

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

อุปกรณ์ควบคุมความดันในถังปิด

TANK PRESSURE CONTROLLER



T104087



นายธงชัย ปานเลิศ  
นายพีรเดช โปธิศรี  
นายสิทธิชัย ศรีภูมิ

เลขหมู่.....  
เลขทะเบียน 104087  
วัน,เดือน,ปี 28 ต.ค. 2552

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

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมการวัดคุม

ภาควิชาวิศวกรรมการวัดคุม คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษานี้เท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# TANK PRESSURE CONTROLLER



A THESIS SUBMITTED IN PARTIAL FULFILLMENT  
OF THE REQUIREMENT FOR THE DEGREE OF  
BACHELOR OF ENGINEER IN INSTRUMENTATION ENGINEERING  
DEPARTMENT OF INSTRUMENTATION ENGINEERING  
FACULTY OF ENGINEERING  
KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

2008

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาควิชาวิศวกรรมการวัดคุม  
คณะวิศวกรรมศาสตร์  
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง  
ใบรับรองปริญญาโท

หัวข้อปริญญาโท อุปกรณ์ควบคุมความดันในถังปิด  
TANK PRESSURE CONTROLLER

นักศึกษาผู้จัดทำ นายธงชัย ปานเลิศ รหัสนักศึกษา 49015369  
นายพีรเดช โพธิ์ศรี รหัสนักศึกษา 49015383  
นายสิทธิชัย ศรีภูมิ รหัสนักศึกษา 49015397

ปริญญา วิศวกรรมศาสตรบัณฑิต  
สาขา วิศวกรรมการวัดคุม  
ปีการศึกษา 2551

อาจารย์ผู้ควบคุมวิทยานิพนธ์	ลายมือชื่อ
ผศ.เชื้อ นกอยู่ รศ.ดร.สาธิต อินทจักร์	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อปริญญานิพนธ์	อุปกรณ์ควบคุมความดันในถังปิด		
	TANK PRESSURE CONTROLLER		
นักศึกษาผู้จัดทำ	นายธงชัย	ปานเลิศ	รหัสนักศึกษา 49015369
	นายพีรเดช	โพธิ์ศรี	รหัสนักศึกษา 49015383
	นายสิทธิชัย	ศรีภูมิ	รหัสนักศึกษา 49015397
อาจารย์ที่ปรึกษา	ผศ.เชื้อ	นกออยู่	
	รศ.ดร.สาธิต	อินทจักร์	
ปีการศึกษา	2551		

### บทคัดย่อ

ปริญญานิพนธ์นี้ นำเสนอการออกแบบระบบการวัดและควบคุมกระบวนการ โดยใช้ไมโครคอนโทรลเลอร์ ARM7 ซึ่งเป็นไมโครคอนโทรลเลอร์ขนาด 32 บิต เพื่อมาออกแบบเป็นตัวควบคุมแบบพีไอดีสำหรับกระบวนการทางอุตสาหกรรม เพื่อเป็นการเพิ่มความน่าเชื่อถือและเพิ่มประสิทธิภาพในการปฏิบัติงาน ตัวควบคุมพีไอดีนี้ยังสามารถที่จะติดต่อสื่อสารผ่านโปรแกรมประยุกต์เพื่อส่งค่า,รับค่าและแสดงผลผ่านหน้าจอแสดงผล สำหรับการปรับค่าพารามิเตอร์ของตัวควบคุมพีไอดีเป็นสิ่งที่จำเป็นเพื่อให้ได้การควบคุมที่ดีที่สุดจะใช้วิธีของ Ziegler-Nichols ซึ่งการปรับค่าพารามิเตอร์ของพีไอดีได้จากหน้าจอแสดงผล

ในที่นี่ได้นำการประยุกต์ใช้ตัวควบคุมพีไอดีในการควบคุมกระบวนการความดัน

<b>Thesis</b>	Tank Pressure Controller	
<b>Authors</b>	Mr.Thongchai	Panlers
	Mr.Peeradech	Posri
	Mr.Sittichai	Sripoom
<b>Thesis Advisor</b>	Asst.Prof. Chuae	Nokyoo
	Assoc.Prof.Dr. Sathit	Intajak
<b>Year</b>	2008	

### ABSTRACT

This thesis represents instrumentation system and process control design by using 32 bits ARM7 microcontroller for PID Controller. It used to increase reliability and performance in operate. The PID Controller will communicate with computer to send, receive and monitoring. For adjustment the parameter of PID Controller to the best values by using Ziegler-Nichols method. In this case we use PID Controller for pressure control.

## กิตติกรรมประกาศ

ปริญญาบัตรฉบับนี้คงไม่อาจสำเร็จไปได้ด้วยดี หากไม่ได้รับความช่วยเหลือและความร่วมมือจากหลาย ๆ ฝ่ายด้วยกัน บุคคลที่ต้องกล่าวถึงเพราะเป็นส่วนสำคัญที่ทำให้ปริญญาบัตรนี้สำเร็จลุล่วงไปได้ด้วยดีคือ ผศ.เชื้อ นกอยู่ รศ.ดร.สาธิต อินทจักร์ และรศ.วิริยะ กองรัตน์ อาจารย์ที่ปรึกษาปริญญาบัตร ที่ให้ความเอาใจใส่ให้คำแนะนำและให้ความช่วยเหลือเสมอมา อีกทั้งภาควิชาวิศวกรรมการวัดคุมเชื้อเพื่อสถานที่และอุปกรณ์ ซึ่งต้องขอขอบพระคุณเป็นอย่างยิ่ง

ขอขอบพระคุณเป็นอย่างยิ่งสำหรับบุคคลซึ่งสำคัญสุดที่ทำให้ข้าพเจ้ามีวันนี้ได้ก็คือ บิดามารดา ผู้ที่ให้ทุกอย่างให้กำลังใจ เลี้ยงดูและมอบโอกาสที่ดีทางการศึกษามาโดยตลอดทำให้ข้าพเจ้าได้องค์ความรู้ต่าง ๆ และประสบการณ์ ซึ่งหลาย ๆ คนมิได้มีโอกาสเช่นนี้อย่างข้าพเจ้าและต้องขอขอบพระคุณอาจารย์ท่านอื่น ๆ ที่ไม่ได้ออกนามที่อยู่ภาควิชาวิศวกรรมการวัดคุมนี้ทุกท่าน ที่ให้ความช่วยเหลือและให้คำปรึกษาที่ดีและเป็นประโยชน์ สุดท้ายนี้ข้าพเจ้าขอระลึกในพระคุณ และขอกราบขอบพระคุณ ณ ที่นี้ด้วย

คณะผู้จัดทำ

# สารบัญ

	หน้า
บทคัดย่อภาษาไทย.....	I
บทคัดย่อภาษาอังกฤษ.....	II
กิตติกรรมประกาศ.....	III
สารบัญ.....	IV
สารบัญตาราง.....	VIII
สารบัญรูป.....	IX

บทที่ 1 บทนำ.....	1
1.1 ความสำคัญของปริญยานิพนธ์.....	1
1.2 วัตถุประสงค์ของปริญยานิพนธ์.....	1
1.3 ขอบเขตของปริญยานิพนธ์.....	1
1.4 ขั้นตอนการศึกษา.....	2
บทที่ 2 ทฤษฎีและหลักการ.....	3
2.1 ระบบควบคุม.....	3
2.2 องค์ประกอบของระบบควบคุม.....	3
2.3 สัญญาณอินพุท (Input signal).....	4
2.4 เครื่องควบคุม.....	5
2.4.1 เครื่องควบคุมแบบเปิด-ปิด (On-Off Control).....	5
2.4.2 กิริยาควบคุมแบบพี (P Action).....	7
2.4.3 กิริยาควบคุมแบบอินทิกรัล (I Action).....	8
2.4.4 กิริยาควบคุมแบบเดริเวทีฟ (D Action).....	9
2.4.5 เครื่องควบคุมแบบพีไอ (PI Controller).....	11
2.4.6 เครื่องควบคุมแบบพีดี (PD Controller).....	11
2.4.7 เครื่องควบคุมแบบพีไอดี (PID Controller).....	12
2.5 กระบวนการควบคุม.....	13
2.5.1 ระบบควบคุมอัตโนมัติแบบวงเปิดหรือไม่มีการป้อนกลับ.....	13
2.5.2 การควบคุมแบบคาสเคด.....	14

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## สารบัญ (ต่อ)

	หน้า
2.5.3 การควบคุมแบบสัดส่วน.....	14
2.5.4 การควบคุมแบบป้อนไปข้างหน้า.....	15
2.6 ผลตอบสนองของระบบต่อสัญญาณอินพุตชนิดต่างๆ.....	15
2.6.1 ผลตอบสนองต่อระบบอันดับหนึ่ง.....	15
2.6.1.1 ผลตอบสนองต่อสัญญาณสแต็ป.....	17
2.6.1.2 ผลตอบสนองเมื่อมีการหน่วงเวลา.....	18
2.6.2 ผลตอบสนองต่อระบบอันดับสอง.....	20
2.7 การหาค่าพารามิเตอร์ของตัวควบคุมแบบ PID ด้วยวิธีของ Ziegler-Nichols.....	24
2.7.1 Quarter decay ration response by ultimate gain.....	24
2.7.2 Process reaction.....	25
2.7.3 วิธี Trial and Error (วิธีการลองผิดลองถูก).....	27
2.8 ลักษณะอุปกรณ์การทำงานและการเลือกใช้อุปกรณ์.....	28
2.8.1 วาล์วควบคุม (Control Valve).....	28
2.8.1.1 ประเภทของวาล์วควบคุมแบ่งตามลักษณะการเปิด-ปิด.....	28
2.8.1.2 Valve Body.....	28
2.8.1.3 Valve Trim.....	29
2.8.2 การเลือกวาล์วควบคุมเพื่อควบคุมการไหล.....	29
2.8.3 คุณลักษณะการไหลของวาล์วควบคุม.....	30
2.8.3.1 คุณลักษณะเส้นตรง.....	30
2.8.3.2 Equal-Percentage Characteristic.....	30
2.8.3.3 Quick Opening Characteristic.....	31
2.9 เพรสเซอร์ทรานสดิวเซอร์.....	31
2.9.1 เอาท์พุทของเพรสเซอร์ทรานสดิวเซอร์.....	32
2.9.1.1 ชนิดเอาท์พุทมีหน่วยเป็นมิลลิโวลต์.....	32
2.9.1.2 ชนิดเอาท์พุทมีหน่วยเป็นโวลต์.....	32
2.9.1.3 ชนิดเอาท์พุทมีหน่วยเป็นมิลลิแอมป์.....	32
2.9.2 ประเภทของความดัน.....	33
2.9.2.1 ความดันเกจ.....	33

# สารบัญ (ต่อ)

	หน้า
2.9.2.2 ความดันสัมบูรณ์.....	33
2.9.2.3 ความดันแตกต่าง.....	33
2.9.2.4 สภาวะสูญญากาศ.....	34
2.10 โครงสร้างไมโครคอนโทรลเลอร์ ARM7.....	34
<b>บทที่ 3 การออกแบบและวิธีการดำเนินการ.....</b>	<b>36</b>
3.1 การออกแบบ Plant.....	36
3.1.1 สมการคุณลักษณะของกระบวนการ.....	37
3.2 การออกแบบตัวควบคุม.....	40
3.2.1 การออกแบบ Hardware.....	41
3.2.1.1 วงจรแปลงแรงดันไฟฟ้าเป็นกระแสไฟฟ้า (V/I).....	41
3.2.1.2 วงจรแปลงสัญญาณกระแสไฟฟ้าเป็นแรงดันไฟฟ้า (I/V).....	44
3.2.1.3 วงจรที่ใช้ในการจ่ายไฟ.....	45
3.2.1.4 การออกแบบ Switch.....	46
3.2.1.5 การแสดงผลทาง LCD.....	47
3.2.1.6 การออกแบบจอแสดงผลทาง Monitor.....	47
3.3 การออกแบบ Software.....	48
3.3.1 การออกแบบอัลกอริทึมสำหรับตัวควบคุม PID.....	48
3.3.2 การแปลงสัญญาณอนาล็อกเป็นดิจิตอล.....	49
3.3.3 การแปลงสัญญาณดิจิตอลเป็นอนาล็อก.....	50
3.3.4 การทำสเกล.....	51
3.3.5 การกำหนดช่วงเวลาของการแซมปลิง (interrupt).....	51
<b>บทที่ 4 ผลการทดลอง.....</b>	<b>53</b>
4.1 อุปกรณ์ที่ใช้ในการทดลอง.....	53
4.2 ลำดับขั้นเตรียมการทดลองควบคุม.....	53
4.3 ขั้นตอนการทดลองสำหรับควบคุมกระบวนการความดัน.....	54
4.3.1 การทดลองวงจรปรับระดับสัญญาณ (V/I) และ (I/V).....	54

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## สารบัญ (ต่อ)

	หน้า
4.3.2 การทดลองการควบคุมแบบ Proportional.....	56
4.3.3 การทดลองการควบคุมแบบ Proportional และ Integral.....	57
4.3.4 การทดลองการควบคุมแบบ Proportional และ Derivative.....	59
4.3.5 การควบคุมแบบ Proportional และ Integral และ Derivative.....	61
4.4 การปรับ PID แบบวิธี Trial and Error.....	63
4.5 สรุปผลการทดลอง.....	65
<b>บทที่ 5 สรุปผลการทดลองและข้อเสนอแนะ.....</b>	<b>66</b>
5.1 สรุปผลการทดลอง.....	66
5.2 ข้อเสนอแนะแนวทางการพัฒนา.....	66
<b>บรรณานุกรม.....</b>	<b>67</b>
<b>ภาคผนวก.....</b>	<b>68</b>

# สารบัญตาราง

ตารางที่	หน้า
2.1 ผลตอบสนองต่อสัญญาณสเต็ปของระบบอันดับหนึ่ง.....	18
2.2 แสดงผลการตอบสนองเมื่อค่าของอัตราความหน่วงที่แตกต่างกัน.....	21
2.3 แสดงค่าพารามิเตอร์ของตัวควบคุม PID ตามวิธีของ Ziegler-Nichols.....	25
2.4 แสดงค่าพารามิเตอร์ที่ใช้ใน process reaction ของตัวควบคุม PID โดยวิธีของ Ziegler-Nichols.....	27
4.1 แสดงความสัมพันธ์ระหว่างสัญญาณอินพุต 0–2.5 V กับสัญญาณเอาต์พุต 4–20 mA.....	54
4.2 แสดงความสัมพันธ์ระหว่างสัญญาณอินพุต 4–20 mA กับ สัญญาณเอาต์พุต 0–5 V.....	55



# สารบัญรูป

รูปที่	หน้า
2.1 การควบคุมระบบ.....	3
2.2 ฟังก์ชันถ่ายโอนระบบควบคุม.....	3
2.3 กราฟแสดงสัญญาณ (a) Unit – Step (b) Unit – Impulse (c) Unit – Ramp และ (c) Sine.....	5
2.4 (ก) แผนผังบล็อกของตัวควบคุมแบบ 2 ตำแหน่ง หรือแบบ On-Off.....	6
2.4 (ข) ช่วง Differential Gap.....	6
2.5 การควบคุมแบบสัดส่วน.....	7
2.6 แสดงความสัมพันธ์ระหว่างตัวแปรควบคุมกับตัวแปรจัดการ.....	8
2.7 การควบคุมแบบอินทิกรัล.....	9
2.8 แสดงผลตอบสนองการควบคุมแบบอินทิกรัลสัญญาณจากขั้นบันได.....	9
2.9 การควบคุมแบบเดริเวทีฟ.....	10
2.10 แสดงผลตอบสนองการควบคุมแบบสัญญาณเดริเวทีฟจากขั้นบันได.....	10
2.11 การตอบสนองของเครื่องควบคุมแบบพีไอต่อ การเปลี่ยนแปลงแบบขั้นบันไดของค่าความคลาดเคลื่อน.....	11
2.12 การควบคุมแบบพีดี.....	12
2.13 ผลการตอบสนองของการควบคุมแบบพีไอดีกับสัญญาณเข้าแบบขั้นบันได.....	12
2.14 ระบบควบคุมอัตโนมัติแบบวงเปิดหรือไม่มีการป้อนกลับ.....	13
2.15 บล็อกไดอะแกรมพื้นฐานของรูปควบคุมแบบส่งผ่าน.....	15
2.16 ผลตอบสนองเชิงเวลาของกระบวนการต่อสัญญาณอินพุตแบบสัญญาณระดับ.....	16
2.17 ผลตอบสนองต่อสัญญาณสแต็ปของระบบอันดับหนึ่ง.....	18
2.18 ผลตอบสนองของระบบอันดับหนึ่ง เมื่อมีการหน่วงเวลา (First-order Step Response with Time Delay).....	19
2.19 แสดงความสัมพันธ์ระหว่างตำแหน่งโพลกับการตอบสนองของกระบวนการ.....	23
2.20 แสดงการหาค่า ( $T_u$ ) จากผลการตอบสนองของระบบ ที่มีการแกว่งอย่างต่อเนื่องเมื่อมีการปรับค่าอัตราขยายเป็น ( $K_u$ ).....	25
2.21 แสดงผลการตอบสนองของระบบแบบรูปตัว S และการหาค่าพารามิเตอร์ของระบบโดยวิธี process reaction.....	26
2.22 ประเภทของความดัน.....	33

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# สารบัญรูป (ต่อ)

รูปที่	หน้า
3.1 โครงสร้างของกระบวนการ.....	36
3.2 ระบบที่ใช้ในการควบคุม.....	37
3.3 Block Diagram ของระบบที่ใช้ในการควบคุม.....	39
3.4 แสดงส่วนของ Plant ที่ประกอบด้วยอุปกรณ์ต่าง ๆ.....	40
3.5 แสดงส่วนของฮาร์ดแวร์ตัวควบคุมโดยรวม.....	41
3.6 วงจรแปลงสัญญาณแรงดันไฟฟ้าเป็นสัญญาณกระแสไฟฟ้า.....	41
3.7 วงจรแปลงสัญญาณสัญญาณกระแสไฟฟ้าเป็นแรงดันไฟฟ้า.....	44
3.8 วงจรที่ใช้ในการจ่ายไฟ.....	45
3.9 สวิตช์ต่างๆ ที่ใช้ในไมโครคอนโทรลเลอร์ ARM7.....	46
3.10 Schematic ของการต่อจอแสดงผล LCD กับไมโครคอนโทรลเลอร์ ADuC7024.....	47
3.11 จอแสดงผลที่ใช้ในการทดลอง.....	47
4.1 แสดงผลตอบสนองการควบคุมแบบ Proportional เมื่อค่า $K_p = 100$ .....	56
4.2 แสดงผลตอบสนองการควบคุมแบบ Proportional เมื่อค่า $K_p = 50$ .....	56
4.3 แสดงผลตอบสนองการควบคุมแบบ Proportional + Integral เมื่อค่า $K_p = 50, T_i = 10$ .....	57
4.4 แสดงผลตอบสนองการควบคุมแบบ Proportional + Integral เมื่อค่า $K_p = 50, T_i = 20$ .....	57
4.5 แสดงผลตอบสนองการควบคุมแบบ Proportional + Integral เมื่อค่า $K_p = 100, T_i = 20$ .....	58
4.6 แสดงผลตอบสนองการควบคุมแบบ Proportional + Integral เมื่อค่า $K_p = 100, T_i = 10$ .....	58
4.7 แสดงผลตอบสนองการควบคุมแบบ Proportional + Derivative เมื่อค่า $K_p = 100, T_d = 10$ .....	59
4.8 แสดงผลตอบสนองการควบคุมแบบ Proportional + Derivative เมื่อค่า $K_p = 100, T_d = 20$ .....	59
4.9 แสดงผลตอบสนองการควบคุมแบบ Proportional + Derivative เมื่อค่า $K_p = 50, T_d = 20$ .....	60

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## สารบัญรูป (ต่อ)

รูปที่	หน้า
4.10 แสดงผลตอบสนองการควบคุมแบบ Proportional + Derivative เมื่อค่า $K_p = 50$ , $T_d = 10$ .....	60
4.11 แสดงผลตอบสนองการควบคุมแบบ Proportional + Integral + Derivative เมื่อค่า $K_p = 50$ , $T_i = 10$ , $T_d = 10$ .....	61
4.12 แสดงผลตอบสนองการควบคุมแบบ Proportional + Integral + Derivative เมื่อค่า $K_p = 50$ , $T_i = 20$ , $T_d = 10$ .....	61
4.13 แสดงผลตอบสนองการควบคุมแบบ Proportional + Integral + Derivative เมื่อค่า $K_p = 100$ , $T_i = 20$ , $T_d = 10$ .....	62
4.14 สัญญาณควบคุมที่ได้จากตัวควบคุมแบบ Proportional + Integral ด้วยวิธี Trial and Error เมื่อค่า $K_p = 100$ , $T_i = 20$ .....	63
4.15 สัญญาณควบคุมที่ได้จากตัวควบคุมแบบ Proportional + Integral ด้วยวิธี Trial and Error เมื่อค่า $K_p = 50$ , $T_i = 20$ .....	63
4.16 สัญญาณควบคุมที่ได้จากตัวควบคุมแบบ Proportional + Integral ด้วยวิธี Trial and Error เมื่อค่า $K_p = 60$ , $T_i = 25$ .....	64
4.17 สัญญาณควบคุมที่ได้จากตัวควบคุมแบบ Proportional + Integral ด้วยวิธี Trial and Error เมื่อค่า $K_p = 70$ , $T_i = 20$ .....	64

# บทที่ 1

## บทนำ

### 1.1 ความสำคัญของปริญญานิพนธ์

ในปัจจุบันนี้งานในด้านอุตสาหกรรมมีมากมาย ในการทำงานในอุตสาหกรรมแล้วถ้าเราประหยัดบุคลากรในการทำงานได้ก็จะเป็นผลดี เราจึงได้คิดวิธีการสร้างความสะดวกมากขึ้นเพื่อตอบสนองความต้องการของมนุษย์ ระบบการวัดและควบคุมกระบวนการก็เป็นส่วนหนึ่งในงานด้านอุตสาหกรรมได้รับความสะดวก ซึ่งเราสามารถวัดค่าพารามิเตอร์ต่าง ๆ จากกระบวนการทางด้านอุตสาหกรรม แล้วทำการเปรียบเทียบกับค่าที่เราต้องการได้โดยการใช้ตัวควบคุมในการควบคุม และยังสามารถดูการแสดงผลของกระบวนการที่ควบคุมโดยผ่านทางหน้าจอคอมพิวเตอร์ อีกอย่างเรายังสามารถที่จะควบคุมกระบวนการนั้น ๆ ได้ให้เป็นไปตามความต้องการได้

โครงการนี้จึงได้นำการเสนอการออกแบบควบคุมกระบวนการ เพื่อที่จะให้กระบวนการเป็นไปตามต้องการ ซึ่งการควบคุมแบบนี้จึงสามารถนำไปประยุกต์ใช้งานในอุตสาหกรรมได้เป็นอย่างดีเนื่องจากมีต้นทุนการผลิตที่ต่ำและการสร้างก็ไม่ซับซ้อน

### 1.2 วัตถุประสงค์ของปริญญานิพนธ์

1. เพื่อศึกษาการทำงานและใช้งานไมโครคอนโทรลเลอร์ Arm7
2. เพื่อศึกษาการทำงานของกระบวนการต่าง ๆ รวมทั้งระบบการวัดของกระบวนการนั้น ๆ
3. เพื่อได้ศึกษาระบบควบคุมแบบป้อนกลับ
4. เพื่อศึกษากระบวนการและออกแบบควบคุมกระบวนการความดัน
5. เพื่อศึกษาการเขียนโปรแกรม Visual Basic 6.0 ในการแสดงผลทางคอมพิวเตอร์
6. เพื่อศึกษาวงจรอิเล็กทรอนิกส์เพื่อทำวงจร Signal Condition

### 1.3 ขอบเขตของปริญญานิพนธ์

1. สามารถวัดค่าจากกระบวนการได้
2. ได้ตัวควบคุมกระบวนการได้โดยใช้ไมโครคอนโทรลเลอร์ Arm7
3. สามารถแสดงผล และส่งค่าพารามิเตอร์ต่างๆผ่านคอมพิวเตอร์
4. ได้กระบวนการความดัน 1 กระบวนการ
5. ได้วงจร Signal Condition

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 1.4 ขั้นตอนการศึกษา

1. ตั้งสมมุติฐานว่าหากนำเอาไมโครคอนโทรลเลอร์ ARM7 ไปควบคุมกระบวนการที่มีอยู่สามารถที่จะให้ค่าความถูกต้องกับผลตอบสนองที่ออกมาจะสามารถควบคุมกระบวนการได้ใหม่
2. ศึกษาหลักการเขียนโปรแกรมของ Microcontroller ARM7
3. ทำการออกแบบกระบวนการที่จะใช้ในการควบคุมได้กระบวนการความดัน
4. นำไมโครคอนโทรลเลอร์ ARM7 ไปใช้ในการควบคุมกระบวนการความดัน
5. ทดลองและปรับค่าในการควบคุมกระบวนการ
6. สรุปผลการทดลองพร้อมทั้งข้อเสนอแนะ



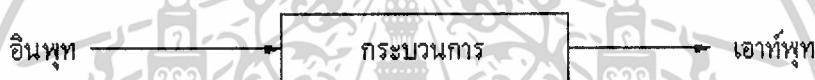
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทที่ 2

# ทฤษฎีและหลักการ

### 2.1 ระบบควบคุม

ระบบควบคุมคือ ส่วนประกอบหลายๆ ส่วนต่อเชื่อมกันขึ้นเป็นระบบเพื่อที่จะให้ผลของการตอบสนองตามที่เราต้องการ พื้นฐานของการวิเคราะห์ระบบมีพื้นฐานจากทฤษฎีระบบเชิงเส้น ซึ่งจะแสดงความสัมพันธ์ของอินพุตและเอาต์พุตในการตอบสนอง ส่วนประกอบที่จะควบคุมของกระบวนการ (Process) ที่เราต้องการที่จะควบคุมนั้นสามารถแทนด้วยบล็อก (block) ดังในรูปที่ 2.1 ส่วนอินพุตและเอาต์พุตของระบบมักจะแทนด้วยสัญญาณ โดยสัญญาณอินพุตจะเป็นส่วนสำคัญของผลลัพธ์หรือเอาต์พุต

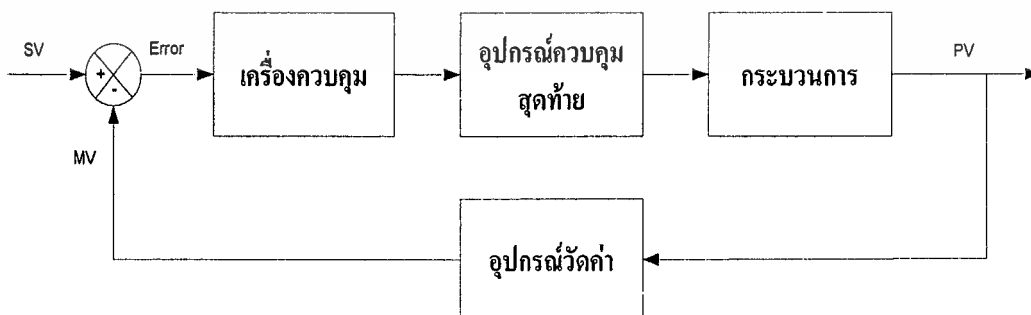


รูปที่ 2.1 การควบคุมระบบ

ระบบควบคุมสามารถแบ่งออกเป็นประเภทใหญ่ๆ ตามลักษณะการทำงานได้เป็น 2 แบบคือ

1. ระบบควบคุมแบบเปิด (Open Loop Control System)
2. ระบบควบคุมแบบปิดหรือระบบควบคุมแบบป้อนกลับ (Closed Loop or Feedback Control System)

### 2.2 องค์ประกอบของระบบควบคุม



รูปที่ 2.2 ฟังก์ชันของระบบควบคุม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**2.2.1 กระบวนการ (Process)** เป็นสิ่งที่เราต้องควบคุมไม่ว่ากระบวนการจะอยู่ในลักษณะรูปแบบใดโดยตัวแปรส่วนมากที่อยู่ในกระบวนการควบคุมหรือสิ่งที่จะทำการควบคุมนั้นก็คือ อุณหภูมิ ความดัน, อัตราการไหล, ระดับ เป็นต้น

**2.2.2 อุปกรณ์การวัด (Measuring devices)** เป็นอุปกรณ์ที่จะนำมาใช้เพื่อวัดค่าต่าง ๆ ของกระบวนการ (Process) เพื่อให้รับรู้ถึงสภาพของกระบวนการ (Process) โดยมีวัตถุประสงค์เพื่อการปรับแต่งการควบคุม (Control) เพื่อให้กระบวนการ (Process) เป็นไปตามที่ต้องการ (Set Point) ในการวัดและการควบคุมเพื่อที่จะให้มีคุณสมบัติตามที่ต้องการ โดยมีตัวแปรที่เข้ามาเกี่ยวข้องกับต้องการวัดค่าหลายตัวแปร เช่น อุณหภูมิ (Temperature) ความดัน (Pressure) ระดับ (Level) และมีปริมาณการไหล (Flow Rate) เป็นต้น

**2.2.3 เครื่องควบคุม (Controller)** เป็นอุปกรณ์ที่ทำการควบคุมกระบวนการในการควบคุมสามารถกระทำได้ด้วยการควบคุมแบบ Manual และ Automatic แต่ในการควบคุมแบบ Manual นั้น การตัดสินใจจะสั่งการโดยมนุษย์จะเป็นผู้กระทำ ส่วนในการควบคุมแบบอัตโนมัติในการตัดสินใจสั่งการจะกระทำด้วยอุปกรณ์เครื่องควบคุม (Controller)

**2.2.4 อุปกรณ์ควบคุมสุดท้าย (Final Control Element)** จะเป็นอุปกรณ์ที่ทำการรับค่ามาจากเครื่องควบคุม เพื่อจะทำการปฏิบัติให้เกิดผลกระทบต่อกระบวนการ ตามที่เครื่องควบคุมได้สั่งการได้แก่ Control Valve เป็นต้น

## 2.3 สัญญาณอินพุต (Input signal)

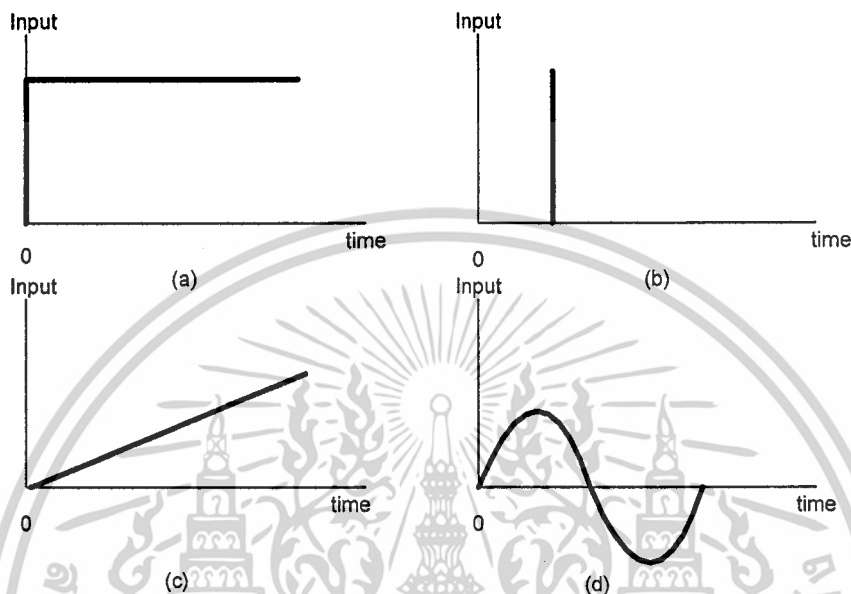
ส่วนอินพุตนั้นจะมีอยู่หลายรูปแบบ โดยทั่วไปเราจะสามารถเห็นอินพุตเหล่านี้อยู่ในรูปอินพุตแบบขั้นบันได (Unit – Step Input) อินพุตแบบนี้มีลักษณะเป็นการเปลี่ยนแปลงค่าอินพุตอย่างทันทีทันใดโดยตัวอย่าง เช่น เมื่อเปิดไฟฟ้าเข้าสู่วงจรไฟฟ้าทันทีที่ความต่างศักย์ในวงจรจะเพิ่มจาก 0 เป็นค่าที่เราให้กับวงจรทันที

**อินพุตแบบการดล (Unit - Impulse Input)** อินพุตแบบนี้จะมีลักษณะเหมือนกับการให้ค่าอินพุตค่าหนึ่งในช่วงเวลาสั้น ๆ เช่น การที่เราใช้ไม้ตีลูกบอลแรงกระทำที่เกิดกับลูกบอลจะมีค่าเฉพาะในช่วงเวลาสั้น ๆ เมื่อไม้สัมผัสลูกบอลเท่านั้น

**อินพุตแบบแรม (Unit – Ramp Input)** จะเป็นอินพุตที่ค่อย ๆ มีค่าเพิ่มขึ้นทีละนิด โดยที่อัตราการเพิ่มของอินพุตจะคงที่ เช่น น้ำที่ไหลเข้าถังด้วยอัตราการไหลคงที่น้ำในถังของน้ำจะค่อย ๆ เพิ่มขึ้น ถ้าน้ำในถังของน้ำเป็นอินพุตจะทำให้อินพุตดังกล่าวมีลักษณะเหมือนแรมอินพุต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อินพุตแบบไซน์ (Sine Input) จะเป็นอินพุตที่มีสมการของอินพุตเป็นฟังก์ชัน  $\sin(\omega t)$  โดยจะเป็นความถี่เชิงมุม  $\omega$  และ  $t$  เป็นเวลา ตัวอย่างของสัญญาณประเภทนี้ก็ เช่น แรงที่กระทำให้เกิดการหมุนที่ไม่สมดุลของเพลา เป็นต้น



รูปที่ 2.3 กราฟแสดงสัญญาณ (a) Unit – Step (b) Unit – Impulse (c) Unit – Ramp และ (d) Sine

## 2.4 เครื่องควบคุม

ลักษณะการทำงานของเครื่องควบคุม (Controller Action) ในการเลือกลักษณะการทำงานของเครื่องควบคุมมีความสำคัญเป็นอย่างมาก ถ้าเลือกผิดกระบวนการจะควบคุมไม่ได้นอกจากนั้น ลักษณะการทำงานของเครื่องควบคุมยังมีความเกี่ยวข้องกับลักษณะการทำงานของวาล์วควบคุม ในลักษณะการทำงานของวาล์วควบคุมนั้น จะเป็นตัวกำหนดอัตราการขยายเชิงสถิติของวาล์วควบคุม ดังนั้นในการเลือกลักษณะการทำงานของอุปกรณ์จะเป็นตัวกำหนดการเลือกลักษณะการทำงานของเครื่องควบคุม

### ประเภทของเครื่องควบคุม (Controller Type)

เครื่องควบคุมจะทำหน้าที่คำนวณที่หาค่าสำหรับใช้ปรับเปลี่ยนตัวแปรปรับได้ จากสมการควบคุมโดยใช้ค่าความแตกต่างระหว่างค่ากระบวนการกับค่าเป้าหมาย

#### 2.4.1 เครื่องควบคุมแบบเปิด-ปิด (On-Off Control)

ในระบบควบคุมแบบ 2 ตำแหน่งการทำงานจะทำในตำแหน่งที่คงที่เพียง 2 ตำแหน่ง แต่ในบางครั้งจะมีการเรียกการควบคุมแบบ ON หรือ OFF การควบคุมแบบ 2 ตำแหน่งหรือ แบบ

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ON - OFF นี้ จะเป็นการควบคุมแบบง่าย ๆ และราคาไม่แพง จึงเป็นที่นิยมใช้ในงานอุตสาหกรรม และในกรณีที่มีผลต่อการแกว่ง (Oscillate) นั้นเป็นที่ยอมรับได้

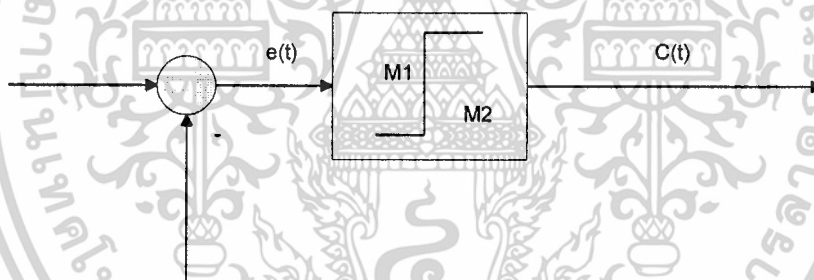
โดยกำหนดให้สัญญาณเอาต์พุตที่ตัวควบคุมเป็น  $c(t)$  และสัญญาณค่าความคลาดเคลื่อนเป็น  $e(t)$  ฉะนั้นการควบคุมแบบ 2 ตำแหน่งนี้ สัญญาณ  $c(t)$  จะมีค่าอยู่เพียงค่าสูงสุดและค่าต่ำสุดเท่านั้น โดยจะขึ้นอยู่กับความคลาดเคลื่อนที่มีค่าเป็น + หรือ - นั่นคือ

$$c(t) = M_1 \quad \text{สำหรับ } e(t) > 0$$

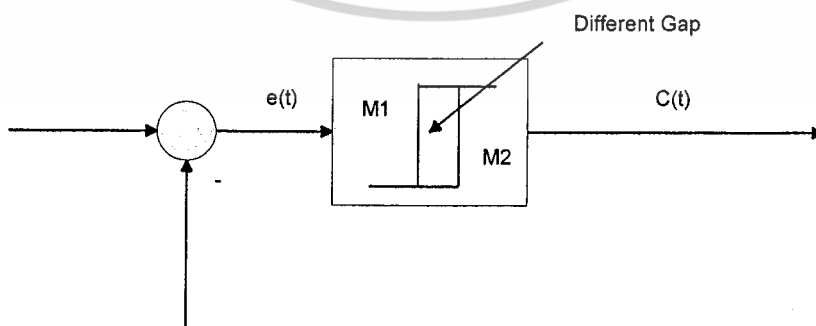
$$c(t) = M_2 \quad \text{สำหรับ } e(t) < 0$$

โดยที่  $M_1$  และ  $M_2$  เป็นค่าคงที่

รูปที่ 2.4 (ก) แสดงถึง Block Diagram ของตัวควบคุมแบบ 2 ตำแหน่ง และสำหรับช่วงที่ค่าสัญญาณของความคลาดเคลื่อนเปลี่ยนแปลงไปก่อนเกิดการเปลี่ยนตำแหน่ง (Switching) ของการควบคุมนั้นเรียกว่า Differential Gap ดังแสดงในรูปที่ 2.4 (ข) ช่วง Differential Gap นี้บางครั้งเป็นการทำให้เกิดขึ้นเพื่อป้องกันการ ON - OFF บ่อยครั้งเกินไป



รูปที่ 2.4 (ก) แผนผังบล็อกของตัวควบคุมแบบ 2 ตำแหน่ง หรือแบบ On-Off

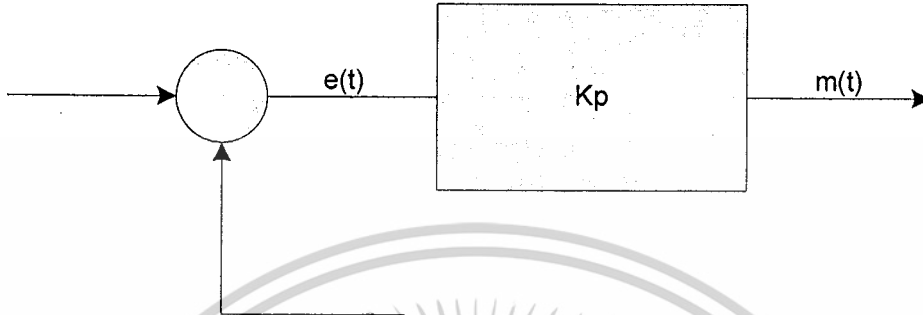


รูปที่ 2.4 (ข) ช่วง Differential Gap

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 2.4.2 กิริยาควบคุมแบบพี (P Action)

เป็นปฏิบัติการควบคุมซึ่งค่อนข้างจะสม่ำเสมอ และการเป็นเชิงเส้นระหว่างการเปลี่ยนแปลงของอินพุตและเอาต์พุต สามารถแสดงความสัมพันธ์ในรูปแบบของ Block Diagram ดังรูปที่ 2.5



รูปที่ 2.5 การควบคุมแบบสัดส่วน

เครื่องควบคุมแบบพี (P Controller) เป็นเครื่องควบคุมแบบง่ายที่สุด มีสมการที่ใช้สำหรับเครื่องควบคุมประเภทนี้คือ

$$m(t) = K_p e(t) + \bar{m} \quad \dots 2.1$$

หรือ

$$\frac{M(s)}{E(s)} = K_p \quad \dots 2.2$$

โดยที่

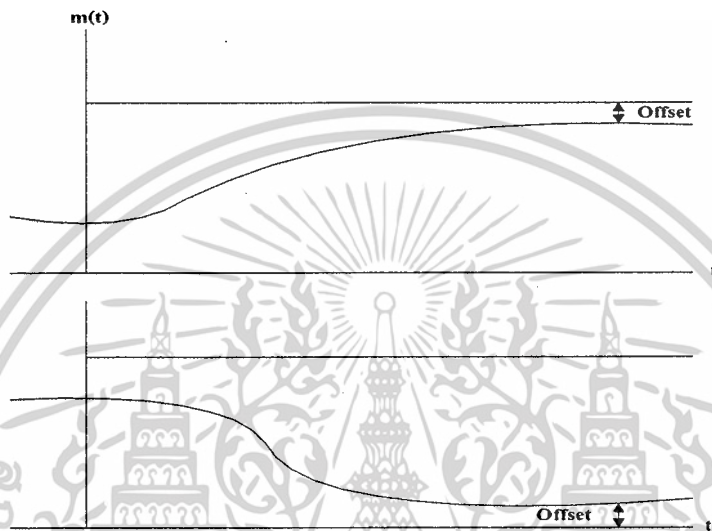
- $m(t)$  = ตัวแปรขาออกเครื่องควบคุม
- $\bar{m}$  = ค่าตัวแปรขาออกเครื่องควบคุมที่สภาวะคงตัว
- $K_p$  = ค่าความไวการตอบสนองของเครื่องควบคุมแบบพี
- $e(t)$  = ค่าความคลาดเคลื่อน

หลักการทำงานของเครื่องควบคุมแบบพีนั้น เครื่องควบคุมแบบพีนี้คำนวณหาค่าสัญญาณของขาออกได้ โดยคำนวณค่าสัญญาณขาออกเป็นสัดส่วนกับขนาดของค่าความคลาดเคลื่อนของกระบวนการ โดยสัญญาณขาออกของเครื่องควบคุมจะมีค่าสูงหรือต่ำขึ้นอยู่กับขนาดของค่าความคลาดเคลื่อนและอัตราขยายเชิงสถิติของเครื่องควบคุม  $K_p$

เครื่องควบคุมแบบพีจะมีข้อดีคือเป็นเครื่องควบคุมแบบง่ายที่สุด เนื่องจากมีพารามิเตอร์ที่ต้องกำหนดค่าเพียงค่าเดียวคือค่า  $K_p$  แต่มีข้อเสียคือในภายใต้การควบคุมตัวแปรควบคุมยังอาจมีค่าแตกต่างจากค่าเป้าหมายได้ โดยความแตกต่างนี้เรียกว่าค่าผลต่าง (Offset)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การเกิด Offset เป็นคุณลักษณะของระบบควบคุมแบบ Proportional เนื่องจากการทำงานของระบบควบคุมแบบ Proportional นั้น ไม่สามารถควบคุมระบบที่มี load ที่เปลี่ยนแปลงไปได้ดีเท่าที่ควร และในกรณีที่มี load คงที่แต่การเปลี่ยนแปลงค่าของระดับ set point ที่ควบคุมไป ก็จะมีการเกิดค่าของ offset ขึ้น โดยที่ offset ก็คือ ค่าความแตกต่างของอินพุตและเอาต์พุตที่สภาวะคงที่เมื่อค่าเป้าหมายคงที่นั่นเองแสดงดังรูปที่ 2.6



รูปที่ 2.6 แสดงความสัมพันธ์ระหว่างตัวแปรควบคุมกับตัวแปรจัดการ

#### 2.4.3 กริยาควบคุมแบบอินทิกรัล (I Action)

เป็นการควบคุมกระบวนการซึ่งจะเอาค่าของเอาต์พุตที่เป็นสัดส่วนโดยตรงกับค่าอินทิกรัลเชิงเวลาของอินพุต โดยจะมีความสัมพันธ์ระหว่างค่าเอาต์พุตของตัวควบคุม  $m(t)$  และค่าความคลาดเคลื่อน  $e(t)$  ดังนี้

$$\frac{dm(t)}{dt} = K_i e(t) \quad \dots 2.3$$

หรือ 
$$m(t) = K_i \int_0^t e(t) dt \quad \dots 2.4$$

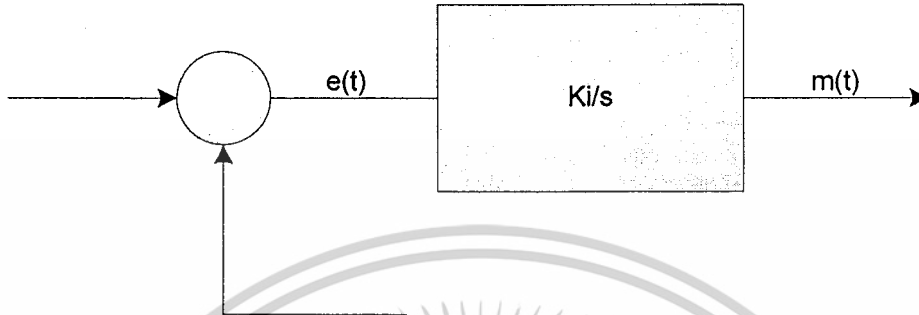
โดยที่  $K_i$  เป็นค่าคงที่ที่สามารถปรับค่าได้

ฟังก์ชันถ่ายโอน (Transfer Function) ของตัวควบคุมแบบอินทิกรัลคือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

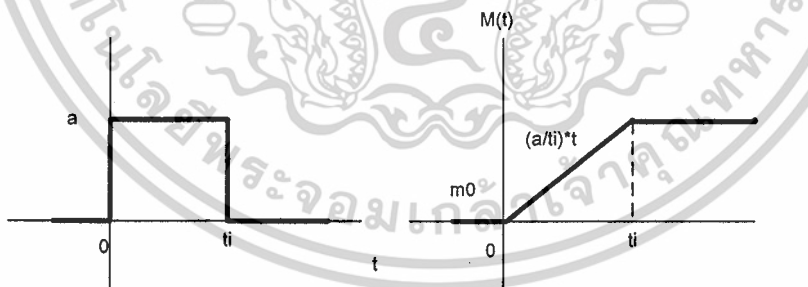
$$\frac{M(s)}{E(s)} = \frac{K_i}{s} \quad \dots\dots 2.5$$

โดยสามารถแสดงในรูปของ Block Diagram ได้ดังรูปที่ 2.7



รูปที่ 2.7 การควบคุมแบบอินทิกรัล

ในการควบคุมแบบอินทิกรัลนั้นค่าเอาต์พุตที่ตัวควบคุม  $m(t)$  จะเปลี่ยนแปลงตามค่าความผิดพลาด  $e(t)$  ดังนั้นถ้าเกิดความผิดพลาดซึ่งได้เกิดขึ้นทำให้ระบบได้ ค่าที่ผิดไปจากค่าที่ต้องการแล้ว อุปกรณ์ควบคุมจะจัดการกับความผิดพลาดโดยเร็ว (โดยลดให้ค่า Error นี้หมดไป) เมื่อตัวแปรควบคุมอยู่ที่ค่าเป้าหมายแล้วอุปกรณ์ควบคุมตัวสุดท้าย (Final element control) จะยังไม่ทำงานซึ่งแสดงให้เห็นว่าระบบอยู่ในสภาวะคงที่นั่นเอง ดังนั้นการควบคุมแบบอินทิกรัลนี้จะไม่ทำให้เกิดค่าของ Offset ขึ้นมาแสดงดังรูปที่ 2.8



รูปที่ 2.8 แสดงผลตอบสนองการควบคุมแบบอินทิกรัลสัญญาณจากขั้นบันได

#### 2.4.4 กิริยาควบคุมแบบเดริเวทีฟ (D Action)

เป็นการควบคุมที่ค่าเอาต์พุตเป็นสัดส่วนกับอัตราการเปลี่ยนแปลงของอินพุต โดยมีสมการความสัมพันธ์ดังนี้

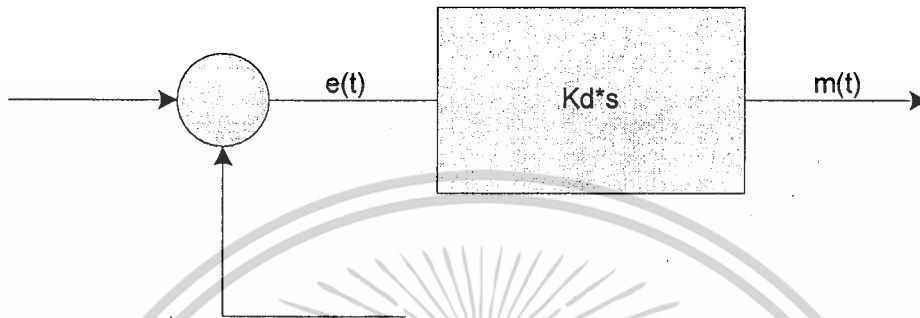
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$m(t) = K_d \times \frac{de(t)}{dt} \quad \dots 2.6$$

หรือ

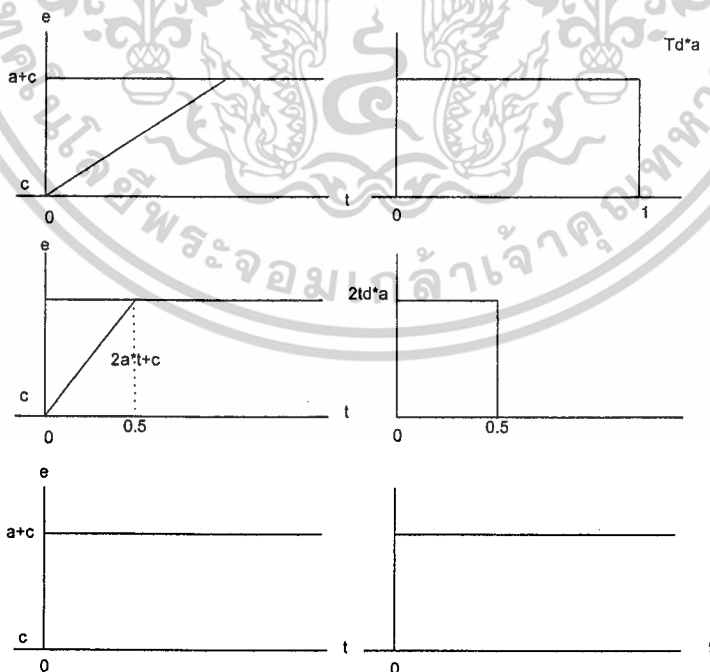
$$\frac{M(s)}{E(s)} = K_d s \quad \dots 2.7$$

โดยที่  $K_d$  เป็นค่าคงที่ที่สามารถปรับค่าได้



รูปที่ 2.9 การควบคุมแบบเดริเวทีฟ

Block Diagram ของตัวควบคุมแบบเดริเวทีฟแสดงดังรูปที่ 2.9 จะมีการเพิ่มของการควบคุมแบบ Derivative ไปในเครื่องควบคุม จะเป็นการบวกม้วนนำในเครื่องควบคุมเพื่อชดเชยมุมตามในฟังก์ชันถ่ายโอนซึ่งขบวนการส่วนใหญ่จะมีค่ามุมเป็นมุมตาม



รูปที่ 2.10 แสดงผลตอบสนองของการควบคุมแบบสัญญาณเดริเวทีฟจากขั้นบันได

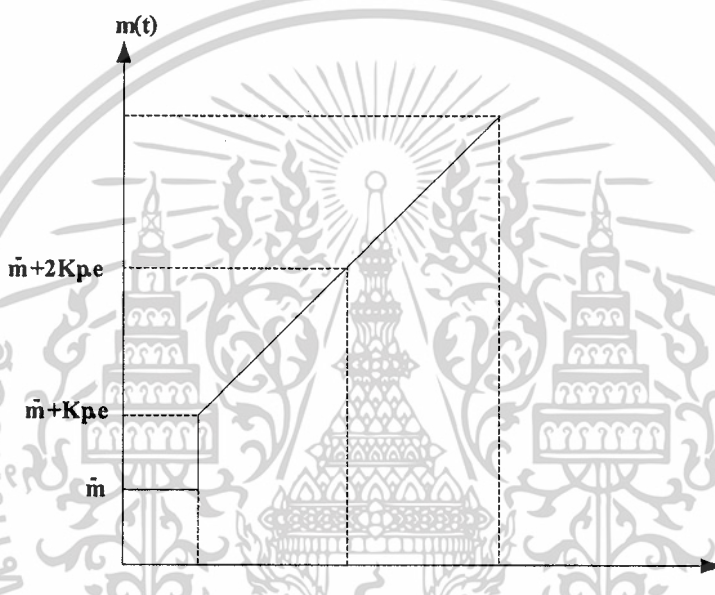
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 2.4.5 เครื่องควบคุมแบบพีไอ (PI Controller)

เครื่องควบคุมแบบพีไอ (PI Controller) นี้เป็นเครื่องควบคุมที่มีการใช้พจน์ในการตัดสินใจเพิ่มขึ้นจากเครื่องควบคุมแบบพีอีกหนึ่งพจน์ ในพจน์ดังกล่าวก็คือพจน์ของการอินทิกรัลค่าความคลาดเคลื่อนตลอดช่วงเวลาของการควบคุม สมการสำหรับเครื่องควบคุมประเภทนี้คือ

$$m(t) = K_p e(t) + \frac{K_p}{\tau_I} \int e(t) dt + \bar{m} \quad \dots 2.8$$

โดยที่  $\tau_I$  = ค่าเวลาคงที่อินทิกรัลในหน่วยของเวลา (วินาที)



รูปที่ 2.11 การตอบสนองของเครื่องควบคุมแบบพีไอต่อการเปลี่ยนแปลงแบบขั้นบันไดของค่าความคลาดเคลื่อน

จากสมการแสดงให้เห็นว่าเครื่องควบคุมแบบ PI นี้ ไม่เพียงแต่การพิจารณาค่าความคลาดเคลื่อนที่เกิดขึ้น ณ เวลานั้นเท่านั้นแต่ยังมีการนำเอาค่าประวัติศาสตร์ของการเปลี่ยนแปลงของความคลาดเคลื่อนจากอดีตถึงปัจจุบันมาพิจารณาด้วย โดยพจน์ของการอินทิกรัลแบบนี้มีบทบาทในการเปลี่ยนแปลงค่าสัญญาณขาออกที่ค่าความคลาดเคลื่อนยังไม่เป็นศูนย์

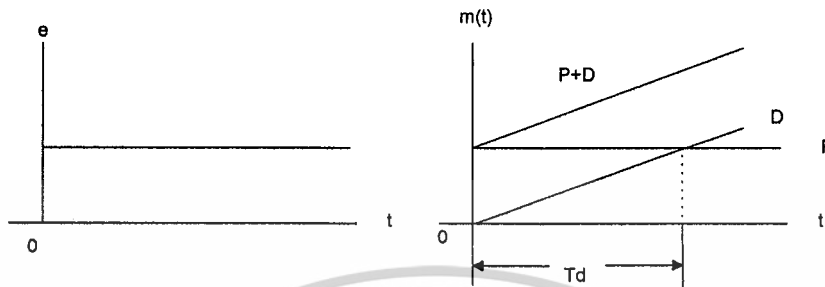
### 2.4.6 เครื่องควบคุมแบบพีดี (PD Controller)

เครื่องควบคุมแบบพีดีนี้ เป็นเครื่องควบคุมที่ใช้กับกระบวนการที่ไม่มีสัญญาณรบกวนและการควบคุมยอมให้มีผลต่าง (Offset) ได้ โดยคาดว่าจะได้ประโยชน์จากพจน์อัตราการเปลี่ยนแปลงของค่าความคลาดเคลื่อน สมการของเครื่องควบคุมแบบพีดีคือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$m(t) = K_p e(t) + K_p \tau_D \frac{de(t)}{dt} + \bar{m} \quad \dots 2.9$$

โดยที่  $\tau_D$  = ค่าเวลาคงที่อนุพันธ์ในหน่วยของเวลา (วินาที)



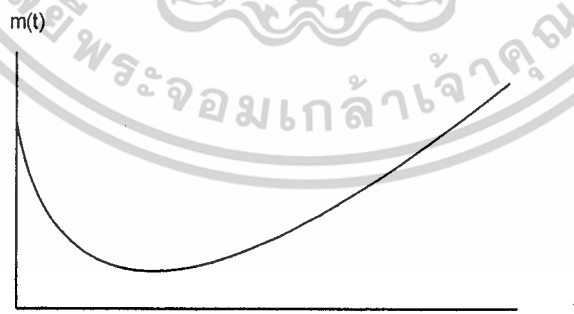
รูปที่ 2.12 การควบคุมแบบพีดี

ข้อดีของการควบคุมแบบพีดีคือ เมื่อมีสัญญาณเข้าเป็นเชิงเส้น (Ramp) จะมีผลตอบสนองทางเวลาได้เปรียบกว่าการควบคุมแบบสัดส่วนเพียงอย่างเดียว

#### 2.4.7 เครื่องควบคุมแบบพีไอดี (PID Controller)

เครื่องควบคุมแบบพีไอดี (PID Controller) นี้ เป็นเครื่องควบคุมที่ใช้พจน์ในการตัดสินใจที่เพิ่มขึ้นอีกหนึ่งพจน์จากเครื่องควบคุมแบบพีไอ โดยพจน์ดังกล่าวคือพจน์ของอัตราการเปลี่ยนแปลงของค่าความคลาดเคลื่อน ณ เวลาที่พิจารณาสมการสำหรับเครื่องควบคุมประเภทนี้คือ

$$m(t) = K_p e(t) + \frac{K_p}{\tau_I} \int e(t) dt + K_p \tau_D \frac{de(t)}{dt} + \bar{m} \quad \dots 2.10$$



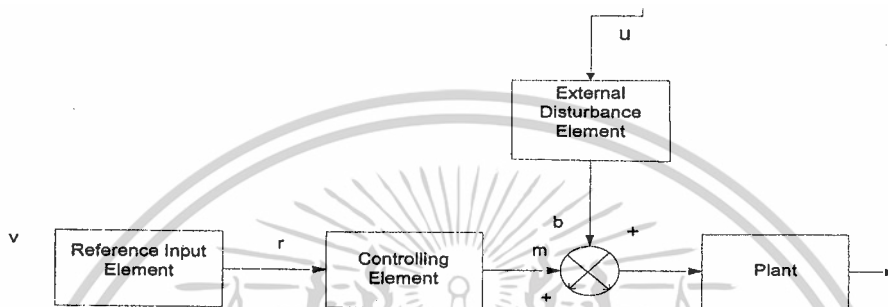
รูปที่ 2.13 ผลการตอบสนองของการควบคุมแบบพีไอดีกับสัญญาณเข้าแบบขั้นบันได

เครื่องควบคุมแบบพีไอดีนี้เหมาะสำหรับใช้กับกระบวนการที่มีการตอบสนองช้า ในส่วนกระบวนการที่มีการตอบสนองเร็ว นั้นมักจะมาพร้อมกับสัญญาณรบกวนและสัญญาณรบกวนทำให้เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ความชันของค่าความคลาดเคลื่อนมีค่าสูง ซึ่งจะส่งผลให้เกิดการปรับสัญญาณเครื่องควบคุมทั้ง ๆ ที่มีค่าความคลาดเคลื่อนของสัญญาณกระบวนการไม่ได้มีความชันสูงอย่างนั้น

## 2.5 กระบวนการควบคุม

### 2.5.1 ระบบควบคุมอัตโนมัติแบบวงเปิดหรือไม่มีการป้อนกลับ



รูปที่ 2.14 ระบบควบคุมอัตโนมัติแบบวงเปิดหรือไม่มีการป้อนกลับ

ระบบควบคุมอัตโนมัติประเภทนี้ดังแสดงในภาพที่ 2.14 จะเป็นเรื่องของ การควบคุมที่ไม่มี การเปรียบเทียบเอาต์พุตของระบบหรือ Control Variable กับ Reference Variable ในระบบกล่าวคือ Error หรือ Actuating Signal ที่มีค่าคงที่ตลอดเวลา และโดยทั่วไปของระบบเรื่องของการควบคุม จะทำงานที่อัตราคงที่นั่นคือข้อสังเกตที่สำคัญของระบบควบคุมประเภทนี้ก็คือค่าความคลาดเคลื่อน ของเอาต์พุตของระบบ เพราะในทางปฏิบัติระบบจะถูกรบกวนจากภายนอกเสมอทำให้ส่วนที่เป็น ตัวรับค่าตัวแปรกระบวนการไม่สามารถทำงานได้เต็มที่ Controller ส่งมาแต่จะเอาการรบกวนเข้า มาและไม่มีการแก้ไขกลับไป (Feed Back) ที่ส่วนที่เป็นตัวควบคุมอีกแต่อย่างใด

ตัวอย่างของระบบควบคุมอัตโนมัติประเภทนี้ได้แก่ การทำงานของเครื่องซักผ้าซึ่งทำงาน เป็นจังหวะ ๆ คือเติมน้ำ ซัก รีดน้ำ แล้วเติมน้ำใหม่ รีดน้ำหรือ ฯลฯ และจะหยุดเครื่องตามแต่ ผู้ผลิตหรือผู้ใช้ตั้งค่าเอาไว้หรือโปรแกรมเอาไว้ เครื่องก็จะไม่ปรับตัวเองให้ทำงานให้เหมาะสมกับ ความสะอาดหรือความสกปรกของผ้าในเครื่องเลยและอีกตัวอย่างหนึ่งได้แก่ สัญญาณไฟจราจรที่ เปลี่ยนไฟเขียวเป็นไฟแดงตามทางแยกผลิตภัณฑ์ข้างละ 1-2 นาทีนั้น แล้วแต่การตั้งเครื่องการทำงาน ของระบบสัญญาณไฟจราจรนี้นั้นและจะถือว่าเป็นแบบมีการป้อนกลับได้ก็ต่อเมื่อมีการนำเอาความ แออัดของรถยนต์หรือยานพาหนะบนท้องถนนแต่ละแยกไปบังคับการทำงานของระบบจึงจะถือว่า เป็นระบบวงจรปิดหรือมีการป้อนกลับ จึงจะสังเกตได้ว่าค่าความแม่นยำของ Controller Variable หรือเอาต์พุตของระบบเทียบกับ Reference Variable ของระบบในระบบควบคุมอัตโนมัติ การตั้งค่า การปรับ การคาดการณ์ล่วงหน้า (Calibration and Programming) เป็นสิ่งที่สำคัญ มิฉะนั้นระบบจะ เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นิยมนำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ทำงานอย่างผิดพลาดและไร้ประโยชน์ ฉะนั้นระบบควบคุมแบบนี้จะทำงาน โดยอาศัยจังหวะหรือเวลาเป็นหลัก ทั้งนี้เพราะเป็นตัวแปรที่มีอยู่ (Available) และมีค่าคงที่ (Consistent) เสมอในระบบควบคุมอัตโนมัติแบบวงจรเปิดหรือแบบที่ไม่มีการป้อนกลับ และถ้าหากนับผู้ใช้ระบบร่วมกับตัวเครื่องด้วย ในระบบใหม่ที่ได้จะได้เป็นระบบควบคุมประเภทวงปิดกล่าวคือในการปิดหรือการป้อนกลับเกิดที่ตัวผู้ใช้เครื่อง เช่น ระบบไฟสัญญาณจราจรที่ได้กล่าวมาโดยจะมีพนักงานจราจรเป็นผู้บังคับสัญญาณไฟแทนการผลัดเปลี่ยนกันเองตามจังหวะเวลา (Timer) ที่ตั้งไว้

### 2.5.2 การควบคุมแบบคาสเคด

รูปควบคุมแบบนี้ดัดแปลงมาจากรูปควบคุมแบบป้อนกลับ โดยมีการเพิ่มรูปควบคุมแบบป้อนกลับเข้าไปในกระบวนการเพื่อจะช่วยให้ตรวจสอบของสภาวะกระบวนการรอง (Intermediate Controller Variable) ซึ่งใช้เป็นค่าอ้างอิงหรือค่าที่ตั้งไว้ (Set Point) สำหรับรูปป้อนกลับหลัก

รูปควบคุมแบบคาสเคดเหมาะกับการประยุกต์ใช้ในการควบคุมกระบวนการ ที่มีผลการตอบสนองที่ช้ามาก ทั้งนี้เนื่องจากกระบวนการประเภทนี้มักเกิดค่าความคลาดเคลื่อนขึ้นเรื่อยๆ ผ่านไปนานมาก และหากมีสัญญาณรบกวนภายนอกเข้ามากระทำกับกระบวนการทำให้สภาพการทำงานเปลี่ยนไป แต่ด้วยเหตุที่กระบวนการมีผลการตอบสนองช้านี้จึงทำให้รูปควบคุมจำเป็นต้องรอนานมากเช่นกันที่จะนำผลที่ได้ไปสร้างสัญญาณปรับแต่ง (Manipulated Variable) เพื่อจะปรับให้สภาพการทำงานของกระบวนการต่อไป ดังนั้นจะเห็นในกรณีเช่นนี้รูปควบคุมป้อนกลับเพียงรูปเดียวไม่สามารถตอบสนองการทำงานได้อย่างมีประสิทธิภาพ แนวทางการออกแบบนั้นจึงนิยมใช้รูปควบคุมแบบคาสเคดมากกว่า เพื่อที่จะให้ผู้ควบคุมกระบวนการนี้สามารถตรวจสอบสภาวะของกระบวนการรอง แล้วนำมาใช้เป็นข้อมูลในการปรับแต่งสภาพการทำงานของกระบวนการหลักให้ มีผลตอบสนองต่อสัญญาณรบกวนที่มากระทำได้อย่างมีประสิทธิภาพและรวดเร็วขึ้น

### 2.5.3 การควบคุมแบบสัดส่วน

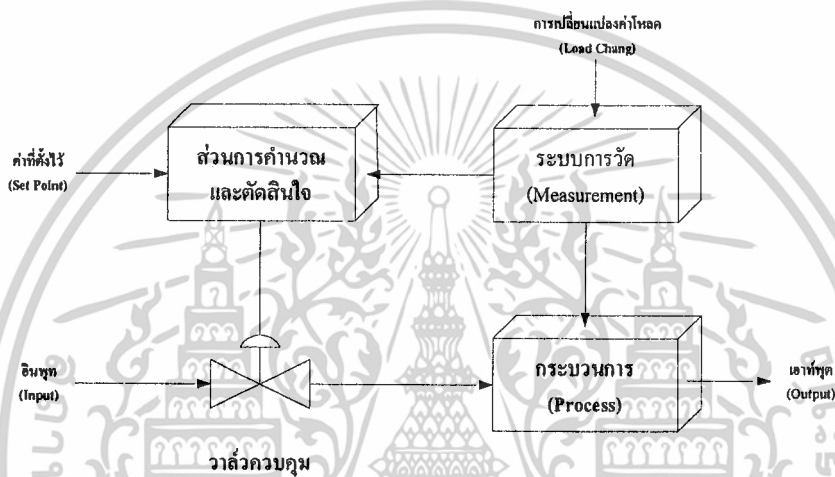
โครงสร้างทางฮาร์ดแวร์ระหว่างรูปควบคุมแบบคาสเคดกับรูปควบคุมแบบสัดส่วนนั้น จะมีความคล้ายคลึงกัน แต่เมื่อพิจารณาของการควบคุมแบบพื้นฐานแล้วจะพบว่าทั้งสองรูปแบบนั้นมีความแตกต่างกันมาก

โดยทั่วไปหลักการควบคุมกระบวนการ โดยใช้รูปควบคุมแบบสัดส่วนนี้ จะนิยมใช้ในกระบวนการที่ต้องการนำเอาปริมาณทางฟิสิกส์สองชุดหรือมากกว่ามาผสมกันเพื่อที่จะทำการรักษาสถานะของส่วนผสมที่ได้นั้นไว้ ซึ่งในแนวทางปฏิบัติสามารถทำได้โดยใช้อุปกรณ์ควบคุมการไหล (Conventional Flow Controller) โดยชุดแรกกับปริมาณตัวแรกและนำมาควบคุมปริมาณอีกตัวหนึ่งด้วยอุปกรณ์ควบคุมอัตราส่วน (Ratio Controller) ภายใต้อัตราส่วนที่เหมาะสมและอัตราส่วนที่ได้จากอุปกรณ์ควบคุมทั้งสองตัวนี้จะนำมาใช้เป็นข้อมูลในการปรับแต่งสภาพกระบวนการ

เอกลักรีนเป็นเอกลักรีนที่สงวนไว้สำหรับการใช้งานเฉพาะเท่านั้น เมื่อผู้ดูแลระบบใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 2.5.4 การควบคุมแบบป้อนไปข้างหน้า

บล็อกไดอะแกรมของรูปควบคุมแบบป้อนไปข้างหน้าแสดงดังรูปที่ 2.15 ในที่นี้ระบบของการวัด ซึ่งภายในจะประกอบด้วยเซนเซอร์เป็นอุปกรณ์ที่สำคัญมากที่จะทำหน้าที่ในการตรวจสอบการเปลี่ยนแปลงค่าไหลต่าง ๆ ของกระบวนการ หรือสัญญาณรบกวนจากภายนอกที่เข้ามากระทำกับกระบวนการ จากนั้นสัญญาณการตรวจวัดที่ได้จะส่งให้กับหน่วยประมวลผลของระบบเพื่อทำการคำนวณประมวลผลและตัดสินใจสร้างสัญญาณควบคุมเอาต์พุตที่เหมาะสมต่อการปรับสภาพในการทำงานของกระบวนการที่เปลี่ยนแปลงไป รวมทั้งสอดคล้องกับค่าที่ตั้งไว้ด้วย



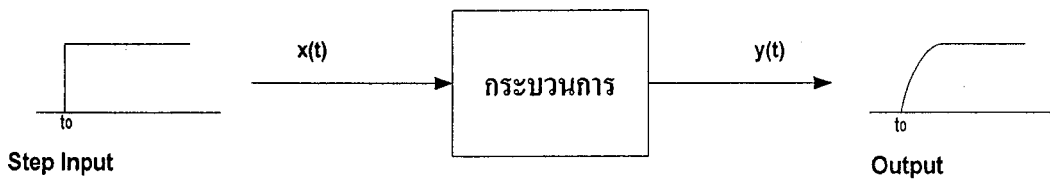
รูปที่ 2.15 บล็อกไดอะแกรมพื้นฐานของรูปควบคุมแบบส่งผ่าน

## 2.6 ผลตอบสนองของระบบต่อสัญญาณอินพุตชนิดต่างๆ

ในที่นี้จะพิจารณาผลตอบสนองของระบบอันดับหนึ่งและอันดับสองดังนี้

### 2.6.1 ผลตอบสนองต่อระบบอันดับหนึ่ง (Response of First - order System)

ผลการตอบสนองเชิงเวลาของกระบวนการอันดับหนึ่ง ก็จะเป็นอีกหนึ่งองค์ประกอบหนึ่ง ที่จำเป็นต้องคำนึงถึงเมื่อต้องการผลตอบสนองของรูปควบคุม ในการศึกษาผลตอบสนองเชิงเวลาของกระบวนการอันดับหนึ่งนี้ สามารถกระทำได้โดยการป้อนสัญญาณอินพุตแบบสัญญาณระดับ (Step Input) แล้วทำการสังเกตผลการตอบสนองของกระบวนการทางด้านเอาต์พุตดังตามรูปที่ 2.16 ลักษณะของสัญญาณระดับนั้นเป็นสัญญาณที่มีสภาวะคงที่ก่อนการเปลี่ยนแปลง และหลังจากนั้นที่เวลา  $t = t_0$  สัญญาณจะเปลี่ยนไปที่สภาวะใหม่อย่างทันทีทันใด ในการศึกษาผลการตอบสนองของกระบวนการต่อสัญญาณอินพุตแบบสัญญาณระดับนี้ จึงเป็นการสังเกตการณ์การเปลี่ยนแปลงของสัญญาณเอาต์พุตของกระบวนการที่ค่อย ๆ ปรับเข้าสู่ค่าสูงสุด ณ สภาวะใหม่นั้นเอง



**รูปที่ 2.16** ผลตอบสนองเชิงเวลาของกระบวนการต่อสัญญาณอินพุตแบบสัญญาณระดับ  
พิจารณาแบบอันดับหนึ่งต่อไปนี้

$$a_1 \frac{dy(t)}{dt} + c_0 y(t) = b x(t) + c \quad \dots\dots 2.11$$

โดยที่

$y(t)$  = เอาท์พุทหรือตัวแปรตาม

$x(t)$  = อินพุท

$t$  = เวลา

$a_0, a_1, b, c$  = ค่าคงที่

ถ้าพิจารณา ณ สถานะเริ่มต้น (Initial Steady State) ของระบบจะได้รับความสัมพันธ์ดังนี้

$$a_0 y(0) = b x(0) + c$$

เราสามารถหาความสัมพันธ์ระหว่างค่าเริ่มต้นของ  $x$  และ  $y$  ได้ดังนี้

$$a_1 \frac{dy(t)}{dt} - a_0 y(0) + a_0 y(t) = b x(t) - b x(0) + c - c$$

$$a_1 \frac{dy(t)}{dt} + a_0 [y(t) - y(0)] = b [x(t) - x(0)]$$

ให้

$$Y(t) = y(t) - y(0)$$

$$X(t) = x(t) - x(0)$$

แต่ในความเป็นจริงแล้วค่า  $\frac{dy(t)}{dt}$  ก็คือค่าความแตกต่างระหว่างค่า ณ เวลาใด ๆ กับค่าที่ของจุดเริ่มต้นจึง

$$\frac{dy(t)}{dt} = \frac{dY(t)}{dt}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ดังนั้นสมการ (2.11) จะกลายเป็น

$$a_1 \frac{dY(t)}{dt} + a_0 Y(t) = bX(t) \quad \dots 2.12$$

ให้  $\tau = a_1/a_0 =$  ค่าคงที่ของเวลา

$K = (b/a_0) =$  อัตราการขยายสัญญาณ ณ สถานะคงตัว (Steady State Gain)

จะได้ว่า

$$\tau \frac{dY(t)}{dt} + Y(t) = KX(t) \quad \dots 2.13$$

$$Y(t) = \left( \frac{K}{\tau s + 1} \right) X(s) \quad \dots 2.14$$

### 2.6.1.1 ผลตอบสนองต่อสัญญาณสตีป (Step Response)

ให้  $\Delta x =$  ขนาดของสัญญาณสตีป (Magnitude)

$u(t) =$  ฟังก์ชันสัญญาณหนึ่งหน่วย (Unit Step Function)

จะได้ว่า

$$X(t) = \Delta x \cdot u(t) \quad ; u(t) = 1$$

$$X(s) = \frac{\Delta x}{s} \quad \dots 2.15$$

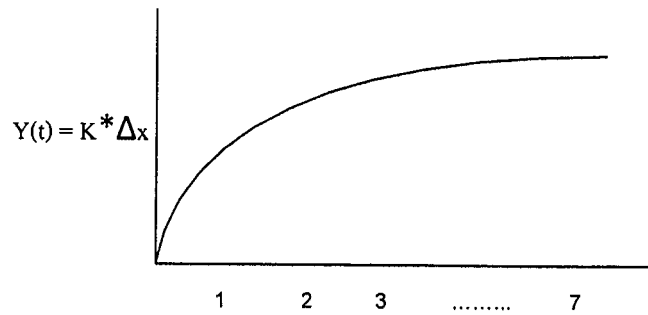
แทนค่าสมการ (2.15) ลงในสมการ (2.14) จะได้ว่า

$$Y(t) = \left( \frac{K}{\tau s + 1} \right) \left( \frac{\Delta x}{s} \right)$$

อินเวอร์ตลาปลาซจะได้

$$Y(t) = K\Delta x [u(t) - e^{-t/\tau}]$$

ซึ่งสามารถแสดงผลตอบสนองของสมการนี้ออกมาเป็นกราฟและเป็นตารางต่อไปนี้



รูปที่ 2.17 ผลตอบสนองต่อสัญญาณสตีปของระบบอันดับหนึ่ง (First-order Step Response)

ตารางที่ 2.1 ผลตอบสนองต่อสัญญาณสตีปของระบบอันดับหนึ่ง (First-order Step Response)

T/τ	Y(t) / K*
0	0
1	0.632
2	0.865
3	0.95
4	0.982
5	0.993
.	.
.	.
.	.
∞	1

จากข้อมูลในตารางข้างต้นจะพบว่าในช่วงต้น ๆ นั้น อัตราการเปลี่ยนแปลงของค่าเอาต์พุตจะมีค่าที่ค่อนข้างมาก แต่เมื่อเวลาผ่านไปอัตราการเปลี่ยนแปลงก็มีค่าลดลงในรูปแบบลักษณะของเอ็กซ์โพเนนเชียล (Exponential) นั่นเอง

### 2.6.1.2 ผลตอบสนองเมื่อมีการหน่วงเวลา (Response with Time Delay)

ในกระบวนการบางอย่างอาจจะมีการหน่วงเวลาของด้านเอาต์พุต (Time Delay or Transportation Lag or Dead - Time) โดยมีสมการดังนี้

$$Y(s) = \left( \frac{Ke^{-t_0s}}{\tau s + 1} \right) X(s) \quad \dots\dots 2.16$$

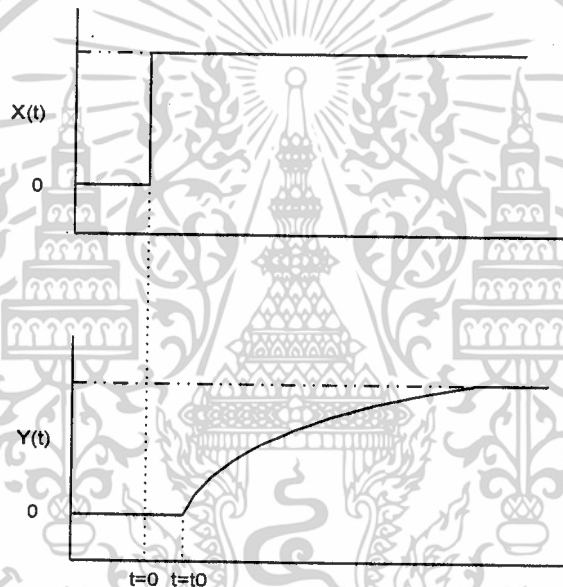
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เราเรียกสมการ (2.16) นี้ว่า “ระบบอันดับหนึ่งที่มีการหน่วงเวลา (First-order-plus-dead Time; FOPDT)” ถ้าสัญญาณอินพุทของระบบนั้นมีความแตกต่างกันสมการของการหน่วงเวลาก็จะแตกต่างกันไปดังนี้

1. สัญญาณสแต็ป (Step Response) จะได้สมการดังนี้

$$Y(t) = [K\Delta x \times u(t - t_0)][1 - e^{-(t-t_0)/\tau}]$$

โดยได้ผลการตอบสนองแสดงดังกราฟ



รูปที่ 2.18 ผลตอบสนองของระบบอันดับหนึ่ง เมื่อมีการหน่วงเวลา  
(First-order Step Response with Time Delay)

2. สัญญาณเชิงเส้น (Ramp Response) จะได้สมการดังนี้

$$Y(t) = u(t - t_0) \left[ \frac{K_r \tau e^{-(t-t_0)/\tau}}{\tau} \right] + [K_r(t - t_0 - \tau)]$$

3. สัญญาณรูปคลื่น (Sinusoidal Response) จะได้สมการดังนี้

$$Y(t) = u(t - t_0) \left\{ \left( \frac{KA\omega\tau}{1 + \tau^2\omega^2} \right) e^{-(t-t_0)/\tau} + \left( \frac{KA}{1 + \tau^2\omega^2} \right) \sin(\omega(t-t_0) + \Theta) \right\}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานในวิทยาลัยการศึกษานานาชาติ มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี เพื่อใช้ในการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 2.6.2 ผลตอบสนองต่อระบบอันดับสอง (Response of Second-order System)

ผลตอบสนองต่อระบบอันดับสองนี้ จะมีความแตกต่างกันขึ้นอยู่กับค่ารากของสัดส่วนในฟังก์ชันถ่ายโอนว่ามีค่าเท่าไรถ้า

- ถ้าค่ารากเป็นจำนวนจริงผลตอบสนองก็จะเรียกว่า “ระบบที่มีกระบวนการความหน่วงมาก (Over damped)

- ถ้าค่ารากเป็นจำนวนเชิงซ้อนผลตอบสนองก็จะเรียกว่า “ระบบที่มีกระบวนการความหน่วงน้อย (Under damped)” ซึ่งมักจะเกิดในกระบวนการที่มีอันดับสูงกว่าอันดับสอง

ระบบอันดับสองสามารถแสดงความสัมพันธ์ระหว่างตัวแปรเข้ากับตัวแปรออก ในรูปแบบสมการทั่วไปได้ดังนี้

$$a_2 \left( \frac{d^2 y(t)}{dt^2} \right) + a_1 \frac{dy(t)}{dt} + a_0 y(t) = b x(t) + c \quad \dots 2.17$$

โดยที่

$$y(t) = \text{ตัวแปรออก}$$

$$x(t) = \text{ตัวแปรเข้า}$$

$$a_0, a_1, a_2, c = \text{ค่าคงที่}$$

สมมติให้สมการ ณ สถานะเริ่มต้นมีค่าเป็น

$$a_0 \times y(0) = b x(0) + c \quad \dots 2.18$$

นำสมการ (2.7) ลบด้วยสมการ (2.8) จะได้ว่า

$$a_2 \left( \frac{d^2 Y(t)}{dt^2} \right) + a_1 \frac{dY(t)}{dt} + a_0 Y(t) = b X(t) \quad \dots 2.19$$

โดยที่

$$Y(t) = y(t) - y(0)$$

$$X(t) = x(t) - x(0)$$

แปลงสมการ (2.9) ให้อยู่ในรูปมาตรฐาน (โดยการหารด้วย  $a_0$  ตลอดสมการ) ดังนี้

$$\tau^2 \left( \frac{d^2 Y(t)}{dt^2} \right) + 2\zeta\tau \frac{dY(t)}{dt} + Y(t) = KX(t) \quad \dots 2.20$$

$$Y(s) = \left( \frac{K}{\tau^2 s^2 + 2\zeta\tau s + 1} \right) X(s) \quad \dots 2.21$$

ให้  $\tau = (a_2/a_0)^{1/2} =$  คุณสมบัติของเวลา (Characteristic Time)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$\zeta = \frac{a_1}{2\tau a_0} = \frac{a_1}{2\sqrt{a_0 a_1}} = \text{อัตราความหน่วง (Damping Rating)}$$

$$K = (b/a_0) = \text{อัตราขยายของกระบวนการ (Steady State Gain)}$$

หาคำรากของ  $\tau^2 s^2 + 2\zeta\tau s + 1 = 0$  (โดยพิจารณาเป็นแบบ Quadratic) จะได้สมการความสัมพัทธ์ดังนี้

$$r_{1,2} = \frac{-\zeta \pm \sqrt{\zeta^2 - 1}}{\tau}$$

จะพบว่าการพิจารณาคำรากของระบบโดยค่าที่เราได้อาจจะเป็นได้ทั้งจำนวนจริงและจำนวนเชิงซ้อน โดยที่ถ้า

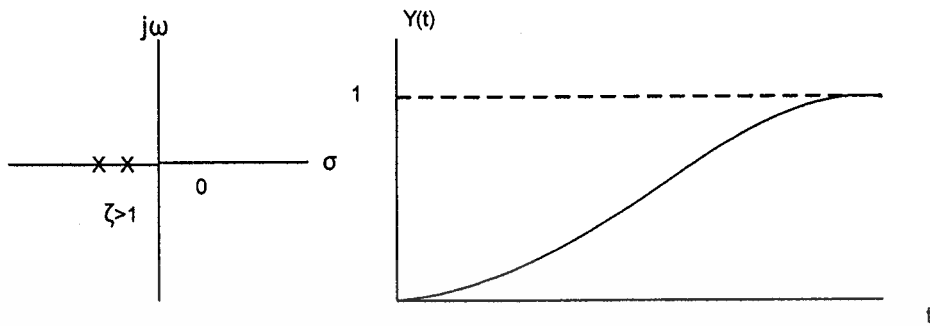
- ค่าของอัตราความหน่วงมากกว่า 1 ( $\zeta > 1$ ) คำรากที่ได้จะเป็นจำนวนจริง
- ค่าของอัตราความหน่วงน้อยกว่า 1 ( $\zeta < 1$ ) คำรากที่ได้จะเป็นจำนวนเชิงซ้อน 2 ค่าที่เป็นคอนจูเกตกัน (Conjugate number)
- ค่าของอัตราความหน่วงเท่ากับ 1 ( $\zeta = 1$ ) คำรากที่ได้จะเป็นจำนวนจริง 2 ค่าที่เท่ากัน โดยมีค่าเท่ากับ  $(-1/\tau)$
- ค่าของอัตราความหน่วงเท่ากับ 0 ( $\zeta = 0$ ) คำรากที่ได้จะเป็นจำนวนเชิงซ้อนที่ค่าจริง มีค่าเป็น (Real Part = 0) นั่นคือ รากจะมีค่าเท่ากับ  $\pm (1/\tau) \times i$

อัตราความหน่วงที่มีค่าความแตกต่างกันดังข้างต้นนี้ จะทำให้มีผลตอบสนองที่แตกต่างกัน ซึ่งสามารถแสดง ได้ดังตารางที่ 2.2

ตารางที่ 2.2 แสดงผลการตอบสนองเมื่อค่าของอัตราความหน่วงที่แตกต่างกัน

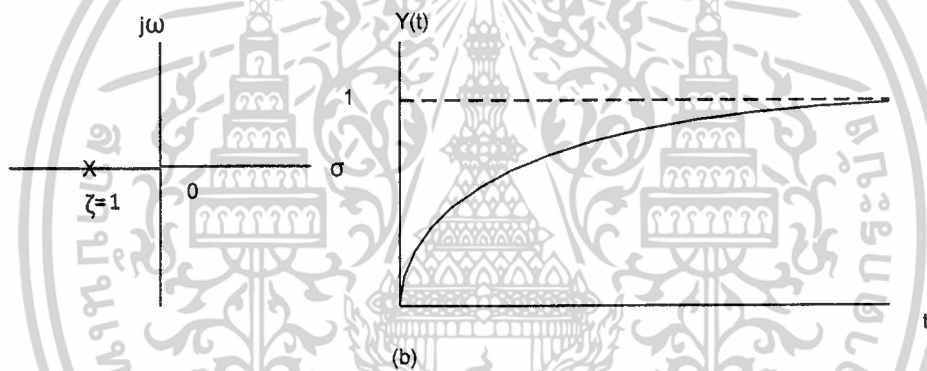
$\zeta$	ผลตอบสนอง (Response)
$\zeta > 1$	กระบวนการที่มีความหน่วงมาก (Over damped)
$\zeta = 1$	กระบวนการที่มีความหน่วงวิกฤต (Critically damped)
$0 < \zeta < 1$	กระบวนการที่มีความหน่วงน้อย (Under damped)
$\zeta = 0$	กระบวนการที่ไม่มีความหน่วง (Undamped)
$-1 < \zeta < 0$	กระบวนการที่ไม่เสถียร (Unstable)
$\zeta \leq -1$	กระบวนการที่หนีออก (Run - Away)

ซึ่งสามารถแสดงผลการตอบสนองได้ดังรูปที่ 2.19



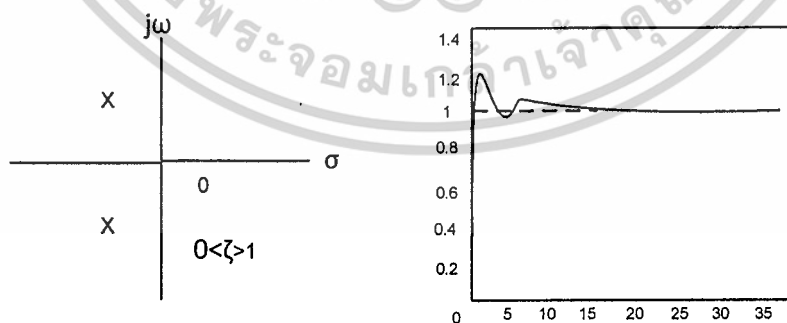
(a)

(a)  $\zeta > 1$  ภาวะบวมการที่มีความหน่วงมาก (Over damped)



(b)

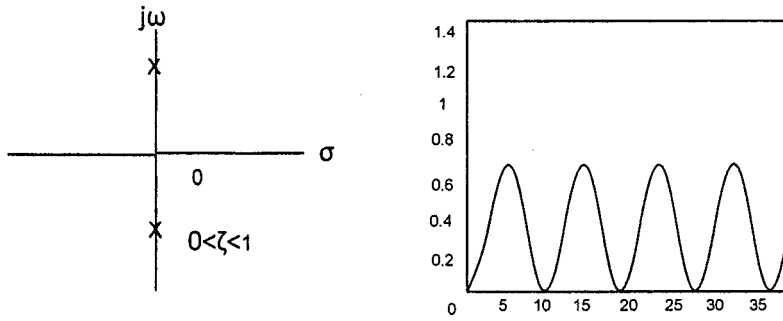
(b)  $\zeta = 1$  ภาวะบวมการที่มีความหน่วงวิกฤต (Critically damped)



(c)

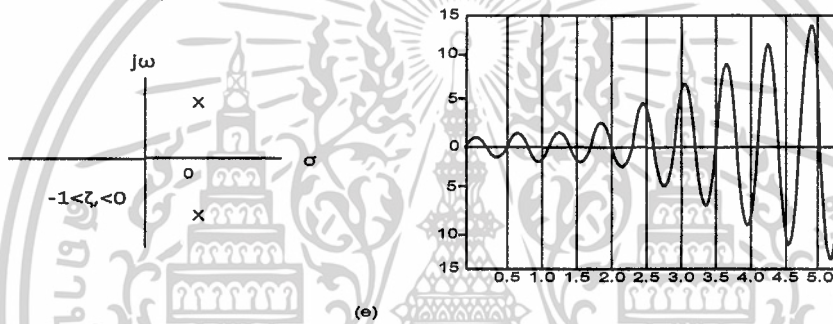
(c)  $\zeta < 0 < 1$  ภาวะบวมการที่มีความหน่วงน้อย (Under damped)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



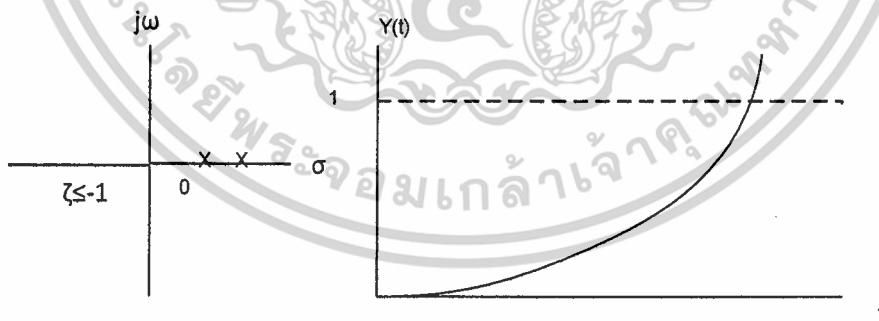
(d)

(d)  $\zeta = 0$  ภาวะวนการที่ไม่มีควมหน่วง (Undamped)



(e)

(e)  $-1 < \zeta < 0$  ภาวะวนการที่ไม่เสถียร (Unstable)



(f)

(f)  $\zeta \leq -1$  ภาวะวนการที่หนีออก (Run-away)

รูปที่ 2.19 แสดงความสัมพันธ์ระหว่างตำแหน่งโพลกับการตอบสนองของภาวะวนการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 2.7 การหาค่าพารามิเตอร์ของตัวควบคุมแบบ PID ด้วยวิธีของ Ziegler-Nichols

หลักการในการคำนวณทางคณิตศาสตร์นั้น ในการหาค่าการตอบสนองของกระบวนการที่ควบคุมนั้น ทำได้ยากเนื่องจากการหาสมการทางคณิตศาสตร์หรือฟังก์ชันถ่ายโอนที่เหมาะสมและเที่ยงตรงสำหรับแทนกระบวนการที่ถูกควบคุมนั้นเป็นไปได้ยาก โดยเฉพาะอย่างยิ่งในระบบของกระบวนการที่มีอันดับสูง ๆ ซึ่งคำตอบของสมการของระบบส่วนใหญ่มักจะเป็นสมการเดี่ยว ทำให้ไม่สามารถหาค่าพารามิเตอร์ที่มีหลายตัวเพื่อใช้ในการออกแบบระบบควบคุมได้ รวมทั้งยังมีค่าของผลกระทบจากตัวแปรอื่น ๆ ที่กระทำต่อระบบ จนทำให้ไม่สามารถหาค่าสมการทางคณิตศาสตร์ที่เหมาะสมแทนระบบนั้นได้

ในการออกแบบตัวควบคุมมีความจำเป็นอย่างยิ่ง ที่จะต้องรู้การตอบสนองทางพลวัตของกระบวนการก่อนซึ่งอยู่ในฟังก์ชันถ่ายโอน โดยสามารถหาได้จากโครงสร้างทางกายภาพต่าง ๆ ของกระบวนการ แต่ในบางกรณีในกระบวนการที่ซับซ้อนการทราบค่าทางกายภาพอาจไม่เพียงพอที่จะหาสมการทางคณิตศาสตร์มาแทนฟังก์ชันถ่ายโอนของระบบได้ แต่ก็สามารถใช้วิธีการทดสอบกระบวนการเพื่อหาพารามิเตอร์ที่เหมาะสมมาทำการออกแบบตัวควบคุมได้ จะเรียกว่าการปรับแต่งระบบควบคุม ซึ่งมีอยู่หลายวิธีแต่จะยกตัวอย่างวิธีที่นิยมกันคือ

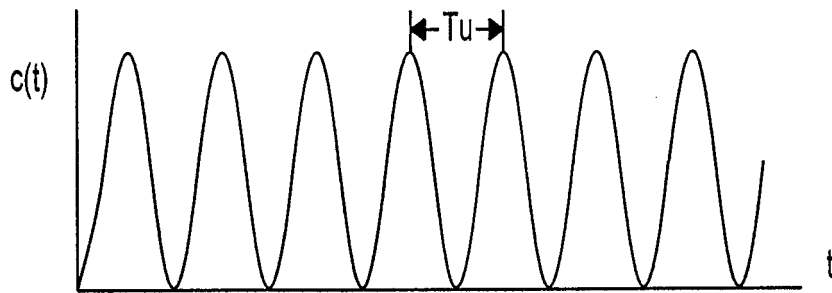
### วิธีการปรับค่าตัวควบคุม PID โดยใช้วิธี Ziegler-Nichols

หลักการประมาณค่าของ Ziegler-Nichols จะใช้สำหรับการประมาณหาค่าพารามิเตอร์ของตัวควบคุมแบบต่าง ๆ คือค่า  $K_p$ ,  $T_i$ ,  $T_d$  ให้ได้ประสิทธิภาพดีที่สุดและเพื่อให้ผลการตอบสนองของระบบต่ออินพุตที่เป็น unit step มีค่าพุ่งเกินสูงสุดไม่เกิน 25% มีอยู่สองแบบคือ

#### 2.7.1 Quarter decay ration response by ultimate gain

วิธีนี้จะทำการหาพารามิเตอร์ของตัวควบคุม จากผลตอบสนองของระบบหรือกระบวนการที่ถูกควบคุมต่อการปรับค่าของอัตราขยายของตัวควบคุมชนิด P เมื่อมีการป้อนสัญญาณขาเข้าเป็นแบบ unit step ดังนั้นวิธีนี้ต้องใช้ตัวควบคุมแบบ P ต่ออยู่เพื่อสามารถปรับอัตราขยายได้

คุณสมบัติของ ultimate gain และ ultimate period ของระบบนั้น ได้ถูกนำมาทำการหาค่าพารามิเตอร์ของตัวควบคุมซึ่งจะเป็นวิธีที่กระทำกับระบบโดยตรงและไม่จำเป็นต้องทราบค่าของฟังก์ชันถ่ายโอนของกระบวนการโดยปรับค่าอัตราขยาย  $K_p$  ของตัวควบคุมแบบ P ให้มีค่าที่สูงขึ้นเรื่อย ๆ จนให้ผลการตอบสนองของระบบเกิดการแกว่งอย่างต่อเนื่องและมีขนาดของการแกว่งคงที่เรียกอัตราขยายที่ทำให้เกิดการแกว่งนี้ว่า ultimate gain ( $K_u$ ) และจะเรียกค่าคาบของการแกว่งนี้ว่า ultimate period ( $T_u$ ) ดังรูปที่ 2.20 จากนั้นนำค่าเหล่านี้ไปปรับเทียบกับค่าในตารางที่ 2.3 เพื่อที่จะกำหนดพารามิเตอร์ของตัวควบคุมตามเงื่อนไขที่มีค่าพุ่งไม่เกิน 25% แต่วิธีนี้มีข้อจำกัดคือต้องใช้กับระบบที่มีผลตอบสนองต่อเวลาที่มีการแกว่งเท่านั้นหรือจะใช้กับกระบวนการที่มีรากของสมการที่มี



รูปที่ 2.20 แสดงการหาค่า ( $T_u$ ) จากผลการตอบสนองของระบบที่มีการแกว่งอย่างต่อเนื่อง เมื่อมีการปรับค่าอัตราขยายเป็น ( $K_u$ )

ตารางที่ 2.3 แสดงค่าพารามิเตอร์ของตัวควบคุม PID ตามวิธีของ Ziegler-Nichols

ชนิดของตัวควบคุม	$K_p$	$T_i$	$T_d$
แบบ P	$\frac{K_u}{2}$	-	-
แบบ PI	$\frac{K_u}{2.2}$	$\frac{T_u}{1.2}$	-
แบบ PID	$\frac{K_u}{1.7}$	$\frac{T_u}{2}$	$\frac{T_u}{8}$

### 2.7.2 Process reaction

เป็นวิธีการหาค่าพารามิเตอร์ของตัวควบคุม PID โดยตรง ซึ่งไม่มีต้องมีตัวควบคุมต่อร่วมอยู่แต่จะใช้ได้กับกระบวนการที่มีผลตอบสนองต่อ unit step เป็นรูปตัว S เท่านั้นคือกระบวนการที่ไม่มีโพลอยู่ที่จุดกำเนิด (กระบวนการชนิด 0) หรือไม่มีโพลเด่นชัดที่เป็นจำนวนเชิงซ้อนสังยุคโดยขั้นแรกจะทำการประมาณให้กระบวนการเป็นแบบใดแบบหนึ่งในสองแบบนี้ก่อนคือ

1. กระบวนการอันดับหนึ่งแบบมีเวลาไร้ผลตอบสนอง (first-order-plus-dead-time หรือเรียกว่า FOPDT) โดยมีฟังก์ชันถ่ายโอนคือ

$$G_1(s) = \frac{Ke^{-t_0s}}{\tau s + 1} \quad \dots 2.22$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. กระบวนการอันดับสองแบบมีเวลาไร้ผลตอบสนอง (second-order-plus-dead-time หรือเรียกว่า SOPDT) โดยมีฟังก์ชันถ่ายโอนคือ

$$G_1(s) = \frac{Ke^{-t_0s}}{(\tau_1s+1)(\tau_2s+1)} \quad \dots\dots 2.23$$

$$G_2(s) = \frac{Ke^{-t_0s}}{\tau^2+2\zeta\tau s+1} \quad \dots\dots 2.24$$

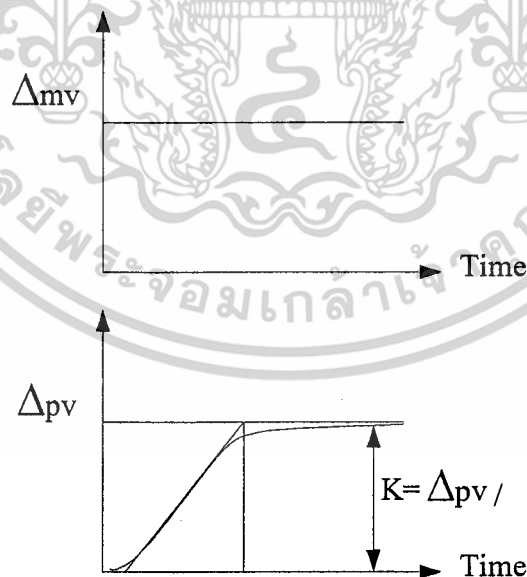
โดย  $K$  คือ ค่าอัตราขยายของระบบที่สถานะอยู่ตัว

$t_0$  คือ ค่าเวลาไร้ผลตอบสนองของกระบวนการ

$\tau, \tau_1, \tau_2$  คือ ค่าคงที่เวลาของกระบวนการ

$\zeta$  คือ ค่าอัตราการหน่วงของกระบวนการ

จากนั้นให้ทำการทดสอบกระบวนการด้วยการป้อนสัญญาณอินพุตที่เป็น Unit Step เข้าที่กระบวนการแล้วทำบันทึกผลการตอบสนองของกระบวนการซึ่งเป็นรูปตัว S นี้ จะนำมาอธิบายคุณลักษณะของกระบวนการด้วยพารามิเตอร์ 3 ค่า คือ อัตราการขยายของกระบวนการ ( $K$ ) ค่าเวลาหน่วงของกระบวนการ ( $t_0$ ) และค่าเวลาคงที่ของกระบวนการ ( $\tau$ ) ซึ่งจะถูกประมาณค่าโดยทำการลากเส้นสัมผัสกับจุดที่มีอัตราการเบี่ยงเบนสูงสุด โดยเส้นที่ลากนี้จะไปตัดกับแกนเวลาและเส้นตรงของผลการตอบสนองที่สถานะอยู่ตัวหรือที่  $\Delta P_v$  ดังรูปที่ 2.21 แสดงว่าการประมาณค่าต่าง ๆ โดยที่ FOPDT นั้นให้ผลที่สอดคล้องกับระบบจริง



รูปที่ 2.21 แสดงผลการตอบสนองของระบบแบบรูปตัว S และการหาค่าพารามิเตอร์ของระบบ โดยวิธี process reaction

ค่าพารามิเตอร์ของกระบวนการทั้ง 3 ค่าที่ได้นี้สามารถนำไปคำนวณหาค่าพารามิเตอร์ของตัวควบคุม PID ได้ Ziegler-Nichols จะกำหนดค่าของ  $K_p$ ,  $T_i$  และ  $T_d$  สำหรับตัวควบคุมแบบต่าง ๆ ตามตารางที่ 2.4

ตารางที่ 2.4 แสดงค่าพารามิเตอร์ที่ใช้ใน process reaction ของตัวควบคุม PID

โดยวิธีของ Ziegler-Nichols

ชนิดของตัวควบคุม	$K_p$	$T_i$	$T_d$
แบบ P	$\frac{1}{K} \left( \frac{\tau}{t_0} \right)$	-	-
แบบ PI	$\frac{0.9}{K} \left( \frac{\tau}{t_0} \right)$	$3.33t_0$	-
แบบ PID	$\frac{1.2}{K} \left( \frac{\tau}{t_0} \right)$	$2t_0$	$0.5t_0$

การหาค่าพารามิเตอร์โดยวิธีของ Ziegler-Nichols นั้น เป็นเพียงหาค่าโดยประมาณเท่านั้น ผู้ออกแบบจำเป็นต้องทำการปรับ โดยละเอียดอีกครั้งหนึ่งเมื่อนำไปใช้กับกระบวนการจริง

### 2.7.3 วิธี Trial and Error (วิธีการลองผิดลองถูก)

วิธีการลองผิดลองถูกเป็นวิธีการที่เหมาะสมกับผู้ที่มิประสบความสำเร็จในด้านการปรับค่าต่าง ๆ ของพารามิเตอร์ PID แล้ว โดยในการควบคุมจะเป็นระบบปิดและตำแหน่งการควบคุมต้องให้อยู่ในตำแหน่ง Automatic Control และมีวิธีการต่อไปก็คือ

- วิธีการปรับแบบ PI

1. ให้ตัวควบคุมทำงานในรูปแบบ Proportional Control เพียงอย่างเดียว
2. ต่อไปทำการปรับค่า  $K_c$  จนกระทั่งค่า PV เข้าใกล้ค่าเป้าหมายซึ่งโดยทั่วไปก็จะต่ำกว่าค่าเป้าหมายเพียงเล็กน้อย

3. ต่อไปค่อย ๆ ปรับค่า  $T_i$  เพิ่มขึ้นเพื่อชดเชยค่า offset ที่เกิดขึ้นแล้วดูผลว่าค่า PV จะเข้าสู่เป้าหมายตามที่ต้องการหรือยัง ถ้ายังก็ค่อย ๆ ปรับขึ้นไปอีกจนกระทั่งได้ตามต้องการ โดยช่วงนี้ก็พยายามลด  $K_c$  เพื่อลดการแกว่ง

- วิธีการปรับแบบ PID

1. ทำตามแบบวิธีการแบบ PI ทั้ง 3 ข้อ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. ถ้าต้องการผลตอบสนองที่รวดเร็วขึ้นมักใช้ค่าอัตราส่วนระหว่าง Integral Time กับค่า Derivative Time เป็น 4/1 เป็นต้นไปพร้อมกับการเพิ่มลดของค่า Kc ไปด้วยเพื่อป้องกันการเกิด Overshoot และเกิดการ Hunting

3. เนื่องด้วยวิธีการแบบนี้เป็นความพึงพอใจเฉพาะบุคคล จึงยากที่จะกำหนดเป็นกฎเกณฑ์ตายตัว แต่วิธีการเริ่มต้นในการปรับค่ามักเหมือนกัน

## 2.8 ลักษณะอุปกรณ์การทำงานและการเลือกใช้อุปกรณ์

### 2.8.1 วาล์วควบคุม (Control Valve)

วาล์วควบคุมเป็นอุปกรณ์ที่ทำหน้าที่ควบคุมอัตราการไหลของของไหลในระบบ ซึ่งได้แก่ ไอน้ำ ของเหลว ก๊าซ และรวมไปถึงสารเคมี โดยจะรับสัญญาณควบคุมจากตัวควบคุมเพื่อให้การควบคุมเป็นไปอย่างอัตโนมัติ

#### 2.8.1.1 ประเภทของวาล์วควบคุมแบ่งตามลักษณะการเปิด-ปิด

สามารถแบ่งประเภทของวาล์วควบคุมนี้ ตามลักษณะการเคลื่อนที่เพื่อเปิดหรือปิด เพื่อให้ของไหลผ่านช่องทางเดิน (Port) ได้เป็น 2 ประเภทใหญ่ ๆ คือ

- Linear-Shaft Valve หรือ Linear Motion Type

จะมีลักษณะการเปิด-ปิดของวาล์วที่อยู่ในแนวเส้นตรง (ขึ้น-ลง) โดยตัวอย่างวาล์วชนิดนี้ที่ใช้งานมากได้แก่ชนิดที่เป็น Globe Valve

- Rotary-Shaft Valve หรือ Rotary Motion Type

จะมีลักษณะการเปิด-ปิดของวาล์ว จะเป็นแบบการเคลื่อนที่ตามแนวเส้นรอบวง ตัวอย่างวาล์วชนิดนี้ได้แก่ Butterfly Valve, Eccentric Rotary Valve และ Ball Valve

#### 2.8.1.2 Valve Body

Valve Body ที่ใช้ในระบบควบคุมมีหลายแบบซึ่งในแต่ละแบบจะมีลักษณะเฉพาะตามความเหมาะสมของการทำงาน โดยทั่วไปชนิดของ Valve Body จะใช้กับการทำงานตามลักษณะการเปิด-ปิดช่องทางเดินของของไหลมีอยู่ 2 แบบ ก็คือแบบ Linear Motion Type และแบบ Rotary Motion Type

ส่วนใหญ่ในงานระบบควบคุมที่นิยมใช้งานกันดังนี้

1. Globe Valve
2. Butterfly Valve
3. Eccentric Rotating Valve
4. Ball Valve

เอกสารนี้เป็นเอกสารที่จัดทำขึ้นเพื่อใช้ในการเรียนการสอนเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 2.8.1.3 Valve Trim

Trim เป็นหัวใจหรือส่วนหลักของวาล์วควบคุมทำหน้าที่ประการแรกคือการสร้างพื้นที่การไหล (Orifice Area) ตามสัดส่วนกับการเคลื่อนที่ของ Valve Plug เพื่อให้ได้ค่าของ Flow Capacity หน้าที่ของประการที่สองคือการปิดไม่ให้ของไหลไหลผ่านไปได้หน้าที่ดังกล่าวจะทำงานร่วมกับ Valve Body, Valve Trim, Actuator, Packing และแรงจากการไหลของของไหลซึ่งขึ้นอยู่กับ Trim

Valve Body จะประกอบด้วยชิ้นส่วนที่อยู่ภายใน Valve Body ที่จะสัมผัสอยู่กับของไหลในขณะที่ไหลผ่านวาล์วควบคุม ซึ่งได้แก่สิ่งต่อไปนี้คือ Valve Plug, Seat, Seat Retainer, Stem Guide, และ Bushing ส่วน Stuffing Box จะมีส่วนประกอบที่ถือว่าเป็น Valve Trim นั่นก็คือ Packing และ Packing Follower

Valve Trim จะเคลื่อนที่ขึ้น-ลงเพื่อในการควบคุมอัตราการไหลของของไหลและปิดไม่ให้ของไหลไหลผ่านโดย Valve Plug จะต้องกดลงบน Seat Ring ด้วยแรงกดของก้านวาล์วที่ส่งมาจาก Actuator

ในขณะที่ Valve Plug เคลื่อนที่นั้น จะทำให้เกิดของแรงการดันการไหลของของไหล (dynamic Fluid Flow) ซึ่งแรงที่มาจาก Actuator นั้นจะต้องเอาชนะความดันจากของไหลให้ได้เพื่อทำให้การควบคุมอย่างต่อเนื่องเป็นไปได้ดีแต่ในการควบคุมการไหลอย่างต่อเนื่องนี้จะขึ้นอยู่กับรูปร่างของ Valve Plug ต่าง ๆ เช่น แบบ Linear หรือ Equal - Percentage

นอกจากนั้นยังมีองค์ประกอบอื่น ๆ ที่มีผลต่อการออกแบบ Valve Trim อันได้แก่อุณหภูมิ ความดัน, การเกิด Flashing Fluids, การเกิด Cavitation และ Fluid Viscosity การควบคุมอัตราการไหลที่ต้องการจำเป็นต้องมีอุปกรณ์อีกอย่างคือตัวควบคุมตำแหน่งวาล์ว (Valve Positioner)

### 2.8.2 การเลือกวาล์วควบคุมเพื่อควบคุมการไหล

1. จะต้องทราบข้อมูลที่เกี่ยวข้องกับระบบของกระบวนการ เช่น ความดัน อุณหภูมิและสภาพการไหล (Flow Condition) และคุณสมบัติของของไหล
2. เลือกรูปแบบ Inherent Flow Characteristic ของบริษัทผู้ผลิตที่แตกต่างกันในรูปร่างของ Valve Trim และเลือกรูปแบบของ Plug และวัสดุที่ใช้ทำ
3. การเปลี่ยนแปลงของ Inherent Flow Characteristic ที่ตามการเปลี่ยนแปลงของ Flow Condition ส่วน Maximum Flow Capacity ขึ้นอยู่กับขนาดของ Port Size
4. การเลือกแบบ Body, Actuator, วัสดุที่ใช้ทำและขนาดกำลังของ Actuator
5. การคำนวณและเลือก  $C_v$  ให้เหมาะสม
6. การหาตัวควบคุมตำแหน่งที่เหมาะสมกับช่วงชัก (Stroke) ของวาล์วควบคุม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

7. เลือก Class ของวาล์วนั้น จะมีผลเกี่ยวกับการรั่วซึมที่ใช้กับกระบวนการที่เป็นก๊าซและน้ำมัน

### 2.8.3 คุณลักษณะการไหลของวาล์วควบคุม (Flow Characteristic of Control Valve)

คุณลักษณะการไหลของวาล์วควบคุมตัวใด ๆ นั้นก็คือความสัมพันธ์ระหว่างอัตราการไหลที่ไหลผ่านวาล์วควบคุมกับการเคลื่อนที่ขึ้น-ลงของ Valve Plug ที่จากช่วงชักในระหว่าง 0-100% โดยทั่วไป Flow Characteristic จะกล่าวถึงคุณลักษณะประจำหรือเฉพาะแบบได้ 2 แบบคือ

1. **Inherent Flow Characteristic** หมายถึงคุณลักษณะการไหลที่ได้จากการที่ของไหล ๆ ผ่านวาล์วเมื่อความดันตกคร่อมวาล์วมีค่าคงที่ (Constant Pressure Drop) และถ้าจะให้กล่าวอีกแนวหนึ่งก็จะเป็นคุณลักษณะของการไหลที่ได้จากการออกแบบและการทดลองของวาล์วควบคุมก่อนนำออกไปใช้งานจริง

2. **Installed Flow Characteristic** จะหมายถึงคุณลักษณะการไหลเมื่อ Pressure Drop ของวาล์วจะเปลี่ยนแปลงไปตามอัตราการไหลหรือการเปลี่ยนแปลงที่เกิดขึ้นในระบบนั้น เมื่อวาล์วติดตั้งอยู่ในกระบวนการใช้งานจริง

วัตถุประสงค์ของการกำหนดคุณลักษณะการไหลนั้นเพื่อเป็นแบบอย่างสำหรับเลือกใช้ตามสภาพและเงื่อนไขของการควบคุม (Control Loop and Process Condition) อันที่จะทำให้ได้ความเสถียรภาพในการควบคุมที่ดีขึ้น ตามปกติคุณลักษณะการไหลที่ใช้กันมากและพบบ่อย ๆ จนกล่าวได้ว่ามีด้วยกัน 3 แบบ คือแบบ Linear, Equal Percentage และ Quick Opening

#### 2.8.3.1 คุณลักษณะเส้นตรง (Linear Characteristic)

อัตราการไหลจะแปรผันโดยตรงกับการเปิด-ปิดของ Valve Plug โดยจะมีการให้ความสัมพันธ์มีลักษณะเป็นเส้นตรงนี้ถ้า Pressure Drop คงที่ Valve Gain ก็จะมีการคงที่ที่อัตราการไหลทุกจุดของค่า Gain นี้และจะมีความสัมพันธ์กับขนาดของวาล์ว การใช้งานและคุณลักษณะของ Valve Plug คุณลักษณะเส้นตรงแบบนี้ นิยมใช้กับการควบคุมระดับควบคุมการไหลที่มี Gain คงที่ของกระบวนการที่มีการตอบสนองช้า (Slow Process)

#### 2.8.3.2 Equal-Percentage Characteristic

อัตราส่วนที่ไหลที่มีการเพิ่มจะแปรผันโดยตรงกับการเปิดเพิ่มของ Valve Plug ในลักษณะเส้นกราฟแสดงลักษณะความสัมพันธ์การเปิด-ปิดของ Valve Plug นั้นจะเท่ากับเปอร์เซ็นต์ของการเปลี่ยนแปลงของอัตราการไหลหรือพิจารณาได้จากสมการข้างล่างนี้

$$Q = Q_0 e^{mx} \quad \dots\dots 2.25$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยที่

$$Q_0 = \text{อัตราการไหล}$$

$$X = \text{ตำแหน่งของวาล์ว}$$

$$M = L_n \frac{R}{T} \text{ เป็นค่าคงที่ของวาล์วแต่ละตัว}$$

$$R = \text{Valve Range ability}$$

$$T = \text{ช่วงชักสูงสุดของวาล์ว} = \frac{Q_m}{Q_0}$$

$$Q_m = \text{อัตราการไหลสูงสุด}$$

จากสมการจะเห็นได้ชัดว่าในช่วงการเปิดของวาล์วที่ 20% นั้น ที่ค่าเปอร์เซ็นต์ของการไหลมีค่าประมาณ 6 % เมื่อมีช่วงการเปิดของวาล์วเพิ่มขึ้นเป็น 40 % เปอร์เซ็นต์ของการไหลจะเพิ่มเป็น 12 % และในช่วงการเปิดของวาล์วที่ 60% นั้น เปอร์เซ็นต์ของการไหลก็จะเพิ่มเป็น 240 % และในที่สุดที่ช่วงการเปิดวาล์วของวาล์วจาก 80 % - 100 % (20 % สุดท้าย) นั้น เปอร์เซ็นต์ของการไหลจะเปลี่ยนแปลงจาก 50% ถึง 100% นั่นก็คือจะเห็นได้ว่าช่วงการเปิดวาล์วที่เพิ่มขึ้นทีละ 20 % นั้น จะทำให้อัตราการไหลเพิ่มขึ้นเป็น 2 เท่า จะนิยมใช้ในการควบคุมระดับและอุณหภูมิและใช้งานที่มีการเปลี่ยนแปลงของความดันตกคร่อมวาล์ว (Pressure Drop) วาล์วควบคุมอยู่เสมอ

### 2.8.3.3 Quick Opening Characteristic

การเปลี่ยนแปลงอัตราการไหลมาในขณะที่ Valve Plug เคลื่อนที่เพียงเล็กน้อยจะให้การไหลในลักษณะเส้นตรงประมาณ 65 – 70% ของอัตราการไหลสูงสุด และถ้าการเพิ่มการเปิดของ Valve Plug มากกว่านี้ การเปลี่ยนแปลงของอัตราการไหลก็จะลดลงจนเกือบไม่เปลี่ยนแปลง คุณลักษณะการไหลแบบนี้นิยมใช้งาน On - Off Service ควบคุมความดัน และการใช้งานเกี่ยวกับประเภท Safety Valve

## 2.9 เพรสเซอร์ทรานสดิวเซอร์ (Pressure transducer)

เพรสเซอร์ทรานสดิวเซอร์ (Pressure transducer) คืออุปกรณ์ที่จะทำหน้าที่แปลงความดันไปเป็นสัญญาณทางไฟฟ้าแบบอนาล็อก (analogue) โดยทั่วไปเพรสเซอร์ทรานสดิวเซอร์มีหลายชนิดด้วยกันแต่ส่วนที่นิยมใช้งานมากที่สุดคือชนิดที่ใช้สเตรนเกจ (strain-gage base transducer)

หลักการของเพรสเซอร์ทรานสดิวเซอร์แบบสเตรนเกจนี้นั้นจะอาศัยการเปลี่ยนรูปแบบทางกายภาพ (Physical deformation) หรือการยืดหดตัวของสเตรนเกจ (Strain gages) ซึ่งจะยึดติดอยู่กับโคอะเฟรมของเพรสเซอร์ทรานสดิวเซอร์ และต่อวงจรไปยังวงจรอิเล็กทรอนิกส์แบบวิทสโตนบริดจ์ (Wheatstone bridge) โดยการแปลงความดัน (Pressure) ให้ไปเป็นสัญญาณทางไฟฟ้านั้น (Electrical signal) ทำได้โดยการอาศัยความดันที่ป้อนเข้าไปที่ตัวของเพรสเซอร์ทรานสดิวเซอร์แล้วทำให้เกิด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การหักเหของไดอะแฟรม (Diaphragm) และความเครียดที่เกิดจะเป็นผลทำให้เกิดการเปลี่ยนแปลงค่าความต้านทานที่วงจรไฟฟ้าซึ่งแปรผันตรงกับความดันด้านอินพุตของทรานสดิวเซอร์

### 2.9.1 เอาท์พุทของเพรสเชอร์ทรานสดิวเซอร์

เพรสเชอร์ทรานสดิวเซอร์ (Pressure transducer) โดยทั่วไปจะมีแรงดันไฟด้านขาออกหรือเอาท์พุทซึ่งเป็นสัญญาณไฟฟ้าสามประเภทด้วยกันคือ ชนิดที่หนึ่งมีหน่วยเป็นมิลลิโวลต์ (Millivolt Output Pressure Transducers), ชนิดที่สองหน่วยเป็น โวลต์ (Voltage Output Pressure Transducers) และชนิดที่สามมีหน่วยเป็นมิลลิแอมป์ (4-20 mA Output Pressure Transducers)

#### 2.9.1.1 ชนิดเอาท์พุทมีหน่วยเป็นมิลลิโวลต์

เพรสเชอร์ทรานสดิวเซอร์ชนิดนี้ราคาจะถูกที่สุด โดยทั่วไปจะมีเอาท์พุทที่ 30mV สัญญาณด้านเอาท์พุทจะเป็นสัดส่วน โดยตรงกับอินพุต และในกรณีสัญญาณอินพุท ไม่นิ่งหรือมีการกระเพื่อมก็จะส่งผลให้สัญญาณเอาท์พุทมีกระเพื่อมตามด้วยเช่นกัน (เอาท์พุทมีการขึ้นอยู่กัอินพุทโดยตรง) จากเหตุผลดังกล่าวทำให้ทรานสดิวเซอร์ชนิดนี้มีข้อจำกัดการใช้งาน เช่น แหล่งจ่ายที่จะนำมาใช้กับทรานสดิวเซอร์ประเภทนี้จะต้องเป็นแหล่งจ่ายที่ค่อนข้างเรียบและสม่ำเสมอ การติดตั้งจะต้องห่างจากบริเวณที่มีสัญญาณรบกวนทางไฟฟ้า นอกจากนั้นตำแหน่งที่ติดตั้งทรานสดิวเซอร์กับอุปกรณ์สำหรับอ่านค่าจะต้องติดตั้งอยู่ใกล้กันด้วยเช่นกัน

#### 2.9.1.2 ชนิดเอาท์พุทมีหน่วยเป็นโวลต์

ทรานสดิวเซอร์ประเภทนี้จะประกอบด้วย integral signal conditioning ซึ่งจะมีค่าเอาท์พุทสูงกว่าแบบที่มีหน่วยเป็นมิลลิโวลต์ และสัญญาณเอาท์พุทโดยทั่วไปจะอยู่ในย่าน 0-5 Vdc หรือ 0-10 Vdc ซึ่งจะขึ้นอยู่กัคุณสมบัติของแต่ละรุ่นและสัญญาณด้านเอาท์พุทก็จะไม่เป็นฟังก์ชันโดยตรงกับอินพุทที่มากกระทำซึ่งถือว่าเป็นข้อได้เปรียบที่จะสามารถใช้ร่วมกับแหล่งจ่ายมีแรงดันอยู่ในย่านที่กำหนดได้โดยไม่จำเป็นต้องใช้แหล่งที่เรียบหรือคงที่สม่ำเสมอตลอด เนื่องจากจะมีระดับสัญญาณแรงดันเอาท์พุทที่สูงกว่าและป้องกันสัญญาณรบกวนทางไฟฟ้าได้ดีกว่า จะทำให้เหมาะสำหรับการใช้งานในภาคอุตสาหกรรมที่มีสัญญาณรบกวนและการกระเพื่อมของสัญญาณทางด้านอินพุทที่มีความรุนแรงได้ดีกว่า

#### 2.9.1.3 ชนิดเอาท์พุทมีหน่วยเป็นมิลลิแอมป์

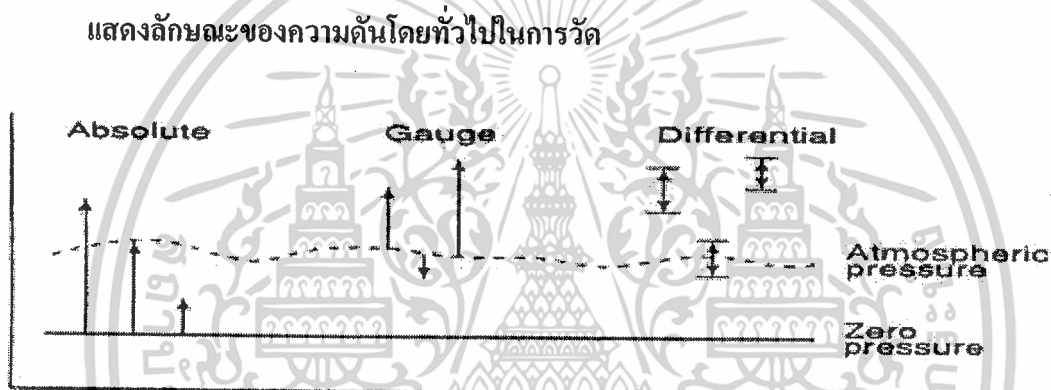
ทรานสดิวเซอร์ชนิดนี้มักเรียกกันโดยทั่วไปว่า Pressure transmitters transducers สัญญาณเอาท์พุทอยู่ในย่าน 4 - 20 mA ซึ่งจะถือว่าเป็นข้อได้เปรียบเนื่องจากจะได้รับผลกระทบจากสัญญาณรบกวนทางไฟฟ้าน้อยและมีค่าความต้านทานในสายค่อนข้างต่ำ เหมาะสำหรับการประยุกต์ใช้

กับงานที่มีการส่งสัญญาณในระยะไกล ๆ โดยทั่ว ๆ ไป นิยมประยุกต์ใช้กับงานที่ต้องใช้สายสัญญาณที่มีระยะทางตั้งแต่ 1000 ฟุต หรือมากกว่า

## 2.9.2 ประเภทของความดัน (Types of Pressure)

ประเภทของความดันมี 4 ประเภทคือ

- ความดันเกจ (Gauge Pressure)
- ความดันสัมบูรณ์ (Absolute Pressure)
- ความดันแตกต่าง (Differential Pressure)
- สภาวะสุญญากาศ (Vacuum Pressure)



รูปที่ 2.22 ประเภทของความดัน

### 2.9.2.1 ความดันเกจ (Gauge Pressure)

เป็นค่าที่ใช้ในงานอุตสาหกรรมโดย Gauge Pressure นี้ จะมีค่าเป็นศูนย์ที่ความดันบรรยากาศ สัญลักษณ์ที่ใช้คือ  $P_g$  หน่วยวัดที่ใช้จะมีตัวอักษร  $g$  กำกับอยู่ เช่น Barg, Psig

### 2.9.2.2 ความดันสัมบูรณ์ (Absolute Pressure)

สัญลักษณ์ที่ใช้คือ Pa หรือ Pabs โดยหน่วยวัดที่ใช้จะมีอักษร abs หรือ a กำกับอยู่ เช่น bar(abs), Psia เป็นต้น ความดันสัมบูรณ์มีค่าเท่ากับ 101.325 kPa ที่ความดันบรรยากาศ(1atm) ค่าความดันสัมบูรณ์จะใช้สำหรับในการคำนวณทาง Thermodynamic เป็นส่วนใหญ่ เช่น ในการหา Boiler Efficiency เป็นต้น

### 2.9.2.3 ความดันแตกต่าง (Differential Pressure)

สัญลักษณ์ที่ใช้คือ  $\Delta P$  หน่วยวัดที่ใช้มักมีอักษร d, D กำกับอยู่ เช่น Psid เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 2.9.2.4 สภาวะสุญญากาศ (Vacuum Pressure)

สัญลักษณ์ที่ใช้ไม่ปรากฏแน่ชัด แต่ส่วนมากจะใช้เครื่องหมายลบกำกับหน้าตัวเลข นอกจากนี้ยังใช้ abs หรือ Pabs ได้ด้วยแต่อาจเกิดความสับสนได้ง่าย วิธีที่ดีที่สุดคือควรกำหนดเป็นค่าตัวเลขติดลบ เช่น  $P = -10 \text{ Psig}$  โดยหน่วยวัดที่ใช้โดยทั่วไปจะมีอักษร Vac จะกำกับไว้อยู่ที่ท้าย เช่น 750 mmHg vac เป็นต้น

## 2.10 โครงสร้างไมโครคอนโทรลเลอร์ ARM7

ARM7024 จะเป็นบอร์ดไมโครคอนโทรลเลอร์ในตระกูล ARM7 มีขนาด 64 Pin ซึ่งเลือกใช้ไมโครคอนโทรลเลอร์เบอร์ ADUC7024 ของ Analog Device เป็น MCU ประจำบอร์ดโดย MCU รุ่นนี้จะบรรจุอยู่ในตัวถังแบบ 64 Pin LQFP โดย MCU ตัวนี้จะมีจุดเด่นคือ ความสามารถในการติดต่อเชื่อมต่อกับสัญญาณแบบ Analog ซึ่งจะมีทั้ง ADC ขนาด 12 บิต จำนวน 10 ช่อง และ DAC ขนาด 12 บิต จำนวน 2 ช่อง นอกจากนี้แล้วความสามารถทางด้านของความเร็วในการประมวลผลก็ถือว่าไม่ด้อยกว่าตัวอื่น ๆ ซึ่งสามารถทำงานได้ด้วยความถี่สูงสุดที่ 41.78 MHz โดยใช้ XTAL ที่ 32.768 KHz ร่วมกับวงจรควบคุมความถี่แบบ Phase Lock Loop ภายในตัว MCU นอกจากนี้แล้วยังมีความเพียบพร้อมด้วยอุปกรณ์พื้นฐานต่าง ๆ ที่จำเป็นต่อการใช้งานต่าง ๆ ไม่ว่าจะเป็นหน่วยความจำโปรแกรมแบบ Flash ขนาด 62 KByte และหน่วยความจำใช้งานแบบ RAM ซึ่งมีมากถึง 8 Kbyte ส่วนในด้านของอุปกรณ์ Peripheral นั้นก็นับว่าครบถ้วนเหมาะแก่การนำไปประยุกต์ใช้งานเกี่ยวกับการควบคุมและประมวลผลต่าง ๆ ได้เป็นอย่างดี โดยมีทั้ง SPI, UART Watchdog, Timer/Counter, PWM โดยการออกแบบโครงสร้างของบอร์ดนั้นจะเน้นเรื่องขนาดของบอร์ดให้มีขนาดเล็กเพื่อให้ง่ายต่อการนำไปประยุกต์ใช้งานและสะดวกต่อการพัฒนาของโปรแกรม

### คุณสมบัติของไมโครคอนโทรลเลอร์ ARM 7

1. เลือกใช้ MCU ตระกูล ARM7 TDMI Core เบอร์ ADUC7024 ของ Analog Device เป็น MCU ประจำบอร์ด โดยการเลือกใช้แหล่งกำเนิดสัญญาณนาฬิกาแบบ XTAL ค่า 32.768 KHz ซึ่งจะสามารถกำหนดการทำงานร่วมกับ Phase Lock Loop โดย MCU สามารถประมวลผลด้วยความเร็วสูงสุดที่ 41.78 MHz ได้ด้วยโดยคุณสมบัติเด่น ๆ ของ MCU ได้แก่

- มีหน่วยความจำ Flash สำหรับเขียนโปรแกรม 62 KByte และมี RAM ที่มีขนาด 8 KByte
- มีพอร์ต I/O จำนวน 5 พอร์ตคือ P0(6Bit), P1(8Bit), P2(1Bit), P3(8Bit) และ P4(8Bit)
- มีวงจรสื่อสารอนุกรม UART จำนวน 1 พอร์ต และมีวงจรสื่อสาร SPI จำนวน 1 พอร์ต
- มีวงจร Timer/Counter จำนวน 4 ชุด
- มีวงจร ADC ขนาด 12 บิต จำนวน 10 ช่อง และ DAC ขนาด 12 บิต จำนวน 2 ช่อง

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้เพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. มีขั้วต่อสัญญาณ I/O แบบ TTL แบบ Header 2x5 จำนวน 3 ชุด (P1, P3 และ P4)
3. มีขั้วต่อ LCD แบบ Header 2x7 รองรับการเชื่อมต่อกับ LCD Character
4. มีขั้วต่อใช้งาน RS232 สำหรับใช้งานและสำหรับ Download ผ่าน RS232
5. ใช้แหล่งจ่ายไฟขนาด +5 VDC ร่วมกับ Regulate 3.3V on Board



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

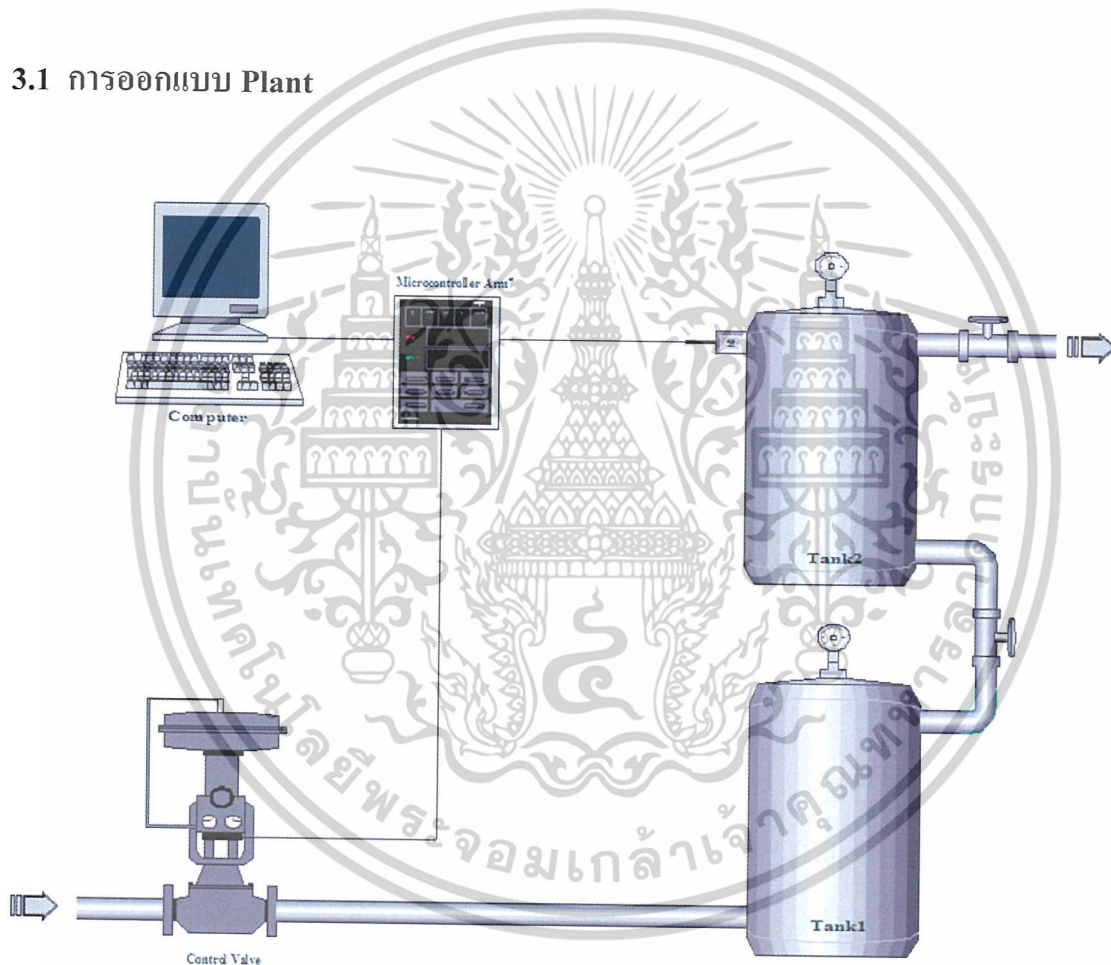
## บทที่ 3

### การออกแบบและวิธีการดำเนินการ

ในการออกแบบของโครงสร้าง จะแบ่งการออกแบบออกเป็น 3 ส่วนหลัก ๆ คือ

- 3.1 การออกแบบ Plant
- 3.2 การออกแบบ Microcontroller ARM7
- 3.3 การออกแบบโปรแกรมการแสดงผล

#### 3.1 การออกแบบ Plant



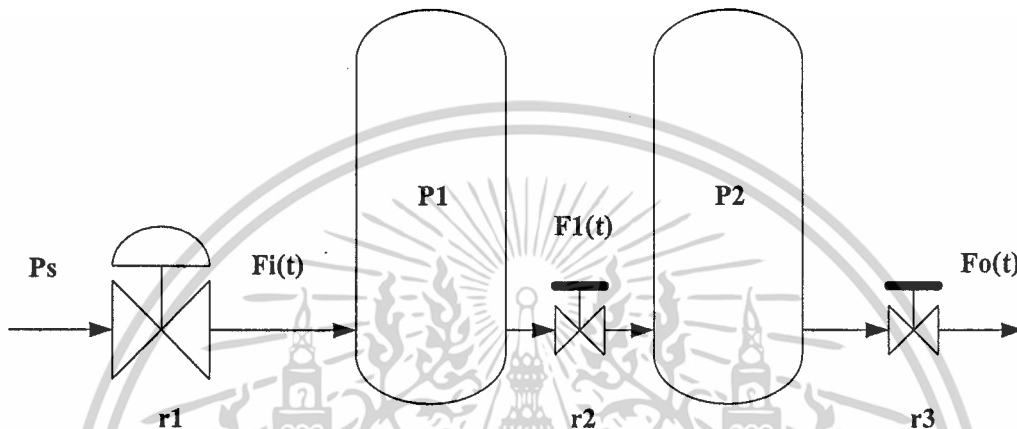
รูปที่ 3.1 โครงสร้างของกระบวนการ

ในโครงการนี้เป็นารออกแบบกระบวนการในการควบคุมความดัน (Pressure) แบบ 2 ถึง ควบคุมความดันในถังให้คงที่ โดยในการควบคุมนั้นจะใช้ไมโครคอนโทรลเลอร์ Arm7 เป็นตัวควบคุมสั่งให้ Control Valve เปิด-ปิด เพื่อควบคุมอัตราการไหลของอากาศเข้าไปในถังเพื่อให้ได้ความดันภายในถังนั้นคงที่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ส่วนประกอบหลักของกระบวนการนั้นจะประกอบไปด้วย วาล์วควบคุม (Control Valve), Pressure Transducer, Pressure Gage และวงจร Signal Condition การทำงานของแต่ละอุปกรณ์นั้นจะทำงานร่วมกัน เพื่อให้เกิดความเสถียรมากที่สุด เราจะมาดูลักษณะอุปกรณ์ต่างๆ รวมทั้งการทำงานและการเลือกใช้แต่ละอุปกรณ์ของกระบวนการนี้

3.1.1 สมการคุณลักษณะของกระบวนการ



รูปที่ 3.2 ระบบที่ใช้ในการควบคุม

Tank 1

$$\begin{aligned} \rho f_i(t) - \rho f_1(t) &= \frac{dP_1(t)}{dt} \\ \rho f_i(t) - \rho f_1(t) &= \frac{V_1}{RT} \frac{dP_1(t)}{dt} \end{aligned} \quad \dots 3.1$$

$V_1$  = ปริมาตร Tank 1

$R$  = ค่าคงที่ของแก๊ส

$T$  = อุณหภูมิ (K)

Tank 2

$$\rho f_1(t) - \rho f_o(t) = \frac{V_2}{RT} \frac{dP_2(t)}{dt} \quad \dots 3.2$$

$$f_1 = r_2 \sqrt{P_1(P_1(t) - P_2(t))} \quad \dots 3.3$$

ประมาณเชิงเส้น  $f_1(t)$  จะได้ว่า

$$f_1 \approx \bar{f}_1 + \left. \frac{\partial f_1}{\partial P_1} \right|_{ss} P_1(t) + \left. \frac{\partial f_1}{\partial P_2} \right|_{ss} P_2(t)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$f_1(t) \approx \bar{f}_1 + C_1 P_1(t) + C_2 P_2(t) \quad \dots 3.4$$

เมื่อ

$$C_1 = r_2 \frac{1}{2} [\bar{P}_1 (\bar{P}_1 - \bar{P}_2)]^{-1/2} (2\bar{P}_1 - \bar{P}_2) \quad \dots 3.5$$

$$C_2 = r_2 \frac{1}{2} [\bar{P}_1 (\bar{P}_1 - \bar{P}_2)]^{-1/2} (-\bar{P}_1) \quad \dots 3.6$$

$$f_o(t) = r_3 \sqrt{P_2 (P_2(t) - P_{atm})} \quad \dots 3.7$$

ประมาณเชิงเส้น  $f_o(t)$  จะได้ว่า

$$f_o(t) \approx \bar{f}_o + \left. \frac{\partial f_o}{\partial P_2(t)} \right|_{ss} P_2(t)$$

$$f_o(t) \approx \bar{f}_o + C_3 P_2(t) \quad \dots 3.8$$

เมื่อ

$$C_2 = r_3 \frac{1}{2} [\bar{P}_2 (\bar{P}_2 - P_{atm})]^{-1/2} (2\bar{P}_2) \quad \dots 3.9$$

Tank 1

$$F_i(t) - F_1(t) = \frac{V_1}{\rho RT} \frac{dP_1(t)}{dt} \quad \dots 3.10$$

แทนค่า  $f_1(t)$

$$F_i(t) - [C_1 P_1(t) + C_2 P_2(t)] = \frac{V_1}{\rho RT} \frac{dP_1(t)}{dt}$$

$$F_i(t) - C_2 P_2(t) = \frac{V_1}{\rho RT} \frac{dP_1(t)}{dt} + C_1 P_1(t)$$

เอา  $C_1$  ทารตลอด

$$\frac{1}{C_1} F_i(t) - \frac{C_2}{C_1} P_2(t) = \frac{V_1}{\rho R T C_1} \frac{dP_1(t)}{dt} + P_1(t)$$

ทำการแปลงลาปลาซ

$$K_1 F_i(s) - K_2 P_2(s) = (\tau_1 s + 1) P_1(s) \quad \dots 3.11$$

$$K_1 = \frac{1}{C_1} \quad K_2 = \frac{C_2}{C_1} \quad \tau_1 = \frac{V_1}{\rho R T C_1}$$

Tank 2

$$F_1(t) - F_o(t) = \frac{V_2}{\rho RT} \frac{dP_2(t)}{dt} \quad \dots 3.12$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แทนค่า  $f_1(t), f_o(t)$

$$C_1 P_1(t) + C_2 P_2(t) - C_3 P_2(t) = \frac{V_2}{\rho RT} \frac{dP_2(t)}{dt}$$

$$C_1 P_1(t) = \frac{V_2}{\rho RT} \frac{dP_2(t)}{dt} + (C_3 - C_2) P_2(t)$$

เอา  $(C_3 - C_2)$  ทารดลอค

$$\frac{C_1}{(C_3 - C_2)} P_1(t) = \frac{V_2}{\rho RT(C_3 - C_2)} \frac{dP_2(t)}{dt} + P_2(t)$$

ทำการแปลงลาปลาซ

$$K_3 P_1(s) = (\tau_2 s + 1) P_2(s) \tag{.....3.13}$$

$$K_3 = \frac{C_1}{(C_3 - C_2)} \quad \tau_2 = \frac{V_2}{\rho RT(C_3 - C_2)}$$

$$f_i = r_1 \sqrt{P_s(P_s - P_1)} \tag{.....3.14}$$

$$f_i \approx \bar{f}_i + \left. \frac{df_i}{dP_1} \right|_{ss} P_1(t)$$

ดังนั้น

$$f_i = \bar{f}_i + C_4 P_1(t)$$

เมื่อ

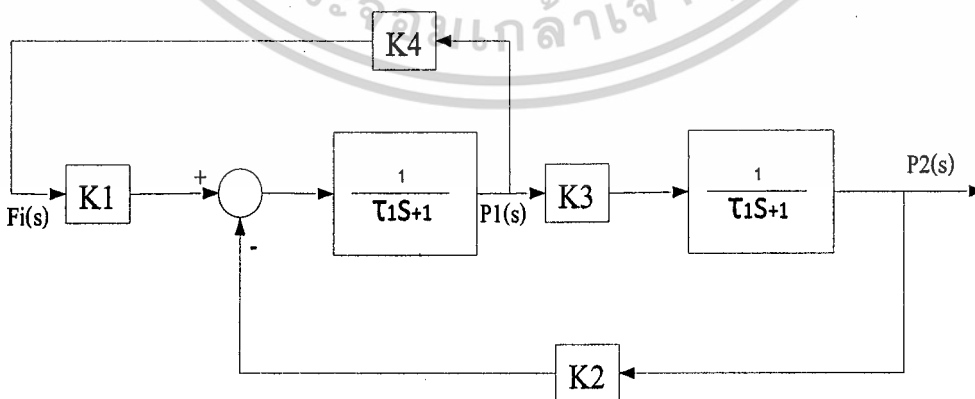
$$C_4 = r_1 [\bar{P}_s(\bar{P}_s - \bar{P}_1)]^{-1/2} (-\bar{P}_1) \tag{.....3.15}$$

$$F_i(t) = C_4 P_1(t)$$

ทำการแปลงลาปลาซ

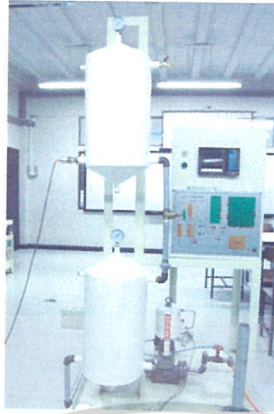
$$F_i(s) = C_4 P_1(s) \tag{.....3.16}$$

$$K_4 = C_4$$



รูปที่ 3.3 Block Diagram ของระบบที่ใช้ในการควบคุม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.4 แสดงส่วนของ Plant ที่ประกอบด้วยอุปกรณ์ต่างๆ

### 3.2 การออกแบบตัวควบคุม

ส่วนของตัวควบคุมเป็นแบบ PID โดยใช้ Microcontroller AMR7 ซึ่งตัวควบคุมสามารถแบ่งออกเป็น 2 ส่วนคือ

#### ส่วนของ Hardware

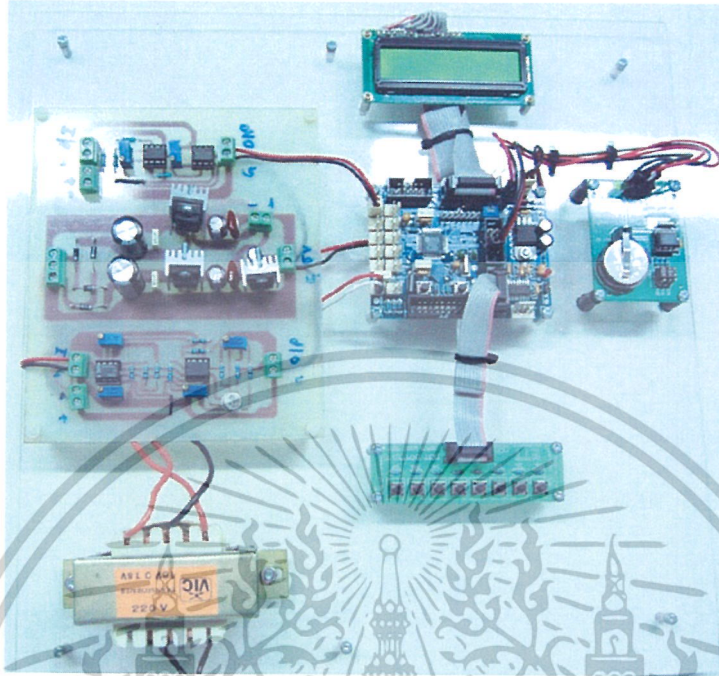
- วงจรแปลงสัญญาณแรงดันไฟฟ้าเป็นสัญญาณกระแสไฟฟ้า (V/I)
- วงจรแปลงสัญญาณกระแสไฟฟ้าเป็นสัญญาณแรงดันไฟฟ้า (I/V)
- วงจรภาคจ่ายไฟ
- การออกแบบ Sw
- การแสดงผลทาง LCD
- การออกแบบจอแสดงผลทาง Monitor

#### ส่วนของ Software

- การออกแบบอัลกอริทึม
- การแปลงอนาลอกเป็นดิจิตอล
- การแปลงดิจิตอลเป็นอนาลอก
- การทำสเกลค่า
- การกำหนดช่วงเวลาของการแซมปลิง (interrupt)

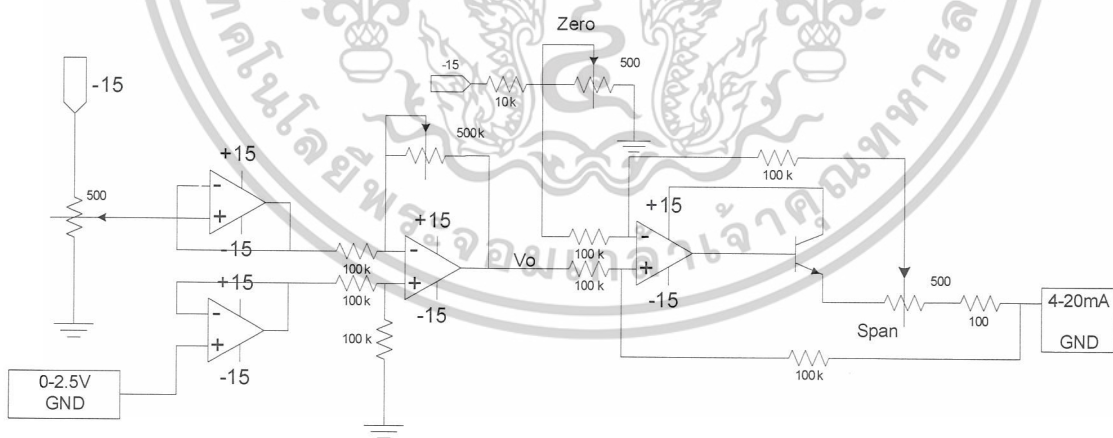
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.2.1 การออกแบบ Hardware



รูปที่ 3.5 แสดงส่วนของฮาร์ดแวร์ตัวควบคุม โดยรวม

#### 3.2.1.1 วงจรแปลงแรงดันไฟฟ้าเป็นกระแสไฟฟ้า (V/I)



รูปที่ 3.6 วงจรแปลงสัญญาณแรงดันไฟฟ้าเป็นสัญญาณกระแสไฟฟ้า

การทำงานของวงจร จากส่วนแรกจะเป็นวงจรขยายสัญญาณร่วมแบบพื้นฐานนั้น ค่าแรงดันเอาต์พุตจะขึ้นอยู่กับผลต่างระหว่างแรงดันอินพุต  $V_1$  และ  $V_2$  ค่าแรงดันอินพุต  $V_1$  และ  $V_2$  มีค่าเท่ากัน จะต้องได้แรงดันเอาต์พุตเท่ากับศูนย์ ค่าความต้านทานแต่ละตัวในวงจรจะต้องมีค่าที่สมนัยกันมาก เพื่อที่จะให้อัตราการขยายของวงจรต่อสัญญาณร่วมเท่ากับศูนย์ จากวงจรส่วนแรกการคำนวณค่าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$V_o = \frac{R_2}{R_1}(V_2 - V_1) \quad \dots 3.17$$

เมื่อมีอินพุตเข้ามาศูนย์โวลต์เราต้องการให้เอาต์พุตออกมา +1 โวลต์ เราจึงใช้ VR มาปรับให้แรงดันไฟลบให้กับวงจรในส่วนแรก  $V_1$  เพื่อเปรียบเทียบกับแรงดัน  $V_2$  ที่เป็นแรงดันจากตัวควบคุมเพื่อให้ได้เอาต์พุต 1 -5 โวลต์ สำหรับวงจรส่วนที่ 1 และวงจรส่วนที่ 2 จะทำหน้าที่แปลงสัญญาณแรงดันไฟฟ้าให้เป็นกระแสไฟฟ้าโดยจะให้เอาต์พุต 4 – 20 มิลลิแอมป์ ซึ่งเมื่อแรงดันจากเอาต์พุตส่วนที่ 1 เข้ามาเป็นอินพุตของวงจรส่วนที่ 2 ซึ่ง VR1จะเป็นการปรับค่าของ Span และ VR2 เป็นการปรับ Zero วงจรนี้เป็นวงจรที่สำคัญวงจรหนึ่งได้มีใช้ในระบบควบคุมกระบวนการทางด้านอุตสาหกรรม ทั้งนี้เนื่องจากบางครั้งอุปกรณ์จำพวกตัวส่งสัญญาณในระบบให้สัญญาณเอาต์พุตเป็นแรงดันไฟฟ้า แต่ถ้าจำเป็นต้องส่งไปเป็นระยะทางที่ไกลมากจึงต้องส่งเป็นกระแสไฟฟ้า

จากวงจรดังรูป 3.6 ต้องการที่จะแปลงสัญญาณแรงดันไฟฟ้าที่แรงดัน 1-5 โวลต์ เพื่อให้เป็นสัญญาณกระแสไฟฟ้า 4-20 มิลลิแอมป์ ซึ่งมีขั้นตอนการคำนวณดังนี้ วงจรขยายความแตกต่างแรงดันไฟฟ้า ดังรูป 3.6 ค่าความต้านทานไฟฟ้าจากรูป ถ้า  $R_1=R_2=R_3=R_4=R$  เราสามารถคำนวณค่าแรงดันเอาต์พุตของวงจรได้โดยใช้ทฤษฎีการทับซ้อน (Superposition)

จะได้

$$V_{out1} = -e_{ref} \quad \dots 3.18$$

$$V_{out} = \left(1 + \frac{R_2}{R_1}\right) VA \quad \dots 3.19$$

ซึ่ง

$$VA = \left(\frac{e_{in} - VL}{2R}\right) R + VL$$

$$V_{out2} = 2VA = 2\left[\left(\frac{e_{in} - VL}{2R}\right) R + VL\right] = e_{in} + VL \quad \dots 3.20$$

$$\text{และ} \quad V_{out} = V_{out1} + V_{out2} = VL + e_{in} - e_{ref} \quad \dots 3.21$$

และจากรูป

$$V_{rs} = V_{out} - VL = VL + e_{in} - e_{ref} - VL = e_{in} - e_{ref}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ดังนั้นกระแสที่ไหลผ่าน  $R_s$  ซึ่งมีค่าเท่ากับกระแสเอาต์พุต คือ

$$i_{rs} = i_0 = \frac{V_{Rs}}{R_s} = \frac{e_{in} - e_{ref}}{R_s} \quad \dots\dots 3.22$$

จากสมการจะเห็นว่ากระแสเอาต์พุตขึ้นอยู่กับค่าแรงดันอินพุต แรงดันอ้างอิง และค่าความต้านทาน  $R_s$

ที่  $e_{in} = 0V; i_0 = 4mA$

$$4mA = \frac{1V - e_{ref}}{R_s}$$

$$e_{ref} = 1 - 4mA \cdot R_s \quad \dots\dots 3.23$$

ที่  $e_{in} = 5V; i_0 = 20mA$

$$20mA = \frac{5V - e_{ref}}{R_s}$$

$$e_{ref} = 5 - 20mA \cdot R_s \quad \dots\dots 3.24$$

แทนสมการที่ (3.23) ลงในสมการที่ (3.24) จะได้

$$R_s = \frac{4}{16 \times 10^{-3}} = 250$$

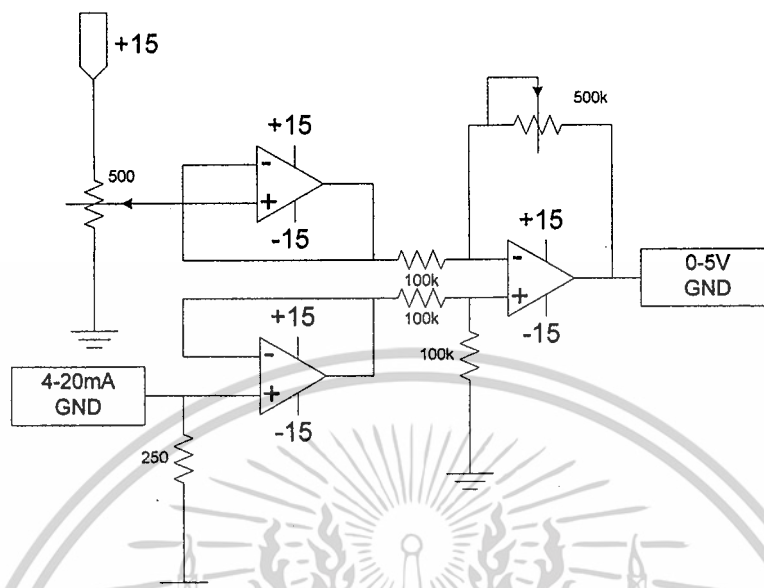
และ  $e_{ref} = 0 - 4mA \times 250 = -1$

ถ้าแทนลงในสมการ

$$i_0 = \frac{e_{in} - e_{ref}}{R_s} = \frac{0 - (-1)}{250} = 4mA$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.2.1.2 วงจรแปลงสัญญาณกระแสไฟฟ้าเป็นแรงดันไฟฟ้า (I/V)



รูปที่ 3.7 วงจรแปลงสัญญาณสัญญาณกระแสไฟฟ้าเป็นแรงดันไฟฟ้า

จากรูป 3.7 การทำงานของวงจรเราจะต่อค่าความต้านทาน 250 โอห์ม อนุกรมกับสัญญาณจากตัวเซนเซอร์เพื่อแปลงสัญญาณที่เป็น 4-20 มิลลิแอมป์ เพื่อให้สัญญาณแรงดันไฟฟ้าที่มีขนาด 1-5 โวลต์ แล้วให้สัญญาณเอาต์พุตเป็น 0-5 โวลต์

จากวงจรดังกล่าวค่าแรงดันเอาต์พุต  $V_o$  จะขึ้นอยู่กับผลต่างแรงดันอินพุต  $V_1$  และ  $V_2$  ถ้าแรงดันระหว่างอินพุต  $V_1$  และ  $V_2$  มีค่าเท่ากันซึ่งมีผลค่าสัญญาณแรงดันเอาต์พุต  $V_o$  จะทำให้มีค่าเท่ากับศูนย์ จะเรียกสัญญาณแรงดันอินพุต  $V_1$  และ  $V_2$  นี้ว่า สัญญาณร่วม ซึ่งค่าความต้านทาน  $R_1$  และ  $R_2$  แต่ละคู่ในวงจรจะต้องมีความสมนัยกันมาก เพื่อให้ได้อัตราขยายของวงจรต่อสัญญาณร่วมเท่ากับศูนย์ ส่วนวงจรถายความแตกต่างที่เพิ่มวงจรตามศักดาหรือวงจรบัฟเฟอร์ (Buffer) เข้ามายังส่วนหน้าของวงจรนี้ เรียกว่า วงจรขยายแบบอินสตรูเมนต์แบบพื้นฐาน

$$V_o = \frac{R_2}{R_1} (V_2 - V_1) \quad \dots\dots 3.25$$

ซึ่ง  $V_2$  จะเท่ากับค่าแรงดันที่ตกคร่อมความต้านทาน 250 โอห์ม แล้ว  $V_1$  จะปรับให้คงที่เท่ากับแรงดัน +1 โวลต์ เพื่อให้จะให้แรงดันเอาต์พุตเป็นไปตามที่เราต้องการ

$$A_V = \frac{V_{out}}{V_2 - V_1} = \frac{R_2}{R_1} \quad \dots\dots 3.26$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

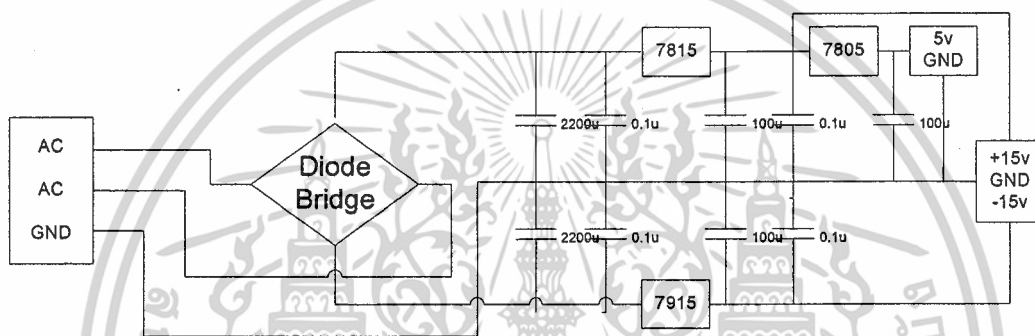
$$1.25R_1 = R_2$$

เลือกค่า  $R_1 = 100\text{ k}\Omega$  เพื่อให้ได้คำตอบของ  $R_1$  และ  $R_2$  เท่ากับ  $A_V$

$$1.25 \times 100\text{ k}\Omega = 125\text{ k}\Omega$$

$$R_2 = 125\text{ k}\Omega$$

### 3.2.1.3 วงจรที่ใช้ในการจ่ายไฟ



รูปที่ 3.8 วงจรภาคแหล่งจ่ายไฟฟ้า

วงจรดังรูปนี้เป็นวงจรแหล่งจ่ายไฟฟ้ากระแสตรงซึ่งสามารถจ่ายไฟฟ้ากระแสตรงได้ทั้งแรงดันไฟฟ้าบวกและไฟฟ้านลบ อีกทั้งยังมีอีกหลายระดับ ซึ่งมีแรงดัน +15V -15V การทำงานของวงจรไฟฟ้ากระแสสลับ 220V หม้อแปลงทำหน้าที่ปรับเปลี่ยนแรงดันไฟฟ้ากระแสสลับ 220V ลดลงเหลือ 18 V

$$V_{dc} = 0.636 \times V \quad \dots 3.27$$

กรณีพิจารณาไดโอด

$$V_{dc} = 0.636 \times (V_p - 1.4) \quad \dots 3.28$$

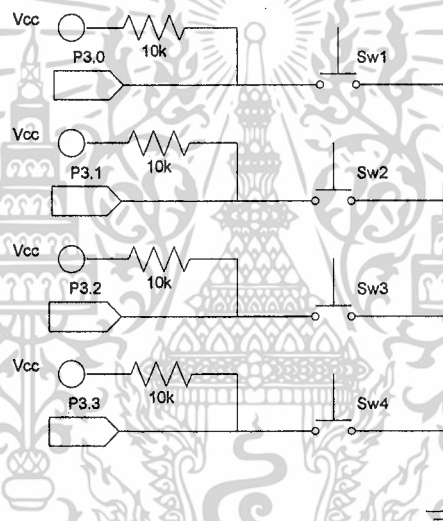
$$V_p = \sqrt{2} \times V_{rms} \quad \dots 3.29$$

เอกสารนี้เป็นเอกสารที่  $V_{dc} = 0.636 \times \sqrt{2} \times 18$  ซึ่งใช้ในการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์อื่นใด การค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$= 16.2V$$

เมื่อหม้อแปลงปรับลดแรงดันไฟฟ้ากระแสสลับแล้ว D1–D4 ที่ต่อเป็นวงจรบริดจ์ เรกติไฟร์ จะทำหน้าที่เรียงไฟฟ้ากระแสสลับให้เป็นไฟฟ้ากระแสตรง จากนั้น C1, C2, C3, และ C4 จะทำหน้าที่กรองกระแสให้เรียบ แล้วผ่านเข้าไปยังขา 1 (Vin) ของ IC1 และขา 2 (Vin) ของ IC2 โดยการทำงานของ IC1 จะเรกติฟายแรงดันให้คงที่ +15 โวลต์ และ IC2 จะเรกติฟายแรงดันไฟฟ้าให้คงที่ -15 โวลต์ แล้วจะถูกกรองให้เรียบยิ่งขึ้น โดย C5, C6, C7 และ C8 สำหรับ IC3 จะเรกติฟายแรงดันให้คงที่ +5 โวลต์ แล้วจะถูกกรองโดย C9 ให้เรียบยิ่งขึ้น

### 3.2.1.4 การออกแบบ Switch



รูปที่ 3.9 สวิตช์ต่างๆ ที่ใช้ในไมโครคอนโทรลเลอร์ ARM7

หน้าที่การทำงานของสวิตช์ต่างๆ

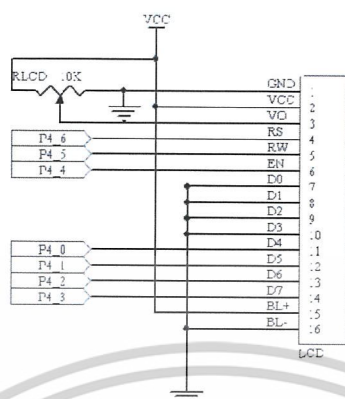
SW1 ทำหน้าที่เป็นปุ่ม Enter เลือกเข้าฟังก์ชัน

SW2 ทำหน้าที่เป็นปุ่ม ใช้ในการเลื่อนฟังก์ชันขึ้น, เพิ่มค่าของพารามิเตอร์ต่างๆ

SW3 ทำหน้าที่เป็นปุ่ม ใช้ในการเลื่อนฟังก์ชันลง, ลดค่าของพารามิเตอร์ต่างๆ

SW4 ทำหน้าที่เป็นปุ่ม Exit ออกจากฟังก์ชัน

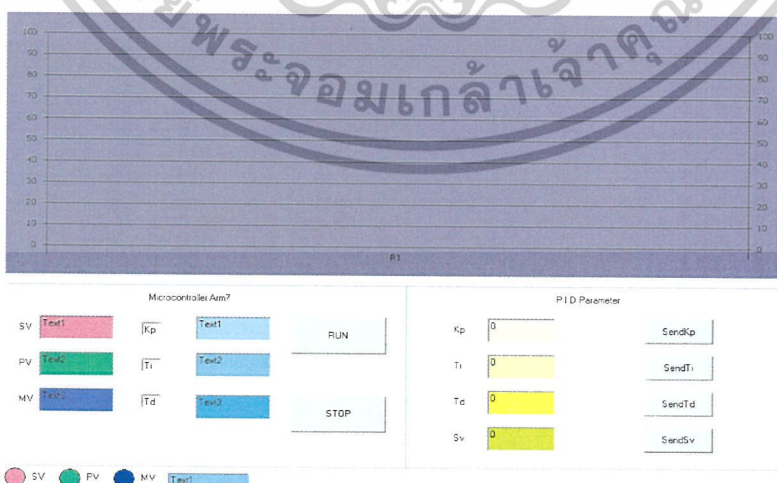
### 3.2.1.5 การแสดงผลทาง LCD



รูปที่ 3.10 Schematic ของการต่อจอแสดงผล LCD กับไมโครคอนโทรลเลอร์ ADuC7024

จากรูป Schematic ของการต่อจอแสดงผล LCD เข้ากับตัวไมโครคอนโทรลเลอร์เบอร์ ADuC 7024 ซึ่งจะเห็นได้ว่าเป็นการต่อแบบ I/O port และอีกการทำงานของจอแสดงผล LCD อยู่ในโหมด 4 บิตซึ่งจะต่อใช้งานที่ D7-D4 หน้าทีของตัวแสดงผล LCD เพื่อให้แสดงฟังก์ชันการทำงานของตัวควบคุมให้รู้ว่าตัวควบคุมอยู่ในฟังก์ชันการทำงานอะไร และแสดงค่าตัวเลขต่าง ๆ ในแต่ละฟังก์ชัน

### 3.2.1.6 การออกแบบจอแสดงผลทาง Monitor



รูปที่ 3.11 จอแสดงผลที่ใช้ในการทดลอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นิยมนำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หน้าจอแสดงผลแบ่งออกเป็น 3 ส่วนคือ

1. ส่วนที่หนึ่งเป็นกราฟไว้แสดงค่าของ SV, PV และ MV
2. ส่วนที่สอง Microcontroller Arm7 ไว้สำหรับแสดงตัวเลขของค่าพารามิเตอร์ต่าง ๆ ซึ่งมีอยู่ 6 ค่า
3. ส่วนที่สาม PID Parameter สำหรับการส่งค่าพารามิเตอร์ให้กับตัวควบคุม

### 3.3 การออกแบบ Software

#### 3.3.1 การออกแบบอัลกอริทึมสำหรับตัวควบคุม PID

ในการออกแบบอัลกอริทึมของตัวควบคุม จะเป็นการใช้สัญญาณเอาต์พุตที่คำนวณได้เป็นเวลาระยะสั้น ๆ แทนที่จะเป็นเวลาที่มีความต่อเนื่อง สมการ PID ทั่ว ๆ ไปคือ

$$M(t) = K_c \left\{ e(t) + \frac{1}{T_i} \int e(t) dt + T_d \frac{de(t)}{dt} \right\} \quad \dots\dots 3.31$$

เนื่องจากสัญญาณที่คอนโทรลเลอร์นำมาคำนวณนี้เป็นสัญญาณแบบดิจิตอล ดังนั้นสมการในการคำนวณ PID เดิมที่ใช้สำหรับสัญญาณอนาล็อกจึงไม่สามารถนำมาเขียนโปรแกรมคำนวณได้โดยตรง จึงพิจารณาดังนี้

สัญญาณควบคุมจากตัวควบคุม PID สามารถคำนวณได้จาก  
โดย

$$e(t) = SV(t) - PV(t) \quad \dots\dots 3.32$$

$$\int e(t) dt = \sum_{k=0}^n e(k) \Delta t \quad \dots\dots 3.33$$

$$\frac{de(t)}{dt} = \frac{[e(t) - e(t_1)]}{\Delta t} \quad \dots\dots 3.34$$

เมื่อรวมสมการเข้าด้วยกันจะได้ว่า สัญญาณควบคุมที่เป็นดิจิตอลสามารถคำนวณได้จาก  
สมการ

$$M(t) = K_c \left[ e(t) + \frac{1}{T_i} \sum_{k=0}^n e(k) \Delta t + \frac{e(t) - e(t_1)}{\Delta t} \right] \quad \dots\dots 3.35$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อ  $\Delta t$  = ช่วงเวลาของการ Sampling

$M(t)$  = ค่าของสัญญาณควบคุมแต่ละช่วงเวลา

$e(t)$  = ค่าความคลาดเคลื่อนของแต่ละช่วงเวลา

$e(t_1)$  = ค่าความคลาดเคลื่อนของแต่ละเวลาที่ผ่านมาแล้ว

ขั้นตอนการเขียนโปรแกรมการประมวลผลการควบคุมแบบพีไอดีในรูปแบบดิจิทัล เราสามารถเขียนสมการขึ้นมาใหม่ได้ดังนี้

$$e(kT) = SV(kT) - PV(kT)$$

$$P(kT) = K_c \times [SV(kT) - PV(kT)]$$

$$I(kT) = \frac{K_c \times T_s}{T_i} [e(kT) + e(kT - T)]$$

$$D(kT) = \frac{K_c \times T_d}{T_s} [e(kT) - e(kT - T)]$$

$$M(kT) = P(kT) + I(kT) + D(kT)$$

$$e(kT - T) = e(kT)$$

### 3.3.2 การแปลงสัญญาณอนาลอกเป็นดิจิทัล

ภายในไมโครคอนโทรลเลอร์จะมีวงจรแปลงอนาลอกเป็นดิจิทัล มีความละเอียดขนาด 12 บิตแบบ successive approximation โดยมีความเร็วสูงสุดที่ใช้ในการแปลงสัญญาณได้สูงถึง 1 MSPS ที่ความถี่ของสัญญาณพิก้า 41.78 Hz ADUC7024 มี A/D ทั้งหมด 10 channel

โหมดการทำงานของ A/D มีด้วยกัน 3 โหมด

1. Fully differential mode
2. Pseudo Differential mode
3. Single – Ended mode

อินพุตอยู่ในช่วง  $0 - V_{REF}$  เอาต์พุตจะเป็นรหัสเลขฐาน 2 คือ

$$1 \text{ LSB} = FS/4096,$$

$$\text{or } 2.5 \text{ V}/4096 = 0.61 \text{ mV},$$

$$\text{or } 610 \mu\text{V when } V_{REF} = 2.5 \text{ V}$$

ในการออกแบบตัวควบคุมได้เลือกใช้แบบ Single – Ended mode ย่านของการใช้

งานที่ใช้สามารถรับอินพุตจากภายนอกเข้ามาจะอยู่ในช่วงแรงดัน 0V ถึง  $V_{REF}$  ในการกำหนดค่า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เริ่มต้นเพื่อให้ Microcontroller ARM7 เบอร์ ADuC7024 ทำหน้าที่แปลงสัญญาณอนาลอกให้เป็นดิจิทัล ซึ่งสามารถทำได้ดังนี้

ส่วนของการกำหนดการทำงาน ADC

```
ADCCON = 0x00000000;           // Reset ADC Config
ADCCON |= 0x00000020;         // Power-ON ADC Function
delay(1000);                   // Wait ADC Power-on Ready
ADCCON |= 0x00001400;         // ADC Clock = fADC/32
ADCCON |= 0x00000300;         // Acquisition Time = 16 Cycle Clock
ADCCON &= 0xFFFFF7;          // ADC = Single-End Mode
ADCCON |= 0x00000004;         // Continue Software Convert
REFCON = 0x00000001;          // Used Internal 2.5V Reference
ADCCON |= 0x00000080;         // ADC Start Conversion
```

ส่วนของการแปลงอนาลอกเป็นดิจิทัลของ MCU

```
ADCCP = 0;                     // Select Channel 0 to Conversion
delay(1000);                   // Wait Select Channel Ready
while (!ADCSTA){};            // Wait ADC Conversion Complete (Bit0="1")
val = (ADCDAT >> 16) & 0x00000FFF; // Shift ADC Result to Integer
pv = (100 * val) / 2047;
```

เมื่ออินพุต 0V ให้สัญญาณที่เป็นดิจิทัลมีค่า 0x000H

เมื่ออินพุต 2.5V ให้สัญญาณดิจิทัลมีค่า 0xFFFFH

สัญญาณแปลงจากอนาลอกให้เป็นดิจิทัลก็คือ สัญญาณที่วัดได้จากกระบวนการ (PV) แต่ในการออกแบบเมื่ออินพุต 1.25 V ให้สัญญาณดิจิทัลมีค่า 0x7FFH เนื่องจากสัญญาณที่ได้จากที่ตัวเซ็นเซอร์ไม่สามารถปรับให้มีย่านกว้างได้

### 3.3.3 การแปลงสัญญาณดิจิทัลเป็นอนาล็อก

สำหรับ DAC ใน Microcontroller ARM7 เบอร์ ADuC7024 ซึ่งมีจำนวน 2 ช่องอินพุตเป็นสัญญาณอนาลอกขนาด 12 บิตขนาดแรงดันเอาต์พุตอยู่ที่  $0-V_{REF}$ ,  $0-A_{VDD}$ , และ  $0-DAC_{REF}$  แล้วแต่จะเลือกใช้ ในการออกแบบโปรเจกต์นี้ใช้ย่าน  $0-V_{REF}$  และ Channel 0 ซึ่งกำหนดค่าเริ่มต้นในการทำงานได้ดังนี้

การกำหนดค่าเริ่มต้นการทำงาน DAC

```
DAC0CON &= 0xDF;              // DAC0 Used System Clock
DAC0CON |= 0x10;              // Enable DAC0
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
DAC0CON |= 0x02;           // DAC0 Output Range = +Vref..AGND
REFCON = 0x01;           // Used Internal 2.5V Reference
```

การส่งค่าข้อมูลออกสู่ DAC Channel 0

```
DAC0DAT = (mv << 16);     // Update DAC0 Output (0-2.5V)
```

### 3.3.4 การทำสเกล

ในการสร้างตัวควบคุมแบบ PID นั้นสัญญาณที่ได้เป็นสัญญาณดิจิทัลจึงมีความจำเป็นต้องทำการ Scaling ของค่าตัวแปรต่าง ๆ เพื่อให้อยู่ในหน่วยการคำนวณเดียวกันและเข้าใจในการออกแบบโปรเจกต์นี้ได้ทำการ Scaling ให้อยู่ในหน่วยของเปอร์เซ็นต์ (0-100%) ทำได้ดังนี้

การ Scaling ตัวแปร Sp

Sp กำหนดให้ปรับค่าได้จาก 0 ไปจนถึง 100

การ Scaling ตัวแปร PV

```
val = (ADCDAT >> 16) & 0x00000FFF; // Shift ADC Result to Integer
```

```
if (val > 2047) val = 2047;
```

```
pv = (100 * val) / 2047;
```

การ Scaling ตัวแปร MV

```
mv = P + I + D; // calculate PID output
```

```
if (mv > 4095) { mv = 4095; I = I1; }
```

```
mv_p = (mv * 100) / 4095;
```

```
if (mv < 0) { mv = 0; I = I1; }
```

```
mv_p = (mv * 100) / 4095;
```

ตัวแปร mv เป็นตัวแปรที่ส่งสัญญาณดิจิทัลเป็นอนาล็อก

ตัวแปร mv\_p เป็นตัวแปรที่ส่งออกไปแสดงผล

### 3.3.5 การกำหนดช่วงเวลาของการแซมปลิง (interrupt)

การออกแบบตัวควบคุมการสุ่มข้อมูลควบคุมจากระบบคือ การเลือกช่วงเวลาเพื่อสำหรับการสุ่มค่าให้มีความเหมาะสมที่ดี เพื่อให้สัญญาณที่คอนโทรลเลอร์ได้รับและส่งออกป้อนมีค่าใกล้เคียงกับตัวควบคุมที่มีความต่อเนื่องของเวลา การออกแบบช่วงเวลาแซมปลิง เลือกช่วงเวลาเท่ากับ 250 msec ซึ่งในการกำหนดให้ไมโครคอนโทรลเลอร์แซมปลิงทุก ๆ 250 msec นั้น จะต้องมี การกำหนดการทำงานในส่วนของ Interrupt ซึ่งเลือกใช้ Timer1 ทำงานโหมด periodic (okresowy)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในกรณีนี้ Timer จะได้เริ่มนับจากค่าสูงสุดใน Load Register (TxLD) ลงมาจนถึงค่าที่ต่ำสุด (timer count-down) จากนั้นจะเริ่มนับจากค่าต่ำสุดใน Load Register (TxLD) ไปจนถึงค่าสูงสุด (timer count-up)

สำหรับโหมด Periodic จะทำงานตามสมการ

$$Interval = \frac{(TxLD)Prescaler}{SourceClock} \quad \dots\dots 3.36$$

$$(TxLD) = \frac{Interval(SourceClock)}{Prescaler} \quad \dots\dots 3.37$$

เมื่อ Interval = 250 ms

SourceClock = HCLK = 41.78 MHz

Prescaler = / 1

แทนค่าในสมการจะได้

$$(TxLD) = \frac{250ms(41.78MHz)}{Prescaler} = 10445042 \text{ Cycle}$$

กำหนด TILD = 10445042; ในโปรแกรมก็จะอินเทอร์รัพท์ ทุกๆ 250 ms

## บทที่ 4

### ผลการทดลอง

#### 4.1 อุปกรณ์ที่ใช้ในการทดลอง

1. Microcontroller ARM7
2. Pressure Sensor
3. Pressure Gauge
4. Control Valve
5. ภาชนะบรรจุความดันแบบ 2 ถังปิด
6. แหล่งจ่ายลมและสายลม

#### 4.2 ลำดับขั้นเตรียมการทดลองควบคุม

1. ต่อสายลมจากแหล่งจ่ายลมเข้ากระบวนการ
2. ติดตั้ง Pressure Gauge และ Pressure Sensor และตรวจสอบความถูกต้อง
3. ติดตั้ง Control Valve และตรวจสอบความถูกต้อง
4. ต่อเกทวาล์วเข้ากับกระบวนการและ Control Valve เพื่อจะใช้เปิด-ปิดของลมที่จะเข้าไปยังกระบวนการ
5. ติดตั้งเครื่องควบคุม และตรวจสอบความถูกต้อง
6. ตรวจสอบความดันแหล่งจ่าย ให้มีค่าประมาณ 25 ปอนด์ต่อตารางนิ้ว
7. เปิดลมเข้ามายังเกทวาล์วที่ต่ออยู่กับ Control Valve
8. เปิดลมให้ไหลเข้าถังทั้ง 2 ถังเพื่อตรวจสอบรอยรั่ว

### 4.3 ขั้นตอนการทดลองสำหรับควบคุมกระบวนการความดัน

#### 4.3.1 การทดลองวงจรปรับระดับสัญญาณ (V/I) และ (I/V)

1. วงจรปรับระดับสัญญาณขนาด 0-2.5 V เป็นสัญญาณมาตรฐานขนาด 4-20 mA จากการปรับให้เป็นไปตามเชิงเส้นซึ่งมีผลการทดลอง ดังแสดงตารางที่ 4.1

ตารางที่ 4.1 แสดงความสัมพันธ์ระหว่างสัญญาณอินพุต 0-2.5 V กับสัญญาณเอาต์พุต 4-20 mA

แรงดัน (V)	กระแส (mA)
0	4.00
0.12	4.85
0.22	5.48
0.31	6.10
0.41	6.72
0.51	7.34
0.60	7.97
0.70	8.60
0.80	9.22
0.92	10.01
1.00	10.64
1.10	11.27
1.21	11.90
1.31	12.53
1.40	13.16
1.50	13.79
1.60	14.42
1.70	15.05
1.82	15.83
1.90	16.31
2.00	17.09
2.10	17.56
2.21	18.35
2.30	18.97
2.40	19.60
2.50	20.03

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. วงจรปรับสัญญาณมาตรฐานขนาด 4-20 mA เป็นสัญญาณขนาด 0-5 V จากการปรับให้ เป็นไปตามเชิงเส้นซึ่งมีผลการทดลอง ดังแสดงตารางที่ 4.2

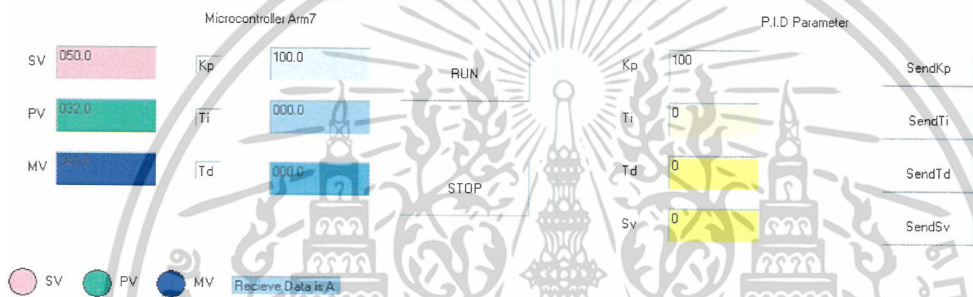
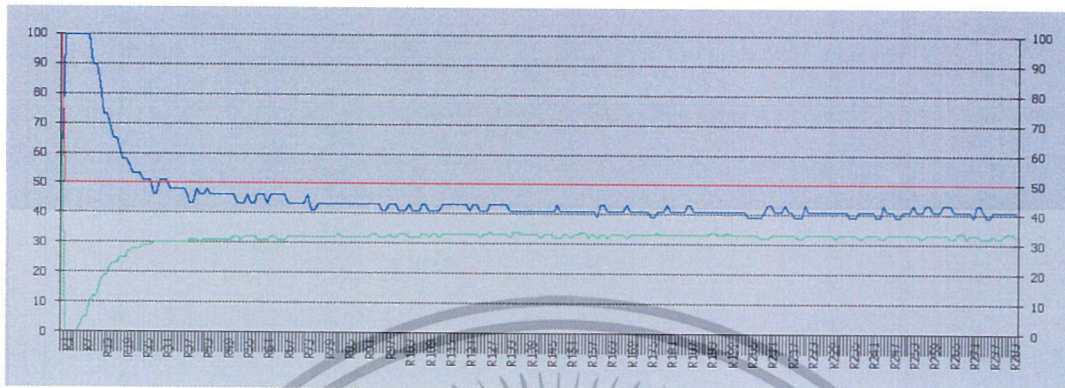
ตารางที่ 4.2 แสดงความสัมพันธ์ระหว่างสัญญาณอินพุต 4 – 20 mA กับ สัญญาณเอาต์พุต 0–5 V

กระแส (mA)	แรงดัน (V)
4	0
5	0.296
6	0.590
7	0.930
8	1.223
9	1.565
10	1.858
11	2.153
12	2.497
13	2.791
14	3.070
15	3.410
16	3.700
17	4.040
18	4.340
19	4.630
20	4.930

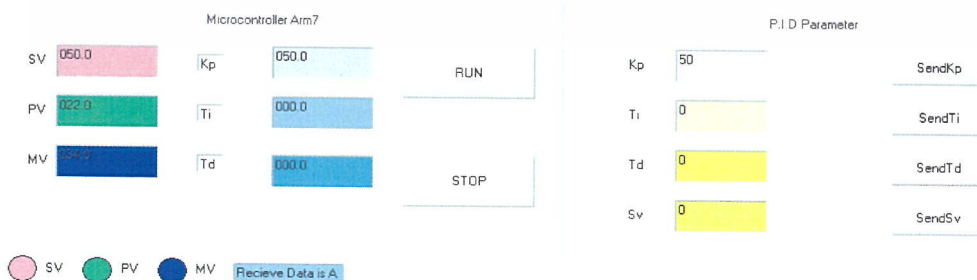
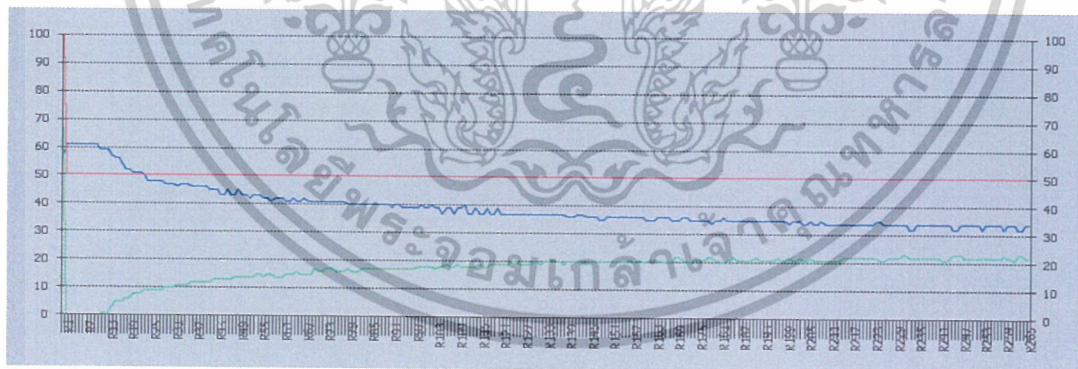
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 4.3.2 การทดลองการควบคุมแบบ Proportional

เมื่อทำการเปิดวาล์วตัวที่ 1 ที่ Stoke 100 % และเปิดวาล์วตัวที่ 2 ที่ Stoke 25 %



รูปที่ 4.1 แสดงผลตอบสนองการควบคุมแบบ Proportional เมื่อค่า  $K_p = 100$

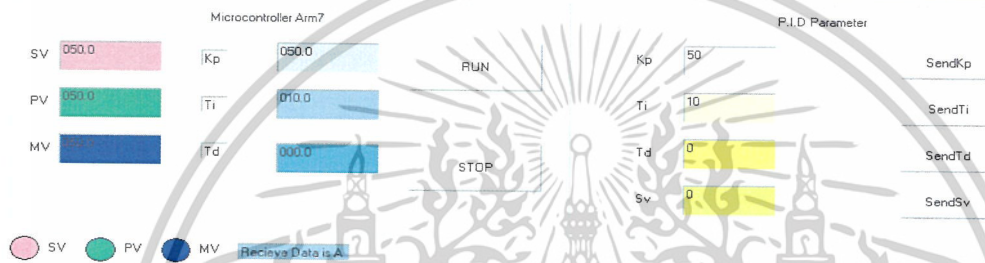
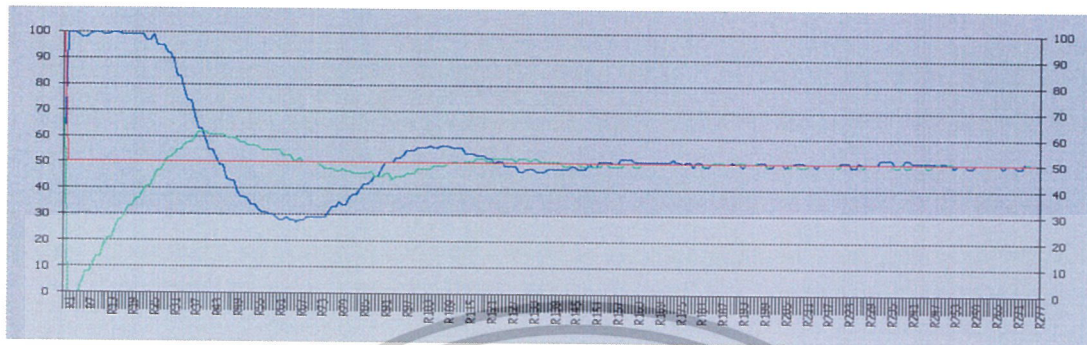


รูปที่ 4.2 แสดงผลตอบสนองการควบคุมแบบ Proportional เมื่อค่า  $K_p = 50$

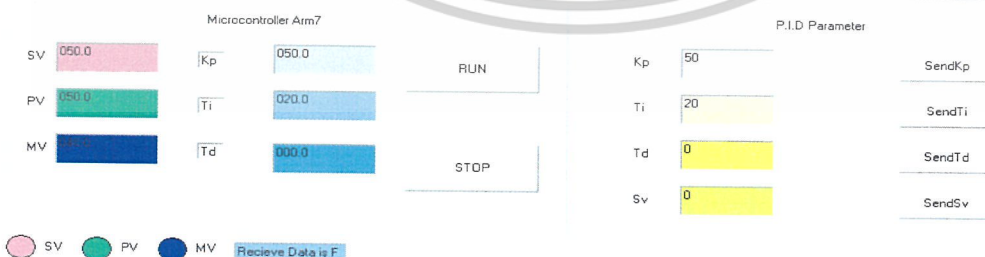
เอกสารนี้เป็นเอกสารที่สวอนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่อนุญาตให้เข้าไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 4.3.3 การทดลองการควบคุมแบบ Proportional และ Integral

เมื่อทำการเปิดวาล์วตัวที่ 1 ที่ Stoke 100 % และเปิดวาล์วตัวที่ 2 ที่ Stoke 25 %

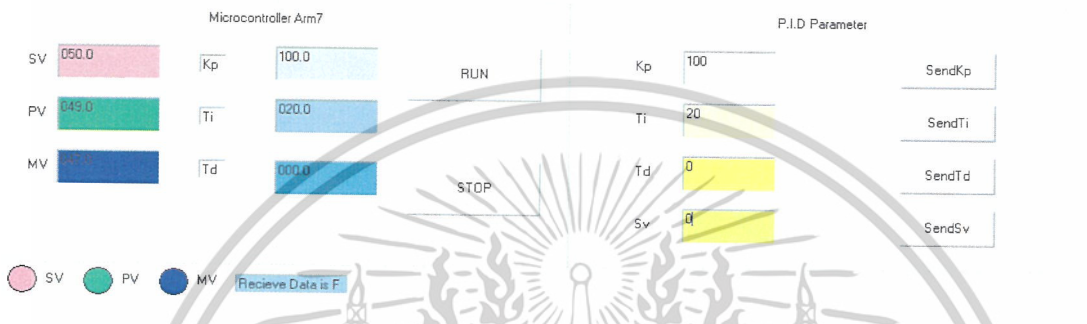
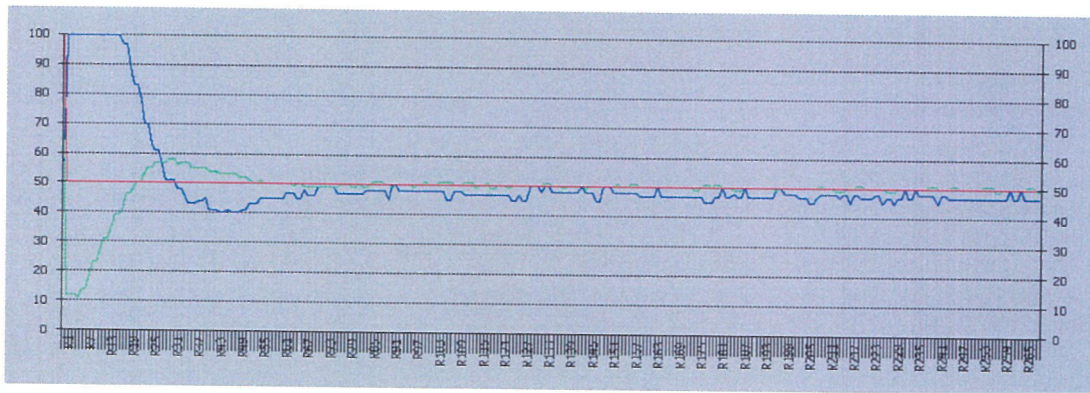


รูปที่ 4.3 แสดงผลตอบสนองของการควบคุมแบบ Proportional + Integral  
เมื่อค่า  $K_p = 50$ ,  $T_i = 10$

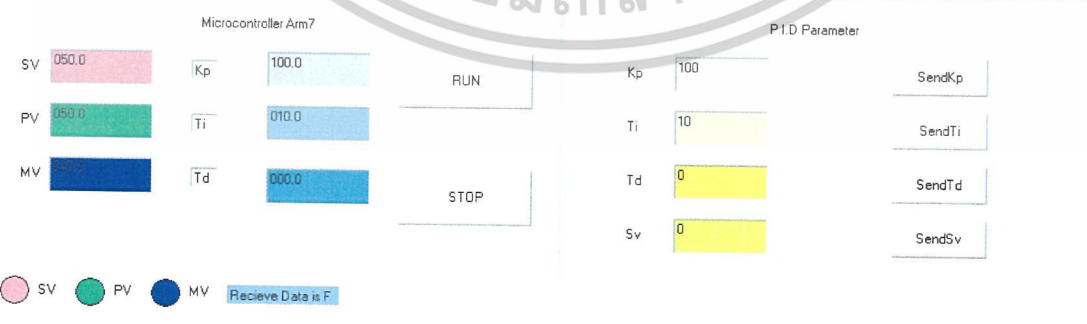
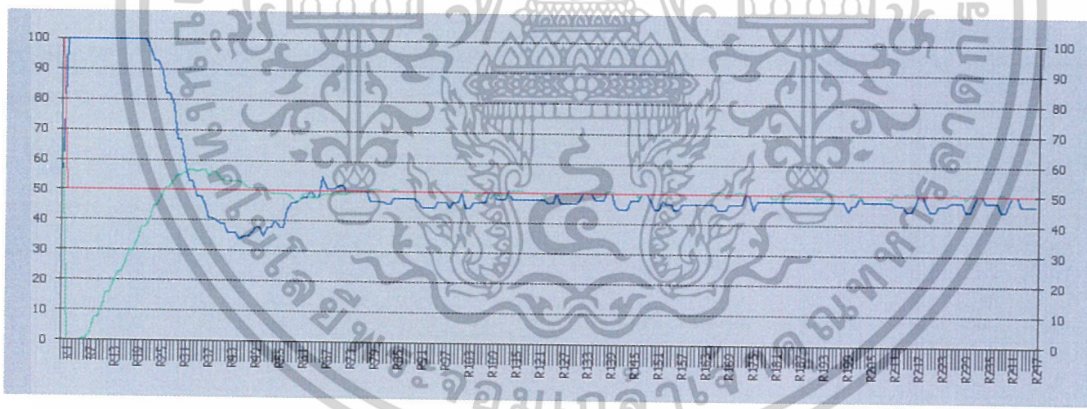


รูปที่ 4.4 แสดงผลตอบสนองของการควบคุมแบบ Proportional + Integral  
เมื่อค่า  $K_p = 50$ ,  $T_i = 20$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.5 แสดงผลตอบสนองการควบคุมแบบ Proportional + Integral  
เมื่อค่า  $K_p = 100$ ,  $T_i = 20$

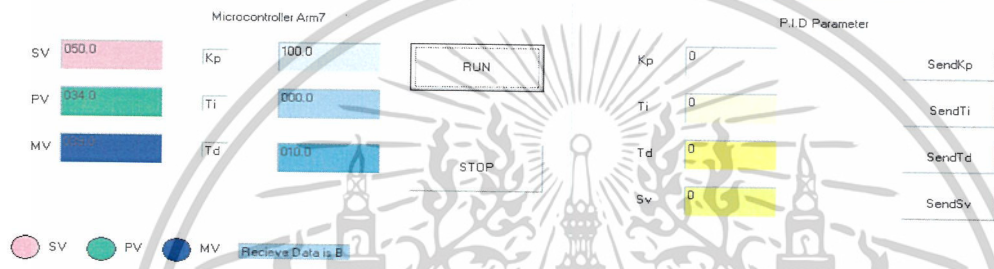
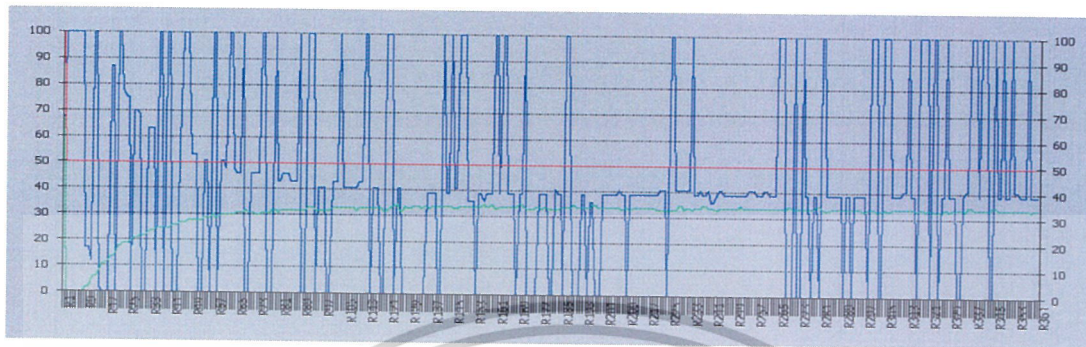


รูปที่ 4.6 แสดงผลตอบสนองการควบคุมแบบ Proportional + Integral  
เมื่อค่า  $K_p = 100$ ,  $T_i = 10$

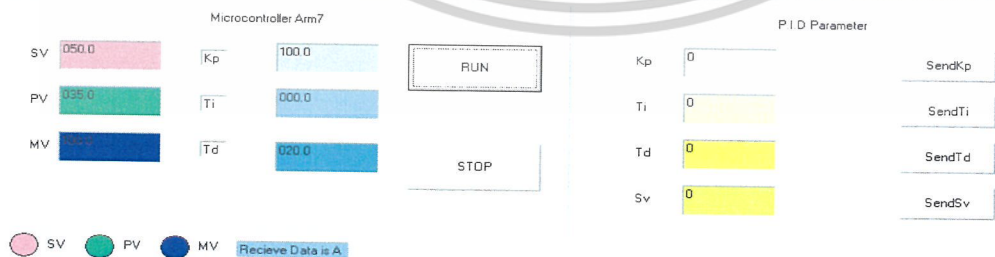
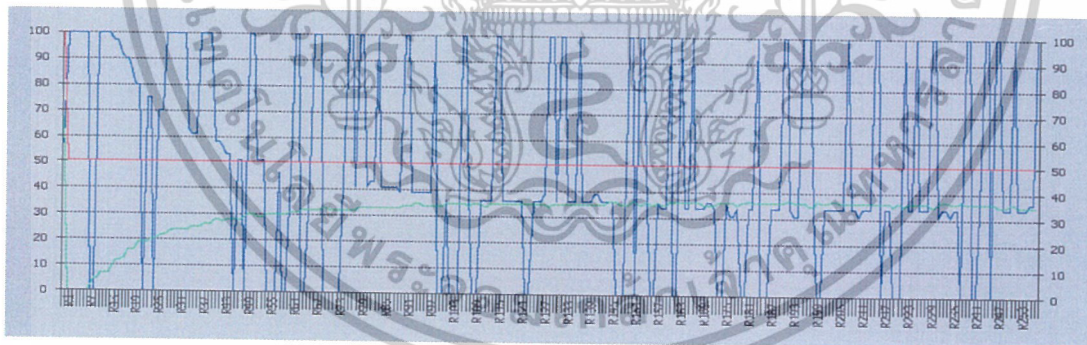
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 4.3.4 การทดลองการควบคุมแบบ Proportional และ Derivative

เมื่อทำการเปิดวาล์วตัวที่ 1 ที่ Stoke 100 % และเปิดวาล์วตัวที่ 2 ที่ Stoke 25 %

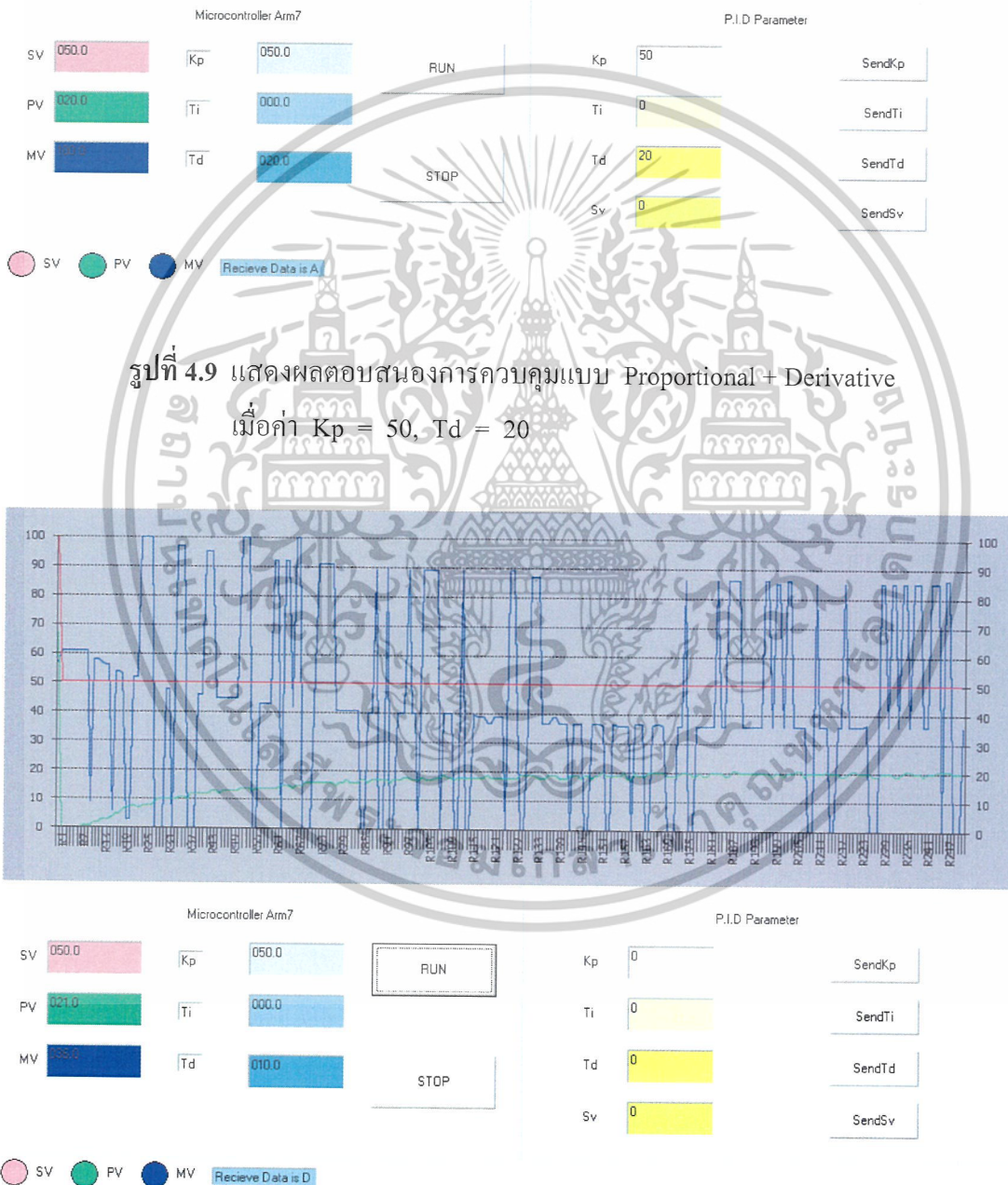
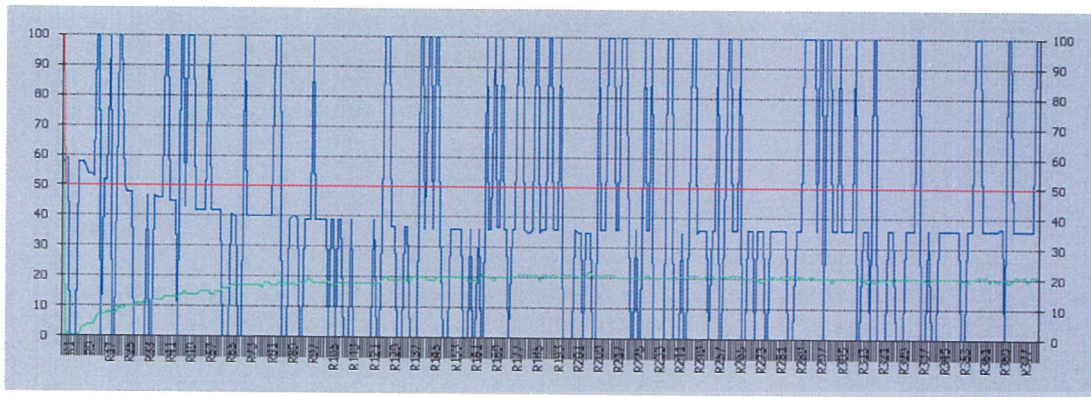


รูปที่ 4.7 แสดงผลตอบสนองของการควบคุมแบบ Proportional + Derivative เมื่อค่า  $K_p = 100$ ,  $T_d = 10$

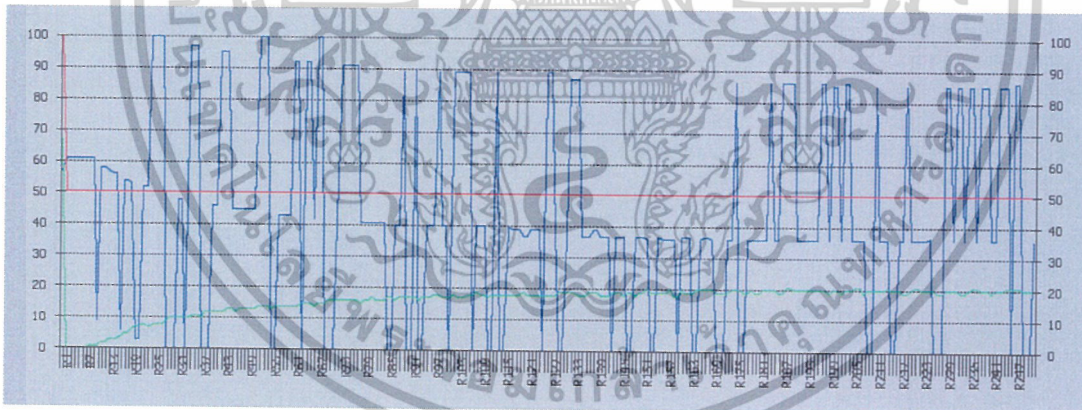


รูปที่ 4.8 แสดงผลตอบสนองของการควบคุมแบบ Proportional + Derivative เมื่อค่า  $K_p = 100$ ,  $T_d = 20$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.9 แสดงผลตอบสนองการควบคุมแบบ Proportional + Derivative  
เมื่อค่า  $K_p = 50$ ,  $T_d = 20$

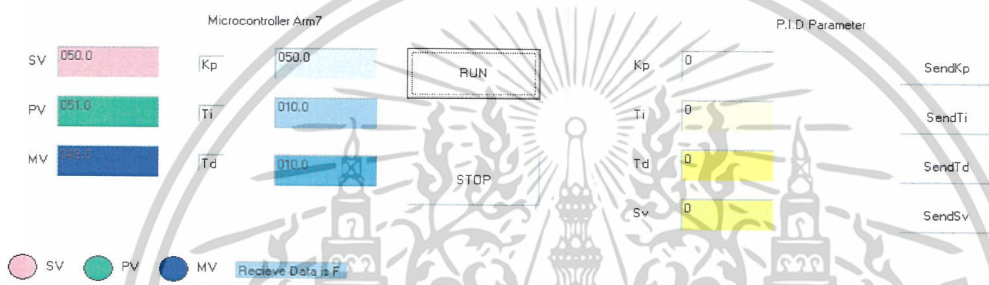
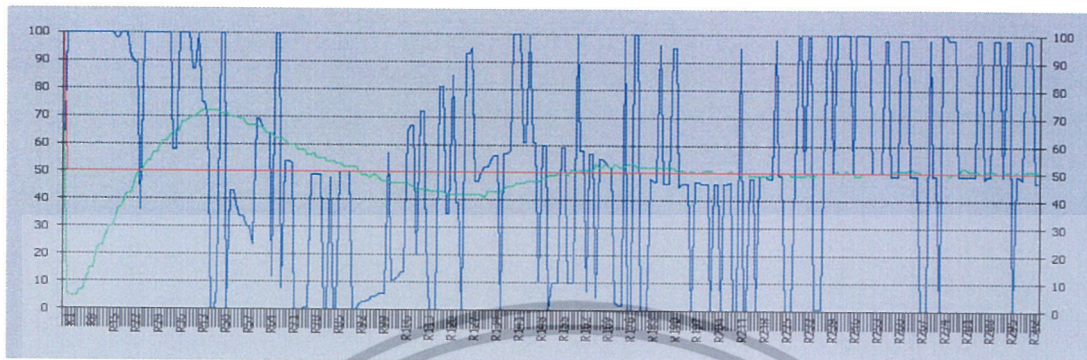


รูปที่ 4.10 แสดงผลตอบสนองการควบคุมแบบ Proportional + Derivative  
เมื่อค่า  $K_p = 50$ ,  $T_d = 10$

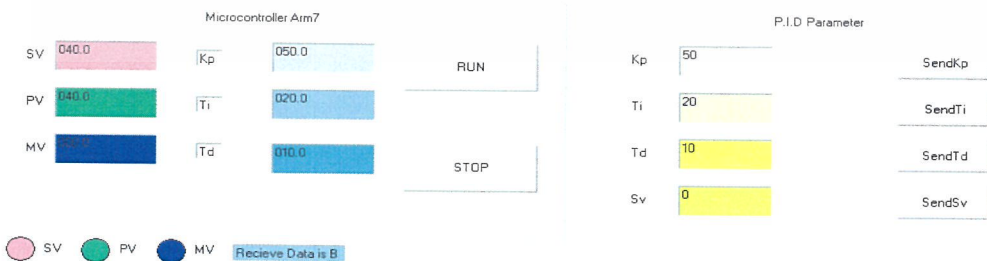
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้เพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 4.3.5 การควบคุมแบบ Proportional และ Integral และ Derivative

เมื่อทำการเปิดวาล์วตัวที่ 1 ที่ Stoke 100 % และเปิดวาล์วตัวที่ 2 ที่ Stoke 25 %



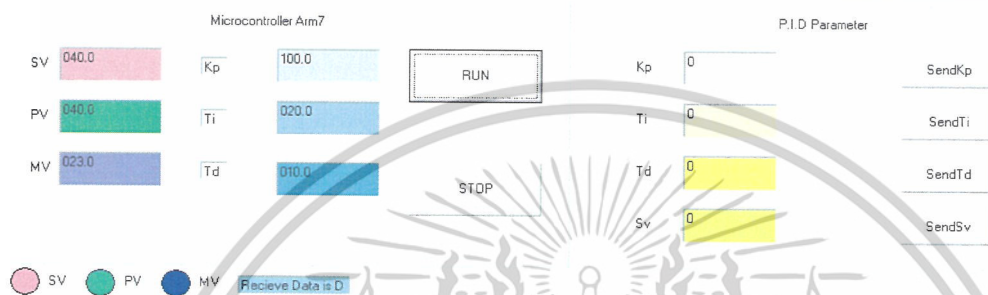
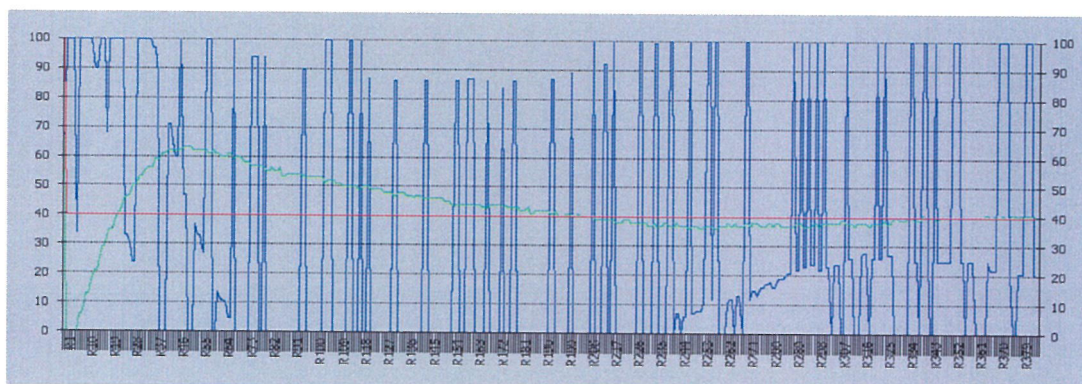
รูปที่ 4.11 แสดงผลตอบสนองการควบคุมแบบ Proportional + Integral + Derivative  
เมื่อค่า  $K_p = 50$ ,  $T_i = 10$ ,  $T_d = 10$



รูปที่ 4.12 แสดงผลตอบสนองการควบคุมแบบ Proportional + Integral + Derivative

เมื่อค่า  $K_p = 50$ ,  $T_i = 20$ ,  $T_d = 10$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

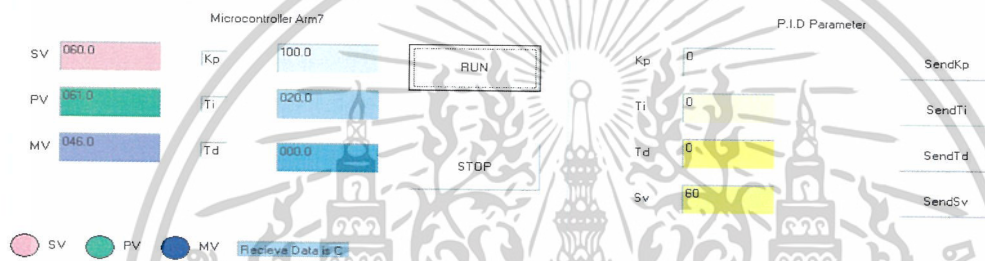
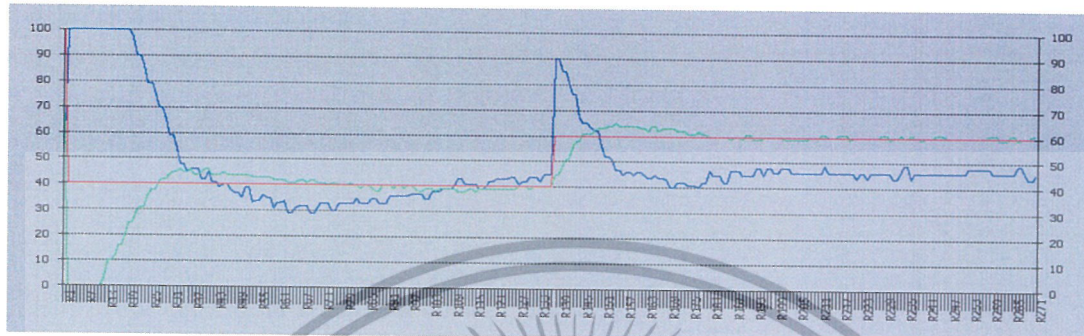


รูปที่ 4.13 แสดงผลตอบสนองการควบคุมแบบ Proportional + Integral + Derivative  
เมื่อค่า  $K_p = 100$ ,  $T_i = 20$ ,  $T_d = 10$

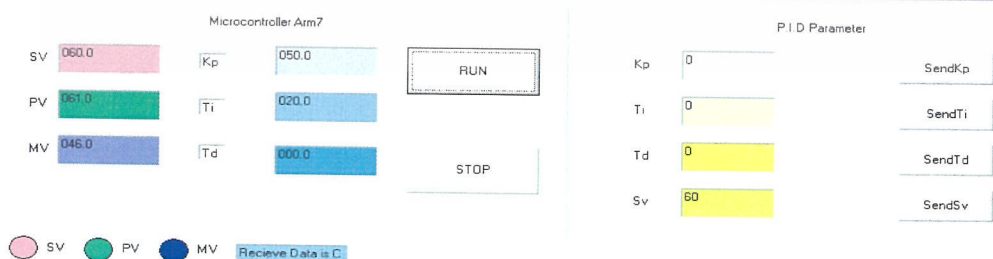
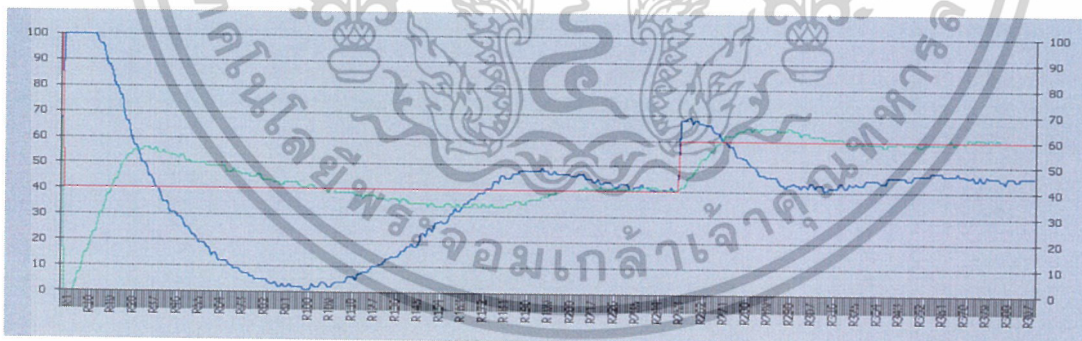
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

#### 4.4 การปรับ PID แบบวิธี Trial and Error

ในการทดลองปรับค่าพารามิเตอร์แบบวิธี Trial and Error เราจะทำการสุ่มตัวอย่างจากการทดลองโดยจะทำการปรับค่าของ  $K_p$  และ  $T_i$  ไปเรื่อย ๆ ตามความเหมาะสม



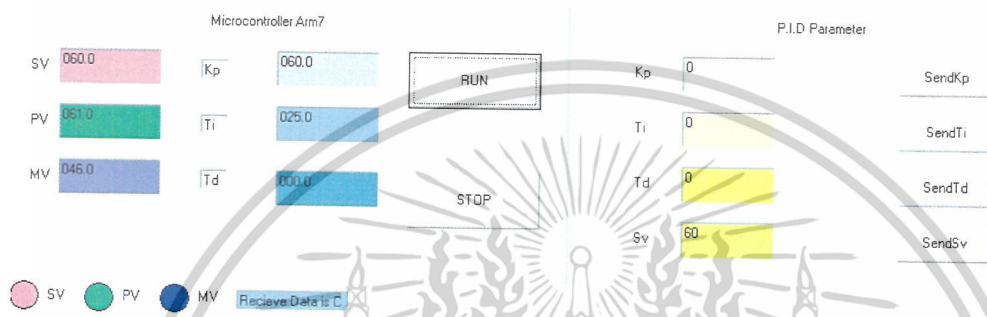
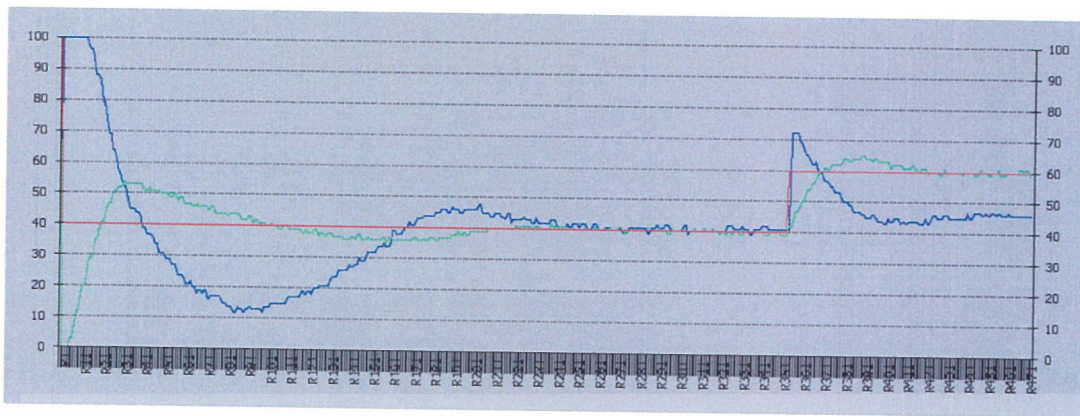
รูปที่ 4.14 สัญญาณควบคุมที่ได้จากตัวควบคุมแบบ Proportional + Integral ด้วยวิธี Trial and Error เมื่อค่า  $K_p = 100$ ,  $T_i = 20$



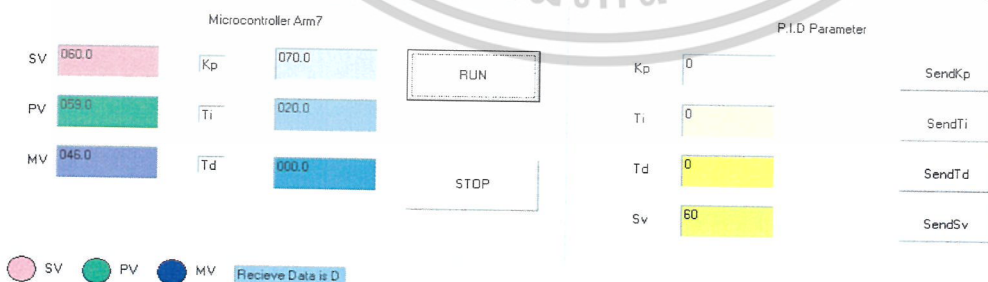
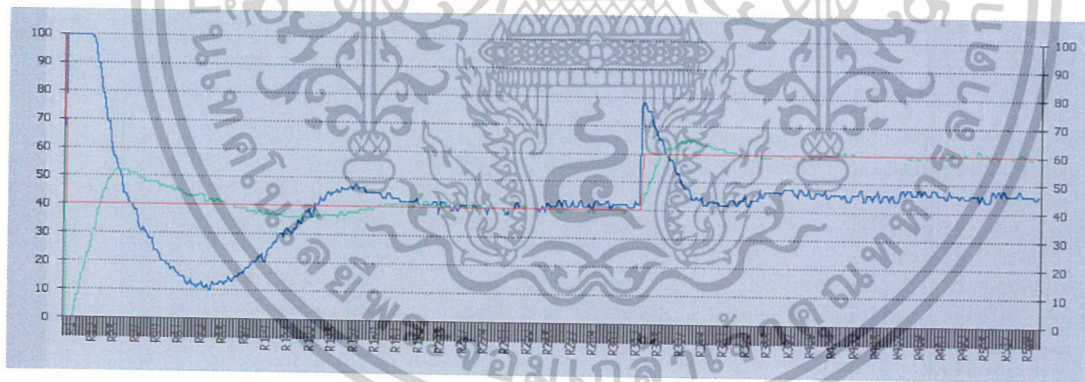
รูปที่ 4.15 สัญญาณควบคุมที่ได้จากตัวควบคุมแบบ Proportional + Integral

ด้วยวิธี Trial and Error เมื่อค่า  $K_p = 50$ ,  $T_i = 20$

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรรมใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.16 สัญญาณควบคุมที่ได้จากตัวควบคุมแบบ Proportional + Integral ด้วยวิธี Trial and Error เมื่อค่า  $K_p = 60$ ,  $T_i = 25$



รูปที่ 4.17 สัญญาณควบคุมที่ได้จากตัวควบคุมแบบ Proportional + Integral

ด้วยวิธี Trial and Error เมื่อค่า  $K_p = 70$ ,  $T_i = 20$

เอกสารนี้เป็นเอกสารที่ สงวนลิขสิทธิ์ไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรรมใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

#### 4.5 สรุปผลการทดลอง

จากการทดลอง การปรับค่าพารามิเตอร์ของตัวควบคุมกระบวนการควบคุมความดันในถังปิด เมื่อทำการปรับค่าพารามิเตอร์แบบพี แบบพีไอ แบบพีดี และแบบพีไอดี โดยใช้วิธีการปรับค่าแบบ Trial and Error เมื่อทำการพิจารณาค่าเป้าหมายและการเพิ่มโหลดโดยการปรับความดันเข้าไปในระบบให้เพิ่มมากขึ้นจะมีผลทำให้ค่าของตัวควบคุมสูงขึ้นเรื่อย ๆ เพื่อที่จะทำให้กระบวนการควบคุมเข้าสู่ค่าเป้าหมายได้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทที่ 5

# สรุปผลการทดลองและข้อเสนอแนะ

### 5.1 สรุปผลการทดลอง

จากการทดลองได้ใช้ตัวควบคุม PID ควบคุมกระบวนการความดันแบบ 2 ถึง โดยจะใช้เวลาในการ sampling 250 ms จะพบว่าตัวควบคุมสามารถควบคุมกระบวนการได้ ถึงแม้ว่าจะมีการเปลี่ยนแปลงตำแหน่งของเกวาล์ว ก็จะทำให้ตัวควบคุมสามารถทำการควบคุมความดันให้ค่าของกระบวนการเข้าสู่ค่าเป้าหมายได้ อีกทั้งยังสามารถแสดงผลและการส่งค่าพารามิเตอร์ต่าง ๆ เพื่อเป็น Monitoring เพื่อได้เฝ้าดูกระบวนการการทำงานของระบบควบคุมความดัน จากการทดลองสามารถทำให้เข้าใจวิธีการปรับค่าพารามิเตอร์ และดูพฤติกรรมของกระบวนการได้สามารถที่จะแก้ปัญหาได้

### 5.2 ข้อเสนอแนะแนวทางการพัฒนา

การควบคุมกระบวนการด้วย PID จึงจำเป็นที่จะต้องทราบค่าต่าง ๆ ของสมการคุณลักษณะของกระบวนการเพื่อที่จะทำให้เราได้คำนวณหาค่า PID เพื่อที่จะควบคุมกระบวนการจึงจะสามารถควบคุมได้อย่างมีประสิทธิภาพและง่ายขึ้น

แนวทางการพัฒนาของการทดลอง เพื่อทำการวิจัยที่จะทำการพัฒนาโดยใช้ตัวควบคุมเพื่อไปควบคุมกับกระบวนการอื่นและกระบวนการที่มีอันดับสูงได้

## บรรณานุกรม

: พื้นฐานวิศวกรรมระบบควบคุม / สุเชียร เกียรติสุนทร. กรุงเทพฯ : ภาควิชาวิศวกรรมระบบควบคุม คณะวิศวกรรมศาสตร์ สจล., 2535.

: ตัวควบคุมแบบพีไอดีหาค่าอัตโนมัติ / ฉัตรแก้ว จิรไพโรจน์, ศิริวรรณ วิทยาการ โกวิท. ปริญญาโท - ภาควิชาวิศวกรรมระบบควบคุม. สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง. 2542

: คู่มือการใช้งานไมโครคอนโทรลเลอร์ ARM 7. สำนักงานอุตสาหกรรมซอฟต์แวร์แห่งชาติ [องค์การมหาชน] SIPA

: วิศวกรรมการวัดคุม / รศ. ประสิทธิ์ จุลเสวีวงศ์ ภาควิชาวิศวกรรมการวัดคุม. คณะวิศวกรรมศาสตร์. สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง. พฤษภาคม 2549.



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาคผนวก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## Code คำสั่งของโปรแกรมไมโครคอนโทรลเลอร์ ARM7

```
#include <ADUc7024.H>           // ADUc7024 MPU Register
#include <stdio.h>               // For Used Function printf
#include <stdlib.h>

// LCD Routines for "ET-ARM7 BASE ADUc7024"
// Character 16x2 4-Bit Mode Interface

// EN = P4.4
// RW = P4.5
// RS = P4.6
// D4 = P4.0
// D5 = P4.1
// D6 = P4.2
// D7 = P4.3

// Define LCD PinIO Mask
#define LCD_EN  0x00100000      // P4.4(0000 0000 000x 0000 0000 0000 0000 0000)
#define LCD_RW  0x00200000      // P4.5(0000 0000 00x0 0000 0000 0000 0000 0000)
#define LCD_RS  0x00400000      // P4.6(0000 0000 0x00 0000 0000 0000 0000 0000)
#define LCD_D4  0x00010000      // P4.0(0000 0000 0000 000x 0000 0000 0000 0000)
#define LCD_D5  0x00020000      // P4.1(0000 0000 0000 00x0 0000 0000 0000 0000)
#define LCD_D6  0x00040000      // P4.2(0000 0000 0000 0x00 0000 0000 0000 0000)
#define LCD_D7  0x00080000      // P4.3(0000 0000 0000 x000 0000 0000 0000 0000)
#define LCD_DATA (LCD_D7|LCD_D6|LCD_D5|LCD_D4)
#define LCD_IOALL (LCD_D7|LCD_D6|LCD_D5|LCD_D4|LCD_RS|LCD_RW|LCD_EN)
#define lcd_rs_set()  GP4SET = LCD_RS          // RS = 1 (Select Instruction)
#define lcd_rs_clr()  GP4CLR = LCD_RS          // RS = 0 (Select Data)
#define lcd_rw_set()  GP4SET = LCD_RW          // RW = 1 (Read)
#define lcd_rw_clr()  GP4CLR = LCD_RW          // RW = 0 (Write)
#define lcd_en_set()  GP4SET = LCD_EN          // EN = 1 (Enable)
#define lcd_en_clr()  GP4CLR = LCD_EN          // EN = 0 (Disable)

#define lcd_dir_write() GP4DAT = 0x7F000000 // LCD Data Bus = Write
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

#define lcd_dir_read() GP4DAT = 0x70000000 // LCD Data Bus = Read
#define lcd_clear()    lcd_write_control(0x01) // Clear Display
#define lcd_cursor_home() lcd_write_control(0x02) // Set Cursor = 0
#define lcd_display_on() lcd_write_control(0x0E) // LCD Display Enable
#define lcd_display_off() lcd_write_control(0x08) // LCD Display Disable
#define lcd_cursor_blink() lcd_write_control(0x0F) // Set Cursor = Blink
#define lcd_cursor_on() lcd_write_control(0x0E) // Enable LCD Cursor
#define lcd_cursor_off() lcd_write_control(0x0C) // Disable LCD Cursor
#define lcd_cursor_left() lcd_write_control(0x10) // Shift Left Cursor
#define lcd_cursor_right() lcd_write_control(0x14) // Shift Right Cursor
#define lcd_display_sleft() lcd_write_control(0x18) // Shift Left Display
#define lcd_display_sright() lcd_write_control(0x1C) // Shift Right Display
/***** prototype section *****/
void IRQ_Handler (void) __irq ;
void ADC_DAC(void);
void Tx_Byte(unsigned char DataOut);
int putchar(int ch); // Write Data to UART
int getchar(void); // Read Data From UART
void WriteDS1307(unsigned char Write_Addr,unsigned char Set_Time);
unsigned char ReadDS1307(unsigned char Read_Addr);
void lcd_init(); // Initial LCD
void lcd_out_data4(unsigned char); // Strobe 4-Bit Data to LCD
void lcd_write_byte(unsigned char); // Write 1 Byte Data to LCD
void lcd_write_control(unsigned char); // Write Instruction
void lcd_write_ascii(unsigned char); // Write LCD Display(ASCII)
void goto_cursor(unsigned char); // Set Position Cursor LCD
void lcd_print(unsigned char*); // Print Display to LCD
char busy_lcd(void); // Read Busy LCD Status
void enable_lcd(void); // Enable Pulse
void delay(unsigned long int); // Delay Function
/***** Sub Function *****/
void select_fn(void);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

void pid_parameter(void);
void adjust_setpoint(void);
void show_pv_mv_sv_err(void);
void manual_function(void);
void function_pid(void);
void wait_sw(void);
/***** Adjust Parameter *****/
void kp_parameter(void);
void ti_parameter(void);
void td_parameter(void);
void adjust_kp(void);
void adjust_ti(void);
void adjust_td(void);
/***** Send Data *****/
void boot_data(void);
void send_data(void);
void Data2com(int Data);
/***** Receive Data *****/
void sendKp(void);
void sendTi(void);
void sendTd(void);
void sendSv(void);
/***** Update Data *****/
void Data_Kp(void);
void Data_Ti(void);
void Data_Td(void);
void Data_Sv(void);
/***** Initial Parameter *****/
char value_set_p[3],value_i[3],value_pv[3];
char value_kp[3],value_ti[3],value_td[3];
char value_err[3],value_mv[3],value_mi[3];

```

```
int sw,i,val,ma,mi,mv;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

/***** Variable memory *****/
int kpp,kp_H,kp_L;
int tii,ti_H,ti_L;
int tdd,td_H,td_L;
int set_pp,set_pL,set_pH;
/***** variable process *****/
float volt,pv,Pv,PV,a;
/***** Variable pid *****/
float sv,mv_1, mv_p, sv2;
float err,err_1;
float kp,ti,td,ts,P,I,D,I1;
/***** Variable send value *****/
int count,data1,data2,data3;
int count1,data4,data5,data6;
/***** initial Receive Data *****/
int temp[5];
int iCount;
int Kp,Kp1,Kp2,Kp3;
int Ti,Ti1,Ti2,Ti3;
int Td,Td1,Td2,Td3;
int Sv,Sv1,Sv2,Sv3;
int evreceive;
char value_Kp[5],value_Kp1[5],value_Kp2[5],value_Kp3[5];
char value_Ti[5],value_Ti1[5],value_Ti2[5],value_Ti3[5];
char value_Td[5],value_Td1[5],value_Td2[5],value_Td3[5];
char value_Sv[5],value_Sv1[5],value_Sv2[5],value_Sv3[5];

/***** Main Program Start Here *****/
int main(void)
{
    GP1CON = 0x00000011;
    GP3CON = 0x00000000;
    GP3DAT = 0xF0FF0000;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

// Initial I2C Interface
GP1CON &= 0xFFFFCCFF;
GP1CON |= 0x00002200;

// I2C-MASTER setup
I2C1CFG = 0x82;

// I2C-CLOCK = 100 KHz
I2C1DIV = 0xCFCF;

// Initial UART = 9600BPS
COMCON0 = 0x80; // Setting DLAB
COMDIV0 = 0x88; // Setting DIV0 and DIV1 to DL calculated
COMDIV1 = 0x00;
COMCON0 = 0x07; // Clearing DLAB
COMIEN0 = 0x01;
IRQEN |= 0x00004000;
// IRQSIG = 0x00FFF000;
IRQSTA = 0x20;
lcd_dir_write(); // Initial LCD Write Data
lcd_init(); // Initial LCD
ADC_DAC();
IRQEN |= 0x00000008; // Enable Timer1 Trigger IRQ Interrupt

// HCLK = 41.78 MHz
// Time 1 Cycle = 1 / 41.78 MHz
// = 23.9348 nS //20890 1ms
// Time 1 KHz = 1 / 1000 //2089008 100ms
// = 1 mS //5222521 250ms=Low+High
// 1KHz Signal = Low 500uS + High 500uS //10445042 500ms
// Count 1 KHz = 500uS / 23.9348nS //20890084 1000ms
// = 20890 Cycle
T1LD =5222521;
T1CON &= 0xFFFFFFF0; // Prescale = HCLK / 1
T1CON &= 0xFFFFF0CF; // Format = Binary Counter

```

```

T1CON |= 0x00000040;           // Timer1 Mode = Periodic
T1CON &= 0xFFFFFEFF;          // Timer1 = Count Down
T1CON &= 0xFFFF1FF;           // Timer1 Clock Source = HCLK
T1CON |= 0x00000080;          // Timer1 Enable

kp=0.1;

mv=0;      mv_p=0;
P=0;  I=0;  D=0;

err_1=0;

boot_data();

while(1)
{
    select_fn();
}

/***** Sub Fuction *****/
/**** Wait Switch *****/
void wait_sw(void)
{
    sw = GP3DAT & 0x0f;
}

/**** select function *****/
void select_fn(void)
{
    goto_cursor(0x00);
    lcd_print("SELECT Fun ");
    goto_cursor(0x40);
    lcd_print(" press up/down ");
    while(1)
    {
        wait_sw();
        if(sw==0x0b) {while(sw==0x0b) wait_sw(); i++; if(i>4) i=1;}
        if(sw==0x0d) {while(sw==0x0d) wait_sw(); i--; if(i<1) i=4;}
    }
}

switch(i)
{

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

case 1 : goto_cursor(0x00);

        lcd_print("Function 1  ");
        goto_cursor(0x40);

                lcd_print("PID PARAMETER  ");
        if(sw==0x0e)
        {
                while(sw==0x0e) wait_sw();
                pid_parameter();
        }

break;

case 2 : goto_cursor(0x00);

        lcd_print("Function 2  ");
        goto_cursor(0x40);

                lcd_print("SHOW PV MV SV E ");
        if(sw==0x0e)
        {
                while(sw==0x0e) wait_sw();
                show_pv_mv_sv_err();
        }

break;

case 3 : goto_cursor(0x00);
lcd_print("Function 3  ");
goto_cursor(0x40);

                lcd_print("SETPOINT  ");
        if(sw==0x0e)
        {
                while(sw==0x0e) wait_sw();
                adjust_setpoint();
        }

break;

case 4 : goto_cursor(0x00);

        lcd_print("Function 4  ");

                goto_cursor(0x40);

        lcd_print("MANUAL  ");

        if(sw==0x0e)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        {
            while(sw==0x0e) wait_sw();
            manual_function();
        }

        break;
    }

}

}

}

/**** PID PARAMETER *****/
void pid_parameter(void)
{
    int i=1;
    lcd_clear();
    while(1)
    {
        wait_sw();
        if(sw==0x0b) {while(sw==0x0b) wait_sw(); i++; if(i>3) i=1;}
        if(sw==0x0d) {while(sw==0x0d) wait_sw(); i--; if(i<1) i=3;}
switch(i)
    {
        case 1 : goto_cursor(0x00);
                lcd_print("PID PARAMETER ");
                goto_cursor(0x40);
                lcd_print("P PARAMETER ");
                if(sw==0x0e)
                {
                    while(sw==0x0e) wait_sw();
                    adjust_kp();
                }

        break;
        case 2 : goto_cursor(0x00);
                lcd_print("PID PARAMETER ");
                goto_cursor(0x40);
                lcd_print("I PARAMETER ");
                if(sw==0x0e)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        {
            while(sw==0x0e) wait_sw();
            adjust_ti();
        }

        break;

case 3 : goto_cursor(0x00);
        lcd_print("PID PARAMETER ");
        goto_cursor(0x40);
        lcd_print("D PARAMETER ");

        if(sw==0x0e)
        {
            while(sw==0x0e) wait_sw();
            adjust_td();
        }

        break;
    }
}

if(sw==0x07){while(sw==0x07)wait_sw();break;}

}

}

/***** Show value page mv and error *****/
void show_pv_mv_sv_err(void)
{
    while(1)
    {
        goto_cursor(0x00);
        lcd_print("PV = ");
        sprintf(value_pv,"%1.1f",pv);
        lcd_print(value_pv);
        lcd_print(" ");
        goto_cursor(0x40);
        lcd_print("SV = ");
        sprintf(value_set_p,"%1.1f",sv);
        lcd_print(value_set_p);
        lcd_print(" ");
        delay(5000000);

        goto_cursor(0x00);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        lcd_print("ERROR = ");
        sprintf(value_err,"%1.1f",err_s);
        lcd_print(value_err);
        lcd_print("      ");
        goto_cursor(0x40);
        lcd_print("MV = ");
        sprintf(value_mv,"%1.1f",mv_p);
        lcd_print(value_mv);
        lcd_print("      ");
delay(5000000);
        wait_sw();
        if(sw==0x07) break;
    }
}
/***** manual function *****/
void manual_function(void)
{
    lcd_clear();
    while(1)
    {
        IRQCLR |= 0x00000008;
        ADCCP = 0; // Select Channel to Conversion
        delay(1000); // Wait Select Channel Ready
        while (!ADCSTA){}; // Wait ADC Conversion Complete (Bit0="1")
        val = (ADCDAT >> 16)& 0x00000FFF; // Shift ADC Result to Integer
        pv= (100* val)/2047;

for(a=0;a<10;a++)
        {
            PV = PV + pv; }
        pv = PV/10;    PV = 0;
        if(pv>100)    pv=100;
        wait_sw();
        if(sw==0x0b) mi++; //up value
        if(sw==0x0d) mi--; //down value
        delay(100000);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        if(mi<0) mi=100;
        if(mi>100) mi=0;
        goto_cursor(0x00);
        lcd_print(" Manual ");
        goto_cursor(0x40);
        sprintf(value_mi,"%d",mi);
        lcd_print("manual = ");
        lcd_print(value_mi);
        if(mi<100) lcd_print(" ");
        if(mi<10) lcd_print(" ");
        lcd_print(" % ");
ma=mi*(4095/100);
        DAC0DAT = (ma<< 16);
        delay(100000);
        mv_p=mi;
        send_data();
if(sw==0x07){while(sw==0x07)wait_sw();break;}
        }
        IRQEN |= 0x00000008;
    }
/***** Function PID *****/
void function_pid(void)
{
    ADCCP = 0; // Select Channel to Conversion
    delay(1000); // Wait Select Channel Ready
    while (!ADCSTA){}; // Wait ADC Conversion Complete (Bit0="1")
    val = (ADCDAT >> 16)& 0x00000FFF; // Shift ADC Result to Integer
    if (val > 2047) val = 2047;
    pv= (100* val)/2047;
    for(a=0;a<10;a++)
    {
        PV = PV + pv; }
    pv = PV/10;    PV = 0;
    if(pv>100)    pv=100;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

err=sv - pv;
P = kp * err ;
if(ti==0) I=0;
I = I1 + (kp *err)/(4*ti) ;
if(td==0) D=0;
D = (kp*td*(err - err_1))*4;
mv = P + I + D ;
if (mv > 4095) {mv = 4095; I=I1;}
mv_p = (mv * 100)/ 4095;
if (mv < 0) {mv = 0; I=I1;}
mv_p = (mv * 100)/ 4095;
err_1=err;
I1=I;
DAC0DAT = (mv << 16);
delay (100000);
}
/***** Adjust Setpoint Value *****/
/***** setpoint value *****/
void adjust_setpoint(void)
{
  lcd_clear();
  while(1)
  {
    IRQCLR |= 0x00000008;
wait_sw();
if(sw==0x0b) sv += 1;
if(sw==0x0d) sv -= 1;
if(sw==0x0a) sv += 10;
if(sw==0x0c) sv -= 10;
delay(100000);
if(sv<0) sv=100;
if(sv>100) sv=0;
set_pp=sv*10;
set_pL = set_pp%100;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

set_pH =      set_pp/100;
WriteDS1307(0x10,set_pL);
delay(300000);
WriteDS1307(0x11,set_pH);
delay(300000);
goto_cursor(0x00);
lcd_print("SETPOINT value ");
goto_cursor(0x40);
sprintf(value_set_p,"%1.1f",sv);
lcd_print("SV = ");
lcd_print(value_set_p);  lcd_print("      ");
if(sw==0x07){while(sw==0x07)wait_sw();break;}
}
IRQEN |= 0x00000008;
}
/***** set value parameter PID *****/
/***** Kp Parameter *****/
void adjust_kp(void)
{
  lcd_clear();
  while(1)
  {
    IRQCLR |= 0x00000008;
    wait_sw();
    if(sw==0x0b) kp=kp+0.1;      //up value
    if(sw==0x0d) kp=kp-0.1;      //down value
    if(sw==0x0a) kp+=10;
    if(sw==0x0c) kp-=10;
    delay(100000);

    if(kp<0.1) kp=999.9;
    if(kp>999.9) kp=0.1;

    kpp = kp*10;
    kp_L =      kpp%100;
    kp_H =      kpp/100;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        WriteDS1307(0x09,kp_L);
        delay(300000);
        WriteDS1307(0x0a,kp_H);
        delay(300000);
        goto_cursor(0x00);
        lcd_print("P parameter  ");
        goto_cursor(0x40);
        sprintf(value_kp,"%1.1f",kp);
        lcd_print("Kp = ");
        lcd_print(value_kp);    if(kp<1000) lcd_print("    ");
if(sw==0x07){while(sw==0x07)wait_sw();break;}
    }
    IRQEN |= 0x00000008;
}
/***** Ti Parameter *****/
void adjust_ti(void)
{
    lcd_clear();
    while(1)
    {
        IRQCLR |= 0x00000008;
        wait_sw();
        if(sw==0x0b) ti+=0.1;    //up value
        if(sw==0x0d) ti-=0.1;    //down value
        if(sw==0x0a) ti+=10;
        if(sw==0x0c) ti-=10;
        delay(100000);
        if(ti<0) ti=999.9;
        if(ti>999.9) ti=0;
        tii=ti*10;
        ti_L  =    tii%100;
        ti_H  =    tii/100;
        WriteDS1307(0x0b,ti_L);
        delay(300000);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

WriteDS1307(0x0c,ti_H);
delay(300000);
goto_cursor(0x00);
lcd_print("I parameter ");
goto_cursor(0x40);
sprintf(value_ti,"%1.1f",ti);
lcd_print("Ti = ");
lcd_print(value_ti); lcd_print(" ");
if(sw==0x07){while(sw==0x07)wait_sw();break;}
}
IRQEN |= 0x00000008;
}
/***** Td Parameter *****/
void adjust_td(void)
{ lcd_clear();
while(1)
{ IRQCLR |= 0x00000008;
wait_sw();
if(sw==0x0b) td+=0.1; //up value
if(sw==0x0d) td-=0.1; //down value
if(sw==0x0a) td+=10;
if(sw==0x0c) td-=10;
delay(100000);
if(td<0) td=999.9;
if(td>999.9) td=0;
tdd=td*10;
td_L = tdd%100;
td_H = tdd/100;
WriteDS1307(0x0d,td_L);
delay(300000);
WriteDS1307(0x0e,td_H);
delay(300000);
}
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        td_L   = ReadDS1307(0x0d);
        td_H   = ReadDS1307(0x0e);
        td     = (td_H*10)+(td_L*0.1);

        goto_cursor(0x00);
        lcd_print("D parameter  ");
        goto_cursor(0x40);
        sprintf(value_td,"%1.1f",td);
        lcd_print("Td = ");
        lcd_print(value_td);      lcd_print("      ");
        if(sw==0x07){while(sw==0x07)wait_sw();break;}
    }
    IRQEN |= 0x00000008;
}

/***** Receive Data Fuction *****/
/**** Send KP *****/
void sendKp(void)
{
    Kp1 = temp[0]; Kp2 = temp[1]; Kp3 = temp[2];
    sprintf(value_Kp1,"%c",Kp1);
    sprintf(value_Kp2,"%c",Kp2);
    sprintf(value_Kp3,"%c",Kp3);
}

/**** Send Ti *****/
void sendTi(void)
{
    Ti1 = temp[0]; Ti2 = temp[1]; Ti3 = temp[2];
    sprintf(value_Ti1,"%c",Ti1);
    sprintf(value_Ti2,"%c",Ti2);
    sprintf(value_Ti3,"%c",Ti3);
}

/**** Send Td *****/
void sendTd(void)
{
    Td1 = temp[0]; Td2 = temp[1]; Td3 = temp[2];
    sprintf(value_Td1,"%c",Td1);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        sprintf(value_Td2,"%c",Td2);
        sprintf(value_Td3,"%c",Td3);
    }
    /***** Send Sv *****/
    void sendSv(void)
    {
        Sv1 = temp[0]; Sv2 = temp[1]; Sv3 = temp[2];
        sprintf(value_Sv1,"%c",Sv1);
        sprintf(value_Sv2,"%c",Sv2);
        sprintf(value_Sv3,"%c",Sv3);
    }

    /***** Update Data *****/
    /***** Data Kp *****/
    void Data_Kp (void)
    {
        IRQCLR |= 0x00000008;
        Kp = (atoi(value_Kp3)*100)+(atoi(value_Kp2)*10)+(atoi(value_Kp1));
        kp = (float)Kp;
        kpp = kp*10;
        kp_L = kpp%100;
        kp_H = kpp/100;
        WriteDS1307(0x09,kp_L);
        delay(300000);
        WriteDS1307(0x0a,kp_H);
        delay(300000);
        IRQEN |= 0x00000008;
    }
    /***** Data Ti *****/
    void Data_Ti (void)
    {
        IRQCLR |= 0x00000008;
        Ti = (atoi(value_Ti3)*100)+(atoi(value_Ti2)*10)+(atoi(value_Ti1));
        ti = (float)Ti;
        tii=ti*10;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

ti_L = tii%100;
ti_H = tii/100;
WriteDS1307(0x0b,ti_L);
delay(300000);
WriteDS1307(0x0c,ti_H);
delay(300000);
IRQEN |= 0x00000008;
}
/**** Data Td *****/
void Data_Td (void)
{
IRQCLR |= 0x00000008;
Td = (atoi(value_Td3)*100)+(atoi(value_Td2)*10)+(atoi(value_Td1));
td =(float)Td;
tdd=td*10;
td_L = tdd%100;
td_H = tdd/100;
WriteDS1307(0x0d,td_L);
delay(300000);
WriteDS1307(0x0e,td_H);
delay(300000);
IRQEN |= 0x00000008;
}
/**** Data Sv *****/
void Data_Sv (void)
{
IRQCLR |= 0x00000008;
Sv = (atoi(value_Sv3)*100)+(atoi(value_Sv2)*10)+(atoi(value_Sv1));
sv = (float)Sv;
set_pp=sv*10;
set_pL = set_pp%100;
set_pH = set_pp/100;
WriteDS1307(0x10,set_pL);
delay(300000);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

WriteDS1307(0x11,set_pH);
delay(300000);
IRQEN |= 0x00000008;
}
/***** Memory Value & Send Data *****/
/**** Memory Value *****/
void boot_data(void)
{
    set_pL      = ReadDS1307(0x10);
    set_pH      = ReadDS1307(0x11);
    sv          = (set_pH*10)+(set_pL*0.1);
    kp_L        = ReadDS1307(0x09);
    kp_H        = ReadDS1307(0x0a);
    kp          = (kp_H*10)+(kp_L*0.1);
    ti_L        = ReadDS1307(0x0b);
    ti_H        = ReadDS1307(0x0c);
    ti          = (ti_H*10)+(ti_L*0.1);
    td_L        = ReadDS1307(0x0d);
    td_H        = ReadDS1307(0x0e);
    td          = (td_H*10)+(td_L*0.1);
}

/**** Calculate Data Send *****/
void Data2com(int Data)
{
    int data_100,data_10,data_1,data_01;
    data_100=Data/1000;
    data_10=(Data%1000)/100;
    data_1=(Data%100)/10;
    data_01=(Data%10);
    Tx_Byte(data_100+0x30);
    Tx_Byte(data_10+0x30);
    Tx_Byte(data_1+0x30);
    Tx_Byte(0x2E);
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        Tx_Byte(data_01+0x30);    }
/***** Send Data *****/
void send_data(void)
{
    data1=sv*10; data2=pv*10; data3=mv_p*10;
    data4=((kp*10)+0.1); data5=ti*10; data6=td*10;
    count=count+1;
if(count==1) Tx_Byte(0x41);
    if(count==2) Data2com(data1);
    if(count==3) Tx_Byte(0x42);
if(count==4) Data2com(data2);
    if(count==5) Tx_Byte(0x43);
    if(count==6) Data2com(data3);
if(count==7) Tx_Byte(0x44);
    if(count==8) Data2com(data4);
if(count==9) Tx_Byte(0x45);
    if(count==10) Data2com(data5);
if(count==11) Tx_Byte(0x46);
    if(count==12) {Data2com(data6); count=0;}
}
/***** Pototype Fucntion *****/
/***** Interrupt Service Rountine *****/
void IRQ_Handler (void) __irq // IRQ Service Routine
{
    char msg;
if ((IRQSTA & 0x00000008) != 0) // if Timer1 IRQ Flag Status
{
        function_pid();
        send_data();
        T1CLR1 = 0; // Clear Timer1 Trigger IRQ Flag
    }
if (((msg = COMIID0)&0x01)==0)

```

```

    {
        switch (msg & 0x0E)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

{
    case 0x04: while(!(0x40==(COMSTA0 & 0x40)));
    evreceive = COMRX;
    temp[iCount] = evreceive;
    if(evreceive=='a')
        {
            sendKp();
            iCount = 0;
            Data_Kp();
        }
    else if(evreceive=='b')
        {
            sendTi();
            iCount = 0;
            Data_Ti();
        }
    else if(evreceive=='c')
        {
            sendTd();
            iCount = 0;
            Data_Td();
        }
    else if(evreceive=='d')
        {
            sendSv();
            iCount = 0;
            Data_Sv();
        }
}

else iCount++;
break ;
case 0x02:break;
default : break;
}

}

return ;
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

/***** Analog To Digital **Digital To Analog *****/
void ADC_DAC(void)
{
    // Power-ON ADC
    ADCCON = 0x00000000;    // Reset ADC Config
    ADCCON |= 0x00000020;   // Power-ON ADC Function
    delay(1000);           // Wait ADC Power-on Ready
    ADCCON |= 0x00001400;   // ADC Clock = fADC/32
    ADCCON |= 0x00000300;   // Acquisition Time = 16 Cycle Clock
    ADCCON &= 0xFFFFFE7;   // ADC = Single-End Mode
    ADCCON |= 0x00000004;   // Continue Software Convert
    REFCON = 0x00000001;    // Used Internal 2.5V Reference
    ADCCON |= 0x00000080;   // ADC Start Conversion
    // Initial DAC0
    DAC0CON &= 0xDF;        // DAC0 Used Sysytem Clock
    DAC0CON |= 0x10;        // Enable DAC0
    DAC0CON |= 0x02;        // DAC0 Output Range = AVDD..AGND
    REFCON = 0x01;
    // Initial DAC1
    DAC1CON &= 0xDF;        // DAC1 Used Sysytem Clock
    DAC1CON |= 0x10;        // Enable DAC1
    DAC1CON |= 0x02;        // DAC1 Output Range = +Vref..AGND
    REFCON = 0x01;         // Used Internal 2.5V Reference
}

/***** Write Character To UART *****/
void Tx_Byte(unsigned char Datasent)    // Write character to Serial Port
{
    while(!(0x40==(COMSTA0 & 0x40)))    // Wait TX Complete
    {
    }
    COMTX = Datasent;                    // Write CR
    while(!(0x40==(COMSTA0 & 0x40)))    // Wait TX Complete
    {
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

}

/***** Write Character To UART *****/
int putchar(int ch)                                // Write character to Serial Port
{
    if(ch == '\n')
    {
while(!(0x20==(COMSTA0 & 0x20)))                // Wait TX Complete
    {
    }
        COMTX = 0x0D;                            // Write CR
    }
        while(!(0x20==(COMSTA0 & 0x20)))        // Wait TX Complete
        {
        }
        return (COMTX = ch);
    }
/***** Read Character From UART *****/
int getchar (void)                                // Read character from Serial Port
{
    while(!(0x01==(COMSTA0 & 0x01)))            // Wait Receive Data Ready
    {
    }
        return (COMRX);
    }
/***** Read Byte From RTC *****/
unsigned char ReadDS1307 (unsigned char Read_Addr)
{
    unsigned char Get_Byte;
    I2C1CNT = 0x00;

//1st Start Condition
    I2C1ADR = 0xD0;                                //data write    slave recieve
    I2C1MTX = Read_Addr;
    while((I2C1FSTA & 0x30)!=0x00);

//2nd Start Condition
    I2C1CNT = 0;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

                I2C1ADR = 0xD1;                //data read    slave transmitter
//enable Generate Stop Condition
                I2C1CCNT = 0x80;

//Wait receive Data From Device
                while((I2C1MSTA & 0x08)!=0x08);
                Get_Byte = I2C1MRX;
                return Get_Byte;
    }

    /**** Write Byte to RTC : DS1307 *****/
    void WriteDS1307(unsigned char Write_Addr,unsigned char Set_Time)
    {
        I2C1MTX = Write_Addr;
        I2C1MTX = Set_Time;
        I2C1ADR = 0xD0;
//Wait I2C Send Complete
        while ((I2C1MSTA & 0x04)!=0x04);
    }

    /**** Initial 4-Bit LCD Interface *****/
    void lcd_init()
    {
        unsigned int i;                // Delay Count
        GP4CLR = (LCD_IOALL);          // Reset (RS,RW,EN,4-Bit Data) Pin
        for (i=0;i<100000;i++);        // Power-On Delay (15 mS)
        GP4CLR = (LCD_IOALL);          // Reset (RS,RW,EN,4-Bit Data) Pin
        GP4SET = (LCD_D5|LCD_D4);      // 0000:0000:(0,RS,RW,EN:0011):0000:0000:0000:0000
        enable_lcd();                  // Enable Pulse
        for (i=0;i<10000;i++);        // Delay 4.1mS
        GP4CLR = (LCD_IOALL);          // Reset (RS,RW,EN,4-Bit Data) Pin
        GP4SET = (LCD_D5|LCD_D4);      // 0000:0000:(0,RS,RW,EN:0011):0000:0000:0000:0000
        enable_lcd();                  // Enable Pulse
        for (i=0;i<1000;i++);         // delay 100uS
        GP4CLR = (LCD_IOALL);          // Reset (RS,RW,EN,4-Bit Data) Pin

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

GP4SET = (LCD_D5|LCD_D4);          // 0000:0000:(0,RS,RW,EN:0011):0000:0000:0000:0000
enable_lcd();                      // Enable Pulse
while(busy_lcd());                // Wait LCD Execute Complete
GP4CLR = (LCD_IOALL);             // Reset (RS,RW,EN,4-Bit Data) Pin
GP4SET = (LCD_D5);                // 0000:0000:(0,RS,RW,EN:0011):0000:0000:0000:0000
enable_lcd();                      // Enable Pulse
while(busy_lcd());                // Wait LCD Execute Complete
lcd_write_control(0x28);          // Function Set (DL=0 4-Bit,N=1 2 Line,F=0 5X7)
lcd_write_control(0x0C);          // Display on/off Control (Entry Display,Cursor off,Cursor not
Blink)
lcd_write_control(0x06);          // Entry Mode Set (I/D=1 Increment,S=0 Cursor Shift)
lcd_write_control(0x01);          // Clear Display (Clear Display,Set DD RAM Address=0)
for (i=0;i<10000;i++);           // Wait Command Ready
}
/**** Strobe 4-Bit Data to LCD *****/
void lcd_out_data4(unsigned char val)
{
    GP4CLR = (LCD_DATA);          // Reset 4-Bit Pin Data
    GP4SET = (val<<16);           // 0000:0000:0,RS,RW,EN:DDDD:0000:0000:0000:0000
}
/**** Write Data 1 Byte to LCD *****/
void lcd_write_byte(unsigned char val)
{
    lcd_out_data4((val>>4)&0x0F); // Strobe 4-Bit High-Nibble to LCD
    enable_lcd();                 // Enable Pulse
    lcd_out_data4(val&0x0F);      // Strobe 4-Bit Low-Nibble to LCD
    enable_lcd();                 // Enable Pulse
    while(busy_lcd());            // Wait LCD Execute Complete
}
/**** Write Instruction to LCD *****/
void lcd_write_control(unsigned char val)
{
    lcd_rs_clr();                 // RS = 0 = Instruction Select
    lcd_write_byte(val);          // Strobe Command Byte
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

/***** Write Data(ASCII) to LCD *****/
void lcd_write_ascii(unsigned char c)
{
    lcd_rs_set();           // RS = 1 = Data Select
    lcd_write_byte(c);      // Strobe 1 Byte to LCD
}

/***** Set LCD Position Cursor *****/
void goto_cursor(unsigned char i)
{
    i |= 0x80;              // Set DD-RAM Address Command
    lcd_write_control(i);
}

/**** Print Display Data(ASCII) to LCD *****/
void lcd_print(unsigned char* str)
{
    int i;
    for (i=0;i<16 && str[i]!=0;i++) // 16 Character Print
    {
        lcd_write_ascii(str[i]); // Print Byte to LCD
    }
}

/***** Wait LCD Ready *****/
char busy_lcd(void)
{
    unsigned char busy_status; // Busy Status Read
    unsigned int i;           // Delay Count
    lcd_dir_read();           // LCD Data Bus = Read
    lcd_rs_clr();             // Instruction Select
    lcd_rw_set();             // Read Direction
    lcd_en_set();             // Start Read Busy
    for (i=0;i<100;i++);      // Delay Before Read
    busy_status = (GP4DAT & 0x08); // Read LCD Data
    if(busy_status == 0x08) // Read & Check Busy Flag
    {
        lcd_en_clr();        // Disable Read
        lcd_rw_clr();        // Default = Write Direction
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        lcd_dir_write();        // LCD Data Bus = Write
        return 1;                // LCD Busy Status
    }
else
    {
        lcd_en_clr();          // Disable Read
        lcd_rw_clr();          // Default = Write Direction
        lcd_dir_write();        // LCD Data Bus = Write
    }
return 0;                        // LCD Ready Statu
}
}
/***** Enable Pulse to LCD *****/
void enable_lcd(void)            // Enable Pulse
{
    unsigned int i;            // Delay Count
    lcd_en_set();              // Enable ON
    for (i=0;i<10;i++);
    lcd_en_clr();              // Enable OFF
}
/***** Delay Time Function *****/
/* 1-4294967296 */
void delay(unsigned long int count1)
{
    while(count1 > 0) {count1--;} // Loop Decrease Counter
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้





หมายเลข 14 คือ Switch RESET ใช้สำหรับ Reset การทำงานของ CPU

หมายเลข 15 คือ Switch LOAD ใช้ร่วมกับ RESET สำหรับสั่ง Download HEX ให้ MCU

หมายเลข 16 คือ ขั้วต่อ ARM-JTAG สำหรับ Debug โปรแกรม

หมายเลข 17 คือ ขั้วต่อ RS232 สำหรับใช้งานทั่วไป และ Download HEX ให้ MCU

หมายเลข 18 คือ LED Power ใช้ แสดงสถานะของแหล่งจ่ายไฟ +5V

### ขั้วต่อสัญญาณต่างๆ

สำหรับขั้วต่อสัญญาณของพอร์ต I/O จาก MCU นั้น จะถูกออกแบบและจัดเตรียมไว้ผ่านทางขั้วต่อแบบ IDE Header ขนาด 10Pin (2x5) จำนวน 3 ชุดคือ PORT-P1, PORT-P3 และ PORT-P4 ตามลำดับ ส่วน PORT-P0 และ PORT-P2 นั้น จะต่อเป็น Header ขนาด 1x7 โดยที่ขั้วต่อสัญญาณแต่ละชุด จะประกอบไปด้วยสัญญาณของ I/O ที่เชื่อมต่อมาจากขาสัญญาณของ MCU โดยตรงทั้งหมด โดยจุดเชื่อมต่อกับสัญญาณภายนอกบอร์ด มีดังนี้

ขั้วต่อแหล่งจ่ายขนาด +5VDC

ขั้วต่อ PORT-P0 มี 6 บิต คือ P0[0,3,4,5,6,7]

ขั้วต่อ PORT-P1 มี 6 บิต คือ P1[2,3,4,5,6,7] ส่วน P1.0 และ P1.1 จะถูกเชื่อมต่อผ่านวงจร Line Driver (MAX232) สำหรับแปลงระดับสัญญาณจากระดับลอจิก TTL ของ MCU ให้เป็นสัญญาณแรงดันตามมาตรฐานของ RS232 โดยสัญญาณที่ได้รับการแปลงเป็นแบบ RS232 จะถูกเชื่อมต่อ ไปรอไว้ที่ขั้วต่อแบบ CPA ขนาด 4 PIN (RS232)

ขั้วต่อ PORT-P2 มีขนาด 1 บิต คือ P2.0

ขั้วต่อ PORT-P3 มีขนาด 8 บิต คือ P2 [0,1,2,3,4,5,6,7]

ขั้วต่อ PORT-P4 มีขนาด 8 บิตคือ P4 [0,1,2,3,4,5,6,7] โดยมีการต่อสัญญาณไปยังขั้ว LCD ด้วยจำนวน 7 บิต คือ P4 [0,1,2,3,4,5,6] ตามลำดับ

ขั้วต่อ ADC มีขนาด 10 ช่อง คือ ADC0-ADC9 โดยรับสัญญาณ Analog ได้ระหว่าง 0-2.5V

จุดรับแรงดันอ้างอิงของ DAC (DAC Reference) จากภายนอกขนาด 0-3V

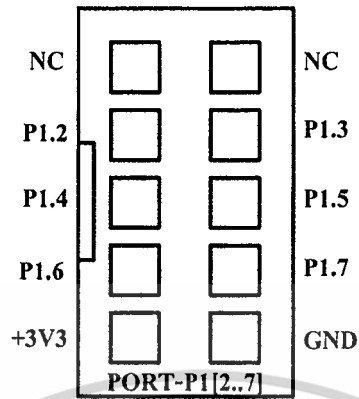
ขั้วต่อ DAC มีขนาด 2 ช่อง คือ DAC0, DAC1 โดยสามารถสร้างสัญญาณ Analog ได้ 0-2.5V

จุดรับแรงดันอ้างอิงของ ADC (+Vref ADC) จากภายนอกขนาด 0-2.5V

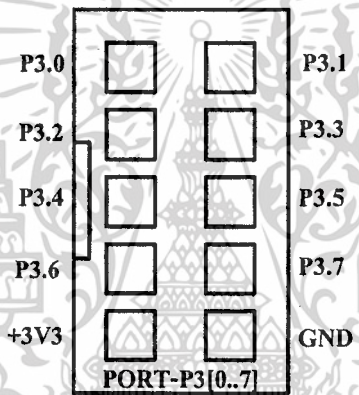
ขั้วต่อ ARM-JTAG

ขั้วต่อสัญญาณ RS2

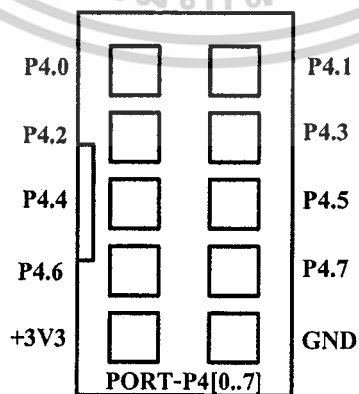
พอร์ต P1 มีขนาด 6 บิต



พอร์ต P3 มีขนาด 8 บิต

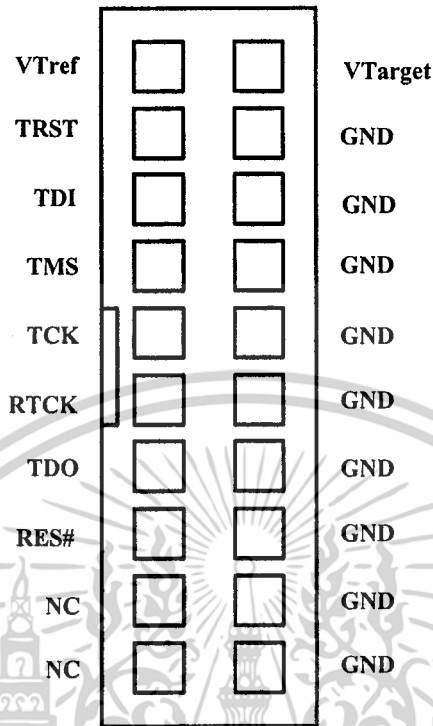


พอร์ต P4 มีขนาด 8 บิต

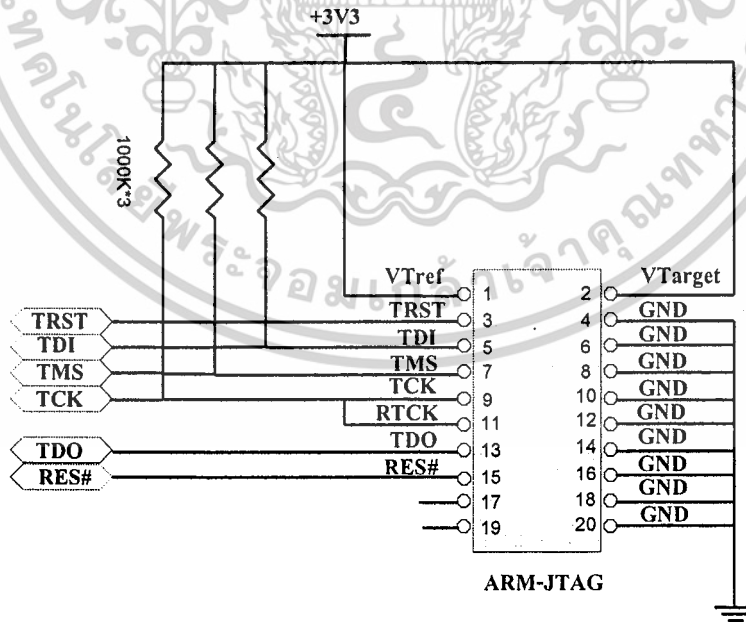


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# พอร์ต ARM-JTAG



ARMJTAG PIN Connector



รูปแสดง วงจรที่ใช้ในการเชื่อมต่อกับ ARM-JTAG

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้