สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

การแสดงผลกระบวนการผ่านการติดต่อสื่อสารด้วยโปรโตคอลมอดบัส

PROCESS MONITORING VIA MODBUS PROTOCOL COMMUNICATION

T104120

นายธีรวัฒน์ ศรีสุข
นายลิขิต โอบอ้อม
นายสามารถ รื่นพิทักษ์

เลขหมู่.....
เลขทะเบียน 104120
วัน,เดือน,ปี 30 ต.ค. 2552
.b.....
.i.....

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิชาวิศวกรรมการวัดคุม
ภาควิชาวิศวกรรมการวัดคุม คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2551

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

PROCESS MONITORING VIA MODBUS PROTOCOL COMMUNICATION

THEERAWAT SRISOOK

LIKHIT OB-OM

SAMART RUENPITAK

A THESIS SUBMITTED IN PARTIAL FULFILLMENT

OF THE REQUIREMENT FOR THE DEGREE OF

BACHELOR OF ENGINEERING IN INSTRUMENTATION ENGINEERING

DEPARTMENT OF INSTRUMENTATION ENGINEERING

FACULTY OF ENGINEERING

KING MONGKUT'S INSTITUE OF TECHNOLOGY LARDKRABANG

2008

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาควิชาวิศวกรรมการวัดคุม

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ใบรับรองปริญญานิพนธ์

.....

หัวข้อปริญญานิพนธ์	การแสดงผลกระบวนการผ่านการติดต่อสื่อสารด้วยโปรโตคอลมอดบัส PROCESS MONITORING VIA MODBUS PROTOCOL COMMUNICATION		
นักศึกษาผู้จัดทำ	นายธีรวัฒน์	ศรีสุข	รหัสนักศึกษา 49015372
	นายลิขิต	โอบอ้อม	รหัสนักศึกษา 49015387
	นายสามารถ	รื่นพิทักษ์	รหัสนักศึกษา 49015395
ปริญญา	วิศวกรรมศาสตรบัณฑิต		
สาขาวิชา	วิศวกรรมการวัดคุม		
ปีการศึกษา	2551		

อาจารย์ผู้ควบคุมปริญญานิพนธ์	ลายมือชื่อ
รศ. สุพรรณ กุลพาณิชย์	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อปริญญานิพนธ์	การแสดงผลกระบวนการผ่านการติดต่อสื่อสารด้วยโปรโตคอลมอดบัส PROCESS MONITORING VIA MODBUS PROTOCOL COMMUNICATION		
นักศึกษาผู้จัดทำ	นายธีรวัฒน์	ศรีสุข	รหัสนักศึกษา 49015372
	นายลิขิต	โอบอ้อม	รหัสนักศึกษา 49015387
	นายสามารถ	รื่นพิทักษ์	รหัสนักศึกษา 49015395
อาจารย์ที่ปรึกษา	รศ. สุพรรณ	กุลพาณิชย์	
ปีการศึกษา	2551		

บทคัดย่อ

เนื่องจากความก้าวหน้าด้านเทคโนโลยีในปัจจุบัน ทำให้อุปกรณ์ควบคุมกระบวนการทางอุตสาหกรรมไม่เพียงมีความสามารถในการควบคุมเท่านั้น แต่ยังสามารถติดต่อสื่อสารกับอุปกรณ์อื่นๆ ได้ โดยหัวใจสำคัญที่ทำให้การติดต่อสื่อสารนั้นเกิดขึ้นได้ ก็คือการที่อุปกรณ์ควบคุมใช้รูปแบบในการรับ-ส่งข้อมูลที่เป็นมาตรฐานเดียวกัน หรือที่เรียกว่า "โปรโตคอล" ปริญญานิพนธ์นี้เป็นการศึกษาการแสดงผลกระบวนการ โดยใช้การติดต่อสื่อสารด้วยโปรโตคอลมอดบัสซึ่งเป็นโปรโตคอลที่ทำงานภายใต้มาตรฐานการรับส่งข้อมูลแบบอนุกรม โดยได้ทำการสร้างกระบวนการจำลองและระบบควบคุมทางอุตสาหกรรม ในรูปแบบการควบคุมอุณหภูมิของเหลวในถังด้วยตัวควบคุมอุณหภูมิ FUJI PXR4 และควบคุมระดับ และการปล่อยของเหลว หรือลำดับขั้นการทำงานด้วยตัวควบคุม PLC KOYO DO-06DD1 ซึ่งตัวควบคุมทั้งสองเป็นตัวควบคุมที่ต่างตระกูลและต่างหน้าที่กัน แต่สามารถทำงานเป็นหนึ่งเดียวกัน ด้วยการสื่อสารข้อมูลถึงกันผ่านเครือข่ายด้วย Modbus Protocol

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Thesis Title	Process Monitoring via Modbus Protocol Communication	
Authors	Mr.Theerawat	Srisook
	Mr.Likhit	Ob-Om
	Mr.Samart	Ruenpitak
Thesis Advisor	Assoc.Prof. Suphan	Gulpanich
Year	2008	

ABSTRACT

Due to in the present the technology about the industrial process controller have developed. It not only supports control function but also communicates function. The heart that makes it possible is the protocol. This project is to study in the area of process monitoring and the Modbus protocol that base on the serial communication standard. We built the temperature control of liquid in open tank with FUJI PXR4 temperature controller and used the KOYO DO-06DD1 programmable logic controller for sequence control. Both of the controllers can work together by data communication via Modbus Protocol network.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

ปริญญานิพนธ์ฉบับนี้สำเร็จลุล่วงไปได้ด้วยดี เพราะได้รับการสนับสนุนในการให้คำปรึกษาและคำแนะนำจากรองศาสตราจารย์สุพรรณ กุลพาณิชย์ นอกจากนี้ยังได้ให้การสนับสนุนในเรื่องอุปกรณ์ต่าง ๆ ในการจัดทำปริญญานิพนธ์ คณะผู้จัดทำขอขอบพระคุณเป็นอย่างสูง

ขอขอบคุณอาจารย์ภาควิชาวิศวกรรมการวัดคุมทุกท่านที่ได้ให้ความรู้ ในด้านวิศวกรรม-การวัดคุมจนทำให้ผู้จัดทำมีความรู้ในการจัดทำปริญญานิพนธ์เล่มนี้

ขอขอบคุณนางสาวจันทิมา หีบนาค นายธเนศ เพชรน้ำทอง นายธนาธิป กองสวัสดิ์ ที่ได้ให้ความช่วยเหลือในการจัดทำปริญญานิพนธ์เป็นอย่างดี

สุดท้ายนี้ขอกราบขอบพระคุณบิดา มารดาและครอบครัวที่ได้ให้การสนับสนุนทุนทรัพย์ในการศึกษาและเป็นกำลังใจให้ตลอดมา ความรู้และประโยชน์ที่ได้รับจากปริญญานิพนธ์ฉบับนี้ คณะผู้จัดทำขอมอบแด่ผู้มีพระคุณทุกท่าน

คณะผู้จัดทำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

หน้า

บทคัดย่อภาษาไทย.....I

บทคัดย่อภาษาอังกฤษ.....II

กิตติกรรมประกาศ.....III

สารบัญ.....IV

สารบัญตาราง.....VX

สารบัญภาพ.....X

บทที่ 1 บทนำ.....1

1.1 ความสำคัญของปริญญานิพนธ์.....1

1.2 วัตถุประสงค์ของปริญญานิพนธ์.....1

1.3 ขอบเขตของปริญญานิพนธ์.....1

1.4 ขั้นตอนการศึกษา.....2

บทที่ 2 ทฤษฎีและหลักการ.....3

2.1 การแสดงผลกระบวนการ.....3

2.2 โปรโตคอลMODBUS.....3

2.2.1 ข้อมูลทางเทคนิคของโปรโตคอล MODBUS.....3

2.2.2 โหมดการรับส่งข้อมูลอนุกรม.....4

2.2.3 MODBUS RTU.....5

2.2.4 MODBUS ASCII.....6

2.2.5 การตรวจสอบความผิดพลาดของข้อมูล.....8

2.2.5.1 Parity.....8

2.2.5.2 Redundancy check.....8

2.2.5.3 Cyclical Redundancy Check (CRC).....8

2.2.5.4 Longitudinal Redundancy Check (LRC).....10

2.3 MODBUS FUNCTION CODE.....11

2.3.1 Read Coil Status (01).....11

2.3.2 Read Input Status (02).....12

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

หน้า

2.3.3 Read Holding Register (03).....13

2.3.4 Read Input Register (04).....14

2.3.5 Force Single Coil (05).....15

2.3.6 Preset Single Register (06).....16

2.3.7 Force Multiple Coils (15,0X0F).....17

2.3.8 Present Multiple Register (16,0X10).....18

2.4 มาตรฐานการรับ-ส่งข้อมูลแบบอนุกรม.....19

2.4.1 มาตรฐาน RS-232.....19

2.4.2 มาตรฐาน RS-232C.....20

2.4.3 มาตรฐาน RS-422 หรือ RS-422-A.....20

2.4.4 มาตรฐาน RS-485.....21

2.5 การควบคุมแบบอัตโนมัติ (Automation Control).....23

2.5.1 การควบคุมแบบ ON - OFF หรือ Two – Position Control.....24

2.5.2 การควบคุมแบบ Proportional (P Control).....25

2.5.3 การควบคุมแบบ Proportional + Integral (PI Control).....28

2.5.4 การควบคุมแบบ Proportional + Derivative (PD Control).....30

2.5.5 การควบคุมแบบ Proportional + Integral + Derivative (PID).....31

2.5.6 จุดมุ่งหมายของการควบคุม.....32

2.5.7 สรุปผลของวิธีการควบคุมแบบต่างๆ.....33

2.5.8 การควบคุมแบบ Fuzzy.....33

2.5.9 การควบคุมแบบ Fuzzy+ PID.....34

2.5.10 AUTO – TUNING.....35

2.5.11 เอาต์พุตของเครื่องควบคุม.....35

2.6 Programmable Logic Controller (PLC).....37

2.6.1 โครงสร้างของ PLC.....38

2.6.2 ตัวอย่างการใช้ PLC ในอุตสาหกรรมต่างๆ.....40

2.7 รีเลย์ (Relay).....40

2.8 RTD (Resistance Temperature Detector).....41

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

หน้า

2.9 การวัดระดับด้วย Sight Glass (แบบใช้กระจกมองระดับ).....44
2.10 หลักการของฮีตเตอร์.....46
2.11 โซลิดสเตทรีเลย์.....50
2.12 การพัฒนาซอฟแวร์โดยใช้ Visual Basic 2008.....52
2.12.1 ส่วนประกอบในหน้าจอ IDE ของ Visual Studio.....53
2.12.2 ขั้นตอนการเขียนโปรแกรมด้วย Visual Basic.....54
2.13 ความรู้พื้นฐานเกี่ยวกับฐานข้อมูลและภาษา SQL.....56
2.13.1 ความหมายของฐานข้อมูล และระบบจัดการฐานข้อมูล (DBMS).....56
2.13.2 ฐานข้อมูลเชิงสัมพันธ์ และคำศัพท์ต่าง ๆ ที่เกี่ยวข้อง.....56
2.14 โปรแกรมสำหรับจัดการฐานข้อมูล Microsoft SQL Server 2005 Express.....58
2.14.1 Microsoft SQL Server Management Studio.....59
2.14.2 ประเภทของฐานข้อมูลของ SQL Server 2005.....61
2.15 การเขียนโปรแกรมเพื่อทำงานกับฐานข้อมูล.....63
2.16 โปรแกรมสร้างและแสดงรายงาน Crystal Report.....69

บทที่ 3 การออกแบบและการสร้าง.....71
3.1 ส่วนฮาร์ดแวร์.....71
3.1.1 การออกแบบและการสร้างกระบวนการจำลอง.....71
3.1.2 การออกแบบและการคำนวณปริมาตรของถัง.....72
3.1.3 การคำนวณหาขนาดของอุปกรณ์ทำความร้อน(Heater).....74
3.1.4 การหาค่ากระแสของ Solid State Relay.....75
3.1.5 อุปกรณ์วัดอุณหภูมิ RTD.....75
3.1.6 เทอร์โมสตัท (Thermostat).....76
3.1.7 อุปกรณ์ควบคุมอุณหภูมิ (Temperature Controller).....76
3.1.8 Programmable Logic Controller (PLC).....77
3.1.9 อุปกรณ์แปลงสัญญาณ RS-232 to RS-485 Converter.....82
3.1.10 โซลินอยด์วาล์ว (Solenoid Valve).....82
3.2 ส่วนซอฟท์แวร์ (Software).....83

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

หน้า

3.2.1 การออกแบบและการพัฒนาโปรแกรมควบคุม PLC.....83

3.2.1.1 PLC I/O Assignment.....83

3.2.1.2 โปรแกรม DirectSOFT32.....83

3.2.1.3 การใช้งานโปรแกรม DirectSOFT32.....84

3.2.2 การออกแบบและการพัฒนาโปรแกรมแสดงผลกระบวนการ.....85

3.2.2.1 การออกแบบระบบการติดต่อสื่อสาร.....85

3.2.2.2 การออกแบบ User Interface 86

3.2.3 การติดต่อสื่อสารกับอุปกรณ์ฮาร์ดแวร์ผ่าน Serial Port ด้วย Visual Basic 2008.....92

3.2.4 การออกแบบฐานข้อมูล.....94

3.2.5 การเชื่อมต่ออุปกรณ์ควบคุมกระบวนการผ่าน RS-485.....96

3.2.5.1 การเชื่อมต่ออุปกรณ์ควบคุมอุณหภูมิ FUJI PXR4.....96

3.2.5.2 การเชื่อมต่อ PLC KOYO DO-06DD1 ผ่าน RS-485.....8

บทที่ 4 การทดลองและผลการทดลอง.....100

4.1 การทดลองการควบคุมแบบต่าง ๆ.....100

4.1.1 การควบคุมอุณหภูมิของน้ำโดยใช้การควบคุมแบบ P.....101

4.1.2 การควบคุมอุณหภูมิของน้ำโดยใช้การควบคุมแบบ PI.....102

4.1.3 การควบคุมอุณหภูมิของน้ำโดยใช้การควบคุมแบบ PD.....103

4.1.4 การควบคุมอุณหภูมิของน้ำโดยใช้การควบคุมแบบ PID.....104

4.1.5 การควบคุมอุณหภูมิของน้ำโดยใช้การควบคุมแบบ Auto-Tuning.....105

4.1.6 การควบคุมอุณหภูมิของน้ำโดยใช้การควบคุมแบบ Fuzzy.....106

4.2 การทดลองการรับ-ส่งค่าผ่านโปรโตคอล MODBUS.....107

4.2.1 การทดลองการรับ-ส่งค่าผ่าน กับ FUJI PXR-4.....107

4.2.2 การทดลองการรับ-ส่งค่า กับ (PLC) KOYO DO-06DD1.....108

4.3 สรุปผลการทดลอง.....108

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

หน้า

บทที่ 5 สรุปผลการวิจัยและข้อเสนอแนะ.....110

5.1 สรุปผลการวิจัย.....110

5.2 ข้อเสนอแนะและแนวทางการพัฒนา.....110

บรรณานุกรม.....112

ภาคผนวก.....113

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

ตารางที่ **หน้า**

2.1 แสดงการส่งข้อมูลโดยใช้ฟังก์ชัน Read Coil Status (01).....11

2.2 แสดงข้อมูลตอบกลับเมื่อใช้ฟังก์ชัน Read Coil Status (01).....12

2.3 แสดงการส่งข้อมูลโดยใช้ฟังก์ชัน Read Input Status (02).....12

2.4 แสดงข้อมูลตอบกลับเมื่อใช้ฟังก์ชัน Read Input Status (02).....13

2.5 แสดงการส่งข้อมูลโดยใช้ฟังก์ชัน Read Holding Register (03) 13

2.6 แสดงข้อมูลตอบกลับเมื่อใช้ฟังก์ชัน Read Holding Register (03).....14

2.7 แสดงการส่งข้อมูลโดยใช้ฟังก์ชัน Read Input Register (04).....14

2.8 แสดงข้อมูลตอบกลับเมื่อใช้ฟังก์ชัน Read Input Register (04).....15

2.9 แสดงการส่งข้อมูลโดยใช้ฟังก์ชัน Force Single Coil (05).....15

2.10 แสดงข้อมูลตอบกลับเมื่อใช้ฟังก์ชัน Force Single Coil (05).....16

2.11 แสดงการส่งข้อมูลโดยใช้ฟังก์ชัน Preset Single Register (06).....16

2.12 แสดงข้อมูลตอบกลับเมื่อใช้ฟังก์ชัน Preset Single Register (06).....17

2.13 แสดงการส่งข้อมูลโดยใช้ฟังก์ชัน Force Multiple Coils (15,0X0F).....17

2.14 แสดงข้อมูลตอบกลับเมื่อใช้ฟังก์ชัน Force Multiple Coils (15,0X0F).....18

2.15 แสดงการส่งข้อมูลโดยใช้ฟังก์ชัน Preset Multiple Register (16,0X10).....19

2.16 แสดงข้อมูลตอบกลับเมื่อใช้ฟังก์ชัน Preset Multiple Register (16,0X10).....19

2.17 เปรียบเทียบคุณสมบัติของ มาตรฐาน RS-232, RS-422 และ RS-485.....21

2.18 แสดงความสัมพันธ์ของ PV, Error, MV 26

2.19 แสดงความสัมพันธ์ของ PV, Error, MV 27

2.20 แสดงการเปรียบเทียบเอาต์พุต 4-20 mA,รีเลย์ กับ 0 -100%.....37

3.1 PLC I/O Assignment 83

3.2 คุณสมบัติด้านการสื่อสารของ FUJI PXR-4.....97

4.1 แสดงผลการรับส่งข้อมูลกับอุปกรณ์ควบคุมอุณหภูมิ FUJI PXR-4.....107

4.2 แสดงผลการรับส่งข้อมูลกับอุปกรณ์ควบคุม PLC KOYO DO-06DD1.....108

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญภาพ

ภาพที่ หน้า

2.1 การติดต่อสื่อสารแบบ Master-Slave.....4

2.2 ความแตกต่างระหว่าง MODBUS ASCII และ RTU.....4

2.3 เฟรมข้อมูลของ MODBUS RTU.....5

2.4 ข้อมูลแต่ละไบต์ของ MODBUS RTU.....5

2.5 เฟรมข้อมูลของ MODBUS ASCII.....7

2.6 ข้อมูลแต่ละไบต์ของ MODBUS ASCII.....7

2.7 โฟลวชาร์ตขั้นตอนการคำนวณค่า CRC.....9

2.8 ตัวอย่างการคำนวณค่า CRC.....10

2.9 แสดงการต่ออุปกรณ์ด้วยมาตรฐาน RS-485 2 WIRE.....22

2.10 แสดงการต่ออุปกรณ์ด้วยมาตรฐาน RS-485 4 WIRE.....22

2.11 องค์ประกอบของระบบควบคุมแบบป้อนกลับ.....23

2.12 การควบคุมแบบ ON-OFF.....24

2.13 การควบคุมแบบ ON-OFF โดยมี Hysteresis.....25

2.14 แสดงลักษณะของ Proportional Band.....26

2.15 แสดง Proportional Band ที่ bias = 50 %.....27

2.16 แสดง Offset.....28

2.17 สัญญาณควบคุมของ PI Control.....29

2.18 ระบบที่ Oscillate และ Unstable เมื่อตั้งค่า T_I ต่ำเกินไป.....30

2.19 สัญญาณควบคุมของ PD Control.....31

2.20 สัญญาณควบคุมของ PID Control.....32

2.21 แสดงไดอะแกรมของระบบควบคุมแบบ Fuzzy+PID.....34

2.22 แสดงผลตอบสนองของระบบควบคุมแบบ Fuzzy+PID.....34

2.23 แสดงการทำ Auto-Tunning.....35

2.24 แสดงการเปรียบเทียบเอาต์พุต 4-20 mA กับ 0 -100%.....36

2.25 แสดงการเปรียบเทียบเอาต์พุตรีเลย์ กับ 0 -100%.....36

2.26 ไดอะแกรมแสดงโครงสร้างของ PLC.....38

2.27 ภาพเครื่องผสมวัตถุดิบที่ใช้ PLC ในการควบคุม.....40

2.28 ภาพการขนถ่ายผลิตภัณฑ์ที่ใช้ PLC ในการควบคุม.....40

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญภาพ (ต่อ)

ภาพที่ **หน้า**

2.29 รีเลย์ที่ใช้ในงานอุตสาหกรรม....41

2.30 การต่อ RTD 2 สาย....42

2.31 การต่อ RTD 4 สาย....43

2.32 แสดง Sight Glass แบบต่างๆ....44

2.33 แสดงการติดตั้งใช้งานจริง....44

2.34 แสดง sight glass แบบแผ่นแก้วเรียบ....46

2.35 แสดงฮีตเตอร์แท่ง(Cartridge Heater)....47

2.36 แสดงฮีตเตอร์ครีบ(Finned Heater)....47

2.37 แสดงฮีตเตอร์จุ่ม(Immersion Heater)....48

2.38 แสดงฮีตเตอร์บอบบิ้น(Bobbin Heater)....48

2.39 แสดงฮีตเตอร์รัดท่อ (Band Heater)....49

2.40 แสดงฮีตเตอร์แผ่น (Strip Heater)....49

2.41 แสดงฮีตเตอร์อินฟราเรด (Infrared Heater)....49

2.42 แสดงฮีตเตอร์เส้น (Cable heater)....50

2.43 โซลิดสเตทรีเลย์แบบไฟฟ้ากระแสตรง....51

2.44 โซลิดสเตทรีเลย์ที่ขับโหลดไฟสลับ....51

2.45 ส่วนประกอบในหน้าจอ IDE ของ Visual Studio 52

2.46 หน้าจอการออกแบบ User Interface 54

2.47 การตั้งค่าพร็อพเพอตี้ให้กับคอนโทรล....55

2.48 การเขียนโปรแกรมควบคุมการทำงาน....55

2.49 โปรแกรมขณะทดสอบการทำงาน....55

2.50 เรคอร์ดและฟิลด์....57

2.51 ความสัมพันธ์ระหว่างตาราง....58

2.52 หน้าตาโปรแกรม Microsoft SQL Server 2005 Express 59

2.53 หน้าจอ Login เข้าสู่โปรแกรม Microsoft SQL Server Management Studio Express 60

2.54 หน้าตาโปรแกรม Microsoft SQL Server Management Studio Express 61

2.55 โครงสร้างของ ADO.NET 63

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญภาพ (ต่อ)

ภาพที่ **หน้า**

2.56 แสดงหน้าตาโปรแกรม Crystal Report 69

3.1 กระบวนการจำลองที่จะสร้าง.....71

3.2 แสดงถังที่ออกแบบโดยใช้โปรแกรม SolidWORK.....73

3.3 แสดงถังควบคุมอุณหภูมิที่ได้จัดสร้าง.....74

3.4 แสดงฮีตเตอร์ที่ติดตั้งในถังควบคุมอุณภูมิ.....75

3.5 แสดงโซลิดสเตทรีเลย์ CARLO GAVAZZI แบบ Analog switching ที่เลือกใช้.....75

3.6 แสดงRTD PT-100 ที่เลือกใช้.....75

3.7 แสดงเทอร์โมสตัทที่เลือกใช้.....76

3.8 แสดงอุปกรณ์ควบคุมอุณหภูมิ FUJI PXR-4 ที่เลือกใช้.....76

3.9 PLC DirectSOFT KOYO D0-06DDI.....77

3.10 แสดง Specification ของ PLC KOYO D0-6DD1.....78

3.11 แสดง Dimensions ของ PLC KOYO D0-6DD1.....78

3.12 แสดงโครงสร้างของ PLC KOYO D0-6DD1.....79

3.13 แสดงวงจรภายในของภาคอินพุต.....80

3.14 แสดงการต่อ Input.....80

3.15 แสดงวงจรภายในของภาคเอาต์พุต.....81

3.16 แสดงจุดต่อ Output.....81

3.17 แสดงRS-232 to RS-485 Converter รุ่น TNA10A ที่เลือกใช้.....82

3.18 โซลินอยด์วาล์ว (Solenoid Valve).....82

3.19 การเชื่อมต่อระหว่าง PLC KOYO DO-06DD1.....83

3.20 หน้าจอเริ่มต้นโปรแกรม DirectSOFT32.....84

3.21 หน้าจอโปรแกรม DirectSOFT32.....84

3.22 หน้าจอตั้งค่าการติดต่อสื่อสาร.....85

3.23 หน้าจอโปรแกรม DirectSOFT32.....85

3.24 ไดอะแกรมแสดงระบบที่ใช้.....86

3.25 แสดงหน้าจอ Login 87

3.26 หน้าจอ Process monitoring.....87

3.27 หน้าจอ Trend monitoring.....88

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญภาพ (ต่อ)

ภาพที่ **หน้า**

3.28 หน้าจอ Data Report....89

3.29 หน้าจอ User Setting (Add new user)....89

3.30 หน้าจอ User Setting (Delete user)....90

3.31 หน้าจอ Communication Setting....90

3.32 หน้าจอ MODBUS Protocol Test....91

3.33 หน้าจอ PXR4 Temperature Controller Setting....91

3.34 คอนโทรล SerialPort ใน Visual Basic 2008....92

3.35 พร็อพเพอตี้ของคอนโทรล SerialPort ใน Visual Basic 2008....92

3.36 แสดงการสร้าง Database ใหม่....95

3.37 แสดงการกำหนดรายละเอียดของฐานข้อมูล....95

3.38 แสดงการสร้างตาราง....96

3.39 แสดงตารางในฐานข้อมูลที่สร้างเสร็จแล้ว....96

3.40 การเชื่อมต่ออุปกรณ์ควบคุมอุณหภูมิ FUJI PXR4 ผ่าน RS-485....97

3.41 การเชื่อมต่อ PLC KOYO DO-06DD1 ผ่าน RS-485....98

3.42 การตั้งค่า Port2 ให้มีการติดต่อสื่อสารแบบ MODBUS....98

4.1 กราฟแสดงการเข้าสู่ค่า SV เมื่อใช้การควบคุมแบบ P....101

4.2 กราฟแสดงการเข้าสู่ค่า SV เมื่อใช้การควบคุมแบบ PI....102

4.3 กราฟแสดงการเข้าสู่ค่า SV เมื่อใช้การควบคุมแบบ PD....103

4.4 กราฟแสดงการเข้าสู่ค่า SV เมื่อใช้การควบคุมแบบ PID....104

4.5 กราฟแสดงการเข้าสู่ค่า SV เมื่อใช้การควบคุมแบบ Auto-Tuning....105

4.6 กราฟแสดงการเข้าสู่ค่า SV เมื่อใช้การควบคุมแบบ Fuzzy....106

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

1.1 ความสำคัญของปริญญานิพนธ์

ในปัจจุบันการใช้ซอฟต์แวร์คอมพิวเตอร์ และการติดต่อสื่อสารข้อมูลกับอุปกรณ์ควบคุมโดยใช้โปรโตคอลต่าง ๆ ทำให้เราสามารถแสดงผลกระบวนการจากระยะไกล ได้ส่งผลให้ผู้ปฏิบัติงานเกี่ยวข้องกับกระบวนการทางอุตสาหกรรมนั้น ไม่จำเป็นต้องเข้าไปอยู่ในบริเวณที่มีกระบวนการอยู่จริง ก็สามารถสังเกตและควบคุมกระบวนการได้ โดยไม่เกิดอันตรายและผลเสียต่อสุขภาพ ปริญญานิพนธ์นี้จึงได้ทำการศึกษาวิธีการสร้างระบบการแสดงผลกระบวนการ โดยใช้ซอฟต์แวร์คอมพิวเตอร์และการติดต่อสื่อสารข้อมูลกับอุปกรณ์ควบคุม โดยใช้โปรโตคอลที่มีการใช้งานในอุตสาหกรรมอย่างกว้างขวางคือ MODBUS

1.2 วัตถุประสงค์ของปริญญานิพนธ์

1. เพื่อศึกษาวิธีการเขียนโปรแกรมติดต่อกับอุปกรณ์ควบคุมทางอุตสาหกรรมและการแสดงผลกระบวนการผ่านคอมพิวเตอร์
2. เพื่อศึกษารูปแบบและวิธีการใช้งานของโปรโตคอล MODBUS
3. เพื่อศึกษาอุปกรณ์ฮาร์ดแวร์ที่ใช้ในการควบคุมกระบวนการทางอุตสาหกรรม

1.3 ขอบเขตของปริญญานิพนธ์

1. สร้างกระบวนการจำลอง โดยกระบวนการที่สร้างขึ้นนั้นต้องประกอบด้วยองค์ประกอบหลักคือ อุปกรณ์การวัด อุปกรณ์ควบคุม อุปกรณ์ควบคุมตัวสุดท้าย และระบบที่ต้องการควบคุม
2. สร้างระบบควบคุมกระบวนการจำลองโดยใช้ PLC และอุปกรณ์ควบคุมที่สามารถติดต่อสื่อสารผ่านโปรโตคอล MODBUS ได้
3. สร้างโปรแกรมแสดงผลกระบวนการจำลองผ่านคอมพิวเตอร์โดยมีความสามารถในการทำงานคือ
   - สามารถแสดงผลกระบวนการแบบ เสมือนจริง ได้
   - สามารถจัดเก็บข้อมูลที่สำคัญในกระบวนการลงฐานข้อมูลได้
   - สามารถเรียกดูรายงานย้อนหลังของข้อมูลที่สำคัญได้ตามระยะเวลาต้องการ
   - สามารถตั้งค่าพารามิเตอร์ที่สำคัญในกระบวนการจากคอมพิวเตอร์ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.4 ขั้นตอนการศึกษา

1. ศึกษาการทำงานของโปรโตคอลมอดบัส
2. ศึกษาการเขียนโปรแกรมด้วยโปรแกรม Visual Basic 2008
3. ศึกษาการติดต่อสื่อสารระหว่างคอมพิวเตอร์กับอุปกรณ์ควบคุมผ่าน RS-232
4. ศึกษาการใช้งานโปรแกรมออกแบบฐานข้อมูล Microsoft SQL Server 2005
5. ศึกษาการออกแบบระบบควบคุมกระบวนการโดยใช้อุปกรณ์ควบคุมทางอุตสาหกรรมที่ใช้งานจริงในโรงงานอุตสาหกรรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2
ทฤษฎีและหลักการออกแบบ

2.1 การแสดงผลกระบวนการ

กระบวนการการผลิตในภาคอุตสาหกรรมในปัจจุบัน มีหลากหลายรูปแบบบางระบวนการอาจเป็นกระบวนการ ที่อาจก่อให้เกิดอันตรายต่อผู้ที่ทำงานรับผิดชอบโดยตรงกับกระบวนการนั้นๆ การแสดงผลกระบวนการจากระยะไกล จึงเป็นทางเลือกที่เหมาะสมเพื่อป้องกันอันตราย ซึ่งอาจเกิดขึ้นได้ โดยการแสดงผลกระบวนการจะเกิดขึ้นได้ ก็จำเป็นต้องมีอุปกรณ์ที่มีความสามารถในการติดต่อสื่อสาร ทั้งฝั่งอุปกรณ์ควบคุมและอุปกรณ์คอมพิวเตอร์ อีกทั้งยังต้องอาศัยซอฟท์แวร์ในการเขียนโปรแกรมควบคุม และจัดการการติดต่อสื่อสารในโครงงานนี้นั้น ได้ทำการศึกษาการแสดงผลกระบวนการจากระยะไกล โดยได้ทำการออกแบบระบบจำลองและใช้อุปกรณ์การวัดและควบคุมทางอุตสาหกรรม จากนั้นจึงทำการเขียนซอฟแวร์ติดต่ออุปกรณ์ควบคุมในกระบวนการโดยใช้โปรโตคอลมอดบัส และซอฟแวร์ที่สร้างขึ้นนั้น มีฟังก์ชันการทำงานในการจัดการข้อมูลที่เหมาะสมกับกระบวนการดังกล่าว

2.2 โปรโตคอล MODBUS

โปรโตคอล MODBUS เป็นโปรโตคอลเพื่อสื่อสารข้อมูลอินพุต/เอาต์พุต และริจิสเตอร์ภายใน PLC ซึ่งถูกคิดค้นมาตั้งแต่ปี 1979 โดย Modicon ซึ่งเป็นหนึ่งในบริษัทผู้ผลิต PLC ในประเทศสหรัฐอเมริกา (ปัจจุบันเป็นบริษัท Schneider Electric) โปรโตคอล MODBUS ได้เป็นที่ยอมรับกันอย่างกว้างขวาง สาเหตุเนื่องมาจาก MODBUS เป็นระบบเปิด เชื่อมต่อ และพัฒนาง่าย และไม่มีค่าใช้จ่าย (royalty-Free) อีกทั้งยังแพร่หลายในการนำโปรโตคอลนี้ไปใช้งานในอุปกรณ์อื่นๆ อาทิ RTU (Remote terminal Unit), Remote I/O,Digital Power Meter,Flow Computer เป็นต้น และในทุกซอฟต์แวร์ SCADA มีความสามารถสื่อสารกับโปรโตคอลMODBUSได้ ปัจจุบัน MODBUS จึงเป็นโปรโตคอลหลักในงานอุตสาหกรรมเพื่อใช้งานในระบบ SCADA และ PLC

2.2.1 ข้อมูลทางเทคนิคของโปรโตคอล MODBUS

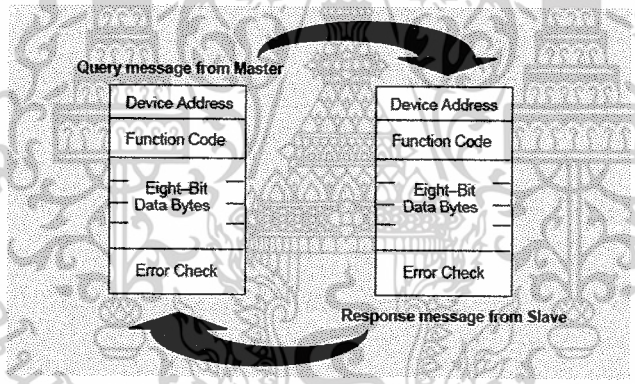
โปรโตคอล MODBUS ได้ออกแบบเพื่อใช้งานงานกับพอร์ตสื่อสารกับอนุกรมมาตรฐานอย่าง RS-232C, RS-422และ RS-485 ในจุดประสงค์แรก จนเกิดการใช้งานอย่างแพร่หลายจึงได้พัฒนาออกไปเป็นโปรโตคอล MODBUS Plus และ MODBUS/TCP อีกด้วย อย่างไรก็ตาม โปรโตคอล MODBUS บนพอร์ตสื่อสารอนุกรมยังเป็นหลักพื้นฐานและครองความนิยมในอุปกรณ์ควบคุมอยู่นั้นเอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรโตคอล MODBUS เป็นการสื่อสารข้อมูลในลักษณะ Master/Slave ซึ่งเป็นการสื่อสารจากอุปกรณ์แม่ (Master) เครื่องเดียว โดยส่วนใหญ่มักเป็นซอฟต์แวร์คอมพิวเตอร์หรืออุปกรณ์แสดงผล HMI ไปยังอุปกรณ์ลูก (Slave) ได้หลาย ๆ เครื่อง โดยสามารถกำหนดหมายเลขอุปกรณ์ได้สูงสุด 255 เครื่อง โดยมีลักษณะการส่งข้อมูล 2 แบบ คือ ข้อมูลแบบแอสกี (ASCII) และ ข้อมูลแบบเลขฐาน 2 (Binary) ในโปรโตคอล MODBUS ที่สื่อสารข้อมูลแบบ ASCII จะเรียกว่า MODBUS ASCII และโปรโตคอล MODBUS ที่สื่อสารข้อมูลแบบเลขฐาน 2 จะเรียกว่า MODBUS RTU ทำให้มีความแตกต่างในการกำหนดค่าพอร์ตสื่อสาร

2.2.2 โหมดการรับส่งข้อมูลอนุกรม

การรับส่งข้อมูลอนุกรมด้วยโปรโตคอล MODBUS สามารถเลือกได้ 2 โหมด คือโหมด RTU และโหมด ASCII ซึ่งทั้ง 2 โหมดนี้มีความแตกต่างกันที่การกำหนดรูปแบบของชุดข้อมูล ภายในเฟรมจะเลือกโหมดใดก็ได้ แต่มีเงื่อนไขว่าอุปกรณ์ทุกตัวที่ต่อร่วมอยู่ในบัสหรือเครือข่ายเดียวกัน จะต้องถูกปรับตั้งให้เลือกใช้โหมดเดียวกันทั้งหมด



ภาพที่ 2.1 การติดต่อสื่อสารแบบ Master-Slave

Properties of Modbus/ASCII and Modbus/RTU

	Modbus/ASCII		Modbus/RTU	
Characters	ASCII **0...9** and **A..F**		Binary **0...255**	
Error check	**LRC** Longitudinal Redundancy Check		**CRC** Cyclic Redundancy Check	
Frame start	character **':'**		3.5 chars silence	
Frame end	characters **CR/LF**		3.5 chars silence	
Gaps in message	1 sec		1.5 times char length	
Start bit	**1**		**1**	
Data bits	**7**		**8**	
Parity	even/odd	none	even/odd	none
Stop bits	**1**	**2**	**1**	**2**

ภาพที่ 2.2 ความแตกต่างระหว่าง MODBUS ASCII และ RTU

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

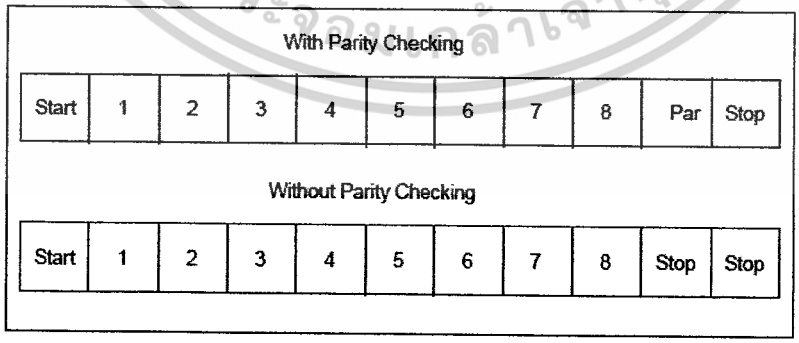
2.2.3 MODBUS RTU

เฟรมข้อมูลในโหมด RTU ประกอบด้วยข้อมูลแสดงตำแหน่งแอดเดรสของสเลฟ 1 ไบต์ หมายเลขฟังก์ชัน 1 ไบต์ ข้อมูลที่ทำการรับส่งจำนวนมากสุดไม่เกิน252 ไบต์ และรหัสตรวจสอบความถูกต้องของข้อมูลแบบ CRC (Cyclical Redundancy Checking) ขนาด 2 ไบต์ ค่า CRC นี้เป็นค่าที่คำนวณมาจากข้อมูลทุกไบต์ ไม่รวมบิต start, Stop และ Parity Check โดยที่ตัวสเลฟตัวที่ส่งข้อมูลออกมาจะสร้างรหัส CRC แล้วส่งตามท้ายไบต์ข้อมูลออกมา หลังจากนั้นเมื่อมาสเตอร์ได้รับเฟรมข้อมูล และถอดข้อมูลออกจากเฟรมแล้วจะทำการคำนวณค่า CRC ตามสูตรเดียวกับสเลฟ เพื่อทำการเปรียบเทียบค่า CRC ทั้ง 2 ค่า ว่าตรงกันหรือไม่ หากไม่ตรงกันแสดงว่าเกิดความผิดพลาดในการรับส่งข้อมูล

ในโหมด RTU การส่งข้อมูล 1 ไบต์ ไม่ว่าจะเป็นข้อมูลส่วนใดภายในเฟรม จะต้องทำการส่งบิตข้อมูลมูลรวม 11 คือ บิต เริ่มต้น (Start) 1 บิต, บิตข้อมูล 8 บิต ซึ่งก็คือ 1 ไบต์, บิตตรวจสอบ Parity ของข้อมูล 1 บิตและบิตหยุด (Stop) 1 บิต หรือหากเลือกแบบไม่มีบิต Parity ก็จะเป็นแบบ Stop แทน2 บิต สำหรับการกำหนดให้มีบิต Parity นั้น สามารถเลือกเป็นแบบคู่ (Even Parity) หรือคี่ (Odd Parity) ก็ได้ และหากต้องการออกแบบให้สอดคล้องกับอุปกรณ์ที่มีใช้กันทั่วไปมากที่สุดควรเลือกแบบคู่ โดยที่สามารถปรับเปลี่ยนเป็นแบบคี่ หรือไม่มีการตรวจสอบ Parity (No Parity) ได้ด้วย

START	ADDRESS	FUNCTION	DATA	CRC CHECK	END
T1–T2–T3–T4	8 BITS	8 BITS	*n* x 8 BITS	16 BITS	T1–T2–T3–T4

ภาพที่ 2. 3 เฟรมข้อมูลของ MODBUS RTU



ภาพที่ 2.4 ข้อมูลแต่ละไบต์ของ MODBUS RTU

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปแสดงช่วงเวลาที่เหมาะสมในกระบวนการส่งเฟรมข้อมูลออกมาในบัสข้อมูล เมื่อส่งเฟรมข้อมูลออกไป 1 เฟรมแล้ว จะต้องรออย่างน้อยเท่ากับเวลาที่ใช้ส่งข้อมูลจำนวน 3.5 ตัวอักษรจึงจะส่งข้อมูลเฟรมต่อไปได้ และภายในเฟรมแต่ละเฟรม ซึ่งประกอบด้วยชุดบิตข้อมูลจำนวนหลายชุดก็จะอยู่ห่างกันไม่เกิน 1.5 บิต วัตถุประสงค์ในการกำหนดช่วงเวลาระหว่างเฟรมข้อมูล และชุดบิตข้อมูลภายในเฟรมก็เพื่อให้อุปกรณ์ต่าง ๆ ไม่ว่าจะเป็นมาสเตอร์หรือสเลฟสามารถรับรู้ถึงจุดเริ่มต้นและจุดสิ้นสุดของเฟรมข้อมูลแต่ละเฟรมได้ และสามารถตรวจสอบได้ว่าการรับส่งข้อมูลในขณะนั้น เกิดความผิดพลาดขึ้นมาหรือไม่ โดยตรวจสอบกับช่วงระยะห่างของเวลาที่ควรจะเป็นกับค่าที่วัดได้จริง

2.2.4 MODBUS ASCII

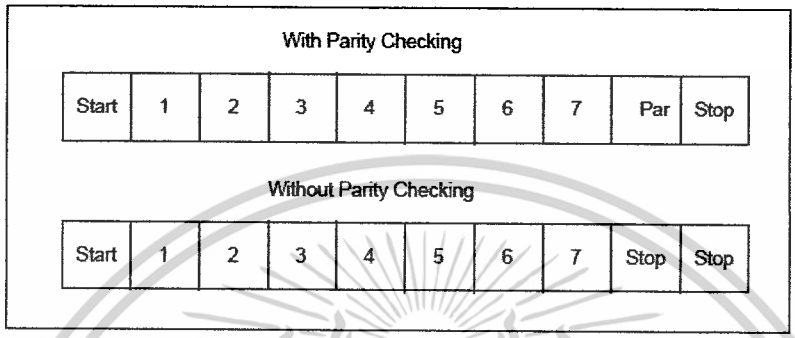
การรับส่งข้อมูลในโหมด ASCII มีความแตกต่างจากโหมด RTUตรงที่ในโหมด RTU ข้อมูลที่จะส่งขนาด 1 ไบต์นำมารวมกับบิตประกอบต่าง ๆ ก็สามารถส่งออกไปได้เลยแต่สำหรับโหมด ASCII จะมองข้อมูล 1 ไบต์นั้นออกมาเป็นตัวอักษร 2 ตัว เช่น ค่า 0x5B ซึ่งเป็นเลขฐานสิบหก ก็จะถูกมองเป็นตัวอักษร '5' และตัวอักษร 'B' จากนั้นก็จะทำการค้นหารหัส ASCII ของตัวอักษรทั้ง 2 ตัวนั้น ซึ่งได้แก่ 0x35 สำหรับ '5' และ 0x42 สำหรับ 'B' แล้วทำการส่งรหัส ASCII ทั้ง 2 ค่านี้ออกไป ซึ่งจะได้ผลเท่ากับการส่งค่า 0x5B ซึ่งเป็นข้อมูลขนาด 1 ไบต์ ในโหมด RTU

จะเห็นได้ว่าการส่งข้อมูลในโหมด ASCII จะต้องทำงานมากกว่าการส่งข้อมูลในโหมด RTU ซึ่งทำให้อัตราเร็วในการสื่อสารมีค่าต่ำกว่าสาเหตุที่เป็นแบบนี้ก็เพราะโหมด ACSII ได้ถูกออกแบบมาสำหรับอุปกรณ์ที่ไม่มีความสามารถในหากำหนดช่วงระยะห่างของเวลา ในการส่งเฟรมข้อมูล อย่างเช่นในโหมด RTU ที่อุปกรณ์สามารถกำหนดได้ว่าจะส่งเฟรมข้อมูลแต่ละเฟรมออกมาด้วยเวลาห่างกันเท่าใดและอุปกรณ์ที่รอรับข้อมูล ก็ต้องสามารถตรวจจับแยกแยะได้ด้วยว่าเฟรมข้อมูลแต่ละเฟรมที่รับเข้ามานั้น มีระยะเวลาห่างกันภายในช่วงเวลาที่กำหนดหรือไม่ เพื่อทำให้สามารถตรวจสอบหาจุดเริ่มต้นและจุดสิ้นสุดของเฟรมข้อมูลแต่ละเฟรมได้ แต่ในความเป็นจริงยังมีอุปกรณ์อีกหลายชนิดที่ไม่มีความสามารถพิเศษนี้ จึงต้องใช้วิธีอื่นที่จะช่วยให้สามารถรับรู้จุดเริ่มต้นและจุดสิ้นสุดของเฟรมข้อมูลได้ได้แก่โหมด ASCII ซึ่งในโหมดนี้จะเริ่มต้นเฟรมข้อมูลด้วยการส่งรหัส ASCII ที่กำหนดให้ หมายถึงจุดเริ่มต้นคือ 0x3A ซึ่งตรงกับอักษร ':' ตามด้วยแอดเดรสของสเลฟ หมายเลขฟังก์ชัน ข้อมูล รหัสตรวจสอบ LRC รหัสASCII 2 ตัว ที่กำหนดให้หมายถึงจุดสิ้นสุด คือ รหัส 0x0D และ 0x0A คือ รหัสCR (Carriage Return) และ LF (Line Feed) ตามลำดับ (ดูภาพที่6) โดยขณะที่บัสข้อมูลว่างจากการรับส่งข้อมูล อุปกรณ์ทุกตัวจะคอยตรวจสอบข้อมูลในบัสว่ามีการส่งรหัส ASCII ของ ':' ออกมาหรือไม่ ถ้ามีก็จะรับรู้ว่าขณะนี้ได้มีการเริ่มต้นส่งเฟรมข้อมูลออกมาแล้ว ก็จะเข้าสู่ กระบวนการรับข้อมูลต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

START	ADDRESS	FUNCTION	DATA	LRC CHECK	END
1 CHAR :	2 CHARS	2 CHARS	n CHARS	2 CHARS	2 CHARS CRLF

ภาพที่ 2.5 เฟรมข้อมูลของ MODBUS ASCII



ภาพที่ 2.6 ข้อมูลแต่ละไบต์ของ MODBUS ASCII

จากรูปแสดงชุดของบิต ข้อมูลที่ส่งทั้งหมด ของข้อมูลแต่ละอักษร จะเห็นได้ว่าหน่วยของชุดข้อมูลในโหมด ASCII คือ ตัวอักษรไม่เหมือนในโหมด RTU ที่มีหน่วยเป็นไบต์ เพราะโหมด ASCII เป็นการส่งข้อมูล ในรูปแบบของรหัส ASCII ของตัวอักษร ซึ่งสามารถกำหนดได้ด้วยบิตข้อมูลจำนวน 7 บิตไม่ต้องใช้ถึง 8 บิต ดังนั้นบิตที่ต้องส่งต่อการส่งรหัส ASCII 1 ตัว ได้แก่ บิต Start 1 บิต, บิตข้อมูลรหัส ASCII 7 บิต,บิตตรวจสอบ Parity 1 บิต และบิต Stop 1 บิต รวมทั้งหมด 0 บิต และเช่นเดียวกับ โหมด RTU คือ สามารถเลือกประเภทของบิตตรวจสอบ Parity 1 บิต ได้ว่าเป็นแบบคู่, คี่หรือไม่มีบิตตรวจสอบ ซึ่งจะเปลี่ยนเป็นบิต Stop แทนเนื่องจากการส่งข้อมูลในโหมด ASCII เป็นการแปลงข้อมูลจากเลขฐานสิบหก เป็นรหัส ASCII ของตัวอักษรที่แสดงค่าเลขฐานสิบหก ดังนั้นรหัส ASCII ที่ปรากฏในบัสข้อมูลนอกเหนือไปจากรหัส เริ่มต้นและรหัสสิ้นสุดแล้วจะเป็นรหัส ASCII ของตัวอักษรตั้งแต่ '0'ถึง'9' และ'A'ถึง'F' เท่านั้น

ถึงแม้โหมด ASCII จะไม่ต้องกำหนดช่วงระยะห่างทางเวลาของเฟรมข้อมูลแต่ละเฟรม แต่อุปกรณ์ยังต้องสามารถตรวจจับช่วงระยะห่างทางเวลาระหว่างการส่งข้อมูลรหัส ASCII แต่ละตัวได้ ซึ่งหากเว้นช่วงห่างกันนานเกินไป แสดงว่าเกิดความผิดพลาดในการสื่อสาร โดยปกติจะกำหนดค่าเวลานี้ไว้ที่ 1 วินาที เรียกค่าเวลานี้ว่า Time Out Period หากเปรียบเทียบระหว่างเฟรมข้อมูลในโหมด ASCII กับโหมด RTU จะพบว่าการส่งข้อมูลในโหมดASCII นั้นหากต้องการส่งไบต์ข้อมูลให้ได้เท่ากับโหมด RTU จะต้องส่งข้อมูลรหัส ASCII ออกไปเป็นจำนวน 2 เท่าของจำนวนไบต์ข้อมูลเช่นในโหมด RTUเฟรมข้อมูล 1 เฟรมสามารถส่งข้อมูลได้มากสุด 252 ไบต์ซึ่งหากเป็นโหมด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ASCII จะต้องส่งข้อมูลตัวอักษรออกไปทั้งหมด 2x252 เท่ากับ 504 ตัวอักษร และเพื่อให้มาตรฐานขนาดของเฟรมข้อมูลของทั้ง 2 โหมด มีขนาดเท่ากันจึงกำหนดให้ค่า 504 เป็นค่าจำนวนตัวอักษรมากสุดในการส่งเฟรมข้อมูลด้วยโหมด ASCII

2.2.5 การตรวจสอบความผิดพลาดของข้อมูล

ในโปรโตคอลแบบ MODBUS ASCII มีการตรวจสอบความผิดพลาดของข้อมูล ในลักษณะ LRC (Longitudinal Redundancy Check) ซึ่งเป็นอัลกอริทึมที่นำค่าข้อมูลของตัวอักษร ASCII ที่ส่งออกไปตั้งแต่ตัวแรกนำมาผ่านลอจิก (Exclusive -OR (XOR) ไปเรื่อยจนถึงตัวสุดท้าย จนค่าตัวอักษรBCC (Block Check Character) จำนวน 2 ตัวอักษร และใน MODBUS RTU มีการตรวจสอบความผิดพลาดของข้อมูลในลักษณะ CRC-16(Cyclic Redundancy Check) ซึ่งเป็นอัลกอริทึมที่นำข้อมูลแต่ละไบต์มาหาร $X^2 + X^2 + X^2 + 1$ ในรูปสมการโพลิโนเมียลเลขฐาน 2

ปกติความผิดพลาด (Error) จะมีอยู่สองลักษณะ คือ Communication error และ Operation error ในกรณี Communication error ถ้าสเลฟพบว่ามีข้อผิดพลาดมันจะไม่ส่งคำตอบให้มาสเตอร์ และมาสเตอร์ก็จะส่งคำถามใหม่ให้ไป เมื่อมันรอคำตอบจนถึงเวลาที่ตั้งไว้ หรือเมื่อพบว่ามีความผิดพลาดสเลฟ อาจจะส่งข้อความแจ้งความผิดพลาดกลับมาให้มาสเตอร์ก็ได้ การตรวจสอบความผิดพลาดของข้อมูลมีดังนี้

2.2.5.1 Parity จะใช้วิธีเพิ่มจำนวนบิต 1 บิตเข้าไปในเฟรมของข้อมูล ในแต่ละไบต์ที่ส่งไปอาจเป็น “1” หรือ “0” แล้วแต่ชนิดของ พาริตี้ที่ใช้ ซึ่งวิธีนี้จะตรวจสอบได้เฉพาะความผิดพลาดที่เกิดขึ้นเพียง 1 บิตเท่านั้น ถ้าความผิดพลาดเกิดขึ้นมากกว่า 1 บิต จะไม่สามารถบอกได้ การตรวจสอบพาริตี้จะมีอยู่ 2 แบบ คือ

- การตรวจสอบพาริตี้คู่ (Even Parity) จะมีอยู่ในตัวอักษรที่ต้องการตรวจสอบรวมทั้งพาริตี้บิตจะต้องมีจำนวนคู่ (2, 4, 6, 8)
- การตรวจพาริตี้คี่ (Odd Parity) จะมีจำนวนบิต 1 เป็นจำนวนคี่ (1, 3, 5, 7)

2.2.5.2 Redundancy check สามารถตรวจสอบความผิดพลาดที่เกิดขึ้นหลายบิตได้ มีอยู่ด้วยกันสองแบบคือ Cyclical Redundancy check (CRC) ซึ่งใช้กับโหมด RTU และแบบ Longitudinal Redundancy check (LRC) ที่ใช้กับโหมด ASCII ในการส่งข้อมูลระหว่างกันตัวส่งจะส่งข้อมูลพร้อมกัน CRC/LRC ออกไปส่วนตัวรับก็จะรับข้อมูล และคำนวณ CRC/LRC ออกมา แล้วเปรียบเทียบกับ CRC/LRC ที่ส่งมาว่าตรงกันหรือไม่ ถ้าไม่ตรงแสดงว่ามีการผิดพลาด

2.2.5.3 Cyclical Redundancy Check (CRC) มีขนาด 16 บิต และมีวิธีการคำนวณดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. โหลดค่าคงที่ FFFFH (16 บิต) จาก Accumulator (ACC)

2. นำข้อมูล 8 บิต (ที่จะส่ง) มา EX-OR

3. Shift right 1 บิต

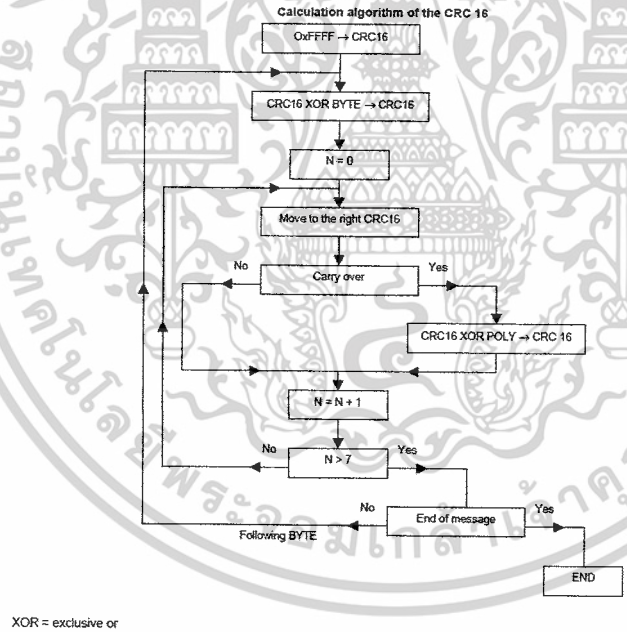
4. ถ้าบิตที่ shift มาเป็น “1” ให้นำ A001H มา EX-OR กับ ACC แต่ถ้าบิตที่ shift มาเป็น “0” ไม่ต้องทำอะไร

5. ถ้ายังไม่ครบ 8 บิต ให้กลับไปทำข้อ 3

6. ใช้ค่าผลลัพธ์ที่ได้ใน ACC เป็นตัวตั้งแทน FFFFH ในการคำนวณสำหรับข้อมูล 8บิตตัวต่อไป

7. ทำเช่นนี้จนครบทุกไบต์ของเพคเกจ

- หมายเหตุ การคำนวณ CRC จะใช้เฉพาะข้อมูล 8 บิตเท่านั้น . โดยบิต starts parity, stop จะไม่นำมาคำนวณ



XOR = exclusive or
N = number of information bits
POLY = calculation polynomial of the CRC 16 = 1010 0000 0000 0001
(Generating polynomial = $1 + x_2 + x_{15} + x_{16}$)
In the CRC 16, the 1st byte transmitted is the least significant one.

ภาพที่ 2.7 โฟลวชาร์ตขั้นตอนการคำนวณค่า CRC

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Example of CRC calculation (frame 02 07)

CRC register initialization		1111	1111	1111	1111
XOR 1st character		0000	0000	0000	0010
		1111	1111	1111	1101
	Move 1	0111	1111	1111	1110|1
Flag to 1, XOR polynomial		1010	0000	0000	0001
		1101	1111	1111	1111
	Move 2	0110	1111	1111	1111|1
Flag to 1, XOR polynomial		1010	0000	0000	0001
		1100	1111	1111	1110
	Move 3	0110	0111	1111	1111|0
	Move 4	0011	0011	1111	1111|1
		1010	0000	0000	0001
		1001	0011	1111	1110
	Move 5	0100	1001	1111	1111|0
	Move 6	0010	0100	1111	1111|1
		1010	0000	0000	0001
		1000	0100	1111	1110
	Move 7	0100	0010	0111	1111|0
	Move 8	0010	0001	0011	1111|1
		1010	0000	0000	0001
		1000	0001	0011	1110
XOR 2nd character		0000	0000	0000	0111
		1000	0001	0011	1001
	Move 1	0100	0000	1001	1100|1
		1010	0000	0000	0001
		1110	0000	1001	1101
	Move 2	0111	0000	0100	1110|1
		1010	0000	0000	0001
		1101	0000	0100	1111
	Move 3	0110	1000	0010	0111|1
		1010	0000	0000	0001
		1100	1000	0010	0110
	Move 4	0110	0100	0001	0011|0
	Move 5	0011	0010	0000	1001|1
		1010	0000	0000	0001
		1001	0010	0000	1000
	Move 6	0100	1001	0000	0100|0
	Move 7	0010	0100	1000	0010|0
	Move 8	0001	0010	0100	0001|0
				Most significant	least significant

The CRC 16 of the frame is then: 4112

ภาพที่ 2.8 ตัวอย่างการคำนวณค่า CRC

2.2.5.4 Longitudinal Redundancy Check (LRC) มีขนาด 8 บิต มีขั้นตอนการคำนวณดังนี้

Transmitter

1. หาผลรวม (SUM) ของข้อมูลทุกไบต์ที่จะส่งไป
2. นำผลลัพธ์จากข้อ 1 มาหาค่า 2'S COMPLEMENT จะได้เป็น LRC ส่งไปพร้อมกับข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Receiver

1. รับค่าทั้งหมดมาแล้วหาผลรวม (SUM) รวม LRC ด้วย
2. ถ้าผลลัพธ์ที่ได้เป็น 00 แสดงว่าไม่มีการผิดพลาด

2.3 MODBUS FUNCTION CODE

2.3.1 Read Coil Status (01)

อ่านสถานะ ON/OFF ของเอาต์พุตจากสเลฟ

การส่งค่า

การใช้ฟังก์ชัน 01 ต้องกำหนดจุดเริ่มต้นและจำนวนของเอาต์พุตที่จะทำการอ่าน ตัวอย่างนี้เป็นการอ่านคอยล์ 20-56 รวมทั้งสิ้น 37 คอยล์ จากสเลฟหมายเลข 3

ตารางที่ 2.1 แสดงการส่งข้อมูลโดยใช้ฟังก์ชัน Read Coil Status (01)

Field Name	Example (Hex)	ASCII Characters	RTU 8-Bit Field (Hex)
Header		: (colon)	None
Slave Address	0X03	0 3	0X03
Function	0X01	0 1	0X01
Starting Address Hi	0X00	0 0	0X00
Starting Address Lo	0X13	1 3	0X13
No. of Registers Hi	0X00	0 0	0X00
No. of Registers Lo	0X25	2 5	0X25
Error Check		LRC (2 chars.)	CRC (16 bits)
Trailer		CR LF	None
	Total Bytes:	17	8

ผลตอบสนอง

สถานะของคอยล์ที่ทำการอ่านจะรวมเป็นชุดข้อมูลจำนวน 8 บิต โดยแต่ละบิตจะแทนสถานะของคอยล์ คือ 1 = ON , 0 = OFF

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.2 แสดงข้อมูลตอบกลับเมื่อใช้ฟังก์ชัน Read Coil Status (01)

Field Name	Example (Hex)	ASCII Characters	RTU 8-Bit Field (Hex)
Header		: (colon)	None
Slave Address	0X03	0 3	0X03
Function	0X01	0 1	0X01
Byte Count	0X05	0 5	0X05
Data 1	0X53	5 3	0X53
Data 2	0X6B	6 B	0X6B
Data 3	0X01	0 1	0X01
Data 4	0XF4	F 4	0XF4
Data 5	0X1B	1 B	0X1B
Error Check		LRC (2 chars.)	CRC (16 bits)
Trailer		CR LF	None
	Total Bytes:	21	10

2.3.2 Read Input Status (02)

อ่านสถานะ ON/OFF ของอินพุตจากสเลฟ

การส่งค่า

การใช้ฟังก์ชัน 02 ต้องกำหนดจุดเริ่มต้นและจำนวนของอินพุตที่จะทำการอ่านตัวอย่างนี้เป็นการอ่านคอยล์ 10101-10120 รวมทั้งสิ้น 20 อินพุต จากสเลฟหมายเลข 3

ตารางที่ 2.3 แสดงการส่งข้อมูลโดยใช้ฟังก์ชัน Read Input Status (02)

Field Name	Example (Hex)	ASCII Characters	RTU 8-Bit Field (Hex)
Header		: (colon)	None
Slave Address	0X03	0 3	0X03
Function	0X02	0 2	0X02
Starting Address Hi	0X00	0 0	0X00
Starting Address Lo	0X64	6 4	0X64
No. of Registers Hi	0X00	0 0	0X00
No. of Registers Lo	0X14	1 4	0X14
Error Check		LRC (2 chars.)	CRC (16 bits)
Trailer		CR LF	None
	Total Bytes:	17	8

ผลตอบสนอง

สถานะของคอยล์ที่ทำการอ่านจะรวมเป็นชุดข้อมูลจำนวน 8 บิต โดยแต่ละบิตจะแทนสถานะของคอยล์ คือ 1 = ON , 0 = OFF

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.4 แสดงข้อมูลตอบกลับเมื่อใช้ฟังก์ชัน Read Input Status (02)

Field Name	Example (Hex)	ASCII Characters	RTU 8-Bit Field (Hex)
Header		: (colon)	None
Slave Address	0X03	0 3	0X03
Function	0X02	0 2	0X02
Byte Count	0X03	0 3	0X03
Data 1	0X53	5 3	0X53
Data 2	0X6B	6 B	0X6B
Data 3	0X01	0 1	0X01
Error Check		LRC (2 chars.)	CRC (16 bits)
Trailer		CR LF	None
	Total Bytes:	17	8

2.3.3 Read Holding Register (03)

อ่านค่าไบนารี่ที่บรรจุอยู่ภายในโฮลดิ้งรีจิสเตอร์จากสเลฟ

การส่งค่า

การใช้ฟังก์ชัน 03 ต้องกำหนดจุดเริ่มต้นและจำนวนของรีจิสเตอร์ที่จะทำการอ่าน ตัวอย่างนี้เป็นการอ่านรีจิสเตอร์ 40201-40203 รวม 3 รีจิสเตอร์จากสเลฟหมายเลข 7

ตารางที่ 2.5 แสดงการส่งข้อมูลโดยใช้ฟังก์ชัน Read Holding Register (03)

Field Name	Example (Hex)	ASCII Characters	RTU 8-Bit Field (Hex)
Header		: (colon)	None
Slave Address	0X07	0 7	0X07
Function	0X03	0 3	0X03
Starting Address Hi	0X00	0 0	0X00
Starting Address Lo	0XC8	C 8	0XC8
No. of Registers Hi	0X00	0 0	0X00
No. of Registers Lo	0X03	0 3	0X03
Error Check		LRC (2 chars.)	CRC (16 bits)
Trailer		CR LF	None
	Total Bytes:	17	8

ผลตอบสนอง

ข้อมูลในรีจิสเตอร์ที่ส่งกลับมาจะเก็บอยู่ในรูปรีจิสเตอร์ 16 บิต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.6 แสดงข้อมูลตอบกลับเมื่อใช้ฟังก์ชัน Read Holding Register (03)

Field Name	Example (Hex)	ASCII Characters	RTU 8-Bit Field (Hex)
Header		: (colon)	None
Slave Address	0X07	0 7	0X07
Function	0X03	0 3	0X03
Byte Count	0X06	0 6	0X06
Data 1 Hi	0X03	0 3	0X03
Data 1 Lo	0XE8	E 8	0XE8
Data 2 Hi	0X01	0 1	0X01
Data 2 Lo	0XF4	F 4	0XF4
Data 3 Hi	0X00	0 0	0X00
Data 3 Lo	0X0A	0 A	0X0A
Error Check		LRC (2 chars.)	CRC (16 bits)
Trailer		CR LF	None
	Total Bytes:	23	11

2.3.4 Read Input Register (04)

อ่านค่าไบนารี่ที่บรรจุอยู่ภายในอินพุตรีจิสเตอร์จากสเลฟ

การส่งค่า

การใช้ฟังก์ชัน 04 ต้องกำหนดจุดเริ่มต้นและจำนวนของอินพุตรีจิสเตอร์ที่จะทำการอ่าน ตัวอย่างนี้เป็นการอ่านรีจิสเตอร์ 30301-30303 รวม 3 รีจิสเตอร์จากสเลฟ์หมายเลข 7

ตารางที่ 2.7 แสดงการส่งข้อมูลโดยใช้ฟังก์ชัน Read Input Register (04)

Field Name	Example (Hex)	ASCII Characters	RTU 8-Bit Field (Hex)
Header		: (colon)	None
Slave Address	0X07	0 7	0X07
Function	0X04	0 4	0X04
Starting Address Hi	0X01	0 1	0X01
Starting Address Lo	0X2C	2 C	0X2C
No. of Registers Hi	0X00	0 0	0X00
No. of Registers Lo	0X03	0 3	0X03
Error Check		LRC (2 chars.)	CRC (16 bits)
Trailer		CR LF	None
	Total Bytes:	17	8

ผลตอบสนอง

ข้อมูลในรีจิสเตอร์ที่ส่งกลับมาจะอยู่ในรูปรีจิสเตอร์ 16 บิต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.8 แสดงข้อมูลตอบกลับเมื่อใช้ฟังก์ชัน Read Input Register (04)

Field Name	Example (Hex)	ASCII Characters	RTU 8-Bit Field (Hex)
Header		: (colon)	None
Slave Address	0X07	0 7	0X07
Function	0X04	0 4	0X04
Byte Count	0X06	0 6	0X06
Data 1 Hi	0X03	0 3	0X03
Data 1 Lo	0XE8	E 8	0XE8
Data 2 Hi	0X01	0 1	0X01
Data 2 Lo	0XF4	F 4	0XF4
Data 3 Hi	0X00	0 0	0X00
Data 3 Lo	0X0A	0 A	0X0A
Error Check		LRC (2 chars.)	CRC (16 bits)
Trailer		CR LF	None
	Total Bytes:	23	11

2.3.5 Force Single Coil (05)

กำหนดสถานะคอยล์เอาต์พุตเป็น ON หรือ OFF

การส่งค่า

การใช้งานฟังก์ชัน 05 ต้องกำหนดคอยล์อ้างอิง โดยการกำหนดจุดเริ่มต้นของคอยล์ และสถานะที่จะทำการกำหนด โดยกำหนดค่า 00FF เพื่อจะทำการ ON คอยล์ และ 0000 เพื่อจะทำการ OFF คอยล์ในตัวอย่างเป็นการกำหนดสถานะ ON ให้คอยล์ 150 ของสเลฟหมายเลข 3

ตารางที่ 2.9 แสดงการส่งข้อมูลโดยใช้ฟังก์ชัน Force Single Coil (05)

Field Name	Example (Hex)	ASCII Characters	RTU 8-Bit Field (Hex)
Header		: (colon)	None
Slave Address	0X03	0 3	0X03
Function	0X05	0 5	0X05
Starting Address Hi	0X00	0 0	0X00
Starting Address Lo	0X95	9 5	0X95
No. of Registers Hi	0XFF	F F	0XFF
No. of Registers Lo	0X00	0 0	0X00
Error Check		LRC (2 chars.)	CRC (16 bits)
Trailer		CR LF	None
	Total Bytes:	17	8

ผลตอบสนอง

ผลตอบสนองจะเป็นการส่งข้อมูลที่ส่งไปกลับมาอีกครั้งหนึ่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.10 แสดงข้อมูลตอบกลับเมื่อใช้ฟังก์ชัน Force Single Coil (05)

Field Name	Example (Hex)	ASCII Characters	RTU 8-Bit Field (Hex)
Header		: (colon)	None
Slave Address	0X03	0 3	0X03
Function	0X05	0 5	0X05
Starting Address Hi	0X00	0 0	0X00
Starting Address Lo	0X95	9 5	0X95
No. of Registers Hi	0XFF	F F	0XFF
No. of Registers Lo	0X00	0 0	0X00
Error Check		LRC (2 chars.)	CRC (16 bits)
Trailer		CR LF	None
	Total Bytes:	17	8

2.3.6 Preset Single Register (06)

กำหนดค่าไปยังโฮลดิ้งรีจิสเตอร์

การส่งค่า

การใช้งานที่ฟังก์ชัน 06 ต้องกำหนดรีจิสเตอร์อ้างอิงที่จะทำการกำหนดค่า โดยกำหนดค่าในรูปของข้อมูลชุดละ 16 บิต จากตัวอย่างเป็นการกำหนดค่าให้กับรีจิสเตอร์ 1000 ถึง 150 ให้กับสเลฟหมายเลข 3

ตารางที่ 2.11 แสดงการส่งข้อมูลโดยใช้ฟังก์ชัน Preset Single Register (06)

Field Name	Example (Hex)	ASCII Characters	RTU 8-Bit Field (Hex)
Header		: (colon)	None
Slave Address	0X03	0 3	0X03
Function	0X06	0 6	0X06
Starting Address Hi	0X00	0 0	0X00
Starting Address Lo	0X95	9 5	0X95
No. of Registers Hi	0X03	0 3	0X03
No. of Registers Lo	0XE8	E 8	0XE8
Error Check		LRC (2 chars.)	CRC (16 bits)
Trailer		CR LF	None
	Total Bytes:	17	8

ผลตอบสนอง

ผลตอบสนองจะเป็นการส่งข้อมูลที่ส่งไปกลับมาอีกครั้งหนึ่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

ตารางที่ 2.12 แสดงข้อมูลตอบกลับเมื่อใช้ฟังก์ชัน Preset Single Register (06)

Field Name	Example (Hex)	ASCII Characters	RTU 8-Bit Field (Hex)
Header		: (colon)	None
Slave Address	0X03	0 3	0X03
Function	0X06	0 6	0X06
Starting Address Hi	0X00	0 0	0X00
Starting Address Lo	0X95	9 5	0X95
No. of Registers Hi	0X03	0 3	0X03
No. of Registers Lo	0XE8	E 8	0XE8
Error Check		LRC (2 chars.)	CRC (16 bits)
Trailer		CR LF	None
	Total Bytes:	17	8

2.3.7 Force Multiple Coils (15,0X0F)

กำหนดค่าคอยล์เอาต์พุตหลายๆตัวให้มีสถานะ ON หรือ OFF

การส่งค่า

การใช้งานต้องกำหนดคอยล์เริ่มต้นและสถานะที่ต้องการกำหนดจากตัวอย่างแสดงการกำหนดให้กับคอยล์ 10 ตัว เริ่มต้นที่คอยล์ 20 ให้กับสเลฟหมายเลข 5

ตารางที่ 2.13 แสดงการส่งข้อมูลโดยใช้ฟังก์ชัน Force Multiple Coils (15,0X0F)

Bit	1	1	0	1	0	0	0	1	0	0	0	0	0	1	0	1
Coil	27	26	25	24	23	22	21	20	...	...	...	...	...	30	29	28

Field Name	Example (Hex)	ASCII Characters	RTU 8-Bit Field (Hex)
Header		: (colon)	None
Slave Address	0X05	0 5	0X05
Function	0X0F	0 F	0X0F
Starting Address Hi	0X00	0 0	0X00
Starting Address Lo	0X13	1 3	0X13
Quantity of Coils Hi	0X00	0 0	0X00
Quantity of Coils Lo	0X0B	0 B	0X0B
Byte Count	0X02	0 2	0X02
Force Data Hi	0XD1	D 1	0XD1
Force Data Lo	0X05	0 5	0X05
Error Check		LRC (2 chars.)	CRC (16 bits)
Trailer		CR LF	None
	Total Bytes:	23	11

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
104120

ผลตอบสนอง

ผลตอบสนองปกติที่ส่งค่าคืนมาจะเป็นค่าแอดเดรส , ฟังก์ชันโค้ด , แอดเดรสเริ่มต้น และจำนวนของคอยล์ที่จะกำหนดค่า

ตารางที่ 2.14 แสดงข้อมูลตอบกลับเมื่อใช้ฟังก์ชัน Force Multiple Coils (15,0X0F)

Field Name	Example (Hex)	ASCII Characters	RTU 8-Bit Field (Hex)
Header		: (colon)	None
Slave Address	0X05	0 5	0X05
Function	0X0F	0 F	0X0F
Starting Address Hi	0X00	0 0	0X00
Starting Address Lo	0X13	1 3	0X13
Quantity of Coils Hi	0X00	0 0	0X00
Quantity of Coils Lo	0X0B	0 B	0X0B
Error Check		LRC (2 chars.)	CRC (16 bits)
Trailer		CR LF	None
	Total Bytes:	17	8

2.3.8 Preset Multiple Register (16,0X10)

ตั้งค่าไปยังชุดโฮลดิ้งรีจิสเตอร์

การส่งค่า

การใช้งานฟังก์ชัน 16 ต้องกำหนดรีจิสเตอร์เริ่มต้นและข้อมูลที่จะทำการกำหนดค่า จากตัวอย่างเป็นการกำหนดค่าให้กับรีจิสเตอร์ 40020-40022 ไปยังสเลฟหมายเลข 5 ให้มีค่าดังนี้

40020 ข้อมูล 0X0164

40021 ข้อมูล 0X0165

40022 ข้อมูล 0X0166

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.15 แสดงการส่งข้อมูลโดยใช้ฟังก์ชัน Preset Multiple Register (16,0X10)

Field Name	Example (Hex)	ASCII Characters	RTU 8-Bit Field (Hex)
Header		: (colon)	None
Slave Address	0X05	0 5	0X05
Function	0X10	1 0	0X10
Starting Address Hi	0X00	0 0	0X00
Starting Address Lo	0X13	1 3	0X13
No. of Registers Hi	0X00	0 0	0X00
No. of Registers Lo	0X03	0 3	0X03
Byte Count	0X06	0 6	0X06
Data 1 Hi	0X01	0 1	0X01
Data 1 Lo	0X64	6 4	0X64
Data 2 Hi	0X01	0 1	0X01
Data 2 Lo	0X65	6 5	0X65
Data 3 Hi	0X01	0 1	0X01
Data 3 Lo	0X66	6 6	0X66
Error Check		LRC (2 chars.)	CRC (16 bits)
Trailer		CR LF	None
	Total Bytes:	31	15

ผลตอบสนอง

ผลตอบสนองตามปกติจะเป็นการส่งค่าสเลฟแอดเดรส ,ฟังก์ชันโค้ด และจำนวนของรีจิสเตอร์ กลับมา

ตารางที่ 2.16 แสดงข้อมูลตอบกลับเมื่อใช้ฟังก์ชัน Preset Multiple Register (16,0X10)

Field Name	Example (Hex)	ASCII Characters	RTU 8-Bit Field (Hex)
Header		: (colon)	None
Slave Address	0X05	0 5	0X05
Function	0X10	1 0	0X10
Starting Address Hi	0X00	0 0	0X00
Starting Address Lo	0X13	1 3	0X13
No. of Registers Hi	0X00	0 0	0X00
No. of Registers Lo	0X03	0 3	0X03
Error Check		LRC (2 chars.)	CRC (16 bits)
Trailer		CR LF	None
	Total Bytes:	17	8

2.4 มาตรฐานการรับ-ส่งข้อมูลแบบอนุกรม

2.4.1 มาตรฐาน RS-232

เป็นมาตรฐานการเชื่อมต่อข้อมูลแบบอนุกรม ที่มีคนนิยมใช้มากที่สุด กำหนดโดย EIA (Electronics Industry Association) หรือสมาคมผู้ประกอบการอุตสาหกรรมอิเล็กทรอนิกส์ของอเมริกาตั้งแต่ปี 1969 โดยมีจุดเริ่มต้นจากความต้องการที่จะกำหนดมาตรฐานการเชื่อมต่อระหว่างคอมพิวเตอร์กับโมเด็มในสมัยนั้น ตัวมาตรฐานจะกำหนดสิ่งที่เกี่ยวข้องกับการเชื่อมต่อนี้ด้วยกันทั้งหมด 4 หัวข้อหลักๆ ด้วยกันคือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. คุณสมบัติทางไฟฟ้าของสัญญาณ
2. คุณสมบัติทางกลของการเชื่อมต่อ ซึ่งหมายถึงตัวคอนเน็กเตอร์นั่นเอง
3. หน้าที่การทำงานของวงจรสำหรับแลกเปลี่ยนข้อมูล
4. มาตรฐานการเชื่อมต่อสำหรับระบบสื่อสารเฉพาะอย่าง

2.4.2 มาตรฐาน RS-232C

เป็นมาตรฐาน RS-232 ที่มีการปรับปรุงแก้ไขจากมาตรฐานเดิม ซึ่งเราอาจคุ้นเคยกับชื่อนี้มากกว่า RS-232-A หรือ RS-232-B อันที่จริงแล้วยังมีมาตรฐาน RS-232-D ที่ใหม่กว่า RS-232-C โดยที่มีการเพิ่มข้อกำหนดของคอนเน็กเตอร์แบบ DB เข้าไปด้วย เช่น DB-25 ซึ่งในขณะนั้นสิทธิบัตรของตัวคอนเน็กเตอร์แบบนี้ได้หมดอายุลงพอดี จึงสามารถรวมข้อกำหนดเข้าไว้ได้

ลักษณะโดยทั่วไปของการเชื่อมต่อข้อมูลแบบอนุกรม ตามมาตรฐาน RS-232 คือเป็นการสื่อสารข้อมูลแบบจุดต่อจุด ซึ่งเดิมทีเป็นการสื่อสารข้อมูลระหว่างคอมพิวเตอร์กับโมเด็ม ซึ่งจริง ๆ แล้วทั้งสองฝั่งจะเป็นอะไรก็ได้การสื่อสารเป็นแบบสองทางพร้อมกัน (Full-duplex) โดยอาจใช้สายสายสัญญาณอื่นร่วมเพื่อทำแฮนด์เชค (Hand-shake) หรือไม่ก็ได้ มาตรฐาน RS-232 จำกัดความยาวสายไว้ที่ 50 ฟุต (หรือประมาณ 15 เมตร) สำหรับการส่งสัญญาณที่ความเร็ว 19,200 บิทต่อวินาที โดยที่ความยาวสายจะต้องสั้นลงถ้าต้องการสื่อสารที่ความเร็วสูงขึ้นและถ้ามีสัญญาณรบกวนมาก ๆ เช่น ในโรงงาน หรือบริเวณ ใกล้เครื่องจักรที่เป็นแบบมีการสวิทช์สัญญาณไฟฟ้าที่กระแสสูงๆ ก็จะทำให้ต้องมีการลดความเร็วในการส่งสัญญาณลงหรือใช้สายที่สั้นลง

2.4.3 มาตรฐาน RS-422 หรือ RS-422-A

ถูกกำหนดขึ้นโดยสมาคมผู้ประกอบการอุตสาหกรรมอิเล็กทรอนิกส์หรือEIA เช่น เดียวกันกับมาตรฐาน RS-232 โดยมีจุดมุ่งหมายที่จะแก้ปัญหาเรื่องความยาวของสายสื่อสาร โดยใช้การส่งสัญญาณแบบผลต่าง (Differential) แทนที่จะใช้การส่งสัญญาณแบบอ้างอิงกับจุดกราวนด์ เช่น เดียวกันกับ RS-232 การส่งสัญญาณแบบ Differential นี้ ช่วยลดปัญหาสัญญาณรบกวนจาก 2 ปัจจัยด้วยกัน ได้แก่ ปัญหาแรงดันกราวนด์ 2 ฝั่งสาย ไม่เท่ากัน อันเกิดจากกระแสไฟฟ้าที่ไหลในสายกราวนด์ที่ยาวมาก ๆ ก่อให้เกิดความต่างศักดิ์ และปัญหาสัญญาณรบกวนที่เกิดจากแม่เหล็กไฟฟ้าเหนี่ยวนำในสายโดยหากสายไฟที่ใช้ถูกดีเกลียวและวางไว้ใกล้กัน เมื่อมีแรงดันเหนี่ยวนำจะปรากฏแรงดันรบกวนบนสายทั้งสองเท่า ๆ กัน เป็นผลให้ตัวรับที่อ่านความต่างศักดิ์ระหว่างสายอ่านข้อมูลได้เช่นเดิมทั้งสองปัจจัยนี้เอง เป็นสาเหตุที่ทำให้ความต้านทานต่อสัญญาณรบกวนของการสื่อสารแบบ RS-232 ด้อยกว่า RS-422 ตามมาตรฐาน RS-422 นี้ จะใช้สายสัญญาณทั้งหมด 4 เส้น (2 เส้นสำหรับการส่งสัญญาณและอีก 2 เส้นสำหรับรับสัญญาณ)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

และสามารถใช้ความยาวสายสัญญาณได้ ถึง 4,000 ฟุต (หรือ 1.2 กม.) ที่ความเร็ว 100,000 บิทต่อวินาที และการสื่อสารเป็นแบบ 2 ทางพร้อมกัน (Full Duplex)

2.4.4 มาตรฐาน RS-485

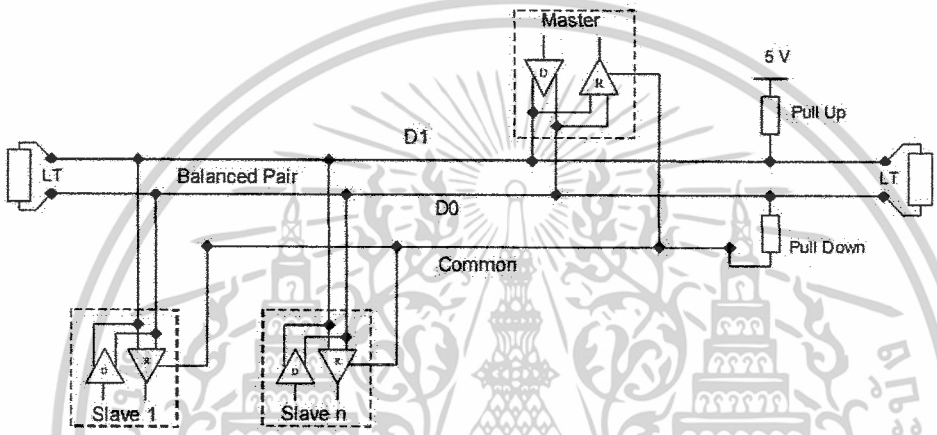
กำหนดโดย สมาคมผู้ประกอบการอุตสาหกรรมอิเล็กทรอนิกส์หรือ EIA เป็นมาตรฐานการเชื่อมต่อสัญญาณแบบอนุกรม (Serial Communication) มีลักษณะการเชื่อมต่อเป็นแบบหลายจุด (Multi-point) หรือ Multi-drop สายสัญญาณที่ใช้มีทั้งแบบที่เป็น 2 สาย และแบบที่เป็น 4 สาย การต่อแบบหลายจุดนี้ทำให้สามารถมองสายสัญญาณเป็นบัสนำสัญญาณได้ (Signal Bus) จำนวนคอมพิวเตอร์หรืออุปกรณ์ที่สามารถอยู่บน RS-485 บัสหนึ่งถูกกำหนดไว้ที่ 32 ตัว ในกรณีที่ต้องการเพิ่มจะต้องมีตัวทวนสัญญาณ (Signal Repeater) หรือใช้ตัวส่ง-รับสัญญาณที่มีอิมพิแดนซ์ (ความต้านทานเสมือน)สูงขึ้น ซึ่งเราอาจเพิ่มจำนวนจุดเชื่อมต่อขึ้นได้ถึง 128 จุด ความยาวของสายสัญญาณตามมาตรฐาน RS-485 นี้ สามารถยาวได้ถึง 1.2 กม เช่นเดียวกับมาตรฐาน RS-422 แต่การสื่อสารจะเป็นแบบสองทางไม่พร้อมกัน (Half Duplex) มีเพียงคอมพิวเตอร์หรืออุปกรณ์ตัวเดียวเท่านั้นที่สามารถส่งสัญญาณออกได้ ณ เวลาหนึ่ง ๆ ส่วนที่เหลือจะเป็นผู้รับสัญญาณ หรือผู้ฟัง

ตารางที่ 2.17 เปรียบเทียบคุณสมบัติของ มาตรฐาน RS-232, RS-422 และ RS-485

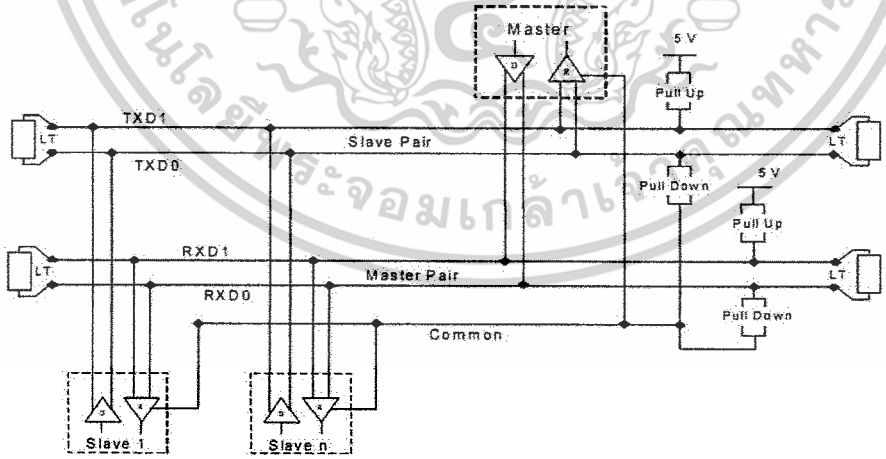
	RS-232	RS-423	RS-422	RS-485
Differential	no	no	yes	yes
Max number of drivers	1	1	1	32
Max number of receivers	1	10	10	32
Modes of operation	half duplex full duplex	half duplex	half duplex	half duplex
Network topology	point-to-point	multidrop	multidrop	multipoint
Max distance (acc. standard)	15 m	1200 m	1200 m	1200 m
Max speed at 12 m	20 kbs	100 kbs	10 Mbs	35 Mbs
Max speed at 1200 m	(1 kbs)	1 kbs	100 kbs	100 kbs
Max slew rate	30 V/µs	adjustable	n/a	n/a
Receiver input resistance	3..7 kΩ	□ 4 kΩ	□ 4 kΩ	□ 12 kΩ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Driver load impedance	3..7 kΩ	□ 450 Ω	100 Ω	54 Ω
Receiver input sensitivity	±3 V	±200 mV	±200 mV	±200 mV
Receiver input range	±15 V	±12 V	±10 V	−7..12 V
Max driver output voltage	±25 V	±6 V	±6 V	−7..12 V
Min driver output voltage (with load)	±5 V	±3.6 V	±2.0 V	±1.5 V



ภาพที่ 2.9 แสดงการต่ออุปกรณ์ด้วยมาตรฐาน RS-485 2 WIRE

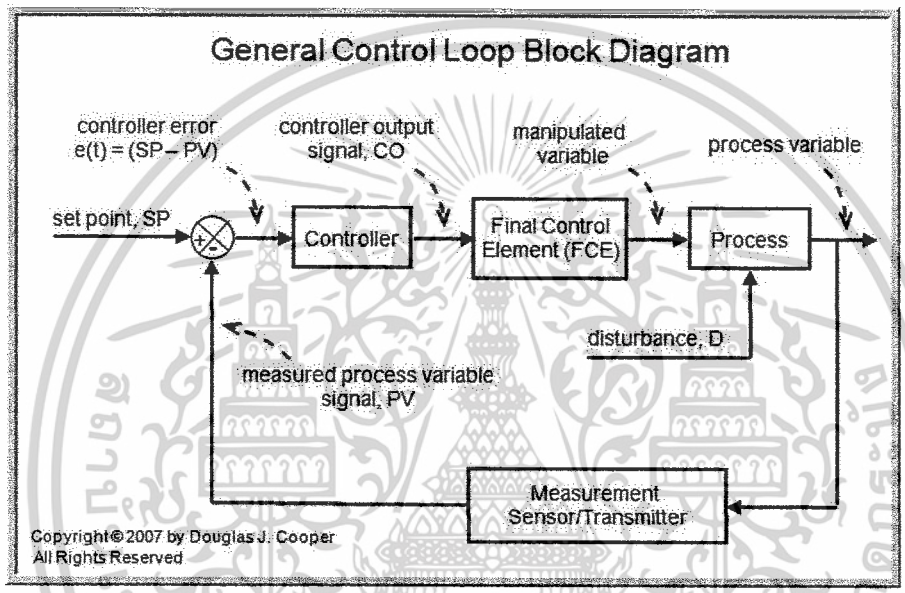


ภาพที่ 2.10 แสดงการต่ออุปกรณ์ด้วยมาตรฐาน RS-485 4 WIRE

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.5 การควบคุมแบบอัตโนมัติ (Automation Control)

การควบคุมแบบดั้งเดิมที่เป็น Manual Control คือ ใช้มนุษย์เป็นผู้ควบคุมเองทั้งหมด เช่น ปิด – เปิดวาล์วเองกดสวิตช์หรือปุ่มจ่ายไฟเองในปัจจุบันได้ถูกแทนที่ด้วยการควบคุมแบบอัตโนมัติแล้วทั้งสิ้นเนื่องจากมีต้นทุนต่ำกว่า และมีเสถียรภาพสูงกว่าการควบคุมดังกล่าวจะเป็นลักษณะของการควบคุมแบบป้อนกลับ (Feedback Control) คือ มีการวัดและส่งค่าที่วัดกลับมา ยังส่วนควบคุมเพื่อคำนวณและจ่ายสัญญาณควบคุมที่เหมาะสมไปยังอุปกรณ์ใช้งานจากที่กล่าวมาการควบคุมแบบอัตโนมัติจะต้องประกอบด้วย 3 ภาคเสมอ คือ



ภาพที่ 2.11 องค์ประกอบของระบบควบคุมแบบป้อนกลับ

1. ภาควัด เช่น เทอร์โมคัปเปิล, หัววัดความชื้น มีหน้าที่วัดค่าที่โปรเซสแล้วส่งมายังภาคควบคุม

2. ภาคควบคุม เช่น เครื่องควบคุมอุณหภูมิ มีหน้าที่รับค่าจากภาควัด นำมาคำนวณแล้วส่งให้ภาคจ่าย

3. ภาคจ่าย เช่น ในระบบทำความร้อนตัวจ่ายคือ ฮิตเตอร์ ในระบบควบคุมอัตราการไหลตัวจ่ายคือ วาล์ว เป็นต้น ซึ่งมีหน้าที่เพิ่มหรือลดค่าที่ควบคุมอยู่บางครั้งเรียกตัวจ่ายว่า final element

กำหนดให้

SP คือ Set point หรือค่าที่ต้องการควบคุม เช่น ต้องการต้มน้ำที่ $80^\circ C$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

PV คือ Process Variable หรือค่าที่วัดมาจาก Process เช่น อุณหภูมิในถังน้ำที่อุณหภูมิปกติเป็น $30^{o}C$

MV คือ Manipulated Variable หรือ สัญญาณควบคุมที่เครื่องควบคุมคำนวณได้มีหน่วยเป็น % (0-100%)

E คือ Error หรือผลต่างระหว่างค่าที่ต้องการควบคุมกับค่าที่วัดได้ (E = SP-PV)

จากรูปจะเห็นว่าเครื่องควบคุมจะได้รับอินพุต 2 ค่า คือ PV (ค่าที่วัดจากโปรเซส) และ SP (ค่าที่ต้องการควบคุมโดยผู้ใช้งานเป็นผู้ป้อน) และ มีเอาต์พุต 1 ค่า คือ MVส่งไปที่โปรเซส เพราะฉะนั้นสิ่งที่ทำให้เครื่องควบคุมแต่ละตัวแตกต่างกันคือส่วนคำนวณของเครื่องควบคุมที่อยู่ภายในซึ่งชี้ถึงรูปแบบการควบคุมที่แตกต่างกัน รูปแบบที่นิยมใช้กันอย่างแพร่หลาย คือการควบคุมแบบ ON-OFF, PID และรูปแบบใหม่ล่าสุด คือ Fuzzy Control

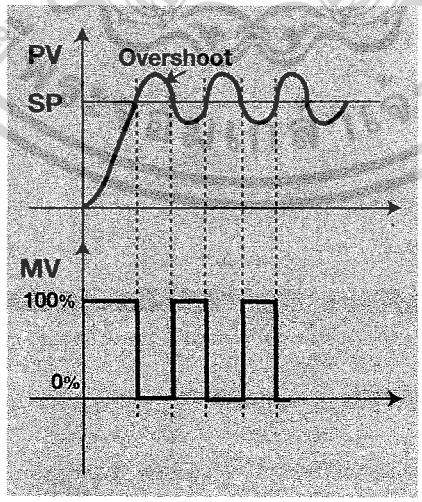
2.5.1 การควบคุมแบบ ON - OFF หรือ Two – Position Control

ในระบบควบคุมแบบ ON-OFF เครื่องควบคุมจะสั่งเอาร์พุตทำงานเพียง 2 สภาวะเท่านั้น คือ ON และ OFF เป็นการควบคุมแบบง่ายๆ และราคาไม่แพง ดังนั้นจึงนิยมใช้กันอย่างกว้างขวางในงานควบคุมทางอุตสาหกรรม ในกรณีที่ผลจากการแกว่งของอุณหภูมิเป็นที่ยอมรับได้

กำหนดให้สัญญาณเอาร์พุตของเครื่องควบคุมเป็น MV และผลต่างระหว่าง SP กับ PV เป็น E (Error) ฉะนั้นในการควบคุมแบบ ON-OFF สัญญาณ MV จะมีค่าเป็น 100% (ON) หรือ 0%(OFF) เท่านั้น โดยจะขึ้นอยู่กับว่า E มีค่าเป็น + หรือเป็น – นั่นคือ

MV = 100 %(ON) เมื่อ E > 0 (PV < SP)

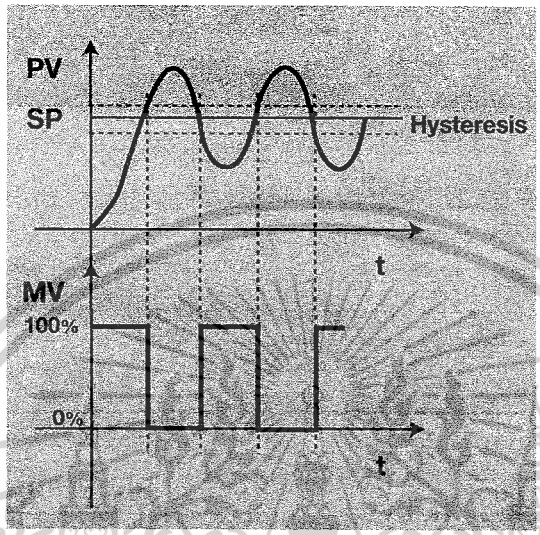
= 0% (OFF) เมื่อ E < 0 (PV > SP)



ภาพที่ 2.12 การควบคุมแบบ ON-OFF

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในกรณีอุณหภูมิที่โปรเซสมีการกระเพื่อมที่ Setpoint จะมีผลทำให้ เอาร์พุตของเครื่องควบคุม ON และ OFF อยู่ตลอดเวลา ซึ่งสามารถแก้ไขได้โดยกำหนด Hysteresis หรือ Differential gap หรือ Dead band เพื่อลดการตัด – ต่อที่เกิดขึ้น แต่ผลเสียคือจะทำให้เกิด Overshoot มากขึ้น



ภาพที่ 2.13 การควบคุมแบบ ON-OFF โดยมี Hysteresis

ลักษณะของ ON – OFF Control คือ จะเกิดการแกว่งของอุณหภูมิ (Oscillation) อยู่ตลอดเวลาโดยในกรณีที่มี Hysteresis ความถี่ในการตัดต่อจะลดลง แต่ค่า Overshoot จะมากขึ้นในระบบที่ช้า ควบในการแกว่งจะยาวกว่าในระบบที่เร็วกว่า

2.5.2 การควบคุมแบบ Proportional (P Control)

ในกระบวนการบางอย่าง การควบคุมแบบ ON-OFF จะไม่ใช่วิธีที่ดีพอ เนื่องจากจะเกิดการแกว่งของอุณหภูมิตลอดเวลา อันเป็นผลมาจากการ ON และ OFF ที่ตัดสินใจจากเครื่องหมายของ Error เท่านั้นนั่นคือ สำหรับ ON – OFF Control ไม่ว่า Error ที่เป็นค่าบวกจะมีค่ามากหรือน้อยเพียงใด เครื่องควบคุมก็จะจ่าย 100% เสมอจนกว่า Error จะกลับเป็นลบจึงจะจ่าย 0% ดังนั้นถ้าต้องการความแน่นอนในการควบคุม สัญญาณควบคุมควรจะเป็นสัดส่วน (Proportional) กับค่า Error ที่รับเข้ามา ซึ่งเป็นลักษณะการควบคุมแบบ Proportional ความสัมพันธ์ระหว่างสัญญาณควบคุม (MV) กับ ค่า Error (E) คือ

$$MV = \frac{100}{PB} E + b$$

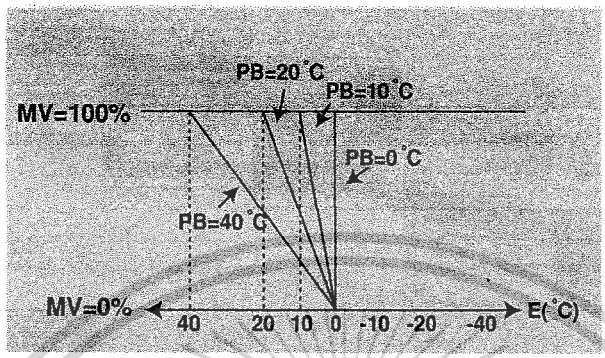
PB = Proportional Band

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

b = bias คือค่าเปอร์เซ็นต์เอาต์พุต เมื่อ E เป็น 0(PV=SP)

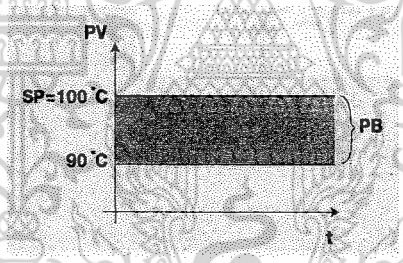
ตัวควบคุมแบบ Proportional มีลักษณะเป็นแอมปลิฟลายที่มีค่า gain คือ kp = $\frac{100}{PB}$

สามารถแสดงความหมายของ PB ได้ดังนี้



ภาพที่ 2.14 แสดงลักษณะของ Proportional Band

เช่น SP = 100 ° C, PB = 10 ° C



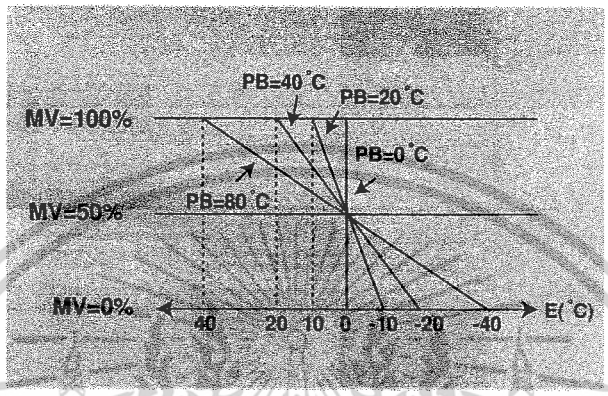
ตารางที่ 2.18 แสดงความสัมพันธ์ของ PV, Error, MV

SP	PV	E	MV
100 °C	50	50	100%
	90	10	100%
	92	8	80%
	94	6	60%
	96	4	40%
	98	2	20%
	100	0	0%
	102	-2	0%
	110	-10	0%

จากตารางสามารถอธิบายได้ว่ากรณี PV เพิ่มขึ้น PB คือย่านอุณหภูมิที่สัญญาณควบคุม MV จะลดลงจาก 100% ไปเป็น 0% นั่นคือเมื่อ PV อยู่ได้ PB MV จะเป็น 100% เสมอ เมื่อ PV อยู่เหนือ B MV จะเป็น 0% เสมอ และ MV จะลดลงอย่างเป็นสัดส่วนเมื่อ PV อยู่ใน PB จะเห็นได้ว่าเมื่อ

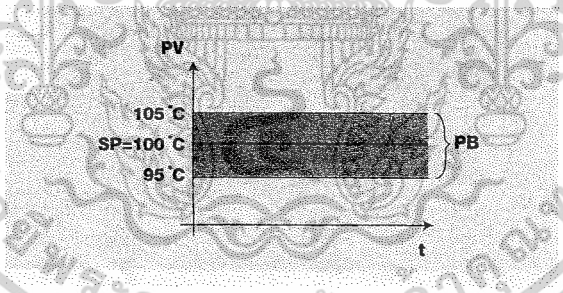
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

PB มีค่าเล็กลง อัตราขยายจะสูงขึ้น และเมื่อ PB เป็น $0^{o}C$ อัตราขยายเป็นอนันต์หรือเสมือนว่าการทำงานของเครื่องควบคุมจะกลายเป็นแบบ ON-OFF นั่นเองส่วนค่า blas เป็นค่าที่ต้องการให้เครื่องควบคุมจ่ายออกไปเมื่อ E = 0 ถ้ากำหนดให้ blas เป็น 0 ในขณะที่ E = 0 MV จะมีค่าเป็น 0% (OFF) ทำให้ไม่สามารถควบคุมให้ PV นิ่งที่ SP ได้ จึงควรตั้งค่า blas ให้เหมาะสมกับการรักษาระดับที่ SP นั้น ๆ โดยทั่วไปเครื่องควบคุม อุณหภูมิจะมี bias ไว้ที่ 50 % และสามารถตั้งค่า Manual Reset ได้



ภาพที่ 2.15 แสดง Proportional Band ที่ bias = 50 %

เช่น SP = 100 ° C, PB = 10 ° C, b = 50%

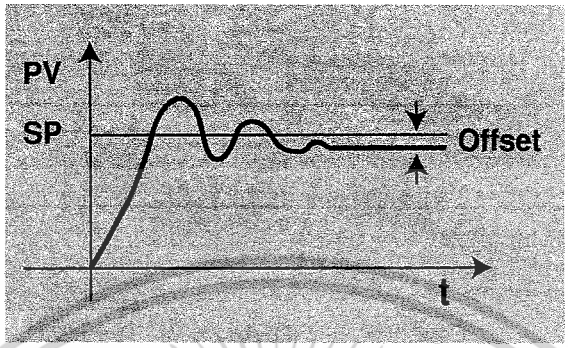


ตารางที่ 2.19 แสดงความสัมพันธ์ของ PV, Error, MV

SP	PV	E	MV
100 °C	50	50	100%
	90	10	100%
	95	5	100%
	98	2	70%
	100	0	50%
	102	-2	30%
	105	-5	0%
	107	-7	0%
	110	-10	0%

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การเกิด Offset คือ ข้อเสียอันเป็นลักษณะของการควบคุมแบบ Proportional ที่ PV จะนิ่งอยู่ที่ค่าค่าหนึ่งซึ่งไม่เท่ากับ SP ซึ่งอาจนิ่งอยู่เหนือ SP หรือได้ SP ก็ได้โดย Offset คือค่าความแตกต่างระหว่าง SP กับ PV ที่เวลาสุดท้าย



ภาพที่ 2.16 แสดง Offset

Offset จะเกิดในระบบที่ใช้ Proportional Control ที่มีค่า PB กว้าง หรือ Load มีการเปลี่ยนแปลง หรือในกรณีที่มีการรบกวน (Disturbance) เข้ามาในโปรเซสการลด PB ให้แคบลงสามารถลด Offset ได้ แต่จะทำให้อุณหภูมิการแกว่งมากขึ้น สามารถแก้ไข Offset ได้ โดยการตั้งค่า Manual Reset ซึ่งเสมือนเป็นการเลื่อน (Shift) กราฟขึ้น (หรือลง) ให้ PV ไปนิ่งที่ SP พอดี

2.5.3 การควบคุมแบบ Proportional + Integral (PI Control)

ลักษณะของ Integral จะมีความสัมพันธ์ระหว่างสัญญาณควบคุม MV กับค่า Error (E) ดังนี้

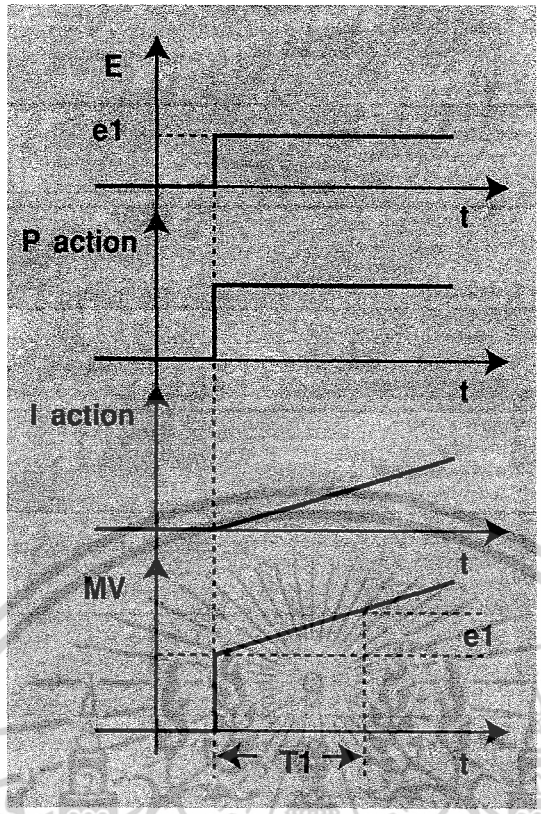
$$MV = \frac{1}{T_I}\int_0^t E \ dt$$

T_I = Integral Time

บางครั้งกิริยาการควบคุมแบบ integral เรียกว่า Reset Control ซึ่งไม่สามารถใช้งานเดี่ยวๆ ได้จะใช้ร่วมกับ Proportional action เสมอ กิริยาการควบคุมแบบ PI สามารถแสดงได้ด้วยสมการต่อไปนี้

$$MV = \frac{100}{PB}E + \frac{1}{T_I}\int_0^t E \ dt$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

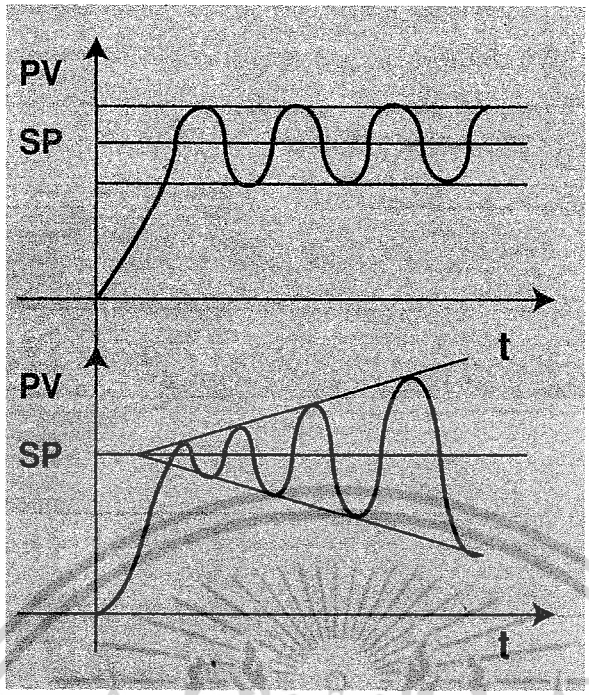


ภาพที่ 2.17 สัญญาณควบคุมของ PI Control

ลักษณะการทำงานของ Integral action คือ สะสมค่า E เมื่อเวลาผ่านไปนั่น คือขณะที่ E มีค่าคงที่สัญญาณควบคุมจะมีค่าเพิ่มขึ้นเรื่อย ๆ ในกรณี E เป็นบวกและจะมีค่าลดลงเรื่อย ๆ ในกรณี E เป็นลบนั่น คือสาเหตุที่ Integral action สามารถแก้ Offset ได้ และทำให้ PV ไปนิ่งที่ SP ได้ท้ายที่สุด

เมื่อลดค่า T_I ลงผลของ Reset action จะมีค่ามากขึ้น ทำให้เกิดการแกว่ง และ Overshoot สูงขึ้น ถ้าลดค่า T_I ต่ำเกินไปจะทำให้ระบบ Oscillate หรือ unstable ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 2.18 ระบบที่ Oscillate และ Unstable เมื่อตั้งค่า T_I ต่ำเกินไป

2.5.4 การควบคุมแบบ Proportional + Derivative (PD Control)

ลักษณะของ Derivative จะมีความสัมพันธ์ระหว่างสัญญาณควบคุม MV กับค่า Error (E) ดังนี้

$$MV = T_D \frac{dE}{dt}$$

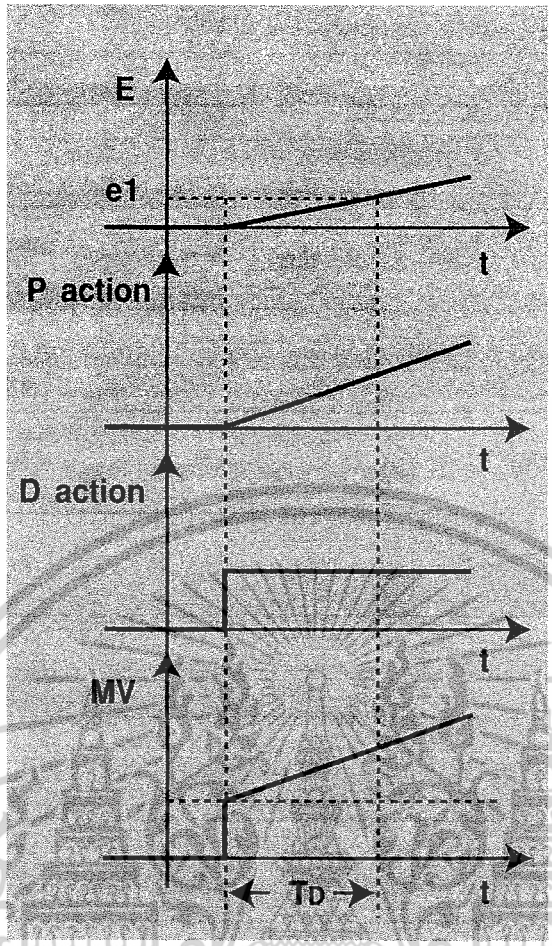
T_D = Derivative Time

ในบางครั้งจะเรียกกิริยาควบคุมแบบ Derivativeว่า Rate control ทั้งนี้ เพราะสัญญาณเอาต์พุตของ เครื่องควบคุม จะเป็นสัดส่วนกับอัตราการเปลี่ยนแปลง ของค่า Error (E) ซึ่งจะใช้ร่วมกับ Proportional action เสมอกิริยาการควบคุมแบบ PD สามารถแสดงได้ด้วยสมการต่อไปนี้

$$MV = \frac{100}{PB}E + T_D \frac{dE}{dt}$$

สัญญาณควบคุมแปรผันตามอัตราการเปลี่ยนแปลงต่อเวลาของค่า Error การควบคุมชนิดนี้ลักษณะเหมือนการคาดการณ์การเปลี่ยนแปลงค่าที่โปรเซสล่วงหน้า ทำให้อุณหภูมิเข้าสู่เสถียรภาพเร็วขึ้น หรือมี Time lag มาก ๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 2.19 สัญญาณควบคุมของ PD Control

เมื่อเพิ่มค่า T_D ขึ้น ผลของ Rate action จะมีค่าเพิ่มขึ้น ถ้าเพิ่มค่า T_D มากเกินไป จะทำให้ระบบ Oscillate ขาดเสถียรภาพ PD Control เหมาะกับ Process ที่มี Time lag มากจะช่วยทำให้การควบคุมถึงจุด ที่ต้องการเร็วขึ้น ไม่เหมาะกับ Process ที่ไวหรือมีการเปลี่ยนแปลงเร็วอยู่แล้ว เช่น ระบบควบคุมการไหล (Flow) ระบบควบคุมความดัน (Pressure) เป็นต้น

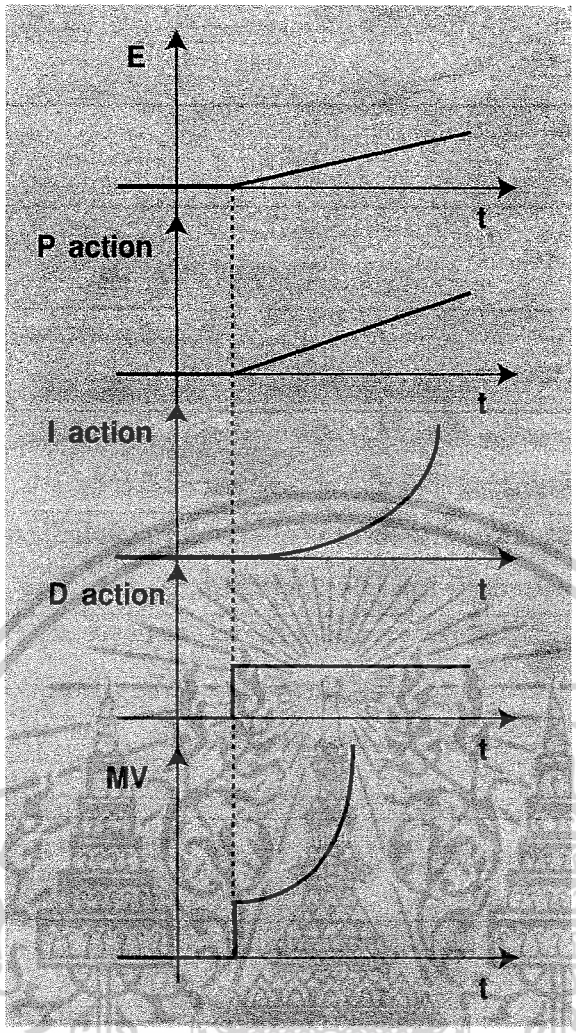
2.5.5 การควบคุมแบบ Proportional + Integral + Derivative (PID)

เป็นการรวมกิริยาควบคุมแบบ Proportional, แบบ Integral และแบบ Derivative เข้าด้วยกัน สามารถแสดงในรูปของสมการได้ดังนี้

$$\mathbf{MV} = \frac{100}{PB}E + \frac{1}{T_I}\int_0^t E\ dt + \mathbf{T}_D\ \frac{dE}{dt}$$

ลักษณะการทำงานของ PID Control สามารถแสดงได้ดังรูป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 2.20 สัญญาณควบคุมของ PID Control

2.5.6 จุดมุ่งหมายของการควบคุม

จุดมุ่งหมายของการควบคุม คือ พยายามรักษาค่า PV ให้มีค่าเท่ากับค่า SP อยู่ตลอดเวลา เมื่อ เกิดการเปลี่ยนแปลงขึ้นในระบบ เช่น Load เปลี่ยน, มีDisturbance หรือมีการเปลี่ยนค่า setpoint ค่า PV จะไม่เท่ากับค่า SP (เกิด Error ขึ้น) อยู่ขณะหนึ่งแล้วเครื่องควบคุมจะต้องสามารถควบคุมให้ค่า PV เข้าสู่ Setpoint ให้ได้โดยลักษณะในการนำค่า PV ไปสู่ค่า SP จะแตกต่างกันตามลักษณะของค่า PID ที่ตั้งไว้ระบบใด ๆ อาจรักษาค่า PV ไว้ที่ Setpoint ได้เหมือนกันแต่ผลตอบสนองของระบบหนึ่งอาจเร็วกว่า และมี Overshoot น้อยกว่า นั่นแสดงว่าระบบนี้ มีการควบคุมที่ดีกว่า ถึงแม้ที่เวลาสุดท้ายของทั้ง 2 ระบบจะมีเสถียรภาพเหมือนกัน ดังนั้นจะเห็นได้ว่าถึงแม้จุดประสงค์หลักของการควบคุมระบบใด ๆ คือ เสถียรภาพหรือการรักษาค่า PV ไว้ที่ Setpoint ให้ได้ ก็ตามแต่ในความเป็นจริงการควบคุมที่ดี จะต้องมองถึงลักษณะของผลตอบสนองด้วยจึงเป็นสาเหตุที่ต้องมีการศึกษา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ลักษณะของโปรเซสนั่นๆ เพื่อให้มีผลตอบสนองที่ดีที่สุด สามารถสรุปลักษณะของการควบคุมที่ดีได้ดังนี้

1. ระบบมีเสถียรภาพ (Stable) สามารถรักษาค่า PV ไว้ที่ Setpoint ได้
2. ระบบมีการตอบสนองเร็ว
3. ระบบไม่แกว่ง มีค่า Overshoot น้อย
4. ไม่เกิด Offset ที่สภาวะสุดท้าย
5. ระบบสามารถรองรับการเกิด Disturbance หรือการเปลี่ยนค่า Setpoint ได้

การปรับค่า PID ที่โปรเซสหนึ่งแล้วมีผลตอบสนองที่ดี เมื่อนำค่า PID นี้ไปใช้กับอีกโปรเซสหนึ่ง อาจได้ผลตอบสนองที่ไม่ดีก็ได้ ผลของการปรับค่า PID มีดังนี้

2.5.7 สรุปผลของวิธีการควบคุมแบบต่างๆ

ผลของ P action

สามารถเพิ่มผลของ P action ได้โดยลดค่า PB ลง จะมีผลทำให้

- มีค่า Offset น้อยลง
- มี Overshoot สูงขึ้น เกิดการแกว่งมากขึ้น
- ระบบขาดเสถียรภาพมากขึ้น ถ้าลดค่า PB มากเกินไป จะทำให้ระบบ Oscillate

ผลของ I action

สามารถเพิ่มผลของ I action ได้โดยลดค่า T_I ลงจะมีผลทำให้

- ไม่มี Offset
- มี Overshoot สูงขึ้น เกิดการแกว่งมากขึ้น
- ระบบขาดเสถียรภาพมากขึ้น ถ้าลดค่า T_I มากเกินไป จะทำให้ระบบ Oscillate หรือ Unstable

ผลของ D action

สามารถเพิ่มผลของ D action ได้โดยลดค่า T_D ลงจะมีผลทำให้

- มี Overshoot ลดลงมีคาบการแกว่งสั้นลง
- ระบบมีเสถียรภาพมากขึ้น ไวขึ้น
- ในระบบที่เร็วอยู่แล้วจะขาดเสถียรภาพ

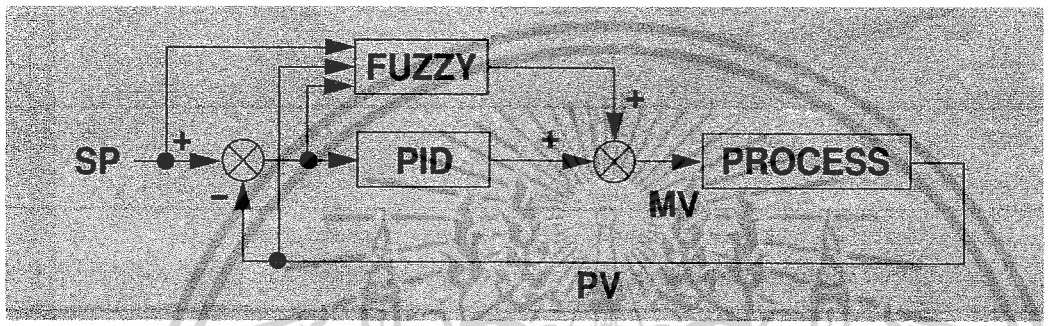
2.5.8 การควบคุมแบบ Fuzzy

Fuzzy เป็นทฤษฎีหนึ่งทางคณิตศาสตร์ คือ Fuzzy Sets ซึ่งเป็นลักษณะของตัวเลขที่สามารถบ่งชี้ได้ถึงคุณภาพ (Quality) ไม่ใช่เพียงปริมาณ (Quantity) เพียงอย่างเดียวเหมือนตัวเลขทั่วไปที่เป็น Crisp sets ลักษณะของ Fuzzy จะมีความใกล้เคียงภาษามนุษย์มากว่าที่เป็นเพียงตัวเลข

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

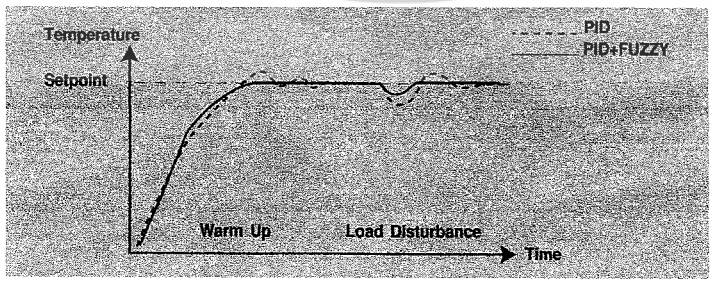
ตายตัวตัวหนึ่ง เช่น หากพวกเราจะสอนคนหนึ่งขับรถให้จอดตรงไฟแดงพอดี เรามักบอกว่า ".ให้ค่อยๆ เหยียบเบรก" มากว่าจะบอกว่า " ให้เหยียบเบรกด้วยอัตราหน่วง -3 m/s^2 " คำว่า Fuzzy แปลว่าคลุมเครือ, ไม่แน่ชัด คือเป็นคณิตศาสตร์ที่ไม่ได้ ระบุเพียงแค่ใช่หรือไม่ใช่แต่สามารถระบุความใกล้เคียงที่จะใช่หรือไม่ใช่ได้นั่น คือ ช่วงตรงกลางระหว่าง ใช่กับไม่ใช่ ซึ่งเป็นลักษณะของภาษามนุษย์(Language) ที่มักใช้คำว่า ค่อย ๆ , ค่อนข้าง,เล็กน้อย,เกือบ เป็นต้น

2.5.9 การควบคุมแบบ Fuzzy+ PID



ภาพที่ 2.21 แสดงไดอะแกรมของระบบควบคุมแบบ Fuzzy+PID

Fuzzy Control เอาชนะข้อด้วยของ PID ได้ และสามารถควบคุมระบบได้ อย่างมีประสิทธิภาพด้วยกรณีต่างๆ ที่ได้กำหนดขึ้นเพื่อนรองรับการเปลี่ยนแปลงในรูปแบบที่แตกต่างกันไป เหมือนกับการบันทึกประสบการณ์ที่เคยผ่านมาแล้ว Fuzzy Control จะปรับแต่ง PID ทางอ้อมเพื่อให้สัญญาณควบคุม MV สามารถแก้ไขปัญหาได้อย่างยืดหยุ่นและรวดเร็วตาม Process ที่เปลี่ยนแปลงไปด้วยวิธีนี้ จะทำให้ Process เข้าสู่ Setpoint ด้วยเวลาที่สั้นที่สุด และมี Overshoot ต่ำสุดภายใต้การรบกวนของ Disturbance ภายนอกมีเสถียรภาพมากว่าแบบ PID แบบดั้งเดิมอย่างเห็นได้ชัด PID เป็น Digital Information ต่างจาก Fuzzy ที่เป็น Language Information

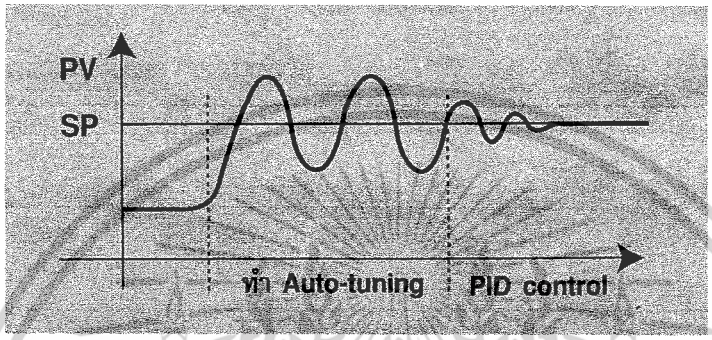


ภาพที่2.22 แสดงผลตอบสนองของระบบควบคุมแบบ Fuzzy+PID

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.5.10 AUTO – TUNING

เป็นฟังก์ชันที่อยู่ในเครื่องควบคุมชนิดดิจิตอล ใช้สำหรับคำนวณหาค่า PID ที่เหมาะสมกับโปรเซสโดยอัตโนมัติ เมื่อเริ่มการทำงาน Auto – Tuning เครื่องควบคุมจะทำงานเป็น ON – OFF ระยะหนึ่งจะทำให้ระบบเกิดการแกว่งเพื่อศึกษาคุณลักษณะของโปรเซสว่าช้าหรือเร็ว มี Overshoot สูงเท่าใด เครื่องควบคุมจะควบคุมในลักษณะ ON – OFF อยู่ประมาณ 3-5 Cycle แล้วจึงทำการคำนวณหาค่า PID ออกมาเริ่มควบคุมระบบด้วย PID Control



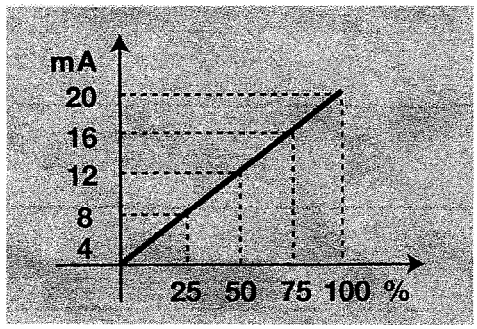
ภาพที่ 2.23 แสดงการทำ Auto-Tunning

โดยทั่วไปตัวเครื่องควบคุมจะมีไฟแสดงผลขณะที่ทำ Auto – Tuning อยู่ และเมื่อทำ Auto – Tuning เสร็จไฟแสดงผลจะดับแสดงว่าเข้าสู่การทำงานแบบ PID แล้วฟังก์ชัน Auto – Tuning ช่วยอำนวยความสะดวกแก่ผู้ใช้ โดยคำนวณหาค่า PID ออกมาโดยอัตโนมัติให้ผู้ใช้ไม่จำเป็นต้องคำนวณด้วยตัวเองแต่อย่างไรก็ตาม ในบางโปรเซสที่ควบคุมได้ยากค่า PID ที่คำนวณได้จากการทำ Auto – Tuning ก็ไม่สามารถควบคุมให้ระบบเสถียรภาพที่ดีพอได้จำเป็นที่ผู้ใช้จะต้องทำการปรับปรับแต่งค่า PID เพิ่มเติมให้ได้ผลตอบสนองที่ดีที่สุด

2.5.11 เอาต์พุตของเครื่องควบคุม

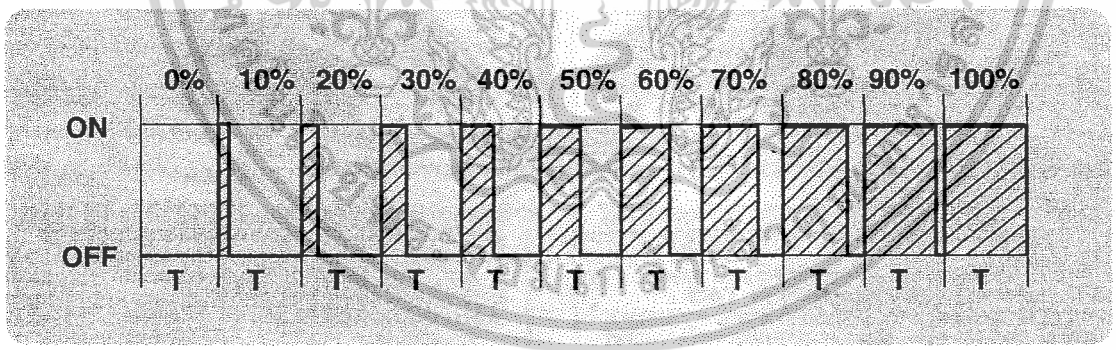
ค่าสัญญาณควบคุม (MV) จากเครื่องควบคุม เป็นค่าที่เกิดจากการคำนวณตามขนาดของค่า Error ในโปรเซสโดยสัญญาณควบคุมนี้จะถูกส่งไปที่ตัวจ่าย (Final element) เพื่อทำการแก้ไข Error ที่ได้รับเข้ามาโดยสัญญาณนี้มีลักษณะเป็นแอนะลอก คือ 1-100% สำหรับเครื่องควบคุมที่มีเอาต์พุตเป็นสัญญาณมาตรฐาน เช่น 4-20 *mA* สามารถเทียบเอาต์พุตในลักษณะของเปอร์เซ็นต์กับสัญญาณมาตรฐานได้ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 2.24 แสดงการเปรียบเทียบเอาต์พุต 4-20 mA กับ 0 -100%

แต่สำหรับเครื่องควบคุมที่มีเอาต์พุตเป็นรีเลย์ หรือพัลส์ซับโซลิดสเตทรีเลย์ซึ่งเป็นลักษณะของสวิตช์ ON-OFF ที่มีเพียง 2 สภาวะ คือ ON (100 %) กับ OFF (0 %) จะไม่สามารถส่งสัญญาณเอาต์พุตในลักษณะแอนะลอกได้โดยตรง แต่จะใช้หลักการของการเปลี่ยนแปลงช่วงเวลา ON (Duty Cycle) บนคาบคงที่แทน โดยในเครื่องควบคุมแบบ PID ที่มีเอาต์พุตเป็นรีเลย์ หรือพัลส์ซับโซลิดสเตทรีเลย์จะต้องมีคารติดตั้งค่าพารามิเตอร์ตัวหนึ่ง คือค่า Proportioning time หรือ Cycle time ซึ่งเป็นค่าคาบคงที่ที่จะมีช่วงเวลา ON (Duty Cycle) ตามค่าสัญญาณควบคุม นั่นแสดงว่าเมื่อสัญญาณควบคุมเป็น 100 % เอาต์พุตจะ ON ตลอดคาบ และเมื่อสัญญาณควบคุมเป็น 0 % เอาต์พุตจะ OFF ตลอดคาบ ส่วนค่าสัญญาณควบคุมที่อยู่ระหว่าง 0-100 % Duty Cycle ที่แตกต่างกันไป ดังรูป



ภาพที่ 2.25 แสดงการเปรียบเทียบเอาต์พุตรีเลย์ กับ 0 -100%

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.20 แสดงการเปรียบเทียบเอาต์พุต 4-20 mA,รีเลย์ กับ 0 -100%

เปอร์เซ็นต์ เอาต์พุต	เอาต์พุต 4-20 mA	เอาต์พุตรีเลย์/พัลส์ขับโซลิดสเตตรีเลย์ ช่วงเวลา ON	ช่วงเวลา OFF
0%	4 mA	0 Sec.	10 Sec.
12.5%	6 mA	1.25 Sec.	8.75 Sec.
25%	8 mA	2.5 Sec.	1.5 Sec.
37.5%	10 mA	3.75 Sec.	6.25 Sec.
50%	12 mA	5 Sec.	5 Sec.
62.5%	14 mA	6.25 Sec.	3.75 Sec.
75%	16 mA	7.5 Sec.	2.5 Sec.
87.5%	18 mA	8.75 Sec.	1.25 Sec.
100%	20 mA	10 Sec.	0 Sec.

2.6 Programmable Logic Controller (PLC)

โปรแกรมแมเบิลลอจิกคอลโทรลเลอร์ (Programmable logic Control : PLC) เป็นอุปกรณ์ควบคุมการทำงานของเครื่องจักรหรือกระบวนการทำงานต่าง ๆ โดยภายในมี Microprocessor เป็นมันสมองสั่งการที่สำคัญ PLC จะมีส่วนที่เป็นอินพุตและเอาต์พุตที่สามารถต่อออกไปใช้งานได้ทันที ตัวตรวจวัดหรือสวิทช์ต่าง ๆ จะต่อเข้ากับอินพุตส่วนเอาต์พุตจะใช้ต่อออกไปควบคุมการทำงานของอุปกรณ์หรือเครื่องจักรที่เป็นเป้าหมาย เราสามารถสร้างวงจรหรือแบบของการควบคุมได้ โดยการป้อนเป็นโปรแกรมคำสั่งเข้าไปใน PLC นอกจากนี้ยังสามารถใช้งานร่วมกับอุปกรณ์อื่น เช่นเครื่องอ่านบาร์โค๊ด (Barcode Reader) เครื่องพิมพ์ (Printer) ซึ่งในปัจจุบันนอกจากเครื่อง PLC จะใช้งานแบบเดี่ยว (Stand alone) แล้วยังสามารถต่อ PLC หลาย ๆ ตัวเข้าด้วยกัน (Network) เพื่อควบคุมการทำงานของระบบให้มีประสิทธิภาพมากยิ่งขึ้นด้วย จะเห็นได้ว่าการใช้งาน PLC มีความยืดหยุ่นมากดังนั้นในโรงงานอุตสาหกรรมต่างๆ จึงเปลี่ยนมาใช้ PLC มากขึ้น

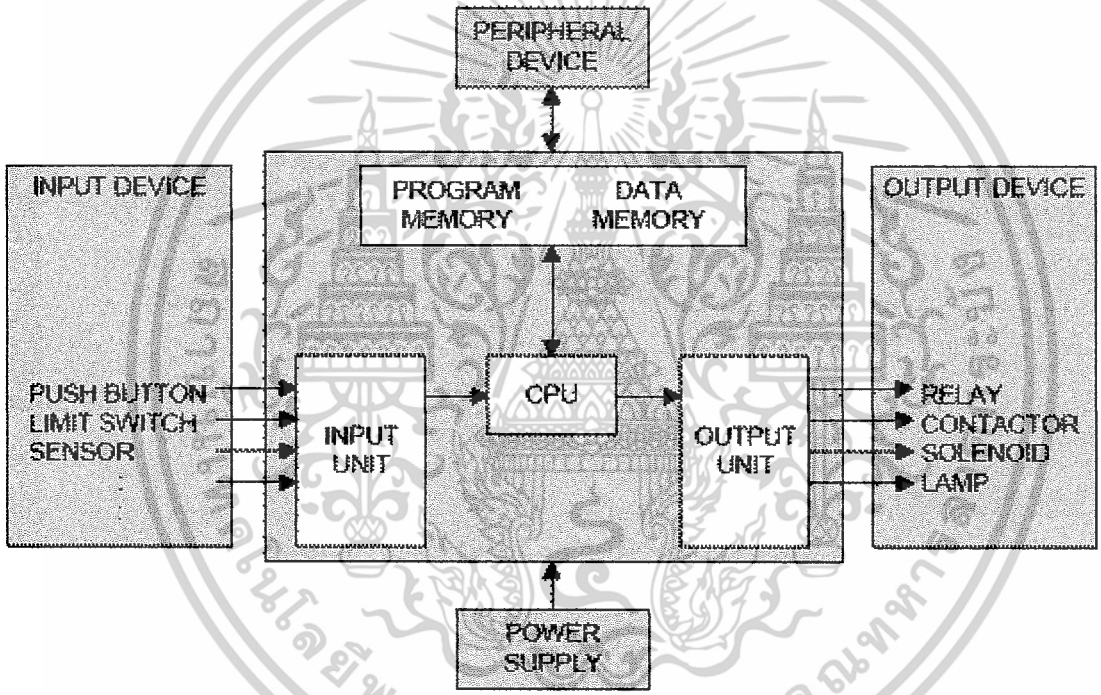
PLC เป็นอุปกรณ์ชนิดโซลิด – สเตท (Solid State) ที่ทำงานแบบลอจิก (Logic Functions) การออกแบบการทำงานของ PLC จะคล้ายกับหลักการทำงานของคอมพิวเตอร์ จากหลักการพื้น ฐานแล้ว PLC จะประกอบด้วยอุปกรณ์ที่เรียกว่า Solid-State Digital Logic Elements เพื่อให้ทำงานและตัดสินใจแบบลอจิก PLC จะใช้สำหรับควบคุมกระบวนการทำงานของเครื่องจักร และอุปกรณ์ในโรงงานอุตสาหกรรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การใช้ PLC สำหรับควบคุมเครื่องจักรหรืออุปกรณ์ต่าง ๆ ในโรงงานอุตสาหกรรมจะมีข้อได้เปรียบกว่าการใช้ระบบของรีเลย์ (Relay) ซึ่งจำเป็นจะต้องเดินสายไฟฟ้า หรือที่เรียกว่า Hard-Wired ฉะนั้นเมื่อมีความจำเป็นที่ต้องเปลี่ยนกระบวนการผลิตหรือลำดับการทำงานใหม่ ก็ต้องเดินสายไฟฟ้าใหม่ซึ่งเสียเวลาและเสียค่าใช้จ่ายสูง แต่เมื่อเปลี่ยนมาใช้ PLC แล้ว การเปลี่ยนกระบวนการผลิตหรือลำดับการทำงานใหม่นั้น ทำได้โดยการเปลี่ยนโปรแกรมใหม่เท่านั้นนอกจากนี้แล้ว PLC ยังใช้ระบบโซลิด – สเตท ซึ่งน่าเชื่อถือกว่าระบบเดิมการกินกระแสไฟฟ้าน้อยกว่าและสะดวกกว่าเมื่อต้องการขยายขั้นตอนการทำงานของเครื่องจักร

2.6.1 โครงสร้างของ PLC

PLC เป็นอุปกรณ์สำหรับใช้ในงานอุตสาหกรรม PLC ประกอบด้วย



ภาพที่ 2.26 ไดอะแกรมแสดงโครงสร้างของ PLC

2.6.1.1 ตัวประมวลผล (CPU)

ทำหน้าที่คำนวณและควบคุมซึ้งเปรียบเสมือนสมองของ PLC ภายในระกอบด้วยวงจรลอจิกหลายชนิด และมีไมโครโปรเซสเซอร์เบส (Micro Processor Based) ใช้แทนอุปกรณ์จำพวกรีเลย์ เคาน์เตอร์/ไทม์เมอร์ และซีเคว้นเซอร์ เพื่อให้ผู้ใช้สามารถออกแบบวงจรโดยใช้ Relay Ladder Diagram ได้ CPU จะยอมรับข้อมูลจากอุปกรณ์อินพุทต่าง ๆ จากนั้นจะทำการประมวลผล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

และเก็บข้อมูล โดยใช้โปรแกรมจากหน่วยความจำ หลังจากนั้นจะส่งส่งข้อมูลที่เหมาะสมและถูกต้องออกไปยังอุปกรณ์เอาท์พุท

2.6.1.2 หน่วยความจำ(Memory Unit)

ทำหน้าที่เก็บรักษาโปรแกรมและข้อมูลที่ใช้ในการทำงาน โดยขนาดของหน่วยความจำจะถูกแบ่งออกเป็นบิตข้อมูล (Data Bit) ภายในหน่วยความจำ 1 บิต ก็จะมีค่าสภาวะทางลอจิก 0 หรือ 1 แตกต่างกันแล้วแต่คำสั่งซึ่ง PLC ประกอบด้วยหน่วยความจำสองชนิด คือ ROM และRAM

RAM ทำหน้าที่เก็บโปรแกรมของผู้ใช้ และข้อมูลที่ใช้ในการปฏิบัติงานของ PLC หน่วยความจำประเภทนี้จะมีแบตเตอรี่เล็ก ๆ ต่อไว้ เพื่อใช้เป็นไฟเลี้ยงข้อมูลเมื่อเกิดไฟดับการอ่านและการเขียนข้อมูลลงใน RAM ทำได้ง่ายมาก เพราะฉะนั้นจึงเหมากับงานในระยะทดลองเครื่องที่มีการเปลี่ยนแปลงแก้ไขโปรแกรมอยู่บ่อยๆ

ROM ทำหน้าที่เก็บโปรแกรมสำหรับใช้ในการปฏิบัติงานของ PLC ตามโปรแกรมของผู้ใช้ หน่วยความจำแบบ ROM ยังสามารถแบ่งได้เป็น EPROM ซึ่งจะต้องใช้อุปกรณ์พิเศษในการเขียนและลบโปรแกรม เหมาะกับงานที่ไม่ต้องการเปลี่ยนแปลงโปรแกรม นอกจากนี้ยังมีแบบ EEPROM หน่วยความจำประเภทนี้ไม่ต้องใช้เครื่องมือพิเศษในการเขียนและลบโปรแกรม สามารถใช้งานได้เหมือนกับ RAM แต่ไม่ต้องใช้แบตเตอรี่สำรองแต่ราคาจะแพงกว่าเนื่องจากรวมคุณสมบัติของ ROM และ RAM ไว้ด้วยกัน

2.6.1.3 หน่วยอินพุต-เอาต์พุต (Input-Output Unit)

หน่วยอินพุต ทำหน้าที่รับสัญญาณจากอุปกรณ์ภายนอก แล้วแปลงสัญญาณให้เป็นสัญญาณที่เหมาะสมแล้วส่งให้หน่วยประมวลผลต่อไป

หน่วยเอาต์พุต หน้าที่รับข้อมูลจากตัวประมวลผล แล้วส่งต่อข้อมูลไปควบคุมอุปกรณ์ภายนอกเช่น ควบคุมหลอดไฟ มอเตอร์ และวาล์ว เป็นต้น

2.6.1.4 แหล่งจ่ายไฟ (Power Supply) ทำหน้าที่จ่ายพลังงาน และรักษาระดับแรงดันไฟฟ้ากระแสตรงให้กับ CPU Unit หน่วยความจำและหน่วยอินพุท/ เอาท์พุท

2.6.1.5 อุปกรณ์ต่อร่วม (Peripheral Devices)

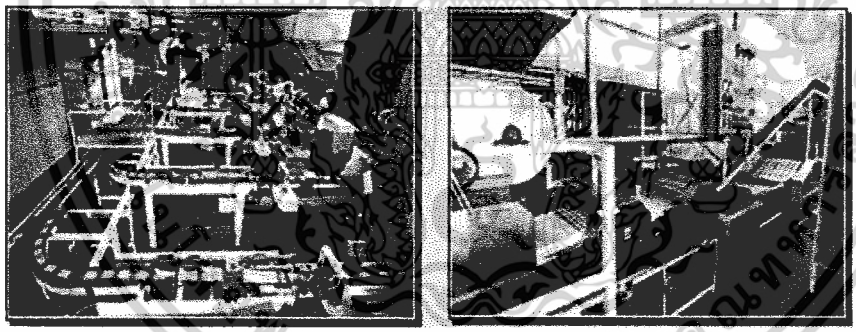
- PROGRAMMING CONSOLE
- EPROM WRITER
- PRINTER

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- GRAPHIC PROGRAMMING
- CRT MONITOR
- HANDHELD
- etc

2.6.2 ตัวอย่างการใช้ PLC ในอุตสาหกรรมต่างๆ

ภาพที่ 2.27 ภาพเครื่องผสมวัตถุดิบที่ใช้ PLC ในการควบคุม

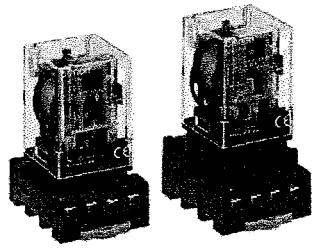


ภาพที่ 2.28 ภาพการขนถ่ายผลิตภัณฑ์ที่ใช้ PLC ในการควบคุม

2.7 รีเลย์ (Relay)

รีเลย์ คือ อุปกรณ์อิเล็กทรอนิกส์ ที่ทำหน้าที่ตัด-ต่อวงจรคล้ายกับสวิตซ์ โดยใช้หลักการหน้าสัมผัส และการที่จะให้มันทำงานก็ต้องจ่ายไฟให้มันตามที่กำหนด เพราะเมื่อจ่ายไฟให้กับตัวรีเลย์ มันจะทำให้หน้าสัมผัสติดกัน กลายเป็นวงจรปิดและตรงข้ามทันทีที่ไม่ได้จ่ายไฟให้มันมันก็จะกลายเป็นวงจรเปิด รีเลย์เป็นสวิตช์ควบคุมที่ทำงานด้วยไฟฟ้า แบ่งออกตามลักษณะการใช้งานได้เป็น 2 ประเภทคือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 2.29 รีเลย์ที่ใช้ในงานอุตสาหกรรม

2.7.1 รีเลย์ควบคุม (Control Relay) มีขนาดเล็กกำลังไฟฟ้าต่ำใช้ในวงจรควบคุมทั่วไป ที่มีกำลังไฟฟ้าไม่มากนักหรือเพื่อการควบคุมรีเลย์หรือคอนแทกเตอร์ขนาดใหญ่ รีเลย์ควบคุมบางทีเรียกกันง่าย ๆ ว่า "รีเลย์"

2.7.2 รีเลย์กำลัง (Power Relay) หรือมักเรียกกันว่าคอนแทกเตอร์ (Contactor or Magnetic Contactor) ใช้ในการควบคุมไฟฟ้ากำลัง มีขนาดใหญ่กว่ารีเลย์ธรรมดา

2.8 RTD (Resistance Temperature Detector)

RTD เป็นอุปกรณ์ตรวจจับอุณหภูมิ โดยอาศัยหลักการเปลี่ยนแปลงความต้านทานของโลหะบริสุทธิ์ เมื่ออุณหภูมิเปลี่ยนมีทั้งแบบเส้นลวดและแบบ Thin film ในที่นี้จะกล่าวถึงแบบเส้นลวดเท่านั้น โดยความสัมพันธ์ของอุณหภูมิและความต้านทานของ RTD เป็น ดังนี้

$$R_T = R_0[1 + \alpha(\Delta T) + \beta(\Delta T)^2 + \lambda(\Delta T)^3]$$

โดย R_0 = ความต้านทาน ณ อุณหภูมิอ้างอิง (โดยปกติเป็น $0\,^{\circ}C$), Ω

R_T = ความต้านทาน ณ อุณหภูมิที่สังเกต, Ω

α = ค่าสัมประสิทธิ์ ของการเปลี่ยนแปลงค่าความต้านทานต่ออุณหภูมิ $1\,^{\circ}C$ ($\Omega/\Omega/^{\circ}C$)

β และ λ = ค่าสัมประสิทธิ์ที่ใช้คำนวณค่าความต้านทาน / อุณหภูมิได้ละเอียดขึ้น ซึ่งค่า β, λ, α นั้นจะขึ้นอยู่กับ การกำหนดของผู้ผลิตโดยทั่วไปการคำนวณจะใช้สมการ

$$R_T = R_0(1 + \alpha(\Delta T))$$

อาร์ทีดีแบบที่ใช้กันมากที่สุด คือ Platinum 100 โอห์ม (Pt100) คือ ที่ $0^{o}C$ จะมีค่า 100 โอห์ม และจะเปลี่ยนค่าความต้านทานโดยเฉลี่ย 0.385 โอห์มต่อ $1^{o}C$ มีย่านอุณหภูมิใช้งานในช่วง -250 ถึง $600\,^{o}C$ ในการใช้งานปกติจะมีแหล่งจ่ายกระแสคงที่ให้อาร์ทีดีอยู่ สมมติเป็น $1\ mA$ นั่นคือ เมื่ออุณหภูมิเปลี่ยนไป $1\,^{o}C$ จะทำให้เกิดการเปลี่ยนแปลงแรงดัน $0.385\ mA$ ซึ่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

มากกว่าเทอร์โมคัปเปิล *Type k* ถึง 10 เท่า ทำให้มีผลกระทบจากสัญญาณรบกวนน้อยกว่าเทอร์โมคัปเปิลที่สภาวะเดียวกัน

เนื่องจากตัวอาร์ทีดีเป็นเพียงค่าความต้านทานจึงต้องมีวงจรจ่ายกระแสให้ เพื่อให้เกิดเป็นแรงดันที่เปลี่ยนไปแล้ว จึงนำแรงดันนี้ไปใช้งาน แต่กระแสจำนวนนี้ก็สร้างความร้อนขึ้นในตัวอาร์ทีดีด้วย ทำให้ค่าความต้านทานสูงขึ้น จึงจำเป็นต้องจำกัดกระแสเลี้ยงอาร์ทีดีนี้มีค่าสูงเกินไป

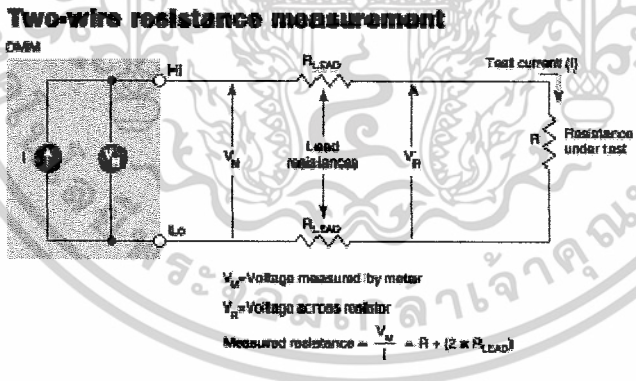
2.8.1 ส่วนประกอบของอาร์ทีดี

อาร์ทีดี ทำจากลวดโลหะที่มีความยาวค่าหนึ่ง ซึ่งที่ $0^{o}C$ จะมีค่าหนึ่งตามที่กำหนดลวดโลหะนี้ จะพันอยู่บนแกนที่เป็นฉนวนไฟฟ้ามีคุณสมบัติทนต่อความร้อน และต้องมีสัมประสิทธิ์การขยายตัวสัมพันธ์กับการขยายตัวของขดลวดอาร์ทีดีจะบรรจุอยู่ใน Metal Sheath ฉนวนที่ใช้เป็นพวกแมกนีเซียมออกไซด์ หรืออะลูมิเนียมออกไซด์

2.8.2 ชนิดของอาร์ทีดี

2.8.2.1 อาร์ทีดี 2 สาย

วงจรใช้งานพื้นฐานของอาร์ทีดี คือ Wheathstone Bridge โดย Rt คือ อาร์ทีดี ซึ่งติดตั้งอยู่ในจุดที่ต้องการวัดอุณหภูมิ มีค่าความต้านทานอีก 3 ค่าในวงจร คือ R1, R2 และ R3 ซึ่งต้องอยู่ที่อุณหภูมิห้อง และเป็นความต่านทานชนิดที่มีความถูกต้องสูง



ภาพที่ 2.30 การต่อ RTD 2 สาย

วงจร Bridge จะอยู่ในสภาวะสมดุล เมื่ออาร์ทีดี (Rt) อยู่ที่ $0^{o}C$ แล้วทำให้Rt/R3 = R1/R2 ซึ่งจะไม่มีกระแสไหลผ่านกัลป์วานอมิเตอร์เมื่ออุณหภูมิที่ Rt สูงขึ้น ค่า Rt จะเพิ่มขึ้นทำให้วงจร Bridge ไม่สมดุลและมีกระแสไหลผ่านกัลป์วานอมิเตอร์ อย่างไรก็ตาม อาร์ทีดี 2 สายเหมาะกับงานที่อาร์ทีดีอยู่ใกล้กับวงจรเท่านั้นไม่เหมาะกับงานที่ต้องลากสายยาว ๆ เนื่องจากจะมีความผิดพลาด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เกิดขึ้นจากค่าความต้านทานสะสมในสายตัวนำ ทำให้ค่าที่อ่านได้ผิดเพี้ยนไปอาร์ทีดี 2 สาย จึงเหมาะสำหรับงานที่ไม่ต้องการความถูกต้องสูงนัก

2.8.2.2 อาร์ทีดี 3 สาย

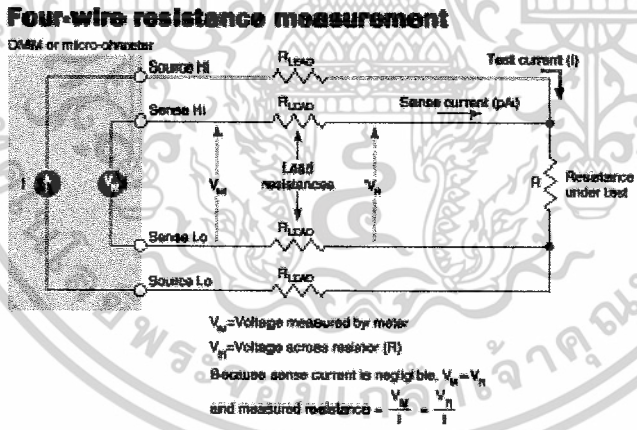
อาร์ทีดี 3 สาย เป็นแบบที่นิยมใช้มากที่สุดในอุตสาหกรรม โดยสายทั้ง 3 ที่อยู่ระหว่างจุดวัดกับวงจร จะต้องมีขนาด ความยาวที่เท่ากันและอุณหภูมิเดียวกันตลอด เพื่อให้ค่าความต้านทาน r1 , r2 และ r3 เปลี่ยนไปในทิศทาง เดียวกันด้วยขนาดที่เท่ากัน นั่นคือ

$$\frac{Rt + r3}{R3} = \frac{R1 + r1}{R2}$$

เนื่องจาก r1 เท่ากับ r3 เพราะฉะนั้นอุณหภูมิที่วัดจึงขึ้นอยู่กับ Rt เพียงตัวเดียวทำให้อาร์ทีดี 3 สาย มีความถูกต้องสูงกว่าอาร์ทีดีแบบ 2 สาย

2.8.2.3 อาร์ทีดี 4 สาย

อาร์ทีดี 4 สาย เป็นแบบที่มีความถูกต้องที่สุด เนื่องจากเลื่อนจุดต่อของ Bridge ออกไปอยู่ภายนอกทั้ง 4 จุด สามารถชดเชยความต้านทานของสายตัวนำได้ทั้งหมด โดยสายทั้ง 4 ต้องมีขนาด ความยาวเท่ากันและอยู่ในอุณหภูมิเดียวกันตลอดเหมือนกับอาร์ทีดี 3 สาย



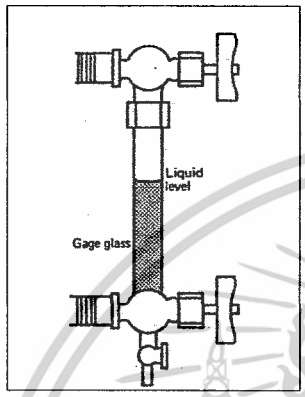
ภาพที่ 2.31 การต่อ RTD 4 สาย

อาร์ทีดี เป็น Sensor วัดอุณหภูมิที่มี Linearity ดีที่สุด มีความถูกต้องสูง และให้แรงเคลื่อนไฟฟ้าสูงแต่มีย่านการใช้งานไม่กว้างเท่ากับเทอร์โมคัปเปิลและมีราคาแพงกว่าพอสมควร

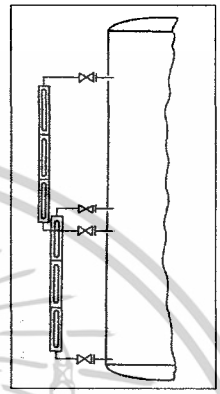
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.9 การวัดระดับด้วย Sight Glass (แบบใช้กระจกมองระดับ)

ลักษณะที่สำคัญคือย่านการวัดมาตรฐานทั่วไปจะใช้วัดระดับไม่เกิน 900 mm หากเกินกว่านี้จะต้องใช้ Sight Glass มากกว่า 2 อัน มาต่อเรียงกันในลักษณะ Overlapping กัน



sight glass 2 อันติดต่อกัน



Sight glass

ภาพที่ 2.32 แสดง Sight Glass แบบต่างๆ

ภาพที่ 2.33 แสดงการติดตั้งใช้งานจริง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.9.1 ข้อดีของ Sight Glass

1. อ่านค่าได้โดยตรง

2. สามารถออกแบบพิเศษให้สามารถทนความดันได้ถึง 10,000 psi และทนอุณหภูมิได้สูงถึง 750 ° F

3. วัสดุที่ใช้ทำ Sight Glass สามารถออกแบบให้ทนต่อสภาพการกัดกร่อนได้เป็นอย่างดี เช่น ใช้พลาสติก หรือแก้ว เป็นต้น

2.9.2 ข้อจำกัดของ Sight Glass

1. การอ่านค่าโดยตรง อาจไม่สะดวก เช่น กรณีที่ติดตั้งไว้บนที่สูงหรือด้านบนของถัง

2. ค่าความแม่นยำ (Accuracy) ขึ้นอยู่กับความสะอาดของช่องมอง สภาพของของไหล เช่น ความใสและผู้อ่าน ซึ่งควบคุมได้ยาก

3. กรณีของเหลวมีสีใส ทำให้ยากสำหรับการอ่านค่า

4. ในสภาพที่มีอากาศหนาวจัด อาจทำให้ของเหลวบริเวณ Sight Glass เกิดการแข็งตัว ในกรณีที่ Sight Glass ติดตั้งไว้ภายนอก

5. การกระเพื่อมของของเหลว ทำให้การอ่านค่ากระทำได้ยาก

2.9.3 ประเภทของ Sight glass

Sight glass แบ่งออกเป็น 2 แบบ คือ

2.9.3.1 แบบหลอดแก้ว (Tubular Glass) ใช้ติดตั้งโดยตรงกับภาชนะที่ต้องการวัดระดับของของเหลวภายใน

ข้อดี

1. ราคาถูก ติดตั้งง่าย ค่าบำรุงรักษาต่ำ

2. สามารถอ่านค่าได้โดยตรงทันที มีความน่าเชื่อถือได้

ข้อจำกัด

1. ใช้งานที่อุณหภูมิสูงไม่ได้

2. ใช้งานที่ความดันสูงไม่ได้

3. แตกง่าย ดังนั้นจึงไม่เหมาะกับระบบที่มีความสำคัญมาก ๆ หรือไม่เหมาะกับการวัดระดับของสารที่เป็นอันตราย เช่น กรดเข้มข้น หรือในบริเวณที่มีอันตรายสูง

4. การมองระดับต้องเข้าไปอ่านค่าใกล้ ๆ บางครั้งก็ยากแต่การสังเกต (ในกรณีที่เป็นสีใส) แต่สามารถแก้ไขโดยใช้แถบสีสะท้อนแสงติดไว้ด้านหลังของหลอดแก้วพร้อม Scale และใช้เลนส์นูนทำเป็นหลอดแก้ว ทำให้การอ่านค่าได้ง่ายขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.9.3.2 แบบแผ่นแก้วเรียบ (Flat Glass) มี 2 แบบ คือ

1. แบบแผ่นแก้วเรียบสะท้อน (Flat Refection Glass) แบบนี้เหมาะสำหรับการวัดระดับของของเหลวที่มีสีใส สะอาด ค่าความหนืดต่ำ โดยส่วนที่เป็นของเหลวจะปรากฏเป็นทึบแสง ส่วนที่เป็นช่องว่าง หรือเหนือระดับของเหลวจะปรากฏเป็นลักษณะโปร่งใส ดังภาพ แผ่นแก้วจะทำจากวัสดุที่มีค่าสัมประสิทธิ์การขยายตัวต่ำมาก ดังนั้นจึงสามารถใช้งานในบริเวณที่มีความร้อนได้ นอกจากนี้ยังสามารถรับแรงกระทำได้ดีอีกด้วยสามารถออกแบบให้ทนความดันได้สูงถึง 4000 psi และทนอุณหภูมิได้สูงถึง 400 °C

ภาพที่ 2.34 แสดง sight glass แบบแผ่นแก้วเรียบ

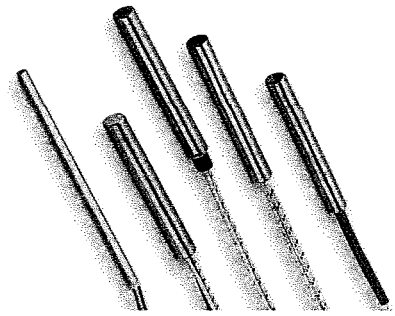
2. แบบหลายช่องมอง (Multi-port Sight Glass) แบบนี้เหมาะสำหรับการวัดระดับของของเหลวที่อยู่ภายใต้สภาวะความดันและอุณหภูมิสูง โดยเฉพาะ เช่น Boiler Drum หลักการวัดระดับของ Boiler Drum โดยใช้ Multiport Sight Glass ใช้หลักการหักเหของแสงในน้ำและไอน้ำ โดยกำหนดให้น้ำเป็นสีเขียว ไอน้ำเป็นสีแดง

2.10 หลักการของฮีตเตอร์

ฮีตเตอร์ เป็นอุปกรณ์ทำความร้อนในอุตสาหกรรม ที่มีหลักการพื้นฐานคือ เมื่อมีกระแสไหลผ่านลวดตัวนำที่มีค่าความต้านทานสูง ลวดตัวนำจะร้อน ดังนั้นลวดที่ใช้ผลิตฮีตเตอร์จะต้องมีคุณสมบัติเหนียวและทนอุณหภูมิได้สูง อีตเตอร์ถูกแบ่งออกเป็นชนิดต่าง ๆ ตามลักษณะการใช้งานที่แตกต่างกัน ดังนี้

2.10.1 ฮีตเตอร์แท่งหรือ Cartridge Heater ลักษณะใช้งานทั่วไป คือ ใส่ไว้ในช่องบนวัตถุความร้อนจะถูกส่งผ่านจากฮีตเตอร์ไปยังวัตถุที่ต้องการให้ความร้อน ตัวอย่างงาน เช่น ให้ความร้อนแม่พิมพ์ของเครื่องบรรจุหีบห่อ Cartridge Heater แบ่งออกเป็น 2 ชนิด คือ High Density และ Low Density

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 2.35 แสดงฮีตเตอร์แท่ง(Cartridge Heater)

2.10.2 ฮีตเตอร์ครีบ (Finned Heater) และฮีตเตอร์ท่อกลม (Tubular Heater) ฮีตเตอร์ครีบทำจาก Tubular Heater ที่ดัดเป็นรูปต่าง ๆ และเพิ่มแผ่นครีบม้วนติดกับท่อฮีตเตอร์อย่างต่อเนื่องจากปลายด้านหนึ่งไปอีกด้านหนึ่ง ส่วนของแผ่นครีบที่เพิ่มขึ้นมาจะทำให้ฮีตเตอร์สามารถถ่ายเทความร้อนได้เร็วขึ้น ส่วนฮีตเตอร์ท่อกลม คือ Tubular Heater ที่ใช้ให้ความร้อนโดยตรงโดยไม่ติดครีบ

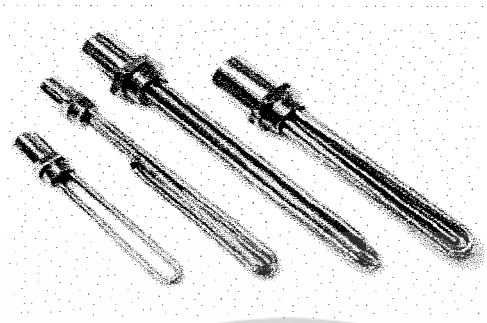
ภาพที่ 2.36 แสดงฮีตเตอร์ครีบ(Finned Heater)

2.10.3 ฮีตเตอร์จุ่มสำหรับของเหลว (Immersion Heater) ทำจาก Tubular Heater ที่ดัดเป็นรูปตัวยูและเชื่อมติดกับเกลียว ซึ่งมีขนาดเกลียวตั้งแต่ 1", $1\frac{1}{4}$", $1\frac{1}{2}$", 2" และ $2\frac{1}{2}$" ขนาดของเกลียวจะขึ้นอยู่กับจำนวนเส้นของฮีตเตอร์ ซึ่งมีตั้งแต่ 1U, 2U, 3U, 6U ตามความเหมาะสมของกำลังวัตต์และความยาวของตัวฮีตเตอร์

ฮีตเตอร์แบบจุ่ม เหมาะสำหรับใช้กับของเหลว เช่น ต้มน้ำหรืออุ่นน้ำมันการติดตั้งสามารถทำได้โดยเชื่อมเกลียวตัวเมียติดกับถังแล้วใส่ฮีตเตอร์แบบเกลียวเข้าไป โดยตัวฮีตเตอร์ขนานกับพื้นถังควรระวังไม่ให้ส่วนของฮีตเตอร์โผล่พ้นของเหลวเนื่องจากจะทำให้ส่วนที่อยู่เหนือของเหลวร้อน

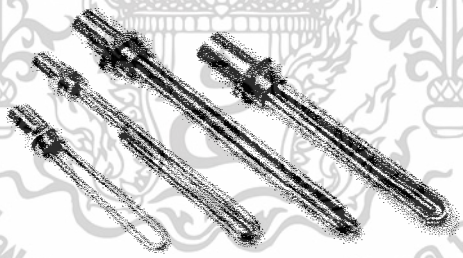
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จัดเกินไป ทำให้อายุการใช้งานสั้นและเพื่อให้ความร้อนกระจายทั่งถึง ควรติดตั้งใบพัดของของเหลวด้วย



ภาพที่ 2.37 แสดงฮีตเตอร์จุ่ม(Immersion Heater)

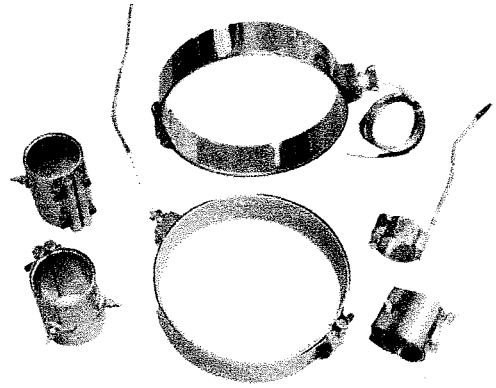
2.10.4 ฮีตเตอร์บอบบิ้น สำหรับของเหลว (Bobbin Heater) เป็นฮีตเตอร์แบบจุ่มชนิดหนึ่ง ถูกออกแบบสำหรับให้วัดความร้อนกับของเหลว สามารถเคลื่อนย้ายได้ง่ายปอกฮีตเตอร์สามารถเลือกให้เหมาะสมกับการใช้งาน มีให้เลือกทั้งสแตนเลส 304 สแตนเลส316 และควอทซ์ โดยแบบสแตนเลส มีข้อดี คือ เมื่อฮีตเตอร์เสียสามารถซ่อมได้ แบบควอทซ์ใช้สำหรับงานชุบโดยใช้ไฟฟ้า แช่ในกรดหรือสารละลาย



ภาพที่ 2.38 แสดงฮีตเตอร์บอบบิ้น(Bobbin Heater)

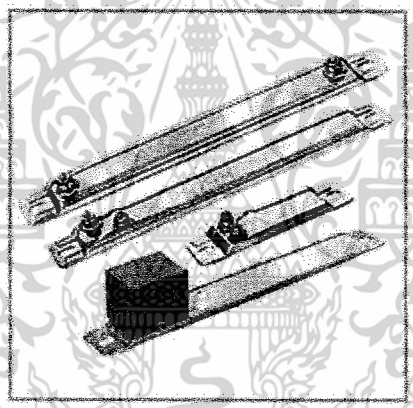
2.10.5 ฮีตเตอร์รัดท่อ (Band Heater) ได้รับการออกแบบสำหรับท่อ หรือถังรูปทรงกระบอก ฉนวนของฮีตเตอร์ทำจากแผ่นไมก้า ลวดฮีตเตอร์เป็นแบบแบน (Ribbon wire heating element) จึงทำให้ฮีตเตอร์ชนิดนี้มีเส้นผ่านศูนย์กลางเล็กขนาด 25 มม. หรืออาจใหญ่ถึง 600 มม. ก็ได้ส่วนความกว้างอยู่ระหว่าง 20-300 มม. ถึงด้านนอกเป็นแผ่นเหล็กหรือสแตนเลสเหมาะสำหรับให้ความร้อนกับเครื่องฉีกพลาสติก มีอีกชื่อนึงว่าฮีตเตอร์กระบอก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



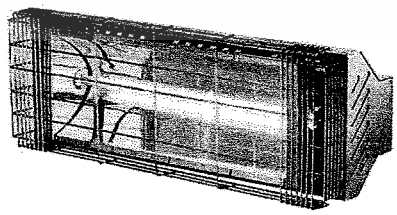
ภาพที่ 2.39 แสดงฮีตเตอร์รัดท่อ (Band Heater)

2.10.6 ฮีตเตอร์แผ่น (Strip Heater) โครงสร้างจะเป็นแบบเดียวกับฮีตเตอร์รัดท่อ แต่รูปทรงจะเป็นแบบสี่เหลี่ยมจัตุรัสหรือสี่เหลี่ยมผืนผ้า เหมาะสำหรับให้ความร้อนกับแม่พิมพ์



ภาพที่ 2.40 แสดงฮีตเตอร์แผ่น (Strip Heater)

2.10.7 ฮีตเตอร์อินฟราเรด (Infrared Heater)



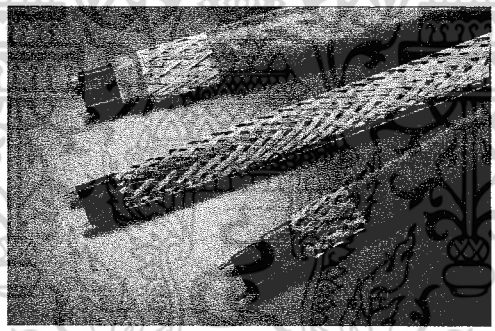
ภาพที่ 2.41 แสดงฮีตเตอร์อินฟราเรด (Infrared Heater)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ลักษณะของอินฟราเรดฮีตเตอร์

- เป็นการส่งผ่านความร้อนแบบแผ่รังสี (เหมือนที่ดวงอาทิตย์ส่งความร้อนมายังโลก)จึงมีประสิทธิภาพสูง ความสูญเสียต่ำ ประหยัดไฟได้ถึง 30-50%
- สามารถให้ความร้อนวัตถุได้ถึงเนื้อในจึงทำให้ประหยัดเวลาได้ 1- 10 เท่า
- มีขนาดเล็กกว่าฮีตเตอร์ทั่วๆไป ทำให้ประหยัดเนื้อที่
- การติดตั้ง และการถอดเปลี่ยนเพื่อซ่อมบำรุงง่าย
- มีความปลอดภัยสูงเนื่องจากไม่มีเปลวไฟ ตัวเรือนมีความเป็นฉนวนสูง ไฟไม่รั่ว
- ให้รังสีช่วง 3-10 μm. ซึ่งเป็นช่วงที่วัสดุเกือบทุกชนิดสามารถดูดซับรังสีได้ดี

2.10.8 ฮีตเตอร์เส้น (Cable heater) ทำงานด้วยหลักการพื้นฐาน โดยการปรับกำลัง output ให้เหมาะสมเมื่ออุณหภูมิเพิ่มขึ้นกำลัง output จะลดลง ฮีตเตอร์เส้นแบบนี้ จะไม่สามารถทำให้เกิดความร้อนสูงได้ ด้วยเหตุนี้จึงเหมาะที่จะประยุกต์ใช้งานเพื่อป้องกันการจับตัวเป็นน้ำแข็งในสภาวะที่อุณหภูมิต่ำ ๆ หรือนำไปประยุกต์ใช้ในอุตสาหกรรมทำความเย็นด้วยโครงสร้างของฮีตเตอร์เส้นจึงสามารถใช้กับท่อและข้อต่อพลาสติกหรือPVCต่างๆได้อย่างปลอดภัย



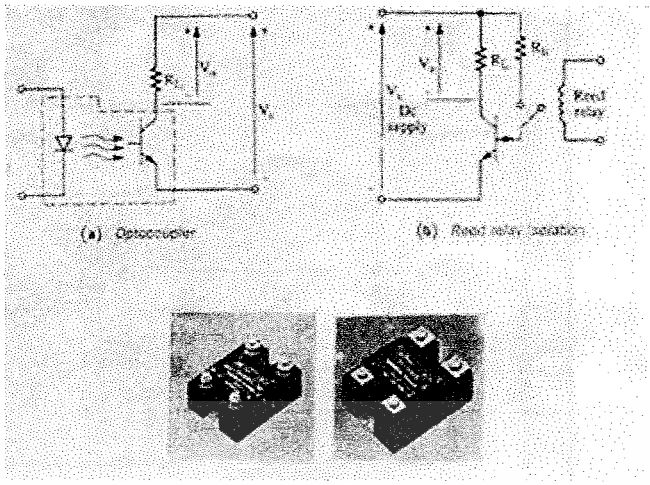
ภาพที่ 2.42 แสดงฮีตเตอร์เส้น (Cable heater)

2.11 โซลิดสเตทรีเลย์

โซลิดสเตทรีเลย์ เป็นการนำเอาสวิตช์สถิตไปใช้งานเพื่อควบคุมกำลังไฟฟ้ากระแสตรงและกระแสสลับ ใช้งานมากในงานอุตสาหกรรม เช่น ควบคุมการสตาร์ทมอเตอร์แทนแมกเนติกคอนแทคเตอร์ ควบคุมหม้อแปลงไฟฟ้าฯลฯ

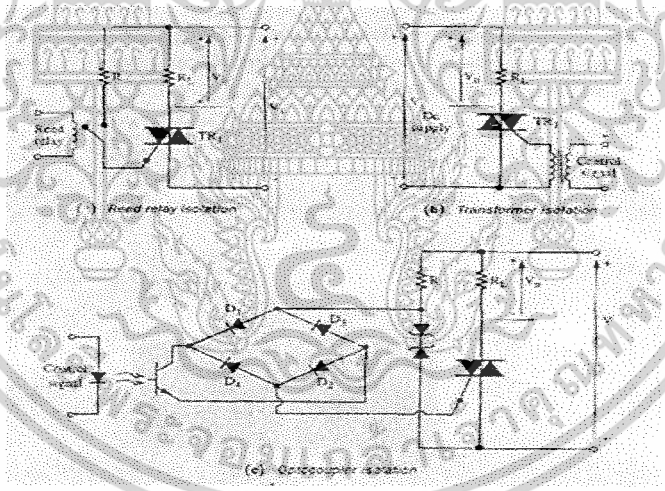
โซลิดสเตทรีเลย์ (SSR) แบบไฟฟ้ากระแสตรงจะใช้ทรานซิสเตอร์เป็นสวิตช์ โดยรับสัญญาณควบคุมจากไฟกระแสตรงจำนวนเล็กภายนอก โดยผ่านทางหรือผ่าน Reed Relay ดังรูปลักษณะภายนอกของโซลิดสเตทรีเลย์แบบไฟฟ้ากระแสตรงแสดงในรูป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 2.43 โซลิดสเตทรีเลย์แบบไฟฟ้ากระแสตรง

โซลิดสเตทรีเลย์(SSR) ที่ควบคุมโหลดไฟสลับ นิยมใช้ไทรแอกเป็นสวิตช์สถิต ดังในรูป ใช้สัญญาณควบคุมป้อนเข้าที่ Reed Relay และใช้สัญญาณควบคุมผ่านหม้อแปลงพัลส์และจากรูปเป็น Isolation แบบ Optocoupler โซลิดสเตทรีเลย์ที่ขับโหลดไฟสลับแสดงดังรูป



ภาพที่ 2.44 โซลิดสเตทรีเลย์ที่ขับโหลดไฟสลับ

โซลิดสเตทรีเลย์ มีหลายชนิด ดังนี้

- ZS : ZERO SWITCHING ใช้ได้กับโหลดทุกประเภท
- IO : INSTANT-ON SWITCHING โหลดที่ต้องการความเร็วในการ ON-OFF
- PS : PEAK SWITCHING ใช้ได้กับโหลดที่เป็นหม้อแปลง
- AS : ANALOG SWITCHING ใช้ได้กับโหลดทุกประเภท

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- DCS : DC SWITCHING ใช้ได้กับโหลดทุกประเภท

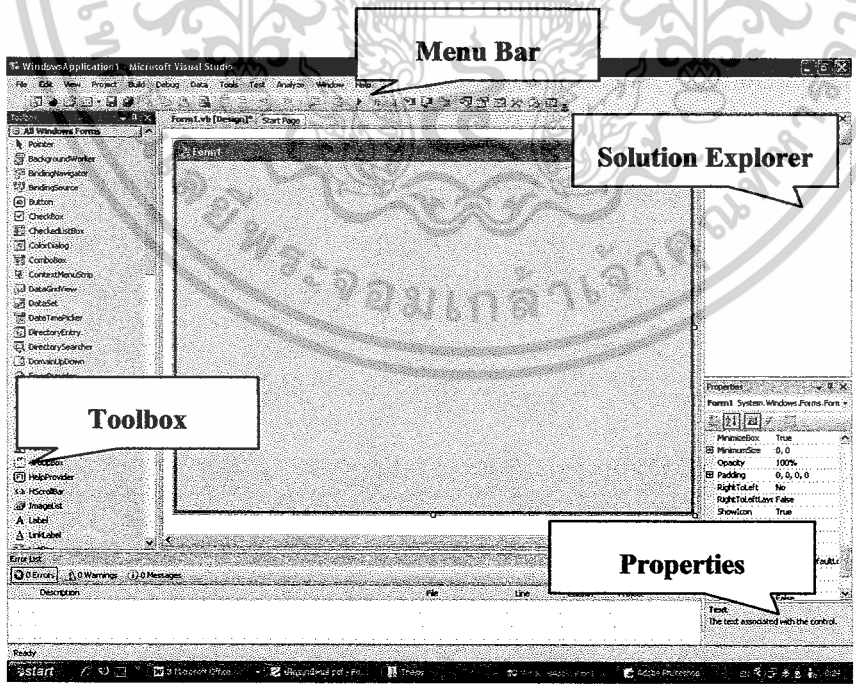
2.12 การพัฒนาซอฟแวร์โดยใช้ Visual Basic 2008

ภาษา Visual Basic มีวิวัฒนาการมาจากภาษา BASIC (Beginner's All-Purpose Symbolic Instruction Code) ซึ่งในยุคหนึ่ง คือภาษาที่ใช้เขียนโปรแกรมบนระบบปฏิบัติการ MS-DOS ต่อมาไมโครซอฟท์ได้พัฒนาภาษา BASIC มาเป็น Visual Basic เพื่อให้เป็นภาษาสำหรับสร้างโปรแกรมที่แสดงผลแบบกราฟิก โดยมีสภาพแวดล้อมในการพัฒนาแบบ visual programming ภาษา Visual Basic เริ่มเป็นที่รู้จักแพร่หลายในเวอร์ชั่น 3 และหลังจากนั้นเป็นต้นมา Visual Basic ก็ถือได้ว่าเป็นภาษาที่มีคนใช้งานมากที่สุดในโลกภาษาหนึ่ง ด้วยจุดเด่นคือสามารถใช้สร้างโปรแกรมได้อย่างสะดวกรวดเร็ว

การเขียนโปรแกรมด้วยภาษา Visual Basic นั้นเป็นรูปแบบที่เรียกว่า event-driven programming หรือ "การเขียนโปรแกรมโดยอิงตามเหตุการณ์" หมายถึงใช้เหตุการณ์เป็นตัวขับเคลื่อนให้เกิดการประมวลผลประโยคคำสั่งในโปรแกรม ทั้งนี้ออบเจ็คต่าง ๆ จะมีเหตุการณ์หรืออีเวนต์ (event) ประจำตัวที่สามารถเกิดขึ้นมาได้ อีเวนต์ส่วนใหญ่จะเกิดขึ้นโดยเป็นผลมาจากการกระทำของผู้ใช้ เช่น ปุ่มคำสั่งมีอีเวนต์ Click ซึ่งจะเกิดขึ้นเมื่อผู้ใช้คลิกปุ่ม เป็นต้น

2.12.1 ส่วนประกอบในหน้าจอ IDE ของ Visual Studio

หน้าจอ IDE ของ Visual Studio ประกอบด้วยส่วนหลัก ๆ ดังนี้



ภาพที่ 2.45 ส่วนประกอบในหน้าจอ IDE ของ Visual Studio

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมนูบาร์ (Menu Bar)

เป็นส่วนที่รวบรวมคำสั่งสำหรับให้เราใช้ควบคุมการทำงานของ Visual Studio โดยแบ่งออกเป็นกลุ่มคำสั่งหรือเมนู (menu) ต่างๆ ซึ่งในแต่ละเมนูจะประกอบด้วยคำสั่ง (command หรือบางทีก็เรียกว่า menu item) ที่เกี่ยวข้องกัน

วินโดว์ Solution Explorer

ทำหน้าที่แสดงรายชื่อไอเท็ม (item) ที่เป็นส่วนประกอบของโซลูชั่น เพื่อให้คุณทำงานกับไอเท็มเหล่านั้นได้ โดยในตอนเริ่มต้นที่เปิด Visual Studio ขึ้นมา Solution Explorer จะว่างเปล่าต่อเมื่อคุณเปิดโปรเจ็คหรือโซลูชั่นขึ้นมาแล้ว จึงจะมีรายชื่อไอเท็มแสดงอยู่ภายในวินโดว์นี้

วินโดว์ Properties

พร็อพเพอร์ตี้ (Property) หมายถึงคุณสมบัติหรือคุณลักษณะของออบเจ็ค เช่น สี, ขนาด ตำแหน่ง ฯลฯ ในช่วงเขียนโปรแกรมนั้นพร็อพเพอร์ตี้ต่าง ๆ ของฟอร์มหรือออบเจ็คบนฟอร์มที่คุณคลิกเลือกไว้ในวินโดว์ Windows Form Designer จะถูกแสดงค่าออกมาในวินโดว์ Properties นี้ และคุณสามารถแก้ไขค่าของพร็อพเพอร์ตี้ได้ โดยใช้วินโดว์นี้เช่นเดียวกันซึ่งถือเป็นการกำหนดพร็อพเพอร์ตี้อีกวิธีหนึ่งจะทำให้โดยใช้โค้ดโปรแกรม ซึ่งถือเป็นการกำหนดพร็อพเพอร์ตี้ในช่วงรันโปรแกรม (run time)

ทูลบ็อกซ์ (Toolbox)

Visual Studio จัดว่าเป็นสภาพแวดล้อมในการพัฒนาโปรแกรมแบบ visual programming หมายถึงความว่า คุณสามารถสร้างหน้าจอหรือส่วนติดต่อกับผู้ใช้ (user interface) ของโปรแกรมได้อย่างง่ายดาย ด้วยการคลิกลากคอนโทรล (control) จากทูลบ็อกซ์มาปล่อยลงบนฟอร์ม แล้วปรับเปลี่ยนขนาดและตำแหน่งของคอนโทรลตามที่ต้องการ

คอนโทรลในทูลบ็อกซ์แบ่งออกเป็น 7 กลุ่มหลัก ๆ ดังนี้

Common Controls	คอนโทรลพื้นฐานต่าง ๆ เช่น ปุ่ม, เท็กซ์บ๊อกซ์, ลิสต์บ๊อกซ์, PictureBox
Containers	คอนโทรลที่ใช้จัดกลุ่มให้กับคอนโทรลอื่น ๆ โดยบรรจุคอนโทรลเหล่านั้นไว้ภายในตัวมัน เช่น GroupBox และ Panel เป็นต้น
Menu & Toolbars	คอนโทรลที่ใช้สร้างเมนูและทูลบาร์
Data	คอนโทรลที่ใช้ทำงานกับฐานข้อมูล
Components	คอนโทรลที่ไม่แสแสดงรูปร่างหน้าตาออกมาบนฟอร์ม แต่จัดเตรียมฟังก์ชั่นการทำงานบางอย่างกับโปรแกรม เช่น Timer และ SerialPort

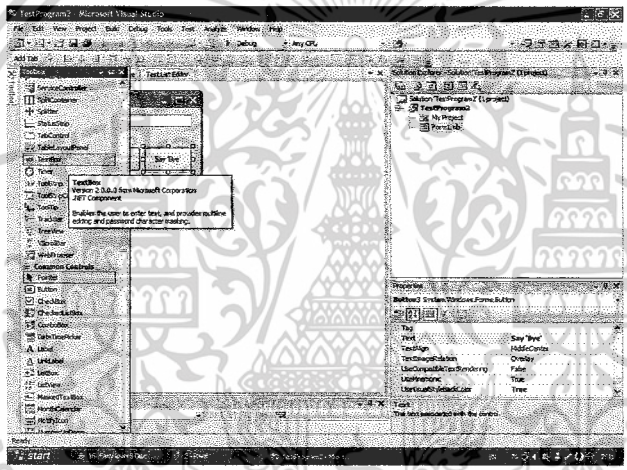
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Printing	คอนโทรลที่เกี่ยวข้องกับการพิมพ์เอกสารออกทางพรินเตอร์
Dialogs	คอนโทรลที่ใช้แสดงไดอะล็อกบ็อกซ์พื้นฐานชนิดต่าง ๆ เช่น ไดอะล็อกบ็อกซ์สำหรับเปิดไฟล์ และไดอะล็อกบ็อกซ์สำหรับเลือกสี เป็นต้น

2.12.2 ขั้นตอนการเขียนโปรแกรมด้วย Visual Basic

การเขียนโปรแกรมด้วยภาษา Visual Basic โดยใช้ Visual Studio เป็นเครื่องมือนั้นจะมีขั้นตอนหลัก ๆ ดังนี้

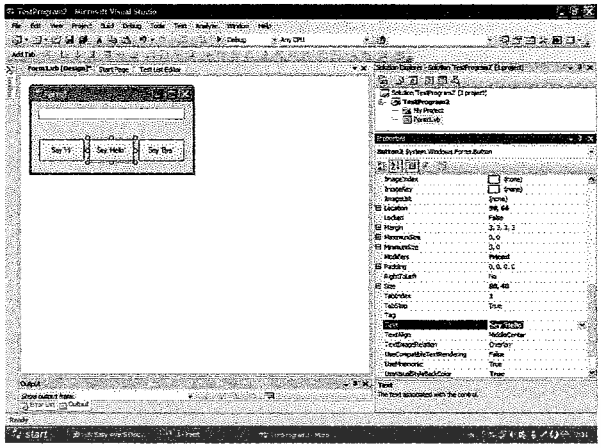
1. ออกแบบหน้าจอ หรือ User interface ของโปรแกรม โดยนำคอนโทรลต่าง ๆ จากทูลบ็อกซ์มาวางลงบนฟอร์ม แล้วปรับขนาดและตำแหน่งตามที่ต้องการ



ภาพที่ 2.46 หน้าจอการออกแบบ User Interface

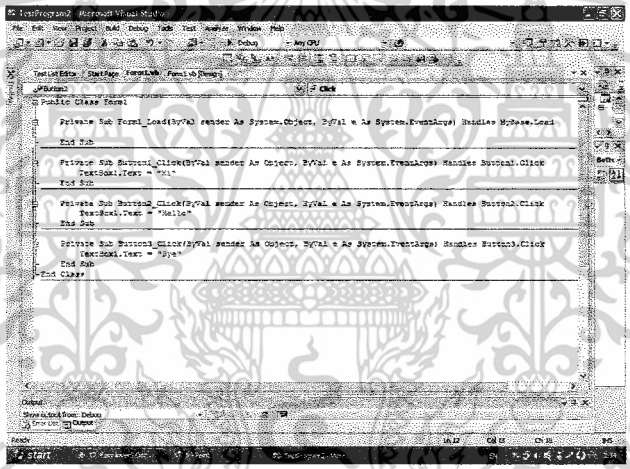
2. กำหนดพร็อพเพอร์ตี้ของออบเจ็คต่าง ๆ (ฟอร์ม คอนโทรลต่าง ๆ ที่เรานำมาวางบนฟอร์ม) โดยใช้วินโดว์ Properties

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



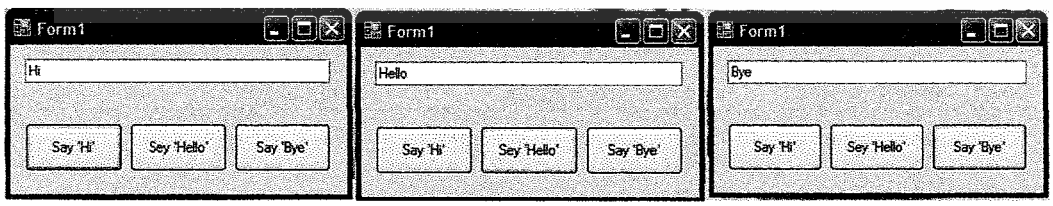
ภาพที่ 2.47 การตั้งค่าพร็อพเพอตี้ให้กับคอนโทรล

3. เขียนโค้ดควบคุมการทำงานของโปรแกรม



ภาพที่ 2.48 การเขียนโปรแกรมควบคุมการทำงาน

4. รันโปรแกรมเพื่อทดสอบดูการทำงานว่าเป็นไปตามที่ต้องการหรือไม่



ภาพที่ 2.49 โปรแกรมขณะทดสอบการทำงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5. ตรวจสอบหาข้อผิดพลาดของโปรแกรม ถ้าหากการทำงานเกิดข้อผิดพลาดหรือไม่เป็นไปตามที่ต้องการ

6. คอมไพล์โปรเจ็คไปเป็นไฟล์โปรแกรมที่รันได้ (Executable file)

2.13 ความรู้พื้นฐานเกี่ยวกับฐานข้อมูลและภาษา SQL

2.13.1 ความหมายของฐานข้อมูล และระบบจัดการฐานข้อมูล (DBMS)

ฐานข้อมูล (Database) คือกลุ่มของข้อมูลที่ถูกเก็บรวบรวมไว้ในที่เดียวกันอย่างเป็นระบบ เพื่อให้สามารถค้นหา เพิ่มเติม ลบ และแก้ไขข้อมูลได้อย่างสะดวก และมีประสิทธิภาพคุณสามารถสร้างฐานข้อมูลได้โดยใช้โปรแกรม อย่างเช่น Microsoft Access Oracle MySQL Paradox และ Microsoft SQL Server เป็นต้น โปรแกรมเหล่านี้นอกจากใช้สร้างฐานข้อมูลขึ้นมาแล้ว ยังมีหน้าที่จัดการและดำเนินการกับฐานข้อมูลตามที่โปรแกรมผู้ใช้ข้อมูล (data consumer) ร้องขอมาด้วยเราจึงเรียกมันว่า โปรแกรมระบบจัดการฐานข้อมูล (database management system-DBMS)

ฐานข้อมูลที่สร้างจากโปรแกรม DBMS แต่ละโปรแกรมจะมีรูปแบบไฟล์ที่แตกต่างกันไป หรือกล่าวได้ว่าเป็นข้อมูลคนละประเภทกัน อย่างไรก็ตามเมื่อใช้บริการของ .NET Framework คุณสามารถที่จะเขียนโปรแกรมเพื่อติดต่อและทำงานกบฐานข้อมูลทุกประเภทได้โดย ใช้คำสั่งแบบเดียวกัน

2.13.2 ฐานข้อมูลเชิงสัมพันธ์ และคำศัพท์ต่าง ๆ ที่เกี่ยวข้อง

เทเบิล (Table)

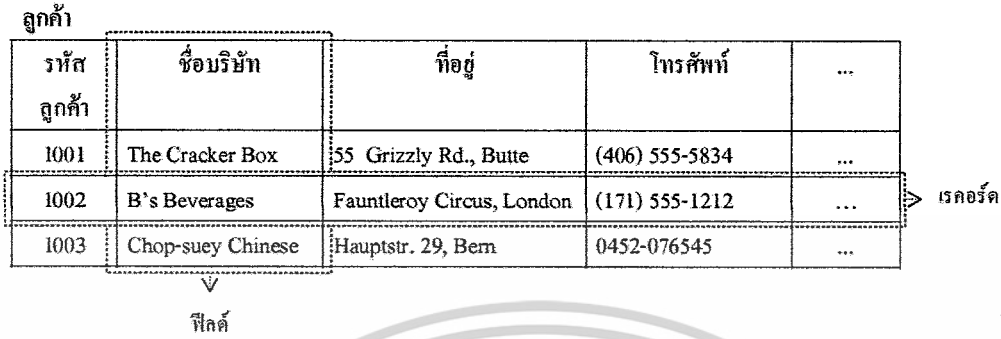
ฐานข้อมูลที่ใช้กันอย่างแพร่หลายในปัจจุบันนี้เป็นรูปแบบหรือโมเดล (Model) ที่เรียกว่าฐานข้อมูลเชิงสัมพันธ์ (relational database) ซึ่งในฐานะข้อมูลหนึ่ง ๆ จะประกอบด้วยตารางข้อมูลหรือเทเบิล (table) ตั้งแต่ 1 เทเบิลขึ้นไป แต่ละเทเบิลจะเก็บข้อมูลเกี่ยวกับสิ่งใดสิ่งหนึ่ง (entity) เอาไว้ เช่น ฐานข้อมูลในระบบงานขายสินค้า อาจมีเทเบิลหนึ่งเก็บข้อมูลเกี่ยวกับลูกค้า, เทเบิลหนึ่งเก็บข้อมูลเกี่ยวกับสินค้า และอีกเทเบิลหนึ่งเห็บข้อมูลเกี่ยวกับการสั่งซื้อสินค้าของลูกค้า เป็นต้น

เรคอร์ดและฟิลด์

เทเบิลจะแบ่งออกเป็นแถว (row) และคอลัมน์ (coulmn) โดยแต่ละแถวในเทเบิลก็คือข้อมูล 1 รายการหรือ 1 เรคอร์ด (record) เช่นสำหรับเทเบิลที่เก็บข้อมูลลูกค้านั้น แต่ละเรคอร์ดก็จะเป็นข้อมูลเกี่ยวกับลูกค้า 1 คน เป็นต้น ข้อมูลแต่ละส่วนของเรคอร์ดหนึ่ง ๆ จะถูกแยกเก็บไว้ในคอลัมน์ต่าง ๆซึ่ง เรียกว่า ฟิลด์ (field) เช่น เทเบิล “ลูกค้า” ก็อาจมีฟิลด์สำหรับเก็บรหัสลูกค้า, ฟิลด์สำหรับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เก็บชื่อลูกค้า (ชื่อบริษัท), ฟิลด์สำหรับเก็บที่อยู่ของลูกค้า และฟิลด์สำหรับเก็บหมายเลขโทรศัพท์ที่ใช้ติดต่อ ฯลฯ ดังรูป



ภาพที่ 2.50 เรคอร์ดและฟิลด์

เรามักกำหนดให้ฟิลด์ใดฟิลด์หนึ่ง หรือกลุ่มของฟิลด์ ทำหน้าที่เป็น "คีย์หลัก" หรือ primary key ของเทเบิล ฟิลด์ที่ถูกกำหนดให้เป็น primary key จะใช้แยกแยะเรคอร์ดต่างๆ ในเทเบิลนั้น หรือพูดง่าย ๆ คือใช้บ่งชี้ (identify) ว่าเรคอร์ดใดเป็นเรคอร์ดใด ด้วยเหตุนี้ฟิลด์ที่เป็น primary key จึงต้องมีค่าอยู่ในทุดเรคอร์ด ไม่สามารถปล่อยว่างได้ และที่สำคัญคือค่าของฟิลด์นั้นหรือฟิลด์เหล่านั้นในเรคอร์ดต่าง ๆ จะต้องไม่ซ้ำกัน โดยทั่วไปถ้าไม่มีฟิลด์หรือกลุ่มของฟิลด์ใดๆ ที่มีลักษณะดังกล่าวเราจะสร้างฟิลด์ขึ้นมาทำหน้าที่เป็น primary key โดยเฉพาะ เช่น ฟิลด์ "รหัสลูกค้า" ในรูปข้างต้น

คีย์นอก

นอกจาก primary key แล้วยังมีคีย์ที่สำคัญอีกชนิดหนึ่งเรียกว่า "คีย์นอก" หรือ foreign key ซึ่งหมายถึงฟิลด์ที่เก็บค่าของฟิลด์ที่เป็น primary key ในเทเบิลอื่นไว้ ทั้งนี้เพื่อแสดงถึงการเชื่องโยงระหว่างเรคอร์ดในเทเบิลทั้งสอง (เทเบิลของ foreign key กับเทเบิลของ primary key ที่ foreign key นั้นอ้างอิงถึง) เช่น เทเบิล "การสั่งซื้อ" ที่เก็บข้อมูลการสั่งซื้อสินค้า (order) อาจมีฟิลด์ "รหัสลูกค้า" สำหรับเก็บรหัสของลูกค้าผู้ที่สั่ง order นั้นมา ซึ่งเป็นการเชื่อมโยงไปยังฟิลด์ "รหัสลูกค้า" ที่เป็น primary key ของเทเบิล "ลูกค้า" การเชื่อมโยงนี้ช่วยให้เราทราบข้อมูลเกี่ยวกับลูกค้าผู้สั่ง order โดยไม่ต้องเก็บข้อมูลทั้งหมดของลูกค้าลงในเทเบิล "การสั่งซื้อ" โดยตรง นตัวอย่างนี้ฟิลด์ "รหัสลูกค้า" ของเทเบิล "การสั่งซื้อ" จึงทำหน้าที่เป็น foreign key

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ลูกค้า

รหัสลูกค้า	ชื่อบริษัท	ที่อยู่	โทรศัพท์	...
1001	The Cracker Box	55 Grizzly Rd., Butte	(406) 555-5834	...
1002	B's Beverages	Fauntleroy Circus, London	(171) 555-1212	...
1003	Chop-suey Chinese	Hauptstr. 29, Bern	0452-076545	...

การสั่งซื้อ

รหัสสั่งซื้อ	รหัสลูกค้า	รหัสสินค้า	...
12010	1001	BE-203	...
12015	1003	ME-010	...
12016	1002	SE-057	...
12025	1003	SE-057	...

ภาพที่ 2.51 ความสัมพันธ์ระหว่างตาราง

ความสัมพันธ์ระหว่างเทเบิล

เมื่อแต่ละเรคอร์ดในเทเบิลหนึ่งเชื่อมโยงไปยังหลายเรคอร์ดในอีกเทเบิลหนึ่ง เช่น กรณีของเทเบิล "ลูกค้า" กับเทเบิล "การสั่งซื้อ" เราจะกล่าวว่าความสัมพันธ์ (relationship) ระหว่างเทเบิลทั้งสองเป็นแบบ "1 เรคอร์ดต่อหลายเรคอร์ด" (one-to-many) หรืออาจเรียกว่าความสัมพันธ์แบบ parent-child หรือ master-details ก็ได้ ความสัมพันธ์ระหว่างเทเบิลยังมีอีก 2 แบบ คือ one-to-one และ many-to-many ซึ่งความหมายตรงไปตรงมาตามชื่อ แต่อย่างไรก็ตาม โดยหลักการของการออกแบบฐานข้อมูลแล้วจะอนุญาตให้มีเฉพาะความสัมพันธ์แบบ one-to-one และ one-to-many เท่านั้น

2.14 โปรแกรมสำหรับจัดการฐานข้อมูล Microsoft SQL Server 2005 Express

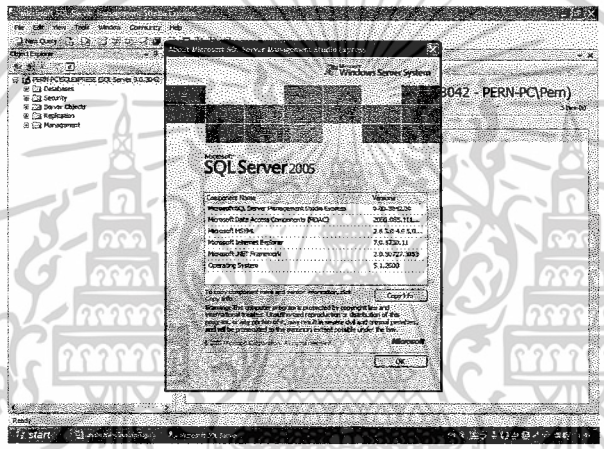
ในสภาวะแวดล้อมปัจจุบันข้อมูลได้กลายเป็นทรัพยากรที่สำคัญที่สุดขององค์กร ข้อมูลเป็นสิ่งจำเป็นอย่างยิ่งในการตัดสินใจของผู้บริหาร เพื่อไปใช้ในการวางแผน และกำหนดแนวทางขององค์กรการบันทึก จัดเก็บข้อมูลจำเป็นอย่างยิ่งที่จะต้องมีเครื่องมือที่สามารถบริหารจัดการข้อมูลได้อย่างมีประสิทธิภาพและมีความปลอดภัยสูงเชื่อถือได้ สามารถตอบสนองต่อความต้องการของผู้ใช้งานได้อย่างครบถ้วน เครื่องมือที่กล่าวถึงนั้นก็คือ โปรแกรมสำหรับจัดการระบบฐานข้อมูล (Relational Database Management System หรือ เรียกแบบย่อว่า RDBMS) ทางบริษัทไมโครซอฟท์ได้ผลิตโปรแกรมสำหรับจัดการฐานข้อมูลรุ่นใหม่ โดยมีชื่อเรียกว่า SQL Server 2005 ซึ่งมีอยู่ด้วยกันหลายรุ่น ได้แก่ SQL Server 2005 Enterprise SQL Server 2005 Standard SQL Server - 2005 Workgroup และ SQL Server 2005 Express โดยมีจุดเด่นที่สามารถใช้งานง่ายมีความสามารถ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในการให้บริการในระดับสูง มีความปลอดภัยและน่าเชื่อถือ ซึ่งเป็นผลิตภัณฑ์ที่สามารถใช้งานได้อย่างครบวงจร

SQL Server 2005 Express ได้รับการออกแบบมาเพื่อช่วยให้สามารถติดตั้ง และดูแลรักษาได้ง่ายด้วยการมีฐานข้อมูลแบบโลคัลในตัว ซึ่งจะช่วยลดค่าใช้จ่ายในการติดตั้งและจัดการ พร้อมทั้งมีเครื่องมือที่ช่วยในการจัดการฐานข้อมูลมาให้อย่างครบครันไม่ด้อยไปกว่าSQL Server 2005 เวอร์-ชันเต็มสำหรับเครื่องมือที่ช่วยในการจัดการฐานข้อมูลที่กล่าวถึงก็คือ SQL Server Management Studio Express

2.14.1 Microsoft SQL Server Management Studio

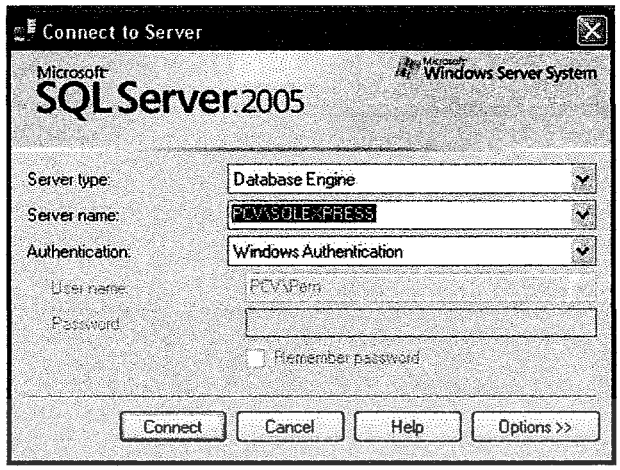


ภาพที่ 2.52 หน้าตาโปรแกรม Microsoft SQL Server 2005 Express

เครื่องมือที่ใช้ในการบริหารจัดการระบบฐานข้อมูลของ SQL Server 2005 เราจะใช้ผ่านเครื่องมือที่เรียกว่า SQL Server Management Studio สำหรับเครื่องมือนี้จะมีมาพร้อมกับ SQL Server 2005 Express with Advance Service หรือรุ่นที่สูงกว่า ถ้าหากผู้อ่านเคยใช้งาน SQL Server 2000 ก็จะเป็นเครื่องมือที่คล้ายกับ Enterprise Manager และ Query Analyzer ที่นำมารวมไว้ด้วยกัน

1. คลิกที่เมนู **Start** จากนั้นเลือก **All Programs** เลือกรายการ **Microsoft SQL Server 2005** คลิกที่โปรแกรม **SQL Server Management Studio Express** จะเข้าสู่โปรแกรมดังภาพ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 2.53 หน้าจอ Login เข้าสู่โปรแกรม Microsoft SQL Server Management Studio Express

Server Name กำหนดดาตาเบสเซิร์ฟเวอร์ที่ต้องการติดต่อ

Authentication กำหนดโหมดในการเข้าถึงดาตาเบสเซิร์ฟเวอร์ ประกอบด้วย Windows Authentication เข้าโดยใช้ผู้ใช้และรหัสผ่านของ Windows SQL Server Authentication เข้าโดยใช้ชื่อผู้ใช้งานของ SQL Server

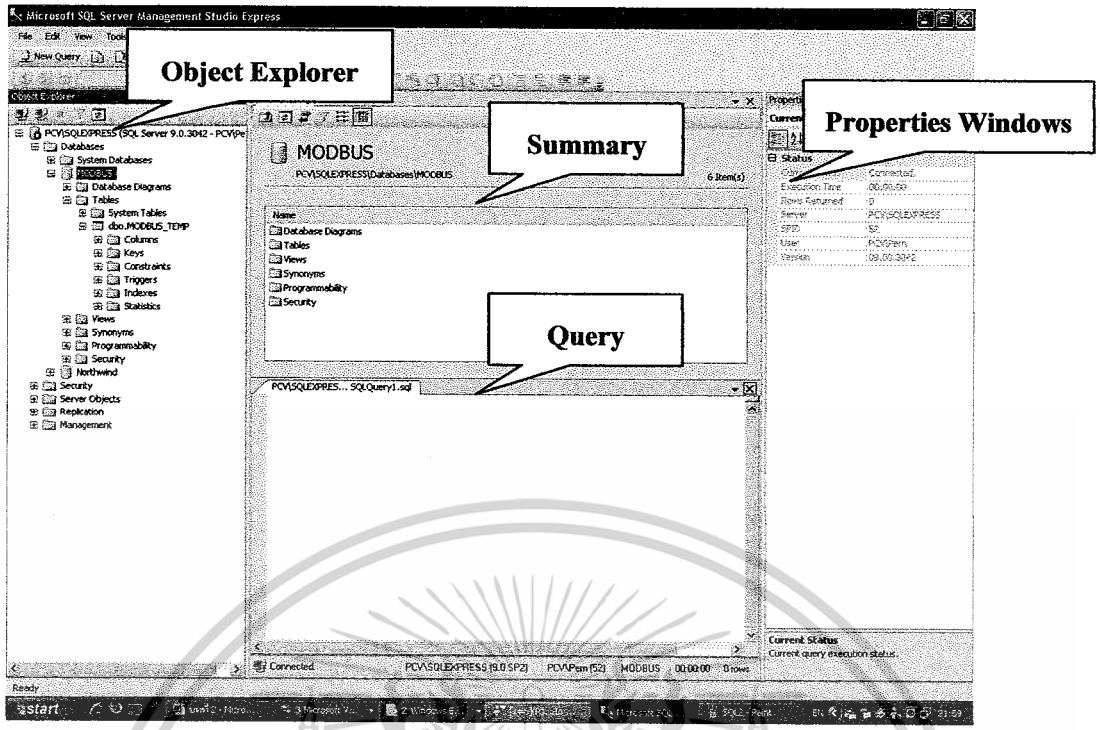
Login คือ ชื่อผู้ใช้สำหรับที่กำหนดโดย SQL Server

Password คือ รหัสผ่านสำหรับผู้ใช้ที่กำหนดโดย SQL Server

ปกติการเข้าใช้งานของ SQL Server จะนิยมเข้าด้วยโหมด SQL Server Authentication จากนั้นป้อน Login และ Password ที่กำหนดโดย SQL Server สำหรับชื่อ Login ที่มีสิทธิสูงสุดจะมีชื่อว่า SA ส่วนรหัสผ่านจะเป็นรหัสผ่านที่กำหนดไว้ในตอนติดตั้ง SQL Server 2005 Express

2. หลังจากป้อนชื่อ Login และ Password ให้คลิกปุ่ม Connect ก็จะเข้าสู่หน้าจอของโปรแกรม Microsoft SQL Server Management Studio Express ดังภาพ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 2.54 หน้าตาโปรแกรม Microsoft SQL Server Management Studio Express

องค์ประกอบที่สำคัญของ Microsoft SQL Server Management Studio Express

Object Explorer	ใช้สำหรับแสดงและจัดการกับออบเจกต์ต่าง ๆ ของคาตาเบสเซิร์ฟเวอร์
Summary	แสดงรายละเอียดที่อยู่ภายใต้ Object Explorer
Query Editor	ใช้สำหรับป้อนและประมวลผลคำสั่งคิวรี เราสามารถเปิดหน้าต่าง Query Editor ได้โดยคลิกที่ปุ่ม New Query
Properties Windows	ใช้สำหรับแสดงและ กำหนดคุณสมบัติของออบเจกต์ต่าง ๆ สามารถเปิดได้โดยเลือกเมนู View แล้วเลือก Properties Windows หรือกดปุ่มฟังก์ชันคีย์ F4 ที่แป้นพิมพ์

2.14.2 ประเภทของฐานข้อมูลของ SQL Server 2005

SQL Server 2005 จะแบ่งแยกฐานข้อมูลออกเป็น 2 ประเภทได้แก่

- ฐานข้อมูลระบบ (System Databases)
- ฐานข้อมูลผู้ใช้ (User Database)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ฐานข้อมูลระบบ (System Databases)

เมื่อเราทำการติดตั้งโปรแกรม SQL Server 2005 จะมีฐานข้อมูลของระบบ (System Databases) ถูกสร้างขึ้นมาจำนวน 4 ฐาน ข้อมูล ด้แก่ master model msdb tempdb ซึ่งทั้งหมดจะถูกเก็บไว้ที่โฟลเดอร์ **C:\Program Files\Microsoft SQL Serer\MSSQL.1\Data** ดาตาเบสเหล่านี้ห้ามลบออกในทุกกรณี เพราะอาจจะทำให้ระบบฐานข้อมูลเกิดความเสียหายได้ ฐานข้อมูลระบบประกอบด้วยฐานข้อมูลต่าง ๆ ดังนี้

Master เป็นฐานข้อมูลที่จัดเก็บค่าคอนฟีกิวเรชันของระบบจัดเก็บข้อมูล และรายละเอียดของฐานข้อมูลที่เราสร้างขึ้นและฐานข้อมูลของระบบข้อมูลเหล่านี้ จะถูกเก็บไว้ที่ตารางชื่อ sysdatabases และทำการบสตอโพรซิเดอร์ของระบบ วิวของระบบ เป็นต้น

Model เป็นฐานข้อมูลที่ใช้เป็นต้นแบบสำหรับสร้างฐานข้อมูลใหม่ กรณีที่ผู้ใช้เริ่มทำการสร้างฐานข้อมูลระบบจะทำการสำเนาข้อมูลที่ใช้งานมาจากฐานข้อมูล model และตัวฐานข้อมูล msdb ก็ยังเป็นต้นแบบสำหรับฐานข้อมูล tempdb เมื่อเริ่มสตาร์ต SQL Server 2005 ด้วยเช่นกัน

Msdb ฐานข้อมูลนี้ จะเก็บงานของระบบเพื่อที่จะส่งให้ SQL Server Agent ทำการประมวลผล เช่น กำหนดตารางเวลาสำหรับทำการสำรองข้อมูลไว้ล่วงหน้าข้อมูลจะถูกเก็บใน msdb เพื่อรอจนถึงเวลาที่กำหนดไว้ จึงเริ่มทำการสำรองข้อมูล หรือ การสั่งให้สตอโพรซิเดอร์ทำการประมวลผลในเวลาที่กำหนดไว้

Tempdb เป็นพื้นที่ชั่วคราวที่ใช้ในการทำงาน เช่น หากมีการคิวรีข้อมูลที่ซับซ้อนจำนวนข้อมูลมีเป็นจำนวนมาก จำเป็นต้องใช้พื้นที่ในการทำงานเพิ่มเติมก็จะมีการใช้เนื้อที่ในฐานข้อมูล tempdb เพื่อใช้เป็นพื้นที่ในการทำงานชั่วคราว สำหรับฐานข้อมูล tempdb จะต่างจากฐานข้อมูลอื่น ตรงที่ตัวมันเองจะถูกสร้างใหม่ (rebuild) ทุกครั้ง เมื่อมีการเริ่มสตาร์ต SQL Server 2005

ฐานข้อมูลผู้ใช้งาน (User Databases)

ฐานข้อมูลผู้ใช้เป็นฐานข้อมูลที่ใช้งานสร้างขึ้นมา สำหรับจัดเก็บข้อมูลตามต้องการสำหรับฐานข้อมูลที่สร้างขึ้น โดยผู้ใช้งานนั้นสามารถสร้างได้โดยอิสระ เราสามารถสร้างฐานข้อมูลเก็บไว้ใน SQL Server 2005 ได้หลายฐานข้อมูล ขึ้นอยู่กับความต้องการของผู้ใช้

2.15 การเขียนโปรแกรมเพื่อทำงานกับฐานข้อมูล

การเขียนโปรแกรมด้วย Visual Basic เพื่อทำงานกับข้อมูลในฐานข้อมูลนั้น โดยภาพรวมก็คือการส่งคิวรี่ (query) ซึ่งหมายถึงคำสั่งในภาษา SQL (Structured Query Language) ไปประมวลผล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

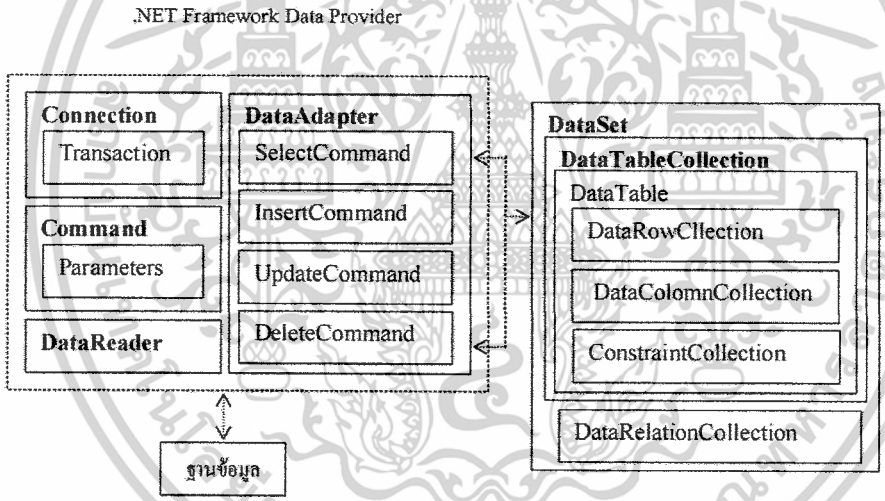
ยังฐานข้อมูล โดยคำสั่งที่ส่งไปอาจเป็นการร้องขอข้อมูล, แก้ไขข้อมูลเดิมที่มีอยู่, เพิ่มข้อมูล หรือลบข้อมูล

การทำงานกับฐานข้อมูลอีกวิธีหนึ่งที่นิยมทำกัน คือการเรียกไปยังชุดคำสั่งที่เราสร้างเก็บไว้ในตัวฐานข้อมูล ชุดคำสั่งเหล่านี้เรียกว่า Stored procedure ซึ่งภาษาที่ใช้เขียน stored procedure จะเป็นภาษาเฉพาะของฐานข้อมูลแต่ละประเภท

2.15.1 การจัดการฐานข้อมูลผ่าน Visual Basic 2008 โดยใช้ ADO.NET

ADO.NET คือเทคโนโลยีการเข้าถึงข้อมูล (data access technology) ของ .NET ที่ช่วยให้เราเขียนโปรแกรม เพื่อเข้าถึงและทำงานกับข้อมูลจากแหล่งข้อมูล (data source) ประเภทต่าง ๆ ได้อย่างเรียบง่ายแต่มีประสิทธิภาพ

ส่วนประกอบในการทำงาน ADO.NET แบ่งออกเป็น 2 ส่วนหลัก ๆ ได้แก่ DataSet Data Provider ดังรายละเอียดต่อไปนี้ (ดูรูปประกอบ)



ภาพที่ 2.55 โครงสร้างของ ADO.NET

2.15.2 DataSet

คือออบเจ็คซึ่งเป็นตัวแทนข้อมูลที่โปรแกรมของเราดึงมาจากฐานข้อมูล โดย DataSet จะเก็บสำเนา (copy) ของข้อมูลส่วนนั้นในฐานข้อมูลไว้ในหน่วยความจำของเครื่อง client (เครื่องที่โปรแกรมของเราทำงานอยู่) การทำงานต่าง ๆกับข้อมูลใน DataSet จึงเป็นอิสระจากฐานข้อมูล และเมื่อถึงเวลาที่เหมาะสมเราสามารถนำข้อมูลจาก DataSet ไปอัพเดทลงฐานข้อมูลได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทำงานกับฐานข้อมูลโดยใช้ออบเจ็ค DataSet จะเป็นรูปแบบที่เรียบกว่า disconnected data access หรือ "การเข้าถึงข้อมูลแบบไม่ต้องเชื่อต่อกับฐานข้อมูล (ไว้ตลอดเวลา)" และเรียกข้อมูลใน DataSet ว่า disconnected data เนื่องจากโปรแกรมจะมีการเชื่อมต่อกับฐานข้อมูลเท่าที่จำเป็นเท่านั้น เช่น ในการดึงข้อมูล โปรแกรมจะเปิดการเชื่อมต่อกับฐานข้อมูลเพื่อดึงข้อมูลเข้ามาแล้วปิดการเชื่อมต่อทันทีหลังจากดึงข้อมูลเสร็จ สำหรับการแก้ไข เพิ่ม หรือลบข้อมูลก็เช่นเดียวกัน โปรแกรมจะเชื่อมต่อไปยังฐานข้อมูลเพื่อรัน query ของคำสั่ง UPDATE, INSERT หรือ DELETE จากนั้นการเชื่อมต่อจะถูกปิดลงไป

2.15.3 Data Provider

คือส่วนที่ดูแลในเรื่องการเชื่อมต่อกับฐานข้อมูล และการส่ง Query ไปประมวลผลยังฐานข้อมูล ทั้งนี้ Data Provider ไม่ใช่ออบเจ็คเดี่ยวๆ เหมือนอย่าง DataSet หากแต่เป็นกลุ่มของออบเจ็คที่เกี่ยวข้องสัมพันธ์กัน ซึ่งร่วมกันทำงานเพื่อเป็นตัวกลางในการรับ-ส่งข้อมูลระหว่างฐานข้อมูล กับ DataSet โดย ADO.NET 2.0 ที่ติดตั้งมาพร้อมกับ Visual Studio 2008 ได้จัดเตรียม Data Provider ไว้ 4 ชนิดแยกตาม ประเภทของฐานข้อมูล ได้แก่

- **SQL Server Data Provider** (หรือ SQLClient Data Provider) เป็น Data Provider ที่ถูกออกแบบมาสำหรับฐานข้อมูล Microsoft SQL Server ตั้งแต่เวอร์ชั่น 7 ขึ้นไปโดยเฉพาะ
- **OLEDB Data Provider** เป็น Data Provider สำหรับแหล่งข้อมูลใดๆ ก็ตามที่สนับสนุนการติดต่อผ่านทางอินเทอร์เฟซ OLEDB เช่น ฐานข้อมูล Microsoft Access เป็นต้น
- **Oracle Data Provider** (หรือ OracleClient Data Provider) เป็น Data Provider ที่ถูกออกแบบมาสำหรับฐานข้อมูล Oracle โดยเฉพาะ (Data Provider นี้ถูกเพิ่มเข้ามาใน ADO.NET 1.1 ที่ติดตั้งมาพร้อมกับ Visual Studio .NET 2003
- **ODBC Data Provider** เป็น Data Provider สำหรับแหล่งข้อมูลใด ๆ ก็ตามที่มี ODBC driver ให้ใช้ในการติดต่อ (Data Provider นี้ถูกเพิ่มเข้ามาใน ADO.NET 1.1 ที่ติดตั้งมาพร้อมกับ Visual Studio .NET 2003)

Data Provider แต่ละชนิดข้างต้นจะประกอบไปด้วยออบเจ็คต่าง ๆ ทำงานร่วมกัน ดังนี้

- **ออบเจ็ค Connection** รับผิดชอบในเรื่องการเชื่อมต่อกับฐานข้อมูล
- **ออบเจ็ค Command** ทำหน้าที่ส่งคำสั่งไปประมวลผลยังฐานข้อมูล ซึ่งคำสั่งนั้นอาจเป็น query ในภาษา SQL หรือเป็นการเรียกใช้ stored procedure ในฐานข้อมูล เป็นต้น
- **ออบเจ็คเจ็ค DataAdapter** ทำหน้าที่ดึงข้อมูลจากฐานข้อมูลมาใส่ (fill) ลงใน DataSet และนำข้อมูลมาจาก DataSet ไปอัพเดทลงฐานข้อมูล
- **ออบเจ็ค DataReader** เป็นออบเจ็คที่ใช้ทำงานกับ result set ในลักษณะ forward-only (เลื่อนตำแหน่งเรคอร์ดไปข้างหน้าได้อย่างเดียว) และ read-only (อ่านข้อมูลได้อย่างเดียวแต่ไม่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เปลี่ยนแปลงแก้ไขไม่ได้) โดยระหว่างที่ใช้ออบเจ็คนี้ทำงานกับข้อมูลในฐานข้อมูล จะต้องเปิดการเชื่อมต่อกับฐานข้อมูลไว้ตลอดเวลา จึงเป็นรูปแบบที่เรียกว่า connected data access ซึ่งตรงกันข้ามกับการทำงานของ DataSet แต่มีข้อดีของ DataReader คือมันจะอ่านข้อมูลจากฐานข้อมูลเข้ามาในโปรแกรมได้รวดเร็วกว่า DataSet เนื่องจากเป็นการอ่านข้อมูลแบบไปข้างหน้าอย่างเดียว โดยไม่อนุญาตให้ย้อนกลับไปยังเรคอร์ดก่อนหน้าได้ DataReader จึงเหมาะสำหรับการอ่านข้อมูลเพื่อนำมาแสดงผลในโปรแกรมโดยไม่ให้ผู้ใช้แก้ไขข้อมูล หรือเพื่อนำข้อมูลมาสร้างรายงาน เป็นต้น

2.15.4 การเชื่อมต่อกับฐานข้อมูล

การเชื่อมต่อกับฐานข้อมูลเป็นหน้าที่ของออบเจ็ค Connection ดังที่อธิบายแล้ว คลาสที่ใช้สร้างออบเจ็ค Connection จะขึ้นอยู่กับชนิดของ Data Provider ที่ใช้ในการเชื่อมต่อฐานข้อมูล ซึ่งขึ้นอยู่กับประเภทของฐานข้อมูลที่ต้องการเชื่อมต่ออีกทีหนึ่ง

หลังจากสร้างออบเจ็คของคลาสที่เหมาะสมขึ้นมาแล้ว ให้กำหนดรายละเอียดการเชื่อมต่อกับฐานข้อมูลที่พร็อพเพอร์ตี้ ConnectionString ในที่นี้จะขอยกตัวอย่างการเชื่อมต่อกับฐานข้อมูลดังนี้

```
Public cn As New SqlConnection()
cn.ConnectionString = "Data Source=.\SQLExpress;InitialCatalog =
MODBUS;Integrated Security=True"
```

สำหรับบรรทัดที่สองคือการกำหนดค่าสตริงเรียกว่า Connection String ให้กับพร็อพเพอร์ตี้ ConnectionString ของออบเจ็ค Connection ที่สร้างขึ้น

เปิดและปิดการเชื่อมต่อ

ถึงแม้ว่าจะสร้างออบเจ็ค Connection และกำหนด connection string แล้วแต่การเชื่อมต่อกับฐานข้อมูลจะยังไม่เกิดขึ้น จนกว่าจะเรียกใช้เมธอด Open ของออบเจ็ค Connection เพื่อเปิดการเชื่อมต่อ ดังตัวอย่าง

```
cn.open()
```

หลังจากทำงานกับฐานข้อมูลเสร็จแล้ว ควรปิดการเชื่อมต่อกับฐานข้อมูล โดยเรียกเมธอด Close ของออบเจ็ค Connection ด้วยทุกครั้ง

```
cn.Close()
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ทำงานกับข้อมูลในฐานข้อมูลด้วยออบเจ็ค Command

ออบเจ็ค Command เป็นตัวแทนของคำสั่งที่ใช้งานกับฐานข้อมูล ซึ่งคำสั่งที่ว่านี้อาจเป็น query ของประโยคคำสั่ง SELECT, INSERT, UPDATE และ DELETE ในภาษา SQL หรือเป็นการเรียกไปยัง Stored procedure ในฐานข้อมูลก็ได้

การสร้างออบเจ็คของคลาส SqlCommand ทำได้โดยประกาศดังนี้

Dim com As New SqlCommand()

หลังจากสร้างออบเจ็คCommand ขึ้นมาแล้ว ต้องกำหนดค่าให้กับพร็อพเพอร์ตี้ต่าง ๆ ของมัน ได้แก่ พร็อพเพอร์ตี้ Connection, CommandText และCommandType ดังรายละเอียดต่อไปนี้

พร็อพเพอร์ตี้ Connection

คำสั่งในออบเจ็ค Command จะถูกส่งไปประมวลผลยังฐานข้อมูลผ่านทางการเชื่อมต่อที่ระบุไว้ที่ระบุไว้ที่พร็อพเพอร์ตี้ Connection นี้ ดังนั้นจะต้องนำออบเจ็ค Connection มากำหนดโดยต้องเป็นชนิดที่เหมาะสมด้วย

```
Dim cn As New SqlConnection()
Dim com As New SqlCommand()
cn.ConnectionString = "Data Source=.\SQLExpress;Initial
Catalog=MODBUS;Integrated Security=True"
com.Connection = cn
```

ทั้งนี้การเชื่อมต่อกับฐานข้อมูลต้องเปิดอยู่ในระหว่าง ที่เรียกใช้เมธอดของออบเจ็ค Command เพื่อส่งคำสั่งไปประมวลผลยังฐานข้อมูล มิฉะนั้นจะเกิดข้อผิดพลาดขึ้นในช่วง run time

พร็อพเพอร์ตี้ CommandText

ใช้กำหนดคำสั่งที่จะส่งไปประมวลผลยังฐานข้อมูล เช่นถ้าต้องการดึงข้อมูลจากฟิลด์ PV และ SV ในเทเบิล MODBUS_TEMP จะต้องกำหนดค่าสตริงที่เป็นคำสั่ง SQL ให้กับพร็อพเพอร์ตี้ CommandText ของออบเจ็ค Commnad ดังนี้

com.CommandText = “SELECT SV, PV FROM MODBUS_TEMP”

พร็อพเพอร์ตี้ CommandType

เป็นพร็อพเพอร์ตี้ที่ใช้กำหนดว่าค่าสตริงในพร็อพเพอร์ตี้ CommandTaxt คือคำสั่งแบบใด โดยค่าที่กำหนดได้คือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Text	กำหนดว่าค่าสตริงในพร็อพเพอร์ตี้ CommandText คือประโยคคำสั่งในภาษา SQL ซึ่งฐานข้อมูลจะประมวลผลคำสั่ง SQL นั้น ค่า Text นี้ เป็นค่าดีฟอลต์ของพร็อพเพอร์ตี้ CommandType ดังนั้นถ้าหากค่าสตริงที่คุณกำหนดไว้ที่พร็อพเพอร์ตี้ CommandText คือคำสั่ง SQL ก็ไม่จำเป็นต้องกำหนดพร็อพเพอร์ตี้ CommandType
StoredProcedure	กำหนดว่าค่าสตริงในพร็อพเพอร์ตี้ CommandText คือชื่อ stored procedure ในฐานข้อมูล ซึ่งจะทำ Stored procedure นั้นถูกประมวลผล
TableDirect	กำหนดว่าค่าสตริงในพร็อพเพอร์ตี้ CommandText คือชื่อเทเบิลในฐานข้อมูล ซึ่งออบเจ็ค Command จะให้ผลลัพธ์เป็น result set ของข้อมูลทุกฟิลด์และทุกเรคอร์ดในเทเบิลนั้น

2.15.5 การสร้างและใช้งาน DataAdapter

DataAdapter ทำหน้าที่เป็นตัวกลางในการรับ-ส่งข้อมูลระหว่างฐานข้อมูลกับ DataSet โดยในบางแอพพลิเคชั่นอาจใช้ DataAdapter เพียงทิศทางเดียวคือการดึงข้อมูลจากฐานข้อมูลมาเก็บไว้ใน DataSet เพื่อใช้งาน แต่ไม่ต้องการอัพเดทข้อมูลกลับลงไปยังฐานข้อมูล อย่างไรก็ตามสำหรับแอพพลิเคชั่นส่วนใหญ่แล้วจะใช้ DataAdapter ทั้งสองทิศทาง คือนอกจากจะดึงข้อมูลมาแล้วยังมีการส่งข้อมูลไปอัพเดทลงฐานข้อมูล ความเป็นจริงเมื่อคุณสั่งให้ DataAdapter ดึงข้อมูลจากฐานข้อมูลมาเก็บไว้ใน DataSet หรือนำข้อมูลจาก DataSet ไปอัพเดทลงฐานข้อมูล มันไม่ได้ส่ง query ไปประมวลผลยังฐานข้อมูลเอง แต่จะเรียกใช้ออบเจ็ค Command ที่ถูกกำหนดไว้ในพร็อพเพอร์ตี้ SelectCommand InsertCommand.DeleteCommand หรือ UpdateCommand การทำงานของ DataAdapter จึงต้องอาศัยออบเจ็ค Command อีกทีหนึ่ง เพียงแต่เราไม่ได้ติดต่อกับออกเจ็คเหล่านี้เองโดยตรง

สร้าง DataAdapter

DataAdapter ก็คือ ออบเจ็คของคลาส OleDbDataAdapter SqlDataAdapter Oracle DataAdapter หรือ OdbcDataAdapter ขึ้นอยู่กับชนิดของ Data Provider ที่ใช้ ตัวอย่างนี้แสดงการสร้าง DataAdapter ที่เป็นออบเจ็คของคลาส SqlDataAdapter

Dim da As New SqlDataAdapter

หลังจากสร้าง DataAdapter ขึ้นมาแล้ว ให้สร้างออบเจ็ค Command จำนวน 4 ออบเจ็ค เพื่อนำมากำหนดให้พร็อพเพอร์ตี้ SelectConnand InsertCommand DeleteCommand และ Update Command ของ DataAdapter ซึ่งออบเจ็ค command ทั้งสี่จะถูกใช้ในการดึงข้อมูล เพิ่มข้อมูล ลบข้อมูลและแก้ไขข้อมูลตามลำดับ โดยต้องกำหนดพร็อพเพอร์ตี้ เช่น Connection Command

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Text และ CommandType ของออบเจ็ค Command ทั้งสี่ไว้ให้เรียบร้อย (ซึ่งแน่นอนว่าต้องมีการสร้างออบเจ็ค Connection ขึ้นมาด้วย) โดยเฉพาะอย่างยิ่งคำสั่ง SQL ในพร็อพเพอร์ตี้ CommandText ที่ต้องสอดคล้องกันทั้งสี่ออบเจ็ค

พร็อพเพอร์ตี้ SelectCommand

พร็อพเพอร์ตี้ SelectCommand ของ DataAdapter ใช้เก็บออบเจ็ค Command ที่มีหน้าที่ดึงข้อมูลจากฐานข้อมูลมาใส่ลง DataSet ตัวอย่างนี้แสดงการสร้างออบเจ็ค Command เก็บไว้ในพร็อพเพอร์ตี้ SelectCommand ของ DataAdapter (สมมติ cn คือออบเจ็ค Connection ที่สร้างไว้แล้ว)

```
Dim strSQL As String = "SELECT * FROM MODBUS_TEMP"
Dim da As New SqlDataAdapter
da.SelectCommand = New SqlCommand(strSQL, cn)
```

เมื่อสร้างออบเจ็ค command เก็บไว้ในพร็อพเพอร์ตี้ SelectCommand ของ DataAdapter แล้ว คุณสามารถทำงานกับออบเจ็ค command นั้นโดยอ้างถึงพร็อพเพอร์ตี้ SelectCommand ได้เลย

เมธอด Fill

เมื่อถึงเวลาที่ต้องการดึงข้อมูลจากฐานข้อมูลมาเก็บไว้ใน DataSet จริง ๆ ให้คุณเรียกเมธอด Fill ของ DataAdapter เมธอดนี้จะเปิดการเชื่อมต่อกับฐานข้อมูล, ดึงข้อมูลมาเก็บลง DataSet แล้วปิดการเชื่อมต่อกับฐานข้อมูลทันทีที่ดึงข้อมูลเสร็จจะเห็นว่า DataAdapter ช่วยให้การดึงข้อมูลจากฐานข้อมูลมีความสะดวกอย่างยิ่ง

รูปแบบการใช้งานของเมธอด Fill คือ

รูปแบบ ⮛

Dataadapter*.Fill*(dataset, table)

Dataadapter	ออบเจ็ค DataAdapter
Dataset	ออบเจ็ค DataSet ที่ DataAdapter จะนำข้อมูลมาเก็บไว้
Table	ชื่อเทเบิลใน DataSet ที่ใช้เก็บข้อมูล ซึ่งถ้าใน DataSet ยังไม่มีเทเบิลชื่อนั้นก็จะเป็นการสร้างเทเบิลใน DataSet ขึ้นมาใหม่

การอ้างถึงฟิลด์ต่าง ๆ ใน DataSet เพื่อทำงานกับข้อมูลในฟิลด์ได้โดยใช้รูปแบบดังนี้

รูปแบบ ⮛

Dataset*.Tables *(table)*.Rows*(row)(field)

dataset	ออบเจ็ค DataSet
table	ชื่อหรือหมายเลขลำดับของเทเบิลใน DataSet

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

row	หมายเลขลำดับของเรคอร์ด
field	ชื่อหรือหมายเลขลำดับของฟิลด์

เมธอด Update

เมธอด Update ของ DataAdapter ใช้อัพเดทข้อมูลจาก DataSet ลงสู่ฐานข้อมูลโดย DataSet จะคอยจดจำว่ามีการเพิ่ม ลบและแก้ไขเรคอร์ดอะไรบ้าง เพื่อให้ DataAdapter สามารถอัพเดทฐานข้อมูลได้อย่างถูกต้อง

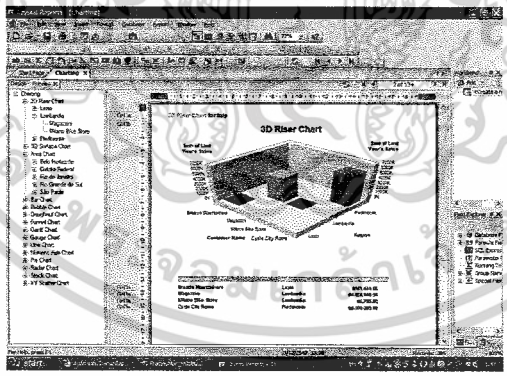
รูปแบบการใช้งานของเมธอด Update คือ

รูปแบบ

dataadapter.Update*(dateaset, table)*

dataadapter	ออบเจ็ค DataAdapter
dataset	ออบเจ็ค DataSet ที่ DataAdapter จะนำข้อมูลไปอัพเดทลงฐานข้อมูล
table	ชื่อเทเบิลใน DataSet ที่คุณต้องการให้นำข้อมูลไปอัพเดทฐานข้อมูล

2.16 โปรแกรมสร้างและแสดงรายงาน Crystal Report



ภาพที่ 2.56 แสดงหน้าตาโปรแกรม Crystal Report

Crystal Report เป็นโปรแรมการทำรายงานด้านสารสนเทศ หรือผู้ที่ต้องการเรียกใช้ข้อมูลจากฐานข้อมูลต่าง ๆ ได้อย่างรวดเร็วและโปรแกรมนี้ ยังมีรูปแบบรายงานให้เลือกได้หลายรูปแบบ เช่น รายงานย่อย , รายงานในรูปแบบตาราง,รายงานในรูปแบบรายละเอียด และยังสามารถนำเสนอข้อมูลในรูปแบบกราฟต่างๆ ได้หลายรูปแบบ เป็นต้น นอกจากนี้การใช้งานของโปรแกรมยังมี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เครื่องมือพิเศษที่ช่วยในการบอกวิธีการทำงาน และยังคอยแนะนำขั้นตอนในการสร้างรายงานให้ โดยที่ผู้ใช้แทบจะไม่มีความจำเป็นต้องเรียนรู้วิธีการมาก่อนอย่างละเอียด ก็สามารถสร้างรายงานง่าย ๆ ได้อย่างรวดเร็ว ความสามารถของ Crystal Report ไม่ใช่มีแต่เพียงแค่นี้เท่านั้น Crystal Report ยังสามารถดึงข้อมูลได้จากโปรแกรมจัดการฐานข้อมูลได้หลายประเภท อาทิเช่น Microsoft SQL Microsoft SQL Server , Foxpro , Visual Foxpro , Xbase Files , Paradox Files Btrieve dictionaries รวมทั้งการติดต่อจากฐานข้อมูลผ่าน ODBC อีกด้วย

Visual Studio 2005 Professional Edition และเอดิชันที่สูงถัดขึ้นไปมีมาพร้อมกับ Crystal Reports for Visual Studio 2005 ซึ่งเป็นผลิตภัณฑ์ที่จะช่วยสร้าง แสดงผล และพิมพ์รายงานในแอปพลิเคชัน Visual Basic

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3
การออกแบบและการสร้าง

ปริญญานิพนธ์นี้แบ่งการออกแบบและการสร้างออกเป็น 2 ส่วน คือ

1. **ส่วนฮาร์ดแวร์ (Hardware)** ได้แก่ การออกแบบและการสร้างกระบวนการทางอุตสาหกรรมจำลอง

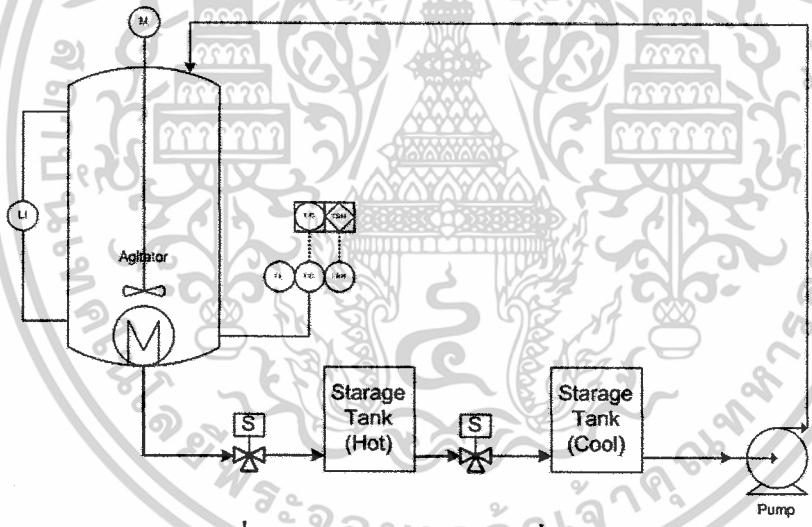
2. **ส่วนซอร์ฟแวร์ (Software)** ได้แก่

   2.1 การออกแบบและการพัฒนาโปรแกรม ควบคุม PLC

   2.2 การออกแบบและการพัฒนาโปรแกรมแสดงผลกระบวนการ

3.1 ส่วนฮาร์ดแวร์

3.1.1 การออกแบบและการสร้างกระบวนการจำลอง



ภาพที่ 3.1 กระบวนการจำลองที่จะสร้าง

จากภาพที่ 3.1 แสดงกระบวนการที่ได้ทำการสร้างในปริญญานิพนธ์นี้ ซึ่งประกอบด้วยอุปกรณ์ในระบบดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.1.1 ถังควบคุมอุณหภูมิ มีหน้าที่ในการควบคุมอุณหภูมิน้ำในถังให้มีอุณหภูมิตามต้องการ ประกอบด้วย

- ท่อใสแสดงระดับของเหลว (Glass Gauge) ใช้สำหรับแสดงระดับของเหลวในถังควบคุมอุณหภูมิ
- มอเตอร์กวน (Agitator) มีหน้าที่ในการกวนน้ำให้มีอุณหภูมิเท่ากันทั้งถัง
- ฮีตเตอร์ (Heater) มีหน้าที่ในการให้ความร้อนกับระบบ
- อุปกรณ์ควบคุมอุณหภูมิ (Temperature Controller) มีหน้าที่ในการควบคุมอุณหภูมิของระบบ
- โซลิดสเตทรีเลย์ (Solid state relay) มีหน้าที่ในการรับค่าอินพุตจากอุปกรณ์ควบคุมอุณหภูมิและควบคุมความร้อนของฮีตเตอร์
- อุปกรณ์วัดอุณหภูมิภายในถังควบคุมอุณหภูมิ มีหน้าที่ในการวัดอุณหภูมิภายในถังควบคุมและส่งค่าไปยังอุปกรณ์ควบคุมอุณหภูมิ
- เทอร์โมสตัท (Thermostat) มีหน้าที่ในการป้องกันความเสียหายของฮีตเตอร์ในกรณีที่เกิดความร้อนสูงเกินกว่าค่าที่ได้ตั้งไว้

3.1.1.2 ถังพักน้ำ (ถังร้อน) มีหน้าที่ในการพักน้ำที่มีอุณหภูมิสูงก่อนที่จะปล่อยกลับเข้าสู่ถังพักน้ำ(ถังเย็น)

3.1.1.3 ถังพักน้ำ (ถังเย็น) มีหน้าที่ในการพักน้ำที่มีอุณหภูมิต่ำที่จะสูบน้ำเข้าสู่ถังควบคุมอุณหภูมิ

3.1.1.4 ปั๊ม มีหน้าที่ในการสูบน้ำเข้าสู่ถังควบคุมอุณหภูมิ

3.1.1.5 โซลินอยด์วาล์ว (Solenoid Valve) มีหน้าที่ในการเปิดให้น้ำไหลจากถังพักน้ำ (ถังเย็น) เข้าสู่ถังควบคุมอุณหภูมิและเปิดให้น้ำไหลจากถังควบคุมอุณหภูมิเข้าสู่ถังพักน้ำ (ถังร้อน)

3.1.1.6 PLC มีหน้าที่ในการควบคุมการทำงานของระบบทั้งหมด

3.1.1.7 อุปกรณ์แปลงสัญญาณ RS-232 to RS-485 Converter มีหน้าที่ในการแปลงสัญญาณจาก RS-232 เป็น RS-485 เพื่อส่งเข้าไปยังคอมพิวเตอร์ให้สามารถประมวลผลข้อมูลได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.2 การออกแบบและการคำนวณปริมาตรของถัง

ถังควบคุมอุณหภูมิของเหลวปริมาตร 20 ลิตร ต้องการการแสดงผลของระดับที่ 30 cm ถังกำหนดความสูง 40cm

จากสมการปริมาตรของทรงกระบอก

$$V = \pi r^2 h$$

แทนค่า V = 20 ลิตร, h = 40 cm จะได้

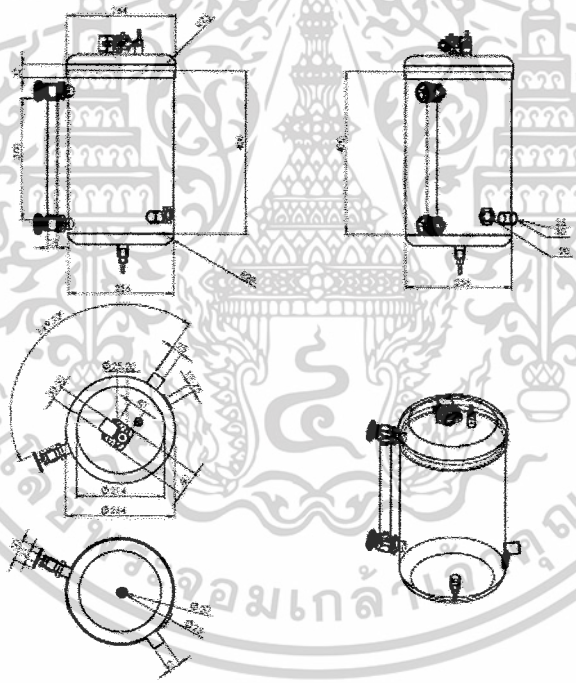
$$20{,}000 = \pi r^2 (40)$$

$$r = \sqrt{\frac{20{,}000}{40 \times 3.14}}$$

$$r = 12.61 \text{ cm}$$

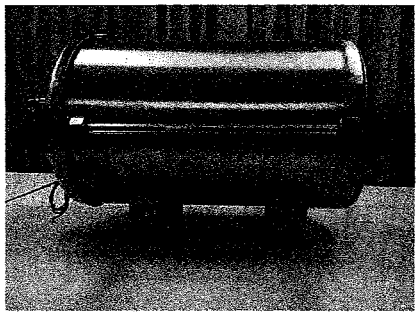
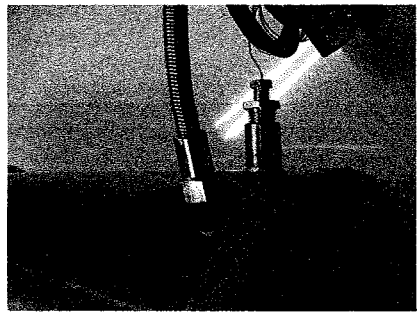
$$r \approx 12.5 \text{ cm}$$

$$D = 25 \text{ cm}$$



ภาพที่ 3.2 แสดงถังที่ออกแบบโดยใช้โปรแกรม SolidWORK

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาพที่ 3.3 แสดงถังควบคุมอุณหภูมิที่ได้จัดสร้าง

3.1.3 การคำนวณหาขนาดของอุปกรณ์ทำความร้อน(Heater)

จากสูตร $Q = mC_y(T_2 - T_1)$

เมื่อ

m = ปริมาตรของน้ำ 20 ลิตร หรือประมาณ 20 kg

C_y = ค่าความจุความร้อนเฉพาะของน้ำเท่ากับ 4.184 ($kJ/kg.K$)

T_1 = อุณหภูมิเริ่มต้น 27 องศาเซลเซียส หรือเท่ากับ 300.15 K

T_2 = อุณหภูมิสุดท้าย 70 องศาเซลเซียส หรือเท่ากับ 343.15 K

แทนค่าลงในสมการที่ 3.2

$$Q = 20(kg) \times 4.184(kJ/kg.K) \times (343.15 - 300.15)K$$
$$= 3{,}598.24\,kJ$$

กำหนดเวลาในการทำความร้อน 45 นาที หรือเท่ากับ 2400 วินาที

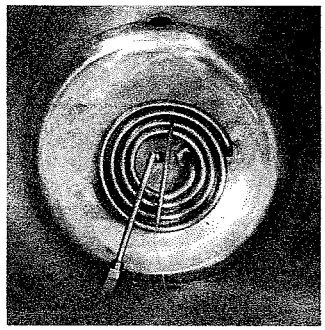
จากสมการ $P = Q/T$

จะได้ $P = \frac{3598.24}{2400} kJ/S$

$= 1.499\ kW$

ดังนั้นในโครงงานนี้จะเลือกใช้ Heater 1600 W

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



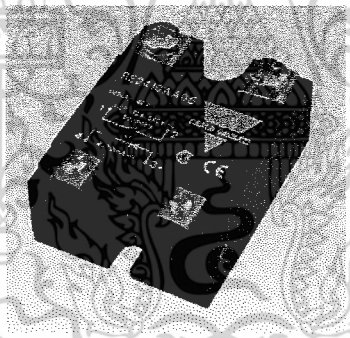
ภาพที่ 3.4 แสดงฮีตเตอร์ที่ติดตั้งในถังควบคุมอุณภูมิ

3.1.4 การหาค่ากระแสของ Solid State Relay

$$I = \frac{\text{watt} \times \text{factor}}{\text{volt}}$$

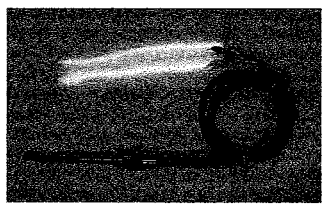
$$I = \frac{1{,}600 \times 1.3}{220} = 9.45\ \text{A}$$

$$\text{SSR} \approx 10\ \text{A}$$



ภาพที่ 3.5 แสดงโซลิดสเตทรีเลย์ CARLO GAVAZZI แบบ Analog switching ที่เลือกใช้

3.1.5 อุปกรณ์วัดอุณหภูมิ RTD

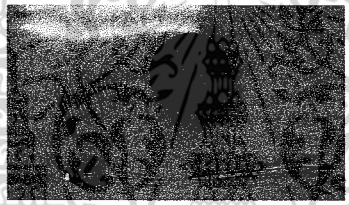


ภาพที่ 3.6 แสดงRTD PT-100 ที่เลือกใช้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ชนิดของไส้ (Element)	PT100 Ω
ขนาดของไส้	Φ 0.65 mm.
ขนาดของปลอก	สแตนเลส SUS 316
ความยาวของปลอก	300 mm.
ความยาวสาย	2 m.
ขนาดเกลียว	3/4 "
อุณหภูมิที่ใช้	PT100: 600 ^{o} C
ลักษณะการใช้งาน	สำหรับวัดอุณหภูมิอากาศ และของเหลวติดตั้งกับผนัง โดยต๊าฟเกลียวที่ผนัง (3A)

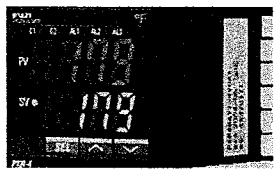
3.1.6 เทอร์โมสตัท (Thermostat)



ภาพที่ 3.7 แสดงเทอร์โมสตัทที่เลือกใช้

ช่วงอุณหภูมิ	0 ^{o} C ถึง 120 ^{o} C
ขนาดของก้านวัด	Φ 6 mm × 120 mm
ความแม่นยำ (Accuracy)	± 5 ^{o} C
ช่วงตัดท่อ (Hysteresis)	4 ± 2 ^{o} C
เอาต์พุต	SPDT 16A/200 V สายเป็นสแตนเลส ยาว 1 เมตร

3.1.7 อุปกรณ์ควบคุมอุณหภูมิ (Temperature Controller)



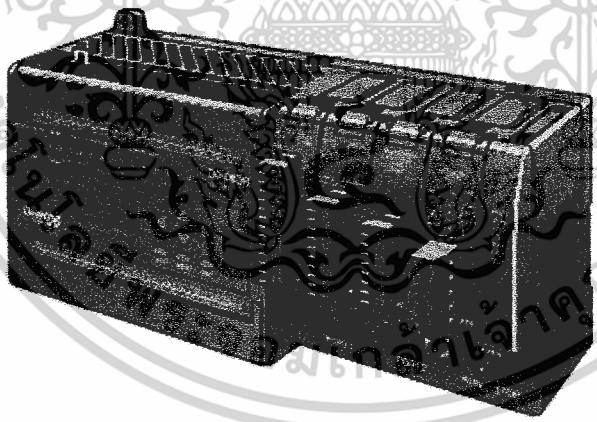
ภาพที่ 3.8 แสดงอุปกรณ์ควบคุมอุณหภูมิ FUJI PXR-4 ที่เลือกใช้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แหล่งจ่าย	100 (-15%) to 240V,(+10%) AC, 50/60Hz
สัญญาณอินพุต	RTD: PT100 ohm, 3-wire type (C)
ลักษณะการควบคุม	PID control (with auto-tuning, self-tuning)
เอาต์พุตควบคุม	4 to 20 mA DC: allowable load resistance 600 ohm or less
มาตรฐานการติดต่อสื่อสาร	EIA RS485
โปรโตคอล	MODBUS (RTU)
วิธีการติดต่อสื่อสาร	2-wire method, Half-duplex bit serial, start-stop sync type
ชนิดข้อมูล	8 bits. Parity: odd/even/none
อัตราการรับส่งข้อมูล	9600 bps
ความสามารถในการเชื่อมต่อ	Multi-drop up to 31 controllers
ระยะทางในการติดต่อสื่อสร	Total extension 500m or less

3.1.8 Programmable Logic Controller (PLC)

ปริญญานิพนธ์นี้เลือกใช้ PLC Koyo D0-06DD1 ซึ่งมีรายละเอียดดังนี้

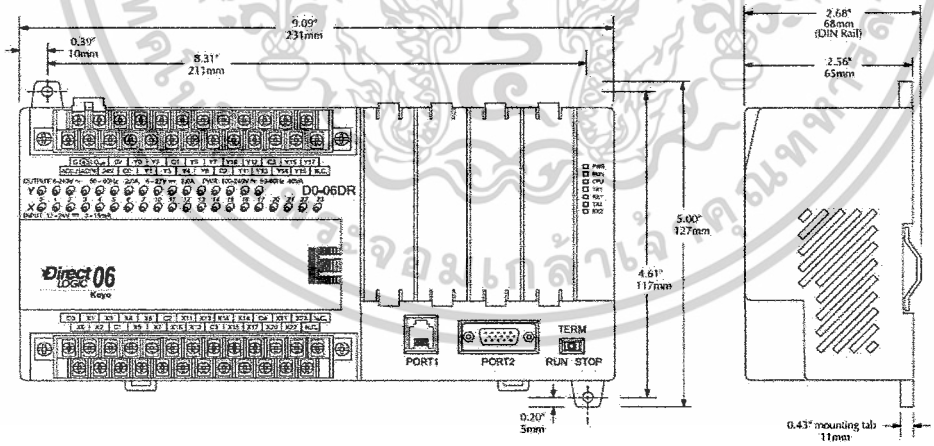


ภาพที่ 3.9 PLC DirectSOFT KOYO D0-06DDI

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

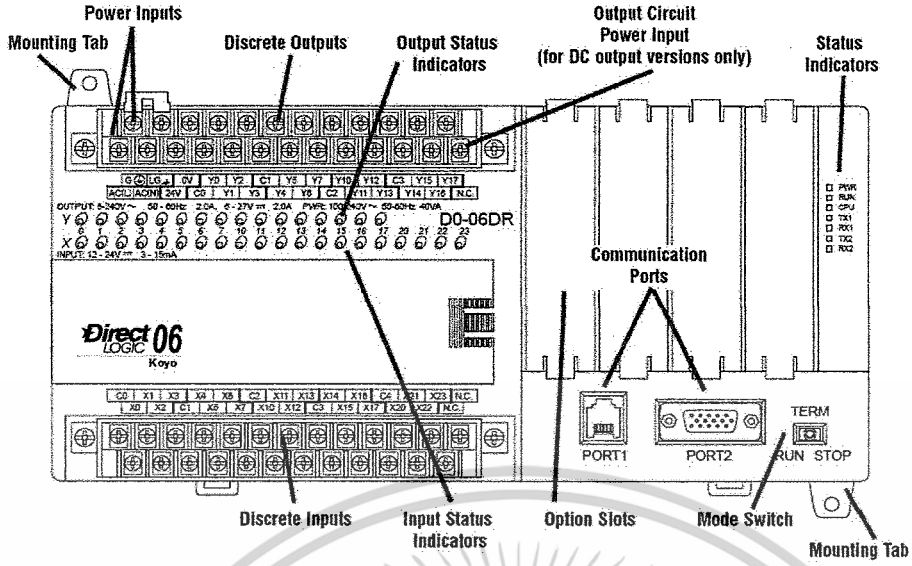
Specifications	
Feature	**DL06**
Total Program memory (words)	14.8K
Ladder memory (words)	7680
Total V-memory (words)	7616
User V-memory (words)	7488
Non-volatile V Memory (words)	128
Contact execution (boolean)	2.0uS
Typical scan (1k boolean)	3 - 4mS
RLL Ladder style Programming	Yes
RLL and RLLPLUS Programming	Yes
Run Time Edits	Yes
Scan	Variable / fixed
Handheld programmer	Yes
*Direct*SOFT32 programming for Windows.	Yes
Built-in communication ports (RS232C)	Yes
FLASH Memory	Standard on CPU
Local Discrete I/O points available	36
Local Analog input / output channels maximum	None
High-Speed I/O (quad., pulse out, interrupt, pulse catch, etc.)	Yes, 2
I/O Point Density	20 inputs, 16 outputs
Number of instructions available (see Chapter 5 for details)	229
Control relays	1024
Special relays (system defined)	512
Stages in RLLPLUS	1024
Timers	256
Counters	128
Immediate I/O	Yes
Interrupt input (external / timed)	Yes
Subroutines	Yes
For/Next Loops	Yes
Math (integer and floating point)	Yes
Drum Sequencer instruction	Yes
Time of Day Clock/Calendar	Yes
Internal diagnostics	Yes
Password security	Yes
System error log	Yes
User error log	Yes
Battery backup	Optional D2-BAT-1 available (not included with unit)

ภาพที่ 3.10 แสดง Specification ของ PLC KOYO D0-6DD1



ภาพที่ 3.11 แสดง Dimensions ของ PLC KOYO D0-6DD1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 3.12 แสดงโครงสร้างของ PLC KOYO D0-6DD1

3.1.8.1 โครงสร้างของ PLC KOYO D0-6DD1 ประกอบด้วย

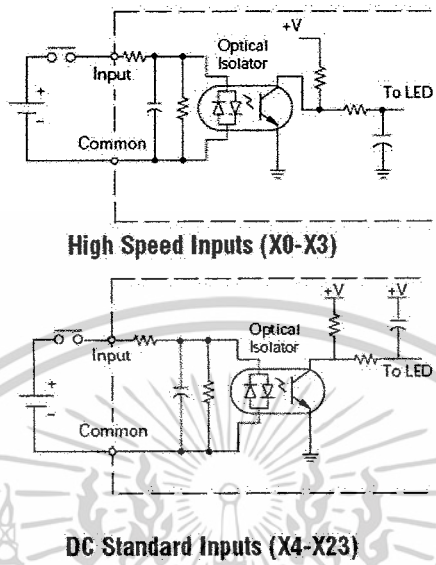
- Mounting tab ใช้สำหรับเป็นจุดยึดกับงานหรือที่ตั้งอุปกรณ์
- Mode switch เป็นตัวเลือกโหมดการทำงาน
- Communication Port เป็นตัวเชื่อมระหว่าง PLC กับอุปกรณ์ภายนอกเพื่อรับส่งข้อมูล
- Output Status Indicators ตัวแสดงการทำงานของ อินพุต
- Status Indicators ตัวแสดงสถานะการทำงานของPLC
- Output Circuit Power Input
- Discrete Input Terminal จุดต่ออินพุต
- Discrete Output Terminal จุดต่อเอาท์พุต
- External Power Inputs จุดต่อแหล่งจ่ายไฟ
- Input Status Indicators ตัวแสดงการทำงานของเอาท์พุต

3.1.8.2 ภาคอินพุต

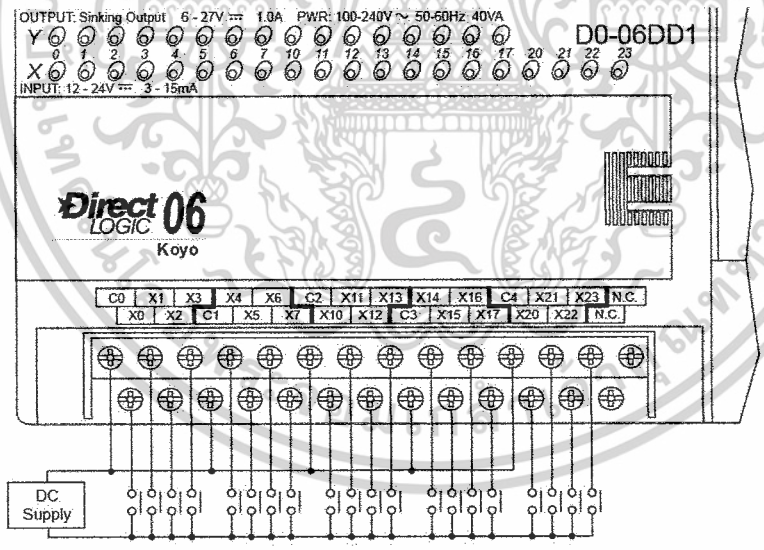
อินพุตจะเป็นตัวที่รับสัญญาณมาจากการใช้งานต่าง ๆ ไม่ว่าจะเป็น Push buttons Selector switches Motor startercontect Thumb wheels Limit switch เมื่ออินพุตได้รับสัญญาณเข้ามาจะทำการเปลี่ยนแปลงระดับแรงดันที่รับให้อยู่ในระดับสัญญาณลอจิก ที่ CPU สามารถที่จะเข้าใจได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ใน PLC รุ่นที่ใช้นี้จะมี Input 20 ช่อง เพื่อนำเงื่อนไขจากภายนอกเข้ามายังตัวของ PLC เอง โดยจะถูกกำหนดให้ใช้เป็น X0-X23 เพื่อให้สามารถนำไปใช้ในการเขียนในโปรแกรม



ภาพที่ 3.13 แสดงวงจรภายในของภาคอินพุต



ภาพที่ 3.14 แสดงการต่อ Input

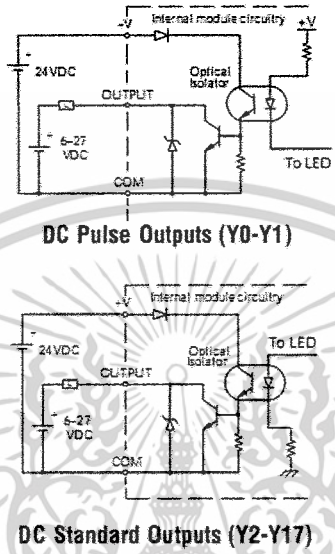
3.1.8.3 ภาคเอาท์พุต

เอาท์พุตจะเป็นตัวจ่ายสัญญาณได้จากการประมวลผลของ CPU เพื่อที่จะนำสัญญาณที่ได้นี้ไปควบคุมกระบวนการ โดยจะออกไปเป็นแรงดันไปยังเครื่องมือทางด้านสวิตช์ชิ่ง

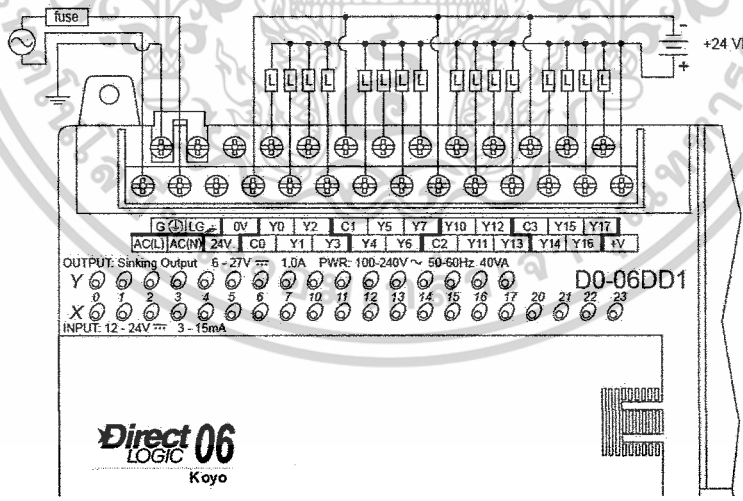
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เพื่อให้สวิตช์ซึ่งเป็นสื่อกลางในการควบคุมกระบวนการ ใน PLC รุ่นที่ใช้นี้ จะมีเอาท์พุต16 ช่อง โดยจะถูกกำหนดให้ใช้เป็น Y0-Y17

ในการต่อ Output นั้น หน้าสัมผัสจะถูกออกแบบให้สามารถรับแรงดันได้ 6-27 VAC VDC 1.0A ซึ่งจะต้องตรวจสอบให้ถูกต้องด้วยก่อนใช้งานอาจจะมีผลเสียแก่ตัว PLCได้



ภาพที่ 3.15 แสดงวงจรภายในของภาคเอาต์พุต



ภาพที่ 3.16 แสดงจุดต่อ Output

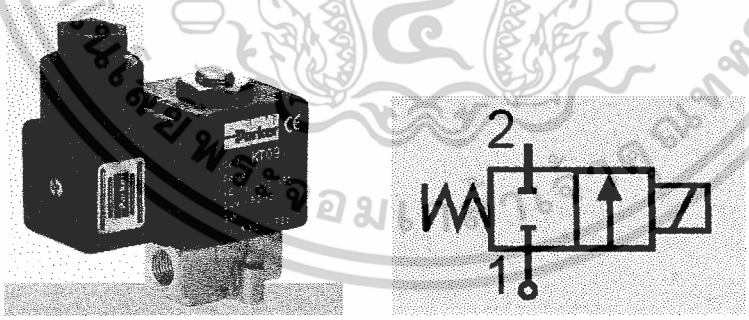
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.9 อุปกรณ์แปลงสัญญาณ RS-232 to RS-485 Converter

ภาพที่ 3.17 แสดงRS-232 to RS-485 Converter รุ่น TNA10A ที่เลือกใช้

มาตรฐานการสื่อสารที่รองรับ	RS-422 และ RS-485
บอดเรท	300 – 38400 bps
การเชื่อมต่ออุปกรณ์	สูงสุด 247 ตัว
แหล่งจ่ายไฟ	90 – 240 VAC

3.1.10 โซลินอยด์วาล์ว (Solenoid Valve)



ภาพที่ 3.18 แสดงโซลินอยด์วาล์ว Parker รุ่น VE131.4 FV และ สัญลักษณ์

ลักษณะคอยล์	ปกติปิด (N.C.)
ลักษณะการทำงาน	Direct operate

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2 ส่วนซอฟท์แวร์ (Software)

3.2.1 การออกแบบและการพัฒนาโปรแกรมควบคุม PLC

การออกแบบจะต้องทำการกำหนด I/O และหน่วยความจำ ว่าจะใช้งานอย่างไรซึ่งขึ้นอยู่กับการออกแบบและลักษณะการนำไปใช้งานในปริญญานิพนธ์นี้ จะใช้ PLC ของ KOYO DL06 โดยทำการกำหนดการแบ่ง I/O และหน่วยความจำดังนี้

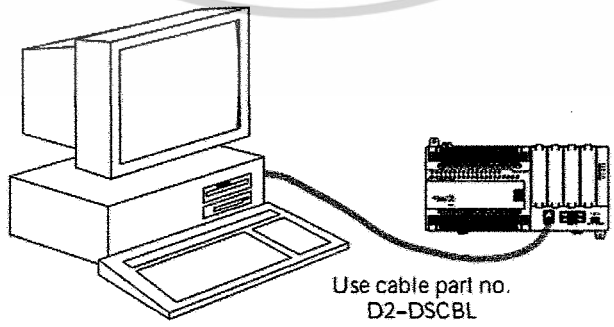
3.2.1.1 PLC I/O Assignment

ตารางที่ 3.1 PLC I/O Assignment

อินพุท- เอาท์พุท	รายการอุปกรณ์และการใช้งาน	หมายเหตุ
X0	Manual Pump	สั่งของเหลวไปใน Tank 3
X1	Manual Motor	สั่งมอเตอร์กวนของเหลวใน Tank 1
X2	Manual Solenoid Valve 1	สั่งให้ปล่อยของเหลวใน Tank1
X3	Manual Solenoid Valve 2	สั่งให้ปล่อยของเหลวใน Tank2
Y0	Pump	สูบของเหลวไปใน Tank 3
Y1	Motor	มอเตอร์กวนของเหลวใน Tank 1
Y2	Solenoid Valve 1	ปล่อยของเหลวใน Tank 1
Y3	Solenoid Valve 2	ปล่อยของเหลวใน Tank 2

3.2.1.2 โปรแกรม DirectSOFT32

ในการเขียนโปรแกรมใน PLC KOYO DO-06DD1นั้น จะใช้โปรแกรม Direct-SOFT32 ในการเขียนLadder ซึ่งเป็นโปรแกรมควบคุมการทำงานของ PLC

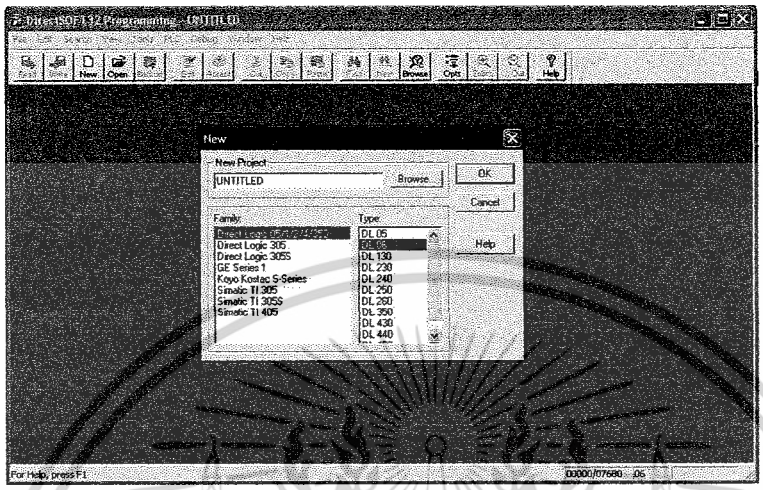


ภาพที่ 3.19 การเชื่อมต่อระหว่าง PLC KOYO DO-06DD1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

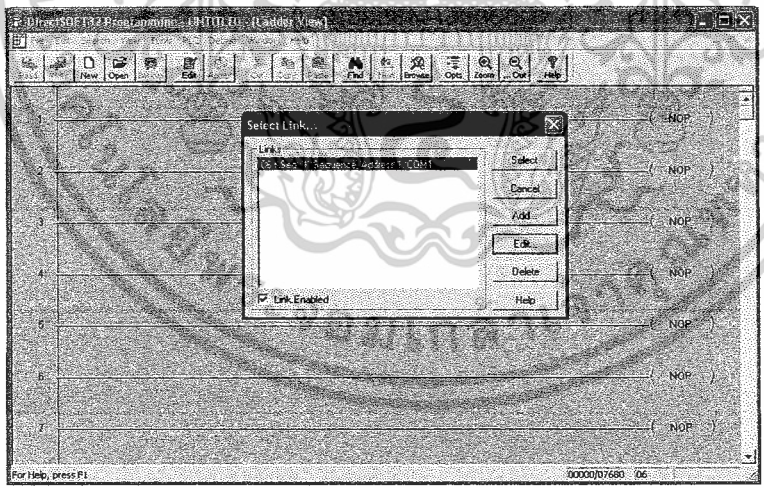
3.2.1.3 การใช้งานโปรแกรม DirectSOFT32

1. เปิดโปรแกรม DirectSOFT32 จะปรากฏหน้าต่างดังภาพ เลือกรุ่น PLC ที่ต้องการเชื่อมต่อ



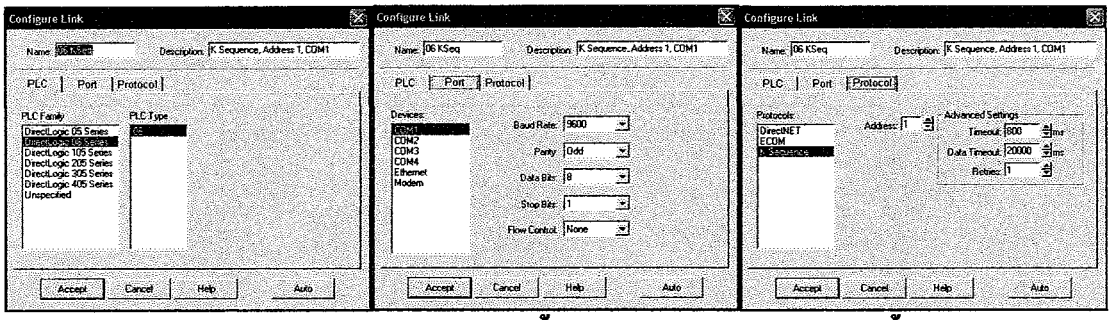
ภาพที่ 3.20 หน้าจอเริ่มต้นโปรแกรม DirectSOFT32

2. เลือกตัวเชื่อมต่อกับ PLC ที่ต้องการ หากต้องการแก้ไข เลือกปุ่ม Edit



ภาพที่ 3.21 หน้าจอโปรแกรม DirectSOFT32

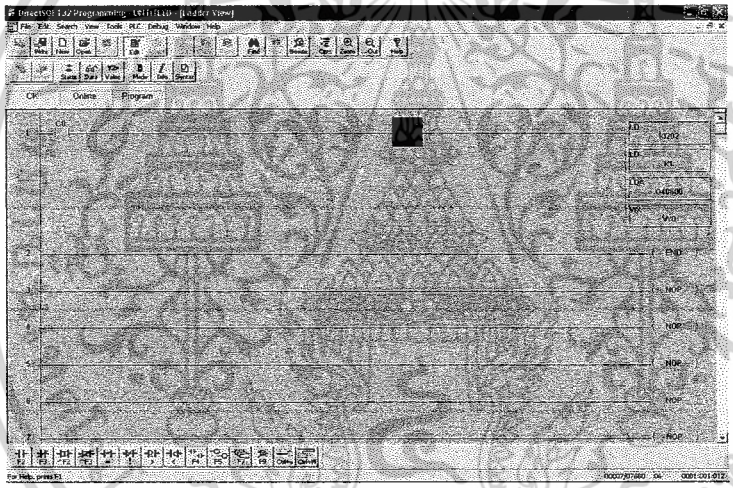
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เลือกรุ่น PLC **เลือกและตั้งค่า Port** **เลือกและตั้งค่าโปรโตคอล**

ภาพที่ 3.22 หน้าจอตั้งค่าการติดต่อสื่อสาร

3. เข้าสู่หน้าจอของโปรแกรม DirectSOFT32 ซึ่งใช้ในการพัฒนาโปรแกรมควบคุมของ PLC DO-06DD1



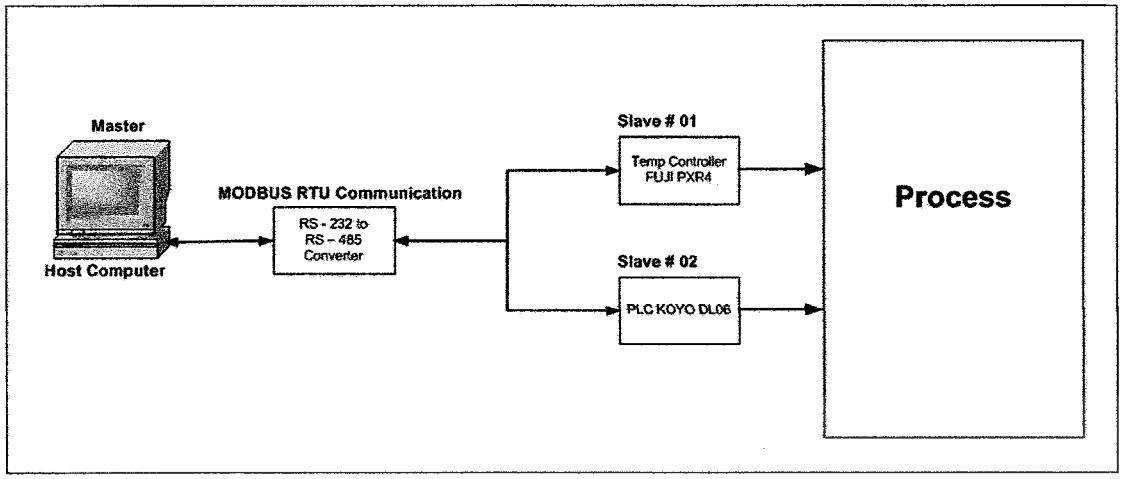
ภาพที่ 3.23 หน้าจอโปรแกรม DirectSOFT32

3.2.2 การออกแบบและการพัฒนาโปรแกรมแสดงผลกระบวนการ

3.2.2.1 การออกแบบระบบการติดต่อสื่อสาร

จากรูปเป็นไดอะแกรมแสดงระบบที่จะทำการสร้าง ประกอบด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 3.24 ไดอะแกรมแสดงระบบที่ใช้

1. โฮสต์คอมพิวเตอร์ เป็นอุปกรณ์มาสเตอร์ในระบบ ทำหน้าที่ควบคุมการรับส่งข้อมูลด้วยโปรโตคอล MODBUS และแสดงผลกระบวนการที่หน้าจอคอมพิวเตอร์

2. RS-232 to RS-485 Converter ทำหน้าที่แปลงข้อมูลระหว่าง RS-485 เป็น RS-232 เพื่อให้โฮสต์คอมพิวเตอร์กับอุปกรณ์ควบคุมที่มีการติดต่อสื่อสารด้วยโปรโตคอล MODBUS สามารถสื่อสารกันได้

3. อุปกรณ์ควบคุมที่มีการติดต่อสื่อสารด้วยโปรโตคอล MODBUS ทำหน้าที่ควบคุมกระบวนการจำลองและเป็นสเลฟในระบบ ประกอบด้วย

- สเลฟหมายเลข 1 อุปกรณ์ควบคุมอุณหภูมิ FUJI PXR4
- สเลฟหมายเลข 2 อุปกรณ์ควบคุมแบบตรรกะที่สามารถโปรแกรมได้ (PLC) KOYO DO-06DD1

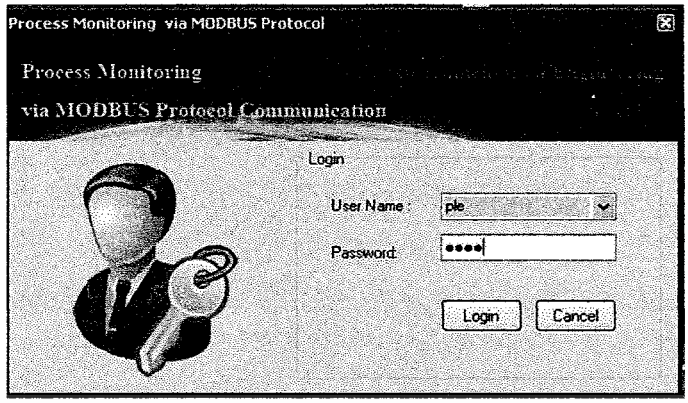
3.2.2.2 การออกแบบ User Interface และ ฟังก์ชันการใช้งานของโปรแกรมแสดงผลกระบวนการ

โปรแกรมที่สร้างจะมีหน้าแสดงการทำงานทั้งหมด 8 หน้า ดังนี้

1. หน้า Login ใช้สำหรับการลงชื่อเข้างาน โดยแยกระดับสิทธิผู้ใช้งานเป็น 2 ระดับ คือ

- Supervisor มีสิทธิในการเข้าถึงการใช้งานทุกฟังก์ชันของโปรแกรม
- Technician มีสิทธิในการเข้าถึงเฉพาะส่วนการดูระบบและควบคุมกระบวนการจากคอมพิวเตอร์เท่านั้น

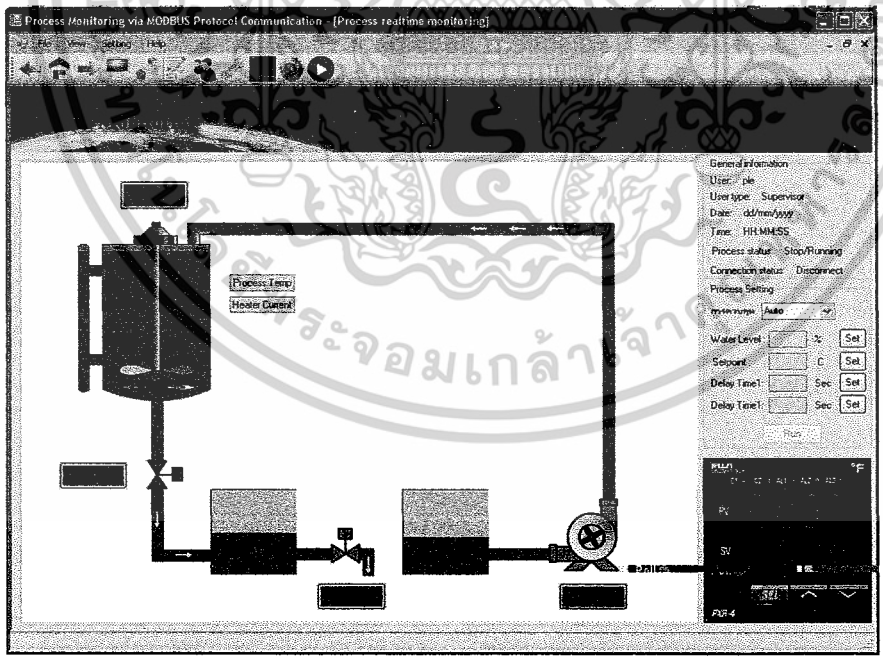
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 3.25 แสดงหน้าจอ Login

2. หน้า Process real-time monitoring มีฟังก์ชันการใช้งานคือ

- การแสดงผลกระบวนการแบบในลักษณะภาพเคลื่อนไหวเสมือนจริง
- การควบคุมกระบวนการโดยตรงจากคอมพิวเตอร์
- การตั้งค่าตัวแปรที่สำคัญในกระบวนการ
- การแสดงค่าตัวแปรที่สำคัญจากระบบ

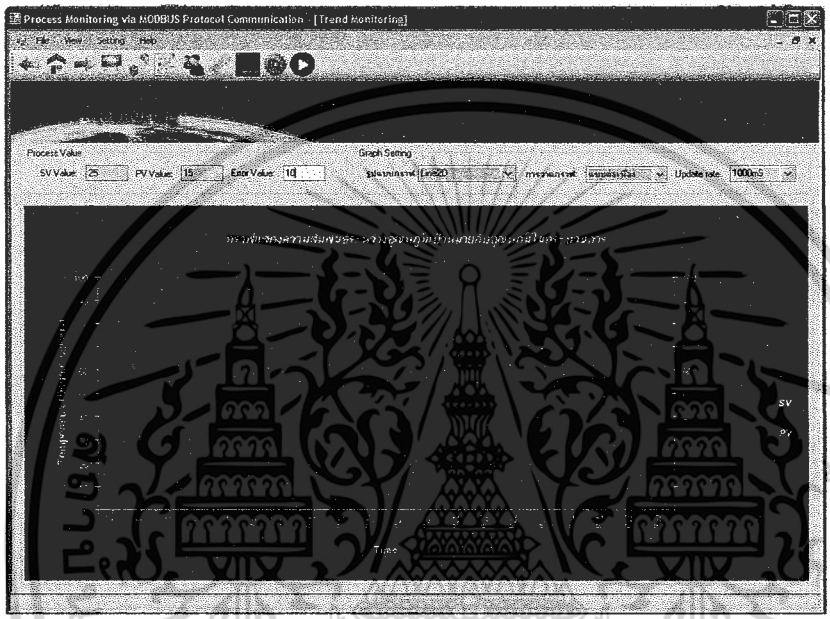


ภาพที่ 3.26 หน้าจอ Process monitoring

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. หน้า Trend Monitoring มีฟังก์ชันการใช้งานคือ

- แสดงค่าตัวแปรในกระบวนการที่สนใจด้วยกราฟในรูปแบบต่างๆ
- แสดงกราฟในสองรูปแบบคือแบบสะสมค่าและแบบแสดงเฉพาะค่าในบางช่วง
- สามารถปรับอัตราการอัพเดทกราฟได้หลายระดับ

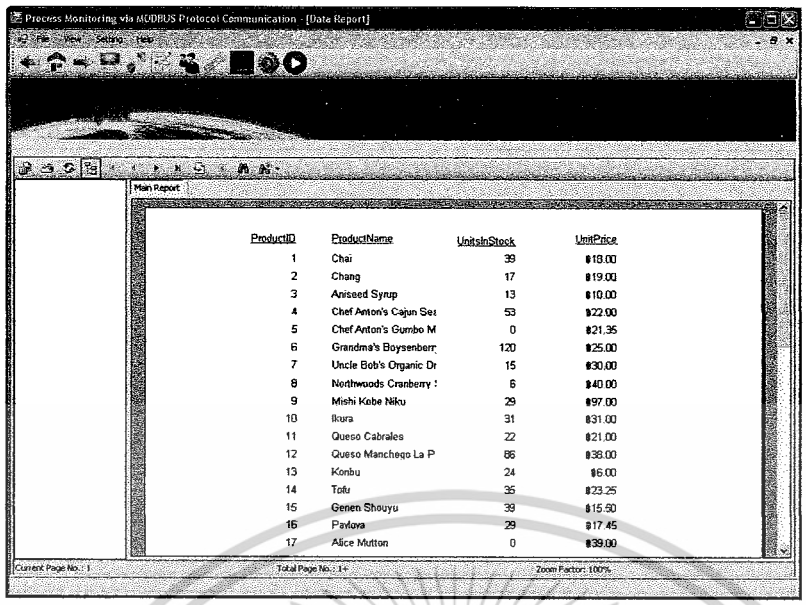


ภาพที่ 3.27 หน้าจอ Trend monitoring

4. หน้า Data Report มีฟังก์ชันการใช้งานคือ

- แสดงรายงานข้อมูลจากระบบฐานข้อมูลโดยสามารถเลือกดูข้อมูลย้อนหลังได้
- สามารถบันทึกข้อมูลรายงานในไฟล์รูปแบบต่างๆได้คือ Microsoft WORD, Microsoft Excel, Adobe PDF, Text Files,

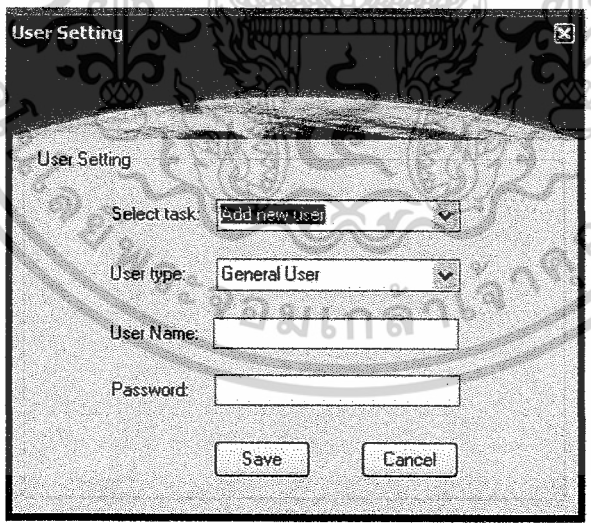
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 3.28 หน้าจอ Data Report

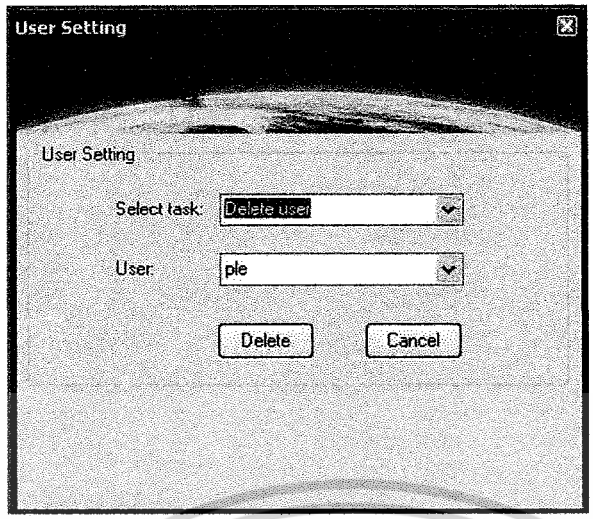
5. หน้าแสดง User setting มีฟังก์ชันการใช้งานคือ

- การเพิ่มชื่อผู้ใช้งาน รหัสผ่าน และประเภทผู้ใช้งานใหม่เข้าไปในระบบ
- การลบชื่อผู้ใช้งาน รหัสผ่าน และประเภทผู้ใช้งานออกจากระบบ



ภาพที่ 3.29 หน้าจอ User Setting (Add new user)

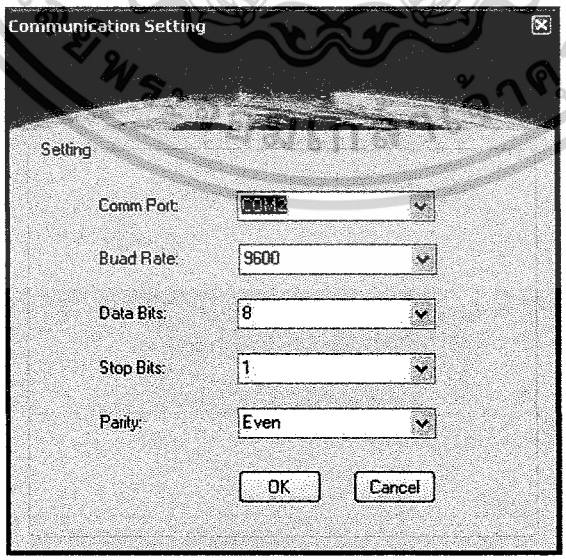
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 3.30 หน้าจอ User Setting (Delete user)

6. หน้าแสดง Communication Setting มีฟังก์ชันการใช้งานคือ ตั้งค่าการติดต่อสื่อสารผ่านพอร์ทอนุกรม โดยสามารถตั้งค่าต่อไปนี้ได้

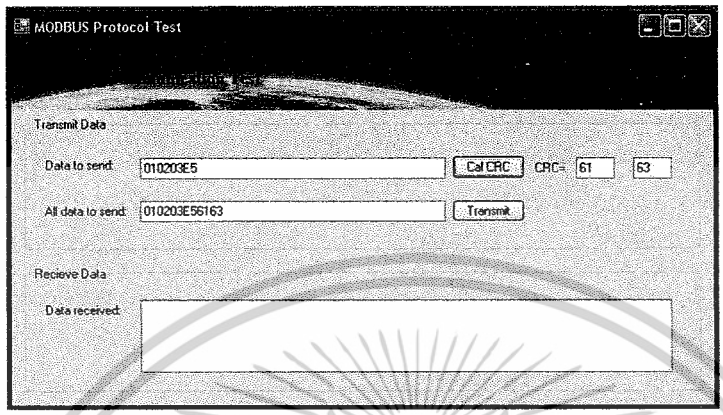
- Com Port
- Baud rate
- Data bits
- Stop bits
- Parity bits



ภาพที่ 3.31 หน้าจอ Communication Setting

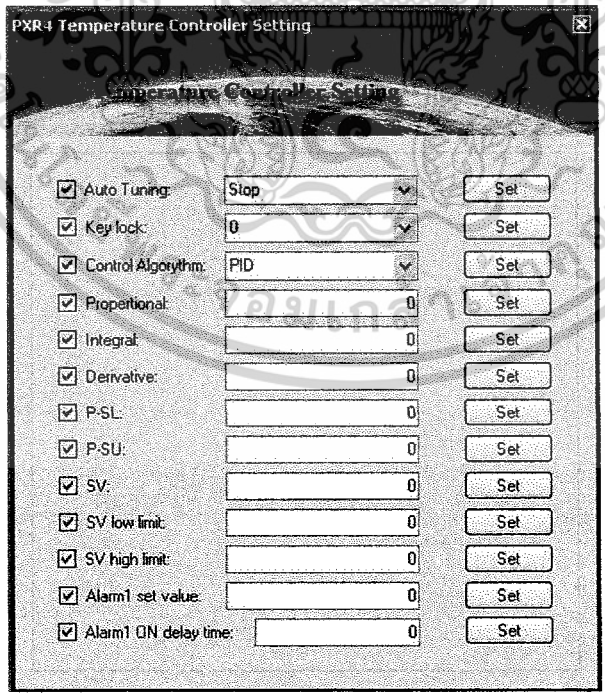
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

7. หน้าแสดง MODBUS Protocol Test มีฟังก์ชันการใช้งานคือ
   - การทดสอบการรับ-ส่ง ค่าข้อมูลด้วยโปรโตคอล MODBUS
   - การคำนวณชุดตรวจสอบความผิดพลาดของข้อมูล CRC



ภาพที่ 3.32 หน้าจอ MODBUS Protocol Test

8. หน้าแสดง PXR4 Setting มีฟังก์ชันการใช้งานคือ
   - การตั้งค่าตัวแปรที่สำคัญไปยัง PXR4 Temperature Controller
   - การอ่านค่าตัวแปรที่สำคัญจาก PXR4 Temperature Controller



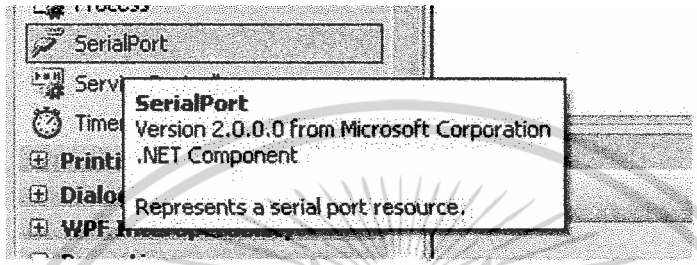
ภาพที่ 3.33 หน้าจอ PXR4 Temperature Controller Setting

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.3 การติดต่อสื่อสารกับอุปกรณ์ฮาร์ดแวร์ผ่าน Serial Port ด้วย Visual Basic 2008

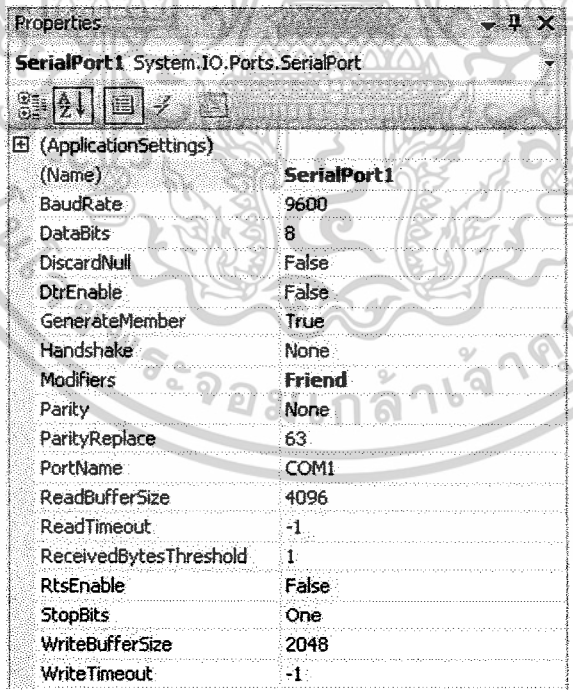
การติดต่อสื่อสารข้อมูลกับอุปกรณ์ภายนอกจาก Visual basic จะใช้การติดต่อผ่านพอร์ตอนุกรม โดย visual basic 2005 มีคอนโทรลที่ทำหน้าที่ในการควบคุมการรับ-ส่งข้อมูลผ่านพอร์ตอนุกรมไว้ให้ การใช้งานและโค๊ดที่ใช้แสดงดังรูป

3.2.3.1 การเลือก Control Serial Port จาก Toolbox



ภาพที่ 3.34 คอนโทรล SerialPort ใน Visual Basic 2008

3.2.3.2 การตั้งค่าให้กับ Control Serialport จาก Property Windows



ภาพที่ 3.35 พร็อพเพอตี้ของคอนโทรล SerialPort ใน Visual Basic 2008

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Properties ของ SerialPort Control ที่สำคัญมีดังนี้

Name	ตั้งค่าชื่อคอนโทรล SerialPort
BaudRate	ตั้งค่าอัตราการรับ-ส่งข้อมูล
DataBits	ตั้งค่าจำนวนบิตต่อชุดข้อมูล
Parity	ตั้งค่าประเภทของการตรวจสอบความผิดพลาดแบบพาริตี้
PortName	ตั้งค่าพอร์ตอนุกรมที่ต้องการใช้งาน
StopBits	ตั้งค่าบิตสุดท้ายของข้อมูลที่จะทำการส่ง

โค๊ดที่ใช้ในการควบคุมการรับส่งข้อมูลผ่าน Serial Port

การตั้งค่า Serial Port และการเปิด Port

```
Private Sub cmdConnect_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles cmdConnect.Click
        SerialPort1.PortName = "COM1"
        SerialPort1.BaudRate = 9600
        SerialPort1.DataBits = 8
        SerialPort1.Parity = IO.Ports.Parity.None
        SerialPort1.StopBits = IO.Ports.StopBits.One
        SerialPort1.Open()
End Sub
```

การปิด Port

```
Private Sub cmdDiscon_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles cmdDiscon.Click
        SerialPort1.Close()
End Sub
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การส่งข้อมูลออก Serial Port

```
Private Sub cmdSend_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles cmdSend.Click
    Dim Data2Send As String
    Data2Send = "TEST SERIAL PORT WITH VB2005"
    SerialPort1.Write(Data2Send)
End Sub
```

การรับข้อมูลจาก Serial Port

```
Private Sub SerialPort1_DataReceived(ByVal sender As Object, ByVal e As
System.IO.Ports.SerialDataReceivedEventArgs) Handles SerialPort1.DataReceived
    Dim DataRecieved As String
    DataRecieved = SerialPort1.ReadExisting
End Sub
```

3.2.4 การออกแบบฐานข้อมูล

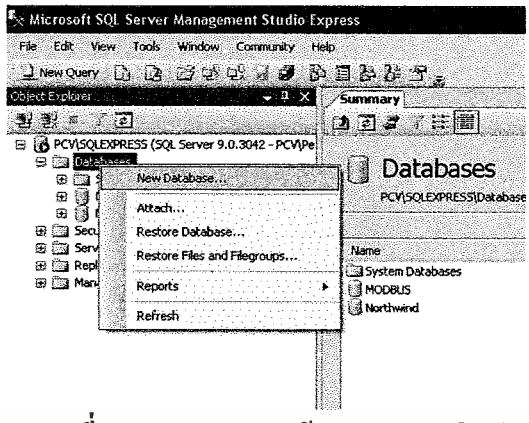
ในปริญญานิพนธ์นี้ใช้โปรแกรมจัดการฐานข้อมูล SQL Server 2005 Express Edition โดยใช้โปรแกรมจัดการฐานข้อมูล SQL Server Management Studio Express ฐานข้อมูลที่ใช้เป็นฐานข้อมูลที่มีเพียงตารางเดียวประกอบด้วยฟิลด์ข้อมูลดังนี้

1. ฟิลด์ Date ทำหน้าที่เก็บข้อมูลวันที่ของกระบวนการ
2. ฟิลด์ Time ทำหน้าที่เก็บข้อมูลเวลาของกระบวนการ
3. ฟิลด์ SV ทำหน้าที่เก็บข้อมูลค่าอุณหภูมิเป้าหมายของกระบวนการ(Setpoint Temperature)
4. ฟิลด์ PV ทำหน้าที่เก็บข้อมูลค่าอุณหภูมิในกระบวนการ(Process Temperature)
5. ฟิลด์ ER ทำหน้าที่เก็บข้อมูลผลต่างระหว่างค่าอุณหภูมิเป้าหมายกับอุณหภูมิในกระบวนการ(SV-PV)

การสร้างฐานข้อมูลโดยใช้ SQL Server Management Studio Express

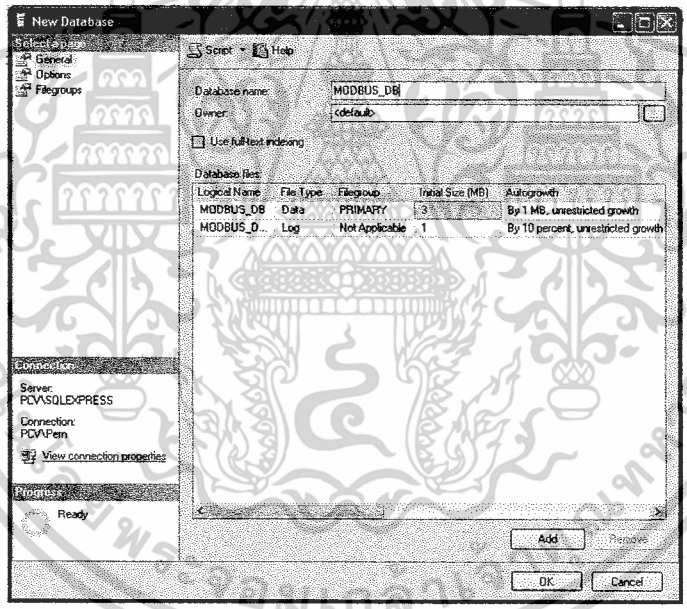
1. ที่หน้าต่าง Object Explorer เลือก Databases จากนั้นคลิกเมาส์ปุ่มขวาแล้วเลือก New Databases...

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 3.36 แสดงการสร้าง Database ใหม่

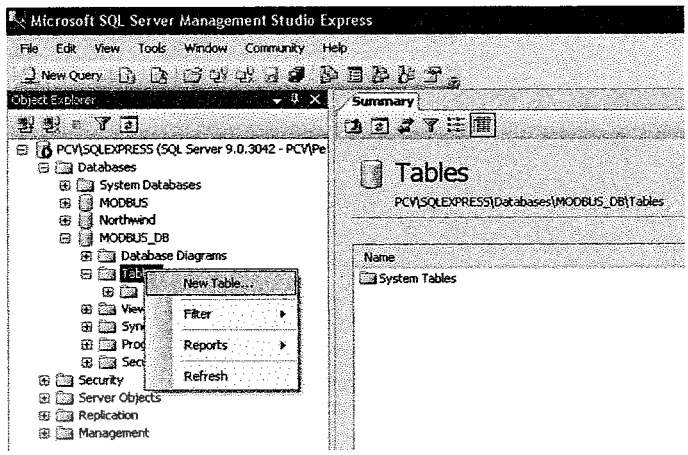
2. โปรแกรมจะแสดงหน้าต่างเพื่อให้เรากำหนดรายละเอียดของฐานข้อมูลที่ทำการสร้าง เมื่อกำหนดชื่อฐานข้อมูล และขนาดเริ่มต้นของไฟล์ให้กับฐานข้อมูลเรียบร้อยแล้ว ให้คลิกปุ่ม OK



ภาพที่ 3.37 แสดงการกำหนดรายละเอียดของฐานข้อมูล

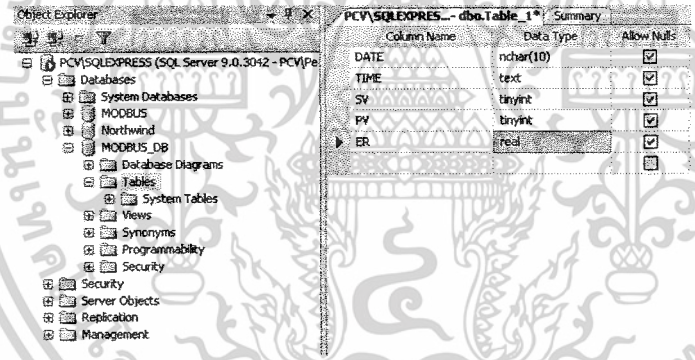
3. สร้างตารางโดยการเลือกฐานข้อมูลที่ต้องการ จากนั้นคลิกเมาส์ปุ่มขวาที่รายการ Tables แล้วเลือก New Table…

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 3.38 แสดงการสร้างตาราง

4. โปรแกรมจะแสดงหน้าต่างเพื่อให้เรากำหนดรายละเอียดของของตาราง เมื่อเรากำหนดคอลัมน์ต่าง ๆ เสร็จสิ้นก็จะได้รายละเอียดของตารางดังรูป แล้วคลิกที่ปุ่ม Save จากนั้นให้ใส่ชื่อตารางที่ต้องการบันทึก



ภาพที่ 3.39 แสดงตารางในฐานข้อมูลที่สร้างเสร็จแล้ว

3.2.5 การเชื่อมต่ออุปกรณ์ควบคุมกระบวนการผ่าน RS-485

3.2.5.1 การเชื่อมต่ออุปกรณ์ควบคุมอุณหภูมิ FUJI PXR4

PXR4 มีการเชื่อมต่อเพื่อติดต่อสื่อสาร โดยใช้ MODBUS Protocol ผ่านการอินเตอร์เฟส แบบ RS-485 2 WIRE ซึ่งใช้การติดต่อสื่อสารกับคอมพิวเตอร์หรือตัวอุปกรณ์ควบคุม เช่น PLC หน้าจอควบคุมแบบกราฟิกได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

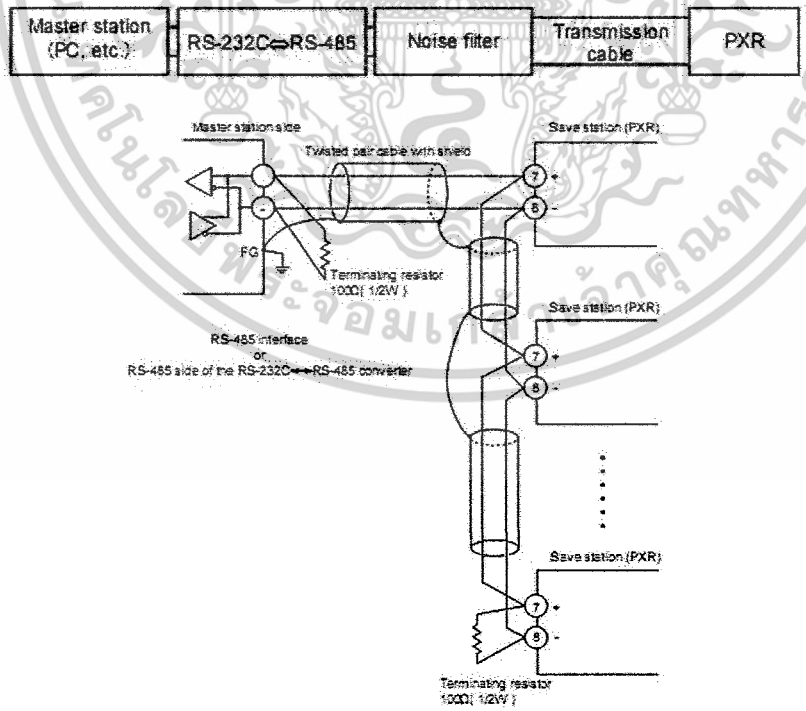
คุณสมบัติด้านการติดต่อสื่อสาร

ตารางที่ 3.2 คุณสมบัติด้านการสื่อสารของ FUJI PXR-4

รายการ	คุณสมบัติ
รูปแบบการเชื่อมต่อ	EIA RS-485 2 wire 1 : N
จำนวนอุปกรณ์ที่รองรับ	31 ตัว
ระยะทางในการส่งข้อมูล	ระยะทางรวมสูงสุด 500 เมตร
ความเร็วในการส่งข้อมูล	9600 bps
รูปแบบข้อมูล	Data bit 8 bit Stop bit 1 bit Parity None, Even, odd
โปรโตคอล	MODBUS RTU

การเชื่อมต่ออุปกรณ์ควบคุมอุณหภูมิ FUJI PXR4 ผ่าน RS-485

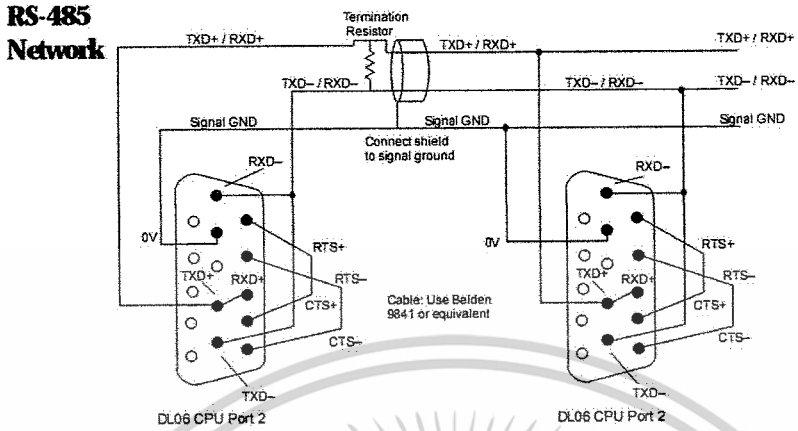
การเชื่อมต่ออุปกรณ์ควบคุมอุณหภูมิ FUJI PXR4 กับ RS-485 ต้องเชื่อมต่อผ่านพอร์ท 7 และ 8 ดังรูป



ภาพที่ 3.40 การเชื่อมต่ออุปกรณ์ควบคุมอุณหภูมิ FUJI PXR4 ผ่าน RS-485

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

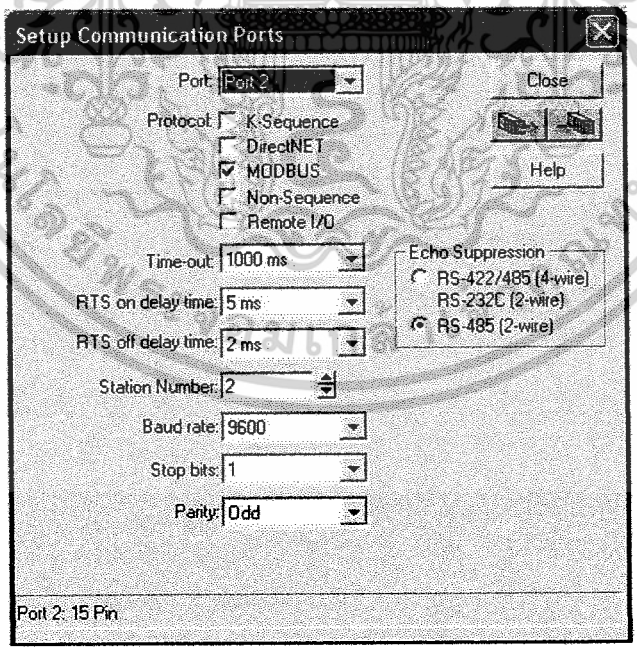
3.2.5.2 การเชื่อมต่อ PLC KOYO DO-06DD1 ผ่าน RS-485



ภาพที่ 3.41 การเชื่อมต่อ PLC KOYO DO-06DD1 ผ่าน RS-485

การตั้งค่า Port2 ให้มีการติดต่อสื่อสารแบบ MODBUS ผ่าน RS-485

จากโปรแกรม Direct Soft 32 เลือกเมนู PLC → Setup → Secondary Com Port จากนั้นตั้งค่าดั้งนี้



ภาพที่ 3.42 การตั้งค่า Port2 ให้มีการติดต่อสื่อสารแบบ MODBUS

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

การทดลองและผลการทดลอง

การทดลองแบ่งออกเป็นสองส่วนคือ

- การทดลองการควบคุมอุณหภูมิของน้ำในถังควบคุมโดยใช้การควบคุมแบบต่างๆ เป็นการทดลองกระบวนการที่ได้สร้างขึ้นว่าเมื่อมีการใช้งานจริงจะมีความสามารถในการเข้าสู่อุณหภูมิที่ตั้งไว้เป็นอย่างไร และคุณสมบัติในการควบคุมแบบต่างๆนั้นเป็นอย่างไร

- การทดลองการรับ-ส่งค่าผ่านโปรโตคอล MODBUS เป็นการทดลองโปรแกรมที่ได้ออกแบบเพื่อใช้ในการรับ-ส่งค่าโดยใช้โปรโตคอลมอดบัสว่ามีการทำงานเป็นไปตามที่ได้ออกแบบไว้หรือไม่

4.1 การทดลองการควบคุมอุณหภูมิของน้ำในถังควบคุม โดยใช้การควบคุมแบบต่างๆ

ขั้นตอนการทดลอง

1. เตรียมกระบวนการให้พร้อม โดยตวงน้ำปริมาตร 5 ลิตร โดยให้มีอุณหภูมิเริ่มต้น 27 องศาเซลเซียส
2. เปิดโปรแกรมแสดงผลกระบวนการและ Login เข้าสู่โปรแกรม
3. เชื่อมต่อโปรแกรมเข้ากับกระบวนการโดยคลิกที่
4. เปิดฟอร์ม PXR4 Setting เพื่อตั้งค่าการควบคุมให้กับกระบวนการ
5. กดปุ่ม RUN ที่หน้าฟอร์ม Process monitoring
6. เปิดหน้าฟอร์ม Trend Monitoring เพื่อสังเกตการทำงานของกระบวนการในรูปแบบกราฟ
7. บันทึกผลการทดลองจากกราฟที่ได้เมื่ออุณหภูมิอยู่ในสภาวะคงที่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.1.1 การควบคุมอุณหภูมิของน้ำโดยใช้การควบคุมแบบ P

ผลการทดลอง

ปริมาตรน้ำ 5 ลิตร

อุณหภูมิเริ่มต้น 27 องศาเซลเซียส

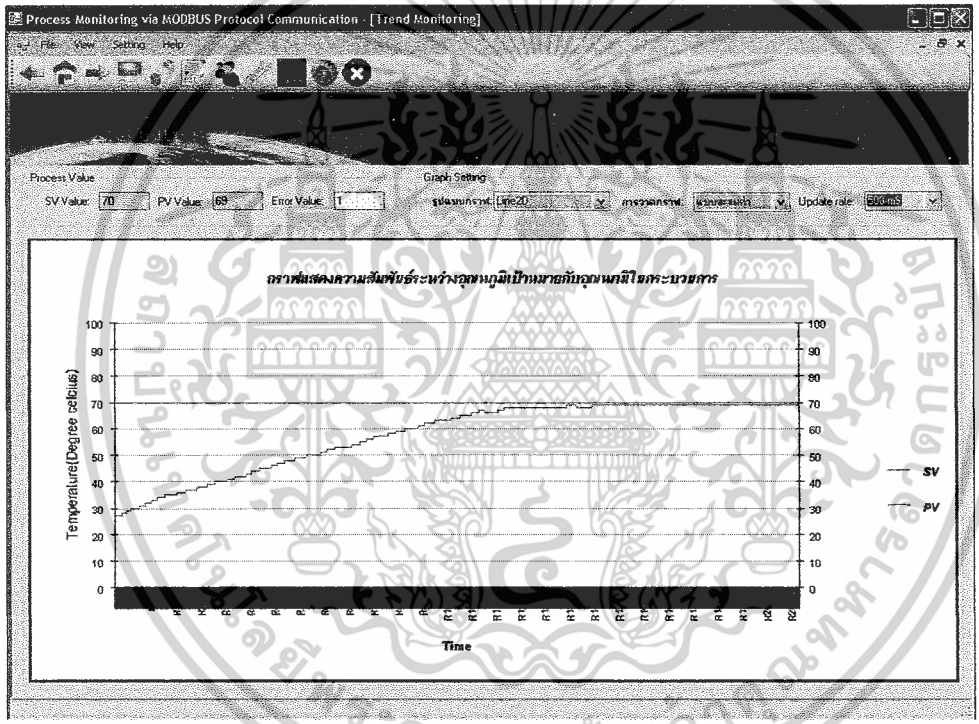
อุณหภูมิที่ต้องการ 70 องศาเซลเซียส

เวลาที่ใช้เพื่อเข้าถึงอุณหภูมิที่ต้องการครั้งแรก 12 นาที 45 วินาที

เวลาที่ใช้ในการคงสภาวะค่าที่ต้องการ ไม่มี

ค่า P = 5.0 ค่า I = 0.0 ค่า D = 0.0

ค่าพุ่งเกิน = 0 องศาเซลเซียส



ภาพที่ 4.1 กราฟแสดงการเข้าสู่ค่า SV เมื่อใช้การควบคุมแบบ P

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

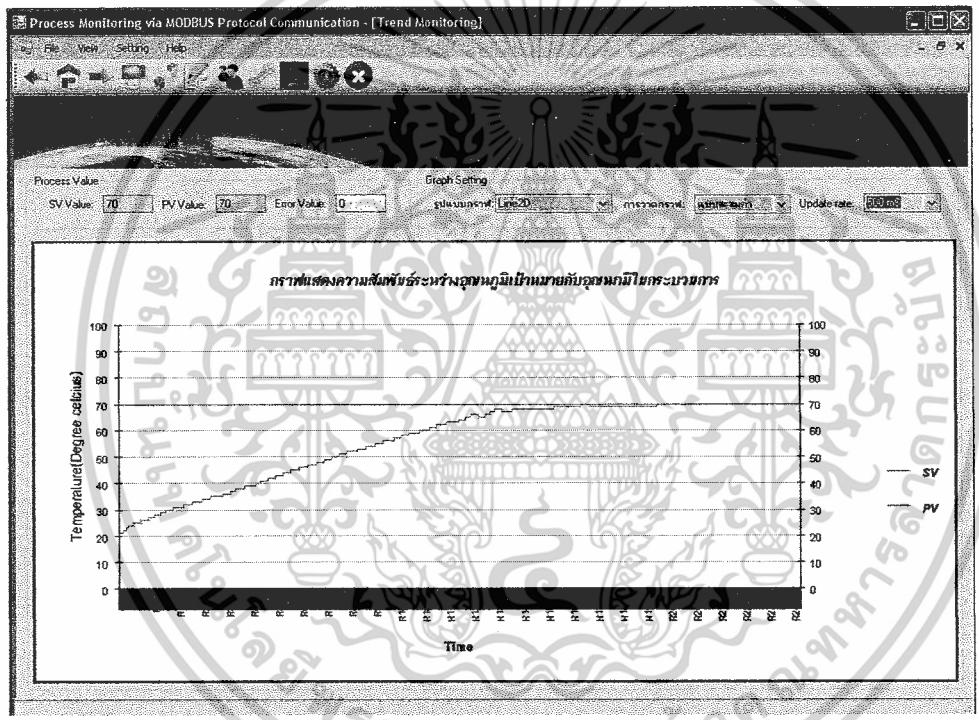
4.1.2 การควบคุมอุณหภูมิของน้ำโดยใช้การควบคุมแบบ PI

ผลการทดลอง

ปริมาตรน้ำ	5	ลิตร				
อุณหภูมิเริ่มต้น	27	องศาเซลเซียส				
อุณหภูมิที่ต้องการ	70	องศาเซลเซียส				
เวลาที่ใช้เพื่อเข้าถึงอุณหภูมิที่ต้องการครั้งแรก			13	นาที	27	วินาที
เวลาที่ใช้ในการคงสภาวะค่าที่ต้องการ			17	นาที	46	วินาที

ค่า P = 5.0 ค่า I = 240.0 ค่า D = 0.0

ค่าพุ่งเกิน = 0 องศาเซลเซียส



ภาพที่ 4.2 กราฟแสดงการเข้าสู่ค่า SV เมื่อใช้การควบคุมแบบ PI

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.1.3 การควบคุมอุณหภูมิของน้ำโดยใช้การควบคุมแบบ PD

ผลการทดลอง

ปริมาตรน้ำ 5 ลิตร

อุณหภูมิเริ่มต้น 27 องศาเซลเซียส

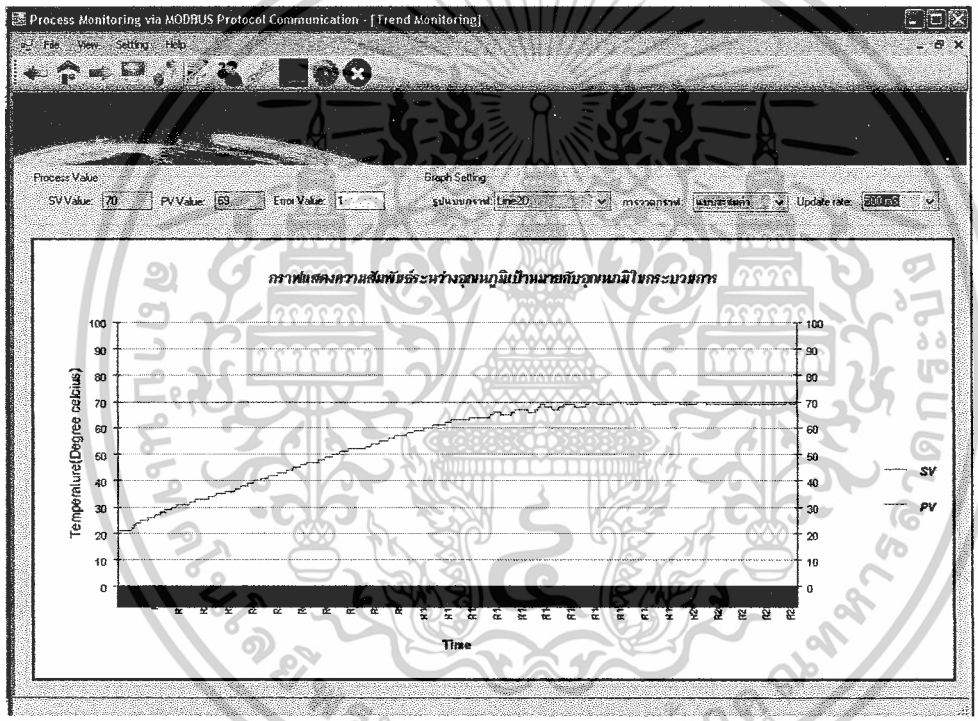
อุณหภูมิที่ต้องการ 70 องศาเซลเซียส

เวลาที่ใช้เพื่อเข้าถึงอุณหภูมิที่ต้องการครั้งแรก 12 นาที 57 วินาที

เวลาที่ใช้ในการคงสภาวะค่าที่ต้องการ ไม่มี

ค่า P = 5.0 ค่า I = 0.0 ค่า D = 60.0

ค่าพุ่งเกิน = 0 องศาเซลเซียส



ภาพที่ 4.3 กราฟแสดงการเข้าสู่ค่า SV เมื่อใช้การควบคุมแบบ PD

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

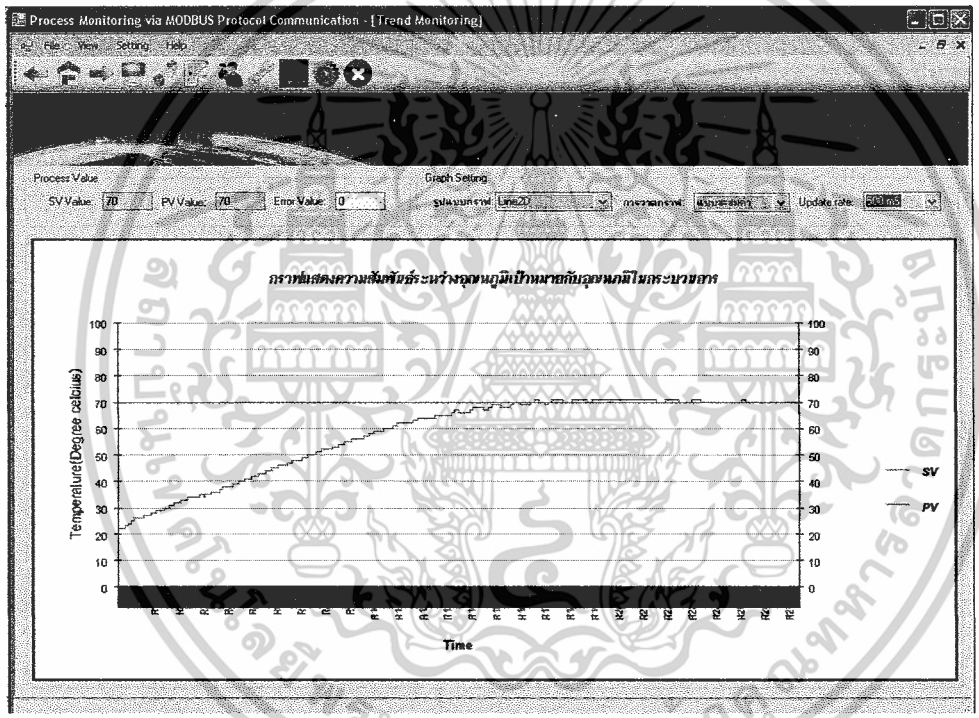
4.1.4 การควบคุมอุณหภูมิของน้ำโดยใช้การควบคุมแบบ PID

ผลการทดลอง

ปริมาตรน้ำ	5	ลิตร			
อุณหภูมิเริ่มต้น	27	องศาเซลเซียส			
อุณหภูมิที่ต้องการ	70	องศาเซลเซียส			
เวลาที่ใช้เพื่อเข้าถึงอุณหภูมิที่ต้องการครั้งแรก	13	นาที	3	วินาที	
เวลาที่ใช้ในการคงสภาวะค่าที่ต้องการ	22	นาที	35	วินาที	

ค่า P = 5.0 ค่า I = 240.0 ค่า D = 60.0

ค่าพุ่งเกิน = 1 องศาเซลเซียส



ภาพที่ 4.4 กราฟแสดงการเข้าสู่ค่า SV เมื่อใช้การควบคุมแบบ PID

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.1.5 การควบคุมอุณหภูมิของน้ำโดยใช้การควบคุมแบบ Auto-Tuning

ผลการทดลอง

ปริมาตรน้ำ 5 ลิตร

อุณหภูมิเริ่มต้น 27 องศาเซลเซียส

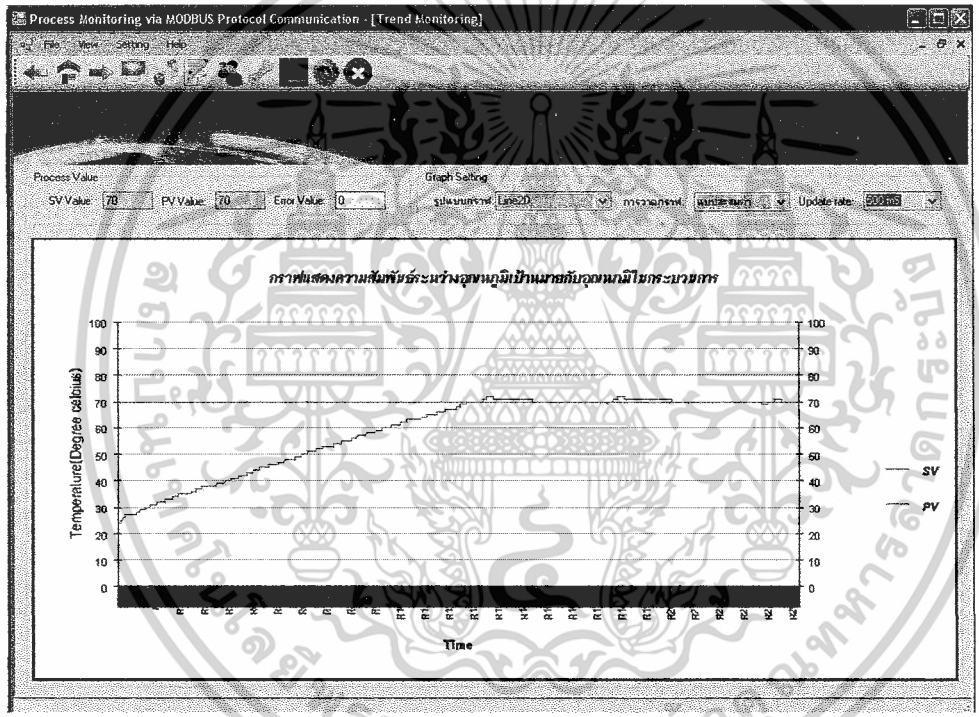
อุณหภูมิที่ต้องการ 70 องศาเซลเซียส

เวลาที่ใช้เพื่อเข้าถึงอุณหภูมิที่ต้องการครั้งแรก 10 นาที 44 วินาที

เวลาที่ใช้ในการคงสภาวะค่าที่ต้องการ 19 นาที 21 วินาที

ค่า P = 1.2 ค่า I = 70.0 ค่า D = 13.5

ค่าพุ่งเกิน = 2 องศาเซลเซียส



ภาพที่ 4.5 กราฟแสดงการเข้าสู่ค่า SV เมื่อใช้การควบคุมแบบ Auto-Tuning

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.1.6 การควบคุมอุณหภูมิของน้ำโดยใช้การควบคุมแบบ Fuzzy

ผลการทดลอง

ปริมาตรน้ำ 5 ลิตร

อุณหภูมิเริ่มต้น 27 องศาเซลเซียส

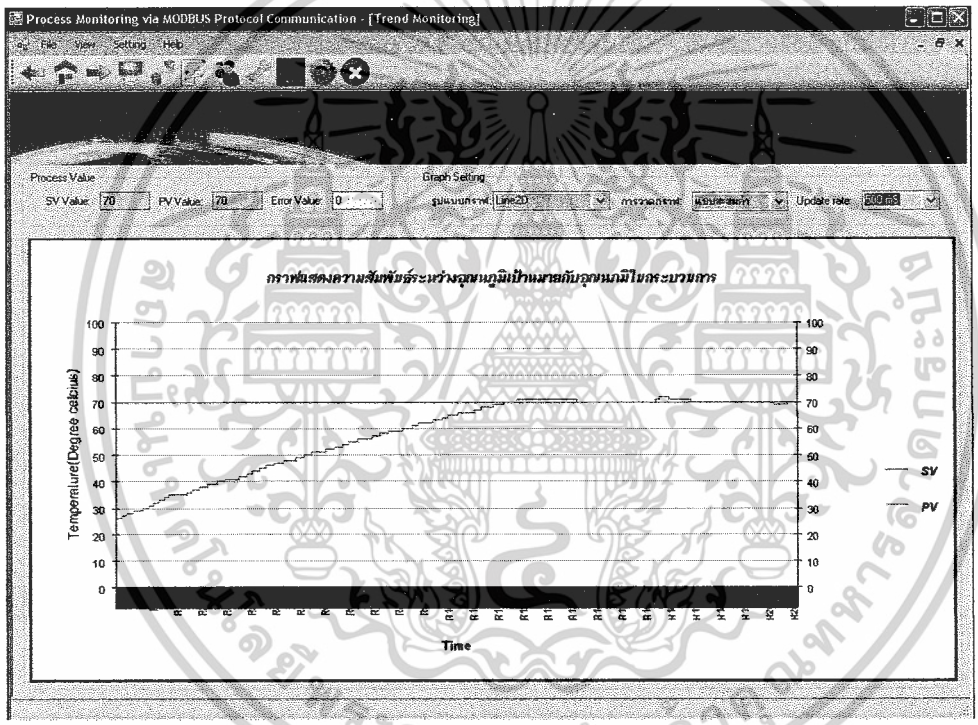
อุณหภูมิที่ต้องการ 70 องศาเซลเซียส

เวลาที่ใช้เพื่อเข้าถึงอุณหภูมิที่ต้องการครั้งแรก 9 นาที 47 วินาที

เวลาที่ใช้ในการคงสภาวะค่าที่ต้องการ 16 นาที 51 วินาที

ค่า P = 5.0 ค่า I = 240.0 ค่า D = 60.0

ค่าพุ่งเกิน = 2 องศาเซลเซียส



ภาพที่ 4.6 กราฟแสดงการเข้าสู่ค่า SV เมื่อใช้การควบคุมแบบ Fuzzy

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2 การทดลองการรับ-ส่งค่าผ่านโปรโตคอล MODBUS

ขั้นตอนการทดลอง

1. เปิดโปรแกรมแสดงผลกระบวนการและตั้งค่าการติดต่อสื่อสาร โดยเปิดฟอร์ม Communication Setting โดยเลือก Setting > Communication Setting

2. เปิดฟอร์ม MODBUS Protocol Test โดยคลิกที่ View > MODBUS Test กดปุ่ม เพื่อเปิดการเชื่อมต่อกับอุปกรณ์ควบคุม

3. กรอกข้อมูลที่ต้องการส่งลงในช่อง Data to send

4. กดปุ่ม Cal CRC เพื่อคำนวณบิตตรวจสอบความผิดพลาดของข้อมูลแบบ CRC

5. กดปุ่ม Transmit เพื่อส่งข้อมูลไปยังอุปกรณ์ควบคุม

6. บันทึกผลข้อมูลที่ตอบกลับ โดยดูจากช่อง Data received และบันทึกการเปลี่ยนแปลงที่เกิดขึ้นที่อุปกรณ์ควบคุม

4.2.1 การทดลองการรับ-ส่งค่าผ่านโปรโตคอล MODBUS กับอุปกรณ์ควบคุมอุณหภูมิ FUJI PXR-4

ในการทดลองนี้เป็นการทดลองตั้งค่า Setpoint จากโปรแกรมที่ได้สร้างขึ้นไปยังอุปกรณ์ควบคุมอุณหภูมิ FUJI PXR-4 โดยตั้งค่า Setpoint = 50 องศาเซลเซียส (0032H)

ตารางที่ 4.1 แสดงผลการรับส่งข้อมูลกับอุปกรณ์ควบคุมอุณหภูมิ FUJI PXR-4

ชื่อข้อมูล	ข้อมูลที่ส่งออก	ข้อมูลที่รับได้
Slave Address	0x01	0x01
Function Code	0x06	0x06
Address Hi	0x03	0x03
Address Lo	0xEA	0xEA
Value Hi	0x00	0x00
Value Lo	0x32	0x32
CRC Hi	0x29	0x29
CRC Lo	0xAF	0xAF

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การเปลี่ยนแปลงที่อุปกรณ์ควบคุมอุณหภูมิ FUJI PXR-4

ค่า SV ที่หน้าจอของอุปกรณ์ควบคุมอุณหภูมิ FUJI PXR-4 เปลี่ยนเป็น 50 องศาเซลเซียส

4.2.2 การทดลองการรับ-ส่งค่าผ่านโปรโตคอล MODBUS กับอุปกรณ์ควบคุมแบบตรรกะที่สามารถโปแกรมได้ (PLC) KOYO DO-06DD1

ในการทดลองนี้เป็นการทดลองตั้งค่าเอาต์พุตจากโปรแกรมที่ได้สร้างขึ้นไปยังอุปกรณ์ควบคุมแบบตรรกะที่สามารถโปแกรมได้ (PLC) KOYO DO-06DD1 โดยตั้งค่า เอาต์พุตให้ ON ทั้งหมด (FFFFH)

ตารางที่ 4.2 แสดงผลการรับส่งข้อมูลกับอุปกรณ์ควบคุม PLC KOYO DO-06DD1

ชื่อข้อมูล	ข้อมูลที่ส่งออก	ข้อมูลที่รับได้
Slave Address	0x02	0x02
Function Code	0x06	0x06
Address Hi	0x41	0x41
Address Lo	0x40	0x40
Value Hi	0xFF	0xFF
Value Lo	0xFF	0xFF
CRC Hi	0x9D	0x9D
CRC Lo	0xA1	0xA1

การเปลี่ยนแปลงที่ PLC KOYO DO-06DD1

ไฟแสดงสถานะเอาต์พุต Y0-Y17 ติดทุกดวง

4.3 สรุปผลการทดลอง

จากการทดลองส่วนแรกการทดลองการควบคุมอุณหภูมิของน้ำในถังควบคุม โดยใช้การควบคุมแบบต่าง ๆ นั้น พบว่าระบบควบคุมกระบวนการที่ได้ออกแบบสามารถตอบสนองความต้องการในการตั้งค่าอุณหภูมิจากผู้ใช้ได้ตามต้องการ แต่พบว่าระยะเวลาในการเข้าสู่อุณหภูมิเป้าหมายผิดพลาดไปจากที่ได้ออกแบบไว้ค่อนข้างมาก โดยมีระยะเวลาในการเข้าสู่ค่าเป้าหมายที่แตกต่างกันในแต่ละรูปแบบการควบคุม ทั้งนี้เป็นไปตามคุณสมบัติของการควบคุมแบบต่าง ๆ ซึ่งจากข้อมูลในการทดลองครั้งนี้ สามารถนำมาเปรียบเทียบกันและเลือกรูปแบบระบบควบคุมที่เหมาะสมได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากการทดลองส่วนที่สอง การทดลองการรับ-ส่งค่าผ่านโปรโตคอล MODBUS กับอุปกรณ์ควบคุม พบว่าโปรแกรมที่ได้ออกแบบสามารถตอบสนองต่อการใช้งานกับโปรโตคอลมอดบัสได้เป็นอย่างดี สามารถทำการคำนวณค่า CRC และส่งค่า-รับค่าได้อย่างถูกต้อง สามารถควบคุมอุปกรณ์จากระยะไกลได้อย่างมีประสิทธิภาพ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5
สรุปผลการวิจัยและข้อเสนอแนะ

5.1 สรุปผลการวิจัย

ในการจัดทำปริญญานิพนธ์นี้ ได้ทำการออกแบบและสร้างกระบวนการควบคุมอุณหภูมิของของเหลวและออกแบบโปรแกรมแสดงผลกระบวนการ ซึ่งมีหน้าที่ในการแสดงผลและควบคุมกระบวนการ โดยใช้การติดต่อสื่อสารกับอุปกรณ์ควบคุมโดยใช้โปรโตคอลมอดบัส โดยในกระบวนการควบคุมอุณหภูมินั้น จะใช้อุปกรณ์เซนเซอร์คือ RTD และอุปกรณ์ควบคุมคือ อุปกรณ์ควบคุมอุณหภูมิ FUJI PXR-4 อุปกรณ์ควบคุมตัวสุดท้ายคือฮีตเตอร์ นอกจากนี้ยังมีอุปกรณ์อื่น ๆ ที่ประกอบในกระบวนการด้วย ได้แก่ ปั๊มสูบน้ำ โซลินอยด์วาล์ว มอเตอร์กวน และอื่น ๆ ซึ่งควบคุมโดยใช้เครื่องควบคุมที่สามารถโปรแกรมได้ คือ PLC KOYO DO-06DD1

ในส่วนการติดต่อสื่อสารนั้น ได้กำหนดให้เครื่องคอมพิวเตอร์ ทำหน้าที่เป็นอุปกรณ์มาสเตอร์ และอุปกรณ์ควบคุมอุณหภูมิ FUJI PXR-4 กับ PLC KOYO DO-06DD1 เป็นอุปกรณ์สเลฟ การติดต่อสื่อสารนั้น ใช้การติดต่อสื่อสารผ่าน RS485 และใช้อุปกรณ์แปลงสัญญาณ RS485 เป็น RS232 เพื่ออินเตอร์เฟซกับคอมพิวเตอร์

การออกแบบโปรแกรมแสดงผลกระบวนการควบคุมการติดต่อสื่อสารนั้น ใช้โปรแกรม Visual Basic 2008 เป็นโปรแกรมหลัก ใช้โปรแกรม SQL Server 2005 Express ในการจัดการฐานข้อมูล และใช้โปรแกรม Crystal Report for Visual Basic ในการออกแบบรายงาน โปรแกรมที่ออกแบบเสร็จสมบูรณ์นั้น มีฟังก์ชันในการใช้งานมากมายได้แก่ การตั้งค่าผู้ใช้งาน การตั้งค่าการติดต่อสื่อสาร การแสดงผลกระบวนการแบบรีลไทม์ การแสดงผลตัวแปรในกระบวนการด้วยกราฟ การทดสอบการรับ ส่งค่าผ่านโปรโตคอลมอดบัส การตั้งค่าอุปกรณ์ควบคุมอุณหภูมิ PXR-4 จากคอมพิวเตอร์ การแสดงรายงานข้อมูลย้อนหลัง

ในการจัดทำปริญญานิพนธ์นี้ ได้ตั้งวัตถุประสงค์และขอบเขตของปริญญานิพนธ์เอาไว้หลายข้อซึ่งผลจากการดำเนินงานนั้นพบว่า ได้จัดทำจนบรรลุตามที่ได้ตั้งไว้ทั้งหมด ในระยะเวลาที่กำหนดไว้ ซึ่งทำให้ได้ความรู้ในด้านต่าง ๆ เป็นอย่างมาก ได้แก่ การออกแบบกระบวนการควบคุมอุณหภูมิ การเลือกใช้อุปกรณ์วัดอุณหภูมิ การเลือกใช้อุปกรณ์ควบคุม การพัฒนาโปรแกรมพีแอลซีและโปรแกรมคอมพิวเตอร์ เป็นต้น

5.2 ข้อเสนอแนะและแนวทางการพัฒนา

การออกแบบกระบวนการควบคุมอุณหภูมินั้น มีข้อจำกัดในเรื่องของงบประมาณซึ่งทำให้ได้ระบบที่ไม่ดีเท่าที่ควร เช่น การวัดระดับของของเหลวซึ่งใช้หลักการวัดโดยใช้ Glass gauge ซึ่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จำเป็นต้องใช้วิธีการสังเกตจากผู้ใช้งาน โดยตรงทำให้ไม่สะดวกในการใช้งานจริง ซึ่งหากใช้การวัดระดับที่สามารถส่งสัญญาณมายังระบบควบคุมได้ จะเป็นการดีกว่าวาล์วที่ใช้ในกระบวนการนั้นเมื่อใช้จริงพบว่า ของเหลวไม่สามารถไหลผ่านได้อย่างสะดวก ซึ่งเกิดจากการเลือกวาล์วที่ไม่เหมาะสม

การออกแบบโปรแกรมแสดงผลกระบวนการการควบคุมการติดต่อสื่อสารผ่านโปรโตคอลมอดบัสนั้นพบว่า พบปัญหาในการจัดการลำดับการรับส่งข้อมูล เนื่องจากมีการทำงานที่ซับซ้อน หากพัฒนาโปรแกรมที่เป็นโปรแกรมกลาง ที่รับผิดชอบการติดต่อสื่อสารผ่านโปรโตคอลมอดบัสเพียงอย่างเดียว แล้วเชื่อมต่อเข้ากับโปรแกรมแสดงผลกระบวนการจะเป็นการดีกว่า เพราะจะทำให้ลดภาระของโปรแกรมลงและลดการผิดพลาดในการควบคุมการติดต่อสื่อสารได้อีกด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

เกษม กมลชัยพิสิฐ. 2550. รอบรู้ประยุกต์ใช้ SQL Server 2005. พิมพ์ครั้งที่ 1. กรุงเทพมหานคร: สำนักพิมพ์แห่งจุฬาลงกรณ์มหาวิทยาลัย.

ดลวัฒน์ พิพุธวัฒน์, นายทิพยเดช วิจิตรกุล, นายนราทิพย์ นิ่มเพ็ง. 2550. การแสดงผลกระบวนการด้วยโปรแกรม WONDERWARE ปริญญานิพนธ์ ภาควิชาวิศวกรรมการวัดคุม คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง.

พร้อมเลิศ หล่อวิจิตร. 2549. คู่มือเรียน Visual Basic 2005. กรุงเทพมหานคร:บริษัท โปรวิชั่น จำกัด

ศุภชัย สมพานิช. 2551. Database Programming ด้วย VB2008 & VC#2008. พิมพ์ครั้งที่ 1. กรุงเทพมหานคร:บริษัท ไอดีซี อินโฟ ดิสทริบิวเตอร์ เซ็นเตอร์ จำกัด.

ศุภชัย สมพาณิช. 2549. สร้างรายงานด้วย Crystal Reports XI ฉบับสมบูรณ์. พิมพ์ครั้งที่ 1. กรุงเทพมหานคร: บริษัท ไอดีซี อินโฟ ดิสทริบิวเตอร์ เซ็นเตอร์ จำกัด.

DL06 Micro PLC User Manual

Instruction Manual MICRO CONTROLLER X COMMUNICATION FUNCTIONS (RS-485 MODBUS)

MODBUS over serial line specification and implementation guide V1.02. MODBUS -IDA.ORG

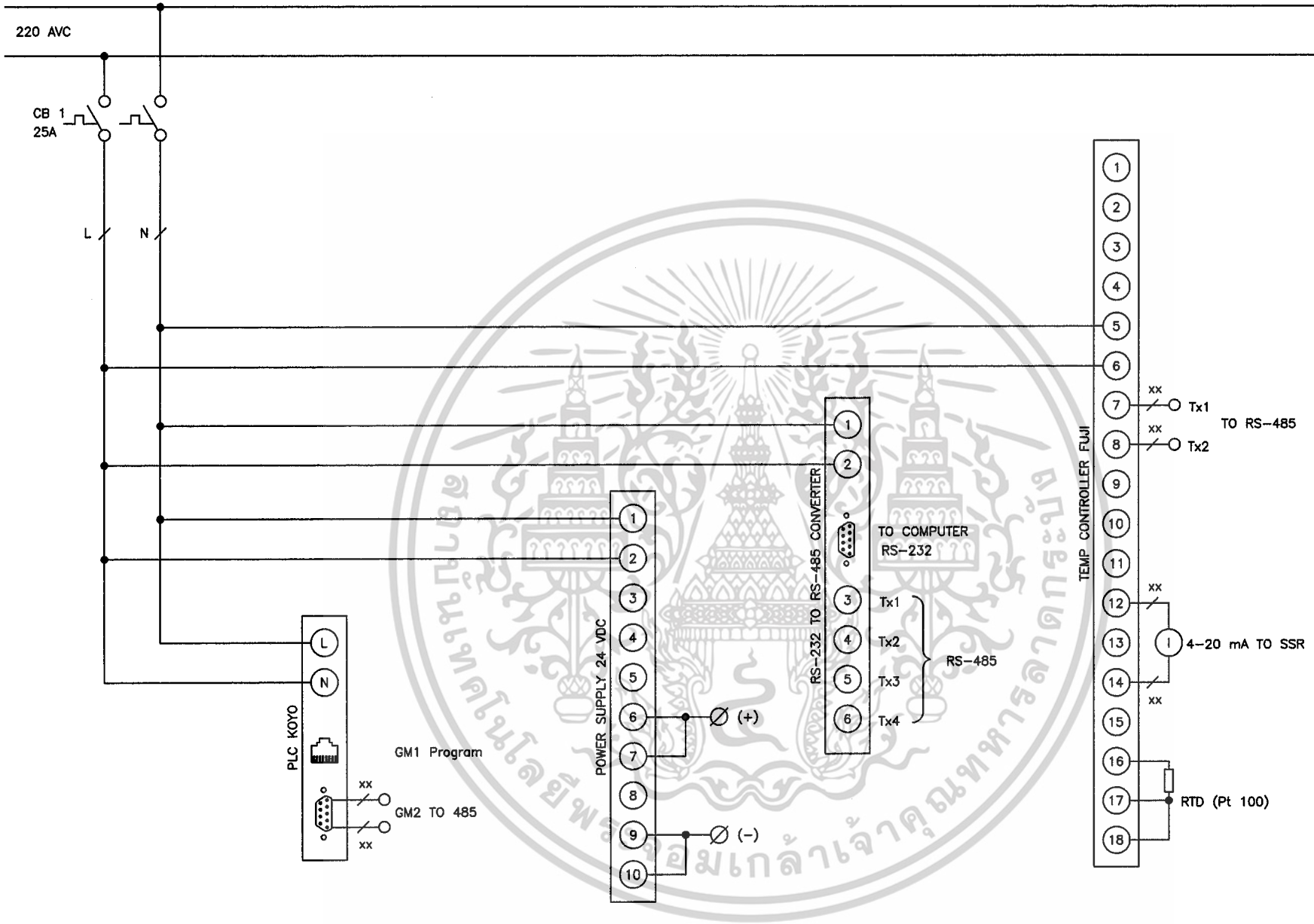
MODBUS Protocol Reference Guide. M-System Co.,LTD.

Modicon Modbus Protocol Reference Guide PI-MBUS-300 Rev. J. MODICON, Inc., Industrial Automation Systems

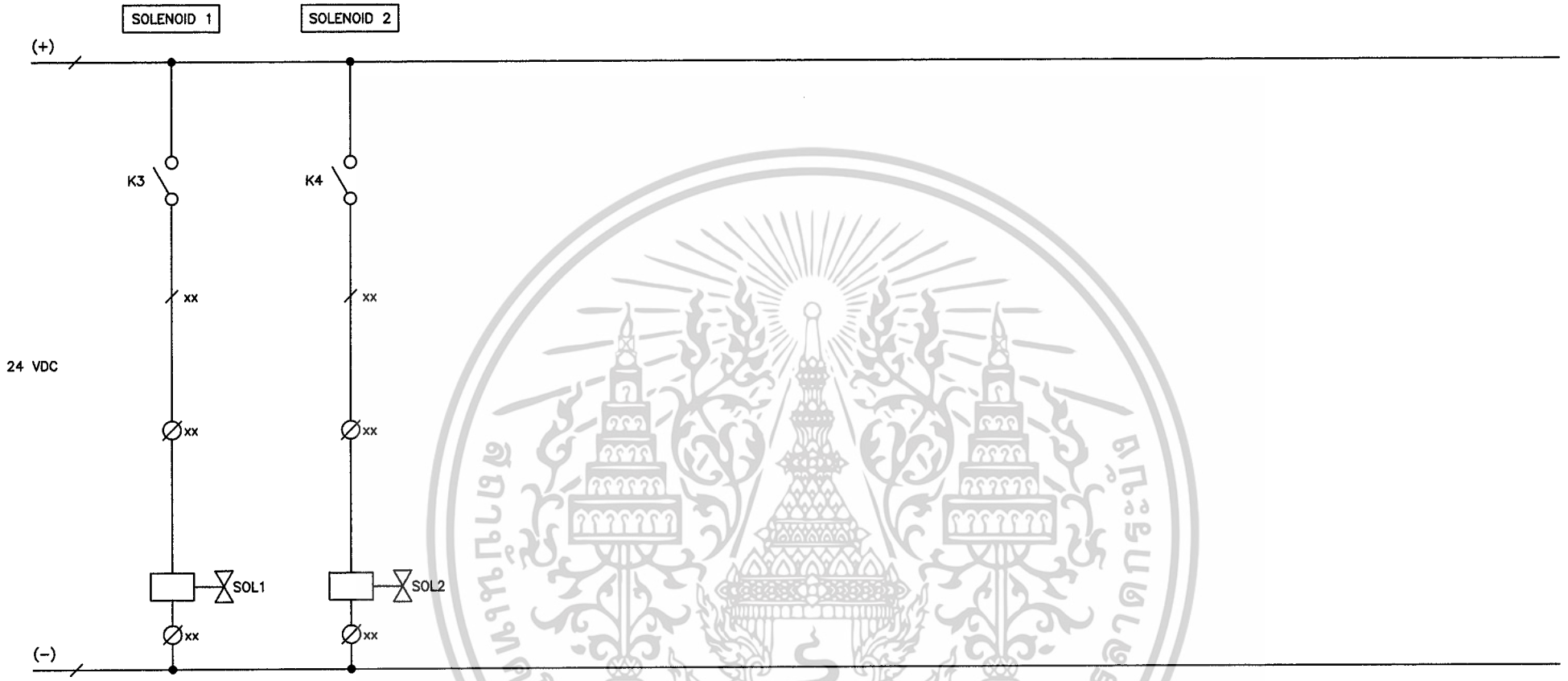
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

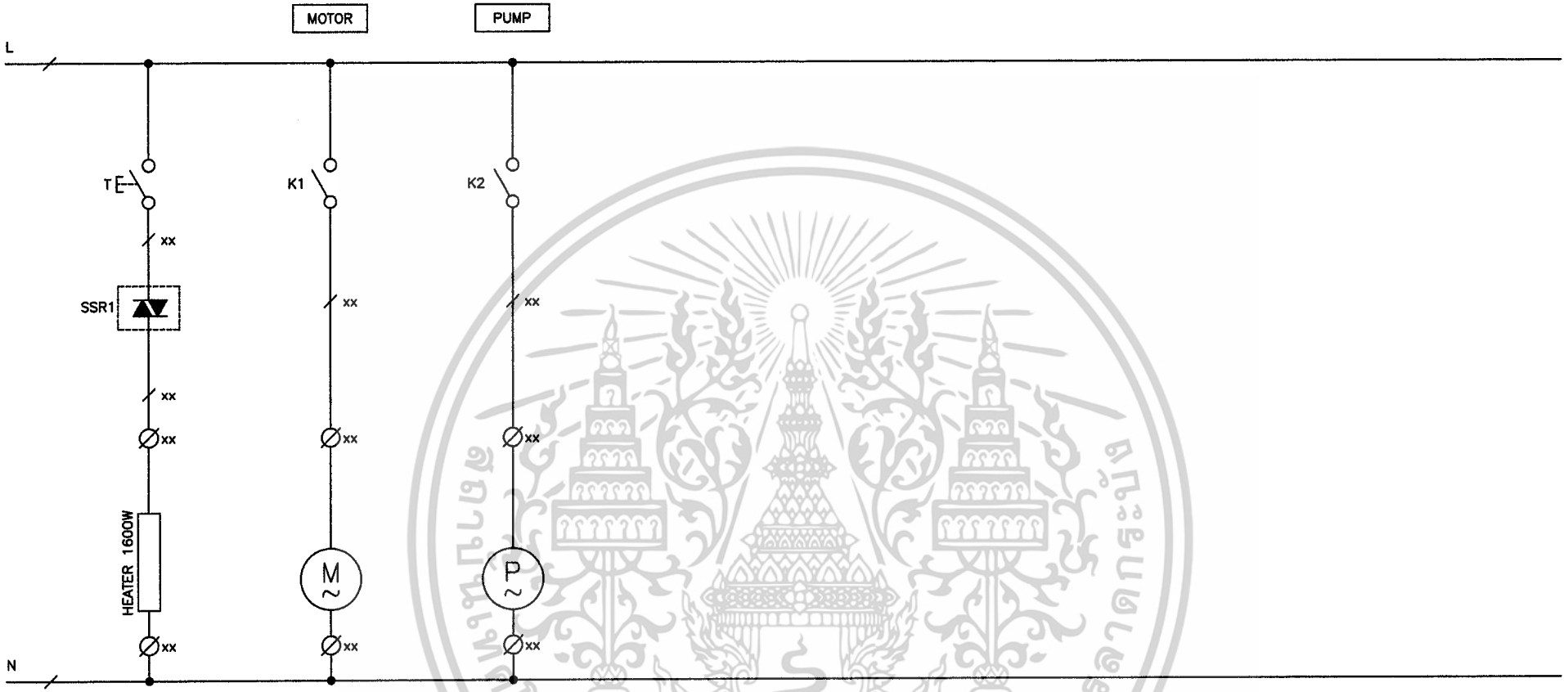


		30/01/09		Page
		INSTRUMENTATION ENGINEERING		1



สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

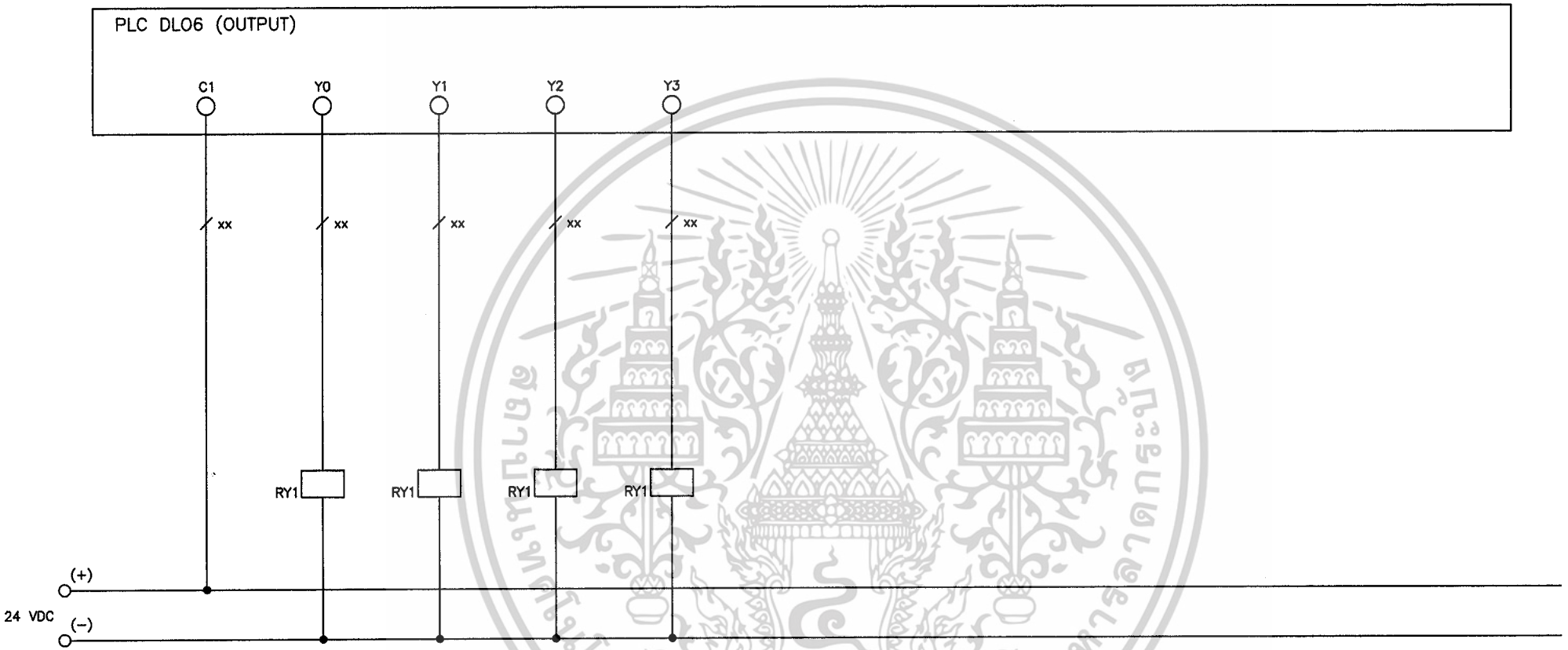
		30/01/09		Page
		INSTRUMENTATION ENGINEERING		2



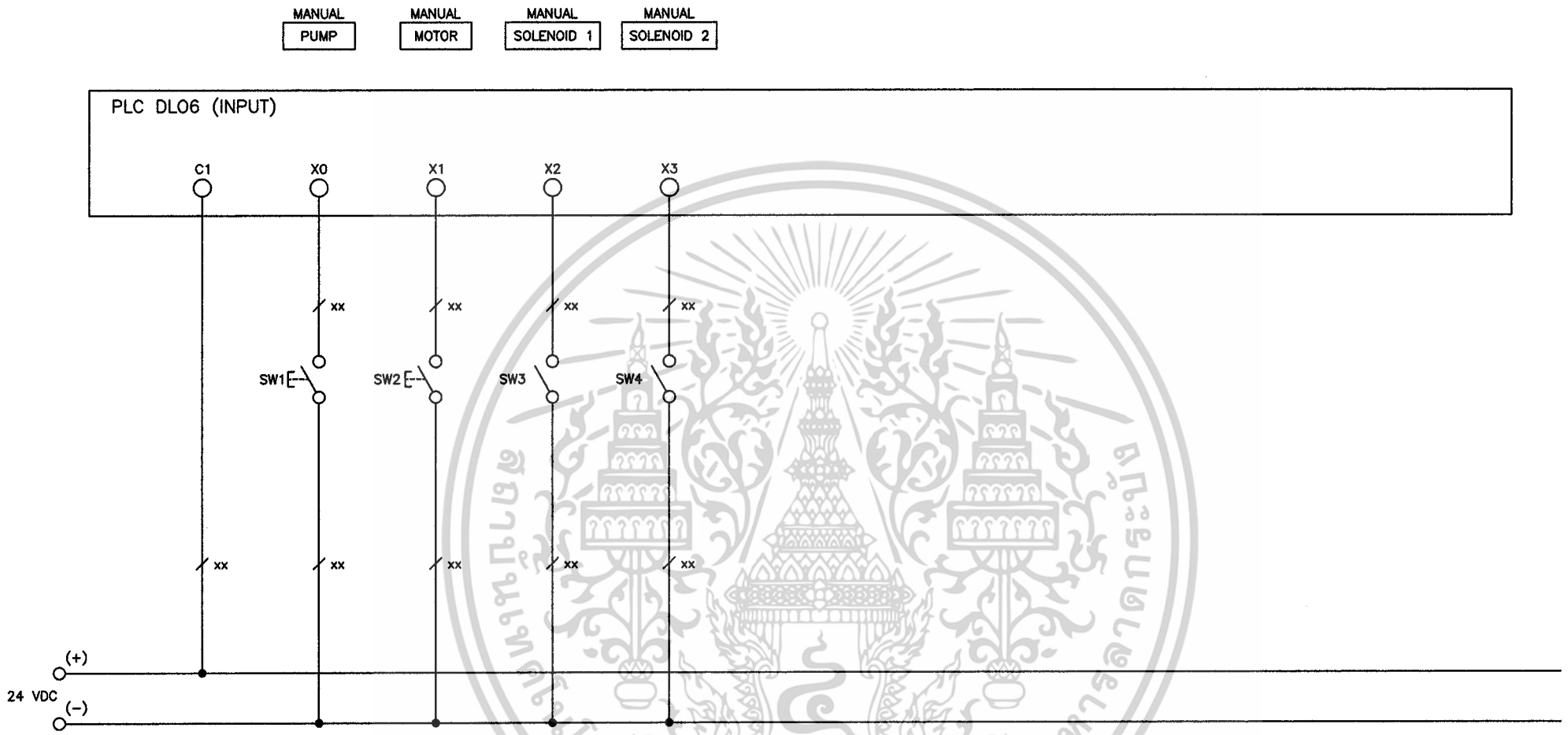
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

		30/01/09		Page
		INSTRUMENTATION ENGINEERING		3

PUMP | MOTOR | SOLENOID 1 | SOLENOID 2



		30/01/09		Page
		INSTRUMENTATION ENGINEERING		4



สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

		30/01/09		Page
		INSTRUMENTATION ENGINEERING		5

Introduction to Serial Communications

In this Appendix

Introduction to Serial Communications K–2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Introduction to Serial Communications

*Direct*LOGIC® PLCs have two built-in serial communiction ports which can be used to communicate to other PLCs or to other serial devices. In order to fully understand the capabilities and limitations of the serial ports, a brief introduction to serial communications is in order.

There are three major components to any serial communications setup:

Wiring standard

Communications protocol

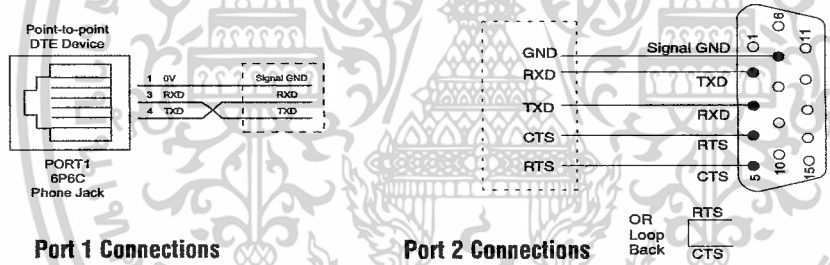
Communications parameters

Each of these will be discussed in more detail as they apply to *Direct*LOGIC PLCs.

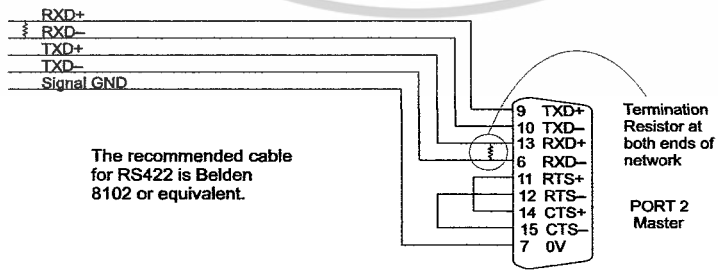
Wiring Standards

There are three different wiring standards that can be used with *Direct*LOGIC PLCs: RS-232C, RS-422 and RS-485.

RS-232C is a point-to-point wiring standard with a practical wiring distance of 15 meters, or 50 feet, maximum. This means that only two devices can communicate on an RS-232C network, a single master device and a single slave device, and the total cable length cannot exceed 50 feet. Belden® 8102, or equivalent, is recommended for RS-232C networks.

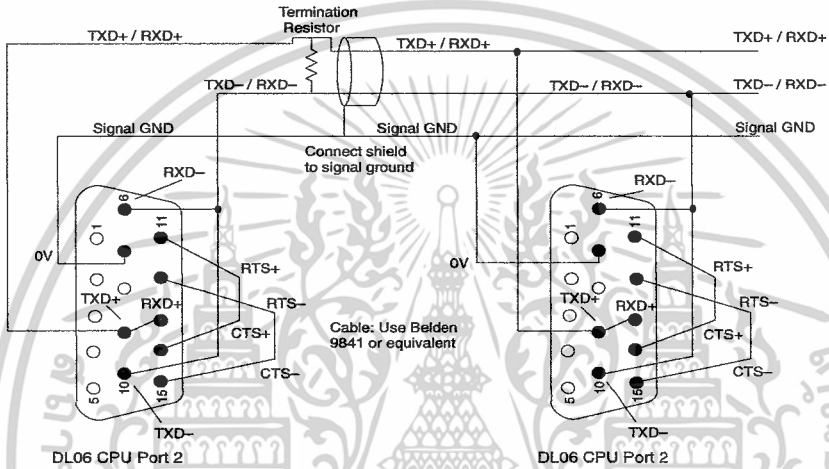


RS-422 is a multi-point wiring standard with a practical wiring distance 0f 1000 meters, or 3280 feet, maximum total length. The RS-422 wiring standard does not specify a network topology, but in practice, a daisy-chain topology with the master at one end is the only way to ensure network reliablility. Belden 8102, or equivalent is recommended for RS-422 networks. Use a terminating resistor equal in value to the characteristic impedance of the cable being used (100 Ω for Belden 8102).



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

RS-485 is a multi-point wiring standard with a practical wiring distance of 4000 feet maximum. This wiring standard provides for the possibility of up to 32 masters communicating to up to 32 slaves all within the maximum distance of 4000 feet. Note that while the RS-485 wiring standard provides for multiple masters on the same network, **the *Direct*LOGIC PLCs do not support multiple masters on a single network.** The RS-485 wiring standard does not specify a network topology, but in practice, a daisy-chain topology with the master at one end is the only way to ensure network reliablility. Belden 9841, or equivalent is recommended for RS-485 networks. Use a terminating resistor equal in value to the characteristic impedance of the cable being used (120 Ω for Belden 9841).



Communications Protocols

A communications protocol is the 'language' the devices on a network use to communicate with each other. All the devices on the network must use the same communications protocol in order to be able to communicate with each other. The protocols avalable in the *Direct*LOGIC PLCs are listed in the following table.

Communications Protocols							
Protocol	**Master**	**Slave**	**Port 1***	**Port 2**	**RS-232C**	**RS-422**	**RS-485****
K-Sequence	No	Yes	Yes	Yes	Yes	Yes	No
***Direct*NET**	Yes	Yes	Yes	Yes	Yes	Yes	No
MODBUS RTU	Yes	Yes	Yes	Yes	Yes	Yes	Yes
ASCII	Out	In	No	Yes	Yes	Yes	No

* Port 1 supports slave only and is only RS-232C with fixed communications parameters of 9600 baud, 8 data bits, 1 start bit, 1 stop bit, odd parity and station address 1. It is an asynchronous, half-duplex DTE port and auto-selects between K-Sequence, *Direct*NET and MODBUS RTU protocols.

** RS-485 is available on Port 2 for MODBUS RTU protocol only.

K-Sequence protocol is not available for use by a master DL06 PLC. Therefore, it cannot be used for networking between PLCs. Its primary use in the DL06 PLC is as a slave to *Direct*SOFT programming software and to an operator interface.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

*Direct*NET protocol is available for use by a master or by a slave DL06 PLC. This, and the fact that it is 'native' protocol, makes it ideal for PLC-to-PLC communication over a point-to-point to multipoint network usijg the RX and WX instructions.

MODBUS RTU protocol is a very common industry standard protocol, and can be used by a master or slave DL06 to communicate with a wide variety of industrial devices which support this protocol.

ASCII is another very common industry standard protocol, and is commonly used where alpha-numeric character data is to be transferred. Manyh input devices, such as, barcode readers and electronis scales use ASCII protocol, and many output devices accept ASCII commands, as well.

It doesn't matter which wiring standard or protocol is used, there are several communications parameters to select for each device before they will be able to communicate. These parameters include:

Baud Rate	Flow Control
Data Bits	Echo Suppression
Parity	Timeouts
Stop Bits	Delay Times
Station Address	Format

All of these parameters may not be necessary, or available, for your application. The parameters used will depend on the protocol being used and whether the device is a master or slave.

NOTE: The point to remember is that when there is the same parameter available in the master and in the slave (i.e. Baud Rate, Parity, Stop Bits, etc), the settings must match.

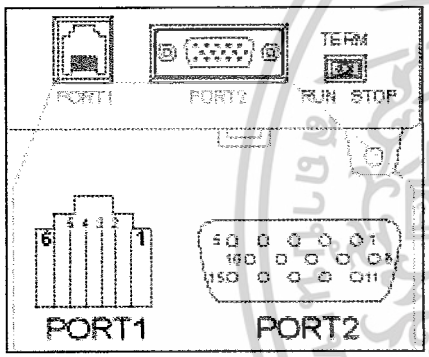
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

DL06 Port Specifications

Communications Port 1	
Port 1	Connects to HPP, *Direct*SOFT 5, operator interfaces, etc.
	6-pin, RS232C
	Communication speed (baud): 9600 (fixed)
	Parity: odd (fixed)
	Station Address: 1 (fixed)
	8 data bits
	1 start, 1 stop bit
	Asynchronous, half-duplex, DTE
	Protocol (auto-select): K-sequence (slave only), *Direct*NET (slave only), MODBUS (slave only)

Communications Port 2	
Port 2	Connects to HPP, *Direct*SOFT 5, operator interfaces, etc.
	15-pin, multifunction port, RS232C, RS422, RS485 (RS485 with 2-wire is only available for MODBUS and Non-sequence).
	Communication speed (baud): 300, 600, 1200, 2400, 4800, 9600, 19200, 38400
	Parity: odd (default), even, none
	Station Address: 1 (default)
	8 data bits
	1 start, 1 stop bit
	Asynchronous, half-duplex, DTE
	Protocols can be pre-selected: K-sequence (slave only), *Direct*NET (master/slave), MODBUS (master/slave), non-sequence/print/ASCII in/out

DL06 Port Pinouts



Port 1 Pin Descriptions		
1	0V	Power (-) connection (GND)
2	5V	Power (+) connection
3	RXD	Receive data (RS-232C)
4	TXD	Transmit data (RS-232C)
5	5V	Power (+) connection
6	0V	Power (-) connection (GND)

Port 2 Pin Descriptions		
1	5V	Power (+) connection
2	TXD	Transmit data (RS-232C)
3	RXD	Receive data (RS-232C)
4	RTS	Ready to send (RS-232C)
5	CTS	Clear to send (RS232C)
6	RXD-	Receive data (-) (RS-422/485)
7	0V	Power (-) connection (GND)
8	0V	Power (-) connection (GND)
9	TXD+	Transmit data (+) (RS-422/485)
10	TXD-	Transmit data (-) (RS-422/485)
11	RTS+	Ready to send (+) (RS-422/485)
12	RTS-	Ready to send (-) (RS-422/485)
13	RXD+	Receive data (+) (RS-422/485)
14	CTS+	Clear to send (+) (RS-422/485)
15	CTS-	Clear to send (-) (RS-422/485)

K

Note that the default configuration for port 2 is:

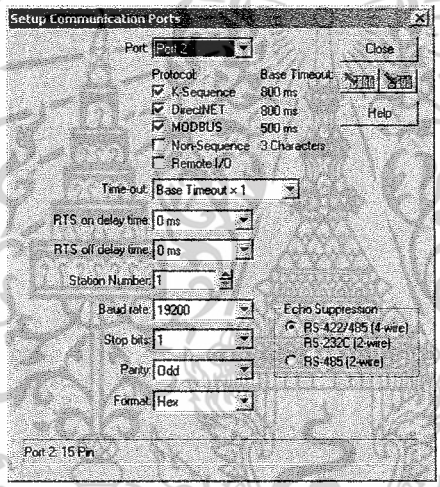
Auto-detect among K-Sequence, *Direct*NET, and MODBUS RTU protocols
Timeout = Base Timeout x 1 (800 ms)
RTS on delay time = 0 ms
RTS off delay time = 0 ms
Station Number = 1
Baud rate = 19200
Stop bits = 1
Parity = odd
Format = Hex
Echo Suppression = RS-422/485 (4-wire) or RS-232C

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Port Setup Using *Direct*SOFT 5 or Ladder Logic Instructions

Port 2 on the DL06 can be configured for communicatiojns using the various protocols which have been previously mentioned. Also, the communications parameters can be configured to match the parameters in the other device(s) with which the PLC will be communicating. The port may be configured using the *Direct*SOFT 5 PLC programming software, or by using ladder logic within the PLC program. It is important to note that the settings for port 2 are never saved to disk with *Direct*SOFT 5, so if you are using port 2 in other than it's default configuration (see page K-6) it is a good idea to include the port setup in the ladder program, typically on a first scan bit, or in an initialization subroutine.

To setup port 2 using *Direct*SOFT 5 the PLC must be turned on and connected to *Direct*SOFT 5. If the PLC Setup toolbar is displayed, either select the **Port 2** button or select **PLC > Setup > Setup Sec. Comm Port...** from the menu bar located at the top of the programming window. A dialog box like the one below will appear. Make the appropriate settings and write them to the PLC.



In order to setup port 2 in your relay ladder logic the appropriate values must be written to V7655 (Word 1), V7656 (Word 2) and V7650 (Word 3, for ASCII only) to specify the settings for the port. Then write the 'setup complete' flag (K0500) to V7657 (Word 3) to request the CPU to accept the port settings. Once the CPU sees the 'setup complete' flag in V7657 it will test the port settings which have selected for validity, and then change the value in V7657 according to the results of this test. If the port settings are valid, the CPU will change the value in V7657 to 0A00 ('A' for Accepted). If there was an error in the port settings, the CPU will change the value in V7657 to 0E00 ('E' for Error).

NOTE: This is a Helpful Hint. Rather than build the setup words manually from the tables, use DirectSOFT 5 to setup the port as desired then use a Dataview to view the setup words as BCD/HEX. Then simply use these numbers in the setup code.

The data that is written to the port setup words has two formats. The format that is used depends on whether K-Sequence, *Direct*NET, MODBUS RTU (method 1) or ASCII (method 2) is selected.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

V7656 (Word 2)	Parity	Stop Bits	Echo Suppression	Baud Rate
K-Sequence, *Direct*NET & MODBUS RTU				
ppsO ebbb xaaa aaaa	pp	s	e	bbb
	00 = None	0 = 1 bit	0 = 232 or 422	000 = 300
	10 = Odd	1 = 2 bits	1 = 485, 2 wire	001 = 600
	11 = Even			010 = 1200
				011 = 2400
				100 = 4800
				101 = 9600
				110 = 19200
				111 = 38400

K

V7656 (Word 2) cont'd	Protocol	Secondary Address
K-Sequence, *Direct*NET & MODBUS RTU	(*Direct*NET)	xaaaaaaa (*Direct*NET)
ppsO ebbb xaaa aaaa	x	_aaaaaaa (K-Seq. & MODBUS RTU)
	0 = Hex	K-Sequence: 1-90
	1 = ASCII	*Direct*NET: 1-90
		MODBUS: 1-247

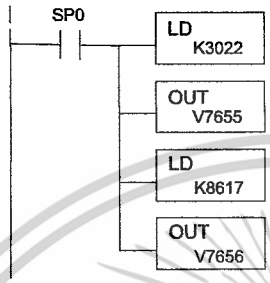
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Port 2 Setup for RLL Using K-Sequence, *Direct*NET or MODBUS RTU

V7655 (Word 1)	RTS On-delay	Timeout (% of timeout)	Protocol	RTS Off-delay
0yyy 0ttt mmmm mxx	yyy	ttt	mmmmm	xxx
	000 = 0ms	000 = 100%	10000 = K-Sequence	000 = 0ms
	001 = 2ms	001 = 120%	01000 = *Direct*NET	001 = 2ms
	010 = 5ms	010 = 150%	00100 = MODBUS RTU	010 = 5ms
	011 = 10ms	011 = 200%		011 = 10ms
	100 = 20ms	100 = 500%		100 = 20ms
	101 = 50ms	101 = 1000%		101 = 50ms
	110 = 100ms	110 = 2000%		110 = 100ms
				111 = 500ms

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

To setup port 2 for for MODBUS protocol for the following: RTS On-delay of 10ms, Base timeout x1, RTS Off-delay of 5ms, Odd parity, 1 Stop bit, echo suppression for RS232-C/RS422, 19,200 baud, Station Number 23 you would use the relay ladder logic shown below.



Port 2 Setup for RLL Using ASCII

Word 1	RTS On-delay	Timeout (in% of std. timeout)	Protocol	RTS Off-delay
0yyy 0ttt mmmm mxx	yyy	ttt	mmmmm	xxx
DL05/06: V7655	000 = 0ms	000 = 100%	00010 = Non-Sequence	000 = 0ms
	001 = 2ms	001 = 120%		001 = 2ms
	010 = 5ms	010 = 150%		010 = 5ms
	011 = 10ms	011 = 200%		011 = 10ms
	100 = 20ms	100 = 500%		100 = 20ms
	101 = 50ms	101 = 1000%		101 = 50ms
	110 = 100ms	110 = 2000%		110 = 100ms
	111 = 500ms	111 = 5000%		111 = 500ms

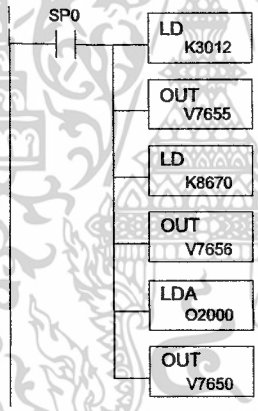
Word 2	Parity	Stop Bits	Echo Suppression (valid for DL06 only)	Baud Rate
K-Sequence, *Direct*NET & MODBUS RTU				
ppsO ebbb xaaa aaaa	pp	s	e	bbb
DL05/06: V7656	00 = None	0 = 1 bit	0 = RS-232C, RS-422 or RS-485 (4 wire)	000 = 300
	10 = Odd	1 = 2 bits	1 = RS-485 (2 wire)	001 = 600
	11 = Even			010 = 1200
				011 = 2400
				100 = 4800
				101 = 9600
				110 = 19200
				111 = 38400

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Word 2 cont'd	Protocol Mode
DL05/06: V7656	01110000 = No flow control
	01110001 = Xon/Xoff flow control
	01110010 = RTS flow control
	01110011 = Xon/Xoff and RTS flow control

Word 3	V-memory address for data
DL05/06: V7656	Hex value of the V-memory location to temporaily store the ASCII data coming into the PLC. Set this parameter to an unused V-memory location which has enough consecutive memory locations free to store the longest string that will come into the PLC.

To setup port 2 for for Non-sequence (ASCII) communications with the following: RTS On-delay of 10ms, Base timeout x1, RTS Off-delay of 5ms, Odd parity, 1 Stop bit, echo suppression for RS232-C/RS422, 19,200 baud, 8 data bits, V-memory buffer starting at V2000 and no flow control you would use the relay ladder logic shown below.



K

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

K-Sequence Communications

The K-Sequence protocol can be used for communication with *Direct*SOFT 5, an operator interface or any other device that can be a K-Sequence master. The DL06 PLC can be a K-Sequence slave on either port 1 or port 2. The DL06 PLC cannot be a K-Sequence master.

In order to use port 2 for K-Sequence communications you first need to setup the port using either *Direct*SOFT 5 or ladder logic as described above.

*Direct*NET Communications

The *Direct*NET protocol can be used to communicate to another PLC or to other devices that can use the *Direct*NET protocol. The DL06 can be either be used as a master using port 2 or as a slave using either port 1 or port 2.

In order to use port 2 for *Direct*NET communications you must first setup the port using either *Direct*SOFT 5 or ladder logic as previously described.

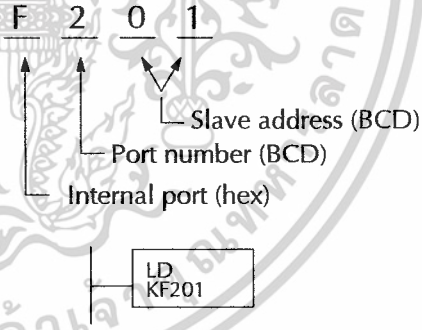
For network slave operation, nothing more needs to be done. Port 2 will function as a slave unless network communications instructions are executed by the ladder logic program.

For a network master operation you will simply need to add some ladder rungs using the network communication instructions RX and/or WX. If you use more than one network communidcation instruction is executed at any given time. If you have just a few network communications instructions in your program, you can use discrete bits to interlock them. If you are using many network communications instructions, a counter or a shift register will be a more convenient way to interlock the instructions.

The following step-by-step procedure will provide you the information necessary to set up your ladder program to receive data from a network slave.

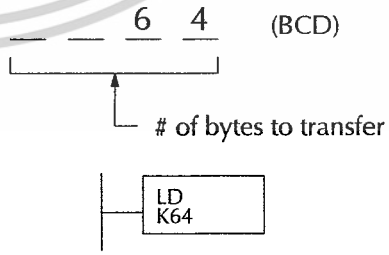
Step 1: Identify Master Port # and Slave

The first Load (LD) instruction identifies the communications port number on the network master (DL06) and the address of the slave station. This instruction can address up to 99 MODBUS slaves, or 90 *Direct*NET slaves. The format of the word is shown to the right. The "F2" in the upper byte indicates the use of the right port of the DL06 PLC, port number 2. The lower byte contains the slave address number in BCD (01 to 99).



Step 2: Load Number of Bytes to Transfer

The second Load (LD) instruction determines the number of bytes which will be transferred between the master and slave in the subsequent WX or RX instruction. The value to be loaded is in BCD format (decimal), from 1 to 128 bytes.



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

The number of bytes specified also depends on the type of data you want to obtain. For example, the DL06 Input points can be accessed by V-memory locations or as X input locations. However, if you only want X0 – X27, you'll have to use the X input data type because the V-memory locations can only be accessed in 2-byte increments. The following table shows the byte ranges for the various types of *Direct*LOGIC products.

DL05 / 06 / 205 / 350 / 405 Memory	Bits per unit	Bytes
V-memory T / C current value	16 16	2 2
Inputs (X, SP)	8	1
Outputs (Y, C, Stage, T/C bits)	8	1
Scratch Pad Memory	8	1
Diagnostic Status	8	1

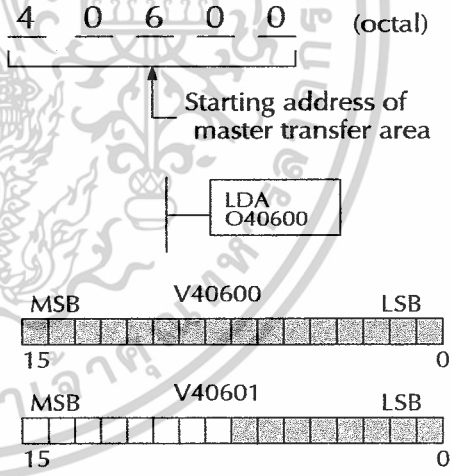
DL330 / 340 Memory	Bits per unit	Bytes
Data registers T / C accumulator	8 16	1 2
I/O, internal relays, shift register bits, T/C bits, stage bits	1	1
Scratch Pad Memory	8	1
Diagnostic Status(5 word R/W)	16	10

Step 3: Specify Master Memory Area

The third instruction in the RX or WX sequence is a Load Address (LDA) instruction. Its purpose is to load the starting address of the memory area to be transferred. Entered as an octal number, the LDA instruction converts it to hex and places the result in the accumulator.

For a WX instruction, the DL06 CPU sends the number of bytes previously specified from its memory area beginning at the LDA address specified.

For an RX instruction, the DL06 CPU reads the number of bytes previously specified from the slave, placing the received data into its memory area beginning at the LDA address specified.

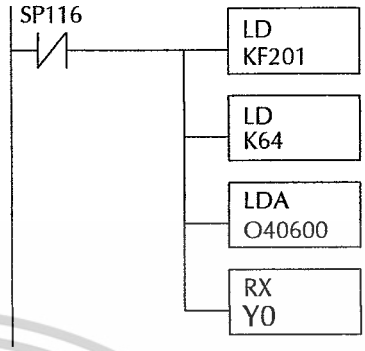


NOTE: Since V-memory words are always 16 bits, you may not always use the whole word. For example, if you only specify 3 bytes and you are reading Y outputs from the slave, you will only get 24 bits of data. In this case, only the 8 least significant bits of the last word location will be modified. The remaining 8 bits are not affected.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Step 4: Specify Slave Memory Area

The last instruction in our sequence is the WX or RX instruction itself. Use WX to write to the slave, and RX to read from the slave. All four of our instructions are shown to the right. In the last instruction, you must specify the starting address and a valid data type for the slave.



- *Direct*NET slaves – specify the same address in the WX and RX instruction as the slave's native I/O address
- MODBUS DL405, DL205, or DL06 slaves – specify the same address in the WX and RX instruction as the slave's native I/O address
- MODBUS 305 slaves – use the following table to convert DL305 addresses to MODBUS addresses

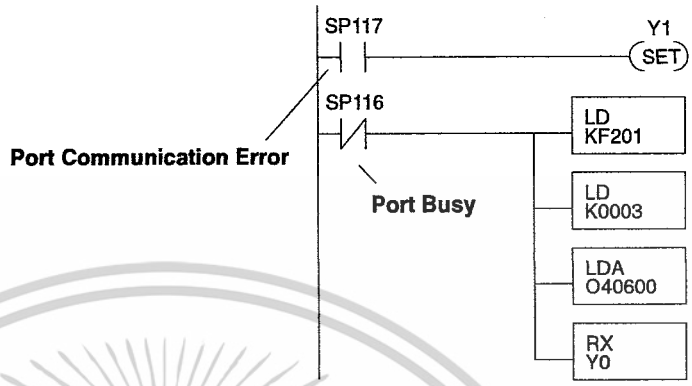
DL305 Series CPU Memory Type to MODBUS Cross Reference (excluding 350 CPU)

PLC Memory Type	PLC Base Address	MODBUS Base Address	PLC Memory Type	PLC Base Address	MODBUS Base Address
TMR/CNT Current Values	R600	V0	TMR/CNT Status Bits	CT600	GY600
I/O Points	IO 000	GY0	Control Relays	CR160	GY160
Data Registers	R401,R400	V100	Shift Registers	SR400	GY400
Stage Status Bits (D3-330P only)	S0	GY200			

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Communications from a Ladder Program

Typically network communications will last longer than 1 scan. The program must wait for the communications to finish before starting the next transaction.



Port 2, which can be a master, has two Special Relay contacts associated with it (see Appendix D for comm port special relays).One indicates "Port busy"(SP116), and the other indicates "Port Communication Error"(SP117). The example above shows the use of these contacts for a network master that only reads a device (RX). The "Port Busy" bit is on while the PLC communicates with the slave. When the bit is off the program can initiate the next network request.

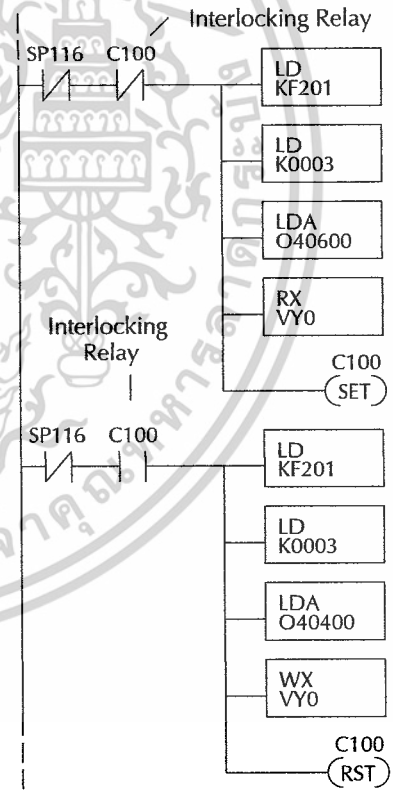
The "Port Communication Error" bit turns on when the PLC has detected an error. Use of this bit is optional. When used, it should be ahead of any network instruction boxes since the error bit is reset when an RX or WX instruction is executed.

Multiple Read and Write Interlocks

If you are using multiple reads and writes in the RLL program, you have to interlock the routines to make sure all the routines are executed. If you don't use the interlocks, then the CPU will only execute the first routine. This is because each port can only handle one transaction at a time.

In the example to the right, after the RX instruction is executed, C100 is set. When the port has finished the communication task, the second routine is executed and C100 is reset.

If you're using RLL^{PLUS} Stage Programming, you can put each routine in a separate program stage to ensure proper execution and switch from stage to stage allowing only one of them to be active at a time.



K

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

MODBUS RTU Communications

The MODBUS RTU protocol can be used for communication with any device that uses the MODBUS RTU protocol. The protocol is very common and is probably the closest thing to an "industry standard" protocol in existence. The DL06 can be a MODBUS RTU slave on either port 1 or port 2, and it can be a MODBUS RTU master on port 2. The RS 485 wiring standard may be used on port 2 for the MODBUS RTU protocol only.

In order to use port 2 for MODBUS RTU communications you must first set up the port using either *Direct*SOFT 5 or ladder logic as previously described.

For network slave operation, nothing more needs to be done. Port 2 will function as a slave unless network communications instructions are executed by the ladder logic program.

For network master operation the MODBUS RTU network communication instructions MRX and/or MWX needs to be added to some ladder rungs. If more than one network communication instruction is used, the rungs need to be interlocked to ensure that only one communication instruction is executed at any given time. If only a few network communications instructions are used in your program, discrete bits can be used to interlock them. If many network communicaitons instructions are used, either a counter or a shift register will be a more convenient way to interlock the instructions.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

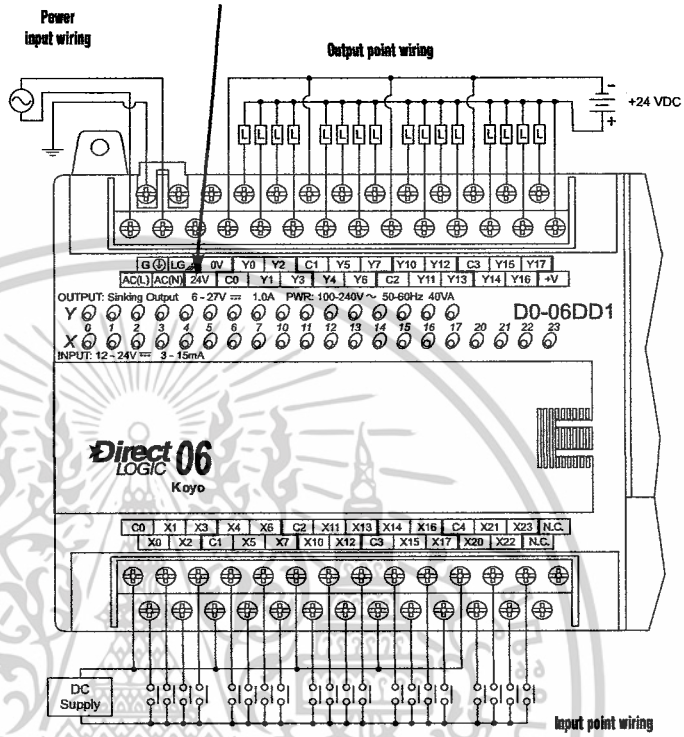
DL06 I/O Specifications

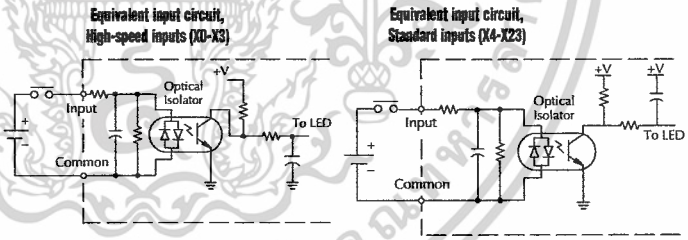
D0-06DD1 <--->

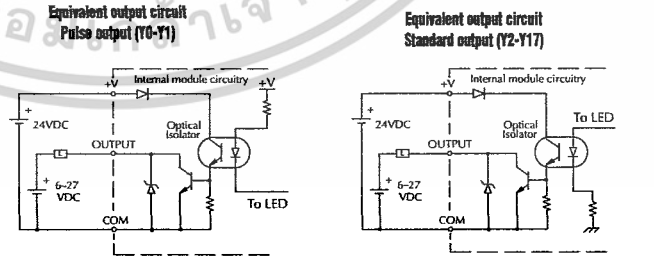
Note: Refer to page 2–29, Power Budgeting, for Auxillary 24VDC current available.

Wiring diagram and specifications

D0-06DD1 Specifications			
AC Power Supply Specifications	Voltage Range	95-240VAC (30VA)	
DC Input Specifications	Number of Input Pts.	20 (sink/source)	
	Number of Commons	5 (isolated)	
	Input Voltage Range	12-24VDC	
	Input Impedance	(X0-X3) 1.8K @ 12-24VDC (X4-X23) 2.8K @ 12-24VDC	
	On Current/ Voltage Level	>5mA/10VDC	
	OFF Current/ Voltage Level	<0.5mA/<2VDC	
	Response Time	X0-X3	X4-X23
	OFF to ON Response	<100µs	<8ms
	ON to OFF Response	<100µs	<8ms
	Fuses	None	
DC Output Specifications	Number of Output Points	16 (sinking)	
	Number of Commons	4 isolated	
	Output Voltage Range	6-27VDC	
	Peak Voltage	50VDC	
	Max. Frequency (Y0, Y1)	7kHz	
	ON Voltage Drop	0.3VDC @ 1A	
	Maximum Current	0.5A / pt (Y0-Y1)* 1.0A pt (Y2-Y17)**	
	Maximum Leakage Current	15µA @ 30VDC	
	Maximum Inrush Current	2A for 100ms	
	OFF to ON Response	<10µs	
	ON to OFF Response	<20µs (Y0-Y1) <60µs (Y2-Y17)	
	External DC Power Required	20-28VDC 150mA max. (Y0-Y1) 280 mA max. (Y2-Y17)	
	Status Indicators	Logic side	
	Fuses	None (external recommended)	



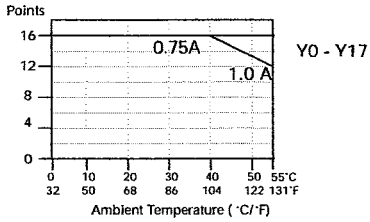




* When Y0-Y1 are not used for pulse outputs, maximum current output is 1.0A**.

** These outputs must be derated to 0.6A for EN61131-2 compliance.

Derating chart for DC outputs



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Features at a Glance

The DL05 and DL06 micro PLCs are complete self-contained systems. The CPU, power supply, and I/O are all included inside the same housing. Option modules are available to expand the capability of each PLC family for more demanding applications. The standard features of these PLCs are extraordinary and compare favorably with larger and more expensive PLCs.

The specification tables to the right are meant for quick reference only. Detailed specifications and wiring information for each model of the DL05 and DL06 PLCs begin on page 2–33.

Program capacity

Most boolean ladder instructions require a single word of program memory. Other instructions, such as timers, counters, etc., require two or more words. Data is stored in V-memory in 16-bit registers.

Performance

The performance characteristics shown in the tables represent the amount of time required to read the inputs, solve the Relay Ladder Logic program and update the outputs.

Instructions

A complete list of instructions is available at the end of this section.

Communications

The DL05 and DL06 offer powerful communication features normally found only on more expensive PLCs.

Special features

The DC input and DC output PLCs offer high-speed counting or pulse output. Option module slots allow for discrete I/O expansion, analog I/O, or additional communication options.

DL05 CPU Specifications

System capacity	
Total memory available (words)	6K
Ladder memory (words)	2,048
V-memory (words)	4,096
User V-memory	3,968
Non-volatile user V-memory	128
Battery backup	Yes[1]
Total built-in I/O	14
Inputs	8
Outputs	6
I/O expansion	Yes[1]
Performance	
Contact execution (Boolean)	0.7µs
Typical scan (1K Boolean)[2]	1.5-3ms.
Instructions and diagnostics	
RLL ladder style	Yes
RLLPLUS/flowchart style (Stages)	Yes/256
Run-time editing	Yes
Scan	Variable/fixed
Number of Instructions	133
Types of Instructions:	
Control relays	512
Timers	128
Counters	128
Immediate I/O	Yes
Subroutines	Yes
For/next loops	Yes
Timed interrupt	Yes
Integer math	Yes
Floating-point math	No
PID	Yes
Drum sequencers	Yes
Bit of word	Yes
ASCII print	Yes
Real-time clock/calendar	Yes[1]
Internal diagnostics	Yes
Password security	Yes
System and user error log	No
Communications	
Built-in ports	Two RS-232C
Protocols supported:	
K-sequence (proprietary protocol)	Yes
DirectNet master/slave	Yes
Modbus RTU master/slave	Yes
ASCII out	Yes
Baud rate	
Port 1	9,600 baud (fixed)
Port 2	selectable 300-38,400 baud (default 9,600)
Specialty Features	
Filtered inputs	Yes[3]
Interrupt input	Yes[3]
High speed counter	Yes, 5kHz[3]
Pulse output	Yes, 7kHz[3]
Pulse catch input	Yes[3]

1- These features are available with use of certain option modules. Option module specifications are located later in this section.

2- Our 1K program includes contacts, coils, and scan overhead. If you compare our products to others, make sure you include their scan overhead.

3- Input features only available on units with DC inputs and output features only available on units with DC outputs.

DL06 CPU Specifications

System capacity	
Total memory available (words)	14.8K
Ladder memory (words)	7680
V-memory (words)	7616
User V-memory	7488
Non-volatile user V-memory	128
Built-in battery backup (D2-BAT-1)	Yes
Total I/O	36
Inputs	20
Outputs	16
I/O expansion	Yes1
Performance	
Contact execution (Boolean)	0.6µs
Typical scan (1K Boolean)2	1-2ms.
Instructions and diagnostics	
RLL ladder style	Yes
RLLPLUS/flowchart style (Stages)	Yes/1024
Run-time editing	Yes
Scan	Variable/fixed
Number of Instructions	229
Types of Instructions:	
Control relays	1024
Timers	256
Counters	128
Immediate I/O	Yes
Subroutines	Yes
For/next loops	Yes
Table functions	Yes
Timed interrupt	Yes
Integer math	Yes
Trigonometric functions	Yes
Floating-point math	Yes
PID	Yes
Drum sequencers	Yes
Bit of word	Yes
Number type conversion	Yes
ASCII in, out, print	Yes
LCD instruction	Yes
Real-time clock/calendar	Yes
Internal diagnostics	Yes
Password security	Yes
System and user error log	No
Communications	
Built-in ports:	One RS-232C
	One multi-function RS232C/RS422/RS485
NOTE: RS485 is for MODBUS RTU only.	
Protocols supported:	
K-sequence (proprietary protocol)	Yes
DirectNet master/slave	Yes
Modbus RTU master/slave	Yes
ASCII in/out	Yes
Baud rate	
Port 1	9,600 baud (fixed)
Port 2	selectable 300-38,400 baud (default 9,600)
Specialty Features	
Filtered inputs	Yes[3]
Interrupt input	Yes[3]
High speed counter	Yes, 7kHz[3]
Pulse output	Yes, 10kHz[3]
Pulse catch input	Yes[3]

1- These features are available with use of certain option module. Option module specifications are located later in this section.

2- Our 1K program includes contacts, coils, and scan overhead. If you compare our products to others, make sure you include their scan overhead.

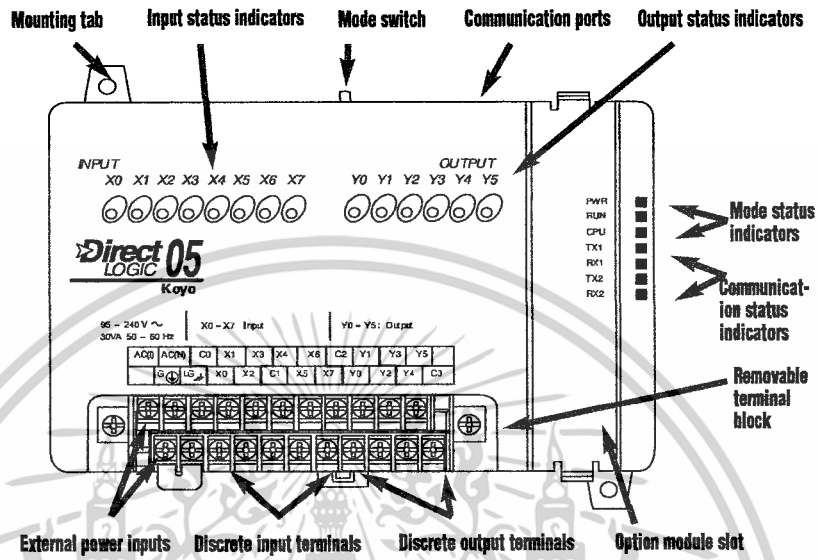
3- Input features only available on units with DC inputs and output features only available on units with DC outputs.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

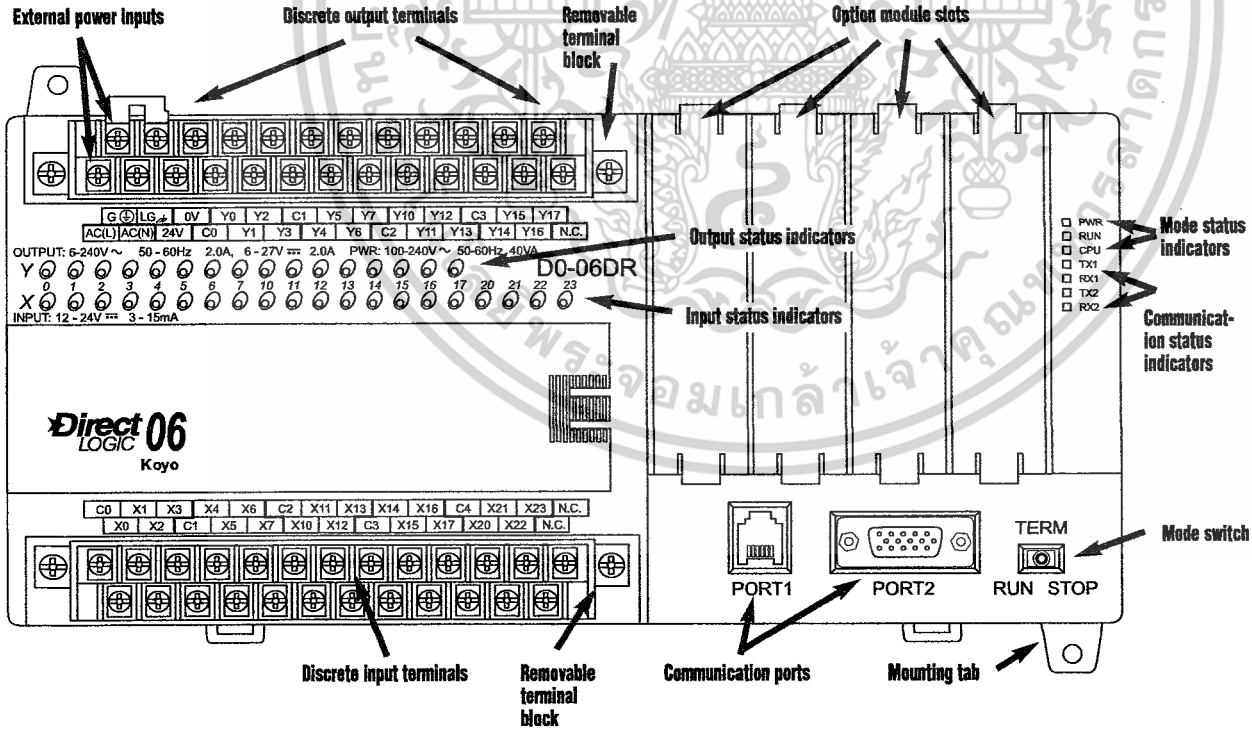
Features at a Glance

*Direct*SOFT software

The DL05 and DL06 PLCs use the same familiar *Direct*SOFT programming software that our larger PLCs use. A FREE version of *Direct*SOFT gives you all the great features of the full version, but with a 100-word PLC program download limitation. For programs larger than 100 words, the full package is required. The FREE PC-DS100 software may be sufficient to program the DL05 and DL06. If you are programming with a full package version prior to v5.0, you will need v2.4 or later for the DL05 PLCs and v4.0 or later for the DL06. We always recommend the latest version for the most robust features. See the Software section in this catalog for a complete description of *Direct*SOFT including features, part numbers of programming packages and upgrades.



Hardware features diagrams



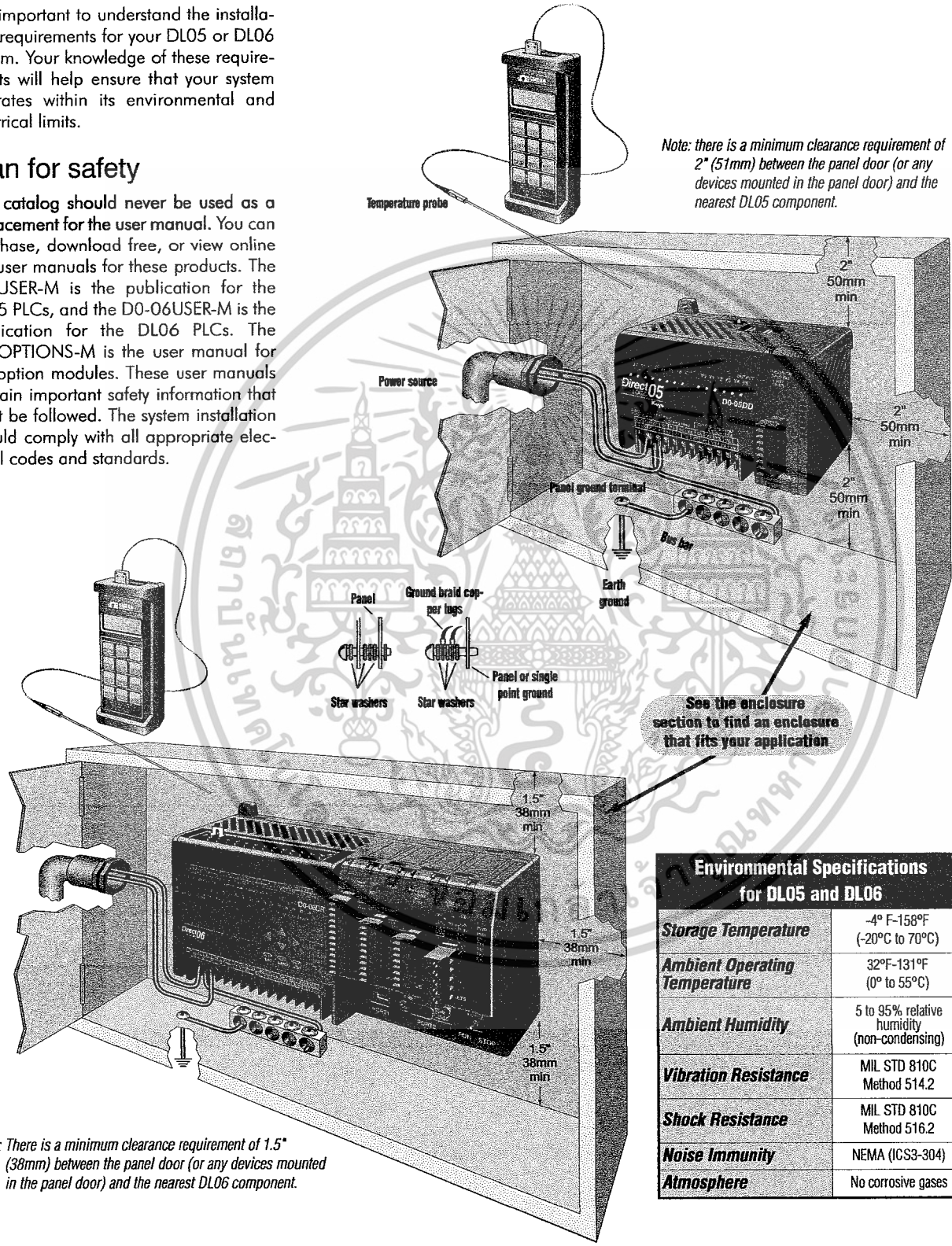
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Product Dimensions and Installation

It is important to understand the installation requirements for your DL05 or DL06 system. Your knowledge of these requirements will help ensure that your system operates within its environmental and electrical limits.

Plan for safety

This catalog should never be used as a replacement for the user manual. You can purchase, download free, or view online the user manuals for these products. The D0-USER-M is the publication for the DL05 PLCs, and the D0-06USER-M is the publication for the DL06 PLCs. The D0-OPTIONS-M is the user manual for the option modules. These user manuals contain important safety information that must be followed. The system installation should comply with all appropriate electrical codes and standards.



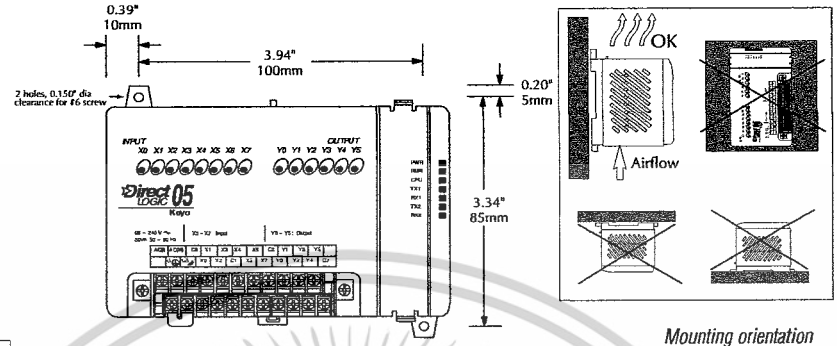
Environmental Specifications for DL05 and DL06	
Storage Temperature	-4° F-158°F (-20°C to 70°C)
Ambient Operating Temperature	32°F-131°F (0° to 55°C)
Ambient Humidity	5 to 95% relative humidity (non-condensing)
Vibration Resistance	MIL STD 810C Method 514.2
Shock Resistance	MIL STD 810C Method 516.2
Noise Immunity	NEMA (ICS3-304)
Atmosphere	No corrosive gases

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

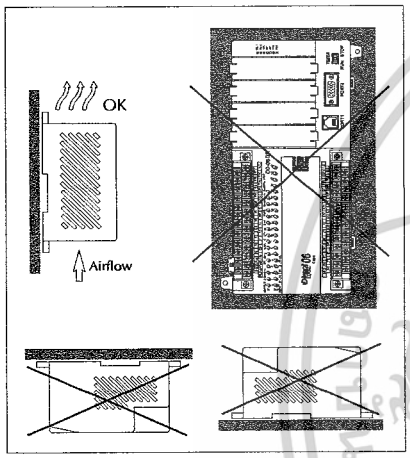
Product Dimensions and Installation

Unit dimensions and mounting orientation

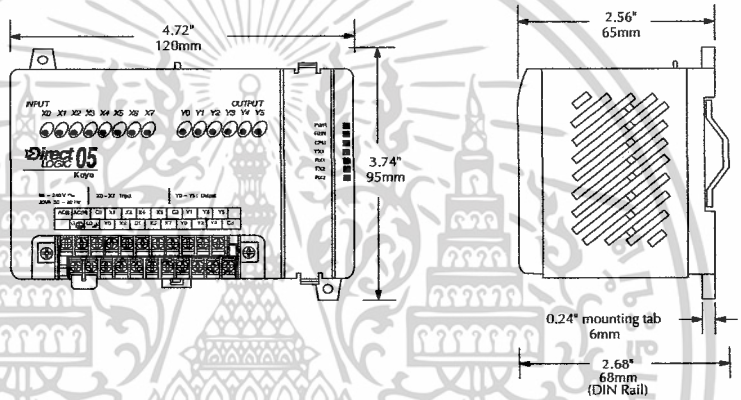
DL05 and DL06 PLCs must be mounted properly to ensure ample airflow for cooling purposes. It is important to follow the unit orientation requirements and to verify that the PLC's dimensions are compatible with your application. Notice particularly the grounding requirements and the recommended cabinet clearances.

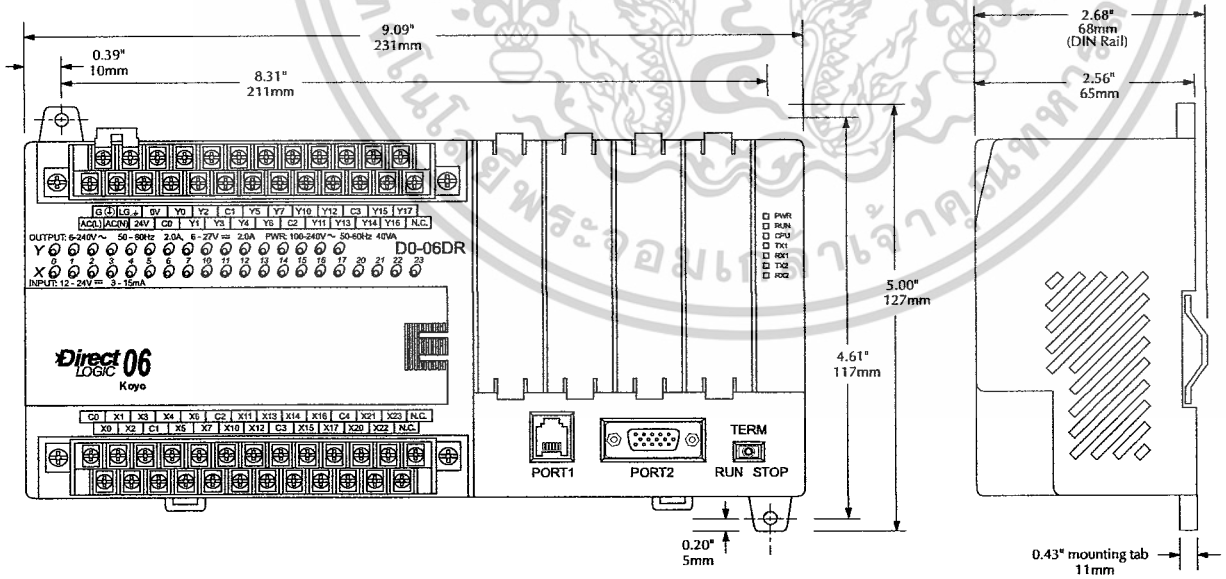


Mounting orientation



Mounting orientation





เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Choosing the I/O Type

DL06 Base Unit I/O Table

Part Number	Inputs			Outputs			Price
	I/O Type/ Commons	Sink or source	Voltage Ranges	I/O Type/ Commons	Sink or Source	Voltage/Current Ratings	
D0-06AA	AC/5	N/A	90-120VAC	AC/4	N/A	17-240VAC, 0.5A 50/60 Hz	<—>
D0-06AR	AC/5	N/A	90-120VAC	Relay/4	N/A	6-27VDC, 2A 6-240VAC, 2A	<—>
D0-06DA	DC/5	Sink or source	12-24VDC	AC/4	N/A	17-240VAC, 0.5A 50/60Hz	<—>
D0-06DD1	DC/5	Sink or source	12-24VDC	DC/4	Sink	6-27VDC, 0.5A (Y0-Y1) 6-27VDC, 1.0A (Y2-Y17)*	<—>
D0-06DD2	DC/5	Sink or source	12-24VDC	DC/4	Source	12-24VDC, 0.5A (Y0-Y1) 12-24VDC, 1.0A (Y2-Y17)	<—>
D0-06DR	DC/5	Sink or source	12-24VDC	Relay/4	N/A	6-27VDC, 2A 6-240VAC, 2A	<—>
D0-06DD1-D	DC/5	Sink or source	12-24VDC	DC/4	Sink	6-27VDC, 0.5A (Y0-Y1) 6-27VDC, 1.0A (Y2-Y17)*	<—>
D0-06DD2-D	DC/5	Sink or source	12-24VDC	DC/4	Source	12-24VDC, 0.5A (Y0-Y1) 12-24VDC, 1.0A (Y2-Y17)	<—>
D0-06DR-D	DC/5	Sink or source	12-24VDC	Relay/4	N/A	6-27VDC, 2A 6-240VAC, 2A	<—>

*** These outputs must be derated to 0.6A for EN61131-2 compliance.**

Discrete I/O Option Moduless

Part Number	Inputs			Outputs			Price
	I/O Type/ Number/ Commons	Sink or source	Voltage Ranges	I/O Type/ Number/ Commons	Sink or Source	Voltage/Current Ratings	
D0-07CDR	DC/4/1	Sink or source	12-24VDC	Relay/3/1	N/A	6-27VDC, 1A 6-240VAC, 1A	<—>
D0-08CDD1	DC/4/2	Sink or source	12-24VDC	DC/4/2	Sink	6-27VDC, 0.3A	<—>
D0-08TR	N/A	N/A	N/A	Relay/8/2	N/A	6-27VDC, 1A 6-240VAC, 1A	<—>
D0-10ND3	DC/10/2	Sink or source	12-24VDC	N/A	N/A	N/A	<—>
D0-10ND3F	DC/10/2	Sink or source	12-24VDC	N/A	N/A	N/A	<—>
D0-10TD1	N/A	N/A	N/A	DC/10/2	Sink	6-27VDC, 0.3A	<—>
D0-10TD2	N/A	N/A	N/A	DC/10/2	Source	12-24VDC, 0.3A	<—>
D0-16ND3	DC/16/4	Sink or source	20-28VDC	N/A	N/A	N/A	<—>
D0-16TD1	N/A	N/A	N/A	DC/16/2	Sink	6-27VDC, 0.1A	<—>
D0-16TD2	N/A	N/A	N/A	DC/16/2	Source	12-24VDC, 0.1A	<—>
F0-04TRS	N/A	N/A	N/A	Relay/4/4	N/A	5-30VDC, 3A 5-125VAC, 3A	<—>
F0-08NA-1	AC/8/2	N/A	80-132VAC 90-150VDC	N/A	N/A	N/A	<—>
F0-08SIM	8-pt. Input simulator						<—>

Communications and Specialty Option Modules

Part Number	Description	Price
H0-ECOM	Ethernet Communications Module 10 Mbit	<—>
H0-ECOM100	Ethernet Communications Module 10/100 Mbit	<—>
D0-DEVNETS	DeviceNET Slave Module	<—>
H0-CTRIO	High Speed Counter I/O Module	<—>
H0-PSCM	Profibus Slave Communications Module	<—>
D0-DCM	Serial Communications Module	<—>
F0-CP128	ASCII CoProcessor Module	<—>

Analog I/O

By using option modules, you can add analog inputs or outputs to your DL05 or DL06 PLC. The table below shows the input and output types at a glance. Detailed specifications are provided later in this section.

Analog I/O Option Modules

Part Number	Inputs		Outputs		Price
	No.	Input Type	No.	Output Type	
F0-04AD-1	4	0-20mA or 4-20mA	0	N/A	<—>
F0-04AD-2	4	0-5VDC or 0-10VDC	0	N/A	<—>
F0-08ADH-1	8	0-20mA	0	N/A	<—>
F0-08ADH-2	8	0-5VDC or 0-10VDC	0	N/A	<—>
F0-04DAH-1	0	N/A	4	4-20mA	<—>
F0-08DAH-1	0	N/A	8	4-20mA	<—>
F0-04DAH-2	0	N/A	4	0-10VDC	<—>
F0-08DAH-2	0	N/A	8	0-10VDC	<—>
F0-4AD2DA-1	4	0-20mA or 4-20mA	2	0-20mA or 4-20mA	<—>
F0-2AD2DA-2	2	0-5VDC or 0-10VDC	2	0-5VDC or 0-10VDC	<—>
F0-4AD2DA-2	4	0-5VDC or 0-10VDC	2	0-5VDC or 0-10VDC	<—>
F0-04RTD	4	RTD	0	N/A	<—>
F0-04THM*	4	Thermo-couple / Voltage	0	N/A	<—>

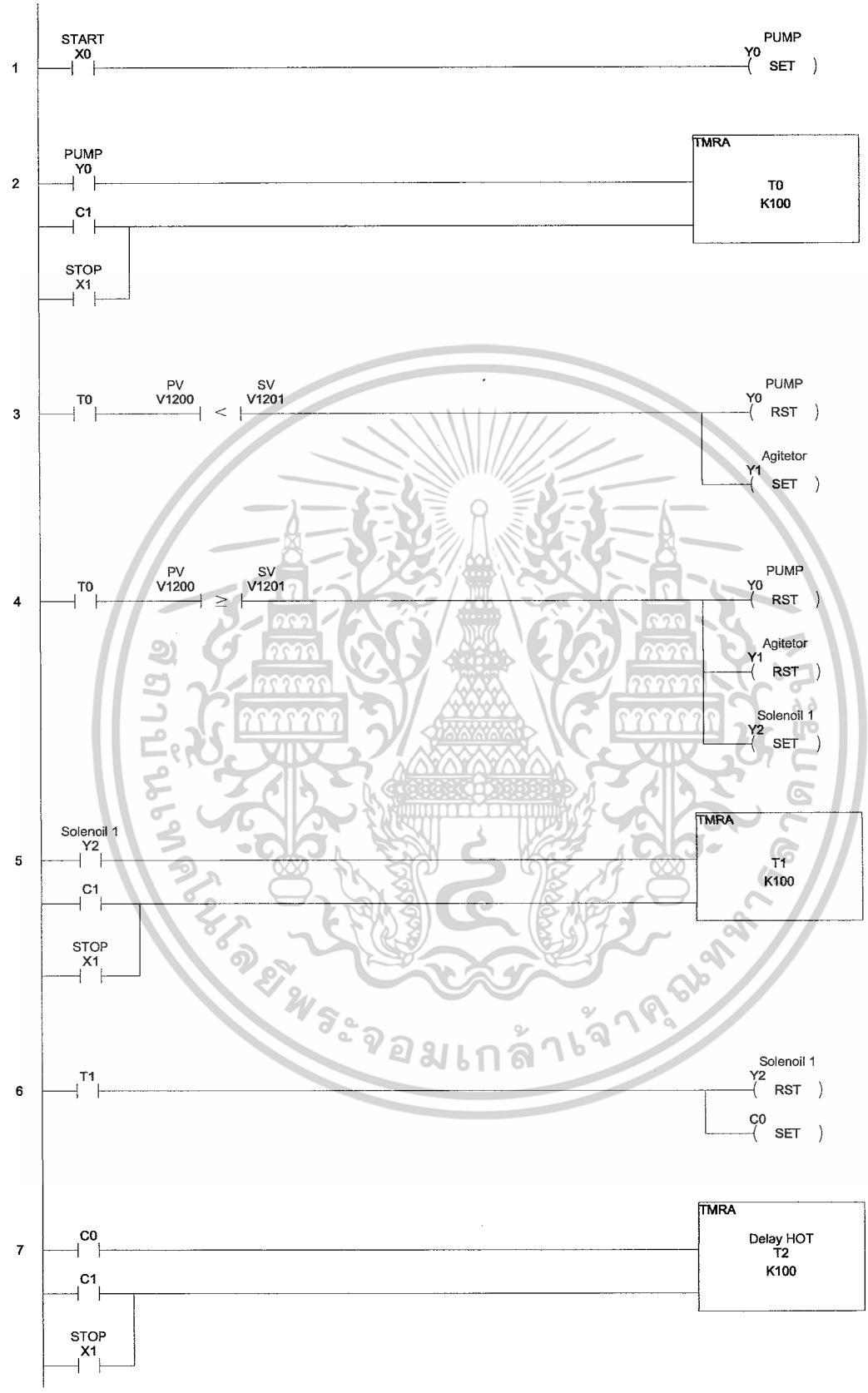
*** See module specifications page for thermocouple types and voltage input ranges supported**

Power budgeting

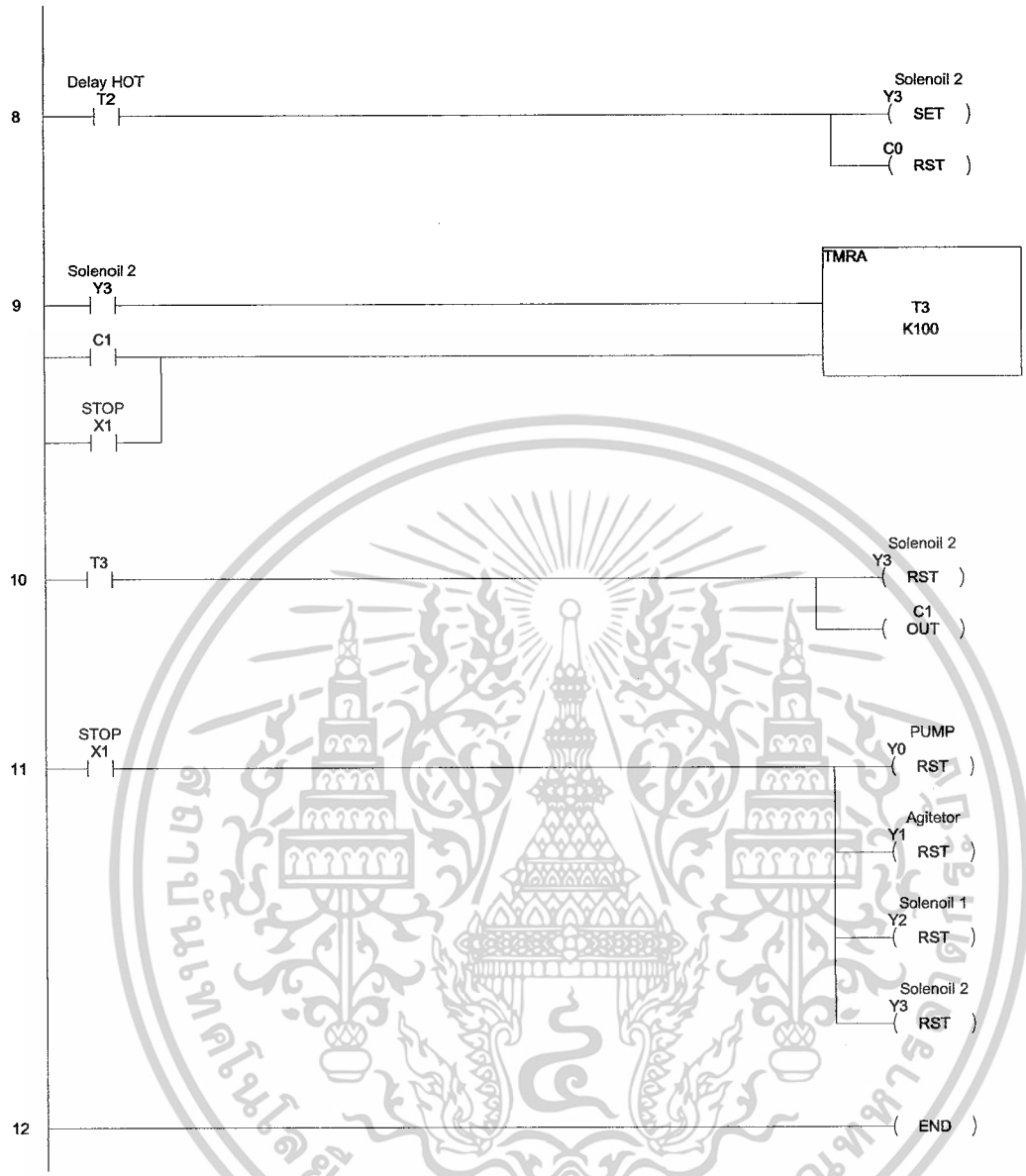
No power budgeting is necessary for the DL05. The built-in power supply is sufficient for powering the base unit, any of the option modules, the hand-held programmer, and even a DV1000 operator interface.

Power budgeting is necessary for the DL06. With four option module slots and an optional LCD display, it is necessary to verify that sufficient power is available for all optional devices. Power budgeting is described in detail on page 2-29 and in the DL06 User Manual.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



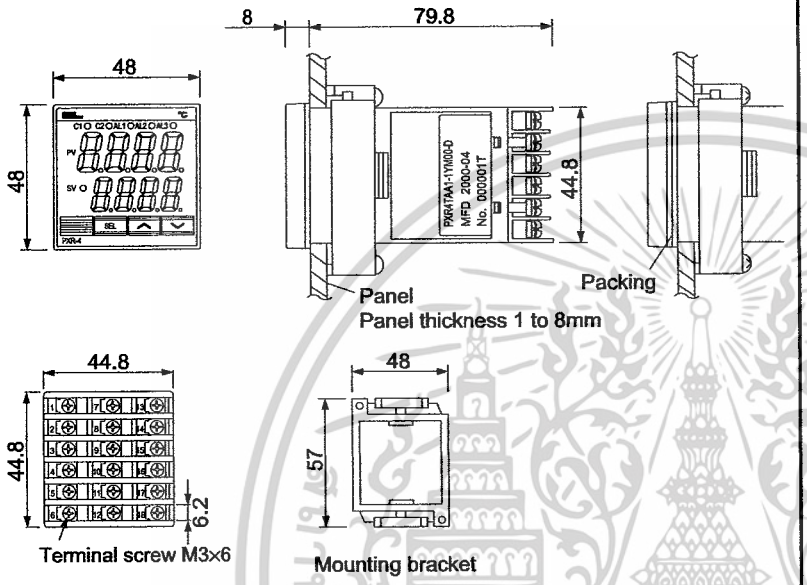
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

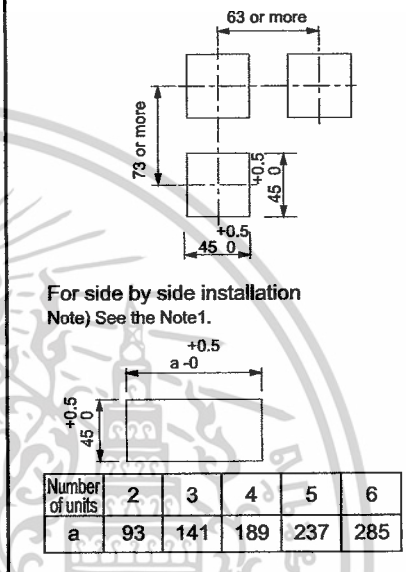
1 Installation/mounting

Outline and Panel Cutout Dimensions (Standard type/ Waterproof type)

(Unit : mm)

Outline dimensions	Panel cutout dimensions





Number of units	2	3	4	5	6
a	93	141	189	237	285

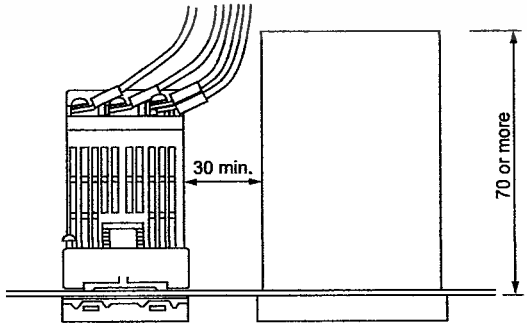
Cautions on wiring

- Wiring should be started from the left side terminal (No. 1 to No. 6).
- Use crimped terminals matched to the screw size. Tightening torque should be 0.8 Nm (Since the case is made of plastic, do not tighten excessively).
- Do not connect anything to terminals not used.

Note 1

Caution on side by side installation

- With the power supply of 200 VAC or more, a maximum ambient temperature is 45°C. (It is recommended to use a fan for cooling.)

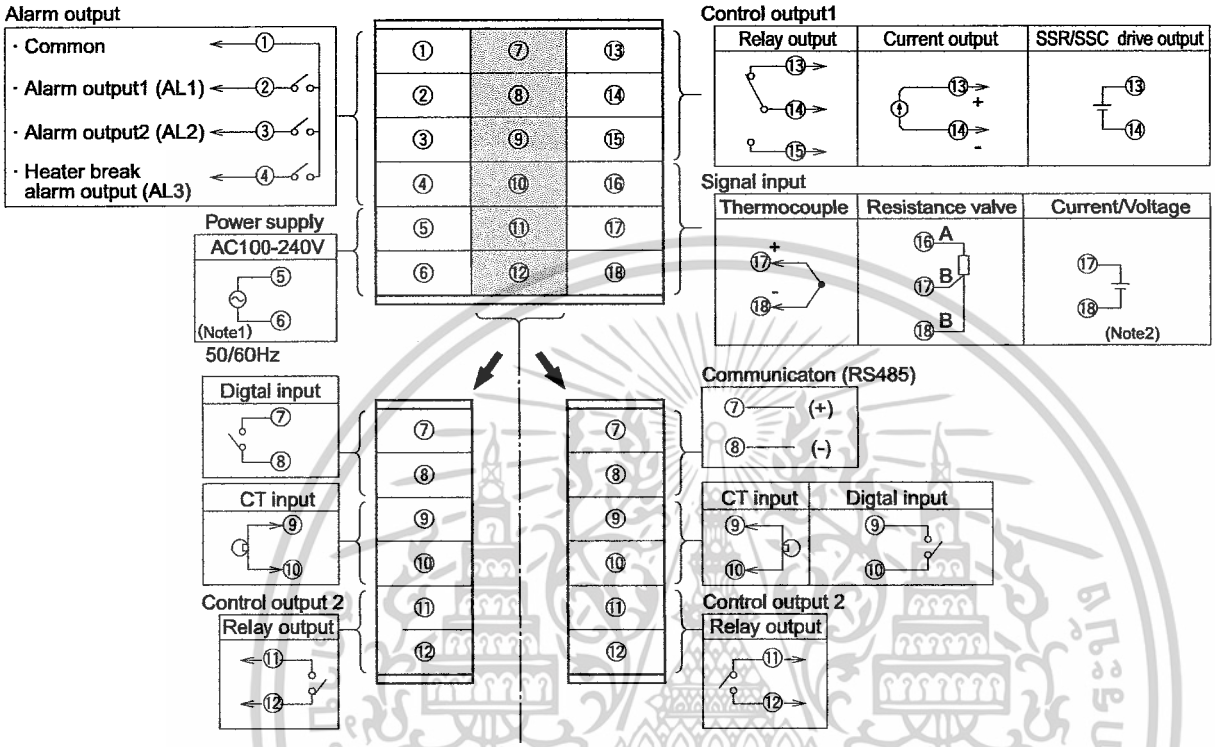


When there is another instrument (larger than 70mm) or a wall on the right side of this controller, be sure to install the controller keeping a space of more than 30mm.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2 Wiring

Terminal Connection Diagram (100 to 240 Vac)



Note1 : Check the power supply voltage before installation.
Note2 : Connect the I/V unit (250Ω resistor) (accesory) between the terminal ⑰ and ⑱ in case of currert input.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

PXR Series

Advanced Self-Tuning Temperature and Process Controllers

The new PXR series controllers are the newest additions to Fuji Electric's trusted line of temperature and process controllers. They are now packed with more features and options than before, and the new low price is unbeatable.

Key Features

- **PID with fuzzy control of self-tuning**
- **16 Ramp/soak segments**
- **Large LED display**
- **Digital input**
- **Auto-tune**
- **Timer function**
- **Heater burnout alarm**
- **Replaces most other controllers in market**
- **Excellent for after-market support**

They come in several sizes – 1/32 DIN, 1/16 DIN, 1/8 DIN and 1/4 DIN. These controllers have all the standard features that you expect from Fuji Electric's superior controllers, and more. In addition to auto-tuning and fuzzy control, it now comes with self-tuning — an innovation in the control field. It automatically retunes the controller under certain conditions, without the need to revert to auto-tuning. The standard 8-segment ramp/soak feature has been expanded to include two patterns that can be linked to create a 16-step profile. The PXR accepts temperature and process inputs and offers a choice of three kinds of outputs to meet a wide variety of needs in the process industries. Low-cost options include dual outputs, programmable alarms, remote setpoint, RS485 communications, analog retransmission, digital input, timer function, heater burnout alarm and 24V AC/DC supply voltage.

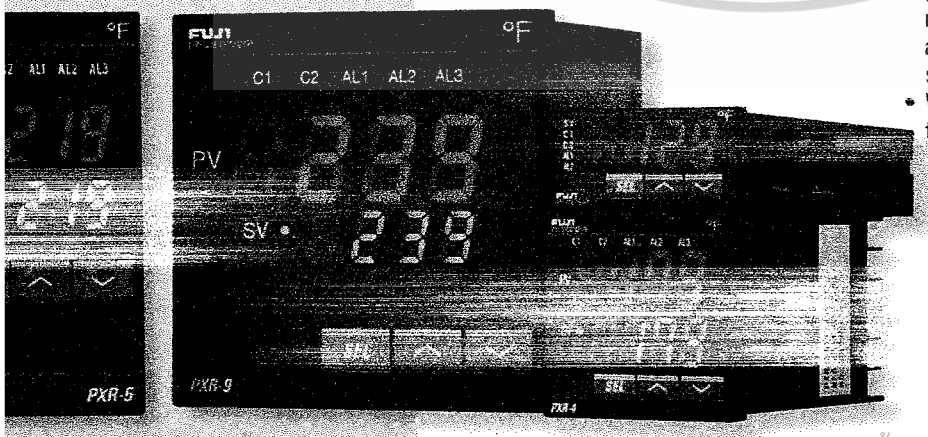
One of the most impressive features is the large LED display. The faceplate, designed for NEMA 4X (IP66 equivalent) is watertight and corrosion-resistant. The easy-to-use 3-button keypad allows for programming similar to the popular PXW controller. The screw-terminal on the back further reduces the cost by eliminating the need for sockets. The PXR3 can be DIN-rail mounted with the optional adapter.

Remote monitoring of up to 31 controllers at a time is possible with the RS485 option that uses the industry-standard Modbus™ protocol. The communications option comes with our free Windows®-based software, PXR-LITE™. The software allows you to program the controller from the PC and view real-time data and trend graph while logging the data into a text file.

A powerful tool for the OEM customer is the Program Loader option with Windows®-based software. Programs for different applications can be saved to and from the controller.

Features

- **Advanced control functions** PID Plus Self Tuning; PID Plus Fuzzy Control; Auto-tuning
- **NEMA 4X faceplate with large LED display** 4-digit, red and green display; Waterproof faceplate conforms to NEMA-4X/IP66
- **Multiple inputs** Choose between thermo-couple/RTD and 4-20mA/0-5V inputs
- **Single or dual control outputs** Relay, SSR driver or 4-20mA
- **Ramp/soak function** Up to 16 ramp/soak segments or two 8-segment patterns, a standard feature
- **Programmable alarms option** 2 programmable SPST relays with On/Off delay function
- **Remote setpoint option** Change setpoint with a 1-5V signal
- **Analog retransmission option** 4-20mA retransmission of PV, SV, MV, DV
- **Digital input option** Change between 2 setpoints; Change between ramp/soak and standby; Start/reset the ramp/soak; Start/stop the auto tuning; Cancel the alarm latch; Start the incorporated timer
- **Timer function** On-delay or off-delay timer activated with digital input; Up to 2 timer outputs can be obtained
- **Heater burnout alarm option** If heater burns out, alarm goes off
- **Communications option** RS485 (ModbusTM protocol) interface permits remote monitoring of up to 31 units from a PC. Comes with free Windows®-based software, PXR-LITE™
- **Warranty** Manufactured in a ISO 9001 facility and backed by a 3-year warranty



เอกสารนี้จัดทำขึ้นไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Specifications

General Specifications

Power supply voltage 100 (-15%) to 240V (+10%) AC, 50/60Hz; 24V (±10%) AC/DC

Power consumption PXR3: 6VA (100 V AC), 8VA (220V, 24V). PXR4: 8VA (100V), 10VA (220V), 12VA (24V). PXR5, 9: 10VA (100V), 12VA (220V, 24V)

Reference junction compensation accuracy ±1°C at 23°C

Input

Input signal Thermocouple: J, K, R, B, S, T, E, N, PL2. RTD: Pt100. Voltage, current. For 1 to 5V/4 to 20 mA DC, 0 to 5V/0 to 20 mA DC, use 250 ohm shunt resister included

Input filter 0 to 900.0 sec set in 0.5 sec steps

Burnout For thermocouple or RTD input, control output direction (upper or lower) is selectable

Control Function

Control action On/Off; PID control (with auto-tuning, self-tuning); Fuzzy Control (with auto-tuning)

Proportional band (P) 0 to 999.9% of measuring range set in 0.1% steps

Integral time (I) 0 to 3200 sec set in 1 sec steps

Differential time (D) 0 to 999.9 sec set in 1 sec steps

Proportional cycle 1 to 150 sec set in 1 sec steps

Hysteresis width 0 to 50% of measuring range; For on/off action only

Input sampling cycle 0.5 sec

Control Output 1 (select one)

Relay contact PXR4, 5, 9: SPDT, 220 V AC/30 V DC, 3A (resistive load). PXR3: SPST contact, 220 V AC/30 V DC, 3A (resistive load)

SSR PXR4, 5, 9: ON–17 to 25 V DC; OFF–0.5 V DC or less. PXR3: 12 to 16 V DC. Max. current: 20mA or less

4 to 20 mA DC PXR4, 5, 9: Allowable load resistance 600 ohms or less. PXR3: 100 to 500 ohms

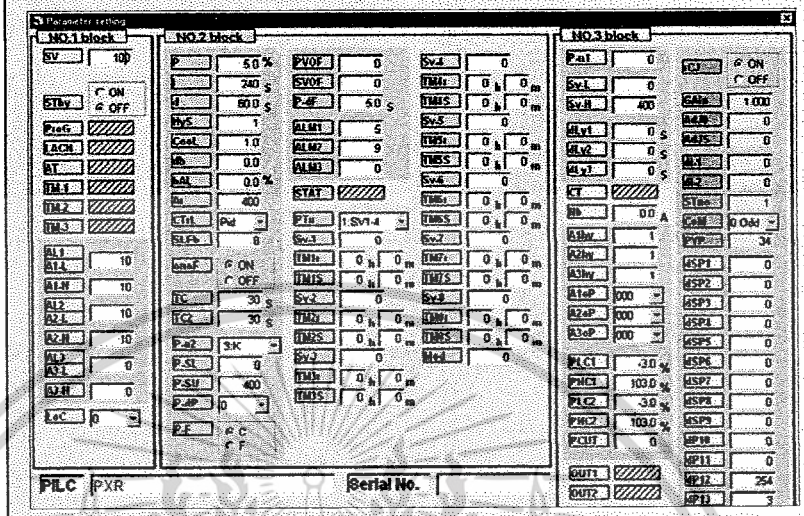
Control Output 2 (Heating/Cooling Control) (select one)

Relay contact SPST, 220 V AC/30 V DC, 3A (resistive load)

SSR PXR4, 5, 9: ON–17 to 25 V DC; OFF–0.5 V DC or less. PXR3: 12 to 16 V DC. Max. current: 20mA or less

4 to 20 mA DC PXR4, 5, 9: Allowable load resistance 600 ohms or less. PXR3: 100 to 500 ohms

Program Loader Interface

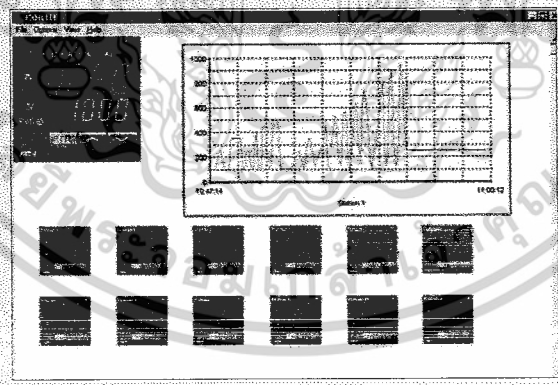


The Program Loader for Fuji Electric's PX and PXR series controllers is a powerful tool for the OEM customer. Using the PXR4 Loader Assembly, the controller can be configured from a PC running on Windows environment.

Features

- Retrieve or store controller data
- Selectively mask or unmask parameters for viewing on the controller
- Clone settings to other controllers from saved files
- Print data report

PXR LITE Communications Software



PXR-LITE™ is free Windows®-based software that is supplied with the communications option on a PXR controller. It is the latest in control and monitoring of Fuji Electric's PXR series controllers. It provides continuous remote monitoring of single or multiple controllers using a single half-duplex RS485 line.

Features

- Monitor and control up to 31 controllers from a PC via RS485-RS232 signal converter
- Real-time charting and data-logging
- Remote setpoint adjustment
- Set control modes, alarms and other control parameters
- Remote auto-tuning and ramp-soak programming
- Live display of process and setpoint values, alarm annunciators
- View single-station or multi-station data
- Comprehensive help file included
- Runs on Windows environment, version 3.1 or later

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Operation and Display Section

Parameter setting method Digital setting by 3 keys; Key lock function provided
Display unit Process value/set value displayed individually 4 digits, 7-segment LED
Status display LED Control output, process alarm output, heater burnout alarm output
Indication accuracy (at 23°C) Thermocouple: ± (0.5% of measuring range) ± 1 digit ±1°C. For thermocouple R at 0 to 500°C: ± (1% of measuring range) ±1 digit ±1°C. For thermocouple B at 0 to 400°C: ± (5% of measuring range) ±1 digit ±1°C. RTD, voltage/current: ±(0.5% of measuring range) ±1 digit

Alarm (option)

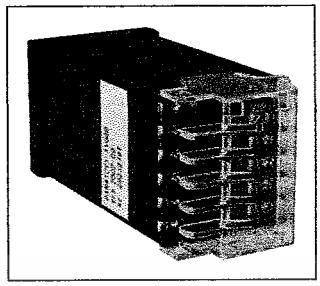
Alarm type Absolute alarm, deviation alarm, zone alarm with upper and lower limits for each; hold function available; alarm latch function provided
Alarm ON-delay Delay setting 0 to 9999 sec set in 1 sec steps
Process alarm output Relay contact: SPST, 220 V AC/30 V DC, 1A (resistive load); Max. 2 points (PXR3), max. 3 points (PXR4, 5, 9)
Heater burnout option (not available on PXR3) Alarm setting range: 1 to 50A Available only when control output is relay or SSR drive.
Heater burnout alarm output Relay contact: SPST, 220 V AC/30 V DC, 1A (resistive load); 1 output point
Current detector CTL-6-S for 1 to 30 A; CTL-12 for 20 to 50 A

Digital Input (option)

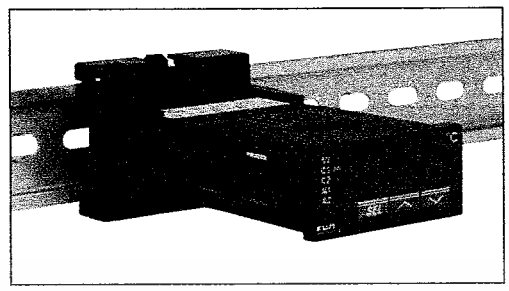
Points 1 or 2; contact closure. 5 V DC, approx. 2mA
Function (select one) Set value (SV, SV1 to 3) changeover, start/stop control action, start/reset ramp/soak action, start/stop auto-tuning, cancel alarm latch, start incorporated timer

Retransmission Output (option)

Output signal 4-20 mA DC
Load resistance 500 ohms or less
Output accuracy ±0.3% FS
Output selection PV, SV, MV, DV (SV-PV)



PXR4 Terminal Cover (option)



PXR3 DIN Rail Adapter

Timer Function (option)

Start By digital input option
Setting 0 to 9999 sec set in 1 sec steps
Action Event ON-delay or OFF-delay
Signal output Alarm output relays used; 2 points are available

Communication Function (option)

Physical specifications EIA RS485
Communication protocol Modbus (RTU). Free Windows®-based software, PXR-LITE™
Communication method 2-wire method; half-duplex, bit serial, start-stop sync type
Data type 8 bits. Parity: odd/even/none
Communication rate 9600 bps
Connection aspect Multi-drop up to 31 controllers
Communication distance Total extension 500m or less
RS232C/RS485 signal converter RSFC24 (recommended, see ordering information)

Remote Setpoint Option

Input signal 1 to 5 V DC, 1 point
Accuracy ±0.5% ±1 digit
Input sampling cycle 0.5 sec
Display of remote mode LED on front panel
Input impedance 1M ohms or more

Other Functions

Parameter mask function Parameter display can be disabled from keypad
Ramp/soak function 8 ramps and 8 soaks; 1 or 2 program patterns; digital input allows start/reset of the action

Operating and Storage Conditions

Ambient operating temperature 14 to 122°F (-10 to 50°C)
Ambient operating humidity Less than 90% RH (no condensation)
Storage temperature -4 to 140°F (-20 to 60°C)

Structure

Mounting method Panel flush mounting. PXR3 can be DIN-rail mounted using the optional adapter
External terminal Screw terminal
Dimensions PXR3: 1 x 2 x 4 in. (24 x 48 x 98mm). PXR4: 1.89 x 1.89 x 3.37 in. (48 x 48 x 79.8 mm). PXR5: 2.07 x 3.96 x 3.77 in. (52.5 x 100.5 x 95.8 mm). PXR9: 3.96 x 3.96 x 3.77 in (100.5 x 100.5 x 95.8 mm)
Protective structure Front panel NEMA4X (IEC standard IP66 equivalent) (when mounted on panel with supplied gasket). Rear case: IEC IP20
Outer color Black (front panel, case)
Agency approvals UL, c-UL recognized (UL873), CSA (C22.2 No.24-93), CE certified (LVD:EN61010-1, EMC:1326-1)

Optional Items

Current transformer For 1 to 30 A: CTL-6-S. For 20 to 50 A: CTL-12
Signal converter for communication function RSFC24
DIN Rail adapter For PXR3 only
Terminal cover For PXR4 only

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Ordering Information (PXR3)

P	X	R	A	–	B	C	D	1	–	E	F	G	A	1

To create a part number fill in the boxes above with the appropriate number and/or letter from the corresponding list below.

Box A: Front Panel Size

3 = 1/32 DIN (24x48mm)	$139

Box B: Input Signal

T = Thermocouple (°C)	N/C
R = Thermocouple (°F)	N/C
N = RTD, Pt100 ohm, 3-wire type (°C)	N/C
S = RTD, Pt100 ohm, 3-wire type (°F)	N/C
B = 4-20mA DC, 1-5V DC	N/C
A = 0-20mA DC, 0-5V DC	N/C

Box C: Control Output 1

A = Relay contact output	N/C
C = SSR or SSC drive output	N/C
E = 4-20mA DC output	N/C

Box D: Control Output 2

Y = None	N/C
A = Relay contact output	$ 35
C = SSR or SSC drive output	35
E = 4-20mA DC output	35

Box E: Alarm Options

4 = None	N/C
5 = High/low alarm 1 point	30
G = High/low alarm 2 points[1]	40

Box F: Power Supply

V = Standard (100-240 VAC, 50/60Hz)	N/C
B = 24V AC/DC (50/60Hz)	30

Box G: Additional Functions

0 = None	N/C
M = RS485 communication (Modbus)	$ 40
Q = Retransmission + DI 1 point	60
R = Retransmission (4-20mA DC)	50
T = Digital Input (DI) x 2	30
V = RS485 communications (Modbus) + DI	70

Note: RS485 option comes with Free software, PXR-LITE. RS485 requires signal converter to connect to PC, P/N RSFC24 recommended.
[1]High/low alarm 2 points not available when control output 2 is selected.

Ordering Information (PXR4, PXR5, PXR7, PXR9)

P	X	R	A	–	B	C	D	1	–	E	F	G	A	1

To create a part number fill in the boxes above with the appropriate number and/or letter from the corresponding list below.

Box A: Front Panel Size

4 = 1/16 DIN (48x48mm)	$179
5 = 1/8 DIN (48x96mm)	219
7 = 72 x 72mm	229
9 = 1/4 DIN (96x96mm)	239

Box B: Input Signal

T = Thermocouple (°C)	N/C
R = Thermocouple (°F)	N/C
N = RTD, Pt100 ohm, 3-wire type (°C)	N/C
S = RTD, Pt100 ohm, 3-wire type (°F)	N/C
B = 4-20mA DC, 1-5V DC	N/C
A = 0-20mA DC, 0-5V DC	N/C

Box C: Control Output 1

A = Relay contact output	N/C
C = SSR or SSC drive output	N/C
E = 4-20mA DC output[1]	N/C
G = Triac 1A, 250 VAC[4]	$35

Box D: Control Output 2

Y = None	N/C
A = Relay contact output	$ 35
C = SSR or SSC drive output	35
E = 4-20mA DC output[1]	35
G = Triac 1A, 250 VAC[4]	35
R = Retransmission (4-20mA DC)	50

Box E: Alarm Options

4 = None	N/C
6 = Heater break alarm[1,2]	50
G = High/low alarm 2 points	30
H = High/low alarm 2 points + heater break alarm[1,2]	80
M = Alarm 3 points	50
D = Remote setpoint[3]	80
P = Remote setpoint + alarm 2 points[3]	110

Box F: Power Supply

V = Standard (100-240V AC, 50/60Hz)	N/C
B = 24V AC/DC (50/60Hz)	$ 30

Box G: Additional Functions

0 = None	N/C
M = RS485 communication (Modbus)	40
S = Digital Input (DI) x 1	30
T = Digital Input (DI) x 2[1]	30
V = RS485 communications (Modbus) + DI[1,3]	70

Note: RS485 option comes with Free software, PXR-LITE. RS485 requires signal converter to connect to PC, P/N RSFC24 recommended.
[1]Heater break option not available with 4-20mA output, or with 2 digital inputs, or with RS485 +1 digital input.
[2]Must order current transformer CTL-6-S or CTL-12 with heater break option.
[3] Remote setpoint option not available with RS485 +1 digital input.
[4] UL and c-UL not available with PXR5 or PXR9.

Accessories

CTL-6-S	Current transformer for 1-30A	$ 23
CTL-12	Current transformer for 20-50A	40
RSFC24	RS485 to RS232 signal converter	135
PXR4 Loader Assembly	Program loader for PXR4 (can be used for PX series also)	250
PXR3 Rail Adapter	Mounting adapter for DIN rail installation	15
PXR4 Terminal Cover	Terminal block protective cover	15

Information subject to change without notice. Prices in USD. 031907

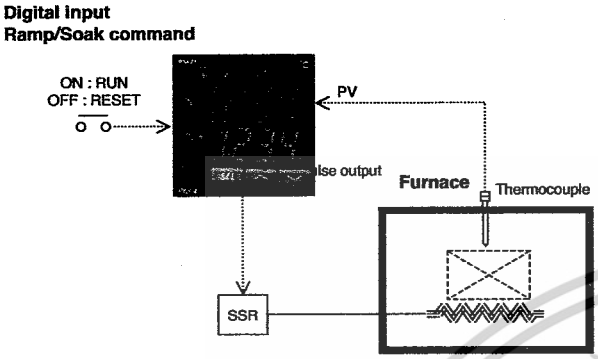
เอกสารนี้จัดทำขึ้นสำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

PXR Series (continued)

Application Examples

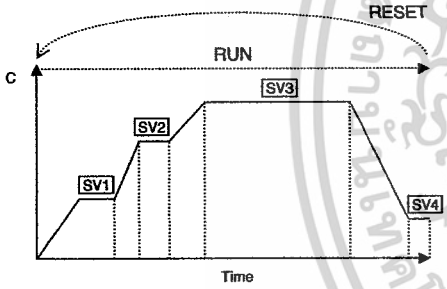
Furnace Heat Pattern Control

Heat Pattern Control — Ramp/Soak Function



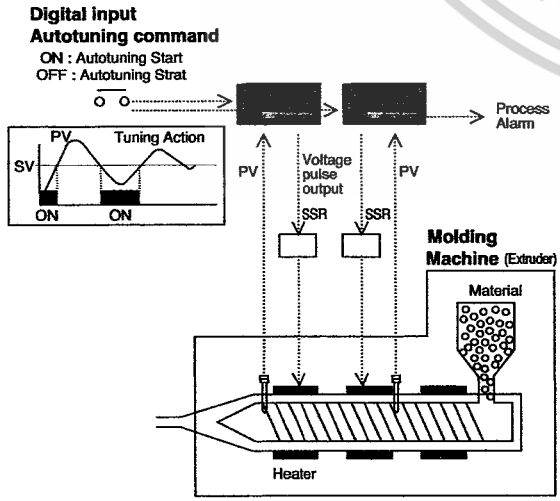
Ramp/Soak Function

- Control temperature according to "Heat pattern with ramp"
- Keep temperature stable for a certain period with "Heat pattern" and then cool down
- "Heat pattern" can be Started (RUN) /Reset by an external digital input.



Plastic Molding Machine

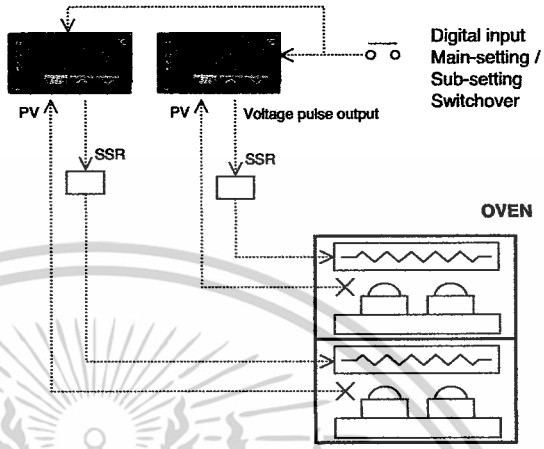
Stable temperature control required — Fuzzy + PID Control



Auto-Tuning can be started/stopped through external digital input

Oven

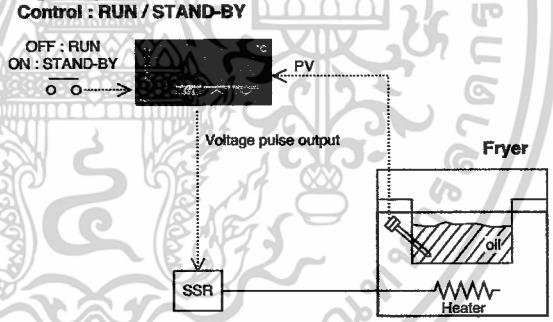
To change SV easily



Set Value (SV) can be selected/changed externally.
<main SV, SV1~3 change over>

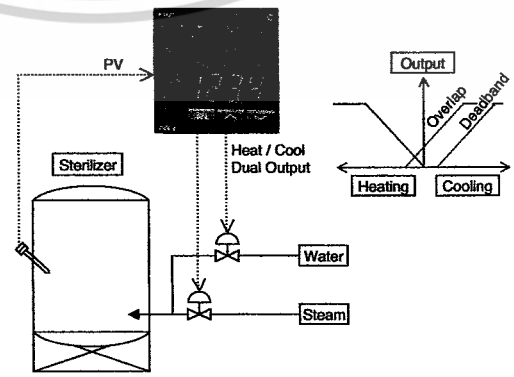
Fryer

To keep oil temperature stable



Control RUN/Stand-by selectable through external digital input

Cooling & Heating Control



Cooling output and Heating output can be overlapped or a "Dead-band" set between them.

เอกสารนี้มีไว้สำหรับการที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

User's Manual

Future Design Controls SNA10A Smart Network Adaptor

1.Features

* Supports both RS-485 and RS-422 Interface
* Baud Rate: 300 ~ 38400 bits/sec configurable
* Allows connection for 247 multi-drop units
* Automatic data direction control for RS-485 without the need to take care of RTS signal.
* Precision timing control for RS-485 allows fast switching between transmit and receive
* Universal (90 ~ 264 VAC) AC power input
* Isolated between RS-232 and RS-485 / 422 eliminate common mode noise problems
* Flexible installation: DIN rail mount or wall mount
* CE Approved

2.Introduction

SNA10A is a smart network adaptor which can be used to convert unbalanced RS-232 signals to balanced RS-485 or RS-422 signals. SN10A is used for single node conversion or when software other than Future Design Controls Multiview is used.

The RS-485 is an enhanced version of the RS-422A balanced line standard. It allows multiple drivers and receivers on a 2-wire system and reduces wiring cost. This 2-wire system can perform half-duplex transmission only. Because RS-422 is a 4-wire system, it can perform full-duplex transmission. The driving capability is dependent on the input impedance of the connected receivers.

As many as 32 standard units can be put on RS-422 or RS-485 port.
Up to 247 high impedance units, such as Future Design Control's interface products, can be put on RS-422 or RS-485 port.

3.RS-232C Interface

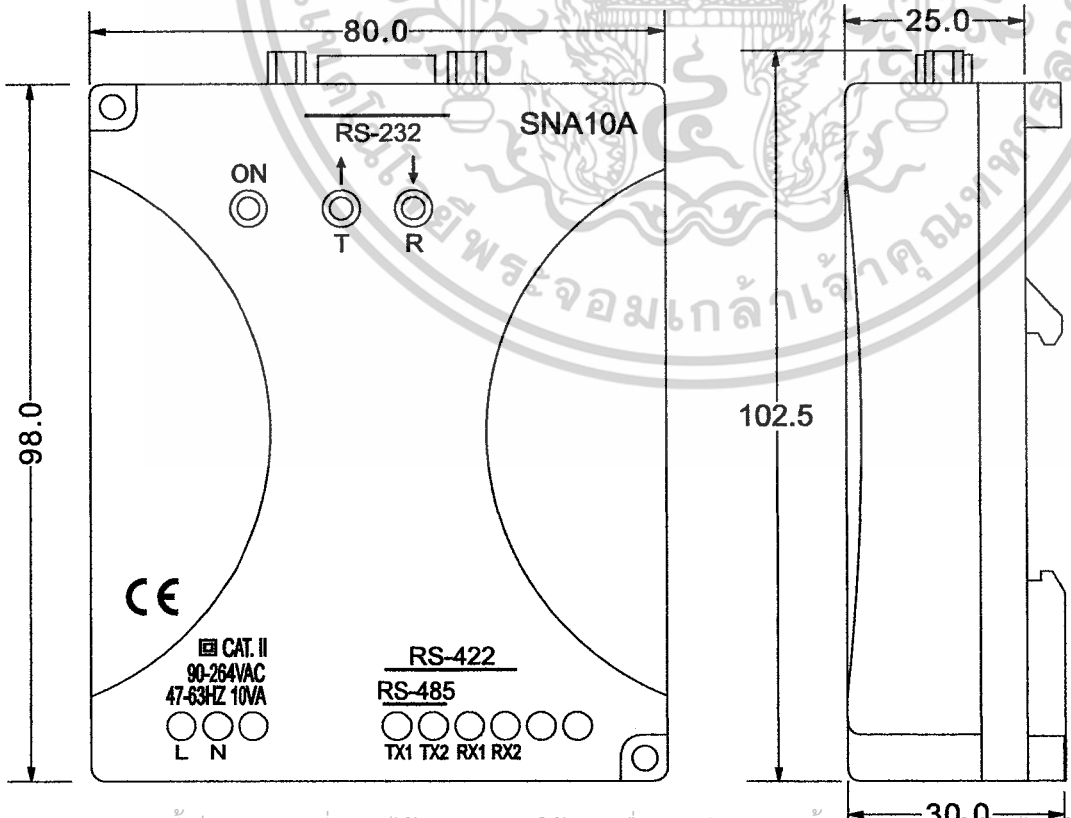
Pin	EIA	Description	Source
1	CF	Carrier Detect (DCD)	DCE
2	BB	Received Data (RD)	DCE
3	BA	Transmitted Data (TD)	DTE
4	CD	Data Terminal Ready (DTR)	DTE
5	AB	Signal Ground (SG)	DTE/DCE
6	CC	Data Set Ready (DSR)	DCE
7	CA	Request to Send (RTS)	DTE
8	CB	Clear to Send (CTS)	DCE
9	CE	Calling Indication (RI)	DCE

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.Specifications

Baud rate:	300 ~ 38400 bits/sec
Parity bit:	None, odd or even
Data bit:	8 bits
Stop bit:	1 or 2 bits
Connectors:	9-pin Female D-SUB (RS-232)
	Screw type terminal block (RS-485/422)
Receiver threshold:	0.8 V min. 2.4 V max. (RS-232)
	K0.2 V (RS-485/422)
Receiver input impedance:	3K ~ 7 Kohm (RS-232)
	96 Kohm (RS-485/422)
Transmission mode:	Single ended (RS-232)
	Differential (RS-485/422)
Transmission distance:	50 ft (RS-232)
	5000 ft (RS-485/422)
Common-mode voltage:	K25 V (RS-232)
	+12 V, -7V (RS-485/422)
Driving capability:	32 receivers (12 Kohm input)
	247 receivers (96 Kohm input)
Power:	90~264 VAC, 47~63 Hz, 10VA, 4W max.
Breakdown Voltage:	2500VAC, 1minute (power to RS-232, RS-485/422)
	400 VAC, 1 minute (between RS-232 and RS-485/422)
Isolation resistance:	>500 Mohm VS. 500 VDC
Ambient temperature:	0~50 LC
Storage temperature:	-20~80 LC
Agency Approvals:	CE Approved
Mounting method:	DIN rail mount or wall mount
Dimension:	102.5 (L) X 80 (W) X 30 (H) mm
Weight:	120 grams

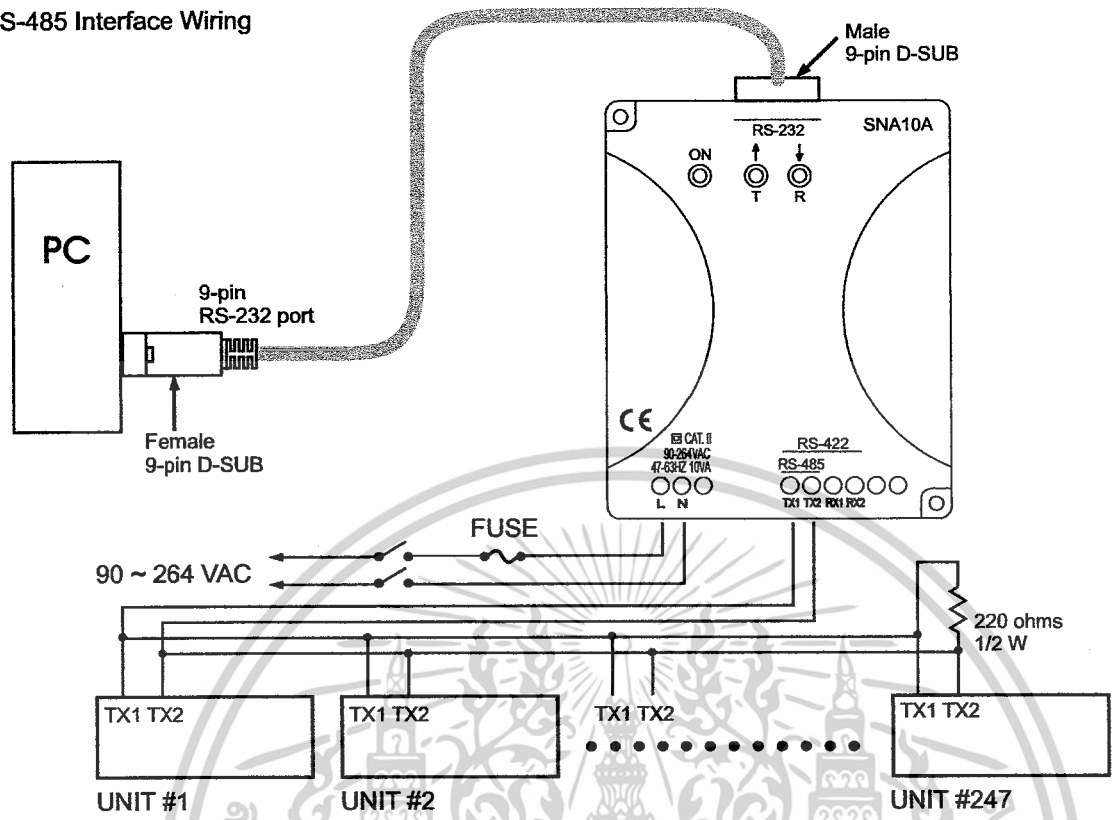
5.Mechanical Data



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

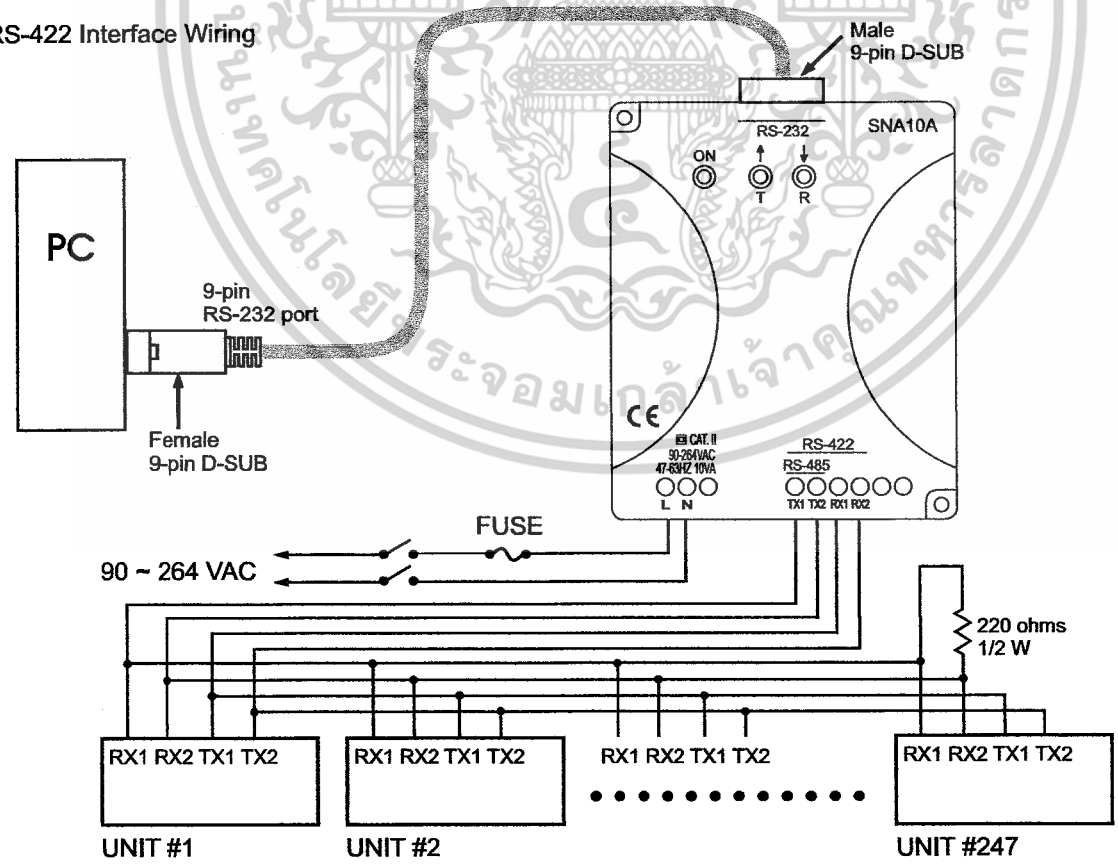
6.Application

(1) RS-485 Interface Wiring



A 220 ohms 1/2 W termination resistor across the TX1 and TX2 terminals of the last unit in the network is required.

(2) RS-422 Interface Wiring



A 220 ohms 1/2 W termination resistor across the receive terminals of the last unit in the network is required.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

7.DIP Switch Setting

SNA10 DIP SWITCH SETTING
■ = ON POSITION
BLANK = OFF POSITION

		1	2	3	4	5	6	7	8
Interface	RS-422	■							
	RS-485								
Parity Bit	None		■	■					
	Even			■					
	Odd								
Stop Bit	1 bit				■				
	2 bit								
Baud Rate (bps)	300					■	■	■	■
	600						■	■	■
	1200					■		■	■
	2400							■	■
	4800					■	■		■
	9600						■		■
	14400					■			■
	19200								■
	28800					■	■	■	
	38400						■	■	

8.Ordering Data

SNA10A: Smart Network Adaptor for Future Design Controls mulit-drop Multiview software or third party software

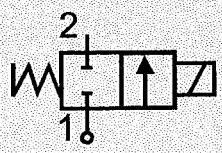
SNA10A
Smart Network Adapter

Future Design Controls
7524 West 98th Place
Bridgeview, IL 60455
Phone 888-751-5444
Fax 888-307-8014
E-mail csr@futuredesigncontrols.com
www.futuredesigncontrols.com

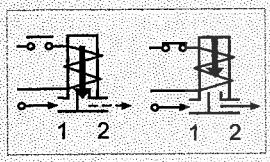
Rev 1 7/2003

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Series 131.4



N.C.
Normally closed
Coil energised - open
Coil de-energised - closed



General description:

PARKER series **131.4** solenoid valves are direct operated.
They are used for general applications with media such as **inert gases and light oils (2°E).**
Series **131.4** valves are **normally closed**.

Temperatures:

The working temperature for media is:
maximum **+140°C**
minimum **-10°C**
The maximum ambient temperature is:
+50°C

Application:

Series **131.4** solenoid valves are used for the automatic control of media where low flow rates are required.
Some examples of applications are:
- welding machines with controlled atmosphere;
- diesel oil burners;
- gas analysers;
- fumes analysers;
- distribution of light oils;
- measurement and control instruments;
- lubrication systems.

Coils:

For series **131.4** valves class **"F"** coils **(155°C)**, encapsulated in thermoplastic containing 30% glass fiber (type KT), are available.
All the coils are for continuous service, 100% E.D.
The rated voltage tolerance is:
±10% for A.C. power supply and
+10% -5% for D.C.
The "K" coil can be used on a.c. with frequency 50 Hz or 60 Hz (single frequency) and has Faston terminals for **DIN 43650A** connector with protection to **IP65.**

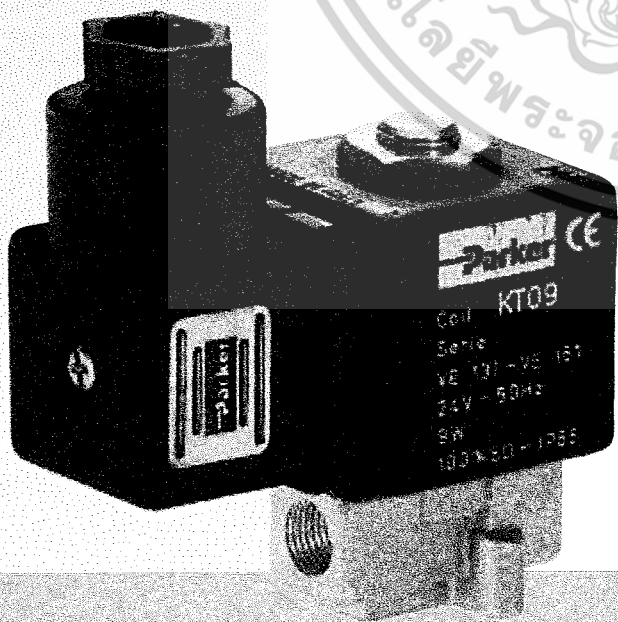
Installation:

The valves can be mounted in any position without jeopardising their operation. It is however advisable to install them with the coil in a vertical position above the body.
The valve body has 2 mounting holes diam. M4 x 7 with centre distance 13 x 13.

Approvals:

- Coil certification:

KH 09 24V/50Hz
115V/50Hz
230V/50Hz
KT 09 115V/50Hz
220-230V/50Hz



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของลิขสิทธิ์ทุกครั้งที่มีการนำไปใช้

Parker Climate & Industrial Controls
Parker Hannifin S.p.A.
via E. Fermi, 5
20060 Gessate (Milano) - Italy

MATERIALS

• Valve body:	OT58 UNI 5705 brass stamping
• Seals:	Viton
• Enclosing tube:	OT58 UNI 5705 brass stamping
• Plunger:	Steel 9SMn Pb23 UNI 5105 nickel plated
• Spring:	AISI 302 SS
• Shading ring:	Copper

SS = stainless steel

ELECTRICAL FEATURES

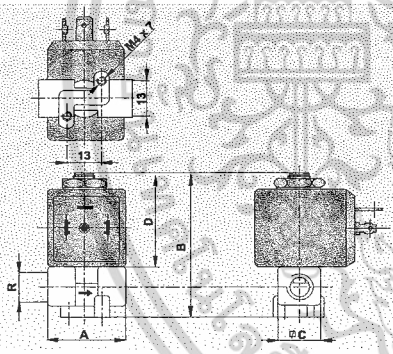
Coil type [ ]		Power [W]		Insulat. class
A.C.(~)	D.C.(=)	A.C.(~)	D.C.(=)	
KT 09	KT 10	9	10	F
KH 09	-	9	-	H

SPECIFICATION

Fittings Ø Rp	Valve type	Nominal orifice Ø	Flow coefficient Kv	Minimum pressure	Max differential pressure (M.O.P.D.)		Coil type	Weight	Notes
					in A.C.(~)	in D.C.(=)			
["]	[ ]	[mm]	[m³/h]	[bar]	[bar]	[bar]	[ ]	[Kg]	[ ]
1/8	131.4 BV	2,8	0,174	0	8	6	K	0,200	1
1/4	131.4 FV	2,8	0,174	0	8	6	K	0,220	1
1/4	131.4 GV	4,0	0,318	0	2	1	K	1,220	1

Note: 1) NP (nominal pressure): 10 bar

DIMENSIONS



Fittings Ø R	A	B	C	D
["]	[mm]	[mm]	[mm]	[mm]
1/8	32	56,5	14	35,5
1/4	38	60	16	35,5

ORDER CODE

VE | **131.4** | | **V** | | [V] | [Hz] / d.c.

COMPLETE VALVE

FITTINGS

B	1/8"
F	1/4"
G	1/4"

SERIES

SEAL

V	Viton

Coil type	24 V 50 Hz*	115V 50 Hz*	220-230 V 50 Hz*	240 V 50 Hz*	12 V d.c.	24 V d.c.
KT 09	●	●	●	●		
KT 10					●	●
KH 09		●	●			

* models with 60 Hz availables

Note: Valve supplied with coil in a multipack. Connector to be ordered separately.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Parker Climate & Industrial Controls
Parker Hannifin S.p.A.
via E. Fermi, 5
20060 Gessate (Milano) - Italy

CARLO GAVAZZI

Input Specifications

Control current range	4 to 20 mA
Voltage drop	≤ 8 VDC @ 20 mA
Dynamic impedance	≥ 100 Ω
Allowable input current	≤ 50 mA
Reverse polarity protected	Yes

Caution:
Current to input must be limited to max. 50 mA.

Insulation

Rated insulation voltage Input to output	≥ 4000 VACrms
Rated insulation voltage Output to case	≥ 4000 VACrms
Insulation resistance Input to output	≥ 10^{10} Ω
Insulation resistance Output to case	≥ 10^{10} Ω
Insulation capacitance Input to output	≤ 8 pF
Insulation capacitance Output to case	≤ 50 pF

Output Specifications

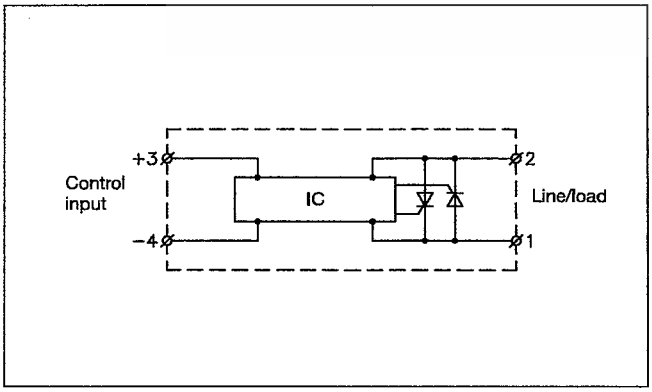
	RE ..10 AA ..	RE ..25 AA ..	RE ..50 AA ..	RE 6050 AA 16
Rated operational current AC 51	10 Arms	25 Arms	50 Arms	50 Arms
Minimum operational current	500 mArms	500 mArms	500 mArms	500 mArms
Rep. overload current t=1 s	< 35 AACrms	55 AACrms	125 AACrms	125 AACrms
Off-state leakage current @ rated voltage and frequency*)	< 6 mA	< 6 mA	< 6 mA	< 6 mA
I^2t for fusing t = 1-10 ms	≤ 130 A^2s	≤ 310 A^2s	≤ 1800 A^2s	≤ 1800 A^2s
Critical dI/dt	≥ 100 A/µs	≥ 100 A/µs	≥ 100 A/µs	≥ 100 A/µs
Critical dV/dt commutating	≥ 500 V/µs	≥ 500 V/µs	≥ 500 V/µs	≥ 500 V/µs
Critical dV/dt off-state	≥ 500 V/µs	≥ 500 V/µs	≥ 500 V/µs	≥ 500 V/µs

*) Min. power let-through ≥ 0.5%. Please refer to Applications.

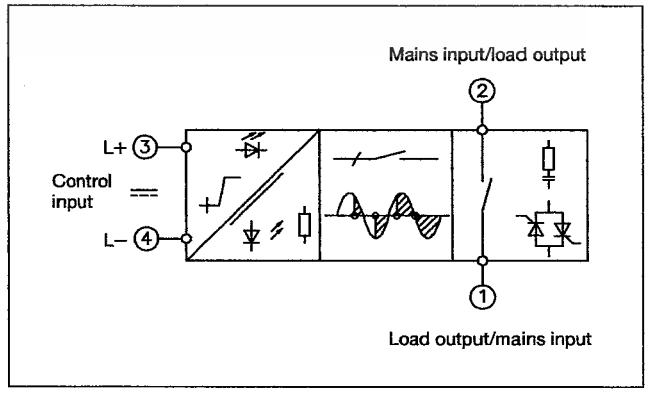
Thermal Specifications

	RE ..10 AA ..	RE ..25 AA ..	RE ..50 AA ..
Operating temperature	-20° to +70°C (-4° to +158°F)	-20° to +70°C (-4° to +158°F)	-20° to +70°C (-4° to +158°F)
Storage temperature	-40° to +100°C (-40° to +212°F)	-40° to +100°C (-40° to +212°F)	-40° to +100°C (-40° to +212°F)
Junction temperature	≤ 125°C (257°F)	≤ 125°C (257°F)	≤ 125°C (257°F)
R_{th} junction to case	≤ 2 K/W	≤ 1.25 K/W	≤ 0.65 K/W
R_{th} junction to ambient	≤ 12.5 K/W	≤ 12 K/W	≤ 12 K/W

Wiring Diagram



Functional Diagram



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้