สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

# กระบวนการจำลองสำหรับพีแอลซี

# PROCESS CONTROL SIMULATOR FOR PLC EXPERIMENT

T119465

นายชนัญญู ตรงชื่น
นายศิริพงษ์ วงศ์ระพี
นายสุธี ภู่พูลทรัพย์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เลขหมู่.....
เลขทะเบียน 119465
วัน,เดือน,ปี - 8 S.A. 2554

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

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิชาวิศวกรรมการวัดคุม
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2553

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# PROCESS CONTROL SIMULATOR FOR PLC EXPERIMENT


CHANUNYU TRONGCHEUN

SIRIPONG WONGRAPEE

SUTEE PHUPHOONSUB


A THESIS SUBMITTED IN PARTIAL FULFILLMENT

OF THE REQUIREMENT FOR THE DEGREE OF

BACHELOR OF ENGINEERING IN INSTRUMENTATION ENGINEERING

FACULTY OF ENGINEERING

KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

ACADEMIC YEAR 2010


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**สาขาวิชาวิศวกรรมการวัดคุม**

**คณะวิศวกรรมศาสตร์**

**สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง**

**ใบรับรองปริญญานิพนธ์**

.....

**หัวข้อปริญญานิพนธ์** กระบวนการจำลองสำหรับพีแอลซี

PROCESS CONTROL SIMULATOR FOR PLC EXPERIMENT

| | | |
|---|---|---|
| **นักศึกษาผู้จัดทำ** | นายชนัญญ ตรงชื่น | รหัสนักศึกษา 50010296 |
| | นางศิริพงษ์ วงศ์ระพี | รหัสนักศึกษา 50011563 |
| | นายสุธี ภู่พูลทรัพย์ | รหัสนักศึกษา 50011723 |
| **ปริญญา** | วิศวกรรมศาสตรบัณฑิต | |
| **สาขา** | วิศวกรรมการวัดคุม | |
| **ปีการศึกษา** | 2553 | |

| อาจารย์ผู้ควบคุมปริญญานิพนธ์ | ลายมือชื่อ |
|---|---|
| รองศาสตราจารย์ วิริยะ กองรัตน์ | |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

| | | |
|---|---|---|
| **หัวข้อปริญญานิพนธ์** | กระบวนการจำลองสำหรับพีแอลซี | |
| | PROCESS CONTROL SIMULATOR FOR PLC EXPERIMENT | |
| **นักศึกษาผู้จัดทำ** | นายชนัญญ ตรงชื่น | รหัสนักศึกษา 50010296 |
| | นายศิริพงษ์ วงศ์ระพี | รหัสนักศึกษา 50011563 |
| | นายสุธี ภู่พูลทรัพย์ | รหัสนักศึกษา 50011723 |
| **อาจารย์ที่ปรึกษา** | รองศาสตราจารย์ วิริยะ กองรัตน์ | |
| **ปีการศึกษา** | 2553 | |

## บทคัดย่อ

ในบทความนี้เป็นการจัดทำโปรแกรมจำลองกระบวนการอุตสาหกรรมควบคุมด้วย PLC แสดงผลการทำงานทางจอคอมพิวเตอร์ เป็นโปรแกรมจำลองการทำงานกระบวนการวัดระดับของ Tank โดยใช้โปรแกรม Microsoft Office Visio 2007 ออกแบบสร้างภาพกราฟฟิคจำลองฉากหลังของกระบวนการ และโปรแกรม Delphi 5 ออกแบบสร้างภาพกราฟฟิคจำลองการเคลื่อนที่ของกระบวนการ เพื่อควบคุมโปรแกรมจำลองการทำงานกระบวนการวัดระดับ และโปรแกรมควบคุมภาพจำลองการทำงาน แล้วนำมาต่อทดสอบให้เป็นไปตามเงื่อนไขที่ถูกกำหนดจากผู้สอน โดยโครงงานนี้สามารถนำไปใช้เป็นชุดประลองการทำงานของ PLC เพื่อเพิ่มทักษะในการศึกษาวิชา PLC นำหลักการจำลองควบคุมการทำงานของกระบวนการด้วย PLC ไปใช้ในการออกแบบโปรแกรมควบคุม และทดสอบโปรแกรมที่ออกแบบได้ เป็นการทดสอบโปรแกรมควบคุมก่อนนำไปประยุกต์ใช้กับงานจริง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

| | |
|---|---|
| **Thesis Title** | Process Control Simulator For PLC Experiment |
| **Authors** | Mr. Chnunyou Tongcheun |
| | Mr. Siripong Wongrapee |
| | Mr. Sutee Phuphoonsub |
| **Thesis Advisor** | Assoc. Prof. Viriya Kongratana |
| **Year** | 2010 |


## ABSTRACT

This project presents a simulation of industrial plant process which controlled by PLC. The result is shown via computer monitor. The simulation of level measurement in the tank is used Microsoft office visio 2007 for background graphic. While the Delphi 5 is used to simulate the movement of the process. This project can be implement for student training in order to simulate the function of PLC. Therefore, the student will be get more experience and skill, before work in the real world.



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# กิตติกรรมประกาศ

ปริญญานิพนธ์ฉบับนี้สำเร็จลุล่วงได้ด้วยดีเพราะได้รับความกรุณาจาก รศ.วิริยะ กองรัตน์ และ รศ.ดร.ทวีพล ซื่อสัตย์ ที่คอยให้ความอนุเคราะห์อุปกรณ์ เครื่องมือต่างๆ ในการทำปริญญานิพนธ์นี้และให้คำแนะนำคำปรึกษาตลอดจนแนวความคิดต่างๆ ในการปรับปรุงแก้ไขงานและการทดลอง ซึ่งผู้จัดทำขอกราบขอบพระคุณเป็นอย่างสูง

ขอขอบคุณ นายอภิชาต กิ่งเกล้า ที่ได้ให้ความช่วยเหลือในด้านการเขียนโปรแกรมพีแอลซีของการกระทำปริญญานิพนธ์นี้

และที่ลืมเสียมิได้คือ ขอกราบขอบพระคุณ คุณพ่อ คุณแม่ อันที่รักยิ่ง ที่สนับสนุนและเป็นแรงบันดาลใจในการทำปริญญานิพนธ์ฉบับนี้

**คณะผู้จัดทำ**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# สารบัญ

หน้า

**บทคัดย่อภาษาไทย** ..... I
**บทคัดย่อภาษาอังกฤษ** ..... II
**กิตติกรรมประกาศ** ..... III
**สารบัญ** ..... IV
**สารบัญตาราง** ..... VI
**สารบัญรูป** ..... VII

**บทที่ 1 บทนำ** ..... 1
1.1 ความสำคัญของปริญญานิพนธ์ ..... 1
1.2 วัตถุประสงค์ของปริญญานิพนธ์ ..... 1
1.3 ขอบเขตของปริญญานิพนธ์ ..... 2
1.4 ขั้นตอนการศึกษา ..... 2
1.5 ประโยชน์ที่คาดว่าจะได้รับ ..... 2

**บทที่ 2 ทฤษฎี** ..... 3
2.1 แนวคิดเบื้องต้น ..... 3
2.2 เครื่องควบคุมแบบโปรแกรมได้ (Programmable Logic Control) ..... 3
2.2.1 ความหมาย ..... 3
2.2.2 พีแอลซี ..... 4
2.2.3 หลักการทำงานของพีแอลซี ..... 4
2.2.4 โครงสร้างของพีแอลซี ..... 4
2.2.4.1 แรม (Random Access Memory) ..... 5
2.2.4.2 อีพรอม (Erasable Programmable Read Only Memory : EPROM) ..... 5
2.4.4.3 อีอีพรอม(Electronic Erasable Programmable Read Only Memory : EEPROM) ..... 5
2.2.5 ส่วนประกอบของพีแอลซี ..... 6

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# สารบัญ

หน้า

**บทคัดย่อภาษาไทย** ..... I
**บทคัดย่อภาษาอังกฤษ** ..... II
**กิตติกรรมประกาศ** ..... III
**สารบัญ** ..... IV
**สารบัญตาราง** ..... VI
**สารบัญรูป** ..... VII

**บทที่ 1 บทนำ** ..... 1
1.1 ความสำคัญของปริญญานิพนธ์ ..... 1
1.2 วัตถุประสงค์ของปริญญานิพนธ์ ..... 1
1.3 ขอบเขตของปริญญานิพนธ์ ..... 2
1.4 ขั้นตอนการศึกษา ..... 2
1.5 ประโยชน์ที่คาดว่าจะได้รับ ..... 2

**บทที่ 2 ทฤษฎี** ..... 3
2.1 แนวคิดเบื้องต้น ..... 3
2.2 เครื่องควบคุมแบบโปรแกรมได้ (Programmable Logic Control) ..... 3
2.2.1 ความหมาย ..... 3
2.2.2 พีแอลซี ..... 4
2.2.3 หลักการทำงานของพีแอลซี ..... 4
2.2.4 โครงสร้างของพีแอลซี ..... 4
2.2.4.1 แรม (Random Access Memory) ..... 5
2.2.4.2 อีพรอม (Erasable Programmable Read Only Memory : EPROM) ..... 5
2.4.4.3 อีอีพรอม(Electronic Erasable Programmable Read Only Memory : EEPROM) ..... 5
2.2.5 ส่วนประกอบของพีแอลซี ..... 6

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# สารบัญ (ต่อ)

หน้า


2.2.5.1 หน่วยประมวลผลและคำนวณ (CPU : Control Processing Unit) ..... 6
2.2.5.2 หน่วยอินพุต (Input Interface) ..... 6
2.2.5.3 หน่วยเอาต์พุต (Output Interface) ..... 7
2.2.5.4 หน่วยป้อนโปรแกรม (Programming Device) ..... 7
2.2.6 อุปกรณ์พื้นฐานภายในพีแอลซี ..... 7
2.2.6.1 อินพุตรีเลย์ (Input Relay) ..... 7
2.2.6.2 เอาต์พุตรีเลย์ (Output Relay) ..... 8
2.2.6.3 รีเลย์ช่วย (Auxiliary Relay) ..... 8
2.2.6.4 ไทเมอร์ (Timer) ..... 8
2.2.6.5 เคาน์เตอร์ (Counter) ..... 8
2.2.7 ตัวอย่างการใช้ พีแอลซี ในอุตสาหกรรม ..... 8
2.2.8 พีแอลซี (Programmable Logic Controller) รุ่น PLC C200HE ..... 9
2.2.8.1 เครื่อง PLC C200HE ..... 9
2.2.8.2 ซอฟแวร์ ..... 10
2.3 การจัดการฐานข้อมูล ..... 10
2.4 การใช้งานโปรแกรม Delphi5 ..... 13
2.5 การใช้งานโปรแกรม Syswin3.4 ..... 17
2.6 การใช้งานโปรแกรมมิ่งคอลโซล (Programming Console) ..... 26
2.7 การติดต่อสื่อสาร ..... 31
2.7.1 การใช้งานพอร์ตอนุกรม RS232 ..... 32
2.7.2 การเชื่อมต่ออุปกรณ์ภายนอกเข้ากับคอมพิวเตอร์ด้วยสาย DB9 ..... 33

**บทที่ 3 วิธีการดำเนินงาน ..... 35**
3.1 คำนำ ..... 35
3.2 ลำดับขั้นตอนการสร้างแบบจำลอง ..... 36
3.3 การออกแบบจำลอง ..... 37



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# สารบัญ (ต่อ)

หน้า


3.3.1 แบบจำลองระบบที่ใช้ในการ Simulator ..... 37
3.3.2 การเขียนโปรแกรมและอธิบายคำสั่ง ..... 37
3.3.3 ส่วนของการตั้งค่ากำหนด Input และ Output ของ Valve ..... 38
3.3.4 วิธีการจำลองระบบควบคุมของของไหลเป็นกราฟฟิค ..... 38
3.3.5 การเขียนโปรแกรมและอธิบายคำสั่ง ..... 39
3.3.6 ผลจากการรันโปรแกรม ..... 50
3.3.7 อธิบายหลักการของแบบจำลองที่ออกแบบไว้ ..... 52
3.3.8 การกำหนด Input ของ Sensor และ Output ของ Valve ..... 52
3.3.9 การกำหนด Input ..... 53
3.3.10 การเขียน Ladder Diagram ..... 55
3.3.11 การกำหนด Output ..... 57
3.3.12 Pack Protocal Read ..... 58

**บทที่ 4 การทดลอง ..... 59**
4.1 การทำงานในส่วนคอมพิวเตอร์ของโปรแกรม Delphi5 ..... 59
4.2 การตั้งค่า Com Port ..... 60
4.3 การส่งบิตไปให้ PLC อ่าน ..... 60
4.4 การทำงานของแบบจำลองจำลองการวัดระดับ ..... 61

**บทที่ 5 สรุปผลการวิจัยและข้อเสนอแนะ ..... 62**
5.1 สรุปผล ..... 62
5.2 ข้อเสนอแนะ ..... 62

**บรรณานุกรม ..... 63**

**ภาคผนวก ..... 64**



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# สารบัญตาราง

**ตารางที่** **หน้า**

2.1 การอ้างอิงตำแหน่งหน่วยความจำแบบบิตและแบบเวิร์ด ..... 11
2.2 ตัวอย่างหน่วยความจำระดับบิต ..... 11
2.3 การอ้างอิง Indirect Address ..... 12
2.4 การแปลง Hex เป็น Binary ..... 13
2.5 แสดงการจัดขา ของคอนเน็กเตอร์ อนุกรมแบบ DB9 และหน้าที่การใช้งานต่างๆ ..... 33

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# สารบัญรูป

**รูปที่** **หน้า**


2.1 แสดงโครงสร้างของพีแอลซี ..... 5
2.2 แสดงตัวอย่างพีแอลซีในโรงงานอุสาหกรรม ..... 9
2.3 แสดงรูปเครื่อง PLC C200HE ..... 9
2.4 แสดงรูปโปรแกรม Delphi5 ..... 14
2.5 แสดงหน้า Form Designer ..... 14
2.6 แสดงหน้า Component Palette ..... 15
2.7 แสดงหน้า Object Inspector ..... 15
2.8 แสดงหน้า Code Editor ..... 16
2.9 แสดงหน้าส่วนประกอบของโปรแกรม Delphi5 ..... 16
2.10 แสดงการเชื่อมต่อ RS232 ..... 17
2.11 แสดงโปรแกรม Syswin3.4 ..... 18
2.12 แสดงเมนูหลัก File ..... 18
2.13 แสดงเมนูหลัก Block ..... 19
2.14 แสดงเมนูหลัก Online ..... 19
2.15 แสดงเมนูหลัก Edit ..... 19
2.16 แสดงเมนูหลัก Project ..... 20
2.17 แสดงการ Set ชนิดของ PLC C200HE ..... 20
2.18 แสดงการเซ็ทการสื่อสารข้อมูล RS232C ..... 21
2.19 แสดงการ Address Symbol Editor ..... 21
2.20 แสดงการเลือกโหมดการทำงาน PLC ..... 22
2.21 แสดงการตั้งวัน เวลา ของ PLC ให้เป็นไปตามโฮส ..... 22
2.22 แสดงการกำหนดข้อมูลแบบเวิลด์ลงหน่วยความจำ IR ของ PLC ..... 23
2.23 แสดงการ Force Set/Reset ..... 23
2.24 แสดงแลดเดอร์ ..... 24
2.25 แสดงการ Up-Down Load ..... 24
2.26 แสดงการตรวจสอบการทำงานในรูปแบบแลดเดอร์โปรแกรม ..... 25
2.27 แสดงการตรวจสอบการทำงาน ..... 25



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# สารบัญรูป(ต่อ)


| รูปที่ | หน้า |
|---|---|
| 2.28 แสดงการใช้งานพอร์ตอนุกรม RS232 | 32 |
| 2.29 พอร์ตอนุกรม RS-232C | 32 |
| 2.30 แสดง DB9 ตัวผู้ เมื่อมองจากด้านหลัง | 30 |
| 2.31 แสดงการทำงานของขาสัญญาณ DB9 | 30 |
| 3.1 แสดงแบบจำลองระบบควบคุมระดับแบบที่ 1 | 37 |
| 3.2 แสดงแบบจำลองระบบควบคุมระดับแบบที่ 2 | 38 |
| 3.3 แสดงแบบจำลองระบบควบคุมระดับแบบที่ 3 | 38 |
| 3.4 แสดงแบบจำลองแบบการวัดระดับในโปรแกรม Visio 2007 | 38 |
| 3.5 แสดงการเขียนโปรแกรมและอธิบายคำสั่ง | 39 |
| 3.6 แสดงแบบจำลองในสภาวะปกติ | 50 |
| 3.7 แสดงการทำงานของแบบจำลอง | 50 |
| 3.8 แสดงการทำงานที่แสดงคำเตือน Over Flow | 51 |
| 3.7 แสดงการเกิด Over Flow ทุก Tank | 51 |
| 4.1 แสดงการแบบจำลองกับตัว Communication เพื่อเชื่อมกับ PLC | 59 |
| 4.2 แสดงการตั้งค่า Com Port | 60 |
| 4.3 แสดงในส่วนที่ส่งบิตไปให้ PLC อ่าน | 60 |
| 4.4 แสดงส่วนการทำงานของโปรแกรม | 61 |


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# บทที่ 1
# บทนำ

## 1.1 ความสำคัญของปริญญานิพนธ์

ปัจจุบันตัวควบคุมแบบโปรแกรมได้ (Programmable Logic Controller : PLC) ได้เข้ามีบทบาทในกระบวนการอุตสาหกรรมอย่างทั่วถึง เนื่องมาจากตัวควบคุมแบบโปรแกรมได้มีความสะดวกต่อการใช้งาน สามารถเชื่อมต่อกับอุปกรณ์อินพุตและเอาต์พุตได้โดยตรง ทำให้การปรับปรุงแก้ไขกระบวนการผลิต หรือลำดับการทำงานใหม่ได้ง่าย โดยทำการเปลี่ยนเพียงโปรแกรมใหม่เท่านั้น สามารถลดขั้นตอนการทำงานที่ยุ่งยาก ช่วยเพิ่มประสิทธิภาพในการผลิตอำนวยความสะดวกต่อผู้ปฏิบัติงานที่ควบคุมการผลิต เป็นการควบคุมแบบอัตโนมัติ โดยหลักการสำคัญของระบบควบคุมอัตโนมัติคือ การที่สามารถนำค่าสัญญาณ หรือข้อมูลจากกระบวนการผลิตที่ได้จากตัววัดมาทำการประมวลผลเพื่อให้ได้ค่า หรือท้ายผลลัพธ์ควบคุมสำหรับส่งออกไปยังอุปกรณ์ควบคุมตัวสุดท้าย เพื่อให้สามารถควบคุมกระบวนการได้อย่างถูกต้องแม่นยำ ทันการและต่อเนื่อง เทคโนโลยีที่ใช้ในระบบควบคุมแบบอัตโนมัติคงหนีไม่พ้นระบบคอมพิวเตอร์ ที่จัดได้ว่าเป็นองค์ประกอบสำคัญของระบบ และได้รับความไว้วางใจจากมนุษย์ว่าสามารถปฏิบัติงานดังกล่าวได้อย่างไม่มีข้อผิดพลาด สำหรับปริญญานิพนธ์นี้เป็นการนำเอาตัวควบคุมแบบโปรแกรมได้ มาประยุกต์ใช้ควบคุมแบบจำลองระดับน้ำในแทงค์จำลองของระบบหนึ่ง โดยแสดงผลและควบคุมกระบวนการผ่านหน้าจอคอมพิวเตอร์ (โดยใช้โปรแกรมจำลองระบบ Delphi5) ที่ทำการเชื่อมต่อกับพีแอลซี

## 1.2 วัตถุประสงค์ของปริญญานิพนธ์

1. เข้าใจหลักการใช้งานและเขียนโปรแกรม เพื่อสั่งงานตัวควบคุมแบบโปรแกรมได้ โดยโปรแกรม Delphi5 จำลองระบบการวัดระดับเพื่อควบคุมของไหล
2. เข้าใจหลักการ Simulator
3. เข้าใจหลักการกำหนดค่า Input/Output ของระบบที่จำลองไว้
4. เข้าใจโปรโตคอล (Protocal) ให้สามารถติดต่อสื่อสารระหว่างพีแอลซีกับคอมพิวเตอร์
5. เข้าใจหลักการเขียนโปรแกรม PLC เขียน Ladder เพื่อควบคุมแบบจำลอง
6. เข้าใจหลักการ Link PLC กับ Simulator ให้ใช้งานได้จริง
7. ประโยชน์ที่ได้รับในการทำโครงงานนี้คือ เป็นกรณีศึกษาการเรียนวิชา PLC และจำลองระบบควบคุมการวัดระดับของของไหลด้วย PLC เพื่อไปประยุกต์ใช้ในงานอุตสาหกรรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 1.3 ขอบเขตของปริญญานิพนธ์

1. สามารถสร้างแบบจำลองระบบควบคุมของของไหลได้
2. สามารถสร้างแบบจำลองตัว Sensor เมื่อของไหลถึงระดับที่กำหนดได้
3. สามารถเขียนโปรแกรม PLC สั่งงานวาล์วควบคุมของของไหลแบบเปิดปิดได้ (On-Off)

## 1.4 ขั้นตอนการศึกษา

1. ศึกษาข้อกำหนดและคุณลักษณะต่างๆ ของอุปกรณ์เซนเซอร์และวาล์วควบคุม
2. ศึกษาการสร้างแบบจำลองของระบบในโปรแกรม Microsoft Visio Pro 2007
3. ศึกษาการเขียนโปรแกรมสร้างการทำงานอย่างเป็นระบบของโปรแกรม Delphi 5
4. ศึกษาการทำงานของอุปกรณ์อินพุตและเอาต์พุตแบบ Read และ Write
5. ศึกษาการทำงานของเครื่อง PLC รุ่น C200HE ของ Omron
6. ศึกษาการเขียน Ladder สถานการณ์ทำงานแต่ละ bit ของเครื่อง PLC ซึ่งเขียนด้วยโปรแกรม Syswin 3.4 หรือ ใช้ Programming Console รุ่น PRO01 ของ Omron

## 1.5 ประโยชน์ที่คาดว่าจะได้รับ

1. เข้าใจทฤษฎีและสร้างแบบจำลองของระบบควบคุมได้
2. สามารถพัฒนาโปรแกรมควบคุมเพื่อใช้งานตัวควบคุมแบบโปรแกรมได้ และอุปกรณ์อินพุตและเอาท์พุตของ PLC ซึ่งเป็นอุปกรณ์ควบคุมในอุตสาหกรรมยี่ห้อ Omron
3. เข้าใจโครงสร้างและพัฒนาโปรแกรม Delphi เพื่อให้คอมพิวเตอร์ติดต่อกับตัวควบคุมแบบโปรแกรมได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# บทที่ 2
# ทฤษฎี

## 2.1 แนวคิดเบื้องต้น

การควบคุมกระบวนการต่างๆในอุตสาหกรรม เป้าหมายของการควบคุมคือ การรักษาปริมาณทางฟิสิกส์ อันได้แก่ อุณหภูมิ ความดัน อัตราการไหล ระดับ และอื่นๆ ให้มีค่าใกล้เคียงกับค่าเป้าหมายซึ่งเป็นค่าที่ต้องการมากที่สุด แม้ว่าสภาวะการทำงานและสภาวะแวดล้อมอาจเปลี่ยนแปลงตลอกเวลาก็ตาม ในอดีตการควบคุมกระบวนการต่างๆจะเลือกใช้การควบคุมแบบง่ายๆ ด้วยมือซึ่งอาศัยพนักงาน หรือผู้ควบคุมคอยทำหน้าที่เกี่ยวกับการตรวจวัด และเปลี่ยนแปลงการควบคุมให้ผลตอบสนองเป็นไปตามที่ต้องการ ดังนั้นการควบคุมจะดีหรือไม่อย่างไรนั้นจะขึ้นอยู่กับพนักงานควบคุมเป็นหลัก ซึ่งในการปฏิบัติงานจริงแล้วพนักงานควบคุมไม่สามารถทำงานได้ดีเท่ากันตลอดเวลา จึงทำให้ประสิทธิภาพของการควบคุมลดลง ดังนั้นปัจจุบันการควบคุมที่มีความต้องการเรื่องความแม่นยำและมีประสิทธิภาพสูง จำเป็นต้องนำการควบคุมแบบอัตโนมัติมาใช้งานที่มีทั้งระบบคอมพิวเตอร์ ระบบสื่อสาร รวมทั้งซอฟแวร์เฉพาะงานและการเก็บข้อมูลตลอดจนการนำข้อมูลไปวิเคราะห์เพื่อวัตถุประสงค์ต่างๆ

## 2.2 ตัวควบคุมแบบโปรแกรมได้พีแอลซี (Programmable Logic Controller : PLC)

### 2.2.1 ความหมายของพีแอลซี

พีแอลซี หรือ โปรแกรมเมเบิลลอจิกคอลโทรลเลอร์ เป็นอุปกรณ์ควบคุมการทำงานของเครื่องจักร หรือกระบวนการทำงานต่างๆ ภายในพีแอลซีมีไมโครโปรเซสเซอร์เป็นสมองกลสั่งการที่สำคัญ พีแอลซีส่วนที่เป็นอินพุตและเอาต์พุตสามารถต่อออกไปใช้งานได้ทันที ตัวตรวจวัดหรือสวิตช์ต่างๆ จะต่อเข้าทางด้านอินพุต ส่วนทางด้านเอาต์พุตใช้ต่อออกไปควบคุมการทำงานของอุปกรณ์หรือเครื่องจักรที่เป็นเป้าหมาย สามารถสร้างวงจรหรือแบบของการควบคุมได้โดยป้อนโปรแกรมคำสั่งให้พีแอลซี นอกจากนี้พีแอลซียังสามารถใช้งานร่วมกับอุปกรณ์อื่น เช่น เครื่องอ่านบาร์โค้ด เครื่องพิมพ์ ซึ่งในปัจจุบันนอกจากนำพีแอลซีมาใช้งานแบบเดี่ยวแล้วยังสามารถต่อพีแอลซีหลายๆตัวเข้าด้วยกัน เพื่อควบคุมการทำงานของระบบให้มีประสิทธิภาพมากยิ่งขึ้น จะเห็นได้ว่าการใช้งานพีแอลซีมีความยืดหยุ่นมาก ดังนั้นในโรงงานอุตสาหรรมต่างๆจึงเปลี่ยนมาใช้พีแอลซีมากขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**2.2.2 พีแอลซี**

พีแอลซีเป็นอุปกรณ์ที่ใช้แทนการควบคุมแบบรีเลย์สมัยเก่า เป็นระบบอิเล็กทรอนิกส์ใช้การเขียนโปรแกรมแทนการเดินสายไฟฟ้า มีหน่วยอินพุตและเอาต์พุตแบบลอจิก (สถานะติดและดับ) และแบบแอนาลอก จึงทำให้สามารถควบคุมเครื่องจักรได้ทุกชนิดและมีหน่วยอินพุตและเอาต์พุตจำนวนมาก พีแอลซีเป็นอุปกรณ์ที่ประกอบด้วยวงจรอิเล็กทรอนิกส์ ที่เป็นโมดูลสามารถตรวจสอบความเสียหายด้วยตัวเองได้ เพราะฉะนั้นเมื่อพีแอลซีเกิดความเสียหายขึ้นสามารถแก้ไขได้ โดยเปลี่ยนโมดูลเท่านั้นและพีแอลซียังสามารถตรวจสอบสภาวะเปิดปิดของอุปกรณ์ภายนอกตามโปรแกรมได้ ให้สามารถตรวจหาข้อบกพร่องได้อย่างรวดเร็วมีความน่าเชื่อถือและมีประสิทธิภาพสูง

**2.2.3 หลักการทำงานของพีแอลซี**

พีแอลซีจะรับสัญญาณที่เป็นคำสั่งจากสวิตซ์ปุ่มกด สวิตซ์เลือก และสวิตซ์ตัวเลข ซึ่งอยู่ที่แผงควบคุมเครื่องจักร นอกจากนี้ยังรับสัญญาณจากอุปกรณ์ตรวจวัดสภาพการทำงานของเครื่องจักร เช่น ลิมิตสวิตซ์ พล็อกซิมิตีสวิตซ์ สวิตซ์แสง และสวิตซ์ตรวจจับชนิดต่างๆ จากนั้นพีแอลซีจะส่งสัญญาณออกไปที่ขั้วออก เพื่อขับเคลื่อนอุปกรณ์ต่างๆ เช่น มอเตอร์ โซลินอยด์ และคลัตซ์แม่เหล็กไฟฟ้า หรือขับพวกหลอดแสดง หลอดตัวเลข เป็นต้น พีแอลซีจะรับสัญญาณเข้ามาทางขั้วเข้าและให้สัญญาณออกทางขั้วออก การให้สัญญาณออกนี้จะเป็นไปตามโปรแกรมที่เก็บไว้ในพีแอลซี การขับอุปกรณ์ซึ่งเป็นโหลดขนาดเล็ก เช่น โซลินอยด์ตัวเล็กหรือหลอดแสดงนั้น พีแอลซีสามารถขับได้โดยตรงจากขั้วออก แต่ถ้าเป็นอุปกรณ์ที่เป็นโหลดขนาดใหญ่ ใช้กำลังมาก เช่น มอเตอร์สามเฟส หรือโซลินอยด์ตัวใหญ่จำเป็นต้องต่อผ่านคอนแทคเตอร์และรีเลย์ เพื่อช่วยขยายกำลังขับคอนแทคเตอร์และรีเลย์ขับเบรกเกอร์เหล่านี้ จะถูกติดตั้งอยู่ภายในตู้ควบคุมเดียวกันกับพีแอลซี

**2.2.4 โครงสร้างของพีแอลซี**

พีแอลซีเป็นอุปกรณ์คอมพิวเตอร์ใช้สำหรับงานอุตสาหกรรม พีแอลซีประกอบด้วย หน่วยประมวลผลกลาง หน่วยความจำ หน่วยรับข้อมูล และหน่วยป้อนโปรแกรมสำหรับพีแอลซีขนาดเล็กส่วนประกอบทั้งหมดของพีแอลซีจะรวมกันเป็นเครื่องเดียว แต่ถ้าเป็นพีแอลซีขนาดใหญ่สามารถแยกออกเป็นส่วนประกอบย่อยๆได้ หน่วยความจำของพีแอลซีประกอบด้วย หน่วยความจำชนิดรอมและแรม หน่วยความจำแรมทำหน้าที่เก็บโปรแกรมของผู้ใช้งานและข้อมูลสำหรับใช้การปฏิบัติงานของพีแอลซี ส่วนหน่วยความจำชนิดรอม ทำหน้าที่เก็บโปรแกรมสำหรับใช้ในการปฏิบัติงานของพีแอลซีตามโปรแกรมของผู้ใช้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**2.2.4.1 แรม (Random Access Memory : RAM)**

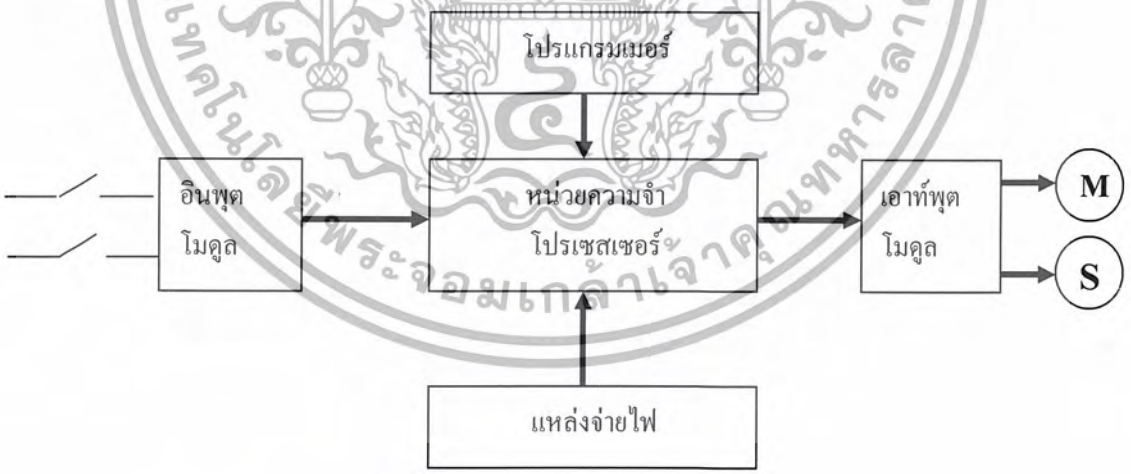
หน่วยความจำประเภทนี้จะมีแบตเตอรี่เล็กๆ ต่อไว้ เพื่อใช้รักษาข้อมูลเมื่อเกิดไฟดับ การอ่านและเขียนโปรแกรมลงในแรมทำได้ง่ายมาก จึงเหมาะกับการใช้งานในระยะทดลองเครื่องที่มีการเปลี่ยนแปลงแก้ไขโปรแกรมบ่อยๆ

**2.2.4.2 อีพรอม (Erasable Programmable Read Only Memory : EPROM)**

หน่วยความจำชนิดอีพีรอมนี้ จะต้องใช้เครื่องมือพิเศษในการเขียนโปรแกรมการลบโปรแกรมทำได้โดยใช้แสงอัลตราไวโอเลต หรือตากแดดร้อนๆนานๆมีข้อดีตรงที่โปรแกรมจะไม่สูญหายแม้ไฟดับ จึงเหมาะกับการใช้งานที่ไม่ต้องการเปลี่ยนโปรแกรม

**2.2.4.3 อีอีพรอม (Electrical Erasable Programmable Read Only Memory : EEPROM)**

หน่วยความจำชนิดนี้ ไม่ต้องใช้เครื่องมือพิเศษในการเขียนและลบโปรแกรมโดยใช้วิธีการทางไฟฟ้าเหมือนกับแรม นอกจากนั้นก็ไม่จำเป็นต้องมีแบตเตอรี่สำรองเมื่อไฟดับ ราคาจะแพงกว่า แต่จะรวมคุณสมบัติที่ดีของทั้งแรมและอีพรอมเอาไว้ด้วยกัน สำหรับลักษณะโครงสร้างของพีแอลซีแสดงดังรูปที่ 2.1




**รูปที่ 2.1** แสดงโครงสร้างของพีแอลซี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**2.2.5 ส่วนประกอบของพีแอลซี**

พีแอลซีแบ่งออกได้ 3 ส่วนด้วยกันคือ

1. หน่วยประมวลผลและคำนวณ(Control Processing Unit : CPU)
2. อินพุตและเอาต์พุต (Input Output : I/O)
3. หน่วยป้อนโปรแกรม (Programming Device)

**2.2.5.1 หน่วยประมวลผล และคำนวณ (CPU : Control Processing Unit)**

ซีพียู เป็นสมองกลของระบบ ภายในซีพียูจะประกอบไปด้วยวงจรลอจิกเกทชนิดต่างๆ หลายชนิดและมีฐานไมโครโปรเซสเซอร์ ใช้สำหรับแทนอุปกรณ์รีเลย์ เคาร์เตอร์ไทเมอร์ และซีแควนเซอร์ เพื่อให้ผู้ใช้ได้ออกแบบวงจรณ์รีเลย์แลดเดอร์ลอจิกเข้าไปได้

ซีพียูจะอ่านข้อมูลอินพุตจากอุปกรณ์ให้สัญญาณต่างๆ จากนั้นจะทำงาน และเก็บข้อมูล โดยใช้โปรแกรมจากหน่วยความจำและส่งข้อมูลที่ถูกต้องไปยังอุปกรณ์ควบคุม และจ่ายไฟฟ้ากระแสตรง ใช้สำหรับสร้างแรงดันไฟฟ้าให้กับโปรเซสเซอร์และหน่วยอินพุต-เอาต์พุต แหล่งจ่ายนี้จะเก็บไว้ที่ซีพียู หรือแยกออกไปติดตั้งที่จุดอื่นก็ได้ขึ้นอยู่กับผู้ผลิตแต่ละราย

การประมวลผลโปรแกรมของซีพียู ทำได้โดยรับข้อมูลจากหน่วยอินพุตและเอาต์พุต และส่งข้อมูลสุดท้ายที่ได้จากการประมวลผลไปยังหน่วยเอาต์พุต เรียกว่า การสแกน ซึ่งใช้เวลาจำนวนหนึ่ง เรียกว่า เวลาสแกน เวลาในการสแกนแต่ละรอบประมาณ 1 ถึง 100 มิลลิวินาที (0.001-0.1วินาที) ขึ้นอยู่กับข้อมูลและความยาวของโปรแกรม หรือจำนวนอินพุตและเอาต์พุต หรือจำนวนอุปกรณ์ที่ต่อจากพีแอลซี เช่น เครื่องพิมพ์ จอภาพ เป็นต้น อุปกรณ์เหล่านี้จะทำให้เวลาในการสแกนยาวนานขึ้น การเริ่มต้นการสแกนเริ่มต้นจากรับคำสั่งสภาวะของอุปกรณ์จากหน่วยอินพุตมาเก็บไว้ในหน่วยความจำ แล้วทำงานตามโปรแกรมที่เขียนไว้ทีละคำสั่งจากหน่วยความจำนั้นจนสิ้นสุดจึงส่งไปที่หน่วยเอาต์พุต ซึ่งการสแกนของพีแอลซีประกอบด้วย

1. สแกนอินพุต-เอาต์พุต คือ การบันทึกสภาวะข้อมูลของอุปกรณ์ที่เป็นอินพุตและให้อุปกรณ์เอาต์พุตทำงาน
2. สแกนโปรแกรม คือ การให้โปรแกรมทำงานตามลำดับก่อนหลัง

**2.2.5.2 หน่วยอินพุติ (Input Interface)**

หน่วยอินพุตเป็นที่รวบรวมสัญญาณจากอินพุต มักจะใช้แรงดันฟ้าดีซีทางด้านอินพุต 24 โวลต์ เพื่อป้องกันความเสียหายจากกระแสและแรงดันสูง ขั้วต่อทางด้านอินพุตใช้วิธีแยกกันด้วยแสง เช่น ใช้ตัวเชื่อมทางแสงเท่านั้นที่เป็นตัวส่งผ่านสัญญาณจากขั้วเข้าจะไม่ส่งผ่านไปยังซีพียูโดยตรง มีเพียงสัญญาณแสงเท่านั้นที่เป็นตัวส่งผ่านสัญญาณจากขั้วเข้าไปให้ซีพียูหน่วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อินพุตรับสัญญาณ หรือสภาวะแล้วส่งไปยังซีพียูเพื่อประมวลผล เมื่อซีพียูประมวลผลแล้วส่งให้ส่วนเอาต์พุตเพื่อให้อุปกรณ์ทำงานตามที่โปรแกรมเอาไว้

สัญญาณอินพุตจากภายนอกที่เป็นสวิตซ์และตัวตรวจจับชนิดต่างๆ จะถูกแปลงให้เป็นสัญญาณที่เหมาะสม ไม่ว่าจะเป็นกระแสสลับ หรือกระแสตรงเพื่อส่งให้ซีพียู ดังนั้นสัญญาณเหล่านี้จึงต้องมีความถูกต้องไม่เช่นนั้นแล้วซีพียูจะเสียหายได้

**2.2..5.3 หน่วยเอาต์พุต (Output Interface)**

หน่วยเอาต์พุต ทำหน้าที่รับค่าสภาวะจากการประมวลผลของซีพียูแล้วนำค่าที่ได้ไปควบคุมอุปกรณ์ เช่น รีเลย์ โซลินอยด์ หรือ หลอดไฟ เป็นต้น นอกจากนั้นยังทำหน้าที่แยกสัญญาณของหน่วยประมวลผลกลางออกจากอุปกรณ์เอาต์พุต

เอาต์พุตมีหลายชนิดให้เลือกขึ้นอยู่กับอุปกรณ์เอาต์พุตที่ใช้ เช่น เอาต์พุตรีเลย์ใช้ได้ทั้งกับไฟฟ้ากระแสสลับและกระแสตรง ถ้าเป็นโหลดที่มีการเปิด-ปิด บ่อย ๆ นิยมใช้เอาต์พุตแบบทรานซิสเตอร์แทนรีเลย์ อุปกรณ์ที่ใช้เป็นเอาต์พุต ได้แก่ รีเลย์ มอเตอร์ไฟฟ้า โซลินอยด์ ขดลวดความร้อน หลอดไฟ เป็นต้น

**2.2.5.4 หน่วยป้อนโปรแกรม(Programming Device)**

เครื่องป้อนโปรแกรม(Hand held) ทำหน้าที่ป้อนโปรแกรมควบคุมของผู้ใช้ลงในหน่วยความจะของพีแอลซี นอกจากนี้ยังทำหน้าที่ติดต่อระหว่างผู้ใช้กับพีแอลซีเพื่อให้ผู้ใช้สามารถตรวจการปฏิบัติงานของพีแอลซี ผลการควบคุมเครื่องจักรและกระบวนการตามโปรแกรมควบคุมที่ผู้ใช้เขียนขึ้นได้อีกด้วย

**2.2.6 อุปกรณ์พื้นฐานในพีแอลซี**

ภายในพีแอลซีจะมีอุปกรณ์ที่ใช้ในการควบคุมซเควันซ์ เช่น รีเลย์ ไทเมอร์ เคาน์เตอร์ และอุปกรณ์อื่นๆ อุปกรณ์แต่ละตัวจะมีหน้าสัมผัสทั้งเอและบี การโปรแกรมให้พีแอลซีทำงานคือ การสร้างวงจรรีเลย์ โดยการนำองค์ประกอบเหล่านี้มาต่อเป็นวงจรนั่นเองประกอบด้วย

**2.2.6.1 อินพุตรีเลย์ (Input Relay)**

อินพุตรีเลย์เป็นรีเลย์สำหรับใช้รับสัญญาณจากสวิตซ์สั่งงานและสวิตซ์ตรวจจับที่ต่อเข้ามายังขั้วของพีแอลซี เบอร์ของอินพุตรีเลย์จะนำด้วย X เสมอ จำนวนของอินพุตรีเลย์จะมีเท่าใดขึ้นอยู่กับขนาดและชนิดของพีแอลซี เช่น X00-X0F เป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**2.2.6.2 เอาต์พุตรีเลย์ (Output Relay)**

เอาต์พุตรีเลย์เป็นรีเลย์ขาออก ที่ใช้ในการขับโหลดที่ต่อกับขั้วออกของพีแอลซี เบอร์ของเอาต์พุตรีเลย์จะมีเบอร์นำหน้าด้วย Y เสมอ จำนวนเอาต์พุตรีเลย์จะมีเท่าใดขึ้นอยู่กับขนาดและชนิดของพีแอลซี เช่น Y00-Y0F เป็นต้น

**2.2.6.3 รีเลย์ช่วย (Auxiliary Relay)**

รีเลย์ช่วยเป็นรีเลย์ที่ช่วยในการสร้างวงจรควบคุมซีเคว้นซ์ ภายในพีแอลซีจะจัดเตรียมรีเลย์ช่วยไว้มากมาย เบอร์ขอรีเลย์ช่วยจะนำหน้าด้วย M เสมอ เช่น M0 M1 หรือ M8000 เป็นต้น

**2.2.6.4 ไทเมอร์ (Timer)**

ไทเมอร์เป็นรีเลย์ที่ใช้ในการหน่วงเวลาในการปิด-เปิดของหน้าสัมผัส ภายในพีแอลซีจะจัดเตรียมไทเมอร์ไว้ใช้หลายตัว เบอร์ขอไทเมอร์จะถูกนำหน้าด้วย T เสมอ เช่น T0 T1 เป็นต้น

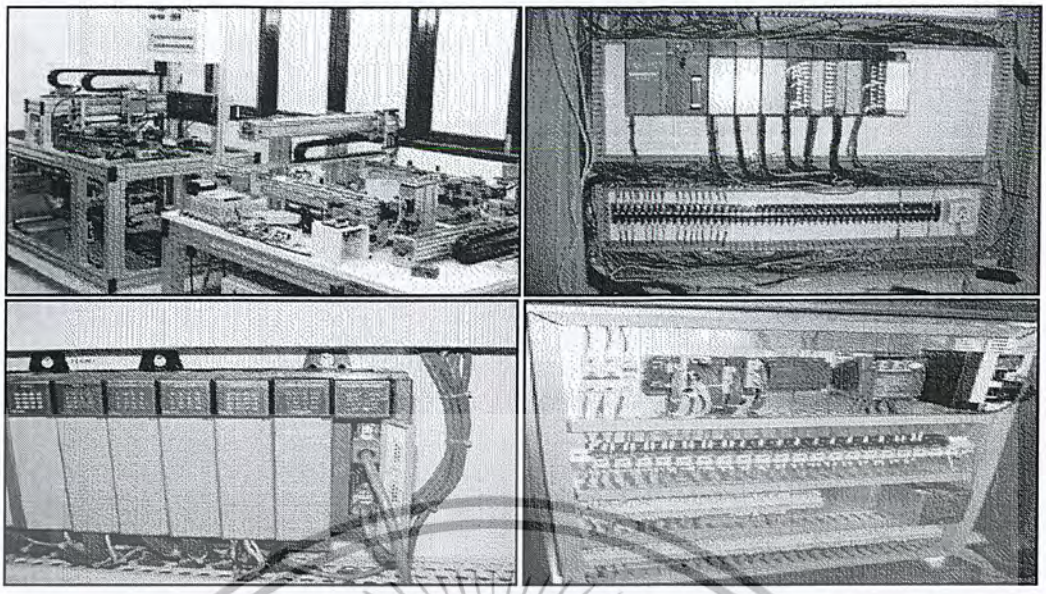
**2.2.6.5 เคาน์เตอร์ (Counter)**

เคาน์เตอร์หรือตัวนับเป็นรีเลย์ที่ใช้นับจำนวนการเปิด-ปิดของหน้าสัมผัสและให้เอาต์พุตเมื่อนับครบตามจำนวนที่ตั้งไว้ เบอร์ของเคาร์เตอร์จะนำหน้าด้วย C เสมอ เช่น C0 C1 เป็นต้น

**2.2.6.7 ตัวอย่างการใช้พีแอลซีในอุตสาหกรรม**

พีแอลซีสามารถนำมาใช้ในระบบงานทางด้านอุตสาหกรรมได้หลายอุตสาหกรรม สำรับตัวอย่างการนะพีแอลซีมาใช้ควบคุมการทำงานนั้น สามมารถแสดงได้ดังรูป 2.2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

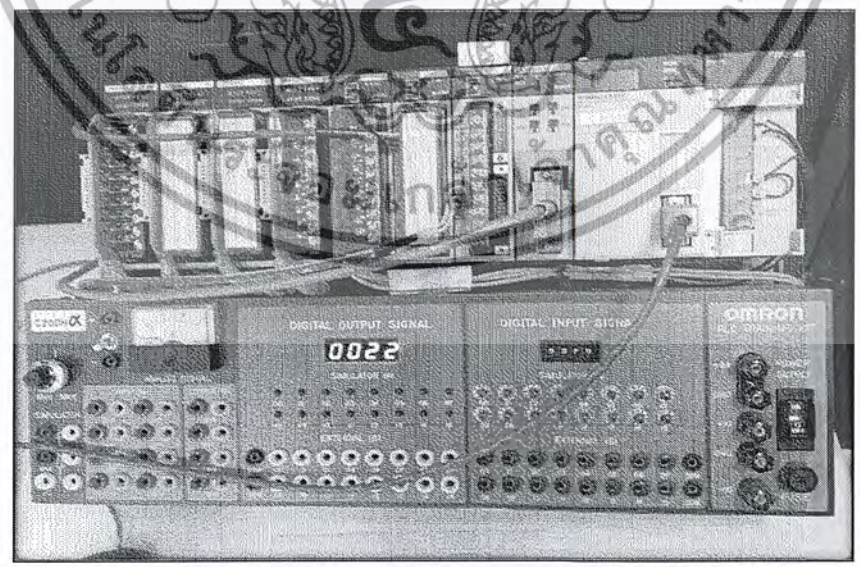


**รูปที่ 2.2** แสดงตัวอย่างพีแอลซีในโรงงานอุสาหกรรม

### 2.2.8 พีแอลซี (Programmable Logic Controller) รุ่น PLC C200HE

การใช้เครื่องควบคุมแบบตรรกะ (Programmable Logic Controller) รุ่น PLC C200HE มีส่วนประกอบของฮาร์ดแวร์ก็คือ เครื่อง PLC C200HE และ ส่วนของซอฟแวร์ ดังนี้

#### 2.2.8.1 เครื่อง PLC C200HE




**รูปที่ 2.3** เครื่อง PLC C200HE

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**2.2.8.2 ซอฟแวร์**

ได้แก่ โปรแกรมการทำงาน หรือคำสั่งบนเครื่องควบคุมโปรแกรมการทำงานต่างๆ เหล่านี้จะสามารถเขียนได้จำเป็นต้องอาศัยเครื่องมือช่วยในการเขียนหรือพัฒนาได้แก่ โปรแกรมมิ่งคอนโซล (PRO27) และโปรแกรมสำเร็จรูป Syswin3.4 ของบริษัทผู้ผลิตสินค้า ที่ใช้งานบนคอมพิวเตอร์

## 2.3 การจัดการฐานข้อมูล

การอ้างอิงหน่วยความจำ PLC C200HE สามารถแบ่งหน่วยความจำได้ดังต่อไปนี้

- หน่วยความจำ IR(Internal relay) รวมถึงหน่วยความจำอินพุทและเอาท์พุทด้วย
- หน่วยหน่วยความจำ HR (Holding relay) หน่วยความจำนี้เป็นหน่วยความจำที่สามารถจดจำค่าสภาวะได้ถึงแม้ว่าจะหยุดจ่ายไฟเลี้ยงเครื่องควบคุมก็ตาม ค่าสภาวะที่จัดเก็บก็ยังคงอยู่เพราะหน่วยความจำส่วนนี้มีแหล่งจ่ายไฟสำรอง คอยสำรองจ่ายให้ตลอดเวลา
- หน่วยความจำ SR (Special relay) เป็นหน่วยความจำที่ทำหน้าที่เฉพาะอย่าง รวมถึงใช้เป็นแฟลกแสดงสถานะ หลังจากที่ปฏิบัติคำสั่งบางกลุ่มคำสั่งที่มีผลต่อสถานะของแฟล็ก
- หน่วยความจำ LR (Link relay) เป็นหน่วยความจำที่ใช้ในการสื่อสารข้อมูลถึงกันกรณีที่เชื่อมต่อ เครื่องควบคุมมากกว่า 1 เครื่องเป็นระบบเครือข่ายในรูปแบบ PC Link บางครั้งเรียกว่า Sysmate link แต่ถ้าทางด้านฮาร์ดแวร์ไม่ได้ใช้งานระบบ PC Link แล้วหน่วยความจำส่วนนี้จะทำหน้าที่เหมือนหน่วยความจำภายในคือมีไว้เพื่อเก็บค่าสภาวะจากการทำงานของโปรแกรม
- หน่วยความจำ AR (Auxiliary relay) เป็นหน่วยความจำแบบพิเศษที่ใช้สำหรับเก็บประวัติความผิดพลาดจากการหยุดเครื่อง อาจจะเป็น วัน เดือน ปี จำนวนครั้ง นอกจากนี้เป็นที่เก็บปฏิทิน นาฬิกา และอื่นๆ อีกจำนวนมาก
- หน่วยความจำ DM (Data Memory) เป็นหน่วยความจำแบบเวิลด์เท่านั้นใช้กับคำสั่งการประมวลผลแบบเวิลด์ไม่สามารถกระทำกับคำสั่งระดับบิตได้ ในการอ้างอิงหน่วยความจำได้ทั้งแบบอ้างอิงโดยตรง (Direct Addressing) และ อ้างอิงโดยอ้อม (Indirect Addressing)
- ในการอ้างอิงตำแหน่งหน่วยความจำของเครื่องควบคุม PLC สามรถอ้างอิงได้ทั้งระดับบิตและระดับเวิร์ด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**ตารางที่ 2.1** การอ้างอิงตำแหน่งหน่วยความจำแบบบิตและแบบเวิร์ด

| ตำแหน่งหน่วยความจำ | แบบบิต | | แบบเวิร์ด | |
|---|---|---|---|---|
| | การอ้างอิง | ตัวอย่าง | การอ้างอิง | ตัวอย่าง |
| หน่วยความจำ IR | ได้ | 00000 | ได้ | 000 |
| หน่วยความจำ HR | ได้ | HR0000 | ได้ | HR00 |
| หน่วยความจำ SR | ได้ | SR25500 | ได้ | SR225 |
| หน่วยความจำ LR | ได้ | LR0000 | ได้ | LR00 |
| หน่วยความจำ AR | ได้ | AR2515 | ได้ | AR18 |
| Timer/Counter | ได้ | Tim000 | ไม่ได้ | --------- |
| หน่วยความจำ DM | ไม่ได้ | ---------- | ได้ | DM0000 |

การอ้างอิงตำแหน่งหน่วยความจำระดับเวิร์ดแบบตรง หมายถึง หน่วยความจำที่ถูกอ้างอิงถึงอยู่นั้นมีข้อมูลเป็นอย่างไร ผู้ใช้งานสามารถเขียนโปรแกรมนำข้อมูลนั้นไปประมวลผลต่อได้โดยตรงลักษณะของข้อมูลจะเป็นเลขฐาน 16 จำนวน 4 หลัก จะเห็นได้ว่าการอ้างอิงหน่วยความจำแบบตรงนั้นไม่ค่อยซับซ้อนมากนัก สำหรับหน่วยความจำ IR เท่านั้นมีความพิเศษกว่าหน่วยความจำประเภทอื่นเพราะไม่ต้องกำหนดอักษร IR นำหน้าแต่จะอาศัยตัวเลขฐาน 10 จำนวน 3 หลักเท่านั้น จะเป็นที่ทราบแล้วว่าอ้างอิงถึงหน่วยความจำในพื้นที่นี้ ส่วนพื้นที่อื่นๆจำเป็นต้องอ้างถึงอักษรนำหน้าเสมอ

**ตารางที่ 2.2** ตัวอย่างหน่วยความจำระดับบิต

| หน่วยความจำเวิลด์ | ข้อมูลเลขฐาน 16 | | | |
|---|---|---|---|---|
| | D3 | D2 | D1 | D0 |
| 030 | 1 | 2 | 3 | 4 |
| 031 | 5 | 6 | 7 | 8 |
| HR00 | 5 | 5 | 5 | 5 |
| DM0000 | A | B | C | D |

การอ้างอิงหน่วยความจำระดับเวิร์ดแบบอ้อม (IndirectAddress) หมายถึง ข้อมูลในหน่วยความจำที่อ้างอิงถึงอยู่นั้นจะไม่ใช่ข้อมูลเหมือนการอ้างอิงแบบตรง แต่จะเป็นเพียงตัวบงชี้ให้ชี้ไปยังตำแหน่งหน่วยความจำอีกทีหนึ่ง ขอเรียกว่าหน่วยความจำตัวชี้ไปยังตำแหน่งหน่วยความจำอีกทีหนึ่งขอเรียกว่าหน่วยความจำตัวชี้ และข้อมูลของหน่วย ความจำตัวชี้นี้จะมีข้อมูลเป็นข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จริง จากที่กล่าวมานี้จะเห็นได้ว่าการอ้างอิงหน่วยความจำระดับเวิร์ดแบบอ้อม (Indirect Address) จะมีความซับซ้อนกว่าแบบตรง ข้อดีของการอ้างอิงแบบอ้อมคือเวลาที่ผู้ใช้งานต้องการกระทำกับหน่วยความจำเพียงแต่เปลี่ยนตำแหน่งตัวชี้ก็จะทำให้ได้ข้อมูลที่เก็บในหน่วยความจำที่แตกต่างกันออกไป ลักษณะของข้อมูลเป็นเลขฐาน 16 จำนวน 4 หลักเช่นกัน ข้อจำกัดอีกอย่างหนึ่งของ การอ้างอิงหน่วยความจำระดับ เวิลด์แบบอ้อมสามารถทำได้เฉพาะหน่วยความจำ DM เท่านั้น และหน่วยความจำตัวชี้ จะต้องเป็นเลขฐาน 10 การอ้างอิงถึงจะต้องอ้างอิงให้อยู่ในพื้นที่ข้อหน่วยความจำ DM ที่สามารถอ่านเขียนได้และไม่เกินนอกพื้นที่ที่มีหน่วยความจำ มิฉะนั้นแล้วจะทำให้เครื่องควบคุมไม่สามรถปฏิบัติโปรแกรมได้จะส่งผลให้เกิดแฟล็กผิดพลาด (Error Flag) เพื่อแสดงสถานะให้ผู้เขียนโปรแกรมทราบตัวอย่าง เช่น Indirect address DM0010 จะเขียนว่า *DM0010 มีข้อมูลเป็น ABCD

**ตารางที่ 2.3** การอ้างอิง Indirect address

| Data Memory | HEX DATA | | | |
|---|---|---|---|---|
| | D3 | D2 | D1 | D0 |
| DM0010 | 0 | 0 | 2 | 0 |
| DM0011 | 0 | 0 | 0 | 0 |
| | | | | |
| | | | | |
| | | | | |
| DM0020 | A | B | C | D |
| DM0021 | 0 | 0 | 0 | 0 |

Address Pointer

Hex data

การอ้างอิงหน่วยความจำระดับบิตในพื้นที่ต่างๆ เป็นการอ้างอิงหน่วยความจำแบบบิตจะเข้าถึงข้อมูล “ON OFF” หรือ “0-1” ที่เป็นเลขฐานสองสำหรับหน่วยความจำ IR เท่านั้น ไม่ต้องกำหนดอักษร IR นำหน้าแต่จะอาศัยตัวเลขฐาน 10 จำนวน 5 หลัก ได้โดยตรงจะเป็นที่ทราบว่าอ้างอิงถึงหน่วยความจำในพื้นที่นี้ ส่วนพื้นที่อื่นๆ จำเป็นต้องอ้างอิงถึงอักษรนำหน้าเสมอ ตัวอย่าง เช่น ต้องการอ้างหน่วยความจำ IR00000 หมายความว่า หน่วยความจำนี้อยู่ที่ เวิลด์ 000 และ บิตที่ 00 หรือบิตที่ 00000 มีข้อมูลเป็น “1” และบิตที่ 00001 มีข้อมูลเป็น “0” จนกระทั้งถึงบิตที่ 00015 มีข้อมูลเป็น “1”

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**ตารางที่ 2.4** การแปลง Hex เป็น Binary

| WORD 000 | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 15 | 14 | 13 | 12 | 11 | 10 | 09 | 08 | 07 | 06 | 05 | 04 | 03 | 02 | 01 | 00 |
| 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 |

BINARY DATA

| 8 | 4 | 2 | 1 |
|---|---|---|---|

HEX DATA

แต่ถ้ามองข้อมูลทั้งเวิลด์แล้วใน เวิลด์ 000 จะมีข้อมูลเป็น 8421 สำหรับหน่วยความจำระดับบิตในพื้นที่ HR, LR, SR, AR ก็สามรถทำได้ในลักษณะเดียวกัน

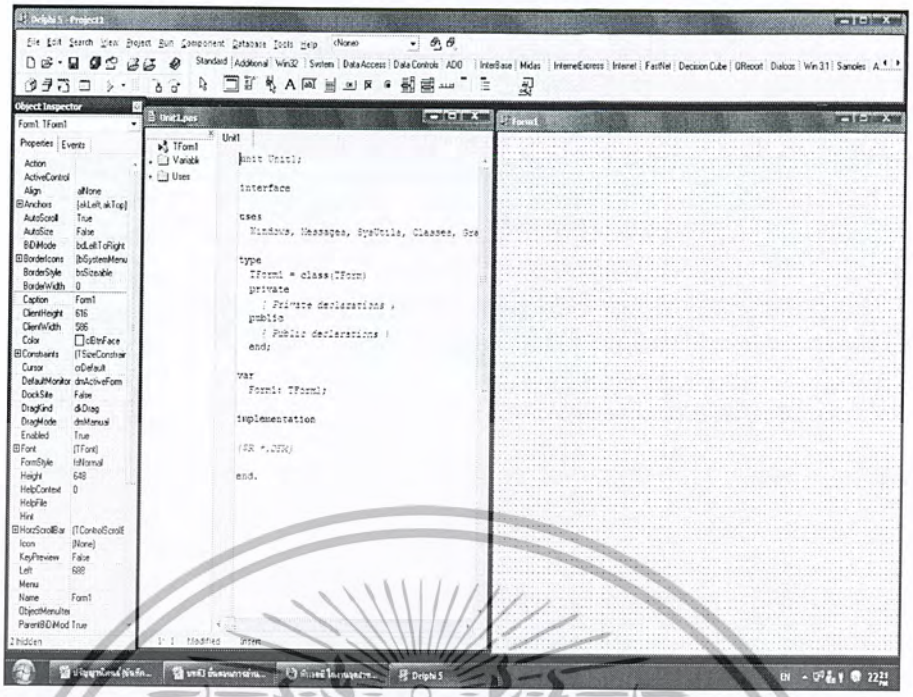
## 2.4 การใช้งานโปรแกรม Delphi5

โปรแกรม Delphi 5 คือซอฟแวร์ที่เรานำมาใช้ในการเขียนโปรแกรมเพื่อสร้างแอพพลิเคชันหรือซอฟแวร์อีกที โดยโปรแกรมนี้จะประกอบไปด้วยเครื่องมือชนิดต่างๆที่ใช้ในการเขียนโปรแกรมทำได้อย่างสะดวก

### 2.4.1 การเข้าโปรแกรม Delphi5

เลือกเมนู Delphi5/Delphi5 เมนูบาร์ เป็นส่วนที่เก็บคำสั่งเพื่อสั่งงาน Dephi โดยมีลักษณะเป็นเมนูให้เลือกใช้งานเหมือนกับโปรแกรมอื่นทั่วไป

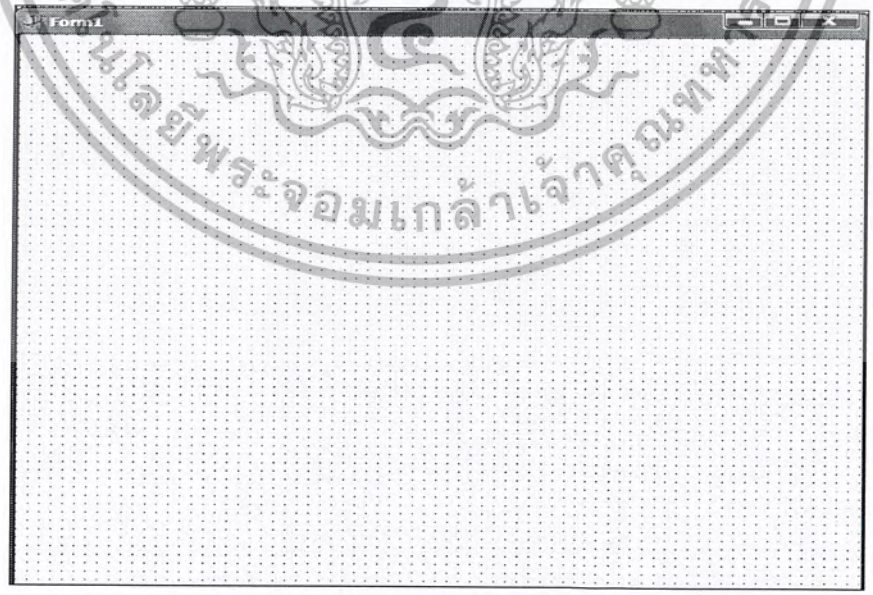
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



**รูปที่ 2.4** โปรแกรม Delphi5

### 2.4.2 Form Designer

เป็นส่วนที่ใช้ในการออกแบบโปรแกรมที่สร้างขึ้นมาด้วย Delphi ที่เราเรียกฟอร์ม (Form) ซึ่งมีลักษณะเหมือนกับวินโดว์ทั่วไป

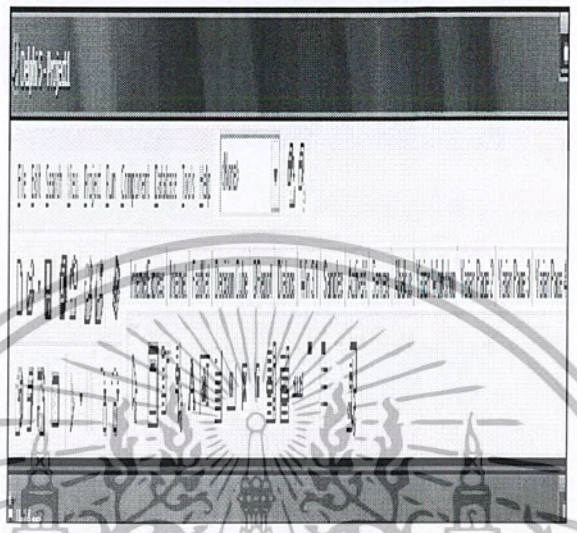



**รูปที่ 2.5** Form Designer

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 2.4.3 Component Palette

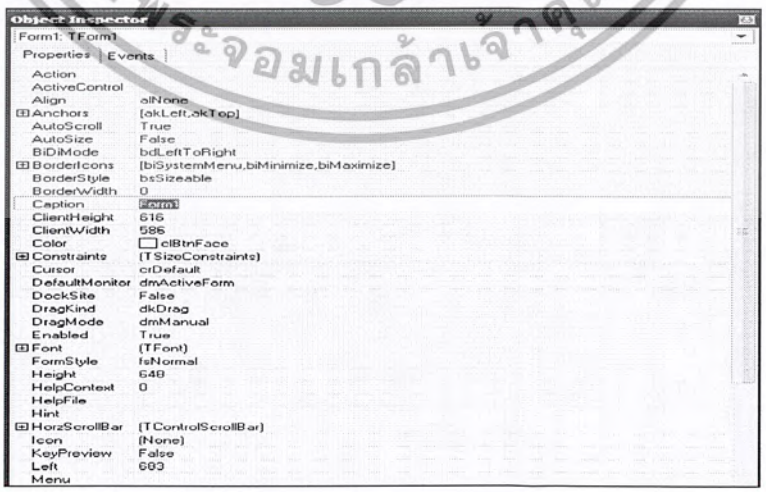
เป็นส่วนที่เก็บคอมโพเนนต์ ซึ่งจะเป็นส่วนสำคัญที่จะเป็นองค์ประกอบภายในฟอร์ม เช่น ปุ่มคลิก ปุ่มเลือกรายการตัวเลือก เป็นต้น



**รูปที่ 2.6** Component Palette

### 2.4.4 Object Inspector

เป็นส่วนที่ใช้กำหนดค่าให้กับฟอร์ม หรือคอมโพเนนต์ที่อยู่บนฟอร์มค่าที่กำหนดให้นี้จะทำให้หน้าตา หรือความสามารถของฟอร์มคอมโพเนนต์เปลี่ยน ไป เช่นเปลี่ยนสีพื้น, เปลี่ยนข้อความบนปุ่มปรับเปลี่ยนขนาดกว้างยาวของฟอร์ม เป็นต้น

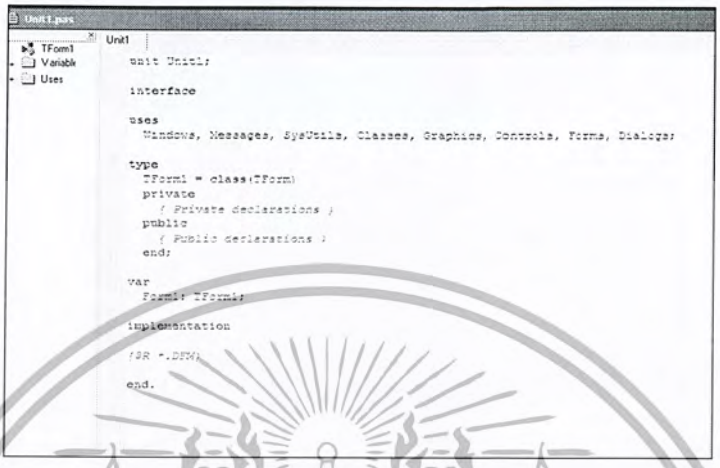



**รูปที่ 2.7** Object Inspector

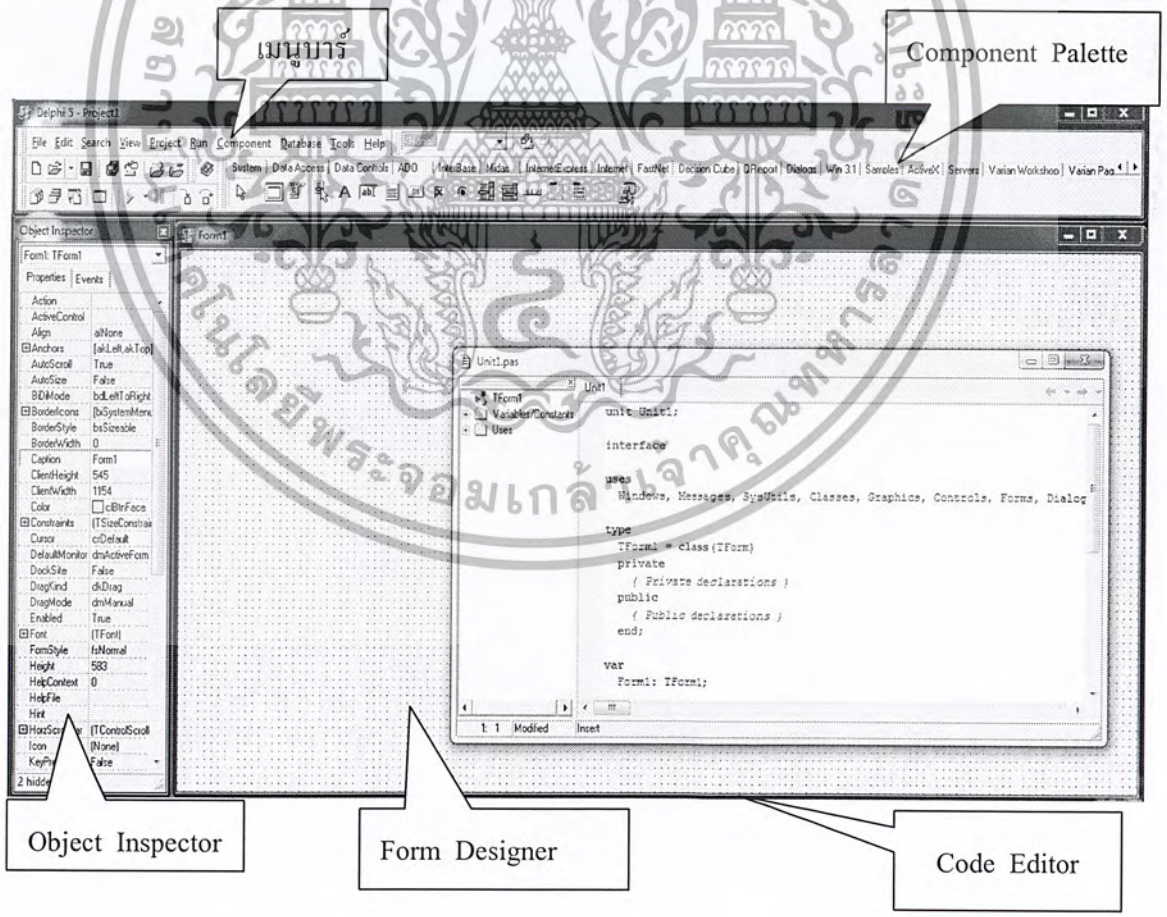
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 2.4.5 Code Editor

เป็นส่วนที่เราใช้เขียนคำสั่งเพื่อให้โปรแกรม หรือแอพลิเคชั่นที่สร้างขึ้นมานั้นทำงานได้ตามที่เราตั้งใจไว้ ซึ่งเราจะเขียนคำสั่งต่างๆด้วยภาษา Object Pascal




**รูปที่ 2.8** Code Editor




**รูปที่ 2.9** ส่วนประกอบโปรแกรม Delphi 5

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

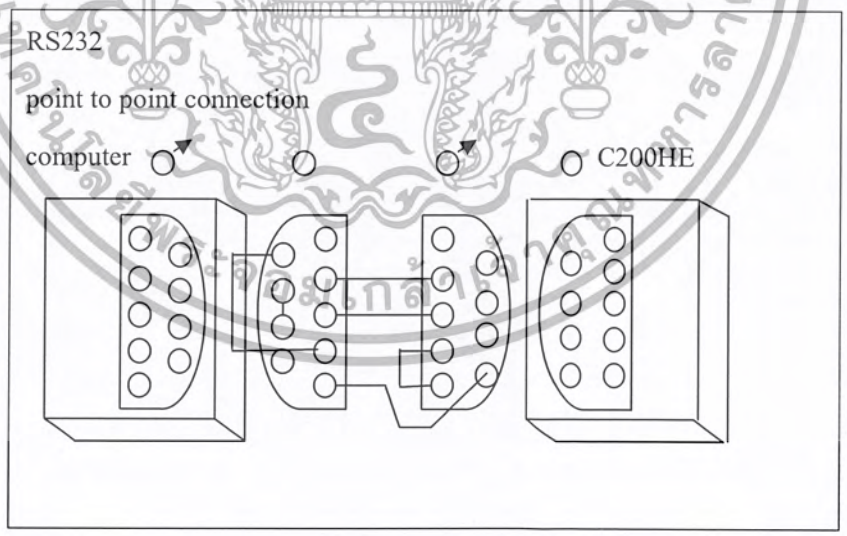
สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง


## 2.5 การใช้งานโปรแกรมสนับสนุน Syswin3.4

โปรแกรมสนับสนุน Syswin3.4 เป็นโปรแกรมแบบแพ็คเกจที่ใช้ในการเขียนบนคอมพิวเตอร์เพื่อพัฒนาโปรแกรมแลดเดอร์ ให้กับเครื่องควบคุม PLC ในโหมดโฮสลิงค์ดังนั้นเครื่องมือชนิดนี้ถือว่ามีความจำเป็นอย่างมากต่อการใช้งานช่วยลดระยะเวลาในการทำงานได้มาก เพราะสามารถตรวจสอบและ แก้ไข ในรูปของโปรแกรมรูปภาพขณะที่ PLC กำลังทำงานอยู่ ในการใช้งานแบ่งการทำงานได้ดังนี้

- การเชื่อมต่อสาย RS232
- การเข้าโปรแกรม syswin3.4
- การกำหนดรุ่น PLC และการสื่อสารข้อมูล
- การกำหนดคำสั่ง PLC
- การกำหนดข้อมูลแบบเวิลด์ลงในหน่วยความจำ PLC
- การ UP-DOWN LOAD ระหว่าง PLC กับโฮสคอมพิวเตอร์
- การตรวจสอบการทำงานในรูปแบบแลดเดอร์โปรแกรม

### 2.5.1 การเชื่อมต่อสายสารสื่อข้อมูล RS232 Point to point Connection in Host-link mode

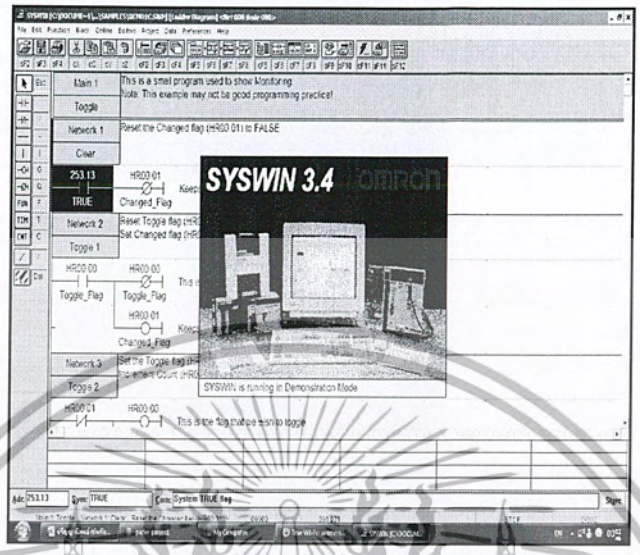



**รูปที่ 2.10** การเชื่อมต่อ RS232

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
119465

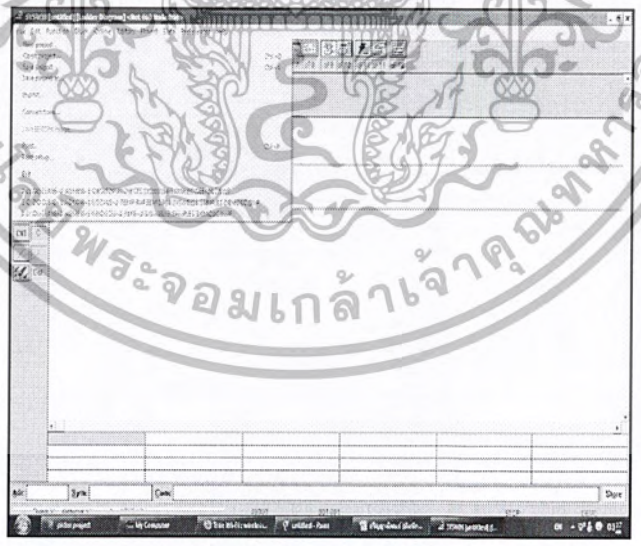
### 2.5.2 การเข้าโปรแกรม Syswin3.4

เลือกเมนู Syswin3.4/Syswin3.4




**รูปที่ 2.11** โปรแกรมsyswin3.4

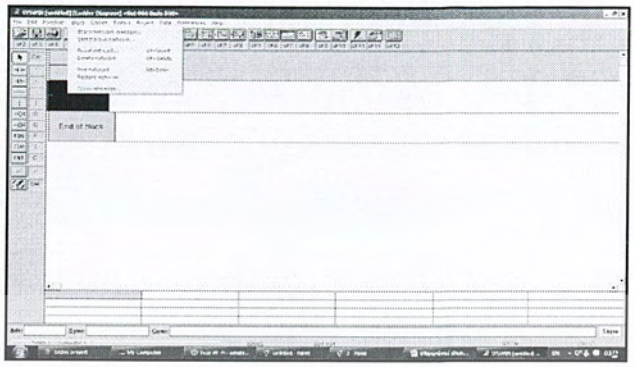
### 2.5.3 แสดงเมนูหลัก File



**รูปที่ 2.12** แสดงเมนูหลัก File

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**2.5.4 แสดงเมนูหลัก Block**

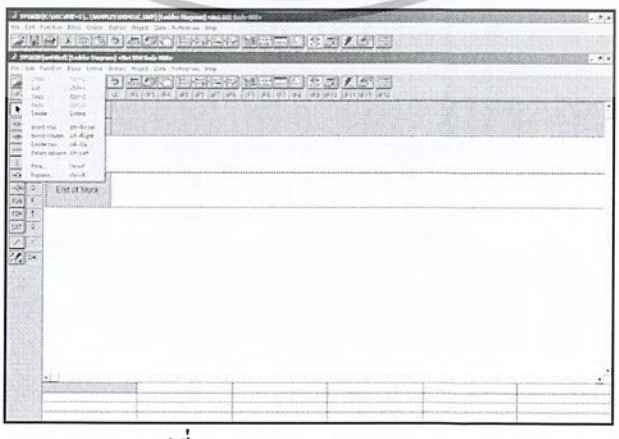


**รูปที่ 2.13** แสดงเมนูหลัก Block

**2.5.5 แสดงเมนูหลัก Online**

**รูปที่ 2.14** แสดงเมนูหลัก Online

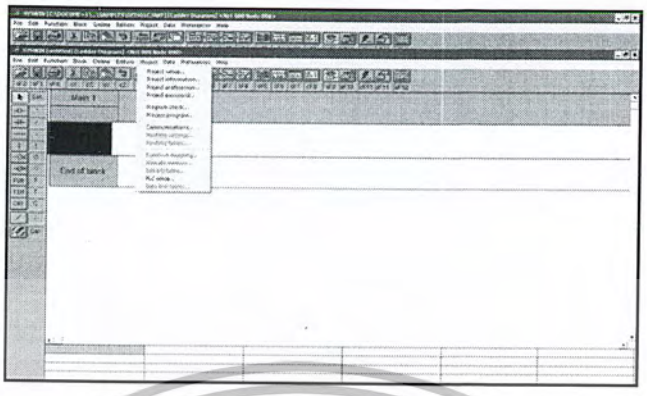
**2.5.6 แสดงเมนูหลัก Edit**



**รูปที่ 2.15** แสดงเมนูหลัก Edit

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

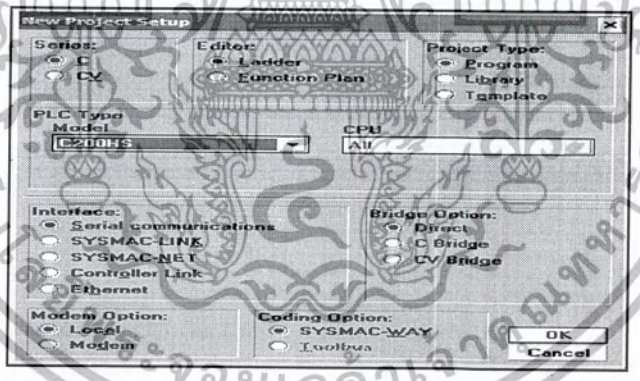
### 2.5.7 แสดงเมนูหลัก Project



รูปที่ 2.16 แสดงเมนูหลัก Project

### 2.5.8 การเซ็ทชนิดของ PLC C200HE

- เลือกเมนู Project/ Project setup
- Mode C200HE



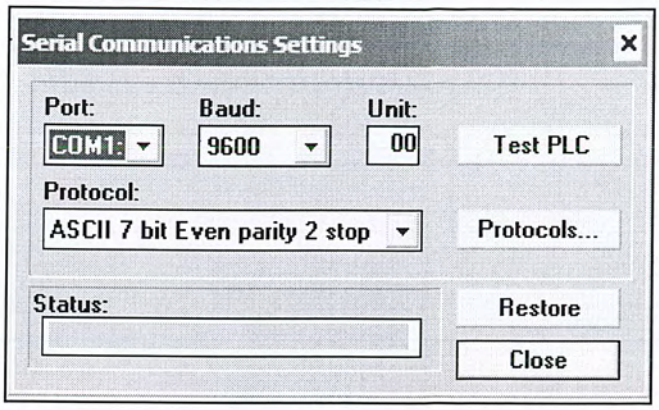

รูปที่ 2.17 การ Set ชนิดของ PLC C200HE

### 2.5.9 การเซ็ทการสื่อสารข้อมูล RS232C

-เลือกเมนู Project/Communication

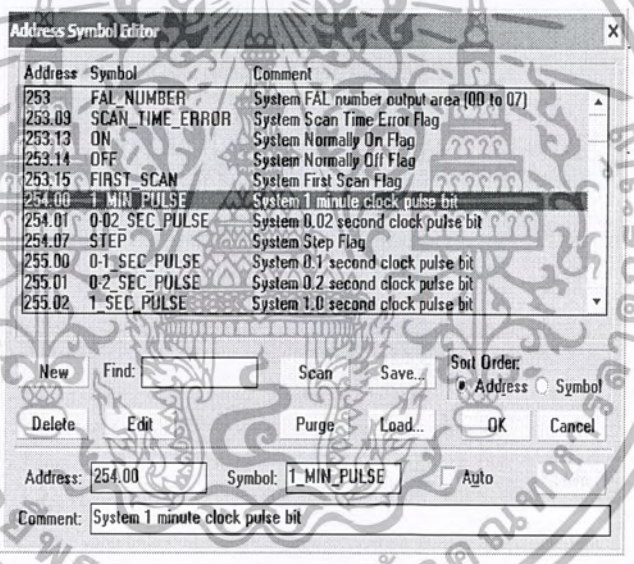
- Coml,9600,00
- Default ASCII 7,E,2
- คลิก test PLC

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้




รูปที่ 2.18 การเซ็ทการสื่อสารข้อมูล RS232C

### 2.5.10 การกำหนดโหมดทำงาน Address Symbol Editor

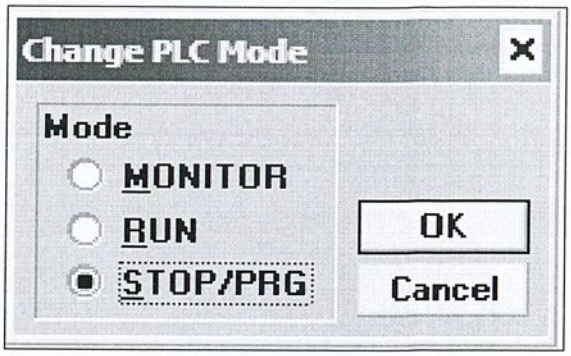



รูปที่ 2.19 Address Symbol Editor

### 2.5.11 การเลือกโหมดการทำงาน PLC

- เลือกเมนู Online/Mode
- คลิก OK

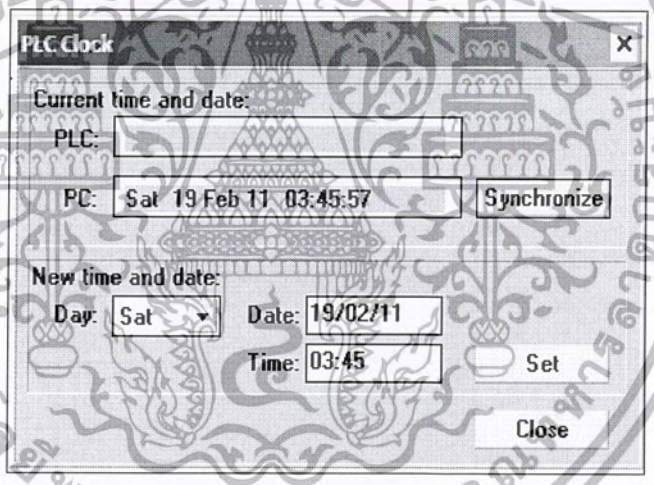
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้




รูปที่ 2.20 การเลือกโหมดการทำงาน PLC

**2.5.12 การตั้งวัน เวลา ของ PLC ให้เป็นไปตามโฮสคอมพิวเตอร์**

- เลือกเมนู Online/Set clock
- คลิก Synchronize

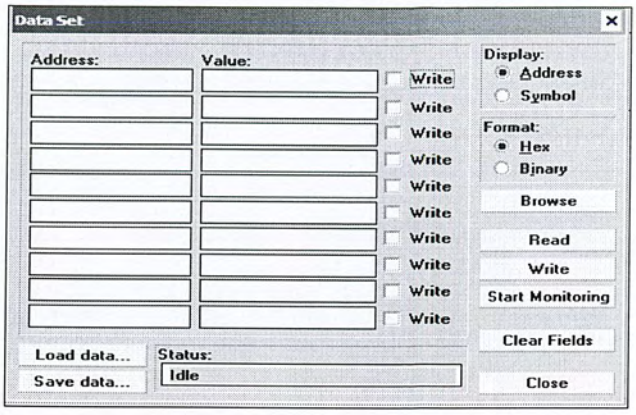



รูปที่ 2.21 การตั้งวัน เวลา ของ PLC ให้เป็นไปตามโฮส

**2.5.13 การกำหนดข้อมูลแบบเวิลด์ลงหน่วยความจำ IR ของ PLC**

- เลือกเมนู Online/Set
- ใส่ข้อมูลรูปภาพ
- คลิก write

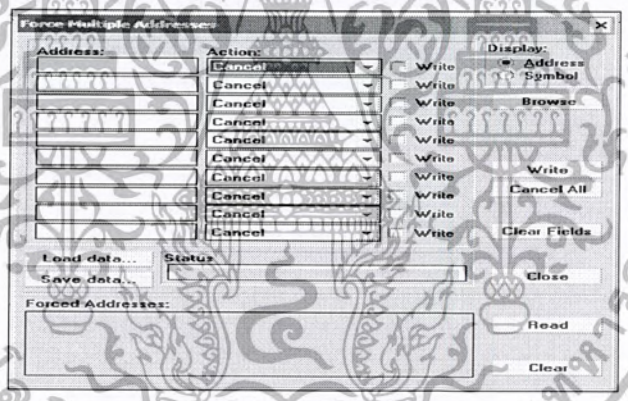
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้




**รูปที่ 2.22** การกำหนดข้อมูลแบบเวิลด์ลงหน่วยความจำ IR ของ PLC

**2.5.14 การ Force Set/Reset ข้อมูลแบบบิตลงหน่วยความจำเอาท์พุต PLC**

- เลือกเมนู Online/Force
- ใส่ข้อมูลรูปภาพ
- คลิก write

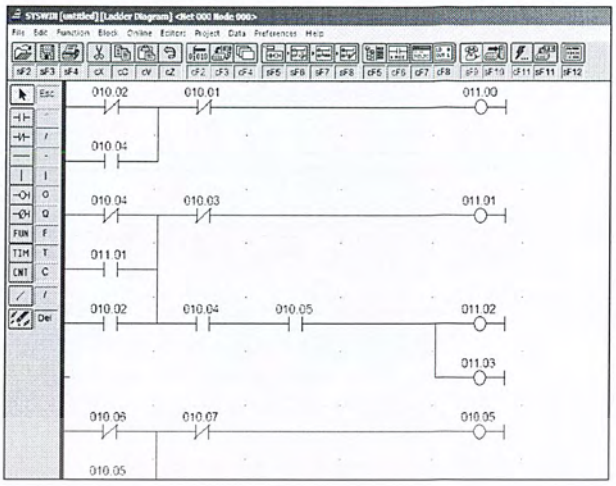



**รูปที่ 2.23** การ Force Set/Reset

**2.5.15 การเขียนโปรแกรมแลดเดอร์**

ทุกครั้งที่ขึ้น RUNG ใหม่ ให้ทำการเลือกเมนู Block/Insetnetwork/below จึงจะเขียนโปรแกรมต่อได้หรือ กล่าวได้อีกนัยหนึ่งว่า การเริ่มต้น RUNG คือ การ เริ่มต้น Network ในโปรแกรมนั้นเอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

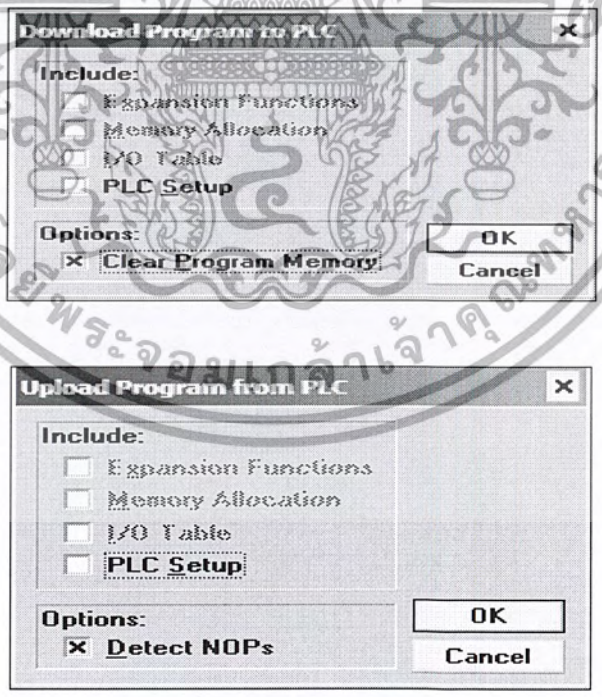



รูปที่ 2.24 แลดเดอร์

**2.5.16 การ Up-Down Load**

เป็นการส่งผ่านโปรแกรมระหว่าง PLC กับโฮสคอมพิวเตอร์

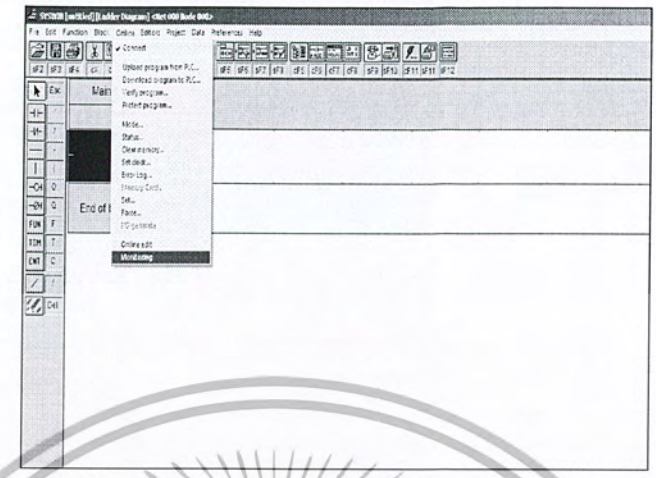
- เลือกเมนู Online/Download to PLC
- หรือ Online/Upload to PLC




รูปที่ 2.25 การ Up-Down Load

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

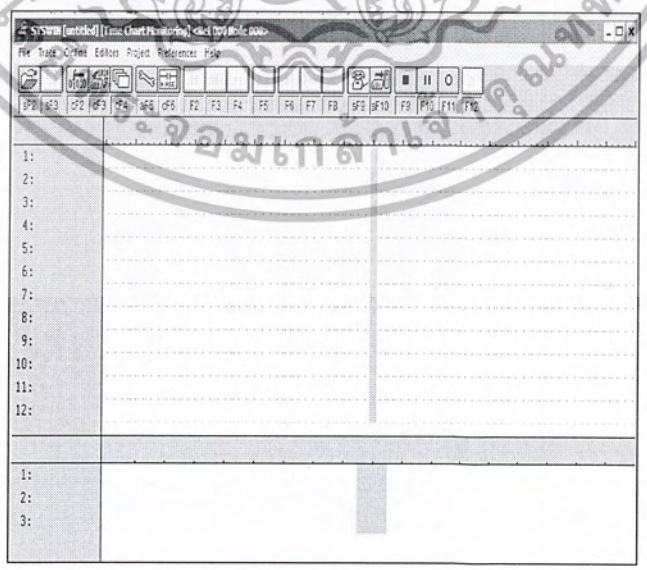
### 2.5.17 การตรวจสอบการทำงานในรูปแบบแลดเดอร์โปรแกรม



**รูปที่ 2.26** การตรวจสอบการทำงานในรูปแบบแลดเดอร์โปรแกรม

### 2.5.18 การตรวจสอบการทำงานในรูปของกราฟฟิค TIME CHART

- เลือกเมนู Editors/Time Chart Monitoring
- คลิก Configure Trace
- ใส่ข้อมูลตามภาพ คลิก OK
- คลิก Recordtrace เป็นการเริ่มต้นทำงาน
- Exit Trace Editor ออกจาก Time Chart



**รูปที่ 2.27** การตรวจสอบการทำงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 2.6 การใช้งานโปรแกรมมิ่งคอลโซล (Programming Console)

โปรแกรมมิ่งคอลโซลถือได้ว่าเป็นเครื่องมือประเภทหนึ่งของผู้ใช้งาน เพื่อพัฒนาโปรแกรมให้กับเครื่องควบคุมในส่วนของหน่วยความจำผู้ใช้งาน (User Memory) หน้าที่และประโยชน์ของโปรแกรมมิ่งคอลโซลมีดังต่อไปนี้

- ใช้ในการเขียนโปรแกรมแบบบูลีน
- ใช้ในการตรวจสอบสภาวะของหน่วยความจำต่างๆ
- ใช้ในการให้ค่า ON-OFF กับหน่วยความจำเอาท์พุทและหน่วยความจำภายใน
- ใช้ในการกำหนดหรือเปลี่ยนแปลงค่าข้อมูลแบบเวิลด์(World Data)ในหน่วยความจำต่างๆ

ตัวอย่างการใช้โปรแกรมมิ่งคอนโซล

### 2.6.1 การลบจอแสดงผล

| เลือกคีย์สวิทช์ให้เป็นโหมด โปรแกรม | RUN | MONITOR | PROGRAM |
|---|---|---|---|
| | × | × | o |

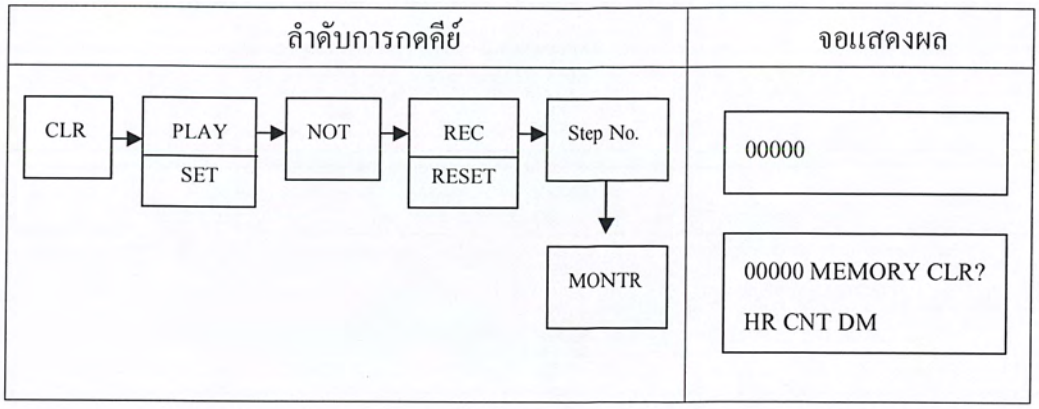
| ลำดับการกดคีย์ | จอแสดงผล |
|---|---|
| CLR → MONTR → CLR | <PROGRAM><br>PASSWORD |

### 2.6.2 การลบโปรแกรม (All Clear Program)

ถ้าผู้ใช้งานมีความประสงค์จะลบโปรแกรมเก่าออกจากหน่วยความจำในพื้นที่ของ RAM ส่วนของโปรแกรมผู้ใช้งานและส่วนของหน่วยความจำดาต้า ที่มี Bettery Back-Up สามารถกดคีย์ที่โปรแกรมมิ่งคอนโซลดังนี้

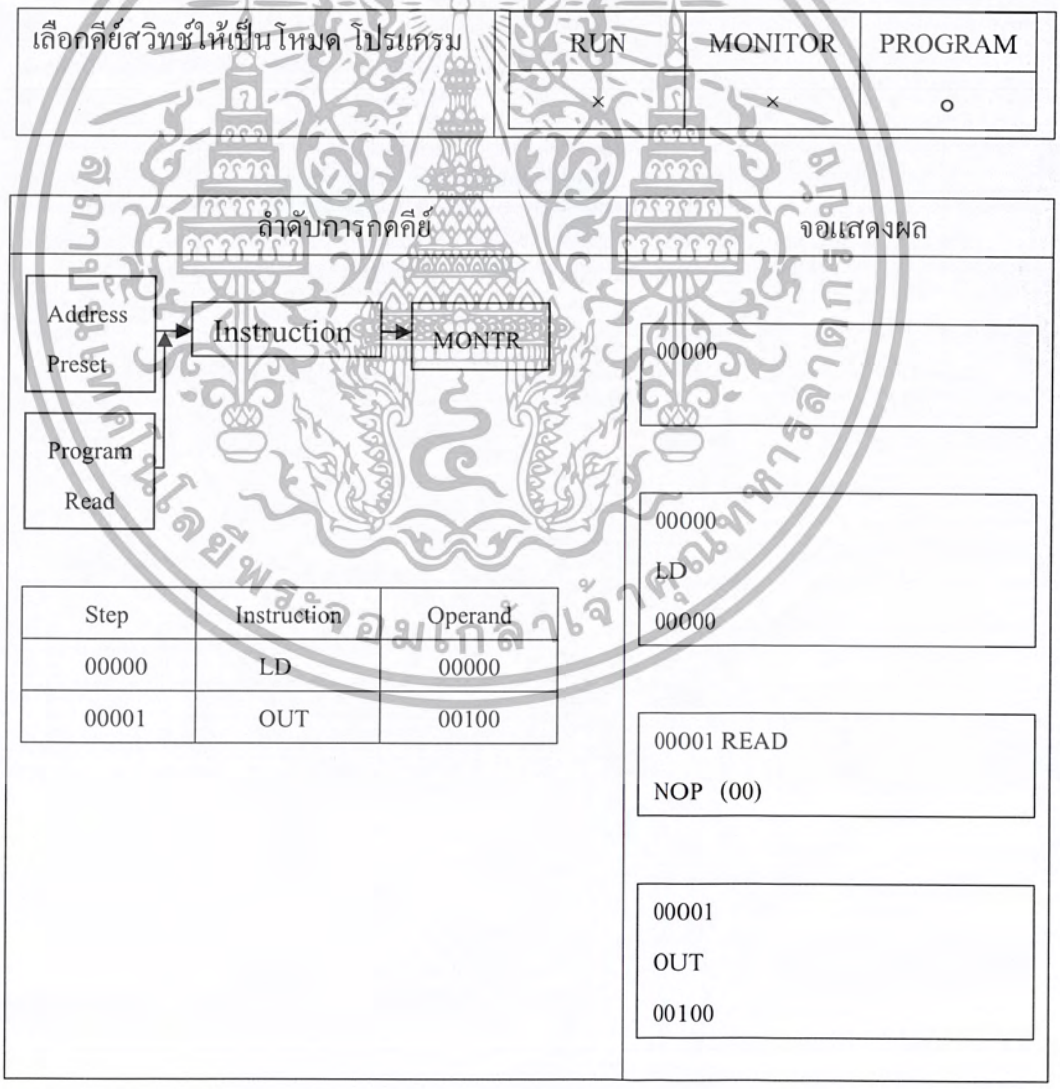
| เลือกคีย์สวิทช์ให้เป็นโหมด โปรแกรม | RUN | MONITOR | PROGRAM |
|---|---|---|---|
| | × | × | o |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้




**2.6.3** การเขียนโปรแกรม

ใช้ในการเก็บโปรแกรมบูลีนซึ่งเป็นโปรแกรมจำผู้ใช้งานลงสู่หน่วยความจำหลังจากที่กดคีย์ Write แล้วจะเป็นการเพิ่มค่า Address ทีละ Address โดยอัตโนมัติ



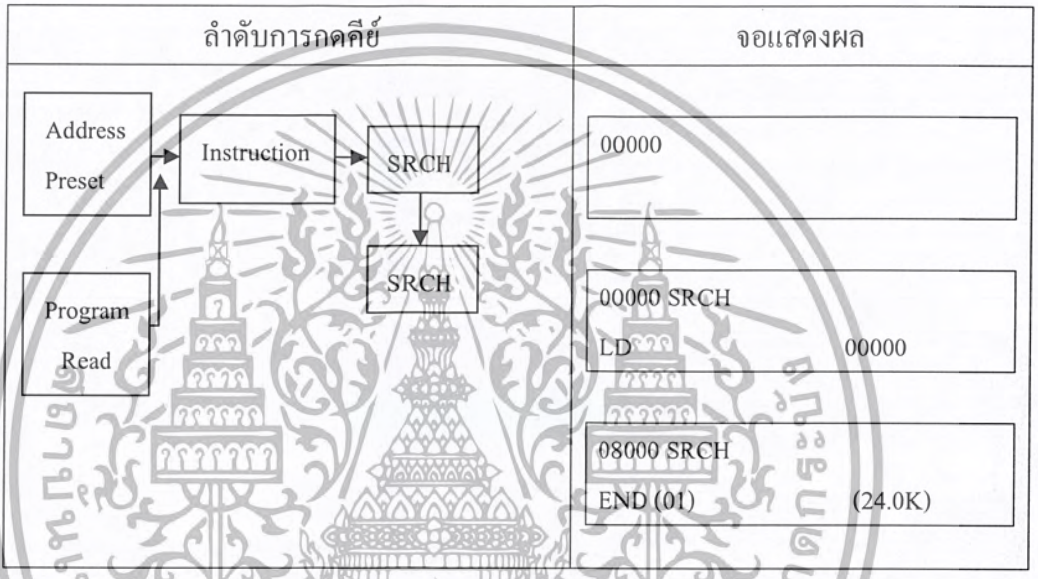

| Step | Instruction | Operand |
|---|---|---|
| 00000 | LD | 00000 |
| 00001 | OUT | 00100 |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**2.6.4 การค้นหาคำสั่ง**

ใช้ในกรณีที่ผู้ใช้เขียนโปรแกรมได้เขียนโปรแกรมคำสั่ง ให้กับเครื่องควบคุมเป็นจำนวนมากจนไม่สามารถจำตำแหน่ง Address คำสั่งที่ต้องการแก้ไขได้ดังนั้นผู้ใช้งานจึงทำการค้นหาคำสั่งเพื่อที่ให้จอแสดงผลจะได้ไปแสดงที่คำสั่งนั้นๆโดยตรง

| เลือกคีย์สวิทช์เป็นทุกโหมด | RUN | MONITOR | PROGRAM |
|---|---|---|---|
| | o | o | o |

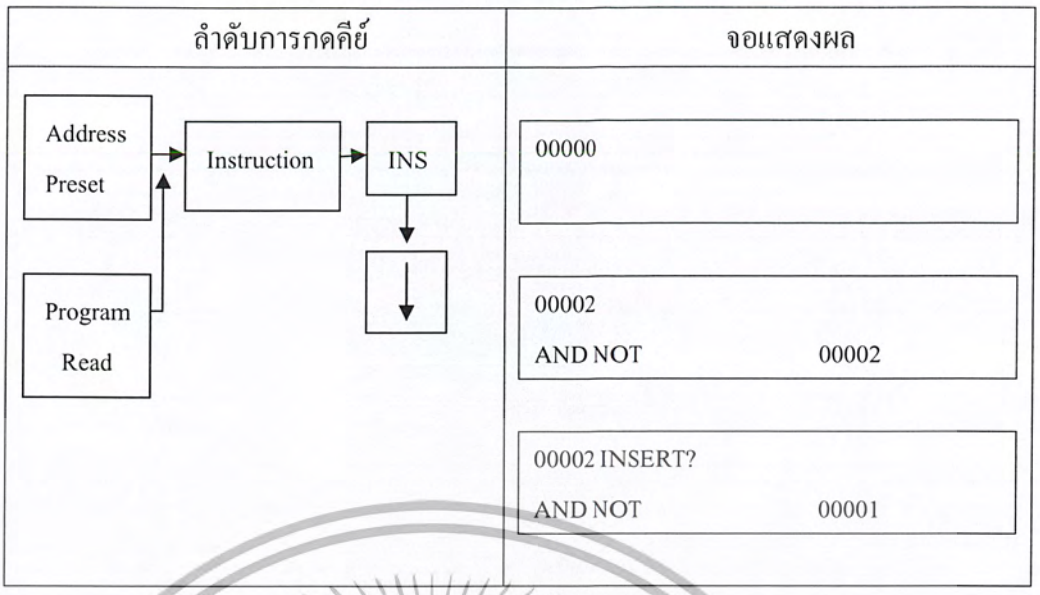



**2.6.5 การแทรกคำสั่ง**

ถ้าต้องการเพิ่มเติมคำสั่งในส่วนหนึ่งส่วนใดของโปรแกรมจะต้องให้จอแสดงผลแสดงที่บริเวณ Address นั้นๆแล้วคำสั่งที่ถูกแทรกจะอยู่หน้าคำสั่งที่แสดงผลก่อนหน้านี้

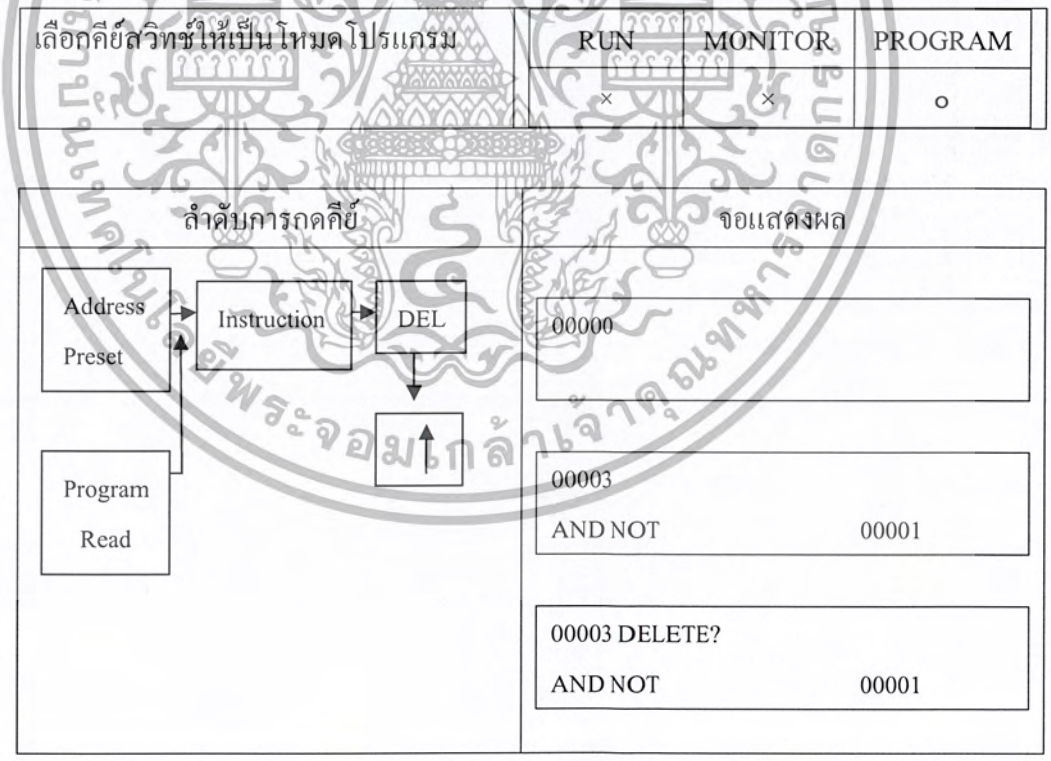
| เลือกคีย์สวิทช์ให้เป็นโหมด โปรแกรม | RUN | MONITOR | PROGRAM |
|---|---|---|---|
| | × | × | o |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้




**2.6.6 การลบคำสั่ง**

ถ้าต้องการลบคำสั่งในส่วนหนึ่งส่วนใดของโปรแกรมออก จะต้องให้จอแสดงผลที่บริเวณ Address นั้นๆ และหลังจากลบคำสั่งแล้วคำสั่งที่ต่อท้ายจะขึ้นมาแทนที่เอง

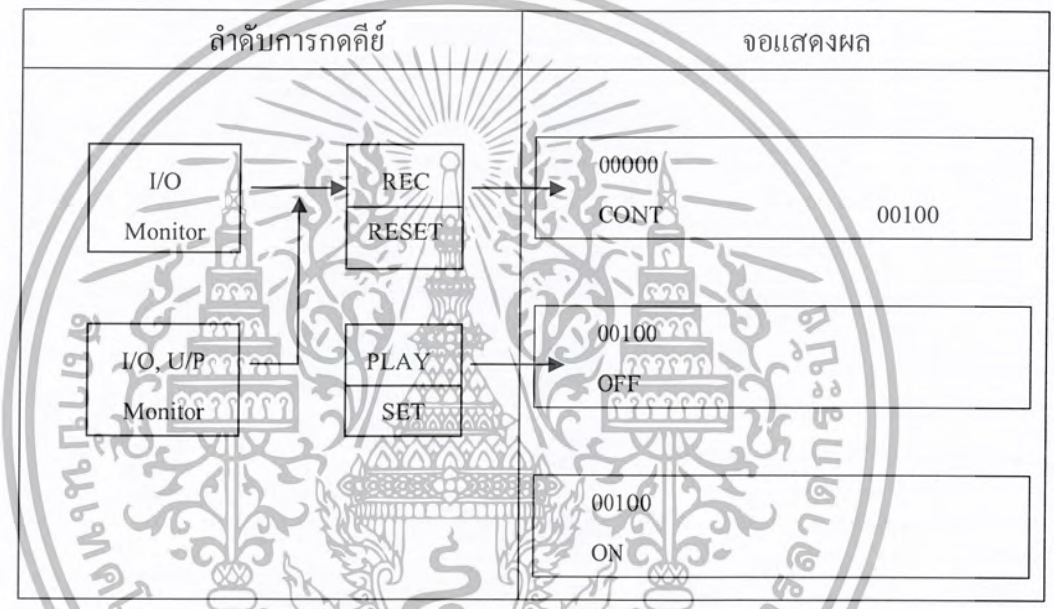



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 2.6.7 การ Force Set/Reset หรือ การกำหนดค่า ON-OFF ให้หน่วยความจำ

การ Force Set/Reset จะใช้เมื่อต้องการ ให้ค่า ON-OFF ให้ค่ากับหน่วยความจำ IR โดยเฉพาะอย่างยิ่งหน่วยความจำเอาท์พุทที่ได้ต่อร่วมกับอุปกรณ์ขับไว้แล้วเพื่อเป็นการตรวจสอบว่าอุปกรณ์ที่ต่อร่วมอยู่นั้นถูกต้องตาม I/O Assignment ที่ระบุไว้ตั้งแต่แรกหรือไม่ และถ้าไม่ถูกต้องผู้ใช้จะต้องทำการแก้ไขให้ถูกต้องเสียก่อนจึงค่อยทำการเขียนโปรแกรม ให้เครื่องควบคุม

| เลือกคีย์สวิทช์เป็นทุกโหมด | RUN | MONITOR | PROGRAM |
|---|---|---|---|
| | o | o | o |

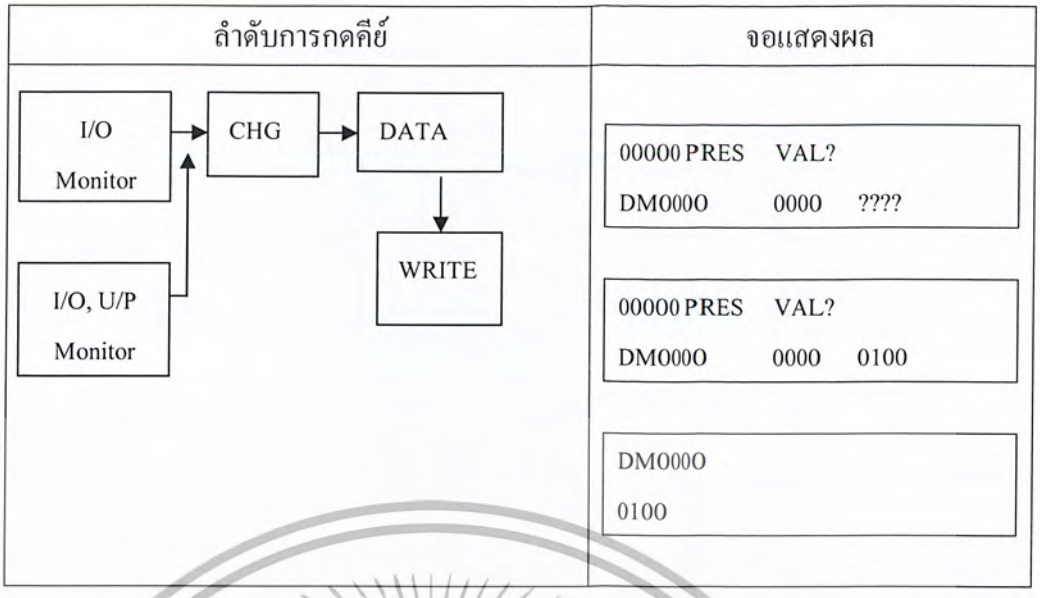



### 2.6.8 การเปลี่ยนหรือแก้ไขข้อมูลในหน่วยความจำ

ในบางครั้งมีความจำเป็นต้องแก้ไขข้อมูลใหม่ให้มากขึ้นหรือน้อยลงตามความเหมาะสมในการทำงานโปรแกรมผู้ใช้งานจะต้องเปลี่ยนหรือแก้ไขตามขั้นตอนต่อไปนี้

| เลือกคีย์ให้เป็นโหมด โมนิเตอร์ หรือ โปรแกรม | RUN | MONITOR | PROGRAM |
|---|---|---|---|
| | × | o | o |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้




## 2.7 การติดต่อสื่อสาร พอร์ตอนุกรม (Serial Port)

พอร์ตอนุกรม RS-232 จะเป็นพอร์ตของเครื่องคอมพิวเตอร์ที่มีขาต่อ (Connector) ทั้งประเภท 9 ขาและ 25 ขา และเราเรียกกันว่า COM1: และ COM2: นั่นเอง ในความเป็นจริงพอร์ตอนุกรมไม่ได้ถูกควบคุม โดยตรงจาก CPU บนเมนบอร์ด แต่การสื่อสารทั้งหมดจะถูกจัดการโดยชิป UART ( Universal Asynchronous Receiver/Transmitter) อีกทีหนึ่ง ซึ่งในปัจจุบันที่ใช้กันมากที่สุดคือเบอร์ 16550C ซึ่งเป็นเวอร์ชั่นที่ได้รับการแก้ไขข้อผิดพลาดแล้ว ซึ่งชิป UART นี้จะทำหน้าที่ในการรับส่งข้อมูลดังต่อไปนี้

การส่งข้อมูล (Data Transmission)

- รับตัวอักษรจากเครื่องคอมพิวเตอร์
- แปลงตัวอักษรให้เป็นสายข้อมูลแบบบิต (เราเรียกว่ากระบวนการ Serialization)
- สร้างเฟรมข้อมูลโดยการเพิ่มบิตที่จำเป็นสำหรับการสื่อสารและการตรวจสอบ เช่น บิต Start, Stop และ Parity เป็นต้น
- ส่งผ่านเฟรมข้อมูลที่สร้างขึ้นมาแล้วจากขั้นตอนที่ผ่านมา ด้วยความเร็วของโมเด็มหรือพอร์ตอนุกรม (Baud Rate)
- แสดงสถานะความพร้อมที่จะรับข้อมูลตัวอักษรถัดไปให้กับเครื่องคอมพิวเตอร์

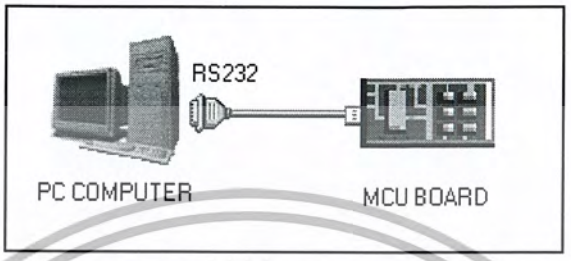
- การรับข้อมูล (Data Receiver)

- รับตัวอักษรจากอินเตอร์เฟช
- ตรวจสอบความถูกต้องของเฟรมข้อมูลตามมาตรฐานที่กำหนด โดยถ้าหากเฟรมข้อมูลมีรูปแบบที่ไม่ถูกต้องก็จะมีการแจ้งข้อผิดพลาดทันที
- ตรวจสอบความถูกต้องของพาริตี้
- แปลงสายข้อมูลแบบบิตเป็นตัวอักษร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

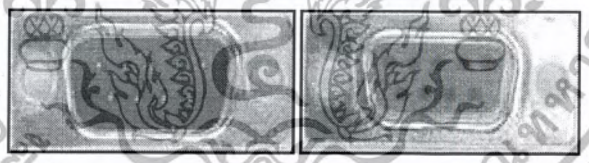
- ส่งตัวอักษรให้กับเครื่องคอมพิวเตอร์
- แสดงสถานะความพร้อม ที่จะรับข้อมูลตัวอักษรถัดไปให้กับอินเตอร์เฟช

### 2.7.1 การใช้งานพอร์ตอนุกรม RS232




**รูปที่ 2.28**

การสื่อสารแบบอนุกรมนับว่ามีความสำคัญต่อการใช้งานไมโครคอนโทรลเลอร์มากเพราะสามารถใช้แป้นพิมพ์ และจอภาพของ PC เป็น อินพุต และ เอาต์พุต ในการติดต่อหรือ ควบคุมไมโครคอนโทรลเลอร์ด้วยสัญญาณอย่างน้อยเพียง 3 เส้นเท่านั้นคือสายส่งสัญญาณ TX, สายรับสัญญาณRXและสาย GND โดยปกติพอร์ตอนุกรม RS-232C จะสามารถต่อสายได้ยาว 50 ฟุต โดยประมาณ ขึ้นอยู่กับชนิดของสายสัญญาณระยะทางและปริมาณสัญญาณรบกวน



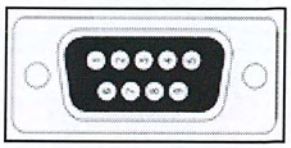
**รูปที่ 2.29**พอร์ตอนุกรม RS-232C

พอร์ตอนุกรมของ PC DB9 ตัวผู้ (Male)

พอร์ตอนุกรมของอุปกรณ์ภายนอก DB9 ตัวเมีย(Female)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แสดงการจัดขา ของคอนเน็กเตอร์ อนุกรมแบบ DB9 และหน้าที่การใช้งานต่างๆ

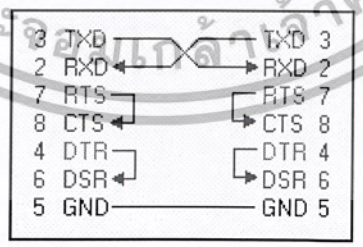


**รูปที่ 2.30** DB9 ตัวผู้ เมื่อมองจากด้านหลัง

ตารางที่ 2.5 แสดงการจัดขา ของคอนเน็กเตอร์ อนุกรมแบบ DB9 และหน้าที่การใช้งานต่างๆ

| Pin | Description | Type |
|---|---|---|
| 1 | Data Carrier Detect (DCD) | Input |
| 2 | Received Data (RXD) | Input |
| 3 | Transmitted Data (TXD) | Output |
| 4 | Data Terminal Ready (DTR) | Output |
| 5 | Signal Ground (GND) | Input |
| 6 | Data Set Ready (DSR) | Input |
| 7 | Request To Send (RTS) | Output |
| 8 | Clear to Send (CTS) | Input |
| 9 | Ring Indicator (RI) | Input |

**2.7.2 การเชื่อมต่ออุปกรณ์ภายนอกเข้ากับคอมพิวเตอร์ด้วยสาย DB9**




**รูปที่ 2.31** การทำงานของขาสัญญาณ DB9

TXD เป็นขาที่ใช้ส่งข้อมูล

RXD เป็นขาที่ใช้รับข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

DTR แสดงสภาวะพอร์ตว่าเปิดใช้งาน, DSR ตรวจสอบว่าพอร์ต ที่ติดต่อด้วยเปิดอยู่หรือไม่ เมื่อเปิดพอร์ตอนุกรมขา DTR จะ ON เพื่อให้อุปกรณ์ได้รับทราบว่าต้องการติดต่อด้วย ในขณะเดียวกันก็จะตรวจสอบขา DSR ว่าอุปกรณ์พร้อมหรือไม่

RTS แสดงสภาวะพอร์ตว่าต้องการส่งข้อมูล, CTS ตรวจสอบว่าพอร์ตที่ติดต่ออยู่ ต้องการส่งข้อมูลหรือไม่ เมื่อต้องการส่งข้อมูลขา RTS จะ ON และจะส่งข้อมูลออกที่ขา TXD เมื่อส่งเสร็จก็จะ OFF ในขณะเดียวกันก็จะตรวจสอบขา CTS ว่าอุปกรณ์ต้องการที่จะส่งข้อมูลหรือไม่

**GND** ขา ground

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# บทที่ 3
# วิธีการดำเนินงาน

## 3.1 กล่าวนำ

โครงสร้างของระบบจำลองการควบคุมระดับของเหลวจะประกอบด้วย

1. กระบวนการจำลองการควบคุมระดับ (Process level) สามารถนำไปประยุกต์ใช้งานได้ เช่นกระบวนการกลั่น กระบวนการผลิตยา เป็นต้น ซึ่งระบบจำลองที่สร้างขึ้นมาโดยใช้โปรแกรม Delphi 5 ในการจำลอง

2. การวัด (field Instrument level) จะจำลองในระบบว่ามีการติดตัวเซนเซอร์ไว้ที่ระดับ low และระดับ high

3. ตัวควบคุม (Controller) ใช้พีแอลซีในการควบคุม

4. การเชื่อมต่อกัน (Connection) เป็นการเชื่อมต่อกันระหว่างคอมพิวเตอร์กับพีแอลซีโดยใช้โปรโตคอลในการเชื่อมต่อ

สำหรับปริญญานิพนธ์ฉบับนี้สามารถแบ่งส่วนการทำงานได้ 2 ส่วน คือ ในส่วนของฮาร์ดแวร์ และส่วนของซอฟท์แวร์

**- ด้านของฮาร์ดแวร์ ประกอบด้วย**

1. เครื่องควบคุมพีแอลซี (Programmable Logic Controller) รุ่น C200HE ของ OMRON
2. โปรแกรมมิ่งคอลโซล (Programming Console) รุ่น PRO01 ของ OMRON

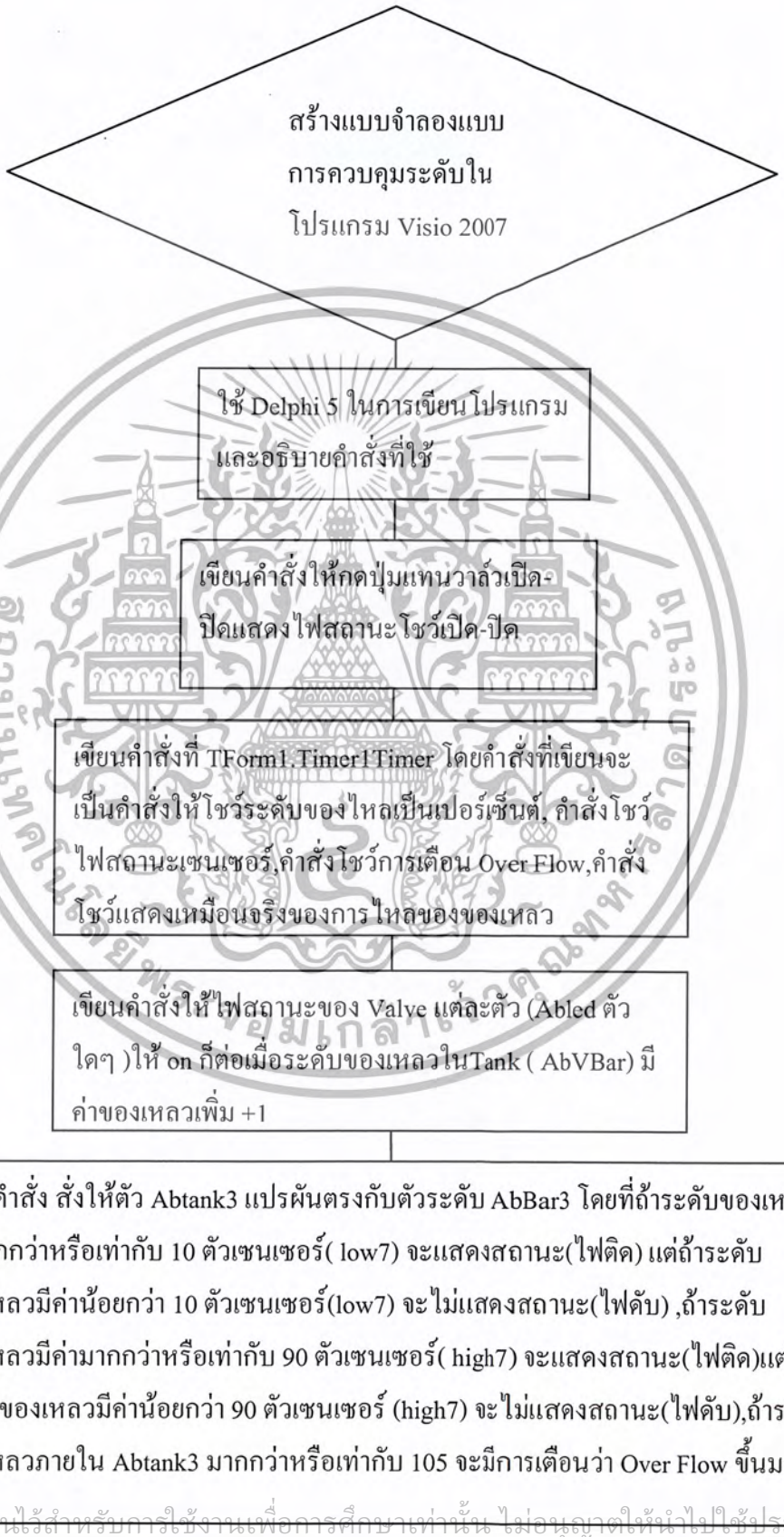
**- ด้านซอฟแวร์ ประกอบด้วย**

1. โปรแกรม Delphi 5
2. โปรแกรม Syswin3.4

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 3.2 ลำดับขั้นตอนการสร้างแบบจำลอง

แสดงขั้นตอนการกำหนดการเขียนคำสั่งในโปรแกรม Delphi 5

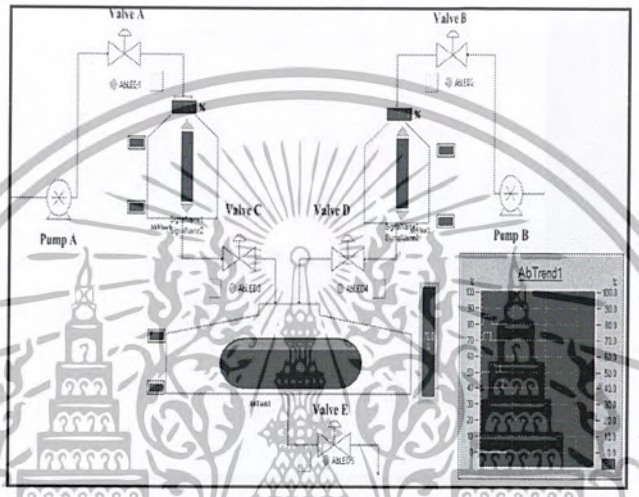



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 3.3 การออกแบบจำลอง

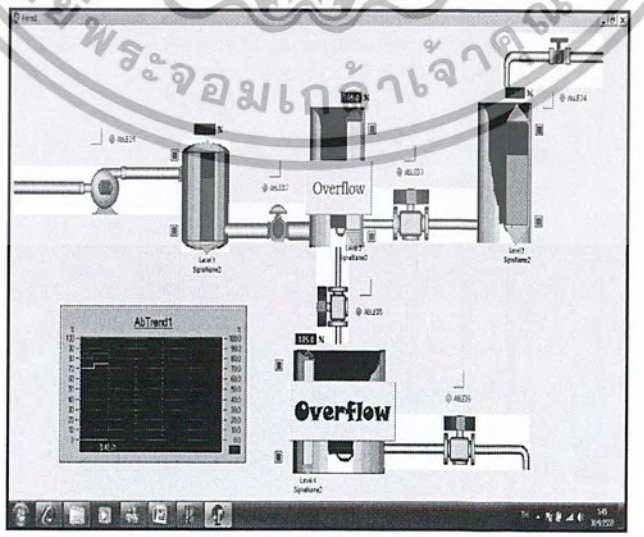
ขั้นตอนการออกแบบจำลองระบบควบคุมของของไหลเป็นกราฟฟิค โดยใช้โปรแกรม Visio 2007 และ โปรแกรม Delphi 5ในการจำลองระบบในส่วนการจำลองระบบควบคุมของของไหลได้ออกแบบมา 3 ระบบ

### 3.3.1 แบบจำลองระบบควบคุมระดับแบบที่ 1




**รูปที่ 3.1** แบบจำลองระบบควบคุมระดับแบบที่ 1

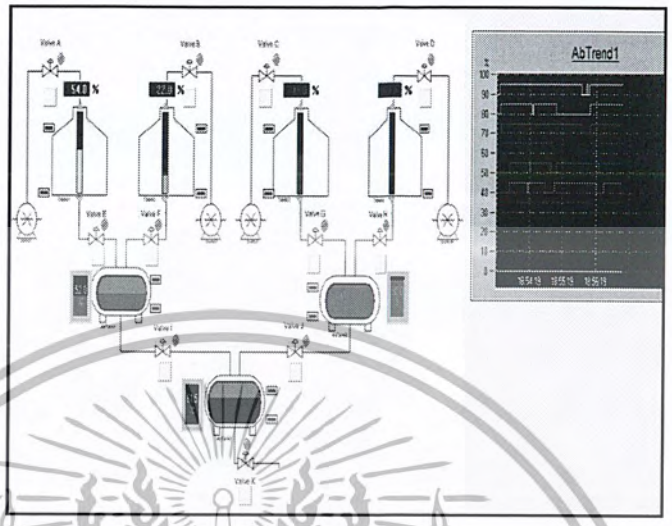
### 3.3.2 แบบจำลองระบบควบคุมระดับแบบที่ 2




**รูปที่ 3.2** แบบจำลองระบบควบคุมระดับแบบที่ 2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

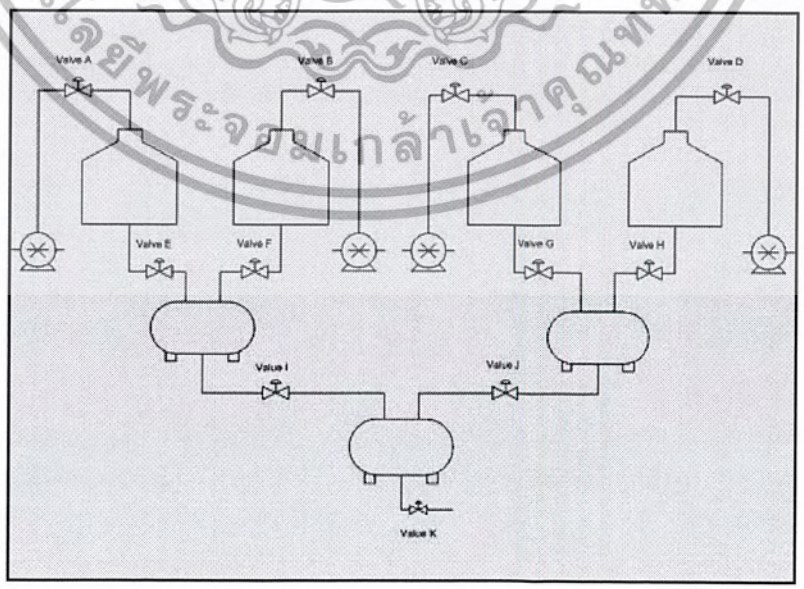
### 3.3.3 แบบจำลองระบบควบคุมแบบที่ 3




**รูปที่ 3.3** แบบจำลองระบบควบคุมระดับแบบที่ 3

### 3.3.4 วิธีการจำลองระบบควบคุมของของไหลเป็นกราฟฟิคซึ่ง เลือกระบบที่ 3 มาใช้ในการ Simulator

1. สร้างแบบจำลองแบบการวัดระดับในโปรแกรม Visio 2007

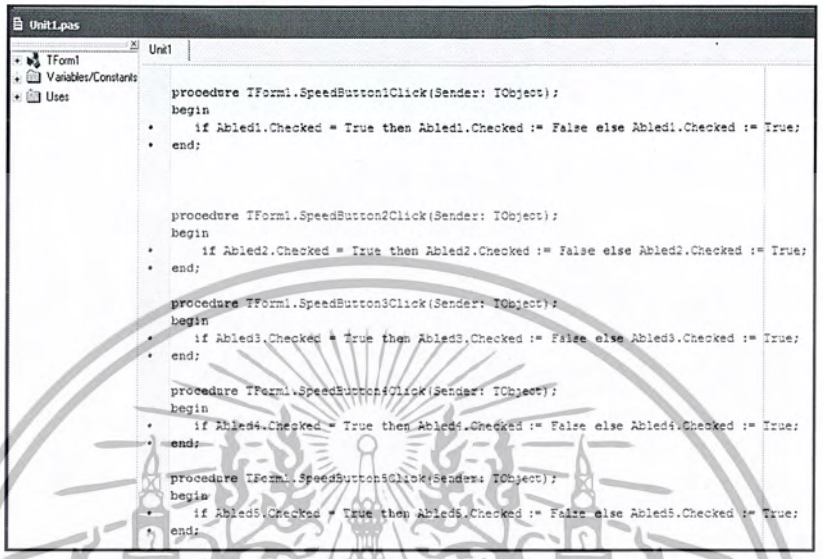



**รูปที่ 3.4** แบบจำลองแบบการวัดระดับในโปรแกรม Visio 2007

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. ใช้ Delphi 5 ในการเขียนโปรแกรมและอธิบายคำสั่งที่ใช้ดังนี้

คำสั่งที่เขียนด้วยโปรแกรม Delphi 5




**รูปที่ 3.5** การเขียนโปรแกรมและอธิบายคำสั่ง

### 3.3.5 การเขียนโปรแกรมและอธิบายคำสั่ง

1. เขียนคำสั่งให้กดปุ่มแทนวาล์วเปิด-ปิดแสดงไฟสถานะโชว์เปิด-ปิดได้ดังนี้

procedure TForm1.SpeedButton1Click(Sender: TObject);

begin

if Abled1.Checked = True then Abled1.Checked := False else Abled1.Checked := True;

end;

procedure TForm1.SpeedButton2Click(Sender: TObject);

begin

if Abled2.Checked = True then Abled2.Checked := False else Abled2.Checked := True;

end;

procedure TForm1.SpeedButton3Click(Sender: TObject);

begin

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
if Abled3.Checked = True then Abled3.Checked := False else Abled3.Checked
:= True;
end;
procedure TForm1.SpeedButton4Click(Sender: TObject);

begin
if Abled4.Checked = True then Abled4.Checked := False else Abled4.Checked
:= True;
end;
procedure TForm1.SpeedButton5Click(Sender: TObject);
begin
if Abled5.Checked = True then Abled5.Checked := False else Abled5.Checked
:= True;
end;
procedure TForm1.SpeedButton6Click (Sender: TObject);
begin
if Abled6.Checked = True then Abled6.Checked := False else Abled6.Checked
:= True;
end;
procedure TForm1.SpeedButton7Click (Sender: TObject);
begin
if Abled7.Checked = True then Abled7.Checked := False else Abled7.Checked
:= True;
end;

procedure TForm1.SpeedButton8Click (Sender: TObject);
begin
if Abled8.Checked = True then Abled8.Checked := False else Abled8.Checked
:= True;
end;
procedure TForm1.SpeedButton9Click (Sender: TObject);
begin
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
if Abled9.Checked = True then Abled9.Checked := False else Abled9.Checked
:= True;
end;
procedure TForm1.SpeedButton10Click (Sender: TObject);
begin
if Abled10.Checked = True then Abled10.Checked := False else
Abled10.Checked := True;
end;
procedure TForm1.SpeedButton11Click (Sender: TObject);
begin
if Abled11.Checked = True then Abled11.Checked := False else
Abled11.Checked := True;
end;
```

2. เขียนคำสั่งที่ TForm1.Timer1Timer โดยคำสั่งที่เขียนจะเป็นคำสั่งให้โชว์ระดับของไหลเป็นเปอร์เซ็นต์,คำสั่งโชว์ไฟสถานะเซนเซอร์,คำสั่งโชว์การเตือน Over Flow,คำสั่งโชว์แสดงเหมือนจริงของการไหลของของเหลว

```
procedure TForm1.Timer1Timer (Sender: TObject);
```

3. เขียนคำสั่งให้ไฟสถานะของ Valve A (Abled1) ให้ on ก็ต่อเมื่อระดับของเหลวใน Tank ( AbVBar1) มีค่าของเหลวเพิ่ม +1

```
begin
if Abled1.Checked = True then
begin
AbVBar1.value := AbVBar1.value+1;
end;
```

4. เขียนคำสั่งให้ไฟสถานะ Valve B (Abled2) ให้ on ก็ต่อเมื่อระดับของเหลวใน Tank (AbVBar2) มีค่าของเหลวเพิ่ม +1

```
if Abled2.Checked = True then
begin
AbVBar2.value := AbVBar2.value+1;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
end;
```

5. เขียนคำสั่งให้ไฟสถานะ Valve C (Abled3) ให้ on ก็ต่อเมื่อระดับของเหลวใน Tank ( AbVBar3) มีค่าของเหลวเพิ่ม +1

```
if Abled3.Checked = True then
begin
AbVBar3.value := AbVBar3.value+1;
end;
```

6. เขียนคำสั่งให้ไฟสถานะ Valve D (Abled4) ให้ on ก็ต่อเมื่อระดับของเหลวใน Tank ( AbVBar4 ) มีค่าของเหลวเพิ่ม +1

```
if Abled4.Checked = True then
begin
AbVBar4.value := AbVBar4.value+1;
end;
```

7. เขียนคำสั่งให้ ไฟสถานะ Valve E (Abled5) ให้ on ก็ต่อเมื่อระดับของเหลวใน Tank (AbVBar1) มีค่าของเหลวลด -1

```
if Abled5.Checked = True then
begin
AbVBar1.value := AbVBar1.value-1;
if AbVBar1.value <=0 then AbVBar1.value := 0;
end;
```

8. เขียนคำสั่งให้ไฟสถานะ Valve E (Abled5) ให้ on ก็ต่อเมื่อระดับของเหลวใน AbTank1 ( AbVbar1) มีค่ามากกว่า 0 ซึ่ง Abtank1 จะมีของเหลวไหล +0.5ในด้านของ valve E

```
if (Abled5.Checked = True) and (AbVbar1.Value > 0) then
begin
Abtank1.value := Abtank1.value+0.5;
end;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

9. เขียนคำสั่งให้ไฟสถานะ Valve F (Abled6) ให้ on ก็ต่อเมื่อระดับของเหลวใน Tank (AbVBar2) มีค่าของเหลวลด -1

```
if Abled6.Checked = True then
begin
AbVBar2.value := AbVBar2.value-1;
if AbVBar2.value <=0 then AbVBar2.value := 0;
end;
```

10. เขียนคำสั่งให้ไฟสถานะ Valve F (Abled6) ให้ on ก็ต่อเมื่อระดับของเหลวใน Tank ( AbVbar2) มีค่ามากกว่า 0 ซึ่ง Abtank1 จะมีของเหลวไหล +0.5ในด้านของ valve F

```
if (Abled6.Checked = True) and (AbVbar2.Value > 0) then
begin
Abtank1.value := Abtank1.value+0.5;
end;
```

11. เขียนคำสั่งให้ไฟสถานะ Valve G (Abled7) ให้ on ก็ต่อเมื่อระดับของเหลวใน Tank ( AbVBar3) มีค่าของเหลวลด -1

```
if Abled7.Checked = True then
begin
AbVBar3.value := AbVBar3.value-1;
if AbVBar3.value <=0 then AbVBar3.value := 0;
end;
```

12. เขียนคำสั่งให้ไฟสถานะ Valve G (Abled7) ให้ on ก็ต่อเมื่อระดับของเหลวใน Tank (AbVbar3) มีค่ามากกว่า 0 ซึ่ง Abtank2 จะมีของเหลวไหล +0.5ในด้านของ valve G

```
if (Abled7.Checked = True) and (AbVbar3.Value > 0) then
begin
Abtank2.value := Abtank2.value+0.5;
end;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

13. เขียนคำสั่งให้ไฟสถานะ Valve H (Abled8) ให้ on ก็ต่อเมื่อระดับของเหลวใน Tank ( AbVBar4 ) มีค่าของเหลวลด -1

```
if Abled8.Checked = True then
begin
AbVBar4.value := AbVBar4.value-1;
if AbVBar4.value <=0 then AbVBar4.value := 0;
end;
```

14. เขียนคำสั่งให้ไฟสถานะ Valve H ( Abled8) ให้ on ก็ต่อเมื่อระดับของเหลวใน Tank ( AbVbar4) มีค่ามากกว่า 0 ซึ่ง Abtank2 จะมีของเหลวไหล +0.5ในด้านของvalve H

```
if (Abled8.Checked = True) and (AbVbar4.Value > 0) then
begin
Abtank2.value := Abtank2.value+0.5;
end;
```

15. เขียนคำสั่งให้ไฟสถานะ Valve I (Abled9) ให้ on ก็ต่อเมื่อระดับของเหลวใน Abtank1 มีค่าของเหลวลด -1

```
if Abled9.Checked = True then
begin
Abtank1.value := Abtank1.value-1;
if Abtank1.value <=0 then Abtank1.value := 0;
end;
```

16. เขียนคำสั่งให้ไฟสถานะValve I ( Abled9) ให้ on ก็ต่อเมื่อระดับของเหลวใน Abtank1 มีค่ามากกว่า 0 ซึ่งระดับของเหลวใน Abtank3 จะมีของเหลวไหล +0.5ในด้านของ valve I

```
if (Abled9.Checked = True) and (Abtank1.Value > 0) then
begin
Abtank3.value := Abtank3.value+0.5;
end;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

17. เขียนคำสั่งให้ไฟสถานะ Valve J (Abled10) ให้ on ก็ต่อเมื่อระดับของเหลวใน Abtank2 มีค่าของเหลวลด -1

```
if Abled10.Checked = True then
begin
Abtank2.value := Abtank2.value-1;
if Abtank2.value <=0 then Abtank2.value := 0;
end;
```

18. เขียนคำสั่งให้ไฟสถานะ Valve J (Abled10) ให้ on ก็ต่อเมื่อระดับของเหลวใน Abtank2 มีค่ามากกว่า 0 ซึ่งระดับของเหลวในAbtank3 จะมีของเหลวไหล +0.5 ในด้านของ valve J

```
if (Abled10.Checked = True) and (Abtank2.Value > 0) then
begin
Abtank3.value := Abtank3.value+0.5;
end;
```

19. เขียนคำสั่งให้ไฟสถานะValve K (Abled11) ให้ on ก็ต่อเมื่อระดับของเหลวใน Abtank3 มีค่าของเหลวลด -1

```
if Abled11.Checked = True then
begin
AbTank3.value := AbTank3.value -1;
if AbTank3.value <=0 then AbTank3.value := 0;
end;
```

20. คำสั่งที่เขียนคือ สั่งให้ตัว Abtank1 แปรผันตรงกับ AbBar1 โดยที่ถ้าระดับของเหลวมีค่ามากกว่าหรือเท่ากับ 10 ตัวเซนเซอร์ ( low1) จะแสดงสถานะ (ไฟติด) แต่ถ้าระดับของเหลวมีค่าน้อยกว่า 10 ตัวเซนเซอร์ (low1) จะไม่แสดงสถานะ (ไฟดับ) ,ถ้าระดับของเหลวมีค่ามากกว่าหรือเท่ากับ 90 ตัวเซนเซอร์ ( high1) จะแสดงสถานะ (ไฟติด) แต่ถ้าระดับของเหลวมีค่าน้อยกว่า 90 ตัวเซนเซอร์ (high1) จะไม่แสดงสถานะ (ไฟดับ), ถ้าระดับของเหลวภายใน Abtank1 มากกว่าหรือเท่ากับ 105 จะมีการเตือนว่า Over Flow ขึ้นมา

```
AbBar1.value := Abtank1.value;
if AbVBar1.value >= 10 then low1.Active := true else low1.Active := false;
if AbVBar1.value >= 90 then high1.Active := true else high1.Active := false;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
if AbVbar1.Value >= 105 then
begin
AbVbar1.Value := 105;
Button1.Visible := True;
end
else
begin
Button1.Visible := False;
end;
```

21. คำสั่งที่เขียนคือ สั่งให้ตัว Abtank2 แปรผันตรงกับ AbBar2 โดยที่ถ้าระดับของเหลวมีค่ามากกว่าหรือเท่ากับ 10 ตัวเซนเซอร์ ( low2) จะแสดงสถานะ (ไฟติด) แต่ถ้าระดับของเหลวมีค่าน้อยกว่า 10 ตัวเซนเซอร์ (low2) จะไม่แสดงสถานะ (ไฟดับ) ,ถ้าระดับของเหลวมีค่ามากกว่าหรือเท่ากับ 90 ตัวเซนเซอร์( high2) จะแสดงสถานะ (ไฟติด) แต่ถ้าระดับของเหลวมีค่าน้อยกว่า 90 ตัวเซนเซอร์ (high2) จะไม่แสดงสถานะ (ไฟดับ), ถ้าระดับของเหลวภายใน Abtank2 มากกว่าหรือเท่ากับ 105 จะมีการเตือนว่า Over Flow ขึ้นมา

```
AbBar2.value := Abtank2.value;
if AbVBar2.value >= 10 then low2.Active := true else low2.Active := false;
if AbVBar2.value >= 90 then high2.Active := true else high2.Active := false;
if AbVbar2.Value >= 105 then
begin
AbVbar2.Value := 105;
Button2.Visible := True;
end
else
begin
Button2.Visible := False;
end;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

22. คำสั่งที่เขียนคือ สั่งให้ตัว AbVBar3 โดยที่ถ้าระดับของเหลวมีค่ามากกว่าหรือเท่ากับ 10 ตัวเซนเซอร์ ( low3) จะแสดงสถานะ (ไฟติด) แต่ถ้าระดับของเหลวมีค่าน้อยกว่า 10 ตัวเซนเซอร์ (low3) จะไม่แสดงสถานะ (ไฟดับ) ,ถ้าระดับของเหลวมีค่ามากกว่าหรือเท่ากับ 90 ตัวเซนเซอร์ ( high3) จะแสดงสถานะ (ไฟติด) แต่ถ้าระดับของเหลวมีค่าน้อยกว่า 90 ตัวเซนเซอร์ ( high3) จะไม่แสดงสถานะ (ไฟดับ), ถ้าระดับของเหลวภายใน AbVBar3 มากกว่าหรือเท่ากับ 105 จะมีการเตือนว่า Over Flow ขึ้นมา

```
if AbVBar3.value >= 10 then low3.Active := true else low3.Active := false;
if AbVBar3.value >= 90 then high3.Active := true else high3.Active := false;
if AbVbar3.Value >= 105 then
begin
AbVbar3.Value := 105;
Button3.Visible := True;
end
else
begin
Button3.Visible := False;
end;
```

23. คำสั่งที่เขียนคือ สั่งให้ตัว AbVBar4 โดยที่ถ้าระดับของเหลวมีค่ามากกว่าหรือเท่ากับ 10 ตัวเซนเซอร์ ( low4) จะแสดงสถานะ (ไฟติด) แต่ถ้าระดับของเหลวมีค่าน้อยกว่า 10 ตัวเซนเซอร์ (low4) จะไม่แสดงสถานะ (ไฟดับ) , ถ้าระดับของเหลวมีค่ามากกว่าหรือเท่ากับ 90 ตัวเซนเซอร์ ( high4) จะแสดงสถานะ (ไฟติด) แต่ถ้าระดับของเหลวมีค่าน้อยกว่า 90 ตัวเซนเซอร์ (high4) จะไม่แสดงสถานะ (ไฟดับ),ถ้าระดับของเหลวภายใน AbVBar4 มากกว่าหรือเท่ากับ 105 จะมีการเตือนว่า Over Flow ขึ้นมา

```
if AbVBar4.value >= 10 then low4.Active := true else low4.Active := false;
if AbVBar4.value >= 90 then high4.Active := true else high4.Active := false;
if AbVbar4.Value >= 105 then
begin
AbVbar4.Value := 105;
Button4.Visible := True;
end
else
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
begin
Button4.Visible := False;
end;
```

24. คำสั่งที่เขียนคือ สั่งให้ตัว Abtank1 โดยที่ถ้าระดับของเหลวมีค่ามากกว่าหรือเท่ากับ 10 ตัวเซนเซอร์ ( low5) จะแสดงสถานะ (ไฟติด) แต่ถ้าระดับของเหลวมีค่าน้อยกว่า 10 ตัวเซนเซอร์ (low5) จะไม่แสดงสถานะ (ไฟดับ) ,ถ้าระดับของเหลวมีค่ามากกว่าหรือเท่ากับ 90 ตัวเซนเซอร์ ( high5) จะแสดงสถานะ (ไฟติด) แต่ถ้าระดับของเหลวมีค่าน้อยกว่า 90 ตัวเซนเซอร์ (high5) จะไม่แสดงสถานะ (ไฟดับ), ถ้าระดับของเหลวภายใน Abtank1 มากกว่าหรือเท่ากับ 105 จะมีการเตือนว่า Over Flow ขึ้นมา

```
if abtank1.value >= 10 then low5.Active := true else low5.Active := false;
if abtank1.value >= 90 then high5.Active := true else high5.Active := false;
if abtank1.Value >= 105 then
begin
Abtank1.Value := 105;
Button5.Visible := True;
end
else
begin
Button5.Visible := False;
end;
```

25. คำสั่งที่เขียนคือ สั่งให้ตัว Abtank2 โดยที่ถ้าระดับของเหลวมีค่ามากกว่าหรือเท่ากับ 10 ตัวเซนเซอร์ ( low6) จะแสดงสถานะ (ไฟติด) แต่ถ้าระดับของเหลวมีค่าน้อยกว่า 10 ตัวเซนเซอร์ (low6) จะไม่แสดงสถานะ (ไฟดับ) , ถ้าระดับของเหลวมีค่ามากกว่าหรือเท่ากับ 90 ตัวเซนเซอร์ ( high6) จะแสดงสถานะ (ไฟติด) แต่ถ้าระดับของเหลวมีค่าน้อยกว่า 90 ตัวเซนเซอร์ (high6) จะไม่แสดงสถานะ (ไฟดับ), ถ้าระดับของเหลวภายใน Abtank2 มากกว่าหรือเท่ากับ 105 จะมีการเตือนว่า Over Flow ขึ้นมา

```
if abtank2.value >= 10 then low6.Active := true else low6.Active := false;
if abtank2.value >= 90 then high6.Active := true else high6.Active := false;
if abtank2.Value >= 105 then
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

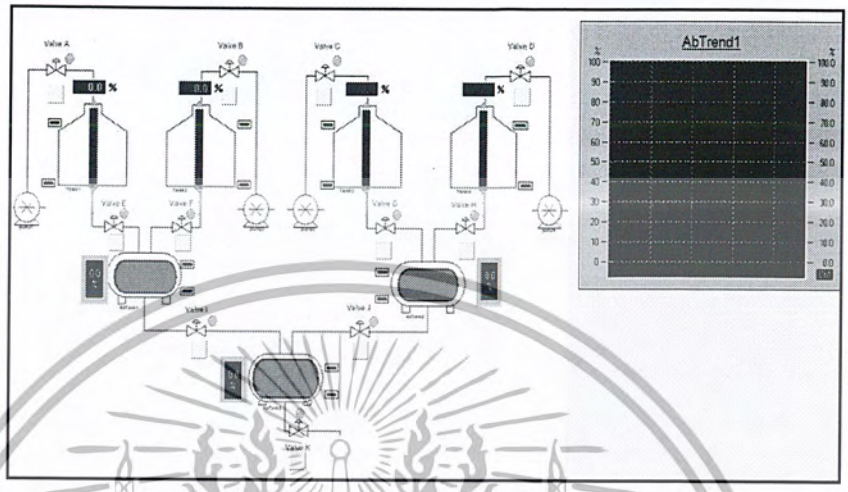
```
begin
Abtank2.Value := 105;
Button6.Visible := True;
end
else
begin
Button6.Visible := False;
end;
```

26. คำสั่งที่เขียนคือ สั่งให้ตัว Abtank3 แปรผันตรงกับตัวระดับ AbBar3 โดยที่ถ้าระดับของเหลวมีค่ามากกว่าหรือเท่ากับ 10 ตัวเซนเซอร์ ( low7) จะแสดงสถานะ (ไฟติด) แต่ถ้าระดับของเหลวมีค่าน้อยกว่า 10 ตัวเซนเซอร์ (low7) จะไม่แสดงสถานะ(ไฟดับ) , ถ้าระดับของเหลวมีค่ามากกว่าหรือเท่ากับ 90 ตัวเซนเซอร์ ( high7) จะแสดงสถานะ (ไฟติด) แต่ถ้าระดับของเหลวมีค่าน้อยกว่า 90 ตัวเซนเซอร์ ( high7) จะไม่แสดงสถานะ (ไฟดับ), ถ้าระดับของเหลวภายใน Abtank3 มากกว่าหรือเท่ากับ 105 จะมีการเตือนว่า Over Flow ขึ้นมา

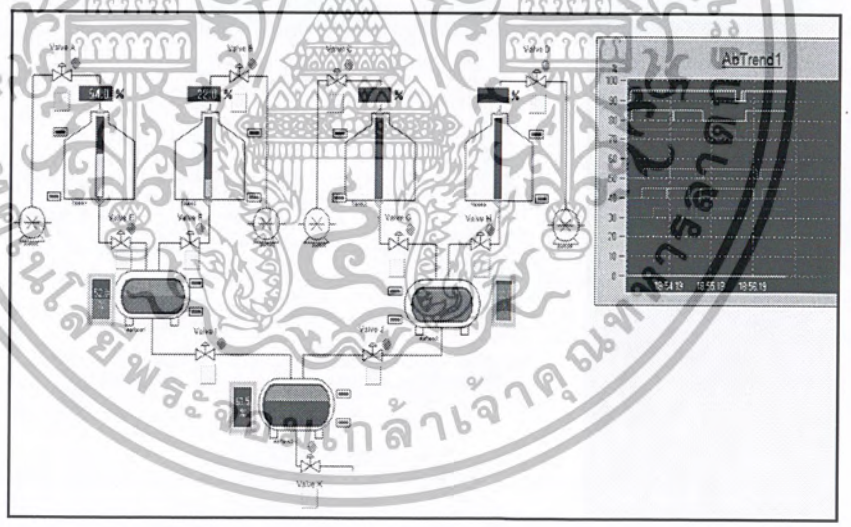
```
AbBar3.value := Abtank3.value;
    if Abtank3.value >= 10 then low7.Active := true else low7.Active := false;
    if Abtank3.value >= 90 then high7.Active := true else high7.Active := false;
    if Abtank3.Value >= 105 then
    begin
    Abtank3.Value := 105;
    Button7.Visible := True;
    end
    else
    begin
    Button7.Visible := False;
    end;
    end.
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.3.6 ผลจากรันโปรแกรมได้ออกมาเป็นแบบจำลองการวัดระดับได้ดังนี้

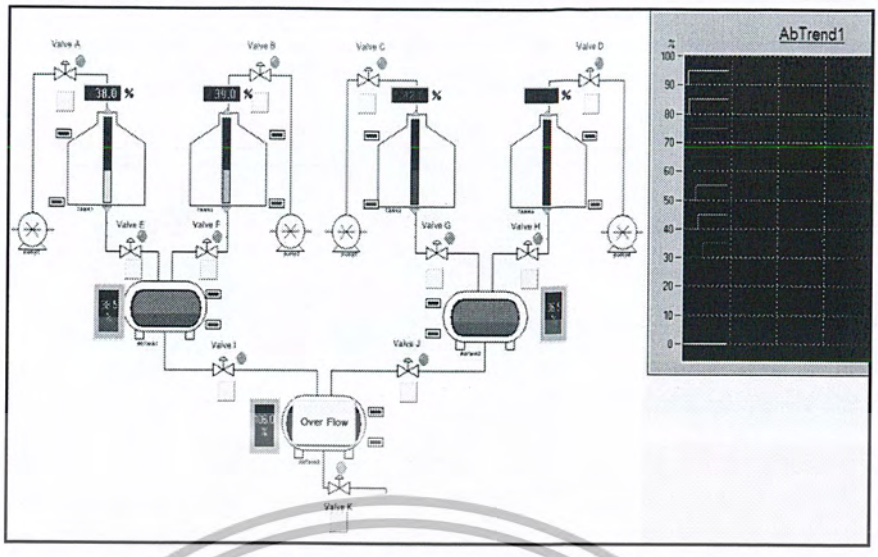



**รูปที่ 3.6** แบบจำลองในสภาวะปกติ

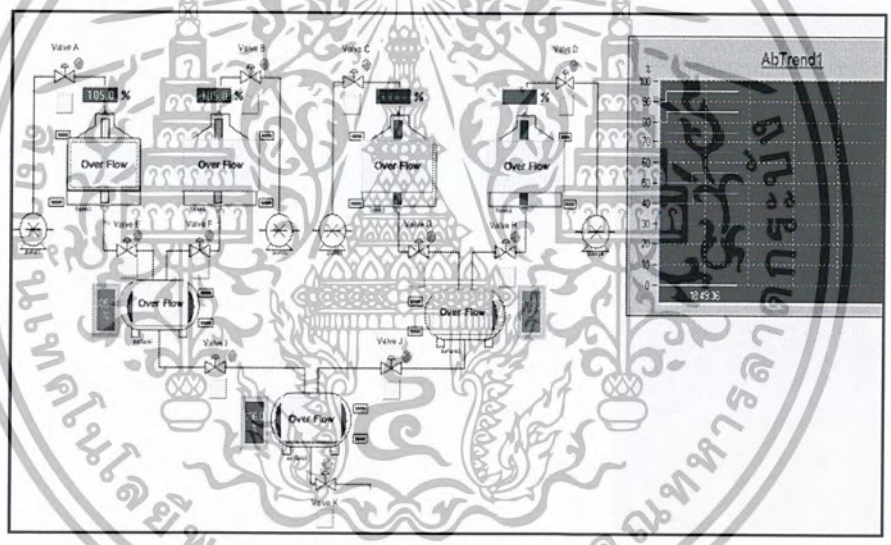



**รูปที่ 3.7** การทำงานของแบบจำลอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้




**รูปที่ 3.8** คำเตือน Over Flow



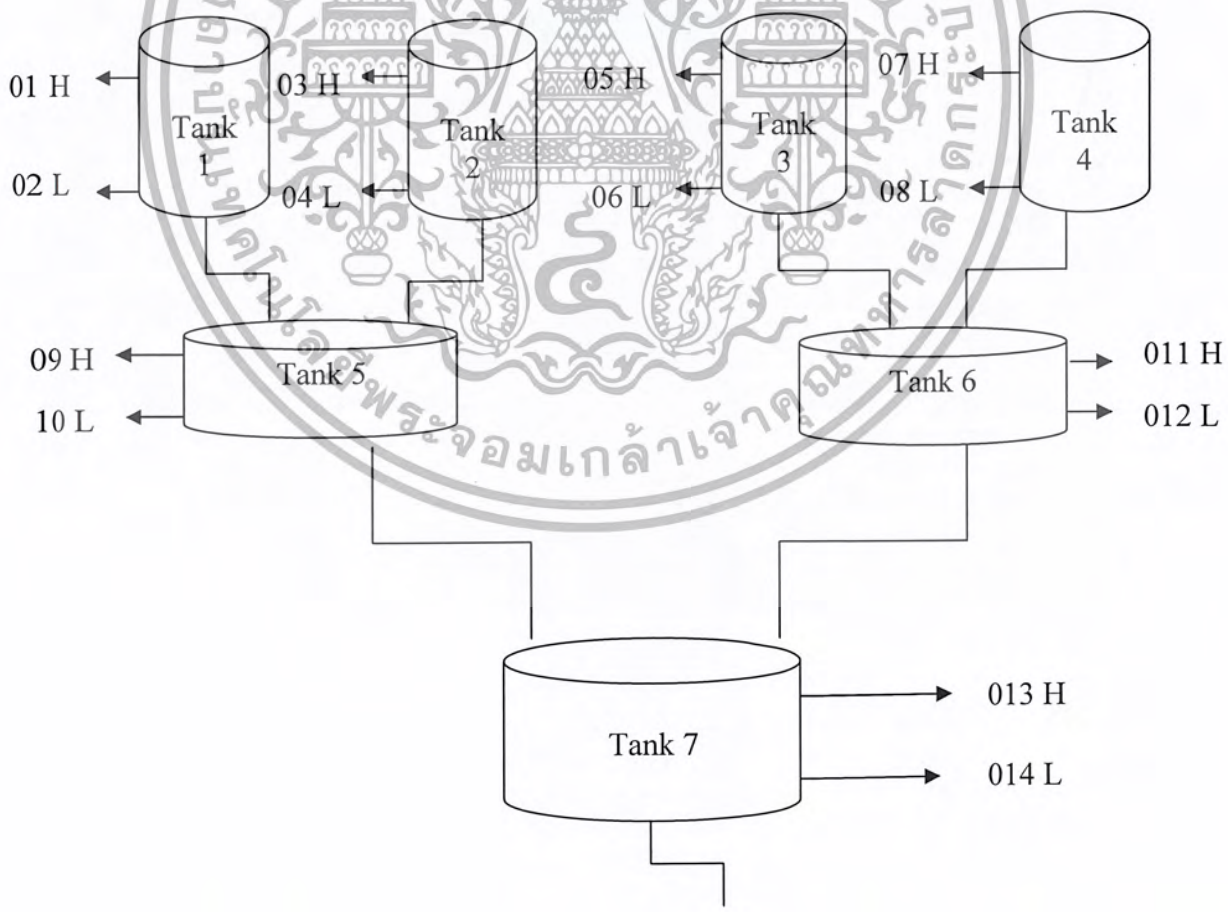

**รูปที่ 3.9** การเกิด Over Flow ทุก Tank

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**3.3.7 อธิบายหลักการของกระบวนการแบบจำลองที่ออกแบบการวัดระดับดังนี้**

จากการที่ทำการสร้างกระบวนการแบบจำลองการวัดระดับใน Visio2007 ในแบบจำลองนั้นจะมีอุปกรณ์ดังนี้ 1.) Valve 2.) Pump 3.) Tank 4.) Sensor จะมีวาล์วทั้งหมด 11 ตัว มีปั๊มทั้งหมด 4ตัว มี Tank ทั้งหมด 7 ถัง มีตัวเซนเซอร์ทั้งหมด 14 ตัว จากภาพเราจำลองการเปิดปิดวาล์วด้วยปุ่มเปิดปิด()เมื่อคลิกที่ปุ่มแล้วจะแสดงไฟสถานะ (AbLED) จะ on-of แสดงว่าวาล์วเปิด-ปิด ในแต่ละ Tank จะมีการจำลองติดตัวเซนเซอร์ (low) บอกระดับที่ระดับของเหลวสูง 10 % ไฟสถานะจะโชว์สีแดงขึ้นมาและตัวเซนเซอร์ (high) บอกระดับของเหลวสูง 90 %ไฟสถานะจะโชว์สีแดงขึ้นมา สำหรับใน Tank แต่ละ Tank นั้นจะมีการโชว์เปอร์เซ็นต์บอกระดับของของเหลวภายใน Tank ซึ่งถ้าของเหลวภายใน Tank แต่ละ Tank เกินค่าที่กำหนดไว้จะมีการเตือนขึ้นมาในที่นี้ตั้งค่าของเหลวไหลล้นไว้ที่ 105 ของแต่ละ Tank ถ้าระดับของเหลวเกินมันจะโชว์สถานะขึ้นมาว่า Over Flow

**3.3.8 ส่วนของการตั้งค่ากำหนด Input ของ Sensor และ Output ของValve โดยระบบเมื่อต่อเชื่อมกับ PLC แล้วมีระบบการทำงานดังนี้**



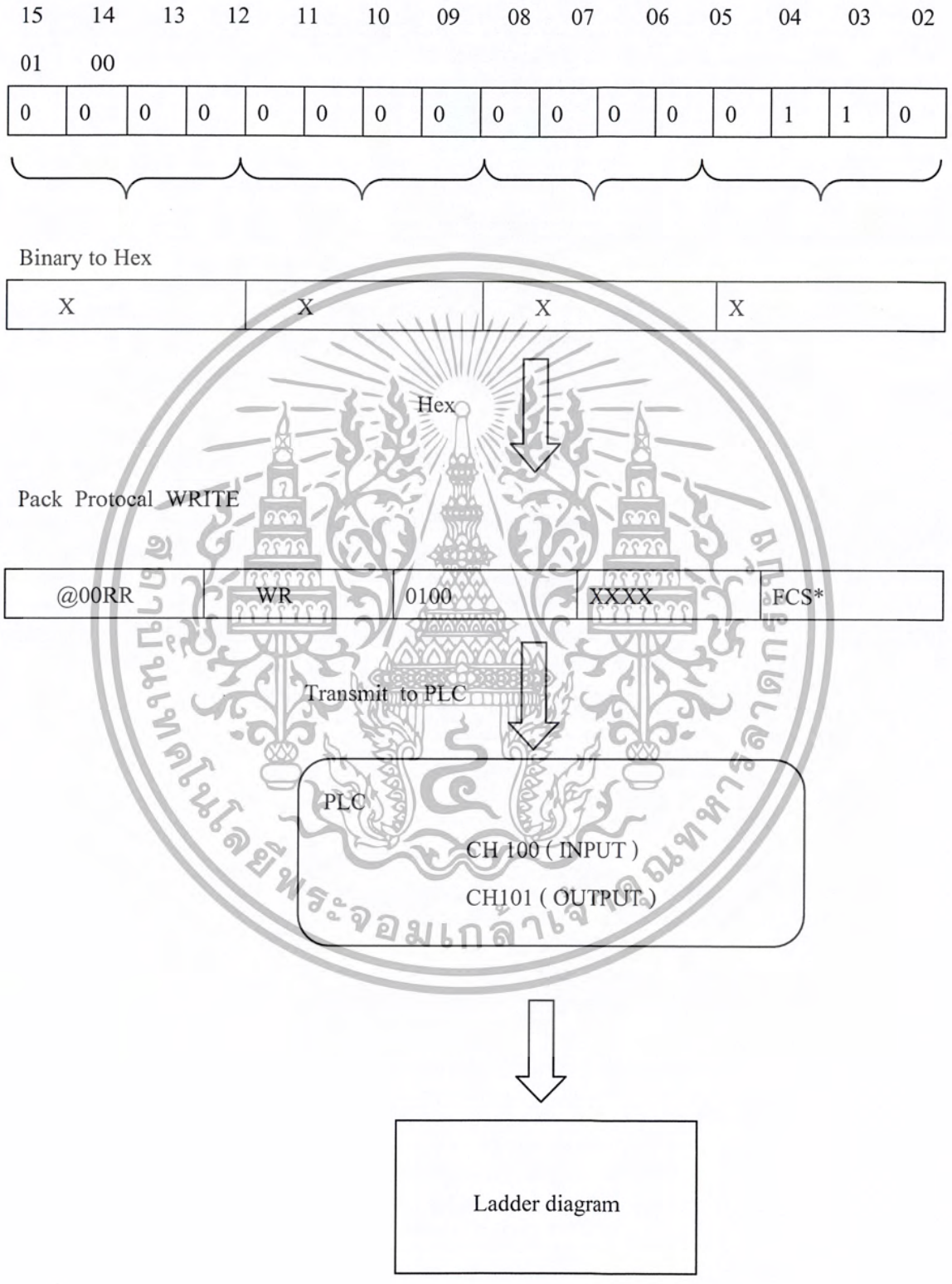

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.3.9 กำหนด Input

| | Bit | INPUT |
|---|---|---|
| CHANEL 100 | 01 | Sensor LA Low Level Tank 1 |
| | 02 | Sensor HA Low Level Tank 1 |
| | 03 | Sensor LB Low Level Tank 2 |
| | 04 | Sensor HB Low Level Tank 2 |
| | 05 | Sensor LC Low Level Tank 3 |
| | 06 | Sensor HC Low Level Tank 3 |
| | 07 | Sensor LD Low Level Tank 4 |
| | 08 | Sensor HD Low Level Tank 4 |
| | 09 | Sensor LE Low Level Tank 5 |
| | 010 | Sensor HE Low Level Tank 5 |
| | 011 | Sensor LF Low Level Tank 6 |
| | 012 | Sensor HF Low Level Tank 6 |
| | 013 | Sensor LG Low Level Tank 7 |
| | 014 | Sensor HG Low Level Tank 7 |

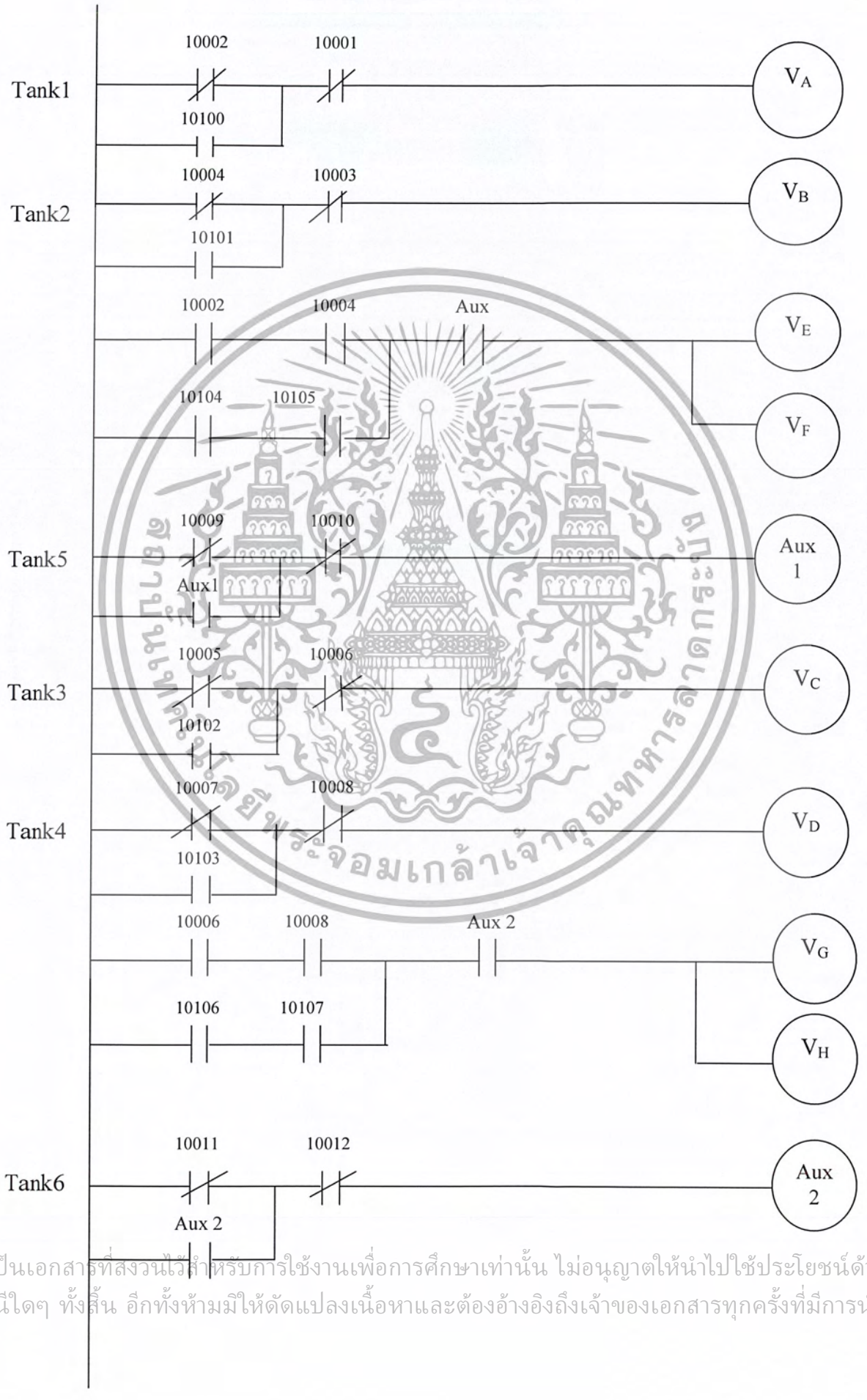
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ถ้าของไหลอยู่ที่ระดับ 01H Sensor ที่ 01H จะทำงาน ค่า Bit จะเปลี่ยนจาก 0 เป็น 1 เก็บค่าสถานะไว้

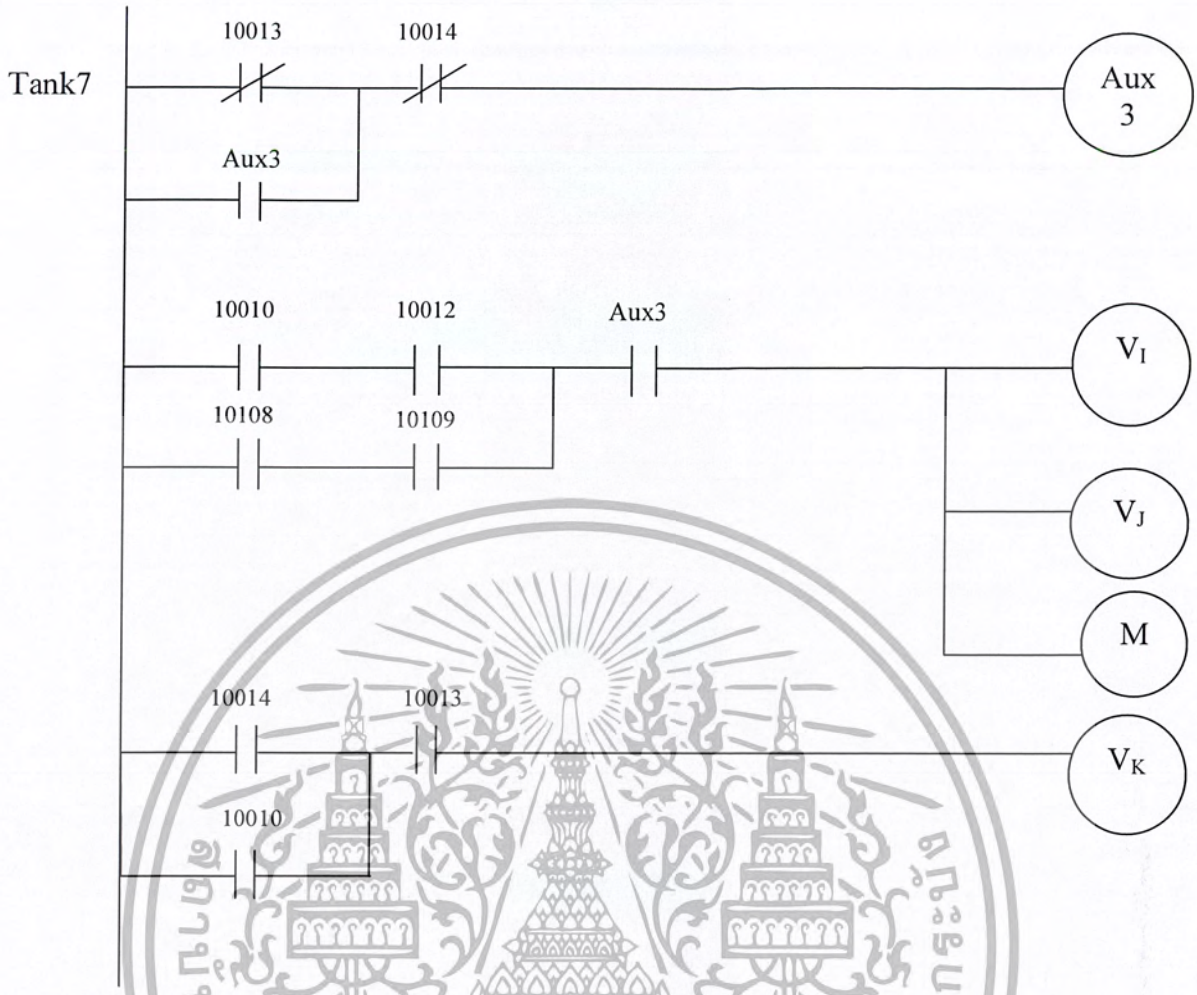



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.3.10 เขียน Ladder Diagram




เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้




ข้อมูลของ Sensor ที่ ON แล้ว PLC จะอ่านค่าแล้วไปควบคุมการเปิดปิดของวาล์ว เพื่อระบายของของ ไหลในกระบวนการ

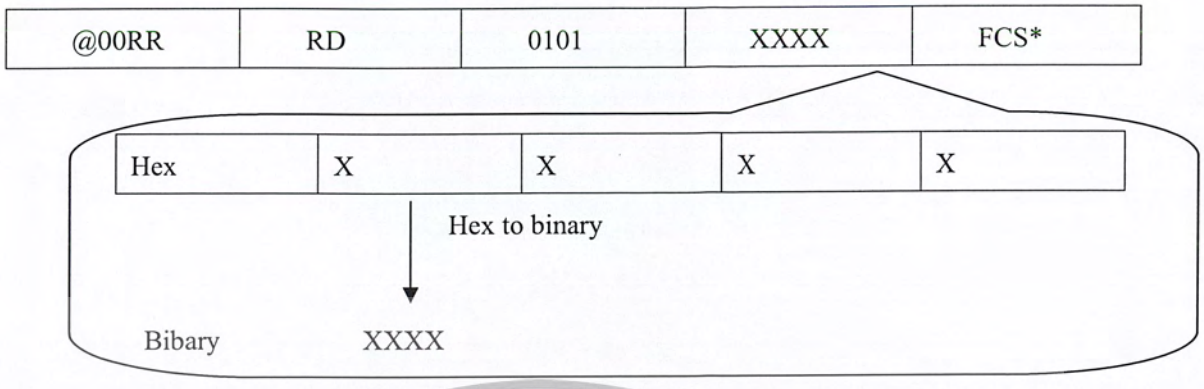
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.3.11 การกำหนด Output

| | Bit | OUTPUT |
|---|---|---|
| CHANEL 101 | 00 | Valve A Tank 1 |
| | 01 | Valve B Tank 2 |
| | 02 | Valve C Tank 3 |
| | 03 | Valve D Tank 4 |
| | 04 | Valve E Tank 1 + Tank 2 |
| | 05 | Valve F Tank 1 + Tank 2 |
| | 06 | Valve G Tank 3 + Tank 4 |
| | 07 | Valve H Tank 3 + Tank 4 |
| | 08 | Valve I Tank 5 + Tank 6 |
| | 09 | Valve J Tank 5 + Tank 6 |
| | 010 | Valve K Tank 7 |
| | 011 | M Motor |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.3.12 Pack Protocal Read




Delphi อ่านค่าจาก PLC มาเก็บไว้ในสถานะ
แปลงค่าจาก Hex to binary แล้วส่งค่ากลับมายัง PLC

PLC อ่านค่าแล้วของแต่ละ Bit ดังนั้น Va จึง
เปลี่ยนจากสถานะ 1 เป็น 0
(คือการปิด วาล์ว A หยุดจ่ายของไหล)

| 15 | 14 | 13 | 12 | 11 | 10 | 09 | 08 | 07 | 06 | 05 | 04 | 03 | 02 | 01 | 00 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | M | $V_k$ | $V_J$ | $V_I$ | $V_H$ | $V_G$ | $V_F$ | $V_E$ | $V_D$ | $V_C$ | $V_B$ | $V_A$ |
| | | | | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# บทที่ 4

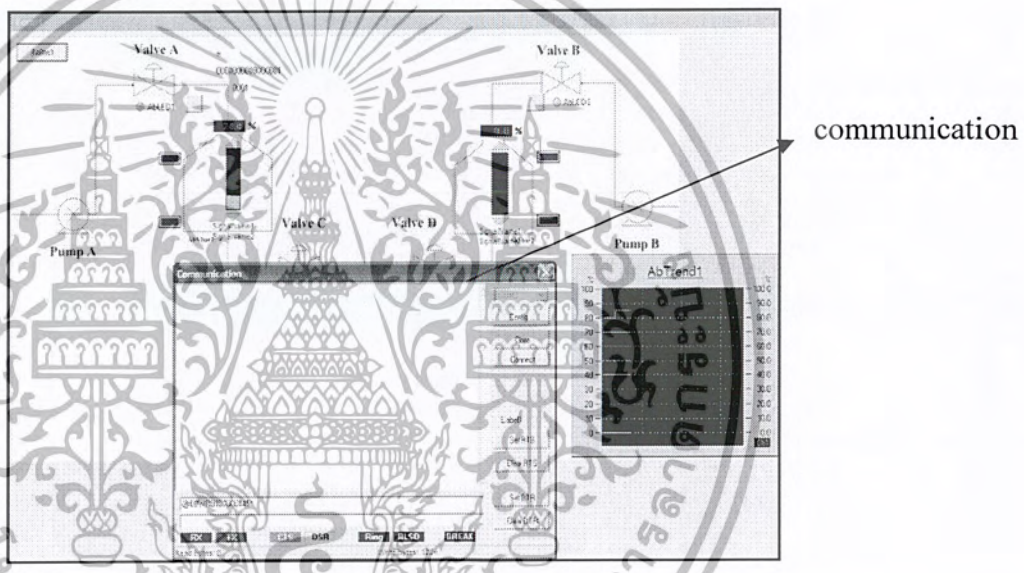
# ผลการทดลอง

## 4.1 การทำงานในส่วนคอมพิวเตอร์ของโปรแกรม Delphi5

จะมีการทำงานอยู่ 2 ส่วน คือ

1 ทำการเขียนโปรแกรมในโปรแกรม Delphi5 ให้ทำงานตามระบบที่ออกแบบ

2 ทำการติดต่อกับ PLC โดยใช้โปรแกรม Delphi5 โดยผ่าน Communication ซึ่งเป็นตัวเชื่อมต่อโปรแกรมเพื่อที่ส่งสถานะไปยัง PLC

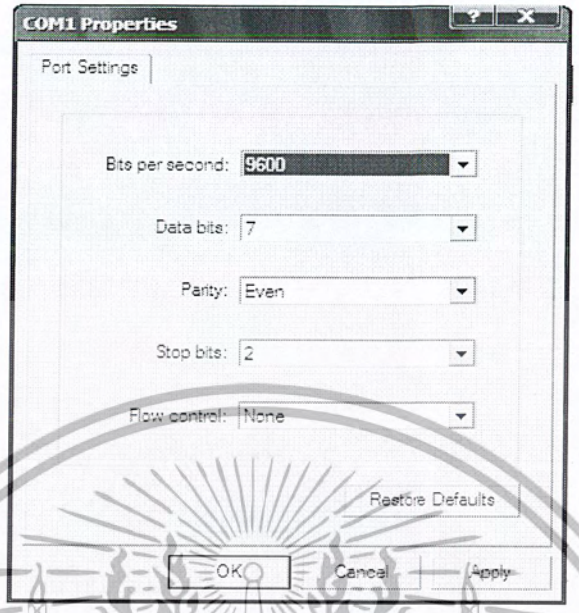



**รูปที่ 4.1** แสดงการแบบจำลองกับตัว Communication เพื่อเชื่อมกับ PLC

ไฟแสดงสถานการณ์เชื่อมต่อกระพริบ เนื่องจากว่าเครื่อง PLC กับคอมพิวเตอร์มีการเขียนคำสั่งจากโปรแกรม Delphi5 และ PLC สามารถที่จะอ่านคำสั่งจาก Delphi5 ได้

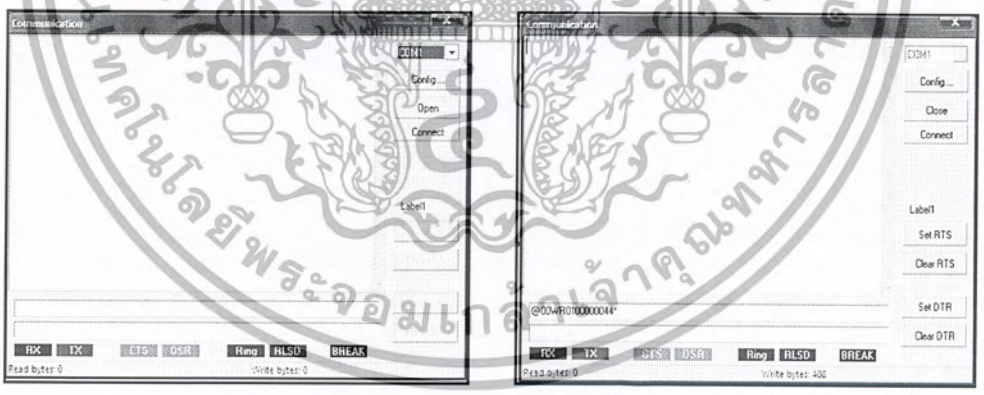
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 4.2 การตั้งค่า Com Port




**รูปที่ 4.2** แสดงการตั้งค่า Com Port

## 4.3 การส่งบิตไปให้ PLC อ่าน



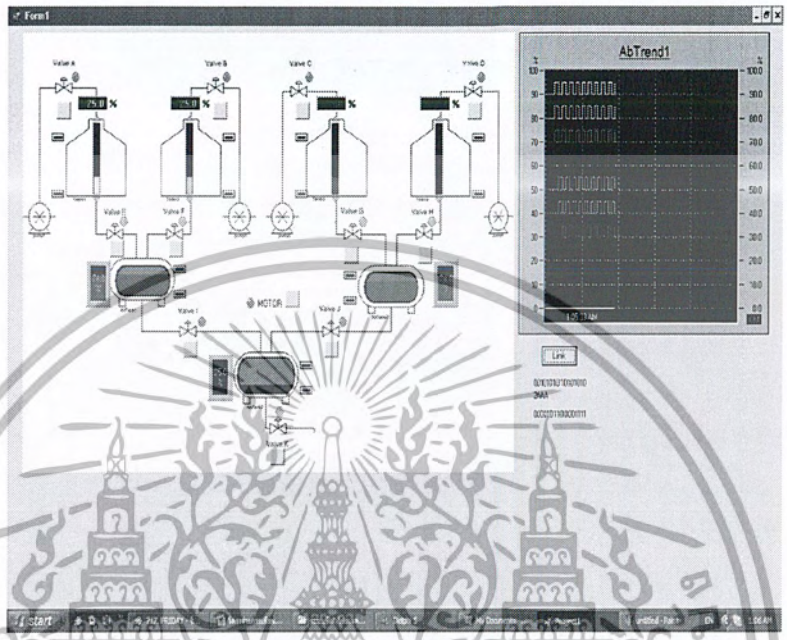

แสดงค่าเริ่มต้น ค่าที่ส่งไปให้ PLC

**รูปที่ 4.3** แสดงในส่วนที่ส่งบิตไปให้ PLC อ่าน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 4.4 การทำงานของแบบจำลองการวัดระดับ

แสดงการทำงานของโปรแกรม Delphi5 และ PLC เป็นไปตาม Ladder ที่ได้ออกแบบไว้ในขั้นตอนแรก




**รูปที่ 4.1.4** แสดงส่วนการทำงานของโปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# บทที่ 5
# สรุปผลการทดลอง

## 5.1 สรุปผลการวิจัย

ปริญญานิพนธ์ฉบับนี้ เป็นการศึกษาหลักการใช้งานและเขียนโปรแกรม เพื่อสั่งงานตัวควบคุมแบบโปรแกรมได้ โดยนำเอาตัวควบคุมแบบโปรแกรมได้ของยี่ห้อ Omron รุ่น C200HE มาใช้งานและใช้ซอฟแวร์ Syswin3.14 หรือตัว Programming Console รุ่น PRO01 สำหรับป้อนแลดเดอร์และแก้แลดเดอร์ โปรแกรมที่เขียนขึ้นจะถูกนำไปควบคุมกระบวนการไหลของของไหล วาล์วควบคุม เซนเซอร์ เช่น เมื่อของไหลมีระดับอยู่ที่ 10 ของถังน้ำตัวเซนเซอร์จะแสดงสถานะ Lowขึ้นมา

จากการทดลองได้ทำการเขียนโปรแกรมควบคุม การเปิดปิดของวาล์วควบคุม เซนเซอร์ การแสดงสถานะเมื่อของไหลเต็มถังจะมีข้อความเตือนขึ้นมาว่า Overflow ในปริญญานิพนธ์ฉบับนี้ทำการทดลองเมื่อของไหลอยู่ที่ระดับ 10 ตัวเซนเซอร์จะแสดงสถานะ Low ขึ้นมา และเมื่อถึงระดับที่ 90 เซนเซอร์อีกตัวจะแสดงสถานะ high ขึ้น และสุดท้ายที่ระดับ 110 จะมีข้อความขึ้นมาเตือนว่า Overflow เนื่องจากว่าข้อไหลเต็มหรือล้นออกจากถังแล้วนั้นเอง ส่วนการเปิดปิดของวาล์วควบคุมจะดูจากการทำงานของตัวเซนเซอร์เมื่อไรที่ต่ำกว่าสถานะ Low วาล์วจะถูกเปิดออกเพื่อปล่อยของไหลใส่เข้ามาในถัง

กระบวนการควบคุมอัตโนมัติสามารถสั่งงานผ่านทางหน้าจอคอมพิวเตอร์หรือเครื่อง PLC โดยใช้โปรแกรม Delphi และ Port RS232 สำหรับการเชื่อมต่อระหว่างตัวควบคุมแบบโปรแกรมได้กับเครื่องคอมพิวเตอร์

## 5.2 ข้อเสนอแนะ

ในอนาคตสามารถนำการควบคุมแบบดูแลตรวจตรา และการเข้าถึงข้อมูลมาประยุกต์ใช้กับการควบคุมกระบวนการต่างๆโดยมีเครื่องคอมพิวเตอร์เป็นส่วนแสดงผลและสั่งงาน และตัวควบคุมแบบโปรแกรมได้เป็นตัวควบคุมต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# บรรณานุกรม


[1] สัจจะ เจริญรุ่งระวีวร. *เริ่มต้นอย่างมืออาชีพด้วย Delphi 7 ฉบับสมบูรณ์*. นนทบุรี, บริษัท ไอดีซี อินโฟ ดิสทริบิวเตอร์ เซ็นเตอร์ จำกัด, 2547.

[2] ไสว พงศ์สวัสดิ์ และ ทวีพล ซื่อสัตย์, ปฏิบัติการวิศวกรรมการวัดคุม3. กรุงเทพมหานคร, คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง, 2547.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# ภาคผนวก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## การกำหนดอินพุทและเอาท์พุท

**Input**

| | Bit | INPUT |
|---|---|---|
| **CHANEL** 100 | 01 | Sensor LA Low Level Tank 1 |
| | 02 | Sensor HA Low Level Tank 1 |
| | 03 | Sensor LB Low Level Tank 2 |
| | 04 | Sensor HB Low Level Tank 2 |
| | 05 | Sensor LC Low Level Tank 3 |
| | 06 | Sensor HC Low Level Tank 3 |
| | 07 | Sensor LD Low Level Tank 4 |
| | 08 | Sensor HD Low Level Tank 4 |
| | 09 | Sensor LE Low Level Tank 5 |
| | 010 | Sensor HE Low Level Tank 5 |
| | 011 | Sensor LF Low Level Tank 6 |
| | 012 | Sensor HF Low Level Tank 6 |
| | 013 | Sensor LG Low Level Tank 7 |
| | 014 | Sensor HG Low Level Tank 7 |

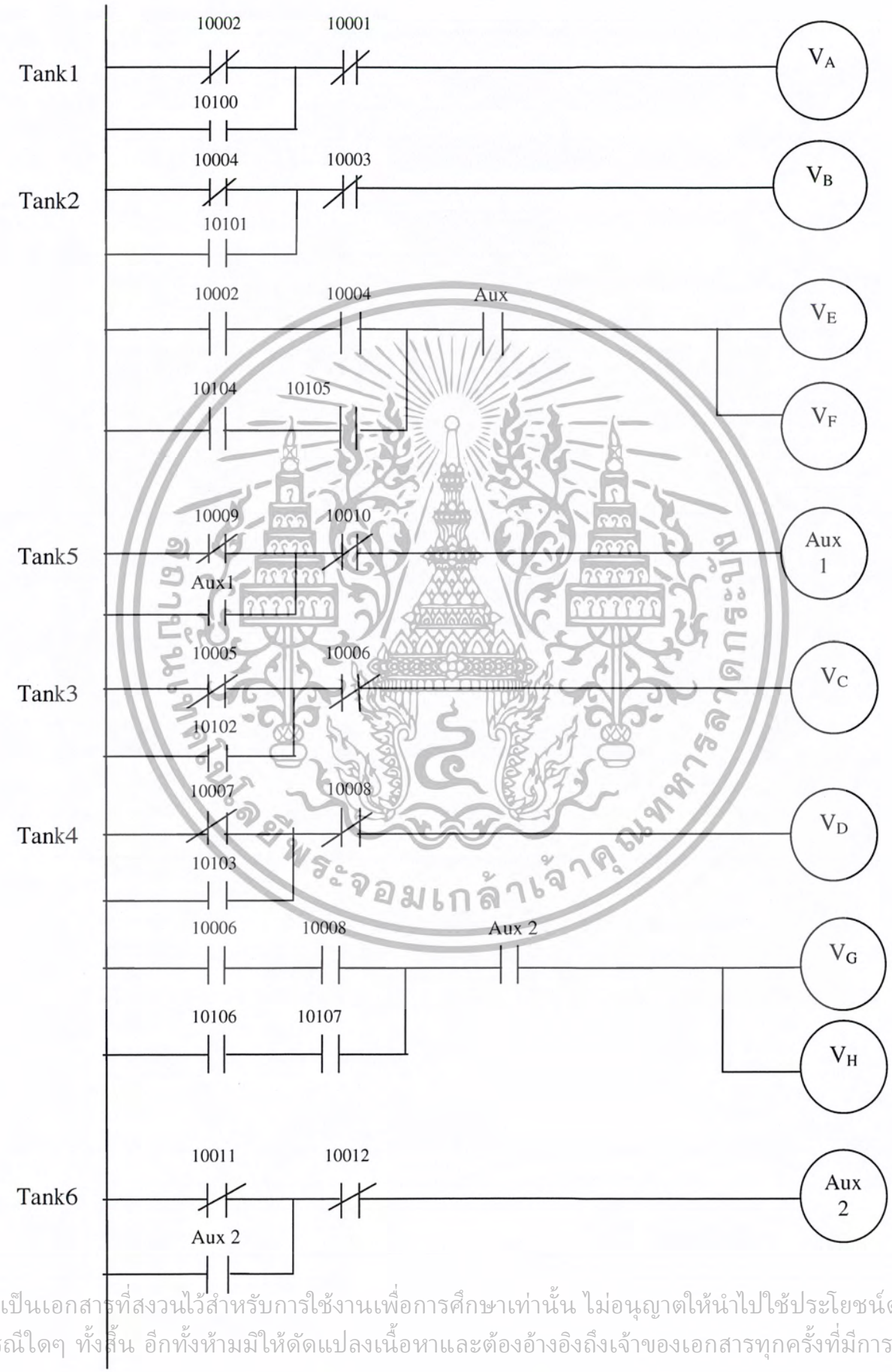
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**Output**

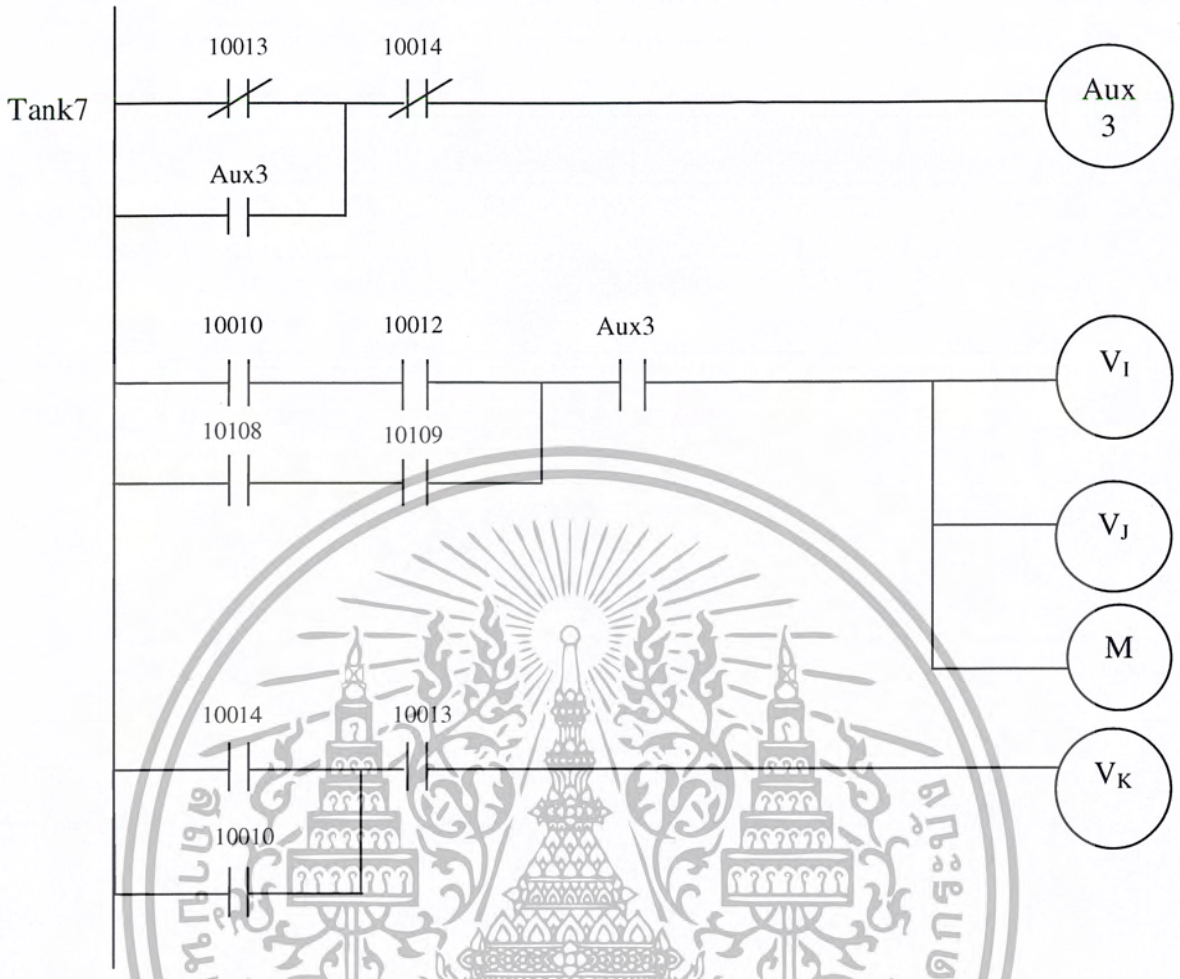
| | Bit | OUTPUT |
|---|---|---|
| **CHANEL** 101 | 00 | Valve A Tank 1 |
| | 01 | Valve B Tank 2 |
| | 02 | Valve C Tank 3 |
| | 03 | Valve D Tank 4 |
| | 04 | Valve E Tank 1 + Tank 2 |
| | 05 | Valve F Tank 1 + Tank 2 |
| | 06 | Valve G Tank 3 + Tank 4 |
| | 07 | Valve H Tank 3 + Tank 4 |
| | 08 | Valve I Tank 5 + Tank 6 |
| | 09 | Valve J Tank 5 + Tank 6 |
| | 010 | Valve K Tank 7 |
| | 011 | M Motor |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**LADDER DIAGRAM**




เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้




เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**Delphi 5**

```
unit Unit1;

interface

uses
  Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
  ExtCtrls, jpeg, Buttons, _BinGC, AbLED, AbTank, _AnlgGC, AbVBar, AbBar,
  VrControls, VrLeds, StdCtrls, AbTrend, _AbFlash, AbFlashT;
type
  TForm1 = class(TForm)
    Panel1: TPanel;
    Image1: TImage;
    SpeedButton1: TSpeedButton;
    SpeedButton2: TSpeedButton;
    SpeedButton3: TSpeedButton;
    SpeedButton4: TSpeedButton;
    SpeedButton5: TSpeedButton;
    SpeedButton6: TSpeedButton;
    SpeedButton7: TSpeedButton;
    SpeedButton8: TSpeedButton;
    SpeedButton9: TSpeedButton;
    SpeedButton10: TSpeedButton;
    SpeedButton11: TSpeedButton;
    AbLED1: TAbLED;
    AbLED2: TAbLED;
    AbLED3: TAbLED;
    AbLED4: TAbLED;
    AbLED5: TAbLED;
    AbLED6: TAbLED;
    AbLED7: TAbLED;
    AbLED8: TAbLED;
    AbLED9: TAbLED;
    AbLED10: TAbLED;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

AbLED11: TAbLED;

AbVBar1: TAbVBar;

AbVBar2: TAbVBar;

AbVBar3: TAbVBar;

AbVBar4: TAbVBar;

AbTank1: TAbTank;

AbTank2: TAbTank;

AbTank3: TAbTank;

AbBar1: TAbBar;

AbBar2: TAbBar;

AbBar3: TAbBar;

low1: TVrUserLed;

high1: TVrUserLed;

low2: TVrUserLed;

high2: TVrUserLed;

low3: TVrUserLed;

high3: TVrUserLed;

low4: TVrUserLed;

high4: TVrUserLed;

low5: TVrUserLed;

high5: TVrUserLed;

low6: TVrUserLed;

high6: TVrUserLed;

low7: TVrUserLed;

high7: TVrUserLed;

Timer1: TTimer;

Button1: TButton;

Button2: TButton;

Button3: TButton;

Button4: TButton;

Button5: TButton;

Button6: TButton;

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    Button7: TButton;
    AbFlashTimer1: TAbFlashTimer;
    AbTrend1: TAbTrend;
    procedure SpeedButton1Click(Sender: TObject);
    procedure SpeedButton2Click(Sender: TObject);
    procedure SpeedButton3Click(Sender: TObject);
    procedure SpeedButton4Click(Sender: TObject);
    procedure SpeedButton5Click(Sender: TObject);
    procedure SpeedButton6Click(Sender: TObject);
    procedure SpeedButton7Click(Sender: TObject);
    procedure SpeedButton8Click(Sender: TObject);
    procedure SpeedButton9Click(Sender: TObject);
    procedure SpeedButton10Click(Sender: TObject);
    procedure SpeedButton11Click(Sender: TObject);
    procedure Timer1Timer(Sender: TObject);

private
    { Private declarations }
  public
    { Public declarations }
  end;
var
  Form1: TForm1;
implementation

{$R *.DFM}
procedure TForm1.SpeedButton1Click(Sender: TObject);
begin
   if Abled1.Checked = True then Abled1.Checked := False else Abled1.Checked := True;
end;
procedure TForm1.SpeedButton2Click(Sender: TObject);
begin
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  if Abled2.Checked = True then Abled2.Checked := False else Abled2.Checked := True;
end;
procedure TForm1.SpeedButton3Click(Sender: TObject);
begin
  if Abled3.Checked = True then Abled3.Checked := False else Abled3.Checked := True;
end;
procedure TForm1.SpeedButton4Click(Sender: TObject);
begin
  if Abled4.Checked = True then Abled4.Checked := False else Abled4.Checked := True;
end;
procedure TForm1.SpeedButton5Click(Sender: TObject);
begin
  if Abled5.Checked = True then Abled5.Checked := False else Abled5.Checked := True;
end;
procedure TForm1.SpeedButton6Click(Sender: TObject);
begin
  if Abled6.Checked = True then Abled6.Checked := False else Abled6.Checked := True;
end;
procedure TForm1.SpeedButton7Click(Sender: TObject);
begin
  if Abled7.Checked = True then Abled7.Checked := False else Abled7.Checked := True;
end;
procedure TForm1.SpeedButton8Click(Sender: TObject);
begin
  if Abled8.Checked = True then Abled8.Checked := False else Abled8.Checked := True;
end;
procedure TForm1.SpeedButton9Click(Sender: TObject);
begin
  if Abled9.Checked = True then Abled9.Checked := False else Abled9.Checked := True;
end;
procedure TForm1.SpeedButton10Click(Sender: TObject);
begin
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  if Abled10.Checked = True then Abled10.Checked := False else Abled10.Checked := True;
end;
procedure TForm1.SpeedButton11Click(Sender: TObject);
begin
  if Abled11.Checked = True then Abled11.Checked := False else Abled11.Checked := True;
end;
procedure TForm1.Timer1Timer(Sender: TObject);
begin
  if Abled1.Checked = True then
  begin
  AbVBar1.value := AbVBar1.value+1;
  end;
if Abled2.Checked = True then
  begin
  AbVBar2.value := AbVBar2.value+1;
  end;
if Abled3.Checked = True then
  begin
  AbVBar3.value := AbVBar3.value+1;
  end;
if Abled4.Checked = True then

 begin

    AbVBar4.value := AbVBar4.value+1;
  end;
    if Abled5.Checked = True then
  begin
    AbVBar1.value := AbVBar1.value-1;
    if AbVBar1.value <=0 then AbVBar1.value := 0;
  end;
  if (Abled5.Checked = True) and (AbVbar1.Value > 0) then
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  begin
    Abtank1.value := Abtank1.value+0.5;
  end;
  if Abled6.Checked = True then
  begin
    AbVBar2.value := AbVBar2.value-1;
    if AbVBar2.value <=0 then AbVBar2.value := 0;
  end;
    if (Abled6.Checked = True) and (AbVbar2.Value > 0) then
  begin
    Abtank1.value := Abtank1.value+0.5;
  end;
if Abled7.Checked = True then
  begin
    AbVBar3.value := AbVBar3.value-1;
    if AbVBar3.value <=0 then AbVBar3.value := 0;
  end;
    if (Abled7.Checked = True) and (AbVbar3.Value > 0) then
  begin
    Abtank2.value := Abtank2.value+0.5;
  end;
 if Abled8.Checked = True then
  begin
    AbVBar4.value := AbVBar4.value-1;
    if AbVBar4.value <=0 then AbVBar4.value := 0;
  end;
    if (Abled8.Checked = True) and (AbVbar4.Value > 0) then
  begin
    Abtank2.value := Abtank2.value+0.5;
  end;
  if Abled9.Checked = True then
  begin
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  Abtank1.value := Abtank1.value-1;
  if Abtank1.value <=0 then Abtank1.value := 0;
end;
  if (Abled9.Checked = True) and (Abtank1.Value > 0) then
begin
  Abtank3.value := Abtank3.value+0.5;
end;
if Abled10.Checked = True then
begin
  Abtank2.value := Abtank2.value-1;
  if Abtank2.value <=0 then Abtank2.value := 0;
end;
  if (Abled10.Checked = True) and (Abtank2.Value > 0) then
begin
  Abtank3.value := Abtank3.value+0.5;
end;
if Abled11.Checked = True then
begin
  AbTank3.value := AbTank3.value -1;
  if AbTank3.value <=0 then AbTank3.value := 0;
end;
  AbBar1.value := Abtank1.value;
  if AbVBar1.value >= 10 then low1.Active := true else low1.Active := false;
  if AbVBar1.value >= 90 then high1.Active := true else high1.Active := false;
  if AbVbar1.Value >= 105 then
 begin
    AbVbar1.Value := 105;
    Button1.Visible := True;
  end
  else
  begin
   Button1.Visible := False;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
end;
AbBar2.value := Abtank2.value;
if AbVBar2.value >= 10 then low2.Active := true else low2.Active := false;
if AbVBar2.value >= 90 then high2.Active := true else high2.Active := false;
if AbVbar2.Value >= 105 then
begin
  AbVbar2.Value := 105;
  Button2.Visible := True;
end
else
begin
 Button2.Visible := False;
end;
if AbVBar3.value >= 10 then low3.Active := true else low3.Active := false;
if AbVBar3.value >= 90 then high3.Active := true else high3.Active := false;
if AbVbar3.Value >= 105 then
begin
  AbVbar3.Value := 105;
  Button3.Visible := True;
end
else
begin
Button3.Visible := False;
end;
if AbVBar4.value >= 10 then low4.Active := true else low4.Active := false;
if AbVBar4.value >= 90 then high4.Active := true else high4.Active := false;
if AbVbar4.Value >= 105 then
begin
  AbVbar4.Value := 105;
  Button4.Visible := True;
end
else
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
begin
 Button4.Visible := False;
end;
if abtank1.value >= 10 then low5.Active := true else low5.Active := false;
if abtank1.value >= 90 then high5.Active := true else high5.Active := false;
if abtank1.Value >= 105 then
begin
Abtank1.Value := 105;
Button5.Visible := True;
end
else
begin
 Button5.Visible := False;
end;
if abtank2.value >= 10 then low6.Active := true else low6.Active := false;
if abtank2.value >= 90 then high6.Active := true else high6.Active := false;
if abtank2.Value >= 105 then
begin
  Abtank2.Value := 105;
  Button6.Visible := True;
end
else
begin
 Button6.Visible := False;
end;
AbBar3.value := Abtank3.value;
if Abtank3.value >= 10 then low7.Active := true else low7.Active := false;
if Abtank3.value >= 90 then high7.Active := true else high7.Active := false;
if Abtank3.Value >= 105 then

begin
  Abtank3.Value := 105;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  Button7.Visible := True;
 end
 else
 begin
  Button7.Visible := False;
 end;
 if AbLed1.Checked = true then AbTrend1.Channel1.Value := 95 else
AbTrend1.Channel1.Value := 90;
 if AbLed2.Checked = true then AbTrend1.Channel2.Value := 85 else
AbTrend1.Channel2.Value := 80;
 if AbLed3.Checked = true then AbTrend1.Channel3.Value := 75 else
AbTrend1.Channel3.Value := 70;
 if AbLed4.Checked = true then AbTrend1.Channel4.Value := 65 else
AbTrend1.Channel4.Value := 60;
 if AbLed5.Checked = true then AbTrend1.Channel5.Value := 55 else
AbTrend1.Channel5.Value := 50;
 if AbLed6.Checked = true then AbTrend1.Channel5.Value := 55 else
AbTrend1.Channel5.Value := 50;
 if AbLed7.Checked = true then AbTrend1.Channel6.Value := 45 else
AbTrend1.Channel6.Value := 40;
 if AbLed8.Checked = true then AbTrend1.Channel6.Value := 45 else
AbTrend1.Channel6.Value := 40;
 if AbLed9.Checked = true then AbTrend1.Channel7.Value := 35 else
AbTrend1.Channel7.Value := 30;
end
end.
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้