สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

# ชุดทดลอง PLC
# PLC SIMULATOR


โดย

นาย อนุชิต อภิชาติธนากูล

นางสาว วราภรณ์ ทรัพย์มาก

อาจารย์ที่ปรึกษา

รศ. สุเธียร เกียรติสุนทร

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมระบบควบคุม

สถาบันเทคโนโลยีพระจอมเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2542

เลขหมู่.....
เลขทะเบียน 36863
วัน, เดือน, ปี 29 ส.ค. 2543

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่อนุญาตให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์ ปีการศึกษา 2542

สาขาวิชาระบบควบคุม

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง ชุดทดลอง PLC


ผู้จัดทำ

1. นายอนุชิต อภิชาติธนากูล รหัส 39014647
2. นางสาววราภรณ์ ทรัพย์มาก รหัส 39014464


.....อาจารย์ที่ปรึกษา

(รศ. สุเธียร เกียรติสุนทร)


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# ชุดทดลอง PLC

นายอนุชิต อภิชาติธนากูล
นางสาววราภรณ์ ทรัพย์มาก
รศ.สุเธียร เกียรติสุนทร อาจารย์ที่ปรึกษา
ปีการศึกษา 2542

บทคัดย่อ

ในปริญญานิพนธ์ฉบับนี้เรียบเรียงขึ้นจากผลงานที่ได้ทำการพัฒนาโปรแกรมจำลองการทำงานของเครื่อง PLC บนเครื่อง IBM PC โดยใช้ไมโครซอฟต์ซีเป็นตัวคอมไพเลอร์ ให้สามารถแสดงผลการทำงานเป็นรูปภาพที่สามารถเคลื่อนไหวได้ในกราฟฟิกส์โหมดบนหน้าจอคอมพิวเตอร์ โดยที่ตัวโปรแกรม PLC ยังสามารถทำงานได้เหมือนเดิม โดยมีการจำลองระบบที่ใช้ในอุตสาหกรรมมา 2 ระบบคือ การจำลองการทำงานของการผสมของเหลวภายในถังอัตโนมัติโดยใช้ PLC ควบคุม กับการจำลองการทำงานของการเคลื่อนย้ายวัตถุอัตโนมัติโดยใช้ PLC ควบคุม

Abstract

This thesis wrote from program PLC simulator on IBM PC that compile by Microsoft C Compiler that develop to show the result of processing in form of animation picture in graphic mode on PC monitor and the requirement is not change any function of program, this new program have 2 simulation process, simulation of automatic mix liquid in tank process by PLC and simulation of automatic moving object by PLC.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# สารบัญ



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# สารบัญรูปภาพ



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# สารบัญตาราง

หน้า



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# บทที่ 1
# บทนำ

ในปัจจุบันระบบการควบคุมต่างๆ ต้องการความถูกต้องในการตรวจสอบมากขึ้น เพื่อความเที่ยงตรง แม่นยำ และให้ผลิตภัณฑ์เป็นไปตามที่ต้องการ ในขณะที่อุปกรณ์เครื่องมือทางด้านคอมพิวเตอร์มีราคาต่ำลง คุ้มกับการลงทุนนำมาใช้งาน แต่ค่าจ้างแรงงานรวมถึงบุคลากรผู้ชำนาญการกลับหายาก และยังต้องลงทุนที่สูงขึ้น ด้วยเหตุผลต่างๆ เหล่านี้ จึงเป็นการเหมาะสมอย่างยิ่ง หากสามารถลดภาระของผู้ควบคุมระบบ ให้บุคลากรเหล่านี้ สามารถใช้เวลาที่มีเพิ่มขึ้นไปพัฒนาการทำงานในส่วนอื่นๆ ต่อไป สิ่งเหล่านี้ จึงเป็นที่มาของแนวคิดในการออกแบบโปรแกรมแสดงผลการทำงานโดยผ่าน PLC

โปรแกรมในโครงงานนี้จะช่วยให้ผู้ทำการควบคุมสามารถติดตามผล และเรียนรู้การทำงานของกระบวนการได้โดยง่าย เนื่องจากเป็นการแสดงผลแบบรูปภาพ ซึ่งเป็นการสื่อความหมายอย่างชัดเจน และไม่จำเป็นต้องไปสังเกตุผลในสถานที่จริง จึงทำให้การควบคุมทำได้สะดวก ผลการควบคุมจะถูกตรวจสอบผลผ่านจอแสดงผลโดยข้อมูลต่างๆ จะถูกส่งผ่านมาจากโปรแกรมที่ใช้ในการควบคุม PLC และตัว PLC นี้ ก็จะไปควบคุมการทำงานของกระบวนการจริงอีกทีหนึ่ง

ลักษณะสัญญาณของ PLC ที่ส่งผ่านมายังคอมพิวเตอร์ อาจจะเป็นการขนส่งแบบขนานหรือแบบอนุกรมก็ได้ขึ้นอยู่กับลักษณะขอของ PLC แต่ละเครื่อง สำหรับสัญญาณที่ส่งเข้ามายังหน่วยควบคุมซึ่งเป็นเครื่องคอมพิวเตอร์ คอมพิวเตอร์นี้อาจยังไม่สามารถรับรู้ข้อมูลได้โดยตรง ดังนั้นเราจึงจำเป็นต้องสร้างโปรแกรมที่ใช้ในการรับรู้ข้อมูล และจัดลำดับการรับรู้ข้อมูล และจัดลำดับข้อมูลแล้วค่อยทำการส่งต่อให้กับ โปรแกรมแสดงผลการทำงานอีกทีหนึ่ง

โปรแกรมที่ใช้รับข้อมูลจาก PLC นี้เราเรียกว่าโปรแกรมไดร์เวอร์(Driver Program) ส่วนโปรแกรมที่แสดงผลการทำงานของกระบวนการเราจะเรียกว่าโปรแกรมแสดงกราฟฟิกส์(Graphics Display Program) จะสังเกตได้ว่าเราแยกโปรแกรมทั้งสองส่วนออกจากกันทั้งๆที่มันต้องมีการทำงานร่วมกัน ทั้งนี้ก็เพราะ PLC ในแต่ละแบบ จะมีคุณสมบัติแตกต่างกันออกไป ดังนั้นเราจึงจำเป็นต้องสร้างโปรแกรมไดร์เวอร์เฉพาะ PLC ในแต่ละรุ่น เพื่อให้การทำงานมีความยืดหยุ่นมากขึ้น สำหรับส่วนแสดงผลกราฟฟิกส์นั้นเราสามารถใช้ร่วมกันได้ตลอด

ในส่วนของโปรแกรมแสดงผลการทำงานของกระบวนการเดิมจะมีลักษณะเป็นเพียงบล็อกไดอะแกรมใน TEXT MODE ธรรมดา มีการแสดงผลในแบบ ON / OFF เท่านั้น แต่โปรแกรมที่ออกแบบในโครงงานนี้จะแสดงผลใน GRAPHICS MODE และยังเป็น Editor Graphics ในตัวอีกด้วย กล่าวคือเมื่อเรานำโปรแกรมนี้ไปใช้ร่วมกับ Process ต่างๆนั้น เราอาจยังไม่ทราบถึงรูปแบบ และ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ลำดับการทำงานของกระบวนการนั้นว่าเป็นอย่างไรบ้าง หรือในบางกรณี อาจจำเป็นต้องมีการปรับเปลี่ยน ขั้นตอนการทำงานของกระบวนการ ดังนั้น เพื่อความยืดหยุ่นในการทำงานเราจึงได้ออกแบบโปรแกรมเหล่านี้ สามารถแก้ไขรูปภาพที่แสดงขั้นตองการทำงาน ในกระบวนการได้สะดวกในลักษณะของ Editor Graphicsนั่นเอง นอกจากนั้น ยังทำให้ผู้ควบคุมสามารถปฏิบัติงานได้อย่างมีประสิทธิภาพมากขึ้น เพราะโปรแกรมนี้จะแสดงผลเป็นรูปภาพ ซึ่งเป็นการสื่อความหมายได้อย่างชัดเจน มิใช่เป็นแค่เพียงสัญลักษณ์ กับรหัสประจำสั้นๆ

สำหรับการจัดเก็บข้อมูลรูปภาพ จะมีการจัดเก็บในระบบ TEXT FILE ซึ่งจะช่วยประหยัดเนื้อที่ในการเก็บข้อมูลได้อย่างมีประสิทธิภาพ เราสามารถแสดงลักษณะการทำงานของโปรแกรมดังกล่าว ต่อร่วมกับกระบวนการจริงได้ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# ประวัติของ PLC

ในอดีตการควบคุมแบบซีเควนซ์หรือแบบหรือแบบลำดับนั้นจะใช้รีเลย์เป็นหลัก และมักประสบปัญหาเมื่อมีการเปลี่ยนแปลงหรือปรับปรุงวงจร เพราะความยุ่งยากในการเดินสายไฟ และขนาดของพื้นที่ ที่ใช้สำหรับติดตั้งอุปกรณ์ควบคุมต่างๆ จำกัด เมื่อมีความก้าวหนน้าทางด้านอิเล็กทรอนิกส์เพิ่มมากขึ้น จึงมีการใช้วงจรอิเล็กทรอนิกส์เข้ามาควบคุมแทนระบบเก่า ซึ่งได้เปรียบกว่าในเรื่องของขนาด ราคา ความเชื่อถือได้ ความเร็วในการทำงาน และความปลอดภัยจากไฟฟ้าที่ใช้ในการควบคุม อย่างไรก็ตาม การนำวงจรอิเล็กทรอนิกส์เข้ามาใช้แทนระบบเก่า ซึ่งเป็นวงจรรีเลย์ก็ยังไม่สามารถแก้ความยุ่งยากในการปรับปรุง และพัฒนาวงจรควบคุมได้ จนกระทั่งไม่กี่ปีนานมานี้ได้มีการพัฒนาระบบควบคุมที่ไม่ผูกติดกับอุปกรณ์ หรือฮาร์ดแวร์ขึ้นสำหรับใช้ในระดับอุตสาหกรรม โดยทำงานทีละคำสั่ง ที่ถูกกำหนดจากผู้ใช้ในลักษณะของโปรแกรม หรือชุดคำสั่งที่ทำให้ระบบควบคุมอัตโนมัติ มีความยืดหยุ่นต่อการใช้งานสูง และสามารถทนต่อสภาวะแวดล้อมต่างๆ ที่เกิดขึ้นในกระบวนการผลิตได้ การปรับปรุงแก้ไขการทำงานก็สามารถทำได้โดยง่าย เพียงแต่แก้ไข หรือเปลี่ยนแปลงโปรแกรม และที่สำคัญใช้ไมโครโพรเซสเซอร์ (Microprocessor) เป็นหน่วยประมวลผลทำให้อุปกรณ์มีขนาดเล็ก แต่ขีดความสามารถและความเร็วในการทำงานสูงกว่าเดิม

เนื่องจากอุปกรณ์ควบคุมที่พัฒนาขึ้นมานี้ ถูกกำหนดการทำงานมาจากโปรแกรมที่ผู้ใช้ได้เขียนขึ้น จึงเรียกเครื่องควบคุมนี้ว่า โปรแกรมเมเบิลคอนโทรลเลอร์ หรือ PC (Programmable Controller) สามารถทำการวิเคราะห์ได้ทั้งสัญญาณแบบดิจิตอลและอนาลอก นอกจากนี้ยังมีอุปกรณ์ควบคุม ที่ถูกออกแบบให้มีขนาดเล็กลงไปอีก เพื่อให้เหมาะสมกับงานที่มีการควบคุมไม่ซับซ้อนมากนัก โดยให้วิเคราะห์ได้แต่สัญญาณดิจิตอลเพียงอย่างเดียว เรียกว่าโปรแกรมเมเบิลลอจิก คอนโทรเลอร์ (Programmable Logic Control) ซึ่งจัดเป็น PC ขนาดเล็กนั่นเอง แต่ก็สามารถใช้ได้ดีกับงานควบคุมทั่วไป และใช้ไมโครโพรเซสเซอร์เป็นหน่วยประมวลผลเหมือนกันทำให้ทั้ง PC และ PLC สามารถติดต่อกับคอมพิวเตอร์ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# ข้อดีของ PLC เมื่อเปรียบเทียบกับระบบควบคุมเดิม

- ติดตั้ง และควบคุมการทำงานได้ง่าย
- มีขนาดเล็ก ใช้พื้นที่ในการติดตั้งน้อย
- มีความยืดหยุ่น และคล่องตัวต่อการใช้งานในอุตสาหกรรมระดับสูง
- การทำงานอาศัยโปรแกรมที่ป้อนจากผู้ใช้ จึงสามารถปรับปรุงและแก้ไขได้ง่าย
- หน่วยความจำที่ใช้สามารถใช้ได้หลายแบบ อาจเป็น RAM,ROM,PROM หรือ EPROM
- สามารถควบคุมอุปกรณ์ที่อยู่ไกล ๆ ได้ด้วยการควบคุมแบบรีโมท
- มี input / output relay, auxiliary relay และ timer / counter ให้เรียกใช้เป็นจำนวนมาก และหน้าสัมผัสของรีเลย์แต่ละตัวสามารถเรียกใช้งานได้ไม่จำกัด
- ความเร็วในการทำงานสูง
- ทนต่อสภาพการใช้งานในหลายลักษณะ การดูแลรักษาทำได้ง่าย
- ไม่มีปัญหาเกี่ยวกับความสกปรกของหน้าสัมผัส
- ความน่าเชื่อถือสูง เพราะเป็นคอมพิวเตอร์ชนิดหนึ่งที่ถูกออกแบบให้นำมาใช้กับงานเฉพาะด้าน
- เหมาะสำหรับอุตสาหกรรมที่อาจมีการขยาย หรือเพิ่มขบวนการผลิตเพราะสามารถติดตั้งและเพิ่มจุดต่ออินพุท / เอาท์พุทได้อีกจำนวนมาก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# โครงสร้างและการทำงานของ PLC

ถ้าพิจารณาโครงสร้างโดยทั่วไปแล้ว อาจกล่าวได้ว่า PLC นั้น ประกอบด้วยโครงสร้างที่สำคัญอยู่ 4 ส่วนคือ

## 1) หน่วยประมวลผล (processor unit or CPU unit)

การประมวลผลของ PLC จะประมวลตามโปรแกรมที่ผู้ใช้ป้อน โดยสถานะภาพการทำงานของอุปกรณ์ ขึ้นอยู่กับข้อมูลที่รับเข้ามา และผลที่ได้รับจากการประมวลผลข้อมูลที่เข้ามานั้นจะแสดงทางด้านเอาท์พุต การตรวจรับข้อมูลอินพุตจะเริ่มตรวจรับ ตั้งแต่อินพุทแรก จนถึงอินพุทสุดท้าย แล้วจึงกลับไปตรวจรับอินพุทแรกอีก กล่าวคือเป็นการทำงานแบบวนรอบ (Loop) หรือแบบทำซ้ำ (Repetitive) ซึ่งการทำงานในแต่ระรอบนี้เรียกว่าการสแกน (Scanning) และการสแกนในแต่ระรอบหากมีการเปลี่ยนแปลงของสัญญาณอินพุทแล้ว ผลของการสแกนก็จะถูกประมวลผลไปยังเอาท์พุทไปในทันที โดยทั่วไปแล้วการสแกนของ PLC ในรอบหนึ่ง ๆ จะเร็วมาก คือใช้เวลาเป็นมิลลิวินาที หรือไมโครวินาที ทั้งนี้ก็ขึ้นอยู่กับขนาดของโปรแกรมด้วย เพราะในระหว่างการสแกนคำสั่งทุกคำสั่งในโปรแกรมจะถูกประมวลผลหมด ดังนั้นโปรแกรมที่ใช้เวลาในการสแกนน้อย ก็จะมีการวนรอบเพื่อตรวจรับอินพุตถี่กว่าโปรแกรมที่ใช้เวลาในการสแกนมากเป็นผลให้การตอบสนองต่อการควบคุมรวดเร็วกว่าด้วย

## 2) หน่วยความจำ (memory unit)

เนื่องจาก PLC เป็นอุปกรณ์ที่ประมวลผลด้วยไมโครโพรเซสเซอร์ ดังนั้นจึงต้องมีหน่วยความจำสำหรับเก็บคำสั่งและข้อมูลต่าง ๆ เพื่อให้ไมโครโพรเซสเซอร์เรียกใช้ในการประมวลผลโดย ขนาดของหน่วยความจำนี้จะเป็นตัวกำหนดขีดความสามารถของระบบด้วย ซึ่งปกติแล้วการอ้างถึงคำสั่งที่ใช้ในโปรแกรมจะอ้างเป็นแอดเดรส (address) โดยคำสั่งเริ่มแรกจากแอดเดรสที่ 0000 แล้วคำสั่งถัดไปก็จะอยู่ในแอดเดรสที่ต่อเนื่องกัน บางครั้งผู้ผลิต PLC จะบอกมาในคู่มือ การใช้เครื่องด้วยว่าผู้ใช้สามารถเขียนได้ไม่เกินแอดเดรสที่เท่าไหร่

นอกจากนี้ PLC ยังให้ผู้ใช้สามารถเลือกใช้หน่วยความจำได้หลายชนิดอีกด้วย เช่น

- ROM (Read Only Memory) เป็นหน่วยความจำที่ไม่สามารถแก้ไขข้อมูลภายในได้ แต่ สามารถเก็บข้อมูลขณะไม่มีไฟเลี้ยง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- RAM (Random Access Memory)เป็นหน่วยความจำที่แก้ไขข้อมูลได้ แต่ต้องใช้ไฟเลี้ยงเพื่อรักษาข้อมูลเอาไว้
- PROM (Programmable ROM) เป็นรอมชนิดหนึ่งที่ผู้ใช้สามารถเขียนโปรแกรมลงได้ด้วยอุปกรณ์พิเศษ
- EPROM (Erasable Programmable ROM ) เหมือน PROM แต่สามารถลบข้อมูลได้ ทั้งนี้สามารถศึกษาได้จากสเปกของ PC/PLC ของแต่ละบริษัท

โดยลักษณะของการใช้งานหน่วยความจำนั้น สามารถแบ่งหน่วยความจำได้เป็น 2 ส่วนใหญ่ ๆ คือ

- หน่วยความจำของระบบ (system memory) เป็นหน่วยความจำที่ใช้เก็บ ระบบการจัดการและข้อมูลที่ใช้งาน
- หน่วยความจำของผู้ใช้ (user memory) เป็นหน่วยความจำที่ใช้เก็บโปรแกรม และข้อมูลการใช้งาน ซึ่งผู้ใช้สามารถเลือกหน่วยความจำชนิดนี้ได้ ดังที่กล่าวมาแล้ว

ทั้งนี้ภายใน PLC อาจมี RAM ประกอบอยู่เป็นส่วนหนึ่ง ของหน่วยความจำเพื่อเก็บสถานะของอินพุท / เอาท์พุทรีเลย์ (relay) ตัวตั้งเวลา (timer) ตัวนับ (counter) หรือข้อมูลอื่น ๆ ของระบบที่อาจมีการเปลี่ยนแปลงค่า (update) ระหว่างการประมวลผลร่วมอยู่ด้วย

### 3) หน่วยอินพุท-เอาท์พุท (input-output unit)

หน่วยอินพุท-เอาท์พุทของ PLC ที่ใช้ติดต่อกับระบบควบคุมนั้น จะเป็นสัญญาณดิจิตอลเท่านั้น ซึ่งในการใช้งาน ผู้ใช้ต้องศึกษาข้อกำหนดในการใช้งาน ของเครื่องอย่างละเอียด

โดยทั่วไปแล้ว สัญญาณไฟฟ้าที่จะรับเข้าไปสู่วงจรภายใน ทั้งทางด้านอินพุท-เอาท์พุทจะไม่ต่อถึงกันโดยตรง แต่จะมีการปรับแรงดันไฟฟ้าให้มีค่าที่พอเหมาะก่อน แล้วส่งสัญญาณผ่านการเชื่อมโยงทางแสง (optic coupling) เพื่อป้องกันผลกระทบจากไฟฟ้าแรงดันสูง และสัญญาณรบกวน ซึ่ง PLC แต่ละรุ่นจะมีลักษณะของวงจรเชื่อมต่อดังกล่าวคล้าย ๆ กัน

การติดต่อระหว่างอุปกรณ์อินพุท-เอาท์พุท กับ PLC บางครั้ง อาจมีการใช้อุปกรณ์แปลงสัญญาณ (signal converter) หรือพวกอุปกรณ์รับ / ส่ง(transmitter) เพื่อปรับแปลงสัญญาณให้ใช้ร่วมกับ PLC ได้ เช่น การแปลงสัญญาณจากแรงดัน หรืออุณหภูมิให้เป็นสัญญาณไฟฟ้าที่ PLC สามารถเข้าใจได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อุปกรณ์ไฟฟ้าสำหรับตรวจจับสัญญาณที่มีการนำมาประยุกต์ใช้กับ PLC ได้แก่ สวิตช์ลูกลอย สวิตช์ความดัน เทอร์โมสตัต ลิมิตสวิตช์ โฟลว์สวิตช์ และสวิตช์แบบไม่ต้องสัมผัส เช่น สวิตช์ลำแสง ตัวตรวจจับสี ตัวตรวจจับวัตถุ ตัวตรวจจับบาร์โคด ตัวตรวจจับความสูงของชิ้นงาน ตรวจจับความเร็วของการไหล หรือแม้แต่ตัวกำเนิดสัญญาณบางชนิดก็มีการนำมานใช้งานกับ PLC ในการควบคุมด้วยเช่นกัน

ข้อสังเกตุอีกอันหนึ่งก็คือ PLC ส่วนใหญ่นั้นจะมีอินพุท-เอาท์พุท ที่ทำงานลักษณะเปิด-ปิด (on-off) โดยทั้งหน้าสัมผัสอินพุท และหน้าสัมผัสเอาท์พุทจะมีจุดร่วม (common)ของหน้าสัมผัสร่วมนั่นเอง

**4) อุปกรณ์ต่อร่วม** (peripheral device)

ดังที่ได้กล่าวมาแล้วว่า PLC ถูกออกแบบมาให้ยืดหยุ่นกับการใช้งานสูง และมีโครงสร้างคล้ายคอมพิวเตอร์ ทำให้ PLC สามารถเชื่อมต่อได้กับอุปกรณ์อื่น ๆ ได้มากมาย ทั้งนี้ความสามารถในการเชื่อมต่อกับอุปกรณ์อื่น ขึ้นอยู่กับการออกแบบของผู้ผลิต และขนาดของเครื่องคอมพิวเตอร์ (PC) นั้น ซึ่งผู้ผลิตส่วนใหญ่มักออกแบบมาคล้าย ๆ กัน อุปกรณ์หลัก ๆ ที่ PC สามารถติดต่อได้แก่

- หน่วยป้อนโปรแกรม (programming console)
- เครื่องป้อนโปรแกรมผ่านจอภาพ (CRT programmer)
- เครื่องพิมพ์ (printer)
- เครื่องบันทึกข้อมูลลงหน่วยความจำ (memory burner)
- จอภาพแสดงผล (terminal monitor)
- หน่วยความจำแบบเทปคาสเซตต์ (memory cassette unit)
- คอมพิวเตอร์ (computer)

การใช้งานทั่วไปนั้นมักจะใช้ PLC ร่วมกับคอมพิวเตอร์ซึ่งมีทั้งฟังก์ชันที่ช่วยในการเขียนและตรวจสอบโปรแกรมมากมาย ประกอบกับทั่วไปตามโรงงาน หรือห้องควบคุมต่าง ๆ มักจะมีเครื่องคอมพิวเตอร์ไว้ใช้งานอยู่แล้ว การใช้คอมพิวเตอร์ติดต่อกับ PLC จึงกระทำได้สะดวก

PLC ที่ติดตั้งอยู่โดยทั่วไปนั้น จะประกอบไปด้วยชุดของ CPU หน่วยอินพุท-เอาท์พุท และอาจมีหน่วยขยายอินพุท-เอาท์พุท หรือหน่วยขยาย CPU ร่วมอยู่ด้วย ซึ่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อุปกรณ์ทั้งหมด อาจถูกจัดให้อยู่ในกล่องเป็นชุดสำเร็จก็ได้ แต่ก็อาจไม่มีอุปกรณ์สำหรับการป้อนโปรแกรม เพราะอุปกรณ์เหล่านี้ เมื่อป้อนโปรปกรมให้กับ CPU แล้วสามารถถอดไปใช้งานกับเครื่องอื่น ๆ ได้

สิ่งที่กล่าวมาทั้งหมดตั้งแต่ต้น ก็คือคุณสมบัติโดยทั่วไปของ PLC ซึ่งโปรแกรมในโครงงานของเราต้องเชื่อมต่อ และทำงานร่วมกับ PLC นี้

**การใช้ซอฟต์แวร์คอมพิวเตอร์ (Computer Software)**

PLC สามารถใช้ซอฟต์แวร์คอมพิวเตอร์ เพื่อทำหน้าที่ได้หลายๆอย่าง เช่น ใช้ซอฟแวร์ทำการป้อนโปรแกรม แก้ไขโปรแกรม ดูการทำงานของโปรแกรม เป็นต้น ซอฟแวร์แต่ละบริษัทจะมีวิธีการไม่เหมือนกันแต่มีจุดประสงค์ใกล้เคียงกัน

**การเรียกชื่ออุปกรณ์ควบคุม**

จะเรียกชื่อตัวควบคุมตัวนี้ว่า PLC หรือ PC ถูกต้องกว่า?

จากหลักการณ์พื้นฐานแล้ว อุปกรณ์ควบคุมตัวนี้จะทำงานในลักษณะเลขฐานสอง คือ “ปิด” หรือ “เปิด” “ON” หรือ "OFF" หรือสัญญาณลอจิก (Logic) เท่านั้น แต่ปัจจุบันมิได้เป็นเช่นนั้นต่อไปอีกแล้วคือ สามารถรับส่งสัญญาณอินพุท(Input)แบบต่อเนื่อง หรือสัญญาณอนาลอก (Analog) ได้ดังนั้น การเรียกชื่อว่า PLC จึงไม่น่าถูกต้อง ควรเรียกว่า PC จึงจะถูกต้องกว่า (ตัว L ในตัวย่อ PLC มาจากคำว่า Logic) อย่างไรก็ตามเพื่อไม่ให้สับสนกับคำว่า PC ที่เป็นชื่อเรียกของ Personal Computer ยังคงเรียกเป็น PLC เช่นเดิม

**คอมพิวเตอร์กับ PLC**

PLC เป็นคอมพิวเตอร์เฉพาะเครื่องหนึ่ง จึงมีโครงสร้างเหมือนคอมพิวเตอร์ แต่มีข้อแตกต่างดังต่อไปนี้คือ

1. PLC ถูกออกแบบให้ทนต่อสภาพแวดล้อมของโรงงานอุตสาหกรรม เช่น ความร้อน ความหนาว ระบบไฟฟ้ารบกวน การสั่นสะเทือน การกระแทก
2. การใช้โปรแกรม PLC จะไม่ยุ่งยากเหมือนคอมพิวเตอร์ PLC จะมีระบบตรวจสอบตัวเอง ทำให้ใช้งานง่ายและบำรุงรักษาง่าย
3. PLC ทำงานตามที่โปรแกรมเอาไว้เพียงโปรแกรมเดียว ทำให้ไม่ยุ่งยาก ส่วนคอมพิวเตอร์จะทำงานที่โปรแกรมหลายๆ โปรแกรมพร้อมกัน จึงมีความยุ่งยากกว่า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. PLC ใช้ควบคุมการผลิตทุกชนิด ทั้งแบบอนาลอก และแบบลอจิก (ON-OFF)

**ความสามารถของ PLC**

PLC สามารถควบคุมงานได้ 3 ลักษณะ คือ

1. **งานที่ทำตามลำดับก่อนหลัง (Sequence Control)** ตัวอย่างเช่น
   (1) การทำงานของระบบรีเลย์
   (2) การทำงานของระบบไทเมอร์ เคาน์เตอร์
   (3) การทำงานของ P.C.B Card
   (4) การทำงานในระบบกึ่งอัตโนมัติ ระบบอัตโนมัติ หรืองานที่เป็นกระบวนการของเครื่องจักรกลต่างๆ
2. **งานควบคุมสมัยใหม่ (Sophisticated Control)** ตัวอย่างเช่น
   (1) การทำงานทางคณิตศาสตร์ เช่น บวก ลบ คูณ หาร
   (2) การควบคุมแบบอนาล็อก (Analog Control) เช่น การควบคุมอุณหภูมิ (Temperature) การควบคุมความดัน (Pressure) เป็นต้น
   (3) การควบคุม P.I.D. ( Proportional-Integral-Derivation)
   (4) การควบคุมเซอร์โวมอเตอร์ ( Servo-motor Control)
   (5) การควบคุม Stepping-motor
   (6) Information Handling
3. **การควบคุมเกี่ยวกับงานอำนวยการ (Supervisory Control)** ตัวอย่างเช่น
   (1) งานสัญญาณเตือน (Alarm) และ Process Monitoring
   (2) Fault Diagnostic and Monitoring
   (3) งานต่อร่วมกับคอมพิวเตอร์ (RS-232C/RS422)
   (4) Printer/ASCII Interfacing
   (5) งานควบคุมอัตโนมัติในโรงงานอุตสาหกรรม (Factory Automation Networking)
   (6) LAN (Local Area Network)
   (7) WAN (Wide Area Network)
   (8) FA.,FMS.,CIM. เป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขนาดของ PLC

1. ขนาดเล็ก มีจำนวนอินพุท/เอาท์พุทไม่เกิน 128 จุด
2. ขนาดกลาง มีจำนวนอินพุท/เอาท์พุทไม่เกิน 1024 จุด
3. ขนาดใหญ่ มีจำนวนอินพุท/เอาท์พุทไม่เกิน 4096 จุด
4. ขนาดใหญ่มาก มีจำนวนอินพุท/เอาท์พุทไม่เกิน 8192 จุด

การติดตั้ง PLC

1. ข้อควรพิจารณาก่อนติดตั้ง PLC
   (1) พื้นที่ในการติดตั้งมีเพียงพอหรือไม่
   (2) จะต้องเผื่อไว้ขยายในอนาคตหรือไม่
   (3) การซ่อมบำรุงต้องทำได้ง่าย
   (4) อุณหภูมิที่เกิดขึ้นกับเครื่องจักรมีผลกระทบกับ PLC หรือไม่
   (5) วิธีการป้องกัน PLC จากสภาพแวดล้อมที่ไม่ปลอดภัย
2. สภาพแวดล้อมหรือสถานที่ไม่ควรติดตั้ง PLC
   (1) มีแสงแดดส่องโดยตรง
   (2) มีอุณหภูมิต่ำกว่า $0^{\circ}$ C หรือสูงกว่า $55^{\circ}$ C
   (3) มีฝุ่น หรือไอเกลือ
   (4) มีความชื้นมาก
   (5) มีก๊าซที่มีคุณสมบัติกัดกร่อน หรือไวไฟ
   (6) สั่นสะเทือนมาก

ตู้ควบคุมสำหรับ PLC

1. ต้องป้องกันไม่ให้ PLC เสียหายจากการใช้งานหรือจากส่วนอื่นๆ เช่น จากสภาพแวดล้อม หรือสิ่งปนเปื้อนในอากาศ เช่น ความชื้น น้ำมัน ฝุ่นผง ก๊าซที่มีฤทธิ์กัดกร่อน
2. มีขนาดใหญ่เพียงพอ สะดวกในการเดินสายไฟต่างๆ
3. ควรติดตั้งตู้ PLC ห่างจากแผงควบคุมไฟฟ้าแรงสูงอย่างน้อย 8 นิ้ว
4. มีสายดิน
5. ควรแยกการติดตั้งกับอุปกรณ์ไฟฟ้าแรงสูง
6. ควรแยกการติดตั้งกับอุปกรณ์ที่มีความร้อนสูง เช่น ฮีทเตอร์ หม้อแปลง หรือตัวต้านทานขนาดใหญ่
7. ไม่ควรให้ PLC ติดตั้งอยู่บนเพดาน หรืออยู่กับพื้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

8. ถ้ามีอุณหภูมิสูงกว่า 60° C ควรติดตั้งพัดลมเป่าระบายความร้อน
9. ควรต่อสายดินแยกจากอุปกรณ์ไฟฟ้าตัวอื่น คือ สายดินควรมีขนาด 2 ตารางมิลลิเมตรหรือใหญ่กว่า และค่าความต้านทานของสายดินไม่ควรเกิน 100 โอห์ม

**ตัวอย่างการใช้ PLC ในอุตสาหกรรมต่าง ๆ**

1. การผสมวัตถุดิบ
2. การขนถ่ายผลิตภัณฑ์
3. การกำจัดน้ำเสีย
4. การขุดเจาะน้ำมัน
5. การแปรรูปไม้
6. การทำเยื่อกระดาษ
7. การย่อยเยื่อไม้
8. การประกอบชิ้นส่วนรถยนต์
9. การพ่นสีรถยนต์
10. การตรวจสอบรถยนต์
11. การประหยัดพลังงาน
12. การแยกแร่
13. การกำจัดน้ำเสีย

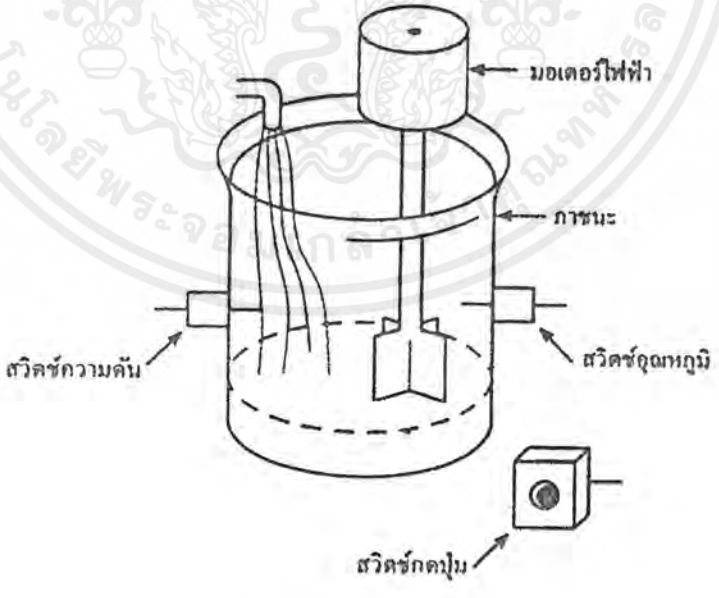
เป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# บทที่ 2
# การทำงานของ PLC

**การทำงานของ PLC**

เครื่องจักรที่ควบคุมด้วย PLC จะมีความสามารถเขียนโปรแกรมการทำงานของเครื่องจักรและมีความยืดหยุ่นในการเขียน โปรแกรม เช่น การเปลี่ยนแปลงแก้ไขเพิ่มเติมก็สามารถกระทำได้ ซึ่งรวมถึงมีไทเมอร์(Timer) เคาท์เตอร์ (Counter) หรือคำสั่งพิเศษ ต่างๆ เช่น MOV Data และอื่นๆ อีกมากมาย เพื่อใช้ควบคุมอุปกรณ์ภายนอก ไม่ว่าจะเป็นมอเตอร์ โซลินอยด์ (Solenoid) หลอดไฟ เป็นต้น นอกจากนี้ถ้ามีการติดต่อสื่อสารระหว่าง PLCกับคอมพิวเตอร์ เพื่อแลกเปลี่ยนข้อมูลระหว่างกันหรืออาจจะติดต่อกับจอ ชนิดสัมผัส (Touch Screen)เพื่ออำนวยความสะดวกต่อสัญญาณอินพุทและสัญญาณ เอาท์พุท ยิ่งไปกว่านั้นการติดต่อกับคอมพิวเตอร์ เพื่อให้คอมพิวเตอร์เป็นตัวควบคุม PLC อีกทีหนึ่ง ซึ่งจะทำให้ขีดความควบคุมสูงขึ้นอีก และต่อไปนี้จะได้กล่าวถึงการทำงานของ PLC โดยเริ่มจากตัวอย่างการควบคุมมอเตอร์กวนของเหลวในภาชนะโดยมีเงื่อนไขต่างๆ

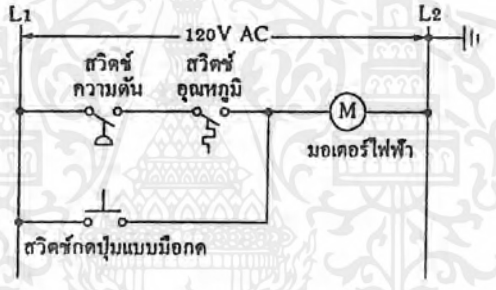



รูปที่ 2.1 ตัวอย่างการกวนของเหลว

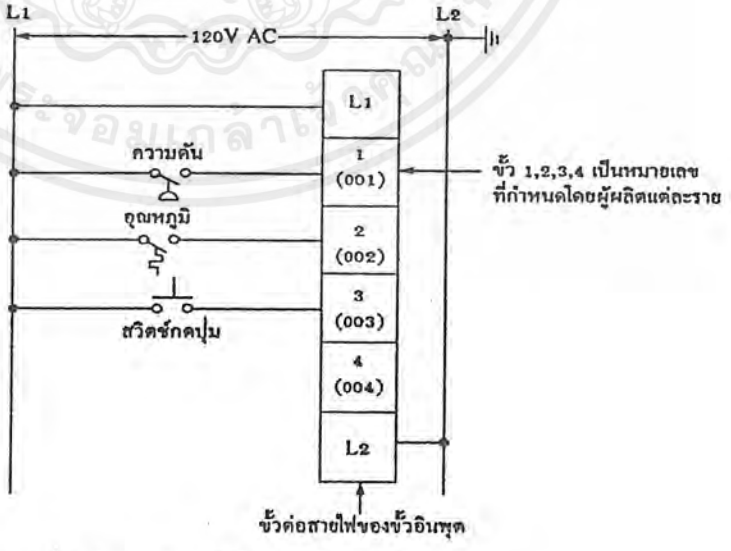
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตามรูปที่ 2.1 ใช้มอเตอร์ไฟฟ้าสำหรับกวนของเหลวที่บรรจุอยู่ในภาชนะ เมื่อมีเงื่อนไขว่ามีอุณหภูมิและความดันถึงค่าที่กำหนดเท่านั้น หรือสามารถใช้มอเตอร์ไฟฟ้าทำงานด้วยการกดสวิตช์ก็ได้ โดยการแยกสวิตช์ออกมาต่างหาก

มอเตอร์ไฟฟ้าจะหมุนได้ก็ต่อเมื่อหน้าคอนแทคของเซ็นเซอร์อุณหภูมิ และความดันต่อกัน การต่อวงจรทำได้จากการใช้วงจรรีเลย์ (Relay) ตามรูปที่ 2.2 และการใช้ PLC ตามรูปที่ 2.3 และ 2.4 ในกรณีของการใช้รีเลย์นั้น มอเตอร์ (M) ทำงานได้เมื่อสวิตช์ของความดันและอุณหภูมิต่อกัน หรือสวิตช์มือกดถูกกด แต่ถ้าใช้ PLC ทำงาน ตามรูปที่ 2.3 เป็นการต่ออุปกรณ์ด้านอินพุทซึ่งมีคอนแทคของความดัน คอนแทคของอุณหภูมิ และคอนแทคของสวิตช์กดปุ่มแบบมือกด ซึ่งต่อไปยังขั้วสัญญาณเข้า (Input Module) ส่วนรูปที่ 2.4 เป็นการต่ออุปกรณ์ทำงาน (มอเตอร์ไฟฟ้า) เข้ากับขั้วต่อสัญญาณออก (Output Module)

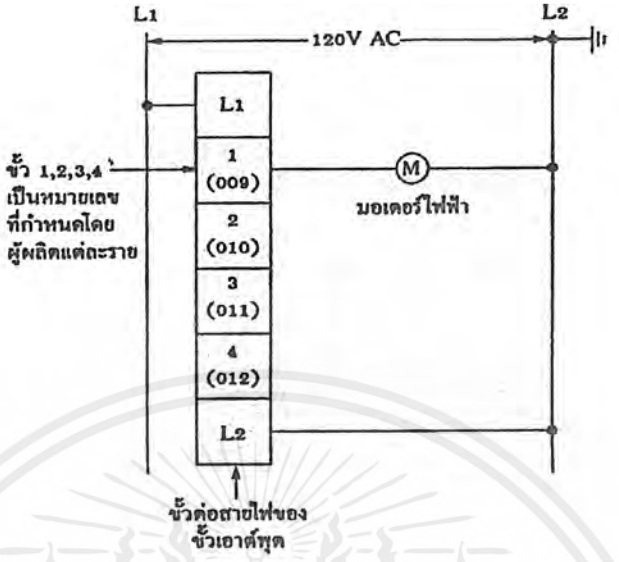



รูปที่ 2.2 วงจรการใช้รีเลย์



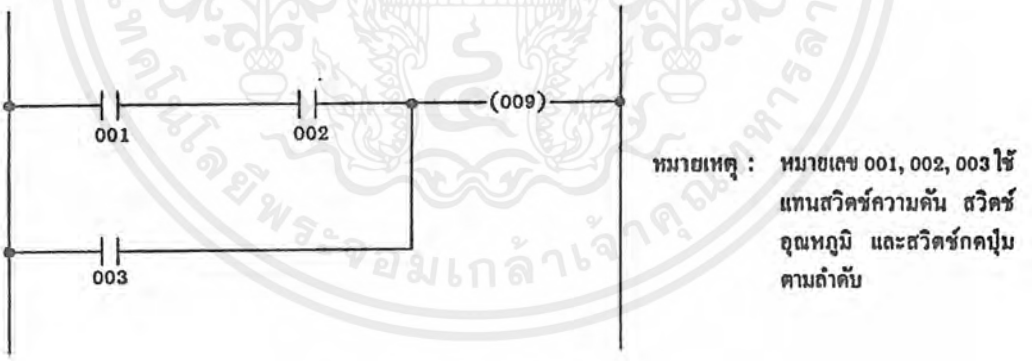

รูปที่ 2.3 วงจรการใช้ PLC (อินพุท โมดูล)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้




รูปที่ 2.4 วงจรการใช้ PLC (เอาท์พุท โมดูล)

การเขียน PLC แลดเดอร์ (PLC Ladder Logic Diagram) เพื่อส่งโปรแกรมเข้าไปยังหน่วยความจำของ CPU ดูได้จากรูปที่ 2.5




รูปที่ 2.5 วงจรแลดเดอร์ (PLC Ladder Logic Diagram)

**การทำงานของวงจรแลดเดอร์**

เมื่อ PLC อยู่ในสภาวะพร้อมทำงาน (RUN Mode) แล้ว เมื่อมีโปรแกรมถูกป้อนไปยังหน่วยความจำของ CPU ทำให้ CPU ประมวลผลและได้ผลลัพธ์เป็นสัญญาณเอาท์พุท หน้าคอนแทคตามรูปที่ 2.5 ซึ่งเป็นชนิดปกติเปิด (Normally Open) เพราะฉะนั้น ถ้าคอนแทค 001 และ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

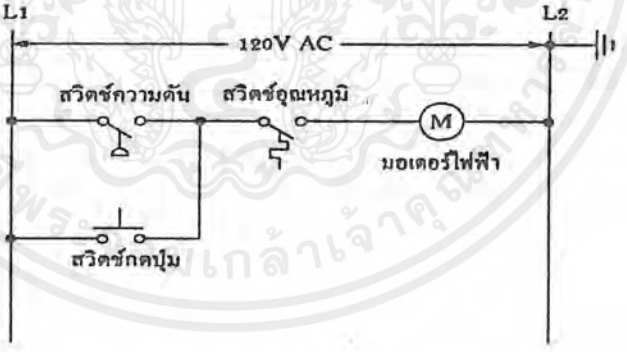
002 ต่อกัน ก็ทำให้เกิดเอาท์พุท 009 หรือหน้าคอนแทค 003 ต่อกัน ก็ทำให้เกิดเอาท์พุท 009 ได้ เช่นกันลักษณะนี้เรียกว่ารัง (Rung) คือ มีสัญญานอินพุทหนึ่งหรือมากกว่า

**หลักการทำงานของ PLC**

อธิบายตามหลักการณ์ต่างๆ มีดังต่อไปนี้คือ เมื่อมีสัญญาณอินพุทเข้ามาจะถูกเก็บเป็นความจำไว้ในส่วนของความจำ (หน้าคอนแทคที่ต่อกัน เรียกว่า ลอจิก 1ส่วนคอนแทคที่ไม่ต่อกัน เรียกว่า ลอจิก 0) หลังจากนั้นแลดเดอร์ไดอะแกรมก็จะสรุปผลรวมกับคอนแทคภายในว่าให้เป็นคอนแทคเปิด (Open) หรือปิด (Closed) ขึ้นอยู่กับการบันทึกของหน่วยความจำ ถ้าต้องการเอาท์พุท ค่าของลอจิกต้องเป็นเลข 1 ซึ่งหมายถึงชุดหน้าคอนแทคของโมดูล อินเตอร์เฟส (Module Interface) ต่อกัน แต่ถ้าไม่มีกระแสไฟฟ้าไหลในวงจรทำให้ขดลวดความจำของคอยล์ (Coil Memory) มีค่าเป็นลอจิกเลข 0 และโมดูลอินเตอร์เฟส (Module Interface) ไม่ต่อกัน

การทำงานของ PLC เมื่อครบ 1 รอบลำดับดังกล่าวนี้เรียกว่า สแกน (Scan) ส่วน Scan Time คือเวลาที่ต้องการสำหรับ 1 รอบการทำงาน ซึ่งเป็นตัววัดค่าความเร็วการทำงานของ PLC 1 สแกน ใช้เวลาประมาณ 1-100 ms ขึ้นอยู่กับความยาวของโปรแกรมและชนิดของอินพุท/เอาท์พุท

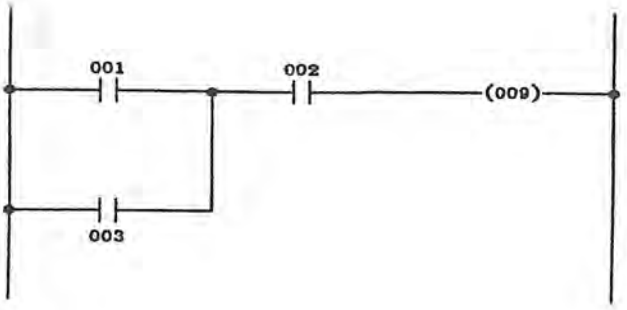
**การปรับปรุงหรือแก้ไขวงจร**




รูปที่ 2.6 วงจรลีเลย์ที่แก้ไขใหม่

จากที่กล่าวในส่วนเครื่องผสมของเหลวข้างต้นนั้น ถ้าต้องการปรับปรุงวงจรเสียใหม่ตามรูปที่ 2.6 นี้ คือ เปลี่ยนแปลงในส่วนของสวิตช์กดปุ่ม (Push Button Switch) ที่สามารถทำงานได้ที่ความดันใดๆ ก็ได้แต่ต้องมีอุณหภูมิตามที่กำหนดเท่านั้น ทำได้โดยให้สวิตช์กดปุ่มกับสวิตช์ความดันต่อแบบขนาน (OR)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้




รูปที่ 2.7 แลดเดอร์ ลอจิก ไดอะแกรมของ PLC ที่แก้ไขใหม่

เมื่อปรับปรุงแก้ไขวงจรของรีเลย์ก็ต้องเดินสายไฟใหม่ แต่ถ้าใช้ PLC เป็นตัวควบคุมนั้น ไม่จำเป็นต้องเดินสายไฟฟ้าที่เข้ากับอุปกรณ์ใหม่ เพียงแต่เขียนแลดเดอร์ ลอจิก ไดอะแกรมของ PLC ใหม่เท่านั้นตามรูปที่ 2.7

**วงจรตรรก (ลอจิก)**

หลักการทำงานของ PLC ใช้วงจรตรรก (ลอจิก) เพื่อให้เกิดสัญญาณเอาท์พุทที่มีเงื่อนไข (สัญญาณอินพุท) ชนิดต่างๆ หลักการณ์ของวงจรตรรก มีดังต่อไปนี้

วงจรตรรก หมายถึง วงจรไฟฟ้าที่ประกอบด้วยอุปกรณ์อิเล็กทรอนิกส์ หรือระบบรีเลย์ที่มีสัญญาณเพียง 2 ระดับ หรือ 2 สภาวะเท่านั้น PLC ใช้สัญญาณไฟฟ้า 2 ระดับ แทน 2 เหตุการณ์ที่แตกต่างกัน เช่น การปิดเปิดวาล์ว การปิดเปิดสวิตช์ เป็นต้น วงจรตรรกมี 2 ชนิด คือ แบบบวก (Positive Logic) แบบลบ (Negative Logic) ลอจิกแบบบวกจะใช้สัญญาณไฟฟ้าระดับสูง แทนสภาวะลอจิก "1" และใช้สัญญาณไฟฟ้าระดับต่ำ แทนสภาวะลอจิก "0" ส่วนวงจรลอจิกแบบลบจะใช้สัญญาณไฟฟ้าระดับต่ำ แทนสภาวะลอจิก "1" และใช้สัญญาณไฟฟ้าระดับสูง แทนสภาวะลอจิก "0"

สภาวะทางลอจิก คือ สภาวะ "1" หรือ "0" ใช้แทนการทำงานของอุปกรณ์ที่เปลี่ยนแปลง 2 สภาวะ ระบบควบคุมที่ใช้ระบบรีเลย์ และ PLC จะนำเอาสภาวะของอุปกรณ์เหล่านี้มาปฏิบัติลอจิกด้วยกัน เพื่อให้เข้ากันกับเงื่อนไขการควบคุม ปฏิบัติการลอจิกประกอบด้วย AND OR และ NOT เพื่อให้สภาวะอินพุทต่างๆ เช่น A,B ทำให้เกิดเอาท์พุท Y เป็นต้น

พีชคณิตบูลลีนมีไว้สำหรับอธิบายความสัมพันธ์ทางลอจิก ทำให้เข้าใจได้ง่ายยิ่งขึ้น ตัวอย่างสมการบูลลีนของรูปที่ 2.18 เขียนได้ว่า Y (แสงไฟฟ้า) = A(สวิตช์).B(หลอดไฟ) วงจรลอจิกที่ใช้วิธีการเดินสายไฟเชื่อมต่ออุปกรณ์ต่างๆ เช่น รีเลย์ สวิตช์ ซึ่งมีความยุ่งยากและแก้ไขเพิ่มเติมได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

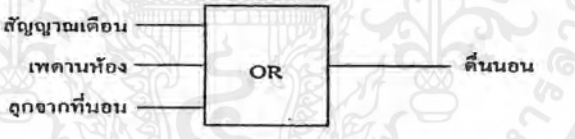
สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง


ยาก ส่วน PLC ใช้โปรแกรมลอจิกกำหนดเงื่อนไขการควบคุม แทนการเดินสายไฟต่อเชื่อมอุปกรณ์ต่างๆ ดังกล่าวมาแล้ว จึงทำให้ง่ายขึ้น

PLC แทนวงจรรีเลย์ด้วยปฏิบัติการทางลอจิก AND OR และ NOT ซึ่งกำหนดตามเงื่อนไขที่ต้องการควบคุม โดยใช้คำสั่งหรือภาษา (PC Language) ภาษาพื้นฐานที่ PLC ใช้ในการควบคุมแบบ "ON" หรือ "OFF" คือภาษาแลดเดอร์ และภาษาบูลลีน ภาษาแลดเดอร์ใช้สัญลักษณ์ของหน้าสัมผัสในการเขียนโปรแกรม การเปลี่ยนวงจรรีเลย์ให้เป็นโปรแกรม PLC ทำได้โดยใช้หน้าสัมผัสแลดเดอร์แทนสัญญาณรีเลย์

การทำงานของอุปกรณ์ดิจิตอล (Digital Equipment) จะอยู่บนหลักการณ์พื้นฐานของลอจิกพื้นฐาน 3 ตัว คือ AND OR และ NOT แต่ละตัวจะมีหลักการณ์ของตัวเอง ต่อไปนี้จะให้ Y เป็นเอาท์พุท (Output) และสัญญาณอินพุท (Input) ให้เป็นตัวอักษร ABC ส่วนเลข 1 หมายถึง มีสัญญาณ เลข 0 หมายถึงไม่มีสัญญาณ


รูปที่ 2.8 ไฟในห้องจะติดได้ก็ต่อเมื่อต่อสะพานไฟและมีหลอดไฟ (Light Bulb) อยู่ในกล่องเท่านั้น




รูปที่ 2.9 สัญญาณให้ตื่นนอน

มีสัญญาณเตือน 3 อย่างเพื่อให้ตื่นนอน คือ มีสัญญาณดังขึ้น (Alarm) หรือเพดานห้องยุบลงมาหรือลูกออกจากที่นอน

**หลักการของ AND Gate**

AND Gate ทำให้เกิดสัญญาณเอาท์พุทได้ก็ต่อเมื่อ มีอินพุททั้ง A และ B มีค่า "1"

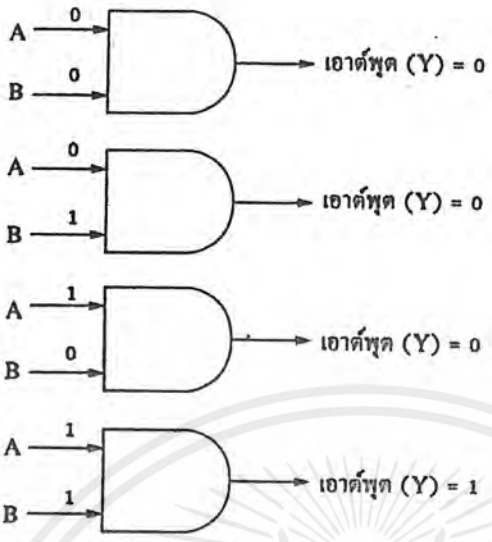

สัญลักษณ์ของ AND Gate

| อินพุต | | เอาต์พุต |
|---|---|---|
| A | B | Y |
| 0 | 0 | 0 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

ตารางตามความเป็นจริงของ AND

รูปที่ 2.10 สัญลักษณ์ของ AND ที่มีอินพุท 2 ตัว

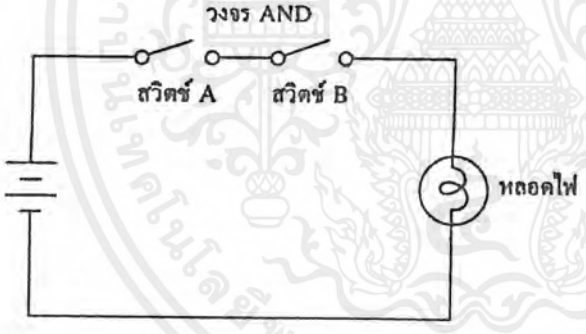
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
36863




| A | B | Y |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

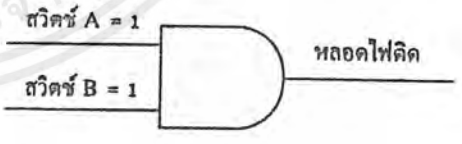
ตารางตามความเป็นจริง

รูปที่ 2.11 ถ้าอินพุททั้งหมดเป็น 1 จะได้เอาท์พุทเป็น 1 แต่ถ้าอินพุทตัวใดตัวหนึ่งเป็น 0 เอาท์พุทจะเป็น 0 ทันที




| สวิตช์ A | สวิตช์ B | หลอดไฟ |
|---|---|---|
| เปิด (0) | เปิด (0) | ดับ (0) |
| เปิด (0) | ปิด (1) | ดับ (0) |
| ปิด (1) | เปิด (0) | ดับ (0) |
| ปิด (1) | ปิด (1) | ติด (1) |

ตารางตามความเป็นจริง

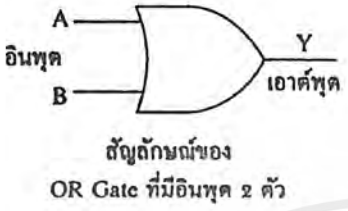



รูปที่ 2.12 หลอดไฟจะติดได้เมื่อสวิตช์ A และ B ปิดเท่านั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**หลักการของ OR Gate**

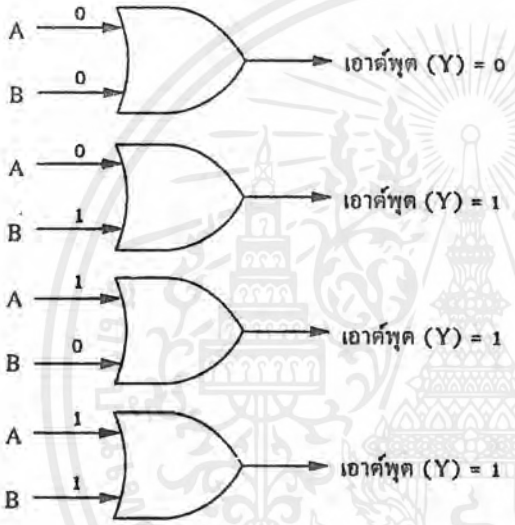
OR Gate สามารถมีอุปกรณ์หลายๆ ตัวก็ได้ แต่จะมีเอาท์พุทเพียงตัวเดียวเท่านั้น ถ้าเอาท์พุทเท่ากับ 1 แสดงว่ามีอินพุทตัวใดตัวหนึ่งหรือหลายตัวเท่ากับ 1




สัญลักษณ์ของ
OR Gate ที่มีอินพุต 2 ตัว

| อินพุต | | เอาต์พุต |
|---|---|---|
| A | B | Y |
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 1 |

ตารางตามความเป็นจริง

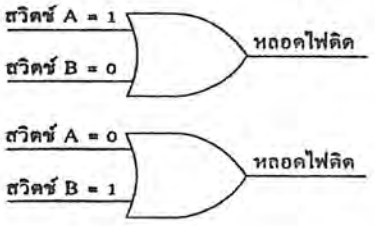



| อินพุต | | เอาต์พุต |
|---|---|---|
| A | B | Y |
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 1 |

ตารางตามความเป็นจริง

รูปที่ 2.13 สัญญาณของ OR ที่มีอินพุท 2 ตัว

หลักการพื้นฐาน คือ ถ้าอินพุทหนึ่งตัวหรือมากกว่า 1 ตัว มีค่าเป็น 1 เอาท์พุทจะเท่ากับ 1 แต่ถ้าอินพุททุกตัวมีค่าเป็น 0 เอาท์พุทจะเท่ากับ 0

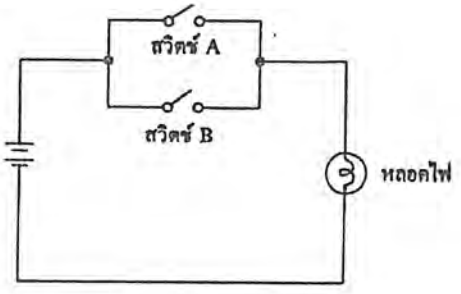



| สวิตช์ A | สวิตช์ B | หลอดไฟ |
|---|---|---|
| เปิด (0) | เปิด (0) | ดับ (0) |
| เปิด (0) | ปิด (1) | ติด (1) |
| ปิด (1) | เปิด (0) | ติด (1) |
| ปิด (1) | ปิด (1) | ติด (1) |

ตารางตามความเป็นจริง

รูปที่ 2.14 แสดงสภาวะของหลอดไฟและสวิตช์

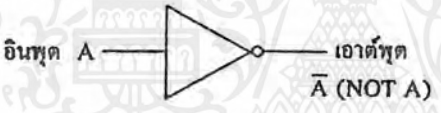
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้




รูปที่ 2.15 การทำงานของ OR Gate (OR Gate Function) หลอดไฟจะติดเมื่อสวิตช์ A หรือ B ปิด

**หลักการของ NOT Gate**

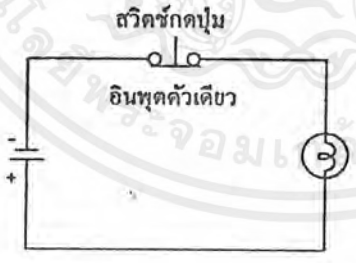
NOT Gate จะไม่เหมือนกับ AND หรือ OR Gate คือ NOT Gate จะมีอินพุทเพียงตัวเดียว เท่านั้น ถ้าเอาท์พุทเท่ากับ 1 แสดงว่าอินพุทเท่ากับ 0 ถ้าเอาท์พุทเท่ากับ 0 แสดงว่าอินพุทเท่ากับ 1




| A | NOT A |
|---|---|
| 0 | 1 |
| 1 | 0 |

ตารางตามความเป็นจริงของ NOT

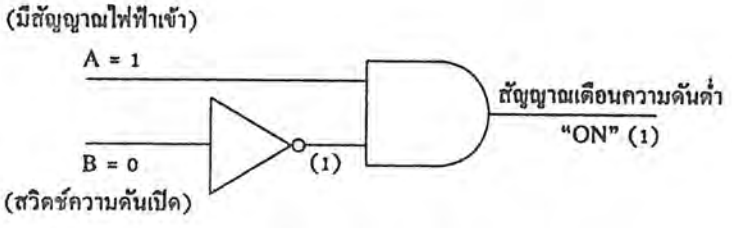
รูปที่ 2.16 สัญลักษณ์ของ NOT Gate




| สวิตช์กดปุ่ม | หลอดไฟ |
|---|---|
| ไม่กดสวิตช์ (0) | ดิด (1) |
| กดสวิตช์ (1) | ดับ (0) |

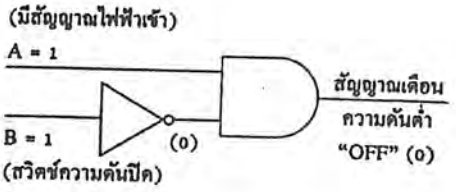
ตารางตามความเป็นจริง

รูปที่ 2.17 หลอดไฟจะติดถ้าสวิตช์ไม่ถูกกด




รูปที่ 2.18 ถ้ามีสัญญาณไฟฟ้ามีค่าเป็น (1) และสวิตช์ความดันเปิดจะมีสัญญาณเตือนเกิดขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



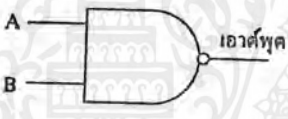

| สวิตช์ความดัน | สัญญาณไฟฟ้า | สัญญาณเตือนของความดัน |
|---|---|---|
| 0 | 1 | 1 |
| 1 | 1 | 0 |

ตารางตามความเป็นจริง

รูปที่ 2.19 ถ้ามีสัญญาณไฟฟ้ามีค่าเป็น (1) และสวิตช์ความดัน(Pressure Switch) ปิด (1) จะไม่มีสัญญาณเตือน

**หลักการของ NAND Gate**

NAND Gate นี้ทำงานตรงข้ามกับ AND Gate



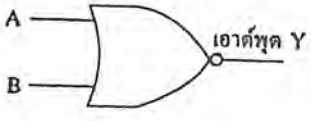

| อินพุต | | เอาต์พุต |
|---|---|---|
| A | B | Y |
| 0 | 0 | 1 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

ตารางตามความเป็นจริงของ NAND

รูปที่ 2.20 สัญลักษณ์ของ NAND Gate ที่มีอินพุท 2 ตัว

**หลักการของ NOR Gate**

NOR Gate จะทำงานตรงข้ามกับ OR Gate




| อินพุต | | เอาต์พุต |
|---|---|---|
| A | B | Y |
| 0 | 0 | 1 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 0 |

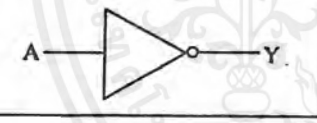
ตารางตามความเป็นจริงของ NOR

รูปที่ 2.21 สัญลักษณ์ของ NOR Gate ที่มีอินพุท 2 ตัว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**บูลลีน แอลจิบา (Boolean Algebra)**

การศึกษาทางพีชคณิตเกี่ยวกับจำนวนของไบนารี่และลอจิก เรียกว่า Boolean Algebra จุดประสงค์เพื่อหาวิธีการเขียนลอจิกที่ซับซ้อนให้ง่ายขึ้น ดังในตารางต่อไปนี้ จะเป็นการสรุปพื้นฐานการทำงานของ Boolean Algebra ซึ่งมีความสัมพันธ์กับ AND , OR และ NOT Gate สัญญาณอินพุทจะเขียนด้วยตัวอักษร A B C เป็นต้น ส่วนสัญญาณเอาต์พุทจะแทนด้วย Y และเครื่องหมายคูณหรือจุด หมายถึง AND เครื่องหมายบวกหมายถึง OR และขีดข้างบนตัวอักษรจะหมายถึง NOT

| สัญลักษณ์ของลอจิก | คำอธิบายของลอจิก | สมการบูลลีน |
|---|---|---|
|  | Y เป็น "1"<br>ถ้า A และ B เป็น "1" | $y = A \cdot B$<br>หรือ<br>$Y = AB$ |
|  | Y เป็น "1"<br>ถ้า A หรือ B เป็น "1" | $Y = A+B$ |
|  | Y เป็น "1" ถ้า A เป็น "0"<br>Y เป็น "0" ถ้า A เป็น "1" | $y = \overline{A}$ |

**Commutative Law**

$$A + B = B + A$$

$$A \cdot B = B \cdot A$$

**Associative Law**

$$(A + B) + C = A + (B + C)$$

$$(A \cdot B) \cdot C = A \cdot (B \cdot C)$$

**Distributive Law**

$$A \cdot (B + C) = (A \cdot B) + (A + C)$$

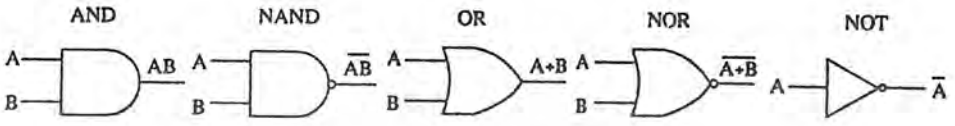
$$A + (B \cdot C) = (A + B) \cdot (A + C)$$

รูปที่ 2.22 สัญลักษณ์ของลอจิกและสมการบูลลีน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

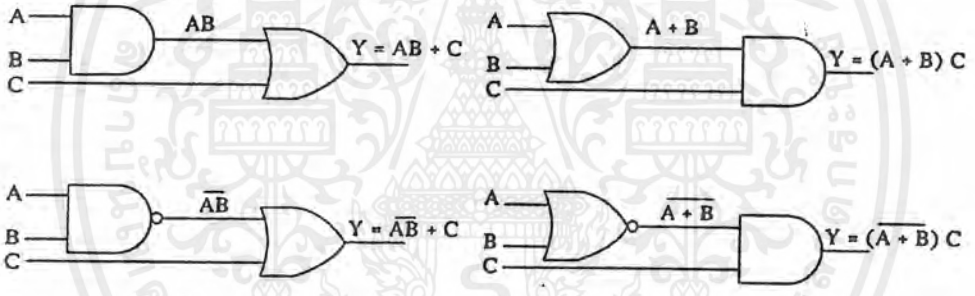
**ลอจิก 5 ชนิด**

สมการทั้งสาม ที่กล่าวมาแล้วใช้สำหรับเขียนโปรแกรมควบคุม PLC ให้ได้ง่ายขึ้น และต่อไปนี้จะเป็นตัวอย่างของลอจิกทั้งห้าชนิดซึ่งเป็นสมการบูลลีน (Boolean Equation) ดังกล่าวมาแล้ว




รูปที่ 2.23 สัญลักษณ์ของลอจิกทั้ง 5 ชนิด

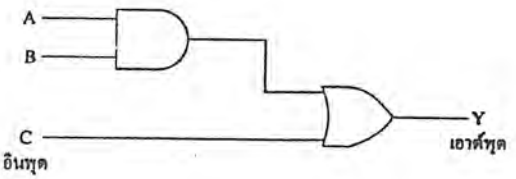
**ตัวอย่างของสมการบูลลีนชนิดต่าง ๆ**




รูปที่ 2.24 สมการบูลลีนชนิดต่างๆ

**เอาท์พุท Y = AB + C**

อินพุท A และ B ต่อกันแบบ AND ซึ่งได้เอาท์พุท AB แล้วต่อกับ C ด้วย OR ผลลัพธ์เป็นเอาท์พุท Y

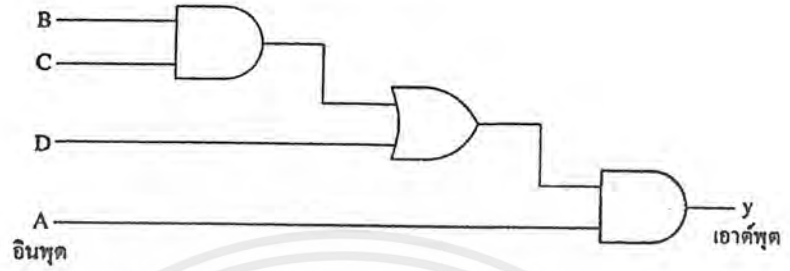



รูปที่ 2.25 เอาท์พุท Y = AB + C

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**เอาท์พุท** Y = A(BC + D)

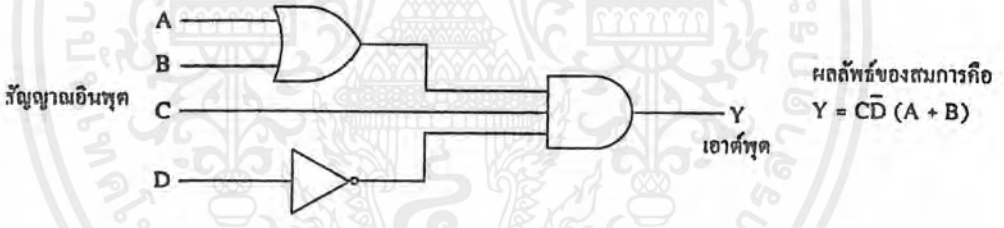
อินพุท B และ C ต่อกันแบบ AND ซึ่งได้เอาท์พุท BC แล้วต่อกับ D ด้วย OR ได้ผลลัพธ์เป็น BC+D แล้ว AND กับอินพุท A จึงได้เอาท์พุท Y




รูปที่ 2.26 เอาท์พุท Y = A(BC +D)

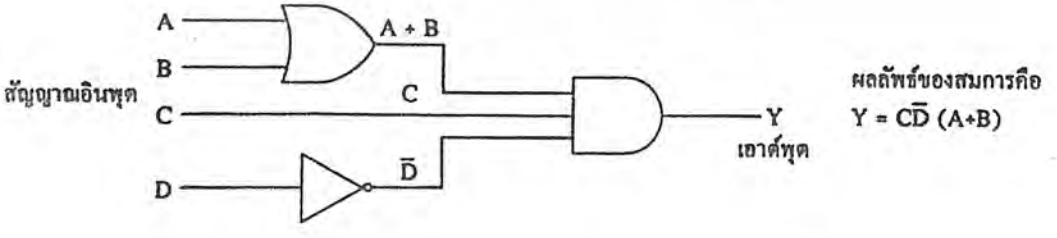
**วงจรในรูปของ Boolean Term**

**วงจรลอจิก A**




รูปที่ 2.27 วงจรลอจิก A

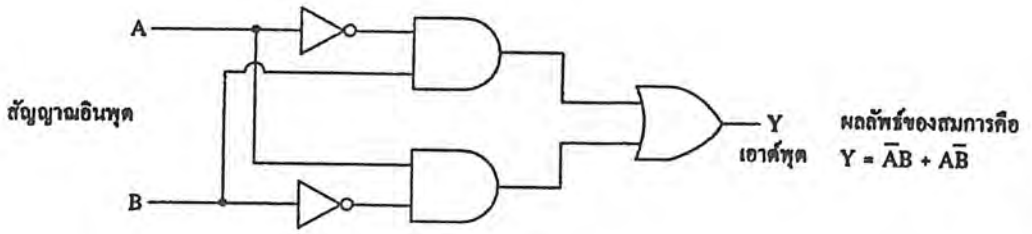
**วงจรในรูปของ Boolean Term จากวงจรลอจิก A**




รูปที่ 2.28 วงจรในรูปของ Boolean Term

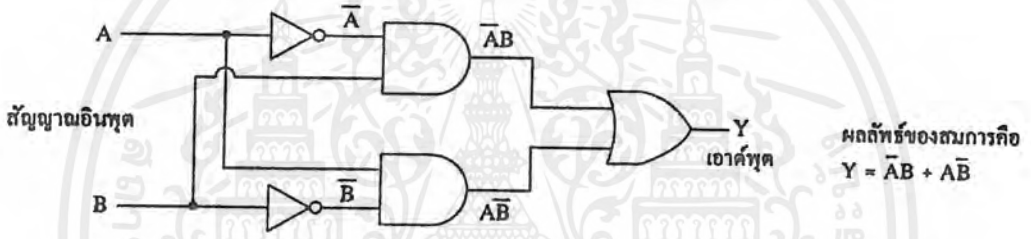
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

วงจรลอจิก B




รูปที่ 2.29 วงจรลอจิก B

วงจรในรูป Boolean Term จากวงจรลอจิก B



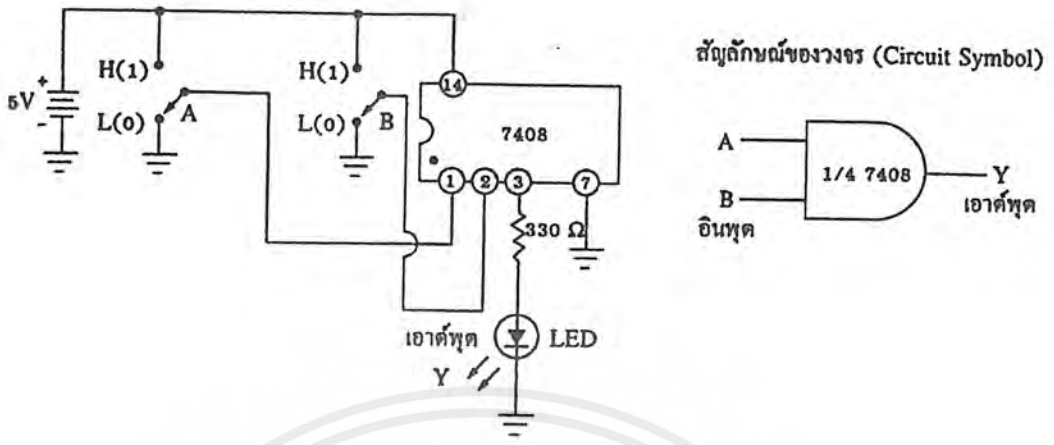

รูปที่ 2.30 วงจรในรูปของ Boolean Term

## ตัวอย่างเปรียบเทียบวงจรรีเลย์แลดเดอร์ ลอจิกแลดเดอร์ และลอจิกเกต

ตัวอย่างที่ 1


รูปที่ 2.31 สมการบูลลีน : AB =Y

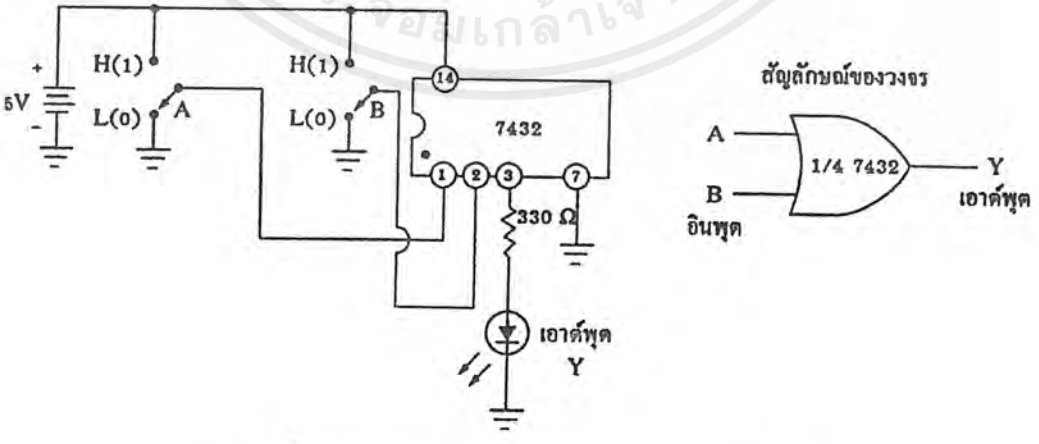
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้




รูปที่ 2.32 Logic Gate Schematic ของ AB = Y

ตัวอย่างที่ 2


รูปที่ 2.33 สมการบูลลีน : A +B = Y

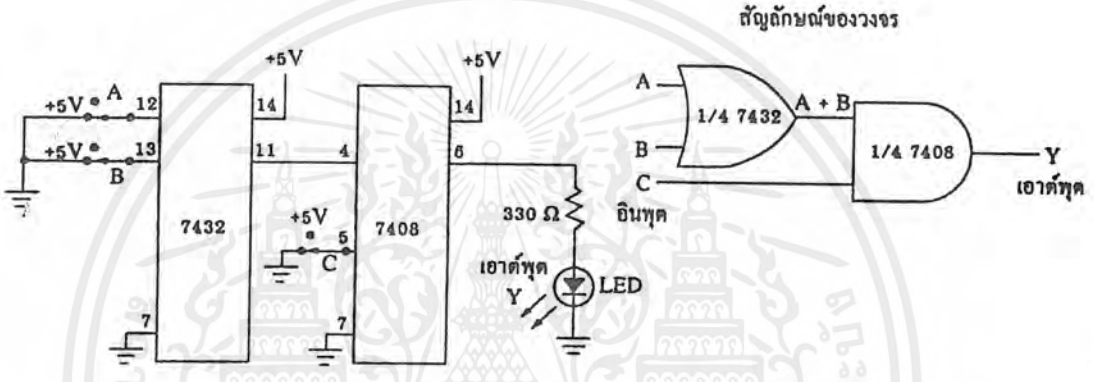



รูปที่ 2.34 Logic Gate Schematic ของ A + B = Y

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**ตัวอย่างที่ 3**


รูปที่ 2.35 สมการบูลลีน : (A + B)C = Y

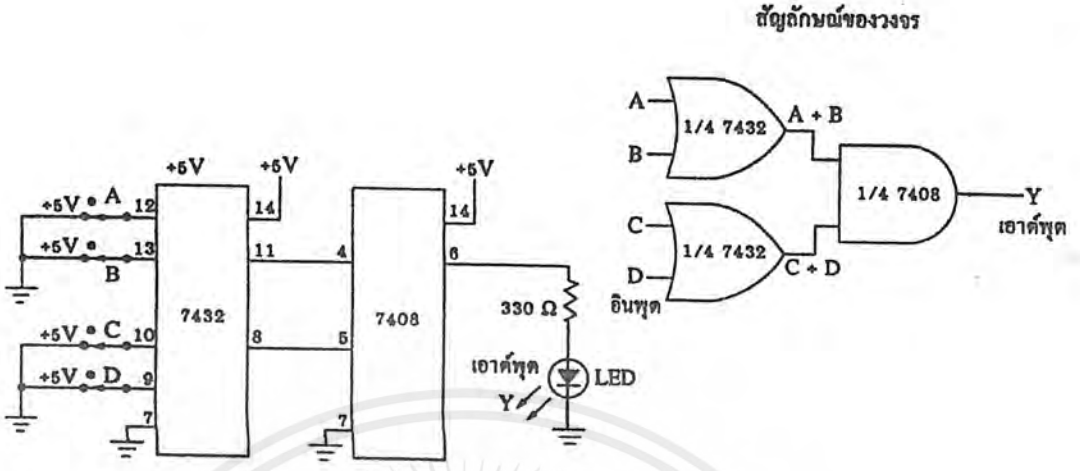



รูปที่ 2.36 Logic Gate Schematic ของ (A + B)C = Y

**ตัวอย่างที่ 4**


รูปที่ 2.37 สมการบูลลีนของ (A +B)(C +D) = Y

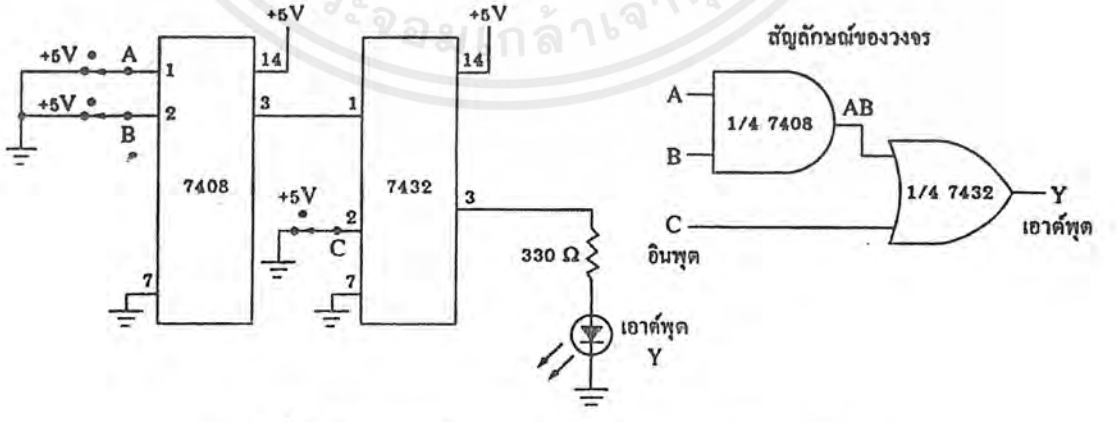
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้




รูปที่ 2.38 Logic Gate Schematic ของ (A + B)(C + D) = Y

ตัวอย่างที่ 5


รูปที่ 2.39 สมการบูลลีน : (AB) + C = Y

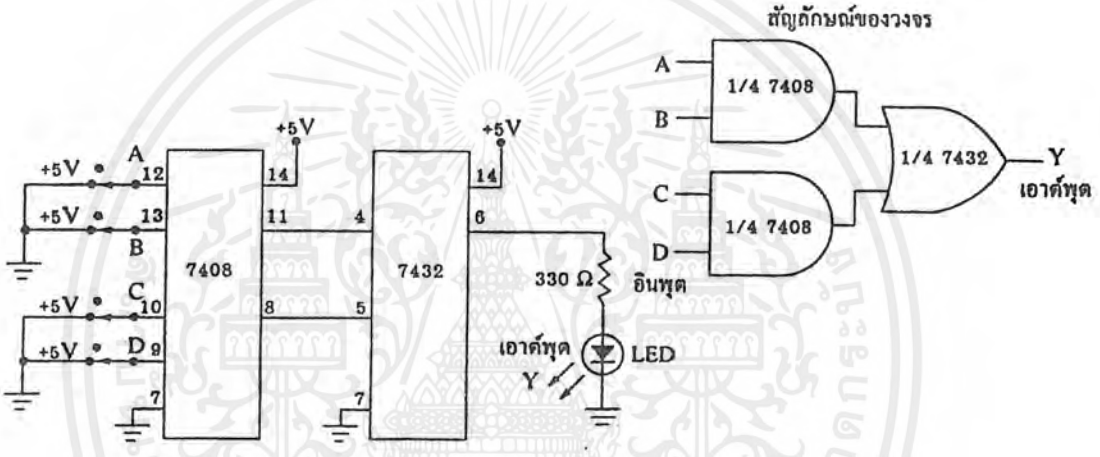



รูปที่ 2.40 Logic Gate Schematic ของ (AB) + C = Y

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวอย่างที่ 6


รูปที่ 2.41 สมการบูลลีน : (AB) + (CD) = Y

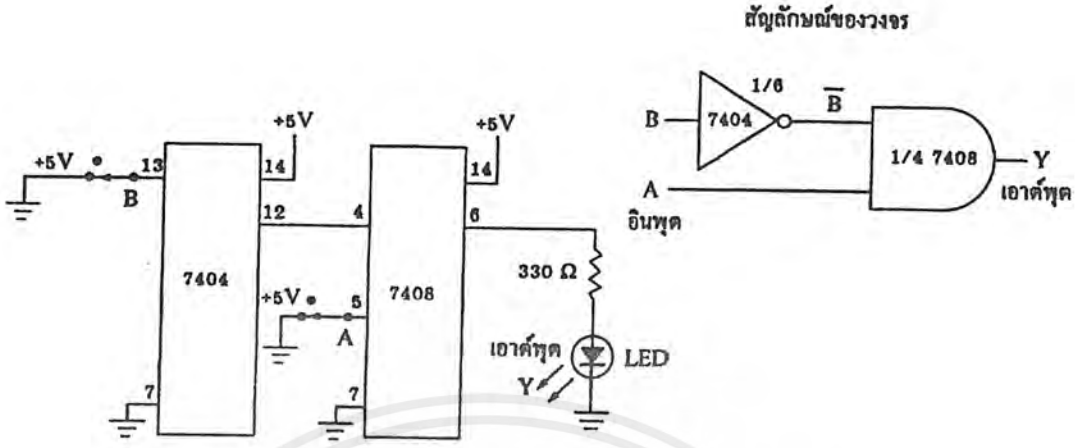



รูปที่ 2.42 Logic Gate Schematic ของ (AB) + (CD) = Y

ตัวอย่างที่ 7


รูปที่ 2.43 สมการบูลลีน : AB = Y

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

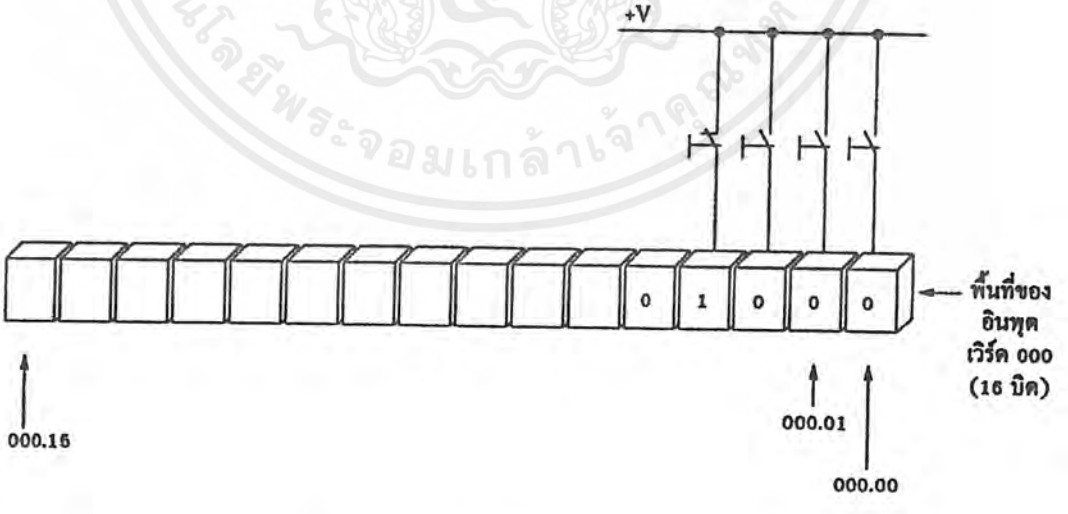



รูปที่ 2.44 Logic Gate Schematic ของ AB = Y

## การต่อสายควบคุม PLC

เมื่อทราบหลักการทำงานเกี่ยวกับเลขฐานและลอจิกต่างๆ ซึ่งเป็นพื้นฐานใน PLC แล้ว ต่อไปนี้จะได้กล่าวถึงการทำงานของ PLC ที่ต้องต่อสายควบคุมให้อุปกรณ์ทำงานตามต้องการ โดยจะเริ่มต้นจากหลักการของอินพุท เอาท์พุทและวงจรควบคุม

### การทำงานของภาคอินพุท

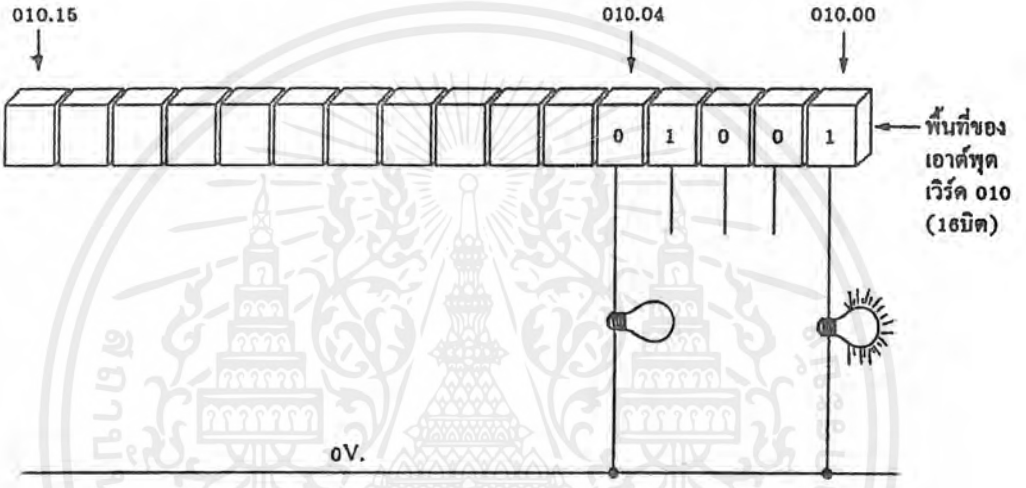



รูปที่ 2.45 การทำงานภาคอินพุท (Input)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปข้างบนแสดงให้เห็นว่า ถ้ามีสัญญาณไฟฟ้าจ่ายเข้ามาที่ภาคอินพุทจะทำให้ข้อมูลของ Input Area (พื้นที่ของอินพุท) ที่บิตนั้นเป็น "1" แต่ถ้าไม่มีสัญญาณไฟฟ้าจ่ายเข้ามาที่ภาคอินพุทจะทำให้ข้อมูลของ Input Area ที่บิตนั้นเป็น "0"

**การทำงานของภาคเอาท์พุท**



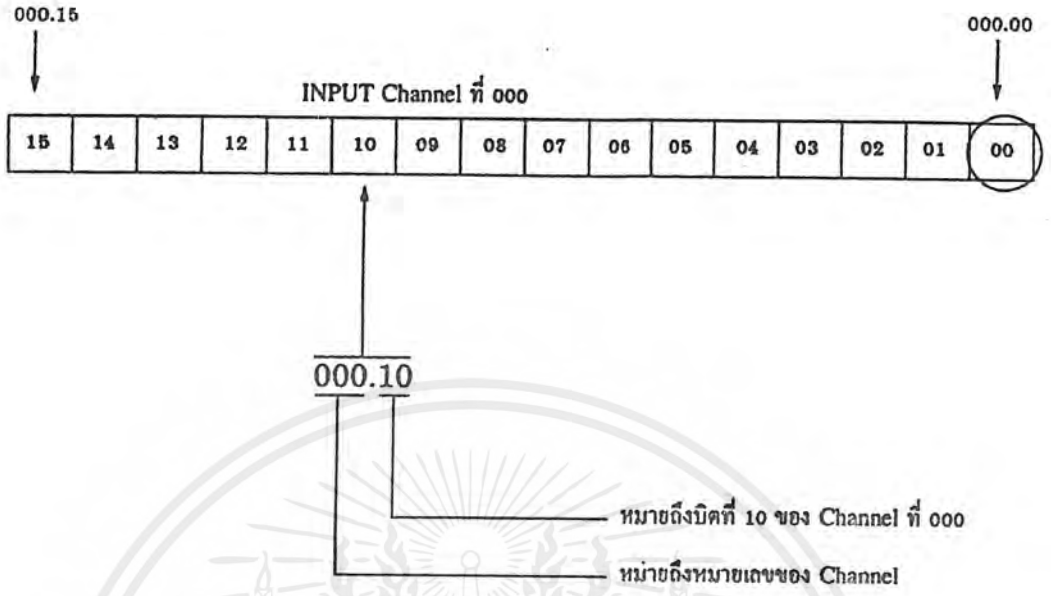

รูปที่ 2.46 การทำงานของภาคเอาท์พุท

สถานะข้อมูลของ Output Area (พื้นที่ของภาคเอาท์พุท) จะเป็น "1" หรือ "0" ขึ้นอยู่กับโปรแกรมใน PLC โดยจะใช้ผลของโปรแกรมหลังสุดเป็นหลัก

**การกำหนดเบอร์ของรีเลย์ใน CPM1 (OMRON)**

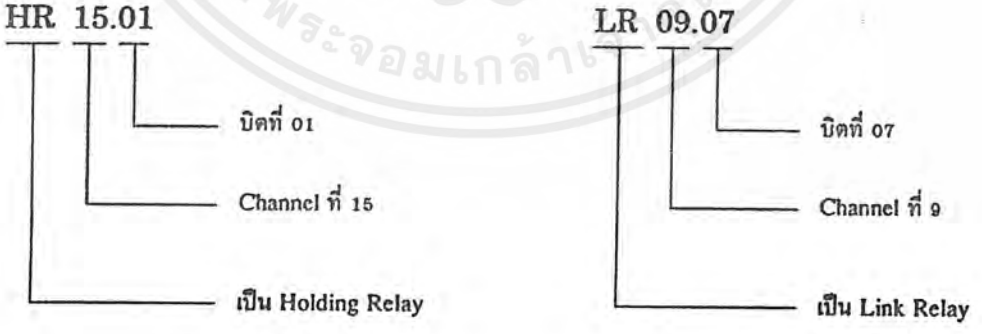
โดยปกติแล้ว PLC ของ OMRON จะกำหนดพื้นที่รีเลย์ (Relay) เป็นเวิร์ด (Word) หรือ Channel (แชนแนล) ซึ่งแต่ละ Channel จะประกอบด้วย 16 บิต ดังตัวอย่างในรูปที่ 2.57 คือ Channel ที่ 000

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้




รูปที่ 2.47 การกำหนดเบอร์ รีเลย์

ในกรณีของเอาท์พุทก็เช่นกัน แต่เริ่มที่ Channel ที่ 010 จนถึง 019 นอกจากอินพุทและเอาท์พุทแล้วยังมีรีเลย์ชนิดอื่นๆ อีกที่มีการกำหนดเบอร์เหมือนกัน เช่น Holding Relay , Link Relay ดังตัวอย่างต่อไปนี้




รูปที่ 2.48 การกำหนดของ Holding และ Link Relay

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

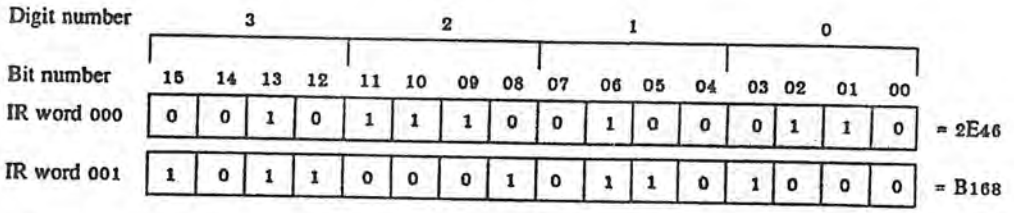
# พื้นที่ความจำ (Memory Area) ใน CQM1

ตารางที่ 2.5 พื้นที่หน่วยความจำของ CPM1

| พื้นที่ของข้อมูล | | เวิร์ด | บิต | หน้าที่ |
|---|---|---|---|---|
| IR area[1] | Input area | IR 000 to IR 009 (10 words) | IR 00000 to IR 00915 (160 bits) | บิตเหล่านี้ใช้สำหรับต่อกับ อินพุต/เอาต์พุตภายนอก |
| | Output area | IR 010 to IR 019 (10 words) | IR 01000 to IR 01915 (160 bits) | |
| | Work area | IR 200 to IR 231 (32 words) | IR 20000 to IR 23115 (512 bits) | เวิร์กบิตเป็นบิตอิสระที่ใช้ในโปรแกรม |
| SR area | | SR 232 to SR 255 (24 words) | SR 23200 to SR 25507 (384 bits) | บิตเหล่านี้ใช้สำหรับหน้าที่เฉพาะ เช่น flags และการควบคุมบิต |
| TR area | | --- | TR 0 to TR 7 (8 bits) | บิตเหล่านี้ใช้สำหรับเก็บสถานะ"ON" "OFF" ชั่วคราวที่สาขาของโปรแกรม |
| HR area[2] | | HR 00 to HR 19 (20 words) | HR 0000 to HR 1915 (320 bits) | บิตเหล่านี้ใช้สำหรับเก็บข้อมูล และรักษาสถานะ "ON" - "OFF" เมื่อไม่มีกระแสไฟฟ้า |
| AR area[3] | | AR 00 to AR 15 (16 words) | AR 0000 to AR 1515 (256 bits) | บิตเหล่านี้ใช้สำหรับหน้าที่เฉพาะ เช่น flags และการควบคุมบิต |
| LR area[1] | | LR 00 to LR 15 (16 words) | LR 0000 to LR 1515 (256 bits) | ใช้สำหรับเชื่อมต่อข้อมูล 1:1 กับ PLC ตัวอื่นๆ |
| Timer/Counter area[5] | | TC 000 to TC 127 (timer/counter numbers)[3] | | หมายเลขเดียวกันใช้สำหรับไทเมอร์และเคาน์เตอร์ |
| DM area | Read/write[7] | DM 0000 to DM 0999 DM 1022 to DM 1023 (1,002 words) | --- | ข้อมูลของพื้นที่ DM สามารถเข้าถึงหน่วยของเวิร์ดเท่านั้น และค่าของเวิร์ดยังคงมีอยู่เมื่อไม่มีกระแสไฟฟ้า |
| | Error log[4] | DM 1000 to DM 1021 (22 words) | --- | ใช้สำหรับเก็บเวลาที่เกิดขึ้นและค่าผิดพลาดที่เกิดขึ้น เวิร์ดเหล่านี้สามารถใช้เป็น read/write DM เมื่อ log function ผิดพลาดจะไม่ใช้ |
| | Read-only[4] | DM 6144 to DM 6599 (456 words) | --- | ไม่สามารถเขียนมากเกินกว่าจากโปรแกรม |
| | PC Setup[4] | DM 6600 to DM 6655 (56 words) | --- | ใช้สำหรับเก็บพารามิเตอร์ต่าง ๆ ที่ควบคุมการทำงานของ PLC |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## โครงสร้างของข้อมูล

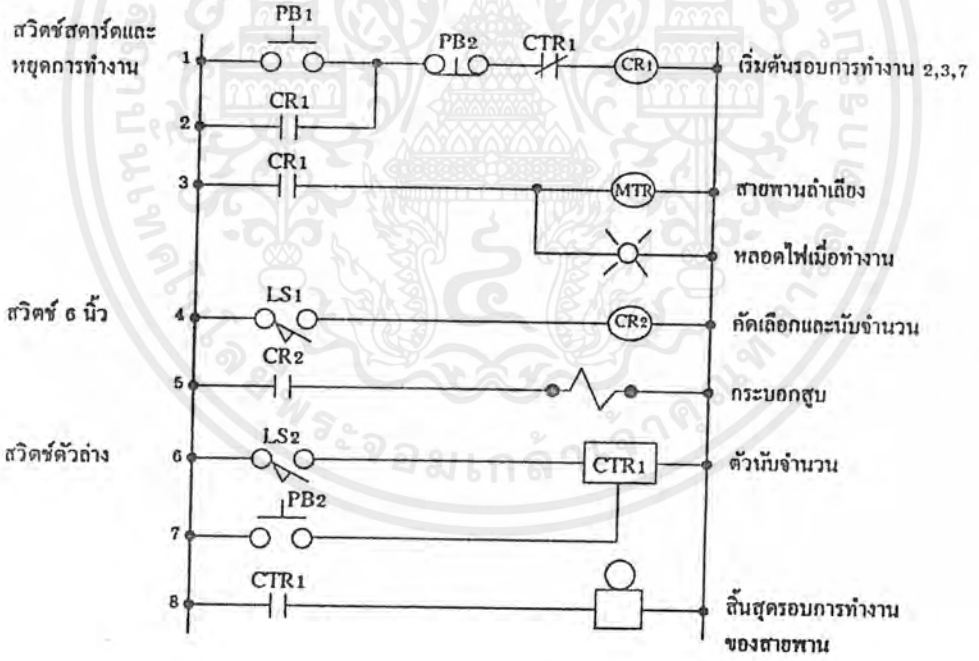



รูปที่ 2.49 โครงสร้างของข้อมูล

## เปรียบเทียบการต่อสายระหว่างวงจรใช้รีเลย์กับ PLC

### ตัวอย่างที่ 1

*1. วงจรใช้รีเลย์*

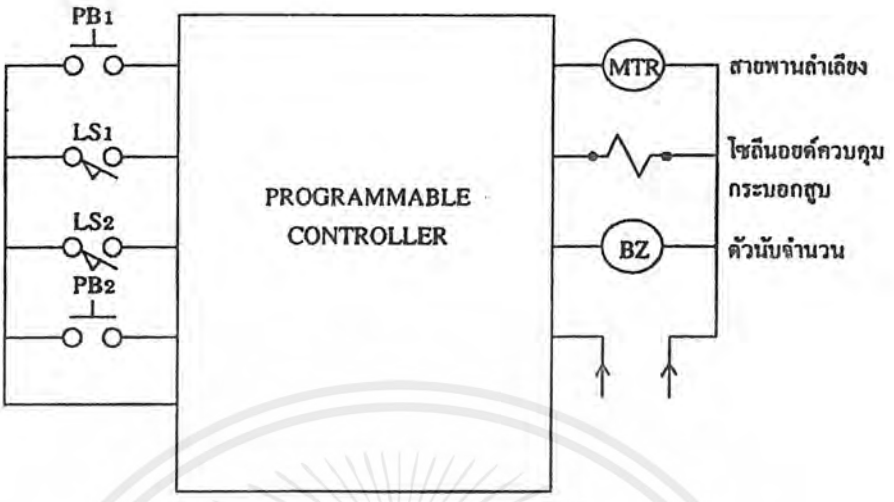



รูปที่ 2.50 การต่อสายไฟฟ้าควบคุมในวงจรใช้รีเลย์

*2. วงจรใช้ PLC*

จากรูปที่ 2.60 จะเห็นว่า ระบบการเดินสายไฟในวงจรที่ใช้รีเลย์จะยุ่งยากมาก เมื่อเปรียบเทียบกับการใช้ PLC ในรูปที่ 2.61

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

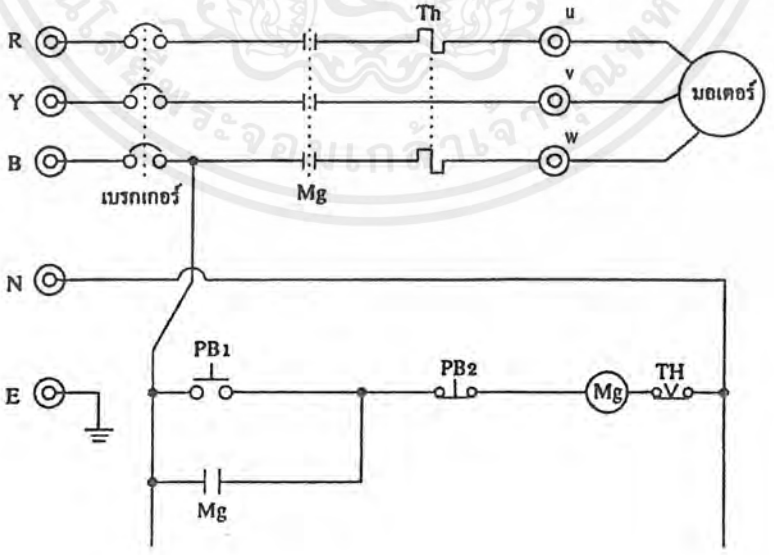



รูปที่ 2.51 การต่อสายไฟฟ้าควบคุมในวงจรใช้ PLC

**ตัวอย่างที่ 2**

*1. วงจรใช้รีเลย์*

กำหนดให้ PB1 เป็นปุ่มสตาร์ตมอเตอร์ และมอเตอร์ยังคงทำงานอยู่หลังจากปล่อยมือจากการกดปุ่มสตาร์ตแล้วก็ตาม จนกว่าจะกด PB2 มอเตอร์ถึงจะหยุดการทำงาน

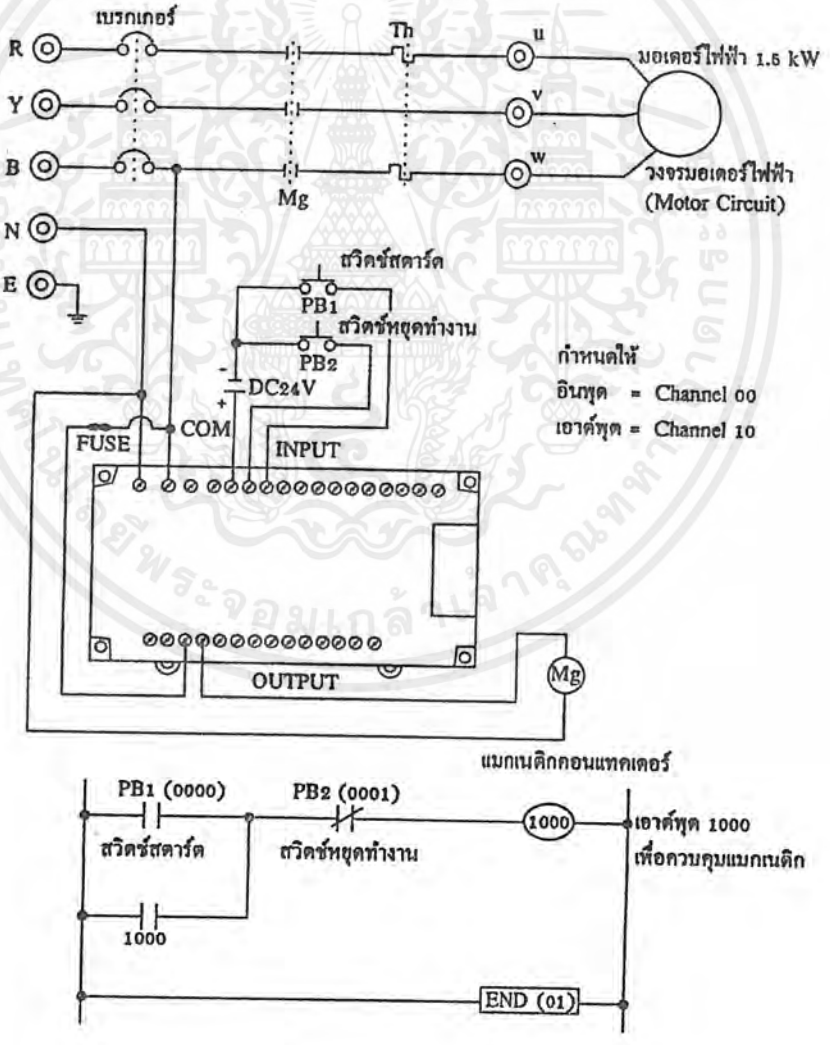



รูปที่ 2.52 รูปแบบการเดินสายไฟของระบบซีเควัน (Sequence)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. *วงจรใช้ PLC*

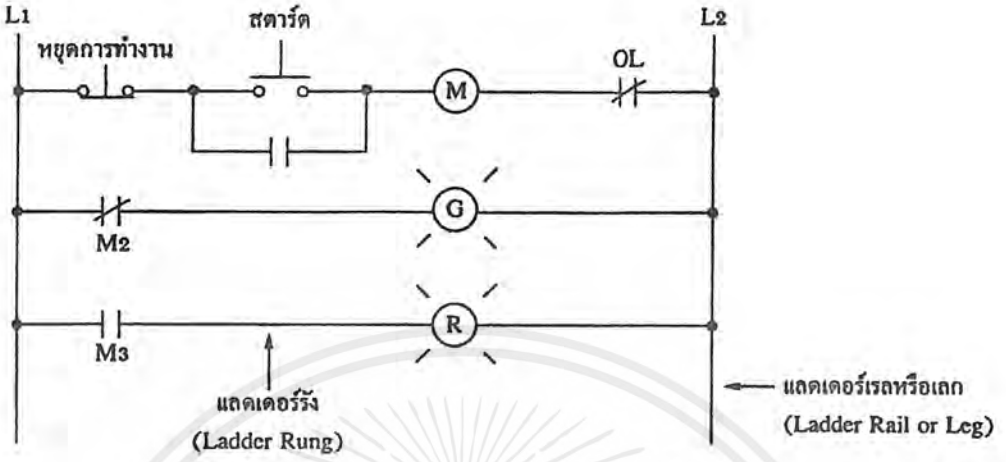
จะเห็นได้ว่าการเดินสายไฟ (Wiring) ในรูปที่ 2.62 เส้นไฟ (Power Line) เข้า PB1 และออกจาก PB1 ไปเข้าที่ PB2 และออกจาก PB2 ไปเข้า Magnetic Contactor ออกจากนี้ไปเข้า Overload และออกจาก Overload ถึงจะไปเข้า Common Line และต่อสายจากหน้าคอนแทคแบบปกติเปิด (ON) ของ Mg. (Magnetic) ไป ทำให้ Mg. ทำงานค้างไว้ (Hold) จะเห็นได้ว่าเป็นการเดินสายไฟที่สลับซับซ้อนหลายขั้นตอนมาก แต่ถ้าใช้ PLC แล้ว การเดินสายไฟเพียงจ่าย Power Supply ให้ PLC โดยที่ PB1 และ PB2 ต่อเข้าที่อินพุทเทอร์มินัล (Input Terminal) 1,2 ส่วนเอาท์พุทต่อจากสายเอาท์พุทเทอร์มินัลเข้าไปยัง Mg. (Magnetic Contactor) จากนั้นก็เขียนโปรแกรมใส่วงใน CPU ตามรูปที่ 2.63




รูปที่ 2.53 วงจรควบคุมของระบบที่ใช้รีเลย์ ที่เปลี่ยนมาใช้ PLC

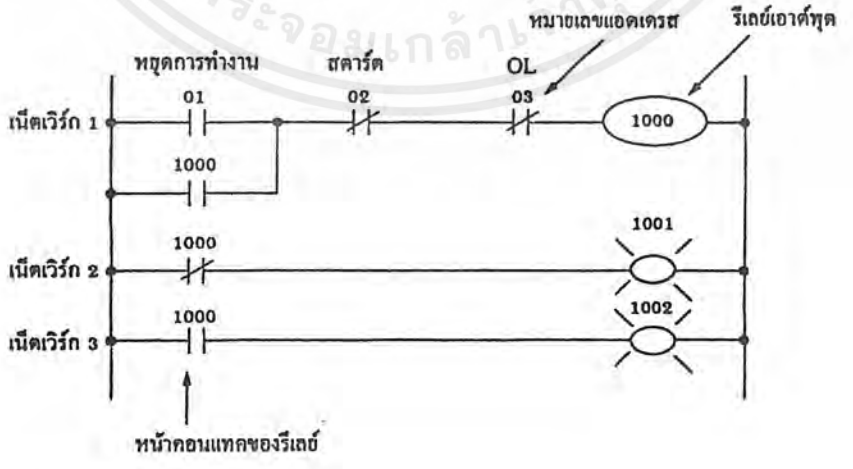
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Hard-Wired ลอจิก กับ Programmed ลอจิก




รูปที่ 2.54 วงจรรีเลย์

Hard-Wired Logic ใช้รีเลย์และวงจรแลดเดอร์ ซึ่งเป็นที่คุ้นเคยในโรงงานอุตสาหกรรมมาก่อน จากรูปที่ 2.64 แสดงวงจรรีเลย์ควบคุมการปิด-เปิดมอเตอร์ และมีหลอดไฟแสดงการทำงาน การเขียนวงจรควบคุมจะอยู่ระหว่างเส้นจ่ายไฟในแนวตั้ง 2 เส้น เรียกว่า Rails หรือ Legs เมื่ออุปกรณ์ภายในวงจรทำงาน (ON-OFF) ก็จะส่งข้อมูลเข้าไปยัง PLC ซึ่งเรียกว่า Language (ภาษา) ภาษาของ PLC ที่คุ้นเคยคือลอจิกแลเดอร์ไดอะแกรม (Logic Ladder Diagram) ต่อไปนี้จะเป็นโปรแกรมสำหรับรีเลย์แลดเดอร์ไดอะแกรมของรูป 2.64

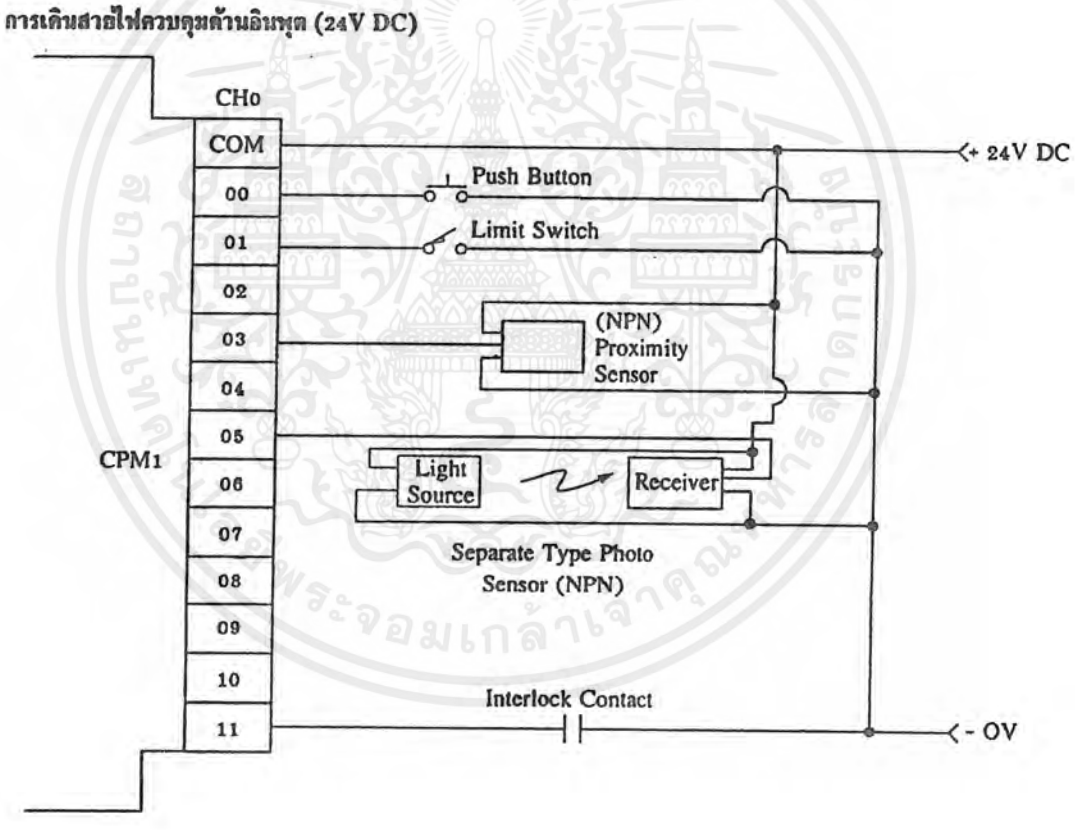



รูปที่ 2.55 วงจรลอจิกแลดเดอร์ (Logic Ladder Diagram)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

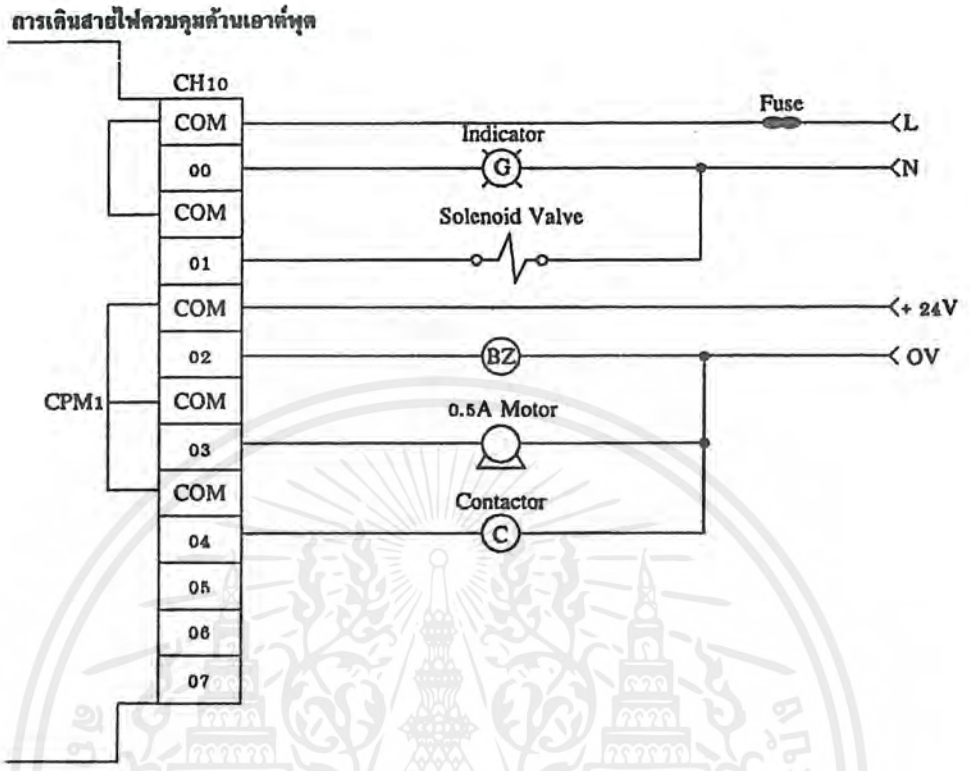
เน็ตเวิร์ก (Network) คือเส้นควบคุมเอาท์พุท PLC บางตัวอนุญาตให้มีเอาท์พุทได้หลายตัวใน 1 เน็ตเวิร์ก แต่ที่ดีที่สุดน่าจะมี 1 เอาท์พุทต่อ 1 เน็ตเวิร์ก ฉะนั้นโปรแกรมหนึ่งๆ จึงมีหลายเน็ตเวิร์กด้วยกัน

**ตัวอย่างการเดินสายไฟฟ้าควบคุมสัญญาณอินพุท/เอาท์พุทของPLC (OMRON,CQM1)**




รูปที่ 2.56 ตัวอย่างการเดินสายไฟควบคุมสัญญาณอินพุท/เอาท์พุทของ OMRON (CQM1)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้




รูปที่ 2.56 (ต่อ)

**ขั้นตอนการใช้ PLC**

1. กำหนดขั้นตอนการทำงานของเครื่องจักร

2. กำหนดอินพุทและเอาท์พุทคือการกำหนด Address ของสวิตช์กดปุ่ม (Push Button) หรือแมกเนติก (Magnetic) ว่าอยู่ในแชนแนลที่เท่าใด เช่น สวิตช์กดปุ่ม (Push Button) จะต่อเข้าที่ขั้วต่อสาย (Terminal) 1 ก็คือ Bit 00 เป็นต้น

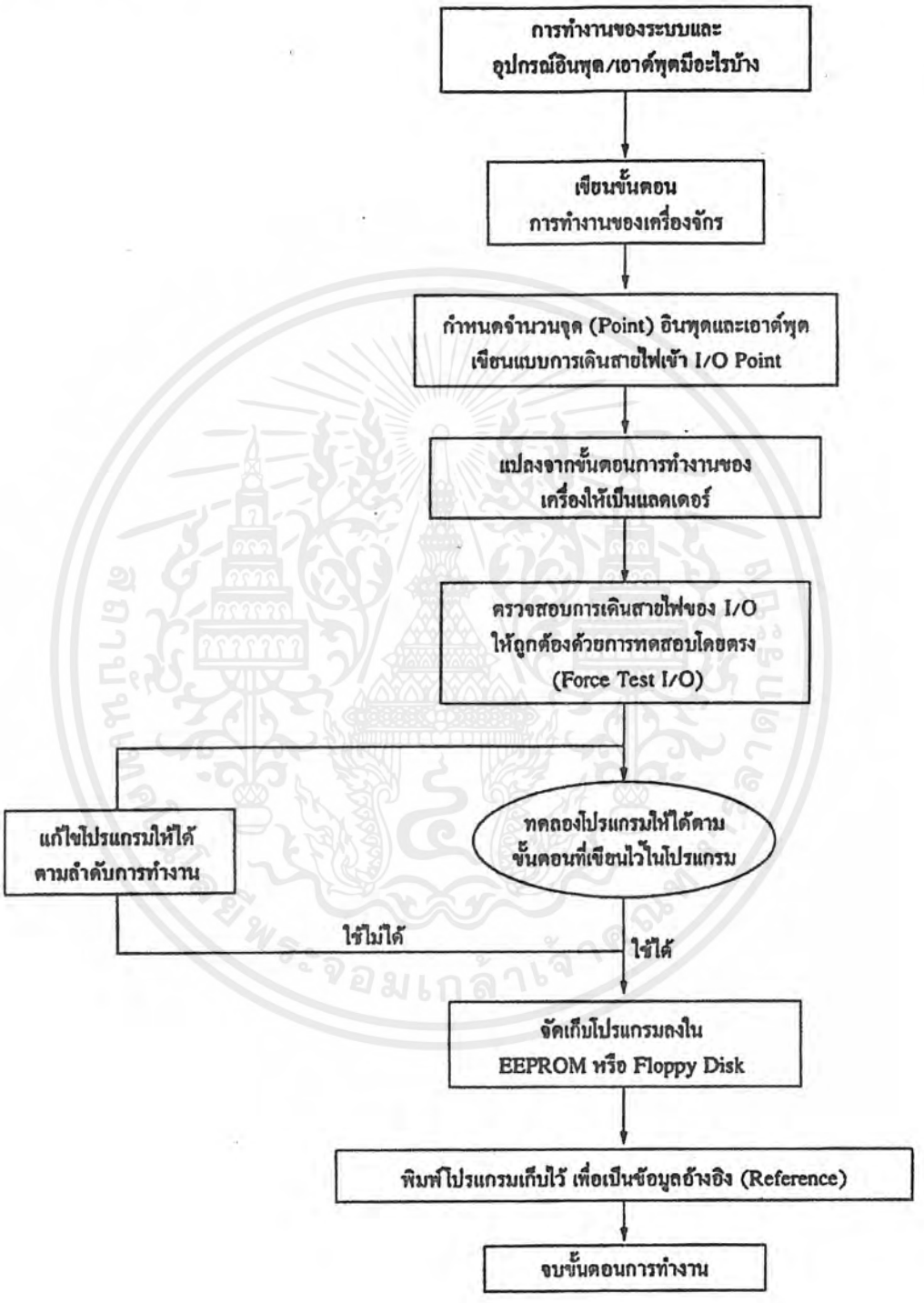
3. เดินสายไฟจากอินพุทเข้าที่ขั้วต่อสายด้านอินพุท (Input Terminal) และจากขั้วต่อสายด้านเอาท์พุท (Output Terminal) เข้าที่โหลด (Load) หรือรีเลย์ (Buffer)

4. เขียนโปรแกรมลงใน CPU ของ PLC เขียนตามขั้นตอนการทำงานของเครื่อง อาจจะเป็นในรูปของนีมอนิก (Mnemonic) หรือแลดเดอร์ก็ได้

5. การให้ PLC ทำงานตามโปรแกรม และการมอนิเตอร์ (Monitor) โปรแกรม หลังจากเขียนโปรแกรมจบแล้ว สั่ง (Run) คือให้เครื่องจักรทำงานตามขั้นตอนที่เขียนไว้ในโปรแกรมตามต้องการและดูสภาวะการทำงานที่หน้าจอ (Monitor)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แผนผังการใช้ PLC




รูปที่ 2.57 แผนผังการใช้ PLC

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# บทที่ 3
# การพัฒนาโปรแกรม

## 3.1 ลักษณะทั่วไป

การศึกษาและทำการออกแบบโปรแกรมที่จะจำลองการทำงานของระบบควบคุมต่างๆ โดยที่นำโปรแกรม PLC ซึ่งสามารถที่จะจำลองการทำงานของ PLC ได้อย่างสมบูรณ์ ขาดแต่เพียงการแสดงผล ซึ่งโปรแกรม PLC ตัวนี้ไม่สามารถแสดงผลการทำงานหรือสถานะปัจจุบันของอินพุท (INPUT) และ เอาพุท (OUTPUT) ของ PLC ได้ จึงทำให้ผู้ใช้ไม่สามารถจะทำความเข้าใจและติดตามการทำงานของระบบได้อย่างเต็มที่ นอกจากนั้นเมื่อผู้ใช้เขียนแลดเดอร์ไดอะแกรม(Ladder Diagram)ลงไปในโปรแกรมแล้วต้องการจะทดสอบการทำงานของโปรแกรมก็ยังไม่สามารถทดสอบได้เนื่องมาจากที่ได้กล่าวไว้ข้างต้น จะทำได้เพียงทดสอบการทำงานทีละอินพุท หรือเอาพุทเท่านั้น

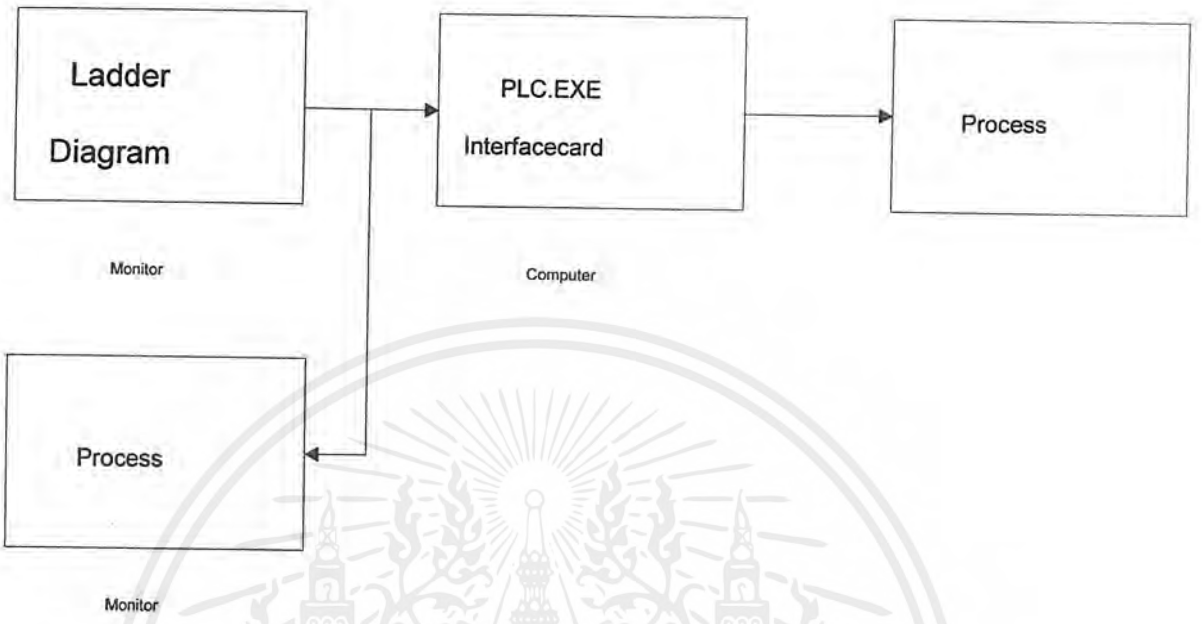
จากข้อมูลต่างๆข้างต้น ทำให้มีการพัฒนาโปรแกรมตัวใหม่เพิ่มขึ้นมาเพื่อที่จะสามารถทำให้โปรแกรม PLC นี้ มีการแสดงผลการทำงานเป็นรูปภาพ (Graphic) และเพื่อที่จะสามารถสื่อความหมายและทำความเข้าใจกับผู้ใช้ได้ดียิ่งขึ้น รูปภาพที่ว่านี้จะมีการแสดงสีที่แตกต่างกันเหมือนมีการติด/ดับของอุปกรณ์ และจะสามารถเคลื่อนไหวได้เหมือนการทำงานของระบบจริงด้วย

โปรแกรม PLC เดิมจะเป็นโปรแกรมที่จะจำลองการทำงานของ PLC บนเครื่องคอมพิวเตอร์ โดยสามารถเขียนแลดเดอร์ไดอะแกรมลงไปในโปรแกรม แล้วโปรแกรมจะทำการประมวลผลและทำงานตามคำสั่งที่เขียนมาในแลดเดอร์ไดอะแกรม นอกจากนี้ยังสามารถส่งค่าควบคุมการทำงานไปยังระบบจริงโดยผ่านการ์ดอินเตอร์เฟส โดยมีการทำงานดังรูป


รูปที่ 3.1 แสดงการทำงานของโปรแกรม PLC

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 3.1 ได้เปลี่ยนการทำงานของโปรแกรม PLC เสียใหม่ให้เป็นดังนี้



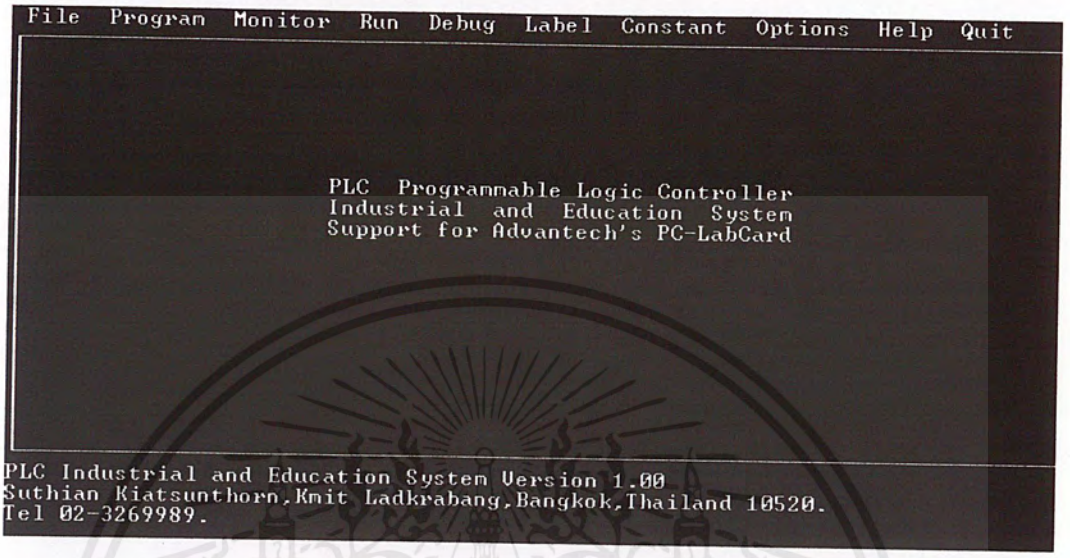

รูปที่ 3.2 แสดงการทำงานของโปรแกรม PLC ที่ได้มีการเปลี่ยนแปลงแล้ว

โดยจะเห็นว่ามีระบบอยู่บนมอนิเตอร์(Monitor) เพิ่มขึ้นมา คือ จำลองการทำงานของระบบจริงมาแสดงการทำงานอยู่บนคอมพิวเตอร์ โดยผ่านมอนิเตอร์มายังผู้ใช้ เพื่อที่จะสามารถเข้าใจและติดตามการทำงานได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

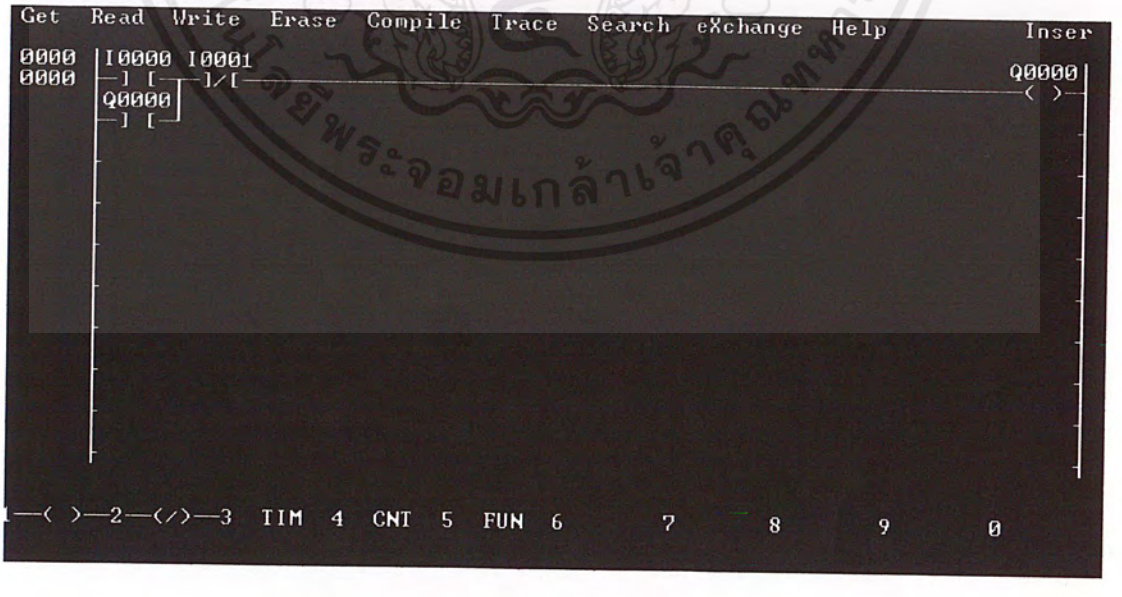
### 3.2 การออกแบบโปรแกรม

จากโปรแกรม PLC ตัวเดิมเมื่อมีการเรียกใช้งานจะได้ดังภาพ




รูปที่ 3.3 หน้าจอหลักโปรแกรม PLC

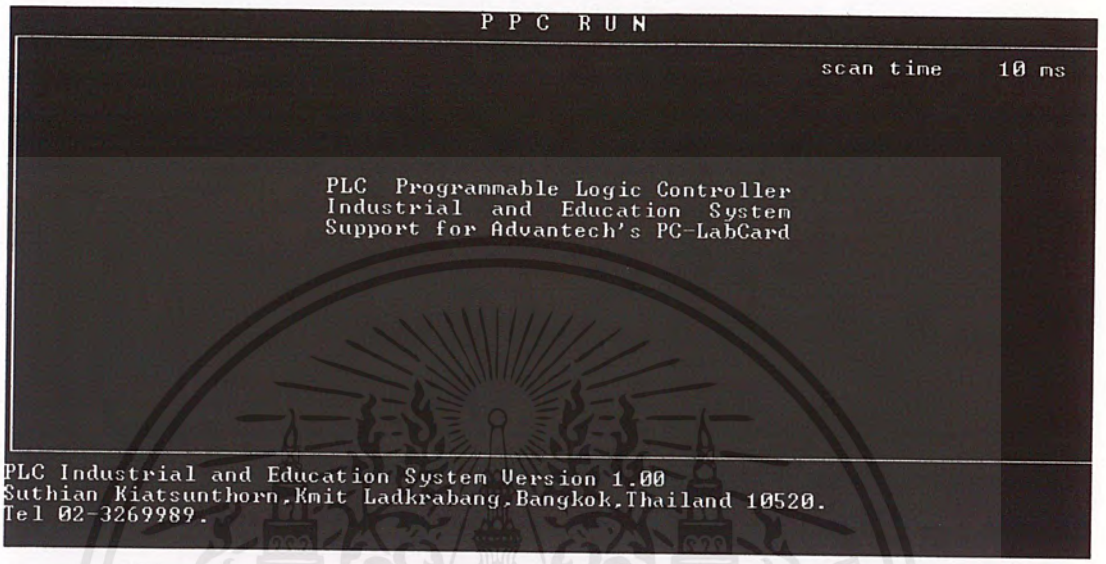
จะเห็นว่ามีคำสั่ง RUN อยู่ในโปรแกรมซึ่งเมื่อทำการกดคีย์ R จะทำการสั่งให้โปรแกรม PLC ทำงานโดยประมวลผลจากแลดเดอร์ไดอะแกรมในตัวโปรแกรมดังภาพ




รูปที่ 3.4 ตัวอย่างแลดเดอร์ไดอะแกรมที่เขียนในโปรแกรม PLC

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อกดคีย์ R จะทำการรันโปรแกรมโดยในโปรแกรมที่ยังไม่ได้เปลี่ยนแปลงใหม่จะมีหน้าจอดังภาพ




รูปที่ 3.5 หน้าจอ RUN ในโปรแกรม PLC ที่ยังไม่มีการเปลี่ยนแปลง

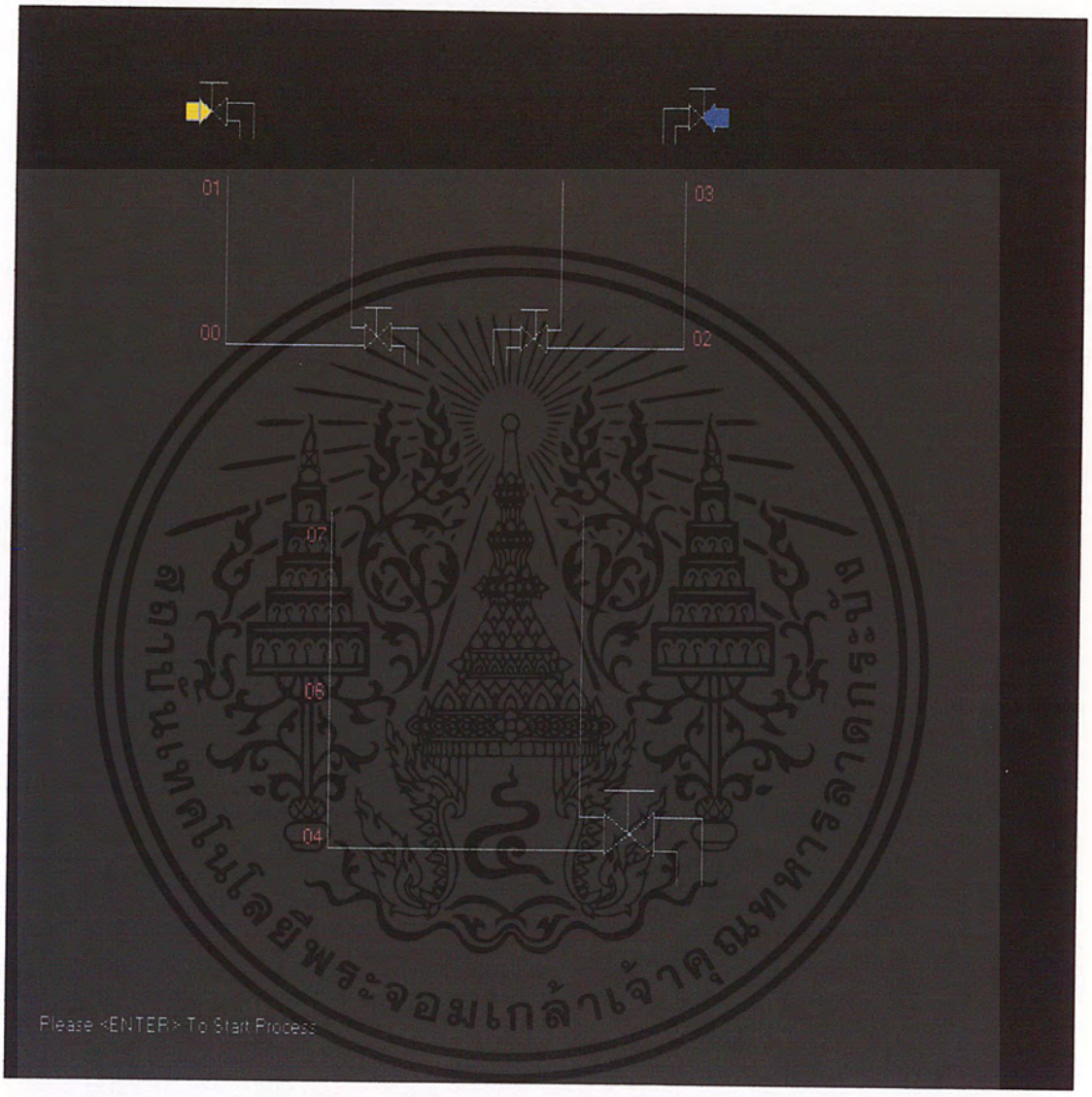

รูปที่ 3.6 เมนูในการเลือกระบบการจำลองการทำงานของระบบในโปรแกรม PLC

ซึ่งจะสังเกตเห็นว่าไม่มีการแสดงภาพการทำงานใดๆเลย จึงจะทำการเปลี่ยนให้เป็นการแสดงการจำลองการทำงานของระบบดังที่ได้กล่าวไว้ข้างต้น โดยได้ทำการออกแบบไว้ว่า เมื่อมีการสั่งรันจะให้มีเมนู (MENU) ขึ้นมาเพื่อให้ผู้ใช้สามารถเลือกระบบที่ต้องการจำลองมาไว้บนหน้าจอคอมพิวเตอร์ดังภาพ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากภาพจะเห็นว่ามีเมนูให้เลือกอยู่ 2 ระบบ คือ

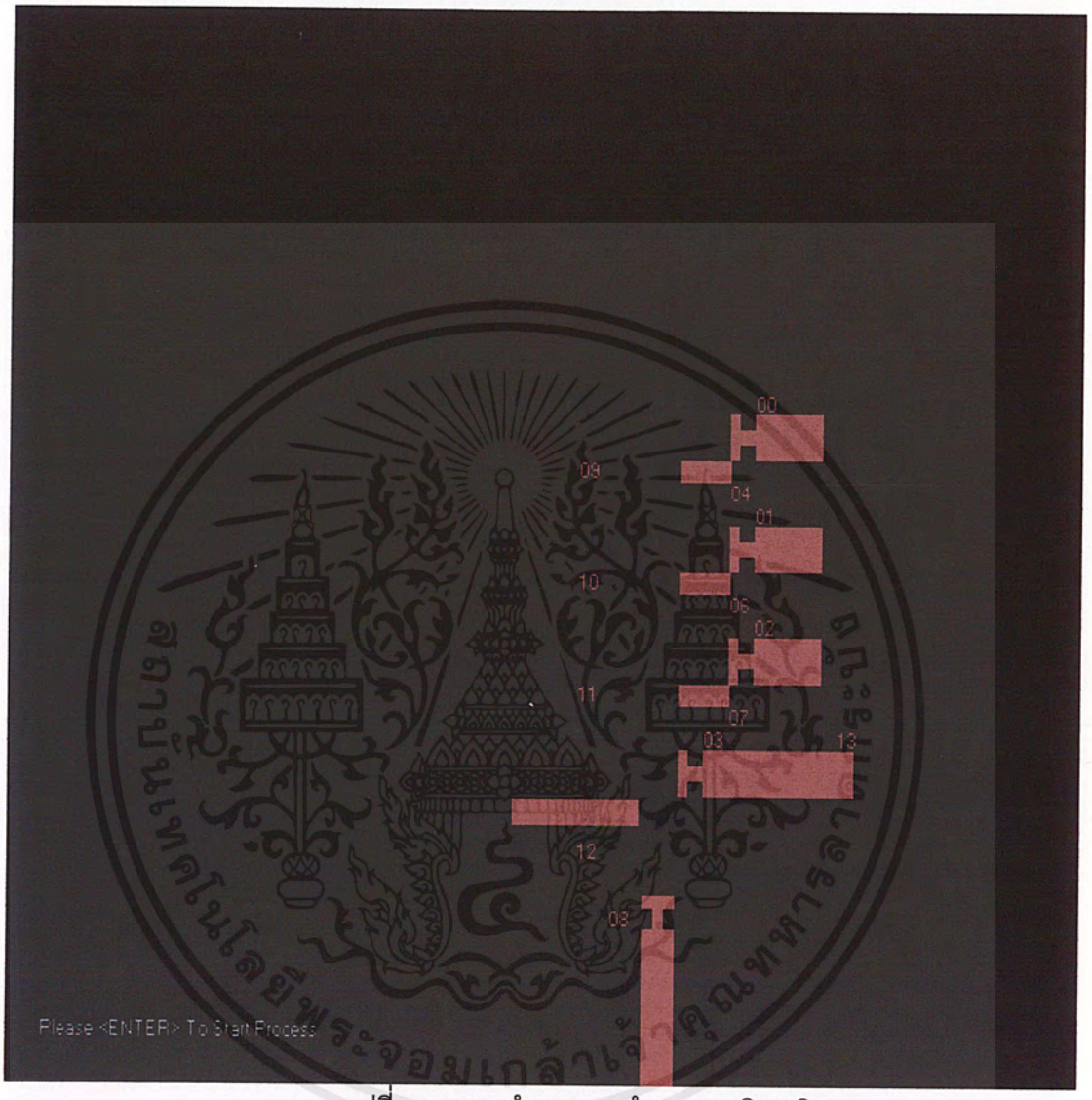
- ระบบที่เป็นถังน้ำ (TANK PROCESS)




รูปที่ 3.7 ระบบจำลองการทำงานของถังน้ำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

➢ ระบบที่เป็นนิวเมติก (PNEUMATIC PROCESS)




รูปที่ 3.8 ระบบจำลองการทำงานของนิวเมติก

ผู้ใช้สามารถเลือกได้โดยใช้ลูกศรเลื่อนขึ้นลงเหมือนเมนูทั่วๆไปแล้วกด<ENTER>เมื่อต้องการเริ่มการทำงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การแก้ไขโปรแกรมให้เป็นแบบนี้ทำโดยแก้ไขในโปรแกรมเดิมในไฟล์ PLC9.C โดยจากเดิมเมื่อสั่งรันหน้าจอจะเป็นดังรูป3.5ได้ทำการแก้ไขเพิ่มเติมโดยไม่เรียกหน้าจอเดิมขึ้นมาแต่จะเรียกโปรแกรมย่อยขึ้นมาแทน โดยโปรแกรมย่อยที่ว่าจะเป็นตัวจัดการเมนูต่างๆเพื่อที่จะสามารถเลือกระบบที่ต้องการจแสดงได้ ซึ่งโปรแกรมย่อยนี้จะอยู่ในไฟล์ CONTROL.C และจะมีระบบนิวเมติก อยู่ในไฟล์ PRO.C และ ระบบที่เป็นถังน้ำอยู่ในไฟล์ PRO2.C

โปรแกรมPLCเดิมซึ่งได้แก่PLC.C,PLC1.C,PLC2.C,PLC3.C,PLC4.C,PLC5.C,PLC6.C,PLC7.C, PLC8.C, PLC9.C กับโปรแกรมที่สร้างขึ้นใหม่ CONTROL.C, PRO.C, PRO2.C ในไฟล์ที่กล่าวมาทั้งหมดถูกเขียนโดยภาษาซี และใช้โปรแกรมไมโครซอฟต์ซีคอมไพเลอร์เวอชั่น 7 (Microsoft C Compliler Version 7) เป็นตัวคอมไพเลอร์ และยังมีไฟล์ PLCLIB.ASM, PLCLIB1.ASM, PLCLIB2.ASM, UTIL.ASM ซึ่งเป็นภาษาแอสแซมบลี้(Assembly) และใช้ ไมโครซอฟต์แอสแซมเบอร์ (Microsoft Assembler) เป็นตัวคอมไพเลอร์

นอกจากนี้ยังต้องมี Include files อีกด้วยคือ PLC.H และ PROJECT.H เพื่อเป็นตัวกำหนดค่าคงที่ต่างๆ โดย PROJECT.H จะเป็นไฟล์ที่เพิ่มเติมเข้ามาใหม่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.3 โครงสร้างโปรแกรม

โครงสร้างของโปรแกรมที่ออกแบบไว้มีดังนี้

- โปรแกรมทุกตัวต้องเป็นอิสระต่อกัน โดยสามารถที่จะเรียกใช้ได้จากทุกโปรแกรม โดยไม่มีเงื่อนไขใดๆในการเรียกใช้
- ต้องไม่ไปรบกวนการทำงานของโปรแกรม PLC เดิม ซึ่งเป็นส่วนในการประมวลผลแลดเดอร์ไดอะแกรมและส่งค่าผ่านอินเตอร์เฟสการ์ดออกไปควบคุมระบบ

โปรแกรมที่แสดงภาพที่สร้างขึ้นประกอบด้วย โปรแกรมย่อย 3 โปรแกรมดังนี้ คือ

- โปรแกรมย่อย CONTROL.C จะมีหน้าที่สร้างเมนูให้เลือกระบบที่จะนำมาจำลองบนหน้าจอคอมพิวเตอร์ และรับค่าจากการเลือกเมนูของผู้ใช้เพื่อที่จะนำไปเรียกโปรแกรมย่อยที่จะแสดงภาพได้ และในโปรแกรมมีซับรูทีนดังต่อไปนี้
  - Animatoin_menu()
  - Control_menu()

- โปรแกรมย่อย PRO.C จะเป็นโปรแกรมแสดงภาพเคลื่อนไหวของระบบนิวเมติก และทำหน้าที่ตรวจสอบและส่งค่าอินพุท เอาพุท ไปยังโปรแกรม PLC เพื่อใช้ในการประเมินผลแลดเดอร์ไดอะแกรม และในโปรมแกรมมีซับรูทีนต่อไปนี้
  - PneAnimation()
  - Initscreen1()
  - RunAnimation()
  - Shelft()
  - CheckSensor()
  - UpDateTemp()
  - Pne1_On()
  - Stuff1Move()
  - Pne1_Off()
  - Pne2_On()
  - Stuff2Move()
  - Pne2_Off()
  - Pne3_On()

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

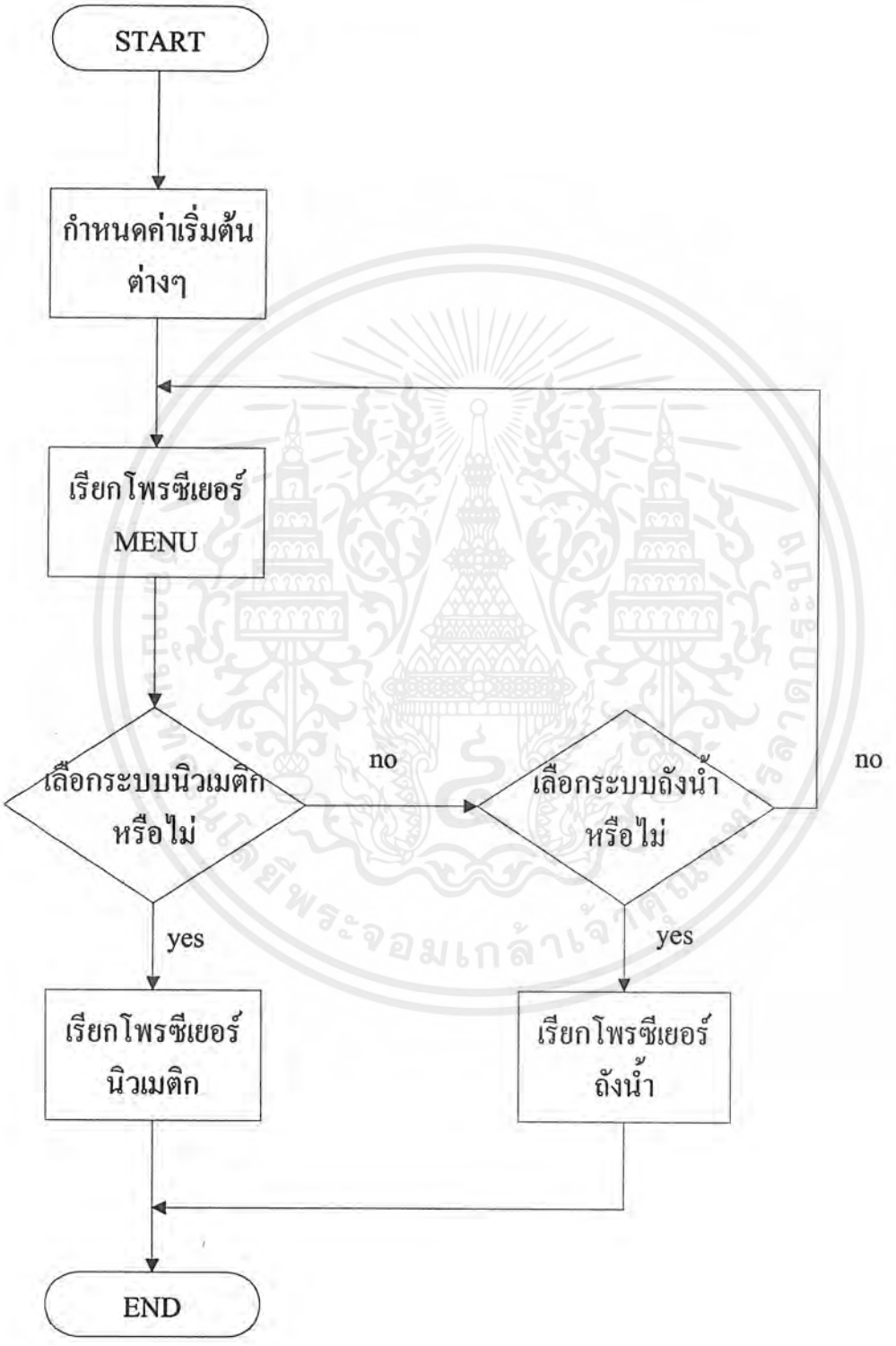
- Stuff3Move()
- Pne3_Off()
- Pne4_On()
- Stuff4Move()
- Pne4_Off()
- Pne5_On()
- Pne5_Off()
- Stuff5Move()
- UpDatePne()

- โปรแกรมย่อย Pro2.c จะเป็นโปรแกรมแสดงภาพเคลื่อนไหวของระบบถังน้ำ ทำหน้าที่ตรวจสอบและส่งค่าของอินพุท เอาพุทไปยังโปรแกรม PLC เพื่อใช้ในการประมวลผลแลดเดอร์ไดอะแกรม และในโปรมแกรมนี้มีซับรูทีนดังต่อไปนี้
  - TankAnimation()
  - Init()
  - RunAnimationTank()
  - CheckSensorTank()
  - CheckTankLeft()
  - CheckTankRight()
  - CheckTank()
  - CheckUpLeft()
  - CheckUpRight()
  - CheckLeft()
  - CheckRight()
  - CheckDown()
  - UpDateTank()

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

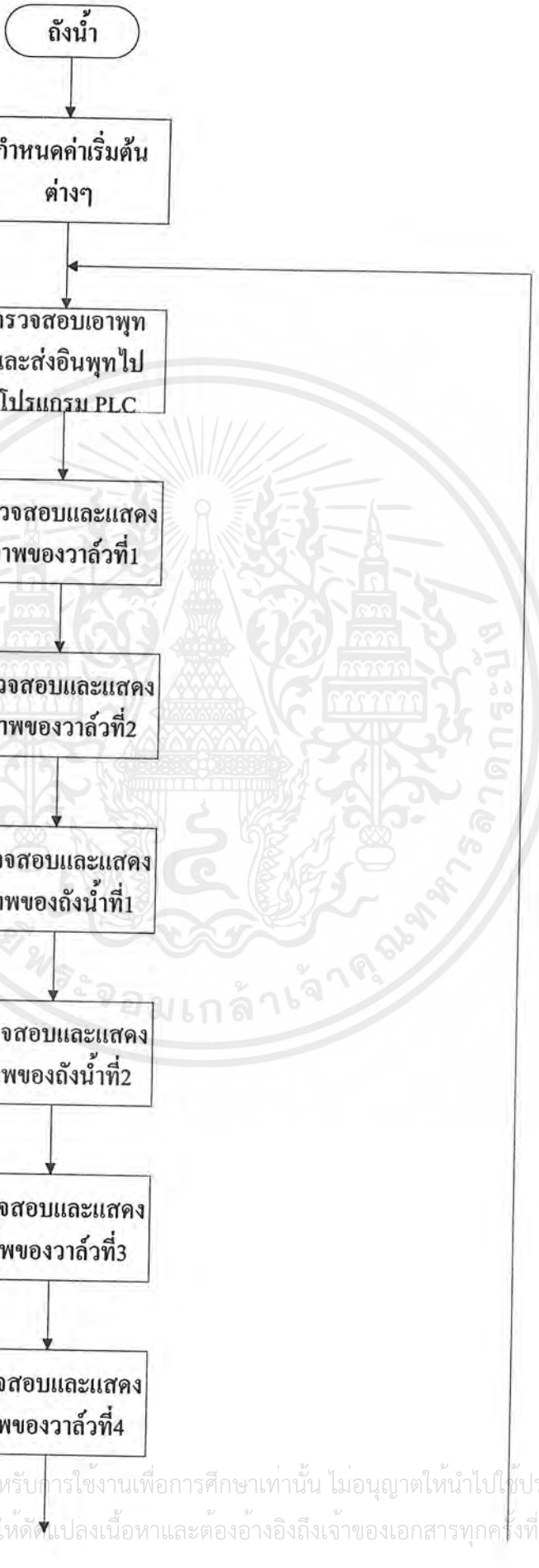
### 3.4 การทำงานของโปรแกรม

การทำงานของโปรแกรมจำลองระบบเป็นไปตามโฟล์วชาร์ทดังต่อไปนี้

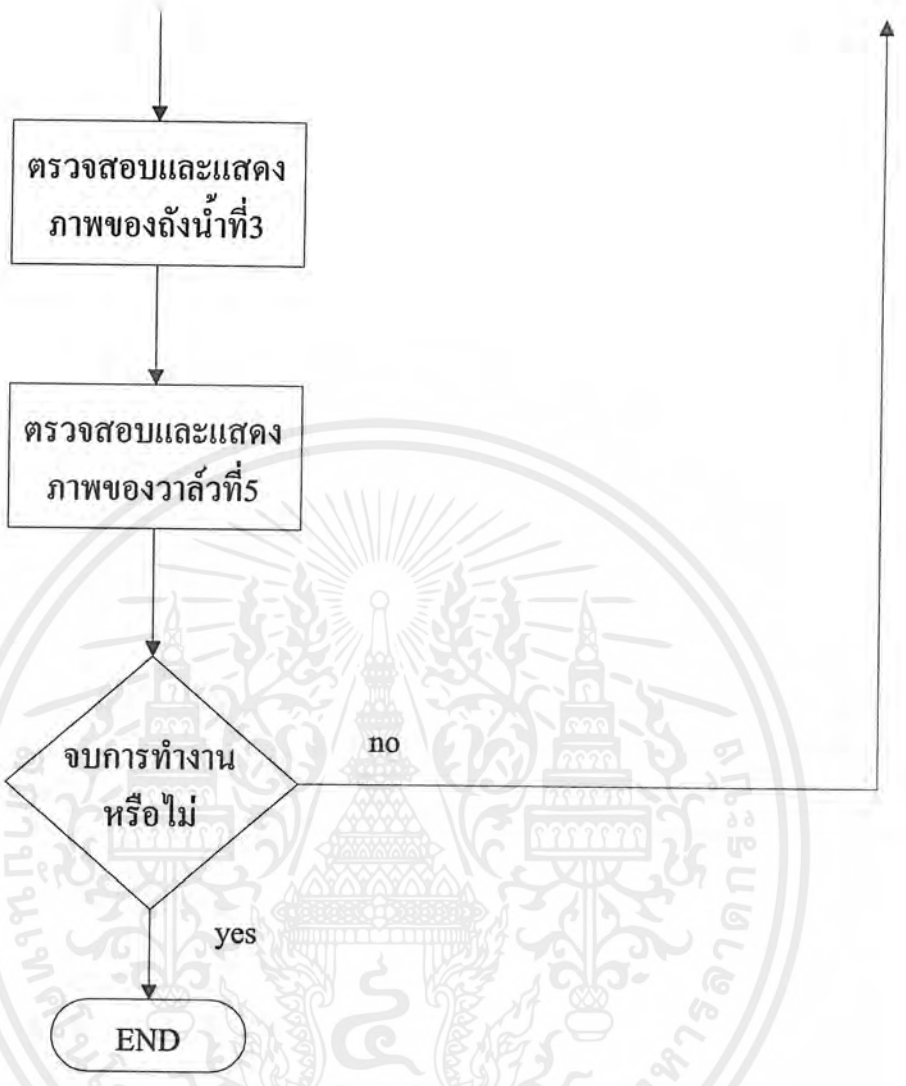



รูป 3.9 โพล์วชาร์ทแสดงการทำงานของโปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

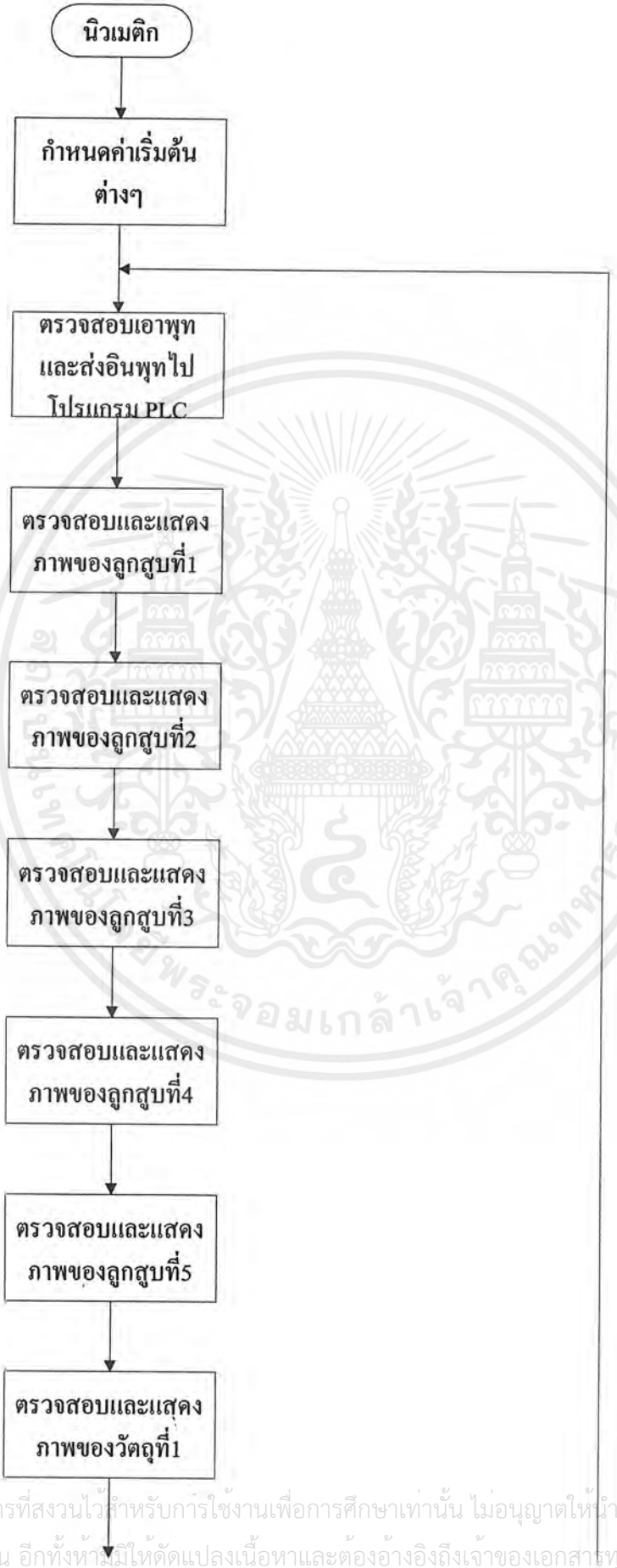



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

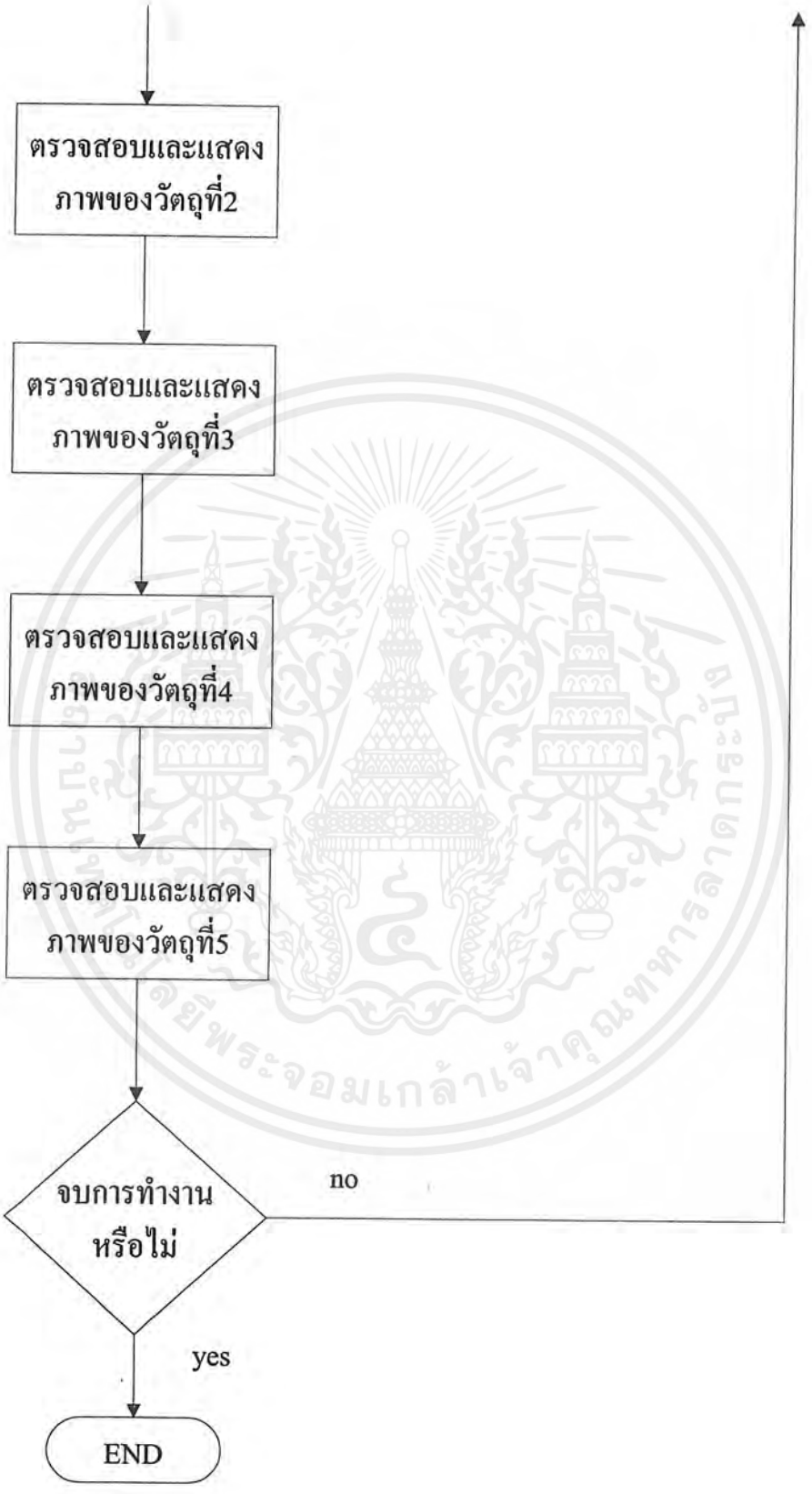



รูปที่ 3.10 โฟล์วชาร์ทแสดงการทำงานของโพรซีเยอร์ถังน้ำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้




เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้




**รูปที่ 3.11 โฟล์วชาร์ทแสดงการทำงานของโพรซีเยอร์นิวเมติก**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.5 การอ้างอิงอินพุท และเอาพุทของระบบ

จากระบบ 2 ระบบที่กล่าวถึงในข้างต้นผู้ใช้จะต้องรู้การอ้างอิงอินพุท และเอาพุทของระบบ เพื่อที่จะนำไปเขียนแลดเดอร์ไดอะแกรมของระบบนั้นได้อย่างถูกต้อง การอ้างอิงของอินพุทจะใช้ตัวอักษร "I" และเอาพุทจะใช้ตัวอักษร "Q" โดยมีการอ้างอิงดังนี้

- ระบบถังน้ำ(Tank Process)


รูปที่ 3.12 ระบบจำลองการทำงานของถังน้ำ

จากภาพจะเห็นตัวเลขอยู่ตามตำแหน่งต่างๆของถัง โดยจะมีตัวเลขตั้งแต่ 00 – 07 โดยตัวเลขแต่ละตัวแสดงถึงตำแหน่งของเซ็นเซอร์(SENSOR) ที่ติดตั้งในระบบจริง เพื่อใช้เป็นอินพุทของ PLC เพื่อใช้ในการประมวลผล โดยมีรายละเอียดแต่ละตัวดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- **ตัวเลข** 00 (I00) เป็นเซ็นเซอร์จับตำแหน่งของของเหลวที่อยู่ในถังที่ 1 โดยเซ็นเซอร์จะติดเมื่อระดับของเหลวอยู่ที่ตำแหน่งก้นถังหรือไม่มีของเหลวอยู่ในถังนั้นเอง และจะดับก็ต่อเมื่อระดับของของเหลวอยู่เหนือเซ็นเซอร์หรืออีกนัยหนึ่งคือมีของเหลวอยู่ในถังนั้นเอง
- **ตัวเลข** 01 (I01) เป็นเซ็นเซอร์จับตำแหน่งของของเหลวที่อยู่ในถังที่ 1 โดยเซ็นเซอร์จะติดเมื่อระดับของเหลวอยู่ที่ตำแหน่งขอบถังหรือมีของเหลวอยู่เต็มถังนั้นเอง และจะดับก็ต่อเมื่อระดับของของเหลวอยู่ต่ำกว่าเซ็นเซอร์หรืออีกนัยหนึ่งคือมีของเหลวอยู่ในถังแต่มีไม่ถึงขอบถัง
- **ตัวเลข** 02 (I02) เป็นเซ็นเซอร์จับตำแหน่งของของเหลวที่อยู่ในถังที่ 2 โดยเซ็นเซอร์จะติดเมื่อระดับของเหลวอยู่ที่ตำแหน่งก้นถังหรือไม่มีของเหลวอยู่ในถังนั้นเอง และจะดับก็ต่อเมื่อระดับของของเหลวอยู่เหนือเซ็นเซอร์หรืออีกนัยหนึ่งคือมีของเหลวอยู่ในถังนั้นเอง
- **ตัวเลข** 03 (I03) เป็นเซ็นเซอร์จับตำแหน่งของของเหลวที่อยู่ในถังที่ 2 โดยเซ็นเซอร์จะติดเมื่อระดับของเหลวอยู่ที่ตำแหน่งขอบถังหรือมีของเหลวอยู่เต็มถังนั้นเอง และจะดับก็ต่อเมื่อระดับของของเหลวอยู่ต่ำกว่าเซ็นเซอร์หรืออีกนัยหนึ่งคือมีของเหลวอยู่ในถังแต่มีไม่ถึงขอบถัง
- **ตัวเลข** 04 (I04) เป็นเซ็นเซอร์จับตำแหน่งของของเหลวที่อยู่ในถังที่ 3 โดยเซ็นเซอร์จะติดเมื่อระดับของเหลวอยู่ที่ตำแหน่งก้นถังหรือไม่มีของเหลวอยู่ในถังนั้นเอง และจะดับก็ต่อเมื่อระดับของของเหลวอยู่เหนือเซ็นเซอร์หรืออีกนัยหนึ่งคือมีของเหลวอยู่ในถังนั้นเอง
- **ตัวเลข** 06 (I06) เป็นเซ็นเซอร์จับตำแหน่งของของเหลวที่อยู่ในถังที่ 3 โดยเซ็นเซอร์จะติดเมื่อระดับของเหลวอยู่ที่ตำแหน่งกึ่งกลางถัง และจะดับก็ต่อเมื่อระดับของของเหลวอยู่ต่ำหรือสูงกว่าตำแหน่งของเซ็นเซอร์
- **ตัวเลข** 07 (I07) เป็นเซ็นเซอร์จับตำแหน่งของของเหลวที่อยู่ในถังที่ 3 โดยเซ็นเซอร์จะติดเมื่อระดับของเหลวอยู่ที่ตำแหน่งขอบถังหรือมีของเหลวอยู่เต็มถังนั้นเอง และจะดับก็ต่อเมื่อระดับของของเหลวอยู่ต่ำกว่าเซ็นเซอร์หรืออีกนัยหนึ่งคือมีของเหลวอยู่ในถังแต่มีไม่ถึงขอบถัง

ส่วนของเอาพุท คือวาล์วของถังที่จะเป็นตัวเปิด/ปิดของเหลวโดยจะมีรายละเอียดดังนี้

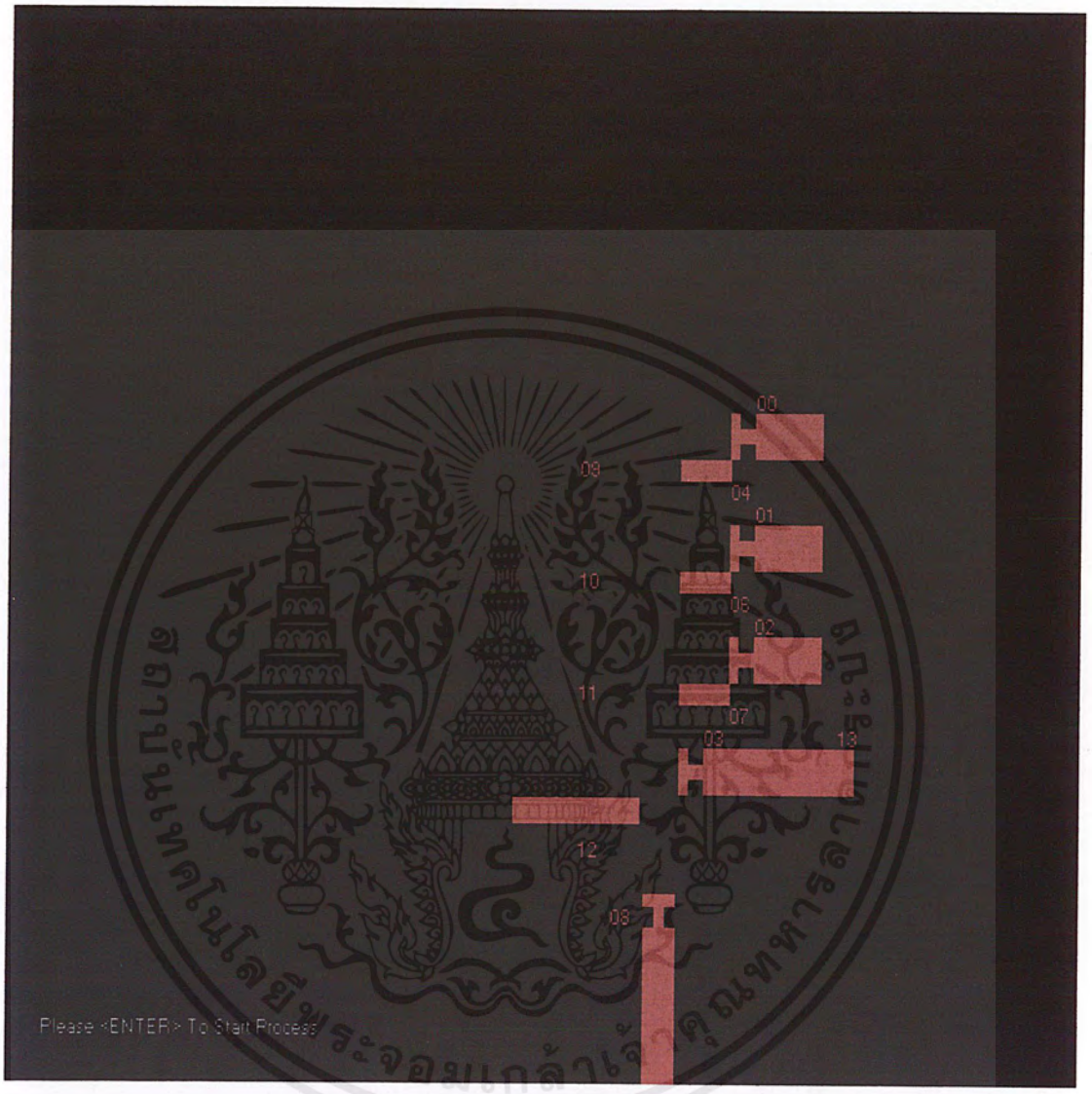
- **เอาพุท0** (Q00) คือวาล์วตัวซ้ายบนสุดที่เปิดของเหลวสีเหลืองเข้ายังถังที่ 1(ถังด้านซ้ายบน)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- **เอาพุท1 (Q01)** คือวาล์วตัวขวาบนสุดที่เปิดของเหลวสีน้ำเงินเข้ายังถังที่ 2(ถังด้านขวาบน)
- **เอาพุท2 (Q02)** คือวาล์วตัวซ้ายที่เปิดของเหลวสีเหลืองจากถังที่ 1 ไปยังถังที่ 3(ถังใหญ่ตรงกลาง)
- **เอาพุท3 (Q03)** คือวาล์วตัวขวาที่เปิดของเหลวสีน้ำเงินจากถังที่ 2 ไปยังถังที่ 3
- **เอาพุท4 (Q04)** คือวาล์วตัวล่างที่เปิดให้ของเหลวไหลออกจากถังที่ 3 ไปใช้ในโรงงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

➢ ระบบนิวเมติก




รูปที่ 3.13 ระบบจำลองการทำงานของนิวเมติก

จากภาพจะเห็นตัวเลขอยู่ตามตำแหน่งต่างๆของลูกสูบ โดยจะมีตัวเลขตั้งแต่ 00 – 12 โดยตัวเลขแต่ละตัวแสดงถึงตำแหน่งของเซ็นเซอร์(SENSOR) ที่ติดตั้งในระบบจริง เพื่อใช้เป็นอินพุทของ PLC เพื่อใช้ในการประมวลผล โดยมีรายละเอียดแต่ละตัวดังนี้

➢ **ตัวเลข 00 (I00)** เป็นเซ็นเซอร์จับตำแหน่งของลูกสูบที่อยู่ที่ลูกสูบที่ 1(ลูกสูบที่อยู่บนสุด) โดยเซ็นเซอร์จะติดเมื่อลูกสูบอยู่ที่ตำแหน่งยืดออกมาสุด และจะดับก็ต่อเมื่อลูกสูบไม่อยู่ที่ตำแหน่งยืดออกมาสุด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- **ตัวเลข** 01 (I01) เป็นเซ็นเซอร์จับตำแหน่งของลูกสูบที่อยู่ที่ลูกสูบที่ 2 โดยเซ็นเซอร์จะติดเมื่อลูกสูบอยู่ที่ตำแหน่งยืดออกมาสุด และจะดับก็ต่อเมื่อลูกสูบไม่อยู่ที่ตำแหน่งยืดออกมาสุด
- **ตัวเลข** 02 (I02) เป็นเซ็นเซอร์จับตำแหน่งของลูกสูบที่อยู่ที่ลูกสูบที่ 3 โดยเซ็นเซอร์จะติดเมื่อลูกสูบอยู่ที่ตำแหน่งยืดออกมาสุด และจะดับก็ต่อเมื่อลูกสูบไม่อยู่ที่ตำแหน่งยืดออกมาสุด
- **ตัวเลข** 03 (I03) เป็นเซ็นเซอร์จับตำแหน่งของลูกสูบที่อยู่ที่ลูกสูบที่ 4 โดยเซ็นเซอร์จะติดเมื่อลูกสูบอยู่ที่ตำแหน่งยืดออกมาสุด และจะดับก็ต่อเมื่อลูกสูบไม่อยู่ที่ตำแหน่งยืดออกมาสุด
- **ตัวเลข** 04 (I04) เป็นเซ็นเซอร์จับวัตถุติดตั้งอยู่บนชั้นชั้นบนสุดโดยเซ็นเซอร์จะติดเมื่อมีวัตถุอยู่บนชั้น และจะดับเมื่อไม่มีวัตถุวางอยู่บนชั้น
- **ตัวเลข** 06 (I06) เป็นเซ็นเซอร์จับวัตถุติดตั้งอยู่บนชั้นที่ 2 โดยเซ็นเซอร์จะติดเมื่อมีวัตถุอยู่บนชั้น และจะดับเมื่อไม่มีวัตถุวางอยู่บนชั้น
- **ตัวเลข** 07 (I07) เป็นเซ็นเซอร์จับวัตถุติดตั้งอยู่บนชั้นที่ 3 โดยเซ็นเซอร์จะติดเมื่อมีวัตถุอยู่บนชั้น และจะดับเมื่อไม่มีวัตถุวางอยู่บนชั้น
- **ตัวเลข** 08 (I08) เป็นเซ็นเซอร์จับตำแหน่งของลูกสูบที่อยู่ที่ลูกสูบที่ 5 (ตัวที่วางในแนวตั้ง) โดยเซ็นเซอร์จะติดเมื่อลูกสูบอยู่ที่ตำแหน่งถอยกลับสุด และจะดับก็ต่อเมื่อลูกสูบไม่อยู่ที่ตำแหน่งถอยกลับสุด
- **ตัวเลข** 09 (I09) เป็นเซ็นเซอร์จับตำแหน่งของลูกสูบที่อยู่ที่ลูกสูบที่ 5 โดยเซ็นเซอร์จะติดเมื่อลูกสูบอยู่ที่ตำแหน่งที่สามารถรองรับวัตถุจากชั้นบนสุดได้ และจะดับก็ต่อเมื่อลูกสูบไม่อยู่ที่ตำแหน่งที่กล่าวมาข้างต้น
- **ตัวเลข** 10 (I10) เป็นเซ็นเซอร์จับตำแหน่งของลูกสูบที่อยู่ที่ลูกสูบที่ 5 โดยเซ็นเซอร์จะติดเมื่อลูกสูบอยู่ที่ตำแหน่งที่สามารถรองรับวัตถุจากชั้นที่ 2 ได้ และจะดับก็ต่อเมื่อลูกสูบไม่อยู่ที่ตำแหน่งที่กล่าวมาข้างต้น
- **ตัวเลข** 11 (I11) เป็นเซ็นเซอร์จับตำแหน่งของลูกสูบที่อยู่ที่ลูกสูบที่ 5 โดยเซ็นเซอร์จะติดเมื่อลูกสูบอยู่ที่ตำแหน่งที่สามารถรองรับวัตถุจากชั้นที่ 3 ได้ และจะดับก็ต่อเมื่อลูกสูบไม่อยู่ที่ตำแหน่งที่กล่าวมาข้างต้น
- **ตัวเลข** 12 (I12) เป็นเซ็นเซอร์จับตำแหน่งของลูกสูบที่อยู่ที่ลูกสูบที่ 5 โดยเซ็นเซอร์จะติดเมื่อลูกสูบอยู่ที่ตำแหน่งที่สามารถที่จะให้ลูกสูบตัวที่ 4 ดันวัตถุไปยังชั้นด้านข้างได้ และจะดับก็ต่อเมื่อลูกสูบไม่อยู่ที่ตำแหน่งที่กล่าวมาข้างต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ส่วนของเอาพุท คือ การสั่งให้ลูกสูบทำงานและสั่งให้นำของออกมา โดยจะมีรายละเอียดดังนี้

- **เอาพุท0 (Q00)** คือ ลูกสูบตัวบนสุด
- **เอาพุท1 (Q01)** คือ ลูกสูบตัวที่ 2
- **เอาพุท2 (Q02)** คือ ลูกสูบตัวที่ 3
- **เอาพุท3 (Q03)** คือ ให้นำของที่อยู่ชั้นบนสุดออกมา
- **เอาพุท4 (Q04)** คือ ให้นำของที่อยู่ชั้นที่ 2 ออกมา
- **เอาพุท5 (Q05)** คือ ให้นำของที่อยู่ชั้นที่ 3 ออกมา
- **เอาพุท6 (Q06)** คือ ลูกสูบตัวที่ 4
- **เอาพุท7 (Q07)** คือ ลูกสูบตัวที่ 5 โดยจะทำให้ลูกสูบเลื่อนขึ้น
- **เอาพุท8 (Q08)** คือ ลูกสูบตัวที่ 5 โดยจะทำให้ลูกสูบเลื่อนลง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

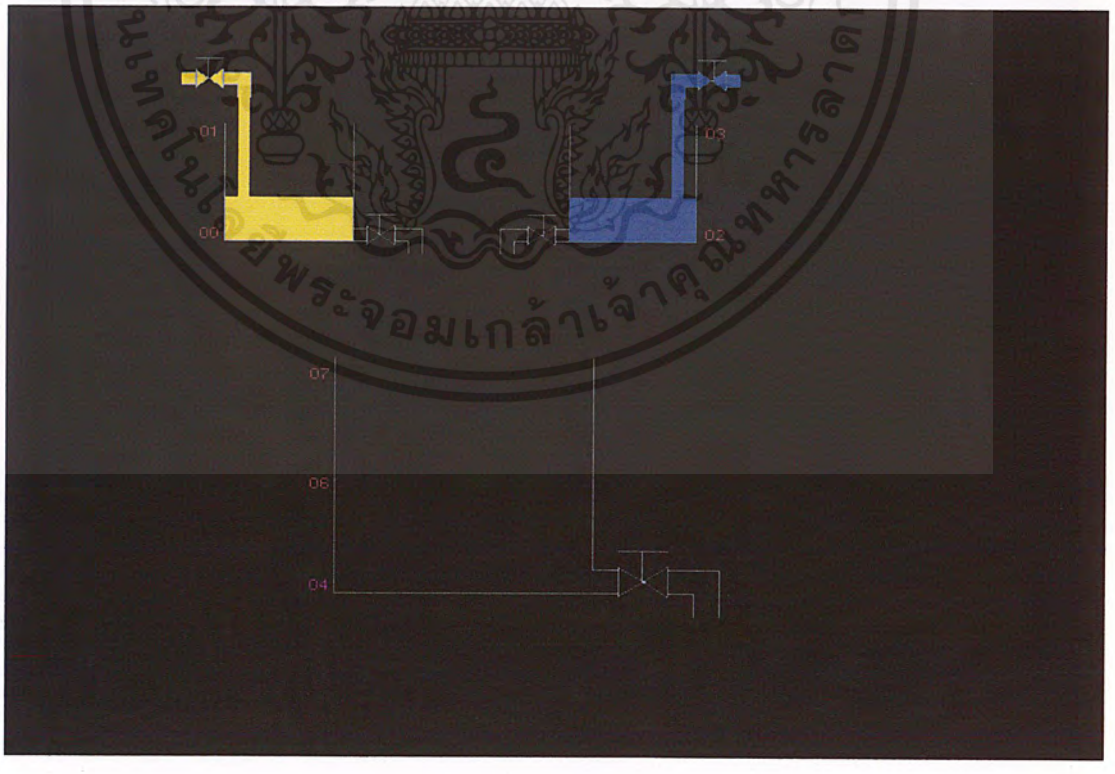
# บทที่ 4
# ผลการทดลอง

โปรแกรมในโครงงานนี้เป็นโปรแกรมที่สามารถจำลองการทำงานของระบบจริงมาไว้ในหน้าจอคอมพิวเตอร์ได้ โดยผ่านโปรแกรม PLC ซึ่งในที่นี้ได้จำลองระบบไว้ 2 ระบบดังที่ได้กล่าวมาไว้ข้างต้นแล้ว

ในที่นี้เราจะทำการทดลองใช้โปรแกรมในระบบถัง ซึ่งขั้นแรกต้องทำการเขียนแลดเดอร์ไดอะแกรมโดยมีลักษณะการทำงานดังต่อไปนี้

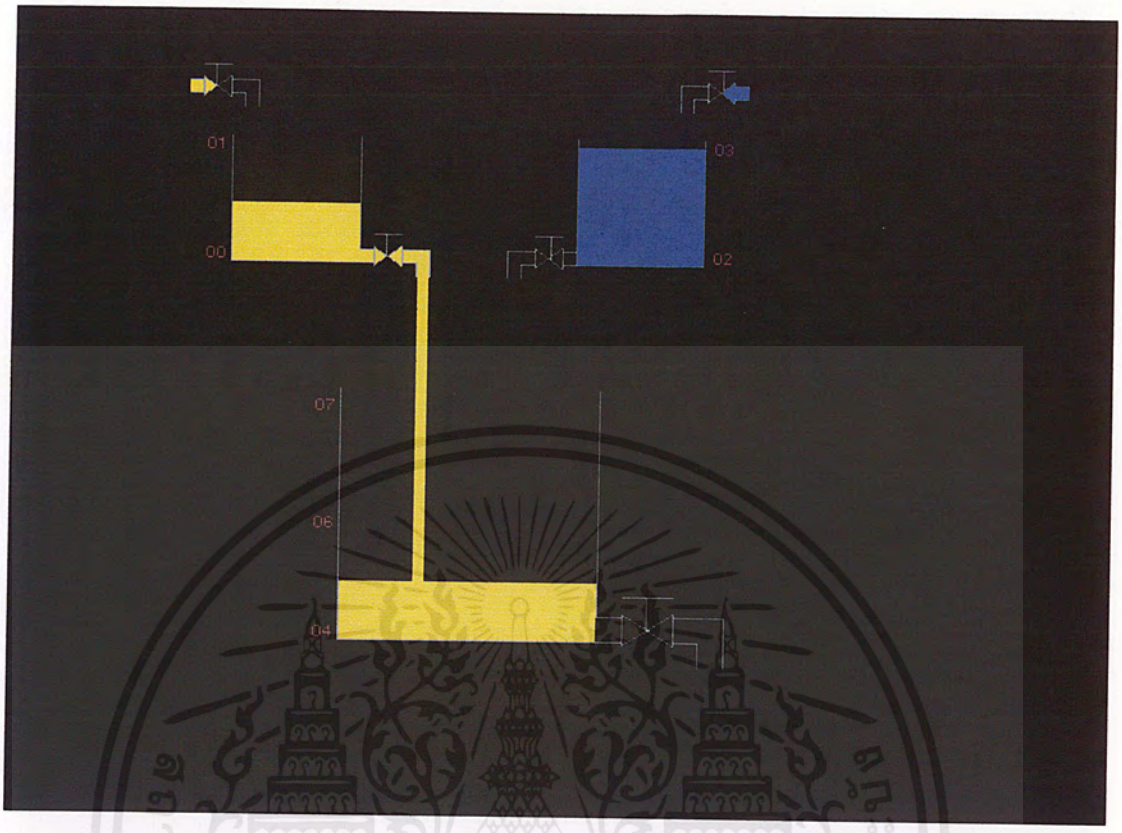
- ถังที่ 1 และ ถังที่ 2 จะต้องมีของเหลวอยู่ตลอดเวลา โดยการเปิด/ปิดวาล์วที่ 1 และ 2
- ถังที่ 3 จะต้องนำของเหลวจากถังที่ 1 และ 2 โดยอาศัยเซ็นเซอร์ที่อยู่กลางถังเป็นตัววัดปริมาณของของเหลวแต่ละชนิดที่ใส่ลงมา
- เมื่อผสมของเหลวจนเต็มถังที่ 3 แล้วให้ปล่อยของเหลวออกจากถังแล้วจึงผสมใหม่เรื่อยๆ

จากข้อมูลดังกล่าวข้างต้น เมื่อเขียนแลดเดอร์ไดอะแกรมเสร็จจึงทำการทดลอง และโปรแกรมแสดงผลเป็นดังรูป

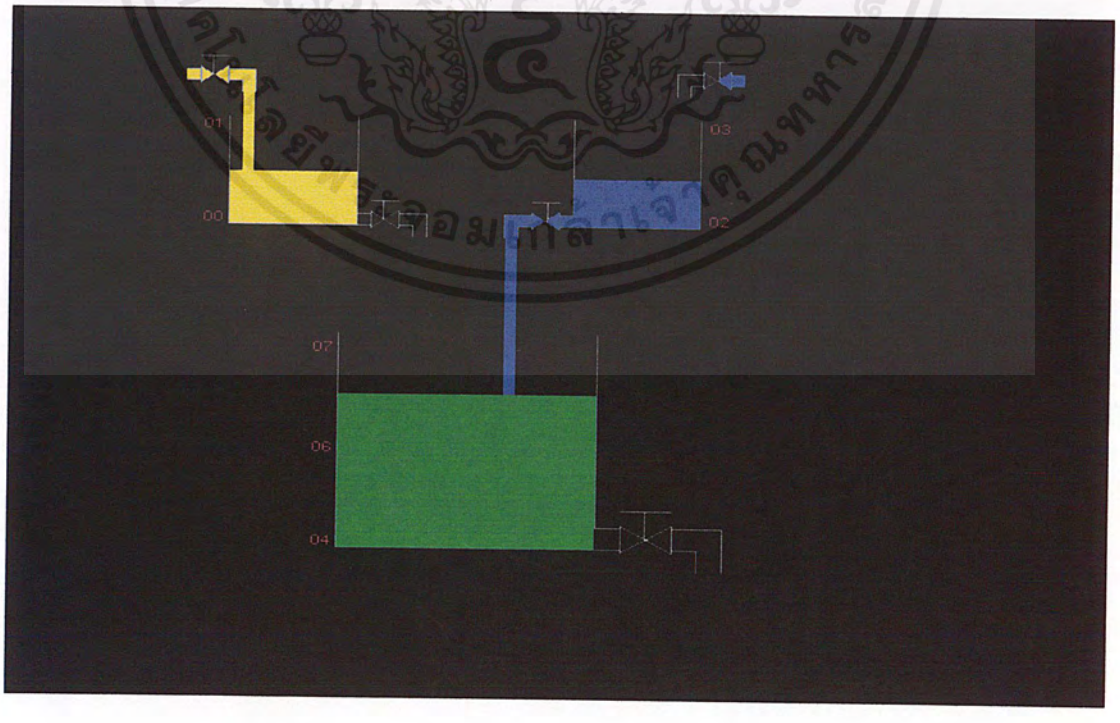



รูปที่ 4.1 เติมของเหลวเข้าถังที่ 1 และ 2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

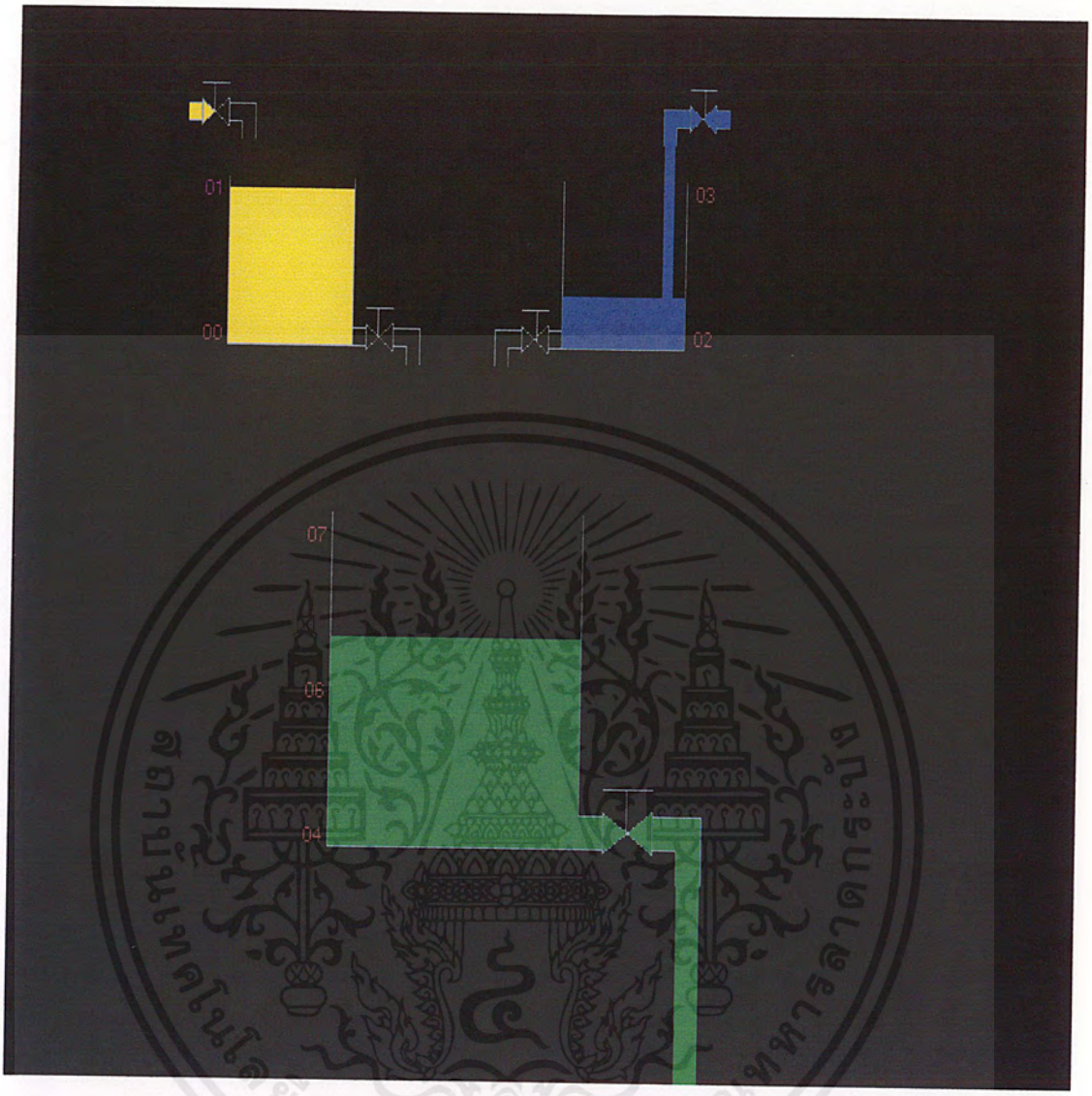



รูปที่ 4.2 การเติมของเหลวจากถังที่ 1 ไปยังถังที่ 3




รูปที่ 4.3 การเติมของเหลวจากถังที่ 2 ไปผสมกับถังที่ 1 ในถังที่ 3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

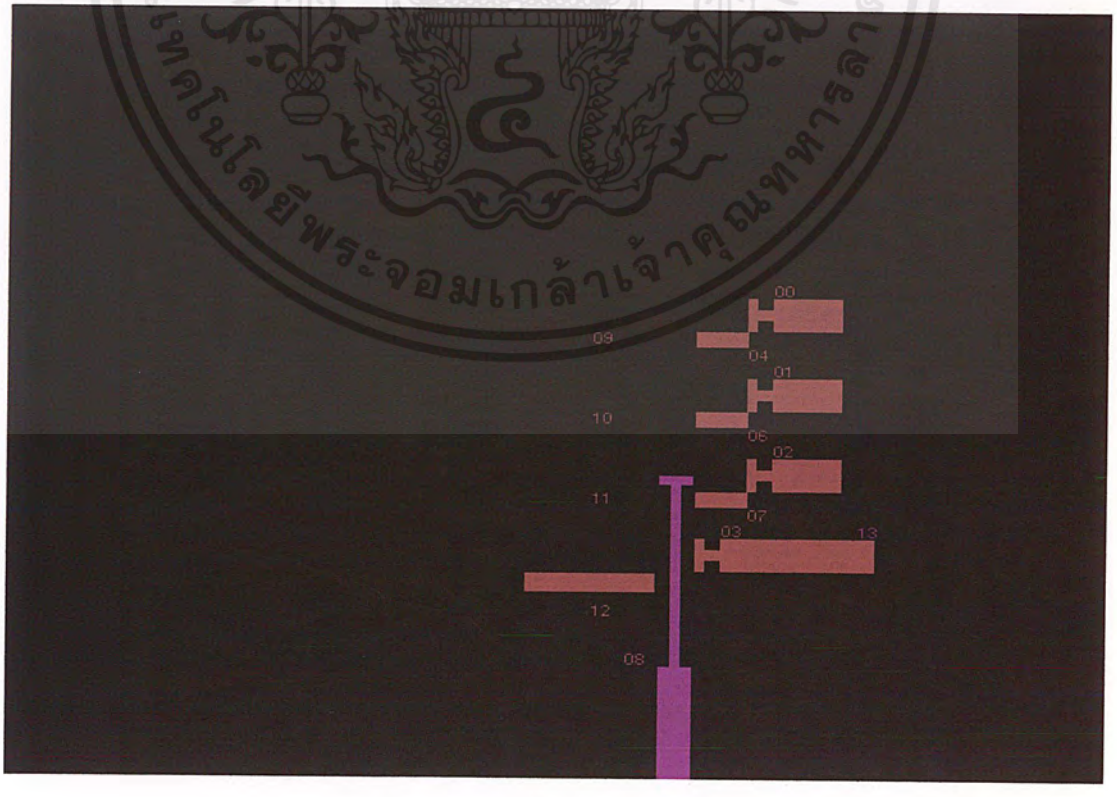



รูปที่ 4.4 การปล่อยของเหลวที่ผสมแล้วออกจากถังที่ 3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

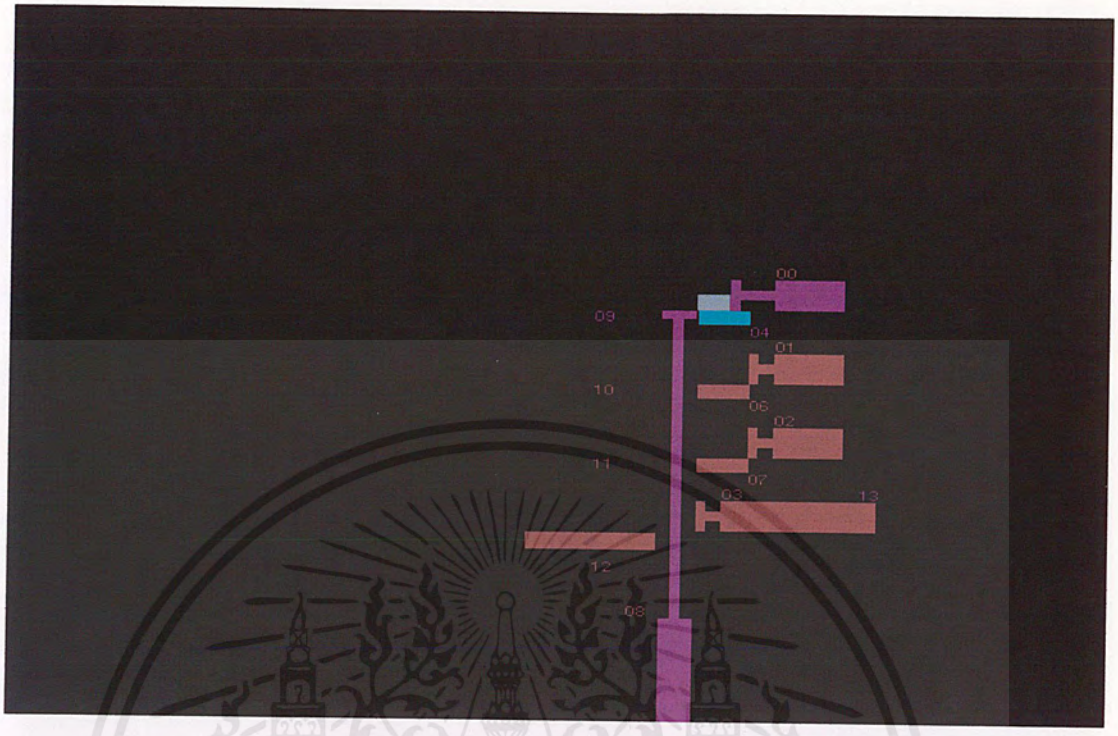
ต่อไปจะเป็นการทดลองระบบที่เป็นนิวเมติก โดยมีลักษณะของแลดเดอร์ไดอะแกรมดังต่อไปนี้

- เริ่มต้นจะต้องให้ลูกสูบตัวที่ 5 เคลื่อนที่ขึ้นไปรับของที่ลูกสูบตัวที่ 1 จะดันมาให้
- สั่งให้นำของออกมาวางยังชั้นที่ 1 แล้วจึงสั่งให้ ลูกสูบตัวที่ 1 เคลื่อนที่ดันของไปให้ลูกสูบตัวที่ 5
- สั่งให้ลูกสูบตัวที่ 5 เคลื่อนลงมารอรับของที่ชั้นที่ 2
- สั่งให้นำของออกมาวางยังชั้นที่ 2 แล้วจึงสั่งให้ ลูกสูบตัวที่ 2 เคลื่อนที่ดันของไปให้ลูกสูบตัวที่ 5
- สั่งให้ลูกสูบตัวที่ 5 เคลื่อนลงมารอรับของที่ชั้นที่ 3
- สั่งให้นำของออกมาวางยังชั้นที่ 2 แล้วจึงสั่งให้ ลูกสูบตัวที่ 2 เคลื่อนที่ดันของไปให้ลูกสูบตัวที่ 5
- สั่งให้ลูกสูบตัวที่ 5 เคลื่อนลงมาที่ชั้นวางของ
- สั่งให้ลูกสูบตัวที่ 4 เคลื่อนที่ดันของออกไป
- หลังจากนั้นให้ลูกสูบตัวที่ 4 และ 5 เคลื่อนที่กลับที่เดิม
- เริ่มต้นทำงานใหม่

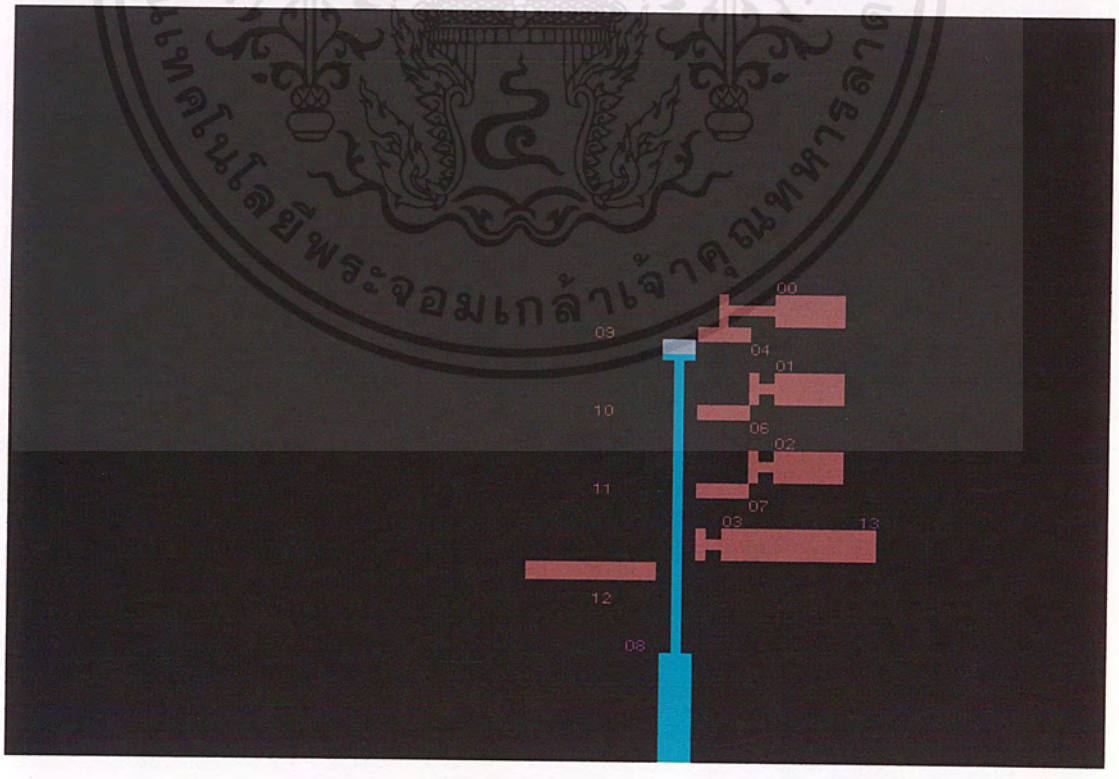



รูปที่ 4.5 ลูกสูบที่ 5 เคลื่อนที่ขึ้นไปรับของที่ชั้นที่ 1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

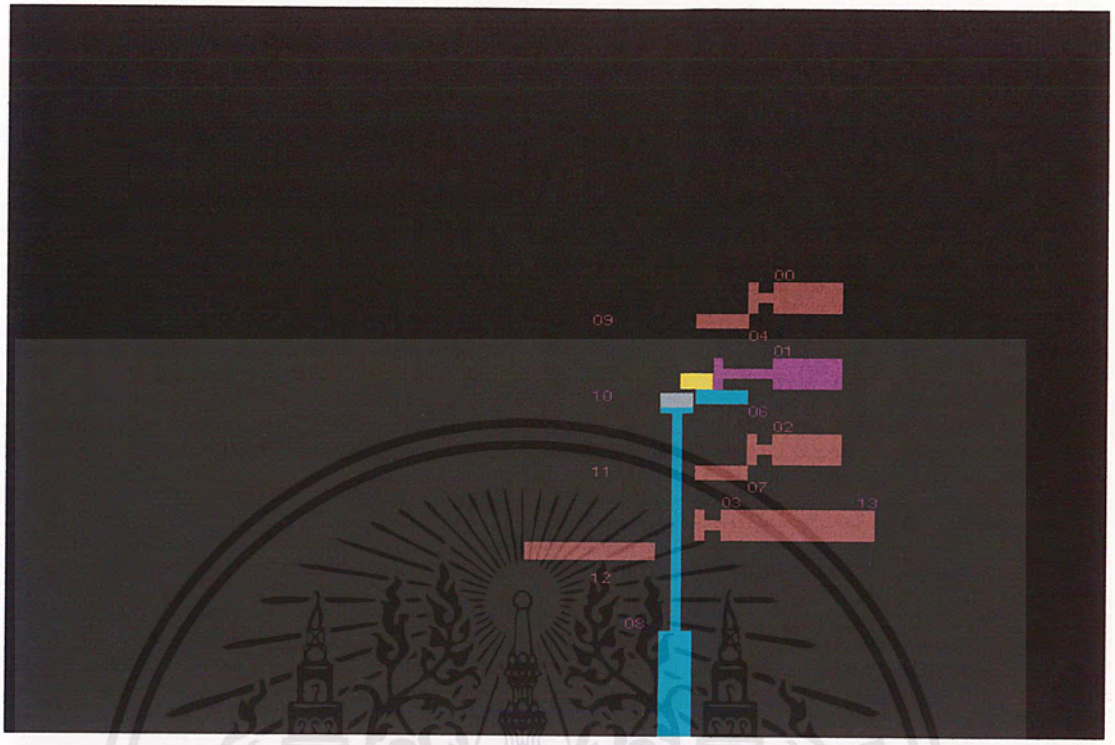



รูปที่ 4.6 ลูกสูบที่ 1 ดันของมายังลูกสูบที่ 5 ที่ขึ้นไปรอรับ

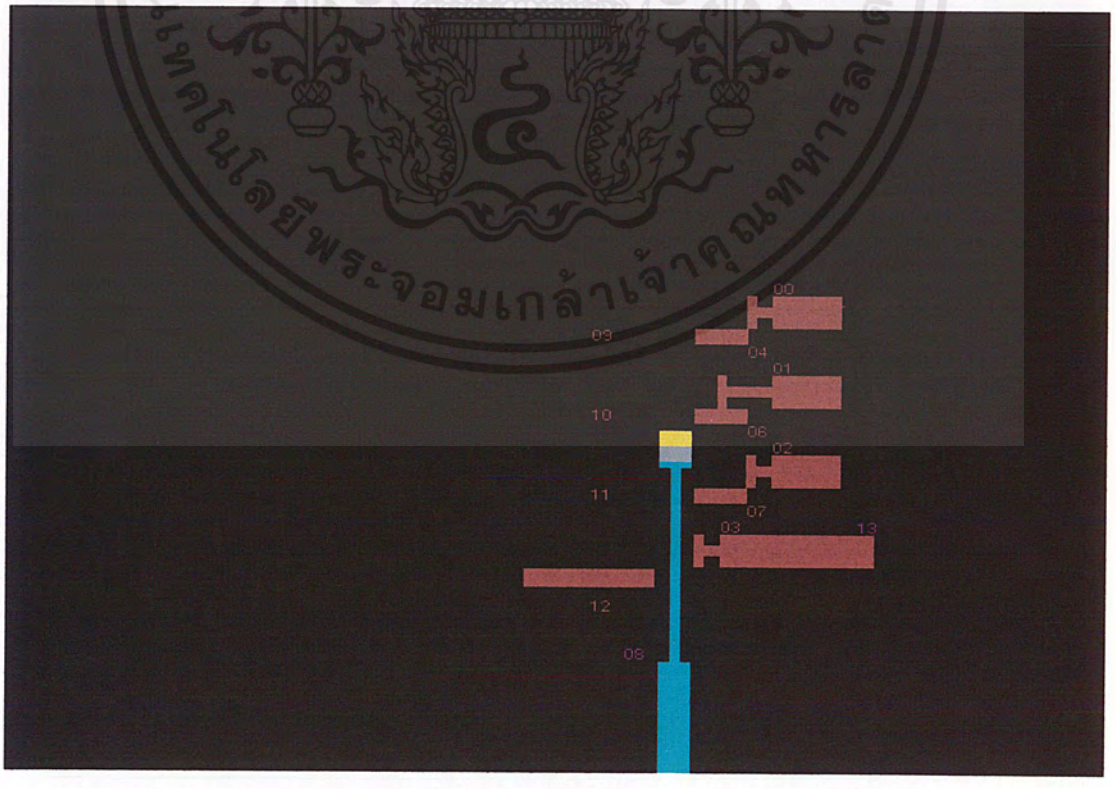



รูปที่ 4.7 ลูกสูบที่ 5 เคลื่อนลงมารอรับวัตถุที่ชั้นที่ 2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

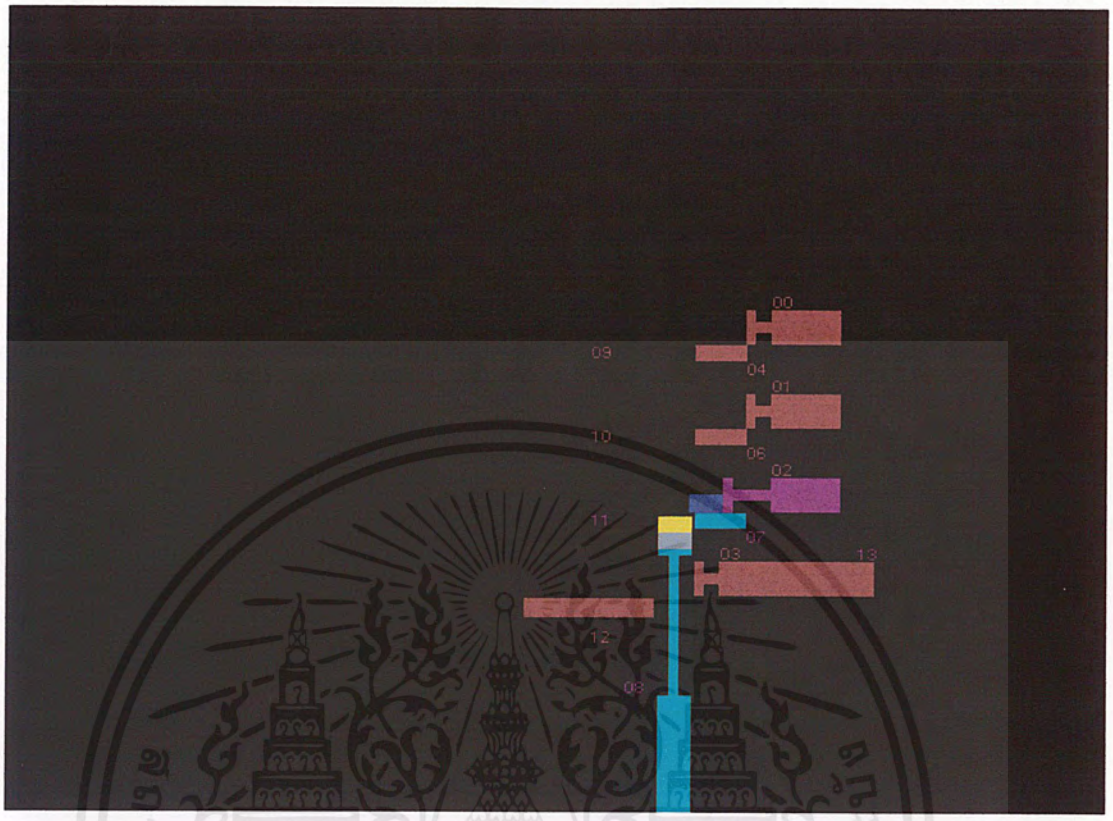



รูปที่ 4.8 ลูกสูบที่ 2 ดันวัตถุไปยังลูกสูบที่ 5 ที่รออยู่

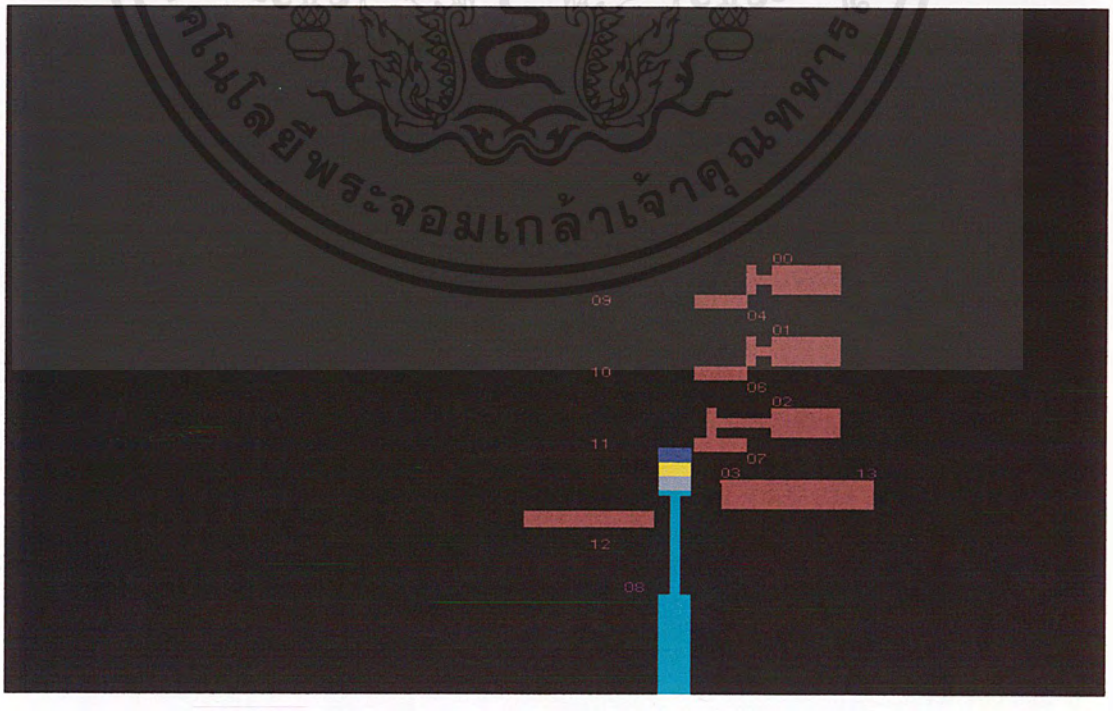



รูปที่ 4.9 ลูกสูบที่ 5 เคลื่อนลงไปรอรับวัตถุที่ชั้นที่ 3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

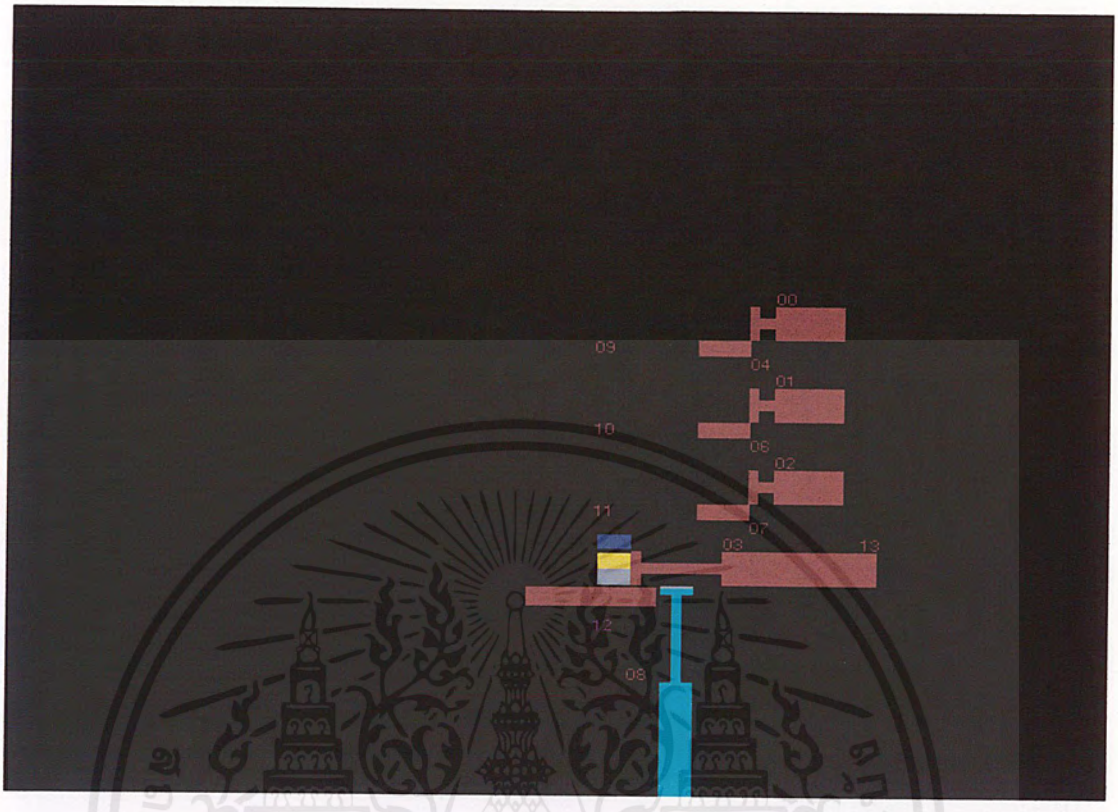



รูปที่ 4.10 ลูกสูบที่ 3 เคลื่อนที่ดันวัตถุไปยังลูกสูบที่ 5 ที่รออยู่

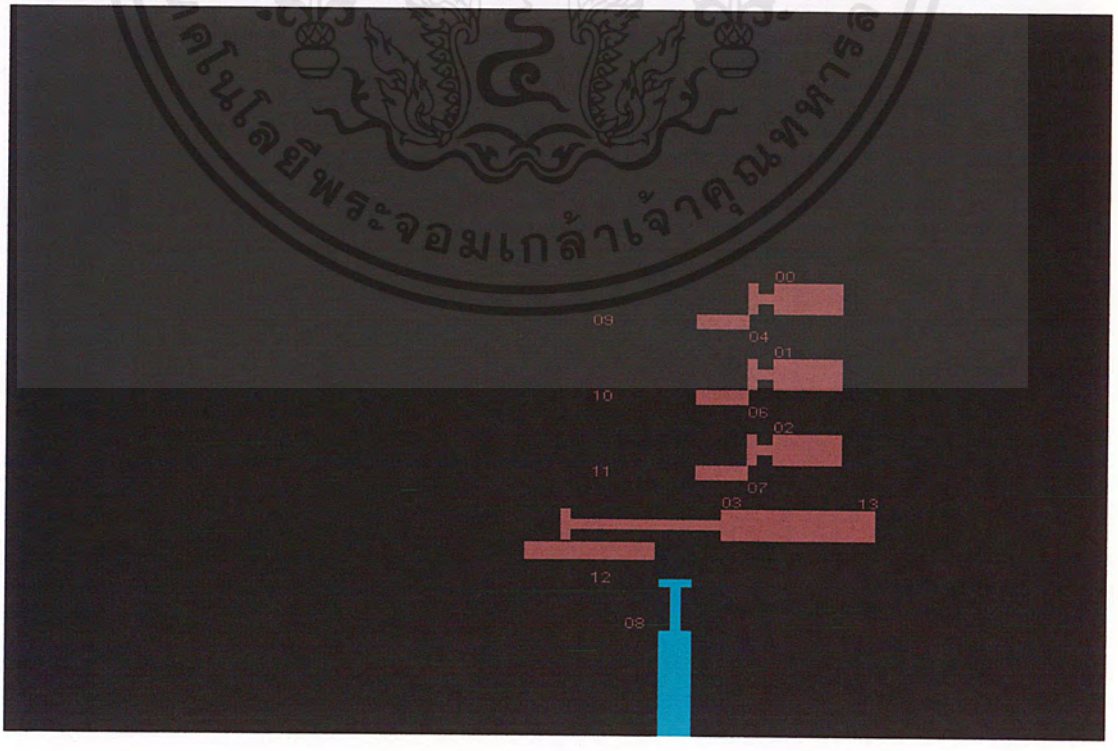



รูปที่ 4.11 ลูกสูบที่ 5 เคลื่อนที่ลงไปยังชั้นวางของ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้




รูปที่ 4.12 ลูกสูบที่ 4 เคลื่อนที่ดันวัตถุออกจากชั้น




รูปที่ 4.13 ลูกสูบทั้งหมดเคลื่อนที่กลับที่เดิม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการ[illegible]ไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากการทดลอง พบว่าทั้งระบบถังและระบบนิวเมติกสามารถจำลองการทำงาน และทำตามโปรแกรมที่เขียนไว้ในแลดเดดอร์ไดอะแกรมได้อย่างถูกต้อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# บทที่ 5
# บทวิจารณ์และสรุป

โครงงานชุดทดลอง PLC นี้จะเป็นโปรแกรมที่ทำการจำลองการทำงานของ PLC ไว้บนเครื่องคอมพิวเตอร์ และยังมีการแสดงผลแบบรูปภาพที่เคลื่อนไหวได้เพื่อที่จะสื่อความหมายกับผู้ใช้ได้ดีกว่า

โปรแกรมนี้ทางผู้จัดทำมีความเห็นว่าเป็นโปรแกรมที่มีความสมบูรณ์พอสมควร สามารถนำไปใช้เพื่อการศึกษาสำหรับนักเรียน นักศึกษาได้เป็นอย่างดี แต่อย่างไรก็ตามยังมีข้อบกพร่องที่ว่าอาจจะมีการจำลองระบบน้อยเกินไป และถ้าต้องการเพิ่มระบบใหม่ลงไปในโปรแกรมจะต้องทำการเขียนโปรแกรมเท่านั้น ไม่สามารถที่จะให้ผู้ใช้สร้างขึ้นมาเองได้เนื่องมาจากการแสดงผลเป็นภาพเคลื่อนไหว และยังมีความสวยงามของภาพที่ไม่ดีนัก เนื่องมาจากตัวโปรแกรมทำงานอยู่บนระบบ DOS และใช้วีดีโอโหมดที่มีสีเพียง 16 สีเท่านั้น แต่ก็มีส่วนดีที่จะทำให้สามารถใช้โปรแกรมบนเครื่องคอมพิวเตอร์ที่มีความสามารถต่ำได้โดยไม่มีปัญหา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# ภาคผนวก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้


CONTROL.C

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
#include <stdio.h>
#include <graph.h>
#include <project.h>

#define BLACK          0
#define BLUE           1
#define GREEN          2
#define CYAN           3
#define RED            4
#define MAGENTA        5
#define BROWN          6
#define WHITE          7
#define GRAYK          8
#define LIGHTBLUE      9
#define LIGHTREEN      10
#define LIGHTCYAN      11
#define LIGHTRED       12
#define LIGHTMAGENTA   13
#define YELLOW         14
#define BRIGHTWHITE    15

Animation_menu()
{
   _setvideomode( _VRES16COLOR );
   _registerfonts("*.FON");
   _setfont("t'helv'h18b");
   Control_Menu();
   setvideomode(3);
   return;
}

/*##############################################################################
###*/
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Control_Menu(int n)
  {
    n = 0;
    Draw_box(1,52,190,80);
    while(1)
          {
          if (n==0)
          {
          _setcolor(RED);
          _moveto(16,65);
          _outgtext("Tank Process");
          _setcolor(BLUE);
          _moveto(16,95);
          _outgtext("Pneumatic Process");
          }
        else if (n==1)
          {
          _setcolor(BLUE);
          _moveto(16,65);
          _outgtext("Tank Process");
          _setcolor(RED);
          _moveto(16,95);
          _outgtext("Pneumatic Process");
          }
          switch(readkey())
          {
          case UP: n = (n>0)? --n:1;break;
          case DOWN: n = (n<1)? ++n:0;break;
          case ESC: return;break;
          case ENTER: if (n==0) TankAnimation();
                      else if (n==1) PneAnimation();
                      else return;
                      break;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            }
        }
    }

/*##########################################################################
###*/
Draw_box(int left,int top,int width,int height)
  {
        _setcolor(CYAN);
        _rectangle(_GFILLINTERIOR,left,top,left+width-1,top+height-1);
        _setcolor(BRIGHTWHITE);
        _rectangle(_GBORDER,left+5,top+5,left+width-7,top+height-7);
        _setcolor(BRIGHTWHITE);
        _rectangle(_GBORDER,left+3,top+3,left+width-5,top+height-5);
  }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
#include <stdio.h>
#include <graph.h>
#include <dos.h>
#include <conio.h>
#include <plc.h>
#include <project.h>

int step1,step2,step3,step4,step5;
int PneX1,PneY1,PneX2,PneY2,PneX3,PneY3,PneX4,PneY4,PneX5,PneY5;
int stepstuff1,stepstuff2,stepstuff3,stepstuff4,stepstuff5;
int Q[15],I[15],T[15];
int flag1,flag2,flag3,flag4;
int outputpne,i,temp;

struct _dostime_t dtime;


PneAnimation()
{
   PneX1 = 485;
   PneY1 = 180;
   PneX2 = 485;
   PneY2 = 230;
   PneX3 = 485;
   PneY3 = 280;
   PneX4 = 505;
   PneY4 = 330;
   PneX5 = 398;
   PneY5 = 500;
   step1 = 0;
   step2 = 0;
   step3 = 0;
   step4 = 0;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
step5 = 0;
stepstuff1 = 0;
stepstuff2 = 0;
stepstuff3 = 0;
stepstuff4 = 0;
stepstuff5 = 0;
Q[0] = 0;
Q[1] = 0;
Q[2] = 0;
Q[3] = 0;
Q[4] = 0;
Q[5] = 0;
Q[6] = 0;
Q[7] = 0;
Q[8] = 0;
I[0] = 0;
I[1] = 0;
I[2] = 0;
I[3] = 0;
I[4] = 0;
I[5] = 0;
I[6] = 0;
I[7] = 0;
I[8] = 0;
I[9] = 0;
I[10] = 0;
I[11] = 0;
I[12] = 0;
I[13] = 0;
T[0] = 0;
T[1] = 0;
T[2] = 0;
T[3] = 0;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    flag1 = 0;
    _setvideomode(_VRES16COLOR);
    _registerfonts("*.FON");
    _setfont("t'helv'h8b");
    initscreen1();
    _getche();
    _moveto(20,450);
    _setcolor(BLACK);
    _outgtext("Please <ENTER> To Start Process");
    RunAnimation();
    setvideomode(3);
    return;
}

initscreen1()
{
    _setcolor(BLACK);
    _rectangle(_GFILLINTERIOR,0,0,640,480);
    Shelft();
    PneumaticHor(PneX1,PneY1,0,RED,WHITE);
    PneumaticHor(PneX2,PneY2,0,RED,WHITE);
    PneumaticHor(PneX3,PneY3,0,RED,WHITE);
    PneumaticHor1(PneX4,PneY4,0,RED,WHITE);
    PneumaticVer(PneX5,PneY5,0,RED,WHITE);
    Sensor(PneX1-40,PneY1-10,RED,"00");
    Sensor(PneX2-40,PneY2-10,RED,"01");
    Sensor(PneX3-40,PneY3-10,RED,"02");
    Sensor(PneX4-90,PneY4-10,RED,"03");
    Sensor(PneX1-55,PneY1+30,RED,"04");
    Sensor(PneX2-55,PneY2+30,RED,"06");
    Sensor(PneX3-55,PneY3+30,RED,"07");
    Sensor(PneX5-40,PneY5-100,RED,"08");
    Sensor(PneX5-60,PneY5-300,RED,"09");
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    Sensor(PneX5-60,PneY5-250,RED,"10");
    Sensor(PneX5-60,PneY5-200,RED,"11");
    Sensor(PneX5-60,PneY5-130,RED,"12");
    Sensor(PneX4-10,PneY4-10,RED,"13");
    _moveto(20,450);
    _setcolor(WHITE);
    _outgtext("Please <ENTER> To Start Process");
}

RunAnimation()
{
   while(!_kbhit())
          {
          Shelft();
          CheckSensor();
          UpDateTemp();
          Pne1_On();
          Stuff1Move();
          Pne1_Off();
          Pne2_On();
          Stuff2Move();
          Pne2_Off();
          Pne3_On();
          Stuff3Move();
          Pne3_Off();
          Pne4_On();
          Stuff4Move();
          Pne4_Off();
          Pne5_On();
          Pne5_Off();
          Stuff5Move();
          UpDatePne();
          }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    return;
}

UpDatePne()
{
  if (I[0] == 1)
        {
          DataTable[0] = DataTable[0]|1;
        }
  else
        {
          DataTable[0] = DataTable[0]&65534;
        }
  if (I[1] == 1)
        {
          DataTable[0] = DataTable[0]|2;
        }
  else
        {
          DataTable[0] = DataTable[0]&65533;
        }
  if (I[2] == 1)
        {
          DataTable[0] = DataTable[0]|4;
        }
  else
        {
          DataTable[0] = DataTable[0]&65531;
        }
  if (I[3] == 1)
        {
          DataTable[0] = DataTable[0]|8;
        }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
else
        {
          DataTable[0] = DataTable[0]&65527;
        }
if (I[4] == 1)
        {
          DataTable[0] = DataTable[0]|16;
        }
else
        {
          DataTable[0] = DataTable[0]&65519;
        }
if (I[5] == 1)
        {
          DataTable[0] = DataTable[0]|64;
        }
else
        {
          DataTable[0] = DataTable[0]&65439;
        }
if (I[6] == 1)
        {
          DataTable[0] = DataTable[0]|128;
        }
else
        {
          DataTable[0] = DataTable[0]&65407;
        }
if (I[7] == 1)
        {
          DataTable[0] = DataTable[0]|256;
        }
else
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        {
          DataTable[0] = DataTable[0]&65279;
        }
 if (I[8] == 1)
        {
          DataTable[0] = DataTable[0]|512;
        }
 else
        {
          DataTable[0] = DataTable[0]&65023;
        }
 if (I[9] == 1)
        {
          DataTable[0] = DataTable[0]|1024;
        }
 else
        {
          DataTable[0] = DataTable[0]&64511;
        }
 if (I[10] == 1)
        {
          DataTable[0] = DataTable[0]|2048;
        }
 else
        {
          DataTable[0] = DataTable[0]&63487;
        }
 if (I[11] == 1)
        {
          DataTable[0] = DataTable[0]|4096;
        }
 else
        {
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            DataTable[0] = DataTable[0]&61439;
        }
if (I[12] == 1)
        {
            DataTable[0] = DataTable[0]|8192;
        }
else
        {
            DataTable[0] = DataTable[0]&57343;
        }
outputpne = DataTable[64];
Q[0] = outputpne&1;
outputpne = DataTable[64];
outputpne = outputpne >> 1;
Q[1] = outputpne&1;
outputpne = DataTable[64];
outputpne = outputpne >> 2;
Q[2] = outputpne&1;
outputpne = DataTable[64];
outputpne = outputpne >> 3;
Q[3] = outputpne&1;
outputpne = DataTable[64];
outputpne = outputpne >> 4;
Q[4] = outputpne&1;
outputpne = DataTable[64];
outputpne = outputpne >> 5;
Q[5] = outputpne&1;
outputpne = DataTable[64];
outputpne = outputpne >> 6;
Q[6] = outputpne&1;
outputpne = DataTable[64];
outputpne = outputpne >> 7;
Q[7] = outputpne&1;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    outputpne = DataTable[64];
    outputpne = outputpne >> 8;
    Q[8] = outputpne&1;
    /*for(i=0;i<5;i++)
            {
                temp = DataTable[i];
                printf("%d",temp);
            }
    _getche();*/
}

Test()
{
    if (step5 >= 195 && T[3] == 0)
            {
                Q[0] = 1;
                Q[4] = 1;
            }
    if (I[0] == 1 )
            {
                Q[0] = 0;
                Q[7] = 0;
                Q[8] = 1;
                Q[4] = 0;
            }
    if (I[8] == 1 && I[7] != 1)
            {
                Q[7] = 0;
                Q[8] = 0;
            }
    if (I[9] == 1 && T[3] == 1)
            {
                Q[7] = 0;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            Q[8] = 0;
            Q[1] = 1;
            Q[5] = 1;
        }
if (I[1] == 1)
        {
            Q[7] = 0;
            Q[8] = 1;
            Q[5] = 0;
            Q[1] = 0;
        }
if (I[10] == 1 && T[3] == 1)
        {
            Q[7] = 0;
            Q[8] = 0;
            Q[2] = 1;
            Q[6] = 1;
        }
if (I[2] == 1)
        {
            Q[7] = 0;
            Q[8] = 1;
            Q[2] = 0;
            Q[6] = 0;
        }
if (I[11] == 1 && T[3] == 1)
        {
            Q[7] = 0;
            Q[8] = 0;
            Q[3] = 1;
        }
if (I[3] == 1)
        {
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            Q[7] = 0;
            Q[8] = 1;
            Q[3] = 0;
          }
   if (I[12] == 1 && I[7] == 0)
          {
            Q[7] = 1;
            Q[8] = 0;
          }
}

UpdateTemp()
{
   if (step5 >= 195 && I[7] == 1)
          {
            T[3] = 1;
          }
   if (I[0] == 1)
          {
            T[0] = 1;
          }
   if (I[1] == 1)
          {
            T[1] = 1;
          }
   if (I[2] == 1)
          {
            T[2] = 1;
          }
   if (I[3] == 1 && I[7] == 0)
          {
            T[0] = 0;
            T[1] = 0;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            T[2] = 0;
            T[3] = 0;
        }
    }

    Delay()
    {
      int oldtime,newtime,result;
      oldtime = 0;
      newtime = 0;
      _dos_gettime(&dtime);
      oldtime = dtime.second;
      for(;;)
            {
                _dos_gettime(&dtime);
                newtime = dtime.second;
                result = newtime-oldtime;
                if (abs(result) > 0 )
                  {
                    break;
                  }
            }
    }

    Delay1()
    {
      unsigned long i,j,k;
      for (i=0;i<2;i++)
          {
            _outtext("");
          }
    }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
/*##########################################################################
##*/
CheckSensor()
{
  if (step1 == 32)
        {
        Sensor(PneX1-40,PneY1-10,RED+1,"00");
        I[0] = 1;
        }
  else
        {
        Sensor(PneX1-40,PneY1-10,RED,"00");
        I[0] = 0;
        }
  if (step2 == 32)
        {
        Sensor(PneX2-40,PneY2-10,RED+1,"01");
        I[1] = 1;
        }
  else
        {
        Sensor(PneX2-40,PneY2-10,RED,"01");
        I[1] = 0;
        }
  if (step3 == 32)
        {
        Sensor(PneX3-40,PneY3-10,RED+1,"02");
        I[2] = 1;
        }
  else
        {
        Sensor(PneX3-40,PneY3-10,RED,"02");
        I[2] = 0;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        }
  if (step4 == 100)
        {
        Sensor(PneX4-90,PneY4-10,RED+1,"03");
        I[3] = 1;
        }
  else
        {
        Sensor(PneX4-90,PneY4-10,RED,"03");
        I[3] = 0;
        }
 if (I[4] == 1)
        Sensor(PneX1-55,PneY1+30,RED+1,"04");
 else Sensor(PneX1-55,PneY1+30,RED,"04");
 if (I[5] == 1)
        Sensor(PneX2-55,PneY2+30,RED+1,"06");
 else Sensor(PneX2-55,PneY2+30,RED,"06");
 if (I[6] == 1)
        Sensor(PneX3-55,PneY3+30,RED+1,"07");
 else Sensor(PneX3-55,PneY3+30,RED,"07");
 if (step1 == 32 && step5 == 195 && flag1 == 1)
        {
        Sensor(PneX5-40,PneY5-100,RED+1,"08");
        I[7] = 1;
        }
 if (step4 == 25 && step5 <= 45 && step5 >= 40)
        {
        Sensor(PneX5-40,PneY5-100,RED,"08");
        I[7] = 0;
        }
 if (step5 == 195)
        {
        Sensor(PneX5-60,PneY5-300,RED+1,"09");
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        I[8] = 1;
        }
    else
        {
        Sensor(PneX5-60,PneY5-300,RED,"09");
        I[8] = 0;
        }
    if (step5 <= 135 && step5 >= 130)
        {
        Sensor(PneX5-60,PneY5-250,RED+1,"10");
        I[9] = 1;
        }
    else
        {
        Sensor(PneX5-60,PneY5-250,RED,"10");
        I[9] = 0;
        }
    if (step5 <= 75 && step5 >= 70)
        {
        Sensor(PneX5-60,PneY5-200,RED+1,"11");
        I[10] = 1;
        }
    else
        {
        Sensor(PneX5-60,PneY5-200,RED,"11");
        I[10] = 0;
        }
    if (step5 <= 45 && step5 >= 43)
        {
        Sensor(PneX5-60,PneY5-130,RED+1,"12");
        I[11] = 1;
        }
    else
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        {
        Sensor(PneX5-60,PneY5-130,RED,"12");
        I[11] = 0;
        }
    if (step4 <= 0)
        {
        Sensor(PneX4-10,PneY4-10,RED+1,"13");
        I[12] = 1;
        }
    else
        {
        Sensor(PneX4-10,PneY4-10,RED,"13");
        I[12] = 0;
        }
}

Pne1_On()
{
   int i;
   i = step1;
   if (i < 32 && Q[0] == 1)
        {
          PneumaticHor(PneX1,PneY1,i,RED+Q[0],WHITE);
          i++;
        }
   step1 = i;
}

Pne1_Off()
{
   int i;
   i = step1;
   if (i >=-1 && Q[0] == 0)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        {
          PneumaticHor(PneX1,PneY1,i,RED,WHITE);
          i--;
        }
    if (i == -2 )
        {
          PneumaticHor(PneX1,PneY1,0,RED,WHITE);
        }
    step1 = i;
}

Stuff1Move()
{
    int i;
    i = stepstuff1;
    if (Q[4] == 1 && flag1 == 0)
        {
          StuffLeft(PneX1-55,PneY1+10,0,WHITE);
          I[4] = 1;
          flag1 = 1;
        }
    if (Q[4] == 0 && flag1 == 1)
        {
          flag1 = 0;
        }
    if (I[4] == 1 && stepstuff1 < 32 )
        {
          StuffLeft(PneX1-55,PneY1+10,i,WHITE);
          if (i < 33 && Q[0] == 1 && stepstuff1+1 == step1)
              {
                i++;
              }
        }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    if (stepstuff1 >=32)
          {
            StuffLeft(PneX1-55,PneY1+10,i,BLACK);
            I[4] = 0;
            i = 0;
          }
    stepstuff1 = i;
}

Pne2_On()
{
   int i;
   i = step2;
   if (i < 32 && Q[1] == 1)
          {
            PneumaticHor(PneX2,PneY2,i,RED+Q[1],WHITE);
            i++;
          }
   step2 = i;
}

Pne2_Off()
{
   int i;
   i = step2;
   if (i >=-1 && Q[1] == 0)
          {
            PneumaticHor(PneX2,PneY2,i,RED,WHITE);
            i--;
          }
  if (i == -2 )
          {
            PneumaticHor(PneX2,PneY2,0,RED,WHITE);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        }
    step2 = i;
}

Stuff2Move()
{
    int i;
    i = stepstuff2;
    if (Q[5] == 1 && flag2 == 0)
        {
            StuffLeft(PneX2-55,PneY2+10,0,YELLOW);
            I[5] = 1;
            flag2 = 1;
        }
    if (Q[5] == 0 && flag2 == 1)
        {
            flag2 = 0;
        }
    if (I[5] == 1 && stepstuff2 < 32)
        {
            StuffLeft(PneX2-55,PneY2+10,i,YELLOW);
            if (i < 33 && Q[1] == 1 && stepstuff2+1 == step2)
                {
                    i++;
                }
        }
    if (stepstuff2 >=32)
        {
            StuffLeft(PneX2-55,PneY2+10,i,BLACK);
            I[5] = 0;
            i = 0;
        }
    stepstuff2 = i;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
}

Pne3_On()
{
   int i;
   i = step3;
   if (i < 32 && Q[2] == 1)
         {
            PneumaticHor(PneX3,PneY3,i,RED+Q[2],WHITE);
            i++;
         }
   step3 = i;
}

Pne3_Off()
{
   int i;
   i = step3;
   if (i >=-1 && Q[2] == 0)
         {
            PneumaticHor(PneX3,PneY3,i,RED,WHITE);
            i--;
         }
   if (i == -2 )
         {
            PneumaticHor(PneX3,PneY3,0,RED,WHITE);
         }
   step3 = i;
}

Stuff3Move()
{
   int i;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    i = stepstuff3;
    if (Q[6] == 1 && flag3 == 0)
          {
            StuffLeft(PneX3-55,PneY3+10,0,BLUE);
            I[6] = 1;
            flag3 = 1;
          }
    if (Q[6] == 0 && flag3 == 1)
          {
            flag3 = 0;
          }
    if (I[6] == 1 && stepstuff3 < 32)
          {
            StuffLeft(PneX3-55,PneY3+10,i,BLUE);
            if (i < 33 && Q[2] == 1 && stepstuff3+1 == step3)
                {
                  i++;
                }
          }
    if (stepstuff3 >=32)
          {
            StuffLeft(PneX3-55,PneY3+10,i,BLACK);
            I[6] = 0;
            i = 0;
          }
    stepstuff3 = i;
}

Pne4_On()
{
    int i;
    i = step4;
    if (i < 100 && Q[3] == 1)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        {
          PneumaticHor1(PneX4,PneY4,i,RED+Q[6],WHITE);
          i++;
        }
    step4 = i;
}

Pne4_Off()
{
   int i;
   i = step4;
   if (i >=-1 && Q[3] == 0)
        {
          PneumaticHor1(PneX4,PneY4,i,RED,WHITE);
          i--;
        }
   if (i == -2 )
        {
          PneumaticHor1(PneX4,PneY4,0,RED,WHITE);
        }
   step4 = i;
}

Stuff4Move()
{
   int i;
   i = stepstuff4;
   if (stepstuff4 < 100 && I[11] == 1 && Q[3] == 1 && I[3] == 0)
        {
          StuffLeft(PneX4-105,PneY4+10,i,WHITE);
          StuffLeft(PneX4-105,PneY4,i,YELLOW);
          StuffLeft(PneX4-105,PneY4-10,i,BLUE);
          if (i < 100 && Q[3] == 1 && stepstuff4+1 == step4)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                {
                  i++;
                }
            }
    if (stepstuff4 >=100)
            {
              StuffLeft(PneX4-105,PneY4+10,i,BLACK);
              StuffLeft(PneX4-105,PneY4,i,BLACK);
              StuffLeft(PneX4-105,PneY4-10,i,BLACK);
              i = 0;
              stepstuff5 = 0;
            }
    stepstuff4 = i;
}

Pne5_On()
{
   int i;
   i = step5;
   if (i < 195 && Q[7] == 1 && Q[8] == 0)
            {
              PneumaticVer(PneX5,PneY5,i,RED+Q[7],WHITE);
              i++;
            }
   step5 = i;
}

Pne5_Off()
{
   int i;
   i = step5;
   if (i >=-1 && Q[8] == 1 && Q[7] == 0)
            {
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            PneumaticVer(PneX5,PneY5,i,RED-Q[8],WHITE);
            i--;
          }
    if (i == -2 )
          {
            PneumaticHor(PneX5,PneY5,0,RED,WHITE);
          }
    step5 = i;
}

Stuff5Move()
{
  int i;
  i = stepstuff5;
  if (I[7] == 1 && stepstuff5 < 193 && Q[8] == 1 && I[12] == 1)
        {
          StuffDown(PneX5,PneY5-309,i,WHITE);
          if (T[1] == 1 )
              StuffDown(PneX5,PneY5-319,i,YELLOW);
          if (T[2] == 1 )
              StuffDown(PneX5,PneY5-329,i,BLUE);
          if (i < 193 && Q[7] == 0 )
              {
                i = i++;
              }
        }
  if (stepstuff5 >= 193)
        {
          I[4] = 0;
          i = 0;
        }
  stepstuff5 = i;
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
/*##########################################################################
##*/
Shelft()
{
   _setcolor(BLACK);
   _rectangle(_GFILLINTERIOR,400,201,430,210);
   _rectangle(_GFILLINTERIOR,400,251,430,260);
   _rectangle(_GFILLINTERIOR,400,301,430,310);
   _setcolor(RED-I[4]);
   _rectangle(_GFILLINTERIOR,400,201,430,210);
   _setcolor(RED-I[5]);
   _rectangle(_GFILLINTERIOR,400,251,430,260);
   _setcolor(RED-I[6]);
   _rectangle(_GFILLINTERIOR,400,301,430,310);
   _setcolor(RED);
   _rectangle(_GFILLINTERIOR,375,363,300,352);
}

Sensor(x,y,color,string)
int x,y,color;
char string[10];
{
   _setcolor(color);
   _moveto(x,y);
   _outgtext(string);
}
/*##########################################################################
##*/
PneumaticHor(x,y,step,color,color1)
int x,y,step,color,color1;
  {
        _setcolor(BLACK);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        _rectangle(_GFILLINTERIOR,x-40,y-10,x-56-step,y+20);
        _setcolor(color);
        _rectangle(_GFILLINTERIOR,x,y,x-40,y+20);
        _rectangle(_GFILLINTERIOR,x,y+7,x-50-step,y+13);
        _rectangle(_GFILLINTERIOR,x-50-step,y,x-55-step,y+20);
    }

PneumaticHor1(x,y,step,color,color1)
int x,y,step,color,color1;
    {
        _setcolor(BLACK);
        _rectangle(_GFILLINTERIOR,x-90,y-10,x-106-step,y+20);
        _setcolor(color);
        _rectangle(_GFILLINTERIOR,x,y,x-90,y+20);
        _rectangle(_GFILLINTERIOR,x,y+7,x-100-step,y+13);
        _rectangle(_GFILLINTERIOR,x-100-step,y,x-105-step,y+20);
    }

PneumaticVer(x,y,step,color,color1)
int x,y,step,color,color1;
    {
        _setcolor(BLACK);
        _rectangle(_GFILLINTERIOR,x,y-90,x-20,y-107-step);
        _setcolor(color);
        _rectangle(_GFILLINTERIOR,x,y,x-20,y-90);
        _rectangle(_GFILLINTERIOR,x-7,y,x-13,y-100-step);
        _rectangle(_GFILLINTERIOR,x,y-100-step,x-20,y-105-step);
    }

/*##########################################################################
##*/
StuffLeft(x,y,step,color)
int x,y,step,color;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  {
        _setcolor(BLACK);
        _rectangle(_GFILLINTERIOR,x-1-step,y,x-20-step,y+10);
        _setcolor(color);
        _rectangle(_GFILLINTERIOR,x-1-step,y,x-20-step,y+10);
  }
/*##########################################################################
##*/
StuffDown(x,y,step,color)
int x,y,step;
  {
        _setcolor(BLACK);
        _rectangle(_GFILLINTERIOR,x,y-1+step,x-20,y+10+step);
        _setcolor(color);
        _rectangle(_GFILLINTERIOR,x,y+1+step,x-20,y+10+step);
  }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้


# PRO2.C

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
#include <graph.h>
#include <dos.h>
#include <conio.h>
#include <plc.h>
#include <project.h>

int TankX1,TankY1,TankX2,TankY2,TankX3,TankY3;
int capa1,capa2,capa3,capaleft,caparight,MIX;
int Q[15],I[15];
int flag1,flag2,flag3,flag4;
int temp,i;
int outputtank;

TankAnimation()
{
   TankX1 = 200;
   TankY1 = 150;
   TankX2 = 400;
   TankY2 = 150;
   TankX3 = 340;
   TankY3 = 375;
   capa1 = 0;
   capa2 = 0;
   capa3 = 0;
   capaleft = 0;
   caparight = 0;
   MIX = 0;
   Q[0] = 0;
   Q[1] = 0;
   Q[2] = 0;
   Q[3] = 0;
   Q[4] = 0;
   Q[5] = 0;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    _setvideomode(_VRES16COLOR);
    _registerfonts("*.FON");
    _setfont("t'helv'h8b");
    Init();
    _getche();
    _moveto(20,450);
    _setcolor(BLACK);
    _outgtext("Please <ENTER> To Start Process");
    RunAnimationTank();
    _moveto(20,450);
    _setcolor(RED);
    _outgtext("Please <ENTER> To Start Process");
    setvideomode(3);
    return;
}

Init()
{
   LeftValve(100,50,YELLOW,BLACK,8);
   RightValve(425,50,BLUE,BLACK,8);
   Tank(TankX1,TankY1,0,YELLOW,75);
   LeftValve(TankX1,TankY1,BLACK,BLACK,8);
   Tank(TankX2,TankY2,0,YELLOW,75);
   RightValve(TankX2-75,TankY2,BLACK,BLACK,8);
   Tank(TankX3,TankY3,0,GREEN,150);
   LeftValve(TankX3,TankY3,BLACK,BLACK,15);
   SensorTank(TankX1-90,TankY1-10,RED,"00");
   SensorTank(TankX1-90,TankY1-75,RED,"01");
   SensorTank(TankX2+5,TankY2-10,RED,"02");
   SensorTank(TankX2+5,TankY2-75,RED,"03");
   SensorTank(TankX3-165,TankY3-10,RED,"04");
   SensorTank(TankX3-165,TankY3-75,RED,"06");
   SensorTank(TankX3-165,TankY3-145,RED,"07");
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    _moveto(20,450);
    _setcolor(WHITE);
    _outgtext("Please <ENTER> To Start Process");
  }

  RunAnimationTank()
  {
    while(!_kbhit())
        {
        /*TestTank();*/
        CheckSensorTank();
        CheckTankLeft();
        CheckTankRight();
        CheckTank();
        CheckUpLeft();
        CheckUpRight();
        CheckLeft();
        CheckRight();
        CheckDown();
        UpDateTank();
        }
}

UpdateTank()
{
  if (l[0] == 1)
      {
        DataTable[0] = DataTable[0]|1;
      }
  else
      {
        DataTable[0] = DataTable[0]&65534;
      }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
if (I[1] == 1)
        {
          DataTable[0] = DataTable[0]|2;
        }
    else
        {
          DataTable[0] = DataTable[0]&65533;
        }
    if (I[2] == 1)
        {
          DataTable[0] = DataTable[0]|4;
        }
    else
        {
          DataTable[0] = DataTable[0]&65531;
        }
    if (I[3] == 1)
        {
          DataTable[0] = DataTable[0]|8;
        }
    else
        {
          DataTable[0] = DataTable[0]&65527;
        }
    if (I[4] == 1)
        {
          DataTable[0] = DataTable[0]|16;
        }
    else
        {
          DataTable[0] = DataTable[0]&65519;
        }
    if (I[6] == 1)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        {
          DataTable[0] = DataTable[0]|64;
        }
    else
        {
          DataTable[0] = DataTable[0]&65439;
        }
    if (l[7] == 1)
        {
          DataTable[0] = DataTable[0]|128;
        }
    else
        {
          DataTable[0] = DataTable[0]&65407;
        }
    outputtank = DataTable[64];
    Q[0] = outputtank&1;
    outputtank = DataTable[64];
    outputtank = outputtank >> 1;
    Q[1] = outputtank&1;
    outputtank = DataTable[64];
    outputtank = outputtank >> 2;
    Q[2] = outputtank&1;
    outputtank = DataTable[64];
    outputtank = outputtank >> 3;
    Q[3] = outputtank&1;
    outputtank = DataTable[64];
    outputtank = outputtank >> 4;
    Q[4] = outputtank&1;
    /*for(i=0;i<5;i++)
        {
          temp = DataTable[i];
          printf("%d",temp);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        }
    _getche();*/
}


TestTank()
{
  if (I[0] == 1)
        {
          Q[0] = 1;
        }
  if (I[1] == 1)
        {
          Q[0] = 0;
        }
  if (I[2] == 1)
        {
          Q[1] = 1;
        }
  if (I[3] == 1)
        {
          Q[1] = 0;
        }
 if (I[4] == 1)
        {
          Q[2] = 1;
        }
 if (I[5] == 1)
        {
          Q[2] = 0;
          Q[3] = 1;
        }
 if (I[6] == 1)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        {
          Q[3] = 0;
          Q[4] = 1;
        }
  if (I[4] == 1)
        {
          Q[4] = 0;
        }
}

CheckSensorTank()
{
  if (capa1 == 0)
        {
          SensorTank(TankX1-90,TankY1-10,RED+1,"00");
          I[0] = 1;
        }
  else
        {
          SensorTank(TankX1-90,TankY1-10,RED,"00");
          I[0] = 0;
        }
  if (capa1 == 70)
        {
          SensorTank(TankX1-90,TankY1-75,RED+1,"01");
          I[1] = 1;
        }
  else
        {
          SensorTank(TankX1-90,TankY1-75,RED,"01");
          I[1] = 0;
        }
  if (capa2 == 0)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        {
            SensorTank(TankX2+5,TankY2-10,RED+1,"02");
            I[2] = 1;
        }
    else
        {
            SensorTank(TankX2+5,TankY2-10,RED,"02");
            I[2] = 0;
        }
    if (capa2 == 70)
        {
            SensorTank(TankX2+5,TankY2-75,RED+1,"03");
            I[3] = 1;
        }
    else
        {
            SensorTank(TankX2+5,TankY2-75,RED,"03");
            I[3] = 0;
        }
    if (capa3 == 0)
        {
            SensorTank(TankX3-165,TankY3-10,RED+1,"04");
            I[4] = 1;
        }
    else
        {
            SensorTank(TankX3-165,TankY3-10,RED,"04");
            I[4] = 0;
        }
    if (capa3 >= 70 && capa3 <= 75)
        {
            SensorTank(TankX3-165,TankY3-75,RED+1,"06");
            I[6] = 1;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        }
    else
        {
          SensorTank(TankX3-165,TankY3-75,RED,"06");
          I[6] = 0;
        }
    if (capa3 >= 140)
        {
          SensorTank(TankX3-165,TankY3-145,RED+1,"07");
          I[7] = 1;
        }
    else
        {
          SensorTank(TankX3-165,TankY3-145,RED,"07");
          I[7] = 0;
        }
}

CheckUpLeft()
{
  if (Q[0] == 1)
        {
          LeftValve(100,50,YELLOW,YELLOW,8);
          FlowUpLeft(100,50,YELLOW,8);
          flag1 = 0;
        }
  else if (Q[0] == 0 && flag1 == 0)
        {
          LeftValve(100,50,YELLOW,BLACK,8);
          FlowUpLeft(100,50,BLACK,8);
          flag1 = 1;
        }
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
CheckUpRight()
{
  if (Q[1] == 1)
        {
          RightValve(425,50,BLUE,BLUE,8);
          FlowUpRight(425,50,BLUE,8);
          flag2 = 0;
        }
  else if (Q[1] == 0 && flag2 == 0)
        {
          RightValve(425,50,BLUE,BLACK,8);
          FlowUpRight(425,50,BLACK,8);
          flag2 = 1;
        }
}

CheckTankLeft()
{
  int i;
  i = capa1;
  if (Q[0] == 1 && i < 70)
        {
          i++;
        }
  if (Q[2] == 1 && i > 0)
        {
          Tank(TankX1,TankY1,i,BLACK,75);
          i--;
          capaleft++;
        }
  Tank(TankX1,TankY1,i,YELLOW,75);
  capa1 = i;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
}

CheckTankRight()
{
  int i;
  i = capa2;
  if (Q[1] == 1 && i < 70)
        {
          i++;
        }
  if (Q[3] == 1 && i > 0)
        {
          Tank(TankX2,TankY2,i,BLACK,75);
          i--;
          caparight++;
        }
  Tank(TankX2,TankY2,i,BLUE,75);
  capa2 = i;
}

CheckLeft()
{
  if (Q[2] == 1)
        {
          LeftValve(TankX1,TankY1,YELLOW,YELLOW,8);
          FlowLeft(TankX1,TankY1,YELLOW,8,224-capa3);
          flag3 = 0;
        }
  else if (Q[2] == 0 && flag3 == 0)
        {
          LeftValve(TankX1,TankY1,BLACK,BLACK,8);
          FlowLeft(TankX1,TankY1,BLACK,8,224);
          flag3 = 1;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        }
}

CheckRight()
{
  if (Q[3] == 1)
        {
          RightValve(TankX2-75,TankY2,BLUE,BLUE,8);
          FlowRight(TankX2,TankY2,BLUE,8,224-capa3);
          flag4 = 0;
        }
  else if (Q[3] == 0 && flag4 == 0)
        {
          RightValve(TankX2-75,TankY2,BLACK,BLACK,8);
          FlowRight(TankX2,TankY2,BLACK,8,224);
          flag4 = 1;
        }
}

CheckTank()
{
  int i;
  i = capa3;
  if (capaleft < 30 && Q[4] == 0)
        {
          MIX = BLUE;
        }
  if (caparight < 30 && Q[4] == 0)
        {
          MIX = YELLOW;
        }
  else
        {
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            MIX = GREEN;
        }
    if (i < 140)
        {
            capa3 = capaleft+caparight;
            i = capa3;
        }
    if (Q[4] == 1 && i > 0)
        {
            Tank(TankX3,TankY3,i,BLACK,150);
            i = i-2;
            capaleft--;
            caparight--;
        }
    Tank(TankX3,TankY3,i,MIX,150);
    capa3 = i;
    if (capa3 == -1)
        {
            capa3 = 0;
            i = 0;
        }
}

CheckDown()
{
  if (Q[4] == 1)
        {
            LeftValve(TankX3,TankY3,MIX,MIX,15);
            FlowDown(TankX3,TankY3,MIX,15);
        }
  else if (Q[4] == 0)
        {
            LeftValve(TankX3,TankY3,BLACK,BLACK,15);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            FlowDown(TankX3,TankY3,BLACK,15);
        }
}


Tank(int x,int y,int capa,int color,int size)
{
   _setcolor(color);
   _rectangle(_GFILLINTERIOR,x,y,x-size,y-capa);
   _setcolor(WHITE);
   _moveto(x,y-size);
   _lineto(x,y);
   _lineto(x-size,y);
   _lineto(x-size,y-size);
}

LeftValve(int x,int y,int color,int color1,int size)
{
   _setcolor(color);
   _rectangle(_GFILLINTERIOR,x,y,x+size,y-size);
   _setcolor(WHITE);
   _rectangle(_GBORDER,x,y,x+size,y-size);
   _setcolor(color);
   _moveto(x,y-1);
   _lineto(x,y-size+1);
   _setcolor(WHITE);
   _moveto(x+size,y+2);
   _lineto(x+size,y-size-2);
   _lineto(x+3*size,y+2);
   _lineto(x+3*size,y-size-2);
   _lineto(x+size,y+2);
   _setcolor(color);
   _floodfill(x+size+2,y-2,WHITE);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    _setcolor(color1);
    _floodfill(x+3*size-2,y-2,WHITE);
    _setcolor(WHITE);
    _moveto(x+2*size-1,y-size/2);
    _lineto(x+2*size-1,y-size/2-size-5);
    _moveto(x+size,y-size/2-size-5);
    _lineto(x+3*size,y-size/2-size-5);
    _moveto(x+3*size,y);
    _lineto(x+4*size,y);
    _lineto(x+4*size,y+size);
    _moveto(x+3*size,y-size);
    _lineto(x+5*size,y-size);
    _lineto(x+5*size,y+size);
    _setcolor(color1);
    _rectangle(_GFILLINTERIOR,x+3*size+1,y-1,x+5*size-1,y-size+1);
    _rectangle(_GFILLINTERIOR,x+4*size+1,y-1,x+5*size-1,y+size);
}

RightValve(int x,int y,int color,int color1,int size)
{
    _setcolor(color);
    _rectangle(_GFILLINTERIOR,x,y,x-size,y-size);
    _setcolor(WHITE);
    _rectangle(_GBORDER,x,y,x-size,y-size);
    _setcolor(color);
    _moveto(x,y-1);
    _lineto(x,y-size+1);
    _setcolor(WHITE);
    _moveto(x-size,y+2);
    _lineto(x-size,y-size-2);
    _lineto(x-3*size,y+2);
    _lineto(x-3*size,y-size-2);
    _lineto(x-size,y+2);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    _setcolor(color);
    _floodfill(x-size-2,y-2,WHITE);
    _setcolor(color1);
    _floodfill(x-3*size+2,y-2,WHITE);
    _setcolor(WHITE);
    _moveto(x-2*size+1,y-size/2);
    _lineto(x-2*size+1,y-size/2-size-5);
    _moveto(x-size,y-size/2-size-5);
    _lineto(x-3*size,y-size/2-size-5);
    _moveto(x-3*size,y);
    _lineto(x-4*size,y);
    _lineto(x-4*size,y+size);
    _moveto(x-3*size,y-size);
    _lineto(x-5*size,y-size);
    _lineto(x-5*size,y+size);
    _setcolor(color1);
    _rectangle(_GFILLINTERIOR,x-3*size-1,y-1,x-5*size+1,y-size+1);
    _rectangle(_GFILLINTERIOR,x-4*size-1,y-1,x-5*size+1,y+size);
}

SensorTank(x,y,color,string)
int x,y,color;
char string[10];
{
    _setcolor(color);
    _moveto(x,y);
    _outgtext(string);
}

FlowLeft(int x,int y,int color,int size,int end)
{
    _setcolor(color);
    _rectangle(_GFILLINTERIOR,x+4*size+1,y+size,x+4*size+7,y+end);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
}

FlowRight(int x,int y,int color,int size,int end)
{
   _setcolor(color);
   _rectangle(_GFILLINTERIOR,x-4*size-1-75,y+size,x-4*size-7-75,y+end);
}

FlowUpLeft(int x,int y,int color,int size)
{
   _setcolor(color);
   _rectangle(_GFILLINTERIOR,x+4*size+1,y+size,x+4*size+7,y+100-1);
}

FlowUpRight(int x,int y,int color,int size)
{
   _setcolor(color);
   _rectangle(_GFILLINTERIOR,x-4*size-1,y+size,x-4*size-7,y+100-1);
}

FlowDown(int x,int y,int color,int size)
{
   _setcolor(color);
   _rectangle(_GFILLINTERIOR,x+4*size+1,y+size,x+4*size+14,y+225-1);
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# กิตติกรรมประกาศ

ผู้จัดทำขอขอบคุณผู้มีรายนามต่อไปนี้ที่ได้ให้ความช่วยเหลือ ให้ข้อมูล การจัดทำ รวมทั้งคำแนะนำต่างๆ ที่เป็นประโยชน์ในการจัดทำปริญญานิพนธ์นี้

- รศ. สุเธียร เกียรติสุนทร
- คุณพ่อ-คุณแม่ที่รักและเคารพ
- เพื่อนๆภาคระบบควบคุมรุ่น 35

นอกจากนี้ยังขอขอบคุณกลุ่มเพื่อนๆน้องๆ รวมทั้งบุคคลใกล้ชิดที่คอยให้กำลังใจและสนับสนุนผู้จัดทำตลอดระยะเวลา 1 ปีที่ได้จัดทำปริญญานิพนธ์ฉบับนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# หนังสืออ้างอิง


1. กฤษดา วิศวธีรานนท์ ,"PC ตัวควบคุมซีเคว้นซ์ หลักการทำงานและการประยุกต์" , ศูนย์หนังสือจุฬาลงกรณ์มหาวิทยาลัย, 273 หน้า, 2539
2. พรจิต ประทุมสุวรรณ และคณะ ," ทฤษฎีและการใช้งาน (PC/PLC)", โรงพิมพ์เรือนแก้วการพิมพ์, 225 หน้า, 2536
3. ธันวา ศรีประโมง ," การเขียนโปรแกรมภาษาซีสำหรับวิศวกรรม", โครงการตำราวิชาการมหาวิทยาลัยเทคโนโลยีมหานคร, 712 หน้า, 2539
4. เฮอร์เบิร์ต ชิลด์ ," การประยุกต์ใช้งานภาษาซี ", ซีเอ็ดยูเคชั่น , 608 หน้า
5. สุทธิศักดิ์ พงศ์ธนาพานิช ," การเขียนโปรแกรมบน 80386/80486 " , ซีเอ็ดยูเคชั่น , 597 หน้า , 2537
6. ณรงค์ ตันชีวะวงศ์ ," ระบบPLC(Programmable Logic Controller) ," สมาคมส่งเสริมเทคโนโลยี(ไทย – ญี่ปุ่น), 405 หน้า, 2541
7. Mitsubishi Electric Corp. ," Programming Manual The Fx series of programmable controllers(Fxo ,Fxos ,Fxon ,Fx ,Fx2c ,Fx2n) " , 2541
8. Kip R. Irvine ," Assembly Language For The IBM – PC " ,Macmillan ,614 p. , 1990
9. Steven Holzner and Peter Norton ," Advanced Assembly Language " , Se – education, 1991



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้