

ระบบควบคุมสารละลายด้วยพีแอลซี

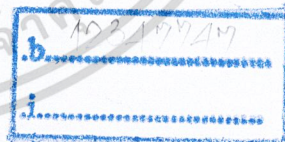
SOLUTION CONTROL SYSTEM USING PLC



T117580



เลขหมู่.....
เลขทะเบียน 117580
วัน,เดือน,ปี..... 9 ค.ค. 2554



ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิชาวิศวกรรมแมคคาทรอนิกส์
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2553

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

SOLUTION CONTROL SYSTEM USING PLC



**THIS THESIS IS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF
BACHELOR OF ENGINEERING IN MECHATRONICS ENGINEERING
FACULTY OF ENGINEERING
KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG
ACADEMIC YEAR 2010**

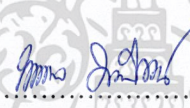
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์ปีการศึกษา 2553

สาขาวิชาวิศวกรรมการวัดและควบคุม คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง ระบบควบคุมสารละลายด้วยพีแอลซี
SOLUTION CONTROL SYSTEM USING PLC

ผู้จัดทำ นายขจิตพงศ์ ป้องกัน 50011621
นายสมบุญ คารมศิลป์ 50011625
นายเสฐียรพงศ์ บริรักษ์ภูสิฐ 50011789


.....อาจารย์ที่ปรึกษา
(ผู้ช่วยศาสตราจารย์ ดร. นพดล มณีรัตน์)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระบบควบคุมสารถ่ายด้วย PLC

โดย

นาย ขจิตพงศ์ ป้องกัน 50011621

นาย สมบุญ คารมศิลป์ 50011625

นาย เสฐียรพงศ์ บริรักษ์ภูสิฐ 50011789

อาจารย์ที่ปรึกษา

ผู้ช่วยศาสตราจารย์ ดร.นพดล มณีรัตน์

ปีการศึกษา 2553

บทคัดย่อ

เนื่องจากปัจจุบันอุตสาหกรรมส่วนใหญ่ในประเทศไทยได้นำเอา PLC (Programmable Logic Controller) มาใช้ในระบบควบคุมอัตโนมัติ ซึ่งมีความน่าเชื่อถือสูงและใช้งานได้ง่าย โดยนำมาใช้ควบคุมกระบวนการต่อเนื่อง (Batch Control) หลายกระบวนการ เช่น การควบคุมป้อนของเหลวในโรงงานอุตสาหกรรม ซึ่งมีลำดับและความต่อเนื่องที่ซับซ้อน ดังนั้นปริญญานิพนธ์ฉบับนี้จึงนำเสนอระบบควบคุมกระบวนการป้อนของเหลวแบบต่อเนื่องที่ควบคุมด้วย PLC โดยสามารถควบคุมการทำงานผ่านทางโปรแกรมบนหน้าจอ Computer (Graphic User Interface) ผ่านพอร์ตอนุกรม RS-232C ของ Computer สำหรับโปรแกรมควบคุมการทำงานของป้อนถูกพัฒนาขึ้นด้วยภาษาวิชวลเบสิก 6.0 (Visual Basic 6.0)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Solution control system using PLC

By

Mr. Kajitpong

Phongkun

Mr. Somboon

Karomsilp

Mr. Satienpong

Borirakphusit

Advisor

Asst. Prof. Dr. Noppadol Maneerat

Academic Year 2010

ABSTRACT

Nowadays, most of Thailand's industrial use PLC machines in their automatic control systems due to the properties of PLC which are reliability and ease to use. PLC machines are used in several batch control systems such as pump liquid control industry, which has sequence and continued complexity. So this thesis presents the continuous pump control system using PLC through a serial port (RS-232C) on computer. The control method can be done via GUI (Graphic User Interface). The pump control program is developed using Visual Basic 6.0

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

	หน้า
บทคัดย่อ	I
บทคัดย่อภาษาอังกฤษ	II
สารบัญ	III
สารบัญรูป	VI
สารบัญตาราง	VIII
บทที่ 1 บทนำ	1
1.1 ความเป็นมา	1
1.2 วัตถุประสงค์ของโครงการ	1
1.3 ขอบเขตของโครงการ	2
1.4 ประโยชน์ที่คาดว่าจะได้รับ	2
บทที่ 2 หลักการและทฤษฎี	3
2.1 PLC (Programmable Logic Controller)	3
2.1.1 โครงสร้างและหลักการทำงานของ PLC	4
2.1.1.1 หน่วยประมวลผลกลางหรือ CPU	5
2.1.1.2 หน่วยความจำ (Memory Unit)	5
2.1.1.3 หน่วยอินพุต / เอาต์พุต (I/O Unit)	6
2.1.1.4 แหล่งจ่ายไฟ (Power Supply)	6
2.1.1.5 อุปกรณ์ต่อร่วม (Peripheral Device)	6
2.1.2 พื้นที่หน่วยความจำของ PLC	7
2.1.3 ความสามารถของ PLC	8
2.1.3.1 งานที่ทำตามลำดับก่อนหลัง (Sequence Control)	8
2.1.3.2 งานควบคุมสมัยใหม่ (Sophisticated Control)	8
2.1.3.3 งานที่ควบคุมเกี่ยวกับงานอำนวยการ (Supervisory Control)	9
2.1.4 การสื่อสารข้อมูลแบบอนุกรม (Serial)	9
2.1.4.1 มาตรฐานในการสื่อสารข้อมูลแบบอนุกรม	9
2.1.4.2 ข้อตกลงในการสื่อสารของ PLC	10

เอกสารนี้เป็นเอกสารที่จัดทำขึ้นเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ในการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น 2.2 โปรแกรม Visual Basic 6.0
2.2.1 การเขียน โปรแกรม Visual Basic 6.0 ขั้นพื้นฐาน

สารบัญ (ต่อ)

	หน้า
2.2.1.1 ที่มาและลักษณะของโปรแกรม Visual Basic 6.0	11
2.2.1.2 รายละเอียดของส่วนประกอบต่างๆ ของหน้าจอ	11
2.2.1.3 จุดเด่นของ Visual Basic 6.0	12
2.2.1.4 รูปแบบการพัฒนาแอปพลิเคชันกับ Visual Basic 6.0	13
2.2.1.5 เริ่มสร้างแอปพลิเคชันด้วย Visual Basic 6.0	14
2.2.2 การสื่อสารกับ Communication Port	15
2.2.2.1 การใช้ MsComm Control	15
2.2.2.2 CommPort	15
2.2.2.3 Setting	15
2.2.2.4 PortOpen	16
2.2.2.5 การเปิดพอร์ตอนุกรม	17
2.3 รีเลย์	18
2.3.1 หน้าที่ของคอนแทกเตอร์ (Contactor)	18
2.3.2 ข้อคำนึงในการใช้งานรีเลย์ทั่วไป	19
2.4 การสื่อสารผ่านพอร์ตขนาน (Parallel Port)	19
2.4.1 การติดต่อพอร์ตขนานด้วย Visual Basic 6.0	21
2.5 การใช้งานพอร์ตอนุกรม RS-232C	22
2.5.1 การเชื่อมต่ออุปกรณ์ภายนอกเข้ากับ Computer ด้วยสาย DB9	22
2.5.2 การทำงานของขาสัญญาณ DB9	23
2.5.3 อัตราการส่งข้อมูล (Baud rate)	24
2.6 อุปกรณ์อิเล็กทรอนิกส์	24
2.5.1 วงจรขับมอเตอร์ L293D	24
บทที่ 3 การคำนวณและการสร้าง	26
3.1 ขอบเขตการทำงานของระบบ	26
3.2 การออกแบบและควบคุมปั๊มเพอริสทอลติก (Peristaltic Pump)	27
3.2.1 การควบคุมผ่านทาง Analog Interface	27
3.2.2 การควบคุมการหมุน (Start/Stop)	27
3.2.3 การควบคุมทิศทางการหมุน (Rotation Direction)	28

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

	หน้า
3.2.4 การควบคุมความเร็วในการหมุน (Speed Control)	28
3.2.5 การออกแบบวงจรแสดงผล	28
3.2.6 การควบคุมความเร็วมอเตอร์ด้วยพอร์ตขนาน	31
3.3 การคำนวณและการสร้างซอฟต์แวร์	32
3.3.1 การออกแบบแลตเตอร์ไดอะแกรมเพื่อควบคุมกระบวนการ	32
3.3.2 การออกแบบหน้าจออินเทอร์เฟซสำหรับเชื่อมต่อและสั่งกระบวนการ	34
3.4 กล้องควบคุม	35
3.4.1 สวิตช์เปิด-ปิด	35
3.4.2 พอร์ตอนุกรม RS-232C	36
3.4.3 ไฟแสดงผล	37
บทที่ 4 การทดลองและผลการทดลอง	39
4.1 ตั้งค่าและเริ่มต้นใช้โปรแกรมควบคุมกระบวนการ	39
4.2 ผลการทดลอง	42
4.2.1 การควบคุมการเปิด-ปิดและทิศทางการหมุน	42
4.2.2 การทดลองควบคุมความเร็วด้วยพอร์ตขนาน	43
บทที่ 5 บทวิจารณ์และสรุป	46
5.1 สรุปผลการปฏิบัติงาน	46
5.2 ปัญหาที่พบในการทำงาน	46
5.3 การแก้ไขปัญหาที่เกิดขึ้น	46
5.4 ข้อเสนอแนะในการค้นคว้าและพัฒนา	47
เอกสารอ้างอิง	48
ภาคผนวก	49
ภาคผนวก ก	50
ภาคผนวก ข	88
ภาคผนวก ค	112
ภาคผนวก ง	119
ประวัติผู้แต่ง	127

สารบัญรูป

รูปที่	หน้า
2.1 PLC รุ่น CPM1A	3
2.2 โครงสร้างภายในของ PLC	4
2.3 รูปแบบของบล็อก (Block Format)	10
2.4 หน้าจอโปรแกรม Visual Basic 6.0	11
2.5 องค์ประกอบสำคัญในโปรแกรม Visual Basic 6.0	12
2.6 รีเลย์	18
2.7 พอร์ตขนาน	19
2.8 การเชื่อมต่ออุปกรณ์ภายนอกผ่าน DB9 แบบ Null modem	23
2.9 การเชื่อมต่ออุปกรณ์ภายนอกผ่าน DB9 แบบ 3 เส้น	23
2.10 ตำแหน่งขาของไอซี L293D	24
2.11 หลักการทำงานของไอซี L293D	25
3.1 ระบบการทำงานของกระบวนการ	26
3.2 อินพุตสำหรับอนาล็อกอินเทอร์เฟซ (Analog Interface Input)	27
3.3 วงจรรีเลย์ควบคุมทิศทางการหมุนและควบคุมการเปิด-ปิด	28
3.4 วงจรแสดงผล	29
3.5 ลายวงจรของวงจรแสดงผล	30
3.6 วงจรแสดงผลจริง	30
3.7 หน้าจอ GUI ของโปรแกรมควบคุมความเร็ว	31
3.8 ฟังก์ชัน BCD	32
3.9 ฟังก์ชัน TIM	33
3.10 ฟังก์ชัน CNT	33
3.11 ฟังก์ชัน MUL	34
3.12 ฟังก์ชัน DIVL	34
3.13 หน้าจออินเทอร์เฟซ	35
3.14 สวิตช์เปิด-ปิดกล่องควบคุม	36
3.15 พอร์ตอนุกรม RS-232C	36

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป (ต่อ)

รูปที่	หน้า
3.16 การเชื่อมต่อระหว่างวงจรรีเลย์กับพอร์ตอนุกรม RS-232C	37
3.17 ไฟแสดงผล	37
3.18 การติดตั้งอุปกรณ์ภายในกล่องควบคุม	38
4.1 หน้าจอ GUI เมื่อเปิดโปรแกรมขึ้นมา	39
4.2 การตั้งค่าต่างๆ ในการควบคุมปั๊ม	40
4.3 การกำหนดสถานะในแต่ละขั้นตอน	40
4.4 การป้อนค่าเวลาให้กับปั๊ม	41
4.5 การสั่งการทำงานของกระบวนการ	42
4.6 กราฟการดูดกลืนแสงของสารละลายที่ควบคุมปั๊มด้วยมือเปิด-ปิด	42
4.7 กราฟการดูดกลืนแสงของสารละลายที่ควบคุมปั๊มด้วยชุดควบคุม	43
4.8 วงจรควบคุมความเร็วมอเตอร์ขับเคลื่อนโดยไอซี L293D	43
4.9 หน้าจอ GUI ของโปรแกรมควบคุม	44
4.10 ผลการทดสอบความเร็วปั๊ม	44

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

ตารางที่	หน้า
2.1 ขาสัญญาของพอร์ตขนาน	20
2.2 Register Address	21
2.3 การจัดขาของคอนเน็กเตอร์อนุกรมแบบ DB9 และหน้าที่การใช้งาน	22



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1

บทนำ

1.1 ความเป็นมา

จากงานวิจัย “ชุดเครื่องมือขนาดเล็กแบบพกพาได้สำหรับวินิจฉัยโรคไตนอกห้องปฏิบัติการ” ของคณะผู้วิจัยคณะวิทยาศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ซึ่งต้องใช้ปั๊มสำหรับดูดสารละลายและสารตัวอย่าง เพื่อนำมาทำปฏิกิริยากัน สำหรับการควบคุมปั๊มจะใช้วิธีควบคุมด้วยมือ คือ เปิด-ปิดปั๊มด้วยมือ เวลาที่ใช้สำหรับการดูดสารละลายก็ใช้การจับเวลาจากนาฬิกาจับเวลา ทำให้พบปัญหา คือ ปริมาณสารที่ได้มีปริมาณไม่ได้ตามที่ต้องการ การควบคุมมีความยุ่งยากเพราะบางกระบวนการต้องใช้ปั๊มหลายตัว การเปิด-ปิดและการจับเวลามีความยุ่งยากมาก ยิ่งถ้ากระบวนการของปั๊มแต่ละตัวใช้เวลาไม่เท่ากัน จะทำให้เกิดปัญหาที่ยุ่งยากมากขึ้นไปอีก ทำให้ขาดความเที่ยงตรง

คณะผู้จัดทำโครงการนี้จึงได้เล็งเห็นว่า เพื่อให้การศึกษาวิจัยนี้มีความสะดวกมากยิ่งขึ้น การควบคุมการเปิด-ปิด และการควบคุมปริมาณสารมีความเที่ยงตรงมากยิ่งขึ้น จึงได้มีการศึกษาและพัฒนาอุปกรณ์ชุดควบคุมเพื่อใช้ทดแทนการควบคุมแบบเดิม ซึ่งจะทำให้การทำงานง่ายมากขึ้น ประสิทธิภาพของการวินิจฉัยมีความเที่ยงตรงมากยิ่งขึ้น เนื่องจากปริมาณสารเคมีกับสารตัวอย่างที่ใช้ทำปฏิกิริยากันมีปริมาตรที่เที่ยงตรง ซึ่งคณะผู้ทำโครงการนี้ได้เลือกใช้ PLC เป็นอุปกรณ์สำหรับใช้ควบคุมกระบวนการต่างๆ เนื่องจากมีความแม่นยำในการใช้งานสูง ใช้งานง่าย มีฟังก์ชันสำหรับใช้ควบคุมกระบวนการที่ต้องการครบถ้วน โดยใช้การสั่งงานควบคุมกระบวนการทางหน้าจออินเตอร์เฟซ (Interface) ผ่านทางพอร์ตอนุกรม RS-232C โดยโปรแกรมควบคุมถูกพัฒนาขึ้นด้วยโปรแกรม Visual Basic 6.0

1.2 วัตถุประสงค์ของโครงการ

1. เพื่อศึกษาการทำงานและโครงสร้างภายในของ PLC
2. เพื่อศึกษาการเขียนแลดเดอร์โปรแกรมในเชิงลึกมากขึ้น
3. เพื่อศึกษาการเชื่อมต่อและรับส่งข้อมูลกับอุปกรณ์ภายนอกของ PLC
4. เพื่อศึกษาการเขียน โปรแกรม Visual Basic 6.0 เพื่อเชื่อมต่อกับอุปกรณ์ภายนอก
5. เพื่อศึกษาการทำงานของปั๊มประเภทเพอริสเทติก
6. เพื่อฝึกประสบการณ์ในการแก้ไขปัญหาที่เกิดขึ้นในการทำงาน

เอกสารนี้เป็นเอกสารเพื่อฝึกการรับผิดชอบในหน้าที่ที่ได้รับมอบหมายนั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ เพื่อเพิ่มความรู้ภาคทฤษฎีที่ได้เรียนรู้มาประยุกต์ใช้กับการปฏิบัติงานจริง

1.3 ขอบเขตของโครงการ

1. สร้างหน้าจอแสดงผล เชื่อมต่อและควบคุมกระบวนการทางพอร์ตอนุกรม RS-232C ผ่านทางหน้าจอ Computer ด้วยการเขียนโปรแกรม Visual Basic 6.0
2. ศึกษาการทำงานของอุปกรณ์ PLC และสามารถเขียนแลดเดอร์โปรแกรม เพื่อควบคุมการทำงานในกระบวนการต่างๆ ได้
3. ศึกษาการติดต่อสื่อสาร รับ-ส่งข้อมูลกับอุปกรณ์ภายนอกของ PLC
4. ศึกษาการทำงานของปั๊มเพอร์สเทลติกในโหมดควบคุมระยะไกลเพื่อควบคุมการเปิด-ปิด ทิศทางการหมุนและการควบคุมความเร็วของปั๊ม

1.4 ประโยชน์ที่คาดว่าจะได้รับ

1. เข้าใจหลักการทำงานของ PLC มากยิ่งขึ้น สามารถเขียนแลดเดอร์โปรแกรมเพื่อควบคุม PLC ให้ทำงานตามกระบวนการต่างๆ ได้
2. มีความรู้ความเข้าใจในการติดต่อสื่อสารข้อมูลระหว่าง PLC กับ โปรแกรม Visual Basic 6.0 ผ่านทางพอร์ตอนุกรม RS-232C
3. มีความรู้ความสามารถในการควบคุมปั๊มเพอร์สเทลติกในโหมดควบคุมระยะไกลให้ทำงานตามที่ต้องการได้
4. สามารถนำความรู้มาใช้ในการปฏิบัติงานจริงได้ และสามารถแก้ไขปัญหาที่เกิดขึ้นได้
5. มีความรับผิดชอบต่อนหน้าที่ที่ได้รับมอบหมายจนสำเร็จ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

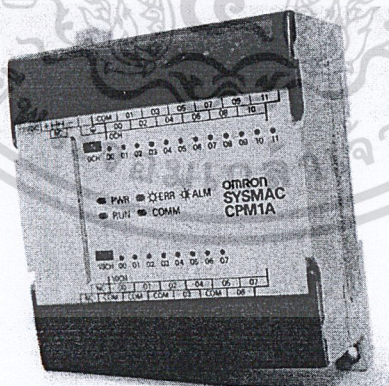
บทที่ 2

หลักการและทฤษฎี

ในการจะออกแบบและพัฒนาระบบตรวจสอบสารละลายในของเหลว จำเป็นต้องศึกษาองค์ประกอบต่างๆ ที่จำเป็นของระบบให้เข้าใจเสียก่อน พบว่าระบบตรวจสอบสารละลายนั้นมีส่วนที่สำคัญหลายส่วน ดังนั้นในบทนี้จะศึกษาและอธิบายถึงองค์ประกอบต่างๆ ที่จะนำไปใช้งานจริงในระบบตรวจสอบสารละลาย ซึ่งประกอบด้วย PLC โปรแกรม Visual Basic 6.0 และรีเลย์ รวมถึงรายละเอียดของพอร์ตที่ใช้กัน ซึ่งมีรายละเอียดดังนี้

2.1 PLC (Programmable Logic Controller)

PLC (Programmable Logic Controller) เป็นอุปกรณ์ที่สร้างขึ้น เพื่อใช้ควบคุมการทำงานของเครื่องจักรหรือระบบต่างๆ แบบอัตโนมัติในโรงงานอุตสาหกรรม ซึ่งสามารถที่จะโปรแกรมได้ ถูกสร้างและพัฒนาขึ้นเพื่อทดแทนวงจรรีเลย์ อันเนื่องมาจากความต้องการที่อยากจะได้เครื่องควบคุมที่มีราคาถูก สามารถใช้งานได้อย่างอนเนกประสงค์ และสามารถเรียนรู้การใช้งานได้ง่าย สำหรับในโครงการนี้ได้เลือกใช้ PLC รุ่น CPM1A ซึ่งมีลักษณะดังรูปที่ 2.1 และสามารถดูรายละเอียดคุณสมบัติเพิ่มเติมได้จากภาคผนวก ข



รูปที่ 2.1 PLC รุ่น CPM1A

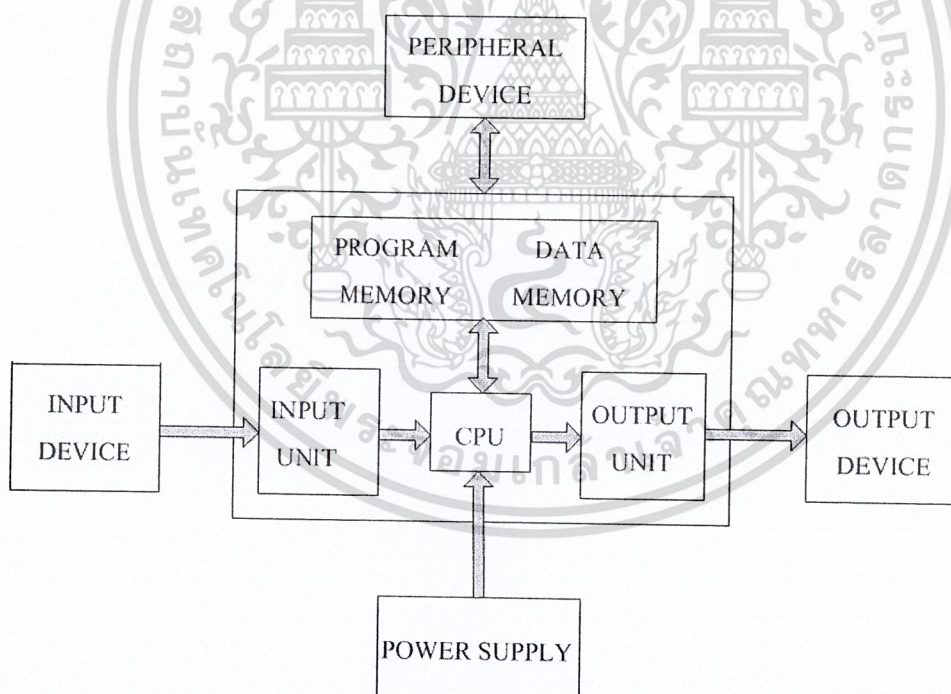
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ข้อแตกต่างระหว่าง PLC กับ Computer

1. PLC ถูกออกแบบและสร้างขึ้นเพื่อให้ทนต่อสภาพแวดล้อมในโรงงานอุตสาหกรรม โดยเฉพาะ
2. การโปรแกรมและการใช้งาน PLC ทำได้ง่ายไม่ยุ่งยากเหมือน Computer ทั่วไป PLC มีระบบการตรวจสอบตั้งแต่ช่วงติดตั้งจนถึงช่วงการใช้งาน ทำให้การบำรุงรักษาทำได้ง่าย
3. PLC ถูกพัฒนาให้มีความสามารถในการตัดสินใจมากขึ้นเรื่อยๆ ทำให้การใช้งานสะดวก ในขณะที่วิธีใช้ Computer ยุ่งยากและซับซ้อนมากยิ่งขึ้น

2.1.1 โครงสร้างและหลักการทำงานของ PLC

PLC เป็นอุปกรณ์ที่ใช้ควบคุมเครื่องจักร หรือระบบกระบวนการให้ทำงานตามโปรแกรมคำสั่งของผู้ใช้ และข้อมูลต่างๆที่ได้รับจากอินพุตหรือเอาต์พุตของ PLC จะทำงานตามช่วงเวลาตามลำดับขั้นตอนฟังก์ชันทางคณิตศาสตร์ มีส่วนประกอบสำคัญดังแสดงในรูปที่ 2.2



รูปที่ 2.2 โครงสร้างภายในของ PLC

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.1.1 หน่วยประมวลผลกลางหรือ CPU

ทำหน้าที่คำนวณและควบคุมซึ่งเปรียบเสมือนสมองของ PLC ภายในประกอบด้วย วงจรลอจิกหลายชนิดและมีไมโครโปรเซสเซอร์ (Microprocessor) ใช้แทนอุปกรณ์จำพวกรีเลย์ ไทมเมอร์/เคาน์เตอร์และซีเควนเซอร์ เพื่อให้ผู้ใช้สามารถออกแบบวงจรโดยใช้ Relay Ladder Diagram ได้ CPU จะยอมรับข้อมูลจากอุปกรณ์อินพุตต่างๆ จากนั้นจะทำการประมวลผลและเก็บข้อมูลโดยใช้โปรแกรมจากหน่วยความจำ หลังจากนั้นจะส่งข้อมูลที่เหมาะสมและถูกต้องออกไปยังอุปกรณ์เอาต์พุต ซึ่งเรียกว่า การสแกน (Scan) ซึ่งจะใช้เวลาจำนวนหนึ่งเรียกว่า เวลาสแกน (Scan Time) เวลาในการสแกนแต่ละรอบจะใช้เวลาประมาณ 0.001 ถึง 0.1 วินาที ทั้งนี้ขึ้นอยู่กับข้อมูลและความยาวของโปรแกรม จำนวนอินพุตและเอาต์พุตหรือ จำนวนอุปกรณ์ที่ต่อจาก PLC เช่น จอภาพ เครื่องพิมพ์ ฯลฯ อุปกรณ์เหล่านี้จะทำให้เวลาในการสแกนยาวนานขึ้น การเริ่มต้นสแกนเริ่มจากรับค่าของสถานะของอุปกรณ์จากหน่วยความจำเสร็จแล้วจะทำการปฏิบัติการตามโปรแกรมที่เขียนไว้ทีละคำสั่งจากหน่วยความจำนั้นจนสิ้นสุด แล้วส่งไปที่หน่วยเอาต์พุต

2.1.1.2 หน่วยความจำ (Memory Unit)

ทำหน้าที่เก็บรักษาโปรแกรมและข้อมูลที่ใช้การทำงาน โดยขนาดของหน่วยความจำจะถูกแบ่งออกเป็นบิตข้อมูล (Data Bit) ภายในหน่วยความจำ 1 บิต ก็จะมีสถานะทางลอจิก 0 หรือ 1 แตกต่างกันไปแล้วแต่คำสั่ง ซึ่ง PLC ประกอบด้วยหน่วยความจำ 2 ชนิดคือ ROM กับ RAM

1. ROM (Read Only Memory) ทำหน้าที่เก็บโปรแกรมสำหรับใช้ในการปฏิบัติงานของ PLC ตามโปรแกรมของผู้ใช้ หน่วยความจำแบบ ROM ยังสามารถแบ่งออกได้เป็น EPROM (Erasable Programmable Read-Only Memory) ซึ่งจะต้องใช้อุปกรณ์พิเศษในการเขียนและลบโปรแกรม เหมาะกับงานที่ไม่ต้องการเปลี่ยนแปลงโปรแกรม นอกจากนี้ยังมีแบบ EEPROM (Electrically Erasable Programmable Read-Only Memory) หน่วยความจำประเภทนี้ไม่ต้องใช้เครื่องมือพิเศษในการเขียนและลบโปรแกรม สามารถใช้งานได้เหมือนกับ RAM แต่ไม่ต้องใช้แบตเตอรี่สำรอง แต่ราคาจะแพงกว่าเนื่องจากรวมคุณสมบัติของ ROM และ RAM เข้าไว้ด้วยกัน
2. RAM (Random-Access Memory) ทำหน้าที่ในการเก็บโปรแกรมข้อมูลของผู้ใช้และข้อมูลที่ใช้ในการปฏิบัติงานของ PLC หน่วยความจำประเภทนี้จะมีแบตเตอรี่เล็กๆ ต่อเอาไว้เพื่อใช้เป็นไฟเลี้ยงข้อมูลกรณีเกิดไฟดับ การอ่านและการเขียนข้อมูลลงใน RAM ทำได้ง่ายมาก เพราะฉะนั้นจึงเหมาะกับงานในระบบทดลองเครื่องที่มีการเปลี่ยนแปลงแก้ไขโปรแกรมอยู่บ่อยๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.1.3 หน่วยอินพุต/เอาต์พุต (I/O Unit)

ส่วนของอินพุตและเอาต์พุต (I/O Unit) จะต่อร่วมกับชุดควบคุมเพื่อรับสถานะและสัญญาณต่างๆ เช่น หน่วยอินพุตรับสัญญาณหรือสถานะแล้วส่งไปยัง CPU เพื่อประมวลผลเมื่อ CPU ประมวลผลแล้วจะส่งให้ส่วนของหน่วยเอาต์พุต เพื่อให้อุปกรณ์ทำงานตามโปรแกรมที่กำหนดไว้ สัญญาณอินพุตจากภายนอกที่เป็นสวิตช์และตัวตรวจจับชนิดต่างๆ จะถูกแปลงให้เป็นสัญญาณที่เหมาะสมถูกต้อง ไม่ว่าจะเป็นสัญญาณไฟฟ้า AC หรือ DC เพื่อส่งให้ CPU ดังนั้น สัญญาณเหล่านี้จึงต้องมีความถูกต้อง ไม่เช่นนั้นแล้ว CPU จะเสียหายได้

สัญญาณอินพุตที่ดีจะต้องมีคุณสมบัติและหน้าที่ดังนี้

1. ทำให้สัญญาณเข้า ได้ระดับที่เหมาะสมกับ PLC
2. การส่งสัญญาณระหว่างอินพุตกับ CPU จะติดต่อกันด้วยลำแสง ซึ่งจะอาศัยอุปกรณ์ประเภทโฟโตทรานซิสเตอร์เพื่อต้องการแยกสัญญาณ (Isolate) ทางไฟฟ้าให้ออกจากกัน เป็นการป้องกันไม่ให้ CPU เสียหายเมื่ออินพุตเกิดลัดวงจร
3. หน้าสัมผัสจะต้องไม่สัมผัสต่อเนื่อง

ส่วนของเอาต์พุตจะทำหน้าที่รับค่าสถานะที่ได้จากการประมวลผลของ CPU แล้วนำค่าเหล่านี้ไปควบคุมอุปกรณ์ทำงานภายนอกเช่น รีเลย์ มอเตอร์ วาล์วหรือหลอดไฟ เป็นต้น

2.1.1.4 แหล่งจ่ายไฟ (Power Supply)

ทำหน้าที่เปลี่ยนสัญญาณจากแหล่งจ่ายภายนอก เช่น 220 VAC หรือ 24 VDC ให้เป็นแรงดันไฟฟ้าขนาด 5 VDC เพื่อจ่ายไฟให้กับอุปกรณ์อิเล็กทรอนิกส์ภายในหน่วยประมวลผล หน่วยอินพุต/เอาต์พุตหรืออุปกรณ์อื่นๆ ใน PLC การทำงานของวงจรภายในแหล่งจ่ายไฟเป็นแบบวงจรสวิตซ์ ทำให้สามารถจ่ายกระแสไฟฟ้าให้กับโหลดได้คงที่ นอกจากนี้ยังผลิตแรงดันไฟฟ้าขนาด 24 VDC เพื่อจ่ายให้กับอุปกรณ์อินพุตภายนอก PLC เช่น สวิตช์และเซนเซอร์ต่างๆ

2.1.1.5 อุปกรณ์ต่อร่วม (Peripheral Device)

อุปกรณ์ต่อร่วม เป็นอุปกรณ์แบบต่างๆ ที่อำนวยความสะดวกในการพัฒนาโปรแกรม สามารถใช้ร่วมกับ PLC ได้ ซึ่งทำหน้าที่ดังนี้

1. ป้อนโปรแกรมเข้าไปในหน่วยความจำของระบบ
2. ใช้ในการแก้ไข (Debug) ของระบบ
3. ใช้ในการเก็บรักษาโปรแกรม
4. ใช้ในการพิมพ์โปรแกรม
5. ใช้แสดงสถานะการควบคุม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.2 พื้นที่หน่วยความจำของ PLC

แบ่งหน่วยความจำเป็น 2 ส่วน คือ

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

หน้าที่ของหน่วยความจำข้อมูลของ PLC หน่วยความจำข้อมูลแต่ละพื้นที่ จะมีหน้าที่แตกต่างกัน

- Internal Relay (IR) แบ่งพื้นที่ออกเป็นส่วนต่างๆ ได้แก่ ส่วนของอินพุต/เอาต์พุตเป็นพื้นที่ใช้กำหนดตำแหน่งให้กับหน่วยอินพุต/เอาต์พุตแบบต่างๆ ที่นำมาติดตั้ง เช่น หน่วยอินพุต/เอาต์พุตแบบธรรมดา หน่วยอินพุต/เอาต์พุตแบบพิเศษ เป็นต้น และนอกจากนั้นยังมีส่วนของพื้นที่ใช้งานซึ่งใช้เป็นอินพุต/เอาต์พุตภายใน PLC เรียกว่า เป็นบิตใช้งานซึ่งสามารถใช้ได้อย่างอิสระใน โปรแกรม แต่ไม่สามารถใช้อ้างอิงตำแหน่งที่ต่อกับอุปกรณ์อินพุต/เอาต์พุตจริง

- Special Relay (SR) ใช้เก็บสถานะของเฟลกต่างๆ ของบิตควบคุมที่ใช้แสดงในรูปแบบ สถานการณ์ทำงานของ PLC สัญญาณ (Clock Pulse) และค่าความผิดพลาดต่างๆ

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

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

- Auxiliary Relay (AR) area สำหรับพื้นที่หน่วยความจำในส่วนนี้จะใช้สำหรับเก็บค่าเวลา เฟลกสถานะต่างๆ และบิตควบคุมการทำงานใน PLC บางรุ่นใช้พื้นที่หน่วยความจำส่วนนี้ในการเก็บค่า Real Time Clock ซึ่งเป็นพื้นที่ที่ปฏิทินนาฬิกาใช้เก็บค่าเวลา/วัน/เดือน/ปี และวันที่ เป็นค่าเวลาปัจจุบันที่สามารถนำไปใช้แทนฟังก์ชันไทม์เมอร์/เคาน์เตอร์ได้ เช่น ต้องการควบคุมการเปิดปิดปั๊มน้ำในช่วงเวลาที่กำหนด แทนที่จะใช้ฟังก์ชันไทม์เมอร์/เคาน์เตอร์หลายๆ ตัวในการตั้งเวลาสามารถนำค่าเวลา Real Time Clock ใช้แทนได้

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

- Timer/Counter (TC) Area ใช้ตั้งค่าเวลาและค่าการนับให้กับ PLC สำหรับ PLC บางรุ่นจะใช้พื้นที่หน่วยความจำในส่วนเดียวกัน เช่น ถ้าใช้ไทม์เมอร์หมายเลข 0 แล้ว จะไม่สามารถใช้

เคาน์เตอร์หมายเลข 0 ได้อีก แต่ PLC บางรุ่นมีการแยกพื้นที่หน่วยความจำในส่วนของไทม์เมอร์ และเคาน์เตอร์ออกจากกัน ดังนั้นจึงสามารถใช้ฟังก์ชันไทม์เมอร์และเคาน์เตอร์หมายเลขเดียวกันได้

- Data Memory (DM) Area เป็นหน่วยความจำที่ใช้ในการเก็บข้อมูลแบ่งเป็น 2 ส่วน คือ ส่วนที่สามารถอ่าน (Read) และเขียน (Write) ข้อมูลลงไปได้สามารถเปลี่ยนแปลงค่าเพื่อใช้งาน PLC ได้ พื้นที่หน่วยความจำส่วนนี้ยังแบ่งออกเป็นหน่วยย่อยอีกหลายส่วน เช่น ส่วนของ Special I/O unit Area จะใช้งานในกรณีที่มีการใช้หน่วยอินพุต/เอาต์พุตแบบพิเศษ โดยการกำหนดหมายเลขของหน่วยความจำที่ใช้งานโดยปรับตั้งค่าทางด้านหน้าของหน่วยอินพุต/เอาต์พุตแบบพิเศษนั้น เช่น

Unit Number หมายเลข 0 จะใช้งานพื้นที่ DM 1000 ถึง DM 1099

Unit Number หมายเลข 1 จะใช้งานพื้นที่ DM 1100 ถึง DM 1199

พื้นที่อีกส่วนหนึ่งคือส่วนที่อ่านค่าได้อย่างเดียว (Read Only) พื้นที่ส่วนนี้ใช้ในการเก็บค่าใช้งานต่างๆ ของ PLC และแบ่งออกเป็นพื้นที่ส่วนต่างๆ ได้แก่ System setting PC setup และ Expansion DM เป็นต้น

2.1.3 ความสามารถของ PLC

PLC สามารถควบคุมงานได้ 3 ลักษณะ คือ

2.1.3.1. งานที่ทำตามลำดับก่อนหลัง (Sequence Control)

การทำงานตามลำดับก่อนหลัง เป็นการควบคุมที่ใช้สัญญาณที่ได้จากตัวตรวจจับ (Sensor) ตามลักษณะงานที่กำหนด สั่งให้กระบวนการทำงานตามการเรียงลำดับงานของระบบหรือการออกแบบควบคุม เช่น

1. การทำงานของระบบรีเลย์
2. การทำงานของไทม์เมอร์/เคาน์เตอร์
3. การทำงานในระบบกึ่งอัตโนมัติ ระบบอัตโนมัติหรืองานที่เป็นกระบวนการทำงานของเครื่องกลต่างๆ

2.1.3.2. งานควบคุมสมัยใหม่ (Sophisticated Control)

เป็นการประยุกต์นำเอา PLC มาใช้ควบคุมอุปกรณ์ต่างๆ ซึ่งมีหลักการทำงานที่ซับซ้อนมากยิ่งขึ้น เช่น

1. การทำงานทางคณิตศาสตร์ เช่น บวก ลบ คูณ หาร
2. การควบคุมแบบอนาล็อก (Analog Control) เช่น การควบคุมอุณหภูมิ
3. การควบคุม PID (Proportional Integral Derivative)
4. การควบคุมเซอร์โวมอเตอร์ (Servo Motor Control)
5. การควบคุม Stepper Motor

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์และใช้เฉพาะเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.3.3. งานที่ควบคุมเกี่ยวกับงานอำนวยการ (Supervisory Control)

เป็นการนำเอา PLC มาใช้ควบคุมดูแลระบบและอุปกรณ์ที่ใช้ในสำนักงานหรือโรงงาน เพื่อความสะดวกมากยิ่งขึ้น เช่น

1. งานสัญญาณเตือน (Alarm) และ Process Monitoring
2. Fault Diagnostic and Monitoring
3. งานต่อร่วมกับ Computer (RS-232C/RS-485)
4. Printer / ASCII Interfacing
5. งานควบคุมอัตโนมัติในโรงงานอุตสาหกรรม
6. LAN (Local Area Network)
7. WAN (Wide Area Network)
8. FS FMS และ CIM เป็นต้น

2.1.4 การสื่อสารข้อมูลแบบอนุกรม (Serial)

ในการติดต่อแบบนี้ ข้อมูลจะถูกส่งออกมาทีละบิตระหว่างจุดส่งและจุดรับ ซึ่งการส่งข้อมูลแบบนี้จะช้ากว่าการส่งแบบขนาน แต่ใช้ตัวกลางในการสื่อสารเพียงช่องเดียวหรือสายเพียงคู่เดียว ทำให้ค่าใช้จ่ายสำหรับการสื่อสารแบบอนุกรมถูกกว่าแบบขนาน การส่งแบบนี้ใช้สำหรับส่งในระยะทางไกลมากกว่า 100 ฟุต ข้อมูลจากจุดส่งจะต้องถูกเปลี่ยนให้เป็นอนุกรมก่อนแล้วค่อยทยอยส่งออกทีละบิตไปยังจุดรับ ที่จุดรับจะต้องมีรูปแบบในการเปลี่ยนข้อมูลที่ส่งมาให้เป็นแบบขนานในการแปลงต้องมีรูปแบบที่เหมาะสมเพื่อป้องกันการผิดพลาดของข้อมูล

2.1.4.1 มาตรฐานในการสื่อสารข้อมูลแบบอนุกรม

การสื่อสารข้อมูลแบบอนุกรมที่ใช้งานอยู่ในปัจจุบันนั้น ได้มีการกำหนดมาตรฐานการรับส่งข้อมูลไว้หลายแบบด้วยกัน แต่ที่ได้รับความนิยมมาใช้งานอย่างมาก คือ การสื่อสารข้อมูลแบบอนุกรมตามมาตรฐาน RS-232C และที่มาตรฐานนี้เป็นที่นิยมเนื่องจากเป็นระบบการสื่อสารข้อมูลที่ใช้ในเครื่องไมโครคอมพิวเตอร์ IBM PC ซึ่งเป็น Computer ที่มีใช้อย่างแพร่หลายมาในอดีตจนถึงปัจจุบัน มาตรฐานการสื่อสารนี้ในการออกแบบเบื้องต้น ได้ออกแบบสำหรับการเชื่อมต่อกับเครื่องโมเด็ม ซึ่งเป็นอุปกรณ์ที่ใช้การสื่อสารข้อมูลระหว่าง Computer ผ่านทางสายโทรศัพท์ ซึ่งทำให้อัตราการรับส่งข้อมูลถูกจำกัดให้มีค่าที่ค่อนข้างต่ำ มาตรฐาน RS-232C นี้ได้ออกแบบให้มีโครงสร้างการสื่อสารเป็นแบบจุดต่อจุดเท่านั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.4.2 ข้อตกลงในการสื่อสารของ PLC

รูปแบบของข้อตกลงในการสื่อสารของ PLC โดยทั่วไปจะเป็นลักษณะการตอบกันระหว่าง PLC กับอุปกรณ์ภายนอก ซึ่งอุปกรณ์ภายนอกมักจะเป็นฝ่ายถามก่อน โดยการส่งบล็อกคำสั่ง (Command Block) ออกไป จากนั้น PLC จะทำการตรวจสอบแล้วส่งบล็อกตอบกลับมา (Response Block) สำหรับรูปแบบของบล็อกจะมีลักษณะดังนี้

1. รูปแบบของบล็อก (Block Format)

ลักษณะของบล็อกของ PLC แต่ละผู้ผลิตจะแตกต่างกันไปแต่จะมีพื้นฐานเดียวกัน ดังตัวอย่างบล็อกคำสั่งต่อไปนี้ ดังแสดงในรูปที่ 2.3

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

รูปที่ 2.3 รูปแบบของบล็อก (Block Format)

- 1.1 ส่วนของ Unit Number คือหมายเลขเครื่องในการเชื่อมต่อที่เป็นโครงข่ายแบบหลายจุดซึ่งเครื่องควบคุมที่เชื่อมต่ออยู่ในระบบจะมีมากกว่า 1 เครื่อง การกำหนดความต้องการส่งข้อมูลให้กับเครื่องควบคุมตัวใด เช่น @05 คือหมายเลขเครื่องที่ 05
- 1.2 ส่วนของ HEADER เป็นส่วนของคำสั่งหลักที่จะกำหนดว่าต้องกระทำกับข้อมูลส่วนใด เช่น ต้องการอ่านข้อมูลของอินพุต ต้องการเขียนข้อมูลให้กับตัวนับ เป็นต้น
- 1.3 ส่วนของ Text เป็นส่วนของข้อมูล เช่น คำที่อ่านได้จากอินพุตหรือค่าที่จะต้องเขียนลงในพื้นที่ต่างๆ
- 1.4 ส่วนของ FCS เป็นส่วนของการควบคุมความผิดพลาดของข้อมูลซึ่งได้จากการคำนวณ
- 1.5 ส่วนของ TERMINAL เป็นส่วนที่ปิดท้ายบอกให้ทราบว่าจบบล็อก และมักจะติดตามด้วยรหัสของ Carrier Return (CR)

2. การคำนวณ

FCS หรือ Frame Check Sequence เป็นสิ่งที่ใช้ในการตรวจสอบความผิดพลาดของการสื่อสาร เป็นการเปลี่ยนข้อมูล 8 บิต เป็น 2 ตัวอักษรของข้อมูล ASCII แล้วนำมาทำการ Exclusive Or (XOR) โดยเริ่มจาก @ จนถึงตัวอักษรตัวสุดท้ายของ Text

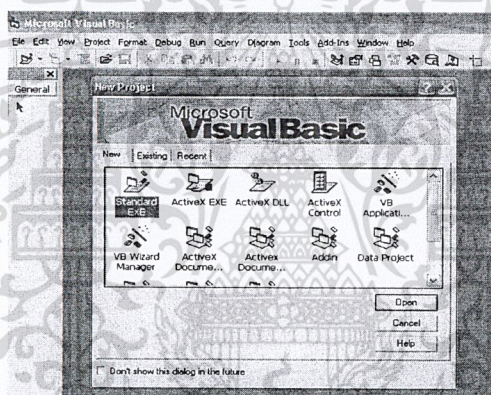
เอกสารนี้เป็นเอกสารของบริษัทฯ ไม่ควรเผยแพร่โดยไม่ได้รับอนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2 โปรแกรม Visual Basic 6.0

2.2.1 การเขียนโปรแกรม Visual Basic 6.0 ขั้นพื้นฐาน

2.2.1.1 ที่มาและลักษณะของโปรแกรม Visual Basic 6.0

Visual Basic 6.0 เป็นเครื่องมือที่ใช้พัฒนาโปรแกรมแบบ Visual Programming ซึ่ง Visual Programming เป็นวิธีการเขียนโปรแกรมที่มีเครื่องมือช่วยพัฒนาโปรแกรมได้ง่าย โดยโปรแกรมที่สร้างจะมีลักษณะเหมือนคอนออกแบบหน้าจอ วิธีการพัฒนาโปรแกรมก็ง่าย เพียงออกแบบหน้าจอที่ต้องการกำหนดคุณสมบัติและเขียนโค้ดกำกับ ซึ่งจะช่วยให้การพัฒนาโปรแกรมสามารถทำได้อย่างรวดเร็ว สำหรับการใช้งานโปรแกรม Visual Basic 6.0 นั้นไม่ได้จำกัดตัวเองอยู่เพียงแค่นบน Computer ที่ใช้ Windows เท่านั้นแต่ยังขยายความสามารถไปยังการใช้งานบนอินเทอร์เน็ต รวมทั้งอุปกรณ์ที่ใช้เทคโนโลยีไร้สาย



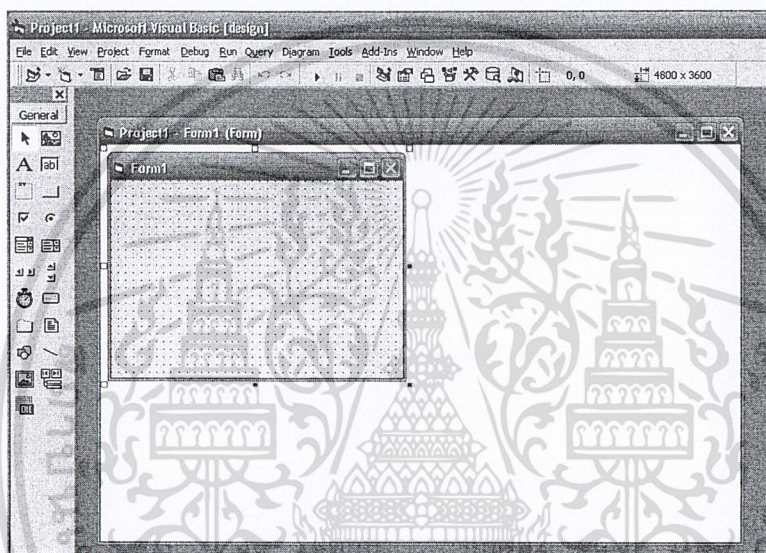
รูปที่ 2.4 หน้าจอโปรแกรม Visual Basic 6.0

2.2.1.2 รายละเอียดของส่วนประกอบต่างๆ ของหน้าจอ

1. Menu Bar เก็บคำสั่งที่สามารถใช้งานได้ทั้งหมดใน Visual Basic 6.0 ประกอบด้วยเมนูทำงาน File View และ Windows
2. Toolbar ประกอบด้วยปุ่มคำสั่งต่างๆ ที่ช่วยให้งานคำสั่งของ Visual Basic 6.0 ได้้อย่างรวดเร็ว
3. Toolbox เป็นกล่องเก็บ ActiveX Control ซึ่งจะนำมาประกอบเป็นส่วนต่างๆ ของแอปพลิเคชัน
4. Project Explorer เป็นเครื่องมือที่ใช้ควบคุมการทำงานของโปรเจกต์
5. Properties windows เป็นส่วนที่กำหนดหรือพอร์ติให้กับออปเจกต์ต่างๆ

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้เพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6. Form Layout เป็นลักษณะคร่าวๆของฟอร์มที่ได้จากการรันแอปพลิเคชัน ทำให้สามารถทราบตำแหน่งที่ปรากฏบนจอภาพเมื่อแอปพลิเคชันทำงาน
7. Form Designer ซึ่งเป็นส่วนที่สามารถเห็นในขณะออกแบบแอปพลิเคชันของ Visual Basic 6.0 ซึ่งออกแบบลักษณะของแอปพลิเคชันผ่านฟอร์มดีไซน์เนอร์
8. Code Windows เป็นส่วนสำหรับการเขียนโปรแกรมเพื่อควบคุมการทำงานของแอปพลิเคชัน



รูปที่ 2.5 องค์ประกอบสำคัญในโปรแกรม Visual Basic 6.0

2.2.1.3 จุดเด่นของ Visual Basic 6.0

1. สร้างแอปพลิเคชันได้ง่ายและรวดเร็ว

Visual Basic 6.0 มีเครื่องมือที่ช่วยให้สร้างแอปพลิเคชันได้อย่างรวดเร็วและง่ายดาย เพื่อลดเวลาในการสร้างแอปพลิเคชันให้สั้นลง ซึ่งเรียกรูปแบบนี้ว่า Rapid Application Development หรือ RAC โดยมีการจัดการที่โปรแกรมเมอร์ต้องทำซ้ำๆ ออกไป เหลือเฉพาะที่เป็นปัญหาของงานจริงๆ ซึ่งต้องเขียนโปรแกรมจัดการปัญหานั้น ส่วนเรื่องอื่นๆ เหลือให้ Visual Basic 6.0 จัดการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. การเขียนโปรแกรมที่ง่ายต่อการเรียนรู้

ภาษาที่ใช้ใน Visual Basic 6.0 นั้นเป็นภาษาที่สามารถเข้าใจง่ายเนื่องจากใกล้เคียงกับภาษาที่ใช้งานปกติ สื่อความหมายและเข้าใจง่ายกว่าภาษาโปรแกรมอื่นๆ ทำให้สามารถทำความเข้าใจกับการเขียนโปรแกรมได้อย่างรวดเร็ว

3. รวมเครื่องมืออำนวยความสะดวกในการเขียนโปรแกรม

Visual Basic 6.0 มีเครื่องมือที่ช่วยในการเขียนโปรแกรมให้ง่ายและรวดเร็ว โดยมีเครื่องมือช่วยให้ไม่ต้องจดจำไวยากรณ์ภาษาที่ยุ่งยาก มีการตรวจสอบความถูกต้องตามหลักภาษาของโปรแกรมที่เขียน มีการแยกแยะส่วนของโปรแกรมอย่างเป็นระเบียบ ทำให้งานของโปรแกรมเมอร์นั้นคล่องตัวมาก นอกจากนี้จะมีเครื่องมือช่วยในการเขียนโปรแกรมแล้วยังมีเครื่องมือที่ใช้ในการทดสอบ แก้ไข โปรแกรม (Debugger) ที่เขียนขึ้นมาทดสอบความถูกต้องของการทำงาน มีระบบขอความช่วยเหลือ (Online Help) ไว้อ้างอิงและขอความช่วยเหลือในจุดที่สงสัย

เครื่องมือทั้งหมดถูกจัดรวมไว้ในสภาพแวดล้อมการทำงานเดียวกัน เรียกย่อๆ ว่า IDE ซึ่งย่อมาจาก Integral Development Environment ทำให้เรียกใช้งานได้สะดวกตั้งแต่เขียนโปรแกรม ทดสอบ แก้ไข สร้างชุดติดตั้ง รวมทั้งระบบขอความช่วยเหลือ ซึ่งสามารถเพิ่มเติมเครื่องมือชนิดใหม่ๆ เข้าไปได้เรื่อยๆ หรือถอดเครื่องมือที่ไม่จำเป็นต้องใช้ได้

2.2.1.4 รูปแบบการพัฒนาแอปพลิเคชันกับ Visual Basic

Visual Basic 6.0 นอกจากจะช่วยให้สามารถสร้างแอปพลิเคชันบน Windows ได้ง่ายและรวดเร็วแล้ว ยังมีรูปแบบที่ Visual Basic 6.0 สามารถสร้างขึ้นมาได้อีก คือ

1. พัฒนาแอปพลิเคชันกับ ActiveX Control

ActiveX เป็นเทคโนโลยีที่ช่วยลดงานที่ซ้ำซ้อนของการเขียนโปรแกรม เช่น การเขียนโปรแกรมเพื่อรับค่าข้อมูลของผู้ใช้ โปรแกรมเมอร์จะสนใจเฉพาะการจัดการกับข้อมูลที่ผู้ใช้ป้อนเข้ามาเท่านั้น ส่วนที่เหลือไม่ว่าจะเป็นช่องรับข้อความและปุ่มต่างๆ จะใช้ ActiveX Control จัดการ ทำให้สามารถเขียนโปรแกรมได้อย่างรวดเร็วมากยิ่งขึ้น

นอกจากจะลดความซับซ้อนลงแล้วการใช้ ActiveX Control ในการเขียนโปรแกรมยังช่วยให้โปรแกรมนั้นตั้งอยู่บนมาตรฐานเดียวกัน ทำให้การบำรุงรักษาง่าย

2. สร้างแอปพลิเคชันที่ใช้งานกับฐานข้อมูล

เป็นแอปพลิเคชันที่มีการใช้งานมากที่สุดเพราะระบบต่างๆ เช่น ร้านค้า คลังสินค้า ระบบบัญชี ระบบบริหารงานบุคคลหรือแม้แต่ E-Commerce ทั้งหมดต่างก็ต้องมีส่วนที่ติดต่อกับฐานข้อมูล Visual Basic 6.0 ได้ช่วยให้การสร้างแอปพลิเคชันกับฐานข้อมูลสามารถทำได้โดยง่าย

เพราะมีเครื่องมือต่างๆ ที่คอยให้ความสะดวกในการเขียนโปรแกรมเพื่อใช้งานข้อมูลจากฐานข้อมูล ไม่ว่าจะเป็นกรณใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ซึ่งไม่จำกัดด้วยว่าจะเห็นข้อมูลแบบใด ทั้งฐานข้อมูลส่วนบุคคล ฐานข้อมูลผ่านเครือข่ายหรือ ฐานข้อมูลผ่านอินเทอร์เน็ต

3. สร้างแอปพลิเคชันแบบใหม่กับอินเทอร์เน็ต

อินเทอร์เน็ตเริ่มมีความสำคัญกับชีวิตของคนที่ใช้ไอทีมากขึ้นทุกวัน ซึ่ง Visual Basic 6.0 สามารถสร้างแอปพลิเคชันเพื่อรองรับการทำงานร่วมกับอินเทอร์เน็ตได้ด้วย

จุดเด่นของ Visual Basic 6.0 อีกข้อหนึ่งคือการที่สามารถนำแอปพลิเคชันปกติที่ทำงานกับ Computer สามารถดัดแปลงใช้งานกับอินเทอร์เน็ตได้อย่างง่าย ทำให้ไม่ต้องเปลี่ยนแปลงรูปแบบงานเดิม

2.2.1.5 เริ่มสร้างแอปพลิเคชันด้วย Visual Basic 6.0

1. ออกแบบแอปพลิเคชัน

ต้องทราบให้แน่ชัดก่อนว่าแอปพลิเคชันที่จะทำการสร้างนั้น จะนำไปใช้ประโยชน์อะไรต้องมีความสำคัญอะไรบ้าง และต้องมีลักษณะเป็นอย่างไร ซึ่งจำเป็นอย่างยิ่งที่ต้องคิดให้รอบคอบและเขียนออกมาให้ชัดเจน โดยอาจจะร่างง่ายๆ ในกระดาษ

2. ตกแต่งลักษณะแอปพลิเคชัน

เป็นการตกแต่งลักษณะของแอปพลิเคชันที่ได้ออกแบบไว้แล้วพร้อมๆ กับการคำนวณค่าหรือฟอร์ตต่างๆ ให้กับคอนโทรลแต่ละตัวในแอปพลิเคชัน

3. เขียนโค้ดกับแอปพลิเคชัน

เขียนโปรแกรมเพื่อควบคุมการทำงานต่างๆ โดยใช้การเขียนโปรแกรมแบบ Event Driven Programming ซึ่งจะเป็นการเขียนโค้ดเพื่อรองรับกับเหตุการณ์ต่างๆ ที่จะเกิดขึ้นกับคอนโทรลในแอปพลิเคชัน

4. ทดสอบการทำงานของแอปพลิเคชัน

เมื่อเขียนโค้ดเสร็จแล้ว ก็ทำการทดสอบการทำงานของแอปพลิเคชันที่สร้างขึ้น ซึ่งประกอบไปด้วยคอนโทรลต่างๆ

5. บันทึกเก็บไว้ใน Computer

หลังจากทดสอบจนแน่ใจว่าแอปพลิเคชันที่สร้างขึ้นทำงานได้ถูกต้อง จึงเก็บบันทึกไว้ ซึ่งสามารถแก้ไขและเพิ่มเติมความสามารถอื่นๆ ได้ภายหลัง

6. สร้างไฟล์ .EXE (Make)

เมื่อสร้างแอปพลิเคชันเสร็จแล้ว หากต้องการนำแอปพลิเคชันที่สร้างขึ้นให้เรียกใช้งานได้เอง โดยไม่ต้องเรียกผ่าน Visual Basic 6.0 หรือต้องการนำไปใช้งานใน Computer เครื่องอื่น สามารถทำได้โดยการสร้างไฟล์เอ็กซ์คิวต์ (ไฟล์ที่มีนามสกุลเป็น .EXE)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.2 การสื่อสารกับ Communication Port

Visual Basic 6.0 ไม่สามารถจัดการกับ Hardware ได้โดยตรงเหมือนภาษา C หรือภาษาอื่นๆ อย่างไรก็ดี Visual Basic 6.0 ได้เตรียม Tool และ Control ต่างๆ สำหรับจัดการกับ Hardware โดยเฉพาะไว้ให้ผู้เขียนโปรแกรมทำที่จำเป็น ในการทำงานกับพอร์ตสื่อสารบน Computer ซึ่ง Visual Basic 6.0 มี Control ชื่อ Microsoft Comm Control สำหรับการรับส่งข้อมูลทางพอร์ตอนุกรม

2.2.2.1 การใช้ MsComm Control

MsComm จัดเตรียมทางเลือกเอาไว้ 2 ทางเพื่อความสะดวกในการสื่อสารข้อมูล ทางแรกคือการสื่อสารข้อมูลที่กระตุ้นด้วยเหตุการณ์ (Event-Driven Communication) เป็นรูปแบบการใช้งานที่มีประสิทธิภาพมากสำหรับตอบสนองแบบทันทีทันใด เช่น เมื่อตัวอักษรถูกส่งมาที่พอร์ตอนุกรมหรือเกิดการเปลี่ยนแปลงที่ขา Data Carrier Detect (DCD) หรือขา Request To Send เหตุการณ์ OnComm ของ MsComm จะสามารถตรวจจับสัญญาณนั้นได้ทันที ส่วนทางเลือกที่สองเป็นการคอยตรวจสอบค่าเหตุการณ์และความผิดพลาดที่เกิดขึ้น ด้วยค่าที่เปลี่ยนแปลงในคุณสมบัติ CommEvent หลังจากให้โปรแกรมทำงานฟังก์ชันต่างๆ ไปเรียบร้อยแล้ว ซึ่งวิธีนี้ใช้งานได้ดีในกรณีที่โปรแกรมขนาดเล็ก

2.2.2.2 CommPort

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

Object.CommPort [=value]

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

2.2.2.3 Setting

ใช้ในการกำหนดและอ่านค่าอัตราบอดเรต พาริตี จำนวนของบิตข้อมูล จำนวนบิตปิดท้าย
รูปแบบการใช้งาน

เอกสาร Object.setting [=value]ไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ค่า value มีชนิดของข้อมูลเป็นแบบ String มีรูปแบบเป็น “BBBB,P,D,S” โดย BBBB เป็นค่าอัตราบอดเรต P เป็นค่าพาริตี D เป็นจำนวนของบิตข้อมูลและ S เป็นจำนวนของบิตปิดท้ายปกติแล้วค่านี้ถูกกำหนดไว้เป็น “9600,N,8,1”

ค่าบอดเรตมาตรฐานที่ใช้กับ MsComm มีดังนี้

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

ค่ามาตรฐานในการกำหนดค่าพาริตีดังนี้

สัญลักษณ์	รายละเอียด
E	พาริตีคู่ (EVEN)
M	ลอจิก “1” (MARK)
N	ไม่ใช่ (ค่าปกติ)
O	พาริตีคี่ (ODD)
S	ลอจิก “0” (SPACE)

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

ค่าที่ระบุจำนวนบิตปิดท้ายมี 3 ค่า คือ 1 (เป็นค่าปกติ) 1.5 และ 2

2.2.2.4 PortOpen

ใช้ในการกำหนดและอ่านค่าสถานะของพอร์ตอนุกรม เพื่อปิดและเปิดพอร์ตอนุกรม

รูปแบบการใช้งาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
Object.PortOpen [=value]
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ค่า value มีชนิดข้อมูลเป็นแบบบูลีนคือ True กับ False โดย True หมายถึงการเปิดพอร์ตอนุกรมและ False หมายถึงการปิดพอร์ตอนุกรม สำหรับการปิดพอร์ตอนุกรม สำหรับการจะปิดพอร์ตนั้นจะมีการเคลียร์บัฟเฟอร์รับข้อมูลและบัฟเฟอร์ส่งข้อมูลด้วย คอนโทรล MsComm ตรวจสอบให้แน่ใจก่อนว่าคุณสมบัติ CommPort นั้นได้ทำการกำหนดตำแหน่งของพอร์ตอนุกรมไว้ถูกต้องหรือไม่ มิเช่นนั้น MsComm จะแสดงผิดพลาด error 68 แจ้งแก่ผู้ใช้งานหรือพอร์ตอนุกรมนั้นถูกเปิดเอาไว้แล้ว โปรแกรมก็จะแจ้งข้อผิดพลาดออกเช่นกัน

ถ้าคุณสมบัติ DTREnable หรือ RTSEnable ถูกกำหนดให้เป็น True ก่อนที่จะทำการเปิดพอร์ตค่าคุณสมบัติของ DTREnable หรือ RTSEnable จะถูกเซตเป็น False หลังจากปิดพอร์ต แต่ถ้าเซตเป็น False หลังจากปิดโปรแกรมแล้ว ค่าที่กำหนดไว้จะเป็นค่าเดิม

2.2.2.5 การเปิดพอร์ตขนาน

การเปิดพอร์ตสื่อสารด้วย MsComm Control สามารถทำได้ง่าย ๆ ด้วยการกำหนด Properties ดังนี้

MSComm 1.CommPort = 4

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

MSComm 1.PortOpen = True

Input คือ การอ่านข้อมูลจากบัฟเฟอร์พร้อมทั้งลบข้อมูลในบัฟเฟอร์ด้านรับทิ้ง ซึ่งสามารถอ่านค่าได้เฉพาะในขณะที่ทำงานเท่านั้น รูปแบบการใช้งาน คือ

Object.Input[=value]

ทุกครั้งที่มีการใช้คุณสมบัติ Input ในการอ่านข้อมูลจากบัฟเฟอร์ด้านรับเข้านั้น จำนวนของตัวอักษรที่อ่านได้จะถูกกำหนดลงในคุณสมบัติ InputLen ทั้งนี้ซึ่งถ้าหากกำหนดให้คุณสมบัติ InputLen มีค่าเท่ากับ 0 ก็จะหมายถึงการกำหนดให้คุณสมบัติของ Input อ่านข้อมูลทั้งหมดจากบัฟเฟอร์ด้านรับเข้านั่นเอง ซึ่งชนิดของข้อมูลที่อ่านโดยคุณสมบัติ Input จะเป็นข้อมูลแบบข้อความหรือไบนารี ก็ขึ้นอยู่กับค่าของการกำหนดค่าของคุณสมบัติ InputMode

Output คือ การส่งข้อมูลไปยังบัฟเฟอร์ด้านส่งออก ซึ่งสามารถกำหนดค่าได้เฉพาะในขณะที่ทำงานเท่านั้น รูปแบบการใช้งาน คือ

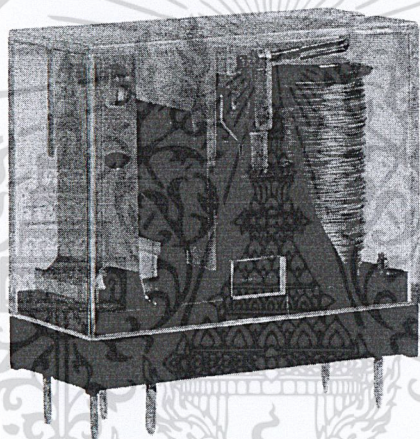
Object.Output[=value]

สำหรับชนิดของข้อมูลที่ถูกส่งโดยคุณสมบัติของ Output จะเป็นข้อมูลแบบข้อความหรือไบนารี ก็ขึ้นกับการกำหนดค่าของคุณสมบัติ InputMode ดังที่กล่าวมาข้างต้น
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่สามารถนำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3 รีเลย์

รีเลย์ (Relay) คือ อุปกรณ์อิเล็กทรอนิกส์ที่ทำหน้าที่ตัด-ต่อวงจร มีหลักการทำงานคล้ายกับ ขดลวดแม่เหล็กไฟฟ้าหรือ โซลินอยด์ (Solenoid) ซึ่งมีลักษณะดังรูปที่ 2.6 รีเลย์ถูกใช้ในการควบคุม วงจรไฟฟ้าได้หลากหลาย รีเลย์เป็นสวิตช์ควบคุมที่ทำงานด้วยไฟฟ้า แบ่งออกตามลักษณะการใช้งานได้ 2 ประเภท คือ

1. รีเลย์กำลัง (Power Relay) หรือมักเรียกกันว่า คอนแทกเตอร์ (Contactor or Magnetic contactor) ใช้ในการควบคุมไฟฟ้ากำลัง มีขนาดใหญ่กว่ารีเลย์ธรรมดา
2. รีเลย์ควบคุม (Control Relay) มีขนาดเล็ก กำลังไฟฟ้าต่ำ ใช้ในวงจรควบคุมทั่วไปที่มี กำลังไฟฟ้าไม่มากนัก รีเลย์ควบคุมบางที่เรียกกันง่ายๆ ว่า “รีเลย์”



รูปที่ 2.6 รีเลย์

2.3.1 หน้าที่ของคอนแทกเตอร์ (Contactor)

หน้าที่ของคอนแทกเตอร์ คือ การใช้กำลังไฟฟ้าจำนวนน้อยเพื่อไปควบคุมการตัดต่อ กำลังไฟฟ้าจำนวนมาก การใช้คอนแทกเตอร์ทำให้สามารถควบคุมกำลังไฟฟ้าในตำแหน่งอื่นๆ ของ ระบบไฟฟ้าได้ สายไฟควบคุมให้รีเลย์กำลังหรือคอนแทกเตอร์ทำงาน เป็นสายไฟฟ้าขนาดเล็กต่อ เข้ากับสวิตช์ควบคุมและคอยล์ของคอนแทกเตอร์ กำลังไฟฟ้าที่ป้อนเข้าคอยล์อาจจะเป็นไฟฟ้า กระแสตรงหรือไฟฟ้ากระแสสลับก็ได้ ขึ้นอยู่กับการออกแบบการใช้คอนแทกเตอร์ทำให้สามารถ ควบคุมวงจรจากระยะไกล (Remote) ได้ ซึ่งทำให้เกิดความปลอดภัยกับผู้ปฏิบัติงานในการควบคุม กำลังไฟฟ้า

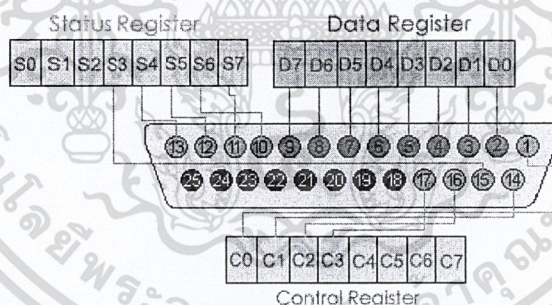
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3.2 ข้อคำนึงในการใช้งานรีเลย์ทั่วไป

1. แรงดันไฟฟ้าใช้งานหรือแรงดันไฟฟ้าที่ทำให้รีเลย์ทำงานได้ หากคู่มือรีเลย์จะระบุค่าแรงดันไฟฟ้าใช้งานไว้ (หากใช้ในงานอิเล็กทรอนิกส์ ส่วนมากจะใช้แรงดันไฟฟ้ากระแสตรงในการใช้งาน) เช่น 12 VDC คือต้องใช้แรงดันไฟฟ้าที่ 12 VDC เท่านั้นหากใช้มากกว่านี้ ขดลวดภายในตัวรีเลย์อาจขาดได้หรือหากใช้แรงดันไฟฟ้าต่ำกว่ามาก รีเลย์จะไม่ทำงาน
2. การใช้งานกระแสผ่านหน้าสัมผัส ซึ่งที่ตัวรีเลย์จะระบุไว้ เช่น 10 A 220 VAC คือหน้าสัมผัสของรีเลย์นั้นสามารถทนกระแสได้ 10 A ที่ 220 VAC แต่การใช้ก็ควรระวังที่ระดับกระแสต่ำกว่านี้จะเป็นการดีกว่า เพราะถ้ากระแสผ่านหน้าสัมผัสของรีเลย์จะละลายเสียหายได้

2.4 การสื่อสารผ่านพอร์ตขนาน (Parallel Port)

พอร์ตขนาน (Parallel Port) เดิมเรียกว่า Printer Port เนื่องจากจะมีการนำพอร์ตขนาน มาใช้งานติดต่อกับเครื่องพริ้นเตอร์เป็นหลัก โดยที่พอร์ตขนานนั้นสามารถให้ความเร็วในการส่งผ่านข้อมูลได้รวดเร็วกว่าพอร์ตอนุกรม และยังสามารถส่งข้อมูลขนาน 8 บิตออกไปได้โดยตรง ซึ่งมีลักษณะการทำงานดังรูปที่ 2.7



รูปที่ 2.7 พอร์ตขนาน

พอร์ตขนานของ Computer ประกอบด้วยสัญญาณทั้งหมด 25 เส้นสัญญาณ (DB25) โดยสัญญาณจะแบ่งออกได้เป็น 3 กลุ่มใหญ่ๆ ตามลักษณะหน้าที่ของสัญญาณ ประกอบด้วย

1. Data Register (Data bus) จำนวน 8 เส้น
2. Control Register จำนวน 4 เส้น
3. Status Register จำนวน 5 เส้น

ซึ่งได้แสดงขาสัญญาณลักษณะหน้าที่ของสัญญาณดังตารางที่ 2.1

เอกสารนี้เป็นเอกสารของบริษัทฯ ทรัพย์สินทางปัญญาของบริษัทฯ ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.1 ขาสัญญาณของพอร์ตขนาน

Pin No (DB25)	Signal name	Direction	Register – bit	Inverted
1	nStrobe	Out	Control-0	Yes
2	Data0	In/Out	Data-0	No
3	Data1	In/Out	Data-1	No
4	Data2	In/Out	Data-2	No
5	Data3	In/Out	Data-3	No
6	Data4	In/Out	Data-4	No
7	Data5	In/Out	Data-5	No
8	Data6	In/Out	Data-6	No
9	Data7	In/Out	Data-7	No
10	nAck	In	Status-6	No
11	Busy	In	Status-7	Yes
12	Paper-Out	In	Status-5	No
13	Select	In	Status-4	No
14	Linefeed	Out	Control-1	Yes
15	nError	In	Status-3	No
16	nInitialize	Out	Control-2	No
17	nSelect-Printer	Out	Control-3	Yes
18-25	Ground	-	-	-

Data Register (Data bus) จะมีอยู่ 8 ขา ตั้งแต่ขาที่ 2 ถึงขาที่ 9 บางทีมักถูกเรียกว่า Data Register ซึ่ง Register ตัวนี้จะส่งค่าได้อย่างเดียว ไม่สามารถรับค่าได้

Status Register เป็นรีจิสเตอร์ที่อ่านได้อย่างเดียวไม่สามารถเขียนข้อมูลได้ รีจิสเตอร์นี้จะมีสัญญาณเข้าอยู่ 5 สัญญาณและสัญญาณ IRQ กับสัญญาณสงวนไว้อีกสองบิต โดยสัญญาณ Busy จะ Active Low ลักษณะการทำงานของแต่ละบิตใน Status Register มีดังนี้

- Bit7 Busy เมื่อ Active หมายถึงพริ้นเตอร์จะไม่รับข้อมูล
- Bit6 nAck เมื่อ Active หมายถึงพริ้นเตอร์พร้อมที่จะทำงาน (Active Low)

- Bit5 Paper End เมื่อ Active หมายถึงพริ้นเตอร์ไม่มีกระดาษ

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้สำหรับใช้ภายในเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- Bit4 Select เมื่อ Active หมายถึงเลือกพรินเตอร์
- Bit3 nError เมื่อ Active หมายถึงพรินเตอร์เกิดข้อผิดพลาด (Active Low)
- Bit2, Bit1, Bit0 ไม่ใช่

Control Register เป็นรีจิสเตอร์ที่ใช้ในการควบคุมพรินเตอร์ สัญญาณในกลุ่มนี้จะ Active Low ยกเว้นสัญญาณ Initialize เท่านั้นที่ไม่ถูก Invert ลักษณะการทำงานของแต่ละบิตใน Control Register มีดังนี้

- Bit3 nSelect Printer เมื่อ Active หมายถึงเลือกพรินเตอร์
- Bit2 nInitialize เมื่อ Active หมายถึงรีเซตพรินเตอร์
- Bit1 nAuto Feed เมื่อ Active หมายถึงพรินเตอร์กระทำ Line Feed
- Bit0 nStrobe เมื่อ Active หมายถึง การบอกให้พรินเตอร์ทราบว่าข้อมูลเข้ามาแล้ว

Register Address

รีจิสเตอร์ คือ หน่วยความจำ RAM ประเภทหนึ่ง โดยสามารถนำข้อมูลเข้าไปเก็บไว้หรืออ่านข้อมูลได้เช่นเดียวกับหน่วยความจำทุกอย่าง ความแตกต่างระหว่างหน่วยความจำกับรีจิสเตอร์คือ CPU จะติดต่อกับรีจิสเตอร์นั้น จะมีชุดคำสั่งที่เรียกชื่อของรีจิสเตอร์โดยตรง ไม่ต้องระบุตำแหน่ง แต่ส่วนของหน่วยความจำนั้น จำเป็นที่จะต้องระบุตำแหน่งโดยชัดเจน สำหรับ Register Address จะแสดงตำแหน่งของหน่วยความจำประเภท Register ซึ่งแสดงได้ดังตารางที่ 2.2

ตารางที่ 2.2 Register Address

Register Address		
Register	LPT1	LPT2
Data register (base address + 0)	0x378	0x278
Status register (base address + 1)	0x379	0x279
Control register (base address + 2)	0x37a	0x27a

2.4.1 การติดต่อพอร์ตขนานด้วย Visual Basic 6.0

สำหรับการเขียนโปรแกรมเพื่อควบคุมอุปกรณ์ภายนอกผ่านทางพอร์ตขนานด้วย Visual Basic 6.0 นั้น ตัว Visual Basic 6.0 เองก็สามารถทำได้ดี แต่การเขียนโปรแกรมควบคุมอุปกรณ์ภายนอกนั้น Visual Basic 6.0 ไม่มีฟังก์ชันสำหรับติดต่อกับพอร์ตขนานได้โดยตรง แต่สามารถเขียนโปรแกรมให้ติดต่อกับพอร์ตขนาน ได้โดยใช้ฟังก์ชันที่เตรียมมาในรูปของฟังก์ชัน API โดยเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สามารถอ่านข้อมูลและส่งข้อมูลไปยังพอร์ตที่ต้องการได้ โดยใช้ไลบรารีพิเศษที่ใช้ในการเขียนโปรแกรมติดต่อพอร์ตนานันั้น คือ Inpout32.dll

2.5 การใช้งานพอร์ตอนุกรม RS-232C

การสื่อสารแบบอนุกรมนับว่ามีความสำคัญต่อการใช้งานไมโครคอนโทรลเลอร์มาก เพราะสามารถใช้เป็นพินท์และจอภาพของ Computer เป็น Input และ Output ในการติดต่อหรือควบคุมไมโครคอนโทรลเลอร์ด้วยสัญญาณอย่างน้อยเพียง 3 เส้นเท่านั้น คือ

1. สายส่งสัญญาณ (TX)
2. สายรับสัญญาณ (RX)
3. สายกราวด์ (GND)

โดยปกติพอร์ตอนุกรม RS-232C จะสามารถต่อสายได้ยาว 50 ฟุตโดยประมาณ ขึ้นอยู่กับชนิดของสายสัญญาณ ระยะทางและปริมาณสัญญาณรบกวน

สำหรับการจัดขาสัญญาณของพอร์ตอนุกรม RS-232C และหน้าที่ของแต่ละขาสัญญาณแสดงได้ดังตารางที่ 2.3

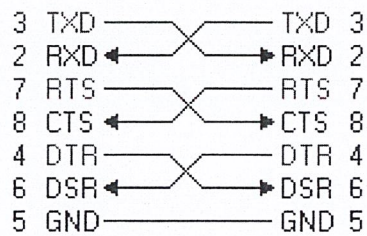
ตารางที่ 2.3 การจัดขาของคอนเน็กเตอร์อนุกรมแบบ DB9 และหน้าที่การใช้งาน

Pin No (DB9)	Description	Type
1	Data Carrier Detect (DCD)	Input
2	Received Data (RXD)	Input
3	Transmitted Data (TXD)	Output
4	Data Terminal Ready (DTR)	Output
5	Signal Ground (GND)	Input
6	Data Set Ready (DSR)	Input
7	Request To Send (RTS)	Output
8	Clear to Send (CTS)	Input
9	Ring Indicator (RI)	Input

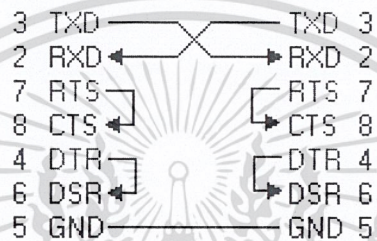
2.5.1 การเชื่อมต่ออุปกรณ์ภายนอกเข้ากับ Computer ด้วยสาย DB9

สำหรับการเชื่อมต่ออุปกรณ์ภายนอกเข้ากับ Computer โดยใช้สาย DB9 จะต้องทำการเชื่อมต่อขาสัญญาณต่างๆ เข้าด้วยกันดังรูปที่ 2.8 ซึ่งเป็นแบบ Null modem และดังรูปที่ 2.9 เป็นการเชื่อมต่อกันแบบใช้สาย 3 เส้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.8 การเชื่อมต่ออุปกรณ์ภายนอกผ่าน DB9 แบบ Null modem



รูปที่ 2.9 การเชื่อมต่ออุปกรณ์ภายนอกผ่าน DB9 แบบ 3 เส้น

2.5.2 การทำงานของขาสัญญาณ DB9

สำหรับขาสัญญาณต่างๆ ของ DB9 จะมีหน้าที่ในการทำงานดังต่อไปนี้

TXD เป็นขาที่ใช้ส่งข้อมูล

RXD เป็นขาที่ใช้รับข้อมูล

DTR แสดงสถานะพอร์ตว่าเปิดใช้งาน

DSR ตรวจสอบว่าพอร์ตที่ติดต่อกับเปิดอยู่หรือไม่

- เมื่อเปิดพอร์ตอนุกรม ขา DTR จะ ON เพื่อให้อุปกรณ์ได้รับทราบว่าการติดต่อกำลังดำเนินอยู่
- ในขณะเดียวกันก็จะตรวจสอบขา DSR ว่าอุปกรณ์พร้อมหรือไม่

RTS แสดงสถานะพอร์ตที่ต้องการส่งข้อมูล

CTS ตรวจสอบว่าพอร์ตที่ติดต่อกำลังต้องการส่งข้อมูลหรือไม่

- เมื่อต้องการส่งข้อมูลขา RTS จะ ON และจะส่งข้อมูลออกที่ขา TXD เมื่อส่งเสร็จก็จะ OFF
- ในขณะเดียวกันก็จะตรวจสอบขา CTS ว่าอุปกรณ์ต้องการที่จะส่งข้อมูลหรือไม่

GND เป็นขา ground

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.5.3 อัตราการส่งข้อมูล (Baud rate)

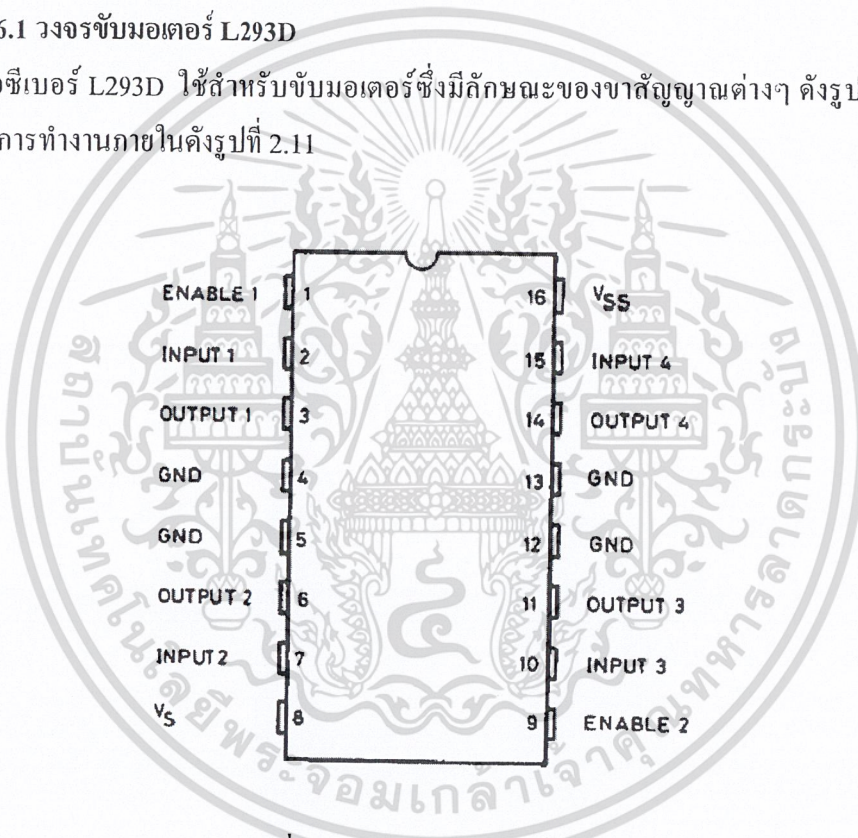
อัตราการส่งข้อมูล คือ ความเร็วของการรับ-ส่งข้อมูล เป็นจำนวนบิตต่อวินาที เช่น 300 1,200 2,400 4,800 9,600 14,400 19,200 38,400 56,000 เป็นต้น

การเลือกอัตราการส่งข้อมูลขึ้นอยู่กับชนิดของสายสัญญาณ ระยะทางและปริมาณสัญญาณรบกวน

2.6 อุปกรณ์อิเล็กทรอนิกส์

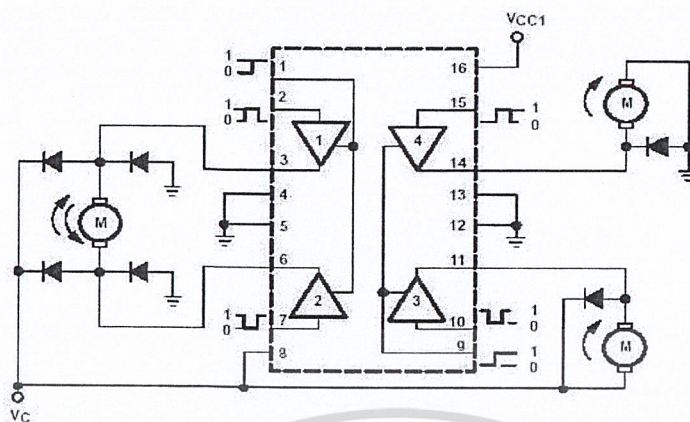
2.6.1 วงจรขับมอเตอร์ L293D

ไอซีเบอร์ L293D ใช้สำหรับขับมอเตอร์ซึ่งมีลักษณะของขาสัญญาณต่างๆ ดังรูปที่ 2.10 และมีหลักการทำงานภายในดังรูปที่ 2.11



รูปที่ 2.10 ตำแหน่งขาของไอซี L293D

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.11 หลักการทำงานของไอซี L293D

จากรูปแบบหลักการทำงานพบว่า สามารถควบคุมมอเตอร์ได้ทั้งแบบทิศทางเดียว ซึ่งจะ
สามารถควบคุมมอเตอร์ได้ 4 ตัว และแบบสองทิศทางซึ่งจะสามารถควบคุมมอเตอร์ได้ 2 ตัว

คุณสมบัติเด่น

1. ขับมอเตอร์ DC ได้ 2 ตัวพร้อมกัน (เหมาะสำหรับขับล้อซ้าย ขวา)
2. รองรับแรงดันไฟฟ้าจากมอเตอร์ 4.5-36 VDC (ที่แรงดันไฟฟ้าสูงควรมีฮีตซิงค์ (Heat Sink) มาติดที่ตัวถังของไอซีเพื่อกระจายความร้อน)
3. จ่ายกระแสได้ 600 mA สำหรับต่อมอเตอร์ 2 ตัวและ 1.2 A สำหรับต่อมอเตอร์ 1 ตัว
4. สามารถทำเป็นวงจรขับสเต็ปเปอร์มอเตอร์ แบบยูนิโพลาไรซ์ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3

การคำนวณและการสร้าง

ในการออกแบบระบบตรวจสอบสารละลาย ในเบื้องต้นจะต้องทำความเข้าใจเกี่ยวกับหลักการทำงานในภาพรวมของระบบก่อน ว่ามีหลักการทำงานอย่างไร มีส่วนประกอบใดบ้างและแต่ละส่วนมีหน้าที่อย่างไร สำหรับบทนี้จะอธิบายถึงหลักการทำงานโดยรวมของระบบ การออกแบบการทำงานของระบบ ทั้งส่วนของหน้าจออินเทอร์เฟซ โปรแกรม PLC อุปกรณ์เชื่อมต่อไปยังปั๊ม รวมทั้งวงจรแสดงผลต่างๆ

3.1 ขอบเขตการทำงานของระบบ



รูปที่ 3.1 ระบบการทำงานของกระบวนการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 3.1 จะแสดงหลักการทำงานของกระบวนการ โดยจะทำการตั้งค่าต่างๆ ไม่ว่าจะเป็นการตั้งค่าพอร์ตการติดต่อ การตั้งค่าสถานะของปั๊มแต่ละตัว การกำหนดเวลาในการควบคุมการเปิด-ปิดปั๊มและการตั้งการควบคุมจากหน้าจ่อินเตอร์เฟซ ซึ่งพัฒนาขึ้นโดยโปรแกรม Visual Basic 6.0 จากนั้นค่าต่างๆที่ถูกตั้งค่าไว้ จะถูกส่งไปยังหน่วยความจำของ PLC เพื่อทำการประมวลผลตามโปรแกรมแลคเคอร์ที่ได้เขียนเอาไว้ เมื่อ PLC ประมวลผลและส่งค่าสัญญาณออกมาทางเอาต์พุต ซึ่งเป็นสัญญาณแรงดันไฟฟ้า 24 VDC ซึ่งแรงดันไฟฟ้านี้จะส่งไปควบคุมการทำงานของของรีเลย์ ซึ่งทำงานเสมือนเป็นสวิตช์เปิด-ปิดสัญญาณที่จะควบคุมการทำงานในกระบวนการต่างๆ ของปั๊ม ซึ่งได้แก่ การควบคุมการหมุนและการควบคุมทิศทางหมุน

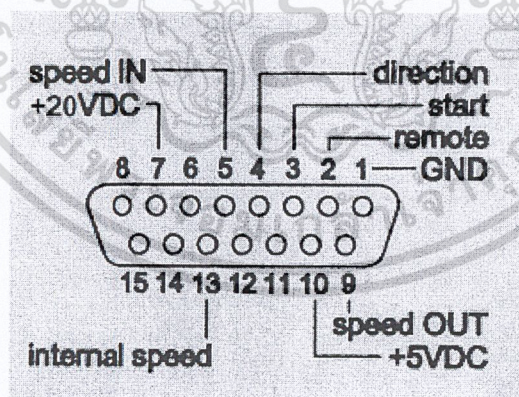
3.2 การออกแบบและควบคุมปั๊มเพริสทอลติก (Peristaltic Pump)

3.2.1 การควบคุมผ่านทาง Analog Interface

การควบคุมผ่านทาง Analog Interface สามารถควบคุมกระบวนการได้ 3 อย่าง คือ

1. การควบคุมการหมุน-หยุดหมุนของปั๊ม (Start/Stop)
2. การควบคุมทิศทางหมุนของปั๊ม (Rotation Direction)
3. การควบคุมความเร็ว (Speed Control)

สำหรับกระบวนการต่างๆ จะสามารถควบคุมโดยการป้อนค่าสัญญาณแรงดันไฟฟ้าเข้าไปยังอินพุตของปั๊ม ซึ่งแสดงได้ดังรูปที่ 3.2



รูปที่ 3.2 อินพุตสำหรับอนาล็อกอินเตอร์เฟซ (Analog Interface Input)

3.2.2 การควบคุมการหมุน (Start/Stop)

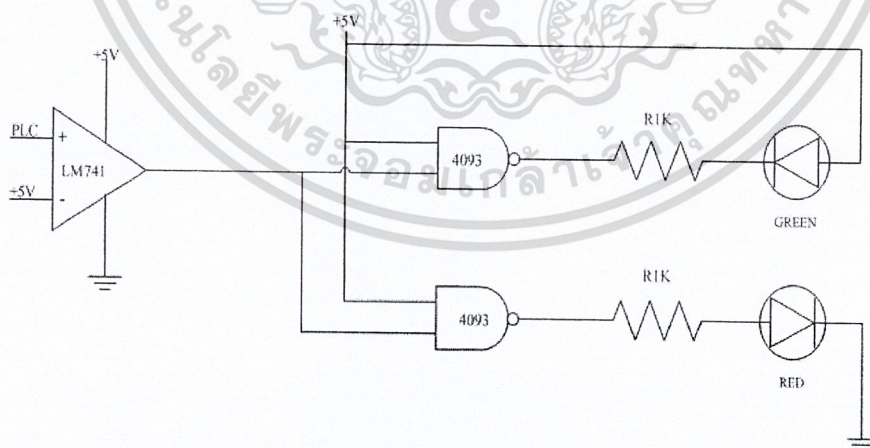
การควบคุมการหมุนจะสอดคล้องกับสัญญาณเอาต์พุตจาก PLC ดังนั้นจึงได้เลือกใช้รีเลย์เพื่อใช้เป็นสวิตช์สำหรับตัด-ต่อสัญญาณไฟที่จะป้อนให้กับปั๊มเพื่อควบคุมให้ปั๊มหมุน-หยุดหมุน

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้สำหรับการใช้ภายในเท่านั้น ไม่อนุญาตให้เผยแพร่หรือใช้เพื่อการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เป็นตัวจำกัดกระแสให้กับ LED แรงดันไฟฟ้าเอาต์พุตจาก PLC จะถูกส่งมาเป็นอินพุตสำหรับวงจร ส่วนแสดงผล ดังรูปที่ 3.4 การทำงานของวงจรจึงขึ้นกับระดับสัญญาณเอาต์พุตจาก PLC ดังนี้

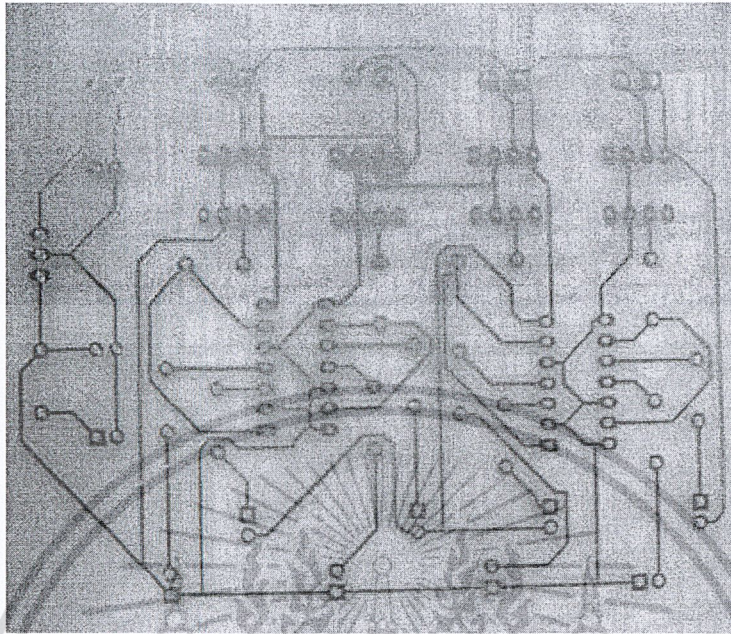
- เมื่อมีสัญญาณเอาต์พุตออกมาจาก PLC เข้าทางขาอินพุตด้าน PLC จะทำให้ไอซี 741 วัดระดับสัญญาณได้เอาต์พุตออกมาเป็น 5 VDC ส่งต่อไปยังไอซีเบอร์ 4093 (NAND gate) ซึ่งขาอินพุตด้านหนึ่งต่ออยู่กับสัญญาณไฟฟ้า 5 VDC กรณีนี้ทำให้ระดับแรงดันไฟฟ้าด้านเอาต์พุตของ NAND gate มีค่าเท่ากับ 0 VDC เป็นผลให้เกิดการนำกระแสขึ้นที่หลอด LED สีเขียวในขณะที่หลอด LED สีแดงไม่เกิดการนำกระแส จึงเกิดการเปล่งแสงสีเขียวขึ้น เป็นการบ่งชี้ว่ามีการทำงานของปั้มนั่นๆ แล้ว
- เมื่อไม่มีสัญญาณเอาต์พุตจาก PLC ส่งเข้ามาทางขาอินพุตของไอซี 741 ทำให้ไอซีเบอร์ 741 วัดระดับสัญญาณได้เอาต์พุตออกมาเป็น 0 VDC ส่งต่อไปยังไอซีเบอร์ 4093 (NAND gate) ซึ่งขาอินพุตด้านหนึ่งต่ออยู่กับสัญญาณไฟฟ้า 5 VDC กรณีนี้ทำให้ระดับแรงดันไฟฟ้าด้านเอาต์พุตของ NAND gate มีค่าเท่ากับ 5 VDC เป็นผลให้เกิดการนำกระแสขึ้นที่หลอด LED สีแดง แต่ไม่เกิดการนำกระแสที่หลอด LED สีเขียว ทำให้เกิดการเปล่งแสงของหลอด LED สีแดงขึ้น เป็นการบ่งชี้ว่าไม่มีสัญญาณจาก PLC ออกมาทำงานให้กับปั้มนั่นๆ

สำหรับลวงจรจริงที่ได้ทำขึ้นแสดงไว้ดังรูปที่ 3.5 เมื่อวางอุปกรณ์ต่างๆ ลงบนลวงจรแล้วแสดงได้ดังรูปที่ 3.6 ซึ่งประกอบไปด้วยวงจรส่วนแสดงผล 4 วงจรและวงจรแปลงไฟฟ้าโดยใช้ ไอซี 7805 อีกหนึ่งวงจร

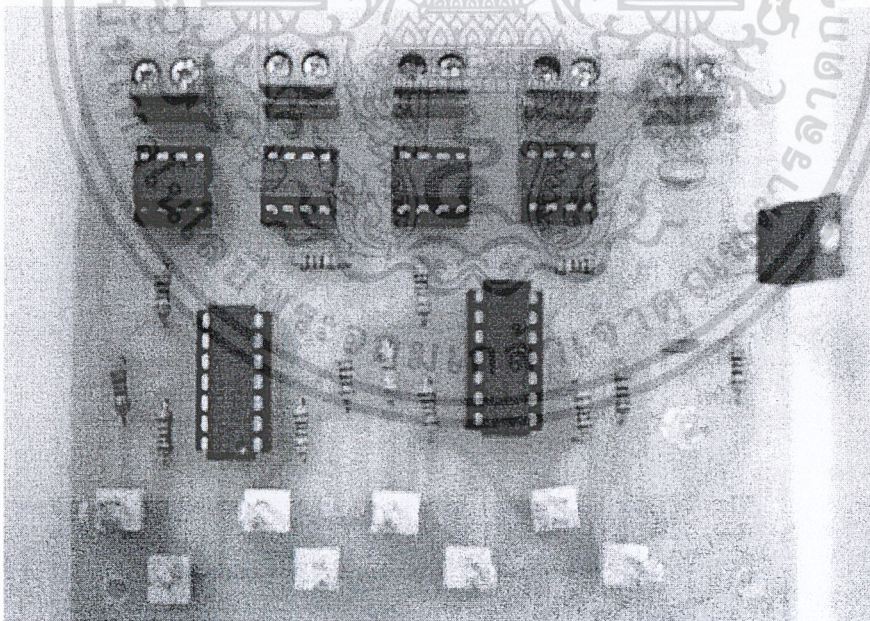


รูปที่ 3.4 วงจรแสดงผล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.5 ถายวงจรของวงจรแสดงผล

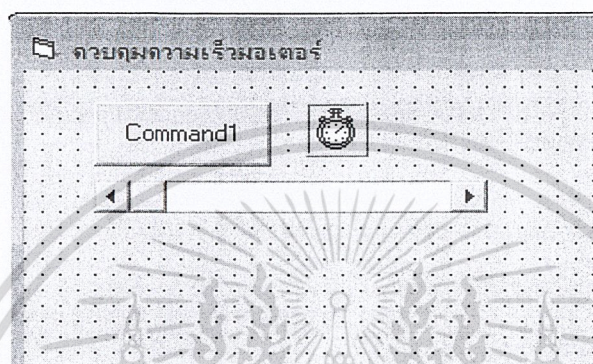


รูปที่ 3.6 วงจรแสดงผลจริง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.6 การควบคุมความเร็วมอเตอร์ด้วยพอร์ตขนาน

ในการควบคุมความเร็วในการหมุน ได้ทดสอบควบคุมทางหน้าจออินเตอร์เฟซส่งสัญญาณผ่านทางพอร์ตขนานของ computer โดยได้มีการออกแบบหน้าจออินเตอร์เฟซดังรูปที่ 3.7 โดยสามารถเลือกกำหนดความเร็วโดยการเลื่อนแถบสกรอลบาร์ (Scroll Bar) และสั่งการทำงานโดยการกดปุ่ม Command1



รูปที่ 3.7 หน้าจอ GUI ของโปรแกรมควบคุมความเร็ว

ซึ่งต้องเขียนโปรแกรมควบคุมดังนี้

```
Private Sub Timer1_Timer()
```

```
Out pwrite, &H3
```

```
Sleep (HScroll1.Value)
```

```
Out pwrite, &H0
```

```
Sleep ( HScroll1.Value)
```

```
End Sub
```

คำสั่งข้างต้นโปรแกรมจะส่งข้อมูล &H3 ออกจากพอร์ตขนานทำให้มอเตอร์หมุน จากนั้นก็จะหน่วงเวลาด้วยคำสั่ง

```
Sleep(HScroll1.value)
```

ซึ่งค่าที่จะทำการหน่วงจะรับค่ามาจากคอนโทรลสกรอลบาร์ จากนั้นโปรแกรมก็จะส่งข้อมูล &H0 ออกไป ทำให้มอเตอร์หยุดหมุนชั่วคราว และหน่วงเวลาด้วยข้อมูลที่ได้จากการคอนโทรลสกรอลบาร์ โปรแกรมจะวนลูปโดยการส่งข้อมูล &H3 ออกมาอีกครั้ง ทำให้มอเตอร์

หมุน เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในการสลับการส่งข้อมูลระหว่างข้อมูล &H3 และข้อมูล &H0 นั้นเปรียบเสมือนการสร้างสัญญาณพัลส์ให้กับตัวไอซี L293D ในการควบคุมจังหวะของสัญญาณพัลส์นั้น จะใช้การหน่วงเวลาเป็นตัวควบคุมจังหวะในการส่งสัญญาณพัลส์ออกมาจากพอร์ตนาน ซึ่งจะควบคุมความเร็วมอเตอร์โดยการควบคุมจังหวะของสัญญาณพัลส์นั่นเอง

3.3 การคำนวณและการสร้างซอฟต์แวร์

ในการสร้างซอฟต์แวร์ได้มีการศึกษาและปฏิบัติงานเป็น 2 ส่วน คือ

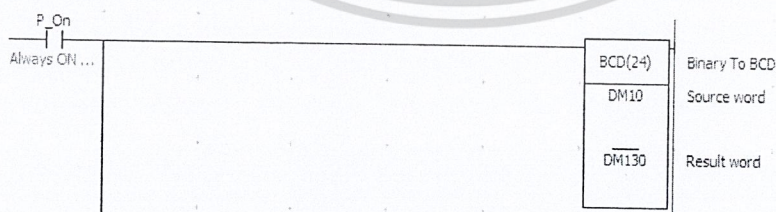
1. การออกแบบกระบวนการทำงานของระบบ การนำค่าจากการอินเทอร์เฟซที่ส่งมาเพื่อทำให้ระบบทำงานได้อย่างสมบูรณ์ตามวัตถุประสงค์ โดยการใช้ซอฟต์แวร์ชื่อ CX-Programmer ทำการเขียนแลตเตอร์ไดอะแกรมควบคุม PLC
2. การออกแบบการเชื่อมต่อรับส่งข้อมูลจากผู้ใช้ รวมทั้งการควบคุมกระบวนการต่างๆ ผ่านทางอินเทอร์เน็ต เพื่อให้ระบบทำงานได้อย่างสมบูรณ์ตามวัตถุประสงค์ โดยการใช้ซอฟต์แวร์ที่ชื่อว่า Visual Basic 6.0 ในการเขียนอินเทอร์เน็ตและโปรแกรม

3.3.1 การออกแบบแลตเตอร์ไดอะแกรมเพื่อควบคุมกระบวนการ

ในการควบคุมการทำงานของกระบวนการต่างๆ ต้องใช้การควบคุมผ่านทาง PLC ซึ่งจะรับค่าต่างๆ ไม่ว่าจะเป็นค่าเวลาในแต่ละขั้นตอนของปั๊ม ซึ่งมีค่าแปรเปลี่ยนไปเรื่อยๆ ตามแต่ผู้ใช้งานกำหนด จึงจำเป็นต้องมีตัวแปรที่สามารถมารับค่าสำหรับไทม์เมอร์ได้ ซึ่งในที่นี้ได้กำหนดให้ใช้ Data Memory เป็นตัวแปรในการรับค่าแล้วนำไปแสดงผล ซึ่งมีฟังก์ชันต่างๆ ที่ใช้ในการเขียนโปรแกรมดังนี้ คือ

1. ฟังก์ชัน BCD (24)

ใช้สำหรับแปลงค่าจาก Binary to BCD คือ แปลงค่าจากเลขฐาน 2 ใน DM10 เป็นเลขฐาน 10 แล้วเก็บค่าไว้ใน DM 130 ดังตัวอย่างในรูปที่ 3.8

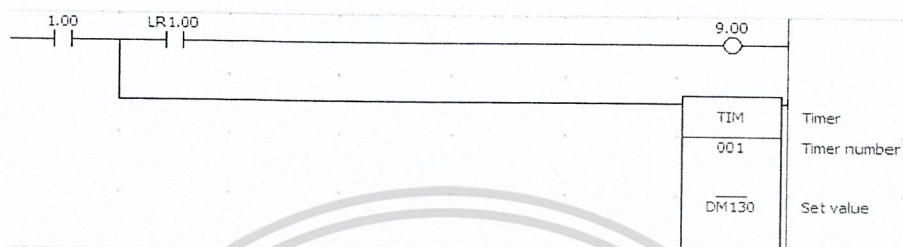


รูปที่ 3.8 ฟังก์ชัน BCD

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. ฟังก์ชัน TIM (timer)

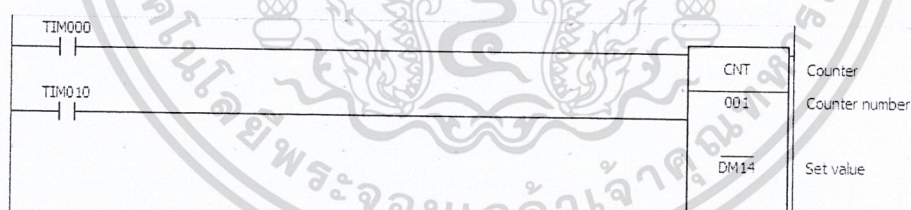
เป็นฟังก์ชันสำหรับนับเวลา โดยฟังก์ชันไทม์เมอร์หนึ่งตัวสามารถตั้งค่าเวลาได้ระหว่าง 000.0 - 999.9 วินาที สามารถตั้งชื่อได้ระหว่าง 000 – 127 โดยที่ไทม์เมอร์กับเคาน์เตอร์ห้ามมีชื่อซ้ำกันสำหรับการเรียกใช้ฟังก์ชัน แสดงได้ดังรูปที่ 3.9



รูปที่ 3.9 ฟังก์ชัน TIM

3. ฟังก์ชัน CNT (Counter)

เป็นฟังก์ชันสำหรับใช้นับจำนวนครั้ง โดยฟังก์ชันเคาน์เตอร์หนึ่งตัวสามารถตั้งค่าได้ระหว่าง 0000 – 9999 หลักการทำงาน คือ มี 2 หน้า contact อันหนึ่งทำหน้าที่ set อีกอันหนึ่งทำหน้าที่ reset ในที่นี้ contact TIM000 ทำหน้าที่ set ส่วน contact TIM010 ทำหน้าที่ reset ดังรูปที่ 3.10

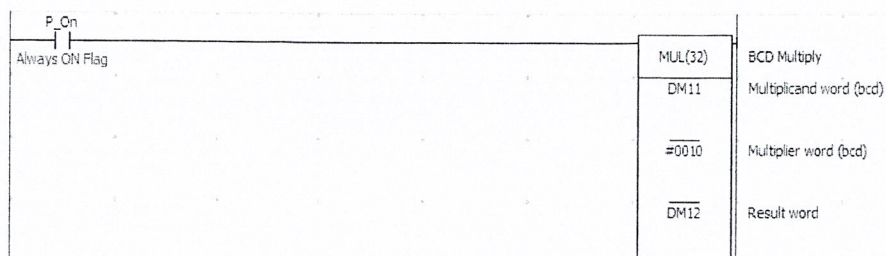


รูปที่ 3.10 ฟังก์ชัน CNT

4. ฟังก์ชัน MUL (BCD Multiply)

เป็นฟังก์ชันที่ใช้สำหรับคูณค่า BCD ตามตัวอย่างดังรูปที่ 3.11 หลักการทำงาน คือ ค่าที่อยู่ใน DM11 ทำหน้าที่เป็นตัวตั้ง ค่าคงที่ #0010 (เท่ากับ 10) ทำหน้าที่เป็นตัวคูณ ส่วนผลลัพธ์ในการคูณถูกเก็บไว้ใน DM12

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.11 ฟังก์ชัน MUL

5. ฟังก์ชัน DIVL (Long BCD divide)

เป็นฟังก์ชันที่ใช้สำหรับหารค่า BCD โดยมีความพิเศษ คือ สามารถหารค่าได้ 8 bit ตามตัวอย่างในรูปที่ 3.12 หลักการทำงานคือ ค่าใน DM12 ทำหน้าที่เป็นตัวตั้ง ค่าใน DM2 ทำหน้าที่เป็นตัวหาร ส่วนผลลัพธ์ในการหารถูกเก็บไว้ใน DM14



รูปที่ 3.12 ฟังก์ชัน DIVL

3.3.2 การออกแบบหน้าจออินเตอร์เฟสสำหรับเชื่อมต่อและตั้งกระบวนการ

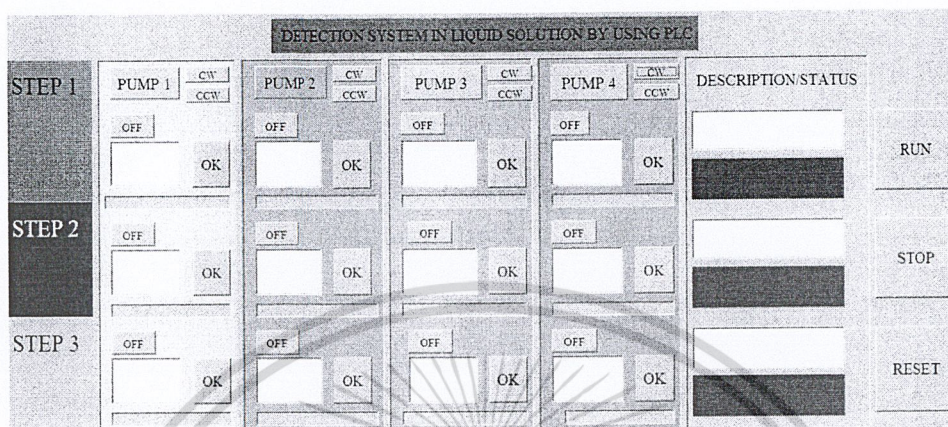
สำหรับหน้าจออินเตอร์เฟสได้พัฒนาขึ้นจากโปรแกรม Visual Basic 6.0 ดังรูปที่ 3.13 ซึ่งความสามารถของหน้าจออินเตอร์เฟสที่ออกแบบมีดังนี้

1. สามารถเลือกปุ่มที่ต้องการใช้งานได้
2. สามารถเลือกทิศทางการหมุนของปุ่มแต่ละตัวได้
3. สามารถเลือกกระบวนการของปุ่มแต่ละตัวให้หมุนหรือไม่หมุนในแต่ละขั้นตอนได้
4. สามารถป้อนค่าเวลาสำหรับกระบวนการในแต่ละขั้นตอนได้

5. มีแถบแสดงสถานะความคืบหน้าของกระบวนการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6. มีแถบแสดงสถานะของแต่ละขั้นตอนซึ่งสามารถที่จะป้อนข้อความได้
7. สามารถสั่งกระบวนการผ่านทางหน้าจออินเทอร์เฟสได้



รูปที่ 3.13 หน้าจออินเทอร์เฟส

สำหรับโปรแกรมคำสั่งต่างๆ สามารถศึกษาได้จากภาคผนวก ก โปรแกรมควบคุม

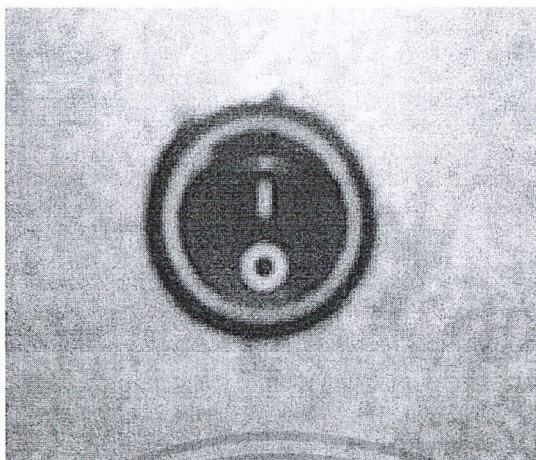
3.4 ก่อตั้งควบคุม

เป็นกล่องที่ใช้รวบรวมอุปกรณ์ต่างๆ เข้าด้วยกัน ไม่ว่าจะเป็น PLC วงจรรีเลย์ วงจรแสดงผลและอุปกรณ์ต่อพ่วงต่างๆ เพื่ออำนวยความสะดวกการพกพา และสามารถสั่งการกระบวนการผ่านทางกล่องควบคุมนี้ได้ ซึ่งตัวกล่องควบคุมจะประกอบไปด้วย

3.4.1 สวิตช์เปิด-ปิด

สวิตช์ใช้สำหรับเปิด-ปิดกระแสไฟฟ้าเข้ากล่องควบคุมเพื่อป้อนให้กับ PLC ซึ่งจะต้องต่อเข้ากับแรงดันไฟฟ้า 220 VAC มีลักษณะดังรูปที่ 3.14

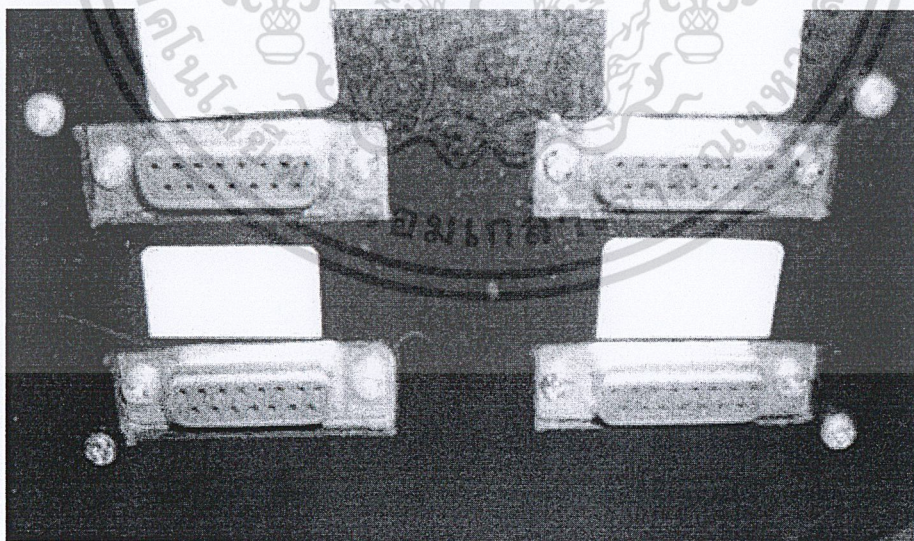
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.14 สวิตช์เปิด-ปิดก่อกำหนดความปลอดภัย

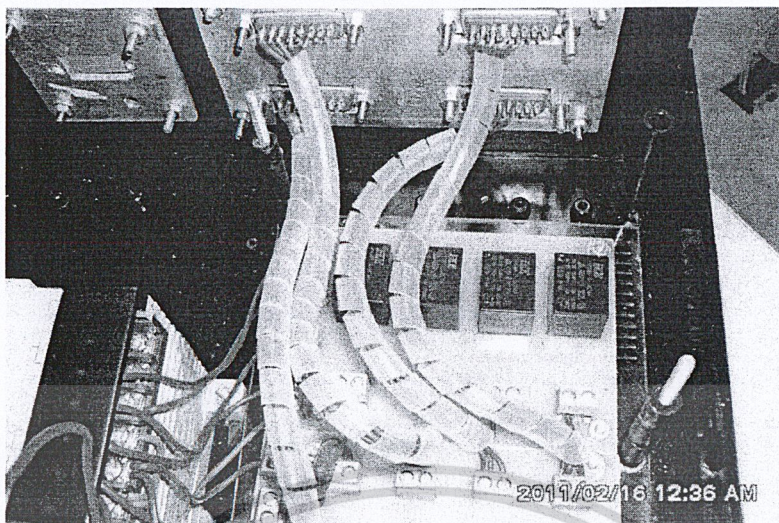
3.4.2 พอร์ตอนุกรม RS-232C

พอร์ตอนุกรม RS-232C เป็นอุปกรณ์ที่เชื่อมต่อมาจากวงจรควบคุมการเปิด-ปิด และวงจรควบคุมทิศทางการหมุนของปั๊ม ดังแสดงในรูปที่ 3.16 ใช้สำหรับเชื่อมต่อไปยังปั๊มเพื่อควบคุมกระบวนการของปั๊ม ดังแสดงในรูปที่ 3.15



รูปที่ 3.15 พอร์ตอนุกรม RS-232C

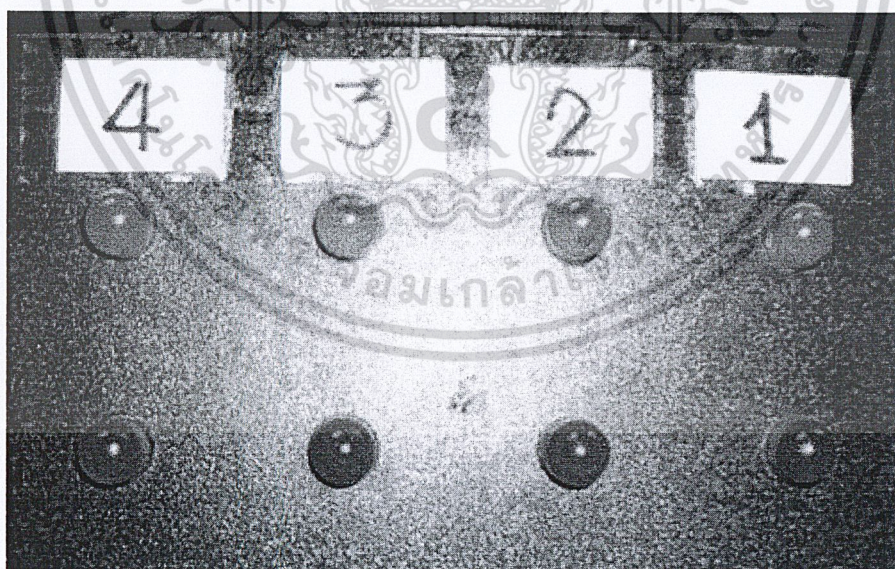
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.16 การเชื่อมต่อระหว่างวงจรรีเลย์กับพอร์ตอนุกรม RS-232C

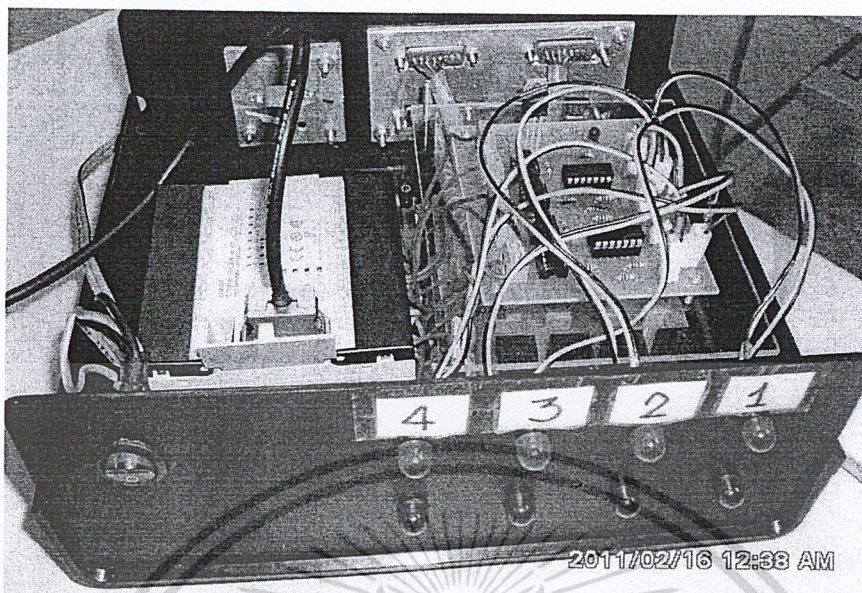
3.4.3 ไฟแสดงผล

ดวงไฟแสดงผล ใช้สำหรับแสดงสถานะการทำงานของปั๊ม ซึ่งใช้หลักการดังที่ได้อธิบายไปแล้วในหัวข้อวงจรแสดงผล มีลักษณะดังรูปที่ 3.17



รูปที่ 3.17 ไฟแสดงผล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.18 การติดตั้งอุปกรณ์ภายในกล่องควบคุม

สำหรับรูปที่ 3.18 แสดงการติดตั้งอุปกรณ์ต่างๆ ภายในกล่องควบคุม ซึ่งประกอบไปด้วย PLC วงจรรีเลย์ วงจรแสดงผล ไฟแสดงผล สวิตช์ พอร์ตอนุกรมเชื่อมต่อไปยังปั๊ม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

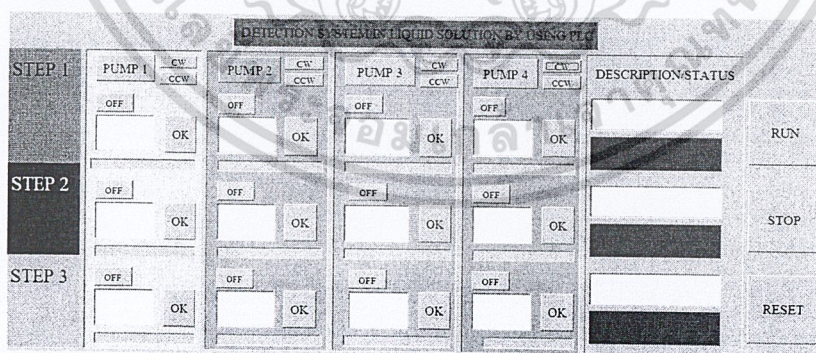
การทดลองและผลการทดลอง

จากการออกแบบระบบควบคุมกระบวนการทั้งการออกแบบหน้าจอบนคอมพิวเตอร์ การออกแบบแลคเคอร์ไคอะแกรมใน PLC ตลอดจนการการออกแบบวงจรรีเลย์และวงจรแสดงผล ซึ่งได้อธิบายไปในบทที่ผ่านมา สำหรับในบทนี้จะเป็นการสาธิตการใช้งานหน้าจอบนคอมพิวเตอร์ การควบคุมกระบวนการ ตลอดจนแสดงผลการทดลองต่างๆ

4.1 ตั้งค่าและเริ่มต้นใช้โปรแกรมควบคุมกระบวนการ

1. เปิดหน้าจอบนระบบควบคุมกระบวนการซึ่งพัฒนาขึ้นจากโปรแกรม Visual basic 6.0 ซึ่งมีหน้าจอลักษณะดังรูปที่ 4.1 โดยจะต้องทำการเชื่อมต่อ Computer กับ PLC เข้ากันด้วยพอร์ตอนุกรม RS-232C และตั้งค่าพอร์ต โดยการคลิกขวา My Computer -> Properties -> Hardware -> Device Manager -> Ports(COM&LPT) แล้วเลือกอุปกรณ์และตั้งค่าดังนี้

- Comport = COM1
- Baud rate = 9600
- Data bits = 7
- Parity = Even
- Stop bits = 2



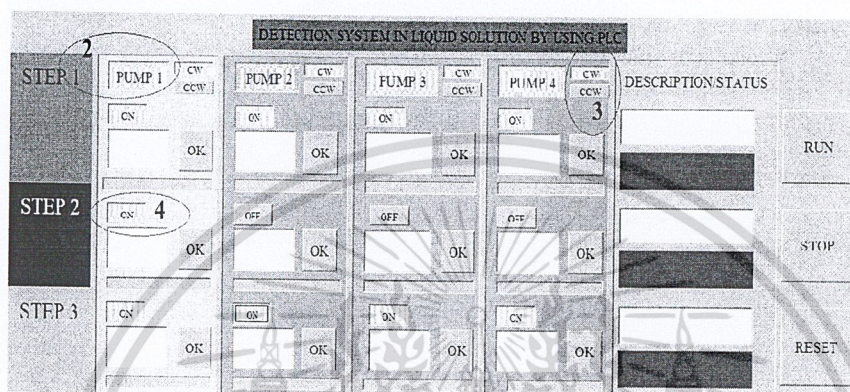
รูปที่ 4.1 หน้าจอ GUI เมื่อเปิดโปรแกรมขึ้นมา

2. กำหนดปั๊มที่ต้องการใช้งาน โดยการกดเลือก PUMP ที่ต้องการจะใช้งาน ดังรูปที่ 4.2

3. กำหนดทิศทางของปั๊มแต่ละตัว ดังรูปที่ 4.2 โดยที่

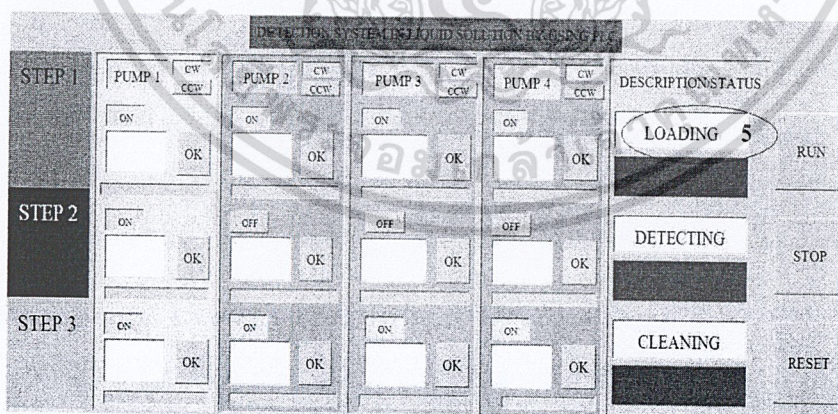
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- CW = Clockwise
 - CCW = Counterclockwise
4. กำหนดการทำงานของปั๊มในแต่ละขั้นตอนการทำงานของงาน ซึ่งประกอบด้วยการกำหนดให้ปั๊มหมุนหรือปั๊มหยุดหมุน โดยถ้าต้องการให้ปั๊มหมุนให้คลิกปุ่มให้เปลี่ยนเป็น ON ถ้าไม่ต้องการให้หมุนก็ให้คงสถานะเป็น OFF ตามรูปที่ 4.2



รูปที่ 4.2 การตั้งค่าต่างๆ ในการควบคุมปั๊ม

5. กำหนดคำอธิบายสถานะสำหรับแต่ละขั้นตอนการทำงานของปั๊ม เช่น Loading Detecting หรือ Cleaning เป็นต้น ดังรูปที่ 4.3



รูปที่ 4.3 การกำหนดสถานะในแต่ละขั้นตอน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6. ป้อนค่าเวลาสำหรับการควบคุมในแต่ละขั้นตอนของปั๊ม ดังรูปที่ 4.4 โดยที่

- ปั๊มที่ 1 สามารถป้อนค่าเวลาได้ตั้งแต่ 1 - 9999 วินาทีในแต่ละขั้นตอน
- ปั๊มที่ 2 - 4 สามารถป้อนค่าเวลาได้ตั้งแต่ 1 - 999 วินาทีในแต่ละขั้นตอน

โดยที่ปั๊มที่ 2 - 4 เมื่อทำงานเสร็จสิ้นขั้นตอนที่ 3 แล้วจะวนกลับมาทำที่ขั้นตอนที่ 1 ต่อไปจนกว่าปั๊มที่ 1 จะทำงานจนเสร็จสิ้นขั้นตอนที่ 3 ทุกปั๊มจึงจะหยุดการทำงานทั้งหมดไม่ว่าปั๊มที่ 2 - 4 จะยังคงทำงานอยู่ที่ขั้นตอนที่ใดก็ตาม

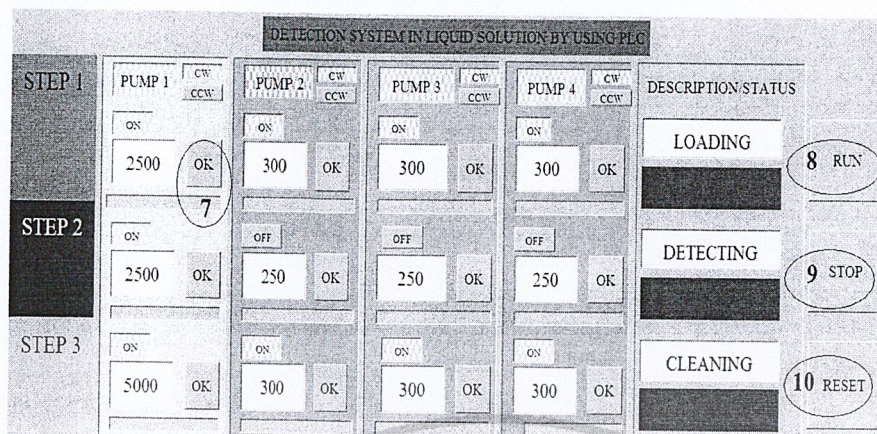
DETECTION SYSTEM IN LIQUID SOLUTION BY USING PLC

	PUMP 1	PUMP 2	PUMP 3	PUMP 4	DESCRIPTION STATUS
STEP 1	CW CCW ON 2500 OK 6	CW CCW ON 300 OK	CW CCW ON 300 OK	CW CCW ON 300 OK	LOADING RUN
STEP 2	ON 2500 OK	OFF 250 OK	OFF 250 OK	OFF 250 OK	DETECTING STOP
STEP 3	ON 5000 OK	ON 300 OK	ON 300 OK	ON 300 OK	CLEANING RESET

รูปที่ 4.4 การป้อนค่าเวลาให้กับปั๊ม

- คลิกปุ่ม OK เพื่อทำการส่งค่าต่างๆ ไปยัง PLC เพื่อทำการเก็บค่าและคำนวณค่าเอาไว้ ดังรูปที่ 4.5
- คลิกปุ่ม RUN เพื่อทำการสั่งการทำงานของระบบ ดังรูปที่ 4.5
- คลิกปุ่ม STOP เพื่อทำการหยุดกระบวนการทำงานต่างๆ ทั้งหมด โดยเมื่อคลิกปุ่มนี้แล้วจะไม่สามารถคลิกปุ่ม RUN เพื่อให้กระบวนการทำงานต่อไปได้อีก ดังรูปที่ 4.5
- คลิกปุ่ม RESET เพื่อทำการเคลียร์ค่าต่างๆ ทั้งหมด ซึ่งจะต้องคลิกปุ่มนี้ทุกครั้งที่เกิดกระบวนการ ไม่ว่าจะหลังจากการคลิกปุ่ม STOP หรือจะเป็นการสิ้นสุดกระบวนการทำงานเมื่อปั๊มที่ 1 ทำงานเสร็จสิ้นก็ตาม ดังรูปที่ 4.5

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

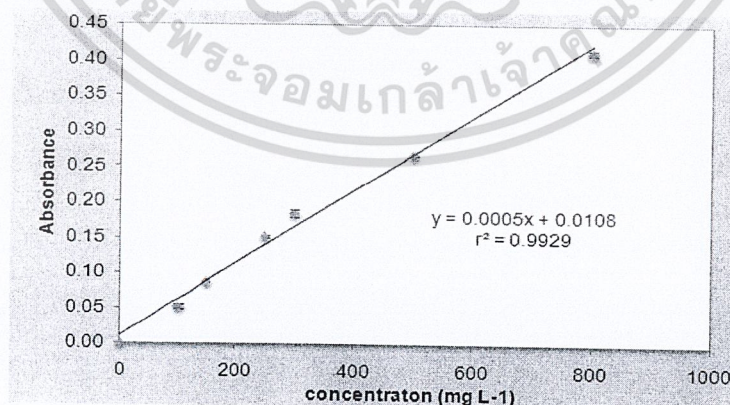


รูปที่ 4.5 การสั่งการทำงานของกระบวนการ

4.2 ผลการทดลอง

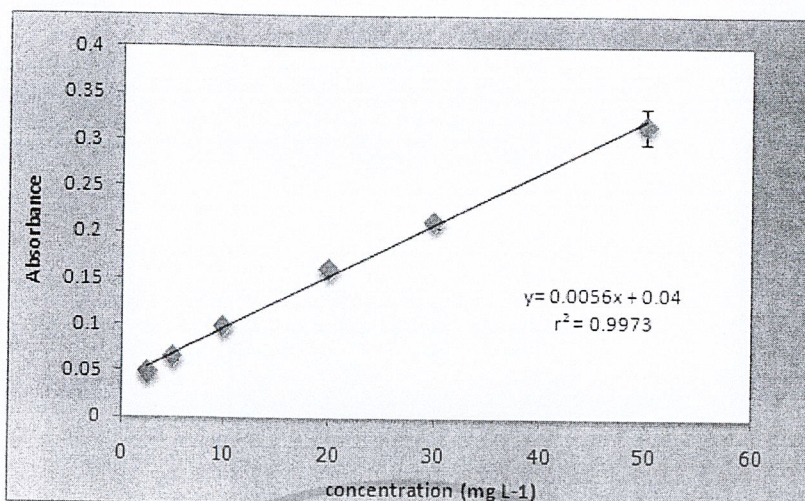
4.2.1 การควบคุมการเปิด-ปิดและทิศทางการหมุน

จากการทดสอบควบคุมการเปิด-ปิดและทิศทางการหมุนของปั๊ม ผลปรากฏว่าโปรแกรมใช้งานสามารถควบคุมการเปิด-ปิดของปั๊มได้อย่างเที่ยงตรงและมีประสิทธิภาพ โดยพิจารณาได้จากกราฟการทดสอบการดูดกลืนแสงของสารละลายหลังจากทำปฏิกิริยากัน โดยเปรียบเทียบกันระหว่างการควบคุมโดยมือเปิด-ปิด กับแบบควบคุมด้วยชุดควบคุม ดังรูปที่ 4.6 และรูปที่ 4.7 ตามลำดับ



รูปที่ 4.6 กราฟการดูดกลืนแสงของสารละลายที่ควบคุมปั๊มด้วยมือเปิด-ปิด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

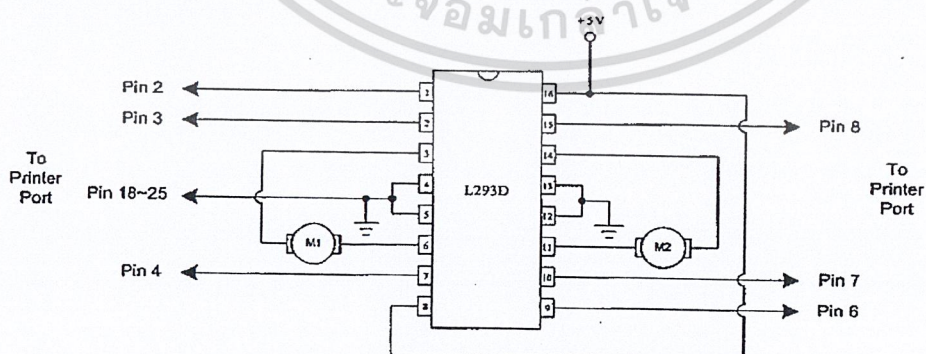


รูปที่ 4.7 กราฟการดูดกลืนแสงของสารละลายที่ควบคุมปริมาณด้วยชุดควบคุม

จากกราฟการทดสอบการดูดกลืนแสงของสารละลายทั้งสองแบบ พบว่าการควบคุมปริมาณด้วยชุดควบคุมให้ผลการทดลองที่มีความเที่ยงตรงมากกว่า โดยการพิจารณาจากค่า r^2 ซึ่งเป็นค่าที่แสดงความเป็นเส้นตรงของกราฟ โดยค่าที่เป็นที่ยอมรับคือ $r^2 = 0.99xx$ ซึ่งจากการทดลองพบว่าสามารถใช้ได้ทั้งแบบควบคุมด้วยมือและแบบควบคุมด้วยชุดควบคุม แต่การควบคุมด้วยชุดควบคุมมีความสะดวกสบายในการใช้งานมากกว่า มีความเที่ยงตรงมากกว่า

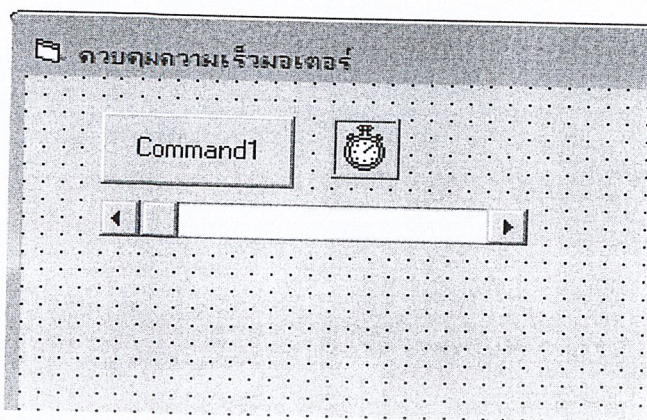
4.2.2 การทดลองควบคุมความเร็วด้วยพอร์ตขนาน

จากการทดลองควบคุมความเร็วโดยการกำหนดค่าจากหน้าจออินเตอร์เฟซ ส่งค่าทางพอร์ตขนานเพื่อควบคุมความเร็วปั๊ม โดยการต่อวงจรดังรูปที่ 4.8 และเขียนโปรแกรม Visual Basic 6.0 ควบคุมตามรูปที่ 4.9



รูปที่ 4.8 วงจรควบคุมความเร็วมอเตอร์ขับเคลื่อนโดยไอซี L293D

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

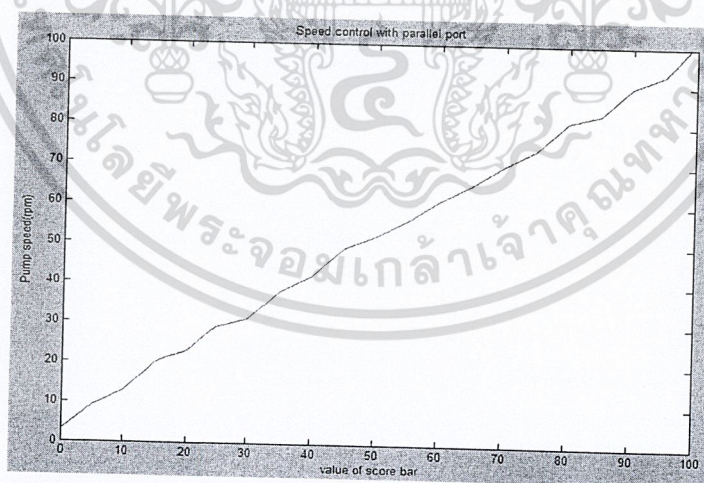


รูปที่ 4.9 หน้าจอ GUI ของโปรแกรมควบคุม

ซึ่งต้องเขียนโปรแกรมควบคุมดังนี้

```
Private Sub Timer1_Timer()
    Out pwrite, &H3      Sleep (HScroll1.Value)
    Out pwrite, &H0      Sleep ( HScroll1.Value)
End Sub
```

ซึ่งจากการทดลองได้ผลการทดสอบความเร็วของปั๊มตามกราฟ ดังในรูปที่ 4.10



รูปที่ 4.10 ผลการทดสอบความเร็วปั๊ม

จากกราฟดังรูปที่ 4.10 เป็นการทดสอบการควบคุมความเร็วปั๊มระหว่างการควบคุมผ่านทางพอร์ตขนานกับการควบคุมความเร็วบนแผงควบคุมของปั๊ม (Control panel) ซึ่งมีอัตราความเร็วไม่ต่างกันใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เป็น 1.03 : 1 ผลปรากฏว่าสามารถที่จะควบคุมความเร็วของปั๊มได้ แต่ไม่สามารถควบคุมความเร็วให้เป็นแบบเชิงเส้นในบางช่วงได้ ทำให้ไม่สามารถนำไปใช้ได้ทันที อาจจะต้องทำการต่อวงจรควบคุมระดับสัญญาณ หรือทำการปรับสัญญาณก่อนนำไปขับปั๊ม เพื่อให้สามารถตั้งการควบคุมความเร็วให้มีประสิทธิภาพตามที่ต้องการ ได้อย่างที่ขงตรงมากยิ่งขึ้น



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

บทวิจารณ์และสรุป

5.1 สรุปผลการปฏิบัติงาน

จากการออกแบบและทดสอบระบบควบคุมปั๊ม สามารถสรุปผลการปฏิบัติงานได้ดังนี้

1. สามารถสร้างหน้าจอควบคุมและส่งข้อมูลเชื่อมต่อไปยังอุปกรณ์ภายนอกได้
2. สามารถออกแบบแลตเตอร์ไดอะแกรมเพื่อรับส่งข้อมูล และตั้งกระบวนการให้ทำงานได้ตามที่ต้องการ
3. สามารถออกแบบวงจรควบคุมเพื่อควบคุมกระบวนการทำงานของปั๊มได้
4. จากการทดสอบระบบควบคุมกับชุดทดลองทางเคมี พบว่าระบบควบคุมที่ออกแบบขึ้นสามารถควบคุมกระบวนการให้ทำงานได้ตามที่ต้องการอย่างแม่นยำ การควบคุมมีความสะดวกสบายมากยิ่งขึ้น

5.2 ปัญหาที่พบในการทำงาน

1. ในการออกแบบแลตเตอร์ไดอะแกรม พบว่า PLC มีขีดความสามารถที่ไม่สามารถตอบสนองต่อความต้องการได้ คือ ไม่สามารถใช้ฟังก์ชันนับเวลา (Timer) ได้เกินกว่า 999.9 วินาที แต่ความต้องการในการใช้งานของฟังก์ชันนี้มีมากกว่า 10,000 วินาที
2. ขาดความรู้ในเรื่องการเชื่อมต่อและส่งข้อมูลระหว่างโปรแกรม Visual Basic 6.0 กับ PLC

5.3 การแก้ไขปัญหที่เกิดขึ้น

1. ศึกษาการเก็บข้อมูลของ PLC ให้ลึกซึ้งมากยิ่งขึ้น และทำการใช้ฟังก์ชันอื่นมาช่วยในการจัดการกับข้อมูลก่อนนำค่าเข้าสู่หน่วยความจำของส่วนของฟังก์ชันนับเวลา ซึ่งผลปรากฏว่า สามารถสั่งการให้นับเวลาทำงานได้มากขึ้นได้ตามที่ต้องการ
2. ศึกษาข้อมูลเพิ่มเติมในส่วนของการเชื่อมต่อกับอุปกรณ์ภายนอกทั้งส่วนของโปรแกรม Visual Basic 6.0 และส่วนของ PLC ซึ่งพบว่าจะต้องทำการเขียนโปรแกรมแปลงข้อมูลเป็นรหัส ASCII ก่อนส่งไปยังหน่วยความจำหรือตั้งกระบวนการทำงานของ PLC

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.4 ข้อเสนอแนะในการค้นคว้าและพัฒนา

ในการนำระบบควบคุมนี้ไปใช้งานเพื่อให้ครอบคลุมมากขึ้นอาจมีการพัฒนาระบบหรือวงจรเพื่อให้สามารถควบคุมกระบวนการได้อย่างครบถ้วนทั้งการควบคุมการเปิด-ปิด ควบคุมทิศทางการหมุนและการควบคุมความเร็ว ในส่วนของหน้าจออินเทอร์เฟซอาจมีการออกแบบตกแต่งให้มีความสวยงามน่าใช้มากยิ่งขึ้น อีกทั้งอาจมีการติดตั้งระบบเพื่อการป้อนข้อมูลกลับมาเพื่อการวิเคราะห์ทั้งจากอุปกรณ์ภายนอกป้อนเข้าสู่ PLC หรือจาก PLC ป้อนกลับมายังหน้าจออินเทอร์เฟซ ซึ่งในกรณีนี้ต้องขึ้นอยู่กับอุปกรณ์ที่เลือกใช้งานและคุณสมบัติของ PLC อีกทั้งมีการป้อนข้อมูลจากปั๊มกลับมาแสดงเพื่อให้สามารถตรวจสอบการทำงานของปั๊มได้ว่ามีความผิดปกติหรือเกิดการขัดข้องหรือไม่ เนื่องจากระบบในปัจจุบันเป็นระบบควบคุมแบบเปิด ทำให้ไม่สามารถตรวจสอบจากหน้าจออินเทอร์เฟซได้ว่าปั๊มทำงานจริงหรือไม่



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เอกสารอ้างอิง

- [1] ฉันทวุฒิ พิษผล, พิชิต สันติภูตานนท์, พร้อมเลิศ หล่อวิจิตร. คู่มือเรียน Visual Basic 6.0. พิมพ์ครั้งที่ 11. กรุงเทพมหานคร : โปรวิชั่น. 2547.
- [2] รศ.ธีรศิลป์ ทุมวิภาต, สุภาพร จำปาทอง. เรียนรู้ PLC ขั้นกลางด้วยตนเอง. กรุงเทพมหานคร : ซีเอ็ดยูเคชั่น. 2547.
- [3] ชาริน สิทธิธรรมชารี. สร้างโปรแกรมบนวินโดวส์ด้วย Visual Basic Version 6.0 ฉบับสมบูรณ์. พิมพ์ครั้งที่ 1. กรุงเทพมหานคร : สำนักพิมพ์ซัมพลีฟาย. 2553.



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาคผนวก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรมควบคุม

โปรแกรมวิชวลเบสิกนี้สร้างแอปพลิเคชันบนระบบปฏิบัติการ Window เพื่อสั่งการให้ PLC ทำตามโปรแกรมแลคเตอร์ไดอะแกรมซึ่งทำหน้าที่ควบคุมกระบวนการทำงานของกระบวนการ

***** ประกาศตัวแปร*****

Option Explicit

Public time1 As Long, Time2 As Long, Time3 As Long, Time4 As Long, Time5 As Long, Time6 As Long, Time7 As Long, Time8 As Long, Time9 As Long, Time10 As Long, Time11 As Long, Time12 As Long, Time16 As Long

Public xx1 As Long, xx2 As Long, xx3 As Long, xx4 As Long, xx5 As Long, xx6 As Long, xx7 As Long, xx8 As Long, xx9 As Long, xx10 As Long, xx11 As Long, xx12 As Long, x16 As Long

Public total As Integer

Dim ArrayHex1(20) As String, ArrayHex2(20) As String, ArrayHex3(20) As String, ArrayHex4(20) As String, ArrayHex5(20) As String, ArrayHex6(20) As String, ArrayHex7(20) As String, ArrayHex8(20) As String, ArrayHex9(20) As String, ArrayHex10(20) As String, 11 ArrayHex11(20) As String, ArrayHex12(20) As String, ArrayHex16(20) As String

Public FCS1 As String, FCS2 As String, FCS3 As String, FCS4 As String, FCS5 As String, FCS6 As String, FCS7 As String, FCS8 As String, FCS9 As String, FCS10 As String, FCS11 As String, FCS12 As String, FCS16 As String, Data As String, Data1 As String

Public HexTime1 As String, HexTime2 As String, HexTime3 As String, HexTime4 As String, HexTime5 As String, HexTime6 As String, HexTime7 As String, HexTime8 As String, HexTime9 As String, HexTime10 As String, HexTime11 As String, HexTime12 As String, HexTime16 As String

Public InitialHostlink1 As String, InitialHostlink2 As String, InitialHostlink3 As String, InitialHostlink4 As String, InitialHostlink5 As String, InitialHostlink6 As String, InitialHostlink7 As String, InitialHostlink8 As String, InitialHostlink9 As String, InitialHostlink10 As String, InitialHostlink11 As String, InitialHostlink12 As String, InitialHostlink16 As String

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้


```
Public x1 As Integer, y1 As Integer, x2 As Integer, y2 As Integer, x3 As Integer, y3 As
Integer, x4 As Integer, y4 As Integer, x5 As Integer, y5 As Integer, x6 As Integer, y6 As Integer,
x7 As Integer, y7 As Integer, x8 As Integer, y8 As Integer, x9 As Integer, y9 As Integer, x10 As
Integer, y10 As Integer, x11 As Integer, y11 As Integer, x12 As Integer, y12 As Integer, x13 As
Integer, y13 As Integer
```

```
Public Buf_Sum1 As Integer, Buf_Sum2 As Integer, Buf_Sum3 As Integer, Buf_Sum4
As Integer, Buf_Sum5 As Integer, Buf_Sum6 As Integer, Buf_Sum7 As Integer, Buf_Sum8 As
Integer, Buf_Sum9 As Integer, Buf_Sum10 As Integer, Buf_Sum11 As Integer, Buf_Sum12 As
Integer, Buf_Sum16 As Integer
```

```
Public Hostlink1 As String, Hostlink2 As String, Hostlink3 As String, Hostlink4 As
String, Hostlink5 As String, Hostlink6 As String, Hostlink7 As String, Hostlink8 As String,
Hostlink9 As String, Hostlink10 As String, Hostlink11 As String, Hostlink12 As String,
Hostlink16 As String
```

Private Sub Form_Load()

ตั้งค่าพอร์ตและกำหนดค่าต่างๆ

```
MSComm1.CommPort = 1
MSComm1.PortOpen = True
MSComm1.Settings = "9600,e,7,2"
MSComm1.Output = "@00SC0252*" & vbCr
MSComm1.Output = "@**" & vbCr
MSComm1.Output = "@00SC0353*" & vbCr
Form1.WindowState = 2
```

End Sub

Private Sub Command1_Click()

```
time1 = Val(Text1.Text) 'TEXT 1 : PUMP 1'
HexTime1 = Hex(time1)
If Len(HexTime1) = 1 Then
    HexTime1 = "000" + HexTime1
```

เอกสารนี้เป็นElseIf Len(HexTime1) = 2 Thenเพื่อการศึกษเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้นHexTime1 = "00" + HexTime1และตั้ง"ตรวจสอบจำนวนข้อมูลที่ส่งมา"มีการนำไปใช้


```

ElseIf Len(HexTime1) = 3 Then
    HexTime1 = "0" + HexTime1
ElseIf Len(HexTime1) = 4 Then
    HexTime1 = HexTime1
End If

''' ส่งค่าไปยังหน่วยความจำในรูปแบบ hostlink '''
InitialHostlink1 = "@" + "00" + "WD" + "0010" + HexTime1

''' คำนวณค่า FCS ของข้อมูล ซึ่งเป็นค่าที่ใช้ควบคุมความผิดพลาดของข้อมูล'''
y1 = Len(InitialHostlink1)
For x1 = 1 To y1
    ArrayHex1(x1) = Asc(Mid(InitialHostlink1, x1, 1))
Next x1
For x1 = 1 To y1
    Buf_Sum1 = Buf_Sum1 Xor ArrayHex1(x1) ''' คำนวณโดยใช้ XOR '''
    FCS1 = Hex(Buf_Sum1)
Next x1
Buf_Sum1 = 0
Hostlink1 = InitialHostlink1 + FCS1 + "*"
MSComm1.Output = Hostlink1 & vbCr ''' ส่งค่าไปยังหน่วยความจำ '''

''' ส่งข้อมูลอื่นๆ ไปยังหน่วยความจำต่างๆ '''

If Check1.Value = 0 Then ''' เมื่อไม่ได้กดปุ่ม Check1 '''
    MSComm1.Output = "@00WL000100005A*" & vbCr
End If

If Check1.Value = 1 Then ''' เมื่อกดปุ่ม Check1 '''
    MSComm1.Output = "@00WL000100015B*" & vbCr
End If

If Check17.Value = 1 Then ''' เมื่อกดปุ่ม Check17 '''
    MSComm1.Output = "@00WH000200015C*" & vbCr

```

เอกสารนี้เป็น**End If** ที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่**End Sub** ใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Private Sub Command2_Click()

Time2 = Val(Text2.Text) 'TEXT 2 : PUMP 2'

HexTime2 = Hex(Time2 * 10)

If Len(HexTime2) = 1 Then

HexTime2 = "000" + HexTime2

ElseIf Len(HexTime2) = 2 Then

HexTime2 = "00" + HexTime2

ElseIf Len(HexTime2) = 3 Then

HexTime2 = "0" + HexTime2

ElseIf Len(HexTime2) = 4 Then

HexTime2 = HexTime2

End If

InitialHostlink2 = "@" + "00" + "WD" + "0040" + HexTime2

y2 = Len(InitialHostlink2)

For x2 = 1 To y2

ArrayHex2(x2) = Asc(Mid(InitialHostlink2, x2, 1))

Next x2

For x2 = 1 To y2

Buf_Sum2 = Buf_Sum2 Xor ArrayHex2(x2)

FCS2 = Hex(Buf_Sum2)

Next x2

Buf_Sum2 = 0

Hostlink2 = InitialHostlink2 + FCS2 + "*"

MSComm1.Output = Hostlink2 & vbCr

If Check2.Value = 0 Then

MSComm1.Output = "@00WL000400005F*" & vbCr

End If

If Check2.Value = 1 Then

MSComm1.Output = "@00WL000400015E*" & vbCr

End If

If Check19.Value = 1 Then

MSComm1.Output = "@00WH000400015A*" & vbCr

End If

End Sub

Private Sub Command3_Click()

Time3 = Val(Text3.Text) 'TEXT 3 : PUMP 3'

HexTime3 = Hex(Time3 * 10)

If Len(HexTime3) = 1 Then

HexTime3 = "000" + HexTime3

ElseIf Len(HexTime3) = 2 Then

HexTime3 = "00" + HexTime3

ElseIf Len(HexTime3) = 3 Then

HexTime3 = "0" + HexTime3

ElseIf Len(HexTime3) = 4 Then

HexTime3 = HexTime3

End If

InitialHostlink3 = "@" + "00" + "WD" + "0070" + HexTime3

y3 = Len(InitialHostlink3)

For x3 = 1 To y3

ArrayHex3(x3) = Asc(Mid(InitialHostlink3, x3, 1))

Next x3

For x3 = 1 To y3

Buf_Sum3 = Buf_Sum3 Xor ArrayHex3(x3)

FCS3 = Hex(Buf_Sum3)

Next x3

Buf_Sum3 = 0

Hostlink3 = InitialHostlink3 + FCS3 + "*"

MSComm1.Output = Hostlink3 & vbCr

If Check3.Value = 0 Then

MSComm1.Output = "@00WL000700005C*" & vbCr

End If

If Check3.Value = 1 Then

MSComm1.Output = "@00WL000700015D*" & vbCr

เอกสารนี้เป็นเอกสารตัวอย่างเพื่อใช้ในการเรียนการสอนเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น หากมีข้อผิดพลาดประการใด ขออภัยเป็นอย่างสูง และจะรีบแก้ไขทันที


```

End If

If Check21.Value = 1 Then

    MSComm1.Output = "@00WH0006000158*" & vbCr

End If

End Sub

```

```
Private Sub Command4_Click()
```

```

    Time4 = Val(Text4.Text)                'TEXT 4 : PUMP 4'
    HexTime4 = Hex(Time4 * 10)
    If Len(HexTime4) = 1 Then
        HexTime4 = "000" + HexTime4
    ElseIf Len(HexTime4) = 2 Then
        HexTime4 = "00" + HexTime4
    ElseIf Len(HexTime4) = 3 Then
        HexTime4 = "0" + HexTime4
    ElseIf Len(HexTime4) = 4 Then
        HexTime4 = HexTime4
    End If
    InitialHostlink4 = "@" + "00" + "WD" + "0100" + HexTime4
    y4 = Len(InitialHostlink4)
    For x4 = 1 To y4
        ArrayHex4(x4) = Asc(Mid(InitialHostlink4, x4, 1))
    Next x4
    For x4 = 1 To y4
        Buf_Sum4 = Buf_Sum4 Xor ArrayHex4(x4)
        FCS4 = Hex(Buf_Sum4)
    Next x4
    Buf_Sum4 = 0
    Hostlink4 = InitialHostlink4 + FCS4 + "*"
    MSComm1.Output = Hostlink4 & vbCr

    If Check4.Value = 0 Then
        MSComm1.Output = "@00WL001100005B*" & vbCr
    End If
End Sub

```



```

End If

If Check4.Value = 1 Then
    MSComm1.Output = "@00WL001100015A*" & vbCr
End If

If Check23.Value = 1 Then
    MSComm1.Output = "@00WH0008000156*" & vbCr
End If

End Sub

```

```

Private Sub Command5_Click()
    Time5 = Val(Text5.Text)
    HexTime5 = Hex(Time5)
    If Len(HexTime5) = 1 Then
        HexTime5 = "000" + HexTime5
    ElseIf Len(HexTime5) = 2 Then
        HexTime5 = "00" + HexTime5
    ElseIf Len(HexTime5) = 3 Then
        HexTime5 = "0" + HexTime5
    ElseIf Len(HexTime5) = 4 Then
        HexTime5 = HexTime5
    End If
    InitialHostlink5 = "@" + "00" + "WD" + "0020" + HexTime5
    y5 = Len(InitialHostlink5)
    For x5 = 1 To y5
        ArrayHex5(x5) = Asc(Mid(InitialHostlink5, x5, 1))
    Next x5
    For x5 = 1 To y5
        Buf_Sum5 = Buf_Sum5 Xor ArrayHex5(x5)
    Next x5
    FCS5 = Hex(Buf_Sum5)
    Buf_Sum5 = 0
    Hostlink5 = InitialHostlink5 + FCS5 + "*"

```

เอกสารนี้เป็นเอกสารที่จัดทำขึ้นเพื่อการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงแก้ไขเอกสารนี้โดยเด็ดขาด และต้องส่งคืนเอกสารทุกครั้งที่มีการนำไปใช้


```

MSComm1.Output = Hostlink5 & vbCr
If Check5.Value = 0 Then
    MSComm1.Output = "@00WL0002000059*" & vbCr
End If
If Check5.Value = 1 Then
    MSComm1.Output = "@00WL0002000158*" & vbCr
End If
End Sub

```

```

Private Sub Command6_Click()
    Time6 = Val(Text6.Text) 'Text 6: PUMP 2
    HexTime6 = Hex(Time6 * 10)
    If Len(HexTime6) = 1 Then
        HexTime6 = "000" + HexTime6
    ElseIf Len(HexTime6) = 2 Then
        HexTime6 = "00" + HexTime6
    ElseIf Len(HexTime6) = 3 Then
        HexTime6 = "0" + HexTime6
    ElseIf Len(HexTime6) = 4 Then
        HexTime6 = HexTime6
    End If
    InitialHostlink6 = "@" + "00" + "WD" + "0050" + HexTime6
    y6 = Len(InitialHostlink6)
    For x6 = 1 To y6
        ArrayHex6(x6) = Asc(Mid(InitialHostlink6, x6, 1))
    Next x6
    For x6 = 1 To y6
        Buf_Sum6 = Buf_Sum6 Xor ArrayHex6(x6)
        FCS6 = Hex(Buf_Sum6)
    Next x6
    Buf_Sum6 = 0
    Hostlink6 = InitialHostlink6 + FCS6 + "*"

```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี การใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น ผู้ที่นำเอกสารนี้ไปเผยแพร่โดยไม่ได้รับอนุญาตจากเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้


```

MSComm1.Output = Hostlink6 & vbCr
If Check6.Value = 0 Then
    MSComm1.Output = "@00WL000500005E*" & vbCr
End If
If Check6.Value = 1 Then
    MSComm1.Output = "@00WL000500015F*" & vbCr
End If
End Sub

Private Sub Command7_Click()
    Time7 = Val(Text7.Text)                                'Text 7: PUMP 3.....
    HexTime7 = Hex(Time7 * 10)
    If Len(HexTime7) = 1 Then
        HexTime7 = "000" + HexTime7
    ElseIf Len(HexTime7) = 2 Then
        HexTime7 = "00" + HexTime7
    ElseIf Len(HexTime7) = 3 Then
        HexTime7 = "0" + HexTime7
    ElseIf Len(HexTime7) = 4 Then
        HexTime7 = HexTime7
    End If
    InitialHostlink7 = "@" + "00" + "WD" + "0080" + HexTime7
    y7 = Len(InitialHostlink7)
    For x7 = 1 To y7
        ArrayHex7(x7) = Asc(Mid(InitialHostlink7, x7, 1))
    Next x7
    For x7 = 1 To y7
        Buf_Sum7 = Buf_Sum7 Xor ArrayHex7(x7)
        FCS7 = Hex(Buf_Sum7)
    Next x7
    Buf_Sum7 = 0
    Hostlink7 = InitialHostlink7 + FCS7 + "*"

```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น

Hostlink7 = InitialHostlink7 + FCS7 + "*" จึงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้


```

MSComm1.Output = Hostlink7 & vbCr
If Check7.Value = 0 Then
    MSComm1.Output = "@00WL0008000053*" & vbCr
End If
If Check7.Value = 1 Then
    MSComm1.Output = "@00WL0008000152*" & vbCr
End If
End Sub

```

```

Private Sub Command8_Click()
    Time8 = Val(Text8.Text) 'TEXT 8 : PUMP 4
    HexTime8 = Hex(Time8 * 10)
    If Len(HexTime8) = 1 Then
        HexTime8 = "000" + HexTime8
    ElseIf Len(HexTime8) = 2 Then
        HexTime8 = "00" + HexTime8
    ElseIf Len(HexTime8) = 3 Then
        HexTime8 = "0" + HexTime8
    ElseIf Len(HexTime8) = 4 Then
        HexTime8 = HexTime8
    End If
    InitialHostlink8 = "@" + "00" + "WD" + "0110" + HexTime8
    y8 = Len(InitialHostlink8)
    For x8 = 1 To y8
        ArrayHex8(x8) = Asc(Mid(InitialHostlink8, x8, 1))
    Next x8
    For x8 = 1 To y8
        Buf_Sum8 = Buf_Sum8 Xor ArrayHex8(x8)
        FCS8 = Hex(Buf_Sum8)
    Next x8

```

เอกสารนี้เป็นเอกสารที่จัดทำขึ้นเพื่อการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น

Buf_Sum8 = 0
Hostlink8 = InitialHostlink8 + FCS8 + "*" จึงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้


```

MSComm1.Output = Hostlink8 & vbCr
If Check8.Value = 0 Then
    MSComm1.Output = "@00WL0012000058*" & vbCr
End If
If Check8.Value = 1 Then
    MSComm1.Output = "@00WL0012000159*" & vbCr
End If
End Sub

Private Sub Command9_Click()
    Time9 = Val(TextBox9.Text)
    HexTime9 = Hex(Time9)
    If Len(HexTime9) = 1 Then
        HexTime9 = "000" + HexTime9
    ElseIf Len(HexTime9) = 2 Then
        HexTime9 = "00" + HexTime9
    ElseIf Len(HexTime9) = 3 Then
        HexTime9 = "0" + HexTime9
    ElseIf Len(HexTime9) = 4 Then
        HexTime9 = HexTime9
    End If
    InitialHostlink9 = "@" + "00" + "WD" + "0030" + HexTime9
    y9 = Len(InitialHostlink9)
    For x9 = 1 To y9
        ArrayHex9(x9) = Asc(Mid(InitialHostlink9, x9, 1))
    Next x9
    For x9 = 1 To y9
        Buf_Sum9 = Buf_Sum9 Xor ArrayHex9(x9)
        FCS9 = Hex(Buf_Sum9)
    Next x9
    Buf_Sum9 = 0
    Hostlink9 = InitialHostlink9 + FCS9 + "*"

```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น ลิขสิทธิ์นี้สงวนไว้สำหรับเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้


```

MSComm1.Output = Hostlink9 & vbCr
If Check9.Value = 0 Then
    MSComm1.Output = "@00WL0003000058*" & vbCr
End If
If Check9.Value = 1 Then
    MSComm1.Output = "@00WL0003000159*" & vbCr
End If
End Sub

```

```

Private Sub Command10_Click()
    Time10 = Val(Text10.Text)
    HexTime10 = Hex(Time10 * 10)
    If Len(HexTime10) = 1 Then
        HexTime10 = "000" + HexTime10
    ElseIf Len(HexTime10) = 2 Then
        HexTime10 = "00" + HexTime10
    ElseIf Len(HexTime10) = 3 Then
        HexTime10 = "0" + HexTime10
    ElseIf Len(HexTime10) = 4 Then
        HexTime10 = HexTime10
    End If
    InitialHostlink10 = "@" + "00" + "WD" + "0060" + HexTime10
    y10 = Len(InitialHostlink10)
    For x10 = 1 To y10
        ArrayHex10(x10) = Asc(Mid(InitialHostlink10, x10, 1))
    Next x10
    For x10 = 1 To y10
        Buf_Sum10 = Buf_Sum10 Xor ArrayHex10(x10)
        FCS10 = Hex(Buf_Sum10)
    Next x10
    Buf_Sum10 = 0
    Hostlink10 = InitialHostlink10 + FCS10 + "*"

```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์การใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องแจ้งถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้


```

MSComm1.Output = Hostlink10 & vbCr
If Check10.Value = 0 Then
    MSComm1.Output = "@00WL000600005D*" & vbCr
End If
If Check10.Value = 1 Then
    MSComm1.Output = "@00WL000600015C*" & vbCr
End If
End Sub

Private Sub Command11_Click()
    Time11 = Val(Text11.Text) ' TEXT 11 : PUMP3*****
    HexTime11 = Hex(Time11 * 10)
    If Len(HexTime11) = 1 Then
        HexTime11 = "000" + HexTime11
    ElseIf Len(HexTime11) = 2 Then
        HexTime11 = "00" + HexTime11
    ElseIf Len(HexTime11) = 3 Then
        HexTime11 = "0" + HexTime11
    ElseIf Len(HexTime11) = 4 Then
        HexTime11 = HexTime11
    End If
    InitialHostlink11 = "@" + "00" + "WD" + "0090" + HexTime11
    y11 = Len(InitialHostlink11)
    For x11 = 1 To y11
        ArrayHex11(x11) = Asc(Mid(InitialHostlink11, x11, 1))
    Next x11
    For x11 = 1 To y11
        Buf_Sum11 = Buf_Sum11 Xor ArrayHex11(x11)
        FCS11 = Hex(Buf_Sum11)
    Next x11
    Buf_Sum11 = 0
    Hostlink11 = InitialHostlink11 + FCS11 + "*"

```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น ลิขสิทธิ์นี้สงวนไว้สำหรับอาจารย์ผู้จัดทำเอกสารนี้ และเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้


```

MSComm1.Output = Hostlink11 & vbCr
If Check11.Value = 0 Then
    MSComm1.Output = "@00WL0009000052*" & vbCr
End If
If Check11.Value = 1 Then
    MSComm1.Output = "@00WL00090000153*" & vbCr
End If
End Sub

```

```

Private Sub Command12_Click()
    Time12 = Val(Text12.Text) ' TEXT 12 : PUMP 4
    HexTime12 = Hex(Time12 * 10)
    If Len(HexTime12) = 1 Then
        HexTime12 = "000" + HexTime12
    ElseIf Len(HexTime12) = 2 Then
        HexTime12 = "00" + HexTime12
    ElseIf Len(HexTime12) = 3 Then
        HexTime12 = "0" + HexTime12
    ElseIf Len(HexTime12) = 4 Then
        HexTime12 = HexTime12
    End If
    InitialHostlink12 = "@" + "00" + "WD" + "0120" + HexTime12
    y12 = Len(InitialHostlink12)
    For x12 = 1 To y12
        ArrayHex12(x12) = Asc(Mid(InitialHostlink12, x12, 1))
    Next x12
    For x12 = 1 To y12
        Buf_Sum12 = Buf_Sum12 Xor ArrayHex12(x12)
        FCS12 = Hex(Buf_Sum12)
    Next x12
    Buf_Sum12 = 0
    Hostlink12 = InitialHostlink12 + FCS12 + "*"

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามเปิดเผยข้อมูลและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้


```
MSComm1.Output = Hostlink12 & vbCr
```

```
If Check12.Value = 0 Then
```

```
    MSComm1.Output = "@00WL0013000059*" & vbCr
```

```
End If
```

```
If Check12.Value = 1 Then
```

```
    MSComm1.Output = "@00WL0013000158*" & vbCr
```

```
End If
```

```
End Sub
```

```
"""" RUN BUTTON """"""""
```

```
Private Sub Command13_Click()
```

```
    MSComm1.Output = "@00WL000000015A*" & vbCr """" ส่งค่าไปยังหน่วยความจำ """"
```

```
    If Check13.Value = 1 Then """" เมื่อเลือกปุ่ม Check 13 (PUMP1) """"
```

```
        MSComm1.Output = "@00WH000100015F*" & vbCr
```

```
        ProgressBar1.Min = 0 """" กำหนดค่าให้ Progressbar """"
```

```
        ProgressBar1.Max = Text1.Text
```

```
        Timer1.Interval = 1000
```

```
        Timer1.Enabled = True
```

```
        xx1 = 0
```

```
        time1 = Val(Text1.Text)
```

```
    End If
```

```
    If Check14.Value = 1 Then """" เมื่อเลือกปุ่ม Check 14 (PUMP2) """"
```

```
        MSComm1.Output = "@00WH000300015D*" & vbCr """" ส่งค่าไปยังหน่วยความจำ """"
```

```
        ProgressBar2.Min = 0 """" กำหนดค่าต่ำสุดให้ Progress bar """"
```

```
        ProgressBar2.Max = Text2.Text """" กำหนดค่าสูงสุดให้ Progress bar """"
```

```
        Timer2.Interval = 1000
```

```
        Timer2.Enabled = True
```

```
        xx2 = 0
```

```
        Time2 = Val(Text2.Text)
```

```
    End If
```

```
    If Check15.Value = 1 Then """" เมื่อเลือกปุ่ม Check 15 (PUMP3) """"
```

```
        MSComm1.Output = "@00WH000500015B*" & vbCr
```



```

ProgressBar3.Min = 0

ProgressBar3.Max = Text3.Text

Timer3.Interval = 1000

Timer3.Enabled = True

xx3 = 0

Time3 = Val(Text3.Text)

End If

If Check16.Value = 1 Then                                     "" เมื่อเลือกปุ่ม Check 16 (PUMP4) ""
    MSComm1.Output = "@00WH0007000159*" & vbCrLf
    ProgressBar4.Min = 0
    ProgressBar4.Max = Text4.Text
    Timer4.Interval = 1000
    Timer4.Enabled = True
    xx4 = 0
    Time4 = Val(Text4.Text)
End If
Text13.BorderStyle = 0
Text14.BorderStyle = 0
Text15.BorderStyle = 0
Text13.BackColor = &H8000000F
Text14.BackColor = &H8000000F
Text15.BackColor = &H8000000F

End Sub

```

```

Private Sub Command14_Click()

```

```

    MSComm1.Output = "@00WL001000015B*" & vbCrLf

    ProgressBar1 = xx1

    ProgressBar2 = xx2

    ProgressBar3 = xx3

    ProgressBar4 = xx4

    ProgressBar5 = xx5

    ProgressBar6 = xx6

```



```

ProgressBar7 = xx7
ProgressBar8 = xx8
ProgressBar9 = xx9
ProgressBar10 = xx10
ProgressBar11 = xx11
ProgressBar12 = xx12      'STOP BUTTON'
Timer1.Enabled = False
Timer2.Enabled = False
Timer3.Enabled = False
Timer4.Enabled = False
Timer5.Enabled = False
Timer6.Enabled = False
Timer7.Enabled = False
Timer8.Enabled = False
Timer9.Enabled = False
Timer10.Enabled = False
Timer11.Enabled = False
Timer12.Enabled = False
If Check13.Value = 1 Then
    MSComm1.Output = "@00WH000100005E*" & vbCr
End If
If Check14.Value = 1 Then
    MSComm1.Output = "@00WH000300005C*" & vbCr
End If
If Check15.Value = 1 Then
    MSComm1.Output = "@00WH000500005A*" & vbCr
End If
If Check16.Value = 1 Then
    MSComm1.Output = "@00WH0007000058*" & vbCr
End If

```

End Sub

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Private Sub Command15_Click()

If Check17.Value = 1 Or Check18.Value = 1 Or Check17.Value = 0 Or Check18.Value = 0 Then

MSComm1.Output = "@00WH000200005D*" & vbCr

End If

If Check19.Value = 1 Or Check20.Value = 1 Or Check19.Value = 0 Or Check20.Value = 0 Then

MSComm1.Output = "@00WH000400005B*" & vbCr

End If

If Check21.Value = 1 Or Check22.Value = 1 Or Check21.Value = 0 Or Check22.Value = 0 Then

MSComm1.Output = "@00WH0006000059*" & vbCr

End If

If Check23.Value = 1 Or Check24.Value = 1 Or Check23.Value = 0 Or Check24.Value = 0 Then

MSComm1.Output = "@00WH0008000057*" & vbCr

End If

MSComm1.Output = "@00WL000000005B*" & vbCr

MSComm1.Output = "@00WL001000005A*" & vbCr

ProgressBar1 = 0

ProgressBar2 = 0

ProgressBar3 = 0

ProgressBar4 = 0

ProgressBar5 = 0

'RESET BUTTON'*****

ProgressBar6 = 0

ProgressBar7 = 0

ProgressBar8 = 0

ProgressBar9 = 0

ProgressBar10 = 0

ProgressBar11 = 0

ProgressBar12 = 0

Text1.Text = ""

เอกสารนี้เป็นเอกสารสงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Text2.Text = ""

Text3.Text = ""

Text4.Text = ""

Text5.Text = ""

Text6.Text = ""

Text7.Text = ""

Text8.Text = ""

Text9.Text = ""

Text10.Text = ""

Text11.Text = ""

Text12.Text = ""

Time1 = 0

Time2 = 0

Time3 = 0

Time4 = 0

Time5 = 0

Time6 = 0

Time7 = 0

Time8 = 0

Time9 = 0

Time10 = 0

Time11 = 0

Time12 = 0

Check1.Value = 0

Check2.Value = 0

Check3.Value = 0

Check4.Value = 0

Check5.Value = 0

Check6.Value = 0

Check7.Value = 0

Check8.Value = 0

Check9.Value = 0



เอกสารนี้เป็นเอกสารที่เผยแพร่ไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Check10.Value = 0

Check12.Value = 0

Check11.Value = 0

Check13.Value = 0

Check14.Value = 0

Check15.Value = 0

Check16.Value = 0

Check17.Value = 0

Check18.Value = 0

Check19.Value = 0

Check20.Value = 0

Check21.Value = 0

Check22.Value = 0

Check23.Value = 0

Check24.Value = 0

Label1.BackColor = vbWhite

Label2.BackColor = vbWhite

Label3.BackColor = vbWhite

Text13.BorderStyle = 1

Text14.BorderStyle = 1

Text15.BorderStyle = 1

Text13.BackColor = vbWhite

Text14.BackColor = vbWhite

Text15.BackColor = vbWhite

total = 0

End Sub

Private Sub Text1_KeyPress(KeyAscii As Integer)

If Not IsNumeric(Chr(KeyAscii)) Then

If Not KeyAscii = vbKeyBack Then

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้ในการเรียนการสอนเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้


```

        KeyAscii = 0
    End If
End If

End Sub

```

```

Private Sub Text2_KeyPress(KeyAscii As Integer)

```

```

    If Not IsNumeric(Chr(KeyAscii)) Then
        If Not KeyAscii = vbKeyBack Then
            KeyAscii = 0
        End If
    End If
End Sub

```

```

Private Sub Text3_KeyPress(KeyAscii As Integer)

```

```

    If Not IsNumeric(Chr(KeyAscii)) Then
        If Not KeyAscii = vbKeyBack Then
            KeyAscii = 0
        End If
    End If
End Sub

```

```

Private Sub Text4_KeyPress(KeyAscii As Integer)

```

```

    If Not IsNumeric(Chr(KeyAscii)) Then
        If Not KeyAscii = vbKeyBack Then
            KeyAscii = 0
        End If
    End If

```

```

End Sub

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้


```

Private Sub Text5_KeyPress(KeyAscii As Integer)
    If Not IsNumeric(Chr(KeyAscii)) Then
        If Not KeyAscii = vbKeyBack Then
            KeyAscii = 0
        End If
    End If
End Sub

```

```

Private Sub Text6_KeyPress(KeyAscii As Integer)
    If Not IsNumeric(Chr(KeyAscii)) Then
        If Not KeyAscii = vbKeyBack Then
            KeyAscii = 0
        End If
    End If
End Sub

```

```

Private Sub Text7_KeyPress(KeyAscii As Integer)
    If Not IsNumeric(Chr(KeyAscii)) Then
        If Not KeyAscii = vbKeyBack Then
            KeyAscii = 0
        End If
    End If
End Sub

```

```

Private Sub Text8_KeyPress(KeyAscii As Integer)
    If Not IsNumeric(Chr(KeyAscii)) Then
        If Not KeyAscii = vbKeyBack Then
            KeyAscii = 0
        End If
    End If
End Sub

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้


```
Private Sub Text9_KeyPress(KeyAscii As Integer)
```

```
    If Not IsNumeric(Chr(KeyAscii)) Then
```

```
        If Not KeyAscii = vbKeyBack Then
```

```
            KeyAscii = 0
```

```
        End If
```

```
    End If
```

```
End Sub
```

```
Private Sub Text10_KeyPress(KeyAscii As Integer)
```

```
    If Not IsNumeric(Chr(KeyAscii)) Then
```

```
        If Not KeyAscii = vbKeyBack Then
```

```
            KeyAscii = 0
```

```
        End If
```

```
    End If
```

```
End Sub
```

```
Private Sub Text11_KeyPress(KeyAscii As Integer)
```

```
    If Not IsNumeric(Chr(KeyAscii)) Then
```

```
        If Not KeyAscii = vbKeyBack Then
```

```
            KeyAscii = 0
```

```
        End If
```

```
    End If
```

```
End Sub
```

```
Private Sub Text12_KeyPress(KeyAscii As Integer)
```

```
    If Not IsNumeric(Chr(KeyAscii)) Then
```

```
        If Not KeyAscii = vbKeyBack Then
```

```
            KeyAscii = 0
```

```
        End If
```

```
    End If
```

```
End Sub
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้


```

Private Sub Timer1_Timer()
    xx1 = xx1 + 1
    ProgressBar1.Value = xx1
    If xx1 >= time1 Then
        Timer1.Enabled = False
        ProgressBar1.Value = 0
        ProgressBar5.Min = 0
        ProgressBar5.Max = Text5.Text
        Timer5.Interval = 1000
        Timer5.Enabled = True
        xx5 = 0
        Time5 = Val(Text5.Text)
    End If
End Sub

```

```

Private Sub Timer2_Timer()
    xx2 = xx2 + 1
    ProgressBar2.Value = xx2
    If xx2 >= Time2 Then
        Timer2.Enabled = False
        ProgressBar2.Value = 0
        ProgressBar6.Min = 0
        ProgressBar6.Max = Text6.Text
        Timer6.Interval = 1000
        Timer6.Enabled = True
        xx6 = 0
        Time6 = Val(Text6.Text)
    End If

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้ในการเรียนการสอนเพื่อจุดประสงค์เฉพาะเท่านั้น ไม่ควรนำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้


```

"Status when Pump 1,2,3,4 is used step 1"

If Check14.Value = 1 And Check15.Value = 1 And Check16.Value = 1 Then
    If Timer2.Enabled = True Or False Or Timer3.Enabled = True Or False Or
    Timer4.Enabled = True Or False Then
        Label1.BackColor = vbGreen
    End If
End If

If Check14.Value = 1 And Check15.Value = 1 And Check16.Value = 1 Then
    If Timer2.Enabled = False And Timer3.Enabled = False And Timer4.Enabled =
    False Then
        Label1.BackColor = vbRed
    End If
End If

"Status when Pump 1,2,3 is used step 1"

If Check14.Value = 1 And Check15.Value = 1 Then
    If Timer2.Enabled = True Or False Or Timer3.Enabled = True Or False Then
        Label1.BackColor = vbGreen
    End If
End If

If Check14.Value = 1 And Check15.Value = 1 Then
    If Timer2.Enabled = False And Timer3.Enabled = False Then
        Label1.BackColor = vbRed
    End If
End If

"Status when Pump 1,2 is used step 1"

If Check14.Value = 1 Then
    If Timer2.Enabled = True Or False Then
        Label1.BackColor = vbGreen
    End If
End If

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้


```

If Check14.Value = 1 Then
    If Timer2.Enabled = False Then
        Label1.BackColor = vbRed
    End If
End If
End Sub

```

```

Private Sub Timer3_Timer()

```

```

    xx3 = xx3 + 1
    ProgressBar3.Value = xx3
    If xx3 >= Time3 Then
        Timer3.Enabled = False
        ProgressBar3.Value = 0
        ProgressBar7.Min = 0
        ProgressBar7.Max = Text7.Text
        Timer7.Interval = 1000
        Timer7.Enabled = True
        xx7 = 0
        Time7 = Val(Text7.Text)
    End If

```

```

    ..... Status when Pump 1,2,3,4 is used step 1 .....

```

```

If Check14.Value = 1 And Check15.Value = 1 And Check16.Value = 1 Then
    If Timer2.Enabled = True Or False Or Timer3.Enabled = True Or False Or
    Timer4.Enabled = True Or False Then
        Label1.BackColor = vbGreen
    End If
End If

```

```

If Check14.Value = 1 And Check15.Value = 1 And Check16.Value = 1 Then

```

```

    If Timer2.Enabled = False And Timer3.Enabled = False And Timer4.Enabled =
    False Then

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        Label1.BackColor = vbRed
    End If
End If

```


End If

End If

"""""""""" Status when Pump 1,2,3 is used step 1""""""""""

If Check14.Value = 1 And Check15.Value = 1 Then

If Timer2.Enabled = True Or False Or Timer3.Enabled = True Or False Then

Label1.BackColor = vbGreen

End If

End If

If Check14.Value = 1 And Check15.Value = 1 Then

If Timer2.Enabled = False And Timer3.Enabled = False Then

Label1.BackColor = vbRed

End If

End If

End Sub

Private Sub Timer4_Timer()

xx4 = xx4 + 1

ProgressBar4.Value = xx4

If xx4 >= Time4 Then

Timer4.Enabled = False

ProgressBar4.Value = 0

ProgressBar8.Min = 0

ProgressBar8.Max = Text8.Text

Timer8.Interval = 1000

Timer8.Enabled = True

xx8 = 0

Time8 = Val(Text8.Text)

End If

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้


```

"Status when Pump 1,2,3,4 is used step 1"
If Check14.Value = 1 And Check15.Value = 1 And Check16.Value = 1 Then
    If Timer2.Enabled = True Or False Or Timer3.Enabled = True Or False Or
    Timer4.Enabled = True Or False Then
        Label1.BackColor = vbGreen
    End If
End If

If Check14.Value = 1 And Check15.Value = 1 And Check16.Value = 1 Then
    If Timer2.Enabled = False And Timer3.Enabled = False And Timer4.Enabled =
    False Then
        Label1.BackColor = vbRed
    End If
End If
End Sub

Private Sub Timer5_Timer()
    xx5 = xx5 + 1
    ProgressBar5.Value = xx5
    If xx5 >= Time5 Then
        Timer5.Enabled = False
        ProgressBar5.Value = 0
        ProgressBar9.Min = 0
        ProgressBar9.Max = Text9.Text
        Timer9.Interval = 1000
        Timer9.Enabled = True
        xx9 = 0
        Time9 = Val(Text9.Text)
    End If
End Sub

Private Sub Timer6_Timer()
    xx6 = xx6 + 1

```



```

        ProgressBar6.Value = xx6
    If xx6 >= Time6 Then
        Timer6.Enabled = False
        ProgressBar6.Value = 0
        ProgressBar10.Min = 0
        ProgressBar10.Max = Text10.Text
        Timer10.Interval = 1000
        Timer10.Enabled = True
        xx10 = 0
        Time10 = Val(Text10.Text)
    End If

    "Status when Pump 1,2,3,4 is used step 2"
    If Check14.Value = 1 And Check15.Value = 1 And Check16.Value = 1 Then
        If Timer6.Enabled = True Or False Or Timer7.Enabled = True Or False Or
        Timer8.Enabled = True Or False Then
            Label2.BackColor = vbGreen
        End If
    End If

    If Check14.Value = 1 And Check15.Value = 1 And Check16.Value = 1 Then
        If Timer6.Enabled = False And Timer7.Enabled = False And Timer8.Enabled =
        False Then
            Label2.BackColor = vbRed
        End If
    End If

```

"Status when Pump 1,2,3 is used step 2"

```

If Check14.Value = 1 And Check15.Value = 1 Then

```

```

    If Timer6.Enabled = True Or False Or Timer7.Enabled = True Or False Then

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้ในการเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น ถือว่าห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    End If

```


End If

If Check14.Value = 1 And Check15.Value = 1 Then

 If Timer6.Enabled = False And Timer7.Enabled = False Then

 Label2.BackColor = vbRed

 End If

End If

"""""""" Status when Pump 1,2 is used step 2""""""""

If Check14.Value = 1 Then

 If Timer6.Enabled = True Or False Then

 Label2.BackColor = vbGreen

 End If

End If

If Check14.Value = 1 Then

 If Timer6.Enabled = False Then

 Label2.BackColor = vbRed

 End If

End If

End Sub

Private Sub Timer7_Timer()

 xx7 = xx7 + 1

 ProgressBar7.Value = xx7

 If xx7 >= Time7 Then

 Timer7.Enabled = False

 ProgressBar7.Value = 0

 ProgressBar11.Min = 0

 ProgressBar11.Max = Text11.Text

 Timer11.Interval = 1000

 Timer11.Enabled = True

 xx11 = 0

 Time11 = Val(Text11.Text)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามนำข้อมูลไปเผยแพร่หรือหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

End If

"""""""" Status when Pump 1,2,3,4 is used step 2""""""""

If Check14.Value = 1 And Check15.Value = 1 And Check16.Value = 1 Then

If Timer6.Enabled = True Or False Or Timer7.Enabled = True Or False Or

Timer8.Enabled = True Or False Then

Label2.BackColor = vbGreen

End If

End If

If Check14.Value = 1 And Check15.Value = 1 And Check16.Value = 1 Then

If Timer6.Enabled = False And Timer7.Enabled = False And Timer8.Enabled =

False Then

Label2.BackColor = vbRed

End If

End If

"""""""" Status when Pump 1,2,3 is used step 2""""""""

If Check14.Value = 1 And Check15.Value = 1 Then

If Timer6.Enabled = True Or False Or Timer7.Enabled = True Or False Then

Label2.BackColor = vbGreen

End If

End If

If Check14.Value = 1 And Check15.Value = 1 Then

If Timer6.Enabled = False And Timer7.Enabled = False Then

Label2.BackColor = vbRed

End If

End If

End Sub

Private Sub Timer8_Timer()

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ProgressBar8.Value = xx8


```

If xx8 >= Time8 Then
    Timer8.Enabled = False
    ProgressBar8.Value = 0
    ProgressBar12.Min = 0
    ProgressBar12.Max = Text12.Text
    Timer12.Interval = 1000
    Timer12.Enabled = True
    xx12 = 0
    Time12 = Val(Text12.Text)
End If

"Status when Pump 1,2,3,4 is used step 2"
If Check14.Value = 1 And Check15.Value = 1 And Check16.Value = 1 Then
    If Timer6.Enabled = True Or False Or Timer7.Enabled = True Or False Or
    Timer8.Enabled = True Or False Then
        Label2.BackColor = vbGreen
    End If
End If

If Check14.Value = 1 And Check15.Value = 1 And Check16.Value = 1 Then
    If Timer6.Enabled = False And Timer7.Enabled = False And Timer8.Enabled =
    False Then
        Label2.BackColor = vbRed
    End If
End If

End Sub

```

```
Private Sub Timer9_Timer()
```

```
    xx9 = xx9 + 1
```

```
    ProgressBar9.Value = xx9
```

```
    If xx9 >= Time9 Then
```

```
        Timer9.Enabled = False
```

```
        ProgressBar1.Value = 0
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้


```

ProgressBar2.Value = 0
ProgressBar3.Value = 0
ProgressBar4.Value = 0
ProgressBar5.Value = 0
ProgressBar6.Value = 0
ProgressBar7.Value = 0
ProgressBar8.Value = 0
ProgressBar9.Value = 0
ProgressBar10.Value = 0
ProgressBar11.Value = 0
ProgressBar12.Value = 0
Timer1.Enabled = False
Timer2.Enabled = False
Timer3.Enabled = False
Timer4.Enabled = False
Timer5.Enabled = False
Timer6.Enabled = False
Timer7.Enabled = False
Timer8.Enabled = False
Timer9.Enabled = False
Timer10.Enabled = False
Timer11.Enabled = False
Timer12.Enabled = False
Label1.BackColor = vbRed
Label2.BackColor = vbRed
Label3.BackColor = vbRed

```

End If

End Sub

Private Sub Timer10_Timer()

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ProgressBar10.Value = xx10


```

If xx10 >= Time10 Then
    Timer10.Enabled = False
    ProgressBar10.Value = 0
    Timer2.Interval = 1000
    Timer2.Enabled = True
    xx2 = 0
    Time2 = Val(Text2.Text)
End If

```

***** Status when Pump 1,2,3,4 is used step 3 *****

```

If Check14.Value = 1 And Check15.Value = 1 And Check16.Value = 1 Then
    If Timer10.Enabled = True Or False Or Timer11.Enabled = True Or False Or
    Timer12.Enabled = True Or False Then
        Label3.BackColor = vbGreen
    End If
End If

If Check14.Value = 1 And Check15.Value = 1 And Check16.Value = 1 Then
    If Timer10.Enabled = False And Timer11.Enabled = False And
    Timer12.Enabled = False Then
        Label3.BackColor = vbRed
    End If
End If

```

***** Status when Pump 1,2,3 is used step 3 *****

```

If Check14.Value = 1 And Check15.Value = 1 Then
    If Timer10.Enabled = True Or False Or Timer11.Enabled = True Or False Then
        Label3.BackColor = vbGreen
    End If

```

End If

```

If Check14.Value = 1 And Check15.Value = 1 Then

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องแจ้งถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้


```

If Timer10.Enabled = False And Timer11.Enabled = False Then
    Label3.BackColor = vbRed
End If
End If

"""""""" Status when Pump 1,2 is used step 3""""""""

If Check14.Value = 1 Then
    If Timer10.Enabled = True Or False Then
        Label3.BackColor = vbGreen
    End If
End If

If Check14.Value = 1 Then
    If Timer10.Enabled = False Then
        Label3.BackColor = vbRed
    End If
End If
End Sub

Private Sub Timer11_Timer()
    xx11 = xx11 + 1
    ProgressBar11.Value = xx11
    If xx11 >= Time11 Then
        Timer11.Enabled = False
        ProgressBar11.Value = 0
        Timer3.Interval = 1000
        Timer3.Enabled = True
        xx3 = 0
        Time3 = Val(Text3.Text)
    End If

```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้สำหรับใช้ในวงจำกัดเท่านั้น ไม่ให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น ถือว่าให้ข้อมูลเบื้องต้นเท่านั้น และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้


```

If Timer10.Enabled = True Or False Or Timer11.Enabled = True Or False Or
Timer12.Enabled = True Or False Then
    Label3.BackColor = vbGreen
End If
End If
If Check14.Value = 1 And Check15.Value = 1 And Check16.Value = 1 Then
    If Timer10.Enabled = False And Timer11.Enabled = False And
Timer12.Enabled = False Then
        Label3.BackColor = vbRed
    End If
End If
End If

"Status when Pump 1,2,3 is used step 3"
If Check14.Value = 1 And Check15.Value = 1 Then
    If Timer10.Enabled = True Or False Or Timer11.Enabled = True Or False Then
        Label3.BackColor = vbGreen
    End If
End If
If Check14.Value = 1 And Check15.Value = 1 Then
    If Timer10.Enabled = False And Timer11.Enabled = False Then
        Label3.BackColor = vbRed
    End If
End If
End Sub

Private Sub Timer12_Timer()
    xx12 = xx12 + 1
    ProgressBar12.Value = xx12
    If xx12 >= Time12 Then
        Timer12.Enabled = False
        ProgressBar12.Value = 0
        Timer4.Interval = 1000
    End If
End Sub

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับอาจารย์ใช้ในการสอนเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้


```

Timer4.Enabled = True

xx4 = 0

Time4 = Val(Text4.Text)

End If

"Status when Pump 1,2,3,4 is used step 3"

If Check14.Value = 1 And Check15.Value = 1 And Check16.Value = 1 Then

    If Timer10.Enabled = True Or False Or Timer11.Enabled = True Or False Or
    Timer12.Enabled = True Or False Then

        Label3.BackColor = vbGreen

    End If

End If

If Check14.Value = 1 And Check15.Value = 1 And Check16.Value = 1 Then

    If Timer10.Enabled = False And Timer11.Enabled = False And
    Timer12.Enabled = False Then

        Label3.BackColor = vbRed

    End If

End If

End Subol

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



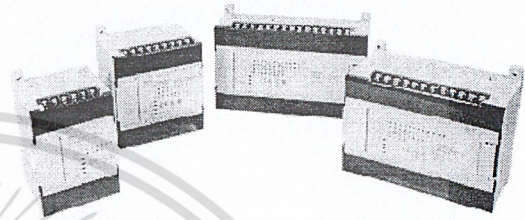
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

PLC รุ่น CPM1A

OMRON

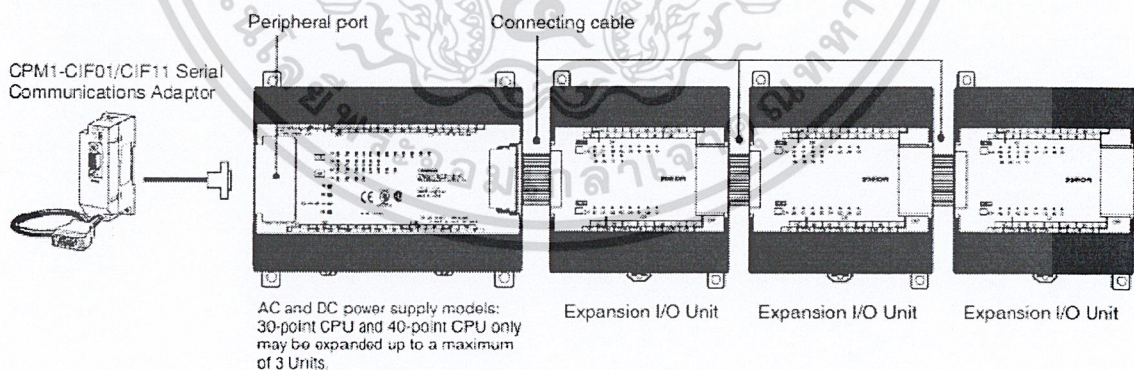
Micro Programmable Controller CPM1A

The CPM1A series micro controllers solve both basic and semi-complex applications. The brick style models include DC inputs/transistor or relay outputs to meet your design requirements. The base I/O for the CPUs ranges from 10, 20, 30, and 40 I/O points with maximum expansion to 100 I/O. Specialized expansion modules include mixed analog I/O, temperature sensor inputs and serial communications.



- 10, 20, 30 and 40 point I/O CPUs
- Expandable up to 100 I/O points
- Peripheral communications port built in
- DC input models
- Analog expansion modules available
- Temperature sensor input expansion modules available
- Auxiliary 24 VDC supply (AC type only)
- Relay or Transistor outputs
- UL, CSA, CE approvals

Basic Configuration



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Ordering Information

■ CPU

Stock Note: Shaded models are normally stocked.

Number of I/O terminals	Inputs	Outputs	Power supply	Part number		
				Relay output	Transistor output	
					Sink type	Source type
10	6 DC points	4 points	AC	CPM1A-10CDR-A-V1	CPM1A-10CDT-A-V1	CPM1A-10CDT1-A-V1
			DC	CPM1A-10CDR-D-V1	CPM1A-10CDT-D-V1	CPM1A-10CDT1-D-V1
20	12 DC points	8 points	AC	CPM1A-20CDR-A-V1	CPM1A-20CDT-A-V1	CPM1A-20CDT1-A-V1
			DC	CPM1A-20CDR-D-V1	CPM1A-20CDT-D-V1	CPM1A-20CDT1-D-V1
30	18 DC points	12 points	AC	CPM1A-30CDR-A-V1	CPM1A-30CDT-A-V1	CPM1A-30CDT1-A-V1
			DC	CPM1A-30CDR-D-V1	CPM1A-30CDT-D-V1	CPM1A-30CDT1-D-V1
40	24 DC points	16 points	AC	CPM1A-40CDR-A-V1	CPM1A-40CDT-A-V1	CPM1A-40CDT1-A-V1
			DC	CPM1A-40CDR-D-V1	CPM1A-40CDT-D-V1	CPM1A-40CDT1-D-V1

■ EXPANSION I/O MODULES

Stock Note: Shaded models are normally stocked.

Description	Max. number of modules	Inputs	Outputs	Part number
20 I/O points 12 inputs, 8 outputs 	3 max. (See Note.)	24 VDC	Relays	CPM1A-20EDR1
		24 VDC	Sinking transistors	CPM1A-20EDT
		24 VDC	Sourcing transistors	CPM1A-20EDT1
8 inputs 		24 VDC	—	CPM1A-8ED
8 outputs 		—	Relays	CPM1A-8ER
		—	Sinking transistors	CPM1A-8ET
		—	Sourcing transistors	CPM1A-8ET1

Note: A maximum of 3 expansion modules can be used with the following CPUs: 30-point and 40-point with DC inputs.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

■ DEDICATED I/O MODULES

Stock Note: Shaded models are normally stocked.

Description	Max. number of modules	Inputs	Outputs	Part number
Analog I/O Module 2 analog inputs (2 words) 1 analog output (1 word) 	3 max.	2 analog inputs	1 analog output	CPM1A-MAD01
				CPM1A-MAD11
Temperature Sensor Input Modules 	3 max.	2 inputs (Types J and K)	—	CPM1A-TS001
		4 inputs (Types J and K)		CPM1A-TS002
	3 max.	2 inputs (Pt100, JPt100)	1 analog output	CPM1A-TS101
	3 max.	2 inputs (Pt100, JPt100)		CPM1A-TS101-DA
	1 max. (See Note.)	4 inputs (Pt100, JPt100)		CPM1A-TS102
	3 max.	8 bits (Inputs from the Master.)	8 bits (Outputs to the Master.)	CPM1A-SRT21
CompoBus/S I/O Link Module 8 inputs and 8 outputs 	3 max.	Flat cable, 4-core, 0.75 mm ² , 100 m length		SCA1-4F10
		Twisted pair cable, 2-core, 0.75 mm ² , available commercially		Belden #9409 cable
		—		—
DeviceNet I/O Link Module 32 inputs and 32 outputs 	3 max.	32 bits (Inputs from the Master.)	32 bits (Outputs to the Master.)	CPM1A-DRT21
		Omron connector with screws (included with DeviceNet I/O Link Module).		XW4B-05C1-H1-D
		Omron Connector for multidrop connections using thick cables.		XW4B-05C4-TF-D
Profibus-DP Slave Module 16 inputs and 16 outputs 	3 max.	16 bits (Inputs from the Master.)	16 bits (Outputs to the Master.)	CPM1A-PRT21
		Shielded twisted pair cable, available commercially		Belden #3079A cable

Note: Only one CPM1A-TS002/TS102 Temperature Sensor Input Module can be connected to the CPU. If a CPM1A-TS002/102 is connected to the CPU, only one additional Special I/O Module (other than a CPM1A-TS002/102) or one Expansion I/O Module can be connected to the CPU.

■ PERIPHERAL DEVICES

Stock Note: Shaded models are normally stocked.

Product	Description	Part number
Programming console	Hand-held programming console with cable attached, 2 m length	CQM1-PRO01-E
	Hand-held programming console with back light (cable not included)	C200H-PRO27-E
Connecting cable	Connects C200H programming console to peripheral port, 2 m length	C200H-CN222
	Connects C200H programming console to peripheral port, 4 m length	C200H-CN422

■ SUPPORT SOFTWARE

Stock Note: Shaded models are normally stocked.

Product	Functions	Part number
CX-Programmer Jr.	Windows-based programming software; reduced instruction set and networking commands.	WS02-CXPC1-EJ-V__
CX-Programmer	Full programming software package programs micro, small and larger controllers.	WS02-CXPC1-E-V__

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

■ COMMUNICATIONS ADAPTERS AND CABLES

Stock Note: Shaded models are normally stocked.

Description	Function	Part number
RS-232C adapter 	Converts data communications from peripheral port for RS-232C devices.	CPM1-CIF01
RS-232C cable 	RS-232C to RS-232C; PC connection for program download; cable length 2 m	C200H-CN229-EU
		CBL-202*
	RS-232C to RS-232C for PLC communication; cable length 2 m	C200H-CN320-EU
		CBL-804*
	Communication cable for NT31C (port B only)	NT31C-CN510-EU
		NT31C-CN320-EU
		NT31C-CN520-EU
RS-422/RS-485 adapter 	Converts data communications from peripheral port for RS-422/RS-485 devices.	CPM1-CIF11

* Available in Canada only.

■ PROGRAM TRANSFER EQUIPMENT

Stock Note: Shaded models are normally stocked.

Product	Description	Part number
Expansion Memory Unit	Uploads and downloads program and setup memory areas to and from the controller.	CPM1-EMU01-V1
EEPROM (256 kbits)	Used with the Expansion Memory Unit	EEPROM-CPM1-EMU01

■ MANUALS

Product	Description	Part number
Operation manual	CPM1A programmable controllers operation manual	W317
Programming manual	CPM1/CPM1A/CPM2A/CPM2C/SRM1(-V2) programming manual	W353

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Specifications

■ GENERAL SPECIFICATIONS

Input type		DC input			
CPU type		10-point I/O	20-point I/O	30-point I/O	40-point I/O
Power supply voltage/fre- quency	AC power supply	100 to 240 VAC, 50/60 Hz			
	DC power supply	24 VDC			
Operating voltage range	AC power supply	85 to 264 VAC			
	DC power supply	20.4 to 26.4 VDC			
Power consumption	AC power supply	30 VA max.		60 VA max.	
	DC power supply	6 W max.		20 W max.	
Inrush current		30 A max.		60 A max.	
External power supply (AC only)	Power supply voltage	24 VDC			
	Power supply output capacity	200 mA		300 mA	
Insulation resistance		20 M Ω min. at 500 VDC between the AC terminals and the protective earth terminal.			
Dielectric strength		2,300 VAC at 50/60 Hz for one minute with a leakage current of 10 mA max. between all the external AC terminals and the protective earth terminal.			
Noise resistance		Conforms to IEC61000-4-4, 2 kV (power lines) 1500 Vp-p, pulse width 0.1 to 1 μ s, rise time: 1 ns (via noise simulation)			
Vibration resistance		10 to 57 Hz with an amplitude of 0.075 mm, and 57 to 150 Hz with an acceleration of 1.5 G in the X, Y, and Z directions for 10 sweeps of minutes each.			
Shock resistance		147 m/s ² in the X, Y and Z directions 3 times each.			
Ambient temperature	Operating	0°C to 55°C (32°F to 131°F)			
	Storage	-20°C to 75°C (-4°F to 167°F)			
Ambient humidity	Operating	10% to 90% RH no condensation			
Ambient environment	Operating	With no corrosive gas			
Terminal screw size		M3			
Power supply holding time		10 ms min. for AC models, and 2 ms min. for DC models			
CPU Weight	AC models	400 g max.	500 g max..	600 g max..	700 g max.
	DC models	300 g max.	400 g max.	500 g max.	600 g max.
Expansion Weight		Units with 20 I/O points:		300 g max.	
		Units with 8 output points:		250 g max.	
		Units with 8 input points:		200 g max.	
		MAD01 Analog I/O unit:		150 g max.	
		MAD11 Analog I/O unit:		250 g max.	
		Temperature sensor units:		250 g max.	
		CompoBus/S I/O link unit:		200 g max.	
		DeviceNet I/O link unit:		200 g max.	
		Profibus-DP slave unit:		125 g	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

■ CHARACTERISTICS

Input type		D/C input			
CPU type		10-point I/O	20-point I/O	30-point I/O	40-point I/O
Control method		Stored program method			
I/O control method		Combination of the cyclic scan and immediate refresh processing methods.			
Programming language		Ladder diagram			
Instruction word		1 step per instruction, 1 to 5 words per instruction			
Types of instructions	Basic instructions	14 types			
	Special instructions	79 types, 139 instructions			
Instruction execution time	Basic instructions	0.72 to 16.2 μ s			
	Special instructions	MOV instruction = 16.3 μ s			
Program capacity		2,048 words			
User data memory		1,024 words			
Maximum I/O points	CPU only	10 points (6 input/ 4 output points)	20 points (12 input/ 8 output points)	30 points (18 input/ 12 output points)	40 points (24 input/ 16 output points)
	With Expansion I/O Module	—	—	Up to 90 points (54 input/ 36 output points)	Up to 100 points (60 input/ 40 output points)
Memory protection		Maintains the contents of the HR, AR, Counter and Data Memory Areas.			
Memory backup		Flash memory: User program, data memory (Read only) and PLC setup area are backed up without a battery. Super capacitor: Data memory (Read/Write), holding bits, auxiliary memory bits, counter values, error log area are backed up by a capacitor for 20-days at an ambient temperature of 25°C.			
Self-diagnostic function		CPU error (watchdog timer), memory errors, I/O bus errors			
Program check		No END instruction programming errors (constantly checked during operation)			
Pulse output		1 point: 2 kHz			
High-speed counter		1 point: Single phase at 5 kHz or two-phase at 2.5 kHz (linear counting method) Incremental mode: 0 to 65535 (16-bit) Decremental mode: -32767 to 32767 (16-bit)			
Quick-response inputs		Together with the external interrupt input (minimum pulse width of 0.2 ms)			
Input time constant		Can be set at 1 ms, 2 ms, 4 ms, 8 ms, 16 ms, 32 ms, 64 ms, or 128 ms.			
Interrupt processing: External interrupt		2 points (Response time of 0.3 ms max.)	4 points (Response time of 0.3 ms max.)		
Analog settings		2 points: (0 to 200 BOD)			

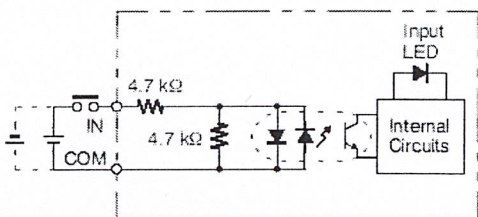
■ I/O ALLOCATION

Input bits		00000 to 00915; words not used for input or output bits can be used for work bits.
Output bits		01000 to 01915; words not used for input or output bits can be used for work bits.
Work bits (IR Area)		512: IR 20000 to IR 23115 (IR 200 to IR 231)
Special bits (SR Area)		384: SR 23200 to SR 25515 (SR 232 to SR 255)
Temporary bits (TR Area)		8: TR 0 to TR 7
Holding bits (HR Area)		320: HR 0000 to HR 1915 (HR 00 to HR 19)
Auxiliary bits (AR Area)		256: AR 0000 to AR 1515 (AR 00 to AR 15)
Link bits (LR Area)		256: LR 0000 to LR 1515 (LR 00 to LR 15)
Timers/Counters		128: TIM/CNT 000 to 127 100-ms timer: TIM 000 to TIM 127 10-ms timer: TIM 000 to TIM 127 Decremental counter, reversible counter
Data memory	Read/Write	1,024 words (DM 0000 to DM 1023)
	Read only	512 words (DM 6144 to DM 6655)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

I/O SPECIFICATIONS

CPU DC Input

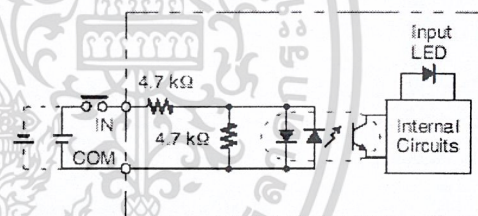
Item	Specifications	Circuit
Input voltage	24 VDC $+10\%$ / -15%	 Note: The polarity of the input power supply can be either positive or negative.
Input impedance	IN0000 to IN0002: 2 kΩ Others: 4.7 kΩ	
Input current (typical)	IN0000 to IN0002: 12 mA Others: 5 mA	
ON voltage	14.4 VDC min.	
OFF voltage	5.0 VDC max.	
ON delay (See Note 1)	1 to 128 ms max. (default: 8 ms) (See Note 1)	
OFF delay (See Note 1)	1 to 128 ms max. (default: 8 ms) (See Note 1)	

Note: 1. The actual ON/OFF delay includes an input constant of 1, 2, 4, 8, 16, 32, 64, or 128 ms (default: 8 ms).

2. When IN0000 to IN0006 are used for the high-speed counter inputs, the delays are as shown below:

Input	Increment mode	Differential phase mode
IN0000 (A-phase)	5 kHz	2.5 kHz
IN0001 (B-phase)	Normal input	
IN0002 (Z-phase)	ON: 100 μs max. OFF: 500 μs max.	
IN0003 to IN0006	0.3 ms max. (From the time of input ON until the interrupt subroutine is executed.)	

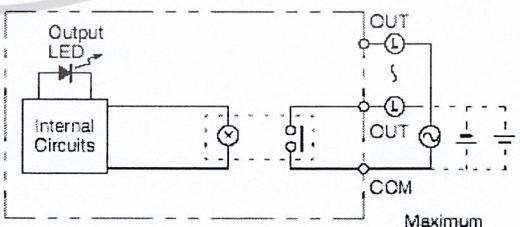
Expansion I/O Unit

Item	Specifications	Circuit
Input voltage	24 VDC, $+10\%$ / -15%	 Note: The polarity of the input power supply can be either positive or negative.
Input impedance	4.7 kΩ	
Input current (typical)	5 mA	
ON voltage	14.4 VDC min.	
OFF voltage	5.0 VDC max.	
ON delay	1 to 128 ms max. (default: 8 ms) (See Note)	
OFF delay	1 to 128 ms max. (default: 8 ms) (See Note)	

Note: The actual ON/OFF delay includes an input constant of 1, 2, 4, 8, 16, 32, 64, or 128 ms (default: 8 ms).

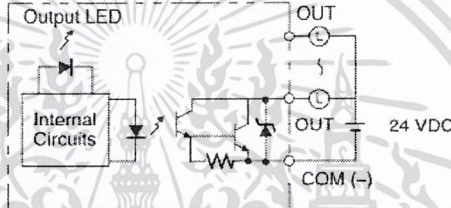
OUTPUT SPECIFICATIONS (CPU AND EXPANSION I/O MODULES)

Relay Output

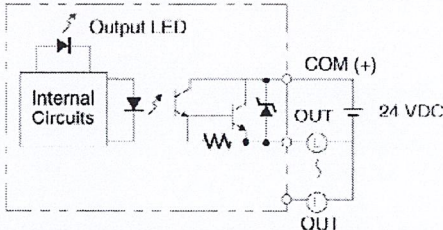
Item			Specifications	Circuit
Maximum switching capacity			2 A, 250 VAC (cos φ = 1) 2 A, 24 VDC (4 A/common)	 Maximum 250 VAC: 2 A 24 VDC: 2 A
Minimum switching capacity			10 mA, 5 VDC	
Relay service life	Electrical	Resistance load	150,000 times	
		Inductive load	100,000 times	
	Mechanical		20 million times	
ON delay			15 ms max.	
OFF delay			15 ms max.	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Transistor Outputs (Sinking)

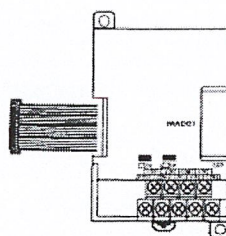
Item	Specifications					
	CPM1A-10CDT-□(-V1)	CPM1A-20CDT-□(-V1)	CPM1A-30CDT-□(-V1)	CPM1A-40CDT-□(-V1)	CPM1A-20EDT	CPM1A-8ET
Max. switching capacity	24 VDC $+10\%/-15\%$, 0.3 A/point (See note)					4.5 to 30 VDC 0.2 A (See note 2.) 0.3 A (See note 3.)
	0.9 A/Unit	0.9 A/common 1.8 A/Unit	0.9 A/common 2.7 A/Unit	0.9 A/common 3.6 A/Unit	0.9 A/common 1.8 A/Unit	
Leakage current	0.1 mA max.					
Residual voltage	1.5 V max.					
ON delay	0.1 ms max.					
OFF delay	OUT01000/01001: 0.2 ms max. (load current: 100 to 300 mA) 0.5 ms max. (load current: 5 to 100 mA)					1 ms max. (24 VDC $+10\%/-15\%$, 5 to 300 mA)
	Other than OUT01000/01001: 1 ms max. (load current 5 to 300 mA)					
Fuse	V1 CPUs: No fuse Expansion I/O Units and Pre-V1 CPUs: 1.25 A/common (cannot be replaced by the user)					
Circuit configuration						

Transistor Outputs (Sourcing)

Item	Specifications					
	CPM1A-10CDT1-□(-V1)	CPM1A-20CDT1-□(-V1)	CPM1A-30CDT1-□(-V1)	CPM1A-40CDT1-□(-V1)	CPM1A-20EDT1	CPM1A-8ET1
Max. switching capacity	24 VDC $+10\%/-15\%$, 0.3 A/point (See note)					4.5 to 30 VDC 0.2 A (See note 2.) 0.3 A (See note 3.)
	0.9 A/Unit	0.9 A/common 1.8 A/Unit	0.9 A/common 2.7 A/Unit	0.9 A/common 3.6 A/Unit	0.9 A/common 1.8 A/Unit	
Leakage current	0.1 mA max.					
Residual voltage	1.5 V max.					
ON delay	0.1 ms max.					
OFF delay	OUT01000/01001: 0.2 ms max. (load current: 100 to 300 mA) 0.5 ms max. (load current: 5 to 100 mA)					1 ms max. (24 VDC $+10\%/-15\%$, 5 to 300 mA)
	Other than OUT01000/01001: 1 ms max. (load current 5 to 300 mA)					
Fuse	V1 CPUs: No fuse Expansion I/O Units and Pre-V1 CPUs: 1.25 A/common (cannot be replaced by the user)					
Circuit configuration						

Note: When using the OUT01000 or OUT01001 as a pulse output, connect dummy resistors as required to set the load current to 0.1 to 0.2 A. If the load current is below 0.1 A, the ON-to-OFF response time will become longer and high-speed pulse will not be output. On the other hand, if the load current is above 0.2 A, the transistor may generate heat and components may be damaged.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



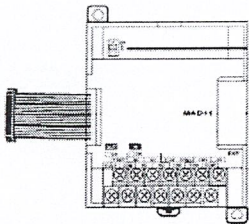
■ ANALOG I/O MODULE

CPM1A-MAD01 Specifications

Item		Specification	
Model		CPM1A-MAD01	
I/O type		Voltage	Current
Analog inputs	Number of inputs	2	
	Input signal range	0 to 10 V or 1 to 5 V	4 to 20 mA
	Maximum rated input	±15 V	±30 mA
	External input impedance	1 MΩ min.	250 Ω rated
	Resolution	1/256	
	Overall precision	1.0% of full scale	
	Converted A/D data	8-bit binary	
Analog output (See Note 1.)	Number of outputs	1	
	Output signal range	0 to 10 V or -10 to 10 V	4 to 20 mA
	External output max. current	5 mA	—
	External output allowed load resistance	—	350 Ω
	Resolution	1/256 (1/512 when the output signal range is -10 to 10 V.)	
	Overall precision	1.0% of full scale	
	Data setting	8-bit binary with sign bit	
Conversion time (See Note 2.)		10 ms/Unit max.	
Isolation method		Photocoupler isolation between I/O terminals and PC (There is no isolation between the analog I/O signals.)	

- Note: 1. The voltage output and current output can be used at the same time, but the total output current cannot exceed 21 mA.
 2. The conversion time is the total time for 2 analog inputs and 1 analog output.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



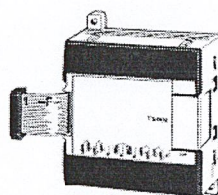
CPM1A-MAD11 Specifications

Item		Specification	
Model		CPM1A-MAD11	
I/O type		Voltage	Current
Analog inputs	Number of inputs	2 inputs (2 words allocated)	
	Input signal range	0 to 5 VDC, 1 to 5 VDC 0 to 10 VDC, -10 to 10 VDC	0 to 20 mA or 4 to 20 mA
	Maximum rated input	±15 V	±30 mA
	External input impedance	1 MΩ min.	250 Ω
	Resolution	1/6000 (full scale)	
	Overall accuracy	25°C	0.3% full scale
		0 to 55°C	0.6% full scale
	Converted A/D data	16-bit binary (4-digit hexadecimal) Full scale for -10 to 10 V: F448 to 0BB8 Hex Full scale for other ranges: 0000 to 1770 Hex	
	Averaging function	Supported (Settable for individual inputs via DIP switch)	
	Open-circuit detection function	Supported	
Analog output	Number of outputs	1 output (1 word allocated)	
	Output signal range	1 to 5 VDC, 0 to 10 VDC, or -10 to 10 VDC	0 to 20 mA or 4 to 20 mA
	Allowable external output load resistance	1 kΩ min.	600 Ω max.
	External output impedance	0.5 Ω max.	—
	Resolution	1/6000 (full scale)	
	Overall accuracy	25°C	0.4% full scale
		0 to 55°C	0.8% full scale
	Set data (D/A conversion)	16-bit binary (4-digit hexadecimal) Full scales for -10 to 10 V: F448 to 0BB8 Hex Full scale for other ranges: 0000 to 1770 Hex	
Conversion time		2 ms/point (6 ms/all points)	
Isolation method		Photocoupler isolation between analog I/O terminals and internal circuits. No isolation between analog I/O signals.	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

■ TEMPERATURE SENSOR MODULES

By connecting a Temperature Sensor Module (CPM1A-TS001/TS002/TS101/TS101A/TS102) to the CPM1A, inputs can be received from thermocouples or temperature-resistance thermometers. Inputs converted to binary data (4-digit hexadecimal) and stored in the IR area.



Specifications

Item	Specification		
Model	CPM1A-TS001/TS002	CPM1A-TS101/TS102	CPM1A-TS101-DA
Number of inputs	TS001: 2; TS002: 4	TS101: 2; TS102: 4	2
Input types (See Note 1)	Thermocouple types K or J, selectable	Platinum resistance thermometer types Pt100 and JPt100, selectable	
Input resolution	0.1°C in 2's complement format	0.1°C in 2's complement format	
Input accuracy	±0.5% or ±2% of the stored value whichever is larger ±1 digit max. (See Note 2)	±0.5% or ±1% of the stored value whichever is larger ±1 digit max. (See Note 2)	1.0% max. full scale
Number of outputs	None	None	1
Output types	—	—	Voltage or current output
Output resolution	—	—	1/256 (0 to 10 V) 1/512 (–10 to +10 V) 1/256 (4 to 20 mA)
Output accuracy	—	—	1.0% max. full scale
Conversion cycle	250 ms for all points		60 ms max. for all points
Converted temperature data	Binary data (4-digit hexadecimal)		Binary data (8-digit hexadecimal)
Isolation method	Photocoupler isolation between I/O terminals and the PLC		

Note: 1. The same input type must be used for all inputs.

2. Accuracy for K thermocouples at temperatures less than –100°C: ±4°C ± 1 digit max.

Input Temperature Ranges for CPM1A-TS001/TS002

The rotary switch can be used to make of the following range and input type settings for CPM1A-TS001/002 models.

Thermocouple input	Range (°C)	Range (°F)
Type K	–200 to 1300	–300 to 2300
	0.0 to 500.0	0.0 to 900.0
Type J	–100 to 850	–100 to 1500
	0.0 to 400.0	0.0 to 750.0

Input Temperature Ranges for CPM1A-TS101/TS101DA/TS102

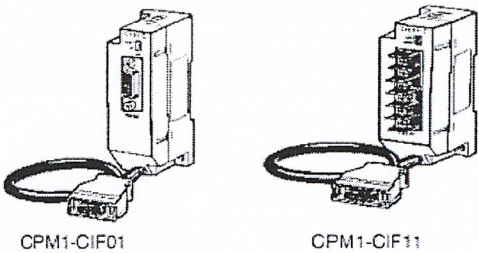
The rotary switch can be used to make of the following range and input type settings for CPM1A-TS101/102 models.

Platinum RTD input	Range (°C)	Range (°F)
Pt100	–200.0 to 650.0	–300 to 1200.0
JPt100	–200.0 to 650.0	–300 to 1200.0

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

■ COMMUNICATIONS ADAPTER

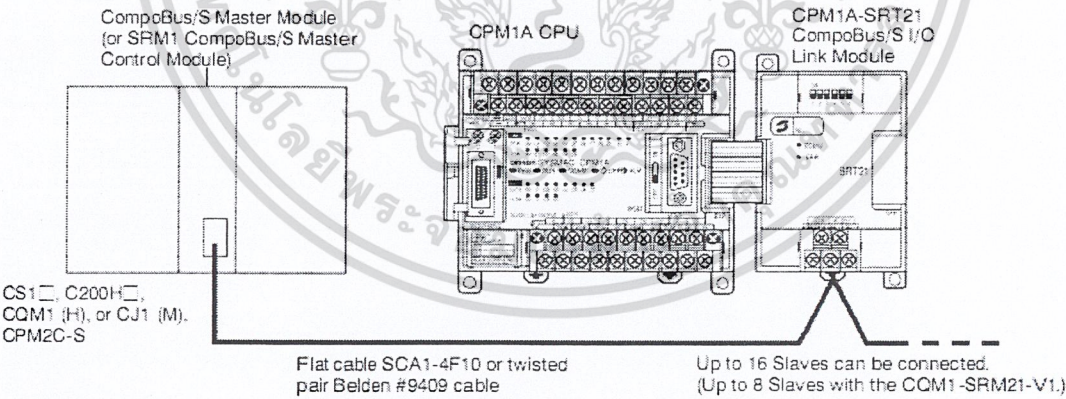
RS-232C Adapter and RS-422 Adapter



Model		CPM1-CIF01	CPM1-CIF11
Functions		Level conversion between the CMOS level (CPU side) and the RS-232C (peripheral device side)	Level conversion between the CMOS level (CPU side) and the RS-422 (peripheral device side)
Insulation		The RS-232C (peripheral device side) is insulated by a DC/DC converter and photocoupler.	The RS-422 (peripheral device side) is insulated by a DC/DC converter and photocoupler.
Power supply		Power is supplied by the CPU.	
Power consumption		0.3 A max.	
Transmission speed		38.4 kbits/s max.	
Vibration resistance		10 to 57 Hz with an amplitude of 0.075 mm, and 57 to 150 Hz with an acceleration of 1 G in the X, Y and Z directions for 80 minutes each (i.e. for 8 minutes each, 10 times).	
Shock resistance		1.5 G in the X, Y and Z directions 3 times each.	
Ambient temperature	Operating	0°C to 55°C (32°F to 131°F)	
	Storage	-20°C to 75°C (-4°F to 167°F)	
Ambient humidity	Operating	10% to 90% RH (with no condensation)	
Ambient environment	Operating	With no corrosive gas	
Weight		200 g max.	

■ COMPOBUS/S I/O LINK MODULE

The CPM1A controller can function as a Slave to a CompoBus/S Master Module (or SRM1 CompoBus/S Master Control Module) when a CPM1A-SRT21 CompoBus/S I/O Link Module is connected. The CompoBus/S I/O Link Module establishes an I/O link of 8 inputs and 8 outputs between the Master Module and the CPM1A. Up to 3 Expansion I/O Modules or Expansion Modules can be connected to a CPM1A CPU.



Specifications

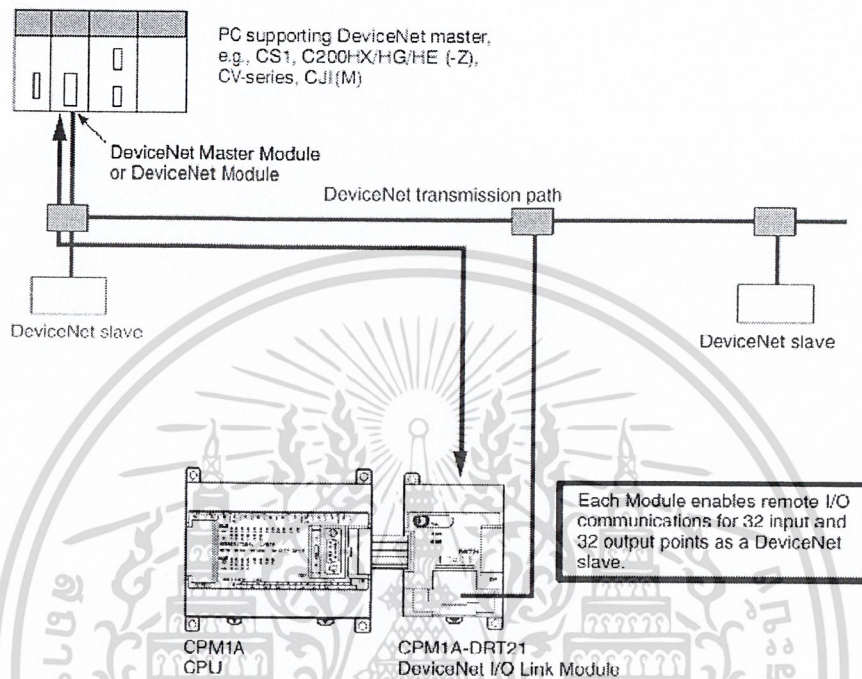
Model	CPM1A-SRT21
Master/Slave	CompoBus/S Slave
Number of I/O bits	8 input bits, 8 output bits
Number of words occupied in CPM2A I/O memory	1 input word, 1 output word (Allocated in the same way as other Expansion I/O Modules or Expansion Modules)
Node number setting	Set using the DIP switch.

Note: See the CompoBus/S section of Omron's *Remote I/O and Wiring Solutions Catalog (GC R101)* for more details on CompoBus/S communications.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

■ DEVICENET I/O LINK MODULE

The CPM1A controller can function as slaves to a DeviceNet Master when a DeviceNet I/O Link Module is connected. The DeviceNet I/O Link Module establishes an I/O link of 32 inputs and 32 outputs between the master and the controller. A maximum of 3 DeviceNet I/O Link Modules can be connected to a CPM1A to create I/O Links for up to 192 points (96 inputs and 96 outputs).



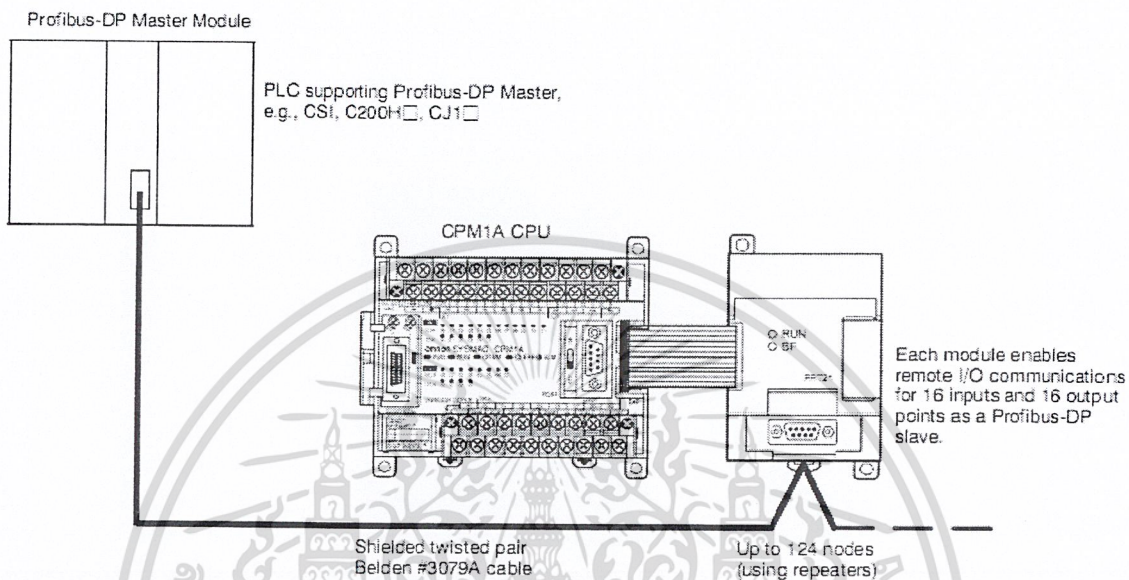
Specifications

Item	Specification
Model number	CPM1A-DRT21
Master/slave	DeviceNet Slave
Number of I/O points	32 input points, 32 output points
Number of words allocated in CPU Unit I/O memory	2 input words, 2 output words (Allocated in the same way as Expansion I/O Units and other Expansion Units)
Node number setting	Set using the rotary switches (Set before turning ON the CPU's power supply.)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

■ PROFIBUS-DP SLAVE MODULE

The CPM1A-V1 controller can function as a slave to a Profibus-DP Master Module when a CPM1A-PRT Profibus-DP Slave Module is connected. The Profibus-DP Slave Module establishes an I/O link of 16 inputs and 16 outputs between the master and the controller. A maximum of 3 Profibus-DP Slave Modules can be connected to a CPM1A or CPM2A to create I/O links for up to 96 points (48 inputs and 48 outputs).



Specifications

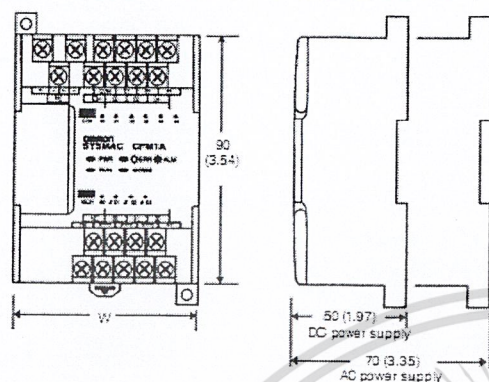
Storage temperature	-20 to +75°C
Ambient temperature	0 to +55°C
Ambient humidity	10 to 90% (non-condensing)
EMC compliance	EN 50081-2, EN 61131-2
Current consumption	100 mA from the PLC I/O bus
Weight	125 g (typical)
Control data	From CPU to unit: none
Status data	From unit to CPU: none
I/O data (in bytes)	2 bytes input, 2 bytes output

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Dimensions

Unit: mm (inch)

■ CPU, EXPANSION I/O AND SPECIAL I/O MODULES

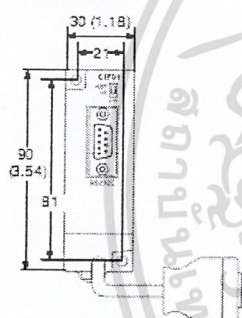


Model	W
CPM1A-10CD□-A/D-V1	66 (2.60)
CPM1A-20CD□-A/D-V1	86 (3.39)
CPM1A-30CD□-A/D-V1	130 (5.12)
CPM1A-40CD□-A/D-V1	150 (5.91)
CPM1A-20ED□	86 (3.39)
CPM1A-8E□	66 (2.60)
CPM1A-SRT21	66 (2.60)
CPM1A-MAD01	66 (2.60)
CPM1A-TS□□□	86 (3.39)
CPM1A-MAD11	86 (3.39)
CPM1A-DRT21	66 (2.60)
CPM1A-PRT21	66 (2.60)

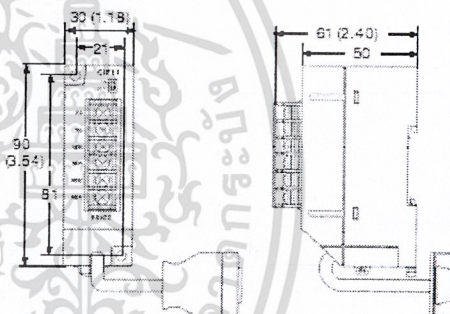
Unit: mm (inch)

■ COMMUNICATION ADAPTER MODULES

CPM1-CIF01

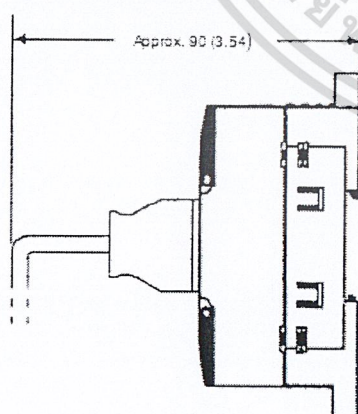


CPM1-CIF11

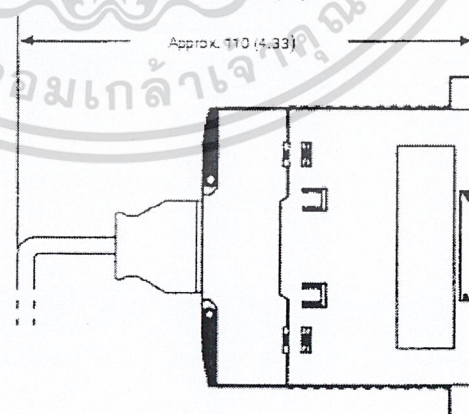


■ DIMENSIONS WITH PERIPHERAL DEVICES ATTACHED

CPU with DC Power Supply



CPU with AC Power Supply



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Functions

■ CONFIGURATION

The CPM1A CPUs feature a compact, one-piece construction that includes 10, 20, 30 or 40 built-in I/O terminals. Three output models are available: Relay outputs, sinking (NPN) transistor output and sourcing (PNP) transistor output.

Expansion

Up to three Expansion I/O Modules can be connected to a 30-point or 40-point CPU to add an extra 8 or 20 I/O points for each, for a maximum of up to 100 I/O points.

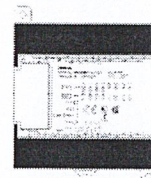
Dedicated I/O Modules

Up to 3 Analog I/O Modules or Temperature Sensor Input Modules can be used with 30-point and 40-point CPUs. Each analog I/O module provides 2 analog inputs and 1 analog output, so a maximum of 6 analog inputs and 3 analog outputs can be achieved by connecting 3 Analog I/O Modules. Each Temperature Sensor Module provides two temperature sensor inputs from either thermocouples or platinum resistance thermometers. Up to 6 inputs can be connected.

CPM1A-10CD□-□-V1
(10 I/O terminals)



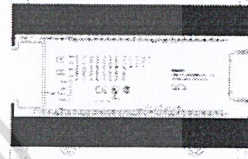
CPM1A-20CD□-□-V1
(20 I/O terminals)



CPM1A-30CD□-□-V1
(30 I/O terminals)



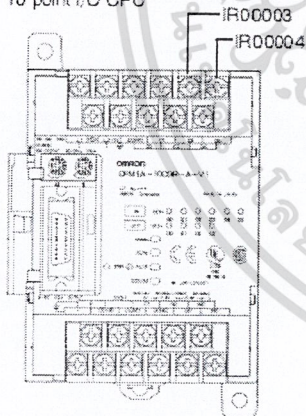
CPM1A-40CD□-□-V1
(40 I/O terminals)



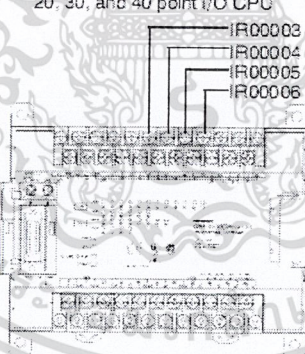
■ INTERRUPT INPUTS

There are two input interrupts in the CPM1A 10-point I/O CPU and four in the 20-, 30-, and 40-point I/O CPUs. Input interrupts are available in two modes. In addition to normal interrupt inputs, the CPM1A has a counter mode that counts high-speed input signals and triggers interrupts at fixed count multiples.

10 point I/O CPU



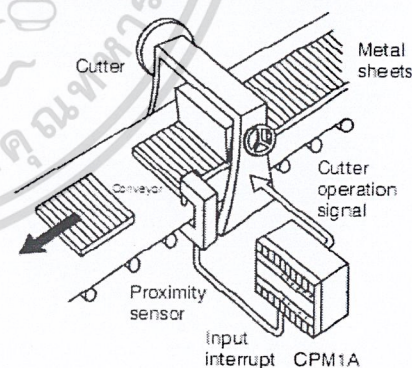
20, 30, and 40 point I/O CPU



Application Example:

Cutting Metal Sheets to Specified Lengths

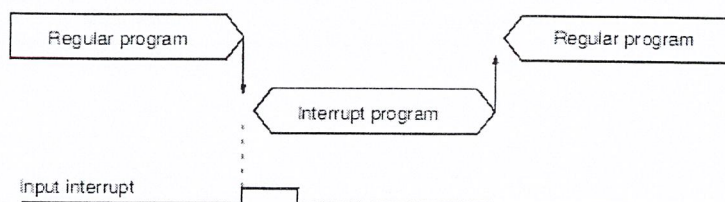
The proximity sensor detects the edge of a metal plate to operate the cutter. Metal sheets can be cut continuously to the specified lengths at a high speed.



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

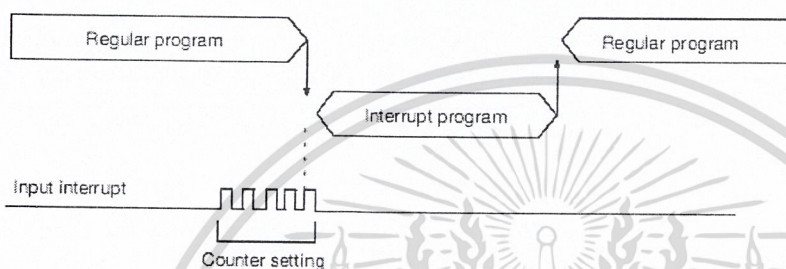
Input Interrupt Mode

If an input interrupt occurs, the regular program shuts down irrelevant of the cycle time, and the interrupt processing program is executed immediately.



Counter Mode

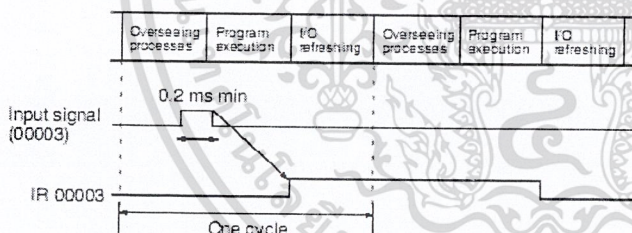
When the number of external signals counted at high speed reaches a specified number of counts, the regular program shuts down, and the interrupt processing program is executed at fixed counts. The count can be set between 0 and 65535.



■ QUICK-RESPONSE INPUTS

Quick-response inputs can detect input signals with a pulse width as short as 0.2 ms regardless of their timing during the scan cycle. Quick-response inputs and interrupt inputs use the same input terminals.

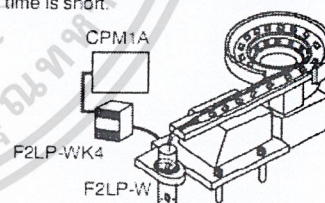
CPU	Input no.	Minimum input pulse width
10 point I/O CPU	00003 to 00004	0.2 ms
20 point, 30 point, 40 point I/O CPU	00003 to 00006	



Application Example:

Calculating the Number of Chips

The metal sensor counts the number of parts that have passed. Steady counting can be achieved even when the input-CN time is short.



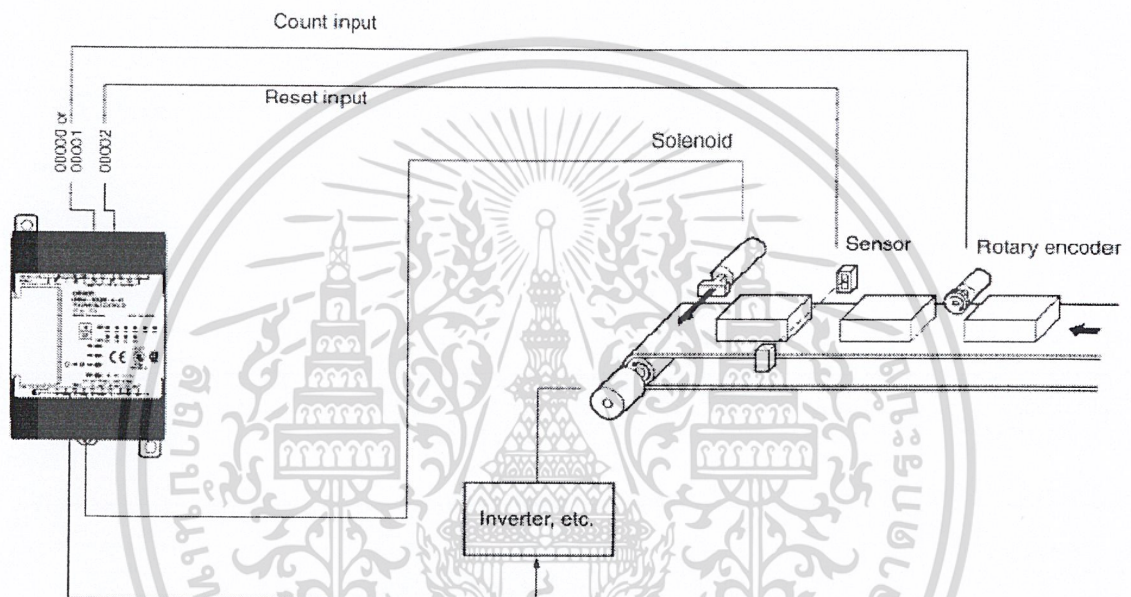
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

■ HIGH-SPEED COUNTER

The CPM1A has a high-speed counter function that can be used in the incrementing and up/down mode. Using this function together with the input interrupts enables zone comparison control or target value control irrelevant of the cycle time.

Counting mode		Incrementing mode	Up/Down mode
Input no.	00000	Count input	A-phase input
	00001	—	B-phase input
	00002	Reset input	Z-phase input
Input method		Single-phase input	Phase-difference, 4× inputs
Count frequency		5.0 kHz	2.5 kHz
Count range		0 to 65535	-32767 to 32767

Note: When using in the incrementing mode, the input 00001 can be used as an input contact.

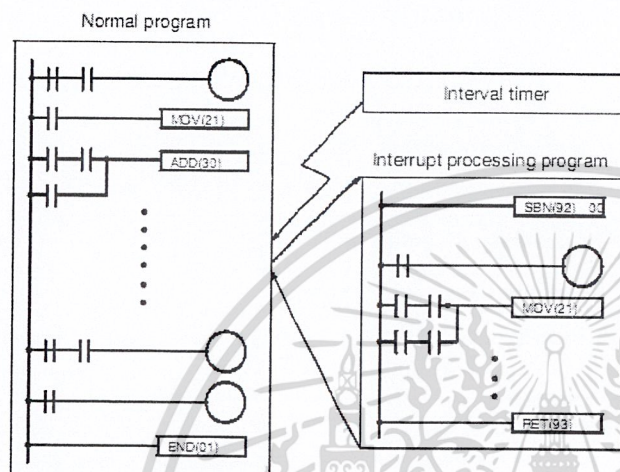


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

■ INTERVAL TIMER INTERRUPTS

The CPM1A has one interval timer. The interval timer shuts down the regular program regardless of the point in the cycle once the time is up, and immediately executes an interrupt processing program. Interval timers are used in the following two modes.

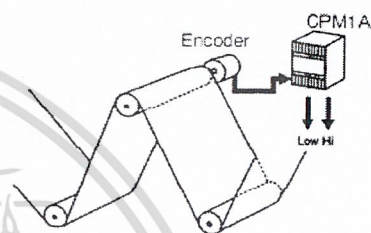
Item	One-shot mode	Scheduled interrupt mode
Operation	An interrupt is executed only once when the time is up.	Interrupts are executed repeatedly at fixed periods.
Setting time	0.5 ms to 319,968 ms (0.1-ms units)	



Application Example:

Computing the Sheet Speed

The number of pulse inputs is computed in the interrupt mode at a fixed time to calculate the speed.



■ ANALOG SETTING

The CPM1A contains two analog setting controls that can be used for a broad range of analog timer and counter settings. Turning the setting control stores values of 0 to 200 (BCD data) in the SR area.

Analog setting	Storage area	Setting value (BCD)
Analog setting 0	SR 250	0000 to 0200
Analog setting 1	SR 251	

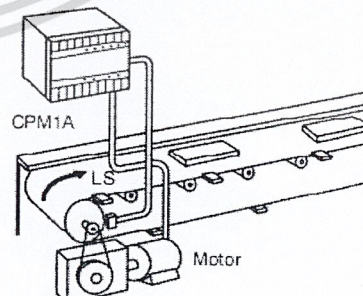
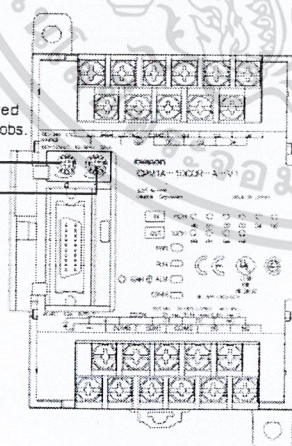
Application Example:

Tact Operation Control of Conveyor Lines

A conveyor can be stopped temporarily as required for assembly processes. When the timer function and limit switches are used in a combination, conveyors can be stopped for a fixed time or can be run at a constant speed for a fixed distance. Fine adjustment of the stopping time can be easily done by using the analog setting controls.

*Phillips screwdriver is required to turn analog adjustment knobs.

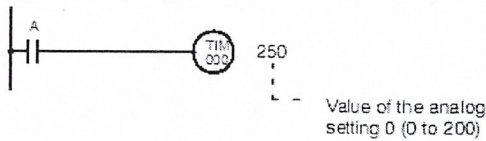
Analog setting 0
Analog setting 1



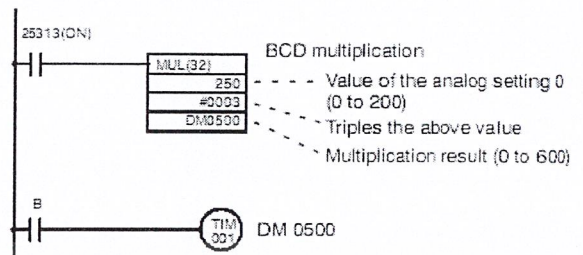
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Program Example

1. Analog timer for 0.0 to 20.0 seconds



2. Analog timer for 0.0 to 60.0 seconds



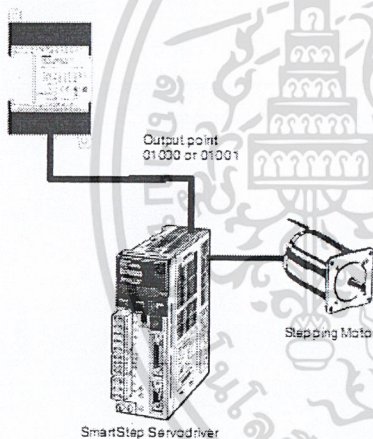
PULSE OUTPUT FUNCTION

The CPM1A transistor output models have an output function capable of outputting a pulse of 20 Hz to 2 kHz (single-phase output).

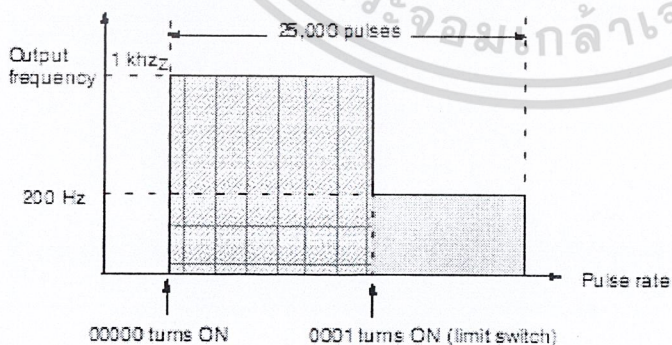
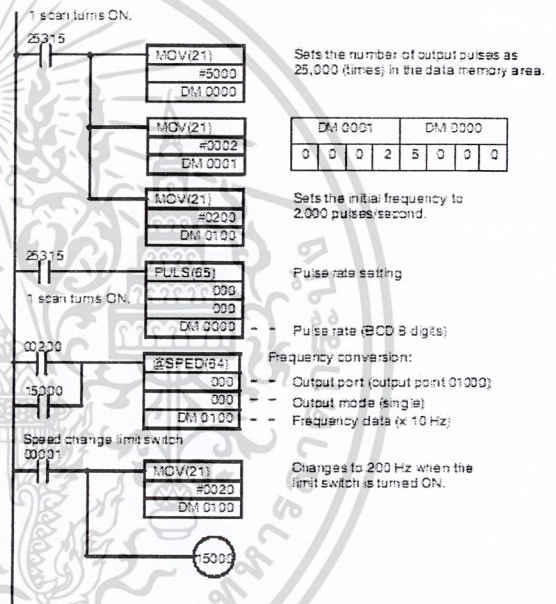
When used in combination with a Stepping Driver or SmartStep Servodriver, positioning can be easily performed.

Application Example

Changing the speed of the Stepping Motor.



Program Example

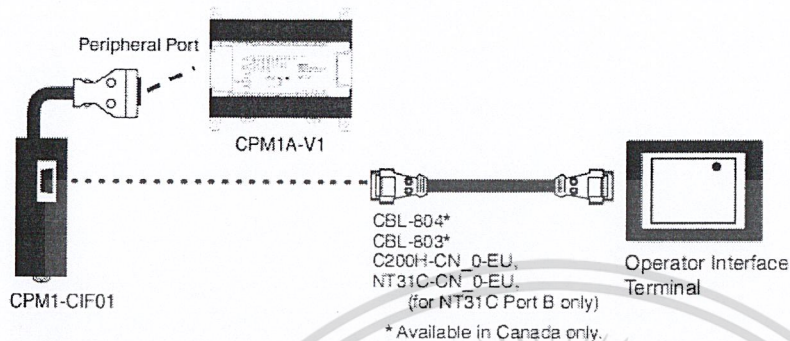


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Communications

■ NT LINK FOR PROGRAMMABLE TERMINALS

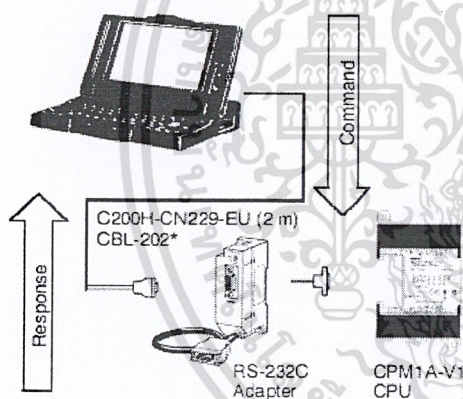
Use Omron's high-speed NT Link for real-time communications between the CPM1A and a Programmable Terminal.



■ HOST LINK COMMUNICATIONS

CPM1A Host Link communications consist of interactive procedures whereby the CPM1A returns a response to a command sent from the IBM PC/AT or compatible computer. These communications allow the IBM PC/AT or compatible computer to read and write in the CPM1A's I/O Areas and Data Memory Areas as well as in areas containing the status of various settings.

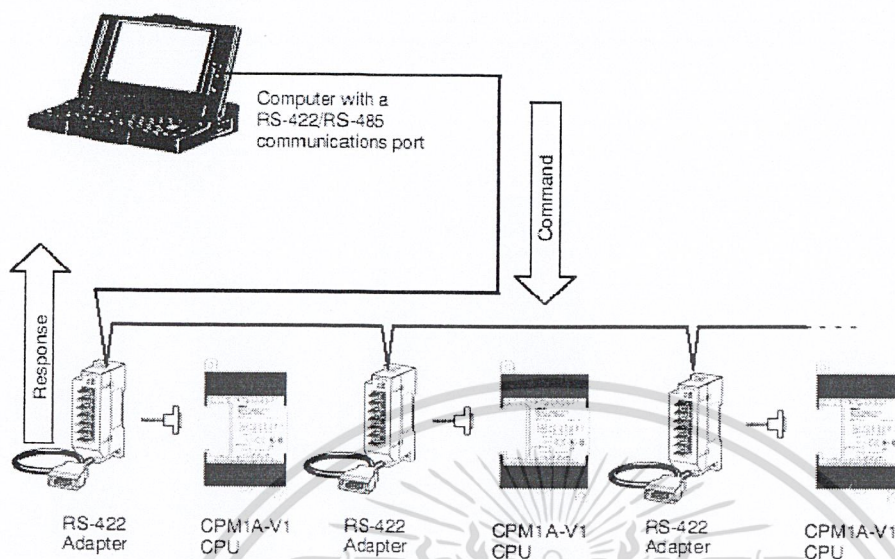
1:1 Host Link Communications



* Available in Canada only.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

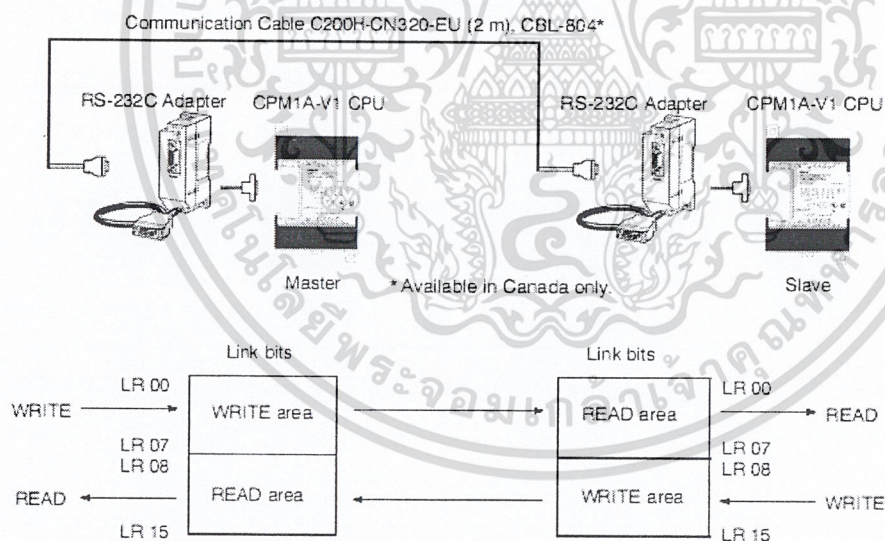
1:N Host Link Communications



1:1 CONNECTIONS FOR DATA EXCHANGE

With a 1:1 Link, two CPM1As or a CPM1A and CQM1 or C200H are connected 1:1 with one side as the Master and the other as the Slave to provide an I/O link of a maximum of 256 points (LR 0000 to LR 1515).

Example of a 1:1 Link between CPM1As



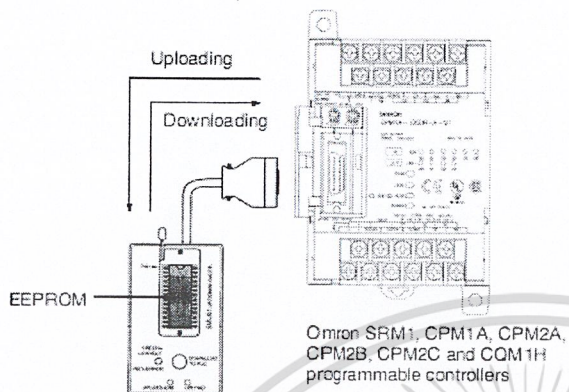
Limitations of the CPM1A 1:1 Link

CPM1A I/O links are limited to 16 words (LR 00 to LR 15). Therefore, use these 16 words (LR 00 to LR 15) on the CQM1 or C200H side when forming 1:1 links with a CQM1 or C200H.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

■ PROGRAM TRANSFER UNIT

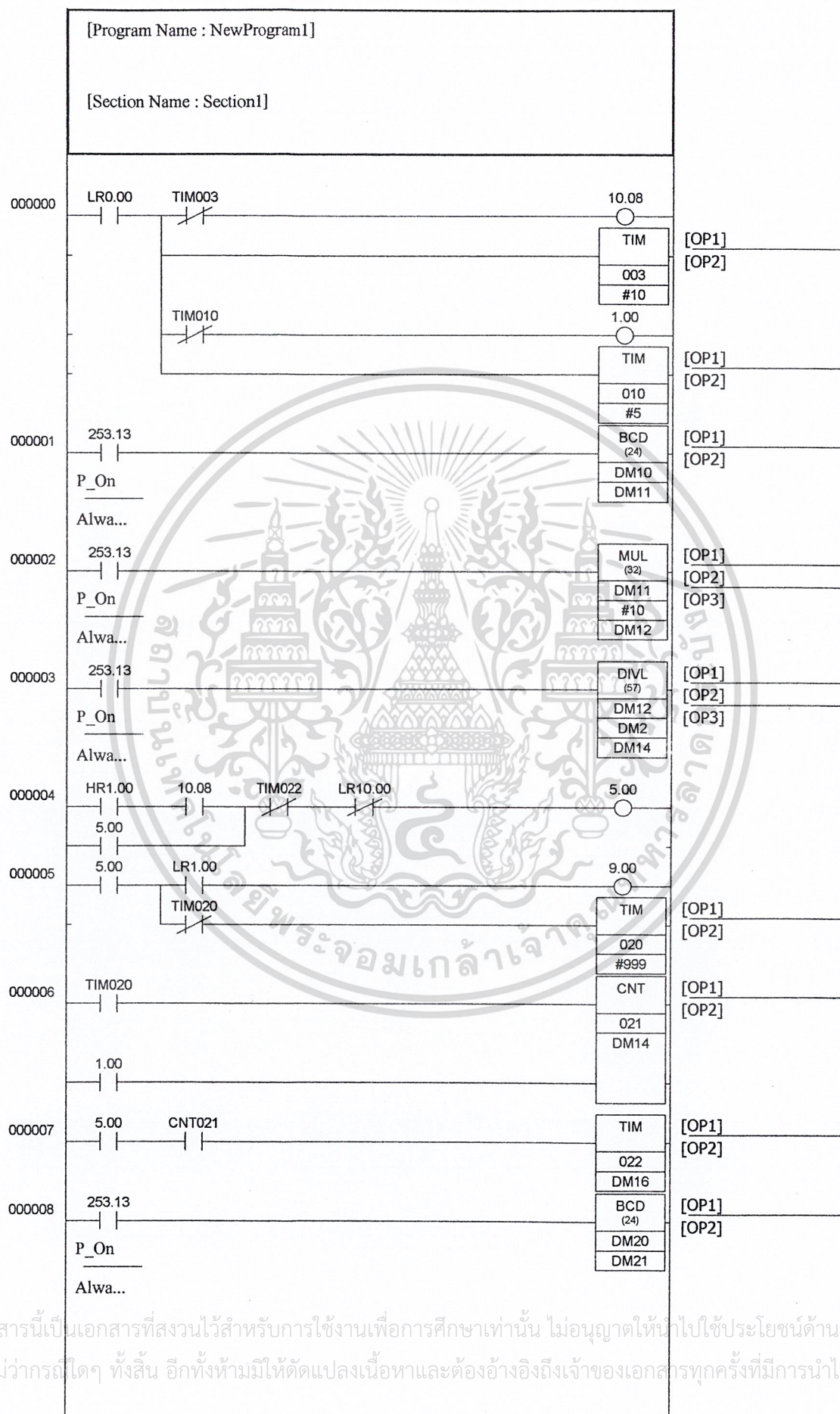
Use Omron's EEPROM program transfer unit to update programs in machines or program multiple controllers with the same program. The CPM1-EMU01-V1 Expansion Memory Unit connects to the peripheral port of micro and small PLCs.



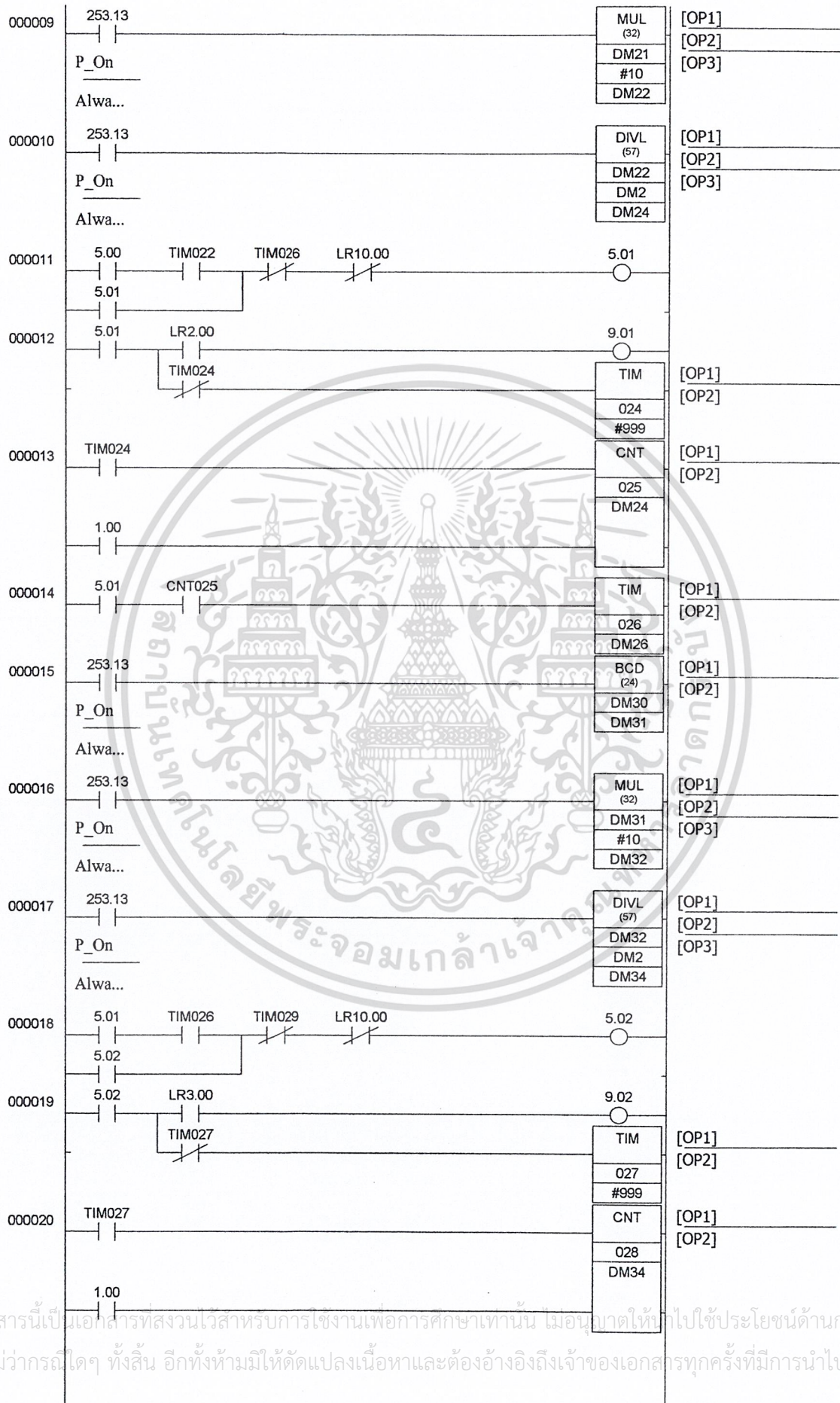
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



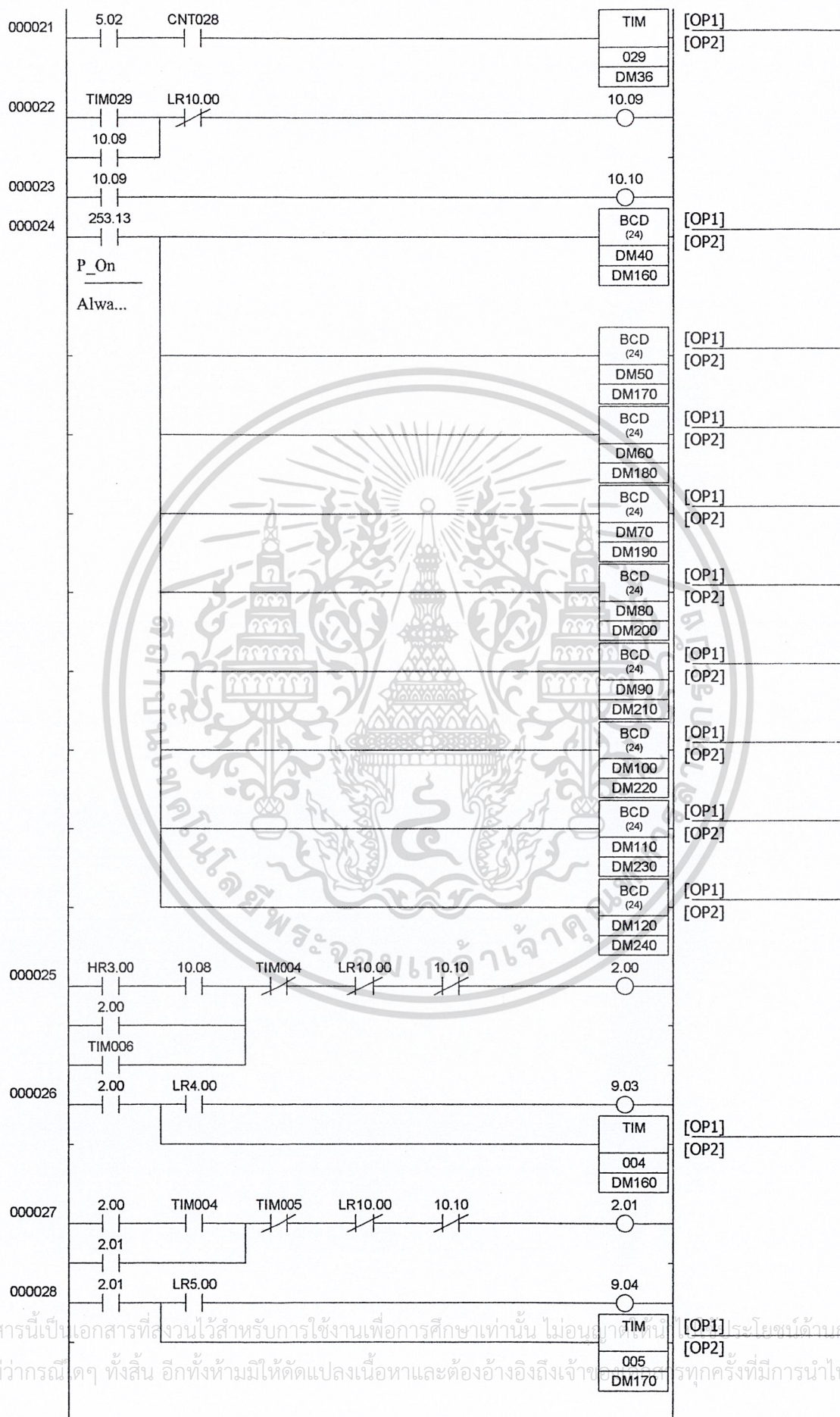
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

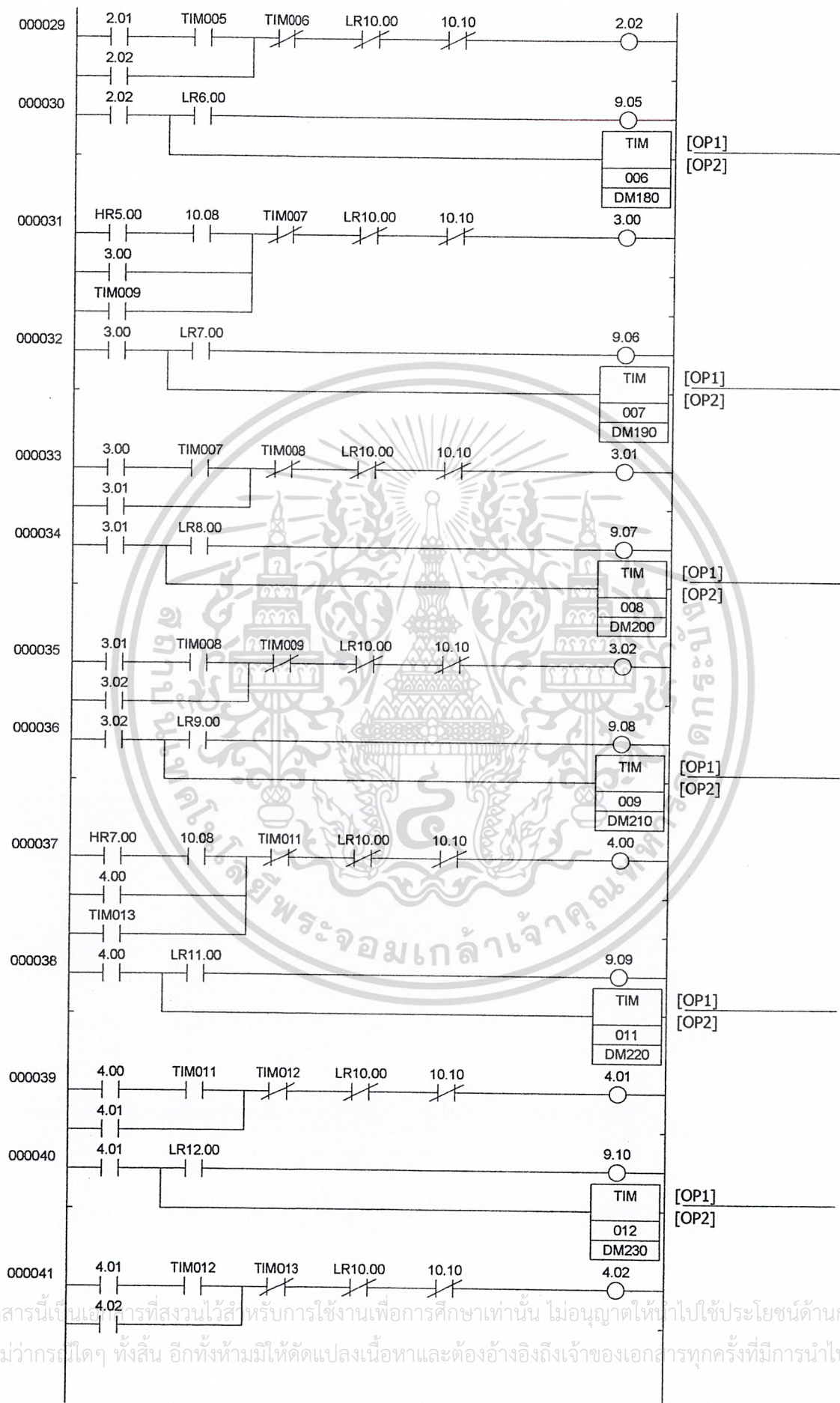


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

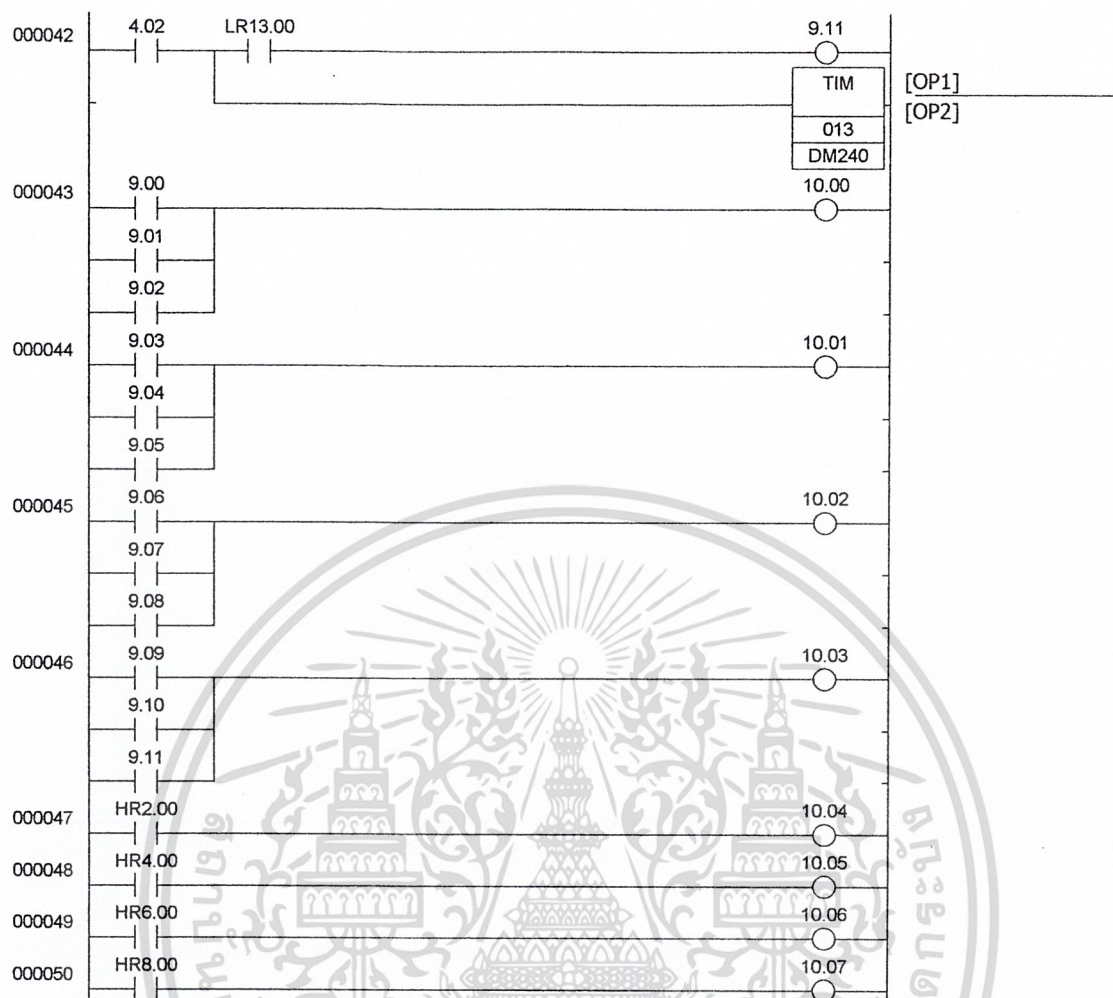


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้拿去ใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้





เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หน้าที่การทำงานของรีเลย์ภายใน

LR 0.00	ทำหน้าที่	เปิดการทำงานของระบบ
LR 10.00	ทำหน้าที่	ปิดการทำงานของระบบ
HR 1.00	ทำหน้าที่	เลือกใช้ Pump1 หรือไม่
LR 1.00	ทำหน้าที่	เลือกใช้ Pump1 Step1 หรือไม่
LR 2.00	ทำหน้าที่	เลือกใช้ Pump1 Step2 หรือไม่
LR 3.00	ทำหน้าที่	เลือกใช้ Pump1 Step3 หรือไม่
HR 3.00	ทำหน้าที่	เลือกใช้ Pump2 หรือไม่
LR 4.00	ทำหน้าที่	เลือกใช้ Pump2 Step1 หรือไม่
LR 5.00	ทำหน้าที่	เลือกใช้ Pump2 Step2 หรือไม่
LR 6.00	ทำหน้าที่	เลือกใช้ Pump2 Step3 หรือไม่
HR 5.00	ทำหน้าที่	เลือกใช้ Pump3 หรือไม่
LR 7.00	ทำหน้าที่	เลือกใช้ Pump3 Step1 หรือ ไม่
LR 8.00	ทำหน้าที่	เลือกใช้ Pump3 Step2 หรือ ไม่
LR 9.00	ทำหน้าที่	เลือกใช้ Pump3 Step3 หรือ ไม่
HR 7.00	ทำหน้าที่	เลือกใช้ Pump4 หรือ ไม่
LR 11.00	ทำหน้าที่	เลือกใช้ Pump4 Step1 หรือ ไม่
LR 12.00	ทำหน้าที่	เลือกใช้ Pump4 Step2 หรือ ไม่
LR 13.00	ทำหน้าที่	เลือกใช้ Pump4 Step3 หรือ ไม่
HR 2.00	ทำหน้าที่	เลือกให้ Pump1 หมุนทวน/ตามเข็มนาฬิกา
HR 4.00	ทำหน้าที่	เลือกให้ Pump2 หมุนทวน/ตามเข็มนาฬิกา
HR 6.00	ทำหน้าที่	เลือกให้ Pump3 หมุนทวน/ตามเข็มนาฬิกา
HR 8.00	ทำหน้าที่	เลือกให้ Pump4 หมุนทวน/ตามเข็มนาฬิกา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ผลงานวิจัยที่ได้รับการตีพิมพ์

- [1] Prawpan Inpota, Arunee Kongsakphaisal, Thitirat Mantim, Duangjai Nacapricha, Tadao Sakai, Norio Teshima, Noppadol Maneerat and Nathawut Choengcha. "PARALLEL CROSS INJECTION ANALYSIS SYSTEM FOR DETERMINATION OF ALBUMIN TO CREATININE RATIO IN URINE". Pure and Applied Chemistry International Conference 2011., Proceedings of PACCON 2011.



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

PARALLEL CROSS INJECTION ANALYSIS SYSTEM FOR DETERMINATION OF ALBUMIN TO CREATININE RATIO IN URINE

P. Inpota^{1,2*}, S. Teerasong^{1,2}, A. Kongsakphaisal², N. Maneerat⁴, D. Nacapricha^{1,3} and N. Choengchan^{1,2}

¹ Flow Innovation-Research for Science and Technology Laboratories (FIRST Labs),

² Department of Chemistry, Faculty of Science, King Mongkut's Institute of Technology Ladkrabang, Bangkok

10520 Thailand

³ Department of Chemistry and Center of Excellent for Innovation in Chemistry, Faculty of Science, Mahidol University, Bangkok 10400 Thailand

⁴ Department of Control Engineering, Faculty of Engineering, King Mongkut's Institute of Technology Ladkrabang, Bangkok 10520 Thailand

*E-mail: Praw.Inpota@gmail.com

Abstract: In this work, a parallel cross injection analysis (CIA) system was developed for determination of albumin to creatinine ratio. By two CIA units, simultaneous measurement of albumin and creatinine was achieved. Detection of albumin is based on using ion-association of albumin with tetrabromophenolphthalein ethyl ester (TBPE) in the presence of Triton X-100 at pH 3.1. A blue product is detected. For creatinine, the absorbance due to complex formation between creatinine and alkaline picrate was monitored. Linearity ranges up to 5 mg dL⁻¹ (Abs = 0.0037[albumin] - 0.0076, $r^2 = 0.990$) and up to 800 mg L⁻¹ (Abs = 0.005[creatinine] + 0.0059, $r^2 = 0.993$) were obtained. Sample and reagent loading were automatically carried out by 'home-built' software. The system was applied to spot collected urine samples. By paired *t*-test, the ratio measured by the proposed system and by the validating methods, was not significant difference ($t_{critical} = 3.18$, $t_{stat} = 0.30$) at 95 % confidence.

Introduction

Quantification of albumin in urine is important because appearance of albumin can indicate kidney damage. Trace amount of albumin in urine (30 to 300 mg/day) or 'microalbuminuria' is exploited as marker for an earliest stage of kidney dysfunction [1].

Standard method for human urine collection is a 24-h collection which is tedious for routine clinical diagnosis. Spot urine collection then become more interesting unless

variation of albumin level from to time. This can affect accuracy of the diagnosis. Therefore, 'albumin to creatinine ratio' has been utilized instead of albumin concentration alone for spot collected urine. The ratio ranging from 30 to 300 is regarded as high risk for microalbuminuria. Creatinine is widely employed as a correction factor when determination of substance in urine since its excreted amount per day is constant. Some publications proved that measurement of albumin:creatinine is more accurate [2].

There are a large number of analytical methods with various detections for measurement of albumin and creatinine in urine and other biological fluids. For albumin, the methods based on spectrofluorometry [3, 4], immunoassay [5, 6], chromatography [7, 8] and spectrophotometry [9-16] were reported. By means of spectrophotometric methods, an albumin-dye binding reaction is the most widely used [11-16]. There are several dyes, exploited as chromogenic reagents, e.g., bromocressol green [11, 12], bromophenol blue [13] and tetrabromophenolphthalein ethyl ester (TBPE) [14-16] which provided higher molar absorptivity than the other dyes [15]. For creatinine, the methods are usually based on reaction between creatinine and picric in alkaline medium which is so-called 'Jaffe's reaction' [17-20]. However, all the methods mentioned above were reported for determination of either albumin or creatinine which is not suitable for measurement of

albumin:creatinine ratio. Therefore, a method based on sequential injection analysis (SIA) [21] was proposed for determination of albumin and creatinine within single system. Nevertheless, these two analytes were still sequentially monitored.

This work, therefore, presents a single system for 'simultaneous' determination of albumin and creatinine. A concept of cross injection analysis or 'CIA' was employed for system developing. Recently, a notion to make flow conduit of a CIA and their applications were already presented [22-24].

By this work, in contrast of the former works [23-24], the CIA system was composed of two parallel acrylic platforms in which sample channel of each platform was linked by PTFE tubing. In addition, each analyte was individually detected by two detectors. Detection principle of albumin is based on albumin-TBPE reaction while the creatinine-alkaline picrate reaction was employed for monitoring of creatinine.

Materials and Methods

Standard and reagent preparation

All reagents used were of analytical reagent grade and deionized-distilled water purified by Milli-Q apparatus was used throughout.

Standard of human serum albumin (HSA, 1000 mg L⁻¹) and creatinine (1000 mg L⁻¹) were prepared by dissolving 0.1xxx g of HSA (Fluka, Switzerland) and 0.1xxx g of creatinine (Sigma, China) in water.

Stock solution of TBPE (1×10^{-3} mol L⁻¹) were prepared by dissolving TBPE (Aldrich, USA) in ethanol. Working solution was prepared to obtain 2×10^{-4} mol L⁻¹ TBPE in 0.2 % v/v Triton X-100.

Buffer solution was prepared by adding 0.1 mol L⁻¹ sodium acetate (Rankem, India) in 0.1 mol L⁻¹ acetic acid (Mallinckrodt, Thailand). This solution was then adjusted to a final pH of 3.1

A stock picric acid solution (0.1 mol L⁻¹) was prepared by dissolving 2.3 g of sodium picrate (Merck, USA) in water. Working picrate solution was prepared by mixing picric acid and sodium hydroxide (Rankem, India) to obtain a final concentration of 0.025 mol L⁻¹ picric acid and 0.2 mol L⁻¹ sodium hydroxide.

Human urine samples were spot collected from patients who attending diabetic clinic of Bangkok Metropolitan Administration Ladkrabang Hospital and from normal person. All samples were diluted with water prior to use. Any further pretreatment was not necessary.

Parallel CIA system and its protocol for operation

A Parallel CIA system for determination of albumin and creatinine is presented in Figure 1. Two acrylic platforms (unit A and B) were exploited. All conduits inside the platforms contain similar i.d. (1.0 mm) except for sample channel (2.0 mm).

The system was constructed from 1.0 mm i.d. PTFE tubing. Peristaltic pump (Ismatec, Switzerland), P1 was used to deliver water (Carrier, C) while P2 and P3 were employed for reagents (R) and sample/standard (S) aspiration, correspondingly. The 'home-built' software (a control box) was used to automatic manipulate the pump operation. The software is based on visual basic and programmable logic controller (PLC). By this software, small plugs of sample and reagent were simultaneous loaded into the system.

Protocol of the system is summarized in Table 1.

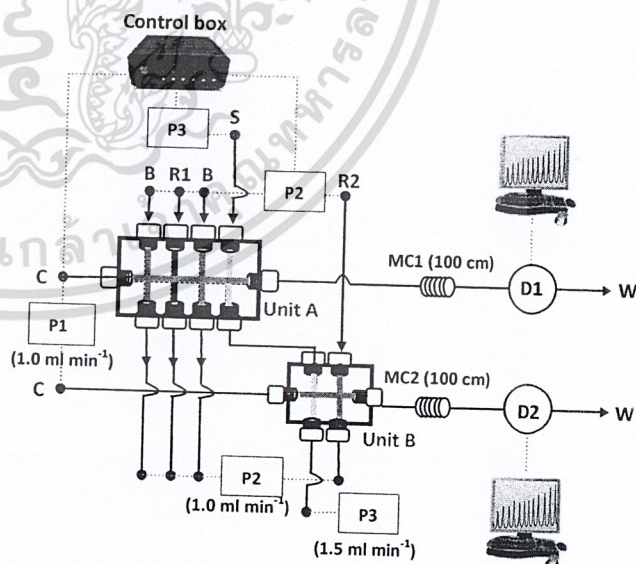


Figure 1. Parallel CIA system for simultaneous determination of albumin (Unit A) and creatinine (Unit B). P: Peristaltic pump, MC: Mixing coil, D1: Spectrometer (600 nm), D2: Green-LED Detector, C: Carrier (water), R1:

Urine sample preparation

TBPE (2×10^{-4} mol L⁻¹) + Triton X-100 (0.2 % v/v), R2: alkaline picrate (0.025 mol L⁻¹ in 0.2 mol L⁻¹ NaOH), B: acetate buffer (pH 3.1), S: sample/standard and W: Waste

Table1: Protocol sequences of the system.

Step	P1	P2	P3	Description
1	on	off	off	Auto zero with carrier
2	on	on (30 s)	on (30 s)	Simultaneous loading of reagent and sample
3	on	off	off	Monitoring of products
4	on	off	on	Cleaning of S-line with water

Results and Discussion

Manifold design

From the previous works [23-24], two CIA platforms were connected each other in series by carrier line (horizontal channel). Only one detector was used for 'sequentially' monitoring of two chasing product zones. Exploitation of delay coil was necessary in order to prevent zone overlapping. This affected directly on sample throughput.

In this work, two platforms were attached in parallel instead. Sample channel of one platform was linked to another platform by PTFE tubing. Each system was composed of its individual detector. By the proposed system, 'simultaneous' monitoring of two product zone was possible.

Optimization

Factors affecting sensitivity of the system were optimized by using uni-variable approach. Studied range and selected values were concluded in Table 2.

1. For albumin determination

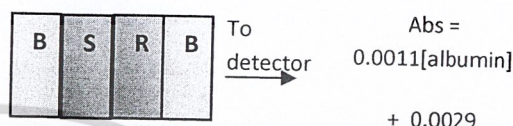
Appropriate Sequence for reacted zone

Two patterns of sequence for reacted zone in unit A were studied. These patterns are depicted in Figure 2. Sensitivity obtained by pattern (I) is lower. It was due to insufficient

mixing between two buffer zones (B) and reagent zone (R = TBPE).

According to K. Watla-id *et. al.* [16], TBPE must be initially protonated by buffer solution ahead of reacting with albumin. This implies that merging of two B zones with R zone prior to combine with sample zone (S) is required. As accounting on this reason, pattern II was therefore considered as the suitable sequence.

(I)



(II)

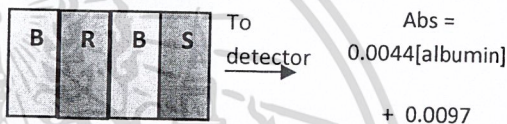


Figure 2. Schematic diagrams of patterns of sequence for reacted zone in CIA platform for albumin determination and their calibration equations.

Effect of TBPE

Concentration of TBPE affected directly on sensitivity [14-16]. Similar to this work, when TBPE concentration was increased (from 1 to 5×10^{-4} mol L⁻¹), absorbance reading was also increased. We selected a concentration of 4×10^{-4} mol L⁻¹ as compromising between sensitivity and reagent consumption.

Effect of Triton X-100

There has been a report that Triton X-100 act as a solubilizing agent for TBPE [15]. Inadequate concentration of the surfactant resulted in dissoluble TBPE. In this work, various concentrations of Triton X-100 were investigated. When the concentration was increased from 0.1 to 0.7 % v/v, the sensitivity was increased by 10 %. The solution became more viscous at higher concentration. Therefore, at 0.7 % v/v was chosen.

2. For creatinine determination

Effect of alkaline picrate

Alkaline picrate is a mixture of picric acid and NaOH. Concentration of these two chemicals can influence on sensitivity for creatinine determination [19-20]. When picric acid concentration was increased from 0.005 to 0.05 mol L⁻¹, absorbance reading was also enhanced. Similar results were obtained for NaOH concentration. Increasing in the alkaline solution resulted in improvement of sensitivity. On account of sensitivity and less reagent consumption, 0.025 mol L⁻¹ picric and 0.2 mol L⁻¹ NaOH was selected.

3. Effect of mixing coil length

Influence of length of MC 1 and MC 2 were examined from 50 to 200 cm. When length was increased, absorbance reading was also increased until reach its plateau at 100 cm. The length of 100 cm was selected for MC 1 and MC 2 because it gave sufficient sensitivity with satisfied throughput.

4. Effect of carrier line flow rate

Flow rates of the carrier (C in Figure 1), ranging from 0.5 to 2.5 ml min⁻¹ were studied. Increasing in flow rate resulted in shortening residence time of the product zone. This led to lower sensitivity. A flow rate at 1.0 ml min⁻¹ was regarded as suitable as compromising of sensitivity and analysis time.

Table 2: Summary for optimization study

Parameters	Studied range	Selected values
TBPE	1- 5 x 10 ⁻⁴ mol L ⁻¹	4 x 10 ⁻⁴ mol L ⁻¹
Triton X-100	0.1 – 1.0 % v/v	0.7 % v/v
Picric acid	0.005 – 0.05 mol L ⁻¹	0.025 mol L ⁻¹
NaOH	0.1 – 0. 5 mol L ⁻¹	0.25 mol L ⁻¹
Mixing coil length	50- 200 cm	100 cm
Carrier line flow rate	0.5-2.5 ml min ⁻¹	1.0 ml min ⁻¹

Analytical features

Some analytical features of the system are presented in Table 3. Examples of profiles were demonstrated in Figure 3A and 3B.

Table 3: Summary of analytical features of the system

Analytical features	Values	
	Albumin	Creatinine
Working range	5-50 mg L ⁻¹	100-800 mg L ⁻¹
Calibration equation	Abs = 0.0037[albumin] - 0.0076	Abs = 0.005[creatinine] + 0.0059
RSD (%)	2.1%	2.2%
Recovery (%)	94.5-114.7	102.3-120.4
LOD	4.5 mg L ⁻¹	45.9 mg L ⁻¹
Throughput (injection h ⁻¹)	60 h ⁻¹	60 h ⁻¹

Application to urine samples

The proposed system was applied to urine, collected from diabetic patients and to spiked urine samples.

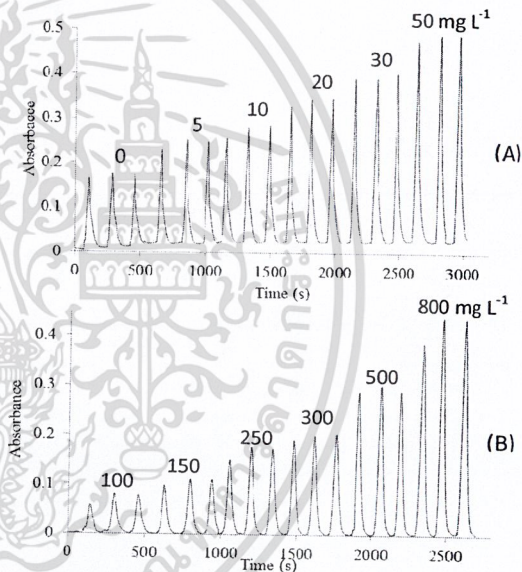


Figure 3. Examples of profiles for standard albumin (A) and creatinine (B).

Results were validated with a batchwise immunoassay and Jaffe methods for albumin and creatinine, respectively. Results, as Albumin (mg) / Creatinine (g) ratio, are shown in Table 4.

Table 4: Albumin/Creatinine ratio in urine samples, evaluated by the proposed CIA and by the validating methods.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Sample	Albumin : Creatinine ratio (mean)	
	This method	Validating method
Normal 1	n.d	n.d
Normal 2	n.d	n.d
Patient 1	12.6	14.5
Patient 2	69.4	58.5
Patient 3	17.0	18.1
Patient 4	7.6	11.5

n.d.: not detected

Statistical comparison of the results was carried out by paired *t*-test [25]. There was no evidence of significant difference between two method ($t_{critical} = 0.30$, $t_{stat} = 3.18$) at 95 % confidence.

However, patient 1 was considered as microalbuminuria since the ratio is in the range of 30 – 300 albumin to creatinine.

Conclusions

The parallel CIA system was successfully developed for simultaneous determination of two analytes. It offers rapid and automation operation with high versatility.

The system was effectively applied to measure albumin/creatinine ratio. Therefore, the developed system can be used as an alternative tool for assessment of microalbuminuria.

Acknowledgements

The authors would like to express their appreciation to KMITL research fund for financial support. We would also like to thank FIRST Lab @ Mahidol University for contribution.

References

- [1] P.A. Sarafidis and G.L. Bakris, J. NDT. 21(2006), pp. 2366-2374
- [2] W. Thaidhanaiswanya, W. Wongsankakorn and W. Leowattana, Siriraj Hosp Gaz. 44 (1992), pp. 503-508.
- [3] C. Jiang and L. Luo, Anal.Chim Acta. 506 (2004), pp. 171-175.
- [4] X. Hou, X. Tong, W. Dong, C. Dong and S. Shuang, Spectro.Acta part A. 66 (2007), pp. 552-556.
- [5] S. Aoyagi, T. Iwata, T. Miyasaka and K. Sakai, Anal. Chim. Acta. 436 (2001), pp. 103-108.
- [6] S. Miki, T. Kaneta and T. Imasaka, J. Chromatogr. 759 (2001), pp. 337-342.
- [7] J.H. Contois, C. Hartigan, L.V. Rao, L.M. Snyder and M.J. Thompson, Clin. Chim. 367 (2006), pp.150-155.
- [8] W.D. Comper, G. Jerums and T.M. Osicka, J.Clin Bioc. 37(2004), pp. 105-111.
- [9] H. Yatzidis, J. Clin. Chem. 23/5(1977), pp. 811-812
- [10] G.C. Luca and B.F. Reis, Spec. Acta. Part A. 60 (2004), pp. 579-583.
- [11] F.L. Rodkey, Clin. Chem. 11 (1965), pp. 478-487.
- [12] B.T. Doumas, W.A. Watson and H.G. Biffs, Clin.Chim. 258 (1997), pp. 21-30.
- [13] K.H. Schosinsky, M. Vargas, A.L. Esquivel and M.A. Chavarria, Clin.Chem. 33(2) (1987), pp. 223-226.
- [14] D.A.M. Zaia, F.R. Marques and T.B.V. Zaia, Braz. Arch. Biol. 48 (2005), pp. 385-388.
- [15] T. Sakai, Y. Kito, N. Teshima, S. Katoh, K. Watla-lad and K. Grudpan, Flow Injec. Anal. 24 (2007), pp 23-26.
- [16] K. Watla-lad, T. Sakai, N. Teshima, S. Katoh and K. Grudpan, Anal. Chim. Acta. 604 (2007), pp. 139-146.
- [17] H. Husdon and A. Ropopart, Clin. Chim. 14 (1968), pp. 222-238.
- [18] P.C. Falco, L.A.T. Genaro, S.M. Loret, F.B. Gomez, A.S. Cabeza and C.M. Legua, Talanta. 55 (2001), pp. 1079-1089.
- [19] T. Sakai, H. Ohta, N. Ohno and J. Imai, Anal. Chim. Acta. 308 (1995), pp. 446-450.
- [20] A.N. Araujo, J.L.F.C. Lima, B.F. Reis and E.A.G. Zagatto, Anal. Chim. Acta. 310 (1995), pp. 447-452.
- [21] W. Siangproh, N. Teshima, T. Sakai, S. Katoh and O. Chailapakul, Talanta. 79 (2009), pp. 1111-1117.
- [22] N.Choengchan, T. Mantim, K. Urisin, P. Wilairat and D. Nacapricha, *Cross Injection Analysis: A simple and Cost-effective Flow-base Technique*, the 14th International Conf. on Flow Injection Analysis, Germany, 2007
- [23] T. Mantim, K. Uraisin, N. Choengchan, P. Jittangprasert, W. Waiyawat, S. Foocharoen, P. Sirankpracha, N. Phumala Moralesy, P. Wilairat and D. Nacapricha, *Simultaneous measurement of iron and creatinine in urine by cross injection analysis*, the 11th Flow analysis, Spin, 2009
- [24] S.Janya, P. Sastranurak, T. Mantim, C. Boonpanaid N. Chauypasatwattana, N. Choengchan, D. Nacapricha and K. Uraisin, *An automatic phosphate analyzer by cross injection analysis*, Pure and applied Chemistry Conf., Ubon Rachathani, Thailand, 2010, pp. 33-36
- [25] J.N. Miller, I.C. Miller, Statistics and Chemometrics for Analytical Chemistry, Fourth ed., Pearson Education, 1993

ประวัติผู้แต่ง



นายจิตพงศ์ ป้องกัน

เกิดวันที่ 19 กุมภาพันธ์ พ.ศ. 2531

ที่อยู่ 271/127 หมู่ 8 ต.หนองตาแต้ม อ.ปราณบุรี จ.ประจวบคีรีขันธ์ 77120

โทรศัพท์ 081-2549667

E-mail : shinji_1902@hotmail.com

Facebook : www.facebook.com/kajitpong

ประวัติการศึกษา

2004 – 2006 โรงเรียนพรหมานุสรณ์จังหวัดเพชรบุรี

สายการเรียน วิทยาศาสตร์ – คณิตศาสตร์

2007 – 2010 สาขาวิชาการวัดและการควบคุม หลักสูตรวิศวกรรมแมคคาทรอนิกส์

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ประวัติผู้แต่ง



นายสมบุญ การมศิลป์

เกิดวันที่ 18 สิงหาคม พ.ศ. 2531

ที่อยู่ 36/1 หมู่ 13 ต. แม่วิน อ.แม่วาง จ. เชียงใหม่ 50360

โทรศัพท์ 087-1725187

E-mail : sbnw501@windowslive.com

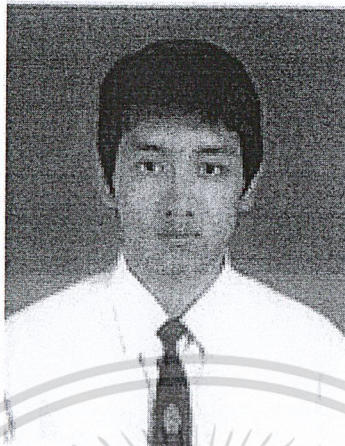
Facebook : www.facebook.com/karomsilp

ประวัติการศึกษา

- 2004 – 2006 โรงเรียนสันป่าตองวิทยาคม อำเภอสันป่าตอง จังหวัดเชียงใหม่
 สาขาการเรียน วิทยาศาสตร์ – คณิตศาสตร์
- 2007 – 2010 สาขาวิชาการวัดและการควบคุม หลักสูตรวิศวกรรมแมคคาทรอนิกส์
 คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ประวัติผู้แต่ง



นายเสฐียรพงศ์ บริรักษ์ภูสิฐ

เกิดวันที่ 25 พฤษภาคม พ.ศ. 2532

ที่อยู่ 144/30 หมู่ 12 ต.เจดีย์หัก อ.เมือง จ.ราชบุรี 70000

โทรศัพท์ 083-8398554

E-mail : sor_thongchai@hotmail.com

Facebook : <http://www.facebook.com/sorthongchai>

ประวัติการศึกษา

- 2004 – 2006 โรงเรียนเบญจมราชูทิศ ราชบุรี อำเภอเมือง จังหวัดราชบุรี
 สายการเรียน วิทยาศาสตร์ – คณิตศาสตร์
- 2007 – 2010 สาขาวิชาการวัดและการควบคุม หลักสูตรวิศวกรรมเมคคาทรอนิกส์
 คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้