

การออกแบบเครื่องควบคุมแบบตรรกะที่โปรแกรมได้

DESIGN OF PROGRAMMABLE LOGIC CONTROLLER



นายฐากร อรุณฤติรัตน์
นายธนวัฒน์ วานมนตรี
นายพิริศ สุขประเสริฐ



เลขหมึ.....
เลขทะเบียน..... 42514
วัน, เดือน, ปี 24 พ.ศ. 2543

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

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของวิชาการศึกษิตตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิชาวิศวกรรมการวัดคุม
ภาควิชาเทคโนโลยีการวัดคุมทางอุตสาหกรรม คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2543

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

DESIGN OF PROGRAMMABLE LOGIC CONTROLLER



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

2000

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาควิชาเทคโนโลยีการวัดคุมทางอุตสาหกรรม
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ใบรับรองปริญญาานิพนธ์

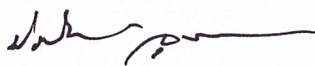
หัวข้อปริญญาานิพนธ์ การออกแบบเครื่องควบคุมแบบตรรกที่โปรแกรมได้
DESIGN OF PROGRAMMABLE LOGIC CONTROLLER

นักศึกษาผู้จัดทำ นายฐากร อรุณศิริรัตน์ รหัสประจำตัว 41013408
นายธนวัฒน์ วานมนตรี รหัสประจำตัว 41013413
นายพิสร สุขประเสริฐ รหัสประจำตัว 41013421

ปริญญา วิศวกรรมศาสตรบัณฑิต
สาขาวิชา วิศวกรรมการวัดคุม
ปีการศึกษา 2543

อาจารย์ผู้ควบคุมปริญญาานิพนธ์	ลายมือชื่อ
อาจารย์ทวีพล ชื้อสัตย์	

วัน/เดือน/ปี ที่สอบ วันจันทร์ที่ 9 เมษายน พ.ศ. 2544
สถานที่สอบ ณ. ห้องสอบปริญญาานิพนธ์ ภาควิชาเทคโนโลยีการวัดคุมทางอุตสาหกรรม

ภาควิชารับรองแล้ว

(ผศ.ประสิทธิ์ จุลเสรีวงศ์)
หัวหน้าภาควิชา ฯ

หัวข้อปริญญานิพนธ์	การออกแบบเครื่องควบคุมแบบตรรกะที่โปรแกรมได้	
	DESIGN OF PROGRAMMABLE LOGIC CONTROLLER	
นักศึกษาผู้จัดทำ	นายฐากร	อรุณฤดีรัตน์
	นายชนวิวัฒน์	วานมนตรี
	นายพรสิร	สุขประเสริฐ
ปริญญา	วิศวกรรมศาสตรบัณฑิต	
สาขาวิชา	วิศวกรรมการวัดคุม	
ปีการศึกษา	2543	

บทคัดย่อ

ปริญญานิพนธ์นี้ ได้นำเสนอการออกแบบเครื่องควบคุมแบบตรรกะที่โปรแกรมได้ โดยทางด้านฮาร์ดแวร์ประกอบด้วยไมโครคอนโทรลเลอร์เบอร์ AT89C52 เป็นตัวประมวลผลกลาง มีจำนวนหน่วยอินพุต 12 จุด และเอาต์พุต 12 จุด และมีซอฟต์แวร์ประยุกต์สำหรับช่วยในการเขียนโปรแกรมที่ทำงานเป็นลำดับ (SEQUENTIAL) และมีฟังก์ชันการตรวจสอบการทำงานของเครื่อง นอกจากนี้เครื่องควบคุมถูกออกแบบให้สามารถควบคุมหน่วยอินพุตและเอาต์พุตในระยะทางไกลเพื่ออำนวยความสะดวกในการใช้งาน โดยเป็นการสื่อสารตามมาตรฐาน RS-422 และสามารถขยายจำนวนอินพุตและเอาต์พุตได้ 4 จุด ระยะทางในการติดต่อควบคุมนับร้อยเมตร ดังนั้นการควบคุมเครื่องจักรที่ถูกจัดวางในสถานที่ที่ต่างกัน และมีระยะทางไกลออกไป จึงสามารถกระทำได้อย่างสะดวก และไม่ยุ่งยากมากนัก ซึ่งเป็นการเพิ่มประสิทธิภาพในการทำงานให้รวดเร็ว และครอบคลุมทั้งระบบ

Thesis	Design Of Programmable Logic Controller	
Authers	Mr. Thakorn	Aroonrudeerat
	Mr. Tanawat	Vanmontree
	Mr. Parit	Sukprasert
Thesis Advisor	Mr. Thaweepol	Suesut
Year	2000	

ABSTRACT

This is a thesis which presents a design of PLC with microcontroller NO.AT89C52 as the Central Processing Unit(CPU). This PLC consists of 12 inputs and 12 outputs and there is the software application for sequential program. Moreover,PLC is also designed to control input and output with remote I/O in order to support the convenience of long distance - controlling. The interface of network is designed by using the standard RS 422 which can expend all 4 terminals to link each other for the communication distance application with a hundred metre. Therefore ,the controlling of machine in different places and for long distance ,can be more easiler and more convenient so that we can work more efficiently to achieve the overall system .

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

ปริญญาานิพนธ์ฉบับนี้สำเร็จลุล่วงไปได้ เพราะความเมตตาจาก อาจารย์ทวีพล ชื้อสัตย์ ที่ได้ให้คำแนะนำแก่ผู้วิจัยตลอดมา อีกทั้งยังเอื้อเฟื้ออุปการณ์ และเครื่องมือต่าง ๆ ในการทำปริญญาานิพนธ์นี้ ผู้วิจัยรู้สึกซาบซึ้ง และขอกราบขอบพระคุณเป็นอย่างสูง

ขอขอบพระคุณ อาจารย์ภาควิชาเทคโนโลยีการวัดคุมทางอุตสาหกรรมทุกท่าน ที่ได้คำแนะนำอันเป็นประโยชน์ต่อการทำปริญญาานิพนธ์ฉบับนี้

และที่ลืมเสียมิได้คือ ขอกราบขอบพระคุณคุณแม่ อันเป็นที่รักยิ่ง ที่สนับสนุน และเป็นแรงบันดาลใจ อีกทั้งทุกท่านที่ได้คำแนะนำ ในการทำปริญญาานิพนธ์ฉบับนี้

คุณค่าและประโยชน์อันพึงมีจากปริญญาานิพนธ์ฉบับนี้ ผู้วิจัยขอมอบแด่ผู้มีพระคุณทุกท่าน



คณะผู้จัดทำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

	หน้า
บทคัดย่อภาษาไทย.....	I
บทคัดย่อภาษาอังกฤษ.....	II
กิตติกรรมประกาศ.....	III
สารบัญ.....	IV
สารบัญตาราง.....	VIII
สารบัญภาพ.....	IX
บทที่ 1 บทนำ.....	1
บทที่ 2 ทฤษฎีและหลักการพื้นฐานของเครื่องควบคุมแบบตรรกะที่โปรแกรมได้.....	2
2.1 ที่มาและวิวัฒนาการของเครื่องควบคุมแบบตรรกะที่โปรแกรมได้.....	2
2.1.1 วิวัฒนาการของเครื่องควบคุมแบบตรรกะที่โปรแกรมได้.....	3
2.1.2 สรุปข้อดีของ PLC / PC.....	3
2.2 โครงสร้างและหลักการทำงานของเครื่องควบคุมแบบตรรกะที่โปรแกรมได้.....	4
2.2.1 หน่วยประมวลผลกลาง.....	5
2.2.2 หน่วยอินพุต/เอาต์พุต.....	9
2.2.3 หน่วยป้อนโปรแกรม.....	13
2.3 ลักษณะการโปรแกรมให้กับเครื่องควบคุมแบบตรรกะที่โปรแกรมได้.....	14
2.3.1 โปรแกรมภาษาแลดเดอร์ไดอะแกรม.....	15
2.3.2 โปรแกรมคำสั่งภาษาแบบบูลีน.....	18
2.3.3 โปรแกรมคำสั่งภาษาบล็อก.....	18
2.3.4 โปรแกรมคำสั่งภาษาระดับสูง.....	19
2.4 การเลือกใช้ภาษา PC.....	19
2.5 ภาษาที่ใช้กับหน่วยลอจิกอินพุต / เอาต์พุต.....	20
2.6 หลักการเขียน LADDER DIAGRAM.....	23
2.7 การเขียนโปรแกรมคำสั่งบูลีนจาก LADDER DIAGRAM.....	29
บทที่ 3 เครื่องควบคุมแบบตรรกะที่โปรแกรมได้กับการติดต่อสื่อสาร.....	31
3.1 การติดต่อสื่อสารข้อมูลทั่วไป.....	31

สารบัญ (ต่อ)

หน้า

3.1.1	วิธีการส่งข้อมูล.....	31
3.1.2	ข้อตกลงในการติดต่อสื่อสาร (โปรโตคอล : Protocols).....	34
3.1.3	การควบคุมความผิดพลาดในการส่งข้อมูล (Error Control).....	35
3.1.4	อัตราการส่งข้อมูล (Data Transmission Rate).....	36
3.1.5	มาตรฐานในการสื่อสารข้อมูลแบบอนุกรม.....	37
3.2	รูปแบบการสื่อสารข้อมูลของเครื่องควบคุมแบบตรรกที่โปรแกรมได้.....	41
3.2.3	การต่อวงจรแบบจุดต่อจุด.....	41
3.2.4	การต่อวงจรแบบหลายจุด (Multi point).....	42
3.2.5	การต่อวงจรสื่อสารแบบเครือข่ายท้องถิ่น (Local area Network for PLC/PC).....	45
3.3	ข้อตกลงในการสื่อสารของเครื่องควบคุม PLC/PC.....	47
3.3.3	รูปแบบของบล็อก (Block Format).....	47
3.3.4	การคำนวณ FCS.....	48
บทที่ 4	การออกแบบและพัฒนาเครื่องควบคุมแบบตรรกที่โปรแกรมได้.....	50
4.1	คุณสมบัติของหน่วยประมวลผลกลางและหน่วยความจำ.....	50
4.1.1	ทางด้านฮาร์ดแวร์.....	50
4.1.2	ทางด้านซอฟต์แวร์.....	50
4.2	คุณสมบัติของหน่วยการรับส่งข้อมูล.....	51
4.2.1	การเชื่อมต่อจากพอร์ตของคอมพิวเตอร์ไปยังบอร์ดของหน่วยประมวลผล.....	51
4.2.2	การเชื่อมต่อจากบอร์ดของหน่วยประมวลผล ไปยังหน่วยอินพุท/เอาต์พุท.....	51
4.3	คุณสมบัติของหน่วยอินพุท/เอาต์พุท.....	52
4.4	การออกแบบวงจร.....	53
4.4.1	วงจรหน่วยประมวลผลกลางและหน่วยความจำ.....	53
4.4.2	วงจรเชื่อมต่อระหว่างคอมพิวเตอร์และหน่วยประมวลผล.....	54

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และห้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

หน้า

4.4.3	วงจรเชื่อมต่อระหว่างบอร์ดหน่วยประมวลผลและบอร์ด หน่วยอินพุท/เอาต์พุท.....	55
4.4.4	วงจรติดต่อกายนอกของบอร์ดอินพุท/เอาต์พุท.....	56
4.5	ระบบปฏิบัติการ.....	57
4.5.1	การแบ่งพื้นที่หน่วยความจำ.....	58
4.5.2	โปรแกรมผู้ใช้และคำสั่งสำหรับเครื่องควบคุม.....	59
4.5.3	การจัดข้อมูลของโปรแกรมผู้ใช้.....	60
บทที่ 5	ขั้นตอนและวิธีการดำเนินงาน.....	63
5.1	ส่วน Hardward.....	63
5.2	ส่วน Software.....	64
5.2.1	ขั้นตอนการดำเนินงาน.....	64
5.2.2	ส่วนประกอบของโปรแกรมในส่วน Software.....	64
5.3	การทำงาน.....	66
5.3.1	ส่วนการสร้าง Text File.....	66
5.3.2	การเปลี่ยนรหัสที่เป็น Text File เป็นรหัส Toggle Code.....	68
5.3.3	การส่ง Toggle Code ไปยังชุด Controller.....	70
5.3.4	การทำงานของโปรแกรมหลักของชุด Controller.....	70
บทที่ 6	การออกแบบโปรแกรมสนับสนุนการทำงานของเครื่องควบคุม PLC.....	74
6.1	คุณสมบัติทั่วไปของโปรแกรมสนับสนุนการทำงาน.....	74
6.2	การออกแบบโปรแกรมเพื่อป้อนโปรแกรมคำสั่งบูลีน.....	74
6.3	การออกแบบโปรแกรมเพื่อถ่ายเทข้อมูลจากเครื่องคอมพิวเตอร์ไปยัง เครื่องควบคุม.....	75
บทที่ 7	การประยุกต์การใช้งานและผลการทดสอบการทำงาน.....	77
7.1	การประยุกต์ใช้งานเครื่องควบคุมที่โปรแกรมได้.....	77

สารบัญ (ต่อ)

	หน้า
7.1.1 ส่วนประกอบของชุดจำลองสายพานลำเลียง.....	77
7.1.2 ขั้นตอนในการทำงานของชุดสายพานลำเลียง.....	78
7.1.3 การกำหนดอินพุต/เอาต์พุตของชุดสายพานลำเลียง.....	78
7.1.4 โปรแกรมแลคเคอร์โคอะแกรมของชุดสายพานลำเลียง.....	79
7.1.5 โปรแกรมชุดคำสั่งบูลีนของชุดสายพานลำเลียง.....	80
7.2 การประยุกต์ใช้งานโปรแกรมสนับสนุน.....	80
บทที่ 8 บทสรุป.....	85
บรรณานุกรม.....	86



สารบัญตาราง

ตารางที่	หน้า
2.1 แสดงอุปกรณ์ อินพุท/เอาต์พุทแบบค่าสถานะลอจิก.....	10
2.2 แสดงอุปกรณ์อินพุท/เอาต์พุทแบบค่าตัวเลข.....	11
2.3 แสดงอุปกรณ์อินพุท/เอาต์พุทสำหรับค่าสถานะแบบหลายบิต.....	11
2.4 แสดงถึงตัวอย่างของการใช้ฟังก์ชันลอจิก และสัญลักษณ์พื้นฐานในการเขียน โปรแกรม แลดเดอร์ของ PLC/PC.....	15
2.5 แสดงตัวอย่างโปรแกรมชุดคำสั่งบูลีน.....	18
3.1 แสดงค่าเปรียบเทียบคุณสมบัติทางไฟฟ้าของ RS-232 C , RS-422A และ RS-485.....	40
4.1 การแบ่งหน่วยความจำ.....	58
4.2 แสดงตารางอินพุท/เอาต์พุทและรีเลย์ภายใน.....	58
4.3 แสดงความสัมพันธ์ของสมการบูลีนกับชุดคำสั่งบูลีน.....	59
4.4 แสดงการเก็บข้อมูลของโปรแกรมผู้ใช้ในหน่วยความจำ.....	60
4.5 แสดงคำสั่งที่ได้ ออกแบบไว้.....	61
4.6 แสดงส่วนของหน่วยความจำที่เก็บค่าอินพุท เอาต์พุท และรีเลย์ภายใน.....	62
4.7 การจัดเก็บรหัสจากโปรแกรมคำสั่งบูลีน.....	62

สารบัญรูป

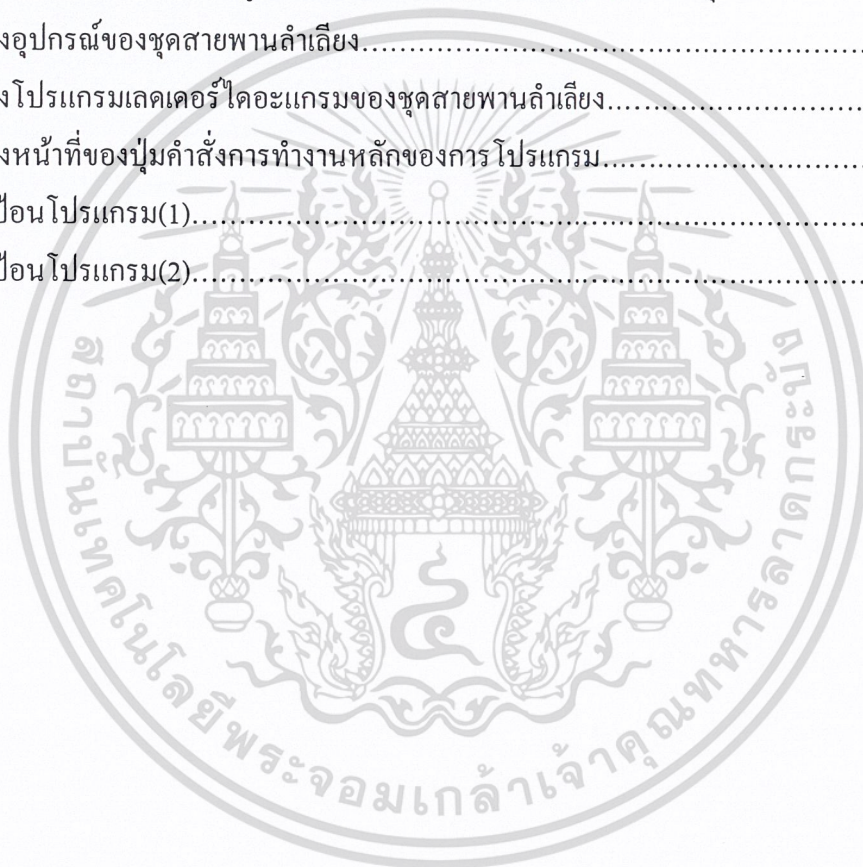
รูปที่	หน้า
2.1 แสดงบล็อกไคอะแกรมของ PLC/PC.....	5
2.2 แสดงบล็อกไคอะแกรมของ CPU.....	6
2.3 แสดงการ SCAN ของ CPU.....	6
2.4 แสดงข้อมูลเป็น Bit Byte และ Word.....	7
2.5 แสดงหน่วยความจำของ PLC/PC.....	8
2.6 แสดงหน่วยความจำสำหรับผู้ใช้.....	8
2.7 แสดงบล็อกไคอะแกรมของหน่วยอินพุทเอาต์พุท.....	9
2.8 แสดงเครื่องป้อนโปรแกรมแบบ CRT.....	13
2.9 แสดงเครื่องป้อนโปรแกรมขนาดเล็ก หรือพกพา.....	14
2.10 แสดงวงจรควบคุมการหมุนกลับทางของมอเตอร์แบบ JOGGING.....	16
2.11 แสดงแลคเคอร์ไคอะแกรมควบคุมการหมุนกลับทางของมอเตอร์แบบ JOGGING.....	17
2.12 แสดงการแบ่งแลคเคอร์ไคอะแกรมออกเป็นรังค์.....	17
2.13 แสดงหมายเลขกำกับอุปกรณ์ต่างๆ ของ PLC เครื่องหนึ่ง.....	24
2.14 แสดงตัวอย่าง Ladder Diagram.....	24
2.15 แสดงตัวอย่าง Ladder Diagram.....	25
2.16 แสดงตัวอย่าง Ladder Diagram.....	25
2.17 แสดงตัวอย่าง Ladder Diagram.....	26
2.18 แสดงตัวอย่าง Ladder Diagram.....	26
2.19 แสดงตัวอย่าง Ladder Diagram.....	27
2.20 แสดงตัวอย่าง Ladder Diagram.....	27
2.21 แสดงตัวอย่าง Ladder Diagram.....	28
2.22 แสดงตัวอย่าง Ladder Diagram.....	28
2.23 แสดงตัวอย่าง Ladder Diagram.....	29
2.24 แสดงโครงสร้างของโปรแกรมคำสั่งบูต.....	29
3.1 แสดงความสัมพันธ์ของส่วนประกอบหลักในการสื่อสารข้อมูล.....	31
3.2 แสดงการส่งแบบทิศทางเดียว (Simplex).....	32
3.3 แสดงการส่งแบบฮาร์ตฟลูเพิล็กซ์ (Half Duplex).....	32
3.4 แสดงการส่งแบบสองทิศทาง (Full Duplex).....	32
3.5 แสดงการสื่อสารข้อมูลแบบขนาน.....	33

สารบัญรูป (ต่อ)

รูปที่	หน้า
3.6 แสดงการสื่อสารข้อมูลแบบอนุกรม.....	33
3.7 แสดงโครงสร้างของแพ็คเกจ.....	35
3.8 แสดงรูปแบบการตรวจสอบด้วย Parity Bit.....	36
3.9 แสดงโครงสร้างของการสื่อสารข้อมูลแบบอนุกรมตามมาตรฐาน RS-232C.....	38
3.10 แสดงโครงสร้างของการสื่อสารข้อมูลแบบอนุกรมตามมาตรฐาน RS-422A.....	39
3.11 แสดงโครงสร้างของการสื่อสารข้อมูลแบบอนุกรมตามมาตรฐาน RS-485.....	41
3.12 แสดงระบบการติดต่อแบบจุดต่อจุด.....	42
3.13 แสดงการเชื่อมโยงแบบอินพุทและเอาต์พุทระยะไกล (Remote I/O System).....	42
3.14 แสดงการจำลองพื้นที่ของตัวควบคุมหลักขณะทำงานปกติ.....	43
3.15 แสดงการจำลองพื้นที่ของตัวควบคุมหลัก ขณะเมื่อมีการทำงานร่วมกับหน่วยอินพุทเอาต์พุทระยะไกล.....	43
3.16 แสดงการต่อแบบ Serial link.....	44
3.17 แสดงการต่อแบบ Parallel link.....	44
3.18 แสดงการแบ่งพื้นที่สำหรับ PC Link System.....	45
3.19 แสดงระบบการติดต่อสื่อสารแบบเครือข่ายท้องถิ่น.....	46
3.20 แสดงระบบโครงข่ายแบบหลายจุดโดยใช้มาตรฐานการสื่อสาร RS 485.....	46
3.21 แสดงการติดต่อระหว่างคอมพิวเตอร์กับเครื่องควบคุม PLC/PC.....	47
3.22 แสดงรูปแบบของบัส.....	47
3.23 แสดงรูปแบบของบัสและการคำนวณ FCS.....	49
4.1 บอร์ดหน่วยประมวลผลกลาง หน่วยความจำ และส่วนของการรับส่งข้อมูล.....	52
4.2 บอร์ดหน่วยอินพุท/เอาต์พุทระยะไกล.....	53
4.3 แสดงวงจรสำหรับการเชื่อมต่อหน่วยประมวลผลและหน่วยความจำ.....	54
4.4 แสดงวงจรสำหรับเชื่อมต่อจากคอมพิวเตอร์ไปยังบอร์ดประมวลผล.....	55
4.5 แสดงวงจรการเชื่อมต่อและการแปลงสัญญาณในการสื่อสารจาก RS-232 เป็น RS-422.....	56
4.6 แสดงวงจรของหน่วยอินพุท/เอาต์พุท.....	57
4.7 แสดงผังภาพการทำงานในวงรอบ.....	60
5.1 ผังการดำเนินงานส่วน Hardware.....	63
5.2 แสดง Flow Chart การดำเนินงานส่วน Software.....	64

สารบัญรูป (ต่อ)

รูปที่	หน้า
5.3 แสดง Block Diagram ส่วนของ Software.....	65
5.4 ส่วนของการสร้าง Text File.....	66
5.5 Flow Chart ของการเปลี่ยน Text File เป็น Toggle Code.....	68
5.6 Flow Chart ของโปรแกรมการส่ง Toggle Code ไปยังชุด Controller.....	70
5.7Flow Chart แสดงส่วนของโปรแกรมชุด Controller.....	72
6.1 แสดงผังของการป้อนโปรแกรมบูต.....	75
6.2 แสดงผังของการถ่ายเทข้อมูลจากเครื่องคอมพิวเตอร์ไปยังเครื่องควบคุม.....	76
7.1 แสดงอุปกรณ์ของชุดสายพานลำเลียง.....	78
7.2 แสดงโปรแกรมเลดเดอร์ไดอะแกรมของชุดสายพานลำเลียง.....	79
7.3 แสดงหน้าที่ของปุ่มคำสั่งการทำงานหลักของการ โปรแกรม.....	81
7.4 การป้อนโปรแกรม(1).....	82
7.5 การป้อนโปรแกรม(2).....	84



บทที่ 1

บทนำ

เครื่องควบคุมแบบตรรกที่โปรแกรมได้ เป็นอุปกรณ์ที่ใช้ในการควบคุมเครื่องจักรอัตโนมัติ ในโรงงานอุตสาหกรรม ซึ่งปัจจุบันมีการนำมาใช้งานกันอย่างแพร่หลาย เป็นอุปกรณ์ที่พัฒนาและออกแบบโดยไม่โครคอนโทรลเลอร์ มาแทนระบบการควบคุมแบบเก่าซึ่งใช้อุปกรณ์ไฟฟ้าเชิงกล เช่น รีเลย์ (Relay) ตั้งเวลา (Timer) ตัวนับ (Counter) เป็นต้น เนื่องจากเทคโนโลยีทางด้านไมโครคอนโทรลเลอร์ มีการพัฒนาเจริญก้าวหน้ามาก ดังนั้นการนำเทคโนโลยีดังกล่าวนี้มาพัฒนาเพื่อควบคุมเครื่องจักรในโรงงานอุตสาหกรรมจึงมีความน่าเชื่อถือสูง และมีข้อดีกว่าระบบเก่าหลายประการ

ปริญญาานิพนธ์นี้ นำเสนอการออกแบบเครื่องควบคุมแบบตรรกที่โปรแกรมได้ โดยใช้ไมโครคอนโทรลเลอร์ เบอร์ AT89C52 เป็นตัวประมวลผลกลาง ประกอบด้วย อินพุต 12 จุด เอาท์พุต 12 จุด มีวงจรถังเวลา 2 ชุด และวงจรถังนับอีก 2 นอกจากนี้ โดยจะแบ่งออกเป็น ส่วนของบอร์ดแม่ คือเป็นบอร์ดที่ประกอบด้วย หน่วยประมวลผล หน่วยความจำ โปรแกรมต่าง ๆ โดยการป้อนโปรแกรมจะถูกป้อนผ่านทางพอร์ตอนุกรมของคอมพิวเตอร์ โดยที่จะมีโปรแกรมสนับสนุนคือ โปรแกรม Visual Basic ตามมาตรฐาน RS-232 เพื่อเป็นส่วนควบคุมการทำงาน และส่งข้อมูลไปยัง บอร์ดลูก ซึ่งเป็นส่วนของการรับอินพุตและการส่งเอาท์พุต โดยที่แต่ละบอร์ด จะมี 4 อินพุต 4 เอาท์พุต โดยใช้มาตรฐาน RS-422 เป็นส่วนของการเชื่อมโยง

ระบบโครงข่ายนี้ออกแบบการเชื่อมต่อสื่อสารตามมาตรฐาน RS-422 ระบบถูกออกแบบให้เชื่อมโยงกันได้ 3 ชุดและระยะทางในการติดต่อ ควบคุมได้ นับร้อยเมตร จึงเป็นผลดี ต่อการควบคุมหรือแก้ไขข้อผิดพลาดของเครื่องควบคุมเครื่องจักรที่ถูก จัดวางในสถานที่ ที่ต่างกัน ซึ่งเป็นการเพิ่มประสิทธิภาพในการทำงานให้รวดเร็วและครอบคลุมทั้งระบบ

สำหรับปริญญาานิพนธ์ในบทที่ 2 จะได้กล่าวในส่วนของทฤษฎีและหลักการพื้นฐานของเครื่องควบคุมแบบตรรกที่โปรแกรมได้ บทที่ 3 เป็นการติดต่อสื่อสารของเครื่องควบคุมกับอุปกรณ์ภายนอก บทที่ 4 เป็นการออกแบบและพัฒนาตัวเครื่องควบคุมแบบตรรกที่โปรแกรมได้ บทที่ 5 เป็นการออกแบบระบบ โครงข่ายสำหรับเครื่องควบคุม บทที่ 6 เป็นการออกแบบโปรแกรมสนับสนุนการทำงานของเครื่องควบคุม บทที่ 7 เป็นการประยุกต์การใช้งานและผลการทดสอบเครื่องควบคุม และระบบโครงข่ายร่วมกับโปรแกรมสนับสนุนการทำงานส่วน บทที่ 8 จะเป็นการสรุปและวิจารณ์

บทที่ 2

ทฤษฎีและหลักการพื้นฐาน ของเครื่องควบคุมแบบตรรกะที่โปรแกรมได้

2.1 ที่มาและวิวัฒนาการของเครื่องควบคุมแบบตรรกะที่โปรแกรมได้

การควบคุมแบบลำดับขั้น (Sequence Control) เป็นการควบคุมแบบหนึ่งที่ต้องการให้เครื่องจักรหรือระบบการทำงานตามช่วงเวลา ตามลำดับขั้นตอน ตลอดจนตามเงื่อนไขที่ได้กำหนด โดยมีลักษณะของการควบคุม เป็นแบบ “ON” หรือ “OFF” การควบคุมแบบลำดับขั้น หรือแบบซีแควนซ์นี้จะพบเห็นอยู่เสมอในงานอุตสาหกรรมแทบทุกชนิด ตัวอย่างของการควบคุมแบบนี้ ได้แก่ ระบบควบคุมเครื่องจักร ระบบลำโพงอัตโนมัติ ระบบป้องกันวัสดุ ระบบการการผสมวัสดุ ระบบควบคุมลิฟท์ เป็นต้น

ในอดีตที่ผ่านมา ระบบควบคุมแบบซีแควนซ์จะใช้อุปกรณ์ไฟฟ้าเชิงกล (Electro-Mechanical Device) เช่น รีเลย์แม่เหล็กไฟฟ้า (Relay) ตัวตั้งเวลา (Timer) ตัวนับ (Counter) มาประกอบกันเป็นวงจรควบคุม เพื่อให้เครื่องจักร หรือระบบกระบวนการทำงานตามช่วงเวลา ตามลำดับขั้นตอน และตามเงื่อนไขที่วิศวกรหรือผู้ออกแบบระบบกำหนดได้ อย่างไรก็ตามระบบควบคุมแบบซีแควนซ์ที่ใช้อุปกรณ์ไฟฟ้าเชิงกลที่กล่าวมานี้ มีข้อเสียมาก คือขนาดของวงจรควบคุมจะมีขนาดใหญ่ สิ้นเปลืองทั้งเนื้อที่และพลังงานสูง ราคาแพง ไม่สามารถจะใช้กับระบบควบคุมที่มีขั้นตอนการทำงานยุ่งยาก ซับซ้อน เมื่อมีปัญหาเกิดขึ้นในวงจรควบคุม ก็ทำการตรวจสอบแก้ไขยาก การขยายระบบทำได้ยาก และที่สำคัญคือ ถ้าต้องการเปลี่ยนแปลงลำดับขั้นตอน หรือเงื่อนไขของการทำงานวงจรควบคุมแบบรีเลย์ที่มีอยู่เดิม จะเปลี่ยนแปลงเพื่อใช้กับงานใหม่ได้ยากต่อมาเมื่ออุปกรณ์สารกึ่งตัวนำ (Solid State Device) ได้แพร่หลาย และก้าวหน้าขึ้นจึงได้มีการนำเอาเกต (Gate) มาประกอบกันเป็นวงจรควบคุมแบบรีเลย์ และสามารถลดข้อเสียของระบบเดิมลงไปได้มาก เช่น วงจรควบคุมแทนรีเลย์ และสามารถลดข้อเสียการขยายระบบทำได้ง่ายขึ้น และยังสามารถต่อเข้ากับคอมพิวเตอร์ (Computer) เพื่อการเก็บข้อมูลและอื่นๆ ได้เป็นต้น แต่ก็ยังไม่สามารถจะลดข้อเสียบางอย่างที่สำคัญลงได้ นั่นคือการตรวจสอบแก้ไข เมื่อมีปัญหาที่ยังทำได้ยาก และไม่สามารถจะเปลี่ยนแปลงลำดับขั้นตอน หรือเงื่อนไขการทำงาน จึงได้มีการสร้างเครื่องควบคุมที่สามารถกำหนดโปรแกรมการทำงาน (Programmable Controller) ขึ้นมาเพื่อนำมาประยุกต์ใช้กับระบบควบคุมแต่อย่างใด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.1 วิวัฒนาการของเครื่องควบคุมแบบตรรกะที่โปรแกรมได้

PLC หรือ Programmable Logic Controllers และ PC หรือ Programmable Controller เป็นเครื่องควบคุมที่สามารถกำหนดโปรแกรมการทำงานได้ PLC/PC ที่ใช้ไมโครเซสเซอร์ นอกจากจะใช้ในการควบคุมเครื่องจักรโดยทั่วไปแล้ว ยังมีการพัฒนาให้มีความสามารถและขอบเขตให้งานได้กว้างขวางขึ้น เช่น มีฟังก์ชันทางคณิตศาสตร์เพิ่มขึ้นทำให้การควบคุมเป็นได้ทั้งแบบ ON-OFF หรือแบบอนาล็อก (Analog) เช่น PID (Proportional+Integral+Derivative) สามารถต่อเข้ากับอุปกรณ์วัดและควบคุม (Instrumentation) อื่นๆ จำนวนของอินพุต / เอาท์พุท (Input / Output) ขยายให้มากขึ้นได้ หรือการใช้อินพุต/เอาท์พุทแบบรีโมท (Remote) เพื่อลดการเดินสาย การติดต่อสื่อสารข้อมูลกับคอมพิวเตอร์ ตลอดจนอุปกรณ์พิเศษอื่นๆ เช่น เครื่องอ่านรหัสแถบเพื่อจำแนกหรือจัดทำฐานข้อมูล การเชื่อมโยง ข้อมูลลักษณะโครงข่ายท้องถิ่น (LAN) เป็นต้น หน่วยความจำก็ขยายได้มาก ทำให้ใช้กับระบบกระบวนการใหญ่ ๆ ได้ และมีการจัดการเกี่ยวกับข้อมูลได้ตามต้องการ

2.1.2 สรุปข้อดีของ PLC / PC

ข้อดีของ PLC / PC สามารถจำแนกออกได้เป็นข้อๆดังนี้

1. สิ้นเปลืองเนื้อที่น้อยเพราะมีขนาดเล็ก
2. สามารถจะใช้ควบคุมเครื่องจักรหรือระบบกระบวนการใด ๆ ก็ได้ ถ้าเลือกขนาดของ PLC / PC ที่เหมาะสม
3. การเปลี่ยนลำดับขั้นตอนหรือเงื่อนไขของการทำงานก็ทำได้ 9 k ต้องการเพราะใช้หลักของการโปรแกรม
4. ตัวตั้งเวลาและตัวนับเป็นซอฟต์แวร์ ทำให้การกำหนดค่าต่างๆ ง่ายต่อการเปลี่ยนแปลงค่าได้ตลอดเวลาไม่ต้องมีฮาร์ดแวร์ร่วม และทำให้ราคาถูกลง
5. รีเลย์ภายใน (Internal relay) ก็เป็นซอฟต์แวร์เช่นเดียวกัน จึงลดค่าใช้จ่ายในการเดินสาย ลวดฮาร์ดแวร์ และทำให้ขนาดเล็กลงด้วย
6. การติดตั้ง ใ้ได้ง่ายและสะดวก
7. ขยายระบบให้ใหญ่ขึ้นทำได้โดยง่าย
8. ราคาถูกกว่าระบบรีเลย์
9. ความน่าเชื่อถือ (Reliability) ดีเพราะเป็นอุปกรณ์สารกึ่งตัวนำ ไม่มีการเดินสายมาก ไม่มีปัญหาเกี่ยวกับหน้าสัมผัส (Contact) แบบรีเลย์
10. มีระบบตรวจสอบหาที่ผิดพลาดด้วยตัวเองการตรวจสอบแก้ไขเมื่อมีปัญหาจึงทำได้เร็ว
11. ลดการเดินสายยาวๆ และลดค่าใช้จ่ายในการเดินสายได้ เพราะมีอินพุต / เอาท์พุทแบบรีโมท
12. การบำรุงรักษาทำได้ง่าย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

13. เวลาในการทำงานเร็วกว่าระบบที่ใช้รีเลย์
14. มีฟังก์ชันทางคณิตศาสตร์ได้แก่ บวก ลบ คูณ หาร และอื่นๆ ทำให้สามารถใช้สำหรับการควบคุมแบบ ON-OFF หรือแบบอนาล็อก เช่น PID ได้
15. สามารถเชื่อมต่อ กับอุปกรณ์วัด เช่น เทอร์โมคัปเปิล (Thermocouple) และอื่นๆ ได้นอกเหนือจากอุปกรณ์ตรวจวัดที่เป็นสวิทช์ (Switch)
16. ต่อเข้ากับคอมพิวเตอร์ เพื่อวัตถุประสงค์ใดๆ เช่น การเก็บข้อมูลการกระจายการควบคุม (Distributed Control) เป็นต้น
17. การโปรแกรมทำได้หลายแบบ เช่น คำสั่งในรูปของแลดเดอร์ไดอะแกรม คำสั่ง บูลีน (Boolean Instruction) คำสั่งในรูปบล็อก (Block Instruction) หรือคำสั่งภาษาเบสิก
18. ใช้ได้ในทุกสภาพแวดล้อมของงานอุตสาหกรรม
19. การโปรแกรมทำได้โดยใช้เครื่องป้อนโปรแกรม (Program Loader) หรือโปรแกรมลงบน CRT

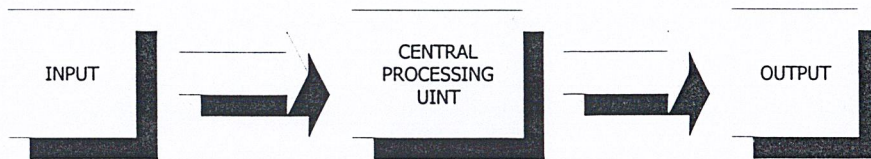
จากที่กล่าวมานี้จะเห็นว่า PLC/PC เป็นเครื่องควบคุมที่มีประสิทธิภาพ และมีประโยชน์อย่างยิ่งต่องานอุตสาหกรรมในปัจจุบันและในอนาคต

2.2 โครงสร้างและหลักการการทำงานของเครื่องควบคุมแบบตรรกะที่โปรแกรมได้

เครื่องควบคุมแบบตรรกะที่โปรแกรมได้ หรือ PLC / PC เป็นอุปกรณ์สารกึ่งตัวนำ ที่ใช้สำหรับควบคุมเครื่องจักร หรือระบบกระบวนการให้ทำงานตามโปรแกรมคำสั่งของผู้ใช้ (User Program) และข้อมูลต่างๆ ที่ได้รับจากอินพุต/เอาต์พุตของ PLC/PC จะเป็นได้ทั้งการทำงานตามช่วงเวลา ตามลำดับขั้นตอนฟังก์ชันทางคณิตศาสตร์ และอื่นๆ PLC/PC มีส่วนประกอบสำคัญ 2 ส่วน คือ

1. หน่วยประมวลผลกลางหรือ CPU (Central Processing Unit)
2. หน่วยอินพุต / เอาต์พุต (Input / Output Unit)

นอกจากส่วนประกอบทั้งสองแล้ว PLC / PC ยังประกอบด้วยหน่วยป้อนโปรแกรม (Programming Unit) ผู้ใช้สามารถติดต่อกับ PLC/PC ยังประกอบด้วย หน่วยป้อนโปรแกรม หรือเปลี่ยนแปลงแก้ไขโปรแกรมคำสั่งตรวจสอบสภาพการทำงานของเครื่องจักร ระบบกระบวนการตลอดจนตรวจสอบสภาพการทำงานของ PLC/PC ได้ทางหน่วยป้อนโปรแกรม รูปที่ 2-1 แสดงส่วนประกอบที่สำคัญทั้ง 2 ส่วนของ PLC/PC



รูปที่ 2.1 บล็อกไดอะแกรมของ PLC/PC

2.2.1 หน่วยประมวลผลกลาง

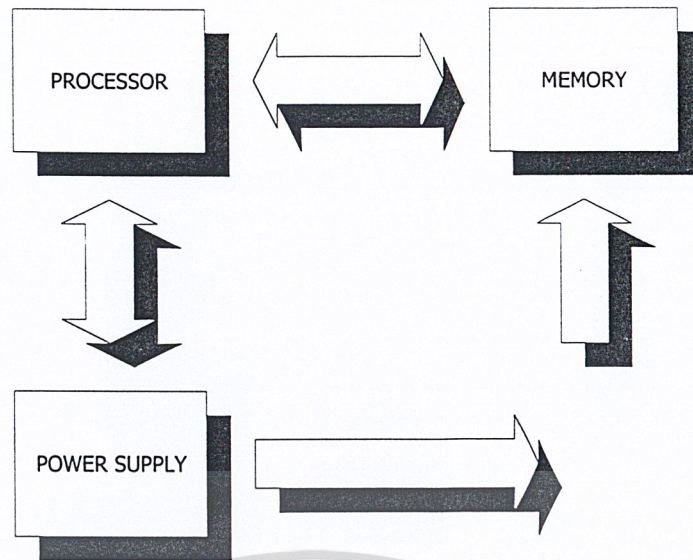
หน่วยประมวลผลหรือ CPU ประกอบด้วย 3 ส่วนคือ

1. หน่วยประมวลผล (Processor)
2. หน่วยความจำ (Memory)
3. หน่วยจ่ายกำลัง (Power Supply)

รูปที่ 2-2 แสดงส่วนประกอบของ CPU โดยทั่วไป CPU จะใช้ไมโครโปรเซสเซอร์ซึ่งเป็นไอซี (I.C.:Integrated Circuit) ที่มีความสามารถทั้งในการคำนวณทางคณิตศาสตร์ การจัดการข้อมูล และการตรวจสอบตัวเอง ในขณะที่วงจรควบคุมที่ใช้รีเลย์หรือไอซีพวกเกทต่าง ๆ ไม่สามารถจะทำได้ PLC/PC จึงมีข้อดีและประโยชน์มากกว่าระบบแบบเก่าอย่างเห็นได้ชัด

1. หน่วยประมวลผล

หน่วยประมวลผลของ CPU ทำหน้าที่ควบคุมและดูแลการทำงานของระบบทั้งหมด โดยรับข้อมูลจากหน่วยอินพุตมาทำการประมวลผลตามโปรแกรมคำสั่งของผู้ใช้ที่เก็บไว้ในหน่วยความจำและส่งผลที่ได้ออกไปยังหน่วยเอาต์พุต ในปัจจุบันหน่วยประมวลผลของ CPU จะใช้ไมโครโปรเซสเซอร์ขนาดตั้งแต่ 4 บิต 8 บิต หรือ 16 บิต จำนวน 1 ตัว หรือหลาย ๆ ตัวมาทำงานร่วมกัน ในกรณีที่ใช้ไมโครโปรเซสเซอร์หลายๆ ตัวจะมีข้อดีคือ การทำงานของระบบจะเร็วขึ้น ทำให้ PLC/PC มีขีดความสามารถสูงขึ้น เช่น การใช้ชุดเชื่อมต่อพิเศษที่มีไมโครโปรเซสเซอร์ และหน่วยความจำภายในจะทำให้การทำงานควบคุมดียิ่งขึ้นและเป็นอิสระไม่ขึ้นกับ CPU ตัวอย่างก็คือชุดควบคุม PID ที่ใช้สำหรับควบคุมระบบแบบลูปปิด (Closed-Loop Control) ซึ่งทำงานอิสระจาก CPU



รูปที่ 2.2 บล็อกไดอะแกรมของ CPU

การทำงานของ CPU 1 รอบ เริ่มจากการรับข้อมูลจากหน่วยอินพุตเข้ามาประมวลผลตามโปรแกรมคำสั่งของผู้ใช้ ส่งผลที่ได้ไปยังหน่วยเอาต์พุต ตรวจสอบการทำงานของระบบทั้งหมด และติดต่อกับผู้ใช้และส่งผลที่ได้ไปยังหน่วยเอาต์พุต เรียกว่าการ SCAN โดยทั่วไปเวลา SCAN ของ CPU จะใช้เวลาประมาณ 0.001 ถึง 0.1 วินาที เวลา SCAN จะอ้างอิงถึงขนาดของหน่วยความจำที่เก็บโปรแกรมของผู้ใช้เสมอ เช่น เวลา SCAN เท่ากับ 0.01 วินาทีต่อโปรแกรมคำสั่งขนาด 1 กิโลไบต์ (K-byte) อย่างไรก็ตามเวลา SCAN จะขึ้นอยู่กับองค์ประกอบอื่นๆด้วย เช่น การใช้ชุดอินพุต/เอาต์พุต แบบรีโมทจะทำให้เวลา SCAN ของ CPU เพิ่มขึ้น



รูปที่ 2.3 การ SCAN ของ CPU

ช่วงเวลา SCAN ของ CPU จะทำให้เราได้ทราบถึงความสามารถในการตอบสนองต่อการเปลี่ยนแปลงของหน่วยอินพุต/เอาต์พุตว่ารวดเร็วเพียงไร เวลา SCAN จึงเป็นองค์ประกอบที่สำคัญอย่างหนึ่งในการเลือกใช้ PLC/PC ให้เหมาะสมกับงาน เช่น ถ้าต้องการที่จะตรวจสอบสัญญาณอินพุตที่เปลี่ยนค่าสภาวะ 2 ครั้งภายใน 0.008 วินาที การใช้ PLC/PC ที่มีช่วงเวลา SCAN

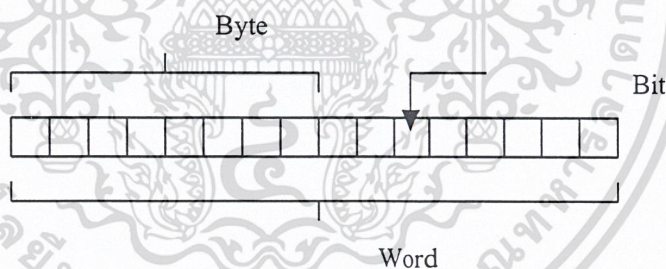
เท่ากับ 0.01 วินาที จะไม่สามารถตรวจสอบค่าสถานะที่ถูกต้อง และอาจทำให้เครื่องจักรหรือกระบวนการที่ควบคุมทำงานผิดพลาดได้

นอกจากนี้ หน่วยประมวลผลใน CPU จะทำหน้าที่ในการติดต่อกับระบบย่อย (Subsystem) ตรวจสอบความผิดพลาดต่างๆ เช่น การทำงานของหน่วยประมวลผล หน่วยความจำ หรือสภาพของแบตเตอรี่จ่ายกำลังสำรอง เป็นต้น

2. หน่วยความจำ

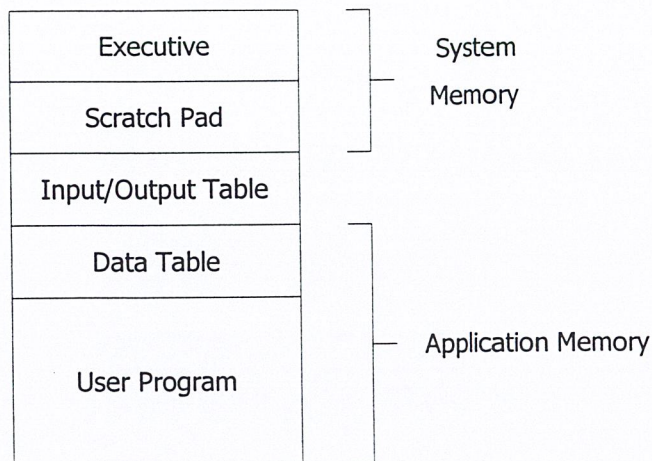
หน่วยความจำเป็นที่เก็บโปรแกรม และข้อมูลที่หน่วยประมวลผล ใช้ในการควบคุมการทำงานของ PLC/PC และการทำงานตามโปรแกรมคำสั่งของผู้ใช้ ขนาดของหน่วยความจำจะแบ่งออกเป็นบิตข้อมูล ภายในหน่วยความจำ 1 บิต จะมีสถานะทางลอจิก “1” หรือ “0” ข้อมูลขนาด 8 บิต และ 16 บิต รวมกันเรียกว่า ไบท์(Byte) และเวิร์ด(Word) ตามลำดับ

หน่วยความจำแบ่งออกได้เป็น 2 ประเภทใหญ่ๆ คือหน่วยความจำที่ข้อมูลสูญหายเมื่อไม่มีแหล่งจ่ายกำลัง (Volatile) และหน่วยความจำที่ข้อมูลคงอยู่แม้ไม่มีแหล่งจ่ายกำลัง (Non-Volatile) หรืออาจจะแบ่งเป็น 2 ชนิดใหญ่ ๆ ตามคุณลักษณะคือ RAM (Random Access Memory) หรือ NOVRAM (Non-Volatile RAM) และ ROM(Read Only Memory) EAROM(Electrically Alterable ROM) และ EEPROM(Electrically Erasable Programable ROM)



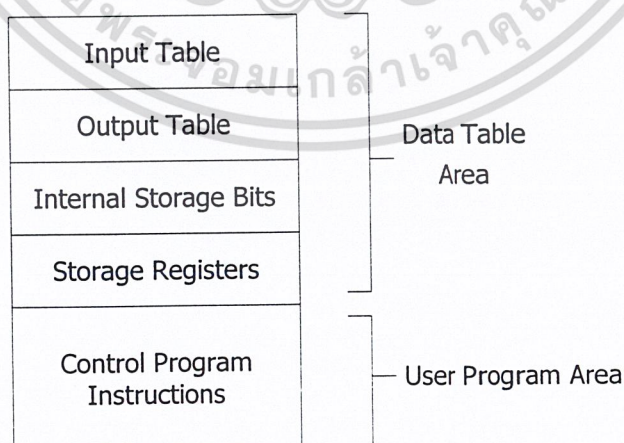
รูปที่ 2.4 ข้อมูลเป็น Bit Byte และ Word

หน่วยความจำของ PLC/PC แบ่งออกเป็น 4 ส่วน ตามรูป 2.5 เพื่อใช้เก็บข้อมูลและโปรแกรมต่างๆ คือ



รูปที่ 2.5 หน่วยความจำของ PLC/PC

1. โปรแกรมจัดการ (Execute Program) เป็นโปรแกรมที่ทำหน้าที่ในการควบคุม ดูแล และตรวจสอบการทำงานของ PLC/PC ทั้งหมด
2. ข้อมูลชั่วคราว (Process Work Area หรือ Scratch Pad) หน่วยความจำส่วนที่สอง ทำหน้าที่เก็บข้อมูลที่เกิดขึ้น ระหว่าง PLC/PC ทำงานตามโปรแกรมจัดการและโปรแกรมคำสั่งของผู้ใช้
3. ตารางข้อมูล (Data Table) หน่วยความจำส่วนที่สาม ทำหน้าที่เก็บค่าของอินพุท/เอาต์พุท และตัวแปรต่าง ๆ จากการทำงานตามคำสั่งโปรแกรมของผู้ใช้ ในรูปแบบที่เป็นค่าสภาวะทางลอจิก และตัวเลข
4. หน่วยความจำสำหรับเก็บ โปรแกรมคำสั่งของผู้ใช้ (User Program Memory)



รูปที่ 2.6 หน่วยความจำสำหรับผู้ใช้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หน่วยความจำของผู้ใช้จะแตกต่างกันตามขนาดของ PLC/PC ใน PLC หรือ PC ขนาดใหญ่ ขนาดของหน่วยความจำดังกล่าว จะสามารถเปลี่ยนแปลงขนาดได้ตามต้องการ การเลือกใช้ PLC/PC จึงต้องคำนึงถึงขีดความสามารถและขีดจำกัดต่างๆประกอบด้วย เช่น ขนาดของโปรแกรมคำสั่งของผู้ใช้ จำนวนอินพุต/เอาต์พุต ที่จะขยายได้สูงสุด ขนาดของหน่วยความจำภายในที่ใช้เก็บข้อมูล และฟังก์ชันพิเศษต่างๆ เช่น รีเลย์ภายใน ตัวตั้งเวลา และตัวนับ

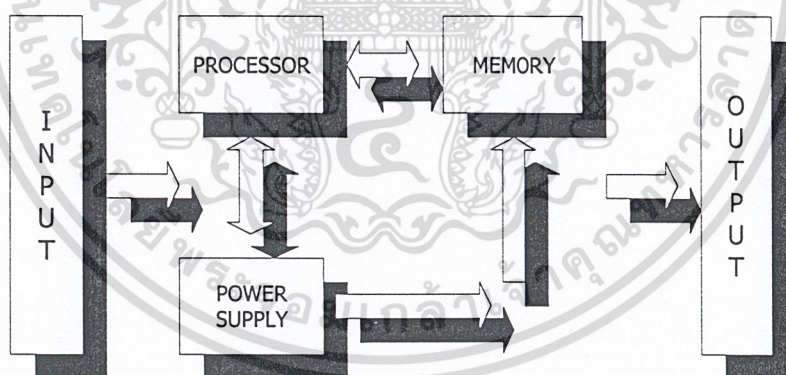
3. หน่วยจ่ายกำลัง

หน่วยจ่ายกำลังจะทำหน้าที่จ่าย และรักษาระดับแรงดันไฟฟ้ากระแสตรง (DC Voltage) ให้กับหน่วยประมวลผล หน่วยความจำ และหน่วยอินพุต/เอาต์พุต ตามความต้องการ และทำหน้าที่เตือนให้หน่วยประมวลผลทราบเมื่อเกิดปัญหา

2.2.2 หน่วยอินพุต/เอาต์พุต

หน่วยอินพุตทำหน้าที่เชื่อมต่อระหว่าง CPU กับอุปกรณ์ภายนอกโดยรับค่าสถานะ หรือ ปริมาณทางกายภาพต่างๆจากอุปกรณ์ตรวจวัด (Sensor) ของเครื่องจักรหรือกระบวนการ เช่น ลิมิตสวิตช์ (Limit Switch) พร็อกซิมีตีส์วิตช์ (Proximity Switch) ตำแหน่ง ระดับ อุณหภูมิ แรงดัน กระแสไฟ และอื่นๆ ส่งไปยัง CPU เพื่อประมวลผลตามโปรแกรมคำสั่งผู้ใช้

หน่วยเอาต์พุตทำหน้าที่รับค่าสถานะหรือคำสั่งที่ได้จาก CPU เพื่อส่งไปควบคุมอุปกรณ์ ภายในเครื่องจักรหรือกระบวนการเช่น วาล์ว (Valve) มอเตอร์ (Motor) ปั๊ม (Pump) และอื่น ๆ ให้ทำงานตามค่าสถานะหรือคำสั่งที่ CPU ต้องการ



รูปที่ 2.7 บล็อกไดอะแกรมของหน่วยอินพุตเอาต์พุต

ดังที่ได้กล่าวมาแล้วว่าในช่วงแรก PLC/PC ถูกใช้งานแทนรีเลย์ สำหรับใช้ในการควบคุมแบบ ON-OFF หน่วยอินพุต/เอาต์พุต ในขณะนั้น จึงเป็นแบบที่ใช้สำหรับค่าสถานะลอจิกเท่านั้น

แต่ต่อมาเมื่อ PLC/PC ได้ถูกพัฒนาขึ้นเพื่อใช้ในการรับส่งสัญญาณแบบอนาล็อกโดยเฉพาะ ปัจจุบันหน่วยอินพุต และ เอาท์พุทของ PLC/PC อาจแบ่งได้ดังนี้คือ

1. อินพุต/เอาท์พุทแบบค่าสถานะลอจิก
2. อินพุต/เอาท์พุทแบบค่าตัวเลข (Numerical Data Input/Output)
3. อินพุต/เอาท์พุทแบบหลายบิต
4. อินพุต/เอาท์พุทแบบพิเศษ (Special Input/Output)

1. อินพุต/เอาท์พุทแบบค่าสถานะลอจิก

อินพุต/เอาท์พุทแบบนี้จะใช้งานกันมากที่สุด ข้อมูลจะมีค่าสถานะอยู่ 2 แบบ คือ “1” หรือ “ON” หมายถึง สวิตช์หรือหน้าสัมผัสของรีเลย์แต่ละกัน และสถานะ “0” หรือ ”OFF” หมายถึง สวิตช์หรือหน้าสัมผัสของรีเลย์เปิดออก

เนื่องจากระดับแรงดันไฟฟ้าทางด้านอินพุตมีหลายแบบ การเลือกใช้งานหน่วยอินพุต/เอาท์พุท แบบค่าสถานะลอจิกจึงจะต้องเลือกใช้ให้ถูกต้อง ตารางที่ 2-1 แสดงให้เห็นตัวอย่างของอุปกรณ์อินพุต/เอาท์พุทแบบลอจิก

ตารางที่ 2.1 อุปกรณ์ อินพุต/เอาท์พุทแบบค่าสถานะลอจิก

	อุปกรณ์ทางด้านอินพุต	อุปกรณ์ทางด้านเอาท์พุท
1	Selector Switches	Alarms
2	Push Buttons	Control Relays
3	Photoelectric Eyes	Fans
4	Limit Switches	Lights
5	Circuit Breakers	Horns
6	Proximity Switches	Valves
7	Motor Starter Contacts	Motor Starters
8	Relay Contacts	Solenoid
9	Level Switches	

2. อินพุต/เอาท์พุทแบบค่าตัวเลข

อินพุต/เอาท์พุทแบบค่าตัวเลขนี้จะมีอยู่ใน PLC/PC ตั้งแต่ขนาดกลางที่มีฟังก์ชันก็คำนวณทางคณิตศาสตร์รวมอยู่ด้วย โดยทั่วไปแล้วอินพุต/เอาท์พุทแบบค่าตัวเลขนี้จะแบ่งเป็น 2 ชนิด คือ อินพุต/เอาท์พุทสำหรับสัญญาณอนาล็อก และอินพุต/เอาท์พุทสำหรับค่าสถานะแบบหลายบิต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หน่วยอินพุตสำหรับสัญญาณอนาล็อกจะรับสัญญาณอินพุตจากอุปกรณ์ต่างๆ และหน่วยเอาต์พุตจะส่งสัญญาณอนาล็อกไปยังอุปกรณ์ต่างๆ ตารางที่ 2.2 แสดงถึงอุปกรณ์อินพุต/เอาต์พุตแบบ

ตารางที่ 2.2 อุปกรณ์อินพุต/เอาต์พุตแบบค่าตัวเลข

	อุปกรณ์ทางด้านอินพุตแบบอนาล็อก	อุปกรณ์ทางด้านเอาต์พุตแบบอนาล็อก
1	Temperature Transducers	Analog Valves and Actuators
2	Pressure Transducers	Chart Recorders
3	Load Cell Transducers	Electric Motor Drives
4	Humidity Transducers	Analog Meters
5	Flow Transducers	

3. อินพุต/เอาต์พุตสำหรับค่าสถานะแบบหลายบิต

หน่วยอินพุต/เอาต์พุตสำหรับค่าสถานะแบบหลายบิตนี้จะทำงานเหมือนกับหน่วยอินพุต/เอาต์พุตแบบค่าสถานะลอจิกทุกอย่าง ต่างกันก็แต่เพียงการรับ-ส่งข้อมูล ซึ่งสามารถทำได้ครั้งละหลายๆบิตในเวลาเดียวกัน เช่น อินพุต/เอาต์พุตแบบBCD เป็นต้น ตารางที่ 2.3 แสดงถึงอุปกรณ์ที่ร่วมกับหน่วยอินพุต/เอาต์พุตสำหรับค่าสถานะแบบหลายบิต

ตารางที่ 2.3 อุปกรณ์อินพุต/เอาต์พุตสำหรับค่าสถานะแบบหลายบิต

	อุปกรณ์ทางด้านอินพุตแบบหลายบิต	อุปกรณ์ทางด้านเอาต์พุตแบบหลายบิต
1	Thumbwheel Switch	Seven-Segment Displays
2	Bar Codes Reader	Intelligent Display

4. อินพุต/เอาต์พุตแบบพิเศษ

สำหรับ PLC/PC ตั้งแต่ขนาดกลางขึ้นไปจะสามารถนำเอาหน่วยอินพุต/เอาต์พุตแบบพิเศษมาต่อเพื่อใช้งานเฉพาะด้านได้ อินพุต/เอาต์พุตสำหรับงานเฉพาะด้านนั้นมีหลายอย่างต่อไปนี้

1. อินพุตที่เป็นเทอร์โมคัปเปิล (Thermocouple Input) หน่วยอินพุต/ สำหรับเทอร์โมคัปเปิลนี้จะทำหน้าที่ปรับระดับสัญญาณ (Signal Conditioning) ที่รับมาโดยตรงจากเทอร์โมคัปเปิลให้มีระดับที่เหมาะสม การทำงานของหน่วย อินพุตชนิดนี้จะเหมือนกับหน่วยอินพุตสัญญาณอนาล็อกจากเทอร์โมคัปเปิลนี้จะถูกรอง ขยาย และทำให้เป็น

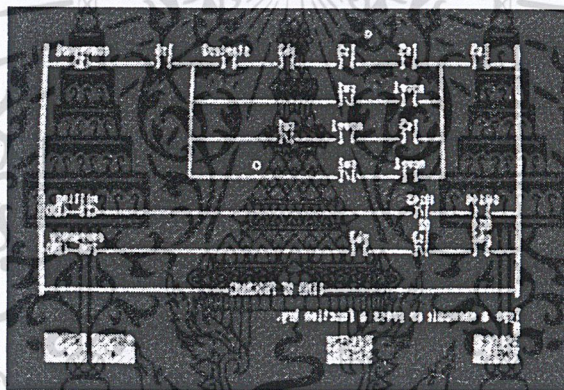
- สัญญาณดิจิทัล จากนั้นถูกส่งไปยังหน่วยประมวลผลเพื่อการทำงานตามโปรแกรมคำสั่งของผู้ใช้
2. อินพุตที่เป็นพัลส์ หน่วยอินพุตสำหรับรับอินพุตที่เป็นพัลส์นี้ จะทำหน้าที่ตรวจจับสัญญาณที่มีการเปลี่ยนแปลงเร็วกว่า 1 ช่วง เวลาของการ SCAN ของ CPU
 3. อินพุต/เอาต์พุตที่เป็นรหัส ASCII หน่วยอินพุต/เอาต์พุตแบบนี้ใช้สำหรับการรับ-ส่งข้อมูลที่เป็นตัวอักษรในรหัส ASCII
 4. อินพุตที่เป็นสเตรนเกจ (Strain Gage Input) หน่วยอินพุตสำหรับสัญญาณอนาล็อกจากสเตรนเกจเหมือนกับหน่วยอินพุตสำหรับเทอร์โมคัปเปิล แตกต่างกันที่วงจรปรับระดับสัญญาณ ซึ่งออกแบบไว้รับสัญญาณจากสเตรนเกจเท่านั้น
 5. เอาต์พุตที่เป็นพัลส์ (Pulse Output) หน่วยเอาต์พุตนี้ จะทำหน้าที่ส่งพัลส์ไปควบคุมการทำงานของสเตปปีงมอเตอร์ โดยเฉพาะ ตำแหน่งของมอเตอร์จะถูกกำหนดจากจำนวนของพัลส์เอาต์พุตที่ถูกนับไว้ก่อน และคำสั่งให้หมุนไปในทิศทางทวนเข็มนาฬิกา หรือตาม เข็มนาฬิกา
 6. หน่วยเชื่อมต่อเซอร์โว (Servo Interface) หน่วยเชื่อมต่อเซอร์โว เป็นหน่วยอินพุต/เอาต์พุตที่ถูกออกแบบขึ้นเพื่อตรวจสอบและควบคุมตำแหน่งของเครื่องจักร โดยเฉพาะ เช่นการควบคุมในระบบ NC (Numerical Control)
 7. หน่วย APM (Axis Positioning Module) หน่วย APM เป็นหน่วยเชื่อมต่อของ PC ที่ได้รับการออกแบบ ขึ้นเพื่อใช้ตรวจสอบ และควบคุมตำแหน่งของเครื่องจักร เช่นเดียวกับหน่วยเชื่อมต่อเซอร์โว แต่สามารถจะเข้าร่วมกับการควบคุมแบบซีเคานซ์ได้โดยที่ทำการแยกเป็นอิสระจาก CPU
 8. หน่วย PID เป็นหน่วยอินพุต/เอาต์พุต ที่ได้รับการออกแบบเพื่อใช้ควบคุมในระบบลูฟปิดที่ต้องการฟังก์ชัน การควบคุมแบบ PID (Proportional,Integal,Derivative)
 9. หน่วยเชื่อมต่อเครือข่าย (Network Interface Module) หน่วยเชื่อมต่อนี้ถูกออกแบบขึ้นเพื่อเพิ่มสมรรถนะการทำงานของ ให้สูงขึ้น โดยช่วยให้ PC หลายๆระบบเชื่อมต่อกันและติดต่อกันได้ หรือติดต่อกับระบบคอมพิวเตอร์เพื่อใช้เป็นหน่วยควบคุมหลัก
 10. หน่วยอินพุต/เอาต์พุตแบบรีโมท เราจะใช้กับPCขนาดใหญ่ ที่มีอินพุต/เอาต์พุตจำนวนมากๆ และมีระบบย่อย หลายระบบ เพื่อทำหน้าที่ตรวจสอบสถานะของอุปกรณ์จากจุดต่างๆ หลายจุดในเวลาเดียวกัน ภายใต้การควบคุมของ CPU

2.2.3. หน่วยป้อนโปรแกรม

หน่วยป้อนโปรแกรมของ PLC/PC ทำหน้าที่ติดต่อระหว่างผู้ใช้และ PLC/PC ทั้งระบบคือการป้อนคำสั่งของผู้ใช้เข้าสู่หน่วยความจำของ CPU ตรวจสอบสถานะการทำงาน รับ-ส่งข้อมูลระหว่าง CPU กับเทป หรืออื่นๆตามฟังก์ชันการใช้งานที่มีอยู่ หน่วยป้อนโปรแกรมของ PLC/PC แบ่งออกได้หลายชนิดคือ

1. เครื่องป้อนโปรแกรมแบบ CRT

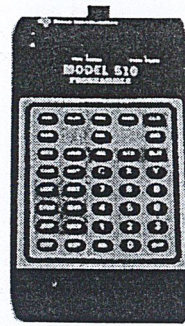
หน่วยป้อนโปรแกรม CRT ประกอบด้วยจอภาพและเป็นพิมพ์ เป็นเครื่องป้อนโปรแกรมแบบที่สะดวกที่สุด เครื่องป้อนโปรแกรมแบบนี้แบ่งออกเป็น 2 ชนิด คือ แบบดัมพ์ (Dumb CRT) ซึ่งเป็นแบบที่ไม่มีหน่วยประมวลผลอยู่ภายในตัว การทำงานทุกอย่างจะถูกควบคุมจาก CPU ของ PLC/PC และแบบอินเทลลิเจนต์ CRT (Intelligent CRT) ซึ่งจะมีหน่วยประมวลผลอยู่ภายในตัวการทำงานเป็นอิสระจาก CPU จึงมีประสิทธิภาพดีกว่าแบบดัมพ์



รูปที่ 2.8 เครื่องป้อน โปรแกรมแบบ CRT

2. เครื่องป้อน โปรแกรมขนาดเล็ก (Mini Programmer)

เป็นเครื่องป้อนโปรแกรมขนาดเล็กที่สามารถพกติดตัวและเคลื่อนย้ายไปมาได้สะดวก แต่ประสิทธิภาพจะด้อยกว่าเครื่องป้อนโปรแกรมแบบ CRT ปกติแล้วจะใช้จอ LCD ในการแสดงผล เครื่องป้อนโปรแกรมแบบนี้จะแบ่งออกเป็น 2 ชนิด คือ แบบอินเทลลิเจนต์และแบบดัมพ์



รูปที่ 2.9 เครื่องป้อนโปรแกรมขนาดเล็ก หรือพกพา

3. เครื่องป้อนโปรแกรมลงในหน่วยความจำของ PLC/PC หรือเทป

เครื่องป้อนโปรแกรมแบบนี้สามารถใช้ป้อนหรือแก้ไขโปรแกรมคำสั่งของผู้ใช้เข้าไปในหน่วยความจำของ PLC/PC ได้โดยตรงหรือใช้ในการรับส่งโปรแกรมและส่งข้อมูลต่างๆ ระหว่าง PLC/PC และเทป

4. เครื่องป้อนโปรแกรมลงในหน่วยความจำ (Memory Burner)

เครื่องป้อนโปรแกรมแบบนี้ทำหน้าที่ถ่ายโปรแกรมที่มีอยู่ลงเก็บไว้ในหน่วยความจำแบบ ROM เพื่อให้โปรแกรมต่างๆ คงอยู่ตลอดเวลา เมื่อโปรแกรมต่างๆ ได้ถูกตรวจสอบและแก้ไขจนเรียบร้อยสมบูรณ์แล้ว

5. คอมพิวเตอร์

เครื่องควบคุม PLC/PC บางรุ่นได้ออกแบบให้ สามารถติดต่อกับคอมพิวเตอร์ได้ เพื่อให้การโปรแกรมทุกอย่างสามารถทำบนคอมพิวเตอร์ได้

2.3 ลักษณะการโปรแกรมให้กับเครื่องควบคุมแบบตรรกะที่โปรแกรมได้

ภาษาที่ใช้เขียนโปรแกรมคำสั่งสำหรับ PLC/PC นั้นจำแนกออกได้เป็น 4 ภาษาคือ

1. ภาษาแลดเดอร์ไคอะแกรม
2. ภาษาคำสั่งบูลีน
3. ภาษาคำสั่งในรูปแบบบล็อก
4. ภาษาระดับสูง

ภาษาแลดเดอร์ไคอะแกรม หรือคำสั่งบูลีนเป็นภาษาที่นิยมใช้ และเข้าใจได้ง่าย สำหรับการเขียนโปรแกรมคำสั่งของ PLC/PC ในการควบคุมแบบซีควเอนซ์ที่เป็น ON-OFF การเขียนโปรแกรมคำสั่งในภาษาแลดเดอร์ไคอะแกรม หรือคำสั่งบูลีนจึงใช้กันมากใน PLC หรือ PC ทุกขนาด ส่วนภาษาคำสั่งในรูปแบบบล็อก หรือภาษาระดับสูง เช่น ภาษาซี ภาษาปาสคาล

ภาษาฟอร์แทรนนั้น จะเหมาะสำหรับ PLC/PC ในการควบคุมแบบอนาล็อกการจัดการเกี่ยวกับข้อมูลหรือหลายภาษารวมกันในการเขียนโปรแกรมคำสั่งก็ได้ เช่น ภาษาแลคเคอร์ไคอะแกรม ร่วมกับคำสั่งบูลีนภาษาเดียว หรือใช้ภาษาแลคเคอร์ไคอะแกรมหรือคำสั่งในรูปบล็อก ภาษาคำสั่งบูลีนร่วมกับคำสั่งในรูปบล็อก หรือภาษาคำสั่งบูลีนร่วมกับภาษาระดับสูง เป็นต้น อย่างไรก็ตาม ในหัวข้อนี้จะกล่าวถึงการเขียนโปรแกรมคำสั่งของ PLC/PC โดยใช้ภาษาแลคเคอร์ไคอะแกรม และคำสั่งบูลีนเท่านั้น

2.3.1. โปรแกรมภาษาแลคเคอร์ไคอะแกรม

ลักษณะพื้นฐานของภาษาแลคเคอร์ไคอะแกรม ดังที่ทราบกันแล้วว่า PLC/PC เป็นเครื่อง ควบคุมที่ใช้การโปรแกรม และใช้ซอฟต์แวร์แทนอุปกรณ์ต่างๆ เช่นรีเลย์ ตัวตั้งเวลา ตัวนับ ดังนั้น การเขียนโปรแกรมคำสั่งจึงใช้สัญลักษณ์พื้นฐานต่อไปนี้แทนอุปกรณ์ต่างๆ

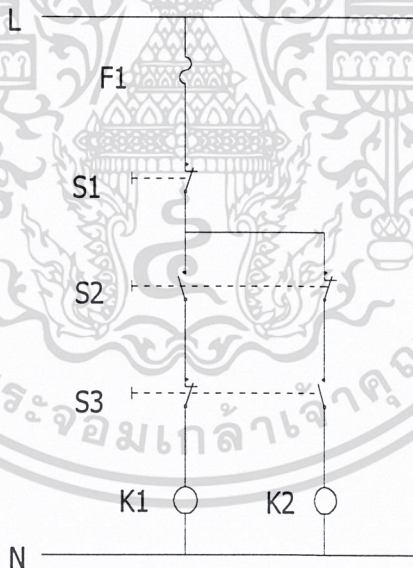
ตารางที่ 2.4 แสดงถึงตัวอย่างของการใช้ฟังก์ชันลอจิก และสัญลักษณ์พื้นฐานในการเขียนโปรแกรมแลคเคอร์ของ PLC/PC

สัญลักษณ์	ความหมาย
+ --] [- - +	จะใช้แทนอุปกรณ์ทางด้านอินพุตต่างๆ ได้แก่ สวิตช์ ไฟฟ้าหรือหน้าสัมผัส ของรีเลย์ ตัวตั้งเวลา ตัวนับ หรือรีเลย์ภายในเป็นต้น โดยที่อุปกรณ์นั้นปกติจะมีค่าสถานะเป็นลอจิก “0” หน้าสัมผัสปกติเปิดหรือ N.O.
+ --] / [- - +	จะใช้แทนอุปกรณ์ทางด้านอินพุตต่างๆ ได้แก่ สวิตช์ ไฟฟ้าหรือหน้าสัมผัส ของรีเลย์ ตัวตั้งเวลา ตัวนับ หรือรีเลย์ภายในเป็นต้น โดยที่อุปกรณ์นั้นปกติจะมีค่าสถานะเป็นลอจิก “1”
+ --() - - +	เอาท์พุทปกติไม่ทำงาน จะแทนอุปกรณ์ทางด้านเอาท์พุทต่างๆ ได้แก่ หลอด ไฟฟ้า มอเตอร์ไฟฟ้า ขดลวดของรีเลย์ ขดลวดโซลินอยด์(Solenoid) ตัวตั้งเวลา ตัวนับ รีเลย์ภายใน เป็นต้น โดยที่ปกติแล้วอุปกรณ์ทางด้านเอาท์พุทจะมีค่าสถานะลอจิกเป็น “0”

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

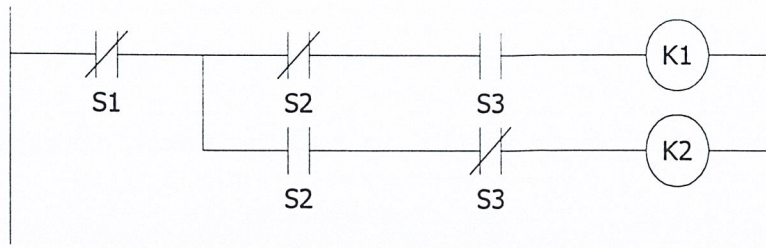
+ -- (/) -- +	เอาท์พุทปกติไม่ทำงาน จะแทนอุปกรณ์ทางด้านเอาท์พุทต่างๆ ได้แก่ หลอด ไฟฟ้า มอเตอร์ไฟฟ้า ขดลวดของรีเลย์ ขดลวดโซลินอยด์(Solenoid) ตัวตั้งเวลา ตัวนับ รีเลย์ภายใน เป็นต้น โดยที่ปกติแล้วอุปกรณ์ทางด้านเอาท์พุท จะมีค่า สภาวะลอจิกเป็น “1”
---------------	---

ภาษาแลดเดอร์ไดอะแกรม เป็นภาษาที่เข้าใจได้ง่าย เนื่องจากมีลักษณะคล้ายกับรีเลย์ ไดอะแกรมจึงทำได้ง่ายดังตัวอย่างแสดงให้เห็นความสัมพันธ์ระหว่างรีเลย์ไดอะแกรม ซึ่งเป็นวงจรควบคุมการกลับทางหมุนของมอเตอร์แบบ JOGGING ในรูปที่ 2.10 และเปลี่ยนมาใช้ PLC/PC ควบคุมโดยเขียนเป็นโปรแกรมแลดเดอร์ไดอะแกรมในรูปที่ 2.11 แลดเดอร์ไดอะแกรมจะแบ่งออกเป็นส่วนๆ แต่ละส่วนคือ เอาท์พุท 1 จุดหรือเอาท์พุทมากกว่า 1 จุด แต่เอาท์พุทเหล่านี้จะมีลอจิกเหมือนกัน แต่ละส่วนเหล่านี้เรียกว่า “ รังค์ (RUNG) ” ดังรูปที่ 2.12 เป็นโปรแกรมที่ให้ผลการควบคุมเหมือนเดิม

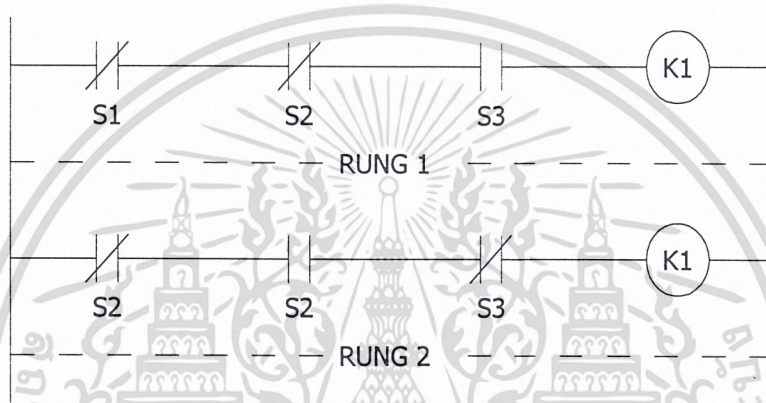


รูปที่ 2.10 วงจรควบคุมการหมุนกลับทางของมอเตอร์แบบ JOGGING

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.11 แลคเตอร์ไคอะแกรมควบคุมการหมุนกลับทางของมอเตอร์แบบ JOGGING



รูปที่ 2.12 แสดงการแบ่งแลคเตอร์ไคอะแกรมออกเป็นรังค์

การโปรแกรมด้วยแลคเตอร์ไคอะแกรมจะสะดวกและมีประสิทธิภาพมาก แต่ข้อเสียของโปรแกรมแบบนี้ คือ อุปกรณ์ที่ใช้ในการเขียนโปรแกรมจะต้องมีหน่วยแสดงผลที่สามารถแสดงอักขระที่ใช้ในโปรแกรมแบบนี้ มักจะเป็นจอ CRT แต่ปัจจุบันจะมีซอฟต์แวร์สนับสนุนการเขียนแลคเตอร์ไคอะแกรมบนคอมพิวเตอร์ และเชื่อมโยงข้อมูลกับ PLC/PC ทางพอร์ตอนุกรม โดยคอมพิวเตอร์ ทำหน้าที่ในการสร้าง และแก้ไขแลคเตอร์ไคอะแกรม แล้วทำการแปลข้อมูลเป็นชุดคำสั่งของเครื่องควบคุม นอกจากนั้นแล้ว ขณะที่เครื่องควบคุมทำงานอยู่ก็สามารถตรวจสอบการทำงานได้ โดยการแสดงผลซึ่งจะแสดงเป็นแลคเตอร์ไคอะแกรม และเมื่ออินพุทหรือเอาต์พุทตัวใดมีการเปลี่ยนแปลงสถานะ จะมีการเปลี่ยนแปลงตามสถานะของอินพุทหรือเอาต์พุทตัวนั้นด้วย

2.3.2. โปรแกรมคำสั่งภาษาแบบบูลีน

การโปรแกรมชนิดนี้ มีลักษณะคล้ายกับสัญลักษณ์ของพีชคณิตบูลีน นอกจากจะเป็นสัญลักษณ์ของพีชคณิตบูลีนแล้ว ก็ยังมีการใช้ฟังก์ชันพิเศษต่าง ๆ อีกมาก เช่น ตัวนับ ตัวตั้งเวลา เป็นต้น ภาษาคำสั่งบูลีน จะมีความสัมพันธ์กันกับภาษาแลดเดอร์ไคอะแกรม สามารถที่จะแปลความหมายถึงกันได้ดังตัวอย่างในตารางที่ 2.5 ซึ่งเป็นชุดคำสั่งบูลีนที่เขียนจากแลดเดอร์ไคอะแกรม ในรูปที่ 2.12 การเขียนโปรแกรมเป็นแลดเดอร์ไคอะแกรมสามารถแสดงกระบวนการของโปรแกรมได้ง่ายต่อความเข้าใจ แต่ไม่สะดวกในการป้อนลงเครื่องควบคุมจึงมีการแปลงแลดเดอร์ไคอะแกรมเป็นคำสั่งบูลีน เพื่อป้อนลงชุดรับข้อมูลแบบพกพาของเครื่องควบคุม PLC/PC ซึ่งจะทำให้กระทำได้สะดวกมากขึ้น

ตารางที่ 2.5 ตัวอย่างโปรแกรมชุดคำสั่งบูลีน

แอดเดรส	คำสั่ง	อุปกรณ์อินพุท/เอาต์พุท
0000	LD NOT	S1
0001	AND	S2
0002	AND NOT	S3
0003	OUT	K1
0004	LD NOT	S1
0005	AND NOT	S2
0006	AND	S3
0007	OUT	K2
0008	END	

2.3.3 โปรแกรมคำสั่งภาษาบล็อค

ภาษาคำสั่งในรูปบล็อค เป็นการเขียนโปรแกรมคำสั่งของ PC โดยใช้ชื่อสัญลักษณ์ต่าง ๆ คล้ายกับภาษาแลดเดอร์ แต่จัดไว้ในบล็อครูปสี่เหลี่ยมภาษาบล็อคนี้ จะใช้กับคำสั่ง หรือการควบคุมที่ค่อนข้างซับซ้อน หรือมีข้อมูลที่เป็นตัวเลขเกี่ยวข้อง เช่น การควบคุมที่มีการคำนวณทางคณิตศาสตร์ และการควบคุมตำแหน่งเครื่องจักร โดยภาษาบล็อคมักใช้ร่วมกับภาษาแลดเดอร์

คำสั่งภาษาบล็อคแบ่งออกเป็น 4 กลุ่มคำสั่ง คือ

1. สั่งหน่วงเวลาและนับจำนวน
2. สั่งการคำนวณทางคณิตศาสตร์
3. คำสั่งการจัดการข้อมูล
4. คำสั่งการเคลื่อนย้ายข้อมูล

ลักษณะคำสั่งบล็อกประกอบด้วย คำสั่งหรือสัญญาณควบคุม ใช้รับข้อมูลสัญญาณจาก หน่วยอินพุท/เอาต์พุท และอุปกรณ์ภายใน สัญญาณเอาต์พุทของคำสั่งบล็อกจะทำหน้าที่ควบคุม และแสดงผลการการปฏิบัติคำสั่งบล็อก

2.3.4 โปรแกรมคำสั่งภาษาระดับสูง

-คำสั่งข้อความภาษาอังกฤษ

คำสั่งข้อความภาษาอังกฤษที่ใช้กับ PC ถูกดัดแปลงจากภาษาระดับสูงของคอมพิวเตอร์ เช่น ภาษา BASIC และ PASCAL ทำให้การเขียนโปรแกรมมีความคล่องตัวและมีประสิทธิภาพสูงขึ้น โปรแกรมที่เขียนขึ้นอ่านและเข้าใจง่ายเหมาะสำหรับการตรวจสอบแก้ไขในภายหลัง PC ที่ใช้คำสั่งข้อความภาษาอังกฤษ มักเป็น PC ขนาดใหญ่มีการคำนวณที่ซับซ้อนและการจัดการข้อมูลจำนวนมาก

-ภาษาฟังก์ชันชาร์ท

ภาษาฟังก์ชันชาร์ท (FUNCTION CHART) หรือ ซีเควินซ์เชี่ยลฟังก์ชันชาร์ท (SEQUENTIAL FUNCTION CHART) เรียกกันย่อ ๆ SFC เป็นภาษาที่ใช้บรรยายการควบคุม ซีเควินซ์ โดยใช้แผนภาพสถานะเป็นภาษาที่เข้าใจง่าย เพราะสามารถเขียนได้ทันทีจากการทำงานของเครื่องจักรหรือข้อกำหนดในการควบคุม ภาษานี้กำลังอยู่ในระหว่างการพิจารณาของ คณะกรรมการมาตรฐานสากล IEC ให้เป็นภาษาสากลโดยกำหนดเป็นมาตรฐาน IEC 484

2.4 การเลือกใช้ภาษา PC

ภาษา PC ทุกภาษามักมีข้อดีและข้อจำกัดแตกต่างกันการเลือกใช้ภาษา และคำสั่ง ในการเขียนโปรแกรมควรพิจารณาในสิ่งต่อไปนี้

1. ความถนัดของผู้ใช้

การเลือกภาษาที่ใช้เขียน โปรแกรม ต้องพิจารณาความถนัดและความคุ้นเคยของผู้ใช้ เป็นอันดับแรก เช่น ผู้ใช้คุ้นเคยกับวงจรลีสเลย์ควบคุมใช้ภาษาแลดเดอร์ในการเขียนโปรแกรม เพราะทำให้เรียนรู้เร็วและปรับตัวได้เร็วกว่าการใช้ภาษาอื่น

2. ลักษณะของภาษาที่จะใช้

คำสั่งของ PC แต่ละภาษามีข้อดี และเหมาะที่จะใช้แก้ปัญหาต่างกัน ภาษาแลดเดอร์และภาษาลูตินเหมาะสำหรับการทดแทนอุปกรณ์รีเลย์ ตัวตั้งเวลาและตัวนับเป็นการควบคุมงานใน ลักษณะ ON – OFF , ภาษาบล็อก เหมาะสำหรับการควบคุมที่ค่อนข้างซับซ้อนมีการใช้ข้อมูลที่เป็นตัวเลข เป็นการควบคุมแบบอนาล็อก , การควบคุมตำแหน่งเครื่องจักรคำสั่งข้อความภาษาอังกฤษ เหมาะสำหรับการควบคุมที่ยุ่งยากซับซ้อนมีการคำนวณทางคณิตศาสตร์และให้ข้อมูลจำนวนมาก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. ลักษณะและขนาดของ PC

การสำรวจ หรือทำความเข้าใจเกี่ยวกับประสิทธิภาพของ PC ที่จะนำมาใช้ดูว่า อินพุท/เอาต์พุท เท่าไหร่ มีฟังก์ชันพิเศษอย่างไรบ้าง

4. ลักษณะงานที่ต้องการควบคุม

การใช้ ภาษา PC เพื่อให้ประสิทธิภาพที่ดีนั้นต้องรู้อุปกรณ์ ลำดับขั้นตอนการทำงานจากเบื้องต้นจนถึงสิ้นสุดการทำงาน

2.5 ภาษาที่ใช้กับหน่วยลอจิกอินพุท / เอาต์พุท

ในที่นี้ข้อกกล่าวถึงภาษาที่เป็นพื้นฐานในการนำมาควบคุมแบบ ON-OFF และภาษาซีเคิร์ฟ์เท่านั้นเพราะเป็นภาษาที่ง่ายต่อการทำความเข้าใจ และ เขียนโปรแกรม ภาษาที่นิยมใช้ก็คือ ภาษาแลคเคอร์ และภาษา บูลีนซึ่งกล่าวถึงในลักษณะที่ควบคุมกันไป ในปัจจุบันคำสั่งภาษาคำสั่งแลคเคอร์และภาษาบูลีนประกอบด้วย 6 กลุ่มคำสั่งด้วยกันคือ วงจรรีเลย์ และปฏิบัติลอจิก , การหนดเวลาและนับจำนวน , การคำนวณทางคณิตศาสตร์ , การจัดการข้อมูล และคำสั่งควบคุมโปรแกรม แต่ในที่นี้จะขอกกล่าวถึงเฉพาะกลุ่มคำสั่ง วงจรรีเลย์และปฏิบัติลอจิก กับการหนดเวลาและนับจำนวนเท่านั้น

ข้อสังเกตความแตกต่างระหว่างภาษาแลคเคอร์และภาษาบูลีนคือ ภาษาบูลีนจะอยู่ในรูปของคำสั่งส่วนภาษาแลคเคอร์จะอยู่ในรูปของสัญลักษณ์

คำสั่ง : LOAD

สัญลักษณ์ : 

ความหมาย : เป็นการนำค่าสถานะของอินพุท เอาต์พุท ตัวตั้งเวลา ตัวนับ ซีฟรีจิสเตอร์ หรือ รีเลย์ภายในที่กำหนดเข้ามา

หมายเหตุ : คำสั่งนี้ PC บ้างเครื่องอาจใช้เป็นคำสั่ง STORE หรือ ORIGIN

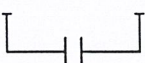
คำสั่ง : AND

สัญลักษณ์ : 

ความหมาย : เป็นการนำค่าสถานะของอินพุท เอาต์พุท ตัวตั้งเวลา ตัวนับ ซีฟรีจิสเตอร์ หรือรีเลย์ภายในที่กำหนดเข้ามาทำลอจิก AND กับค่าสถานะปัจจุบัน

- การปฏิบัติลอจิก AND เอาพุท Y จะมีสถานะลอจิก “1” ถ้าทุกสถานะเป็นลอจิก “1”

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คำสั่ง : OR
 สัญลักษณ์ : 

ความหมาย : เป็นการนำค่าสถานะของอินพุต เอาท์พุต ตัวตั้งเวลา ตัวนับ ชิฟท์รีจิสเตอร์ หรือรีเลย์ภายใน ที่กำหนดเข้ามาทำลอจิก OR กับค่าสถานะปัจจุบัน
 - การปฏิบัติลอจิก OR เอาท์พุต Y จะมีสถานะ “1” ถ้าอินพุต เพียง 1 อินพุตมีสถานะเป็นลอจิก “1”

คำสั่ง : NOT
 สัญลักษณ์ : 

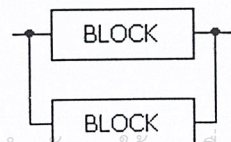
ความหมาย : เป็นการกระทำลอจิก NOT กับค่าสถานะปัจจุบัน โดยปกติแล้วคำสั่งนี้จะหมายถึงหน้าสัมผัสปกติของอุปกรณ์ต่าง ๆ ของ PC โดยจะใช้ร่วมกับ LOAD, AND และ OR

หมายเหตุ : คำสั่งนี้ PC บางเครื่องอาจใช้เป็นคำสั่ง INVERSE
 - การปฏิบัติลอจิก NOT เอาท์พุตจะมีสถานะตรงข้ามกับสถานะอินพุตกล่าวคือค่าอินพุตมีสถานะ “0” เอาท์พุตจะมีสถานะ “1” และในทำนองเดียวกันถ้าอินพุตมีสถานะ “1” เอาท์พุตจะมีสถานะ “0”

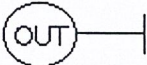
คำสั่ง : AND LOAD
 สัญลักษณ์ : 

ความหมาย : เป็นการนำค่าสถานะที่เก็บรักษาไว้มากมาทำลอจิก AND กับค่าสถานะปัจจุบัน

หมายเหตุ : คำสั่งนี้ PC บางเครื่องอาจใช้เป็นคำสั่ง AND BLOCK หรือ AND STORE

คำสั่ง : OR LOAD
 สัญลักษณ์ : 

ความหมาย : เป็นการนำค่าสถานะที่เก็บรักษาไว้มากระทำลอจิก OR กับค่าสถานะปัจจุบัน
 หมายเหตุ : คำสั่งนี้ PC บางเครื่องอาจใช้เป็นคำสั่ง OR BLOCK หรือ OR STORE

คำสั่ง : OUT
 สัญลักษณ์ : 

ความหมาย : เป็นการให้ค่าสถานะแก่อุปกรณ์ทางเอาต์พุตต่าง ๆ โดยทั่วไป จะได้แก่
 Output , Internal Relay , Special Relay


คำสั่ง : TIMER
 สัญลักษณ์ : 

ความหมาย : เป็นการเรียกใช้ตัวตั้งเวลา ซึ่งสามารถหนดเวลาการทำงานหรือกำหนด
 ค่าเวลาได้

ซึ่งครั้งนี้จะขึ้นอยู่กับความสามารถของเครื่อง PC ที่ใช้อยู่เช่น กำหนดเวลาได้
 ระหว่าง 0.1 - 999.9 วินาที เมื่อมีการเรียกใช้ตัวตั้งเวลาภายในเครื่อง PC จะทำหน้าที่แทนตัวตั้ง
 เวลาของวงจรรีเลย์ในการปิดหรือเปิดวงจรไฟฟ้าเมื่อเวลาผ่านไปช่วงระยะเวลาหนึ่งตามที่ได้ตั้งเวลา
 จะนับสัญญาณนาฬิกา ของ CPU ที่มีค่าเวลา 0.1 หรือ 1 วินาที ซึ่งแล้วแต่ความสามารถ
 ของเครื่อง เช่นต้องการหนดเวลา 3 วินาที ตัวตั้งเวลาจะนับสัญญาณนาฬิกาที่มีค่าเวลา 0.1 วินาที
 เป็นจำนวน 30 สัญญาณ หรือ 3 สัญญาณ ในกรณีสัญญาณนาฬิกามีคาบเวลาเป็น 1 วินาที

คำสั่ง : COUNTER
 สัญลักษณ์ : 

ความหมาย : เป็นการเรียกใช้ตัวนับ ซึ่งทำหน้าที่แทนตัวนับของวงจรรีเลย์ในการเปิดหรือ
 ปิด วงจรไฟฟ้า เมื่อนับได้ตามจำนวนที่ต้องการ ตัวนับจะทำการตรวจนับสัญญาณพัลส์ จาก
 อุปกรณ์อินพุต / เอาต์พุต และอุปกรณ์ภายในสัญญาณเมื่อสัญญาณมีการเปลี่ยนแปลง 1 ครั้งจะนับ
 1 ครั้ง เมื่อมีการใช้ COUNTER จะมีการใช้สัญญาณควบคู่กัน 2 สัญญาณ คือสัญญาณสำหรับ
 RESET และสัญญาณพัลส์ ที่ต้องการนับ

คำสั่ง : END
 สัญลักษณ์ : -- 

ความหมาย : คำสั่งนี้จะถูกใช้ เมื่อสิ้นสุดการเขียนโปรแกรม หรืออาจกล่าวได้ว่าเป็น ส่วนสุดท้ายของโปรแกรมนั้นเอง

2.6 หลักการเขียน LADDER DIAGRAM

LADDER DIAGRAM เป็นระบบการเขียนโปรแกรมที่นิยมมากแบบหนึ่ง ลักษณะของ LADDER DIAGRAM แตกต่างไปจาก LOGIC DIAGRAM โดยหลักของงานด้านการควบคุม ในยุคของรีเลย์ ไม่ต้องเปลี่ยนแปลงแนวความคิดความเข้าใจกับ DIAGRAM ของระบบเพียงแต่ เรียนรู้ถึงวิธีการ โปรแกรมเพิ่มขึ้นเท่านั้น

หลักการเขียน LADDER DIAGRAM พอจะสรุปเป็นข้อ ๆ ได้ดังต่อไปนี้

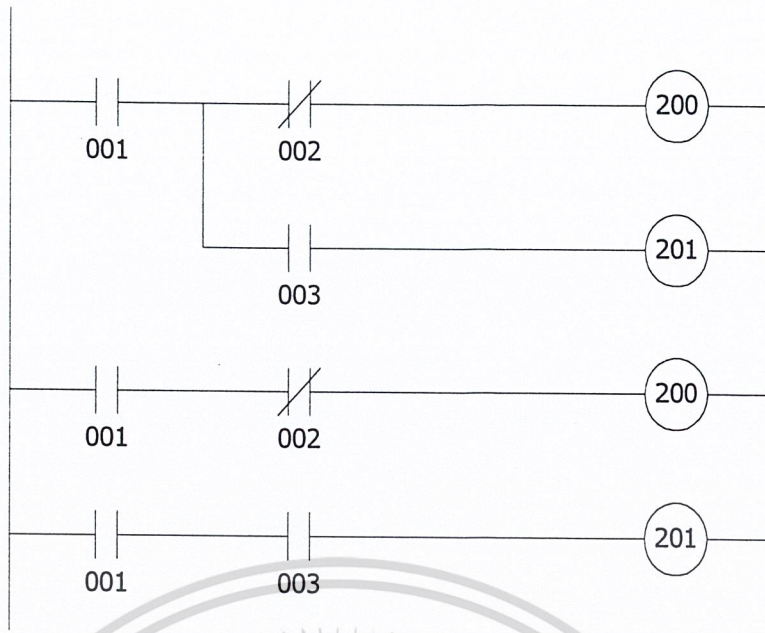
1. หมายเลขกำกับหน้าสัมผัสหรือ ขดลวดของอินพุท/เอาต์พุตรีเลย์ รีเลย์ภายใน ตัวตั้ง เวลาและตัวนับ ของ LADDER DIAGRAM จะนำมาจากข้อกำหนดของเครื่อง PLC ที่ใช้อยู่

Name	CH No. and Relay No									
	CH 00	CH 01	CH 02	CH 03	CH 04	CH 05	CH 06	CH 07	CH 08	CH 09
Input relay (0000 to 0415)	00	08	00	08	00	08	00	08	00	08
	01	09	01	09	01	09	01	09	01	09
	02	10	02	10	02	10	02	10	02	10
	03	11	03	11	03	11	03	11	03	11
	04	12	04	12	04	12	04	12	04	12
	05	13	05	13	05	13	05	13	05	13
	06	14	06	14	06	14	06	14	06	14
	07	15	07	15	07	15	07	15	07	15
Output relay (0500 to 0915)	05	00	05	00	05	00	05	00	05	00
	01	09	01	09	01	09	01	09	01	09
	02	10	02	10	02	10	02	10	02	10
	03	11	03	11	03	11	03	11	03	11
	04	12	04	12	04	12	04	12	04	12
	05	13	05	13	05	13	05	13	05	13
	06	14	06	14	06	14	06	14	06	14
	07	15	07	15	07	15	07	15	07	15
Internal auxiliary relay (1000 to 1207)	10	00	08	00	08	00	08	00	08	00
	01	09	01	09	01	09	01	09	01	09
	02	10	02	10	02	10	02	10	02	10
	03	11	03	11	03	11	03	11	03	11
	04	12	04	12	04	12	04	12	04	12
	05	13	05	13	05	13	05	13	05	13
	06	14	06	14	06	14	06	14	06	14
	07	15	07	15	07	15	07	15	07	15
	15	00	08	00	08	00	08	00	08	00
	01	09	01	09	01	09	01	09	01	09
	02	10	02	10	02	10	02	10	02	10
	03	11	03	11	03	11	03	11	03	11
	04	12	04	12	04	12	04	12	04	12
	05	13	05	13	05	13	05	13	05	13
	06	14	06	14	06	14	06	14	06	14
	07	15	07	15	07	15	07	15	07	15

รูปที่ 2.13 แสดงหมายเลขกำกับอุปกรณ์ต่าง ๆ ของ PLC เครื่องหนึ่ง

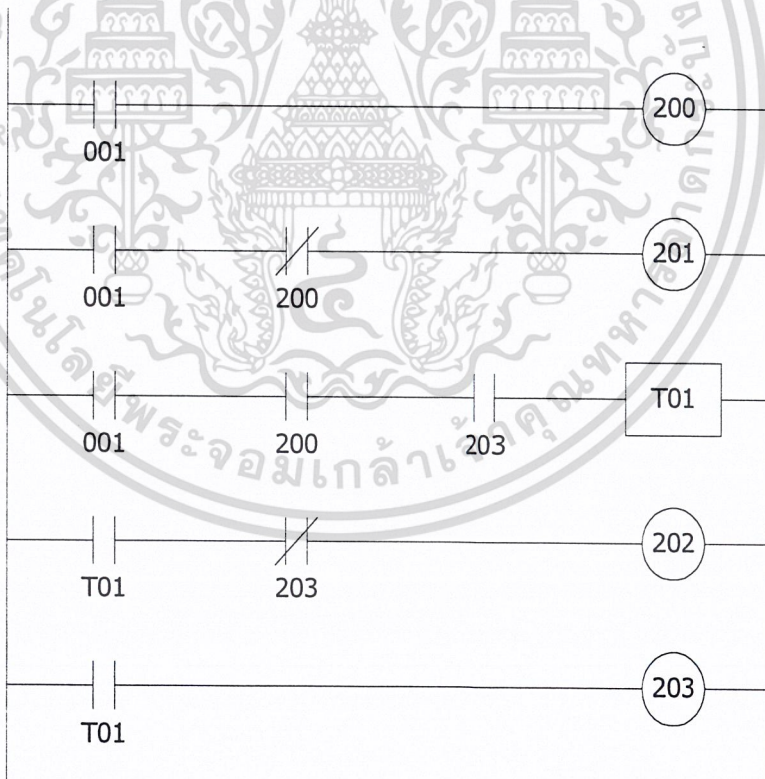
2. การเขียนโปรแกรม ควรเขียนให้อ่าน และเข้าใจง่าย ไม่ควรเขียนโปรแกรมให้สั้น แต่ซับซ้อนเพียงเพื่อประหยัดหน่วยความจำเพียงเล็กน้อย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.14 แสดงตัวอย่าง Ladder Diagram

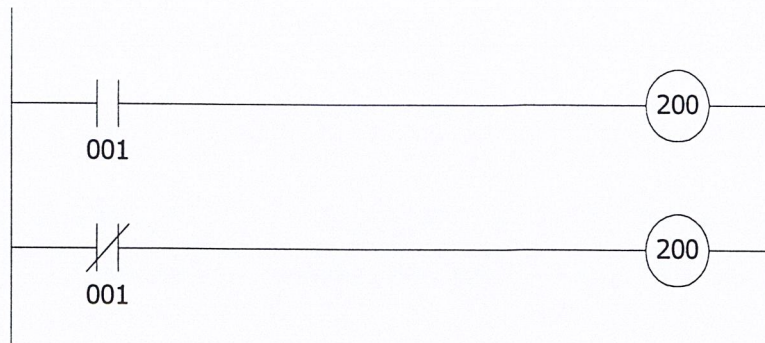
3. หน้าสัมผัสของอินพุต/เอาต์พุตรีเลย์, รีเลย์ภายใน, ตัวตั้งเวลาและตัวนับหมายเลขเดียวกัน สามารถเรียกใช้ในโปรแกรมทีครั้งก็ได้ไม่จำกัดจำนวน



รูปที่ 2.15 แสดงตัวอย่าง Ladder Diagram

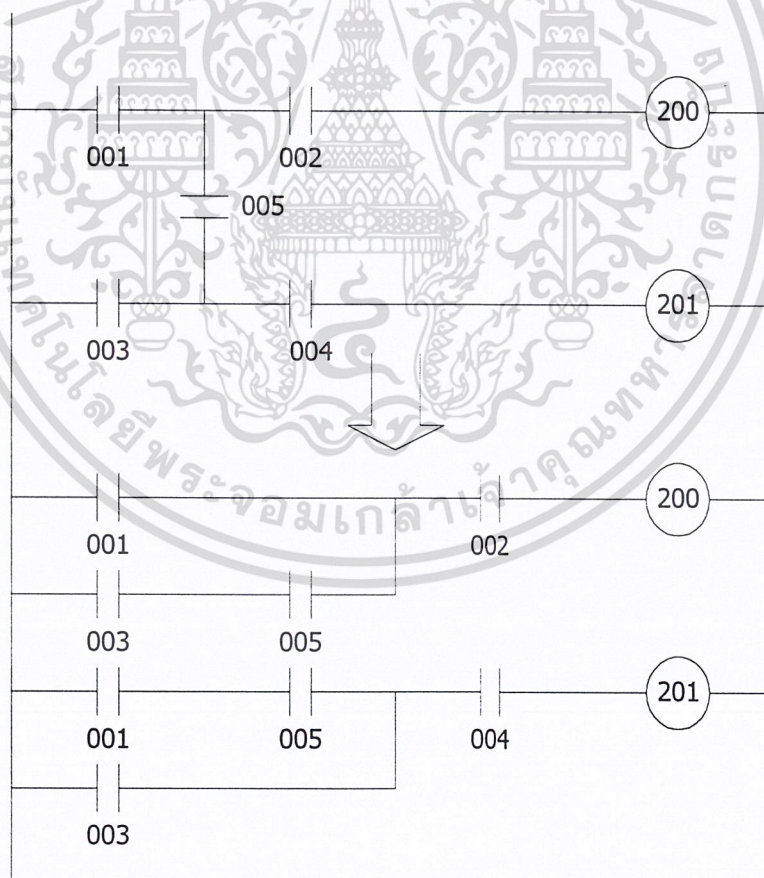
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. ส่งสัญญาณควบคุมซ้ำกันมากกว่าหนึ่งครั้งไปที่ ขดลวดเอาต์พุตรีเลย์ หรือรีเลย์ภายใน หมายเลขเดียวกันไม่ได้ (นอกจากมีฟังก์ชันพิเศษเข้ามาช่วย)



รูปที่ 2.16 แสดงตัวอย่าง Ladder Diagram

5. สัญญาณควบคุมไหลจากซ้ายไปขวาทิศทางเดียวไม่ย้อนกลับ



รูปที่ 2.17 แสดงตัวอย่าง Ladder Diagram

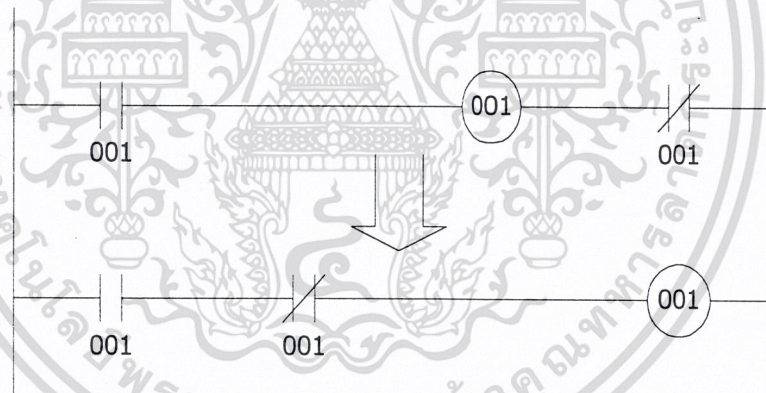
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6. ตัวตั้งเวลาหรือตัวนับหมายเลขเดียวกัน ก็จะถูกเรียกใช้ในโปรแกรมมากกว่าหนึ่งครั้งไม่ได้ และ PC บางเครื่องไม่อนุญาตให้ใช้ตัวตั้งเวลา และตัวนับหมายเลขเดียวกันในโปรแกรม



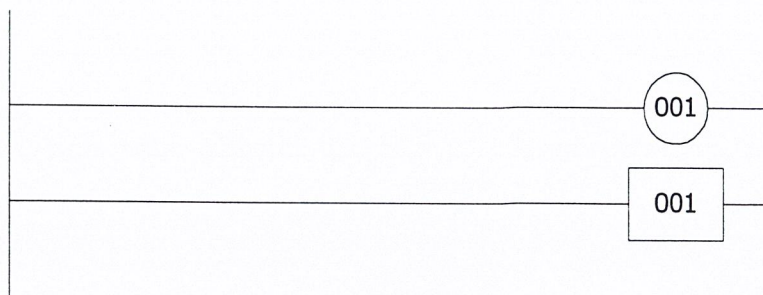
รูปที่ 2.18 แสดงตัวอย่าง Ladder Diagram

7. วางตำแหน่งหน้าสัมผัสหลังขดลวดรีเลย์ไม่ได้



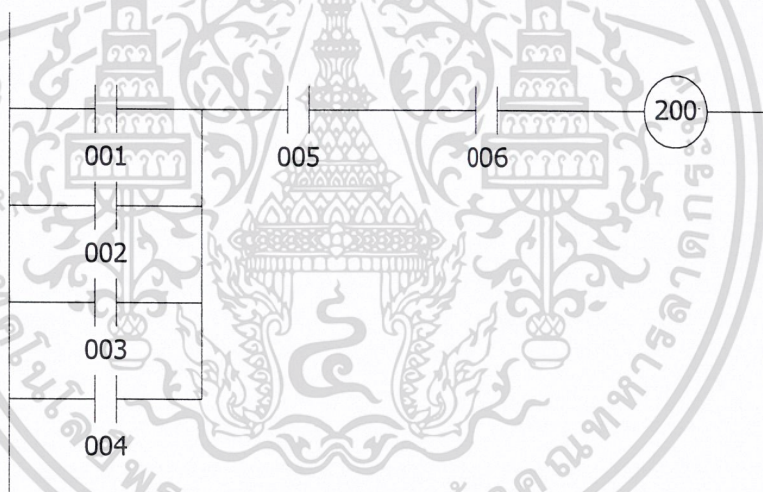
รูปที่ 2.19 แสดงตัวอย่าง Ladder Diagram

8. ขดลวดของเอาต์พุทหรือรีเลย์ภายในต่อโดยตรงกับบัสด้านซ้ายมือไม่ได้ (ถ้าจำเป็นต้องต่อหน้าสัมผัสที่มีสถานะ “NO” ระหว่างบัสกับขดลวดรีเลย์โดยใช้หน้าสัมผัสของรีเลย์ภายในที่มีสถานะ“NO” ตลอดเวลา)



รูปที่ 2.20 แสดงตัวอย่าง Ladder Diagram

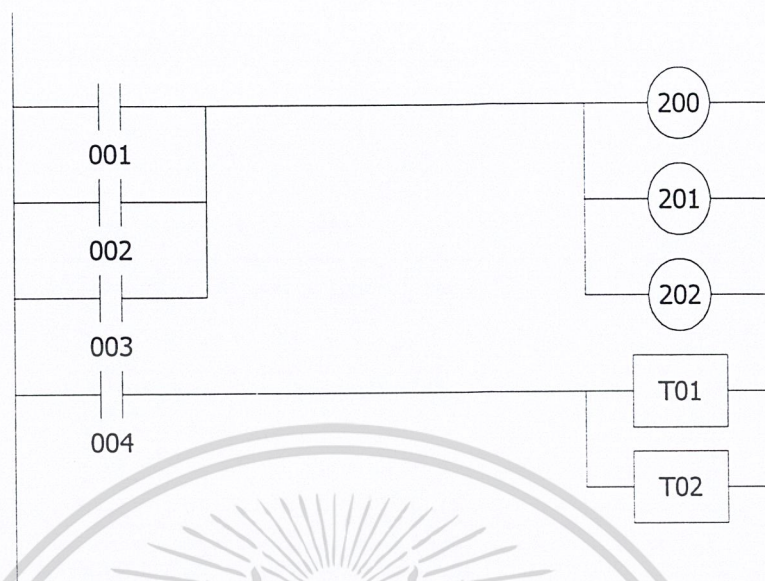
9. หน้าสัมผัสของอินพุท/เอาต์พุทรีเลย์ ,รีเลย์ภายใน ตัวตั้งเวลาและตัวนับจะต่อขนาน หรืออนุกรมกันจำนวนมากเท่าใดก็ได้ไม่จำกัดจำนวน



รูปที่ 2.21 แสดงตัวอย่าง Ladder Diagram

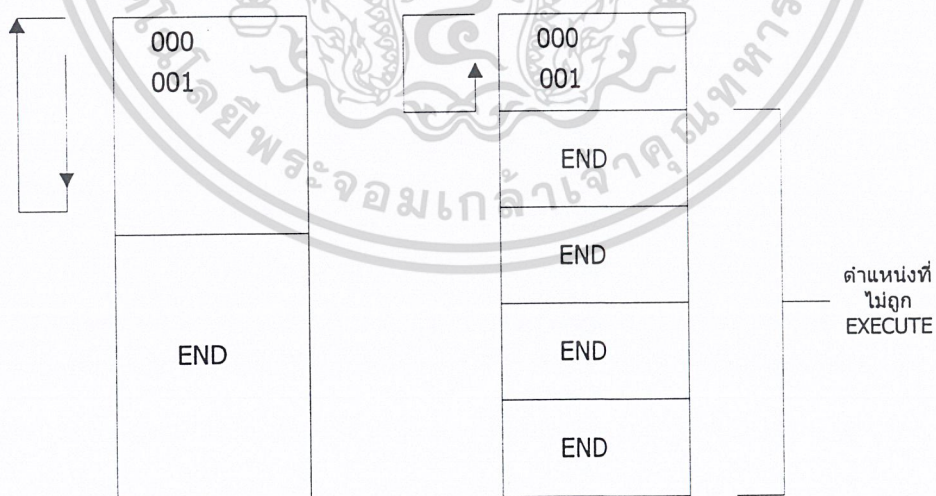
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

10. ขดลวดของเอาต์พุตรีเลย์,รีเลย์ภายใน และตัวตั้งเวลาสามารถนำมาต่อขนานกันได้



รูปที่ 2.22 แสดงตัวอย่าง Ladder Diagram

11. โปรแกรมที่เขียนขึ้น CPU จะทำการ EXECUTED จากบรรทัดแรกจนกระทั่งถึงคำสั่ง END ที่เป็นคำสั่งแรกซึ่งถูกใช้ โดยที่ END อาจมีหลายตำแหน่งก็เป็นได้ ซึ่งอาจจะต้องมีคำสั่งหรือฟังก์ชันพิเศษเข้ามาช่วย ที่เป็นเช่นนี้เพื่อจุดประสงค์สำหรับการทดลองโปรแกรม ทำให้ง่ายต่อการตรวจสอบ และแก้ไขโปรแกรม



รูปที่ 2.23 แสดงตัวอย่าง Ladder Diagram

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.7 การเขียนโปรแกรมคำสั่งบูลีนจาก LADDER DIAGRAM

สิ่งที่ทราบแล้วตั้งแต่ตอนต้นภาษาแลดเดอร์ มีลักษณะโครงสร้างคล้ายกับการควบคุมด้วยระบบรีเลย์ จึงทำให้ง่ายต่อการทำความเข้าใจและเขียนโปรแกรม แต่เครื่อง PC ตั้งแต่ขนาดกลางลงมาซึ่งใช้กันในปัจจุบันจะนิยมใช้คำสั่งบูลีนในการเขียนโปรแกรมการควบคุมกระบวนการทำงานต่าง ๆ ซึ่งการเขียนโปรแกรมคำสั่งบูลีนนั้นเป็นการเทียบมาจาก LADDER DIAGRAM นั้นเอง โครงสร้างของโปรแกรม คำสั่งบูลีนประกอบด้วยส่วนหลัก ๆ ที่สำคัญ 3 ส่วนด้วยกันคือ

1. หมายเลขกำหนดบรรทัดของโปรแกรม (ADDRESS OR STEP)
2. คำสั่ง (INSTRUCTION WORD)
3. หมายเลขกำกับอุปกรณ์และหน้าสัมผัสต่าง ๆ (DATA)

Address	Instruction	Operand	Code
0000	LD NOT	0000	90
0001	AND	0105	85
0002	OR NOT	2007	99
0003	OUT	0700	9E
0004	END		00

รูปที่ 2.24 แสดงโครงสร้างของโปรแกรมคำสั่งบูลีน

1. หมายเลขกำหนดบรรทัดของโปรแกรม :
เพื่อความสะดวกในการเขียนโปรแกรม โดยปกติแล้วหมายเลขกำหนดบรรทัดของโปรแกรมจะขึ้นอยู่กับขนาดของหน่วยความจำผู้ใช้ของ PC
2. คำสั่ง :
เป็นการนำเอาคำสั่งต่าง ๆ ที่กล่าวถึงในตอนต้นมาใช้งาน เพื่อควบคุมกระบวนการต่าง ๆ ตามความต้องการ
3. หมายเลขกำกับอุปกรณ์และหน้าสัมผัสต่าง ๆ :

จะเป็นหมายเลขที่กำหนดของอุปกรณ์ต่างๆ ของ PC โดยที่หมายเลขต่างๆ เหล่านี้จะนำมากำหนดบน LADDER DIAGRAM แล้วเปลี่ยนมาเป็นคำสั่งบูลีน โดยทั่วไปแล้ว หมายเลขต่างๆ เหล่านี้จะแตกต่างกันออกไปแล้วแต่เครื่อง PC ที่ใช้อยู่



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3

เครื่องควบคุมแบบตรรกะที่โปรแกรมได้กับการติดต่อสื่อสาร

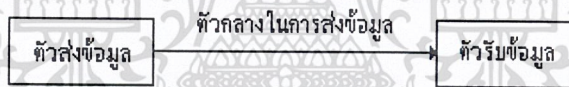
เครื่องควบคุม PLC/PC นอกจากจะใช้ในการควบคุมเครื่องจักรแล้ว ยังได้มีการพัฒนาขีดความสามารถในการสื่อสารข้อมูล การควบคุมระยะไกล การจัดการข้อมูล เพื่อเพิ่มประสิทธิภาพในการควบคุมให้กว้างมากยิ่งขึ้น ในบทนี้จะกล่าวถึงพื้นฐานการสื่อสารข้อมูลทั่วไปและการสื่อสารข้อมูลของเครื่องควบคุม PLC/PC

3.1 การติดต่อสื่อสารข้อมูลทั่วไป

ส่วนประกอบเบื้องต้นในการสื่อสารข้อมูลแบ่งออกได้เป็น 3 ส่วนคือ

1. ฝ่ายกำเนิดข้อมูล (Transmitter)
2. ตัวกลางในการส่งผ่านข้อมูล (Medium)
3. ฝ่ายรับข้อมูล (Receiver)

ซึ่งแต่ละส่วนมีความสัมพันธ์กันดังรูปที่ 3-1



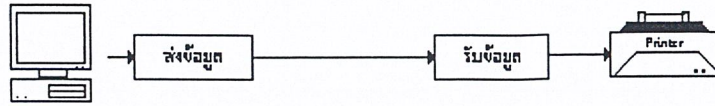
รูปที่ 3.1 แสดงความสัมพันธ์ของส่วนประกอบหลักในการสื่อสารข้อมูล

การจำแนกรูปแบบในการสื่อสารข้อมูลอาจจัดแบ่งได้หลายแบบ เช่น แบ่งตามชนิดของสัญญาณของข้อมูล คือ การสื่อสารข้อมูล อนาล็อก (Analog) และ การสื่อสารข้อมูลดิจิทัลชนิดที่เกี่ยวข้องกับเครื่องคอมพิวเตอร์และเครื่องควบคุม PLC/PC

3.1.1 วิธีการส่งข้อมูล

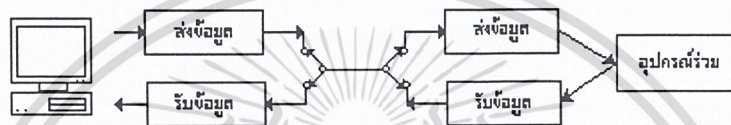
แบ่งได้ออกเป็น 3 แบบ คือ

1. การส่งแบบทิศทางเดียว (Simplex) การส่งแบบนี้ ทิศทางการส่งจะคงที่โดยกำหนดในครั้งแรกว่าฝ่ายใดเป็นฝ่ายส่ง ฝ่ายใดเป็นฝ่ายรับ การส่งแบบนี้แม้ไม่คล่องตัวเหมาะสมในการใช้งานบางประเภท



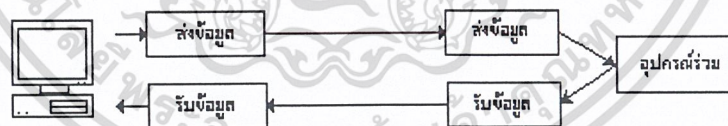
รูปที่ 3.2 การส่งแบบทิศทางเดียว (Simplex)

2. การส่งแบบฮาล์ฟดูเพล็กซ์ (Half Duplex) เป็นการส่งข้อมูลในทิศทางหนึ่งในเวลาหนึ่งคือ ทั้ง 2 สถานีสามารถผลัดกันส่งได้แต่จะส่งพร้อมๆกันไม่ได้



รูปที่ 3.3 การส่งแบบฮาล์ฟดูเพล็กซ์ (Half Duplex)

3. การส่งข้อมูลแบบสองทิศทาง (Full Duplex) เป็นการส่งข้อมูลในทิศทางใดก็ได้ เวลาใดก็ได้ คือ สามารถส่งแบบรับข้อมูลในเวลาเดียวกันได้



รูปที่ 3.4 การส่งแบบสองทิศทาง (Full Duplex)

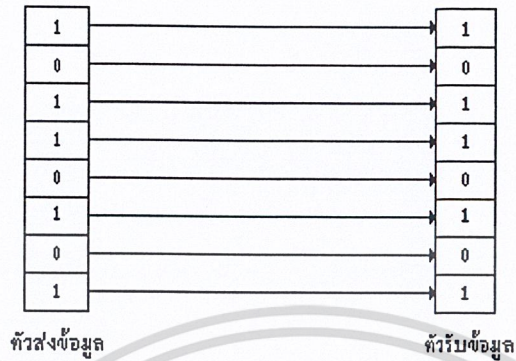
วิธีการสื่อสารข้อมูลดิจิทัลของคอมพิวเตอร์แบ่งเป็น 2 ชนิด

1. การสื่อสารข้อมูลแบบขนาน (Parallel)

เป็นการติดต่อสื่อสารโดยส่งข้อมูลออกเป็นครั้งละ 1 ไบต์ คือ ครั้งละ 8 บิต จากอุปกรณ์ส่งไปยังอุปกรณ์รับ ดังนั้นตัวกลางระหว่างเครื่องคอมพิวเตอร์ทั้งสองเครื่องจะต้องมีช่องทางให้ข้อมูลเดินทางอย่างน้อย 8 ช่องทาง โดยมากจะให้เป็นสายขนาน แต่เนื่องจากมีสัญญาณสูญหายไปกับอิมพีแดนซ์ ของสาย ซึ่งมีความสัมพันธ์กับระยะทางของสาย ดังนั้นระยะทางระหว่าง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

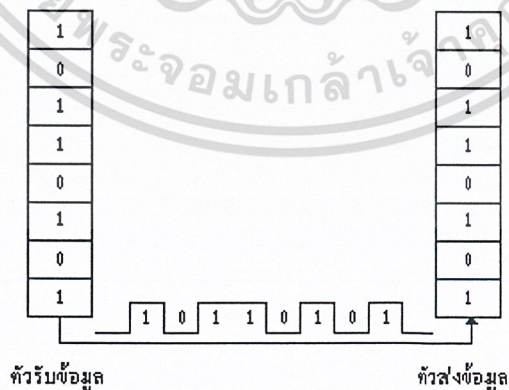
เครื่องสองเครื่องไม่ควรจะเกิน 100 ฟุต ทำให้การส่งแบบนี้ได้ระยะทางไม่ไกลนักแต่สามารถส่งข้อมูลได้รวดเร็วเพราะส่งออกมาพร้อมกันทีละ 8 บิต อุปกรณ์ที่ติดต่อบนขานานกับคอมพิวเตอร์ เช่น เครื่องพิมพ์ เป็นต้น



รูปที่ 3.5 การสื่อสารข้อมูลแบบขนาน

2. การสื่อสารข้อมูลแบบอนุกรม (Serial)

ในการติดต่อแบบนี้ ข้อมูลจะถูกส่งออกมาทีละบิต ระหว่าง จุดส่งและจุดรับ ซึ่งการส่งข้อมูลแบบนี้จะช้ากว่าการส่งแบบขนาน แต่ใช้ตัวกลางในการสื่อสารเพียงช่องเดียวหรือสายเพียงคู่เดียว ทำให้ค่าใช้จ่ายสำหรับสื่อกลางถูกกว่าแบบขนาน การส่งแบบนี้ใช้สำหรับส่งในระยะทางไกลมากกว่า 100 ฟุต ข้อมูลจากจุดส่งจะต้องถูกเปลี่ยนให้เป็นอนุกรมก่อนแล้วค่อยทยอยส่งออกไปทีละบิต ไปยังจุดรับ ที่จุดรับจะต้องมีรูปแบบในการเปลี่ยนข้อมูลที่ถูกส่งมาให้ เป็นแบบขนาน ในการแปลงต้องมีรูปแบบที่เหมาะสมเพื่อป้องกันการผิดพลาดของข้อมูล



รูปที่ 3.6 การสื่อสารข้อมูลแบบอนุกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.2 ข้อตกลงในการติดต่อสื่อสาร (โพรโทคอล : Protocols)

ในการสื่อสารข้อมูลจะต้องมีกฎหรือข้อกำหนดในการสื่อสารข้อมูล หรือที่นิยมเรียกว่า โพรโทคอล (Protocols) ซึ่งเป็นส่วนที่จะกำหนดมาตรฐานในการควบคุม และ จัดการระบบ การสื่อสารข้อมูล

สำหรับรายละเอียดที่กล่าวในหัวข้อนี้เกี่ยวข้องกับเฉพาะในส่วนของโพรโทคอลการควบคุม การเชื่อมโยงข้อมูล (Data Link Control Protocols หรือ DLCP) ซึ่งจัดการในส่วนของขั้นตอนและ หลักการต่างๆ คือ โครงสร้างและรายละเอียดของข้อมูล วิธีในการควบคุมการติดต่อสื่อสาร โดย DLCP แบ่งได้ตามโครงสร้างของข้อมูล 2 แบบ คือ Byte –Oriented Protocols และ Bit- Oriented Protocols

-ไบท์โอเรียนตโพรโทคอล (Byte-Oriented protocols)

โพรโทคอลแบบนี้ เป็นโพรโทคอลที่การสื่อสารข้อมูล และการควบคุมการทำงาน จะทำโดยใช้ลักษณะข้อมูลที่เป็นตัวอักษร (character) หรือไบท์ (byte) หรือ อาจหรือว่า Character Oriented Protocols ซึ่งแบ่งออกเป็น

1. อะซิงโครนัสโพรโทคอล (Asynchronous protocols) โพรโทคอลในการสื่อสารข้อมูล จะใช้การสื่อสารข้อมูลแบบ Half-Duplex ที่มีลักษณะการสื่อสารแบบอะซิงโครนัส ซึ่งเป็นการสื่อสารข้อมูลแบบพื้นฐานที่ใช้มานานเป็นเวลานานแล้ว และยังมีข้อดีที่ การสื่อสารข้อมูลแบบนี้มีใน โครงสร้างการทำงานที่ง่าย อุปกรณ์ที่ใช้ในการสื่อสาร ข้อมูลก็ไม่สลับซับซ้อน และมีราคาถูกโพรโทคอลแบบนี้จึงเหมาะสำหรับใช้ในระบบ ขนาดเล็ก
2. ไบนารีซิงโครนัสโพรโทคอล (Binary synchronus protocols) โพรโทคอลแบบนี้จะมี ลักษณะการทำงานข้อมูลเป็นลักษณะไบท์ และยังคงใช้การสื่อสารข้อมูลแบบ ซิงโครนัสมีรายละเอียดและขั้นตอนในการสื่อสารข้อมูลที่ทำให้ความน่าเชื่อถือมากกว่า อีกทั้งยังสามารถใช้อัตราเร็วในการสื่อสารข้อมูลที่สูงกว่าโดยตัวอย่างของการสื่อสาร ข้อมูลแบบนี้ที่ได้กำหนดเป็นมาตรฐานแล้วคือ การสื่อสารข้อมูลตามมาตรฐาน BSC (Binary Synchronous Communications) ซึ่งเป็น โพรโทคอลที่มีลักษณะของข้อมูล แบบไบท์ที่ได้รับความนิยมนำไปใช้งาน

- บิทโอเรียนตโพรโทคอล (Bit-oriented protocols)

โพรโทคอลแบบนี้เป็นโพรโทคอลที่การสื่อสารข้อมูล และการควบคุมการทำงาน จะทำโดยใช้ลักษณะข้อมูลที่เป็นบิท โดยมีตัวอย่างของการสื่อสารข้อมูล ในลักษณะนี้ที่มี การกำหนดขึ้นเป็น มาตรฐานแล้ว คือ HDLC (High – level Data Link Control) โดยมีโครงสร้าง ของข้อมูลแบบซิงโครนัส เช่นเดียวกับ BSC แต่ต่างกันที่มีลักษณะของข้อมูลเป็นแบบบิท ซึ่งโพรโทคอลแบบนี้ มีข้อดีที่สามารถสื่อสารข้อมูลแบบสองทางได้ทำให้การสื่อสารข้อมูลได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รวดเร็วกว่าแต่โปรโตคอลแบบนี้ก็มีรายละเอียด และโครงสร้างในการสื่อสารข้อมูลที่สลับซับซ้อนมากทำให้การควบคุมการทำงานทำได้ยากและต้องใช้อุปกรณ์ที่มีราคาสูง จึงไม่เหมาะสมที่จะนำไปใช้งานกับระบบขนาดเล็ก

- แพ็กเก็ต (Packet of Information)

รูปแบบของแพ็กเก็ตในระบบโครงข่ายจะประกอบด้วยส่วนต่างๆดังนี้

1. HEADER จะประกอบด้วย

-Preamble or start of packet indicator เป็นส่วนเริ่มแรกของแพ็กเก็ต และในบางระบบอาจใช้ในการซิงค์กับสัญญาณพิกาสองตัวส่งและตัวรับด้วย

- Control information ส่วนนี้เป็นข้อมูลที่บอกถึงวัตถุประสงค์ของแพ็กเก็ตนั้นว่า ใช้ทำอะไรเช่นเพื่อการจัดการระบบเพื่อดู status ของ mode หรือ อื่นๆ

นอกจากส่วนต่างๆ เหล่านี้แล้ว ในส่วน Header อาจจะมีส่วนที่เป็น Sequential Number เป็นส่วนที่บอกให้ทราบถึงลำดับของแพ็กเก็ตในกรณีที่ข้อมูลมีความยาวหลายแพ็กเก็ต

2. INFORMATION

- Data field เป็นส่วนของข้อมูลจริงที่ต้องการจะส่ง

3. TAILER

- Frame Check Sequence (FCS) เป็นส่วนที่ใช้ตรวจสอบความถูกต้องของข้อมูล ซึ่งอาจเป็น parity bit, Check sum หรือ CRC เป็นต้น

- End of Packet Indicator เป็นส่วนที่บอกให้ทราบว่าสิ้นสุดของข้อมูลแล้ว

PREAMBLE	DA	SA	CONTROL	INFORMATION	FCS	STOP
HEADER			INFORMATION		TAILER	

รูปที่ 3.7 แสดงโครงสร้างของแพ็กเก็ต

3.1.3 การควบคุมความผิดพลาดในการส่งข้อมูล (Error Control)

การที่วงจรส่งข้อมูลขาดชั่วขณะ และผลกระทบต่อสัญญาณรบกวน (Noise) ทำให้ระดับแรงดันไฟฟ้าลดลงเป็นผลให้เกิดความผิดพลาดในการส่ง ดังนั้นต้องทำการค้นหาความผิดพลาดที่เกิดขึ้นระหว่างการส่งข้อมูลและแก้ไขความผิดพลาดให้ถูกต้อง

- วิธีการเพิ่มบิตเข้าไปที่ข้อมูล (Parity Bit)

เพื่อตรวจสอบความผิดพลาดมี 2 วิธี

1. วิธี Parity แนวนอน

จะเพิ่ม 1 บิต เข้าไปที่แต่ละตัวอักษรที่จะส่ง โดยจะกำหนดว่าเป็นการตรวจสอบแบบ Odd หรือ Even Number แล้วผลรวมของข้อมูลจะเป็น Odd Number ที่ด้านรับจะทำการตรวจสอบว่าผลรวมของข้อมูลจะเป็น Odd Number หรือ Even Number

2. วิธี Parity แนวตั้ง

จะตรวจสอบความผิดพลาดโดยการเพิ่ม 1 บิตเข้าไปที่ในแต่ละบิต

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

Parity
แนวนอน

รูปที่ 3.8 แสดงรูปแบบการตรวจสอบด้วย Parity Bit

- วิธี Patrol Diffuse Inspection หรือ การใช้ FCS : Frame Check Sequence

สำหรับข้อมูลอันหนึ่งจะสร้าง Error Inspection Sign (CRC Sign) ขนาด 2 Byte ที่คำนวณได้บนพื้นฐานของกฎที่กำหนดไว้ แล้วเพิ่มเครื่องหมายไปที่ข้อมูลอันนั้นที่ทางด้านรับจะตรวจสอบข้อมูลโดย Inspection Sign ได้กำหนดไว้โดยสูตรการคำนวณที่เหมือนกัน (Blast Error ความผิดพลาดของข้อมูลต่อเนื่อง) ก็สามารถตรวจสอบได้ ทำให้มีความน่าเชื่อถือสูง

3.1.4 อัตราการส่งข้อมูล (Data Transmission Rate)

Transmission Rate นั้นแยกออกได้เป็น “Data Transmission Speed” และ “Modulation Speed”

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Data Transmission Speed หรือ Bit rate นั้น แสดงในรูปของจำนวน Bit ที่สามารถส่งได้ภายใน 1 วินาที มีหน่วยเป็น บิต ต่อวินาที โดยเขียนว่า Bit / Sec หรือ Bps จะแสดง Definition ของ Data Transmission Speed จะแสดงด้วยสมการข้างล่างนี้

$$S = (1/T) \log_2 n \quad (3.1)$$

โดยที่ $S =$ Data Transmission Speed

$T =$ Continue Time ของ 1 pulse ของ Transmission Pulse

$n =$ จำนวนของค่าสภาวะใน 1 Pulse

ในกรณีที่ Pulse 1 ลูก มีค่า 2 ค่าคือ 0 และ 1 เป็น สมการจะเป็น $S = 1/T$ โดยที่ T เป็นเวลาต่อเนื่อง (Continue Time) ของ Pulse 1 $1/T$ จึงแสดงถึงจำนวนของ Pulse ใน 1 วินาที ซึ่งก็คือจำนวนบิตที่สามารถส่งไปได้ใน 1 วินาทีนั่นเอง

Modulation Speed หรือ Baud rate ก็คือจำนวนครั้งที่สามารถ Modulate ได้ใน 1 วินาที วิธีการ Modulation นั้นมี Amplitude Mod , Frequency Mod , Phase Mod และอื่นๆ ในกรณีที่ T แสดงถึงช่วงเวลาในการเปลี่ยนแปลงสภาวะนั้น จะสามารถกำหนดได้ดังสมการข้างล่างนี้ โดยหน่วยเป็น Baud

$$B = 1/T \quad (3.2)$$

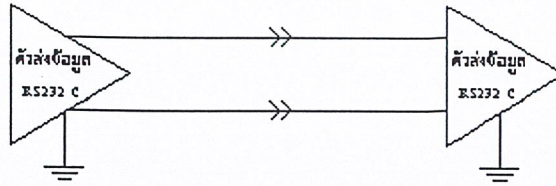
โดยที่ $B =$ Modulation Speed (Baud)

$T =$ เวลาในการเปลี่ยนจากสภาวะหนึ่งเป็นอีกสภาวะ

3.1.5 มาตรฐานในการสื่อสารข้อมูลแบบอนุกรม

-การสื่อสารข้อมูลแบบอนุกรมตามมาตรฐาน RS-232C

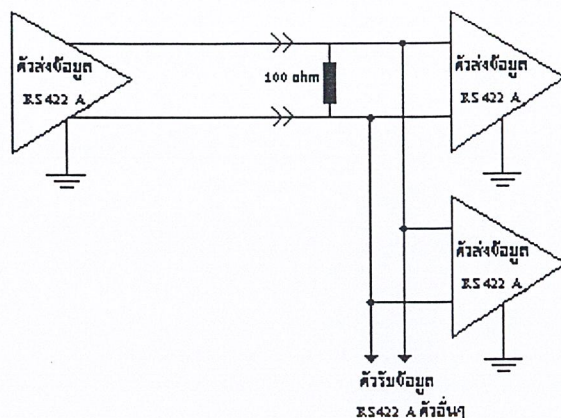
การสื่อสารข้อมูลแบบอนุกรมที่ใช้งานอยู่ในปัจจุบันนั้น ได้มีการกำหนดมาตรฐานการรับส่งข้อมูลไว้หลายแบบด้วยกัน แต่ที่ได้รับความนิยมนำมาใช้งานอย่างมาก คือ การสื่อสารข้อมูลแบบอนุกรมตามมาตรฐาน RS-232C และที่มาตรฐานนี้เป็นที่นิยมเนื่องจากเป็น ระบบ การสื่อสารข้อมูลที่ใช้ในเครื่องไมโครคอมพิวเตอร์ IBM PC ซึ่งเป็นคอมพิวเตอร์ที่มีใช้อย่างแพร่หลายมาก จากอดีตจนถึงปัจจุบันมาตรฐานการสื่อสารนี้ในการออกแบบเบื้องต้นได้ออกแบบ สำหรับการเชื่อมต่อกับเครื่องโมเด็ม (MODEM) ซึ่งเป็นอุปกรณ์ที่ใช้ในการสื่อสารข้อมูลระหว่างคอมพิวเตอร์ผ่านทางโทรศัพท์ ซึ่งทำให้อัตราการรับส่งข้อมูลถูกจำกัด ให้มีค่าที่ค่อนข้างต่ำ มาตรฐาน RS-232C นี้ได้ออกแบบให้มีโครงสร้างการสื่อสารเป็นแบบจุดต่อจุดเท่านั้น โดยมีลักษณะสมบัติทางไฟฟ้าและทางกายภาพ ดังแสดงในตารางที่ 5 และรูป ที่ 3.9



รูปที่ 3.9 แสดงโครงสร้างของการสื่อสารข้อมูลแบบอนุกรมตามมาตรฐาน RS-232C

-การสื่อสารข้อมูลแบบอนุกรมตามมาตรฐาน RS-422A

ในการออกแบบระบบสื่อสารข้อมูลจากที่ผ่านมาจนถึงปัจจุบัน ได้มีการพยายามที่จะออกแบบให้การสื่อสารข้อมูลได้รวดเร็วขึ้นและมีระยะในการสื่อสารที่มากขึ้นด้วย ซึ่งที่ผ่านการสื่อสารข้อมูลตามมาตรฐาน RS-232C ได้ออกแบบเพื่อใช้เชื่อมต่อกับโมเด็มเท่านั้น จึงไม่ได้คำนึงถึงความรวดเร็วและระยะทางในการสื่อสารต่อมาได้มีมาตรฐานในการสื่อสารข้อมูลใหม่ที่ได้รับออกแบบมาเพื่อรองรับความต้องการของผู้ใช้งานที่ต้องการให้การรับส่งข้อมูลได้ไกลและรวดเร็ว มาตรฐานนี้คือ RS-422A ซึ่งการที่มาตรฐานสื่อสารนี้สามารถรับส่งข้อมูลได้ระยะทางไกลและรวดเร็วขึ้น เนื่องจากหลักการที่ใช้สัญญาณเป็นแบบดิฟเฟอเรนเชียลดังแสดงในรูปที่ 3-10 ซึ่งหลักการก็คือสัญญาณทุกสัญญาณจะเปรียบเทียบกับระหว่างสัญญาณ 2 เส้นเทียบกับมาตรฐาน RS-232C ที่สัญญาณทุกสัญญาณจะเทียบกับกราวด์ ซึ่งในการสื่อสารในระยะทางไกล ๆ แล้วสัญญาณจะถูกลดทอนไปและเมื่อสัญญาณถูกลดทอนถึงจุดๆหนึ่งสัญญาณนั้นก็จะมีผลขาดไปจากความเป็นจริง ทำให้การรับส่งข้อมูลเกิดผิดพลาดขึ้น แต่สำหรับสัญญาณแบบดิฟเฟอเรนเชียลลดทอนของสัญญาณก็จะไปลดทอนทั้งสองสาย ด้วยค่าที่เท่ากันหรือใกล้เคียงกัน และความแตกต่างของสัญญาณทั้ง 2 เส้น จากตัวส่งไปยังตัวรับก็ยังเท่าเดิมหรือเปลี่ยนแปลงน้อย จึงทำให้ผลของการลดทอนต่อสัญญาณที่ระยะการสื่อสารที่ไกลไม่มีผลต่อการสื่อสารข้อมูล พร้อมทั้งสามารถติดต่อกับตัวรับได้ถึง 10 ตัว ดังแสดงตารางในตารางที่ 3-1 การเปรียบเทียบมาตรฐานการสื่อสารข้อมูล



รูปที่ 3.10 แสดงโครงสร้างของการสื่อสารข้อมูลแบบอนุกรมตามมาตรฐาน RS-422A

- การสื่อสารข้อมูลแบบอนุกรมตามมาตรฐาน RS-485

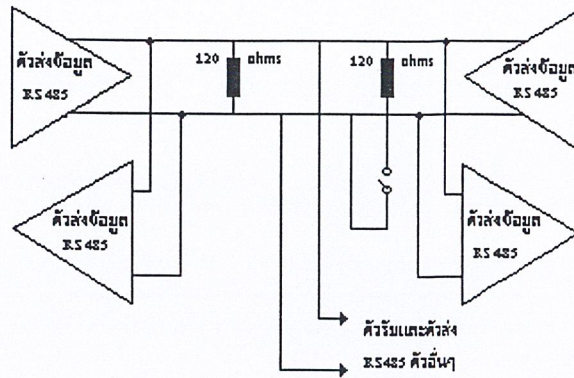
การสื่อสารข้อมูลตามมาตรฐานที่กล่าวมาข้างต้นคือ RS-232C นั้นเป็นมาตรฐานการสื่อสารข้อมูลในแบบที่สื่อสารระหว่างอุปกรณ์ต่ออุปกรณ์ หรือ จุดต่อจุด (Point-to-Point) ส่วน RS-422A นั้นเป็นมาตรฐานที่พัฒนามาจาก RS-232 C ให้ได้ระยะทางไกลขึ้นและอัตราการสื่อสารเพิ่มขึ้น แต่ก็ยังเป็นการสื่อสารข้อมูลจากอุปกรณ์หนึ่งตัวไปยังอุปกรณ์อื่นๆ ได้สูงสุด 10 ตัวเท่านั้น ไม่สามารถส่งย้อนกลับจากอุปกรณ์ตัวรับมาตัวส่งได้ หรือ กล่าวได้ว่า การสื่อสารข้อมูลตามมาตรฐาน RS-422A นั้นเป็นการสื่อสารข้อมูลแบบ Simplex คือทิศทางของข้อมูลเป็นแบบทางเดียวตลอดเวลา ดังนั้นถ้าต้องการ ออกแบบระบบให้เป็นลักษณะโครงข่ายข้อมูลก็จะไม่สามารถทำได้ จึงมีการพัฒนามาตรฐาน การสื่อสารข้อมูลขึ้นใหม่เพื่อรองรับความต้องการนี้ คือ มาตรฐาน RS-485 ซึ่งเป็นมาตรฐานที่อาศัยหลักการของสัญญาณแบบดิฟเฟอเรนเชียล เช่นเดียวกับมาตรฐาน RS-422A แต่สามารถสื่อสารข้อมูลได้ทั้ง 2 ทิศทางในสายสัญญาณเพียงคู่เดียว ซึ่งก็คือการสื่อสารข้อมูลแบบ Half – Duplex จากผลของการใช้สัญญาณในลักษณะดิฟเฟอเรนเชียลนี้ ทำให้ระยะทางและความเร็วในการสื่อสารข้อมูลมีค่าสูง เช่นเดียวกับมาตรฐานการสื่อสารข้อมูล RS-422A แต่มาตรฐาน RS-485 สามารถที่จะสื่อสารระหว่างอุปกรณ์ทั้งการส่งของอุปกรณ์ได้สูงสุด 32 ตัว หรืออาจกล่าวได้ว่าการสื่อสารตามมาตรฐาน RS-485 เป็นการสื่อสารแบบหลายจุด (Multipoint-Communication) ดังแสดงค่าเปรียบเทียบในตารางที่ 3-1 และแสดงโครงสร้างในรูปที่ 3-11

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3.1 แสดงค่าเปรียบเทียบคุณสมบัติทางไฟฟ้าของ RS-232 C , RS – 422A และ RS-485

พารามิเตอร์	RS-232-C	RS-422-A	RS-485
โหมดการทำงาน	Single-ended	Differential	Differential
จำนวนของตัวรับ และส่งที่ยอมรับได้	1 ตัวส่ง 1 ตัวรับ	1 ตัวส่ง 10 ตัวรับ	32 ตัวส่ง 32 ตัวรับ
ความยาวของคู่สาย สูงสุด	50 ฟุต	4000 ฟุต	4000 ฟุต
อัตราการส่งข้อมูล สูงสุด	20k bps	10 M bps	10 M bps
Maximum common Mode voltage	2.5 V	6 V -2.5 V	12 V -7 V
Driver output	5 V ต่ำสุด 15 V สูงสุด	2 V ต่ำสุด	1.5 V ต่ำสุด
Driver load (ohm)	3 K ถึง 7 K	100 ต่ำสุด	60 ต่ำสุด
Driver slew rate	30 V/us สูงสุด	NA	NA
กระแสลิมิต เมื่อ เอาต์พุตลัดวงจร	500 mA ลัดวงจร กับ Vcc หรือ GND	150 mA ลัดวงจร กับ GND	150 mA ลัดกับ GND 250 mA ลัดกับ 12 V
ค่าความต้านทาน เอาต์พุตของตัวส่ง (โอห์ม)	NA (Power on) 300 (Power off)	NA (Power on) 60k (Power off)	120k(Power on) 120k(Power off)
ค่าความต้านทานอิน พุทตัวรับ	3k ถึง 7k ohm	4k ohm	12k ohm
ความไวของตัวรับ	3V	200 mV	200 mV

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.11 แสดง โครงสร้างของการสื่อสารข้อมูลแบบอนุกรมตามมาตรฐาน RS-485

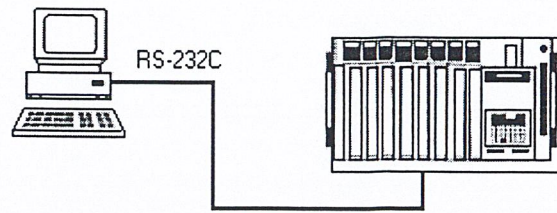
3.2 รูปแบบการสื่อสารข้อมูลของเครื่องควบคุมแบบตรรกะที่โปรแกรมได้

การต่อวงจรในการติดต่อสื่อสารข้อมูลของเครื่องควบคุม PLC/PC อาจแบ่งออกได้ 3 แบบ ด้วยกันคือ

1. การต่อวงจรแบบจุดต่อจุด (Point to Point)
2. การต่อวงจรแบบหลายจุด (Multi point)
3. การต่อวงจรแบบเครือข่ายแลน (LAN : Local Area Network)

3.2.1 การต่อวงจรแบบจุดต่อจุด

ในการติดต่อสื่อสารข้อมูลแบบจุดต่อจุด จะใช้มาตรฐานการสื่อสารแบบ RS 232C ซึ่งเป็นการติดต่อในระยะทางที่ไม่ไกลมาก เช่น การติดต่อระหว่างเครื่องควบคุม PLC/PC กับเครื่องคอมพิวเตอร์ เพื่อทำการตรวจสอบสถานะการทำงาน หรือ การควบคุมจากคอมพิวเตอร์ หรือ สนับสนุนการทำงานอื่นๆ การติดต่อระหว่างเครื่องควบคุม PLC/PC กับอุปกรณ์อื่น เช่น โมเด็ม (MODEM) อุปกรณ์ควบคุมสัมผัสหน้าจอ (Touch Screen) คีย์บอร์ด (Key board) เครื่องอ่านรหัส แถบ (bar-code Reader) เครื่องพิมพ์ เครื่องบันทึก (Recorder) เครื่องรายงานผล (Reporter) เป็นต้น

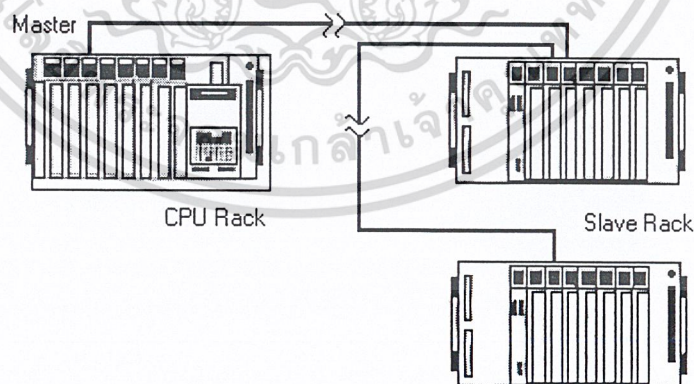


รูปที่ 3.12 ระบบการติดต่อแบบจุดต่อจุด

3.2.2 การต่อวงจรแบบหลายจุด (Multi point)

-การติดต่อแบบอินพุทและเอาต์พุทระยะไกล (Remote I/O System)

เป็นการเชื่อมโยงสื่อสารข้อมูลในลักษณะที่ตัวประมวลผลของเครื่องควบคุมหลัก กับ ปลายทางอุปกรณ์ที่ถูกควบคุมอยู่ไกลกัน และอาจมีมากกว่า 1 จุด ดังนั้นจึงมีความจำเป็นที่จะมีการเชื่อมโยงในระยะไกล และอาจมีมากถึง 1 จุด ดังนั้นจึงมีความจำเป็นที่จะมีการเชื่อมโยงในระยะไกล ในการติดต่อลักษณะนี้จะใช้มาตรฐานการสื่อสารแบบ RS-485 บนตัวนำสองเส้น โดยที่ตัวควบคุมหลัก (Master) จะทำการรับอินพุททั้งจากพื้นที่ของตัวควบคุม และหน่วยระยะไกล (Slave) จากนั้น จะทำการประมวลผล แล้วส่งเอาต์พุทไปยังส่วนต่างๆ โดยที่อินพุทและเอาต์พุทของหน่วยระยะไกล จะจองพื้นที่ของรีเลย์ภายใน (Internal Relay) ของตัวควบคุมหลัก โดยการเซทระบบนั้นจะแตกต่างกันไปตามบริษัทผู้ผลิต



รูปที่ 3.13 การเชื่อมโยงแบบอินพุทและเอาต์พุทระยะไกล (Remote I/O System)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Input Area
Output Area
Internal Relay Area

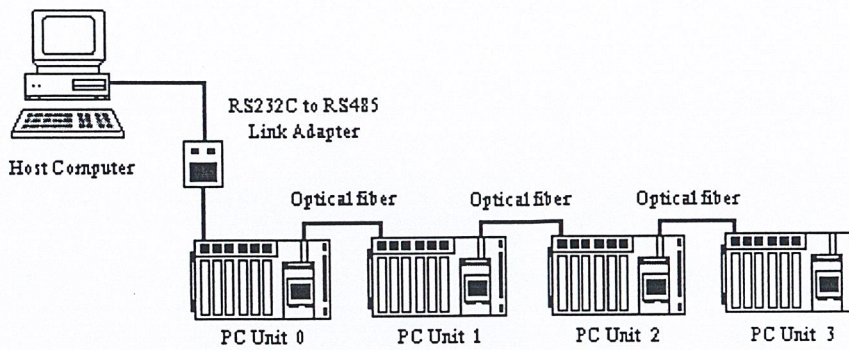
รูปที่ 3.14 แสดงการจำลองพื้นที่ของตัวควบคุมหลักขณะทำงานปกติ

Input Area
Output Area
Internal Relay Area
Input Remote
Output Remote

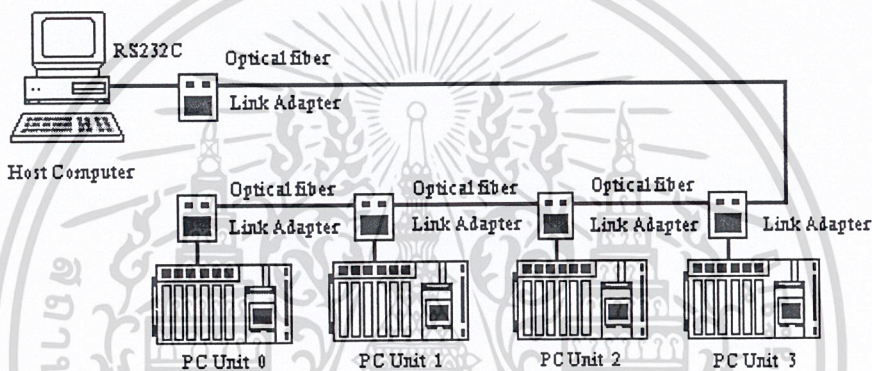
รูปที่ 3.15 แสดงการจำลองพื้นที่ของตัวควบคุมหลักขณะเมื่อมีการทำงานร่วมกับหน่วยอินพุทเอาต์พุทระยะไกล

3.2.2.2 การติดต่อแบบพีซีลิงค์ (PC Link System)

เป็นการเชื่อมโยงสื่อสารข้อมูลในลักษณะของการแบ่งพื้นที่ในการอ่านเขียนของ PLC/PC แต่ละตัว เครื่องควบคุม PLC/PC แต่ละตัวสามารถรับรู้ข่าวสาร ซึ่งกันและกันได้ มาตรฐานในการสื่อสารข้อมูลชนิดนี้มักจะเป็นแบบ RS-422A หรือเป็นรูปแบบการสื่อสารผ่านสายใยแก้วนำแสง (Fiber optic) หรือผ่านสายโคแอกเซียล โดยจะมีตัว Link Adapter เป็นตัวแปลงรูปแบบข้อมูลให้เป็นมาตรฐานตรงกัน อาจจะมีการเชื่อมโยงกับเครื่องคอมพิวเตอร์ เข้ามาในระบบการลิงค์ เพื่อสนับสนุนการทำงานของเครื่องควบคุม PLC/PC ซึ่งจะเรียกว่าโฮสคอมพิวเตอร์ (Host Computer) มีลักษณะการต่อวงจรดังนี้

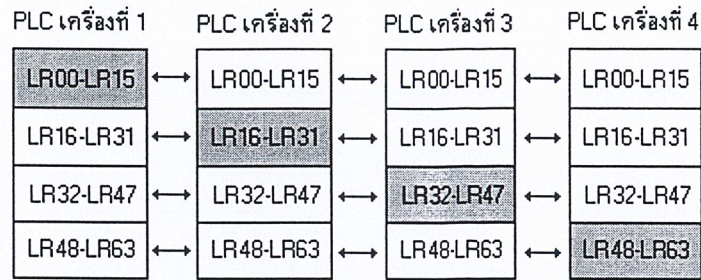


รูปที่ 3.16 การต่อแบบ Serial link



รูปที่ 3.17 การต่อแบบ Parallel link

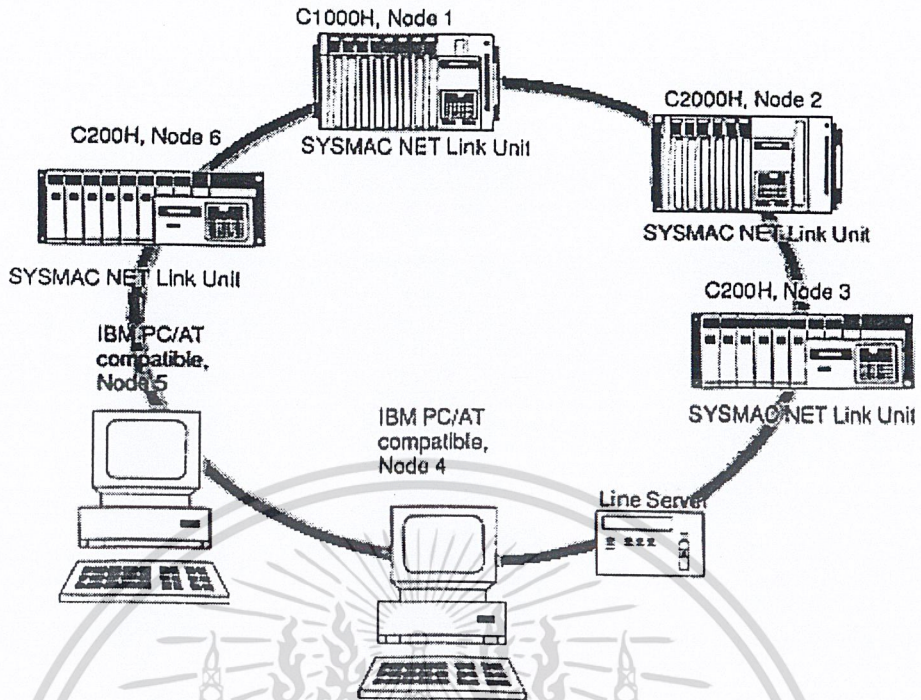
ตัวอย่างการจัดแบ่งพื้นที่ของเครื่องควบคุม PLC/PC กำหนดให้มีเครื่องควบคุมจำนวน 4 ชุด เชื่อมโยงในเครื่องควบคุมและแต่ละเครื่องจะมีพื้นที่ของรีเลย์ที่ทำหน้าที่ในการเชื่อมโยงข้อมูล (Link - Relay) ถ้ามีเครื่องควบคุมที่เชื่อมต่อในระบบมากพื้นที่ก็จะถูกแบ่งออกไปตามจำนวน เช่น 4 ชุด พื้นที่จะถูกแบ่งออกเป็น 4 ส่วน เช่นเครื่องควบคุมมีพื้นที่ของ รีเลย์จำนวน 64 Word ก็จะถูกแบ่งเป็นส่วนละ 16 Word ส่วนแรกจะเป็นพื้นที่ของเครื่องควบคุมตัวที่ 1 ส่วนที่สองจะเป็นของเครื่องควบคุม ตัวที่ 2 และ 3 ตามลำดับ โดยพื้นที่ส่วนอื่นจะถูกกั้นไว้ไม่ให้อ่านเขียนข้อมูลลงไปได้ (Read Only) ยกเว้นพื้นที่ของตัวเอง ดังนั้นถ้า PLC/PC เครื่องที่ 1 ต้องการรับข้อมูลของ PLC เครื่องที่ 3 ก็สามารถอ่านได้จากพื้นที่ของเครื่องควบคุมส่วนที่สาม และถ้าต้องการส่งข้อมูลให้ก็เขียนลงในพื้นที่ของส่วนแรกแล้ว PLC ตัวที่ 3 จะทำการอ่านข้อมูลในส่วนแรก ก็จะสามารทราบข้อมูลของเครื่องควบคุม PLC ตัวที่ 1



รูปที่ 3.18 การแบ่งพื้นที่สำหรับ PC Link System

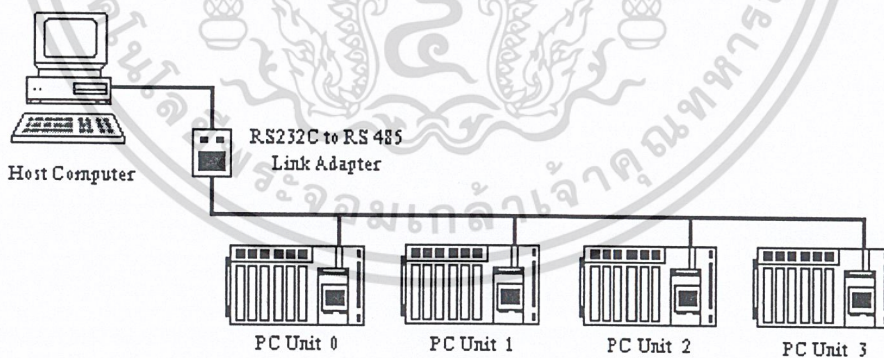
3.2.3 การต่อวงจรสื่อสารแบบเครือข่ายท้องถิ่น (Local Area Network for PLC/PC)

การเชื่อมโยงในลักษณะของเครือข่ายท้องถิ่นนี้เป็นระบบขนาดใหญ่ประกอบด้วยอุปกรณ์หลายชนิดและหลายวงรอบด้วยกัน ซึ่งจะเป็นลักษณะเครือข่ายแบบวงแหวน (Ring Network) ในแต่ละวงรอบ สามารถที่จะเชื่อมโยงไปยังวงรอบอื่นๆ ได้ ซึ่งจะมีความซับซ้อนในระบบการติดต่อสื่อสารมากขึ้น รูปแบบสายนำสัญญาณจะมีทั้งสายใยแก้วนำแสง (Fiber optic) และสายตัวนำในแบบต่าง ๆ ในระบบโครงข่ายก็จะมีเครื่องคอมพิวเตอร์ส่วนบุคคลทำหน้าที่เป็นตัวจัดการข้อมูล เช่นเดียวกับกับเครื่องเซิร์ฟเวอร์ (Computer Server) บนโครงข่ายของคอมพิวเตอร์



รูปที่ 3.19 ระบบการติดต่อสื่อสารแบบเครือข่ายท้องถิ่น

สำหรับระบบ โครงข่ายที่ได้ออกแบบในวิทยานิพนธ์นี้เป็นการเชื่อมโยงแบบหลายจุดที่ใช้มาตรฐานสื่อสารแบบ RS 485 โดยมีโฮสคอมพิวเตอร์เป็นตัวจัดการข้อมูลต่างๆ ให้กับเครื่องควบคุมที่เชื่อมต่อในระบบโครงข่ายดังรูปต่อไปนี้ รายละเอียดจะได้กล่าวในบทที่ 5

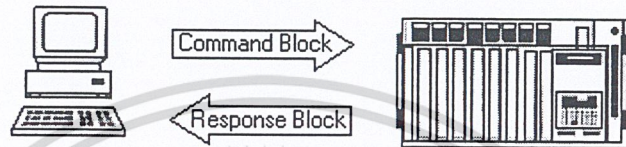


รูปที่ 3.20 แสดงระบบโครงข่ายแบบหลายจุดโดยใช้มาตรฐานการสื่อสาร RS 485

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3 ข้อตกลงในการสื่อสารของเครื่องควบคุม PLC/PC

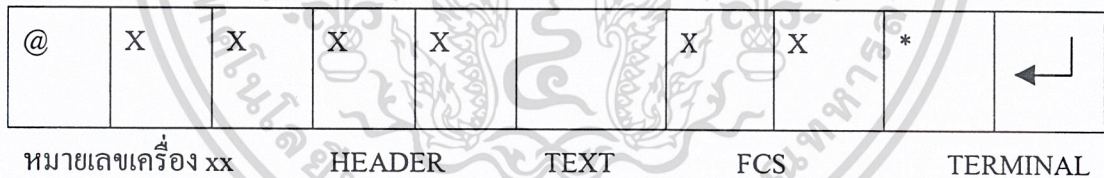
รูปแบบของข้อตกลงในการสื่อสารของเครื่องควบคุม PLC/PC โดยทั่วไปจะเป็นลักษณะการถามตอบกันระหว่างเครื่องควบคุมกับอุปกรณ์ภายนอก ซึ่งอุปกรณ์ภายนอกมักจะเป็นฝ่ายถามก่อน โดยการส่งบล็อกคำสั่ง (Command Block) ออกไป จากนั้นเครื่องควบคุมจะทำการตรวจสอบแล้วส่งบล็อกตอบสนองกลับมา (Response Block) ตัวอย่างการสื่อสารข้อมูลกันระหว่างเครื่องคอมพิวเตอร์กับเครื่องควบคุม PLC/PC



รูปที่ 3.21 แสดงการติดต่อระหว่างคอมพิวเตอร์กับเครื่องควบคุม PLC/PC

3.3.1 รูปแบบของบล็อก (Block Format)

ลักษณะของบล็อกของเครื่องควบคุม PLC/PC แต่ละผู้ผลิตจะแตกต่างกันไปแต่จะมีพื้นฐานเดียวกันดังตัวอย่างของบล็อกคำสั่งต่อไปนี้



รูปที่ 3.22 แสดงรูปแบบของบล็อก

1. หมายเลขเครื่อง ในการเชื่อมต่อ ที่เป็นโครงข่ายแบบหลายจุดนั้นเครื่องควบที่เชื่อมต่ออยู่ในระบบจะมีมากกว่า 1 เครื่อง การกำหนดว่าต้องการส่งข้อมูลให้กับเครื่องควบคุมตัวใด เช่น @ 05 คือ เครื่องหมายเลขที่ 05
2. ส่วน HEADER เป็นส่วนของคำสั่งหลัก ที่จะกำหนดว่าต้องการกระทำกับข้อมูลส่วนใด เช่น ต้องการอ่านข้อมูลของ อินพุท ต้องการเขียนข้อมูลให้กับตัวนับ เป็นต้น
3. ส่วน TEXT เป็นส่วนของข้อมูล เช่น ค่าที่อ่านได้จากอินพุท หรือ ค่าที่จะต้องเขียนลงในพื้นที่ต่ำลง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

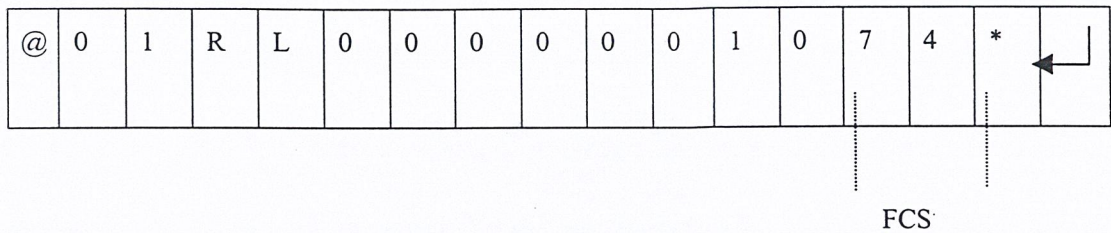
4. ส่วนของ FCS เป็นส่วนของการควบคุมความผิดพลาดของข้อมูลซึ่งได้จากการคำนวณ
5. ส่วนของ TERMINAL เป็นส่วนที่ปิดท้ายบอกให้ทราบว่าจบบล็อก และมักจะติดตามด้วยรหัสของ Carrier Return (CR)

3.3.2 การคำนวณ FCS

FCS หรือ Frame Check Sequence เป็นสิ่งที่ใช้ในการตรวจสอบความผิดพลาดของการสื่อสาร เป็นการเปลี่ยนข้อมูล 8 บิต เป็น 2 ตัวอักษรข้อมูล ASCII แล้วนำมาทำการ Exclusive – Or (XOR) โดยเริ่มจาก @ จนถึงตัวอักษรตัวสุดท้ายของ TEXT การคำนวณ FCS สามารถจะใช้วิธีอื่นนอกจากจะใช้วิธี XOR ดังที่ได้กล่าวไว้แล้วในหัวข้อที่ 3.1.3 เรื่อง การควบคุมความผิดพลาด

อักขระ	รหัส(ASCII BINARY)		XOR(HEX)
@	0100 0000	[40]	
0	0011 0000	[30]	112[70]
1	0011 0001	[31]	65[41]
R	0101 0010	[52]	19[13]
L	0100 1100	[4C]	95[5F]
0	0011 0000	[30]	111[6F]
0	0011 0000	[30]	95[5F]
0	0011 0000	[30]	111[6F]
0	0011 0000	[30]	95[5F]
0	0011 0000	[30]	111[6F]
0	0011 0000	[30]	95[5F]
1	0011 0001	[31]	111[6F]
0	0011 0000	[30]	94[5F]
*	0010 1010	[2A]	116[74]

จากการคำนวณที่ได้จะสร้างเป็นบล็อกได้ดังนี้



รูปที่ 2.23 แสดงรูปแบบของบล็อกและการคำนวณ FCS

ในบทนี้แสดงให้เห็นถึงรูปแบบในการสื่อสารข้อมูลของคอมพิวเตอร์ เปรียบเทียบกับการติดต่อสื่อสารของเครื่องควบคุม PLC/PC ซึ่งจะเป็แนวทางในการพัฒนาการติดต่อสื่อสารข้อมูลของเครื่องควบคุม PLC/PC เพื่อให้เครื่องควบคุม PLC/PC สามารถควบคุมเครื่องจักรในระบบที่มีขนาดใหญ่ ให้สามารถทำงานได้อย่างสัมพันธ์กัน ซึ่งจะทำให้การควบคุมมีประสิทธิภาพมากยิ่งขึ้น



การออกแบบและพัฒนาเครื่องควบคุมแบบตรรกะที่โปรแกรมได้

บทนี้จะกล่าวถึงลักษณะของโครงสร้างแต่ละส่วนของเครื่องควบคุมแบบตรรกะที่โปรแกรมได้ที่ได้ออกแบบมา เครื่องควบคุมแบบตรรกะที่โปรแกรมได้เป็นระบบคอมพิวเตอร์ขนาดเล็กชนิดหนึ่ง ที่ได้ออกแบบมาเพื่อใช้งานควบคุมเฉพาะที่ ประกอบด้วยส่วนประมวลผลกลาง ส่วนของหน่วยความจำ และทางเข้าออกของข้อมูล หรืออินพุท/เอาต์พุท ซึ่งอุปกรณ์ต่าง ๆ จะต้องมี ความเชื่อถือได้สูง เนื่องจากส่วนประมวลผลโดยทั่วไปของคอมพิวเตอร์ มีลักษณะการทำงานเป็นแบบตรรกะ สามารถที่จะทำงานในลักษณะต่าง ๆ ได้ทุกรูปแบบตามแต่ผู้ใช้งานจะโปรแกรมให้มีการทำงานในลักษณะใด ทั้งนี้จะต้องขึ้นอยู่กับความสามารถของระบบปฏิบัติการที่ออกแบบด้วย ซึ่งจะได้กล่าวต่อไป และด้วยความสามารถประมวลผลดังกล่าว จึงเป็นที่มาในการออกแบบเครื่อง ควบคุมแบบตรรกะที่โปรแกรมได้

4.1 คุณสมบัติของหน่วยประมวลผลกลางและหน่วยความจำ

ในการออกแบบเครื่องควบคุมแบบตรรกะที่โปรแกรมได้ เราจะต้องมีการพิจารณาคือ ความสามารถของหน่วยประมวลผล และอุปกรณ์ต่อรวม เช่น พอร์ต หน่วยสื่อสารข้อมูล หน่วย แสดงผล และรับข้อมูลจากผู้ใช้ ว่าสถาปัตยกรรมของไมโครโปรเซสเซอร์เบอร์ใดที่เหมาะสมที่สุด เมื่อพิจารณาสถาปัตยกรรมของไมโครโปรเซสเซอร์ AT89C52 ซึ่งเห็นว่ามีเหมาะสมมากที่สุด ทั้งระบบฮาร์ดแวร์ และซอฟต์แวร์ จึงได้นำมาเป็นหน่วยประมวลผลของเครื่องควบคุมแบบตรรกะที่ โปรแกรมได้ ซึ่งมีรายละเอียดดังนี้

4.1.1 ทางด้านฮาร์ดแวร์

เป็นไมโครโปรเซสเซอร์ที่ประมวลผลคราวละ 8 บิต ทำงานที่ความถี่สูงสุด 24 MHz มี หน่วยความจำแบบ Flash ขนาด 8 Kbytes ที่สามารถลบและเขียนใหม่ได้ถึง 1000 ครั้ง และยังมี หน่วยความจำชั่วคราวภายใน (RAM) ขนาด 128 x 8 บิต ซึ่งทำให้ไม่ต้องใช้หน่วยความจำภายนอก มากนัก ส่วนของพอร์ตก็มีให้ใช้งาน อยู่ถึง 3 พอร์ต ขนาด 8 บิต และยังมีส่วนของตัวนับเวลาให้ใช้ ได้อีกถึง 3 ตัวด้วยกัน ทำให้การออกแบบการใช้งานนั้นจะไม่ค่อยยุ่งยากมากนัก และที่สำคัญคือมี ขนาดเล็ก

4.1.2 ทางด้านซอฟต์แวร์

ชุดคำสั่งของไมโครโปรเซสเซอร์ ไม่ว่าจะเป็นชุดคำสั่งแบบตรรกะ เช่น AND OR XOR COMPLEMENT ชุดคำสั่งในการคำนวณเชิงตัวเลข เช่น SUB ADD MUL DIV ชุดคำสั่งของ การเคลื่อนย้ายข้อมูลทั้งแบบ 8 บิต และ 16 บิต สามารถนำมาใช้ในการประยุกต์โปรแกรมได้ง่าย

เมื่อเราสามารถกำหนดส่วนของหน่วยประมวลผล และสามารถกำหนดชุดคำสั่งที่ใช้ในการโปรแกรมแล้ว ก็จัดการในส่วนของการจัดการหน่วยความจำ การสื่อสารข้อมูลกับอุปกรณ์ร่วม และการออกแบบระบบปฏิบัติการ ให้สะดวกในการใช้งาน

4.2 คุณสมบัติของหน่วยการรับส่งข้อมูล

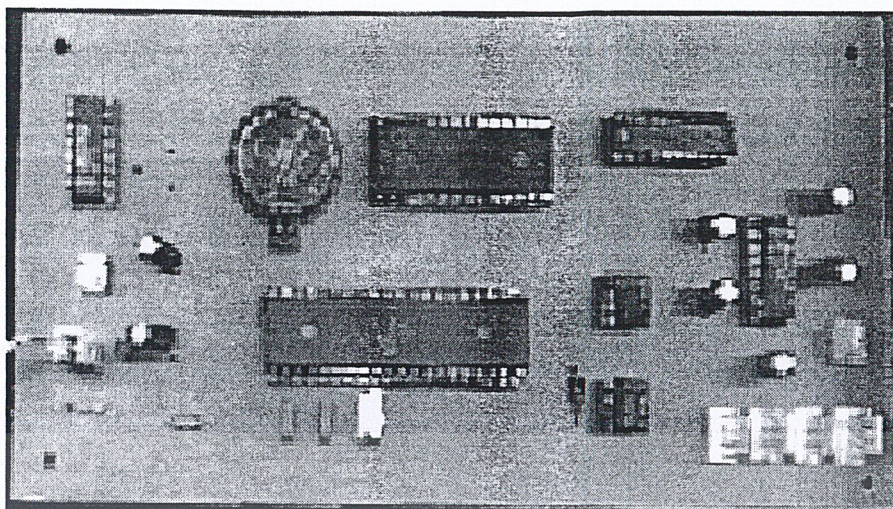
ในการรับส่งข้อมูล จะแบ่งออกเป็น 2 ส่วน คือ จะเป็นการเชื่อมต่อจากพอร์ตของคอมพิวเตอร์ ไปยังบอร์ดของหน่วยประมวลผล และการเชื่อมต่อจากบอร์ดของหน่วยประมวลผล ไปยังส่วนของบอร์ดหน่วยอินพุท/เอาต์พุท

4.2.1 การเชื่อมต่อจากพอร์ตของคอมพิวเตอร์ ไปยังบอร์ดของหน่วยประมวลผล

การเชื่อมต่อในส่วนนี้จะเป็นการเชื่อมต่อจากพอร์ตอนุกรมของคอมพิวเตอร์ ผ่านทางพอร์ต DB-9 โดยจะมีการส่งสัญญาณออกมาเพียง 3 เส้น คือ Tx Rx GND เพื่อใช้ในการถ่ายเท (Upload) ข้อมูลจากคอมพิวเตอร์ โดยที่สัญญาณ Tx และ Rx จะเข้าสู่ ขา Tx และ Rx ของวงจรรวมไมโครโปรเซสเซอร์ AT89C52 ผ่านทางวงจรรวม MAX-232 เพื่อเป็นการเปลี่ยนจาก ขนาดแรงดัน 12 V ให้อยู่ในระดับแรงดัน 5 V

4.2.2 การเชื่อมต่อจากบอร์ดของหน่วยประมวลผล ไปยังหน่วยอินพุท/เอาต์พุท

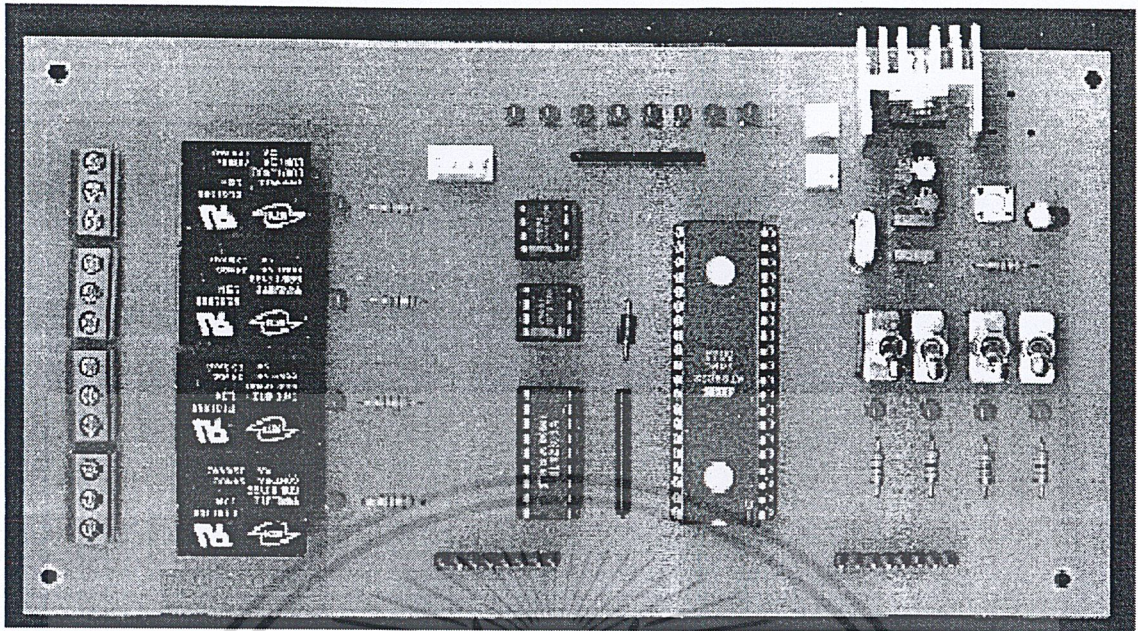
การเชื่อมต่อในส่วนนี้จะเป็นการเชื่อมต่อจากบอร์ดของหน่วยประมวลผล โดยจะเป็นสัญญาณที่ออกมาทาง ขา Tx และ Rx ของไมโครโปรเซสเซอร์ AT89C52 ผ่านทางวงจรรวมเบอร์ 75176 ซึ่งจะเป็นส่วนของการเปลี่ยนแปลงระดับของแรงดันในการส่งข้อมูลให้เป็นแบบแรงดันแตกต่าง หรือจากมาตรฐาน RS-232 เป็น RS-422 ซึ่งเป็นส่วนของการขยายระยะทางของการส่งสัญญาณ ทำให้การติดต่อระหว่างหน่วยประมวลผลกับหน่วยอินพุท/เอาต์พุท สามารถทำได้ไกลขึ้น หรือที่เรียกว่า Remote Input/Output



รูปที่ 4.1 บอร์ดหน่วยประมวลผลกลาง หน่วยความจำ และส่วนของการรับส่งข้อมูล

4.3 คุณสมบัติของหน่วยอินพุท/เอาต์พุท

ในส่วนของหน่วยอินพุท/เอาต์พุทนั้น จะประกอบด้วยวงจรรวมซึ่งใช้ในการอ่านรหัส ที่หน่วยประมวลผลส่งมา และกระทำกระบวนการตามรหัส ที่ได้รับ และส่วนที่เป็นพอร์ตของอินพุทและเอาต์พุท รวมไปถึงหน่วยของการรับส่งข้อมูลและการแปลงข้อมูลที่ได้จากหน่วยประมวลผลกลาง โดยในส่วนของวงจรรวมควบคุมการทำงานต่างๆ จะเป็นเบอร์ AT89C52 หน่วยอินพุทจะอยู่ในรูปแบบของสวิตช์ และหน่วยเอาต์พุทจะอยู่ในรูปแบบของรีเลย์ ส่วนของการเชื่อมต่อข้อมูลจากหน่วยประมวลผลจะใช้วงจรรวมเบอร์ 75176 เป็นตัวแปลงมาตรฐานสัญญาณ จาก RS-422 ให้อยู่ในรูปแบบ RS-232



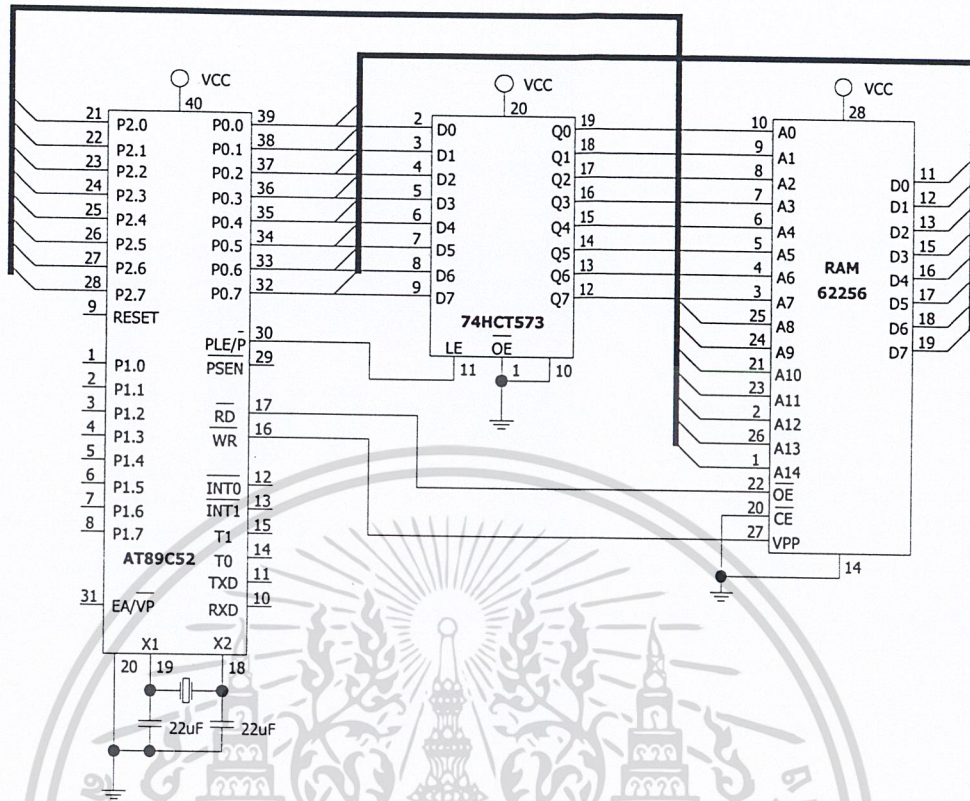
รูปที่ 4.2 บอร์ดหน่วยอินพุท/เอาต์พุทระยะไกล

4.4 การออกแบบวงจร

วงจรที่ออกแบบ ถูกแบ่งออกเป็นส่วนต่าง ๆ ได้ดังนี้

4.4.1 วงจรหน่วยประมวลผลกลางและหน่วยความจำ

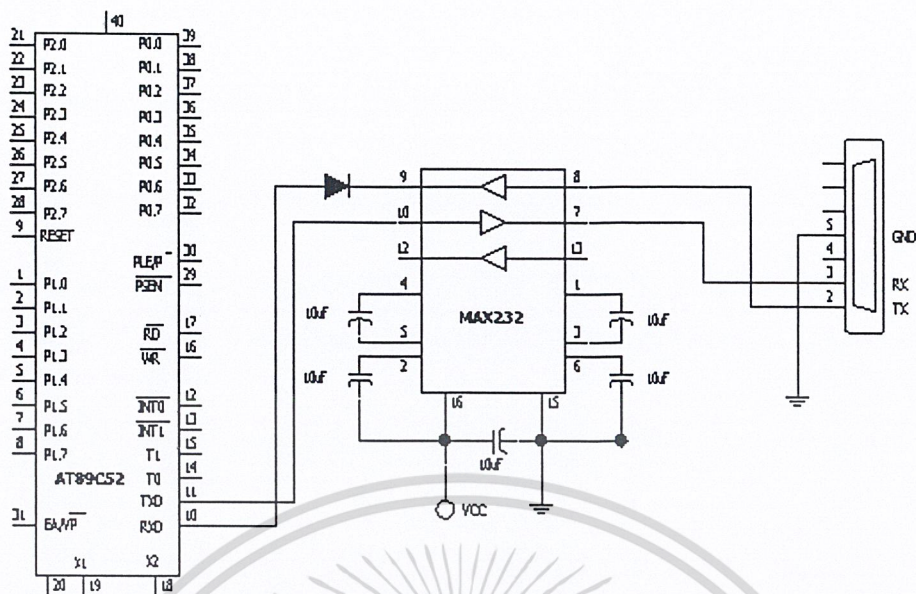
เราจะใช้ไมโครโปรเซสเซอร์ เบอร์ AT89C52 เป็นหน่วยประมวลผล ขนาด 8 บิต มีหน่วยความจำภายในที่เป็นแบบ Flash ขนาด 8 Kbyte และหน่วยความจำชั่วคราวภายในอีก 128 บิต ต่อร่วมกับหน่วยความจำชั่วคราวภายนอกขนาด 32 Kbytes ซึ่งอยู่ในรูปของวงจรรวมเบอร์ 62256 ซึ่งจะต้องมีการ Latch ข้อมูล โดยอาศัยวงจรรวมเบอร์ 74LS373 เป็นตัวช่วย ในการส่งข้อมูลที่มีขา A0-A12 เป็นขา Address และขา Data 8 เส้น โดยมีการควบคุมการส่งข้อมูลโดยขา ALE ขา PSEN เป็นส่วนของการควบคุมการส่งผ่านข้อมูลจากหน่วยประมวลผลไปสู่หน่วยความจำ



รูปที่ 4.3 แสดงวงจรสำหรับการเชื่อมต่อหน่วยประมวลผลและหน่วยความจำ

4.4.2 วงจรเชื่อมต่อระหว่างคอมพิวเตอร์และหน่วยประมวลผล

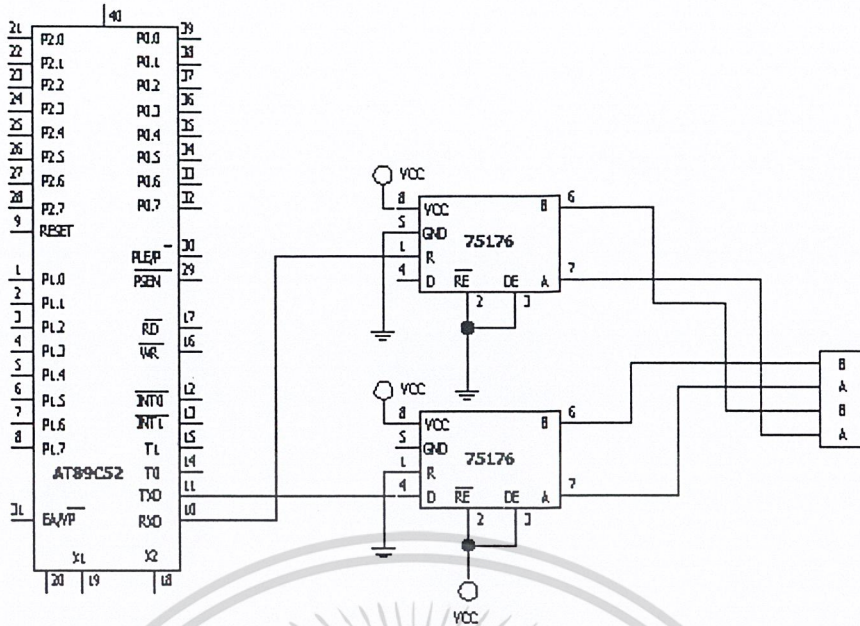
การส่งข้อมูลจากคอมพิวเตอร์ไปยังบอร์ดของหน่วยประมวลผล จะต้องมีการแปลงของระดับสัญญาณที่ทำการส่งข้อมูล เนื่องจากระดับสัญญาณที่ออกมาจากคอมพิวเตอร์จะเป็น 12 V แต่ส่วนที่จะเข้ามายังไมโครโปรเซสเซอร์จะเป็น 5 V จึงต้องมีวงจรรวมเบอร์ MAX-232 เป็นตัวแปลงระดับสัญญาณ จากสัญญาณ Tx Rx ของคอมพิวเตอร์ที่พอร์ตอนุกรม DB-9 ไปสู่ ขาสัญญาณ Tx Rx ของไมโครโปรเซสเซอร์



รูปที่ 4.4 แสดงวงจรสำหรับเชื่อมต่อจากคอมพิวเตอร์ไปยังบอร์ดประมวลผล

4.4.3 วงจรเชื่อมต่อระหว่างบอร์ดหน่วยประมวลผลและบอร์ดหน่วยอินพุท/เอาต์พุท

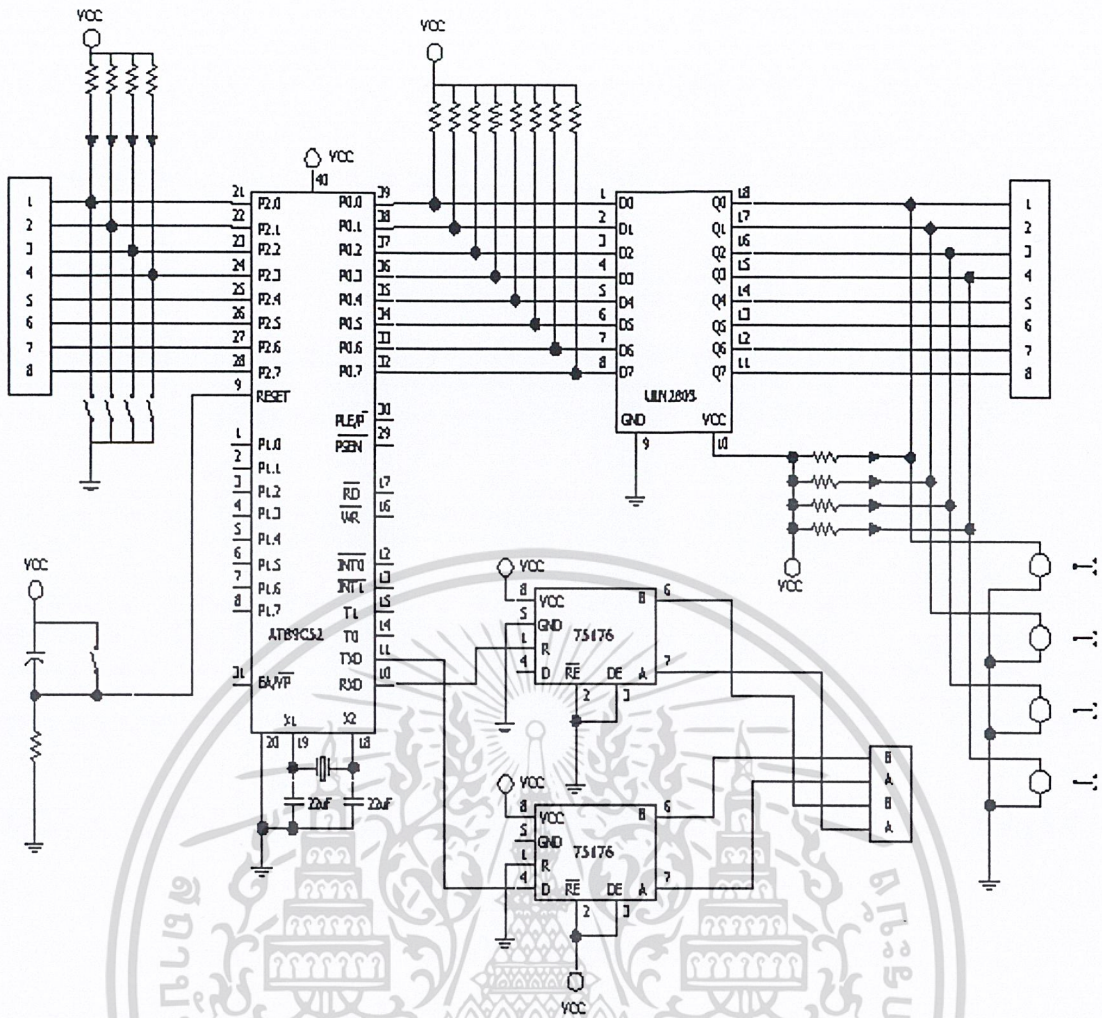
ในส่วนของการเชื่อมต่อสัญญาณนั้น ถ้าระยะการส่งที่มากกว่า 50 ฟุต จะใช้การสื่อสารตามมาตรฐาน RS-232 ไม่ได้ เนื่องจากข้อจำกัดของระดับสัญญาณ จึงต้องมีการเปลี่ยนแปลงมาตรฐานของการส่งข้อมูลใหม่ ซึ่งต้องใช้การส่งที่เป็นแบบแรงดันแตกต่างกัน และยังมีผลให้ความเร็วในการส่งเพิ่มขึ้นด้วย โดยจะมีการใช้วงจรรวมเบอร์ 75176 เป็นส่วนของการแปลงสัญญาณ ซึ่งตามมาตรฐานการส่งนี้จะใช้สายนำสัญญาณจำนวน 4 เส้น ซึ่งจะแบ่งออกเป็นการส่งและการรับข้อมูลอย่างละ 2 เส้น ซึ่งจะทำให้การส่งข้อมูลเป็นไปอย่างสะดวกและรวดเร็ว



รูปที่ 4.5 แสดงวงจรการเชื่อมต่อและการแปลงสัญญาณในการสื่อสารจาก RS-232 เป็น RS-422

4.4.4 วงจรติดต่อภายนอกของบอร์ดอินพุท/เอาต์พุท

การติดต่อรับค่าที่เป็นอินพุทนั้น จะใช้สวิทช์ ในการกำหนดสถานะของทางด้านอินพุท ผ่านทางพอร์ต 1 ของไมโครโปรเซสเซอร์ ซึ่งสามารถกำหนดได้เป็น 8 จุด แต่ทางผู้ออกแบบทำให้เหลือเพียง 4 จุดที่เป็นสวิทช์เท่านั้น ส่วนทางด้านเอาต์พุทก็จะเป็นการส่งข้อมูลออกมาทางพอร์ต 3 ของไมโครโปรเซสเซอร์ เข้าสู่วงจรรวมวงจรรวมเบอร์ ULN2803 เพื่อทำการยกระดับสัญญาณให้เป็น 12 V เพื่อที่จะใช้เป็นแรงดันขับวงจรรีเลย์ที่เป็นส่วนของเอาต์พุท โดยที่จะแสดงออกเพียง 4 จุดเท่านั้น และยังมีส่วนของชุดแสดงการทดสอบการส่งโปรแกรมด้วย



รูปที่ 4.6 แสดงวงจรของหน่วยอินพุท/เอาต์พุท

4.5 ระบบปฏิบัติการ

ระบบปฏิบัติการ สำหรับเครื่องควบคุมแบบตรรกะที่โปรแกรมได้นี้ จะหมายถึงระบบที่ใช้ควบคุมการทำงานทั้งหมด มีหน้าที่รับข้อมูลที่ส่งมาจากคอมพิวเตอร์ นำรหัสปฏิบัติการที่ได้มาเป็นการประมวลผล ทำงานตามฟังก์ชัน ควบคุมการรับส่งข้อมูลไปยังหน่วยอินพุท/เอาต์พุท โดยที่จะแยกการพิจารณาได้ดังนี้

1. การแบ่งพื้นที่หน่วยความจำ
2. โปรแกรมผู้ใช้และคำสั่งสำหรับเครื่องควบคุม
3. การจัดข้อมูลของโปรแกรมผู้ใช้
4. การแปลชุดคำสั่งจากโปรแกรมผู้ใช้เป็นภาษาเครื่อง
5. การทำงานของฟังก์ชันควบคุม

4.5.1 การแบ่งพื้นที่หน่วยความจำ

หน่วยความจำของระบบปฏิบัติการมีจำนวน 64 Kbytes นั้น จะถูกแบ่งออกเป็นส่วน ๆ ดัง

นี้

ตารางที่ 4.1 การแบ่งหน่วยความจำ

32 Kbyte R only	โปรแกรมบริหารงานระบบ
	โปรแกรมควบคุม
24 Kbyte R / W	โปรแกรมผู้ใช้
8 Kbyte R / W	ตารางข้อมูล และ รีจิสเตอร์

1. โปรแกรมบริหารงานระบบ ถูกกำหนดให้เป็นโปรแกรมจัดวางระบบข้อมูล โปรแกรมที่ผู้ใช้ทำการแปล
2. โปรแกรมควบคุม เป็นพื้นที่หน่วยความจำของ โปรแกรมที่ใช้ในการประมวลผลฟังก์ชันควบคุมที่ผู้ใช้ทำการ โปรแกรม
3. โปรแกรมผู้ใช้ เป็นส่วนของการเก็บข้อมูล ที่ผู้ใช้ทำการป้อนข้อมูลเข้ามา เพื่อเป็นกระบวนการควบคุม
4. ตารางข้อมูล และรีจิสเตอร์ เป็นส่วนของหน่วยความจำที่มีการเปลี่ยนแปลงแก้ไขได้ ภายในตารางข้อมูล จะถูกเก็บข้อมูลที่มีลักษณะ “1” หรือ “0” ซึ่งใช้แทนสภาวะ “ON” และ “OFF” ทางไฟฟ้าของข้อมูลทางค่านินพุท และเอาต์พุท ข้อมูลของรีเลย์ภายใน ข้อมูลในแต่ละพื้นที่ได้ออกแบบไว้ดังนี้

ตารางอินพุท/เอาต์พุท และตารางรีเลย์ภายใน ถูกกำหนดขึ้นเป็นจำนวน 48 ตำแหน่ง โดยเริ่มจาก 00-47 โดยแบ่งออกได้ดังนี้

ตารางที่ 4.2 แสดงตารางอินพุท/เอาต์พุทและรีเลย์ภายใน

ตำแหน่ง	หน้าที่
X000-X016	สำหรับเก็บข้อมูลอินพุท
Y000-Y016	สำหรับใช้เป็นข้อมูลของเอาต์พุท
TR00-TR16	สำหรับเป็นรีเลย์ภายในหรือรีจิสเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.5.2 โปรแกรมผู้ใช้และคำสั่งสำหรับเครื่องควบคุม

คำสั่งที่ใช้กันโดยทั่วไปของเครื่องควบคุมแบบตรรก จะอยู่ในรูปแบบของภาษาบูลีน ที่มีการสื่อความหมายทางตรรกที่ตรงไปตรงมากับผู้ใช้ได้เป็นอย่างดี เช่นสมการบูลีนต่อไปนี้ สามารถที่จะแสดงคำสั่งบูลีนได้ดังนี้

$$X=[(A*B)+(C*D)]*F$$

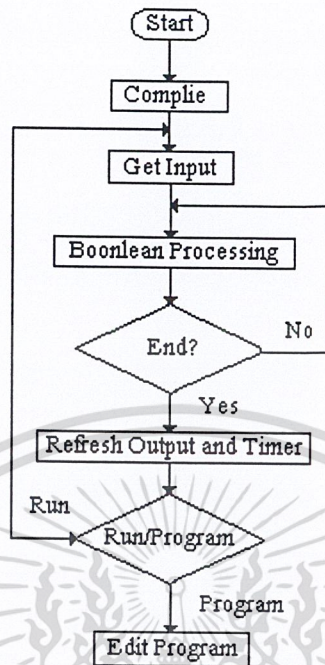
ตารางที่ 4.3 แสดงความสัมพันธ์ของสมการบูลีนกับชุดคำสั่งบูลีน

ลำดับ	คำสั่ง	อินพุท/เอาต์พุท	ความหมาย
0000	LD	A	นำค่าสถานะ A
0001	AND	B	กระทำการ AND กับ B
0002	LD	NOT C	นำค่าสถานะตรงข้าม C
0003	AND	D	กระทำการ AND กับ D
0004	ORLD		กระทำการ OR ทั้งสองวงเล็บ
0005	AND	F	กระทำการ AND กับ F
0006	OUT	X	ส่งผลลัพธ์ออกที่ X
0007	END		จบสิ้นคำสั่ง (กลับไปทำใหม่)

คำสั่ง END เป็นสิ่งที่จะบ่งบอกว่าจะต้องกระโดดไปกระทำลำดับที่เริ่มต้นใหม่ และกระทำเรื่อยไปจนกว่าจะมีคำสั่งให้หยุดการทำงาน ลักษณะของการทำงานจะเป็นการทำงานที่เป็นวงรอบ โดยมาสารที่จะรับค่าสถานะที่มีการเปลี่ยนแปลงในตัวแปรอินพุทได้เกือบจะต่อเนื่องตรงเท่าที่เวลาในการวนรอบ (Scan Time) จะเอื้ออำนวย เพื่อที่จะนำมาประมวลผลและให้ผลลัพธ์ออกไปควบคุม โดยที่เวลาในการวนรอบจะอยู่ที่ความสามารถของหน่วยประมวลผลและการทำงานของระบบการจัดการ ซึ่งพอที่จะแสดงเป็นผังงานได้ดังนี้

1. รับค่าสถานะจากภายนอกทางอินพุท
2. นำค่าที่ได้ไปเก็บไว้ในหน่วยความจำ
3. ปฏิบัติการตามคำสั่งบูลีน
4. นำค่าที่ได้จากการปฏิบัติการไปใส่ในหน่วยความจำ
5. ส่งผลลัพธ์ออกทางเอาต์พุท

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.7 แสดงผังภาพการทำงานในวงรอบ

4.5.3 การจัดข้อมูลของโปรแกรมผู้ใช้

ข้อมูลของ โปรแกรมผู้ใช้ จะมีลักษณะเป็นชุดคำสั่ง แต่ละชุดคำสั่งจะถูกจัดเก็บเป็นเรคคอร์ด ซึ่งภายในประกอบด้วย Code และ Operand

ตารางที่ 4.4 แสดงการเก็บข้อมูลของโปรแกรมผู้ใช้ในหน่วยความจำ

Record NO.	Byte	Word
RECORD 1	CODE	DATA
RECORD 2	CODE	DATA
:	:	:
RECORD END	CODE	DATA

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เรคคอร์ดของชุดคำสั่งจะต้องทำการเก็บรหัสชุดคำสั่ง และข้อมูลของตำแหน่งตารางข้อมูล ส่วนฟิลด์ Code ใช้กำหนดรหัสของคำสั่งที่ผู้ใช้ทำการ โปรแกรมขึ้นเพื่อให้เครื่องควบคุมทำงานตามฟังก์ชันการควบคุม คำสั่งที่สร้างขึ้นมามีทั้งหมด ** คำสั่ง ดังรายละเอียดต่อไปนี้

ตารางที่ 4.5 แสดงคำสั่งที่ได้ออกแบบไว้

	คำสั่ง	ความหมาย
00	LD XX	นำค่าจากสถานะจากตารางข้อมูลอินพุตมาเป็นผลลัพธ์
01	LD YY	นำค่าจากสถานะจากตารางข้อมูลเอาต์พุตมาเป็นผลลัพธ์
02	LD NOT XX	นำค่าจากสถานะตรงข้ามจากตารางข้อมูลอินพุตมาเป็นผลลัพธ์
03	LD NOT YY	นำค่าจากสถานะตรงข้ามจากตารางข้อมูลเอาต์พุตมาเป็นผลลัพธ์
04	OR XX	กระทำลอจิก OR กับสถานะที่นำจากตารางข้อมูลอินพุต
05	OR YY	กระทำลอจิก OR กับสถานะที่นำจากตารางข้อมูลเอาต์พุต
06	OR NOT XX	กระทำลอจิก OR กับสถานะตรงข้ามที่นำจากตารางข้อมูลอินพุต
07	OR NOT YY	กระทำลอจิก OR กับสถานะตรงข้ามที่นำจากตารางข้อมูลเอาต์พุต
08	AND XX	กระทำลอจิก AND กับสถานะที่นำจากตารางข้อมูลอินพุต
09	AND YY	กระทำลอจิก AND กับสถานะที่นำจากตารางข้อมูลเอาต์พุต
0A	AND NOT XX	กระทำลอจิก AND กับสถานะตรงข้ามที่นำจากตารางข้อมูลอินพุต
0B	AND NOT YY	กระทำลอจิก AND กับสถานะตรงข้ามที่นำจากตารางข้อมูลเอาต์พุต
0C	OUT YY	นำสถานะทางลอจิกไปเก็บไว้ในข้อมูล
0D	OUT NOT YY	นำสถานะทางลอจิกตรงข้ามไปเก็บไว้ในข้อมูล
0E	TIM	เรียกใช้ตัวตั้งเวลา
0F	CNT	เรียกใช้ตัวนับ
10	END	สิ้นสุดคำสั่งให้ทำรอบใหม่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในส่วนของหน่วยอินพุตและเอาต์พุตนั้น จะถูกจองพื้นที่ในหน่วยความจำ โดยจะมีรหัส ในการอ้างถึงหน่วยความจำที่ตำแหน่งดังกล่าวซึ่งจะแบ่งเป็น

ตารางที่ 4.6 แสดงส่วนของหน่วยความจำที่เก็บค่าอินพุต เอาต์พุต และรีเลย์ภายใน

รหัส	ความหมาย
X000-X016	หน่วยอินพุต
Y000-Y016	หน่วยเอาต์พุต
TR00-TR16	รีเลย์ภายใน
T00-T01	ตัวตั้งเวลา
C00	ตัวนับ

เมื่อผู้ใช้โปรแกรมคำสั่งบูตลิน จะได้รับรหัสที่ได้นำไปเก็บไว้ในหน่วยความจำ ในพื้นที่ โปรแกรมผู้ใช้ดังนี้

ตารางที่ 4.7 การจัดเก็บรหัสจากโปรแกรมคำสั่งบูตลิน

Address	Instruction	Operand	CODE	ADDRESS
0000	LD	X00	00	20
0001	AND	X01	08	21
0002	OR NOT	Y00	07	40
0003	OUT	Y00	0C	40
0004	END		10	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

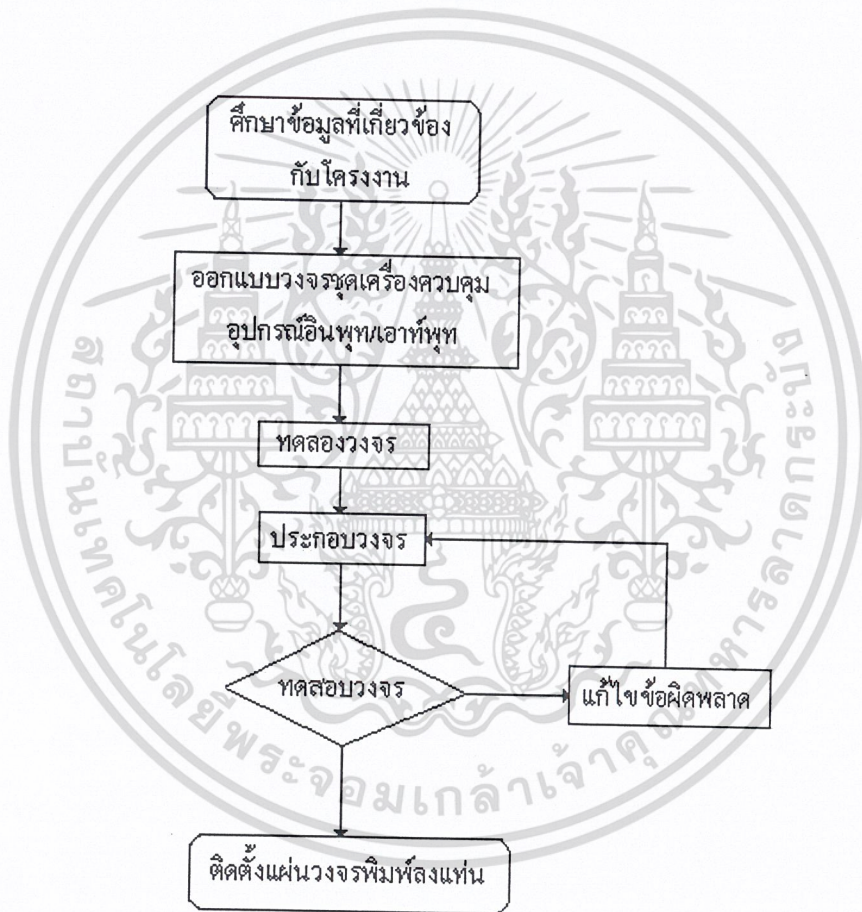
บทที่ 5

ขั้นตอนและวิธีการดำเนินงาน

โครงการ เครื่องควบคุมแบบตรรกที่โปรแกรมได้นี้ สามารถแบ่งขั้นตอนและวิธีดำเนินงานออกเป็น 2 ส่วน คือ ส่วน Hardware และ Software

5.1 ส่วน Hardware

ขั้นตอนในส่วนของ Hardware เราสามารถเขียนเป็นผังงาน ได้ดังนี้



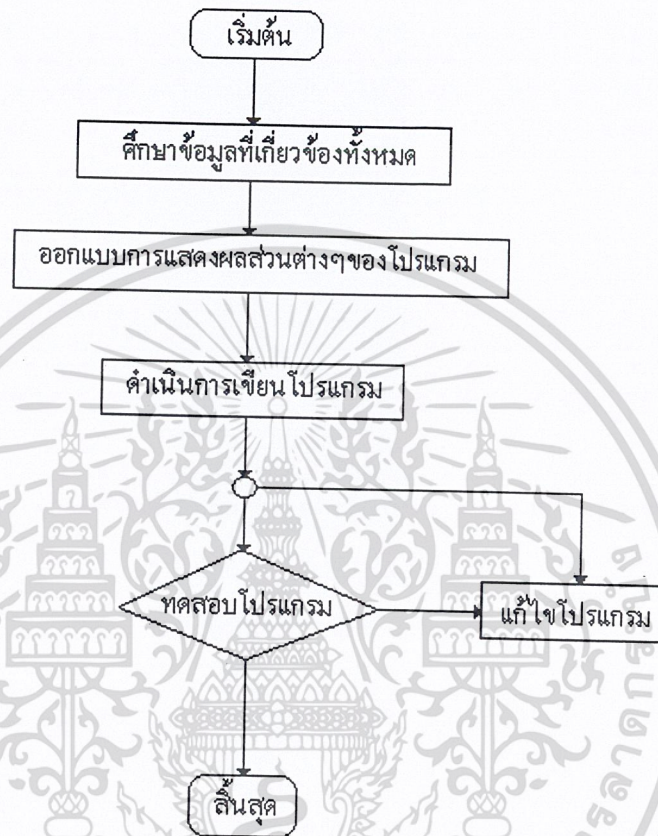
รูปที่ 5.1 ผังการดำเนินงานส่วน Hardware

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.2 ส่วน Software

5.2.1 ขั้นตอนการดำเนินงาน

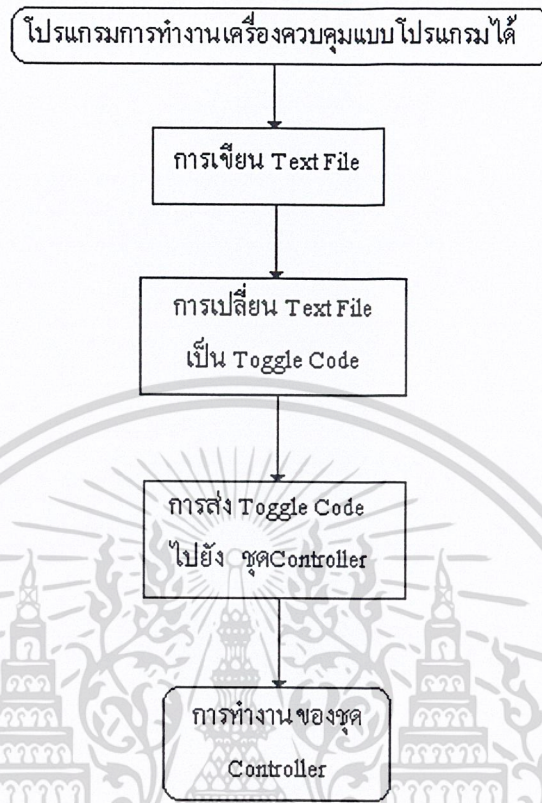
ขั้นตอนดำเนินงาน เป็นการแสดงลำดับขั้นตอนการดำเนินงานส่วน Software เครื่องควบคุมแบบโปรแกรมคังรูป



รูปที่ 5.2 แสดง Flow Chart การดำเนินงานส่วน Software

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.2.2 ส่วนประกอบของโปรแกรมในส่วน Software



รูปที่ 5.3 แสดง Block Diagram ส่วนของ Software

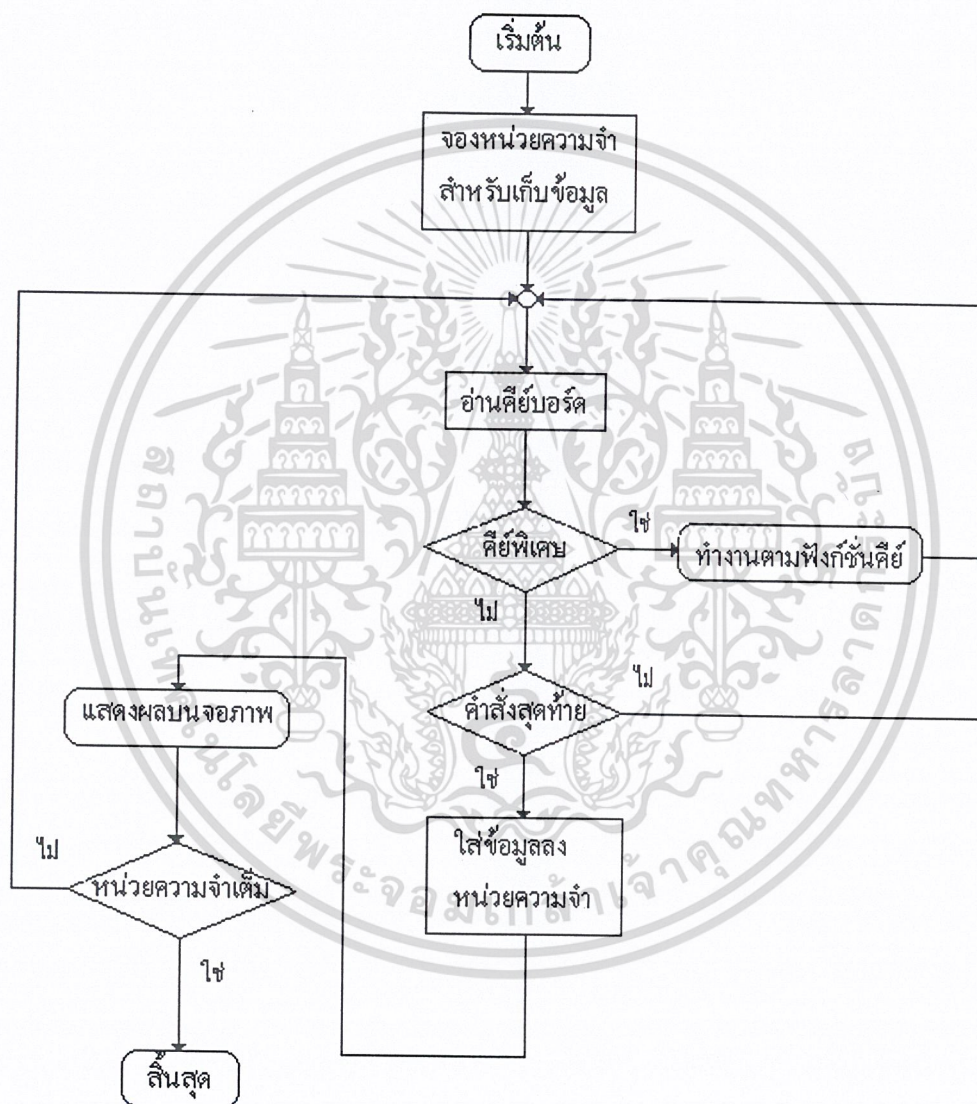
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.3 การทำงาน

โปรแกรมจำลองการทำงานเครื่องควบคุมแบบโปรแกรมได้ สามารถที่จะแบ่งเป็นส่วนต่าง ๆ ดังนี้ คือ

5.3.1 ส่วนการสร้าง Text File

การทำงานของส่วนการสร้าง Text File แสดงได้ดังรูป



รูปที่ 5.4 ส่วนของการสร้าง Text File

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การสร้าง Text File มีขั้นตอนการทำงานดังนี้

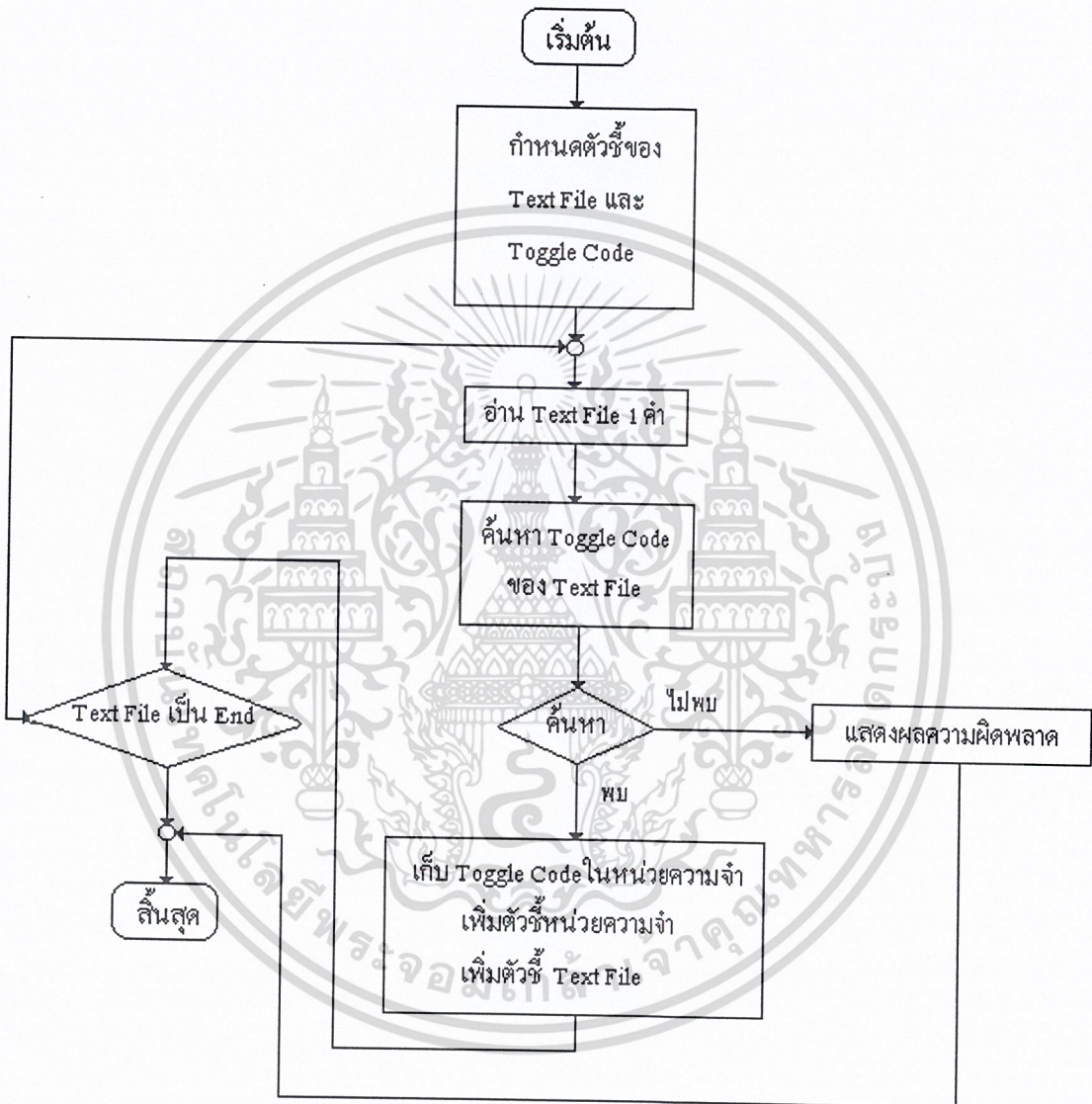
1. การจองหน่วยความจำไว้สำหรับเก็บข้อมูลที่เป็น Text File และกำหนดตัวชี้ตำแหน่งของข้อมูลที่จะจองไว้ในหน่วยความจำ
2. การอ่านข้อมูลจาก Key Board
3. ตรวจสอบว่าเป็นฟังก์ชันคีย์พิเศษหรือไม่ ถ้าใช่จะทำโปรแกรมของฟังก์ชันคีย์พิเศษ
4. ตรวจสอบว่าเป็นคำสั่งสุดท้ายหรือไม่ ถ้าไม่ใช่ให้กลับไปอ่านค่าคีย์บอร์ดตัวต่อไป
5. แสดงผลของข้อมูลที่อยู่ในหน่วยความจำ
6. ตรวจสอบหน่วยความจำเต็มหรือไม่ ถ้ายังจะกลับไปอ่านข้อมูลตัวใหม่ แต่ถ้าเต็มจะออกจากการเขียนโปรแกรมทันที



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.3.2 การเปลี่ยนรหัสที่เป็น Text File เป็นรหัส Toggle Code

การเปลี่ยนข้อมูลจากรหัสคำสั่งที่เป็น Text File เป็นรหัส Toggle Code มีขั้นตอนการทำงานดังรูป



รูปที่ 5.5 Flow Chart ของการเปลี่ยน Text File เป็น Toggle Code

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การเปลี่ยนข้อมูลจากรหัสคำสั่งที่เป็น Text File เป็นรหัส Toggle Code มีขั้นตอนการทำงานดังนี้

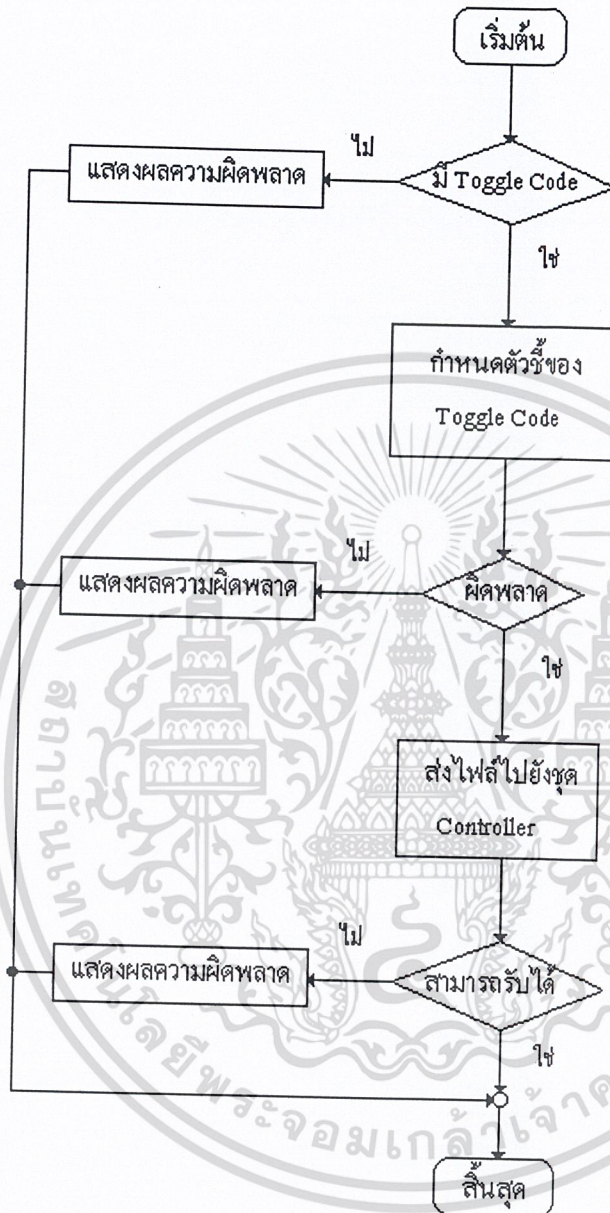
1. จองหน่วยความจำที่จะทำการรหัส Toggle Code และ กำหนดตัวชี้สำหรับ Text File และ Toggle Code
2. อ่านค่าตัวอักษรของ Text File จำนวน 1 คำ
3. นำค่าที่มาจาก Text File มาเปรียบเทียบกับข้อมูลของโปรแกรมเพื่อให้ได้ Toggle Code
4. ในกรณีที่ไม่สามารถเปลี่ยน Text File เป็น Toggle Code จะแสดงข้อผิดพลาดแล้วออกจากโปรแกรม
5. ในกรณีที่สามารถเปลี่ยน Text File เป็น Toggle Code ได้ จะทำการเก็บ Toggle Code ไว้ในหน่วยความจำ
6. ตรวจสอบคำสั่งของ Text File ถ้าเป็น END จะออกจากโปรแกรม แต่ถ้าไม่ใช่ จะทำการเลื่อนตัวชี้ของ Text File และ Toggle Code ไปยังตำแหน่งถัดไปแล้วทำตามขั้นตอน 2 ใหม่



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.3.3 การส่ง Toggle Code ไปยังชุด Controller

การส่ง Toggle Code ไปยังชุด Controller มีขั้นตอนดังนี้



รูปที่ 5.6 Flow Chart ของโปรแกรมการส่ง Toggle Code ไปยังชุด Controller

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การส่ง Toggle Code ไปยังชุด Controller มีขั้นตอนการทำงานดังนี้

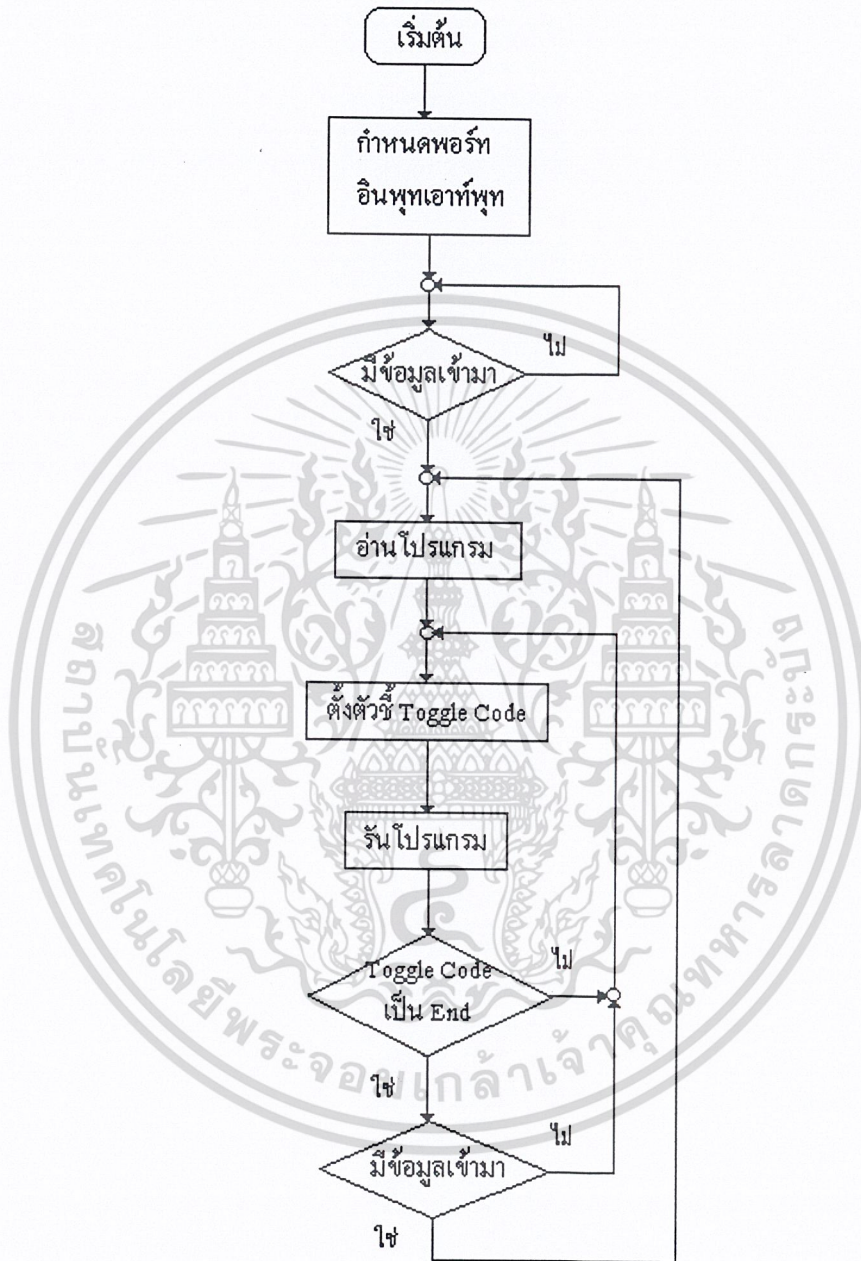
1. ตรวจสอบมี Toggle Code อยู่ในหน่วยความจำหรือไม่ ถ้ามี จะกำหนดตัวชี้ Toggle Code ขึ้น แต่ถ้าไม่แสดงข้อความผิดพลาดและออกจากโปรแกรม
2. ตรวจสอบความผิดพลาดที่อาจจะเกิดขึ้น
3. ส่ง Toggle Code ไปยังชุด Controller โดยใช้การส่งข้อมูล แบบอนุกรมตามมาตรฐาน RS 232 ความเร็วในการส่ง 9600 b/s
4. ตรวจสอบชุด Controller ว่าได้รับได้หรือไม่ ถ้ารับได้ จะออกจากโปรแกรม แต่ถ้ารับไม่ได้จะแสดงข้อความผิดพลาดและออกจากโปรแกรม



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.3.4 การทำงานของโปรแกรมหลักของชุด Controller

การทำงานของโปรแกรมหลักของชุด Controller เป็นการนำ Toggle Code ที่ได้รับจากเครื่องคอมพิวเตอร์มาทำงานตามรหัสของ Toggle Code ขั้นตอนการทำงานแสดงได้ดังนี้



รูปที่ 5.7 Flow Chart แสดงส่วนของโปรแกรมชุด Controller

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทำงานของโปรแกรมในชุด Contoller เป็นการนำ Toggle Code ซึ่งคือ Code โปรแกรมมาทำงาน มีขั้นตอนดังนี้

1. ทำการ Set CPU ให้มีการทำงานตามที่โปรแกรมต้องการและกำหนดตัวชี้สำหรับ Toggle Code, Stack
2. ตรวจสอบว่ามีข้อมูลเข้ามาหรือไม่ ถ้ามีก็ให้ทำการอ่านข้อมูลเข้ามา แต่ถ้าไม่ก็ทำการวนรอต่อไป
3. อ่านข้อมูลที่ส่งมาจากคอมพิวเตอร์
4. กำหนดตัวชี้สำหรับ Toggle Code
5. ทำงานตาม Toggle Code ที่อยู่ในหน่วยความจำ
6. ตรวจสอบค่าของ Toggle Code เป็น END หรือไม่ ถ้าใช่ให้ไปข้อ 7 แต่ถ้าไม่ใช่ไปข้อ 4
7. ตรวจสอบว่ามีคำสั่งข้อมูลใหม่มาจากคอมพิวเตอร์หรือไม่ ถ้ามีการส่งข้อมูลมาใหม่ไปยังข้อ 3 แต่ถ้าไม่มีการส่งข้อมูลมาใหม่ไปข้อ 4



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 6

การออกแบบโปรแกรมสนับสนุนการทำงานของเครื่องควบคุม PLC

โปรแกรมสนับสนุนการทำงานของเครื่องควบคุมแบบตรรกะที่โปรแกรมได้ ที่ออกแบบขึ้นมาี้ ทำหน้าที่จัดการข้อมูลการทำงานในส่วนของการทำงาน การตรวจสอบการทำงาน โปรแกรม รวมถึงการจัดการส่งข้อมูลไปยังหน่วยประมวลผลหลักของตัวเครื่องควบคุมหลัก โดยอาศัยการสื่อสาร (Protocol) ที่ได้ออกแบบมา ทำงานภายใต้ระบบการจัดการของ Microsoft Windows ซึ่งใช้สร้างบนโปรแกรม Visual Basic โดยจะกล่าวถึงการนำข้อตกลงของการสื่อสาร มาประยุกต์ใช้งานในการส่งข้อมูล และโครงสร้างรวมไปถึงการออกแบบโปรแกรมในแต่ละส่วน

6.1 คุณสมบัติทั่วไปของโปรแกรมสนับสนุนการทำงาน

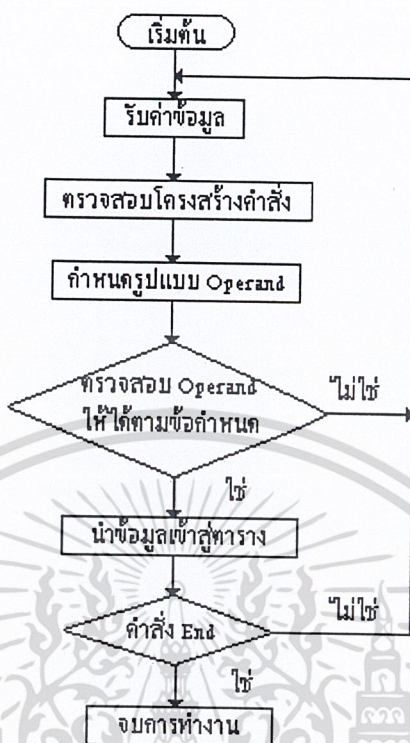
1. เป็นโปรแกรมที่ใช้ในการเขียนหรือแก้ไขชุดคำสั่งบูลีน (Boolean Editor) ตามรูปแบบของเครื่องควบคุมที่ได้ออกแบบมา
2. สามารถนำโปรแกรมชุดคำสั่งบูลีน ที่ได้ทำการเขียนหรือนำมาแก้ไข จัดเก็บลงในแฟ้มข้อมูล (Data File) หรือโหลดมาใช้งาน โดยผ่านทางโปรแกรม Microsoft Access
3. สามารถนำโปรแกรมชุดคำสั่งบูลีน ที่ทำการเขียนหรือแก้ไขแล้ว หรือเปิดจากแฟ้มข้อมูล เพื่อถ่ายเท (Upload) จากเครื่องคอมพิวเตอร์ ไปยังเครื่องควบคุม PLC ที่ต่ออยู่ในระบบได้

6.2 การออกแบบโปรแกรมเพื่อป้อนโปรแกรมคำสั่งบูลีน

ในการป้อนโปรแกรมภาษาบูลีน ส่วนประกอบของการทำงานดังนี้

1. ส่วนของการรับค่าของข้อมูล
2. ส่วนของการตรวจสอบโครงสร้างของแต่ละคำสั่ง
3. ส่วนของการตรวจสอบการใช้ตำแหน่งหน่วยความจำ
4. ส่วนของการเก็บค่าเข้าสู่ตารางข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

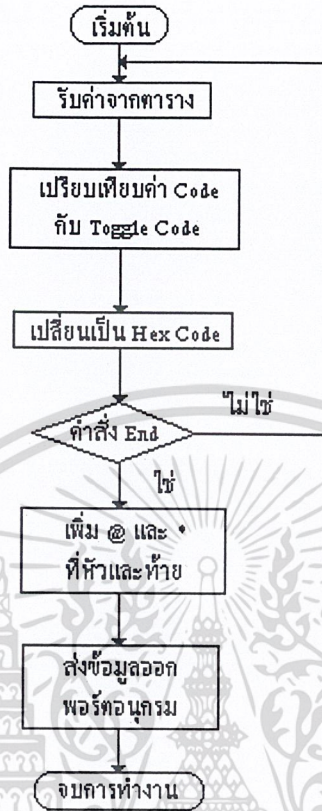


รูปที่ 6.1 แสดงผังของการป้อน โปรแกรมบูตลิน

6.3 การออกแบบโปรแกรมเพื่อถ่ายเทข้อมูลจากเครื่องคอมพิวเตอร์ไปยังเครื่องควบคุม

ในการถ่ายเทข้อมูลจากภาษาบูตลินที่ได้จากตาราง จะต้องมีส่วนของการแปลงให้เป็นรหัส ASCII ก่อนจึงจะทำการส่งข้อมูลได้ โดยที่จะมีขั้นตอนการทำงานคือ

1. อ่านค่าที่ได้จากตาราง มาแปรเปลี่ยนเป็นค่าของรหัส ASCII เพื่อส่งออกเป็นรหัสของเลขฐาน 16 (Hex Code) ที่หน่วยประมวลผลของเครื่องควบคุมเข้าใจ
2. เพิ่มรหัสขึ้นต้นและรหัสปิดท้ายให้กับข้อมูลโดยที่มี "@" เป็นรหัสขึ้นต้น และ "*" เป็นรหัสปิดท้าย
3. ส่งข้อมูลจากคอมพิวเตอร์ไปยังเครื่องควบคุม ผ่านทางพอร์ตอนุกรมตามมาตรฐาน RS-232



รูปที่ 6.2 แสดงผังของการถ่ายเทข้อมูลจากเครื่องคอมพิวเตอร์ไปยังเครื่องควบคุม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 7

การประยุกต์การใช้งานและผลการทดสอบการทำงาน

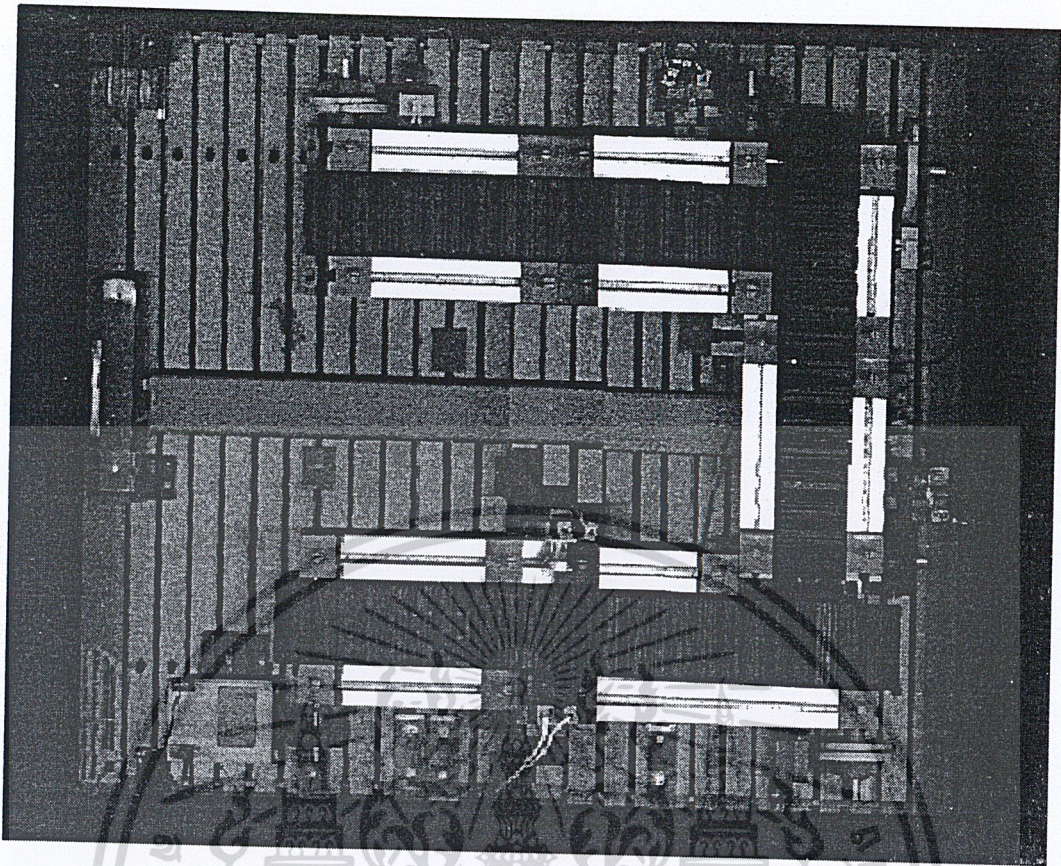
ในบทนี้จะนำเสนอผลการทดสอบ และวิธีการทำงานของเครื่องควบคุม PLC ที่ได้ ออกแบบ โดยมาชุดสายพานลำเลียงเป็นตัวอย่างในการแสดงการควบคุม ตลอดจนเป็นการทดสอบ การป้อนโปรแกรมทางคอมพิวเตอร์ การตรวจความผิดพลาดของข้อมูล การเปลี่ยนข้อมูลเป็น Hex Code การถ่ายเทข้อมูลจากโปรแกรมสนับสนุนมาสู่เครื่องควบคุม และการติดต่อจากบอร์ดหน่วยประมวลผลกลาง ไปยังหน่วยอินพุต/เอาต์พุต ที่เป็นส่วนของลูกข่าย โดยการทดสอบจะแบ่งออกเป็น ส่วนต่าง ๆ ดังนี้

7.1 การประยุกต์ใช้งานเครื่องควบคุมที่โปรแกรมได้

การทดสอบการใช้งานโปรแกรมสนับสนุนนี้ จะเป็นการจำลองการควบคุมสายพานลำเลียง โดยจะมีส่วนของการนับจำนวนของสิ่งของ เมื่อได้จำนวนสิ่งของครบตามจำนวนที่กำหนดแล้ว ก็จะหยุดการทำงานทั้งหมด

7.1.1 ส่วนประกอบของชุดจำลองสายพานลำเลียง

- M1 (มอเตอร์กระแสตรง 12 โวลต์)
- M2 (มอเตอร์กระแสตรง 12 โวลต์)
- M3 (มอเตอร์กระแสตรง 12 โวลต์)
- L1 (หลอดไฟเป็นสัญญาณนับ)
- S1 (PHOTO SENSOR)
- SW1 (สวิตช์ START)
- SW2 (สวิตช์ STOP)
- SW3 (สวิตช์ RESET การนับ)



รูปที่ 7.1 แสดงอุปกรณ์ของชุดสายพานลำเลียง

7.1.2 ขั้นตอนในการทำงานของชุดสายพานลำเลียง

เมื่อเรากด สวิตช์ START มอเตอร์ทั้งสามตัว และหลอดไฟที่ใช้ในการตรวจนับจำนวน จะเริ่มทำงาน เมื่อการนับค่าของชิ้นส่วนของที่ต้องการนับได้ตามที่กำหนด จะมีการหน่วงเวลาให้ของชิ้นสุดท้าย ผ่านไปก่อน จึงจะหยุดการทำงาน เมื่อจะให้เริ่มนับใหม่ก็ทำการรีเซ็ตการนับให้เป็นศูนย์ จึงเริ่มทำการนับใหม่ หรือเมื่อต้องการจบการทำงานก็กด สวิตช์ STOP

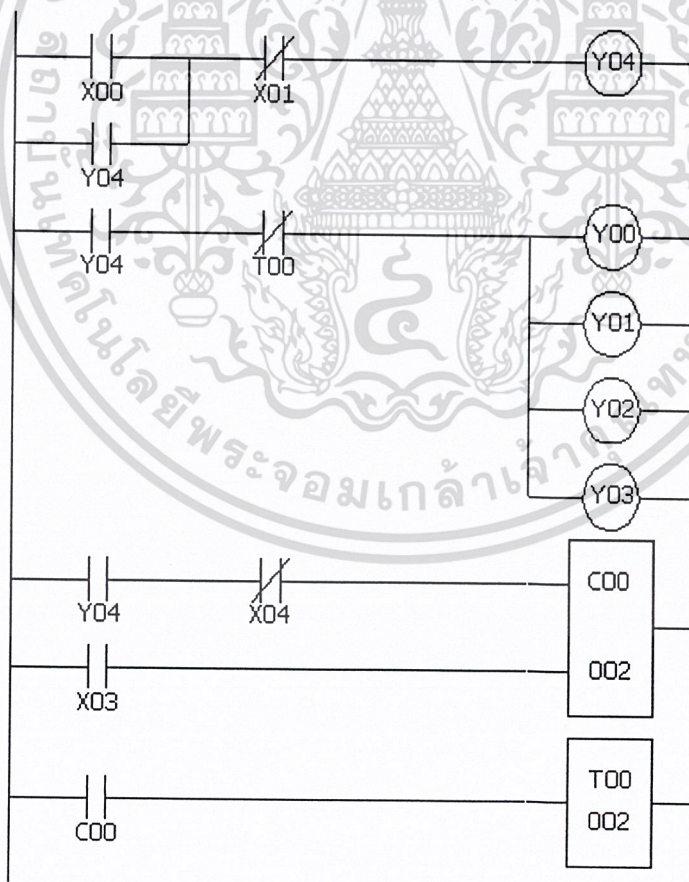
7.1.3 การกำหนดอินพุท/เอาต์พุทของชุดสายพานลำเลียง

ในการกำหนดอินพุทและเอาต์พุท เราจะแบ่งแยกตัวลูกข่ายให้เห็นชัดเจนว่า ส่วนของการควบคุมการ START และ STOP จะอยู่ที่หนึ่ง ส่วนของการติดต่อกับชุดสายพานจะอยู่อีกที่หนึ่ง โดยจะแบ่งออกได้ดังนี้

อินพุทรีเลย์	อุปกรณ์	หมายเหตุ
X00	สวิตช์ START	ปกติเปิด
X01	สวิตช์ STOP	ปกติเปิด
X03	สวิตช์รีเซ็ตการนับ	ปกติเปิด
X04	PHOTO SENSOR	ปกติเปิด

เอาต์พุทรีเลย์	อุปกรณ์	หมายเหตุ
Y00	มอเตอร์ M1	12 VDC
Y01	มอเตอร์ M2	12 VDC
Y02	มอเตอร์ M3	12 VDC
Y03	หลอดไฟสำหรับการนับ	12 VDC

7.1.4 โปรแกรมแลดเดอร์ไดอะแกรมของชุดสายพานลำเลียง



รูปที่ 7.3 แสดง โปรแกรมแลดเดอร์ไดอะแกรมของชุดสายพานลำเลียง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

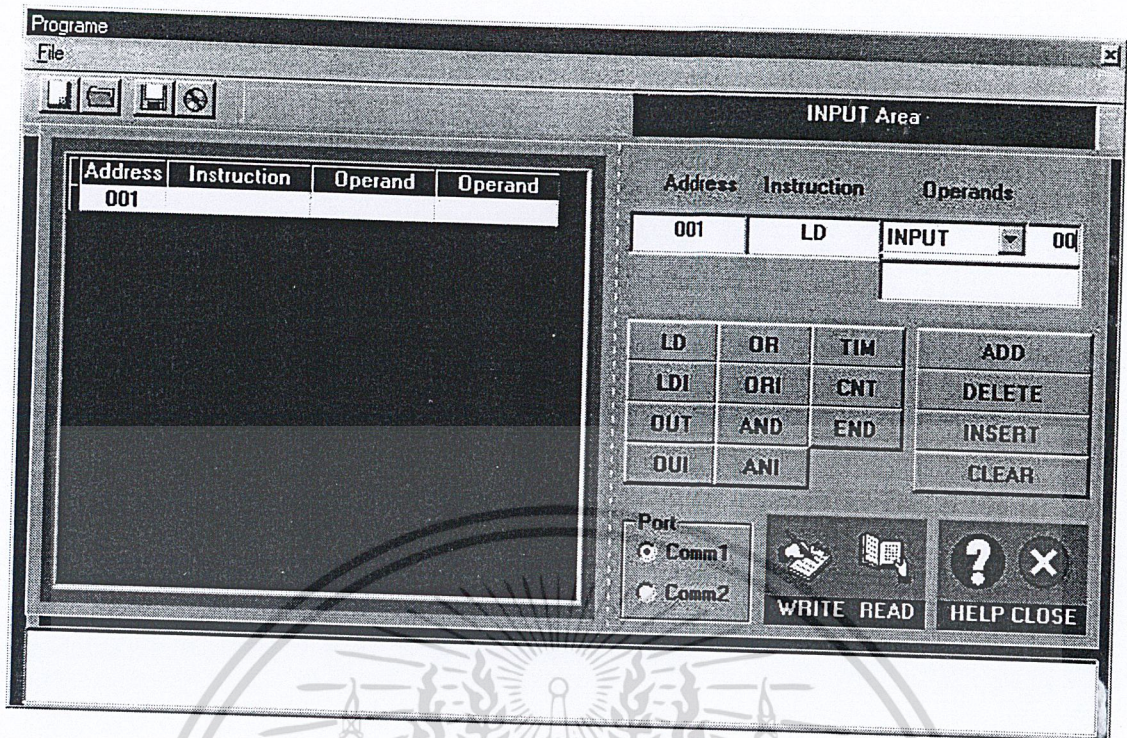
7.1.5 โปรแกรมชุดคำสั่งบูลีนของชุดสายพานลำเลียง

ADDRESS	INSTRUCTION	AREA	DATA
0000	LD	X00	
0001	OR	Y04	
0002	ANDNOT	X01	
0003	OUT	Y04	
0004	LD	Y04	
0005	ANDNOT TIM	00	
0006	OUT	Y00	
0007	OUT	Y01	
0008	OUT	Y02	
0009	OUT	Y03	
0010	LD	Y04	
0011	AND	X04	
0012	LD	X03	
0013	CNT	00	12
0014	LD CNT	00	
0015	TIM	00	01
0016	END		

7.2 การประยุกต์ใช้งานโปรแกรมสนับสนุน

การทดสอบการทำงานของโปรแกรมสนับสนุน โดยจะใช้การป้อนโปรแกรมที่เป็นชุดคำสั่งในตารางเข้าในเครื่องควบคุม โดยจะมีหน้าต่างของการป้อนโปรแกรมดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



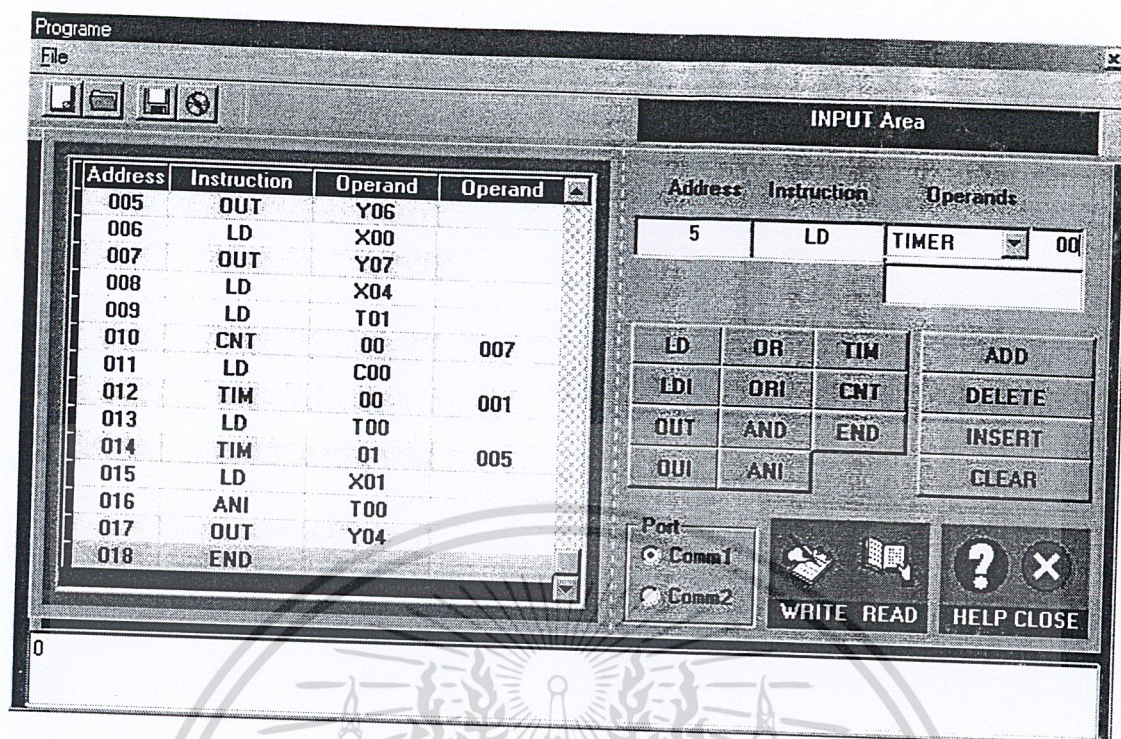
รูปที่ 7.5 แสดงหน้าที่ของปุ่มคำสั่งการทำงานหลักของการโปรแกรม

หน้าที่ของปุ่มคำสั่งในการของโปรแกรม

1. New การสร้างไฟล์ชุดคำสั่งบูลีนใหม่
2. Open การเปิดไฟล์จากหน่วยความจำ
3. Save การบันทึกไฟล์ลงในหน่วยความจำ
4. Add การเพิ่มบรรทัดของข้อมูล
5. Delete การลบบรรทัดของข้อมูล
6. Insert การแทรกบรรทัดของข้อมูล
7. Clear การทำให้บรรทัดของข้อมูลว่าง
8. Write การส่งข้อมูลออกทางพอร์ตอนุกรมไปสู่เครื่องควบคุม
9. Close การปิดโปรแกรม

ในการเขียนโปรแกรม เราจะเริ่มที่ Address ที่ 0001 จนกระทั่งจบโปรแกรมด้วยคำสั่ง END โดยที่เมื่อมีการป้อนโปรแกรม 1 ชุดคำสั่ง เช่น LD X00 แล้วก็ทำการคลิกที่ปุ่ม ADD 1 ครั้ง แล้วชุดคำสั่งจะเข้าไปอยู่ที่ตารางในบรรทัดนั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 7.6 การป้อนโปรแกรม(1)

เมื่อได้โปรแกรมที่ป้อนจบแล้ว คือเมื่อมีคำสั่ง End เราจะทำการถ่ายเทข้อมูลที่ได้แปลงไปเป็นรหัส HEX CODE เพื่อออกเป็นข้อมูลที่ไมโครโปรเซสเซอร์ที่บอร์ดควบคุมเข้าใจ อย่างเช่น โปรแกรมในตารางเมื่อเปลี่ยนเป็น HEX CODE จะได้ดังนี้

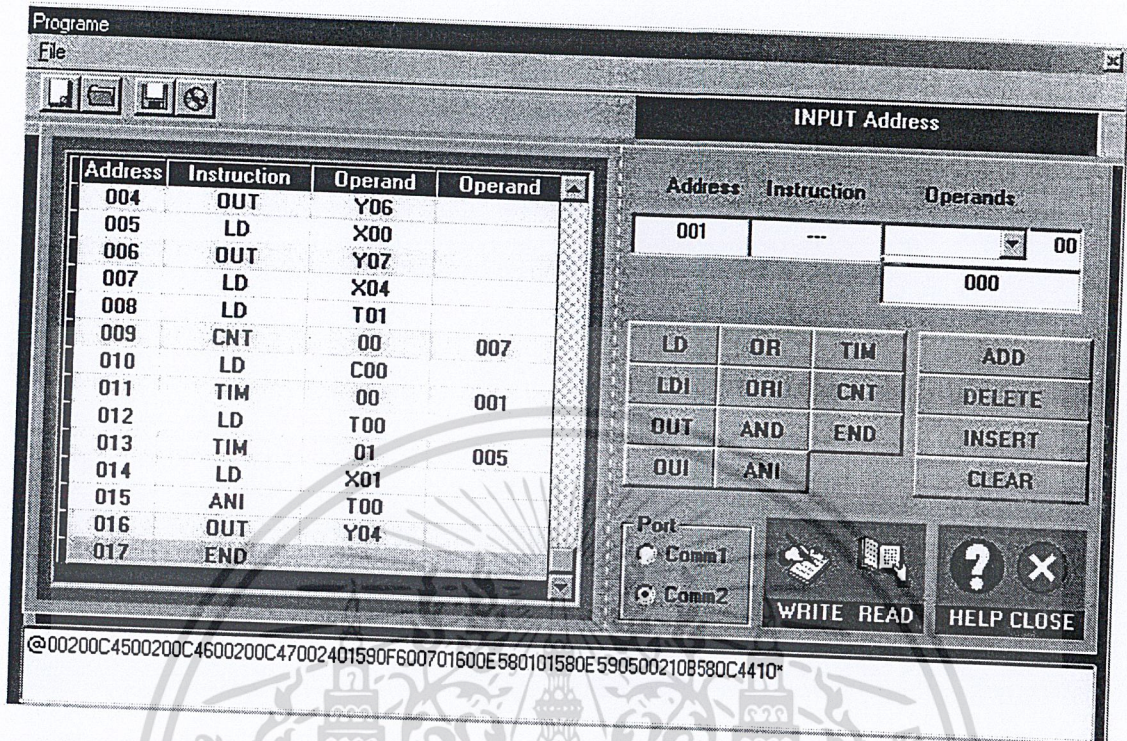
คำสั่งบูลีน	HEX CODE
LD	00
X00	20
OR	06
Y04	44
AND NOT	0A
X01	21
OUT	0C
Y04	44
LD	01
Y04	44

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

AND NOT	0B
TIM 00	58
OUT	0C
Y00	40
OUT	0C
Y01	41
OUT	0C
Y02	42
OUT	0C
Y03	43
LD	01
Y04	44
AND	08
X04	24
LD	00
X03	23
CNT	0F
00	5C
12	0C
LD CNT	01
00	5C
TIM	0E
00	58
01	01
END	10

โปรแกรมที่เขียนแล้ว และได้แปลงเป็น HEX CODE แล้ว จะถูกส่งออกทางพอร์ต
อนุกรมเมื่อเราทำการกดปุ่ม WRITE

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 7.7 การป้อนโปรแกรม(2)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 8

บทสรุป

การพัฒนาระบบการทำงานของเครื่องควบคุมแบบตรรกที่โปรแกรมได้ หรือเครื่องควบคุม PLC/PC โดยทั่วไปแล้วการพัฒนาจะเป็นในรูปของการพัฒนากระบวนการทำงานที่ซับซ้อน ซึ่งสามารถที่จะควบคุมกระบวนการที่ยุงยากได้ จริงอยู่ที่เครื่องควบคุมที่ได้ทำการออกแบบมานี้ จะมีความสามารถต่ำกว่าที่มีอยู่ทั่วไปในท้องตลาด แต่ในส่วนที่มีความสามารถต่ำกว่า ก็จะได้ถูกทดแทนด้วยส่วนของราคา และการใช้งานได้ง่าย เนื่องจากอุปกรณ์ที่นำมาใช้ประกอบนี้ ก็เป็นอุปกรณ์ที่หาได้ง่าย เหมาะสำหรับผู้ที่ยากจะควบคุมกระบวนการไม่ซับซ้อน การควบคุมการทำงานที่มีส่วนของอินพุทและเอาต์พุทน้อย กระบวนการไม่ยุ่งยาก จะสามารถใช้ประโยชน์จากทรัพยากรที่เตรียมไว้ได้อย่างคุ้มค่า แทนการที่จะนำเครื่องควบคุมที่มีประสิทธิภาพสูงแต่ใช้งานได้ไม่เต็มประสิทธิภาพ และอีกส่วนหนึ่งที่จะมาทดแทนก็คือ ระยะเวลาการควบคุมระหว่างหน่วยประมวลผล และส่วนของอินพุทและเอาต์พุท เนื่องจากการติดต่อของอินพุทและเอาต์พุทเป็นแบบ Remote Input/Output ซึ่งสามารถที่จะมีระยะห่างกันได้หลายร้อยเมตร ตามมาตรฐาน RS-422 จึงเหมาะที่จะควบคุมในพื้นที่กว้าง เช่น งานทางด้านเกษตรกรรมที่มีการควบคุมที่ไม่ยุ่งยาก ต้องการราคาถูก

นอกจากนี้ในการป้อนโปรแกรม ยังสามารถที่จะเข้าใจง่าย ด้วยภาษาที่ใช้กันโดยทั่วไปของเครื่องควบคุมแบบตรรกอื่น ๆ โดยจะมีโปรแกรมสนับสนุนการป้อนโปรแกรม ซึ่งเขียนขึ้นโดยใช้ภาษา Visual Basic ซึ่งจะมีความสะดวกต่อการเปลี่ยนแปลงกระบวนการทำงาน

เครื่องควบคุมแบบตรรก ที่ได้ทำการออกแบบมานี้ ถึงแม้การออกแบบที่ได้มาจะมีประสิทธิภาพเป็นที่น่าพอใจ และยังสามารถจะใช้ร่วมกับโปรแกรมสนับสนุนที่ได้ออกแบบมาได้เป็นอย่างดี แต่ยังมีจุดด้อยอยู่บ้าง ซึ่งก็คือ ส่วนของความสามารถของวงจรรวม ซึ่งนำมาใช้เป็นหน่วยประมวลผลกลางนั้น ยังมีความสามารถจำกัด เช่นเมื่อการทำงานมีการต่อหน่วยอินพุทและเอาต์พุทหลายยูนิต อาจทำให้การประมวลผลช้าลง ซึ่งถ้าใช้หน่วยประมวลผลที่มีประสิทธิภาพดีขึ้น ก็จะส่งผลให้การทำงานดีขึ้นเช่นกัน

บรรณานุกรม

- [1] เอกสารประกอบการอบรมเชิงวิชาการปฏิบัติการ PROGRAMMABLE LOGIC CONTROLLER, ภาควิชาเทคโนโลยีการวัดคุมทางอุตสาหกรรม คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
- [2] สุพรรณ กุลพานิชย์, “เทคนิคและการประยุกต์ใช้งาน Programmable Controller”, ภาควิชาเทคโนโลยีการวัดคุมทางอุตสาหกรรม คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
- [3] สุเชียร เกียรติสุนทร, “หลักการทํางานและเทคนิคการประยุกต์ใช้งาน PC/PLC”, ภาควิชาระบบควบคุม คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
- [4] ชูชัย ธนสารตั้งเจริญ, ทินกร คึก, “สื่อสารข้อมูล”, ฟิสิกซ์ เซ็นเตอร์ กรุงเทพมหานคร
- [5] สุทธิศักดิ์ พงศ์ธนาพานิชย์, “Visual Basic 4.0 Professional, ซีเอ็ดยูเคชั่น
- [6] ทวีพล ช่อสตั๋ย, วิริยะ กองรัตน์, ผ.ศ. วิทยา ทิพสุวรรณพร, ผ.ศ. สุพรรณ กุลพานิชย์, “การออกแบบโปรแกรมโฮสคอมพิวเตอร์กับเครื่องควบคุมแบบตรรกจํานวนหลายจุด”, การประชุมวิชาการทางวิศวกรรมไฟฟ้า ครั้งที่ 19 มหาวิทยาลัยขอนแก่น, 1996, หน้า cp7-cp14

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้