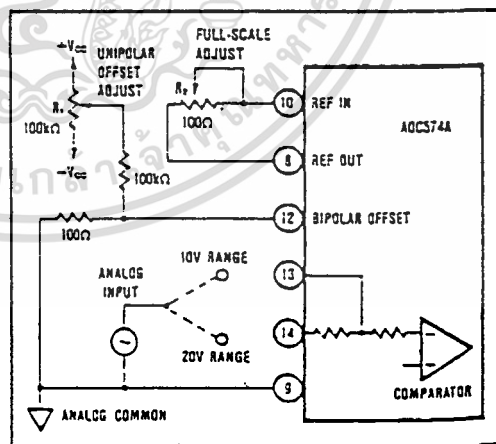


FIGURE 2. Unipolar Configuration.

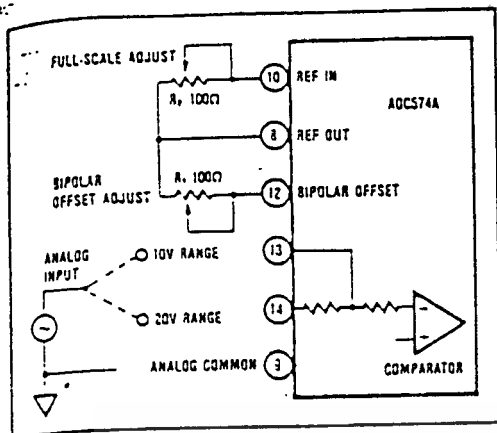


FIGURE 3. Bipolar Configuration.

## CALIBRATION

### OPTIONAL EXTERNAL FULL-SCALE AND OFFSET ADJUSTMENTS

Offset and full-scale errors may be trimmed to zero using external offset and full-scale trim potentiometers connected to the ADC574A as shown in Figures 2 and 3 for unipolar and bipolar operation.

### CALIBRATION PROCEDURE—UNIPOLAR RANGES

If adjustment of unipolar offset and full scale is not required, replace  $R_1$  with a 50Ω, 1% metal film resistor and connect pin 12 to pin 9, omitting the adjustment network.

If adjustment is required, connect the converter as shown in Figure 2. Sweep the input through the end-point transition voltage ( $0V \pm 1/2\text{LSB}$ ;  $\pm 1.22\text{mV}$  for the 10V range,  $\pm 2.44\text{mV}$  for the 20V range) that causes the output code to be DBO ON (high). Adjust potentiometer  $R_1$  until DBO is alternately toggling ON and OFF with all other bits OFF. Then adjust full scale by applying an input voltage of nominal full-scale value minus  $3/2\text{LSB}$ , the value which should cause all bits to be ON. This

value is  $-9.9963\text{V}$  for the 10V range and  $\pm 19.9927\text{V}$  for the 20V range. Adjust potentiometer  $R_2$  until bits DB1-DB11 are ON and DBO is toggling ON and OFF.

### CALIBRATION PROCEDURE—BIPOLAR RANGES

If external adjustments of full-scale and bipolar offset are not required, the potentiometers may be replaced by 50Ω, 1% metal film resistors.

If adjustments are required, connect the converter as shown in Figure 3. The calibration procedure is similar to that described above for unipolar operation, except that the offset adjustment is performed with an input voltage which is 1/2LSB above the minus full-scale value ( $-4.9988\text{V}$  for the  $\pm 5\text{V}$  range,  $-9.9976\text{V}$  for the  $\pm 10\text{V}$  range). Adjust  $R_1$  for DBO to toggle ON and OFF with all other bits OFF. To adjust full-scale, apply a DC input signal which is 3/2LSB below the nominal plus full-scale value ( $-4.9963\text{V}$  for  $\pm 5\text{V}$  range,  $-9.9927\text{V}$  for  $\pm 10\text{V}$  range) and adjust  $R_2$  for DBO to toggle ON and OFF with all other bits ON.

## CONTROLLING THE ADC574A

The Burr-Brown ADC574A can be easily interfaced to most microprocessor systems and other digital systems. The microprocessor may take full control of each conversion, or the converter may operate in a stand-alone mode, controlled only by the  $R/\overline{C}$  input. Full control consists of selecting an 8- or 12-bit conversion cycle, initiating the conversion, and reading the output data when ready—choosing either 12 bits all at once, or 8 bits followed by 4 bits in a left-justified format. The five control inputs (12/8,  $\overline{CS}$ ,  $A_0$ ,  $R/\overline{C}$ , and CE) are all TTL CMOS-compatible. The functions of the control inputs are described in Table II. The control function truth table is listed in Table III.

### STAND-ALONE OPERATION

For stand-alone operation, control of the converter is accomplished by a single control line connected to  $R/\overline{C}$ . In this mode  $\overline{CS}$  and  $A_0$  are connected to digital common and CE and 12/8 are connected to  $V_{\text{Logic}}$  ( $+5\text{V}$ ). The output data are presented as 12-bit words. The

TABLE II. ADC574A Control Line Functions.

Pin Designation	Definition	Function
CE (Pin 5)	Chip Enable (active high)	Must be high ("1") to either initiate a conversion or read output data. 0-1 edge may be used to initiate a conversion.
$\overline{CS}$ (Pin 3)	Chip Select (active low)	Must be low ("0") to either initiate a conversion or read output data. 1-0 edge may be used to initiate a conversion.
$R/\overline{C}$ (Pin 5)	Read/Convert ("1" = read, "0" = convert)	Must be low ("0") to initiate either 8 or 12-bit conversions. 1-0 edge may be used to initiate a conversion. Must be high ("1") to read output data. 0-1 edge may be used to initiate a read operation.
$A_0$ (Pin 4)	Byte Address Short Cycle	In the start-convert mode, $A_0$ selects 8-bit ( $A_0 = "1"$ ) or 12-bit ( $A_0 = "0"$ ) conversion mode. When reading output data in 2 8-bit bytes, $A_0 = "0"$ accesses 8 MSBs (high byte) and $A_0 = "1"$ accesses 4 LSBs and trailing "0s" (low byte).
12/8 (Pin 2)	Data Mode Select ("1" = 12 bits, "0" = 8 bits)	When reading output data, 12/8 = "1" enables all 12 output bits simultaneously. 12/8 = "0" will enable the MSB's or LSB's as determined by the $A_0$ line.

TABLE III. Control Input Truth Table.

CE	$\overline{CS}$	R/ $\overline{C}$	12/8	$A_0$	Operation
0	X	X	X	X	None
X	1	X	X	X	None
$\uparrow$	0	0	X	0	Initiate 12-bit conversion
A	0	0	X	1	Initiate 8-bit conversion
1	Y	0	X	0	Initiate 12-bit conversion
1	Y	0	X	1	Initiate 8-bit conversion
1	Z	$\downarrow$	X	0	Initiate 12-bit conversion
1	Z	$\downarrow$	X	1	Initiate 8-bit conversion
1	0	1	0	X	Enable 12-bit output
1	0	1	1	0	Enable 8 MSBs only
1	0	1	0	1	Enable 4 LSBs plus 4 trailing zeros

stand-alone mode is used in systems containing dedicated input ports which do not require full bus interface capability.

Conversion is initiated by a high-to-low transition of R/ $\overline{C}$ . The three-state data output buffers are enabled when R/ $\overline{C}$  is high and STATUS is low. Thus, there are two possible modes of operation: conversion can be initiated with either positive or negative pulses. In either case the R/ $\overline{C}$  pulse must remain low for a minimum of 50nsec.

Figure 4 illustrates timing when conversion is initiated by an R/ $\overline{C}$  pulse which goes low and returns to the high state during the conversion. In this case, the three-state outputs go to the high-impedance state in response to the falling edge of R/ $\overline{C}$  and are enabled for external access of the data after completion of the conversion. Figure 5 illustrates the timing when conversion is initiated by a positive R/ $\overline{C}$  pulse. In this mode the output data from the previous conversion is enabled during the positive portion of R/ $\overline{C}$ . A new conversion is started on the falling edge of R/ $\overline{C}$ , and the three-state outputs return to the high-impedance state until the next occurrence of a high R/ $\overline{C}$  pulse. Table IV lists timing specifications for stand-alone operation.

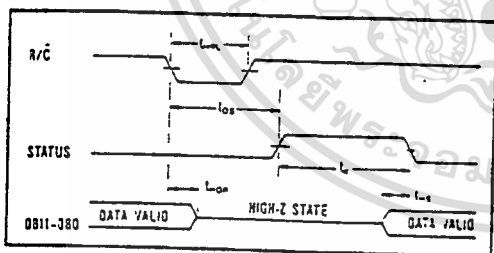


FIGURE 4. R/ $\overline{C}$  Pulse Low — Outputs Enabled After Conversion.

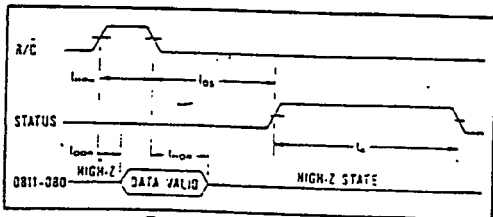


FIGURE 5. R/ $\overline{C}$  Pulse High — Outputs Enabled Only While R/ $\overline{C}$  Is High.

TABLE IV. Stand-Alone Mode Timing.

Symbol	Parameter	Min	Typ	Max	Units
$t_{low}$	Low R/ $\overline{C}$ Pulse Width	50			nS
$t_{stl}$	STS Delay from R/ $\overline{C}$			200	nS
$t_{val}$	Data Valid After R/ $\overline{C}$ Low	25			nS
$t_{st2}$	STS Delay After Data Valid	300	500	1000	nS
$t_{high}$	High R/ $\overline{C}$ Pulse Width	150			nS
$t_{acc}$	Data Access Time			150	nS

### FULLY CONTROLLED OPERATION

#### Conversion Length

Conversion length (8-bit or 12-bit) is determined by the state of the  $A_0$  input, which is latched upon receipt of a conversion start transition (described below). If  $A_0$  is latched high, the conversion continues for 8 bits. The full 12-bit conversion will occur if  $A_0$  is low. If all 12 bits are read following an 8-bit conversion, the 3LSBs (DB0-DB2) will be low (logic 0) and DB3 will be high (logic 1).  $A_0$  is latched because it is also involved in enabling the output buffers. No other control inputs are latched.

#### CONVERSION START

The converter is commanded to initiate conversion by a transition occurring on any of three logic inputs (CE,  $\overline{CS}$ , and R/ $\overline{C}$ ) as shown in Table III. Conversion is initiated by the last of the three to reach the required state and thus all three may be dynamically controlled. If necessary, all three may change states simultaneously, and the nominal delay time is the same regardless of which input actually starts conversion. If it is desired that a particular input establish the actual start of conversion, the other two should be stable a minimum of 50nsec prior to the transition of that input. Timing relationships for start of conversion timing are illustrated in Figure 6. The specifications for timing are contained in Table V.

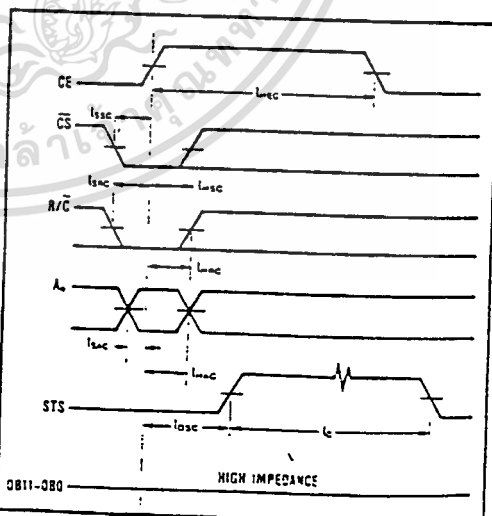


FIGURE 6. Conversion Cycle Timing.

TABLE V. Timing Specifications.

Symbol	Parameter	Min	Typ	Max	Units
<b>Convert Mode</b>					
$t_{\text{load}}$	STS delay from CE		100	200	ns
$t_{\text{wec}}$	CE Pulse width	50	30		ns
$t_{\text{sc}}$	$\overline{\text{CS}}$ to CE setup	50	20		ns
$t_{\text{sc}}$	$\overline{\text{CS}}$ low during CE high	50	20		ns
$t_{\text{sc}}$	R/ $\overline{\text{C}}$ to CE setup	50	0		ns
$t_{\text{sc}}$	R/ $\overline{\text{C}}$ low during CE high	50	20		ns
$t_{\text{sc}}$	$A_0$ to CE setup	0	0		ns
$t_{\text{sc}}$	$A_0$ valid during CE high	50	20		ns
$t_c$	Conversion time, 12 bit cycle	15	20	25	$\mu\text{s}$
$t_c$	8 bit cycle	10	13	17	$\mu\text{s}$
<b>Read Mode</b>					
$t_{\text{ao}}$	Access time from CE		75	150	ns
$t_{\text{vo}}$	Data valid after CE low	25	35		ns
$t_{\text{oh}}$	Output float delay		100	150	ns
$t_{\text{sc}}$	$\overline{\text{CS}}$ to CE setup	50	0		ns
$t_{\text{sc}}$	R/ $\overline{\text{C}}$ to CE setup	0	0		ns
$t_{\text{sc}}$	$A_0$ to CE setup	50	25		ns
$t_{\text{sc}}$	$\overline{\text{CS}}$ valid after CE low	0	0		ns
$t_{\text{sc}}$	R/ $\overline{\text{C}}$ high after CE low	0	0		ns
$t_{\text{sc}}$	$A_0$ valid after CE low	50	25		ns
$t_{\text{sc}}$	STS delay after data valid	300	500	1000	ns

NOTE: Specifications are at  $-25^\circ\text{C}$  and measured at 50% level of transitions.

The STATUS output indicates the current state of the converter by being in a high state only during conversion. During this time the three state output buffers remain in a high-impedance state, and therefore data cannot be read during conversion. During this period additional transitions of the three digital inputs which control conversion will be ignored, so that conversion cannot be prematurely terminated or restarted. However, if  $A_0$  changes state after the beginning of conversion, any additional start conversion transition will latch the new state of  $A_0$ , possibly resulting in an incorrect conversion length (8 bits vs 12 bits) for that conversion.

**READING OUTPUT DATA**

After conversion is initiated, the output data buffers remain in a high-impedance state until the following four logic conditions are simultaneously met: R,  $\overline{\text{C}}$  high, STATUS low, CE high, and  $\overline{\text{CS}}$  low. Upon satisfaction of these conditions the data lines are enabled according to the state of inputs  $\overline{\text{CS}}$  and  $A_0$ . See Figure 7 and Table V for timing relationships and specifications.

In most applications the  $\overline{\text{CS}}$  input will be hard-wired in either the high or low condition, although it is fully TTL- and CMOS-compatible and may be actively driven if desired. When  $\overline{\text{CS}}$  is high, all 12 output lines (DB0-DB11) are enabled simultaneously for full data word transfer to a 12-bit or 16-bit bus. In this situation the  $A_0$  state is ignored.

When  $\overline{\text{CS}}$  is low, the data is presented in the form of two 8-bit bytes, with selection of the byte of interest

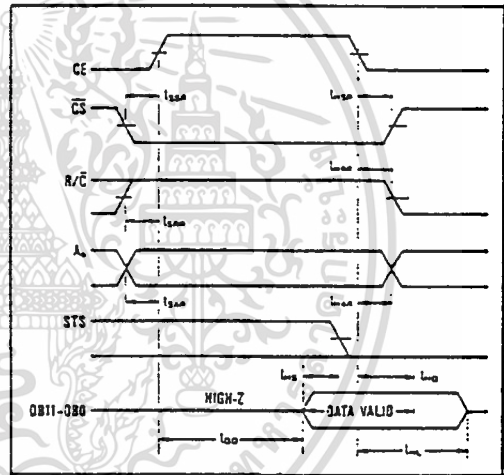


FIGURE 7. Read Cycle Timing.

accomplished by the state of  $A_0$  during the read cycle. Connection of the ADC574A to an 8-bit bus for transfer of left-justified data is illustrated in Figure 8. The  $A_0$  input is usually driven by the least significant bit of the address bus, allowing storage of the output data word in two consecutive memory locations.

When  $A_0$  is low, the byte addressed contains the 3MSBs. When  $A_0$  is high, the byte addressed contains the 4LSBs from the conversion followed by four logic zeros which have been forced by the control logic. The left-justified

	Word 1								Word 2							
Processor	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
Converter	DB11	DB10	DB9	DB8	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0	0	0	0	0

FIGURE 8. 12-Bit Data Format for 8-Bit Systems.

formats of the two 8-bit bytes are shown in Figure 8. The design of the ADCS74A guarantees that the  $A_0$  input may be toggled at any time with no damage to the converter; the outputs which are tied together as illustrated in Figure 9 cannot be enabled at the same time.

In the majority of applications the read operation will be attempted only after the conversion is complete and the STATUS output has gone low. In those situations requiring the earliest possible access to the data, the read may be started as much as  $1.15\mu\text{sec}$  ( $t_{\text{RD}} \text{ max} - t_{\text{HS}} \text{ max}$ ) before STATUS goes low. Refer to Figure 7 for these timing relationships.

## ORDERING INFORMATION

Model	Temperature Range	Linearity Error, max (T <sub>min</sub> to T <sub>max</sub> )	Resolution, No Missing Codes (T <sub>min</sub> to T <sub>max</sub> )	Full-Scale TC, max (ppm/°C)
AOC574AJH	0°C to -75°C	±1LSB	11 Bits	±45
AOC574AKH	0°C to -75°C	±1/2LSB	12 Bits	±25
AOC574ASH	-55°C to -125°C	±1LSB	11 Bits	±50
AOC574ATH	-55°C to -125°C	±1LSB	12 Bits	±25

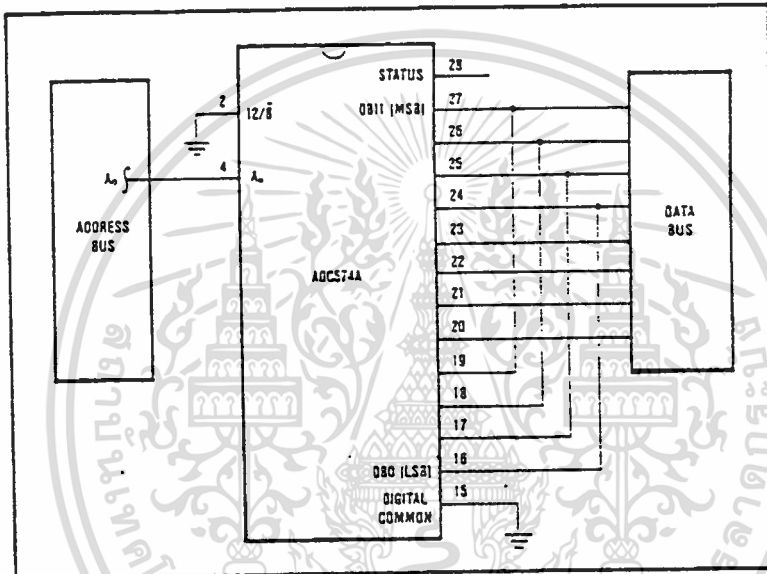


FIGURE 9. Connection to an 8-bit Bus.

DAC1208/DAC1209/DAC1210/DAC1230/DAC1231/DAC1232



## MICRO-DAC™ DAC1208, DAC1209, DAC1210, DAC1230, DAC1231, DAC1232 12-Bit, $\mu$ P Compatible, Double-Buffered D to A Converters

### General Description

The DAC1208 and the DAC1230 series are 12-bit multiplying D to A converters designed to interface directly with a wide variety of microprocessors (8080, 8048, 8085, Z-80, etc.). Double buffering input registers and associated control lines allow these DACs to appear as a two-byte "stack" in the system's memory or I/O space with no additional interfacing logic required.

The DAC1209 series provides all 12 input lines to allow single buffering for maximum throughput when used with 16-bit processors. These input lines can also be externally configured to permit an 8-bit data interface. The DAC1230 series can be used with an 8-bit data bus directly as it internally formulates the 12-bit DAC data from its 8 input lines. All of these DACs accept left-justified data from the processor.

The analog section is a precision silicon-chromium (Si-Cr) R-2R ladder network and twelve CMOS current switches. An inverted R-2R ladder structure is used with the binary weighted currents switched between the  $I_{OUT1}$  and  $I_{OUT2}$  maintaining a constant current in each ladder leg independent of the switch state. Special circuitry provides TTL logic input voltage level compatibility.

The DAC1208 series and DAC1230 series are the 12-bit members of a family of microprocessor compatible DACs (MICRO-DACs™). For applications requiring other resolutions, the DAC1000 series for 10-bit and DAC0830 series for 8-bit are available alternatives.

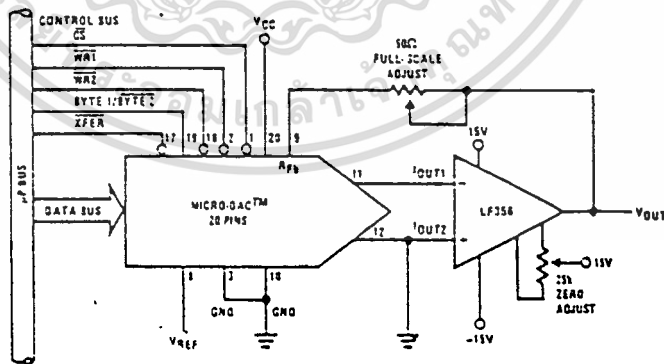
### Features

- Linearity specified with zero and full-scale adjust only
- Direct interface to all popular microprocessors
- Double-buffered, single-buffered or flow through digital data inputs
- Logic inputs which meet TTL voltage level specs (1.4V logic threshold)
- Works with  $\pm 10V$  reference—full 4-quadrant multiplication
- Operates stand-alone (without  $\mu P$ ) if desired
- All parts guaranteed 12-bit monotonic
- DAC1230 series is pin compatible with the DAC0830 series 8-bit MICRO-DACs

### Key Specifications

- Current Settling Time 1  $\mu$ s
- Resolution 12 Bits
- Linearity (Guaranteed over temperature) 10, 11, or 12 Bits of FS
- Gain Tempco 1.3 ppm/°C
- Low Power Dissipation 20 mW
- Single Power Supply 5  $V_{CC}$  to 15  $V_{CC}$

### Typical Application



TL/H/5690-1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 .ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### Absolute Maximum Ratings

If Military/Aerospace specified devices are required, contact the National Semiconductor Sales Office/Distributors for availability and specifications. (Notes 1 and 2)

Supply Voltage ( $V_{CC}$ )	17 $V_{CC}$
Voltage at Any Digital Input	$V_{CC}$ to GND
Voltage at $V_{REF}$ Input	$\pm 25V$
Storage Temperature Range	$-65^{\circ}C$ to $+150^{\circ}C$
Package Dissipation at $T_A = 25^{\circ}C$ (Note 3)	500 mW
DC Voltage Applied to $I_{OUT1}$ or $I_{OUT2}$ (Note 4)	$-100$ mV to $V_{CC}$
ESD Susceptibility	800V

### Operating Conditions

Lead Temperature (Soldering, 10 seconds)	$300^{\circ}C$
Temperature Range	$T_{MIN} \leq T_A \leq T_{MAX}$
DAC1208LCJ, DAC1209LCJ, DAC1210LCJ, DAC1230LCJ, DAC1231LCJ, DAC1232LCJ	$-40^{\circ}C \leq T_A \leq +85^{\circ}C$
DAC1208LCJ-1, DAC1209LCJ-1, DAC1210LCJ-1, DAC1230LCJ-1, DAC1231LCJ-1, DAC1232LCJ-1	$0^{\circ}C \leq T_A \leq +70^{\circ}C$
Range of $V_{CC}$	$4.75 V_{CC}$ to $16 V_{CC}$
Voltage at Any Digital Input	$V_{CC}$ to GND

### Electrical Characteristics

$V_{REF} = 10,000 V_{CC}$ ,  $V_{CC} = 11.4 V_{CC}$  to  $15.75 V_{CC}$  unless otherwise noted. Boldface limits apply from  $T_{MIN}$  to  $T_{MAX}$  (see Note 13); all other limits  $T_A = T_J = 25^{\circ}C$ .

Parameter	Conditions	Notes	Typ (Note 10)	Tested Limit (Note 5)	Design Limit (Note 6)	Units
Resolution			12	12	12	Bits
Linearity Error (End Point Linearity)	Zero and Full-Scale Adjusted DAC1208, DAC1230 DAC1209, DAC1231 DAC1210, DAC1232	4, 7, 13		<b>0.012</b> 0.024 0.050	<b>0.012</b> 0.024 0.05	Bits % of FSR % of FSR % of FSR
Differential Non-Linearity	Zero and Full-Scale Adjusted DAC1208, DAC1230 DAC1209, DAC1231 DAC1210, DAC1232	4, 7, 13		<b>0.018</b> 0.024 0.050	<b>0.018</b> 0.024 0.05	% of FSR % of FSR % of FSR
Monotonicity		4	12	12	12	Bits
Gain Error (Min)	Using Internal $R_{FB}$	7	-0.1	0.0		% of FSR
Gain Error (Max)	$V_{REF} = \pm 10V, \pm 1V$	7	-0.1	-0.2		% of FSR
Gain Error Tempco		7	$\pm 1.3$		$\pm 6.0$	ppm of FSR/ $^{\circ}C$
Power Supply Rejection	All Digital Inputs Latched High	7	$\pm 3.0$	$\pm 30$		ppm of FSR/V
Reference Input Resistance (Min)		13	15	10	10	k $\Omega$
Reference Input Resistance (Max)		13	15	20	20	k $\Omega$
Output Feedthrough Error	$V_{REF} = 20$ Vp-p, $f = 100$ kHz All Data Inputs Latched Low	9	3.0			mVp-p
Output Capacitance	All Data Inputs Latched High $I_{OUT1}$ All Data Inputs $I_{OUT2}$ Latched Low $I_{OUT1}$ Latched Low $I_{OUT2}$				200 70 70 200	pF pF pF pF
Supply Current Drain		13		2.0	2.5	mA
Output Leakage Current						
$I_{OUT1}$	All Data Inputs Latched Low	11, 13	0.1	15	15	nA
$I_{OUT2}$	All Data Inputs Latched High	11, 13	0.1	15	15	nA
Digital Input Threshold	Low Threshold	13		0.8	0.8	$V_{CC}$
	High Threshold	13		2.2	2.2	$V_{CC}$
Digital Input Currents	Digital Inputs $< 0.8V$	13		-200	-200	$\mu A_{DC}$
	Digital Inputs $> 2.2V$	13		10-	10	$\mu A_{DC}$

### Electrical Characteristics (Continued)

$V_{REF} = 10.000 V_{DC}$ ,  $V_{CC} = 11.4 V_{DC}$  to  $15.75 V_{DC}$  unless otherwise noted. Boldface limits apply from  $T_{MIN}$  to  $T_{MAX}$  (see Note 13); all other limits  $T_A = T_J = 25^\circ C$ .

Symbol	Parameter	Conditions	See Note	Typ (Note 10)	Tested Limit (Note 5)	Design Limit (Note 6)	Units
<b>AC CHARACTERISTICS</b>							
$t_s$	Current Setting Time	$V_{IL} = 0V, V_{IH} = 5V$		1.0			$\mu s$
$t_w$	Write and XFER Pulse Width Min.	$V_{IL} = 0V, V_{IH} = 5V$	8	50		<b>320</b> <b>320</b>	
$t_{DS}$	Data Setup Time Min.	$V_{IL} = 0V, V_{IH} = 5V$		70		<b>320</b> <b>320</b>	
$t_{DH}$	Data Hold Time Min.	$V_{IL} = 0V, V_{IH} = 5V$		30		90 <b>90</b>	ns
$t_{CS}$	Control Setup Time Min.	$V_{IL} = 0V, V_{IH} = 5V$		60		<b>320</b> <b>320</b>	
$t_{CH}$	Control Hold Time Min.	$V_{IL} = 0V, V_{IH} = 5V$		0		10	

Note 1: Absolute Maximum Ratings indicate limits beyond which damage to the device may occur. DC and AC electrical specifications do not apply when operating the device beyond its specified operating conditions.

Note 2: All voltages are measured with respect to GND, unless otherwise specified.

Note 3: This 500 mW specification applies for all packages. The low intrinsic power dissipation of this part (and the fact that there is no way to significantly modify the power dissipation) removes concern for heat sinking.

Note 4: Both  $I_{OUT1}$  and  $I_{OUT2}$  must go to ground or the virtual ground of an operational amplifier. The linearity error is degraded by approximately  $V_{CS} - V_{REF}$ . For example, if  $V_{REF} = 10V$  then a 1 mV offset,  $V_{CS}$ , on  $I_{OUT1}$  or  $I_{OUT2}$  will introduce an additional 0.01% linearity error.

Note 5: Tested and guaranteed to National's AOQL (Average Outgoing Quality Level).

Note 6: Design limits are guaranteed but not 100% tested. These limits are not used to calculate outgoing quality levels. Guaranteed for  $V_{CC} = 11.4V$  to  $15.75V$  and  $V_{REF} = -10V$  to  $-10V$ .

Note 7: The unit FSR stands for full-scale range. Linearity Error and Power Supply Rejection specs are based on this unit to eliminate dependence on a particular  $V_{REF}$  value to indicate the true performance of the part. The Linearity Error specification of the DAC1208 is 0.012% of FSR(max). This guarantees that after performing a zero and full-scale adjustment, the plot of the 4096 analog voltage outputs will each be within  $0.012\% \times V_{REF}$  of a straight line which passes through zero and full-scale. The unit ppm of FSR (parts per million of full-scale range) and ppm of FS (parts per million of full-scale) are used for convenience to define specs of very small percentage values, typical of higher accuracy converters. In this instance, 1 ppm of FSR =  $V_{REF}/10^6$  is the conversion factor to provide an actual output voltage quantity. For example, the gain error tempco spec of  $\pm 6$  ppm of FS/ $^\circ C$  represents a worst-case full-scale gain error change with temperature from  $-40^\circ C$  to  $+35^\circ C$  of  $\pm (6)(V_{REF}/10^6)(125^\circ C)$  or  $\pm 0.75 (10^{-3}) V_{REF}$  which is  $\pm 0.075\%$  of  $V_{REF}$ .

Note 8: This spec implies that all parts are guaranteed to operate with a write pulse or transfer pulse width ( $t_w$ ) of 520 ns. A typical part will operate with  $t_w$  of only 100 ns. The entire write pulse must occur within the valid data interval for the specified  $t_w$ ,  $t_{DS}$ ,  $t_{DH}$  and  $t_s$  to apply.

Note 9: To achieve this low feedthrough in the D package, the user must ground the metal lid. If the lid is left floating the feedthrough is typically 5 mV.

Note 10: Typicals are at  $25^\circ C$  and represent the most likely parametric norm.

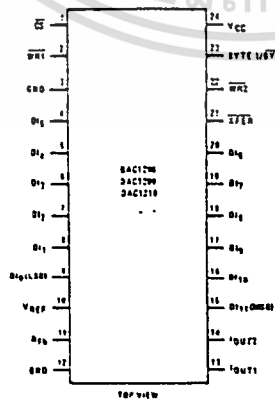
Note 11: A 10 nA leakage current with  $R_{DS} = 20k$  and  $V_{REF} = 10V$  corresponds to a zero error of  $(10 \times 10^{-9} \times 20 \times 10^3) \times 100\% / 10V$  or 0.002% of FS.

Note 12: human body model, 100 pF discharged through a 1.5 k $\Omega$  resistor.

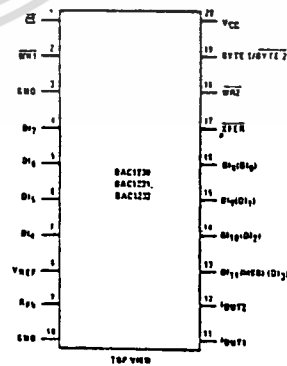
Note 13: Tested limit for -1 suffix parts applies only at  $25^\circ C$ .

### Connection Diagrams

Dual-In-Line Package



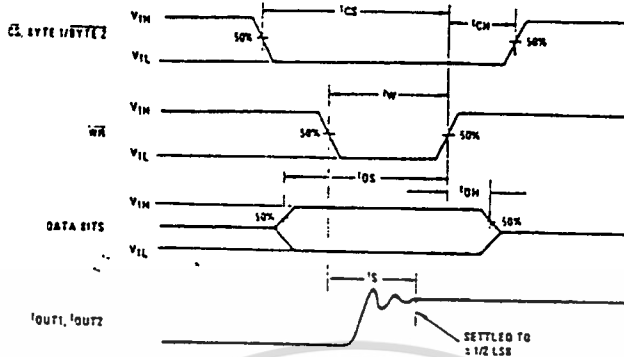
Dual-In-Line Package



See Ordering Information

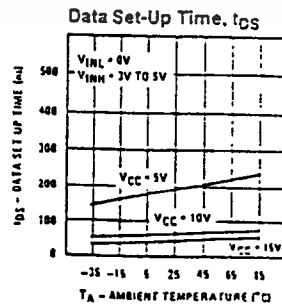
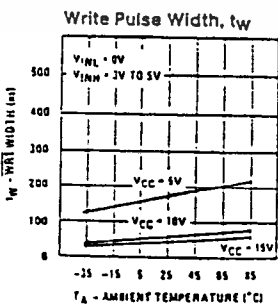
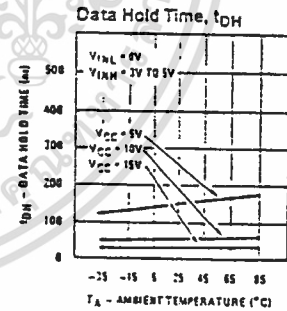
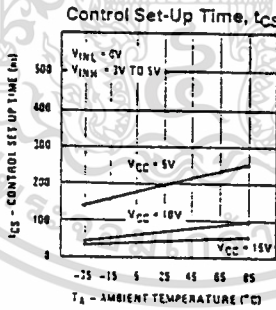
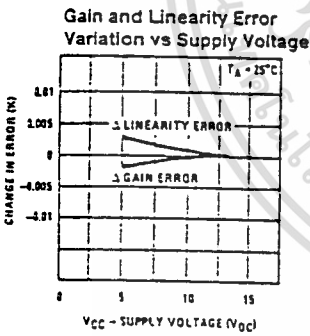
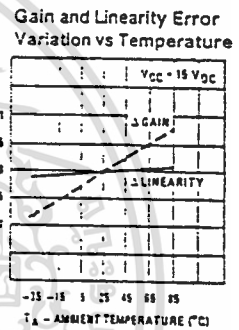
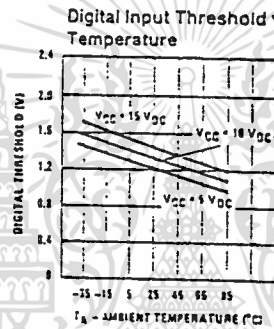
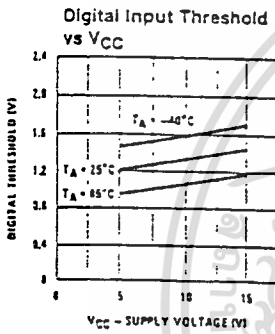
TL/H/5690-2

### Switching Waveforms



TUM/5690-3

### Typical Performance Characteristics



TUM/5690-4

## Definition of Package Pinouts

**CONTROL SIGNALS** (all control signals are level actuated)

**$\overline{CS}$ :** Chip Select (active low). The  $\overline{CS}$  will enable  $\overline{WR1}$ .

**$\overline{WR1}$ :** Write 1. The active low  $\overline{WR1}$  is used to load the digital data bits (DI) into the input latch. The data in the input latch is latched when  $\overline{WR1}$  is high. The 12-bit input latch is split into two latches. One holds the first 8 bits, while the other holds 4 bits. The Byte 1/Byte 2 control pin is used to select both latches when Byte 1/Byte 2 is high or to overwrite the 4-bit input latch when in the low state.

**Byte 1/Byte 2:** Byte Sequence Control. When this control is high, all 12 locations of the input latch are enabled. When low, only the four least significant locations of the input latch are enabled.

**$\overline{WR2}$ :** Write 2 (active low). The  $\overline{WR2}$  will enable  $\overline{XFER}$ .

**$\overline{XFER}$ :** Transfer Control Signal (active low). This signal, in combination with  $\overline{WR2}$ , causes the 12-bit data which is available in the input latches to transfer to the DAC register.

**DI<sub>0</sub> to DI<sub>11</sub>:** Digital Inputs. DI<sub>0</sub> is the least significant digital input (LSB) and DI<sub>11</sub> is the most significant digital input (MSB).

**I<sub>OUT1</sub>:** DAC Current Output 1. I<sub>OUT1</sub> is a maximum for a digital code of all 1s in the DAC register, and is zero for all 0s in the DAC register.

**I<sub>OUT2</sub>:** DAC Current Output 2. I<sub>OUT2</sub> is a constant minus I<sub>OUT1</sub>, or I<sub>OUT1</sub> + I<sub>OUT2</sub> = constant (for a fixed reference voltage). This constant current is

$$V_{REF} \times \left(1 - \frac{1}{4095}\right)$$

divided by the reference input resistance.

**R<sub>FB</sub>:** Feedback Resistor. The feedback resistor is provided on the IC chip for use as the shunt feedback resistor for the external op amp which is used to provide an output voltage for the DAC. This on-chip resistor should always be used (not an external resistor) since it matches the resistors in the on-chip R-2R ladder and tracks these resistors over temperature.

**V<sub>REF</sub>:** Reference Voltage Input. This input connects an external precision voltage source to the internal R-2R ladder. V<sub>REF</sub> can be selected over the range of 10V to -10V. This is also the analog voltage input for a 4-quadrant multiplying DAC application.

**V<sub>CC</sub>:** Digital Supply Voltage. This is the power supply pin for the part. V<sub>CC</sub> can be from 5 V<sub>DC</sub> to 15 V<sub>DC</sub>. Operation is optimum for 15 V<sub>DC</sub>.

**GND:** Pins 3 and 12 of the DAC1208, DAC1209, and DAC1210 must be connected to ground. Pins 3 and 10 of

the DAC1230, DAC1231, and DAC1232 must be connected to ground. It is important that I<sub>OUT1</sub> and I<sub>OUT2</sub> are at ground potential for current switching applications. Any difference of potential (V<sub>OS</sub> on these pins) will result in a linearity change of

$$\frac{V_{OS}}{3V_{REF}}$$

For example, if V<sub>REF</sub> = 10V and these ground pins are 9 mV offset from I<sub>OUT1</sub> and I<sub>OUT2</sub>, the linearity change will be 0.03%.

## Definition of Terms

**Resolution:** Resolution is defined as the reciprocal of the number of discrete steps in the DAC output. It is directly related to the number of switches or bits within the DAC. For example, the DAC1208 has 2<sup>12</sup> or 4096 steps and therefore has 12-bit resolution.

**Linearity Error:** Linearity error is the maximum deviation from a straight line passing through the endpoints of the DAC transfer characteristic. It is measured after adjusting for zero and full-scale. Linearity error is a parameter intrinsic to the device and cannot be externally adjusted.

National's linearity test (a) and the best straight line test (b) used by other suppliers are illustrated below. The best straight line (b) requires a special zero and FS adjustment for each part, which is almost impossible for the user to determine. The end point test uses a standard zero FS adjustment procedure and is a much more stringent test for DAC linearity.

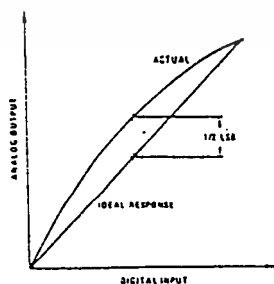
**Power Supply Sensitivity:** Power supply sensitivity is a measure of the effect of power supply changes on the DAC full-scale output.

**Settling Time:** Full-scale current settling time requires zero to full-scale or full-scale to zero output change. Settling time is the time required from a code transition until the DAC output reaches within ± 1/2 LSB of the final output value.

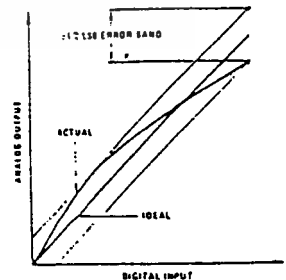
**Full-Scale Error:** Full-scale error is a measure of the output error between an ideal DAC and the actual device output. Ideally, for the DAC1208 or DAC1230 series, full-scale error is V<sub>REF</sub> - 1 LSB. For V<sub>REF</sub> = 10V and unipolar operation, V<sub>FULL-SCALE</sub> = 10.0000V - 2.44 mV = 9.9975V. Full-scale error is adjustable to zero.

**Differential Non-Linearity:** The difference between any two consecutive codes in the transfer curve from the theoretical 1 LSB is differential non-linearity.

**Monotonic:** If the output of a DAC increases for increasing digital input code, then the DAC is monotonic. A 12-bit DAC which is monotonic to 12 bits simply means that input increasing digital input codes will produce an increasing analog output.



a) End Point Test After Zero and FS Adjust



b) Shifting FS Adjust to Pass Best Straight Line Test

TL/M/5690

## Application Hints

### 1.0 DIGITAL INTERFACE

These DACs are designed to provide all of the necessary digital input circuitry to permit a direct interface to a wide variety of microprocessor systems. The timing and logic level convention of the input control signals allow the DACs to be treated as a typical memory device or I/O peripheral with no external logic required in most systems. Essentially these DACs can be mapped as a two-byte stack in memory (or I/O space) to receive their 12 bits of input data in two successive 8-bit data writing sequences. The DAC1230 series is intended for use in systems with an 8-bit data bus. The DAC1208 series provides all 12 digital input lines which can be externally configured to be controlled from an 8-bit bus or can be driven directly from a 16-bit data bus.

All of the digital inputs to these DACs contain a unique threshold regulator circuit to maintain TTL voltage level compatibility independent of the applied  $V_{CC}$  to the DAC. Any input can also be driven from higher voltage CMOS logic levels in non-microprocessor based systems. To prevent damage to the chip from static discharge, all unused digital inputs should be tied to  $V_{CC}$  or ground. As a troubleshooting aid, if any digital input is inadvertently left floating, the DAC will interpret the pin as a logic "1".

Double buffered digital inputs allow the DAC to internally format the 12-bit word used to set the current switching R-2R ladder network (see section 2.0) from two 8-bit data write cycles. Figures 1 and 2 show the internal data registers and their controlling logic circuitry. The timing diagrams for updating the DAC output are shown in sections 1.1, 1.2 and 1.3 for three possible control modes. The method used depends strictly upon the particular application.

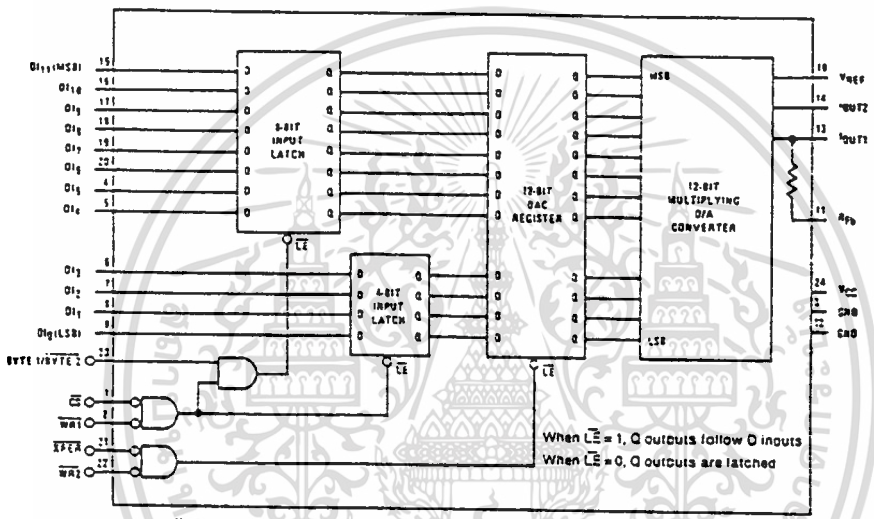


FIGURE 1. DAC1208, DAC1209, DAC1210 Functional Diagram

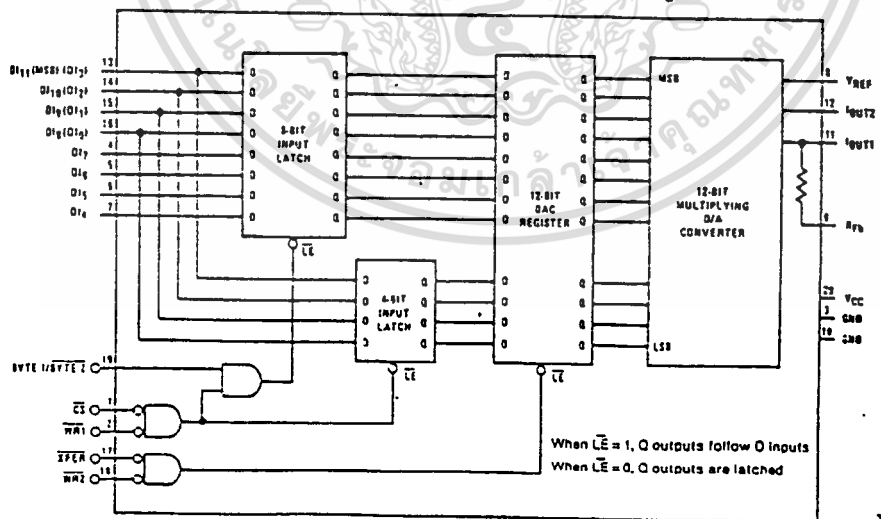


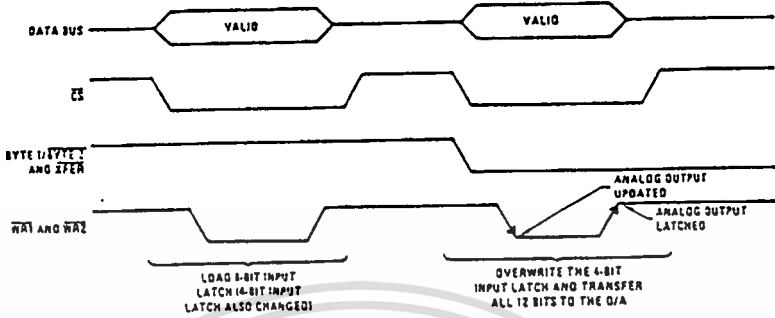
FIGURE 2. DAC1230, DAC1231, DAC1232 Functional Diagram

TL/M/5690-6

### Application Hints (Continued)

#### 1.1 Automatic Transfer

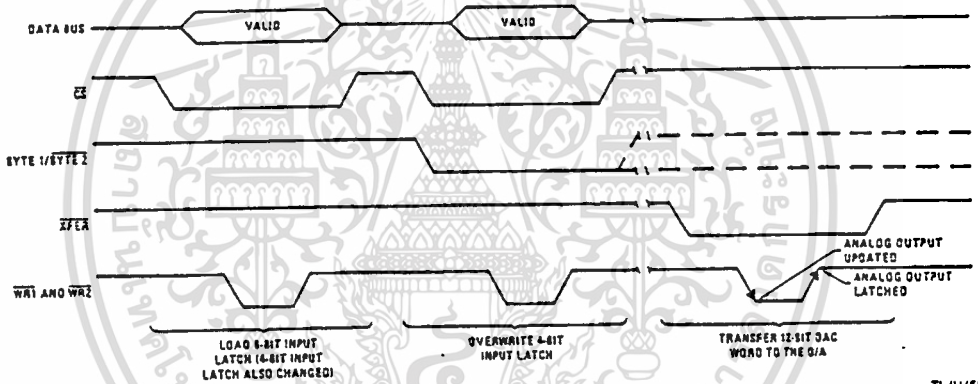
The 12-bit DAC word is automatically transferred to the DAC register and the R-2R ladder when the second write (the 4 LSBs of the data) occurs.



TL/H/5690-7

#### 1.2 Independent Processor Transfer Control

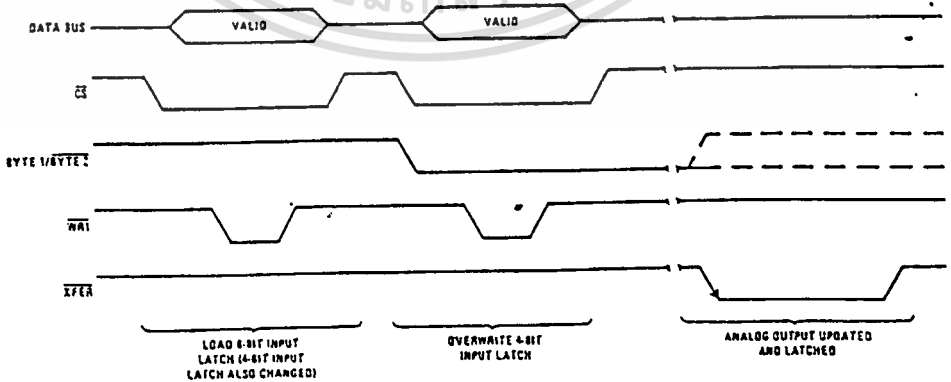
In this case a separate address is decoded to provide the  $\overline{XFER}$  signal. This allows the processor to load the next required DAC word but not change the analog output until some time later, most useful for the simultaneous updating of several DACs in a system where their  $\overline{XFER}$  lines would be tied together.



TL/H/5690-8

#### 1.3 Transfer via an External Strobe

This method is basically the same as the previous operation except the  $\overline{XFER}$  signal is provided by a device other than the processor. This allows the DAC to hold the code for a conditional analog output signal which will be required on demand from an external monitoring device (an analog voltage comparator for instance).



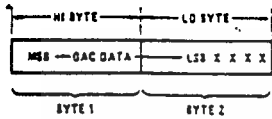
$\overline{WR2}$  tied to a logic low (0V)

TL/H/5690-9

## Application Hints (Continued)

### 1.4 Left-Justified Data Format

It is important to realize that the input registers of these DACs are arranged to accept a left-justified data word from the microprocessor with the most significant 8 bits coming first (Byte 1) and the lower 4 bits second. Left justification simply means that the binary point is assumed to be located to the left of the most significant bit. Figure 3 shows how the 12 bits of DAC data should be arranged in 2 8-bit registers of an 8-bit processor before being written to the DAC.



X = don't care

TUM/5890-10

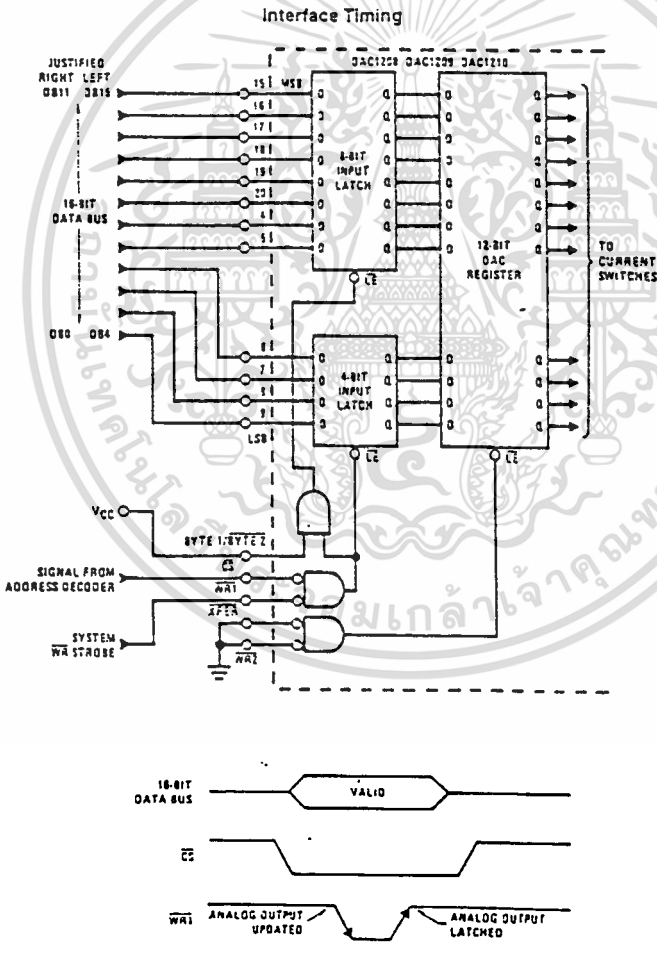
FIGURE 3. Left-Justified Data Format

### 1.5 16-Bit Data Bus Interface

The DAC1208 series provides all 12 digital input lines to permit a direct parallel interface to a 16-bit data bus. In this instance, double buffering is not always necessary (unless a simultaneous updating of several DACs or a data transfer via an external strobe is desired) so the 12-bit DAC register can be wired to flow-through whereby its Q outputs always reflect the state of its D inputs. The external connections required and the timing diagram for this single buffered application are shown in Figure 4. Note that either left or right-justified data from the processor can be accommodated with a 16-bit data bus.

### 1.6 Flow-Through Operation

Through primarily designed to provide microprocessor interface compatibility, the MICRO-DACs can easily be configured to allow the analog output to continuously reflect the state of an applied digital input. This is most useful in appli-



TUM/5890-11

FIGURE 4. 16-Bit Data Bus Interface for the DAC1208 Series

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### Application Hints (Continued)

cations where the DAC is used in a continuous feedback control loop and is driven by a binary up/down counter, or in function generation circuits where a ROM is continuously providing DAC data.

Only the DAC1208, DAC1209, DAC1210 devices can have all 12 inputs flow-through. Simply grounding  $\overline{CS}$ ,  $\overline{WR1}$ ,  $\overline{WR2}$  and  $\overline{XFER}$  and tying Byte 1/Byte 2 high allows both internal registers to follow the applied digital inputs (flow-through) and directly affect the DAC analog output.

#### 1.7 Address Decoding Tips

It is possible to map the MICRO-DACs into system ROM space to allow more efficient use of existing address decoding hardware. The DAC in effect can share the same addresses of any number of ROM locations. The ROM outputs will only be enabled by a READ of its address (gated by the system READ strobe) and the DAC will only accept data that is written to the same address (gated by the system WRITE strobe).

The Byte 1/Byte 2 control function can easily be generated by the processor's least significant address bit (A0) by placing the DAC at two consecutive address locations and utilizing double-byte WRITE instructions which automatically increment or decrement the address. The  $\overline{CS}$  and  $\overline{XFER}$  signals can then be decoded from the remaining address bits. Care must be taken in selecting the actual address used for Byte 1 of the DAC to prevent a carry (as a result of

incrementing the address for Byte 2) from propagating through the address word and changing any of the bits decoded for  $\overline{CS}$  or  $\overline{XFER}$ . Figure 5 shows how to prevent this effect.

The same problem can occur from a borrow when an auto-decremented address is used; but only if the processor's address outputs are inverted before being decoded.

#### 1.8 Control Signal Timing

When interfacing these MICRO-DACs to any microprocessor, there are two important time relationships that must be considered to insure proper operation. The first is the minimum  $\overline{WR}$  strobe pulse width which is specified as 320 ns for  $V_{CC} = 11.4V$  to 15.75V and operation over temperature, but typically a pulse width of only 250 ns is adequate. A second consideration is that the guaranteed minimum data hold time of 90 ns should be met or erroneous data can be latched. This hold time is defined as the length of time data must be held valid on the digital inputs after a qualified (via  $\overline{CS}$ )  $\overline{WR}$  strobe makes a low to high transition to latch the applied data.

If the controlling device or system does not inherently meet these timing specs the DAC can be treated as a slow memory or peripheral and utilize a technique to extend the write strobe. A simple extension of the write time, by adding a wait state, can simultaneously hold the write strobe active and data valid on the bus to satisfy the minimum  $\overline{WR}$  pulse

Write Cycle	Address Bits			
	15	2	1*	0**
First (Byte 1)	Decoded to		0	1
Second (Byte 2)	Address DAC		1	0

\*Starting with a 0 prevents a carry on address incrementing.  
 \*\*Used as Byte 1/Byte 2 Control.

FIGURE 5

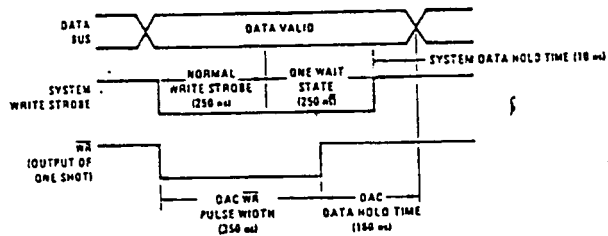
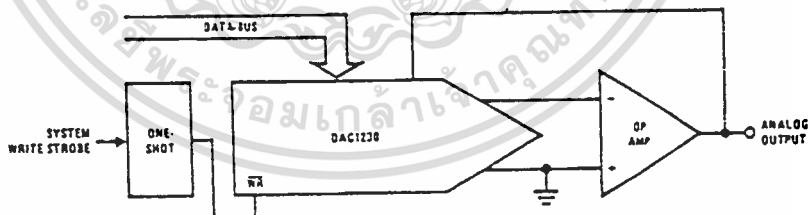


FIGURE 6. Accommodating a High Speed System

## Application Hints (Continued)

### 2.0 ANALOG APPLICATIONS

The analog output signal for these DACs is derived from a conventional R-2R current switching ladder network. A detailed description of this network can be found on the DAC1000 series data sheet. Basically, output  $I_{OUT1}$  provides a current directly proportional to the product of the applied reference voltage and the digital input word. A second output,  $I_{OUT2}$  will be a current proportional to the complement of the digital input. Specifically:

$$I_{OUT1} = \frac{V_{REF}}{R} \times \frac{D}{4096}$$

$$I_{OUT2} = \frac{V_{REF}}{R} \times \frac{4095 - D}{4096}$$

where  $D$  is the decimal equivalent of the applied 12-bit binary word (ranging from 0 to 4095),  $V_{REF}$  is the voltage applied to the  $V_{REF}$  terminal and  $R$  is the internal resistance of the R-2R ladder.  $R$  is nominally 15 k $\Omega$ .

### 2.1 Obtaining a Unipolar Output Voltage

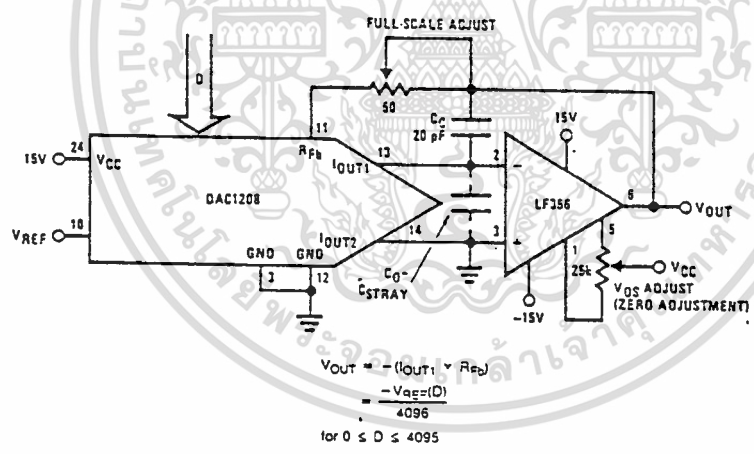
To maintain linearity of output current with changes in the applied digital code, it is important that the voltages at both of the current output pins be as near ground potential (0  $V_{DC}$ ) as possible. With  $V_{REF} = \pm 10V$  every millivolt appearing at either  $I_{OUT1}$  or  $I_{OUT2}$  will cause a 0.01% linearity error. In most applications this output current is converted to a voltage by using an op amp as shown in Figure 9.

The inverting input of the op amp is a virtual ground created by the feedback from its output through the internal 15 k $\Omega$  resistor,  $R_{FB}$ . All of the output current (determined by the digital input and the reference voltage) will flow through  $R_{FB}$  to the output of the amplifier. Two-quadrant operation can be obtained by reversing the polarity of  $V_{REF}$  thus causing  $I_{OUT1}$  to flow into the DAC and be sourced from the output of the amplifier. The output voltage, in either case, is always equal to  $I_{OUT1} \times R_{FB}$  and is the opposite polarity of the reference voltage.

The reference can be either a stable DC voltage source or an AC signal anywhere in the range from -10V to +10V. The DAC can be thought of as a digitally controlled attenuator: the output voltage is always less than the applied reference voltage. The  $V_{REF}$  terminal of the device presents a nominal impedance of 15 k $\Omega$  to ground to external circuitry.

Always use the internal  $R_{FB}$  resistor to create an output voltage since this resistor matches (and tracks with temperature) the value of the resistors used to generate the output current ( $I_{OUT1}$ ).

The selected op amp should have as low a value of input bias current as possible. The product of the bias current times the feedback resistance creates an output voltage error which can be significant in low reference voltage applications. BI-FET™ op amps are highly recommended for use with these DACs because of their very low input current.



TL/H/5690-15

FIGURE 9. Unipolar Output Configuration

Application Hints (Continued)

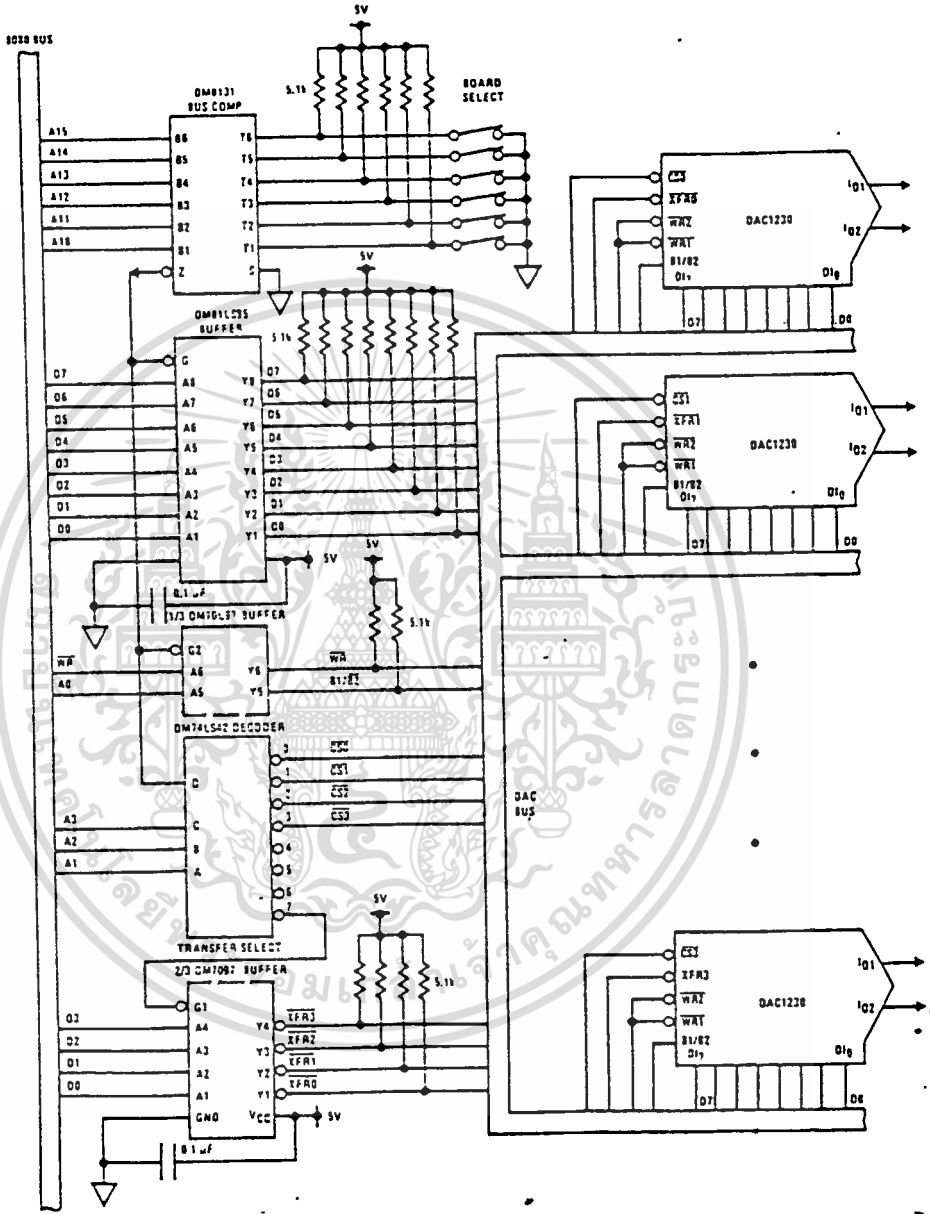


FIGURE 8. TRI-STATE® Buffers Isolate the Data and Control Lines from the DACs. A Transfer Word Provides a Flexible Update.

TL/M/5690-14

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### Application Hints (Continued)

width. If this does not provide a sufficient data hold time at the end of the write cycle, a negative edge triggered one-shot can be included between the system write strobe and the  $\overline{WR}$  pin of the DAC. This is illustrated in Figure 6 for an exemplary system which provides a 250 ns  $\overline{WR}$  strobe time with a data hold time of only 10 ns.

The proper data set-up time prior to the latching edge (low to high transition) of the  $\overline{WR}$  strobe, is insured if the  $\overline{WR}$  pulse width is within spec and the data is valid on the bus for the duration of the DAC  $\overline{WR}$  strobe.

#### 1.9 Digital Signal Feedthrough

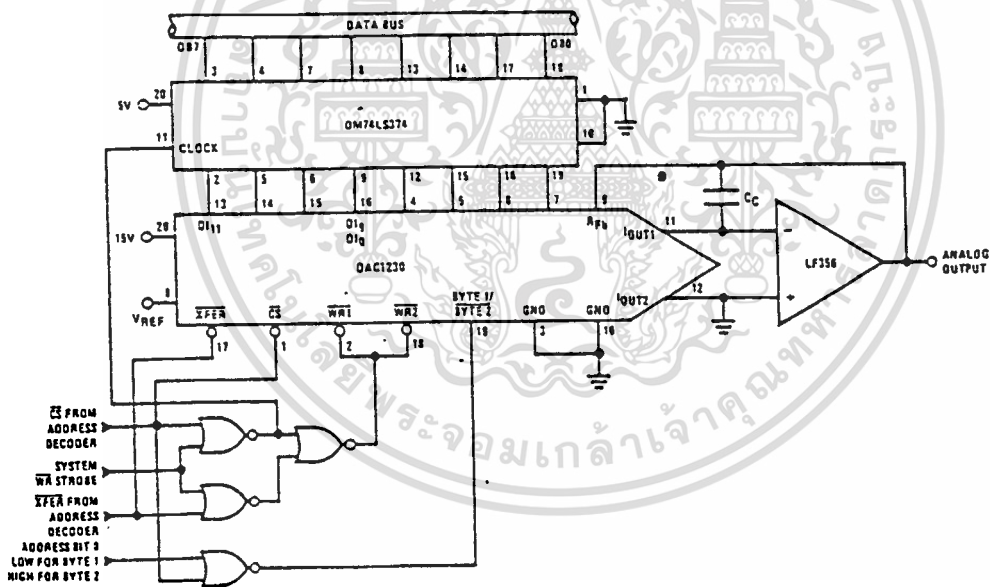
A typical microprocessor is a tremendous potential source of high frequency noise which can be coupled to sensitive analog circuitry. The fast edges of the data and address bus signals generate frequency components of 10's of megahertz and may cause fast transients to appear at the DAC output, even when data is latched internally.

In low frequency or DC applications, low pass filtering can reduce the magnitude of any fast transients. This is most

easily accomplished by over-compensating the DAC output amplifier by increasing the value of its feedback capacitor.

In applications requiring a fast output response from the DAC and op amp, filtering may not be feasible. In this event, digital signals can be completely isolated from the DAC circuitry, by the use of a DM74LS374 latch, until a valid  $\overline{CS}$  signal is applied to update the DAC. This is shown in Figure 7.

A single TRI-STATE<sup>®</sup> data buffer such as the DM81LS95 can be used to isolate any number of DACs in a system. Figure 8 shows this isolating circuitry and decoding hardware for a multiple DAC analog output card. Pull-up resistors are used on the buffer outputs to limit the impedance at the DAC digital inputs when the card is not selected. A unique feature of this card is that the DAC  $\overline{XFER}$  strobes are controlled by the data bus. This allows a very flexible update of any combination of analog outputs via a transfer word which would contain a zero in the bit position assigned to any of the DACs required to change to a new output value.



TL/H/5690-13

FIGURE 7. Isolating Data Bus from DAC Circuitry to Eliminate Digital Noise Coupling

## Application Hints (Continued)

Transient response and settling time of the op amp are important in fast data throughput applications. The largest stability problem is the feedback pole created by the feedback resistance,  $R_{FB}$ , and the output capacitance of the DAC. This appears from the op amp output to the (-) input and includes the stray capacitance at this node. Addition of a lead capacitance,  $C_C$  in Figure 9, greatly reduces overshoot and ringing at the output for a step change in DAC output current.

### 2.1.1 Zero and Full-Scale Adjustments

For accurate conversions, the input offset voltage of the output amplifier must always be nulled. Amplifier offset errors create an overall degradation of DAC linearity.

The fundamental purpose of zeroing is to make the voltage appearing at the DAC outputs as near 0  $V_{CC}$  as possible. This is accomplished by shorting out  $R_{FB}$ , the amplifier feedback resistor, and adjusting the  $V_{OS}$  nulling potentiometer of the op amp until the output reads zero volts. This is done, of course, with an applied digital code of all zeros if  $I_{OUT1}$  is driving the op amp (all ones for  $I_{OUT2}$ ). The short around  $R_{FB}$  is then removed and the converter is zero adjusted.

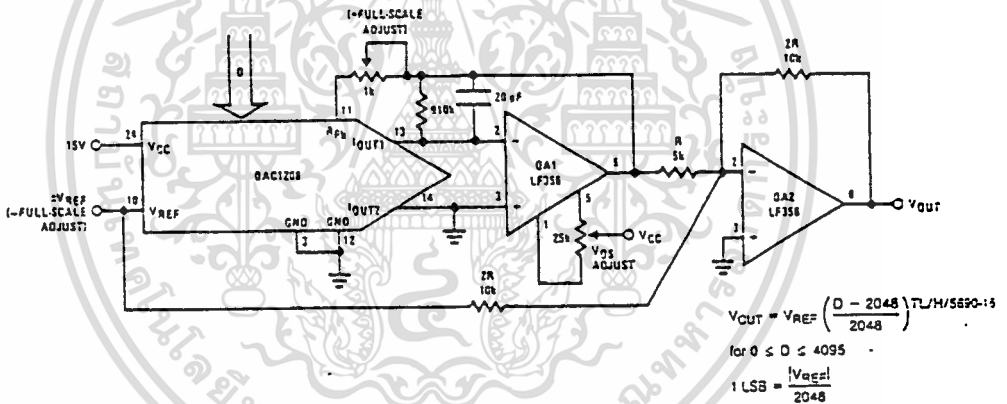
A unique feature of this series of DACs is that the full-scale or gain error is guaranteed to be negative. The gain error specification is a measure of how close the value of the

internal feedback resistor,  $R_{FB}$ , matches the R-2R ladder resistors. A negative gain error indicates that  $R_{FB}$  is a smaller resistance value than it should be. To adjust this gain error, some resistance must always be added in series with  $R_{FB}$ . The 50  $\Omega$  potentiometer shown is sufficient to adjust the worst-case gain error for these devices.

### 2.2 Bipolar Output Voltage from a Fixed Reference

The addition of a second op amp to the unipolar circuit can generate a bipolar output voltage from a fixed reference voltage. This, in effect, gives sign significance to the MSB of the digital input word to allow two quadrant multiplication of the reference voltage. The polarity of the reference can also be reversed to realize full 4-quadrant multiplication. This circuit is shown in Figure 10.

This configuration features several improvements over existing circuits for a bipolar output shown with other multiplying DACs. Only the offset voltage of amplifier 1 affects the linearity of the DAC. The offset voltage error of the second op amp (although a constant output error) has no effect on linearity. In addition, this configuration offers a non-interactive positive and negative full-scale calibration procedure.



Input Code MSB.....LSB	Ideal $V_{OUT}$	
	$+V_{REF}$	$-V_{REF}$
111111111111	$V_{REF} - 1 \text{ LSB}$	$- V_{REF}  + 1 \text{ LSB}$
110000000000	$V_{REF}/2$	$- V_{REF} /2$
100000000000	0	0
011111111111	$-1 \text{ LSB}$	$+1 \text{ LSB}$
001111111111	$-\frac{V_{REF}}{2} - 1 \text{ LSB}$	$\frac{ V_{REF} }{2} + 1 \text{ LSB}$
000000000000	$-V_{REF}$	$+ V_{REF} $

FIGURE 10. Bipolar Output Voltage Configuration

## Application Hints (Continued)

### 2.2.1 Zero and Full-Scale Adjustments

To calibrate the bipolar output circuit, three adjustments are required. The first step is to set all of the digital inputs LOW (to force  $I_{OUT1}$  to 0) then null the  $V_{OS}$  of amplifier 1 by setting the voltage at its inverting input (pin 2) to zero volts. Next, with a code of all zeros still applied, adjust "-full-scale adjust", the reference voltage, for  $V_{OUT} = \pm |V_{REF}$  ideal. The polarity of the output voltage at this time will be opposite that of the applied reference. Finally, set all of the digital inputs HIGH and adjust "+full-scale adjust" for

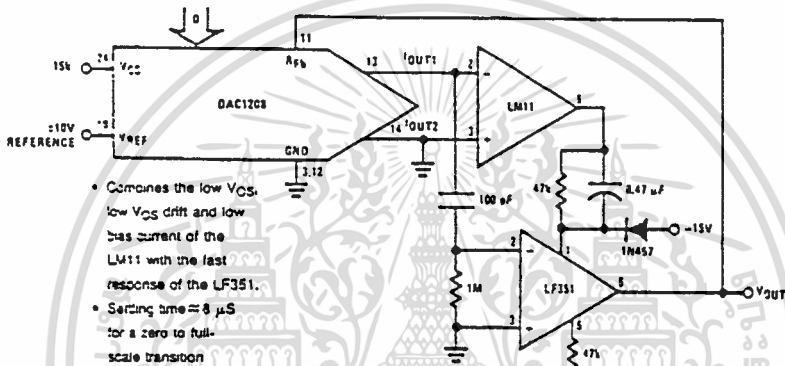
$$V_{OUT} = V_{REF} \frac{2047}{2048}$$

The polarity of the output will be the same as that of the reference voltage.

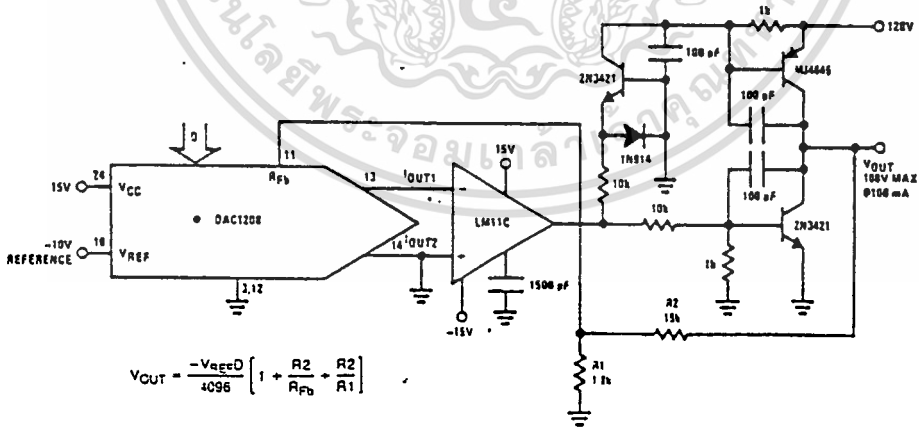
### 3.0 APPLICATION IDEAS

In this section the digital input word is represented by the letter D and is equal to the decimal equivalent of the 12-bit binary input. Hence D can be any integer value between 0 and 4095.

Composite Amplifier for Good DC Characteristics and Fast Output Response



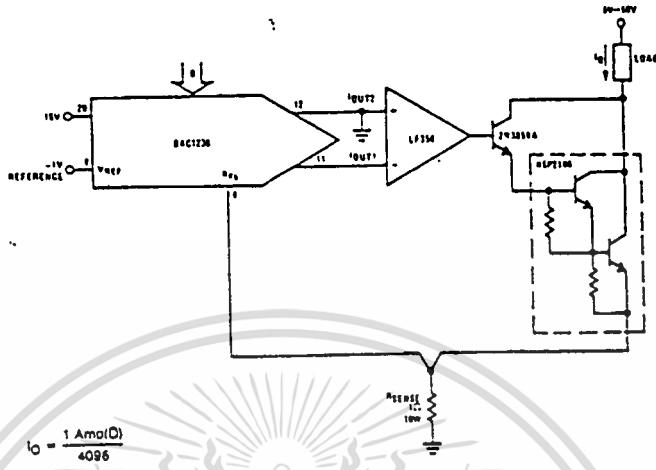
High Voltage, Power DAC



TU/H/5690-17

**Application Hints (Continued)**

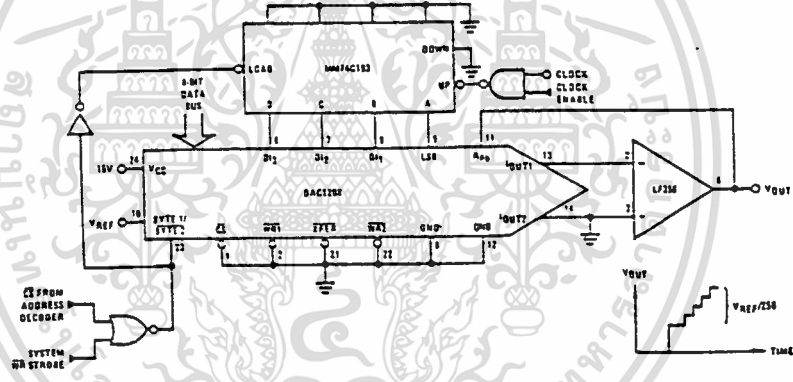
**High Current Controller**



$$I_o = \frac{1 \text{ Amp}(D)}{4096}$$

TL/H/5690-18

**8-Bit Course, 4-Bit Vernier DAC**



TL/H/5690-20

**Ordering Information**

Part Number	Non-Linearity	Package	Temperature Range
DAC1208LCJ	0.012%	J24A Cerdip	-40°C to +85°C
DAC1208LCJ-1	0.012%	J24A Cerdip	0°C to +70°C
DAC1209LCJ	0.024%	J24A Cerdip	-40°C to +85°C
DAC1209LCJ-1	0.024%	J24A Cerdip	0°C to +70°C
DAC1210LCJ	0.050%	J24A Cerdip	-40°C to +85°C
DAC1210LCJ-1	0.050%	J24A Cerdip	0°C to +70°C
DAC1230LCJ	0.012%	J20A Cerdip	-40°C to +85°C
DAC1230LCJ-1	0.012%	J20A Cerdip	0°C to +70°C
DAC1231LCJ	0.024%	J20A Cerdip	-40°C to +85°C
DAC1231LCJ-1	0.024%	J20A Cerdip	0°C to +70°C
DAC1232LCJ	0.050%	J20A Cerdip	-40°C to +85°C
DAC1232LCJ-1	0.050%	J20A Cerdip	0°C to +70°C

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นิยมนำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## กิตติกรรมประกาศ

โครงการนี้สามารถสำเร็จลุล่วงตามวัตถุประสงค์ที่วางไว้ เนื่องจากได้รับคำแนะนำ และความช่วยเหลือจากอาจารย์หลายๆท่าน คณะผู้จัดทำโครงการขอขอบพระคุณอาจารย์ทุกท่าน ในภาควิชาเทคโนโลยีการวัดคุมทางอุตสาหกรรม ไว้เป็นอย่างสูง ณ.ที่นี้ด้วย

โครงการนี้จำเป็นต้องมีการใช้ห้องทดลอง(PROCESS CONTROL LAB) ซึ่งท่านอาจารย์ได้ให้คำปรึกษา และความช่วยเหลือในการแก้ปัญหาที่เกิดขึ้น รวมถึงการอำนวยความสะดวกทางด้านสถานที่ เครื่องมือและอุปกรณ์ต่างๆ และต้องขอขอบคุณเพื่อนๆ ทุกคน ที่ช่วยเหลือและให้กำลังใจ หากมีข้อบกพร่องและความผิดพลาดของโครงการนี้ คณะผู้จัดทำขอน้อมรับผิดและขออภัยไว้ ณ.ที่นี้ด้วย

คณะผู้จัดทำ  
พฤศจิกายน 2538



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# หนังสืออ้างอิง

ธานีินทร์ ถาวรศาสนวงศ์, ทินกร ดุ๊ก . การอินเทอร์เฟซ IBM PC

รองศาสตราจารย์ มัณณา ปราการสมุทร . การเขียนชุดคำสั่งภาษา C  
ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย

**Ernest F. Johnson . Automatic Process Control.** Mc Graw Hill

**Zhen-yu Zhao , Masayoshi Tomizuka , Satoru Isaka . Fuzzy Gain Scheduling of**  
**PID Controllers IEEE Trans. syst. Man, Cybern. , VOL 23 pp. 1392-1398, 1993**

**Kandel And Langholz . Fuzzy Control System**

**William H. Murry And Charis H. Pappas . Windows 3.1 programming**  
Mc Graw Hill

