สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

การประยุกต์ใช้งานโปรแกรม LabVIEW สำหรับระบบควบคุมอัตโนมัติ

THE APPLICATION OF LabVIEW FOR AUTOMATION CONTROL SYSTEM

T061422

นายอดิศักดิ์ อ่อนโสม

นายพรชัย จรัสโรจนังกูร

นางสาวจิเรศภา เอี่ยมเงิน

เลขหมู่.....
เลขทะเบียน 61422
วัน,เดือน,ปี 17 ก.ค. 2549

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

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมการวัดคุม

ภาควิชาวิศวกรรมการวัดคุม คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2547

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

THE APPLICATION OF LabVIEW FOR AUTOMATION CONTROL SYSTEM

ADISAK ONSOM

PORNCHAI JARATRODNANGKURN

JIRADPA EIAMNGERN

A THESIS SUBMITTED IN PATIAL FULFILLMENT

OF THE REQUIRMENT FOR THE DEGREE OF

BACHELOR OF ENGINEERING IN INSTRUMENTATION ENGINEERING

DEPARTMENT OF INSTRUMENTATION ENGINEERING

FACULTY OF ENGINEERING

KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

2004

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาควิชาวิศวกรรมการวัดคุม

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ใบรับรองปริญญานิพนธ์

หัวข้อปริญญานิพนธ์ **การประยุกต์ใช้งานโปรแกรม LabVIEW ร่วมกับระบบควบคุมอัตโนมัติ**
THE APPLICATION OF LabVIEW FOR AUTOMATION CONTROL SYSTEM

นักศึกษาผู้จัดทำ

นายอดิศักดิ์	อ่อนโสม	**รหัสประจำตัว**	45015539
นายพรชัย	จรัสโรจนังกูร	**รหัสประจำตัว**	45015513
นางสาวจิเรศภา	เอี่ยมเงิน	**รหัสประจำตัว**	45015500

ปริญญา วิศวกรรมศาสตรบัณฑิต

สาขาวิชา วิศวกรรมการวัดคุม

ปีการศึกษา 2547

อาจารย์ผู้ควบคุมปริญญานิพนธ์	**ลายมือชื่อ**
ผศ.อาจินต์ น่วมสำราญ **รศ.วิริยะ กองรัตน์**	

วัน/เดือน/ปี ที่สอบ วันพฤหัสบดีที่ 24 มีนาคม พ.ศ. 2548

สถานที่สอบ ณ. ห้องสอบปริญญานิพนธ์ ภาควิชาวิศวกรรมการวัดคุม

ภาควิชารับรองแล้ว

(รศ.ประสิทธ์ จุลเสรีวงศ์)

หัวหน้าภาควิชาวิศวกรรมการวัดคุม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อปริญญานิพนธ์	การประยุกต์ใช้งานโปรแกรม LabVIEW สำหรับระบบควบคุมอัตโนมัติ THE APPLICATION OF LabVIEW FOR AUTOMATION CONTROL SYSTEM
นักศึกษาผู้จัดทำ	นายอดิศักดิ์ อ่อนโสม นายพรชัย จรัสโรจนังกูร นางสาวจิเรศภา เอี่ยมเงิน
อาจารย์ที่ปรึกษา	ผศ.อาจินต์ น่วมสำราญ รศ.วิริยะ กองรัตน์
ปีการศึกษา	2547

บทคัดย่อ

ปัจจุบันจะเห็นได้ว่า LabVIEW เริ่มเป็นที่รู้จักและมีบทบาทในกระบวนการอุตสาหกรรมหลายสาขา เนื่องจาก LabVIEW เป็นโปรแกรมที่ใช้งานง่าย และมีประสิทธิภาพสูง อีกทั้งการพัฒนาและการแก้ไขโปรแกรมสามารถทำได้ในเวลาอันสั้น ดังนั้นผู้จัดทำปริญญานิพนธ์จึงมีความสนใจในตัว LabVIEW และได้นำเอา LabVIEW มาประยุกต์ใช้งานร่วมกับระบบควบคุม ซึ่งในปริญญานิพนธ์นี้เป็นการศึกษาการใช้งานโปรแกรม LabVIEW เพื่อเขียนโปรแกรมโดย ควบคุมแรงดันในถังปิดให้ได้ค่าเท่ากับค่าเป้าหมายของกระบวนการ การติดต่อสื่อสารระหว่างกระบวนการกับตัวโปรแกรมบนคอมพิวเตอร์จะเป็นการสื่อสารผ่านทาง พอร์ตอนุกรม (RS-232) เชื่อมต่อกับชุดแปลงสัญญาณอนาลอกเป็นดิจิตอลและดิจิตอลเป็นอนาลอก(RTAD) โดยค่าตัวแปรกระบวนการ(PV) ที่ได้จากทรานสมิตเตอร์จะส่งเข้ามาที่ชุดแปลงสัญญาณอนาลอกเป็นดิจิตอลและดิจิตอลเป็นอนาลอก (RTAD)จากนั้นจะถูกส่งต่อเข้าไปคำนวณหาค่าตัวแปรปรับกระบวนการ (MV) ค่าตัวแปรปรับกระบวนการที่ได้จะส่งกลับไปที่ชุดแปลงสัญญาณอนาลอกเป็นดิจิตอลและดิจิตอลเป็นอนาลอก (RTAD) และสุดท้ายจะส่งเป็นสัญญาณอนาลอกในรูปของกระแสไปที่วาล์วควบคุมเพื่อปรับกระบวนการให้ได้ค่าเท่ากับค่าเป้าหมายตามที่ต้องการหาค่าพารามิเตอร์ของตัวควบคุมจะใช้วิธี Process reaction curve method (Open loop method) ของ Ziegler-Nichols ค่าพารามิเตอร์ที่หามาได้ให้ผลตอบสนองของกระบวนการเป็นที่น่าพอใจ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Thesis Title	The Application of LabVIEW for Automation Control System
Auther	Mr. Adisak Onsom
	Mr. Pronchai Jarasrodnangkurn
	Mrs. Jiradpa Eiamngern
Thesis Advisor	Asst.Prof. Arjin Numsomran
	Assoc.Prof. Viriya Kongratana
Year	2004

ABSTRACT

In present the LabVIEW were use in many kind of industrial because the LabVIEW is easy to use and high efficiency. Especially it can develop and improve in the short time. So that we use the LabVIEW to apply for the control system. This thesis is study to use the LabVIEW program and write program for control the pressure in close tank to the set point. The communication between process and LabVIEW program are communicate by serial port to link with A/D and D/A converter . The process variable from transmitter will sent to the A/D and D/A converter. Then it used in calculation for determine the manipulate variable. After that the manipulate variable will sent back to the A/D and D/A converter again. Finally it will sent in the standard current signal to the control valve for adjust the process variable to the set point. For finding the parameter of controller we use the process reaction curve method of Ziegler-Nichols. The parameter of controller from the calculation give the good respond to our requirement.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

ปริญญานิพนธ์ฉบับนี้สำเร็จลุล่วงได้ด้วยดีเพราะได้รับคำปรึกษาและความอนุเคราะห์จาก ผศ.อาจินต์ น่วมสำราญ และรศ.วิริยะ กองรัตน์ ซึ่งเป็นอาจารย์ผู้ควบคุมปริญญานิพนธ์ ผู้วิจัยรู้สึกทราบซึ้งและกราบขอบพระคุณ เป็นอย่างสูง ที่ได้ให้คำแนะนำและคำปรึกษาที่ถูกต้องในการวิจัย และยังได้รับข้อคิดดีดีในการดำรงชีวิตต่อไป

ขอขอบพระคุณ ผศ.อาจินต์ น่วมสำราญ และรศ.วิริยะ กองรัตน์ ซึ่งได้ให้คำปรึกษาและให้ความช่วยเหลือในการทำวิจัยและช่วยตรวจสอบแก้ไขข้อผิดพลาดต่างๆ พร้อมทั้งวิธีแก้ปัญหา

ขอขอบพระคุณท่านคณาจารย์ภาควิชาวิศวกรรมการวัดคุมทุกท่านที่เอื้อเฟื้ออุปกรณ์และเครื่องมือทั้งหมด ตลอดจนสถานที่ จนสามารถทำปริญญานิพนธ์นี้สำเร็จ

ขอขอบคุณ บิดา มารดา เพื่อนทุกท่านซึ่งคอยเป็นกำลังใจที่ดีในการทำปริญญานิพนธ์ครั้งนี้

คุณค่าและประโยชน์อันพึงมีจากปริญญานิพนธ์ฉบับนี้ผู้วิจัยขอมอบแด่ผู้มีพระคุณทุกท่าน

คณะผู้จัดทำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

หน้า

บทคัดย่อภาษาไทย.....I

บทคัดย่อภาษาอังกฤษ.....II

กิตติกรรมประกาศ.....III

สารบัญ.....IV

สารบัญตาราง.....VIII

สารบัญภาพ.....IX

บทที่ 1 บทนำ.....1

1.1 แนวความคิดในการนำเสนอปริญญานิพนธ์.....1

1.2 วัตถุประสงค์ของปริญญานิพนธ์.....1

1.3 ขอบเขตของปริญญานิพนธ์.....1

1.4 รายละเอียดของปริญญานิพนธ์.....1

บทที่ 2 ที่มาและฟังก์ชันการทำงานบน LabVIEW.....3

2.1 ที่มาของ LabVIEW.....3

2.2 การสร้าง VI.....3

2.2.1 Front panel.....4

2.2.2 Block diagram.....4

2.2.2.1 Node.....4

2.2.2.2 Terminal.....5

2.2.2.3 Wire.....5

2.2.2.3.1 สีของControlหรือIndicator.....5

2.2.2.3.2 รูปแบบของWiring.....5

2.3 Front Panel Toolbar.....6

2.4 Block Diagram Toolbar.....7

2.5 การพัฒนาโปรแกรม.....7

2.5.1 หลักการทำงานของ Dataflow Programming.....7

2.5.2 Data Type 8

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ(ต่อ)

หน้า

2.6 เทคนิคของการDebugging.....10

2.6.1 Finding Error.....10

2.6.2 Execution Highlight.....10

2.6.3 Execution Highlight button Singer Step Through.....11

2.6.4 Button Probe.....11

2.6.5 Breakpoint.....12

2.7 การสร้างโปรแกรมย่อย (SubVI).....12

2.7.1 การใช้งานโปรแกรมย่อย (SubVI).....14

2.8 ตัวอย่างการใช้งานฟังก์ชันที่มีใน LabVIEW.....15

2.8.1 While loop.....15

2.8.2 Waveform Chart.....16

2.8.2.1 Single plot chart.....17

2.8.2.2 Multiple chart plot waveform chart.....17

2.9 การใช้ Shift Register.....18

2.9.1 For Loop.....19

2.9.2 Case Structure.....20

บทที่ 3 การติดต่อสื่อสารและส่งผ่านข้อมูล.....22

3.1 กล่าวนำ.....22

3.2 การสื่อสารข้อมูลแบบอนุกรม.....22

3.2.1 การส่งข้อมูลแบบSimplexและDuplex.....23

3.2.2 โปรโตคอลของการสื่อสารแบบอนุกรม.....24

3.2.3 มาตรฐานสัญญาณอนุกรมแบบ RS232C.....25

3.2.4 พอร์ทสื่อสารอนุกรม.....26

3.3 ชุด REMOTE TERMINAL ANALOG DIGITAL : RTAD.....27

3.3.1 ส่วนประกอบที่สำคัญชุด

REMOTE TERMINAL ANALOG DIGITAL : RTAD.....28

3.3.2 โปรโตคอลในการติดต่อสื่อสาร.....29

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ(ต่อ)

หน้า

บทที่ 4 ความรู้เบื้องต้นเกี่ยวกับการควบคุมแบบป้อนกลับด้วยตัวควบคุม PID.....30

4.1 เครื่องควบคุม.....30

4.2 .หลักการของตัวควบคุม.....30

4.3 กริยาต่าง ๆ ของการควบคุม.....31

4.3.1 การควบคุมแบบ Proportional.....31

4.3.2 การควบคุมแบบ Proportional + Integral.....32

4.3.3 การควบคุมแบบ Proportional + Integral + Derivative.....32

4.4 การปรับแต่งหาค่าพารามิเตอร์ของเครื่องควบคุม.....32

4.4.1 วิธีการแบบ Reaction Curve (Open Loop Method).....33

4.4.2 วิธี Ultimate Method ของ Ziegler-Nichols (Closed Loop).....34

4.4.3 วิธี Trial and Error.....36

4.5 กระบวนการควบคุมแรงดัน.....37

บทที่ 5 โปรแกรมควบคุมกระบวนการ.....40

5.1 แนวคิดและหลักการในการออกแบบโปรแกรม.....40

5.2 โปรแกรมควบคุมแรงดัน.....41

5.2.1 Front panel ของโปรแกรมควบคุมแรงดัน.....41

5.2.2 Block diagram ของโปรแกรมควบคุมแรงดัน.....42

5.2.2.1 ส่วนของโปรแกรมตัวควบคุม.....43

5.2.2.2 ส่วนของโปรแกรมติดต่อสื่อสารกับ RTAD.....44

5.2.2.2.1 โปรแกรมสั่ง Write Analog.....44

5.2.2.2.2 โปรแกรมสั่ง Read Analog.....45

บทที่ 6 ผลการทดลองหาค่าพารามิเตอร์ของตัวควบคุม.....47

6.1 การหาค่าพารามิเตอร์ของตัวควบคุมโดยวิธี Process Reaction Curve.....47

6.2 ผลการทดลองหาค่าพารามิเตอร์ของตัวควบคุม.....48

6.3 ผลตอบสนองของกระบวนการเมื่อใช้ตัวควบคุมแบบ PI.....49

6.4 ผลตอบสนองของกระบวนการเมื่อใช้ตัวควบคุมแบบ PID.....51

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ(ต่อ)

หน้า

6.5 สรุปผลการทดลอง.....52

บทที่ 7 สรุปผลการวิจัยและข้อเสนอแนะ.....54

7.1 บทสรุป.....54

7.2 ข้อเสนอแนะและแนวทางพัฒนา.....54

บรรณานุกรม.....55

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

ตารางที่ **หน้า**

3.1 แสดงขาสัญญาณต่าง ๆ ในการรับส่งข้อมูลแบบอนุกรม RS-232 Standard.....27

4.1 แสดงสูตรสำหรับหาค่าพารามิเตอร์ของตัวควบคุม PIDตามวิธี Process Reaction Curve.....34

4.2 แสดงสูตรสำหรับหาค่าพารามิเตอร์ของตัวควบคุม PIDตามวิธี Ultimate Method.....35

6.1 แสดงสูตรสำหรับหาค่าพารามิเตอร์ของตัวควบคุม PIDตามวิธีProcess Reaction Curve....48

6.2 แสดงค่าพารามิเตอร์ของตัวควบคุมที่หาได้จากวิธี Process Reaction Curve Method.....52

6.3 แสดงค่าพารามิเตอร์ของตัวควบคุมหลังการปรับแต่ง.....53

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญภาพ

ภาพที่ **หน้า**

2.1 แสดงFront panel และ Block Diagram.....4

2.2 แสดงFront Panel Toolbar.....6

2.3 แสดงหลักการทำงานของDataflow Programming.....7

2.4 แสดงData Type 8

2.5 แสดงการWiring ที่ผิด.....9

2.6 แสดงError List.....9

2.7 แสดงFinding Error.....10

2.8 แสดงExecution Highlight.....10

2.9 แสดงExecution Highlight Button Singer Step Through.....11

2.10 แสดงExecution Highlight Button Singer Step Through.....11

2.11 แสดBreakpoint.....12

2.12 แสดงIcon Editor.....13

2.13 แสดงTerminal Pop Up Menu.....14

2.14 แสดงการเรียกใช้งานโปรแกรย่อย(SubVI).....14

2.15 แสดงSubplateของ Function While loop.....15

2.16 แสดงFunction While Loop บนหน้า Diagram.....16

2.17 แสดงFunction 2.9.2 Waveform Chartบนหน้าFront Panel.....16

2.18 แสดงFunction Strip Chart, Scope Chart และ Sweep Chart บนหน้า Front Panel.....17

2.19 แสดงFunction1 Single Plot Chart.....17

2.20 แสดงFunction1 Multiple Chart Plot Waveform Chart.....17

2.21 แสดShift Register.....18

2.22 แสดงการทำงานของShift Register.....18

2.23 แสดงStructure Subplateของ Function For Loop 19

2.24 แสดงFor Loop.....20

2.25 แสดง For Loopในการสุ่มค่าตัวเลขตั้งแต่ 1 ถึง 100 จำนวน 20 ครั้ง.....20

2.26 แสดงStructure SubplateของCase Structure.....21

2.27 แสดง Case Structureซึ่งสามารถรับค่าได้ทั้งที่เป็นNumeric , Boolean , String.....21

3.1 แสดงการติดต่อสื่อสารผ่านพอร์ทอนุกรมRS232.....22

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญภาพ(ต่อ)

ภาพที่ **หน้า**

3.2 แสดงการสื่อสารข้อมูลอนุกรมแบบต่างๆ.....23

3.3 แสดงแบบมาตรฐานการสื่อสารข้อมูลแบบ Synchronous 25

3.4 แสดงลักษณะสมบัติทางไฟฟ้าของการ Interface แบบRS232.....25

3.5 แสดงตำแหน่งขาของDB-9 และ DB-25.....26

3.6 แสดงชุดRTAD ใช้งานจริง.....27

3.7 แสดงส่วนประกอบของชุดRTAD.....28

4.1 แสดงโครงสร้างพื้นฐานของตัวควบคุPID.....31

4.2 แสดงโครงสร้างของระบบที่ควบคุมด้วยตัวควบคุPID.....32

4.3 แสดงผลตอบสนองรูปตัวSเมื่อใช้วิธี Process Reaction Curve.....33

4.4 แสดงผลตอบสนองเวลาเกิดการแกว่งอย่างต่อเนื่องเมื่อปรับโดยใช้วิธี Ultimate Metho.....35

4.5 แสดงกระบวนการควบคุมแรงดัน.....37

4.6 แสดงส่วนประกอบของกระบวนการควบคุมแรงดัน.....38

4.7 แสดงการควบคุมแรงดันด้วยโปรแกรมLabVIEWบนคอมพิวเตอร์.....38

4.8 แสดง P & ID ของกระบวนการ.....39

4.9 แสดง Block diagram ของกระบวนการควบคุมแรงดัน.....39

5.1 แสดงการติดต่อสื่อสารระหว่างคอมพิวเตอร์กับกระบวนการ.....40

5.2 แสดงFlow Chartการทำงานของโปรแกรมหลัก.....41

5.3 แสดงFront panelของโปรแกรมควบคุมแรงดัน.....42

5.4 แสดงBlock diagramของโปรแกรมควบคุมแรงดัน.....43

5.5 แสดงโปรแกรมตัวควบคุม.....44

5.6 แสดงโปรแกรมสั่งWrite Analog.....44

5.7 แสดงโปรแกรมในCON_TO_PLANT.vi 45

5.8 แสดงโปรแกรมสั่งRead Analog.....45

5.9 แสดงโปรแกรมใน DATA From RT.vi..... 46

6.1 แสดงผลตอบสนองรูปตัว S เมื่อใช้วิธี Process Reaction Curve.....47

6.2 แสดงผลตอบสนองของกระบวนการเมื่อป้อนค่า MV จาก 3% ไปเป็น 10%.....49

6.3 แสดงผลตอบสนองของกระบวนการเมื่อใช้ตัวควบคุมแบบ PI.....49

6.4 แสดงผลตอบสนองของกระบวนการเมื่อปรับค่าพารามิเตอร์อย่างละเอียด.....50

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญภาพ(ต่อ)

ภาพที่ **หน้า**

6.5 แสดงผลตอบสนองของกระบวนการเมื่อป้อนค่า SV เป็น Unit Step.....50
6.6 แสดงผลตอบสนองของกระบวนการเมื่อใช้ตัวควบคุมแบบ PID.....51
6.7 แสดงผลตอบสนองของกระบวนการเมื่อเพิ่มTi และลด Td.....51
6.8 แสดงผลตอบสนองของกระบวนการเมื่อปรับค่า SV.....52

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

1.1 ความเป็นมาและเหตุจูงใจของการวิจัย

ปัจจุบันจะเห็นได้ว่า LabVIEW เริ่มเป็นที่รู้จักและมีบทบาทในกระบวนการอุตสาหกรรมหลายสาขา เนื่องจาก LabVIEW เป็นโปรแกรมที่ใช้งานง่าย และมีประสิทธิภาพสูง อีกทั้งการพัฒนาและการแก้ไขโปรแกรมสามารถทำได้ในเวลาอันสั้น ดังนั้นผู้จัดทำปริญญานิพนธ์มีความสนใจในตัว LabVIEWจึงได้นำเอา LabVIEW มาประยุกต์ใช้งานร่วมกับระบบควบคุมเนื่องจากตัวผู้จัดทำเองก็เรียนในสาขาวิชาวิศวกรรมการวัดคุม จึงเป็นถือเอาโอกาสนี้นำเอา LabVIEW มาประยุกต์ใช้งานร่วมกับระบบควบคุมนอกจากจะเป็นการศึกษาตัว LabVIEW ยังเป็นการสร้างความรู้ความเข้าใจในระบบควบคุมมากขึ้น สร้างทักษะในการปรับแต่งค่าพารามิเตอร์ของตัวควบคุม นับว่าผู้จัดทำได้รับประโยชน์จากการทำปริญญานิพนธ์นี้อย่างมาก

1.2 วัตถุประสงค์ของปริญญานิพนธ์

ปริญญานิพนธ์นี้จะเป็นการศึกษาการใช้งาน LabVIEW เพื่อที่จะประยุกต์ใช้งานร่วมกับระบบควบคุม โดยจะใช้ LabVIEW เขียนโปรแกรมเพื่อควบคุมกระบวนการให้ได้ตามค่าเป้าหมายของการควบคุมที่ตั้งไว้ ซึ่งกระบวนการที่จะควบคุมก็คือ กระบวนการควบคุมแรงดันในถังปิดให้ได้ค่าตามที่ต้องการไม่ว่าโหลดการใช้งาน จะเปลี่ยนแปลงหรือมีสิ่งรบกวน(Disturbance) โปรแกรมที่เขียนขึ้นโดย LabVIEW จะต้องพยายามรักษาแรงดันในถังให้คงที่อยู่ตลอดเวลา

1.3 ขอบเขตของปริญญานิพนธ์

ขอบเขตของปริญญานิพนธ์เล่มนี้จะเป็นการศึกษาการใช้งาน LabVIEW เพื่อเขียนโปรแกรมควบคุมกระบวนการ รวมทั้งศึกษาวิธีการติดต่อสื่อสารกับตัวเชื่อมต่อกับกระบวนการ (RTAD) ศึกษาทำความเข้าใจกระบวนการที่จะทำการควบคุม หาค่าพารามิเตอร์ของตัวควบคุม และสุดท้ายก็ใช้โปรแกรมที่เขียนขึ้นโดย LabVIEW ควบคุมกระบวนการให้ได้ตามค่าเป้าหมายของกระบวนการที่ตั้งไว้

1.4 ขั้นตอนการศึกษา

ขั้นตอนการทำปริญญานิพนธ์เริ่มต้นจากเรียนรู้วิธีการใช้งาน LabVIEW ศึกษาการใช้ฟังก์ชันที่เป็นประโยชน์สามารถประยุกต์ใช้งานร่วมกับระบบควบคุมได้ ศึกษาการติดต่อสื่อสาร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ทาง RS-232 เพื่อที่จะติดต่อกับตัวเชื่อมต่อกระบวนการ ซึ่งตัวเชื่อมต่อกระบวนการจะทำหน้าที่แปลงสัญญาณอนาลอกเป็นดิจิตอลและดิจิตอลเป็นอนาลอก ดังนั้นจึงต้องศึกษาโปรโตคอลที่จะใช้ในการติดต่อสื่อสารด้วยเมื่อศึกษาทั้งสองส่วนเป็นที่เข้าใจแล้ว จากนั้นจึงศึกษาและทำความเข้าใจกับกระบวนการที่จะทำการควบคุม เพื่อที่จะสามารถเขียนโปรแกรมควบคุมกระบวนการได้อย่างมีประสิทธิภาพและครอบคุมทุก เงื่อนไข เมื่อเขียนโปรแกรมควบคุมกระบวนการเสร็จสมบูรณ์จึงทำการหาค่าพารามิเตอร์ของตัวควบคุม ซึ่งวิธีการที่ใช้หาค่าพารามิเตอร์ของตัวควบคุมจะใช้วิธี Process reaction curve method (Open loop method) ของ Ziegler-Nicholsใช้ค่าพารามิเตอร์ที่คำนวณได้กับตัวควบคุม สังเกตผลตอบสนองของกระบวนการ ผลตอบสนองของกระบวนการที่ได้อยู่ในระดับที่ดี แต่ถึงอย่างไรได้ทำการปรับแต่งค่าพารามิเตอร์ที่ได้เพื่อให้ผลตอบสนองดียิ่งขึ้นและเป็นการศึกษาปฏิกิริยาของค่าพารามิเตอร์แต่ละตัวที่มีผลต่อระบบควบคุม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2
ที่มาและฟังก์ชันการทำงานบน LabVIEW

2.1 ที่มาของ LabVIEW

LabVIEW ย่อมาจากLaboratory Virtual Instrument Engineering Workbench โปรแกรมที่พัฒนาขึ้นโดยใช้ LabVIEW จะเรียกว่า Virtual Instrument หรือเรียกย่อ ๆ ว่า VI ซึ่งหมายถึงเครื่องมือวัดเสมือน

LabVIEW จะม Front Panel ซึ่งเปรียบเสมือนได้กับสิ่งที่ผู้ใช้จะเห็นและควบคุมการทำงาน ผู้ใช้สามารถสร้างรูปแบบขึ้นเองได้อย่างรวดเร็วเพราะ LabVIEW มีส่วนประกอบต่าง ๆ ที่ใช้สำหรับออกแบบหน้าจอมากมาย เช่น จอแสดงผลแบบออก Oscilloscope , ปุ่มหมุน (Dial) และ Switch เป็นต้น โดย LabVIEW จะแสดงผลและควบคุมการทำงานผ่านทางคอมพิวเตอร์พื้นที่ส่วนที่เขียนโปรแกรมจะเรียกว่า Block Diagram เปรียบเสมือนกับ Hardware ภายในเครื่องมือวัด โดย LabVIEW จะเขียนโปรแกรมโดยอาศัยรูปภาพ LabVIEW อาศัยหลักการทำงานของเครื่องมือวัดหรือการวัดคุมทำให้ผู้ใช้สามารถออกแบบรูปแบบโปรแกรมตามที่ผู้ใช้ต้องการหลักการดังกล่าวแบ่งออกเป็น 3 ส่วนใหญ่ ๆ คือ

1. Acquisition ซึ่งเป็นส่วนที่รับข้อมูล (Input) จากสิ่งแวดล้อมภายนอกเข้าสู่ระบบในที่นี้คือ คอมพิวเตอร์ โดยข้อมูลที่เข้าสู่ระบบนี้อาจจะมาจาก การ์ด DAQ (สำหรับสัญญาณทางไฟฟ้า), การ์ด IMAQ (สำหรับข้อมูลประเภทรูป) หรือ GPIB (สำหรับควบคุมเครื่องมือวัด)

2. หลังจากที่ได้รับข้อมูลแล้วอาจจะผ่านฟังก์ชันในการทำ Analysis หรือ วิเคราะห์ข้อมูลซึ่งจะแสดงผลในรูปที่สื่อความหมายในสิ่งที่ผู้ใช้งานสามารถนำไปแสดงแทนสื่อที่วัดได้และใช้งานได้

3. Presentation คือ การแสดงผลในรูปแบบที่เป็นประโยชน์ต่อผู้ใช้งาน โดยอาจแสดงบนหน้าจอคอมพิวเตอร์ เช่น DMM (Digital Multimeter) แสดงผลเฉพาะสัญญาณที่วัดได้โดยไม่จำเป็นต้องรู้ความสัมพันธ์กับเวลา หรือ Oscilloscope แสดงผลของข้อมูลที่สัมพันธ์กับเวลา หรือ Spectrum Analyzer จะแสดงสัญญาณในรูปความถี่หรือการพิมพ์ออกมาเป็นรายงานหรือเก็บข้อมูลในฮาร์ดดิสก์

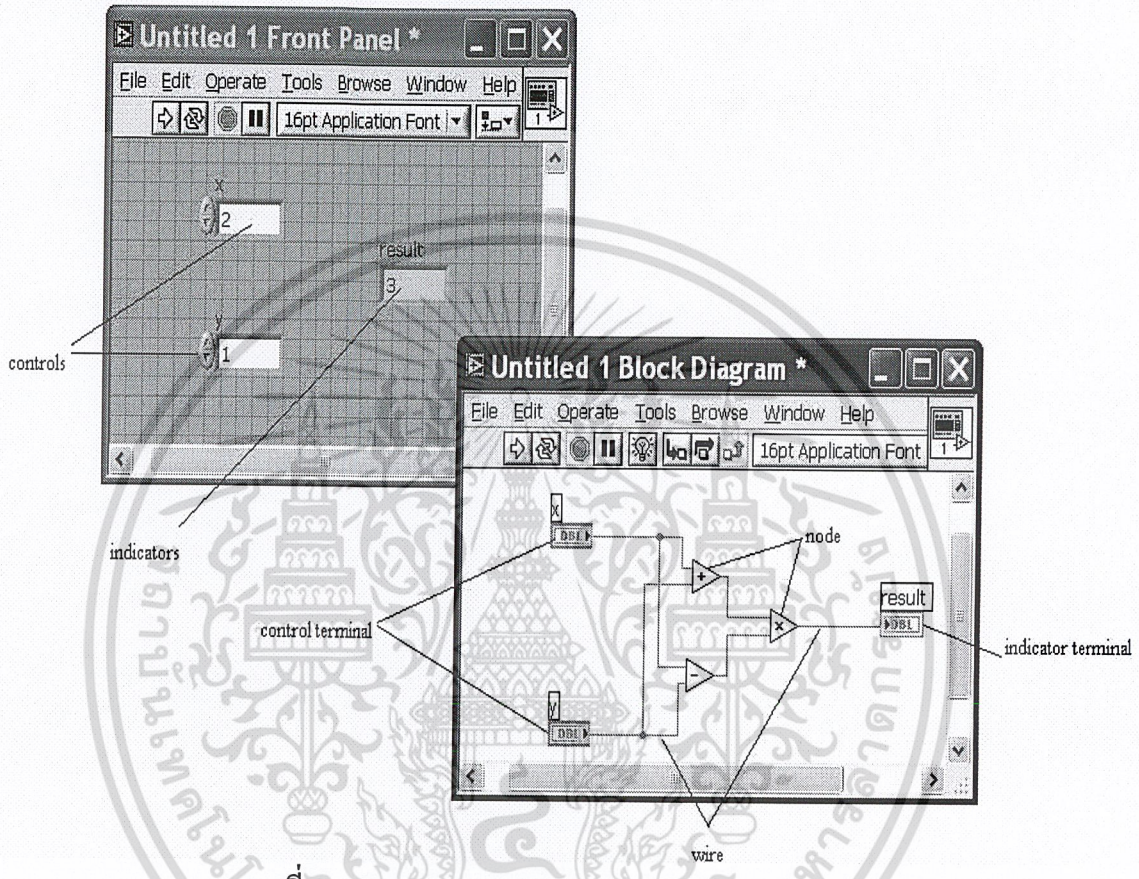
2.2 การสร้าง VI

VI (Virtual Instrument) ประกอบด้วยสองส่วนคือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.1 Front panel

ใช้สำหรับสร้างControlหรือ Indicators ของ Numeric , Boolean , String , Graph , Table หรืออื่นๆที่อยู่บนControl Plate ลงบน Front Panel เพื่อใช้รับติดต่อระหว่างคอมพิวเตอร์กันผู้ใช้งาน (User Interface)



ภาพที่ 2.1 แสดง Front panel และ Block diagram

2.2.2 Block diagram

เป็นส่วนที่เก็บตัวโปรแกรม(Source Code) โดยจะอยู่ในรูปแบบของกราฟิกโปรแกรมมิ่ง ประกอบด้วย Nodes, Terminal และ Wires

2.2.2.1 Node

แบ่งเป็น 4 ชนิด คือ

1. Function ติดตั้งมาพร้อมกับ LabVIEW
2. SubVI Node คือ VI ที่สร้างขึ้นเองและถูกนำมาใช้เป็น SubVI ในภายหลัง
3. Structure เช่น Case , loop , Sequence ที่ใช้ในการควบคุมการ ทำงานของโปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. Code Interface Node (CIN) ใช้ติดต่อระหว่าง LabVIEW และ User Supplied Code ของภาษา C

2.2.2.2 Terminals

เป็นตัวรับ-ส่งข้อมูลระหว่าง Block Diagram และFront Panel หรือระหว่าง Node ใน Block Diagram แบ่งเป็น 2 ชนิด คือ Control หรือ Indicator Terminal ซึ่งจะถูกสร้างจากFront Panel 2.2.1 Node Terminal จะเกิดจากฟังก์ชั่นต่างๆที่สร้างขึ้นบน Block Diagram

2.2.2.3 Wires

คือทางเดินของข้อมูลระหว่าง Terminals ซึ่งเป็นแบบส่งทางเดียว สีและรูปแบบของWires จะบอกถึงชนิดของข้อมูลที่ส่งผ่านโดยใช้ Wiring Tool (รูปหลอดด้าย) ใน Tool Palette เป็นตัวต่อเชื่อมข้อมูลระหว่างTerminal ต่างๆ

2.2.2.3.1 สีของ Control หรือ Indicator

1. สีน้ำเงิน แทนตัวเลขที่เป็นจำนวนเต็ม
2. สีส้ม แทนตัวเลขที่มีค่าทศนิยม
3. สีเขียว แทนLogic หรือ Boolean
4. สีชมพู แทนค่าของString

2.2.2.3.2 รูปแบบของWiring

1. เส้นบางแทนค่าจำนวนเดียว
2. เส้นหนาแทนค่าของอาร์เรย์ ขนาด 1มิติ
3. เส้นคู่แทนค่าของอาร์เรย์ ขนาด 2 มิติ

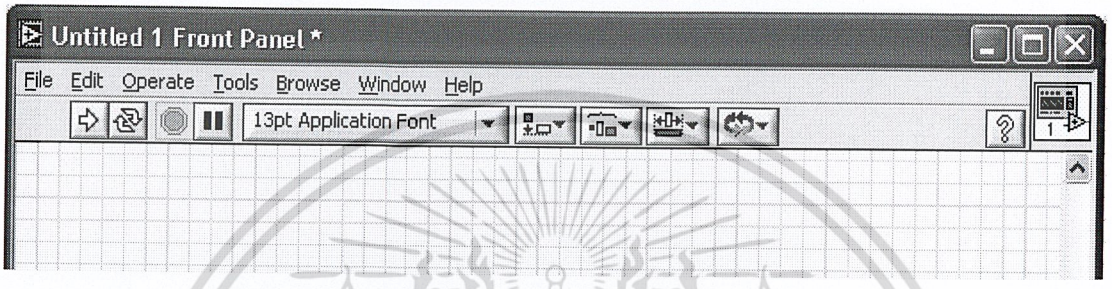
การWiring ทำโดยใช้ Wiring Tool เลื่อนไปให้ตรง Terminals แรก สังเกตให้ Terminals กระพริบแล้วคลิกเมาส์ จากนั้นให้ลากไปยังTerminals ที่สอง โดยจะเกิดเส้นประสีดำลากตามcursor ให้คลิกเมาส์บนTerminalsที่สองในขณะที่ Terminalsนั้นกำลังกระพริบ ข้อแตกต่างระหว่าง Control กับIndicatorบน Block Diagram คือ Control จะมีขอบเป็นเส้นหนา ส่วน Indicator จะมีขอบเป็นเส้นบาง การเก็บข้อมูลโดยปกติ LabVIEW จะเก็บเป็นไฟล์ 2 ชนิด คือ

1. VI file : เก็บเป็นไฟล์ที่มีนามสกุล เป็น *.VI
2. VI libraries : เป็นไลบารี่ไฟล์ที่รวมเอาหลายๆ VI ไว้ในไฟล์เดียวกันโดยจะมี นามสกุลเป็น *.lbb

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

LabVIEWสามารถเขียนโปรแกรมได้จากการสร้าง Flont Panel ซึ่งจะใช้เป็นที่สร้าง Control และ Indicator สำหรับรับและแสดงข้อมูลระหว่างผู้ใช้งานและเครื่องคอมพิวเตอร์ ส่วน Block Diagram จะเป็นที่สำหรับใช้เขียนโปรแกรมเพื่อให้คอมพิวเตอร์ทำงานตามที่ต้องการ ในการเก็บโปรแกรมลงไฟล์ทำได้ทั้งแบบ VI file หรือ VI libraries

2.3 Front Panel Toolbar



ภาพที่ 2.2 แสดง Front Panel Toolbar

โปรแกรมสามารถ Run ได้เมื่อ Click ปุ่มนี้บน Toolbar

โปรแกรมกำลัง Run อยู่

โปรแกรมกำลัง Run อยู่ SubVI

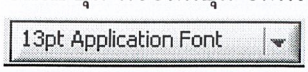
โปรแกรม Error ถ้า Click จะแสดงรายการ Error List

โปรแกรมสามารถ Run แบบต่อเนื่อง (Continous) เมื่อ Click ปุ่มนี้

กำลัง Run อย่างต่อเนื่องจนกระทั่ง Click ปุ่ม Stop/Pause

เมื่อโปรแกรมกำลัง Run อยู่สามารถหยุดการทำงานโปรแกรมโดยการ Click ปุ่ม Stop

หยุดชั่วคราว เพื่อตรวจสอบการทำงานของโปรแกรม ซึ่งสามารถ Run ต่อไปได้เมื่อ Click ปุ่ม หรือกดปุ่ม อีกครั้ง

 เลือกหรือแก้ไขรูปแบบอักษรที่ต้องการ

จัดแนวของสิ่งต่าง ๆ (Objects)

จัดระยะทางระหว่างสิ่งต่าง ๆ (Objects)

ปรับขนาดสิ่งต่าง ๆ ที่อยู่บน Front Panelหลาย ๆ สิ่ง ซ้อนกัน

จัดลำดับของ Objects ที่วางซ้อนกัน

แสดง Context Help

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4 Block Diagram Toolbar

สิ่งที่เพิ่มเติมจาก Front Panel Toolbar ในส่วนของ Block Diagram มีดังต่อไปนี้

Highlight Execution เมื่อ Click แล้ว ในการ Run Program จะทำให้โปรแกรม Run

ช้าลงเพื่อให้เราเห็นถึงการไหลของข้อมูล และทำงานของโปรแกรมเมื่อ Click แล้ว จอออกจาก Mode ที่โปรแกรมRun ช้า (Highlight Execution)

เป็นการ Run โปรแกรมแบบทีละคำสั่ง และจะลงไปถึงคำสั่งของ SubVI

เป็นการ Run โปรแกรมแบบทีละคำสั่ง แต่จะไม่ลงไปถึงคำสั่งของ SubVI

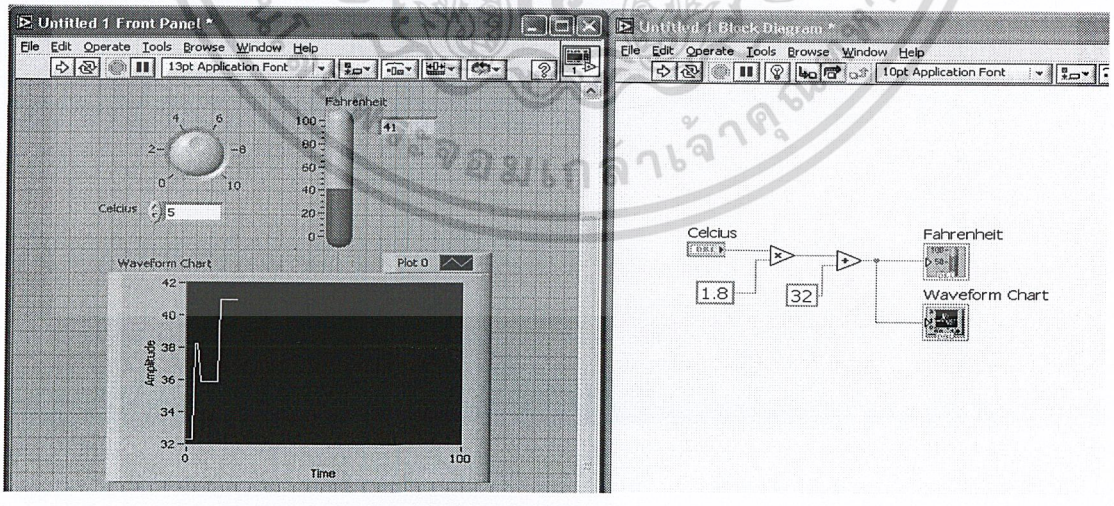
ให้ Run จนออกจาก Loop และ SubVI

2.5 การพัฒนาโปรแกรม

2.5.1 หลักการทำงานของ Dataflow Programming

โปรแกรมที่เขียนขึ้นด้วย LabVIEW จะทำงานโดยอาศัยหลักการของ Dataflow ซึ่งมีหลักการดังต่อไปนี้ คือ

1. ฟังก์ชันหรือ SubVI จะทำงานเมื่อมีข้อมูล (Input)
2. เมื่อฟังก์ชันหรือ SubVI ทำงานเสร็จจะให้ข้อมูล (Output) ไปยังฟังก์ชันหรือ SubVI อื่นๆที่ต้องการข้อมูล
3. ข้อมูลจะถูกส่งผ่านโดย Wires



ภาพที่ 2.3 แสดงหลักการทำงานของ Dataflow Programming

จากภาพขั้นตอนการทำงานของโปรแกรมจะเป็นดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.ฟังก์ชัน Multiply จะทำงานก่อนฟังก์ชันบวก เพราะฟังก์ชัน Multiply มีข้อมูล (Input)พร้อม แต่ฟังก์ชันบวกต้องรอ Output จากฟังก์ชัน Multiply (หลักการ Dataflow ข้อที่1)

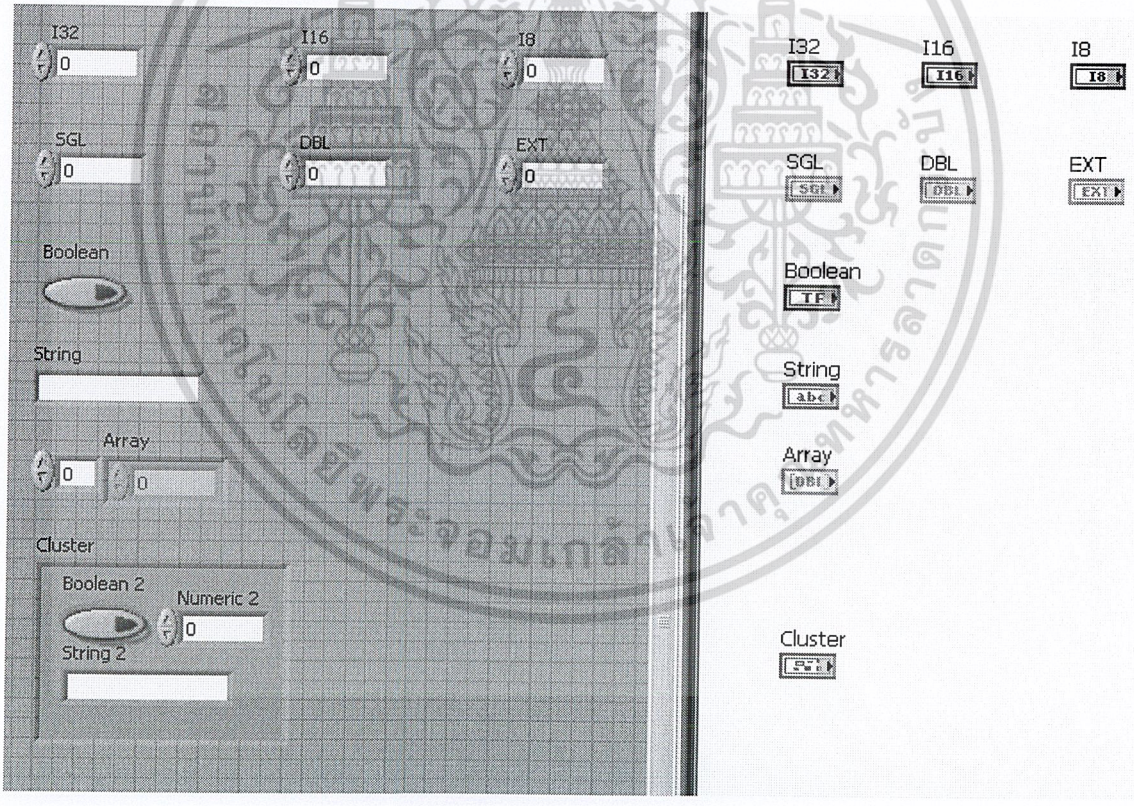
2. หลังจากฟังก์ชัน Multiply ทำงานเสร็จจะส่งผ่านข้อมูลไปยังฟังก์ชันบวก (หลักการ Dataflow ข้อที่ 2 และ 3)

3. ฟังก์ชันบวกทำงาน (หลักการ ข้อที่1) เพราะมีข้อมูลพร้อม

4. หลังจากฟังก์ชันบวกทำงานจะส่งผลลัพธ์ไปให้ Terminal ทั้ง 2 คือ Fahrenheit และ Waveform Chart พร้อมกัน (หลักการ ข้อที่ 2 และ 3)

2.5.2 Data Type (ประเภทของข้อมูล)

เช่นเดียวกันกับภาษาโปรแกรมอื่นๆ เช่น ภาษา C ที่มีประเภทของข้อมูล ประเภทของข้อมูลใน LabVIEW ที่สำคัญจะประกอบด้วย Numeric , Boolean , String และ Waveform ซึ่งแต่ละรูปแบบของประเภทข้อมูลจะมีการแยกสีให้แตกต่างกัน (Color Code)

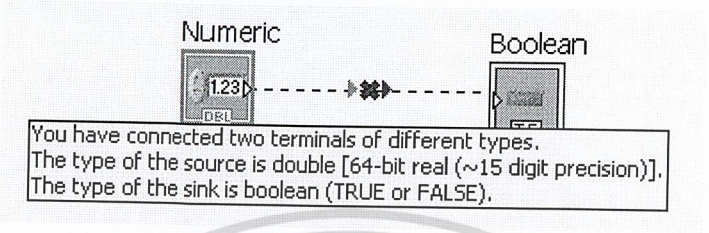


ภาพที่ 2.4 แสดง Data Type (ประเภทของข้อมูล)

ซึ่งแต่ละประเภทของข้อมูลจะมีฟังก์ชันที่แยกตามประเภทกัน รวมทั้งสีของ Wiring จะขึ้นอยู่กับประเภทของข้อมูล ประเภทของข้อมูลมีความสำคัญในการเขียนโปรแกรม เพราะ

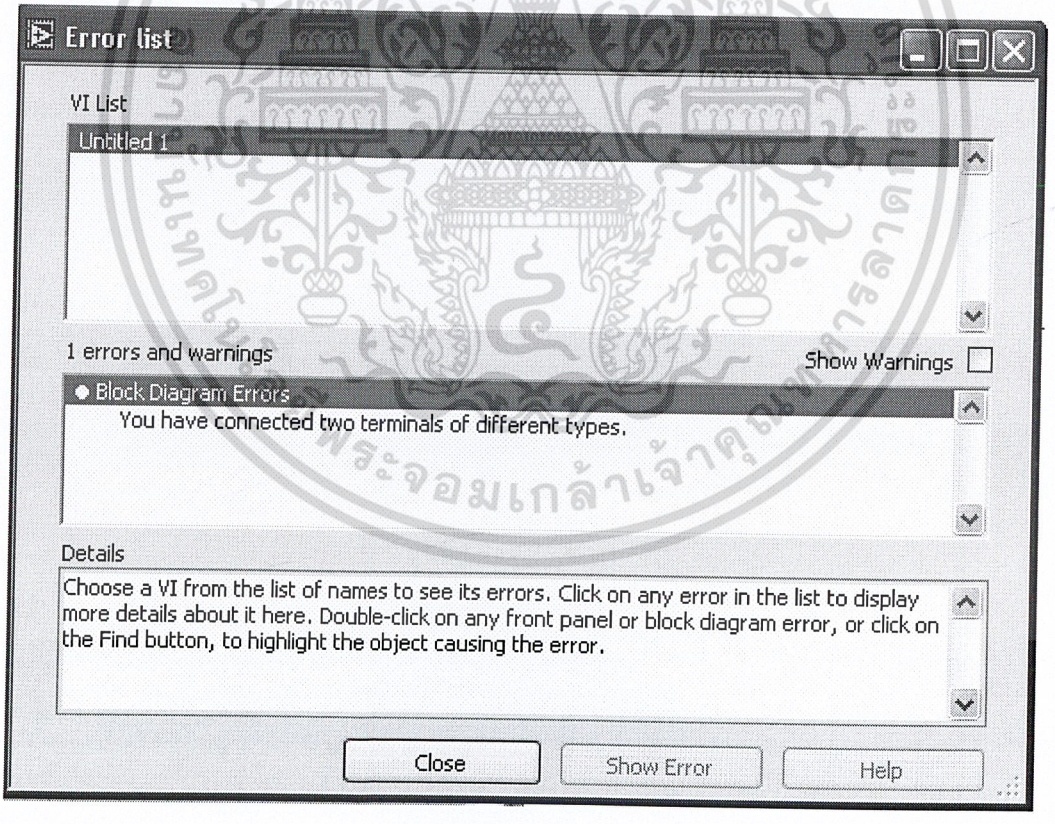
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Function หรือ SubVI รับรูปแบบ Input และให้ Output แตกต่างกันไป เช่น ฟังก์ชัน Multiply รับข้อมูลประเภทตัวเลขหรือ Array ของตัวเลข ซึ่งถ้านำเอาข้อมูลประเภทตัวอักษร (String Data Type) มาป้อนที่ด้าน Input Terminal แล้ว LabVIEW จะแสดงWires เป็นเส้นประ ซึ่งเราสามารถดูคำอธิบายได้โดยการเลื่อน Cursor ไปเหนือเส้นประนั้น



ภาพที่ 2.5 แสดงการ Wiring ที่ผิด

หรือ Click ที่ปุ่ม Run เพื่อให้แสดง Error List



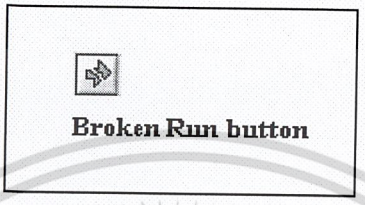
ภาพที่ 2.6 แสดงError List

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.6 เทคนิคของการ Debugging

การ Debugging คือ การตรวจสอบหาข้อผิดพลาดของโปรแกรม ในกรณีที่โปรแกรมที่เขียนขึ้นมีปัญหาโดย LabVIEW จะมีเครื่องมือต่างๆ ในการ Debugging ดังนี้

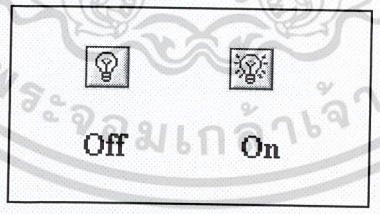
2.6.1 Finding Error



ภาพที่ 2.7 แสดง Finding Error

เมื่อโปรแกรมไม่สามารถ Run ได้ เครื่องหมายลูกศรแตก (Broken arrow) จะปรากฏขึ้นที่ Run Button บน Front Panel เราสามารถดูว่ามีข้อผิดพลาดอะไรบ้าง โดยกดปุ่มเครื่องหมายลูกศรแตกนั้น และเมื่อคลิกเมาส์ที่ข้อผิดพลาดเพื่อเลือก และกดปุ่ม Find เพื่อที่ให้เกิด Highlight ที่ Object หรือ Terminal ที่ตรวจพบว่ามีปัญหา

2.6.2 Execution Highlight

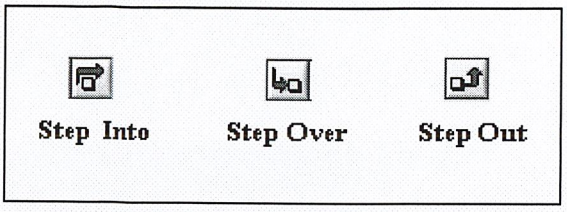


ภาพที่ 2.8 แสดง Execution Highlight

เราสามารถให้ Block Diagram แสดงการทำงานให้เห็นได้โดยการคลิกที่ปุ่มExecution Highlight ปุ่มนี้จะเปลี่ยนไปเป็นเหมือนที่แสดงในรูป Execution Highlight มักจะใช้ร่วมกับ Singer Step Mode เพื่อดูข้อมูลที่เกิดขึ้นขณะที่โปรแกรมรัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.6.3 Execution Highlight Button Singer Step Through

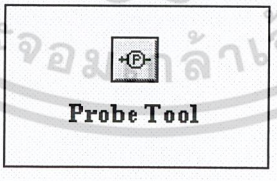


ภาพที่ 2.9 แสดง Execution Highlight button Singer Step Through

สำหรับวัตถุประสงค์ในการDebugging เรามักจะต้องการดูการทำงานของ Block Diagram Node ต่อ Node ซึ่งเรียกกันว่า Singer Stepping ในการที่จะให้โปรแกรมทำงานแบบ Single Step ให้กดที่ปุ่ม Step Into หรือ Step Over ซึ่งจะมีผลให้ Node แรกที่เริ่มทำงานกระพริบหมายถึงพร้อมที่จะทำงานต่อไป

เมื่อต้องการให้โปรแกรมทำงานต่อ ก็ให้กดปุ่ม Step Into หรือ Step Over อีกครั้ง ถ้าในกรณีที่ Node ต่อไปที่จะทำงานเป็น SubVI และเรากด Step Over โปรแกรมจะทำงานโดยข้ามการทำงานใน SubVI นั้นไปเลย และส่งเพียงผลที่ได้จากการทำงานของ SubVI ออกมาเท่านั้น แต่ถ้าเลือกกด Step Into โปรแกรมจะทำการเปิด SubVI นั้นและแสดงการทำงานของ SubVI นั้นกดปุ่ม Step Out เพื่อให้โปรแกรมจบการทำงานของ Block Diagram นั้น ๆ

2.6.4 Button Probe

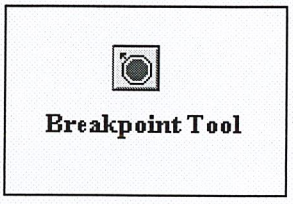


ภาพที่ 2.10 แสดง Execution Highlight Button Singer Step Through

เราสามารถที่จะดูข้อมูลที่อยู่ในจุดใดของ Block Diagram ได้โดยใช้ Probe การสร้าง Probe ทำได้โดยเลือก Probe Tool จาก Tool Platte และ Click ลงบน Wires ที่ต้องการดูข้อมูล หรืออาจจะสร้างได้อีกวิธีคือ Click Mouse ขวาตรงตำแหน่งที่ต้องการสร้าง Probe เพื่อแสดง Pop Up Menuและเลือก Custom Probe

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.6.5 Breakpoint



ภาพที่ 2.11 แสดง Breakpoint

บางครั้งเราอาจจะต้องการที่จะหยุดการทำงานของโปรแกรมที่จุดใดๆในโปรแกรม เพื่อใช้ Probe ดูข้อมูลหรือ Run แบบ Single Step ต่อไป ให้ใช้ Breakpoint Tool คลิกในตำแหน่งที่ต้องการ Set หรือ Clear Breakpoint ซึ่งเมื่อ Set แล้วจะปรากฏให้เห็นเป็นจุดสีแดงกรณีที่ใช้กับ Wires หรือเป็นกรอบสีแดงรอบ Node หรือ Diagram

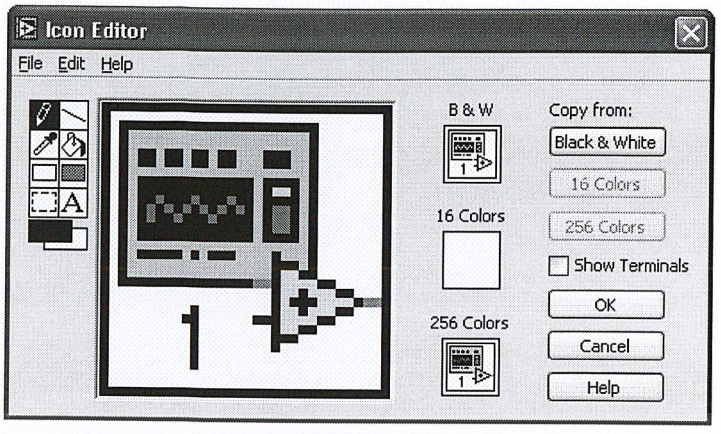
2.7 การสร้างโปรแกรมย่อย (SubVI)

ในการสร้างโปรแกรมด้วย LabVIEW เพื่อใช้งานนั้น มักจะสร้างโปรแกรมย่อย (SubVI) ขึ้นมาเพื่อสามารถที่จะนำมาเรียกใช้อีก โปรแกรมย่อยจะคล้ายกับ Function หรือ Subroutinesในโปรแกรมทั่วๆ ไป

VIที่นำมาใช้เป็นโปรแกรมย่อยจะต้องมี Icon เพื่อแทนตัวโปรแกรมย่อยใน Block Diagram ของ VI ที่เรียกใช้และยังต้องมี Connection กับ Terminal เพื่อที่จะในการรับหรือส่งค่าไปยัง VI ที่เรียกใช้

Icon -ทุก VI จะมี Icon อยู่ที่มุมขวาบนโดย LabVIEW จะสร้างมาให้อัตโนมัติ แต่ผู้ใช้งานสามารถแก้ไขได้โดย Double Click Mouse ที่ Icon หรือ Click Mouse ขวาและเลือก Edit Icon จาก Pop Up Menu เมื่อเข้าไปใน Edit Icon แล้วสามารถที่จะใช้ Tool ต่างๆ เช่น Pencil , Line, Dropper, Text เป็นต้นเพื่อทำการวาดรูปบน Icon หรืออาจจะ Copy ภาพที่มีนามสกุล .BMP, .WMF , .EMF หรือ PCT มาใส่ลงบน Icon ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

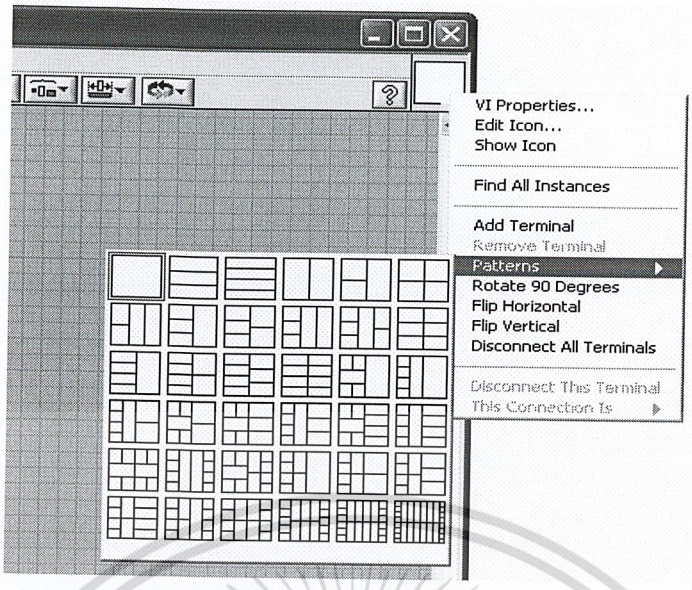


ภาพที่ 2.12 แสดง Icon Editor

Connection -เป็นตัวที่ใช้ติดต่อ รับ-ส่งข้อมูลระหว่าง Control และ Indicator ของโปรแกรมย่อยนั้นๆ ไปยัง VI ที่เรียกใช้งาน การกำหนด Connection ว่าจะเป็น Control หรือ Indicatorใดๆ ทำโดยเลือก Show Connector จาก Pop Up Menu LabVIEW จะนับจำนวน Connectionให้อัตโนมัติแล้วให้ผู้ใช้งานกำหนดว่า Connectionใด แทนตัว Control หรือ Indicatorโดยใช้ Mouse Click เลือกจับคู่กัน รูปแบบของจำนวน Terminals สามารถที่จะแก้ไขหรือเปลี่ยนแปลงได้โดยเลือกจาก Pop Up Menu ของ Connectiorn ซึ่งประกอบด้วย

1. Add Terminal เพื่อเพิ่ม Terminal เมื่อมีการเพิ่ม Control หรือ Indicator
2. Remove Terminal เพื่อลบ Terminal ออกจาก Connection
3. Pattems เพื่อเลือกรูปแบบและจำนวน Terminal
4. Rotate 90 degree, Flip Horizontal และ Flip Vertical เพื่อหมุน Connection
5. Disconnect All Terminal เพื่อใช้ลบ Control และ Indicator ออกจากทุก Terminal
6. Disconnect ThisTerminal เพื่อใช้ลบ Control หรือ Indicator ของ Terminalที่เคอร์เซอร์ชี้อยู่
7. This Connector is Required เพื่อกำหนดว่าเวลาใช้งานจะต้องมีค่าส่งมาจาก VI ที่เรียกใช้
8. This Connector Recommended , Optional เพื่อกำหนดว่าเวลาใช้งานจะมีค่าส่งมาจาก VI ที่เรียกใช้หรือไม่ก็ได้

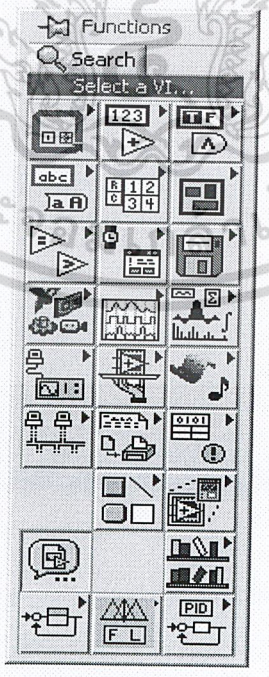
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 2.13 แสดง Terminal Pop Up Menu

โปรแกรมย่อยสร้างได้โดยการเขียนโปรแกรมให้เสร็จก่อน จากนั้นทำการแก้ไข Icon และทำการกำหนด Connection ของ Control และ Indicator บน Front Panel

2.7.1 การใช้งานโปรแกรมย่อย (SubVI)



ภาพที่ 2.14 แสดงการเรียกใช้งานโปรแกรมย่อย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

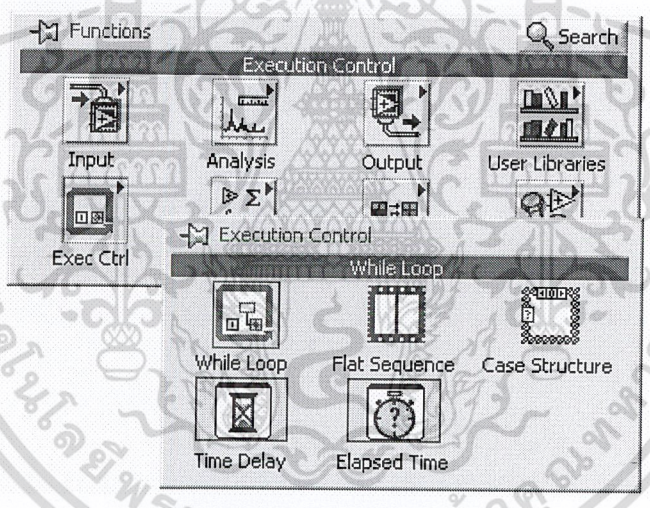
ในกรณีที่ในโปรแกรมมีการทำงานชนิดเดียวกันหลายๆครั้งสามารถสร้างส่วนของโปรแกรมนั้นเป็นโปรแกรมย่อยซึ่งจะช่วยให้ใช้เนื้อที่ในการเก็บโปรแกรมน้อยลง ลดเวลาในการเขียนโปรแกรมและง่ายในการแก้ไขข้อผิดพลาดของโปรแกรม นอกจากนี้ยังสามารถเรียกโปรแกรมย่อยจากโปรแกรมย่อยได้อีกด้วย

สำหรับ LabVIEW วิธีการเรียกโปรแกรมย่อยมาใช้งานทำได้โดยเลือก Select a VI...บน Function Palette ต่อจากนั้นให้เรียกโปรแกรมย่อยที่จะใช้งานมาใส่ลงบน VI ที่กำลังเขียนอยู่

2.8 ตัวอย่างการใช้งานฟังก์ชันที่มีใน LabVIEW

2.8.1 While loop

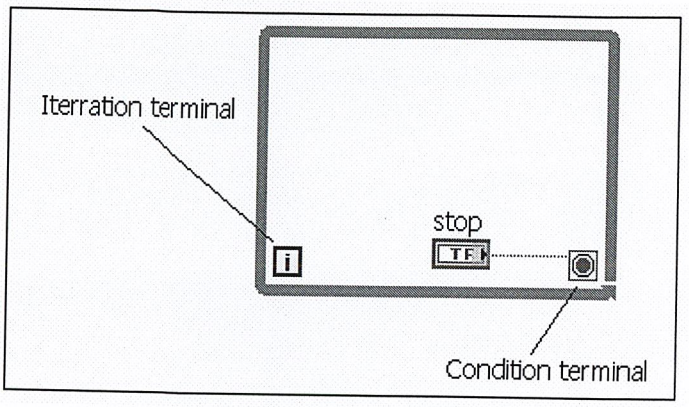
เป็น Structure อยู่บน Function Palette สร้างจาก Structure Sub palette เราสามารถสร้างได้ Block Diagram โดยเลือกจาก Structure Sub palette ของ Function Palette



ภาพที่ 2.15 แสดง Sub palette ของ Function While loop

การทำงานจะสั่งให้ทำงานภายใน Loop เมื่อได้รับค่าจริงเข้าที่ Condition Terminal และจะหยุดการทำงานเมื่อได้รับค่าเท็จ ส่วนจำนวนครั้งที่ Loop ทำงาน จะให้ค่าออกมาที่ Interation Terminal โดยจะมีค่าเริ่มต้นที่ศูนย์

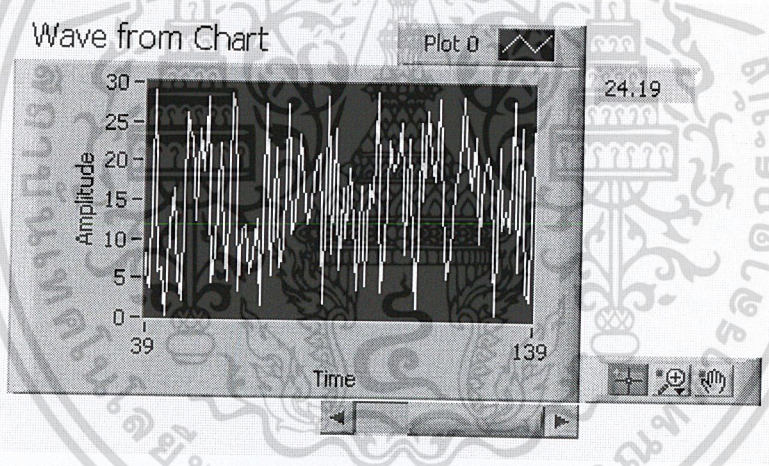
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 2.16 แสดงFunction While loop บนหน้า Diagram

2.8.2 Waveform Chart

เป็น Numeric Indicatorที่ใช้แสดงผลข้อมูลที่เป็นตัวเลขเลือกมาจาก Graph Sub paletteของ Control Palette



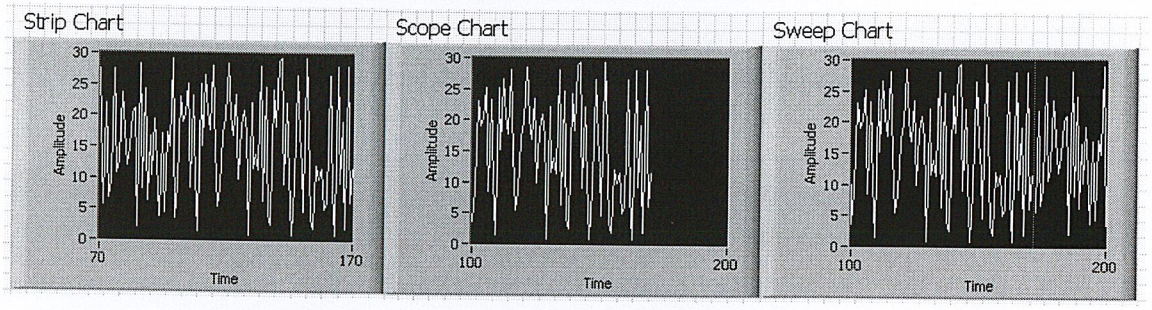
ภาพที่ 2.17 แสดงFunction2.9.2Waveform Chartบนหน้า Front Panel

การแสดงผลสามารถกำหนดได้ 3 แบบ คือ Strip Chart, Scope Chart และ Sweep Chart โดยเลือกจาก Data Operation >> Update Mode Menu

1. Strip Chart จะเลื่อนหน้าจอที่แสดงผลไปตามข้อมูลที่รับเข้ามาคล้ายกับกระดาษบันทึกข้อมูลของ Recorder
2. Scope Chart จะเขียนข้อมูลไปจนเต็ม Chart จากนั้นจะลบจอก่อนจึงจะเขียนข้อมูลใหม่
3. Sweep Chart คล้ายกับ Scope Chart แต่จะไม่ลบหน้าจอเหมือนเขียนข้อมูลใหม่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

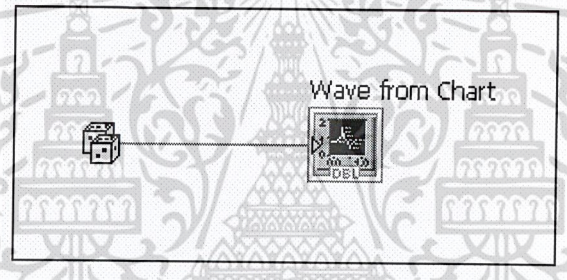
สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง



ภาพที่ 2.18 แสดงFunction Strip Chart, Scope Chart และ Sweep Chart บนหน้า Front Panel

2.8.2.1 Single plot chart

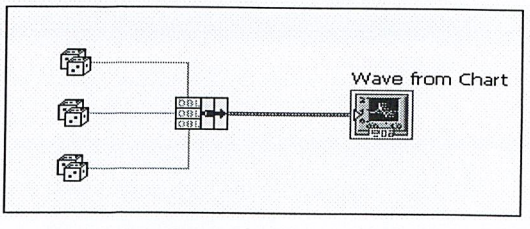
เป็นการแสดงข้อมูลค่าเดียวบน Chart สามารถลาก Output ของ Numeric Terminal ไปยัง Waveform Chart Terminal ได้เลย



ภาพที่ 2.19 แสดงFunction1Single Plot Chart

2.8.2.2 Multiple Chart Plot Waveform Chart

เป็นการแสดงข้อมูลมากกว่า 1 ค่าบน Chart โดยใช้ Fubction Bundle เพื่อรวมข้อมูลหลายค่าเข้าด้วยกัน Bundle จะอยู่ใน Cluster Palette บน Function Palette ในส่วนของ Block Diagram Window การขยาย Input ของ Bundle ทำได้โดยใช้ Positioning Tool ไป Drag ที่มุมของ Bundle แล้วขยายขนาดของ Input ได้ตามต้องการ

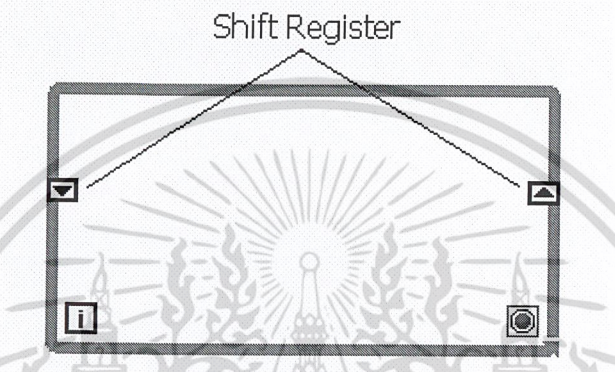


ภาพที่ 2.20 แสดงFunction1 Multiple Chart Plot Waveform Chart

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
61422

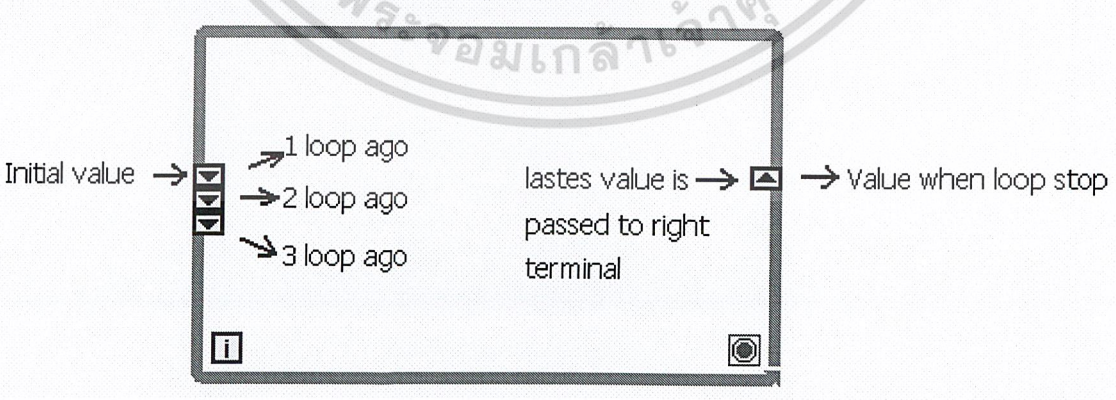
2.9 Shift Register

Shift Register ประกอบด้วย Terminal เป็นคู่ โดยจะอยู่ต้องข้ามกันบนขอบแนวตั้งของ Loop ข้อมูลจะถูกส่งจาก Shift Register Terminal ด้านซ้ายไปยังด้านขวา Shift Register สามารถใช้ได้กับข้อมูลทุกชนิด (มีเฉพาะในWhile Loop และ For Loop เท่านั้น) ใช้ส่งค่าจาก Loop ที่กำลังทำงานอยู่ไปยัง Loop ถัดไป สร้างโดยClick Mouseขวาที่ขอบของ Loop แล้วเลือก Add Shift Register จาก Pop Up Menu



ภาพที่ 2.21 แสดงShift Register

Shift Register สามารถเก็บค่าได้หลายลูป ซึ่งจะมีประโยชน์ในการเก็บข้อมูล (Averaging) ทำโดยกด Pop Up Menu ของ Shift Register ด้านซ้าย และเลือก Add Element ในรูปด้านล่าง ถ้าเราเพิ่มจำนวนของ Shift Register ให้เท่ากับ 3 Terminal เราก็สามารถนำเอาข้อมูลใน 3 Loop ที่ผ่านมา มาใช้งานได้



ภาพที่ 2.22 แสดงการทำงานของ Shift Register

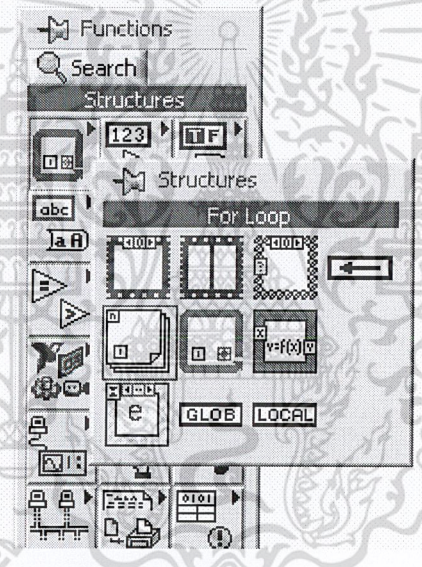
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ค่าเริ่มต้นของ Shift Register กำหนดได้โดยใส่ค่าที่ต้องการให้กับ Shift Register นั้นๆ แต่ถ้าปล่อยให้ Shift Register นั้นว่างอยู่ ค่าเริ่มต้นจะถูกกำหนดให้เป็นค่า Default เช่น ถ้าเป็น Boolean ก็จะเป็นเท็จ แต่ถ้าเป็น Numeric ก็จะเป็นค่าศูนย์

Shift Register ใช้ในการส่งค่าตัวแปรได้ทุกชนิดไม่ว่าจะเป็นตัวเลข, ตัวหนังสือ , ลอจิก ทั้งที่เป็นค่าเดียวหรือเป็นอาร์เรย์จากลูปที่เริ่มต้นไปยัง Loop ต่อไปได้เรื่อยๆ จนกว่าจะสิ้นสุดการทำงานของ Loop

2.9.1 For Loop

For Loop สร้างจาก Structure Sub palette ของ Function Palette บนBlock Diagram โดยนำไปล้อมรอบ Object ที่ต้องการให้ทำงานซ้ำๆ กัน



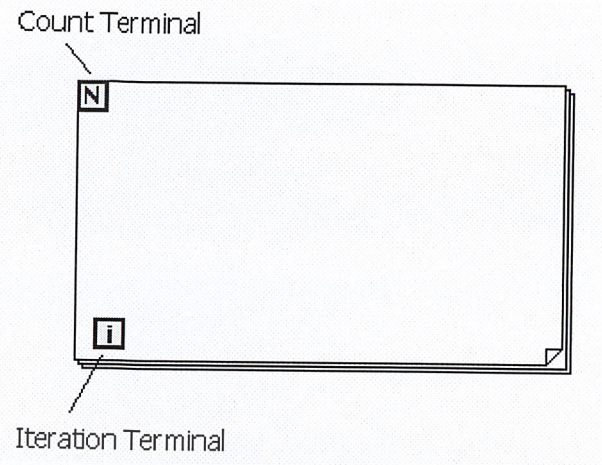
ภาพที่ 2.23 แสดง Structure Sub palette ของ Function For Loop

For Loop จะประกอบด้วย Terminal 2 ตัว คือ

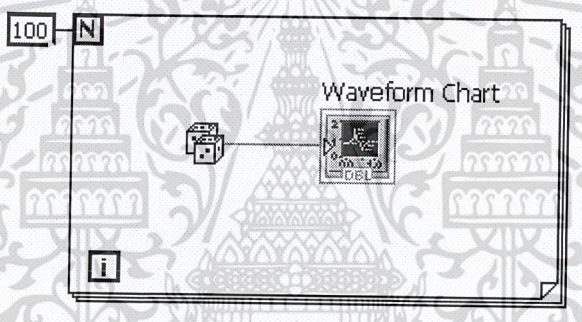
1. Count Terminal เป็น Input Terminal ที่ใช้กำหนดจำนวนครั้งในการทำงาน
2. Iteration Terminal เป็น Output Terminal ที่ใช้แสดงจำนวนครั้งที่ Loop ทำงาน

ความแตกต่างระหว่าง For Loop และ While Loop คือ For Loop จะมีการกำหนดจำนวนครั้งในการทำงานที่แน่นอน แต่ While Loop จะทำงานตามเงื่อนไขที่ได้รับ ตัวอย่างข้างล่างเป็นการใช้ For Loop ในการสุ่มค่าตัวเลขตั้งแต่ 1 ถึง 100 จำนวน 20 ครั้ง และนำมา Plot บน Waveform Chart

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 2.24 แสดง For Loop



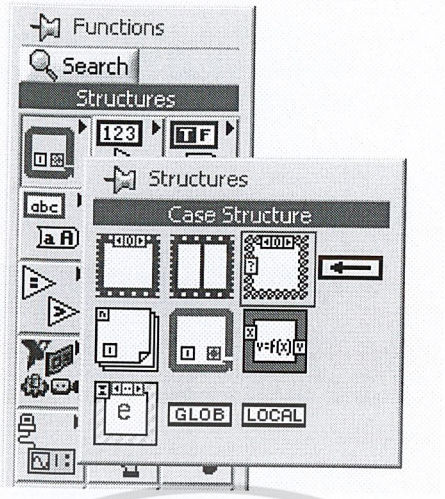
ภาพที่ 2.25 แสดง For Loopในการสุ่มค่าตัวเลขตั้งแต่ 1 ถึง 100 จำนวน 20 ครั้ง

เราสามารถเลือกใช้ For Loop และ While loop ในโปรแกรมของเราได้ ขึ้นอยู่กับเงื่อนไขในการทำงานซึ่ง For Loop จะใช้กับการทำงานที่รู้จำนวนครั้งที่แน่นอน

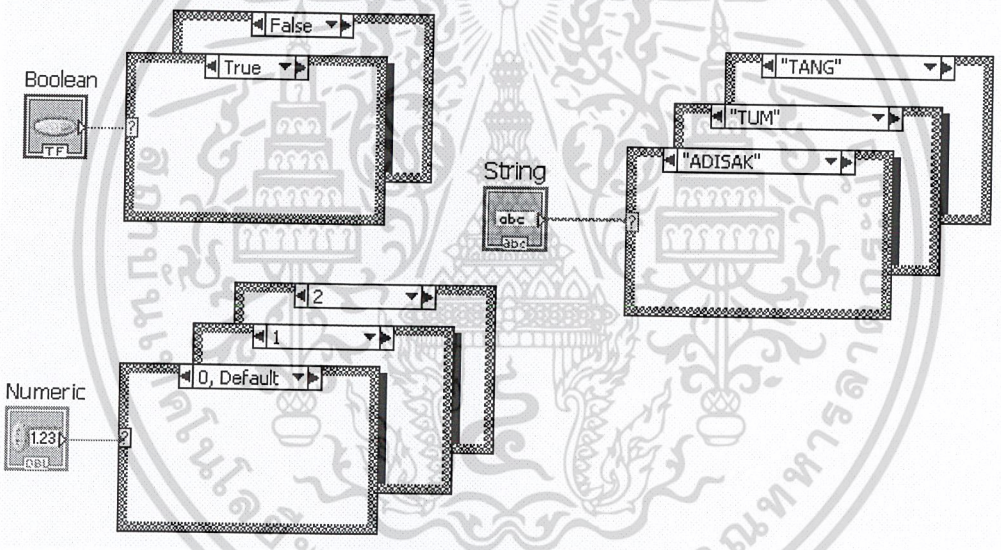
2.9.2 Case Structure

จะเหมือนกับคำสั่ง if...then...else ในโปรแกรมทั่วไป เลือกสร้างจาก Structure Sub palette โดยสามารถเลือกดูเงื่อนไขได้ที่ละ 1 เงื่อนไขเท่านั้น เมื่อเงื่อนไขทำงานจะเลือกทำงานที่เงื่อนไขใดเงื่อนไขหนึ่งเท่านั้น โดยจะตรวจสอบว่าจะทำเงื่อนไขใด จาก Selector Terminal ภายในมีรูปเครื่องหมายคำถาม ซึ่งสามารถรับค่าได้ทั้งที่เป็น Numeric , Boolean หรือ String ถ้าค่าที่ส่งให้ Selector Terminal เป็น Boolean จะมีเพียงเงื่อนไขจริงและเท็จเท่านั้น แต่ถ้าเป็น Numeric หรือ String จะสามารถสร้างเงื่อนไขได้มากถึง 2^{31} -1 เงื่อนไข

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 2.26 แสดง Structure Sub palette ของCase Structure



ภาพที่ 2.27 แสดง Case Structureซึ่งสามารถรับค่าได้ทั้งที่เป็น Numeric , Boolean หรือ String

ในกรณีที่มีการส่งค่าออกจากเงื่อนไขใดเงื่อนไขหนึ่ง จะเกิด Output Tunnel ที่ขอบของเงื่อนไขที่ส่งข้อมูลออก โดยจะอยู่ที่ตำแหน่งเดียวกันหมด เงื่อนไขใดที่มีข้อมูลส่งมาที่ Tunnel ครบทุกเงื่อนไขแล้วจะแสดงเป็นสี่เหลี่ยมสีดำ แต่ถ้า Tunnel เป็นสี่เหลี่ยมสีขาว แสดงว่ายังมีเงื่อนไขที่ยังไม่ได้ส่งค่ามาที่ Tunnel ซึ่งเราจำเป็นที่จะต้องส่งค่าให้กับ Tunnel จากเงื่อนไขทุกเงื่อนไขที่มีอยู่

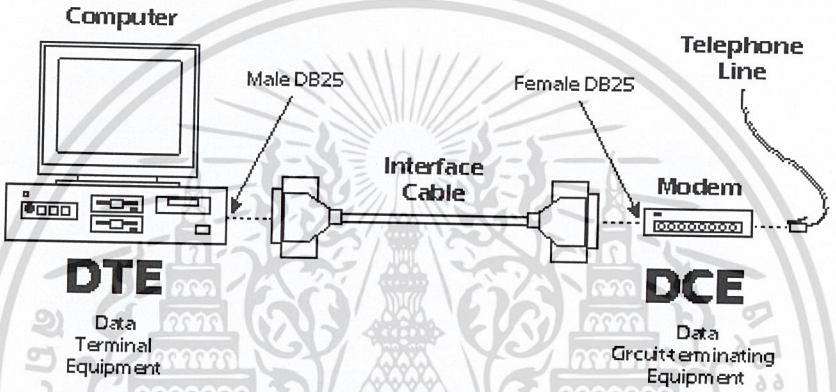
Case Structure ใน LabVIEW จะใช้ในการเลือกทำงานตามเงื่อนไขที่เท่ากับค่าที่ส่งมาที่ Selector Terminal โดยใช้ได้กับค่าที่เป็น Numeric , Boolean หรือ String ในกรณีที่มีค่าส่งออกจากเงื่อนไข จะเกิดเป็น Tunnel ขึ้น ซึ่งต้องทำการส่งค่าจากภายในเงื่อนไข ออกไปยัง Tunnel ให้ครบทุกเงื่อนไข

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3
การติดต่อสื่อสารและส่งผ่านข้อมูล

3.1 กล่าวนำ

การสื่อสารแบบ Host Link เป็นการเชื่อมต่อ PLC กับคอมพิวเตอร์ทั่วไปผ่านทางพอร์ตของคอมพิวเตอร์ (COM1 หรือ COM2) ส่วนมากจะนิยมใช้มาตรฐานการส่งแบบอนุกรมเพื่อให้สามารถควบคุม PLC จากคอมพิวเตอร์ได้



ภาพที่ 3.1 แสดงการติดต่อสื่อสารผ่านพอร์ทอนุกรม RS232

สำหรับคอมพิวเตอร์ 1 เครื่องสามารถต่อเข้ากับ PLC ได้จำนวนมากโดยใช้การเชื่อมต่อหลายๆตัวเข้าด้วยกัน เรียกว่า PC Link ในการติดต่อแบบ Host Link จะต้องผ่านอุปกรณ์ที่เรียกว่า Host ink Units ซึ่งจะต้องตั้งค่าต่างๆที่จำเป็นที่ใช้ในการติดต่อแบบ Host ink Units (C-500-LK203) ผ่านสวิทซ์ของเครื่อง PLC

3.2 การสื่อสารข้อมูลแบบอนุกรม

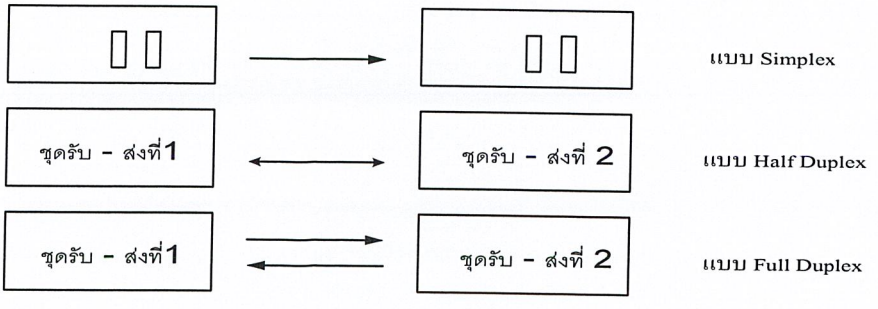
ถ้ากล่าวถึงการสื่อสารข้อมูลแบบอนุกรมแล้ว แสดงว่าจะต้องมีการสื่อสารข้อมูลแบบขนานด้วยการสื่อสารแบบขนานก็คือข้อมูลทุกๆ Bit ในแต่ละWord จะถูกส่งออกไปพร้อม ๆ กัน ขึ้นอยู่กับว่าเวิร์ดดังกล่าวมีขนาดเท่าไร โดยทั่วก็คือ 1 ไบท์ หรือ 8 Bit นั้นเอง การส่งข้อมูลแบบขนานนี้จะมีข้อจำกัดทางด้านระยะทาง โดยทั่วไปจะส่งในระยะไม่เกิน 3 – 5 ฟุต เท่านั้น ทั้งนี้ขึ้นอยู่กับอัตราความเร็วที่ใช้ในการส่งข้อมูลด้วยยิ่งอัตราการส่งสูงก็จะได้ระยะทางที่สั้นลง การส่งข้อมูลแบบขนานนั้นนิยมในระบบที่ต้องการความเร็วสูงมากๆ แต่อุปกรณ์อยู่ไม่ห่างกันมากนัก ส่วนการส่งข้อมูลแบบอนุกรมนั้นข้อมูลจะถูกทยอยส่งออกไปทีละ Bit จนครบทั้ง Word

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สายสัญญาณเพียงเส้นเดียว แต่ในการใช้งานจริงจะต้องมีสายสัญญาณอีกเส้นเป็นสายสัญญาณ Ground ดังนั้นเมื่อเราส่งข้อมูลในแบบอนุกรมเราจะสามารถใช้สายสัญญาณอย่างน้อยที่สุดเพียง 2 เส้น ในขณะที่ส่งข้อมูลแบบขนานจะต้องใช้อย่างน้อยเท่ากับจำนวนBit บวกกับสายสัญญาณระดับแรงดัน Ground อีก 1 เส้น ที่สำคัญการส่งข้อมูลแบบอนุกรมนั้นจะสามารถใช้สายสัญญาณอย่างน้อยที่สุดเพียง 2 เส้น ในขณะที่การส่งข้อมูลแบบขนานจะต้องใช้อย่างน้อยเท่ากับจำนวน Bit บวกกับสายสัญญาณระดับ Ground อีก 1 เส้น ที่สำคัญการส่งข้อมูลแบบอนุกรมนั้นจะสามารถส่งข้อมูลได้ไกลกว่ามาก เช่น ถ้าส่งตามมาตรฐานของ RS–232 ที่จะกล่าวต่อไปในภายหลังจะสามารถส่งได้ไกลถึง 30 ถึง 40 ฟุต โดยไม่ต้องใช้อุปกรณ์ขับสัญญาณเพิ่มเติมแต่อย่างใด อย่างไรก็ตามในการส่งข้อมูลยังมีข้อกำหนดบางประการเพื่อให้ได้รับข้อมูลที่ต้อง แม่นยำและมีความน่าเชื่อถือสูง จะต้องมีวิธีตรวจสอบความถูกต้องของข้อมูลได้ในระหว่างที่มีการสื่อสารกันอยู่ด้วย ซึ่งจะได้กล่าวต่อไป

3.2.1 การส่งข้อมูลแบบ Simplex และ Duplex

ในการสื่อสารไม่ว่าจะเป็นการสื่อสารข้อมูลหรือการสื่อสารทั่วไปนั้นย่อมจะต้องประกอบด้วยผู้รับและผู้ส่ง ผู้รับในขณะนี้อาจสามารถเป็นผู้ส่งในอนาคตได้ แต่มีบางกรณีที่เป็นผู้รับและผู้ส่งแน่นอนตายตัวอยู่ตลอดเวลา เช่น การสื่อสารข้อมูลระหว่างเครื่องคอมพิวเตอร์กับเครื่องพิมพ์เป็นต้น การสื่อสารของอุปกรณ์ที่มีผู้รับและผู้ส่งตายตัวนั้น เราเรียกว่าการสื่อสารแบบ Simplex กล่าวคือ การสื่อสารเป็นไปในลักษณะทิศทางเดียวตลอดเวลา ซึ่งจะมีที่ใช้ไม่มากนัก การสื่อสารโดยทั่วไปนั้นจะเป็นแบบ Duplex คือมีทิศทางการสื่อสาร 2 ทิศทางทั้งไปและกลับ การสื่อสารในลักษณะ Duplex นี้ยังแบ่งออกได้เป็น 2 ชนิด คือ แบบ Half Duplex นิยมเขียนย่อกันว่า HDX ซึ่งจะมีทิศทางการสื่อสารในลักษณะที่ผลัดกันเป็นผู้ส่งและผู้รับพร้อมกันไป และแบบ Full Duplex นิยมเขียนย่อว่า FDX จะมีทิศทางการสื่อสารในลักษณะสัญญาณรับทิศทางหนึ่งสัญญาณส่งอีกทิศทางหนึ่งหรือกล่าวได้อีกนัยหนึ่งว่าสัญญาณรับและส่งจะมีสายตัวนำสัญญาณแยกออกจากกันโดยเด็ดขาด



ภาพที่ 3.2 แสดงการสื่อสารข้อมูลอนุกรมแบบต่างๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

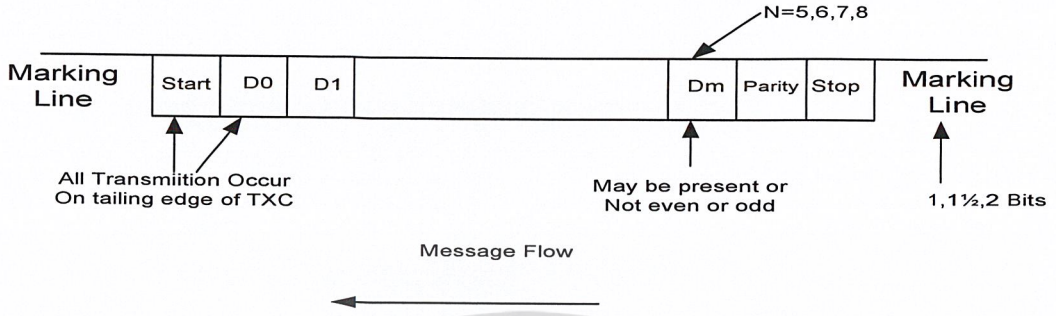
3.2.2 โปรโตคอลของการสื่อสารแบบอนุกรม

เมื่อพิจารณาการส่งข้อมูลในแบบอนุกรมให้ดีจะพบว่าปัญหาหนึ่งที่ควรจะเกิดขึ้นอยู่เสมอ ก็คือการตัดสินว่าข้อมูลที่ได้รับนั้นมีจุดเริ่มต้นที่ใด ดังนั้นมีการกำหนดข้อตกลงในการสื่อสารขึ้นเพื่อแก้ปัญหานี้ ข้อตกลงดังกล่าวเราเรียกว่า โปรโตคอล (Protocol) ของการสื่อสารข้อมูลแบบอนุกรม สามารถแบ่งออกเป็น 2 ประเภทใหญ่ๆ คือโปรโตคอลสำหรับการสื่อสารข้อมูลแบบ Synchronous และโปรโตคอลสำหรับการสื่อสารข้อมูลแบบ Asynchronous การสื่อสารแบบซิงโครนัสนั้นข้อมูลจะถูกส่งออกไปอย่างสม่ำเสมอ ช่วงเวลาระหว่าง Bit และระหว่าง Word จะมีค่าเท่ากันเสมอ ดังนั้นในการสื่อสารข้อมูลอนุกรมในแบบ Synchronous จึงต้องมีสัญญาณเพิ่มเติมเพื่อกำกับการส่งว่าควรจะส่งเมื่อใดและควรจะหยุดเมื่อใด ระบบที่เป็นซิงโครนัสจะเป็นระบบที่มีความเร็วสูงแต่ก็ยังต่ำกว่าการสื่อสารแบบขนาน การสื่อสารแบบ Asynchronous นี้เป็นหัวใจของการสื่อสารข้อมูลผ่านทางสายโทรศัพท์ในปัจจุบันการสื่อสารแบบนี้ช่วงระยะเวลาระหว่าง Bit จะมีค่าเท่ากันเช่นเดียวกับ Synchronous แต่จะมีระยะห่างระหว่าง Word นั้นแตกต่างกันออกไปเป็นกี่วินาที นาที ชั่วโมง หรือวัน เป็นต้นได้ทั้งสิ้น ขึ้นอยู่กับทางฝ่ายรับสามารถรอได้หรือไม่เท่านั้น เมื่อไม่มีข้อกำหนดทางด้านระยะเวลาระหว่าง Word แล้ว ทางผู้ส่งและผู้รับจะเข้าใจตรงกันได้อย่างไรว่าที่ใดคือจุดเริ่มต้นและจุดสิ้นสุดของแต่ละ Word เพื่อแก้ปัญหานี้จึงมีการกำหนดข้อตกลงเกี่ยวกับรูปแบบของข้อมูลที่จะส่งให้ทางผู้รับสามารถเข้าใจว่าจุดใดเป็นจุดเริ่มต้นของ Word ข้อกำหนดดังกล่าวกำหนดให้แต่ละ Word จะต้องขึ้นต้นด้วย Bit เรียกว่า Start Bit ซึ่งจะต้องมีข้อมูลเป็นลอจิก 0 เสมอ จากนั้นตามด้วย Bit ข้อมูลที่ต้องการส่งซึ่งมีความยาว 5 ถึง 8 Bit ถัดจาก Bit ข้อมูลก็จะเป็น Parity Bit ซึ่งทำหน้าที่เป็น Bit สำหรับตรวจสอบความถูกต้องของข้อมูลที่ได้รับว่ามีความถูกต้องหรือไม่ Parity Bit นี้มี 2 ประเภท คือEven Parity ซึ่งจะกำหนดจำนวน Bit ที่เป็นลอจิก 1 ใน Bit ที่เป็นข้อมูลมีจำนวนเป็นจำนวนคี่ ในการส่งข้อมูลบางครั้งอาจจะไม่มีการใช้ Parity Bit ก็ได้ถ้าหากการสื่อสารในครั้งนั้นมีความน่าเชื่อถือสูงคือ มีสัญญาณรบกวนต่ำเป็นการเพิ่มคามเร็วในการสื่อสารได้ด้วย Bit สุดท้ายในรูปแบบก็คือ Stop Bitทำหน้าที่บอกทางผู้รับว่าขณะนี้ข้อมูลที่ทางผู้รับได้รับนั้นครบ Word แล้วขอให้เตรียมชุดรับ Word ต่อไปได้ Bit สุดท้ายนี้ถูกกำหนดให้เป็นลอจิก 1 เสมอทั้งนี้เพื่อให้ระบบสามารถตรวจสอบ Bitเริ่มต้นได้ Bit สุดท้ายนี้อาจมี 1 Bit หรือ 2 Bit ก็ได้

จากรูปแบบดังกล่าว จะเห็นว่าเรามีรูปแบบสำหรับการสื่อสารมากมายไปหมด เช่น 5E1(5Databit,Even Parity,1 Stop Bit),7E1(7 Data bit ,Even Parity,1 Stop Bit) และ 8N1(8 Databit,No Parity,1 Stop Bit) เป็นต้น ในการใช้งานทั่วไปเรานิยมใช้งานทั่วไปเรานิยมใช้กันอยู่เพียง 2 รูปแบบคือ 7E1 และ 8N1 จะเลือกใช้รูปแบบใดขึ้นอยู่กับสภาพของสายส่งสัญญาณว่ามีสัญญาณรบกวนมากเพียงใด ถ้าหากสายส่งมีสัญญาณรบกวนมากก็ควรจะใช้ 7E1 แต่ถ้าสายส่งสัญญาณมีสภาพดีสัญญาณรบกวนต่ำการใช้ 8N1 จะเร็วกว่า เป็นต้น ทั้งนี้จะต้องมีการตกลงกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

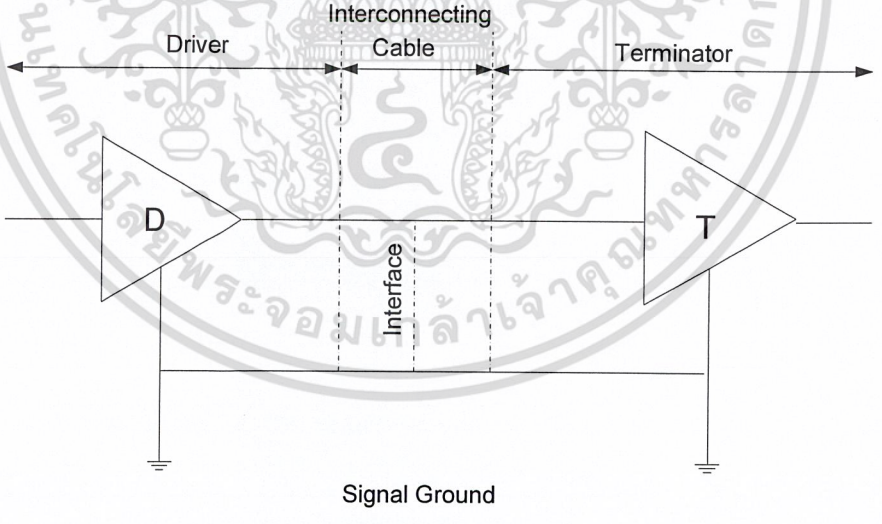
ล่วงหน้าระหว่างผู้รับและผู้ส่งว่าจะใช้รูปแบบใดในการสื่อสาร ลักษณะของข้อมูลที่ถูกส่งออกไปจะมีลักษณะดังรูป



ภาพที่ 3.3 แสดงแบบมาตรฐานการสื่อสารข้อมูลแบบ Asynchronous

3.2.3 มาตรฐานสัญญาณอนุกรมแบบ RS232C

มาตรฐานการสื่อสารข้อมูลแบบอนุกรมที่กำหนดโดย EIA (Electronics Industrial Association) มาตรฐาน RS232C ได้ถูกตีพิมพ์ในปี คศ.1969 ตัวอักษรRS แทน "Recommend Standard" 232 แทนหมายเลขของมาตรฐานส่วนอักษร C แสดงให้รู้ว่ามาตรฐานได้รับการแก้ไขกี่ครั้ง



ภาพที่ 3.4 แสดงลักษณะสมบัติทางไฟฟ้าของการอินเทอร์เฟสแบบ RS232

ลักษณะสมบัติทางไฟฟ้าของการอินเทอร์เฟสแบบ RS232

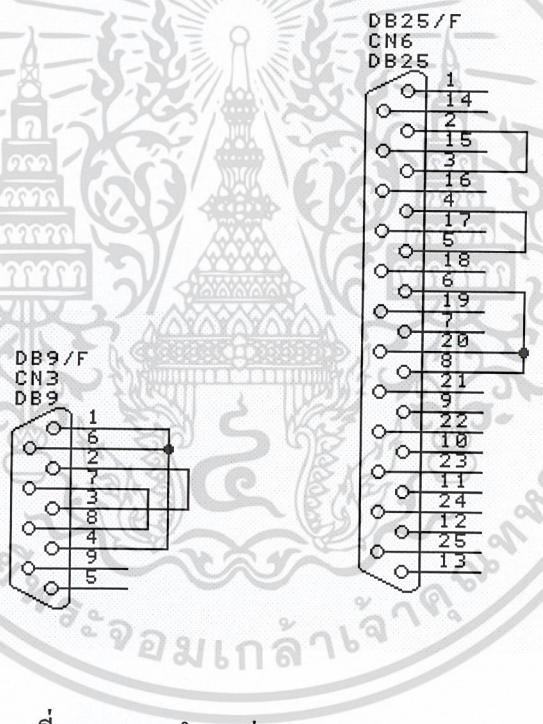
- ถูกออกแบบให้ใช้กับอุปกรณ์พวกสัญญาณ Discrete
- ใช้การอินเทอร์เฟสแบบ Unbalanced

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ในแต่ละวงจรใช้ลวดนำในการนำสัญญาณ 1 เส้น และมีสาย Ground รวมของทุกวงจรอีกหนึ่งเส้น
- อัตราเร็วในการส่งข้อมูลมีค่า < 20 กิโลบิทต่อวินาที (Kbps)
- ระยะทางสูงสุดที่ใช้ในการส่งข้อมูลมีค่า < 15 เมตร
- ทำให้เกิด Crosstalk ที่มีค่ามาก

3.2.4 พอร์ทสื่อสารอนุกรม

พอร์ทสื่อสารอนุกรม (Serial Communication Device) หรือ เรียกว่า "Serial Port" เครื่องคอมพิวเตอร์โดยปกติจะมีพอร์ท ชนิดนี้อยู่แล้ว 2 พอร์ท คือพอร์ทขนาด 9 ขา (9-pins) มีรูปร่างเหมือนสี่เหลี่ยมคางหมู มีเข็มยื่นออกมา 9 เข็ม เรียกหัวชนิดนี้ว่า "DB-9 Connector Male Type"



ภาพที่ 3.5 แสดงตำแหน่งขาของ DB-9 และ DB-25

มาตรฐานของการรับส่งข้อมูลแบบอนุกรม (RS-232) นี้ได้กำหนดขึ้นเพื่อให้คอมพิวเตอร์ต่างยี่ห้อกัน หรืออุปกรณ์ต่อพ่วงแต่ละชนิดรับส่งข้อมูลกันได้เมื่อทำตามมาตรฐานไม่สนใจว่าอุปกรณ์หรือคอมพิวเตอร์นั้นจะผลิตมาจากที่ใด

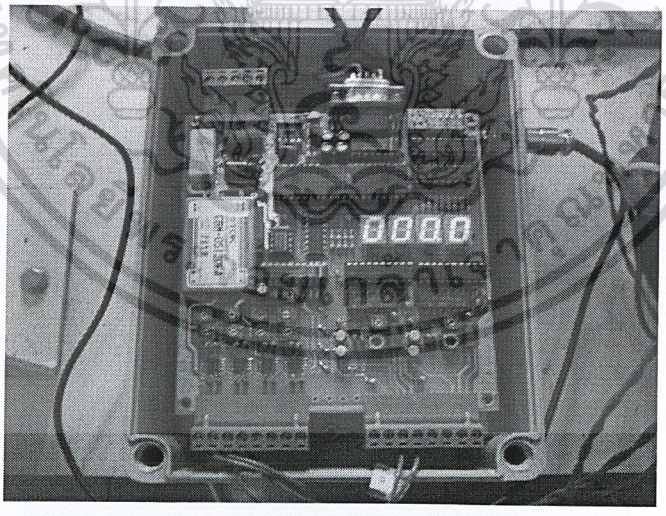
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3.1 แสดงขาสัญญาณต่าง ๆ ในการรับส่งข้อมูลแบบอนุกรม RS-232 Standard

DB-9 pins	DB-25 pins	Function
1	8	Carrier Detect
2	3	Received Data
3	2	Transmitted Data
4	20	Data Terminal Ready
5	7	Signal Ground
6	6	Data Set Ready
7	4	Request to Send
8	5	Clear to Send
9	22	Ring Indictor

3.3 ชุด REMOTE TERMINAL ANALOG DIGITAL: RTAD

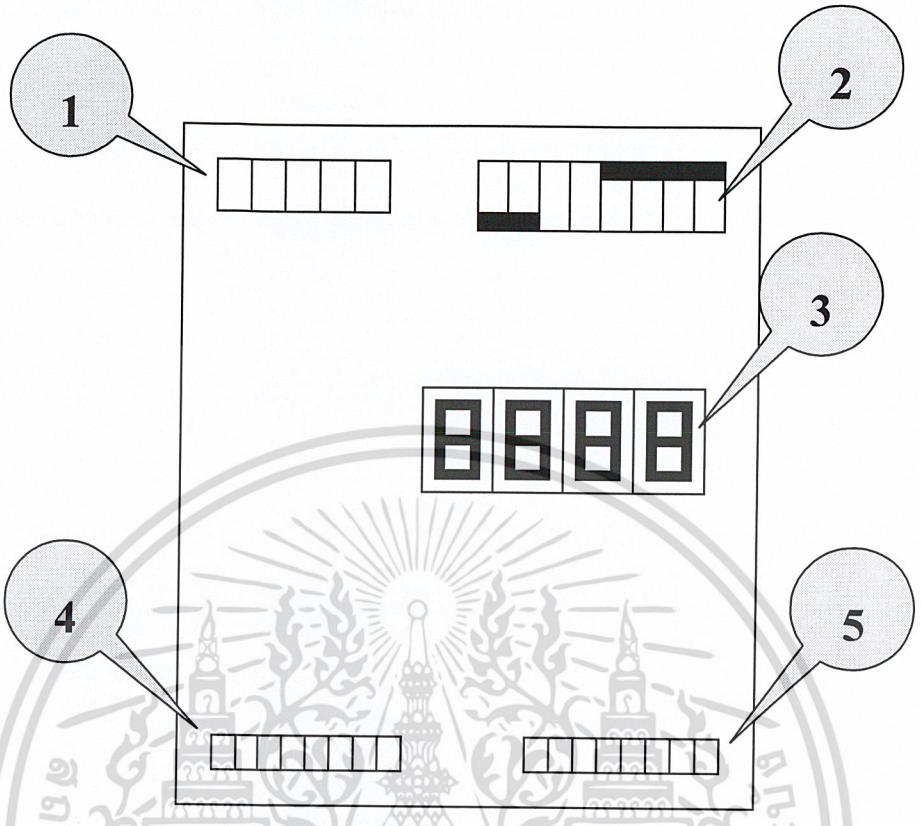
ชุด REMOTE TERMINAL ANALOG DIGITAL จะเป็นตัวแปลงสัญญาณอนาลอกเป็นดิจิตอล และดิจิตอลเป็นอนาลอกซึ่งมีส่วนประกอบต่าง ๆ ดังนี้



ภาพที่ 3.6 แสดงชุด RTAD ใช้งานจริง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3.1 ส่วนประกอบที่สำคัญชุด REMOTE TERMINAL ANALOG DIGITAL : RTAD



ภาพที่ 3.7 แสดงส่วนประกอบของชุด RTAD

จากภาพ 3.7 อธิบายรายละเอียดส่วนประกอบ RTAD ตามลำดับหมายเลข

หมายเลข 1 คือช่องสำหรับต่อแหล่งจ่ายไฟ (24 โวลต์) และช่องต่อสำหรับสาย RS-485 เมื่อใช้สาย RS-485 ในการสื่อสาร

หมายเลข 2 คือ SELECTOR SWITCH สำหรับปรับอัตราเร็วในการส่งผ่านข้อมูล และปรับตั้งค่า UNIT NUMBER

หมายเลข 3 คือ SEVEN SEGMENT แสดงค่าที่อยู่ใน CH0 ถึง CH3 ของอินพุท โดยตัวแรกจะแสดง CHANNEL และอีกสามตัวที่เหลือจะแสดงค่าที่อยู่ในแต่ละ CHANNEL

หมายเลข 4 คือ INPUT สำหรับรับสัญญาณแรงดัน 0 ถึง 10 โวลต์ (DC) มีทั้งหมด 4 Channel โดยสัญญาณอนาลอก 0 –10 โวลต์ จะถูกแปลงให้ไปเป็น สัญญาณดิจิตอล 000-999 ขนาด 10 บิต

หมายเลข 5 คือ OUTPUT สำหรับสร้างสัญญาณแรงดัน 0 ถึง 10 โวลต์ตามค่าที่สั่ง White มาที่ตัวบอร์ด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3.2 โปรโตคอลในการติดต่อสื่อสาร

โปรโตคอลที่ใช้ในการติดต่อสื่อสารกับตัว REMOTE TERMINAL ANALOG DIGITAL นั้นจะมีอยู่สองตัวคือ คำสั่งสำหรับอ่านข้อมูล และคำสั่งสำหรับเขียนข้อมูล

1. คำสั่งที่ใช้อ่านข้อมูล

Command **@00RA*XX↵**

Response **@00RA*aaaabbbbccccdddd**

aaaa , bbbb , cccc และ dddd คือค่าข้อมูล 000 –999 ที่อยู่ใน channel 0,1,2 และ 3 ตามลำดับ

2. คำสั่งที่ใช้เขียนข้อมูล

Command **@00WAaabb*XX↵**

aa และ bb คือค่าข้อมูล 00 –FF ที่ต้องการเขียนลงไปใน CH0 และCH1 ของ OUTPUT ตามลำดับ ส่วน Response นั้นถ้ารูปแบบคำสั่งถูกต้องจะไม่มี Response แต่ถ้ารูปแบบคำสั่งไม่ถูกต้องจะมี Response แจ้ง Error กลับมา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4
ความรู้เบื้องต้นเกี่ยวกับการควบคุมแบบป้อนกลับด้วยตัวควบคุม PID

4.1 เครื่องควบคุม

เครื่องควบคุมหรือตัวควบคุมเป็นส่วนประกอบที่สำคัญของระบบควบคุมแบบอัตโนมัติ โดยทำหน้าที่ในการสั่งการด้วยการสร้างสัญญาณสำหรับควบคุมกระบวนการต่าง ๆ ให้มีผลตอบสนองเป็นไปตามต้องการ ซึ่งสัญญาณดังกล่าวจะเป็นไปตามกฎและรูปแบบของการควบคุมที่ผู้ควบคุมได้เลือกและกำหนดไว้ล่วงหน้า โดยสัญญาณควบคุมจะขึ้นอยู่กับค่าความคลาดเคลื่อนของการควบคุม (คือค่าแตกต่างระหว่างค่าเป้าหมายและตัวแปรกระบวนการนั่นเอง)

ปัจจุบันเครื่องควบคุมสามารถแบ่งตามโครงสร้างการทำงานได้ 3 ชนิด คือ เครื่องควบคุมแบบนิวเมติกส์ที่ทำงานโดยใช้สัญญาณลม เครื่องควบคุมแบบอิเล็กทรอนิกส์ทำงานโดยใช้วงจร Linear Circuit ในการสร้างสัญญาณควบคุม และเครื่องควบคุมแบบดิจิตอลทำงานโดยใช้วงจร Logic Circuit หรือไมโครโปรเซสเซอร์สร้างสัญญาณควบคุม โดยเครื่องควบคุมที่ถูกนำมาใช้และเป็นที่รู้จักกันดีมากที่สุดในวงการอุตสาหกรรมการผลิตก็คือ ตัวควบคุมแบบ PID

4.2 หลักการของตัวควบคุม

โดยส่วนใหญ่การควบคุมในอุตสาหกรรมการผลิตจะแบ่งเป็นการควบคุมแบบป้อนกลับกับแบบ Sequence แต่ในที่นี่จะขอกล่าวเฉพาะที่เป็นแบบป้อนกลับเท่านั้นและระบบการควบคุมที่เป็นแบบป้อนกลับก็จะเป็นตัวควบคุมแบบ PID ซึ่งตัวควบคุมแบบนี้มีฟังก์ชันถ่ายโอน คือ

$$G_c(s) = K_c\left(1 + \frac{1}{T_i s} + T_d s\right) \tag{4.1}$$

โดยที่ K_c = อัตราขยายของตัวควบคุม

T_i = ค่า Integral or Reset Time (วินาที)

T_d = ค่า Derivative or rate Time (วินาที)

ค่าพารามิเตอร์ของตัวควบคุม PID ทั้ง 3 ค่านี้ จะใช้สำหรับปรับให้กับตัวควบคุมเพื่อควบคุมกระบวนการที่ต้องการให้มีผลตอบสนองเป็นไปตามต้องการ ดังนั้นจะเห็นได้ว่าสัญญาณควบคุมหรือตัวแปรปรับกระบวนการ (mv) ที่ได้จากตัวควบคุม PID จะถูกกำหนดด้วย

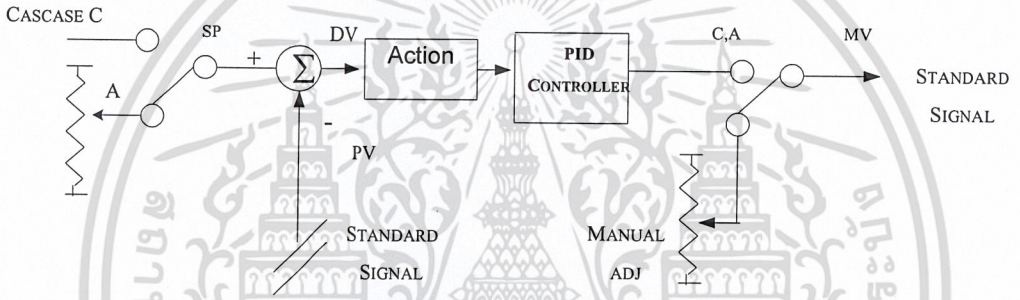
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ความสัมพันธ์ระหว่างสัญญาณขาเข้าตัวควบคุมหรือตัวแปรกระบวนการ (pv) กับสัญญาณอ้างอิงหรือค่าเป้าหมาย (sp) โดยที่ความสัมพันธ์ดังกล่าวจะขึ้นอยู่กับกฎเกณฑ์ของการควบคุมที่พนักงานหรือผู้ควบคุมได้ปรับแต่งไว้ล่วงหน้า (คือการปรับแต่งค่าพารามิเตอร์ K_c , T_i และ T_d ของตัวควบคุม PID นั่นเอง) ดังนั้นสัญญาณควบคุมของตัวควบคุม PID คือ

$$mv = \frac{100}{PB}\left[(sp - pv) + \frac{1}{Ti}\int_0^t (sp - pv)dt + T_d * \frac{d(sp - pv)}{dt}\right] + b \qquad (4.2)$$

หรือ

$$mv = Kc\left[e(t) + \frac{1}{Ti}\int_0^t e(t)dt + T_d * \frac{de(t)}{dt}\right] + b \qquad (4.3)$$



ภาพที่ 4.1 แสดงโครงสร้างพื้นฐานของตัวควบคุม PID

โครงสร้างพื้นฐานของตัวควบคุม PID ที่ใช้ในอุตสาหกรรมแสดงดังรูปที่ 3.2 โดยที่ pv : Process Variable Value ค่าสัญญาณจากเครื่องมือวัด sp : Setpoint

DV : Deviation หรือค่าความคลาดเคลื่อน (Error) mv : Manipulated Value

C : Cascade หรือ Remote Set Mode A : Automated Value

M : Manual Mode

4.3 กริยาต่างๆ ของการควบคุม

4.3.1 การควบคุมแบบ Proportional

การทำงานของเครื่องควบคุมที่จะเปิดหรือปิด (สัญญาณควบคุม) วาล์วควบคุมนั้นจะเป็นปฏิภาคกับค่าเบี่ยงเบน โดยการควบคุมแบบนี้จะเหมาะนี้จะเหมาะกับกระบวนการที่มี Time Lag ไม่สูงมากแต่การควบคุมแบบนี้มีจุดอ่อนก็คือการเกิด Offset

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3.2 การควบคุมแบบ Proportional + Integral

การเพิ่ม Integral เข้าไปก็เพื่อลดการเกิด Offset นั่นเองการควบคุมแบบนี้นิยมใช้ในการควบคุมความดันระดับของไหล แต่ถ้าเราปรับค่า Integral มากไปก็จะทำให้การควบคุมเกิดอาการแกว่ง (Hunting) ขึ้นได้

4.3.3 การควบคุมแบบ Proportional + Integral + Derivative

การควบคุมแบบนี้จะเหมาะกับกระบวนการที่มี Time Delay มาก ๆ เช่นการควบคุมอุณหภูมิเป็นต้นแต่ถ้าเราปรับค่า Derivative และ Integral ไม่เหมาะสมก็จะเกิดการแกว่งเช่นเดียวกัน

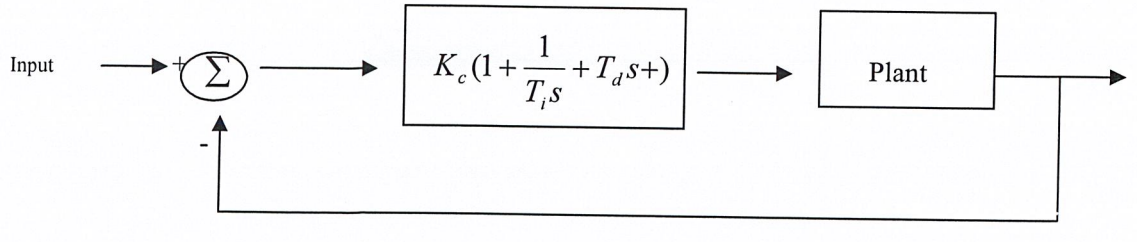
จากที่กล่าวมาในหัวข้อนี้การปรับค่าพารามิเตอร์เหล่านี้จึงต้องการวิศวกรที่มีความเข้าใจทั้งภาคทฤษฎีและภาคปฏิบัติสูงเป็นผู้ปรับค่าต่าง ๆเหล่านี้จึงจะทำให้การควบคุมเป็นไปตามความต้องการ

4.4 การปรับแต่งหาค่าพารามิเตอร์ของเครื่องควบคุม

การปรับค่าพารามิเตอร์ทั้ง 3 ค่า ของตัวควบคุม PID เป็นสิ่งจำเป็นเนื่องจากผลตอบสนองของกระบวนการที่ถูกควบคุมจะเป็นเช่นไรนั้นจะขึ้นอยู่กับการปรับแต่งตัวควบคุม ซึ่งการปรับแต่งหาค่าพารามิเตอร์ของตัวควบคุม PID สามารถหาได้ทั้งทางคณิตศาสตร์และทางปฏิบัติแต่ส่วนใหญ่ในโรงงานอุตสาหกรรมจะนิยมใช้วิธีการทางปฏิบัติมากกว่า เนื่องจากการหาค่าพารามิเตอร์ทางคณิตศาสตร์จำเป็นจะต้องรู้ตัวแปรทั้งหมดในกระบวนการซึ่งเป็นเรื่องที่ยุ่งยากมาก

การปรับแต่งหาค่าพารามิเตอร์ของตัวควบคุม PID ที่นิยมใช้ในทางปฏิบัติมีอยู่ 3 วิธีด้วยกัน คือ

- วิธี Process Reaction Curve Method ของ Ziegler – Nichols (Open Loop Method)
- วิธี Ultimate Method ของของ Ziegler – Nichols (Closed Loop)
- วิธี Trial and Error (วิธีการลองผิดลองถูก)



ภาพที่ 4.2 แสดงโครงสร้างของระบบที่ควบคุมด้วยตัวควบคุม PID

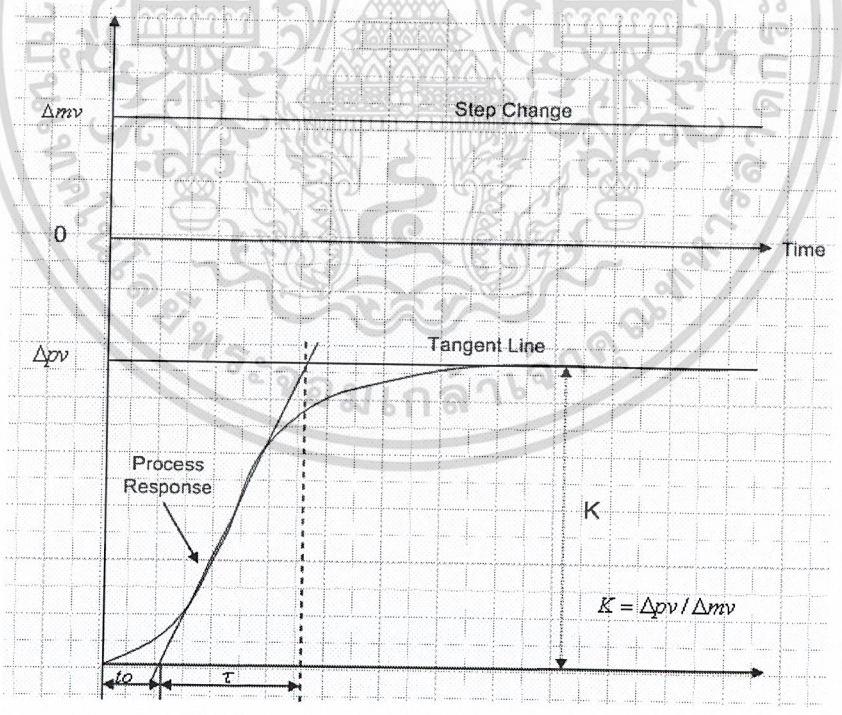
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.4.1 วิธีการแบบ Process Reaction Curve (Open Loop Method)

วิธีนี้จะหาค่าพารามิเตอร์ของตัวควบคุม PID จากผลตอบสนองเวลาของระบบหรือกระบวนการที่ถูกควบคุมแบบลูปปิดต่ออินพุทแบบ Step โดยที่ระบบหรือกระบวนการในกรณีนี้จะไม่มีPole ที่จุด Origin หรือไม่มี Dominant Complex Conjugate Poles และ ไม่มีตัวควบคุมต่อร่วมอยู่ ดังนั้นผลตอบสนองของเวลาจะเป็นรูปตัว S (ถ้าผลตองสนองเวลาไม่เป็นรูปตัว S วิธีนี้จะใช้ไม่ได้) ผลตอบสนองของเวลารูปตัว S นี้จะถูกนำมาอธิบายคุณลักษณะของกระบวนการ (Process Characteristic) ด้วยพารามิเตอร์ 3 ค่า คือ ค่าอัตราขยายของกระบวนการ K ค่าเวลาหน่วงของกระบวนการ t_o (Dead Time) และจะถูกประมาณโดยการลากเส้นสัมผัสกับจุดที่มีอัตราการเบี่ยงเบนสูงสุด (Maximum Slope) ดังรูปที่ 3.5 ส่วนค่า Kทราบได้เมื่อผลตอบสนองของกระบวนการเข้าสู่สภาวะคงที่

ดังนั้นจะเห็นได้ว่า ฟังก์ชันถ่ายโอนของกระบวนการจะประมาณได้ด้วยระบบอันดับหนึ่งแบบมีการหน่วงเวลา (First Order Lag Plus Dead Time : FOPDT) คือ

$$G(s) = \frac{C(s)}{U(s)} = \frac{Ke^{-t_0 s}}{\tau s + 1} \tag{4.4}$$



ภาพที่ 4.3 แสดงผลตอบสนองรูปตัว S เมื่อใช้วิธี Process Reaction Curve

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- วิธีการทดลองการทดลอง

ปรับตัวควบคุมอยู่ในตำแหน่ง Manual Mode แล้วทำการปรับค่า mv ไว้ค่า ๆ หนึ่งให้คงที่ตลอดในช่วงการทดลองหาค่า (เช่น 10 %) แล้วสังเกตค่า pv จากเครื่องบันทึกค่าว่าได้ผลตามรูปที่ 3.5 หรือไม่ ถ้าได้ก็ทำการลากเส้นสัมผัส Tangent Line แล้วทำการหาค่าอัตราขยายของกระบวนการ K ค่าเวลาคงที่ของกระบวนการ τ (Time Constant) และค่า t_0 (Dead Time)จากนั้นก็เลือกวิธีการว่าต้องการควบคุมรูปแบบไหน เช่น PI หรือ PID โดยสามารถคำนวณได้ใน

รูปแบบของ Ziegler – Nichols ได้กำหนดค่าของ K_c , T_i และ T_d สำหรับตัวควบคุมแบบต่าง ๆ ที่ใช้การปรับค่าพารามิเตอร์ของตัวควบคุม PID ตามวิธี Process Reaction Curve ได้ตามตารางที่ 4.1

ตารางที่ 4.1 แสดงสูตรสำหรับหาค่าพารามิเตอร์ของตัวควบคุม PID ตามวิธี Process Reaction Curve

Controller Type	Proportional Gain Kc	Integral Time Ti	Derivative Time Td
Proportional only (P)	$\frac{1}{K}\left(\frac{\tau}{to}\right)$	-	-
Proportional- Integral (PI)	$\frac{0.9}{K}\left(\frac{\tau}{to}\right)$	$3.33t_d$	-
Proportional- Integral- Derivative (PID)	$\frac{1.2}{K}\left(\frac{\tau}{to}\right)$	$2.0t_d$	$0.5t_d$

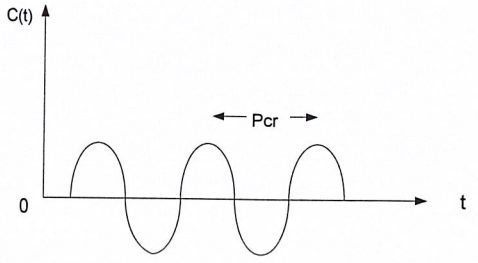
4.4.2 วิธี Ultimate Method ของ Ziegler-Nichols (Closed Loop)

วิธีนี้จะหาค่าของพารามิเตอร์ของตัวควบคุม PID จากผลตอบสนองเวลาของระบบหรือกระบวนการที่ถูกควบคุมด้วยตัวควบคุมแบบ P แบบลูปปิดต่ออินพุทแบบ Unit Step (ตัวควบคุมทำงานในแบบ Automatic Mode โดยปรับพารามิเตอร์ของ Integral และ Derivative ไม่ให้ทำงาน และทำการปรับค่าของ K_c (Proportional Gain) ไปเรื่อยๆจนกระทั่งผลของการตอบสนองของค่า pv เกิดการแกว่งอย่างต่อเนื่อง(Sustained Oscillations) ดังรูปที่ 3.10 (ถ้าผลตอบสนองเวลาไม่เกิดการแกว่งอย่างต่อเนื่อง วิธีนี้จะใช้ไม่ได้) จากนั้นค่าของ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

K_{cr}(Critical Gain) คือ อัตราขยายที่ทำให้ผลตอบสนองเวลาเกิดการแกว่งอย่างต่อเนื่อง

T_{cr} (oscillation Period) คือคาบเวลาของการแกว่งอย่างต่อเนื่อง



ภาพที่ 4.4 แสดงผลตอบสนองเวลาเกิดการแกว่งอย่างต่อเนื่อง เมื่อปรับโดยใช้วิธี Ultimate Method

Ziegler-Nichols ได้กำหนดค่าของ K_c , T_i และ T_d สำหรับตัวควบคุมแบบต่างๆที่ใช้การปรับค่าพารามิเตอร์ของตัวควบคุม PID ตามวิธี Ultimate Method ไว้ในตารางที่ 4.2

ตารางที่ 4.2 แสดงสูตรสำหรับหาค่าพารามิเตอร์ของตัวควบคุม PID ตามวิธี Ultimate Method

Controller Type	Proportional Gain Kc	Integral Time Ti	Derivative Time Td
Proportional only (P)	Kcs/2	-	-
Proportional- Integral (PI)	Kcs/2.2	Tcs/1.2	-
Proportional- Integral- Derivative (PID)	Kcs/1.7	Tcs/2	Tcs/8

การหาค่าพารามิเตอร์ของตัวควบคุม PID โดยวิธีของ Ziegler-Nichols นี้ ไม่ใช่ค่าที่ดีที่สุดที่จะนำมาใช้งานได้ทันทีเป็นเพียงค่าที่ใกล้เคียงเท่านั้น ดังนั้นผู้ควบคุมจะต้องทำการปรับค่าแบบละเอียด (Fine Tuning) อีกครั้งหนึ่งด้วยวิธีการปรับรายละเอียดเพิ่มเติมถ้าต้องการผลการตอบสนองเป็นที่พอใจ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.4.3 วิธี Trial and Error (วิธีการลองผิดลองถูก)

วิธีการนี้เป็นวิธีการที่เหมาะสมกับผู้ที่มีประสบการณ์ในด้านการปรับพารามิเตอร์ PID มามากแล้วโดยการควบคุมจะต้องเป็นระบบปิดและตำแหน่งการควบคุมจะต้องอยู่ในตำแหน่ง Automatic Control ส่วนวิธีการต่อไปก็คือ

- วิธีการปรับแบบ Proportional + Integral

1.ให้ตัวควบคุมทำงานในรูปแบบ Proportional Control เพียงอย่างเดียวคือไม่ให้ Integral และ Derivative ทำงาน

2. ทำการปรับค่า Proportional Gain ไปจนกระทั่งค่า pv เข้าใกล้ค่าเป้าหมายซึ่งโดยทั่วไปก็จะต่ำกว่าค่าเป้าหมายเพียงเล็กน้อย

3. ทดลองค่อยๆปรับค่า Ti (วินาที) เพิ่มขึ้นเพื่อชดเชยค่า Offset ที่เกิดขึ้นแล้วดูผลว่าค่า pv เข้าสู้เป้าหมายตามที่ต้องการหรือยังและถ้ายังก็ค่อยๆปรับขึ้นไปอีกจนกระทั่งได้ตามต้องการส่วนในช่วงนี้ก็พยายามลด Proportional Gain เพื่อลดการเกิดการแกว่ง (Oscillation)

- วิธีการปรับแบบ Proportional + Integral + Derivative

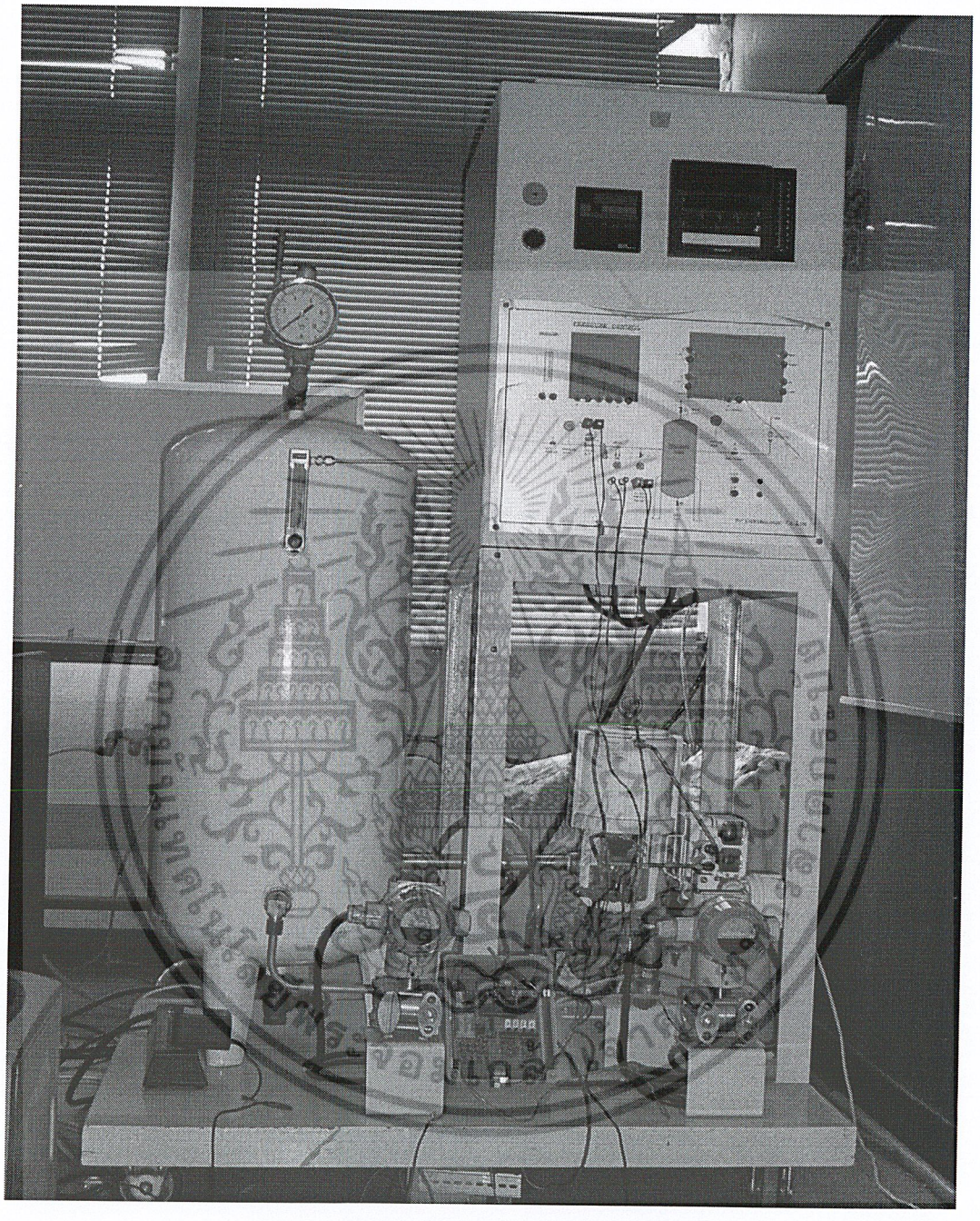
1. ทำตามวิธีการแบบ Proportional + Integral ทั้ง 3 หัวข้อ

2. เมื่อต้องการการตอบสนองที่รวดเร็วขึ้นเรามักใช้ค่าอัตราส่วนโดยทั่วไประหว่างค่า Integral Time และ Derivative Time เป็น 4/1 เป็นต้นไปพร้อมกับการเพิ่มลดของค่า Proportional Gain ไปด้วยเพื่อป้องกันการเกิด Overshoot และการเกิด Hunting

3. เนื่องด้วยวิธีการแบบนี้เป็นความพึงพอใจเฉพาะบุคคลจึงยากที่จะบอกกฎเกณฑ์ตายตัวแต่วิธีการเริ่มต้นในการปรับค่ามักเหมือนกัน

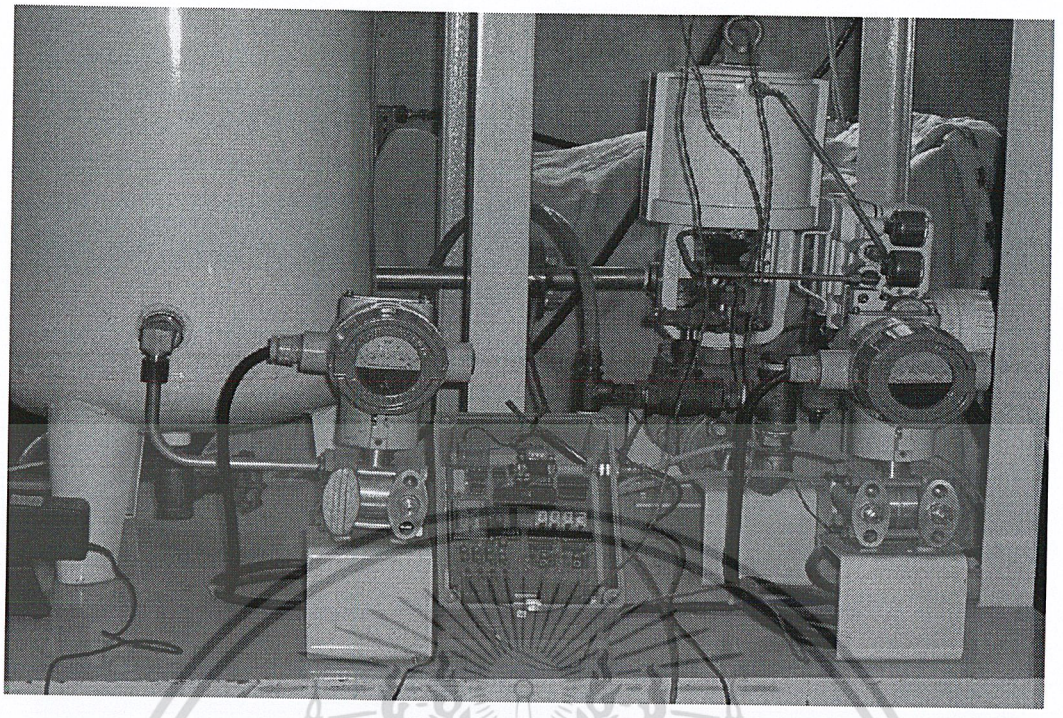
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.5 กระบวนการควบคุมแรงดัน

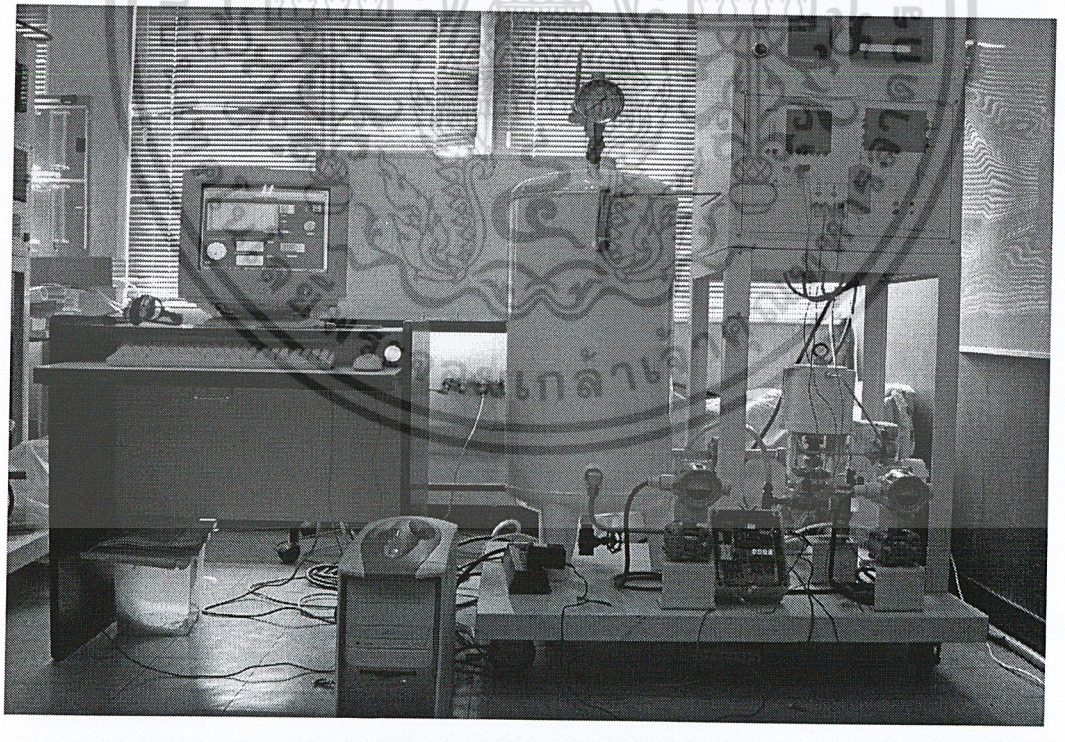


ภาพที่ 4.5 แสดงกระบวนการควบคุมแรงดัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

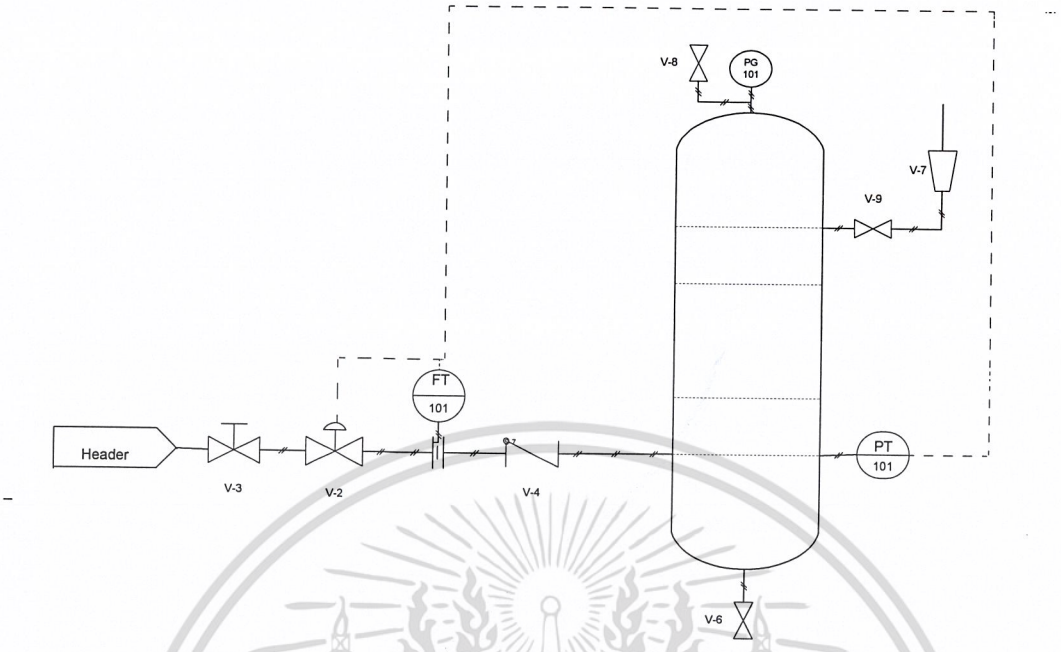


ภาพที่ 4.6 แสดงส่วนประกอบของกระบวนการควบคุมแรงดัน

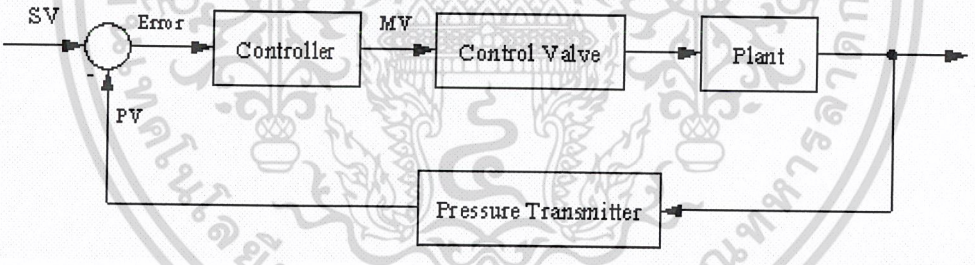


ภาพที่ 4.7 แสดงการควบคุมแรงดันด้วยโปรแกรม LabVIEW บนคอมพิวเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 4.8 แสดงP & ID ของกระบวนการ



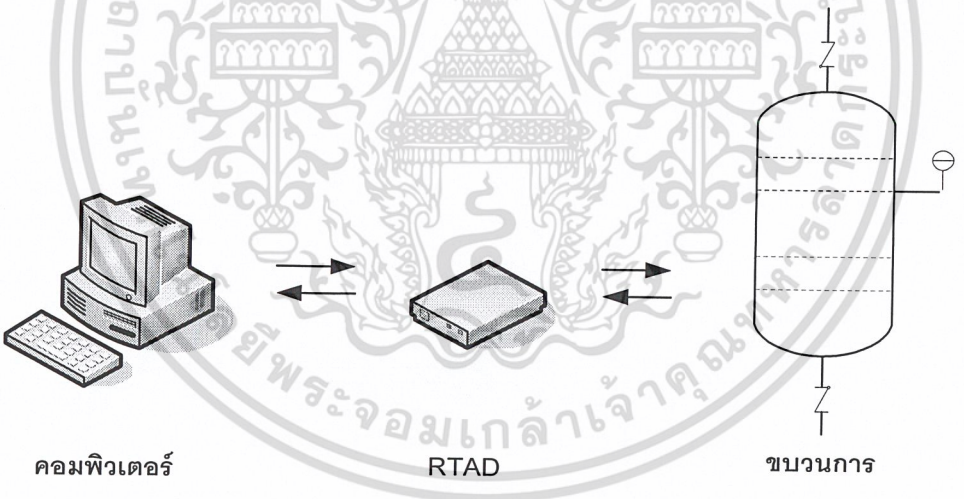
ภาพที่ 4.9 แสดง Block diagram ของกระบวนการควบคุมแรงดัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5
โปรแกรมควบคุมกระบวนการ

5.1 แนวคิดและหลักการในการออกแบบโปรแกรม

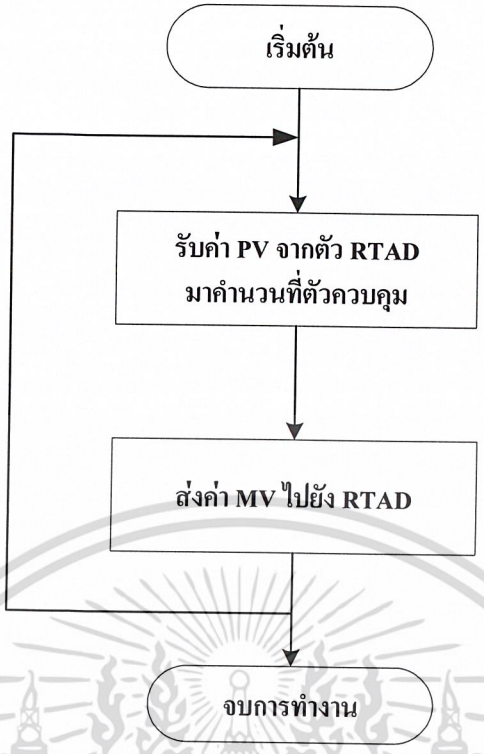
กระบวนการที่จะทำการควบคุมคือกระบวนการกักเก็บแรงดันโดยจะควบคุมแรงดันในถังเก็บให้คงที่ตามค่าที่ตั้งไว้ โดยตัวควบคุมจะเขียนขึ้นโดย LabVIEW บนคอมพิวเตอร์ ดังนั้นค่า PV จะต้องถูกนำมาประมวลผลใน LabVIEW บนคอมพิวเตอร์ และจากนั้น LabVIEW จะทำการคำนวณหาค่า MV เพื่อส่งไปยังตัวปรับกระบวนการ เพื่อทำให้ PV และ SV มีค่าเท่ากันตลอดเวลา ซึ่งจะมีตัวกลางในการติดต่อสื่อสารระหว่างคอมพิวเตอร์กับกระบวนการคือ RTAD โดยจะทำหน้าที่แปลงสัญญาณดิจิตอลเป็นอนาล็อกและอนาล็อกเป็นดิจิตอล ค่า MV จาก LabVIEW บนคอมพิวเตอร์จะถูกส่งมาที่ RTAD เพื่อทำการแปลงเป็นสัญญาณอนาล็อกส่งไปยังวาล์วควบคุม ในทำนองเดียวกัน ค่า PV ที่ได้จาก Transmitter ก็จะส่งมาที่ RTAD เพื่อแปลงเป็นสัญญาณดิจิตอลส่งไปคำนวณใน LabVIEW บนคอมพิวเตอร์ การส่งผ่านข้อมูลระหว่างคอมพิวเตอร์กับ RTAD จะใช้การสื่อสารทาง RS-232



ภาพที่ 5.1 แสดงการติดต่อสื่อสารระหว่างคอมพิวเตอร์กับกระบวนการ

การทำงานของโปรแกรมหลักเมื่อเริ่มต้น Run โปรแกรมก็คือ รับค่า PV ที่อ่านได้จาก RTAD เข้ามาคำนวณเพื่อหาค่า MV จากนั้นจะส่งค่า MV ที่คำนวณได้ออกไปที่ RTAD เพื่อส่งต่อ MV ไปให้วาล์วควบคุมเพื่อปรับให้ได้ค่า PV ที่เท่ากับค่า SV หลังจากที่กระบวนการได้ปรับแล้วค่า PV ตัวใหม่ก็จะถูกอ่านเข้ามาคำนวณหาค่า MV อีกครั้ง การทำงานของโปรแกรมหลักจะวนลูปเช่นนี้จนกว่าจะมีการหยุดรันโปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 5.2 แสดงFlow Chart แสดงการทำงานของโปรแกรมหลัก

5.2 โปรแกรมควบคุมแรงดัน

สำหรับโปรแกรมควบคุมแรงดันนี้ PV ก็คือค่าของแรงดันที่อยู่ในถังเก็บ ซึ่งได้รับมาจาก Pressure Transmitter ที่ติดตั้งอยู่ที่ก้นถังสัญญาณที่ได้จะเป็นค่าแรงดัน 1 ถึง 5 Vdc โดยที่ 1 Vdc เท่ากับ 0 kgf/cm^2 และที่ 5 Vdc จะเท่ากับ 2 kgf/cm^2 ส่วนค่า MV คือค่าที่ได้จากการคำนวณของตัวควบคุมที่จะส่งไปให้กับวาล์วควบคุม ซึ่งสัญญาณที่จะส่งไปให้วาล์วควบคุมจะอยู่ในรูปของกระแส 4 ถึง 20 mA โดยที่ 4 mA วาล์วควบคุมจะปิดและที่ 20 mA วาล์วควบคุมจะเปิดร้อยเปอร์เซ็นต์

5.2.1 Front panel ของโปรแกรมควบคุมแรงดัน

การควบคุมและการเฝ้าดูกระบวนการสามารถทำได้บนหน้า Front panel นี้

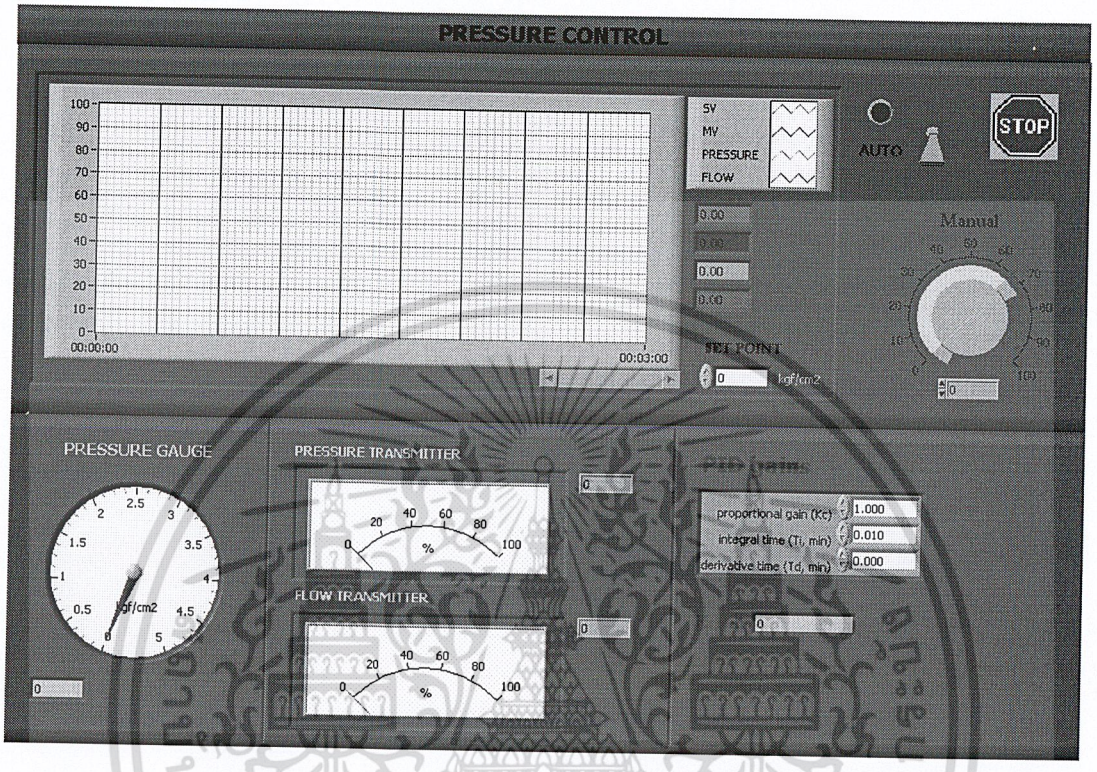
บน Front panel ก็จะประกอบไปด้วยส่วนต่างๆ ต่อไปนี้

1. กราฟแสดงค่าของ SV , MV , แรงดัน และอัตราไหล ซึ่งค่าต่าง ๆ เหล่านี้จะอยู่ในกราฟเดียวกันเพื่อแสดงให้เห็นความสัมพันธ์ของค่าต่าง ๆ เหล่านั้น

2. สวิทซ์เลือก Auto/Manual เลือกที่จะควบคุมกระบวนการเองหรือจะควบคุมโดยใช้ตัวควบคุม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. ปุ่มหมุนปรับเปอร์เซ็นต์การเปิดของวาล์ว โดยปุ่มหมุนนี้จะสามารถใช้ได้ก็ต่อเมื่อสวิทซ์เลือกถูกปรับให้อยู่ที่ Manual เท่านั้น Gauge วัดแรงดันในถัง ซึ่งแรงดันที่อ่านได้จะสัมพันธ์กับแรงดันที่วัดได้จริงใน



ภาพที่ 5.3 แสดง Front panel ของโปรแกรมควบคุมแรงดัน

4. กระบวนการ

5. Pressure Transmitter

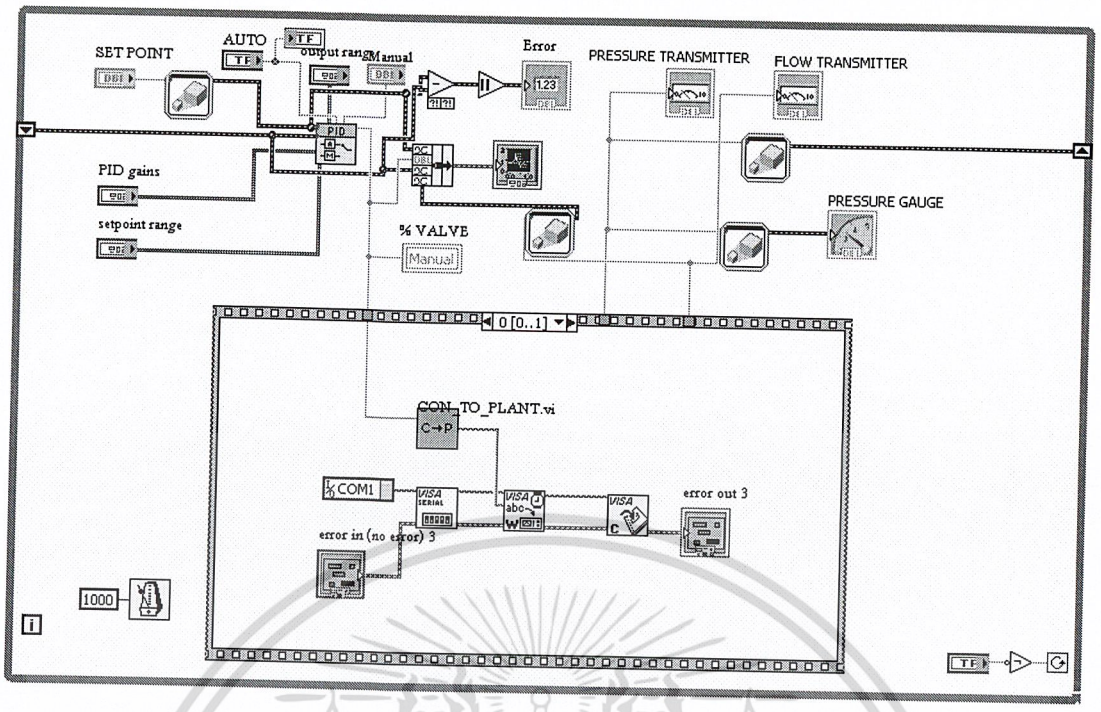
6. Flow Transmitter

7. ที่สำหรับป้อนค่าพารามิเตอร์ของตัวควบคุม

5.2.2 Block Diagram ของโปรแกรมควบคุมแรงดัน

ในส่วนของ Block Diagram จะเป็นที่สำหรับเขียนโปรแกรมควบคุมการทำงานทั้งหมดซึ่งการทำงานของโปรแกรมในส่วนต่างอธิบายได้ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 5.4 แสดงBlock diagram ของโปรแกรมควบคุมแรงดัน

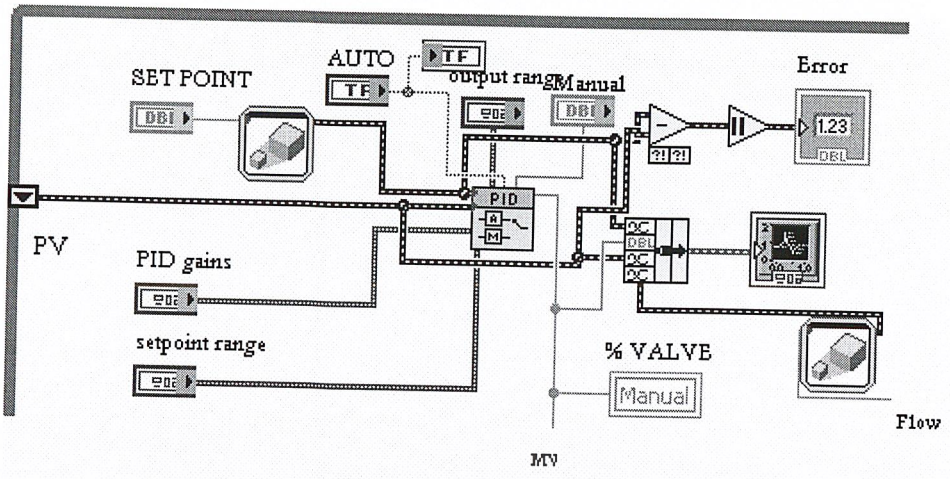
เริ่มต้นการทำงานของโปรแกรมเมื่อเราป้อนค่าพารามิเตอร์ของตัวควบคุมรวมถึงค่า SV ในหน้า Front panel และเลือกสวิทช์ไปที่ Auto จากนั้นกดรันโปรแกรม การทำงานของโปรแกรมจะเป็นดังนี้

5.2.2.1 ส่วนของโปรแกรมตัวควบคุม

ค่าต่าง ๆ จะถูกส่งไปที่ PID Advance.vi โดยการทำงานของตัวควบคุมจะอยู่ในเงื่อนไขที่ว่าค่าที่ได้จากสวิทช์เลือก Auto/Manual เป็น True หรือ False ถ้าเป็น True ตัวควบคุมจะอยู่ในโหมด Auto ค่าต่างๆ เช่น SV , PV และค่าพารามิเตอร์ PID จะถูกนำเข้ามาคำนวณใน PID Advance.vi และจะส่งค่า MV ที่ได้จากการคำนวณส่งออกไปที่ โปรแกรมในส่วนของการติดต่อระหว่างคอมพิวเตอร์กับ RTAD แต่ถ้าค่าที่ได้จากสวิทช์เลือก Auto/Manual เป็นFalse การทำงานของตัวควบคุมจะอยู่ในโหมด Manual คือค่า MV จะได้จากการปรับปุ่มหมุนการเปิดปิดของวาล์วควบคุม

นอกจากนี้แล้วค่า SV และค่า PV จะนำไปลบกันหาค่า Error เพื่อแสดงที่หน้า Front panel ในส่วนของการแสดงค่า ค่า SV, PV, MV และค่าของ Flow จะถูกส่งออกไปแสดงที่กราฟ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



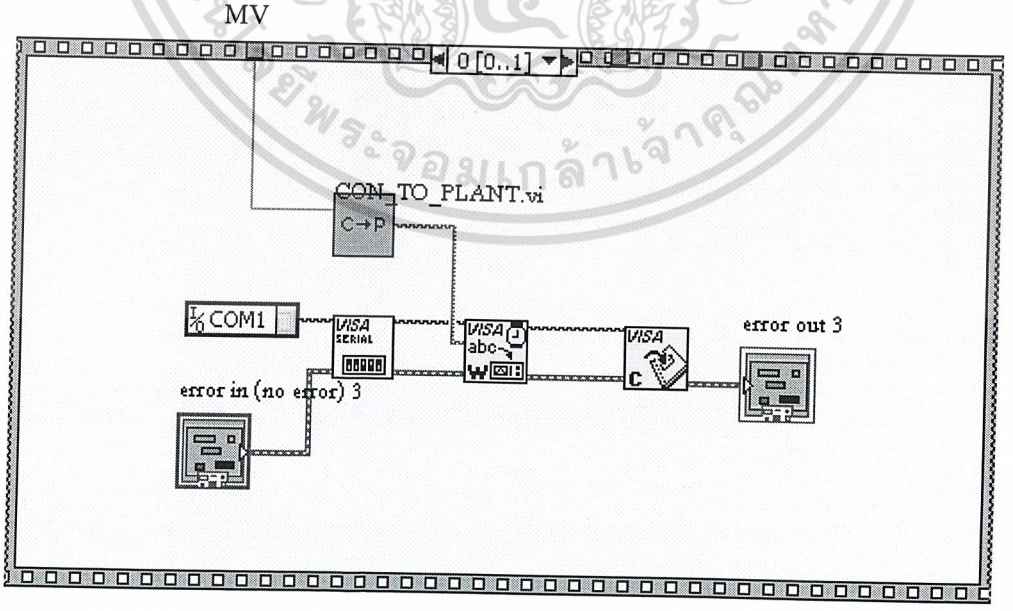
ภาพที่ 5.5 แสดงโปรแกรมตัวควบคุม

5.2.2.2 ส่วนของโปรแกรมติดต่อสื่อสารกับ RTAD

ในส่วนของการติดต่อสื่อสารกับRTADจะใช้ ฟังก์ชัน Stacked Sequence Structure ช่วยในการเขียนโปรแกรมในการทำงานแบบลำดับ ส่วนการส่งคำสั่งออกไปยัง RTAD นั้นจะใช้ฟังก์ชัน Serial ในการ Write และ Read จาก Com port ซึ่งโปรแกรมในส่วนนี้จะประกอบไปด้วย 2 ส่วนคือ

1. ส่วนของการสั่ง Write Analog
2. ส่วนของการสั่ง Read Analog

5.2.2.2.1 โปรแกรมสั่ง Write Analog

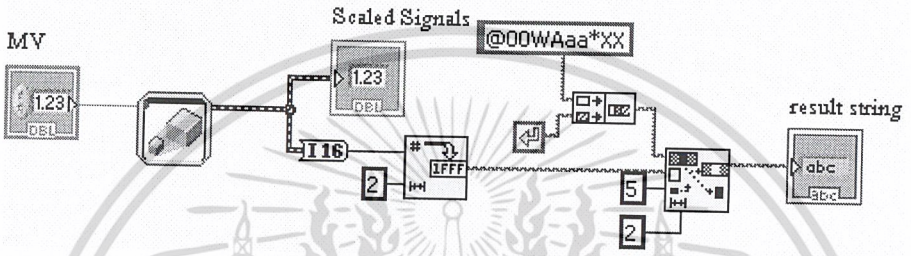


ภาพที่ 5.6 แสดงโปรแกรมสั่ง Write Analog

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

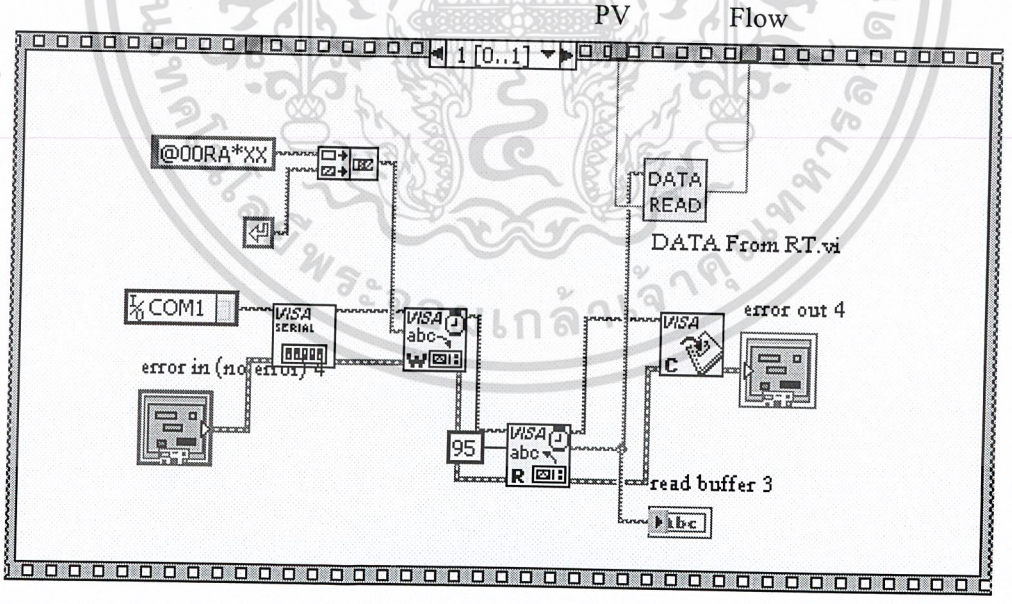
ในลำดับที่ศูนย์ของ ฟังก์ชัน Stacked Sequence Structure เป็นโปรแกรมสั่ง Write Analogจะรับค่า MV ที่ได้จากตัวควบคุมเข้ามาใน CON_TO_PLANT.vi เพื่อเอาค่า MV ไปแปลงเป็นโปรโตคอลที่สามารถติดต่อกับ RTAD ดังนั้นค่าที่ออกมาจาก CON_TO_PLANT.vi จะเป็นชุดโปรโตคอลที่พร้อมจะส่งออกทาง Serial port RS-232

ส่วนของโปรแกรมที่เขียนอยู่ใน CON_TO_PLANT.vi ค่า MV จะถูกนำมาทำ Scaling จาก 0 - 100 % ไปเป็น 0-255 แล้วนำค่าที่ Scale ได้แปลงเป็น Hex String (00-FF) จากนั้นจะถูกส่งเข้าไปแทรกในชุดโปรโตคอล สุดท้ายก็จะได้ชุดคำสั่งที่พร้อมจะส่งไปยัง RTAD



ภาพที่ 5.7 แสดงโปรแกรมใน CON_TO_PLANT.vi

5.2.2.2.2 โปรแกรมสั่ง Read Analog



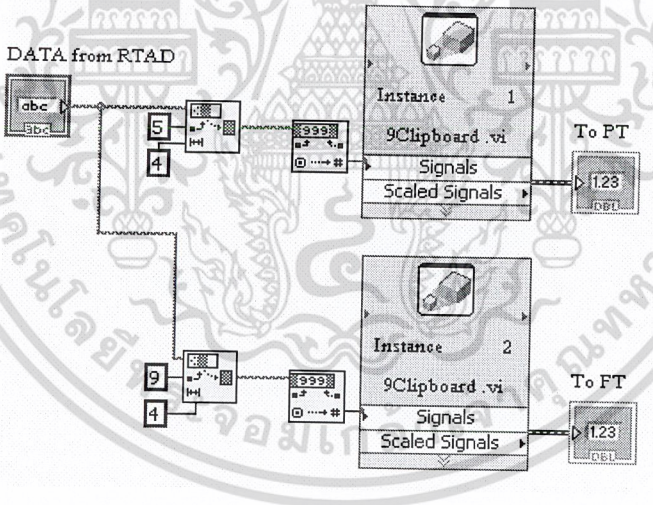
ภาพที่ 5.8 แสดงโปรแกรมสั่ง Read Analog

ในลำดับที่หนึ่งของ ฟังก์ชัน Stacked Sequence Structure เป็นโปรแกรมสั่ง Read Analog โดยจะส่งชุดคำสั่ง Read Analog ออกไปที่ RTAD เพื่ออ่านค่าจากทั้ง 4 Channel ของ RTAD ค่าที่ได้จะถูกส่งไปยัง DATA From RT.vi เพื่อทำการ Scaling ข้อมูลที่ได้มา ข้อมูลที่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Scaling แล้วจะส่งไปแสดงที่ Pressure Transmitter , Flow Transmitter , Chart โดยเฉพาะค่าที่อ่านได้จาก Channel 1 จะ Scaling แล้วส่งต่อไปเป็น PV ให้กับตัวควบคุม ซึ่งการส่งค่า PV ในโปรแกรมจะส่งทาง Shift Register ของ While Loop โดยทุก ๆ ครั้งที่มีการวนรอบใหม่ค่า PV ที่อ่านได้ในแต่ละรอบจะถูกนำไปเก็บไว้ใน Shift Register ตัวที่อยู่ทางฝั่งขวาของ While Loop และเมื่อโปรแกรมวนขึ้นรอบใหม่ ค่า PV Shift Register ฝั่งขวาจะส่งค่าไปที่ Shift Register ตัวที่อยู่ทางฝั่งซ้ายของ While Loop เพื่อส่งค่า PV ให้กับ PID Advance.vi เพื่อใช้คำนวณหา MV ในรอบต่อไป

ส่วนของโปรแกรมที่เขียนอยู่ใน DATA From RT.vi ค่าที่อ่านได้จากทั้ง 4 Channel ของ RTAD จะถูกนำมา Scaling โดยข้อมูลใน Channel 1 คือตัวที่ 5 ถึงตัวที่ 8 จะถูกดึงออกมาทำ Scaling แล้วส่งต่อออกไปยัง Pressure Transmitter ส่วนข้อมูลใน Channel 2 คือตัวที่ 9 ถึงตัวที่ 12 จะถูกดึงออกมาทำ Scaling แล้วส่งต่อออกไปยัง Flow Transmitter โดยการ Scaling ทั้งสอง Channel จะทำการ Scaling จาก 0000 – 9999 ไปเป็น 0-10 ทั้งสอง Channel และค่าที่ Scaling จากทั้งสอง CHANNEL จะถูก Scaling อีกครั้งให้เป็น 0 – 100 % เพื่อเอาไปเป็น PV ป้อนให้กับ PID Advance.vi และเป็น Flow แสดงที่กราฟ



ภาพที่ 5.9 แสดงโปรแกรมใน DATA From RT.vi

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 6
ผลการทดลองหาค่าพารามิเตอร์ของตัวควบคุม

6.1 การหาค่าพารามิเตอร์ของตัวควบคุมโดยวิธี Process Reaction Curve

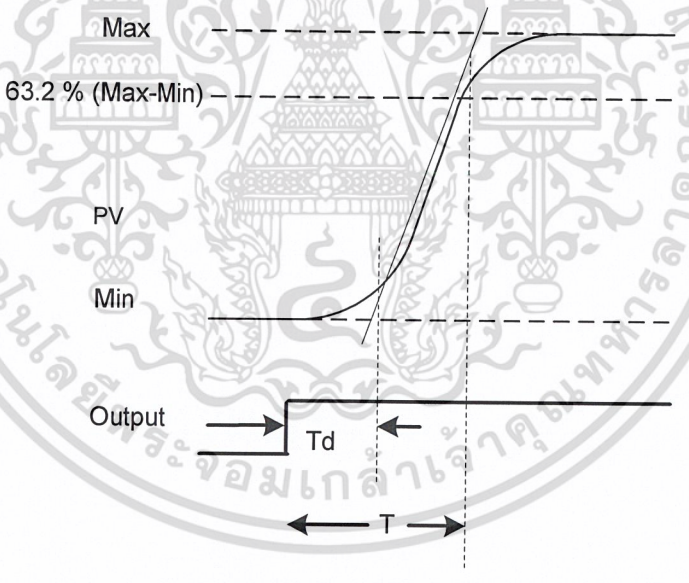
มีวิธีการดังต่อไปนี้คือ

1. ปรับตัวควบคุมให้อยู่ในโหมด Manual

2. จ่าย Output (MV) ที่ค่า ๆ หนึ่ง แล้วสังเกตค่า PV รอจนกว่าค่า PV จะคงที่บันทำค่า MV และ PV ที่ได้

3. จากนั้นหาค่าต่างดังรูปข้างล่าง โดยที่

- Td คือ Deadtime (minutes)
- T คือ Time constant (minutes)
- K คือ Process gain = MV / PV



ภาพที่ 6.1 แสดงผลตอบสนองรูปตัว S เมื่อใช้วิธี Process Reaction Curve

4. คำนวณค่า PB , Ti และ Td ตามสูตรใน ตาราง โดยที่ PB = 100/Kc

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 6.1 แสดงสูตรสำหรับหาค่าพารามิเตอร์ของตัวควบคุม PID ตามวิธี Process Reaction Curve

Controller Type	PB	Ti (minutes)	Td (minutes)
Proportional only (P) Proportional- Integral (PI)	100KTd/T	-	-
Proportional- Integral (PI)	110KTd/T	3.33Td	-
Proportional- Integral- Derivative (PID)	80KTd/T	2.0Td	0.5Td

6.2 ผลการทดลองหาค่าพารามิเตอร์ของตัวควบคุม

ในหน้า Front panel ของโปรแกรมควบคุมแรงดัน เลือกสวิทซ์ให้อยู่ให้อยู่ในMode Manual จากนั้นป้อนค่า MV 3 % รอจนกระทั่ง PV คงที่จากนั้นป้อน MV เพิ่มเป็น10% จะได้ผลตอบสนองของกระบวนการดังกราฟข้างล่าง

จากกราฟผลตอบสนองของกระบวนการจะได้ค่า (สเกลช่องเล็กในกราฟ ช่องละ 5 วินาที) Td = 0.042 นาที T = 0.2 นาที และ K = (10-5)/(4.91-0.9) = 1.247

ค่าพารามิเตอร์ของตัวควบคุมคำนวณได้ดังนี้

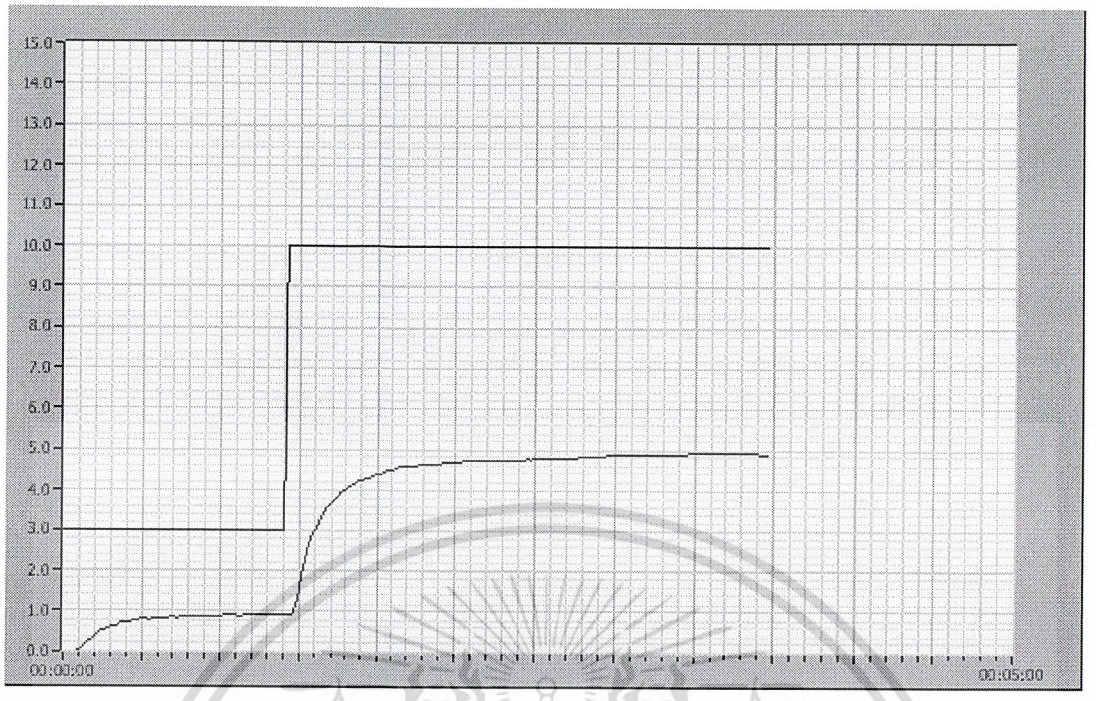
ตัวควบคุมแบบ PI

- Kc = 3.472
- Ti = 0.140

ตัวควบคุมแบบ PID

- Kc = 4.773
- Ti = 0.084
- Td = 0.021

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

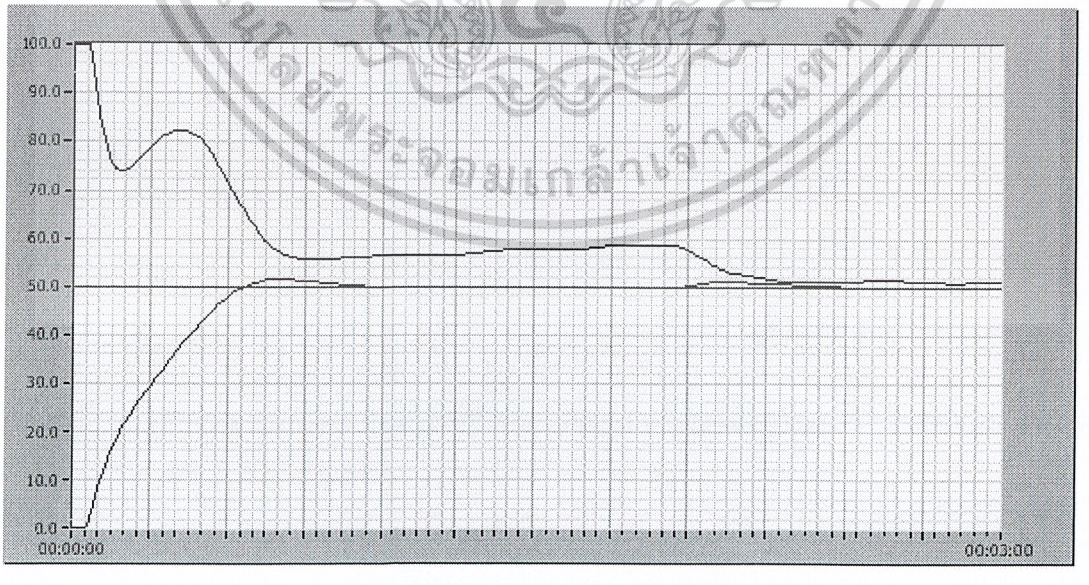


ภาพที่ 6.2 แสดงผลตอบสนองของกระบวนการเมื่อป้อนค่า MV จาก 3% ไปเป็น 10%

6.3 ผลตอบสนองของกระบวนการเมื่อใช้ตัวควบคุมแบบ PI

ตั้งค่า SETPOINT ไว้ที่ 1 kgf/cm^2 หรือที่ 50 % ได้ผลตอบสนองดังนี้

Kc = 3.472 Ti = 0.140

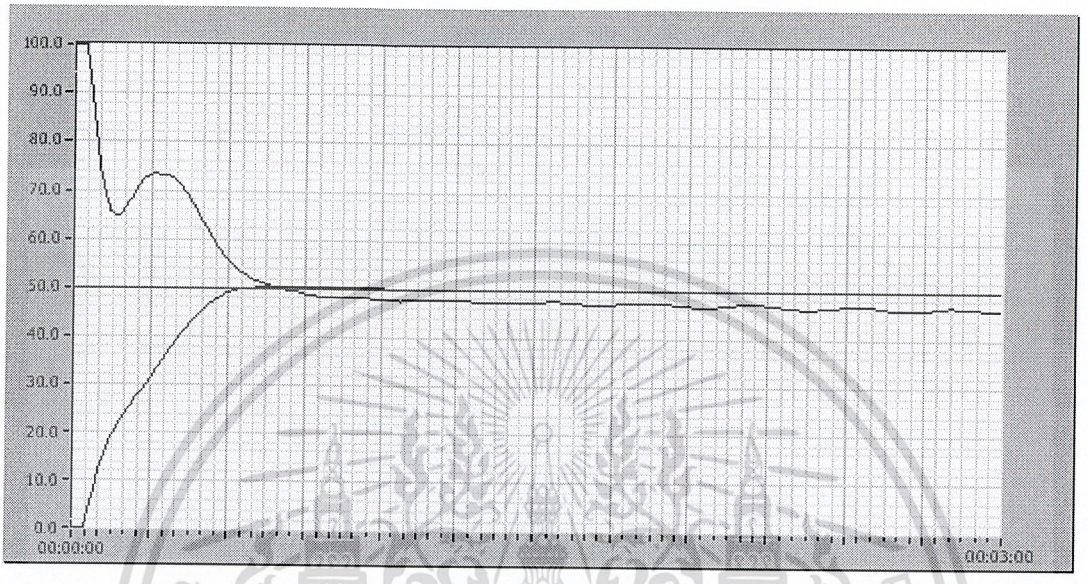


ภาพที่ 6.3 แสดงผลตอบสนองของกระบวนการเมื่อใช้ตัวควบคุมแบบ PI

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

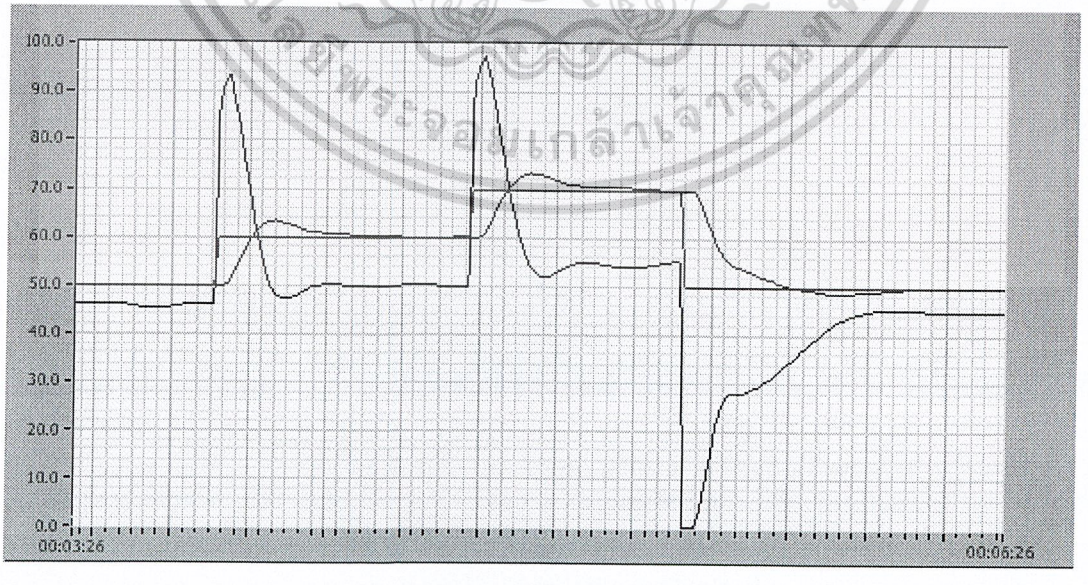
จากผลที่ได้จะเห็นว่าค่าพารามิเตอร์ของตัวควบคุมแบบ PI ที่คำนวณได้ให้ผลตอบสนองเป็นที่น่าพอใจ จะทำการปรับค่าพารามิเตอร์อย่างละเอียดอีกครั้งจะได้ผลตอบสนองดังนี้

Kc = 3.95 Ti = 0.139



ภาพที่ 6.4 แสดงผลตอบสนองของกระบวนการเมื่อปรับค่าพารามิเตอร์อย่างละเอียด

จะเห็นว่าผลตอบสนองที่ได้จะไม่มี Overshoot จากนั้นทำการป้อน SV เป็น 60 %,70% และลดลงมาที่ 50 % ตามลำดับ



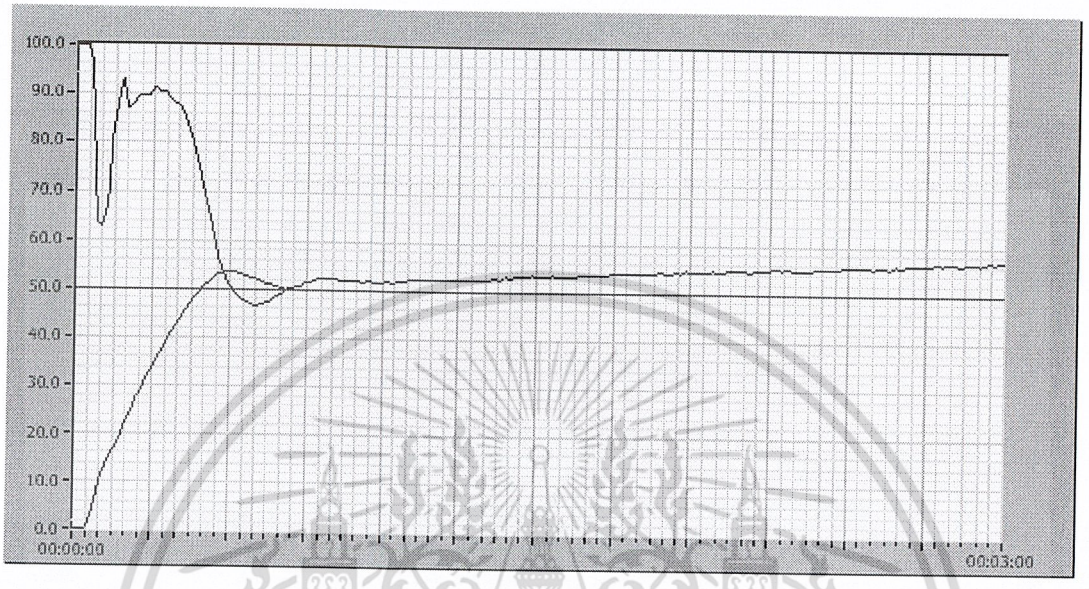
ภาพที่ 6.5 แสดงผลตอบสนองของกระบวนการเมื่อป้อนค่า SV เป็น unit step

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6.4 ผลตอบสนองของกระบวนการเมื่อใช้ตัวควบคุมแบบ PID

ตั้งค่า SETPOINT ไว้ที่ 1 kgf/cm^2 หรือที่ 50% ได้ผลตอสนองดังนี้

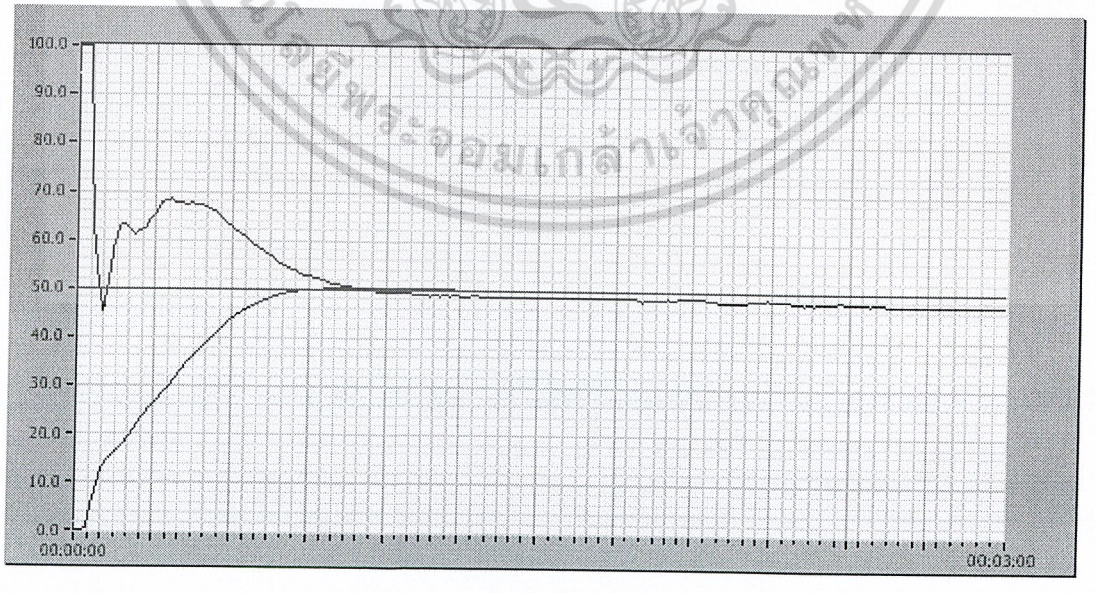
Kc=4.773 Ti = 0.084 Td = 0.021



ภาพที่ 6.6 แสดงผลตอบสนองของกระบวนการเมื่อใช้ตัวควบคุมแบบ PID

จากผลที่ได้จะทำการปรับอย่างละเอียดอีกครั้งเพื่อลด Overshoot จะได้ผลตอบสนองดังรูป

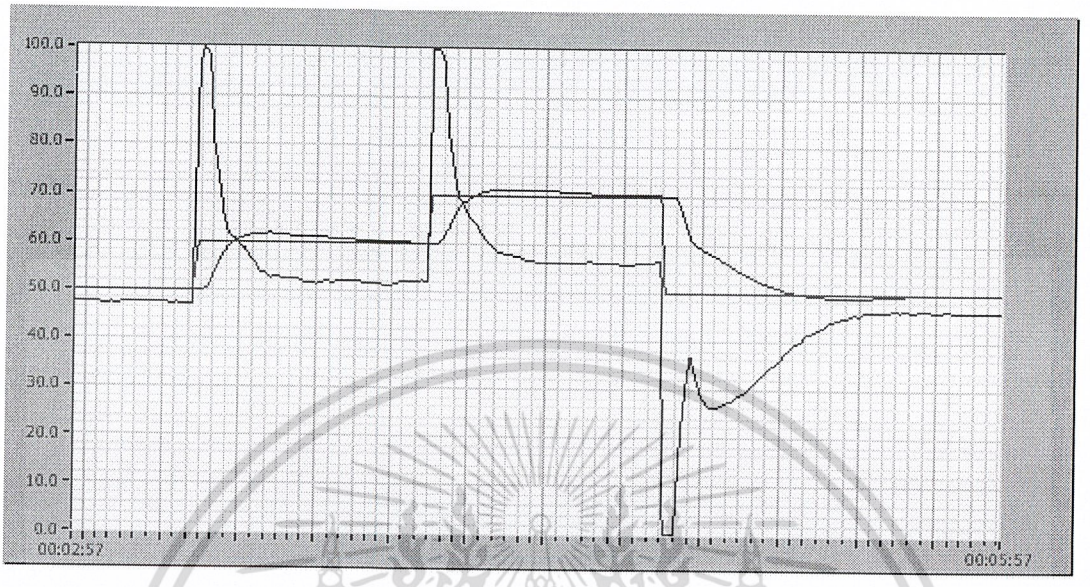
Kc=4.773 Ti = 0.175 Td = 0.017



ภาพที่ 6.7 แสดงผลตอบสนองของกระบวนการเมื่อเพิ่มTi และลด Td

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากนั้นปรับค่า SV เป็น 60% , 70% และ 50% ตามลำดับ ผลตอบสนองที่ได้แสดงดังรูปข้างล่าง



ภาพที่ 6.8 แสดงผลตอบสนองของกระบวนการเมื่อปรับค่า SV

6.5 สรุปผลการทดลอง

ค่าพารามิเตอร์ของตัวควบคุมที่หาได้จากวิธี Process Reaction Curve Method สรุปได้ดังตาราง

ตารางที่ 6.2 แสดงค่าพารามิเตอร์ของตัวควบคุมที่หาได้จากวิธี Process Reaction Curve Method

พารามิเตอร์ ชนิดของตัวควบคุม	Proportional gain (Kc)	Integral time (Ti , min)	Derivative time (Td , min)
PI	3.472	0.140	-
PID	4.773	0.084	0.021

ค่าพารามิเตอร์ของตัวควบคุมหลังจากการปรับแต่งเพื่อให้ผลตอบสนองดีขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 6.3 แสดงค่าพารามิเตอร์ของตัวควบคุมหลังการปรับแต่ง

พารามิเตอร์ ชนิดของตัวควบคุม	Proportional gain (Kc)	Integral time (Ti , min)	Derivative time (Td , min)
PI	3.95	0.139	-
PID	4.773	0.175	0.17

สรุปผลการทดลอง ค่าพารามิเตอร์ของตัวควบคุมที่หาได้จากโปรแกรมควบคุมกระบวนการที่เขียนขึ้น โดย LabVIEW ด้วยวิธี Process Reaction Curve Method ของ Ziegler-Nichols ให้ผลตอบสนองของกระบวนการเป็นที่น่าพอใจ ทั้ง Overshoot น้อย Setting time ก็อยู่ในระดับที่น่าพอใจและเมื่อทดลองสร้าง Disturbance ให้กับระบบ โปรแกรมควบคุมก็ยังสามารถรักษาค่าแรงดันให้กลับเข้าสู่ค่าเป้าหมายได้ดังเดิม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 7
สรุปผลการวิจัยและข้อเสนอแนะ

7.1 บทสรุป

จากผลการทดลองหาค่าพารามิเตอร์ของตัวควบคุมและผลตอบสนองของกระบวนการเมื่อใช้ค่าพารามิเตอร์ที่หามาได้นั้น จะเห็นได้ว่าโปรแกรมควบคุมที่เขียนขึ้นโดย LabVIEW สามารถควบคุมกระบวนการได้เป็นอย่างดี ทางด้านการนำเสนอและแสดงข้อมูล LabVIEW ก็สามารถทำได้ดีและมีฟังก์ชันการใช้งานที่หลากหลาย การเขียนรวมทั้งการแก้ไขและพัฒนาโปรแกรม ก็ทำได้ง่าย สะดวกและใช้เวลาน้อยเมื่อเทียบกับภาษาโปรแกรมอื่น ๆ ดังนั้น LabVIEW จึงเป็นโปรแกรมภาษาอีกตัวหนึ่งที่น่าสนใจไม่น้อย

7.2 ข้อเสนอแนะและแนวทางพัฒนา

จะเห็นได้ว่าโปรแกรมในส่วนของการติดต่อสื่อสารยังทำได้ไม่ดีนักเนื่องจากข้อมูลที่จะทำการส่งออกไปให้ตัว RTAD ต้องทำการแปลงให้เป็นสตริงฐาน 16 เพื่อแทรกเข้าไปในชุดโปรโตคอลที่ใช้ในการติดต่อ ข้อมูลที่ได้หลังจากการแปลงจะคลาดเลื่อนเนื่องจากข้อมูลก่อนทำการแปลงบางครั้งจะไม่ใช่จำนวนเต็ม เป็นทศนิยมบ้างทำให้สตริงฐาน 16 ที่ได้จะเป็นการปัดเศษ ทำให้ไม่ได้ค่าที่ต้องการจะส่งออกไปจริง ๆ การแก้ปัญหาก็คือเปลี่ยนวิธีการติดต่อกับกระบวนการจากเดิมใช้ RTAD ไปใช้การ์ดแทนปัญหาตรงนี้ก็จะหมดไป

ส่วนแนวทางในการพัฒนาโปรแกรมนอกจากที่จะเปลี่ยนมาเป็นใช้การ์ดในการติดต่อสื่อสารแล้ว อาจจะพัฒนาตัวโปรแกรมให้มีฟังก์ชั่นการใช้งานที่มากขึ้น เช่น การใช้ระบบฐานข้อมูล(Data Base), การควบคุมผ่าน Internetฯลฯ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

[1] LabVIEW for everyone:graphical programming made even easier / Lisa K. Wells,Jeffrey Travis, Upper Saddle River.NI: Prentice Hall PTR,1997

[2] Laboratory of Microprocessor and Interfacing Technique,สถาบันเทคโนโลยีราชมงคลวิทยาเขตพระนครเหนือ

[3] เรียนลัด LabVIEW,เจริญ เพชรมุณี, กรุงเทพ:ซีเอ็ดยูเคชั่น, 2547

[4] เอกสารประกอบการสอนวิชา Instrumentation Engineering ภาควิชาวิศวกรรมการวัดคุม สาขาวิศวกรรมการวัดคุม คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้