

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

โปรแกรมการทำงานของลิฟต์ผ่านเครื่องควบคุม PLC
และโปรแกรม VISUAL BASIC สำหรับเฝ้ามอง

LIFT CONTROL BY PLC AND VISUAL BASIC FOR MONITORING



นายธีรุตม์ วัชรทัศนกุล
นายประเสริฐ เหล่ามงคลชัยศรี
นายภาณุพงศ์ สุรินทรามนต์

๒๖๓.
๕ 699๔/
๒๕๔๙

เลขหมู่.....
เลขทะเบียน..... 72230
วัน,เดือน,ปี..... 12 ส.ย. 2550

b. 117 6๕380
i.....

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิชาวิศวกรรมการวัดคุม
ภาควิชาวิศวกรรมการวัดคุม คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา ๒๕๔๙


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาควิชาวิศวกรรมวัดคุม
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ใบรับรองปริญญาโท

หัวข้อปริญญาโท โปรแกรมการทำงานของลิฟต์ผ่านเครื่องควบคุม PLC
และโปรแกรม VISUAL BASIC สำหรับเฝ้ามอง
LIFT CONTROL BY PLC AND VISUAL BASIC FOR MONITORING

นักศึกษาผู้จัดทำ นายธีรุตม์ วัชรทัศนกุล รหัสนักศึกษา 46012100
นายประเสริฐ เหล่ามงคลชัยศรี รหัสนักศึกษา 46012108
นายภาณุพงศ์ สุรินทรามนต์ รหัสนักศึกษา 46012119

ปริญญา วิศวกรรมศาสตรบัณฑิต
สาขาวิชา วิศวกรรมการวัดคุม
ปีการศึกษา 2549

อาจารย์ผู้ควบคุมปริญญาโท	ลายมือชื่อ
รศ.สุพรรณ กุลพานิชย์	

ภาควิชารับรองแล้ว



(รศ.ประภาพร อุดคภูมิพันธ์)

หัวหน้าภาควิชาวิศวกรรมการวัดคุม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อปริญญานิพนธ์ โปรแกรมการทำงานของลิฟต์ผ่านเครื่องควบคุม PLC และ
โปรแกรม VISUAL BASIC สำหรับเฝ้ามอง
LIFT CONTROL BY PLC AND VISUAL BASIC FOR MONITORING

นักศึกษาผู้จัดทำ นายธีรุตม์ วัชรทัศนกุล รหัสนักศึกษา 46012100
นายประเสริฐ เหล่ามงคลชัยศรี รหัสนักศึกษา 46012108
นายภาณุพงศ์ สุรินทรามนต์ รหัสนักศึกษา 46012119

อาจารย์ที่ปรึกษา รศ.สุพรรณ กุลพานิชย์

ปีการศึกษา 2549

บทคัดย่อ

โครงการนี้เป็นการประยุกต์ใช้ PLC เป็นเครื่องควบคุม โดยมีคอมพิวเตอร์เป็นเทอร์มินอล เพื่อเฝ้ามองการทำงานของระบบ สำหรับระบบที่ใช้ในการศึกษานี้จะเป็นระบบจำลองขนาดเล็กของลิฟต์ เพื่อให้เห็นการทำงานที่ชัดเจน จะมีโปรแกรมอีกส่วนหนึ่งสำหรับใช้เฝ้ามองระบบที่พัฒนาจากโปรแกรม Visual Basic เพื่อให้การเฝ้ามองมีประสิทธิภาพมากขึ้น จะมีการแสดงผลในรูปแบบของภาพเคลื่อนไหวที่แสดงสถานะการทำงาน ตลอดจนผล Alarm กรณีที่ระบบเกิดขัดข้อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

LIFT CONTROL BY PLC AND VISUAL BASIC FOR MONITORING



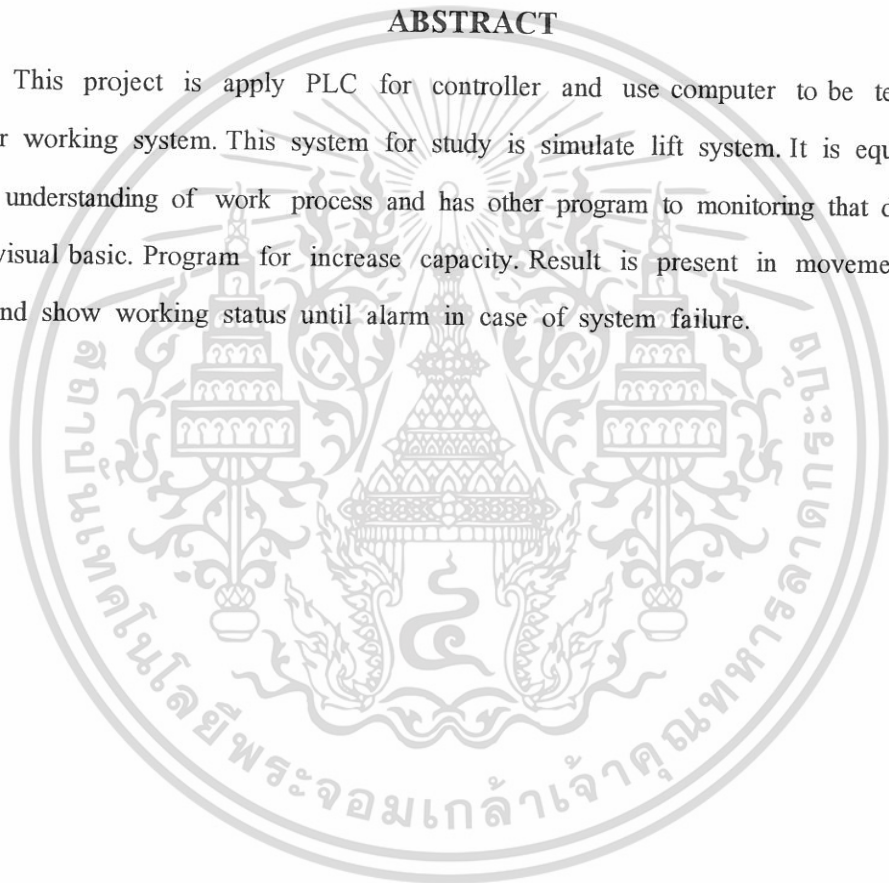
A THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENT FOR THE DEGREE OF
BACHELOR OF ENGINEERING IN INSTRUMENTATION ENGINEERING
DEPARTMENT IN INSTRUMENTATION ENGINEERING
FACULTY OF ENGINEERING
KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
2006

Thesis Title Lift Control By PLC and Visual Basic for Monitoring
Authors Mr.Teerut Watcharatussanakul
Mr.Prasert Laomongkholchaisri
Mr.Panupong Surintramon
Thesis Advisor Assoc.Prof. Suphan Gulpanich
Year 2006

ABSTRACT

This project is apply PLC for controller and use computer to be terminal for monitor working system. This system for study is simulate lift system. It is equipment for clearly understanding of work process and has other program to monitoring that development from visual basic. Program for increase capacity. Result is present in movement pictures form and show working status until alarm in case of system failure.



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

ปริญญานิพนธ์ฉบับนี้สำเร็จลุล่วงได้อย่างดี ด้วยคำแนะนำและคำปรึกษาเกี่ยวกับลำดับขั้นตอนการวิจัยจาก รศ.สุพรรณ กุลพาณิชย์ ซึ่งท่านเป็นอาจารย์ผู้ควบคุมปริญญานิพนธ์และเอื้อเพื่ออุปกรณ์และเครื่องมือในระหว่างทำวิจัย กลุ่มผู้วิจัยรู้สึกซาบซึ้งในความอนุเคราะห์จากท่าน และขอกราบขอบพระคุณท่านเป็นอย่างสูง

ขอขอบพระคุณห้องสมุดและภาควิชาวิศวกรรมศาสตรจารย์วัดคุม คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ที่เป็นแหล่งค้นหาข้อมูลและเป็นสถานที่ทำวิจัย พร้อมทั้งอนุเคราะห์เป็นสถานที่ที่ใช้ทดสอบงานวิจัย จนสำเร็จลุล่วงไปด้วยดี

ขอขอบคุณเพื่อนๆ ห้อง J และคุณเจตติยะ ศรีพิทักษ์ สำหรับคำแนะนำระหว่างทำวิจัยและสอนวิธีการแก้ไขปัญหาเมื่อเจอปัญหา ทั้งปัญหาในเรื่องการเขียนโปรแกรม ปัญหาระบบการทำงานของลิฟท์ และให้ยืมอุปกรณ์และเครื่องมือในระหว่างทำวิจัย

ขอบคุณเพื่อนๆ ห้อง JS ที่คอยเป็นกำลังใจและชวนเล่นกีฬา เมื่อกลุ่มของผู้วิจัยรู้สึกเครียดและขอบคุณทุกๆ คำกำลังใจที่คอยส่งกำลังใจให้งานวิจัยสำเร็จ

สุดท้ายนี้ขอกราบขอบพระคุณ คุณพ่อและคุณแม่ ที่คอยสั่งสอนและอบรมให้เป็นคนดี รวมทั้งเป็นผู้สนับสนุนทุนทรัพย์ในการศึกษาของกลุ่มผู้วิจัย และเป็นกำลังใจตั้งแต่เริ่มต้นทำงานวิจัยจนงานวิจัยสำเร็จลุล่วงไปด้วยดี

คุณค่าและประโยชน์อันพึงมีจากปริญญานิพนธ์ฉบับนี้ กลุ่มผู้วิจัยขอมอบแด่ผู้มีพระคุณทุกท่าน

คณะผู้จัดทำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

	หน้า
บทคัดย่อภาษาไทย.....	I
บทคัดย่อภาษาอังกฤษ.....	II
กิตติกรรมประกาศ.....	III
สารบัญ.....	IV
สารบัญรูป.....	VII
สารบัญตาราง.....	VIII
บทที่ 1 บทนำ.....	1
1.1 ความสำคัญของปริญญานิพนธ์.....	1
1.2 วัตถุประสงค์ของปริญญานิพนธ์.....	1
1.3 ขอบเขตของปริญญานิพนธ์.....	2
1.4 ขั้นตอนการศึกษา.....	2
บทที่ 2 หลักการและทฤษฎี.....	3
2.1 โปรแกรม Visual Basic 6.0.....	3
2.1.1 การเขียนโปรแกรม Visual Basic 6.0 ขั้นพื้นฐาน.....	3
2.1.1.1 ที่มาและลักษณะโปรแกรมเขียนด้วย Visual Basic 6.0.....	3
2.1.1.2 รายละเอียดของส่วนประกอบต่างๆ ของหน้าจอ.....	4
2.1.1.3 จุดเด่นของ Visual Basic.....	5
2.1.1.4 รูปแบบการพัฒนาแอปพลิเคชันกับ Visual Basic.....	5
2.1.1.5 เริ่มสร้างแอปพลิเคชันด้วย Visual Basic	6
2.1.2 การสื่อสาร Communication Port.....	7
2.2 โปรแกรมเมเบิลลอจิกคอนโทรลเลอร์ Programmable Logic Controller (PLC).....	10
2.2.1 โครงสร้างและหลักการทำงานของเครื่องควบคุม PLC.....	10
2.2.2 พื้นที่หน่วยความจำของเครื่องควบคุม PLC.....	13
2.2.3 ความสามารถของ PLC.....	16
2.3 การสื่อสารข้อมูลระหว่างโปรแกรม Visual Basic กับเครื่องควบคุม PLC.....	17
2.3.1 โปรโตคอล.....	17

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นิยมนำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ(ต่อ)

	หน้า
2.3.1.1 ข้อตกลงในการติดต่อสื่อสาร.....	17
2.3.1.2 การควบคุมความผิดพลาดในการส่งข้อมูล.....	19
2.3.1.3 อัตราการส่งข้อมูล.....	20
2.3.2 การสื่อสารข้อมูลแบบอนุกรม.....	21
2.3.2.1 มาตรฐานในการสื่อสารข้อมูลแบบอนุกรม.....	21
2.3.2.2 ข้อตกลงในการสื่อสารของเครื่องควบคุม PLC.....	22
2.3.2.3 คำสั่งและคำตอบสนอง.....	23
บทที่ 3 การสร้างและการออกแบบ.....	25
3.1 ข้อกำหนดการทำงานของลิฟท์.....	26
3.2 วางแผนการปฏิบัติงานในการสร้างโครงการ.....	27
3.3 การออกแบบทางด้านซอฟต์แวร์.....	27
3.3.1 ลักษณะการทำงานของโปรแกรม Visual Basic ในการสื่อสารข้อมูลกับ PLC เพื่อแสดงผลในส่วนของกราฟฟิก.....	27
3.3.2 ลักษณะการทำงานของโปรแกรม Visual Basic ในส่วนที่ใช้ควบคุม PLC.....	28
3.3.3 แอปพลิเคชันแสดงการทำงานของลิฟท์ ที่สร้างจากโปรแกรม.....	29
3.3.4 โปรแกรม Visual Basic และคำสั่งจากเครื่อง PLC ที่เขียนตามข้อกำหนดในการสั่งให้ลิฟท์ทำงานและมีความสัมพันธ์กับ Hardware (ลิฟท์จำลอง).....	31
บทที่ 4 การทดลองและผลการทดลอง.....	32
4.1 ส่วนประกอบทางด้าน Hardware.....	32
4.1.1 Lift.....	32
4.1.2 เครื่องควบคุม PLC ยี่ห้อ OMRON.....	34
4.2 การเชื่อมต่อ.....	34
4.3 ตัวอย่างการทดลอง.....	35

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ(ต่อ)

	หน้า
บทที่ 5 สรุปผลการวิจัยและข้อเสนอแนะ.....	37
5.1 สรุปผลการทดลอง.....	37
บรรณานุกรม.....	38
ภาคผนวก ก ตัวอย่างการพิมพ์ชุดคำสั่งบูตลิน.....	39
ภาคผนวก ข CQM1/CPM1/CPM1A/SRM1 Programmable Controllers Programming Manual (OMRON).....	57



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป

รูปที่	หน้า
2.1 แสดงหน้าจอโปรแกรม Visual Basic 6.0.....	3
2.2 องค์ประกอบสำคัญในโปรแกรม Visual Basic 6.0.....	4
2.3 การ Scan ของ CPU.....	11
2.4 ข้อมูลในบิต ไบท์ และเวิร์ท.....	12
2.5 แสดงรูปแบบการตรวจสอบด้วย Parity Bit.....	19
2.6 การสื่อสารข้อมูลแบบอนุกรม.....	21
2.7 แสดงโครงสร้างของการสื่อสารข้อมูลแบบอนุกรมตามมาตรฐาน RS -232C.....	22
2.8 แสดงรูปแบบของบล็อก (Block Format).....	22
3.1 ข้อกำหนดการทำงานของลิฟต์ แสดงโดย Flowchart.....	27
3.2 หน้า Log in.....	28
3.3 หน้า Menu.....	29
3.4 หน้า Control Lift.....	29
3.5 หน้า Check Bit Status.....	30
3.6 หน้า Credit.....	30
4.1 ลิฟต์จำลองที่มีเครื่อง PLC ยี่ห้อ FACON เป็นตัวสั่งการ.....	32
4.2 Lift.....	33
4.3 Control Panel.....	33
4.4 ปุ่มกดชั้น.....	33
4.5 ส่วนเชื่อมต่อกับ PLC.....	34
4.6 เครื่องควบคุม PLC ยี่ห้อ OMRON.....	34
4.7 แสดงการเชื่อมต่อ.....	35
4.8 กราฟฟีกในโปรแกรม Visual Basic ที่แสดงเมื่อมีการกดลิฟท์ขึ้นชั้น 4 ขณะที่ลิฟท์อยู่ชั้น 3 ในหน้าจอ Check Bit Status.....	35
4.9 แสดงการทำงานของลิฟท์ใน Bit ต่างๆ.....	35
4.10 กราฟฟีกในโปรแกรม Visual Basic ที่แสดงเมื่อมีการกดลิฟท์ขึ้นชั้น 4 ขณะที่ลิฟท์อยู่ชั้น 3 ในหน้าจอ Control Lift	36
4.11 มีการกดลิฟท์ขึ้นชั้น 4 ขณะที่ลิฟท์อยู่ชั้น 3 ส่วน Hardware.....	36

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

ตารางที่	หน้า
2.1 พื้นที่ใช้งานของหน่วยความจำข้อมูล.....	13
2.2 พื้นที่หน่วยความจำในพื้นที่หน่วยความจำ IR.....	14
2.3 แสดงค่า Real Time Clock ของ PLC.....	15
2.4 พื้นที่หน่วยความจำในพื้นที่หน่วยความจำ LR.....	16



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1

บทนำ

1.1 ความสำคัญของปริญญานิพนธ์

เนื่องจากในปัจจุบันเครื่องควบคุมแบบตรรกะเข้ามามีบทบาทในวงการอุตสาหกรรม อาทิ เช่น ใช้ควบคุมเครื่องจักรกลในโรงงานอุตสาหกรรม พวกเราจึงได้มีความคิดที่จะนำคอมพิวเตอร์ มาใช้แสดงผลการทำงานของ PLC ในรูปแบบของกราฟฟิก โดยได้นำโปรแกรม Visual Basic มา เชื่อมต่อกับเครื่องควบคุม PLC โดยโครงงานของเราได้ยกตัวอย่างการทำงานของ PLC ที่ควบคุม ลิฟท์ ซึ่งโปรแกรม Visual Basic จะทำหน้าที่แสดงผลการทำงานของลิฟท์ และยังสามารถควบคุม ลิฟท์ผ่านการทำงานของ PLC เนื่องจากเราสามารถจะรู้ค่า Bit การทำงานของ PLC จึงทำให้เราสามารถที่จะดูส่วนที่มีปัญหาผ่านทางหน้าจอคอมพิวเตอร์ทำให้สามารถแก้ไขปัญหาได้อย่างรวดเร็ว และถูกต้อง ซึ่งเป็นการช่วยเพิ่มประสิทธิภาพในการควบคุมของ PLC ทำให้ระบบการทำงานของ ลิฟท์ดียิ่งขึ้น

นอกจากนี้แล้วเรายังสามารถใช้โปรแกรม Visual Basic มาใช้แสดงกราฟฟิกและควบคุม การทำงานของระบบต่างๆ อีกมากมาย

1.2 วัตถุประสงค์ของปริญญานิพนธ์

1. สามารถควบคุมการทำงานของลิฟท์โดยเครื่องควบคุม PLC
2. ทำการเชื่อมต่อระหว่างโปรแกรม Visual Basic กับเครื่องควบคุม PLC ที่ควบคุม Hardware
3. สามารถนำค่าที่ได้จากการสื่อสารข้อมูลระหว่างโปรแกรม Visual Basic ที่ควบคุม เครื่องควบคุม PLC มาแสดงผลในรูปแบบกราฟฟิก

1.3 ขอบเขตของปริญญาโท

1. ศึกษาการลักษณะการทำงานของลิฟท์ที่มี เครื่อง PLC เป็นตัวควบคุม
2. ศึกษาการใช้โปรแกรม Visual Basic เพื่อแสดงผลในรูปแบบกราฟฟิกการทำงานของลิฟท์
3. ศึกษาเฉพาะการเชื่อมต่อระหว่างโปรแกรม Visual Basic เครื่องควบคุม PLC ยี่ห้อ OMRON
4. ขอบเขตการเฝ้ามองและควบคุมค่าเครื่องควบคุม PLC ของโปรแกรม Visual Basic คือตั้งแต่ Word ที่ 000.00 – 010.00

1.4 ขั้นตอนการศึกษา

ในส่วนของขั้นตอนการศึกษานั้นแบ่งออกเป็น 5 ส่วนหลัก ๆ คือ

ส่วนแรก ศึกษาการลักษณะขั้นตอนการทำงานของลิฟท์

ส่วนที่สอง ศึกษาการเขียนโปรแกรม Visual Basic และสร้างแอปพลิเคชันที่แสดงการจำลองการทำงานของลิฟท์

ส่วนที่สาม ศึกษาการใช้เครื่องควบคุม PLC ยี่ห้อ OMRON

ส่วนที่สี่ ศึกษาการเชื่อมต่อระหว่างโปรแกรม Visual Basic กับเครื่องควบคุม PLC

ส่วนที่ห้า ศึกษาการควบคุมลิฟท์ที่แสดงจากโปรแกรม Visual Basic ให้สัมพันธ์กับ Hardware (ลิฟท์จำลอง)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2

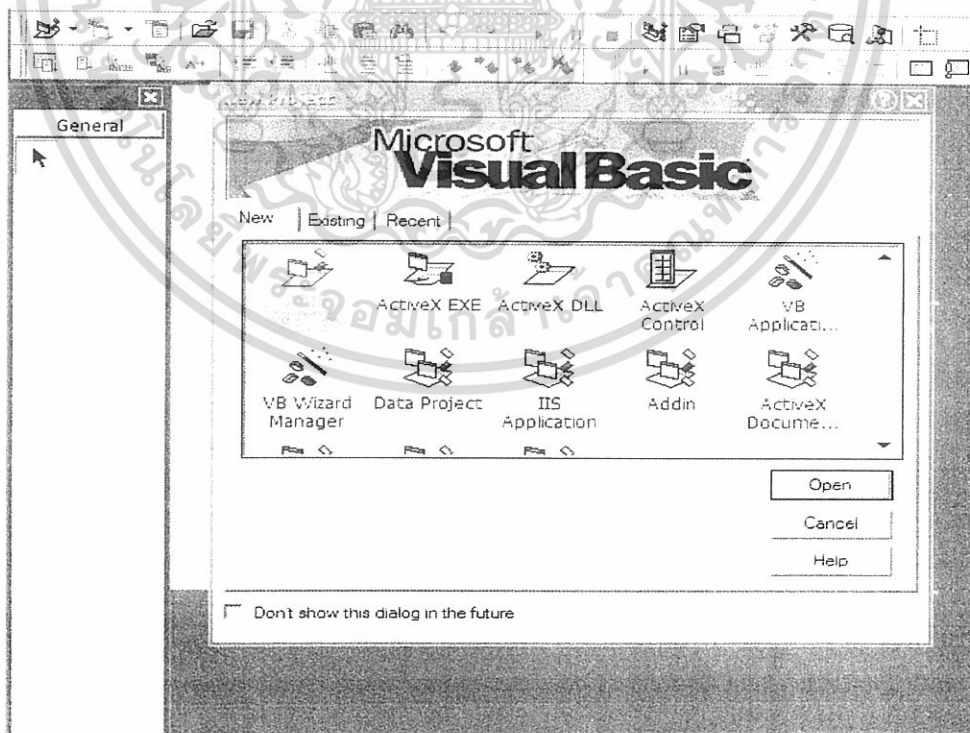
หลักการและทฤษฎี

2.1 โปรแกรม Visual Basic 6.0

2.1.1 การเขียนโปรแกรม Visual Basic 6.0 ขั้นพื้นฐาน

2.1.1.1 ที่มาและลักษณะของโปรแกรม Visual Basic 6.0

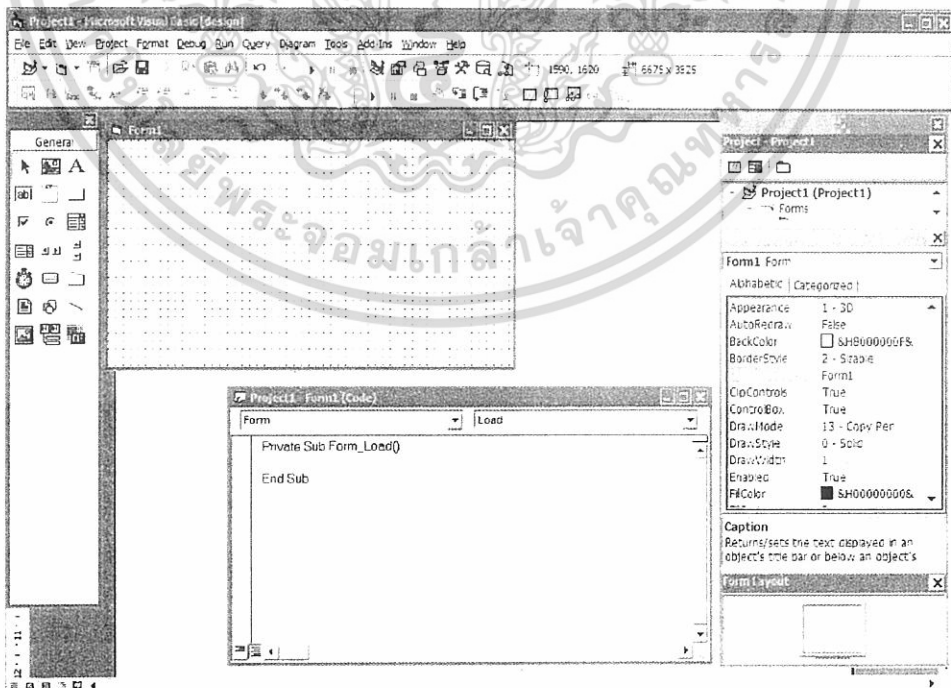
Visual Basic เป็นเครื่องมือที่ใช้พัฒนาโปรแกรมแบบ Visual Programming ซึ่ง Visual Programming เป็นวิธีการเขียนโปรแกรมที่มีเครื่องมือช่วยพัฒนาโปรแกรมได้ง่าย โดยโปรแกรมที่สร้างจะมีลักษณะเหมือนตอนออกแบบหน้าจอวิธีการพัฒนาโปรแกรมก็ง่ายเพียงออกแบบหน้าจอที่ต้องการ กำหนดคุณสมบัติและเขียนโค้ดกำกับ ซึ่งจะช่วยให้การพัฒนาโปรแกรมสามารถทำได้อย่างรวดเร็ว สำหรับการใช้งาน โปรแกรม Visual Basic นั้นไม่ได้จำกัดตัวเองอยู่เพียงแค่บน PC ที่ใช้วินโดวส์เท่านั้นแต่ยังขยายความสามารถไปยังการใช้งานบน Internet รวมทั้งอุปกรณ์ที่ใช้เทคโนโลยีไร้สาย



เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ 2.1 แสดงหน้าจอโปรแกรม Visual Basic 6.0
การนำเอกสารนี้ไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.1.2 รายละเอียดของส่วนประกอบต่างๆของหน้าจอ

1. Menu Bar เก็บคำสั่งที่สามารถใช้งานได้ทั้งหมดใน Visual Basic 6.0 ประกอบด้วยเมนูทำงาน File, View, Windows
2. Toolbar ประกอบด้วยปุ่มคำสั่งต่างๆ ที่ช่วยให้งานคำสั่งของ Visual Basic 6.0 ได้อย่างรวดเร็ว
3. Toolbox เป็นกล่องเก็บ ActiveX Control ซึ่งเราจะนำมาประกอบเป็นส่วนต่างๆของแอปพลิเคชัน
4. Project Explorer เป็นเครื่องมือที่ใช้ควบคุมการทำงานของโปรเจกต์
5. Properties windows เป็นส่วนที่กำหนดพร็อพเพอร์ตี้ให้กับออบเจกต์ต่างๆ ใน แอปพลิเคชัน
6. Form Layout เป็นรูปหน้าต่างคร่าวๆ ของฟอร์มที่ได้จากการรันแอปพลิเคชัน ทำให้เราทราบตำแหน่งที่ปรากฏบนจอภาพเมื่อแอปพลิเคชันทำงาน
7. Form Designer ซึ่งเป็นส่วนที่เรามองเห็นในขณะออกแบบแอปพลิเคชันของ Visual Basic 6.0 ซึ่งเราออกแบบหน้าต่างของแอปพลิเคชันผ่านฟอร์มดีไซน์เนอร์
8. Code Window เป็นส่วนที่เราทำการเขียนโปรแกรมเพื่อควบคุมการทำงานของแอปพลิเคชัน



รูปที่ 2.2 องค์ประกอบสำคัญในโปรแกรม Visual Basic 6.0

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.1.3 จุดเด่นของ Visual Basic

1. สร้างแอปพลิเคชันได้ง่ายและรวดเร็ว

Visual Basic ได้รับการวางตัวให้เป็นเครื่องมือที่ช่วยให้สร้างแอปพลิเคชันได้อย่างรวดเร็วและง่ายดาย เพื่อลดเวลาในการสร้างแอปพลิเคชันให้สั้นลง ซึ่งเรียกรูปแบบนี้ว่า Rapid Application Development หรือ RAC ทั้งนี้เพราะมีการขจัดงานที่โปรแกรมเมอร์ต้องทำซ้ำๆ ซากๆ ออกไปเหลือเฉพาะที่ต้องโฟกัสเกี่ยวกับปัญหาของงานจริงๆ แล้วเขียนโปรแกรมจัดการปัญหานั้นๆ ส่วนเรื่องอื่นๆ เหลือให้ Visual Basic จัดการ

2. การเขียนโปรแกรมที่ง่ายต่อการเรียนรู้

ถ้ามีโอกาสเขียนโปรแกรมด้วย Visual Basic แล้วจะเห็นว่า ภาษา Basic ที่ใช้ใน Visual Basic นั้นเข้าใจง่ายคือใกล้เคียงกับภาษาที่เราใช้งานปกติ สื่อความหมายและเข้าใจง่ายกว่าภาษาโปรแกรมอื่นๆ ทำให้ผู้เริ่มต้นเขียนโปรแกรมทำความเข้าใจกับการเขียนโปรแกรมได้อย่างรวดเร็ว

3. รวมเครื่องมืออำนวยความสะดวกในการเขียนโปรแกรม

นอกจากจะง่ายต่อการเรียนรู้แล้ว Visual Basic ยังมีเครื่องมือที่ช่วยในการเขียนโปรแกรมเป็นสิ่งที่ไม่ยุ่งยาก เพราะจะมีเครื่องมือที่ช่วยให้ไม่ต้องจดจำไวยากรณ์ภาษาที่ยุ่งยาก มีการตรวจสอบอัตโนมัติว่าโปรแกรมที่เขียนนั้นถูกต้องตามหลักภาษาหรือไม่มีการแยกแยะส่วนของโปรแกรมอย่างเป็นระเบียบ ทำให้งานของโปรแกรมเมอร์นั้นลดลงได้มาก นอกจากนี้ยังมีเครื่องมือช่วยในการเขียนโปรแกรมแล้วยังมีเครื่องมือที่ใช้ในการทดสอบแก้ไขโปรแกรม (Debugger) ที่เขียนขึ้นมาว่าทำงานได้ถูกต้องหรือไม่ มีระบบขอความช่วยเหลือ (Online Help) ไว้อ้างอิงและขอความช่วยเหลือในจุดที่เราสงสัยข้อใจได้

เครื่องมือทั้งหมดที่กล่าวมาถูกจัดรวมไว้ในสภาพแวดล้อมการทำงานเดียวกัน (เรียกย่อๆ ว่า IDE ซึ่งย่อมาจาก Integral Development Environment) ทำให้เรียกใช้งานได้สะดวกตั้งแต่เขียนโปรแกรม ทดสอบแก้ไข สร้างชุดติดตั้ง รวมทั้งระบบขอความช่วยเหลือ ซึ่งเราสามารถเพิ่มเติมเครื่องมือชนิดใหม่ๆ เข้าไปได้เรื่อยๆ หรือถอดเครื่องมือที่ไม่จำเป็นต้องใช้เพื่อประหยัดพื้นที่ฮาร์ดดิสก์ก็ได้เช่นกัน

2.1.1.4 รูปแบบการพัฒนาแอปพลิเคชันกับ Visual Basic

เมื่อเรามองเห็นว่า Visual Basic จะช่วยให้เราสร้างแอปพลิเคชันบน Windows ได้ง่ายและรวดเร็วแล้ว ยังมีรูปแบบที่ Visual Basic สามารถสร้างขึ้นมาได้อีกคือ

1. พัฒนาแอปพลิเคชันกับ Active X Control

เทคโนโลยีที่มีชื่อว่า Active X เป็นตัวอยู่เบื้องหลังความสำเร็จของ Visual Basic ซึ่งช่วยลดงานที่ซ้ำซ้อนของการเขียนโปรแกรมลงไปมาก

ตัวอย่าง เช่น การเขียนโปรแกรมเพื่อรับค่าข้อมูลของผู้ใช้แต่ก่อนเราต้องเขียนโปรแกรมเพื่อวาดหน้าจอ เขียนโปรแกรมวาดรูปปุ่มและช่องรับข้อความรวมทั้งการเขียนโปรแกรมเพื่อจัดการกับข้อมูลที่ใช้ป้อนเข้ามาแต่ Active X จะทำให้เราสนใจเฉพาะการจัดการกับข้อมูลที่ใช้ป้อนเข้ามาเท่านั้นส่วนที่เหลือจะจัดการให้เองโดยช่องรับข้อความและปุ่มต่างๆจะใช้ Active X Control จัดการ

นอกจากจะลดความซับซ้อนลงแล้วการใช้ Active X Control ในการเขียนโปรแกรมยังช่วยให้โปรแกรมที่เราเขียนกับโปรแกรมที่คนอื่นๆ เขียนนั้นตั้งอยู่บนมาตรฐานเดียวกัน ทำให้การบำรุงรักษาง่ายเพราะใครๆ ก็เข้าใจมาตรฐานของ Active X Control นี้ ทำให้ไม่ต้องกังวลใจว่าโปรแกรมที่เขียนขึ้นจะมีเฉพาะคนเขียนเท่านั้นที่เข้าใจ

2. สร้างแอปพลิเคชันที่ใช้งานกับฐานข้อมูล

เป็นแอปพลิเคชันที่มีการใช้งานมากที่สุดเพราะระบบต่างๆ เช่น ร้านค้า คลังสินค้า ระบบบัญชี ระบบบริหารงานบุคคลหรือแม้แต่ E-Commerce ทั้งหมดต่างก็ต้องมีส่วนที่ติดต่อกับฐานข้อมูล Visual Basic ได้ช่วยในการสร้างแอปพลิเคชันกับฐานข้อมูลเป็นเรื่องที่ทำได้โดยง่าย เพราะมีเครื่องมือต่างๆ ที่คอยให้ความสะดวกในการเขียนโปรแกรมเพื่อใช้งานข้อมูลจากฐานข้อมูลซึ่งไม่จำกัดด้วยว่าจะเห็นข้อมูลแบบใดทั้งฐานข้อมูลส่วนบุคคล ฐานข้อมูลผ่านเครือข่ายหรือฐานข้อมูลผ่านอินเทอร์เน็ต จากความสามารถที่หลากหลายนี้ Visual Basic จึงเป็นตัวเลือกอันดับต้นๆ ของการสร้างแอปพลิเคชันที่เกี่ยวกับงานฐานข้อมูลในโลกธุรกิจ

3. สร้างแอปพลิเคชันแบบใหม่กับอินเทอร์เน็ต

อินเทอร์เน็ตนับว่ามีความสำคัญกับชีวิตของคนที่ใช้ไอทีมากขึ้นทุกวัน ซึ่ง Visual Basic เปิดโอกาสให้เราสามารถสร้างแอปพลิเคชันเพื่อรองรับการทำงานร่วมกับอินเทอร์เน็ตได้ด้วยการใช้ความรู้เดิมที่เราเคยได้จากการสร้างแอปพลิเคชันปกติ รวมทั้งเครื่องมือเสริมการทำงานต่างๆ มากมาย

จุดเด่นของ Visual Basic อีกข้อหนึ่งคือเปิดโอกาสให้เรานำแอปพลิเคชันปกติที่เดิมทำงานกับคอมพิวเตอร์ส่วนบุคคลมาดัดแปลงใช้งานกับอินเทอร์เน็ตได้อย่างไม่ยากเย็นนัก ทำให้ไม่ต้องเปลี่ยนแปลงรูปแบบงานเดิม

2.1.1.5 เริ่มสร้างแอปพลิเคชันด้วย Visual Basic

1. ออกแบบแอปพลิเคชัน

ต้องทราบให้แน่ชัดก่อนว่าแอปพลิเคชันที่เราจะทำการสร้างนั้น จะนำไปใช้ประโยชน์อะไรต้องมีความสำคัญอะไรบ้าง และต้องมีรูปร่างหน้าตาเป็นอย่างไร ซึ่งจำเป็นอย่างยิ่งที่ต้องคิดให้รอบคอบและเขียนออกมาให้ชัดเจนโดยอาจจะร่างง่ายๆ ในกระดาษ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. ตกแต่งหน้าต่างแอปพลิเคชัน

เป็นการตกแต่งรูปร่างของแอปพลิเคชันที่ได้ออกแบบไว้แล้วพร้อมๆ กับการคำนวณค่าพรีออพเพอร์ตี้ต่างๆ ให้กับคอนโทรลแต่ละตัวในแอปพลิเคชัน

3. เขียนโค้ดกำกับแอปพลิเคชัน

หลังจากตกแต่งหน้าต่างเสร็จแล้วขั้นต่อไปคือการเขียนโค้ดหรือการเขียนโปรแกรมเพื่อควบคุมการทำงานต่างๆ ซึ่งเราจะใช้การเขียนโปรแกรมแบบ Event Driven Programming ซึ่งจะเป็นการเขียนโค้ดเพื่อรองรับกับเหตุการณ์ต่างๆ ที่จะเกิดขึ้นกับคอนโทรลในแอปพลิเคชันของเรา

4. ทดสอบการทำงานของแอปพลิเคชัน

เมื่อเขียนโค้ดเสร็จแล้วก็ถึงเวลาที่จะทดสอบการทำงานของแอปพลิเคชันที่เราสร้างขึ้นซึ่งประกอบไปด้วยคอนโทรลต่างๆ

5. บันทึกเก็บไว้ในคอมพิวเตอร์

หลังจากทดสอบจนแน่ใจว่าแอปพลิเคชันที่สร้างนั้นทำงานได้ถูกต้อง เราจึงบันทึกเก็บไว้ ซึ่งสามารถแก้ไขและเพิ่มเติมความสามารถอื่นๆ ได้ภายหลัง

6. สร้างไฟล์ .EXE (Make)

เมื่อเราสร้างแอปพลิเคชันเสร็จแล้ว หากเราต้องการนำแอปพลิเคชันที่สร้างขึ้นให้เรียกใช้งานได้เอง โดยไม่ต้องเรียกผ่าน Visual Basic หรือต้องการนำไปใช้งานในเครื่องคอมพิวเตอร์เครื่องอื่นๆ เราสามารถทำได้โดยสร้างไฟล์เอกซ์คิวต์ (ไฟล์ที่มีนามสกุลเป็น .EXE)

2.1.2 การสื่อสารกับ Communication Port

Visual Basic ไม่สามารถจัดการกับ Hardware ได้โดยตรงเหมือนภาษา C หรือภาษาอื่นๆ อดี Visual Basic ได้เตรียม Tool และ Control ต่างๆ สำหรับจัดการกับ Hardware โดยเฉพาะไว้ให้ผู้เขียนโปรแกรมเท่าที่จำเป็น ในการทำงานกับพอร์ตสื่อสารบนเครื่อง PC ซึ่ง Visual Basic มี Control ชื่อ Microsoft Comm Control สำหรับการรับส่งข้อมูลทาง Port อนุกรม

การใช้ MsComm Control

MsComm จัดเตรียมทางเลือกเอาไว้ 2 ทางเพื่อความสะดวกในการสื่อสารข้อมูล ทางแรกคือ การสื่อสารข้อมูลที่กระตุ้นด้วยเหตุการณ์ (event-driven communication) เป็นรูปแบบการใช้งานที่มีประสิทธิภาพมากสำหรับตอบสนองแบบทันทีทันใด เช่น เมื่อตัวอักษรถูกส่งมาที่พอร์ตอนุกรมหรือเกิดการเปลี่ยนแปลงที่ขา Data Carrier Detect (DCD) หรือขา Request To Send เหตุการณ์ OnComm ของ MsComm จะสามารถตรวจจับสัญญาณนั้นได้ทันที ส่วนทางเลือกที่สองเป็นการคอยตรวจสอบค่าเหตุการณ์และความผิดพลาดที่เกิดขึ้นด้วยค่าที่เปลี่ยนแปลงใน

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่ออนุญาตให้เผยแพร่จะระบุชื่อต้นฉบับไว้
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คุณสมบัติ CommEvent หลังจากให้โปรแกรมทำงานฟังก์ชันต่างๆ ไปเรียบร้อยแล้ว ซึ่งวิธีนี้ใช้งานได้ดีในกรณีที่โปรแกรมขนาดเล็ก

CommPort

ใช้ในการกำหนดและอ่านค่าพอร์ตอนุกรมที่ติดต่อยู่ (Com 1, Com 2, Com 3, Com 4)

รูปแบบการใช้งาน

Object.CommPort [= value]

โดย Value เป็นค่าของพอร์ตอนุกรม ชนิดของข้อมูลเป็น Integer ค่า Value สามารถกำหนดได้ในช่วง 1- 16 (ค่าเริ่มต้นกำหนดไว้ที่ 1) เมื่อมีการกำหนดค่าแล้วทำการเปิดพอร์ตโดยใช้คุณสมบัติ PortOpen แต่ว่าพอร์ตนั้นไม่มีอยู่ในระบบ MsComm จะสร้างสัญญาณแสดงข้อผิดพลาด error 68 ขึ้นมา ซึ่งหมายถึง อุปกรณ์ตัวนี้ไม่มีอยู่ในระบบ ดังนั้นการเขียนโปรแกรมจึงจำเป็นต้องกำหนดตำแหน่งของพอร์ตอนุกรม ก่อนที่ใช้คำสั่ง OpenPort

Setting

ใช้ในการกำหนดและอ่านค่าอัตราบอด, พาริตี, จำนวนของบิตข้อมูล, จำนวนของบิตปิดท้าย รูปแบบการใช้งาน

Object.setting [= value]

ค่า Value มีชนิดของข้อมูลเป็นแบบ String มีรูปแบบเป็น “BBBB,P,D,S” โดย BBBB เป็นค่าอัตราบอด, P เป็นค่าพาริตีบิต, D เป็นจำนวนของบิตข้อมูลและ S เป็นจำนวนของบิตปิดท้ายปกติแล้วค่านี้ถูกกำหนดไว้เป็น “9600,N,8,1”

ค่าบอดเรตมาตรฐานที่ใช้กับ MSComm มีดังนี้

110	บิตต่อวินาที
300	บิตต่อวินาที
600	บิตต่อวินาที
1,200	บิตต่อวินาที
2,400	บิตต่อวินาที
9,600	บิตต่อวินาที
14,400	บิตต่อวินาที
19,200	บิตต่อวินาที
28,800	บิตต่อวินาที
38,400	บิตต่อวินาที
56,000	บิตต่อวินาที
128,000	บิตต่อวินาที
256,000	บิตต่อวินาที

เอกสารนี้เป็นเอกสารสงวนไว้สำหรับใช้เรียนเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ค่ามาตรฐานในการกำหนดค่าพาริตีมีดังนี้

สัญลักษณ์	รายละเอียด
E	พาริตีคู่ (EVEN)
M	ลอจิก "1" (MARK)
N	ไม่ใช่ (ค่าปกติ)
O	พาริตีคี่ (Odd)
S	ลอจิก "0" (Space)

ค่าที่ใช้ในการกำหนดจำนวนบิตมี 5 ค่า คือ 4, 5, 6, 7 และ 8 (เป็นค่าปกติ)

ค่าที่ระบุจำนวนบิตปิดท้ายมี 3 ค่า คือ 1 (เป็นค่าปกติ), 1.5 และ 2

PortOpen

ใช้ในการกำหนดและอ่านค่าสถานะของพอร์ตอนุกรม เพื่อเปิดและปิดพอร์ตอนุกรม

รูปแบบการใช้งาน

Object.PortOpen [=Value]

ค่า Value มีชนิดข้อมูลเป็นแบบบูลีนคือ True กับ False โดย True หมายถึงการเปิดพอร์ตอนุกรมและ False หมายถึงการปิดพอร์ตอนุกรม สำหรับการปิดพอร์ตอนุกรม สำหรับการปิดพอร์ตนั้นจะมีการเคลียร์บัฟเฟอร์รับข้อมูลและบัฟเฟอร์ส่งข้อมูลด้วย คอนโทรล MSComm จะปิดพอร์ตอนุกรมโดยอัตโนมัติเมื่อออกจากโปรแกรม ก่อนที่ใช้คุณสมบัติ PortOpen ต้องตรวจสอบให้แน่ใจก่อนว่าคุณสมบัติ CommPort นั้นได้ทำการกำหนดตำแหน่งของพอร์ตอนุกรมไว้ถูกต้องหรือไม่ มิเช่นนั้น MSComm จะแสดงผิดพลาด error 68 แจ้งแก่ผู้ใช้งาน หรือพอร์ตอนุกรมนั้นถูกเปิดเอาไว้แล้ว โปรแกรมก็จะแจ้งข้อผิดพลาดออกเช่นกัน

ถ้าคุณสมบัติ DTREnable หรือ RTSEnable ถูกกำหนดให้เป็น True ก่อนที่จะทำการเปิดพอร์ตค่าคุณสมบัตินี้ของ DTREnable หรือ RTSEnable จะถูกเซตเป็น False หลังจากปิดพอร์ต แต่ถ้าเซตเป็น False หลังจากปิดโปรแกรมแล้ว ค่าที่กำหนดไว้จะเป็นเท่าเดิม

การเปิด Serial Port

การปิดพอร์ตสื่อสารด้วย MsComm Control สามารถทำได้ง่ายๆ ด้วยการกำหนด Properties ดังนี้

MSComm 1. CommPort = 4

MSComm 1. Settings = "19,200, N, 8, 1"

MSComm 1. PortOpen = True

Input เป็นการอ่านข้อมูลจากบัฟเฟอร์ พร้อมทั้งลบข้อมูลในบัฟเฟอร์ด้านรับทิ้ง ซึ่งสามารถอ่านค่าได้เฉพาะในขณะที่ทำงานเท่านั้น รูปแบบการใช้งานคือ

Object.Input

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.1.1 หน่วยประมวลผลกลางหรือ CPU ประกอบด้วย 3 ส่วนคือ

1. หน่วยประมวลผลของ CPU

หน่วยประมวลผลของ CPU ทำหน้าที่ควบคุมและดูแลการทำงานของระบบทั้งหมด โดยรับข้อมูลมาจากหน่วยอินพุตและส่งข้อมูลสุดท้ายที่ได้จากการประมวลผลไปยังหน่วยเอาต์พุต เรียกว่า การสแกน (Scan) ซึ่งใช้เวลาจำนวนหนึ่ง เรียกว่า เวลาสแกน (Scan Time) เวลาในการสแกนแต่ละรอบใช้เวลาประมาณ 0.001 ถึง 0.1 วินาที ทั้งนี้ขึ้นอยู่กับข้อมูลและความยาวของโปรแกรมหรือจำนวนอินพุตและเอาต์พุต หรือจำนวนอุปกรณ์ที่ต่อจาก PLC เช่น จอภาพ เครื่องพิมพ์ ฯลฯ อุปกรณ์เหล่านี้จะทำให้เวลาในการสแกนยาวนานขึ้น การเริ่มต้นการสแกนเริ่มจากรับค่าของสถานะของอุปกรณ์จากหน่วยความจำ (Memory) เสร็จแล้วจะทำการปฏิบัติการตามโปรแกรมที่เขียนไว้ทีละคำสั่งจากหน่วยความจำนั้นจนสิ้นสุด แล้วส่งไปที่หน่วยเอาต์พุต



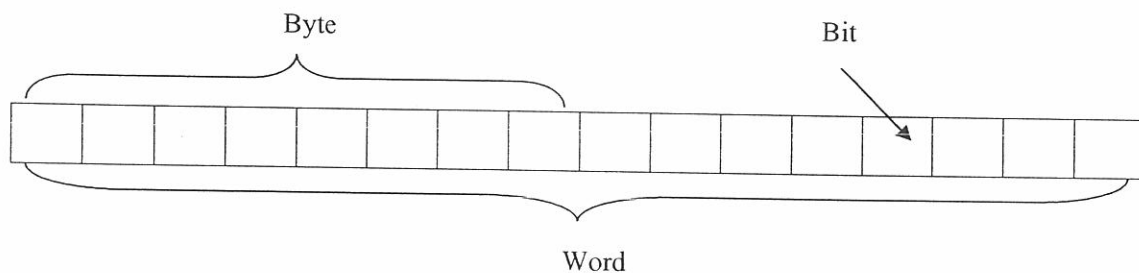
รูปที่ 2.3 การ Scan ของ CPU

ช่วงเวลา Scan ของ CPU จะทำให้ทราบถึงความสามารถในการตอบสนองต่อการเปลี่ยนแปลงของอินพุต/เอาต์พุตของ PLC ว่ารวดเร็วเพียงไร ช่วงเวลา Scan จึงเป็นองค์ประกอบที่สำคัญอย่างหนึ่งในการเลือกใช้ PLC ให้เหมาะสมกับงาน เช่น ถ้าต้องการที่ตรวจสอบสัญญาณอินพุตที่เปลี่ยนค่าสถานะถึง 2 ครั้งภายใน 0.008 วินาที การที่ใช้ PLC ที่ซึ่งมีช่วงเวลา Scan เท่ากับ 0.01 จะไม่สามารถตรวจสอบค่าสถานะที่ถูกต้อง และอาจทำให้เครื่องจักรหรือกระบวนการที่ควบคุมทำงานผิดพลาดได้ และหน่วยประมวลผลยังทำหน้าที่ในการติดต่อกับระบบย่อย ตรวจสอบความผิดพลาดต่างๆ

2. หน่วยความจำ

หน่วยความจำเป็นที่เก็บโปรแกรมและข้อมูลที่หน่วยประมวลผลใช้ในการควบคุมการทำงานของ PLC และการทำงานตามโปรแกรมคำสั่งของผู้ใช้ขนาดของหน่วยความจำจะแบ่งออกเป็นบิตข้อมูลภายในหน่วยความจำ 1 บิต จะมีค่าสถานะทางลอจิก “1” หรือ “0” ข้อมูลขนาด 8 บิต และ 16 บิต รวมกันเรียกว่าไบต์ (Byte) และเวิร์ด (Word)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.4 ข้อมูลเป็นบิต ไบท์ และเวิร์ด

3. หน่วยจ่ายกำลัง

หน่วยจ่ายกำลังจะทำหน้าที่จ่ายและรักษาระดับแรงดันไฟฟ้ากระแสตรงให้กับหน่วยประมวลผล หน่วยความจำ และหน่วยอินพุต/เอาต์พุต ตามความต้องการและทำหน้าที่เตือนให้หน่วยประมวลผลทราบเมื่อเกิดปัญหา

2.2.1.2 หน่วยอินพุตและเอาต์พุต (I/O Unit)

ส่วนของอินพุตและเอาต์พุต (I/O Unit) จะต่อร่วมกับชุดควบคุมเพื่อรับสถานะและสัญญาณต่างๆ เช่น หน่วยอินพุตรับสัญญาณหรือสถานะแล้วส่งไปยัง CPU เพื่อประมวลผล เมื่อ CPU ประมวลผลแล้วจะส่งให้ส่วนของเอาต์พุต เพื่อให้อุปกรณ์ทำงานตามโปรแกรมที่กำหนดไว้ สัญญาณอินพุตจากภายนอกที่เป็นสวิตช์และตัวตรวจจับชนิดต่างๆ จะถูกแปลงให้เป็นสัญญาณที่เหมาะสมถูกต้อง ไม่ว่าจะเป็น AC หรือ DC เพื่อส่งให้ CPU ดังนั้นสัญญาณเหล่านี้จึงต้องมีความถูกต้อง ไม่เช่นนั้นแล้ว CPU จะเสียหายได้

สัญญาณอินพุตที่ดีจะต้องมีคุณสมบัติและหน้าที่ดังนี้

- ทำให้สัญญาณเข้า ได้ระดับที่เหมาะสมกับ PLC
- การส่งสัญญาณระหว่างอินพุตกับ CPU จะติดต่อกันด้วยลำแสง ซึ่งจะอาศัยอุปกรณ์ประเภทโฟโตทรานซิสเตอร์เพื่อต้องการแยกสัญญาณ (Isolate) ทางไฟฟ้าให้ออกจากกัน เป็นการป้องกันไม่ให้ CPU เสียหายเมื่ออินพุตเกิดลัดวงจร

- หน้าสัมผัสจะต้องไม่สั่นสะเทือน (Contact Chattering)

ส่วนของเอาต์พุต จะทำหน้าที่รับค่าสถานะที่ได้จากการประมวลผลของ CPU แล้วนำค่าเหล่านี้ไปควบคุมอุปกรณ์ทำงานเช่น รีเลย์ โซลินอยด์ หรือหลอดไฟ เป็นต้น

นอกจากนั้นแล้วยังทำหน้าที่แยกสัญญาณของหน่วยประมวลผล (CPU) ออกจากอุปกรณ์เอาต์พุต โดยปกติเอาต์พุตนี้จะมีความสามารถขับโหลดด้วยกระแสไฟฟ้าประมาณ 1-2 แอมแปร์ แต่ถ้าโหลดต้องการกระแสไฟฟ้ามากกว่านี้ จะต้องต่อเข้ากับอุปกรณ์ขับอื่นเพื่อขยายให้รับกระแสไฟฟ้ามากขึ้น เช่น รีเลย์หรือคอนแทคเตอร์ u3648 เป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ส่วนป้อนโปรแกรม (Programming Unit)

มีหน้าที่คือควบคุมโปรแกรมของผู้ใช้ลงในหน่วยความจำของ PLC นอกจากนั้นแล้วยังทำหน้าที่ติดต่อระหว่างผู้ใช้กับ PLC เพื่อให้ผู้ใช้สามารถตรวจการปฏิบัติงานของ PLC และผลการควบคุมเครื่องจักรและกระบวนการตามโปรแกรมควบคุมที่ผู้ใช้เขียนขึ้นอีกด้วย แบ่งออกได้หลายชนิด คือ

- เครื่องป้อนโปรแกรมแบบ CRT
- เครื่องป้อนโปรแกรมขนาดเล็ก (Mini Programmer)
- เครื่องป้อนโปรแกรมลงในหน่วยความจำของ PLC หรือเทป
- เครื่องป้อนโปรแกรมลงในหน่วยความจำ (Memory Burner)
- คอมพิวเตอร์

2.2.2 พื้นที่หน่วยความจำของเครื่องควบคุม PLC

แบ่งหน่วยความจำเป็น 2 ส่วน คือ

1. หน่วยความจำที่เก็บโปรแกรม (User Program Memory)
2. หน่วยความจำที่ใช้เก็บข้อมูล (Data Memory Area)

หน้าที่ของหน่วยความจำข้อมูลของ PLC หน่วยความจำข้อมูลแต่ละพื้นที่จะมีหน้าที่แตกต่างกัน

ตารางที่ 2.1 พื้นที่ใช้งานของหน่วยความจำข้อมูล

พื้นที่หน่วยความจำ
Internal Relay (IR)
Special Relay (SR)
Temporary Relay (TR)
Holding Relay (HR)
Auxiliary Relay (AR)
Link Relay (LR)
Time/Counter (TC)
Data Memory (DM)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- Internal Relay (IR) แบ่งพื้นที่ออกเป็นส่วนต่างๆ ได้แก่ ส่วนของอินพุท/เอาต์พุทเป็นพื้นที่ที่ใช้กำหนดตำแหน่งให้กับหน่วยอินพุท/เอาต์พุทแบบต่างๆที่นำมาติดตั้ง เช่น หน่วยอินพุท/เอาต์พุทแบบธรรมดา, หน่วยอินพุท/เอาต์พุทแบบพิเศษ เป็นต้น และนอกจากนั้นยังมีส่วนของ Work Area ซึ่งใช้เป็นอินพุท/เอาต์พุทภายใน PLC เรียกว่าเป็น Work Bit ซึ่งสามารถใช้งานได้โดยอิสระในโปรแกรม แต่ไม่สามารถใช้อ้างอิงกับตำแหน่งที่ต่อกับอุปกรณ์อินพุท/เอาต์พุทจริง

ตารางที่ 2.2 พื้นที่หน่วยความจำในพื้นที่หน่วยความจำ IR

พื้นที่หน่วยความจำ IR	เวิร์ด	หน้าที่
I/O Area	IR 000 – IR 029	จองไว้ให้หน่วยอินพุทแบบธรรมดาที่ติดตั้งอยู่ในซีพียูแร็คและเอ็กซ์แพนชันแร็คใช้งาน
Group- 2 High-density I/O Unit and B7A Interface Unit Area	IR 030 – IR 049	จองไว้ให้หน่วยอินพุท/เอาต์พุทแบบที่มีความหนาแน่นของอินพุท/เอาต์พุทต่อหนึ่งยูนิตมากกว่า 16 จุด และหน่วยอินพุท/เอาต์พุทที่สามารถต่อกับอุปกรณ์ระยะไกลใช้งาน
SYSMAC Bus and Compo Bus/D Output Area	IR 050 – IR 099	จองไว้ให้ใช้งานกับระบบ Compo Bus/D และการขยายแร็คเพิ่มเติมจากซีพียูแร็ค
Special I/O Area	IR 100 – IR 199	จองไว้เพื่อให้หน่วยอินพุท/เอาต์พุทแบบพิเศษใช้งาน
Work Area	IR 232 – IR 235	ใช้เป็น Work Bit ในโปรแกรม

- Special Relay (SR) ใช้เก็บสถานะของแฟลคต่างๆ ของบิตควบคุม ที่ใช้แสดงในรูปแบบสถานะการทำงานของ PLC สัญญาณ (Clock Pulse) และค่าความผิดพลาดต่างๆ

- Temporary Relay (TR) Area ใช้ร่วมกับการเขียนโปรแกรม ในกรณีที่มีการแยกสาขา โดยพื้นที่นี้ใช้คำสั่ง LD และ OUT เท่านั้น

- Holding Relay (HR) Area เป็นพื้นที่หน่วยความจำที่ใช้เก็บและถ่ายข้อมูลต่างๆ สามารถเรียกใช้งานได้หลายครั้งในโปรแกรม สามารถใช้เก็บข้อมูลไว้ได้ในขณะที่ไฟที่จ่ายให้กับ PLC เกิดดับหรือหยุดทำงานกะทันหัน เนื่องจากพื้นที่หน่วยความจำในส่วนของ HR จะใช้แบตเตอรี่ในการเก็บสำรองข้อมูล

- Auxiliary Relay (AR) Area สำหรับพื้นที่หน่วยความจำในส่วนนี้ใช้เก็บค่าเวลา แฟลคสถานะต่างๆ และบิตควบคุมการทำงานของ PLC บางรุ่นใช้พื้นที่หน่วยความจำส่วนนี้ในการเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เก็บค่า Real Time Clock ซึ่งเป็นพื้นที่ปฏิทินนาฬิกาใช้เก็บค่าเวลา/วัน/เดือน/ปีและวันที่ เป็นค่าเวลาปัจจุบันที่สามารถนำไปใช้แทนฟังก์ชัน Time/Counter ได้ เช่น ต้องการควบคุมการเปิดปิดปั๊มน้ำในช่วงเวลาที่กำหนด แทนที่จะใช้คำสั่ง Time/Counter หลายๆตัวในการตั้งเวลาสามารถนำค่าเวลา Real Time Clock ใช้แทนได้

ตารางที่ 2.3 แสดงค่า Real Time Clock ของ PLC

Words	Bit	Function
AR 17	00 to 07	Minutes : 00 to 59
	08 to 15	Hours : 00 to 23 (24-hour system)
AR 18	00 to 07	Seconds : 00 to 59
	08 to 15	Minutes : 00 to 59
AR 19	00 to 07	Hours : 00 to 23 (24-hour system)
	08 to 15	Day of Month : 01 to 31
AR 20	00 to 07	Month : 1 to 12
	08 to 15	Year : 00 to 99
AR 21	00 to 07	Day of week : 00 to 06
	08 to 12	Not used
	13	30 Second Compensation Bit
	14	Clock Stop
	15	Clock Set Bit

- Link Relay (LR) Area ใช้ในการแลกเปลี่ยนข้อมูลระหว่าง PLC กับ PLC แบบเครื่องต่อเครื่อง คือ PLC เครื่องหนึ่งจะทำหน้าที่ Master ส่วนอีกเครื่องหนึ่งเป็น Slave พื้นที่ LR ของ PLC ทั้งสองจะใช้ร่วมกันและเมื่อเขียนข้อมูลลงใน LR ของ PLC ที่เป็น Master จะทำให้ LR หมายเลขเดียวกันของ PLC เครื่องที่เป็น Slave ถูกเขียนด้วยข้อมูลเดียวกันโดยอัตโนมัติ เหมาะสำหรับระบบที่ต้องการส่งข้อมูลเพื่อทำงานร่วมกัน

ตารางที่ 2.4 พื้นที่หน่วยความจำในพื้นที่หน่วยความจำ LR

DM 6645 setting	LR00 to LR63	LR00 to LR31	LR 00 to LR15
Master words	LR 00 to LR 31	LR 00 to LR 15	LR 00 to LR 07
Slave words	LR 00 to LR 63	LR 16 to LR 31	LR 08 to LR 15

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาค้นคว้า ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- Time/Counter (TC) Area ใช้ตั้งค่าเวลา และค่าการนับ ให้กับ PLC สำหรับ PLC บางรุ่นจะใช้พื้นที่หน่วยความจำในส่วนเดียวกัน เช่น ถ้าใช้ไทเมอร์หมายเลข 0 แล้ว จะไม่สามารถใช้เคาน์เตอร์หมายเลข 0 ได้อีก แต่ PLC บางรุ่นมีการแยกพื้นที่หน่วยความจำในส่วน ของไทเมอร์และเคาน์เตอร์ออกจากกันดังนั้น จึงสามารถใช้ไทเมอร์และเคาน์เตอร์หมายเลข เดียวกันได้

- Data Memory (DM) Area เป็นหน่วยความจำที่ใช้ในการเก็บข้อมูล แบ่งเป็น สองส่วน คือ ส่วนที่สามารถอ่าน (Read) และเขียน (Write) ข้อมูลลงไปได้สามารถเปลี่ยนแปลง ค่า เพื่อใช้งาน PLC ได้ พื้นที่หน่วยความจำส่วนนี้ยังแบ่งออกเป็นส่วนย่อยอีกหลายส่วน ยกตัวอย่าง เช่น ส่วนของ Special I/O Unit Area จะใช้งานในกรณีที่มีการใช้หน่วยอินพุท/ เอาต์พุทแบบพิเศษ โดยการกำหนดหมายเลขของหน่วยความจำที่ใช้งานโดยปรับตั้งค่าทางด้าน หน้าของหน่วยอินพุท/เอาต์พุทแบบพิเศษนั้น เช่น

Unit Number หมายเลข 0 จะใช้งานพื้นที่ DM 1000 ถึง DM 1099

Unit Number หมายเลข 1 จะใช้งานพื้นที่ DM 1100 ถึง DM 1199

พื้นที่อีกส่วนหนึ่งคือส่วนที่อ่านค่าได้อย่างเดียว (Read Only) พื้นที่ส่วนนี้ใช้ในการเก็บค่าใช้งานต่างๆ ของ PLC และแบ่งออกเป็นพื้นที่ส่วนต่างๆ ได้แก่ System Setting, PC Setup และ Expansion DM เป็นต้น

2.2.3 ความสามารถของ PLC

PLC สามารถควบคุมงานได้ 3 ลักษณะ คือ

1. งานที่ทำตามลำดับก่อนหลัง (Sequence Control) ตัวอย่างเช่น
 - การทำงานของระบบรีเลย์
 - การทำงานของไทเมอร์ เคาน์เตอร์
 - การทำงานของ P.C.B.C
 - การทำงานในระบบกึ่งอัตโนมัติ ระบบอัตโนมัติ หรืองานที่เป็นกระบวนการทำงานของเครื่องกลต่างๆ

2. งานควบคุมสมัยใหม่ (Sophisticated Control) ตัวอย่างเช่น

- การทำงานทางคณิตศาสตร์ เช่น บวก ลบ คูณ หาร
- การควบคุมแบบอนาล็อก (Analog Control) เช่นการควบคุมอุณหภูมิ (Temperature)
- การควบคุม P.I.D. (Proportional Integral Derivative)
- การควบคุมเซอร์โวมอเตอร์ (Servo Motor Control)
- การควบคุม Stepper Motor

- Information Handling

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์และเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. งานควบคุมเกี่ยวกับงานอำนวยการ (Supervisory Control) ตัวอย่างเช่น
 - งานสัญญาณเตือน (Alarm) และ Process Monitoring
 - Fault Diagnostic and Monitoring
 - งานต่อร่วมกับคอมพิวเตอร์ (RS-232/RS-485)
 - Printer / ASCII Interfacing
 - งานควบคุมอัตโนมัติในโรงงานอุตสาหกรรม(Factory Automation Networking)
 - LAN (Local Area Network)
 - WAN (Wide Area Network)
 - FS , FMS และ CIM เป็นต้น

2.3 สื่อสารข้อมูลระหว่างโปรแกรม Visual Basic กับเครื่องควบคุม PLC

2.3.1 โพรโทคอล (Protocols)

2.3.1.1 ข้อตกลงในการติดต่อสื่อสาร (โพรโทคอล:Protocols)

ในการสื่อสารข้อมูลจะต้องมีกฎหรือข้อกำหนดในการสื่อสารข้อมูล หรือที่นิยมเรียกว่า โพรโทคอล (Protocols) ซึ่งเป็นส่วนที่จะกำหนดมาตรฐานในการควบคุมและจัดการระบบการสื่อสารข้อมูล

สำหรับรายละเอียดที่กล่าวในหัวข้อนี้จะเกี่ยวข้องเฉพาะในส่วนของโปรโตคอลการควบคุมการเชื่อมโยง (Data Link Control Protocols หรือ DLCP) ซึ่งจัดการในส่วนของขั้นตอนและหลักการต่างๆ คือ โครงสร้างและรายละเอียดของข้อมูล วิธีการสื่อสารข้อมูลการตรวจสอบแก้ไขความผิดพลาดของข้อมูลและขบวนการในการควบคุมการติดต่อสื่อสาร โดย DLCP แบ่งได้ตามโครงสร้างของข้อมูล 2 แบบคือ Byte-Oriented Protocols และ Bit-Oriented Protocols

1. ไบท์โอเรียนโปรโตคอล (Byte-oriented Protocols)

โพรโทคอลแบบนี้เป็นโพรโทคอลที่การสื่อสารข้อมูล และการควบคุมการทำงานจะใช้ลักษณะข้อมูลที่เป็นตัวอักษร (character) หรือไบท์ (Byte) หรืออาจเรียกว่า Character Oriented Protocol ซึ่งแบ่งออกเป็น

1.1 อะซิงโครนัสโปรโตคอล (Asynchronous protocols)

โพรโทคอลในการสื่อสารข้อมูลนี้จะใช้การสื่อสารข้อมูลแบบ Half-Duplex ที่มีลักษณะการสื่อสารข้อมูลแบบอะซิงโครนัสซึ่งเป็นการสื่อสารข้อมูลแบบพื้นฐานที่ใช้งานในเวลานานแล้ว จึงมี

เอกสารรายละเอียดและขั้นตอนในการสื่อสารข้อมูลที่ทำให้มีโอกาสดีผลผลิตได้น้อยและยังมีข้อดีที่การไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สื่อสารข้อมูลแบบนี้มีโครงสร้างการทำงานที่ง่าย อุปกรณ์ที่ใช้ในการสื่อสารข้อมูลก็ไม่ สลับซับซ้อน และมีราคาถูกโปรโตคอลแบบนี้จึงเหมาะสำหรับใช้ในระบบขนาดเล็ก

1.2 ไบนารีซิงโครนัสโปรโตคอล (Binary synchronous protocols)

โปรโตคอลแบบนี้จะมีลักษณะการทำงานที่ใช้งานข้อมูลเป็นลักษณะไบนารี และยังคงใช้การสื่อสาร ข้อมูลแบบซิงโครนัสมีรายละเอียดและขั้นตอนในการสื่อสารข้อมูลที่ทำให้ความน่าเชื่อถือมากกว่า อีกทั้งยังสามารถใช้อัตราเร็วในการสื่อสารข้อมูลที่สูงกว่า โดยตัวอย่างของการสื่อสารข้อมูลแบบนี้ ที่ได้กำหนดเป็นมาตรฐานแล้วคือ การสื่อสารข้อมูลแบบนี้ที่ได้กำหนดเป็นมาตรฐานแล้วคือ การ สื่อสารข้อมูลตามมาตรฐาน BSC (Binary Synchronous Communications) ซึ่งเป็นโปรโตคอลที่มี ลักษณะของข้อมูลแบบไบนารีที่ได้รับความนิยมนำไปใช้งาน

2. บิทโอเรียนท์โปรโตคอล (Bit-oriented protocols)

โปรโตคอลแบบนี้เป็นโปรโตคอลที่การสื่อสารข้อมูล และการควบคุม การทำงานจะทำโดยใช้ลักษณะข้อมูลที่เป็นบิทโดยมีตัวอย่างของการสื่อสารข้อมูลในลักษณะนี้ที่มี การกำหนดขึ้นเป็นมาตรฐานแล้ว คือ HDLC (High-Level Data Link Control) ยังมีโครงสร้าง ของข้อมูลแบบซิงโครนัส เช่นเดียวกับ BSC แต่ต่างกันที่มีลักษณะของข้อมูลแบบบิท ซึ่ง โปรโตคอลแบบนี้มีข้อดีที่สามารถสื่อสารข้อมูลได้รวดเร็วกว่าแต่โปรโตคอลแบบนี้ก็มีรายละเอียด และโครงสร้างในการสื่อสารข้อมูลที่สลับซับซ้อนมากทำให้การควบคุมการทำงานทำได้ยากและต้อง ใช้อุปกรณ์ที่มีราคาสูง จึงไม่เหมาะที่จะนำไปใช้งานกับระบบขนาดเล็ก

3. แพ็กเก็ต (Packet of Information)

รูปแบบของแพ็กเก็ตในระบบโครงข่ายจะประกอบด้วยส่วนต่างๆ ดังนี้

3.1 HEADER จะประกอบด้วย

- Preamble or start of packet indicator เป็นส่วนแรกเริ่มของ แพ็กเก็ตและในบางระบบอาจใช้ในการซิงค์กับสัญญาณพิกษาของตัวส่งและตัวรับ

- Control information ส่วนนี้เป็นข้อมูลที่บอกถึงวัตถุประสงค์ ของแพ็กเก็ตนั้นว่าใช้ทำอะไรเช่นเพื่อการจัดการระบบเพื่อดู Status ของ mode หรืออื่นๆ

นอกจากส่วนต่างๆ เหล่านี้แล้ว ในส่วน Header อาจจะมีส่วนที่ เป็น Sequential Number เป็นส่วนที่บอกให้ทราบถึงลำดับของแพ็กเก็ตในกรณีที่ข้อมูลมีความยาว หลายแพ็กเก็ต

3.2 INFORMATION

- Data field เป็นส่วนของข้อมูลจริงที่ต้องการจะส่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3 TAILER

- Frame Check Sequence (FCS) เป็นส่วนที่ใช้ตรวจสอบความถูกต้องของข้อมูลซึ่งอาจเป็น parity bit , Check sum หรือ CRC เป็นต้น
- End of Packet Indicator เป็นส่วนที่บอกให้ทราบว่าสิ้นสุดของข้อมูลแล้ว

2.3.1.2 การควบคุมความผิดพลาดในการส่งข้อมูล

การที่วงจรส่งข้อมูลขาดช่วงขณะ และผลกระทบต่อสัญญาณรบกวน (Noise) ทำให้ระดับแรงดันไฟฟ้าลดลงเป็นผลให้เกิดความผิดพลาดในการส่ง ดังนั้นต้องทำการค้นหาความผิดพลาดที่เกิดขึ้นระหว่างการส่งข้อมูลและแก้ไขความผิดพลาดให้ถูกต้อง

1. วิธีการเพิ่มบิตเข้าไปที่ข้อมูล (Parity Bit) เพื่อตรวจสอบความผิดพลาดมี 2 วิธี

- วิธี Parity แนวนอน

จะเพิ่ม 1 บิต เข้าไปที่แต่ละตัวอักษรที่จะส่ง โดยจะกำหนดว่าเป็นการตรวจสอบแบบ Odd Number หรือ Even Number แล้วผลรวมของข้อมูลที่ด้านรับจะทำการตรวจสอบว่าผลรวมของข้อมูลจะเป็น Odd Number หรือ Even Number

- วิธี Parity แนวตั้ง

จะตรวจสอบความผิดพลาดโดยการเพิ่ม 1 บิตเข้าไปที่ในบล็อก

01101101	1	
01100110	0	
10111010	1	
10010111	1	
00101001	1	Parity แนวนอน
00010110	1	
10101110	1	
11110010	1	
01000101		
Parity แนวตั้ง		

รูปที่ 2.5 แสดงรูปแบบการตรวจสอบด้วย Parity Bit

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2 วิธี Patrol Diffuse Inspection

วิธี Patrol Diffuse Inspection หรือการใช้ FCS : Frame Check Sequence สำหรับข้อมูลอันหนึ่งจะสร้าง Error Inspection Sign (CRC Sign) ขนาด 2 Byte ที่จะคำนวณได้บนพื้นฐานของกฎที่กำหนดไว้ แล้วเพิ่มเครื่องหมายไปที่ข้อมูลอันนั้นที่ทางด้านรับจะตรวจสอบข้อมูลโดย Inspection Sign ได้กำหนดไว้โดยสูตรการคำนวณเหมือนกัน (Blast Error ความผิดพลาดของข้อมูลที่ต่อเนื่อง) ก็สามารถตรวจสอบได้ ทำให้มีความน่าเชื่อถือสูง

2.3.1.3 อัตราการส่งข้อมูล (Data Transmission Rate)

Transmission Rate นั้นแยกออกได้เป็นส่วนของ “Data Transmission Speed” และส่วนของ “Modulation Speed”

1. Data Transmission Speed

Data Transmission Speed หรือ Bite rate นั้น จะแสดงในรูปของจำนวน Bite ที่สามารถส่งได้ภายใน 1 วินาที มีหน่วยเป็นบิตหรือวินาที โดยเขียนว่า Bit/sec หรือ Bps จะแสดง Definition ของ Data Transmission Speed จะแสดงด้วยสมการข้างล่างนี้

$$S = (1/T) \log_2 n$$

S : Data Transmission Speed

T : Continue Time ของ 1 Pulse ของ Transmission Pulse

n : จำนวนของค่าสภาวะใน 1 Pulse

ในกรณีที่ Pulse 1 ถูกมีค่า 2 ค่าคือ 0 และ 1 เป็นสมการจะเป็น $S = 1/T$ โดยที่ T เป็นเวลาต่อเนื่อง (Continue Time) ของ Pulse 1 $1/T$ จึงแสดงถึงจำนวนของ Pulse ใน 1 วินาที ซึ่งก็คือจำนวนบิตสามารถส่งไปได้ใน 1 วินาทีนั่นเอง

2. Modulation Speed

Modulation speed หรือ Baud rate ก็คือจำนวนครั้งที่สามารถ Modulate ได้ใน 1 วินาที การ Modulate นั้นมี Amplitude Mod, Frequency Mod, Phase Mod และอื่นๆ ในกรณีที่ T แสดงถึงช่วงเวลาในการเปลี่ยนแปลงสภาวะนั้น จะสามารถกำหนดได้ดังสมการข้างล่างนี้โดยหน่วยจะเป็น Baud

$$B = 1/T$$

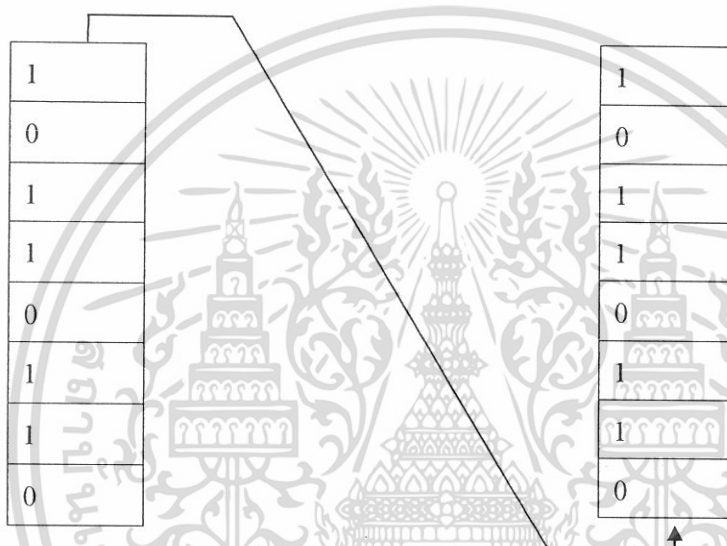
B : Modulation Speed (Baud)

T : เวลาในการเปลี่ยนสภาวะหนึ่งเป็นอีกสภาวะหนึ่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3.2 การสื่อสารข้อมูลแบบอนุกรม (Serial)

ในการติดต่อแบบนี้ ข้อมูลจะถูกส่งออกมาทีละบิต ระหว่างจุดส่งและจุดรับ ซึ่งการส่งข้อมูลแบบนี้จะช้ากว่าการส่งแบบขนาน แต่ใช้ตัวกลางในการสื่อสารเพียงช่องเดียวหรือสายเพียงคู่เดียวทำให้ค่าใช้จ่ายสำหรับสื่อสารกลางถูกกว่าแบบขนาน การส่งแบบนี้ใช้สำหรับส่งในระยะทางไกลมากกว่า 100 ฟุต ข้อมูลจากจุดส่งจะต้องถูกเปลี่ยนให้เป็นอนุกรมก่อนแล้วค่อยทยอยส่งออกไปทีละบิต ไปยังจุดรับ ที่จุดรับจะต้องมีรูปแบบในการเปลี่ยนข้อมูลที่ถูกส่งมาให้มาเป็นแบบขนานในการแปลงต้องมีรูปแบบที่เหมาะสมเพื่อป้องกันการผิดพลาดของข้อมูล

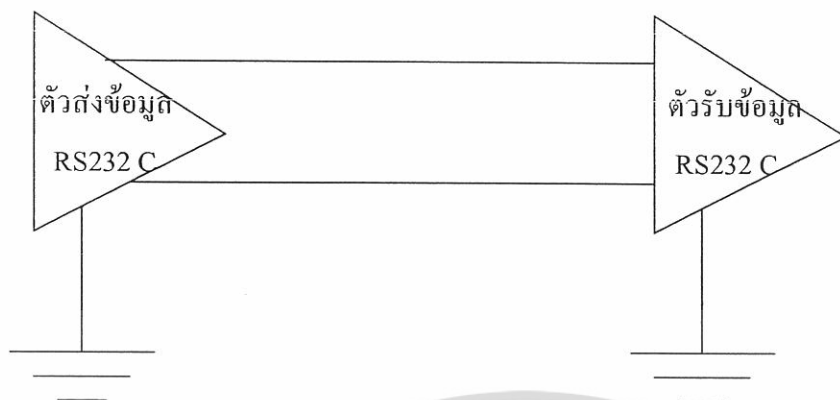


รูปที่ 2.6 การสื่อสารข้อมูลแบบอนุกรม

2.3.2.1 มาตรฐานในการสื่อสารข้อมูลแบบอนุกรม

การสื่อสารข้อมูลแบบอนุกรมตามมาตรฐาน RS-232C

การสื่อสารข้อมูลแบบอนุกรมที่ใช้งานอยู่ในปัจจุบันนั้น ได้มีการกำหนดมาตรฐานการรับส่งข้อมูลไว้หลายแบบด้วยกัน แต่ที่ได้รับความนิยมนำมาใช้งานอย่างมาก คือ การสื่อสารข้อมูลแบบอนุกรมตามมาตรฐาน RS-232C และที่มาตรฐานนี้เป็นที่นิยมเนื่องจากเป็นระบบการสื่อสารข้อมูลที่ใช้ในการเชื่อมโยงไมโครคอมพิวเตอร์ IBM PC ซึ่งเป็นคอมพิวเตอร์ที่มีใช้อย่างแพร่หลายมากจากอดีตจนถึงปัจจุบันมาตรฐานการสื่อสารนี้ในการออกแบบเบื้องต้นได้ออกแบบ สำหรับการเชื่อมต่อกับเครื่องโมเด็ม (MODEM) ซึ่งเป็นอุปกรณ์ที่ใช้การสื่อสารข้อมูลระหว่างคอมพิวเตอร์ผ่านทางสายโทรศัพท์ ซึ่งทำให้อัตราการรับส่งข้อมูลถูกจำกัดให้มีค่าที่ค่อนข้างต่ำ มาตรฐาน RS-232C นี้ได้ออกแบบให้มีโครงสร้างการสื่อสารเป็นแบบจุดต่อจุด เอกสารเท่านั้น โดยมีลักษณะสมบัติทางไฟฟ้าและทางกายภาพดังแสดงรูป อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.7 แสดงโครงสร้างของการสื่อสารข้อมูลแบบอนุกรมตามมาตรฐาน RS -232C

2.3.2.2 ข้อตกลงในการสื่อสารของเครื่องควบคุม PLC

รูปแบบของข้อตกลงในการสื่อสารของเครื่องควบคุม PLC โดยทั่วไปจะเป็นลักษณะการตอบกันระหว่างเครื่องควบคุมกับอุปกรณ์ภายนอก ซึ่งอุปกรณ์ภายนอกมักจะเป็นฝ่ายถามก่อน โดยการส่งบล็อกรหัสคำสั่ง (Command Block) ออกไปจากนั้นเครื่องควบคุมจะทำการตรวจสอบแล้วส่งบล็อกตอบกลับมา (Response Block)

1. รูปแบบของบล็อก (Block Format)

ลักษณะของบล็อกของเครื่องควบคุม PLC แต่ละผู้ผลิตจะแตกต่างกันไป แต่จะมีพื้นฐานเดียวกันดังตัวอย่างบล็อกคำสั่งต่อไปนี้

@	Unit Number	HEADER	Text	FCS	*	CR
---	----------------	--------	------	-----	---	----

รูปที่ 2.8 แสดงรูปแบบของบล็อก (Block Format)

1.1 หมายเลขเครื่องในการเชื่อมต่อที่เป็นโครงข่ายแบบหลายจุด ซึ่งเครื่องควบคุมที่เชื่อมต่ออยู่ในระบบจะมีมากกว่า 1 เครื่อง การกำหนดว่าต้องการส่งข้อมูลให้กับเครื่องควบคุมตัวใด เช่น @05 คือหมายเลขเครื่องที่ 05

1.2 ส่วน HEADER เป็นส่วนของคำสั่งหลักที่จะกำหนดว่าต้องการทำกับข้อมูลส่วนใด เช่น ต้องการอ่านข้อมูลของอินพุตต้องการเขียนข้อมูลให้กับตัวนับ เป็นต้น เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.3 ส่วน TEXT เป็นส่วนของข้อมูล เช่น คำที่อ่านได้จากอินพุท หรือ คำที่จะต้องเขียนลงในพื้นที่ต่างๆ

1.4 ส่วนของ FCS เป็นส่วนของการควบคุมความผิดพลาดของข้อมูลซึ่งได้จากการคำนวณ

1.5 ส่วนของ TERMINAL เป็นส่วนที่ปิดท้ายบอกให้ทราบว่าจบบล็อก และมักจะติดตามด้วยรหัสของ Carrier Return (CR)

2. การคำนวณ FCS

FCS หรือ Frame Check Sequence เป็นสิ่งที่ใช้ในการตรวจสอบความผิดพลาดของการสื่อสาร เป็นการเปลี่ยนข้อมูล 8 บิต เป็น 2 ตัวอักษรข้อมูล ASCII แล้วนำมาทำการ Exclusive or (XOR) โดยเริ่มจาก @ จนถึงตัวอักษรตัวสุดท้ายของ TEXT

2.3.2.3 คำสั่งและคำตอบสนอง

ในขณะที่ทำการเชื่อมต่อเพื่อสื่อสารข้อมูล Host สามารถที่จะทำการเฝ้ามองการดำเนินการ และสามารถที่จะทำการควบคุมการทำงานของเครื่องควบคุม ถ้าอยู่ในสถานะการเฝ้ามอง โฮสคอมพิวเตอร์จะต้องส่งคำสั่งไปถามตามชนิดของข้อมูลที่ต้องการ แต่ละเครื่องควบคุม หรือ ถ้าในสถานะการควบคุมก็สามารถส่งคำสั่งไปทำการเปลี่ยนแปลงค่าของข้อมูลที่มีอยู่ในหน่วยความจำโดยตรง เช่น ข้อมูลของอินพุท/เอาต์พุท เป็นต้น เวลาของการตอบสนองจะแปรไปขึ้นอยู่กับความเร็วในการส่งผ่านข้อมูล จำนวนของข้อมูล และเวลาในการสแกนของเครื่องควบคุม และถ้าเวลาการสื่อสารมากขึ้นก็เป็นผลให้เวลาในการสแกนมากขึ้นตามไปด้วย ต่อไปจะเป็นคำสั่งต่างๆ ที่เป็นข้อกำหนด (โปรโตคอล) ที่ใช้ในการสื่อสาร

1. บล็อกคำสั่งประกาศ

บล็อกคำสั่งประกาศถูกส่งโดยโฮสไปเพื่อตรวจสอบว่ามีเครื่องควบคุมอยู่ในการเชื่อมต่อหรือเครือข่ายหรือไม่

รูปแบบคำสั่ง

@	[Unit Number]	At	*	[FCS]	[CR]
---	------------------	----	---	-------	------

รูปแบบคำตอบสนอง

@	[Unit Number]	PC84	*	[FCS]	[CR]
---	------------------	------	---	-------	------

2. บล็อกคำสั่งอ่านพื้นที่ Input/Output/Internal Relay

รูปแบบคำสั่ง

@	[Unit Number]	RI	[ตำแหน่งเริ่มต้น]	[จำนวนข้อมูล]	*	[FCS]	[CR]
---	------------------	----	-------------------	---------------	---	-------	------

รูปแบบคำตอบสนอง

@	[Unit Number]	RI	[ตำแหน่ง เริ่มต้น]	[ข้อมูล XX(1)]	[ข้อมูล XX(..)]	[ข้อมูล XX(n)]	*	[FCS]	[CR]
---	-------------------	----	-----------------------	---------------------	----------------------	---------------------	---	-------	------

3. บล็อกคำสั่งเขียนพื้นที่ Input/Output/Internal Relay

รูปแบบคำสั่ง

@	[Unit Number]	WI	[ตำแหน่ง เริ่มต้น]	[ข้อมูล XX(1)]	[ข้อมูล XX(..)]	[ข้อมูล XX(n)]	*	[FCS]	[CR]
---	-------------------	----	-----------------------	---------------------	----------------------	---------------------	---	-------	------

รูปแบบคำตอบสนอง

@	[Unit Number]	WI	[รหัสตอบสนอง]	*	[FCS]	[CR]
---	---------------	----	---------------	---	-------	------

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3

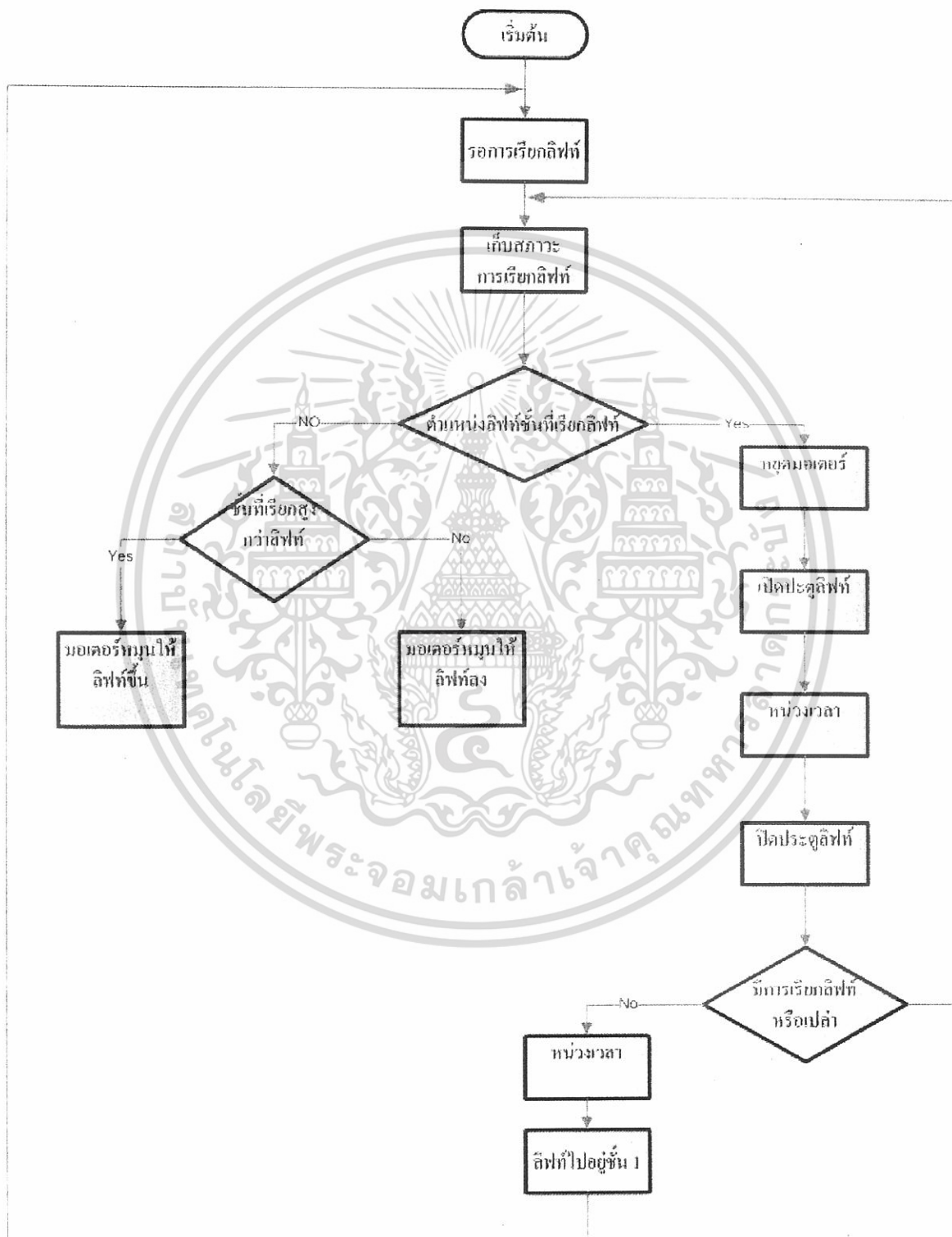
การออกแบบและการสร้าง

3.1 ข้อกำหนดการทำงานของลิฟต์

- ลิฟต์ขึ้นหรือลงได้นั้น ประตูลิฟต์จะต้องปิดสนิทเสียก่อน
- การทำงานจะขึ้นหรือลงนั้น จะขึ้นอยู่กับว่ามีการกดให้ขึ้นหรือลงก่อน ถ้าลิฟต์ทำงานขึ้นมันต้องทำงานในกรณีการขึ้นให้หมดการทำงานก่อน แล้วจึงจะลงได้และในกรณีตอนลงก็เช่นกัน ก็จะต้องลงจนหมดทำงานก่อนจึงจะขึ้นได้
- ในกรณีที่ลิฟต์กำลังวิ่งขึ้น โดยตัวลิฟต์อยู่ต่ำกว่าชั้นที่เราอยู่ แล้วเราเรียกลิฟต์ต้องการที่จะขึ้นไปในชั้นที่สูงกว่าตัวลิฟต์ก็จะหยุดรับ แต่กรณีที่ตัวลิฟต์กำลังขึ้นและตัวลิฟต์อยู่สูงกว่าชั้นที่เรากำลังอยู่นั้น ลิฟต์จะยังไม่มารับเรา แต่จะกลับมารับเราเมื่อเสร็จสิ้นภาระกิจในการวิ่งขึ้นเรียบร้อยแล้ว
- ในกรณีที่ลิฟต์กำลังวิ่งลง โดยตัวลิฟต์อยู่สูงกว่าชั้นที่เราอยู่ แล้วเราเรียกลิฟต์ต้องการที่จะลงชั้นล่าง ตัวลิฟต์ก็จะหยุดรับ แต่กรณีที่ตัวลิฟต์กำลังวิ่งลง ตัวลิฟต์อยู่ในชั้นที่ต่ำกว่า ลิฟต์จะยังไม่มารับเรา แต่จะมารับเราเมื่อเสร็จสิ้นภาระกิจในการวิ่งลงเรียบร้อยแล้ว
- ในกรณีที่ลิฟต์กำลังวิ่งขึ้นและอยู่ในชั้นที่ต่ำกว่านั้น แล้วเราเรียกลิฟต์ต้องการจะลงนั้น ลิฟต์จะไม่หยุดรับเรา แต่จะเก็บสถานะไว้ จนเมื่อตัวลิฟต์วิ่งลงมานั้นจึงจะมาจอด
- ในกรณีที่ลิฟต์กำลังวิ่งลงนั้นและอยู่ในชั้นที่สูงกว่า แล้วเราเรียกลิฟต์ต้องการขึ้น ลิฟต์จะไม่หยุดรับเรา แต่จะเก็บสถานะไว้จนเมื่อตัวลิฟต์วิ่งขึ้น จึงจะหยุดรับ
- ไม่ว่าตัวลิฟต์จะขึ้นหรือลง ถ้ามีการกดปุ่มเรียกลิฟต์แล้วนั้นสถานะต่าง ๆ นั้นจะถูกเก็บค่าไว้หมด
- ในกรณีที่ลิฟต์กำลังลงมานั้น คนภายในลิฟต์ต้องการที่จะลงในชั้นที่เราอยู่ แล้วเราก็จะเรียกลิฟต์ต้องการที่จะขึ้น เมื่อลิฟต์มาหยุดที่ชั้นที่เราอยู่แล้วนั้น เพื่อให้คนภายในลิฟต์ลง แต่ถ้าตัวลิฟต์จะต้องลงไปยังชั้นล่างอีก ในกรณีมีคนชั้นล่างเรียกลิฟต์ หรือมีคนภายในลิฟต์ต้องการที่จะลงไปชั้นล่างอีก สถานะการเรียกชั้นที่เราเรียกซึ่งต้องการที่จะขึ้นนั้นจะต้องถูกเก็บไว้ และเมื่อลิฟต์ขึ้นมากก็ต้องหยุดรับให้ขึ้นด้วย
- ในกรณีที่ลิฟต์กำลังขึ้นมานั้น คนภายในลิฟต์ต้องการที่จะมาในชั้นที่เราอยู่นั้น และเราเรียกลิฟต์ต้องการที่จะลง เมื่อลิฟต์มาหยุดที่ชั้นที่เราอยู่แล้วนั้น เพื่อให้คนภายในลิฟต์ออกมาจากลิฟต์และเมื่อลิฟต์ต้องการขึ้นต่อไป การเรียกลงของเรานั้น จะต้องถูกเก็บสถานะไว้ และเมื่อลิฟต์ลงมากก็จะหยุดรับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ในระหว่างที่ลิฟท์กำลังเคลื่อนที่ไม่สามารถเปิดประตูลิฟท์ได้
- ในกรณีที่ไม่มีใครอยู่ในลิฟท์ภายใน 15 วินาที ลิฟท์จะวิ่งลงมารออยู่ที่ชั้น 1 สำหรับรอการใช้ครั้งต่อไป



รูปที่ 3.1 ข้อกำหนดการทำงานของลิฟท์ แสดงโดย Flowchart

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2 วางแผนการปฏิบัติงานในการสร้างโครงการ ดังนี้

ทำการวางแผนการทำงาน โดยกำหนดเป็นตารางวางแผนการทำงานและปฏิบัติงาน

- ทำการศึกษาวิธีใช้โปรแกรม Visual Basic และกราฟฟิกในการแสดงลักษณะการทำงานของ Hardware (ลิฟท์จำลอง)
- ทำการศึกษาวิธีการใช้งาน Hardware (ลิฟท์จำลอง)
- ศึกษาการเงื่อนไขการสื่อสารข้อมูลระหว่างโปรแกรม Visual Basic กับเครื่องควบคุม PLC คือ การเขียนโปรโตคอล
- โปรแกรม Visual Basic แสดงการทำงานของลิฟท์ สัมพันธ์กับ Hardware (ลิฟท์จำลอง)

3.3 การออกแบบทางด้านซอฟต์แวร์

3.3.1 ลักษณะการทำงานของโปรแกรม Visual Basic ในการสื่อสารข้อมูลกับ PLC เพื่อแสดงผลในส่วนของกราฟฟิก

3.3.1.1 การส่งคำสั่งสอบถามสถานะ Bit ของ PLC

เราใช้ คำสั่ง Timer ในส่วน Control Box ของ Visual Basic ในการส่งคำสั่ง

`MSComm1.Output = @00RR0000001144*`

เพื่อสอบถามข้อมูลไปยัง PLC โดยเราสามารถกำหนด

ความถี่ในการส่ง ได้จากการกำหนดค่า Properties ของ Timer.inteval = value โดย value จะมีค่าเท่ากับ 1/1000 วินาที

3.3.1.2 การรับค่าสถานะ Bit จาก PLC

ค่าที่รับได้จะอยู่ในรูป Response ที่ตอบกลับมาจาก PLC โดยค่าที่ได้จะอยู่ในรูป ฐาน 16 เราจึงต้องนำค่าที่ได้รับมาแปลงให้อยู่ ในรูปเลขฐาน 2 แล้วจัดเก็บให้อยู่ภายในแต่ละ Chanel ของ PLC เพื่อที่เราสามารถกำหนดกรณี ต่างๆ ให้กราฟฟิกทำงานสอดคล้องกับ Bit ที่เราสนใจ

3.3.1.3 การกำหนดกรณีต่างๆของ Bit ให้กับกราฟฟิกของโปรแกรม

กราฟฟิกของลิฟท์จะทำงานสัมพันธ์ กับบิตของ PLC ที่เราสนใจ

- กราฟฟิกลิฟท์เคลื่อนที่ขึ้น อ้างอิงจาก Bit ที่ 000.00 ของ PLC
- กราฟฟิกลิฟท์เคลื่อนที่ลง อ้างอิงจาก Bit ที่ 000.01 ของ PLC
- กราฟฟิกแสดงการทำงานของมอเตอร์ประตูเปิดอ้างอิงจาก Bit 000.08 ของ PLC

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- กราฟฟิกแสดงการทำงานของมอเตอร์ประตูเปิดอ้างอิงจาก Bit 000.09 ของ PLC
- กราฟฟิกเมื่อประตูลิฟท์เปิด อ้างอิงจาก Bit 000.02 ของ PLC
- กราฟฟิกเมื่อประตูลิฟท์เปิด อ้างอิงจาก Bit 000.03 ของ PLC
- กราฟฟิกแสดงตำแหน่งของลิฟท์ที่อยู่ชั้น 1 อ้างอิงจาก Bit 009.04 ของ PLC
- กราฟฟิกแสดงตำแหน่งของลิฟท์ที่อยู่ชั้น 2 อ้างอิงจาก Bit 009.05 ของ PLC
- กราฟฟิกแสดงตำแหน่งของลิฟท์ที่อยู่ชั้น 3 อ้างอิงจาก Bit 009.06 ของ PLC
- กราฟฟิกแสดงตำแหน่งของลิฟท์ที่อยู่ชั้น 4 อ้างอิงจาก Bit 009.07 ของ PLC
- แสดงตำแหน่งชั้น ในขณะที่ลิฟท์เคลื่อนที่อยู่ชั้นใด
- การทำงานของมอเตอร์เปิด - ปิดประตูที่สัมพันธ์กับ Hardware โดยแสดงทางสถานะของ Bit
- การทำงานของมอเตอร์ขึ้น - ลงที่สัมพันธ์กับ Hardware โดยแสดงทางสถานะของ Bit

3.3.2 ลักษณะการทำงานของโปรแกรม Visual Basic ในส่วนที่ใช้ควบคุม PLC

3.3.2.1 ลักษณะของการส่งข้อมูล

การส่งข้อมูลจะกระทำการส่งข้อมูลก็ต่อเมื่อมีการกดปุ่ม Command ต่างๆ ที่ทำการควบคุมลิฟท์ รูปแบบของข้อมูลที่ถูกส่งออกไป เช่น `MSComm1.Output = "@00KSCIO 0003013F*"` คำสั่งจะถูกส่งออกไปทาง Port อนุกรม RS232 เพื่อควบคุมการทำงานของ PLC

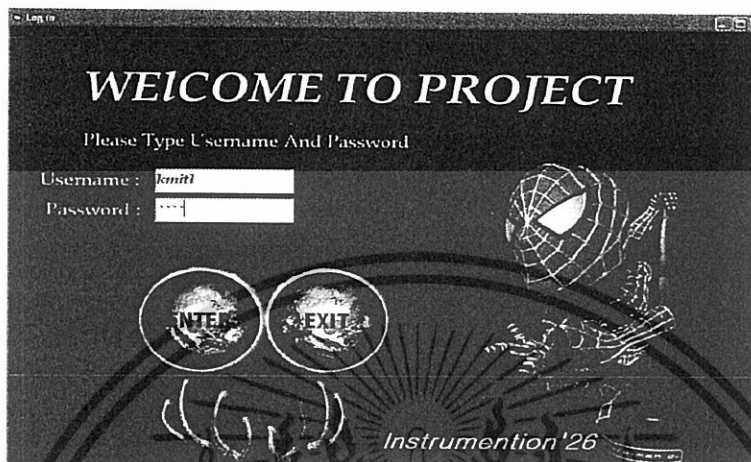
3.3.2.2 การกำหนด Output ของ Command ให้สัมพันธ์กับโปรแกรมของ PLC

- ปุ่ม Command ชั้น 1 เมื่อกดจะส่ง Output set ค่า Bit ที่ 003.01 ของ PLC
- ปุ่ม Command ชั้น 2 เมื่อกดจะส่ง Output set ค่า Bit ที่ 003.02 ของ PLC
- ปุ่ม Command ชั้น 3 เมื่อกดจะส่ง Output set ค่า Bit ที่ 003.03 ของ PLC
- ปุ่ม Command ชั้น 4 เมื่อกดจะส่ง Output set ค่า Bit ที่ 003.04 ของ PLC
- ปุ่ม Command เปิดประตู เมื่อกดจะส่ง Output set ค่า Bit ที่ 003.05 ของ PLC
- ปุ่ม Command ชั้น 1 เมื่อกดจะส่ง Output set ค่า Bit ที่ 003.06 ของ PLC

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

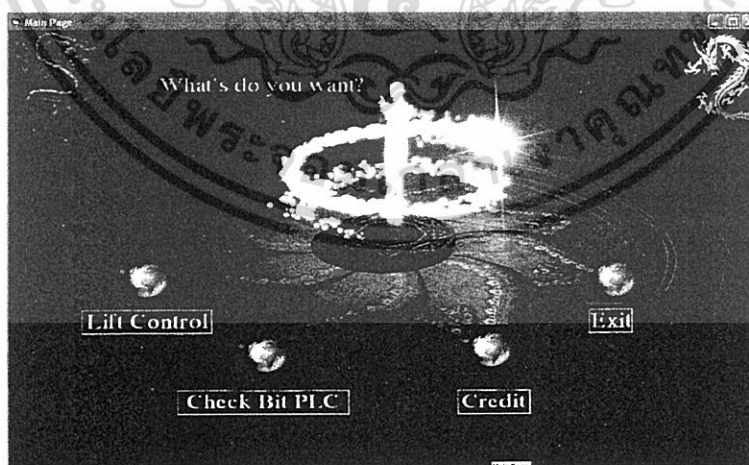
3.3.3 แอปพลิเคชันแสดงการทำงานของลิฟท์ ที่สร้างจากโปรแกรม

3.3.3.1 หน้า Log in แสดง Username และ Password เพื่อหน้าป้องกันบุคคลภายนอกมารบกวนเกี่ยวกับการตรวจสอบลักษณะการทำงานของลิฟท์



รูปที่ 3.2 หน้า Log in

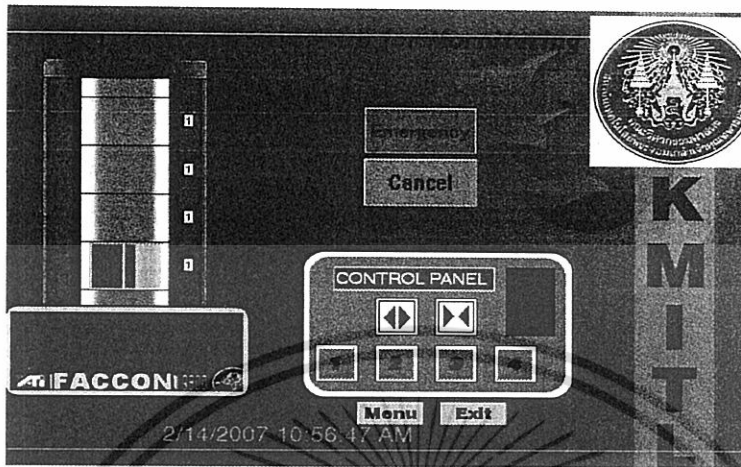
3.3.3.2 หน้า Menu มีหน้าที่แสดงการ Link ไปยังหน้าต่างๆ ทั้งในส่วน of หน้าแสดงลักษณะเคลื่อนที่ของลิฟท์ที่สร้างจากในโปรแกรม Visual Basic , ส่วนแสดงการทำงานของลิฟท์จากสถานะของ Bit และส่วนของผู้จัดทำ



รูป 3.3 หน้า Menu

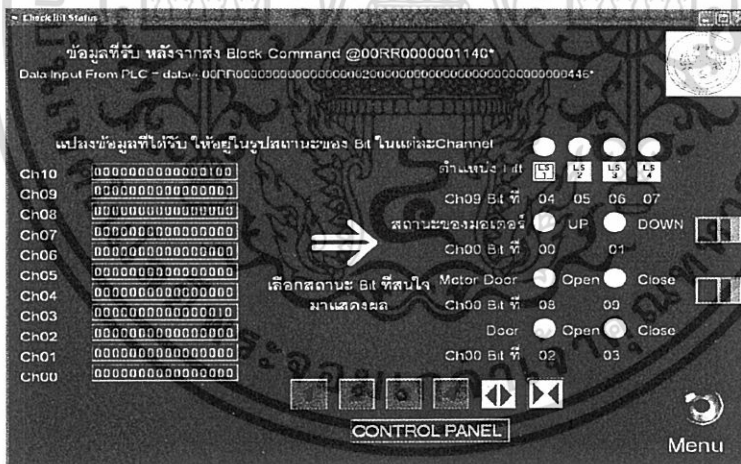
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3.3.3 หน้า Control Lift แสดงลักษณะการเคลื่อนที่ของลิฟท์ที่สร้างจากในโปรแกรม Visual Basic โดยจะสัมพันธ์กับ Hardware



รูปที่ 3.4 หน้า Control Lift

3.3.3.4. หน้า Check Bit Status แสดงการทำงานของลิฟท์ในรูปแบบของสถานะของ Bit



รูปที่ 3.5 หน้า Check Bit Status

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3.3.5 หน้า Credit แสดงรายชื่อของผู้จัดทำโครงการ และอาจารย์ที่ปรึกษา



รูปที่ 3.6 หน้า Credit

3.3.4 โปรแกรม Visual Basic และคำสั่งจากเครื่อง PLC ที่เขียนตามข้อกำหนดในการสั่งให้ลิฟท์ทำงานและมีความสัมพันธ์กับ Hardware (ลิฟท์จำลอง)

- ภาษามูลิน ที่ใช้แสดงความสัมพันธ์โปรแกรม Visual Basic กับการทำงานของ Hardware (ลิฟท์จำลอง)
- โปรแกรมแปลงเลขฐานสอง

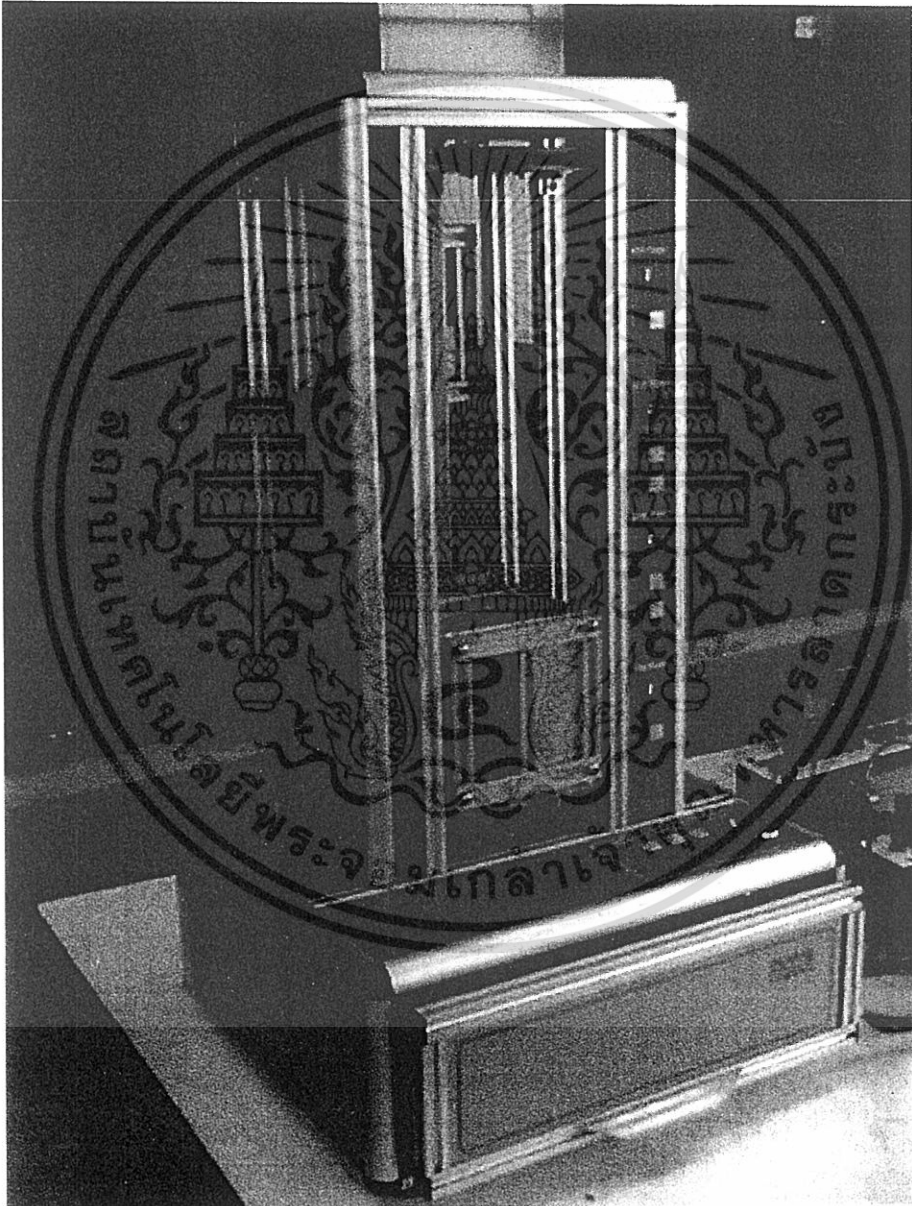
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

การทดลองและผลการทดลอง

4.1 ส่วนประกอบทางด้าน Hardware

4.1.1 Lift

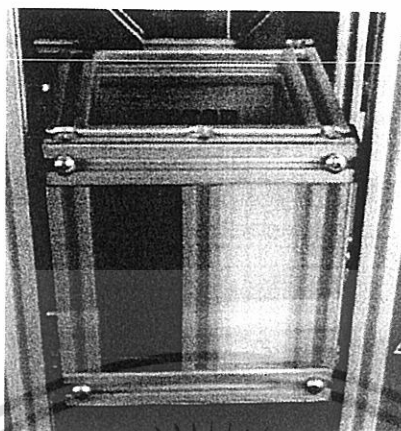


รูปที่ 4.1 ลิฟท์จำลองที่มีเครื่อง PLC ยี่ห้อ FACON เป็นตัวสั่งการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

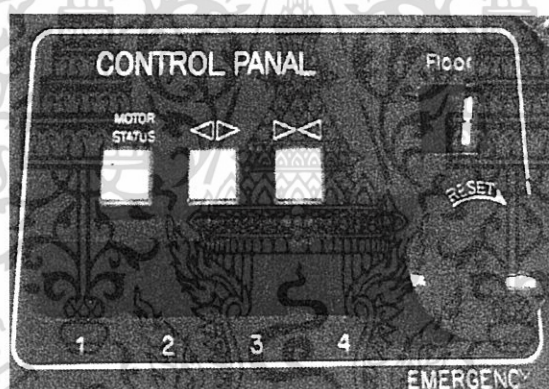
ส่วนประกอบของ Lift

1. Lift



รูปที่ 4.2 Lift

2. Control Panal



รูปที่ 4.3 Control Panal

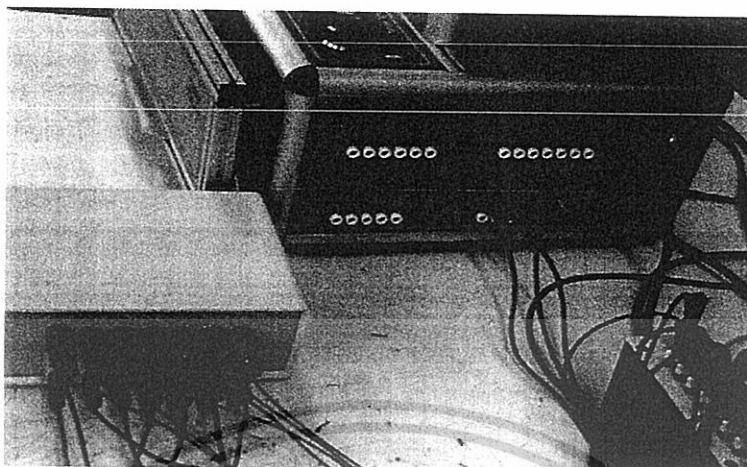
3. ปุ่มกดชั้น



รูปที่ 4.4 ปุ่มกดชั้น

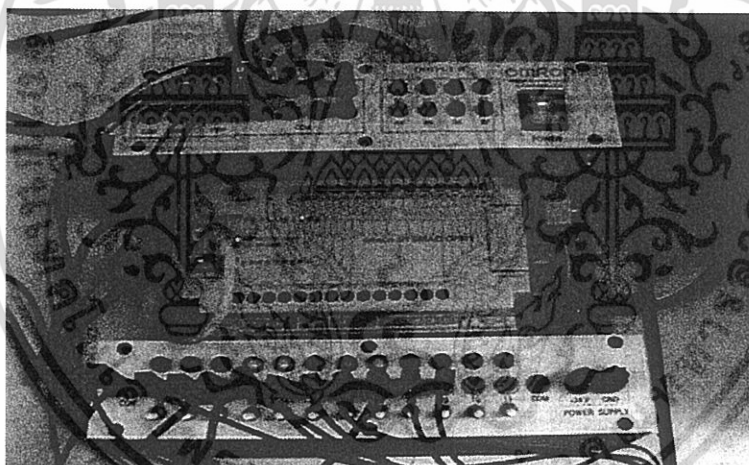
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. ส่วนเชื่อมต่อกับ PLC



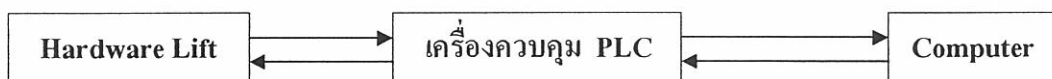
รูปที่ 4.5 ส่วนเชื่อมต่อกับ PLC

4.1.2 เครื่องควบคุม PLC ยี่ห้อ OMRON



รูปที่ 4.6 เครื่องควบคุม PLC ยี่ห้อ OMRON

4.2 การเชื่อมต่อ



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.7 แสดงการเชื่อมต่อ

4.3 ตัวอย่างการทดลอง ส่วน Software

เมื่อมีการกดลิฟท์ขึ้นชั้น 4 ขณะที่ลิฟท์อยู่ชั้น 3



รูปที่ 4.8 กราฟฟิกในโปรแกรม Visual Basic ที่แสดงเมื่อมีการกดลิฟท์ขึ้นชั้น 4 ขณะที่ลิฟท์อยู่ชั้น 3 ในหน้าของ Check Bit Status



รูป 4.9 แสดงการทำงานของลิฟท์ใน Bit ต่างๆ

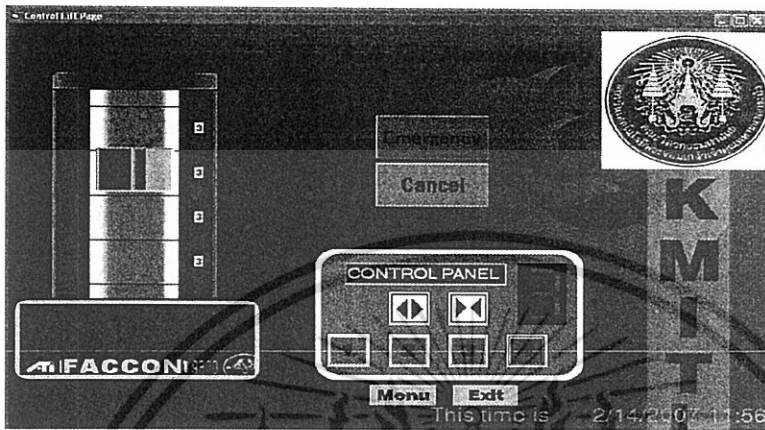
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อธิบายความหมายในรูป 4.9 คือ

Bit ที่ 000.03 แสดงประตูลิฟท์ที่ปิด

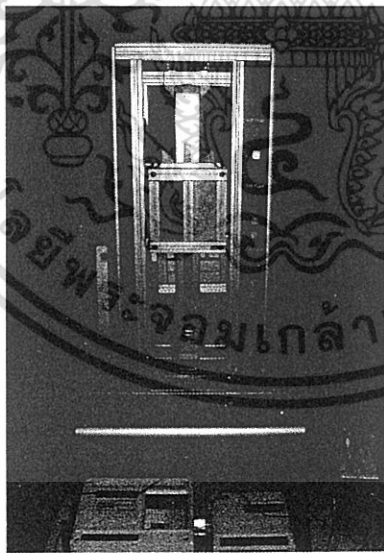
Bit ที่ 000.09 แสดงมอเตอร์ประตูลิฟท์ที่ปิด

Bit ที่ 009.05 แสดงตำแหน่งลิฟท์ว่าอยู่ชั้น 3



รูป 4.10 กราฟฟิกในโปรแกรม Visual Basic ที่แสดงเมื่อมีการกดลิฟท์ขึ้นชั้น 4 ขณะที่ลิฟท์อยู่ชั้น 3 ในหน้าของ Control Lift

ส่วน Hardware



รูป 4.11 มีการกดลิฟท์ขึ้นชั้น 4 ขณะที่ลิฟท์อยู่ชั้น 3 ส่วน Hardware

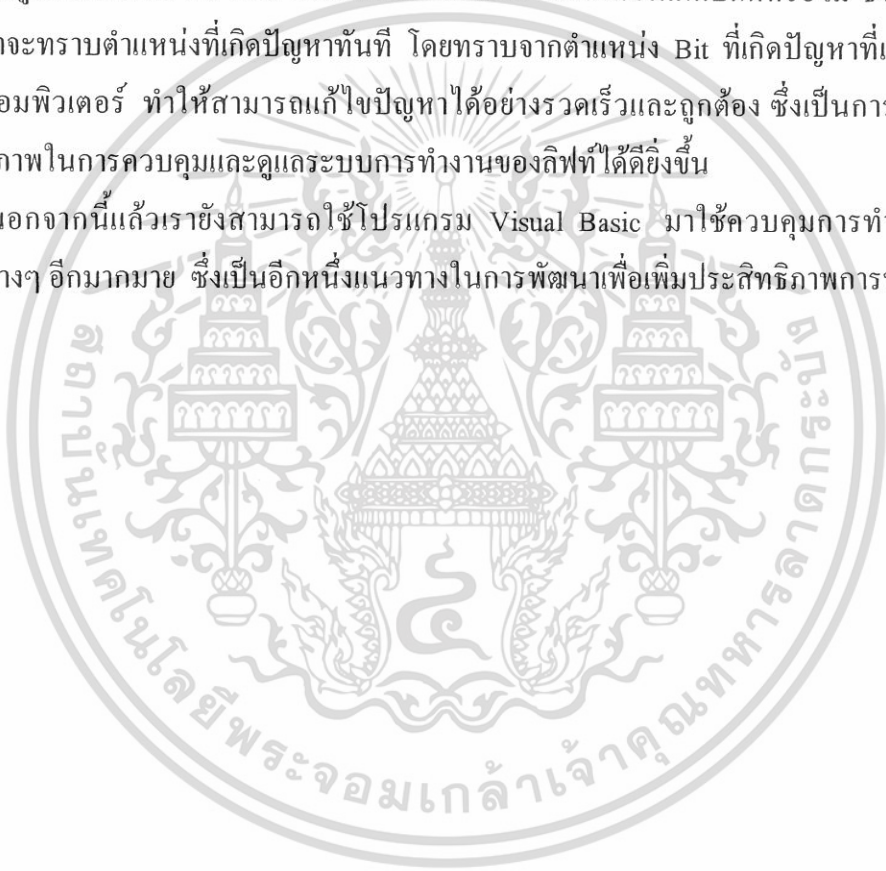
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

สรุปผลการวิจัยและข้อเสนอแนะ

ปฏิญานินท์เล่มนี้เป็นการเขียนโปรแกรม Visual Basic เพื่อใช้สำหรับเชื่อมต่อกับเครื่องควบคุม PLC และแสดงแอปพลิเคชันการทำงานของลิฟท์ให้สัมพันธ์แล้วกับ Hardware (ลิฟท์จำลอง) ที่นำมาต่อโดยมีการแสดงการรับ - ส่งค่าคำสั่งทั้งจากส่วนโปรแกรม Visual Basic และจากส่วนของ Hardware (ลิฟท์จำลอง) มาแสดงทางหน้าจอคอมพิวเตอร์ ซึ่งสามารถนำไปใช้งานในการดูแลขั้นตอนการทำงานของลิฟท์ระหว่างทำงานว่ามีสิ่งใดผิดปกติหรือไม่ ซึ่งเมื่อลิฟท์เกิดปัญหาจะทราบตำแหน่งที่เกิดปัญหาทันที โดยทราบจากตำแหน่ง Bit ที่เกิดปัญหาที่แสดงทางหน้าจอคอมพิวเตอร์ ทำให้สามารถแก้ไขปัญหานั้นได้อย่างรวดเร็วและถูกต้อง ซึ่งเป็นการช่วยเพิ่มประสิทธิภาพในการควบคุมและดูแลระบบการทำงานของลิฟท์ได้ดียิ่งขึ้น

นอกจากนี้แล้วเรายังสามารถใช้โปรแกรม Visual Basic มาใช้ควบคุมการทำงานของอุปกรณ์ต่างๆ อีกมากมาย ซึ่งเป็นอีกหนึ่งแนวทางในการพัฒนาเพื่อเพิ่มประสิทธิภาพการทำงาน



บรรณานุกรม

- [1] สัจจะ จรัสรุ่งรวี. บริษัท ไอ ดี ซี อินโฟ ดิสทริบิวเตอร์ เซ็นเตอร์ จำกัด คู่มือการเขียนโปรแกรม และใช้งาน Visual Basic 6.
- [2] อภิชาติ ภูพลับ.บริษัท ไอ ดี ซี อินโฟ ดิสทริบิวเตอร์เซ็นเตอร์ จำกัด เริ่มต้นการเขียนโปรแกรม ติดต่อและควบคุมฮาร์ดแวร์ ด้วย Visual Basic 6
- [3] รศ.ธีรศิลป์ ทุมวิภาต.บริษัท ซีเอ็ดยูนิเคชั่น จำกัด เรียนรู้ PLC ขั้นต้นด้วยตนเอง
- [4] นายทวีพล ชื้อสัตย์ วิทยานิพนธ์การออกแบบเครื่องควบคุมแบบตรรกและระบบโครงข่ายผ่าน โสคคอมพิวเตอร์



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

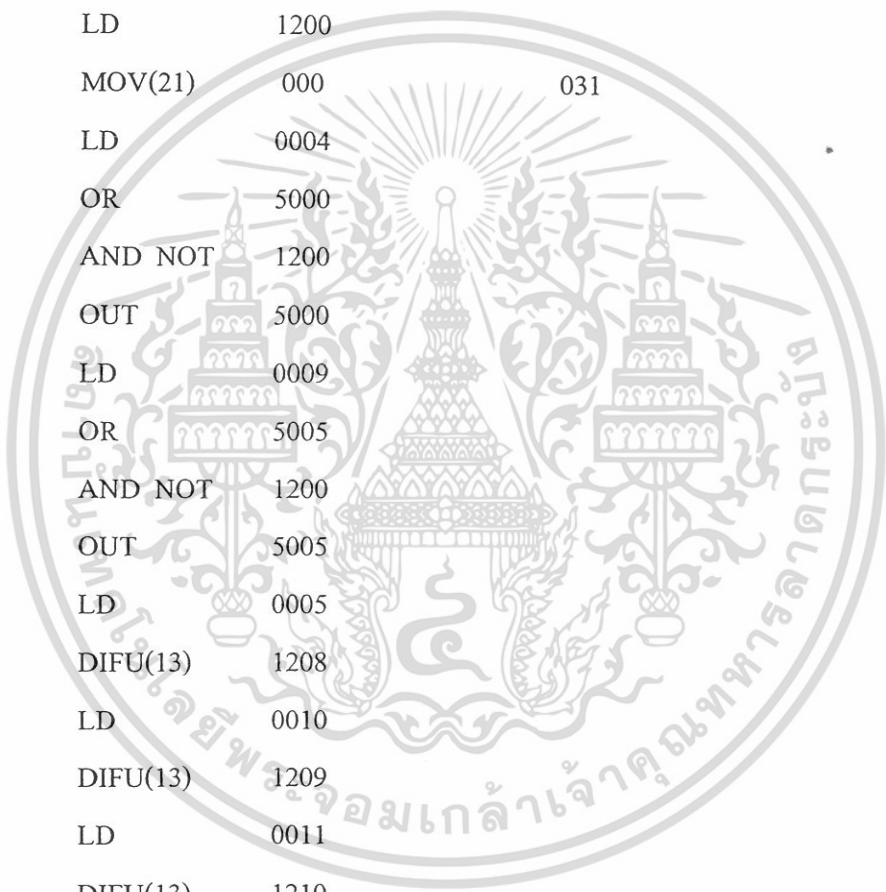


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Address	Instruction	Area	Timer/Bit	Control/Data	Movement/Comparison
00000	LD NOT	6500			
00001	AND NOT	7908			
00002	WRO(71)	000		050	
00003	LD	0013			
00004	OUT	7908		0009	
00005	LD	0009			
00006	MOV(21)	FFF		031	
00007	LD	0010			
00008	MOV(21)	FFF		032	
00009	LD	0011			
00010	MOV(21)	FFF		032	
00011	LD	0012			
00012	MOV(21)	FFF		042	
00013	LD	0004			
00014	MOV(21)	FFF		031	
00015	LD	0005			
00016	AND	1706			
00017	MOV(21)	FFF		042	
00018	LD	0005			
00019	AND NOT	1706			
00020	MOV(21)	FFF		032	
00021	LD	0006			
00022	AND	1706			
00023	MOV(21)	FFF		033	
00024	LD	0006			
00025	AND NOT	1706			
00026	MOV(21)	FFF		033	
00027	LD	0101			
00028	MOV(21)	FFF		043	
00029	LD	0100			
00030	MOV(21)	FFF		034	

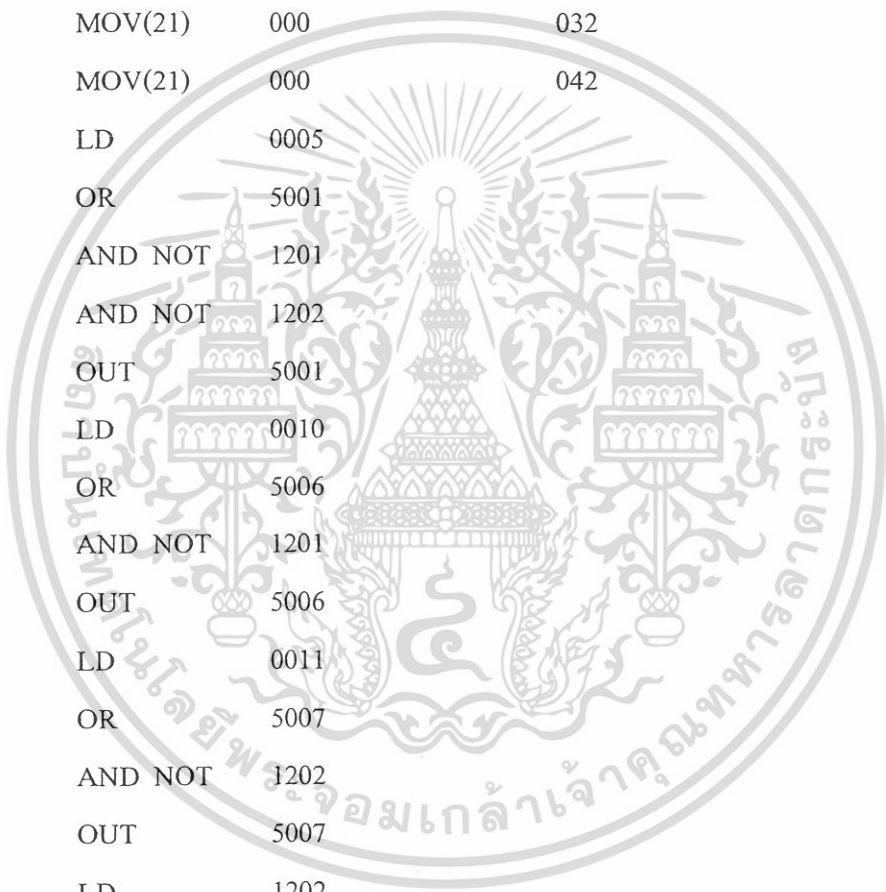
เอกสารนี้เป็นเอกสารที่จัดทำขึ้นเพื่อการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Address	Instruction	Area	Timer/Bit	Control/Data	Movement/Comparison
00031	LD	0102			
00032	MOV(21)	FFF		034	
00033	LD	5000			
00034	OR	5005			
00035	CMP(20)	DM000		001	
00036	LD	7906			
00037	DIFU(13)	1200			
00038	LD	1200			
00039	MOV(21)	000		031	
00040	LD	0004			
00041	OR	5000			
00042	AND NOT	1200			
00043	OUT	5000			
00044	LD	0009			
00045	OR	5005			
00046	AND NOT	1200			
00047	OUT	5005			
00048	LD	0005			
00049	DIFU(13)	1208			
00050	LD	0010			
00051	DIFU(13)	1209			
00052	LD	0011			
00053	DIFU(13)	1210			
00054	LD	1208			
00055	OR	1209			
00056	OR	1210			
00057	OR	5001			
00058	OR	5006			
00059	OR	5007			
00060	CMP(20)	DM000		002	
00061	LD	7906			



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับกวดขันงานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Address	Instruction	Area	Timer/Bit	Control/Data	Movement/Comparison
00062	AND NOT	1706			
00063	DIFU(13)	1201			
00064	LD	7906			
00065	AND	1706			
00066	DIFU(13)	1202			
00067	LD	1201			
00068	OR	1202			
00069	MOV(21)	000		032	
00070	MOV(21)	000		042	
00071	LD	0005			
00072	OR	5001			
00073	AND NOT	1201			
00074	AND NOT	1202			
00075	OUT	5001			
00076	LD	0010			
00077	OR	5006			
00078	AND NOT	1201			
00079	OUT	5006			
00080	LD	0011			
00081	OR	5007			
00082	AND NOT	1202			
00083	OUT	5007			
00084	LD	1202			
00085	OR	1214			
00086	AND	1706			
00087	OUT	1214			
00088	LD	1201			
00089	OR	1215			
00090	AND NOT	1706			
00091	OUT	1215			



เอกสารนี้เป็นเอกสารที่LD 00062-00091 00062-00091 เพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Address	Instruction	Area	Timer/Bit	Control/Data	Movement/Comparison
00093	DIFU(13)	2206			
00094	LD	0006			
00095	DIFU(13)	2212			
00096	LD	0101			
00097	DIFU(13)	2201			
00098	LD	2206			
00099	OR	2212			
00100	OR	2201			
00101	OR	5002			
00102	OR	5008			
00103	OR	5009			
00104	CMP(20)	DM000		003	
00105	LD	2206			
00105	OR	2212			
00106	OR	2212			
00107	AND	7916			
00108	AND NOT	1706			
00109	DIFU(13)	6200			
00110	LD	2206			
00111	OR	2201			
00112	AND	7906			
00113	AND	1706			
00114	DIFU(13)	6201			
00115	LD	6200			
00116	OR	6201			
00117	MOV(21)	000		033	
00118	MOV(21)	000		043	
00119	LD	0006			
00120	OR	5002			
00121	AND NOT	6200			
00122	AND NOT	6201			

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้สำหรับใช้ภายในเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Address	Instruction	Area	Timer/Bit	Control/Data	Movement/Comparison
00123	AND NOT	TIM040			
00124	OUT	5002			
00125	LD	5002			
00126	OR	5008			
00127	OR	5009			
00128	AND NOT	1700			
00129	TIM	040		020	
00130	LD	0012			
00131	OR	5008			
00132	AND NOT	6200			
00133	AND NOT	TIM040			
00134	OUT	5008			
00135	LD	0101			
00136	OR	5009			
00137	AND NOT	6201			
00138	AND NOT	TIM040			
00139	OUT	5009			
00140	LD	6201			
00141	OR	6214			
00142	AND	1706			
00143	OUT	6214			
00144	LD	6200			
00145	OR	6215			
00146	AND NOT	1706			
00147	OUT	6215			
00148	LD	5010			
00149	OR	5011			
00150	CMP(20)	DM000		004	
00151	LD	7906			
00152	DIFU(13)	6315			

เอกสารนี้เป็นเอกสารที่ LD ไว้สำหรับภา 7906 เพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Address	Instruction	Area	Timer/Bit	Control/Data	Movement/Comparison
00154	MOV(21)	000			034
00155	LD	0100			
00156	OR	5011			
00157	AND NOT	6315			
00158	OUT	5011			
00159	LD	0102			
00160	OR	5010			
00161	OR	5010			
00162	OUT	5010			
00163	LD	0007			
00164	OR	5003			
00165	AND NOT	0003			
00166	AND NOT	1700			
00167	OUT	5003			
00168	LD	0008			
00169	OR	5004			
00170	AND NOT	0002			
00171	AND NOT	1700			
00172	OUT	5004			
00173	LD	1706			
00174	CMP(20)	FFF			DM042
00175	LD	7906			
00176	AND NOT	1214			
00177	OUT	4000			
00178	LD	1706			
00179	CMP(20)	002			DM000
00180	LD	7906			
00181	OUT	4012			
00182	MOV(21)	000			042
00183	LD	4000			
00184	OR	4004			

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Address	Instruction	Area	Timer/Bit Control/Data Movement/Comparison
00185	AND NOT	4012	
00186	OUT	4004	
00187	LD	4004	
00188	AND	0003	
00189	OUT	4104	
00190	LD	4000	
00191	OR	4512	
00192	AND NOT	0002	
00193	OUT	4512	
00194	LD	1706	
00195	CMP(20)	FFF	043
00196	LD	7906	
00197	AND NOT	6214	
00198	OUT	5100	
00199	LD	1706	
00200	CMP(20)	003	DM000
00201	LD	7906	
00202	OUT	5112	
00203	MOV(21)	000	043
00204	LD	5100	
00205	OR	5104	
00206	AND NOT	5112	
00207	AND NOT	4512	
00208	OUT	5104	
00209	LD	5104	
00210	AND	0003	
00211	AND NOT	4512	
00212	OUT	5204	
00213	LD	5100	
00214	OR	5512	
00215	AND NOT	0002	

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Address	Instruction	Area	Timer/Bit	Control/Data	Movement/Comparison
00216	AND NOT	4516			
00217	OUT	5512			
00218	LD NOT	1706			
00219	CMP(20)	FFF		033	
00220	LD	7906			
00221	AND NOT	6215			
00222	OUT	4001			
00223	LD NOT	1706			
00224	CMP(20)	003		DM000	
00225	LD	7906			
00226	OUT	4013			
00227	MOV(21)	000		033	
00228	LD	4001			
00229	OR	4005			
00230	AND NOT	4013			
00231	OUT	4005			
00232	LD	4005			
00233	AND	0003			
00234	OUT	4105			
00235	LD	4001			
00236	OR	4515			
00237	AND NOT	0002			
00238	OUT	4515			
00239	LD	1706			
00240	CMP(20)	FFF		034	
00241	LD	1704			
00245	CMP(20)	004		DM000	
00246	LD	7906			
00247	OUT	4009			
00248	MOV(21)	000		034	
00249	LD	4008			

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Address	Instruction	Area	Timer/Bit	Control/Data	Movement/Comparison
00250	OR	4010			
00251	AND NOT	4009			
00252	AND NOT	5512			
00253	AND NOT	4512			
00254	OUT	4010			
00255	LD	4010			
00256	AND	0003			
00257	AND NOT	5512			
00258	AND NOT	4512			
00259	OUT	5105			
00260	LD	4008			
00261	OR	5114			
00262	AND NOT	0002			
00263	AND NOT	5512			
00264	AND NOT	4512			
00265	OUT	5114			
00266	LD	5114			
00267	AND	0003			
00268	MOVB(82)	000	DM005		DM051
00269	LD NOT	1706			
00270	CMP(20)	FFF		032	
00271	LD	7906			
00272	AND NOT	1215			
00273	OUT	4002			
00274	LD NOT	1706			
00275	CMP(20)	002		DM000	
00276	LD	7906			
00277	OUT	4014			
00278	MOV(21)	000		032	
00279	LD	4002			
00280	OR	4006			

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Address	Instruction	Area	Timer/Bit	Control/Data	Movement/Comparison
00281	AND NOT	4014			
00282	AND NOT	4514			
00283	OUT	4006			
00284	LD	4006			
00285	AND	0003			
00286	AND NOT	4513			
00287	AND NOT	5114			
00288	OUT	4106			
00289	LD	4002			
00290	OR	4513			
00291	AND NOT	0002			
00292	AND NOT	4514			
00293	AND NOT	5114			
00294	OUT	4513			
00295	LD NOT	1706			
00296	CMP(20)	FFF		031	
00297	LD	7906			
00298	OUT	4003			
00299	LD NOT	1706			
00300	CMP(20)	001		DM000	
00301	LD	7906			
00302	OUT	4015			
00303	MOV(21)	000		031	
00304	LD	4003			
00305	OR	4007			
00306	AND NOT	4015			
00307	AND NOT	5114			
00308	AND NOT	4513			
00309	AND NOT	4514			
00310	AND NOT	5114			
00311	OUT	4007			

เอกสารนี้เป็นเอกสารที่จัดทำไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Address	Instruction	Area	Timer/Bit	Control/Data	Movement/Comparison
00312	LD	4007			
00313	AND	0003			
00314	AND NOT	4513			
00315	AND NOT	5114			
00316	AND NOT	4514			
00317	OUT	4107			
00318	LD NOT	1706			
00319	CMP(20)	DM000		001	
00320	LD NOT	1706			
00321	AND	7906			
00322	MOVB(82)	001	600		017
00323	LD	1706			
00324	CMP(20)	DM000		004	
00325	LD	1706			
00326	AND	7906			
00327	MOVB(82)	000	600		017
00328	LD	1304	LD		1304
00329	OR	4104			
00330	OR	4105			
00331	OR	4106			
00332	OR	4107			
00333	OR	5105			
00334	OR	5204			
00335	OUT	1700			
00336	OUT	1705			
00337	LD	1700			
00338	AND NOT	1706			
00339	AND	0003			
00340	OUT	0704			
00341	LD	1700			
00342	AND	1706			

เอกสารที่สงวนลิขสิทธิ์ไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Address	Instruction	Area	Timer/Bit	Control/Data	Movement/Comparison
00343	AND	0003			
00344	OUT	0706			
00345	LD	0000			
00346	DIFU(14)	1600			
00347	LD	1600			
00348	OR	1601			
00349	OUT	1601			
00350	LD	1601			
00351	AND	0000			
00352	DIFU(13)	1400			
00353	LD	1400			
00354	AND	1706			
00355	INC(38)	DM001			
00356	LD	1400			
00357	AND NOT	1706			
00358	DEC(39)	DM001			
00359	LD	1400			
00360	CMP(20)	DM001		001	
00361	LD	7906			
00362	MOV(21)	001		DM000	
00363	LD	1400			
00364	CMP(20)	DM001		002	
00365	LD	7906			
00366	MOV(21)	002		DM000	
00367	LD	1400			
00368	CMP(20)	DM001		003	
00369	LD	7906			
00370	MOV(21)	003		DM000	
00371	LD	1400			
00372	CMP(20)	DM001		004	

เอกสารนี้เป็นเอกสารที่ LD 7906 ให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Address	Instruction	Area	Timer/Bit	Control/Data	Movement/Comparison
00374	MOV(21)	004			
00375	LD NOT	1401			
00376	MOVB(83)	DM000	200		007
00377	DIFU(13)	1402			
00378	LD	1400			
00379	OR	1402			
00380	OUT	0707			
00381	LD	0007			
00382	OR	2007			
00383	AND NOT	0003			
00384	AND NOT	1700			
00385	OUT	2007			
00386	LD	0008			
00387	OR	2008			
00388	AND NOT	0002			
00389	AND NOT	1700			
00390	OUT	2008			
00391	LD TIM	000			
00392	OR	2007			
00393	OR	4004			
00394	OR	4005			
00395	OR	4006			
00396	OR	4007			
00397	OR	4010			
00398	OR	5104			
00399	AND NOT	0003			
00400	OUT	0701			
00401	LD	1705			
00402	AND	0003			
00403	DIFU(14)	1101			

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Address	Instruction	Area	Timer/Bit	Control/Data	Movement/Comparison
00404	LD	1101			
00405	OR	0700			
00406	OR	2008			
00407	AND NOT	0002			
00408	OUT	0700			
00409	LD NOT	5900			
00410	CMP(20)	DM001		001	
00411	LD	7905			
00412	OR	6000			
00413	OUT	6000			
00414	LD NOT	5901			
00415	CMP(20)	DM001			
00416	LD	7906			
00417	OR	6001			
00418	OUT	6001			
00419	LD	6000			
00420	AND NOT	6001			
00421	OUT	6015			
00422	LD	0005			
00423	AND	1706			
00424	OR	0011			
00425	CMP(20)	DM000		001	
00426	LD	0005			
00427	OR	0011			
00428	AND	7906			
00429	OUT	6700			
00430	LD	0005			
00431	AND NOT	1706			
00432	OR	0010			
00433	CMP(20)	DM000		001	

เอกสารที่ 00434 LD 0005 นี้เป็นเอกสารที่จัดทำขึ้นไว้สำหรับภาคเรียนที่ 1 ปีการศึกษา 2564 เพื่อใช้ในการเรียนการสอนเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Address	Instruction	Area	Timer/Bit	Control/Data	Movement/Comparison
00435	OR	0010			
00436	AND	7906			
00437	OUT	6701			
00438	LD	0006			
00439	AND	1706			
00440	OR	0101			
00441	CMP(20)	DM000			
00442	LD	0006			
00443	OR	0101			
00444	AND	7906			
00445	OUT	6702			
00446	LD	0006			
00447	AND NOT	1706			
00448	OR	0012			
00449	CMP(20)	DM000		003	
00450	LD	0006			
00451	OR	0012			
00452	AND	7906			
00453	OUT	6703			
00454	LD	0100			
00455	AND	1706			
00456	OR	0102			
00457	CMP(20)	DM000		004	
00458	LD	0100			
00459	OR	0102			
00460	AND	7907			
00461	OUT	6704			
00462	LD	0704			
00463	OR	0706			
00464	DIFU(14)	2500			
00465	LD	2500			

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Address	Instruction	Area	Timer/Bit	Control/Data	Movement/Comparison
00466	OR	6015			
00467	OR	1115			
00468	AND NOT	0004			
00469	AND NOT	6700			
00470	AND NOT	6701			
00471	AND NOT	6702			
00472	AND NOT	6703			
00473	AND NOT	6704			
00474	AND NOT	5204			
00475	AND NOT	5105			
00476	AND NOT	4105			
00477	AND NOT	4106			
00478	AND NOT	4107			
00479	AND NOT	4114			
00480	OUT	1115			
00481	LD	1115			
00482	CMP(20)	DM001		001	
00483	LD	7905			
00484	OR	1113			
00485	AND	1115			
00486	OUT	1113			
00487	LD	1113			
00488	AND NOT	5003			
00489	AND NOT	5004			
00490	TIM	000		100	
00491	LD TIM	000			
00492	AND	0002			
00493	OR	1304			
00494	AND NOT	1114			
00495	OUT	1304			
00496	MOVB(82)	000	000	600	017

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้สำหรับใช้ในงานเพื่อการศึกษาค้นคว้าเท่านั้น ไม่อนุญาติให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Address	Instruction	Area	Timer/Bit	Control/Data	Movement/Comparison
00497	CMP(20)	DM001			001
00498	LD	7906			
00499	OR	1114			
00500	AND	1115			
00501	OUT	1114			
00502	LD	0013			
00503	AND	0007			
00504	AND	0008			
00505	MOV(21)	001			DM000
00506	MOV(21)	001			DM001
00507	MOV(21)	000			031
00508	MOV(21)	000			032
00509	MOV(21)	000			033
00510	MOV(21)	000			034
00511	MOV(21)	000			042
00512	MOV(21)	000			043
00513	END(01)				



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาคผนวก ข
CQM1/CPM1/CPM1A/SRM1 Programmable Controllers
Programming Manual (OMRON)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3-1-9 DM Area

Data is accessed in word units. As shown below, the DM area contains both an area that can be freely used and areas with specific functions.

DM0000		This area has no specific functions and can be used freely. Both reading and writing are possible from the program.
DM1024		
(see note 1)	Fixed DM (see note 2)	This area cannot be written from the program. It is used for storing information that is not to be changed. Writing can be executed by means of peripheral devices only.
DM6144		
DM6569	Error Log	This area stores the error log. The user can only read this area, and cannot write into it.
DM6600	PC Setup	This area stores information related to CQM1 operation. The settings are made by means of peripheral devices.
DM6655	(see note 2)	

- Note**
1. The CQM1-CPU11-E and CQM1-CPU21-E do not support DM 1024 through DM 6143.
 2. Turning ON pin 1 of the DIP switch on the CPU Unit will prevent writing even by means of peripheral devices.

Fixed DM contents, the PC Setup, the user program, and the instructions table can all be saved to and loaded from a Memory Cassette as a single unit. Refer to page 146 for details.



Caution

Although the CQM1-CPU11-E and CQM1-CPU21-E do not support DM 1024 through DM 6143, an error will not occur if they are addressed. Any attempt to write to these words will have no effect and any reads will produce all zeros.

3-1-10 UM Area

The UM area stores the user's program. UM area contents can be read and written only as program data, and not as words. The following table shows the size of the UM area in the CQM1 CPU Units.

CPU Unit Model	UM area size
CQM1-CPU11/21-E	3.2 KW
CQM1-CPU4□-EV1	7.2 KW

3-2 CPM1/CPM1A Memory Area Functions

3-2-1 Memory Area Structure

The following memory areas can be used with the CPM1/CPM1A.

Data area		Words	Bits	Function
IR area ¹	Input area	IR 000 to IR 009 (10 words)	IR 00000 to IR 00915 (160 bits)	These bits can be allocated to the external I/O terminals.
	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)	Work bits can be freely used within the program.
SR area		SR 232 to SR 255 (24 words)	SR 23200 to SR 25515 (384 bits)	These bits serve specific functions such as flags and control bits.
TR area		---	TR 0 to TR 7 (8 bits)	These bits are used to temporarily store ON/OFF status at program branches.
HR area ²		HR 00 to HR 19 (20 words)	HR 0000 to HR 1915 (320 bits)	These bits store data and retain their ON/OFF status when power is turned off.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Data area		Words	Bits	Function
AR area ²		AR 00 to AR 15 (16 words)	AR 0000 to AR 1515 (256 bits)	These bits serve specific functions such as flags and control bits.
LR area ¹		LR 00 to LR 15 (16 words)	LR 0000 to LR 1515 (256 bits)	Used for a 1:1 data link with another PC.
Timer/Counter area ²		TC 000 to TC 127 (timer/counter numbers) ³		The same numbers are used for both timers and counters.
DM area	Read/write ²	DM 0000 to DM 0999 DM 1022 to DM 1023 (1,002 words)	---	DM area data can be accessed in word units only. Word values are retained when the power is turned off.
	Error log	DM 1000 to DM 1021 (22 words)	---	Used to store the time of occurrence and error code of errors that occur. These words can be used as ordinary read/write DM when the error log function isn't being used.
	Read-only ⁴	DM 6144 to DM 6599 (456 words)	---	Cannot be overwritten from program.
	PC Setup ⁴	DM 6600 to DM 6655 (56 words)	---	Used to store various parameters that control PC operation.

- Note**
1. IR and LR bits that are not used for their allocated functions can be used as work bits.
 2. The contents of the HR area, LR area, Counter area, and read/write DM area are backed up by a capacitor. The backup time varies with the ambient temperature, but at 25°C, the capacitor will back up memory for 20 days. If the power supply is off longer than the backup time, memory contents will be cleared and AR1314 will turn ON. (This flag turns ON when data can no longer be retained by the built-in capacitor.) Refer to 2-1-2 *Characteristics* in the *CPM1 and CPM1A Operation Manual* for a graph showing the backup time vs. temperature.
 3. When accessing a PV, TC numbers are used as word data; when accessing Completion Flags, they are used as bit data.
 4. Data in DM 6144 to DM 6655 cannot be overwritten from the program, but they can be changed from a Peripheral Device.

3-2-2 IR Area

The functions of the IR area are explained below.

I/O Bits

IR area bits from IR 00000 to IR 01915 are allocated to terminals on the CPU Unit and I/O Units. They reflect the ON/OFF status of input and output signals. Input bits begin at IR 00000, and output bits begin at IR 01000.

The following table shows which IR bits are allocated to the I/O terminals on the CPM1 CPU Units and CPM1-20EDR I/O Unit.

CPM1 CPU Unit	I/O	CPU Unit Terminals	I/O Unit Terminals
CPM1-10CDR-□	Inputs	6 points: 00000 to 00005	12 points: 00100 to 00111
	Outputs	4 points: 01000 to 01003	8 points: 01100 to 01107
CPM1-20CDR-□	Inputs	12 points: 00000 to 00011	12 points: 00100 to 00111
	Outputs	8 points: 01000 to 01007	8 points: 01100 to 01107
CPM1-30CDR-□	Inputs	18 points: 00000 to 00011, 00100 to 00105	12 points: 00200 to 00211
	Outputs	12 points: 01000 to 01007, 01100 to 01103	8 points: 01200 to 01207
CPM1-30CDR-□ -V1	Inputs	18 points: 00000 to 00011, 00100 to 00105	36 points: 00200 to 00211 00300 to 00311 00400 to 00411
	Outputs	12 points: 01000 to 01007, 01100 to 01103	24 points: 01200 to 01207 01300 to 01407 01400 to 01407

The following table shows which IR bits are allocated to the I/O terminals on the CPM1A's CPU Units and Expansion I/O Unit.

Number of I/O terminals on the CPU Unit	CPU Unit terminals		CPM1A-20ED□ Expansion I/O Unit Terminals						Power supply	Model number
	Inputs	Outputs	Inputs	Outputs	Inputs	Outputs	Inputs	Outputs		
10	6 points: 00000 to 00005	4 points: 01000 to 01003	---	---	---	---	---	---	AC	CPM1A-10CD□-A
			---	---	---	---	---	---	DC	CPM1A-10CD□-D
20	12 points: 00000 to 00011	8 points: 01000 to 01007	---	---	---	---	---	---	AC	CPM1A-20CD□-A
			---	---	---	---	---	---	DC	CPM1A-20CD□-D
30	18 points: 00000 to 00011 00100 to 00105	12 points: 01000 to 01007 01100 to 01103	12 points: 00200 to 00211	8 points: 01200 to 01207	12 points: 00300 to 00311	8 points: 01300 to 01307	12 points: 00400 to 00411	8 points: 01400 to 01407	AC	CPM1A-30CD□-A
			---	---	---	---	---	---	DC	CPM1A-30CD□-D
40	24 points: 00000 to 00011 00100 to 00111	16 points: 01000 to 01007 01100 to 01107	12 points: 00200 to 00211	8 points: 01200 to 01207	12 points: 00300 to 00311	8 points: 01300 to 01307	12 points: 00400 to 00411	8 points: 01400 to 01407	AC	CPM1A-40CD□-A
			---	---	---	---	---	---	DC	CPM1A-40CD□-D

Work Bits

The work bits can be used freely within the program. They can only be used within the program, however, and not for direct external I/O.

3-2-3 SR Area

These bits mainly serve as flags related to CPM1/CPM1A operation or contain present and set values for various functions. For details on the various bit functions, refer to relevant sections in this manual or to *Appendix C Memory Areas*.

SR 244 to SR 247 can also be used as work bits, when input interrupts are not used in Counter Mode.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3-2-4 TR Area

When a complex ladder diagram cannot be programmed in mnemonic code just as it is, these bits are used to temporarily store ON/OFF execution conditions at program branches. They are used only for mnemonic code. When programming directly with ladder diagrams using the Ladder Support Software (LSS) or the SYSMAC Support Software (SSS), TR bits are automatically processed for you.

The same TR bits cannot be used more than once within the same instruction block, but can be used again in different instruction blocks. The ON/OFF status of TR bits cannot be monitored from a Peripheral Device.

Examples showing the use of TR bits in programming are provided on page 169.

3-2-5 HR Area

These bits retain their ON/OFF status even after the CPM1/CPM1A power supply has been turned off or when operation begins or stops. They are used in the same way as work bits.

3-2-6 AR Area

These bits mainly serve as flags related to CPM1/CPM1A operation. These bits retain their status even after the CPM1/CPM1A power supply has been turned off or when operation begins or stops. For details on the various bit functions, refer to relevant sections in this manual or to *Appendix C Memory Areas*.

3-2-7 LR Area

When the CPM1/CPM1A is linked one-to-one with another CPM1/CPM1A, a CQM1, an SRM1 or a C200HS PC, these bits are used to share data. For details, refer to page 101.

LR bits can be used as work bits when not used for data links.

3-2-8 Timer/Counter Area

This area is used to manage timers and counters created with TIM, TIMH(15), CNT, and CNTR(12). The same numbers are used for both timers and counters and each number can be used only once in the user program. Do not use the same TC number twice even for different instructions.

TC numbers are used to create timers and counters, as well as to access Completion Flags and present values (PVs). If a TC number is designated for word data, it will access the present value (PV); if it is used for bit data, it will access the Completion Flag for the timer/counter.

Refer to instructions beginning on page 208 for details on timers and counters.

3-2-9 DM Area

DM area data is accessed in word units only. The contents of the DM area are retained even after the CPM1/CPM1A power supply has been turned off or when operation begins or stops.

DM words DM 0000 through DM 0999, DM 1022, and DM 1023 can be used freely in the program; other DM words are allocated specific functions, described below.

Error Log

DM 1000 through DM 1021 contain the error log information. Refer to *Section 8 Troubleshooting* for details on the error log.

PC Setup

DM 6600 through DM 6655 contain the PC Setup. Refer to *1-5 PC Setup* for details.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้