

การออกแบบโปรแกรมจำลองและควบคุมกระบวนการระดับแบบสองถัง

THE DESIGN OF SIMULATION AND LEVEL CONTROL PROGRAM  
FOR SERIES-TANK PROCESS



ปริญญาโทนี้เป็นส่วนหนึ่งของการศึกษาระดับปริญญาโท สาขาวิชาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมการวัดคุม

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2556

การออกแบบโปรแกรมจำลองและควบคุมกระบวนการระดับแบบสองถัง  
THE DESIGN OF SIMULATION AND LEVEL CONTROL PROGRAM  
FOR SERIES-TANK PROCESS



ภัทรพร จงบุญทรัพย์  
ภาณุพงศ์ โตเต็ม  
เกตุณภัส พลอำนาจเดช

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต  
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานภายในของสถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงแก้ไข และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้  
คณะวิศวกรรมศาสตร์  
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง  
ปีการศึกษา 2555

THE DESIGN OF SIMULATION AND LEVEL CONTROL PROGRAM  
FOR SERIES-TANK PROCESS



PHATTARAPORN JONGBOONSUB  
PANUPONG TOTEM  
KETNAPHAT PONAMNUAYDET

A THESIS SUBMITTED IN PARTIAL FULFILLMENT  
OF THE REQUESTMENT FOR THE DEGREE OF  
BACHELOR OF ENGINEERING IN INSTRUMENTATION ENGINEERING  
FACULTY OF ENGINEERING

KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

ACADEMIC YEAR 2012

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้ในการเรียนการสอนเท่านั้น ไม่สามารถนำออกจำหน่ายหรือทำซ้ำโดยไม่ได้รับอนุญาต  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญาานิพนธ์ปีการศึกษา 2555  
คณะวิศวกรรมศาสตร์  
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง  
ใบรับรองปริญญาานิพนธ์

หัวข้อปริญญาานิพนธ์ การออกแบบโปรแกรมจำลองและควบคุมกระบวนการระดับแบบสองถัง  
THE DESIGN OF SIMULATION AND LEVEL CONTROL PROGRAM  
FOR SERIES-TANK PROCESS

นักศึกษาผู้จัดทำ นางสาวภัทรพร จงบุญทรัพย์ รหัสนักศึกษา 52010896  
นายภาณุพงศ์ โตเต็ม รหัสนักศึกษา 52010905  
นางสาวเกตุณภัส พลอำนวยการ เลข รหัสนักศึกษา 52011091

ปริญญา วิศวกรรมศาสตรบัณฑิต  
สาขา วิศวกรรมการวัดคุม  
ปีการศึกษา 2555

อาจารย์ผู้ควบคุมปริญญาานิพนธ์	ลายมือชื่อ
รศ.อาจันต์ น่วมสำราญ	
รศ.ดร.วิทยา ทิพย์สุวรรณพร	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อปริญญานิพนธ์	การออกแบบโปรแกรมจำลองและควบคุมกระบวนการระดับแบบสองถัง THE DESIGN OF SIMULATION AND LEVEL CONTROL PROGRAM FOR SERIES-TANK PROCESS	
นักศึกษาผู้จัดทำ	นางสาวภัทราพร จงบุญทรัพย์	รหัสนักศึกษา 52010896
	นายภาณุพงศ์ โตเต็ม	รหัสนักศึกษา 52010905
	นางสาวเกตุณภัส พลอำนาจเดช	รหัสนักศึกษา 52011091
อาจารย์ที่ปรึกษา	รศ.อาจินต์ น่วมสำราญ รศ.ดร.วิทยา ทิพย์สุวรรณพร	
ปีการศึกษา	2555	

### บทคัดย่อ

ปริญญานิพนธ์ฉบับนี้เป็นการนำเสนอ การควบคุมระดับน้ำแบบสองถังที่มีการต่อแบบซีรี่ส์ โดยใช้การควบคุมแบบพีไอดี และใช้วิธีการหาค่าพารามิเตอร์ของพีไอดีด้วยวิธีทางเดินราก ในการวิเคราะห์ทางเดินรานั้นจะใช้โปรแกรม MATLAB ในการวิเคราะห์เพื่อหาค่าพารามิเตอร์พีไอดีที่เหมาะสมเพื่อให้ระบบที่ถูกควบคุมนั้นเป็นไปตามที่ต้องการ โดยในการควบคุมระบบและการจำลองกระบวนการจะใช้โปรแกรม LabVIEW ในการเขียนโปรแกรมในลักษณะของภาษากาฟิก เพื่อให้โปรแกรมเป็นไปตามที่ต้องการ นอกจากนี้ไมโครคอนโทรลเลอร์ยังถูกนำมาใช้เป็นตัวเชื่อมต่อระหว่างตัวควบคุมที่ถูกโปรแกรมขึ้นมากับกระบวนการที่ถูกควบคุม เพื่อให้การควบคุมระหว่างคอมพิวเตอร์กับกระบวนการ สามารถเชื่อมต่อและควบคุมได้ จากนั้นจะนำผลที่ได้จากกระบวนการจริงและการจำลองมาเปรียบเทียบผลกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

<b>Thesis Title</b>	THE DESIGN OF SIMULATION AND LEVEL CONTROL PROGRAM FOR SERIES-TANK PROCESS	
<b>Authors</b>	Mrs. Phattaraporn	Jongboonsub
	Mr. Panupong	Totem
	Mrs. Ketnaphat	Ponamnuaydet
<b>Thesis Advisor</b>	Assoc.Prof. Arjin	Numsomran
	Assoc.Prof.Dr.Vittaya	Tipsuwanporn
<b>Year</b>	2012	

### ABSTRACT

This thesis presents a controlled series tanks by using PID controller and finding PID parameters base on root locus theory. For analysis is using MATLAB program to analyze a proper parameter that made the system is stable. LabVIEW is a graphic design program that used in control and simulation. In addition a microcontroller is connected between a control program and plant to be controlled and connected. Finally the result of real process is compared with the simulation process.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## กิตติกรรมประกาศ

ปริญญานิพนธ์ฉบับนี้สำเร็จลุล่วงได้ด้วยดี ด้วยคำแนะนำและคำปรึกษา รวมถึงการอนุเคราะห์อุปกรณ์และเครื่องมือต่างๆที่ใช้สำหรับทำโครงการ จากท่าน รศ.อาจินต์ น่วมสำราญ และท่าน รศ.ดร. วิทยา ทิพย์สุวรรณพร ซึ่งเป็นอาจารย์ผู้ควบคุมปริญญานิพนธ์ ที่เป็นผู้ปรับวิธีคิดและทัศนคติที่เกี่ยวกับการทำโครงการรวมถึงกระบวนการคิดอย่างมีระบบและเป็นแบบแผนต่อการดำเนินชีวิตให้กับคณะผู้จัดทำโครงการด้วยดีเสมอมา คณะผู้จัดทำโครงการรู้สึกซาบซึ้งใจในความอนุเคราะห์จากท่านทั้งสอง และขอขอบพระคุณเป็นอย่างสูง

ขอกราบขอบพระคุณคณาจารย์สาขาวิชาวิศวกรรมการวัดคุม คณะวิศวกรรมศาสตร์สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบังทุกท่าน ที่ได้ประสิทธิ์ประสาทวิชาให้กับคณะผู้จัดทำโครงการ

ขอขอบพระคุณ คุณสรารุช พลเสน เป็นอย่างยิ่ง สำหรับคำแนะนำ ที่ดีเสมอมา

สุดท้ายนี้คณะผู้จัดทำโครงการขอกราบขอบพระคุณบิดามารดาและครอบครัว ที่เป็นกำลังใจและให้การสนับสนุนมาด้วยดีตลอด ทำให้คณะผู้จัดทำโครงการสามารถทำปริญญานิพนธ์ฉบับนี้สำเร็จลุล่วงไปด้วยดี

คุณค่าและประโยชน์อันพึงมาจากปริญญานิพนธ์ฉบับนี้ คณะผู้จัดทำโครงการขอบแต่ผู้มีพระคุณทุกท่าน

คณะผู้จัดทำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# สารบัญ

	หน้า
บทคัดย่อภาษาไทย.....	I
บทคัดย่อภาษาอังกฤษ.....	II
กิตติกรรมประกาศ.....	III
สารบัญ.....	IV
สารบัญตาราง.....	VII
สารบัญภาพ.....	VIII
<b>บทที่ 1 บทนำ.....</b>	<b>1</b>
1.1 ความสำคัญของปริญญาานิพนธ์.....	1
1.2 วัตถุประสงค์ของปริญญาานิพนธ์.....	1
1.3 ขอบเขตของปริญญาานิพนธ์.....	1
1.4 ขั้นตอนการศึกษา.....	1
1.5 ประโยชน์ที่คาดว่าจะได้รับ.....	2
<b>บทที่ 2 ทฤษฎีที่เกี่ยวข้อง.....</b>	<b>3</b>
2.1 กล่าวนำ.....	3
2.2 ระบบควบคุม.....	3
2.2.1 ประเภทของระบบควบคุม.....	3
2.3 ทฤษฎีการควบคุมแบบฟีดแบ็ค.....	4
2.3.1 การควบคุมแบบป้อนกลับด้วยตัวควบคุมฟีดแบ็ค.....	4
2.3.2 การคำนวณหาค่าพารามิเตอร์ของตัวควบคุมแบบฟีดแบ็ค.....	9
2.4 ทฤษฎีทางเดินราก (Root Locus).....	13
2.4.1 แนวคิดการวิเคราะห์ทางเดินราก.....	13
2.4.1.1 ปัญหาการควบคุม (The Control System Problem).....	13
2.4.1.2 แนวคิดการวิเคราะห์ทางเดินราก.....	14
2.4.1.3 การร่างเส้นทางเดินราก (Sketching The Root Locus).....	15
2.5 เครื่องมือวิเคราะห์และออกแบบระบบหนึ่งอินพุตหนึ่งเอาต์พุต (SISO Tool).....	17
2.5.1 เมนูบาร์ (Menu Bar).....	17
2.5.2 ทูลบาร์ (Tool Bar).....	20
2.5.3 เมนูที่เกิดจากการคลิกขวา (Right-Click Menu).....	21
<b>บทที่ 3 การประยุกต์ใช้งานฮาร์ดแวร์และซอฟต์แวร์.....</b>	<b>22</b>
3.1 กล่าวนำที่ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้.....	22
3.2 การทำงานของฮาร์ดแวร์ในกระบวนการ.....	22
3.3 องค์ประกอบด้านฮาร์ดแวร์.....	22

## สารบัญ (ต่อ)

	หน้า
3.3.1 กระบวนการวัดระดับน้ำ.....	22
3.3.1.1 ถังบรรจุน้ำ (Tank).....	22
3.3.1.2 วาล์วควบคุมอัตราการไหล (Flow Control Valve).....	23
3.3.1.3 สายยาง (Tube).....	23
3.3.1.4 ถังพักน้ำ (Reservoir Tank).....	23
3.3.1.5 อุปกรณ์ตรวจวัดระดับน้ำชนิด D/P Transmitter .....	24
3.3.1.6 Signal Converter.....	25
3.3.1.7 มอเตอร์ปั้มน้ำ .....	25
3.3.1.8 ชุดขับปั้มน้ำ .....	26
3.3.2 ส่วนของการอินเตอร์เฟสระหว่างคอมพิวเตอร์กับกระบวนการ.....	29
3.3.2.1 บอร์ดไมโครคอนโทรลเลอร์ รุ่น ET Base ARM 7024.....	29
3.4 องค์ประกอบด้านซอฟต์แวร์.....	34
3.4.1 ไมโครคอนโทรลเลอร์ ARM 7 พื้นฐานการเรียนรู้ภาษาซี.....	34
3.4.1.1 การประกาศตัวแปร.....	37
3.4.1.2 การประกาศตัวแปรพร้อมกับค่าเริ่มต้น.....	37
3.4.1.3 การประกาศตัวแปรแบบค่าคงที่.....	37
3.4.1.4 ตัวแปรแบบอาร์เรย์.....	37
3.4.1.5 เครื่องหมายตัวดำเนินการ.....	38
3.4.1.6 รูปแบบการเขียนคำสั่ง .....	40
3.4.1.7 คำสั่งเงื่อนไขและการทำซ้ำ.....	41
3.4.1.7.1 คำสั่งเงื่อนไข (Condition).....	41
3.4.1.7.2 การทำซ้ำ (Loop).....	43
3.4.1.8 Standard Function และ User Define Function .....	43
3.4.2 โปรแกรมคอมไพเลอร์ Keil uVision3.....	44
3.4.3 การใช้งานโปรแกรม LabVIEW .....	45
3.4.3.1 พาเนล.....	45
3.4.3.2 บล็อกไดอะแกรม.....	46
3.4.3.3 ไอคอนและจุดเชื่อมต่อ.....	48
3.4.3.4 เมนูคำสั่ง.....	48
3.4.3.5 Palette .....	50
3.4.3.6 เครื่องมือ (Tool).....	52
3.4.3.7 การเปิดและการบันทึกไฟล์.....	53
3.4.3.8 รูปแบบโครงสร้างการควบคุมการทำงานของโปรแกรม.....	54
3.4.3.9 พื้นฐานการใช้งานอาร์เรย์.....	60
3.4.3.10 พื้นฐานการใช้งานคลัสเตอร์.....	66

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการเรียนเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านธุรกิจ  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้นำไปเผยแพร่ในที่สาธารณะ

## สารบัญ (ต่อ)

	หน้า
3.4.3.11 พื้นฐานการใช้งาน Charts และ Graph Waveform Charts.....	69
3.4.3.12 พื้นฐานเกี่ยวกับสตริง .....	72
3.4.3.13 การสร้างสตริง .....	74
3.4.3.14 องค์ประกอบต่างๆใน Palette PID Control Toolkit.....	72
<b>บทที่ 4 วิธีการดำเนินงาน.....</b>	<b>79</b>
4.1 กล่าวนำ .....	79
4.2 กระบวนการที่ใช้ในการทดลอง.....	79
4.3 การออกแบบโปรแกรมสำหรับบอร์ดไมโครคอนโทรลเลอร์รุ่น ET Base ARM 7024 สำหรับกระบวนการ .....	80
4.4 การออกแบบโปรแกรมสำหรับควบคุม.....	80
4.5 การหาแบบจำลองทางคณิตศาสตร์ของกระบวนการที่ถูกควบคุม .....	81
4.6 การหาพารามิเตอร์พีไอดีด้วยวิธี Root Locus Technique.....	83
4.7 การออกแบบโปรแกรมสำหรับการจำลองกระบวนการ.....	85
4.8 การทดลองการควบคุมโดยใช้ตัวควบคุมแบบพีไอ.....	86
4.9 สรุปผล.....	87
<b>บทที่ 5 สรุปผลการวิจัยและข้อเสนอเชิงทฤษฎีจากการวิจัย.....</b>	<b>88</b>
5.1 สรุปผลการทดลอง .....	88
5.2 ปัญหาและอุปสรรค.....	88
5.3 ข้อเสนอแนะ .....	89
<b>บรรณานุกรม .....</b>	<b>90</b>
<b>ภาคผนวก.....</b>	<b>91</b>

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# สารบัญตาราง

ตารางที่	หน้า
2.1 แสดงค่าพารามิเตอร์ของตัวควบคุมแบบต่างๆตามวิธี Process Reaction Curve.....	11
2.2 แสดงค่าพารามิเตอร์ของตัวควบคุมแบบต่างๆตามวิธี Ultimate Method .....	12
3.1 ตารางแสดงคุณสมบัติของมอเตอร์ปั้มน้ำ .....	26
3.2 ตารางแสดงคุณสมบัติของ Switching Power Supply.....	28
3.3 เครื่องหมายตัวดำเนินการทางคณิตศาสตร์.....	38
3.4 เครื่องหมายตัวดำเนินการด้านการเปรียบเทียบ .....	38
3.5 เครื่องหมายตัวดำเนินการกระทำระดับบิต.....	39
3.6 เครื่องหมายตัวดำเนินการเพิ่มและลดค่า.....	39
3.7 เครื่องหมายตัวดำเนินการเลื่อนบิต (ซ้าย,ขวา).....	39
3.8 ตัวกำหนดชนิดข้อมูลของภาษาซี .....	39
3.9 แสดง Back slash Codes.....	74



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# สารบัญภาพ

รูปที่	หน้า
2.1 แผนภาพระบบเปิด .....	3
2.2 แผนภาพระบบปิด .....	4
2.3 ตัวควบคุมแบบพี.....	5
2.4 ตัวควบคุมแบบไอ.....	6
2.5 ตัวควบคุมแบบดี.....	7
2.6 ตัวควบคุมแบบพีไอ.....	7
2.7 ตัวควบคุมแบบพีดี .....	8
2.8 ตัวควบคุมแบบพีไอดี .....	9
2.9 แสดงค่าพ่วงเกินที่ต้องการเมื่อใช้การปรับด้วยวิธี Ziegler-Nichols .....	9
2.10 แผนภาพกระบวนการทำ Step Response .....	10
2.11 แสดงผลตอบสนองรูปตัว S เมื่อปรับโดยใช้วิธี Process Reaction Curve.....	10
2.12 แสดงผลตอบสนองเวลาเกิดการแกว่งอย่างต่อเนื่องเมื่อปรับโดยใช้วิธี Ultimate Method....	11
2.13 บล็อกไดอะแกรมของระบบปิดเมื่อเพิ่มแกนลงใน Forward Path.....	13
2.14 เครื่องมือในการวิเคราะห์และออกแบบระบบควบคุมแบบหนึ่งอินพุตหนึ่งเอาต์พุต.....	17
2.15 แสดงเมนูบาร์ในกลุ่มชุดคำสั่ง .....	17
2.16 แสดงเมนูเกี่ยวกับ EDIT.....	18
2.17 แสดงเมนู VIEW และ Dialog Box ของโพลวงรอบปิดของระบบควบคุม .....	18
2.18 แสดงแผนภาพในการวิเคราะห์ระบบควบคุมทั้งสิ้นี่แผนภาพ .....	19
2.19 แสดง Dialog Box ของข้อมูลระบบ .....	19
2.20 เมนูย่อยของตัวชดเชยและ Dialog Box ของคำสั่งรูปแบบและการเก็บค่าการออกแบบ .....	19
2.21 เมนูย่อยของการวิเคราะห์และกราฟแสดงผลตอบสนองของระบบควบคุม.....	20
2.22 แสดงโครงสร้างทั้งสามรูปแบบในการชดเชยระบบควบคุม.....	20
2.23 เมนูที่เกิดจากการคลิกขวา .....	21
3.1 ถังบรรจุน้ำวัดอะคริลิค.....	23
3.2 วาล์วควบคุมอัตราการไหล.....	23
3.3 ถังพักน้ำ .....	23
3.4 D/P Transmitter ยี่ห้อ Rousemount รุ่น 3051CD .....	24
3.5 การติดตั้ง D/P Transmitter เพื่อใช้ในการวัดระดับน้ำ.....	24
3.6 ช่วงความดันแตกต่างของถังน้ำ .....	25
3.7 Signal Converting ยี่ห้อ M-unit รุ่น DS-24-H.....	25
3.8 มอเตอร์ปั้มน้ำของบริษัท API Instrument รุ่น P6 10.0006.....	26
3.9 ชุดขับปั้มน้ำ.....	26
3.10 กระจับน้ํางานของชุดขับปั้มน้ำ .....	27
3.11 การเชื่อมต่อในอุปกรณ์เพื่อนำไปใช้งาน .....	27
3.12 Switching Power Supply .....	28

## สารบัญภาพ (ต่อ)

รูปที่	หน้า
3.13 บอร์ดทดลองไมโครคอนโทรลเลอร์ ARM 7 .....	29
3.14 แสดงการจัดเรียงขาสัญญาณของ Character LCD มาตรฐาน.....	33
3.15 แสดงวงจรเชื่อมต่อกับ RS232.....	34
3.16 โครงสร้างของภาษาซี.....	35
3.17 การเปิดโปรแกรม uVision3 .....	44
3.18 การสร้างไฟล์โปรเจกต์และเลือก ARM7 เบอร์ ADUC7024.....	44
3.19 การเขียน Source Code เพิ่มเข้าไปใน Source File.....	45
3.20 การสร้างโปรเจกต์และ Hex ไฟล์สำหรับ PROM.....	45
3.21 แสดงหน้าต่างพาเนล .....	46
3.22 แสดงหน้าต่างบล็อกไดอะแกรม.....	46
3.23 แสดงส่วนประกอบต่างๆของบล็อกไดอะแกรม .....	47
3.24 แสดงการเชื่อมต่อสัญญาณ.....	48
3.25 การใช้โครงสร้างของคำสั่งแบบ Shortcut .....	49
3.26 แสดงหน้าต่างคำสั่งแบบ Pull Down.....	49
3.27 แสดง Tools Palette .....	50
3.28 แสดง Control Palette.....	51
3.29 แสดง Function Palette.....	52
3.30 แสดงทูลบาร์ในหน้าต่างพาเนล.....	52
3.31 แสดงการเริ่มต้นใช้งานโปรแกรม .....	53
3.32 แสดงไดอะล็อกบล็อกสำหรับเปิดใช้งานโปรแกรม LabVIEW.....	54
3.33 แสดงโครงสร้างการควบคุม.....	54
3.34 แสดงโครงสร้างของ For Loop .....	55
3.35 แสดงโครงสร้างของ While Loop.....	56
3.36 แสดงโครงสร้างของ Shift Register .....	56
3.37 แสดงการทำงานของ Shift Register.....	57
3.38 แสดงการใช้งาน Feedback Node.....	57
3.39 แสดงโครงสร้างของ Case Structure .....	58
3.40 แสดงโครงสร้างของ Flat และ Stack Sequence Structures.....	59
3.41 การ Add Sequence Local ของ Stack Sequence Structures.....	59
3.42 การสร้างโปรแกรมจากอุปกรณ์ทั่วไป.....	59
3.43 โครงสร้างและการใช้งาน Formula Node.....	60
3.44 แสดงการเพิ่มอินพุตและเอาต์พุตของ Formula Node.....	60
3.45 แสดง Array One Dimension (1-D).....	61
3.46 การเรียกใช้งานอาร์เรย์ .....	61
3.47 แสดงอาร์เรย์สร้างจาก Control และ Indicator.....	62

## สารบัญภาพ (ต่อ)

รูปที่	หน้า
3.48 ขั้นตอนการสร้างอาร์เรย์จาก Control และ Indicator .....	62
3.49 เทอร์มินอลของ Array Shell.....	62
3.50 แสดงวิธีการเพิ่มมิติของอาร์เรย์.....	63
3.51 แสดงฟังก์ชันอาร์เรย์.....	64
3.52 แสดงการใช้งาน Array Size Function.....	64
3.53 แสดงการใช้งาน Intialize Array Function.....	64
3.54 แสดงการใช้งาน Build Array Function.....	65
3.55 แสดงการใช้งาน Subset Array Function .....	65
3.56 แสดงการใช้งาน Index Array Function.....	66
3.57 แสดงการใช้งาน Index Array Function กับอาร์เรย์สองมิติ.....	66
3.58 แสดงการใช้งานคลัสเตอร์.....	67
3.59 แสดงการสร้างคลัสเตอร์ด้วย Control และ Indicator .....	67
3.60 แสดงการใช้งาน Bundle Function .....	68
3.61 แสดงการใช้งาน Bundle Function .....	69
3.62 แสดงการใช้งาน Unbundle Function .....	69
3.63 แสดงวิธีการเรียกใช้งาน Waveform Chart .....	70
3.64 แสดงคุณสมบัติของ Waveform Chart .....	70
3.65 แสดงคุณสมบัติของ Waveform Graph .....	71
3.66 แสดงคุณสมบัติของ Waveform Graph .....	71
3.67 แสดงคุณสมบัติของ XY Graph (a),(b).....	72
3.68 การเข้าถึง String Control และ String Indicator .....	73
3.69 การเพิ่ม Scrollbar ให้กับบล็อก String Indicator และ String Control .....	73
3.70 แสดงตัวอย่างการเข้ารหัสแบบต่างๆ.....	74
3.71 โครงสร้างโปรแกรมที่มี Format into String เป็นส่วนประกอบ.....	75
3.72 กำหนดรูปแบบสตริง.....	76
3.73 โครงสร้างโปรแกรมที่มี Scan From String เป็นส่วนประกอบ.....	76
3.74 แสดงการเรียง Menu Function ขึ้นมาใช้งาน.....	77
3.75 แสดงพาเนลไอคอนของพีไอดี.....	77
4.1 ไดอะแกรมของกระบวนการ .....	79
4.2 กระบวนการสำหรับการทดลอง .....	80
4.3 โปรแกรม Keil uVision3 ที่ใช้เขียนภาษาซี .....	80
4.4 บล็อกไดอะแกรมของโปรแกรมส่วนที่ใช้ควบคุม .....	81
4.5 โปรแกรมส่วนที่ใช้ควบคุม.....	81
4.6 ผลตอบสนองของกระบวนการจากสัญญาณแบบขั้นบันได (Step Response).....	82
4.7 ทางเดินรากขณะที่ยังไม่มีตัวควบคุม .....	83

## สารบัญญภาพ (ต่อ)

รูปที่	หน้า
4.8 ทางเดินรากของระบบที่มีตัวควบคุมแบบพีไอ .....	84
4.9 ผลตอบสนองของระบบที่มีตัวควบคุมแบบพีไอ .....	84
4.10 แสดงค่าตัวควบคุมแบบพีไอที่ได้จากโปรแกรม.....	85
4.11 บล็อกไดอะแกรมโปรแกรมสำหรับจำลองกระบวนการ .....	85
4.12 โปรแกรมสำหรับจำลองกระบวนการ .....	86
4.13 ผลที่ได้จากการจำลอง .....	86
4.14 ผลที่ได้จากกระบวนการจริง .....	87



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# บทที่ 1

## บทนำ

### 1.1 ความสำคัญของปริญญานิพนธ์

ปัจจุบันมีธุรกิจทางอุตสาหกรรมเกิดขึ้นมากมาย วิวัฒนาการและเทคโนโลยีของเครื่องมือวัดทางอุตสาหกรรมและกระบวนการในอุตสาหกรรมก็เติบโตเร็วขึ้นเช่นกัน ซึ่งปัจจุบันธุรกิจทางอุตสาหกรรมส่วนใหญ่จะให้การควบคุมเป็นแบบอัตโนมัติ หนึ่งใน การควบคุมอย่างหนึ่งที่สำคัญคือกระบวนการควบคุมระดับของเหลวในถังซึ่งมีความสำคัญเป็นอย่างมาก เนื่องจากในบางอุตสาหกรรมต้องใช้ของเหลวบางชนิดเป็นองค์ประกอบหลักของกระบวนการ ในการติดตั้งเครื่องมือวัดและการควบคุมนั้นต้องมีประสิทธิภาพ มิเช่นนั้นอาจก่อให้เกิดอันตรายต่อผู้ปฏิบัติงาน รวมทั้งอุปกรณ์และกระบวนการอันเป็นเหตุให้ต้นทุนการผลิต การซ่อมบำรุงและการประกันชีวิตและทรัพย์สินของพนักงานเพิ่มขึ้น ดังที่กล่าวมานี้วิศวกรจึงมีหน้าที่ออกแบบตัวควบคุมให้เหมาะสมกับกระบวนการ ผู้ศึกษาจึงได้นำเสนอการออกแบบตัวควบคุมระดับน้ำ แบบสองถังใบบนที่ต่อแบบซีรี่กัน อันจะก่อให้เกิดแนวคิดต่อผู้ที่ต้องการศึกษาและผู้สนใจการออกแบบตัวควบคุมกระบวนการต่อไป

### 1.2 วัตถุประสงค์ของปริญญานิพนธ์

1. ศึกษาการใช้งานและการเขียนโปรแกรมบน LabVIEW
2. ศึกษาการจำลองกระบวนการระดับน้ำแบบสองถัง
3. ศึกษาการออกแบบตัวควบคุมกระบวนการระดับน้ำแบบสองถังโดยใช้โปรแกรม
4. ศึกษาการเชื่อมต่อคอมพิวเตอร์กับกระบวนการระดับน้ำแบบสองถัง

### 1.3 ขอบเขตของปริญญานิพนธ์

1. จัดทำโปรแกรมจำลองและควบคุมกระบวนการระดับน้ำสองถัง
2. จัดทำ Website พร้อม วิดีโอสาธิตการทำงานของระบบและโปรแกรม
3. จัดทำอุปกรณ์เชื่อมต่ออุปกรณ์การวัดและชุดขับเคลื่อนปั้มน้ำ
4. จัดทำรายงาน พร้อม Power point ในการนำเสนอโครงงาน

### 1.4 ขั้นตอนการศึกษา

ปริญญานิพนธ์ฉบับนี้ได้แบ่งเนื้อหาออกเป็น 5 บทด้วยกัน ได้แก่

บทที่1 บทนำ กล่าวถึงความเป็นมา วัตถุประสงค์ของการศึกษา ขอบเขตการทำปริญญานิพนธ์ ขั้นตอนของการศึกษา และประโยชน์ที่คาดว่าจะได้รับ

บทที่2 ทฤษฎีที่เกี่ยวข้องกับปริญญานิพนธ์ กล่าวถึงชนิดของระบบควบคุมและทฤษฎีทั้งหมดที่เกี่ยวข้อง ได้แก่ ทฤษฎีของโมเดลอ้างอิง ทฤษฎีการควบคุมแบบพีไอดี ทฤษฎีทางเดินรอก และเครื่องมือวิเคราะห์และออกแบบระบบหนึ่งอินพุตหนึ่งเอาต์พุต (SISO Tool)

เอกสารนี้เป็นเอกสารลิขสิทธิ์ของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี ห้ามเผยแพร่โดยไม่ได้รับอนุญาต  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งยังมีเหตุผลบางประการที่ต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่3 การประยุกต์ใช้งานฮาร์ดแวร์และซอฟต์แวร์ กล่าวถึงองค์ประกอบทั้งหมดของกระบวนการ การเชื่อมต่ออุปกรณ์ต่างๆ และโปรแกรมที่ใช้สำหรับการทดลอง

บทที่4 วิธีการดำเนินงาน กล่าวถึงการสร้างโปรแกรมจำลองกระบวนการ โปรแกรมสำหรับควบคุม การหาสมการคุณลักษณะของกระบวนการ และวิธีการทดลอง พร้อมทั้งนำเสนอผลการทดลอง

บทที่5 สรุปผลการวิจัยและข้อเสนอแนะ เป็นการสรุปปัญหาที่พบในโครงงานวิจัยพร้อมทั้งเสนอแนะ แนวทางการแก้ปัญหาและแนวทางการพัฒนา

## 1.5 ประโยชน์ที่คาดว่าจะได้รับ

กระบวนการใดๆ ในอุตสาหกรรมล้วนสามารถเปลี่ยนแปลงพารามิเตอร์ของกระบวนการได้ ตลอดเวลา อันเนื่องมาจากปัจจัยต่างๆ เช่น สิ่งรบกวนภายนอก ความร้อน การสั่นสะเทือน เป็นต้น สิ่งเหล่านี้ล้วนมีผลทำให้ค่าพารามิเตอร์ของกระบวนการเปลี่ยนแปลงไป เพราะฉะนั้นค่าพารามิเตอร์ของ ตัวควบคุมจึงมีความจำเป็นที่จะต้องปรับค่าให้เหมาะสมกับค่าพารามิเตอร์ของกระบวนการที่ เปลี่ยนแปลงด้วย

ซึ่งในปฏิญญาพันธนี้จึงมีสมมุติฐานว่า ถ้าสามารถออกแบบตัวควบคุมให้สามารถปรับตัวเอง ตามการเปลี่ยนแปลงได้ จะเป็นผลดีกับระบบควบคุมอย่างมาก เนื่องจากจะทำให้สามารถควบคุมค่า เป้าหมายให้เป็นไปตามที่เราต้องการมากที่สุด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทที่ 2

# ทฤษฎีที่เกี่ยวข้องและเครื่องมือที่ใช้วิเคราะห์

### 2.1 กล่าวนำ

ในส่วนของบทที่ 2 ในเรื่องของทฤษฎีที่เกี่ยวข้องกับโครงการ ซึ่งจะกล่าวด้วยเรื่องของ ทฤษฎีระบบควบคุม ทฤษฎีทางเดินรอก ทฤษฎีการควบคุมแบบพีโอดี และเครื่องมือวิเคราะห์และออกแบบระบบหนึ่งอินพุตหรือเอาต์พุต (SISO Tool) เพื่อให้ผู้อ่านและผู้ที่มีสนใจมีความรู้ทางด้านระบบควบคุม และระบบควบคุมแบบพีโอดี นอกจากนี้ยังสามารถหาค่าพารามิเตอร์พีโอดีได้จากทฤษฎีทางเดินรอกด้วยโปรแกรม MATLAB เพื่อที่สามารถนำมาใช้ในการออกแบบและควบคุมกระบวนการให้มีประสิทธิภาพมากยิ่งขึ้น

### 2.2 ระบบควบคุม

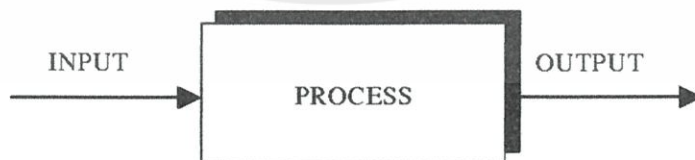
ระบบควบคุม คือ รูปแบบของระบบใด ๆ ที่มีการจัดองค์ประกอบต่าง ๆ ภายในระบบ เพื่อให้มีผลตอบสนอง ของระบบเป็นไปตามที่ต้องการ ส่วนมากอาศัยพื้นฐานทฤษฎีระบบเชิงเส้นมาช่วยในการวิเคราะห์พิจารณา ถึงความสัมพันธ์ระหว่างเหตุผล (Cause-effect) ของแต่ละองค์ประกอบของระบบไม่ว่าระบบควบคุมนี้จะมี ความซับซ้อนเพียงใดก็ตาม พื้นฐานของระบบควบคุมจะมีองค์ประกอบสำคัญ 3 ส่วน ดังนี้

- วัตถุประสงค์ของการควบคุม (Input)
- กระบวนการ, ขั้นตอน, หลักที่ใช้ในการควบคุม (Process)
- ค่าที่ได้รับจริง (Output)

#### 2.2.1 ประเภทของระบบควบคุม

มีอยู่ 2 ประเภทใหญ่ๆ คือ

1. ระบบควบคุมแบบเปิด (Open loop control system) เป็นระบบควบคุมที่ไม่มีการนำเอาต์พุตการป้อนกลับมาเปรียบเทียบกับอินพุตจึงง่ายต่อการสร้างและประหยัด แต่ค่าเอาต์พุตจะไม่มีผลต่อการควบคุม ขบวนการของระบบดังแสดงในรูปที่ 2.1

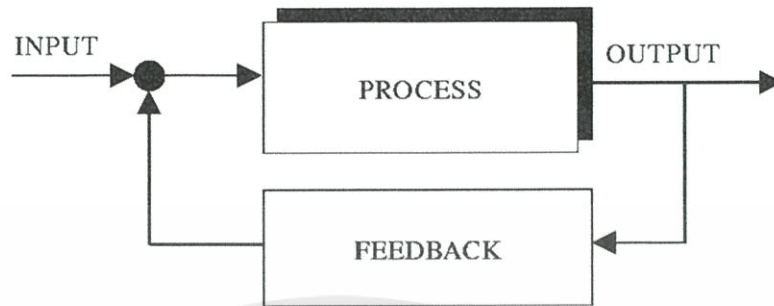


รูปที่ 2.1 แผนภาพระบบเปิด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

2. ระบบควบคุมแบบปิด (Closed loop control system) เป็นระบบควบคุมที่มีการป้อนกลับ (Feedback) โดยนำเอาเอาต์พุตมาเปรียบเทียบกับสัญญาณอินพุต ความแตกต่างที่

เกิดขึ้นจะถือเป็นความผิดพลาด เพื่อเอาสัญญาณนี้ป้อนเข้าระบบแล้วตัวควบคุมจะนำไปสร้างสัญญาณควบคุมใหม่ เพื่อลดความผิดพลาดที่เกิดขึ้นกับระบบดังแสดงในรูปที่ 2.2



รูปที่ 2.2 แผนภาพระบบปิด

จากระบบควบคุมทั้งสองประเภทจะกล่าวได้ว่า เมื่อต้องการจะควบคุมค่าเอาต์พุตของกระบวนการ จะมีการใส่ตัวควบคุม เพื่อคอยควบคุมเอาต์พุตของกระบวนการให้เป็นไปตามที่ต้องการ โดยในหัวข้อถัดไปจะเป็นการอธิบายถึง ทฤษฎีของตัวควบคุมพีไอดี ซึ่งเป็นตัวควบคุมแบบป้อนกลับที่ใช้กันอย่างกว้างขวาง รายละเอียดจะขอกกล่าวในหัวข้อถัดไป

### 2.3 ทฤษฎีการควบคุมแบบพีไอดี

เป็นระบบควบคุมแบบป้อนกลับ ซึ่งค่าที่นำไปใช้ในการคำนวณเป็นค่าความผิดพลาดที่หามาจากความแตกต่างของตัวแปรในกระบวนการและค่าที่ต้องการ ตัวควบคุมจะพยายามลดค่าผิดพลาดให้เหลือน้อยที่สุดด้วยการปรับค่าสัญญาณขาเข้าของกระบวนการ วิธีคำนวณของพีไอดี ขึ้นอยู่กับสามตัวแปรคือ ค่า  $K_p$ ,  $K_i$  และ  $K_d$  ค่า  $K_p$  กำหนดจากผลของความผิดพลาดในปัจจุบัน, ค่า  $K_i$  กำหนดจากผลบนพื้นฐานของผลรวมความผิดพลาดที่ซึ่งพืงผ่านพ้นไป, และค่า  $K_d$  กำหนดจากผลบนพื้นฐานของอัตราการเปลี่ยนแปลงของค่าความผิดพลาด น้ำหนักที่เกิดจากการรวมกันของทั้งสามนี้จะใช้ในการปรับกระบวนการ

#### 2.3.1 การควบคุมแบบป้อนกลับด้วยตัวควบคุมพีไอดี

##### 1. การควบคุมแบบ Proportional (P)

เทอม Proportional หรือเรียกว่า เทอมสัดส่วน ซึ่งจะเปลี่ยนแปลงเป็นสัดส่วนของค่าความผิดพลาด การตอบสนองของสัดส่วนสามารถทำได้โดยการคูณค่าความผิดพลาดด้วยค่าคงที่  $K_p$  หรือที่เรียกว่าอัตราขยายสัดส่วน เทอมของสัดส่วนจะเป็นไปตามสมการที่ (2-1)

$$M_p = K_p e(t) \quad (2-1)$$

เมื่อ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามนำที่ติดกับเอกสารนี้ไปเผยแพร่หรือทำซ้ำ และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

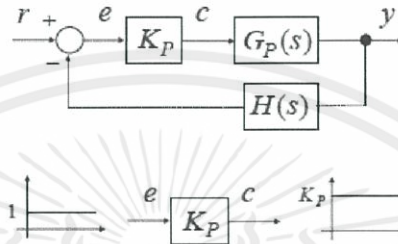
$M_p$  : สัญญาณขาออกของเทอมสัดส่วน

$K_p$  : อัตราขยายสัดส่วน

$e$  : ค่าความผิดพลาด =  $sp - pv$

$t$  : เวลา

ตัวควบคุมแบบนี้จะนำเอาสัญญาณค่าความผิดพลาดระหว่าง สัญญาณอ้างอิงกับสัญญาณเอาต์พุตมาเป็นอินพุตของตัวควบคุม แล้วตัวควบคุมจะทำการสร้างสัญญาณเอาต์พุตด้วยการขยายสัญญาณความผิดพลาดดังกล่าวด้วยค่าเกนของตัวควบคุม บล็อกไดอะแกรมและลักษณะของการประมวลผลสัญญาณเป็นดังรูปที่ 2.3



รูปที่ 2.3 ตัวควบคุมแบบพี

จุดเด่นของตัวควบคุมแบบนี้เมื่อนำไปใช้งานก็คือ การปรับค่าเกนให้สูงขึ้นจะมีผลทำให้ระบบมีผลตอบสนองที่เร็วขึ้น ปัญหาที่อาจจะเกิดขึ้นในการนำไปใช้งานก็คือ ตัวควบคุมแบบนี้จะไม่สามารถขจัดค่าความผิดพลาดในสภาวะคงตัวได้ แต่ก็สามารถทำให้ค่าความผิดพลาดดังกล่าวมีค่าน้อยลงได้ด้วยการปรับค่าเกนให้สูง ซึ่งในทางปฏิบัติแล้วการปรับค่าเกนให้สูงมากขนาดไหนเอาต์พุตที่ออกจริงๆ จากตัวควบคุมมักมีค่าจำกัด และการปรับเกนให้มีค่าสูงสำหรับระบบที่มีอันดับสูง อาจจะทำให้ได้ผลตอบสนองที่ไม่เป็นที่พึงประสงค์ เช่นการปรับเกนให้สูงขึ้นสำหรับระบบอันดับสอง ผลที่ตามมาก็คือค่าพุงเกินก็จะสูงขึ้นตามด้วย ซึ่งอาจจะเป็นอันตรายต่อระบบได้

## 2. การควบคุมแบบ Integral (I)

เทอม Integral หรือเรียกว่า เทอมปริพันธ์ จะเป็นสัดส่วนของขนาดความผิดพลาดและระยะเวลาของความผิดพลาด ผลรวมของความผิดพลาดในทุกช่วงเวลา (ปริพันธ์ของความผิดพลาด) จะให้ออฟเซตสะสมที่ควรจะเป็นในก่อนหน้า ความผิดพลาดสะสมจะถูกคูณโดยอัตราขยายปริพันธ์ ขนาดของผลของเทอมปริพันธ์จะกำหนดโดยอัตราขยายปริพันธ์  $K_i$  เทอมปริพันธ์จะเป็นไปตามสมการที่ (2-2)

$$M_I = K_i \int_0^t e(t) dt \quad (2-2)$$

เมื่อ

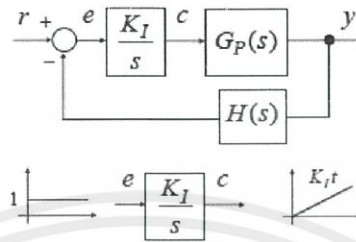
$M_I$  : สัญญาณขาออกของเทอมปริพันธ์

เอกสารนี้เป็นเอกสาร  $K_i$  ใช้งาน : อัตราขยายปริพันธ์ เพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น  $e$  อกทั้งนี้ : ความผิดพลาด =  $sp - pv$  ต้องอ้างอิงถึงเข้าของเอกสารทุกครั้งที่มีการนำไปใช้

$t$  : เวลา

ตัวควบคุมแบบนี้จะนำเอาสัญญาณความผิดพลาดระหว่าง สัญญาณอ้างอิงกับสัญญาณเอาต์พุตมาเป็นอินพุตของตัวควบคุม แล้วตัวควบคุมจะทำการสร้างสัญญาณเอาต์พุตด้วยการปริพันธ์ สัญญาณความผิดพลาดดังกล่าว แล้วคูณด้วยค่าเกนของตัวควบคุมบล็อกไดอะแกรมและลักษณะของการประมวลผลสัญญาณเป็นดังรูปที่ 2.4



รูปที่ 2.4 ตัวควบคุมแบบไอ

จุดเด่นของตัวควบคุมแบบนี้เมื่อนำไปใช้งานก็คือ ถ้านำไปใช้ ตัวควบคุมแบบนี้จะสามารถขจัดค่าความผิดพลาดในสภาวะคงตัวได้ ข้อด้อยที่อาจเกิดขึ้นในการนำไปใช้งานก็คือ ตัวควบคุมแบบนี้ไม่สามารถลดผลของการพุ่งเกินของผลตอบสนองได้ และการปรับเกนให้มึค่าสูง อาจจะทำให้ได้ผลตอบสนองที่ไม่เป็นที่พึงประสงค์ เช่นการปรับเกนให้สูงขึ้นอาจจะมีผลทำให้ผลตอบสนองของระบบ เกิดการแกว่งตัวได้

### 3. การควบคุมแบบ Derivative (D)

เทอม Derivative หรืออัตราการเปลี่ยนแปลงของความผิดพลาดจากกระบวนการซึ่งเป็นการคำนวณหาจากความชันของความ ผิดพลาดทุกๆเวลา (นั่นคือ เป็นอนุพันธ์อันดับหนึ่งสัมพันธ์กับเวลา) และคูณด้วยอัตราขยายอนุพันธ์  $K_d$  ขนาดของผลของเทอมอนุพันธ์ (บางครั้งเรียกว่าอัตรา) ขึ้นกับอัตราขยายอนุพันธ์  $K_d$  เทอมอนุพันธ์เป็นไปตามสมการที่ (2-3)

$$M_D = K_d \frac{d}{dt} e(t) \quad (2-3)$$

เมื่อ

$M_D$  : สัญญาณขาออกของเทอมอนุพันธ์

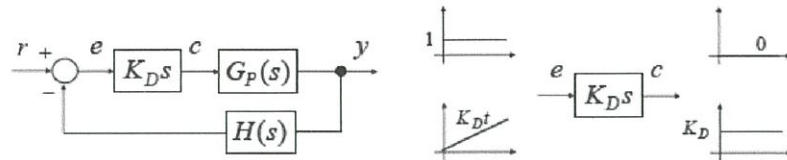
$K_d$  : อัตราขยายอนุพันธ์, ตัวแปรปรับค่าได้

$e$  : ความผิดพลาด =  $sp - pv$

$t$  : เวลา

ตัวควบคุมแบบนี้จะนำเอาสัญญาณความผิดพลาดระหว่าง สัญญาณอ้างอิงกับสัญญาณเอาต์พุตมาเป็นอินพุตของตัวควบคุม แล้วตัวควบคุมจะทำการสร้างสัญญาณเอาต์พุตด้วยการอนุพันธ์ สัญญาณความผิดพลาดดังกล่าว แล้วคูณด้วยค่าเกนของตัวควบคุมบล็อกไดอะแกรมและลักษณะของการประมวลผลสัญญาณเป็นดังรูปที่ 2.5

เอกสารนี้เป็นเอกสารลิขสิทธิ์ของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง  
ไม่ว่ากรณีใดๆ ห้ามสงวนลิขสิทธิ์ในเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.5 ตัวควบคุมแบบดี

จุดเด่นของตัวควบคุมแบบนี้เมื่อนำไปใช้งานก็คือ ตัวควบคุมแบบนี้ใช้สำหรับลดผลของการพุ่งเกินของผลตอบสนองได้ ลดผลตอบสนองที่มีการเปลี่ยนแปลงไปมาได้ แต่ต้องปรับค่าเกณฑ์ให้เหมาะสมด้วย ไม่เช่นนั้น อาจจะทำให้ได้ผลตอบสนองที่ไม่เป็นที่พึงประสงค์ ปัญหาที่อาจจะเกิดขึ้นในการนำเอาตัวควบคุมนี้ไปใช้งานก็คือ ตัวควบคุมแบบนี้จะไม่สามารถจัดค่าความผิดพลาดในสถานะคงตัวได้ และการใช้ตัวควบคุมนี้อาจจะทำให้ได้ผลตอบสนองที่ช้าลงได้

#### 4. การควบคุมแบบ Proportional-Integral (PI)

สำหรับการควบคุมแบบ Proportional-Integral จะเป็นการรวมกันของเทอม Proportional และ Integral ซึ่งเทอม Integral นั้นจะเป็นตัวที่ทำหน้าที่ขจัดออฟเซตที่เกิดจากเทอม Proportional แต่จะก่อให้เกิดการพุ่งเกินของผลตอบสนอง โดยสมการเป็นไปตามสมการที่ (2-4) และ (2-5)

$$M_{PI}(t) = \bar{m} + K_p e(t) + K_i \int_0^t e(t) dt \quad (2-4)$$

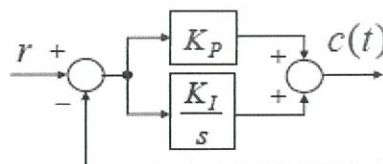
หรือ

$$M_{PI}(t) = \bar{m} + K_p e(t) + \frac{K_p}{T_i} \int_0^t e(t) dt \quad (2-5)$$

เมื่อ  $K_i = \frac{K_p}{T_i}$

$T_i =$  ค่าเวลา Integral

ส่วนลักษณะการทำงานก็คือการทำงานร่วมกันของตัวควบคุมแบบสัดส่วนและตัวควบคุมแบบปริพันธ์ โดยจะสามารถแสดงบล็อกไดอะแกรมและฟังก์ชันการทำงานดังรูปที่ 2.6



รูปที่ 2.6 ตัวควบคุมแบบพีไอ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเฉพาะที่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 5. การควบคุมแบบ Proportional-Derivative (PD)

สำหรับการควบคุมแบบ Proportional-Derivative เป็นการรวมกันของเทอม Proportional และเทอม Derivative โดยจะทำให้ผลตอบสนองของระบบรวดเร็วขึ้น แต่จะไม่มีผลโดยตรงต่อผลตอบสนองของระบบที่สภาวะคงที่ซึ่ง สมการเอาต์พุตของกริยา การควบคุมแบบพีดี จะแสดงดังสมการที่ (2-6) และ (2-7)

$$M_{PD}(t) = \bar{m} + K_p e(t) + K_d \frac{de(t)}{dt} \quad (2-6)$$

หรือ

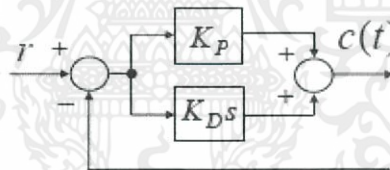
$$M_{PD}(t) = \bar{m} + K_p e(t) + K_p T_d \frac{de(t)}{dt} \quad (2-7)$$

เมื่อ

$$K_d = K_p T_d$$

$T_d =$  ค่าเวลา Derivative

ส่วนลักษณะการทำงานก็คือการทำงานร่วมกันของตัวควบคุมแบบสัดส่วนและตัวควบคุมแบบอนุพันธ์ บล็อกไดอะแกรมและฟังก์ชันการทำงานเป็นดังรูปที่ 2.7



รูปที่ 2.7 ตัวควบคุมแบบพีดี

### 6. การควบคุมแบบ Proportional – Integral- Derivative (PID)

สำหรับการควบคุมแบบ Proportional – Integral- Derivative เป็นการรวมกันของเทอม Proportional Integral และ Derivative เพื่อให้ผลตอบสนองของระบบควบคุมมีสมรรถนะ เป็นไปตามต้องการ จึงใช้กริยาการควบคุมทั้ง 3 แบบร่วมกัน ซึ่งจะทำให้ได้กริยาการควบคุมแบบ พีไอดี ดังสมการ (2-8) และ (2-9)

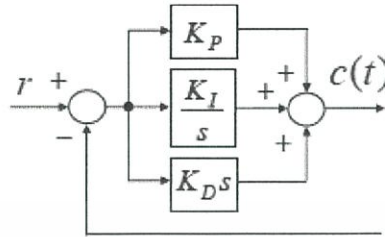
$$M_{PID}(t) = \bar{m} + K_p e(t) + K_p K_i \int_0^t e(t) dt + K_p K_d \frac{de(t)}{dt} \quad (2-8)$$

หรือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น

$$M_{PID}(t) = \bar{m} + K_p e(t) + \frac{K_p}{T_i} \int_0^t e(t) dt + K_p T_d \frac{de(t)}{dt} \quad (2-9)$$

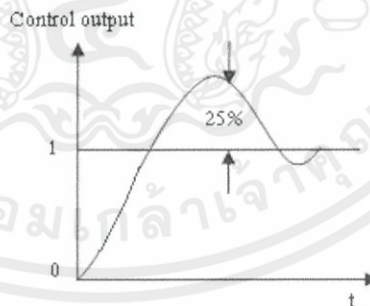
ส่วนลักษณะการทำงานก็คือการทำงานร่วมกันของตัวควบคุมแบบสัดส่วน ตัวควบคุมแบบปริพันธ์และตัวควบคุมแบบอนุพันธ์ บล็อกไดอะแกรมและฟังก์ชันการทำงานเป็นดังรูปที่ 2.8



รูปที่ 2.8 ตัวควบคุมแบบพีไอดี

### 2.3.2 การคำนวณหาค่าพารามิเตอร์ของตัวควบคุมแบบพีไอดีจากผลตอบสนองของกระบวนการ

การนำตัวควบคุมพีไอดีไปใช้ในการควบคุมกระบวนการต่างๆ จำเป็นต้องทำการปรับค่าพารามิเตอร์ของตัวควบคุมพีไอดี ให้เหมาะสม เพื่อให้ได้ผลตอบสนองที่ดีตามความต้องการโดยการหาค่าพารามิเตอร์ของตัวควบคุมพีไอดีนี้สามารถทำได้หลายวิธี ซึ่งบางวิธีอาจพิจารณาได้จากผลตอบสนองของกระบวนการ จากการทดสอบกระบวนการหรืออาศัยประสบการณ์จากตัวควบคุม บางวิธีอาจจะอาศัยทฤษฎีทางคณิตศาสตร์ และส่วนใหญ่จะพิจารณาจากเงื่อนไขในรูปแบบของโดเมนเวลามากกว่าในโดเมนความถี่ ซึ่งวิธี Ziegler-Nichols นี้เป็นวิธีที่ได้รับความนิยมมากที่สุด ในทางปฏิบัติการปรับค่าพารามิเตอร์ของตัวควบคุมพีไอดีโดยวิธีของ Ziegler-Nichols การหาค่า  $K_p$ ,  $K_i$  และ  $K_d$  จะขึ้นอยู่กับคุณลักษณะของผลตอบสนองชั่วคราวของกระบวนการที่ถูกควบคุม โดยจะมีอยู่ 3 วิธีด้วยกัน ซึ่งในแต่ละวิธีนั้นจะมีจุดมุ่งหมายที่จะทำให้ผลตอบสนองต่อเวลาของกระบวนการต่ออินพุตแบบขั้นบันได ที่มีค่าพุ่งเกินสูงสุดไม่เกิน 25% ดังแสดงในรูปที่ 2.9



รูปที่ 2.9 แสดงค่าพุ่งเกินที่ต้องการเมื่อใช้การปรับด้วยวิธี Ziegler-Nichols

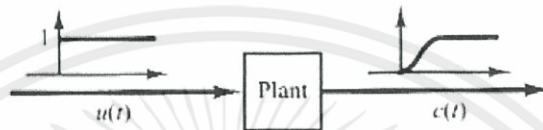
#### 1. วิธี Process Reaction Curve (การทดลองแบบลูปเปิด)

วิธีนี้จะหาแบบจำลองทางคณิตศาสตร์ของกระบวนการด้วยวิธี Process Reaction Curve จากผลตอบสนองของกระบวนการแบบลูปเปิดที่ถูกควบคุมด้วยสัญญาณอินพุตแบบขั้นบันได โดยที่กระบวนการในกรณีนี้จะไม่มี โพล ที่จุด ออริจินหรือไม่มี Dominant complex-conjugate poles ดังนั้นผลตอบสนองของเวลาจะได้ในลักษณะเป็นรูปตัว S (ถ้าผลตอบสนองของเวลาไม่มีลักษณะเป็นรูปตัว S วิธีนี้จะใช้ไม่ได้) จากผลตอบสนองต่อเวลาสามารถแยกพิจารณาเป็นค่าเวลาหน่วงของ

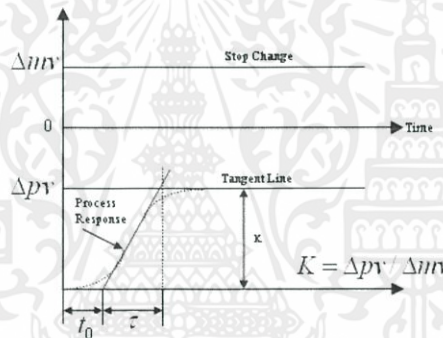
กระบวนการ (Delay time:  $t_0$ ) และเวลาคงที่ของกระบวนการ (Time Constant:  $\tau$ ) โดยลากเส้นสัมผัสกับจุดที่มีอัตราการเปลี่ยนแปลงสูงสุด เส้นนี้จะตัดกับแกนเวลา และเส้นตรงของเอาท์พุท  $C(t)$  ที่  $C(t) = K$

ดังนั้นฟังก์ชันถ่ายโอนจะประมาณได้ด้วยระบบอันดับหนึ่ง แบบมีการหน่วงเวลา (FOPDT) ดังสมการ (2-10)

$$G(s) = \frac{C(s)}{U(s)} = \frac{Ke^{-t_0s}}{\tau s + 1} \quad (2-10)$$



รูปที่ 2.10 แผนภาพกระบวนการหา Step Response



รูปที่ 2.11 แสดงผลตอบสนองรูปตัว S เมื่อปรับโดยใช้วิธี Process Reaction Curve

จากกราฟที่แสดงในรูปที่ 2.11 จะทำให้สามารถหาแบบจำลองทางคณิตศาสตร์ของกระบวนการได้โดยเริ่มต้นด้วยการหาค่าคงตัวทางเวลา, Delay time และอัตราการตอบสนอง (Process response) โดยอาศัยผลตอบสนองวงเปิด (open-loop response) ขั้นตอนการทดสอบเป็นดังนี้

1. ทำให้เกิดการเปลี่ยนแปลงแบบขั้น (step change) ด้วยขนาด  $\Delta PV$  ที่สัญญาณออกของตัวควบคุม (ซึ่งเป็นสัญญาณเข้าของระบบ) ในขณะเดียวกันก็หาค่า Delay time ดังรูปที่ 2.11 ค่า Delay time (หน่วยเป็นวินาที) จะเป็นระยะเวลานับจากจุดที่เริ่มการเปลี่ยนแปลงแบบขั้นของสัญญาณเข้า ไปยังจุดที่ระบบเริ่มตอบสนองต่อสัญญาณเข้านั้น

2. จากนั้นสัญญาณออกที่วัดได้จะมีค่าค่อยๆ เพิ่มขึ้น จนกระทั่งเข้าสู่สภาวะอยู่ตัวที่ค่าใหม่ ซึ่งเป็นผลมาจากการเปลี่ยนแปลงแบบขั้นที่สัญญาณออกของตัวควบคุมเส้นโค้งแสดงดังรูปที่ 2.11 อาจประมาณได้ว่าผลตอบสนองมีลักษณะเป็นการล่าหลังอันดับหนึ่ง (first-order lag)

เอกสารนี้เป็นเอกสาร 3. ค่าคงตัวทางเวลา  $\tau$  หาได้จากระยะเวลาที่สัญญาณออกเริ่มเปลี่ยนแปลง ไปจนถึงค่า 63.2% ของ  $\Delta PV$  (หรือเป็นส่วน  $1 - e^{-1}$  ของ  $\Delta PV$ ) โดย  $\Delta PV$  คือการเปลี่ยนแปลงของสัญญาณออก เมื่อเทียบกับค่าเดิม

4. ค่า  $K$  ซึ่งบ่งบอกถึงอัตราขยายของระบบในภาวะอยู่ตัวหาได้จาก  $K = \Delta PV / \Delta MV$

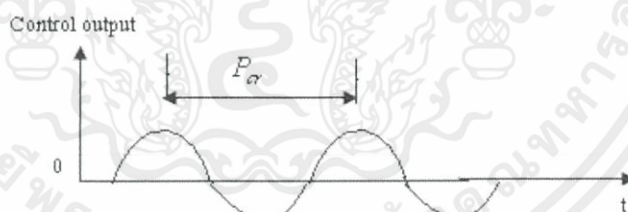
5. ค่าความชันของกราฟสัญญาณออก จะมีค่าเท่ากับ  $K/\tau$  เราจะเรียกค่าความชันนี้ว่า process reaction rate =  $R$  และจากรูปที่ 2.11 สามารถหาค่าพารามิเตอร์ของพีไอดีได้จากตารางที่ 2.1

ตารางที่ 2.1 แสดงค่าพารามิเตอร์ของตัวควบคุมแบบต่างๆตามวิธี Process Reaction Curve

Type of Controller	$K_p$	$T_i$	$T_d$
P	$\frac{1}{K} \left( \frac{\tau}{t_0} \right)$	—	—
PI	$\frac{0.9}{K} \left( \frac{\tau}{t_0} \right)$	$3.33t_0$	—
PID	$\frac{1.2}{K} \left( \frac{\tau}{t_0} \right)$	$2.0t_0$	$0.5t_0$

## 2. วิธี Ultimate Method (การทดลองลูบปิด)

วิธีนี้จะหาค่าพารามิเตอร์ของตัวควบคุมพีไอดี จากผลตอบสนองของกระบวนการแบบลูบปิด ที่ถูกควบคุมด้วยตัวควบคุมแบบพี ต่อสัญญาณอินพุตแบบขั้นบันได โดยปรับค่า  $K_p$  ไปเรื่อยๆ จนผลตอบสนองของกระบวนการเกิดการแกว่งอย่างต่อเนื่อง (Sustained Oscillations) ถ้าผลตอบสนองเวลาไม่เกิดการแกว่งอย่างต่อเนื่อง วิธีการนี้จะไม่ใช้ได้ ดังแสดงในรูปที่ 2.12



รูปที่ 2.12 แสดงผลตอบสนองเวลาเกิดการแกว่งอย่างต่อเนื่องเมื่อปรับโดยใช้วิธี Ultimate method

จากการหาค่าของ

$K_{cr}$  (Critical Gain) คือ อัตราการขยายที่ทำให้ผลตอบสนองเวลาเกิดการแกว่งอย่างต่อเนื่อง

$P_{cr}$  (Oscillation Period) คือ คาบเวลาของการแกว่งอย่างต่อเนื่อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.2 แสดงค่าพารามิเตอร์ของตัวควบคุมแบบต่างๆตามวิธี Ultimate Method

Type of Controller	$K_p$	$T_i$	$T_d$
P	$\frac{K_{cr}}{2}$	—	—
PI	$\frac{K_{cr}}{2.2}$	$\frac{t_{cr}}{1.2}$	—
PID	$\frac{K_{cr}}{1.7}$	$\frac{t_{cr}}{2}$	$\frac{t_{cr}}{8}$

การคำนวณหาค่าพารามิเตอร์ของตัวควบคุมพีไอดีโดยวิธีของ Ziegler-Nichols นั้นไม่ใช่เป็นค่าที่เที่ยงตรงที่จะนำไปใช้งานได้ทันที และมีค่าพุ่งเกินสูงสุดไม่เกิน 25 % ตามที่กล่าวไว้แต่ค่าที่ได้ อาจจะเพียงแค่อ้างอิงเท่านั้น หรือไม่ใกล้เคียงก็ตาม ผู้ควบคุมจะต้องทำการปรับค่าพารามิเตอร์เหล่านี้แบบละเอียด (Fine Tuning) อีกครั้งหนึ่ง

### 3. วิธี Trial and Error (วิธีการลองผิดลองถูก)

เป็นวิธีการที่เหมาะสมสำหรับผู้ที่มีประสบการณ์ในด้านการปรับพารามิเตอร์พีไอดีแล้ว โดยการควบคุมจะต้องเป็นระบบปิดและตำแหน่งการควบคุมจะต้องอยู่ในตำแหน่ง Automatic Control ส่วนวิธีการต่อไปนี่ก็คือวิธีการปรับแบบพีไอ

1. ให้ตัวควบคุมทำงานในรูปแบบ Proportional Control เพียงอย่างเดียว

2. ต่อไปทำการปรับค่า  $K_p$  จนกระทั่งค่า PV เข้าใกล้ค่าเป้าหมายซึ่งโดยทั่วไปก็จะต่ำกว่าค่าเป้าหมายเพียงเล็กน้อย

3. ต่อไปค่อยๆปรับค่า  $T_i$  เพิ่มขึ้นเพื่อชดเชยค่าออฟเซตที่เกิดขึ้นแล้วดูผลว่าค่า PV เข้าสู่เป้าหมายตามที่ต้องการหรือยัง ถ้ายังก็ค่อยๆปรับขึ้นอีกจนกระทั่งได้ตามต้องการโดยช่วงนี้ก็พยายามลด  $K_p$  เพื่อลดการแกว่ง

วิธีการปรับแบบพีไอดี

1. ทำตามวิธีการแบบพีไอ ทั้ง 3 ข้อ

2. ถ้าต้องการผลตอบสนองที่รวดเร็วขึ้นมักใช้ค่าอัตราส่วนระหว่างค่า Integral Time กับ Derivative Time เป็น 4/1 เป็นต้นไปพร้อมกับการเพิ่มลดของค่า  $K_p$  ไปด้วย เพื่อป้องกันการเกิด Overshoot และการเกิด Hunting

3. เนื่องด้วยวิธีการนี้เป็นความพึงพอใจเฉพาะบุคคลจึงยากที่จะกำหนดเป็นกฎเกณฑ์ตายตัว แต่วิธีการเริ่มต้นในการปรับนั้นมักเหมือนกัน

เอกสารนี้เป็นเอกสารสงวนลิขสิทธิ์ของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ไม่ควรเผยแพร่โดยไม่ได้รับอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 2.4 ทฤษฎีทางเดินราก (Root locus)

วิธีทางเดินราก เป็นวิธีการการนำเสนอโพลของระบบปิด (closed-loop pole) ในรูปแบบกราฟิก เมื่อมีการเปลี่ยนค่าพารามิเตอร์ในระบบนั้น วิธีทางเดินรากเป็นวิธีการที่มีประโยชน์มากสำหรับในการวิเคราะห์และออกแบบความเสถียรภาพและการตอบสนองชั่วขณะ (transient response) เนื่องจากระบบควบคุมแบบป้อนกลับนั้นยากต่อการทำความเข้าใจถึงคุณภาพของระบบ และต้องใช้คณิตศาสตร์ที่มีความซับซ้อนมาก ดังนั้นการแสดงเส้นทางเดินรากในรูปแบบกราฟิกจะช่วยอธิบายถึงสมรรถนะโดยคร่าวๆของระบบ ที่ใช้ในการวิเคราะห์และออกแบบความเสถียรภาพและการตอบสนองชั่วขณะ (transient response)

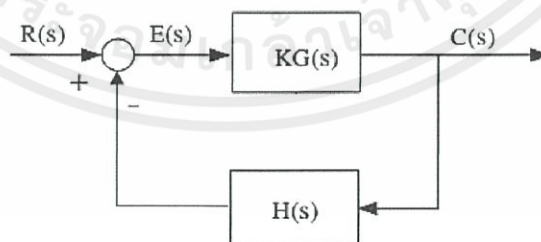
เส้นทางเดินรากสามารถใช้ทำการวิเคราะห์คุณภาพหรือสมรรถนะโดยคร่าวๆ ของระบบเมื่อพารามิเตอร์ใดๆ มีการเปลี่ยนแปลง เช่น ผลของการเปลี่ยนแปลงเกนต่อเปอร์เซ็นต์ของ Overshoot, settling time และค่าเวลาสูงสุด (Peak time) สามารถที่จะแสดงให้เห็นได้อย่างชัดเจน นอกจากนี้เส้นทางเดินรากยังสามารถแสดงกราฟิกของความเสถียรของระบบ ทำให้สามารถทราบขอบเขตความเสถียร, ขอบเขตที่ไม่เสถียรและเงื่อนไขที่เป็นสาเหตุให้ระบบอยู่สถานะภาพการแกว่งไม่หยุด (Oscillation) ได้อย่างชัดเจน

### 2.4.1 แนวคิดการวิเคราะห์ทางเดินราก

เส้นทางเดินของรากของสมการคุณลักษณะ (Characteristic equation) หรือการเปลี่ยนแปลงของโพลของระบบปิดจะพิจารณาจากการเปลี่ยนอัตราขยาย (Gain, K) ของฟังก์ชันถ่ายโอนของระบบเปิดจากค่าเริ่มต้น  $K=0$  จนถึงค่าสุดท้ายที่  $K=\infty$  การวิเคราะห์ทางเดินของรากเป็นวิธีที่จะทราบตำแหน่งของโพล และตำแหน่งของซีโรของระบบปิดจากสมการคุณลักษณะของระบบเปิดในระนาบเอส (s-plane)

#### 2.4.1.1 ปัญหาของระบบควบคุม (The Control System Problem)

โพลในระบบเปิดนั้นสามารถหาได้ง่าย และไม่เปลี่ยนแปลงเมื่อเกนหรืออัตราขยายของระบบเปลี่ยนแปลง ส่วนโพลของระบบปิดนั้นหาได้ยากและยิ่งกว่านั้นโพลในลูปปิดจะเปลี่ยนแปลง เมื่อเกนในระบบเปลี่ยนแปลง



รูปที่ 2.13 บล็อกไดอะแกรมของระบบปิดเมื่อเพิ่มเกนลงใน Forward Path

เอกสารนี้เป็นเอกสารลิขสิทธิ์ของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี  
 ลักษณะของระบบควบคุมแบบป้อนกลับดังแสดงในรูปที่ 2.13 ฟังก์ชันถ่ายโอนของระบบเปิด (Open Loop Transfer Function) คือ  $KG(s)H(s)$  โดยสามารถหาโพลของ  $KG(s)H(s)$  ได้ทันทีเมื่อระบบประกอบด้วยระบบอันดับหนึ่งหรือสองที่เชื่อมต่อกันแบบ Cascade และยังสามารถที่จะเปลี่ยนค่าของ K ได้โดยไม่มีผลต่อตำแหน่งของโพลของ  $KG(s)H(s)$

### 2.4.1.2 แนวคิดของการวิเคราะห์ทางเดินราก

เส้นทางเดินของรากของสมการคุณลักษณะ หรือการเปลี่ยนแปลงของโพลของระบบปิดจะพิจารณาจากการเปลี่ยนอัตราขยาย (Gain, K) ของฟังก์ชันถ่ายโอนของระบบเปิดจากค่าเริ่มต้น  $K=0$  จนถึงค่าสุดท้ายที่  $K=\infty$  การวิเคราะห์ทางเดินของรากเป็นวิธีที่จะทำให้ทราบตำแหน่งของโพลและตำแหน่งของซีโรของระบบปิดจากสมการคุณลักษณะของระบบเปิดในระนาบเอส (s-plane)

- นิยามของเส้นทางเดินราก (Definite the root locus)

จากบล็อกไดอะแกรมในรูปที่ 2.13 เป็นฟังก์ชันถ่ายโอนแบบปิด ซึ่งมีฟังก์ชันถ่ายโอนคือ

$$T(s) = \frac{G(s)}{1 + G(s)H(s)}$$

ฟังก์ชันถ่ายโอนแบบเปิด (Open Loop Transfer Function) คือ  $G(s)H(s)$  และสมการคุณลักษณะ คือ  $1+G(s)H(s)=0$

เมื่อเพิ่มแกนหรืออัตราขยายบน forward path ดังรูปที่ 2.13 ฟังก์ชันถ่ายโอนของระบบปิด (Closed Loop Transfer Function) คือ

$$T(s) = \frac{KG(s)}{1 + KG(s)H(s)}$$

สมการคุณลักษณะของระบบคือ

$$1 + KG(s)H(s)$$

เมื่อ K เป็นอัตราขยาย สามารถจัดรูปสมการใหม่จะได้ว่า

$$KG(s)H(s) = -1$$

เนื่องจากฟังก์ชันถ่ายโอนเป็นตัวแปรเชิงซ้อน จากสมการจะได้ว่าขนาดของ  $KG(s)H(s)=1$  และมุมมีค่าเท่ากับ 180 องศา เมื่อพิจารณาเฉพาะขนาด (gain) จะได้ว่า

$$|K||G(s)H(s)| = 1$$

$$|G(s)H(s)| = \frac{1}{|K|}$$

เอกสารนี้เป็นเมื่อพิจารณาเฉพาะมุม (phase) จะได้ว่า การศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$\begin{aligned} \angle KG(s)H(s) &= 180 + K360 \\ &= (2K+1)180 \text{ เมื่อ } 0 < K < \infty \end{aligned}$$

### 2.4.1.3 การร่างเส้นทางเดินราก (sketching the root locus)

#### 1. จำนวนเส้นทางเดินราก (Number of branch)

จำนวนของเส้นทางเดินของรากจะมีค่าเท่ากับจำนวนโพลของระบบเปิด (Open loop poles) พิจารณาในรูปที่ 2.13 ฟังก์ชันถ่ายโอนของระบบของระบบเปิดคือ  $KG(s)H(s)$  ดังนั้นจำนวนของเส้นทางเดินของรากจะมีค่าเท่ากับจำนวนโพลของฟังก์ชันถ่ายโอนของระบบ  $G(s)H(s)$

#### 2. สมมาตรของเส้นทางเดินของราก (Symmetry)

เส้นทางเดินของรากจะสมมาตรกับแกนจริง

#### 3. เส้นทางเดินของรากบนแกนจริง (Real-axis segments)

บนแกนจริงเมื่อ  $K > 0$  เส้นทางเดินของรากจะอยู่ทางด้านซ้ายของจำนวนคี่ของจำนวนโพลและซีโรที่มีค่าจำกัดของระบบเปิด

#### 4. จุดเริ่มต้นและจุดสิ้นสุดของเส้นทางเดินของราก (Starting and ending points)

เส้นทางเดินของรากจะเริ่มจากโพลของฟังก์ชันถ่ายโอนในระบบเปิด  $G(s)H(s)$  และสิ้นสุดที่ตำแหน่งซีโรของฟังก์ชันถ่ายโอนในระบบเปิด  $G(s)H(s)$  เราสามารถแสดงให้เห็นได้ดังนี้กำหนดให้

$$G(s) = \frac{N_G(s)}{D_G(s)}$$

$$H(s) = \frac{N_H(s)}{D_H(s)}$$

ฟังก์ชันถ่ายโอนของระบบปิดสามารถแสดงได้โดย

$$T(s) = \frac{KG(s)}{1 + KG(s)H(s)} = \frac{KN_G(s)D_H(s)}{D_G(s)D_H(s) + KN_G(s)N_H(s)}$$

เมื่อ  $K$  เข้าใกล้ 0

$$T(s) \approx \frac{KN_G(s)D_H(s)}{D_G(s)D_H(s) + \epsilon}$$

จะสังเกตเห็นว่าโพลของระบบปิดมีค่าเท่ากับโพลของระบบเปิดคือ  $D_G(s)D_H(s)$  สรุปได้ว่าจุดเริ่มต้นของเส้นทางเดินของรากอยู่ที่โพลของระบบเปิดคือ  $D_G(s)D_H(s)$

เมื่อ  $K$  เข้าใกล้อนันต์ (high gain)

$$T(s) \approx \frac{KN_G(s)D_H(s)}{\epsilon + KN_G(s)N_H(s)}$$

จะสังเกตเห็นว่าโพลของระบบปิดมีค่าเท่ากับซีโรของระบบเปิดคือ  $N_G(s)N_H(s)$  สรุปได้ว่าจุดเริ่มต้นของเส้นทางเดินของรากอยู่ที่ซีโรของระบบเปิดคือ  $N_G(s)N_H(s)$

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5. ลักษณะของเส้นทางเดินของรากที่อนันต์(Behavior at infinity)

- ถ้าฟังก์ชันเข้าสู่อินฟินิตี้แล้วเอสเข้าสู่อินฟินิตี้ ดังนั้นฟังก์ชันจะมีโพลที่อินฟินิตี้
- ถ้าฟังก์ชันเข้าสู่ซีโร่แล้วเอสเข้าสู่อินฟินิตี้ ดังนั้นฟังก์ชันจะมีซีโร่ที่อินฟินิตี้
- ทุกๆฟังก์ชันของเอสมีเลขที่เท่ากันของโพลและซีโร่

ทางเดินรากจะวิ่งเข้าสู่เส้นตรงไปจนถึงอินฟินิตี้ ยิ่งกว่านั้นความสมมาตรของแอสซิมโทสจะ ถูกแบ่งโดยแกนจริง  $\sigma_a$  และ  $\theta_b$  ตามสมการที่ (2-11)

$$\sigma_a = \frac{\sum \text{finite poles} - \sum \text{finite zeros}}{\text{Number of finite poles} - \text{Number of finite zeros}}$$

$$\theta_b = \frac{(2k + 1)\pi}{\text{Number of finite poles} - \text{Number of finite zeros}}$$

$$k = 0, \pm 1, \pm 2, \dots$$

6. จุดแยกออกจากแกนจริงและจุดเข้าหาแกนจริงของเส้นทางเดินของราก(Real-axis Breakaway and Break in points)

จุดแยกออกและจุดเข้าหา หาได้จากความสัมพันธ์

จุดแยกออกและจุดเข้าหา หาได้จากความสัมพันธ์

$$\sum_{i=1}^m \frac{1}{\sigma + z_i} = \sum_{i=1}^n \frac{1}{\sigma + p_i}$$

เมื่อ  $z_i$  และ  $p_i$  คือค่าลบของค่าโพลและค่าซีโร่ ของ  $G(s)H(s)$

7. จุดตัดแกนจินตภาพของเส้นทางเดินของราก(The  $j\omega$  - axis crossing)

ในบางกรณีเมื่อขนาด  $K$  เพิ่มขึ้น เส้นทางเดินของรากอาจจะตัดแกนจินตภาพซึ่งหมายความว่า ค่าอัตราขยาย  $K$  จะมีค่าสูงสุดที่ตำแหน่งของรากอยู่บนแกนจินตภาพ เพราะถ้าค่าของ  $K$  มากกว่าค่านี้จะทำให้ตำแหน่งของรากอยู่ทางด้านขวามือของระนาบ  $s$  ซึ่งทำให้ระบบไม่เสถียร ดังนั้นจุดตัดแกนจินตภาพของเส้นทางของรากก็คือจุดที่ค่าของ  $K$  มีค่าสูงสุดที่ระบบยังคงอยู่ในสภาวะเสถียร ค่าของ  $K$  และ  $\omega$  สูงสุดนั้น สามารถหาได้ 2 วิธีคือ

7.1 แทนค่า  $s=j\omega$  ในสมการคุณลักษณะ(Characteristic equation) จากนั้นใน ส่วนจริงและส่วนจินตภาพมีค่าเป็นศูนย์ แก้สมการเพื่อหาค่า  $K$  และ  $\omega$

7.2 นำสมการคุณลักษณะไปสร้างเป็นตารางเรย์ท์เฮอริวิตซ์ แล้วหาค่า  $K$  เมื่อแถว  $s$  เป็น 0 แก้สมการเพื่อหาค่า  $K$

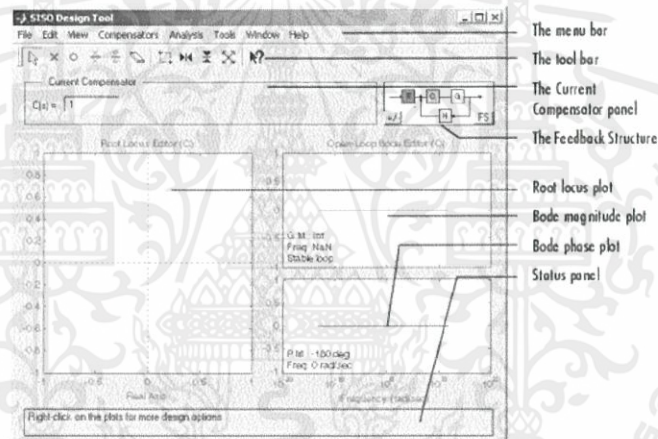
เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 2.5 เครื่องมือวิเคราะห์และออกแบบระบบหนึ่งอินพุตหนึ่งเอาต์พุต (SISO Tool)

เครื่องมือในการวิเคราะห์และออกแบบระบบควบคุมแบบหนึ่งอินพุตหนึ่งเอาต์พุตในโปรแกรม MATLAB มีเครื่องมือที่เรียกว่า SISO Tool ในการเรียกใช้งานโดยการพิมพ์คำสั่งดังนี้

```
>>sisotool
```

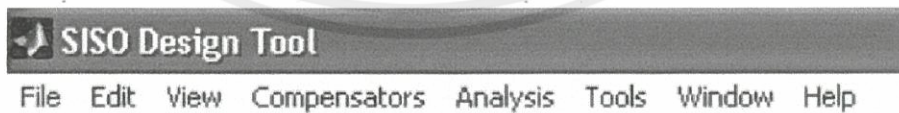
หลังจากเรียกใช้งานจะปรากฏดังรูป ซึ่งส่วนประกอบที่สำคัญในหน้าต่างนี้มีอยู่ 3 ส่วนได้แก่ เมนูบาร์ ทูลบาร์ และเมนูที่เกิดจากการคลิกขวา ในส่วนของเครื่องมือในการวิเคราะห์ระบบควบคุมมี แผนภาพทางเดินราก แผนภาพโบด และแผนภาพแบบอื่น ซึ่งในการวิเคราะห์ระบบควบคุมนั้นสามารถเปลี่ยนแปลงอัตราการขยายของระบบควบคุมหรือเปลี่ยนโพล ซีโรของตัวชดเชยหรือตัวควบคุมนั้นซึ่งสามารถที่จะเห็นผลการเปลี่ยนแปลงของการวิเคราะห์ที่ในทันทีและยังสามารถดูผลตอบสนองของระบบควบคุมที่ต้องการออกแบบว่าเป็นไปตามข้อกำหนดที่ต้องการได้ในทันที



รูปที่ 2.14 เครื่องมือในการวิเคราะห์และออกแบบระบบควบคุมแบบหนึ่งอินพุตหนึ่งเอาต์พุต

### 2.5.1 เมนูบาร์ (MENUBAR)

ในเมนูบาร์ประกอบไปด้วยกลุ่มคำสั่งที่ใช้งานอยู่ 8 กลุ่มดังต่อไปนี้ คือ



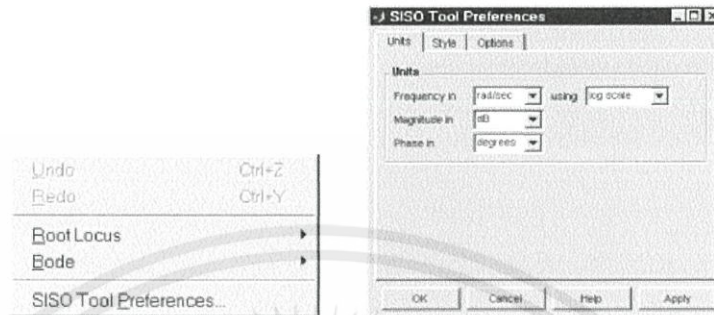
รูปที่ 2.15 แสดงเมนูบาร์ในกลุ่มชุดคำสั่ง

1. Edit เป็นกลุ่มคำสั่งเกี่ยวกับการแก้ไขค่าต่างๆในระบบควบคุม

- Change unit คำสั่งเปลี่ยนหน่วยของระบบเช่น ความถี่ rad/sec เป็น Hz

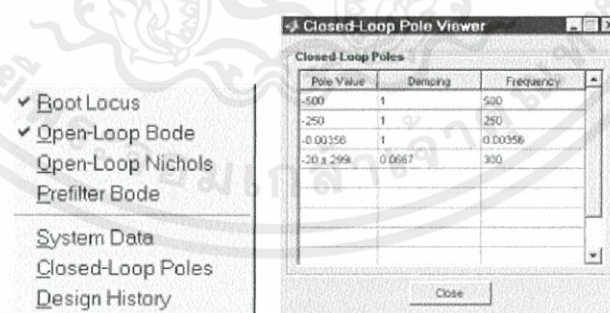
- Add plot grids, change font styles for titles, labels, etc., and change axes foreground colors คำสั่งในการเปลี่ยนรูปแบบเส้น สีเส้น และลักษณะตัวอักษร

- Change the compensator format คำสั่งเปลี่ยนฟอร์ของตัวชดเชย
- Show or hide system poles and zeros in Bode diagrams คำสั่งแสดงหรือไม่แสดงสัญลักษณ์ซีโรและโพลในแผนภาพโบด



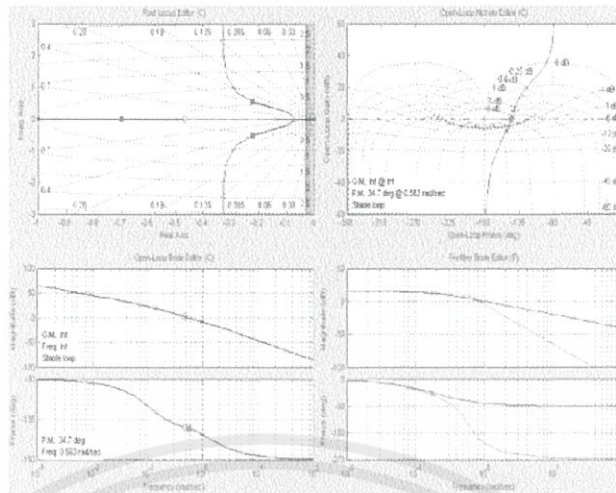
รูปที่ 2.16 แสดงเมนูเกี่ยวกับ EDIT

- VIEW เป็นกลุ่มคำสั่งเกี่ยวกับการดูข้อมูลของระบบควบคุม
  - Root Locus and Bode Diagrams คือ คำสั่งในการแสดงแผนภาพทางเดินรากและแผนภาพแบบโบด
  - Open-Loop Nichols คือคำสั่งในการแสดงแผนภาพนิโคล
  - Prefilter Bode คือคำสั่งในการแสดงกราฟโบดร่วมกับ Prefilter
  - System Data คือคำสั่งในการแสดงข้อมูลของระบบเช่น ข้อมูลตัวชดเชย ข้อมูลระบบควบคุม เป็นต้น
  - Closed-Loop Poles คือคำสั่งแสดงตำแหน่งโพลวงรอบปิดของระบบควบคุม
  - Design History คือคำสั่งแสดงข้อมูลที่เรากระทำกับระบบควบคุมตั้งแต่ต้น

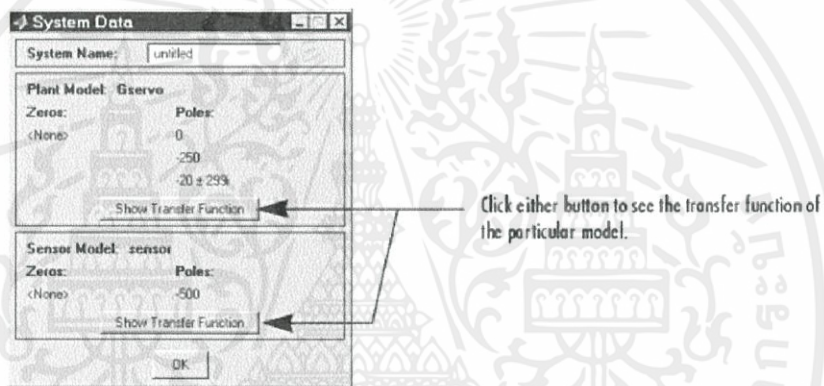


รูปที่ 2.17 แสดงเมนู VIEW และ Dialog box ของโพลวงรอบปิดของระบบควบคุม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

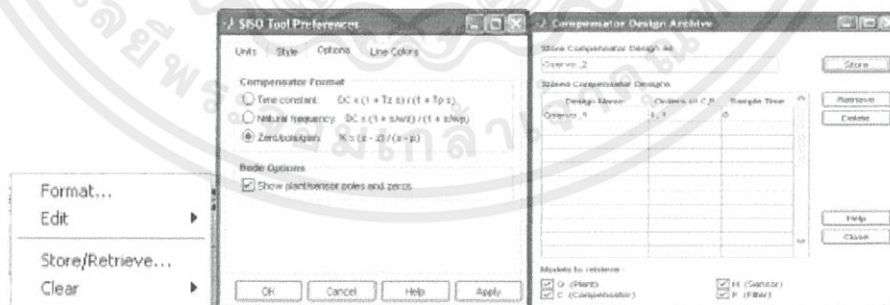


รูปที่ 2.18 แสดงแผนภาพในการวิเคราะห์ระบบควบคุมทั้งสี่แผนภาพ



รูปที่ 2.19 แสดง Dialog Box ของข้อมูลระบบ

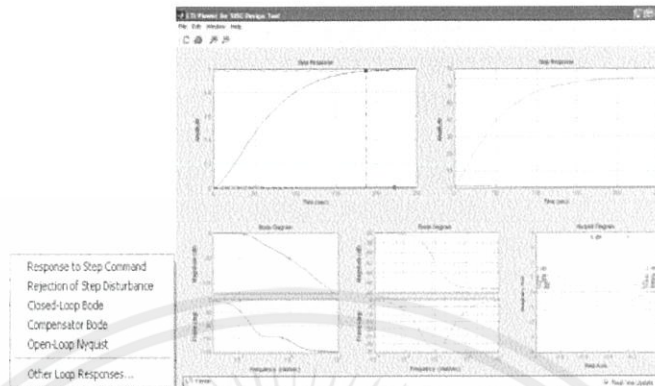
3. Compensation กลุ่มคำสั่งเกี่ยวกับตัวชดเชยระบบควบคุม



รูปที่ 2.20 เมนูย่อยของตัวชดเชยและ Dialog Box ของคำสั่งรูปแบบและการเก็บค่าการออกแบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. Analysis กลุ่มคำสั่งในการแสดงการวิเคราะห์ผลตอบสนองของระบบควบคุมและตัวชดเชยในรูปแบบกราฟในโดเมนเวลา

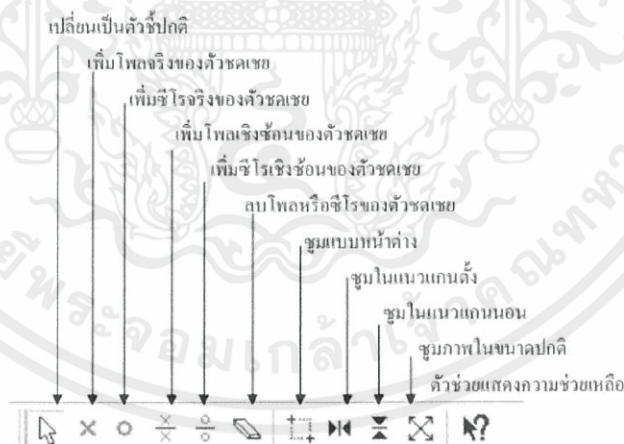


รูปที่ 2.21 เมนูย่อยของการวิเคราะห์และกราฟแสดงผลตอบสนองของระบบควบคุม

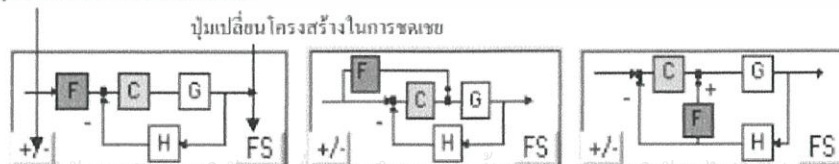
5. Tools กลุ่มคำสั่งที่เป็นเครื่องมือในการแปลงระบบและวางแผนภาพ Simulink
6. Window กลุ่มคำสั่งในการเลือกหน้าต่างที่ต้องการทำงาน
7. Help กลุ่มคำสั่งที่ใช้แสดงความช่วยเหลือในโปรแกรม

### 2.5.2 ทูลบาร์ (ToolBar)

มีคำสั่งในการใช้งานดังนี้



ปุ่มเปลี่ยนเครื่องหมายการป้อนกลับ



รูปที่ 2.22 แสดงโครงสร้างทั้งสามรูปแบบในการชดเชยระบบควบคุม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับบริการใช้งานเพื่อการศึกษาเท่านั้น มิอนุญาติให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งยังมีเหตุผลบางประการที่ต้องสงวนลิขสิทธิ์ของเอกสารทุกครั้งที่มีการนำไปใช้

### 2.5.3 เมนูที่เกิดจากการคลิกขวา (Right-Click Menu)



รูปที่ 2.23 เมนูที่เกิดจากการคลิกขวา

- ADD คือคำสั่งในการเพิ่มโพลหรือซีโรของตัวขดเซย
  - EDIT Compensator คือคำสั่งในการแก้ไขตัวขดเซย
  - Show คือ คำสั่งในการแสดงค่าข้อกำหนดในโดเมนความถี่
  - Design Constraint คือคำสั่งในการกำหนดเงื่อนไขข้อกำหนดในการออกแบบแบ่งเป็น 2 ส่วนคือ โดเมนเวลาและโดเมนความถี่
1. ข้อกำหนดในโดเมนเวลา ได้แก่ settling Time, Percent Overshoot, Damping Ratio, Natural Frequency
  2. ข้อกำหนดโดเมนความถี่ได้แก่ Phase Margin, Gain Margin, Close-loop Peak Gain
    - Grid คือคำสั่งในการแสดงกริด
    - Properties คือคำสั่งในการกำหนดรูปแบบในกราฟ เช่น ชื่อแกน หน่วย เป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทที่ 3

# การประยุกต์ใช้งานฮาร์ดแวร์และซอฟต์แวร์

### 3.1 กล่าวนำ

ในส่วนของบทที่ 3 การประยุกต์ใช้งานฮาร์ดแวร์และซอฟต์แวร์ จะเป็นบทที่กล่าวถึงรายละเอียดของอุปกรณ์ต่างๆที่นำมาใช้ในโครงงาน เช่น วาล์วควบคุมอัตราการไหล (Flow Control Valve), อุปกรณ์ตรวจวัดระดับน้ำ ชนิด D/P Transmitter (Range 0-250 inH2O), Signal Converter, มอเตอร์ปั้มน้ำ, Switching Power Supply เป็นต้น และรวมถึงข้อมูลรายละเอียดของซอฟต์แวร์ เช่น ไมโครคอนโทรลเลอร์ ARM7 พื้นฐานการเขียนภาษาซี (Basic C For ARM7), โปรแกรมคอมไพเลอร์ Keil uVision3 และ การใช้งานโปรแกรม LabVIEW เพื่อให้กระบวนการควบคุมสามารถทำงานได้อย่างมีประสิทธิภาพ อีกทั้งเพื่อให้ผู้ที่สนใจได้รับความรู้และข้อมูลทางด้านฮาร์ดแวร์และซอฟต์แวร์เพิ่มขึ้น

### 3.2 การทำงานของฮาร์ดแวร์ในกระบวนการ

การทำงานด้านฮาร์ดแวร์จะเริ่มทำงานจากวัดระดับน้ำโดยใช้ D/P Transmitter แล้วนำสัญญาณที่ได้จาก D/P Transmitter ต่อเข้ากับไมโครคอนโทรลเลอร์ แล้วนำมาประมวลผลโดยใช้คอมพิวเตอร์ผ่านพอร์ตอนุกรม และเมื่อประมวลผลแล้วจะส่งค่าที่ได้ไปยังไมโครคอนโทรลเลอร์ แล้วไมโครคอนโทรลเลอร์จะทำการสร้างสัญญาณเพื่อไปขับปั้มน้ำต่อไป

### 3.3 องค์ประกอบด้านฮาร์ดแวร์

องค์ประกอบด้านฮาร์ดแวร์มีส่วนประกอบต่างๆดังต่อไปนี้

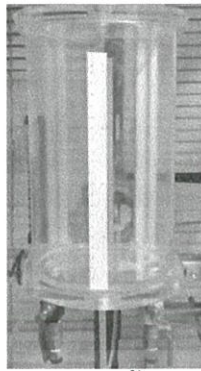
#### 3.3.1 กระบวนการวัดระดับน้ำ

ในการวัดระดับน้ำนั้นมีทั้งทางตรงและทางอ้อม ณ ที่นี้จะใช้วิธีการวัดแบบทางอ้อมคือใช้หลักการของการเปลี่ยนแปลงความดันเนื่องจากระดับน้ำ จากความสัมพันธ์  $P = \rho gh$  เมื่อมีระดับน้ำที่สูงขึ้นหรือลดลงก็จะทำให้ค่าความดันเปลี่ยนแปลงไป และนำความดันแตกต่างนี้ไปใช้งานในการวัดระดับน้ำต่อไป

##### 3.3.1.1 ถังบรรจุน้ำ (Tank)

ใช้วัสดุอะคริลิกในการทำถังเนื่องจากมีความแข็งแรงและวัสดุมีความโปร่งใส ทำให้สามารถสังเกตระดับน้ำได้ชัด มีเส้นผ่านศูนย์กลาง 15 เซนติเมตร ความสูง 25 เซนติเมตร ความจุ 4,418 ลูกบาศก์เซนติเมตร

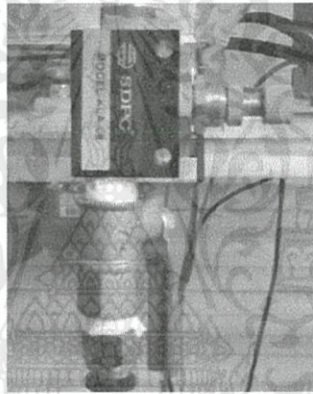
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.1 ถังบรรจุน้ำวัสดุอะคริลิก

### 3.3.1.2 วาล์วควบคุมอัตราการไหล (Flow Control Valve)

วาล์วควบคุมอัตราการไหล เป็นวาล์วชนิดแบบปรับหมุนด้วยมือ ซึ่งอัตราการไหลขึ้นอยู่กับ การปรับหมุน



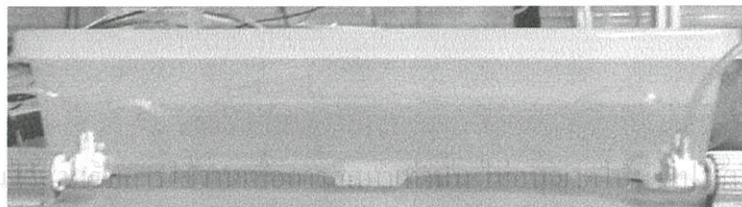
รูปที่ 3.2 วาล์วควบคุมอัตราการไหล

### 3.3.1.3 สายยาง (Tube)

ทำมาจากยาง ทำหน้าที่เป็นตัวส่งผ่านน้ำเข้าสู่ถังบรรจุน้ำและถังพักน้ำ มีเส้นผ่านศูนย์กลาง ของสายยาง 0.7 เซนติเมตร

### 3.3.1.4 ถังพักน้ำ (Reservoir Tank)

ทำมาจากถังพลาสติกทำหน้าที่พักน้ำเพื่อจ่ายน้ำให้กับถังบรรจุน้ำทั้ง 2 ถัง ปริมาตรความจุ ของถังพักน้ำ 26,629 ลูกบาศก์เซนติเมตร

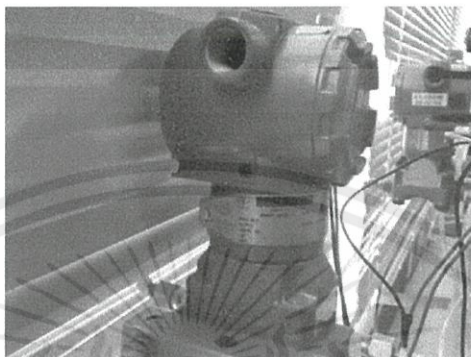


รูปที่ 3.3 ถังพักน้ำ

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี ระเบียบข้อบังคับการนำเอกสารไปใช้  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา

### 3.3.1.5 อุปกรณ์ตรวจวัดระดับน้ำ ชนิด D/P Transmitter

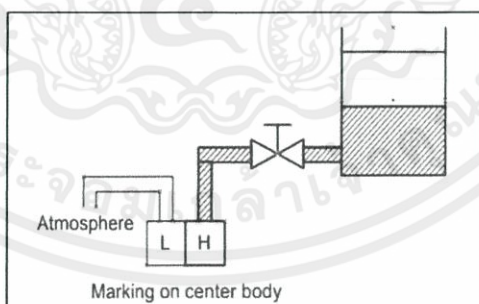
อุปกรณ์ที่ใช้ในการวัดระดับของน้ำในถัง จะใช้ Pressure Transmitter ยี่ห้อ Rosemount รุ่น 3051CD ซึ่งมีช่วงการวัด 0-250 inH<sub>2</sub>O โดยใช้วัดความดันด้านล่างของถังน้ำเพื่อวัดความดันซึ่งจะแปรผันตรงกับระดับความสูงของน้ำภายในถัง จากนั้นจึงส่งสัญญาณออกไป โดยสัญญาณที่ส่งออกไปจะเป็นสัญญาณมาตรฐาน 4-20 mA



รูปที่ 3.4 D/P Transmitter ยี่ห้อ Rosemount รุ่น 3051CD

จากรูปที่ 3.4 เป็นอุปกรณ์วัดระดับน้ำของ Rosemount รุ่น 3051CD โดยมีคุณสมบัติดังนี้

- Performance up to 0.075% accuracy with 100:1 turndown
- Coplanar platform enables integrated manifold, primary element and diaphragm seal solutions
- Calibrated spans/ranges from 0.1 inH<sub>2</sub>O to 4000 psi (0,25 mbar to 276 bar)
- 316L SST, Hastelloy C®, Monel®, Tantalum, Gold-plated Monel, or Gold-plated 316L SST process isolators



รูปที่ 3.5 การติดตั้ง D/P Transmitter เพื่อใช้ในการวัดระดับน้ำ

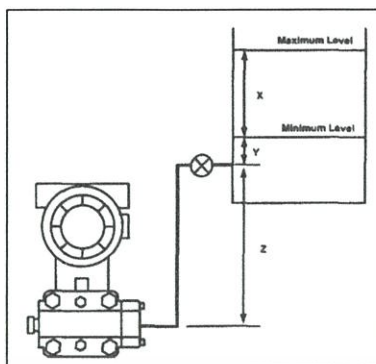
จากรูปที่ 3.5 เป็นการติดตั้ง D/P Transmitter วัดระดับของเหลวในถังโดยจะติดตั้ง D/P Transmitter ไว้ในระดับของถังแล้วต่อด้าน High ของ D/P Transmitter กับล่างของถังและด้าน

เอกสารนี้เป็นเอกสารลิขสิทธิ์ของ บริษัท อุตสาหกรรมปิโตรเลียมไทย จำกัด (มหาชน) การนำเอกสารนี้ไปใช้โดยไม่ได้รับอนุญาตถือว่าผิดกฎหมาย

Low ของ D/P Transmitter จะเปิดปล่อยว่างไว้กับบรรยากาศภายนอก (Atmosphere) เป็น

ไม่ว่ากรณีใดๆก็ตาม บริษัทฯ ขอสงวนสิทธิ์ในข้อมูลและข้อมูลที่เกี่ยวข้องกับการดำเนินงานของบริษัทฯ

ลักษณะการวัดแบบถังเปิด



รูปที่ 3.6 ช่วงความดันแตกต่างของถังน้ำ

การนำ D/P Transmitter มาใช้ในการตรวจวัดระดับน้ำในกระบวนการจะต้องมีการปรับเทียบเพื่อความแม่นยำ ของการตรวจวัดระดับน้ำ โดยการปรับซีโรและสเปน ที่ตัว D/P Transmitter ให้ที่ระดับน้ำในถังบรรจุน้ำ 0 เซนติเมตร ให้เอาท์พุตเท่ากับ 4 mA และที่ระดับน้ำในถังบรรจุน้ำ 20 เซนติเมตร เอาท์พุตเท่ากับ 20 mA

### 3.3.1.6 Signal Converter

Signal Converting ยี่ห้อ M-Unit รุ่น DS-24-H เป็นตัวแปลงสัญญาณไฟฟ้าเพื่อใช้ในการเปลี่ยนสัญญาณไฟฟ้าให้เหมาะสม และทำให้อุปกรณ์ต่างๆสามารถเชื่อมต่อเข้ากันได้ ซึ่งจะจ่ายออกเป็นกระแสไฟฟ้า 4-20 mA DC Current Loop

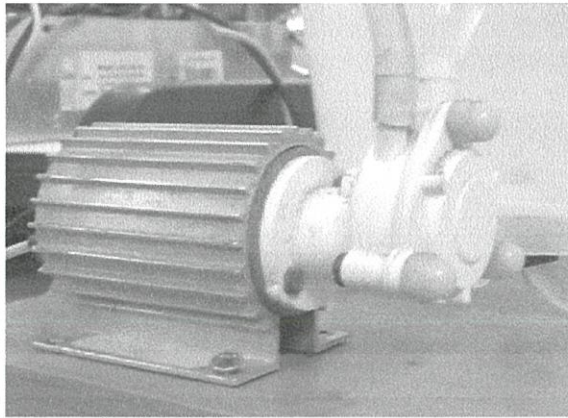


รูปที่ 3.7 Signal Converting ยี่ห้อ M-Unit รุ่น DS-24-H

### 3.3.1.7 มอเตอร์ปั้มน้ำ

ปั้มน้ำบริษัท API Instrument รุ่น P6 10.0006 สำหรับการควบคุมระดับน้ำ เป็นปั้มน้ำแบบเกียร์ปั้มซึ่งมีลักษณะเป็นเชิงเส้นที่ดีสำหรับการควบคุมอัตราการไหล ลักษณะของปั้มแสดงดังรูปที่ 3.8 และข้อมูลคุณสมบัติของปั้มแสดงดังตารางที่ 3.1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



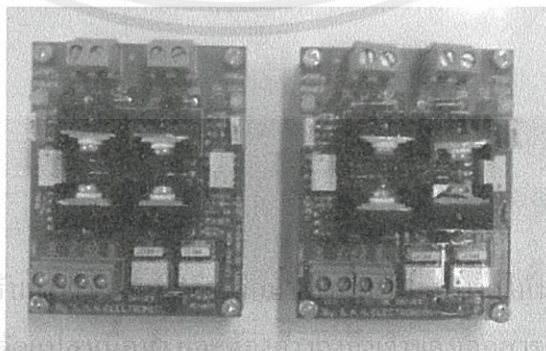
รูปที่ 3.8 มอเตอร์ปั๊มน้ำ ของบริษัท API Instrument รุ่น P6 10.0006

ตารางที่ 3.1 ตารางแสดงคุณสมบัติของมอเตอร์ปั๊มน้ำ

รายละเอียด	คุณสมบัติ
Max. Flow rate	2200 ml/min (Vdc)
Max. Suction	10 ft. H <sub>2</sub> O wet, 4" dry
Max. Pressure	20 psig (46 ft. H <sub>2</sub> O)
Port Size (OD)	4.8 mm (0.18") hose barb
Power Require (Vdc)	24 Vdc (up to 28 W)
Max. Operating Temp	93 °C (200 °F)
Ambient Fluid Temp	0 to 93 °C (32 to 200 °F)
Max. Viscosity	200 cps
Dimensions	88x81x92 mm (4x1.75x2.63)
Weight	1.4 kg (3 lb)

### 3.3.1.8 ชุดขับปั๊มน้ำ

เป็นชุดอุปกรณ์ที่รับสัญญาณจาก Microcontroller ARM7 เป็น Pulse Width Modulation (PWM) แล้วนำสัญญาณไปใช้ในการควบคุมมอเตอร์ปั๊มน้ำ ให้ปั๊มน้ำเข้าสู่กระบวนการซึ่งภายในชุดปั๊มน้ำจะประกอบไปด้วยส่วนประกอบต่างๆ ดังนี้



รูปที่ 3.9 ชุดขับปั๊มน้ำ

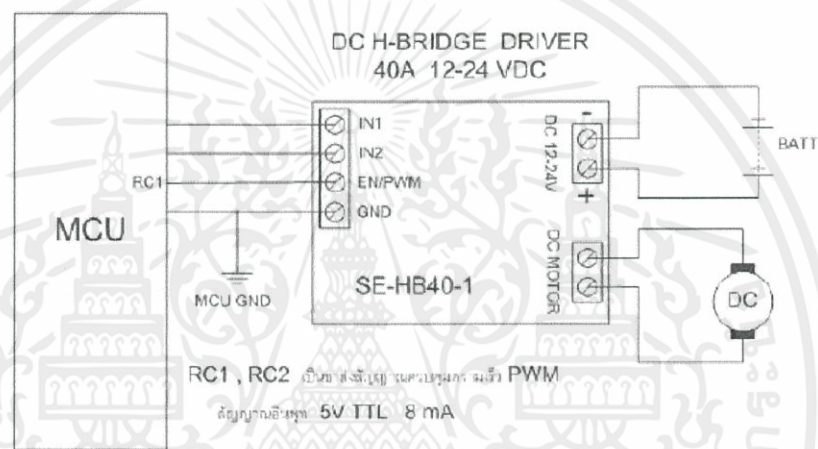
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้ภายในเท่านั้น ไม่ควรเผยแพร่โดยไม่ได้รับอนุญาต หากมีการนำออกไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกหรือเผยแพร่เอกสารฉบับนี้



รูปที่ 3.10 กระบวนการทำงานของชุดขับปั๊ม

- บอร์ดขับมอเตอร์ไฟฟ้ากระแสตรง

ใช้ในการควบคุมการทำงานของมอเตอร์ปั๊มน้ำ โดยรับสัญญาณ Pulse Width Modulation (PWM) จากไมโครคอนโทรลเลอร์เพื่อนำไปควบคุมมอเตอร์ดีซี โดยบอร์ดขับมอเตอร์ดีซีนี้เป็นบอร์ดสำเร็จรูป ทำงานโดยใช้มอสเฟต แบบ H-Bridge รุ่น SE-HB40-1



รูปที่ 3.11 การเชื่อมต่อในอุปกรณ์เพื่อนำไปใช้งาน

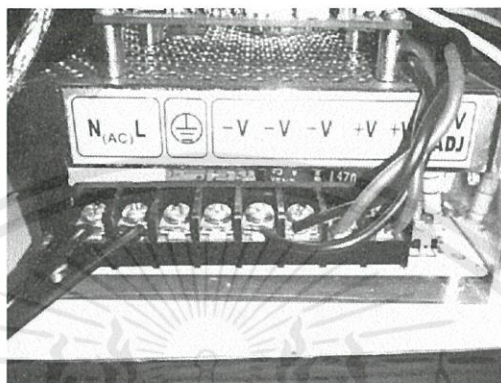
รายละเอียดทางเทคนิค

1. Output: Single motor driver:
  - Motor DC Supply 12-24 V 40A (Max.)
  - Full-Complementary Power MOSFET Driver With ultra-fast reverse recovery protection diodes
2. Input:
  - Full Opto-isolated input interface signals
  - 5V 8 mA TTL – Level
3. Drive Mode: independently with:
  - ON – OFF Control
  - Direction Control
  - Speed Control (PWM Drives)

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์การใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งยังมีให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- Switching Power Supply

เป็นแบบสวิตซ์ซิ่ง Input Signal 220 Ac Output Signal 12 Vdc 12 Amp Model S-145-12 ใช้สำหรับบอร์ด ARM7 บอร์ดไดรฟ์มอเตอร์และวงจร Signal Conditioning เป็นวงจรสำหรับปรับสภาพสัญญาณเพื่อให้เหมาะสมกับอุปกรณ์ที่ต่อร่วม ประกอบด้วยวงจรออปแอมป์ต่อแบบ Inverting และ อนุกรม กับ Inverting กับอีกชุดหนึ่ง เพื่อกลับเฟสให้แรงดันอยู่ทางด้านบวก



รูปที่ 3.12 Switching Power Supply

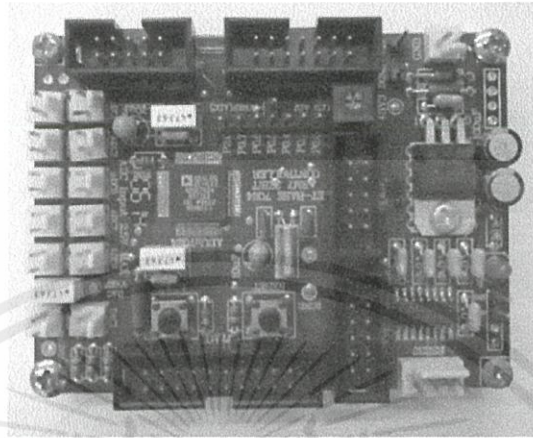
ตารางที่ 3.2 ตารางแสดงคุณสมบัติของ Switching Power Supply

รายละเอียด	คุณสมบัติ
Output DC Voltage	24 V
Output Rated Current	6.5 A
Output Current Range	0-6.5 A
Output Rated Power	156 W
Output Voltage Adjust Range	21-28 V
Input Voltage Range	85-132 VAC/170-264 VAC
Input Frequency Range	50-60 Hz
Input AC Current	2.0 Amax
Input Leakage Current	3.5 mA/240V
Overload	105-150% Rate Output Power
Overvoltage	30-34.8V
Working Temp.	-10 - +60 (Refer to Output Load derating Curve)
Dimension	199 x 110 x 50 mm.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.3.2 ส่วนของการอินเตอร์เฟสระหว่างคอมพิวเตอร์กับกระบวนการ

#### 3.3.2.1 บอร์ดไมโครคอนโทรลเลอร์ รุ่น ET-BASE ARM7024



รูปที่ 3.13 บอร์ดทดลองไมโครคอนโทรลเลอร์ ARM7

ET-BASE ARM7024 เป็นบอร์ดไมโครคอนโทรลเลอร์ในตระกูล ARM7 ขนาด 64 Pin ซึ่งเลือกใช้ไมโครคอนโทรลเลอร์ เบอร์ ADUC7024 ของ Analog Device เป็น MCU ประจำบอร์ด โดย MCU รุ่นนี้จะบรรจุอยู่ในตัวถังแบบ 64 Pin LQFP โดย MCU ตัวนี้จะมีจุดเด่น คือ ความสามารถในการเชื่อมต่อกับสัญญาณแบบ Analog ซึ่งมีทั้ง ADC ขนาด 12บิต จำนวน 10 ช่อง และ DAC ขนาด 12บิต จำนวน 2 ช่องนอกจากนี้แล้วความสามารถทางด้านของความเร็วในการประมวลผลก็ถือว่าไม่ด้อยกว่าตัวอื่นๆ ซึ่งสามารถทำงานได้ด้วยความเร็วสูงสุด 41.78 MHz โดยใช้ XTAL 32.768KHz ร่วมกับวงจรคูณความถี่แบบ PhaseLock Loop ภายในตัว MCU นอกจากนี้แล้วยังมีความเพียบพร้อมด้วยอุปกรณ์พื้นฐานต่างๆที่จำเป็นต่อการใช้งาน ไม่ว่าจะเป็นหน่วยความจำโปรแกรมแบบ Flash ขนาด 62 KByte และหน่วยความจำใช้งานแบบ RAM ซึ่งมีมากถึง 8 KByte ส่วนในด้านของอุปกรณ์ Peripheral นั้นก็นับว่าครบถ้วนเหมาะแก่การนำไปประยุกต์ใช้งานเกี่ยวกับการควบคุมและประมวลผลต่างๆได้เป็นอย่างดี โดยจะมีทั้ง SPI, UART, Watchdog, Timer/Counter, PWM โดยการออกแบบโครงสร้างของบอร์ดนั้นจะเน้นเรื่องขนาดของบอร์ดให้มีขนาดเล็กเพื่อให้ง่ายต่อการนำไปประยุกต์ใช้งาน และสะดวกต่อการพัฒนาโปรแกรม

- คุณสมบัติของบอร์ด

1. เลือกใช้ MCU ตระกูล ARM7 TDMI Core เบอร์ ADUC7024 ของ Analog Device เป็น MCU ประจำบอร์ด โดยเลือกใช้แหล่งกำเนิดสัญญาณนาฬิกาแบบ XTAL ค่า 32.768 KHz ซึ่งสามารถกำหนดการทำงานร่วมกับ Phase Lock Loop ให้ MCU สามารถประมวลผลด้วยความเร็วสูงสุดที่ 41.78 MHz ได้ด้วย โดยคุณสมบัติเด่นๆของ MCU ได้แก่

- มีหน่วยความจำ Flash สำหรับเขียนโปรแกรม 62KByte และมี RAM ขนาด 8Kbyte
- 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



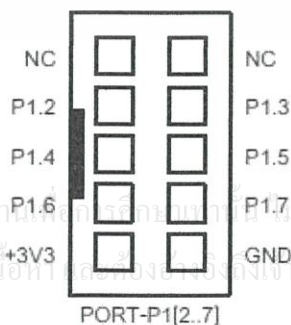
- หมายเลข 14 คือ Switch RESET ใช้สำหรับ Reset การทำงานของ CPU
- หมายเลข 15 คือ Switch LOAD ใช้ร่วมกับ RESET สำหรับสั่ง Download HEX ให้ MCU
- หมายเลข 16 คือ ขั้วต่อ ARM-JTAG สำหรับ Debug โปรแกรม
- หมายเลข 17 คือ ขั้วต่อ RS232 สำหรับใช้งานทั่วไป และ Download HEX ให้ MCU
- หมายเลข 18 คือ LED Power ใช้แสดงสถานะของแหล่งจ่ายไฟ +5VDC

### ● ขั้วต่อสัญญาณต่างๆ

สำหรับขั้วต่อสัญญาณของพอร์ต I/O จาก MCU นั้น จะถูกออกแบบและจัดเตรียมไว้ผ่านทางขั้วต่อแบบ IDE Header ขนาด 10Pin (2x5) จำนวน 3 ชุด คือ PORT-P1, PORT-P3 และ PORT-P4 ตามลำดับ ส่วน PORT-P0 และ PORT-P2 นั้นจะต่อเป็น Header ขนาด 1x7 ไร่ โดยที่ขั้วต่อสัญญาณแต่ละชุด จะประกอบไปด้วยสัญญาณของ I/O ที่เชื่อมต่อมาจากขาสัญญาณของ MCU โดยตรงทั้งหมด โดยจุดเชื่อมต่อกับสัญญาณภายนอกของบอร์ด 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 จะจะถูกเชื่อมต่อกับวงจร Line Driver (MAX232) สำหรับแปลงระดับสัญญาณจากระดับลอจิก TTL ของ MCU ให้เป็นสัญญาณแรงดันตามมาตรฐานของ RS232 โดยสัญญาณที่ได้รับการแปลงเป็นแบบ RS232 จะถูกเชื่อมต่อไปรอไว้ที่ขั้วต่อแบบ 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 บิต คือ 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-JATG
- ขั้วต่อสัญญาณ RS232

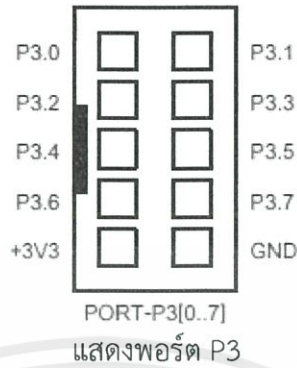
พอร์ต P1มีขนาด 6 บิต



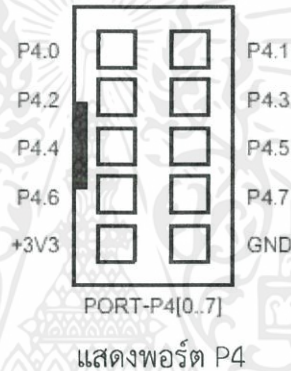
แสดงพอร์ต P1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานในกรณีศึกษาเท่านั้น มิอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาใดๆของเอกสารทุกครั้งที่มีการนำไปใช้

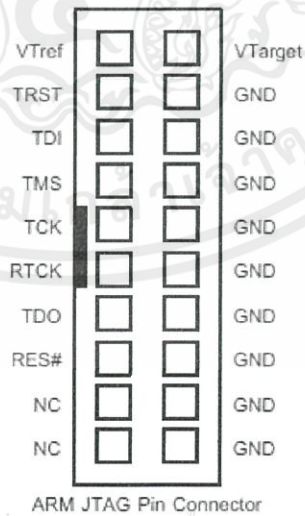
พอร์ต P3 มีขนาด 8 บิต



พอร์ต P4 มีขนาด 8 บิต

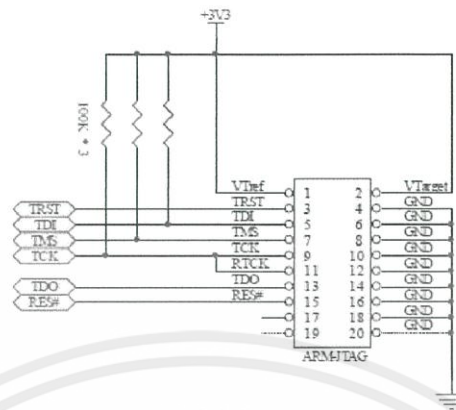


แสดงข้อต่อ ARM-JTAG

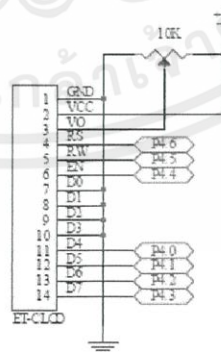
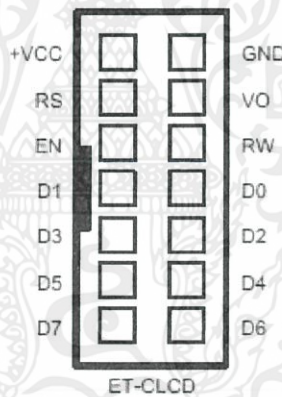


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แสดงวงจรส่วนที่ใช้ในการเชื่อมต่อกับ ARM-JTAG



พอร์ต CLCD ใช้กับ Character LCD โดยเชื่อมต่อแบบ 4 บิต โดยสัญญาณที่ใช้เชื่อมต่อกับ LCD จะเป็นสัญญาณชุดเดียวกับที่ต่อไปยังขั้วต่อของ PORT-P4 โดยในการเชื่อมต่อสายสัญญาณจากขั้วต่อของพอร์ต LCD ไปยังจอแสดงผล LCD นั้น ให้ยึดชื่อสัญญาณเป็นจุดอ้างอิง โดยให้ต่อสัญญาณที่มีชื่อตรงกันเข้าด้วยกันให้ครบทั้ง 14 เส้น

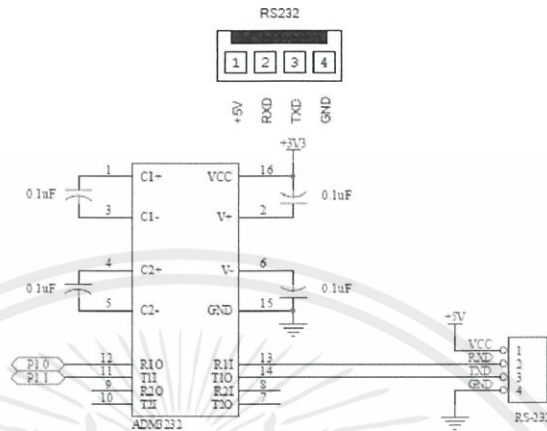


1	2	3	4	5	6	7	8	9	10	11	12	13	14
GND	+VCC	VO	RS	RW	EN	D0	D1	D2	D3	D4	D5	D6	D7

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับ... ไขข้อข้องใจด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

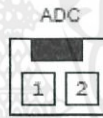
รูปที่ 3.14 แสดงการจัดเรียงขาสัญญาณของ Character LCD มาตรฐาน

พอร์ต RS232 เป็นสัญญาณ RS232 ซึ่งผ่านวงจรแปลงระดับสัญญาณ MAX232 เรียบร้อยแล้วสามารถใช้เชื่อมต่อกับสัญญาณ RS232 เพื่อรับส่งข้อมูล นอกจากนี้แล้วยังสามารถใช้งานร่วมกับ SwitchPSEN และ Switch RESET เพื่อทำการ Download แบบ Manual ได้ด้วย

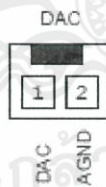


รูปที่ 3.15 แสดงวงจรที่เชื่อมต่อกับ RS232

แสดงขั้วต่อ ACD



แสดงขั้วต่อ DAC



### 3.4 องค์ประกอบด้านซอฟต์แวร์

ในการเขียนโปรแกรมนั้นต้องมีการศึกษาภาษาที่ใช้ในการเขียน ซึ่งได้แก่

#### 3.4.1 ไมโครคอนโทรลเลอร์ ARM7 พื้นฐานการเขียนภาษาซี (Basic C For ARM7)

ภาษาซีเป็นโปรแกรมภาษาระดับกลาง มีโครงสร้างการเขียนโปรแกรมเป็นแบบลำดับ ถูกพัฒนาขึ้นในปี 1972 ที่ AT&T Bell Lab เราสามารถใช้ภาษาซี มาเขียนเป็นคำสั่งต่างๆ ที่คอมพิวเตอร์สามารถเข้าใจได้ เพื่อสร้างงานประยุกต์ใช้งานในด้านต่างๆ เช่น โปรแกรมทางด้านงานควบคุม โปรแกรมระบบ สำหรับคอมพิวเตอร์ที่ใช้ในงานนี้คือคอมพิวเตอร์ CARM ของบริษัท Keil รุ่น

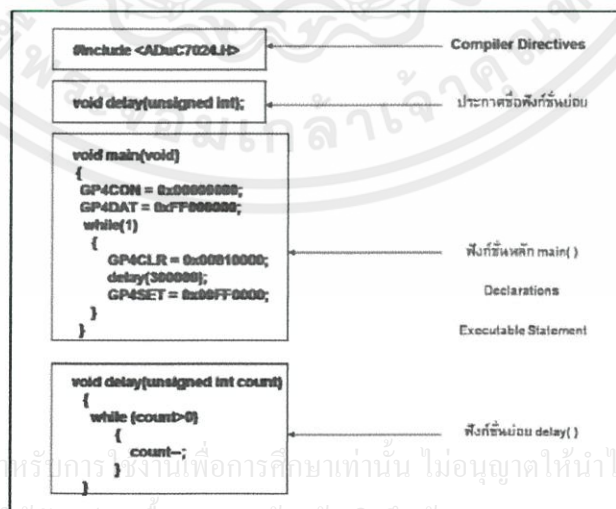
Evaluation ซึ่งได้อธิบายการติดตั้ง และการใช้งานพื้นฐานโดยเราจะศึกษาถึง โครงสร้างโปรแกรม ภาษาซี สำหรับ ARM7, ตัวแปรชนิดต่างๆและค่าคงที่, การประกาศตัวแปร, ตัวกำหนดชนิดข้อมูลของ ภาษาซี, รูปแบบการเขียนคำสั่ง

- ความหมายภาษาซี สำหรับ ARM7

ภาษาซีสำหรับ ARM7 คือ ภาษาที่ใช้เขียนโปรแกรมเพื่อสั่งงานและควบคุม ไมโครคอนโทรลเลอร์ ARM7ซึ่งภาษาซีสำหรับ ARM7 ถือว่าเป็นภาษาระดับกลางมีโครงสร้าง (Structure) และไวยากรณ์ (Syntax) เหมือนกับภาษาซีทั่วไปการทำงานของภาษาซีเริ่มจากใช้ Editor เขียนโปรแกรมเมื่อบันทึกเป็นไฟล์นามสกุล \*.C จากนั้นทำการคอมไพล์ถ้าหากไม่มีการผิดพลาดจะได้ไฟล์ \*.Obj แล้วทำการ Link ไฟล์ \*.Obj เข้ากับไฟล์ Libraries หากไม่มีข้อผิดพลาดจะได้ไฟล์ \*.EXE

- โครงสร้างโปรแกรมภาษาซีสำหรับ ARM7

1. Compiler Directive คือ ส่วนที่ใช้บอกคอมไพเลอร์ว่าให้ทำอะไรก่อน ก่อนที่คอมไพเลอร์ จะทำการคอมไพล์โปรแกรม เช่นการรวมไฟล์อื่นๆเข้ากับโปรแกรม การสร้างเงื่อนไขของการคอมไพล์ โปรแกรมจะขึ้นต้นด้วยเครื่องหมาย #
2. Functions คือ โปรแกรมย่อย (Sub Program) ชนิดหนึ่งสามารถทำงานได้อย่างหนึ่งเสร็จสิ้นภายในตัวเอง โดยที่ function อาจจะมีการรับค่าเข้า function และส่งกลับในชื่อของ function หรืออาจจะมีก็ได้โดยฟังก์ชัน ในภาษาซีแบ่งได้เป็น 2 ประเภทด้วยกัน คือ ฟังก์ชันที่เราสร้าง ขึ้นมาเอง และฟังก์ชัน ที่ภาษาซีสร้างมาเพื่อให้ใช้ในการทำงานทั่วไป อย่างเช่น ฟังก์ชันสำหรับแสดงผล ฟังก์ชันที่ใช้ในการคำนวณ หรือฟังก์ชัน สำหรับจัดการกับไฟล์ เป็นต้น
3. ส่วนการประกาศ (Declarations) เป็นส่วนที่ใช้กำหนดประเภทของข้อมูลแบบต่างๆเช่นการประกาศตัวแปร (Variable) การประกาศค่าคงที่ (Constants) หรือชนิดข้อมูล (Data Type) เป็นต้น
4. ส่วนของคำสั่ง (Executable Statements) เป็นส่วนของประโยคคำสั่ง ที่มีผลให้มีการปฏิบัติการอย่างใดอย่างหนึ่ง ส่วนของคำสั่ง นี้มักเป็นส่วนที่อยู่ในขอบเขตของเครื่องหมาย {...}



รูปที่ 3.16 โครงสร้างของภาษาซี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงชื่อเอกสารทุกครั้งที่มีการนำไปใช้

- ตัวแปรชนิดต่างๆ และค่าคงที่  
ภาษาซี สำหรับ ARM7 มีโครงสร้างเหมือนกับภาษาซีทั่วไปทำให้การประกาศตัวแปรและค่าคงที่จึงคล้ายกัน ตัวแปรและค่าคงที่ ที่ประกาศไว้จะถูกจองพื้นที่และเก็บค่าไว้ในหน่วยความจำ (RAM)

- ชนิดของตัวแปรในภาษาซี

ชนิดของตัวแปร ขนาด ช่วงของข้อมูล การกำหนดชนิดของตัวแปร

1. Character 8 บิต -128 ถึง +127 char
2. Unsigned character 8 บิต 0 ถึง 255 unsigned char
3. Integer 16 บิต -32768 ถึง +32767 int
4. Unsigned integer 16 บิต 0 ถึง 65535 unsigned int
5. Long integer 32 บิต -2147483648 ถึง 2147483647 long int
6. Unsigned Long 32 บิต 0 ถึง 4,294,967,295 unsigned long int
7. Floating point 32 บิต  $3.4 \times 10^{-38}$  ถึง  $3.4 \times 10^{38}$  float
  - Character ตัวแปรชนิดตัวอักษรหรือจำนวนเต็ม 8 บิต ใช้เก็บข้อมูลที่เป็นตัวอักษรหรือจำนวนเต็มที่มีค่าอยู่ในช่วง -128 ถึง +127
  - Unsigned character ตัวแปรชนิดตัวอักษรหรือจำนวนเต็ม 8 บิต แบบไม่คิดเครื่องหมายใช้เก็บข้อมูลที่เป็นตัวอักษรหรือจำนวนเต็มบวกที่มีค่าอยู่ในช่วง 0 ถึง 255
  - Integer ตัวแปรชนิดจำนวนเต็มขนาด 16 บิต ใช้เก็บข้อมูลที่เป็นจำนวนเต็มที่มีค่าอยู่ในช่วง -32768 ถึง +32767
  - Unsigned integer ตัวแปรชนิดจำนวนเต็มขนาด 16 บิต แบบไม่คิดเครื่องหมายหรือจำนวนเต็มบวก ใช้เก็บข้อมูลที่เป็นจำนวนเต็มที่มีค่าอยู่ในช่วง 0 ถึง 65535
  - Long integer ตัวแปรชนิดจำนวนเต็มขนาด 32 บิต ใช้เก็บข้อมูลที่เป็นจำนวนเต็มที่มีค่าอยู่ในช่วง -2147483648 ถึง +2147483647
  - Unsigned Long ตัวแปรชนิดจำนวนเต็มขนาด 32 บิต แบบไม่คิดเครื่องหมายหรือจำนวนเต็มบวก ใช้เก็บข้อมูลที่เป็นจำนวนเต็มที่มีค่าอยู่ในช่วง 0 ถึง 4,294,967,295
  - Floating point ตัวแปรชนิดทศนิยม ใช้เก็บข้อมูลเลขทศนิยมที่มีค่าอยู่ในช่วง  $3.4 \times 10^{-38}$  ถึง  $3.4 \times 10^{38}$

ตัวอย่างการประกาศตัวแบบ char และ int ARM7 จะไปจองพื้นที่ในหน่วยความจำได้ตามชนิดของตัวแปร คือ char = 1 Byte และ int = 2 Byte

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

<b>char i;</b>	<b>i</b>	<b>0x10000H</b>
<b>int j;</b>	<b>j</b>	<b>0x10001H</b>
	<b>j</b>	<b>0x10002H</b>
<b>i = 100;</b>	<b>64h</b>	<b>0x10000H</b>
<b>j = 10,000;</b>	<b>10h</b>	<b>0x10001H</b>
	<b>27h</b>	<b>0x10002H</b>
<b>Remark : 100 = 64h, 10,000 = 2710h</b>		

### 3.4.1.1 การประกาศตัวแปร

ในการเขียนโปรแกรมภาษาซีสำหรับ ARM7 การประกาศตัวแปรเพื่อใช้เก็บค่าข้อมูลมีรูปดังนี้

Syntax : ชนิดของตัวแปร ชื่อตัวแปร;

ตัวอย่าง :

**char i;** สร้างตัวแปรชื่อ i สำหรับเก็บค่าข้อมูล-128 ถึง 127

**int j;** สร้างตัวแปรชื่อ j สำหรับเก็บค่าข้อมูล -32,768 ถึง 32,767

**unsigned int count;** สร้างตัวแปรชื่อ count สำหรับเก็บค่าข้อมูล 0 ถึง 65,535

**long x;** สร้างตัวแปรชื่อ x สำหรับเก็บค่าข้อมูล -2,147,483,648 ถึง 2,147,483,647

**float y;** สร้างตัวแปรชื่อ y สำหรับเก็บค่าข้อมูลทศนิยม

### 3.4.1.2 การประกาศตัวแปรพร้อมกับค่าเริ่มต้น

Syntax : ชนิดของตัวแปร ชื่อตัวแปร = ค่าเริ่มต้น;

ตัวอย่าง:

```
unsigned int count = 10000;
float y = 99.9999;
```

### 3.4.1.3 การประกาศตัวแปรแบบค่าคงที่ (Constant)

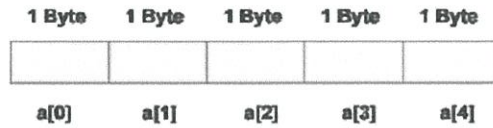
การประกาศตัวแปรแบบค่าคงที่ เป็นการประกาศตัวแปรพร้อมกับกำหนดค่าให้ตัวนั้นเลย ซึ่งค่าในตัวแปรจะไม่สามารถแก้ไขหรือเปลี่ยนแปลงได้ภายหลัง การประกาศตัวแปรค่าคงที่จะประกาศไว้ตรงส่วนหัวของโปรแกรม

```
#include <ADuC7024.H>
static const float pi =3.14;
static const int y = 5;
```

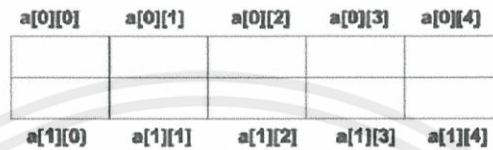
### 3.4.1.4 ตัวแปรแบบอาร์เรย์

ตัวแปรแบบอาร์เรย์ (Array) คือ ตัวแปรที่ใช้เก็บข้อมูลแบบชุด หรือกลุ่มข้อมูลที่เป็นชนิดเดียวกัน โดยมีโครงสร้างดังนี้

unsigned char a[5];



unsigned char a[2][5];



### 3.4.1.5 เครื่องหมายตัวดำเนินการ

เครื่องหมายและตัวดำเนินการภาษาซี ที่เรารู้จักและจำเป็นต้องใช้งานอยู่เป็นประจำ สามารถแบ่งออกเป็น 5 แบบตามตารางดังนี้ คือ

ตารางที่ 3.3 เครื่องหมายตัวดำเนินการทางคณิตศาสตร์

เครื่องหมาย	การดำเนินการ	ตัวอย่าง
+	บวก	$a = b + c$
-	ลบ	$a = b - c$
*	คูณ	$a = b * c$
/	หาร	$a = b / c$
%	หารเอาเศษ	$a = b \% c$

ตารางที่ 3.4 เครื่องหมายตัวดำเนินการทางด้านการเปรียบเทียบ

เครื่องหมาย	การดำเนินการ	ตัวอย่าง
>	มากกว่า	$a > b$ ( ถ้า a มากกว่า b ผลลัพธ์ = จริง )
<	น้อยกว่า	$a < b$ ( ถ้า a น้อยกว่า b ผลลัพธ์ = จริง )
>=	มากกว่าหรือเท่ากับ	$a >= b$ ( ถ้า a มากกว่าหรือเท่ากับ b ผลลัพธ์ = จริง )
<=	น้อยกว่าหรือเท่ากับ	$a <= b$ ( ถ้า a น้อยกว่า b หรือเท่ากับผลลัพธ์ = จริง )
==	เท่ากับ	$a == b$ ( ถ้า a เท่ากับ b ผลลัพธ์ = จริง )
&&	AND	$a \&\& b$ ( ผลลัพธ์เป็นจริงก็ต่อเมื่อ a และ b เป็นจริง )
	OR	$a \ \  b$ ( ผลลัพธ์จะเป็นจริงก็ต่อเมื่อ a หรือ b เป็นจริง )
!	NOT	$!a$ ( เมื่อ a เป็นจริง !a ผลลัพธ์จะเป็นเท็จ )

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3.5 เครื่องหมายตัวดำเนินการกระทำระดับบิต

เครื่องหมาย	การดำเนินการ	ตัวอย่าง
&	AND	$a = b \& 0xF8$ ; (1111 1000) ถ้า $b = 1010$ 1010 ; a เท่ากับ 1010 1000
	OR	$a = b   0xF8$ ; (1111 1000) ถ้า $b = 1010$ 1010 ; a เท่ากับ 1111 1010
^	XOR	$a = b \wedge 0xF8$ ; (1111 1000) ถ้า $b = 1010$ 1010 ; a เท่ากับ 0101 0010
~	Complement	$a = \sim b$ (1111 1000) ; a เท่ากับ 0000 0111

ตารางที่ 3.6 เครื่องหมายตัวดำเนินการเพิ่มและลดค่า

เครื่องหมาย	การดำเนินการ	ตัวอย่าง
++	เพิ่มค่า	$a = i++$ ; นำค่าตัว $i$ ไปเก็บที่ตัวแปร $a$ แล้วบวกค่า $i$ เพิ่มหนึ่งค่า $a = ++i$ ; บวกค่า $i$ เพิ่มขึ้นหนึ่งค่า แล้วนำค่า $i$ ไปเก็บที่ตัวแปร $a$
-	ลดค่า	$a = i--$ ; นำค่าตัว $i$ ไปเก็บที่ตัวแปร $a$ แล้วลบค่า $i$ ลงหนึ่งค่า $a = --i$ ; ลบค่า $i$ ลงหนึ่งค่า แล้วนำค่า $i$ ไปเก็บที่ตัวแปร $a$

ตารางที่ 3.7 เครื่องหมายตัวดำเนินการเลื่อนบิต (ซ้าย, ขวา)

เครื่องหมาย	การดำเนินการ	ตัวอย่าง
<<	เลื่อนบิตไปทางซ้าย	$a \ll 3$ เลื่อนบิตข้อมูลในตัวแปร $a$ ไปทางซ้าย 3 บิต $a$ (0000 0001) ผลลัพธ์ $a = 0000$ 1000
>>	เลื่อนบิตไปทางขวา	$a \gg b$ เลื่อนบิตข้อมูลในตัวแปร $a$ ไปทางซ้าย $b$ บิต $a$ (0011 0000), $b = 4$ ; ผลลัพธ์ $a = 0000$ 0011

ตารางที่ 3.8 ตัวกำหนดชนิดข้อมูลของภาษาซี

ตัวกำหนดชนิดข้อมูล	ความหมายของตัวกำหนดชนิดข้อมูล
%c	แสดงข้อมูลในรูปตัวอักษร
%d	แสดงข้อมูลในรูปตัวเลขจำนวนเต็ม
%e	แสดงข้อมูลในรูปเอกซ์โพเนนเชียล
%f	แสดงข้อมูลในรูปตัวเลขทศนิยม
%o	แสดงข้อมูลในรูปตัวเลขฐาน 8
%x	แสดงข้อมูลในรูปตัวเลขฐาน 16
%u	แสดงข้อมูลในรูปของจำนวนเต็มบวก
%s	แสดงข้อมูลในรูปของข้อความ (String)
%p	แสดงข้อมูลในรูปของที่อยู่ของตัวแปร(Address)
%%	แสดงเครื่องหมาย %

การกำหนดชนิดของตัวแปร ควรกำหนดให้สอดคล้องกับการใช้งาน เช่น ถ้ามีการคำนวณเป็นจุดทศนิยมควรกำหนดชนิดของตัวแปรเป็นแบบ float หรือการกำหนดตัวแปรเพื่อใช้ในการวนรอบที่ไม่เกิน 127 รอบควรกำหนดตัวแปรเป็น char เป็นต้น เพื่อลดขนาดของโปรแกรมให้เล็กลงและทำให้การประมวลผลเร็วขึ้น

### 3.1.4.6 รูปแบบการเขียนคำสั่ง

คำสั่งต้องเขียนด้วยตัวพิมพ์เล็ก ในภาษาซีถือว่าตัวพิมพ์เล็กและตัวพิมพ์ใหญ่ต่างกัน for () , while , if..else.. , unsigned int count; เมื่อจบคำสั่ง จะมีต้องลงท้ายด้วยเครื่องหมาย “ ; ” และสามารถเขียนหลายคำสั่ง ในบรรทัดเดียวกันได้

```
Unsigned char I,j;
I=0;
J=0; j++
```

- หลังฟังก์ชัน ไม่ต้องลงท้ายด้วยเครื่องหมาย “ ; ”

```
void delay (unsigned long count)
{
While (count>0)
{
Count--;
}
}
```

ในหนึ่งโปรแกรมจะมีฟังก์ชันก็ได้แต่จะต้องมีฟังก์ชันที่ชื่อ main() เสมอ

```
void main(void)
{
.
}
```

- การตั้ง ชื่อตัวแปรใดๆ ในโปรแกรม : เป็นการกำหนดชื่อต่างๆ เช่น ชื่อค่าคงที่ ชื่อโปรแกรม ชื่อฟังก์ชัน ชื่อของประเภทข้อมูล ฯลฯ เพื่อนำไปใช้ในโปรแกรม ซึ่งเป็นชื่อที่ผู้เขียนโปรแกรมกำหนดขึ้นมา โดยการกำหนดชื่อมีกฎเกณฑ์ดังต่อไปนี้

- ประกอบด้วยตัวอักษรตั้งแต่ A ถึง Z หรือตัวเลขตั้งแต่ 0 ถึง 9 ที่ไม่มีสัญลักษณ์พิเศษอื่นๆอยู่
- เริ่มต้นด้วยตัวอักษรภาษาอังกฤษ (ห้ามเป็นตัวเลข) และตัวถัดไปอาจเป็นตัวเลข ตัวอักษร หรือ

เครื่องหมาย Underscore ( \_ ) ก็ได้

- ตัวอักษรตัวใหญ่กับตัวเล็กจะถือว่าเป็นคนละตัวกัน

i , j , count, Count, count1, count2, x\_y, \_irq

29x,"xy", +x,y- , 12345

การแทรกคำบรรยาย (Comment) : การเขียนคำบรรยายแทรกไว้ภายในโปรแกรม จะช่วยเพิ่มความเข้าใจของตัวเรา หรือเพื่อให้คนอื่นมาอ่านโปรแกรมได้เข้าใจสิ่งที่เราเขียนด้วย คำบรรยายแทรก โดยใช้เครื่องหมาย

```
/* ข้อความที่ต้องการ */
// ข้อความแทรกทั้งบรรทัด
/* function delay time
   Calling by main function */
void delay (unsigned long count)
{
While (count>0)
Count==; //Counter
}
```

### 3.4.1.7 คำสั่งเงื่อนไขและการทำซ้ำ

#### 3.4.1.7.1 คำสั่งเงื่อนไข (Condition)

การเขียนโปรแกรมภาษาซี ในบางกรณีจำเป็นต้องให้ ARM7 ตัดสินใจดำเนินการอย่างใดหนึ่ง โดยการกำหนดเป็นเงื่อนไข (Condition) เช่น ถ้าเงื่อนไขเป็น “จริง” ให้ ARM7 ดำเนินการตามชุดคำสั่ง A แต่ถ้าเป็นเงื่อนไขเป็น “เท็จ” ให้ ARM7 ดำเนินการตามชุดคำสั่ง B เป็นต้น คำสั่ง กำหนดเงื่อนไขได้แก่ คำสั่ง if, if-else, if-else-if และ Switch-Case

- If

```
{
คำสั่ง
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ตัวอย่าง คำสั่ง if-else

```

if(เงื่อนไข)
{
    คำสั่ง
}
else
{
    คำสั่ง
}
else
{

```

- ตัวอย่าง คำสั่ง if-else-if

```

if(เงื่อนไข)
{
    คำสั่ง
}
else if(เงื่อนไข)
{
    คำสั่ง
}
else
{

```

- ตัวอย่าง คำสั่ง Switch - Case

```

switch(เงื่อนไข)
{
    case 1 :
        คำสั่ง;
        Break;
    case 2 :
        คำสั่ง;
        Break;
    case n :

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งยังมีให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

คำสั่ง;
Break;
}

```

### 3.4.1.7.2 การทำซ้ำ (Loop)

การทำซ้ำหรือการทำ Loop เป็นการสั่ง ให้ ARM7 ทำชุดคำสั่ง เดิมซ้ำๆจนครบตามจำนวนรอบที่กำหนดหรือ ตามเงื่อนไขที่กำหนด ซึ่งคำสั่งการทำซ้ำ ได้แก่ for, while และ do-while

#### - คำสั่ง for

คำสั่ง for เป็นการทำซ้ำที่ผู้เขียนโปรแกรมสามารถกำหนดจำนวนครั้งให้กระทำคำสั่งได้โดยการกำหนดเริ่มต้นให้แก่ตัวแปร Counter และจำนวนรอบที่ต้องการทำซ้ำ เมื่อกระทำซ้ำคำสั่ง ใน {...} จนครบตามจำนวนรอบก็จะไปทำสิ่งหลัง } (ออกจาก Loop)

```

Unsigned int i;
for (i=0;i<100;i++)
{
    คำสั่ง
}

```

#### - คำสั่ง while

คำสั่ง while เป็นการทำซ้ำโดยการตรวจสอบเงื่อนไขก่อน ถ้าเงื่อนไขเป็นจริงจะกระทำซ้ำคำสั่งที่อยู่ใน { .. } และหากเงื่อนไขเป็นเท็จจะกระทำคำสั่ง หลัง }.. (หลุดออกจาก Loop)

```

Unsigned int i;
i = 0;
while(i<100)
{
    คำสั่ง
    i++;
}
คำสั่ง

```

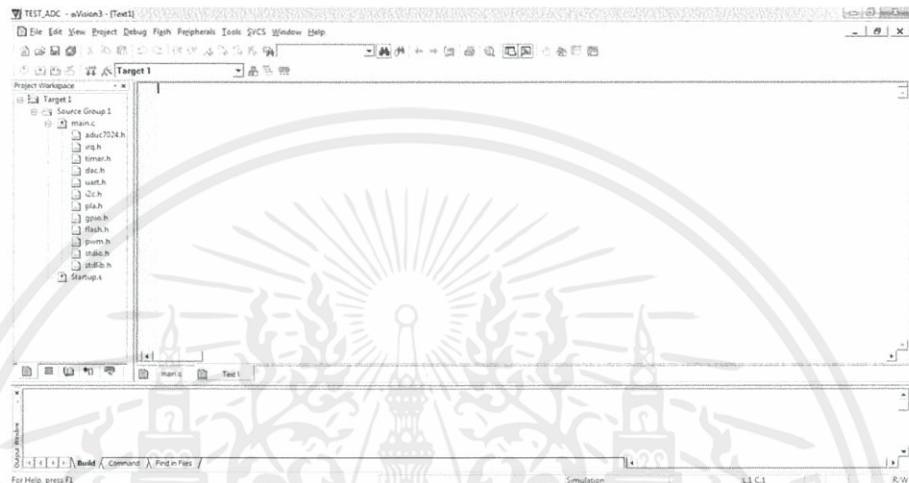
### 3.4.1.8 Standard Function และ User define Function

Standard Function เป็นฟังก์ชันมาตรฐานที่ไมโครคอนโทรลเลอร์ ARM7 ได้เตรียมไว้ให้ ผู้ใช้งานสามารถเรียกใช้งานได้ ซึ่งฟังก์ชันมาตรฐานของไมโครคอนโทรลเลอร์ ARM7 (ADuC7024) เช่น <stdio.h>, printf(),scanf(), clear(), getchar() เป็นต้น ส่วน User define Function เป็นฟังก์ชันที่นอกเหนือจาก Standard Function ที่ไมโครคอนโทรลเลอร์ ARM7 ให้มาโดยผู้เขียนโปรแกรมสร้างขึ้นมาใช้งานเองเพื่อตอบสนองความต้องการในการเขียนโปรแกรม

### 3.4.2 โปรแกรมคอมไพเลอร์ Keil uVision3

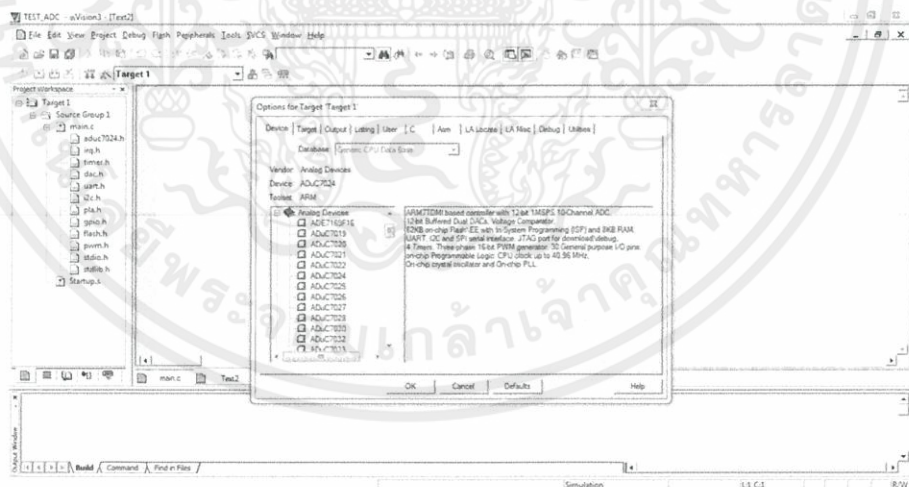
ในการเขียนโปรแกรมให้ไมโครคอนโทรลเลอร์สำหรับ ARM7024 นั้น จะใช้โปรแกรมที่มีชื่อว่า Keil uVision 3 ในการเขียนโปรแกรมเนื่องจากใช้ภาษาซีซึ่งง่ายต่อความเข้าใจ และ Keil uVision 3 นั้นยังสามารถสร้างไฟล์ Hex ที่ใช้สำหรับลงโปรแกรมบนไมโครคอนโทรลเลอร์ได้อีกด้วย สำหรับการสร้างโปรแกรมบนโปรแกรม Keil uVision 3 สรุปพอสังเขปได้ดังนี้

#### 1. เปิดโปรแกรม Keil uVision 3



รูปที่ 3.17 การเปิดโปรแกรม uVision 3

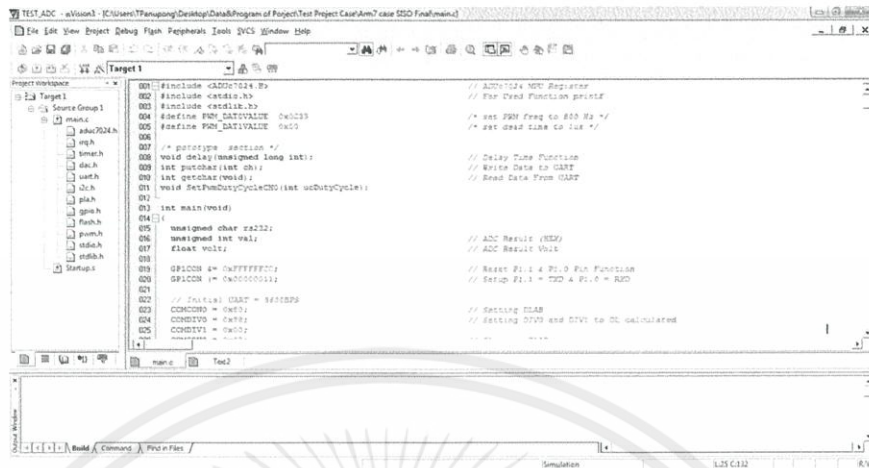
#### 2. สร้างไฟล์โปรเจกต์และเลือก ARM7 เบอร์ ADuC7024



รูปที่ 3.18 การสร้างไฟล์โปรเจกต์และเลือก ARM7 เบอร์ ADuC7024

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3. เขียน Source code เพิ่มเข้าไปใน Source file ของโปรเจค



รูปที่ 3.19 การเขียน Source code เพิ่มเข้าไปใน Source file

### 4. สร้างโปรเจคและ HEX ไฟล์สำหรับ PROM

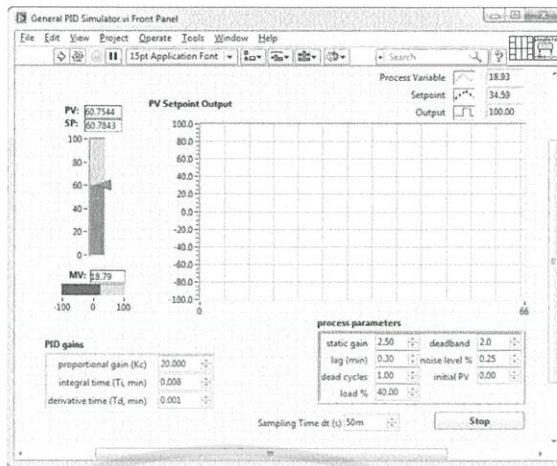


รูปที่ 3.20 การสร้างโปรเจคและ HEX ไฟล์สำหรับ PROM

#### 3.4.3 การใช้งานโปรแกรม LabVIEW

##### 3.4.3.1 พาเนล

พาเนลเป็นส่วนที่ใช้ในการแสดงผลค่าต่างๆ รวมทั้งการแสดงผลในเชิงตัวเลข ตัวอักษร กราฟ และอื่นๆ และใช้ในการควบคุมโดยการป้อนค่าที่ผู้ใช้ต้องการ ไม่ว่าจะเป็นตัวเลขหรือเป็นอักษรก็ตาม พุดง่ายๆก็คือพาเนลเป็นได้ทั้ง indicator และ control หรือเป็น เอาท์พุตและอินพุตนั่นเอง ส่วนประกอบของพาเนลมีอยู่ 3 ส่วน คือ พื้นที่สำหรับเขียนโปรแกรม เมนูบาร์ และทูลบาร์ การวางเอกสารนี้เป็นอุปกรณ์ทำได้โดยการใช้เมาส์คลิกค้างไว้ที่อุปกรณ์ที่ต้องการจากนั้นลากไปวางในหน้าต่างพาเนลเป็น ริก้า ไม่ว่าจะมิได้นั้นเสร็จเรียบร้อย หน้าต่างในส่วนของพาเนลแสดงดังรูปที่ 3.21

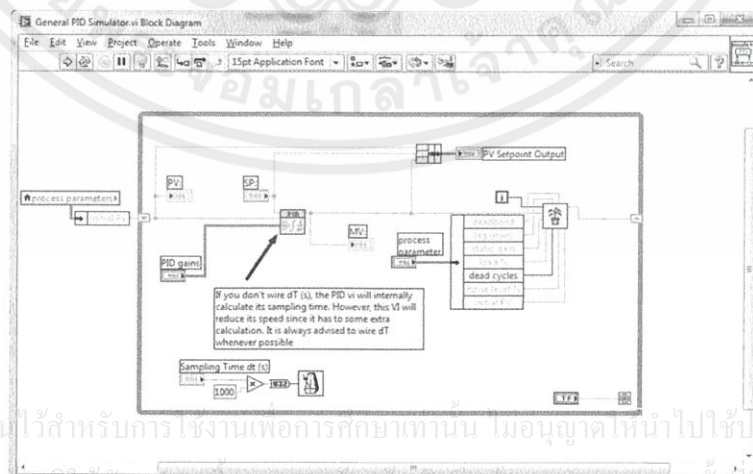


รูปที่ 3.21 แสดงหน้าต่างพาดวง

เมื่อผู้ใช้ทำการลากตัวอุปกรณ์ลงไปวางที่พาดวงแล้ว ผู้ใช้สามารถสั่งให้ตัวอุปกรณ์ตัวนั้นเป็นตัวควบคุมหรือตัวแสดงผลก็ได้ วิธีการคือ คลิกขวาที่ตัวอุปกรณ์ที่ผู้ใช้งานต้องการจะเปลี่ยน จากนั้นจะเห็นคำสั่ง Change to Indicator หรือ Change to Control ซึ่งแล้วแต่ว่า ณ ตอนนั้นอุปกรณ์ที่ผู้ใช้ลากไปวางเป็นอะไร ถ้าเป็นประเภท Control เมื่อคลิกขวาที่ตัวอุปกรณ์ก็จะเห็นเป็น Change to Indicator แต่ถ้าอุปกรณ์ ณ ตอนนั้นเป็น Indicator เมื่อคลิกขวาที่ตัวอุปกรณ์ก็จะเห็นเป็น Change to Control ที่ได้กล่าวไปนั้นเป็นส่วนหนึ่งของพาดวงเท่านั้น

### 3.4.3.2 บล็อกไดอะแกรม

บล็อกไดอะแกรมเป็นส่วนที่ใช้สำหรับการเขียนโปรแกรมในโปรแกรม LabVIEW โดยการเขียนโปรแกรมด้วยภาษากกราฟิก (G-Languages) ในเบื้องต้นนี้อาจมองบล็อกไดอะแกรมว่าเป็น Data Flow Chart และตัวโปรแกรมหรือ code ของ LabVIEW ซึ่งความแตกต่างระหว่าง code ทั่วไปกับบล็อกไดอะแกรมใน LabVIEW ก็คือ บล็อกไดอะแกรมนั้นพร้อมที่จะทำการประมวลผลตลอดเวลา แต่สำหรับบางภาษาจะทำงานเป็นแบบซ้ายไปขวาและบนลงล่าง ซึ่งหน้าต่างในส่วนของบล็อกไดอะแกรมแสดงดังรูปที่ 3.22



รูปที่ 3.22 แสดงหน้าต่างบล็อกไดอะแกรม

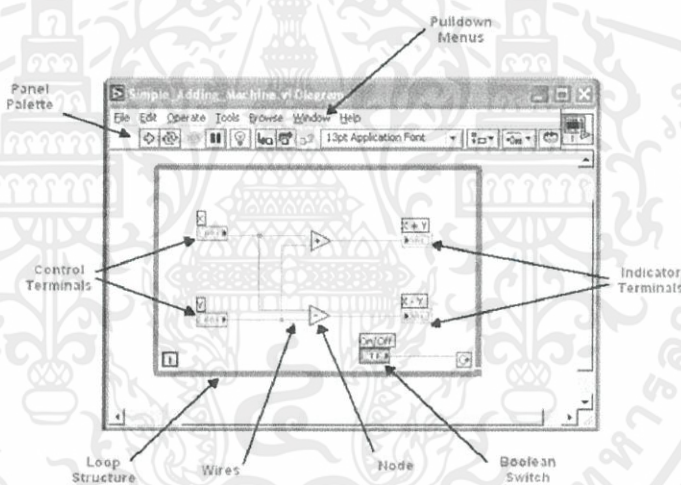
ถ้าหากพิจารณาองค์ประกอบในบล็อกไดอะแกรม ผู้ใช้จะพบว่าในส่วนบล็อกไดอะแกรม จะมี ส่วนประกอบที่สำคัญ 3 ส่วน ทั้งสามส่วนจะมีหน้าที่หลัก คือ ทำหน้าที่ควบคุมการส่งผ่านข้อมูลหรือ ผู้ใช้อาจเรียกอีกอย่างว่าการไหลของข้อมูล (Data Flow) นอกจากนั้นยังทำหน้าที่กำหนดวิธีการ ประมวลผลของข้อมูลอีกด้วย ซึ่งจะขออธิบายทั้ง 3 ส่วน ดังต่อไปนี้

- Terminal

การสร้างตัวควบคุมหรือตัวที่ใช้แสดงผลบนพาเนลในหน้าต่างของบล็อกไดอะแกรมจะปรากฏ Terminal ขึ้น Terminal เป็นส่วนที่ใช้สำหรับการรับส่งข้อมูลระหว่าง Terminal อื่นๆ

- Node

Node คือ บล็อกที่มีเพื่อประมวลผล เมื่อมีข้อมูลเข้าสู่ node สิ่งที่เกิดขึ้นภายใน node ก็ จะขึ้นอยู่กับว่าจะกำหนดให้ข้อมูลที่ส่งเข้าไบนั้น จะมีการประมวลผลอย่างไร ซึ่งอาจจะเป็นการบวก ลบ คูณ หาร ทหาราก ยกกำลัง หรือเป็นประเภทการเปรียบเทียบข้อมูล ว่ามากหรือน้อยกว่า หรืออื่นๆ ซึ่ง จะเป็นการประมวลผลทางคณิตศาสตร์ทั่วไป รูปที่ 3.23 แสดงถึงลักษณะของ Node และ Terminal และส่วนอื่นๆ ที่บรรจุอยู่ภายในบล็อกไดอะแกรมของ LabVIEW



รูปที่ 3.23 แสดงส่วนประกอบต่างๆของบล็อกไดอะแกรม

นอกเหนือจากการประมวลผลทางคณิตศาสตร์แล้ว ผู้ใช้ยังมี node ประเภท Structure หรือ Control Flow อีกด้วย (ในภาษาตัวอักษร Structure Command จะเป็น คำสั่งจำพวก IF...THEN, FOR..., WHILE... เป็นต้น)

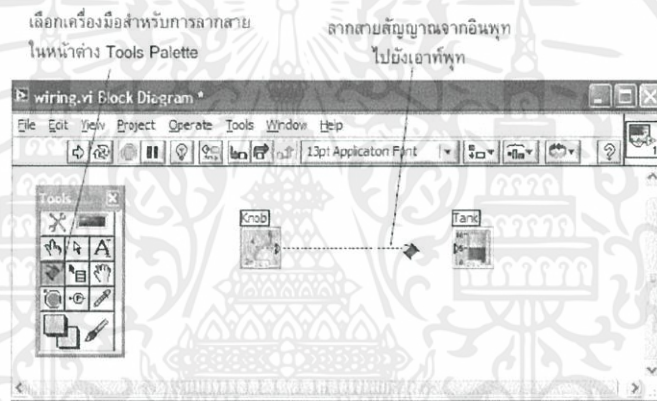
- Wires

การเชื่อมต่อโดยใช้สายส่งผ่านข้อมูลจาก terminal หรือ node ต่างๆที่มีในบล็อกไดอะแกรม นี้เข้าด้วยกัน โดย wires นี้จะเป็นการกำหนดเส้นทางของข้อมูลว่าเมื่อออกจาก terminal หนึ่งแล้ว จะกำหนดการไหลไปที่ node ไตบ้าง มีลำดับเป็นอย่างไร และสุดท้ายจะให้แสดงผลที่ terminal ไต นั้นเอง

เนื่องจากข้อมูลนั้นมีหลายแบบไม่ว่าจะเป็นเลขทศนิยม, เลขจำนวนจริง, ตัวอักษร หรือค่า จริง-เท็จ (Boolean) ดังนั้นเพื่อแสดงถึงความแตกต่างของข้อมูลแต่ละแบบ LabVIEW จึงได้

กำหนดให้ลักษณะของ wires สำหรับข้อมูลแต่ละแบบมีลักษณะของเส้นและสีที่แตกต่างกัน โดยการ  
ใช้ Wiring Tool (รูปหลอดด้าย) ใน Tool Palette โดยที่

- สีของ Control หรือ Indicator
  1. สีน้ำเงิน แทนตัวเลขที่เป็นจำนวนเต็ม
  2. สีส้ม แทนตัวเลขที่มีค่าทศนิยม
  3. สีเขียว แทน Logic หรือ Boolean
  4. สีชมพู แทนค่าของ String
- รูปแบบของWiring
  1. เส้นบางแทนค่าจำนวนเดียว
  2. เส้นหนาแทนค่าของอาร์เรย์ ขนาด 1 มิติ
  3. เส้นคู่แทนค่าของอาร์เรย์ ขนาด 2 มิติ



รูปที่ 3.24 แสดงการเชื่อมต่อสัญญาณ

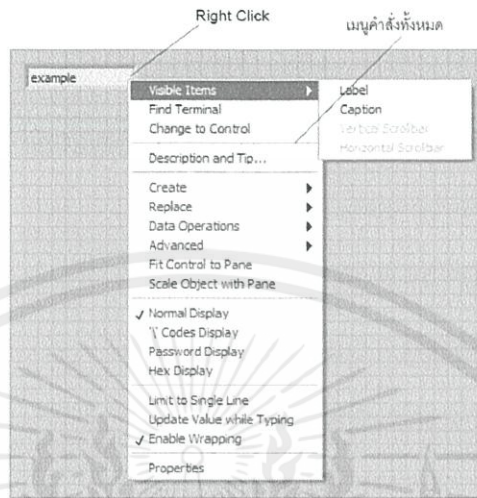
### 3.4.3.3 ไอคอนและจุดเชื่อมต่อ

ไอคอน (Icon) และจุดเชื่อมต่อ (Connector) ในโปรแกรม LabVIEW ก็คืออุปกรณ์ตัวเดียวกันเพียงแต่อุปกรณ์นั้นสามารถปรับเปลี่ยนคุณสมบัติให้เหมาะสมกับการใช้งานซึ่งโปรแกรมที่เห็น สัญลักษณ์ของอุปกรณ์โดยทั่วไป จะเรียกสัญลักษณ์เหล่านั้นว่า ไอคอน (Icon) เมื่อต้องการเชื่อมต่อ สายสัญญาณของไอคอน จำเป็นต้องรู้ว่าจุดเชื่อมต่อ (Connector) ของไอคอนนั้นอยู่ที่ตำแหน่งใดของไอคอนและมีคุณสมบัติอย่างไร จึงจำเป็นต้องเปลี่ยนคุณสมบัติของไอคอนให้เป็นคอนเนคเตอร์ โดยคลิกเมาส์ทางขวาที่รูปไอคอนแล้วเลือก Visible Item» Terminal ไอคอนก็จะเปลี่ยนไปเป็นเทอร์มินอลที่แสดงจุดเชื่อมต่ออย่างชัดเจน

### 3.4.3.4 เมนูคำสั่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับ ● เมนูคำสั่งแบบ Shortcut เท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
การเรียกใช้งานชุดคำสั่งในโปรแกรม LabVIEW มีอยู่สองวิธีคือ เรียกในเมนูบาร์ (Menu bar) และเรียกจากเมนูคำสั่งแบบย่อ โดยวิธีการเรียกใช้งานชุดคำสั่งแบบย่อทำได้โดยการลากเมาส์ไปวาง บริเวณ object บนหน้าต่างพร้อมท์พาเนลหรือหน้าต่างบล็อกไดอะแกรมแล้วคลิกเมาส์ทาง

ด้านขวาซึ่งจะปรากฏชุดคำสั่งย่อขึ้นมาภายในคำสั่งก็จะมีชุดคำสั่งย่อ การเรียกใช้งานนั้นทำได้โดยการลากเมาส์ไปชี้ ตรงคำสั่งที่ต้องการซึ่งจะเกิดแถบสีน้ำเงินที่บริเวณที่เมาส์ชี้อยู่ถ้าต้องการใช้คำสั่งให้คลิกเมาส์บริเวณ คำสั่งที่ต้องการ ดังรูปที่ 3.25



รูปที่ 3.25 การใช้โครงสร้างของคำสั่งแบบ Shortcut

#### • เมนูคำสั่งแบบ Pull Down

เมนูคำสั่งแบบ Pull Down ในหน้าต่างพาเนลและไดอะแกรมจะมีชุดคำสั่งที่เมนูบาร์อยู่ 8 คำสั่ง คือ File, Edit, View, Project, Operate, Tools, Window และ Help

| File Edit View Project Operate Tools Window Help

รูปที่ 3.26 แสดงหน้าต่างคำสั่งแบบ Pull Down

คำสั่ง File ในคำสั่ง File จะประกอบไปด้วยคำสั่งย่อยสำหรับการจัดการเกี่ยวกับการสร้าง VI การเปิดไฟล์ VI การปิด VI สำหรับหน้าต่างที่ผู้ใช้เปิดอยู่ การปิด VI ทุกหน้าต่างที่กำลังเปิดอยู่ การบันทึก การสร้าง Project การเปิดไฟล์ Project การตั้งค่าหน้าต่างกระดาษ การปรับหน้าต่างพาเนลหรือไดอะแกรม คุณสมบัติของ VI ไฟล์ที่เปิดล่าสุดและสุดท้ายออกจากโปรแกรม อีกทั้งยังมีคำสั่งลัดที่สั่งเปิดจากคีย์บอร์ด เช่น Ctrl+N สำหรับสร้างหน้าต่าง VI ขึ้นมาใหม่ Ctrl+O เพื่อเปิดไฟล์ VI Ctrl+S เพื่อบันทึก VI Ctrl+P เพื่อปรับหน้าต่างพาเนลหรือไดอะแกรมของ VI Ctrl+Q เพื่อปิด VI เป็นต้น

คำสั่ง Edit ในคำสั่ง Edit จะประกอบไปด้วยคำสั่งย่อยสำหรับการจัดการเกี่ยวกับการยกเลิก (Ctrl+Z) การทำซ้ำ (Ctrl+Shift+Z) การตัด (Ctrl+X) การคัดลอก (Ctrl+C) การวาง (Ctrl+V) การเลือกวัตถุทั้งหมด (Ctrl+A) การทำให้ค่าที่ผู้ใช้ต้องการนั้นเป็นค่าเริ่มต้น สร้าง SubVI การแทรกรูปภาพ การเอาสายที่ Broken ออก (Ctrl+B) คำสั่งย่อยในคำสั่ง Edit ที่ถูกใช้บ่อยครั้งที่สุดคือคำสั่ง Undo Data Chang และ Redo ซึ่งคำสั่งนี้จะช่วยผู้ใช้ในเรื่องของการแก้ไขโปรแกรม หากการแก้ไขผิดพลาดหรือไม่ต้องการ ผู้ใช้สามารถกลับไปยังจุดที่ผู้ใช้เริ่มต้นแก้ไขได้โดยใช้คำสั่งนี้

คำสั่ง View ในคำสั่ง View จะประกอบไปด้วยชุดแสดงผลของโปรแกรม เช่น Tools Palette, Controls palette และ Function Palette และยังมีส่วนแสดงผลของค่าความผิดพลาด

คำสั่ง Project ในคำสั่ง Project ประกอบไปด้วยชุดคำสั่งพื้นฐานของโปรแกรม ใช้สำหรับจัดการกับโปรแกรมที่เปิดพร้อมกันหลายๆ โปรแกรม ให้เปิด ปิด บันทึก และสร้างโปรแกรม

คำสั่ง Operate ใช้สำหรับสั่งให้โปรแกรมทำงาน(Ctrl+R) หรือหยุดทำงาน(Ctrl+.) เปลี่ยนหรือเซ็หค่ามาตรฐานของ VI และสามารถเปลี่ยนโหมดการทำงานจากโหมดรันไปเป็นโหมดแก้ไขได้ (Ctrl+M)

คำสั่ง Tools สำหรับเมนูนี้ใช้เกี่ยวกับการติดต่อระหว่างโปรแกรมกับอุปกรณ์ภายนอก เปรียบเทียบ VI เช่น Instrument Real-Time Module FPGA Module DSC Module เป็นต้น การสร้างแอปพลิเคชันต่างๆ และติดต่อกับเว็บไซต์ของ NI ด้วย

คำสั่ง Window ใช้สำหรับเลือกการเปิดหน้าต่างพาเนลและไดอะแกรมสลับกันหรือเปิดทั้ง 2 หน้าจอพร้อมกันก็ได้ เช่น Show Block Diagram(Ctrl+E) แสดงหน้าต่างไดอะแกรม Tile Left and Right(Ctrl+T) แสดงหน้าต่างทั้งพาเนลและไดอะแกรมซ้ายและขวา เป็นต้น

คำสั่ง Help ใช้ในการแสดงรายละเอียดของ Tool แต่ละตัวและใช้สำหรับสืบค้นข้อมูลต่างๆ ภายในโปรแกรม LabVIEW เช่น ตัวอย่างโปรแกรม เป็นต้น

### 3.4.3.5 Palette

Palette เป็นหน้าต่างย่อยที่มีส่วนประกอบของ object ที่ใช้สำหรับการเขียน VI การนำไปใช้งานนั้นทำได้ง่ายเพียงแค่คลิกเมาส์ทางด้านขวาบริเวณที่ว่างของหน้าต่างพาเนลหรือบล็อกไดอะแกรม ซึ่งจะปรากฏ Palette ขึ้นมาจากนั้นใช้เมาส์คลิกที่ object ที่ต้องการแล้วลากไปวางบนหน้าต่างพาเนลหรือหน้าต่างไดอะแกรม Palette มีองค์ประกอบหลักๆ อยู่ 3 ส่วนคือ Tools, Control และ Function

- Tools Palette

Tools Palette จะเป็นตัวบอกหรือกำหนดสถานะของเมาส์ว่าทำงานอยู่ในโหมดใด การเรียกใช้ Palette นี้ทำได้โดยการเลือกที่เมนู View แล้วเลือกคำสั่งย่อย Tools Palette แล้วจะปรากฏ Tools Palette ขึ้นมาดังรูปที่ 3.27 การใช้งานนั้นเพียงแค่เราใช้เมาส์ไปคลิกที่สัญลักษณ์ใน Tools Palette เคอร์เซอร์ของเมาส์ก็จะเปลี่ยนไปตามลักษณะที่เราเลือก

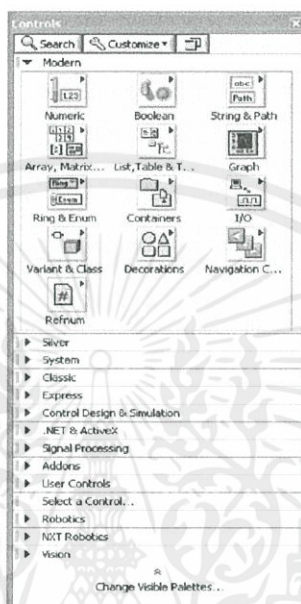


รูปที่ 3.27 แสดง Tools Palette

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- Control Palette

มีส่วนประกอบของอุปกรณ์ที่เป็นตัวควบคุมและตัวแสดงผลที่ใช้สำหรับสร้าง VI โดยภายใน Control Palette ก็จะมี Palette ย่อยๆ นอกจากนี้แล้วเรายังสามารถเปิดใช้งาน Control Palette ได้อีกทางหนึ่งคือเลือกที่เมนู View แล้วเลือก Control Palette ก็สามารถเลือกใช้ Control Palette ได้เช่นกัน ซึ่ง Control Palette แสดงดังรูปที่ 3.28

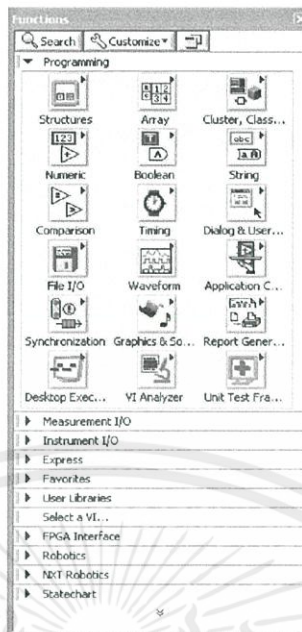


รูปที่ 3.28 แสดง Control Palette

- Function Palette

การใช้งานโดยทั่วไปจะมีความคล้ายกันกับ Control Palette ซึ่งภายในประกอบไปด้วยอุปกรณ์ในการสร้าง VI ตัวอย่างอุปกรณ์หลักๆ ใน Function Palette เช่น For Loops, While Loops และ Formula Nodes สามารถเรียกใช้งานได้ 2 วิธีเช่นเดียวกันกับการเรียกใช้งาน Control Palette คือ เลือกที่เมนู View ในหน้าต่างของบล็อกไดอะแกรม แล้วเลือก Function Palette หรือคลิกเมาส์ทางด้านขวามือบริเวณพื้นที่ว่างในหน้าต่างไดอะแกรม ก็จะปรากฏ Function Palette เช่นกัน ซึ่ง Function Palette แสดงดังรูปที่ 3.29

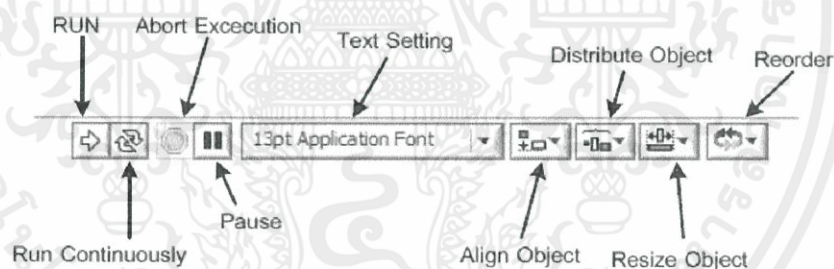
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.29 แสดง Function Palette

### 3.4.3.6 เครื่องมือ (Tool)

- เครื่องมือของหน้าต่างพาเนล



รูปที่ 3.30 แสดงทุลบาร์ในหน้าต่างพาเนล

RUN

ใช้สำหรับรันโปรแกรม

RUN Continuously

ใช้สำหรับรันโปรแกรมอย่างต่อเนื่อง

Abort Execution

หรือเรียกอีกอย่างหนึ่งว่าปุ่ม Stop ใช้สำหรับหยุดการรันโปรแกรม

Pause

ใช้สำหรับหยุดการรันโปรแกรมชั่วคราวเพื่อตรวจสอบการทำงานของโปรแกรม

เมื่อต้องการรันโปรแกรมต่อให้กดปุ่มนี้ซ้ำอีกครั้ง

โปรแกรมก็จะรันต่อไป

Text setting

ใช้สำหรับจัดการกับตัวอักษรทั้งหมดไม่ว่าจะเป็น ฟรอนต์ ขนาด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้รูปแบบตัวอักษรรวมไปถึงการกำหนดสีให้กับตัวอักษรด้วย

Align Object

ใช้จัด Object ต่างๆ ที่เราใช้เขียนโปรแกรมให้อยู่ในแนวเดียวกัน

ไม่ว่ากรณีใดก็ตามห้ามมิให้ดัดแปลงเนื้อหาหรือข้อมูลของเอกสารนี้โดยไม่ได้รับอนุญาต

ทั้งแนวตั้งและ แนวนอนเพื่อความเป็นระเบียบ

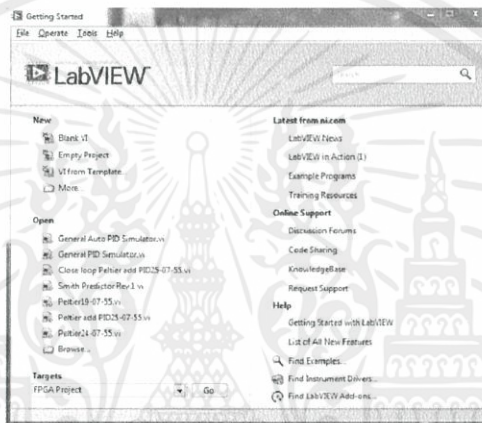
Distribute Object

ใช้สำหรับการกำหนดระยะห่างระหว่าง Objects อัตโนมัติ

Resize Object	ใช้สำหรับปรับขนาดของ Object
Reorder	ใช้สำหรับกำหนดลำดับก่อนหลังสำหรับการวางซ้อนทับกันของ Object
Context Help	ใช้แสดงคุณสมบัติหรือรายละเอียดของ Object แต่ละตัว

### 3.4.3.7 การเปิดและการบันทึกไฟล์

การเปิดใช้งานและการบันทึกโปรแกรม LabVIEW มีด้วยกันหลายวิธีด้วยกัน ซึ่งวิธีการเรียกใช้งานและการบันทึกแต่ละแบบก็มีรูปแบบที่แตกต่างกันออกไป ซึ่งผู้ใช้งานสามารถเรียกใช้งานได้ตามความเหมาะสม เพราะไม่ว่าจะเปิดด้วยวิธีใดก็สามารถใช้งานได้เช่นเดียวกัน



รูปที่ 3.31 แสดงการเริ่มต้นใช้งานโปรแกรม

- การเปิดไฟล์

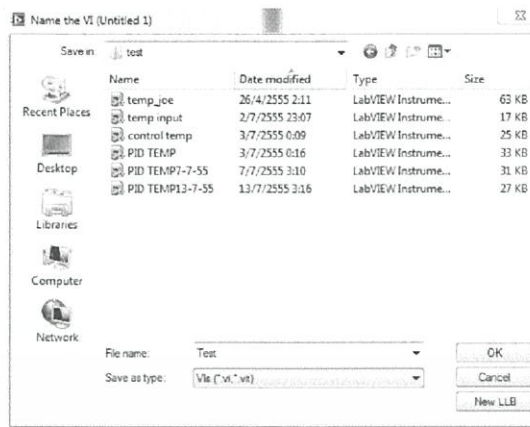
การเปิดไฟล์ใหม่ จากรูปที่ 3.31 คลิกเลือกไปที่ Blank VI ก็จะได้หน้าต่างพาเนลและหน้าต่างไดอะแกรมใหม่ขึ้นมา

การเรียกโปรแกรมที่สร้างไว้แล้วมาใช้งานทำได้โดยเปิดโปรแกรม LabVIEW แล้วเลือกที่เมนู File » Open จะปรากฏไดอะล็อกบ็อกที่บันทึกโปรแกรมไว้แล้วทำการเลือกไฟล์ที่ต้องการ ก็จะได้โปรแกรมที่บันทึกไว้มาใช้งานได้

- การบันทึกไฟล์

การบันทึกไฟล์ VI สามารถทำได้โดยเลือก Save, Save As, Save All, หรือ Save with Option จากเมนูไฟล์ หลังจากนั้นจะปรากฏหน้าต่างสำหรับบันทึกขึ้นมา แล้วทำการเลือก โฟลเดอร์ที่ต้องการบันทึกไฟล์ ตั้งชื่อไฟล์แล้วทำการบันทึก ไฟล์ที่บันทึกไว้จะมีนามสกุล \*.vi

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.32 แสดงไดอะล็อกบ็อกสำหรับเปิดใช้งานโปรแกรม LabVIEW

### 3.4.3.8 รูปแบบโครงสร้างการควบคุมการทำงานของโปรแกรม

ในหัวข้อนี้จะกล่าวถึงโครงสร้างการควบคุมการทำงานของโปรแกรม VIs ซึ่งโครงสร้างในโปรแกรม LabVIEW ที่จะกล่าวถึงคือ For Loop, While Loop, Shift Register and Feedback Node, Case Structure, Flat and Strack Sequency, Formula Node, MathScrip



รูปที่ 3.33 แสดงโครงสร้างการควบคุม

- For Loop

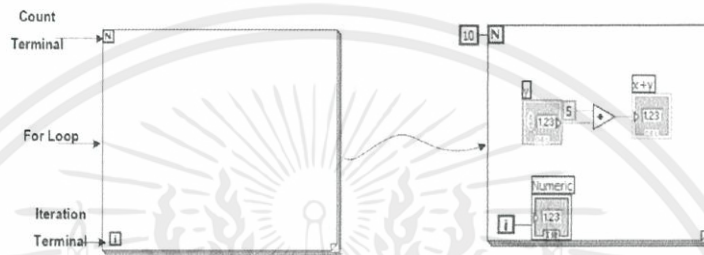
For Loop เป็นฟังก์ชันคำสั่งให้โปรแกรมทำงานซ้ำจนกว่าการกระทำซ้ำนั้นจะเท่ากับจำนวน N เมื่อ N คือจำนวนครั้งที่ต้องการให้ทำซ้ำ

ใน For Loop นั้นจะมีเทอร์มินอลอยู่ 2 จุดคือ count terminal และ iteration terminal สำหรับ count terminal นั้นมีหน้าที่เป็นตัวกำหนดว่าจะให้โปรแกรมทำงานซ้ำกี่รอบซึ่งสามารถจะกำหนดให้ปรับเปลี่ยนจำนวนรอบหรือเป็นค่าคงที่ก็ได้ ส่วน iteration terminal จะทำหน้าที่แสดงจำนวนรอบการทำงานว่าโปรแกรมทำงานในวงรอบไปกี่ครั้งแล้วตัวเลขที่แสดงของ iteration terminal จะมีค่าเท่ากับ N-1 การใช้งาน For Loop แสดงดังรูปที่ 3.46

การสร้าง For Loop ทำได้โดยการเลือกไปที่หน้าต่างบล็อกไดอะแกรมแล้วเลือก Function» Structure แล้วใช้เมาส์เลือกที่ For Loop นำเมาส์ไปคลิกที่หน้าต่างบล็อกไดอะแกรมแล้ว ลากให้มี

ขนาดที่ต้องการแล้วปล่อยเมาส์ก็จะได้โครงสร้างการทำงานของ For Loop นอกจากนี้แล้วยังสามารถทำการย่อขยายขนาดของ For Loop ได้โดยใช้เมาส์ไปชี้ตรงบริเวณเส้นขอบของ For Loop จะปรากฏจุดสี่เหลี่ยมเล็กๆ บริเวณเส้นขอบของ For Loop ใช้เมาส์ไปวางตรงบริเวณจุดดังกล่าวแล้วเมาส์จะเปลี่ยนเป็นรูปลูกศรจากนั้นคลิกเมาส์ค้างแล้วลากย่อหรือขยายได้ตามต้องการ

การทำงานของ For Loop นอกจากจะทำงานเพียงวงรอบเดียวๆ แล้วยังสามารถทำงานซ้อนวงรอบได้ โดยการทำงานนั้นจะทำงานวงรอบด้านในก่อนจนครบเงื่อนไขแล้วจึงมาทำวงรอบนอกหนึ่งรอบแล้วก็กลับไปทำงานในวงรอบในอีกครั้ง การทำงานจะทำจนกว่าวงรอบนอกครบตามเงื่อนไขโปรแกรมจึงจะหยุดทำงาน



รูปที่ 3.34 แสดงโครงสร้างของ For Loop

#### ● While Loop

While Loop เมื่อเปรียบเทียบกับ การเขียนโปรแกรมด้วยภาษาอื่นๆ ก็จะเป็นได้กับโครงสร้างของคำสั่ง Repeats Until นั่นเองซึ่งลักษณะของ While Loop แสดงดังรูปที่ 3.35 หลักการทำงานของ While Loop คือจะทำงานภายในวงรอบไปจนกว่าเงื่อนไขที่ Condition Terminal จะเป็นจริงจึงจะหยุดทำงาน

สำหรับ While Loop นั้นจะมีเทอร์มินอลอยู่ 2 เทอร์มินอลเช่นเดียวกับกับ For Loop คือ Iteration Terminal และ Condition Terminal การทำงานของ Iteration Terminal จะทำงานเหมือนกันกับ Iteration Terminal ของ For Loop คือทำหน้าที่แสดงจำนวนรอบที่โปรแกรมกระทำในวงรอบซึ่งแสดงผลออกมาในลักษณะของตัวเลขส่วน Condition Terminal เป็นการกำหนดเงื่อนไขให้โปรแกรมหยุดทำงานซึ่งเงื่อนไขในการหยุดทำงานคือจะหยุดทำงานเมื่อเงื่อนไขเป็นจริง (True) หรือจะให้หยุดทำงานเมื่อเงื่อนไขเป็นเท็จ (False) ก็สามารถที่จะกำหนดได้ นอกจากนี้แล้วเรายังสามารถกำหนดเงื่อนไขในการหยุดทำงานได้อีกหนึ่งวิธีคือการนำเอาค่าเอาท์พุทของ Iteration Terminal มาเปรียบเทียบกับค่าที่เราต้องการไม่ว่าจะเป็นค่าคงที่หรือปรับเปลี่ยนได้ โปรแกรมจะทำงานวนในวงรอบและนำค่าเอาท์พุทของ Iteration Terminal มาเปรียบเทียบกับค่าที่เรากำหนดไว้แล้วจะส่งเอาท์พุทที่ได้ไปยัง Condition Terminal เมื่อจำนวนรอบการทำงานมีค่าเท่ากับกับค่าที่เรากำหนดโปรแกรมจะหยุดทำงาน ในวงรอบนี้ทันทีที่กำหนดเงื่อนไขแบบนี้เปรียบได้กับการกำหนดค่า N ใน For Loop นั่นเอง

การสร้าง While Loop ทำได้โดยการเลือกไปที่หน้าต่างบล็อกไดอะแกรมแล้วเลือก Function >> Structure แล้วใช้เมาส์เลือกที่ While Loop นำเมาส์ไปคลิกที่หน้าต่างบล็อกไดอะแกรม แล้วลากให้มีขนาดที่ต้องการแล้วปล่อยเมาส์ก็จะได้โครงสร้างการทำงานของ While Loop นอกจากนี้ แล้วยังสามารถทำการย่อขยายขนาดของ While Loop ได้โดยใช้เมาส์ไปชี้ตรงบริเวณเส้นขอบของ While Loop จะปรากฏจุดสี่เหลี่ยมเล็กๆ บริเวณเส้นขอบของ While Loop

ใช้เมาส์ไปวางตรงบริเวณจุดตั้งกล่าวแล้วเมาส์จะเปลี่ยนเป็นรูปลูกศรจากนั้นคลิกเมาส์ค้างแล้วลากย่อหรือขยายได้ตามต้องการ



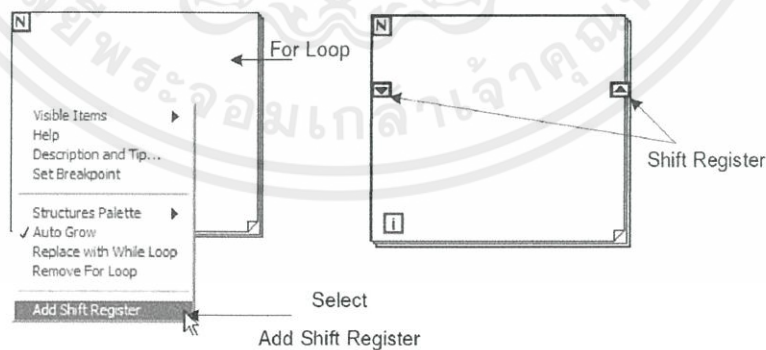
รูปที่ 3.35 แสดงโครงสร้างของ While Loop

- Shift Register and Feedback Node

- Shift Register

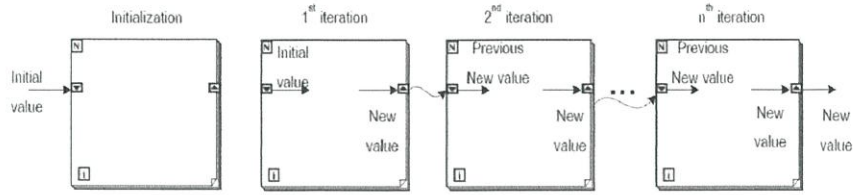
Shift Register ประกอบไปด้วยเทอร์มินอลสองตัวคือเทอร์มินอลทางด้านอินพุตและเทอร์มินอลทางด้านเอาต์พุต Shift Register สามารถใช้ทำงานร่วมกับ For Loop และ While Loop เท่านั้นการสร้าง Shift Register ทำได้โดยการใช้เมาส์คลิกขวาบริเวณเส้นขอบของ For Loop หรือ While Loop แล้วเลือก Add Shift Register ก็จะได้ Shift Register ที่อยู่บนเส้นขอบทางซ้ายและขวา ของ For Loop และ While Loop ดังแสดงให้เห็นในรูปที่ 3.36

การทำงานของ Shift Register คือจะส่งค่าข้อมูลจากเทอร์มินอลทางซ้ายไปยังเทอร์มินอลทางขวาค่าเอาต์พุตของวงรอบที่ผ่านมาจะมาเป็นค่าเริ่มต้นของวงรอบปัจจุบันดังแสดงในรูปที่ 3.37 ค่าเริ่มต้นของ Shift Register สามารถกำหนดได้แต่ถ้าเราไม่กำหนดค่าเริ่มต้นก็จะมีค่ามาตรฐานที่มาจากโปรแกรม Shift Register สามารถใช้ได้กับข้อมูลทุกชนิดไม่ว่าจะเป็นตัวเลขตัวหนังสือลอจิกทั้งที่เป็นค่าเดียวหรืออาร์เรย์



รูปที่ 3.36 แสดงโครงสร้างของ Shift Register

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

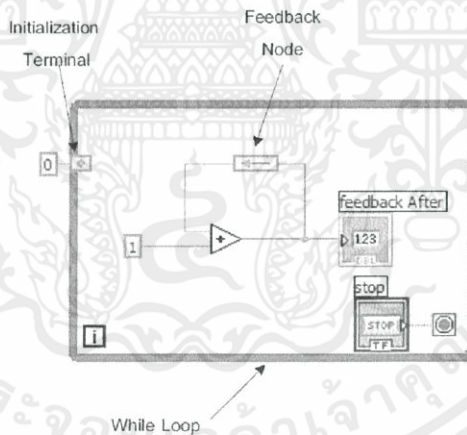


รูปที่ 3.37 แสดงการทำงานของ Shift Register

#### ■ Feedback Node

Feedback Node สามารถทำงานร่วมกับ For Loop และ While Loop ได้ เช่นเดียวกันกับ Shift Register การทำงานของ Feedback Node คือจะเก็บข้อมูลไว้ที่ Iteration เมื่อ โปรแกรมทำงานภายในวงรอบเสร็จแล้วจะส่งข้อมูลดังกล่าวไปที่ Iteration ของวงรอบต่อไป ทั้งนี้ Feedback Node ยังสามารถกำหนดค่าเริ่มต้นได้ด้วยและสามารถใช้ได้กับข้อมูลทุกชนิด ลักษณะการใช้งาน Feedback Node แสดงดังรูปที่ 3.38

การสร้าง Feedback Node ทำได้โดยการเลือกไปที่หน้าต่างบล็อกไดอะแกรมแล้วเลือก Function» Structure แล้วใช้เมาส์เลือกที่ Feedback Node (แสดงดังรูปที่ 3.38) นำเมาส์ไปคลิก ที่ หน้าต่างบล็อกไดอะแกรม (ที่มี For Loop หรือ While Loop อยู่แล้ว) แล้วลากเมาส์ไปวางภายใน กรอบของ For Loop หรือ While Loop เท่านั้นเพราะถ้าวางเมาส์ภายนอกกรอบของ For Loop หรือ While Loop จะไม่สามารถวาง Feedback Node

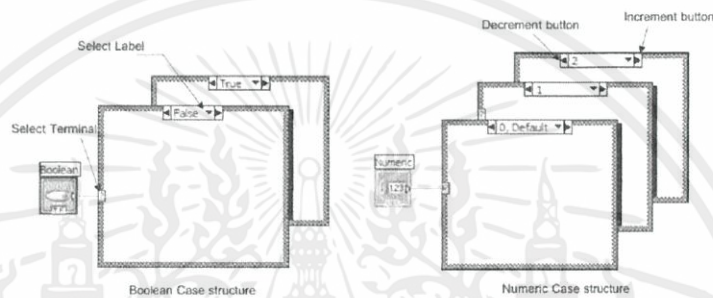


รูปที่ 3.38 แสดงการใช้งาน Feedback Node

#### ● Case Structure

Case Structure เป็นโครงสร้างการเขียนโปรแกรมตามเงื่อนไขของตัวหนังสือ (Conditional text) ซึ่งถ้าเปรียบเทียบกับภาษาเขียนโปรแกรมภาษาอื่นๆ (text-based programming languages) สามารถเปรียบเทียบได้กับโครงสร้างของคำสั่ง If...Then...Else การเรียกใช้งานฟังก์ชัน Case Structure ลงบนหน้าต่างบล็อกไดอะแกรมทำได้โดยเลือก Function» Structure แล้วใช้เมาส์เลือก ที่ Case Structure นำเมาส์ไปคลิกที่หน้าต่างบล็อกไดอะแกรมแล้วลากให้มี ขนาดที่ต้องการแล้ว ปล่อยเมาส์ก็จะได้โครงสร้างการทำงานของ Case Structure นอกจากนี้แล้วยัง สามารถทำการย่อ ขยายขนาดของ Case Structure ได้โดยใช้เมาส์ไปชี้ตรงบริเวณเส้นขอบของ Case Structure จะ

ปรากฏจุดสี่เหลี่ยมเล็กๆ บริเวณเส้นขอบของ While Loop ใช้เมาส์ไปวางตรงบริเวณจุดดังกล่าวแล้ว เมาส์จะเปลี่ยนเป็นรูปลูกศรจากนั้นคลิกเมาส์ค้างแล้วลากย่อหรือขยายได้ตามต้องการ สามารถแสดง ลักษณะของ Case Structure ได้ดังรูปที่ 3.39 ซึ่งมีส่วนประกอบที่สำคัญอยู่ 2 ส่วนคือ Select Terminal และ Select Label โดยที่ Select Terminal จะทำหน้าที่รับอินพุตเข้ามาแล้วทำการ ตรวจสอบหรือเปรียบเทียบว่าข้อมูลที่รับเข้ามาเป็นข้อมูลชนิดใดและตรงกับเงื่อนไขใดใน Case Structure เพื่อให้โปรแกรมทำงานตามเงื่อนไขที่ต้องการ ส่วน Select Label จะทำหน้าที่ สร้างหน้าต่างย่อยของ Case Structure ซึ่งสามารถเพิ่มและลดเงื่อนไขได้ถึง  $2^{31}-1$  เงื่อนไขการส่ง ข้อมูลออกเอาต์พุตนั้นทุกๆ กรณีจะส่งข้อมูลออกที่เอาต์พุตเดียวกับซึ่งทางออกของสัญญาณเอาต์พุต เรียกว่า Output Tunnel



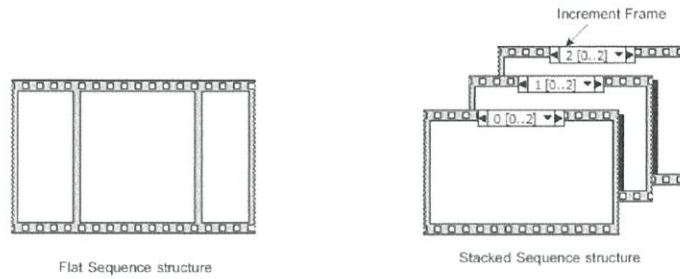
รูปที่ 3.39 แสดงโครงสร้างของ Case Structure

- Flat and Stack Sequence Structures

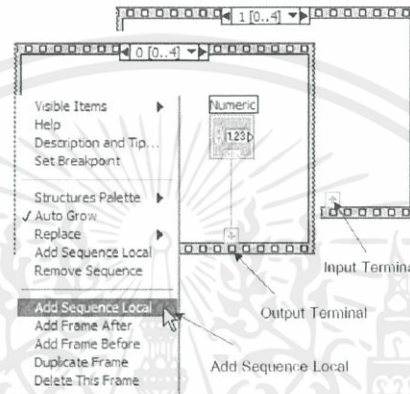
Sequence Structures เป็นการทำงานแบบลำดับขั้นรูปแบบของ Sequence Structures มีลักษณะเหมือนแผ่นฟิล์ม สำหรับ Sequence Structures มีทั้งหมด 2 แบบคือ Flat Sequence และ Stack Sequence ลักษณะของ Flat Sequence และ Stack Sequence แสดง ได้ดังรูปที่ 3.52 ข้อแตกต่างของทั้ง 2 แบบ คือ Flat Sequence มีมีลักษณะของเฟรมต่อกันไปไม่ว่าจะ เพิ่มหรือลดจำนวนเฟรมเราก็จะทั้งหมดว่ามีกี่เฟรมส่วน Stack Sequence จะมีลักษณะคล้ายกับ Case Structure กล่าวคือจะมี Select Label สำหรับเลือกเฟรมที่ซ่อนอยู่ภายในโดยจะตัวเลขบอกว่ากำลัง เปิด เฟรมใดอยู่

การสร้าง Flat และ Stack Sequence Structures ทำได้โดยการเลือกไปที่หน้าต่าง บล็อกไดอะแกรมแล้วเลือก Function» Structure แล้วใช้เมาส์เลือกที่ Flat หรือ Stack Sequence Structures ตามที่ต้องการ (แสดงดังรูปที่ 3.40) นำเมาส์ไปคลิกที่หน้าต่างบล็อกไดอะแกรมแล้วลาก ให้มีขนาดที่ต้องการแล้วปล่อยเมาส์ก็จะได้โครงสร้างการทำงานของ Flat และ Stack Sequence Structures

การที่จะนำข้อมูลของเฟรมหนึ่งไปยังอีกเฟรมหนึ่งสามารถทำได้โดยคลิกเมาส์ทางขวาที่ บริเวณกรอบ Sequence Structures แล้วเลือก Add Sequence Local จะปรากฏเทอร์มินอลขึ้นที่ ขอบของ Sequence Structures จะนั้นลากสายสัญญาณที่เราต้องการเอาออกเอาต์พุตไปต่อกับ เทอร์มินอลซึ่งเทอร์มินอลดังกล่าวก็จะเปลี่ยนเป็นลักษณะที่มีลูกศรพุ่งออกจากเฟรม ในขณะที่ทุกเฟรมที่เหลือก็จะปรากฏเทอร์มินอลขึ้นเช่นกันแต่จะมีลักษณะลูกศรชี้เข้าไปข้างในเฟรมแสดงดังรูป ที่ 3.41



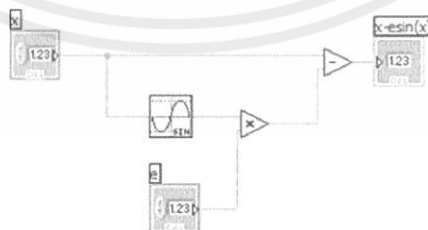
รูปที่ 3.40 แสดงโครงสร้างของ Flat และ Stack Sequence Structures



รูปที่ 3.41 การ Add Sequence Local ของ Stack Sequence Structures

- Formula Node

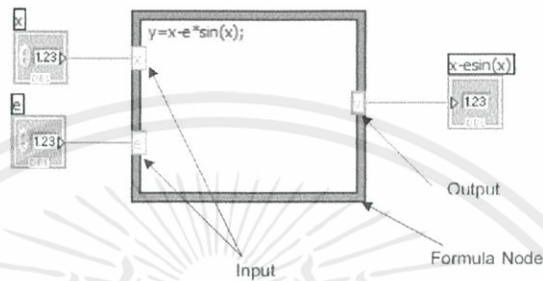
Formula Node เป็นโครงสร้างของคำสั่งที่มีความใกล้เคียงกับการเขียนโปรแกรมด้วยภาษาอื่นๆ ทั่วไปซึ่ง Formula Node จะถูกนำมาใช้เมื่อต้องการให้โปรแกรมทำงานตามฟังก์ชันทางคณิตศาสตร์ที่มีความยุ่งยาก การใช้อุปกรณ์ในการสร้างให้โปรแกรมทำงานตามฟังก์ชันทางคณิตศาสตร์ หนึ่งฟังก์ชันอาจต้องใช้อุปกรณ์หลายตัวซึ่งอาจทำให้การตรวจสอบโปรแกรมเป็นไปได้ด้วยความยุ่งยากและ โปรแกรมมีขนาดใหญ่เกินไป ยกตัวอย่างการสร้างโปรแกรมให้ทำงานตามสมการ  $y = x - e \sin(x)$  ซึ่งถ้าเขียนโปรแกรมด้วยการใช้อุปกรณ์ทั่วไปจะได้ลักษณะของโปรแกรมดังรูปที่ 3.42 แต่หากใช้ Formula Node จะได้ลักษณะของโปรแกรมดังรูปที่ 3.43



รูปที่ 3.42 การสร้างโปรแกรมจากอุปกรณ์ทั่วไป

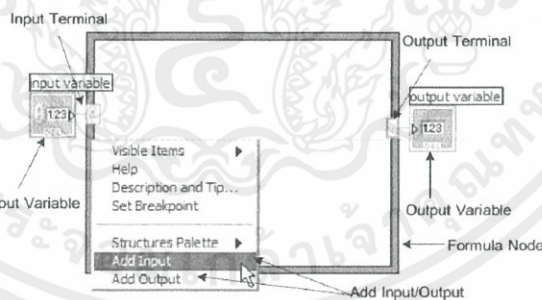
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นิยามให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทำงานของ Formula Node จะเป็นลักษณะของการกำหนดฟังก์ชันทางคณิตศาสตร์ ลงไปใน Formula Node (แสดงดังรูปที่ 3.43) แล้วจึงกำหนดเทอร์มินอลเพื่อเชื่อมต่ออินพุตและเอาต์พุต การสร้าง Formula Node ทำได้โดยการเลือกไปที่หน้าต่างบล็อกไดอะแกรมแล้วเลือก Function» Structure แล้วใช้เมาส์คลิกที่ Formula Node (แสดงดังรูปที่ 3.43) นำเมาส์ไปคลิกที่หน้าต่างบล็อกไดอะแกรมแล้วลากให้มีขนาดที่ต้องการแล้วปล่อยเมาส์ก็จะได้โครงสร้างการทำงานของ Formula Node



รูปที่ 3.43 โครงสร้างและการใช้งาน Formula Node

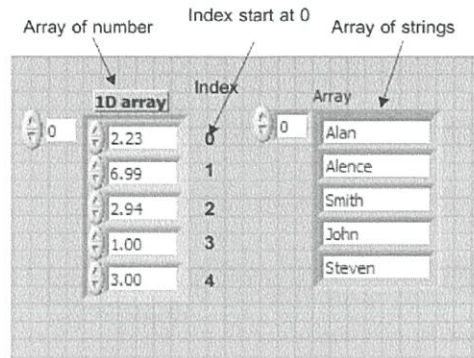
การกำหนดเทอร์มินอลอินพุตและเอาต์พุตสามารถทำได้โดยการคลิกเมาส์ขวาที่ขอบของ Formula Node แล้วเลือก Add Input หรือ Add Output (แสดงดังรูปที่ 3.44) ที่เทอร์มินอลสามารถที่จะกำหนดชื่อได้และค่าอินพุตและเอาต์พุตจะเป็นข้อมูลแบบตัวเลข (Numeric) การกำหนดตัวแปรอินพุตสามารถกำหนดให้เป็นค่าคงที่หรือค่าที่ปรับเปลี่ยนได้ส่วนเอาต์พุตจะกำหนดให้เป็นตัวแสดงผล ชื่อที่เทอร์มินอลก็คือชื่อของตัวแปรของฟังก์ชันทางคณิตศาสตร์ที่สร้างขึ้นใน Formula Node



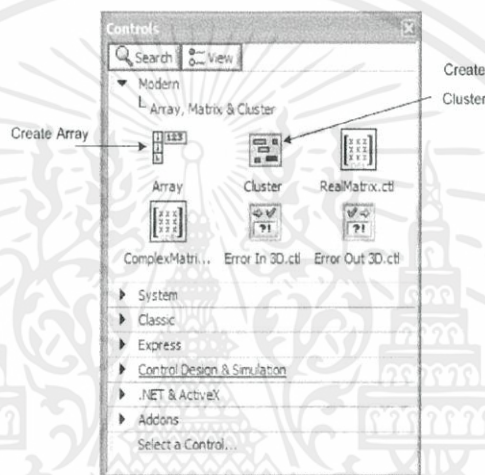
รูปที่ 3.44 แสดงการเพิ่มอินพุตและเอาต์พุตของ Formula Node

### 3.4.3.9 พื้นฐานการใช้งานอาร์เรย์

อาร์เรย์เป็นการเก็บตัวแปรหรือข้อมูลชนิดเดียวกันหลาย ๆ ข้อมูลไว้ด้วยกันส่วนคลัสเตอร์จะเก็บข้อมูลที่ชนิดต่างกันไว้ด้วยกันได้ ปัญหาที่พบบ่อยคือเวลาโปรแกรมทำงานจะมีข้อมูลเป็นจำนวนมาก ฉะนั้นอาร์เรย์หรือคลัสเตอร์จะช่วยให้การเก็บข้อมูลอาร์เรย์สามารถสร้างได้หลายมิติเริ่มตั้งแต่ 1 มิติ ซึ่ง ในอาร์เรย์ 1 มิติสามารถเก็บข้อมูลได้สูงถึง  $2^{31}-1$  ข้อมูลค่าสูงสุดที่จะเก็บได้ก็ขึ้นอยู่กับขนาดของหน่วยความจำอาร์เรย์ในโปรแกรม LabVIEW สามารถเก็บข้อมูลได้หลายชนิดยกเว้น Chart และ Graph วิธีการอ้างถึงข้อมูลในอาร์เรย์เรียกว่าดัชนี (Index) ซึ่งมีค่าเริ่มต้นตั้งแต่ 0 ไปจนถึง N-1 เมื่อ N คือ จำนวนข้อมูลในอาร์เรย์ ในรูปที่ 3.45 เป็นอาร์เรย์ 1 มิติ โดยมีค่าดัชนีเริ่มต้นที่ 0



รูปที่ 3.45 แสดง Array One Dimension (1-D)



รูปที่ 3.46 การเรียกใช้งานอาร์เรย์

- การสร้างอาร์เรย์ด้วย Control และ Indicator

เราสามารถสร้างอาร์เรย์ด้วย Control และ Indicator ของ Numeric, Boolean, Strings ซึ่งการสร้างอาร์เรย์จะประกอบไปด้วย Array Shell และ Matrix & Cluster อยู่ในพาเลทย่อยของ Controls» Modern ดูได้จากรูปที่ 3.47 ขั้นตอนในการสร้างอาร์เรย์ด้วย Control และ Indicator มีดังนี้

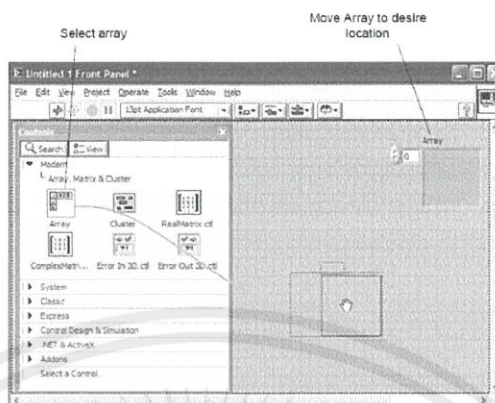
- เลือกบล็อกของอาร์เรย์จาก Array, Matrix & Cluster ในพาเลทย่อยของ Control» Modern พาเลท แล้วลากไปวางในหน้าต่างพร้อมท์พาเนล ดังแสดงในรูปที่ 3.47

- เลือกข้อมูลที่ต้องการ เช่น Numeric, Boolean และ String แล้วลากไปวางไว้ในบล็อก อาร์เรย์ ดังแสดงในรูปที่ 3.47 โดยที่อาร์เรย์จะปรับขนาดให้เหมาะสมกับชนิดของข้อมูลที่น่ามาวาง

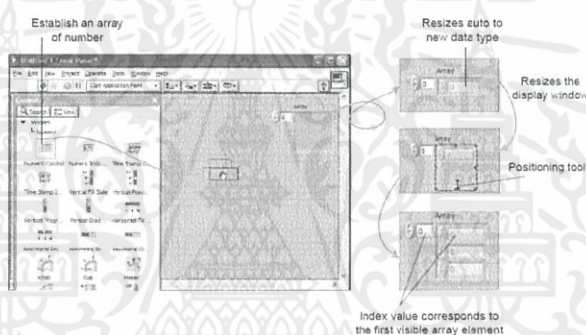
เราสามารถที่จะย่อหรือขยายอาร์เรย์เพื่อดูค่าของสมาชิกที่อยู่ภายในอาร์เรย์ได้โดยใช้เมาส์ไปวางบริเวณอาร์เรย์ แล้วจะปรากฏจุดสี่เหลี่ยมเล็กๆ ขึ้นรอบๆ บล็อกของอาร์เรย์ จากนั้นใช้เมาส์ไปวางบริเวณจุดเหล่านั้นจุดใดก็ได้ตามต้องการ แล้วเมาส์จะเปลี่ยนไอคอนไปเป็นลูกศรแล้วทำการคลิกเมาส์ค้างแล้วลากย่อขยายตามที่ต้องการ

ครั้งแรกที่เราลากบล็อกของอาร์เรย์ไปวางในหน้าต่างพร้อมท์พาเนลจะปรากฏบล็อกของ array shell ซึ่งจะเป็นแสดงชนิดของข้อมูลที่ไม่ว่าค่า ดังรูปที่ 3.48 เมื่อเรลาก string, Boolean

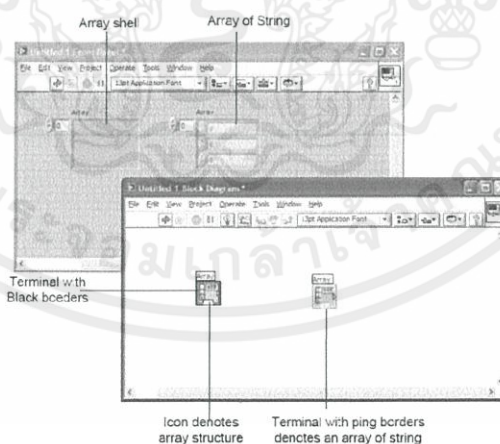
หรือ numeric ไปวางในบล็อกดังกล่าวในหน้าต่างบล็อกไดอะแกรมจะสังเกตเห็นได้ว่าเทอร์มินอลจะเปลี่ยนเป็นสีส้ม แสดงดังรูปที่ 3.49



รูปที่ 3.47 แสดงอาร์เรย์สร้างจาก Control และ Indicator



รูปที่ 3.48 ขั้นตอนการสร้างอาร์เรย์จาก Control และ Indicator

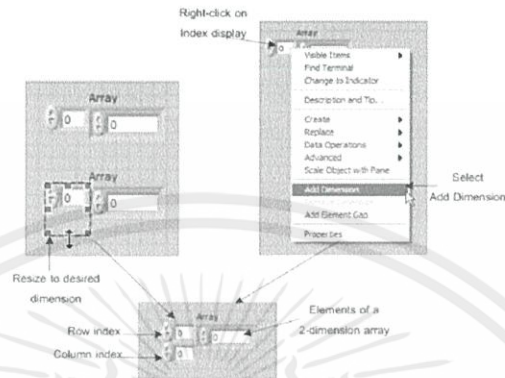


รูปที่ 3.49 เทอร์มินอลของ array shell

#### ■ อาร์เรย์หลายมิติ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษายเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
อาร์เรย์ 2 มิติ (2D) จะมีตัวบอกคุณลักษณะอยู่ 2 ตัว คือ บอกแถวและบอกหลัก อาร์เรย์ 3 มิติ ก็จะมีตัวบอกคุณลักษณะ 3 ตัว ดังนั้นอาร์เรย์ n มิติ ก็จะมีตัวบอกคุณลักษณะ n ตัว การเพิ่มมิติของอาร์เรย์สามารถทำได้ 2 วิธีคือ ใช้ Positioning Tool เพื่อเพิ่มมิติของอาร์เรย์และคลิกเมาส์ขวาที่

อาร์เรย์แล้วเลือกคำสั่ง Add Dimension จากเมนูย่อย ซึ่งวิธีการทั้งสองที่ใช้ในการเพิ่มมิติของอาร์เรย์ สามารถแสดงได้ดังรูปที่ 3.50 การลดมิติของอาร์เรย์ก็สามารถทำได้สองวิธีเช่นเดียวกันกับการเพิ่มคือ ใช้ Positioning Tool เพื่อลดขนาดมิติของอาร์เรย์และคลิกเมาส์ขวาที่อาร์เรย์แล้วเลือกคำสั่ง Remove Dimension จากเมนูย่อย



รูปที่ 3.50 แสดงวิธีการเพิ่มมิติของอาร์เรย์

- ฟังก์ชันของอาร์เรย์

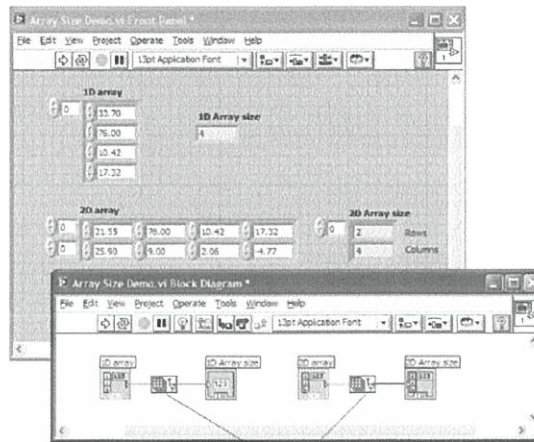
- Array Size

ฟังก์ชัน Array Size จะเป็นตัวบอกขนาดของอาร์เรย์ว่ามีขนาดใด เช่น ถ้าอาร์เรย์อินพุตเป็นอาร์เรย์ 1 มิติ (4 แถว 1 หลัก) อาร์เรย์เอาต์พุตจะเป็นอาร์เรย์ 1 มิติ มีจำนวนสมาชิกทั้งหมด 4 ตัว หรือถ้าอาร์เรย์อินพุตเป็นอาร์เรย์ 2 มิติ (2 แถว 4 หลัก) อาร์เรย์เอาต์พุตจะแสดงเป็นอาร์เรย์ 2 มิติ (แสดงจำนวนของแถวและหลัก) แสดงดังรูปที่ 3.51

- Initialize Array

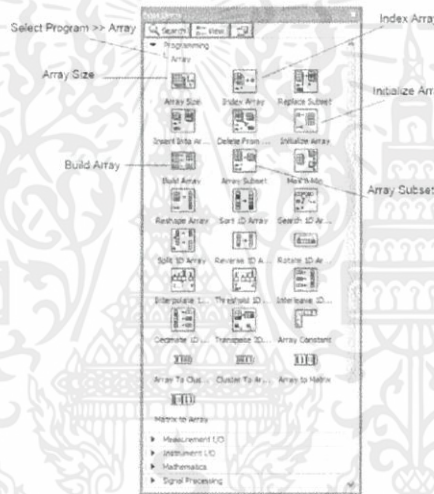
เป็นการสร้างอาร์เรย์ตามขนาดที่กำหนดพร้อมทั้งสามารถกำหนดค่าเริ่มต้นให้กับอาร์เรย์ที่สร้าง ได้อีกด้วย เช่น ต้องการสร้างอาร์เรย์ 1 มิติ มีจำนวนสมาชิกทั้งหมด 50 ตัว และกำหนดให้ค่าเริ่มต้นของอาร์เรย์เป็น 2.5 สามารถสร้างได้โดยคลิกเมาส์ขวาที่เทอร์มินอล element ซึ่งอยู่ทางด้านบนซ้ายของ บล็อก Initialize Array แล้วเลือก create» constant และกำหนดค่าเริ่มต้นเป็น 2.5 ที่เทอร์มินอล dimension size ของบล็อก Initialize Array ที่อยู่ด้านล่างเทอร์มินอล element คลิกเมาส์ขวาแล้ว เลือก create» constant และกำหนดให้เป็น 50 (จำนวนสมาชิก) ถ้าต้องการเพิ่มมิติของอาร์เรย์ก็สามารถทำได้โดยคลิกเมาส์ขวาที่บล็อกของ Initialize Array แล้วเลือก Add dimension ซึ่งจะเพิ่มเทอร์มินอลเพื่อกำหนดขนาดของแถวและหลักแสดงดังรูปที่ 3.52

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

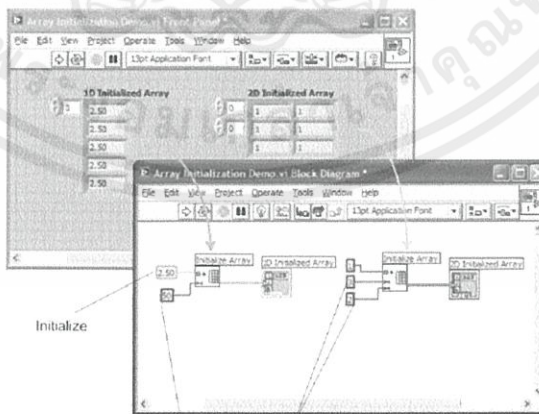


Array Size function

รูปที่ 3.51 แสดงฟังก์ชันอาร์เรย์



รูปที่ 3.52 แสดงการใช้งาน Array Size Function



Initialize

1D Array 2D Array

รูปที่ 3.53 แสดงการใช้งาน Initialize Array Function

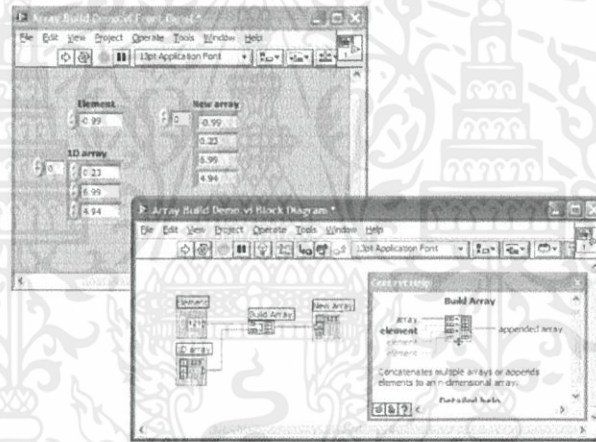
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### ■ Build Array

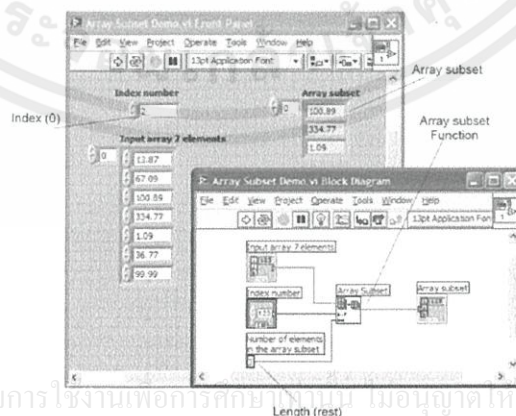
เป็นการนำเอาอาร์เรย์หรือสมาชิกในอาร์เรย์หลาย ๆ ตัวมารวมกันแล้วเกิดเป็นอาร์เรย์ใหม่ ซึ่งอินพุตที่สามารถที่จะเป็นทั้งอาร์เรย์และเป็นจำนวนสเกลล่า โดยปกติแล้ว Build Array จะมีอินพุต 2 อินพุต ถ้าต้องการเพิ่มจำนวนอินพุตสามารถทำได้โดยการคลิกเมาส์ขวาที่บล็อก Build Array แล้วเลือก Add Input หรือจะใช้ Positioning Tool คลิกค้างแล้วลากขยายเพื่อเพิ่มอินพุตและถ้าต้องการลดจำนวนอินพุตก็สามารถทำได้โดยคลิกเมาส์ขวา Build Array แล้วเลือก Remove Input ตัวอย่างการใช้งาน Build Array แสดงดังรูปที่ 3.54

### ■ Array Subset

เป็นวิธีการเลือกค่าของข้อมูลที่อยู่ในอาร์เรย์ไปใช้งานโดยมีสิ่งที่จะต้องกำหนดอยู่ 2 อย่างคือ Index (0) ซึ่งเป็นการกำหนดตำแหน่งเริ่มต้นที่เราต้องการ (ตำแหน่งแรกนับเป็นตำแหน่งที่ 0) และ Length (rest) เพื่อกำหนดจำนวนข้อมูลที่ต้องการ ดังตัวอย่างในรูปที่ 3.54 อาร์เรย์ 1 มิติ มีจำนวนข้อมูล 7 ข้อมูล ต้องการใช้อินพุตตั้งแต่ตำแหน่งที่ 3 ถึง 5 ทำได้โดยกำหนด Index (0) ให้มีค่าเท่ากับ 2 และ Length (rest) ให้มีค่าเท่ากับ 3 แสดงดังรูปที่ 3.55



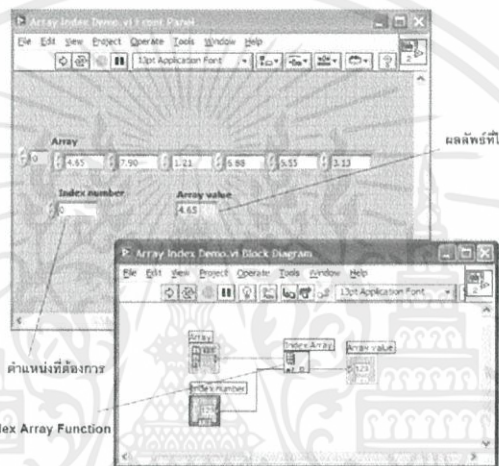
รูปที่ 3.54 แสดงการใช้งาน Build Array Function



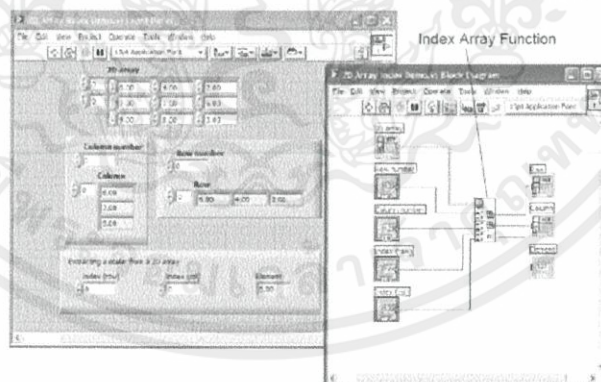
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้รูปที่ 3.55 แสดงการใช้งาน Subset Array Function ครั้งที่มีการนำไปใช้

### ■ Index Array

ฟังก์ชัน Index Array สามารถที่จะเลือกข้อมูลในอาร์เรย์ในตำแหน่งที่ต้องการมาใช้งานได้ โดยการเลือกตำแหน่งของข้อมูลได้เลย ในกรณีที่เป็นอาร์เรย์ขนาด 2 มิติ ก็สามารถใช้ Index Array เลือกได้ว่า จะเลือกข้อมูลออกเอาที่พุดที่ตำแหน่งแถวและหลัก แสดงดังรูปที่ 3.56 มีอาร์เรย์ 1 มิติ มีจำนวนข้อมูล ทั้งหมด 6 ข้อมูล ต้องการเลือกข้อมูลที่ 1 ไปใช้งานสามารถทำได้โดยกำหนดให้ Index Number มีค่า เท่ากับ 0 (ตำแหน่งแรกนับเป็นตำแหน่งที่ 0) และในรูปที่ 3.57 เป็นการใช้ฟังก์ชัน Index Array ในการ เลือกข้อมูลจากอาร์เรย์ 2 มิติ ขนาด 3 x 3 โดย Index Array จะต้องมี 2 ตัว เพื่อเลือกตำแหน่งแถวและตำแหน่งหลักที่ต้องการ ซึ่งการเพิ่มอินพุตของ Index Array สามารถทำได้ โดยการใช้ Position Tool คลิกที่ Index Array แล้วลากขยายเพื่อเพิ่มอินพุตให้กับ Index Array



รูปที่ 3.56 แสดงการใช้งาน Index Array Function



รูปที่ 3.57 แสดงการใช้งาน Index Array Function กับอาร์เรย์ 2 มิติ

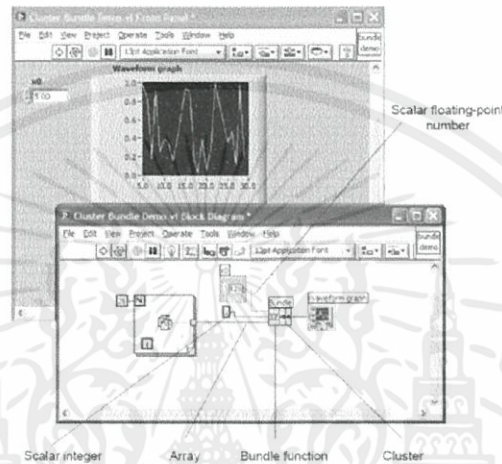
#### 3.4.3.10 พื้นฐานการใช้งานคลัสเตอร์

คลัสเตอร์เป็นเครื่องมือที่ทำหน้าจัดเก็บข้อมูลคล้ายกับอาร์เรย์เพียงแต่ว่าคลัสเตอร์สามารถเก็บข้อมูลที่ต่างชนิดกันได้ เช่นเก็บข้อมูลที่เป็น Numeric, Boolean และ String ไปด้วยกันได้ วิธีการเรียกใช้งานคลัสเตอร์ทำได้โดยเลือกที่ Controls Palette» Modern» Array & Cluster» Cluster



จะ อยู่ด้านบนของบล็อก เราสามารถที่จะเพิ่มหรือลดจำนวนเทอร์มินอลอินพุตให้มีจำนวนเท่าที่เราจำเป็นต้องใช้งานโดยใช้เมาส์คลิกที่บล็อก แล้วเลื่อนขึ้นลงในแนวตั้งหรือใช้เมาส์ไปวางตรงเทอร์มินอลอินพุตแล้วคลิกเมาส์ขวา แล้วเลือก Add Input ถ้าต้องการเพิ่มเทอร์มินอล และเลือก Remove Input ถ้าต้องการลดเทอร์มินอล

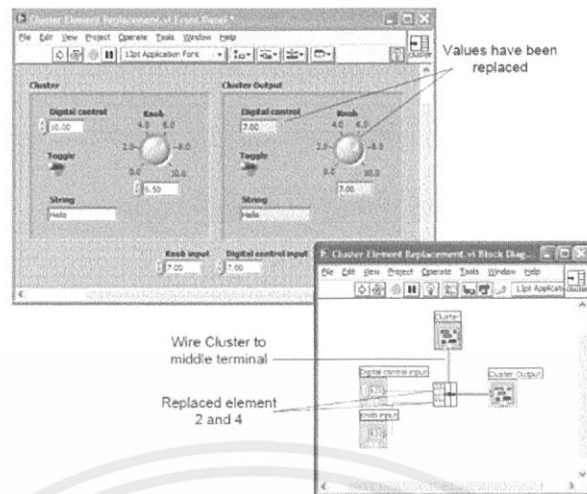
รูปที่ 3.60 จะเป็นตัวอย่างการสร้างคลัสเตอร์ด้วย Bundle โดยมีอินพุต 3 ชนิดคือ floating-point number, Scalar integer และ array ที่สร้างจาก For Loop และมีเอาท์พุตที่แสดงผลด้วยกราฟ



รูปที่ 3.60 แสดงการใช้งาน Bundle Function

การเพิ่มเทอร์มินอลอินพุตของฟังก์ชัน Bundle ทางซ้ายในขณะที่มีอินพุตที่เป็นคลัสเตอร์อยู่ด้วยโดยไม่มีผลกระทบต่อโปรแกรมเดิม ดังตัวอย่างในรูปที่ 3.60 ซึ่งมีคลัสเตอร์ 2 ตัว หนึ่งตัวเป็นอินพุตและอีกหนึ่งตัวเป็นเอาท์พุต การทำงานของโปรแกรมคือเอาท์พุตจะทำงานแปรตามอินพุต ถ้าเราต้องการควบคุมเอาท์พุตโดยที่ไม่ต้องไปแก้ไขอินพุตเราสามารถที่จะเพิ่มอินพุตให้กับ Bundle และสร้างอินพุตให้กับ Bundle ให้ตรงกับเอาท์พุตที่เราต้องการควบคุมจากรูปที่ 3.61 การทำงานของโปรแกรมในขณะที่ มีอินพุตที่เป็นคลัสเตอร์เพียงอย่างเดียว เอาท์พุตจะแปรตามอินพุตแต่เมื่อเราเพิ่มอินพุตแบบ knop และ อินพุต numeric control เข้าไปที่เทอร์มินอลอินพุตของ Bundle เมื่อเราทดสอบโปรแกรมจะเห็นได้ว่า เมื่อเราเปลี่ยนค่าอินพุตแบบ knop และอินพุต numeric control ในคลัสเตอร์อินพุต ค่าภายในคลัสเตอร์เอาท์พุตจะไม่เปลี่ยนแปลงแต่เมื่อเราเปลี่ยนค่าอินพุตที่เราเพิ่มเข้าไปค่าเอาท์พุตก็จะเปลี่ยนตาม

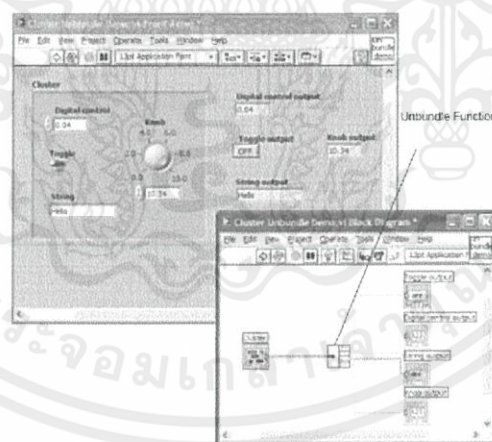
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.61 แสดงการใช้งาน Bundle Function

#### ■ Unbundle Function

ในคลัสเตอร์จะประกอบไปด้วยชนิดของข้อมูลที่แตกต่างกัน Unbundle Function จะช่วยในการแยกข้อมูลอินพุตที่แตกต่างกันออกจากกันเพื่อจะนำเอาที่พุดนั้นไปใช้งานต่อไป ฟังก์ชันของ Unbundle ที่วางลงไปบนหน้าต่างบล็อกไดอะแกรมครั้งแรกจะมีอินพุต 1 อินพุต และมีเอาต์พุต 2 อินพุต เอาต์พุต Unbundle จะเปลี่ยนไปตามอินพุตที่เป็นคลัสเตอร์ ตัวอย่างเช่น มีคลัสเตอร์อินพุตที่มีชนิดของข้อมูลแตกต่างกัน 4 ชนิด เมื่อต่อสายของคลัสเตอร์เข้ากับอินพุตของ Unbundle เอาต์พุตของ Unbundle จะเปลี่ยนเป็น 4 เอาต์พุตโดยอัตโนมัติ

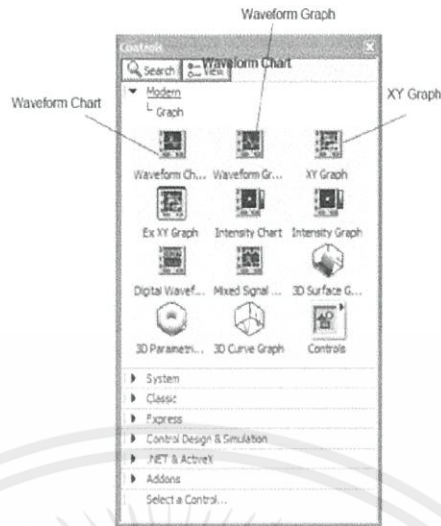


รูปที่ 3.62 แสดงการใช้งาน Unbundle Function

#### 3.4.3.11 พื้นฐานการใช้งาน Charts และ Graph Waveform Charts

การแสดงผลข้อมูลในลักษณะของ Chart เป็นการแสดงผลการพล็อตรูปคลื่นสัญญาณข้อมูลแบบหนึ่งที่สามารถพล็อตข้อมูลที่เป็นสเกลาร์และเป็นอาร์เรย์ได้ การเรียกใช้งาน คือเลือกที่ Controls Palette » Graph » Waveform Chart

Waveform Chart ในโปรแกรม LabVIEW มีเพียงชนิดเดียวแต่มีคุณสมบัติในการแสดงผลข้อมูลที่แตกต่างกัน 3 อย่างคือ Strip Chart, Scope Chart และ Sweep Chart ซึ่งข้อแตกต่างของ Chart ทั้ง 3 แบบ อธิบายได้ดังรูปที่ 3.63

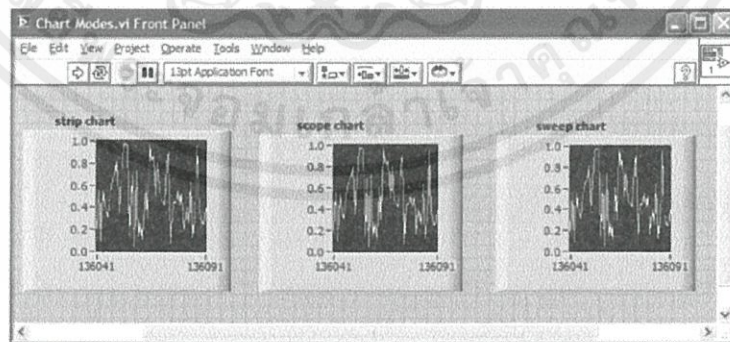


รูปที่ 3.63 แสดงวิธีการเรียกใช้งาน Waveform Chart

Strip Chart : มีคุณสมบัติในการพล็อตข้อมูลไปจนกว่าจะหยุดพล็อตซึ่งจะสังเกตเห็นว่าเส้นของสัญญาณข้อมูลที่พล็อตนั้นจะเลื่อนจากขวาไปซ้ายต่อเนื่องกันไป

Scope Chart : คุณสมบัติของการพล็อตแบบนี้คือจะแบ่งเป็นช่วงของข้อมูลโดยการกำหนดที่ แกนเวลาเมื่อโปรแกรมทำการพล็อตข้อมูลไปถึงค่าสุดท้ายของช่วงแรกแล้วการพล็อตข้อมูลของช่วงต่อไป โปรแกรมจะทำการลบกราฟในช่วงแรกไปแล้วเริ่มการพล็อตใหม่โดยเวลาเริ่มต้นจะนับต่อจากค่าสุดท้าย ของช่วงแรก

Sweep Chart : จะทำงานคล้ายกับ Scope Chart แต่เมื่อเริ่มทำการพล็อตกราฟในช่วงต่อไป เส้นกราฟในช่วงแรกจะไม่ถูกลบจะทำงานในลักษณะพล็อตทับเส้นกราฟเดิมและจะมีเส้นสีแดงในแนว เลื่อนจากซ้ายไปขวาเพื่อแสดงถึงจุดการพล็อตกราฟครั้งล่าสุด คุณสมบัติทั้ง 3 ของ Waveform Chart แสดงดังรูปที่ 3.64 เป็นการแสดงค่าของข้อมูลที่เก็บอยู่ในอาร์เรย์เทียบกับเวลาที่รับเข้ามา



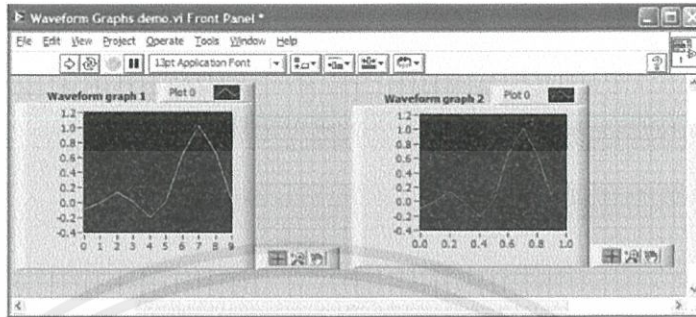
รูปที่ 3.64 แสดงคุณสมบัติของ Waveform Chart

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

- Waveform Graphs

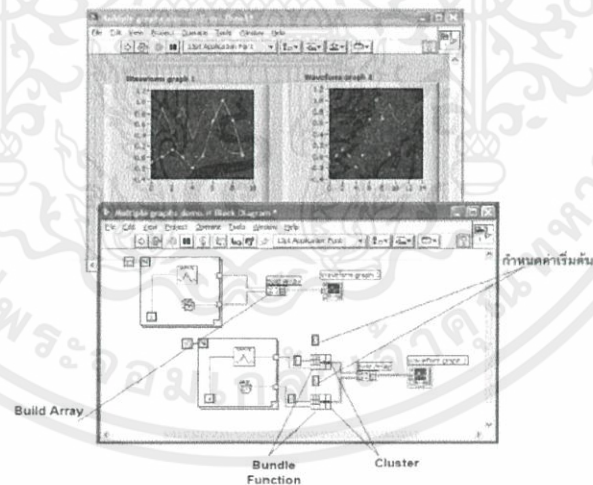
แตกต่างจาก Waveform Chart คือการพล็อตข้อมูลด้วย Waveform Graphs จะมีการกำหนด จำนวนของช่วงข้อมูลอย่างชัดเจนจำนวนข้อมูลทั้งหมดที่อยู่ในช่วงของกราฟจะถูกแสดงออกมาทั้งหมด การพล็อตกราฟหลายเส้นทำได้โดยการนำอาร์เรย์แต่ละตัวมาต่อรวมเพื่อสร้างอาร์เรย์

ใหม่จำนวน เส้นกราฟที่แสดงจะเท่ากับมิติของอาร์เรย์ การเรียกใช้งาน Waveform Graphs ทำได้คือ เลือกที่ Controls Palette» Modern» Graph» Waveform Graphs ลักษณะของการพล็อตกราฟ ลงบน waveform graph แสดงได้รูปที่ 3.65



รูปที่ 3.65 แสดงคุณสมบัติของ Waveform Graph

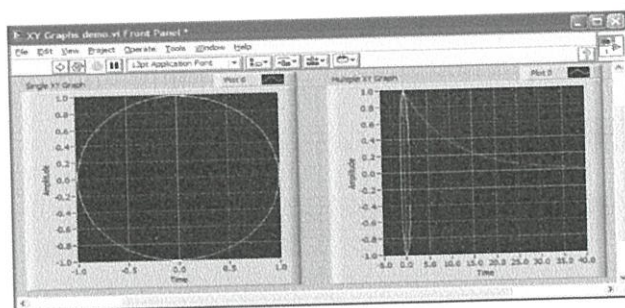
นอกจากนี้แล้วเรายังสามารถใช้ Bundle และ Cluster มาช่วยในการกำหนดค่าเริ่มต้นและความถี่ในการแสดงข้อมูลแสดงดังรูปที่ 3.66 กราฟแรกจะแสดงการสร้างสัญญาณ 2 เส้นให้อยู่บนกราฟเดียวกันโดยใช้ฟังก์ชัน Build Array เข้ามาช่วย กราฟที่สองจะเป็นการสร้างสัญญาณ 2 เส้นให้อยู่บนกราฟเดียวกันและให้สามารถกำหนดค่าเริ่มต้นของกราฟแต่ละเส้นได้ ซึ่งเราจะนำเอา Bundle และคลัสเตอร์โดยกราฟเส้นแรกให้เริ่มต้นที่ค่า 0 แสดงค่าทั้งหมด 10 ค่า และอีกเส้นให้เริ่มต้นที่ 5 และค่าที่ แสดงทั้งหมด 10 ค่า เช่นกัน ดังแสดงในรูปที่ 3.66



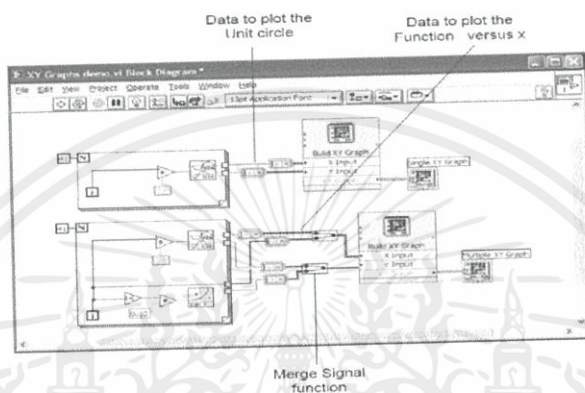
รูปที่ 3.66 แสดงคุณสมบัติของ Waveform Graph

- XY Graphs

การพล็อตกราฟใน XY Graph เป็นการนำค่าของอาร์เรย์ 2 ชุดมาพล็อตในระนาบ XY โดยกำหนดให้อาร์เรย์ตัวที่ 1 และตัวที่ 2 เป็นค่าที่แสดงในแกน X และแกน Y ตามลำดับ โดยจะมีเอกสารนี้เป็นที่รู้จักในชื่อ Build XY Graph จะช่วยในการแปลงข้อมูลในอาร์เรย์ให้เป็นค่าใด ๆ ที่สามารถพล็อตในแกน XY ได้ แสดงวิธีการพล็อต XY Graph ดังรูปที่ 3.67



(a) Front Panel



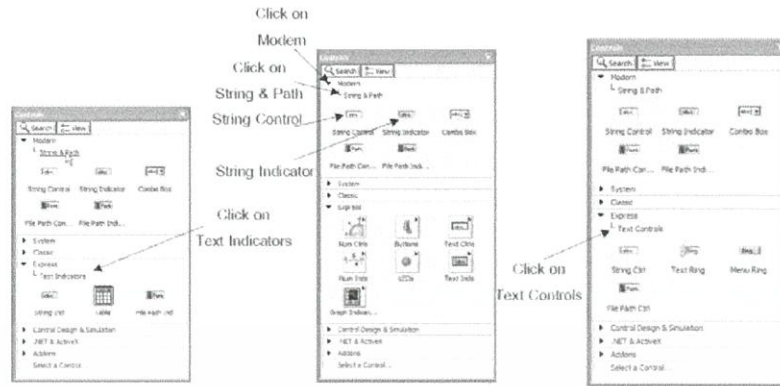
(b) Block Diagram

รูปที่ 3.67 แสดงคุณสมบัติของ XY Graph

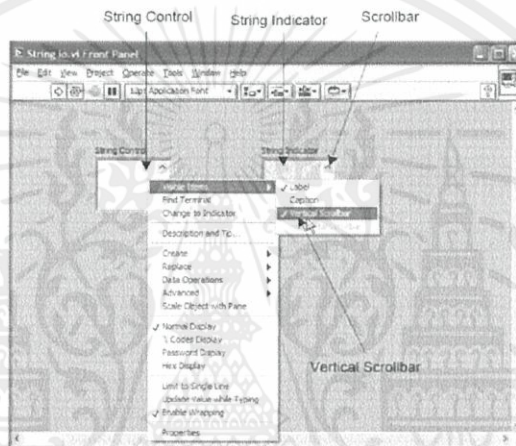
### 3.4.3.12 พื้นฐานเกี่ยวกับสตริง

สตริงเป็นชนิดของข้อมูลที่อยู่ในรูปแบบของตัวอักษร ซึ่งสามารถที่จะกำหนดให้แสดงหรือไม่แสดงค่าก็ได้หลายครั้งที่สตริงถูกใช้ในลักษณะของรหัส ASCII ในโปรแกรม LabVIEW สตริงจะถูกใช้ใน instrument control การใช้งานข้อมูลชนิดสตริงและข้อมูลชนิดตัวเลขจะต้องมีการเปลี่ยนชนิดของข้อมูลเพื่อให้ชนิดของข้อมูลเป็นชนิดเดียวกันก่อน ซึ่งก็จะมีวิธีการเปลี่ยนชนิดของข้อมูลซึ่งจะได้กล่าวในหัวข้อต่อไป ชนิดของข้อมูลก่อนที่จะบันทึกลงไฟล์บนฮาร์ดดิสก์จะต้องแปลงให้เป็นข้อมูลชนิดสตริงก่อน ในหัวข้อนี้จะพูดถึงสตริงและการใช้ไฟล์อินพุตไฟล์เอาต์พุต (File I/O)

การใช้งาน String Control และ String Indicator ที่อยู่ใน String & Path ซึ่งเป็นพาเลทย่อยของ Control» Modern ซึ่งแสดงในรูปที่ 3.67 ซึ่งเราสามารถเลือกใช้งานได้ทั้งแบบ Text Controls และ Text Indicators ซึ่งอยู่ในพาเลทย่อยของ Controls»Express การใช้งาน String Control และ String Indicator ทำได้โดยการคลิกเมาส์ทางซ้ายค้ำที่สัญลักษณ์ของ String Control และ String Indicator แล้วลากไปวางที่หน้าต่างพร้อมท์พาเนล ซึ่งเราจะเห็นลักษณะของ String Control และ String Indicator เป็นบล็อกสี่เหลี่ยมสำหรับรับข้อมูลอินพุตและแสดงผลตามลำดับ ถ้าบล็อกนั้นมีขนาดเล็กไม่เพียงพอสำหรับข้อมูลที่ป้อนเข้าไป ก็สามารถขยายได้ใช้เมาส์คลิกแล้วลากขยายให้ใหญ่ขึ้นหรือถ้าขนาดของพร้อมท์พาเนลมีขนาดจำกัดซึ่งทำให้เราไม่สามารถขยายขนาดของบล็อกสตริงได้ก็สามารถเพิ่ม Scrollbar เข้าไปเพื่อให้สามารถกรอกข้อมูลชนิดสตริง ได้เป็นจำนวนมากขึ้น ซึ่งทำได้โดยคลิกเมาส์ ทางขวาที่บริเวณบล็อกของ String Control หรือ String Indicator ที่ต้องการเพิ่ม Scrollbar แล้วเลือก Visible Item» Vertical Scrollbar ดังรูปที่ 3.68



รูปที่ 3.68 การเข้าถึง String controls และ String Indicators

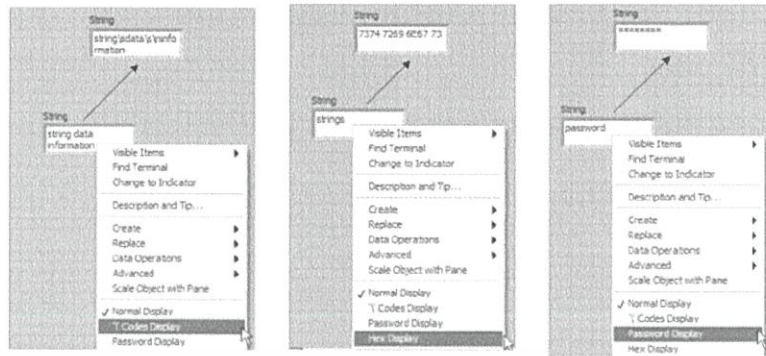


รูปที่ 3.69 การเพิ่ม Scrollbar ให้กับบล็อก String Indicator และ String Control

สกอร์บาร์ที่เพิ่มเข้าไปใน String Control หรือ String Indicator ดังแสดงในรูปที่ 3.70 สกอร์บาร์ที่เพิ่มเข้าไปแสดงอยู่ในแนวตั้ง การแสดงข้อมูลในบล็อกของ String Control และ String Indicator นั้นสามารถกำหนดให้แสดงได้ 4 รูปแบบ คือ

1. Normal Display จะแสดงข้อความแบบปกติตามที่เรากำหนดเข้าไป
2. “\” Codes Display จะแสดงข้อความในลักษณะของ Codes โดยมี เครื่องหมาย “\” เป็น ตัวขึ้นแทนการเว้นวรรค
3. Password Display จะแสดงข้อความแบบรหัสผ่านแทนด้วย “\*”
4. Hex Display จะแสดงข้อความในลักษณะของเลขฐาน 16

โดยที่ลักษณะของข้อความหลังจากที่ถูกเข้ารหัสในรูปแบบต่างๆ แล้วสามารถแสดงให้เห็นได้ ดังรูปที่ 3.69 ซึ่งในรูปจะแสดงวิธีการเข้ารหัสแบบ “\” Codes Display, Hex Display และ Password Display ตามลำดับ การกำหนดรูปแบบของชนิดของข้อมูลแบบสตริงว่าจะให้แสดงอยู่ในรูปแบบใดนั้นสามารถทำได้โดยคลิกเมาส์ทางขวามือของบล็อกของ String Control และ String Indicator จะปรากฏเมนูย่อยขึ้นมาซึ่งในเมนูย่อยนั้นจะมีชุดคำสั่งสำหรับกำหนดรูปแบบการแสดงผลอยู่ ทั้งหมด 4 รูปแบบ



รูปที่ 3.70 แสดงตัวอย่างการเข้ารหัสแบบต่างๆ

ถ้าต้องการให้แสดงผลของข้อมูลในรูปแบบใดก็ใช้เมาส์คลิกที่คำสั่งนั้น รูปแบบที่แสดงผลก็จะเปลี่ยนไปตามรูปแบบที่เลือกแสดงได้ดังรูปที่ 3.70

ในตารางที่ 3.9 จะเป็นตารางแสดงสัญลักษณ์และความหมายของข้อมูลเมื่อเราทำการแสดงผลข้อมูลแบบโค้ด

ตารางที่ 3.9 แสดง Backslash Codes

Code	G Interpretation
\00 - \FF	Hex value of an 8-bit character; must be uppercase
\b	Backspace (ASCII BS, equivalent to \08)
\f	Form feed (ASCII FF, equivalent to \0C)
\n	Line feed (ASCII LF, equivalent to \0A)
\r	Carriage return (ASCII CR, equivalent to \0D)
\t	Tab (ASCII HT, equivalent to \09)
\s	Space (equivalent to \20)
\\	Backslash (ASCII \, equivalent to \5C)

#### 3.4.3.13 การสร้างสตริง

ในหัวข้อนี้จะยกตัวอย่างการใช้งานสตริง ซึ่งจะยกตัวอย่างการใช้งานทั้งหมด 2 แบบ คือ

- Format Into String:
- Scan From String

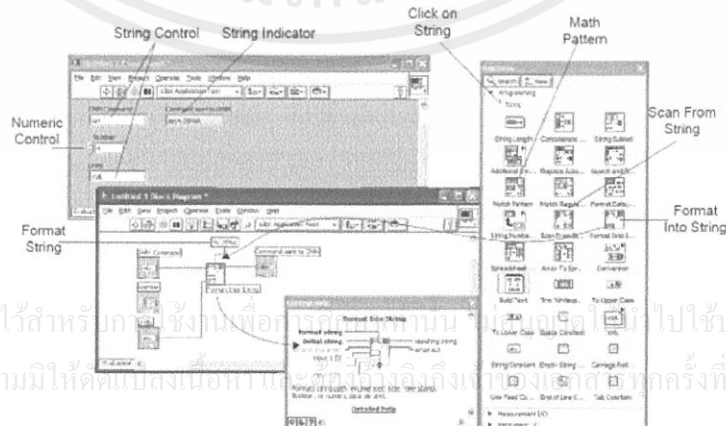
ในโปรแกรม LabVIEW สามารถที่จะสร้างโปรแกรมจำลองการทำงานของดิจิตอลมัลติมิเตอร์ (DMM) โดยเริ่มจาก เปิดหน้าต่างพร้อมท์พาเนลของโปรแกรม LabVIEW จากนั้นวาง String control ลงไป 2 ตัว, numeric control 1 ตัว, และ String Indicator อีก 1 ตัว จากนั้นเลือกหน้าต่างบล็อกไดอะแกรมขึ้นมาแล้วเลือก Format Into String จาก Palette ย่อยของ String แสดงดังรูปที่ 3.83 สำหรับตัวอย่างนี้จะแสดงการแปลงข้อมูลที่เป็นตัวอักษรและการนำข้อมูลอินพุตมาเชื่อมต่อกันเพื่อกำหนดรูปแบบคำสั่งสำหรับดิจิตอลมัลติมิเตอร์ (DMM) โดยใช้ Positioning Tool ในการขยายให้ Format Into String เพิ่มเป็น 2 อินพุต ซึ่ง input connector จะอยู่ทางด้านล่างซ้ายมือของบล็อก Format Into String จากนั้นลากสายสัญญาณเชื่อมต่อระหว่างบล็อกอินพุตกับบล็อก Format

Into String โดยที่ต่ออินพุตตัวแรกเข้ากับ numeric control และต่ออินพุตตัวที่ 2 เข้ากับ String control ชนิดของสัญญาณอินพุตสำหรับ Format Into String จะเปลี่ยนเป็นแบบ DBL หรือ abc แบบอัตโนมัติ จากนั้นต่อ String control เข้ากับอินพุต initial string ที่อยู่ทางด้านบนขวาของบล็อก Format Into String ซึ่งจะกำหนดให้เป็นชุด DMM Command และสุดท้ายให้ต่อ resulting string เข้ากับ string indicator ซึ่งเป็นชุด Command คำสั่งที่จะส่งไปยัง DMM

รูปแบบของคำสั่งสามารถกำหนดรูปแบบของคำสั่งได้โดยคลิกเมาส์ทางขวาที่ connector Format string ซึ่งอยู่ด้านบนตรงกลางของบล็อก Format Into String แล้วเลือก create» constant ในเมนูย่อย ซึ่งจะแสดงบล็อกของค่าคงที่ที่เชื่อมต่อกับ string format การกำหนดรูปแบบของ string สามารถทำได้โดย คลิกเมาส์ทางขวาที่บล็อกของ Format Into String แล้วเลือกคำสั่ง Edit Format String จะปรากฏไดอะล็อกบล็อกขึ้นมาสำหรับเพิ่ม ลด และแก้ไข รูปแบบของ string ซึ่งในไดอะล็อกบล็อกนี้มีจุดสำคัญอยู่ 3 จุด คือ Current format sequence, selected operation (example) และ corresponds format string (ดังแสดงในภาพ) ซึ่งในแต่ละจุดสามารถอธิบายได้ดังนี้

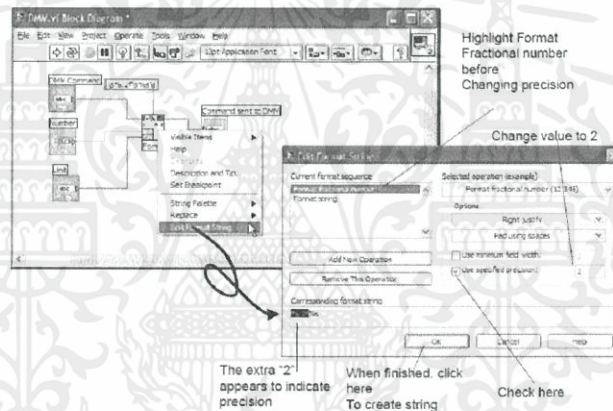
1. Current format sequence : แสดงจำนวนอินพุตของ Format Into String และระบุด้วยว่าแต่ละอินพุตรูปแบบของ string เป็นแบบใด
2. Selected operation (example) : จะบรรจุรูปแบบต่างๆ ของ string ซึ่งเราสามารถเลือกรูปแบบที่ต้องการได้โดยใช้เมาส์เลือกรูปแบบที่ต้องการ
3. Corresponds format string : จะแสดงสัญลักษณ์ของรูปแบบของ string เช่น %s %2f ซึ่งจะแสดงต่อกันตามจำนวนของอินพุต

หลังจากที่กำหนดค่าต่างๆ เสร็จเรียบร้อยแล้วให้กลับไปหน้าจอแผงพร้อมท์พาเนล ลองป้อนข้อความลงไปในบล็อกต่างๆ ดังตัวอย่างในรูปที่ 3.71 ซึ่งป้อนคำสั่ง set ลงในช่อง DMM Command ในบล็อก Number ให้ใส่ตัวเลขใดก็ได้ลงไป (ในที่นี้เป็นเลข 4) และในบล็อก Units ให้ใส่ข้อความเป็นตัวอักษร ในที่นี้จะใส่หน่วยของแรงดันซึ่งมีหน่วยเป็น โวลท์ (volt) หลังจากทีกรอกข้อมูลต่างๆ เสร็จเรียบร้อยแล้วขั้นตอนต่อไปคือการรับโปรแกรมโดยกดที่ปุ่มรัน สังเกตที่บล็อกแสดงผลข้อมูล ซึ่งจะเห็นว่าโปรแกรมจะทำการนำเอาข้อมูลอินพุตทั้ง 3 ตัวมาจัดการแพ็คข้อมูลเพื่อจะส่งไปยัง DMM โดยการจัดเรียงข้อมูลจะเริ่มจากการนำข้อมูลของบล็อก DMM Command ตามด้วยข้อมูลจากบล็อก Number และปิดท้ายด้วยข้อมูลจากบล็อก Unit ดังที่แสดงในรูปที่ 3.70

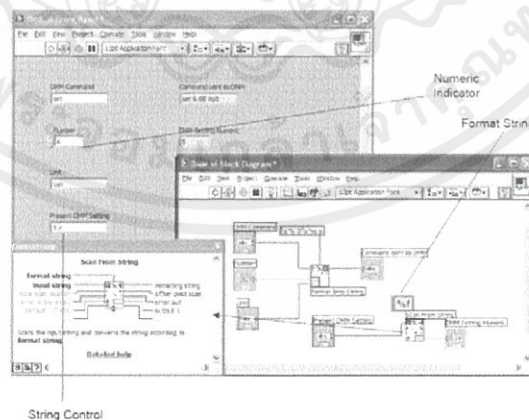


รูปที่ 3.71 โครงสร้างโปรแกรมที่มี Format into String เป็นส่วนประกอบ

ในตัวอย่างต่อไปต้องการที่เพิ่มฟังก์ชันการทำงานของโปรแกรมให้มากขึ้นโดยจะแสดงวิธีการเปลี่ยนจากคุณลักษณะของสัญลักษณ์ เช่น 0 ถึง 9, +, -, e, E เป็นต้น ให้เป็นตัวเลขดังแสดงในรูปที่ 3.71 วิธีการก็คือในหน้าต่างพร้อมท์พาเนลให้เพิ่มบล็อกของ string control และ string indicator จากนั้นกลับไปหน้าต่างบล็อกไดอะแกรมแล้วเลือกฟังก์ชันของบล็อก Scan Format String จาก Programming »String Palette ดังแสดงในรูปที่ 3.84 ก่อนที่จะเชื่อมต่อบล็อก Scan Format String เข้ากับ string control และ string indicator ให้ทำการดับเบิลคลิกที่บล็อก Scan Format String (หรือ คลิกเมาส์ขวาที่บล็อกแล้วเลือก Edit Format String จากเมนูย่อย) จะปรากฏไดอะล็อกล็อกขึ้นมา โดย ในบล็อกของ current scan sequence จะแสดงรูปแบบของสตริงเป็นแบบ Scan Number และบล็อก corresponds scan string จะแสดงสัญลักษณ์ %f จากนั้นให้กดปุ่ม OK จากนั้นต่อสายสัญญาณอินพุต เข้ากับ string control และต่อเอาท์พุตเข้ากับ string indicator จากนั้นทำการทดสอบโปรแกรม โดยการลองป้อนคำสั่งที่บล็อก Present DMM Setting ยกตัวอย่างเช่น 5 V จากนั้นทำการรันโปรแกรม สิ่งที่ได้ที่บล็อกแสดงผลจะแสดงเฉพาะเลข 5 เท่านั้น นั่นแสดงว่าเกิดจากการที่เราเลือกรูปแบบของ string เป็นแบบ Scan Number



รูปที่ 3.72 การกำหนดรูปแบบของสตริง



รูปที่ 3.73 โครงสร้างโปรแกรมที่มี Scan From String เป็นส่วนประกอบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

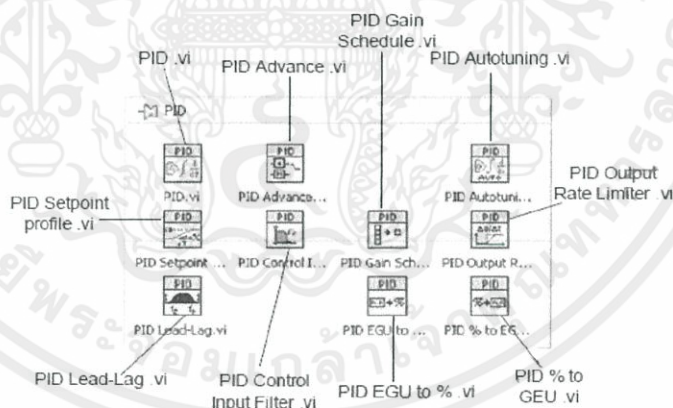
### 3.4.3.14 องค์ประกอบต่างๆใน Palette PID Control Toolkit

เมื่อทำการ Create New Project ขึ้นมาแล้วให้เข้าไปที่หน้า Block Diagram ทำการคลิก ขวามบนพื้นที่ว่างจะปรากฏ Menu Function ขึ้นมาจากนั้นเข้าไปที่ Icon Addons, Icon Control และเข้าไปที่ Icon PID ดังรูปที่ 3.74



รูปที่ 3.74 แสดงการเรียก Menu function ขึ้นมาใช้งาน

เมื่อคลิกเข้าไปที่ Icon PID แล้วจะมี Palette แสดง Icon ของ PID ที่เราต้องการใช้ขึ้นมาแสดงใน รูปที่ 3.75



รูปที่ 3.75 แสดง พาเนล ไอคอน ของ พีไอดี

ใน Palette PID ที่เราเปิดออกมาได้นั้นจะประกอบด้วย Function PID แบบต่างๆซึ่งจะมีความแตกต่างกันออกไป ดังต่อไปนี้

- PID .vi : ฟังก์ชัน PID เป็น PID อัลกอริทึม สำหรับการ ใช้ PID แบบที่ไม่ซับซ้อน หรือ การใช้ควบคุมแบบความเร็วสูงโดยต้องการประสิทธิภาพ คุณลักษณะเด่นของของ PID อัลกอริทึมควบคุมขอบเขตเอาต์พุตโดย Integrator anti-windup และ bumpless ตัวควบคุมเอาต์พุตสำหรับการเปลี่ยนแปลงค่าอัตราขยาย PID

- PID Advanced .vi : ฟังก์ชัน PID ที่สามารถปรับค่าคุณสมบัติได้ PID อัลกอริทึมนี้จะรวมคุณสมบัติของ PID.VI ไว้เหมือนกันและยังสามารถควบคุมด้วยมือ โดยลดการกระเพื่อมในระหว่างที่มีการเปลี่ยนจากการควบคุมด้วยมือไปเป็นการควบคุมแบบอัตโนมัติ, การกระทำที่ไม่เป็นเชิงเส้น, การควบคุมสอง องศา อิสระ, และการควบคุมค่าผิดพลาด squared
- PID Auto tuning .vi : PID แบบที่มีการรวมเครื่องมือในการหาค่าพารามิเตอร์ PID อัตโนมัติเข้าไปกับ PID อัลกอริทึมพื้นฐาน ฟังก์ชันนี้สามารถใช้ในที่เดียวกับ PID.VI โดยที่ใช้เพียงแค่ PID อัลกอริทึมพื้นฐาน การเพิ่ม อินพุต และเอาต์พุต ในฟังก์ชันนี้สามารถใช้ในกำหนดค่าพารามิเตอร์แบบอัตโนมัติ
- PID Control Input Filter .vi : เป็นการประยุกต์ใช้ ตัวกรองความถี่ต่ำ อันดับที่ ๓ กับค่าอินพุตที่เข้ามา ตัวกรองจะตัดความถี่ออกในช่วงที่ออกแบบไว้คือ หนึ่งในสิบของตัวอย่างความถี่อินพุตที่เข้ามา
- PID Gain Schedul .vi : สามารถเลือกปรับอัตราขยาย PID จากตารางอัตราขยายสำหรับควบคุมกระบวนการ โดยปรับค่าอัตราขยายที่แตกต่างสำหรับแต่ละช่วงการทำงานได้
- PID Lead Lag .vi : โดยปกติใช้เป็นตัวชดเชยในการควบคุมแบบล่วงหน้า : ซึ่งฟังก์ชันนี้จะทำการประมาณค่าที่ถูกต้อง นำหน้า/ล่าหลัง แบบยกกำลัง
- PID Output Rate Limit .vi : ฟังก์ชันนี้จะใช้กำหนดค่าสูงสุดของอัตราการเปลี่ยนแปลงการควบคุมเอาต์พุต โดยต่อฟังก์ชันนี้ที่ทางด้านเอาต์พุตของตัวควบคุม PID ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

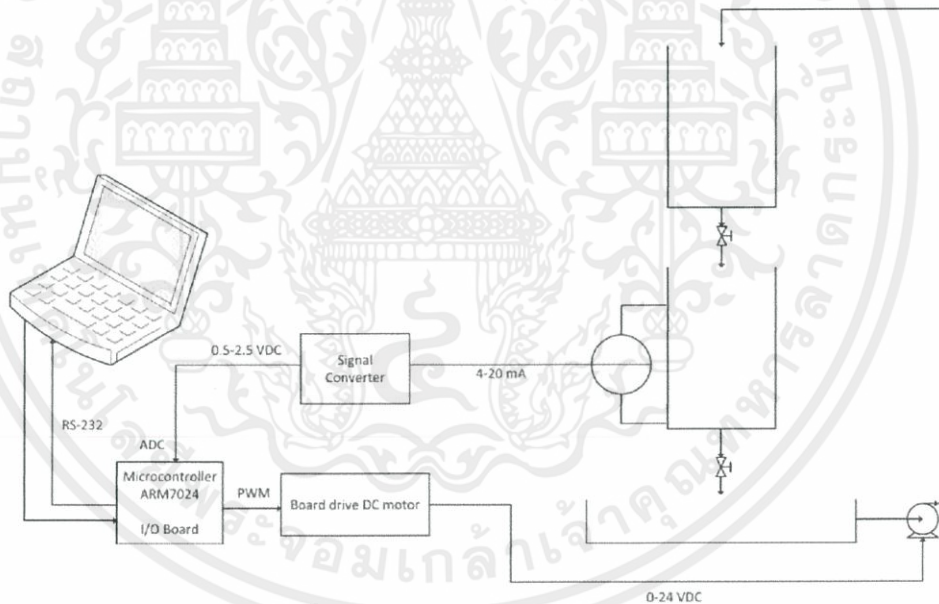
## บทที่ 4 วิธีการดำเนินงาน

### 4.1 กล่าวนำ

ในบทที่ 4 วิธีการดำเนินงาน เป็นการกล่าวถึงวิธีการทำงานของโครงการทั้งหมด เริ่มตั้งแต่การออกแบบโปรแกรมสำหรับบอร์ดไมโครคอนโทรลเลอร์ด้วยการเขียนโปรแกรมภาษาซี โดยใช้โปรแกรม Keil uVision 3 สำหรับการออกแบบโปรแกรมสำหรับควบคุมโดยใช้โปรแกรมนั้นจะใช้โปรแกรม LabVIEW 2011 นอกจากนี้ยังมีการหาฟังก์ชันถ่ายโอนของกระบวนการ การหาค่าพารามิเตอร์ที่เหมาะสมด้วยวิธีทางเดินรากล พร้อมทั้งทำการทดลองตามที่ได้คำนวณ หลังจากนั้นนำผลที่ได้จากการจำลองและผลที่ได้จากกระบวนการจริงมาเปรียบเทียบกัน

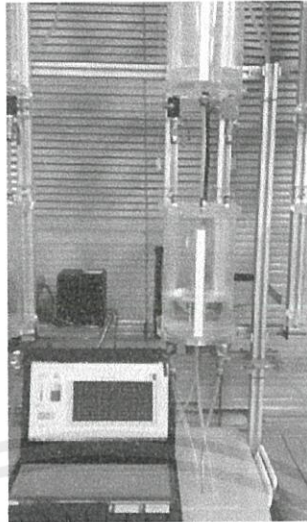
### 4.2 กระบวนการที่ใช้ในการทดลอง

กระบวนการที่ถูกทดลองนั้นเป็นกระบวนการควบคุมระดับน้ำสองถัง ที่ต่อแบบซีรียส์กัน และใช้อุปกรณ์ตามที่ได้กล่าวไว้ในข้างต้น นำมาเชื่อมต่อกันตามไดอะแกรมดังรูปที่ 4.1



รูปที่ 4.1 ไดอะแกรมของกระบวนการ

จากรูปที่ 4.1 กระบวนการทำงานนั้นจะเริ่มจากไมโครคอนโทรลเลอร์รับค่ากระบวนการจาก D/P Transmitter ที่ผ่านตัวแปลงสัญญาณแล้ว และไมโครคอนโทรลเลอร์จะส่งค่ากระบวนการที่ได้ผ่านพอร์ตอนุกรม (RS232) ขึ้นไปยังคอมพิวเตอร์ที่ทำหน้าที่เป็นตัวควบคุม หลังจากคอมพิวเตอร์ทำการประมวลผลแล้ว จะทำการส่งค่าควบคุมออกมายังไมโครคอนโทรลเลอร์ เพื่อให้ไมโครคอนโทรลเลอร์สร้างสัญญาณ Pulse width modulation (PWM) โดยใช้ความถี่ 800 Hz และส่งให้กับบอร์ดขับเคลื่อนมอเตอร์ เพื่อปั้มน้ำต่อไป



รูปที่ 4.2 กระบวนการสำหรับการทดลอง

### 4.3 การออกแบบโปรแกรมสำหรับบอร์ดไมโครคอนโทรลเลอร์ รุ่น ET-BASE ARM7024 สำหรับกระบวนการ

ในการออกแบบโปรแกรมสำหรับบอร์ดไมโครคอนโทรลเลอร์ รุ่น ET-BASE ARM7024 นั้น จะถูกทำให้เป็นไปในลักษณะของรูปแบบของการรับค่าและส่งค่าตามที่ต้องการระหว่างกระบวนการและคอมพิวเตอร์ที่เป็นตัวควบคุม โดยผ่านพอร์ตอนุกรม (RS232) ในส่วนของโปรแกรมที่ใช้ในการเขียนโปรแกรมภาษาซี คือ Keil uVision 3 และในส่วนของโปรแกรมนั้นจะแสดงไว้ในส่วนของภาคผนวก

```

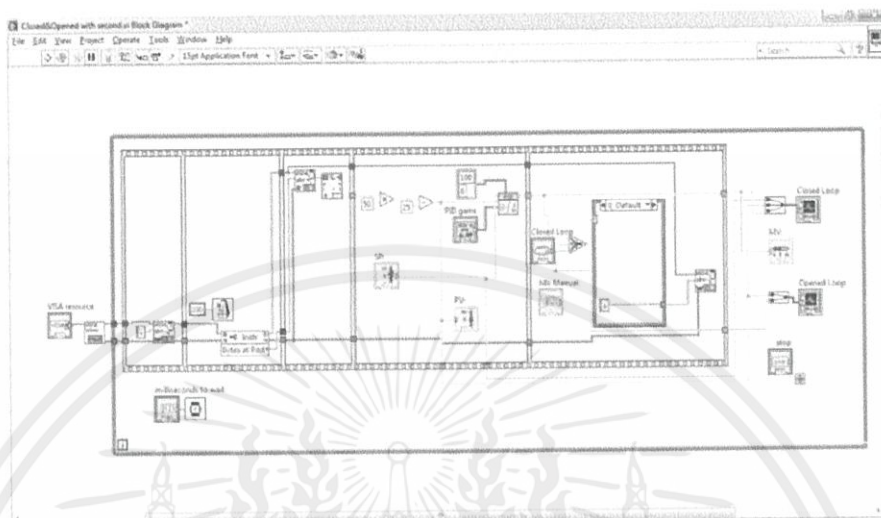
001 #include <stm32f10x.h> //stm32f10x.h
002 #include <stdio.h> //stdio.h
003 #include <stdlib.h> //stdlib.h
004 #define FREQ_CK48MHZ 48000 //FREQ_CK48MHZ
005 #define FREQ_CALCULATE 4800 //FREQ_CALCULATE
006
007 /* prototype section */
008 void delay_management(long int); //delay_management
009 int putchar(int ch); //putchar
010 int getchar(void); //getchar
011 void SerialDataToUART(int uDataByte); //SerialDataToUART
012
013 int main(void)
014 {
015     unsigned char rData; //ADC Result (uData)
016     unsigned int val; //ADC Result (uData)
017     float vData; //ADC Result (uData)
018
019     GPIOCON = 0x00000000; //Reset P1.0 & P1.0 Pin Function
020     //GPIOCON = 0x00000000; //Reset P1.0 & P1.0 = ADC
021
022     // Enable UART // Enable UART
023     CONTCFG = 0x00; //Setting UART
024     CONTCFG = 0x00; //Setting UART and CONTCFG to 0x00 calculated
025     //CONTCFG = 0x00;
026 }
    
```

รูปที่ 4.3 โปรแกรม Keil uVision 3 ที่ใช้เขียนภาษาซี

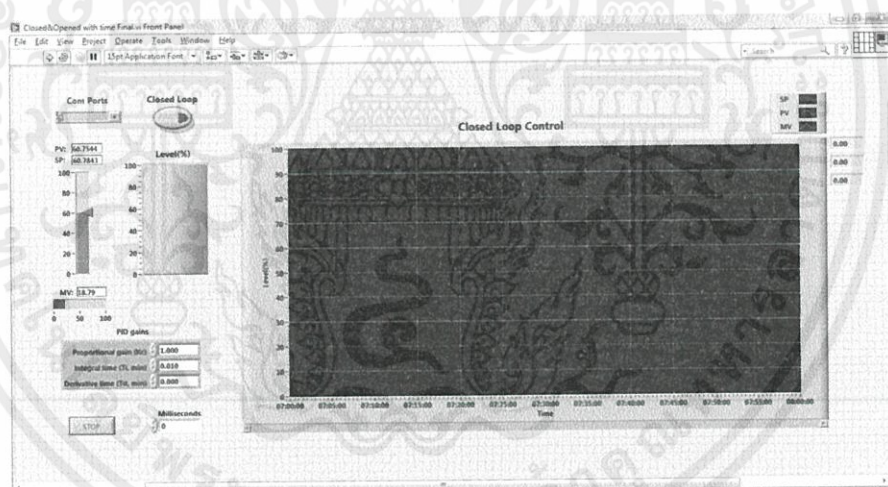
### 4.4 การออกแบบโปรแกรมสำหรับควบคุม

ในการออกแบบโปรแกรมสำหรับควบคุมนั้น จะใช้โปรแกรม LABVIEW 2011 ในการเขียนโปรแกรมจะเป็นในลักษณะของภาษากาฟิก โดยจะกระทำการเขียนโปรแกรมให้สอดคล้องกับโปรแกรมที่อยู่บนไมโครคอนโทรลเลอร์ ที่มีการรับส่งข้อมูลผ่านพอร์ตอนุกรม (RS232) และเขียน

โปรแกรมให้มีลักษณะของการควบคุมโดยจะใช้การควบคุมแบบพีไอดี ในส่วนของบล็อกไดอะแกรมของโปรแกรมจะเป็นไปตามที่แสดงดังรูปที่ 4.4 และหน้าต่างโปรแกรมจะเป็นไปตามที่แสดงดังรูปที่ 4.5



รูปที่ 4.4 บล็อกไดอะแกรมของโปรแกรมส่วนที่ใช้ควบคุม



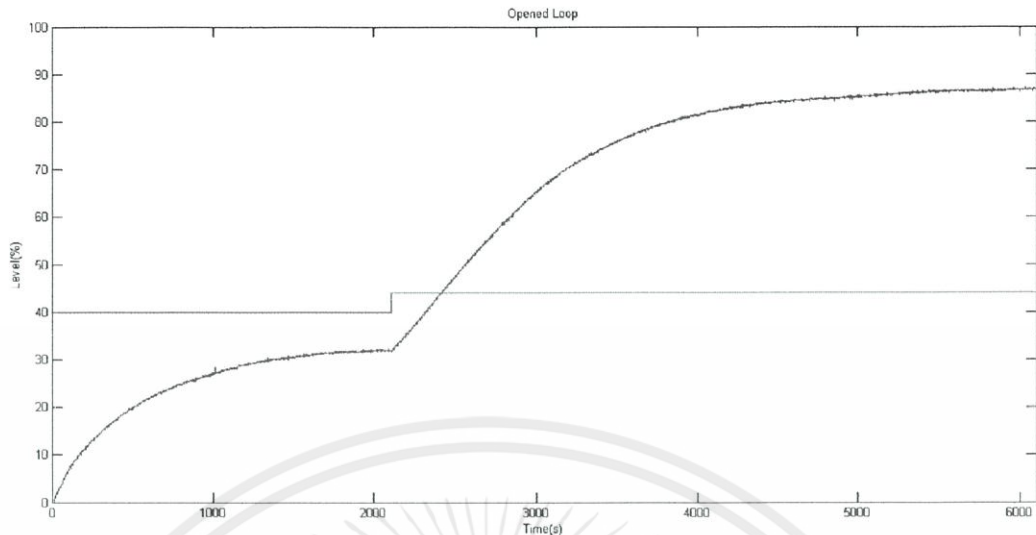
รูปที่ 4.5 โปรแกรมส่วนที่ใช้ควบคุม

#### 4.5 การหาแบบจำลองทางคณิตศาสตร์ของกระบวนการที่ถูกควบคุม

ในการหาค่าพารามิเตอร์พีไอดีนั้น จำเป็นที่จะต้องหาแบบจำลองทางคณิตศาสตร์ของกระบวนการเสียก่อน โดยจะหาแบบจำลองทางคณิตศาสตร์ด้วยวิธี Process Reaction Curve Method (Open Loop Method) จากผลตอบสนองของกระบวนการแบบเปิดที่ถูกควบคุมด้วยสัญญาณอินพุตแบบขั้นบันได โดยที่กระบวนการในกรณีนี้จะไม่มีโพล ที่จุด ออริจิน หรือไม่มี Dominant complex conjugate poles ดังนั้นผลตอบสนองของเวลาจะได้ในลักษณะเป็นรูปตัว S ถ้า

ไม่ว่ากรณีใด โดยฟังก์ชันถ่ายโอนของกระบวนการสามารถหาได้จากสมการที่ (4-1) สารทุกครั้งที่มีการนำไปใช้

$$G_p = \frac{K_{ss}e^{-T_0s}}{\tau s + 1} \quad (4-1)$$



รูปที่ 4.6 ผลตอบสนองของกระบวนการจากสัญญาณแบบขั้นบันได (Step response)

จากผลตอบสนองของกระบวนการ สามารถจำลองฟังก์ชันถ่ายโอนของกระบวนการได้ด้วยการหาค่าของ  $K$ ,  $\tau$  และ  $T_0$  โดยคำนวณจากการประมาณค่าได้ดังนี้

เวลา  $t_1$  คือ เวลาที่เออร์พุดมีค่า 0.238 เท่าของค่าเออร์พุดที่หยุดนิ่ง

เวลา  $t_2$  คือ เวลาที่เออร์พุดมีค่า 0.632 เท่าของค่าเออร์พุดที่หยุดนิ่ง

$$\text{โดยที่ } K = \frac{\Delta PV \%}{\Delta MV \%}, \tau = \frac{3}{2}(t_2 - t_1) \text{ และ } T_0 = t_2 - \tau$$

$$\text{และ } t_1 = 2440 - 2105 = 335 \text{ s}$$

$$t_2 = 3050 - 2105 = 945 \text{ s}$$

$$\text{จะได้ } K_{ss} = 13.7325$$

$$\tau = 915 \text{ s}$$

$$T_0 = 30 \text{ s}$$

ฟังก์ชันถ่ายโอนของกระบวนการคือ  $G_p = \frac{13.73e^{-30s}}{915s+1}$  ถัดมาจะทำการประมาณค่าในเทอมของ  $e^{-30s}$  ซึ่งใช้การประมาณของ PADE Approximation จากนั้นเมื่อได้ทำการประมาณค่าแล้วจะได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งาน  $e^{-30s} = \frac{-s + 0.0667}{-s - 0.0667}$  กรุณาให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เพราะฉะนั้นจะได้ฟังก์ชันถ่ายโอนของกระบวนการคือ

$$G_p = \frac{-13.73s + 0.9159}{915s^2 + 62.0305s + 0.0667}$$

#### 4.6 การหาพารามิเตอร์พีโอดีด้วยวิธี Root-Locus Technique

สำหรับการหาพารามิเตอร์พีโอดีด้วยวิธี Root-Locus Technique นั้นจะใช้เครื่องมือที่เป็น Tool ของ MATLAB ที่มีชื่อว่า SISO Tool และจากฟังก์ชันถ่ายโอนของกระบวนการ

$$G_p = \frac{-13.73s + 0.9159}{915s^2 + 62.0305s + 0.0667}$$

เมื่อนำมาทำการวิเคราะห์โดยใช้โปรแกรม MATLAB จะได้ทางเดินรากแสดงดังรูปที่ 4.7

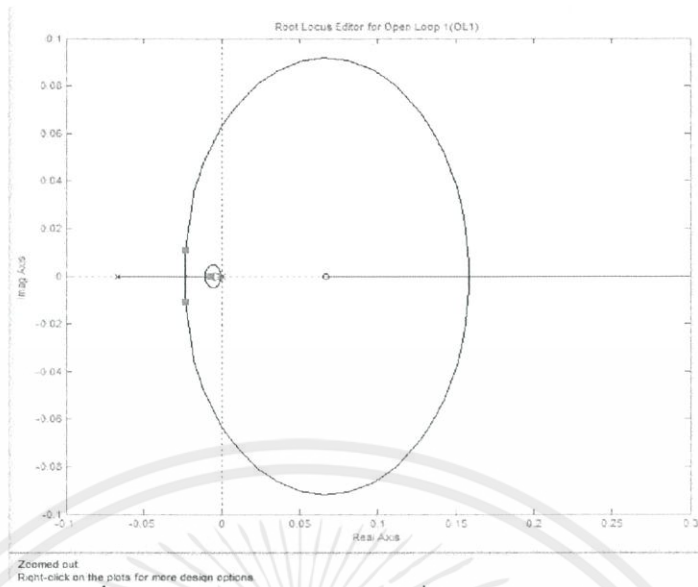


รูปที่ 4.7 ทางเดินรากขณะที่ยังไม่มีตัวควบคุม

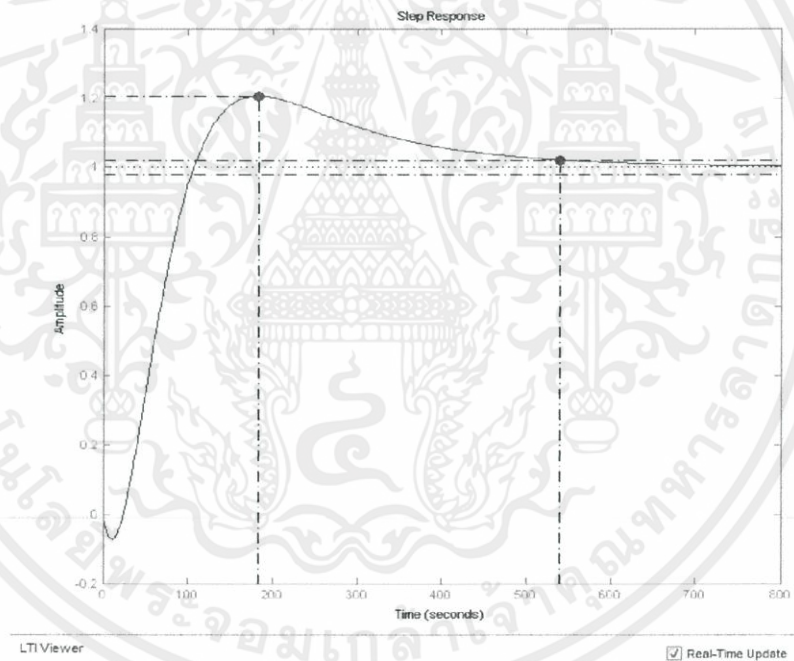
เมื่อทำการใส่ตัวควบคุมแบบพีโอเข้าไปในระบบจะพบว่าต้องทำการเดิม โพล และ ซีโร่ เข้าไป โดยตำแหน่งของ โพล จะอยู่ที่จุดกำเนิด (0,0) และทำการปรับตำแหน่ง ซีโร่ ให้เหมาะสม จากนั้น จะทำการปรับโพลของระบบเพื่อให้เป็นไปตามต้องการ

โดยจะกำหนดระบบให้มี Overshoot = 20.6% และ Settling time = 540 วินาที จะได้ทางเดินรากตามที่แสดงดังรูปที่ 4.8 และผลตอบสนองต่ออินพุตแบบขั้นบันไดตามที่แสดงดังรูปที่ 4.9

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



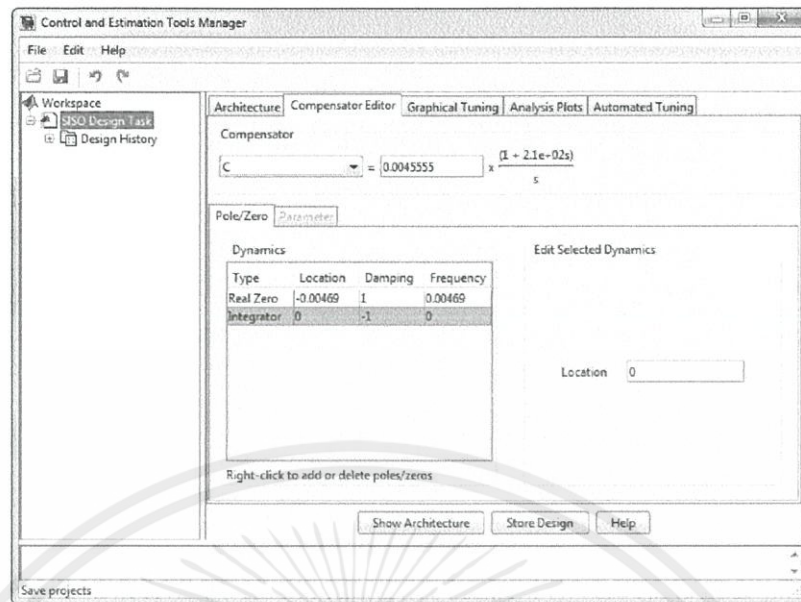
รูปที่ 4.8 ทางเดินรากของระบบที่มีตัวควบคุมแบบพีโอ



รูปที่ 4.9 ผลตอบสนองของระบบที่มีตัวควบคุมแบบพีโอ

และจาก SISO Tool จะทำให้ทราบค่าของพีโอ และสามารถทราบค่า  $C = 0.0045555 \times \frac{1-210s}{s}$  ซึ่งก็คือค่าพีโอ โดยจะแสดงดังรูปที่ 4.10

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



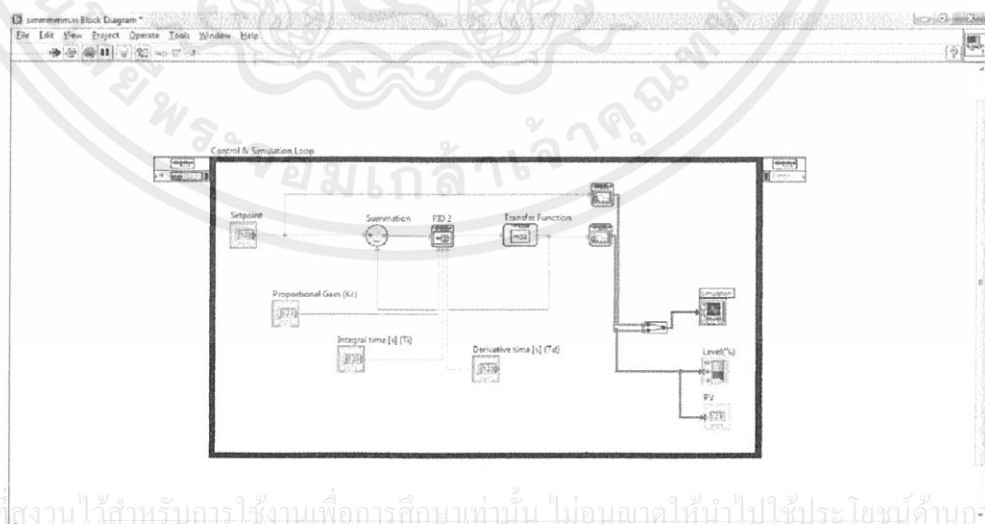
รูปที่ 4.10 แสดงค่าตัวควบคุมแบบพีโอที่ได้จากโปรแกรม

จากนั้นนำมาเทียบสัมประสิทธิ์ของโครงสร้างพีโอ ซึ่งก็คือ  $K_p + \frac{K_p}{T_i s}$

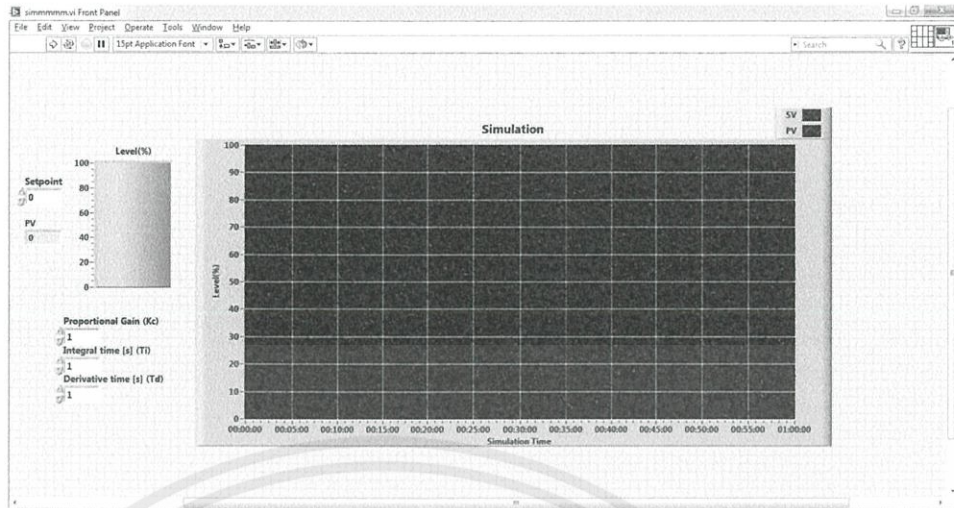
เพราะฉะนั้นจะได้ค่า  $K_p = 0.95665$  และค่า  $T_i = 210$  วินาที หรือ 3.5 นาที

#### 4.7 การออกแบบโปรแกรมสำหรับการจำลองกระบวนการ

ในส่วนของการออกแบบการจำลองกระบวนการนั้น จะใช้โปรแกรม LabVIEW ในการจำลองกระบวนการเพื่อดูผลตอบสนองของระบบ โดยจะใช้ฟังก์ชันถ่ายโอนของกระบวนการที่ได้คำนวณในข้างต้น ในส่วนของบล็อกไดอะแกรมของโปรแกรมจะเป็นไปตามที่แสดงดังรูปที่ 4.11 และโปรแกรมสำหรับจำลองกระบวนการเป็นไปตามที่แสดงดังรูปที่ 4.12



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านธุรกิจ  
รูปที่ 4.11 บล็อกไดอะแกรมโปรแกรมสำหรับจำลองกระบวนการ

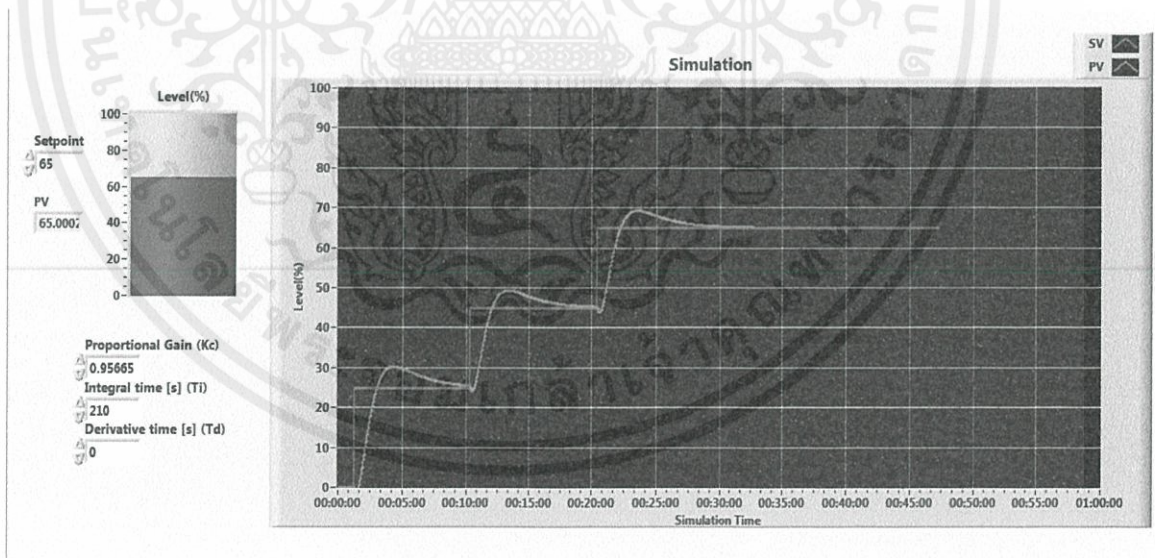


รูปที่ 4.12 โปรแกรมสำหรับจำลองกระบวนการ

#### 4.8 การทดลองการควบคุมโดยใช้ตัวควบคุมแบบพีไอ

ในส่วนของการทดลองนั้นจะแบ่งออกเป็น 2 ส่วนคือ การทดลองที่ได้จากการจำลองและการทดลองที่ได้จากกระบวนการจริง

เมื่อหาค่า  $K_p$  และ  $T_i$  ในข้างต้นได้แล้ว จะนำไปควบคุมกับกระบวนการที่ถูกจำลอง โดยเปลี่ยนค่า Set point จาก 25%, 45% และ 65% ตามลำดับและทำการสังเกตผลที่ได้จากการคำนวณในโปรแกรม MATLAB จะได้ผลดังรูปที่ 4.13

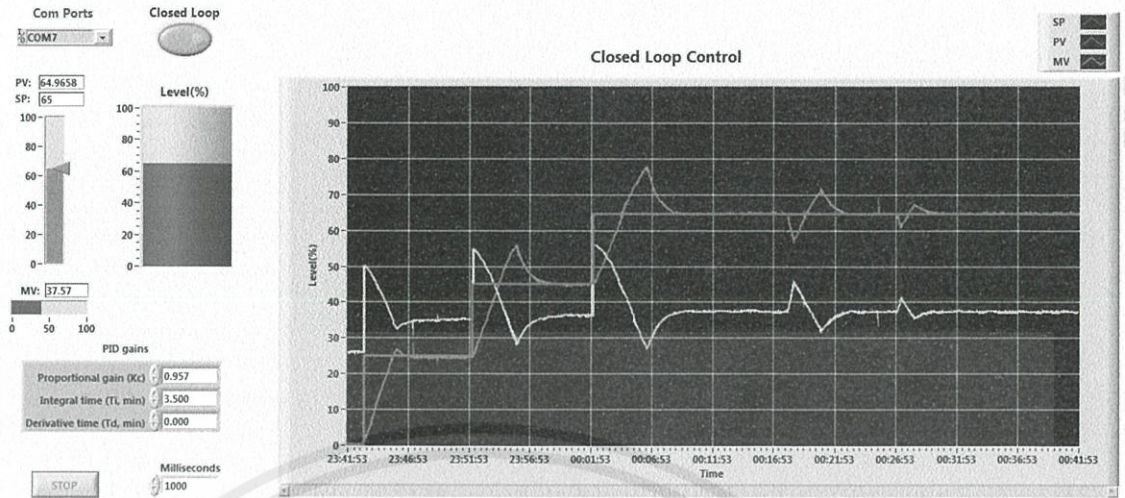


รูปที่ 4.13 ผลที่ได้จากการจำลอง

ผลตอบสนองทางด้านเอาต์พุตของกระบวนการ เป็นดังนี้

เอกสารนี้เป็นเอกสารที่จัดทำไว้สำหรับนักศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น - Settling Time = 540 วินาที และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำหรับการทดลองกับกระบวนการจริงนั้นจะกระทำเช่นเดียวกันกับการจำลองซึ่งก็คือการเปลี่ยนค่า Set point จาก 25%, 45% และ 65% ตามลำดับ จะได้ผลแสดงดังรูปที่ 4.14



รูปที่ 4.14 ผลที่ได้จากกระบวนการจริง

ผลตอบสนองทางด้านเอาต์พุตของกระบวนการ เป็นดังนี้

- Maximum Overshoot = 60%
- Settling Time = 489 วินาที

นอกจากนี้เมื่อทำการปรับระบบโดยกระทำการปิดวาล์วของถังด้านบน และเมื่อปรับกระบวนการเข้าสู่เช่นเดิมระบบก็สามารถควบคุมให้ค่ากระบวนการนั้นอยู่ตามค่าเป้าหมายได้ดังเช่นเดิม

#### 4.9 สรุปผล

สำหรับในส่วนของบทที่ 4 นั้น จะเป็นการดำเนินการออกแบบกระบวนการ และทดลองควบคุมกระบวนการโดยจะใช้ตัวควบคุมที่เป็นแบบพีไอ ซึ่งผลที่ได้จากการทดลองในส่วนของการจำลองจะเป็นไปตามที่ได้จากการวิเคราะห์ด้วยโปรแกรม MATLAB ที่มีเครื่องมือที่ชื่อว่า SISO Tool และสำหรับการทดลองกับกระบวนการจริงนั้น พบว่าค่า Percent Overshoot ที่ได้นั้นมีค่าสูงกว่าผลที่ได้จากการจำลอง แต่ค่า Settling Time ที่ได้พบว่ามีค่าน้อยกว่าผลที่ได้จากการจำลอง ซึ่งอาจเป็นผลมาจากการสภาพแวดล้อม และปัจจัยภายนอกที่แตกต่างกันขณะที่ทำการหาฟังก์ชันถ่ายโอนของระบบและระหว่างการทดลอง นอกจากนี้เมื่อระบบได้เข้าสู่สมดุลก็สามารถทำการควบคุมเพื่อให้ค่ากระบวนการเป็นไปตามค่าเป้าหมายที่ต้องการได้ รวมทั้งเมื่อได้ทำการปรับระบบโดยกระทำการปิดวาล์วของถังบน และปรับกระบวนการให้เข้าสู่สภาวะเช่นเดิม ระบบยังสามารถควบคุมให้ค่ากระบวนการนั้นอยู่ตามค่าเป้าหมายได้ดังเช่นเดิม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## สรุปผลการวิจัย และข้อเสนอเชิงทฤษฎีจากการวิจัย

### 5.1 สรุปผลการทดลอง

จากการศึกษาและการทดลองควบคุมระดับน้ำแบบสองถังในลักษณะที่ต่อแบบซีรียกัน จากกระบวนการทำงานนั้นจะเริ่มจาก ไมโครคอนโทรลเลอร์ที่ถูกเขียนโปรแกรมให้เป็นไปในลักษณะของบอร์ดรับค่าและส่งค่าระหว่างกระบวนการกับคอมพิวเตอร์ที่เป็นตัวควบคุมรับค่า โดย ไมโครคอนโทรลเลอร์นั้นจะรับค่ากระบวนการจาก D/P Transmitter ที่ผ่านตัวแปลงสัญญาณแล้ว และไมโครคอนโทรลเลอร์นั้นจะส่งค่ากระบวนการที่ได้ ผ่านพอร์ตอนุกรม (RS232) ขึ้นไปยังคอมพิวเตอร์ที่ทำหน้าที่เป็นตัวควบคุม เพื่อให้คอมพิวเตอร์ทำการประมวลผล จากนั้นจะส่งค่าควบคุมกลับยังไมโครคอนโทรลเลอร์ เพื่อให้ไมโครคอนโทรลเลอร์สร้างสัญญาณ Pulse width modulation (PWM) โดยการเปลี่ยนค่าดิวตี้ไซเคิล และส่งต่อให้กับบอร์ดขับเคลื่อนมอเตอร์เพื่อทำการปั้มน้ำถัดไป เมื่อได้ตัวควบคุมและมีกระบวนการสมบูรณ์แล้ว ต่อมาก็ทำการควบคุมโดยใช้การควบคุมแบบพีไอดี สำหรับการหาค่าพีไอดีที่เหมาะสมจะใช้วิธีทางเดินราก เพื่อใช้ดูทางเดินรากและค่าพีไอดีที่เหมาะสมเพื่อให้ผลนั้นเป็นไปตามที่ต้องการ สำหรับผลที่ได้จากการควบคุมและผลที่ได้จากการจำลองนั้น พบว่ากระบวนการจริงมีค่า Overshoot สูงกว่าแต่ Settling time นั้นน้อยกว่าการจำลอง

### 5.2 ปัญหาและอุปสรรค

ปัญหาที่พบในระหว่างการทดลองนั้น พบว่าเกิดขึ้นหลายปัญหาด้วยกันคือ วาล์วควบคุมอัตราการไหลเกิดการอุดตันเมื่อใช้ไปนานๆ ซึ่งส่งผลกระทบต่อการทำงานโดยจะทำให้ฟังก์ชันถ่ายโอนของกระบวนการนั้นเปลี่ยนไป และทำให้ผลที่ได้จากการทดลองนั้นไม่เป็นไปตามที่จำลอง สำหรับวิธีการแก้ปัญหาคือ ทำการเปลี่ยนน้ำที่ใช้ในการทดลองรวมทั้งกำจัดสิ่งอุดตันที่อยู่ในวาล์วควบคุม นอกจากนี้ยังทราบสาเหตุที่มาของสิ่งอุดตันซึ่งก็คือ แสงแดดที่ส่องผ่านเข้ามาพร้อมกับออกซิเจนทำให้เกิดตะไคร่น้ำขึ้นนั่นเอง

ปั้มน้ำที่ใช้ในการทดลองนั้นเมื่อทดลองระยะเวลาานจะเกิดความร้อนสะสมที่ตัวปั้ม ซึ่งจะส่งผลกระทบต่อประสิทธิภาพในการทดลอง นอกจากนี้ยังเกิดจากความฝืดของตัวปั้มซึ่งจะส่งผลต่ออัตราการไหล สำหรับวิธีการแก้ปัญหาคือ ทดลองในสภาพแวดล้อมที่มีอุณหภูมิเหมาะสมต่อการทำงานของปั้ม

เวลาที่ทำการทดลองนั้น พบว่าทดลองในสภาพแวดล้อมที่ต่างกันจะทำให้ผลที่ได้จากการทดลองนั้นแตกต่างกันไป เนื่องจากปัจจัยภายนอก อาทิเช่น อุณหภูมิ การรบกวนของสัญญาณแม่เหล็ก ซึ่งส่งผลโดยตรงกับไมโครคอนโทรลเลอร์ สำหรับวิธีการแก้ปัญหาคือ พยายามควบคุมปัจจัยภายนอก สภาพแวดล้อมต่างๆสำหรับการทดลองให้มีความเหมือนกันในแต่ละครั้งของการทดลอง เพื่อให้ผลที่ได้เป็นไปตามที่ต้องการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 5.3 ข้อเสนอแนะ

สำหรับวิธีการดำเนินการและโปรแกรมที่ถูกเขียนนั้นสามารถนำไปประยุกต์ใช้กับระบบที่มีหนึ่งอินพุตหนึ่งเอาต์พุต โดยไม่จำเป็นว่าจะต้องนำไปควบคุมระดับเพียงอย่างเดียวแต่ยังสามารถนำไปควบคุมกับระบบอื่นๆที่เป็นหนึ่งอินพุตหนึ่งเอาต์พุต

ส่วนการดำเนินการทดลองนั้นควรทำที่สภาวะแวดล้อมเดียวกันหรือภายในวันเดียว ที่เดียวไปเลย เพื่อขจัดปัญหาการรบกวนระบบจากปัจจัยอื่นๆภายนอก เช่น อุณหภูมิ ช่วงเวลา เป็นต้น



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บรรณานุกรม

ธีรวัฒน์ เทพมณี, วิริยะ กองรัตน์ และคณะ. (2547). *ปฏิบัติการวิศวกรรมการวัดคุม 3*. พิมพ์ครั้งที่ 1. กรุงเทพฯ: แผนกตำรา คณะ วิศวกรรมศาสตร์ สจล.

รศ.สุมาลี อุณหวนิชย์. (2545). *ระบบควบคุม(Control System)*. กรุงเทพฯ: ว.เพ็ชรสกุล.

รศ.อาจินต์ น่วมสำราญ, ผศ.ทรงชัย วีระทวีมาศ. (2549). *ปฏิบัติการวิศวกรรมการวัดคุม 2*. พิมพ์ครั้งที่ 1. กรุงเทพฯ: แผนกตำรา คณะ วิศวกรรมศาสตร์ สจล.

โอภาส ศิริครรชิตถาวร. (2548). *เรียนรู้ และพัฒนาไมโครคอนโทรลเลอร์ ARM7 ด้วยภาษาซี*. กรุงเทพฯ: กราฟิกเมลอน.

คู่มือ ET-BASE ARM7024. ค้นเมื่อ 9 กรกฎาคม 2555, จาก <http://www.etteam.com/>

ทฤษฎีและรูปภาพประกอบ ระบบควบคุมพีไอดี. ค้นเมื่อ 28 กันยายน 2555, จาก <http://th.wikipedia.org/>

ระบบควบคุม. ค้นเมื่อ 1 ตุลาคม 2555, จาก [fivedots.coe.psu.ac.th/](http://fivedots.coe.psu.ac.th/)

Advance Virtual Instrument. ค้นเมื่อ 1 พฤศจิกายน 2555, จาก <http://arjin.info/>

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## ภาคผนวก

ตัวอย่างโค้ดโปรแกรมสำหรับ ARM 7024

```
#include <ADUc7024.H> // ADUc7024 MPU Register
#include <stdio.h>
#include <stdlib.h>
#define PWM_DAT0VALUE 0x0A35 /* set PWM freq to 800 Hz */
#define PWM_DAT1VALUE 0x00 /* set dead time to 1us */
/* pototype section */
void delay(unsigned long int); // Delay Time Function
int putchar(int ch); // Write Data to Serial
int getchar(void); // Read Data From Serial
void SetPwmDutyCycleCH0(int ucDutyCycle);
int main(void)
{
    unsigned char rs232;
    unsigned int val;
    float volt;
    GP1CON &= 0xFFFFFAC;
    GP1CON |= 0x00000001;
// Initial UART = 9600BPS
COMCON0 = 0x80;
COMDIV0 = 0x88;
COMDIV1 = 0x00;
COMCON0 = 0x07;

// Power-ON ADC
ADCCON = 0x00000000;
ADCCON |= 0x00000020;
delay(1000);
ADCCON |= 0x00000300;
ADCCON &= 0xFFFFFE7;
ADCCON |= 0x00000004;
REFCON = 0x00000001;
ADCCON |= 0x00000080;

// Config PWM
GP3CON = 0x00001111;
PWMCON = 0xF0000000;
PWMDAT0 = 0x0A35;
PWMDAT1 = 0x00;
PWMEN = 0x00000000;
```

เอกสารนี้เป็นเอกสารสงวนลิขสิทธิ์สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ กรุณาแจ้งผู้จัดทำต้นฉบับเพื่อปรับปรุงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

while (1)
{
    rs232 = getchar();
    switch (rs232)
    {
        case 'A' :
            ADCCP = 0;
            delay(1000);
            while (!ADCSTA){};
            val = (ADCDAT >> 16)& 0x00000FFF;
            volt = val*2.50/4096.0;
            printf("%f",volt);
            delay(1000);
            break;
        case 'a' :
            SetPwmDutyCycleCH0(0);
            break;
        case 'b' :
            SetPwmDutyCycleCH0(1);
            break;
        case 'c' :
            SetPwmDutyCycleCH0(2);
            break;
        case 'd' :
            SetPwmDutyCycleCH0(3);
            break;
        case 'e' :
            SetPwmDutyCycleCH0(4);
            break;
        case 'f' :
            SetPwmDutyCycleCH0(5);
            break;
        case 'g' :
            SetPwmDutyCycleCH0(6);
            break;
        case 'h' :
            SetPwmDutyCycleCH0(7);
            break;
        case 'i' :
            SetPwmDutyCycleCH0(8);
            break;
        case 'j' :
            SetPwmDutyCycleCH0(9);
            break;
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งยังมีให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

case 'k' :
    SetPwmDutyCycleCH0(10);
    break;
case 'l' :
    SetPwmDutyCycleCH0(11);
    break;
case 'm' :
    SetPwmDutyCycleCH0(12);
    break;
case 'n' :
    SetPwmDutyCycleCH0(13);
    break;
case 'o' :
    SetPwmDutyCycleCH0(14);
    break;
case 'p' :
    SetPwmDutyCycleCH0(15);
    break;
case 'q' :
    SetPwmDutyCycleCH0(16);
    break;
case 'r' :
    SetPwmDutyCycleCH0(17);
    break;
case 's' :
    SetPwmDutyCycleCH0(18);
    break;
case 't' :
    SetPwmDutyCycleCH0(19);
    break;
case 'u' :
    SetPwmDutyCycleCH0(20);
    break;
case 'v' :
    SetPwmDutyCycleCH0(21);
    break;
case 'w' :
    SetPwmDutyCycleCH0(22);
    break;
case 'x' :
    SetPwmDutyCycleCH0(23);
    break;
case 'y' :
    SetPwmDutyCycleCH0(24);

```

เอกสารนี้เป็นเอกสารที่...  
 เอกสารนี้เป็นเอกสารที่...  
 ไม่ว่ากรณีใดๆทั้งสิ้น

ไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งยังมีให้ดาวน์โหลดฟรีที่... ต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



```

        SetPwmDutyCycleCH0(39);
        break;
    case 'O' :
        SetPwmDutyCycleCH0(40);
        break;
    case 'P' :
        SetPwmDutyCycleCH0(41);
        break;
    case 'Q' :
        SetPwmDutyCycleCH0(42);
        break;
    case 'R' :
        SetPwmDutyCycleCH0(43);
        break;
    case 'S' :
        SetPwmDutyCycleCH0(44);
        break;
    case 'T' :
        SetPwmDutyCycleCH0(45);
        break;
    case 'U' :
        SetPwmDutyCycleCH0(46);
        break;
    case 'V' :
        SetPwmDutyCycleCH0(47);
        break;
    case 'W' :
        SetPwmDutyCycleCH0(48);
        break;
    case 'X' :
        SetPwmDutyCycleCH0(49);
        break;
    case 'Y' :
        SetPwmDutyCycleCH0(50);
        break;
    case 'Z' :
        SetPwmDutyCycleCH0(51);
        break;
    case '0' :
        SetPwmDutyCycleCH0(52);
        break;
    case '1' :
        SetPwmDutyCycleCH0(53);
        break;

```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งยังมีให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

case '2' :
    SetPwmDutyCycleCH0(54);
    break;
case '3' :
    SetPwmDutyCycleCH0(55);
    break;
case '4' :
    SetPwmDutyCycleCH0(56);
    break;
case '5' :
    SetPwmDutyCycleCH0(57);
    break;
case '6' :
    SetPwmDutyCycleCH0(58);
    break;
case '7' :
    SetPwmDutyCycleCH0(59);
    break;
case '8' :
    SetPwmDutyCycleCH0(60);
    break;
case '9' :
    SetPwmDutyCycleCH0(61);
    break;
case '!' :
    SetPwmDutyCycleCH0(62);
    break;
case '@' :
    SetPwmDutyCycleCH0(63);
    break;
case '#' :
    SetPwmDutyCycleCH0(64);
    break;
case '$' :
    SetPwmDutyCycleCH0(65);
    break;
case '%' :
    SetPwmDutyCycleCH0(66);
    break;
case '^' :
    SetPwmDutyCycleCH0(67);
    break;
case '&' :
    SetPwmDutyCycleCH0(68);

```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



```

        SetPwmDutyCycleCH0(83);
        break;
    case ';':
        SetPwmDutyCycleCH0(84);
        break;
    case '!':
        SetPwmDutyCycleCH0(85);
        break;
    case '<':
        SetPwmDutyCycleCH0(86);
        break;
    case '>':
        SetPwmDutyCycleCH0(87);
        break;
    case ',':
        SetPwmDutyCycleCH0(88);
        break;
    case ':':
        SetPwmDutyCycleCH0(89);
        break;
    case '?':
        SetPwmDutyCycleCH0(90);
        break;
    case 'ä':
        SetPwmDutyCycleCH0(91);
        break;
    case '~':
        SetPwmDutyCycleCH0(92);
        break;
    case '¿':
        SetPwmDutyCycleCH0(93);
        break;
    case 'Ó':
        SetPwmDutyCycleCH0(94);
        break;
    case '¡':
        SetPwmDutyCycleCH0(95);
        break;
    case 'í':
        SetPwmDutyCycleCH0(96);
        break;
    case 'à':
        SetPwmDutyCycleCH0(97);
        break;

```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง วิชาศึกษาค้นคว้าด้วยตนเอง (S-Learning) ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งยังมีให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    case 'é' :
        SetPwmDutyCycleCH0(98);
        break;
    case 'è' :
        SetPwmDutyCycleCH0(99);
        break;
    case 'Ò' :
        SetPwmDutyCycleCH0(100);
        break;
    default:
        SetPwmDutyCycleCH0(0);
        break;
}
}
}
void delay(unsigned long int count1)
{while(count1 > 0) {count1--;}
}

/*****/
/* Write Character To Serial */
/*****/
int putchar(int ch)
{
    if (ch == '\n')
    {
        while(!(0x40==(COMSTA0 & 0x40)))
        { }
        COMTX = 0x0D;
    }
    while(!(0x40==(COMSTA0 & 0x40)))
    { }
    return (COMTX = ch);
}

/*****/
/* Read Character From Serial */
/*****/
int getchar (void)
{
    while(!(0x01==(COMSTA0 & 0x01)))
    { }
    return (COMRX);
}
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น ยกเว้นกรณีเกิดแบบสงวนเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
void SetPwmDutyCycleCH0(int ucDutyCycle)
{
    PWMCH0 = PWM_DAT1VALUE + (PWM_DAT0VALUE * (ucDutyCycle - 50)) / 100;
    return;
}
```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้