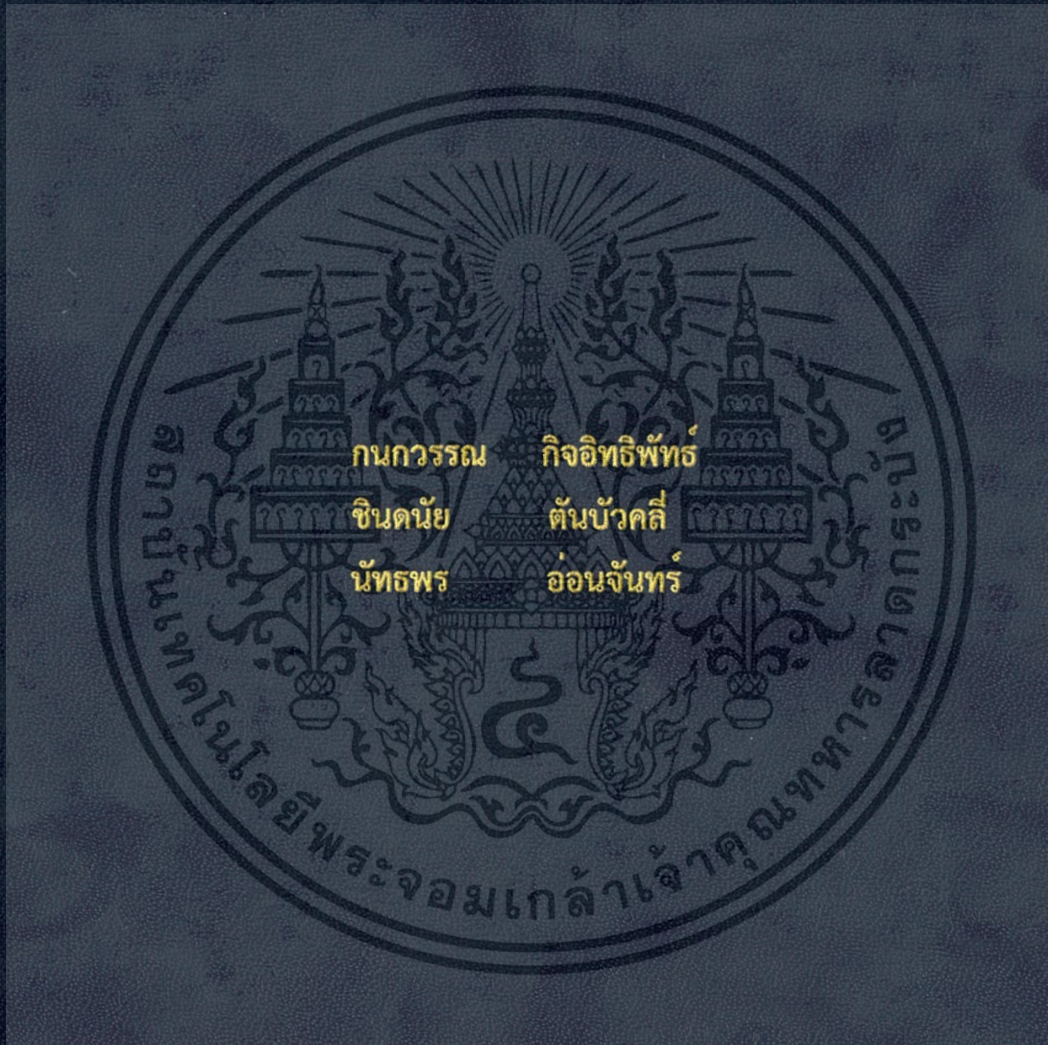


การควบคุมอุณหภูมิด้วยเครื่องควบคุม PLC ผ่าน SERIAL PORT  
ในระบบ SCADA  
TEMPERATURE CONTROL BY PLC THROUGH SERIAL  
PORT NETWORK BASE NO SCADA SYSTEM



ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต  
สาขาวิชาวิศวกรรมการวัดคุม  
คณะวิศวกรรมศาสตร์  
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง  
ปีการศึกษา 2557

การควบคุมอุณหภูมิด้วยเครื่องควบคุม PLC ผ่าน SERIAL PORT  
ในระบบ SCADA  
TEMPERATURE CONTROL BY PLC THROUGH SERIAL  
PORT NETWORK BASE ON SCADA SYSTEM



ปฏิญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต  
สาขาวิศวกรรมการวัดคุม  
คณะวิศวกรรมศาสตร์  
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง  
ปีการศึกษา 2557

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

TEMPERATURE CONTROL BY PLC THROUGH SERIAL  
PORT NETWORK BASE ON SCADA SYSTEM



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 2014


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญาานิพนธ์ปีการศึกษา 2557  
คณะวิศวกรรมศาสตร์  
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง  
ใบรับรองปริญญาานิพนธ์

หัวข้อปริญญาานิพนธ์ การควบคุมอุณหภูมิด้วยเครื่องควบคุม PLC ผ่าน Serial Port ในระบบ SCADA  
TEMPERATURE CONTROL BY PLC THROUGH SERIAL PORT NETWORK BASE ON SCADA SYSTEM

นักศึกษาผู้จัดทำ นางสาวนัทพร อ่อนจันทร์ รหัสนักศึกษา 54010694  
นางสาวกนกวรรณ กิจอิทธิพัทธ์ รหัสนักศึกษา 54010004  
นายชินดนัย ดันบัวคลี่ รหัสนักศึกษา 54010330

ปริญญา วิศวกรรมศาสตรบัณฑิต  
สาขาวิชา วิศวกรรมการวัดคุม  
ปีการศึกษา 2557

อาจารย์ผู้ควบคุมปริญญาานิพนธ์	ลายมือชื่อ
รองศาสตราจารย์สุพรรณ กุลพานิชย์	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อปริญญานิพนธ์ การควบคุมอุณหภูมิด้วยเครื่องควบคุม PLC ผ่านเครือข่าย SERIAL PORT ในระบบ SCADA

TEMPERATURE CONTROL BY PLC THROUGH SERIAL PORT NETWORK  
BASE ON SCADA SYSTEM

นักศึกษาผู้จัดทำ นางสาวกนกวรรณ กิจอิทธิพัทธ์ รหัสนักศึกษา 54010004

นายชินดนัย ต้นบัวคลี่ รหัสนักศึกษา 54010330

นางสาวนัทพร อ่อนจันทร์ รหัสนักศึกษา 54010694

อาจารย์ที่ปรึกษา รองศาสตราจารย์สุพรรณ กุลพามิษฐ์

ปีการศึกษา 2557

### บทคัดย่อ

ปริญญานิพนธ์ฉบับนี้มีวัตถุประสงค์เพื่อศึกษาการควบคุมอุณหภูมิ ด้วยเครื่องควบคุมพีแอลซีผ่าน Serial Port ในรูปแบบ SCADA โดยอาศัยแบบจำลองกระบวนการ Temperature Simulator ซึ่งภายในประกอบด้วยตัวกำเนิดความร้อน (Heater) ที่เป็นตัวแปรเอาต์พุต และทำการวัดอุณหภูมิป้อนกลับด้วยเทอร์โมคัปเปิลที่เป็นตัวแปรอินพุต อาศัยโปรแกรมแลตเตอร์และฟังก์ชัน PID ของเครื่องควบคุมพีแอลซี เป็นตัวควบคุมกระบวนการแบบป้อนกลับชนิด First Order ค่าตัวแปรกระบวนการสามารถนำมาแสดงบนจอคอมพิวเตอร์ ที่มีการใช้โปรแกรม OPC Server ทำงานร่วมกับโปรแกรมกราฟฟิก WinCC ทำหน้าที่เป็น HMI (Human Machine Interface) เพื่อเฝ้ามองและติดตามค่าตัวแปรกระบวนการได้อย่างต่อเนื่อง

**Thesis Title** TEMPERATURE CONTROL BY PLC THROUGH SERIAL PORT NETWORK  
BASE ON SCADA SYSTEM

**Authors** Ms. Kanokwan Kitaitthiphath  
Mr. Chindanai Tanbuaklee  
Ms. Nuttaporn Onchan

**Thesis Advisor** Assoc. Prof.Suphan Kulphanit

### ABSTRACT

This project objective is to study to control the temperature with PLC by using the serial port in the SCADA. Using the temperature simulator that have a thermal generator (Heater) is a variable output and feedback the measure of the temperature with thermocouple that is a variable input. Using the ladder programs and the PID function of the PLC to control process feedback of First Order. Variable of the process can be shown on a computer screen with the program OPC Server working with the WinCC graphic serve as HMI (Human Machine Interface) to watch and track variable process.

## กิตติกรรมประกาศ

ปริญญาานิพนธ์ฉบับนี้สำเร็จลุล่วงได้ด้วยดีเพราะได้รับความอนุเคราะห์จาก รองศาสตราจารย์ สุพรรณ กุลพานิชย์ ที่ได้ให้คำปรึกษา คำแนะนำ รวมถึงอุปกรณ์แก่ผู้วิจัยตลอดมา คณาจารย์ภาควิชาวิศวกรรมการวัดคุมทุกท่านที่ได้มอบคำแนะนำและกำลังใจในการทำวิจัยตลอดมา เพื่อนๆ พี่ๆ ที่คอยช่วยให้กำลังใจในการทำวิจัยและความช่วยเหลือในด้านต่างๆ

คุณค่า ประโยชน์และคำชมเชยอันได้จากปริญญาานิพนธ์ฉบับนี้ ผู้วิจัยขอมอบแด่ผู้มีพระคุณทุกท่าน

คณะผู้จัดทำ



# สารบัญ

	หน้า
บทคัดย่อภาษาไทย.....	I
บทคัดย่อภาษาอังกฤษ.....	II
กิตติกรรมประกาศ.....	III
สารบัญ.....	IV
สารบัญตาราง.....	VII
สารบัญภาพ.....	VIII
บทที่ 1 บทนำ.....	1
1.1 ความเป็นมาและแรงจูงใจในการทำปริญญานิพนธ์.....	1
1.2 วัตถุประสงค์ของปริญญานิพนธ์.....	2
1.3 ขอบเขตของปริญญานิพนธ์.....	2
1.4 ขั้นตอนการศึกษา.....	2
บทที่ 2 ทฤษฎีและหลักการที่เกี่ยวข้อง.....	3
2.1 กล่าวนำ.....	3
2.2 การควบคุมแบบป้อนกลับด้วยตัวควบคุม PID.....	3
2.3 กริยาการควบคุมของการควบคุมแบบป้อนกลับ.....	5
2.3.1 กริยาการควบคุมแบบ ON – OFF .....	5
2.3.2 กริยาการควบคุมแบบ Proportional .....	5
2.3.3 กริยาการควบคุมแบบ Integral.....	6
2.3.4 กริยาการควบคุมแบบ Derivative .....	7
2.3.5 กริยาการควบคุมแบบ Proportional- Integral .....	7
2.3.6 กริยาการควบคุมแบบ Proportional- Derivative.....	8
2.3.7 กริยาการควบคุมแบบ Proportional-Integral-Derivative.....	9
2.4 การปรับค่าพารามิเตอร์ของตัวควบคุมพีไอดี.....	10
2.4.1 การปรับค่าพารามิเตอร์ของตัวควบคุมพีไอดี โดยวิธีของ Ziegler-Nichols.....	10
2.4.1.1 วิธี Process Reaction Curve.....	10
2.4.1.2 วิธี Ultimate Method.....	11
2.4.2 การปรับค่าพารามิเตอร์ของตัวควบคุมพีไอดี โดยวิธี Damped Oscillation....	12
2.4.3 การปรับค่าพารามิเตอร์ของตัวควบคุมพีไอดี โดยวิธี Trial & Error Close-Loop Tuning.....	13
2.5 ความรู้พื้นฐานเกี่ยวกับพีแอลซี.....	14

## สารบัญ (ต่อ)

	หน้า
2.5.1 ชนิดของพีแอลซี.....	15
2.5.1.1 พีแอลซีชนิดบัสล๊อค.....	15
2.5.1.2 พีแอลซีชนิดโมดูล.....	17
2.5.2 ภาษาที่ใช้ในการเขียนโปรแกรมให้กับพีแอลซี.....	19
2.5.3 อุปกรณ์สำหรับการโปรแกรม.....	21
2.5.3.1 ตัวป้อนโปรแกรมแบบมือถือ.....	21
2.5.3.2 คอมพิวเตอร์.....	22
2.5.4 ระบบสื่อสาร.....	23
2.5.5 โครงสร้างของพีแอลซี.....	24
2.5.5.1 ซีพียู.....	24
2.5.5.2 หน่วยความจำ.....	25
2.5.5.3 ภาคอินพุต.....	25
2.6 ระบบ SCADA.....	26
2.6.1 โครงสร้างของ SCADA.....	27
2.6.2 โครงสร้างด้านซอฟต์แวร์.....	27
2.6.3 โครงสร้างด้านการสื่อสาร.....	28
2.6.4 โครงสร้างอินเทอร์เน็ต.....	29
2.6.5 โครงสร้างความสามารถในการขยายระบบ.....	29
2.6.6 โครงสร้างการสำรองระบบ.....	29
2.6.7 หน้าที่การทำงาน.....	30
2.6.8 ระบบแสดงผลแบบ MMI.....	30
2.6.9 ระบบแสดงกราฟสัญญาณแบบต่อเนื่อง.....	30
2.6.10 ระบบแจ้งเตือน.....	30
2.6.11 การทำงานแบบ Automation.....	31
2.6.12 การสร้างและพัฒนา.....	31
2.6.13 เครื่องมือในการพัฒนา.....	31
2.7 OPC.....	32
<b>บทที่ 3 ขั้นตอนการดำเนินงาน.....</b>	<b>35</b>
3.1 อุปกรณ์และสถานที่ดำเนินการ.....	35
3.1.1 อุปกรณ์.....	35
3.1.2 สถานที่ดำเนินการ.....	36

## สารบัญ (ต่อ)

	หน้า
3.2 การห่าย่นควบคุมการทำงานด้านตัวแปรอินพุท.....	36
3.3 การห่าย่นควบคุมการทำงานด้านตัวแปรเอาต์พุท.....	39
3.4 ศึกษาผลตอสองการเคลื่อนที่แบบสแต็ปวงรอบเปิด.....	44
3.5 ศึกษาผลตอสองการเคลื่อนที่แบบสแต็ปวงรอบปิดด้วย PID.....	50
3.6 ตั้งค่าโปรแกรมสำหรับการเชื่อมต่อ OPC Server และ SCADA.....	56
3.7 เขียนกราฟฟิกสำหรับการแสดงผล.....	58
<b>บทที่ 4 ผลการทดลอง.....</b>	<b>59</b>
4.1 ย่นควบคุมการทำงานด้านตัวแปรอินพุท.....	59
4.2 ย่นควบคุมการทำงานด้านตัวแปรเอาต์พุท.....	61
4.3 ผลตอสองการเคลื่อนที่แบบสแต็ปวงรอบเปิด.....	64
4.4 ผลตอสองวงรอบปิดด้วย PID.....	65
4.5 การแสดงผลผ่าน OPC Server ด้วยโปรแกรม WinCC.....	73
<b>บทที่ 5 สรุปผลการทดลองและข้อเสนอแนะ.....</b>	<b>76</b>
5.1 สรุปผลการทดลอง.....	76
5.2 ข้อเสนอแนะ.....	76
<b>บรรณานุกรม.....</b>	<b>77</b>
<b>ภาคผนวก.....</b>	<b>78</b>

## สารบัญตาราง

ตารางที่	หน้า
2.1 ตารางแสดงค่าพารามิเตอร์ของตัวควบคุมแบบต่างๆตามวิธี Process Reaction Curve.....	11
2.2 ตารางแสดงค่าพารามิเตอร์ของตัวควบคุมแบบต่างๆตามวิธี Ultimate Method.....	12
2.3 ตารางแสดงเปรียบเทียบคุณสมบัติของ PLC.....	19



## สารบัญรูป

รูปที่	หน้า
2.1 แสดงโครงสร้างของระบบควบคุมแบบป้อนกลับโดยทั่วไป.....	4
2.2 แสดงการทำงานของระบบควบคุมแบบ ON-OFF.....	5
2.3 แสดงคุณสมบัติของการทำงานของระบบควบคุมแบบ Proportional.....	6
2.4 แสดงผลตอบสนองของการทำงานของระบบควบคุมแบบ Integral.....	7
2.5 แสดงตัวอย่างผลตอบสนองของการทำงานของระบบควบคุมแบบ Derivative.....	7
2.6 แสดงตัวอย่างผลตอบสนองของการทำงานของระบบควบคุมแบบ PI (Direct action).....	8
2.7 แสดงตัวอย่างผลตอบสนองของการทำงานของระบบควบคุมแบบ PD.....	9
2.8 แสดงตัวอย่างผลตอบสนองของการทำงานของระบบควบคุมแบบ PID.....	9
2.9 แสดงค่าพ่วงเกินที่ต้องการเมื่อใช้การปรับด้วยวิธี Ziegler-Nichols.....	10
2.10 แสดงผลตอบสนองรูปตัว S เมื่อปรับโดยใช้วิธี Process Reaction Curve.....	11
2.11 แสดงผลตอบสนองเวลาเกิดการแกว่งอย่างต่อเนื่อง เมื่อปรับโดยใช้วิธี Ultimate method.....	12
2.12 แสดงค่าอัตราการเสื่อม ¼ เมื่อปรับโดยวิธี Damped Oscillation.....	13
2.13 แสดงค่า quarter-amplitude decay (Decay Ratio ~ 0.25).....	14
2.14 แสดงรูปร่าง PLC ชนิด Block Type.....	15
2.15 แสดงโครงสร้างภายนอก ของ PLC.....	15
2.16 แสดงหน่วยขยายอินพุต/เอาต์พุต (Expansion I/O Units).....	16
2.17 แสดงรูปร่างของ PLC ชนิดโมดูล.....	17
2.18 แสดงชนิดของ PLC ชนิดโมดูล ที่ใช้คอนเนคเตอร์ในการเชื่อมต่อ.....	17
2.19 แสดงชนิดของ PLC ชนิดโมดูล ที่ใช้ Backplane ในการเชื่อมต่อ.....	18
2.20 แสดงตัวป้อนโปรแกรมแบบมือถือ (Programming Console).....	21
2.21 แสดงวิธีการต่อใช้งานคอมพิวเตอร์กับ PLC.....	22
2.22 ตัวอย่างซอฟต์แวร์ (CX-Programmer).....	22
2.23 ระบบการติดต่อสื่อสารของ PLC ในโรงงานอุตสาหกรรม (PLC Network).....	23
2.24 ไดอะแกรมภายใน PLC.....	24
2.25 แสดงอุปกรณ์อินพุตต่างๆ.....	26
2.26 โครงสร้างด้านซอฟต์แวร์ (Software Architecture).....	27
2.27 โครงสร้างด้านซอฟต์แวร์ของระบบ SCADA.....	28
2.28 การเข้าถึงข้อมูลของโปรแกรมการใช้งาน.....	33
2.29 การเข้าถึงข้อมูลจากอุปกรณ์ผ่าน Driver.....	34
2.30 การใช้งานโปรแกรมแบบพิเศษต่างๆบนระบบควบคุม.....	34
3.1 ตัวแปรอินพุตของกระบวนการ.....	36
3.2 การเชื่อมต่อสัญญาณระหว่าง Temperature Simulator กับเครื่องควบคุม PLC.....	37

## สารบัญรูป(ต่อ)

รูปที่	หน้า
3.3 แสดงโปรแกรม Ladder.....	37
3.4 แสดง Min และ Max ข้อมูลต้นทางและปลายทางคำสั่ง SCL( ).....	38
3.5 แสดงกระบวนการของตัวแปรเอาร์ทพุท.....	39
3.6 แสดงการต่อสัญญาณควบคุมจากแหล่งจ่ายไฟตรงจากภายนอก.....	40
3.7 แสดงการต่อสัญญาณควบคุมจาก D/A Analog Output.....	41
3.8 แสดงลำดับและบล็อกคำสั่งการเขียนโปรแกรมกำหนด SV.....	41
3.9 แสดงโปรแกรม Ladder.....	42
3.10 แสดงกระบวนการสังเกตผลตอบสนองแบบสแต็ป.....	44
3.11 การเชื่อมต่อสัญญาณระหว่าง Temperature Simulator กับเครื่องควบคุม PLC.....	45
3.12 แสดงลำดับและบล็อกคำสั่งการเขียนโปรแกรมอ่านค่า PV.....	45
3.13 แสดงโปรแกรม Ladder.....	46
3.14 แสดงหน้าโปรแกรม ProLab.....	49
3.15 แสดงการ Configuration / Port setting.....	49
3.16 แสดงตำแหน่งของ Data Memory ที่ใช้ในการ Plot Graph.....	50
3.17 แสดงบล็อกไดอะแกรมและกระบวนการควบคุมแบบวงรอบปิด.....	51
3.18 แสดงลำดับและบล็อกคำสั่งการเขียนโปรแกรมกำหนด SV.....	51
3.19 แสดงลำดับและบล็อกคำสั่งการเขียนโปรแกรมอ่านค่า PV.....	51
3.20 แสดงลำดับและบล็อกคำสั่งการเขียนโปรแกรมให้ค่า MV.....	52
3.21 แสดงโปรแกรม Ladder.....	54
3.22 แสดงการเลือกชนิดของ Device driver.....	56
3.23 แสดงการเลือก Device model.....	56
3.24 แสดงการตั้งค่า Tag Properties.....	57
3.25 แสดงการดูค่าของ tag ในแอดเดรสที่เราต้องการอ่านหรือเขียนค่า.....	57
3.26 แสดงการตั้งค่า OPC connection.....	58
3.27 แสดงหน้ากราฟฟิคบน WinCC ที่ต้องการแสดงผล.....	58
4.1 แสดงแรงเคลื่อนไฟฟ้าต่ำสุด-สูงสุดที่วัดได้.....	60
4.2 แสดงการเปลี่ยนแปลงของอุณหภูมิเมื่อมีป้อนแรงดัน.....	62
4.3 แสดงความสัมพันธ์ระหว่างข้อมูลใน DM030 กับ แรงดันที่ป้อนให้ Heater.....	63
4.4 แสดงค่า PV ของระบบเมื่อยังไม่ผ่านตัวควบคุม PID.....	65
4.5 การควบคุม PID เมื่อค่า PB, Ti, Td เท่ากับ 800, 0, 0 แบบสแต็ปจาก 0%, 50%, 0%.....	66
4.6 การควบคุม PID เมื่อค่า PB, Ti, Td เท่ากับ 500, 0, 0 แบบสแต็ปจาก 0%, 50%, 0%.....	66
4.7 การควบคุม PID เมื่อค่า PB, Ti, Td เท่ากับ 200, 0, 0 แบบสแต็ปจาก 0%, 50%, 0%.....	67

## สารบัญรูป(ต่อ)

รูปที่	หน้า
4.8 การควบคุม PID เมื่อค่า Pb, Ti, Td เท่ากับ 100, 0, 0 แบบสลับจาก 0%, 50%, 0%.....	67
4.9 การควบคุม PID เมื่อค่า Pb, Ti, Td เท่ากับ 800, 0, 0 ตามลำดับ.....	68
4.10 การควบคุม PID เมื่อค่า Pb, Ti, Td เท่ากับ 500, 0, 0 ตามลำดับ.....	68
4.11 การควบคุม PID เมื่อค่า Pb, Ti, Td เท่ากับ 200, 0, 0 ตามลำดับ.....	69
4.12 การควบคุม PID เมื่อค่า Pb, Ti, Td เท่ากับ 100, 0, 0 ตามลำดับ.....	69
4.13 การควบคุม PID เมื่อค่า Pb, Ti, Td เท่ากับ 200, 0, 0 ตามลำดับ.....	70
4.14 การควบคุม PID เมื่อค่า Pb, Ti, Td เท่ากับ 200, 100, 0 ตามลำดับ.....	71
4.15 การควบคุม PID เมื่อค่า Pb, Ti, Td เท่ากับ 200, 200, 0 ตามลำดับ.....	71
4.16 การควบคุม PID เมื่อค่า Pb, Ti, Td เท่ากับ 200, 300, 0 ตามลำดับ.....	72
4.17 แสดงหน้ากราฟฟิค WinCC.....	73
4.18 กราฟแสดงความสัมพันธ์ระหว่างค่า SV กับ PV.....	73
4.19 กราฟแสดงแนวโน้มของค่า MV.....	74

# บทที่ 1

## บทนำ

### 1.1 ความเป็นมาและแรงจูงใจในการทำปริญญานิพนธ์

เนื่องจากในปัจจุบันกระบวนการทางอุตสาหกรรมต่างๆ นิยมที่จะใช้ระบบควบคุมในการควบคุมกระบวนการ ดังนั้นจึงจำเป็นต้องใช้บุคลากรที่มีความรู้ ความสามารถในด้านของเครื่องมือวัดและการควบคุมชนิดต่างๆ เพื่อเข้าไปดูแลกระบวนการทางอุตสาหกรรมนั้นๆ เพื่อให้เกิดประสิทธิภาพสูงสุด โดยประสิทธิภาพสูงสุดคือการทำได้ผลผลิตเพิ่มขึ้นและมีต้นทุนในการผลิตลดต่ำลง ซึ่งแต่ละกระบวนการทางอุตสาหกรรมนั้นต้องมีหลายๆกระบวนการควบคุมที่ใช้ เช่นกระบวนการควบคุมอุณหภูมิ กระบวนการควบคุมระดับ กระบวนการควบคุมแรงดัน เป็นต้น จึงเห็นว่ากระบวนการควบคุมมีความสำคัญต่อกระบวนการทางอุตสาหกรรมอย่างมาก ประกอบกับการที่อุณหภูมิเป็นสิ่งที่ใกล้ตัวคนเรามาก เพียงแต่เราอาจไม่ได้รู้สึกถึงความสำคัญ อันเนื่องมาจากความคุ้นเคย ความต้องการให้อุณหภูมิคงที่ ที่ค่าใดค่าหนึ่งนั้น เป็นความต้องการพื้นฐานมากในชีวิตประจำวัน โดยตัวอย่างที่ใกล้ตัวเราที่สุดที่เราเห็นได้นั้นคือ ร่างกายคนเรามีอุณหภูมิประมาณ 37 องศาเซลเซียส ถ้าอุณหภูมิภายนอกร้อนเกินไปร่างกายก็จะมีวิธีการควบคุมอุณหภูมิ เช่นการขับเหงื่อเพื่อระบายความร้อน ส่งสัญญาณไปกระตุ้นร่างกายให้อยากดื่มน้ำเย็นๆเพิ่ม ซึ่งในทางวิศวกรรมแล้ว การควบคุมอุณหภูมิเป็นเรื่องที่สำคัญมาก ทั้งในอุตสาหกรรม ในกระบวนการผลิตต่างๆมากมาย เช่น กระบวนการผลิตเหล็ก กระบวนการผลิตทางเคมี กระบวนการผลิตกระเบื้องเซรามิก กระบวนการปิโตรเคมี และยังมีมีการใช้การควบคุมอุณหภูมิในเครื่องมือและอุปกรณ์ต่างๆ ในชีวิตประจำวันอีกด้วย เช่น เตอบนขม ตู้แช่แข็ง หรือแม้กระทั่งเครื่องทำน้ำอุ่น

ดังนั้นจึงได้เลือกศึกษากระบวนการควบคุมอุณหภูมิในการทำการศึกษาเพื่อฝึกปฏิบัติการในการควบคุมอุณหภูมิโดยใช้หลักการควบคุมแบบพีไอดี และใช้พีแอลซีในการควบคุม และใช้แบบจำลองอุณหภูมิเป็นตัวทดลอง รวมทั้งศึกษาการทำให้ค่าที่ได้จากแบบจำลองแสดงผลแบบทันที สามารถดูค่าได้ผ่านทางระบบ Scada เพื่อเตรียมพร้อมสำหรับการเข้าไปเป็นบุคลากรที่มีความรู้ความสามารถในการควบคุมกระบวนการ เพราะในแต่ละกระบวนการควบคุมนั้นจะมีหลักการที่คล้ายคลึงกัน จึงสามารถนำไปประยุกต์ใช้ในกระบวนการที่แตกต่างกันได้

## 1.2 วัตถุประสงค์ของปริญญานิพนธ์

1. ศึกษาวิธีการต่อเครื่องควบคุม PLC , Temperature Simulator ,Computer และ OPC Server
2. เพื่อศึกษาวิธีการใช้งาน PLC SYSMAC C200HS OMRON และ โปรแกรม CX –Programmer
3. เพื่อศึกษาการควบคุมและฟังก์ชันการทำงานของ PID
4. เพื่อศึกษาคุณลักษณะของ Temperature Simulator
5. เพื่อศึกษาการแสดงผลบน OPC Server

## 1.3 ขอบเขตของปริญญานิพนธ์

1. ศึกษาโครงสร้างของ PLC ยี่ห้อ Omron รุ่น C200HS
2. ศึกษาการใช้งาน CX – Programmer
3. ศึกษาการทำงานของ Temperature Simulator
4. ศึกษาหลักการควบคุมอุณหภูมิโดยใช้หลักการควบคุมแบบ PID
5. ศึกษาการใช้ PLC เพื่อควบคุมอุณหภูมิของ Temperature Simulator
6. ศึกษาวิธีแสดงผลการควบคุมผ่าน OPC Server
7. ทดสอบกระบวนการควบคุม

## 1.4 ขั้นตอนการศึกษา

1. ศึกษาโครงสร้างของ PLC และ Temperature Simulator
2. ศึกษาการใช้งาน PLC ยี่ห้อ Omron รุ่น C200HS
3. ศึกษา Characteristic ของ Temperature Simulator
4. ศึกษาวิธีการเชื่อมต่อระหว่าง PLC , Temperature Simulator และ Computer
5. เขียนคำสั่งลงใน PLC
6. ทดลองควบคุมอุณหภูมิโดยใช้ PLC
7. ศึกษาวิธีเชื่อมต่อ OPC Server
8. ทดลองแสดงผลโดยใช้ OPC Server

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทที่ 2

# ทฤษฎีและหลักการที่เกี่ยวข้อง

### 2.1 กล่าวนำ

การฝึกการควบคุมอุณหภูมิของ Temperature Simulator เราใช้ PLC ในการควบคุมอุณหภูมิ และใช้หลักการควบคุมอุณหภูมิแบบ PID สำหรับโปรแกรมที่ใช้ในการเขียนคำสั่งลงบน PLC คือโปรแกรม CX – Programmer หลังจากนั้นได้นำค่าจาก Temperature Simulator ไปแสดงผลบน OPC Server

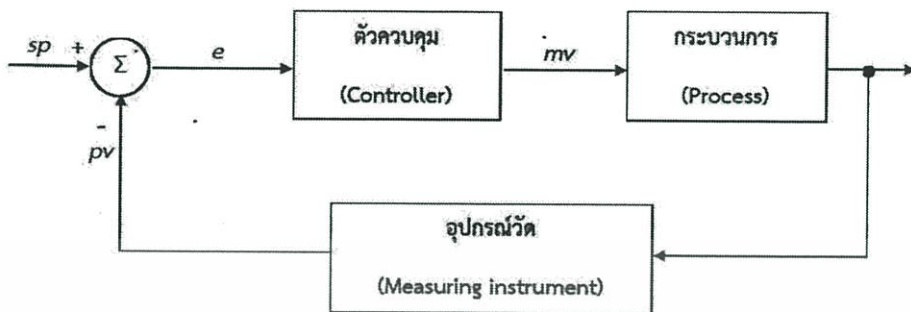
### 2.2 การควบคุมแบบป้อนกลับด้วยตัวควบคุม PID

ระบบควบคุมแบบป้อนกลับ หมายถึง ระบบควบคุมที่ต้องการควบคุมให้เอาต์พุตหรือผลตอบสนองของระบบมีค่าตามที่ต้องการ โดยการนำเอาสัญญาณเอาต์พุตป้อนกลับมาเปรียบเทียบกับสัญญาณอินพุต ซึ่งการป้อนกลับนี้จะต้องเป็นการป้อนกลับแบบลบ (Negative Feedback) จากนั้นนำค่าความแตกต่างระหว่างอินพุตกับเอาต์พุต ส่งต่อไปยังส่วนสร้างสัญญาณควบคุมซึ่งจะทำการสร้างสัญญาณควบคุมที่เหมาะสมตามกฎเกณฑ์การควบคุมที่เลือกไว้และส่งสัญญาณออกไปควบคุมระบบเพื่อให้เอาต์พุตของกระบวนการที่ถูกควบคุมมีค่าตามต้องการ ระบบควบคุมป้อนกลับนี้อาจจะเรียกอีกอย่างหนึ่งว่า ระบบควบคุมแบบลูปปิดหรือระบบอัตโนมัติก็ได้

ระบบควบคุมแบบป้อนกลับโดยทั่วไปประกอบด้วยอุปกรณ์ 3 ส่วน คือ

1. **ตัวควบคุม (Controller)** หมายถึงเครื่องมือหรืออุปกรณ์ที่ใช้ในการสร้างสัญญาณควบคุมเพื่อทำหน้าที่ควบคุมให้ระบบหรือกระบวนการที่ต้องการควบคุม มีเอาต์พุตหรือผลตอบสนองตามความต้องการ ตัวควบคุมจะมีหลายแบบ เช่น ตัวควบคุมแบบ ON – OFF ตัวควบคุมแบบ Proportional (P) ตัวควบคุมแบบ Integral (I) ตัวควบคุมแบบ Derivative (D) หรือการใช้ตัวควบคุมหลายๆ แบบรวมกัน เช่น ตัวควบคุมแบบ PI ตัวควบคุมแบบ PD และตัวควบคุมแบบ PID เป็นต้น ซึ่งสัญญาณที่ออกจากตัวควบคุมจะเรียกว่าตัวแปรปรับกระบวนการ (Manipulated Variable: mv)
2. **กระบวนการ (Plant or Process)** หมายถึงระบบหรือกระบวนการที่ถูกควบคุม หรืออาจจะเป็นวัตถุทางกายภาพที่ถูกควบคุมก็ได้ เช่น กระบวนการเกี่ยวกับการควบคุมอุณหภูมิ กระบวนการเกี่ยวกับการควบคุมความดัน เป็นต้น กระบวนการนี้รวมถึงอุปกรณ์ควบคุมสุดท้าย (Final Control Element) ซึ่งสถานะของกระบวนการเรียกว่าตัวแปรกระบวนการ (Process Variable: pv)
3. **อุปกรณ์วัด (Measuring Instruments)** หมายถึงอุปกรณ์ที่อาจจะได้แก่ เซนเซอร์ (Sensor) ทรานสดิวเซอร์ (Transducer) อุปกรณ์แปลง (Transmitter) หรือวัดสัญญาณอื่นๆ ที่ทำหน้าที่วัดค่าของเอาต์พุตของระบบที่ถูกควบคุมแล้วป้อนกลับไปเพื่อทำการเปรียบเทียบกับอินพุตของระบบ โดยสัญญาณที่ใช้เป็นสัญญาณมาตรฐานทางอุตสาหกรรม ได้แก่ สัญญาณไฟฟ้ากระแสตรง 4 – 20 มิลลิแอมป์แอมป์ ( 4 – 20 mA ) สัญญาณแรงดันไฟฟ้า

กระแสดร่ง 1-5 โวลต์ (1-5 VDC) หรือ สัญญาณลมขนาด 3 – 15 ปอนด์/ตารางนิ้ว ( 3-15 PSI ) เป็นต้น



รูปที่ 2.1 แสดงโครงสร้างของระบบควบคุมแบบป้อนกลับโดยทั่วไป

การควบคุมกระบวนการทางอุตสาหกรรมโดยทั่วไปนิยมใช้ตัวควบคุมแบบพีไอดี (PID) เพราะรูปแบบของตัวควบคุมเป็นตัวควบคุมที่สามารถควบคุมกระบวนการต่างๆ ได้อย่างกว้างขวางเนื่องจากมีโครงสร้างการทำงานที่ไม่ซับซ้อน สามารถเข้าใจได้ง่าย การใช้งานตัวควบคุมพีไอดีขึ้นอยู่กับค่าปรับค่าพารามิเตอร์ของตัวควบคุม PID ให้เหมาะสมเพื่อให้ได้ผลตอบสนองของกระบวนการตามความต้องการ

ตัวควบคุมแบบ PID ประกอบด้วยตัวควบคุมแบบ Proportional (P) ตัวควบคุมแบบ Integral (I) ตัวควบคุมแบบ Derivative (D) ฟังก์ชันถ่ายโอน (Transfer function) ดังนี้

$$m(t) = K_p \left( e(t) + \frac{1}{T_i} \int_0^t e(t) dt + T_d \frac{de(t)}{dt} \right) \quad (2.1)$$

โดย  $K_p$  = ค่าอัตราขยายของตัวควบคุมแบบ P (Proportional Gain)

$T_i$  = ค่าเวลา Integral (Integral Time)

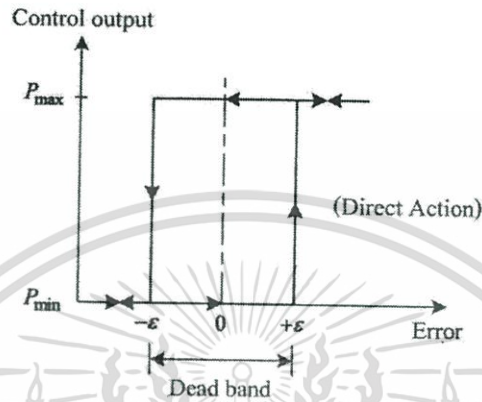
$T_d$  = ค่าเวลา Derivative (Derivative Time)

จากรูปที่ 2.1 จะเห็นได้ว่าสัญญาณควบคุมหรือตัวแปรปรับกระบวนการ (mv) ที่ได้จากตัวควบคุม PID จะถูกกำหนดด้วยความสัมพันธ์ระหว่างสัญญาณขาเข้าตัวควบคุมกับตัวแปรกระบวนการ (pv) กับสัญญาณอ้างอิงหรือค่าเป้าหมาย (sp) โดยที่ความสัมพันธ์ดังกล่าวจะขึ้นอยู่กับกฎเกณฑ์การควบคุมที่ผู้ควบคุมปรับแต่งไว้ล่วงหน้า (ค่าพารามิเตอร์ของตัวควบคุม PID) ซึ่งสัญญาณควบคุมของตัวควบคุมจะเป็นไปตามวิธีการควบคุมแบบต่างๆดังที่จะกล่าวต่อไป

## 2.3 การยการควบคุมของการควบคุมแบบป้อนกลับ

### 2.3.1 การยการควบคุมแบบ ON-OFF

การควบคุมแบบ ON-OFF เป็นการควบคุมแบบที่ง่ายที่สุดและนิยมใช้ในการควบคุมกระบวนการที่ไม่ต้องการความเที่ยงตรงสูง โดยการควบคุมจะทำงานเพียง 2 สถานะ คือ เปิด (100%) กับปิด (0%) เท่านั้น การยการควบคุมแบบ ON-OFF ดังรูปที่ 2.2



รูปที่ 2.2 แสดงการยการควบคุมแบบ ON-OFF

จากรูปที่ 2.2 จะเห็นว่าถ้าค่าความคลาดเคลื่อนมากกว่าค่าวิกฤต (+ $\epsilon$ ) ค่าเอาต์พุตของตัวควบคุมจะเปลี่ยนจาก 0% เป็น 100% เมื่อค่าความคลาดเคลื่อนน้อยกว่าค่าวิกฤต (- $\epsilon$ ) ค่าเอาต์พุตของตัวควบคุมจะเปลี่ยนจาก 100% เป็น 0% ค่าเอาต์พุตที่อยู่ในช่วงเดธแบนด์ (Dead Band) จะไม่มีการเปลี่ยนแปลงแต่อย่างใดซึ่งอาจจะเป็นผลมาจากการเสียดทานที่ไม่ได้คาดไว้ก่อนหรือบางครั้งก็จำเป็นต้องทำให้เกิดช่วงเดธแบนด์ขึ้นเพื่อป้องกันการ ON - OFF บ่อยเกินไปอันจะทำให้อุปกรณ์ควบคุมหรือกระบวนการได้รับความเสียหาย แต่ช่วงเดธแบนด์นี้ต้องไม่กว้างนักเพราะจะทำให้ค่าความเที่ยงตรงของการควบคุมลดลง การยการควบคุมแบบ ON-OFF สามารถเขียนเป็นสมการได้ดังนี้

$$m(t) = \begin{cases} 0\% & , e < -\epsilon \\ 100\% & , e > +\epsilon \end{cases} \quad (2.2)$$

เมื่อ  $m(t)$  = สัญญาณควบคุมหรือเอาต์พุตของตัวควบคุม

$e(t)$  = ค่าความคลาดเคลื่อน

$\epsilon$  =  $\frac{1}{2}$  ของค่าเดธแบนด์

### 2.3.2 การยการควบคุมแบบ Proportional(P)

การยการควบคุมแบบ P นั้น ค่าเอาต์พุตของตัวควบคุมจะแปรผันตรงกับค่าความคลาดเคลื่อน คือ ถ้าค่าความคลาดเคลื่อนมีค่ามากขึ้น ค่าเอาต์พุตของตัวควบคุมก็จะมีค่ามากขึ้นตาม และถ้าค่าความคลาดเคลื่อนมีค่าน้อยลง ค่าเอาต์พุตของตัวควบคุมก็จะมีค่าน้อยลงตามการยการควบคุมแบบ P สามารถเขียนเป็นสมการได้ดังนี้

$$m_p(t) = K_p e(t) + \bar{m} \quad (2.3)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

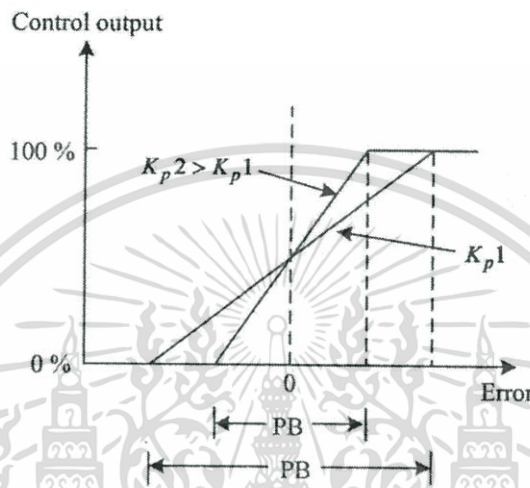
เมื่อ  $m_p(t)$  = ค่าเอาต์พุทของตัวควบคุมแบบ Proportional

$K_p$  = อัตราขยายของตัวควบคุมแบบ Proportional

$\bar{m}$  = ค่าเอาต์พุทของตัวควบคุมที่ค่าความคลาดเคลื่อนเท่ากับศูนย์

ตัวควบคุมแบบ P บางตัวอาจจะใช้ค่า Proportional Band (PB) แทนการใช้ค่า  $K_p$  ซึ่ง PB คือ ช่วงของค่าความคลาดเคลื่อนระหว่างที่เอาต์พุทของตัวควบคุมมีค่า 0-100% ดังสมการ (2.4)

$$PB = \frac{100\%}{K_p} \quad (2.4)$$



รูปที่ 2.3 แสดงคุณสมบัติของกิริยาการควบคุมแบบ Proportional

ข้อเสีย ของกิริยาการควบคุมแบบ Proportional คือ ไม่สามารถกำจัดค่าออฟเซ็ทได้

### 2.3.3 กิริยาการควบคุมแบบ Integral(I)

กิริยาการควบคุมแบบ I หรือสามารถเรียกอีกอย่างหนึ่งว่า การควบคุมแบบรีเซ็ต (Reset Control) โดยค่าเอาต์พุทของตัวควบคุมหาได้จากการหาค่าพื้นที่ทั้งหมดภายในกราฟของค่าความคลาดเคลื่อนต่อเวลาคูณกับค่าคงที่ที่เรียกว่า อัตราขยายของตัวควบคุมแบบ I (Integral Gain) ดังสมการต่อไปนี้

$$m_I(t) = K_I \int_0^t e(t) dt + \bar{m}_I(0) \quad (2.5)$$

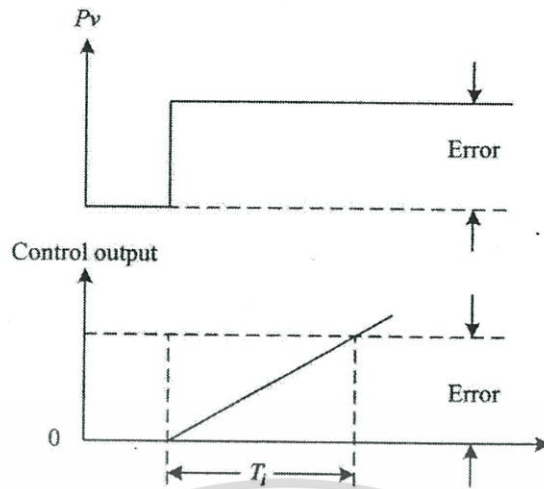
เมื่อ  $m_I(t)$  = ค่าเอาต์พุทของตัวควบคุมแบบ Integral

$K_I$  = อัตราขยายของตัวควบคุมแบบ Integral

$\int_0^t e(t) dt$  = พื้นที่ทั้งหมดของค่าความคลาดเคลื่อน

$\bar{m}_I(0)$  = ค่าเอาต์พุทของตัวควบคุมที่เวลา  $t$  เท่ากับศูนย์

ผลของกิริยาการควบคุมแบบ I นี้จะทำให้ไม่เกิดออฟเซ็ทขึ้นในระบบและช่วยลดค่าพุ่งเกิน (Overshoot) ของระบบที่เกิดขึ้นลงได้ แต่ถ้ากิริยาการควบคุมมีค่าสูงเกินไปจะทำให้ผลตอบสนองของกระบวนการช้าลง ผลตอบสนองของกิริยาการควบคุมแบบ I แสดงดังรูปที่ 2.4



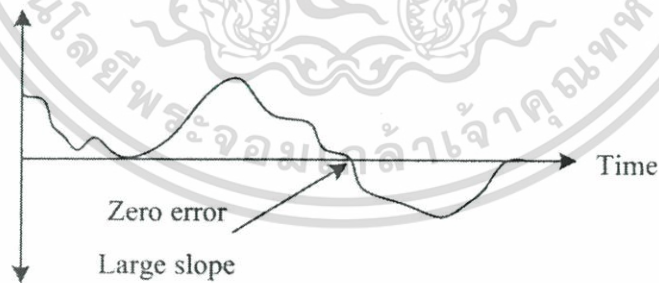
รูปที่ 2.4 แสดงผลตอบสนองของกริยาการควบคุมแบบ Integral

### 2.3.4 กริยาการควบคุมแบบ Derivative (D)

กริยาการควบคุมแบบ D เรียกอีกชื่อหนึ่งว่าการควบคุมแบบอัตราส่วน (Rate Action) โดยสัญญาณเอาต์พุตของตัวควบคุมจะขึ้นอยู่กับอัตราการเปลี่ยนแปลงของค่าความคลาดเคลื่อนต่อเวลา จะเห็นว่าค่าความคลาดเคลื่อนนี้มีโอกาสเป็นศูนย์ได้ และค่าเอาต์พุตก็สามารถเปลี่ยนแปลงให้มีค่าสูงขึ้น เมื่อความคลาดเคลื่อนเปลี่ยนแปลง ซึ่งเรียกการกระทำดังกล่าวว่า อัตราการกระทำ (Rate Action) ดังสมการต่อไปนี้

$$m_D(t) = K_D \frac{de(t)}{dt} \quad (2.6)$$

เมื่อ  $m_D(t)$  = ค่าเอาต์พุตของตัวควบคุมแบบ Derivative  
 $K_D$  = อัตราขยายของตัวควบคุมแบบ Derivative  
 $\frac{de(t)}{dt}$  = ค่าความคลาดเคลื่อนที่เวลา t



รูปที่ 2.5 แสดงตัวอย่างผลตอบสนองของกริยาการควบคุมแบบ Derivative

### 2.3.5 กริยาการควบคุมแบบ Proportional-Integral (PI)

ตามที่กล่าวมาแล้วว่ากริยาการควบคุมแบบ P นั้น จะมีออฟเซ็ทเกิดขึ้น ซึ่งการกำจัดค่าออฟเซ็ทนี้สามารถทำได้โดยการเพิ่มกริยาการควบคุมแบบ I เข้าไป ดังนั้นสมการเอาต์พุตของตัวควบคุมแบบ PI นี้ ดังสมการ

$$m_{PI}(t) = \bar{m} + K_p e(t) + K_p K_I \int_0^t e(t) dt \quad (2.7)$$

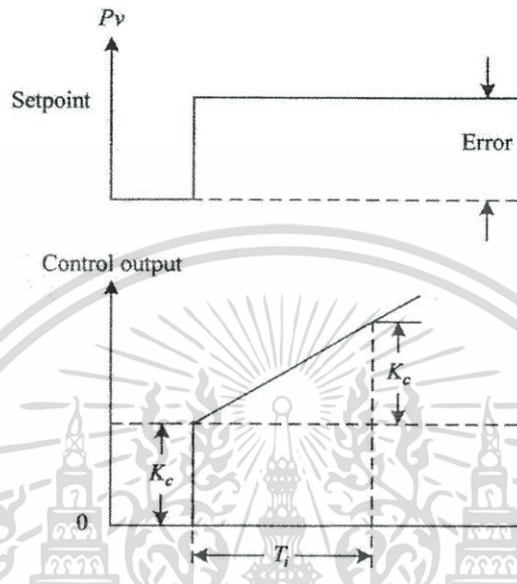
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หรือ

$$m_{PI}(t) = \bar{m} + K_c e(t) + \frac{K_c}{T_i} \int_0^t e(t) dt \quad (2.8)$$

เมื่อ  $K_c = K_P$  และ  $K_I = 1/T_i$

$T_i$  = ค่าเวลา Integral



รูปที่ 2.6 แสดงตัวอย่างผลตอบสนองของกริยาการควบคุมแบบ PI (Direct action)

### 2.3.6 กริยาการควบคุมแบบ Proportional-Derivative (PD)

การประยุกต์ใช้กริยาการควบคุมแบบ P ร่วมกับกริยาการควบคุมแบบ D เพื่อทำให้ผลตอบสนองของระบบรวดเร็วขึ้นแต่จะไม่มีผลโดยตรงต่อผลตอบสนองของระบบที่สภาวะคงที่ ซึ่งสมการเอาท์พุทของกริยาการควบคุมแบบ PD แสดงดังสมการต่อไปนี้

$$m_{PD}(t) = \bar{m} + K_P e(t) + K_P K_D \frac{de(t)}{dt} \quad (2.9)$$

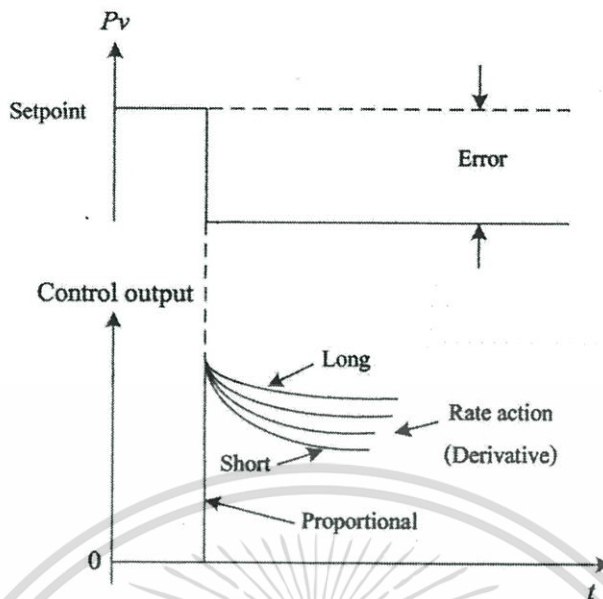
หรือ

$$m_{PD}(t) = \bar{m} + K_c e(t) + K_c T_d \frac{de(t)}{dt} \quad (2.10)$$

เมื่อ

$$K_D = T_d$$

$T_d$  = ค่าเวลา Derivative



รูปที่ 2.7 แสดงตัวอย่างผลตอบสนองของกริยาการควบคุมแบบ PD

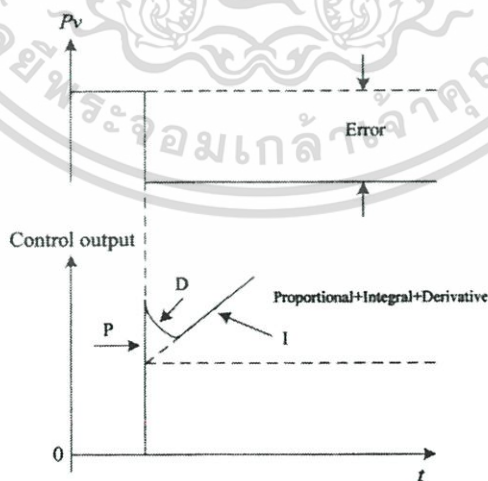
### 2.3.7 กริยาการควบคุมแบบ Proportional-Integral-Derivative (PID)

จากที่ได้กล่าวมาแล้ว เพื่อให้ผลตอบสนองของระบบควบคุมมีสมรรถนะเป็นไปตามต้องการจึงใช้กริยาการควบคุมทั้ง 3 แบบร่วมกัน ซึ่งจะทำให้ได้กริยาการควบคุมแบบ PID ที่มีสมการดังต่อไปนี้

$$m_{PID}(t) = \bar{m} + K_p e(t) + K_p K_I \int_0^t e(t) dt + K_p K_D \frac{de(t)}{dt} \quad (2.11)$$

หรือ

$$m_{PID}(t) = \bar{m} + K_C e(t) + \frac{K_C}{T_i} \int_0^t e(t) dt + K_C T_d \frac{de(t)}{dt} \quad (2.12)$$



รูปที่ 2.8 แสดงตัวอย่างผลตอบสนองของกริยาการควบคุมแบบ PID

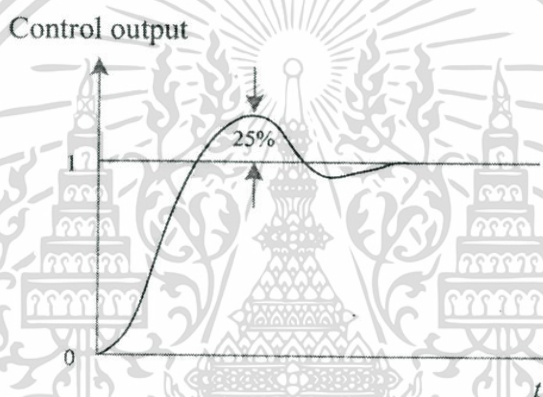
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 2.4 การปรับค่าพารามิเตอร์ของตัวควบคุมพีไอดี

การนำตัวควบคุมพีไอดีไปใช้ในการควบคุมกระบวนการต่างๆ จำเป็นต้องปรับค่าพารามิเตอร์ของตัวควบคุมพีไอดีให้เหมาะสม เพื่อให้ได้ผลตอบสนองที่ดีตามความต้องการ ซึ่งการหาค่าพารามิเตอร์ของตัวควบคุมพีไอดีนี้สามารถทำได้หลายวิธี บางวิธีพิจารณาจากผลตอบสนองของกระบวนการจากการทดสอบกระบวนการและอาศัยประสบการณ์ บางวิธีจะอาศัยทฤษฎีทางคณิตศาสตร์ และส่วนใหญ่จะพิจารณาจากเงื่อนไขในโดเมนเวลามากกว่าในโดเมนความถี่ ซึ่งวิธีของ Ziegler-Nichols เป็นวิธีที่ได้รับความนิยมมากที่สุดในทางปฏิบัติ

### 2.4.1 การปรับค่าพารามิเตอร์ของตัวควบคุมพีไอดี โดยวิธีของ Ziegler-Nichols

การหาค่า  $K_p, T_i$  และ  $T_d$  จะขึ้นอยู่กับคุณลักษณะของผลตอบสนองชั่วครู่ของกระบวนการที่ถูกควบคุม ซึ่งมีอยู่ 2 วิธี แต่ละวิธีมีจุดมุ่งหมายที่จะทำให้ผลตอบสนองต่อเวลาของกระบวนการต่ออินพุตแบบขั้นบันได มีค่าพุ่งเกินสูงสุดไม่เกิน 25% ดังรูปที่ 2.9



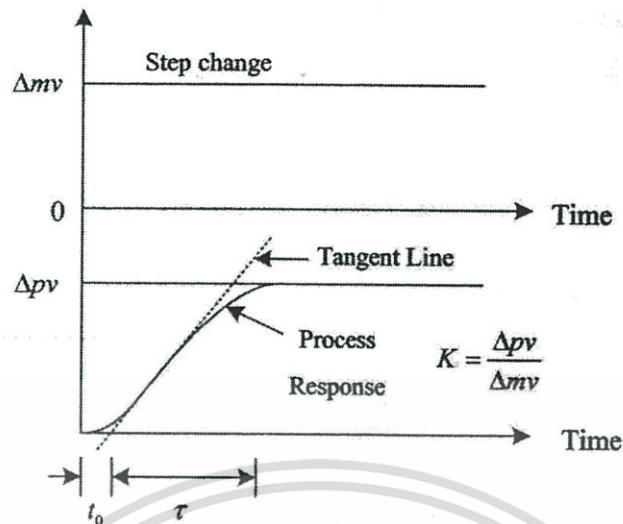
รูปที่ 2.9 แสดงค่าพุ่งเกินที่ต้องการเมื่อใช้การปรับด้วยวิธี Ziegler-Nichols

#### 2.4.1.1 วิธี Process Reaction Curve

วิธีนี้จะหาค่าพารามิเตอร์ของตัวควบคุมพีไอดีจากผลตอบสนองการของกระบวนการแบบลูเปิดที่ถูกควบคุมด้วยสัญญาณอินพุตแบบขั้นบันได โดยที่กระบวนการในกรณีนี้ จะไม่มี Pole ที่จุด Origin หรือไม่มี Dominant complex-conjugate poles ดังนั้นผลตอบสนองของเวลาจะได้ในลักษณะเป็นรูปตัว S (ถ้าผลตอบสนองของเวลาไม่มีลักษณะเป็นรูปตัว S วิธีนี้จะใช้ไม่ได้) จากผลตอบสนองต่อเวลาสามารถแยกพิจารณาเป็นค่าเวลาหน่วงของกระบวนการ (Delay time :  $t_0$ ) และเวลาคงที่ของกระบวนการ (Time constant :  $T$ ) โดยลากเส้นสัมผัสกับจุดที่มีอัตราการเปลี่ยนแปลงสูงสุด เส้นนี้จะตัดกับแกนเวลาและเส้นตรงของเอาท์พุท  $C(t)$  ที่  $C(t) = K$

ดังนั้นฟังก์ชันถ่ายโอนของระบบจะถูกประมาณได้ด้วยระบบอันดับหนึ่งแบบมีการหน่วงเวลา (FOPDT) คือ

$$G(s) = \frac{C(s)}{U(s)} = \frac{K e^{-t_0 s}}{\tau s + 1} \quad (2.13)$$



รูปที่ 2.10 แสดงผลตอบสนองรูปตัว S เมื่อปรับโดยใช้วิธี Process Reaction Curve

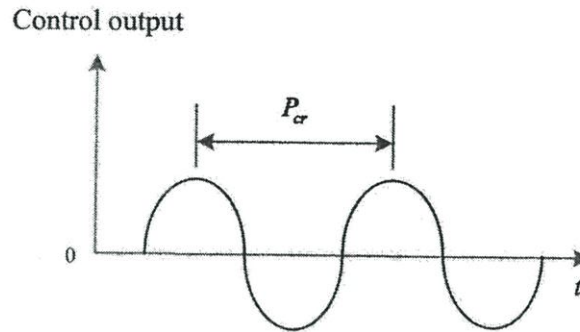
Ziegler-Nichols ได้กำหนดค่าของ  $K_p$ ,  $T_i$  และ  $T_d$  สำหรับตัวควบคุมแบบต่างๆ ดังตาราง 2.1

ตารางที่ 2.1 แสดงค่าพารามิเตอร์ของตัวควบคุมแบบต่างๆตามวิธี Process Reaction Curve

Controller Type		Proportional Gain $K_C$	Integral Time $T_i$	Derivative Time $T_d$
Proportional only	P	$\frac{1}{K} \left( \frac{\tau}{t_0} \right)$	-	-
Proportional-Integral	PI	$\frac{0.9}{K} \left( \frac{\tau}{t_0} \right)$	$3.33t_0$	-
Proportional-Integral-Derivative	PID	$\frac{1.2}{K} \left( \frac{\tau}{t_0} \right)$	$2.0t_0$	$0.5t_0$

#### 2.4.1.2 วิธี Ultimate Method

วิธีนี้จะหาค่าพารามิเตอร์ของตัวควบคุมพีไอดีจากการสังเกตผลตอบสนองของกระบวนการแบบลูปปิดที่ถูกควบคุมด้วยตัวควบคุมแบบ P ต่อสัญญาณอินพุตแบบขั้นบันได โดยปรับค่า  $K_p$  ไปเรื่อยๆ จนผลตอบสนองของกระบวนการเกิดการแกว่งอย่างต่อเนื่อง (Sustained Oscillations) ถ้าผลตอบสนองเวลาไม่เกิดการแกว่งอย่างต่อเนื่อง วิธีการนี้จะใช้ไม่ได้ แสดงดังรูปที่ 2.11



รูปที่ 2.11 แสดงผลตอบสนองเวลาเกิดการแกว่งอย่างต่อเนื่อง เมื่อปรับโดยใช้วิธี Ultimate method.

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

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

Ziegler-Nichols ได้กำหนดค่าของ  $K_p$ ,  $T_i$  และ  $T_d$  สำหรับตัวควบคุมแบบต่างๆดังตารางที่ 2.2

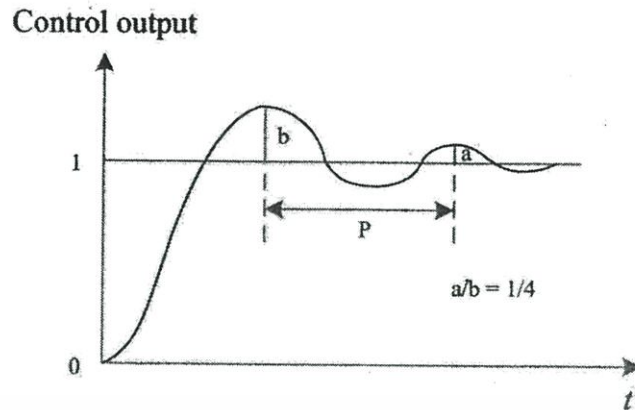
ตารางที่ 2.2 แสดงค่าพารามิเตอร์ของตัวควบคุมแบบต่างๆตามวิธี Ultimate Method

Controller Type		Proportional Gain $K_C$	Integral Time $T_i$	Derivative Time $T_d$
Proportional only	P	$K_{cr}/2$	-	-
Proportional-Integral	PI	$K_{cr}/2.2$	$T_{cr}/1.2$	-
Proportional-Integral-Derivative	PID	$K_{cr}/1.7$	$T_{cr}/2$	$T_{cr}/8$

การคำนวณหาค่าพารามิเตอร์ของตัวควบคุม PID โดยวิธีของ Ziegler-Nichols นั้น ไม่ใช่เป็นค่าที่เที่ยงตรงที่จะนำไปใช้งานได้ทันที และมีค่าพุ่งเกินสูงสุด 25% ตามที่กล่าวไว้แต่เป็นเพียงค่าที่ใกล้เคียงเท่านั้น ผู้ควบคุมจะต้องทำการปรับค่าพารามิเตอร์เหล่านี้แบบละเอียด (Fine Tuning) อีกครั้งหนึ่ง

#### 2.4.2 การปรับค่าพารามิเตอร์ของตัวควบคุมพีไอดี โดยวิธี Damped Oscillation

วิธีนี้ปรับปรุงมาจากวิธี Ultimate Method โดย Harriott เพื่อใช้ในกรณีที่การปรับค่าของ  $K_p$  ไปอย่างไรก็ตาม แต่ผลตอบสนองต่อเวลาไม่เกิดการแกว่งอย่างต่อเนื่อง แสดงดังรูปที่ 2.12



รูปที่ 2.12 แสดงค่าอัตราการเสื่อม ¼ เมื่อปรับโดยวิธี Damped Oscillation

วิธี Damped Oscillation นี้จะปรับค่าของ  $K_p$  ไปจนผลตอบสนองต่อเวลาของระบบควบคุมแบบลูปปิดมีอัตราเสื่อม ¼ ดังรูป จากนั้นวัดค่าของ  $P$  และใช้ค่าของ  $K_p(1/4)$  เพื่อคำนวณหาค่า  $K_p, T_i$  และ  $T_d$  ดังนี้

$$K_p = K_p(1/4)$$

$$T_i = P/1.5$$

$$T_d = P/6$$

#### 2.4.3 การปรับค่าพารามิเตอร์ของตัวควบคุมพีไอดี โดยวิธี Trial & Error Close-Loop Tuning

วิธีนี้คือการทดลองปรับ  $K_p, K_i$  เข้าคอนโทรล แล้วสังเกตค่าที่ทำให้ระบบสมดุล

##### ขั้นตอนการจูน

1. เริ่มพล็อตกราฟของ Process variable (PV)
2. เซ็ต  $K_i$  และ  $K_d$  เป็นศูนย์
3. เซ็ต  $K_p$  ค่าน้อยๆ
4. เปลี่ยน PID Controller ให้อยู่ในโหมด Auto
5. ทดสอบระบบโดยเปลี่ยนค่า Set point แล้วสังเกตว่า damping ของระบบ
6. ปรับค่า  $K_p$  เพิ่มขึ้นเรื่อยๆ จนระบบมี damping ตามต้องการ หรือ ให้เป็น quarter-amplitude decay (Decay Ratio  $\sim 0.25$ ) ดังรูป (b)

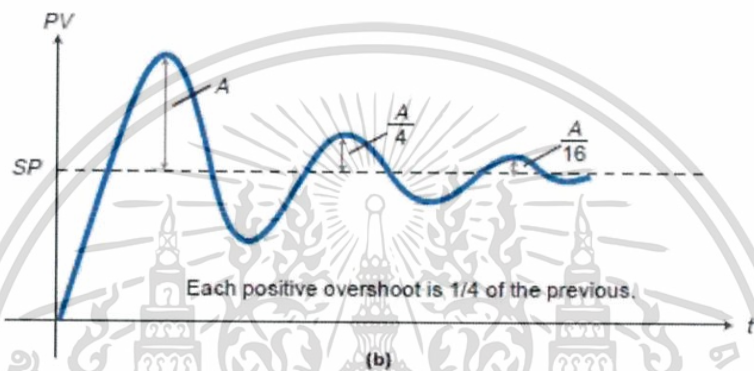
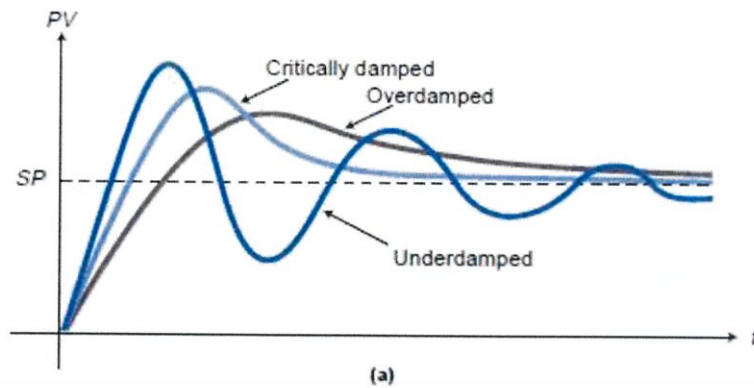


Figure 15-77: Process variable responses: (a) overdamped, critically damped, underdamped, and (b) quarter-amplitude decay.

รูปที่ 2.13 แสดงค่า quarter-amplitude decay (Decay Ratio  $\sim 0.25$ )

7. ถ้าระบบยังมี Offset ระหว่าง Set point และ Process variable ให้ปรับ  $K_i$  เพิ่มขึ้นจนไม่มี Offset

## 2.5 ความรู้พื้นฐานเกี่ยวกับพีแอลซี

PLC (Programmable Logic Controller) หรือปัจจุบันใช้คำว่า PC (Programmable Controller) ในที่นี้จะใช้คำว่า PLC แทน PC เพื่อป้องกันความสับสนกับคำว่า PC (Personal Computer)

PLC เป็นอุปกรณ์ที่คิดค้นขึ้นมา เพื่อใช้ควบคุมการทำงานของเครื่องจักรหรือระบบต่างๆ แทนวงจรรีเลย์แบบเก่า ซึ่งวงจรรีเลย์มีข้อเสียคือ การเดินสายและการเปลี่ยนแปลงเงื่อนไขในการควบคุมมีความยุ่งยาก และเมื่อใช้งานไปนานๆ หน้าสัมผัสของรีเลย์จะเสื่อม ทำให้ขาดเสถียรภาพในการควบคุม ดังนั้นปัจจุบัน PLC จึงเข้ามาทดแทนวงจรรีเลย์ เพราะ PLC ใช้งานได้ง่ายกว่า สามารถต่อเข้ากับอุปกรณ์อินพุต/เอาต์พุตได้โดยตรง นอกจากนั้นเพียงแค่เขียนโปรแกรมควบคุมก็สามารถใช้งานได้ทันที ถ้าต้องการจะเปลี่ยนเงื่อนไขใหม่สามารถทำได้โดยการเปลี่ยนแปลงโปรแกรมเท่านั้น นอกจากนี้ PLC ยังสามารถใช้งานร่วมกับอุปกรณ์อื่นๆ เช่น เครื่องอ่านบาร์โค้ด, เครื่องพิมพ์ (Printer) และระบบ RFID เป็นต้นในปัจจุบันนอกจาก PLC จะใช้งานแบบเดี่ยว (Stand alone) แล้ว ยังสามารถต่อ PLC หลายๆ ตัวเข้าด้วยกันเป็นเครือข่าย (Network) เพื่อควบคุมการทำงานของระบบให้มีประสิทธิภาพมากยิ่งขึ้นอีกด้วย จะเห็นได้ว่าการใช้งาน PLC มีความยืดหยุ่น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

มากกว่าการใช้งานวงจรรีเลย์แบบเก่าตั้งนั้นในปัจจุบันโรงงานอุตสาหกรรมต่างๆ จึงใช้ PLC เป็นหัวใจหลักในการควบคุมการทำงานของเครื่องจักรเราสามารถจำแนกประเภทของ PLC ตามลักษณะภายนอกได้เป็น 2 ชนิด คือ

### 2.5.1 ชนิดของ PLC

เราสามารถจำแนก PLC ตามโครงสร้างหรือลักษณะภายนอกได้เป็น 2 ชนิด คือ

#### 2.5.1.1 PLC ชนิดบล็อก (Block Type PLCs)

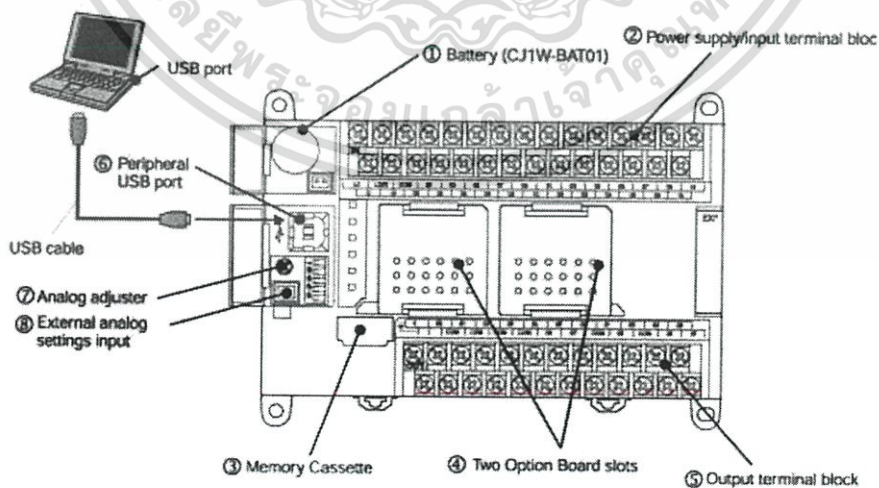
PLC ประเภทนี้ จะรวมส่วนประกอบทั้งหมดของ PLC อยู่ในบล็อกเดียวกัน ไม่ว่าจะเป็นตัวประมวลผล หน่วยความจำ ภาคอินพุต/เอาต์พุต และแหล่งจ่ายไฟ สามารถแสดงตัวอย่าง PLC แบบ Block Type ให้เห็นดังรูปที่ 2.14



รูปที่ 2.14 แสดงรูปร่าง PLC ชนิด Block Type

#### • ส่วนประกอบของ PLC แบบ Block Type

ในที่นี้จะยกตัวอย่าง PLC แบบ Block Type ของ OMRON รุ่น CP1L และ CP1H



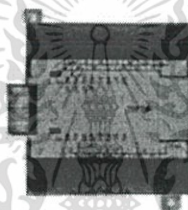
รูปที่ 2.15 โครงสร้างภายนอก ของ PLC

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 2.15 สามารถอธิบายความหมายของแต่ละส่วนได้ดังนี้

- ① คือ แบตเตอรี่ (Battery)
- ② คือ ขั้วต่อแหล่งไฟและอินพุต (Power Supply/Input Terminal)
- ③ คือ ช่องเสียบหน่วยความจำ (Memory Cassette)
- ④ คือ ช่องเสียบเพื่อเพิ่มพอร์ดติดต่อสื่อสาร (Option Board slots)
- ⑤ คือ ขั้วต่อเอาต์พุต (Output Terminal)
- ⑥ คือ พอร์ตเชื่อมต่อกับอุปกรณ์ป้อนโปรแกรม (USB Port)
- ⑦ คือ ปุ่มปรับอนาล็อก (Analog Adjuster)
- ⑧ คือ ขั้วต่ออินพุตสำหรับอนาล็อก setting (External analog setting input)
- ⑨ คือ พอร์ตขยายอินพุต/เอาต์พุต (Expansion I/O Unit Connector)

ในกรณีที่ท่านต้องการเพิ่มจำนวนอินพุต/เอาต์พุต สามารถใช้หน่วยขยายอินพุต/เอาต์พุต (Expansion I/O Units) เพื่อเพิ่มจำนวนอินพุต/เอาต์พุตได้โดยการต่อเข้ากับพอร์ตขยายอินพุต เอาต์พุต (Expansion I/O Unit Connector) สามารถแสดงโครงสร้างของหน่วยขยายอินพุต/เอาต์พุตได้ดังรูปที่ 2.16



รูปที่ 2.16 แสดงหน่วยขยายอินพุต/เอาต์พุต (Expansion I/O Units)

#### • ข้อดีและข้อเสียของ PLC แบบ Block Type

สามารถยกตัวอย่างข้อดีข้อเสียของ PLC แบบ Block Type ดังนี้

ข้อดี	ข้อเสีย
<ol style="list-style-type: none"> <li>1. มีขนาดเล็กสามารถติดตั้งได้ง่ายจึงเหมาะกับงานควบคุมขนาดเล็กๆ</li> <li>2. สามารถใช้งานแทนวงจรรีเลย์ได้</li> <li>3. มีฟังก์ชันพิเศษ เช่นฟังก์ชันทางคณิตศาสตร์และฟังก์ชันอื่น ๆ</li> <li>4. มีราคาถูกกว่าแบบแร็คหรือโมดูลในจำนวนอินพุต/เอาต์พุตที่เท่ากัน</li> </ol>	<ol style="list-style-type: none"> <li>1. การเพิ่มจำนวนอินพุต/เอาต์พุตสามารถเพิ่มได้น้อยกว่า PLC ชนิดโมดูล</li> <li>2. เมื่ออินพุต/เอาต์พุตเสียจุดใดจุดหนึ่งต้องนำ PLC ออกไปทั้งชุดทำให้ระบบต้องหยุดทำงานชั่วระยะเวลาหนึ่ง</li> <li>3. มีฟังก์ชันใหม่ให้เลือกใช้งานน้อยกว่า PLC ชนิดโมดูล</li> </ol>

เนื้อหาในหัวข้อต่อไปจะกล่าวถึง PLC อีกชนิดหนึ่งซึ่งแยกส่วนประกอบต่างๆ ออกจากกัน เรียกว่า PLC ชนิดโมดูล (Modular Type PLCs)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 2.5.1.2 PLC ชนิดโมดูล (Modular Type PLCs) หรือแร็ค (Rack Type PLCs)

PLC ชนิดนี้ส่วนประกอบแต่ละส่วนจะสามารถแยกออกจากกันได้เป็นแต่ละโมดูล (Modules) เช่นภาคอินพุต/เอาต์พุต จะอยู่ในส่วนของโมดูลอินพุต/เอาต์พุต (Input/Output Units) ซึ่งสามารถเลือกใช้งานได้ว่าจะใช้โมดูลขนาดกี่อินพุต/เอาต์พุต ซึ่งมีให้เลือกใช้งานหลายรูปแบบอาจจะใช้เป็นอินพุตอย่างเดียวขนาด 8 /16 จุด หรือเป็นเอาต์พุตอย่างเดียวขนาด 4/8/12/16 จุด ขึ้นอยู่กับรุ่นของ PLC ด้วย

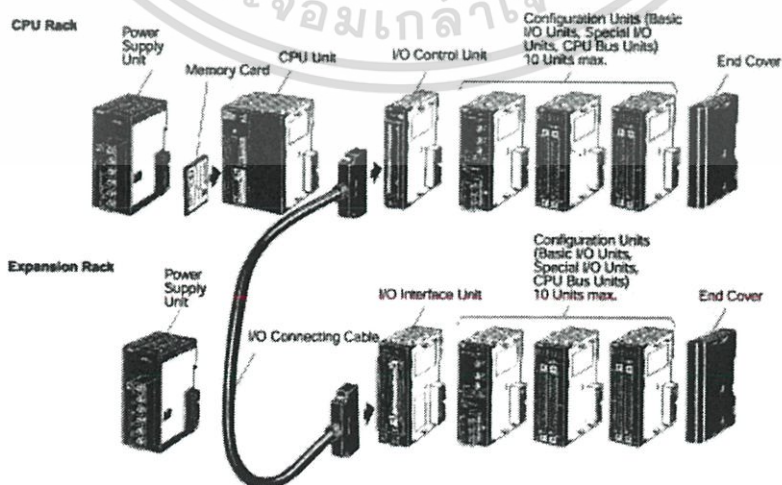
ในส่วนของตัวประมวลผลและหน่วยความจำจะรวมอยู่ในซีพียูโมดูล (CPU Unit) เราสามารถเปลี่ยนขนาดของ CPU Unit ให้เหมาะสมตามความต้องการใช้งาน เช่น PLC รุ่น CS1 จะมี CPU ให้เลือกใช้งานหลายรุ่นเช่น รุ่น CS1G-CPU42H จะมีความแตกต่างกับ PLC รุ่น CS1H-CPU65H (ทั้งสองรุ่นเป็น PLC ตระกูล CS1 เหมือนกัน) ตรงขนาดความจุของโปรแกรมและการรองรับจำนวนอินพุต/เอาต์พุต เป็นต้น

ส่วนประกอบต่างๆ ของ PLC ชนิดโมดูลที่กล่าวมาทั้งหมดนั้น เมื่อต้องการใช้งานจะถูกนำมาต่อรวมกัน บางรุ่นใช้เป็นคอนเนคเตอร์ในการเชื่อมต่อกันระหว่างยูนิต เช่น รุ่น CQM1 / CQM1H หรือ CJ1M/H/G แต่บางรุ่นใช้ Backplane ในการรวมยูนิตต่างๆ เข้าด้วยกัน เพื่อให้สามารถใช้งานร่วมกันได้สามารถยกตัวอย่าง PLC ชนิดโมดูลได้ดังแสดงรูปที่ 2.17



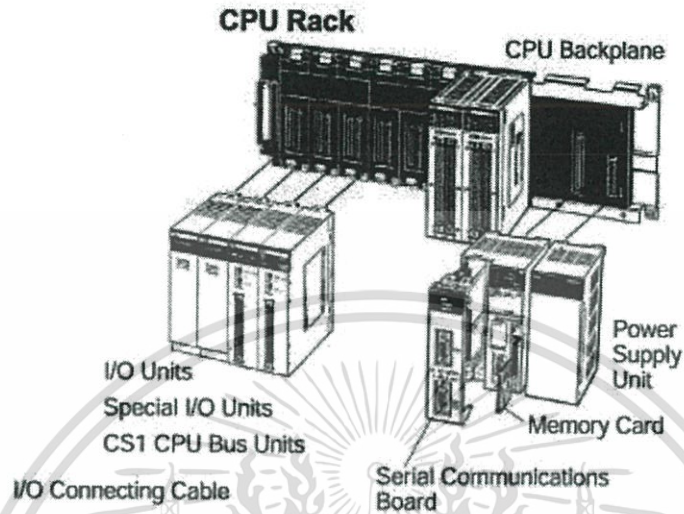
รูปที่ 2.17 แสดงรูปร่างของ PLC ชนิดโมดูล

ยกตัวอย่าง PLC รุ่น CJ1 จะใช้คอนเนคเตอร์ในการเชื่อมต่อแต่ละโมดูลเข้าด้วยกัน เพื่อให้สามารถทำงานร่วมกันได้ ดังแสดงในรูปที่ 2.18



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 2.18 แสดงชนิดของ PLC ชนิดโมดูล ที่ใช้คอนเนคเตอร์ในการเชื่อมต่อ ส่วน PLC รุ่น CS1 จะใช้ Backplane ในการเชื่อมต่อแต่ละโมดูลเข้าด้วยกันเพื่อให้ทำงานร่วมกัน ดังแสดงในรูปที่ 2.19



รูปที่ 2.19 แสดงชนิดของ PLC ชนิดโมดูล ที่ใช้ Backplane ในการเชื่อมต่อ

• ข้อดีและข้อเสียของ PLC ชนิดโมดูล

ข้อดี	ข้อเสีย
<ol style="list-style-type: none"> <li>1. เพิ่มขยายระบบได้ง่ายเพียงแค่ติดตั้งโมดูลต่างๆ ที่ต้องการใช้งานลงไปบน Back plane</li> <li>2. สามารถขยายจำนวนอินพุต/เอาต์พุตได้มากกว่าแบบ Block Type</li> <li>3. อุปกรณ์อินพุต/เอาต์พุตเสียจุดใดจุดหนึ่งสามารถถอดเฉพาะโมดูลนั้นไปซ่อม ทำให้ระบบสามารถทำการต่อได้</li> <li>4. มียูนิต และรูปแบบการติดต่อสื่อสารให้เลือกใช้งานมากกว่าแบบ Block Type</li> </ol>	<ol style="list-style-type: none"> <li>1. ราคาแพงเมื่อเทียบกับ PLC แบบ Block Type ที่มีจำนวน I/O เท่ากัน</li> </ol>

จะเห็นว่า PLC แต่ละชนิดจะมีคุณสมบัติแตกต่างกัน PLC รุ่นที่ใหญ่ขึ้น จะมีคุณสมบัติและฟังก์ชันพิเศษอื่นๆ มากกว่า PLC รุ่นเล็กซึ่งสามารถเปรียบเทียบให้เห็นความแตกต่างดังตารางต่อไปนี้

ตารางที่ 2.3 เปรียบเทียบคุณสมบัติของ PLC

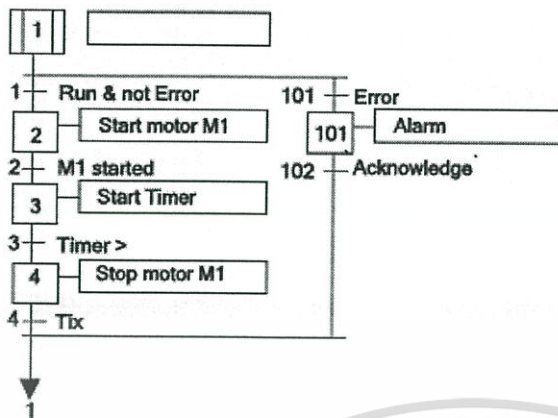
คุณสมบัติ	รุ่น			
	CPM1A	CP1L	CP1H	CS1
จำนวนอินพุต/เอาต์พุต (Max.)	100 จุด	160 จุด	256 จุด	5,120 จุด
ความจุโปรแกรม(Max.)	2 KWords	10 KSteps	20 KSteps	250 KSteps
ความเร็วในการประมวลผล	0.72 $\mu$ S	0.55 $\mu$ S	0.1 $\mu$ S	0.04 $\mu$ S
ไทมเมอร์/เคาน์เตอร์	128	4,096/4,096	4,096/4,096	4,096/4,096
หน่วยความจำในส่วนของ DM	1,024 Words	32K Words	32KWords	32KWords
ระบบสื่อสาร	<ul style="list-style-type: none"> <li>• CompoBus/S</li> <li>• Host Link</li> <li>• NT Link</li> <li>• 1:1 Link</li> </ul>	<ul style="list-style-type: none"> <li>• CompoBus/S</li> <li>• Host Link</li> <li>• NT Link</li> <li>• 1:1 Link</li> <li>• Modbus-RTU</li> <li>• Comonet</li> </ul>	<ul style="list-style-type: none"> <li>• Controller Link</li> <li>• CompoBus/D</li> <li>• Ethernet</li> <li>• Protocol Macro</li> <li>• รวมทั้งระบบสื่อสารที่มีใน PLC รุ่นต่ำกว่า</li> </ul>	<ul style="list-style-type: none"> <li>• Ethernet</li> <li>• Sysmac Link</li> <li>• Profibus-DP</li> <li>• Modbus</li> <li>• รวมทั้งระบบสื่อสารที่มีใน รุ่นต่ำกว่า</li> </ul>

### 2.5.2 ภาษาที่ใช้ในการเขียนโปรแกรมให้กับ PLC

PLC แต่ละยี่ห้อจะใช้ภาษาในการเขียนโปรแกรมเพื่อสั่งให้ PLC ทำงานตามความต้องการแตกต่างกันซึ่งตามมาตรฐาน IEC1131-3 ได้แบ่งมาตรฐานภาษาต่างๆ ออกเป็น 5 แบบตามรูปที่แสดง ข้างล่างนี้ ภาษาที่นิยมใช้มากที่สุดคือ Ladder Diagram เพราะเป็นภาษาที่ง่ายมีลักษณะคล้ายวงจรควบคุมแบบรีเลย์ ส่วนภาษาที่นิยมเป็นอันดับสองคือ Function Block

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 1) Sequential Flow Chart Language

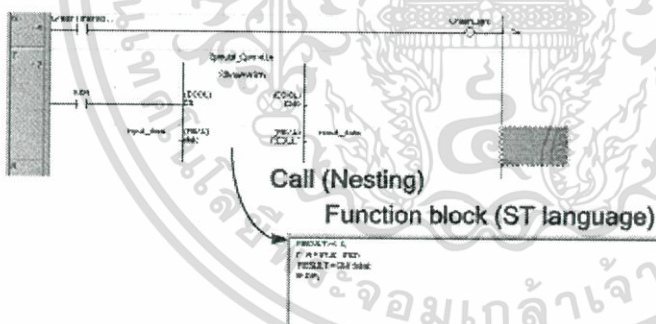


## 2) Structure Text Language

```

D: = B*B - 4*A*C;
IF D < 0.0 THEN Nroots := 0;
ELSIF D = 0.0 THEN
  Nroots := 1;
  X1 := -B/(2.0*A);
ELSE Nroots := 2;
  X1 := (-B+sqrt(D)) / (2.0*A);
  X2 := (-B-sqrt(D)) / (2.0*A);
END_IF
  
```

## 3) Function Block Diag

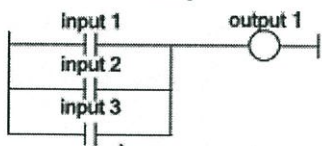


## 4) Instruction List Language

Label:	LD	a1	(*result := a1*)
	ADD	a2	(*delayed ADD, result := a2*)
	MUL	a3	(*delayed MUL, result := a3*)
	SUB	a4	(*result := a3 - a4*)
	)		(*execute delayed MUL,*)
	ADD	a6	(*result := a1 + (a2*(a3 - a4)*a5)*)
	ST	res	(*a1 + (a2*(a3 - a4)*a5) + a6*)
			(*store current result in res*)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 5) Ladder Diagram



หลังจากที่ได้เรียนรู้ภาษาที่ใช้ในการเขียนโปรแกรมให้กับ PLC แล้ว ในหัวข้อต่อไปจะกล่าวถึงอุปกรณ์ที่ใช้ในการป้อนโปรแกรมให้กับ PLC ซึ่งจะกล่าวถึงรายละเอียดดังนี้

## 2.5.3 อุปกรณ์สำหรับการโปรแกรม

การสั่งให้ PLC ทำงาน จะต้องป้อนโปรแกรมให้กับ PLC ก่อน ซึ่งอุปกรณ์ที่ใช้ในการป้อนโปรแกรมให้กับ PLC นั้น สามารถแบ่งได้เป็น 2 ประเภท

## 2.5.3.1 ตัวป้อนโปรแกรมแบบมือถือ (Hand Held Programmer)

แต่ละยี่ห้อจะมีชื่อเรียกแตกต่างกัน เช่น OMRON จะเรียกว่า Programming Console เป็นต้น สามารถยกตัวอย่างให้เห็นดังรูปที่ 2.20 ในปัจจุบัน PLC รุ่นใหม่ๆ ของออมนอนไม่ได้ใช้ Programming Console ในการเขียนโปรแกรมแล้วเพราะใช้งานยาก



รูปที่ 2.20 แสดงตัวป้อนโปรแกรมแบบมือถือ (Programming Console)

การเขียนโปรแกรมให้กับ PLC โดยการใช้ Programming Console จะป้อนเป็นภาษา Statement List หรือ Mnemonic เช่น คำสั่ง LD, AND, OR ซึ่งเป็นคำสั่งพื้นฐาน สามารถเรียกใช้งานโดยการกดปุ่มที่อยู่บนตัว Programming Console นั้น แต่เมื่อต้องการใช้งานฟังก์ชันอื่นๆ ที่มีอยู่ใน PLC สามารถเรียกใช้งานได้โดยการกดปุ่มเรียกใช้คำสั่งพิเศษ

การใช้ Programming Console มีข้อดีตรงที่มีความสะดวกในการเคลื่อนย้าย และสามารถพกพาได้สะดวกเนื่องจากมีขนาดเล็ก แต่ก็มีข้อเสียคือในการใช้งานผู้ใช้ต้องศึกษาวิธีการใช้งานของอุปกรณ์เหล่านี้ว่ามีวิธีการกดอย่างไร ถึงจะสั่งงาน PLC ได้ ในหัวข้อต่อไปจะกล่าวถึงอุปกรณ์ที่ใช้ในการป้อนโปรแกรมให้กับ PLC อีกชนิดหนึ่งคือ คอมพิวเตอร์ส่วนบุคคล (Personal Computer)

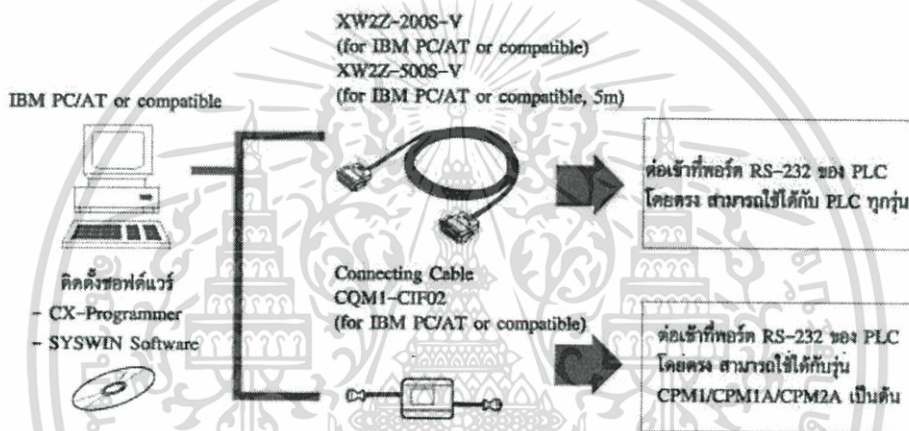
### 2.5.3.2 คอมพิวเตอร์

สามารถใช้ในการเขียนโปรแกรมให้กับ PLC ได้ โดยใช้งานร่วมกับซอฟต์แวร์ (Software) เฉพาะของ PLC ยี่ห้อนั้น เช่น PLC ของ OMRON จะใช้ซอฟต์แวร์ที่มีชื่อเรียกแตกต่างกันไป สามารถยกตัวอย่างได้เช่น

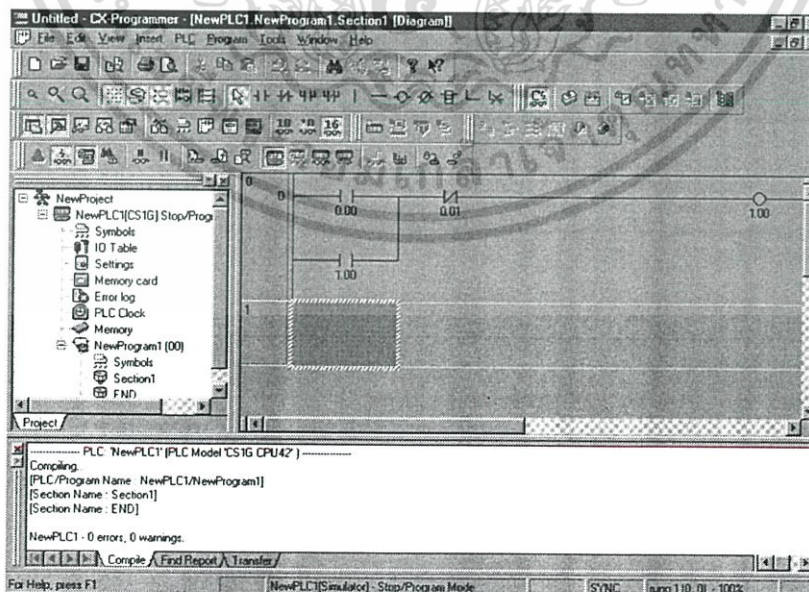
- Syswin Support Software
- CX-Programmer

ใช้ได้กับระบบปฏิบัติการตั้งแต่ Window XP ขึ้นไป หรือ Window NT ซึ่งซอฟต์แวร์ต่างๆ เหล่านี้ได้ถูกพัฒนาขึ้นเพื่อใช้กับ PLC รุ่นใหม่ที่เกิดขึ้นมาอย่างเช่น CX-Programmer มีการพัฒนาเป็นเวอร์ชันที่สูงขึ้นเรื่อยๆ เพื่อรองรับกับ PLC รุ่นใหม่ๆ และฟังก์ชันใหม่ๆ ของ PLC

วิธีการต่อคอมพิวเตอร์กับ PLC สามารถแสดงให้เห็นดังนี้



รูปที่ 2.21 แสดงวิธีการต่อใช้งานคอมพิวเตอร์กับ PLC



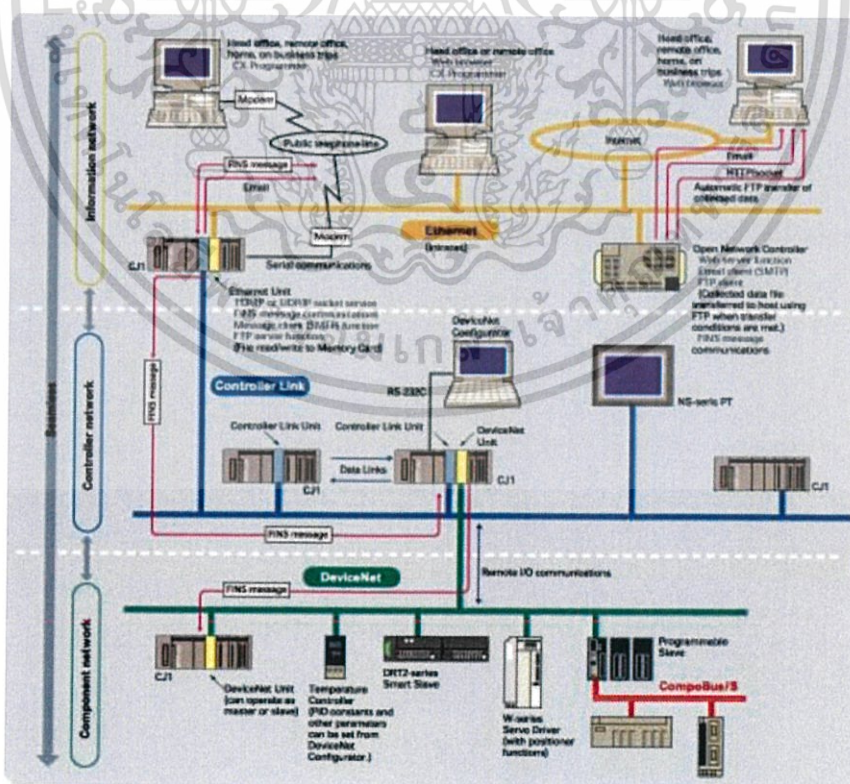
รูปที่ 2.22 ตัวอย่างซอฟต์แวร์ (CX-Programmer)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ข้อดีของการใช้เครื่องคอมพิวเตอร์ในการป้อนโปรแกรมให้กับ PLC คือ ใช้งานง่าย เช่นในกรณีใช้ CX-Programmer ร่วมกับระบบปฏิบัติการ Window จากรูปที่ 2.22 ท่านจะเห็นว่า การเขียนโปรแกรมเป็นภาษา Ladder Diagram จะเป็นการนำสัญลักษณ์ต่างๆ เข้ามาใช้แทนการเขียนคำสั่ง ทำให้เข้าใจง่ายเพียงแคคลิกเลือกสัญลักษณ์ต่างๆ จากส่วนของ Toolbar นอกจากนั้นยังมี Toolbar อื่นๆ ให้เลือกใช้งานซึ่งง่ายกว่าการใช้ Programming Console ดังนั้นสามารถสรุปได้ว่าการป้อนโปรแกรมให้กับ PLC สามารถทำได้ 2 วิธีคือ การใช้ Programming Console และการใช้เครื่องคอมพิวเตอร์ ซึ่งขึ้นอยู่กับความสะดวกของผู้ใช้ ในหัวข้อต่อไปจะกล่าวถึงระบบการติดต่อสื่อสารของ PLC

### 2.5.4 ระบบสื่อสาร (Communications)

ระบบสื่อสารของ PLC คือการนำ PLC ไปต่อใช้งานร่วมกับอุปกรณ์อื่นๆ เพื่อให้อุปกรณ์อื่นควบคุมการทำงานของ PLC หรือ ให้ PLC ไปควบคุมการทำงานของอุปกรณ์อื่น หรือ เป็นระบบที่ใช้ในการแลกเปลี่ยนข้อมูลระหว่าง PLC กับ PLC ก็ได้ ซึ่งปัจจุบัน PLC สามารถนำไปต่อร่วมกับอุปกรณ์ของยี่ห้อเดียวกัน หรืออุปกรณ์ภายนอกต่างยี่ห้อกัน เพื่อควบคุมการทำงานของระบบให้ใช้งานได้อย่างกว้างขวางมากขึ้น สำหรับระบบสื่อสารของแต่ละยี่ห้อจะมีชื่อเรียกไม่เหมือนกัน นอกจากนี้ PLC แต่ละรุ่นยังมีระบบการติดต่อสื่อสารบางรูปแบบแตกต่างกันด้วย เช่น PLC รุ่นเล็กจะมีความสามารถในการติดต่อสื่อสารได้น้อยกว่า PLC รุ่นใหญ่ เช่น PLC รุ่น CP1 สามารถใช้ระบบสื่อสารได้เฉพาะ Compobus/S, Host link, 1:1 link, NT link ส่วนรุ่นที่สูงขึ้นมาเช่น CJ1 หรือ CS1 นอกจากจะใช้ระบบที่เป็นระบบการติดต่อสื่อสารพื้นฐานที่มีใน PLC รุ่นเล็กแล้ว ยังสามารถติดต่อสื่อสารในลักษณะของ Ethernet ได้อีกด้วย สามารถแสดงตัวอย่างรูปแบบการ ติดต่อสื่อสารของ PLC ได้ดังรูปที่ 2.23

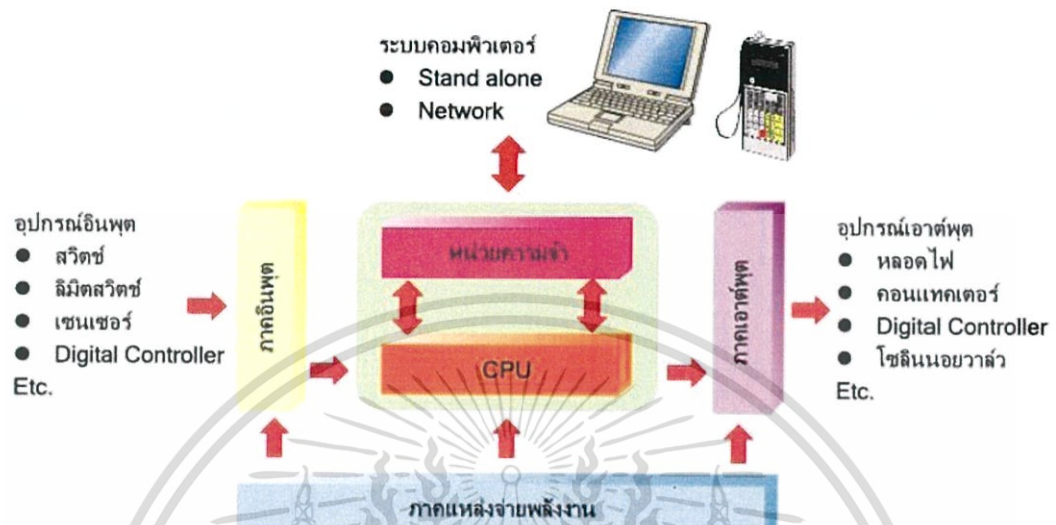


รูปที่ 2.23 ระบบการติดต่อสื่อสารของ PLC ในโรงงานอุตสาหกรรม (PLC Network)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 2.5.5 โครงสร้างของ PLC

โครงสร้างภายในของ PLC แต่ละส่วนจะประกอบกันทำงานเป็นระบบควบคุมที่เราเรียกว่า PLC ซึ่งประกอบไปด้วยส่วนสำคัญดังรูปที่ 2.24



รูปที่ 2.24 ไดอะแกรมภายใน PLC

จากไดอะแกรมดังรูปที่ 2.24 PLC จะมีส่วนประกอบสำคัญด้วยกันทั้งหมด 5 ส่วนดังนี้

1. ซีพียู (CPU; Central Processing Unit)
2. หน่วยความจำ (Memory Unit)
3. ภาคอินพุต (Input Unit)
4. ภาคเอาต์พุต (Output Unit)
5. ภาคแหล่งจ่ายพลังงาน (Power Supply Unit)

ยูนิตทั้ง 5 ส่วนเมื่อประกอบเข้าด้วยกันแล้วก็จะกลายเป็น PLC ชุดหนึ่งที่สามารถทำงานได้ แต่ละยูนิตจะมีหน้าที่และคุณสมบัติดังนี้

### 2.5.5.1 ซีพียู (CPU; Central Process Unit)

ซีพียูหรือหน่วยประมวลผลกลาง ทำหน้าที่ประมวลผลการทำงานตามคำสั่งของส่วนต่างๆ ตามที่ได้รับมา ผลจากการประมวลผลก็จะถูกส่งออกไปส่วนต่างๆ ตามที่ระบุไว้ด้วยคำสั่งนั่นเอง ซีพียูจะใช้เวลาในการประมวลผลช้าหรือเร็ว ขึ้นอยู่กับการเลือกขนาดของซีพียู และความยาวของโปรแกรมที่เขียนด้วย

ปกติแล้วซีพียูจะใช้ไมโครโปรเซสเซอร์ขนาดตั้งแต่ 4 บิต, 8 บิต, 16 บิต, 32 บิต, 64 บิต หรือ 128 บิตมาทำงาน โดยที่ซีพียูแต่ละขนาดก็จะมีประสิทธิภาพจำกัดไม่เท่ากันจึงทำให้ PLC ในแต่ละรุ่นมีความสามารถต่างกันอย่างนั้นเอง หรือแม้กระทั่งว่าภายใน PLC บางรุ่นจะใช้ไมโคร - โปรเซสเซอร์ถึง 2 ตัวช่วยกันทำงาน เวลาการประมวลผลก็จะเร็วกว่า PLC ที่ใช้ไมโครโปรเซสเซอร์เพียงแคตัวเดียว

โดยปกติแล้วการเลือกใช้งาน PLC จะเลือกจากการประยุกต์ใช้งานจึงทำให้ผู้ใช้งาน (User) ไม่รู้ว่าผู้ผลิตใช้ไมโครโปรเซสเซอร์รุ่น หรือเบอร์อะไรในการสร้างเครื่อง PLC ดังนั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เวลาพิจารณาเลือกใช้ PLC ซึ่งไม่มีการระบุเบอร์หรือรุ่นของไมโครโปรเซสเซอร์ผู้ใช้งานสามารถเลือกจากคุณสมบัติอื่น เช่น จำนวนอินพุต/เอาต์พุต, ความเร็วในการประมวลผลของคำสั่ง, ขนาดความจุโปรแกรม และข้อมูล เป็นต้น

### 2.5.5.2 หน่วยความจำ (Memory Unit)

หน่วยความจำเป็นอุปกรณ์ที่ใช้เก็บโปรแกรมและข้อมูลต่างๆ ของ PLC กรณีที่สั่งให้ PLC ทำงาน (RUN) มันจะนำเอาโปรแกรมและข้อมูลในหน่วยความจำมาประมวลผลการทำงานสำหรับหน่วยความจำที่ใช้งานมีด้วยกัน 2 ชนิด คือ

- หน่วยความจำชั่วคราว (RAM: Random Access Memory)

- หน่วยความจำถาวร (ROM: Read Only Memory)

หน่วยความจำชั่วคราว (RAM: Random Access Memory) โปรแกรมและข้อมูลที่สร้างขึ้นโดยผู้ใช้จะถูกจัดเก็บในหน่วยความจำชั่วคราวนี้ คุณสมบัติของ RAM เมื่อไม่มีไฟเลี้ยงจะทำให้โปรแกรมและข้อมูลหายไปทันที ดังนั้นภายใน PLC จะพบว่าจะมีแบตเตอรี่สำรองข้อมูล (Backup Battery) เอาไว้สำรองข้อมูล (Backup Data) กรณีที่ไฟหลัก (Main Power Supply) ไม่จ่ายไฟให้กับ PLC ข้อควรระวังคือ ไม่ควรที่จะถอดแบตเตอรี่สำรอง (Backup Battery) กรณีที่ไม่มีไฟจ่ายให้ PLC

หน่วยความจำถาวร (ROM: Read Only Memory) เป็นหน่วยความจำอีกชนิดหนึ่ง โดยที่ข้อมูลใน ROM ไม่จำเป็นต้องมีแบตเตอรี่สำรองข้อมูล แต่ก็มีปัญหาเรื่องเวลาในการเข้าถึงข้อมูล (Time Access) ช้ากว่า RAM จึงปรากฏให้ผู้ใช้เห็นว่า PLC จะมีหน่วยความจำใช้งานทั้ง RAM และ ROM ร่วมกันอยู่

ROM แบ่งออกเป็น 3 ชนิด ดังนี้

- 1) PROM (Programmable ROM)
- 2) EPROM (Erasable Programmable ROM)
- 3) EEPROM (Electrical Erasable Programmable ROM)

PROM จัดเป็น ROM รุ่นแรก เขียนข้อมูลลงชิปได้เพียงครั้งเดียว ถ้าเขียนข้อมูลไม่สมบูรณ์ชิปก็จะเสียทันที ไม่สามารถนำกลับมาเขียนใหม่ได้อีก จึงได้มีการพัฒนามาเป็นรุ่น EPROM ซึ่งสามารถเขียนข้อมูลลงชิปได้หลายครั้ง เพียงแค่นำชิปไปฉายแสงอัลตราไวโอเลตก็จะเป็นการลบข้อมูลในชิปด้วยสัญญาณทางไฟฟ้าได้เลย จึงทำให้เกิดความสะดวกสบายมากขึ้น แต่เรื่องเวลาในการเข้าถึงข้อมูลก็ยิ่งช้ากว่า RAM อยู่

การใช้งานหน่วยความจำใน PLC

- RAM จะใช้เก็บโปรแกรมและข้อมูลที่ทำงานจากการสั่ง RUN/STOP PLC

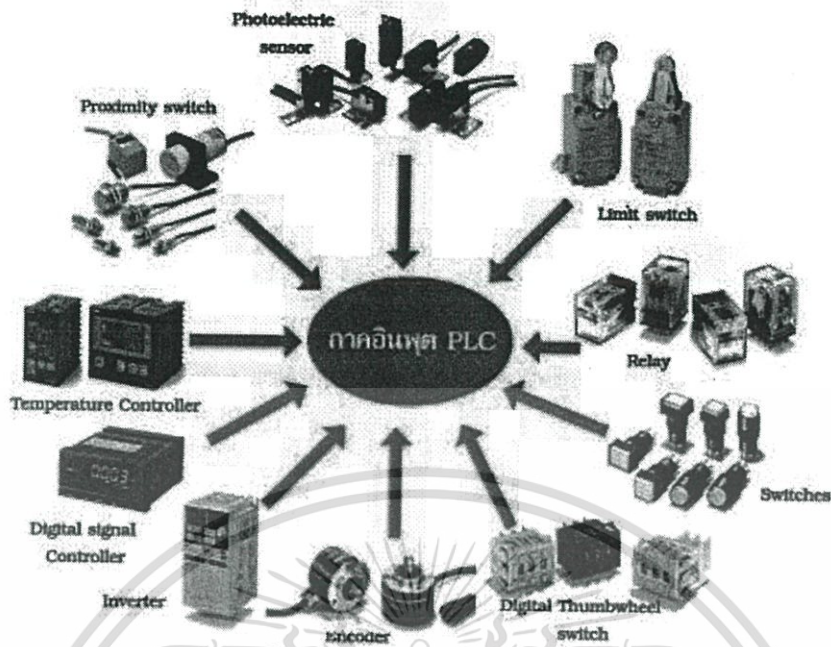
- ROM จะใช้จัดเก็บซอฟต์แวร์ระบบ (System Software) และเป็นชุดสำรอง

โปรแกรมและข้อมูล (Backup Program and Data) เพื่อป้องกันกรณีที่โปรแกรมและข้อมูลใน RAM หายไป ผู้ใช้สามารถที่จะถ่ายโปรแกรมและข้อมูลเข้าไปที่ RAM ใหม่ได้

### 2.5.5.3 ภาคอินพุต (Input Unit)

ภาคอินพุตของ PLC ทำหน้าที่รับสัญญาณอินพุตเข้ามาแปลงสัญญาณ ส่งเข้าไปภายใน PLC อุปกรณ์อินพุต (Input Device) ต่างๆ ที่นำมาต่อกับภาคอินพุตได้นั้น สามารถแสดงตัวอย่างได้ตามรูปที่แสดงดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.25 แสดงอุปกรณ์อินพุตต่างๆ

อุปกรณ์ที่สามารถนำมาต่อกับภาคอินพุต PLC ได้จัดออกเป็นกลุ่มๆ ดังรูปที่ 2.25 โดยกลุ่มอุปกรณ์แต่ละกลุ่มจะมีวิธีต่อวงจรเข้าภาคอินพุต PLC แตกต่างกันไป เวลาใช้งานอุปกรณ์แต่ละกลุ่ม จำเป็นต้องศึกษาข้อมูลเพิ่มเติมของอุปกรณ์แต่ละชนิดก่อน เพื่อความเข้าใจขั้นตอนการทำงาน และสามารถต่อวงจรได้ถูกต้อง

อุปกรณ์ที่นำมาต่อกับภาคอินพุตของ PLC อุปกรณ์บางกลุ่มจะมีสัญญาณทั้งอินพุต/เอาต์พุต เช่น Inverter, Digital Signal, Controller, ตัวควบคุมอุณหภูมิ, เซนเซอร์รุ่นพิเศษ เป็นต้น จำเป็นต้องต่อใช้งานให้ถูกต้อง ซึ่งสามารถแนะนำได้ในขั้นต้นคือ ต่อวงจรภาคเอาต์พุตของอุปกรณ์นั้นๆ เข้ากับภาคอินพุต PLC

ภาคเอาต์พุตของอุปกรณ์จะมีเอาต์พุตให้เลือกใช้งานหลายแบบ ซึ่งภาคอินพุต PLC มีวงจรภาคอินพุตอยู่หลายแบบเช่นกัน เพื่อรองรับอุปกรณ์อินพุตในแต่ละแบบให้เหมาะสม

## 2.6 ระบบ SCADA

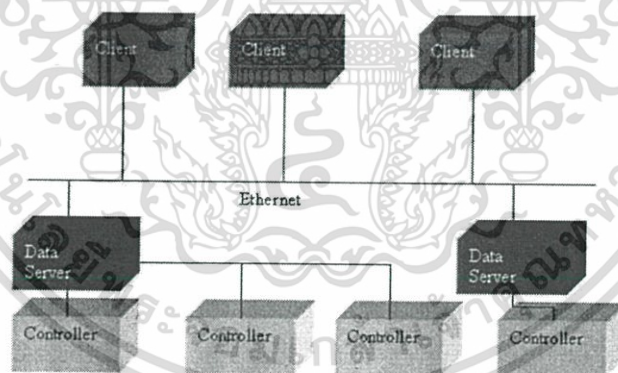
SCADA นั้นย่อมาจากคำว่า Supervisory Control And Data Acquisition เป็นระบบตรวจสอบและวิเคราะห์ข้อมูลแบบ Real-time ใช้ในการตรวจสอบสถานะตลอดจนถึงควบคุมการทำงานของระบบควบคุมในอุตสาหกรรมและงานวิศวกรรมต่าง ๆ เช่น งานด้านโทรคมนาคมสื่อสาร การประปา การบำบัดน้ำเสีย การจัดการด้านพลังงาน อุตสาหกรรมการกลั่นน้ำมันและก๊าซ อุตสาหกรรมเคมี อุตสาหกรรมประกอบรถยนต์ การขนส่ง กระบวนการนิวเคลียร์ในโรงไฟฟ้า เป็นต้น ตัวอย่างการใช้งานเช่นใช้ SCADA ตรวจสอบข้อมูลเช่น การรั่วไหลของของเหลวที่เกิดขึ้นในท่อขนส่ง จากตัวตรวจจับแล้วส่งสัญญาณแจ้งเตือนให้พนักงานทราบ โดยจะทำการส่งข้อมูลสู่ส่วนกลางของระบบ SCADA เป็นต้น นอกจากนั้น SCADA อาจทำหน้าที่คำนวณและประมวลผลข้อมูลที่ได้จากฮาร์ดแวร์ต่าง ๆ เช่น PLC, Controller, DCS, RTU แล้วแสดงข้อมูลทางหน้าจอ หรือส่งสัญญาณ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ควบคุมฮาร์ดแวร์ดังกล่าว เช่นหากอุณหภูมิของอุปกรณ์สูงเกินพิกัด ให้ทำการปิดอุปกรณ์นั้นเป็นต้น โดยสั่งงานผ่าน PLC หรือ Controller ที่ติดต่อกันอยู่ ทั้งนี้ SCADA สามารถเก็บรวบรวมข้อมูลที่ได้จากระบบควบคุมทั้งหมดไว้ในฐานข้อมูลเพื่อให้พนักงานหรือโปรแกรมอื่นๆ สามารถที่จะนำไปใช้งานได้ SCADA นั้นเข้าไปมีส่วนในงานควบคุมทั้งเล็กและใหญ่ที่ต้องการแสดงผล แลกเปลี่ยนข้อมูล หรือควบคุมระบบต่าง ๆ จากส่วนกลาง เพื่อการทำงานของระบบรวมที่สัมพันธ์กัน มองเห็นภาพรวมได้อย่างชัดเจนและมีความรวดเร็วต่อเหตุการณ์ต่าง ๆ ที่เกิดขึ้น ระบบ SCADA ในปัจจุบันมีความสามารถในการสื่อสาร ควบคุม และประมวลผลข้อมูลจาก I/O ของอุปกรณ์เช่น PLC, DCS, RTU ได้ถึงระดับที่เกินหนึ่งแสน I/O แล้ว และได้รับการพัฒนาให้มีความสามารถรองรับความต้องการใหม่ ๆ ของผู้ใช้งานอย่างต่อเนื่องตลอดมา

### 2.6.1 โครงสร้างของ SCADA (Architecture)

SCADA แบ่งตามโครงสร้างฮาร์ดแวร์ได้สองระดับคือ Client และ Data Server หรือเรียกสั้น ๆ ว่า Server โดยที่ Client คือคอมพิวเตอร์ที่รับและส่งข้อมูลไปยัง Data Server โดยฝั่ง Client นี้จะแสดงผลการทำงานของระบบควบคุมเช่น แสดงเป็นกราฟิก กราฟแบบต่อเนื่อง หรือระบบแจ้งเตือนเมื่อเกิดเหตุการณ์ฉุกเฉินหรือต้องการแจ้งเตือน เป็นต้น ฝั่ง Client สามารถสั่งงานควบคุมไปยัง Data Server เพื่อส่งสัญญาณไปยัง PLC, DCS หรือ Controller อีกทอดหนึ่ง ส่วน Data Server จะทำหน้าที่ติดต่อกับ PLC, DCS, Controller หรือ RTU ต่าง ๆ เพื่อรับสัญญาณและส่งสัญญาณไปยัง Client และรับการร้องขอจาก Client เพื่อควบคุมอุปกรณ์ PLC และ Controller ต่าง ๆ Client และ Data Server ส่วนใหญ่ติดต่อกันผ่านระบบเครือข่าย Ethernet ดังรูปที่ 2.26



รูปที่ 2.26 โครงสร้างด้านฮาร์ดแวร์ของระบบ SCADA

จากรูปที่ 2.26 นั้น Controller จะติดต่อกับอุปกรณ์ Field Instrument ต่าง ๆ เช่น เซ็นเซอร์ รีเลย์ เป็นต้น เพื่อนำสัญญาณมาให้กับ Data Server

### 2.6.2 โครงสร้างด้านซอฟต์แวร์ (Software Architecture)

โครงสร้างด้านซอฟต์แวร์ของระบบ SCADA นั้นมีข้อที่ต้องทราบคือ SCADA ใช้เทคโนโลยีในการสื่อสารกับฮาร์ดแวร์ (เช่น PLC, DCS) ต่าง ๆ กันไปตามผู้ผลิต เช่นการใช้ Driver เฉพาะของผู้ผลิต SCADA เพื่อสื่อสารกับ PLC, DCS เป็นต้น ซึ่งในปัจจุบันมีการกำหนดมาตรฐานกลางคือ OPC

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



Controller จะส่งค่าพารามิเตอร์ตามที่ถูกร้องขอให้กับ Data Server พร้อมค่าเวลาขณะนั้น (Time Stamp) การสื่อสารกับอุปกรณ์ของ Data Server นั้นอาจเป็นการสื่อสารแบบ Modbus, Profibus, CAN bus เป็นต้น ขึ้นอยู่กับมาตรฐานการสื่อสารของอุปกรณ์นั้น ๆ ว่าเป็นแบบใด ในปัจจุบันมีการสร้าง OPC Server ที่สนับสนุนการติดต่อด้วยมาตรฐานต่างๆเพิ่มขึ้นมากมายจนครอบคลุมอุปกรณ์ทุกประเภท และมีการพัฒนาให้ทั่วถึงไปยังอุปกรณ์ใหม่ ๆ อย่างต่อเนื่อง

#### 2.6.4 โครงสร้างอินเทอร์เฟซ (Interface)

การติดต่อระหว่าง Data Server กับอุปกรณ์หรือระหว่าง Data Server และ Data Server และกับ Client นั้น มีการผลิตเป็น Driver ออกมามากมายตามเทคนิคเฉพาะของแต่ละผู้ผลิต ต่อมาจึงมีการกำหนดมาตรฐานของอินเทอร์เฟซขึ้นมาเป็น OPC (OLE for Process Control) ซึ่งมีความรวดเร็วในการสื่อสารและบริการข้อมูลโดยมีการจัดตั้ง OPC Foundation ขึ้นเป็นองค์กรหลักในการกำหนดมาตรฐานและถ่ายทอดเทคโนโลยีให้แก่สมาชิก OPC จึงเป็นมาตรฐานกลางที่เปิดกว้างมากที่สุด

การติดต่อกับฐานข้อมูลภายนอกของ SCADA Software นั้น มีการสร้างให้สามารถติดต่อได้ผ่าน ODBC (Open Data Base Connectivity), OLEDB (Linking and Embedding Data Base), DDE (Dynamic Data Exchange) เป็นต้น เพื่อให้สามารถแลกเปลี่ยนข้อมูลหรือทำการเก็บข้อมูลไว้ในฐานข้อมูลรูปแบบต่าง ๆ ในปัจจุบันมีการพัฒนาให้สามารถติดต่อกับโปรแกรม ERP ต่าง ๆ เช่น SAP เป็นต้นได้ด้วย

#### 2.6.5 โครงสร้างความสามารถในการขยายระบบ (Scalability)

Scalability คือความสามารถในการรองรับและต่อขยายระบบ SCADA กับส่วนต่าง ๆ เช่น I/O ของอุปกรณ์ Controller และจำนวนเครื่อง SCADA Client ที่เพิ่มขึ้น หรือการต่อพ่วงกับระบบ SCADA ของยี่ห้ออื่น ๆ เป็นต้น ถ้าหาก Data Server เป็นแบบ Driver ที่สร้างด้วยเทคโนโลยีเฉพาะในการติดต่อกับอุปกรณ์ ก็เป็นเรื่องลำบากในการต่อขยาย เพราะ Driver บางประเภทสามารถติดต่อกับเฉพาะ SCADA Software บางยี่ห้อเท่านั้น ปัญหานี้เป็นที่วิพากษ์วิจารณ์กันอย่างกว้างขวาง ซึ่งปัจจุบันได้หันมาใช้มาตรฐานกลางคือ OPC เพื่อแก้ไขปัญหานี้

#### 2.6.6 โครงสร้างการสำรองระบบ (Redundancy)

SCADA Software ส่วนใหญ่มีความสามารถในการสำรองระบบของ Data Server โดยที่เมื่อ Data Server เกิดความขัดข้องก็จะสั่งงานให้ Data Server อีกตัวหนึ่งทำงานแทนที่ โดยจะมีการกำหนดคอนฟิกูเรชันไว้ที่ Client ว่าจะให้เลือกติดต่อกับ Data Server ตัวไหนเมื่อเกิดความขัดข้องเกิดขึ้น

ในบางครั้งโมดูลที่ทำหน้าที่จัดการด้าน Redundancy นี้อาจจะทำหน้าที่อีกประการหนึ่งคือเป็นจุดพักข้อมูลที่รับมาจาก Data Server เพื่อนำไปส่งให้กับ Client ต่าง ๆ เพราะในกรณีที่ Client จำนวนมากติดต่อกับ Data Server ตัวเดียวนั้นอาจมีความล่าช้าในการบริการข้อมูลของ Data Server เพราะต้องให้บริการข้อมูล Client ให้ครบจำนวนก่อนที่จะไปรับข้อมูลใหม่จากอุปกรณ์มาได้ ดังนั้นโมดูลที่ทำหน้าที่ Redundant จึงทำหน้าที่เป็นจุดรับข้อมูลแล้วช่วยส่งต่อให้ Client

ต่างๆ อีกทอดหนึ่ง Data Server จะได้ทำหน้าที่บริการข้อมูลให้แก่หนดเพียงจุดเดียว จึงมีความรวดเร็วในการบริการข้อมูล

### 2.6.7 หน้าที่การทำงาน (Functionality)

การเข้าถึงพารามิเตอร์ของอุปกรณ์ หมายถึงความสามารถในการเข้าถึงกลุ่มของพารามิเตอร์ในอุปกรณ์เช่น I/O ของ PLC เป็นต้น ความสามารถของ Data Server ในการกำหนดว่าพารามิเตอร์ใด อ่านได้อย่างเดียว เขียนได้อย่างเดียว หรือทั้งอ่านทั้งเขียน เป็นต้น

### 2.6.8 ระบบแสดงผลแบบ MMI (Man Machine Interface)

คือความสามารถในการแสดงผลการทำงานของอุปกรณ์ในรูปแบบ กราฟิก ข้อความ สัญลักษณ์ แผนภาพ เป็นต้น โดยสามารถเชื่อมโยงลักษณะการเปลี่ยนแปลงของกราฟิกเหล่านี้กับพารามิเตอร์จาก Data Server ได้ ความสามารถในการสั่งงานผ่านระบบกราฟิกเช่น การปิด/เปิด สวิตช์บนจอมอนิเตอร์ส่งผลไปยัง I/O ของ PLC เป็นต้น

ความสามารถในการจัดการกราฟิกเช่น การย่อ ขยาย การกำหนดการเคลื่อนไหวแบบต่าง ๆ เช่น การหมุน การเคลื่อนที่แบบซิกแซ็กตามสัญญาณของ Data Server การแสดงผลสัญญาณในรูปแบบมิเตอร์และเกจวัดแบบต่าง ๆ การนำเข้ากราฟิกประเภทต่างๆ การจัดแบ่งเลย์เออร์ เป็นต้น สิ่งเหล่านี้เป็นข้อเปรียบเทียบความสามารถของ SCADA Software ทั้งสิ้น

### 2.6.9 ระบบแสดงกราฟสัญญาณแบบต่อเนื่อง (Trending)

Trending เป็นความสามารถในการพล็อตกราฟต่อเนื่องกันไปในจอภาพเพื่อแสดงค่าสัญญาณจาก Data Server โดยอาจจะสามารถพล็อตสัญญาณได้หลายสัญญาณเช่น 8 – 24 สัญญาณพร้อมกันในหน้าต่างเดียว เพื่อให้สามารถเปรียบเทียบสัญญาณที่พล็อตได้ และไม่จำกัดว่าจะสร้างหน้าต่างพล็อตจำนวนเท่าใด Trending อาจมีความสามารถในการ ชุมสัญญาณที่พล็อต และหยุดการพล็อตเพื่อเลื่อนดูค่าที่พล็อตในแต่ละช่วงเวลาได้ด้วยตัวของผู้ใช้งานเอง นอกจากนั้นการพล็อตอาจสามารถเลือกได้ว่าจะให้เป็นการพล็อตแบบใดเช่น Time plot, Logarithmic plot, Strip Chart, Bar Chart, Circular, X-Y plot เป็นต้น นอกจากนั้นบางผู้ผลิตยังสามารถนำค่า Historian หรือข้อมูลสัญญาณที่เก็บไว้ในฐานข้อมูลออกมาพล็อต ได้อีกด้วย โดย Trending Module นี้ อาจเป็นแบบ ActiveX Control คือสามารถนำไปใช้งานในแอปพลิเคชันอื่นที่สนับสนุนการนำเข้า ActiveX ได้

### 2.6.10 ระบบแจ้งเตือน (Alarm)

SCADA Software ส่วนใหญ่มีระบบแจ้งเตือนโดย Alarm Display จะรับสัญญาณมาจาก Alarm DB ในฝั่ง SCADA Server โดย Alarm DB สามารถที่จะทำการกำหนดคอนฟิกเรชั่นว่าจะนำสัญญาณตัวใดมาเป็นตัวพารามิเตอร์ในการแจ้งเตือนบ้าง และมีการแบ่งระดับของ Priority, Limit อย่างไร เป็นต้นระบบแจ้งเตือนยังสามารถที่จะเก็บข้อมูลการแจ้งเตือนไว้ในฐานข้อมูลประเภทต่าง ๆ ได้เช่น MS SQL Server, MS Access, Oracle, MS Excel เป็นต้น และบางยี่ห้อสามารถแสดงออกมาเป็นรายงานในรูปแบบตารางหรือ แผนภูมิได้อีกด้วย

### 2.6.11 การทำงานแบบ Automation

เป็นความสามารถที่ SCADA ทำหน้าที่ต่าง ๆ ตามที่กำหนด เช่น ส่งอีเมล แสดงข้อความแบบ Instance Message บนหน้าจอ เปิดไปยังหน้าจออื่น ๆ เก็บข้อมูลลงฐานข้อมูล เปิดโปรแกรมหรือรันคำสั่งสคริปต์ เป็นต้น ตามสัญญาณที่ได้รับจาก Data Server และข้อกำหนดที่สร้างขึ้น

### 2.6.12 การสร้างและพัฒนา (Application Development)

การกำหนดคอนฟิกูเรชัน ขั้นแรกต้องมีการกำหนดว่าจะติดต่อกับพารามิเตอร์หรือ Tag ใดบ้างจาก Data Server ดังนั้นจะต้องทำการ Define หรือสร้าง Tag ที่ Data Server ก่อนว่า Tag แต่ละตัวหมายถึง Address ที่เท่าใดของอุปกรณ์ (PLC, DCS, RTU, Controller ต่างๆ) โดยทั่วไปสามารถทำการนำเข้าคอนฟิกูเรชันไฟล์ที่สร้างไว้ก่อนเข้ามาได้ และสามารถ Export ไปยัง Data Server อื่น ๆ ได้ จากนั้นโปรแกรมย่อยอื่น ๆ ของ SCADA Software ผังไคลเอนต์ จึงทำคอนฟิกูเรชันตามหน้าที่การทำงานของตนเอง เช่น โมดูลที่มีหน้าที่แสดงผลกราฟก็ต้องกำหนดว่ากราฟนั้น ๆ จะเชื่อมโยงกับ Tag ใดจาก Data Server ส่วนโมดูลที่ทำหน้าที่แจ้งเตือนก็ต้องทำคอนฟิกูเรชันว่าจะนำ Tag ใด มาเป็นสัญญาณแจ้งเตือน และกำหนดระดับสัญญาณ Limit เป็นต้น

### 2.6.13 เครื่องมือในการพัฒนา (Development Tool)

เครื่องมือในการสร้างและพัฒนาระบบ SCADA โดยทั่วไปจะประกอบด้วย

1. เครื่องมือในการสร้างระบบกราฟิก ที่ประกอบด้วยเครื่องมือวาดภาพ เครื่องมือกำหนดเอฟเฟกพิเศษต่าง ๆ โลโก้ของกราฟิกสำเร็จรูปในอุตสาหกรรมด้านต่าง ๆ
2. เครื่องมือในการสร้าง Trending
3. เครื่องมือในการสร้างระบบ Alarm
4. เครื่องมือในการกำหนดการติดต่อกับฐานข้อมูลเพื่อทำการเก็บรวบรวมข้อมูลของ Trending และ Alarm ลงไว้ในฐานข้อมูล
5. เครื่องมือในการช่วยสร้าง Script เช่น Java script, VB Script
6. เครื่องมือจัดการด้านความปลอดภัย การแบ่งระดับ User และขอบเขตการใช้งานของ User
7. เครื่องมือในการสร้าง Web application เพื่อให้สามารถควบคุมและตรวจสอบระบบควบคุมผ่าน Web browser ได้

ที่กล่าวมาทั้งหมดนี้เป็นลักษณะของ SCADA และ SCADA Software ส่วนใหญ่ ทั้งนี้คุณผู้อ่านก็จะพอเห็นภาพว่า SCADA นั้นสามารถเป็นศูนย์กลางของระบบควบคุมทั้งหมดขององค์กร และมีส่วนช่วยในการตรวจสอบการทำงานของระบบให้เป็นไปตามปกติได้อย่างมีประสิทธิภาพและทั่วถึง ภายในเวลาอันรวดเร็ว มีส่วนช่วยในการตัดสินใจในการดำเนินงานจากข้อมูลต่าง ๆ ที่ได้รับจากระบบ SCADA นอกจากนี้เรายังสามารถเชื่อมโยงข้อมูลที่ได้จาก SCADA เข้ากับข้อมูลทางธุรกิจอื่น ๆ เพื่อประมวลผลร่วมกัน เช่น ข้อมูลจำนวนของเสียเป็นกิโลกรัมที่ตรวจสอบได้จาก SCADA ถูกนำมาคำนวณร่วมกับค่าใช้จ่ายอื่น ๆ แบบ Real time เพื่อสรุปเป็นรายงานค่าใช้จ่ายประจำวัน เป็นต้นได้อย่างรวดเร็ว

## 2.7 OPC (OLE for Process Control (OPC))

OLE (Object Link and Embedding Interface) คือตัวแรกในคำย่อของ OPC เนื่องจากในปัจจุบันระบบปฏิบัติการ window ได้มีบทบาทเพิ่มมากขึ้นในระบบควบคุมอุตสาหกรรมกระบวนการผลิตที่มีใช้งานกันอยู่ทั่วไป อาทิเช่น ระบบ DCS (Distributed Control System), SCADA (Supervisory Control and Data Acquisition) และ PLC (Programmable Logic Controller) โดยระบบควบคุมเหล่านี้จะประกอบไปด้วยส่วนต่างๆหลายส่วน โดยทั้งหน่วยแสดงผล (Human Machine Interface) หรือหน่วยปฏิบัติการวิศวกรรม (Engineering Unit) ส่วนใหญ่แล้วทั้งสองส่วนนี้จะทำงานอยู่บนระบบปฏิบัติการ Windows และหน่วยปฏิบัติการดังกล่าวจะเป็นส่วนประกอบพื้นฐานที่สำคัญของระบบควบคุมปัจจุบันเพื่อความสะดวกในการติดต่อกับผู้ปฏิบัติการในการควบคุมกระบวนการผลิตหรือใช้สำหรับแก้ไขเพิ่มเติมคำสั่งในฟังก์ชันควบคุม ซึ่งถ้าเป็นข้อดีของระบบปฏิบัติการ Windows คือ ได้มีการจัดเตรียมมาตรฐานการสื่อสารกับส่วนอื่นๆ ให้เลือกใช้งานได้หลายวิธี ซึ่งประกอบไปด้วย DDE (Dynamic Data Exchange) และ OLE (Object Link and Embedding) ซึ่ง OLE เป็นมาตรฐานการสื่อสารที่ทำงานอยู่บนพื้นฐานของ COM (Component Object Model) นอกจากนั้นยังมีมาตรฐานการสื่อสารสำหรับระบบการควบคุมโดยเฉพาะนั่นคือการสื่อสารด้วย OPC หรือ OLE for Process Control ซึ่งเป็นเทคโนโลยี การสื่อสารรูปแบบใหม่ที่ได้ถูกพัฒนาขึ้นมาสำหรับระบบควบคุมอุตสาหกรรมกระบวนการผลิตโดยสมาคมของผู้ผลิตผู้จัดจำหน่ายที่เกี่ยวข้องระบบเครื่องมือวัดและควบคุมในอุตสาหกรรมกระบวนการผลิต การสื่อสารด้วย OPC จะเป็นรูปแบบการสื่อสารที่มีความเชื่อมั่นและประสิทธิภาพสูงกว่าการสื่อสารแบบ DDE และ OLE

สำหรับในทางอุดมคติแล้วการเชื่อมต่อระบบและอุปกรณ์ต่างๆเข้าด้วยกันควรจะมีความสะดวกและไม่ซับซ้อนในการเชื่อมต่อ ถึงแม้ว่าอุปกรณ์ต่างๆจะถูกจำหน่ายหรือผลิตมาจากแหล่งเดียวกันหรือผลิตจากแหล่งที่ต่างกันนั้น ซึ่งในการเชื่อมต่อควรจะทำได้เพียงการเชื่อมต่ออุปกรณ์เหล่านี้เข้ากับเครือข่าย (Network) มาตรฐานแบบใดแบบหนึ่ง แต่ในความเป็นจริงแล้วไม่ได้ง่ายและสะดวกอย่างที่ต้องการ ถึงแม้ว่าการจัดหาอุปกรณ์ต่างๆ อาทิเช่น PLC , DCS ชุดควบคุมมอเตอร์, เครื่องมือวัด หรืออุปกรณ์จัดเก็บข้อมูลจะมาจากผู้ผลิตเดียวกัน หรืออุปกรณ์แบบเดียวกันแต่รุ่นที่ผลิตต่างกันไปยังไม่สามารถเชื่อมต่อเพื่อทำการสื่อสารกันได้อย่างสมบูรณ์

OPC (OLE for Process Control) เป็นชุดการสื่อสารมาตรฐานสำหรับการเชื่อมต่อและวิธีการที่จะนำไปใช้ในการสื่อสารกับระบบควบคุมในอุตสาหกรรมกระบวนการผลิตและอุตสาหกรรมแบบอัตโนมัติต่างๆ OPL เป็นเทคโนโลยีที่อยู่บนพื้นฐาน Windows's OLE , Com และ DCOM (Distributed Component Object Model) ซึ่งเทคโนโลยีต่างๆเหล่านี้จะเป็นตัวที่ใช้กำหนดให้โปรแกรมที่ทำงานบนเครื่องคอมพิวเตอร์แต่ละชนิดมีความสามารถที่ใช้ข้อมูลร่วมกันได้ผู้ใช้งานอาจเคยได้ยินหรือเคยใช้ ความสามารถในการสื่อสารแบบ OLE มาบ้างแล้ว ดังเช่น การเพิ่มเติม Spreadsheet บนเอกสาร Word ได้ตลอดเวลา จากตัวอย่างจะเห็นว่าผู้ใช้งานไม่จำเป็นต้องกำหนดรายละเอียดบนโปรแกรมเพิ่มเติมเพียงแต่ใช้เมาส์คลิกไปยังส่วนต่างๆในการทำงานเท่านั้น หลังจากนั้น OLE จะทำการกำหนดให้ Spreadsheet ทำหน้าที่เป็นโปรแกรมแม่ข่าย (OLE Server) สำหรับส่งข้อมูลไปยังเอกสาร Word (ถูกกำหนดให้ทำหน้าที่เป็นโปรแกรมลูกข่าย (OLE Client))

OPC จะเป็นรูปแบบการสื่อสารที่ถูกพัฒนามาจาก OLE โดยการเพิ่มคุณลักษณะที่เป็นประโยชน์ต่างๆเข้าไปใน OLE เพื่อทำให้มีความน่าเชื่อถือสูงกับการนำไปใช้งานในระบบควบคุม

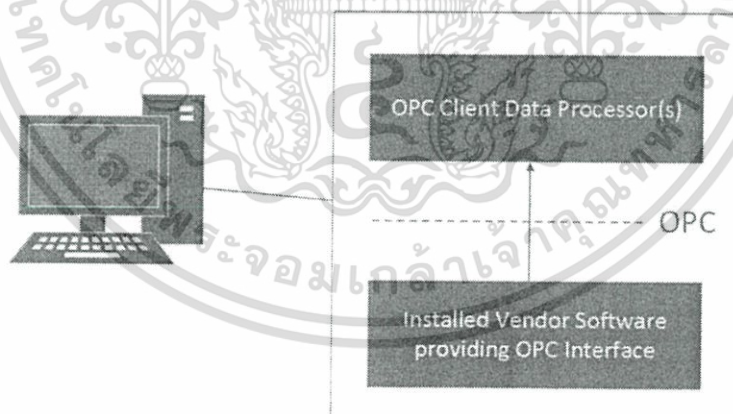
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อุตสาหกรรมกระบวนการผลิต เนื่องจากโปรแกรมแม่ข่ายของ OLE (OLE Server) จะไม่มีคุณสมบัติในการตรวจสอบข้อมูลที่ได้ถูกส่งออกไปยังโปรแกรมลูกข่ายของ OLE (OLE Client) และการตรวจสอบข้อมูลที่ได้รับว่าถูกต้องหรือไม่ ซึ่งจะเห็นได้ว่าความถูกต้องและความน่าเชื่อถือได้ของการรับส่งข้อมูลจะมีค่าต่ำมาก ดังนั้นจึงไม่เหมาะสมกับการนำไปใช้งานกับระบบควบคุมอุตสาหกรรมกระบวนการผลิตได้ ซึ่ง OPC ได้ทำการเพิ่มเติมความสามารถในการตรวจสอบข้อมูลระหว่างการสื่อสารระหว่างโปรแกรมลูกข่าย (Client Application) และโปรแกรมแม่ข่าย (Server Application) จึงทำให้ข้อมูลที่ได้มีความถูกต้องแม่นยำและมีความเชื่อถือได้มากยิ่งขึ้น

ค่าข้อมูลจริงของ OPC จะถูกจัดเตรียมไว้สำหรับเชื่อมต่อร่วมกันระหว่างการสื่อสารกับอุปกรณ์หลายชนิดรวมถึงโปรแกรมคอมพิวเตอร์สำหรับใช้ในการควบคุมหรืออุปกรณ์ในกระบวนการผลิต ก่อนหน้าที่ OPC ได้ถูกพัฒนาขึ้นผู้พัฒนาโปรแกรมคอมพิวเตอร์ต้องทำการพัฒนา Driver สำหรับการสื่อสารกับอุปกรณ์แต่ละชนิดอย่างเฉพาะเจาะจง สำหรับระบบควบคุมที่จะนำไปเชื่อมต่อกับอุปกรณ์นั้นๆ เช่น หน่วยแสดงผลหนึ่งชุดผู้จำหน่ายต้องทำการพัฒนา Driver หลายๆ แบบเพื่อใช้สำหรับระบบ PLC หรือระบบ DCS ที่แตกต่างกันไป

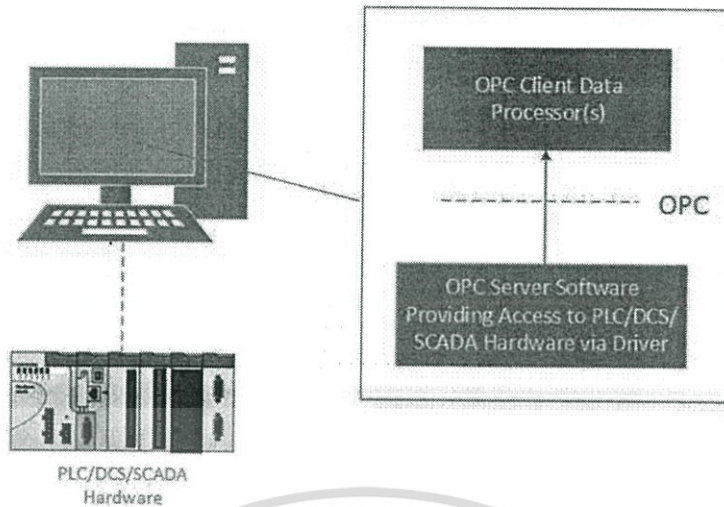
สำหรับการใช้งาน OPC ผู้จัดจำหน่ายหน่วยแสดงผลไม่จำเป็นต้องทำการพัฒนา Driver หลายๆ แบบเพื่อเตรียมไว้สำหรับเครือข่ายหรือระบบควบคุมที่แตกต่างกัน และในทางกลับกันผู้จัดจำหน่ายเหล่านี้เพียงแต่ทำการพิจารณา OPC Driver ที่เหมาะสมกับผลิตภัณฑ์ของตนเอง ไว้ใช้ในการเชื่อมต่อเพื่อทำการสื่อสารไปยัง OPC Driver ของผลิตภัณฑ์จากผู้ผลิตรายอื่นๆ

ในปัจจุบันกลไกสื่อสารมาตรฐานสำหรับระบบปฏิบัติการ Windows อย่างเช่น OPC ได้มีการจัดเตรียมส่วนเชื่อมต่อสำหรับการใช้งาน (Application Programmer's Interface : API) ของผลิตภัณฑ์หลายๆชนิด โดยระบบนี้สามารถเข้าถึงข้อมูลได้ทางโปรแกรมการใช้งาน ดังแสดงในรูปที่ 2.28 หรืออาจจะเข้าถึงข้อมูลจากอุปกรณ์ควบคุมโดยผ่าน Driver ดังแสดงในรูปที่ 2.29



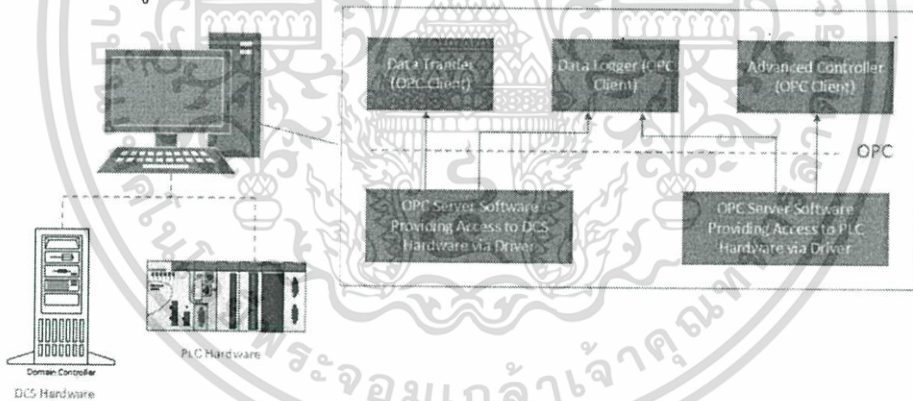
รูปที่ 2.28 การเข้าถึงข้อมูลของโปรแกรมการใช้งาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.29 การเข้าถึงข้อมูลจากอุปกรณ์ผ่าน Driver

การใช้งาน OPC ในการเชื่อมต่อระบบควบคุมต่างๆเข้าด้วยกัน เริ่มมีจำนวนเพิ่มขึ้นสำหรับการเชื่อมต่อระบบควบคุมอื่นๆ (Third party system) เข้ากับระบบควบคุมหลัก ข้อดีที่เห็นได้อย่างชัดเจนจะเป็นความสามารถในการใช้งานโปรแกรมแบบพิเศษต่างๆบนระบบควบคุมกระบวนการผลิต ข้อดีนี้จะทำให้ผู้ใช้งานมีทางเลือกในการใช้งานโปรแกรมแบบพิเศษเหล่านี้ได้อย่างกว้างขวางดังตัวอย่างเช่น Advance Control, Trending, Data logging หรือ Data Conditioning เป็นต้น ซึ่งสามารถแสดงได้ในรูปที่ 2.30



รูปที่ 2.30 การใช้งานโปรแกรมแบบพิเศษต่างๆบนระบบควบคุม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

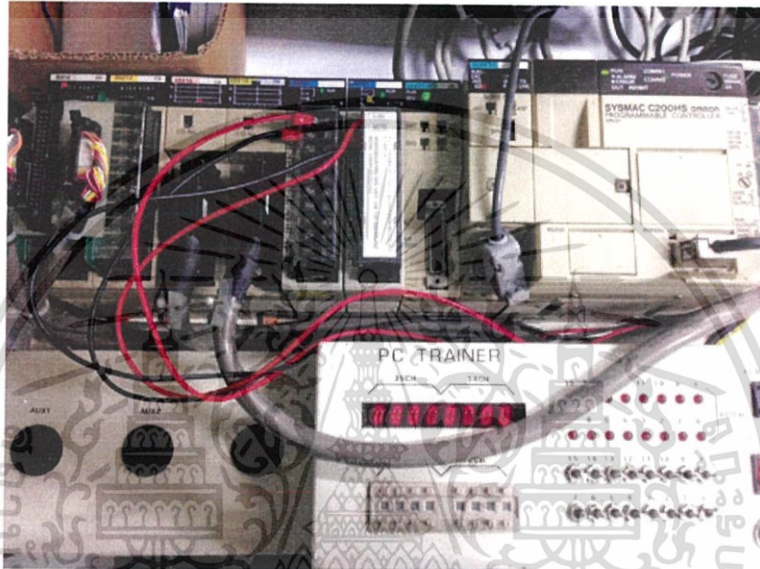
## บทที่ 3

### ขั้นตอนการดำเนินงาน

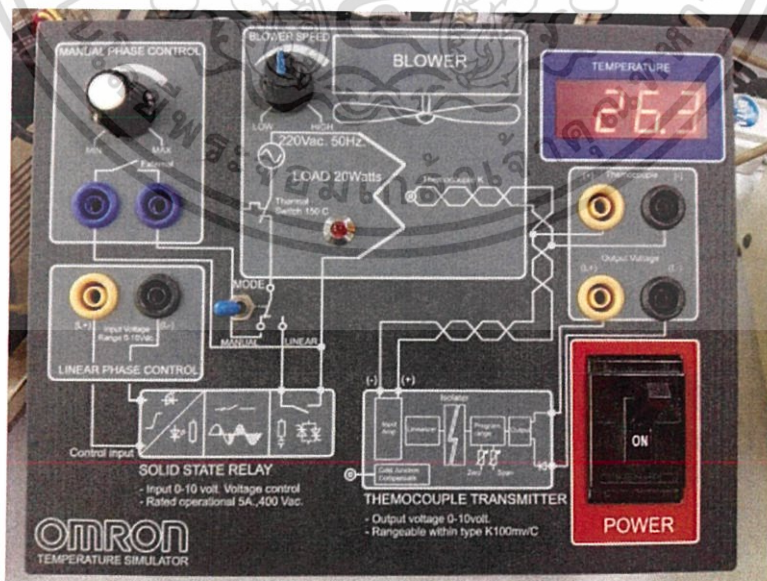
#### 3.1 อุปกรณ์และสถานที่ดำเนินการ

##### 3.1.1 อุปกรณ์

1. PLC ของ OMRON รุ่น SYSMAC C200HS CPU31



2. Temperature Simulator ของ OMRON



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3. DC. Power Supply

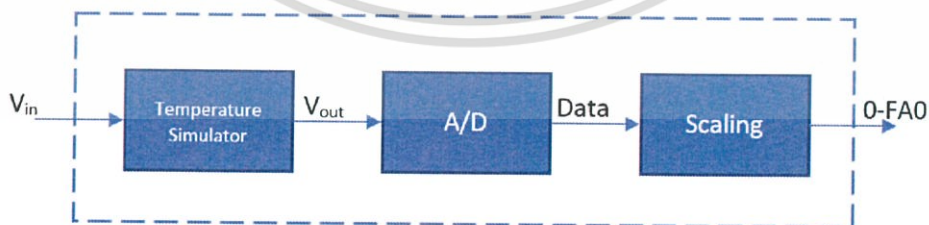


#### 3.1.2 สถานที่ดำเนินการ

ห้องปฏิบัติการ Factory Automation ภาควิชาวิศวกรรมการวัดคุม คณะ  
วิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

#### 3.2 การห่าย่นควบคุมการทำงานด้านตัวแปรอินพุท

ในการทดลองเพื่อศึกษ่างานการทำงานด้านตัวแปรอินพุท จะดำเนินการโดยการป้อนแรงดัน  
จากแหล่งจ่าย DC. Power Supply 0-10 VDC ให้กับ Heater แล้วทำการอ่านค่าแรงเคลื่อนไฟฟ้าที่  
ได้จาก Thermocouple นำข้อมูลที่ได้เป็นข้อมูลเข้า A/D Analog Input Unit ของเครื่องควบคุม  
PLC เพื่อไปคำนวณให้ย่นการเปลี่ยนแปลงเป็น 0-FA0 (HEX) ซึ่งถือว่าเป็นตัวแปรอินพุทของ  
กระบวนการดังรูปที่ 3.1 และในการเชื่อมต่อสัญญาณระหว่าง Temperature Simulator กับเครื่อง  
ควบคุม PLC ดังรูปที่ 3.2

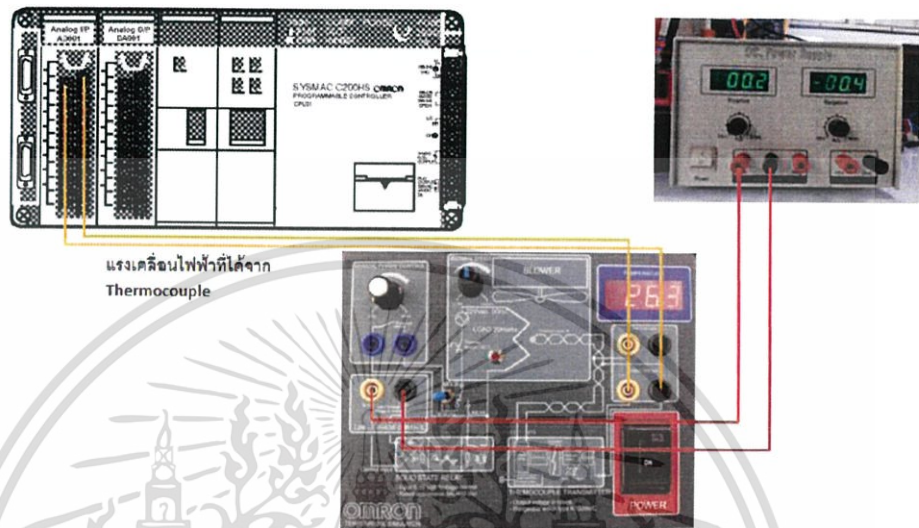


รูปที่ 3.1 ตัวแปรอินพุทของกระบวนการ

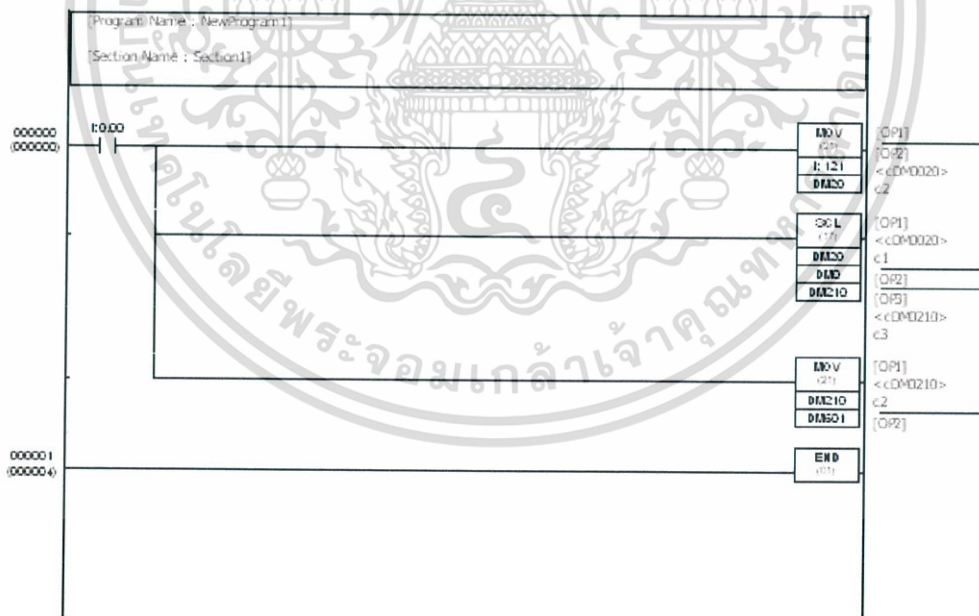
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### ลำดับการทดลอง

1. Temperature Simulator เลือกโหมดสวิตซ์มาที่ตำแหน่ง Manual
2. ทำการต่อวงจรตามรูปที่ 3.2
3. เขียนโปรแกรมการทำงานตาม Ladder รูปที่ 3.3



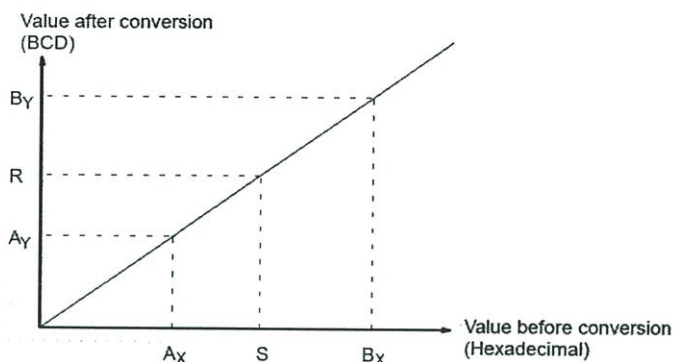
รูปที่ 3.2 การเชื่อมต่อสัญญาณระหว่าง Temperature Simulator กับเครื่องควบคุม PLC



รูปที่ 3.3 แสดงโปรแกรม Ladder

4. กำหนดค่าพารามิเตอร์ให้กับ DM000-DM003 ของคำสั่ง Scaling-SCL( )ให้สอดคล้องกับรูปที่ 3.4 และค่า Min. Max. ให้สอดคล้องกับรูปที่ 3.4

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.4 แสดง Min และ Max ข้อมูลต้นทางและปลายทางคำสั่ง SCL( )

ตารางที่ 3.1 (นำผลจากตารางที่ 3.2 มาใส่เป็นค่าที่ได้จากการทดลอง)

หน่วยความจำ	ข้อมูล	ค่าตัวแปรคำสั่ง SCL( )
DM000		P1 (ค่า Min ในแนวแกน Y) หรือ Ay
DM001		P1+1 (ค่า Min ในแนวแกน X) หรือ Ax
DM002		P1+2 (ค่า Max ในแนวแกน Y) หรือ By
DM003		P1+3 (ค่า Max ในแนวแกน X) หรือ Bx

5. ทำการดาวน์โหลดโปรแกรมให้กับเครื่องควบคุม PLC
6. ตรวจสอบความถูกต้องของวงจรพร้อมเปลี่ยนสวิทซ์ให้อยู่ในตำแหน่ง LINEAR ขณะเดียวกัน “ON” สวิทซ์ อินพุต 0.00 ให้ PLC พร้อมทั้ง RUN โปรแกรมการทำงาน
7. ป้อนแรงดันจากแหล่งจ่าย DC. Power Supply 0-10 VDC โดยเริ่มจาก 0, 0.1, 0.2, ..., 0.5, 1.0, 1.5, ... ในแต่ละช่วงให้ทำการวัด และบันทึกค่าแรงเคลื่อนไฟฟ้า และค่าอุณหภูมิที่วัดได้จาก Thermocouple ซึ่งเป็นข้อมูลของ A/D ลงในตารางที่ 3.2
8. จากตารางที่ 3.2 นำค่าแรงเคลื่อนไฟฟ้าย้อนกลับเพื่อหาค่าของข้อมูล A/D เป็น 0-4000<sub>(10)</sub> ที่แรงเคลื่อนไฟฟ้า 0-10 โวลต์หรือความละเอียดเท่ากับ 400/โวลต์ ได้ข้อมูลเลขฐาน 16 แล้วบันทึกค่าไว้
 

ตัวอย่าง แรงเคลื่อนไฟฟ้าเป็น 2.5 โวลต์ อ่านค่าอุณหภูมิได้ 25 องศาเซลเซียส  
 ข้อมูลเลขฐาน 10 A/D =  $2.5 \times 400 = 1000_{(BCD)}$   
 ข้อมูลเลขฐาน 16 A/D =  $3E8_{(HEX)}$
9. เปลี่ยนโหมดสวิทซ์ของ Temperature Simulator ให้อยู่ตำแหน่ง Manual ขณะเดียวกัน “OFF” สวิทซ์อินพุต 0.00 ให้ PLC
10. นำข้อมูลจากตาราง 3.2 ไปเขียนกราฟความสัมพันธ์ระหว่าง แรงดันที่ป้อนให้ Heater ในแนวแกน X กับ ค่าแรงเคลื่อนไฟฟ้าที่ได้จาก Thermocouple ในแนวแกน Y

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3.2

No.	แรงดันที่ ป้อนให้ Heater	อุณหภูมิจาก Thermocouple	แรงเคลื่อนไฟฟ้าที่ ได้จาก Thermocouple	A/D, Wd121 หรือ DM20	
				BCD	HEX
1					
2					
3					
4					
5					
6					
7					
8					
9					

### 3.3 การห่าย่นควบคุมการทำงานด้านตัวแปรเอาต์พุท

ในการห่าย่นการทำงานของตัวแปรเอาต์พุทของกระบวนการกระทำได้โดยการป้อนแรงดันไฟตรงจาก D/A Analog Output Unit ของเครื่องควบคุม PLC ให้กับ Heater แล้วอ่านค่าแรงเคลื่อนไฟฟ้าที่ได้จาก Thermocouple กลับเข้ามาคำนวณทำเป็น 0-100% เช่นเดียวกับตัวแปรอินพุท วิธีการนี้เป็นการห่าย่นของตัวแปรเอาต์พุทของกระบวนการ

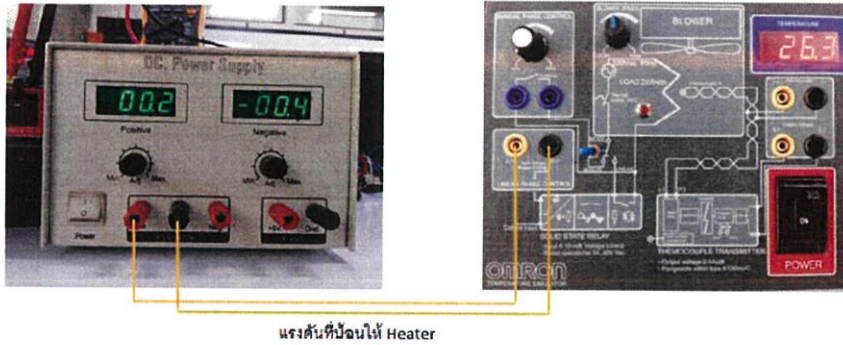


รูปที่ 3.5 แสดงกระบวนการของตัวแปรเอาต์พุท

#### ลำดับขั้นการทดลอง

แบ่งออกเป็นขั้นตอนที่ 1 คือ สัญญาณควบคุมจากแหล่งจ่ายไฟตรงภายนอก และขั้นที่ 2 จากชุด D/A

1. Temperature Simulator เลือกโหมดให้อยู่ในตำแหน่ง Manual
2. ต่อดังตามรูปที่ 3.6 เป็นขั้นที่ 1 คือ สัญญาณควบคุมจากแหล่งจ่ายไฟตรงภายนอก



รูปที่ 3.6 แสดงการต่อสัญญาณควบคุมจากแหล่งจ่ายไฟตรงจากภายนอก

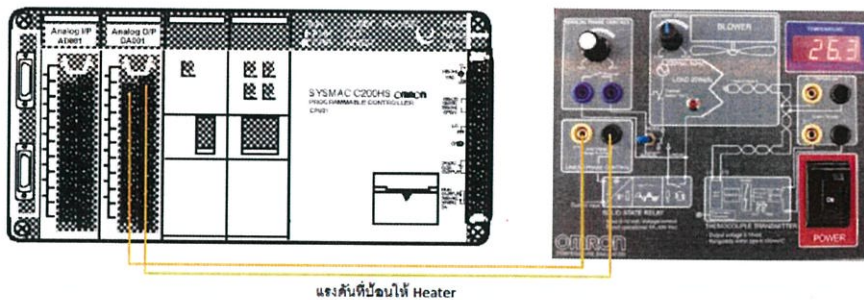
3. ตรวจสอบความถูกต้องของวงจร
4. ปรับแรงดันที่ป้อนให้กับ Heater ไปที่ค่าต่างๆดังนี้ 0, 0.1, 0.2, 0.3, ... ตามลำดับพร้อมทั้งบันทึกผลการทดลองลงในตาราง

ตารางที่ 3.3

No.	แรงดันที่ป้อนให้ Heater	DM200		อุณหภูมิจาก Thermocouple
		BCD	HEX	
1				
2				
3				
4				
5				
6				
7				
8				
9				
10				
11				
12				
13				
14				
15				

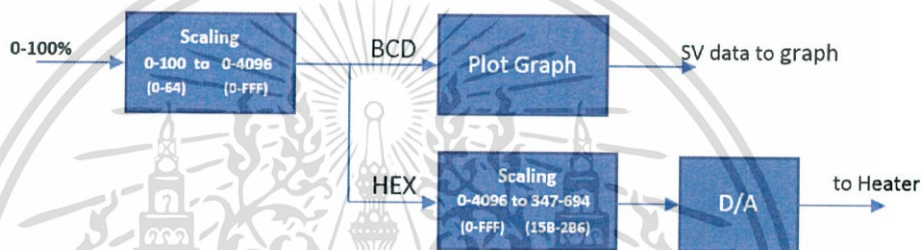
5. ในขั้นตอนที่ 2 ต่อวงจรตามรูปที่ 3.7 ให้กับ Heater ได้รับสัญญาณควบคุมจาก D/A Analog Output Unit โดยตรง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



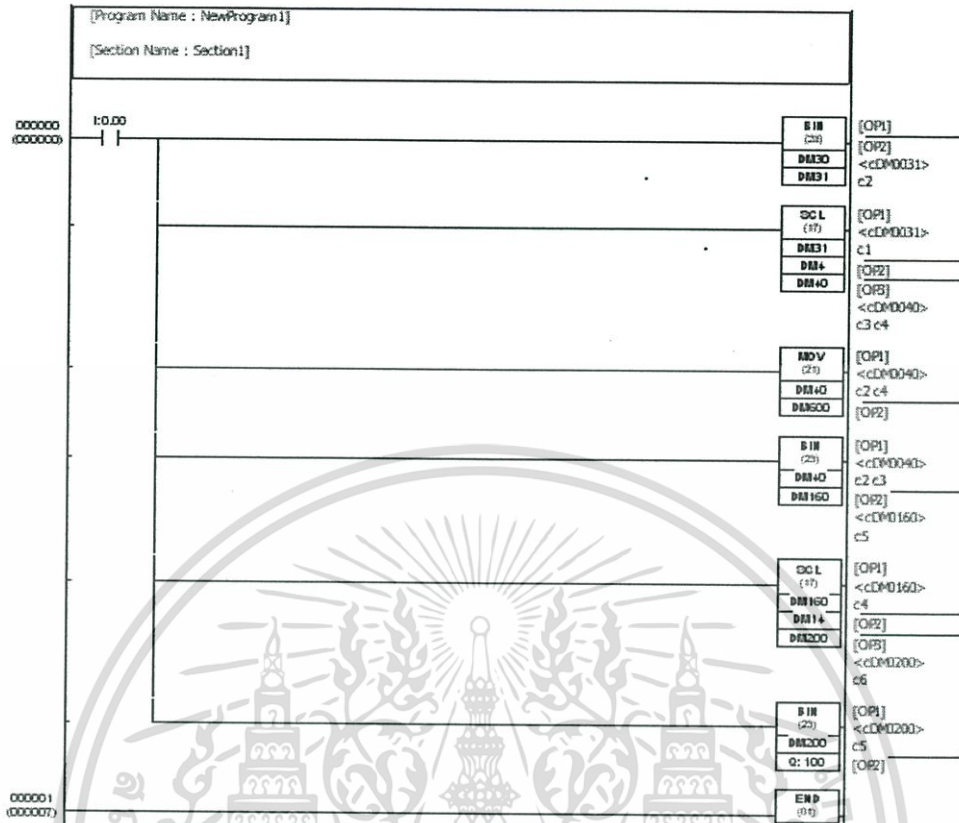
รูปที่ 3.7 แสดงการต่อสัญญาณควบคุมจาก D/A Analog Output

6. ลำดับการเขียนโปรแกรมให้ขั้นตอนดังรูปที่ 3.8 และเขียนโปรแกรมการทำงานดังรูปที่ 3.9



รูปที่ 3.8 แสดงลำดับและบล็อกคำสั่งการเขียนโปรแกรมกำหนด SV

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.9 แสดงโปรแกรม Ladder

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

7. เตรียมข้อมูลก่อนใช้คำสั่ง Scaling-SCL( ) กำหนดค่าเริ่มต้นให้กับ DM004-DM007,DM014-DM017 ในตารางที่ 3.4

ตารางที่ 3.4

หน่วยความจำ	ข้อมูล	ค่าตัวแปรคำสั่ง SCL( )
DM004		P1 (ค่า Min ในแนวแกน Y) เป็น BCD (Ay)
DM005		P1+1 (ค่า Min ในแนวแกน X) เป็น HEX (Ax)
DM006		P1+2 (ค่า Max ในแนวแกน Y) เป็น BCD (By)
DM007		P1+3 (ค่า Max ในแนวแกน X) เป็น HEX (Bx)
DM014		P2 (ค่า Min ในแนวแกน Y) เป็น BCD (Ay)
DM015		P2+1 (ค่า Min ในแนวแกน X) เป็น HEX (Ax)
DM016		P2+2 (ค่า Max ในแนวแกน Y) เป็น BCD (By)
DM017		P2+3 (ค่า Max ในแนวแกน X) เป็น HEX (Bx)

8. ทำการดาวน์โหลดโปรแกรมให้กับเครื่องควบคุม PLC
9. ตรวจสอบความถูกต้องของวงจรพร้อมเปลี่ยนโหมดสวิตช์ของ Temperature Simulator มาอยู่ในตำแหน่ง Linear ขณะเดียวกัน “ON” สวิตช์อินพุต 0.00 ให้กับ PLC พร้อมทั้ง RUN โปรแกรมทำงาน
10. ให้เปลี่ยนข้อมูลที่ DM030 มีค่าเป็นตามตารางที่ 3.5 ในขณะที่ PLC ทำงานในโหมด Monitor บันทึกผลการทดลอง ทั้งค่าแรงดันที่ป้อนให้ Heater, ข้อมูลที่ DM200, D/A(Wd100) และอุณหภูมิที่ได้จาก Thermocouple
11. ทำการ “OFF” สวิตช์อินพุต 0.00 ให้ PLC
12. นำผลการทดลองจากตารางไปเขียนกราฟความสัมพันธ์ระหว่างค่าแรงดันที่ป้อนให้ Heater ในแนวแกน X กับอุณหภูมิ ในแนวแกน Y
13. นำผลการทดลองจากตารางไปเขียนกราฟความสัมพันธ์ระหว่างค่าแรงดันที่ป้อนให้ Heater ในแนวแกน X กับข้อมูลใน DM200 ในแนวแกน Y

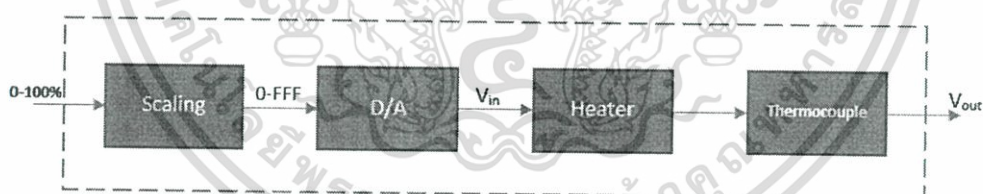
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3.5

No.	%, DM030	แรงดันที่ป้อนให้ Heater	อุณหภูมิ °C	DM200	Wd100
				BCD	HEX
1	0				
2	10				
3	20				
4	30				
5	40				
6	50				
7	60				
8	70				
9	80				
10	90				
11	100				

### 3.4 ศึกษาผลตอบสนองการเคลื่อนที่แบบสแต็ปวงรอบเปิด

เมื่อทำการห่างานทั้งตัวแปรอินพุท และตัวแปรเอาต์พุทของ Temperature Simulator ในขั้นต่อไปจะพิจารณาอุณหภูมิที่เปลี่ยนแปลงของ Thermocouple ที่ต้องการควบคุมเป็นหลัก ดังนั้นย่านที่เหมาะสมต่อการใช้งานของอุณหภูมิ โดยมากแล้วจะพิจารณาค่าที่ประมาณ 50% ของค่าอุณหภูมิที่เปลี่ยนแปลงแบบเต็มย่าน (Full Scale) ในการทดสอบได้ทำการเปลี่ยนแปลงอินพุทแบบสแต็ปจาก 0% ไปเป็น 10, 20, 30, ..., 100% จากนั้นสังเกตผลตอบสนองของระบบ

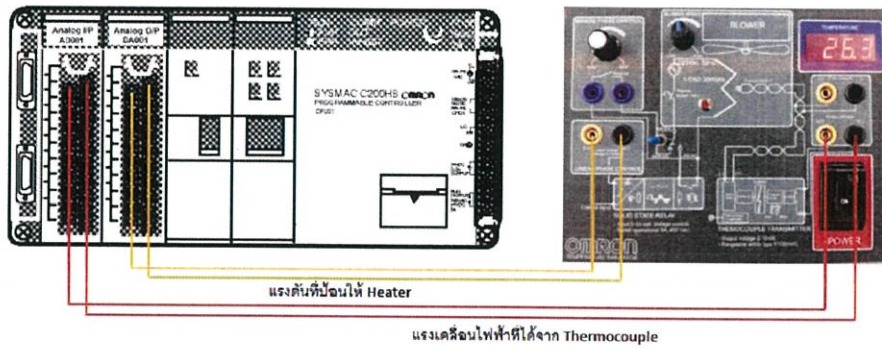


รูปที่ 3.10 แสดงกระบวนการสังเกตผลตอบสนองแบบสแต็ป

ลำดับขั้นตอนการทดลอง

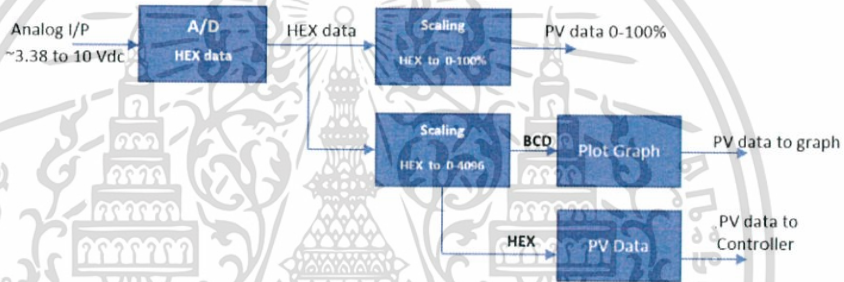
1. Temperature Simulator เลือกโหมดให้อยู่ในตำแหน่ง Manual
2. ต้องจรงตามรูปที่ 3.11

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



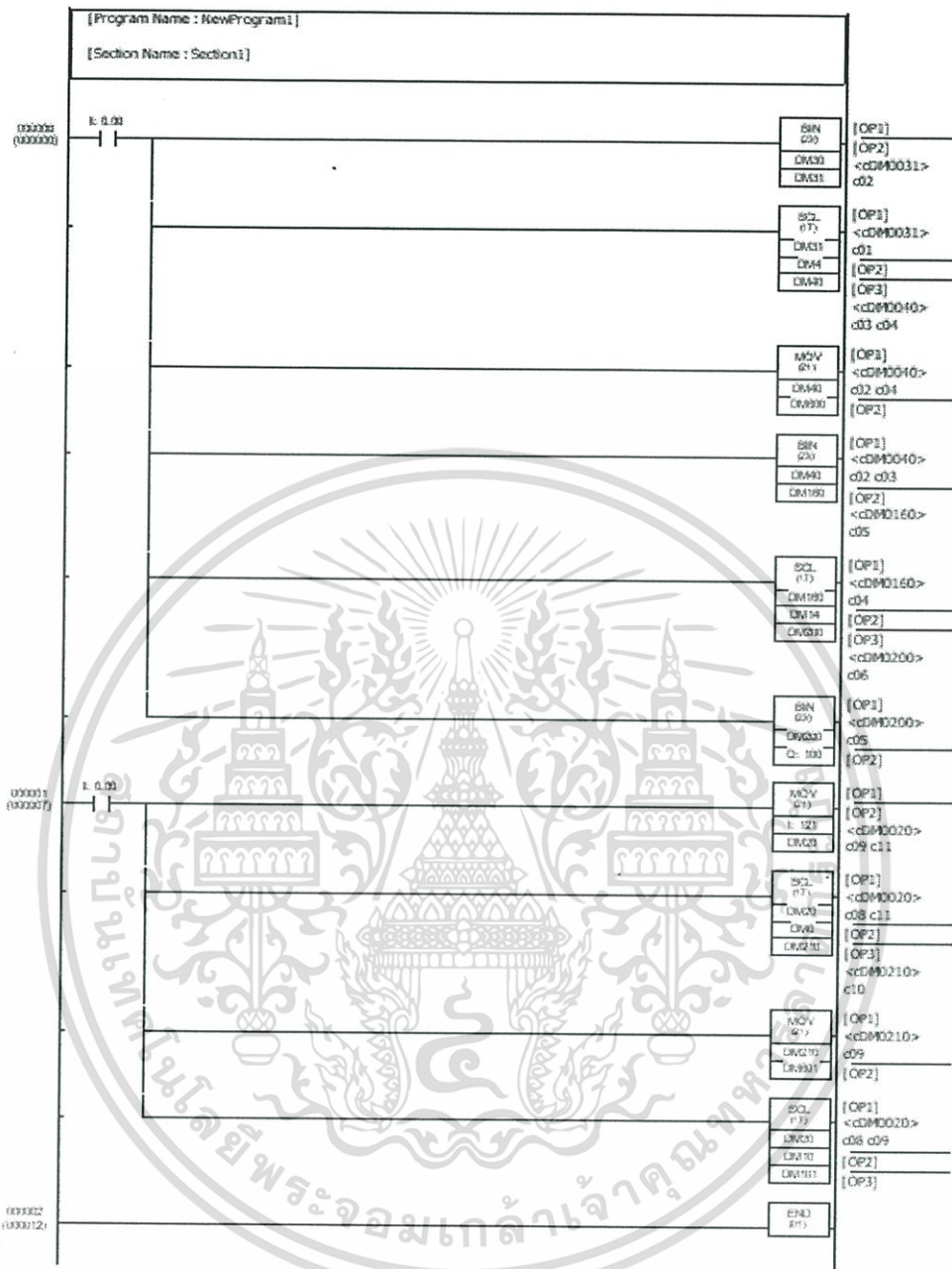
รูปที่ 3.11 การเชื่อมต่อสัญญาณระหว่าง Temperature Simulator กับเครื่องควบคุม PLC

- ลำดับการเขียนโปรแกรมดังรูปที่ 3.8, 3.12 และเขียนโปรแกรมการทำงานตามรูปที่ 3.13



รูปที่ 3.12 แสดงลำดับและบล็อกคำสั่งการเขียนโปรแกรมอ่านค่า PV

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.13 แสดงโปรแกรม Ladder

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. นำผลที่ได้จากการดำเนินการที่ 3.2 และ 3.3 มาปรับย่านแกน Y ใหม่เพื่อให้สอดคล้องกับการนำผลไป Plot graph ที่ Full Scale 4096<sub>(HEX)</sub> ใส่ข้อมูลลงใน DM000-DM007 และ DM010-DM017

จากตารางที่ 3.1

หน่วยความจำ	ข้อมูล	ค่าตัวแปรคำสั่ง SCL( )
DM000		P1 (ค่า Min ในแนวแกน Y) หรือ Ay
DM001		P1+1 (ค่า Min ในแนวแกน X) หรือ Ax
DM002		P1+2 (ค่า Max ในแนวแกน Y) หรือ By
DM003		P1+3 (ค่า Max ในแนวแกน X) หรือ Bx
DM010		P2 (ค่า Min ในแนวแกน Y) หรือ Ay
DM011		P2+1 (ค่า Min ในแนวแกน X) หรือ Ax
DM012		P2+2 (ค่า Max ในแนวแกน Y) หรือ By
DM013		P2+3 (ค่า Max ในแนวแกน X) หรือ Bx

จากตารางที่ 3.4

หน่วยความจำ	ข้อมูล	ค่าตัวแปรคำสั่ง SCL( )
DM004		P3 (ค่า Min ในแนวแกน Y) เป็น BCD (Ay)
DM005		P3+1 (ค่า Min ในแนวแกน X) เป็น HEX (Ax)
DM006		P3+2 (ค่า Max ในแนวแกน Y) เป็น BCD (By)
DM007		P3+3 (ค่า Max ในแนวแกน X) เป็น HEX (Bx)
DM014		P4 (ค่า Min ในแนวแกน Y) เป็น BCD (Ay)
DM015		P4+1 (ค่า Min ในแนวแกน X) เป็น HEX (Ax)
DM016		P4+2 (ค่า Max ในแนวแกน Y) เป็น BCD (By)
DM017		P4+3 (ค่า Max ในแนวแกน X) เป็น HEX (Bx)

5. ทำการดาวน์โหลดโปรแกรมให้กับเครื่องควบคุม PLC
6. ตรวจสอบความถูกต้องของวงจรพร้อมเปลี่ยนโหมดสวิทช์ให้อยู่ในตำแหน่ง Linear ขณะเดียวกันก็ “ON” สวิทช์อินพุต 0.00 เพื่อควบคุมการทำงานของเอาต์พุต พร้อมทั้ง Run โปรแกรมการทำงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

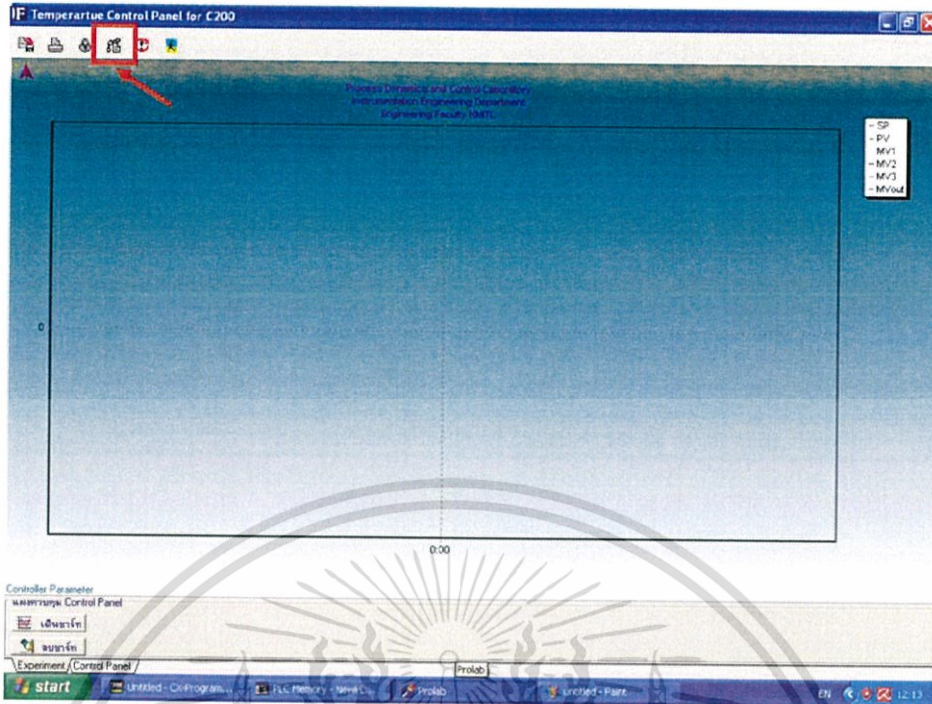
7. ทำการเปลี่ยนข้อมูลที่ DM030 มีค่าเป็นไปตามตารางที่ 3.8 ในขณะที่ PLC ทำงานในโหมด Monitor บันทึกผลการทดลอง ข้อมูลที่ DM040, Wd121, Wd100, DM101 และอุณหภูมิที่เปลี่ยนแปลงของ Thermocouple

ตารางที่ 3.8

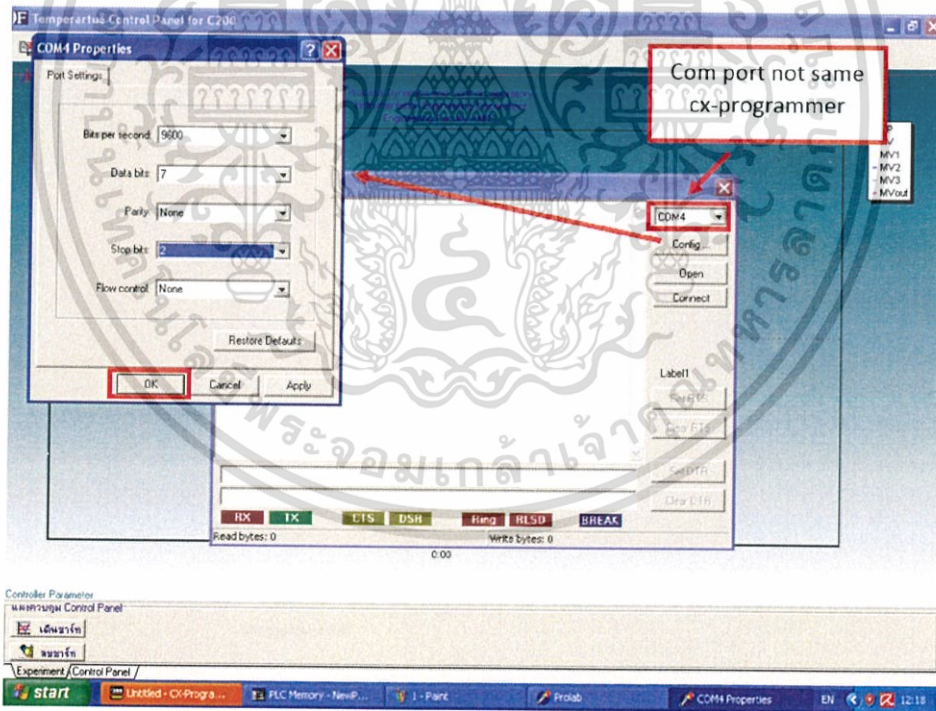
No.	DM030	DM040	Wd121		Wd100		DM101	อุณหภูมิ °C
	%		HEX	BCD (ค่านวน)	HEX	BCD (ค่านวน)	BCD	
1	0							
2	10							
3	20							
4	30							
5	40							
6	50							
7	60							
8	70							
9	80							
10	90							
11	100							

8. ทำการบันทึก Trend Graph ของสเต็ปอินพุท DM40(SV), DM210(PV) และผลตอบสนองของสเต็ปเอาต์พุท Wd100
- 8.1 ทำการบันทึกบนโปรแกรม ProLab (Temperature Control Panel for C200)
- 8.2 ทำการตั้งค่า communication โดยกำหนดค่า Port Setting ให้ตรงกับ Network setting ของ CX-Programmer ดังรูปที่ 3.14 และรูปที่ 3.15

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



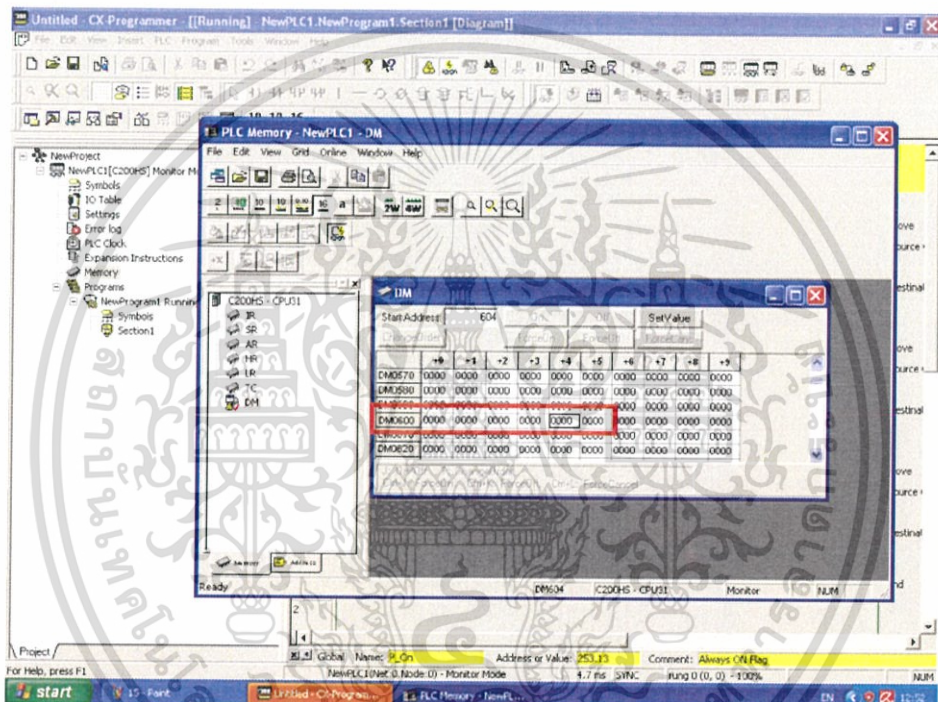
รูปที่ 3.14 แสดงหน้าโปรแกรม Prolab



รูปที่ 3.15 แสดงการ Configuration / Port setting

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- 8.3) เมื่อทำการตั้งค่าต่างๆแล้ว คลิกที่ “Open” และ “Connect”
- 8.4) เริ่มการ Trend Graph โดยการคลิกที่ “Online”
- 8.5) ค่าที่ต้องการ Plot Graph จะอยู่ที่ DM600-605
  - DM600 (SV) แสดงเส้นกราฟสีแดง
  - DM601 (PV) แสดงเส้นกราฟสีเขียว
  - DM602 (MV1) แสดงเส้นกราฟสีเหลือง
  - DM603 (MV2) แสดงเส้นกราฟสีน้ำเงิน
  - DM604 (MV3) แสดงเส้นกราฟสีน้ำตาล
  - DM605 (MV out) แสดงเส้นกราฟสีชมพู



รูปที่ 3.16 แสดงตำแหน่งของ Data Memory ที่ใช้ในการ Plot Graph

9. กำหนดข้อมูลให้ DM030 โดยเลือกจากเมนู Memory / DM และโหมด Monitor เพื่อเปลี่ยนข้อมูลได้ขณะที่เครื่องควบคุม PLC ยัง Online อยู่
10. หยุดการบันทึกโดยการคลิกที่ “Online” อีกครั้ง
11. ทำการ “OFF” สวิตช์อินพุต 0.00 ให้ PLC

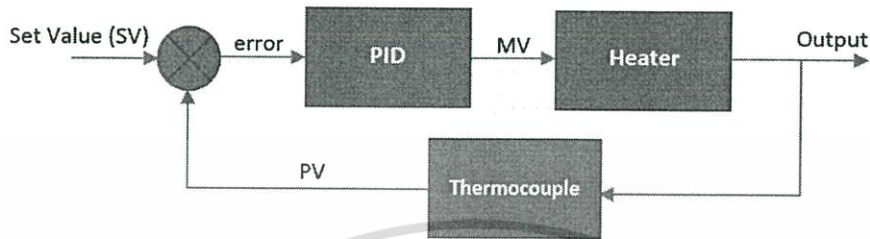
### 3.5 ศึกษาผลตอบสนองการเคลื่อนที่แบบสแต็ปวงรอบปิดด้วย PID

จากการดำเนินการที่ 3.4 เป็นการควบคุมแบบวงรอบเปิดจะให้ผลตอบสนองต่อระบบในลักษณะหนึ่ง สำหรับการควบคุมแบบวงรอบปิดที่มีฟังก์ชันควบคุมแบบ PID จะมีความยืดหยุ่นสูงในเรื่องการปรับค่าตัวแปรการควบคุม ทั้งนี้เพื่อให้ได้ผลตอบสนองที่ดีกับกระบวนการทั้งในช่วงเริ่มต้นและช่วงสภาวะคงที่ ดังรูปที่ 3.17

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ส่วนประกอบสำคัญในการควบคุมแบบวงรอบปิด มีดังนี้

- เครื่องควบคุม PLC ควบคุมด้วยฟังก์ชัน PID
- Heater (อยู่ใน Temperature Simulator)
- Thermocouple (อยู่ใน Temperature Simulator)



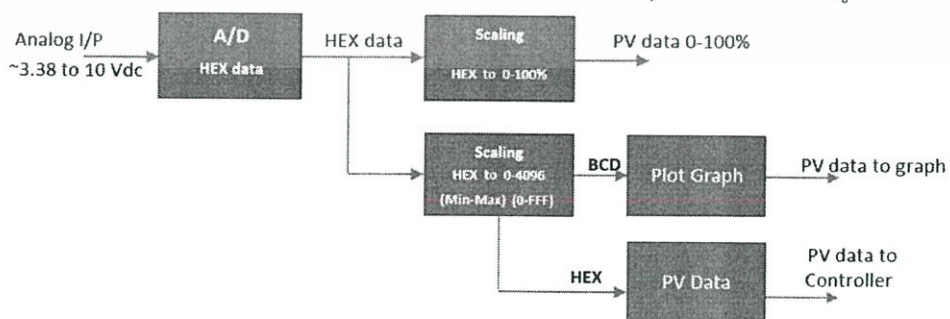
รูปที่ 3.17 แสดงบล็อกไดอะแกรมและกระบวนการควบคุมแบบวงรอบปิด

สำหรับกระบวนการ ค่า SV เป็นค่าเป้าหมายที่กำหนดได้โดยตรงในหน่วยความจำของเครื่องควบคุม มีย่านการควบคุมตั้งแต่ 0-100% โดยมีแนวทางในการเขียนโปรแกรมควบคุมให้กับ PLC ตามรูปที่ 3.18



รูปที่ 3.18 แสดงลำดับและบล็อกคำสั่งการเขียนโปรแกรมกำหนด SV

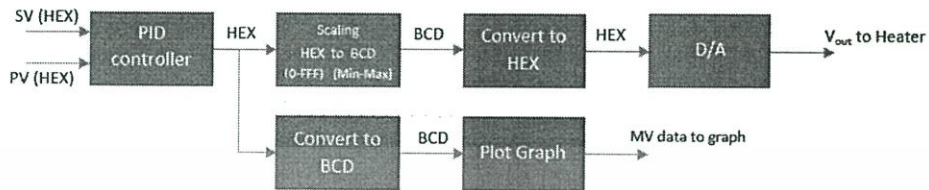
ส่วนค่า PV เป็นค่าตัวแปรที่ได้จากการวัดกระบวนการผ่านเข้ามาทาง A/D Analog Input Unit จำเป็นต้องมีการปรับแต่งค่าให้อยู่ในย่านเดียวกันกับค่า SV เพื่อให้เครื่องควบคุม PLC นำผลไปคำนวณได้ต่อไป โดยมีแนวทางในการเขียนโปรแกรมควบคุมให้กับ PLC ตามรูปที่ 3.19



รูปที่ 3.19 แสดงลำดับและบล็อกคำสั่งการเขียนโปรแกรมอ่านค่า PV

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

และค่า MV เป็นค่าที่ได้จากการประมวลผลของเครื่องควบคุม PLC ผ่านฟังก์ชัน PID เพื่อนำผลลัพธ์ที่ได้ออกไปยังส่วน D/A Analog Output Unit สำหรับป้อนแรงดันไฟฟ้าให้กับ Heater ในลักษณะเดียวกันต้องมีการปรับแต่งย่านการควบคุมให้ถูกต้อง โดยมีแนวทางในการเขียนโปรแกรมควบคุมให้กับ PLC ตามรูปที่ 3.20

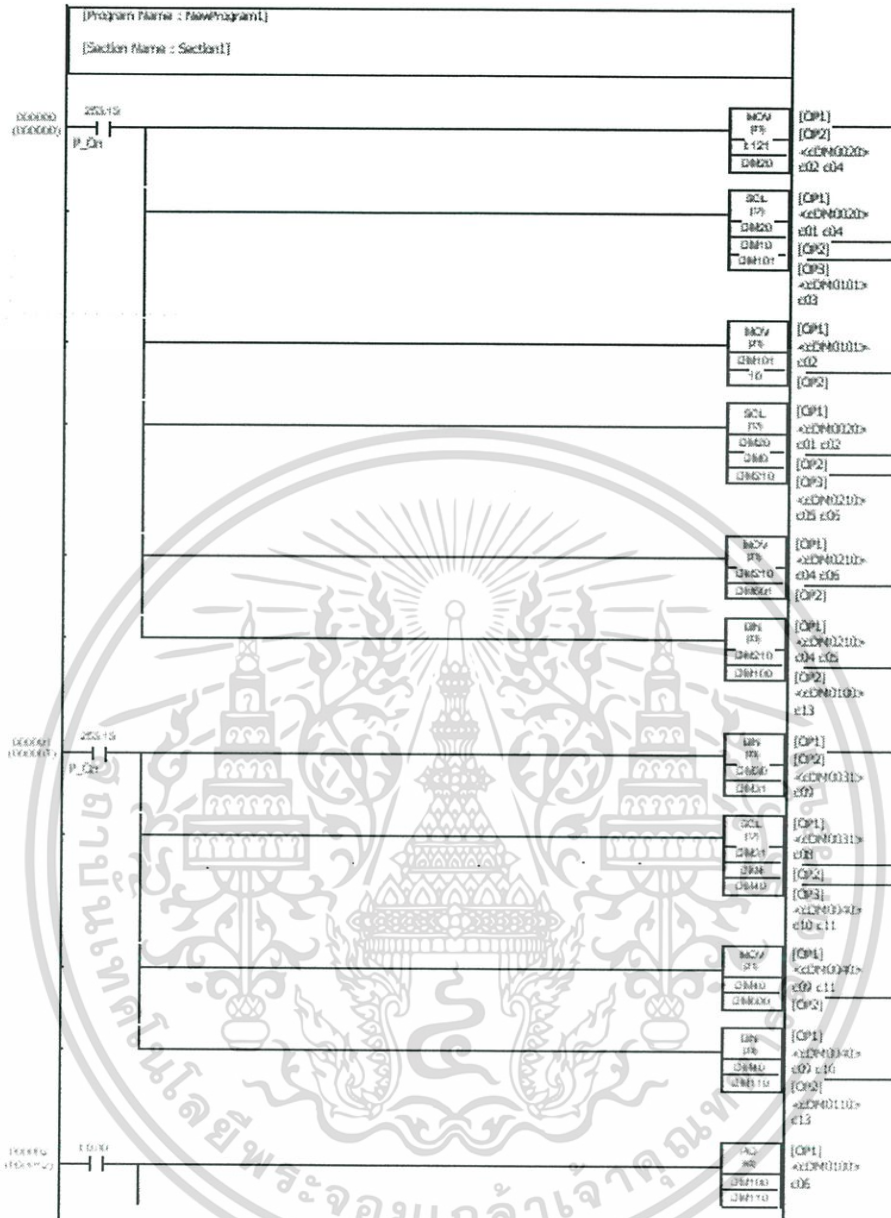


รูปที่ 3.20 แสดงลำดับและบล็อกคำสั่งการเขียนโปรแกรมให้ค่า MV

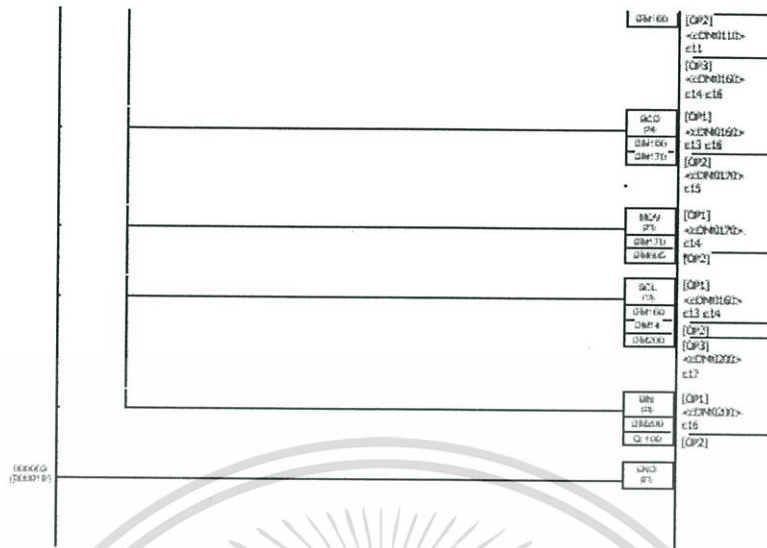
#### ลำดับขั้นการทดลอง

1. Temperature Simulator เลือกสวิตช์โหมด อยู่ในตำแหน่ง Manual
2. ต่อดำเนินการตามรูปที่ 3.11
3. เขียนโปรแกรมการทำงานตามรูปที่ 3.21

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.21 แสดงโปรแกรม Ladder

- ทำการ Online ในพื้นที่หน่วยความจำ DM เพื่อเตรียมการแก้ไข
- นำผลที่ได้จากการดำเนินการที่ 3.2 และ 3.3 มาปรับย่านแกน Y ใหม่เพื่อให้สอดคล้องกับการนำผลไป Plot graph ที่ Full Scale 4096(HEX) ใส่ข้อมูลลงใน DM000-DM007 และ DM010-DM017

จากตารางที่ 3.1

หน่วยความจำ	ข้อมูล	ค่าตัวแปรคำสั่ง SCL( )
DM000		P1 (ค่า Min ในแนวแกน Y) หรือ Ay
DM001		P1+1 (ค่า Min ในแนวแกน X) หรือ Ax
DM002		P1+2 (ค่า Max ในแนวแกน Y) หรือ By
DM003		P1+3 (ค่า Max ในแนวแกน X) หรือ Bx
DM010		P2 (ค่า Min ในแนวแกน Y) หรือ Ay
DM011		P2+1 (ค่า Min ในแนวแกน X) หรือ Ax
DM012		P2+2 (ค่า Max ในแนวแกน Y) หรือ By
DM013		P2+3 (ค่า Max ในแนวแกน X) หรือ Bx

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## จากตารางที่ 3.4

หน่วยความจำ	ข้อมูล	ค่าตัวแปรคำสั่ง SCL( )
DM004		P3 (ค่า Min ในแนวแกน Y) เป็น BCD (Ay)
DM005		P3+1 (ค่า Min ในแนวแกน X) เป็น HEX (Ax)
DM006		P3+2 (ค่า Max ในแนวแกน Y) เป็น BCD (By)
DM007		P3+3 (ค่า Max ในแนวแกน X) เป็น HEX (Bx)
DM014		P4 (ค่า Min ในแนวแกน Y) เป็น BCD (Ay)
DM015		P4+1 (ค่า Min ในแนวแกน X) เป็น HEX (Ax)
DM016		P4+2 (ค่า Max ในแนวแกน Y) เป็น BCD (By)
DM017		P4+3 (ค่า Max ในแนวแกน X) เป็น HEX (Bx)

6. กำหนดค่าข้อมูลตัวแปร PID ลงใน DM110-DM116
7. ทำการดาวน์โหลดหน่วยความจำ DM และ โปรแกรมการทำงานให้กับเครื่องควบคุม PLC
8. ตรวจสอบความถูกต้องของวงจรพร้อมเปลี่ยนโหมดสวิตซ์ให้อยู่ในตำแหน่ง Linear ขณะเดียวกันก็ “ON” สวิตซ์อินพุต 0.00 เพื่อควบคุมการทำงานของเอาต์พุต พร้อมทั้ง Run โปรแกรมการทำงาน
9. ทดลองการทำงานของคำสั่ง PID สังเกตค่าตัวแปร SV, PV และ MV โดยเปลี่ยนข้อมูลที่ DM030 (SV) เป็นแบบสลับจาก 0, 20, 40, 60 % สังเกตการณ์เปลี่ยนแปลงของ DM210 (PV) จะตามหลังและไล่ตามมาจนเท่ากัน
10. ทำการบันทึก Trend Graph ของสแต็ปอินพุต DM040 (SV) และ DM210 (PV)
  - DM040 (SV) แสดงเส้นกราฟสีแดง
  - DM210 (PV) แสดงเส้นกราฟสีเขียว
11. เปลี่ยนโหมดสวิตซ์ของ Temperature Simulator ให้อยู่ในตำแหน่ง Manual ขณะเดียวกัน “OFF” สวิตซ์อินพุต 0.00 ให้ PLC
12. ทำการเปลี่ยนค่าตัวแปรใหม่ให้กับคำสั่ง PID ดังนี้
  - 12.1 เปลี่ยนค่า P ค่าใดๆ ตั้งแต่ค่ามากไปหาค่าน้อย ในขณะที่ Ti และ Td ยังมีค่าเท่าเดิม จนได้ผลตอบสนองที่ดีที่สุด แล้วบันทึกกราฟผลทดลองของ SV, PV และ MV

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

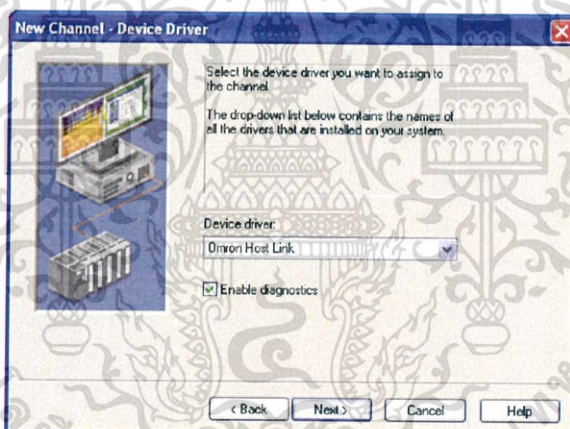
12.2 เพิ่มค่า Ti ใดๆ เพื่อทำการลดโอเวอร์ชุตของกระบวนการ หรือลดค่า Ti ถ้าผลตอบสนองช้า จนได้ผลตอบสนองที่เหมาะสมกับกระบวนการ แล้วบันทึกกราฟผลทดลองของ SV, PV และ MV

### 3.6 ตั้งค่าโปรแกรมสำหรับการเชื่อมต่อ OPC Server และ SCADA

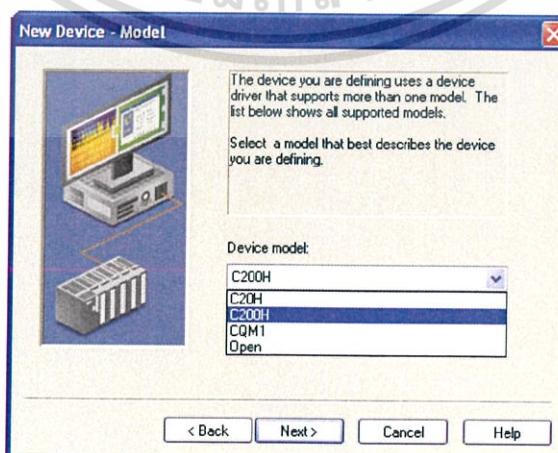
โปรแกรมสำหรับการเชื่อมต่อ OPC Server เลือกใช้โปรแกรม KEPServerEx ซึ่งมีประโยชน์ในการแลกเปลี่ยนข้อมูลกันระหว่าง SCADA พีแอลซี คอมพิวเตอร์ หรืออุปกรณ์อื่นๆ สามารถทำการอ่านและเขียนค่าใน Memory Area ของพีแอลซี โดยก่อนเริ่มใช้งานต้องมีการกำหนดโครงสร้างของฮาร์ดแวร์ที่เชื่อมต่อและสร้าง Tag ในแอดเดรสที่เราต้องการมอนิเตอร์ค่า

ในส่วนของการเขียนกราฟิก SCADA เลือกใช้โปรแกรม WinCC (Window Control Center) ซึ่งเป็นซอฟต์แวร์ที่ใช้ในการควบคุมงานทางด้านการผลิตและกระบวนการอัตโนมัติ ผ่านหน้าจอคอมพิวเตอร์ ภายในซอฟต์แวร์ประกอบโมดูลที่ใช้จัดการทางด้านกราฟิก แมสเสจไฟล์ข้อมูล และการทำรายงาน ซึ่งเหมาะสมสำหรับงานทางด้านอุตสาหกรรมการผลิตต่างๆทั่วไป ลำดับการทดลอง

1. ทำการสร้าง Channel และ Device ให้ตรงตามรุ่นของพีแอลซีเลือกใช้



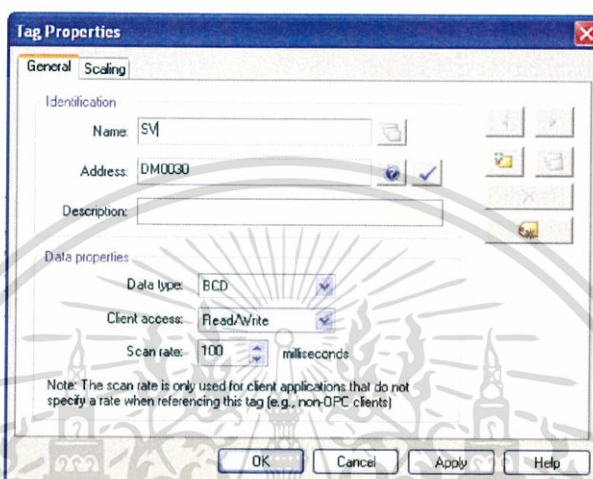
รูปที่ 3.22 แสดงการเลือกชนิดของ Device driver



รูปที่ 3.23 แสดงการเลือก Device model

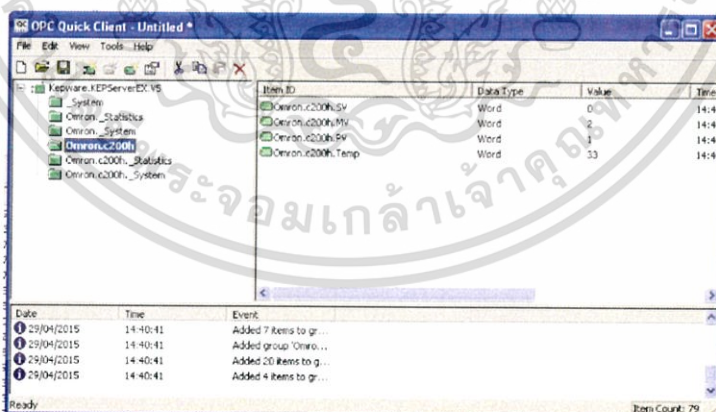
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. เพิ่ม Static Tag ในแอดเดรสที่เราต้องการอ่านหรือเขียนค่า ซึ่งค่า Tag ที่ต้องการอ่านและเขียนมีดังนี้
  - เขียนค่า SV ในตำแหน่งแอดเดรส DM0030 ในรูปของ BCD
  - อ่านค่า PV ในตำแหน่งแอดเดรส DM0101 ในรูปของ BCD
  - อ่านค่า MV ในตำแหน่งแอดเดรส DM0050 ในรูปของ BCD
  - อ่านค่าอุณหภูมิในตำแหน่งแอดเดรส DM0022 ในรูปของ BCD



รูปที่ 3.24 แสดงการตั้งค่า Tag Properties

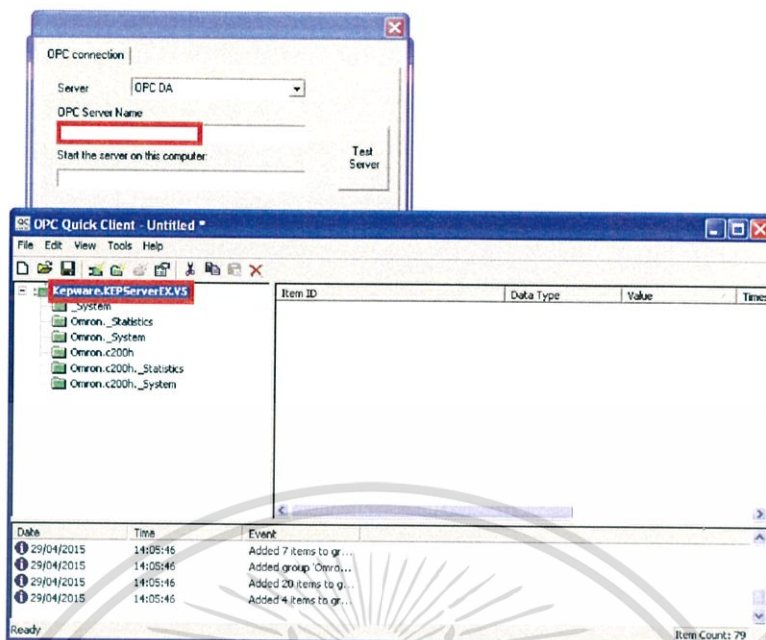
3. ทำการดูค่าของ tag ตำแหน่งแอดเดรสที่เพิ่มเข้ามา โดยการกดปุ่ม quick client แล้วเลือกที่ชื่อและรุ่นของอุปกรณ์



รูปที่ 3.25 แสดงการดูค่าของ tag ในแอดเดรสที่เราต้องการอ่านหรือเขียนค่า

4. เชื่อมต่อกับกราฟฟิก SCADA ใน WinCC โดยทำการเพิ่ม Driver OPC.chm ที่ Tag Management
5. ทำการเพิ่ม Driver Connection แล้วใส่ค่า OPC Server Name ให้ตรงกับ ในโปรแกรม KEPServer

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

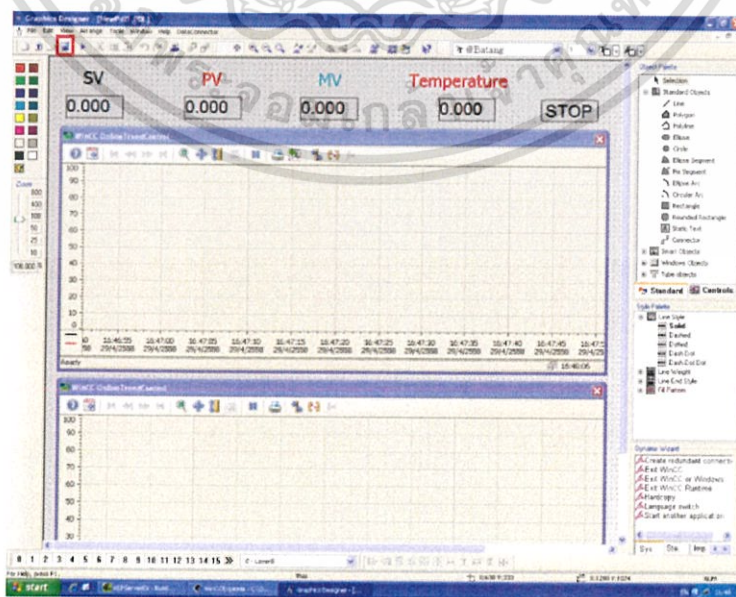


รูปที่ 3.26 แสดงการตั้งค่า OPC connection

6. เพิ่ม Tag ที่ต้องการแสดงค่าตามที่เลือกไว้บน KEP server

### 3.7 เขียนกราฟฟิกสำหรับการแสดงผล

การเขียนกราฟิกนั้นจะเริ่มจากการใช้งานหน้าต่าง Graphics Designer ซึ่งเป็นส่วนประกอบของโปรแกรม WinCC ที่ใช้สำหรับออกแบบหน้าจอที่เป็นตัวกลางระหว่างผู้ใช้กับเครื่องจักร โดย Graphics Designer มีความสามารถในการสร้างรูปภาพหรือวัตถุที่มีความสามารถในการเข้าถึงข้อมูลจากภายนอก มีความสามารถในการเขียนโปรแกรมเพื่อจัดการกับข้อมูลที่ได้รับมา และยังสามารถนำข้อมูลที่รับมาแสดงผลในรูปของกราฟเพื่อให้มองเห็นความเป็นไปของข้อมูลได้ง่ายมากขึ้น



รูปที่ 3.27 แสดงหน้ากราฟฟิกบน WinCC ที่ต้องการแสดงผล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทที่ 4

### ผลการทดลอง

#### 4.1 ย่านควบคุมการทำงานด้านตัวแปรอินพุท

ตัวแปรอินพุทของกระบวนการ คือ ค่าแรงเคลื่อนไฟฟ้าที่ได้จาก Thermocouple จากการทดลองป้อนแรงดันไฟฟ้าจาก DC. Power Supply ให้กับ Heater ค่าอุณหภูมิและแรงเคลื่อนไฟฟ้าที่วัดได้จาก Thermocouple มีค่าเป็น 28.4-107.8 องศาเซลเซียส และ 2.853-10.79 โวลต์ ดังแสดงในตารางที่ 4.2 ซึ่งคำนวณย่านการเปลี่ยนแปลงของข้อมูลเข้า A/D Analog Input Unit ของเครื่องควบคุม PLC ได้เป็น 475-FA0

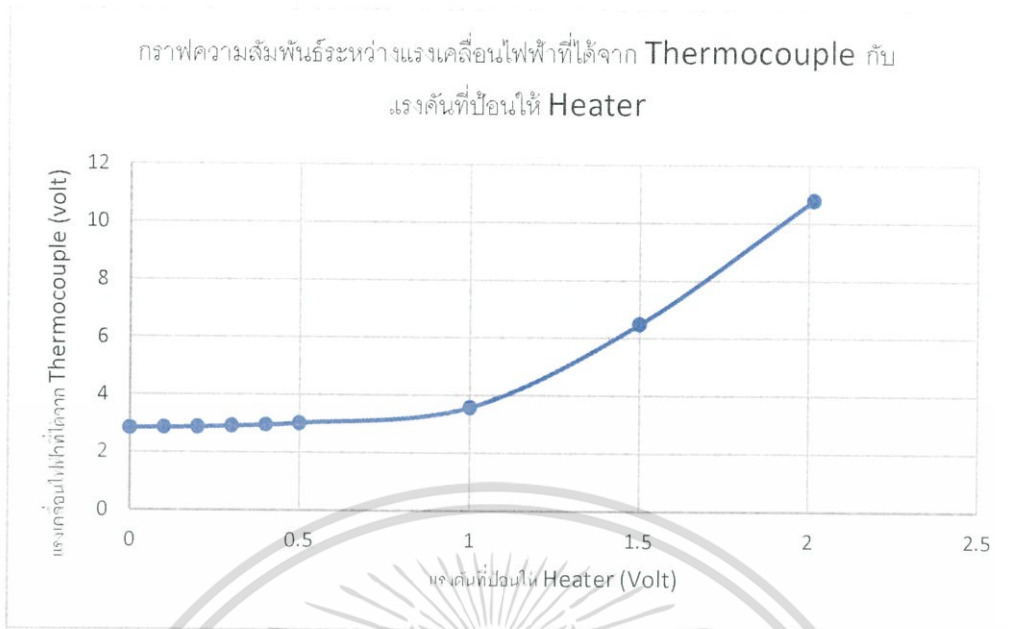
ตารางที่ 4.1 (นำผลจากตารางที่ 4.2 มาใส่เป็นค่าที่ได้จากการทดลอง)

หน่วยความจำ	ข้อมูล	ค่าตัวแปรคำสั่ง SCL(_)
DM000	0	P1 (ค่า Min ในแนวแกน Y) หรือ Ay
DM001	475	P1+1 (ค่า Min ในแนวแกน X) หรือ Ax
DM002	4096	P1+2 (ค่า Max ในแนวแกน Y) หรือ By
DM003	FA0	P1+3 (ค่า Max ในแนวแกน X) หรือ Bx

ตารางที่ 4.2

No.	แรงดันที่ป้อนให้ Heater	อุณหภูมิจาก Thermocouple	แรงเคลื่อนไฟฟ้าที่ได้จาก Thermocouple	A/D, Wd121 หรือ DM20	
				BCD	HEX
1	0.000	28.4	2.853	1141	475
2	0.101	28.6	2.872	1149	47D
3	0.200	28.8	2.890	1156	484
4	0.300	29.1	2.923	1169	491
5	0.401	29.5	2.962	1185	4A1
6	0.500	30.1	3.025	1210	4BA
7	1.001	35.7	3.585	1434	59A
8	1.499	64.5	6.487	2595	A23
9	2.013	107.8	10.79	4000	FA0

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.1 แสดงแรงเคลื่อนไฟฟ้าต่ำสุด-สูงสุดที่วัดได้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 4.2 ยานควบคุมการทำงานด้านตัวแปรเอาท์พุท

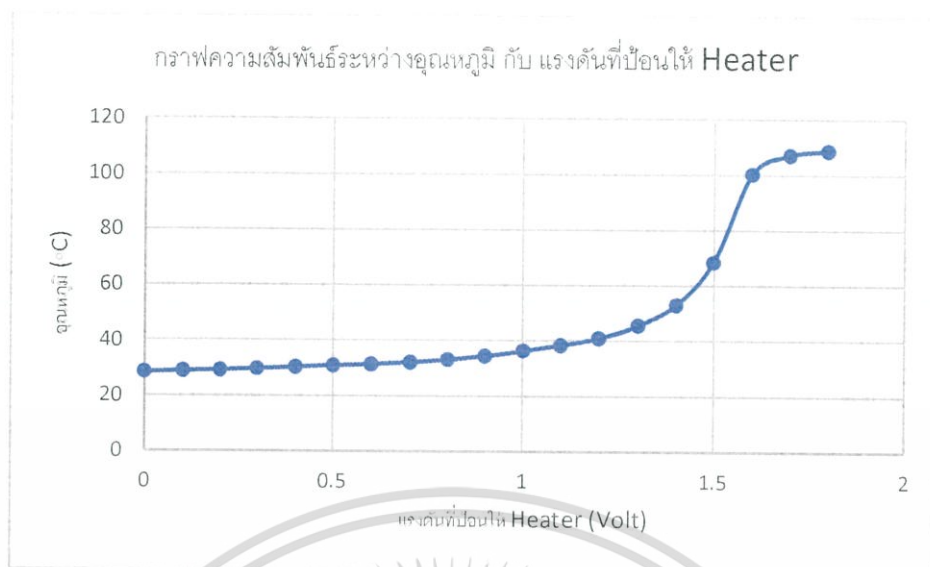
ตัวแปรเอาท์พุทของกระบวนการ คือ แรงดันที่ป้อนให้กับ Heater

ตอนที่ 1 การป้อนสัญญาณควบคุมจากแหล่งจ่ายไฟตรงภายนอก จากการป้อนแรงดัน สังเกตเห็นว่าช่วงที่ค่าอุณหภูมิเริ่มมีการเปลี่ยนแปลงชัดเจนที่ค่าแรงดันไฟฟ้าเท่ากับ 0.8 โวลต์ และมีค่าสูงสุดที่ค่าแรงดันไฟฟ้าเท่ากับ 1.6 โวลต์ (เนื่องจาก Voltage Input ของ A/D มีค่าเท่ากับ 0-10 V เพราะฉะนั้นจะวัดอุณหภูมิได้สูงสุดที่ประมาณ 100 องศาเซลเซียส ) แสดงดังรูปที่ 4.2

ตารางที่ 4.3

No.	แรงดันที่ป้อนให้ Heater	DM200		อุณหภูมิจาก Thermocouple
		BCD	HEX	
1	0.000	0	0	28.5
2	0.101	044	02C	28.9
3	0.200	087	057	29.2
4	0.299	130	082	29.7
5	0.401	174	0AE	30.3
6	0.500	217	0D9	30.9
7	0.599	260	104	31.4
8	0.701	304	130	32.2
9	0.799	347	15B	33.2
10	0.899	390	186	34.5
11	1.001	434	1B2	36.4
12	1.100	477	1DD	38.4
13	1.199	520	208	41.0
14	1.301	564	234	45.6
15	1.400	607	25F	53.1
16	1.499	650	28A	68.4
17	1.601	694	2B6	100.2
18	1.700	737	2E1	107.1
19	1.799	780	30C	108.7

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.2 แสดงการเปลี่ยนแปลงของอุณหภูมิเมื่อมีป้อนแรงดัน

ตอนที่ 2 สัญญาณควบคุมจากชุด D/A Analog Output Unit ค่าเริ่มต้นของ DM014-DM017 อ้างอิงจากข้อมูลที่ได้จากตารางที่ 4.3

ตารางที่ 4.4

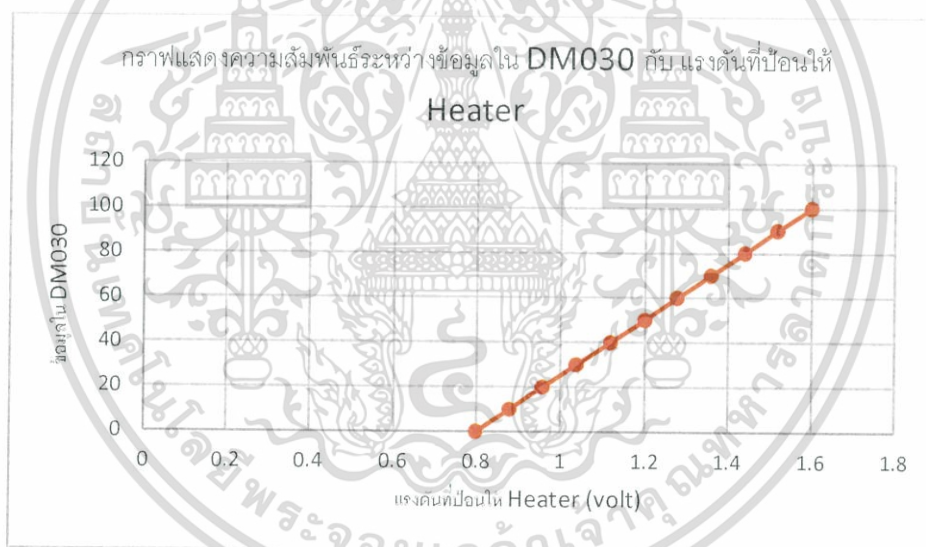
หน่วยความจำ	ข้อมูล	ค่าตัวแปรคำสั่ง SCL ( )
DM004	0	P1 (ค่า Min ในแนวแกน Y) เป็น BCD (Ay)
DM005	0	P1+1 (ค่า Min ในแนวแกน X) เป็น HEX (Ax)
DM006	4096	P1+2 (ค่า Max ในแนวแกน Y) เป็น BCD (By)
DM007	64	P1+3 (ค่า Max ในแนวแกน X) เป็น HEX (Bx)
DM014	347	P2 (ค่า Min ในแนวแกน Y) เป็น BCD (Ay)
DM015	0	P2+1 (ค่า Min ในแนวแกน X) เป็น HEX (Ax)
DM016	694	P2+2 (ค่า Max ในแนวแกน Y) เป็น BCD (By)
DM017	FFF	P2+3 (ค่า Max ในแนวแกน X) เป็น HEX (Bx)

การป้อนข้อมูลจากชุด D/A Analog Output Unit ผ่านโปรแกรม Ladder ที่ DM030 เป็น 0-100% เทียบกับค่าแรงดันที่ป้อนให้ Heater เป็น 0.799-1.601 โวลต์ จากผลการทดลองตารางที่ 4.5 แสดงดังกราฟที่ 4.3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.5

No.	%, DM030	แรงดันที่ป้อนให้ Heater	อุณหภูมิ °C	DM200	Wd100
				BCD	HEX
1	0	0.799	33.8	347	15B
2	10	0.880	35.0	382	17E
3	20	0.959	36.2	416	1A0
4	30	1.039	37.6	451	1C3
5	40	1.120	39.7	486	1E6
6	50	1.201	41.9	521	209
7	60	1.279	45.8	555	22B
8	70	1.360	51.9	590	24E
9	80	1.441	61.7	625	271
10	90	1.519	80.7	659	293
11	100	1.600	102.6	694	2B6



รูปที่ 4.3 แสดงความสัมพันธ์ระหว่างข้อมูลใน DM030 กับ แรงดันที่ป้อนให้ Heater

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 4.3 ผลตอบสนองการเคลื่อนที่แบบสแต็ปวงรอบเปิด

ขั้นตอนการพิจารณาอุณหภูมิของ Thermocouple ที่ต้องการควบคุมเป็นหลักโดยยังไม่ผ่านตัวควบคุม PID ทำการป้อนและรับค่าแรงดันจาก Analog I/O ของ PLC จะได้ความสัมพันธ์ของตัวแปรด้านอินพุทและเอาต์พุตดังรูปที่ 4.4

ตารางที่ 4.6

หน่วยความจำ	ข้อมูล	ค่าตัวแปรคำสั่ง SCL( )
DM000	0000	P1 (ค่า Min ในแนวแกน Y) หรือ Ay
DM001	0548	P1+1 (ค่า Min ในแนวแกน X) หรือ Ax
DM002	4096	P1+2 (ค่า Max ในแนวแกน Y) หรือ By
DM003	0FA0	P1+3 (ค่า Max ในแนวแกน X) หรือ Bx
DM010	0000	P2 (ค่า Min ในแนวแกน Y) หรือ Ay
DM011	0548	P2+1 (ค่า Min ในแนวแกน X) หรือ Ax
DM012	0100	P2+2 (ค่า Max ในแนวแกน Y) หรือ By
DM013	0FA0	P2+3 (ค่า Max ในแนวแกน X) หรือ Bx

ตารางที่ 4.7

No.	DM030	DM040	Wd121		Wd100		DM101	อุณหภูมิ °C
	%		HEX	BCD (จำนวน)	HEX	BCD (จำนวน)	BCD	
1	0	0	548	1352	15B	347	0	33.8
2	10	410	56C	1388	17E	382	2	34.7
3	20	819	5A0	1440	1A0	416	4	36.0
4	30	1229	5E0	1504	1C3	451	6	37.6
5	40	1638	638	1592	1E6	486	10	39.8
6	50	2048	6B4	1716	209	521	14	42.6
7	60	2458	764	1892	22B	555	21	47.3
8	70	2867	844	2116	24E	590	29	52.9
9	80	3277	980	2432	271	625	41	60.8
10	90	3686	C68	3176	293	659	68	79.4
11	100	4096	FA0	4000	2B6	694	100	101.1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.4 แสดงค่า PV ของระบบเมื่อยังไม่ผ่านตัวควบคุม PID

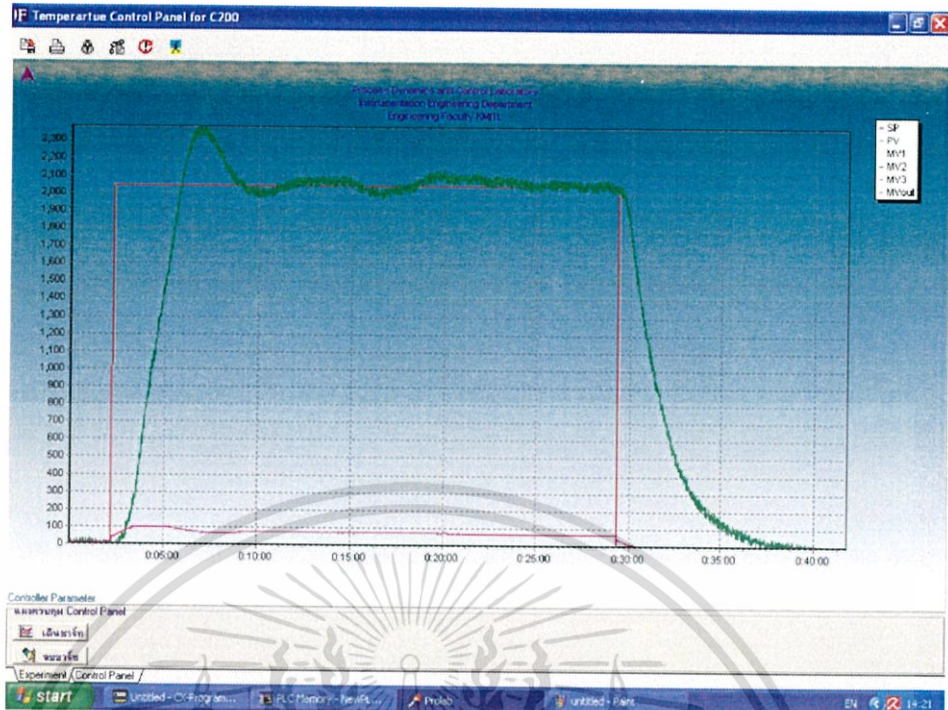
#### 4.4 ผลตอบสนองการเคลื่อนที่แบบสเต็ปวงรอบปิดด้วย PID

พิจารณาอุณหภูมิของ Thermocouple ที่ต้องการควบคุมเป็นหลักโดยผ่านตัวควบคุม PID ซึ่งจะใช้วิธีการปรับค่า PID Controller แบบ Trial & Error Close-Loop Tuning หรือวิธีการลองผิดลองถูก เริ่มต้นกำหนดค่าพารามิเตอร์ของ PID ใน DM110-DM116 เป็นดังนี้

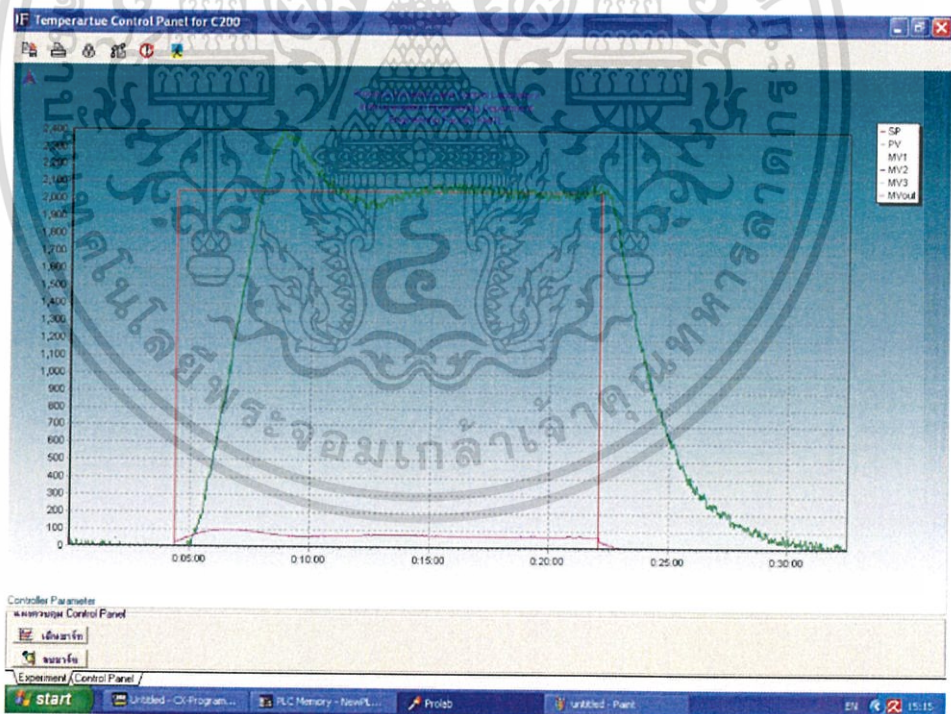
C	:	DM110	0000	Set Value (SV)
C+1	:	DM111	0800	Proportional Band (PB)
C+2	:	DM112	0000	Ti (Integral Constant)
C+3	:	DM113	0000	Td (Derivative Constant)
C+4	:	DM114	0032	Sampling Period
C+5	:	DM115	0000	Reverse Action
C+6	:	DM116	0404	I/O range 12 bit

เมื่อทำการเปลี่ยนค่า SV (Set Value) แบบสเต็ปจาก 0%, 50%, 0% โดยเลือกทดลองที่ค่า Proportional Band (PB) ที่ 800, 500, 200 และ 100 ตามลำดับลักษณะของกราฟของค่า SV และ PV ที่บันทึกแสดงดังรูปที่ 4.5-4.8

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

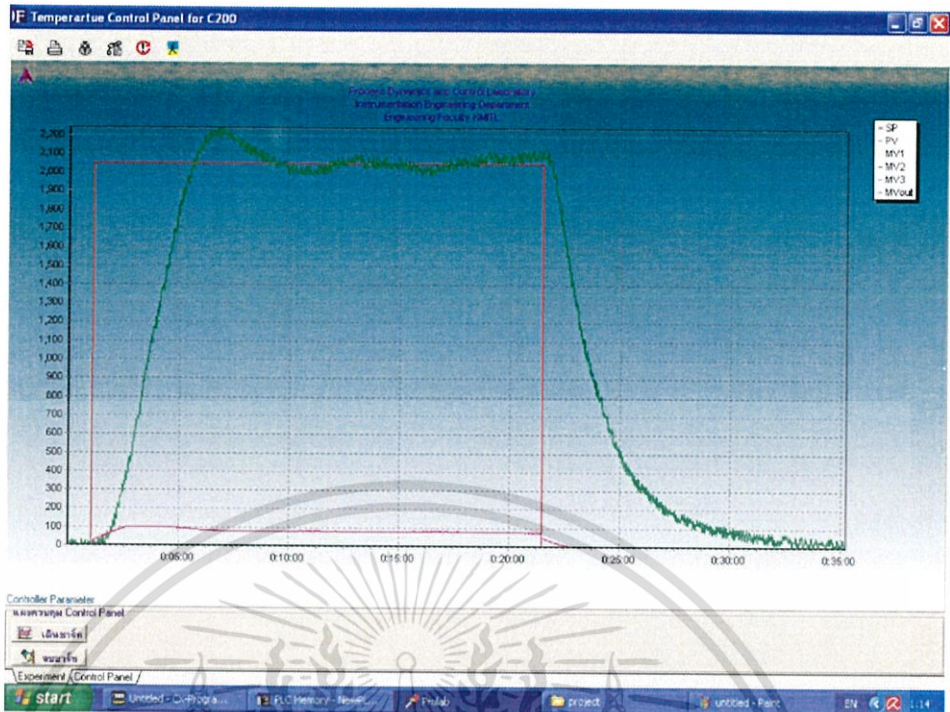


รูปที่ 4.5 การควบคุม PID เมื่อค่า PB, Ti, Td เท่ากับ 800, 0, 0 แบบสแต็ปจาก 0%, 50%, 0%

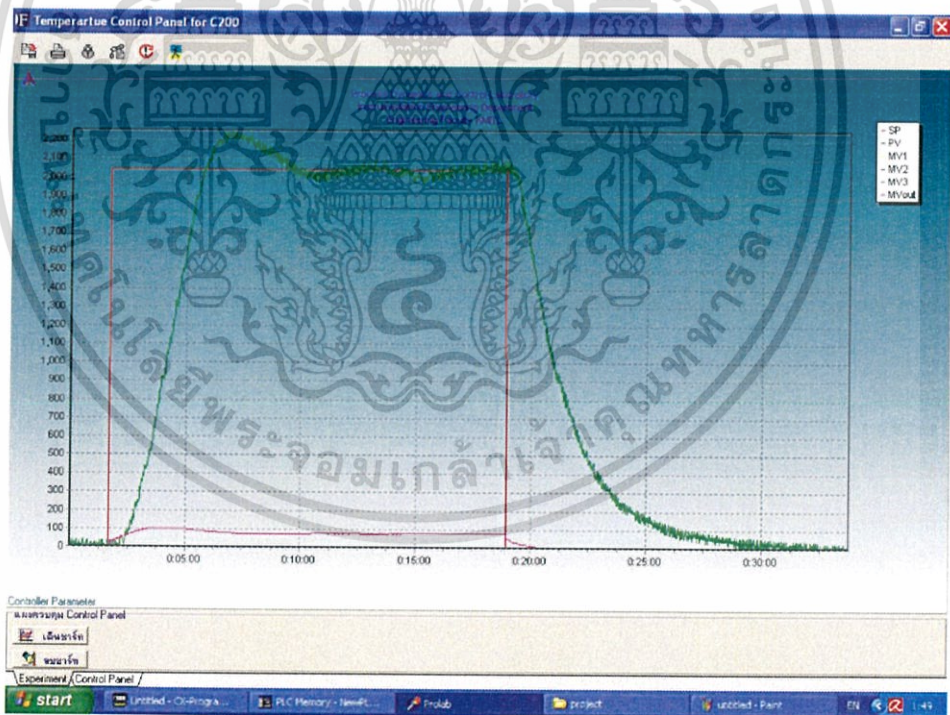


รูปที่ 4.6 การควบคุม PID เมื่อค่า PB, Ti, Td เท่ากับ 500, 0, 0 แบบสแต็ปจาก 0%, 50%, 0%

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



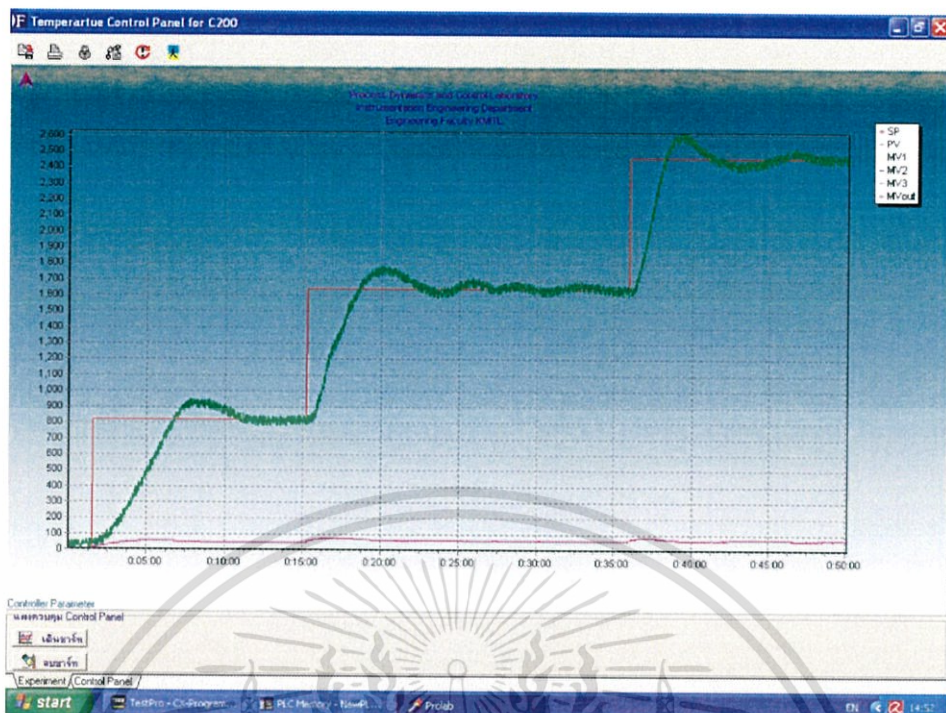
รูปที่ 4.7 การควบคุม PID เมื่อค่า PB, Ti, Td เท่ากับ 200, 0, 0 แบบสแต็ปจาก 0%, 50%, 0%



รูปที่ 4.8 การควบคุม PID เมื่อค่า PB, Ti, Td เท่ากับ 100, 0, 0 แบบสแต็ปจาก 0%, 50%, 0%

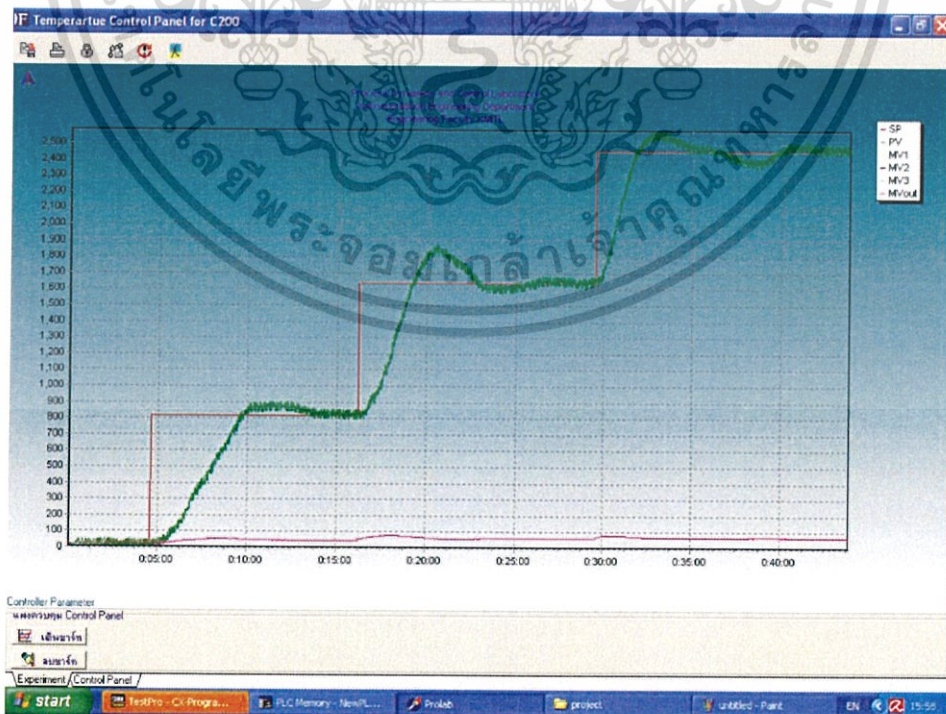
เมื่อทำการเปลี่ยนค่า SV (Set Value) แบบสแต็ปจาก 0, 20, 40, 60% ลักษณะของกราฟของค่า SV และ PV ที่บันทึกแสดงดังรูปที่ 4.9

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



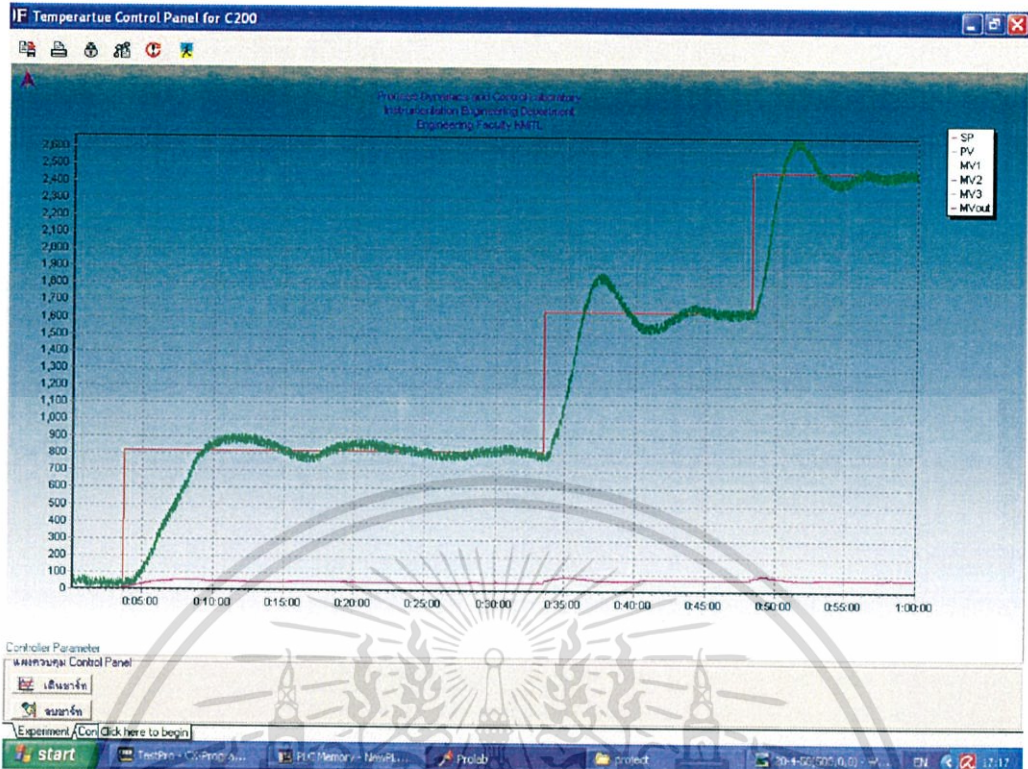
รูปที่ 4.9 การควบคุม PID เมื่อค่า PB, Ti, Td เท่ากับ 800, 0, 0 ตามลำดับ

สังเกตเห็นว่าระบบมีช่วงเวลากการตอบสนองต่อการสลับค่า SV ช้า จึงต้องทำการลดค่า Proportional Band (PB) ลง เพื่อให้ระบบมีผลตอบสนองที่ดีขึ้น โดยลดเปลี่ยนค่า PB ที่ 500, 200, 100 ตามลำดับ

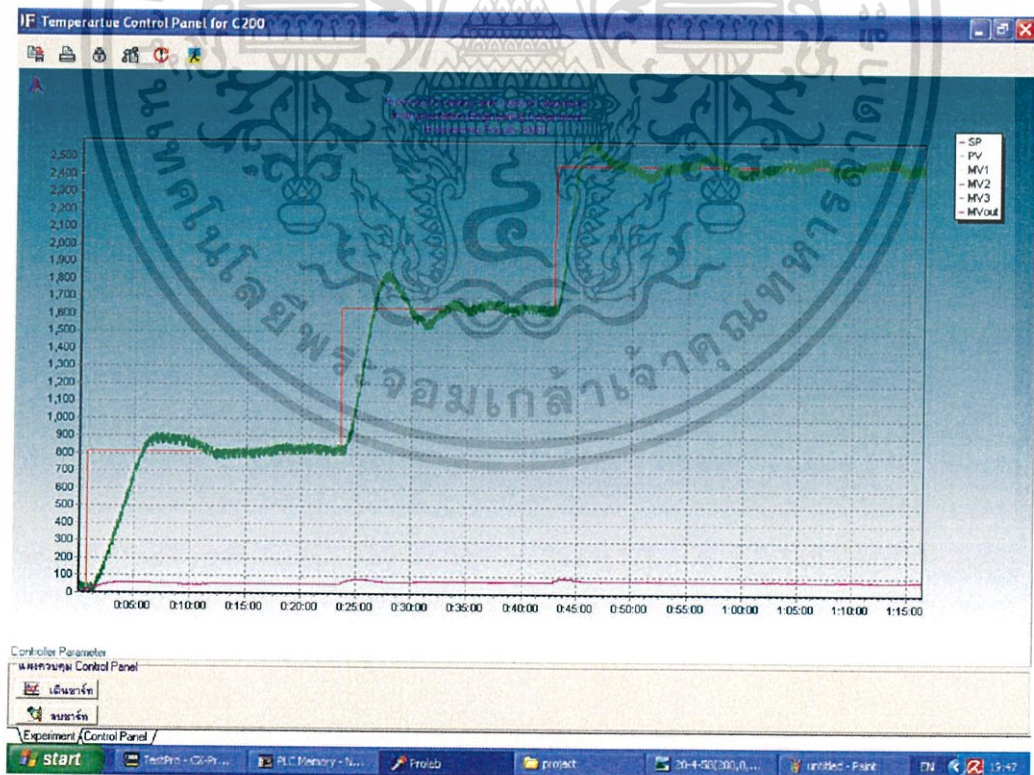


รูปที่ 4.10 การควบคุม PID เมื่อค่า PB, Ti, Td เท่ากับ 500, 0, 0 ตามลำดับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.11 การควบคุม PID เมื่อค่า PB, Ti, Td เท่ากับ 200, 0, 0 ตามลำดับ

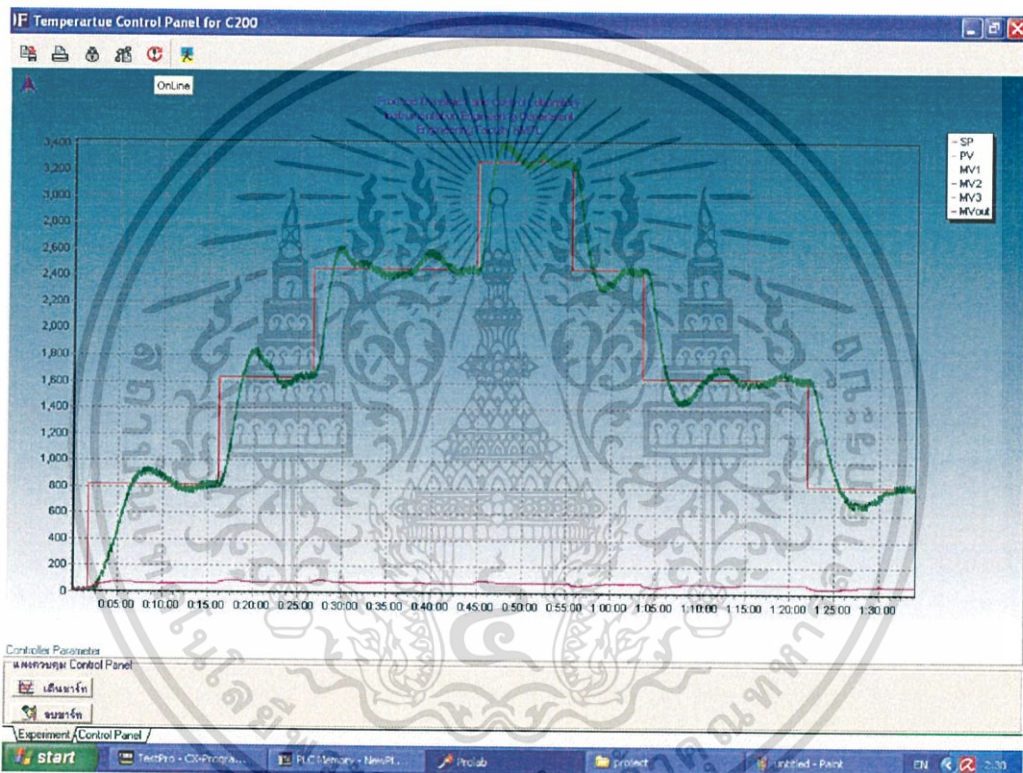


รูปที่ 4.12 การควบคุม PID เมื่อค่า PB, Ti, Td เท่ากับ 100, 0, 0 ตามลำดับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

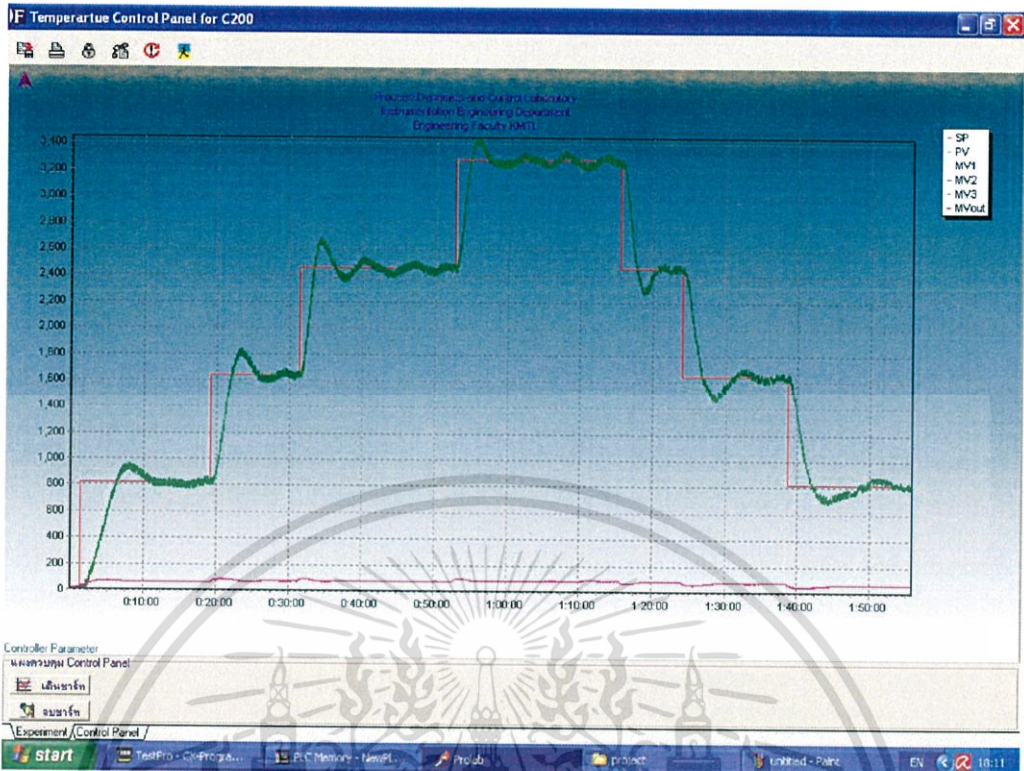
จากรูปที่ 4.9 - 4.12 จะสังเกตเห็นว่าที่ค่า PB เท่ากับ 200 เป็นค่าที่มีความเหมาะสมมากที่สุด เนื่องจากระบบมีช่วงเวลาการตอบสนองต่อการสลับค่า SV ที่เหมาะสมไม่เร็วหรือช้าเกินไป รวมถึงช่วงเวลาในการเข้าสู่สภาวะสมดุลโดยรวมแล้วน้อยกว่าค่าอื่นๆ จากรูปที่ 4.12 ถึงแม้ว่าระบบจะมีช่วงเวลาการตอบสนองที่เร็วแต่เมื่อเริ่มเข้าสู่ช่วงที่กำหนดหรือ SV เกิดการแกว่งของผลตอบสนองระบบขึ้น

หลังจากได้ค่า PB ที่เหมาะสมแล้ว ทำการเปลี่ยนค่า SV ทั้งการสลับค่าขาขึ้นและขาลง จาก 0, 20, 40, 60, 40, 20 ได้ผลดังรูปที่ 4.13 จะพบว่าเมื่อมีการสลับค่า SV ที่สูงขึ้น ผลการตอบสนองเกิดการแกว่งหรือโอเวอร์ชูตขึ้น แก้ไขโดยการเพิ่มค่า Ti (Integral constant) ทดลองที่ค่า 100, 200 และ 300 ตามลำดับ

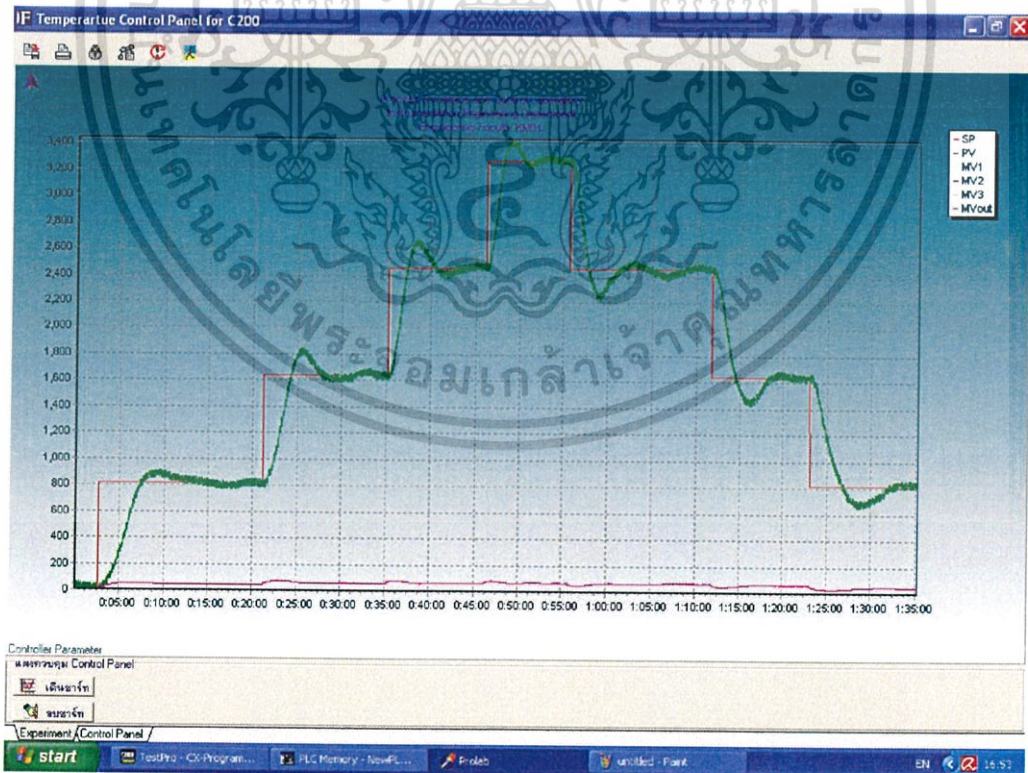


รูปที่ 4.13 การควบคุม PID เมื่อค่า PB, Ti, Td เท่ากับ 200, 0, 0 ตามลำดับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

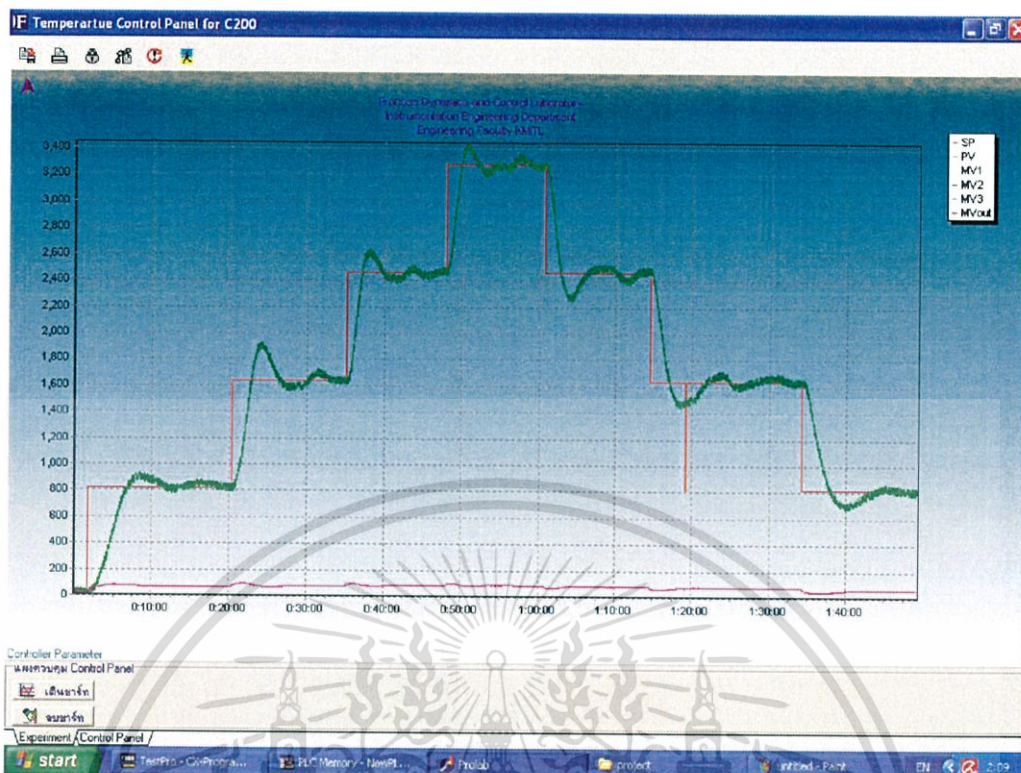


รูปที่ 4.14 การควบคุม PID เมื่อค่า PB, Ti, Td เท่ากับ 200, 100, 0 ตามลำดับ



รูปที่ 4.15 การควบคุม PID เมื่อค่า PB, Ti, Td เท่ากับ 200, 200, 0 ตามลำดับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.16 การควบคุม PID เมื่อค่า PB, Ti, Td เท่ากับ 200, 300, 0 ตามลำดับ

พบว่ารูปที่ 4.15 เมื่อค่า PB, Ti, Td เท่ากับ 200, 200, 0 มีผลตอบสนองที่เหมาะสมกับกระบวนการมากที่สุด เนื่องจากหากมีการเพิ่มค่า Ti ที่น้อยหรือมากเกินไป ผลตอบสนองของกระบวนการจะเกิดโอเวอร์ชูตเมื่อเข้าสู่ช่วงที่กำหนดหรือ SV ดังแสดงในรูปที่ 4.14 และ รูปที่ 4.16

หลังจากที่ได้ค่า PB, Ti, Td ที่เหมาะสมกับกระบวนการแล้ว จึงได้ทำการทดลองในช่วงเวลาที่แตกต่างกัน เพื่อสังเกตค่าอุณหภูมิจาก Thermocouple ที่ SV ต่างๆ

ตารางที่ 4.8

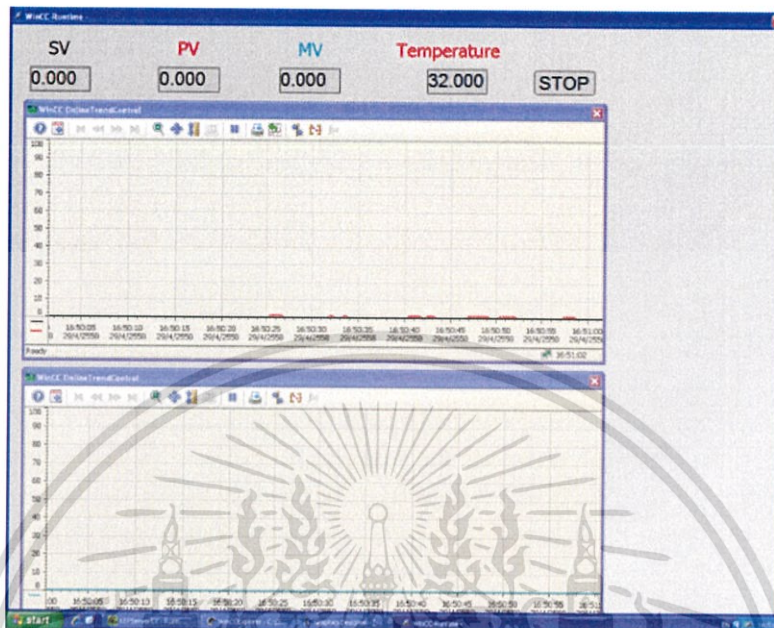
การทดลอง ที่	เวลาที่ทำการทดลอง	อุณหภูมิห้อง (°C)	อุณหภูมิจาก Thermocouple ที่ SV (°C)			
			20%	40%	60%	80%
1	20.30 น.	24	46.7	60.2	73.5	86.4
2	15.30 น.	25	47.8	60.8	73.5	86.4
3	17.30 น.	24	46.5	59.5	72.8	86.4
4	19.30 น.	24	46.6	59.9	73.5	86.4
5	18.30 น.	24	46.6	60.0	73.3	86.5

จากตารางที่ 4.8 พบว่าทั้ง 5 การทดลองอุณหภูมิจาก Thermocouple ที่ SV ต่างๆจะมีค่าที่ใกล้เคียงกัน และยังพบว่าการทดลองที่เวลาต่างๆมีผลต่อค่าอุณหภูมิ เนื่องจากในตอนกลางวันอุณหภูมิสูงก็จะส่งผลให้อุณหภูมิของห้องที่ทำการทดลองสูงกว่าการทดลองในตอนเย็นที่มีอุณหภูมิต่ำกว่า และการทดลองที่อุณหภูมิที่ต่างกันจะส่งผลให้ค่าอุณหภูมิจาก Thermocouple ต่างกันอีกด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

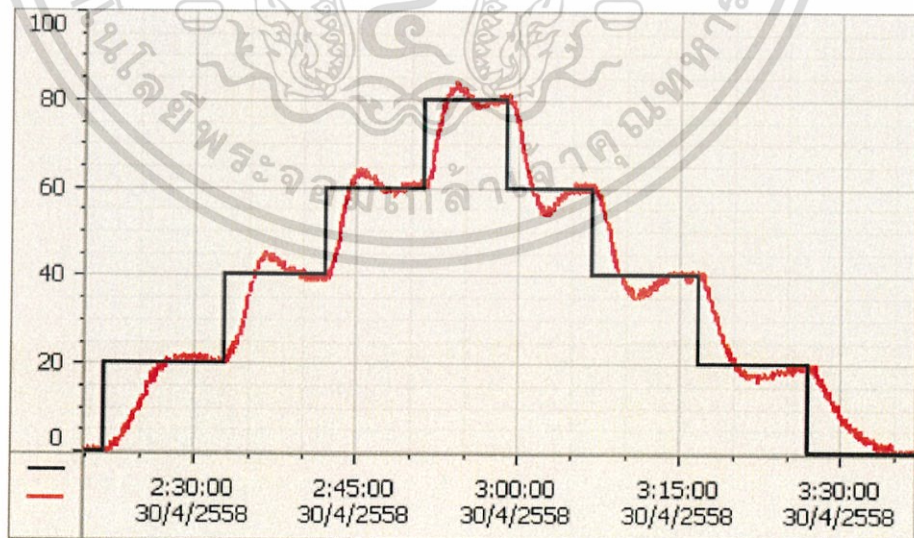
#### 4.5 การแสดงผลผ่าน OPC Server ด้วยโปรแกรม WinCC

เมื่อทำการแสดงผล (Activate) กราฟฟิก จะแสดงหน้ากราฟฟิกดังรูปที่ 4.13



รูปที่ 4.17 แสดงหน้ากราฟฟิก WinCC

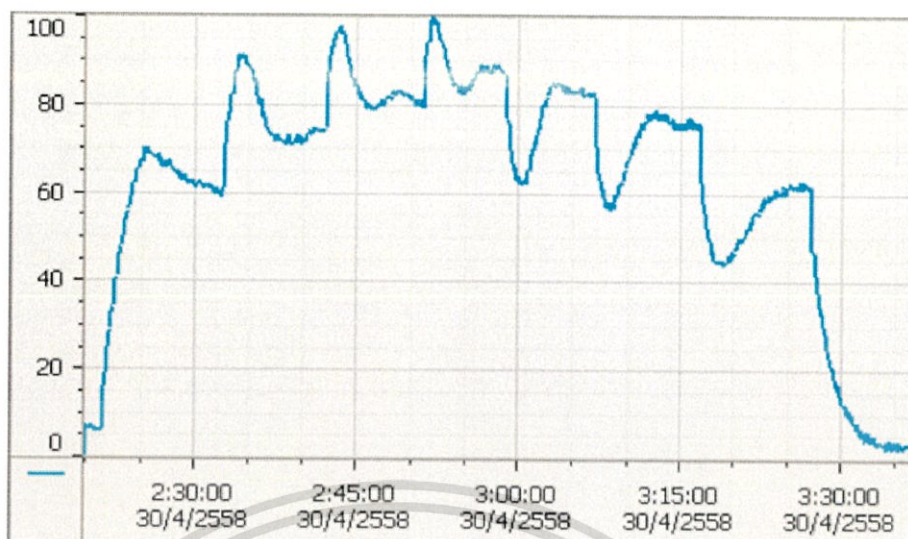
จากการทดลองเมื่อทำการสลับค่า SV จาก 0% เป็น 20, 40, 60, 80, และสลับกลับจาก 80% เป็น 60, 40 และ 20% ตามลำดับ จะได้กราฟความสัมพันธ์ระหว่างค่า SV กับ PV ดังรูปที่ 4.18 และได้กราฟแนวโน้มของค่า MV ดังรูปที่ 4.19



รูปที่ 4.18 กราฟแสดงความสัมพันธ์ระหว่างค่า SV กับ PV

โดยที่ กราฟเส้นสีดำคือค่า SV และกราฟเส้นสีแดงคือค่า PV

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.19 กราฟแสดงแนวโน้มของค่า MV

จะสังเกตว่าบนกราฟพิก WinCC สามารถที่จะแสดงค่า SV, PV, MV และค่าอุณหภูมิได้ชัดเจน เนื่องจากช่องแสดงเป็นตัวเลขพร้อมกับแสดงกราฟความสัมพันธ์เพื่อให้สามารถมองแนวโน้มของผลได้ชัดเจนยิ่งขึ้น เมื่อนำผลการทดลองมาทำการวิเคราะห์จะได้ผลดังตาราง

ตารางที่ 4.9

SV (%)	ช่วงเวลาเข้าสู่สมดุล (นาที)	PV (%)	MV	อุณหภูมิ (°C)
0	15	0	0	32
20	12	20	60	46
40	13	40	73	59
60	12	61	81	73
80	8	80	87	86
60	8	60	82	73
40	10	40	76	60
20	10	20	62	46
0	11	0	3	32

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.10

SV (%)	ช่วงเวลาเข้าสู่สมดุล (นาที)	PV (%)	MV	อุณหภูมิ(°C)
0	15	0	1	32
20	12	21	56	46
40	13	40	72	60
60	12	61	81	73
80	8	80	87	86
60	8	59	83	72
40	10	40	77	59
20	10	20	61	45
0	11	0	3	32

ผลการทดลองจากกราฟความสัมพันธ์และตารางจะพบว่าไม่ว่าจะทำการเปลี่ยนแปลงค่า SV ไปที่ค่าต่างๆ ระบบ PID Control จะทำการปรับค่า MV เพื่อให้ค่า PV ที่วัดกลับมาจากเทอร์โมคัปเปิลค่อยๆ เพิ่มขึ้นจนมีค่าเท่ากับ SV ที่ทำการเปลี่ยนแปลงไป โดยช่วงที่ PV มีค่าน้อยกว่า SV ที่ตั้งไว้ ค่า MV จะมีค่าเพิ่มขึ้นเรื่อยๆจนกว่า PV จะมีค่าเท่ากับ SV ในทางกลับกันหาก PV มีค่าเกินกว่าที่ SV ตั้งไว้ MV จะมีค่าลดลงจนกว่า PV จะมีค่าเข้าสู่ SV

## บทที่ 5

# สรุปผลการทดลองและข้อเสนอแนะ

### 5.1 สรุปผลการทดลอง

จากวิธีการทดลองและผลการทดลองในบทที่ 4 สามารถสรุปผลการทดลองได้ดังนี้ ผู้ศึกษาสามารถใช้คอมพิวเตอร์ทำงานร่วมกับพีแอลซีเพื่อใช้ในการควบคุมอุณหภูมิของชุดทดลอง Temperature Simulator โดยใช้การทำงานของฟังก์ชัน PID Control ซึ่งผลการควบคุมอุณหภูมิของกระบวนการเมื่อผ่านอุปกรณ์ทำความร้อนและรับค่าความร้อนผ่านทางเทอร์โมคัปเปิลเพื่อให้อุณหภูมิอยู่ที่ระดับที่ตั้งไว้ (Set Point) ได้ เมื่อค่า Set point เปลี่ยนแปลง ระบบ PID Control จะทำการปรับค่าเพื่อให้เข้าสู่ภาวะคงตัว (Steady-state) เองโดยอัตโนมัติ ซึ่งในการทดลองนี้การสังเกตผลผ่านทาง OPC Server ด้วยโปรแกรม WinCC จะสามารถแสดงผลให้เห็นค่า SV, PV, MV ในรูปของเปอร์เซ็นต์ และแสดงค่าอุณหภูมิในหน่วยองศาเซลเซียส นอกจากนั้นยังแสดงกราฟแนวโน้มของค่า PV และ MV ที่ SV ค่าต่างๆ

จากการทดลองทำให้พบว่าค่า PB (Proportional Band) ที่น้อยลงมีผลทำให้ช่วงเวลารับค่าลดลง สังเกตจากการควบคุมการทดลองที่ทำให้ค่า PB มีผลต่อการปรับค่าเข้าสู่ภาวะคงตัวได้ดีขึ้น แต่เมื่อเริ่มเข้าสู่ SV เกิดการแกว่งของผลตอบสนองระบบขึ้น จึงต้องใช้ค่า Ti มาช่วยในการกำจัดโอเวอร์ชูตที่เกิดขึ้นจาก PB เนื่องจากเป็นการทดลองของ Temperature Simulator ซึ่งเป็นชุดทดลองขนาดเล็กจึงใช้เพียง PI Controller ก็สามารถทำให้ระบบเกิดเสถียรภาพได้แล้ว แต่หากเป็นการควบคุมระบบที่ใหญ่มากขึ้นจำเป็นต้องใช้เป็น PID Controller จึงจะเหมาะสมกับกระบวนการ

### 5.2 ข้อเสนอแนะ

การทดลองเป็นเรื่องของการควบคุมอุณหภูมิซึ่งผลการทดลองนั้นจะมีความคลาดเคลื่อนได้ง่ายมาก เนื่องจากอุณหภูมิของสิ่งแวดล้อมของห้องปฏิบัติการที่ทำการทดลองนั้นเกิดการเปลี่ยนแปลงได้ง่ายทั้งจากการเข้าออกของนักศึกษาตลอดเวลาและช่วงเวลาที่ทำการทดลองก็ยังมีผลต่ออุณหภูมิของสิ่งแวดล้อม ในเวลากลางวันจะมีอุณหภูมิที่สูงกว่าเวลาในช่วงกลางคืน หากทำการทดลองในห้องปฏิบัติการที่มีอุณหภูมิที่คงที่ก็จะทำให้ผลการทดลองมีความเที่ยงตรงมากยิ่งขึ้น

## บรรณานุกรม

- [1] Operation Manual Omron C200HS
- [2] SIEMENS SIMATIC HMI WinCC V7.2 Getting Started
- [3] OLE for Process Control from <http://en.wikipedia.org/>

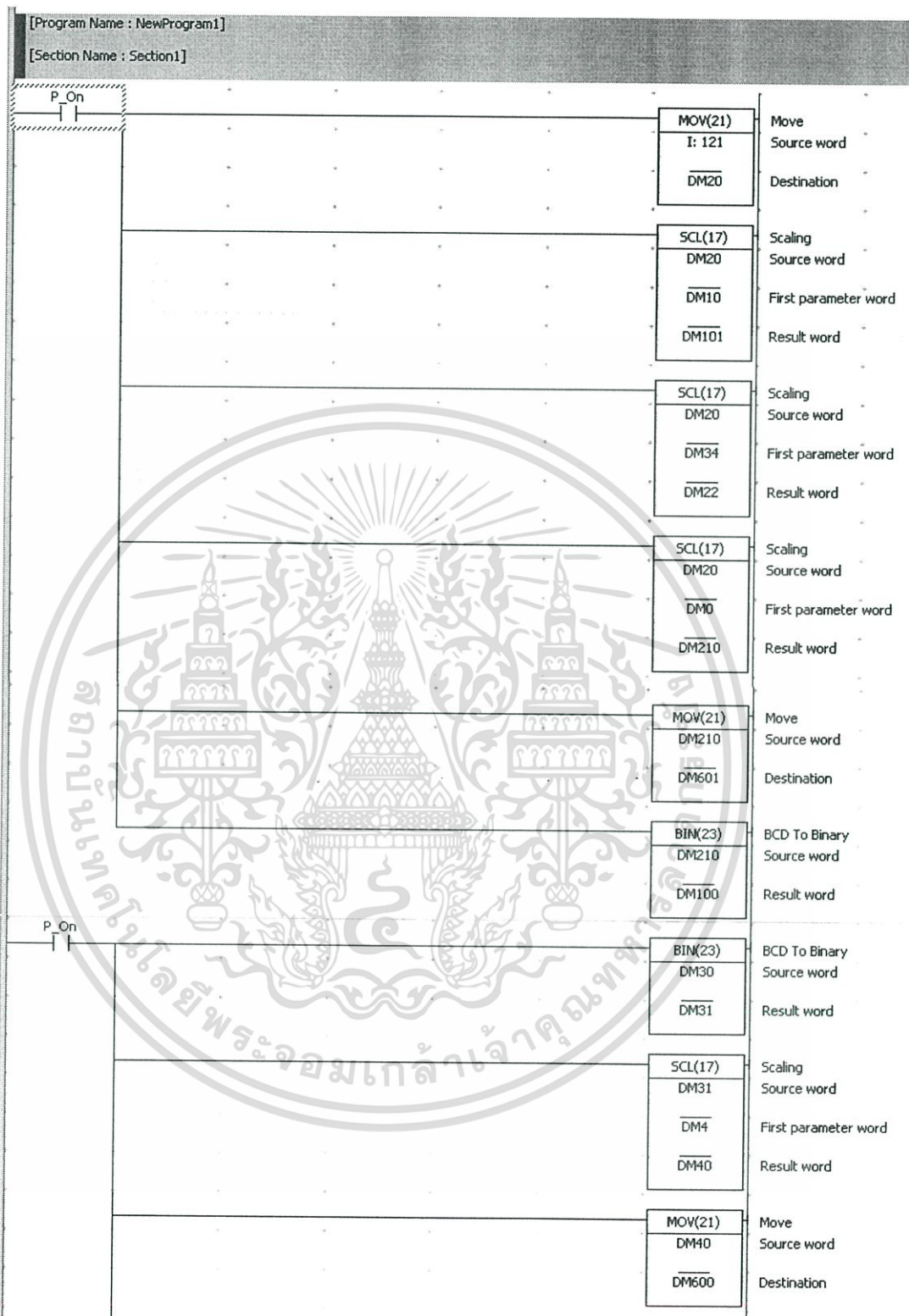


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

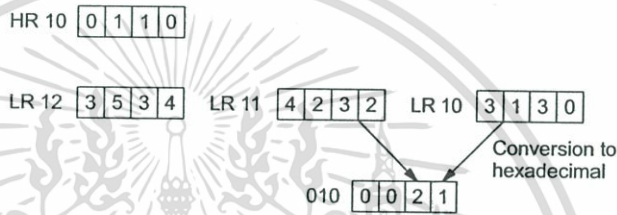
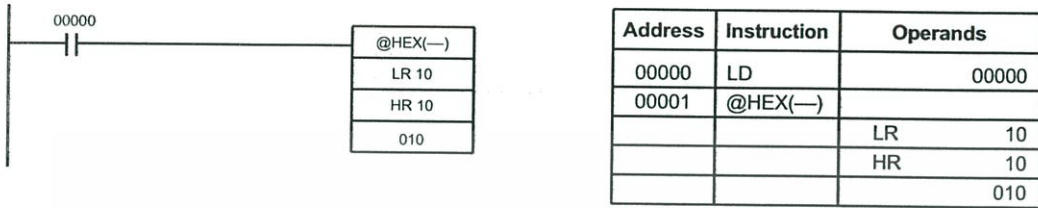


Flags

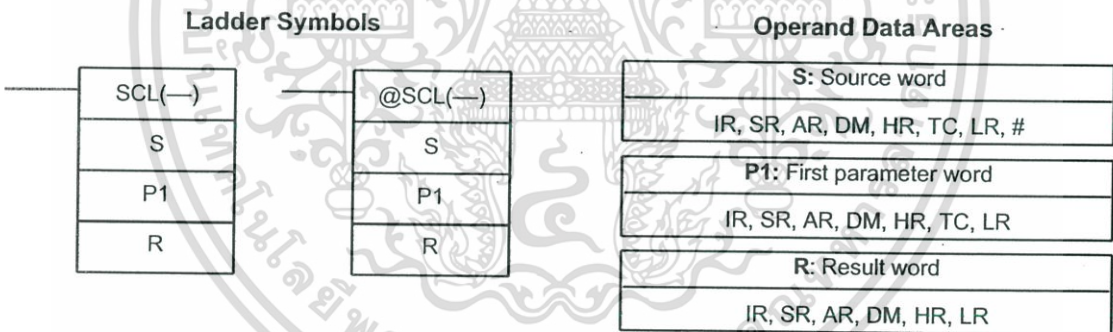
**ER:** Incorrect digit designator, or data area for destination exceeded.  
Indirectly addressed DM word is non-existent. (Content of \*DM word is not BCD, or the DM area boundary has been exceeded.)

Example

In the following example, the 2<sup>nd</sup> byte of LR 10 and the 1<sup>st</sup> byte of LR 11 are converted to hexadecimal values and those values are written to the first and second bytes of IR 010.



5-18-12 SCALING – SCL(—)



Limitations

P1 and P1+2 must be BCD.  
P1 through P1+3 must be in the same data area.  
P1+1 and P1+3 must not be set to the same value.

Description

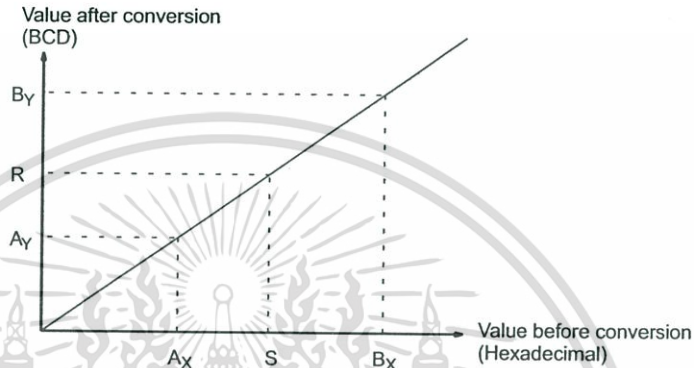
SCL(—) is used to linearly convert a 4-digit hexadecimal value to a 4-digit BCD value. Unlike BCD(24), which converts a 4-digit hexadecimal value to its 4-digit BCD equivalent ( $S_{hex} \rightarrow S_{BCD}$ ), SCL(—) can convert the hexadecimal value according to a specified linear relationship. The conversion line is defined by two points specified in the parameter words P1 to P1+3.

When the execution condition is OFF, SCL(—) is not executed. When the execution condition is ON, SCL(—) converts the 4-digit hexadecimal value in S to the 4-digit BCD value on the line defined by points (P1, P1+1) and (P1+2, P1+3) and places the result in R. The result is rounded off to the nearest integer. If the result is less than 0000, then 0000 is written to R, and if the result is greater than 9999, then 9999 is written to R.

The following table shows the functions and ranges of the parameter words:

Parameter	Function	Range	Comments
P1	BCD point #1 (A <sub>Y</sub> )	0000 to 9999	---
P1+1	Hex. point #1 (A <sub>X</sub> )	0000 to FFFF	Do not set P1+1=P1+3.
P1+2	BCD point #2 (B <sub>Y</sub> )	0000 to 9999	---
P1+3	Hex. point #2 (B <sub>X</sub> )	0000 to FFFF	Do not set P1+3=P1+1.

The following diagram shows the source word, S, converted to D according to the line defined by points (A<sub>Y</sub>, A<sub>X</sub>) and (B<sub>Y</sub>, B<sub>X</sub>).



The results can be calculated by first converting all values to BCD and then using the following formula.

$$\text{Results} = B_Y - [(B_Y - A_Y) / (B_X - A_X) \times (B_X - S)]$$

Flags

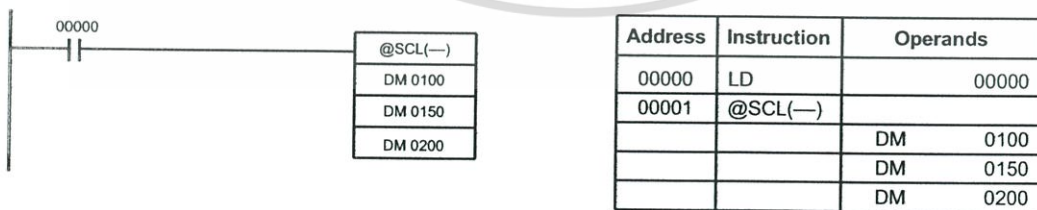
**ER:** The value in P1+1 equals that in P1+3.  
Indirectly addressed DM word is non-existent. (Content of \*DM word is not BCD, or the DM area boundary has been exceeded.)

P1 and P1+3 are not in the same data area, or other setting error.

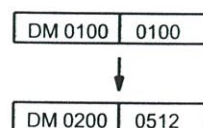
**EQ:** ON when the result, R, is 0000.

Example

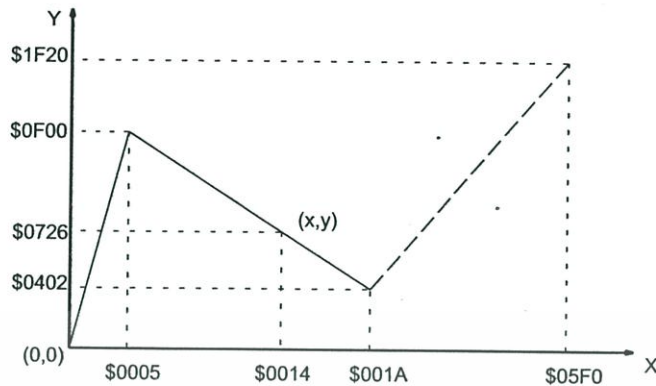
When 00000 is turned ON in the following example, the BCD source data in DM 0100 (#0100) is converted to hexadecimal according to the parameters in DM 0150 to DM 0153. The result (#0512) is then written to DM 0200.



DM 0150	0010
DM 0151	0005
DM 0152	0050
DM 0153	0019

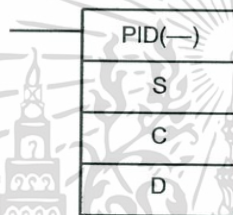


In this case, the input data word, IR 010, contains #0014, and f(0014) = #0726 is output to R, IR 011.



5-21-6 PID CONTROL – PID(—)

Ladder Symbol



Operand Data Areas

<b>S:</b> Input word
IR, SR, AR, DM, HR, LR,
<b>C:</b> First parameter word
IR, SR, DM, HR, LR
<b>D:</b> Output word
IR, SR, AR, DM, HR, LR

Limitations

C and C+32 must be within the same data area.

**Caution**

Do not program PID(—) in the following situations. Doing so may produce unexpected behavior: In interrupt programs, in subroutines, between IL(02) and ILC(03), between JMP(04) and JME(05), and in step programming (when using STEP(08) and SNXT(09)).

Description

PID(—) carries out PID control according to the designated parameters. It takes the specified input range of binary data from the contents of input word S and carries out the PID operation according to the parameters that are set. The results are then stored as the operation output amount in output word D.

PID parameter words range from C through C+32. The PID parameters are configured as shown below.

Word	15 to 12	11 to 8	7 to 4	3 to 0
C	Set value (SV)			
C+1	Proportional band (P)			
C+2	Tik = Integral time T <sub>1</sub> /sampling period γ (See note 1.)			
C+3	Tdk = Derivative time T <sub>d</sub> /sampling period γ (See note 1.)			
C+4	Sampling period γ			
C+5	2-PID parameter (α) (See note 2.)			PID forward/ reverse designation
C+6	0	Input range	0	Output range
C+7 to C+32	Work area (Cannot be accessed directly from program.)			

- Note**
1. What is set in words 2 and 3 is not the actual integral and derivative times, but rather their values divided by the value of the sampling period γ.
  2. Setting the 2-PID parameter (α) to 000 yields 0.65, the normal value.

Parameter Settings

Item	Contents	Setting range
Set value (SV)	This is the target value of the process being controlled.	Binary data (of the same number of bits as specified for the input range)
Proportional band	This is the parameter for P control expressing the proportional control range/total control range.	0001 to 9999 (4 digits BCD); (0.1% to 999.9%, in units of 0.1%)
Tik	This is a constant expressing the strength of the integral operation. As this value increases, the integral strength decreases.	0001 to 8191 (4 digits BCD); (9999 = Integral operation not executed)
Tdk	This is a constant expressing the strength of the derivative operation. As this value increases, the derivative strength increases.	0000 to 8191 (4 digits BCD)
Sampling period	This sets the period for executing the PID operation.	0001 to 1023 (4 digits BCD); (0.1 to 102.3 s, in units of 0.1 s)
PID forward/reverse designation	This is the parameter that determines the direction of the proportional operation.	0: Reverse operation 1: Forward operation
2-PID parameter ( $\alpha$ )	This is the input filter coefficient. Normally use 0.65 (i.e., a setting of 000). The filter efficiency decreases as the coefficient approaches 0.	000: $\alpha = 0.65$ Setting from 100 to 199 means that the value of the rightmost two digits is set from $\alpha = 0.00$ to $\alpha = 0.99$ .
Input range	This is the number of input data bits.	0: 8 bits 1: 9 bits 2: 10 bits 3: 11 bits 4: 12 bits
Output range	This is the number of output data bits. (The number of output bits is automatically the same as the number of input bits.)	5: 13 bits 6: 14 bits 7: 15 bits 8: 16 bits

**PID CONTROL Operation**

**Execution Condition OFF**

All data that has been set is retained. Then the execution condition is OFF, the operation amount can be written to the output word (D) to achieve manual control.

**Rising Edge of the Execution Condition**

The work area is initialized based on the PID parameters that have been set and the PID control operation is begin. Sudden and radical changes in the operation output amount are not made when starting operation to avoid adverse affect on the controlled system (bumpless operation).

When PID parameters are changed, they first become valid when the execution condition changes from OFF to ON.

**Execution Condition ON**

The PID operation is executed at the intervals based on the sampling period, according to the PID parameters that have been set.

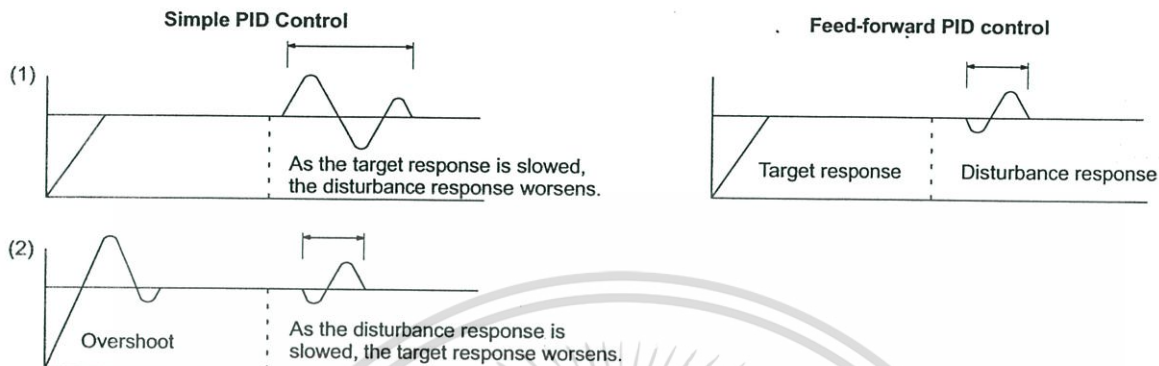
**Sampling Period and PID Execution Timing**

The sampling period is the time interval to retrieve the measurement data for carrying out a PID operation. PID(—), however, is executed according to C200HS cycle time, so there may be cases where the sampling period is exceeded. In such cases, the time interval until the next sampling is reduced.

**PID Control Method**

C200HS PID control operations are executed by means of PID control with feed-forward control (two degrees of freedom).

When overshooting is prevented with simple PID control, stabilization of disturbances is slowed (1). If stabilization of disturbances is speeded up, on the other hand, overshooting occurs and response toward the target value is slowed (2). With feed-forward PID control, there is no overshooting, and response toward the target value and stabilization of disturbances can both be speeded up (3).

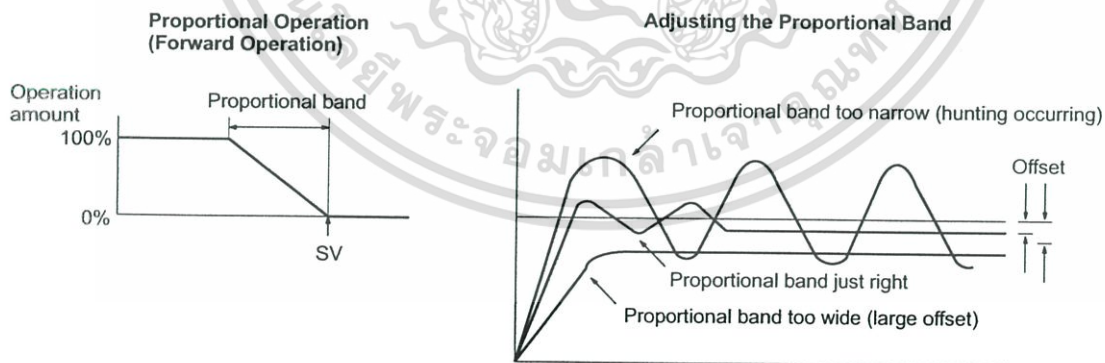


Control Operations

Proportional Operation (P)

Proportional operation is an operation in which a proportional band is established with respect to the set value (SV), and within that band the operation amount (the control output amount) is made proportional to the deviation. If the present value (PV) is smaller than the proportional band, the operation amount will be 100%. If within the proportional band the operation amount is made proportional to the deviation and gradually decreased until the SV and PV match (i.e., until the deviation is 0), the operation amount will return to the previous value (forward operation).

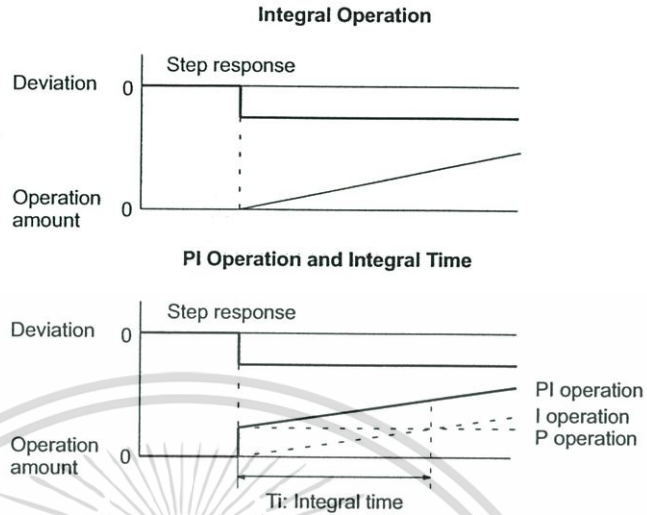
The proportional band is expressed as a percentage with respect to the total input range. With proportional operation an offset (residual deviation) occurs, and the offset is reduced by making the proportional band smaller. If it is made too small, however, hunting will occur.



Integral Operation (I)

Combining integral operation with proportional operation reduces the offset according to the time that has passed. The strength of the integral operation is indicated by the integral time, which is the time required for the integral operation amount to reach the same level as the proportional operation amount with respect to the step deviation, as shown in the following illustration. The shorter the integral time, the stronger the correction by the integral operation will be. If the

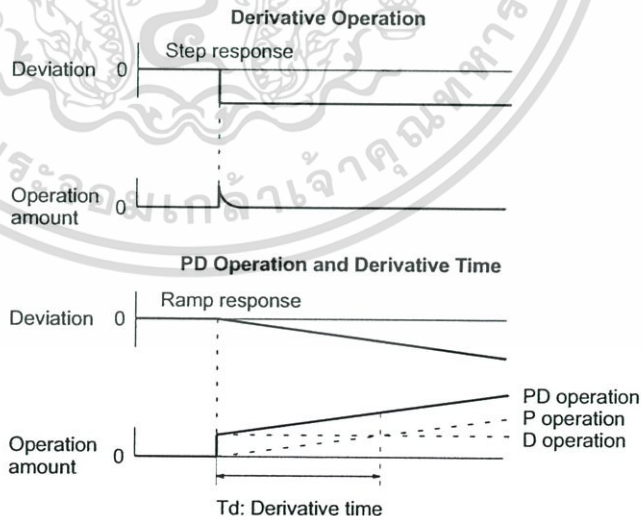
integral time is too short, the correction will be too strong and will cause hunting to occur.



**Derivative Operation (D)**

Proportional operation and integral operation both make corrections with respect to the control results, so there is inevitably a response delay. Derivative operation compensates for that drawback. In response to a sudden disturbance it delivers a large operation amount and rapidly restores the original status. A correction is executed with the operation amount made proportional to the incline (derivative coefficient) caused by the deviation.

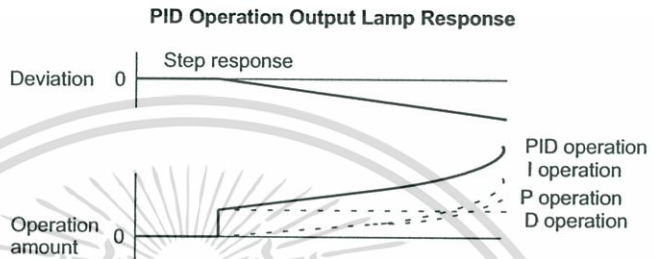
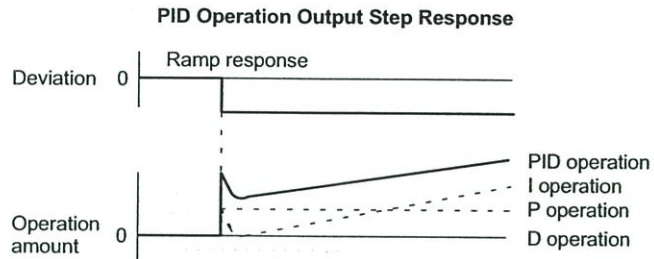
The strength of the derivative operation is indicated by the derivative time, which is the time required for the derivative operation amount to reach the same level as the proportional operation amount with respect to the step deviation, as shown in the following illustration. The longer the derivative time, the stronger the correction by the derivative operation will be.



**PID Operation**

PID operation combines proportional operation (P), integral operation (I), and derivative operation (D). It produces superior control results even for control objects with dead time. It employs proportional operation to provide smooth control

without hunting, integral operation to automatically correct any offset, and derivative operation to speed up the response to disturbances.



**Direction of Operation**

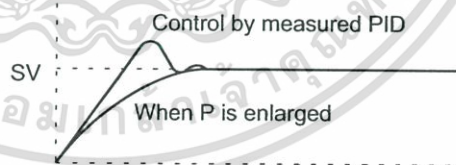
When using PID operation, select either of the following two control directions. In either direction, the operation amount increases as the difference between the SV and the PV increases.

- Forward operation: Control amount is increased when the PV is larger than the SV.
- Reverse operation: Control amount is increased when the PV is smaller than the SV.

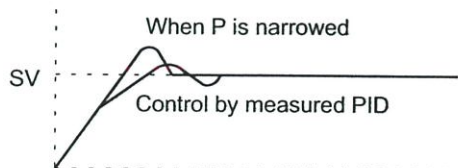
**Adjusting PID Parameters**

The general relationship between PID parameters and control status is shown below.

- When it is not a problem if a certain amount of time is required for stabilization (settlement time), but it is important not to cause overshooting, then enlarge the proportional band.

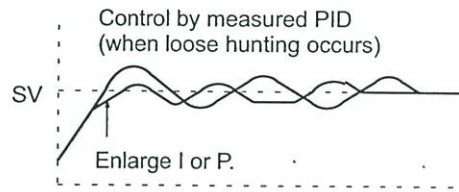


- When overshooting is not a problem but it is desirable to quickly stabilize control, then narrow the proportional band. If the proportional band is narrowed too much, however, then hunting may occur.

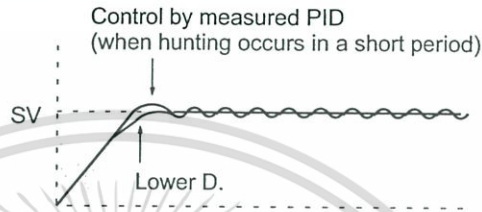


- When there is broad hunting, or when operation is tied up by overshooting and undershooting, it is probably because integral operation is too strong. The

hunting will be reduced if the integral time is increased or the proportional band is enlarged.



- If the period is short and hunting occurs, it may be that the control system response is quick and the derivative operation is too strong. In that case, set the derivative operation lower.

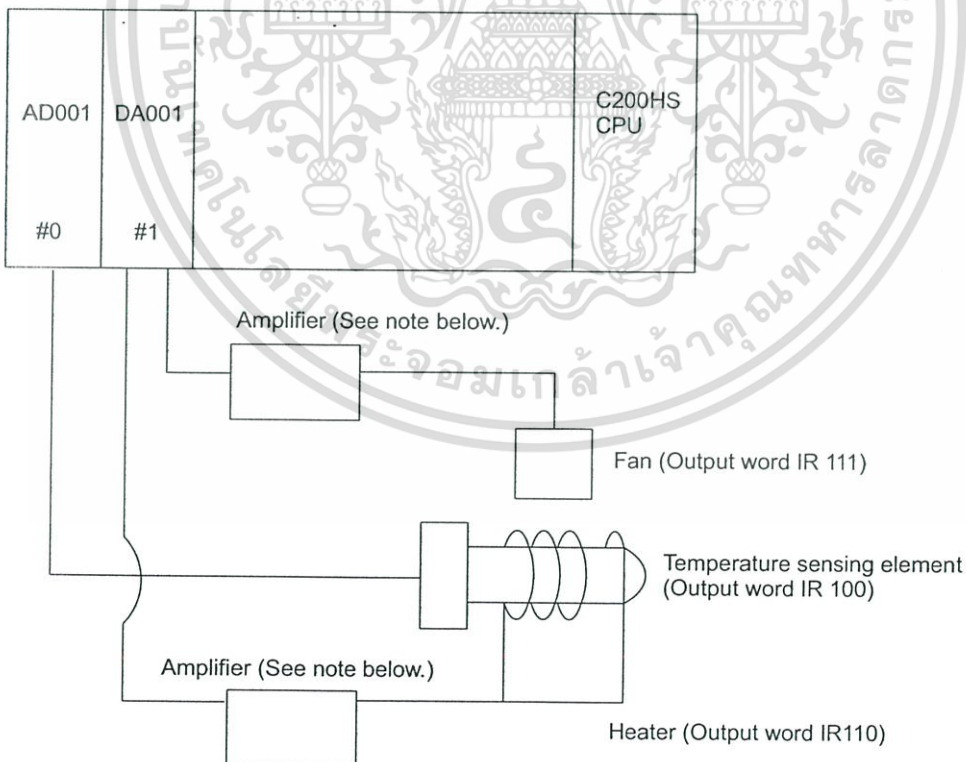


Flags

- ER:** Content of \*DM word is not BCD, or the DM area boundary has been exceeded.  
 A PID parameter SV is out of range.  
 The PID operation was executed but the cycle time was two times the sampling period. PID(—) will be executed for this error only even when ER (SR25503) is ON.
- CY:** The PID operation is being executed.

Example

This example shows a PID control program using PID(—).



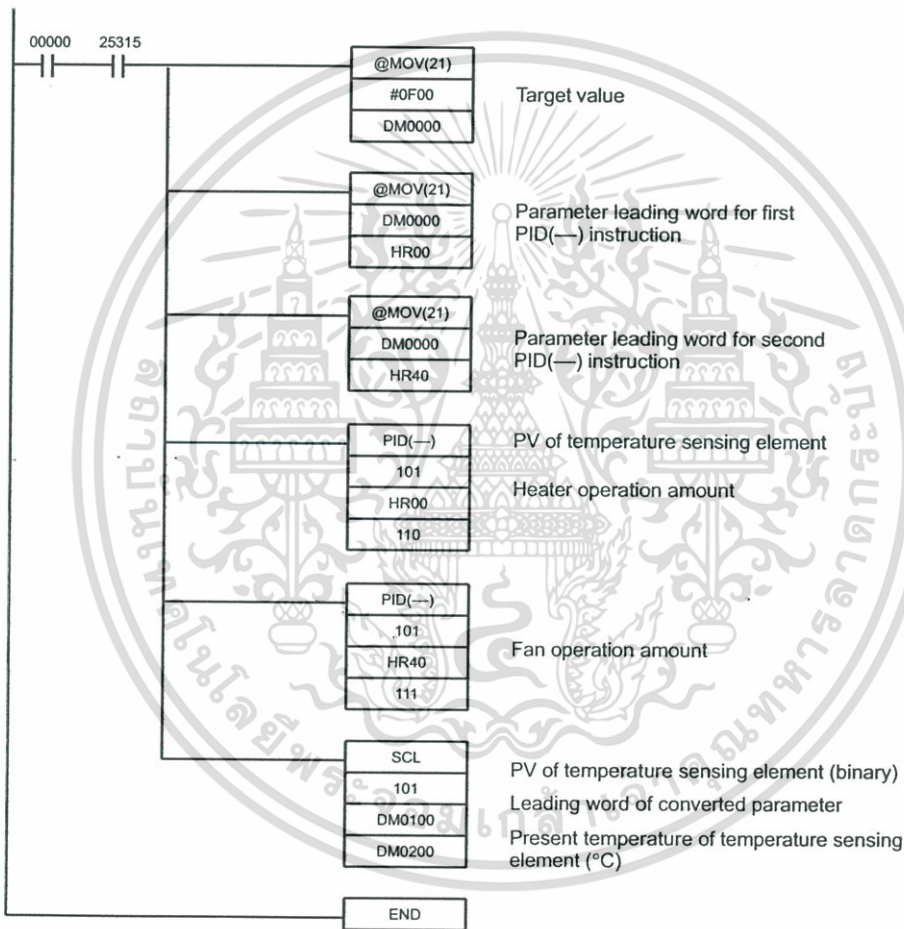
**Note** Motors and heaters cannot be directly connected from a Analog Output Unit. An amplifier (i.e., a current amplification circuit) is required.

Creating the Program

Follow the procedure outlined below in creating the program.

- 1, 2, 3... 1. Set the target value (binary 0000 to 0FFF) in DM 0000.
2. Input the PV of the temperature sensing element (binary 000 to 0FFF) in bits 0 to 11 of word 101.
3. Output the operation amount of the heater to bits 0 to 11 of word 110 by means of the first PID(—) instruction in the following program.
4. Output the operation amount of the fan to bits 0 to 11 of word 111 by means of the second PID(—) instruction in the following program.
5. Convert the PV of the temperature sensing element (binary 000 to 0FFF) to temperature data (0000°C to 0200°C) by means of SCL(—), and output it to DM 0200.

Program



**Note** When using PID(—) or SCL(—), make the data settings in advance with a Peripheral Device such as the Programming Console or LSS.

Heater		
HR 00	(DM0000)	Target value HR
HR 01	0080	Proportional band
HR 02	0200	Integral time/sampling period
HR 03	0100	Derivative time/sampling period
HR 04	0001	Sampling period
HR 05	0000	Forward/reverse designation/ PID parameters
HR 06	0404	I/O range

SCL Parameters		Fan	
DM 0100	0000	HR 40	(DM0000)
DM 0101	0000	HR 41	0060
DM 0102	0200	HR 42	0150
DM 0103	0FFF	HR 43	0100
		HR 44	0001
		HR 45	0001
		HR 46	0404

## 5-22 Logic Instructions

The logic instructions – COM(29), ANDW(34), ORW(35), XORW(36), and XNRW(37) – perform logic operations on word data.

### 5-22-1 COMPLEMENT – COM(29)

#### Ladder Symbols



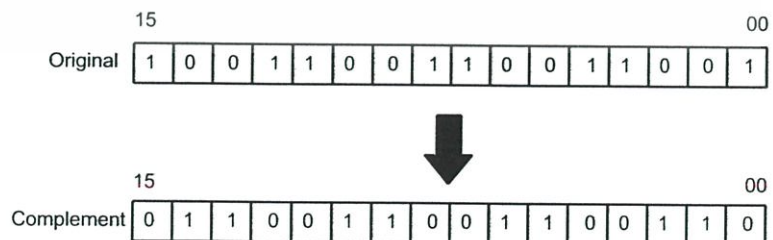
#### Operand Data Areas

<b>Wd:</b> Complement word
IR, SR, AR, DM, HR, LR

#### Description

When the execution condition is OFF, COM(29) is not executed. When the execution condition is ON, COM(29) clears all ON bits and sets all OFF bits in Wd.

#### Example



#### Flags

- ER:** Indirectly addressed DM word is non-existent. (Content of \*DM word is not BCD, or the DM area boundary has been exceeded.)
- EQ:** ON when the result is 0.