

การออกแบบชุดทดลองควบคุมระดับน้ำโดยใช้ MCU (Arduino)

THE DESIGN OF LEVEL CONTROL LABORATORY

USING (ARDUINO) MICROCONTROLLER



ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมการวัดคุม

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2556

การออกแบบชุดทดลองควบคุมระดับน้ำโดยใช้ MCU (Arduino)
THE DESIGN OF LEVEL CONTROL LABORATORY
USING (ARDUINO) MICROCONTROLLER



ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิชาวิศวกรรมการวัดคุม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษานั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเผยแพร่โดยไม่ได้รับอนุญาตจากเจ้าของลิขสิทธิ์
ปีการศึกษา 2556

THE DESIGN OF LEVEL CONTROL LABORATORY
USING (ARDUINO) MICROCONTROLLER



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

KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

ACADEMIC YEAR 2013

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเนื้อหาและข้อมูลข้างต้นนี้เข้าสู่ของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญาานิพนธ์ปีการศึกษา 2556
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ใบรับรองปริญญาานิพนธ์

หัวข้อปริญญาานิพนธ์ การออกแบบชุดทดลองควบคุมระดับน้ำโดยใช้ MCU (Arduino)
THE DESIGN OF LEVEL CONTROL LABORATORY USING
(ARDUINO) MICROCONTROLLER

นักศึกษาผู้จัดทำ นายพัสกร นนทอง รหัสนักศึกษา 53011111
นายพีรศักดิ์ สุภัทราพันธ์กุล รหัสนักศึกษา 53011168
นายพีรสิษฐ์ ทับเที่ยง รหัสนักศึกษา 53011169

ปริญญา วิศวกรรมศาสตรบัณฑิต
สาขาวิชา วิศวกรรมการวัดคุม
ปีการศึกษา 2556

อาจารย์ผู้ควบคุมปริญญาานิพนธ์	ลายมือชื่อ
รองศาสตราจารย์อานันต์ น่วมสำราญ	
รองศาสตราจารย์ ดร.วิทยา ทิพย์สุวรรณพร	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อปริญญานิพนธ์	การออกแบบชุดทดลองควบคุมระดับน้ำโดยใช้ MCU (Arduino) THE DESIGN OF LEVEL CONTROL LABORATORY USING (ARUINO) MICROCONTROLLER		
นักศึกษาผู้จัดทำ	นายพัสกร นนทอง	รหัสนักศึกษา	53011111
	นายพีรศักดิ์ สุภัทราพันธ์กุล	รหัสนักศึกษา	53011168
	นายพีรสิษฐ์ ทับเที่ยง	รหัสนักศึกษา	53011169
อาจารย์ที่ปรึกษา	รองศาสตราจารย์อาจินต์	นามสำราญ	
	รองศาสตราจารย์ ดร.วิทยา	ทิพย์สุวรรณพร	
ปีการศึกษา	2556		

บทคัดย่อ

ในปัจจุบันการศึกษาค้นคว้าเกี่ยวกับกระบวนการระดับน้ำโดยใช้ตัวควบคุมนั้นมีส่วนสำคัญเป็นอย่างมากซึ่งทางผู้ศึกษาเองได้เล็งเห็นถึงความสำคัญดังกล่าวจึงได้จัดทำโครงการการศึกษาในครั้งนี้ขึ้นมาโดยได้ศึกษาเกี่ยวกับการการออกแบบชุดทดลองควบคุมระดับน้ำโดยใช้ MCU (Arduino) โดยการออกแบบชุดทดลองและศึกษาการออกแบบตัวควบคุมPID เพื่อใช้ในการทดลอง และนำผลตอบสนองของระบบที่ได้มาพล็อตกราฟเพื่อดูเสถียรภาพของตัวควบคุมที่ออกแบบว่าสามารถนำมาใช้กับชุดทดลองได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Thesis Title	THE DESIGN OF LEVEL CONTROL LABORATORY USING (ARUINO) MICROCONTROLLER	
Authors	Mr. Patsakorn	Nontong
	Mr. Peerasak	Supattrapankul
	Mr. Perasit	TabTiang
Thesis Advisor	Assoc.Prof. Arjin	Numsomran
	Assoc.Prof.Dr. Wittaya	Tipsuwannaporn
Year	2013	

Abstract

In the present study on the water level process by using the controller is very important that the researcher have seen important it is to get the project up in the study about design of level control laboratory using MCU(Arduino) for design of laboratory and study design of PID controller used to experiment and response of system used to plot graph for check stability of controller that design can be applied to laboratory

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

ปริญญานิพนธ์ฉบับนี้สำเร็จลุล่วงได้ด้วยดี เพราะได้รับความเมตตาจาก รองศาสตราจารย์อา
จินต์ น่วมสำราญ และรองศาสตราจารย์ ดร. วิทยา ทิพย์สุวรรณพร ที่ได้ให้คำแนะนำแก่ผู้วิจัยตลอด
มา อีกทั้งยังเอื้อเฟื้ออุปกรณ์และเครื่องมือต่างๆ ในการทำปริญญานิพนธ์ ผู้วิจัยรู้สึกซาบซึ้งและขอ
กราบพระคุณเป็นอย่างสูง

ขอขอบพระคุณ อาจารย์ภาควิชาวิศวกรรมการวัดคุมทุกท่านและเพื่อนนักศึกษาทุกคนที่ให้
คำแนะนำอันเป็นประโยชน์ต่อการทำปริญญานิพนธ์ฉบับนี้

และที่ลืมไม่ได้ คือ ขอกราบขอบพระคุณคุณพ่อและคุณแม่อันเป็นที่รักยิ่ง ที่สนับสนุนและ
เป็นแรงบันดาลใจในการทำปริญญานิพนธ์ฉบับนี้

คุณค่าและประโยชน์อันพึงมีจากปริญญานิพนธ์ฉบับนี้ ผู้วิจัยขอมอบแด่ผู้มีพระคุณทุกท่าน

คณะผู้จัดทำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

	หน้า
บทคัดย่อภาษาไทย.....	I
บทคัดย่อภาษาอังกฤษ.....	II
กิตติกรรมประกาศ.....	III
สารบัญ.....	IV
สารบัญตาราง.....	VV
สารบัญภาพ.....	VIV
บทที่ 1 บทนำ.....	1
1.1 ความสำคัญของปริญญาโท.....	1
1.2 วัตถุประสงค์ของปริญญาโท.....	1
1.3 ขอบเขตของปริญญาโท.....	1
1.4 ขั้นตอนการศึกษา.....	2
1.5 ประโยชน์ที่คาดว่าจะได้รับ.....	2
บทที่ 2 ทฤษฎีและหลักการที่เกี่ยวข้อง.....	3
2.1 กล่าวนำ.....	3
2.2 ระบบควบคุม.....	3
2.2.1 รูปแบบของระบบควบคุม.....	4
2.2.1.1 ระบบควบคุมแบบวงรอบเปิด.....	4
2.2.1.2 ระบบควบคุมแบบวงรอบปิด.....	4
2.3 ทฤษฎีการควบคุมแบบพีไอดี.....	5
2.3.1 การควบคุมแบบป้อนกลับด้วยตัวควบคุมพีไอดี.....	6
2.3.1.1 การควบคุมแบบ Proportional (P).....	6
2.3.1.2 การควบคุมแบบ Integral (I).....	7
2.3.1.3 การควบคุมแบบ Derivative (D).....	8
2.3.1.4 การควบคุมแบบ Proportional-Integral (PI).....	9
2.3.1.5 การควบคุมแบบ Proportional-Derivative (PD).....	10
2.3.1.6 การควบคุมแบบ Proportional-Integral-Derivative (PID).....	10

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้สำหรับใช้ภายในคณะวิศวกรรมศาสตร์ มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้ง

สารบัญ (ต่อ)

	หน้า
2.3.2 ผลของพารามิเตอร์ของตัวควบคุมแบบพีไอดีกับกระบวนการ	11
2.3.2.1 ผลของ Proportional Action (Kp) ต่อระบบ	11
2.3.2.2 ผลของ Integral Action (Ki) ต่อระบบ	12
2.3.2.3 ผลของ Derivative Action (Kd) กับผลตอบสนองต่อระบบ	12
2.3.3 การคำนวณหาค่าพารามิเตอร์ของตัวควบคุมแบบพีไอดี	13
จากผลตอบสนองของกระบวนการ	
2.3.3.1 วิธี Process Reaction Curve	14
2.3.3.2 วิธี Ultimate Method	16
2.3.3.3 วิธี Trial and Error	18
2.4 LabVIEW	20
2.4.1 Front Panel	20
2.4.1.1 Control and Indicator	21
2.4.1.2 Front Panel Window Toolbar	22
2.4.2 Block Diagram	22
2.4.2.1 Block Diagram Window Toolbar	23
2.4.3 ประเภทของข้อมูล	24
2.4.3.1 Numeric	24
2.4.3.2 Boolean	24
2.4.3.3 String	25
2.4.3.4 Enum	25
2.4.3.5 Dynamic	26
2.4.3.6 Time Stamp	27
2.4.3.7 Waveform	27
2.4.4 ลูป	28
2.4.4.1 While Loop	28
2.4.4.2 For Loop	29
2.5 ทฤษฎีเบื้องต้นที่ใช้ในการเขียนโปรแกรมภาษาซีของ Arduino	30
2.5.1 โครงสร้างการเขียนโปรแกรมภาษาซีของ Arduino	30
2.5.2 ประเภทและชนิดของตัวแปร	32

สารบัญ (ต่อ)

	หน้า
2.6 สรุป.....	33
บทที่ 3 การประยุกต์ใช้ฮาร์ดแวร์และซอฟต์แวร์	34
3.1 กล่าวนำ	34
3.2 การทำงานของฮาร์ดแวร์ในกระบวนการ	34
3.3 องค์ประกอบด้านฮาร์ดแวร์.....	35
3.3.1 กระบวนการระดับน้ำ.....	35
3.3.2 อุปกรณ์วัดระดับน้ำ.....	36
3.3.2.1 Feature.....	36
3.3.2.2 Package Dimension	36
3.3.3 มอเตอร์ปั้มน้ำ.....	36
3.3.4 ชุดขับปั้มน้ำ.....	37
3.3.4.1 Switching Power Supply	38
3.3.4.2 บอร์ดขับมอเตอร์ดีซี.....	39
3.3.5 ไมโครคอนโทรลเลอร์ Arduino.....	40
3.4 องค์ประกอบด้านซอฟต์แวร์	43
3.4.1 โปรแกรมคอมไพเลอร์ Arduino IDE	43
3.4.1.1 ขั้นตอนการติดตั้งโปรแกรม.....	43
3.4.1.2 การแก้ปัญหาในกรณีที่ระบบไม่สามารถค้นพบไดรเวอร์.....	45
3.5 การติดตั้งซอฟต์แวร์ LabVIEW2011.....	47
3.6 สรุป.....	51
บทที่ 4 การทดลองและผลการทดลอง	52
4.1 กล่าวนำ	52
4.2 กระบวนการที่ใช้ในการทดลอง.....	52
4.3 การจำลองกระบวนการ	53
4.3.1 การทดลองควบคุมกระบวนการระดับน้ำแบบ 1 ถึงด้วยวิธีวงรอบเปิด.....	53
4.3.2 การทดลองควบคุมกระบวนการระดับน้ำแบบ 2 ถึง.....	56
4.3.2.1 การควบคุมกระบวนการระดับน้ำโดยใช้วงรอบเปิด.....	56

สารบัญ (ต่อ)

	หน้า
4.3.2.2 การควบคุมกระบวนการระดับน้ำโดยใช้วงรอบปิด.....	58
4.3.3 การทดลองควบคุมกระบวนการระดับน้ำแบบ 4 ถัง โดยวิธีวงรอบปิด	61
4.4 สรุป.....	63
บทที่ 5 สรุปผลการวิจัยและข้อเสนอแนะ.....	64
5.1 สรุปผลการทดลอง.....	64
5.2 ปัญหาและอุปสรรค.....	64
5.3 ข้อเสนอแนะ	65
บรรณานุกรม.....	66



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้.

สารบัญตาราง

ตารางที่	หน้า
2.1 แสดงค่าพารามิเตอร์ของตัวควบคุมแบบต่างๆตามวิธี Process Reaction Curve.....	16
2.2 แสดงค่าพารามิเตอร์ของตัวควบคุมแบบต่างๆตามวิธี Ultimate Method.....	18
2.3 แสดงคุณสมบัติของตัวแปรภาษาซี.....	33
3.1 แสดงคุณสมบัติของมอเตอร์ปั้มน้ำ.....	37
3.2 แสดงคุณสมบัติของ Switching Power Supply.....	38
4.1 ผลตอบสนองของกระบวนการดังรูปที่ 4.15.....	60
4.2 ผลตอบสนองของกระบวนการดังรูปที่ 4.16.....	61
4.3 ผลตอบสนองของกระบวนการดังรูปที่ 4.19.....	63



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป

รูปที่	หน้า
2.1 ส่วนประกอบของระบบควบคุม.....	3
2.2 การควบคุมอุณหภูมิห้องด้วยเครื่องปรับอากาศ.....	3
2.3 ระบบควบคุมวงจรถูกเปิด.....	4
2.4 ระบบควบคุมวงจรถูกปิด.....	4
2.5 ระบบควบคุมแบบป้อนกลับ.....	5
2.6 แสดงคุณสมบัติของปฏิกิริยาการควบคุมแบบ Proportional.....	7
2.7 แสดงผลตอบสนองของปฏิกิริยาการควบคุมแบบ Integral.....	8
2.8 แสดงตัวอย่างผลตอบสนองของปฏิกิริยาการควบคุมแบบ Derivative.....	9
2.9 แสดงตัวอย่างผลตอบสนองของปฏิกิริยาการควบคุมแบบ PI (Direct action).....	9
2.10 แสดงตัวอย่างผลตอบสนองของปฏิกิริยาการควบคุมแบบ PD.....	10
2.11 แสดงตัวอย่างผลตอบสนองของปฏิกิริยาการควบคุมแบบ PID (Direct action).....	11
2.12 ผลของ Proportional Gain ต่อระบบควบคุมแบบ PID.....	11
2.13 ผลของ Integral Action ต่อระบบควบคุมแบบ PID.....	12
2.14 ผลของ Derivative Action ต่อระบบควบคุมแบบ PID.....	13
2.15 แสดงค่า Overshoot ที่ยอมรับได้ในการปรับวิธีต่างๆ.....	14
2.16 ผลตอบสนองวงเปิดของกระบวนการที่มี Deadtime และการล่าช้า.....	15
2.17 การแกว่งด้วยแอมพลิจูดที่คงที่เมื่อปรับโดยใช้วิธี Ultimate method.....	17
2.18 การหาค่าอัตราขยายที่ทำให้ระบบเริ่มแกว่ง.....	17
2.19 หน้าต่างของ Front Panel.....	20
2.20 หน้าต่างของ Control Palette.....	21
2.21 หน้าต่างของ Front Panel แสดงตัว Control และ Indicator.....	21
2.22 หน้าต่างของ Functions Palette.....	23
2.23 หน้าต่างของ Mechanical Action.....	24
2.24 หน้าต่างของ Signal Manipulation.....	26
2.25 กราฟ Waveform.....	27
2.26 แสดง Flow Chart ของ While Loop.....	28
2.27 แสดง Flow Chart ของ For Loop.....	29
2.28 วิธีสร้างเงื่อนไขให้ For Loop.....	29
3.1 ภาพการทำงานของฮาร์ดแวร์ในกระบวนการ.....	34

สารบัญรูป (ต่อ)

รูปที่	หน้า
3.2 กระบวนการระดับน้ำประกอบด้วยถังบรรจุน้ำ 2 ใบ วาล์วที่ใช้เปิด-ปิดระดับน้ำ.....	35
3.3 อุปกรณ์วัดระดับน้ำ (MPXV5004G)	36
3.4 มอเตอร์ปั้มน้ำ.....	36
3.5 ชุดขับมอเตอร์ปั้มน้ำ.....	37
3.6 Switching Power Supply.....	38
3.7 บอร์ดขับมอเตอร์ดีซี	39
3.8 การเชื่อมต่อในอุปกรณ์เพื่อนำไปใช้งาน	40
3.9 บอร์ด Arduino	40
3.10 รายละเอียดแสดงขาของ Atmega 328.....	42
3.11 แสดงไฟล์ที่ดาวน์โหลดเพื่อมาติดตั้ง.....	43
3.12 แสดงการติดตั้งระหว่างคอมพิวเตอร์กับไมโครคอนโทรลเลอร์	44
3.13 แสดงการเลือกรุ่นของ Arduino Board	44
3.14 แสดงหน้าต่างโปรแกรมเมื่อ Compile ผ่านแล้ว	45
3.15 หน้าต่าง Device Manager.....	45
3.16 แสดงการเข้าไปที่หน้าต่าง Update Driver Software.....	46
3.17 หน้าต่าง Update Driver Software.....	46
3.18 แสดงการเลือกไฟล์ที่จะนำมาอัปเดต	46
3.19 เริ่มต้นการติดตั้ง.....	47
3.20 กรอกชื่อผู้ใช้งานและสถานที่ทำงาน.....	47
3.21 กรอก Serial Number	48
3.22 เลือกตำแหน่งที่จะทำการติดตั้ง.....	48
3.23 เช็ควอร์ชันออนไลน์.....	49
3.24 แสดงลิขสิทธิ์และสิทธิในการใช้งาน.....	49
3.25 แสดงลิขสิทธิ์และสิทธิในการใช้งาน.....	50
3.26 ขั้นตอนการดำเนินการติดตั้งโปรแกรม.....	50
4.1 กระบวนการระดับน้ำ.....	52
4.2 การออกแบบกระบวนการ.....	53
4.3 แสดงกระบวนการควบคุมระดับน้ำแบบ 1 ถัง.....	54

สารบัญรูป (ต่อ)

รูปที่	หน้า
4.4 แสดงโปรแกรมที่ใช้ควบคุมกระบวนการ.....	54
4.5 ป้อนค่า $MV = 30$	55
4.6 สัญญาณ Step จาก $MV = 30$ ไปยัง $MV = 33$	55
4.7 ผลตอบสนองรูปตัว S เมื่อใช้วิธีแบบวงเปิด.....	56
4.7 ป้อนค่า $MV = 30$	56
4.8 แสดงกระบวนการควบคุมระดับน้ำแบบ 2 ถึง.....	56
4.9 ป้อนค่า $MV = 28$	57
4.10 สัญญาณ Step จาก $MV = 28$ ไปยัง $MV = 31$	57
4.11 ผลตอบสนองรูปตัว S เมื่อใช้วิธีแบบวงเปิด.....	58
4.12 แสดงส่วนของ Block Diagram ที่ใช้ควบคุมกระบวนการ.....	58
4.13 กำหนดค่า $SV = 50$	59
4.14 การกำหนดค่าพารามิเตอร์.....	59
4.15 ผลตอบสนองของกระบวนการเมื่อใช้วงรอบปิด.....	60
4.16 แสดงผลตอบสนองของกระบวนการควบคุมระดับน้ำแบบ 2 ถึง.....	61
4.17 แสดงกระบวนการควบคุมระดับน้ำแบบ 4 ถึง.....	61
4.18 แสดงโปรแกรมการควบคุมกระบวนการระดับน้ำ 4 ถึง.....	62
4.19 แสดงผลตอบสนองของกระบวนการควบคุมระดับ 4 ถึง แบบ PID 2 ชุด.....	62

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1

บทนำ

1.1 ความสำคัญของปริญญานิพนธ์

ในกระบวนการอุตสาหกรรมในปัจจุบัน การควบคุมระดับน้ำในถังหรือกระบวนการมีความสำคัญเป็นอย่างมาก เนื่องจากในกระบวนการทางอุตสาหกรรมที่เกี่ยวข้องกับของเหลว จำเป็นต้องมีที่เก็บผลิตภัณฑ์ที่เกี่ยวข้อง และจำเป็นต้องมีการควบคุมระดับของของเหลวในกระบวนการหรือปริมาณของเหลวที่เป็นปัจจัยในการผลิต เพื่อให้กระบวนการผลิตเป็นไปอย่างมีประสิทธิภาพ และเป็นไปตามความต้องการของผู้ผลิต หากเราใช้ตัวควบคุมที่มีประสิทธิภาพและคอยตรวจสอบอยู่เสมอ ก็จะขจัดปัญหาความผิดพลาดไปได้ ซึ่งถ้าตัวควบคุมผิดพลาดหรือคลาดเคลื่อนไป อาจทำให้กระบวนการผลิตเกิดความสูญเสีย หรือต้องหยุดทำงานชั่วคราว ซึ่งจะส่งผลให้ผู้ผลิตต้องเพิ่มต้นทุนในการผลิตสูงขึ้นเพื่อให้ทันต่อการใช้งาน และถ้าเกิดความร้ายแรงมากกว่านั้น กล่าวคือ โรงงานหรือกระบวนการขาดการควบคุมระดับของเหลวที่มีประสิทธิภาพ อาจส่งผลให้เครื่องจักรเสียหายหรือกระบวนการที่เกี่ยวข้องเนื่องกันทำงานต่อไม่ได้ ส่งผลให้ต้องหยุดการผลิต ทำให้เกิดความล่าช้าในการผลิต ทำให้สิ่งของที่จะผลิตผิดพลาดจากเป้าหมายที่วางไว้ ซึ่งทางผู้ศึกษาได้เล็งเห็นความสำคัญของการศึกษากระบวนการควบคุมระดับน้ำหรือของเหลว โดยในการศึกษานำเสนอการออกแบบชุดทดลองควบคุมระดับน้ำในถัง โดยจะใช้ตัวควบคุมแบบ PID ควบคุมผ่านทางไมโครคอนโทรลเลอร์ Arduino UNO R2 และนำผลที่ได้จากการทดลองมาแสดงเป็นกราฟแบบ Realtime โดยใช้โปรแกรม LabVIEW ซึ่งในการทดลองนี้ เพื่อให้เกิดประโยชน์แก่ผู้ที่อยากศึกษากระบวนการนี้และผู้ที่สนใจสามารถเข้าใจกระบวนการควบคุมระดับน้ำหรือของเหลวได้ชัดเจนมากยิ่งขึ้น

1.2 วัตถุประสงค์ของปริญญานิพนธ์

1. เพื่อศึกษากระบวนการระดับน้ำและออกแบบชุดทดลอง
2. เพื่อศึกษาการทำงานของ MCU (Arduino)
3. เพื่อศึกษาและออกแบบตัวควบคุมพีไอดี

เอกสารที่ 1.3 ขอบเขตของปริญญานิพนธ์ งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ 1. จัดทำชุดทดลองกระบวนการระดับน้ำ ซึ่งประกอบด้วย อุปกรณ์วัดระดับน้ำ ชุดขับปั๊มน้ำ ไปใช้

แบบมอเตอร์กระแสตรง และถ่วงพิกน้ำ

2. จัดทำโปรแกรมควบคุมสำหรับกระบวนการระดับน้ำ
3. จัดทำปริญญานิพนธ์
4. จัดทำวีดิทัศน์สาธิตการทำงานของระบบ

1.4 ขั้นตอนการศึกษา

1. วางแผนงานการทำโครงการ โดยกำหนดระยะเวลาในการดำเนินงาน
2. ดำเนินการตามแผน โดยเริ่มจากการศึกษาความรู้และทฤษฎีที่ต้องใช้
3. จัดทำชุดทดลองกระบวนการระดับน้ำ และทำการออกแบบตัวควบคุม

1.5 ประโยชน์ที่คาดว่าจะได้รับ

1. สามารถเขียนโปรแกรมด้วย ไมโครคอนโทรเลอร์ เพื่อควบคุมกระบวนการระดับน้ำได้
2. สามารถออกแบบชุดทดลองระดับน้ำและตัวควบคุมพีไอดีได้
3. มีความรู้ความเข้าใจเกี่ยวกับกระบวนการระดับน้ำ



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

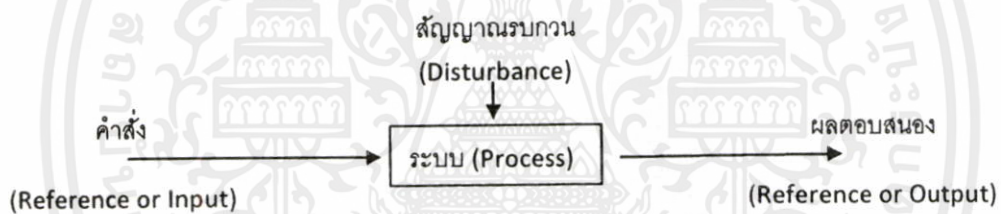
บทที่ 2

ทฤษฎีและหลักการที่เกี่ยวข้อง

2.1 กล่าวนำ

ในบทนี้จะว่าด้วยทฤษฎีของควบคุม ทฤษฎีการควบคุมแบบพีไอดี และทฤษฎีการใช้งานภาษาซีเบื้องต้น เพื่อให้ผู้อ่านได้มีความรู้ทางด้านระบบควบคุม และการประยุกต์ใช้ภาษาซีในการเขียนโปรแกรมควบคุมกระบวนการ ซึ่งในการออกแบบตัวควบคุมและการเขียนโปรแกรมควบคุมกระบวนการจำเป็นต้องมีความรู้ความเข้าใจเกี่ยวกับทฤษฎีเหล่านี้ เพื่อให้สามารถออกแบบตัวควบคุมได้อย่างมีประสิทธิภาพ

2.2 ระบบควบคุม



รูปที่ 2.1 ส่วนประกอบของระบบควบคุม

ระบบควบคุมคือ ส่วนหรือหน่วยที่ได้รวบรวมสิ่งต่าง ๆ เข้าด้วยกัน เพื่อใช้บังคับหรือสั่งการเพื่อให้สิ่งใดสิ่งเป็นไปตามความต้องการ

ตัวอย่างเช่น การควบคุมอุณหภูมิห้องด้วยเครื่องปรับอากาศ ในรูปที่ 2.2

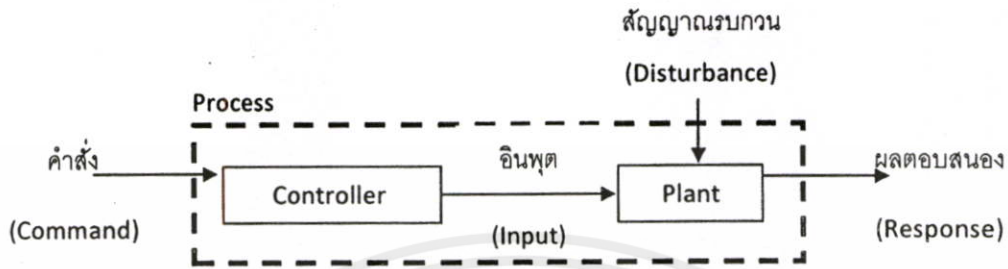


รูปที่ 2.2 การควบคุมอุณหภูมิห้องด้วยเครื่องปรับอากาศ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.1 รูปแบบของระบบควบคุม

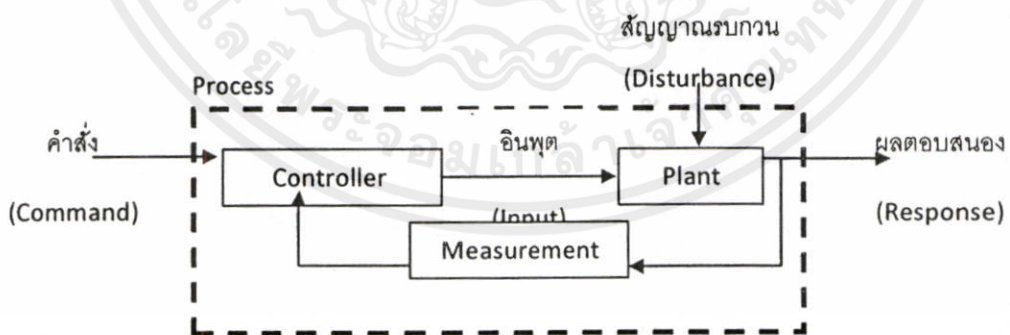
2.2.1.1 ระบบควบคุมแบบวงรอบเปิด (Open Loop Control System)



รูปที่ 2.3 ระบบควบคุมแบบวงรอบเปิด

ระบบควบคุมแบบวงรอบเปิดหรือที่เรียกอีกอย่างหนึ่งว่า ระบบควบคุมแบบที่ไม่มีป้อนกลับ ลักษณะทั่วไปจะเป็นไปตามรูปที่ 2.3 ในการควบคุมแบบวงรอบเปิด ตัวควบคุม (Controller) จะส่งสัญญาณป้อน(Input) ให้กับสิ่งที่ต้องการควบคุม(Plant) ตามคำสั่งหรือสัญญาณอ้างอิง(Command or reference) ที่รับมา โดยเมื่อตัวควบคุมได้รับสัญญาณป้อนแล้ว ก็จะผลิตเอาต์พุตหรือผลตอบสนอง(Response) ให้ได้ตามที่คาดหวัง โดยที่ไม่มีการตรวจสอบสัญญาณเอาต์พุตจริงว่าเป็นไปตามคำสั่งหรือไม่

2.2.1.2 ระบบควบคุมแบบวงรอบปิด (Closed Loop Control System)

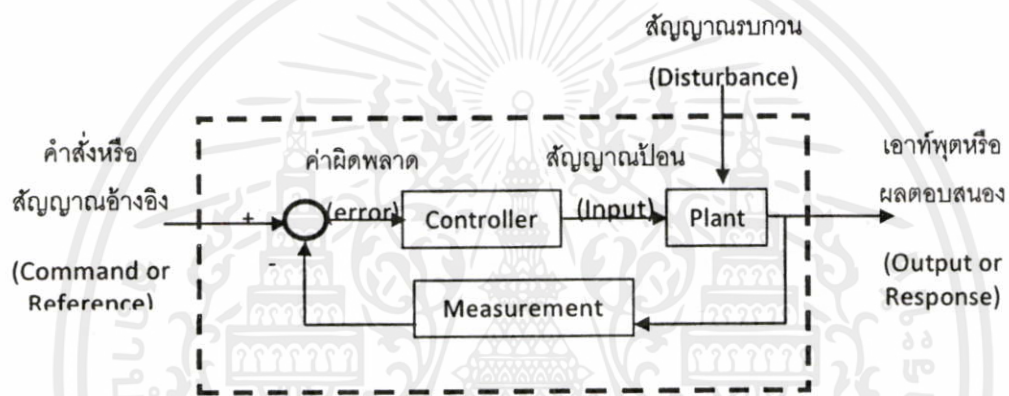


รูปที่ 2.4 ระบบควบคุมแบบวงรอบปิด

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ของระบบควบคุมแบบวงรอบปิดจะเป็นไปตามรูปที่ 2.4 ในการดำเนินการค้า ไม่ว่าจะกรณีใดก็ตาม ระบบควบคุมแบบวงรอบปิด ตัวควบคุม(Controller) จะทำการเปรียบเทียบสัญญาณอ้างอิงหรือคำสั่งป้อน

(Reference or command) กับสัญญาณเอาต์พุตหรือผลตอบสนอง(Output or response) ที่ป้อนกลับมาโดยตัวตรวจจับ(Measurement or sensor) แล้วนำไปสร้างสัญญาณป้อนหรืออินพุต(Input) ให้กับสิ่งที่ต้องการควบคุม(System under controlled or plant) เพื่อที่จะให้ผลิตเอาต์พุตหรือผลตอบสนองให้เป็นไปตามสัญญาณอ้างอิงที่ต้องการ

ระบบควบคุมแบบวงรอบปิดอาจจะเรียกได้อีกอย่างหนึ่งว่า ระบบควบคุมแบบป้อนกลับ (Feedback Control System) ตามรูปที่ 2.5 ระบบนี้เป็นระบบควบคุมที่พยายามรักษาเอาต์พุตให้ได้ตามที่ต้องการ โดยนำเอาสัญญาณเอาต์พุตมาเปรียบเทียบกับสัญญาณอ้างอิงที่ต้องการ แล้วนำค่าความแตกต่างไปใช้ในการควบคุมสัญญาณป้อนให้กับสิ่งที่ต้องการควบคุม



รูปที่ 2.5 ระบบควบคุมแบบป้อนกลับ

จากระบบควบคุมทั้งสองประเภทจะกล่าวได้ว่า ถ้าต้องการจะควบคุมค่าเอาต์พุตของกระบวนการ ก็ต้องใส่ตัวควบคุม เพื่อคอยควบคุมเอาต์พุตของกระบวนการให้เป็นไปตามที่ต้องการโดยในหัวข้อถัดไปจะเป็นการอธิบายถึง ทฤษฎีของตัวควบคุมพีไอดี ซึ่งเป็นตัวควบคุมแบบป้อนกลับที่ใช้กันอย่างกว้างขวาง รายละเอียดจะกล่าวในหัวข้อถัดไป

2.3 ทฤษฎีการควบคุมแบบพีไอดี

เป็นระบบควบคุมแบบป้อนกลับ ซึ่งค่าที่นำไปใช้ในการคำนวณเป็นค่าความผิดพลาดที่หา มาจากความแตกต่างของตัวแปรในกระบวนการและค่าที่ต้องการ ตัวควบคุมจะพยายามลดค่าผิดพลาดให้เหลือน้อยที่สุดด้วยการปรับสัญญาณป้อนให้กับกระบวนการ

เอกสารนี้เป็นเอกสารวิจัยของคณะวิศวกรรมศาสตร์ มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี
 ไม่ว่าจะตีพิมพ์หรือเผยแพร่ในรูปแบบใดก็ตาม กรุณาแจ้งที่มาของเอกสารฉบับนี้แก่ผู้จัดทำไว้ด้วย
 ไม่ว่ากรณีใดๆก็ตาม การนำเอกสารฉบับนี้ไปใช้โดยไม่ได้รับอนุญาตจากคณะวิศวกรรมศาสตร์ มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี จะถือว่าผิดกฎหมาย

พันไป, และค่า K_d กำหนดจากผลบนพื้นฐานของอัตราการเปลี่ยนแปลงของค่าความผิดพลาดผลรวมตามน้ำหนักของทั้งสามนี้ จะใช้ในการปรับกระบวนการ

2.3.1 การควบคุมแบบป้อนกลับด้วยตัวควบคุมพีโอดี

2.3.1.1 การควบคุมแบบ Proportional (P)

กิริยาการควบคุม P นั้น ค่าเอาต์พุตของตัวควบคุมจะแปรผันตรงกับค่าความผิดพลาด กล่าวคือ ถ้าค่าความผิดพลาดมีค่ามากขึ้น ค่าเอาต์พุตของตัวควบคุมก็จะมีค่ามากขึ้นตาม และถ้าค่าความผิดพลาดมีค่าน้อยลง ค่าเอาต์พุตของตัวควบคุมก็จะมีค่าน้อยลงตามกิริยาการควบคุมแบบ P สามารถเขียนสมการ (2.1) ได้ดังนี้

$$mp(t) = Kp e(t) + \bar{m} \quad (2.1)$$

$mp(t)$ = ค่าเอาต์พุตของตัวควบคุมแบบ Proportional

Kp = อัตราขยายของตัวควบคุมแบบ Proportional

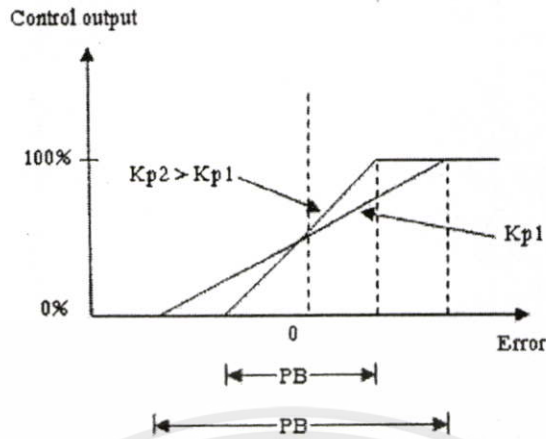
\bar{m} = ค่าเอาต์พุตของตัวควบคุมที่ค่าความผิดพลาดเป็นศูนย์

ตัวควบคุมแบบ P บางตัวอาจจะใช้ค่า Proportional Band (PB) แทนการใช้ค่า Kp ซึ่ง PB คือช่วงของค่าความผิดพลาดระหว่างที่เอาต์พุตของตัวควบคุมมีค่า 0 – 100% ดังสมการ

$$PB = \frac{100\%}{Kp}$$

ผลอัตราขยายสัดส่วนที่สูงค่าความผิดพลาดก็จะเปลี่ยนแปลงมากเช่นกัน แต่ถ้าสูงเกินไประบบจะไม่เสถียรได้ ในทางตรงกันข้าม ถ้าผลอัตราขยายสัดส่วนที่ต่ำ ระบบควบคุมจะมีผลตอบสนองต่อกระบวนการน้อยตามไปด้วย และข้อเสียของกิริยาการควบคุมแบบ Proportional (P) คือ ไม่สามารถกำจัดออฟเซต(Offset)ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.6 แสดงคุณสมบัติของกริยาการควบคุมแบบ Proportional

2.3.1.2 การควบคุมแบบ Integral (I)

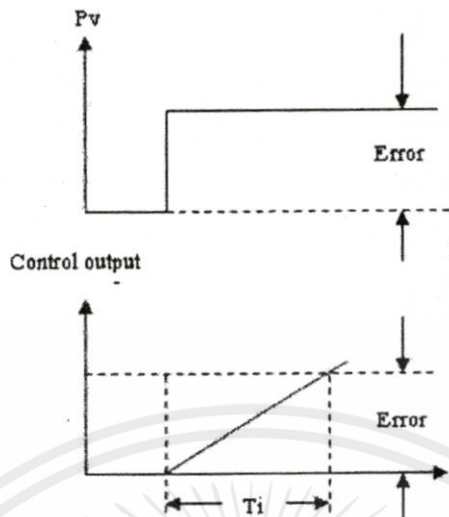
กริยาควบคุมแบบ I หรือเรียกอีกอย่างหนึ่งว่า การควบคุมแบบรีเซ็ต (Reset Control) โดยค่าเอาต์พุตของตัวควบคุม หาได้จากค่าพื้นที่ทั้งหมดภายใต้กราฟของค่าความผิดพลาดต่อเวลาคูณกับค่าคงที่ ที่เรียกว่า อัตราขยายของตัวควบคุมแบบ I (Integral Gain) ดังสมการ (2.2)

$$m_I(t) = K_I \int_0^t e(t) dt + \bar{m}_I(0) \quad (2.2)$$

เมื่อ $m_I(t)$ = ค่าเอาต์พุตของตัวควบคุมแบบ Integral
 K_I = อัตราขยายของตัวควบคุมแบบ Integral
 $\int_0^t e(t) dt$ = พื้นที่ทั้งหมดของความผิดพลาด
 $\bar{m}_I(0)$ = ค่าเอาต์พุตของตัวควบคุมที่เวลา t เท่ากับศูนย์

ผลของกริยาควบคุมแบบ I นี้จะเร่งกระบวนการให้เข้าสู่จุดที่ต้องการและขจัดค่าความผิดพลาดที่เหลืออยู่ที่เกิดจากการใช้เพียงเทอมสัดส่วนหรือขจัดออฟเซต(offset) นั้นเอง แต่อย่างไรก็ตาม ตัวควบคุมแบบ I เป็นการตอบสนองต่อความผิดพลาดสะสมในอดีต จึงสามารถทำให้เกิดโอเวอร์ชูต(Overshoot)ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.7 แสดงผลตอบสนองของกิริยาการควบคุมแบบ Integral

2.3.1.3 การควบคุมแบบ Derivative (D)

กิริยาควบคุมแบบ D เรียกอีกชื่อหนึ่งว่า การควบคุมอัตราส่วน (Rate Action) โดยสัญญาณเอาต์พุตของตัวควบคุมจะขึ้นอยู่กับอัตราการเปลี่ยนแปลงของค่าความผิดพลาดต่อเวลา จะเห็นได้ว่าความผิดพลาดนี้มีโอกาสเป็นศูนย์ได้ และค่าเอาต์พุตก็สามารถเปลี่ยนแปลงให้มีค่าสูงขึ้นเมื่อความผิดพลาดเปลี่ยนแปลง ซึ่งเรียกการกระทำดังกล่าวว่า อัตราการกระทำ (Rate Action) ดังสมการ (2.3)

$$m_D = K_d \frac{de(t)}{dt} \tag{2.3}$$

เมื่อ m_D = ค่าเอาต์พุตของตัวควบคุมแบบ Derivative

K_d = อัตราขยายของตัวควบคุมแบบ Derivative

$\frac{de(t)}{dt}$ = ค่าความผิดพลาดที่เวลา t

จุดเด่นของตัวควบคุมแบบ D คือ ใช้สำหรับลดผลของการพุ่งเกินของ

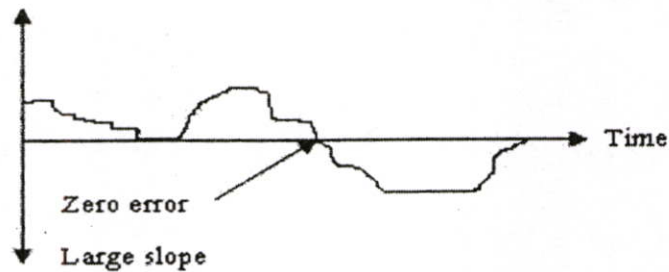
ผลตอบสนอง(Overshoot)ได้ แต่ต้องปรับค่าให้เหมาะสม ไม่เช่นนั้น อาจจะทำให้ได้ผลตอบสนองที่

ไม่พึงประสงค์ได้ก็คือ ตัวควบคุมแบบนี้ไม่สามารถขจัดค่าความผิดพลาดในสภาวะคงตัวได้ และอาจทำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไมออนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ให้ผลตอบสนองที่ช้าลงได้

ไม่ว่ากรณีใดๆทั้งนั้น อีกทั้งยังมีให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.8 แสดงตัวอย่างผลตอบสนองของรียาควบคุมแบบ Derivative

2.3.1.4 การควบคุมแบบ Proportional-Integral (PI)

ตามที่กล่าวมาแล้วว่ากรียาการควบคุมแบบ P นั้น จะมีออฟเซ็ทเกิดขึ้น ซึ่งการจำกัดค่าออฟเซ็ทนี้ สามารถทำได้โดยการเพิ่มกรียาควบคุมแบบ I เข้าไป ดังสมการ

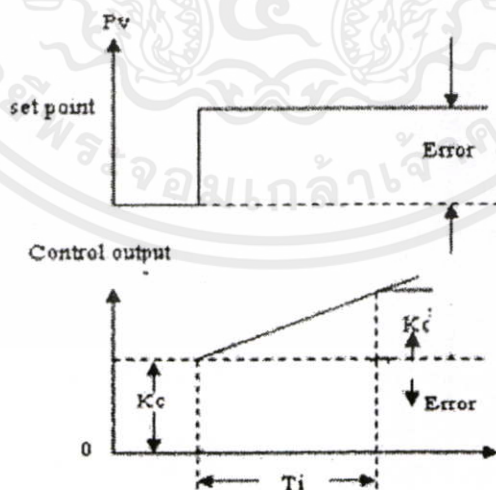
$$m_{PI}(t) = \bar{m} + K_P e(t) + K_P K_I \int_0^t e(t) dt \tag{2.4}$$

หรือ

$$m_{PI}(t) = \bar{m} + K_C e(t) + \frac{K_C}{T_i} \int_0^t e(t) dt \tag{2.5}$$

เมื่อ

$$K_C = K_P \text{ และ } K_I = \frac{1}{T_i}, T_i = \text{ค่าเวลา Integral}$$



เอกสารนี้เป็นเอกสารที่ 2.9 แสดงตัวอย่างผลตอบสนองของกรียาการควบคุมแบบ PI (Direct Action) ระบุข้อดีข้อเสียของการนำ

ไม่ว่ากรณีใดๆทั้งสิ้น 2.3.1.5 การควบคุมแบบ Proportional-Derivative (PD) เอกสารทุกครั้งที่มีการนำไปใช้

การประยุกต์ใช้การควบคุมแบบ P ร่วมกับกริยาควบคุมแบบ D เพื่อให้ผลตอบสนองของระบบรวดเร็วขึ้น แต่จะไม่มีผลโดยตรงต่อผลตอบสนองของระบบที่สภาวะคงที่ ซึ่งสมการเอาต์พุตของกริยาการควบคุมแบบ PD แสดงดังสมการต่อไปนี้

$$m_{PD}(t) = \bar{m} + K_P e(t) + K_P K_D \frac{de(t)}{dt} \quad (2.6)$$

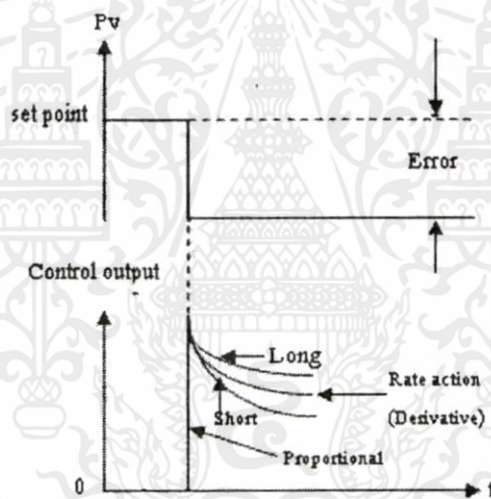
หรือ

$$m_{PD}(t) = \bar{m} + K_c e(t) + K_c T_d \frac{de(t)}{dt} \quad (2.7)$$

เมื่อ

$$K_D = T_d$$

$$T_d = \text{ค่าเวลา Derivative}$$



รูปที่ 2.10 แสดงตัวอย่างผลตอบสนองของกริยาการควบคุมแบบ PD

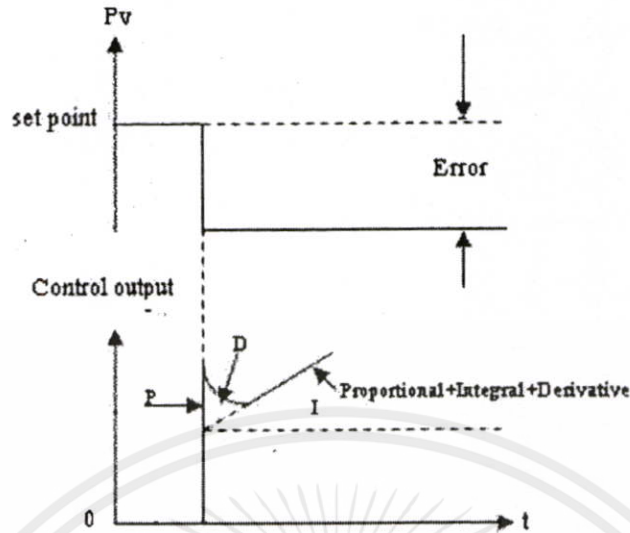
2.3.1.6 การควบคุมแบบ Proportional - Integral - Derivative (PID)

เพื่อให้ผลตอบสนองของระบบควบคุมมีสมรรถนะเป็นไปตามที่ต้องการ จึงใช้กริยาการควบคุมทั้ง 3 แบบร่วมกัน ซึ่งจะทำให้ได้กริยาการควบคุมแบบ PID ดังสมการนี้

$$m_{PID}(t) = \bar{m} + K_P e(t) + K_P K_I \int_0^t e(t) dt + K_P K_D \frac{de(t)}{dt} \quad (2.8)$$

หรือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ $m_{PID}(t) = \bar{m} + K_c e(t) + \frac{K_c}{T_i} \int_0^t e(t) dt + K_c T_D \frac{de(t)}{dt}$ รังที่มีก 2.9) ไปใช้

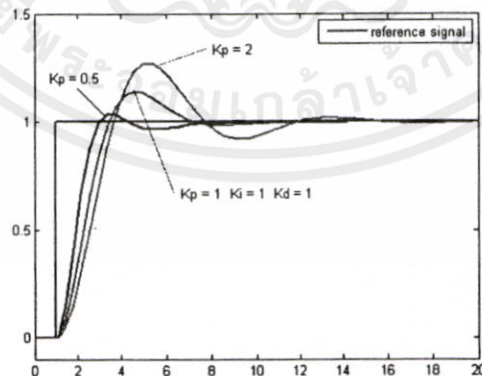


รูปที่ 2.11 แสดงตัวอย่างผลตอบสนองของกริยาการควบคุมแบบ PID (Direct Action)

2.3.2 ผลของพารามิเตอร์ของตัวควบคุมแบบ PID กับกระบวนการ

2.3.2.1 ผลของ Proportional Action (K_p) ต่อระบบ

ผลของ Proportional Gain (K_p) ต่อระบบ คือ เมื่อเพิ่มค่า K_p มากเกินไป จะส่งผลทำให้ค่าความผิดพลาด (Error) ในสภาวะคงตัว (Steady State) มากขึ้นและทำให้ความไวของผลตอบสนองเพิ่มขึ้นด้วย ในทางกลับกันค่าของ K_p น้อยเกินไปจะทำให้ระบบเกิดการแกว่งขึ้นหรือเข้าใกล้จุดไม่เสถียร (Unstable) มากขึ้น รูปที่ 2.12 แสดงผลการเพิ่มหรือลดค่าของ K_p ต่อกระบวนการกำลังสอง

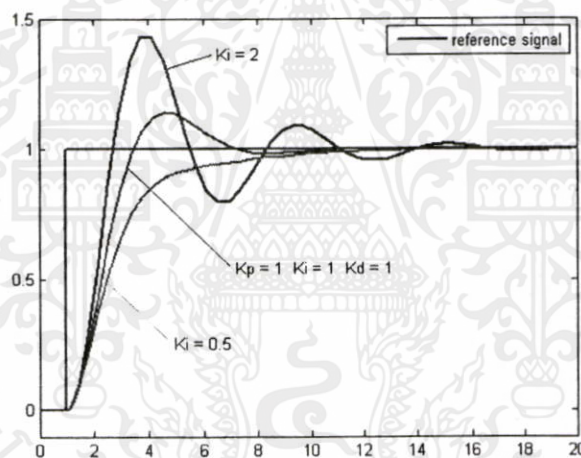


รูปที่ 2.12 ผลของ Proportional Gain ต่อระบบควบคุมแบบ PID

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3.2.2 ผลของ Integral Action (Ki) ต่อระบบ

สำหรับ Integral Action (Ki) นั้นจะมีผลตอบสนองของค่าสะสมแต่จะไม่มีผลกับผลตอบสนองในลักษณะทันทีทันใด และเป็นสัดส่วนโดยตรงกับขนาดของความผิดพลาดสะสมของผลตอบสนอง แต่ถ้าค่า Ki มากเกินไป จะทำให้ผลตอบสนองของระบบเกิดการแกว่งขึ้น และเวลาที่ใช้ในการเข้าสู่สภาวะคงตัวจะนานขึ้น ในทำนองกลับกัน การลดค่า Ki จะมีผลทำให้เวลาที่ใช้ในการเข้าสู่สภาวะคงตัวเร็วขึ้น คือจะเพิ่มค่าความผิดพลาดในสถานะคงตัวของระบบ ในกรณีที่ระบบนั้นมีค่า Integral Action เดิมมากเกินไป เมื่อมีการเริ่มต้นเดินเครื่องแบบอัตโนมัติ (Automatic Starting) ด้านการควบคุมของตัว PID Integral Action จะทำให้เกิด Integral Windup เนื่องจากขณะเริ่มต้นเดินเครื่อง ความผิดพลาดจะเกิดขึ้นเป็นเวลานาน ทำให้เทอมของ Integral มีค่าเกิน 100% รูปที่ 2.13 แสดงผลของการเพิ่มหรือลดค่า Integral Action ในการควบคุมแบบ PID



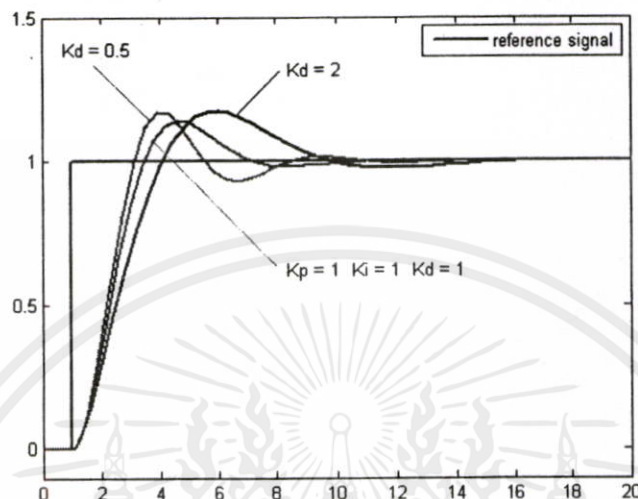
รูปที่ 2.13 ผลของ Integral Action ต่อระบบควบคุมแบบ PID

2.3.2.3 ผลของ Derivative Action (Kd) กับผลตอบสนองต่อระบบ

ผลของ Derivative Action จะเป็นสัดส่วนโดยตรงกับค่าอัตราการเปลี่ยนแปลงของอินพุต ซึ่งมีผลทำให้ระบบตอบสนองการเปลี่ยนแปลงได้เร็วขึ้น สำหรับตัวควบคุมแบบ PID นั้นค่าของ Derivative Action จะมีผลกับการเปลี่ยนแปลงของตัวแปรที่เกี่ยวข้องกับระบบ โดยจะทำให้ระบบมีการตอบสนองต่อการเปลี่ยนแปลงที่เร็วขึ้น แต่ Derivative Action จะไม่มีผลต่อ

การเปลี่ยนแปลงของ Set Point แต่อย่างไรก็ตาม Derivative Action จะมีผลเสียในการควบคุมระบบที่มีสัญญาณรบกวนมากเนื่องจาก Derivative Action จะไวต่อการเปลี่ยนแปลงไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งยังมีให้ดูเปลืองเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้ สัญญาณที่เข้ามาอย่างมาก ดังนั้นระบบที่มีสัญญาณรบกวน ตัวอย่างเช่น ระบบการควบคุมอัตราการ

ไหล หรือ ระบบความดัน Derivative Action อาจจะมีผลทำให้ระบบไม่มีเสถียรภาพ (Unstable) แสดงดังรูปที่ 2.14



รูปที่ 2.14 ผลของ Derivative Action ต่อระบบการควบคุมแบบ PID

สำหรับผลของต่างพารามิเตอร์ทั้ง 3 ตัว ที่ได้กล่าวมาแล้วนั้น จะมีการพิจารณาผลตอบสนองของระบบอย่างไม่มีความสัมพันธ์ซึ่งกันและกันหรือใช้การพิจารณาทางทฤษฎีเท่านั้น แต่ในทางปฏิบัติจริงๆแล้ว ความสัมพันธ์ของพารามิเตอร์ทั้งสามกับผลตอบสนองของระบบนั้นยังเกี่ยวกับสิ่งอื่นๆ อีก เช่น รูปแบบของการควบคุม ใช้การควบคุมเป็นลักษณะใด และยังเกี่ยวเนื่องกับสภาพแวดล้อม รวมทั้งสัญญาณรบกวนที่ระบบได้รับอีกด้วย ดังนั้นในการปรับค่าพารามิเตอร์ของตัวควบคุมโดยปกติแล้วต้องอาศัยความรู้ ความคุ้นเคยและความชำนาญกับระบบนั้น

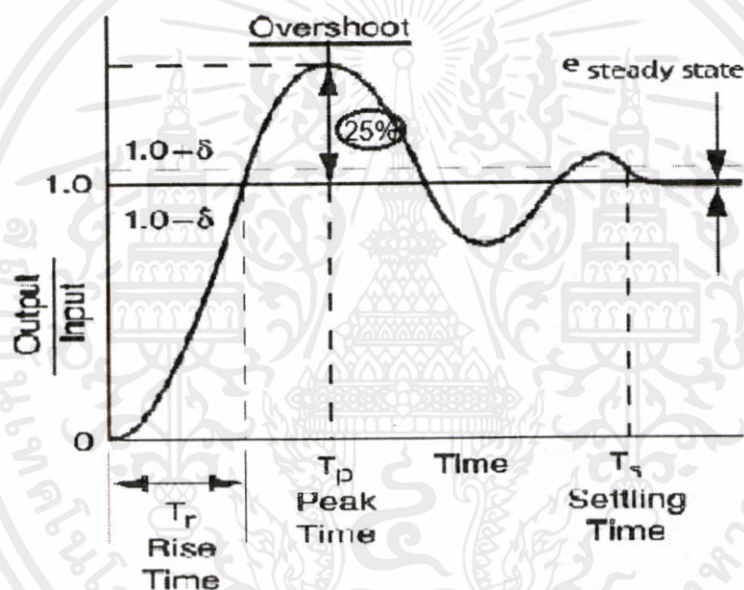
2.3.3 การคำนวณหาค่าพารามิเตอร์ของตัวควบคุมแบบพีไอดีจากผลตอบสนองของกระบวนการ

การนำตัวควบคุม PID ไปใช้ในการควบคุมกระบวนการต่างๆ จำเป็นต้องทำการปรับค่าพารามิเตอร์ของตัวควบคุม PID ให้เหมาะสม เพื่อให้ได้ผลตอบสนองที่ดีตามความต้องการซึ่งการหาค่าพารามิเตอร์ของตัวควบคุม PID นี้สามารถทำได้หลายวิธี ซึ่งบางวิธีอาจพิจารณาได้จากผลตอบสนองของกระบวนการ จากการทดสอบกระบวนการหรืออาศัยประสบการณ์จากตัวควบคุม บางวิธีอาจจะอาศัยทฤษฎีทางคณิตศาสตร์ และส่วนใหญ่จะพิจารณาจากเงื่อนไขในรูปแบบของโดเมนเวลาทั้งด้านเวลาทั้งด้านความถี่ ซึ่งวิธี Ziegler-Nichols นี้เป็นวิธีที่ได้รับความนิยมมากที่สุด การหาค่า

K_p , K_i และ K_d จะขึ้นอยู่กับคุณลักษณะของผลตอบสนองชั่วคราวของกระบวนการที่ถูกควบคุม ซึ่งมีอยู่ 3 วิธี คือ

1. วิธีวงเปิด (Open-loop step response method/ Reaction curve method)
2. วิธีวงปิด (Closed-loop method/ Ultimate sensitivity method)
3. วิธีลองผิดลองถูก (trial-and-error method)

ซึ่งแต่ละวิธีมีจุดมุ่งหมายที่จะทำให้ผลตอบสนองต่อเวลาของกระบวนการต่ออินพุตแบบขั้นบันได (Unit Step) มีค่าพุ่งเกิน (Overshoot) สูงสุดไม่เกิน 25% ดังรูปที่ 2.15



รูปที่ 2.15 แสดงค่า Overshoot ที่ยอมรับได้ในการปรับโดยใช้วิธีต่าง ๆ

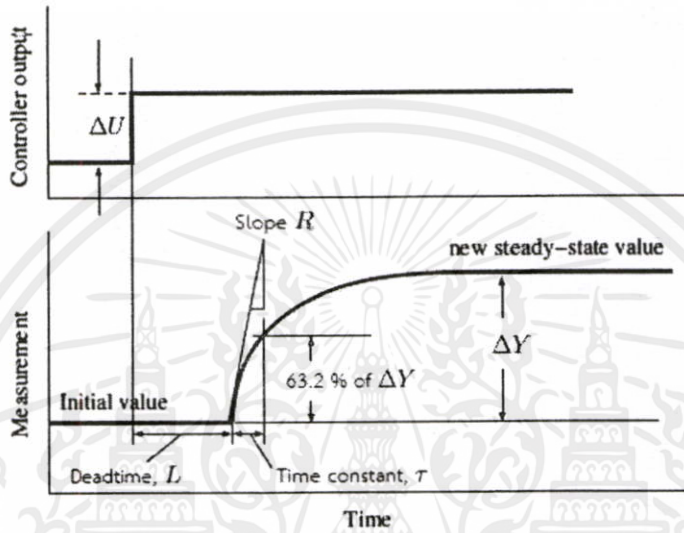
2.3.3.1 วิธี Process Reaction Curve (การทดลองแบบรูปเปิด)

วิธีนี้เริ่มมาจากความคิดที่ว่า ผลตอบสนองจากสัญญาณขั้น (Step response) ของกระบวนการส่วนใหญ่ มีลักษณะกราฟคล้ายกับในรูป 2.16 ซึ่งเรียกว่า process reaction curve รูปกราฟลักษณะดังกล่าวสามารถประมาณได้จากผลตอบสนองของสัญญาณขั้นของระบบลำดับอันดับหนึ่ง (First-order lag) ซึ่งมีฟังก์ชันถ่ายโอนเป็นดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$G(s) = \frac{Y(s)}{U(s)} = \frac{Me^{-sL}}{(\tau s + 1)}$$

นั่นก็คือเป็นระบบอันดับหนึ่งที่มี deadtime (หรือ Time delay) เท่ากับ L วินาที มีค่าคงตัวทางเวลา เท่ากับ τ วินาที และผลตอบสนองจากสัญญาณขั้นหนึ่งหน่วยจะลู่เข้าหาค่า M เมื่อ $t \rightarrow \infty$



รูปที่ 2.16 ผลตอบสนองวงเปิดของกระบวนการที่มี Deadtime และการล่าช้า

จากกราฟที่แสดงในรูปที่ 2.16 จะทำให้สามารถหาแบบจำลองทางคณิตศาสตร์ของกระบวนการได้โดยเริ่มต้นด้วยการหาค่าคงตัวทางเวลา, deadtime และอัตราการตอบสนอง (response rate) โดยอาศัยผลตอบสนองวงเปิด (Open-loop response) ขั้นตอนการทดสอบเป็นดังนี้

1. ทำให้เกิดการเปลี่ยนแปลงแบบขั้น (Step change) ด้วยขนาด ΔU ที่สัญญาณออกของตัวควบคุม (ซึ่งเป็นสัญญาณเข้าของระบบ) ในขณะเดียวกันก็หาค่า deadtime ดังรูปที่ 2.16 ค่า deadtime (หน่วยเป็นวินาที) จะเป็นระยะเวลานับจากจุดที่เริ่มการเปลี่ยนแปลงแบบขั้นของสัญญาณเข้า ไปยังจุดที่ระบบเริ่มตอบสนองต่อสัญญาณเข้านั้น
2. จากนั้นสัญญาณออกที่วัดได้จะมีค่าค่อยๆเพิ่มขึ้น จนกระทั่งเข้าสู่ภาวะอยู่ตัวที่

เอกสารนี้เป็นเอกสารที่สงวนค่าใหม่ ซึ่งเป็นผลมาจากการเปลี่ยนแปลงแบบขั้นที่สัญญาณออกของตัวควบคุม ด้านการคำนวณ ไม่ว่าจะเป็นใดๆทั้งสิ้น อีกทั้งเส้นโค้งแสดงดังรูปที่ 2.16 เป็นผลตอบสนองสำหรับระบบที่เป็น single มีการนำไปใช้

capacitance อาจประมาณได้ว่าผลตอบสนองที่มีลักษณะเป็นการล่าหลังอันดับหนึ่ง (first-order lag)

3. ค่าคงตัวทางเวลา τ หาได้จากระยะเวลาที่สัญญาณออกเริ่มเปลี่ยนแปลง ไปจนถึงค่า 63.2% ของ ΔY (หรือเป็นสัดส่วน $1 - e^{-1}$ ของ ΔY คือการเปลี่ยนแปลงของสัญญาณออก เมื่อเทียบกับค่าเดิม
4. ค่า M ซึ่งบ่งบอกถึงอัตราขยายของระบบในภาวะอยู่ตัว หาได้จาก $M = \Delta Y / \Delta U$
5. ค่าความชันของกราฟสัญญาณออก ณ inflection point จะมีค่าเท่ากับ M/τ เราจะเรียกค่าความชันนี้ว่า process reaction rate = R จากรูปที่ 2.16 สามารถหาค่าพารามิเตอร์ของพีไอดีได้จากตารางที่ 2.1

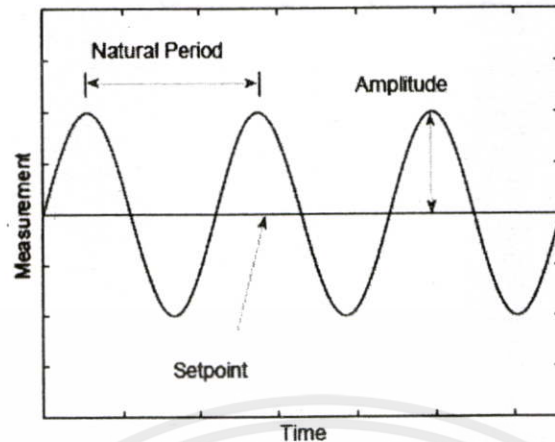
ตารางที่ 2.1 แสดงค่าพารามิเตอร์ของตัวควบคุมแบบต่างๆ ตามวิธี Process Reaction Curve

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	$\frac{1.2}{K} \left(\frac{\tau}{t_0} \right)$	$2.0t_0$	$0.5t_0$

2.3.3.2 วิธี Ultimate Method (การทดลองลูปปิด)

วิธีนี้นิยมใช้กันอย่างแพร่หลาย เพราะการทำกรวัดค่าพารามิเตอร์เพียงค่าเดียวเท่านั้น แต่มีข้อเสียเปรียบอยู่คือ กระบวนการบางอย่างที่มีลักษณะออนไลน์ (Online) ไม่สามารถจะทำให้เกิดการแกว่งได้ ถึงแม้ว่าจะเป็นช่วงเวลาเพียงสั้นๆ โดยการทำให้วงควบคุมแกว่งที่ค่าแอมพลิจูดคงที่และคาบเวลาคงที่ ทำให้สามารถหาคาบเวลาธรรมชาติ (natural period, T_0) ได้ ตัวอย่างของการวัดวิธีนี้แสดงในรูปที่ 2.17 ซึ่งมีลักษณะการแกว่งแบบไซน์ (sinusoid)

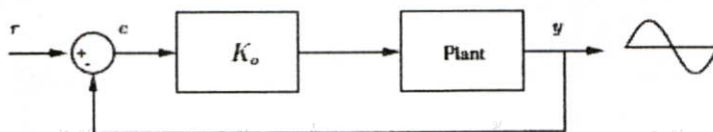
เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.17 การแกว่งด้วยแอมพลิจูดที่คงที่เมื่อปรับโดยใช้วิธี Ultimate method

เพื่อให้เกิดการแกว่งที่ค่าแอมพลิจูดคงที่ในวงควบคุมของกระบวนการจะต้องดำเนินการตามขั้นตอนดังนี้

1. จัดระบบวงปิดดังรูป 2.18
2. ตรวจสอบให้แน่ใจว่า วงควบคุมมีเสถียรภาพ
3. ปรับส่วนอินทิกรัล (I) และ/หรือ ส่วนอนุพันธ์ (D) ให้มีค่าต่ำสุด (นั่นคือ เหลือแค่ตัวควบคุมแบบ-สัดส่วน)
4. ให้การเปลี่ยนแปลงแบบขั้นที่ค่ากำหนด (r) ของค่ากำหนด (Set value) และสังเกตสัญญาณออกที่วัดได้
5. ถ้าหากสัญญาณออกค่อยๆ ลดลงสู่ค่าที่สภาวะอยู่ตัวให้ลด %PB อีก (เพิ่มอัตราขยาย K)
6. กลับไปทำขั้นที่ 3 และ 4 จนกระทั่งได้ผลตอบสนองที่แกว่งด้วยแอมพลิจูดและคาบเวลาคงที่ ดังแสดงในรูปที่ 2.17 (K_0 คืออัตราขยายที่ทำให้ผลตอบสนองแกว่งอย่างต่อเนื่อง)



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 รูปที่ 2.18 การหาค่าอัตราขยายที่ทำให้ระบบเริ่มแกว่ง
 ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.2 แสดงค่าพารามิเตอร์ของตัวควบคุมแบบต่างๆ ตามวิธี Ultimate Method

Type of Controller	K_p	T_i	T_d
P	$\frac{K_0}{2}$	-	-
PI	$\frac{K_0}{2.2}$	$\frac{t_0}{1.2}$	-
PID	$\frac{K_0}{1.7}$	$\frac{t_0}{2}$	$\frac{t_0}{8}$

หลังจากการวัดค่าคาบเวลา และ/หรือ deadtime และค่าคงตัวเวลาแล้ว การปรับตั้งตัวควบคุม (Controller Setting) PID โดยวิธีของ Ziegler-Nichols นั้นไม่ใช่เป็นค่าที่เที่ยงตรงที่จะนำไปใช้งานได้ทันที วัตถุประสงค์เพื่อค่าหาพารามิเตอร์ที่ทำให้ผลรวมของความคลาดเคลื่อนในช่วงเวลาทำงานมีค่าต่ำสุด ผู้ควบคุมจำเป็นต้องทำการปรับค่าพารามิเตอร์เหล่านี้แบบละเอียด (Fine Tuning) อีกครั้งหนึ่ง

2.3.3.3 วิธี Trial and Error (วิธีการลองผิดลองถูก)

เป็นวิธีการที่เหมาะสมสำหรับผู้ที่มีประสบการณ์ในด้านปรับค่าพารามิเตอร์ PID แล้ว โดยการควบคุมจะต้องเป็นระบบปิดและตำแหน่งการควบคุมจะต้องอยู่ในตำแหน่ง Automatic Control โดยขั้นตอนในการทดสอบและปรับจูนค่าเกณฑ์ของตัวควบคุมพีไอดีสำหรับระบบไม่ทราบแบบจำลองทางคณิตศาสตร์ มีขั้นตอนดังนี้

- 1) ต่อระบบควบคุมวงรอบปิดด้วยตัวควบคุมแบบพี
- 2) ปรับค่าเกณฑ์ให้สูงขึ้นเรื่อยๆ สังเกตผลตอบสนอง
- 3) ถ้าปรับค่าเกณฑ์ให้สูงขึ้นเรื่อยๆ แล้ว ผลตอบสนองที่ได้ไม่มีการพุ่งเกิน (ระบบมีอันดับ-เป็นหนึ่ง) ให้สังเกตดูว่าระบบมีค่าความผิดพลาดในสภาวะคงตัวหรือไม่

3.1) ถ้าไม่มีค่าความผิดพลาดในสภาวะคงตัวก็ใช้ตัวควบคุมแบบพีอย่าง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ใดๆ โดยปรับค่าเกณฑ์ของตัวควบคุมให้สูงขึ้นเรื่อยๆ
 ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแบบส่งเนื้อหา และต้องอ้างอิงถึงเจ้าของลิขสิทธิ์ทุกครั้งที่มีการนำไปใช้

- 3.2) ถ้ามีค่าความผิดพลาดในสภาวะคงตัวก็ให้ใช้ตัวควบคุมแบบพีไอ โดยปรับค่าเกนของตัวควบคุมพีให้สูงพอประมาณ แล้วจึงปรับค่าเกนของตัวควบคุมไอจนกระทั่ง ค่าความผิดพลาดในสภาวะคงตัวเป็นศูนย์
- 4) ถ้าผลตอบสนองที่ได้มีการพุ่งเกินค่าที่กำหนดไว้
 - 4.1) ถ้าต้องการลดการพุ่งเกินของผลตอบสนองอย่างเดียวโดยยอมรับค่าความผิดพลาดในสภาวะคงตัวได้ ก็ให้ใช้ตัวควบคุมแบบพีดี โดยปรับค่าเกนพีให้สูงพอประมาณ จนกระทั่งได้ค่าความผิดพลาดในสภาวะคงตัวเป็นที่พอใจ แล้วจึงปรับค่าเกนของตัวควบคุมแบบดีเพื่อลดการพุ่งเกินของผลตอบสนอง
 - 4.2) ถ้าต้องการปรับทั้งความเร็วและการพุ่งเกินของผลตอบสนอง รวมไปถึงค่าความผิดพลาดในสภาวะคงตัว ก็ให้ใช้ตัวควบคุมแบบพีไอดีซึ่งอัตราส่วนระหว่างค่า Integral Time กับ Derivative Time เป็น 4/1 เป็นต้น ไปพร้อมการเพิ่มลดของค่าเกน-พีไปด้วย เพื่อป้องกันการเกิด Overshoot และการเกิด Hunting
 - 4.3) เนื่องด้วยวิธีการนี้เป็นความพึงพอใจเฉพาะบุคคลจึงยากที่จะกำหนดเป็นเกณฑ์ตายตัวแต่วิธีการเริ่มต้นปรับนั้นมักเหมือนกัน

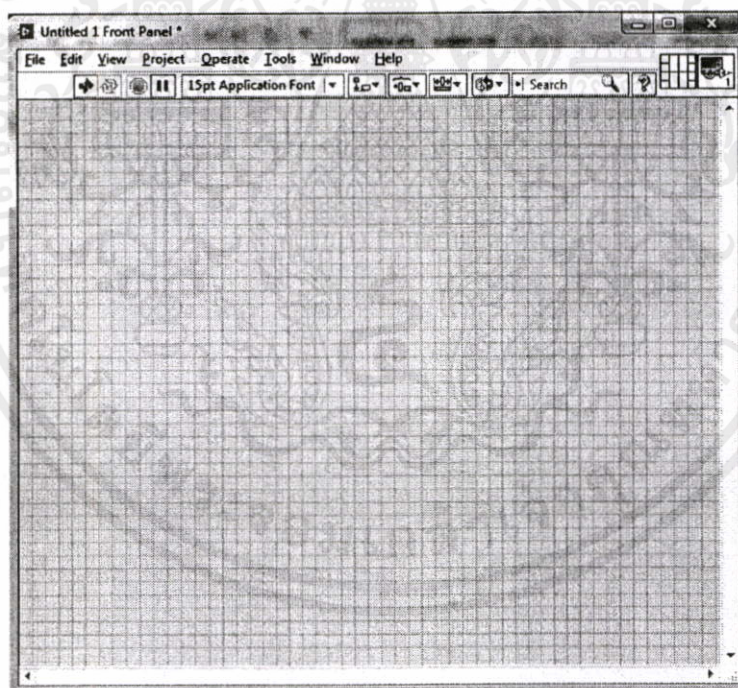
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4 LabVIEW

ปัจจุบันจะเห็นได้ว่า LabVIEW เริ่มเป็นที่รู้จักและมีบทบาทในกระบวนการอุตสาหกรรมหลายสาขา เนื่องจาก LabVIEW เป็นโปรแกรมที่ใช้งานง่าย และมีประสิทธิภาพสูง อีกทั้งการพัฒนาและการแก้ไขโปรแกรมสามารถทำได้ในเวลาอันสั้น โดยส่วนประกอบของ LabVIEW จะแบ่งออกเป็น 2 ส่วนคือ

2.4.1 Front Panel

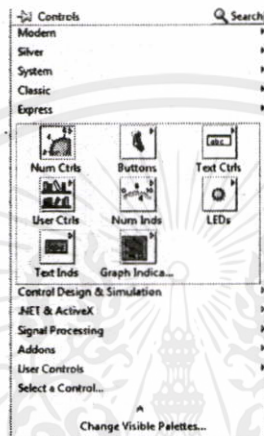
เป็นส่วนของหน้าต่างที่ทำหน้าที่เป็นส่วนติดต่อกับ user หรือเป็นส่วนของ Virtual Instrument Front Panel ประกอบไปด้วย Control ทำหน้าที่เป็น input และ Indicator ทำหน้าที่เป็น output ซึ่งทำหน้าที่ติดต่อกับผู้ใช้งาน ในขณะที่เราวาง Indicator และ Control ลงบน Front Panel โปรแกรม LabVIEW จะทำการสร้าง Control และ Indicator ลงบน Block Diagram อัตโนมัติ



รูปที่ 2.19 หน้าต่างของ Front Panel

หลังจากเราเปิดโปรแกรม LabVIEW ขึ้นมาแล้ว เราสามารถที่จะสร้าง new VI ขึ้นมา ได้ หน้าต่าง Front Panel จะปรากฏขึ้นมา โดยเราสามารถที่จะทำการสร้างหน้าต่าง User Interface หรือหน้าต่างโปรแกรมของเรา เพื่อให้ผู้ใช้งานได้ตามที่เรำได้ออกแบบไว้แล้ว

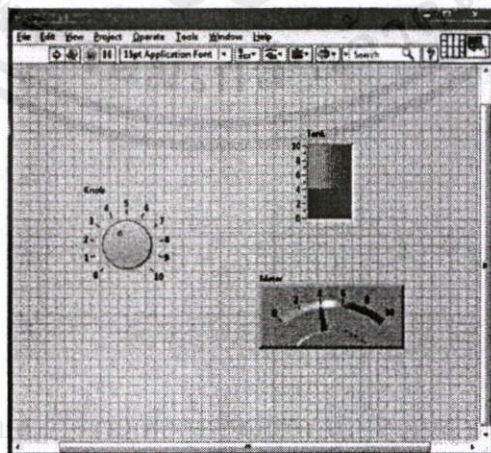
ที่หน้าต่างของ Front Panel ประกอบไปด้วยส่วนของ Toolbar ที่อยู่ด้านบน และส่วนของ Control Palette ซึ่งเราสามารถที่จะเปิดตรงไหนก็ได้ที่หน้าต่าง Front Panel โดยทำการคลิกขวาที่หน้าต่าง Front Panel หลังจากเราเปิดหน้าต่าง Control Palette แล้วเราก็จะพบกับ Control และ Indicator แล้วสามารถที่ลากลงบนหน้าต่าง Front Panel ได้ทันที



รูปที่ 2.20 หน้าต่างของ Control Palette

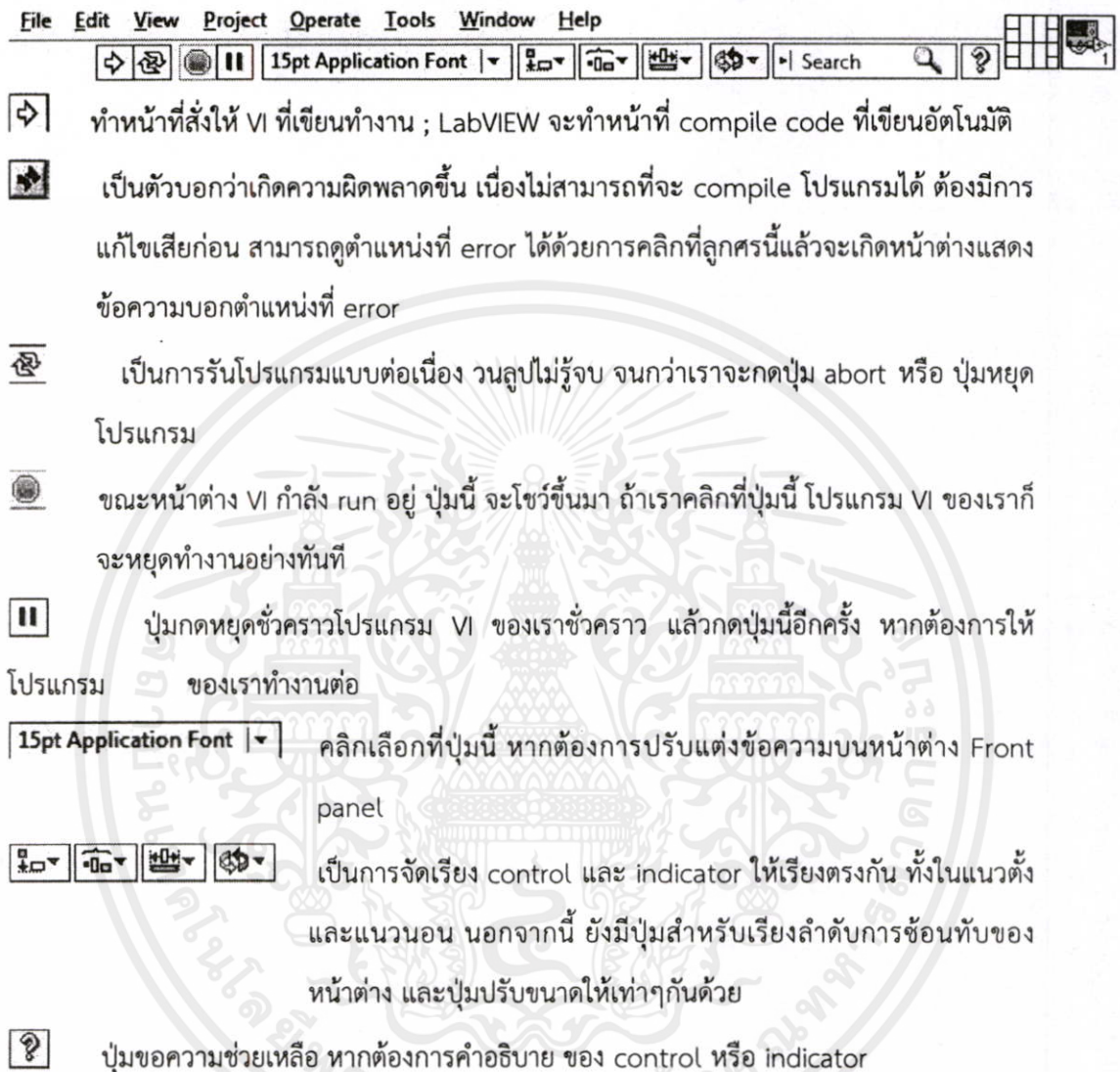
2.4.1.1 Control and Indicator

Control ได้แก่ Knob , Push button , dial , และอุปกรณ์ที่ทำหน้าที่เป็น input อื่นๆ ทำหน้าที่เป็นตัวโต้ตอบ ทำหน้าที่เป็นตัวรับ ในขณะที่ Indicator ซึ่งได้แก่ graphic , LED ,หรืออุปกรณ์ที่ทำหน้าที่เป็นตัวแสดงผล ทำหน้าที่เป็นผลลัพธ์ ของหน้าต่าง VI






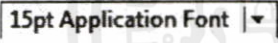




เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับ... หน้าไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น รูปที่ 2.21 หน้าต่างของ Front Panel แสดงตัว Control และ Indicator... รั้งที่มีการนำไปใช้

2.4.1.2 Front Panel Window Toolbar



The screenshot shows the LabVIEW Front Panel Window Toolbar with the following icons and their descriptions:

-  ทำหน้าที่สั่งให้ VI ที่เขียนทำงาน ; LabVIEW จะทำหน้าที่ compile code ที่เขียนอัตโนมัติ
-  เป็นตัวบอกว่าเกิดความผิดพลาดขึ้น เนื่องจากไม่สามารถที่จะ compile โปรแกรมได้ ต้องมีการแก้ไขเสียก่อน สามารถดูตำแหน่งที่ error ได้ด้วยการคลิกที่ลูกศรนี้แล้วจะเกิดหน้าต่างแสดงข้อความบอกตำแหน่งที่ error
-  เป็นการรันโปรแกรมแบบต่อเนื่อง วนลูปไม่รู้จบ จนกว่าเราจะกดปุ่ม abort หรือ ปุ่มหยุดโปรแกรม
-  ขณะหน้าต่าง VI กำลัง run อยู่ ปุ่มนี้ จะโชว์ขึ้นมา ถ้าเราคลิกที่ปุ่มนี้ โปรแกรม VI ของเราก็จะหยุดทำงานอย่างทันที
-  ปุ่มกดหยุดชั่วคราวโปรแกรม VI ของเราชั่วคราว แล้วกดปุ่มนี้อีกครั้ง หากต้องการให้โปรแกรมของเราทำงานต่อ
-  15pt Application Font | ▾ คลิกเลือกที่ปุ่มนี้ หากต้องการปรับแต่งข้อความบนหน้าต่าง Front panel
-  เป็นการจัดเรียง control และ indicator ให้เรียงตรงกัน ทั้งในแนวตั้งและแนวนอน นอกจากนี้ ยังมีปุ่มสำหรับเรียงลำดับการซ้อนทับของหน้าต่าง และปุ่มปรับขนาดให้เท่าๆกันด้วย
-  ปุ่มขอความช่วยเหลือ หากต้องการคำอธิบาย ของ control หรือ indicator

2.4.2 Block Diagram ทำหน้าที่เสมือนเป็น Sourcecode โดยใช้โปรแกรมภาษารูปภาพฟังก์ชันประกอบของ block diagram ในเบื้องต้นจะมีอยู่ 6 ประเภทหลัก คือ

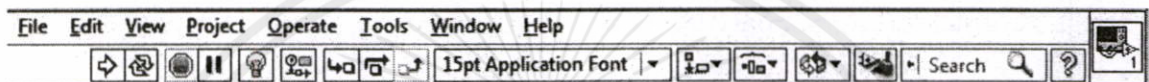
1. Terminal คือไอคอนที่เกิดมาจากการสร้าง Control/Indicator ซึ่งจะเป็นจุดเริ่ม (source) หรือ จุดสิ้นสุด (sink) ของข้อมูล
2. Constant คือ source ของข้อมูลที่เป็นค่าคงที่ไม่สามารถรับค่าจากผู้ใช้ได้ในขณะที่





เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง หากมีการนำเอกสารนี้ไปใช้โดยไม่ได้รับอนุญาตถือว่าผิดกฎหมาย

3. Node คือบล็อกที่มีการประมวลผลหรือคำนวณอยู่ภายใน

4. Wire คือสายที่ใช้สำหรับส่งผ่านข้อมูลจากจุดหนึ่งไปยังอีกจุดหนึ่ง
5. Structure คือกรอบที่สามารถล้อมโค้ดไว้ได้ ซึ่งเป็นส่วนหนึ่งของการเขียนโปรแกรม ยกตัวอย่างเช่น While Loop, For Loop, Case Structure, Event Structure เป็นต้น
6. Decoration คือส่วนกราฟิกที่ใช้จัดระเบียบ หรือ อาจจะเป็น Free Label สำหรับเขียนอธิบายการทำงานของโค้ด ซึ่งจะไม่มีผลต่อการรันโปรแกรม

2.4.2.1 Block Diagram Window Toolbar



-  ถ้ากดปุ่มหลอดไฟจะติด การรันโปรแกรมจะช้ากว่าปกติและจะมีจุดของข้อมูลไหลตามสายให้ดูที่ละจุด
-  จะคงข้อมูลล่าสุดที่สายแต่ละเส้นไว้ เพื่อให้สามารถ Probe สายตรวจสอบข้อมูลหลังจบการทำงาน
-  เครื่องมือสำหรับการ Debug โปรแกรม
-  สำหรับจัดระเบียบ object ทุกอย่างที่อยู่บน Block Diagram แบบอัตโนมัติ

ถ้าหากคลิกขวาที่พื้นที่ว่างของ Block Diagram ก็จะได้อุปกรณ์อีกชุดขึ้นมาที่เรียกว่า Functions Palette ซึ่งเป็นเครื่องมือสำหรับการเขียนโปรแกรม การคำนวณและประมวลผลต่างๆ ซึ่งเป็น สัญลักษณ์ของโค้ดกราฟิกทั้งหมด สังเกตที่มุมขวาบนจะมีปุ่ม Search ซึ่งเราสามารถใช้ค้นหาเครื่องมือต่างๆได้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้นำไปเผยแพร่ในที่สาธารณะหรือสื่อออนไลน์ใดๆทั้งสิ้น
รูปที่ 2.22 หน้าต่างของ Functions Palette

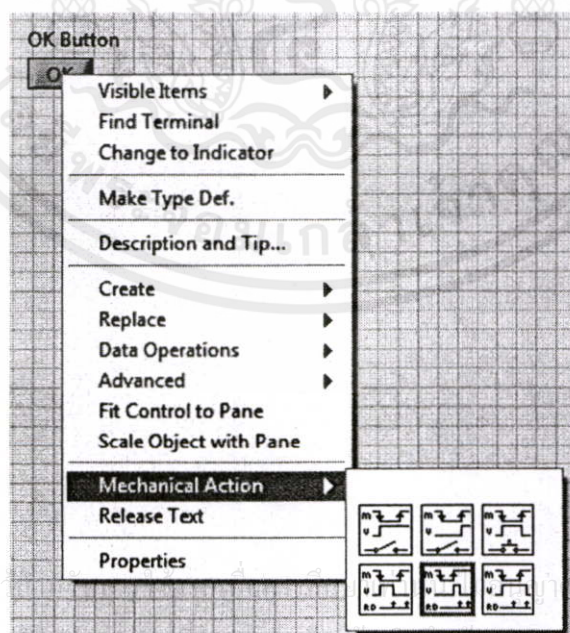
2.4.3 ประเภทของข้อมูล (Data Type)

ในการเขียนโปรแกรมทั่วไปจะต้องมีการประกาศตัวแปร (declare) ก่อนที่จะใช้ตัวแปรนั้น สำหรับ LabVIEW เราใช้วิธีเลือกประเภทข้อมูลมาวางบนโค้ด ประเภทข้อมูลใน LabVIEW มีหลายอย่างที่ใช้เหมือนกับในภาษาอื่นๆ ยกตัวอย่างเบื้องต้น คือ

2.4.3.1 Numeric คือข้อมูลประเภทตัวเลข เมื่อเราสร้าง Numeric Control/Indicator/Constant ขึ้นมา ค่าเริ่มต้น (default) ของมันจะมีค่าเป็นศูนย์ ข้อมูลรูปแบบ Numeric มีทั้งจำนวนเต็มซึ่งไอคอนและสายใน Block Diagram จะเป็นสีน้ำเงิน และจำนวนทศนิยม ซึ่งจะแสดงเป็นสีส้ม วิธีการเปลี่ยนประเภทของตัวเลข ให้คลิกขวาที่ Numeric แล้วเลือก Representation แล้วสามารถเลือกประเภทของตัวเลขได้ทันที

สำหรับผู้ที่ไม่เคยเขียนโปรแกรมมาก่อน เบื้องต้นแนะนำว่าถ้าต้องการใช้ตัวเลขแบบมีทศนิยม ควรเลือกแบบ DBL (Double Precision) ถ้าต้องการจำนวนเต็มให้เลือก L32 (Long)

2.4.3.2 Boolean คือข้อมูลประเภทที่มีสองค่า คือ True และ False โดยที่ค่า Default เดิมคือ False สำหรับ Block Diagram จะแสดงสีของไอคอนและสายของข้อมูลประเภทนี้ด้วยสีเขียว และสำหรับบน Front Panel ตัว Boolean Control จะมีคุณสมบัติเป็นสวิตช์ (Mechanical Action) ซึ่งมีหลายประเภท วิธีเปลี่ยนชนิดของสวิตช์ให้คลิกขวาที่ Boolean Control บน Front Panel แล้วเลือก Mechanical Action ซึ่งจะมียู่ 6 แบบ



เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี ห้ามทำซ้ำโดยไม่ได้รับอนุญาต การนำภาพไปใช้โดยไม่ได้รับอนุญาตจะถือว่าผิดกฎหมาย

รูปที่ 2.23 หน้าต่างของ Mechanical Action

แต่ละ Mechanical Action จะมีผลดังนี้

- Switch When Pressed คือสวิตช์กดติด-กดดับ
- Switch When Released คือกดติด-กดดับ แต่จะมีผลเมื่อปล่อยมือแล้ว
- Switch Until Released คือกดติด-ปล่อยดับ
- Latch When Pressed คือสวิตช์จะเปลี่ยนค่าทันทีเมื่อคลิกแล้วจะดังกลับมาเป็นค่าเดิมเมื่อโปรแกรมรับรู้
- Latch When Released คือหลังจากกดแล้วสวิตช์จะเปลี่ยนค่าเมื่อปล่อยมือแล้วจะดังกลับเมื่อโปรแกรมรับรู้
- Latch Until Released คือคล้ายกับกดติด-ปล่อยดับ แต่จะมีการรอให้โปรแกรมอ่านค่าตอนปล่อยมือก่อน

2.4.3.3 String คือข้อมูลที่เป็นตัวอักษร โดยที่ค่า default คือว่างเปล่า (empty string) ไอคอนและสายของ String ใน Block Diagram จะเป็นสีชมพู สำหรับการแสดงผลของ String บน Front Panel และ Block Diagram จะมีอยู่ 4 แบบ

- Normal Display คือการแสดงผลปกติ
- '\ Code Display คือการแสดงผลแบบ \ โค้ด มีประโยชน์สำหรับป้อนตัวอักษรบางตัวที่เราพิมพ์โดยตรงไม่ได้ หรือแสดงตัวอักษรที่เรามองไม่เห็น เช่น การเว้นวรรค (\s) แท็บ (\t) หรือตัวขึ้นบรรทัดใหม่ (\n) เป็นต้น
- Password Display จะแทนตัวอักษรด้วยเครื่องหมายดอกจัน (*)
- Hex display แสดงเป็นรหัส ASCII ด้วยเลขฐานสิบหก

การเปลี่ยนรูปแบบการแสดงผลของ String Control/Indicator ให้คลิกขวาที่ String บน Front Panel แล้วเลือกเปลี่ยนเป็นแบบที่ต้องการจากเมนูได้ทันที

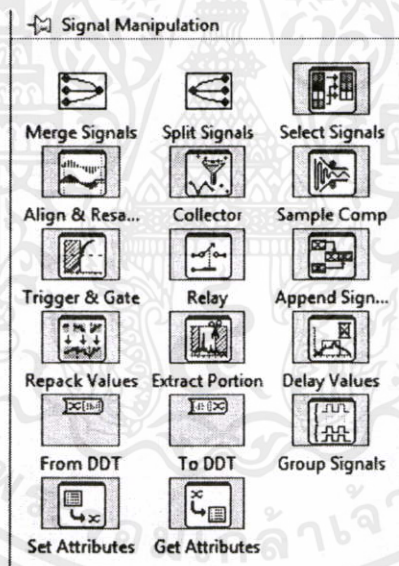
2.4.3.4 Enum คือข้อมูลประเภทที่แสดงให้ผู้เห็นเป็นตัวหนังสือ แต่ค่าจริงของมันคือเลขจำนวนเต็มดังนั้นบน Block Diagram เราจะมองเห็น Terminal และสายของข้อมูลเป็นสีน้ำเงิน

เอกสารที่ ซึ่งเหมือนกับจำนวนเต็ม วิธีการสร้าง Enum Control ให้เลือกมาจาก Controls Palette : Modern ด้านการตั้งค่า
ไม่ว่ากรณี >> Ring&Enum >> Enum แล้ววางลงบน Front Panel แล้วคลิกขวาและเลือกที่ Edit Items...ไปใช้

จากนั้นที่หน้าต่าง Enum Properties ก็ให้พิมพ์ตัวหนังสือตรงคอลัมน์ซ้าย (Items) ซึ่งจะไปจับคู่กับตัวเลขทางคอลัมน์ขวา Digital Display (Enum มีประโยชน์สำหรับการสร้าง Control ที่ต้องการกำจัด input ไม่ให้ผู้ใช้ใส่ข้อมูลอย่างอื่นนอกจากที่กำหนดไว้ก่อนใน Enum เท่านั้น)

2.4.3.5 Dynamic (DDT) เป็นข้อมูลที่อยู่ในรูปของสัญญาณเวฟฟอร์ม สายบน Block Diagram ถูกแสดงด้วยสีน้ำเงินเข้ม เส้นใหญ่ ซึ่งภายในประกอบด้วยข้อมูลหลายอย่าง เช่น Array ของเวฟฟอร์ม, Time, Stamp, ชื่อของสัญญาณ ฯลฯ ข้อมูลประเภท DDT นี้ส่วนใหญ่ใช้ใน Express VI สำหรับการอ่าน, กำเนิด และวิเคราะห์สัญญาณเป็นต้น

สายข้อมูลแบบ DDT สามารถส่งข้อมูลหลายๆ Chanel ได้ในเส้นเดียวโดยการ merge สัญญาณหลายช่องเข้าด้วยกัน เครื่องมือต่าง ๆ สำหรับการจัดการตัดแปลงแก้ไขข้อมูลแบบ DDT จะอยู่ใน Functions Palette : Express >> Signal Manipulation



รูปที่ 2.24 หน้าต่างของ Signal Manipulation

ยกตัวอย่างเครื่องมือใน Signal Manipulation ที่ใช้บ่อย

- Merge Signals ใช้ในการรวมสัญญาณหลาย Chanel เข้าไว้ในเส้นเดียวกัน
- Split Signals ใช้ในการแยกสัญญาณ
- Sample Compression ใช้สำหรับลดปริมาณจำนวน sample ของสัญญาณเวฟฟอร์ม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับอ่านใช้งานเพื่อการศึกษาค้นคว้าเท่านั้น ไม่สามารถนำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้นำไปเผยแพร่หรือทำซ้ำ และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- From DDT ใช้เปลี่ยน DDT ให้เป็นข้อมูลแบบปกติ เช่น เปลี่ยนเป็น Numeric, Array
- To DDT ใช้แปลงข้อมูลแบบปกติให้เป็น DDT เพื่อนำกลับไปใช้ใน Express VI

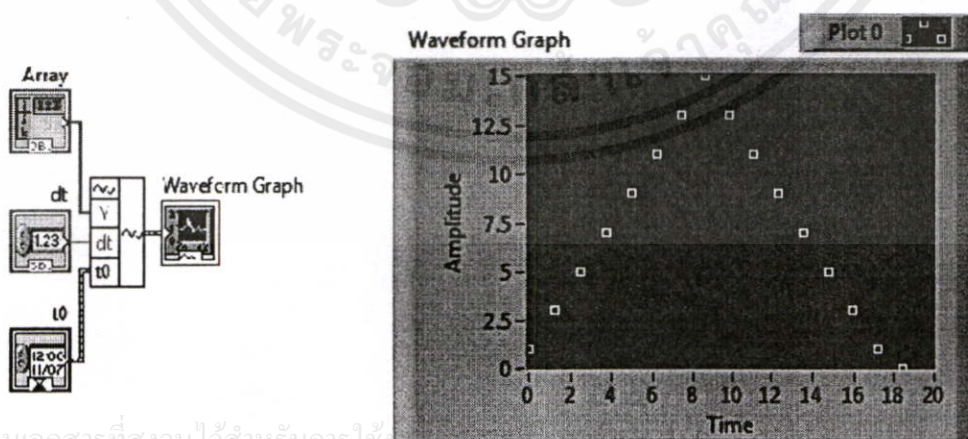
2.4.3.6 Time Stamp เป็นข้อมูลที่ประกอบด้วยวันที่และเวลาที่มีความละเอียดถึงมิลลิวินาที LabVIEW คำนวณ Time Stamp โดยนับเป็นจำนวนวินาทีที่เริ่มตั้งแต่เที่ยงคืนวันที่ 1 มกราคม ค.ศ. 1904 ในเวลามาตรฐาน หรือ UTC (เวลาในประเทศไทยเร็วกว่า UTC 7 ชั่วโมง) แล้วนำมาแปลงเป็นรูปแบบวันที่และเวลา ยกตัวอย่างเช่น ถ้าแปลงตัวเลข 3.4 พันล้านด้วยฟังก์ชัน To Time Stamp วิธีคิดคือให้เริ่มนับจากเที่ยงคืน UTC ไป 3.4 พันล้านวินาที ผลลัพธ์ก็จะเป็นวันที่ 28 กันยายน ค.ศ. 2011 เวลาในประเทศไทย 3:26:40 นาฬิกา

2.4.3.7 Waveform เป็นข้อมูลที่ประกอบด้วยข้อมูลหลัก 3 อย่าง คือ

Y : จุดของตัวเลขหลายๆจุดประกอบเรียงกันเป็นเวฟฟอร์ม ซึ่งเรียกว่า Array

dt : ข้อมูลที่ระบุว่าแต่ละจุดมีเวลาห่างกันกี่วินาที

t0 : ข้อมูลแบบ Time Stamp ที่ระบุว่าจุดแรกของสัญญาณนี้ถูกสร้างขึ้นเมื่อเวลาใด ซึ่งหมายความว่าจุดข้อมูลทุกจุดจะสามารถหา Time Stamp ของมันได้ด้วยการคำนวณจาก t0, dt, และลำดับที่จุด (Index) บน Array Y

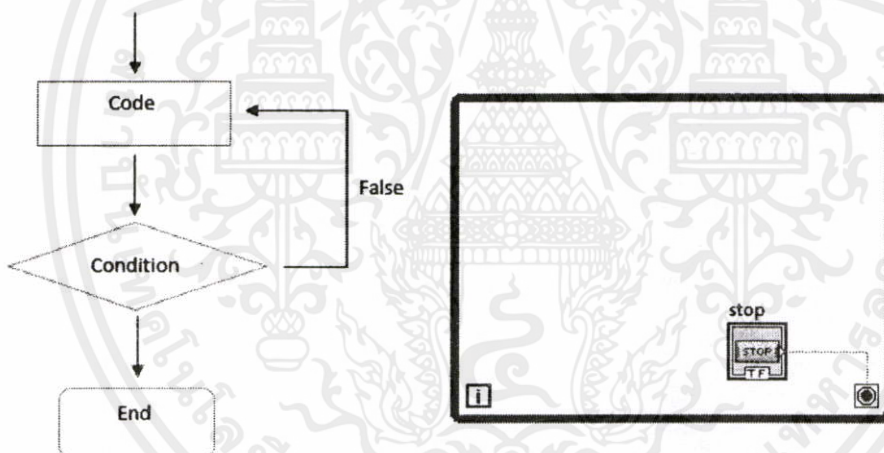


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้เฉพาะภายในเท่านั้น ห้ามเผยแพร่หรือใช้ซ้ำโดยไม่ได้รับอนุญาต
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
รูปที่ 2.25 กราฟ Waveform

2.4.4 ลูป (Loop)

การเขียนโค้ดให้มีการทำซ้ำหรือวนลูปเป็นสิ่งที่ใช้แทบทุกโปรแกรม เนื่องจากการประมวลผลบางอย่างจำเป็นต้องทำในลักษณะซ้ำๆ สำหรับ LabVIEW ลูปคือกรอบสี่เหลี่ยมที่ล้อมรอบโค้ด ส่วนที่จะรันซ้ำเอาไว้ มีสองประเภทการใช้งานดังนี้

2.4.4.1 While Loop เป็นกรอบหนา ประกอบด้วย Iteration Terminal ซึ่งถ้าเราต่อกับ Numeric Indicator จะส่งค่าออกมาบอกว่า While Loop วิ่งมากี่ครั้งแล้วโดยเริ่มนับจากศูนย์ และ While Loop จะรันจนกว่า Terminal เงื่อนไข จะได้รับ Boolean ค่า True จึงจะจบลูป ซึ่งเงื่อนไขตัวกลมสีแดงนี้เรียกว่า Stop If True ก็คือหยุดเมื่อได้รับค่า True เป็นไปตามโฟลว์ชาร์ตดังนี้



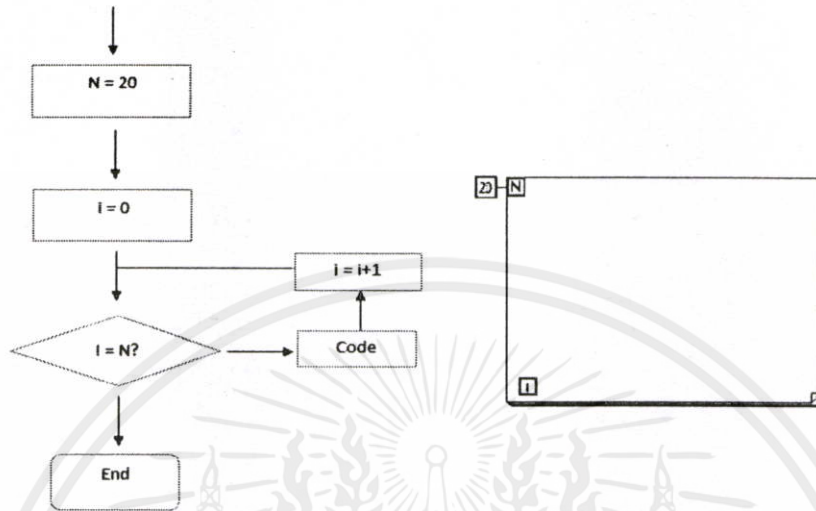
รูปที่ 2.26 แสดง Flow Chart ของ While Loop

การสร้าง While Loop ทำได้โดยเลือกจาก Functions Palette : Programming >> Structures >> While Loop จากนั้นให้นำเมาส์มาลากติกรอบสี่เหลี่ยมตามขนาดที่ต้องการ

2.4.4.2 For Loop

มีไว้สำหรับการรันลูปที่รู้แน่นอนว่าต้องการรันทั้งหมดกี่รอบ โดยเราจำเป็นต้องระบุจำนวนครั้ง (N) ของ For Loop ไว้ก่อน มิฉะนั้นโค้ดจะไม่ทำงาน ส่วนตัวไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

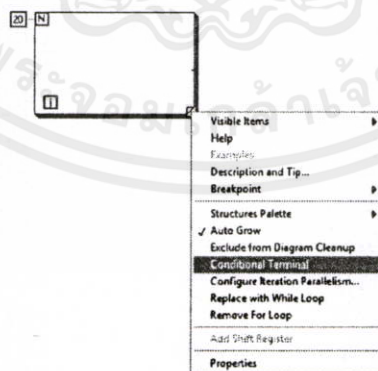
Iteration จะเหมือนกับ While Loop คือให้ผลเป็นตัวเลขแสดงจำนวนลูปที่รันไปแล้วโดยเริ่มนับจากศูนย์ ซึ่งไกลไกใน For Loop เป็นไปดั่งโฟลว์ชาร์ตดังนี้



รูปที่ 2.27 แสดง Flow Chart ของ For Loop

จะเห็นได้ว่าโค้ดใน For Loop มีโอกาสที่จะไม่ถูกรันเลยถ้า N มีค่าเป็นศูนย์ และ For Loop จะมีเงื่อนไขเดียวเท่านั้นที่จะจบได้ คือเมื่อรับครบ N ครั้งเท่านั้น

For Loop ยังสามารถเพิ่มเงื่อนไขการจบให้เหมือนกับ While Loop ได้อีกด้วย โดยการคลิกขวาที่ขอบของ For Loop แล้วเลือก Conditional Terminal



รูปที่ 2.28 วิธีการสร้างเงื่อนไขให้ For Loop

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.5 ทฤษฎีเบื้องต้นที่ใช้ในการเขียนโปรแกรมภาษาซีของ Arduino

2.5.1 โครงสร้างการเขียนโปรแกรม ภาษาซี ของ Arduino

ภาษาซีของ Arduino จะจัดแบ่งรูปแบบโครงสร้างของการเขียนโปรแกรมออกเป็น ส่วนย่อยๆ หลายๆ ส่วน โดยเรียกแต่ละส่วนว่า ฟังก์ชัน และ เมื่อนำฟังก์ชัน มารวมเข้าด้วยกัน ก็ จะเรียกว่าโปรแกรม โดยโครงสร้างการเขียนโปรแกรมของ Arduino นั้นทุกๆ โปรแกรมจะต้องประกอบ ไปด้วยฟังก์ชันจำนวนเท่าใดก็ได้ แต่อย่างน้อยที่สุดต้องมีฟังก์ชัน จำนวน 2 ฟังก์ชัน คือ setup() และ loop() ดังตัวอย่าง

```
#include <Servo.h>           //สั่งผนวกไฟล์ ชื่อ Servo.h เข้ามาในโปรแกรม
Servo myservo;              //สร้าง object ชื่อ myservo เพื่อควบคุม Servo
int potpin = 0;             //อนาล็อกพินใช้เชื่อมต่อกับ Potentiometer
int val;                    //ตัวแปรเพื่ออ่านค่าจากอนาล็อกพิน

void setup()
{
  myservo.attach(9);        //กำหนดให้ใช้ขา Digital-9 สร้างสัญญาณควบคุม Servo
}

void loop()
{
  val = analogRead(potpin); //อ่านค่าจาก Potentiometer (ค่าระหว่าง 0 – 1023)
  val = map(val, 0, 1023, 0, 179); //ปรับสเกลของมันเพื่อใช้กับ Servo (ค่าระหว่าง 0 – 180)
  myservo.write(val);       //กำหนดค่าตำแหน่งของ Servo ตามค่าสเกล
  delay(15);                //รอให้ Servo ทำงาน
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จะเห็นได้ว่าโครงสร้างพื้นฐานของภาษาซีที่ใช้กับ Arduino นั้น จะประกอบไปด้วย 3 ส่วนใหญ่ๆ ด้วยกัน คือ

- **Header** ในส่วนนี้จะมีหรือไม่มีก็ได้ ถ้ามีต้องกำหนดไว้ส่วนเริ่มต้นของโปรแกรม ซึ่งส่วนของ Header ได้แก่ ส่วนที่เป็น Compiler Directive ต่างๆ รวมไปถึงส่วนของ การประกาศตัวแปร และค่าคงที่ต่างๆ ที่จะใช้ในโปรแกรม
- **Setup()** ในส่วนนี้เป็นฟังก์ชันบังคับที่ต้องกำหนดให้มีในทุกๆ โปรแกรม ถึงแม้ว่าในบางโปรแกรมจะไม่ต้องการใช้งานก็ยังคงจำเป็นต้องประกาศไว้ด้วยเสมอ เพียงแต่ไม่ต้องเขียนคำสั่งใดๆไว้ในระหว่างวงเล็บปีกกา { } ที่ใช้เป็นตัวกำหนดขอบเขตฟังก์ชัน โดยฟังก์ชันนี้จะใช้สำหรับบรรจุคำสั่งในส่วนที่ต้องการให้โปรแกรมทำงานเพียงรอบเดียวตอนเริ่มทำงานของโปรแกรมครั้งแรกเท่านั้น ซึ่งได้แก่คำสั่งเกี่ยวกับการ Setup ค่าการทำงานต่างๆ เช่น การกำหนดหน้าที่การใช้งานของ PinMode และการกำหนดค่าBaudrate สำหรับใช้งานพอร์ตสื่อสารอนุกรมเป็นต้น
- **Loop()** เป็นส่วนฟังก์ชันบังคับที่ต้องกำหนดให้มีทุกๆโปรแกรมเช่นเดียวกับฟังก์ชัน setup() โดยฟังก์ชัน loop() นี้จะใช้บรรจุคำสั่งที่ต้องการให้โปรแกรมทำงานเป็นวงรอบซ้ำๆกันไม่รู้จบ

#Header

```
#include<Servo.h>
Servo myservo;
int potpin=0;
int val;
```

Setup()

```
Setup()
{
myservo.attach(9);
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Loop()

```

Loop()
{
  val = analogRead(potpin);
  val = map(val, 0, 1023, 0, 179);
  myservo.write(val);
  delay(15);
}

```

แสดงโครงสร้างโปรแกรมของ Arduino

2.5.2 ประเภทและชนิดของตัวแปร

เราจะต้องรู้จักคุณสมบัติของสิ่งของที่ต้องการจะจัดเก็บ และจุดประสงค์การใช้งาน ก่อน จากนั้นจึงจัดหาภาชนะที่มี ขนาด และ รูปร่างของภาชนะ เหมาะสมที่จะใช้เก็บสิ่งของเพื่อให้สามารถจัดเก็บ และ นำสิ่งของออกมาใช้งาน ได้อย่างง่ายประหยัดมากที่สุด

ในภาษาซีนั้น มีการกำหนดและจำแนก ชนิดของตัวแปรไว้เป็น 5 ชนิดด้วยกัน โดยแต่ละชนิดจะมีคุณสมบัติการใช้งานที่ต่างกัน เพื่อใช้ในการเก็บข้อมูลที่มีรูปแบบแตกต่างกัน คือ

- char ใช้เก็บข้อมูลที่เป็นตัวอักษร (Character) ใช้เก็บข้อมูลที่เป็นเลขจำนวนเต็มได้ 256 ค่า
- int ใช้เก็บข้อมูลที่เป็นเลขจำนวนเต็ม (integer) ใช้เก็บข้อมูลที่เป็นเลขจำนวนเต็มได้ 65536
- float ใช้เก็บข้อมูลที่เป็นเลขทศนิยมแบบ Single Precision
- double ใช้เก็บข้อมูลที่เป็นเลขทศนิยมแบบ Double Precision ซึ่งสามารถเก็บค่าตัวเลขทศนิยมที่มีความละเอียดและถูกต้องของทศนิยมมากกว่าแบบ float ถึง 2 เท่า
- void ใช้เก็บตัวแปรที่ไม่มีค่า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.3 แสดงคุณสมบัติของตัวแปรของภาษาซี

ชนิดตัวแปร	จำนวนบิต	ค่าข้อมูลที่เก็บได้
char	8	-128 ถึง +127
int	16	-32768 ถึง +32767
float	32	3.4E-38 ถึง 3.4E+38
double	64	1.7E-308 ถึง 1.7E+308
void	0	ไม่มีค่า

2.6 สรุป

จากการได้ศึกษาทฤษฎีที่เกี่ยวข้องกับกระบวนการระดับน้ำทั้งสามแบบคือ พีเอตี่ LabVIEW และ Arduino ทำให้เรารู้จักกับหลักการคร่าวๆและความรู้เกี่ยวกับทฤษฎีทั้งสามนั้นจะถูกนำมาใช้เพื่อการออกแบบตัวควบคุมในกระบวนการระดับน้ำต่อไป โดยค่าพารามิเตอร์ต่างๆที่จะนำเอาไปใช้ในการออกแบบตัวควบคุม เพื่อควบคุมระดับน้ำในกระบวนการ โดยจะใช้วิธีการ Trial and Error

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3

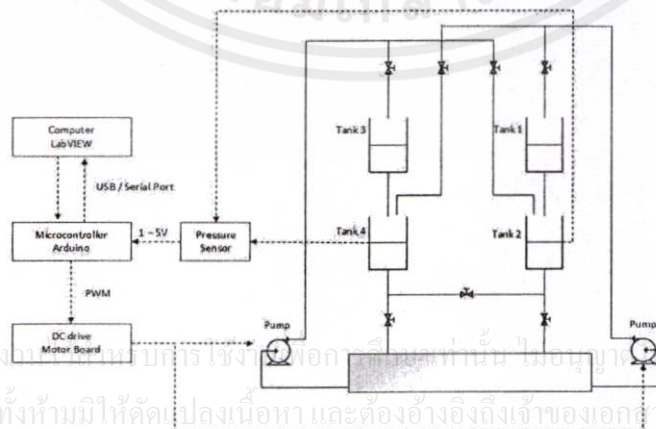
การประยุกต์ใช้งานฮาร์ดแวร์และซอฟต์แวร์

3.1 กล่าวนำ

ในบทที่ 3 เป็นบทที่กล่าวถึงองค์ประกอบของอุปกรณ์ในโครงงานที่ประกอบกันขึ้นเพื่อให้กระบวนการควบคุมสามารถทำงานได้อย่างมีประสิทธิภาพ โดยจะบอกถึงรายละเอียดของทางด้านฮาร์ดแวร์ และซอฟต์แวร์ เพื่อให้ผู้ศึกษาได้รับความรู้และข้อมูลทางด้านฮาร์ดแวร์และซอฟต์แวร์ และนำข้อมูลดังกล่าวไปออกแบบตัวควบคุมตามต้องการ ซึ่งได้แก่ กระบวนการระดับน้ำ, อุปกรณ์วัดระดับน้ำ, มอเตอร์ปั้มน้ำ, ชุดขับมอเตอร์, บอร์ด ADUINO UNO R2 และความรู้ทางด้านภาษาซีที่ใช้กับบอร์ด ADUINO UNO R2 ส่วนองค์ประกอบทางด้านซอฟต์แวร์ได้แก่ โปรแกรมคอมพิวเตอร์ ADUINO และโปรแกรมLabVIEW2011

3.2 การทำงานของฮาร์ดแวร์ในกระบวนการ

การทำงานของกระบวนการจะเริ่มต้นเมื่อกำหนดค่า SV และเลือกฟังก์ชันการทำงานที่บอร์ดไมโครคอนโทรลเลอร์ จากนั้นไมโครคอนโทรลเลอร์จะประมวลผลเพื่อควบคุมกระบวนการและสร้างสัญญาณ PWM ไปที่ชุดขับมอเตอร์เพื่อให้ชุดขับมอเตอร์ทำงาน เมื่อชุดขับมอเตอร์ได้รับสัญญาณ PWM แล้วก็จะขับกระแสไปที่ปั้มน้ำ เพื่อให้ปั้มทำงานและปั้มน้ำเข้าสู่ถังน้ำ เมื่อน้ำเข้าสู่ถังก็จะมีชุดเซนเซอร์วัดระดับคอยวัดระดับน้ำที่ได้ในถังแล้วส่งสัญญาณเป็นแรงดันไฟฟ้า 1-5 V ไปที่ไมโครคอนโทรลเลอร์แล้วไมโครคอนโทรลเลอร์จะทำการประมวลผลสัญญาณ

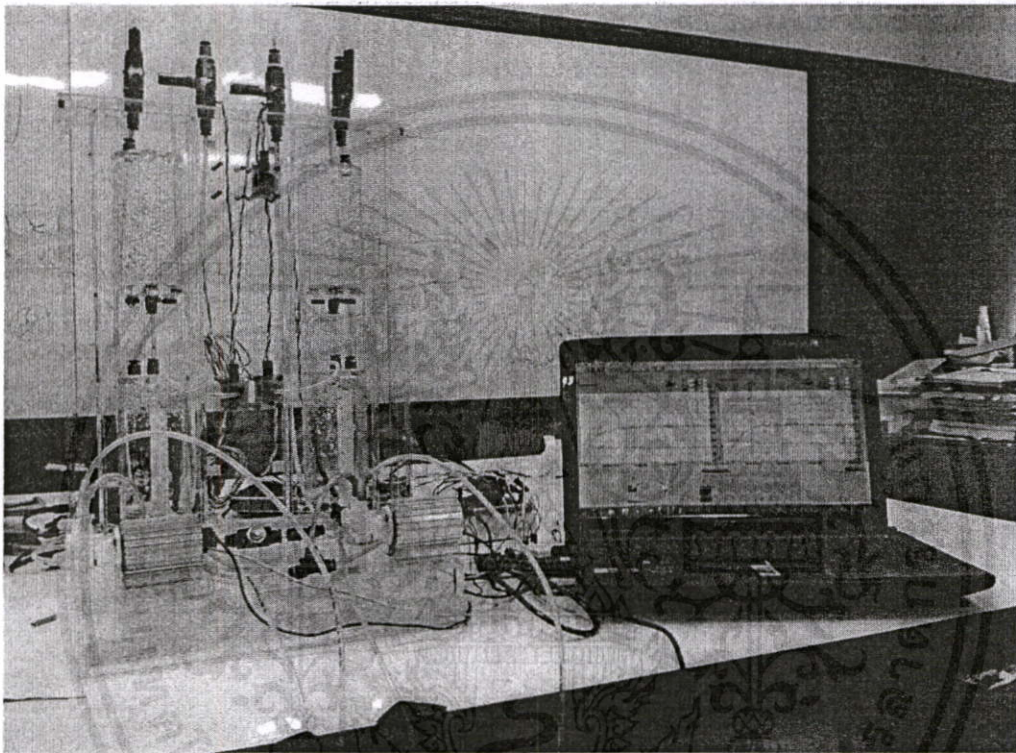


รูปที่ 3.1 ภาพการทำงานของฮาร์ดแวร์ในกระบวนการ

3.3 องค์ประกอบด้านฮาร์ดแวร์

3.3.1 กระบวนการระดับน้ำ

ในหัวข้อนี้จะกล่าวถึงองค์ประกอบในกระบวนการระดับน้ำ ซึ่งเป็นกระบวนการที่ต้องการจะควบคุม โดยในการทดลอง ต้องการที่จะควบคุมกระบวนการระดับน้ำในถังบรรจุน้ำ ถึงที่ 1



รูปที่ 3.2 กระบวนการระดับน้ำประกอบด้วยถังบรรจุน้ำ 2 ใบ วาล์วที่ใช้เปิด-ปิดระดับน้ำ

สายยาง และถังพักน้ำ 1 ใบ โดยมีรายละเอียดของอุปกรณ์ต่างๆดังต่อไปนี้

- ถังบรรจุน้ำ (Tank) เป็นถังทำมาจากอะคริลิกโปร่งใสเพื่อให้สามารถมองเห็นระดับที่เปลี่ยนแปลง ถังบรรจุน้ำนั้นจะมีขนาดของเส้นผ่านศูนย์กลางภายในของถัง 6 ซม. เส้นผ่านศูนย์กลางภายนอกของท่อน้ำทิ้ง 1 ซม. ความสูงของถัง 11 ซม.
- วาล์วควบคุมอัตราการไหล (Flow control Valve) ใช้สำหรับปรับเปลี่ยนค่าสัมประสิทธิ์ของกระบวนการ โดยให้มีมือหมุนเพื่อควบคุมอัตราการไหลเข้าและออกของน้ำที่เข้าถังบรรจุน้ำ ซึ่งภายในกระบวนการมี 1 ตัว คืออยู่บนถังบรรจุน้ำที่ 2
- สายยาง (Tube) ทำมาจากยาง ทำหน้าที่เป็นตัวส่งผ่านน้ำเข้าสู่ถังบรรจุน้ำและถังพักน้ำ
- ข้อต่อ (Fitting) ทำหน้าที่เชื่อมสายยางกับถังบรรจุน้ำ มีขนาดเส้นผ่านศูนย์กลาง 1cm

3.3.2 อุปกรณ์วัดระดับน้ำ

เป็นเซนเซอร์ตรวจวัดความดัน ซึ่งความดันจะแปรผันตรงกับความสูงจากสมการ $P = \rho gh$ ซึ่งในที่นี้เราใช้เซนเซอร์วัดความดันรุ่น MPXV5004G ซึ่งมี range ดังนี้ 0 to 3.92 kPa , 0 to 400 mm H₂O และ 1.0 to 4.9 V Output



รูปที่ 3.3 อุปกรณ์วัดระดับน้ำ (MPXV5004G)

3.3.2.1 Feature

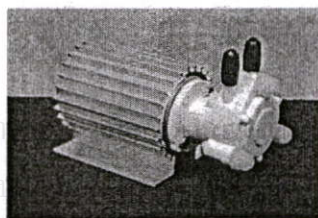
0 to 3.92 kPa
(0 to 400 mm H₂O)
(1.0 to 4.9 V Output)

3.3.2.2 Package Dimension

PIN 1: N/C	PIN 5: N/C
PIN 2: Vs	PIN 6: N/C
PIN 3: GND	PIN 7: N/C
PIN 4: Vout	PIN 8: N/C

3.3.3 มอเตอร์ปั๊มน้ำ

อุปกรณ์ที่ใช้ในการปั๊มน้ำเข้าสู่กระบวนการ ประกอบด้วยสองส่วนใหญ่ๆคือ ส่วนมอเตอร์ไฟฟ้ากระแสตรง และส่วนตัวปั๊มน้ำ โดยมอเตอร์ปั๊มน้ำที่ใช้ในครั้งนี้เป็นมอเตอร์ปั๊มน้ำของบริษัท Grayler รุ่น PQ-24



รูปที่ 3.4 มอเตอร์ปั๊มน้ำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับก
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัด

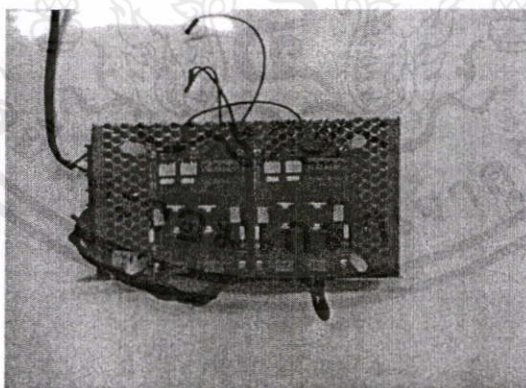
ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
เจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3.1 แสดงคุณสมบัติของมอเตอร์ปั้มน้ำ

รายละเอียด	คุณสมบัติ
Max. Flow rate	2200 ml/min (Vdc)
Max. Suction	10 ft. H ₂ O wet, 4 dry
Max. Pressure	20 psig (46 ft. H ₂ O)
Port Size (OD)	4.8 mm (0.18) hose barb
Power Require (Vdc)	24 Vdc (Up to 28W)
Max. operating Temp	93 องศาเซลเซียส (200 องศาฟาเรนไต์ไฮ)
Ambient Fluid Temp	0 to 93 องศาเซลเซียส (32 to 200 องศาฟาเรนไต์ไฮ)
Max. Viscosity	200 cps
Dimensions	88 × 81 × 92 mm
Weight	1.4 kg (3 lb)

3.3.4 ชุดขับปั้มน้ำ

เป็นชุดอุปกรณ์ที่ใช้รับสัญญาณ PWM จาก Microcontroller แล้วนำไปใช้ในการขับมอเตอร์ปั้มน้ำเข้าสู่กระบวนการ ซึ่งภายในชุดปั้มน้ำจะประกอบไปด้วยส่วนประกอบต่างๆดังนี้

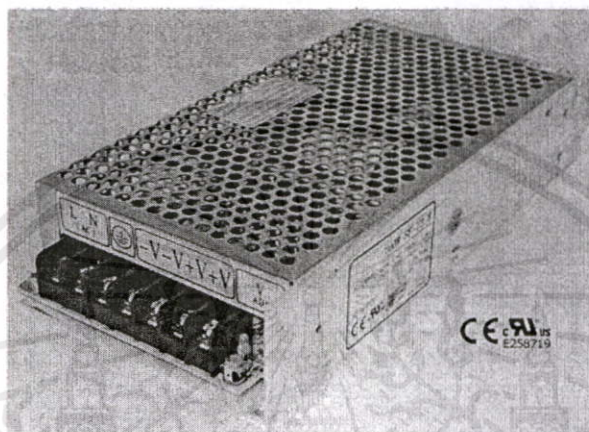


รูปที่ 3.5 ชุดขับมอเตอร์ปั้มน้ำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3.4.1 Switching Power Supply

เป็นแหล่งจ่ายไฟฟ้ากระแสตรงที่มีกำลังนี้สูง มีน้ำหนักเบา และขนาดเล็ก แหล่งจ่ายแบบเชิงเส้น ใช้ในการขับเคลื่อนมอเตอร์ปั้มน้ำ โดย switching power supply โดยรุ่นที่ใช้ในครั้งนี้เป็นของบริษัท Phoka รุ่น LCS 100-24



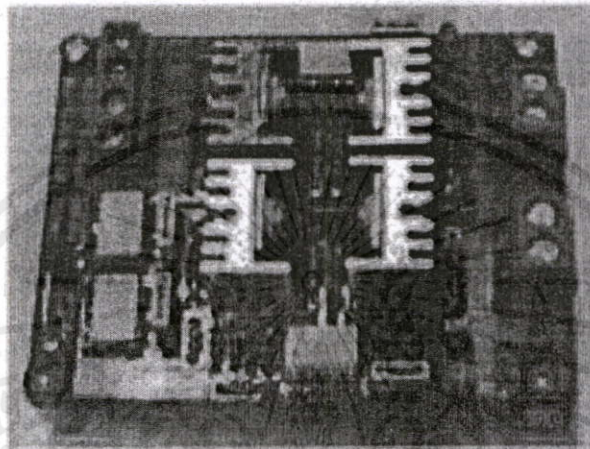
รูปที่ 3.6 Switching Power Supply

ตารางที่ 3.2 แสดงคุณสมบัติของ Switching Power Supply

รายละเอียด	คุณสมบัติ
Output DC Voltage	24 V
Output Rate Current	4.5 A
Output Current Range	0-4.5 A
Output Rate Power	108 W
Output Voltage Adjust Range	21.6-26.4 V
Input Voltage Range	85-132 VAC/170-264 VAC
Input Frequency Range	50-60 Hz
Input AC Current	2.4A/115V, 1.2A/230V
Input Leakage Current	<1 mA/240 VAC
Overload	105-150% Rate Output Power
Overvoltage	13.8-16.2 V
Working Temp.	-10 to +60 (Refer to Output Load derating Curve)
Dimension	199 × 98 × 38 mm.

3.3.4.2 บอร์ดขับมอเตอร์ดีซี

ใช้ในการควบคุมการทำงานของมอเตอร์ปั้มน้ำ โดยรับสัญญาณ PWM (Pulse Width Modulation) จากไมโครคอนโทรลเลอร์เพื่อนำไปควบคุมมอเตอร์ดีซี โดยบอร์ดขับมอเตอร์ดีซีนี้เป็นบอร์ดสำเร็จรูป

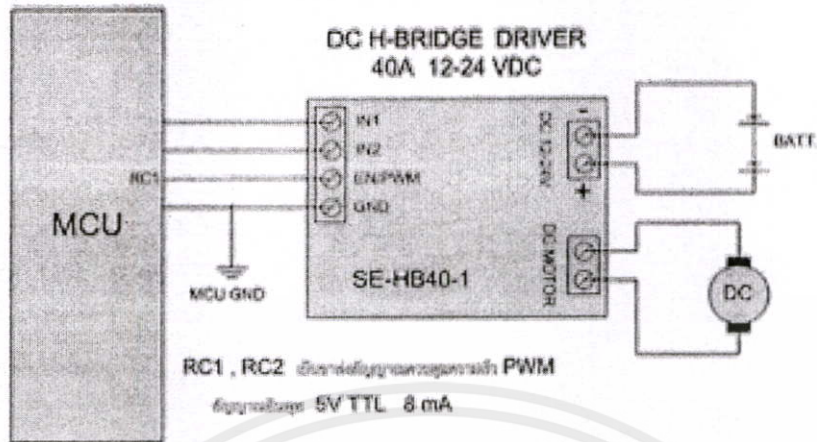


รูปที่ 3.7 บอร์ดขับมอเตอร์ดีซี

รายละเอียดของบอร์ดขับมอเตอร์ดีซี

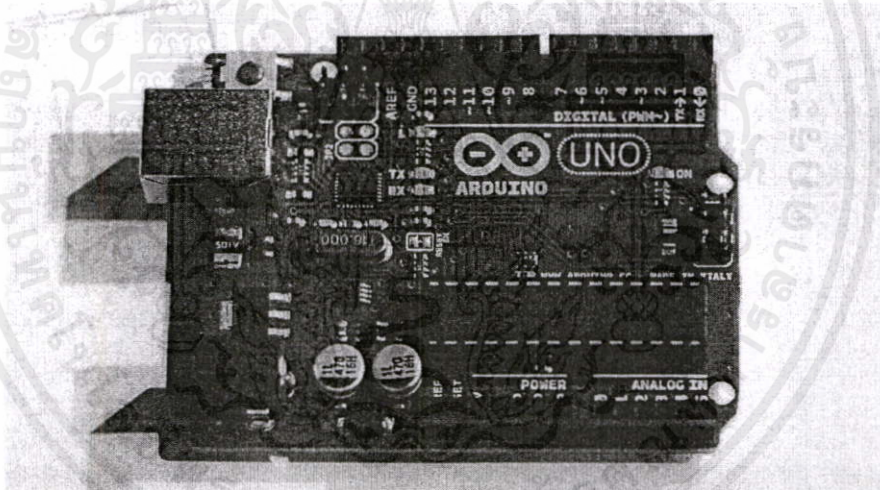
1. Output : Single motor driver
 - Motor DC Supply 12-24 V 40A (Max.)
 - Full-Complementary Power MOSFET Driver With ultrafast reverse recovery protection diodes
2. Input :
 - Full Opto-isolated input interface signals
 - 5V 8 mA TTL – Level
3. Drive Mode : independently with :
 - On – OFF Control
 -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

เอกสารนี้เป็นเอกสาร Speed Control (PWM Drives) การศึกษาเท่านั้น ไม่นอนุญาติให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสี่ 4. PWM Frequency : 400 Hz – 1000 Hz (800 Hz Recommend) ทุกครั้งที่มีการนำไปใช้



รูปที่ 3.8 การเชื่อมต่อในอุปกรณ์เพื่อนำไปใช้งาน

3.3.5 ไมโครคอนโทรลเลอร์ ARDUINO UNO R2



รูปที่ 3.9 บอร์ด Arduino

ARDUINO เป็นระบบที่ใช้ในการพัฒนาอุปกรณ์อิเล็กทรอนิกส์ต้นแบบ ซึ่งออกแบบให้ใช้งานง่ายทั้งฮาร์ดแวร์ และซอฟต์แวร์

บอร์ด ARDUINO ที่มีอยู่หลายแบบนี้จะใช้โปรแกรมตัวเดียวกันในการเขียนชุดคำสั่ง โดยโปรแกรมที่ว่ามีชื่อเป็นทางการว่า Arduino IDE (IDE นั้นย่อมาจาก Integrated Development Environment) ซึ่งใช้งานได้ทั้งบนระบบปฏิบัติการ Window (XP Vista 7 8) ทั้ง 32 และ 64 บิต,

ไม่ว่ากรณีใดก็ตาม อีกทั้งยังมีเว็บไซต์ของ Arduino และเว็บไซต์ของเอกสารทุกครั้งที่มีการนำไปใช้ Mac OS X และ Linux ก็ใช้ได้ ใช้งานได้กับทุกระบบปฏิบัติการเลย

ADUINO มีจุดเด่นในเรื่องของ ความง่ายในการเรียนรู้และใช้งาน เนื่องจากมีการออกแบบคำสั่งต่างๆ ขึ้นมาสนับสนุนการใช้งาน ด้วยรูปแบบที่ง่ายไม่ซับซ้อน ซึ่งถึงแม้ว่า ADUINO เองจะมีรูปแบบการใช้งานคล้ายๆกับไมโครคอนโทรลเลอร์อย่าง Basic Stamp ของ Parallax, BX-24 ของ Netmedias และ Handy Board ของ MIT แต่ก็มีจุดเด่นกว่าของรายอื่นๆหลายอย่าง เป็นต้นว่า

- ราคาไม่แพง เนื่องจากมี Source Code และวงจรแจกให้ฟรี สามารถต่อวงจรขึ้นมาใช้งานเองได้
- โปรแกรมที่ใช้พัฒนาของ ADUINO รองรับการทำงานทั้ง Windows, Linux และ Macintosh OSX
- มีรูปแบบคำสั่งที่ง่ายต่อการใช้งาน แต่สามารถนำไปใช้งานจริงๆที่มีความซับซ้อนมากๆได้ และยังสามารถสร้างคำสั่งและ Library ใหม่ๆขึ้นมาใช้งานเองได้ เมื่อมีความชำนาญมากขึ้น
- มีการเปิดเผยวงจรและ Source Code ทั้งหมดสามารถนำไปพัฒนาต่อยอดเพิ่มเติมได้ตาม ความ ต้องการทั้ง Hardware และ Software

Arduino เป็นบอร์ดไมโครคอนโทรลเลอร์ โดยใช้ AVR ขนาดเล็กเป็นตัวประมวลผลและสั่งงาน เหมาะสำหรับนำไปใช้ในการศึกษาเรียนรู้ระบบไมโครคอนโทรลเลอร์ และนำไปประยุกต์ใช้งานเกี่ยวกับการควบคุมอุปกรณ์ Input / Output ต่างๆได้มากมาย ทั้งในแบบที่เป็นการทำงานตัวเดียวอิสระ หรือเชื่อมต่อสั่งงานร่วมกับอุปกรณ์อื่นๆ เช่น คอมพิวเตอร์ PC ทั้งนี้ก็เนื่องมาจากว่า Arduino สนับสนุนการเชื่อมต่อกับอุปกรณ์ Input / Output ต่างๆได้มากมาย ทั้งแบบ Digital และ Analog เช่น การรับค่าจากสวิตช์ หรืออุปกรณ์ตรวจจับ(Sensor) แบบต่างๆ รวมไปถึง การควบคุมอุปกรณ์ Output ต่างๆตั้งแต่ LED , หลอดไฟ , มอเตอร์ , รีเลย์ และอื่นๆ โดยระบบฮาร์ดแวร์ของ Arduino สามารถสร้างและประกอบขึ้นใช้งานเองได้ ในกรณีที่ผู้ใช้พอมีความรู้ด้านอิเล็กทรอนิกส์อยู่บ้าง หรือสามารถซื้อแผงวงจรสำเร็จรูปที่มีการผลิตออกจำหน่ายกันในราคาที่ไม่แพง สำหรับเรื่องของโปรแกรมที่จะใช้เป็นเครื่องมือในการพัฒนานั้น สามารถ Download มาใช้งานได้ฟรีโดยไม่ต้องเสียค่าใช้จ่ายใดๆ โดน Arduino มีจุดเด่นในเรื่องของความง่ายต่อการพัฒนาโปรแกรม และมีเอกสารข้อมูลรวมทั้งตัวอย่างต่างๆให้ใช้เป็นแนวทางในการศึกษาเรียนรู้เป็นจำนวนมาก เนื่องจาก Arduino เป็นระบบการพัฒนาไมโครคอนโทรลเลอร์แบบ Open Source ซึ่งมีการตีพิมพ์เอกสารต่างๆที่เกี่ยวข้องออกมาเผยแพร่ รวมทั้งการเปิดเผย Source Code และตัวอย่างต่างๆให้ผู้ใช้นำไปใช้งานไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งยังมีเทคนิคแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้หรือพัฒนาต่อยอดได้โดยไม่ต้องเสียค่าใช้จ่าย

ความสามารถของไมโครคอนโทรลเลอร์ Arduino UNO R2 พอสรุปได้ดังนี้

Microcontroller	ATmega328
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limits)	6-20V
Digital I/O Pins	14 (of which 6 provide PWM output)
Analog Input Pins	6
DC Current per I/O Pin	40 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	32 KB (ATmega328) of which 0.5 KB used by bootloader
SRAM	2 Kb (ATmega328)
EEPROM	1 Kb (ATmega328)
Clock Speed	16 MHz

(PCINT14/RESET) PC6	1	28	PC5 (ADC5/SCL/PCINT13)
(PCINT16/RXD) PD0	2	27	PC4 (ADC4/SDA/PCINT12)
(PCINT17/TXD) PD1	3	26	PC3 (ADC3/PCINT11)
(PCINT18/INT0) PD2	4	25	PC2 (ADC2/PCINT10)
(PCINT19/OC2B/INT1) PD3	5	24	PC1 (ADC1/PCINT9)
(PCINT20/XCK/T0) PD4	6	23	PC0 (ADC0/PCINT8)
VCC	7	22	GND
GND	8	21	AREF
(PCINT6/XTAL1/TOSC1) PB6	9	20	AVCC
(PCINT7/XTAL2/TOSC2) PB7	10	19	PB5 (SCK/PCINT5)
(PCINT21/OC0B/T1) PD5	11	18	PB4 (MISO/PCINT4)
(PCINT22/OC0A/AIN0) PD6	12	17	PB3 (MOSI/OC2A/PCINT3)
(PCINT23/AIN1) PD7	13	16	PB2 (SS/OC1B/PCINT2)
(PCINT0/CLKO/ICP1) PB0	14	15	PB1 (OC1A/PCINT1)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 3.10 รายละเอียดแสดงขาของ Atmega 328

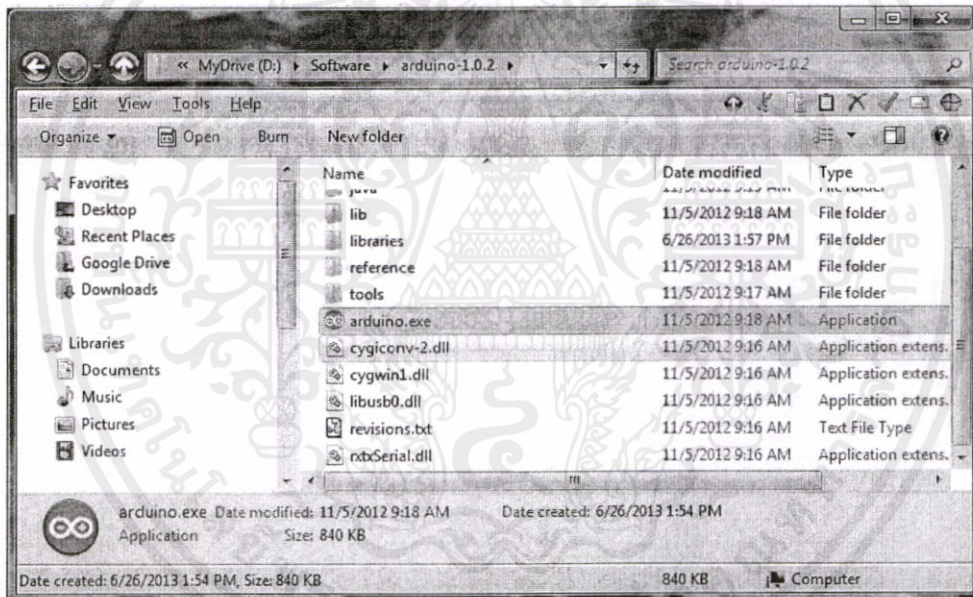
3.4 องค์ประกอบด้านซอฟต์แวร์

3.4.1 โปรแกรมคอมพิวเตอร์ Arduino IDE 1.0.5

การเขียนโปรแกรมให้ไมโครคอนโทรลเลอร์ (Arduino UNO R2) สามารถดาวน์โหลดโปรแกรม Arduino IDE จาก <http://arduino.cc/en/Main/Software> โดยไม่เสียค่าใช้จ่าย และสามารถ Upload โปรแกรมที่เขียนเข้าไปโดยง่าย และภายในตัวโปรแกรมายังมี Library เพื่อนำเอาไปเป็นตัวอย่างศึกษาและพัฒนาโปรแกรมต่อไป

3.4.1.1 ขั้นตอนการติดตั้งโปรแกรม Arduino IDE 1.0.5

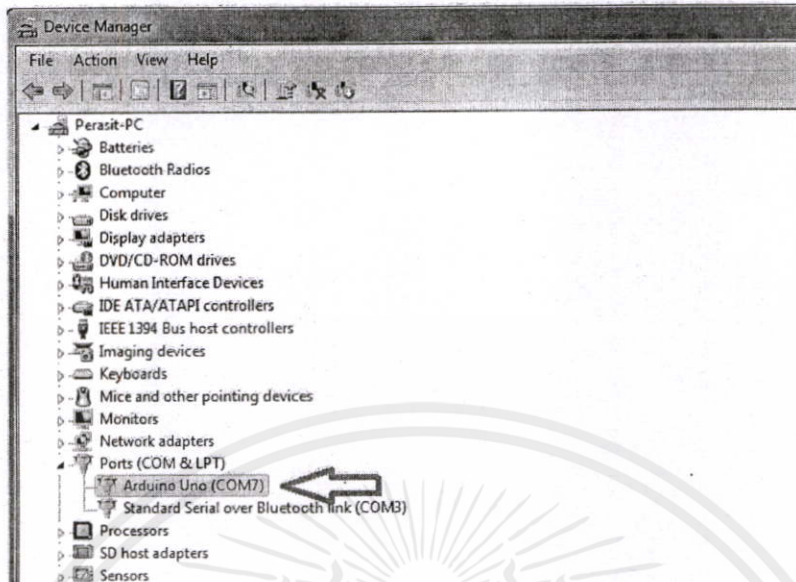
1. ดาวน์โหลดโปรแกรมจาก <http://arduino.cc/en/Main/Software> จะได้ไฟล์ดังรูป



รูปที่ 3.11 แสดงไฟล์ที่ดาวน์โหลดเพื่อนำมาติดตั้ง

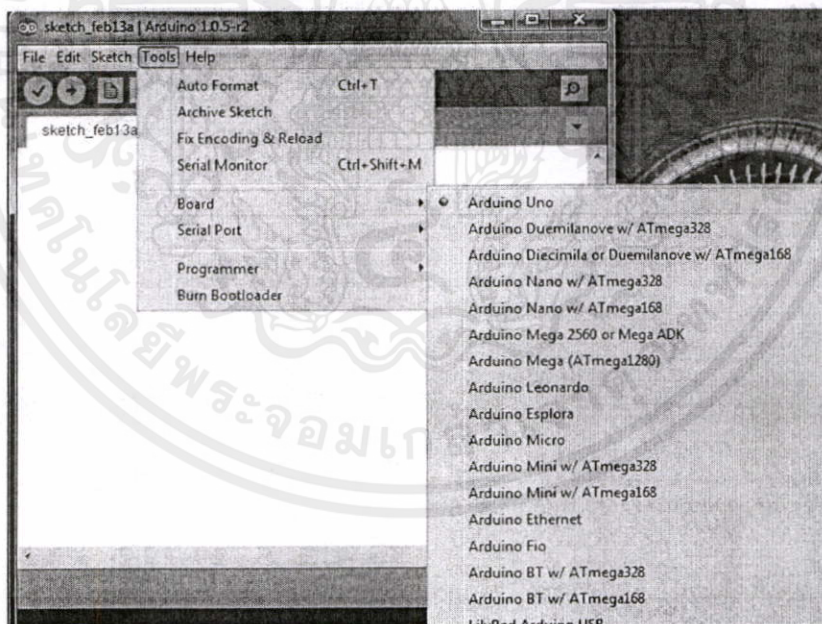
2. เมื่อติดตั้งเสร็จก็ทำการ เชื่อมต่อพอร์ตของ Arduino เข้ากับคอมพิวเตอร์ แล้วเข้าไปดูหมายเลขพอร์ตเพื่อให้คอมพิวเตอร์กับบอร์ด Arduino ติดต่อกันได้ โดยเข้าไปที่ control panel>device manager>PORT

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.12 แสดงการติดต่อกันระหว่างคอมพิวเตอร์กับไมโครคอนโทรลเลอร์

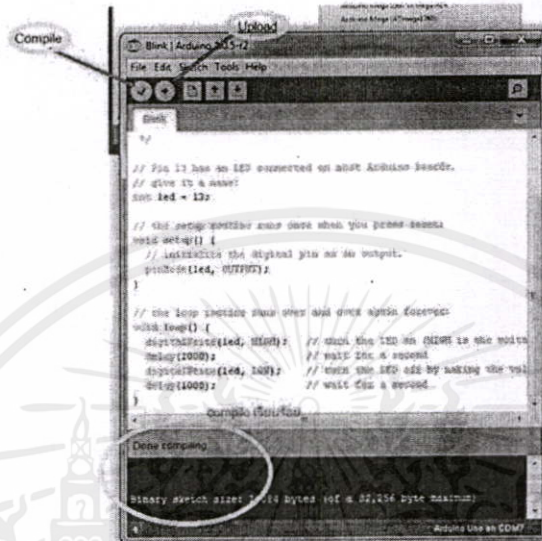
3. การเลือกรุ่นของ Arduino Board ที่เรานำมาใช้



รูปที่ 3.13 แสดงการเลือกรุ่นของ Arduino Board

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

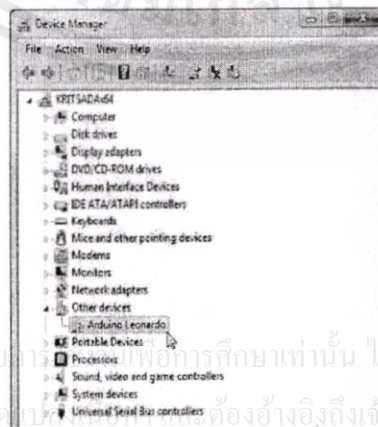
4. ก่อน Upload โปรแกรมเข้าไปในตัวไมโครคอนโทรลเลอร์ควรจะ compile โปรแกรมนั้นก่อน ดังรูป



รูปที่ 3.14 แสดงหน้าต่างโปรแกรมเมื่อ Compile ผ่านแล้ว

3.4.1.2 การแก้ปัญหาในกรณีที่ระบบแจ้งว่า ไม่สามารถค้นพบไดรเวอร์เพื่อติดต่อ กับฮาร์ดแวร์ได้ ให้ทำตามขั้นตอนนี้

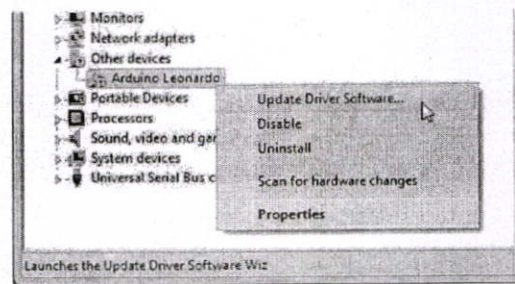
1. เมื่อเปิดหน้าต่าง Device Manager โดยเข้าไปที่ start > control panel > device manager ที่หน้าต่าง Device manager จะพบอุปกรณ์ที่มี เครื่องหมาย ! ปราบกฏอยู่



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้ในการเรียนการสอนเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกหรือทำซ้ำโดยไม่ได้รับอนุญาตของเอกสารทุกครั้งที่มีการนำไปใช้

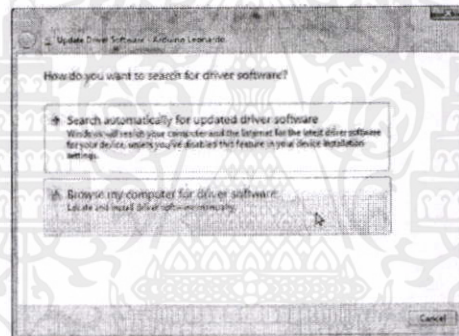
รูปที่ 3.15 หน้าต่าง Device Manager

2. คลิกเมาส์ปุ่มขวาที่รายการนั้น แล้วเลือก Update Driver



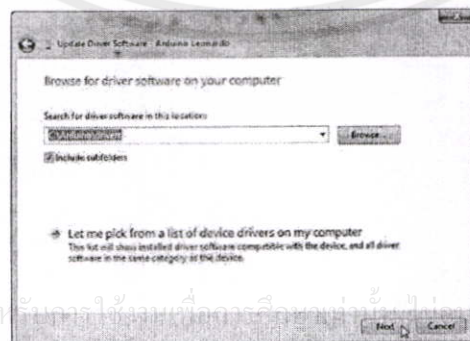
รูปที่ 3.16 แสดงการเข้าไปที่หน้าต่าง Update Driver Software

3. เลือก Browse my computer for driver software



รูปที่ 3.17 หน้าต่าง Update Driver Software

4. เลือกโฟลเดอร์ของไดรเวอร์ไปที่ C:\Program Files (x86)\Arduino\drivers และกด Next เพื่ออัปเดตไดรเวอร์

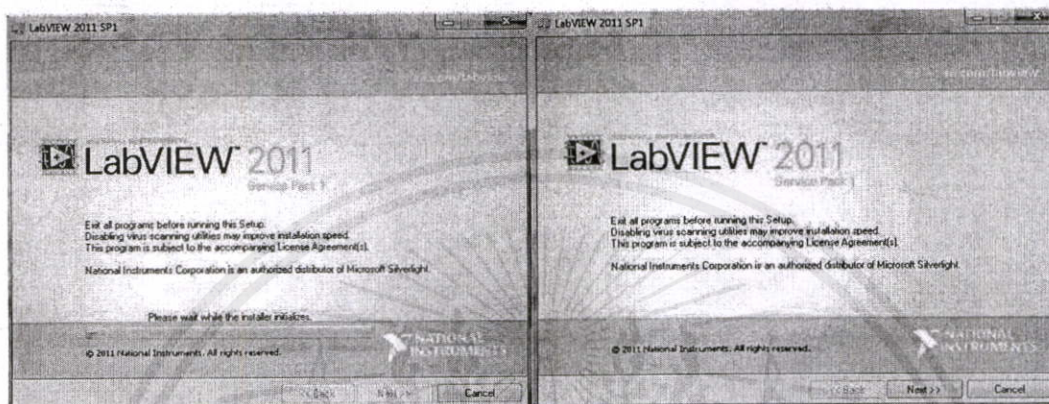


รูปที่ 3.18 แสดงการเลือกโฟลด์ที่จะนำมาอัปเดต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับงานวิจัยที่ดำเนินการโดยศูนย์วิจัยและพัฒนาให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้นำไปเผยแพร่หรือใช้งานนอกเหนือจากเอกสารทุกครั้งที่มีการนำไปใช้

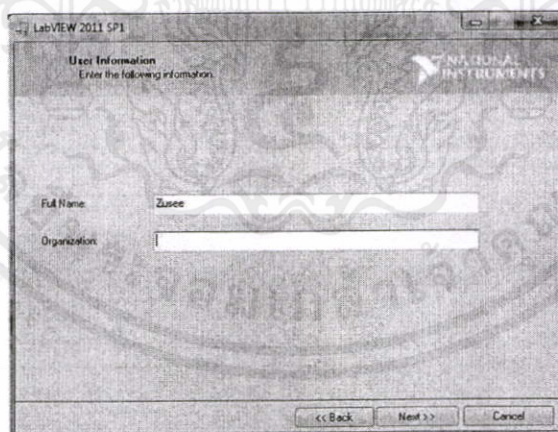
3.5 การติดตั้งซอฟต์แวร์ LabVIEW 2011

1. เมื่อใส่แผ่นซีดีเข้าไปในเครื่องคอมพิวเตอร์แล้วจะปรากฏหน้าต่างดังรูปที่ 3.19 และทำการกดปุ่ม Next



รูปที่ 3.19 เริ่มต้นการติดตั้ง

2. หลังจากที่ถูกปุ่ม Next แล้วจะปรากฏหน้าต่าง ดังรูป 3.20 ซึ่งให้กรอกข้อมูลของผู้ใช้ โดยจะให้กรอก ชื่อผู้ใช้งานและสถานที่ทำงาน

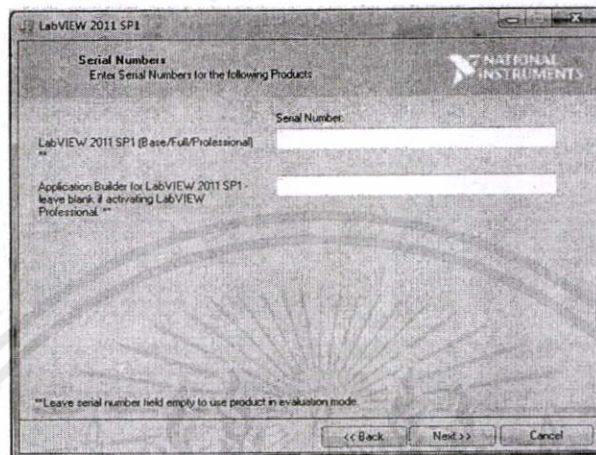


รูปที่ 3.20 กรอกชื่อผู้ใช้งานและสถานที่ทำงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

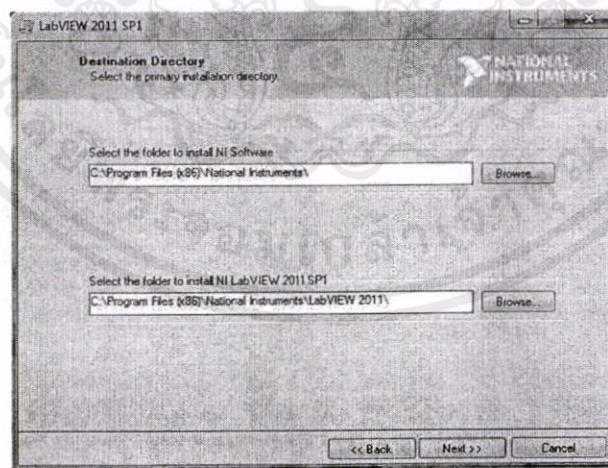
3. ขั้นตอนในรูปที่ 3.21 ให้กรอก Serial Number กรณีที่ติดตั้งแบบ Evaluation แล้วกดปุ่ม

Next



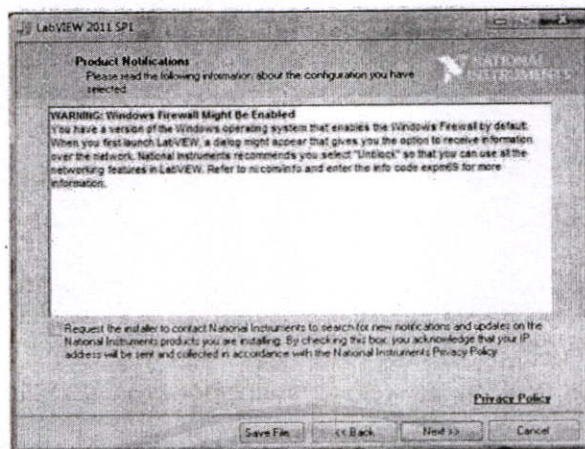
รูปที่ 3.21 กรอก Serial Number

4. หลังจากกด Next จะปรากฏหน้าต่างดังรูป 3.22 ซึ่งจะให้ทำการเลือก Directory ในการติดตั้งโปรแกรม โดยเราสามารถเปลี่ยนตำแหน่ง Directory โดยทำการกดปุ่ม Browse และเลือกตำแหน่ง Directory และกดปุ่ม Next



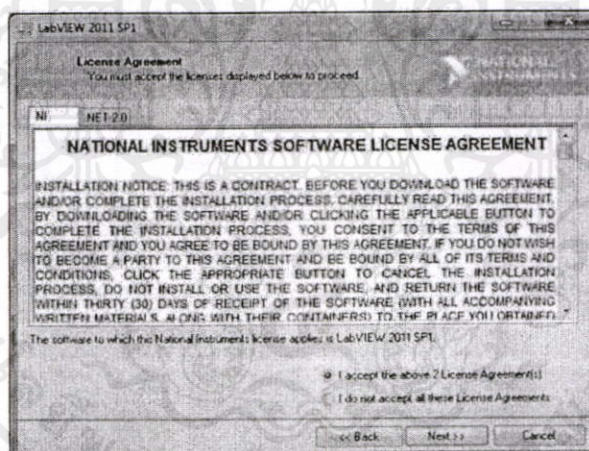
รูปที่ 3.22 เลือกตำแหน่งที่จะทำการติดตั้ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 5. หลังจากกดปุ่ม Next จากรูป 3.22 จะมีหน้าต่าง Feature ปรากฏขึ้นมาเพื่อให้เลือก
 ว่าจะกรณีใดบ้างที่สนใจติดตั้งและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
 ติดตั้ง แสดงในรูปที่ 3.23



รูปที่ 3.23 เช็คเวอร์ชันออนไลน์

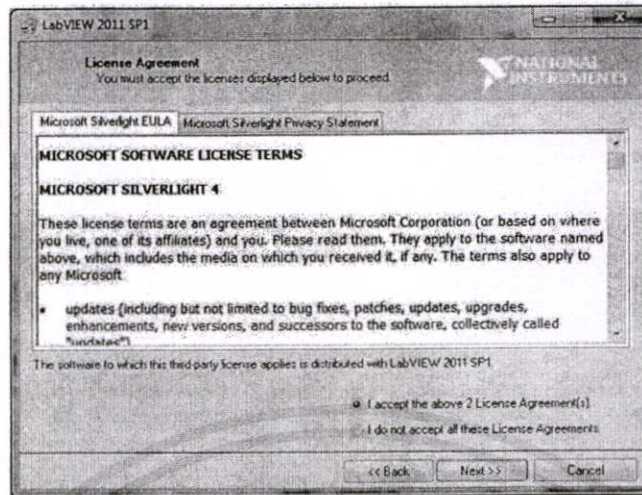
6. หลังจากผ่านขั้นตอนที่ 5 มาแล้วขั้นตอนต่อไปจะเป็นการยอมรับในเงื่อนไขของลิขสิทธิ์ ให้เลือก I accept License Agreement แล้วกดปุ่ม Next แสดงในรูป 3.24



รูปที่ 3.24 แสดงลิขสิทธิ์และสิทธิ์ในการใช้งาน

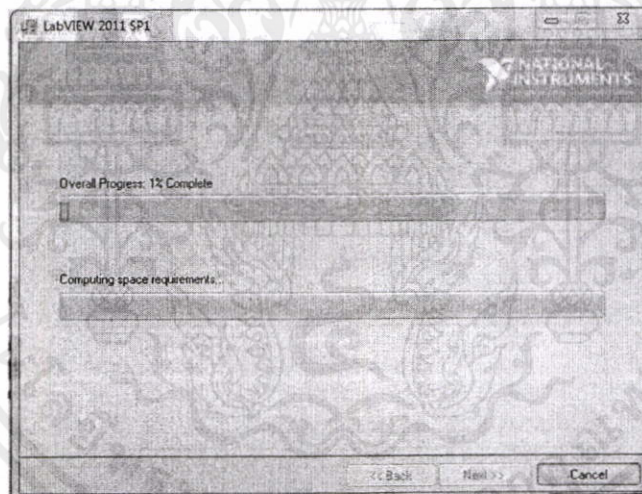
7. หลังจากนั้นจะปรากฏหน้าต่างขึ้นมาเพื่อแสดงรายละเอียดในการติดตั้งโปรแกรม ให้เลือก I accept แล้วทำการกด Next

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.25 แสดงลิขสิทธิ์และสิทธิ์ในการใช้งาน

8. หลังจากกด Next ในรูปที่ 3.25 โปรแกรมจะทำการติดตั้งในรูป 3.26



รูปที่ 3.26 ขั้นตอนการดำเนินการติดตั้งโปรแกรม

9. รอจนโปรแกรมติดตั้งครบ 100% หลังจากนั้นกดปุ่ม Finish

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.6 สรุป

ในบทนี้เราได้เรียนรู้เกี่ยวกับคุณสมบัติขั้นพื้นฐานขององค์ประกอบทั้งด้าน Hardware และ Software โดยผู้ศึกษาได้ทำการเสนอวิธีการติดตั้ง Software ทั้ง Arduino, LabVIEW และยังสามารถเสนอวิธีการแก้ปัญหาที่พบบ่อยหลังจากการติดตั้งโปรแกรม ในส่วนของ Hardware ได้มีการเลือกอุปกรณ์ในการใช้งานที่เหมาะสมกับระบบ



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

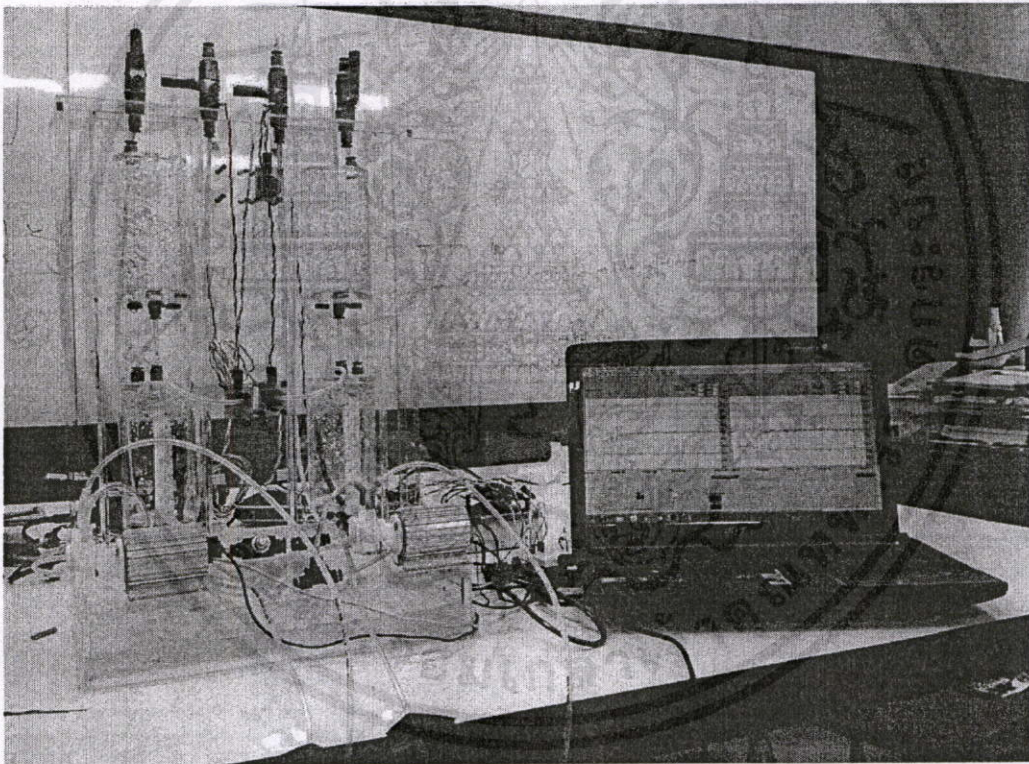
บทที่ 4

การทดลองและผลการทดลอง

4.1 กล่าวนำ

ในบทนี้จะกล่าวถึงการจำลองกระบวนการควบคุมระดับน้ำ โดยจัดทำโมเดลทดลองแบบ 4 ถัง และจัดทำกรทดลองในแบบต่างๆ คือ การควบคุมระดับในถังปิดแบบ 1 ถัง และ 2 ถัง โดยการหาค่าพารามิเตอร์จะใช้วิธี Trial and Error และนำผลการทดลองที่ได้มา พล็อตกราฟ

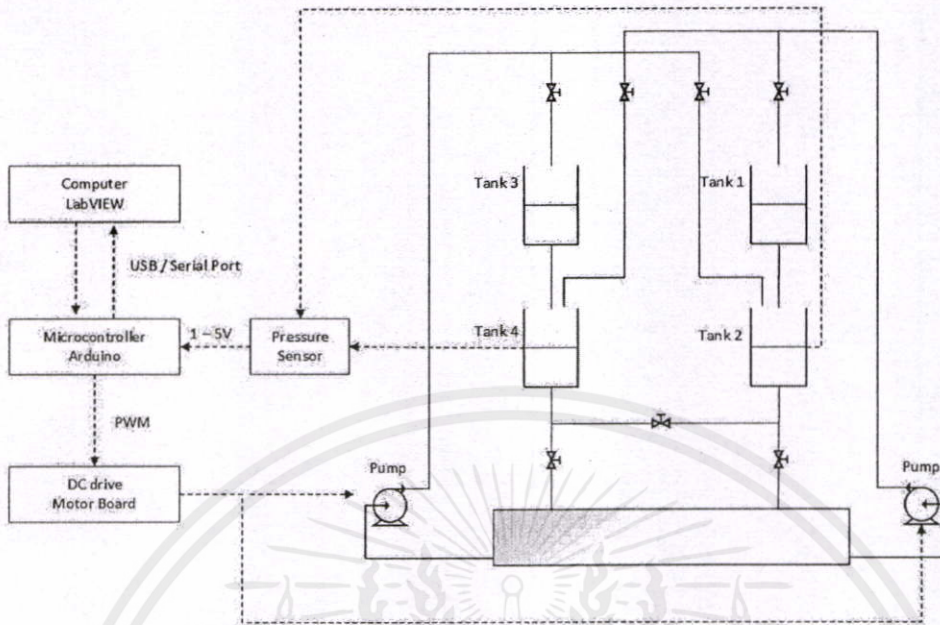
4.2 การบวนการที่ใช้ในการทดลอง



รูปที่ 4.1 กระบวนการระดับน้ำ

จากรูปที่ 4.1 เป็นกระบวนการวัดระดับน้ำที่ใช้ในการทดลอง และอุปกรณ์ที่กล่าวไว้ในบทที่ 3 เราสามารถออกแบบชุดทดลองกระบวนการวัดระดับน้ำเพื่อใช้ในการทดลอง ได้ตามไดอะแกรมของกระบวนการในรูปที่ 4.2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่จำกัดใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.2 การออกแบบระบบการควบคุมน้ำ

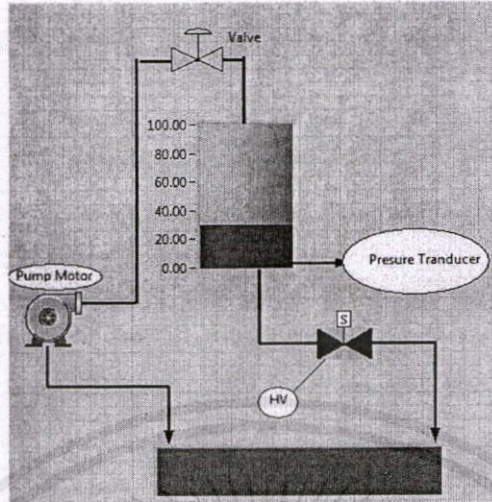
จากกระบวนการในรูปที่ 4.2 การทำงานของระบบในกระบวนการจะเริ่มต้นจากชุดควบคุม ซึ่งก็คือตัวไมโครคอนโทรลเลอร์ Arduino โดยจะทำการกำหนดค่าเป้าหมาย(Set Point) จากนั้นไมโครคอนโทรลเลอร์จะประมวลผลเพื่อสร้างสัญญาณควบคุมตามอัลกอริทึมที่ได้ออกแบบไว้ สัญญาณที่ออกไปจะออกเป็นสัญญาณPWM(Pulse Width Modulation) ไปยังชุดขับเคลื่อนมอเตอร์ปั๊มน้ำ เพื่อนำน้ำเข้าสู่กระบวนการ เมื่อมีการเปลี่ยนแปลงของระดับน้ำเกิดขึ้น อุปกรณ์ที่วัดก็จะตรวจจกระดับของน้ำและส่งสัญญาณแรงดันออกมาเป็น 1-5Vdc จากนั้นจึงนำสัญญาณที่ได้ส่งเข้าสู่ตัวควบคุม เพื่อนำไปประมวลผลต่อไป

ในส่วนของการทดลองนั้น จะทำการควบคุม SV(Set Point) เท่ากับ 40% เพื่อประสิทธิภาพของการทดลองในแบบต่างๆ

4.3 การจำลองกระบวนการ

4.3.1 การทดลองควบคุมกระบวนการระดับน้ำแบบ 1 ถึงด้วยวิธีวงรอบเปิด

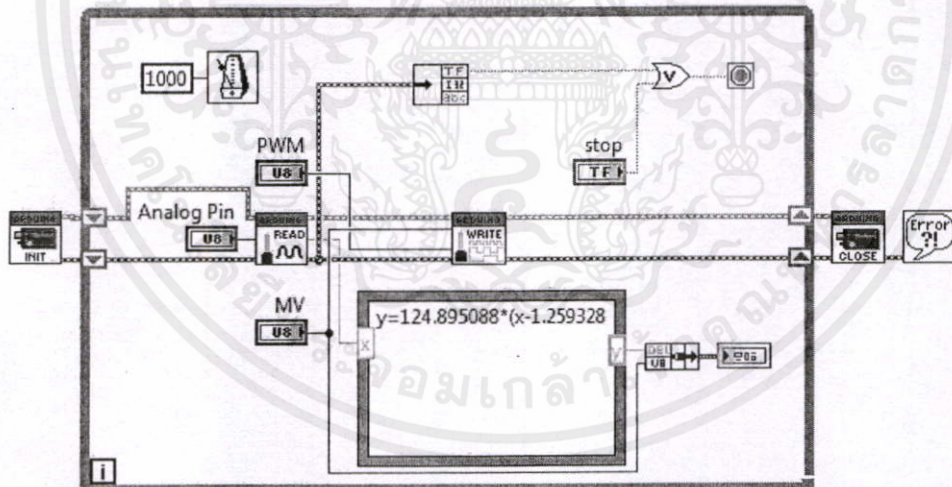
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.3 แสดงกระบวนการควบคุมระดับน้ำแบบ 1 ถึง

คือจะทำการใส่ค่า MV ในไมโครคอนโทรลเลอร์โดยตรงให้กับบอร์ดขับเคลื่อนมอเตอร์ปั้มน้ำเพื่อไปสั่งให้มอเตอร์ปั้มน้ำทำงาน มีขั้นตอนการทำงานดังนี้

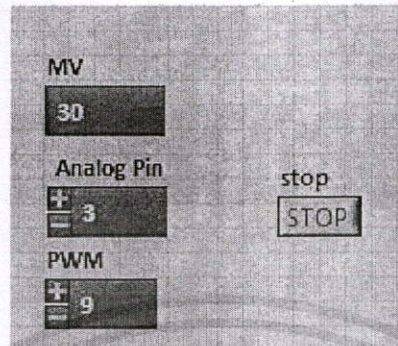
- 1) เขียนโปรแกรม LabVIEW ในส่วนของ Block Diagram เพื่อใช้ในการควบคุมกระบวนการดังรูป



รูปที่ 4.4 แสดงโปรแกรมที่ใช้ควบคุมกระบวนการ

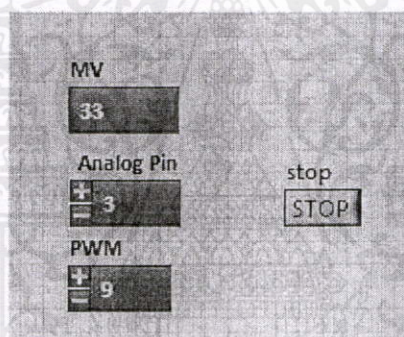
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2) ใส่ค่า MV = 30 ที่ต้องการในโปรแกรม LabVIEW ดังรูป



รูปที่ 4.5 ป้อนค่า MV = 30

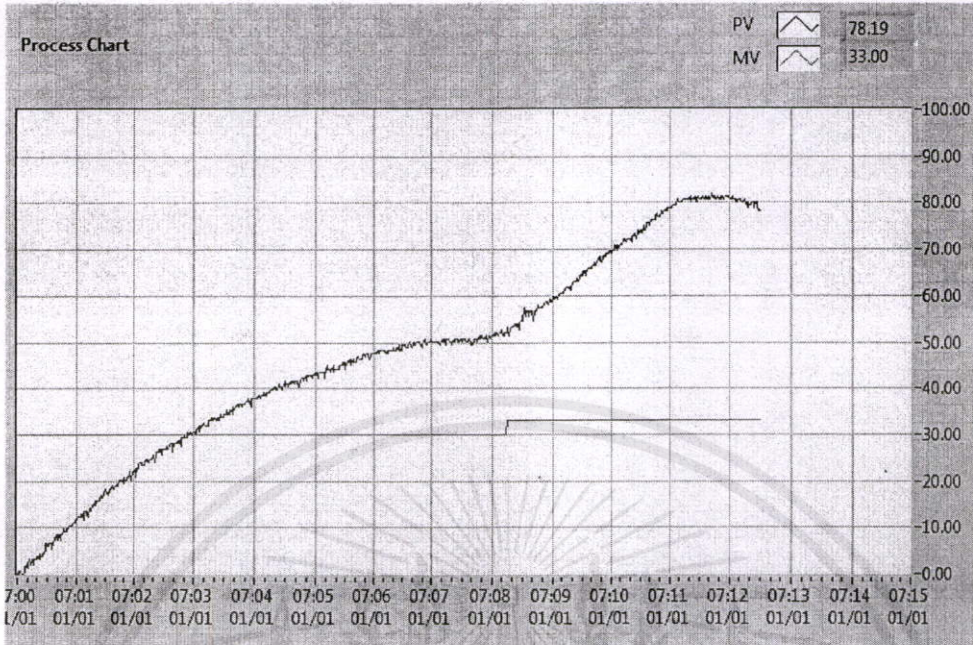
3) เมื่อระดับเริ่มนิ่งจากนั้น step สัญญาณ จาก MV = 30 ไปที่ MV = 33



รูปที่ 4.6 สัญญาณ Step จาก MV = 30 ไปที่ MV = 33

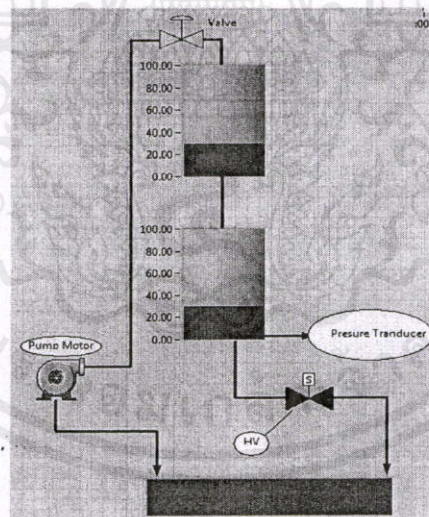
4) จะได้กราฟดังรูปที่ 4.7

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.7 ผลตอบสนองรูปตัว S เมื่อใช้วิธีแบบวงเปิด

4.3.2 การทดลองควบคุมกระบวนการระดับน้ำแบบ 2 ถัง



รูปที่ 4.8 แสดงกระบวนการควบคุมระดับน้ำแบบ 2 ถัง

4.3.2.1 การควบคุมกระบวนการระดับน้ำโดยใช้วิธีวงรอบเปิด

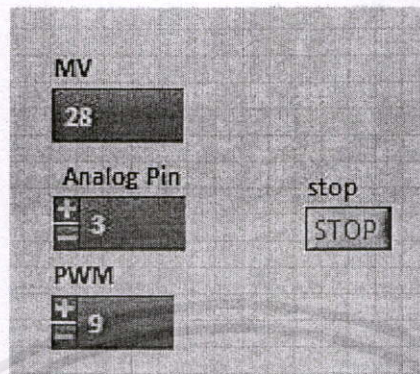
คือ การป้อนค่า MV ให้กับชุดขับเคลื่อนมอเตอร์ซีซี เพื่อนำน้ำเข้าไปในแทงค์ทั้งสอง มี

เอกสารนี้เป็นเอกสารที่ดัดแปลงจากเอกสารต้นฉบับเพื่อการศึกษานี้เท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีก 1.1 เขียนโปรแกรมใน LabVIEW เพื่อควบคุมเหมือนกับกระบวนการควบคุมระดับน้ำ

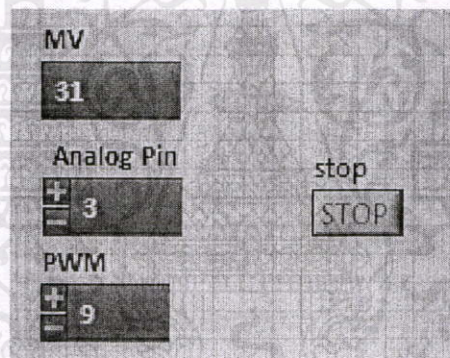
น้ำแบบ 1 ถัง

1.2 เข้ากำหนดค่า MV ในโปรแกรม LabVIEW ให้ MV = 28 ดังรูป



รูปที่ 4.9 ป้อนค่า MV = 28

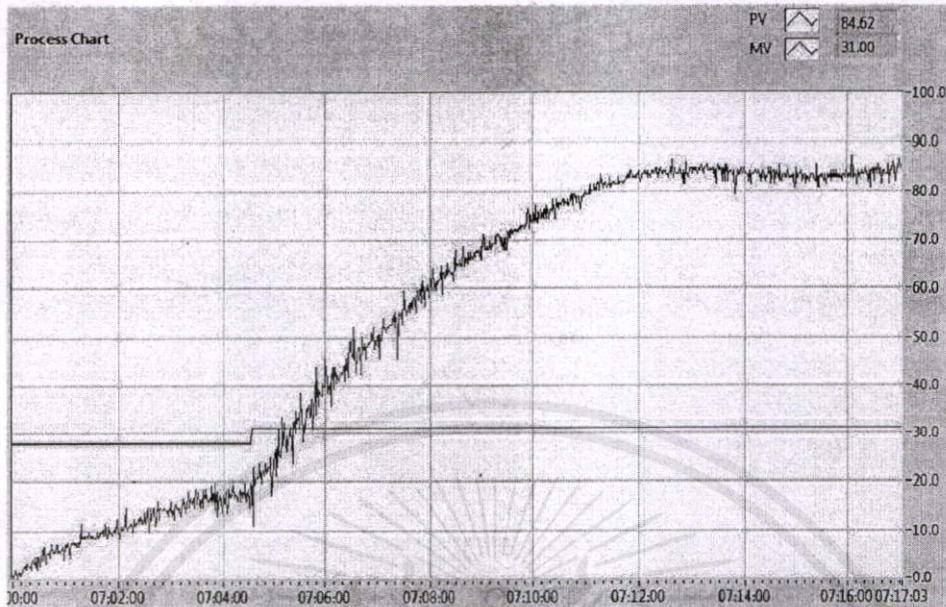
1.3 เมื่อระดับน้ำนิ่งจากนั้น Step สัญญาณจาก MV = 28 ไปที่ MV = 31



รูปที่ 4.10 สัญญาณ Step จาก MV = 28 ไปที่ MV = 31

1.4 จะได้กราฟผลตอบสนองดังรูปที่ 4.11

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

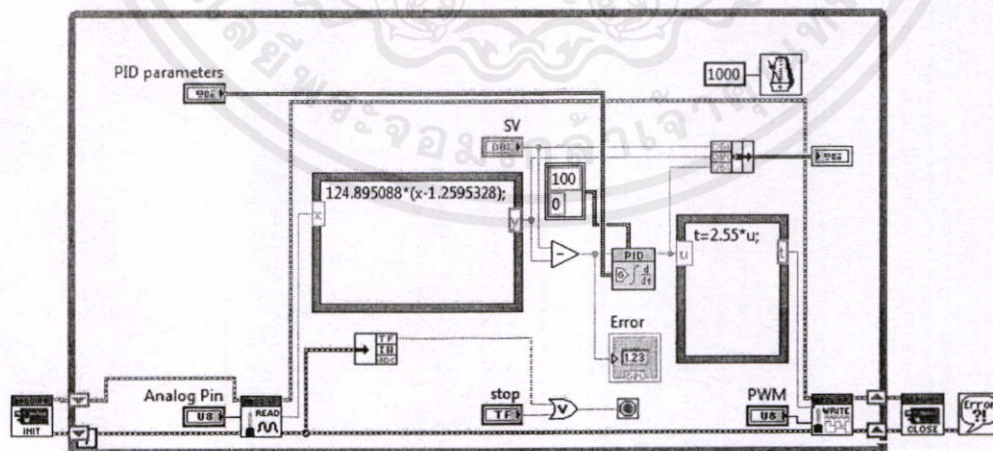


รูปที่ 4.11 ผลตอบสนองรูปตัว S เมื่อใช้วิธีวงรอบเปิด

4.3.2.2 การควบคุมกระบวนการระดับน้ำโดยใช้วิธีวงรอบเปิด

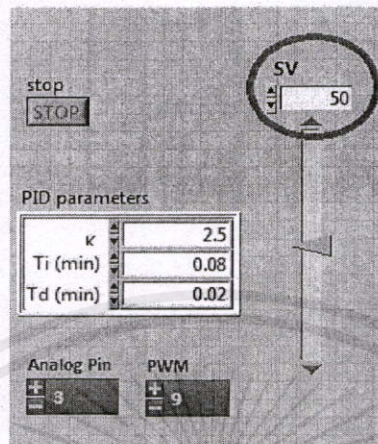
คือ การที่เรากำหนดค่าเป้าหมาย (SV) จากนั้นไมโครคอนโทรลเลอร์แล้วก็ส่งสัญญาณ(MV) ไปขับบอร์ดมอเตอร์ดีซี เพื่อนำน้ำเข้าแท่งจากนั้นอุปกรณ์วัดจะส่งสัญญาณแรงดันออกมา 1-5Vdc (PV) เข้าที่ไมโครคอนโทรลเลอร์เพื่อประมวลผลต่อไป มีขั้นตอนกระบวนการดังนี้

2.1 เขียนโปรแกรมใน LabVIEW เพื่อควบคุมกระบวนการระดับน้ำแบบ 2 ดังรูปด้านล่าง



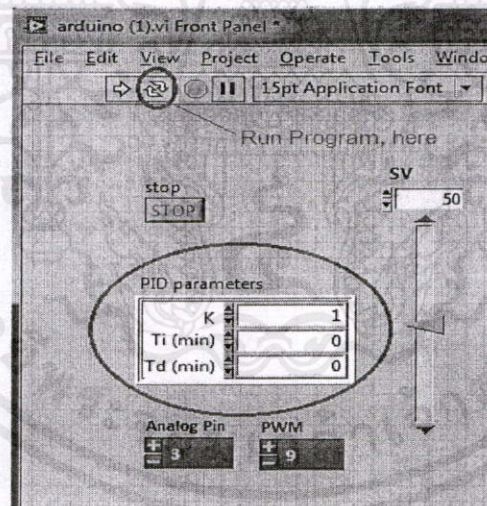
เอกสารนี้เป็นเอกสารที่รูปที่ 4.12 แสดงส่วนของ Block Diagram ที่ใช้ควบคุมกระบวนการ ไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2 กำหนดค่า Set Point(SV) = 50 ให้กับโปรแกรม LabVIEW ดังรูป



รูปที่ 4.13 กำหนดค่า SV = 50

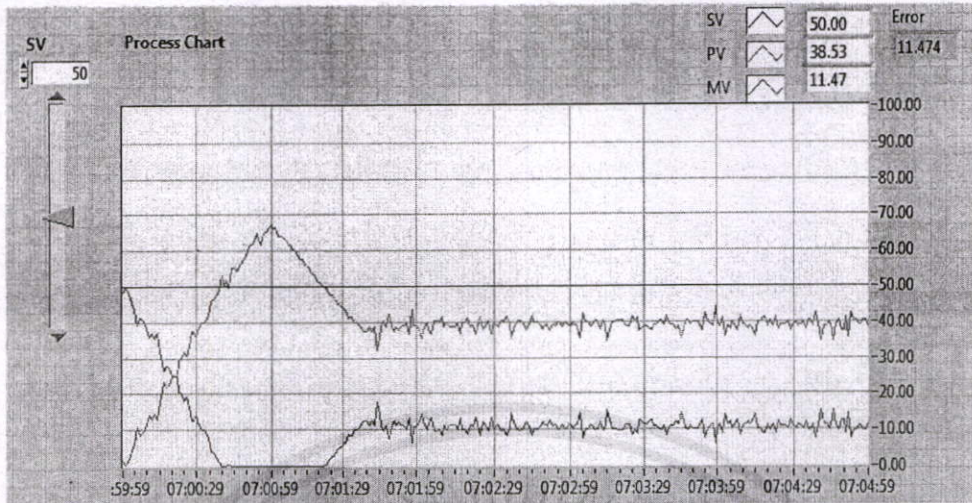
2.3 กำหนดค่าพารามิเตอร์ PID $K_c = 1, T_i = 0, T_d = 0$ แล้วกดรันโปรแกรม



รูปที่ 4.14 การกำหนดค่าพารามิเตอร์

2.4 ผลตอบสนองที่ได้เป็นดังรูปที่ 4.15

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.15 ผลตอบสนองของกระบวนการเมื่อใช้วิธีวงรอบปิด

ตารางที่ 4.1 ผลตอบสนองของกระบวนการดังรูปที่ 4.15

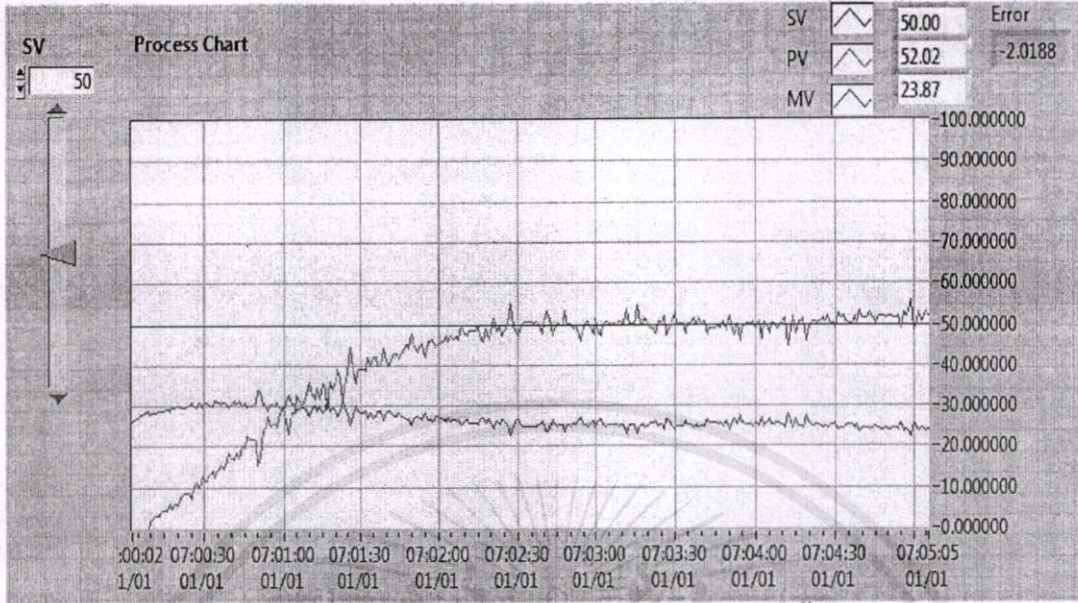
Delay time (t_d)	Rise time (t_r)	Peak time (t_p)	Maximum Overshoot (M_p)	Setting time (t_s)
20 sec	85 sec	60 sec	15 %	100 sec

จากการใช้วิธีวงรอบปิด จะต้องกำหนดค่าพารามิเตอร์ PID ให้กับไมโครคอนโทรลเลอร์เพื่อประมวลและควบคุมกระบวนการ ดังนั้นเราจะใช้วิธีการลองผิดลองถูกเพื่อหาค่าพารามิเตอร์ PID เพื่อใช้ในการทดลองกระบวนการ

ส่วนต่อไปจะเป็นค่าพารามิเตอร์ PID ที่ปรับจนเหมาะสมกับกระบวนการควบคุมระดับน้ำแบบ 2 แทงค์ และผลตอบสนองที่ได้จะแสดงในรูปที่ 4.12

$$K_c = 0.5, T_i = 1, T_d = 1.6$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

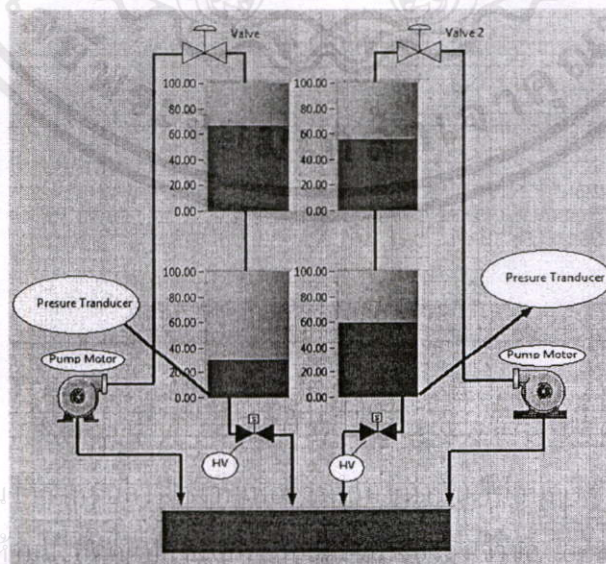


รูปที่ 4.16 แสดงผลตอบสนองของกระบวนการควบคุมระดับน้ำแบบ 2 ถึง

ตารางที่ 4.2 ผลตอบสนองของกระบวนการดังรูปที่ 4.16

Delay time (t_d)	Rise time (t_r)	Peak time (t_p)	Maximum Overshoot (M_p)	Setting time (t_s)
65 sec	97.5 sec	145 sec	0 %	150 sec

4.3.3 การทดลองควบคุมกระบวนการระดับน้ำแบบ 4 ถึง โดยวิธีวงรอบปิด

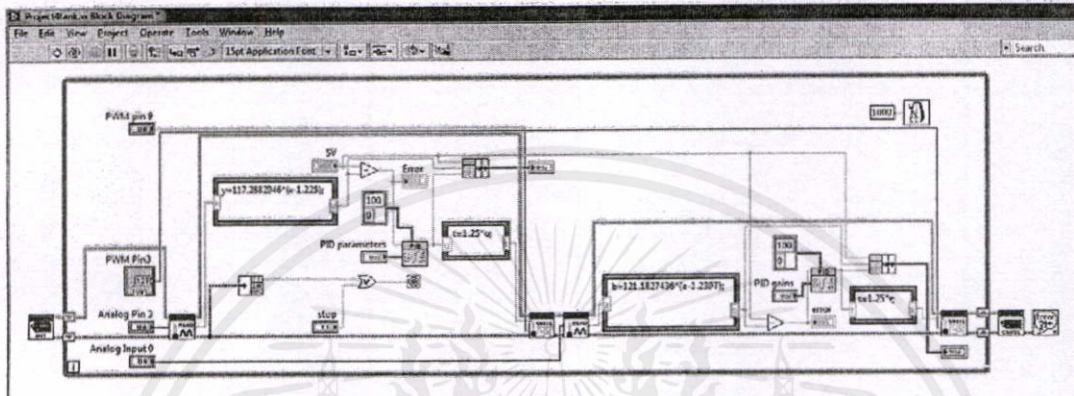


รูปที่ 4.17 แสดงกระบวนการควบคุมระดับน้ำแบบ 4 ถึง

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้เพื่อใช้ในการศึกษาเท่านั้น ไม่ควรนำไปใช้ประโยชน์ด้านการค้า โดยไม่ขออนุญาตจากเจ้าของลิขสิทธิ์ ทุกครั้งที่มีการนำไปใช้

คือ ทำการออกแบบโปรแกรมควบคุมและออกแบบตัวควบคุมเพื่อที่จะควบคุม กระบวนการระดับน้ำแบบ 4 ถังได้ โดยจะใช้ตัวควบคุม PID 2 ชุด เพื่อควบคุมกระบวนการระดับน้ำ ให้เป็นไปตามค่าเป้าหมาย (Set Point) โดยมีขั้นตอนการออกแบบการทดลองดังนี้

1. เขียนโปรแกรมควบคุมใน LabVIEW ดังรูป ที่ 4.15

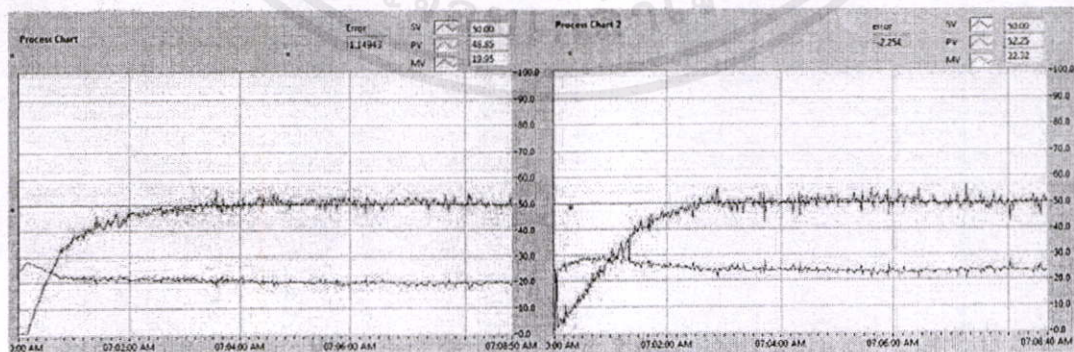


รูปที่ 4.18 แสดงโปรแกรมควบคุมกระบวนการระดับน้ำ 4 ถัง

2. กำหนดค่า Set Point (SV) ให้กับกระบวนการ, $SV = 50$
3. ออกแบบตัวควบคุม PID ทั้ง 2 ชุด ที่ใช้ควบคุมกระบวนการ แล้วนำค่า PID ที่ได้ ไปใส่ในตัวโปรแกรมเพื่อดูผลตอบสนองของกระบวนการ
4. จะได้ผลตอบสนองของกระบวนการควบคุมระดับน้ำแบบ 4 ถัง แสดงดังรูปที่ 4.19

ค่าของ PID ชุดที่แรก $K_C = 0.5, T_i = 1, T_d = 1.6$ กราฟทางด้านซ้าย

ค่าของ PID ชุดที่สอง $K_C = 0.5, T_i = 1, T_d = 1.5$ กราฟทางด้านขวา



เอกสารนี้เป็นเอกสารที่ 4.19 แสดงผลตอบสนองของกระบวนการควบคุมระดับน้ำแบบ PID 2 ชุด ะโยชน์ด้านการศึกษา

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.3 ผลตอบสนองของกระบวนการดังรูปที่ 4.19

กราฟ	Delay time (t_d)	Rise time (t_r)	Peak time (t_p)	Maximum Overshoot (M_p)	Setting time (t_s)
ซ้าย	35 sec	105 sec	210 sec	0 %	235 sec
ขวา	55 sec	115 sec	175 sec	0 %	200 sec

4.4 สรุป

เนื้อหาในบทนี้ได้อธิบายถึงการออกแบบการทดลองในแบบต่างๆและการหาค่าพารามิเตอร์ในการควบคุมกระบวนการในแบบนั้นๆ โดยการลองผิดลองถูก ซึ่งในการใช้วิธีนี้เป็นวิธีที่ต้องใช้เวลาและประสบการณ์ด้วย ต่อมาจะใช้โปรแกรม LabVIEW 2011 เพื่อเขียนโปรแกรมในการควบคุมกระบวนการระดับน้ำ ซึ่งสามารถนำไปใช้ในการสอนและเรียนรู้เกี่ยวกับการควบคุมกระบวนการแบบอัตโนมัติสำหรับวิศวกรหรือผู้ที่สนใจอยากจะศึกษาการทดลองของกระบวนการนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

สรุปการวิจัยและข้อเสนอแนะ

5.1 สรุปผลการทดลอง

ปริญญานิพนธ์นี้ นำเสนอการออกแบบชุดทดลองควบคุมระดับน้ำโดยใช้ไมโครคอนโทรลเลอร์ Arduino เพื่อใช้ในการศึกษากระบวนการควบคุมระดับน้ำในถังและส่วนประกอบในการออกแบบชุดทดลองควบคุมระดับน้ำ การออกแบบชุดทดลองจำเป็นต้องเข้าใจกระบวนการทั้งหมดเสียก่อนแล้วจึงเลือกอุปกรณ์ที่จะใช้ทำงานร่วมกัน อย่างเช่น การเลือกเซนเซอร์ จะต้องรู้ถึงความสูงของถังก่อน เป็นต้น ส่วนการออกแบบตัวควบคุมได้ศึกษาตัวควบคุมแบบพีไอดี เพื่อนำมาควบคุมกระบวนการ โดยจะใช้วิธีลองผิดลองถูก(Trial and Error) ในหาค่าพารามิเตอร์ เพื่อให้ได้ผลตอบสนองที่ดีที่สุดสำหรับชุดทดลองที่ทำการออกแบบมา

ในการทดลองควบคุมกระบวนการ ด้วยตัวควบคุมพีไอดี โดยจะทำการทดลองในหลายๆ แบบคือ ควบคุมกระบวนการระดับน้ำในถัง 1 ถัง, 2 ถัง และ 4 ถัง ตามลำดับ และสามารถควบคุมระดับน้ำในถังของแต่ละการทดลองได้ตามค่าเป้าหมายได้อย่างถูกต้อง โดยไม่เกิดค่าความคลาดเคลื่อน และได้ข้อเปรียบเทียบความแตกต่างของผลตอบสนองที่ได้ของการทดลองในแต่ละแบบ

5.2 ปัญหาและอุปสรรค

1. ปัญหาเกี่ยวกับความร้อนของอุปกรณ์ฮาร์ดแวร์ เช่น ชุดขับเคลื่อนมอเตอร์ดีซี ตัวปั๊มมอเตอร์ และ Switching power supply จากการทำงานต่อเนื่องเป็นเวลานาน ทำให้เสถียรภาพของอุปกรณ์ลดลงไป
2. Port A/D ของ Microcontroller มีจำนวนบิตเพียง 8 บิต จึงทำให้ความสามารถในการประมวลผลยังไม่ดีเท่าที่ควร
3. การออกแบบโครงสร้างของชุดทดลองเป็นสิ่งที่ยากเพราะต้องคิดถึงความเหมาะสมและการเชื่อมต่อของอุปกรณ์ที่ลงตัวพอดีทำให้เสียเวลาในการออกแบบนานเกินไป
4. อุปกรณ์ในกระบวนการมีราคาแพง, มีความเปราะบางเสียหายง่าย และหาซื้อได้ยาก ทำให้ต้องใช้งานอุปกรณ์อย่างระมัดระวัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.3 ข้อเสนอแนะ

ในการออกแบบชุดทดลองของกระบวนการนั้น ควรจะคำนึงถึงทั้งด้านฮาร์ดแวร์และซอฟต์แวร์ว่าให้มีความเหมาะสมกับโครงงาน โดยเฉพาะชุดทดลองในส่วนของโครงสร้างควรคิดให้รอบคอบและคำนวณค่าต่างๆที่นำมาเชื่อมต่อให้เหมาะสม เนื่องจากฮาร์ดแวร์จะส่งผลต่อประสิทธิภาพของระบบ



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

รศ.อาจินต์ น่วมสำราญ. 2554. Virtual Instrumentation Using LabVIEW. กรุงเทพฯ : สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง.

นายเอกชัย มะกาล. 2552. เรียนรู้เข้าใจใช้งานไมโครคอนโทรลเลอร์ตระกูล AVR ด้วย Arduino. กรุงเทพฯ : อีทีที

กิจไพบูลย์ ชิวพันธ์ศรี. 2554. LabVIEW ซอฟต์แวร์เพื่อการพัฒนาาระบบการวัดและควบคุม. กรุงเทพฯ : ซีเอ็ดดูเคชั่น

อธิฏ สันติภูมิโพธา, อภินันท์ จันทิมา, อรรถกร เกิดกิจ. 2554. การออกแบบตัวควบคุมพีซีลोजิกสำหรับกระบวนการระดับน้ำ. “วิทยานิพนธ์ปริญญาวิศวกรรมศาสตรบัณฑิต. กรุงเทพฯ : ภาควิชาวิศวกรรมการวัดคุม สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้