การควบคุมและการแสดงผลการทำงานแบบลำดับด้วยเครื่องควบคุม PLC

The sequential control and monitoring system Base on PLC

นายจุฬศิษฏ์ สมัตถะ

นายพีรวัส ศรีม่วง

นายรัฐภูมิ ชาวเวียง

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมการวัดคุม

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2556

การควบคุมและการแสดงผลการทำงานแบบลำดับด้วยเครื่องควบคุม PLC

The sequential control and monitoring system Base on PLC

นายจุฬศิษฏ์ สมัตถะ
นายพีรวัส ศรีม่วง
นายรัฐภูมิ ชาวเวียง

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิชาวิศวกรรมการวัดคุม
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
พ.ศ. 2556

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อปริญญานิพนธ์	การควบคุมและการแสดงผลการทำงานแบบลำดับด้วยเครื่องควบคุม PLC The sequential control and monitoring system Base on PLC			
นักศึกษาผู้จัดทำ	**นายจุฬศิษฏ์**	**สมัตถะ**	**รหัสนักศึกษา**	53010248
	นายพีรวัส	**ศรีม่วง**	**รหัสนักศึกษา**	53011164
	นายรัฐภูมิ	**ชาวเวียง**	**รหัสนักศึกษา**	53011356
ปีการศึกษา	2556			

บทคัดย่อ

โครงงานนี้นำเสนอแบบจำลองการทำงานของลูกสูบ เพื่อให้สอดคล้องกับการทำงานของเครื่องจักรกลในอุตสาหกรรมที่มีการทำงานเป็นลำดับ โดยอาศัยการจ่ายกำลังแบบลมที่มีระบบควบคุมเป็นไฟฟ้า การตรวจจับตำแหน่งของลูกสูบอาศัยเซ็นเซอร์แบบแม่เหล็ก (Magnetic) ส่งสัญญาเป็นอินพุตให้กับเครื่องควบคุม PLC จากนั้น เครื่องควบคุมจะทำการประมวลผลตามโปรแกรมการทำงานที่ได้กำหนดไว้ แล้วส่งค่าเอาท์พุทที่ได้ไปยังโซลินอยด์วาล์ว (Solenoid Valve) ที่ถือว่าเป็นอุปกรณ์ทางด้านเอาท์พุทให้ เปิด/ปิด การทำงานตามเงื่อนไข จึงทำให้สามารถทำงานได้ตามความต้องการ ในการทดสอบ PLC จะควบคุมลูกสูบให้มีการทำงานในหลายรูปแบบ โดยแสดงผลการทำงานบนจอแสดงผลผ่านคอมพิวเตอร์ในระบบการเชื่อมต่อแบบโฮสลิงค์ (Host link) เพื่อให้ผู้ปฏิบัติงานสามารถมองเห็นการทำงานของระบบได้เสมือนจริง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Thesis Title	The sequential control and monitoring system Base on PLC	
Authors	Mr. Jurasit	Samatha
	Mr. Peerawas	Srimuang
	Mr. Rattapoom	Chawvieng
Thesis Advisor	Mr. Suphan	Kullapanich
Year	2013	

Abstract

This project presents the operation of piston for belong to the operation of machine in industrial with sequence control using air-supply (Pneumatic) and control by electrical. The Magnetic sensor is detect the position of the piston and send electrical signal to controller (PLC) and then, controller will use the input signal (from Magnetic sensor) then processing by program in memory of controller and send the signal output to the solenoid valve for operate like a pattern in program. In case controller (PLC) will monitoring based on Hostlink system for operater .

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

ปริญญานิพนธ์ฉบับนี้สำเร็จลุล่วงได้ด้วยดี เพราะได้รับคณาจารย์ภาควิชาวิศวกรรมการวัดคุม ทุกท่านที่ได้มอบคำแนะนำและกำลังใจในการทำวิจัยตลอดมา โดยเฉพาะอย่างยิ่ง ท่านรศ.สุพรรณ กุลพาณิชย์ ซึ่งเป็นอาจารย์ที่ปรึกษา ที่ให้คำปรึกษา คำแนะนำรวมถึงอุปกรณ์ต่างๆแก่ผู้วิจัยตลอดมา จนทำให้การทำปริญญานิพนธ์ฉบับนี้สำเร็จลุล่วงไปด้วยดี

รวมไปถึงเพื่อนๆ พี่ๆ ที่คอยช่วยให้กำลังใจในการทำวิจัยและความช่วยเหลือในด้านต่างๆ แก่คณะผู้จัดทำ

สุดท้ายนี้ขอขอบคุณภาควิชาวิศวกรรมการวัดคุม คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ที่มอบสิ่งดีๆตลอดที่คณะผู้จัดทำได้เล่าเรียนมาโดยตลอด คุณค่าและประโยชน์อันพึงมีจากปริญญานิพนธ์ฉบับนี้ คณะผู้จัดทำขอมอบแด่ผู้มีพระคุณทุกท่าน

คณะผู้จัดทำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

หน้า

บทคัดย่อภาษาไทย I

บทคัดย่อภาษาอังกฤษ II

กิตติกรรมประกาศ III

สารบัญ IV

สารบัญตาราง IX

สารบัญภาพ X

บทที่ 1 บทนำ

1.1 ความสำคัญของปริญญานิพนธ์ 1

1.2 วัตถุประสงค์ของปริญญานิพนธ์ 1

1.3 ขอบเขตของปริญญานิพนธ์ 2

1.4 ขั้นตอนการศึกษา 2

1.5 ประโยชน์ที่คาดว่าจะได้รับ 2

บทที่ 2 ทฤษฎีที่ใช้ในการออกแบบ 3

2.1 กล่าวนำ 3

2.2 ทฤษฎีและหลักการของ Visual Basic 2010 3

2.2.1 จุดเด่นของ Visual Basic 3

2.2.1.1 สร้างแอพพลิเคชั่นได้ง่ายและรวดเร็ว 3

2.2.1.2 การเขียนโปรแกรมที่ง่ายต่อการเรียนรู้ 4

2.2.1.3 รวมเครื่องมืออำนวยความสะดวกในการเขียนโปรแกรม 4

2.2.2 รูปแบบการพัฒนาแอพพลิเคชั่นด้วย Visual Basic 4

2.2.2.1 พัฒนาแอพพลิเคชั่นกับ ActiveX Control 4

2.2.2.2 สร้างแอพพลิเคชั่นที่ใช้กับฐานข้อมูล 5

2.2.2.3 สร้างแอพพลิเคชั่นแบบใหม่กับอินเตอร์เน็ต 5

2.2.3 สร้างแอพพลิเคชั่นด้วย Visual Basic 5

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ(ต่อ)

หน้า

2.3 ฐานข้อมูล (Database) 6
2.3.1 องค์ประกอบฐานข้อมูล 6
2.3.1.1 แอพพลิเคชั่นฐานข้อมูล (Database Application) 6
2.3.1.2 ระบบจัดการฐานข้อมูล (DBMS) 7
2.3.1.3 ดาต้าเบสเซิฟเวอร์ (Database Server) 7
2.3.1.4 ข้อมูล (Data) 7
2.3.1.5 ผู้บริหารฐานข้อมูล (Database Administrator) 7
2.3.2 ชุดคำสั่งที่ใช้จัดการข้อมูล 8
2.3.3 โครงสร้างของภาษา 8
2.3.3.1 ตัวดำเนินการ (Operator) 9
2.3.3.2 ตัวดำเนินการด้านตรรกะ (Logical Operator) 9
2.3.3.3 กลุ่มฟังก์ชัน Aggregate 9
2.3.4 ลักษณะการใช้งานของกลุ่มคำสั่ง DML 10
2.3.5 การเชื่อมโยงตาราง 11
2.3.6 เทคโนโลยี ActiveX Data Object 13
2.3.7 วิธีติดต่อกับฐานข้อมูลของ Visual Basics 14
2.3.7.1 การใช้งานดาต้าคอนโทรล (Data Control) 15
2.4 แลดเดอร์ไดอะแกรม (Ladder Diagrams) 15
2.4.1 กลุ่มคำสั่งลอจิกพื้นฐาน 15
2.4.1.1 การใช้คำสั่ง LOAD (LD), LOAD NOT (LD NOT) 16
2.4.1.2 การใช้คำสั่ง AND , AND NOT 16
2.4.1.3 การใช้คำสั่ง OR, OR NOT 17
2.4.1.4 การใช้คำสั่ง OUT 17
2.4.1.5 การใช้คำสั่ง AND LOAD (AND LD), OR LOAD (OR LD) 18
2.4.1.6 การใช้คำสั่ง END (END) 19
2.4.1.7 การใช้งานคำสั่ง SFT (10) 19

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของผลงานทุกครั้งที่มีการนำไปใช้

สารบัญ(ต่อ)

หน้า

2.4.1.8 การใช้งานคำสั่ง IL (02), ILC (03) 19

2.4.1.9 การใช้งานคำสั่ง Differential UP-DIFU(013) และ Differential DOWN-DIFD(014) 20

2.5 การติดต่อสื่อสารและส่งผ่านข้อมูล 20

2.5.1 การสื่อสารข้อมูลแบบอนุกรม 21

2.5.1.1 การส่งข้อมูลแบบซิมเพล็กซ์ (Simplex) และ ดูเพล็กซ์ (Duplex) 21

2.5.1.2 โปรโตคอลของการสื่อสารแบบอนุกรม 22

2.5.1.3 มาตรฐานสัญญาณอนุกรมแบบ RS232C 23

2.5.1.3.1 ลักษณะสมบัติทางไฟฟ้าของการอินเทอร์เฟสแบบ RS232 24

2.5.1.4 พอร์ตสื่อสารอนุกรม 24

2.6 เซนเซอร์ตรวจจับแบบแม่เหล็กแบบรีดสวิตช์ 25

2.6.1 หลักการการทำงานของรีดสวิตช์ 26

2.7 วาล์วควบคุมโดยใช้ไฟฟ้า (Solenoid Valve) 26

2.8 กระบอกสูบแบบ 2 ทาง (Two way cylinder) 27

2.9 ชุดตัวควบคุม Programmable Logic Control : PLC 28

2.9.1 ส่วนประกอบที่สำคัญของชุดตัวควบคุม 29

2.9.2 โปรโตคอลในการติดต่อสื่อสาร 30

2.9.2.1 คำสั่งไรท์ (Write command) 30

2.9.2.2 คำสั่งรีด (Read command) 31

2.9.3 Frame Check Sequence : FCS Calculation 32

บทที่ 3 การออกแบบและการทำงาน 33

3.1 กล่าวนำ 33

3.2 การออกแบบระบบควบคุม 33

3.2.1 การออกแบบกล่องสัญญาณขาเข้าและขาออก 35

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ(ต่อ)

หน้า

3.2.2 การออกแบบระบบนิวเมติก 36
3.3 การออกแบบโปรแกรมควบคุม 36
3.3.1 โปรแกรมการควบคุมและการแสดงสถานะของการทำงาน 37
3.3.1.1 ส่วนการสั่งการ 37
3.3.1.2 ส่วนการแสดงผลสถานะของกระบอกสูบและเซนเซอร์ 41
3.3.1.3 ส่วนของกราฟ Timing Diagrams 42
3.3.1.4 ส่วนของไฟเตือนสภาพการใช้งานของกระบอกสูบ 43
3.3.2 โปรแกรมการควบคุมแบบกำหนดเอง 44
3.3.3 โปรแกรมการแสดงค่าจำนวนครั้งการทำงานของกระบอกสูบ 44
3.3.4 โปรแกรมการแก้ไขข้อมูลเมื่อมีการซ่อมบำรุงกระบอกสูบ 44
3.4 รายละเอียดของโปรแกรม 45
3.4.1 เข้าใช้โดยผู้ดูแล (ADMIN) 46
3.4.2 เข้าใช้โดยผู้ใช้ 52

บทที่ 4 ผลการทดลอง 53
4.1 ผลการทดลอง RUN โปรแกรมควบคุมการทำงานของกระบอกสูบ 53
4.1.1 หน้าโปรแกรมหลัก 53
4.1.2 โปรแกรมการควบคุมและการแสดงสถานะของการทำงาน 54
4.1.3 โปรแกรมการควบคุมแบบกำหนดเอง 56
4.1.4 โปรแกรมแสดงข้อมูลจำนวนครั้งการทำงานของกระบอกสูบ 57
4.1.5 โปรแกรมการแก้ไขข้อมูลเมื่อมีการซ่อมบำรุงกระบอกสูบ 57
4.2 การทดสอบการทำงาน 58
4.2.1 การทดสอบการทำงานของระบบด้วยการเลือก “Pattern1” 58
4.2.2 การทดสอบการทำงานของระบบด้วยการเลือก “Pattern2” 60
4.2.3 การทดสอบการทำงานของระบบด้วยการเลือก “Pattern3” 61
4.2.4 การทดสอบการทำงานของระบบด้วยการเลือก “Pattern4” 62

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ(ต่อ)

หน้า

4.2.5 การทดสอบการทำงานของระบบด้วยการเลือก “Pattern5”.....63
4.2.6 การทดสอบการทำงานของสัญญาณไฟเตือนสภาพ
การใช้งานของกระบอกสูบ.....63
4.2.7 การทดสอบโปรแกรมการควบคุมแบบกำหนดเอง.....65
4.2.8 การทดสอบโปรแกรมการแก้ไขข้อมูลเมื่อมีการซ่อมบำรุงกระบอกสูบ.....65

บทที่ 5 สรุปผลการทดลองและข้อเสนอแนะ.....68
5.1 สรุปผลการทดลอง.....68
5.2 ข้อเสนอแนะ.....69
5.3 อุปสรรคและปัญหา.....69

บรรณานุกรม.....70

ภาคผนวก ก.....72
ภาคผนวก ข.....107
ภาคผนวก ค.....123

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

ตารางที่ หน้า

2.1 ขาสัญญาณต่างๆในการส่งข้อมูลแบบอนุกรม RS-232 Standard.....25

2.2 แสดงรายละเอียดโครงสร้างของกระบอกสูบชนิดทางานสองทิศทาง.....27

3.1 แสดงการเชื่อมต่อสัญญาณขาเข้าและสัญญาณขาออกกับตัวควบคุม (I/O Assignment) 34

4.1 การทดสอบการทำงานของกระบอกสูบ รูปแบบที่ 1.....59

4.2 การทดสอบการทำงานของกระบอกสูบ รูปแบบที่ 2.....60

4.3 การทดสอบการทำงานของกระบอกสูบ รูปแบบที่ 3.....61

4.4 การทดสอบการทำงานของกระบอกสูบ รูปแบบที่ 4.....62

4.5 การทดสอบการทำงานของกระบอกสูบ รูปแบบที่ 5 63

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญภาพ

รูปที่ หน้า

2.1 แสดงแลดเดอร์ไดอะแกรมของคำสั่ง LOAD (LD), LOAD NOT (LD NOT) 16

2.2 แสดงแลดเดอร์ไดอะแกรมของคำสั่ง AND, AND NOT.....16

2.3 แสดงแลดเดอร์ไดอะแกรมของคำสั่ง OR, OR NOT.....17

2.4 แสดงแลดเดอร์ไดอะแกรมของคำสั่ง OUT.....17

2.5 แสดงแลดเดอร์ไดอะแกรมของคำสั่ง AND LOAD (AND LD), OR LOAD (OR LD) 18

2.6 แสดงสว่นประกอบของคำสั่ง SHT(10) 19

2.7 แสดงการเชื่อมต่อคอมพิวเตอร์กับตัวควบคุมผ่านพอร์ตอนุกรม RS232.....20

2.8 การสื่อสารแบบอนุกรม.....22

2.9 การส่งข้อมูลผ่านการสื่อสารแบบอนุกรม.....23

2.10 ตำแหน่งขาของ DB-9 และ DB-25.....24

2.11 วงจรและส่วนประกอบภายในของรีดสวิตช์.....25

2.12 การทำงานของรีดสวิตช์.....26

2.13 วาล์วควบคุมด้วยไฟฟ้าแบบ 5/2.....26

2.14 ส่วนประกอบของกระบอกสูบ.....27

2.15 โครงสร้างภายในของกระบอกสูบ.....27

2.16 แสดงหลักการทำงานภายในของกระบอกสูบสองทางเมื่อจ่ายลมอัดภายใน.....28

2.17 ตัวควบคุม PLC Omron Sysmac C28H.....28

2.18 ตัวควบคุม PLC Omron Sysmac C28H.....29

2.19 การตอบโต้กันระหว่างคอมพิวเตอร์กับตัวควบคุม PLC.....30

2.20 หลักการตรวจสอบแบบ FCS Calculation.....32

3.1 แสดงการเชื่อมต่อกันของอุปกรณ์ภายในระบบ.....34

3.2 กล่องสัญญาขาเข้าและสัญญาณขาออก.....35

3.3 การเชื่อมต่อกันระหว่างกล่องสัญญาณขาเข้าและสัญญาณขาออกกับตัวควบคุม.....35

3.4 การเชื่อมต่อกันในระบบนิวเมติก.....36

3.5 รูปแบบการทำงานที่ 1.....37

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญภาพ(ต่อ)

หน้า

3.6 รูปแบบการทำงานที่ 2.....38
3.7 รูปแบบการทำงานที่ 3.....38
3.8 รูปแบบการทำงานที่ 4.....39
3.9 รูปแบบการทำงานที่ 5.....39
3.10 Flow Chart ขั้นตอนการทำงานส่วนสั่งการ.....40
3.11 Flow Chart ขั้นตอนการทำงานส่วนแสดงผล.....41
3.12 Flow Chart ขั้นตอนการทำงานส่วนกราฟ Timing Diagrams.....42
3.13 Flow Chart ขั้นตอนการทำงานส่วนของไฟเตือนสภาพการใช้งานกระบอกสูบ.....43
3.14 แสดงหน้าต่าง Log in ของผู้ดูแล.....45
3.15 แสดงหน้าต่าง Log in ของผู้ใช้งาน.....45
3.16 แสดงหน้าต่างโปรแกรมหลักของผู้ดูแล.....46
3.17 แสดงหน้าต่างโปรแกรมการกรอกข้อมูลของผู้ใช้.....47
3.18 แสดงหน้าต่างข้อมูลของผู้ใช้.....47
3.19 แสดงหน้าต่างรายละเอียดข้อมูลของผู้ที่เข้าใช้.....48
3.20 แสดงหน้าต่างแก้ไขข้อมูลของผู้ที่เข้าใช้.....48
3.21 แสดงหน้าต่างแก้ไขข้อมูล Username และ Password ของผู้ที่เข้าใช้.....49
3.22 แสดงหน้าต่างโปรแกรมการควบคุมและแสดงผล.....49
3.23 แสดงหน้าต่างโปรแกรมแสดงค่าจำนวนครั้งการทำงานของกระบอกสูบ.....50
3.24 แสดงหน้าต่างโปรแกรมแก้ไขข้อมูลเมื่อมีการซ่อมบำรุงกระบอกสูบ.....51
3.25 แสดงปุ่มที่มีในหน้าโปรแกรมหลักของผู้ใช้.....52
3.26 แสดงปุ่มที่มีในหน้าโปรแกรมหลักของผู้ใช้.....52
4.1 แสดงหน้าต่างโปรแกรมหลัก.....53
4.2 แสดงหน้าต่างโปรแกรมการควบคุมและการแสดงผล.....54
4.3 ส่วนของการแสดงผลสถานะของกระบอกสูบและเซนเซอร์.....55
4.4 ส่วนของการแสดงกราฟ Timing Diagrams.....56
4.5 หน้าจอการควบคุมแบบอัตโนมัติและแบบกำหนดเอง.....56

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญภาพ(ต่อ)

หน้า

4.6 แสดงโปรแกรมแสดงข้อมูลจำนวนครั้งการทำงานของกระบอกสูบ.....57
4.7 แสดงหน้าต่างโปรแกรมแก้ไขข้อมูลเมื่อมีการซ่อมบำรุงกระบอกสูบ.....58
4.8 แสดงสภาวะปกติไฟเตือนสภาพ แสดงสีเขียว.....64
4.9 สภาวะการทำงานมากกว่า 5 ครั้ง ไฟเตือนสภาพแสดงสีเหลือง.....64
4.10 สภาวะการทำงานมากกว่า 10 ครั้ง ไฟเตือนสภาพแสดงสีแดง.....65
4.11 แสดงจำนวนครั้งการทำงานของกระบอกสูบทั้งหมด.....66
4.12 แสดงการทำงานเมื่อมีการกดปุ่ม “Clear”.....66
4.13 แสดงการค่ากลับเป็นค่าเริ่มต้น หลังจากกดปุ่ม “Clear”.....67

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

1.1 ความสำคัญของปริญญานิพนธ์

ในปัจจุบัน เทคโนโลยีได้ถูกพัฒนาขึ้นอย่างรวดเร็ว มีความสะดวกสบาย และตอบสนองต่อความต้องการของผู้ใช้มากยิ่งขึ้น เทคโนโลยีนั้นเข้ามามีบทบาทอย่างมากต่อการดำรงชีวิตของมนุษย์ ในปัจจุบันแม้แต่ในทางอุตสาหกรรมก็เช่นกัน เทคโนโลยี ก็เข้ามามีบทบาทอย่างมาก เพื่อให้การผลิตในโรงงานหรือการควบคุมต่างๆ มีผลตอบสนองที่รวดเร็ว และสามารถทำงานได้ดีกว่าแรงงานของมุนษย์ โดยในขั้นตอนการผลิตของโรงงานอุตสาหกรรมนั้น จะต้องมีระบบแบบแผนของโรงงาน มีการทำงานเป็นลำดับขั้นตอน มีกระบวนการต่างๆเข้ามาตามลำดับ ไม่ว่าจะเป็นตั้งแต่การคัดเลือกวัตถุดิบ การผสมกันของสารเคมี ตลอดไปจนถึงการบรรจุลงภาชนะ ล้วนแต่ต้องใช้การทำงานของเครื่องมือเป็นลำดับให้สัมพันธ์กัน โดยเครื่องมือต่างๆเหล่านี้จะทำงานไม่ได้ ถ้าขาดโปรแกรมการควบคุมการทำงาน และรูปแบบที่จะให้เครื่องมือทำงาน รวมไปถึงการที่จะใช้เครื่องมือเพื่อควบคุมอะไร และใช้อะไรเป็นตัวขับเคลื่อนเครื่องมือ (เช่น แรง ไฟฟ้า นิวเมติก) ซึ่งในปัจจุบันก็มีเทคโนโลยีที่สามารถเขียนโปรแกรมเพื่อให้อุปกรณ์เหล่านี้ไปควบคุมการทำงานได้

จากความสนในการทำงานของเครื่องจักรในอุตสาหกรรม เราจึงเกิดแนวคิดว่าลักษณะที่ใช้ในการควบคุม และหลักการทำงานเป็นอย่างไร โดยการเขียนคำสั่งให้ตัวควบคุม (Programmable Logic Control : PLC) เพื่อทำการควบคุมวาล์วควบคุมโดยใช้ไฟฟ้าให้ทำงานตามคำสั่ง โดยจะทำงานจำลองเป็นลำดับขั้นตอนเสมือนเครื่องจักรในโรงงานอุตสาหกรรม และทำให้เห็นภาพรวมของการทำงานแบบลำดับขั้นตอนมากยิ่งขึ้นโดยแสดงผลออกทางจอมอนิเตอร์ ซึ่งจะรับค่าจากตัวควบคุมมาแสดงผลบนหน้าจอ HMI (Human Machine Interface) แบบเรียลไทม์

1.2 วัตถุประสงค์ของปริญญานิพนธ์

1. ศึกษาการทำงานแบบเป็นลำดับขั้นตอน
2. เขียนโปรแกรมให้ตัวควบคุม (Programmable Logic Control : PLC) เพื่อควบคุมระบบให้เป็น ตามต้องการ
3. สามารถสังเกตการณ์ของระบบที่เราสนใจผ่านโปรแกรมที่แสดงผลบนจอคอมพิวเตอร์
4. เป็นแนวทางในการประยุกต์การควบคุมอุปกรณ์ให้เป็นลำดับขั้นตอน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเอกสารนี้ทุกครั้งที่มีการนำไปใช้

1.3 ขอบเขตของปริญญานิพนธ์

1. สามารถเขียนโปรแกรมผ่านทางคอมพิวเตอร์ให้กับตัวควบคุม เพื่อให้ทำงานตามที่ต้องการ
2. สามารถนำสัญญาณจากตัวควบคุมไปควบวาล์วควบคุมด้วยไฟฟ้า (Solenoid Valve)
3. สามารถรับค่าจากตัวควบคุมมาแสดงผลในหน้า Human Machine Interface (HMI)

1.4 ขั้นตอนการศึกษา

1. ศึกษาการทำงานของระบบแบบเป็นลำดับขั้นตอน
2. ศึกษาการทำงานของระบบนิวเมติก อุปกรณ์ที่เกี่ยวข้อง และหลักการทำงานของอุปกรณ์
3. ศึกษาการใช้โปรแกรม Cx-Programmer เพื่อทำการเขียนคำสั่งให้ตัวควบคุม (Programmable Logic Control : PLC)
4. ศึกษาอุปกรณ์ที่ใช้เชื่อมต่อระหว่างคอมพิวเตอร์กับตัวควบคุม ชนิด RS-232 เช่น ข้อกำหนดต่างๆ ในการเชื่อมต่อ
5. ศึกษาข้อมูลเกี่ยวกับตัวควบคุม และกำหนดสัญญาณขาเข้า และสัญญาณขาออก
6. ศึกษาการใช้โปรแกรม Visual Basic 2010 เพื่อทำการเขียนโปรแกรมหน้าจอ HMI
7. เขียนโปรแกรมตามรูปแบบของการทำงานให้กับตัวควบคุม โดยจะมีทั้งหมด 5 รูปแบบในการทำงานของระบบ
8. ทำการเชื่อมต่ออุปกรณ์ต่างๆเข้าด้วยกัน โดยต่อคอมพิวเตอร์เข้ากับตัวควบคุมในทางด้านสัญญาณขาเข้า และต่อวาล์วควบคุมด้วยไฟฟ้าทางด้านสัญญาขาออก
9. ทำการเขียนหน้าจอ HMI ด้วยโปรแกรม Visual Basic 2010 โดยจะแสดงผลทั้งสถานะของลูกสูบ สถานะของเซนเซอร์ และกราฟแสดงการทำงานของระบบ ในรูปแบบเวลาจริง
10. ทำการศึกษาโปรแกรมแอ็กเซส (MS Access) เพื่อทำการบันทึกค่าการทำงานของลูกสูบลงในตารางซ่อมบำรุง เสมือนตารางซ่อมบำรุงของโรงงานอุตสาหกรรม

1.5 ประโยชน์ที่คาดว่าจะได้รับ

ในปริญญานิพนธ์นี้เป็นการควบคุมชุดลูกสูบขับเคลื่อนด้วยระบบนิวเมติก โดยการเขียนโปรแกรมผ่านทางคอมพิวเตอร์ให้กับตัวควบคุมเพื่อทำการควบคุมเครื่องมือให้เป็นไปตามที่ต้องการ และแสดงผลการทำงานบนจอมอนิเตอร์แบบเวลาจริง

ซึ่งสามารถนำปริญญานิพนธ์นี้มาอ้างอิงเพื่อจะได้เข้าใจถึงหลักการการทำงานที่เป็นลำดับขั้นตอน และเห็นภาพมากขึ้นสำหรับการทำงานในโรงงานอุตสาหกรรม

เอกสารนี้เป็นเอกสาร... ด้านการค้า ไม่ว่ากรณีใด... เอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2
ทฤษฎีที่ใช้ในการออกแบบ

2.1 กล่าวนำ

การควบคุมการทำงานของกระบอกสูบนั้น จะเป็นการนำเอาคอมพิวเตอร์เข้ามาใช้ทำการควบคุม โดยจะทำการเขียนโปรแกรมสั่งงานเพื่อกำหนดเงื่อนไขที่ได้กำหนดไว้โดยใช้โปรแกรม Visual Basic 2010 มาทำการสร้างเงื่อนไขในการกำหนดการทำงานของกระบอกสูบให้ทำงานตามลำดับที่กำหนด ซึ่งโปรแกรม Visual Basic 2010 จะนำเอาเหตุการณ์ที่เกิดขึ้นกับตัวคอนโทรลต่างๆ ใน Visual Basic 2010 มาใช้ในการตัดสินใจตามเงื่อนไขและส่งข้อมูลที่เป็นดิจิตอลออกผ่านทางพอร์ตอนุกรมผ่านสาย RS-232 ไปที่ตัวควบคุม Programmable Logic Control : PLC ซึ่งจะทำหน้าที่ในการขับเอาท์พุตรีเลย์ 24 VDC ให้ทำการเปิด-ปิด (ON-OFF) ซึ่งในนี้จะเป็นการขับวาล์วควบคุมด้วยไฟฟ้า ให้ทำการควบคุมแรงดันลมอัดที่จ่ายให้กระบอกสูบต่อไป

2.2 ทฤษฎีและหลักการของ Visual Basic 2010

ภาษาวิชวลเบสิกนั้นได้มีการพัฒนามาตั้งแต่รุ่นแรกๆที่ทำงานบนระบบปฏิบัติการ DOS จนมาเป็น Visual Basic 6 ในปี 1998 และเมื่อมีการพัฒนา .NET Framework ทำให้เกิดการเปลี่ยนแปลงทางด้านโครงสร้างของภาษา BASIC ครั้งใหญ่จนกลายมาเป็น Visual Basic.NET ที่มีใช้กันอยู่ในปัจจุบัน

2.2.1 จุดเด่นของ Visual Basic

2.2.1.1 สร้างแอพพลิเคชั่นได้ง่ายและรวดเร็ว

โปรแกรม Visual Basic ได้รับการยอมรับให้เป็นเครื่องมือที่ช่วยในการสร้างแอพพลิเคชั่นให้เป็นไปได้อย่างรวดเร็วและง่ายดาย เพื่อลดเวลาในการสร้างแอพพลิเคชั่นให้สั้นลง ซึ่งเรียกรูปแบบอันนี้ว่า Rapid Application Development หรือ RAD เพราะได้ทำการขจัดงานที่โปรแกรมเมอร์ต้องทำซ้ำๆ ออกไปเหลือเฉพาะที่ต้องโฟกัสเกี่ยวกับปัญหาที่เกิดขึ้นของงานจริงๆ แล้วเขียนโปรแกรมจัดการปัญหานั้นๆ ส่วนเรื่องอื่นๆ จะให้ Visual Basic จัดการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.1.2 การเขียนโปรแกรมที่ง่ายต่อการเรียนรู้

ถ้าหากมีโอกาสได้เขียนโปรแกรมด้วย Visual Basic แล้วจะเห็นว่าภาษ Basic ในตัวของ Visual Basic นั้นอ่านง่าย คืออ่านแล้วใกล้เคียงกับภาษาที่เราใช้งานกันปกติ ซึ่งทำให้สื่อความหมายเข้าใจได้ง่ายกว่าภาษาของโปรแกรมอื่นๆ ทำให้ผู้ที่เริ่มต้นในการเขียนโปรแกรมสามารถทำความเข้าในกับการเขียนโปรแกรมได้อย่างรวดเร็ว

2.2.1.3 รวมเครื่องมืออำนวยความสะดวกในการเขียนโปรแกรม

Visual Basic มีเครื่องช่วยในการเขียนโปรแกรมที่ไม่ยุ่งยาก เพราะจะมีเครื่องมือที่ช่วยให้ไม่ต้องจดจำไวยากรณ์ภาษาที่ยุ่งยาก และจะตรวจสอบโดยอัตโนมัติว่าโปรแกรมที่เขียนนั้นมีความถูกต้องตามหลักของภาษาหรือไม่ ซึ่งจะมีการแยกแยะส่วนของโปรแกรมอย่างเป็นระเบียบ ทำให้งานของโปรแกรมเมอร์นั้นลดลงได้มาก

นอกจากจะมีเครื่องช่วยในการเขียนโปรแกรมแล้ว ยังมีเครื่องมือที่ใช้ทดสอบแก้ไขโปรแกรม (Debugger) ที่เขียนขึ้นมาว่าทำงานได้ถูกต้องหรือไม่ มีระบบขอความช่วยเหลือ (Online Help) ไว้อ้างอิง และขอความช่วยเหลือในจุดที่เราสงสัยได้

เครื่องมือทั้งหมดที่กล่าวมาข้างต้น จะถูกจัดรวมไว้ในสภาพแวดล้อมการทำงานเดียวกันเรียกย่อๆว่า IDE ซึ่งย่อมาจาก Integrate Development Environment ทำให้เรียกใช้งานได้สะดวก ตั้งแต่เขียนโปรแกรม, ทดสอบ, แก้ไข, สร้างชุดติดตั้ง รวมทั้งระบบการขอช่วยเหลือ ซึ่งเราสามารถเพิ่มเติมเครื่องมือชนิดใหม่ๆเข้าไปได้ หรือถอดเครื่องมือที่ไม่จำเป็นต้องใช้ออกเพื่อประหยัดพื้นที่ฮาร์ดดิสก์ได้เช่นกัน

2.2.2 รูปแบบการพัฒนาแอพพลิเคชั่นด้วย Visual Basic

เมื่อมองเห็นว่าโปรแกรม Visual Basic สามารถช่วยให้เราทำการสร้างแอพพลิเคชั่นบน Windows ได้ง่าย และรวดเร็วแล้ว ยังมีรูปแบบหนึ่งที่โปรแกรม Visual Basic สร้างขึ้นมาได้อีก คือ

2.2.2.1 พัฒนาแอพพลิเคชั่นกับ ActiveX Control

เทคโนโลยีที่มีชื่อว่า ActiveX เป็นตัวอยู่เบื้องหลังความสำเร็จของ Visual Basic ซึ่งช่วยลดงานที่ซ้ำซ้อนของการเขียนโปรแกรม

ตัวอย่างเช่น การเขียนโปรแกรมเพื่อรับค่าของข้อมูลจากผู้ใช้แต่เดิมจะต้องมาเขียนโปรแกรมเพื่อวาดหน้าจอ, เขียนโปรแกรมวาดรูปของปุ่ม และช่องรับข้อความรวมทั้งเขียนโปรแกรมเพื่อจัดการกับข้อมูลที่ผู้ใช้ป้อนเข้ามา แต่ใน ActiveX จะทำให้เราสนใจเฉพาะการจัดการกับส่วนของ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารนี้ทุกครั้งที่มีการนำไปใช้

ข้อมูลที่ผู้ใช้ป้อนเข้ามาเท่านั้น ที่เหลือโปรแกรม Visual Basic จะจัดการให้เอง โดยช่องรับข้อความและปุ่มต่างๆนั้นเราจะทำการใช้ ActiveX Control จัดการ

นอกจากจะลดความซับซ้อนลงแล้วการที่เราใช้ ActiveX Control ช่วยในการเขียนโปรแกรมจะทำให้โปรแกรมที่เราเขียนกับโปรแกรมที่ผู้อื่นเขียนนั้น ตั้งอยู่บนมาตรฐานเดียวกัน ทำให้การปรับปรุงโปรแกรมทำได้ง่าย

2.2.2.2 สร้างแอพพลิเคชั่นที่ใช้งานกับฐานข้อมูล

เป็นแอพพลิเคชั่นที่มีการใช้งานกันมากที่สุดเพราะระบบร้านค้า, คลังสินค้า, ระบบบัญชี, ระบบบริหารงานบุคคลหรือแม้แต่ E-Commerce ทั้งหมดต่างๆ ต้องมีส่วนที่ติดต่อกับฐานข้อมูลที่แน่นอน

โปรแกรม Visual Basic ได้ช่วยในการสร้างแอพพลิเคชั่นกับฐานข้อมูลเป็นเรื่องที่ทำได้ง่ายดาย เพราะมีเครื่องมือต่างๆ ที่อำนวยความสะดวกในการเขียนโปรแกรมเพื่อใช้งานข้อมูลจากฐานข้อมูล แลไม่จำกัดด้วยว่าเป็นฐานข้อมูลแบบใดทั้งฐานข้อมูลส่วนบุคคล, ฐานข้อมูลผ่านเครือข่าย, หรือฐานข้อมูลผ่านอินเตอร์เน็ต จากความสามารถที่หลากหลายนี้จึงทำให้ Visual Basic เหมาะกับการสร้างแอพพลิเคชั่นที่เกี่ยวกับฐานข้อมูลในโลกธุรกิจยุคปัจจุบันนี้

2.2.2.3 สร้างแอพพลิเคชั่นแบบใหม่กับอินเตอร์เน็ต

ปัจจุบันอินเตอร์เน็ตนับว่ามีความสำคัญกับชีวิตของคนที่ใช้ไอทีมากขึ้น ซึ่ง Visual Basic เปิดโอกาสให้เราสามารถสร้างแอพพลิเคชั่นเพื่อรองรับการทำงานร่วมกับอินเตอร์เน็ตได้ด้วย

จุดเด่นของ Visual Basic อีกข้อหนึ่งคือ เปิดโอกาสให้เรานำแอพพลิเคชั่นปกติที่เดิมทำงานกับคอมพิวเตอร์ส่วนบุคคล สามารถดัดแปลงให้ใช้กับงานอินเตอร์เน็ตได้ไม่ยากทำให้ไม่ต้องทิ้งงานเดิมที่มีอยู่

2.2.3 เริ่มสร้างแอพพลิเคชั่นด้วย Visual Basic

ขั้นที่ 1: ออกแบบแอพพลิเคชั่น

ต้องทราบให้แน่ชัดก่อนว่าแอพพลิเคชั่นที่เราจะสร้างนั้นใช้ประโยชน์อะไร, ต้องมีความสามารถอะไรบ้าง, มีรูปร่างหน้านาเป็นอย่างไร ซึ่งจำเป็นที่จะต้องคิดให้รอบคอบและเขียนออกมาให้ชัดเจน โดยอาจจะร่างคร่าวๆ ในกระดาษเป็นต้น

ขั้นที่ 2: ตกแต่งหน้าแอพพลิเคชั่น

เป็นการตกแต่งรูปร่างของแอพพลิเคชั่น ตามที่ได้ออกแบบไว้ พร้อมกับการกำหนดค่าพร็อพเพอร์ตี้ต่างๆ ให้กับคอนโทรลแต่ละตัวในแอพพลิเคชั่นนั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขั้นที่ 3: เขียนโค้ดกำกับแอพพลิเคชั่น

ขั้นตอนการเขียนโค้ด หรือการเขียนโปรแกรมเพื่อทำการควบคุมการทำงานของแอพพลิเคชั่นนั้น เราจะใช้การเขียนโปรแกรมแบบ Event Driven Programming ซึ่งจะเป็นการเขียนโค้ดเพื่อรองรับเหตุการณ์ต่างๆที่จะเกิดขึ้นกับตัวคอนโทรลต่างๆ ในแอพพลิเคชั่นของเรา

ขั้นที่ 4: การทดสอบการทำงานของแอพพลิเคชั่น

เมื่อเขียนโค้ดเสร็จเรียบร้อยแล้ว ก็มาถึงเวลาที่จะทำการทดสอบการทำงานของแอพพลิเคชั่นที่ได้ทำการสร้างขึ้น ซึ่งประกอบไปด้วยคอนโทรลต่างๆที่ปรับแต่งไว้และโค้ดที่เขียนเพื่อจัดการกับเหตุการณ์ต่างๆ

ขั้น 5: บันทึกเก็บไว้ในคอมพิวเตอร์

หลังจากทดสอบจนแน่ใจว่าแอพพลิเคชั่นที่สร้างนั้นทำงานได้ถูกต้อง เราจึงจะบันทึกเก็บไว้ ซึ่งสามารถแก้ไขและเพิ่มเติมความสามารถอื่นๆได้ในภายหลัง

ขั้นที่ 6: สร้างไฟล์ .EXE (Make)

เมื่อเราสร้างแอพพลิเคชั่นเสร็จแล้ว เราอาจจะต้องการที่จะนำเอาแอพพลิเคชั่นที่ได้สร้างขึ้นมานั้น ให้สามารถเรียกใช้งานได้เองโดยไม่ต้องเรียกผ่านโปรแกรม Visual Basic หรือต้องการนำไปใช้งานในคอมพิวเตอร์เครื่องอื่นๆ ซึ่งเราจะทำได้โดยการสร้างไฟล์เอ็กซ์คิวต์ (ที่มีนามสกุล .EXE)

2.3 ฐานข้อมูล (Database)

ฐานข้อมูล (Data Base) คือ วิธีการจัดเก็บข้อมูลที่สัมพันธ์กันอย่างมีระเบียบ ซึ่งจะทำให้ง่ายต่อการงานและค้นหาข้อมูล ซึ่งฐานข้อมูลที่คนส่วนใหญ่คุ้นเคยคือ ฐานข้อมูลเชิงสัมพันธ์ เป็นรูปแบบการจัดเก็บข้อมูลที่มีความสัมพันธ์กัน โดยจะมองข้อมูลในลักษณะของตารางต่างๆ ที่มีความสัมพันธ์กัน

2.3.1 องค์ประกอบของฐานข้อมูล

การใช้งานฐานข้อมูลจะต้องประกอบด้วยองค์ประกอบดังต่อไปนี้

2.3.1.1 แอพพลิเคชั่นฐานข้อมูล (Database Application)

แอพพลิเคชั่นฐานข้อมูล เป็นแอพพลิเคชั่นที่สร้างไว้ให้ผู้ใช้ติดต่อกับฐานข้อมูลได้อย่างสะดวก ซึ่งมีรูปแบบของการติดต่อกับฐานข้อมูลแบบ เมนู หรือกราฟฟิก โดยไม่จำเป็นต้องมีความรู้มากก็สามารถเรียกใช้ฐานข้อมูลได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3.1.2 ระบบจัดการฐานข้อมูล (DBMS)

ระบบจัดการฐานข้อมูล เป็นซอฟแวร์ที่ทำหน้าที่จัดการข้อมูลในฐานข้อมูล ทั้งการจัดเก็บ การแสดงผล การค้นหา การสำรองข้อมูล เป็นต้น โดยจะเป็นเครื่องมือในการทำงานของผู้บริหารข้อมูล และเป็นตัวกลางที่เชื่อมผ่านระหว่างแอพพลิเคชั่นฐานข้อมูลที่สร้างขึ้นกับตัวข้อมูลในฐานข้อมูล เช่น Microsoft Access, FoxPro เป็นต้น

2.3.1.3 ดาต้าเบสเซิฟเวอร์ (Database Server)

ดาต้าเบสเซิฟเวอร์ เป็นคอมพิวเตอร์ที่ให้บริการในหารจัดการฐานข้อมูล ซึ่งก็คือคอมพิวเตอร์ที่มีระบบจัดการฐานข้อมูล

2.3.1.4 ข้อมูล (Data)

ข้อมูล คือ ตัวเนื้อหาของข้อมูลที่เราใช้งาน ซึ่งจะถูกเก็บในหน่วยความจำของดาต้าเบสเซิฟเวอร์ โดยจะถูกเรียกมาใช้งานจากระบบจัดการฐานข้อมูล

2.3.1.5 ผู้บริหารฐานข้อมูล (Database Administrator)

ผู้บริหารฐานข้อมูลเป็นคนที่ทำหน้าที่ในการดูแลข้อมูลที่มีในฐานข้อมูลผ่านระบบจัดการฐานข้อมูล ซึ่งจะทำการควบคุมให้การทำงานนั้นเป็นไปอย่างราบรื่นและยังทำหน้าที่ในการกำหนดผู้ที่มีสิทธิ์ในการใช้งานฐานข้อมูล จะกำหนดในเรื่องความปลอดภัยของการใช้งาน พร้อมทั้งดูแล Database Server ให้ทำงานอย่างปกติด้วย

ฐานข้อมูลมีหลักการพื้นฐานมาจากคณิตศาสตร์ ในเรื่องทฤษฎีบทของเซท (Theory of Set) โดยที่การจัดเก็บหรือแสดงข้อมูลให้ผู้ใช้เห็นจะเป็นแบบตาราง

ข้อมูลต่างๆจะแสดงในรูปของตาราง โดยในแต่ละตารางนั้นจะเป็นการจัดรวบรวมข้อมูลที่ประเภทเดียวกันไว้ด้วยกัน แต่ละแถวที่ประกอบขึ้นเป็นตารางนั้นก็คือ Record ซึ่งจะเป็นที่เก็บข้อมูลแต่ละชุดของตารางนั้น และในแต่ละแถวก็จะประกอบด้วยคอลัมน์ซึ่งเป็นหน่วยย่อยที่แสดงคุณสมบัติของข้อมูลแต่ละแถว

ในแต่ละตารางมักจะมีบางคอลัมน์หรืออาจจะหลายๆ คอลัมน์ประกอบกันที่สามรถบอกถึงความแตกต่างของฐานข้อมูลในแต่ละแถวได้ คอลัมน์ดังกล่าวเรียกว่า Primary Key เช่น คอลัมน์เลขประจำตัวพนักงานที่แตกต่างกันทุกคน และในการเชื่องโยงตารางฐานข้อมูลที่มีหลายตารางนั้น จะกระทำได้ก็ต่อเมื่อแต่ละตารางมีคอลัมน์เกี่ยวโยงกัน ซึ่งคอลัมน์ที่เกี่ยวโยงนี้เรียกว่า foreign key

เอกสารนี้เป็นสงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3.2 ชุดคำสั่งที่ใช้จัดการกับข้อมูล

แบ่งออกเป็น 2 ประเภทหลักๆคือ

1. จัดการกับโครงสร้างฐานข้อมูล เป็นการสร้าง ลบ หรือการแก้ไขฐานข้อมูล และตารางเช่น สร้างตารางฐานข้อมูลของพนักงานขึ้นมาชุดคำสั่งต่างๆ ที่เกี่ยวกับการสร้างหรือเปลี่ยนแปลงโครงสร้างของฐานข้อมูลมีศัพท์เรียกว่า Data Definition Language หรือ DDL
2. จัดการกับส่วนของฐานข้อมูลที่มีอยู่ในตารางต่างๆ ในฐานข้อมูล เป็นการทำงานในเรื่องของการเพิ่ม การลบ หรือการแก้ไขข้อมูลในตารางต่างๆ เช่น การเพิ่มข้อมูลพนักงานใหม่เข้าไปในตารางข้อมูลพนักงาน คำสั่งประเภทนี้เรียกว่า Data Manipulation Language หรือ DML

2.3.3 โครงสร้างของภาษา

Structure Query Language หรือ SQL เป็นภาษามาตรฐานที่ใช้ในการจัดการกับฐานข้อมูลโดยเฉพาะที่เป็นแบบแอพพลิเคชั่นที่ใช้งานกับฐานข้อมูลแบบของ SQL เป็นภาษาที่มีประสิทธิภาพสูงสามารถเรียนรู้และใช้งานอย่างง่ายดาย จึงทำให้เป็นที่นิยมอย่างรวดเร็ว นอกจากนี้ยังสามารถใช้กับการจัดการฐานข้อมูลโดยผ่านโปรแกรมก็ได้ เช่น หากเราใช้ Microsoft Access เราก็สามารถพิมพ์คำสั่ง SQL ได้โดยตรงในโปรแกรม Access หรือ จะใช้คำสั่ง SQL ผ่าน Visual Basic ก็ได้ เราจะต้องใช้ SQL เพื่อทำการจัดการข้อมูลในฐานข้อมูลได้หลายอย่าง เช่น การแสดงข้อมูลจากฐานข้อมูลแบบมีเงื่อนไข การเพิ่ม การลบ และการนำข้อมูลจากตารางหลายๆตารางมาแสดงร่วมกันได้เป็นต้น

ภาษาของ SQL ประกอบไปด้วย 3 ส่วนใหญ่ๆคือ

1. Data Definition Language (DDL) เป็นกลุ่มคำสั่งในภาษา SQL ที่ใช้สำหรับโครงสร้างของฐานข้อมูล เช่น การสร้างฐานข้อมูล ปรับปรุงโครงสร้างของฐานข้อมูล เป็นต้น
2. Data Manipulation Language (DML) เป็นกลุ่มคำสั่งในภาษา SQL ที่ใช้สำหรับจัดการข้อมูลในฐานข้อมูล เช่น การแสดงข้อมูลแบบมีเงื่อนไข การลบข้อมูล การเพิ่มข้อมูล และการแสดงที่มาจากตารางหลายตาราง เป็นต้น
3. กลุ่มฟังก์ชั่น Aggregate Function เป็นฟังก์ชั่นพิเศษของภาษา SQL ที่มีหน้าที่เฉพาะอย่าง เช่น หาผลรวมของเรคคอร์ด ค่าสูงสุด ค่าต่ำสุด เป็นต้น เป็นกลุ่มฟังก์ชั่นที่เป็นประโยชน์มาก เพราะจะช่วยลดภาระให้ไม่ต้องเขียนโค้ดจัดการเอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำหรับการใช้งานภาษา SQL ร่วมกับ Visual Basic เพื่อจัดการฐานข้อมูล จะใช้งานกลุ่มคำสั่ง DML ร่วมกับกลุ่มฟังก์ชั่น Aggregate และกำหนดเงื่อนไขโดยใช้ตัวดำเนินการด้านต่างๆ

คำสั่งในกลุ่ม DML จะมีคำสั่งพื้นฐานอยู่ 4 คำสั่งคือ

1. DELETE เป็นคำสั่งที่ใช้สำหรับลบข้อมูลหรือลบเรคคอร์ดใดๆในตาราง
2. INSERT เป็นคำสั่งที่ใช้สำหรับเพิ่มข้อมูลหรือเพิ่มเรคคอร์ดใดๆเข้าไปในตาราง
3. SELECT เป็นคำสั่งที่ใช้สำหรับเลือกข้อมูลหรือแสดงเรคคอร์ดที่ต้องการ
4. UPDATE ใช้สำหรับแก้ไขข้อมูลหรือแก้ไขเรคคอร์ดใดๆในตาราง

2.3.3.1 ตัวดำเนินการ (Operator)

ตัวดำเนินการเปรียบเทียบที่น่าสนใจ ได้แก่

=	คือ เท่ากับ (Equality)
<>	คือ ไม่เท่ากับ (Inequality)
<	คือ น้อยกว่า (Less Than)
>	คือ มากกว่า (Greater Than)
<=	คือ น้อยกว่าหรือเท่ากับ (Less Than or Equal To)
>=	คือ มากกว่าหรือเท่ากับ (Greater Than or Equal To)

2.3.3.2 ตัวดำเนินการด้านตรรกะ (Logical Operator)

ตัวดำเนินการด้านตรรกะที่นิยมใช้จะมีอยู่ 3 คือ And, Or, Not

2.3.3.3 กลุ่มฟังก์ชัน Aggregate

กลุ่มฟังก์ชัน Aggregate เป็นฟังก์ชันที่ใช้คำนวณทางคณิตศาสตร์ โดยช่วยให้การนำเสนอผลการค้นหาเป็นไปอย่างมีประสิทธิภาพ ซึ่งมีฟังก์ชันพื้นฐานดังต่อไปนี้

AGV ()	หน้าที่ หาค่าเฉลี่ยของฟิลด์จากเร็คคอร์ดทั้งหมด
COUNT ()	หน้าที่ นับจำนวนเร็คคอร์ด
FIRST	หน้าที่ หาค่าแรกในฟิลด์
LAST	หน้าที่ หาค่าสุดท้ายในฟิลด์
MAX	หน้าที่ หาค่ามากที่สุด หรือค่าสูงสุด
MIN	หน้าที่ หาค่าน้อยที่สุด หรือค่าต่ำสุด
SUM	หน้าที่ หาผลรวมทั้งหมดของฟิลด์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3.4 ลักษณะการใช้งานของกลุ่มคำสั่ง DML

เป็นคำสั่งที่ใช้สำหรับ ลบ หรือ เร็คคอร์ดใดๆ ออกจากตารางมีรูปแบบการใช้งาน 2 ลักษณะใหญ่ๆ คือ

รูปแบบที่ 1	DELETE FORM	ชื่อตาราง WHERE เงื่อนไข
รูปแบบที่ 2	DELETE FORM	ชื่อตาราง

ชื่อตารางนี้ หมายถึงชื่อของตารางที่ต้องการลบส่วนเงื่อนไขจะหมายถึงเงื่อนไขที่ใช้ในการลบข้อมูล หรือเร็คคอร์ดในตารางนั้น

คำสั่ง INSERT

เป็นคำสั่งที่ใช้สำหรับเพิ่มข้อมูล หรือเพิ่มเร็คคอร์ดเข้าไปในตารางในกรณีที่ฟิลด์เป็นข้อมูลชนิด TEXT ต้องใช้เครื่องหมาย “,”กำกับด้วยฟิลด์นั้นด้วยมีรูปแบบการใช้งาน 2 ลักษณะ

รูปแบบที่ 1 INSERT INTO table name (field1, field2,...) VALUES(value1, value2,...)
รูปแบบที่ 2 INSERT INTO table name 1

ตัวแปร Table name	หมายถึง ชื่อตารางที่ต้องการเพิ่มเร็คคอร์ดเข้าไป
ตัวแปร Table channel	หมายถึง เลือกข้อมูลจากตารางที่ชื่อ tablename1 ตามเงื่อนไขที่ระบุไว้ในตัวแปร criteria แล้วนำมาเพิ่มที่ตาราง tablename2
ตัวแปล Field	หมายถึง ชื่อของฟิลด์ต่างๆที่อยู่ในตารางที่ต้องการเพิ่มข้อมูล
ตัวแปล Value	หมายถึงค่าของฟิลด์จะเพิ่มเข้าไปโดยที่ต้องระบุค่าให้ตรงกับฟิลด์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คำสั่ง SELECT

สำหรับเลือกดึงข้อมูล (Retrieve Data) ที่เราต้องการ จากตารางที่ระบุไว้ในคำสั่งที่มีความยืดหยุ่นสูงมาก เพราะว่าเงื่อนไขในการนำข้อมูลออกมาจากตารางมีมากมายหลายลักษณะแต่มีรูปแบบการใช้งานพื้นฐานอยู่ 2 ลักษณะคือ

```
SELECT*FORM ชื่อตาราง
```

หรือ

```
Select ฟิลด์ที่1,ฟิลด์ที่2,...,ฟิลด์ที่n FORM ชื่อตาราง WHERE เงื่อนไข
```

ชื่อตาราง	หมายถึง ชื่อตารางที่ต้องการดึงข้อมูล
ตัวแปร ฟิลด์ที่1,ฟิลด์ที่2,...,ฟิลด์ที่n	หมายถึง ชื่อฟิลด์ที่ต้องการดึงข้อมูลถ้ามีมากกว่า 1 ฟิลด์จะใช้เครื่องหมาย,คั่นระหว่างฟิลด์
เงื่อนไข	หมายถึง เงื่อนไขในการดึงข้อมูลอาจเป็นคำสั่งทางคณิตศาสตร์ หรือเป็นคำสั่ง SELECT ซ้อนอยู่ข้างไหนก็ได้

คำสั่ง UPDATE

เป็นคำสั่งที่ใช้สำหรับแก้ไข หรือเปลี่ยนแปลงข้อมูลในเร็คคอร์ดที่มีอยู่ในตาราง มีรูปแบบการใช้งานดังต่อไปนี้

```
UPDATE ชื่อตาราง SET ชื่อฟิลด์ = ค่าที่กำหนด WHERE เงื่อนไข
```

2.3.5 การเชื่อโยงตาราง

การเชื่อมโยงตารางโดยใช้คำสั่ง SELECT

วิธีนี้เป็นการเชื่อมโยงตาราง โดยระบุฟิลด์ที่มีความสัมพันธ์กัน เช่น

```
SELECT Student, FirstName,LastName,MajorDesc
FROM Student, Major
WHERE Student, MajorID = majored
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น ไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การเชื่อมโยงตารางโดยใช้คำสั่ง INNER JOIN

เป็นการเชื่อมโยงของข้อมูลจาก 2 ตารางเข้าด้วยกันโดยระบุชื่อฟิลด์ที่มีความสำคัญกันหลังคำสั่ง ON เป็นเงื่อนไขที่ใช้การ JOIN ตาราง โดยที่ฟิลด์ดังกล่าว จะต้องเป็นข้อมูลชนิดเดียวกัน มีลักษณะเช่นเดียวกับการใช้คำสั่ง SELECT มีรูปแบบการใช้งานดังนี้

```
SELECT fieldname
FORM table name 1
INNER JOIN table name 2
ON tablename1.fieldname1 OPERATION tablename2.fieldname 2
```

ตัวแปรFieldname	หมายถึงชื่อฟิลด์ที่อยู่ในตารางที่1 หรือ 2
ตัวแปรTablename1	หมายถึงชื่อตารางที่1
ตัวแปรTablename2	หมายถึงชื่อตารางที่2
ตัวแปรFieldname1	หมายถึงชื่อฟิลด์ที่อยู่ตารางที่ 1
ตัวแปรFieldname2	หมายถึงชื่อฟิลด์ที่อยู่ตารางที่ 2
ตัวแปรOperation	หมายถึงเงื่อนไขหรือตัวดำเนินการต่างๆที่ใช้ในการเชื่อมโยง

การเชื่อมโยงโดยใช้คำสั่ง LEFT JOIN

การเชื่อมโยงของ 2 ตารางเข้าด้วยกัน โดยกำหนดให้ตารางแรก (ตารางทางซ้าย ของคำสั่ง LEFT JOIN) เป็นหลัก แล้วนำตารางที่ 2 เข้ามาทำการเชื่อมโยงตามเงื่อนไขที่ได้ระบุเอาไว้ มีรูปแบบการใช้งานได้ดังนี้

```
SELECT fieldname
FORM table name 1 LEFT JOIN table name 2
ON tablename1.fieldname1 OPERATION tablename2.fieldname 2
```

ตัวแปรFieldname	หมายถึงชื่อฟิลด์ที่อยู่ในตารางที่1 หรือ 2
ตัวแปรTablename1	หมายถึงชื่อตารางที่1
ตัวแปรTablename2	หมายถึงชื่อตารางที่2
ตัวแปรFieldname1	หมายถึงชื่อฟิลด์ที่อยู่ตารางที่ 1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวแปรFieldname2	หมายถึงชื่อฟิลด์ที่อยู่ตารางที่ 2
ตัวแปรOperation	หมายถึงเงื่อนไขหรือตัวดำเนินการต่างๆที่ใช้ในการเชื่อมโยง

การเชื่อมโยงคำสั่งโดยให้คำสั่ง RIGHT JOIN

จะเป็นการเชื่อมโยง 2 ตารางเข้าด้วยกัน แต่กำหนดให้ตารางที่อยู่ทางด้านขวาของคำสั่ง RIGHT JOIN เป็นหลัก แล้วนำเร็คคอร์ดของตารางที่อยู่ทางด้านซ้ายของคำสั่ง RIGHT JOIN เข้ามาร่วมแสดงเฉพาะที่เร็คคอร์ดที่ตรงตามเงื่อนไขที่ระบุไว้ในเครื่องแบบการใช้งานดังนี้

```
SELECT fieldname
FORM table name 1 RIGHT JOIN table name 2
ON tablename1.fieldname1 OPERATION tablename2.fieldname 2
```

ตัวแปรFieldname	หมายถึงชื่อฟิลด์ที่อยู่ในตารางที่1 หรือ 2
ตัวแปรTablename1	หมายถึงชื่อตารางที่1
ตัวแปรTablename2	หมายถึงชื่อตารางที่2
ตัวแปรFieldname1	หมายถึงชื่อฟิลด์ที่อยู่ตารางที่ 1
ตัวแปรFieldname2	หมายถึงชื่อฟิลด์ที่อยู่ตารางที่ 2
ตัวแปรOperation	หมายถึงเงื่อนไขหรือตัวดำเนินการต่างๆที่ใช้ในการเชื่อมโยง

2.3.6 เทคโนโลยี ActiveX Data Object

ADO เป็นเทคโนโลยีที่มีแนวความคิดมาจาก DAO บางส่วน กล่าวคือจะมองฐานข้อมูลเป็นออบเจ็กต์เช่นกัน แต่จะใช้ OLEDB Provider เป็นตัวจัดการ ข้อมูลและโครงสร้างของฐานข้อมูลแทน โดยที่จะไม่นิยามออบเจ็กต์ขึ้นมา เพื่อแทนโครงสร้างของฐานข้อมูลแต่ละส่วนเหมือนกับ DAO แต่จะใช้วิธีการสร้าง OLEDB Provider ให้กับ RDBMS แต่ละชนิดแทน เช่น เมื่อฐานข้อมูลเป็นชนิด JET ก็จะใช้ Microsoft Jet OLEDB Provider ถ้าเป็น Oracle ก็จะใช้ Microsoft OLEDB Provider for Oracle เป็นต้น ทำให้ออบเจ็กต์ในโมเดล ADO ไม่ยุ่งยากซับซ้อนเหมือนออบเจ็กต์ DAO ใน Visual Basic สามารถเรียกใช้ OLEDB Provider ได้ 2 วิธีคือ

1. อาศัยคอนโทรล ADO Data (ADO Data Control) ร่วมกับกลุ่มคอนโทรล Bound & ActiveX Bound Control ที่มีคำว่า OLEDB ต่อท้าย
2. โดยเรียกใช้งานกลุ่มออบเจ็กต์ ADO โดยตรง

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ชุดออบเจ็กต์ใน ADO 2.5

สำหรับในโมเดลของ ADO 2.5 จะประกอบด้วยออบเจ็กต์ 9 ตัวคือ

ออบเจ็กต์ Command	เป็นออบเจ็กต์ตัวกลาง ที่ใช้สำหรับการส่งคำสั่ง (Execute) ไปยังฐานข้อมูลเป้าหมายที่ต้องการติดต่อ โดยมีศัพท์เรียกฐานข้อมูลดังกล่าวว่า Data Sources
ออบเจ็กต์ Start	ใช้สำหรับเริ่มต้นเปิดการเชื่อมต่อกับฐานข้อมูล
ออบเจ็กต์ Error	จะเป็นออบเจ็กต์ที่เก็บข้อมูลเกี่ยวข้องกับการผิดพลาดในส่วนของการเข้าถึงของข้อมูลของ OLEDB Provider
ออบเจ็กต์ Field	ใช้สำหรับเก็บฟิลด์ใดฟิลด์หนึ่งที่อยู่ในฐานข้อมูล
ออบเจ็กต์ Parameter	ใช้สำหรับในการรับ-ส่ง ตัวแปร หรือ อากิวเมนต์ระหว่างการทำคิวรี่หรือ Stored Procedure ในออบเจ็กต์ Command
ออบเจ็กต์ Property	จะใช้สำหรับในการทำการกำหนดค่าของคุณสมบัติต่างๆ ของออบเจ็กต์ Record set
ออบเจ็กต์ Record	จะใช้สำหรับในการเก็บค่าของเร็คคอร์ดใดเร็คคอร์ดหนึ่ง ที่อยู่ในออบเจ็กต์ Record set
ออบเจ็กต์ Record set	จะเป็นผลลัพธ์ที่ได้จากฐานข้อมูลโดยจะประกอบด้วยกลุ่มของเร็คคอร์ด และฟิลด์ที่ได้มาจากตารางที่ เชื่อมต่อเป็นออบเจ็กต์ที่มีความสำคัญมากที่สุด

2.3.7 วิธีติดต่อกับฐานข้อมูลของ Visual Basics 10

โปรแกรม Visual Basics 10 มีความสามารถในการจัดการกับฐานข้อมูลได้โดยจะมีวิธีการจัดการอยู่ 2 แบบ คือ

1. การใช้ Data Control เป็นวิธีการที่ง่ายและมีความสะดวกที่สุดในการที่จะทำการติดต่อกับฐานข้อมูลเนื่องจาก Data Control จะทำการติดต่อกับฐานข้อมูลและจัดการกับฐานข้อมูลในตารางโดยอัตโนมัติเช่น การเปิดฐานข้อมูล การแสดงและแก้ไขข้อมูลในตาราง อย่างไรก็ดีการใช้ Data Control ยังมีข้อจำกัดอยู่พอสมควร เช่น ไม่มีฟังก์ชั่นในการลบข้อมูล

2. การใช้ Data Object วิธีนี้จะต้องเขียนโปรแกรมเพื่อทำการติดต่อกับฐานข้อมูล โดยการใช้ Object ต่างๆ ที่ Visual Basics 10 มีมาให้โดยการเพิ่ม ลบ หรือการแก้ไขข้อมูลจะต้องทำการเขียนโปรแกรมเอง แต่ข้อดีของวิธีนี้คือสามารถติดต่อข้อมูลจากหลายๆ ตารางพร้อมกัน สามารถสร้างคิวรีตอนรันโปรแกรมได้ และสามารถควบคุมความผิดพลาดต่างๆ ได้ดีกว่าใช้ Data Control รวมทั้งสามารถใช้ภาษา SQL เพื่อจัดการกับฐานข้อมูลได้อีกด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3.7.1 การใช้งานดาต้าคอนโทรล (Data Control)

ดาต้าคอนโทรลเป็นเครื่องมือที่ช่วยให้สามารถจัดการกับข้อมูลในตาราง เช่นการเพิ่ม ลบ แก้ไข หรือการค้นหาข้อมูลที่ถูกต้องเป็นต้น

คุณสมบัติที่สำคัญสำหรับดาต้าคอนโทรล

Data Source	เป็นการกำหนดชื่อของดาต้าคอนโทรลที่ต้องการเชื่อมโยง
Delafield	เป็นการกำหนดว่าเราต้องการให้แสดงหรือแก้ไขข้อมูลใดในตาราง
Database Name	เป็นที่อยู่ของไฟล์ฐานข้อมูลที่ต้องการ โดยมีการระบุชื่อฟิลด์นั้น
Connect	เป็นคุณสมบัติที่ใช้บอกว่าต้องการติดต่อกับฐานข้อมูลประเภทใด เช่น MS-Access
RecordSetType	เป็นชนิดของเร็คคอร์ดเซตที่ต้องการเข้าถึงฐานข้อมูล

2.4 แลดเดอร์ไดอะแกรม (Ladder Diagrams)

การเขียนโปรแกรม PLC ในปัจจุบันนิยมใช้ภาษาที่เรียกว่าแลดเดอร์ไดอะแกรม (Ladder Diagram) หรือลอจิกแลดเดอร์ (Ladder logic) จัดเป็นภาษาสัญลักษณ์ที่เทียบเคียงมาจากวงจรรีเลย์ สามารถทำความเข้าใจการทำงานได้ง่ายด้วยรูปภาพสัญลักษณ์ จึงทำให้เป็นที่นิยมใช้งานกันอย่างกว้างขวางในการเขียนโปรแกรมควบคุม PLC โดยเฉพาะมือใหม่ทั้งหลาย

ปัจจุบันการเขียน Ladder ของ PLC ส่วนใหญ่จะใช้ซอฟท์แวร์ในการเขียนภาพสัญลักษณ์ซึ่งทำให้ง่ายและสะดวกมากขึ้น แต่ในการทำงานจริงของ PLC ไม่ได้ทำงานด้วยรูปภาพแต่จะอาศัยชุดคำสั่ง (Instructions) โดยการเขียนลงในหน่วยความจำเป็นรหัส (Mnemonic code) ซึ่งไม่สามารถจัดเก็บในลักษณะของแลดเดอร์ไดอะแกรมได้โดยตรง ดังนั้นผู้ใช้งานจึงจำเป็นต้องทำความเข้าใจชุดคำสั่งซึ่งเป็นลอจิกพื้นฐานต่างๆ เพื่อสร้างความเข้าใจและสามารถนำไปประยุกต์ใช้งานได้

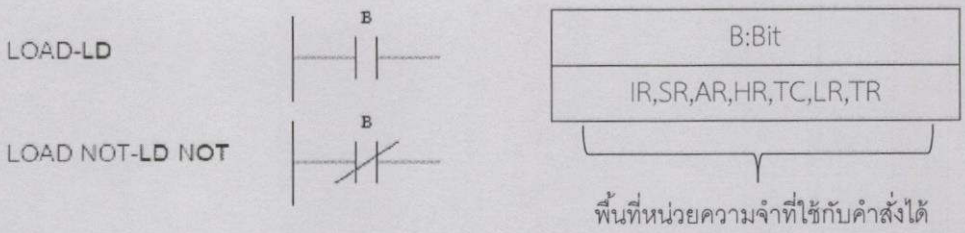
ท่านสามารถค้นหาข้อมูล PLC หรือคู่มือการใช้งาน PLC ได้จากเว็ปไซต์ของ PLC ยี่ห้อนั้นๆ ได้ ซึ่งคู่มือดังกล่าวจะมีรายละเอียดเกี่ยวกับการเขียนโปรแกรมด้วยเช่นกัน

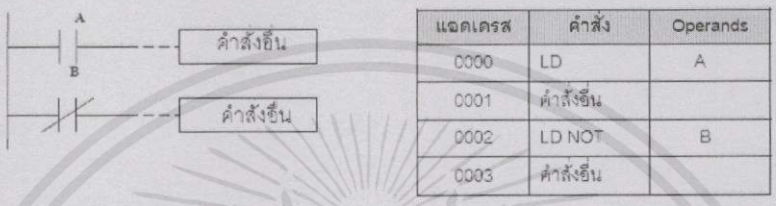
2.4.1 กลุ่มคำสั่งลอจิกพื้นฐาน

PLC แต่ละยี่ห้อจะมีรูปแบบของคำสั่ง Mnemonic ที่คล้ายกัน ในที่นี้เราจะอ้างอิงคำสั่ง PLC ของ Omron เพื่อใช้ประกอบคำอธิบาย ส่วนแอดเดรสในตารางคำสั่งหมายถึงตำแหน่งหน่วยความจำที่ PLC เก็บคำสั่งและลำดับในการประมวลผล การเขียน Ladder ของ Omron น่าจะเป็นต้นแบบในการเรียนรู้ที่ดีสำหรับผู้เริ่มต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4.1.1 การใช้คำสั่ง LOAD (LD), LOAD NOT (LD NOT)





แอดเดรส	คำสั่ง	Operands
0000	LD	A
0001	คำสั่งอื่น	
0002	LD NOT	B
0003	คำสั่งอื่น	

รูปที่ 2.1 แสดงแลดเดอร์ไดอะแกรมของคำสั่ง LOAD (LD), LOAD NOT (LD NOT)

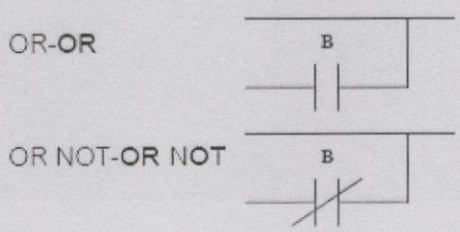
2.4.1.2 การใช้คำสั่ง AND, AND NOT

แอดเดรส	คำสั่ง	Operands
0000	LD	A
0001	AND NOT	B
0002	AND	C
0003	คำสั่งอื่น	

รูปที่ 2.2 แสดงแลดเดอร์ไดอะแกรมของคำสั่ง AND, AND NOT

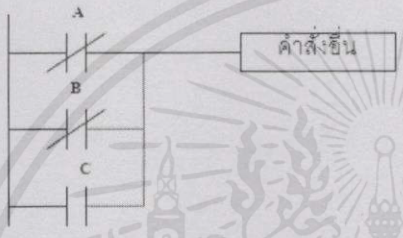
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4.1.3 การใช้คำสั่ง OR, OR NOT



B:Bit
IR,SR,AR,HR,TC,LR

พื้นที่หน่วยความจำที่ใช้กับคำสั่งได้

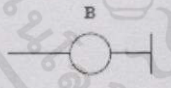


แอดเดรส	คำสั่ง	Operands
0000	LD NOT	A
0001	OR NOT	B
0002	OR	C
0003	คำสั่งอื่น	

รูปที่ 2.3 แสดงแลดเดอร์ไดอะแกรมของคำสั่ง OR, OR NOT

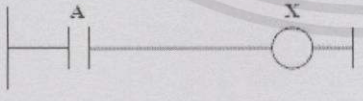
2.4.1.4 การใช้คำสั่ง OUT

OUTPUT-OUT



B:Bit
IR,SR,AR,HR,LR,TR

พื้นที่หน่วยความจำที่ใช้กับคำสั่งได้



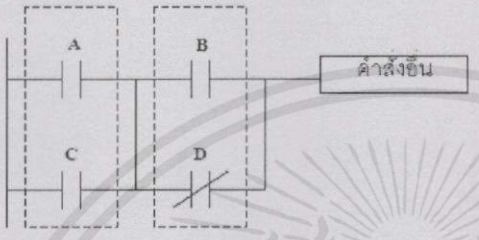
แอดเดรส	คำสั่ง	Operand
0000	LD	A
0001	OUT	X

รูปที่ 2.4 แสดงแลดเดอร์ไดอะแกรมของคำสั่ง OUT

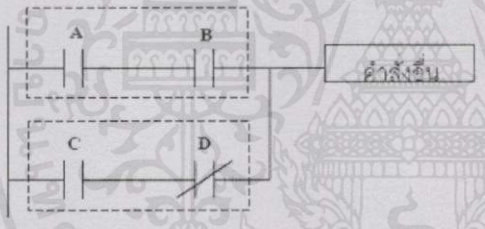
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4.1.5 การใช้คำสั่ง AND LOAD (AND LD), OR LOAD (OR LD)

คำสั่งทั้งสองจะทำหน้าที่เชื่อมต่อกลุ่มแลดเดอร์ไดอะแกรม ในกรณีที่ต่ออนุกรม หรือขนานกันมากกว่า 1หน้าสัมผัส ซึ่งการใช้คำสั่ง AND หรือ OR นั้น จะกระทำทีละ 1 หน้าสัมผัสเท่านั้นจึงต้องใช้ AND LD หรือ OR LD

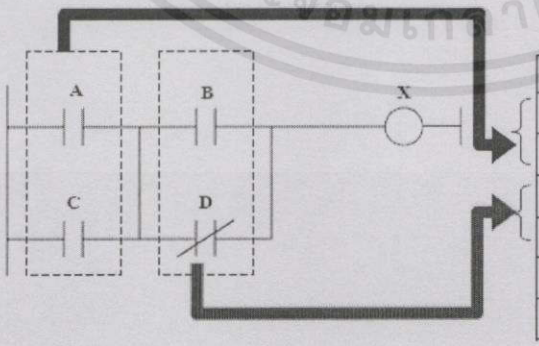


แอดเดรส	คำสั่ง	Operands
0000	LD	A
0001	OR	C
0002	LD	B
0003	OR NOT	D
0004	AND LD	



แอดเดรส	คำสั่ง	Operands
0000	LD	A
0001	AND	B
0002	LD	C
0003	AND NOT	D
0004	OR LD	---

- AND LD เป็นคำสั่งในการเชื่อมโปรแกรม 2 บล็อก ในแบบอนุกรม



แอดเดรส	คำสั่ง	Operands
0000	LD	A
0001	OR	C
0002	LD	B
0003	OR NOT	D
0004	AND LD	---
0005	OUT	X

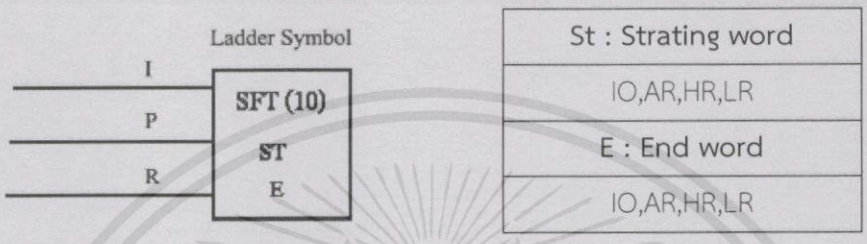
รูปที่ 2.5 แสดงแลดเดอร์ไดอะแกรมของคำสั่ง AND LOAD (AND LD), OR LOAD (OR LD)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4.1.6 การใช้คำสั่ง END (END)

เมื่อสิ้นสุดการเขียนโปรแกรมแล้วจะต้องจบด้วยคำสั่ง END เสมอ ถ้าไม่มีคำสั่งนี้ เมื่อผู้ใช้สั่งให้ PLC ทำงานจะเกิด Error โดยสังเกตได้จากไฟ Error/Alarm สีแดงจะติดค้าง แต่ซอท์ฟแวร์ของ PLC หลายยี่ห้อจะช่วยเพิ่มคำสั่งนี้ให้อัตโนมัติเพื่อป้องกันการลืม

2.4.1.7 การใช้งานคำสั่ง SFT (10)



รูปที่ 2.6 แสดงส่วนประกอบของคำสั่ง SHT(10)

ST: กำหนด CH หรือ WORD เริ่มต้นของการ SHIFT

E : กำหนด CH หรือ WORD สุดท้ายของการ SHIFT

คำสั่งนี้เป็นคำสั่งเลื่อนข้อมูล " 0 " หรือ " 1 " ของ CHANNEL ที่ต้องการเลื่อนข้อมูล ซึ่งประกอบด้วย 3 อินพุท ดังนี้

DATA INPUT (I) หมายถึง เป็นค่าข้อมูลที่กำหนด " 0 " หรือ " 1 "

PLUSE INPUT (P) หมายถึง เป็นจังหวะพัลซ์หรือสเต็ปของการเลื่อนข้อมูล

RESET INPUT (R) หมายถึงการเซ็ทข้อมูลเพื่อเริ่มต้น

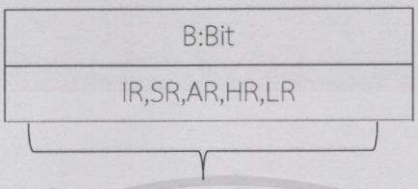
2.4.1.8 การใช้งานคำสั่ง IL (02), ILC (03)

คำสั่ง IL และ ILC จะต้องใช้ร่วมกันคือ ถ้าเริ่มต้นมีการใช้คำสั่งด้วย IL เมื่อใดแล้วถ้าต้องการสิ้นสุดการทำงานต้องจบด้วย ILC เงื่อนไขของคำสั่งคือ คอนแทคตรงหน้าส่วนของ IL มีสภาวะ “ON” จะทำให้โปรแกรมที่อยู่ระหว่าง IL และ ILC ทำงานเป็นปกติ แต่ถ้าคอนแทคตำแหน่งดังกล่าวมีสภาวะ “OFF” จะทำให้การทำงานของโปรแกรมระหว่าง IL และ ILC ไม่ทำงาน ในขณะเดียวกัน Output Coil ในช่วงนั้นจะมีสภาวะ “OFF” ด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

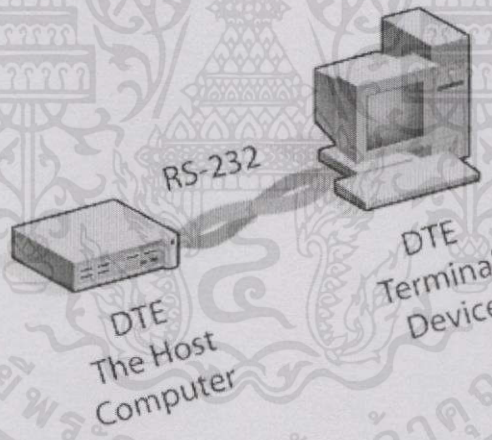
2.4.1.9 การใช้งานคำสั่ง Differential UP-DIFU(013) และ Differential DOWN-DIFD(014)

การใช้งานของคำสั่งนี้ จะทำงานเพียงขอบขาขึ้น หรือขอบขาลงของอินพุตเท่านั้น และจะทำงานเพียง One cycle time เท่านั้น



2.5 การติดต่อสื่อสารและส่งผ่านข้อมูล

การติดต่อสื่อสารแบบ Host Link เป็นการเชื่อมต่อกับตัวควบคุมแบบ PLC กับคอมพิวเตอร์ทั่วไปผ่านทางพอร์ทของคอมพิวเตอร์ (COM 1 ,2 ,3 ,4 ,5 ฯลฯ) ส่วนมากจะนิยมใช้มาตรฐานในการส่งแบบอนุกรมเพื่อทำให้สามารถควบคุม PLC จากคอมพิวเตอร์ได้



รูปที่ 2.7 แสดงการเชื่อมต่อคอมพิวเตอร์กับตัวควบคุมผ่านพอร์ตอนุกรม RS232

สำหรับคอมพิวเตอร์หนึ่งเครื่องสามารถที่ที่จะต่อเข้ากับตัวควบคุมแบบ PLC ได้เป็นจำนวนมากโดยใช้การเชื่อมต่อหลายๆ ตัวเข้าด้วยกัน เรียกว่า PC Link โดยจะใช้ในการติดต่อแบบ Host Link จะต้องผ่านอุปกรณ์ที่เรียกว่า Host Link Units ซึ่งจะต้องทำการตั้งค่าต่างๆ ที่จำเป็นที่จะทำการใช้ในการติดต่อแบบ Host Link Units ผ่านสวิตช์ของเครื่องควบคุมแบบ PLC

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

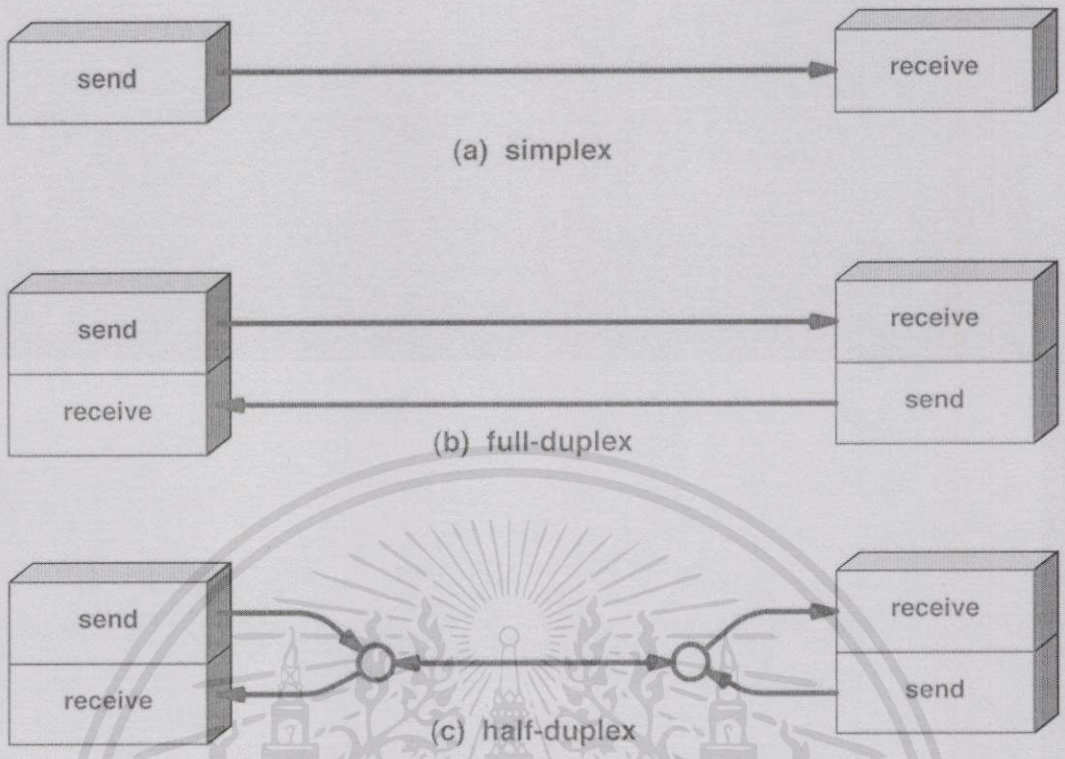
2.5.1 การสื่อสารข้อมูลแบบอนุกรม

ถ้าทำการกล่าวถึงเรื่องของการสื่อสารของข้อมูลแบบอนุกรมแล้วนั้น แสดงว่าจะต้องมีการสื่อสารข้อมูลแบบขนานด้วย ซึ่งการสื่อสารแบบขนานก็คือข้อมูลในทุกๆ บิทในแต่ละเวิร์ดจะถูกส่งออกไปพร้อมๆกัน ขึ้นอยู่กับว่าเวิร์ดดังกล่าวมีขนาดเท่าไร ทั่วไปก็คือ 1 ไบร์ หรือ 8 บิทนั้นเอง การส่งข้อมูลแบบขนานนี้จะมีข้อกำหนดทางด้านระยะทาง ซึ่งโดยทั่วไปจะส่งในระยะไม่เกิน 3 – 5 ฟุต เท่านั้น ทั้งนี้ยังขึ้นอยู่กับอัตราความเร็วที่ใช้ในการส่งข้อมูลด้วย ยิ่งอัตราในการส่งสูงก็จะได้ระยะทางที่มีความสั้นลง การส่งข้อมูลแบบขนานนั้นนิยมในระบบที่ต้องการความเร็วสูงมากๆ แต่อุปกรณ์ไม่อยู่ห่างกันมากนัก ส่วนการส่งข้อมูลแบบอนุกรมนั้น ข้อมูลจะถูกทยอยส่งออไปทีละบิทจนหมดครบทั้งเวิร์ด โดยส่งผ่านสัญญาณเพียงเส้นเดียวและในการใช้งานจริงนั้นจะต้องมีสายสัญญาณอีกเส้นเป็นสายสัญญาณกราวด์ (Ground) ดังนั้น เมื่อเราส่งข้อมูลในแบบอนุกรมเราจะสามารถใช้สายสัญญาณอย่างน้อยที่สุดเพียง 2 เส้น ในขณะที่ส่งข้อมูลแบบขนานจะต้องใช้อย่างน้อยเท่ากับจำนวนบิทบวกกับสายสัญญาณระดับแรงดัน Ground อีก 1 เส้น ที่สำคัญการส่งข้อมูลแบบอนุกรมนั้นจะสามารถส่งได้ไกลกว่า เช่น ถ้าส่งตามมาตรฐานของ RS-232 ที่จะกล่าวต่อไปในภายหลังจะสามารถส่งได้ไกลถึง 30 ถึง 40 ฟุต โดยไม่ต้องใช้อุปกรณ์ขับสัญญาณเพิ่มเติมแต่เดิมอย่างใด อย่างไรก็ตามในการส่งข้อมูลยังมีข้อกำหนดบางประการเพื่อให้ได้รับข้อมูลที่ถูกต้อง แม่นยำ และมีความน่าเชื่อถือสูง จะต้องมีวิธีตรวจสอบความถูกต้องของข้อมูลได้ในระหว่างที่มีการสื่อสารกันอยู่ด้วย ซึ่งจะได้กล่าวต่อไป

2.5.1.1 การส่งข้อมูลแบบซิมเพล็กซ์ (Simplex) และ ดูเพล็กซ์ (Duplex)

ในการสื่อสารไม่ว่าจะเป็นการสื่อสารข้อมูล หรือเป็นการสื่อสารทั่วๆไปนั้นย่อมจะต้องประกอบด้วยผู้รับและผู้ส่งแน่นอนตายตัวอยู่ตลอดเวลา เช่น การสื่อสารข้อมูลระหว่างเครื่องคอมพิวเตอร์กับเครื่องพิมพ์ เป็นต้น การสื่อสารของอุปกรณ์ที่มีผู้รับและผู้ส่งตายตัวนั้น เราเรียกว่าการสื่อสารแบบซิมพล็กซ์ กล่าวคือ การสื่อสารเป็นไปในลักษณะทิศทางเดียวตลอดเวลาซึ่งจะมีที่ใช้ไม่มากนัก การสื่อสารโดยทั่วไปนั้นจะเป็นแบบดูเพล็กซ์ คือมีทิศทางการสื่อสารเป็นแบบสองทิศทางมีทั้งไปและกลับ โดยการสื่อสารในลักษณะดูเพล็กซ์นั้นแบ่งออกได้เป็น 2 ชนิด คือ แบบฮาร์ฟดูเพล็กซ์ (Half Duplex) นิยมเขียนย่อกันว่า HEX ซึ่งจะมีทิศทางในการสื่อสารในลักษณะที่ผลัดกันเป็นผู้ส่งและผู้รับพร้อมกัน และแบบฟูลดูเพล็กซ์ (Full Duplex) นิยมเขียนย่อว่า FDX จะมีทิศทางการสื่อสารในลักษณะสัญญาณรับสัญญาณหนึ่ง และสัญญาณส่งอีกทิศทางหนึ่งหรือกล่าวได้อีกนัยหนึ่งว่า สัญญาณรับและส่งจะมีสายตัวนำสัญญาณแยกออกจากันโดยเด็ดขาด

เอกสารนี้เป็น[illegible]นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



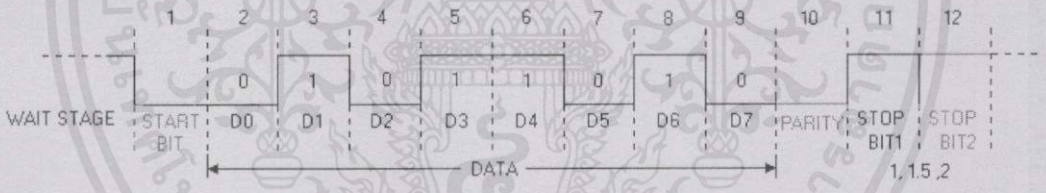
รูปที่ 2.8 การสื่อสารแบบอนุกรม

2.5.1.2 โปรโตคอลของการสื่อสารแบบอนุกรม

เมื่อพิจารณาการส่งข้อมูลแบบอนุกรมให้ดีจะพบว่ามีปัญหาหนึ่งที่เกิดขึ้นอยู่เสมอ ก็คือ การตัดสินว่าข้อมูลที่ได้รับนั้นมีจุดเริ่มต้นที่ใด ดังนั้น จึงมีการกำหนดข้อตกลงในการสื่อสารขึ้นเพื่อแก้ปัญหานี้ โดยข้อตกลงดังกล่าวจะเรียกว่า โปรโตคอล (Protocol) ของการสื่อสารข้อมูลแบบอนุกรม โดยสามารถที่จะแบ่งออกเป็น 2 ประเภทใหญ่ๆ ด้วยกันคือ โปรโตคอลสื่อสารข้อมูลแบบซิงโครนัส (Synchronous) และแบบอะซิงโครนัส (Asynchronous) โดยการสื่อสารข้อมูลแบบซิงโครนัสนี้ ข้อมูลจะถูกทำการส่งออกมาอย่างสม่ำเสมอและช่วงเวลาระหว่างบิทและเวิร์ด จะมีค่าที่เท่ากันเสมอ ส่วนการสื่อสารข้อมูลแบบอะซิงโครนัสจะเป็นหัวใจของการสื่อสารข้อมูลผ่านทางสายโทรศัพท์ ในปัจจุบันซึ่งการสื่อสารแบบนี้ช่วงระยะเวลาระหว่างบิทจะมีค่าที่เท่ากันกับแบบซิงโครนัส แต่จะมีระยะห่างระหว่างเวิร์ดนั้นแตกต่างกันไปเป็นวินาที นาที ชั่วโมง หรือ วัน เป็นต้น ขึ้นอยู่กับฝ่ายรับสามารถรอได้หรือไม่เท่านั้น เมื่อไม่มีข้อกำหนดทางด้านระยะเวลาระหว่างเวิร์ดแล้ว ทางผู้ส่งและผู้รับจะเข้าใจกันตรงกันได้อย่างไร ว่าที่ใดคือจุดที่เริ่มต้นและจุดที่สิ้นสุดของแต่ละเวิร์ด เพื่อแก้ไขปัญหาตรงนี้ จึงมีการกำหนดข้อตกลงเกี่ยวกับรูปแบบของข้อมูลที่ใช้ในการทำการส่งให้ทางผู้รับสามารถเข้าใจว่าจุดใดที่เป็นจุดเริ่มต้นของเวิร์ด ข้อกำหนดดังกล่าวกำหนดให้แต่ละเวิร์ดจะต้องขึ้นต้นด้วยบิทที่ เรียกว่า บิทเริ่มต้น (Start Bit) ซึ่งจะต้องมีข้อมูลเป็นลอจิก 0 เสมอ จากนั้นตามด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บิทข้อมูลที่ต้องการส่งมีความยาว 5 ถึง 8 บิท ถัดจากบิทข้อมูลก็จะเป็นบิทพาริตีบิท ซึ่งทำหน้าที่เป็นบิทสำหรับตรวจสอบความถูกต้องของข้อมูลที่ได้รับว่ามีความถูกต้องหรือไม่ บิทพาริตีนี้มี 2 ประเภท คือ อีเวนพาริตี (Even Parity) ซึ่งจะกำหนดจำนวนบิทที่เป็นลอจิก 1 ในบิทที่เป็นข้อมูลมีจำนวนเป็นคี่ ในการส่งข้อมูลบางครั้งอาจจะไม่มีการใช้บิทพาริตีก็ได้ ถ้าหากการสื่อสารในครั้งนั้นมีความน่าเชื่อถือสูง มีสัญญาณรบกวนต่ำเป็นการเพิ่มความเร็วในการสื่อสารได้ด้วยบิทสุดท้ายในรูปแบบก็คือ บิทสุดท้าย (Stop Bit) ทำหน้าที่บอกทางผู้รับว่าในขณะนี้ข้อมูลที่ทางผู้รับได้รับนั้นครบเวิร์ดแล้วขอให้เตรียมชุดรับเวิร์ดต่อไปได้ บิทสุดท้ายนี้ถูกกำหนดให้เป็นลอจิก 1 บิท หรือ 2 บิท ก็ได้จากรูปแบบดังกล่าวจะเห็นว่าเรามีรูปแบบสำหรับการสื่อสารมากมาย เช่น 5E1 (5Databit,Even Parity,1 Stop Bit),7E1(7Databit,Even Parity,1 Stop Bit) และ 8E1 (8Databit,Even Parity,1 Stop Bit) เป็นต้น ในการใช้งานทั่วไปเรานิยมใช้กันอยู่เพียง 2 รูปแบบคือ 7E1 และ 8E1 จะเลือกรูปแบบใดขึ้นอยู่กับสภาพของสายส่งสัญญาณว่ามีสัญญาณรบกวนมากเพียงใด ถ้าหากสายส่งมีสัญญาณรบกวนมากก็ควรจะใช้ 7E1 แต่ถ้าสายส่งสัญญาณมีสภาพดีสัญญาณรบกวนต่ำการใช้ 8E1 จะเร็วกว่า เป็นต้น ทั้งนี้ต้องมีการตกลงกันล่วงหน้าระหว่างทางผู้รับและผู้ส่งว่าจะใช้รูปแบบใดในการสื่อสาร ลักษณะของข้อมูลที่ถูกส่งออกไปจะมีลักษณะดังรูป



รูปที่ 2.9 การส่งข้อมูลผ่านการสื่อสารแบบอนุกรม

2.5.1.3 มาตรฐานสัญญาณอนุกรมแบบ RS232C

มาตรฐานการสื่อสารข้อมูลแบบอนุกรมกำหนดโดย EIA (Electronics Industries Association) มาตรฐาน RS232C ได้ถูกตีพิมพ์ในปี คศ. 1969 เริ่มต้นจากความต้องการที่จะกำหนดมาตรฐานการเชื่อมต่อระหว่างคอมพิวเตอร์กับโมเด็มในสมัยนั้น

- ตัวอักษร RS แทน “Recommend Standard”
- 232 แทนหมายเลขของมาตรฐาน
- ตัวอักษร C แทนให้รู้ว่ามาตรฐานได้รับการแก้ไขกี่ครั้ง

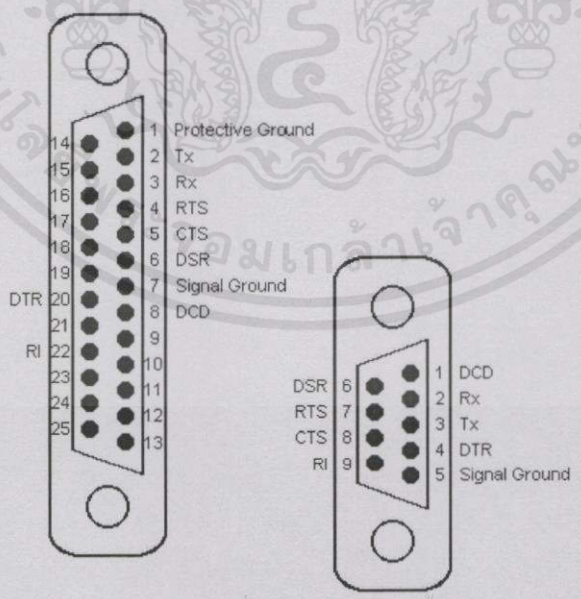
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4.1.3.1 ลักษณะสมบัติทางไฟฟ้าของการอินเทอร์เฟสแบบ RS232

- ถูกออกแบบให้ใช้กับอุปกรณ์พวกสัญญาณ Discrete
- ใช้การอินเทอร์เฟสแบบ Unbalanced
- ในแต่ละวงจรใช้ลวดนำในการนำสัญญาณ 1 เส้น และมีสายกราวด์รวมทุกวงจรอีก 1 เส้น
- อัตราเร็วในการส่งข้อมูลมีค่า < 20 กิโลบิทต่อวินาที (Kbps)
- ระยะทางสูงสุดในการส่งข้อมูลมีค่า < 15 เมตร
- ทำให้เกิด Crosstalk ที่มีค่ามาก

2.5.1.4 พอร์ตสื่อสารอนุกรม

พอร์ตสื่อสารอนุกรม (Serial Communication Device) หรือเรียกว่า "Serial Port" เครื่องคอมพิวเตอร์โดยปกติจะมีพอร์ตชนิดนี้อยู่แล้ว 2 พอร์ต คือ พอร์ตขนาด 9 ขา (9-pins) มีรูปร่างเหมือนสี่เหลี่ยมคางหมู มีเข็มยื่นออกมา 9 เข็ม เรียกหัวชนิดนี้ว่า "DB-9 Connector Male Type" อีกชนิดหนึ่งคือ พอร์ตขนาก 25 ขา (25-pins) มีรูปร่างเช่นเดียวกันกับแบบ 9 ขา แต่มีขนาดที่ยาวกว่า เรียกหัวชนิดนี้ว่า "DB-25 Connector Male Type"



รูปที่ 2.10 ตำแหน่งขาของ DB-9 และ DB-25

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

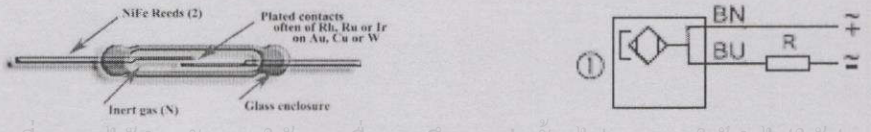
ตารางที่ 2.1 ขาสัญญาณต่างๆในการส่งข้อมูลแบบอนุกรม RS-232 Standard

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

มาตรฐานของการรับส่งข้อมูลแบบอนุกรม (RS-232) นี้ได้กำหนดขึ้นเพื่อให้คอมพิวเตอร์ต่างยี่ห้อกัน หรืออุปกรณ์ต่อพ่วงแต่ละชนิดรับส่งข้อมูลกันได้เมื่อทำตามมาตรฐาน ไม่สนใจว่าอุปกรณ์หรือคอมพิวเตอร์นั้นจะผลิตมาจากที่ใด

2.6 เซนเซอร์ตรวจจับแบบแม่เหล็กแบบรีดสวิตซ์

ความหมายของเซนเซอร์ตรวจจับแบบแม่เหล็ก (Magnetic Sensor) นั้น คือ การใช้สนามแม่เหล็กเข้ามาเกี่ยวข้อง ทำให้เกิดแรงเคลื่อนไฟฟ้า เพื่อให้ตัวเซนเซอร์สามารถตรวจจับและสัญญาณออกไปยังอุปกรณ์ที่รับค่า โดย **รีดสวิตซ์ (Reed switch)** คือ เซนเซอร์ตรวจจับแบบแม่เหล็กที่มีลักษณะเป็นแบบหน้าสัมผัส ซึ่งโดยปกติทั่วไปแล้ว จะเป็นหน้าสัมผัสแบบปกติเปิด (Normally Open : NO) รีดสวิตซ์นี้จะทำงานโดยอาศัยสนามแม่เหล็ก ซึ่งอาจจะเป็นแม่เหล็กถาวรหรือแม่เหล็กไฟฟ้าก็ได้ แผ่นหน้าสัมผัสจะทำมาจากสารที่มีผลต่อสนามแม่เหล็ก (ferromagnetic) และติดตั้งอยู่ภายในกระเปาะแก้วเล็กๆที่มีการเติมก๊าซเฉื่อย เพื่อทำให้การตัดต่อการส่งกระแสไฟฟ้าได้เร็วยิ่งขึ้น

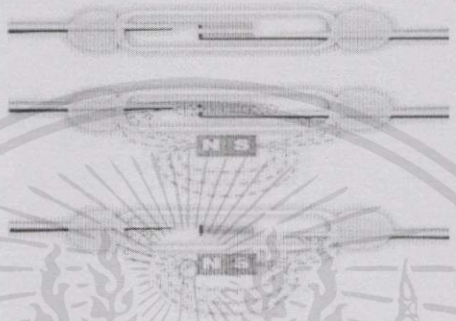


รูปที่ 2.11 วงจรและส่วนประกอบภายในของรีดสวิตซ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.6.1 หลักการการทำงานของรีดสวิตช์

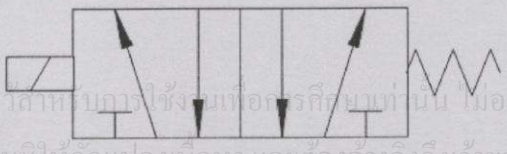
โดยปกติแล้วรีดสวิตช์นี้จะอยู่ในสถานะปิด คือ วงจรจะเปิดอยู่เพราะแท่งเหล็กทั้งสองแท่งไม่แตะกัน ทำให้ไม่สามารถที่จะส่งแรงดันไฟฟ้าถึงกันได้ แต่เมื่อมีการตรวจจับแม่เหล็กได้ แม่เหล็กจะทำการเหนี่ยวนำให้แกนเหล็กทั้งสองมาติดกันทำให้สวิตช์อยู่ในสถาเปิด วงจรจะปิดทำให้แรงดันไฟฟ้าสามารถไหลได้ครบวงจร



รูปที่ 2.12 การทำงานของรีดสวิตช์

2.7 วาล์วควบคุมโดยใช้ไฟฟ้า (Solenoid Valve)

โซลินอยด์ (Solenoid) เป็นอุปกรณ์แม่เหล็กไฟฟ้าชนิดหนึ่ง ที่มีหลักการทำงาน คล้ายกับรีเลย์(Relay) ภายในโครงสร้างของ โซลินอยด์จะประกอบด้วยขดลวดที่พันอยู่ รอบแท่งเหล็กที่ภายในประกอบด้วยแม่เหล็ก ชุดบนกับชุดล่าง เมื่อมีกระแสไฟฟ้าไหลผ่านขดลวดที่พันรอบแท่งเหล็ก ทำให้แท่งเหล็กชุดล่างมีอำนาจแม่เหล็กดึงแท่งเหล็กชุดบนลงมาสัมผัสกันทำให้ครบวงจรทำงาน เมื่อวงจรถูกตัดกระแสไฟฟ้าทำให้แท่งเหล็กส่วนล่างหมดอำนาจแม่เหล็ก สปริงก็จะดันแท่งเหล็กส่วนบนกลับ สู่ตำแหน่งปกติ จากหลักการดังกล่าวของ โซลินอยด์ก็จะนำมาใช้ในการเลื่อนลิ้นวาล์วของระบบนิวแมติกส์ไฟฟ้า โครงสร้างของ Solenoid Valve โดยทั่วไปแบ่งออกเป็น 2 ชนิดคือ เลื่อนวาล์วด้วยโซลินอยด์วาล์วกลับด้วยสปริง (Single Solenoid Valve) และเลื่อนวาล์วด้วยโซลินอยด์วาล์วกลับด้วยโซลินอยด์วาล์ว (Double Solenoid Valve) ที่ใช้อยู่ในปัจจุบัน เช่น โซลินอยด์วาล์ว 2/2, 3/2, 4/2 , 5/2 เป็นต้น โดยที่นำมาใช้จะเป็น **วาล์ว 5/2 ปกติปิด เลื่อนวาล์วด้วยโซลินอยด์วาล์วกลับด้วยแรงสปริง (5/2 Way Single Solenoid Valve)**

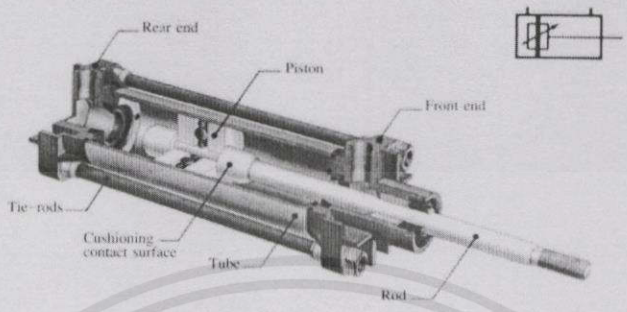


รูปที่ 2.13 วาล์วควบคุมด้วยไฟฟ้าแบบ 5/2

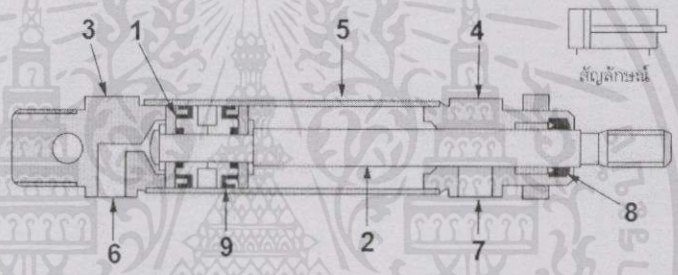
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.8 กระบอกสูบแบบ 2 ทาง

กระบอกสูบชนิดนี้จะมีลักษณะการทำงาน ได้ทั้งสองทิศทาง ด้วยการจ่ายลมอัดเข้ากระบอกที่หัว หรือท้ายกระบอกสูบ จะทำให้ก้านสูบเคลื่อนที่เข้าออก



รูปที่ 2.14 ส่วนประกอบของกระบอกสูบ



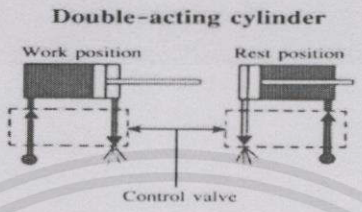
รูปที่ 2.15 โครงสร้างภายในของกระบอกสูบ

ตารางที่ 2.2 แสดงรายละเอียดโครงสร้างของกระบอกสูบชนิดทางานสองทิศทาง

หมายเลข	รายละเอียด
1	ลูกสูบ (piston)
2	ก้านสูบ (piston rod)
3	ฝาครอบท้าย (base end cover)
4	ฝาครอบหัว (head end cover)
5	กระบอกสูบ (cylinder tube)
6	รูต่อลมด้านลูกสูบ (pressure connector , base side)
7	รูต่อลมด้านก้านสูบ (pressure connector, head side)
8	ซีลก้านสูบ (bush and sealing element)
0	ซีลลูกสูบ (piston seal)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

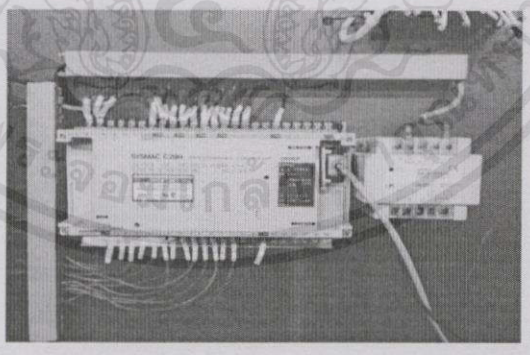
เมื่อจ่ายลมอัดเข้าที่ท้ายกระบอกสูบจะทำให้ก้านลูกสูบเคลื่อนที่ออกและเกิดการระบายลมที่ค้าง ในกระบอกสูบออกทางด้านหัวกระบอกสูบ เมื่อไม่มี ลมอัดจ่ายให้กระบอกสูบ ก้านสูบจะหยุดค้างอยู่ ณ ตำแหน่งสุดท้ายที่เคลื่อนที่ และเราสามารถใช้มือดึง ก้านสูบไปมาได้โดยอิสระอีกด้วย



รูปที่ 2.16 แสดงหลักการทำงานภายในของกระบอกสูบสองทางเมื่อจ่ายลมอัดภายใน

2.9 ชุดตัวควบคุม Programmable Logic Control : PLC

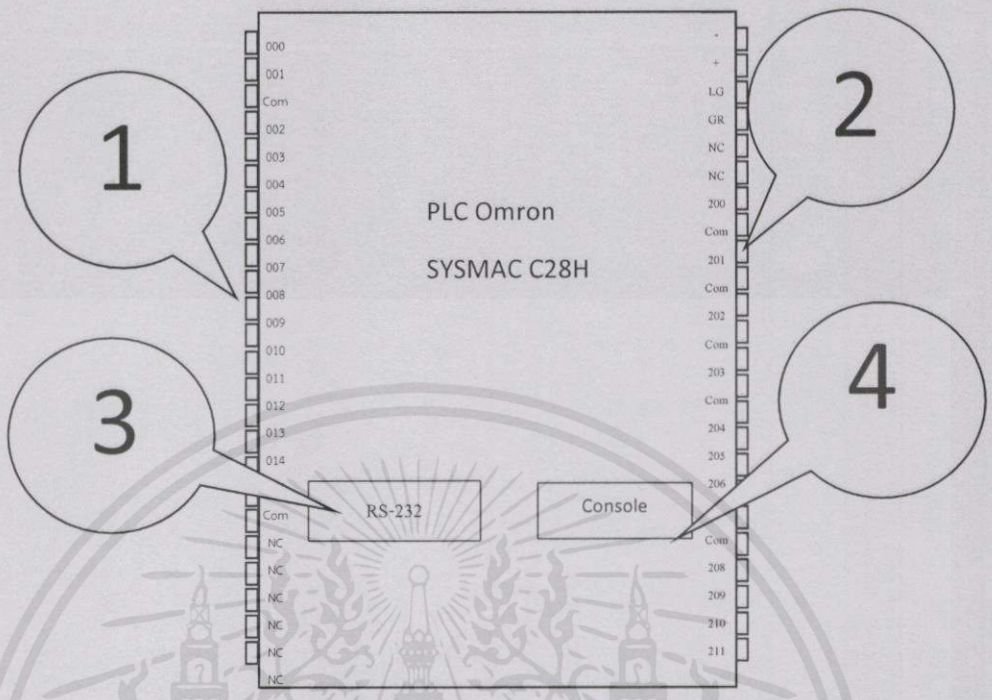
อุปกรณ์ทางด้าน Hardware ที่ใช้ในการควบคุมกระบอกสูบ จะเป็นการนำเอา ตัวควบคุม (PLC) ของ Omron รุ่น Sysmac C28H โดยจะนำสัญญาณจากเซนเซอร์ที่ตรวจจับทางด้านหัวกระบอกสูบมาเชื่อมต่อทางด้านสัญญาณขาเข้า และจะทำงานตามโปรแกรมส่งสัญญาณขาออกไปควบคุมวาล์วควบคุมให้ทำงาน และเชื่อมต่อกับคอมพิวเตอร์โดยการใช้ RS-232 แบบ 9 to 9 แปลงสัญญาณผ่านตัวแปลงสัญญาณจาก RS-232 แบบ 9 พิน เป็น USB



รูปที่ 2.17 ตัวควบคุม PLC Omron Sysmac C28H

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.9.1 ส่วนประกอบที่สำคัญของชุดตัวควบคุม PLC Omron Syamac C28H



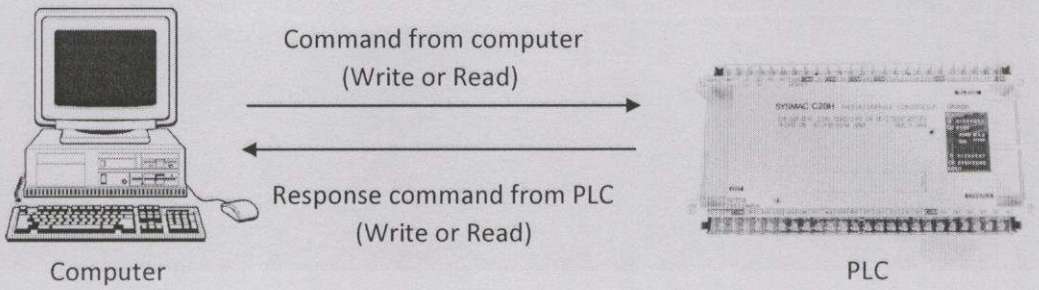
รูปที่ 2.18 ตัวควบคุม PLC Omron Sysmac C28H

จากรูป อธิบายรายละเอียดส่วนประกอบ PLC ตามลำดับหมายเลข

- หมายเลข 1 คือ ช่องเสียบทางด้านสัญญาณขาเข้า มีทั้งหมด 15 สัญญาณขาเข้า (Input) คือ 0 -15 และ 2 สัญญาณร่วม (Common)
- หมายเลข 2 คือ ช่องเสียบทางด้านสัญญาณขาออก มีทั้งหมด 12 สัญญาณขาออก (Output) คือ 200 – 211 , 6 สัญญาณร่วม (Common) และช่องเสียบไฟ 24 VDC รวมไปถึงช่องเสียบกราวด์ด้วย
- หมายเลข 3 คือ ช่องเสียบสายเชื่อมต่อ RS-232 เพื่อเชื่อมต่อกับคอมพิวเตอร์
- หมายเลข 4 คือ ช่องเสียบของเครื่องคอนโซล (Console) แบบปรับด้วยมือ เมื่อกรณีที่ตัวควบคุมมีปัญหา และต้องการแก้ปัญหาแบบทันท่วงที

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.9.2 โปรโตคอลในการติดต่อสื่อสาร



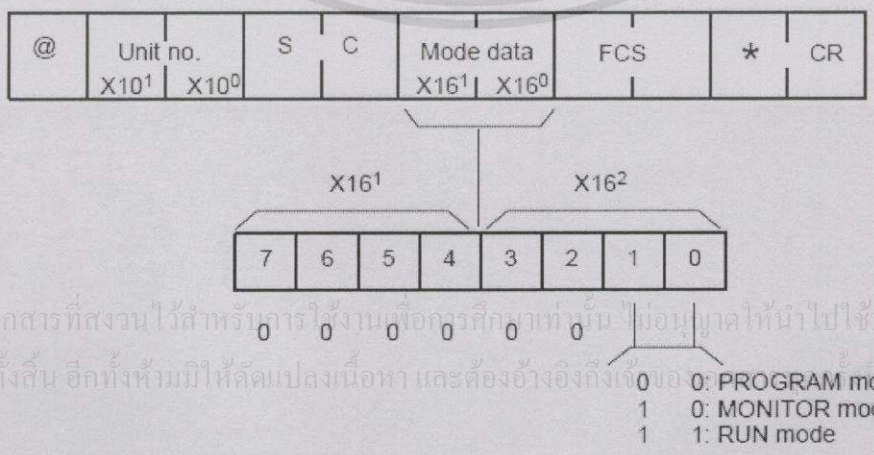
รูปที่ 2.19 การตอบโต้กันระหว่างคอมพิวเตอร์กับตัวควบคุม PLC

ในการสื่อสารนั้น จะทำการเชื่อมต่อคอมพิวเตอร์กับตัวควบคุม โดยจะเป็นการสื่อสารกันในระบบ Hostlink ซึ่งตัวคอมพิวเตอร์จะมีชุดคำสั่งส่งผ่านไปที่ตัวควบคุมเพื่อทำการเขียนค่าให้กับตัวควบคุม หรือ การไรท์ (Write) เมื่อตัวควบคุมได้คำสั่งเขียนค่าจากคอมพิวเตอร์แล้ว ก็จะมีการตอบสนองกลับมาว่ามีการได้รับการเขียนค่าแล้ว โดยเป็นชุดคำสั่งเช่นเดียวกัน และจะเหมือนกันในการอ่านค่าจากตัวควบคุม หรือคำสั่ง รีด (Read) หากต้องการทราบที่อยู่ของเลขลำดับของบิตที่สนใจในระบบHostlink ก็ใช้คำสั่งอ่านค่าในชาแนลนั้นๆ เพื่อสังเกตว่าเมื่ออุปกรณ์มีการทำงานแล้ว ค่าในตัวควบคุมเป็นเช่นไร

2.9.2.1 คำสั่งไรท์ (Write command)

คำสั่งไรท์ (Write) จะเป็นคำสั่งที่คอมพิวเตอร์ใช้สั่งการเข้าไปที่ตัวควบคุม โดยจะเป็นการเปลี่ยนโหมดการดำเนินงานของเครื่องคอมพิวเตอร์ ตามข้อมูลของอินพุต ให้อยู่ในรูปข้อมูลแบบเลขฐาน 16 โดยจะมีรูปแบบคำสั่ง ดังต่อไปนี้

Command format (Write)



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของลิขสิทธิ์ทุกครั้งที่มีการนำไปใช้

Response format (Write)

@	Unit no. $X10^1$ $X10^0$	S	C	Response code $X16^1$ $X16^0$	FCS	*	CR

ตัวอย่างการใช้งานคำสั่งนี้คือ หากเรากำหนดแอดเดรสของบิตในตัวควบคุมเป็นบิตที่ 13.00 และเราต้องการที่จะสั่งให้มันทำงานโดยคอมพิวเตอร์ เราจะต้องใช้คำสั่ง @00WR00130001 เพื่อเข้าไปสั่งการ หากตัวควบคุมได้รับคำสั่งแล้ว ก็จะมีการตอบสนองกลับมาที่คอมพิวเตอร์เพื่อบอกว่าตัวควบคุมได้รับคำสั่งนี้แล้ว เช่น @00WR0045* เป็นต้น

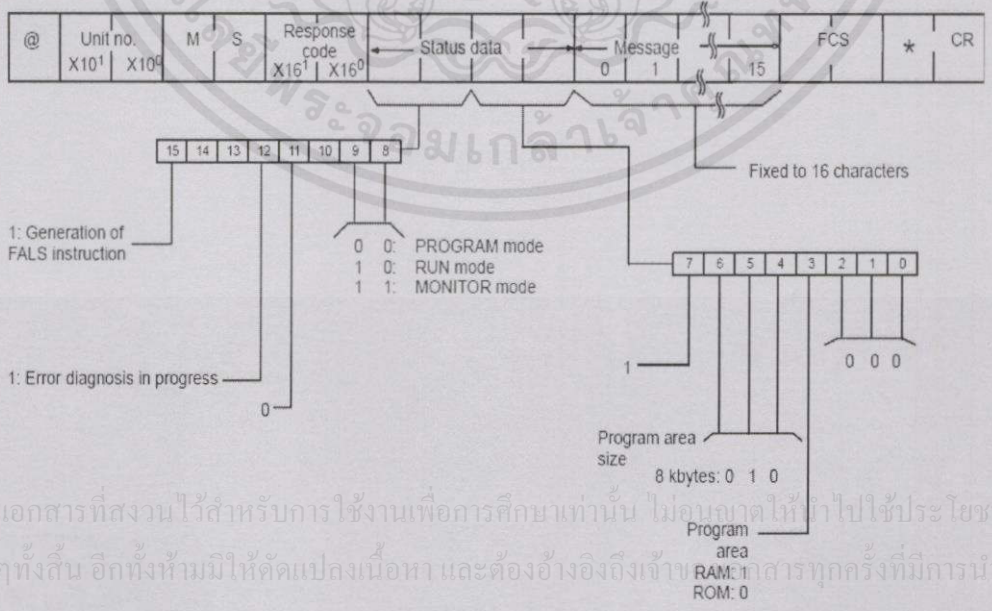
2.9.2.2 คำสั่งรีด (Read command)

คำสั่งรีดนี้จะเป็นการที่คอมพิวเตอร์ถามไปที่ตัวควบคุมว่าบิตของอุปกรณ์ที่กำลังทำงานนั้น โดยคำสั่งนี้จะเป็นการทำให้เครื่องคอมพิวเตอร์อ่านค่าสถานะของตัวควบคุม ข้อความจะถูกป้อนเข้าไปเมื่อ MSG(46) กำลังดำเนินการอยู่

Command format (Read)

@	Unit no. $X10^1$ $X10^0$	M	S	FCS	*	CR

Response format (Read)

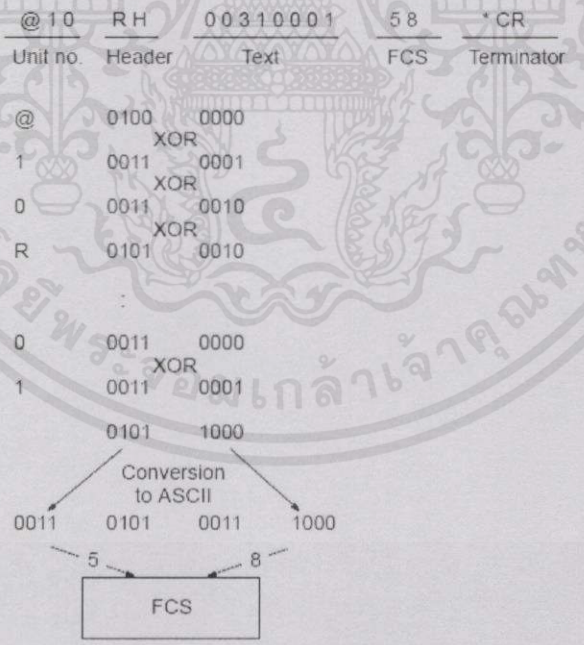


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวอย่างการใช้งานคำสั่งนี้คือ หากเราต้องการทราบสถานะของตัวควบคุมในขณะนั้น เราจะใช้คำสั่งรีด เพื่อเป็นการอ่านค่าในชาแนลที่เราสนใจ โดยหากในชาแนลที่เราสนใจ คือ ชาแนล 13 ในตัวควบคุม เราจะต้องทำการป้อนคำสั่ง @00RR00130000 เพื่อทำการถามว่าตอนนี้ ชาแนลนี้ในตัวควบคุมมีสถานะเป็นเช่นไร โดยจะมีการตอบสนองกลับมาจากตัวควบคุม เช่น @00RR00155445* เป็นต้น โดยเราจะนำเลขตรงส่วนของ Message ไปใช้งานต่อไปในโปรแกรม Visual Basic

2.9.3 Frame Check Sequence : FCS Calculation

เป็นกระบวนการตรวจสอบความถูกต้องของข้อมูลชนิดหนึ่ง เป็นส่วนที่ใช้ในการตรวจสอบความผิดพลาดในการส่ง-รับข้อมูลที่อาจเกิดขึ้น ซึ่งอาจจะตรวจสอบเฉพาะพาริตี้บิต หรือทั้งบล็อกข้อมูล โดยในส่วนของตัวควบคุมนี้จะเป็นการตรวจสอบข้อมูลในการส่งถึงกันระหว่างคอมพิวเตอร์กับตัวควบคุม PLC ซึ่งระบบการตรวจสอบแบบ FCS นั้นมีวิธีการตรวจสอบ คือ การทำข้อมูลขนาด 8 บิต แปลงเป็นรหัสแอสกี้ 2 หลัก โดยข้อมูล 8 บิตนั้น จะเป็นผลของการนำข้อมูลมาทำการ EXCLUSIVE OR กัน ตั้งแต่ข้อมูลตัวแรกในเฟรม จนถึงข้อมูลสุดท้ายในเฟรม ซึ่งวิธีนี้เป็นการตรวจสอบข้อมูลที่รับ หรือ ส่ง ว่ามีความถูกต้องหรือไม่นั่นเอง ดังรูปต่อไปนี้



รูปที่ 2.20 หลักการตรวจสอบแบบ FCS Calculation

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3
การออกแบบและการทำงาน

3.1 กล่าวนำ

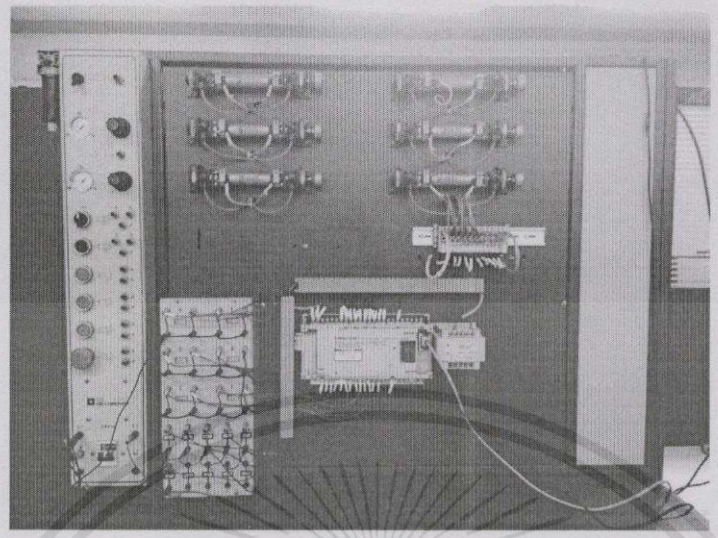
การออกแบบการทำงานของเครื่องควบคุมการทำงานของกระบอกสูบแบบเรียงลำดับที่ใช้ในโรงงานอุตสาหกรรม การทำงานของโปรแกรมควบคุมจะให้ผู้ใช้ทำการกำหนดลักษณะการทำงานของกระบอกสูบซึ่งจะมีลักษณะการทำงานทั้งหมด 5 รูปแบบ เมื่อสั่งเริ่มการทำงานจากโปรแกรมควบคุมคอมพิวเตอร์จะทำการส่งข้อมูลผ่านพอร์ตอนุกรมและผ่านสาย RS-232 ไปยังตัวควบคุม Programmable Logic Control : PLC ซึ่งจะทำหน้าที่ในการขับเอาท์พุตรีเลย์ 24 VDC ให้ทำการเปิด-ปิด (ON-OFF) ซึ่งในนี้จะเป็นการขับวาล์วควบคุมด้วยไฟฟ้า ให้ทำการควบคุมแรงดันลมอัดที่จ่ายให้กระบอกสูบต่อไป เป็นผลทำให้กระบอกสูบเคลื่อนที่ตามรูปแบบที่ผู้ใช้กำหนดไว้ ซึ่งการออกแบบแบ่งเป็นทั้งด้าน Software และ Hardware เป็นดังนี้

3.2 การออกแบบระบบควบคุม

ในการควบคุมนั้นจะประกอบไปด้วยสัญญาณทางด้านขาเข้า และสัญญาณทางด้านขาออก โดยในส่วนของสัญญาณขาเข้านั้น จะออกแบบให้เป็นสัญญาณจากตัวเซนเซอร์ที่ตรวจจับบนกระบอกสูบ เชื่อมต่อเข้าสู่ตัวควบคุม หากเซนเซอร์มีการตรวจจับได้ก็จะส่งค่า “1” เข้าตัวควบคุม หากไม่สามารถตรวจจับได้ก็จะส่งค่าเป็น “0” และทางสัญญาณขาออกนั้นน จะส่งสัญญาณไปควบคุมวาล์วควบคุมด้วยไฟฟ้าให้ทำงานตามโปรแกรมในตัวควบคุม เพื่อทำการเปิด-ปิดวาล์วเพื่อควบคุมระบบนิวเมติกต่อไป

ในการควบคุมแบบเรียงลำดับนั้นเราจะออกแบบให้แลดเดอร์ในตัวควบคุม นั้นรอรับค่า โดยเปรียบเสมือนสวิตช์ปุ่มกด หากกดก็จะมีสัญญาณ “1” เข้ามาจะเป็นคำสั่งให้เริ่มกระบวนการ เมื่เริ่มกระบวนการแล้วตัวควบคุมจะทำตามโปรแกรมเป็นลำดับขั้นตอน โดยตัวอย่างอย่างเช่น หากเซนเซอร์สามารถตรวจจับได้ก็จะมีสัญญาณขาเข้าเป็น “1” ให้สั่งให้วาล์วควบคุมด้วยไฟฟ้าแบบ 5/2 ทำงาน เมื่อวาล์วทำงานจะทำให้ลมอัดจะไหลเข้าสู่ด้านหัวของกระบอกสูบ ทำให้กระบอกสูบเลื่อนออก และถ้าเซนเซอร์ไม่สามารถตรวจจับได้ ก็จะส่งสัญญาณเป็น “0” ให้กับตัวควบคุมแทน จึงทำให้วาล์วปิดลมอัดก็จะสลับการไหล ไหลเข้าทางท้ายกระบอกสูบแทน ทำให้กระบอกสูบเลื่อนปิด โดยจะนำแนวคิดนี้ไปออกแบบโปรแกรมให้ตัวควบคุม (ภาคผนวก ข.)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.1 แสดงการเชื่อมต่อกันของอุปกรณ์ภายในระบบ

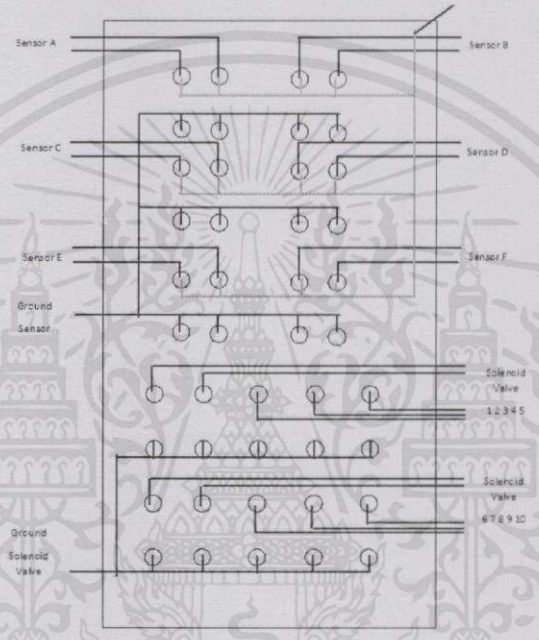
ตารางที่ 3.1 แสดงการเชื่อมต่อสัญญาณขาเข้าและสัญญาณขาออกกับตัวควบคุม (I/O Assignment)

Input		Output	
Sensor A1	002	Solenoid Valve 1	200
Sensor A2	003	Solenoid Valve 2	201
Sensor B1	004	Solenoid Valve 3	202
Sensor B2	005	Solenoid Valve 4	203
Sensor C1	006	Solenoid Valve 5	204
Sensor C2	007	Solenoid Valve 6	205
Sensor D1	008		
Sensor D2	009		
Sensor E1	010		
Sensor E2	011		
Sensor F1	012		
Sensor F2	013		

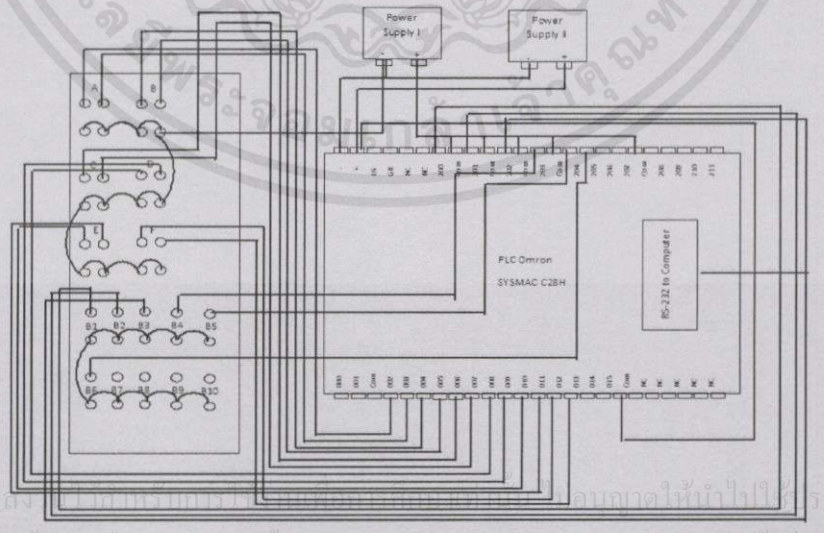
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.1 การออกแบบกล่องสัญญาณขาเข้าและขาออก

กล่องสัญญาณขาเข้าและขาออกนั้นจะแบ่งออกเป็นสองส่วนด้วยกัน ประกอบด้วยส่วนสัญญาณขาเข้าที่มีการเชื่อมต่ออยู่กับเซนเซอร์ โดยมีการต่อเพิ่มความต้านขนาด 4.7 กิโลโอห์ม เพื่อป้องกันการรับกระแสเกินของตัวเซนเซอร์ และอีกส่วนหนึ่งคือส่วนสัญญาณขาออก จะเชื่อมต่อกับวาล์วควบคุมด้วยไฟฟ้าที่มีหน้าที่ทำการควบคุมระบบนิวเมติกให้เป็นไปตามโปรแกรมของตัวควบคุม



รูปที่ 3.2 กล่องสัญญาขาเข้าและสัญญาณขาออก

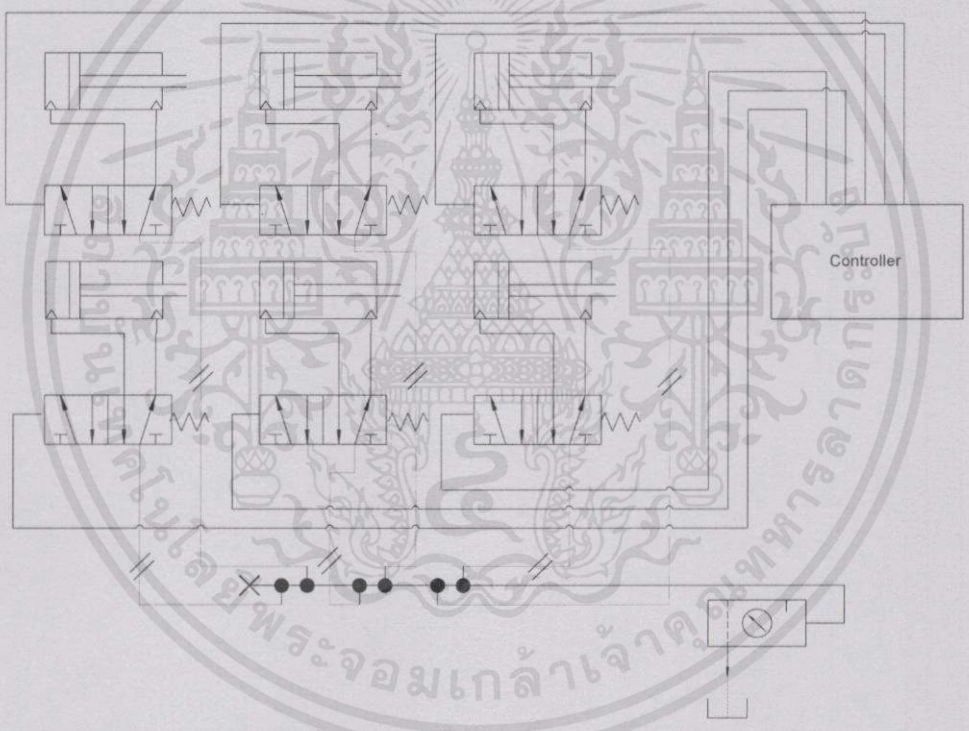


รูปที่ 3.3 การเชื่อมต่อกันระหว่างกล่องสัญญาณขาเข้าและสัญญาณขาออกกับตัวควบคุม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น ไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารนี้ทุกครั้งที่มีการนำไปใช้

3.2.2 การออกแบบระบบนิวเมติก

ในส่วนของการออกแบบระบบนิวเมติกที่นำมาใช้ในปริญญานิพนธ์นั้น จะออกแบบโดยการใช้วาล์วควบคุมทางไฟฟ้าทำหน้าที่เปิด/ปิด ระบบนิวเมติกเข้าสู่กระบอกสูบ หากวาล์วคุมทำงานก็จะมีลมเข้าสู่กระบอกสูบ ทำให้กระบอกสูบเลื่อนออก หากวาล์วควบคุมไม่ทำงานกระบอกสูบก็จะอยู่ที่เดิม เพราะกระบอกสูบที่เราใช้เป็นกระบอกสูบแบบสองทาง ซึ่งจะมีการเชื่อมต่อกับระบบนิวเมติกทั้งหัวและท้ายกระบอกสูบ โดยหากวาล์วควบคุมไม่ทำงาน กระบอกสูบจะอยู่ในสภาวะปิด คือจะมีลมทางด้านท้ายกระบอกสูบอัดอยู่ และในทางกลับกันหากวาล์วควบคุมทำงาน กระบอกสูบจะเลื่อนออก เพราะมีการเปิดของวาล์วควบคุมทางไฟฟ้า ทำให้ลมอัดจะสลับทิศทางไปอยู่ที่ทางด้านหัวกระบอกสูบ



รูปที่ 3.4 การเชื่อมต่อกันในระบบนิวเมติก

3.3 การออกแบบโปรแกรมควบคุม

ในส่วนของการออกแบบโปรแกรมควบคุมนั้น จะเป็นการกำหนดรูปแบบการทำงานของกระบอกสูบผ่านทางคอมพิวเตอร์โดยจะส่งข้อมูลผ่านพอร์ตอนุกรม RS-232 ไปยังตัวควบคุมซึ่งมีหน้าที่ในการแปลงสัญญาณดิจิตอลไปเป็นสัญญาณอะนาลอก เพื่อใช้ควบคุมวาล์วควบคุมโดยใช้ไฟฟ้าเป็นผลให้กระบอกสูบทำงานตามรูปแบบที่กำหนด โดยการออกแบบโปรแกรมควบคุมจะใช้โปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

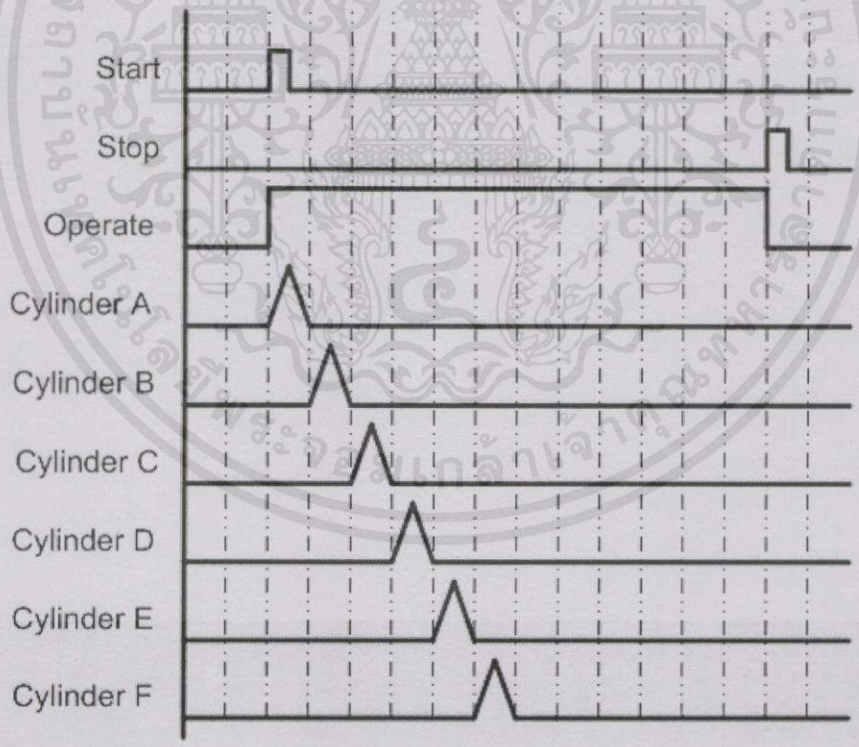
Visual Basic บนคอมพิวเตอร์ที่จะทำหน้าที่ส่ง Protocol ไปยังตัวควบคุมและรับค่าสถานะการทำงานของกระบอกสูบจากเซนเซอร์แบบแม่เหล็กไฟฟ้า เพื่อนำมาแสดงผลสถานะของกระบอกสูบบนคอมพิวเตอร์ โปรแกรมควบคุมจะประกอบไปด้วยส่วนต่างๆดังนี้

3.3.1 โปรแกรมการควบคุมและการแสดงสถานะของการทำงาน

ในส่วนของการสั่งการควบคุมและการแสดงผลสถานะของการทำงาน จะกระทำได้โดยทำการ Click ที่ปุ่ม Control จากนั้นจะแสดงหน้าต่าง Control ขึ้นมา ซึ่งจะแบ่งส่วนต่างๆ ได้ 4 ส่วนหลักๆ คือ ส่วนการสั่งการ, ส่วนการแสดงผลสถานะของกระบอกสูบและเซนเซอร์, ส่วนของ Timing Diagrams และไฟเตือนสภาพการใช้งานของกระบอกสูบซึ่งจะมีรายละเอียดดังนี้

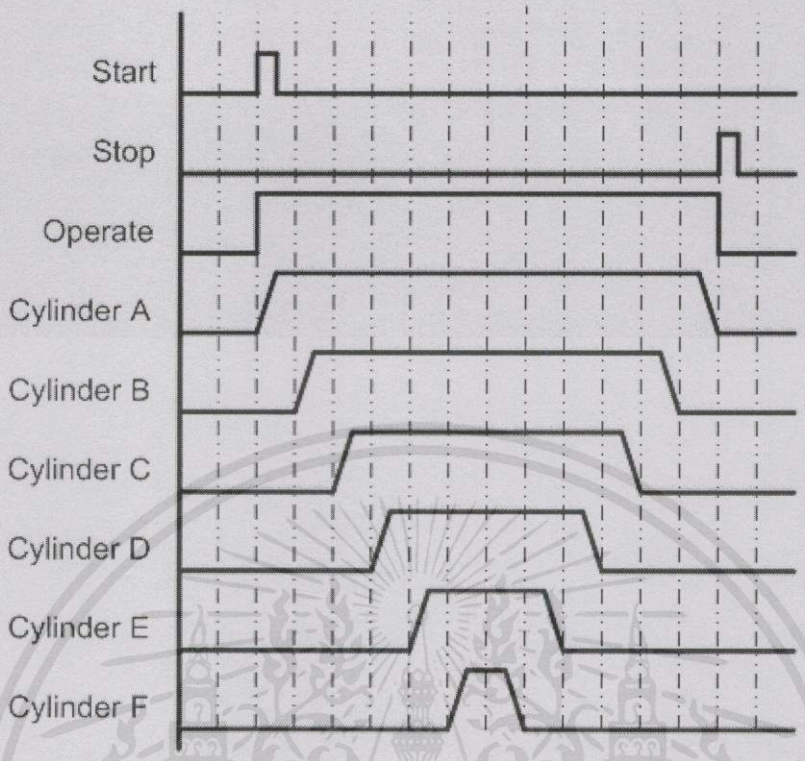
3.3.1.1 ส่วนการสั่งการ

ในส่วนการสั่งการควบคุม จะให้ผู้ใช้เลือกรูปแบบการทำงานของกระบอกสูบว่าผู้ใช้ต้องการให้กระบอกสูบทำงานเรียงลำดับในรูปแบบใด ซึ่งในการออกแบบรูปแบบการทำงานของกระบอกสูบมีทั้งหมด 5 แบบที่แตกต่างกันไปคือ

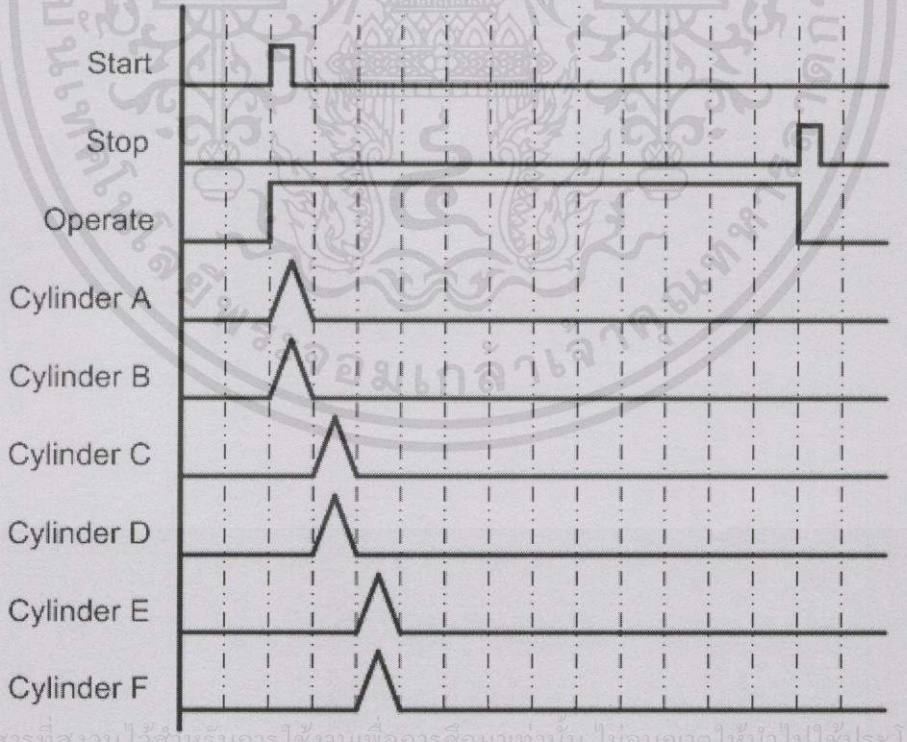


รูปที่ 3.5 รูปแบบการทำงานที่ 1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

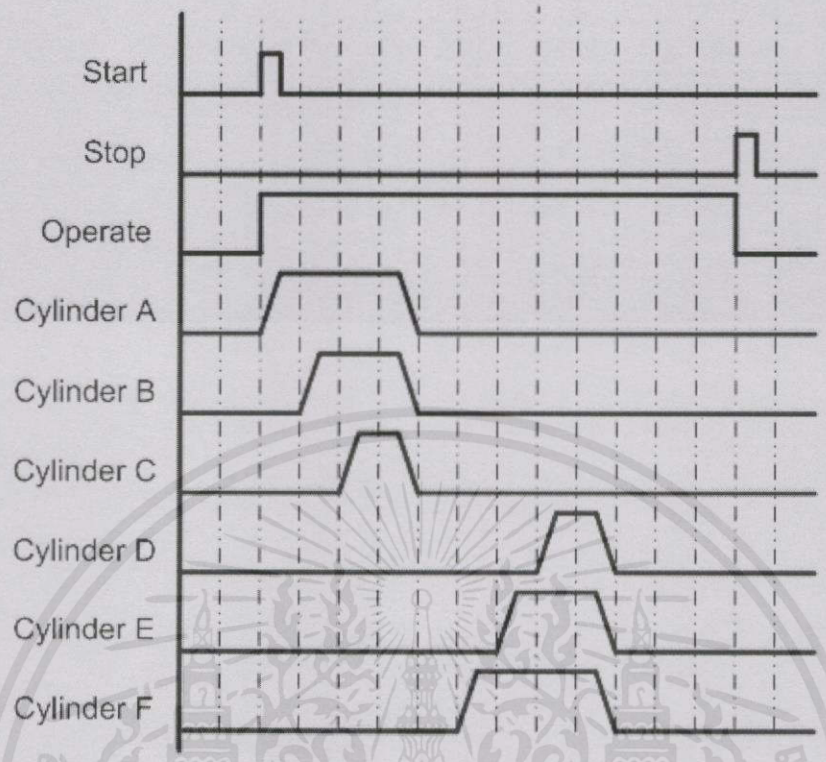


รูปที่ 3.6 รูปแบบการทำงานที่ 2

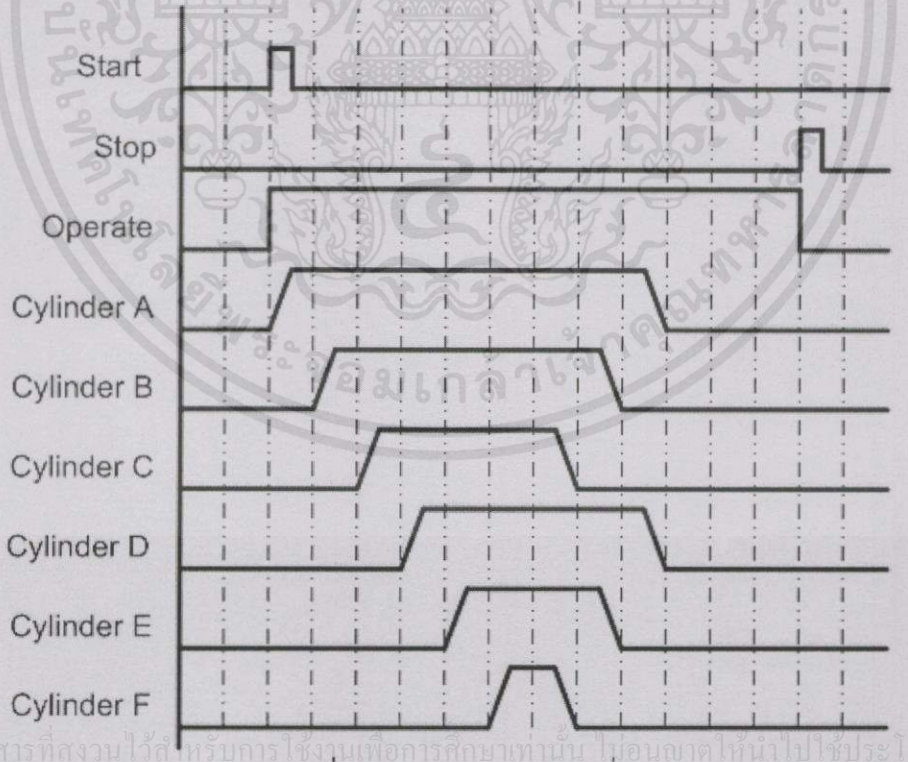


รูปที่ 3.7 รูปแบบการทำงานที่ 3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



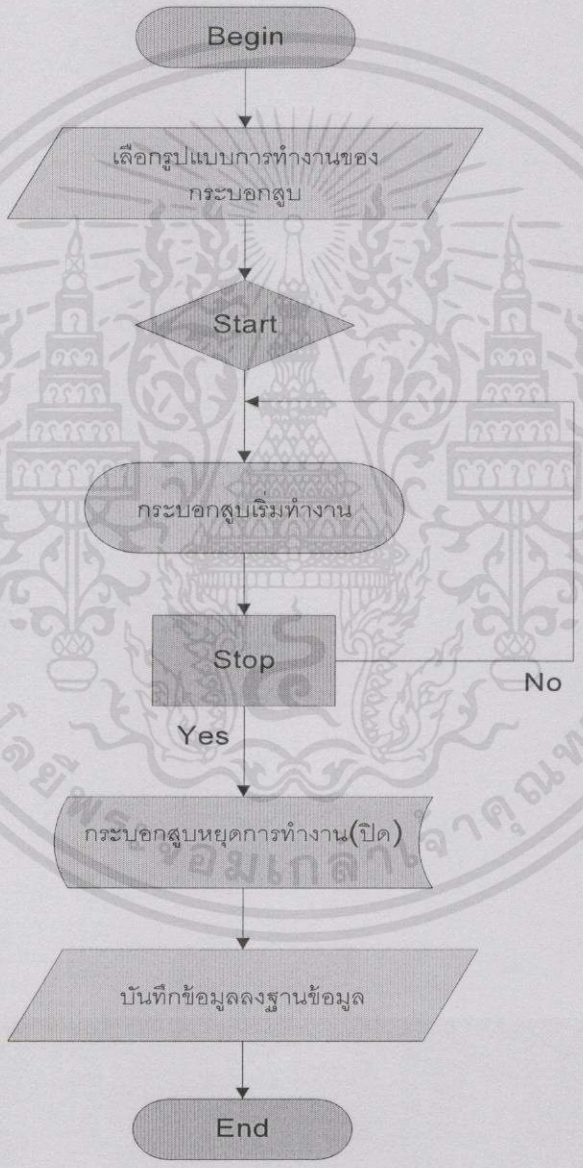
รูปที่ 3.8 รูปแบบการทำงานที่ 4



รูปที่ 3.9 รูปแบบการทำงานที่ 5

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อผู้ใช้เลือกรูปแบบการทำงานของกระบอกสูบเรียบร้อยร้อยแล้ว เมื่อคลิกปุ่ม Start จะเป็นการส่งโพรโตคอลไปยังตัวควบคุม ซึ่งตัวควบคุมก็จะตรวจสอบโพรโตคอลที่ส่งมาและทำการแปรโพรโตคอลที่ส่งมาว่าต้องการให้ควบคุมที่ Channel ใด และนำค่าของข้อมูลดิจิตอลที่ส่งมาด้วยนั้นทำการแปลงให้เป็นสัญญาณอะนาลอก (4-20 mAdc) ไปควบคุมวาล์วควบคุมทิศทางแบบใช้ไฟฟ้าเพื่อควบคุมการทำงานกระบอกสูบอีกทีหนึ่ง หากผู้ใช้ต้องการที่จะเปลี่ยนรูปแบบการทำงานของการบอกสูบ จะต้องคลิกปุ่ม Stop จึงจะเปลี่ยนรูปแบบการทำงานได้

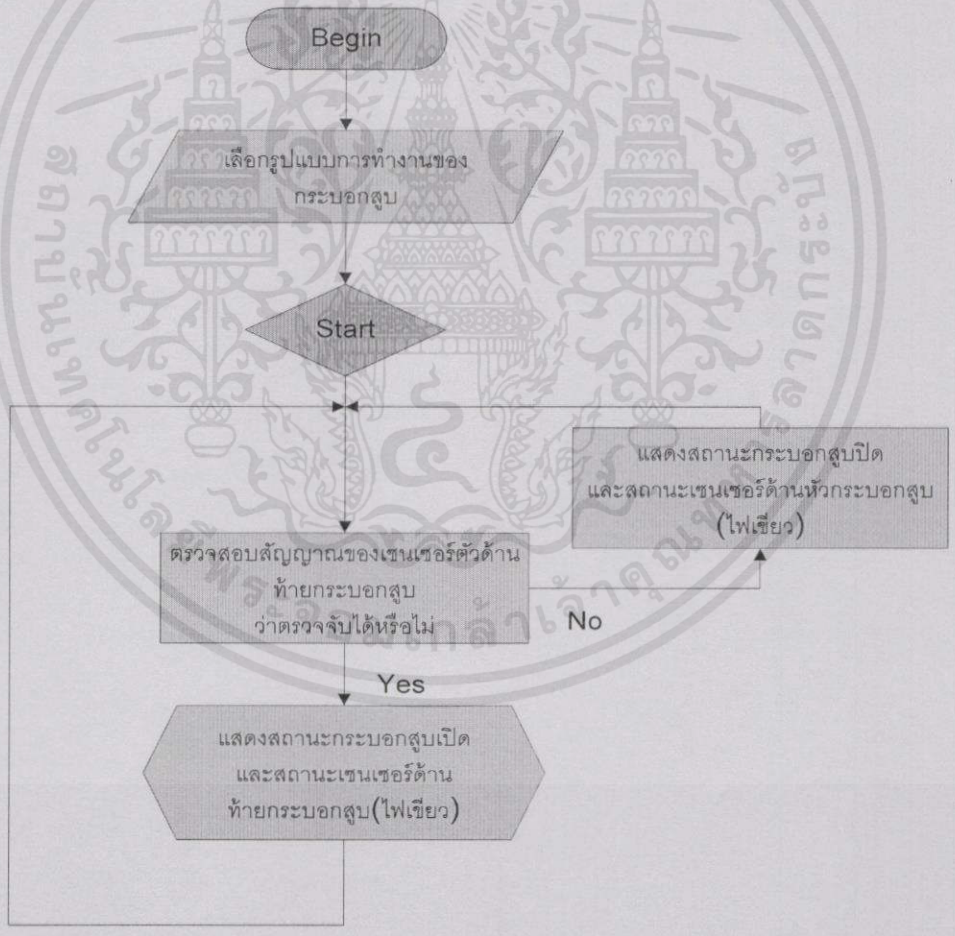


รูปที่ 3.10 Flow Chart ขั้นตอนการทำงานส่วนสั่งการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3.1.2 ส่วนการแสดงผลสถานะของกระบอกสูบและเซนเซอร์

ในส่วนของการแสดงผลนี้ เมื่อผู้ใช้ทำการเลือกรูปแบบการทำงานของลูกสูบแล้วคลิกที่ปุ่ม Start กระบอกสูบจะเริ่มทำงานและจะมีการรับค่าสัญญาณจากตัวควบคุม โดยจะทำการส่งโพรโตคอลเพื่อไปอ่านค่าของสถานะที่ Channel 0 บิตที่ 2 ถึง 13 ซึ่งถูกต่อเข้ากับเซนเซอร์แบบแม่เหล็กไฟฟ้าว่ามีสัญญาณหรือไม่ โดยเซนเซอร์จะส่งสัญญาณแรงดันมาที่ Channel 0 ตัวควบคุม ถ้าหากเซนเซอร์ตัวด้านท้ายกระบอก (บิตที่ 3, 5, 7, 9, 11, 13) มีแรงดันเข้ามา จะเป็นผลทำให้แสดงสถานะของกระบอกสูบอยู่ในตำแหน่งเปิด และเซนเซอร์ด้านท้ายกระบอกสูบตรวจจับได้ (ไฟสีเขียว) ถ้าหากเซนเซอร์ตัวด้านท้ายกระบอกสูบตรวจจับสัญญาณไม่ได้ก็จะแสดงสถานะกระบอกสูบอยู่ในตำแหน่งปิดและเซนเซอร์ตัวที่ติดตั้งด้านหัวกระบอกสูบตรวจจับได้(ไฟเขียว) ซึ่งจะตรวจสอบสัญญาณที่ Channel 0 ไปเรื่อยๆจนกว่าจะคลิกปุ่ม Stop

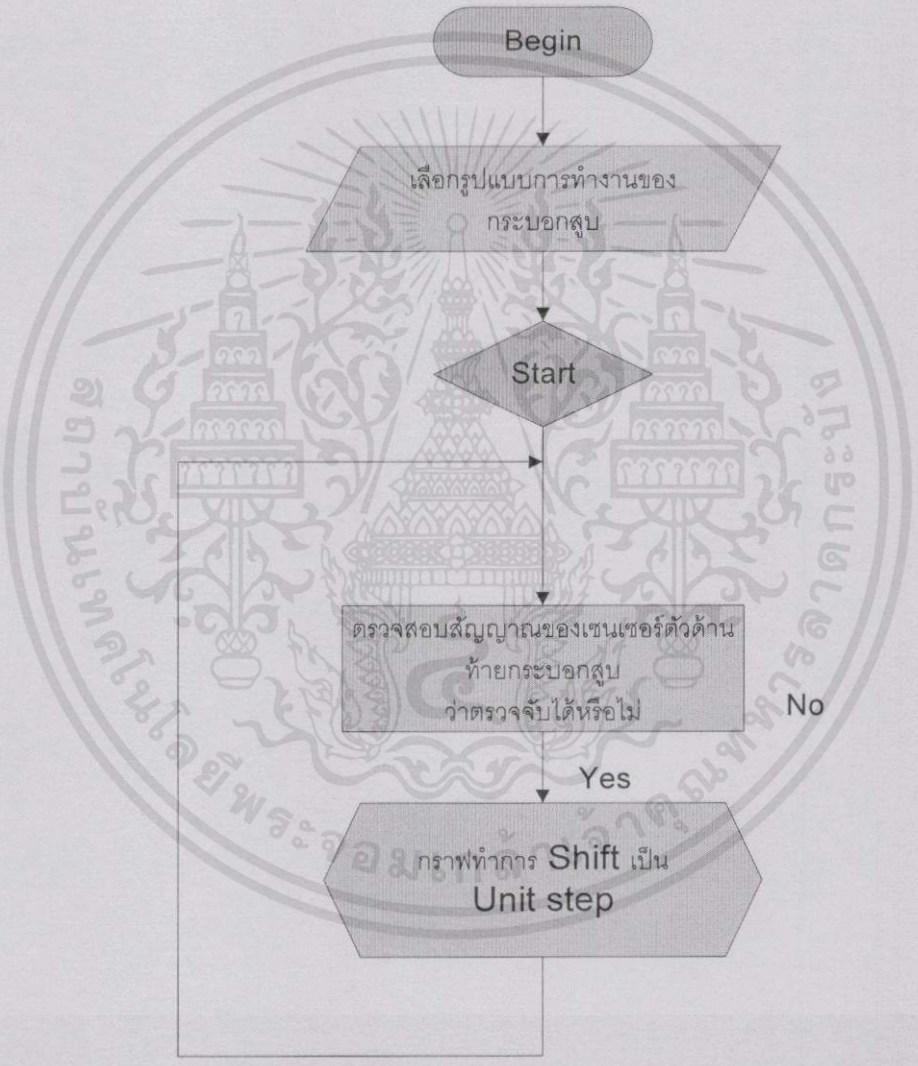


รูปที่ 3.11 Flow Chart ขั้นตอนการทำงานส่วนแสดงผล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3.1.3 ส่วนของกราฟ Timing Diagrams

เมื่อผู้ใช้ทำการเลือกรูปแบบแล้วคลิกปุ่ม Start เริ่มการทำงาน จะมีการส่งโพรโตคอลไปอ่านค่าจากตัวควบคุมใน Channel 0 ซึ่งต่อเข้ากับเซนเซอร์แบบแม่เหล็กไฟฟ้า ถ้าหากเซนเซอร์ตัวที่ติดตั้งด้านท้ายกระบอกสูบไม่มีสัญญาณ กราฟจะแสดงเป็นเส้นตรงทั้ง 6 เส้น (กระบอกสูบทั้ง 6 ตัว) หากมีสัญญาณจะส่งผลทำให้เส้นกราฟของกระบอกสูบนั้นๆ Shift กราฟในลักษณะ Unit step ซึ่งจะตรวจสอบสัญญาณที่ Channel 0 ไปเรื่อยๆจนกว่าจะคลิกปุ่ม Stop

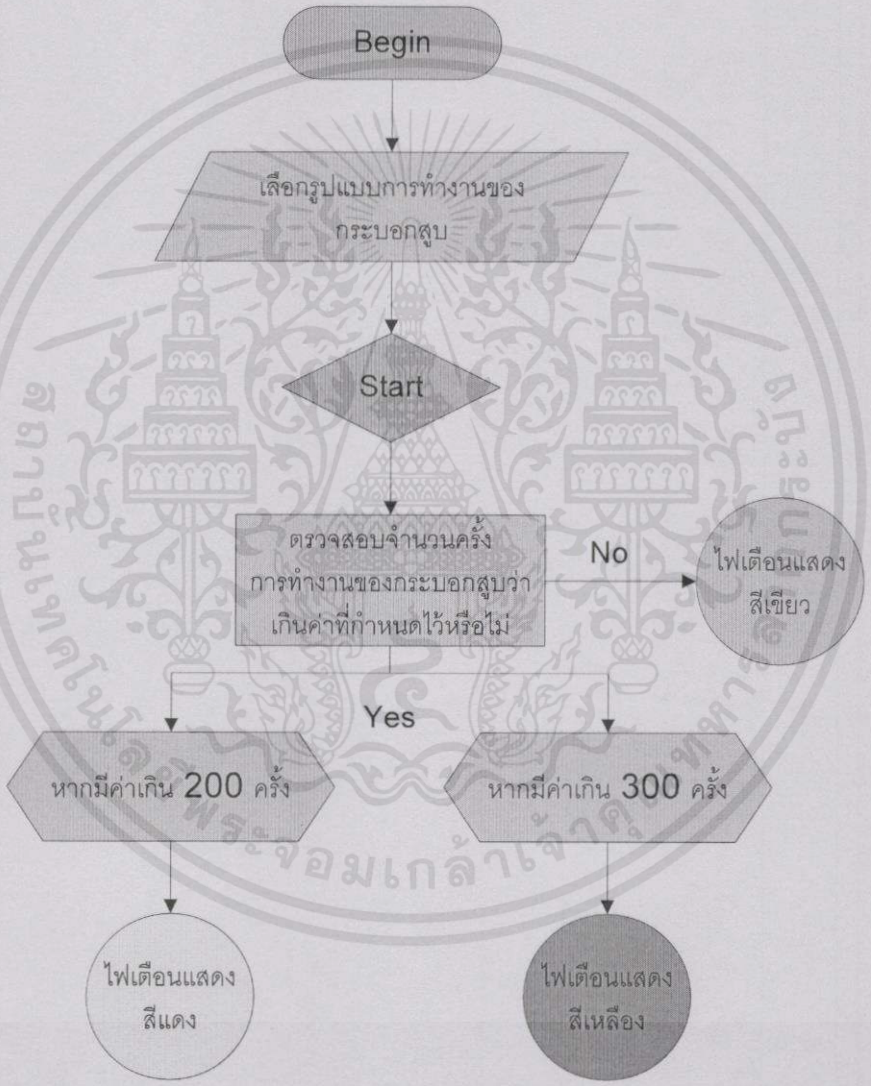


รูปที่ 3.12 Flow Chart ขั้นตอนการทำงานส่วนกราฟ Timing Diagrams

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3.1.4 ส่วนของไฟเตือนสภาพการใช้งานของกระบอกสูบ

ในหน้าต่างของโปรแกรมควบคุม จะมีไฟแสดงเตือนสภาพของกระบอกสูบว่าผู้ใช้ควรที่จะต้องซ่อมบำรุงหรือเปลี่ยนกระบอกสูบใหม่หรือไม่ โดยจะอ่านค่าจากฐานข้อมูลว่ากระบอกสูบแต่ละตัวทำงานไปทั้งหมดกี่ครั้ง แล้วนำมาเข้าเงื่อนไขที่กำหนดไว้เช่น ถ้าหากกำหนดจำนวนครั้งไว้สูงสุดที่ 300 ครั้ง หากกระบอกสูบ A ทำงานเกิน 300 ครั้ง ไฟก็จะแสดงเป็นสีแดงหมายถึงควรที่จะทำการซ่อมบำรุงหรือเปลี่ยนกระบอกสูบใหม่



รูปที่ 3.13 Flow Chart ขั้นตอนการทำงานส่วนของไฟเตือนสภาพการใช้งานกระบอกสูบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3.2 โปรแกรมการควบคุมแบบกำหนดเอง

โปรแกรมบการควบคุมด้วยมือเป็น โปรแกรมที่ผู้ควบคุมหน้าจอคอมพิวเตอร์ ต้องการจะทำการควบคุมการทำงานของกระบอกสูบเอง โดยเริ่มจากการกดปุ่ม Auto Manual ที่หน้าโปรแกรมหลัก ก็จะปรากฏหน้าโปรแกรมควบคุมด้วยมือขึ้นมา ผู้ใช้สามารถเลือกได้ว่าต้องการควบคุมแบบอัตโนมัติหรือควบคุมด้วยมือ หากควบคุมอัตโนมัติการใช้งานจะเหมือนกับหน้าโปรแกรมการควบคุม ถ้าหากควบคุมด้วยมือสามารถกดที่ปุ่มของแต่ละกระบอกสูบได้ทันที ซึ่งจะมีผลทำให้กระบอกสูบทำงานตามที่ผู้ใช้กด

3.3.3 โปรแกรมการแสดงค่าจำนวนครั้งการทำงานของกระบอกสูบ

ส่วนของโปรแกรมย่อยนี้เป็นการแสดงจำนวนการทำงานของกระบอกสูบ ซึ่งผู้ใช้จะต้องคลิกที่ปุ่ม Cylinder จึงจะปรากฏหน้าต่างที่แสดงเป็นตารางจำนวนครั้งการทำงานของกระบอกสูบ โดยจะรับค่าสัญญาณจากตัวควบคุมว่ามีสัญญาณหรือไม่ ซึ่งจะทำการส่งโพรโตคอลไปยังตัวควบคุมเพื่อไปอ่านค่าสถานะที่ Channel 0 ซึ่งถูกต่อเข้ากับเซนเซอร์แบบแม่เหล็กไฟฟ้าว่ามีสัญญาณหรือไม่ โดยเซนเซอร์จะส่งสัญญาณแรงดันไฟฟ้า (4-20 mADC)มาที่ Channel 0 ของตัวควบคุม ถ้าหากเซนเซอร์ตัวด้านท้ายกระบอก (บิตที่ 3, 5, 7, 9, 11, 13) มีแรงดันไฟฟ้าเข้ามาก็จะนับจำนวนครั้งในการทำงานของกระบอกสูบบวกเพิ่มจากค่าเดิม เมื่อผู้ใช้หยุดการทำงานจึงจะเก็บค่าของจำนวนครั้งการทำงานของกระบอกสูบลงในฐานข้อมูล

3.3.4 โปรแกรมการแก้ไขข้อมูลเมื่อมีการซ่อมบำรุงกระบอกสูบ

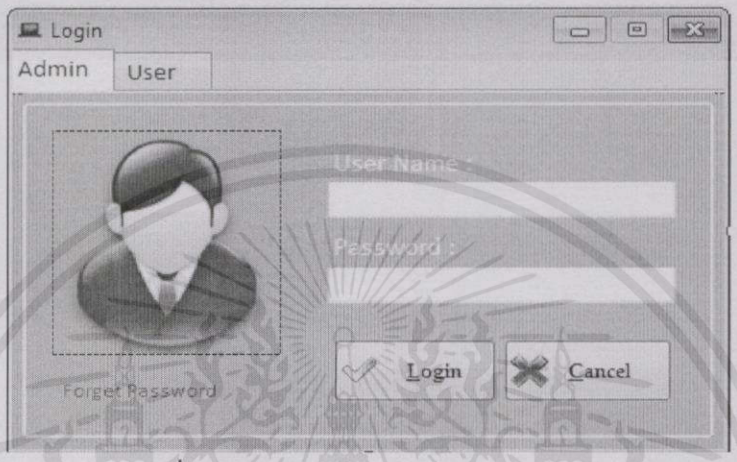
ในส่วนนี้จะเป็นการแก้ไขข้อมูลจำนวนครั้งการทำงานของกระบอกสูบ คือโดยปกติแล้วการที่กระบอกสูบทำงานมากๆ อาจมีผลทำให้กระบอกสูบเกิดการสึกหรอหรือเสียหายได้ เพราะฉะนั้นเมื่อทำการซ่อมบำรุงหรือทำการเปลี่ยนกระบอกสูบ ข้อมูลจำนวนครั้งการทำงานในฐานข้อมูลจะต้องเริ่มค่าใหม่ (0) ซึ่งสามารถทำโดยการคลิกปุ่ม Maintenance จะปรากฏหน้าต่างขึ้นมา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4 รายละเอียดของโปรแกรม

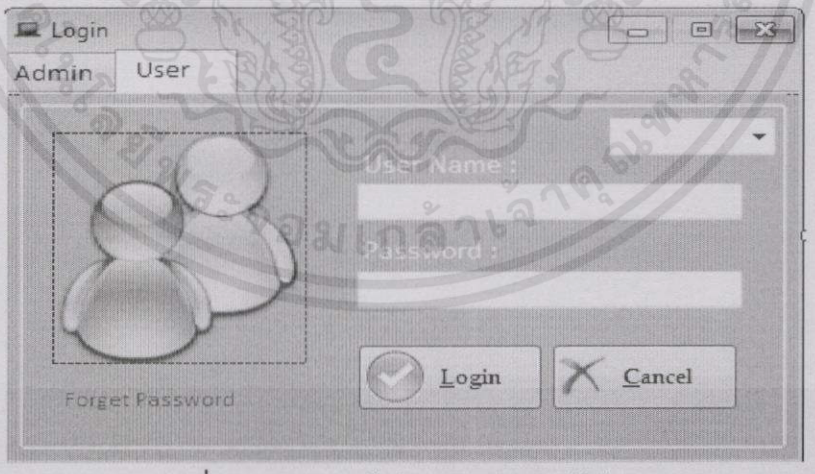
โปรแกรมควบคุมการทำงานกระบอกสูบแบบเป็นลำดับมีรายละเอียดต่างๆดังนี้

1.เมื่อเปิดโปรแกรมขึ้นมาจะปรากฏหน้าต่างสำหรับให้ผู้ดูแลหรือผู้ใช้ Log in ซึ่งผู้ใช้ต้องกรอกข้อมูล Username และ Password ให้ถูกต้อง



รูปที่ 3.14 แสดงหน้าต่าง Log in ของผู้ดูแล

หากเป็นผู้ใช้ต้องการที่จะเข้าใช้งาน จะต้องคลิกที่ปุ่ม User จึงจะสามารถใส่ Username และ Password ได้

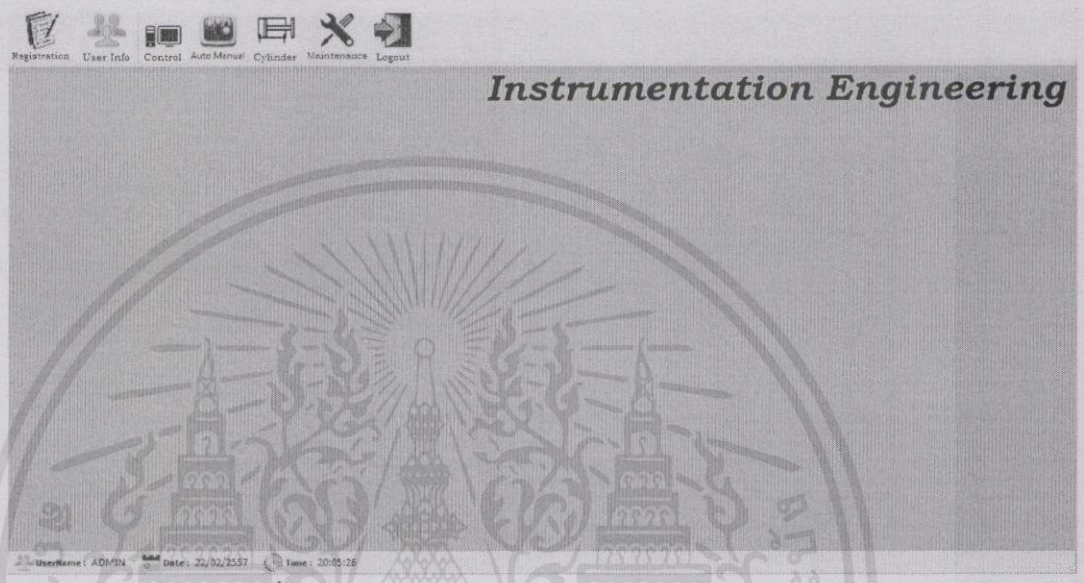


รูปที่ 3.15 แสดงหน้าต่าง Log in ของผู้ใช้งาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4.1 เข้าใช้โดยผู้ดูแล (ADMIN)

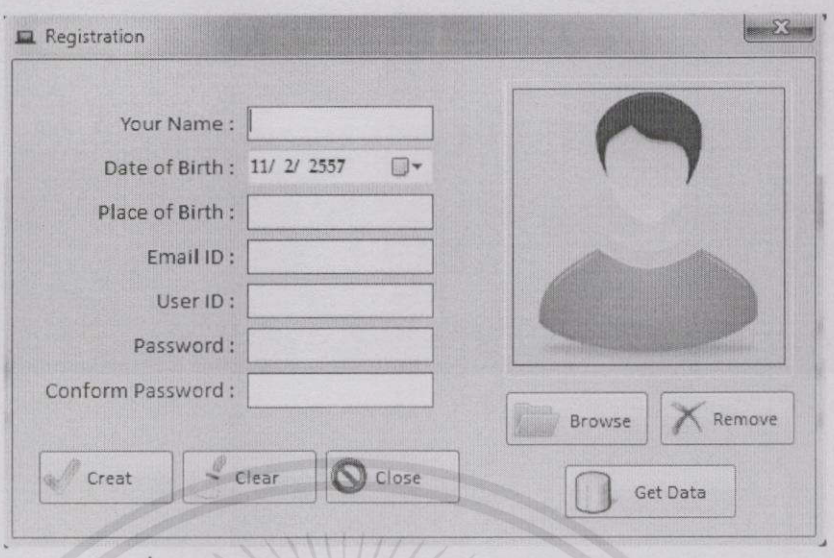
2.เมื่อผู้ดูแล Log in เข้ามาเรียบร้อยจะปรากฏหน้าโปรแกรมหลักขึ้นมา หน้านี้จะเชื่อมต่อกับโปรแกรมต่างๆ และจะมีการบอก วันเวลาปัจจุบันและชื่อผู้ที่เข้าใช้ หน้านี้ยังไม่มีการควบคุมต่างๆแต่อย่างใด



รูปที่ 3.16 แสดงหน้าต่างโปรแกรมหลักของผู้ดูแล

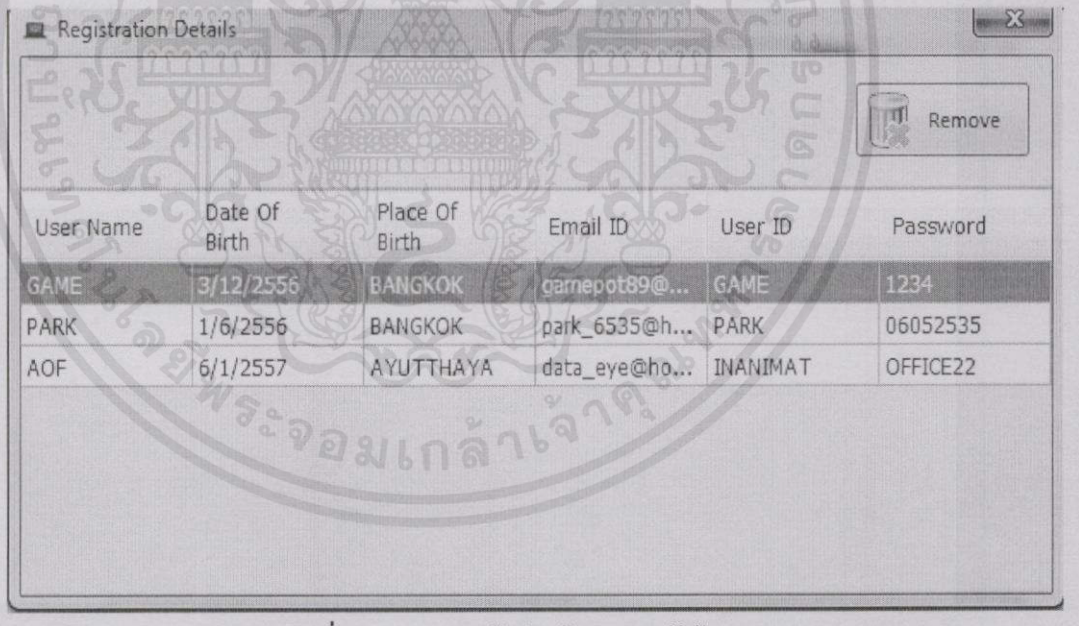
3. เมื่อคลิกปุ่ม Registration จะเข้าสู่หน้าต่าง Registration โดยในหน้านี้จะเป็นการกรอกข้อมูลให้กับผู้ใช้ และกำหนด Username และ Password ให้กับผู้ใช้ ซึ่งจะเก็บข้อมูลต่างเหล่านี้ไว้ในฐานข้อมูล และผู้ดูแลสามารถเข้าไปดูข้อมูลของผู้ใช้แต่ละคนได้ โดยคลิกที่ปุ่ม Get Data ก็จะแสดงข้อมูลต่างๆของผู้ใช้แต่ละคนไว้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.17 แสดงหน้าต่างโปรแกรมการกรอกข้อมูลของผู้ใช้

หากคลิกปุ่ม Get Data จะปรากฏหน้าต่างรายละเอียดของผู้เข้าใช้แต่ละคนขึ้นมา



รูปที่ 3.18 แสดงหน้าต่างข้อมูลของผู้ใช้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.เมื่อคลิกปุ่ม User Info จะปรากฏหน้าต่างแสดงข้อมูลของผู้ที่เข้าอยู่ขณะนั้น และสามารถเข้าไปแก้ไขข้อมูลหรือเปลี่ยน Password ได้ โดยคลิกที่ Change Info และ Change Password

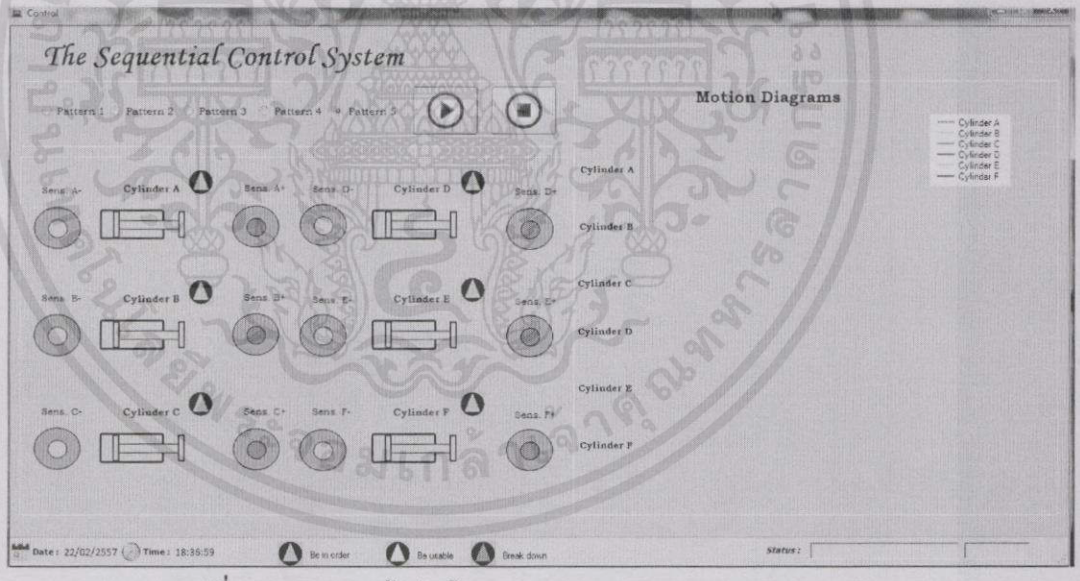
รูปที่ 3.19 แสดงหน้าต่างรายละเอียดข้อมูลของผู้ที่เข้าใช้

รูปที่ 3.20 แสดงหน้าต่างแก้ไขข้อมูลของผู้ที่เข้าใช้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 3.21 แสดงหน้าต่างแก้ไขข้อมูล Username และ Password ของผู้ที่เข้าใช้

5.โปรแกรมการควบคุมและการแสดงสถานะของการทำงาน คือเมื่อผู้ใช้ต้องการสั่งการให้กระบอกสูบทำงานจะสามารถทำได้โดยการคลิกปุ่ม Control ที่หน้าโปรแกรมหลัก ก็จะปรากฏหน้าโปรแกรมการควบคุมและแสดงสถานะกระบอกสูบขึ้นมา



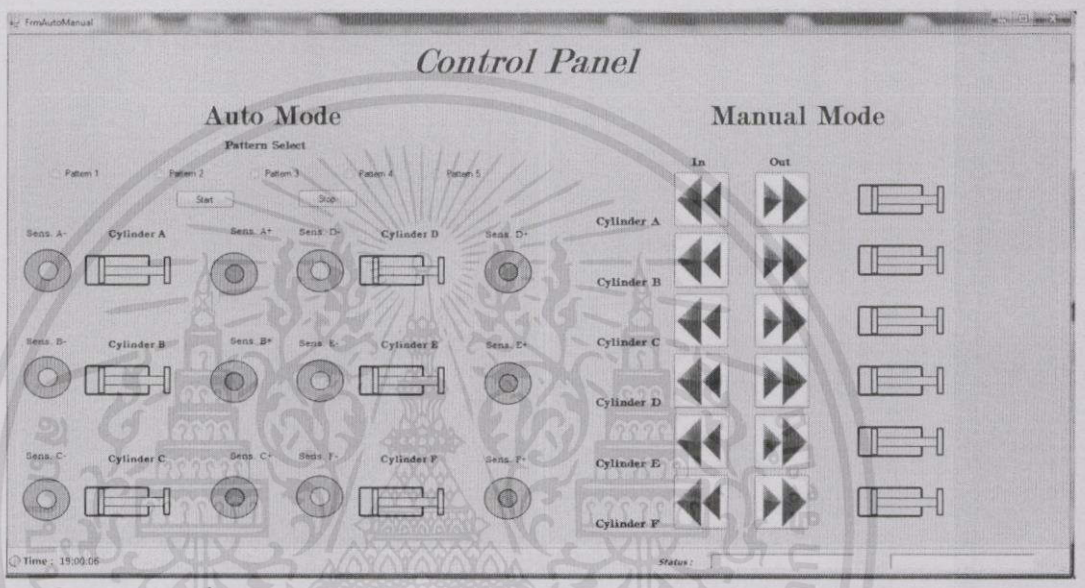
รูปที่ 3.22 แสดงหน้าต่างโปรแกรมการควบคุมและแสดงผล

การที่ผู้ใช้ต้องการให้กระบอกสูบทำงาน จะต้องเลือกรูปแบบการทำงานของกระบอกสูบก่อน จากนั้นจึงทำการคลิกปุ่ม Start กระบอกสูบก็จะทำงานตามรูปแบบที่ผู้ใช้กำหนด ในขณะที่กระบอกสูบทำงาน ที่หน้าต่างหน้า Control จะมี ส่วนแสดงสถานะของกระบอกสูบและเซนเซอร์ซึ่งจะทำงานสอดคล้องกับอุปกรณ์ภายนอก เพื่อใช้ในการสังเกตการณ์สถานะของกระบอกสูบและเซนเซอร์ และยัง

เอกสารนี้เป็น
ไม่ว่ากรณี

มีไฟเตือนสภาพการใช้งานของกระบอกสูบว่า ขณะที่สั่งการควบคุม กระบอกสูบมีความพร้อมที่จะใช้งานได้หรือไม่ โดยในการทดลองได้แบ่งระดับไว้ 3 ระดับ คือ เขียว=พร้อมใช้งาน, เหลือง=ยังพอใช้งานต่อไปได้, แดง=ควรที่จะซ่อมบำรุงหรือเปลี่ยนกระบอกสูบ หากต้องการเปลี่ยนรูปแบบการทำงานจะต้องคลิกปุ่ม Stop ก่อนเสมอ กระบอกสูบจะกลับมาอยู่ในสถานะปิดหมดทุกกระบอก

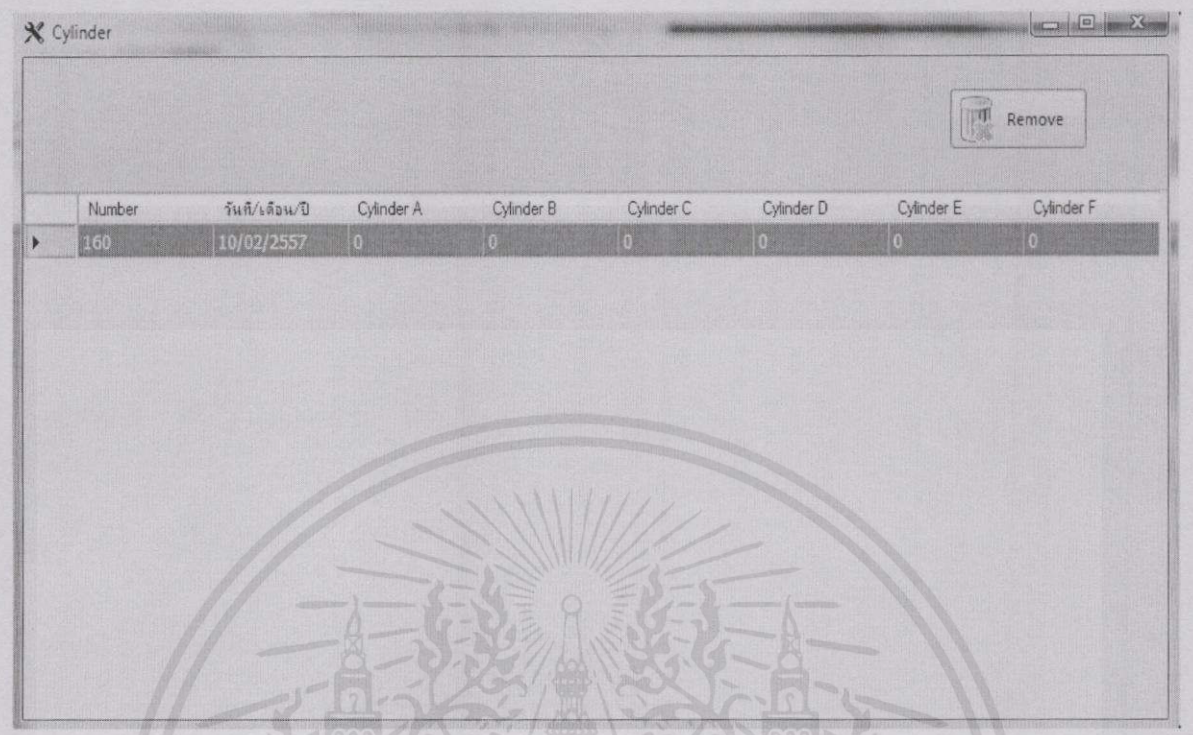
6.โปรแกรมการควบคุมแบบกำหนดเอง เมื่อมีการคลิกปุ่ม Auto Manual ที่หน้าโปรแกรมหลักแล้ว ก็จะปรากฏหน้าต่างโปรแกรมการควบคุมด้วยมือขึ้นมา



รูปที่ 3.23 แสดงหน้าต่างโปรแกรมการควบคุมแบบกำหนดเอง

7.โปรแกรมการแสดงค่าจำนวนครั้งการทำงานของกระบอกสูบ เป็นการแสดงจำนวนครั้งที่กระบอกสูบทำงานใน 1 ครั้งที่ผู้ใช้สั่งการควบคุมและหยุดการทำงาน เมื่อผู้ใช้คลิกปุ่ม Cylinder ที่หน้าต่างโปรแกรมหลักก็จะปรากฏหน้าต่างโปรแกรมแสดงจำนวนครั้งการทำงานกระบอกสูบขึ้นมา โดยจะแสดงผลในแบบของตาราง หากผู้ใช้ต้องการลบค่าแถวใดในตาราง สามารถทำได้โดยเลือกแถวที่ต้องการจะลบ จากนั้นคลิกที่ปุ่ม Remove ค่าในแถวนั้นก็จะถูกลบออกไปจากฐานข้อมูล

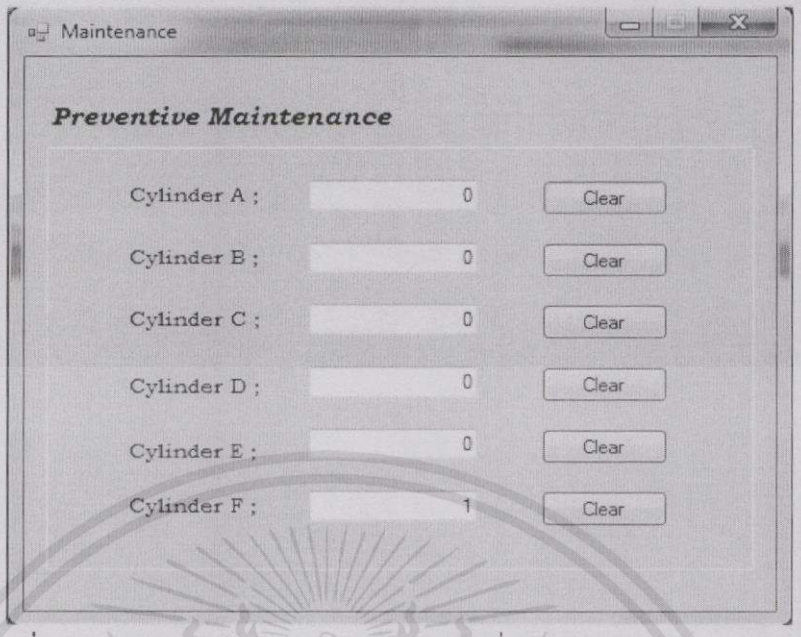
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.24 แสดงหน้าต่างโปรแกรมแสดงค่าจำนวนครั้งการทำงานของกระบอกสูบ

8.โปรแกรมแก้ไขข้อมูลเมื่อมีการซ่อมบำรุงกระบอกสูบ หากผู้ใช้ต้องการจะเข้าสู่โปรแกรมนี้ สามารถทำได้โดยคลิกที่ปุ่ม Maintenance ก็จะแสดงหน้าต่างของโปรแกรมขึ้นมา ซึ่งจะมีค่าจำนวนครั้งทั้งหมดที่กระบอกสูบทำงานของทั้ง 6 กระบอก เมื่อค่าที่แสดงนี้เกินเงื่อนไขที่กำหนด เช่น ตั้งค่าไว้ว่าการทำงานของกระบอกสูบแต่ละกระบอกทำงานได้สูงสุด 100 ครั้ง หากกระบอกสูบทำงานเกิน ไฟเตือนสภาพกระบอกสูบก็จะเป็นสีแดง (ผู้ใช้ควรที่จะซ่อมบำรุงหรือเปลี่ยนใหม่) เมื่อทำการซ่อมบำรุงหรือเปลี่ยนกระบอกสูบแล้ว ผู้ใช้จะต้องทำการ Reset ค่าจำนวนครั้งการทำงานของกระบอกสูบใหม่ด้วย โดยคลิกที่ปุ่ม Clear ของกระบอกสูบนั้นๆ ค่าในช่อง Textbox ก็จะกลับเริ่มที่ 0 ใหม่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

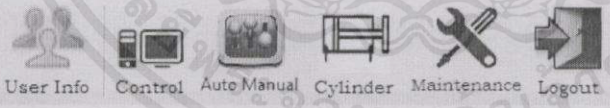


รูปที่ 3.25 แสดงหน้าต่างโปรแกรมแก้ไขข้อมูลเมื่อมีการซ่อมบำรุงกระบอกสูบ

8.เมื่อต้องการออกจากโปรแกรมจะต้องคลิกที่ปุ่ม Log out จากนั้นจะแสดงหน้าต่าง Login ดังเดิม

3.4.2 เข้าใช้โดยผู้ใช้

ถ้าหากโปรแกรม Log in โดยผู้ใช้ หน้าต่างหลักของโปรแกรมจะคล้ายคลึงกับการเข้าใช้โดยผู้ดูแล แตกต่างที่หน้าต่างหลักจะไม่มีปุ่มเข้าหน้าโปรแกรม Registration เท่านั้น ส่วนรายละเอียดอื่นๆจะเหมือนกับการ Log in โดยผู้ดูแล



รูปที่ 3.26 แสดงปุ่มที่มีในหน้าโปรแกรมหลักของผู้ใช้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

ผลการทดลอง

4.1 ผลการทดลอง RUN โปรแกรมควบคุมการทำงานของกระบอกสูบ

เมื่อเปิดโปรแกรมขึ้นมาจะปรากฏหน้าต่าง Log in เสมอ ซึ่งจะเป็นการให้ผู้ดูแลหรือผู้ใช้ทำการ Log in เพื่อเข้าใช้งานโปรแกรม

โปรแกรมที่ใช้ในการควบคุมกระบอกสูบนั้นมีอยู่ด้วยกัน 5 ส่วนใหญ่ๆ ดังนี้

4.1.1 หน้าโปรแกรมหลัก มีหน้าที่เชื่อมต่อกับโปรแกรมต่างๆ ประกอบด้วยปุ่มต่างๆดังนี้

รูปที่ 4.1 แสดงหน้าต่างโปรแกรมหลัก

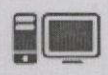
1.

ใช้ในการเพิ่ม เปลี่ยน หรือลบข้อมูลของผู้ใช้

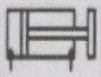
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

*** ถ้าหากผู้ใช้ทำการ log in เข้ามา จะไม่มีปุ่มนี้ปรากฏที่หน้าต่างหลัก

2. User Info .การแสดงข้อมูลของผู้ที่เข้าใช้อยู่ขณะนั้น



3. Control เป็นการควบคุมและแสดงสถานะการทำงาน

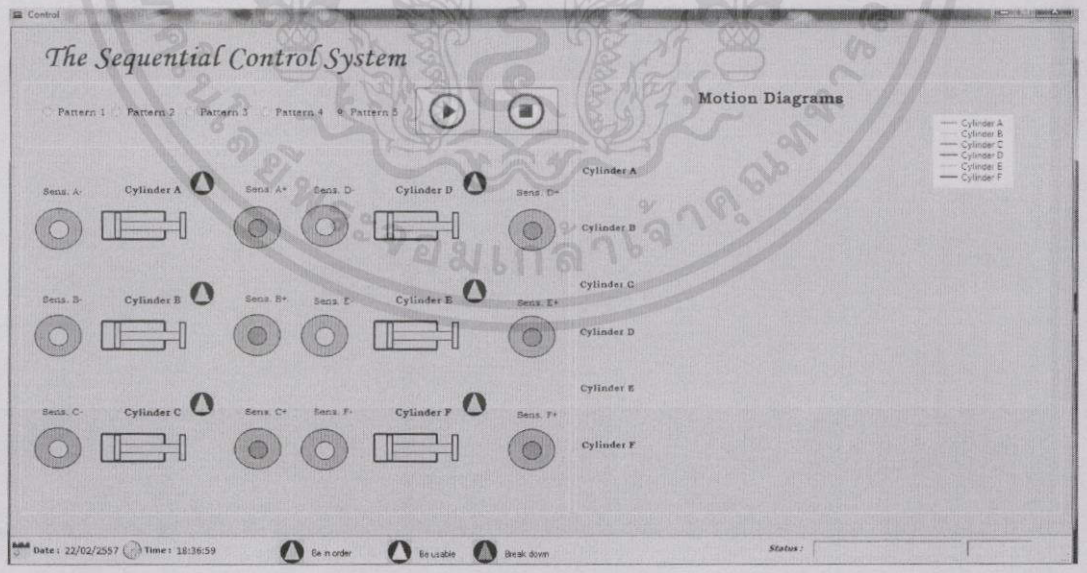


4. Cylinder การแสดงข้อมูลการทำงานของกระบอกสูบ โดยแสดงออกเป็นตาราง

5. Maintenance เป็นการแก้ไขข้อมูล เมื่อมีการซ่อมบำรุงกระบอกสูบ

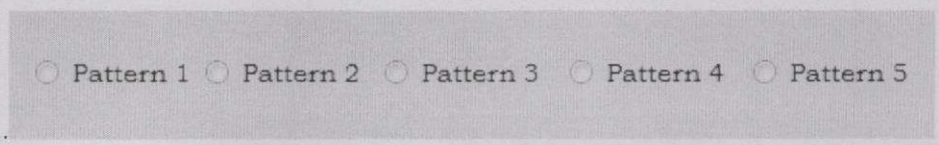
6. Logout การออกจากโปรแกรม

4.1.2 โปรแกรมการควบคุมและการแสดงสถานะของการทำงาน เมื่อกดปุ่ม Control ที่หน้าโปรแกรมหลัก จะปรากฏหน้าโปแกรมที่ใช้ควบคุมขึ้นมา



รูปที่ 4.2 แสดงหน้าต่างโปรแกรมการควบคุมและการแสดงผล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

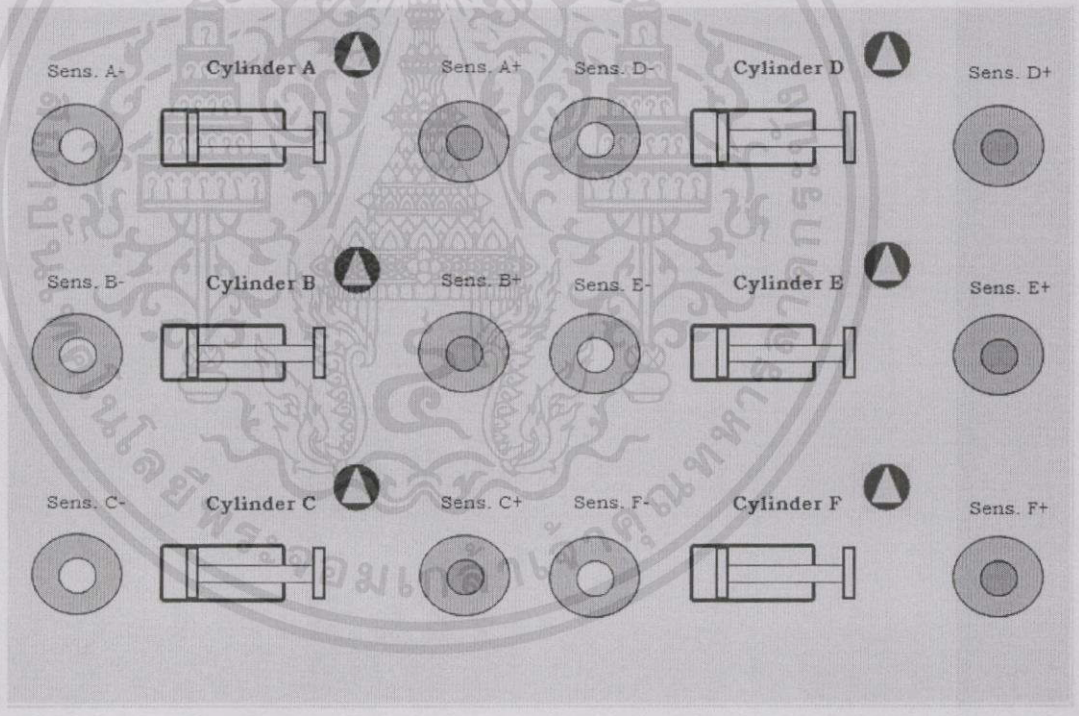
1. 

เป็นการเลือกว่าต้องการให้กระบอกสูบทำงานเป็นลำดับในรูปแบบใด

2. เป็นปุ่มที่ใช้ในการสั่งเริ่มการทำงาน

3. เป็นปุ่มสั่งหยุดการทำงาน และเมื่อคลิกปุ่มนี้จะมีการถามผู้ใช้ว่าต้องการบันทึกข้อมูลลงในฐานข้อมูลหรือไม่

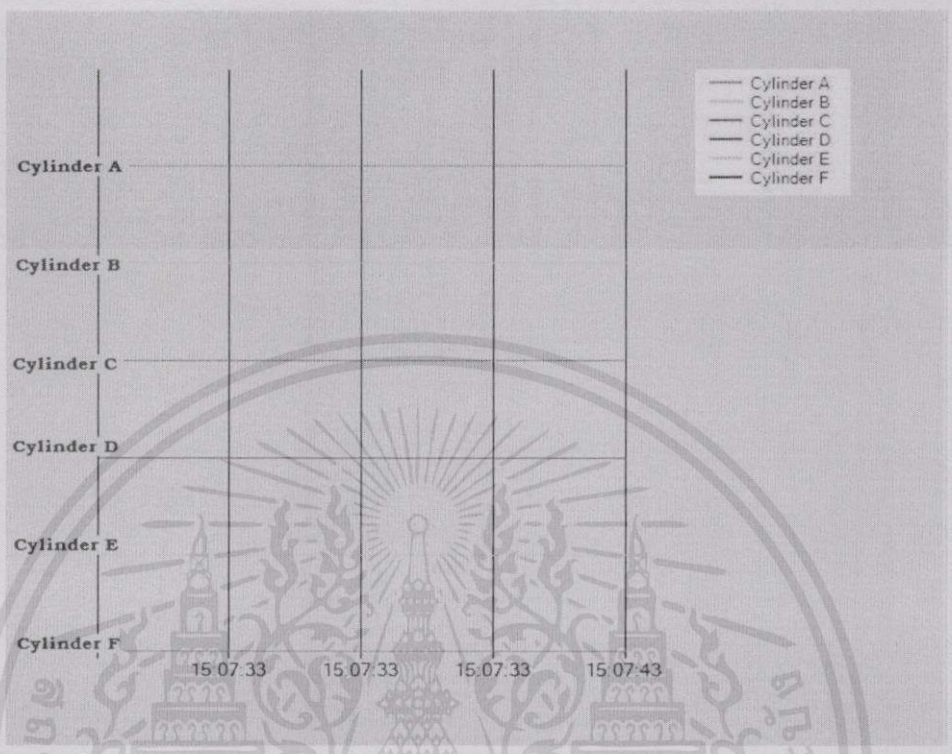
4. ส่วนแสดงสถานะของกระบอกสูบและเซนเซอร์ และไฟเตือนสภาพการใช้งานของกระบอกสูบ



รูปที่ 4.3 ส่วนของการแสดงผลสถานะของกระบอกสูบและเซนเซอร์

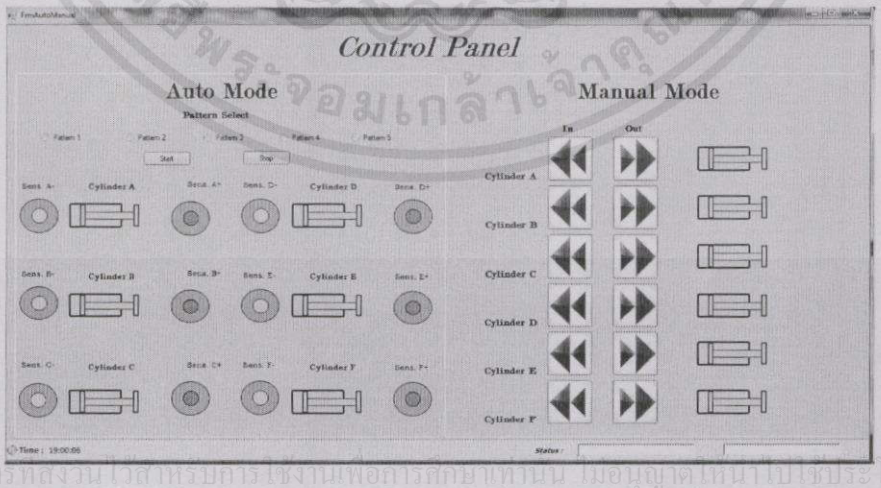
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5. ส่วนแสดงผลการทำงานของกระบอกสูบในรูปแบบของกราฟ Timing Diagrams แบบ Real time



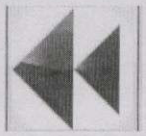
รูปที่ 4.4 ส่วนของการแสดงกราฟ Timing Diagrams

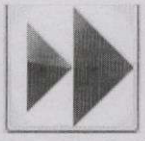
4.1.3 โปรแกรมการควบคุมแบบกำหนดเอง เมื่อคลิกปุ่ม Auto Manual ที่หน้าโปรแกรมหลัก จะปรากฏหน้าโปรแกรมการควบคุมด้วยมือขึ้นมา



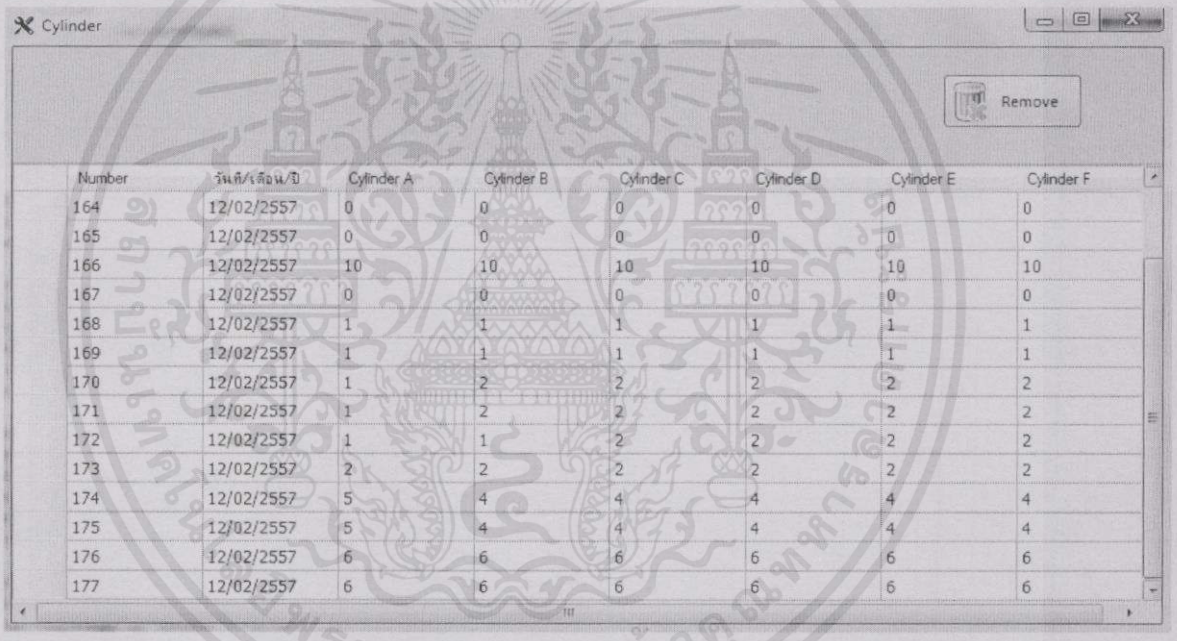
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 4.5 หน้าจอการควบคุมแบบอัตโนมัติและแบบกำหนดเอง

1.  เป็นปุ่มที่มีผลทำให้กระบอกสูบเลื่อนเข้า

2.  เป็นปุ่มที่มีผลทำให้กระบอกสูบเคลื่อนที่ออก

4.1.4 โปรแกรมแสดงข้อมูลจำนวนครั้งการทำงานของกระบอกสูบ เมื่อคลิกปุ่ม Cylinder ที่หน้าโปรแกรมหลัก จะปรากฏหน้าโปแกรมแสดงข้อมูลขึ้นมา ซึ่งจะแสดงข้อมูลจำนวนครั้งการทำงานของกระบอกสูบแต่ละกระบอกในรูปแบบตาราง

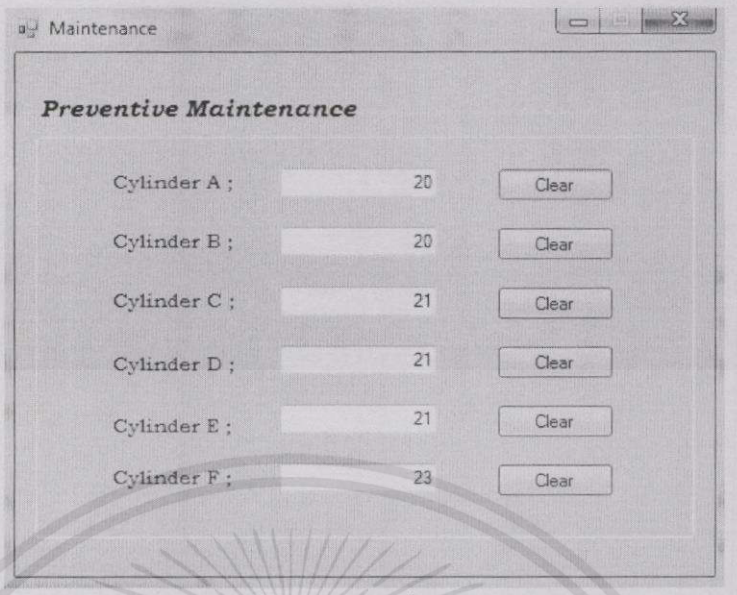


Number	วันที่/เดือน/ปี	Cylinder A	Cylinder B	Cylinder C	Cylinder D	Cylinder E	Cylinder F
164	12/02/2557	0	0	0	0	0	0
165	12/02/2557	0	0	0	0	0	0
166	12/02/2557	10	10	10	10	10	10
167	12/02/2557	0	0	0	0	0	0
168	12/02/2557	1	1	1	1	1	1
169	12/02/2557	1	1	1	1	1	1
170	12/02/2557	1	2	2	2	2	2
171	12/02/2557	1	2	2	2	2	2
172	12/02/2557	1	1	2	2	2	2
173	12/02/2557	2	2	2	2	2	2
174	12/02/2557	5	4	4	4	4	4
175	12/02/2557	5	4	4	4	4	4
176	12/02/2557	6	6	6	6	6	6
177	12/02/2557	6	6	6	6	6	6

รูปที่ 4.6 แสดงโปรแกรมแสดงข้อมูลจำนวนครั้งการทำงานของกระบอกสูบ

4.1.5 โปรแกรมการแก้ไขข้อมูลเมื่อมีการซ่อมบำรุงกระบอกสูบ เมื่อคลิกปุ่ม Maintenance ที่หน้าโปรแกรมหลัก จะปรากฏหน้าต่างโปรแกรมขึ้นมา ซึ่งจะมีช่องแสดงถึงจำนวนครั้งที่กระบอกสูบทำงานไปทั้งหมด และมีปุ่ม Clear ของแต่ละกระบอกสูบ ที่ใช้ในการแก้ไขข้อมูลให้เป็น0เมื่อมีการซ่อมบำรุงกระบอกสูบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.7 แสดงหน้าต่างโปรแกรมแก้ไขข้อมูลเมื่อมีการซ่อมบำรุงกระบอกสูบ

4.2 การทดสอบการทำงาน

เพื่อศึกษาว่าการสั่งการที่ได้จากส่งข้อมูลจากคอมพิวเตอร์ไปหาตัวควบคุมนั้น ได้ผลการทำงานเป็นไปตามที่เลือกไว้หรือไม่ และการแสดงผล (กระบอกสูบ, เซนเซอร์, กราฟ Timing Diagrams) ตรงตามการทำงานหรือไม่

เมื่อ Run โปรแกรมจากคอมพิวเตอร์ จากนั้นทำการเลือกรูปแบบการทำงานของกระบอกสูบและคลิกปุ่ม Start

หมายเหตุ +, 1 คือ กระบอกสูบเคลื่อนที่ออก (เปิด)

-, 0 คือ กระบอกสูบเคลื่อนที่เข้า (ปิด)

4.2.1 การทดสอบการทำงานของระบบด้วยการเลือก “Pattern1”

เพื่อสังเกตการเปลี่ยนแปลงของหน้าจอ HMI ว่ามีการเปลี่ยนแปลงตามการทำงานของระบบหรือไม่ เมื่อเลือก “Pattern1”

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.1 การทดสอบการทำงานของกระบอกสูบ รูปแบบที่ 1

สถานะกระบอกสูบ	ข้อมูลที่อ่านจากตัวควบคุม	ข้อมูลที่นำมาเข้าเงื่อนไข	การแสดงผลกระบอกสูบ A	การแสดงผลกระบอกสูบ B	การแสดงผลกระบอกสูบ C	การแสดงผลกระบอกสูบ D	การแสดงผลกระบอกสูบ E	การแสดงผลกระบอกสูบ F
ปิดหมด	@00RR00155445*	1554	0	0	0	0	0	0
A+	@00RR00155849*	1550	1	0	0	0	0	0
A-	@00RR00155445*	1554	0	0	0	0	0	0
B+	@00RR00156446*	1564	0	1	0	0	0	0
B-	@00RR00155445*	1554	0	0	0	0	0	0
C+	@00RR00159449*	1594	0	0	1	0	0	0
C-	@00RR00155445*	1554	0	0	0	0	0	0
D+	@00RR00165446*	1654	0	0	0	1	0	0
D-	@00RR00155445*	1554	0	0	0	0	0	0
E+	@00RR00196449*	1954	0	0	0	0	1	0
E-	@00RR00155445*	1554	0	0	0	0	0	0
F+	@00RR00255446*	2554	0	0	0	0	0	1
F-	@00RR00155445*	1554	0	0	0	0	0	0

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2.2 การทดสอบการทำงานของระบบด้วยการเลือก “Pattern2”

เพื่อสังเกตการเปลี่ยนแปลงของหน้าจอ HMI ว่ามีการเปลี่ยนแปลงตามการทำงานของระบบหรือไม่ เมื่อเลือก “Pattern2”

ตารางที่ 4.2 การทดสอบการทำงานของกระบอกสูบ รูปแบบที่ 2

สถานะกระบอกสูบ	ข้อมูลที่อ่านจากตัวควบคุม	ข้อมูลที่นำมาเข้าเงื่อนไข	การแสดงผลกระบอกสูบ A	การแสดงผลกระบอกสูบ B	การแสดงผลกระบอกสูบ C	การแสดงผลกระบอกสูบ D	การแสดงผลกระบอกสูบ E	การแสดงผลกระบอกสูบ F
ปิดหมด	@00RR00155445*	1554	0	0	0	0	0	0
A+	@00RR00155849*	1550	1	0	0	0	0	0
B+	@00RR0015684A*	1560	1	1	0	0	0	0
C+	@00RR0015A83D*	15A0	1	1	1	0	0	0
D+	@00RR0016A83E*	16A0	1	1	1	1	0	0
E+	@00RR001AA849*	1AA0	1	1	1	1	1	0
F+	@00RR002AA84A*	2AA0	1	1	1	1	1	1
F-	@00RR001AA849*	1AA0	1	1	1	1	1	0
E-	@00RR0016A83E*	16A0	1	1	1	1	0	0
D-	@00RR0015A83D*	15A0	1	1	1	0	0	0
C-	@00RR0015684A*	1560	1	1	0	0	0	0
B-	@00RR00155849*	1550	1	0	0	0	0	1
A-	@00RR00155445*	1554	0	0	0	0	0	0

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2.3 การทดสอบการทำงานของระบบด้วยการเลือก “Pattern3”

เพื่อสังเกตการเปลี่ยนแปลงของหน้าจอ HMI ว่ามีการเปลี่ยนแปลงตามการทำงานของระบบหรือไม่ เมื่อเลือก “Pattern3”

ตารางที่ 4.3 การทดสอบการทำงานของกระบอกสูบ รูปแบบที่ 3

สถานะกระบอกสูบ	ข้อมูลที่อ่านจากตัวควบคุม	ข้อมูลที่นำมาเข้าเงื่อนไข	การแสดงผลกระบอกสูบ A	การแสดงผลกระบอกสูบ B	การแสดงผลกระบอกสูบ C	การแสดงผลกระบอกสูบ D	การแสดงผลกระบอกสูบ E	การแสดงผลกระบอกสูบ F
ปิดหมด	@00RR00155445*	1554	0	0	0	0	0	0
A+,B+	@00RR0015684A*	1560	1	1	0	0	0	0
A-,B-	@00RR00155445*	1554	0	0	0	0	0	0
C+,D+	@00RR0016944A*	1694	0	0	1	1	0	0
C-,D-	@00RR00155445*	1554	0	0	0	0	0	0
E+,F+	@00RR0029544A*	2954	0	0	0	0	1	1
E-,F-	@00RR00155445*	1554	0	0	0	0	0	0

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2.4 การทดสอบการทำงานของระบบด้วยการเลือก “Pattern4”

เพื่อสังเกตการเปลี่ยนแปลงของหน้าจอ HMI ว่ามีการเปลี่ยนแปลงตามการทำงานของระบบหรือไม่ เมื่อเลือก “Pattern4”

ตารางที่ 4.4 การทดสอบการทำงานของกระบอกสูบ รูปแบบที่ 4

สถานะกระบอกสูบ	ข้อมูลที่อ่านจากตัวควบคุม	ข้อมูลที่นำมาเข้าเงื่อนไข	การแสดงผลกระบอกสูบ A	การแสดงผลกระบอกสูบ B	การแสดงผลกระบอกสูบ C	การแสดงผลกระบอกสูบ D	การแสดงผลกระบอกสูบ E	การแสดงผลกระบอกสูบ F
ปิดหมด	@00RR00155445*	1554	0	0	0	0	0	0
A+	@00RR00155849*	1550	1	0	0	0	0	0
B+	@00RR0015684A*	1560	1	1	0	0	0	0
C+	@00RR0015A83D*	15A0	1	1	1	0	0	0
A-,B-,C-	@00RR00155445*	1554	0	0	0	0	0	0
D+	@00RR00165446*	1654	0	0	0	1	0	0
E+	@00RR001A5431*	1A54	0	0	0	1	1	0
F+	@00RR002A5432*	2A54	0	0	0	1	1	1
D-,E-,F-	@00RR00155445*	1554	0	0	0	0	0	0

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2.5 การทดสอบการทำงานของระบบด้วยการเลือก “Pattern5”

เพื่อสังเกตการเปลี่ยนแปลงของหน้าจอ HMI ว่ามีการเปลี่ยนแปลงตามการทำงานของระบบหรือไม่ เมื่อเลือก “Pattern5”

ตารางที่ 4.5 การทดสอบการทำงานของกระบอกสูบ รูปแบบที่ 5

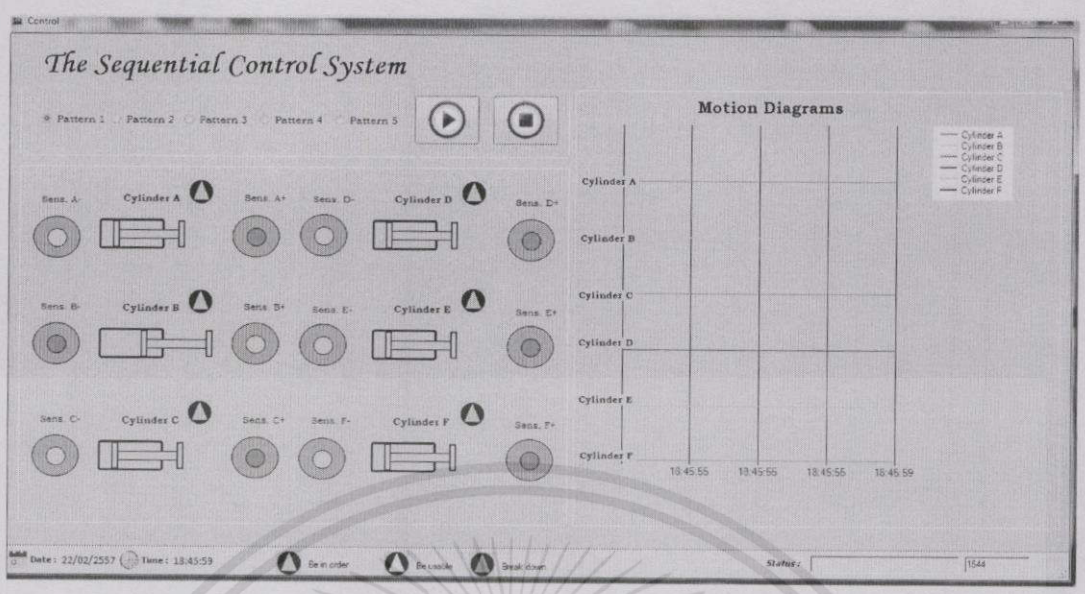
สถานะกระบอกสูบ	ข้อมูลที่อ่านจากตัวควบคุม	ข้อมูลที่นำมาเข้าเงื่อนไข	การแสดงผลกระบอกสูบ A	การแสดงผลกระบอกสูบ B	การแสดงผลกระบอกสูบ C	การแสดงผลกระบอกสูบ D	การแสดงผลกระบอกสูบ E	การแสดงผลกระบอกสูบ F
ปิดหมด	@00RR00155445*	1554	0	0	0	0	0	0
A+	@00RR00155849*	1550	1	0	0	0	0	0
B+	@00RR0015684A*	1560	1	1	0	0	0	0
C+	@00RR0015A83D*	15A0	1	1	1	0	0	0
D+	@00RR0016A83E*	16A0	1	1	1	1	0	0
E+	@00RR001AA849*	1AA0	1	1	1	1	1	0
F+	@00RR002AA84A*	2AA0	1	1	1	1	1	1
C-,F-	@00RR001A683E*	1A60	1	1	0	1	1	0
B-,E-	@00RR0016584A*	1650	1	0	0	1	0	0
A-,D-	@00RR001544445*	1554	0	0	0	0	0	0

4.2.6 การทดสอบการทำงานของสัญญาณไฟเตือนสภาพการใช้งานของกระบอกสูบ

เพื่อสังเกตการณ์ของสภาวะกระบอกสูบว่ามีการทำงานใกล้จะครบกำหนดแล้วหรือยัง ในโรงงานอุตสาหกรรมนั้นหากเครื่องมือมีการทำงานเกินค่าที่กำหนดไว้อาจจะทำให้เครื่องมือเสียหายได้

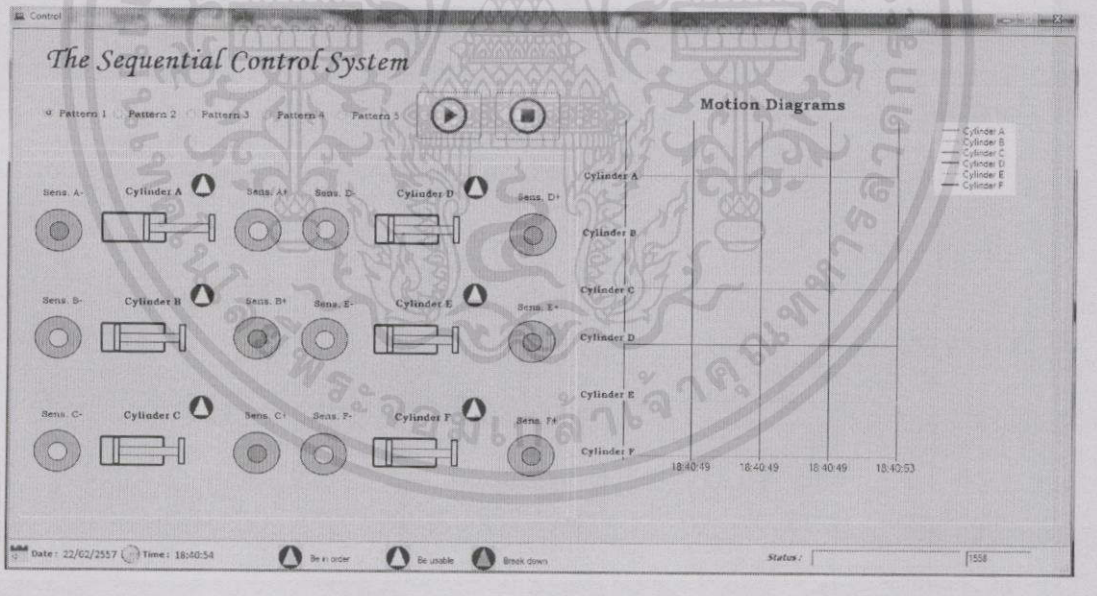
ในการทดสอบได้กำหนดจำนวนครั้งการทำงานของกระบอกสูบ โดยจะกำหนดให้ไฟเตือนสภาพในการเริ่มต้นนั้นมีสีเขียว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.8 แสดงสภาวะปกติไฟเตือนสภาพ แสดงสีเขียว

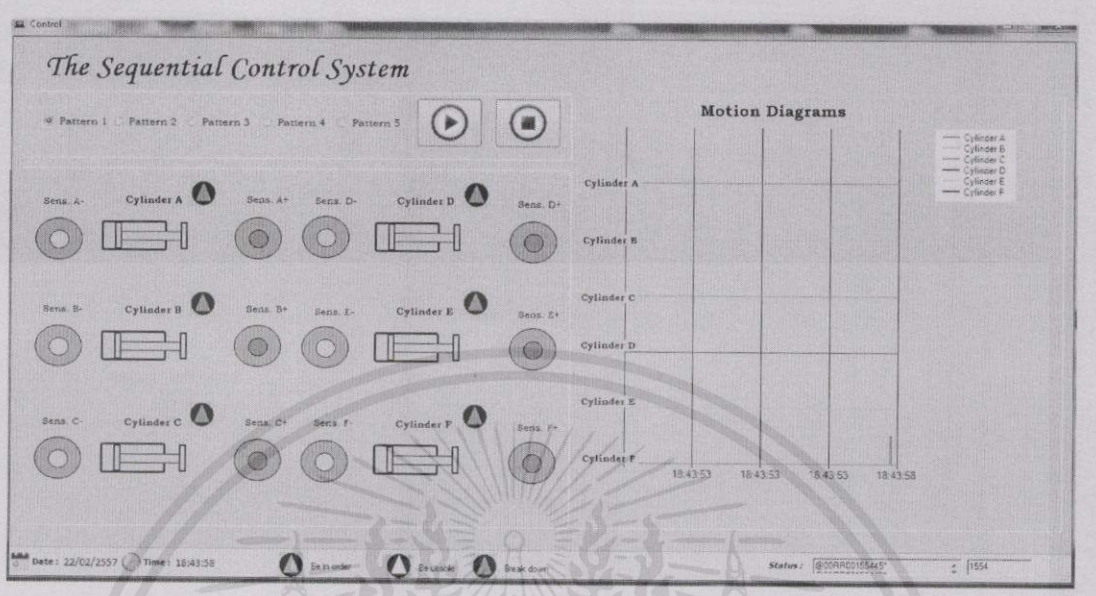
ถ้ากระบอกสูบทำงานมากกว่า 5 ครั้ง ไฟเตือนสภาพจะแสดงเป็นสีเหลือง



รูปที่ 4.9 สภาวะการทำงานมากกว่า 5 ครั้ง ไฟเตือนสภาพแสดงสีเหลือง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หากกระบอกสูบยังคงทำงานเกิน 10 ครั้ง ไฟเตือนสภาพจะเปลี่ยนเป็นสีแดง



รูปที่ 4.10 สภาวะการทำงานมากกว่า 10 ครั้ง ไฟเตือนสภาพแสดงสีแดง

4.2.7 การทดสอบโปรแกรมการควบคุมแบบกำหนดเอง

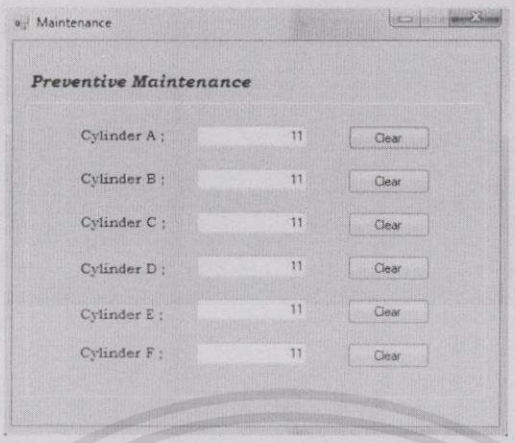
เมื่อรันโปรแกรมแล้ว เข้าไปยังหน้าโปรแกรมการควบคุมแบบกำหนดเอง ได้ทดลองสั่งการทั้งแบบอัตโนมัติและแบบกำหนดเอง ถ้าสั่งการแบบอัตโนมัติกระบอกสูบจะทำงานตามรูปแบบการทำงานที่ผู้ใช้ได้เลือก และทดลองสั่งการแบบกำหนดเองกระบอกสูบก็ทำงานตามที่ผู้ใช้สั่งการ และการแสดงผลของสถานะกระบอกสูบและเซนเซอร์เป็นไปตามการทำงานของกระบอกสูบ

4.2.8 การทดสอบโปรแกรมการแก้ไขข้อมูลเมื่อมีการซ่อมบำรุงกระบอกสูบ

จากการทดลองในหัวข้อ 4.2.6 การทำงานของกระบอกสูบจะมีการบันทึกค่าการทำงานลงในหน้าต่างของโปรแกรมซ่อมบำรุง โดยจะแสดงจำนวนครั้งการทำงานของกระบอกสูบ หากมีการซ่อมบำรุงกระบอกสูบค่าที่แสดงในหน้าต่างจะกลับไปเป็น “0” แสดงว่ามีการซ่อมบำรุงแล้ว

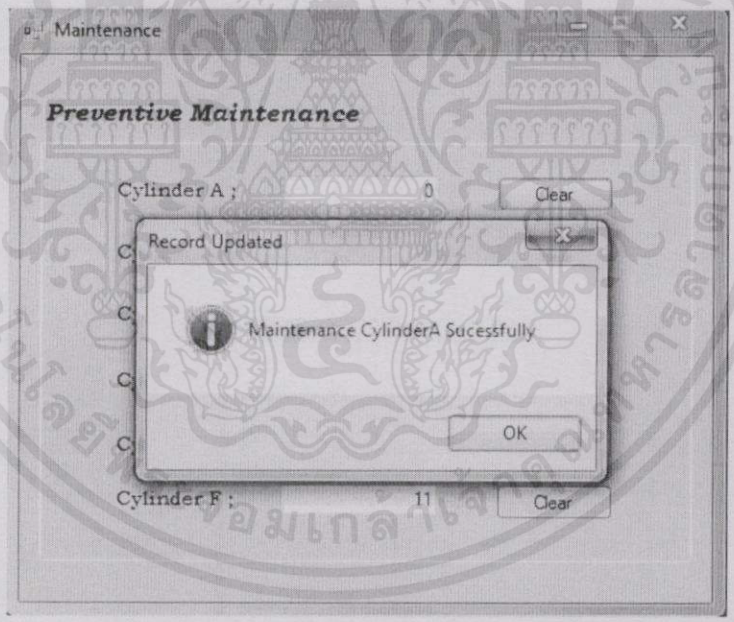
ในการทดลองนั้น เมื่อเปิดหน้าต่างการซ่อมบำรุงขึ้นมาจะแสดงผลการทำงานของกระบอกสูบ ดังรูป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.11 แสดงจำนวนครั้งการทำงานของกระบอกสูบทั้งหมด

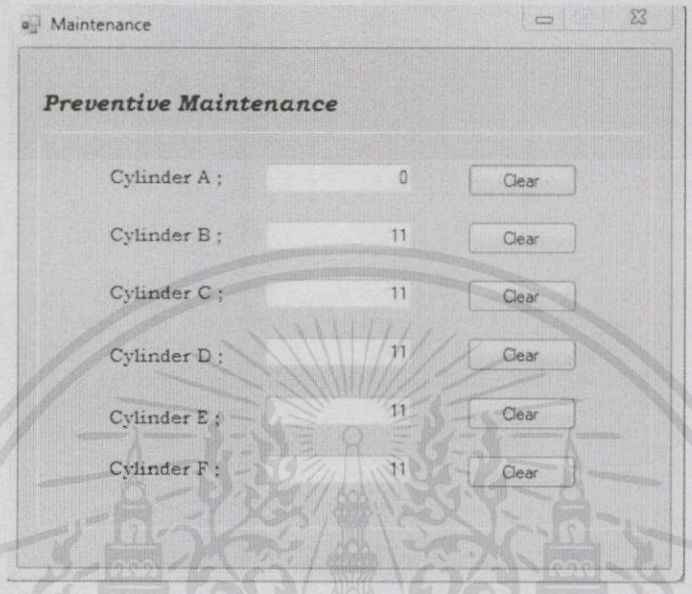
หากกดปุ่ม "Clear" จะทำการรีเซ็ตค่าของกระบอกสูบนั้นๆ กลับไปเป็นค่าเริ่มต้น และส่งผลให้ไฟเตือนสภาพของกระบอกสูบนั้นๆ กลับเป็นสีเขียว



รูปที่ 4.12 แสดงการทำงานเมื่อมีการกดปุ่ม "Clear"

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ถ้ากระบอกสูบใดได้รับการซ่อมบำรุง หรือมีการเปลี่ยนกระบอกสูบใหม่ ผู้ใช้จะต้องเข้ามาในโปรแกรมนี้



รูปที่ 4.13 แสดงการค่ากลับเป็นค่าเริ่มต้น หลังจากกดปุ่ม “Clear”

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5
สรุปผลการทดลองและข้อเสนอแนะ

5.1 สรุปผลการทดลอง

ในปริญญานิพนธ์ฉบับนี้ ได้นำเสนอการเกี่ยวกับการควบคุมการทำงานเป็นลำดับขั้นตอน ด้วยคอมพิวเตอร์ ผ่านทางตัวควบคุมแบบพีแอลซี (PLC) โดยการใช้โปรแกรม CX-Programmer ในการเขียนชุดคำสั่ง และการแสดงผลออกทางจอมอนิเตอร์ในรูปแบบ HMI ซึ่งมีการแสดงผลการทำงานเป็นแบบเรียลไทม์ ตามการทำงานของเครื่องมือที่เป็นอยู่ ณ เวลาปัจจุบัน ซึ่งปริญญานิพนธ์ฉบับนี้ถือเป็นการจำลองการทำงานของระบบในอุตสาหกรรม ซึ่งในปัจจุบันนั้นมีการใช้คอมพิวเตอร์สั่งการงานเพื่อให้ทำงานเป็นอัตโนมัติมากขึ้น และมีการเก็บค่าเพื่อซ่อมบำรุงอีกด้วย โดยในการควบคุมจะรับข้อมูลจากคอมพิวเตอร์ ส่งข้อมูลผ่านสาย USB to RS232 เข้าสู่ตัวควบคุม PLC ในการเชื่อมต่อกันระหว่างตัวควบคุมกับคอมพิวเตอร์นั้นจะใช้การเชื่อมต่อที่เรียกว่า Host Link ซึ่งตัวควบคุมจะทำการประมวลผลจากโปรแกรมที่เขียนเข้ามา เพื่อส่งสัญญาณไปควบคุมวาล์วควบคุมด้วยกระแสไฟฟ้า (Solenoid valve) เพื่อให้ทำงานเป็นลำดับขั้นตอนตามโปรแกรม และนอกจากนี้ยังมีหน้าต่าง HMI เพื่อให้ผู้ใช้งานสามารถสั่งการได้ง่าย และสามารถที่จะดูการเปลี่ยนแปลงของระบบได้โดยการรับค่าจากเซนเซอร์ที่ติดอยู่กับกระบอกสูบเป็นตัวชี้ว่า สถานะของกระบอกสูบเป็นอย่างไร มีการทำงานอย่างไร จำนวนครั้งที่ลูกสูบทำงานมีความเสี่ยงหรือยังที่อุปกรณ์จะเสียหาย และสุดท้ายจะนำค่าที่ได้บันทึกลงตารางซ่อมบำรุงซึ่งมีการบันทึกค่าลงในฐานข้อมูลเพื่อดูย้อนหลังได้อีกด้วย โดยในภาคเรียนที่ 1 นั้น ได้ทำการศึกษาข้อมูลเกี่ยวกับอุปกรณ์ต่างๆที่เป็นส่วนประกอบในระบบ ซ่อมบำรุงวาล์วควบคุมทางไฟฟ้า และศึกษาการเชื่อมต่อระหว่างคอมพิวเตอร์กับตัวควบคุม ในระบบ Host Link การสร้างหน้า HMI ด้วยโปรแกรม Visual Basic ส่วนในภาคเรียนที่ 2 ได้ทำการสร้างหน้าจอ HMI โดยโปรแกรม Visual Basic และเชื่อมต่อกับตัวควบคุม ส่งค่าและรับค่าจากตัวควบคุมเพื่อทำการสั่งการและแสดงผล และมีการบันทึกค่างฐานข้อมูลโดยใช้โปรแกรม Access อีกด้วย

จากผลการทดลองควบคุมผ่านตัวควบคุม PLC และการสร้างหน้าจอ HMI ทำให้พบว่า ชุดอุปกรณ์สามารถทำงานได้จริง ทั้งส่งค่าและรับค่า การแสดงผลนั้นค่อนข้างถูกต้อง สามารถนำระบบและโปรแกรมนี้ไปพัฒนาต่อ และประยุกต์ในการควบคุมอุปกรณ์ลักษณะอื่นได้หลากหลาย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.2 ข้อเสนอแนะ

ในระบบนี้นั้นตัวควบคุมนั้นมีอัตราการส่งข้อมูลที่ต่ำจึงทำให้บางทีการตอบสนองต่อการรับส่งข้อมูลนั้นเกิดการล่าช้าขึ้น และวาล์วควบคุมทางไฟฟ้ากับกระบอกสูบนั้นเริ่มสึกหรอเนื่องจากใช้ในการทดลอง หากมีการนำไปพัฒนาต่อ จึงควรเปลี่ยนตัวควบคุมให้มีอัตราการรับส่งข้อมูลที่เร็วขึ้น และเพิ่มในส่วนของโปรแกรมเพื่อทำการตรวจสอบการคงทนต่อความเสียหาย (Fault Tolerance) เพื่อทำการตรวจสอบอุปกรณ์ก่อนจะเกิดการเสียหายขึ้น

5.3 อุปสรรคและปัญหา

ในการทดลองเขียนโปรแกรมนั้นได้เกิดความผิดพลาดขึ้นหลายครั้ง จากการที่เซนเซอร์ที่ตรวจจับนั้นระยะเวลาผ่านมานานแล้วจึงทำให้สมบัติของแม่เหล็กหายไป รวมไปถึงอุปกรณ์อื่นๆอีกมากมายที่เกิดความเสื่อมสภาพไปตามเวลา แก้ไขได้โดยการเปลี่ยนเซนเซอร์ใหม่ทั้งชุด ดูและอุปกรณ์เพิ่ม เช่น กระบอกสูบก็ทำความสะอาด เปลี่ยนวาล์วควบคุมด้วยไฟฟ้าใหม่ เป็นต้น อีกหนึ่งสาเหตุคืออาจเกิดจากการเขียนโปรแกรมในการควบคุมไม่ได้อย่างที่ตั้งใจ หรือวิธีในการควบคุมเปลี่ยนไป วิธีแก้ไขก็คือ ทำการศึกษาเกี่ยวกับส่วนประกอบของโปรแกรม ชุดคำสั่งต่างๆ แล้วนำมาเขียนใหม่ เพื่อพัฒนาให้เป็นอย่างที่ต้องการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

บัญชา ปะสีละเตสัง. 2556. พัฒนาแอพพลิเคชั่นด้วย Visual Basic 2010. บริษัท ซีเอ็ดยูเคชั่น จำกัด (มหาชน)

ศุภชัย สมพานิช. 2553. Professional Database Programming with VB 2010 & VC# 2010. บริษัท ไอดีซี พรีเมียร์ จำกัด

ปานเพชร ชินินทร, ขวัญชัย สินทิพย์สมบูรณ์. 2555. นิวแมติกอุตสาหกรรม. บริษัท ซีเอ็ดยูเคชั่น จำกัด (มหาชน)

W. Bolton. 2006. Programmable Logic Controllers Fourth Edition

Microsoft Technical Support. 2013. http://msdn.microsoft.com/en-US/ .

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ก

โปรแกรมที่ใช้ในการทดลอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
'FormMain
Imports System.Data.OleDb

Public Class FrmMain


    Private Sub Timer1_Tick(ByVal sender
As System.Object, ByVal e As
System.EventArgs) Handles Timer1.Tick
        TimeStatusLabel.Text =
Now.ToLongTimeString.ToString()
    End Sub

    Private Sub FrmMain_FormClosing(ByVal
sender As Object, ByVal e As
System.Windows.Forms.FormClosingEventArgs
) Handles Me.FormClosing
        If FrmLogin.CBformState.Text = ""
Then
            Me.Dispose()
            FrmLogin.Show()

FrmLogin.TxtAdminUserName.Text = ""

FrmLogin.TxtAdminPassword.Text = ""
            FrmLogin.txtUserName.Text =
""
            FrmLogin.txtPassword.Text =
""
            Return
        End If

        Try
            Dim cn As New
OleDbConnection("Provider=Microsoft.ACE.O
LEDB.12.0;Data Source=" &
My.Application.Info.DirectoryPath.ToStrin
g() & "\BackUp\testing.Accdb;Persist
Security Info=False;")
            If cn.State =
ConnectionState.Open Then
                cn.Close()
            End If
            cn.Open()

            Dim dr1 As OleDbDataReader
            Dim com As New OleDbCommand


            If FrmLogin.CBformState.Text
= "User" Then
                com.CommandText = "select
[UserID],[UserName] from userinfo where
UserID = '" & FrmLogin.txtUserName.Text &
"'"
                com.Connection = cn
                dr1 = com.ExecuteReader
                If dr1.Read Then
                    Dim intResponse As
Integer
                    intResponse =
MessageBox.Show("Do you really want to
logout?", "Response",
MessageBoxButtons.YesNo,
MessageBoxIcon.Question)
                    If intResponse =
MsgBoxResult.No Then
                        e.Cancel = True
                        ' Exit Sub
                    Else
                        MsgBox("Thank
you!!  '" &
UCase(dr1("UserName")).ToString() & "'" &
vbNewLine & "" & vbNewLine & " God Bless
U...", MsgBoxStyle.Information, "God
Bless...")
                        cn.Close()
                        Me.Dispose()
                        FrmLogin.Show()

FrmLogin.TxtAdminUserName.Text = ""

FrmLogin.TxtAdminPassword.Text = ""

FrmLogin.txtUserName.Text = ""

FrmLogin.txtPassword.Text = ""

                    End If

                End If
            Else
                If
FrmLogin.CBformState.Text = "Admin" Then
                    Dim intResponse As
Integer
                    intResponse =
MessageBox.Show("Do you really want to
logout? ", "Response",
MessageBoxButtons.YesNo,
MessageBoxIcon.Question)
                    If intResponse =
MsgBoxResult.No Then
                        e.Cancel = True
                        ' Exit Sub
                    Else
                        MsgBox("Thank
you", MsgBoxStyle.Information, "Thank
you")
                        cn.Close()
                        Me.Dispose()
                        FrmLogin.Show()

FrmLogin.TxtAdminUserName.Text = ""
FrmLogin.TxtAdminPassword.Text = ""
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
FrmLogin.txtUserName.Text = ""

FrmLogin.txtPassword.Text = ""

                End If
            End If
        End If
    Catch ex As Exception
        MsgBox(ex.Message(),
MsgBoxStyle.Critical, "Error...")
    End Try

End Sub

Private Sub FrmMain_Load(ByVal sender
As Object, ByVal e As System.EventArgs)
Handles Me.Load
    ToolStripStatusLabel1.Text =
"UserName :"
    ToolStripStatusLabel3.Text =
"Date :"
    DateStatusLabel.Text =
Format(Now, "dd/MM/yyy")
    ToolStripStatusLabel5.Text =
"Time :"
    Try
        If FrmLogin.CBformState.Text
= "User" Then
            Dim cn As New
OleDbConnection("Provider=Microsoft.ACE.O
LEDB.12.0;Data Source=" &
My.Application.Info.DirectoryPath.ToStrin
g() & "\BackUp\testing.Accdb;Persist
Security Info=False;")
            If cn.State =
ConnectionState.Open Then
                cn.Close()
            End If
            cn.Open()

            Dim dr1 As
OleDbDataReader
            Dim com As New
OleDbCommand
            com.CommandText = "select
[UserID],[UserName] from UserInfo where
UserID = @UName"

            ' UserName
            Dim UName As
OleDbParameter = New
OleDbParameter("@UName",
OleDbType.VarWChar, 30)
            UName.Value =
UCase(FrmLogin.txtUserName.Text.ToString(
))
            com.Parameters.Add(UName)
            com.Connection = cn

            dr1 = com.ExecuteReader
            If dr1.Read Then

UserNameStatusLabel.Text =
UCase(dr1("UserName").ToString())

ToolStripRegistartion.Visible = False

ToolStripSeparator4.Visible = False
                cn.Close()
                Return
            End If
        Else
            UserNameStatusLabel.Text
= "ADMIN"

ToolStripRegistartion.Visible = True
        End If

        Catch ex As Exception
            MsgBox(ex.Message(),
MsgBoxStyle.Critical, "Error...")
        End Try

    End Sub

    Private Sub PictureBox1_Click(ByVal
sender As System.Object, ByVal e As
System.EventArgs) Handles
PictureBox1.Click
        Me.Close()
    End Sub

    Private Sub
ToolStripRegistartion_Click(ByVal sender
As System.Object, ByVal e As
System.EventArgs) Handles
ToolStripRegistartion.Click
        Me.Dispose()
        Registration.Show()
    End Sub

    Private Sub
ToolStripUserInfo_Click(ByVal sender As
System.Object, ByVal e As
System.EventArgs) Handles
ToolStripUserInfo.Click
        Me.Dispose()
        UserInfo.Show()
    End Sub

    Private Sub
ToolStripLogout_Click_1(ByVal sender As
System.Object, ByVal e As
System.EventArgs) Handles
ToolStripLogout.Click
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        Me.Close()
    End Sub

    Private Sub
ToolStripButtonPM_Click(ByVal sender As
System.Object, ByVal e As
System.EventArgs) Handles
ToolStripButtonPM.Click
        Me.Dispose()
        FrmPM.Show()
    End Sub

    Private Sub
ToolStripButtonControl_Click(ByVal sender
As System.Object, ByVal e As
System.EventArgs) Handles
ToolStripButtonControl.Click
        Me.Dispose()
        FrmControl.Show()
    End Sub

    Private Sub
ToolStripMaintenance_Click(ByVal sender
As System.Object, ByVal e As
System.EventArgs) Handles
ToolStripMaintenance.Click
        Me.Dispose()
        FrmMaintenance.Show()
    End Sub
End Class
```

```
''FormControl
Imports System
Imports System.ComponentModel
Imports System.Threading
Imports System.IO.Ports
Imports System.Data.OleDb
Imports System.IO
Imports
System.Windows.Forms.DataVisualization.Ch
arting

Public Class FrmControl
    Dim _myPort As String = ""
    Dim myPort As Array
    Delegate Sub SetTextCallback(ByVal
[text] As String)

    Dim A As Integer
    Dim B As Integer
    Dim C As Integer
    Dim D As Integer
    Dim G As Integer
    Dim F As Integer

    Dim maxItem As Integer = 400
    Dim maxTimeItem As Integer = maxItem
+ 5

    Dim chartDataA As New List(Of
Integer)
    Dim chartDataB As New List(Of
Integer)
    Dim chartDataC As New List(Of
Integer)
    Dim chartDataD As New List(Of
Integer)
    Dim chartDataE As New List(Of
Integer)
    Dim chartDataF As New List(Of
Integer)
    Dim chartDataTime As New List(Of
String)

    Dim _A As Double, _B As Double
    Dim _C As Double, _D As Double
    Dim _E As Double, _F As Double
    Dim _G As Double, _H As Double
    Dim _I As Double, _J As Double
    Dim _K As Double, _L As Double

    Dim A1, B1, C1, D1, E1, F1 As Double
    Dim i, j, k, l, m, n As Integer
    Dim flGChnage As Double = False


    Private Sub
FrmControl_FormClosing(ByVal sender As
Object, ByVal e As
System.Windows.Forms.FormClosingEventArgs
) Handles Me.FormClosing
        Me.Dispose()
        FrmMain.Show()

    End Sub

    Private Sub FrmControl_Load(ByVal
sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load

        Timer1.Interval = 100

        Chart1.Series("Cylinder
A").XValueType = ChartValueType.Time
        Chart1.Series("Cylinder
B").XValueType = ChartValueType.Time
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        Chart1.Series("Cylinder
C").XValueType = ChartValueType.Time
        Chart1.Series("Cylinder
D").XValueType = ChartValueType.Time
        Chart1.Series("Cylinder
E").XValueType = ChartValueType.Time
        Chart1.Series("Cylinder
F").XValueType = ChartValueType.Time

        ToolStripStatusLabel1.Text =
"UserName :"
        ToolStripStatusLabel1.Text =
"Date :"
        DateStatusLabel.Text =
Format(Now, "dd/MM/yyy")
        ToolStripStatusLabel2.Text =
"Time :"

        For i As Integer = 0 To maxItem -
1
            chartDataA.Add(10)
            chartDataB.Add(8)
            chartDataC.Add(6)
            chartDataD.Add(4)
            chartDataE.Add(2)
            chartDataF.Add(0)
        Next

        For i As Integer = 0 To
maxTimeItem - 1
            Dim currentDateTime As
DateTime = DateTime.Now
            Dim format As String =
"HH:mm:ss"
            Dim test12324 As String =
currentDateTime.ToString(format)

            chartDataTime.Add(test12324)
        Next

        Me.OvalShape2.FillColor =
Color.Lime
        Me.OvalShape4.FillColor =
Color.Red
        Me.OvalShape6.FillColor =
Color.Lime
        Me.OvalShape8.FillColor =
Color.Red
        Me.OvalShape10.FillColor =
Color.Lime
        Me.OvalShape12.FillColor =
Color.Red
        Me.OvalShape14.FillColor =
Color.Lime
        Me.OvalShape16.FillColor =
Color.Red
        Me.OvalShape18.FillColor =
Color.Lime
        Me.OvalShape20.FillColor =
Color.Red
        Me.OvalShape22.FillColor =
Color.Lime
        Me.OvalShape24.FillColor =
Color.Red

        Me.ledAlarm1.FillColor =
Color.Lime
        Me.ledAlarm2.FillColor =
Color.Lime
        Me.ledAlarm3.FillColor =
Color.Lime
        Me.ledAlarm4.FillColor =
Color.Lime
        Me.ledAlarm5.FillColor =
Color.Lime
        Me.ledAlarm6.FillColor =
Color.Lime

        PictureBox1.Image =
My.Resources.Cylinder_in
        PictureBox2.Image =
My.Resources.Cylinder_in
        PictureBox3.Image =
My.Resources.Cylinder_in
        PictureBox4.Image =
My.Resources.Cylinder_in
        PictureBox5.Image =
My.Resources.Cylinder_in
        PictureBox6.Image =
My.Resources.Cylinder_in

        'Connect
        SerialPort1.PortName = "COM4"
        SerialPort1.BaudRate = 9600
        SerialPort1.Parity =
IO.Ports.Parity.Even
        SerialPort1.StopBits =
IO.Ports.StopBits.Two
        SerialPort1.DataBits = 7
        SerialPort1.Open()

        Dim cn As New
OleDbConnection("Provider=Microsoft.ACE.O
LEDB.12.0;Data Source=" &
My.Application.Info.DirectoryPath.ToStrin
g() & "\BackUp\testing.Accdb;Persist
Security Info=False;")
        If cn.State =
ConnectionState.Open Then
            cn.Close()
        End If
        cn.Open()

        Dim dr1 As OleDbDataReader
        Dim com1 As New OleDbCommand
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        com1.CommandText = "select
[CylinderA],[CylinderB],[CylinderC],[Cyli
nderD],[CylinderE],[CylinderF] from
Table1 where number = 1"
        com1.Connection = cn
        dr1 = com1.ExecuteReader
        If dr1.Read Then
            A1 = "" &
UCase(dr1(0)).ToString() & ""
            B1 = "" &
UCase(dr1(1)).ToString() & ""
            C1 = "" &
UCase(dr1(2)).ToString() & ""
            D1 = "" &
UCase(dr1(3)).ToString() & ""
            E1 = "" &
UCase(dr1(4)).ToString() & ""
            F1 = "" &
UCase(dr1(5)).ToString() & ""

        End If
        cn.Close()
        dr1.Close()
        Return



    End Sub

    Private Sub
SerialPort1_DataReceived(ByVal sender As
Object, ByVal e As
System.IO.Ports.SerialDataReceivedEventAr
gs) Handles SerialPort1.DataReceived

ReceivedText(SerialPort1.ReadExisting())
    End Sub

    Private Sub ReceivedText(ByVal [text]
As String)

        If Me.rtbReceived.InvokeRequired
Then
            Dim x As New
SetTextCallback(AddressOf ReceivedText)
            Me.Invoke(x, New Object()
{(text)})
        Else
            Me.rtbReceived.Text &= [text]
            Me.RichTextBox1.Text &=
Mid(rtbReceived.Text, 8, 4)
        End If
    End Sub

    Public Function FcsCal(ByVal strText
As String) As String
        Dim fcstemp As String
        Dim fcs As Byte = 0
        Dim i As Byte
        For i = 0 To
(CByte(strText.Length) - 1)
            fcs = fcs Xor
Val(Asc(strText(i)))
        Next
        fcstemp = CStr(Hex(fcs)) + "*"
        Return fcstemp
    End Function

    Private Sub ButtonStart_Click(ByVal
sender As System.Object, ByVal e As
System.EventArgs) Handles
ButtonStart.Click
        If RadioButtonPattern1.Checked
Then

SerialPort1.Write("@00WR00130001" &
FcsCal("@00WR00130001") & vbCr)
            Timer1.Start()
        ElseIf
RadioButtonPattern2.Checked Then

SerialPort1.Write("@00WR00130002" &
FcsCal("@00WR00130002") & vbCr)
            Timer1.Start()
        ElseIf
RadioButtonPattern3.Checked Then

SerialPort1.Write("@00WR00130004" &
FcsCal("@00WR00130004") & vbCr)
            Timer1.Start()
        ElseIf
RadioButtonPattern4.Checked Then

SerialPort1.Write("@00WR00130008" &
FcsCal("@00WR00130008") & vbCr)
            Timer1.Start()
        ElseIf
RadioButtonPattern5.Checked Then

SerialPort1.Write("@00WR00130010" &
FcsCal("@00WR00130010") & vbCr)
            Timer1.Start()
        End If

        'Alarm A
        If A1 > 5 Then
            Me.ledAlarm1.FillColor =
Color.Yellow
        End If
        If A1 > 10 Then
            Me.ledAlarm1.FillColor =
Color.Red
        End If
        'Alarm B
        If B1 > 5 Then
```

เอกสารนี้จัดทำขึ้นไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            Me.ledAlarm2.FillColor =
Color.Yellow
        End If
        If B1 > 10 Then
            Me.ledAlarm2.FillColor =
Color.Red
        End If
        'Alarm C
        If C1 > 5 Then
            Me.ledAlarm3.FillColor =
Color.Yellow
        End If
        If C1 > 10 Then
            Me.ledAlarm3.FillColor =
Color.Red
        End If
        'Alarm D
        If D1 > 5 Then
            Me.ledAlarm4.FillColor =
Color.Yellow
        End If
        If D1 > 10 Then
            Me.ledAlarm4.FillColor =
Color.Red
        End If
        'Alarm E
        If E1 > 5 Then
            Me.ledAlarm5.FillColor =
Color.Yellow
        End If
        If E1 > 10 Then
            Me.ledAlarm5.FillColor =
Color.Red
        End If
        'Alarm F
        If F1 > 5 Then
            Me.ledAlarm6.FillColor =
Color.Yellow
        End If
        If F1 > 10 Then
            Me.ledAlarm6.FillColor =
Color.Red
        End If
    End Sub

    Private Sub ButtonStop_Click(ByVal
sender As System.Object, ByVal e As
System.EventArgs) Handles
ButtonStop.Click
        SerialPort1.Write("@00WR00130020"
& FcsCal("@00WR00130020") & vbCr)

        Chart1.Series("Cylinder
A").Points.Clear()
        Chart1.Series("Cylinder
B").Points.Clear()
        Chart1.Series("Cylinder
C").Points.Clear()
        Chart1.Series("Cylinder
D").Points.Clear()
        Chart1.Series("Cylinder
E").Points.Clear()
        Chart1.Series("Cylinder
F").Points.Clear()

        Timer1.Stop()
        Try

            Dim cn As New
OleDbConnection("Provider=Microsoft.ACE.O
LEDB.12.0;Data Source=" &
My.Application.Info.DirectoryPath.ToStrin
g() & "\BackUp\testing.Accdb;Persist
Security Info=False;")
            If cn.State =
ConnectionState.Open Then
                cn.Close()
            End If
            cn.Open()

            Dim sSQL As String = "insert
into
Cylinder(DateofCylinder,CylinderA,Cylinde
rB,CylinderC,CylinderD,CylinderE,Cylinder
F) values(@d1,@d2,@d3,@d4,@d5,@d6,@d7)"
            Dim cmd As OleDbCommand = New
OleDbCommand(sSQL, cn)

            ' DateofCylinder
            Dim DateofCylinder As
OleDbParameter = New
OleDbParameter("@d1", OleDbType.VarWChar,
15)
            DateofCylinder.Value =
DateStatusLabel.Text.ToString()

cmd.Parameters.Add(DateofCylinder)

            ' CylinderA
            Dim CylinderA As
OleDbParameter = New
OleDbParameter("@d2", OleDbType.VarWChar,
15)
            CylinderA.Value =
LabelA.Text.ToString()
            cmd.Parameters.Add(CylinderA)

            ' CylinderB
            Dim CylinderB As
OleDbParameter = New
OleDbParameter("@d3", OleDbType.VarWChar,
15)
            CylinderB.Value =
LabelB.Text.ToString()
            cmd.Parameters.Add(CylinderB)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            ' CylinderC
            Dim CylinderC As
OleDbParameter = New
OleDbParameter("@d4", OleDbType.VarWChar,
15)
            CylinderC.Value =
LabelC.Text.ToString()
            cmd.Parameters.Add(CylinderC)

            ' CylinderD
            Dim CylinderD As
OleDbParameter = New
OleDbParameter("@d5", OleDbType.VarWChar,
15)
            CylinderD.Value =
LabelD.Text.ToString()
            cmd.Parameters.Add(CylinderD)


            ' CylinderE
            Dim CylinderE As
OleDbParameter = New
OleDbParameter("@d6", OleDbType.VarWChar,
15)
            CylinderE.Value =
LabelG.Text.ToString()
            cmd.Parameters.Add(CylinderE)

            ' CylinderF
            Dim CylinderF As
OleDbParameter = New
OleDbParameter("@d6", OleDbType.VarWChar,
15)
            CylinderF.Value =
LabelF.Text.ToString()
            cmd.Parameters.Add(CylinderF)

            If cmd.ExecuteNonQuery() Then
                cn.Close()
                MsgBox("Add To Database
successfully... ",
MsgBoxStyle.Information, "Record Saved")

            Else
                MsgBox("Add item
failed... ", MsgBoxStyle.Critical,
"failed")
                Return
            End If


        Catch ex As Exception

MessageBox.Show(ex.Message.ToString(),
"Data Error")
        Exit Sub
    End Try
```

```
    Try
        Dim cn As New
OleDbConnection("Provider=Microsoft.ACE.O
LEDB.12.0;Data Source=" &
My.Application.Info.DirectoryPath.ToStrin
g() & "\BackUp\testing.Accdb;Persist
Security Info=False;")
        If cn.State =
ConnectionState.Open Then
            cn.Close()
        End If
        cn.Open()
        Dim dr1 As OleDbDataReader
        Dim com As New OleDbCommand
        com.CommandText = "select
[number] from Table1 where number =
@Number"


        ' Number
        Dim Number As OleDbParameter
= New OleDbParameter("@Number",
OleDbType.VarWChar, 20)
        Number.Value = 1
        com.Parameters.Add(Number)
        com.Connection = cn

        dr1 = com.ExecuteReader
        If dr1.Read Then
            If UCase(dr1("number")) =
1 Then


                Dim sSQL As String =
"Update Table1 Set
CylinderA=@d11,CylinderB=@d12,CylinderC=@
d13,CylinderD=@d14,CylinderE=@15,Cylinder
F=@16 where number = 1"
                Dim cmd As
OleDbCommand = New OleDbCommand(sSQL, cn)
                Dim A2 As Double =
LabelA.Text

                Dim B2 As Double =
LabelB.Text

                Dim C2 As Double =
LabelC.Text

                Dim D2 As Double =
LabelD.Text

                Dim E2 As Double =
LabelG.Text

                Dim F2 As Double =
LabelF.Text


                ' CylinderA
                Dim AA As
OleDbParameter = New
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
OleDbParameter("@d11",
OleDbType.VarWChar, 15)
                    AA.Value = A2 + A1

cmd.Parameters.Add(AA)
                    'CylinderB
                    Dim BB As
OleDbParameter = New
OleDbParameter("@d12",
OleDbType.VarWChar, 15)
                    BB.Value = B2 + B1

cmd.Parameters.Add(BB)
                    'CylinderC
                    Dim CC As
OleDbParameter = New
OleDbParameter("@d13",
OleDbType.VarWChar, 15)
                    CC.Value = C2 + C1

cmd.Parameters.Add(CC)
                    'CylinderD
                    Dim DD As
OleDbParameter = New
OleDbParameter("@d14",
OleDbType.VarWChar, 15)
                    DD.Value = D2 + D1

cmd.Parameters.Add(DD)
                    'CylinderE
                    Dim EE As
OleDbParameter = New
OleDbParameter("@d15",
OleDbType.VarWChar, 15)
                    EE.Value = E2 + E1

cmd.Parameters.Add(EE)
                    'CylinderF
                    Dim FF As
OleDbParameter = New
OleDbParameter("@d16",
OleDbType.VarWChar, 15)
                    FF.Value = F2 + F1

cmd.Parameters.Add(FF)

                    If
cmd.ExecuteNonQuery() Then
                        cn.Close()
                        'MsgBox("Your
info updated successfully... ",
MsgBoxStyle.Information, "Record
Updated")

                        'Else
                        '    MsgBox("Your
info modification failed ",
MsgBoxStyle.Critical, "Modificatio
Failed")
                        '    Return
                    End If

                End If

            End If
        Catch ex As Exception
            MsgBox(ex.Message(),
MsgBoxStyle.Critical, "Error")
            Exit Sub
        End Try
    End Sub

    Private Sub Timer1_Tick(ByVal sender
As System.Object, ByVal e As
System.EventArgs) Handles Timer1.Tick
        SerialPort1.Write("@00RR00000001"
& FcsCal("@00RR00000001") & vbCr)
        rtbReceived.Text = ""

    End Sub

    Private Sub
RichTextBox1_TextChanged(ByVal sender As
System.Object, ByVal e As
System.EventArgs) Handles
RichTextBox1.TextChanged
        Dim txt As String =
Mid(rtbReceived.Text, 8, 4)
        RichTextBox1.Text = txt

        Dim currentDateTime As DateTime =
DateTime.Now
        Dim format As String = "HH:mm:ss"
        Dim test12324 As String =
currentDateTime.ToString(format)

        chartDataTime.Add(test12324)
        If chartDataTime.Count >
maxTimeItem Then
            chartDataTime.RemoveAt(0)
        End If

        'Pattern1
        If RadioButtonPattern1.Checked
Then
            If Mid(rtbReceived.Text, 8,
4) = "1554" Then
                flGChnage = True

                chartDataA.Add(10)
                chartDataB.Add(8)
                chartDataC.Add(6)
                chartDataD.Add(4)
                chartDataE.Add(2)
                chartDataF.Add(0)
```

เอกสารนี้สงวนลิขสิทธิ์ ไม่อนุญาตให้ใช้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                removeNode()

                'Chart1.Series("Cylinder
A").Points.AddXY(test12324, 10)
                'Chart1.Series("Cylinder
B").Points.AddXY(test12324, 8)
                'Chart1.Series("Cylinder
C").Points.AddXY(test12324, 6)
                'Chart1.Series("Cylinder
D").Points.AddXY(test12324, 4)
                'Chart1.Series("Cylinder
E").Points.AddXY(test12324, 2)
                'Chart1.Series("Cylinder
F").Points.AddXY(test12324, 0)
            End If

            If Mid(rtbReceived.Text, 8,
4) = "1558" Then
                flGChnage = True

                chartDataA.Add(11)
                chartDataB.Add(8)
                chartDataC.Add(6)
                chartDataD.Add(4)
                chartDataE.Add(2)
                chartDataF.Add(0)
                removeNode()

                'Chart1.Series("Cylinder
A").Points.AddXY(test12324, 11)
                'Chart1.Series("Cylinder
B").Points.AddXY(test12324, 8)
                'Chart1.Series("Cylinder
C").Points.AddXY(test12324, 6)
                'Chart1.Series("Cylinder
D").Points.AddXY(test12324, 4)
                'Chart1.Series("Cylinder
E").Points.AddXY(test12324, 2)
                'Chart1.Series("Cylinder
F").Points.AddXY(test12324, 0)

                PictureBox1.Image =
My.Resources.Cylinder_out

                Me.OvalShape2.FillColor =
Color.Red
                Me.OvalShape4.FillColor =
Color.Lime

            End If
            If (OvalShape4.FillColor =
Color.Lime) And (i = 1) Then
                A = A + 1
                LabelA.Text = A
                i = 0
            End If
            If Mid(rtbReceived.Text, 8,
4) = "1554" Then
                PictureBox1.Image =
My.Resources.Cylinder_in

                flGChnage = True
                chartDataA.Add(10)
                removeNode()
                'Chart1.Series("Cylinder
A").Points.AddXY(test12324, 10)

                Me.OvalShape2.FillColor =
Color.Lime
                Me.OvalShape4.FillColor =
Color.Red
                i = 1
            End If
            If Mid(rtbReceived.Text, 8,
4) = "1564" Then
                PictureBox2.Image =
My.Resources.Cylinder_out

                flGChnage = True

                chartDataA.Add(10)
                chartDataB.Add(9)
                chartDataC.Add(6)
                chartDataD.Add(4)
                chartDataE.Add(2)
                chartDataF.Add(0)
                removeNode()

                'Chart1.Series("Cylinder
A").Points.AddXY(test12324, 10)
                'Chart1.Series("Cylinder
B").Points.AddXY(test12324, 9)
                'Chart1.Series("Cylinder
C").Points.AddXY(test12324, 6)
                'Chart1.Series("Cylinder
D").Points.AddXY(test12324, 4)
                'Chart1.Series("Cylinder
E").Points.AddXY(test12324, 2)
                'Chart1.Series("Cylinder
F").Points.AddXY(test12324, 0)

                Me.OvalShape6.FillColor =
Color.Red
                Me.OvalShape8.FillColor =
Color.Lime

            End If
            If (OvalShape8.FillColor =
Color.Lime) And (j = 1) Then
                B = B + 1
                LabelB.Text = B
                j = 0
            End If
            If Mid(rtbReceived.Text, 8,
4) = "1554" Then
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                PictureBox2.Image =
My.Resources.Cylinder_in

                flGChnage = True
                chartDataB.Add(8)
                removeNode()
                'Chart1.Series("Cylinder
B").Points.AddXY(test12324, 8)

                Me.OvalShape6.FillColor =
Color.Lime
                Me.OvalShape8.FillColor =
Color.Red
                j = 1
            End If
            If Mid(rtbReceived.Text, 8,
4) = "1594" Then
                PictureBox3.Image =
My.Resources.Cylinder_out

                flGChnage = True

                chartDataA.Add(10)
                chartDataB.Add(8)
                chartDataC.Add(7)
                chartDataD.Add(4)
                chartDataE.Add(2)
                chartDataF.Add(0)
                removeNode()

                'Chart1.Series("Cylinder
A").Points.AddXY(test12324, 10)
                'Chart1.Series("Cylinder
B").Points.AddXY(test12324, 8)
                'Chart1.Series("Cylinder
C").Points.AddXY(test12324, 7)
                'Chart1.Series("Cylinder
D").Points.AddXY(test12324, 4)
                'Chart1.Series("Cylinder
E").Points.AddXY(test12324, 2)
                'Chart1.Series("Cylinder
F").Points.AddXY(test12324, 0)

                Me.OvalShape10.FillColor
= Color.Red
                Me.OvalShape12.FillColor
= Color.Lime

            End If
            If (OvalShape12.FillColor =
Color.Lime) And (k = 1) Then
                C = C + 1
                LabelC.Text = C
                k = 0
            End If
            If Mid(rtbReceived.Text, 8,
4) = "1554" Then
```

```
                PictureBox3.Image =
My.Resources.Cylinder_in

                flGChnage = True
                chartDataC.Add(6)
                removeNode()
                'Chart1.Series("Cylinder
C").Points.AddXY(test12324, 6)

                Me.OvalShape10.FillColor
= Color.Lime
                Me.OvalShape12.FillColor
= Color.Red
                k = 1
            End If
            If Mid(rtbReceived.Text, 8,
4) = "1654" Then
                PictureBox4.Image =
My.Resources.Cylinder_out

                flGChnage = True

                chartDataA.Add(10)
                chartDataB.Add(8)
                chartDataC.Add(6)
                chartDataD.Add(5)
                chartDataE.Add(2)
                chartDataF.Add(0)
                removeNode()

                'Chart1.Series("Cylinder
A").Points.AddXY(test12324, 10)
                'Chart1.Series("Cylinder
B").Points.AddXY(test12324, 8)
                'Chart1.Series("Cylinder
C").Points.AddXY(test12324, 6)
                'Chart1.Series("Cylinder
D").Points.AddXY(test12324, 5)
                'Chart1.Series("Cylinder
E").Points.AddXY(test12324, 2)
                'Chart1.Series("Cylinder
F").Points.AddXY(test12324, 0)

                Me.OvalShape14.FillColor
= Color.Red
                Me.OvalShape16.FillColor
= Color.Lime

            End If
            If (OvalShape16.FillColor =
Color.Lime) And (l = 1) Then
                D = D + 1
                LabelD.Text = D
                l = 0
            End If
            If Mid(rtbReceived.Text, 8,
4) = "1554" Then
```

เอกสารนี้มีการสงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                PictureBox4.Image =
My.Resources.Cylinder_in

                flGChnage = True
                chartDataD.Add(4)
                removeNode()
                'Chart1.Series("Cylinder
D").Points.AddXY(test12324, 4)

                Me.OvalShape14.FillColor
= Color.Lime
                Me.OvalShape16.FillColor
= Color.Red
                l = 1
            End If
            If Mid(rtbReceived.Text, 8,
4) = "1954" Then
                PictureBox5.Image =
My.Resources.Cylinder_out

                flGChnage = True

                chartDataA.Add(10)
                chartDataB.Add(8)
                chartDataC.Add(6)
                chartDataD.Add(4)
                chartDataE.Add(3)
                chartDataF.Add(0)
                removeNode()

                'Chart1.Series("Cylinder
A").Points.AddXY(test12324, 10)
                'Chart1.Series("Cylinder
B").Points.AddXY(test12324, 8)
                'Chart1.Series("Cylinder
C").Points.AddXY(test12324, 6)
                'Chart1.Series("Cylinder
D").Points.AddXY(test12324, 4)
                'Chart1.Series("Cylinder
E").Points.AddXY(test12324, 3)
                'Chart1.Series("Cylinder
F").Points.AddXY(test12324, 0)

                Me.OvalShape18.FillColor
= Color.Red
                Me.OvalShape20.FillColor
= Color.Lime

            End If
            If (OvalShape20.FillColor =
Color.Lime) And (m = 1) Then
                G = G + 1
                LabelG.Text = G
                m = 0
            End If
            If Mid(rtbReceived.Text, 8,
4) = "1554" Then
```

```
                PictureBox5.Image =
My.Resources.Cylinder_in

                flGChnage = True
                chartDataE.Add(2)
                removeNode()
                'Chart1.Series("Cylinder
E").Points.AddXY(test12324, 2)

                Me.OvalShape18.FillColor
= Color.Lime
                Me.OvalShape20.FillColor
= Color.Red
                m = 1
            End If
            If Mid(rtbReceived.Text, 8,
4) = "2554" Then
                PictureBox6.Image =
My.Resources.Cylinder_out

                flGChnage = True

                chartDataA.Add(10)
                chartDataB.Add(8)
                chartDataC.Add(6)
                chartDataD.Add(4)
                chartDataE.Add(2)
                chartDataF.Add(1)
                removeNode()

                'Chart1.Series("Cylinder
A").Points.AddXY(test12324, 10)
                'Chart1.Series("Cylinder
B").Points.AddXY(test12324, 8)
                'Chart1.Series("Cylinder
C").Points.AddXY(test12324, 6)
                'Chart1.Series("Cylinder
D").Points.AddXY(test12324, 4)
                'Chart1.Series("Cylinder
E").Points.AddXY(test12324, 2)
                'Chart1.Series("Cylinder
F").Points.AddXY(test12324, 1)

                Me.OvalShape22.FillColor
= Color.Red
                Me.OvalShape24.FillColor
= Color.Lime

            End If
            If (OvalShape24.FillColor =
Color.Lime) And (n = 1) Then
                F = F + 1
                LabelF.Text = F
                n = 0
            End If
            If Mid(rtbReceived.Text, 8,
4) = "1554" Then
```

เอกสารนี้สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                PictureBox6.Image =
My.Resources.Cylinder_in
                flGChnage = True
                chartDataF.Add(0)
                removeNode()
                'Chart1.Series("Cylinder
F").Points.AddXY(test12324, 0)

                Me.OvalShape22.FillColor
= Color.Lime
                Me.OvalShape24.FillColor
= Color.Red
                n = 1
            End If
        End If


        'Pattern2
        If RadioButtonPattern2.Checked
Then

            If Mid(rtbReceived.Text, 8,
4) = "1554" Then
                PictureBox1.Image =
My.Resources.Cylinder_in
                PictureBox2.Image =
My.Resources.Cylinder_in
                PictureBox3.Image =
My.Resources.Cylinder_in
                PictureBox4.Image =
My.Resources.Cylinder_in
                PictureBox5.Image =
My.Resources.Cylinder_in
                PictureBox6.Image =
My.Resources.Cylinder_in

                flGChnage = True
                chartDataA.Add(10)
                chartDataB.Add(8)
                chartDataC.Add(6)
                chartDataD.Add(4)
                chartDataE.Add(2)
                chartDataF.Add(0)
                removeNode()

                'Chart1.Series("Cylinder
A").Points.AddXY(_A, 10)
                'Chart1.Series("Cylinder
B").Points.AddXY(_C, 8)
                'Chart1.Series("Cylinder
C").Points.AddXY(_F, 6)
                'Chart1.Series("Cylinder
D").Points.AddXY(_G, 4)
                'Chart1.Series("Cylinder
E").Points.AddXY(_I, 2)
                'Chart1.Series("Cylinder
F").Points.AddXY(_K, 0)

                Me.OvalShape2.FillColor =
Color.Lime
                Me.OvalShape4.FillColor =
Color.Red
                Me.OvalShape6.FillColor =
Color.Lime
                Me.OvalShape8.FillColor =
Color.Red
                Me.OvalShape10.FillColor
= Color.Lime
                Me.OvalShape12.FillColor
= Color.Red
                Me.OvalShape14.FillColor
= Color.Lime
                Me.OvalShape16.FillColor
= Color.Red
                Me.OvalShape18.FillColor
= Color.Lime
                Me.OvalShape20.FillColor
= Color.Red
                Me.OvalShape22.FillColor
= Color.Lime
                Me.OvalShape24.FillColor
= Color.Red

                i = 1
                j = 1
                k = 1
                l = 1
                m = 1
                n = 1
            End If

            If Mid(rtbReceived.Text, 8,
4) = "1558" Then
                PictureBox1.Image =
My.Resources.Cylinder_out
                PictureBox2.Image =
My.Resources.Cylinder_in
                PictureBox3.Image =
My.Resources.Cylinder_in
                PictureBox4.Image =
My.Resources.Cylinder_in
                PictureBox5.Image =
My.Resources.Cylinder_in
                PictureBox6.Image =
My.Resources.Cylinder_in

                flGChnage = True
                chartDataA.Add(11)
                chartDataB.Add(8)
                chartDataC.Add(6)
                chartDataD.Add(4)
                chartDataE.Add(2)
                chartDataF.Add(0)
                removeNode()
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                'Chart1.Series("Cylinder
A").Points.AddXY(_A, 11)
                'Chart1.Series("Cylinder
B").Points.AddXY(_C, 8)
                'Chart1.Series("Cylinder
C").Points.AddXY(_F, 6)
                'Chart1.Series("Cylinder
D").Points.AddXY(_G, 4)
                'Chart1.Series("Cylinder
E").Points.AddXY(_I, 2)
                'Chart1.Series("Cylinder
F").Points.AddXY(_K, 0)

                Me.OvalShape2.FillColor =
Color.Red
                Me.OvalShape4.FillColor =
Color.Lime
                Me.OvalShape6.FillColor =
Color.Lime
                Me.OvalShape8.FillColor =
Color.Red
                Me.OvalShape10.FillColor
= Color.Lime
                Me.OvalShape12.FillColor
= Color.Red
                Me.OvalShape14.FillColor
= Color.Lime
                Me.OvalShape16.FillColor
= Color.Red
                Me.OvalShape18.FillColor
= Color.Lime
                Me.OvalShape20.FillColor
= Color.Red
                Me.OvalShape22.FillColor
= Color.Lime
                Me.OvalShape24.FillColor
= Color.Red
            End If
            If (OvalShape4.FillColor =
Color.Lime) And (i = 1) Then
                A = A + 1
                LabelA.Text = A
                i = 0
            End If

            If Mid(rtbReceived.Text, 8,
4) = "1568" Then
                PictureBox1.Image =
My.Resources.Cylinder_out
                PictureBox2.Image =
My.Resources.Cylinder_out
                PictureBox3.Image =
My.Resources.Cylinder_in
                PictureBox4.Image =
My.Resources.Cylinder_in
                PictureBox5.Image =
My.Resources.Cylinder_in
                PictureBox6.Image =
My.Resources.Cylinder_in

                flGChnage = True
                chartDataA.Add(11)
                chartDataB.Add(9)
                chartDataC.Add(6)
                chartDataD.Add(4)
                chartDataE.Add(2)
                chartDataF.Add(0)
                removeNode()

                'Chart1.Series("Cylinder
A").Points.AddXY(_A, 11)
                'Chart1.Series("Cylinder
B").Points.AddXY(_C, 9)
                'Chart1.Series("Cylinder
C").Points.AddXY(_F, 6)
                'Chart1.Series("Cylinder
D").Points.AddXY(_G, 4)
                'Chart1.Series("Cylinder
E").Points.AddXY(_I, 2)
                'Chart1.Series("Cylinder
F").Points.AddXY(_K, 0)

                Me.OvalShape2.FillColor =
Color.Red
                Me.OvalShape4.FillColor =
Color.Lime
                Me.OvalShape6.FillColor =
Color.Red
                Me.OvalShape8.FillColor =
Color.Lime
                Me.OvalShape10.FillColor
= Color.Lime
                Me.OvalShape12.FillColor
= Color.Red
                Me.OvalShape14.FillColor
= Color.Lime
                Me.OvalShape16.FillColor
= Color.Red
                Me.OvalShape18.FillColor
= Color.Lime
                Me.OvalShape20.FillColor
= Color.Red
                Me.OvalShape22.FillColor
= Color.Lime
                Me.OvalShape24.FillColor
= Color.Red
            End If
            If (OvalShape8.FillColor =
Color.Lime) And (j = 1) Then
                B = B + 1
                LabelB.Text = B
                j = 0
            End If
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            If Mid(rtbReceived.Text, 8,
4) = "15A8" Then
                PictureBox1.Image =
My.Resources.Cylinder_out
                PictureBox2.Image =
My.Resources.Cylinder_out
                PictureBox3.Image =
My.Resources.Cylinder_out

                PictureBox4.Image =
My.Resources.Cylinder_in
                PictureBox5.Image =
My.Resources.Cylinder_in
                PictureBox6.Image =
My.Resources.Cylinder_in

                flGChnage = True
                chartDataA.Add(11)
                chartDataB.Add(9)
                chartDataC.Add(7)
                chartDataD.Add(4)
                chartDataE.Add(2)
                chartDataF.Add(0)
                removeNode()

                'Chart1.Series("Cylinder
A").Points.AddXY(_A, 11)
                'Chart1.Series("Cylinder
B").Points.AddXY(_C, 9)
                'Chart1.Series("Cylinder
C").Points.AddXY(_F, 7)
                'Chart1.Series("Cylinder
D").Points.AddXY(_G, 4)
                'Chart1.Series("Cylinder
E").Points.AddXY(_I, 2)
                'Chart1.Series("Cylinder
F").Points.AddXY(_K, 0)

                Me.OvalShape2.FillColor =
Color.Red
                Me.OvalShape4.FillColor =
Color.Lime
                Me.OvalShape6.FillColor =
Color.Red
                Me.OvalShape8.FillColor =
Color.Lime
                Me.OvalShape10.FillColor
= Color.Red
                Me.OvalShape12.FillColor
= Color.Lime
                Me.OvalShape14.FillColor
= Color.Lime
                Me.OvalShape16.FillColor
= Color.Red
                Me.OvalShape18.FillColor
= Color.Lime
                Me.OvalShape20.FillColor
= Color.Red
                Me.OvalShape22.FillColor
= Color.Lime
                Me.OvalShape24.FillColor
= Color.Red
            End If
            If (OvalShape12.FillColor =
Color.Lime) And (k = 1) Then
                C = C + 1
                LabelC.Text = C
                k = 0
            End If
            If Mid(rtbReceived.Text, 8,
4) = "16A8" Then
                PictureBox1.Image =
My.Resources.Cylinder_out
                PictureBox2.Image =
My.Resources.Cylinder_out
                PictureBox3.Image =
My.Resources.Cylinder_out
                PictureBox4.Image =
My.Resources.Cylinder_out
                PictureBox5.Image =
My.Resources.Cylinder_in
                PictureBox6.Image =
My.Resources.Cylinder_in

                flGChnage = True
                chartDataA.Add(11)
                chartDataB.Add(9)
                chartDataC.Add(7)
                chartDataD.Add(5)
                chartDataE.Add(2)
                chartDataF.Add(0)
                removeNode()

                'Chart1.Series("Cylinder
A").Points.AddXY(_A, 11)
                'Chart1.Series("Cylinder
B").Points.AddXY(_C, 9)
                'Chart1.Series("Cylinder
C").Points.AddXY(_F, 7)
                'Chart1.Series("Cylinder
D").Points.AddXY(_G, 5)
                'Chart1.Series("Cylinder
E").Points.AddXY(_I, 2)
                'Chart1.Series("Cylinder
F").Points.AddXY(_K, 0)

                Me.OvalShape2.FillColor =
Color.Red
                Me.OvalShape4.FillColor =
Color.Lime
                Me.OvalShape6.FillColor =
Color.Red
                Me.OvalShape8.FillColor =
Color.Lime
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                Me.OvalShape10.FillColor
= Color.Red
                Me.OvalShape12.FillColor
= Color.Lime
                Me.OvalShape14.FillColor
= Color.Red
                Me.OvalShape16.FillColor
= Color.Lime
                Me.OvalShape18.FillColor
= Color.Lime
                Me.OvalShape20.FillColor
= Color.Red
                Me.OvalShape22.FillColor
= Color.Lime
                Me.OvalShape24.FillColor
= Color.Red
            End If
            If (OvalShape16.FillColor =
Color.Lime) And (l = 1) Then
                D = D + 1
                LabelD.Text = D
                l = 0
            End If
            If Mid(rtbReceived.Text, 8,
4) = "1AA8" Then
                PictureBox1.Image =
My.Resources.Cylinder_out
                PictureBox2.Image =
My.Resources.Cylinder_out
                PictureBox3.Image =
My.Resources.Cylinder_out
                PictureBox4.Image =
My.Resources.Cylinder_out
                PictureBox5.Image =
My.Resources.Cylinder_out
                PictureBox6.Image =
My.Resources.Cylinder_in

                flGChnage = True
                chartDataA.Add(11)
                chartDataB.Add(9)
                chartDataC.Add(7)
                chartDataD.Add(5)
                chartDataE.Add(3)
                chartDataF.Add(0)
                removeNode()

                'Chart1.Series("Cylinder
A").Points.AddXY(_A, 11)
                'Chart1.Series("Cylinder
B").Points.AddXY(_C, 9)
                'Chart1.Series("Cylinder
C").Points.AddXY(_F, 7)
                'Chart1.Series("Cylinder
D").Points.AddXY(_G, 5)
                'Chart1.Series("Cylinder
E").Points.AddXY(_I, 3)
                'Chart1.Series("Cylinder
F").Points.AddXY(_K, 0)

                Me.OvalShape2.FillColor =
Color.Red
                Me.OvalShape4.FillColor =
Color.Lime
                Me.OvalShape6.FillColor =
Color.Red
                Me.OvalShape8.FillColor =
Color.Lime
                Me.OvalShape10.FillColor
= Color.Red
                Me.OvalShape12.FillColor
= Color.Lime
                Me.OvalShape14.FillColor
= Color.Red
                Me.OvalShape16.FillColor
= Color.Lime
                Me.OvalShape18.FillColor
= Color.Red
                Me.OvalShape20.FillColor
= Color.Lime
                Me.OvalShape22.FillColor
= Color.Lime
                Me.OvalShape24.FillColor
= Color.Red

            End If
            If (OvalShape20.FillColor =
Color.Lime) And (m = 1) Then
                G = G + 1
                LabelG.Text = G
                m = 0
            End If
            If Mid(rtbReceived.Text, 8,
4) = "2AA8" Then
                PictureBox1.Image =
My.Resources.Cylinder_out
                PictureBox2.Image =
My.Resources.Cylinder_out
                PictureBox3.Image =
My.Resources.Cylinder_out
                PictureBox4.Image =
My.Resources.Cylinder_out
                PictureBox5.Image =
My.Resources.Cylinder_out
                PictureBox6.Image =
My.Resources.Cylinder_out

                flGChnage = True
                chartDataA.Add(11)
                chartDataB.Add(9)
                chartDataC.Add(7)
                chartDataD.Add(5)
                chartDataE.Add(3)
                chartDataF.Add(1)
                removeNode()
```

เอกสารนี้มีลิขสิทธิ์ สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                'Chart1.Series("Cylinder
A").Points.AddXY(_A, 11)
                'Chart1.Series("Cylinder
B").Points.AddXY(_C, 9)
                'Chart1.Series("Cylinder
C").Points.AddXY(_F, 7)
                'Chart1.Series("Cylinder
D").Points.AddXY(_G, 5)
                'Chart1.Series("Cylinder
E").Points.AddXY(_I, 3)
                'Chart1.Series("Cylinder
F").Points.AddXY(_K, 1)

                Me.OvalShape2.FillColor =
Color.Red
                Me.OvalShape4.FillColor =
Color.Lime
                Me.OvalShape6.FillColor =
Color.Red
                Me.OvalShape8.FillColor =
Color.Lime
                Me.OvalShape10.FillColor
= Color.Red
                Me.OvalShape12.FillColor
= Color.Lime
                Me.OvalShape14.FillColor
= Color.Red
                Me.OvalShape16.FillColor
= Color.Lime
                Me.OvalShape18.FillColor
= Color.Red
                Me.OvalShape20.FillColor
= Color.Lime
                Me.OvalShape22.FillColor
= Color.Red
                Me.OvalShape24.FillColor
= Color.Lime
            End If
            If (OvalShape24.FillColor =
Color.Lime) And (n = 1) Then
                F = F + 1
                LabelF.Text = F
                n = 0
            End If
        End If

        'Pattern3
        If RadioButtonPattern3.Checked
Then

            If Mid(rtbReceived.Text, 8,
4) = "1568" Then
                PictureBox1.Image =
My.Resources.Cylinder_out
                PictureBox2.Image =
My.Resources.Cylinder_out
                flGChnage = True
                chartDataA.Add(11)
                chartDataB.Add(9)
                chartDataC.Add(6)
                chartDataD.Add(4)
                chartDataE.Add(2)
                chartDataF.Add(0)
                removeNode()

                'Chart1.Series("Cylinder
A").Points.AddXY(_A, 11)
                'Chart1.Series("Cylinder
B").Points.AddXY(_C, 9)
                'Chart1.Series("Cylinder
C").Points.AddXY(_F, 6)
                'Chart1.Series("Cylinder
D").Points.AddXY(_G, 4)
                'Chart1.Series("Cylinder
E").Points.AddXY(_I, 2)
                'Chart1.Series("Cylinder
F").Points.AddXY(_K, 0)

                Me.OvalShape2.FillColor =
Color.Red
                Me.OvalShape4.FillColor =
Color.Lime
                Me.OvalShape6.FillColor =
Color.Red
                Me.OvalShape8.FillColor =
Color.Lime
                Me.OvalShape10.FillColor
= Color.Lime
                Me.OvalShape12.FillColor
= Color.Red
                Me.OvalShape14.FillColor
= Color.Lime
                Me.OvalShape16.FillColor
= Color.Red
                Me.OvalShape18.FillColor
= Color.Lime
                Me.OvalShape20.FillColor
= Color.Red
                Me.OvalShape22.FillColor
= Color.Lime
                Me.OvalShape24.FillColor
= Color.Red

            End If
            If (OvalShape4.FillColor =
Color.Lime And OvalShape8.FillColor =
Color.Lime) And (i And j = 1) Then
                A = A + 1
                B = B + 1
                LabelA.Text = A
                LabelB.Text = B
                i = 0
                j = 0
            End If
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            If Mid(rtbReceived.Text, 8,
4) = "1694" Then
                PictureBox3.Image =
My.Resources.Cylinder_out
                PictureBox4.Image =
My.Resources.Cylinder_out

                PictureBox1.Image =
My.Resources.Cylinder_in
                PictureBox2.Image =
My.Resources.Cylinder_in

                flGChnage = True
                chartDataA.Add(10)
                chartDataB.Add(8)
                chartDataC.Add(7)
                chartDataD.Add(5)
                chartDataE.Add(2)
                chartDataF.Add(0)
                removeNode()

                'Chart1.Series("Cylinder
A").Points.AddXY(_A, 10)
                'Chart1.Series("Cylinder
B").Points.AddXY(_C, 8)
                'Chart1.Series("Cylinder
C").Points.AddXY(_F, 7)
                'Chart1.Series("Cylinder
D").Points.AddXY(_G, 5)
                'Chart1.Series("Cylinder
E").Points.AddXY(_I, 2)
                'Chart1.Series("Cylinder
F").Points.AddXY(_K, 0)

                Me.OvalShape2.FillColor =
Color.Lime
                Me.OvalShape4.FillColor =
Color.Red
                Me.OvalShape6.FillColor =
Color.Lime
                Me.OvalShape8.FillColor =
Color.Red
                Me.OvalShape10.FillColor
= Color.Red
                Me.OvalShape12.FillColor
= Color.Lime
                Me.OvalShape14.FillColor
= Color.Red
                Me.OvalShape16.FillColor
= Color.Lime
                Me.OvalShape18.FillColor
= Color.Lime
                Me.OvalShape20.FillColor
= Color.Red
                Me.OvalShape22.FillColor
= Color.Lime
                Me.OvalShape24.FillColor
= Color.Red


            End If
            If (OvalShape12.FillColor =
Color.Lime And OvalShape16.FillColor =
Color.Lime) And (k And l = 1) Then
                C = C + 1
                D = D + 1
                LabelC.Text = C
                LabelD.Text = D
                k = 0
                l = 0
            End If
            If Mid(rtbReceived.Text, 8,
4) = "2954" Then
                PictureBox5.Image =
My.Resources.Cylinder_out
                PictureBox6.Image =
My.Resources.Cylinder_out

                PictureBox3.Image =
My.Resources.Cylinder_in
                PictureBox4.Image =
My.Resources.Cylinder_in

                flGChnage = True
                chartDataA.Add(10)
                chartDataB.Add(8)
                chartDataC.Add(6)
                chartDataD.Add(4)
                chartDataE.Add(3)
                chartDataF.Add(1)
                removeNode()

                'Chart1.Series("Cylinder
A").Points.AddXY(_A, 10)
                'Chart1.Series("Cylinder
B").Points.AddXY(_C, 8)
                'Chart1.Series("Cylinder
C").Points.AddXY(_F, 6)
                'Chart1.Series("Cylinder
D").Points.AddXY(_G, 4)
                'Chart1.Series("Cylinder
E").Points.AddXY(_I, 3)
                'Chart1.Series("Cylinder
F").Points.AddXY(_K, 1)

                Me.OvalShape2.FillColor =
Color.Lime
                Me.OvalShape4.FillColor =
Color.Red
                Me.OvalShape6.FillColor =
Color.Lime
                Me.OvalShape8.FillColor =
Color.Red
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                Me.OvalShape10.FillColor
= Color.Lime
                Me.OvalShape12.FillColor
= Color.Red
                Me.OvalShape14.FillColor
= Color.Lime
                Me.OvalShape16.FillColor
= Color.Red
                Me.OvalShape18.FillColor
= Color.Red
                Me.OvalShape20.FillColor
= Color.Lime
                Me.OvalShape22.FillColor
= Color.Red
                Me.OvalShape24.FillColor
= Color.Lime


            End If
            If (OvalShape20.FillColor =
Color.Lime And OvalShape24.FillColor =
Color.Lime) And (m And n = 1) Then
                G = G + 1
                F = F + 1
                LabelG.Text = G
                LabelF.Text = F
                m = 0
                n = 0
            End If
            If Mid(rtbReceived.Text, 8,
4) = "1554" Then
                PictureBox1.Image =
My.Resources.Cylinder_in
                PictureBox2.Image =
My.Resources.Cylinder_in
                PictureBox3.Image =
My.Resources.Cylinder_in
                PictureBox4.Image =
My.Resources.Cylinder_in
                PictureBox5.Image =
My.Resources.Cylinder_in
                PictureBox6.Image =
My.Resources.Cylinder_in

                flGChnage = True
                chartDataA.Add(10)
                chartDataB.Add(8)
                chartDataC.Add(6)
                chartDataD.Add(4)
                chartDataE.Add(2)
                chartDataF.Add(0)
                removeNode()

                'Chart1.Series("Cylinder
A").Points.AddXY(_A, 10)
                'Chart1.Series("Cylinder
B").Points.AddXY(_C, 8)
                'Chart1.Series("Cylinder
C").Points.AddXY(_F, 6)
                'Chart1.Series("Cylinder
D").Points.AddXY(_G, 4)
                'Chart1.Series("Cylinder
E").Points.AddXY(_I, 2)
                'Chart1.Series("Cylinder
F").Points.AddXY(_K, 0)

                Me.OvalShape2.FillColor =
Color.Lime
                Me.OvalShape4.FillColor =
Color.Red
                Me.OvalShape6.FillColor =
Color.Lime
                Me.OvalShape8.FillColor =
Color.Red
                Me.OvalShape10.FillColor
= Color.Lime
                Me.OvalShape12.FillColor
= Color.Red
                Me.OvalShape14.FillColor
= Color.Lime
                Me.OvalShape16.FillColor
= Color.Red
                Me.OvalShape18.FillColor
= Color.Lime
                Me.OvalShape20.FillColor
= Color.Red
                Me.OvalShape22.FillColor
= Color.Lime
                Me.OvalShape24.FillColor
= Color.Red

                i = 1
                j = 1
                k = 1
                l = 1
                m = 1
                n = 1
            End If
        End If

        'Pattern4
        If RadioButtonPattern4.Checked
Then

            If Mid(rtbReceived.Text, 8,
4) = "1558" Then
                PictureBox1.Image =
My.Resources.Cylinder_out

                flGChnage = True
                chartDataA.Add(11)
                chartDataB.Add(8)
                chartDataC.Add(6)
                chartDataD.Add(4)
                chartDataE.Add(2)
```

เอกสารนี้มีลิขสิทธิ์ สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                chartDataF.Add(0)
                removeNode()

                'Chart1.Series("Cylinder
A").Points.AddXY(_A, 11)
                'Chart1.Series("Cylinder
B").Points.AddXY(_C, 8)
                'Chart1.Series("Cylinder
C").Points.AddXY(_F, 6)
                'Chart1.Series("Cylinder
D").Points.AddXY(_G, 4)
                'Chart1.Series("Cylinder
E").Points.AddXY(_I, 2)
                'Chart1.Series("Cylinder
F").Points.AddXY(_K, 0)

                Me.OvalShape2.FillColor =
Color.Red
                Me.OvalShape4.FillColor =
Color.Lime
                Me.OvalShape6.FillColor =
Color.Lime
                Me.OvalShape8.FillColor =
Color.Red
                Me.OvalShape10.FillColor
= Color.Lime
                Me.OvalShape12.FillColor
= Color.Red
                Me.OvalShape14.FillColor
= Color.Lime
                Me.OvalShape16.FillColor
= Color.Red
                Me.OvalShape18.FillColor
= Color.Lime
                Me.OvalShape20.FillColor
= Color.Red
                Me.OvalShape22.FillColor
= Color.Lime
                Me.OvalShape24.FillColor
= Color.Red
            End If
            If (OvalShape4.FillColor =
Color.Lime) And (i = 1) Then
                A = A + 1
                LabelA.Text = A
                i = 0
            End If
            If Mid(rtbReceived.Text, 8,
4) = "1568" Then
                PictureBox1.Image =
My.Resources.Cylinder_out
                PictureBox2.Image =
My.Resources.Cylinder_out

                flGChnage = True
                chartDataA.Add(11)
                chartDataB.Add(9)
                chartDataC.Add(6)
                chartDataD.Add(4)
                chartDataE.Add(2)
                chartDataF.Add(0)
                removeNode()

                'Chart1.Series("Cylinder
A").Points.AddXY(_A, 11)
                'Chart1.Series("Cylinder
B").Points.AddXY(_C, 9)
                'Chart1.Series("Cylinder
C").Points.AddXY(_F, 6)
                'Chart1.Series("Cylinder
D").Points.AddXY(_G, 4)
                'Chart1.Series("Cylinder
E").Points.AddXY(_I, 2)
                'Chart1.Series("Cylinder
F").Points.AddXY(_K, 0)

                Me.OvalShape2.FillColor =
Color.Red
                Me.OvalShape4.FillColor =
Color.Lime
                Me.OvalShape6.FillColor =
Color.Red
                Me.OvalShape8.FillColor =
Color.Lime
                Me.OvalShape10.FillColor
= Color.Lime
                Me.OvalShape12.FillColor
= Color.Red
                Me.OvalShape14.FillColor
= Color.Lime
                Me.OvalShape16.FillColor
= Color.Red
                Me.OvalShape18.FillColor
= Color.Lime
                Me.OvalShape20.FillColor
= Color.Red
                Me.OvalShape22.FillColor
= Color.Lime
                Me.OvalShape24.FillColor
= Color.Red
            End If
            If (OvalShape8.FillColor =
Color.Lime) And (j = 1) Then
                B = B + 1
                LabelB.Text = B
                j = 0
            End If

            If Mid(rtbReceived.Text, 8,
4) = "15A8" Then
                PictureBox1.Image =
My.Resources.Cylinder_out
                PictureBox2.Image =
My.Resources.Cylinder_out
                PictureBox3.Image =
My.Resources.Cylinder_out
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                flGChnage = True
                chartDataA.Add(11)
                chartDataB.Add(9)
                chartDataC.Add(7)
                chartDataD.Add(4)
                chartDataE.Add(2)
                chartDataF.Add(0)
                removeNode()

                'Chart1.Series("Cylinder
A").Points.AddXY(_A, 11)
                'Chart1.Series("Cylinder
B").Points.AddXY(_C, 9)
                'Chart1.Series("Cylinder
C").Points.AddXY(_F, 7)
                'Chart1.Series("Cylinder
D").Points.AddXY(_G, 4)
                'Chart1.Series("Cylinder
E").Points.AddXY(_I, 2)
                'Chart1.Series("Cylinder
F").Points.AddXY(_K, 0)

                Me.OvalShape2.FillColor =
Color.Red
                Me.OvalShape4.FillColor =
Color.Lime
                Me.OvalShape6.FillColor =
Color.Red
                Me.OvalShape8.FillColor =
Color.Lime
                Me.OvalShape10.FillColor
= Color.Red
                Me.OvalShape12.FillColor
= Color.Lime
                Me.OvalShape14.FillColor
= Color.Lime
                Me.OvalShape16.FillColor
= Color.Red
                Me.OvalShape18.FillColor
= Color.Lime
                Me.OvalShape20.FillColor
= Color.Red
                Me.OvalShape22.FillColor
= Color.Lime
                Me.OvalShape24.FillColor
= Color.Red
            End If
            If (OvalShape12.FillColor =
Color.Lime) And (k = 1) Then
                C = C + 1
                LabelC.Text = C
                k = 0
            End If
            If Mid(rtbReceived.Text, 8,
4) = "1654" Then
                PictureBox4.Image =
My.Resources.Cylinder_out
                PictureBox1.Image =
My.Resources.Cylinder_in
                PictureBox2.Image =
My.Resources.Cylinder_in
                PictureBox3.Image =
My.Resources.Cylinder_in

                flGChnage = True
                chartDataA.Add(10)
                chartDataB.Add(8)
                chartDataC.Add(6)
                chartDataD.Add(5)
                chartDataE.Add(2)
                chartDataF.Add(0)
                removeNode()

                'Chart1.Series("Cylinder
A").Points.AddXY(_A, 10)
                'Chart1.Series("Cylinder
B").Points.AddXY(_C, 8)
                'Chart1.Series("Cylinder
C").Points.AddXY(_F, 6)
                'Chart1.Series("Cylinder
D").Points.AddXY(_G, 5)
                'Chart1.Series("Cylinder
E").Points.AddXY(_I, 2)
                'Chart1.Series("Cylinder
F").Points.AddXY(_K, 0)

                Me.OvalShape2.FillColor =
Color.Lime
                Me.OvalShape4.FillColor =
Color.Red
                Me.OvalShape6.FillColor =
Color.Lime
                Me.OvalShape8.FillColor =
Color.Red
                Me.OvalShape10.FillColor
= Color.Lime
                Me.OvalShape12.FillColor
= Color.Red
                Me.OvalShape14.FillColor
= Color.Red
                Me.OvalShape16.FillColor
= Color.Lime
                Me.OvalShape18.FillColor
= Color.Lime
                Me.OvalShape20.FillColor
= Color.Red
                Me.OvalShape22.FillColor
= Color.Lime
                Me.OvalShape24.FillColor
= Color.Red
            End If
            If (OvalShape16.FillColor =
Color.Lime) And (l = 1) Then
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                D = D + 1
                LabelD.Text = D
                l = 0
            End If
            If Mid(rtbReceived.Text, 8,
4) = "1A54" Then
                PictureBox4.Image =
My.Resources.Cylinder_out
                PictureBox5.Image =
My.Resources.Cylinder_out

                flGChnage = True
                chartDataA.Add(10)
                chartDataB.Add(8)
                chartDataC.Add(6)
                chartDataD.Add(5)
                chartDataE.Add(3)
                chartDataF.Add(0)
                removeNode()

                'Chart1.Series("Cylinder
A").Points.AddXY(_A, 10)
                'Chart1.Series("Cylinder
B").Points.AddXY(_C, 8)
                'Chart1.Series("Cylinder
C").Points.AddXY(_F, 6)
                'Chart1.Series("Cylinder
D").Points.AddXY(_G, 5)
                'Chart1.Series("Cylinder
E").Points.AddXY(_I, 3)
                'Chart1.Series("Cylinder
F").Points.AddXY(_K, 0)

                Me.OvalShape2.FillColor =
Color.Lime
                Me.OvalShape4.FillColor =
Color.Red
                Me.OvalShape6.FillColor =
Color.Lime
                Me.OvalShape8.FillColor =
Color.Red
                Me.OvalShape10.FillColor
= Color.Lime
                Me.OvalShape12.FillColor
= Color.Red
                Me.OvalShape14.FillColor
= Color.Red
                Me.OvalShape16.FillColor
= Color.Lime
                Me.OvalShape18.FillColor
= Color.Red
                Me.OvalShape20.FillColor
= Color.Lime
                Me.OvalShape22.FillColor
= Color.Lime
                Me.OvalShape24.FillColor
= Color.Red
            End If
            If (OvalShape20.FillColor =
Color.Lime) And (m = 1) Then
                G = G + 1
                LabelG.Text = G
                m = 0
            End If
            If Mid(rtbReceived.Text, 8,
4) = "2A54" Then
                PictureBox4.Image =
My.Resources.Cylinder_out
                PictureBox5.Image =
My.Resources.Cylinder_out
                PictureBox6.Image =
My.Resources.Cylinder_out

                flGChnage = True
                chartDataA.Add(10)
                chartDataB.Add(8)
                chartDataC.Add(6)
                chartDataD.Add(5)
                chartDataE.Add(3)
                chartDataF.Add(1)
                removeNode()

                'Chart1.Series("Cylinder
A").Points.AddXY(_A, 10)
                'Chart1.Series("Cylinder
B").Points.AddXY(_C, 8)
                'Chart1.Series("Cylinder
C").Points.AddXY(_F, 6)
                'Chart1.Series("Cylinder
D").Points.AddXY(_G, 5)
                'Chart1.Series("Cylinder
E").Points.AddXY(_I, 3)
                'Chart1.Series("Cylinder
F").Points.AddXY(_K, 1)

                Me.OvalShape2.FillColor =
Color.Lime
                Me.OvalShape4.FillColor =
Color.Red
                Me.OvalShape6.FillColor =
Color.Lime
                Me.OvalShape8.FillColor =
Color.Red
                Me.OvalShape10.FillColor
= Color.Lime
                Me.OvalShape12.FillColor
= Color.Red
                Me.OvalShape14.FillColor
= Color.Red
                Me.OvalShape16.FillColor
= Color.Lime
                Me.OvalShape18.FillColor
= Color.Red
                Me.OvalShape20.FillColor
= Color.Lime
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                Me.OvalShape22.FillColor
= Color.Red
                Me.OvalShape24.FillColor
= Color.Lime

            End If
            If (OvalShape24.FillColor =
Color.Lime) And (n = 1) Then
                F = F + 1
                LabelF.Text = F
                n = 0
            End If
            If Mid(rtbReceived.Text, 8,
4) = "1554" Then
                PictureBox4.Image =
My.Resources.Cylinder_in
                PictureBox5.Image =
My.Resources.Cylinder_in
                PictureBox6.Image =
My.Resources.Cylinder_in

                flGChnage = True
                chartDataA.Add(10)
                chartDataB.Add(8)
                chartDataC.Add(6)
                chartDataD.Add(4)
                chartDataE.Add(2)
                chartDataF.Add(0)
                removeNode()

                'Chart1.Series("Cylinder
A").Points.AddXY(_A, 10)
                'Chart1.Series("Cylinder
B").Points.AddXY(_C, 8)
                'Chart1.Series("Cylinder
C").Points.AddXY(_F, 6)
                'Chart1.Series("Cylinder
D").Points.AddXY(_G, 4)
                'Chart1.Series("Cylinder
E").Points.AddXY(_I, 2)
                'Chart1.Series("Cylinder
F").Points.AddXY(_K, 0)

                Me.OvalShape2.FillColor =
Color.Lime
                Me.OvalShape4.FillColor =
Color.Red
                Me.OvalShape6.FillColor =
Color.Lime
                Me.OvalShape8.FillColor =
Color.Red
                Me.OvalShape10.FillColor
= Color.Lime
                Me.OvalShape12.FillColor
= Color.Red
                Me.OvalShape14.FillColor
= Color.Lime
                Me.OvalShape16.FillColor
= Color.Red
                Me.OvalShape18.FillColor
= Color.Lime
                Me.OvalShape20.FillColor
= Color.Red
                Me.OvalShape22.FillColor
= Color.Lime
                Me.OvalShape24.FillColor
= Color.Red

                i = 1
                j = 1
                k = 1
                l = 1
                m = 1
                n = 1
            End If
        End If

        'Pattern5
        If RadioButtonPattern5.Checked
Then

            If Mid(rtbReceived.Text, 8,
4) = "1558" Then
                PictureBox1.Image =
My.Resources.Cylinder_out

                flGChnage = True
                chartDataA.Add(11)
                chartDataB.Add(8)
                chartDataC.Add(6)
                chartDataD.Add(4)
                chartDataE.Add(2)
                chartDataF.Add(0)
                removeNode()

                'Chart1.Series("Cylinder
A").Points.AddXY(_A, 11)
                'Chart1.Series("Cylinder
B").Points.AddXY(_C, 8)
                'Chart1.Series("Cylinder
C").Points.AddXY(_F, 6)
                'Chart1.Series("Cylinder
D").Points.AddXY(_G, 4)
                'Chart1.Series("Cylinder
E").Points.AddXY(_I, 2)
                'Chart1.Series("Cylinder
F").Points.AddXY(_K, 0)

                Me.OvalShape2.FillColor =
Color.Red
                Me.OvalShape4.FillColor =
Color.Lime
                Me.OvalShape6.FillColor =
Color.Lime
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                Me.OvalShape8.FillColor =
Color.Red
                Me.OvalShape10.FillColor
= Color.Lime
                Me.OvalShape12.FillColor
= Color.Red
                Me.OvalShape14.FillColor
= Color.Lime
                Me.OvalShape16.FillColor
= Color.Red
                Me.OvalShape18.FillColor
= Color.Lime
                Me.OvalShape20.FillColor
= Color.Red
                Me.OvalShape22.FillColor
= Color.Lime
                Me.OvalShape24.FillColor
= Color.Red
            End If
            If (OvalShape4.FillColor =
Color.Lime) And (i = 1) Then
                A = A + 1
                LabelA.Text = A
                i = 0
            End If
            If Mid(rtbReceived.Text, 8,
4) = "1568" Then
                PictureBox1.Image =
My.Resources.Cylinder_out
                PictureBox2.Image =
My.Resources.Cylinder_out

                flGChnage = True
                chartDataA.Add(11)
                chartDataB.Add(9)
                chartDataC.Add(6)
                chartDataD.Add(4)
                chartDataE.Add(2)
                chartDataF.Add(0)
                removeNode()

                'Chart1.Series("Cylinder
A").Points.AddXY(_A, 11)
                'Chart1.Series("Cylinder
B").Points.AddXY(_C, 9)
                'Chart1.Series("Cylinder
C").Points.AddXY(_F, 6)
                'Chart1.Series("Cylinder
D").Points.AddXY(_G, 4)
                'Chart1.Series("Cylinder
E").Points.AddXY(_I, 2)
                'Chart1.Series("Cylinder
F").Points.AddXY(_K, 0)

                Me.OvalShape2.FillColor =
Color.Red
                Me.OvalShape4.FillColor =
Color.Lime
                Me.OvalShape6.FillColor =
Color.Red
                Me.OvalShape8.FillColor =
Color.Lime
                Me.OvalShape10.FillColor
= Color.Lime
                Me.OvalShape12.FillColor
= Color.Red
                Me.OvalShape14.FillColor
= Color.Lime
                Me.OvalShape16.FillColor
= Color.Red
                Me.OvalShape18.FillColor
= Color.Lime
                Me.OvalShape20.FillColor
= Color.Red
                Me.OvalShape22.FillColor
= Color.Lime
                Me.OvalShape24.FillColor
= Color.Red
            End If
            If (OvalShape8.FillColor =
Color.Lime) And (j = 1) Then
                B = B + 1
                LabelB.Text = B
                j = 0
            End If
            If Mid(rtbReceived.Text, 8,
4) = "15A8" Then
                PictureBox1.Image =
My.Resources.Cylinder_out
                PictureBox2.Image =
My.Resources.Cylinder_out
                PictureBox3.Image =
My.Resources.Cylinder_out

                flGChnage = True

                chartDataA.Add(11)
                chartDataB.Add(9)
                chartDataC.Add(7)
                chartDataD.Add(4)
                chartDataE.Add(2)
                chartDataF.Add(0)
                removeNode()

                'Chart1.Series("Cylinder
A").Points.AddXY(_A, 11)
                'Chart1.Series("Cylinder
B").Points.AddXY(_C, 9)
                'Chart1.Series("Cylinder
C").Points.AddXY(_F, 7)
                'Chart1.Series("Cylinder
D").Points.AddXY(_G, 4)
                'Chart1.Series("Cylinder
E").Points.AddXY(_I, 2)
                'Chart1.Series("Cylinder
F").Points.AddXY(_K, 0)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ในเชิงพาณิชย์ ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                Me.OvalShape2.FillColor =
Color.Red
                Me.OvalShape4.FillColor =
Color.Lime
                Me.OvalShape6.FillColor =
Color.Red
                Me.OvalShape8.FillColor =
Color.Lime
                Me.OvalShape10.FillColor
= Color.Red
                Me.OvalShape12.FillColor
= Color.Lime
                Me.OvalShape14.FillColor
= Color.Lime
                Me.OvalShape16.FillColor
= Color.Red
                Me.OvalShape18.FillColor
= Color.Lime
                Me.OvalShape20.FillColor
= Color.Red
                Me.OvalShape22.FillColor
= Color.Lime
                Me.OvalShape24.FillColor
= Color.Red
            End If
            If (OvalShape12.FillColor =
Color.Lime) And (k = 1) Then
                C = C + 1
                LabelC.Text = C
                k = 0
            End If
            If Mid(rtbReceived.Text, 8,
4) = "16A8" Then
                PictureBox1.Image =
My.Resources.Cylinder_out
                PictureBox2.Image =
My.Resources.Cylinder_out
                PictureBox3.Image =
My.Resources.Cylinder_out
                PictureBox4.Image =
My.Resources.Cylinder_out

                flGChnage = True
                chartDataA.Add(11)
                chartDataB.Add(9)
                chartDataC.Add(7)
                chartDataD.Add(5)
                chartDataE.Add(2)
                chartDataF.Add(0)
                removeNode()

                'Chart1.Series("Cylinder
A").Points.AddXY(_A, 11)
                'Chart1.Series("Cylinder
B").Points.AddXY(_C, 9)
                'Chart1.Series("Cylinder
C").Points.AddXY(_F, 7)
                'Chart1.Series("Cylinder
D").Points.AddXY(_G, 5)
                'Chart1.Series("Cylinder
E").Points.AddXY(_I, 2)
                'Chart1.Series("Cylinder
F").Points.AddXY(_K, 0)

                Me.OvalShape2.FillColor =
Color.Red
                Me.OvalShape4.FillColor =
Color.Lime
                Me.OvalShape6.FillColor =
Color.Red
                Me.OvalShape8.FillColor =
Color.Lime
                Me.OvalShape10.FillColor
= Color.Red
                Me.OvalShape12.FillColor
= Color.Lime
                Me.OvalShape14.FillColor
= Color.Red
                Me.OvalShape16.FillColor
= Color.Lime
                Me.OvalShape18.FillColor
= Color.Lime
                Me.OvalShape20.FillColor
= Color.Red
                Me.OvalShape22.FillColor
= Color.Lime
                Me.OvalShape24.FillColor
= Color.Red
            End If
            If (OvalShape16.FillColor =
Color.Lime) And (l = 1) Then
                D = D + 1
                LabelD.Text = D
                l = 0
            End If
            If Mid(rtbReceived.Text, 8,
4) = "1AA8" Then
                PictureBox1.Image =
My.Resources.Cylinder_out
                PictureBox2.Image =
My.Resources.Cylinder_out
                PictureBox3.Image =
My.Resources.Cylinder_out
                PictureBox4.Image =
My.Resources.Cylinder_out
                PictureBox5.Image =
My.Resources.Cylinder_out

                flGChnage = True
                chartDataA.Add(11)
                chartDataB.Add(9)
                chartDataC.Add(7)
                chartDataD.Add(5)
                chartDataE.Add(3)
                chartDataF.Add(0)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                removeNode()

                'Chart1.Series("Cylinder
A").Points.AddXY(_A, 11)
                'Chart1.Series("Cylinder
B").Points.AddXY(_C, 9)
                'Chart1.Series("Cylinder
C").Points.AddXY(_F, 7)
                'Chart1.Series("Cylinder
D").Points.AddXY(_G, 5)
                'Chart1.Series("Cylinder
E").Points.AddXY(_I, 3)
                'Chart1.Series("Cylinder
F").Points.AddXY(_K, 0)

                Me.OvalShape2.FillColor =
Color.Red
                Me.OvalShape4.FillColor =
Color.Lime
                Me.OvalShape6.FillColor =
Color.Red
                Me.OvalShape8.FillColor =
Color.Lime
                Me.OvalShape10.FillColor
= Color.Red
                Me.OvalShape12.FillColor
= Color.Lime
                Me.OvalShape14.FillColor
= Color.Red
                Me.OvalShape16.FillColor
= Color.Lime
                Me.OvalShape18.FillColor
= Color.Red
                Me.OvalShape20.FillColor
= Color.Lime
                Me.OvalShape22.FillColor
= Color.Lime
                Me.OvalShape24.FillColor
= Color.Red
            End If
            If (OvalShape20.FillColor =
Color.Lime) And (m = 1) Then
                G = G + 1
                LabelG.Text = G
                m = 0
            End If
            If Mid(rtbReceived.Text, 8,
4) = "2AA8" Then
                PictureBox1.Image =
My.Resources.Cylinder_out
                PictureBox2.Image =
My.Resources.Cylinder_out
                PictureBox3.Image =
My.Resources.Cylinder_out
                PictureBox4.Image =
My.Resources.Cylinder_out
                PictureBox5.Image =
My.Resources.Cylinder_out
                PictureBox6.Image =
My.Resources.Cylinder_out

                flGChnage = True
                chartDataA.Add(11)
                chartDataB.Add(9)
                chartDataC.Add(7)
                chartDataD.Add(5)
                chartDataE.Add(3)
                chartDataF.Add(1)
                removeNode()

                'Chart1.Series("Cylinder
A").Points.AddXY(_A, 11)
                'Chart1.Series("Cylinder
B").Points.AddXY(_C, 9)
                'Chart1.Series("Cylinder
C").Points.AddXY(_F, 7)
                'Chart1.Series("Cylinder
D").Points.AddXY(_G, 5)
                'Chart1.Series("Cylinder
E").Points.AddXY(_I, 3)
                'Chart1.Series("Cylinder
F").Points.AddXY(_K, 1)

                Me.OvalShape2.FillColor =
Color.Red
                Me.OvalShape4.FillColor =
Color.Lime
                Me.OvalShape6.FillColor =
Color.Red
                Me.OvalShape8.FillColor =
Color.Lime
                Me.OvalShape10.FillColor
= Color.Red
                Me.OvalShape12.FillColor
= Color.Lime
                Me.OvalShape14.FillColor
= Color.Red
                Me.OvalShape16.FillColor
= Color.Lime
                Me.OvalShape18.FillColor
= Color.Red
                Me.OvalShape20.FillColor
= Color.Lime
                Me.OvalShape22.FillColor
= Color.Red
                Me.OvalShape24.FillColor
= Color.Lime
            End If
            If (OvalShape24.FillColor =
Color.Lime) And (n = 1) Then
                F = F + 1
                LabelF.Text = F
                n = 0
            End If
            If Mid(rtbReceived.Text, 8,
4) = "1A68" Then
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                PictureBox1.Image =
My.Resources.Cylinder_out
                PictureBox2.Image =
My.Resources.Cylinder_out
                PictureBox3.Image =
My.Resources.Cylinder_in
                PictureBox4.Image =
My.Resources.Cylinder_out
                PictureBox5.Image =
My.Resources.Cylinder_out
                PictureBox6.Image =
My.Resources.Cylinder_in

                flGChnage = True
                chartDataA.Add(11)
                chartDataB.Add(9)
                chartDataC.Add(6)
                chartDataD.Add(5)
                chartDataE.Add(3)
                chartDataF.Add(0)
                removeNode()

                'Chart1.Series("Cylinder
A").Points.AddXY(_A, 11)
                'Chart1.Series("Cylinder
B").Points.AddXY(_C, 9)
                'Chart1.Series("Cylinder
C").Points.AddXY(_F, 6)
                'Chart1.Series("Cylinder
D").Points.AddXY(_G, 5)
                'Chart1.Series("Cylinder
E").Points.AddXY(_I, 3)
                'Chart1.Series("Cylinder
F").Points.AddXY(_K, 0)

                Me.OvalShape2.FillColor =
Color.Red
                Me.OvalShape4.FillColor =
Color.Lime
                Me.OvalShape6.FillColor =
Color.Red
                Me.OvalShape8.FillColor =
Color.Lime
                Me.OvalShape10.FillColor
= Color.Lime
                Me.OvalShape12.FillColor
= Color.Red
                Me.OvalShape14.FillColor
= Color.Red
                Me.OvalShape16.FillColor
= Color.Lime
                Me.OvalShape18.FillColor
= Color.Red
                Me.OvalShape20.FillColor
= Color.Lime
                Me.OvalShape22.FillColor
= Color.Lime
                Me.OvalShape24.FillColor
= Color.Red
            End If
            If Mid(rtbReceived.Text, 8,
4) = "1658" Then
                PictureBox1.Image =
My.Resources.Cylinder_out
                PictureBox2.Image =
My.Resources.Cylinder_in
                PictureBox3.Image =
My.Resources.Cylinder_in
                PictureBox4.Image =
My.Resources.Cylinder_out
                PictureBox5.Image =
My.Resources.Cylinder_in
                PictureBox6.Image =
My.Resources.Cylinder_in

                flGChnage = True
                chartDataA.Add(11)
                chartDataB.Add(8)
                chartDataC.Add(6)
                chartDataD.Add(5)
                chartDataE.Add(2)
                chartDataF.Add(0)
                removeNode()

                'Chart1.Series("Cylinder
A").Points.AddXY(_A, 11)
                'Chart1.Series("Cylinder
B").Points.AddXY(_C, 8)
                'Chart1.Series("Cylinder
C").Points.AddXY(_F, 6)
                'Chart1.Series("Cylinder
D").Points.AddXY(_G, 5)
                'Chart1.Series("Cylinder
E").Points.AddXY(_I, 2)
                'Chart1.Series("Cylinder
F").Points.AddXY(_K, 0)

                Me.OvalShape2.FillColor =
Color.Red
                Me.OvalShape4.FillColor =
Color.Lime
                Me.OvalShape6.FillColor =
Color.Lime
                Me.OvalShape8.FillColor =
Color.Red
                Me.OvalShape10.FillColor
= Color.Lime
                Me.OvalShape12.FillColor
= Color.Red
                Me.OvalShape14.FillColor
= Color.Red
                Me.OvalShape16.FillColor
= Color.Lime
                Me.OvalShape18.FillColor
= Color.Lime
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                Me.OvalShape20.FillColor
= Color.Red
                Me.OvalShape22.FillColor
= Color.Lime
                Me.OvalShape24.FillColor
= Color.Red
            End If
            If Mid(rtbReceived.Text, 8,
4) = "1554" Then
                PictureBox1.Image =
My.Resources.Cylinder_in
                PictureBox2.Image =
My.Resources.Cylinder_in
                PictureBox3.Image =
My.Resources.Cylinder_in
                PictureBox4.Image =
My.Resources.Cylinder_in
                PictureBox5.Image =
My.Resources.Cylinder_in
                PictureBox6.Image =
My.Resources.Cylinder_in

                flGChnage = True
                chartDataA.Add(10)
                chartDataB.Add(8)
                chartDataC.Add(6)
                chartDataD.Add(4)
                chartDataE.Add(2)
                chartDataF.Add(0)
                removeNode()


                Chart1.Series("Cylinder
A").Points.AddXY(_A, 10)
                Chart1.Series("Cylinder
B").Points.AddXY(_C, 8)
                Chart1.Series("Cylinder
C").Points.AddXY(_F, 6)
                Chart1.Series("Cylinder
D").Points.AddXY(_G, 4)
                Chart1.Series("Cylinder
E").Points.AddXY(_I, 2)
                Chart1.Series("Cylinder
F").Points.AddXY(_K, 0)


                Me.OvalShape2.FillColor =
Color.Lime
                Me.OvalShape4.FillColor =
Color.Red
                Me.OvalShape6.FillColor =
Color.Lime
                Me.OvalShape8.FillColor =
Color.Red
                Me.OvalShape10.FillColor
= Color.Lime
                Me.OvalShape12.FillColor
= Color.Red
                Me.OvalShape14.FillColor
= Color.Lime
                Me.OvalShape16.FillColor
= Color.Red
                Me.OvalShape18.FillColor
= Color.Lime
                Me.OvalShape20.FillColor
= Color.Red
                Me.OvalShape22.FillColor
= Color.Lime
                Me.OvalShape24.FillColor
= Color.Red

                i = 1
                j = 1
                k = 1
                l = 1
                m = 1
                n = 1
            End If
        End If

        plotGraph()

    End Sub

    Private Sub Timer2_Tick(ByVal sender
As System.Object, ByVal e As
System.EventArgs) Handles Timer2.Tick
        TimeStatusLabel.Text =
Now.ToLongTimeString.ToString()
    End Sub

    Private Sub plotGraph()
        removeNode()

        Dim currentDateTime As DateTime =
DateTime.Now
        Dim format As String = "HH:mm:ss"
        Dim test12324 As String =
currentDateTime.ToString(format)

        chartDataTime.Add(test12324)
        If chartDataTime.Count >
maxTimeItem Then
            chartDataTime.RemoveAt(0)
        End If

        Chart1.Series("Cylinder
A").Points.Clear()
        Chart1.Series("Cylinder
B").Points.Clear()
        Chart1.Series("Cylinder
C").Points.Clear()
        Chart1.Series("Cylinder
D").Points.Clear()
        Chart1.Series("Cylinder
E").Points.Clear()
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        Chart1.Series("Cylinder
F").Points.Clear()

        For i As Integer = 0 To
chartDataA.Count - 1
            Chart1.Series("Cylinder
A").Points.AddXY(chartDataTime(i),
chartDataA(i))
        Next
        For i As Integer = 0 To
chartDataB.Count - 1
            Chart1.Series("Cylinder
B").Points.AddXY(chartDataTime(i),
chartDataB(i))
        Next
        For i As Integer = 0 To
chartDataC.Count - 1
            Chart1.Series("Cylinder
C").Points.AddXY(chartDataTime(i),
chartDataC(i))
        Next
        For i As Integer = 0 To
chartDataD.Count - 1
            Chart1.Series("Cylinder
D").Points.AddXY(chartDataTime(i),
chartDataD(i))
        Next
        For i As Integer = 0 To
chartDataE.Count - 1
            Chart1.Series("Cylinder
E").Points.AddXY(chartDataTime(i),
chartDataE(i))
        Next
        For i As Integer = 0 To
chartDataF.Count - 1
            Chart1.Series("Cylinder
F").Points.AddXY(chartDataTime(i),
chartDataF(i))
        Next
    End Sub

    Private Sub removeNode()

        If chartDataA.Count > maxItem
Then
            chartDataA.RemoveAt(0)
        End If
        If chartDataB.Count > maxItem
Then
            chartDataB.RemoveAt(0)
        End If
        If chartDataC.Count > maxItem
Then
            chartDataC.RemoveAt(0)
        End If
        If chartDataD.Count > maxItem
Then
            chartDataD.RemoveAt(0)
        End If
        If chartDataE.Count > maxItem
Then
            chartDataE.RemoveAt(0)
        End If
        If chartDataF.Count > maxItem
Then
            chartDataF.RemoveAt(0)
        End If
    End Sub
End Class
```

```
''FormPM
Imports System.Data.OleDb
Public Class FrmPM
    Dim cn As New
System.Data.OleDb.OleDbConnection
    Dim cmd As
System.Data.OleDb.OleDbCommand
    Dim myDA As
System.Data.OleDb.OleDbDataAdapter
    Dim myDataSet As System.Data.DataSet
    Sub DataSetFill()
        Try
            cn.ConnectionString =
"Provider=Microsoft.ACE.OLEDB.12.0;Data
Source=" &
My.Application.Info.DirectoryPath.ToStrin
g() & "\BackUp\testing.Accdb;Persist
Security Info=False;"
            If cn.State -
System.Data.ConnectionState.Closed Then
                cn.Open()
            End If
            cmd = New
System.Data.OleDb.OleDbCommand("SELECT
(number)as[Number],(DateofCylinder)as
[วันที่/เดือน/ปี],(CylinderA)as [Cylinder
A],(CylinderB) as [Cylinder
B],(CylinderC)as [Cylinder C],
(CylinderD) as [Cylinder D],
(CylinderE)as [Cylinder E], (CylinderF)as
[Cylinder F] FROM Cylinder ", cn)
            myDA = New
System.Data.OleDb.OleDbDataAdapter(cmd)
            myDataSet = New
System.Data.DataSet
            myDA.Fill(myDataSet,
"Cylinder")
            DataGridView1.DataSource =
myDataSet.Tables("Cylinder").DefaultView
            cn.Close()
        Catch ex As Exception
            MsgBox(ex.Message(),
MsgBoxStyle.Critical, "Error")
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        End Try
    End Sub
    Private Sub FrmPM_FormClosing(ByVal
sender As Object, ByVal e As
System.Windows.Forms.FormClosingEventArgs
) Handles Me.FormClosing
        Me.Dispose()
        FrmMain.Show()
    End Sub
    Private Sub FrmPM_Load(ByVal sender
As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
        Call DataSetFill()
    End Sub

    Private Sub Delete_Click(ByVal sender
As System.Object, ByVal e As
System.EventArgs) Handles Delete.Click
        Try
            If MessageBox.Show("Do you
really want to delete the record?",
"Response", MessageBoxButtons.YesNo,
MessageBoxIcon.Question) =
Windows.Forms.DialogResult.Yes Then
                Dim cn As New
OleDbConnection("Provider=Microsoft.ACE.O
LEDB.12.0;Data Source=" &
My.Application.Info.DirectoryPath.ToStrin
g() & "\BackUp\testing.Accdb;Persist
Security Info=False;")
                If cn.State =
ConnectionState.Open Then
                    cn.Close()
                End If
                cn.Open()
                Dim temp As Integer = 0
                Dim cq As String =
"delete from Cylinder where
number=@DELETE1;"

                cmd = New
OleDbCommand(cq)
                cmd.Connection = cn

                cmd.Parameters.Add(New
OleDbParameter("@DELETE1",
System.Data.OleDb.OleDbType.VarChar, 8,
"number"))

cmd.Parameters("@DELETE1").Value =
DataGridView1.SelectedCells(0).Value.ToSt
ring()

                temp =
cmd.ExecuteNonQuery()
                If temp > 0 Then
                    cn.Close()
                    Call DataSetFill()
MessageBox.Show("Successfully deleted",
"Record", MessageBoxButtons.OK,
MessageBoxIcon.Information)

                Else
                    Call DataSetFill()
                    MessageBox.Show("No
record found", "Sorry",
MessageBoxButtons.OK,
MessageBoxIcon.Error)
                End If

            End If
        Catch ex As Exception
            MsgBox(ex.Message(),
MsgBoxStyle.Critical, "Error")
        End Try
    End Sub

End Class


''FormMaintenance
Imports System.Data.OleDb
Public Class FrmMaintenance

    Private Sub
FrmMaintenance_FormClosing(ByVal sender
As System.Object, ByVal e As
System.Windows.Forms.FormClosingEventArgs
) Handles MyBase.FormClosing
        Me.Dispose()
        Timer1.Stop()
        FrmMain.Show()
    End Sub

    Private Sub FrmMaintenance_Load(ByVal
sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
        Timer1.Start()
    End Sub

    Private Sub btnClearA_Click(ByVal
sender As System.Object, ByVal e As
System.EventArgs) Handles btnClearA.Click
        Try
            Dim cn As New
OleDbConnection("Provider=Microsoft.ACE.O
LEDB.12.0;Data Source=" &
My.Application.Info.DirectoryPath.ToStrin
g() & "\BackUp\testing.Accdb;Persist
Security Info=False;")
            If cn.State =
ConnectionState.Open Then
                cn.Close()
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            End If
            cn.Open()
            Dim dr1 As OleDbDataReader
            Dim com As New OleDbCommand
            com.CommandText = "select
[number] from Table1 where number =
@Number"


            ' Number
            Dim Number As OleDbParameter
= New OleDbParameter("@Number",
OleDbType.VarWChar, 20)
            Number.Value = 1
            com.Parameters.Add(Number)
            com.Connection = cn

            dr1 = com.ExecuteReader
            If dr1.Read Then
                If UCase(dr1("number")) =
1 Then


                    Dim sSQL As String =
"Update Table1 Set CylinderA=@d11 where
number = 1"
                    Dim cmd As
OleDbCommand = New OleDbCommand(sSQL, cn)


                    ' CylinderA
                    Dim AA As
OleDbParameter = New
OleDbParameter("@d11",
OleDbType.VarWChar, 15)
                    AA.Value = "0"

cmd.Parameters.Add(AA)


                    If
cmd.ExecuteNonQuery() Then
                        cn.Close()

MsgBox("Maintenance CylinderA
Sucessfully", MsgBoxStyle.Information,
"Record Updated")

                    Else
                        MsgBox("Your info
modification failed ",
MsgBoxStyle.Critical, "Modificatio
Failed")
                        Return
                    End If

                End If
            End If
        Catch ex As Exception
            MsgBox(ex.Message(),
MsgBoxStyle.Critical, "Error")
            Exit Sub
        End Try
    End Sub

    Private Sub btnClearB_Click(ByVal
sender As System.Object, ByVal e As
System.EventArgs) Handles btnClearB.Click
        Try
            Dim cn As New
OleDbConnection("Provider=Microsoft.ACE.O
LEDB.12.0;Data Source=" &
My.Application.Info.DirectoryPath.ToStrin
g() & "\BackUp\testing.Accdb;Persist
Security Info=False;")
            If cn.State =
ConnectionState.Open Then
                cn.Close()
            End If
            cn.Open()
            Dim dr1 As OleDbDataReader
            Dim com As New OleDbCommand
            com.CommandText = "select
[number] from Table1 where number =
@Number"


            ' Number
            Dim Number As OleDbParameter
= New OleDbParameter("@Number",
OleDbType.VarWChar, 20)
            Number.Value = 1
            com.Parameters.Add(Number)
            com.Connection = cn

            dr1 = com.ExecuteReader
            If dr1.Read Then
                If UCase(dr1("number")) =
1 Then


                    Dim sSQL As String =
"Update Table1 Set CylinderB=@d12 where
number = 1"
                    Dim cmd As
OleDbCommand = New OleDbCommand(sSQL, cn)


                    ' CylinderB
                    Dim AA As
OleDbParameter = New
OleDbParameter("@d12",
OleDbType.VarWChar, 15)
                    AA.Value = "0"
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
cmd.Parameters.Add(AA)


                    If
cmd.ExecuteNonQuery() Then
                        cn.Close()

MsgBox("Maintenance CylinderB
Sucessfully", MsgBoxStyle.Information,
"Record Updated")

                    Else
                        MsgBox("Your info
modification failed ",
MsgBoxStyle.Critical, "Modificatio
Failed")
                        Return
                    End If

                End If

            End If
        Catch ex As Exception
            MsgBox(ex.Message(),
MsgBoxStyle.Critical, "Error")
            Exit Sub
        End Try
    End Sub

    Private Sub btnClearC_Click(ByVal
sender As System.Object, ByVal e As
System.EventArgs) Handles btnClearC.Click
        Try
            Dim cn As New
OleDbConnection("Provider=Microsoft.ACE.O
LEDB.12.0;Data Source=" &
My.Application.Info.DirectoryPath.ToStrin
g() & "\BackUp\testing.Accdb;Persist
Security Info=False;")
            If cn.State =
ConnectionState.Open Then
                cn.Close()
            End If
            cn.Open()
            Dim dr1 As OleDbDataReader
            Dim com As New OleDbCommand
            com.CommandText = "select
[number] from Table1 where number =
@Number"


            ' Number
            Dim Number As OleDbParameter
= New OleDbParameter("@Number",
OleDbType.VarWChar, 20)
            Number.Value = 1
            com.Parameters.Add(Number)

            com.Connection = cn

            dr1 = com.ExecuteReader
            If dr1.Read Then
                If UCase(dr1("number")) =
1 Then


                    Dim sSQL As String =
"Update Table1 Set CylinderC=@d13 where
number = 1"
                    Dim cmd As
OleDbCommand = New OleDbCommand(sSQL, cn)


                    ' CylinderC
                    Dim AA As
OleDbParameter = New
OleDbParameter("@d13",
OleDbType.VarWChar, 15)
                    AA.Value = "0"

cmd.Parameters.Add(AA)


                    If
cmd.ExecuteNonQuery() Then
                        cn.Close()

MsgBox("Maintenance CylinderC
Sucessfully", MsgBoxStyle.Information,
"Record Updated")

                    Else
                        MsgBox("Your info
modification failed ",
MsgBoxStyle.Critical, "Modificatio
Failed")
                        Return
                    End If

                End If

            End If
        Catch ex As Exception
            MsgBox(ex.Message(),
MsgBoxStyle.Critical, "Error")
            Exit Sub
        End Try
    End Sub

    Private Sub btnClearD_Click(ByVal
sender As System.Object, ByVal e As
System.EventArgs) Handles btnClearD.Click
        Try
            Dim cn As New
OleDbConnection("Provider=Microsoft.ACE.O
LEDB.12.0;Data Source=" &
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
My.Application.Info.DirectoryPath.ToStrin
g() & "\BackUp\testing.Accdb;Persist
Security Info=False;")
            If cn.State =
ConnectionState.Open Then
                cn.Close()
            End If
            cn.Open()
            Dim dr1 As OleDbDataReader
            Dim com As New OleDbCommand
            com.CommandText = "select
[number] from Table1 where number =
@Number"


            ' Number
            Dim Number As OleDbParameter
= New OleDbParameter("@Number",
OleDbType.VarWChar, 20)
            Number.Value = 1
            com.Parameters.Add(Number)
            com.Connection = cn

            dr1 = com.ExecuteReader
            If dr1.Read Then
                If UCase(dr1("number")) =
1 Then


                    Dim sSQL As String =
"Update Table1 Set CylinderD=@d14 where
number = 1"
                    Dim cmd As
OleDbCommand = New OleDbCommand(sSQL, cn)


                    ' CylinderD
                    Dim AA As
OleDbParameter = New
OleDbParameter("@d14",
OleDbType.VarWChar, 15)
                    AA.Value = "0"

cmd.Parameters.Add(AA)


                    If
cmd.ExecuteNonQuery() Then
                        cn.Close()

MsgBox("Maintenance CylinderD
Sucessfully", MsgBoxStyle.Information,
"Record Updated")

                    Else
                        MsgBox("Your info
modification failed ",
MsgBoxStyle.Critical, "Modificatio
Failed")
                        Return
                    End If

                End If

            End If
        Catch ex As Exception
            MsgBox(ex.Message(),
MsgBoxStyle.Critical, "Error")
            Exit Sub
        End Try
    End Sub

    Private Sub btnClearE_Click(ByVal
sender As System.Object, ByVal e As
System.EventArgs) Handles btnClearE.Click
        Try
            Dim cn As New
OleDbConnection("Provider=Microsoft.ACE.O
LEDB.12.0;Data Source=" &
My.Application.Info.DirectoryPath.ToStrin
g() & "\BackUp\testing.Accdb;Persist
Security Info=False;")
            If cn.State =
ConnectionState.Open Then
                cn.Close()
            End If
            cn.Open()
            Dim dr1 As OleDbDataReader
            Dim com As New OleDbCommand
            com.CommandText = "select
[number] from Table1 where number =
@Number"


            ' Number
            Dim Number As OleDbParameter
= New OleDbParameter("@Number",
OleDbType.VarWChar, 20)
            Number.Value = 1
            com.Parameters.Add(Number)
            com.Connection = cn

            dr1 = com.ExecuteReader
            If dr1.Read Then
                If UCase(dr1("number")) =
1 Then


                    Dim sSQL As String =
"Update Table1 Set CylinderE=@d15 where
number = 1"
                    Dim cmd As
OleDbCommand = New OleDbCommand(sSQL, cn)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                    ' CylinderE
                    Dim AA As
OleDbParameter = New
OleDbParameter("@d15",
OleDbType.VarWChar, 15)
                    AA.Value = "0"

cmd.Parameters.Add(AA)


                    If
cmd.ExecuteNonQuery() Then
                        cn.Close()

MsgBox("Maintenance CylinderE
Sucessfully", MsgBoxStyle.Information,
"Record Updated")

                    Else
                        MsgBox("Your info
modification failed ",
MsgBoxStyle.Critical, "Modificatio
Failed")
                        Return
                    End If

                End If

            End If
        Catch ex As Exception
            MsgBox(ex.Message(),
MsgBoxStyle.Critical, "Error")
            Exit Sub
        End Try
    End Sub

    Private Sub btnClearF_Click(ByVal
sender As System.Object, ByVal e As
System.EventArgs) Handles btnClearF.Click
        Try
            Dim cn As New
OleDbConnection("Provider=Microsoft.ACE.O
LEDB.12.0;Data Source=" &
My.Application.Info.DirectoryPath.ToStrin
g() & "\BackUp\testing.Accdb;Persist
Security Info=False;")
            If cn.State =
ConnectionState.Open Then
                cn.Close()
            End If
            cn.Open()
            Dim dr1 As OleDbDataReader
            Dim com As New OleDbCommand
            com.CommandText = "select
[number] from Table1 where number =
@Number"
            ' Number
            Dim Number As OleDbParameter
= New OleDbParameter("@Number",
OleDbType.VarWChar, 20)
            Number.Value = 1
            com.Parameters.Add(Number)
            com.Connection = cn

            dr1 = com.ExecuteReader
            If dr1.Read Then
                If UCase(dr1("number")) =
1 Then

                    Dim sSQL As String =
"Update Table1 Set CylinderF=@d16 where
number = 1"
                    Dim cmd As
OleDbCommand = New OleDbCommand(sSQL, cn)


                    ' CylinderF
                    Dim AA As
OleDbParameter = New
OleDbParameter("@d16",
OleDbType.VarWChar, 15)
                    AA.Value = "0"

cmd.Parameters.Add(AA)


                    If
cmd.ExecuteNonQuery() Then
                        cn.Close()

MsgBox("Maintenance CylinderF
Sucessfully", MsgBoxStyle.Information,
"Record Updated")

                    Else
                        MsgBox("Your info
modification failed ",
MsgBoxStyle.Critical, "Modificatio
Failed")
                        Return
                    End If

                End If

            End If
        Catch ex As Exception
            MsgBox(ex.Message(),
MsgBoxStyle.Critical, "Error")
            Exit Sub
        End Try
    End Sub
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

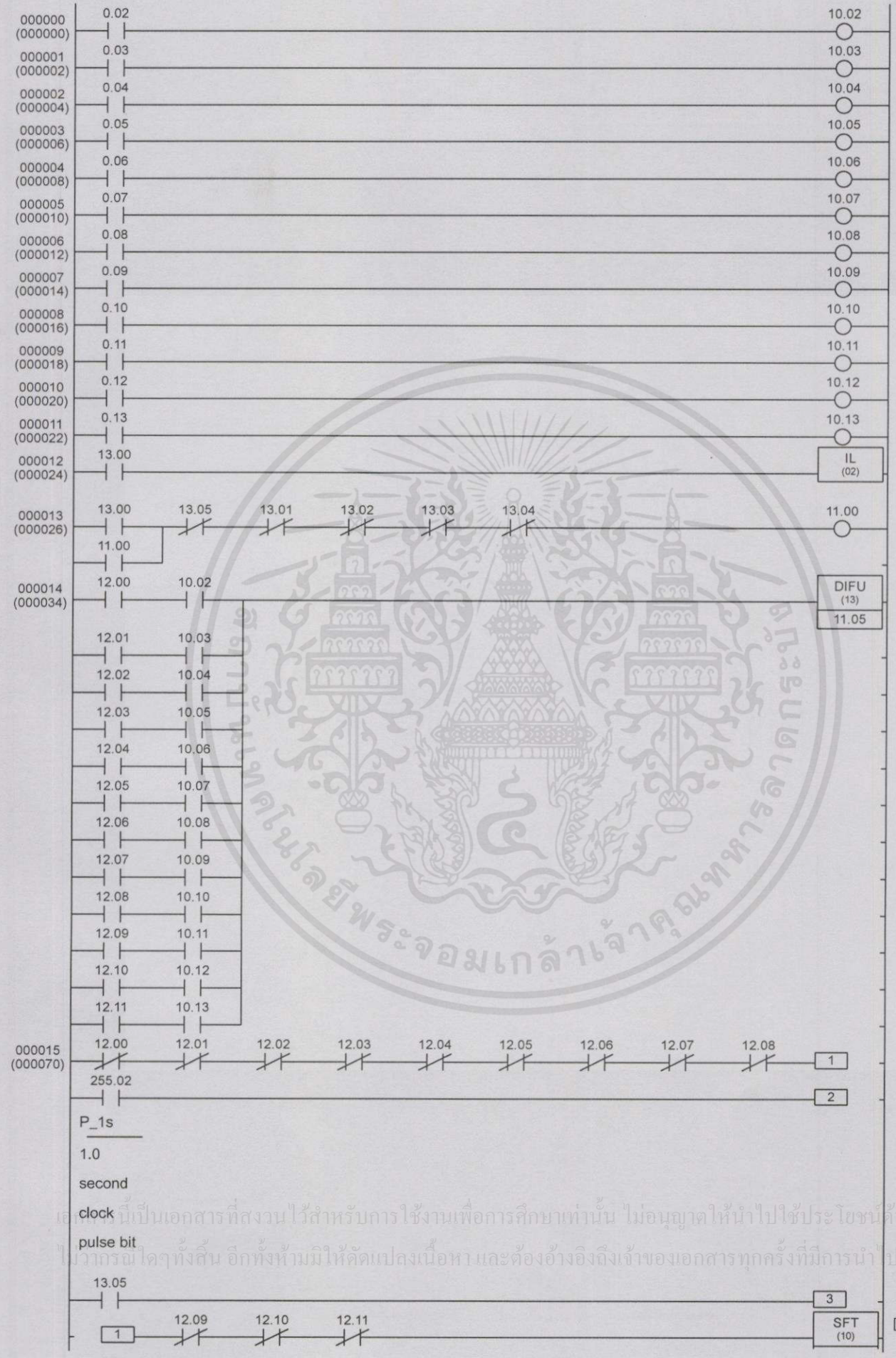
```
    Private Sub Timer1_Tick(ByVal sender
As System.Object, ByVal e As
System.EventArgs) Handles Timer1.Tick
        Dim cn As New
OleDbConnection("Provider=Microsoft.ACE.O
LEDB.12.0;Data Source=" &
My.Application.Info.DirectoryPath.ToStrin
g() & "\BackUp\testing.Accdb;Persist
Security Info=False;")
        If cn.State =
ConnectionState.Open Then
            cn.Close()
        End If
        cn.Open()

        Dim dr1 As OleDbDataReader
        Dim com1 As New OleDbCommand

        com1.CommandText = "select
[CylinderA],[CylinderB],[CylinderC],[Cyli
nderD],[CylinderE],[CylinderF] from
Table1 where number = 1"
        com1.Connection = cn
        dr1 = com1.ExecuteReader
        If dr1.Read Then
            TextBox1.Text = "" &
UCase(dr1(0)).ToString() & ""
            TextBox2.Text = "" &
UCase(dr1(1)).ToString() & ""
            TextBox3.Text = "" &
UCase(dr1(2)).ToString() & ""
            TextBox4.Text = "" &
UCase(dr1(3)).ToString() & ""
            TextBox5.Text = "" &
UCase(dr1(4)).ToString() & ""
            TextBox6.Text = "" &
UCase(dr1(5)).ToString() & ""

        End If
        cn.Close()
        dr1.Close()
        Return
    End Sub
End Class
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

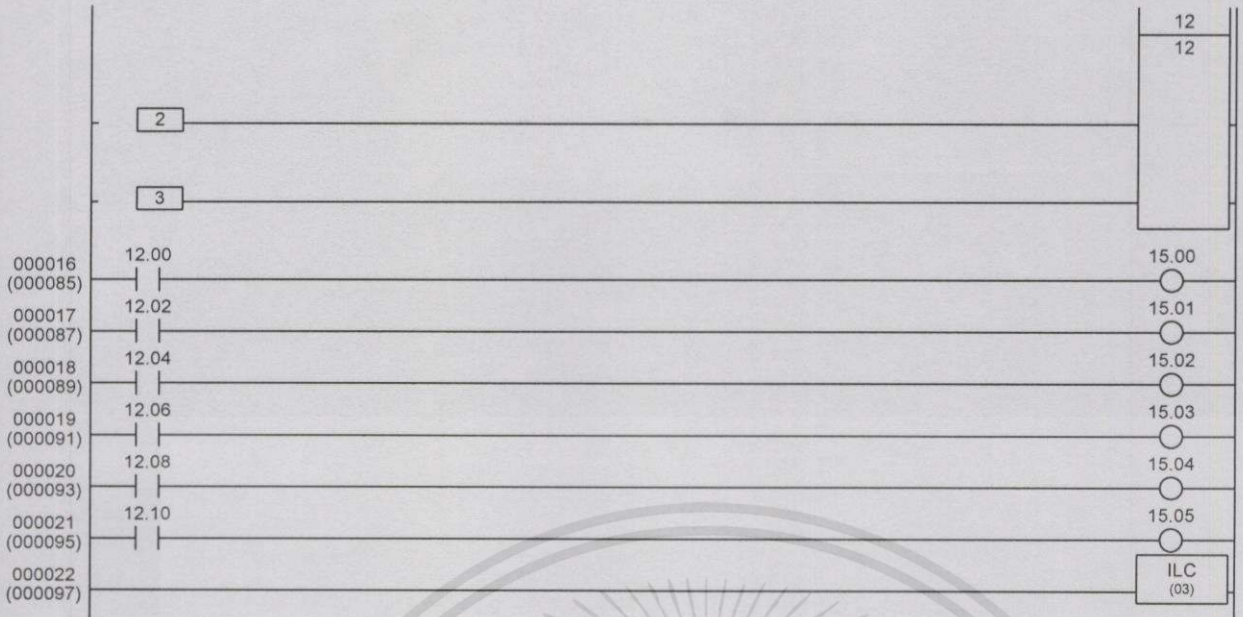
ภาคผนวก ข

โปรแกรมของตัวควบคุม (PLC)

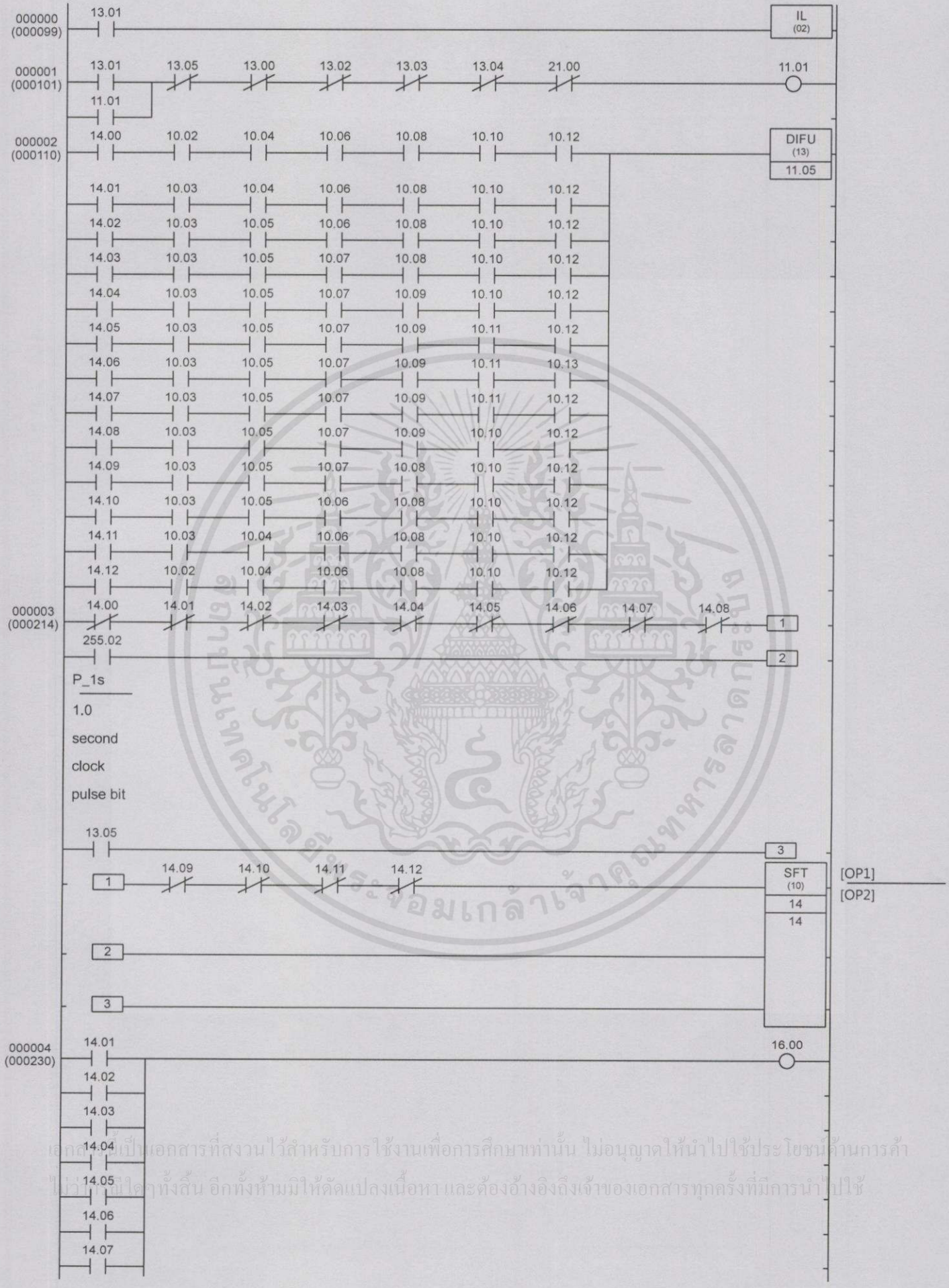
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



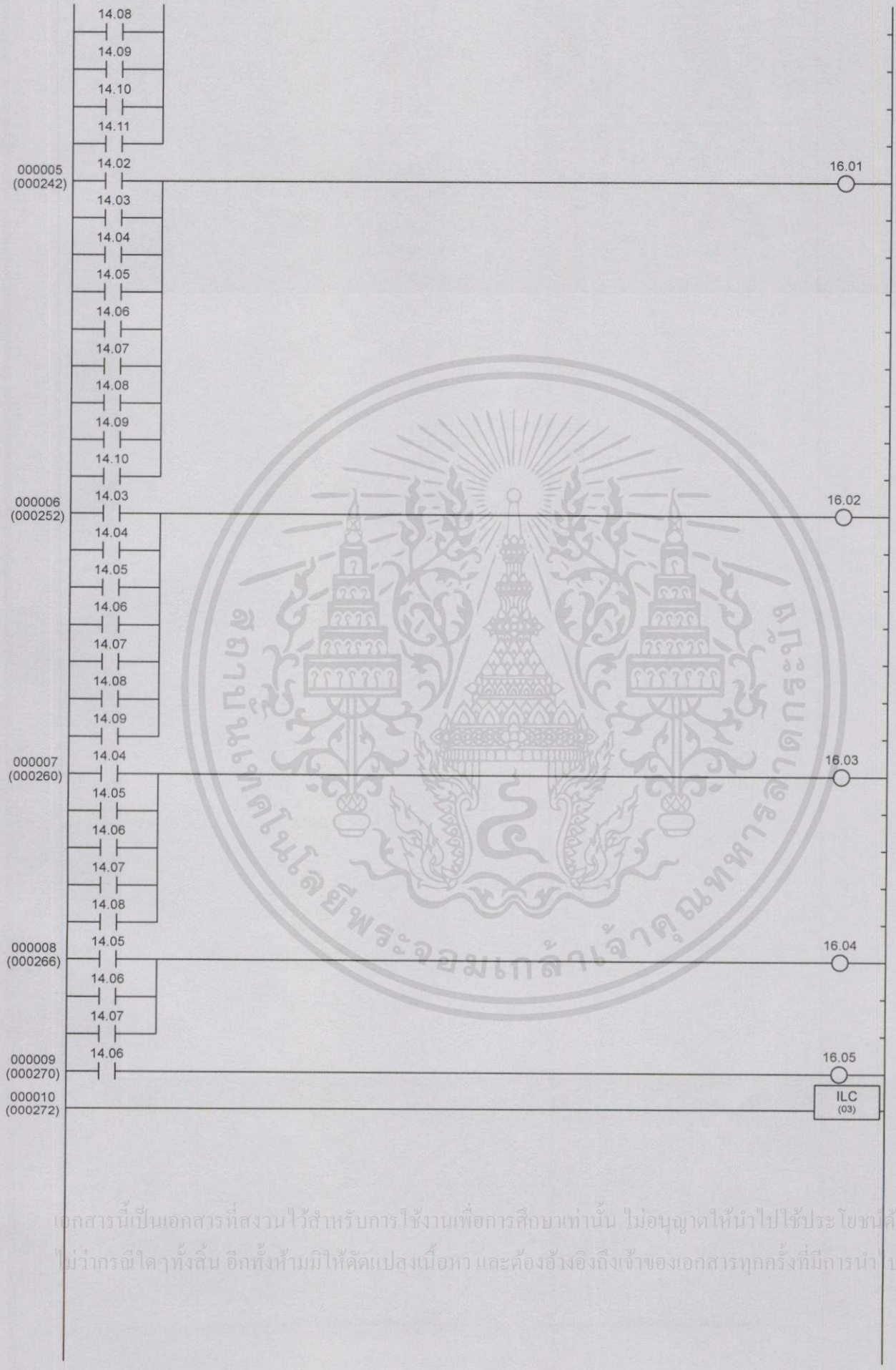
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



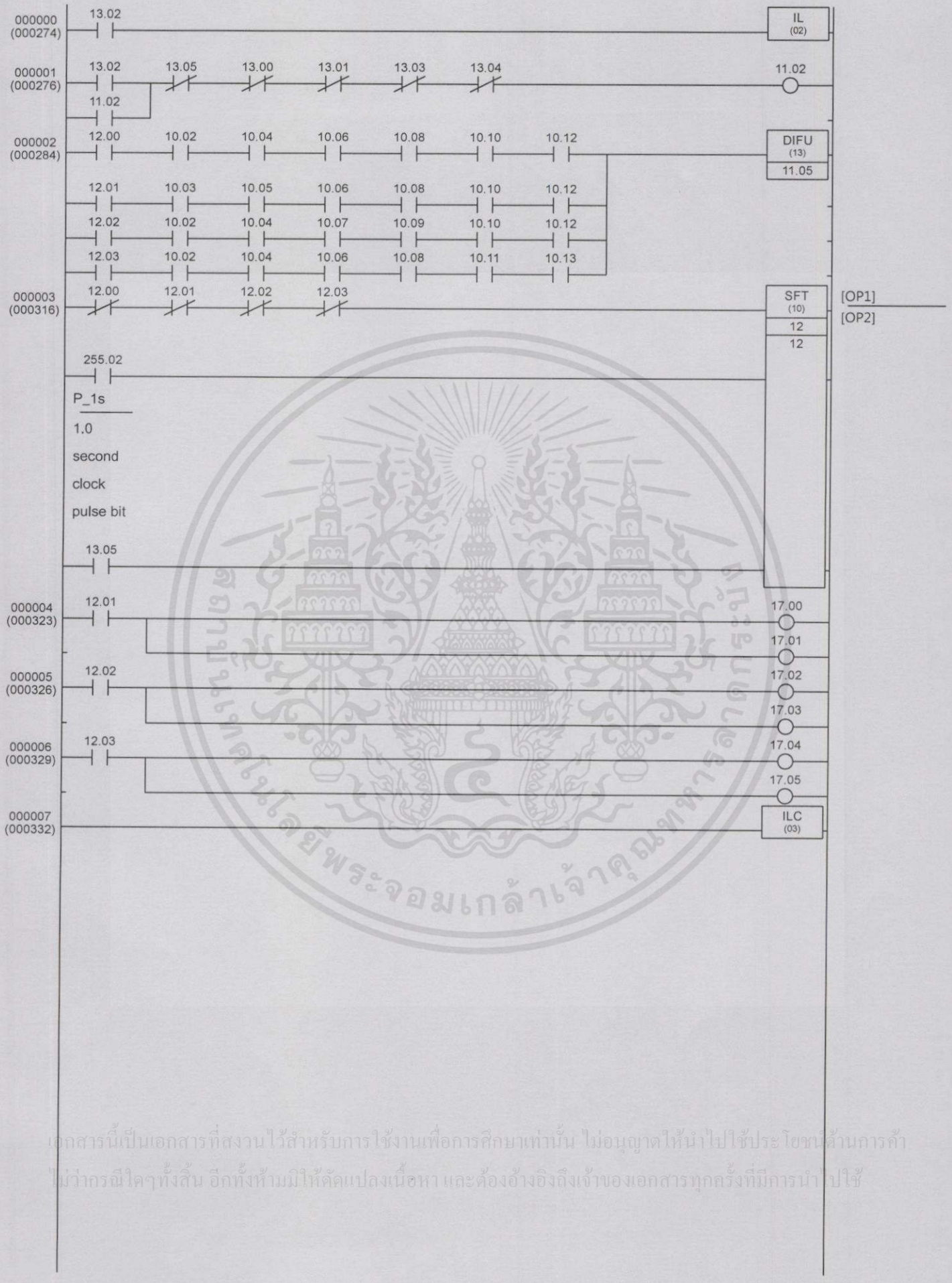
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



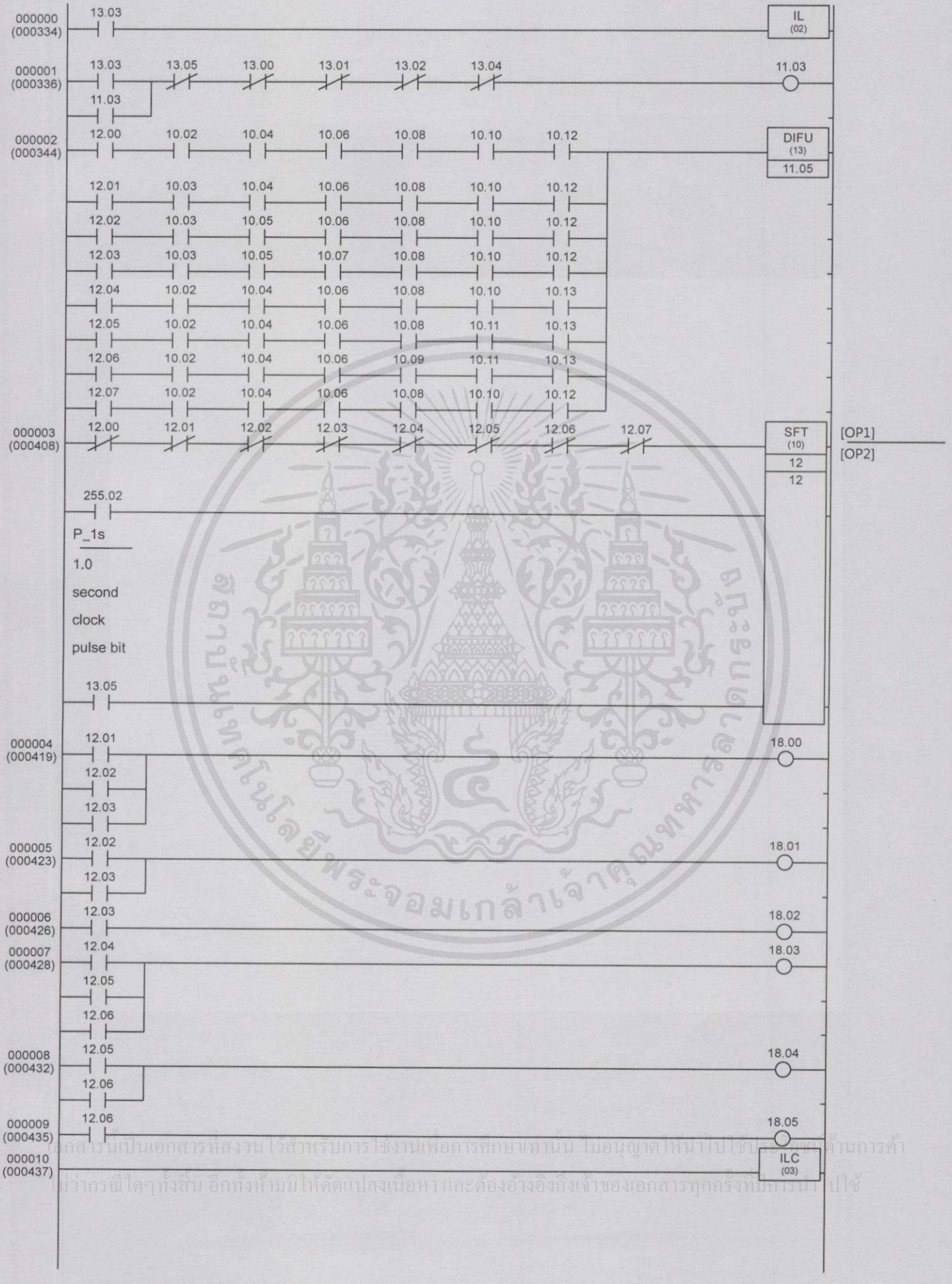
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



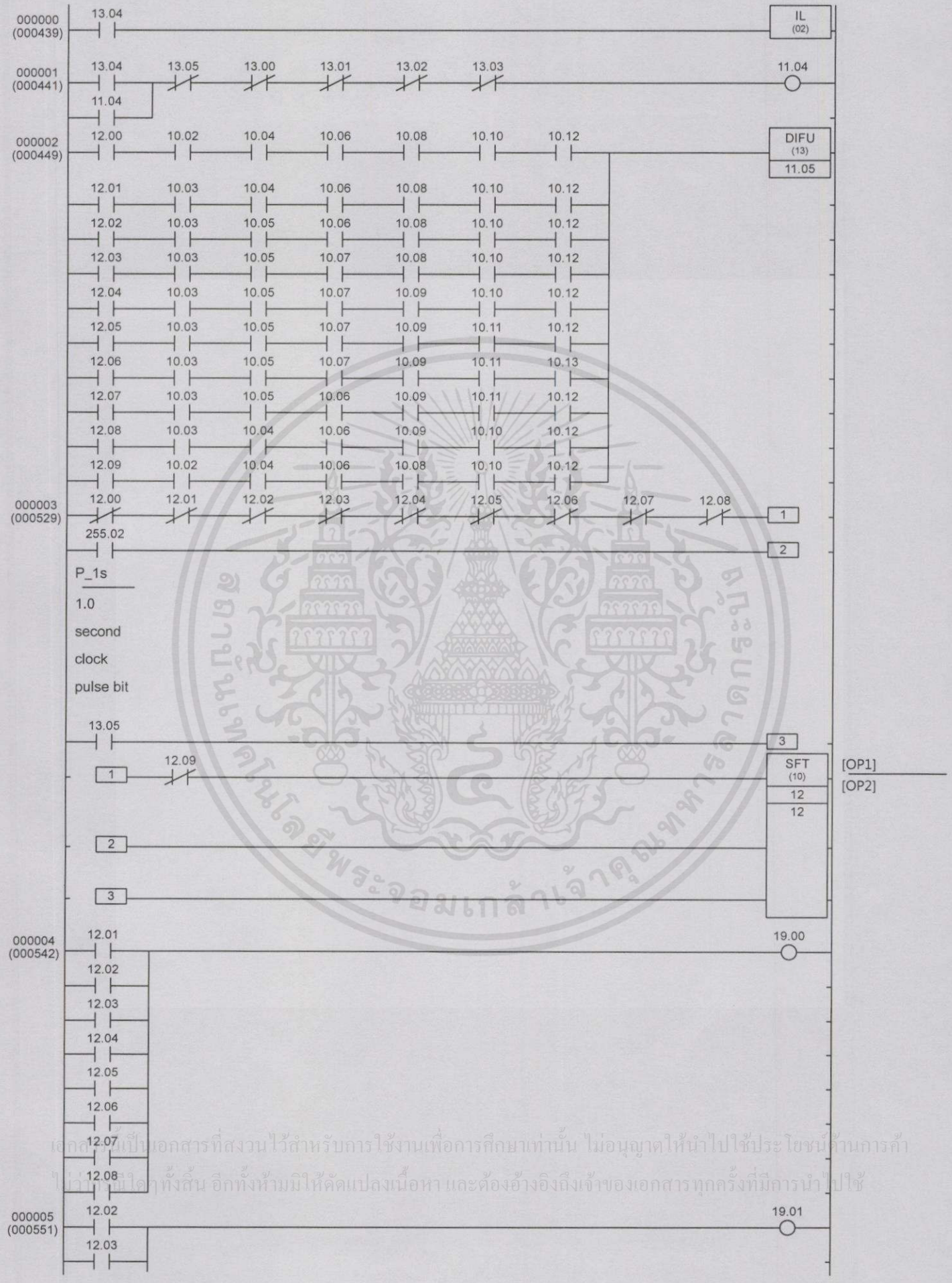
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



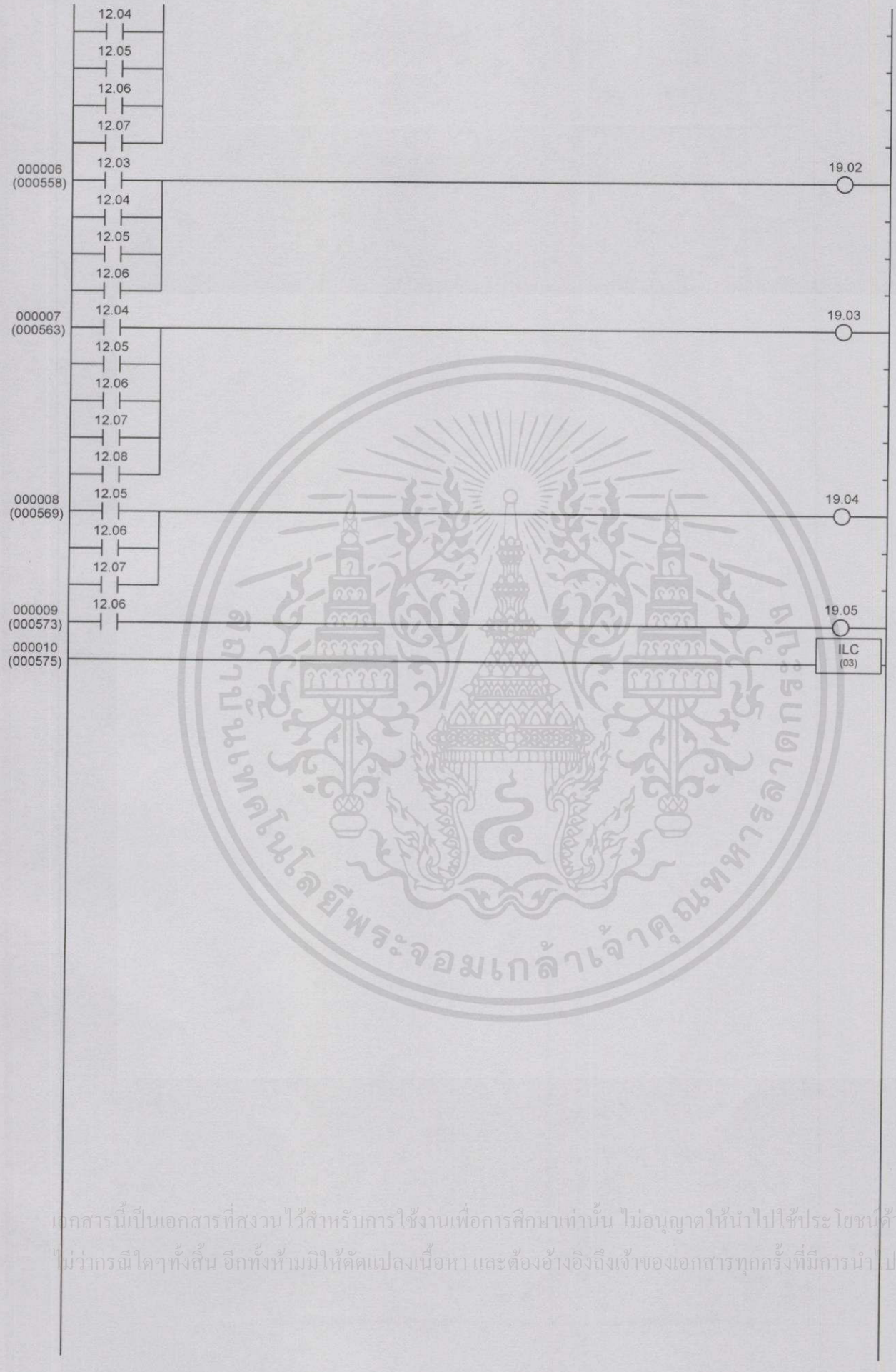
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



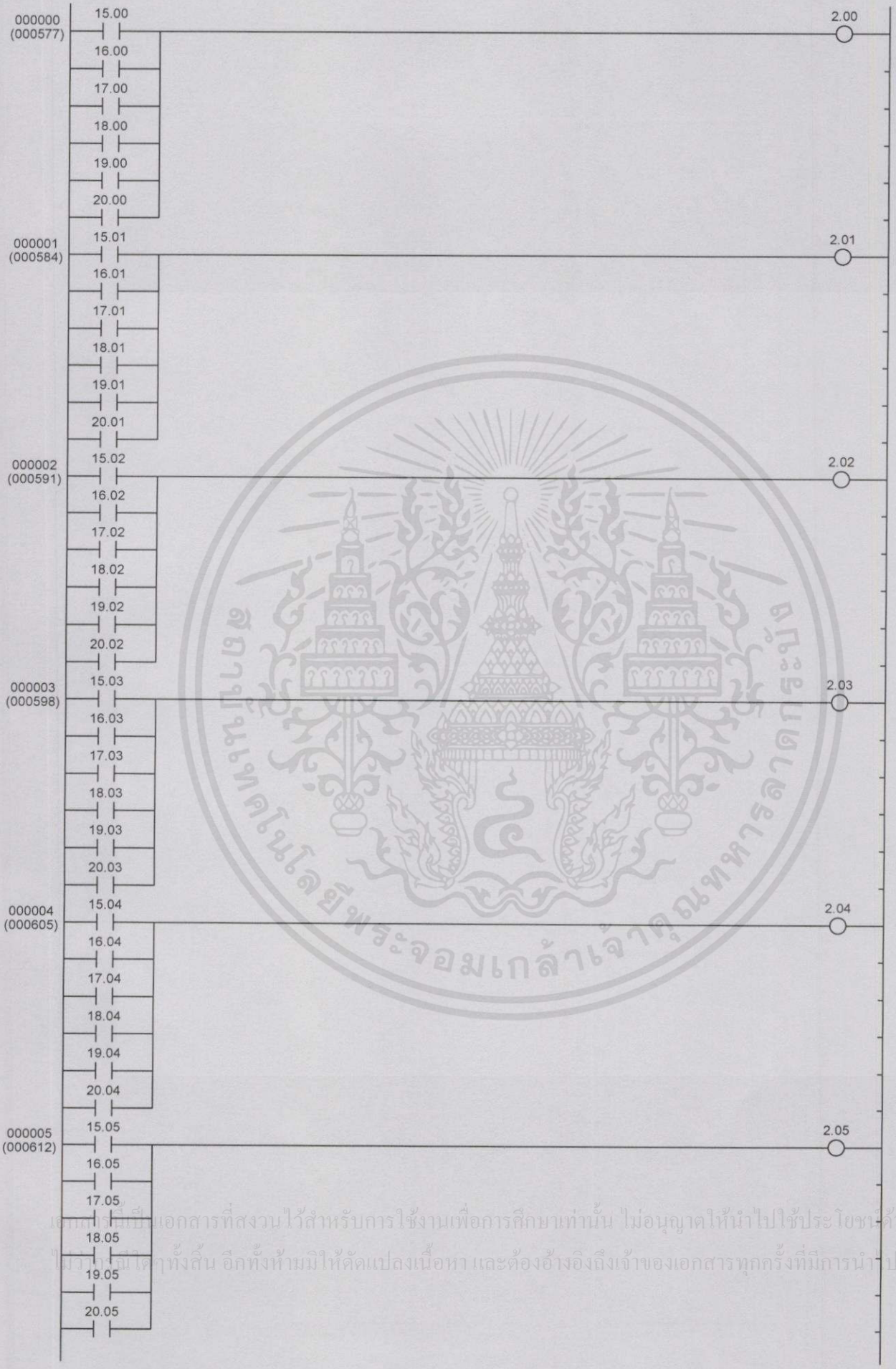
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



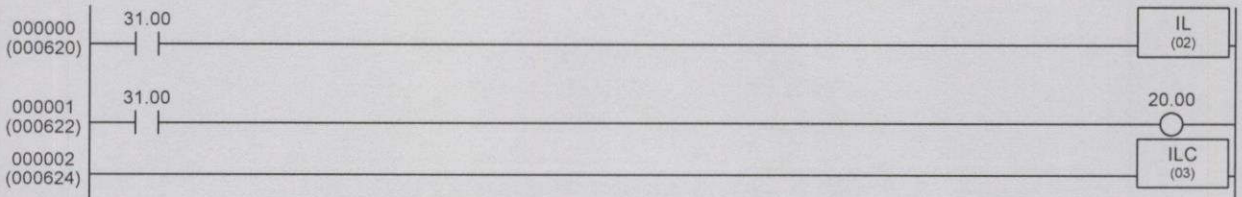
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



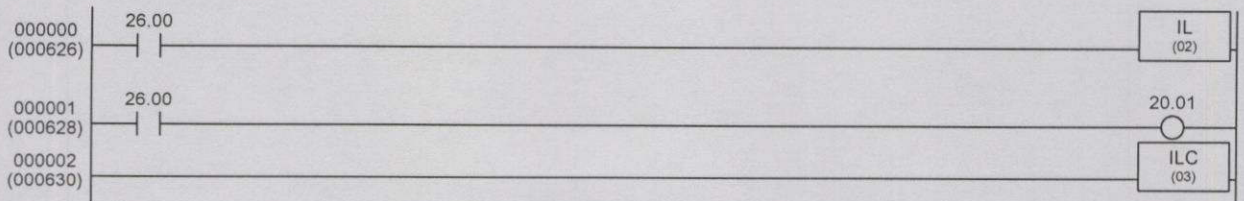
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



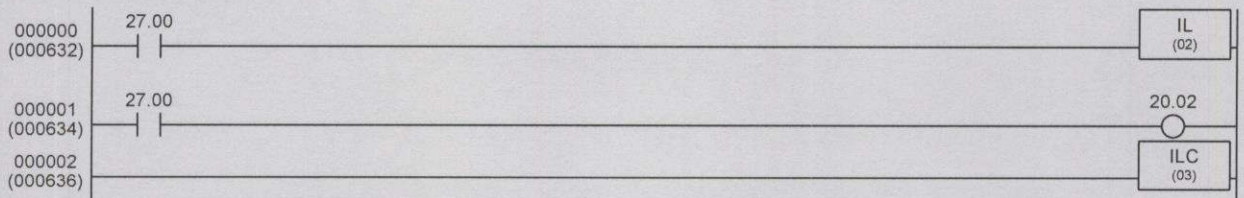
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



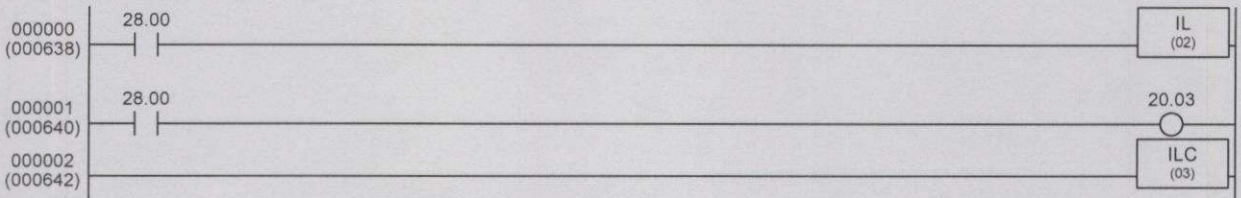
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

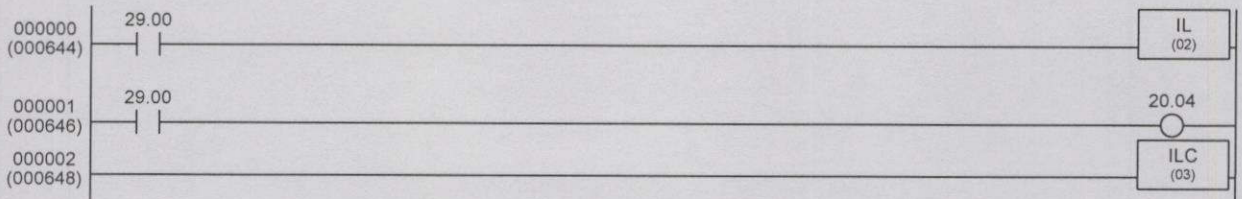


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

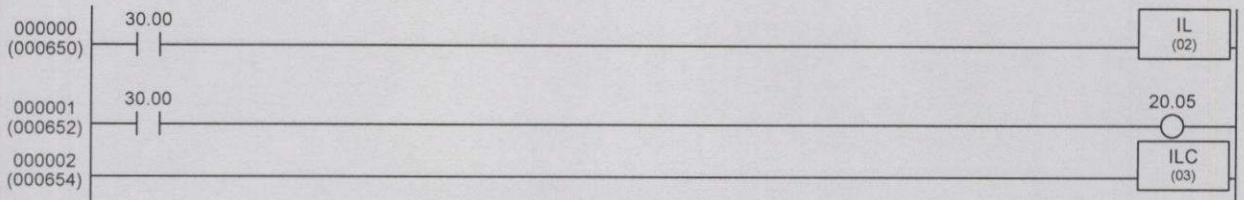


สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ค

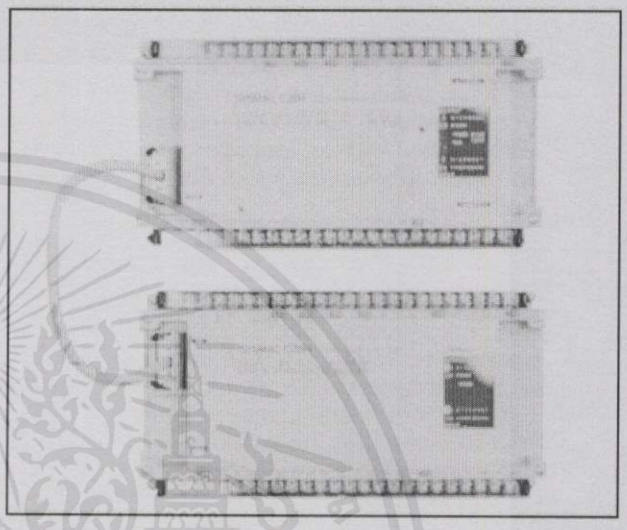
เอกสารที่เกี่ยวข้อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Data Sheet

RS Omron CH, 'mini-H' range programmable logic controller (PLC)

The Omron CH or 'mini-H' range is a highly specified range of compact, ruggedly designed and easy to use PLCs. The main features of the range include an in-built bi-directional RS-232 port, real time clock and a large number of data registers. These PLCs are ideal for small to medium I/O applications, or where the advanced functions are required.



Main system features:

- Bi-directional RS-232 ports on all CPUs
- Up to 240 I/O capacity
- In-built high speed counter
- Real time clock and calendar
- Programmable with PC based software
- NPN and PNP inputs as standard
- Individual outputs on a CPU or expansion unit can be changed from relay to transistor or triac as required
- 2000 data registers
- Execution time from 0.75μs per instruction
- Advanced programming and diagnostic features.

Section index

1. What is a PLC?
2. System configuration
3. How to choose CH range
   - 3.1 Advantages of the system
   - 3.2 Range overview
4. Main and expansion units
   - 4.1 Main units
     - 4.1.1 RS-232 port
   - 4.2 Expansion units
5. Programmers
   - 5.1 'LSS' programming documentation and monitoring software
   - 5.2 Advanced handheld programmer console/terminal
   - 5.3 Data access console
6. Accessories
   - 6.1 Communications modules
   - 6.2 Program memory chips
   - 6.3 Replacement outputs
   - 6.4 Input simulator
   - 6.5 CH operation manual
7. Dimensions
8. Programming
   - 8.1 Ladder diagram programming
   - 8.2 Symbols and what they mean
   - 8.3 Example circuits
   - 8.4 High speed counters
   - 8.5 Reversible drum counter
   - 8.6 Interlocks and jumps
   - 8.7 Shift register
   - 8.8 Subroutines
   - 8.9 Step sequence
   - 8.10 I/O Refresh
   - 8.11 Block instructions
   - 8.12 Failure alarm
   - 8.13 Resident real time clock and calendar
   - 8.14 Error history tables
   - 8.15 Scan time control
   - 8.16 Program header
9. Programming instructions
10. Glossary.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. What is a PLC?

A programmable logic controller is a system for controlling industrial and production processes, electromechanical equipment and automated assembly operations. This control function is performed by continuously monitoring the states of all the input devices connected to the controller (eg. proximity switches and mechanical contacts). Following the user instructions stored in the controller (known as the program), the states of all its output devices are set as appropriate (eg. relays, solid state relays or transistors). These output elements can in turn control the operation of an external system.

Due to the simplicity of entering and modifying the programmed instructions to suit the requirements of the process under control, the PLC is a truly versatile and flexible device that can be employed easily and efficiently to repeatedly control tasks that vary in nature and complexity.

A schematic diagram of the basic PLC control system is shown in Figure 1.

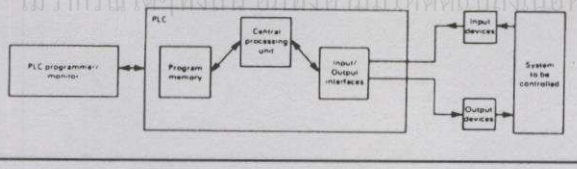
In this representation the central processing unit controls the overall operation of the system. Input devices may be switches, relay contacts, timers, solid state switches, proximity and limit switches, sensors, analogue to digital converters or other electronic circuits. The output devices may be external relays, heating elements, lights, alarms, sub-assemblies, electronic and electrical circuits, motors or other electromechanical devices.

To illustrate the advantages of using a PLC over a traditional electromechanical system consider a control system with 20 input/output points. This assembly could comprise 60-80 relays, some counters and/or timers and a great deal of wiring. This assembly would be cumbersome with a power consumption of 30-40VA. A considerable time would be required to design, build, test and commission the assembly and once it is in full working order any desired modifications, even of a minor nature, could require major hardware changes.

Alternatively a PLC can be used. A programmer is used to set up the control sequence as a software program with relays, counters and timers together with input and output devices. Programming and testing may be done off site before the PLC is connected up to actual input and output devices.

Programming will typically take 1-2 hours, any alterations to the control function of the system can be simply implemented within minutes by modifying the stored program. Other advantages over more conventional systems include compact size, low power consumption, fast response, versatility and lower overall cost. As the input/output requirements of a system increase, the advantage of using a PLC becomes much more significant due to the complexity of the equivalent hardwired system and its labour and material costs.

Figure 1 Schematic diagram



2. System configuration

A system is principally made up of the following parts:

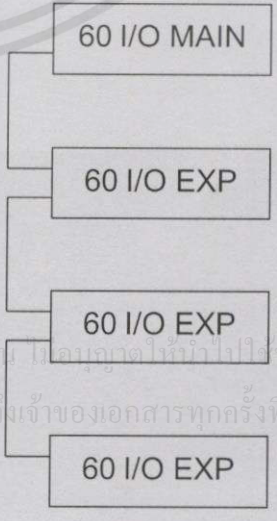
- 24Vdc powered main unit
- **Up to 3 extension units can be linked in series with the main unit.** Extension units may be placed in any order
- The main unit can be programmed in either LSS PC based software or by the handheld programmer. The LSS software may be connected using the built-in RS-232C port or the separate RS-232 module which sits on top of the main unit.

3. How to choose CH range

1. Assess the requirement for **input and output points** in the chosen application and whether outputs should be standard **relay** output, solid state relay (**triac**) or **transistor** output.
2. Choose **main unit** and **expansion units** to optimise the number of inputs and outputs. However, extra redundant I/O are often included for system additions and alterations. Up to three expansion units can be used with a main unit. Main unit and expansion units can be used in any combination. Units can be changed to transistor, triac or any combination using the spare outputs available.
3. Choose the most suitable **programming method**, advanced hand held programmer or LSS programming and monitoring PC based software.
4. Decide if the program should be stored on **EEPROM** or **EPROM memory**. No program memory is supplied with the main unit.

   Note: Program storage is recommended, but is not essential.
5. Decide if information needs to be displayed via the Omron **data access module** or a **message display** (see current RS Catalogue).
6. Refer to **CH Operation Manual** for a fuller explanation of CH devices and programming.

Maximum I/O configuration



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1 Advantages of the system:

- The 'mini-H' series has a **powerful, fast instruction set** which enables the range to be implemented efficiently, whilst maintaining the principle of ease of use. Instructions include simple 'send' and 'receive' commands for use with the two-way communications port, comprehensive data manipulation, maths and many other instructions normally only associated with much larger controllers – these instructions being processed from 0.75μs
- All of the 'mini-H' CPUs have the ability to **communicate on RS-232** to devices such as printers, man-machine interfaces, bar code readers via an 'in-built' port which can both send and receive information on this RS-232 link. ASCII characters can be stored in the 'mini-H' controller and sent to the required device using a simple instruction from the PLCs set. ASCII characters and strings can be read by the CPU using an equally simple instruction
- Amongst the features resident in the 'mini-H' software table are a **real-time clock** and **calendar** both of which are accessible by either the user program, programming devices or any equipment connected to the RS-232 port. When used in combination with facilities such as the 'send and receive' communication commands and resident history tables, these time and data functions offer unique logging features. Events can be initiated at specific points in time
- There are **2000** 16 bit **registers** resident in the 'mini-H' CPUs which not only offer the user the ability to perform data manipulation, but also system settings and monitoring can be performed in the program using these registers. For example, diagnostic and error data and system setting parameter can be stored in 'look-up' register tables.

3.2 Range overview

Table 1 **RS** Omron CH PLC range overview

Product	**RS** stock no.	Omron no.	Description and features
Main units	731-518	C20H C6DRDEV1	CPU 24Vdc power supply, RS-232 port, 12 24Vdc inputs, 8 relay contact outputs
	731-524	C28H C6DRDEV1	CPU 24Vdc power supply, RS-232 port, 16 24Vdc inputs, 12 relay contact outputs
	731-530	C40H C6DRDEV1	CPU 24Vdc power supply, RS-232 port, 24 24Vdc inputs, 16 relay contact outputs
	731-546	C60H C6DRDEV1	CPU24Vdc power supply, RS-232 port, 32 24Vdc inputs, 28 relay contact outputs
Expansion units	731-552	C20H EDRD	Expansion 24Vdc power supply with 12 24Vdc inputs, 8 relay contact outputs
	731-568	C28H EDRD	Expansion 24Vdc power supply with 16 24Vdc inputs, 12 relay contact outputs
	731-574	C40H EDRD	Expansion 24Vdc power supply with 24 24Vdc inputs, 16 relay contact outputs
	731-580	C60H EDRD	Expansion 24Vdc power supply with 32 24Vdc inputs, 28 relay contact outputs
Programming equipment	731-653	C200H PRO27ENL	Advanced programming console
	731-669	C200H DACO1ENL	Data access console
	731-675	C200H CN222	2m connecting cable for C200H DACO1ENL and C200H PRO27ENL
	731-647	LSS 3.5/5.25	PC based programming software for CK, CH and all other C-series PLCs
Communications modules	731-596	3G2A6 LK201EV1	RS-232 communications module for CH range
	731-603	3G2A6 LK202EV1	RS-422 communications module for CH range
Accessories	731-625	RAMH	RAM for all CH controllers, 6264
	731-619	EEROMH	EEPROM for all CH controllers, 28C64
	734-545	ROMIDB	EPROM, 27128
	734-494	G3SDZO1PUS24DC	Transistor module (NPN) for CK and CH
	734-501	G3S DZO1PPE24DC	Transistor module (PNP) for CK and CH
	734-517	G3S201PLPDUS24DC	Triac module for CK and CH
	734-567	C20K ETLO1	Simulator switches for C20K, C28K, C20H and C28H
Manual	731-631	CH MANUAL	Manual for CH PLCs

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Table 2 Comparison of CK and CH CPU characteristics

	CK	CH
Supply voltage	100 to 240Vac −15 to +10% 50/60Hz	24Vdc −15 to +10%
dc output	0.3A 24Vdc ±10%	–
Operative ambient temperature	0° to 55°C	0° to 55°C
I/O control method	Cyclical scan	Cyclical scan
Programming method	Ladder diagram	Ladder diagram
Instruction length	1 address/ instruction, 6 bytes/instructions	1 address/ instruction, 1 to 4 words/instruction
Number of instructions	49	142 (12 basic 130 special)
Execution time	10μs/instruction (average)	0.75 to 2.25μs
Memory capacity	1194 address (≈ 3K bytes)	2878 words
Memory type	RAM/EPROM	RAM/EPROM/ EEPROM

	CK	CH
Internal relays	136	3472
Special relays	16	136
Holding relays (battery backed)	160	1600
Temporary relays	8	8
Data memory (registers)	464	1000 read/write 1000 processor generated
Auxiliary relays	–	448
Link relays	–	1024
Timers/Counters	48	512
High speed counter	2kHz	2kHz
Battery life	5 years at 25°C	5 years at 25°C
Features	Self diagnostics program check	Self diagnostics program check
Maximum number of inputs/outputs	148	240

Table 3 General ratings

Supply voltage		24Vdc
Operating voltage range		20.4 to 26.4Vdc
Power consumption		C20H/C28H/C40H: 20W max.; C60H: 25W max.
Insulation resistance		20MΩ min. (at 500Vdc) between ac terminals and GR terminals
Dielectric strength		1000Vac 50/60Hz for 1 minute (between dc and GR terminals)
Noise immunity		1000 Vp-p , pulse-width: 100ns to 1μs , rise time 1ns
Vibration		10 to 35Hz, 1mm double amplitude, in X, Y, and Z directions; 2 hours each (when mounted on a DIN rail: 16.7Hz, 1mm double amplitude, in X, Y, and Z directions, 1 hour each)
Shock		10G in X, Y, and Z directions, 3 times each
Ambient temperature		Operating: 0° to 55°C Storage: −20° to 65°C
Humidity		35% to 85% (without condensation)
Grounding		Less than 100Ω
Construction		Conforms to IEC IP-30 (when panel-mounted)
Weight	CPU	C20H/C28H 1.2 kilograms max. C40H 1.3 kilograms max. C60H 1.8 kilograms max.
	I/O unit	C20H/C28H 1 kilogram max. C40H 1.1 kilograms max. C60H 1.6 kilograms max.
Dimensions (CPUs and I/O units)		C20H/C28H 250 × 130 × 64mm (W × H × D) C40H 300 × 130 × 64mm (W × H × D) C60H 350 × 140 × 70mm (W × H × D)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Table 4 CPU specifications

Control method	Stored program
I/O control method	Cyclical scan
Programming method	Ladder diagram
Instruction length	1 address/instruction, 1 to 4 words/instruction
Number of instructions	142 (12 basic instructions, 130 special instructions)
Execution time	Basic instructions 0.75 to 2.25μs
Memory capacity	2878 words
I/O bits	20 to 240 (00000 to 03915) The number of I/O bits depends on the number of Units
IR bits	3472 (0400 to 24615)
SR bits	136 (24700 to 25507) Normally ON, normally OFF, battery error, 0.1s clock, 02.s clock, 1.0s clock, etc.
HR bits	1600 (HR 0000 to 9915)
TR bits	8 (TR 0 to 7)
AR bits	448 (AR 0000 to 2715)
LR bits	1024 (LR 0000 to 6315)
Timer/Counters	512 TIM/CNT 000 to 511) TIMs 0 to 999.9s TIMHs 0 to 99.99s CNT 0 to 999 counts When using the high-speed counter, CNT 511 is the present value area.
High-speed counter	Count input 00000 Hardware reset input 00001 Maximum response frequency 2kHz Preset count range 0000 to 9999 Number of outputs 16
DM words	Read/Write 1000 words (DM 0000 through DM 0999) Read only 1000 words (DM 1000 through DM 1999) Words DM 0900 through DM 0999 and DM 1900 through DM 1999 are allocated as the system setting area.
Memory protection	The data in HR, AR, CNT, and DM areas is protected by a battery. If the battery is running low, discharged or removed, the data will be lost.
Battery life	5 years at 25°C Battery life is shortened at temperatures higher than 25°C. Replace the battery within one week when the ALM ERR LED blinks.
Self diagnostic features	CPU failure (watchdog timer) I/O bus failure Host link error Memory error Battery error, etc.
Program check	Program checked (executed at the beginning of a RUN operation) END instruction missing Instruction error The program can also be checked with a programming console or GPC. The program can be checked at three levels.
RS-232C interface	Communications: Half-duplex Sync: Start-stop Baud rates: 300, 600, 1200, 2400, 4800, 9600bps Transmission distance: 15m max.
Real-time clock (on specific models only)	Provides year, month, date, day, hour, minutes, seconds. Adjusts for leap year and backed up by battery. Accuracy: Within 30 seconds a month at 25°C.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Table 5 Input/Output specifications

CPUs and I/O units with relay contact outputs (with sockets)

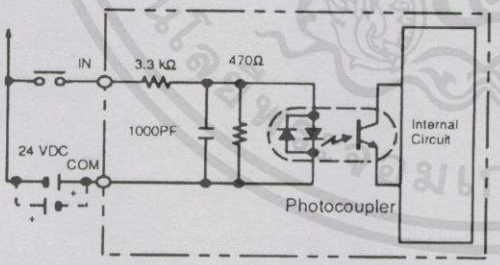
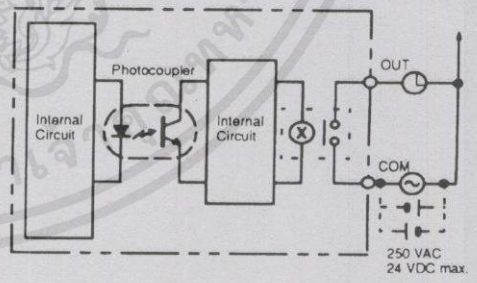
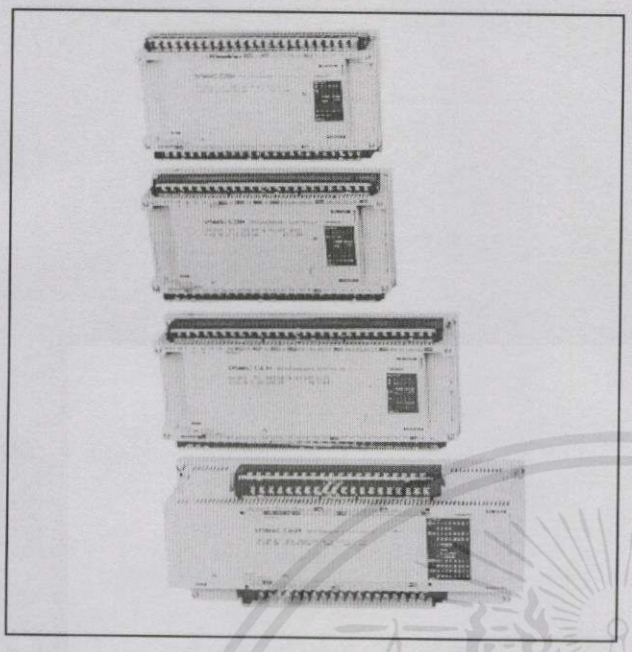
Item	dc input				Relay contact output			
Input voltage	24Vdc +10%/–15%				–			
Input impedance	3.3kΩ							
Input current	7mA typical at 24Vdc							
ON voltage	16Vdc min.							
OFF voltage	5Vdc max.							
ON delay time	2.5ms max. (input 00000, 00001 on CPUs: 0.15ms max.)				15ms max.			
OFF delay time	2.5ms max. (input 00000, 00001 on CPUs: 0.15ms max.)				15ms max.			
Number of circuits	C20H	C28H	C40H	C60H	C20H	C28H	C40H	C60H
	12pts 2pts/com 10pts/com 1 circuit each	16pts 2pts/com 14pts/com 1 circuit each	24pts 2pts/com 8pts/com 14pts/com 1 circuit each	32pts 2pts/com 14pts/com 16pts/com 1 circuit each	8pts 1pt/com 4 circuits 4pts/com 1 circuit	12pts 1pt/com 4 circuits 4pts/com 2 circuits	16pts 1pt/com 4 circuits 4pts/com 3 circuits	28pts 1pt/com 4 circuits 4pts/com 2 circuits 8pts/com 2 circuits
	When using the high-speed counter, the 2pts per common circuit is used for the high-speed counter input and hardware reset on the CPU.				–			
Maximum switching capacity	–				Resistance load 250Vac/2A (cos of phase angle = 1) or 24Vdc/2A for each pt, 4A/common (4pts/common) Inductive load 250Vac/0.5A (cos of phase angle = 0.4)/pt			
Minimum switching capacity	–				5Vdc, 10mA			
Relay durability	–				Resistance load: Electrical 300,000 times Mechanical 50,000,000 times Inductive load: Electrical 100,000 times Mechanical 50,000,000 times Use relay G6B-1174P-FD-US, 24Vdc			
Circuit configuration	 24Vdc power supply can be connected to a positive common or a negative common.				 A separate load power supply is required. Use relay G6B-1174P-FD-US, 24Vdc.			

Table 6 Transistor and triac specifications

Item	Transistor		Triac
Model	G3SD-Z01P (-PE)-US	G3SD-Z01P-PD-US	G3S-201PL-PD-US
Maximum switching capacity	0.3A at 5 to 24Vdc	0.6A at 5 to 24Vdc	0.6A at 85 to 250Vac
Minimum switching capacity	10mA at 5Vdc		10mA at 100vac/20mA at 200Vac
Leakage current	100μA max.		2mA at 100Vac max./5mA at 200Vac max.
ON response time	1.5ms max.		1.5ms max.
OFF response time	1.5ms max.		½ of load frequency + 1ms max.

4. Main and expansion units

4.1 Main unit



The 'mini-H' controllers consist of four different sizes of CPUs (20, 28, 40 and 60 I/O versions) which can be connected to up to three expansion blocks (again, 20, 28, 40 and 60 I/O).

Each CPU has a built-in RS-232 port which can be used for programming with Omron LSS PC software, or can be configured as a port which can perform two-way ASCII communications to devices such as printers, temperature controllers, bar-code readers, man-machine interfaces, data loggers, etc.

Communications is controlled by simple send and receive commands in combination with easily accessed data areas. When used in conjunction with the resident real-time clock and calendar the 'mini-H' provides an extremely capable, flexible control system.

Features

- An additional 'mini-H' RS-232 or RS-422 communications module can be added to facilitate communications to a host computer for programming or monitoring
- Built-in high speed counter with hardware and software reset facilities
- Real-time clock and calendar whose values can be accessed and changed if required; even caters for leap years
- Display messages can be changed to one of six languages – English, German, French, Italian, Spanish and Japanese
- High speed program execution from 0.75μs per instruction
- Same programming language as CK range and other Omron C-series PLCs (not sold by RS)
- CPU power supply 24Vdc
- Accepts NPN and PNP inputs as standard
- Removable terminal strips on all CPUs and expansion units
- Outputs can be configured such that a relay can be changed for a transistor or triac in the same unit – this feature is facilitated by 'plug-in' sockets and grouped commons
- On-line programming is possible via dedicated programmers or PC based software
- Advanced instruction set includes sub-routines, I/O refresh, step-sequencing, scan time and watchdog adjustments, jump to number and interlocks, block compare, shift register and word shift, block transfer, in-depth maths and data handling and various message instructions
- Unique error history table where records can be stored to show nature, time and date of events and errors that have occurred
- Full diagnostics and debugging facilities
- A 'key' command enables programming console keystrokes to be retained in the PLC memory and can be recalled and performed at a later stage without the console being present
- A standard RS-232 EPROM writer, eg. RS stock no. 617-232 can be connected to the CPU communication port for storing programs onto EPROM
- The programming console can also be used as a basic terminal with keys being designated as inputs to the CPU; these can be used in combination with a 32 character message instruction
- LED indication of all CPU and I/O status
- All maintenance parts – battery, relays – are easily accessed and changed if required.

Supplied with comprehensive instruction manual. See overview table for equivalent Omron part no.

4.1.1 RS-232 port

The in-built RS-232 port can be used in the following ways:

1. As a programming port for Ladder Support Software where full programming, monitoring and diagnostic functions can be performed, including full on-line editing of user programs. Up to 32 'mini-H' PLCs can be connected in a network to a computer by simply converting the signal from RS-232 to RS-422 – this means that one computer could access and of the 32 controllers via LSS from one point.
2. By using the same principle outlined in the section above, up to 32 PLCs could be connected to a computer which is running a SCADA package (Supervisory control and data acquisition) for real-time data display and manipulation.
3. The RS-232 ports on the 'mini-H' controllers mean that a PLC to PLC link can be established such that information is being exchanged between the PLCs without the need for I/O wiring or supervisory computers. This can be particularly useful for example where individual 'mini-Hs' are controlling machines on a production line, and timing or product information is required to be sent to other machines along the production line.
4. A very powerful feature of the RS-232 ports is that they can be set to be bi-directional, ASCII ports whereby they can communicate to RS-232 devices such as printers, bar-code readers, temperature controllers, man-machine interfaces, data loggers, recorders, etc.

   The control for the communications to and from these devices is performed by two straightforward commands called Port-in (PIN) and port-out (POUT); the ASCII strings of data which are being sent out and read in are stored in the 'mini-H' data word memory area (there are 2000 of these registers).

As an example, you might want a printer to print-out every time a pumped stopped in a process – printing the time and date of the occurrences. All you have to do is to store the message 'Pumped Stopped' in an ASCII format in data memory word registers, for instance DM0000 to DM0005 (you can convert the characters you wish to store manually using an ASCII table, or LSS will do it automatically for you). Having stored your message, the port-out command (POUT) will instruct all of the ASCII data in DM0000 to DM0005 to be sent to the RS-232 port (to which the printer would be connected). When used in combination with the resident Realtime Clock, you can reference the printout to time and date.

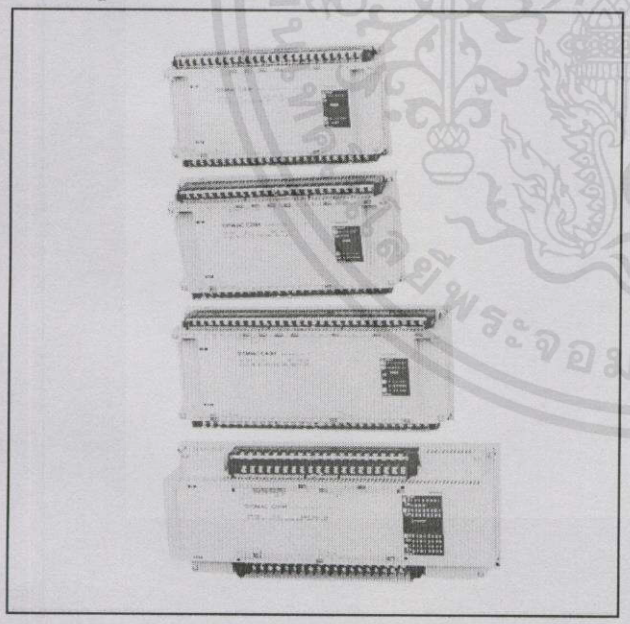
This simple principle is used to read in data from devices such as bar-code readers; you simply instruct the CPU where to store the incoming data.

The power, flexibility and ease of use of this port give unprecedented control capability to controllers of such a size and cost.

5. The RS-232 port can also be used to download user programs and data register contents to commercially available RS-232 PROM writers.

RS stock no.	Type
731-518	20 I/O relay
731-524	28 I/O relay
731-530	40 I/O relay
731-546	60 I/O relay

4.2 Expansion units



The 'mini-H' family of controllers is easily expanded by adding up to **3 expansion I/O modules** – these modules can be added to any CPU or other expansion in any order. When an expansion module is added it is simply addressed from the next available word of I/O in the memory map. Being the same physical size as the CPU with the I/O count, the expansion modules maintain the 'mini-H's' very compact design.

Features

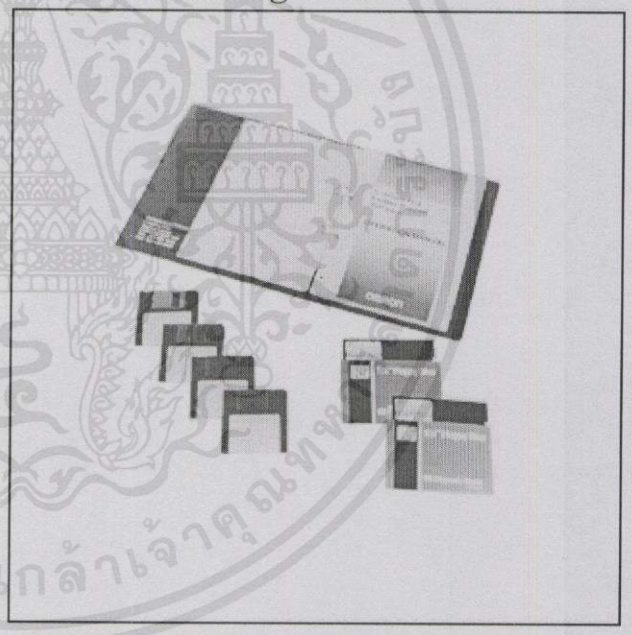
- Any expansion can be connected to any CPU or other expansion module
- Accepts NPN or PNP inputs as standard
- Outputs can be changed whereby relays can be replaced by transistors or triacs in the same unit
- Same slim, compact design and dimensions as CPUs
- Removable terminal strips on all modules
- LED indication of all I/O status
- 24Vdc power supply
- Supplied with connection cables.

See overview table for equivalent Omron part no.

RS stock no.	Type
731-552	20 I/O relay
731-568	28 I/O relay
731-574	40 I/O relay
731-580	60 I/O relay

5. Programmers

5.1 'LSS' programming, documentation and monitoring software

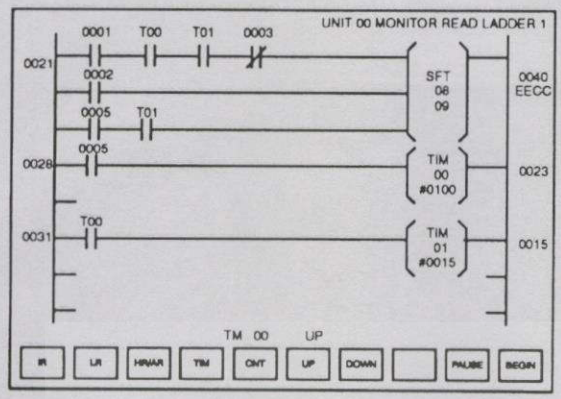


Ladder Support Software (LSS) provides the ability to program CK, CH and all other Omron C-series PLCs (not available through RS) using a personal computer.

It has the advantage of enabling programs to be written on a computer in either ladder diagram or mnemonic code and offers extensive documentation facilities so that once a program has been written and commissioned, it can be documented so that others looking at a machine or process for the first time have a clear indication of procedures, etc.

The 'built-in' RS-232 ports on all the 'mini-H' controllers can be used for connection to a computer for a programming via Ladder Support Software (LSS) and then used for communication to a SCADA (Supervisor control and data acquisition) package, or converted to a bi-directional ASCII RS-232 port (this feature is explained further in this data sheet). Alternatively a separate RS-232 communications module can be used.

Typical screen layout



Features

Software library for easy management

- Each data disk can contain a library of up to 255 program files with a file name directory, serving to enable standardisation and program development.

Finger-tip switching between ladder diagrams and mnemonic code

- Switch quickly between ladder diagrams and the mnemonic code
- Display of up to 50 instruction lines and line comments
- Immediate programming (deletion or addition) capability.

ASCII-based DM area editor

- Display of up to 160 words at once
- ASCII-code input of messages, word copying, and input of same contents to designated word range
- Transfer of DM area contents to PC.

Monitoring

- Any instruction instantly accessed and monitored
- ON bits displayed in yellow; OFF bits, in blue; and comments, in green
- Change between modes and change set values.

Multibit monitoring

- Monitor any PC data by bit or word with a 20 bit maximum
- Set or reset bits, or change present values.

Ladder diagram output

Printing out ladder diagrams or reference lists greatly increases programming and debugging efficiency. I/O cross references are provided with ladder diagrams to quickly show the application of bits, data words, timers, and counters.

Programming in ladder diagram symbols

- Up to 22 lines each containing a maximum of nine inputs and one output can be written at one time
- Display of up to 11 lines without comments or six lines with comments
- Deleting, moving, or copying designated program blocks
- Program section saves
- Copying comments from other programs.

Check program syntax

Three checking levels are available for new or altered programs.

Comment editing on I/O list

- On-screen editing of comments for all bits, including overwrites and searches
- I/O status readouts
- Advanced comment input to facilitate programming (comments provided when bits are accessed).

PLC to PC connections

Programs are transferred to the PLC through the RS-232C communications module, RS stock no. 731-978, ordered separately, or the inbuilt RS-232C communications port.

Package consists of

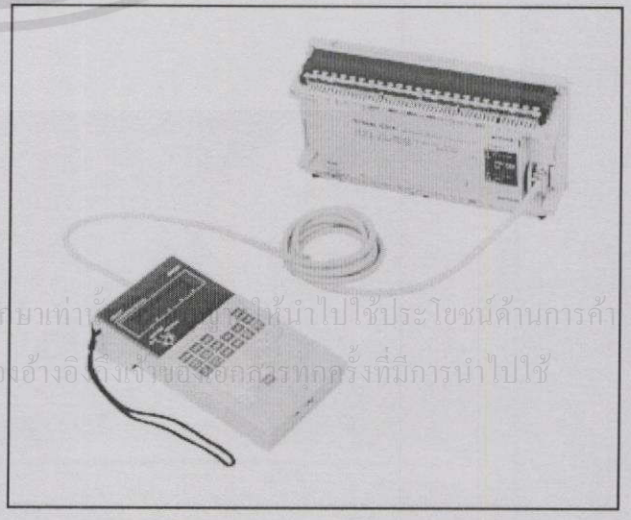
- Manual
- 3½in and 5¼in double sided, double density disks.

Minimum computer requirements

- 100% IBM XT, AT compatibility
- DOS, 3.2 or above
- 640K byte RAM
- 3M byte hard disk
- Serial communications port
- Monochrome CGA, EGA or VGA display
- Floppy disk drive.

RS stock no.	Type
731-647	LSS software

5.2 Advanced handheld programming console/terminal



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของลิขสิทธิ์ทุกครั้งที่มีการนำไปใช้

The handheld programming console can be used as a method of inputting programs into the CK or CH processor via mnemonic code and as a very fast monitoring and debugging tool during commissioning and fault-finding.

It is recommended that this programming console is used when it it intended to use the 'built-in' RS-232 port for communications to other devices – this will allow for efficient and flexible set-up procedures relating to this port.

In addition to its use as a programming, de-bugging and setting tool, when used with CH PLCs, this console can be used in a unique 'Terminal Mode' – this is where it relinquishes its programming capability and is used purely as a terminal where it can display 32 character messages (which are stored in the CPU), and offer operator interfacing by allowing its keyboard to be 'mapped' into the PLC; this mapping feature means that certain keys can be seen simply as inputs by the CPU.

There are two instructions resident in the CH CPU which are particularly useful when in Terminal Mode, they are:

1. Long Message – This command simply tells the CPU where to look for a string of ASCII characters making up a message in Data memory words in the memory map. When operated, this instruction sends these ASCII characters to the console's display automatically – the console has the ability to store subsequent messages in a buffer area.
2. Key Input – This command allows a console key sequence to be registered in data memory words, and to be recalled by maybe one of the 'mapped' key inputs – therefore, rather than an operator remembering a long key sequence, he simply presses a key on the console which has been mapped, which in turn operates the required key sequence (rather like a macro on a computer).

   The console is changed to and from Terminal Mode by a simple key sequence (again, this could be mapped if required). To enable customisation of the console, there is the facility to place overlays on the keypad with relevant text for individual applications.

Features

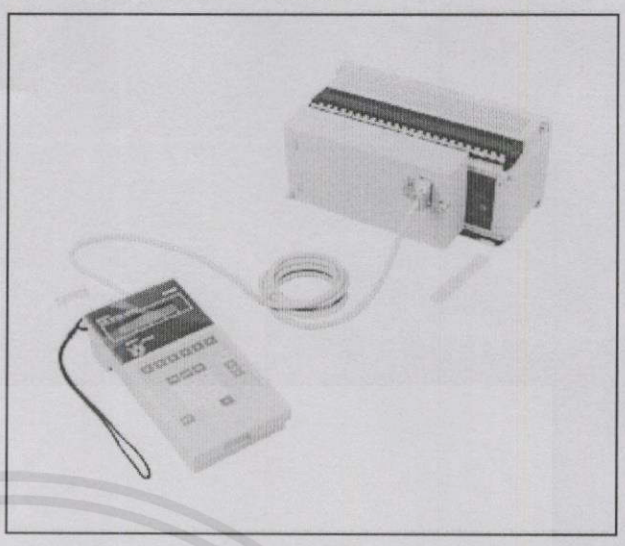
- Performs full programming, monitoring and diagnostic operations
- ASCII, HEX or BCD monitoring
- Word or bit monitoring and setting
- Can be used in terminal mode where keys can be designated as inputs and messages can be sent to the 2 line LCD
- 2m cable connects console to the PLC.

Supplied with instruction leaflet.

See overview table for equivalent Omron part no.

RS stock no.	Type
731-653	Console
731-675	2m cable

5.3 Data Access Console



The Data Access Console is a device which only allows access to timers, counters and data registers for modification; it will not facilitate the modification of the user program at all. It is ideal for use as an operator setting console where certain timers, counters and data registers on a machine or process may need to be changed periodically, but where program alteration (even by mistake) is not possible.

In addition to its data modification, the Data Access Console (DAC) can be used as a basic terminal rather like the programming console operating in terminal mode, allowing keyboard mapping and 32 character messages to be displayed. As with the programmer, an overlay can be placed on the DAC to customise its operation.

A very useful feature on the DAC is its three level security operations; there are two different security levels which are entered via a keyswitch which allow varying levels of alteration; within these two security levels, there is a third level predetermined in software, which only allows only certain timers and counters, etc. to be modified. For example, on a particular operation the operator may need to have access to timers and counters, but only to maybe three or four in each case – this is where the third security level can be used to 'block' out all other timers and counters.

The connecting cable is required.

Supplied with Data Access Console operation guide.

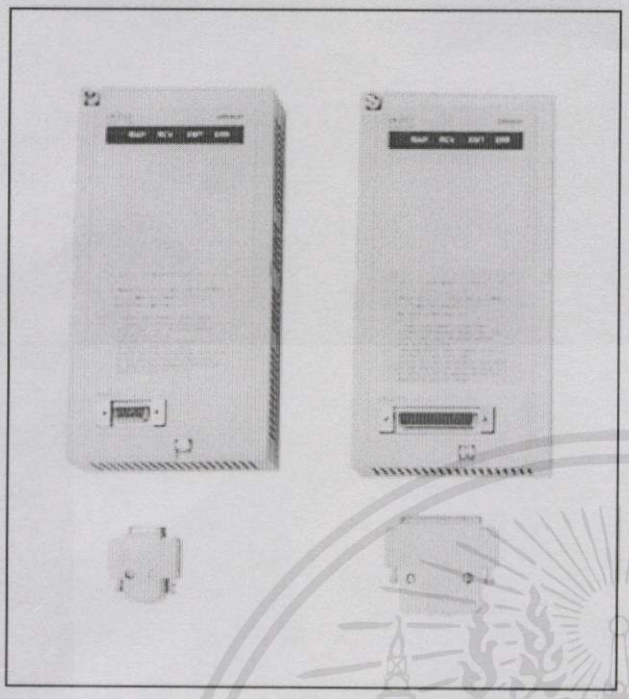
See overview table for equivalent Omron part no.

RS stock no.	Type
731-697	Data access console
731-675	2m cable

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งไม่อนุญาตให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6. Accessories

6.1 Communications module

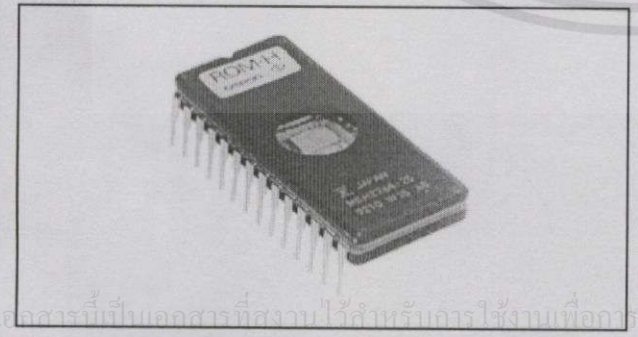


In addition to the 'in-built' RS-232 ports on the 'mini-H' processors, it is possible to add another RS-232 or RS-422 port (ordered separately) and have these ports running in conjunction with the existing ports. These additional ports are for connection to a Host Computer for running LSS, a SCADA package or can even be used to 'communicate' to the PLC with a user's own computer program. With the two ports working at the same time (although totally independent of each other) it is possible to have two computers accessing one PLC for different purposes, or using the same principle a computer could be accessing the PLC which in turn is communicating to a device such as a bar-code reader.

See overview table for equivalent Omron part no.

RS stock no.	Type
731-596	RS-232
731-603	RS-422

6.2 Program memory chips



The program memory is stored in either plug in RAM, EPROM or EEPROM memory. There is no in-built memory in the CH range.

RAM memory is useful for short term retention but the program will be lost on power off and battery fail. EEPROM memory is useful for long term retention where modifications can be made easily. EEPROM memory is read and written to in the same easy way as RAM memory. **EEPROM memory cannot be used with the CK range.**

EPROM memory provides long term retention and program protection. the program can only be altered by erasing the EPROM using an EPROM eraser and writing to it with a special EPROM writer. All chips have 8K word memories.

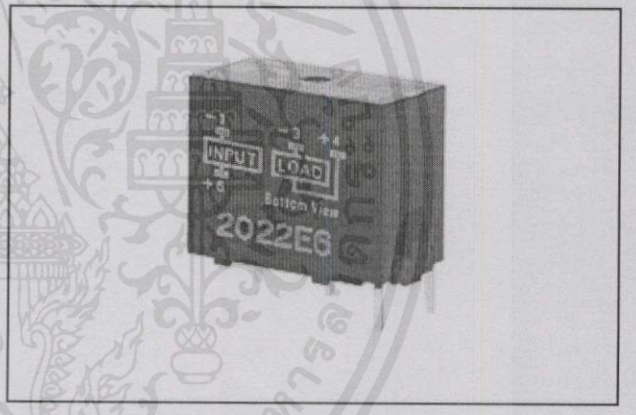
Chip installation

The memory chip is installed by unscrewing the top of the main unit and installing in a socket on the PCB under the cover.

See overview table for equivalent Omron part no.

RS stock no.	Type
734-845	EPROM
731-619	EEPROM
731-625	RAM

6.3 Replacement outputs

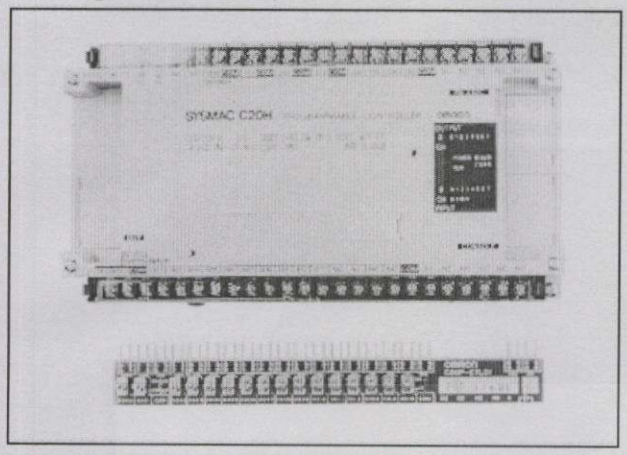


All relay output versions of the CH range have their relays mounted into sockets. This not only allows for easy maintenance should an output become faulty, but also enables individual outputs to be removed and exchanged for another switching device, transistor or triac, which may have more suitable switching characteristics for a particular application. A tool for removing the outputs is conveniently housed inside the cover of the CPU. Now, because the outputs terminals of the CPUs are arranged so that they are served by various common terminals (some output terminals being served by individual commons – see wiring details), it is possible to have the majority of outputs being switched by relay contacts, some being switched at 24Vdc via transistors (NPN or PNP) and others being switched at 110Vac by triacs. This tailoring of output circuits allows for very flexible system design and can negate the need for future replacement of outputs because loads are being switched by correctly matched devices.

RS stock no.	Type
734-494	NPN transistor
734-501	PNP transistor
734-517	Triac

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า การแสวงหากำไร หรือการอื่นใดที่ไม่ได้รับอนุญาต ต้องขออนุญาตก่อนการใช้งานเอกสารทุกครั้งที่มีการนำไปใช้

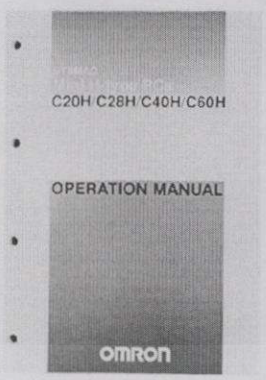
6.4 Input simulator



The CH series uses the same simulator input switches as the CK range. Available for 20 and 28 I/O main units.

RS stock no.	Type
734-567	C20K-ETL01

6.5 CH Operation Manual



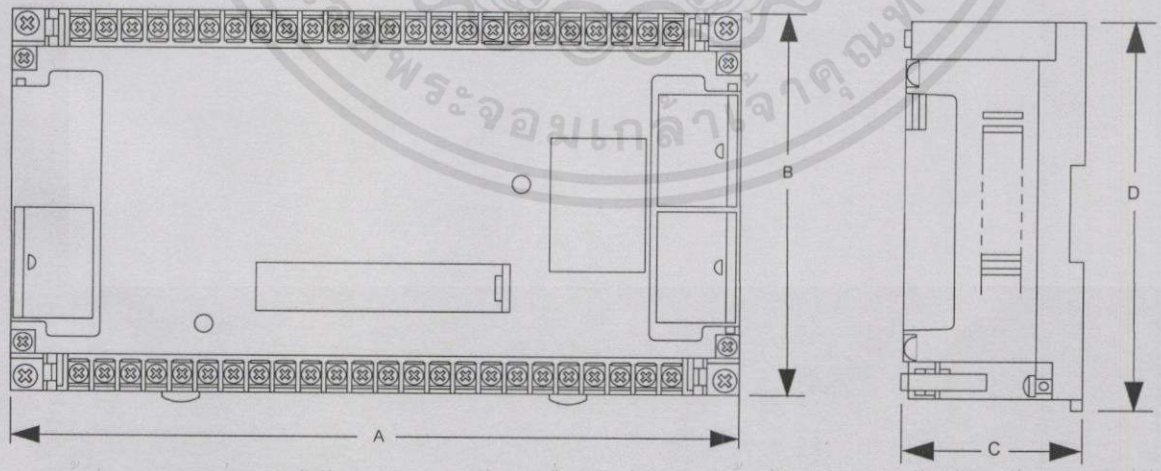
The operation manual:

- Provides an introduction to Omron PLCs and ladder diagram programming
- Looks at the hardware considerations in configuration
- Explains the memory areas
- Explains how to write and enter programs
- Describes the instruction set
- Explains the program execution timing so input and output signals occur as required
- Explains program debugging and execution
- Describes the flexible RS-232C interface
- Provides information on troubleshooting.

RS stock no.	Type
731-631	CH operation manual

7. Dimensions

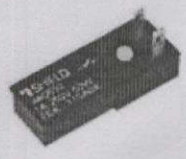
Main units

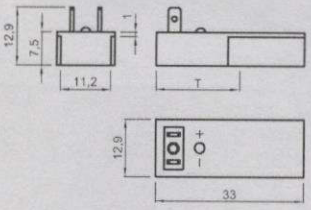


Model	A	B	C	D	Weight
C20H	250mm	130mm	64mm	134mm	1.2 kg
C28H	250mm	130mm	64mm	134mm	1.2 kg
C40H	300mm	130mm	64mm	134mm	1.3 kg
C60H	350mm	140mm	70mm	144mm	1.8 kg

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

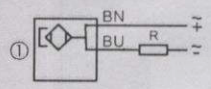
FEK





High switching power. Long electrical life.
Mountings are available for all the cylinders on the market.
Standard sensitivity suitable for most of cylinders on the market.
Other sensitivities on demand
Body material: Polyamide.
Centre of the sensitive area: reed, T = 14 mm

Alta potenza di commutazione.
Lunga vita elettrica.
Sistema di montaggio disponibile per tutti i cilindri sul mercato.
Sensibilità standard adatta alla maggior parte dei cilindri in commercio. A richiesta altre sensibilità.
Materiale del corpo: Poliammide.
Centro della zona sensibile: reed, T = 14 mm

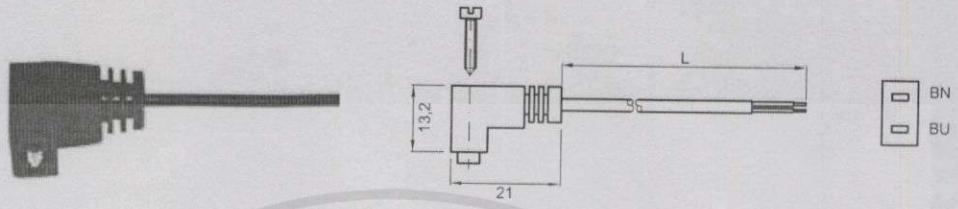
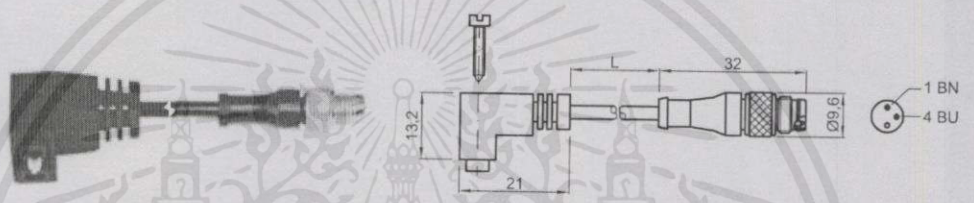


CE

	electronic (magneto-resistance)						reed											
wiring diagram collegamento																		
switching logic | tipo contatto	-	-	-	-	-	-									-	-	-	-
output | uscita	-	-	-	-	-	-	①	①	①	①	①	①	①	①	-	-	-	-
	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
wires | fili	-	-	-	-	-	-	2	2	2	2	2	2	2	2	-	-	-	-
led colour colore led																		
signal | segnale	-	-	-	-	-	-	-	-	-	yellow	-	-	-	yellow	-	-	-	-
supply | alimentazione	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
ratings dati elettrici																		
ac [V]	-	-	-	-	-	-	3...250	3...250	3...250	3...250	3...250	3...130	3...130	3...130	-	-	-	-
dc [V]	-	-	-	-	-	-	3...250	3...250	3...250	3...250	3...250	3...130	3...130	3...130	-	-	-	-
voltage drop | caduta tensione **@10mA** [V]	-	-	-	-	-	-	0,1	0,1	0,1	2,7	0,1	0,1	0,1	2,7	-	-	-	-
max current | corrente max [mA]	-	-	-	-	-	-	500	500	500	500	1000	1000	1000	300	-	-	-	-
max power | potenza max [W]	-	-	-	-	-	-	50	50	50	50	50	20	20	20	-	-	-	-
protection protezione																		
degree | grado **(IEC 60529)**	-	-	-	-	-	-	IP65	IP65	IP65	IP65	IP65	IP65	IP65	IP65	-	-	-	-
overcurrent transients | transitori extracorrente	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
overvoltage transients | transitori extra tensione	-	-	-	-	-	-	-	-	●	●	-	-	●	●	-	-	-	-
polarity reversal | inversione polarità	-	-	-	-	-	-	●	●	●	●	●	●	●	●	-	-	-	-
short circuit | corto circuito	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
corrosion resistance | resistenza corrosione	-	-	-	-	-	-	3	3	3	3	3	3	3	3	-	-	-	-
explosion | esplosione (atex)	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
welding disturbance | disturbi da saldatura	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
other data altri dati																		
temperature | temperatura [°C]	-	-	-	-	-	-	-20+70	-20+120	-20+70	-20+70	-20+70	-20+120	-20+70	-20+70	-	-	-	-
housing material | materiale custodia	-	-	-	-	-	-	PP	PA	PP	PP	PP	PA	PP	PP	-	-	-	-
switching time | commutazione [ms] on	-	-	-	-	-	-	0,5	0,5	0,5	0,5	0,5	0,5	0,5	0,5	-	-	-	-
switching time | commutazione [ms] off	-	-	-	-	-	-	0,1	0,1	0,1	0,1	0,1	0,1	0,1	0,1	-	-	-	-
circuit description | descrizione circuito	-	-	-	-	-	-	100A00	100A00	101A00	110A0T	700C00	800C60	801C60	810C6T	-	-	-	-
cable cavo 4-6 — material / colour / max temp. / description — part number | codice ordine																		**A**
pvc — ral 9005 — black nero — 90° — N107	-	-	-	-	-	-	F2100	F2125	F2101	F2102	F2103	☺	F2104	F2105	-	-	-	-
pvc — ral 9005 — black nero — 90° — N108	-	-	-	-	-	-	☹	☹	☹	☹	☹	☹	☹	☹	-	-	-	-
pvc CEI 20-22 II — ral 7035 — grey grigio — 90° — N111	-	-	-	-	-	-	F2141	☺	F2142	F2143	F2144	☺	F2145	F2146	-	-	-	-
pur — ral 9005 — black nero — 80° — C145	-	-	-	-	-	-	☹	☹	☹	☹	☹	☹	☹	☹	-	-	-	-
pur — ral 7001 — grey grigio — 80° — C144	-	-	-	-	-	-	☹	☹	☹	☹	☹	☹	☹	☹	-	-	-	-
armoured stainless steel armato inox — 120° — S131	-	-	-	-	-	-	☺	F2115	☺	☺	☺	F2118	☺	☺	-	-	-	-

? choose your product in three easy steps
scegli il tuo prodotto in tre comodi passaggi

A + B + C = F0000 00 3000
(part number composition example | esempio composizione codice d'ordine)

connector type | tipo connettore		part number | codice ordine **B**
without connector, only switch senza connettore, solo interruttore		00
CK 00		41
CK00 + RELM M8 straight 24V suitable for distribution box M8 diritto 24V adatto per scatole di distribuzione		47

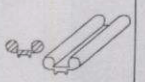
NOTE | NOTA

- **preferred items** — articoli preferenziali
- **not possible product** — prodotto non possibile
- **page reference** — pagina di riferimento
- **only on request** — solo a richiesta

BN = **brown** marrone | BU = **blue** blu | BK = **black** nero

cable length | lunghezza cavo (L) other special lengths available on request altre lunghezze speciali disponibili a richiesta	part number | codice ordine **C**
without connector | senza connettore	0000
2000 mm	2000
3000 mm	3000
6000 mm	6000
10000 mm	MT10
15000 mm	MT15

clamps and brackets | fascette e staffe

										accessories | accessori	
FA 21	4-1	FA 44	4-2	FA 55	4-4	FA 43	4-2			 FA 01	4-1
FA 23	4-1	FA 45	4-2								
FA 45	4-2	FA 49	4-3								

round | rotondo
tie-rod | tirante
mickey | estruso
rail | binario

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้