



ระบบโครงข่ายเพื่อการควบคุมโรงงานแบบอัตโนมัติ

LAN FOR FACTORY AUTOMATION

โดย

น.ส. ลักษิกา นุ่มตึก รหัส 38014245

น.ส. สรียา แซ่จิ่ง รหัส 38014544

อาจารย์ที่ปรึกษา

รศ. สุพรรณ กุลพานิชย์

ผศ. วิริยะ กองรัตน์

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาเทคโนโลยีการวัดคุมทางอุตสาหกรรม

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2541

เลขที่.....  
เลขทะเบียน.....33968.....  
วัน, เดือน, ปี 23 ก.ย. 2542

การใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญาโท ปีการศึกษา 2541

ภาควิชาเทคโนโลยีการวัดคุมทางอุตสาหกรรม

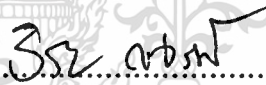
คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง  
เรื่อง ระบบ โครงข่ายเพื่อการควบคุมโรงงานแบบอัตโนมัติ

ผู้จัดทำ

1. นางสาวถกษิภา นุ่มถึก
2. นางสาวสรียา แซ่จิ่ง



.....อาจารย์ที่ปรึกษา  
(รศ.สุพรรณ กุลพาณิชย์)



.....อาจารย์ที่ปรึกษา  
(ผศ.วิริยะ กองรัตน์)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## ระบบโครงข่ายสำหรับโรงงานอัตโนมัติ



ถักษิกา นุ่มถัก

สรียา แซ่จิ่ง

อาจารย์ที่ปรึกษา

รศ.สุพรรณ กุลพานิชย์

ศศ. วิริยะ กองรัตน์

ปีการศึกษา 2541

### บทคัดย่อ

ปริญญานิพนธ์ฉบับนี้นำเสนอการสร้างระบบโครงข่ายสื่อสารร่วมที่มีขอบเขตอยู่ในพื้นที่จำกัด (Local Area Network : LAN) ของเครื่องควบคุมแบบโปรแกรมได้หรือ PLC (Programmable Logic Controller) โดยใช้เส้นใยแสงเป็นตัวกลางในการนำสัญญาณ พร้อมทั้งพัฒนาโปรแกรมให้ผู้ใช้สามารถสื่อสารกับโครงข่ายที่สร้างขึ้นผ่านระบบปฏิบัติการบน Microsoft Windows โดยใช้โปรแกรม Microsoft Visual Basic 5.0

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# LAN FOR FACTORY AUTOMATION

Luxsika Noomtug

Sareeya Saechueng

Advisor

Supun Kulpanich

Viriya Kongrut

1998



## ABSTRACT

This thesis presents creation and communication setting of Local Area Network (LAN) using Programmable Logic Controller (PLC) with fiber optic cable as transmission path. In addition to software development by Microsoft Visual Basic 5.0 to enable user interface on Microsoft Windows.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## กิตติกรรมประกาศ

ขอขอบคุณภาควิชาเทคโนโลยีการวัดคุมทางอุตสาหกรรมที่ให้การสนับสนุนเครื่องมือและอุปกรณ์ต่างๆ ตลอดจนสถานที่ที่ใช้ในการทำโครงการ

ขอขอบคุณอาจารย์ที่ปรึกษา ร.ศ. สุพรรณ กุลพานิชย์ และ ผ.ศ. วิริยะ กองรัตน์ ที่ให้การสนับสนุนทางด้านวิชาการ และให้คำแนะนำปรึกษาทางด้านเทคนิคต่างๆ ตลอดจนทั้งได้เป็นกำลังใจที่ดีตลอดมา

ขอขอบคุณอาจารย์ในภาควิชาทุกท่าน และเพื่อนๆ นักศึกษา 4F ทุกคนที่ให้คำปรึกษาในการพัฒนาโปรแกรม Visual Basic 5.0



คณะผู้จัดทำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# สารบัญ

เรื่อง	หน้า
<b>บทที่ 1 บทนำ</b>	<b>1</b>
1.1 ความสำคัญของโครงการ	1
1.2 วัตถุประสงค์ของการทำโครงการ	1
1.3 ขอบเขตของโครงการ	1
1.4 วิธีการดำเนินการ	1
1.5 ประโยชน์ที่คาดว่าจะได้รับจากการทำโครงการ	2
<b>บทที่ 2 ทฤษฎีที่ใช้ในการทำโครงการ</b>	<b>3</b>
2.1 ลักษณะทั่วไปของระบบโครงข่าย	3
2.2 การสื่อสารแบบ Host Link	12
2.3 ชุดโปรโตคอลที่ซีพี/ไอพี (TCP/IP)	14
2.4 โปรโตคอลในชั้นอินเทอร์เน็ต	16
2.5 ทฤษฎีการทำงานของโปรแกรม Visual Basic	22
2.6 เครื่องควบคุมแบบโปรแกรมได้	29
<b>บทที่ 3 การติดตั้งระบบโครงข่าย</b>	<b>34</b>
3.1 การติดตั้ง PLC (รุ่น C1000H)	34
3.2 การติดตั้ง Network Support Board	43
3.3 การติดตั้ง Bridge	45
3.4 การติดตั้ง Line Server	48
3.5 การติดตั้ง Host Link Unit	49
<b>บทที่ 4 โปรแกรมที่พัฒนาขึ้น</b>	<b>54</b>
4.1 ขั้นตอนการทำงานของโปรแกรม	54
4.2 โปรแกรม	55
<b>บทที่ 5 ผลการทดลอง</b>	<b>73</b>
5.1 อธิบายการใช้งานของโปรแกรมที่พัฒนาขึ้น	73
5.2 ผลการอ่านข้อมูลจากพื้นที่หน่วยความจำ PLC	74
5.3 ผลการเขียนข้อมูลลงบนพื้นที่หน่วยความจำ PLC	75

เรื่อง

หน้า

5.4 ผลการเรียก Command อื่นๆ

76

บทที่ 6 บทสรุป

78



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# สารบัญรูป

เรื่อง	หน้า
<b>บทที่ 2 ทฤษฎีที่ใช้ในการทำโครงงาน</b>	
รูปที่ 2.1 โครงสร้าง LAN ของ PLC	3
รูปที่ 2.2 ตัวอย่างรหัสส่งใน LAN	7
รูปที่ 2.3 การสื่อสารแบบย้อนกลับของ Token Ring	11
รูปที่ 2.4 การเชื่อมต่อสัญญาณระหว่าง CPU ของเครื่องคอมพิวเตอร์กับพอร์ตอนุกรม	13
รูปที่ 2.5 รูปแบบของอินเทอร์เน็ตแอดเดรส	20
รูปที่ 2.6 รูปแบบของอินเทอร์เน็ตแอดเดรสคลาส A	21
รูปที่ 2.7 รูปแบบของไอพีแอดเดรสคลาส A	22
รูปที่ 2.8 รูปแบบ Label ของโปรแกรม Visual Basic	25
รูปที่ 2.9 รูปแบบ CommandButton ของโปรแกรม Visual Basic	25
รูปที่ 2.10 รูปแบบ Frame ของโปรแกรม Visual Basic	25
รูปที่ 2.11 รูปแบบ Combo Box ของโปรแกรม Visual Basic	26
รูปที่ 2.12 รูปแบบ Image and Picture Box ของโปรแกรม Visual Basic	26
รูปที่ 2.13 รูปแบบ List Box ของโปรแกรม Visual Basic	26
รูปที่ 2.14 รูปแบบ Option Button ของโปรแกรม Visual Basic	27
รูปที่ 2.15 รูปแบบการคอนโทรล Command ของโปรแกรม Visual Basic	27
รูปที่ 2.16 โครงสร้างของ PLC	31
รูปที่ 2.17 Input/Output Scan และ Program Scan	32
<b>บทที่ 3 การติดตั้งระบบโครงข่าย</b>	
รูปที่ 3.1 การติดตั้ง Hardware ในระบบโครงข่าย	34
รูปที่ 3.2 โครงสร้าง Network Support Board	43
รูปที่ 3.3 โครงสร้าง Line Server	48
รูปที่ 3.4 รูป Dip switch ของ Line Server	49
รูปที่ 3.5 การขยายโครงข่ายโดยใช้ Repeater	49
รูปที่ 3.6 รูป Dip switch1 ของ Host Link Unit	50

เรื่อง	หน้า
รูปที่ 3.7 รูป Dip switch2 ของ Host Link Unit	51
<b>บทที่ 4 โปรแกรมที่พัฒนาขึ้น</b>	
รูปที่ 4.1 แสดงขั้นตอนการทำงานของ โปรแกรม	54
<b>บทที่ 5 ผลการทดลอง</b>	
รูปที่ 5.1 แสดงการอ่านข้อมูลจากพื้นที่หน่วยความจำ PLC	74
รูปที่ 5.2 ผลการเขียนข้อมูลลงบนพื้นที่หน่วยความจำ PLC	75
รูปที่ 5.3 การเรียก Command อื่นๆ	76
รูปที่ 5.4 ผลการเรียก Command อื่นๆ	77



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# สารบัญตาราง

เรื่อง	หน้า
<b>บทที่ 2 ทฤษฎีที่ใช้ในการทำโครงการ</b>	
ตารางที่ 2.1 การแบ่งระดับการทำงานของทีซีพี/ไอพี และ OSI	15
ตารางที่ 2.2 รูปแบบส่วนหัวอินเตอร์เน็ตโปรโตคอล (Internet Protocol Header Format)	18
ตารางที่ 2.3 รูปแบบชนิดของบริการ	18
ตารางที่ 2.4 รูปแบบแฟลก (Flag) ของอินเตอร์เน็ตโปรโตคอล	19
<b>บทที่ 3 การติดตั้งระบบโครงข่าย</b>	
ตารางที่ 3.1 การแบ่งพื้นที่หน่วยความจำของ PLC (รุ่น C1000H)	35
ตารางที่ 3.2 Flag ที่เกี่ยวข้องกับการสื่อสารด้วยคำสั่ง SEND(90) และRECV(98)	37
ตารางที่ 3.3 ส่วนควบคุมการการส่งของคำสั่ง SEND(90) ระหว่าง โครงข่าย	40
ตารางที่ 3.4 ส่วนควบคุมการการส่งของคำสั่ง SEND(90) ภายในโครงข่าย	40
ตารางที่ 3.5 ส่วนควบคุมการการรับคำสั่ง RECV(98) ระหว่างโครงข่าย	42
ตารางที่ 3.6 ส่วนควบคุมการการรับคำสั่ง RECV(98) ภายในโครงข่าย	42
ตารางที่ 3.7 พารามิเตอร์ในการติดตั้ง Network Support Board	44
ตารางที่ 3.8 พารามิเตอร์ในการติดตั้ง Bridge	45
ตารางที่ 3.9 พารามิเตอร์ในการตั้ง เส้นทางการสื่อสารของ LAN 1	46
ตารางที่ 3.10 พารามิเตอร์ในการตั้ง เส้นทางการสื่อสารของ LAN 2	47

# บทที่ 1

## บทนำ

### 1.1 ความสำคัญของโรงงาน

เนื่องจากปัจจุบัน โรงงานอุตสาหกรรมได้หันมาพัฒนาเป็นโรงงานในลักษณะโครงข่ายหรือโรงงานแบบอัตโนมัติ (FA :Factory Automation) มากขึ้น ซึ่งช่วยให้ผู้ใช้คนเดียวหรือหลายคนสามารถติดต่อหรือควบคุมระบบผ่านทาง Host Computer ได้จากสถานที่ต่างๆ โดยไม่ต้องลงไปภาคสนาม ทำให้เกิดความสะดวกรวดเร็ว ง่ายต่อการควบคุมและประหยัดค่าใช้จ่ายในระยะยาว ดังนั้นการเชื่อมต่อโครงข่ายของอุปกรณ์ควบคุมชนิดต่างๆ ที่ใช้ในโรงงานอุตสาหกรรม เช่น เครื่องควบคุมที่สามารถโปรแกรมได้หรือ PLC (Programmable Logic Controller) ซึ่งมีใช้กันอย่างแพร่หลาย จึงมีความสำคัญเป็นอย่างยิ่งในการสนับสนุนระบบโครงข่ายดังกล่าว

### 1.2 วัตถุประสงค์ของการทำโครงการ

- 1 เพื่อสร้างระบบโครงข่าย (LAN) ของ PLC รุ่น C1000H แบบ Token Ring โดยใช้เส้นใยแสง (Fiber Optic) เป็นตัวกลางในการเชื่อมต่อ
- 2 เพื่อพัฒนาโปรแกรม Microsoft Visual Basic สำหรับใช้ในการสื่อสารกับระบบที่สร้างขึ้น

### 1.3 ขอบเขตของโครงการ

- 1 สามารถสร้าง LAN แบบ Token Ring โดยใช้เส้นใยแสง (Fiber Optic) เป็นตัวกลางในการเชื่อมต่อ
- 2 สามารถพัฒนาโปรแกรม Microsoft Visual Basic เพื่อใช้ในการสื่อสารกับระบบที่สร้างขึ้นได้
- 3 สามารถอ่านและเขียนข้อมูลไปยัง PLC ในระบบที่สร้างขึ้นผ่านทาง Host Computer ได้

### 1.4 วิธีการดำเนินการ

- 1 ศึกษา LAN
- 2 ศึกษา Hardware และ PLC ที่จะใช้ในระบบ
- 3 สร้าง LAN โดยใช้เส้นใยแสงเป็นตัวกลางในการเชื่อมต่อ

เอกสารที่ 4 ทดสอบการทำงานของระบบการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

5 ศึกษาโปรแกรม Microsoft Visual Basic 5.0 และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- 6 พัฒนาโปรแกรมบน Microsoft Visual Basic 5.0 เพื่อใช้ในการสื่อสารกับระบบที่สร้างขึ้น
- 7 ทดสอบการทำงานและการสื่อสารภายในระบบ
- 8 ปรับปรุงแก้ไข

## 1.5 ประโยชน์ที่คาดว่าจะได้รับจากการทำวิจัย

- 1 ได้รับความรู้เรื่องการสื่อสารใน LAN
- 2 สามารถเข้าใจการสร้าง LAN ของเครื่อง PLC
- 3 ได้รับความรู้เรื่องการพัฒนาโปรแกรม โดยใช้ Microsoft Visual Basic
- 4 สามารถพัฒนาโปรแกรมโดยใช้ Microsoft Visual Basic เพื่อใช้ในการติดต่อกับระบบที่สร้างขึ้นได้
- 5 ได้รับความรู้และประสบการณ์เพิ่มเติมในการใช้งาน PLC
- 6 ได้เรียนรู้ถึงวิธีการแก้ปัญหาในการทำงาน การทำงานร่วมกับผู้อื่น ตลอดจนฝึกการค้นคว้าด้วยตนเอง

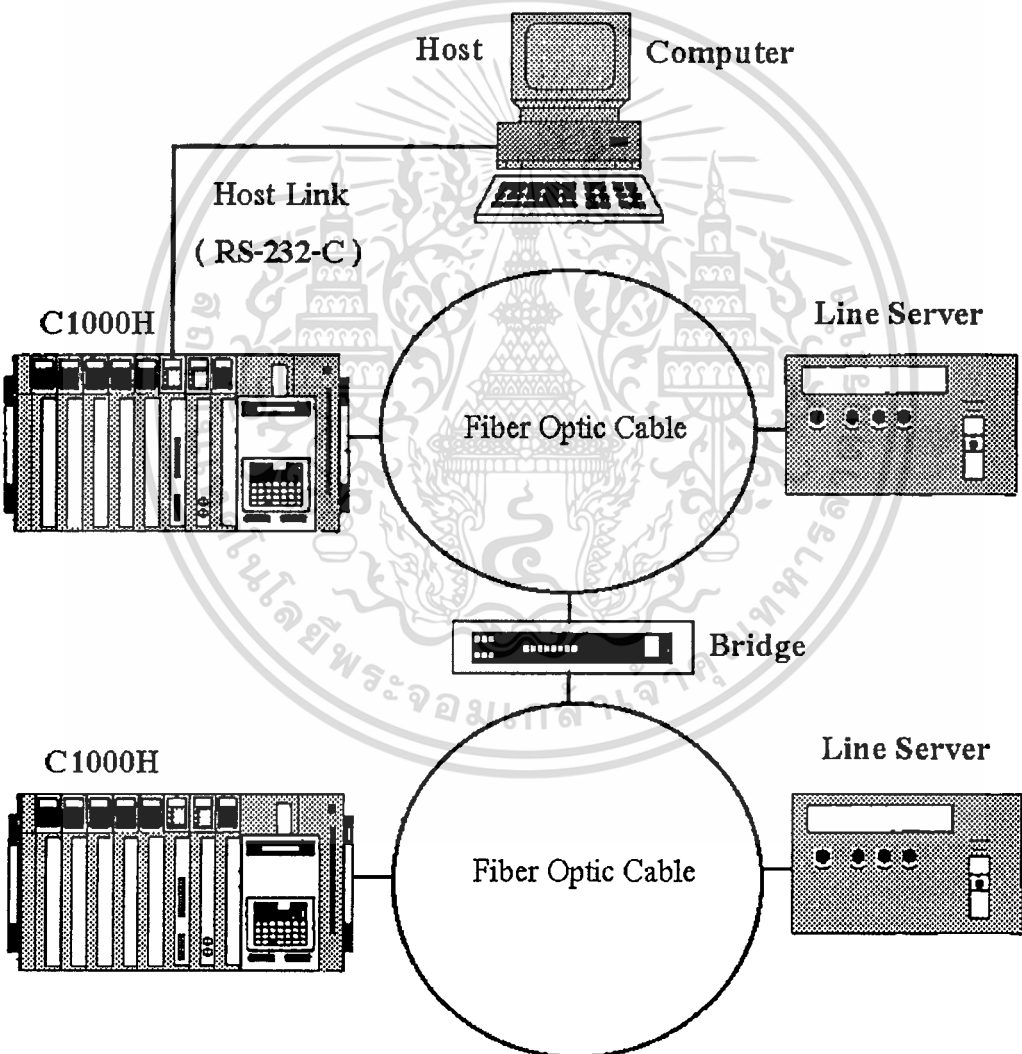


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทที่ 2

### ทฤษฎีที่ใช้ในการติดตั้ง Local Area Network (LAN)

#### 2.1 ลักษณะทั่วไปของระบบโครงข่าย



เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ โดย บริษัท ทรู คอร์ปอเรชั่น จำกัด (มหาชน) ทุกรายการสงวนลิขสิทธิ์ไว้ ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 2.1.1 ความหมายและความสำคัญของ Local Area Network (LAN)

โลคัลเอเรียเน็ตเวิร์ค (Local Area Network) หรือ LAN มีความหมายตามคำนิยามของ IEEE หรือ ISO คือ “โครงข่ายที่เป็นไปได้สำหรับการส่งข้อมูลที่มีความเร็วขนาดกลางไปจนกระทั่งความเร็วสูงที่ถูกจำกัดขอบเขตอยู่ภายในสำนักงาน อาคาร หรืออาคารหนึ่งๆ” นอกจากนี้สำหรับเครือข่ายอื่นๆ ที่นอกเหนือจาก LAN แล้ว โครงข่ายที่มีขอบเขตกว้างขวางทั่วประเทศจะถูกเรียกว่า WAN(Wide Area Network) และโครงข่ายที่ขนาดระหว่าง LAN กับ WAN จะถูกเรียกว่า MAN (Metropolitan Area Network)

สำหรับโครงสร้างของ LAN นั้นจะเป็นโครงข่ายที่เชื่อมต่อโฮตคอมพิวเตอร์ เวิร์ดโปรเซสเซอร์ ไมโครคอมพิวเตอร์ หรือโทรสาร เป็นต้นที่อยู่กระจายภายในอาคารหรือภายในอาคารเดียวกันของสำนักงาน หรือโรงงาน ด้วยความเร็วในการส่งประมาณ 1 Mbps- 100 Mbps โดยมีจุดมุ่งหมายเพื่อให้การใช้รีสอร์ซ (Resource)เช่นข่าวสารหรืออุปกรณ์ร่วมกัน

LAN จะมีพื้นที่เป้าหมายของโครงข่ายค่อนข้างแคบ นอกจากนี้รูปแบบการติดตั้งโดยทั่วไปจะคล้ายคลึงกันและมีความเสถียรภาพ LAN มีบทบาทเพื่อเพิ่มประสิทธิภาพของงานในสำนักงานของแต่ละองค์กร เพิ่มความก้าวหน้าของการผลิตและลดค่าใช้จ่ายโดยมีการทำงานในลักษณะของสำนักงานอัตโนมัติ :OA(Office Automation) โรงงานอัตโนมัติ:FA (Factory Automation) กล่าวคือการนำเอาอุปกรณ์แต่ละชนิดมาใช้งานร่วมกัน

ดังนั้นเราอาจจะกล่าวได้ว่า LAN นั้นมีบทบาทสำหรับการสื่อสารข่าวสารที่ก้าวหน้าและเป็นอิสระด้วยการเชื่อมต่อเทอร์มินัลของแต่ละแผนกภายในองค์กร

สำหรับผลลัพธ์ของการนำเอา LAN มาใช้นั้นคือ

1. การใช้แหล่งข่าวสารร่วม เทอร์มินัลใดๆ ที่ต่อเข้ากับโครงข่ายสามารถติดต่อสื่อสารกันได้อย่างอิสระเมื่อเป็นเช่นนี้ข่าวสารทั้งหมดที่ถูกเก็บรักษาไว้ที่อุปกรณ์ไฟล์ร่วมสามารถถูกเรียกออกมาใช้ร่วมกันด้วยฟังก์ชันสื่อสารความเร็วสูงของ LAN ซึ่งจะทำการใช้ข่าวสารร่วมกันได้ทั้งองค์กร จึงอาจกล่าวได้ว่าเป็นการระบบไฟล์อิเล็กทรอนิกส์หรือระบบค้นหาดาต้าเบสนั่นเอง
2. ความรวดเร็วในการส่งข่าวสาร การ ส่งเอกสารโดยใช้คนนำส่งแต่ละแผนกนั้นสามารถเปลี่ยนเป็นการส่งจากเทอร์มินัลของแผนกหนึ่ง ไปยังอีกแผนกหนึ่งหรือการส่งข่าวสารเดียวกันโดยใช้อิเล็กทรอนิกส์เมล์และสามารถกระทำด้วยความรวดเร็ว
3. ความประหยัดเนื่องจากการใช้อุปกรณ์ร่วมกัน อุปกรณ์ราคาแพงบางชนิดไม่จำเป็นต้องใช้ในลักษณะส่วนตัว เช่น เครื่องพิมพ์เลเซอร์ ดังนั้นหากนำเอาฟังก์ชันการสื่อสารความเร็วสูงของ LAN มาใช้งานแล้วเทอร์มินัลทั้งหมดสามารถใช้อุปกรณ์ราคาแพงร่วมกัน ได้ จึงทำให้ประหยัดค่าใช้จ่าย

## 2.1.2 โครงสร้างของ LAN ประกอบด้วย

1. เทอร์มินัล (สถานี) อุปกรณ์ที่ถูกต่อเชื่อมเข้ากับ LAN นั้นได้แก่ไมโครคอมพิวเตอร์ เวิร์คสเตชัน โฮสคอมพิวเตอร์ และเครื่องพิมพ์ เป็นต้น อุปกรณ์ดังกล่าวจะถูกเรียกว่าเป็นเทอร์มินัลของ LAN และโดยทั่วไปอาจจะเรียกว่าสถานีหรือเซิร์ฟเวอร์ก็ได้

2. คอนโทรลเลอร์ อุปกรณ์ที่ทำหน้าที่ควบคุมการสื่อสารเช่นควบคุมความผิดพลาด ควบคุมแอดเดรสเพื่อที่จะทำให้การส่งข่าวสารระหว่างสถานีสามารถกระทำได้อย่างถูกต้อง

3. ทรานซิปเวอร์ อุปกรณ์ที่รับข่าวสารจากคอนโทรลเลอร์เปลี่ยนเป็นสัญญาณแสงหรือสัญญาณไฟฟ้าที่เหมาะสมกับเส้นทางแล้วทำการส่งออกไป หรือรับเอาสัญญาณจากเส้นทางส่งมาแปลงเป็นสัญญาณข่าวสารแล้งส่งต่อไปยังคอนโทรลเลอร์

4. เส้นทางส่ง ตัวกลางที่เชื่อมต่อโหนดเข้ากับระบบโครงข่ายเพื่อให้การรับส่งข้อมูลระหว่างสถานีใดๆ สามารถกระทำได้ ตัวกลางที่ใช้ใน LAN นั้นได้แก่ ทวิสเตอร์เคเบิล โคแอกเชียลเคเบิลและเส้นใยแสง นอกจากนี้ช่วงที่ไม่สามารถใช้เคเบิลก็จะใช้ระบบวิทยุ

สำหรับทรานซิปเวอร์และ คอนโทรลเลอร์เมื่อรวมกันแล้วจะถูกเรียกว่า”โหนด”

## 2.1.3 เนตเวิร์คโทโปโลยี

รูปแบบของโครงข่ายของเส้นทางส่งที่เชื่อมต่อระหว่างสถานีจะถูกเรียกว่า”เน็ตเวิร์คโทโปโลยี” สำหรับเน็ตเวิร์คโทโปโลยีจะถูกแบ่งออกเป็น 3 จำพวกคือ แบบสตาร์ แบบบัส และแบบริงค์ (แบบลูป) ในที่นี้จะขอกล่าวถึงเฉพาะแบบริงค์ซึ่งเป็นรูปแบบที่ใช้ในโครงข่ายนี้

โครงข่ายแบบริงค์ (แบบลูป) เส้นทางส่งความเร็วสูงรูปวงแหวนจะถูกใช้ร่วมกันโดยแต่ละโหนด การเชื่อมต่อโหนดเข้าไปนั้นจะเป็นลักษณะการอินเตอร์เฟซระบบ สำหรับโครงข่ายแบบริงค์นั้น ถ้าหากมีการส่งข่าวสารสองทิศทางพร้อมกันก็จะทำให้เกิดการชนกันของสัญญาณขึ้น ดังนั้นการส่งข่าวสารต้องมีการส่งเพียงทิศทางเดียว เมื่อเป็นเช่นนี้ ข่าวสารที่ถูกส่งมาจากแต่ละโหนดจะถูกรับและส่งต่อตามลำดับ และถ้าหากเป็นข่าวสารของโหนดตัวเองก็จะคัดลอกนำไปจัดการต่อไป และจะส่งต่อไปยังสถานีถัดไป ลักษณะพิเศษของโครงข่ายแบบริงค์ได้แก่

1. ฟังก์ชันควบคุมการสื่อสารถูกกระจายไปที่แต่ละโหนด ซึ่งทำให้ค่าใช้จ่ายในโครงข่ายจะเป็นสัดส่วนตามจำนวนโหนด ดังนั้นสำหรับ LAN ที่มีขนาดเล็กและกลางนั้นจะช่วยประหยัดค่าใช้จ่าย

2. สัญญาณจะถูกสร้างใหม่และทวนสัญญาณต่อ ดังนั้นสามารถใช้สร้างระบบที่มีขนาด

ใหญ่ แต่จำเป็นต้องเพิ่มฟังก์ชันควบคุมการสื่อสารในกรณีที่สัญญาณที่ถูกสร้าง  
ไม่ว่ากรณีใดๆ และทวนจากโหนดใดๆ ได้รับความเสียหาย อ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. เนื่องจากเป็นโครงข่ายในลักษณะแอดฮ็อก ดังนั้นถ้าโหนดใดๆ เสียหายจะทำให้การสื่อสารทั้งระบบล้มเหลว เมื่อเป็นเช่นนี้ฟังก์ชันต่อผ่านหรือโครงข่ายซ้อนจะถูกนำมาใช้ในลักษณะของการสร้างเส้นทางกลับในกรณีที่มีความเสียหายที่ตำแหน่งใดๆ นอกจากนี้สำหรับโครงข่ายสื่อสารแบบริงค์ที่มีโหนดใดโหนดหนึ่งควบคุมการสื่อสารของ LAN ทั้งระบบจะถูกเรียกว่า "ลูป" (Loop)

#### 2.1.4 ตัวกลางในการส่ง

ตัวกลางในการส่งสำหรับ LAN นั้นจะใช้ ทวิสแพร่เคเบิล(สายคู่ตีเกลียว) โคแอกเชียลเคเบิล และเส้นใยแสง ในที่นี้จะกล่าวถึงเฉพาะเส้นใยแสงซึ่งใช้ในโครงข่ายนี้

เคเบิลเส้นใยแสง (Fiber Optic) โดยทั่วไปแล้วจะเป็นเคเบิลที่มีซิลิกาออปติกไฟเบอร์เป็นตัวนำขนาดเส้นผ่านศูนย์กลาง 0.05-0.1 mm. ในการนำเอาสายเคเบิลเส้นใยแสงมาเป็นเส้นทางส่งนั้นจำเป็นต้องมีอุปกรณ์ส่งและรับแสงซึ่งจะทำหน้าที่แปลงสัญญาณไฟฟ้าจากสถานีให้เป็นสัญญาณแสง สำหรับอุปกรณ์กำเนิดแสงได้แก่ LED(Light Emitting Diode) หรือ LD(Laser Diode) ส่วนอุปกรณ์รับแสงนั้นได้แก่ PD(Photo Diode) หรือ APD(Avalanche Photo Diode)

การใช้เส้นใยแสงเป็นตัวกลางในการส่งนั้น ถ้าเปรียบเทียบกับตัวกลางอื่นๆ แล้วจะมีราคาแพงกว่า แต่จะมีข้อดีตรงที่เป็นเคเบิลที่มีขนาดเล็กและเบา รวมทั้งสามารถใช้ในการส่งย่านความถี่กว้างความเร็วสูง นอกจากนี้จะใช้ในการส่งสัญญาณข้อมูลแล้วยังสามารถใช้ในการประกอบ LAN ที่รวมเอาเม็ดเดี่ยวแต่ละชนิด เช่น เสียงและภาพ เป็นต้น และเนื่องจากจะไม่รับอิทธิพลของสัญญาณรบกวนของสนามแม่เหล็กไฟฟ้าจากภายนอก ดังนั้นคุณภาพในการส่งจะคงที่จึงเหมาะที่จะใช้ในสถานที่ที่มีสัญญาณรบกวนประเภทนี้อยู่มาก เช่น ในโรงงาน เป็นต้น แต่ในทางตรงข้ามการเชื่อมต่อโหนดกับสายเคเบิลจะเป็นเรื่องยุ่งยาก ดังนั้นการเพิ่มหรือเคลื่อนย้ายโหนดจะลำบาก ซึ่งเป็นข้อจำกัดของระบบ

#### 2.1.5 วิธีการส่ง

การส่งสัญญาณ LAN นั้นได้แก่วิธีการส่งแบบเบสแบนด์ ซึ่งสัญญาณดั้งเดิมจากสถานี(โดยทั่วไปจะเป็นดิจิทัล)จะถูกส่งในลักษณะเช่นนั้นเข้าไปในชั้นแนล ส่วนวิธีการส่งแบบบรอดแบนด์สัญญาณดั้งเดิมจะถูกโมดูเลทแล้วเข้าฟรีควีนซีมัลติเพลกซ์ (TDM) เป็นกลุ่มสัญญาณเข้าไปในชั้นแนล

วิธีการส่งแบบเบสแบนด์ วิธีการส่งสัญญาณแบบเบสแบนด์ จะมีชั้นแนลเพียงชั้นแนลเดียวเท่านั้นที่สามารถส่งข่าวสารได้ ดังนั้นเพื่อที่จะให้หลาย ๆ สถานีสามารถใช้ชั้นแนลร่วมกันได้ก็ต้องเพิ่มความเร็วในการส่งให้สูงขึ้น และแบ่งเวลาใช้งานให้กับแต่ละสถานี

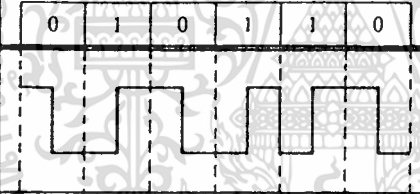
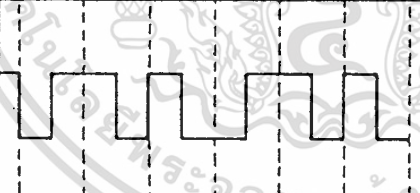
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ลักษณะพิเศษของวิธีการส่งแบบเบสแบนด์นั้นจะเป็นดังต่อไปนี้

- 1) วงจรรับส่งจะง่ายเนื่องจากไม่ต้องใช้โมเดม
- 2) เนื่องจากรูปแบบการส่งจะใช้สัญญาณดิจิทัล ดังนั้นความกลมกลื่นกับการสื่อสารข้อมูลจะมีมาก

LANที่ใช้วิธีการส่งแบบเบสแบนด์นั้นโดยทั่วไปจะมีวิธีการชิงโครนัสด้วยวิธีการดึงเอาไทม์มิงจากสัญญาณที่รับเข้ามา ดังนั้นรหัสสัญญาณส่งที่ใช้แทนสัญญาณข่าวสารนั้นจำเป็นต้องมีรูปแบบที่ง่ายสำหรับให้ทางด้านรับสามารถดึงเอาไทม์มิงออกมาและสามารถที่จะตรวจตราคุณภาพในการส่ง การตรวจหาความผิดพลาดของสัญญาณส่งซึ่งโดยทั่วไปแล้วจะใช้รหัสแมนเชสเตอร์และดิฟเฟอเรนเชียลแมนเชสเตอร์ เป็นต้น

โดยที่วิธีเข้ารหัสทั้งสองแบบนี้จะต้องมีการกลับทิศทางของพัลส์ตรงตำแหน่งกลาง ดังนั้นการปรับบิตชิงโครนัสจะทำได้ง่าย

รหัสส่ง	สัญญาณดั้งเดิม						หมายเหตุ
	0	1	0	1	1	0	
รหัสแมนเชสเตอร์							สัญญาณดั้งเดิม[1]จะได้ค่า[01] สัญญาณดั้งเดิม[0]จะได้ค่า[10]
รหัสดิฟเฟอเรนเชียลแมนเชสเตอร์							สัญญาณดั้งเดิม[0]จะกลับขั้ว เริ่มต้นรหัส สัญญาณดั้งเดิม[1]จะไม่กลับขั้ว เริ่มต้นรหัส

รูปที่ 2.2 แสดงตัวอย่างรหัสส่งใน LAN

วิธีการส่งแบบบรอดแบนด์ วิธีการส่งแบบบรอดแบนด์นั้นจะใช้วิธีการมัลติเพลกซ์เพื่อสร้างเป็นหลายชั้นแนลด้วยคลื่นพาห้หลายคลื่นต่อเส้นทางส่งหนึ่งด้วย ดังนั้นในเวลาเดียวกันสามารถใช้ในการส่งข่าวสารได้หลายชุด และนอกเหนือจากการใช้ในการส่งข้อมูลแล้วยังใช้ในการส่งสัญญาณเสียงภาพเทเลคอนเฟอเรนหรือ CATV ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

วิธีการส่งแบบบรอดแบนด์นั้น โดยทั่วไปจะใช้วิธี FDM เป็นวิธีการมัลติเพล็กซ์ ดังนั้นการรับส่งสัญญาณของ LAN จะแบ่งชั้นแชนเนลออกเป็น 2 ชั้นแชนเนล สำหรับเป็นชั้นแชนเนลรับและชั้นแชนเนลส่ง โดยชั้นแชนเนลแต่ละคู่จะมีทิศทางสวนกัน

กล่าวคือสัญญาณที่ถูกส่งออกจากโหนดนั้นจะถูกส่งเข้าชั้นแชนเนลส่งและเมื่อถึงปลายทางส่งก็จะวกเข้ามายังชั้นแชนเนลรับ

ลักษณะพิเศษของวิธีการส่งแบบบรอดแบนด์นั้น ได้แก่

- 1) เนื่องจากจำเป็นต้องใช้โมเดมความถี่สูง (RF) ดังนั้นค่าใช้จ่ายจะมากขึ้น
- 2) มีความกลมกลืนกับสัญญาณอะนาล็อกสูง ดังนั้นสามารถใช้กับการส่งมีเดียแต่ละชนิดรวมเข้ากันได้
- 3) สามารถส่งด้วยความจุสูงด้วยจำนวนชั้นแชนเนลหลายชั้นแชนเนลด้วย FDM
- 4) สามารถส่งในระยะทางไกลที่ประหยัดค่าใช้จ่าย

วิธีการโมดูเลทที่ใช้สำหรับวิธีการส่งแบบบรอดแบนด์นั้น ได้แก่ AM (Amplitude Modulation), FM (Frequency Modulation), PM (Phase Modulation) โดยที่ข่าวสาร “1” และ “0” ของสัญญาณข้อมูลจะเปลี่ยนเป็นข่าวสารตามปริมาณแอมพลิจูด ความถี่ และเฟส แล้วส่งออกไปด้วยสัญญาณอะนาล็อก

สำหรับ LAN นั้นจะใช้วิธีการ FM และ PM ซึ่งจะมีความทนทานต่อสัญญาณรบกวนและการเปลี่ยนระดับได้สูง

### 2.1.6 การขยายขนาดของ LAN

การขยายขนาดของ LAN นั้นสามารถทำได้ 2 วิธีคือ วิธีแรกจะเป็นการขยายขอบเขตของเน็ตเวิร์คเดี่ยว ๆ นั้นด้วยอุปกรณ์ที่ใช้เชื่อมต่อหรือ อีกวิธีหนึ่งก็เป็นการเชื่อมต่อโลคัลแอเรียเน็ตเวิร์คหลาย ๆ ตัวด้วยอุปกรณ์เชื่อมต่อ สำหรับอุปกรณ์ที่ใช้เชื่อมต่อ LAN นั้นปัจจุบันเราสามารถแบ่งแยกอุปกรณ์เหล่านี้ออกได้เป็น 4 จำพวกใหญ่ ดังนี้คือ รีพีทเตอร์ (Repeater), บริดจ์ (Bridge), รุทเตอร์ (Router) และเกตเวย์ (Gateway)

#### บริดจ์ (Bridge)

ในกรณีการเชื่อมต่อ LAN หลายตัวที่มีรูปร่างทางฟิสิกส์และมีเคเบิลหรือสายเคเบิลที่ต่างบริดจ์จะเป็นอุปกรณ์สำคัญที่ใช้เชื่อมต่อเน็ตเวิร์คในลักษณะนี้ จากหลักการที่ว่า บริดจ์มีฟังก์ชันการทำงานที่มากกว่ารีพีทเตอร์ คือมันจะทำหน้าที่เหมือนสถานีธรรมชาติสถานีหนึ่งใน LAN ดังนั้นบริดจ์สามารถรับและตรวจสอบแพคเกจที่ส่งผ่านเข้ามาในเน็ตเวิร์ค ยิ่งกว่านั้นฟังก์ชันของบริดจ์จะเปรียบเทียบกับแอดเดรสของแพคเกจที่รับได้กับตารางแอดเดรสเพื่อตัดสินใจว่าควรจะทิ้งหรือส่งต่อ

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แพคเกจนั้นเมื่อบริดจ์ตรวจพบแพคเกจที่ถูกส่งไปยัง LAN อื่น บริดจ์จะเก็บแพคเกจนั้นและเตรียมส่งต่อด้วยการแอกเซส LAN ดำรงส่งออก

การทำงานของบริดจ์

- เมื่อแพคเกจถูกส่งมายังบริดจ์ บริดจ์จะรับแพคเกจนั้นและจากนั้นมันจะเก็บพักไว้ในบัฟเฟอร์ (Buffer) เพื่อเตรียมตรวจสอบต่อไป
- หลังจากที่ส่วนตรวจสอบแพคเกจว่างลง แพคเกจจะถูกส่งเข้าไปตามลำดับหน้า-หลัง
- ถ้าแพคเกจนั้นมีที่อยู่ปลายทางอยู่ใน LAN อื่นมันจะถูกเก็บไว้และรอการส่ง หรือมิเช่นนั้นแพคเกจนั้นจะถูกขจัดทิ้ง

สำหรับวิธีการข้างต้น แพคเกจจะต้องเสียเวลาบ้างในการรอการตรวจสอบที่บริดจ์บัฟเฟอร์ (Bridge Buffer) ซึ่งจะทำให้เกิดความล่าช้าในการส่ง อย่างไรก็ตามในเมื่อบริดจ์จะเลือกแพคเกจที่มีปลายทางที่อยู่ใน LAN อื่นเท่านั้น ดังนั้นบริดจ์จะทำหน้าที่ลดความหนาแน่นของโหลด (Load) ทั่วเนตเวิร์คทั้งหมด

ระดับโปรโตคอลที่บริดจ์ทำงานจะเห็นได้ว่าการทำงานของบริดจ์นั้นจะครอบคลุมชั้นดาต้าลิงก์ (Data-Link) ในขณะที่พีทีทีเตอร์นั้นจะทำงานเพียงชั้นฟิสิกส์เท่านั้น ดังนั้นในกรณีของบริดจ์ บริดจ์จะปฏิบัติงานการส่งแพคเกจระหว่างสถานีและแอกเซสชันแนตในแต่ละ LAN ที่เชื่อมติดอยู่ ดังนั้น LAN ที่มีแอกเซสคอนโทรลที่ต่างกันสามารถถูกเชื่อมต่อโดยใช้บริดจ์ และเนื่องจากการส่งแพคเกจออกไปไม่สามารถทำในทันทีทันใด ดังนั้นการเตรียมบัฟเฟอร์เพื่อใช้ในการรอการส่งจึงมีความจำเป็น

เมื่อเปรียบเทียบฟังก์ชันการทำงานของบริดจ์กับพีทีทีเตอร์แล้วจะเพิ่มขึ้นมากมายโดยเฉพาะฟังก์ชันที่เกี่ยวกับแอกเซส LAN ที่มีมีเดียแอกเซสคอนโทรลที่ต่างกัน ดังนั้นการเพิ่มงานปฏิบัติมากมายเท่าไรจะก่อให้เกิดความล่าช้าของการส่ง ดังนั้นบริดจ์จึงได้รับความสนใจจากการเชื่อม LAN ที่เหมือนกัน เพราะไม่เพียงแต่จะลดฟังก์ชันการทำงานลดลงเท่านั้น มันยังสามารถลดโหลดที่มีจำนวนสถานีมากและขนาดเนตเวิร์คที่ใหญ่ขึ้นด้วย

โครงสร้างของ LAN ที่ถูกเชื่อมต่อโดยบริดจ์

LAN ที่มีโครงสร้างชั้นดาต้า-ลิงก์เหมือนกันนั้นสามารถเชื่อมต่อโดยใช้บริดจ์ โดยที่โครงสร้างของเนตเวิร์คที่ใช้บริดจ์เป็นเชื่อม สามารถแยกออกเป็นโครงสร้างใหญ่ๆ 3 โครงสร้าง คือ 1) โครงสร้างแบบริงซึ่งจะประกอบด้วย LAN หลายตัวเชื่อมต่อผ่านบริดจ์เป็นรูปวงแหวน 2) โครงสร้างเป็นชั้น ๆ (Hierachy) ซึ่งจะใช้เป็นแบคบอนด์ (Back Bond) LAN มาเชื่อมต่อผ่านบริดจ์ และ 3) แบบดาว (Star) ที่มี LAN หลาย ๆ ตัวเชื่อมต่อไปยังบริดจ์อันเดียว

อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แบบริงค์ (Ring) LAN จะถูกเชื่อมผ่านบริดจ์ 2 ด้านเพื่อให้มีการสื่อสารแบบวงกลม ฟังก์ชันควบคุมของบริดจ์จะง่ายลง ถ้าหากเป็นการสื่อสารแบบทางเดียวคือการไหลของข้อมูลจะผ่านเข้าทางด้านหนึ่งและหลังจากข้อมูลถูกตรวจสอบแล้วมันก็จะผ่านออกไปอีกทางหนึ่ง นอกจากนี้บริดจ์ยังเตรียมบัฟเฟอร์ทางเดียวเพื่อใช้ในการบริการแก่แพคเกจที่มาถึง อย่างไรก็ตามสำหรับการสื่อสารทางเดียวเช่นนี้จะเสี่ยงต่อการสูญเสียของแพคเกจถ้าหากบริดจ์หรือ LAN ใด ๆ เกิดหยุดทำงานลง

เมื่อพิจารณาถึงการสื่อสารแบบสองทางคือ แพคเกจสามารถไหลผ่านบริดจ์ทั้งสองด้านในกรณีเช่นนี้ถ้าหากอุปกรณ์ใด ๆ หยุดการทำงานระบบก็ยังสามารถทำงานต่อไปได้ เพียงแต่ฟังก์ชันของบริดจ์จะถูกเพิ่มขึ้นเป็น 2 เท่า เพื่อเทียบกับการสื่อสารทางเดียว และในทั้งสองกรณีนั้นการส่งผ่านแพคเกจไปยังปลายทางจำเป็นที่จะต้องผ่าน LAN และบริดจ์หลาย ๆ ตัวซึ่งจะเป็นสาเหตุของการล่าช้าในการส่งโดยเฉพาะเวลาที่สูญเสียให้กับบริดจ์ซึ่งเป็นเวลาสำหรับการรอการตรวจสอบ เวลาสำหรับการส่งใหม่เป็นต้น นอกจากนี้แล้วอัตราของแพคเกจที่สามารถส่งถึงปลายทาง (Throughput) นั้นจะขึ้นอยู่กับ LAN ที่มีความเร็วต่ำ และปรากฏว่าอัตราค่านี้นี้จะต่ำมาก

### 2.1.7 วิธีการส่งโดยใช้โทเคน (Token)

ANSI ( American National Standard Institute ) ได้เสนอวิธีการควบคุมแบบ Token Ring สำหรับการส่งด้วยความเร็ว 100 Mbps เรียกว่า FDDI ( Fiber Distribute Interface )

การทำงานพื้นฐานของวิธีนี้ก็คือ การส่งชุดข้อมูลของแต่ละสถานีจะถูกควบคุมโดยแพคเกจเล็ก ๆ ที่ถูกเรียกว่า Token โดย Token อันนี้จะหมุนไปรอบ ๆ Ring

ขั้นที่ 1 : ตรวจสอบ Token

สถานีที่เตรียมส่งแพคเกจจะขอการส่งจนกว่า Token จะผ่านมาถึงสถานี เมื่อสถานีนั้นๆ ตรวจสอบพบ Token มันจะปฏิบัติขั้นที่ 2

ขั้นที่ 2 : การส่งแพคเกจ

เมื่อสถานีได้นำเอา Token ออกจาก Ring แล้ว มันก็จะส่งแพคเกจเข้าไปใน Ring และแพคเกจนั้นจะถูกส่งโดยแต่ละสถานี จนกระทั่งถึงสถานีรับและแพคเกจนั้นจะถูกตัดออก จากนั้นแพคเกจจะถูกส่งต่อไปจนถึงสถานีส่ง

ขั้นที่ 3 : การนำเอาแพคเกจออก

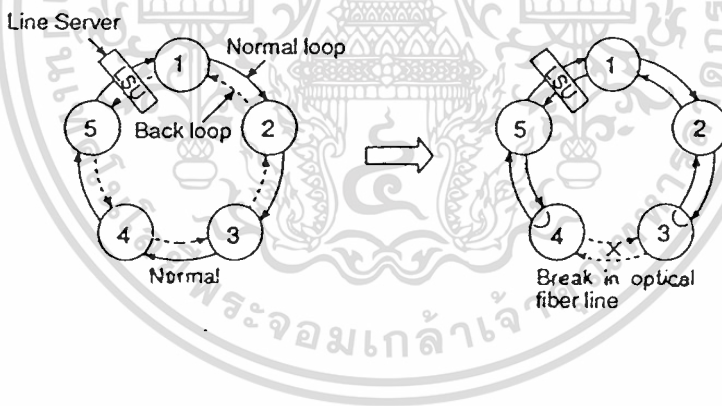
เมื่อแพคเกจผ่านมาถึงสถานีส่งแล้ว แพคเกจนั้นจะถูกนำออกจาก Ring สถานีส่งสามารถตรวจสอบได้ว่าเป็นแพคเกจที่ตัวเองส่งโดยการตรวจสอบ Address ของสถานีส่งหรืออาจ จะโดยการวัดเวลาหนึ่งรอบของสัญญาณ ผ่านรอบของ Ring แล้วนำแพคเกจที่มาถึงออก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### เส้นทางการสื่อสารแบบพิเศษ

เป็นวิธีการที่จะเพิ่มความน่าเชื่อถือในการรับส่งข้อมูลระหว่าง Node ในระบบโครงข่าย เมื่อมีความผิดพลาดเกิดขึ้น แบ่งเป็น 2 วิธี ดังนี้

**Back Loop** โดยทั่วไปการรับส่งข้อมูลภายในลูบ ต้องส่งข้อมูลไปในทิศทางเดียวกันโดยส่งผ่านในลักษณะ Dual – ring คือภายในสายของเส้นใยแสงจะประกอบด้วย 2 strands ในภาวะปกติจะส่งข้อมูลไปในทาง Primary Path คือจาก Normal Send Data(NSD) ไปยัง Normal Receive Data (NRD) แต่ถ้าสายเคเบิลมีการหักหรือมี Error Node จะมีการส่งข้อมูลไปในทิศทางสวนกับเส้นทางเดิมเรียก Secondary Path ซึ่งเป็นเส้นทางจาก Back Send Data(BSD) ไปยัง Back Receive Data (BRD) เพื่อช่วยป้องกันไม่ให้ระบบหยุดการทำงานเนื่องจากเส้นทางการสื่อสาร(Transmission Path) ตอนใดตอนหนึ่งถูกตัดขาด



รูปที่ 2.3 แสดงการสื่อสารแบบย้อนกลับ

**Node Bypass Function** เป็นวิธีการส่งข้อมูลอ้อม Error Node เพื่อป้องกันไม่ให้โครงข่ายทั้งหมดหยุดการทำงาน เมื่อมี Error Node อยู่ในโครงข่าย จะสามารถใช้การทำงานแบบนี้ได้เมื่อมี Local power supply ในแต่ละ Node

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 2.2 การสื่อสารแบบ Host Link

การสื่อสารแบบ Host Link เป็นการเชื่อมต่อ PLC กับ เครื่องคอมพิวเตอร์ทั่วไปผ่านทางพอร์ต ของเครื่องคอมพิวเตอร์(COM1:หรือCOM2:)ส่วนมากจะนิยมใช้มาตรฐานการส่งแบบอนุกรม เพื่อให้สามารถควบคุม PLC จากคอมพิวเตอร์ได้ สำหรับคอมพิวเตอร์ 1 เครื่องสามารถต่อเข้ากับ PLC ได้จำนวนมาก โดยใช้การเชื่อมต่อ PLC หลาย ๆ ตัวเข้าด้วยกัน เรียก PC Link ในการติดต่อแบบ Host Link จะต้องผ่านอุปกรณ์ที่เรียกว่า Host Link Unitsซึ่งจะต้องตั้งค่าต่างๆที่จำเป็นที่ใช้ในการติดต่อแบบ Host Link Units(C-500-LK203) ผ่านสวิทช์ของเครื่อง PLC

### 2.2.1 การติดต่อผ่านพอร์ตอนุกรม RS-232-C

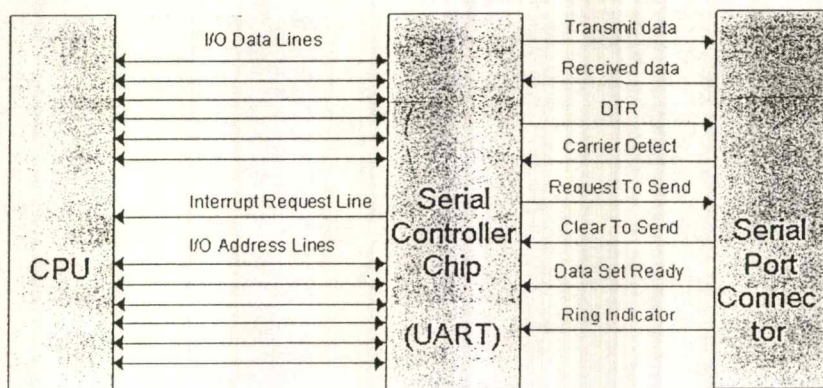
- พอร์ตอนุกรม (Serial Port)

เนื่องจากมาตรฐานการเชื่อมต่อแบบ RS-232-C มีใช้กันอย่างแพร่หลาย ซึ่งเป็นมาตรฐานที่ถูกกำหนดโดย EIA ซึ่งเป็นองค์การอุตสาหกรรมอิเล็กทรอนิกส์ของอเมริกาพอร์ตอนุกรม RS-232-C จะเป็นพอร์ตของเครื่องคอมพิวเตอร์ที่มีขาต่อ (Connector) ทั้งประเภท 9 ขา และ 25 ขา ซึ่งเราจะเรียกว่าพอร์ต COM1: และ COM2: นั่นเอง ในความเป็นจริงพอร์ตอนุกรมมิได้ถูกควบคุมโดยตรงจาก CPU บนเมนบอร์ด แต่การสื่อสารทั้งหมดจะถูกจัดการโดยชิป UART (Universal Asynchronous Receiver/Transmitter) อีกทีหนึ่ง ซึ่งในปัจจุบันนี้ เบอร์ที่ใช้กันมากที่สุดคือ 16550C

- การส่งข้อมูล (Data Transmission)

1. รับตัวอักษรจากเครื่องคอมพิวเตอร์
2. แปลงตัวอักษรให้เป็นสายข้อมูลแบบบิต (เราเรียกว่าขบวนการ Serialization)
3. สร้างเฟรมข้อมูลโดยการเพิ่มบิตที่จำเป็นสำหรับการสื่อสารและการตรวจสอบ เช่น บิต Start , Stop และ Parity เป็นต้น
4. ส่งผ่านเฟรมข้อมูลที่สร้างขึ้นมาแล้วจากขั้นตอนที่ผ่านมา ด้วยความเร็วของโมเด็มหรือพอร์ต อนุกรม (Baud Rate)
5. แสดงสถานะความพร้อมที่จะรับข้อมูลตัวอักษรถัดไปให้กับเครื่องคอมพิวเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.4 แสดงการเชื่อมต่อสัญญาณระหว่าง CPU ของเครื่องคอมพิวเตอร์กับพอร์ตอนุกรม

- การรับข้อมูล (Data Receiver)

1. รับตัวอักษรจาก Interface
2. ตรวจสอบความถูกต้องของเฟรมข้อมูลตามมาตรฐานที่กำหนด ถ้าเกิดข้อผิดพลาดขึ้นจะมีการแจ้งข้อผิดพลาดทันที
3. ตรวจสอบความถูกต้องของ PARITY
4. แปลงสายของมุกแบบบิตให้เป็นตัวอักษร
5. ส่งตัวอักษรให้กับเครื่องคอมพิวเตอร์แสดงสถานะความพร้อมที่จะรับตัวอักษรถัดไปให้กับ Interface

- **Baud Rate** คืออัตราความเร็วในการส่งถ่ายข้อมูลของโมเด็ม ในปัจจุบันโมเด็มได้รับการพัฒนาไปอย่างมาก อัตราความเร็วในการส่งถ่ายข้อมูล 1 บิตจะไม่เท่ากับอัตรา 1 Baud อีกต่อไป เช่นตามมาตรฐาน V.22 bis นั้นอัตรา 1Baud จะเท่ากับ 4 บิต ดังนั้นโมเด็มที่มีอัตราเร็ว 9600 Baud จะเท่ากับอัตราความเร็วในการส่งถ่ายข้อมูล 38,400 บิต/วินาที มาตรฐาน V.22 bis นั้นอัตรา 1Baud จะเท่ากับ 4 บิต ดังนั้นโมเด็มที่มีอัตราเร็ว 9600 Baud จะเท่ากับอัตราความเร็วในการส่งถ่ายข้อมูล 38,400 บิต/วินาที

- **Bit Per Second** คือ อัตราความเร็วในการส่งถ่ายจำนวนบิตของโมเด็มต่อ 1 วินาที
- **Through Put** คือจำนวนของบิตที่สามารถส่งผ่าน โมเด็มได้จริงๆ ภายในเวลา 1 วินาที
- **รูปแบบของเฟรมข้อมูล (Data Format)**

ในการสื่อสารระหว่าง Node สองตัวในระบบโครงข่ายจะต้องมีความเข้าใจถึงรูปแบบของเฟรมข้อมูลซึ่งกัน ซึ่งการส่งข้อมูลแบบ Asynchronous นั้น นอกจากข้อมูลที่ต้องส่งผ่านแล้วไม่วกรณใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ยังต้องจัดการช่วงจังหวะในการส่งหรือรับข้อมูลระหว่างกันอีกด้วย โดยที่เฟรมข้อมูลในการส่งหรือรับจะประกอบไปด้วยบิตข้อมูลที่มีความหมายดังต่อไปนี้

#### บิตข้อมูล (Data Bit)

หลังจากกระบวนการ Serialization ก็จะมีการส่งแต่ละบิตไปยังปลายทาง โดยเริ่มส่งจากบิตที่มีนัยสำคัญต่ำสุด (least signification bit) ไปยังตำแหน่งที่มีนัยสำคัญสูงสุด (most signification bit)

#### บิตเริ่มต้นข้อมูล (Start bit)

คือ บิตที่บอกให้ทราบถึงจุดเริ่มต้นของข้อมูลที่ต้องการส่ง ซึ่งในทุกๆ เฟรมจะต้องนำหน้าด้วยบิตเริ่มต้นข้อมูล (Start bit) เสมอ

#### บิตสิ้นสุดข้อมูล (Stop bit)

คือ บิตที่บอกถึงจุดสิ้นสุดของข้อมูลที่ต้องการส่ง

#### บิตพาริตี (Parity bit)

คือ บิตที่ใช้ในการตรวจสอบความถูกต้องในการรับส่งของข้อมูล ซึ่งโดยส่วนมากจะเพิ่มเข้าไปอีก 1 บิต จากบิตของข้อมูล 7 บิต

## 2.3 ชุดโพรโทคอลทีซีพี/ไอพี (TCP/IP Protocol)

โพรโทคอลทีซีพี/ไอพี (TCP/IP – Transmission Controller Protocol/Internet Protocol) เป็นกลุ่มโพรโทคอลที่พัฒนาขึ้นเพื่อให้คอมพิวเตอร์สามารถใช้ทรัพยากรและบริการฟังก์ชันพื้นฐานสำหรับการใช้งานบนระบบสื่อสารข้อมูลคอมพิวเตอร์ได้ ข้อมูลที่ใช้ทีซีพี/ไอพีนำส่งจะถูกแบ่งออกเป็นข้อมูลย่อยหลายๆ ส่วนเพื่อทยอยส่งไปตามลำดับเพื่อให้เหมาะสมกับระบบโครงข่ายในชั้นถัดไปที่อาจจะไม่สามารถส่งข้อมูลขนาดใหญ่ได้ทันที และเมื่อส่งไปถึงปลายทางก็จะรวบรวมข้อมูลนั้นกลับเป็นข้อมูลชุดเดิมอีกครั้ง

### 2.3.1 ชั้นต่างๆ ของทีซีพี/ไอพี (TCP/IP Layer)

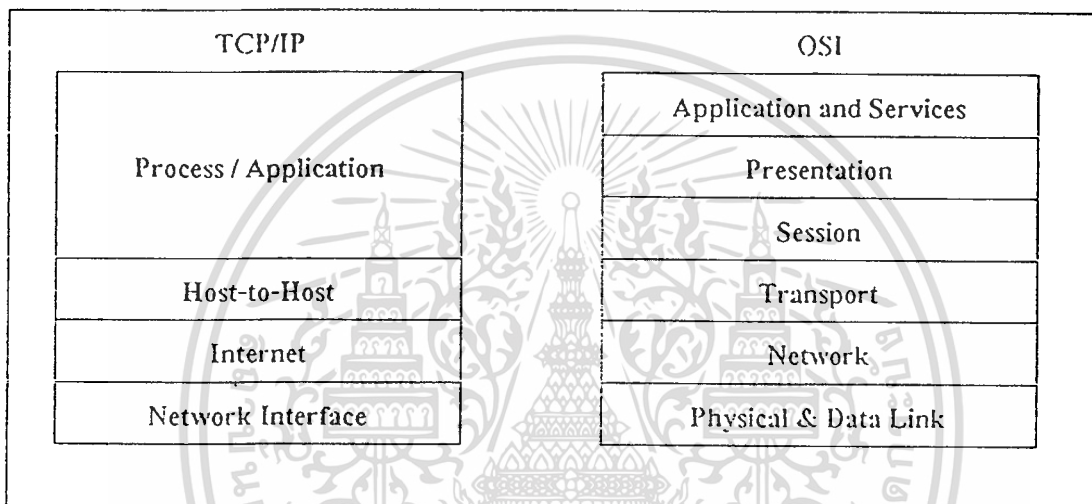
การติดต่อสื่อสารของทีซีพี/ไอพีถูกกำหนดให้มีการทำงานเป็นระดับชั้น(layer) เพื่อให้มีการทำงานเป็นอิสระต่อกันในแต่ละระดับชั้น และเพื่อให้มีขั้นตอนการทำงานในการแลกเปลี่ยนข้อมูลระหว่างคอมพิวเตอร์เป็นไปอย่างถูกต้องดังนี้

- กำหนดรูปแบบข้อมูล
- จัดเตรียมชุดข้อมูล
- กำหนดเส้นทางการส่งข้อมูล

เอกสารนี้เป็นเอกสารที่จัดทำขึ้นเพื่อใช้ในการเรียนการสอนวิชาเทคโนโลยีสารสนเทศศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ จะถือว่าผิดกฎหมายหากมีการนำข้อมูลไปใช้โดยไม่ได้รับอนุญาตจากเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- รวบรวมและจัดลำดับชุดข้อมูลที่ส่งมา
- ตรวจสอบว่ามีชุดข้อมูลซ้ำหรือไม่
- ตอบกลับไปให้ผู้ส่งรู้ว่าได้รับข้อมูลแล้ว
- ส่งผ่านข้อมูลไปให้ชั้นการทำงานถัดไป

เมื่อเปรียบเทียบกับ โมเดลอ้างอิงการเชื่อมต่อระบบเปิด(Open System Interconnection Reference Model : OSI-RM) ได้ดังนี้



ตารางที่ 2.1 การแบ่งระดับการทำงานของทีซีพี/ไอพีโดยอ้างอิงจามาตรฐาน OSI

### ● ชั้นเชื่อมต่อระบบโครงข่าย (Network Interface Layer)

ชั้นนี้จะทำหน้าที่ในการสื่อสารข้อมูลทางกายภาพในระดับสัญญาณนำส่ง(สัญญาณทางลอจิก “0” หรือ “1” )ตัวอย่างของระบบที่ทำงานในชั้นนี้ เช่น ระบบโครงข่ายอีเทอร์เน็ต หรือระบบโครงข่ายแบบ Token Ring โดยจัดข้อมูลเป็นกลุ่มที่เรียกว่าเฟรม(Frame)โดยในส่วนหัวของเฟรมจะบรรจุด้วยตำแหน่งต้นทางและปลายทาง ข่าวสารที่ใช้ในการควบคุม และส่วนท้ายที่ใช้ในการตรวจสอบข้อผิดพลาดติดต่อระดับต่ำสุดโดยผ่านระบบสื่อสัญญาณต่างๆไปถึงอุปกรณ์เชื่อมต่อปลายทาง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- **ชั้นอินเทอร์เน็ต (Internet Layer)**

ชั้นนี้จะมีอินเทอร์เน็ตโพรโทคอล(Internet Protocol) หรือ IP คอยทำหน้าที่ส่งข้อมูลที่เรียกว่า “ดาต้าแกรม”(Datagram) โดยในชั้นนี้IPจะทำงานโดยไม่มีการเชื่อมต่อก่อน(Connectionless) ดาต้าแกรมจะถูกจัดเส้นทางเป็นอิสระต่อกัน ไม่มีการรับประกันความถูกต้องความน่าเชื่อถือ หรือแม้แต่การจัดเรียงลำดับดาต้าแกรมให้อยู่ในลำดับที่ถูกต้อง

ข้อมูลจะส่งเข้าไปในระบบโครงข่ายโดยมี Router เป็นตัวจัดเส้นทาง กว่าข้อมูลจะไปถึงปลายทางอาจต้องผ่านตัวจัดเส้นทาง(Router)ของหลายโครงข่ายจึงต้องมีการผนวกหมายเลขเครื่องต้นทางและปลายทางเข้าไปในชุดข้อมูลเพื่อให้Routerรู้ข้อมูลที่ผ่านมาต้องการจะไปไหนถ้าไม่ใช่หมายเลขโครงข่ายของตัวเองก็จะส่งต่อไปยังโครงข่ายอื่น แต่ถ้าใช่ก็จะส่งไปให้สมาชิกทั้งหมดของเครือข่าย

- **ชั้นโฮสต์ทูโฮสต์ (Host-to-Host Layer) –TCP และ UDP**

โพรโทคอลที่ทำงานในชั้นนี้มี 2 แบบ

- แบบที่เรียกว่า TCP จะทำงานแบบมีการเชื่อมต่อก่อน(Connection Orient) มีการส่งข้อมูลโดยรับประกันความน่าเชื่อถือและเรียงลำดับข้อมูลที่ถูกต้อง โพรโทคอล TCP จะทำการเพิ่มส่วนหัวเพื่อสร้างเป็นเช็กเมนต์
- แบบที่เรียกว่า UDP (Users Datagram Protocol) ซึ่งการทำงานไม่มีการเชื่อมต่อก่อน (Connectionless) และไม่มีการรับประกันความถูกต้องของข้อมูล

- **ชั้นโปรแกรมประยุกต์ (Application Layer)**

ชุดโพรโทคอลที่ซีพี/ไอพี จะมีโพรโทคอลในชั้นโปรแกรมประยุกต์ให้ใช้งานอยู่มาก ที่นิยมใช้กันมากและจัดเป็นบริการพื้นฐานของที่ซีพี/ไอพี เช่น การล็อกอินระยะไกล(Remote Login) การส่งผ่านแฟ้มข้อมูล(Files Transfer) การส่งไปรษณีย์อิเล็กทรอนิกส์(Electronic Mail) บริการเวิลด์ไวด์เว็บ(World Wide Web) เน็ตเวิร์คไฟล์ซิสเต็ม(NFS-Network File System) การพิมพ์ระยะไกล(Remote Printing) เป็นต้น

## 2.4 โพรโทคอลในชั้นอินเทอร์เน็ต

ที่ซีพี/ไอพีเป็นโพรโทคอลที่ใช้สื่อสารข้อมูลกันในระบบโครงข่ายคอมพิวเตอร์ โดยที่คอมพิวเตอร์แต่ละตัวจะส่งข้อมูลเข้าไปในระบบโครงข่าย จากนั้นจะใช้อุปกรณ์เชื่อมต่อระบบโครงข่ายที่เรียกว่าเราเตอร์ (router) ในการสื่อสารข้อมูลเข้าหากัน ซึ่งจะอาศัยการทำงานในชั้นอินเทอร์เน็ตของชุดโพรโทคอลที่ซีพี/ไอพี เนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

### 2.4.1 อินเทอร์เน็ตโปรโตคอล (Internet Protocol)

อินเทอร์เน็ตโปรโตคอลหรือไอพี เป็นโปรโตคอลหลักที่ทำหน้าที่ส่งข้อมูลในชั้นอินเทอร์เน็ตนี้ ข้อมูลในส่วนของการทำงานในชั้นนี้จะเรียกว่า คาต้าแกรม (datagram) ชั้นนี้ไม่มีการรับประกันว่าข้อมูลที่ผ่านจากไอพีนี้ขึ้นไปนั้นจะถูกดอง อาจมีการสูญหาย การซ้ำซ้อนของข้อมูล การไม่เรียงลำดับของข้อมูล ซึ่งจะเป็นหน้าที่ของโปรโตคอลชั้นบนต่อไปเป็นผู้จัดการ ไอพีจะไม่มี การตรวจสอบข้อผิดพลาดของข้อมูล จะมีเฉพาะผลรวมตรวจสอบส่วนหัวเท่านั้น ไม่มีการส่งซ้ำ (re-transmission) ไม่มีการควบคุมการรับส่ง (flow control)

ไอพีมีหน้าที่หลักคือรับข้อมูลจากที่ซึ่พีหรือยูดีพีแล้วสร้างเป็นคาต้าแกรม จากนั้นจึงทำการ ค้นหาเส้นทางที่จะให้นำส่งคาต้าแกรมตัวนี้ไปยังแอดเดรสปลายทางต่อไป ซึ่งคาต้าแกรมแต่ละตัว จะถูกจัดเส้นทางเป็นอิสระต่อกัน ในการหาเส้นทางนั้นไอพีจำเป็นจะต้องมีความสามารถในการ แลกเปลี่ยนข่าวสารที่ใช้กับโปรโตคอลตัวอื่นด้วย เช่น อาร์ไอพี (RIP : Routing Information Protocol) จีจีพี (GGP : Gateway-Gateway Protocol) อีจีพี (EGP : External Gateway Protocol) เป็นต้น

การส่งข้อมูลของไอพีจำเป็นจะต้องบอกอินเทอร์เน็ตแอดเดรสของเครื่องปลายทาง งานหลักของไอพีคือ ค้นหาเส้นทางเพื่อที่จะส่งคาต้าแกรมไปยังปลายทาง ในการที่จะให้อุปกรณ์ ระหว่างทาง (Intermedia System) ส่งคาต้าแกรมจะต้องมีข้อมูลในส่วนหัวของคาต้าแกรมเป็นตัวอย่าง ในส่วนนี้จะบอกถึงอินเทอร์เน็ตแอดเดรสต้นทางและปลายทางขนาด 32 บิต โปรโตคอลนามเบอร์ (Protocol Number) และผลรวมตรวจสอบตำแหน่งต้นทางและปลายทางใช้เพื่อบอกให้ทั้ง 2 ฝ่ายทราบว่าข้อมูลจากที่ใดและจะไปไปที่ใด โปรโตคอลนามเบอร์ใช้บอกไอพีว่าจะส่งคาต้าแกรมไปยัง ชั้นที่ซึ่พีหรืออื่น ๆ ผลรวมตรวจสอบใช้เพื่อตรวจสอบว่าข้อมูลของส่วนหัวไม่มีข้อผิดพลาด ระหว่างการส่ง แฟล็ก (flag) และแฟร็กเมนต์ออฟเซต (fragment offset) ใช้เพื่อแบ่งคาต้าแกรมเป็น ขนาดเล็ก ๆ เนื่องจากอาจจะส่งข้อมูลขนาดใหญ่เกินกว่าที่จะให้ระบบโครงข่ายนั้นส่งได้ทั้งหมด ภายในการส่งครั้งเดียว เวลาคงอยู่เป็นตัวเลขที่ใช้บอกถึงช่วงเวลาทีคาต้าแกรมตัวนี้จะคงอยู่ใน ระบบโครงข่ายได้ ค่านี้อาจลดลงเรื่อย ๆ เมื่อมีการประมวลผลคาต้าแกรมในแต่ละระบบ เมื่อค่านี เป็น "0" คาต้าแกรมนี้ก็จะถูกกำจัดทิ้งไป เพื่อป้องกันการส่งข้อมูลวนในเน็ตเวิร์ค

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Version	IHL	Type of Service	Total Length	
Identification		Flags	Fragment Offset	
Time to Live	Protocol	Header Checksum		
Source Address				
Destination Address				
Options			Padding	

## ตารางที่ 2.2 แสดงรูปแบบส่วนหัวอินเทอร์เน็ตโพรโทคอล (Internet Protocol Header Format)

- เวอร์ชันขนาด 4 บิต

เป็นเลขรุ่นของโพรโทคอลไอพีที่ใช้กันอยู่ในปัจจุบันเป็นรุ่นที่ 4

- ความยาวส่วนหัว (IHL-Internet Header Length) ขนาด 4 บิต

ใช้เก็บความยาวส่วนหัวในรูปของ 32 บิตเวิร์ด ค่าต่ำสุดที่มีได้เท่ากับ 5 นั่นคือส่วนหัวของคำสั่งแกรมหนึ่งๆต้องมีความยาวอย่างน้อย 160 บิต ถ้าคำสั่งแกรมนี้มีข้อมูลส่วน option เพิ่มเติมด้วย จะต้องทำการเติม "0" ไปเพื่อให้ได้ความยาวส่วนหัวเป็นจำนวนเท่าของ 32 บิตเวิร์ด

Precedence c] และชนิดของการบริการ (Type of Service) ขนาด 8 บิต

ใช้บ่งบอกคุณภาพของข่าวสารบริการว่าจะประมวลผลคำสั่งแกรมตัวนี้อย่างไร

ตำแหน่งบิต	ความหมาย
บิต 0-2	Precedence Level 0-7, 0 is normal and level 7 is highest priority
บิต 3	0=Normal Delay, 1=Low Delay
บิต 4	0=Normal Throughput, 1=High Throughput
บิต 5	0=Normal Reliability, 1=High Reliability
บิต 6-7	สำรองไว้ใช้ในอนาคต จะมีค่าเป็น 0

## ตารางที่ 2.3 แสดงรูปแบบชนิดของการบริการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ความยาวทั้งหมด (Total Length) ขนาด 16 บิต  
เป็นความยาวทั้งหมดของดาต้าแกรม ซึ่งนับรวมทั้งส่วนหัวและส่วนของข้อมูลในหน่วย  
อ็อกเตท ซึ่งจะมีค่าได้สูงสุด 65535 อ็อกเตท ในข้อกำหนดของไอพี โสตต์ในระบบโครงข่ายต้อง  
สามารถรองรับดาต้าแกรมที่มีความยาวได้ต่ำสุด 576 อ็อกเตท
- ตัวบ่งชี้ (Identification) ขนาด 16 บิต  
กำหนดโดยผู้ส่ง ใช้สำหรับการแยกดาต้าแกรมออกเป็นดาต้าแกรมย่อย ๆ
- แฟล็ก (Flag) ขนาด 3 บิต

ตำแหน่งบิต	ความหมาย
บิต 0	Reserved, must be zero
บิต 1	(DF) 0=May Fragment, 1=Don't Fragment
บิต 2	(DF) 0=Last Fragment, 1=More Fragment

#### ตารางที่ 2.4 แสดงรูปแบบแฟล็กของอินเทอร์เน็ตโพรโตคอล

- แฟร็กเมนต์ออฟเซต (Fragment Offset) ขนาด 13 บิต  
ใช้ชี้ตำแหน่งของแฟร็กเมนต์ (fragment) ในดาต้าแกรมเดิมก่อนทำการแฟร็กเมนต์ ค่า  
ของแฟร็กเมนต์ออฟเซตจะอยู่ในหน่วยของ 8 อ็อกเตท
- เวลาคงอยู่ (TTL – Time to Live) ขนาด 8 บิต  
ใช้แสดงถึงระยะเวลาสูงสุดที่ดาต้าแกรมสามารถอยู่ในระบบอินเทอร์เน็ตได้เพื่อป้องกันไม่  
ให้มีดาต้าแกรมไหลวนอยู่ในระบบโครงข่ายที่มีการผิดพลาดอย่างไม่รู้จบ โสตต์ที่ตั้งจะกำหนดค่านี้  
และเราเตอร์ที่นำดาต้าแกรมไปทำการประมวลผลจะลดค่า TTL ลง 1 ค่า ทุกครั้งที่ดาต้าแกรมถูก  
ประมวลผล ซึ่งถ้าฟิลด์นี้มีค่าเป็น “0” ดาต้าแกรมตัวนี้จะถูกทำลายทิ้งโดยอัตโนมัติ ทั้งนี้แล้วแต่ด้วย  
ว่าใช้โพรโตคอลในการหาเส้นทางแบบใด
- โพรโตคอล (Protocol) ขนาด 8 บิต  
บอกถึงโพรโตคอลในเลขอร์ถัดไปที่จะนำข้อมูลในดาต้าแกรมตัวนี้ไปใช้งาน เช่น “1” คือ  
โพรโตคอลไอซีเอ็มพี “6” คือ โพรโตคอลทีซีพี เป็นต้น

- ผลรวมตรวจสอบส่วนหัว (Header Checksum) ขนาด 16 บิตจะมีการคำนวณใหม่  
ทุกครั้งถ้ามีบางฟิลด์ของส่วนหัวเปลี่ยนค่าไป เช่นค่า TTL  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ที่อยู่ต้น (Source Address) ขนาด 32 บิตเป็นหมายเลขอินเทอร์เน็ตแอดเดรสของเครื่องที่ส่งคำสั่งแกรมตัวนี้ไป
- ที่อยู่ปลายทาง (Destination Address) ขนาด 32 บิตเป็นหมายเลขอินเทอร์เน็ตแอดเดรสของเครื่องปลายทางที่ส่งคำสั่งแกรมตัวนี้ไป
- ออปชัน (Options) ขนาดเปลี่ยนแปลงได้ ออปชันจะใช้เป็นตัวกำหนดถึงข้อมูลที่มาในคำสั่งแกรม ซึ่งอาจจะไม่มีหรือมีได้หลาย ๆ ออปชันก็ได้ สามารถยาวได้ถึง 40 อ็อกเตท แต่ปกติจะไม่มีการใช้งานในฟิลด์นี้
- แพดดิ้ง (Padding) ขนาดเปลี่ยนแปลงได้ใช้เพิ่ม "0" เข้าไปเพื่อทำให้ส่วนหัวของอินเทอร์เน็ตแอดเดรสมีความยาวเป็นจำนวนเท่าของ 32 บิต

#### 2.4.2 โพรโตคอลในชั้นอินเทอร์เน็ต

ทีซีพี/ไอพีเป็นโพรโตคอลที่ใช้สื่อสารข้อมูลกันในระบบโครงข่ายคอมพิวเตอร์ โดยที่คอมพิวเตอร์แต่ละตัวจะส่งข้อมูลเข้าไปในระบบโครงข่าย จากนั้นจะใช้อุปกรณ์เชื่อมต่อระบบโครงข่ายที่เรียกว่าเราท์เตอร์ (router) ในการสื่อสารข้อมูลเข้าหากัน ซึ่งจะอาศัยการทำงานในชั้นอินเทอร์เน็ตของชุดโพรโตคอลทีซีพี/ไอพี

อินเทอร์เน็ตแอดเดรสเป็นตัวเลขขนาด 32 บิต ใช้บ่งบอกเลขประจำตัวของโฮสต์นั้น ๆ เพื่อความสะดวกในการอ่าน จึงมักจะเขียนอินเทอร์เน็ตแอดเดรสเป็นเลขฐานสิบจำนวน 4 ชุด แต่ละชุดจะแบ่งด้วยเครื่องหมายจุด (dot) เรียกกันว่า dotted decimal notation ค่า 32 บิตนี้จะถูกแบ่งออกเป็นค่า 8 บิตต่อหนึ่งฟิลด์ เรียกกันว่า อ็อกเตท (octet) ซึ่งจะมีจำนวนทั้งหมด 4 ฟิลด์ และกำหนดค่าในแต่ละฟิลด์ด้วยค่าเลขฐานสิบ เช่น อินเทอร์เน็ตแอดเดรสของระบบหนึ่ง ๆ เป็น 10100001 1111 0110 0000 1010 0001 0101 จะได้ค่าในแต่ละอ็อกเตทคือ 161 246 10 21 ซึ่งเมื่อเขียนอยู่ในรูป dotted decimal notation จะได้เป็น 161.246.10.21

อินเทอร์เน็ตแอดเดรสขนาด 32 บิตนี้จะถูกมองเป็น 2 ส่วน ส่วนแรกเป็นหมายเลขเน็ตเวิร์ค (Network Number) ซึ่งเป็นโครงข่ายที่โฮสต์นั้นเชื่อมต่ออยู่ และส่วนที่เหลือเป็นหมายเลขท้องถิ่น (Local Host Address) หรือหมายเลขโฮสต์ของตัวเอง (Host Address)

0

31

Network Address	Local Host Address
-----------------	--------------------

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ 2.5 แสดงรูปแบบของอินเทอร์เน็ตแอดเดรสให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ซึ่งจากอินเทอร์เน็ตแอดเดรสขนาด 32 บิตนี้ จะแบ่งระบบโครงข่ายออกเป็น 5 คลาสใหญ่ดังนี้

1. ระบบโครงข่ายในClassA
2. ระบบโครงข่ายในClassB
3. ระบบโครงข่ายในClassC
4. ระบบโครงข่ายในClassD
5. ระบบโครงข่ายในClassE

เนื่องจากการคิดต่อแบบโครงข่ายของโครงข่ายนี้ใช้ IP Address แบบคลาสA

ดังนั้นจึงขอกว่าตั้งแต่ IP Address แบบคลาสA

- ระบบโครงข่ายในClassA

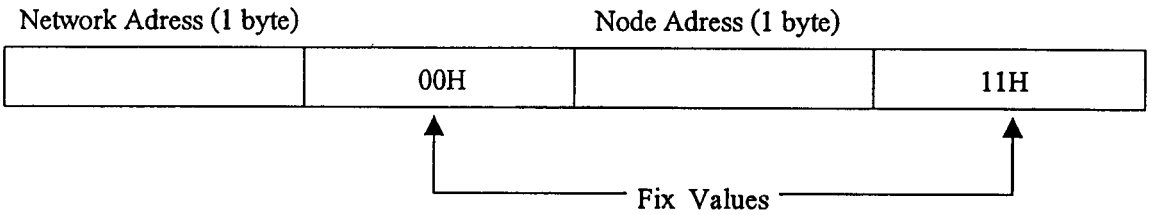
0 1	7 8	31
0	Network Address	Host Address

รูปที่ 2.6 แสดงรูปแบบของอินเทอร์เน็ตแอดเดรสClassA

ClassA จะใช้ 8 บิตแรกในการกำหนดหมายเลขเน็ตเวิร์ค และอีก 24 บิตที่เหลือในการกำหนดหมายเลขโฮสต์ โดยที่บิตแรกของหมายเลขเน็ตเวิร์คต้องเป็น 0 ทำให้สามารถกำหนดระบบโครงข่ายในClassA ได้  $2^7$  ระบบโครงข่าย ซึ่งในการกำหนดอินเทอร์เน็ตแอดเดรสจะมีข้อกำหนดบางอย่างว่า ห้ามมีระบบโครงข่ายที่หมายเลขโครงข่ายเป็น “0” ตลอด หรือเป็น “1” ตลอด ทำให้สามารถกำหนดโครงข่ายในClassA ได้ 126 โครงข่าย (ระบบโครงข่ายหมายเลข 0 และ 127 จะถูกสงวนไว้) การกำหนดหมายเลขโฮสต์ให้มีขนาด 24 บิตทำให้ระบบโครงข่ายในClassA แต่ละระบบนี้มีโฮสต์ได้สูงสุด  $2^{24}$  เครื่องด้วย ซึ่งก็จะมีข้อกำหนดเช่นเดียวกับการกำหนดหมายเลขโครงข่ายทำให้มีเครื่องได้สูงสุด 16777214 เครื่อง

- SYSMAC NET Address Format

คือ IP Address โดยใช้อ้างอิงจาก IP Address มาตรฐาน ClassA และเป็นข้อจำกัดในจำนวนของ Node ใน 1 โครงข่าย และจำนวนโครงข่ายทั้งหมดที่สามารถคิดตั้งได้



**รูปที่ 2.7 แสดงรูปแบบ IP Address Class A**

### 2.4.2 อินเทอร์เน็ตแอดเดรสสงวน

นอกจากนี้ยังมีข้อกำหนดบางอย่างของการกำหนดไอพีแอดเดรสดังนี้

บรอดคาสต์แอดเดรส (Broadcast Address) บิตที่ถูกกำหนดเป็นหมายเลขท้องถิ่นนั้นห้ามกำหนดทุกบิตในแอดเดรสเป็น “1” ทั้งหมด เนื่องจากในการจัดตั้งอินเทอร์เน็ตแอดเดรสนั้น ถ้าทุกบิตของหมายเลขท้องถิ่นเป็น “1” หมด จะหมายถึงสถานีงานทุกเครื่องที่อยู่บนระบบโครงข่ายนี้ เช่น 161.246.255.255 จะเป็นบรอดคาสต์แอดเดรสของระบบโครงข่าย Class B หมายเลข 161.246.0.0

ลูปแบ็คแอดเดรส (Loopback Address) อินเทอร์เน็ตแอดเดรสในโครงข่าย Class A เบอร์ 127.0.0.0 จะถูกกำหนดไว้ทำเป็น “loopback” เพื่อใช้ในการทดสอบและติดต่อสื่อสารระหว่างโปรแกรมบนเครื่องนั้น ๆ เอง หมายความว่าเมื่อมีโปรแกรมต้องการส่งข้อมูลไปยัง loopback address ส่วนโปรโตคอลจะส่งข้อมูลนั้นกลับโดยไม่มีการส่งข้อมูลไปยังระบบโครงข่ายใด ๆ เนื่องจากเป็นการส่งข้อมูลไปยังแอดเดรสที่หมายถึงตัวมันเอง

## 2.5 ทฤษฎีการทำงานของโปรแกรม Microsoft Visual Basic

Visual Basic 5.0 เป็นโปรแกรมที่สนับสนุนคอนโทรลที่ทำหน้าที่ติดต่อกับอินเทอร์เน็ต โดยอาศัยโปรโตคอล TCP/IP โดยอาศัยการสื่อสาร ผ่านพอร์ตอนุกรม (Serial Port) โดยอาศัยคอลโทรล (MSComm) ผ่านพอร์ตของคอมพิวเตอร์ที่วินโดวส์รู้จักในชื่อของ COM1: , COM2: ซึ่ง MSComm สามารถใช้ในการติดต่อหรือควบคุมบอร์ดต่างๆ หรืออุปกรณ์อิเล็กทรอนิกส์ที่มีใช้ในโรงงานอุตสาหกรรมโดยทั่วไป นอกจากนี้ยังสามารถใช้สื่อสารกับเครื่องคอมพิวเตอร์อื่นๆ ในระบบโครงข่าย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 2.5.1 รายละเอียดของส่วนประกอบต่างๆ ใน Visual Basic

- เมนูบาร์ (Menu Bar)

เป็นคำสั่งที่เราสามารถใช้งานได้ทั้งหมดใน VB

- ทูลบาร์ (Toolbar)

เป็นส่วนที่รวบรวมเอาไอคอนเล็กๆ มากมายเอาไว้ ซึ่งแต่ละไอคอน (Icon) เหล่านี้ก็เปรียบเสมือนคำสั่งต่างๆ ของเมนู นั่นคือไอคอนเหล่านี้ได้ถูกออกแบบมาเพื่อให้การใช้งานคำสั่งต่างๆ ทำได้รวดเร็วและมีลักษณะที่สื่อความหมายกับผู้ใช้มากขึ้น

- ทูลบ็อกซ์ (Toolbox)

เป็นแถบเครื่องมือที่ประกอบไปด้วยไอคอนต่างๆ ซึ่งเรียกว่า “คอนโทรล” (Control) โดยจะนำมาใช้ร่วมกับฟอร์มเพื่อสร้างจอภาพของโปรแกรม แต่แต่ละคอนโทรลจะใช้เป็นเครื่องมือในการสร้างส่วนติดต่อกับผู้ใช้ (User Interface) โดยการคลิกเลือกที่คอนโทรลนั้นแล้วนำมาวางลงบนฟอร์มด้วยการลากแล้ววาง (Drag and Drop) หรือ โดยการดับเบิลคลิกที่ไอคอน ในทูลบ็อกซ์ก็ได้

#### หน้าต่างโปรเจกต์ (Project Explorer Window)

เป็นหน้าต่างที่รวบรวมรายชื่อของฟอร์มโมดูลไฟล์ (Custom Control) โมดูล Class (Class Module) หรือไฟล์ทรัพยากร (Resource File) สำหรับการสร้างแอปพลิเคชันต่างๆ ซึ่งการรวมเอาไฟล์เหล่านี้เข้าด้วยกันเพื่อสร้าง แอปพลิเคชันภายใต้ VB เรียกว่า โปรเจกต์ (Project)

#### หน้าต่างคุณสมบัติ (Properties Window)

เป็นหน้าต่างที่รวบรวมคุณสมบัติทั้งหมดของฟอร์มหรือคอนโทรลเอาไว้ ซึ่งคุณสมบัติทั้งหมดที่ปรากฏในหน้าต่างนี้จะเป็นคุณสมบัติที่ผู้อ่านสามารถกำหนดค่าได้ในขณะกำลังออกแบบ เมื่อผู้อ่านทำการแก้ไขค่าคุณสมบัติต่างๆ ในหน้าต่างคุณสมบัตินี้ก็จะส่งผลต่อคอนโทรลตัวนั้นทันที ซึ่งบางคุณสมบัติสามารถแสดงให้เห็นการเปลี่ยนแปลงคุณลักษณะได้ทันที ส่วนคุณสมบัติบางอย่างจะแสดงผลให้เห็นก็ต่อเมื่อผู้อ่านมีการรันแอปพลิเคชันเท่านั้น

#### ฟอร์ม (Form)

เป็นหน้าต่างที่ใช้ในการออกแบบส่วนติดต่อกับผู้ใช้ของแอปพลิเคชันของเรา เนื่องจาก VB ได้รับการออกแบบให้ผู้นำนามาท่ออกแบบแอปพลิเคชันในลักษณะของการสื่อสารด้วยรูป (Visual) ดังนั้นการสื่อสารการทำงานต่างๆ ระหว่างแอปพลิเคชันกับผู้ใช้จะต้องกระทำผ่านฟอร์ม โดยในแอปพลิเคชันหนึ่งสามารถมีได้หลายฟอร์ม และภายในฟอร์มก็จะถูกใช้ในการบรรจุคอนโทรลต่างๆ ฟอร์มจึงทำหน้าที่เป็นตัวบรรจุ (Container) และฟอร์มยังเป็นออบเจกต์ (Object) ตัวหนึ่งของ VB

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ที่อนุญาตให้ ผู้ใช้แก้ไขคุณสมบัติได้ในขณะออกแบบจากหน้าต่างคุณสมบัติ และสามารถควบคุมพฤติกรรมต่างๆ ของฟอร์มด้วยวิธีเดียวกับออบเจกต์อื่นๆ

หน้าต่างโค้ดอิดิเตอร์ ( Code Editor Window )

เป็นหน้าต่างที่ใช้ในการพิมพ์คำสั่งโปรแกรมเข้าไปในแอฟพลิเคชันของเรา

หน้าต่างฟอร์มเลย์เอาท์ ( Form Layout Window )

เป็นหน้าต่างแสดงตำแหน่งฟอร์มบนจอ ช่วยให้เรากำหนดตำแหน่งของฟอร์มบนหน้าจอได้สะดวกขึ้น โดยการคลิกเมาส์ ลากฟอร์มที่แสดงบนหน้าต่างให้เคลื่อนไปยังตำแหน่งที่ต้องการ

- ฟอร์ม ( Form )

ฟอร์มเป็นส่วนประกอบพื้นฐานในการติดต่อกับผู้ใช้ ที่ใช้แสดงผลการทำงาน รวมทั้งรับอินพุตต่างๆ จากผู้ใช้

- คอนโทรลของ VB

คอนโทรลเป็นเครื่องมืออย่างหนึ่งซึ่งช่วยให้การสร้างแอฟพลิเคชันด้วย VB ทำได้ง่ายและรวดเร็วยิ่งขึ้น แต่เมื่อได้นำไปวางบนฟอร์มแล้ว จะเรียกว่าออบเจกต์ ( Object ) ซึ่งจริงๆ แล้วคอนโทรลก็คือไฟล์ไดนามิกลิงค์ไลบรารีต่างๆ ไปของวินโดว ไฟล์เหล่านี้จะมีนามสกุล .VBX หรือ .OCX ซึ่งเราเรียกว่า Custom Control ซึ่งไฟล์เหล่านี้จะถูกจัดเก็บในไคลเรคทอรีย่อย System ของ Window

คอนโทรลพื้นฐานใน VB จะเป็นคอนโทรลติดต่อกับผู้ใช้ในการทำงานพื้นฐานต่างๆ เช่น การแสดงข้อความและรับข้อความจากผู้ใช้ การแสดงรูปภาพ การแสดงตัวเลือก เป็นต้น ซึ่งจะกล่าวถึงเฉพาะคอนโทรลที่ได้นำมาใช้งานดังต่อไปนี้

เท็กซ์บ็อกซ์ ( Text Box )

เป็นคอนโทรลที่ผู้ใช้สามารถเติม แก้ไขข้อความในเท็กซ์บ็อกซ์นี้ได้โดยผ่านคีย์บอร์ดเข้าสู่โปรแกรม หรือแสดงข้อมูลต่างๆ ในโปรแกรมของเราได้

- เลเบล ( Label )

เป็นคอนโทรลที่ใช้ในการแสดงข้อความตามที่เรต้องการ เพื่อแสดงข้อมูลบางอย่างแก่ผู้ใช้ โดยผู้ใช้ไม่สามารถแก้ไขข้อความนั้นได้ในตอนรันโปรแกรม การแก้ไขข้อความในเลเบลจะต้องทำในขณะที่ออกแบบ หรือในคำสั่งโปรแกรมตอนรันเท่านั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.8 แสดง Label

- ปุ่มคำสั่ง (Command Button)

เป็นคอนโทรล ที่ใช้ในการทำงานอย่างหนึ่งตามที่เราต้องการ โดยเราสามารถกำหนดสิ่งที่เราต้องการให้ทำงานในอีเวนต์ Click คอนโทรลนี้จะอยู่ในทูลบ็อกซ์ ดังรูป การกำหนดข้อความที่ให้แสดงบนคอนโทรล

ในการกำหนดข้อความที่ต้องการให้แสดงผลบนคอนโทรล ให้เรากำหนดที่คุณสมบัติ Caption ของคอนโทรลในหน้าต่างคุณสมบัติ หรือในคำสั่งโปรแกรมก็ได้



รูปที่ 2.9 แสดง Command Button

- เฟรม (Frame)

เป็นคอนโทรลที่ใช้ในการจัดกลุ่มคอนโทรลที่ต้องใช้ร่วมกัน ทำงานอย่างเดียวกันไว้ด้วยกัน ซึ่งโดยทั่วไปคอนโทรลเฟรมเป็นคอนโทรลที่ช่วยเพิ่มความเรียบร้อย และความสวยงามของโปรแกรม



รูปที่ 2.10 แสดง Frame

- คอมโบบ็อกซ์ (ComboBox)

จะเป็นคอนโทรลที่ใช้ในการแสดงรายการที่ต้องการให้ผู้ใช้เลือกรายการ หรือสามารถแก้ไขรายการที่เลือกได้ คอนโทรลนี้จะใช้พื้นที่ในการวางคอนโทรลน้อยกว่าลิสต์บ็อกซ์ คอนโทรลนี้จะไม่เหมือนกับคอนโทรลเท็กซ์บ็อกซ์ และลิสต์บ็อกซ์รวมกัน



รูปที่ 2.11 แสดง ComboBox

- **พิกเจอร์บ็อกซ์ (Image and Picture Box)**

พิกเจอร์บ็อกซ์คอนโทรล ทำให้สามารถวางสารสนเทศทางกราฟฟิกไว้ในตำแหน่งเฉพาะบนฟอร์ม พิกเจอร์บ็อกซ์จะมีความยืดหยุ่นมากกว่า และด้วยเหตุนี้จึงต้องการหน่วยความจำและหน่วยประมวลผลในปริมาณที่มากกว่า พิกเจอร์บ็อกซ์เหมาะอย่างยิ่งสำหรับสภาพแวดล้อมไดนามิก เช่น การวาดกราฟฟิกเข้าไปที่จอภาพโดยตรงในขณะที่โปรแกรมกำลังทำงาน หรือการทำไอคอนให้เคลื่อนไหวได้ ด้วยการเคลื่อนย้ายไอคอนไปรอบๆจอภาพ อิมเมจคอนโทรลเหมาะสำหรับสถานการณ์สถิตินั้นคือ ไม่มีการเปลี่ยนแปลงภาพที่สร้างหรือวางเอาไว้บนจอ



รูปที่ 2.12 แสดง Image and Picture Box

- **คอนโทรลลิสต์บ็อก (ListBox)**

คอนโทรลลิสต์บ็อก (ListBox) เป็นคอนโทรลที่มีลักษณะการใช้งานในรูปแบบการแสดงข้อความหลายๆบรรทัดภายในคอนโทรล ListBox และถ้ามีข้อความจำนวนมากจนไม่สามารถแสดงผลได้ในหน้าจอเดียว Visual Basic ก็จะเพิ่มแถบเลื่อน (Scroll Bar) ให้กับคอนโทรลโดยอัตโนมัติโดยแถบเลื่อนนี้สามารถเลื่อนขึ้นได้ทั้งแนวตั้งและแนวนอนทั้งนี้ขึ้นอยู่กับวิธีการแสดงผลของ ListBox



รูปที่ 2.13 แสดง ListBox

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- คอนโทรล (OptionButton)

คอนโทรล (OptionButton) เป็นคอนโทรลที่ถูกออกแบบมาให้ใช้ในลักษณะของการเลือกอย่างใดอย่างหนึ่งเท่านั้นในกลุ่ม OptionButton ที่อยู่บน Frame, Form หรือ PictureBox เดียวกัน เป็นต้น



รูปที่ 2.14 แสดง OptionButton

- คอนโทรล MSComm

คอนโทรล MSComm (Communication) เป็นคอนโทรลตัวหนึ่งที่ทำหน้าที่เสมือนช่วยติดต่อกับพอร์ตอนุกรม(Serial Port)โดยใช้ฟังก์ชันในวินโดวส์ API ซึ่งวินโดวส์จะทำหน้าที่โดยตรงในการส่งหรือรับข้อมูลผ่านทางพอร์ตอนุกรมโดยอาศัยไดรเวอร์ Comm.drv อีกทอดหนึ่งข้อมูลที่ถูกจัดในรูปแบบมาตรฐานการสื่อสารแล้วก็สามารถทำการรับ-ส่งได้

ในการกำหนดคุณสมบัติในรูปแบบการสื่อสารต่างควรให้ตรงกันระหว่างอุปกรณ์ตัวรับกับตัวส่ง



รูปที่ 2.15 แสดงคอนโทรล MSComm

## 2.5.2 คุณสมบัติของตัวคอนโทรล(Property)

เป็นสิ่งที่ใช้บรรยายลักษณะต่างๆ ของคอนโทรล ทำให้คอนโทรลสามารถใช้งานที่หลากหลายรูปแบบมากขึ้น

และมีลักษณะแปลกตาไม่ซ้ำกัน การกำหนดค่าคุณสมบัติอาจทำในขณะที่ออกแบบผ่านทางหน้าต่างคุณสมบัติหรือกำหนดขณะรันแอปพลิเคชัน โดยเขียนคำสั่งโปรแกรมให้ตั้งคุณสมบัติขณะโปรแกรมทำงานคุณสมบัติที่สำคัญได้แก่

เอกสารนี้เป็น BackColor ใช้กำหนดสีส่วนหลัง (Background) ของ อกอปเจ็คต์ ให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณี Caption ใช้กำหนดข้อความที่แสดงบน ใต้เคปชั่นบาร์ของอกอปเจ็คต์ เอกสารทุกครั้งที่มีการนำไปใช้

ControlBox เป็นคุณสมบัติที่บอกว่าให้มีเมนูที่ซ่อน ขยาย หรือปิดหน้าต่างอยู่ที่มุมบนด้านซ้ายของหน้าต่างหรือไม่ ถ้าเป็น True จะมีการแสดงปุ่มเมนู ถ้าเป็น False จะไม่มีการแสดงปุ่มเมนู

ForeColor เป็นคุณสมบัติที่ใช้กำหนดสีของข้อความหรือกราฟฟิกบนออบเจกต์

Height,Width เป็นคุณสมบัติที่ใช้กำหนดความสูงและความกว้างของออบเจกต์ซึ่งวัดอยู่ในหน่วยที่เรียกว่า ทวิพ ( 1 ทวิพ = 1/1440 นิ้ว )

Name เป็นคุณสมบัติที่ใช้กำหนดชื่อของฟอร์มในรหัสโปรแกรม ซึ่งเปลี่ยนแปลงได้เฉพาะในขณะออกแบบเท่านั้น

Visible เป็นคุณสมบัติที่กำหนดว่าจะให้เห็นออบเจกต์หรือไม่ในขณะรันโปรแกรม

Hide เป็นเมธอดที่ใช้ซ่อนออบเจกต์ไว้ชั่วคราว

Show เป็นเมธอดที่ใช้ในการแสดงออบเจกต์ที่ต้องการ

LoadPicture เป็นเมธอดที่ใช้ตั้งคุณสมบัติ Picture ของออบเจกต์

SetFocus เป็นเมธอดที่ใช้ในการ โฟกัสมายังออบเจกต์นี้เพื่อให้ผู้ใช้สามารถทำงานด้วยได้

FontBold,FontItalic,FontName,FontSize,FontUnderline เป็นคุณสมบัติที่กำหนดรูปแบบของข้อความที่แสดงในคอนโทรลว่าจะให้ใช้ตัวอักษรแบบใด ( FontName )ขนาดเท่าใด ( FontSize) เป็นตัวหนา ( FontBold ) ตัวเอียง ( FontItalic) หรือขีดเส้นใต้ (FontUnderline)

Maxlength เป็นคุณสมบัติที่กำหนดจำนวนอักขระสูงสุดของข้อความ โดยถ้ากำหนดเป็น 0 จะรับข้อความได้ไม่จำกัด

Multiline เป็นคุณสมบัติที่กำหนดให้แท็กบ็อกซ์แสดงข้อความในรูปแบบหลายบรรทัดได้

ScrollBars เป็นคุณสมบัติที่กำหนดรูปแบบของข้อความที่แสดงในคอนโทรลว่าจะให้แสดงสกรอลบาร์ด้วยหรือไม่อย่างไร เมื่อข้อความยาวเกินความกว้างหรือความยาวของคอนโทรล

SelLength เป็นคุณสมบัติที่บอกว่าความยาวของข้อความที่เลือกอยู่ในขณะนั้นมีค่าเท่าไร

SelStart เป็นคุณสมบัติที่บอกว่าตำแหน่งแรกของข้อความที่เลือกอยู่ในขณะนั้นอยู่ในตำแหน่งที่เท่าไร ถ้าไม่มีการเลือกข้อความอยู่ จะบอกตำแหน่งที่จะแทรกข้อความอยู่ในขณะนั้น

SelText เป็นคุณสมบัติที่บอกว่าข้อความส่วนที่เลือกอยู่ในขณะนั้นคืออะไร

Text เป็นคุณสมบัติที่เก็บข้อความของแท็กบ็อกซ์ในปัจจุบันไว้ และผู้ใช้ยังสามารถโปรแกรมดัดแปลงแก้ไขข้อความที่แสดงออกมาได้

### 2.5.3 เมธอด (Method )

เป็นโปรแกรมย่อยประเภทหนึ่ง ซึ่งเป็นสมาชิกของคอนโทรลนั้นๆ เหมือนกับคุณสมบัติของคอนโทรล เมื่อเรียกใช้เมธอดจะเป็นการสั่งให้คอนโทรลทำงานให้ ซึ่งอาจจะมีการเปลี่ยนแปลงค่าคุณสมบัติต่างๆ ของคอนโทรลนั้น

Cls เป็นเมธอดที่ใช้เคลียร์สิ่งที่วาดบนออบเจกต์

Hide เป็นเมธอดที่ใช้ซ่อนออบเจกต์ไว้ชั่วคราว

Show เป็นเมธอดที่ใช้ในการแสดงออบเจกต์ที่ต้องการ

LoadPicture เป็นเมธอดที่ใช้ตั้งคุณสมบัติ Picture ของออบเจกต์

SetFocus เป็นเมธอดที่ใช้ในการโฟกัสมายังออบเจกต์นี้เพื่อให้ผู้ใช้สามารถทำงานด้วยได้

### 2.5.4 อีเวนต์ ( Event )

เป็นการตอบสนองต่อเหตุการณ์ภายนอกของคอนโทรล และเป็นโปรแกรมย่อยที่ทำงานทันทีที่เกิดเหตุการณ์นั้นขึ้นมาอีเวนต์ที่สำคัญได้แก่

Click เป็นอีเวนต์ที่เกิดขึ้นเมื่อมีการคลิกเมาส์บนออบเจกต์ อีเวนต์ Click ก็จะทำกรรันคำสั่งที่เขียน

Change เป็นอีเวนต์ที่เกิดขึ้นเมื่อมีการเปลี่ยนแปลงคุณสมบัติ Text ของ เท็กซ์บ็อกซ์ ไม่ว่าจะมาจากผู้ใช้ป้อนข้อความเข้าไปใหม่ หรือ จากการตั้งค่าคุณสมบัติ Text ใหม่ ในโปรแกรม

LostFocus เป็นอีเวนต์ที่เกิดขึ้นเมื่อผู้ใช้งานตั้งระยะเคลื่อนออกไปจากเท็กซ์บ็อกซ์หรือใช้เมาส์เลือกออบเจกต์อื่นๆ

เมธอด

SetFocus เป็นวิธีการที่มีประโยชน์ที่สุดในการชี้ความสนใจไปยังเท็กซ์บ็อกซ์ที่สำคัญที่สุดในหลายๆ เท็กซ์บ็อกซ์ โดยเคอร์เซอร์จะไปอยู่ในเท็กซ์บ็อกซ์ที่เรากำหนด

## 2.6 เครื่องควบคุมแบบโปรแกรมได้ (Programmable Logic Controller)

### 2.6.1 ความหมายและความสำคัญ

PLC (Programmable Logic Controller) เป็นอุปกรณ์ควบคุมชนิดหนึ่ง ที่นำมาแทนที่การควบคุมที่ใช้รีเลย์ ทำให้สะดวกขึ้น PLC เป็นอุปกรณ์ชนิด Solid State ที่ทำงานแบบลอจิก (Logic Function) การออกแบบการทำงานของ PLC จะคล้ายกับหลักการทำงานของคอมพิวเตอร์ จากหลักการพื้นฐานแล้ว PLC จะประกอบด้วยอุปกรณ์ที่เรียกว่า Solid-State Digital Logic Elements เพื่อให้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่ออนุญาตเห็นไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ทำงานและตัดสินใจแบบลอจิก PLC ใช้สำหรับควบคุมกระบวนการทำงานของเครื่องจักรและอุปกรณ์ในโรงงานอุตสาหกรรม

การใช้ PLC สำหรับควบคุมเครื่องจักรหรืออุปกรณ์ต่าง ๆ ในโรงงานอุตสาหกรรมจะมีข้อได้เปรียบกว่าการใช้ระบบของรีเลย์ (Relay) ซึ่งจำเป็นจะต้องเดินสายไฟ ฉะนั้นเมื่อมีความจำเป็นที่ต้องเปลี่ยนกระบวนการผลิต หรือลำดับการทำงานใหม่ ก็ต้องเดินสายไฟใหม่ ซึ่งเสียเวลาและเสียค่าใช้จ่ายสูง แต่เมื่อเปลี่ยนมาใช้ PLC แล้ว การเปลี่ยนกระบวนการผลิตหรือลำดับการทำงานใหม่นั้น ทำได้โดยการเปลี่ยนโปรแกรมใหม่เท่านั้น นอกจากนี้แล้ว PLC ยังใช้ระบบไซคลิก-สเตท ซึ่งนำเชื่อถือกว่าระบบเดิม การกินกระแสไฟฟ้าน้อยกว่า และสะดวกกว่าเมื่อต้องการขยายขั้นตอนการทำงานของเครื่องจักร จึงกล่าวได้ว่า PLC สามารถควบคุมเครื่องจักรได้ทุกชนิด อีกทั้งมีประสิทธิภาพสูง นำเชื่อถือกว่าระบบควบคุมแบบเดิม เมื่อเปรียบเทียบกับระบบซีควีนซ์ (Sequence) หรือใช้การเดินสายไฟแบบเก่าแล้ว PLC มีข้อดีกว่าดังต่อไปนี้

- แก้ไขได้ง่าย
- ติดต่อกับระบบอื่นได้ง่าย
- ติดตั้งง่าย
- ลดการเดินสายไฟฟ้าควบคุม
- เนื้อที่ติดตั้งน้อยกว่า
- มีความน่าเชื่อถือสูงกว่า
- บำรุงรักษา และซ่อมแซมง่าย
- มีประสิทธิภาพการทำงานสูงกว่า

## 2.6.2 ส่วนประกอบของ PLC

PLC มีส่วนประกอบหลักดังนี้

1. หน่วยความจำ (Memory Unit)
2. ส่วนที่เป็นหน่วยประมวลผลกลาง ( Central Processing Unit : CPU)
3. ส่วนที่เป็นหน่วยรับข้อมูลและหน่วยส่งข้อมูล( Input /Output : I/O )
4. ส่วนที่เป็นอุปกรณ์การ โปรแกรม ( Programming Device)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



## หน่วยประมวลผลกลาง (CPU)

เป็นส่วนมันสมองของระบบ โดยภายในประกอบด้วยวงจร Logic Gate ชนิดต่างๆหลายชนิด และมี Microprocessor-based ใช้สำหรับแทนอุปกรณ์จำพวกรีเลย์ (Relay) เคาน์เตอร์ (Counter) ไทเมอร์ (Timer) และซีควเอนเซอร์ (Sequencer) เพื่อให้ผู้ใช้ได้ออกแบบใช้วงจรรีเลย์ แลคเตอร์ ลอจิก (Relay Ladder Logic) เข้าไปได้

หน่วยประมวลผลกลางจะรับ (Read) ข้อมูล (Input Data) จากอุปกรณ์ให้สัญญาณ (Sensing Device) ต่าง ๆ จากนั้นจะปฏิบัติการเก็บข้อมูลโดยใช้โปรแกรมจากหน่วยความจำ และส่งข้อมูลที่เหมาะสมถูกต้องไปยังอุปกรณ์ควบคุม (Control Device) แหล่งของกระแสไฟฟ้าตรง (DC Current) สำหรับใช้สร้างแรงดันไฟฟ้าต่ำ ๆ (Low Level Voltage) ซึ่งใช้โดยโปรเซสเซอร์ (Processor) และ I/O Modules ซึ่งแหล่งจ่ายไฟนี้จะเก็บไว้ที่ หน่วยประมวลผลกลาง หรือแยกออกไปติดตั้งที่จุดอื่นก็ได้

การประมวลผลของหน่วยประมวลผลกลางจากโปรแกรมทำได้โดยรับข้อมูลจากหน่วยอินพุตเข้ามาทำการประมวลผล แล้วส่งข้อมูลที่ได้ออกไปยังเอาต์พุต จากนั้นก็วกกลับไปรับข้อมูลอินพุตเข้ามาอีก การทำในลักษณะนี้เรียกว่า การสแกน (Scan Time)



รูปที่ 2.17 Input / Output Scan และ Program Scan

### ส่วนของหน่วยรับข้อมูลและหน่วยส่งข้อมูล (I/O UNIT)

หน่วยรับข้อมูล ทำหน้าที่รับสัญญาณจากอุปกรณ์ภายนอกที่เป็นสวิตช์และตัวตรวจจับชนิดต่าง ๆ (sensor) ของเครื่องจักรหรือกระบวนการ แล้วแปลงเป็นสัญญาณ AC หรือ DC ที่เหมาะสมเพื่อส่งให้แก่หน่วยประมวลผลกลาง

ในส่วนของหน่วยส่งข้อมูล จะทำหน้าที่รับค่าสถานะที่ได้จากการประมวลผลของหน่วยประมวลผลกลาง แล้วนำค่าที่ได้นี้ไปขยายสัญญาณออกให้มีขนาดใหญ่พอจะขับอุปกรณ์ภายนอกไปใช้

เช่นรีเลย์, โซลินอยด์ หรือหลอดไฟ,มอเตอร์, วาล์ว, บี้ม เป็นต้น นอกจากนั้นแล้ว ยังทำหน้าที่แยกสัญญาณของหน่วยประมวลผลกลางออกจากอุปกรณ์เอาต์พุตเพื่อป้องกันความเสียหายที่อาจเกิดขึ้นได้

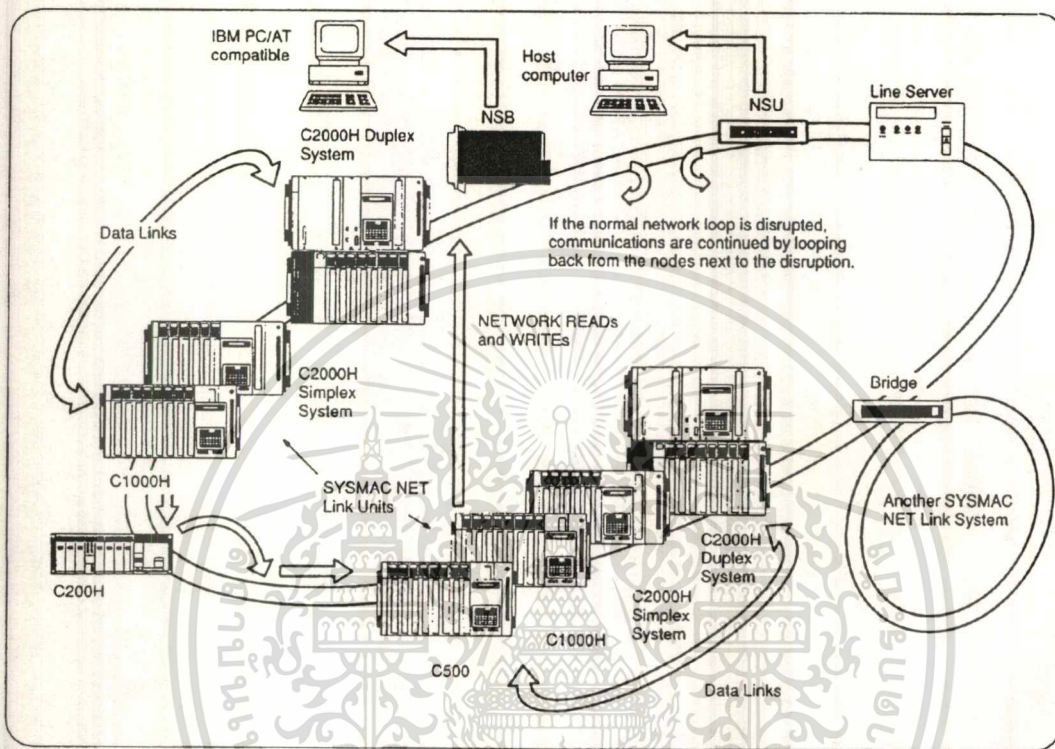
โดยปกติเอาต์พุตนี้จะสามารถขับโหลดได้ด้วยกระแสไฟฟ้าประมาณ 1-2 แอมแปร์ ถ้าโหลดต้องการกระแสไฟฟ้ามากกว่านี้ จะต้องต่อเข้ากับอุปกรณ์ขับหรือขยายอีกทีหนึ่ง เช่น รีเลย์ โซลิตสเตรรีเลย์ และคอนแทคเตอร์ เป็นต้น

#### เครื่องป้อนโปรแกรม (Programming Device)

หน้าที่ของเครื่องป้อนโปรแกรมคือ ควบคุมโปรแกรมของผู้ใช้ลงในหน่วยความจำของ PLC นอกจากนั้นแล้วยังทำหน้าที่ติดต่อระหว่างผู้ใช้กับ PLC เพื่อให้ผู้ใช้สามารถตรวจการปฏิบัติงานของ PLC และผลการควบคุมเครื่องจักรและกระบวนการตาม โปรแกรมควบคุมที่ผู้ใช้เขียนขึ้นได้อีกด้วย

เครื่องป้อนโปรแกรม (Hand Held) แต่ละยี่ห้อจะไม่เหมือนกันแต่มีจุดประสงค์อันเดียวกัน

### บทที่ 3 การติดตั้งระบบโครงข่าย



รูปที่ 3.1 ภาพแสดงการติดตั้ง Hardware ในระบบโครงข่าย

### 3.1 การติดตั้ง PLC (รุ่น C1000H, Omron co.,ltd)

PLC (Programmable Logic Controller) อาศัยการทำงานบนพื้นฐานของ CPU (Central Processing Unit) ซึ่งถูกโปรแกรม และเชื่อมต่อกับอุปกรณ์ Input ,Output (I/O) ส่วนของโปรแกรม จะควบคุมการทำงานของ PLCL เมื่อสัญญาณอินพุตจากอุปกรณ์อินพุตเป็น ON ก็จะเกิดการตอบสนองขึ้น ซึ่งการตอบสนองนี้มักจะถูกนำไปต่อกับอุปกรณ์เอาต์พุตตัวอื่นให้เกิดการทำงานตามที่ต้องการ

อุปกรณ์อินพุตอาจเป็น Photoelectric Sensors, ปุ่มกด (Push Bottom), Limit Switch หรือ อุปกรณ์อื่น ๆ ที่สามารถสร้างสัญญาณส่งไปยัง PLC ส่วนอุปกรณ์เอาต์พุตอาจเป็น Solenoids , สวิตช์ควบคุมหลอดไฟ หรือ LED, รีเลย์สำหรับเปิด motor หรืออุปกรณ์อื่น ๆ ที่สามารถทำให้เกิด

การทำงานเมื่อมีสัญญาณจาก PLC มากระตุ้นเพื่อการศึกษาเท่านั้น ไม่นิยมนำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

PLC มีต้นกำเนิดมาจากระบบการควบคุมรีเลย์ ถึงแม้ว่า Integrated Circuit และการทำงานเป็น Logic ภายใน PLC สามารถทดแทนการทำงานของรีเลย์ ,Timer,Counter ได้จึงเสมือนว่ามีอุปกรณ์เหล่านี้อยู่ใน PLC แต่ PLC สามารถใช้งานร่วมกับ Computer ได้และยังให้ความเที่ยงตรงสูง ให้การทำงานที่สามารถยืดหยุ่น และมีความน่าเชื่อถือมากกว่ารีเลย์

C1000H เป็น PLC รุ่นหนึ่งของ OMRON มีลักษณะเป็น Rack PLC มีหลาย Unit รวมกันเพื่อให้เกิดประสิทธิภาพและบรรลุผลในการประยุกต์ใช้งาน สามารถขยาย Rack ในอนาคตได้ โดยเพิ่ม “Additional Unit” เข้าไป เช่น Special I/O Units เพื่อใช้ในงานพิเศษหรือเป็น Link Unit สำหรับระบบ Optical LAN สามารถใช้ต่อร่วมกับ SNT-31-V4 ซึ่งอุปกรณ์เสริมขึ้นมาเพื่อให้ PLC สามารถติดต่อกับอุปกรณ์อื่นได้โดยแปลงเป็นสัญญาณแสงผ่านเส้นใยแสงนอกจากนี้ยังมีคำสั่งพิเศษเพิ่มเติมที่ช่วยในการติดต่อข้อมูลผ่านระบบโครงข่ายโดยตรงคือคำสั่ง Send(90) , Receive(98) ใช้ในการรับ-ส่งข้อมูล

### 3.1.1 การแบ่งพื้นที่หน่วยความจำของ PLC (รุ่น C1000H)

พื้นที่	ตัวย่อ	RANGE	หน้าที่
Internal Relay	IR	Words : 000 to 236 Bits : 0000 to 23615	ใช้ในการควบคุม I/O points, timers และ counter และใช้เก็บข้อมูลชั่วคราว
Special relay	SR	Words : 237 to 255 Bits : 23700 to 25515	เก็บ system clock ,flags,control bits ,ข้อมูลเกี่ยวกับสถานะ และมีหลาย word ที่ใช้ในระบบการเชื่อมต่อ
Auxiliary Relay	AR	Words : AR 00 to AR 27 Bits : AR 00 to AR2715	เก็บ flag และ bit สำหรับ function พิเศษ เช่น write-protecting FM area
Data Memory	DM	DM 0000 to DM 4095 (words only)	ใช้สำหรับจัดเก็บข้อมูลและกระทำข้อมูล
Holding Relay	HR	Words : HR 00 to HR 99 Bits : HR 0000 to HR 9915	ใช้สำหรับจัดเก็บและสงวนข้อมูล เมื่อ PLC หยุดการทำงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น อนุญาตให้นำไปใช้ประโยชน์ด้านการศึกษา  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้เผยแพร่ลงเนื้อหาและต้องอ้างอิงแหล่งที่มาของเอกสารทุกแห่งที่มีกรณีไปใช้

Timer/Counter	TC	TC 000 to TC 511 (TC number ใช้ในการเข้าถึงข้อมูลอื่น)	ใช้ในการกำหนดTimerCounter และการเข้าถึง completion flag PV ,SV
พื้นที่	ตัวย่อ	RANGE	หน้าที่
Temporary Relay	TR	TR 00 to TR 07 (bit only)	ใช้ในการจัดเก็บ Execution Condition ชั่วคราว
Program Memory	UM	UM: depends on Memory Unit used	เก็บโปรแกรมที่ถูก Executed โดย CPU
File Memory	FM	FM: 0000 to 0999 or 0000 to 1999	กำหนดตำแหน่งของ File Memory ที่ถูกติดตั้งใน CPU Rack และใช้เก็บโปรแกรมหรือข้อมูล
File Memory	TM	TM: Traces of 250 instructions	ใช้เก็บผลลัพธ์จากการติดตามการ Execution ของ Program

ตารางที่ 3.1 แสดงการแบ่งพื้นที่หน่วยความจำของ PLC (รุ่น C1000H)

### IR Area (Internal Relay Area)

ใช้ทั้งในการควบคุม I/O points และใช้เป็น bit ทำงาน(Work bit) เพื่อกระทำหรือจัดเก็บข้อมูลภายใน IR Area สามารถเข้าถึงได้ทั้งระดับ bit และระดับ word

word ที่ใช้ในการควบคุม I/O points เรียกว่า I/O word ซึ่ง word ที่เหลือจาก I/O word จะเป็น word ทำงาน (work word) สามารถนำมาใช้ในการเขียนโปรแกรมเพื่อกระทำกับข้อมูลและควบคุม bit อื่นๆ bit ทำงานจะถูก reset เมื่อไฟเลี้ยงถูกตัด หรือ PLC หยุดทำงาน การแบ่งพื้นที่ I/O word และ work word เป็นดังนี้

I/O words : 000 to 063

I/O bits : 00000 to 06315

work words : 064 to 236

work bits : 06400 to 23615

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



### DM Area (Data Memory Area)

พื้นที่ DM ใช้สำหรับเก็บข้อมูลภายในและกระทำข้อมูล การเข้าถึงหน่วยความจำในพื้นที่นี้สามารถเข้าถึงในระดับ word ไม่สามารถเข้าถึงในระดับ bit address range อยู่ระหว่าง DM0000 ถึง DM4095 DM Area จะสามารถคงสถานะอยู่ได้เมื่อ PLC หยุดทำงาน หรือ Power Supply ไม่จ่ายไฟเลี้ยง

### HR Area (Holding Relay Area)

พื้นที่ของ HR ใช้สำหรับเก็บและกระทำข้อมูลได้หลากหลาย สามารถเข้าถึงได้ทั้งในระดับ word และระดับ bit address range อยู่ระหว่าง HR 00 ถึง HR 99 HR bit สามารถใช้งานได้ ในหลายลักษณะ และสามารถ Programmed ได้บ่อยเท่าที่ต้องการ

HR Area จะคงสถานะอยู่ได้เมื่อมีการเปลี่ยนแปลง Operation Mode , Power supply ไม่จ่ายไฟเลี้ยง หรือเมื่อ PLC หยุดทำงาน

### TC Area (Timer/Counter Area)

พื้นที่ TC ใช้สำหรับสร้าง โปรแกรม Timer/counter และคงค่า Completion Flags ,Set Value (SV),Present Value (PV) TC Number มี range ตั้งแต่ TC 000 ถึง TC 511

### LR Area (Link Relay Area)

พื้นที่ LR ใช้สำหรับสื่อสารข้อมูลระหว่าง PLCs LR สามารถเข้าถึงได้ทั้งในระดับ word และ bit มี address range ตั้งแต่ LR 00 ถึง LR 63 พื้นที่ LR ส่วนที่ไม่ได้ใช้ใน PLC Link สามารถใช้เป็น word ทำงานหรือ bit ทำงานได้

พื้นที่ LR จะไม่สามารถคงสถานะไว้ได้เมื่อ Power Supply ไม่จ่ายไฟเลี้ยง หรือ PLC หยุดทำงาน

### Program Memory

เป็นส่วนที่ใช้บรรจุ User Program หน่วยความจำ Program Memory เป็นได้ทั้ง 8K,16K,24K หรือ 32K ขึ้นอยู่กับชนิดของ Memory Unit ที่ติดตั้งใน CPU

### File Memory

พื้นที่นี้จะมีก็ต่อเมื่อ ได้ติดตั้ง File Memory Unit เข้าไปใน PLC แล้วเท่านั้น พื้นที่นี้จะสามารถเข้าถึงโดยใช้ block unit เท่านั้น

### Trace Memory

พื้นที่นี้จะใช้เก็บผลจากการติดตาม Execution

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.1.2 คำสั่งในการรับ-ส่งข้อมูลระหว่าง PLC (รุ่น C1000H)

การใช้คำสั่งของ PLC(C1000H) ในระดับโครงข่ายจะใช้สำหรับการสื่อสารข้อมูลระหว่าง PLC แต่ละตัวผ่าน ระบบการสื่อสารระหว่างโครงข่าย (SYSMAC NET LINK SYSTEM) หรือ ระบบการสื่อสารภายในโครงข่าย (SYSMAC LINK SYSTEM) ซึ่งการสื่อสารระดับโครงข่ายนี้มี 2 คำสั่งคือ

#### NETWORK SEND - SEND(90)

เป็นคำสั่งเมื่อต้องการส่งข้อมูลจาก PLC ต้นทาง (Source) ไปยัง PLC ปลายทาง (Destination)

#### Ladder Symbol

SEND(90)
S
D
C

@SEND(90)
S
D
C

S: พื้นที่หน่วยความจำที่ต้องการส่งจากต้นทาง (IR,SR,AR,DM,HR,TC,LC)

D: พื้นที่หน่วยความจำปลายทางที่ต้องการส่งข้อมูล ไปถึง (IR,AR,DM,HR,LC,TC)

C: ส่วนที่ใช้ควบคุมในการส่งข้อมูล (IR,AR,DM,HR,TC,LR)

- การกำหนดว่าจะส่งข้อมูลภายในโครงข่ายหรือระหว่างโครงข่าย สามารถกำหนดได้ที่ส่วนควบคุม การส่ง (Control Word :C) ดังนี้

- การส่งข้อมูลระหว่างโครงข่าย (SYSMAC NET LINK SYSTEM) กำหนด Control Word ดังนี้

หมายเลข port ปลายทาง ( destination port number) ต้องตั้งค่าเป็น 0 เสมอ ถ้าตั้งหมายเลข nodeปลายทางเป็น 0 หมายความว่าต้องการส่งข้อมูล ไปยังทุกๆnodeในโครงข่าย

Word	Bit 00 to 07	Bit 08 to 15
C	จำนวนคำ (Words) ที่ต้องการส่ง (0-1000,4-digit)	
C + 1	หมายเลขของ network ปลายทาง (0-127)	Bit 14 ON: ปฏิบัติการที่ระดับ 0 OFF: ปฏิบัติการที่ระดับ 1 Bit 08 ถึง 13 และ 15: ตั้งค่าเป็น 0
C + 2	หมายเลขของ node ปลายทาง (1-126)	Port ปลายทาง (Destination port) NSB : 00 NSU : 01 / 02

### ตาราง 3.3 ส่วนควบคุมการส่งของคำสั่ง SEND (90) ระหว่างโครงข่าย

- การส่งข้อมูลภายในโครงข่าย (SYSMAC LINK SYSTEM) กำหนด Control Word ดังนี้ ตั้งค่าหมายเลขnodeปลายทางเป็น(0)เมื่อต้องการส่งข้อมูลไปทุกๆ node

Word	Bit 00 ถึง 07	Bit 08 ถึง 15
C	จำนวนคำ (Words) ที่ต้องการส่ง ( 0-1000,4-digits)	
C+1	ขีดจำกัดของเวลาตอบสนองในหน่วยวินาที ถ้าตั้งค่าเป็น 0(hex)เวลาตอบสนองจะเป็น 2 วินาที ถ้าตั้งค่าเป็น FF(hex)เวลาตอบสนองจะไม่มีสิ้นสุด	Bit 12 ตั้งค่าเป็น(0) Bit 13 ON : Response ไม่กลับมา OFF: Response กลับมา Bit 14 ON:ปฏิบัติการที่ระดับ0 OFF:ปฏิบัติการที่ระดับ1 Bit 15 ตั้งค่าเป็น1
C+2	หมายเลขของ Node ปลายทาง	ตั้งค่าเป็น(0)

### ตาราง 3.4 แสดงส่วนควบคุมการส่งของคำสั่ง SEND(90) ภายในโครงข่าย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**NETWORK RECEIVE-RECV(98)**

เป็นคำสั่งเมื่อต้องการรับข้อมูลจาก PLC ต้นทาง (Source)มายัง PLC ปลายทาง (Destination)

**Ladder Symbol**

RECV(98)
S
D
C

@RECV(98)
S
D
C

S:พื้นที่หน่วยความจำต้นทางที่ต้องการรับข้อมูลมา (IR,SR,AR,DM,HR,TC,LR)

D:พื้นที่หน่วยความจำปลายทางที่ต้องการรับข้อมูลมาถึง (IR,AR,DM,HR,TC,LR)

C:ส่วนที่ใช้ควบคุมในการรับข้อมูล (IR,AR,DM,HR,TC,LR)

- การกำหนดว่าจะรับข้อมูลภายในโครงข่ายหรือระหว่างโครงข่าย สามารถกำหนดได้ที่ส่วนควบคุมการรับข้อมูล ( Control Word) ดังนี้

- การรับข้อมูลระหว่างโครงข่าย (SYSMAC NET LINK SYSTEM)กำหนด Control

Wordดังนี้ หมายเลข port ต้นทาง(source port number) ต้องตั้งเป็น(0)เสมอ

Word	Bit 00 to 07	Bit 08 to 15
C	จำนวนคำ (Words) ที่ต้องการส่ง (4-digit)	
C + 1	หมายเลขของ network ต้นทางที่ต้องการรับข้อมูลมา (0-127)	Bit 14 ON: ปฏิบัติการที่ระดับ 0 OFF: ปฏิบัติการที่ระดับ 1 Bit 08 ถึง 13 และ 15: ตั้งค่าเป็น 0
C + 2	หมายเลขของ node ต้นทางที่ต้องการรับข้อมูลมา (1-126)	Port ต้นทาง (Source port) NSB : 00 NSU : 01 / 02

ตาราง 3.5 แสดงส่วนควบคุมการรับของคำสั่ง RECV(98) ระหว่างโครงข่าย

- การรับข้อมูลภายในโครงข่าย (SYSMAC LINK SYSTEM)

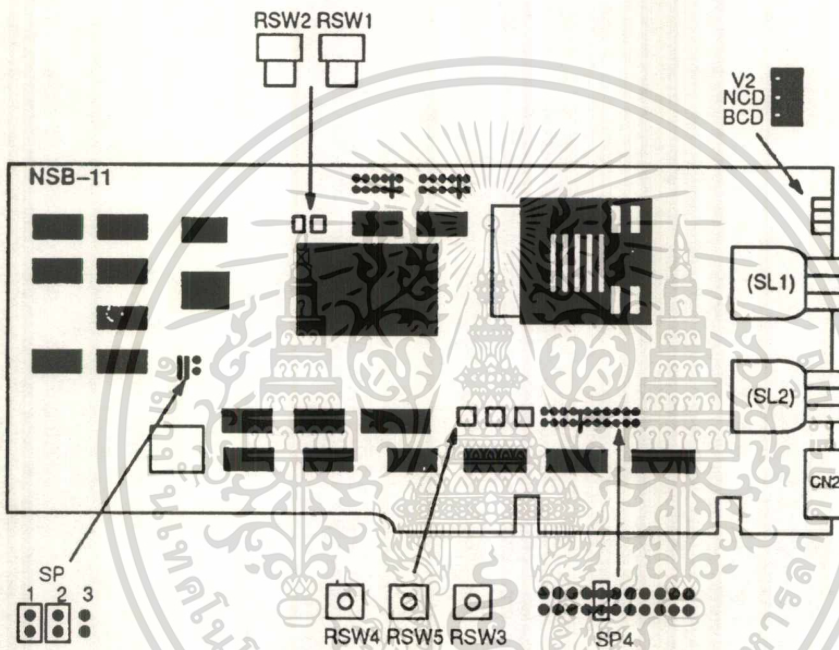
Word	Bit 00 ถึง 07	Bit 08 ถึง 15
C	จำนวนคำ (Words) ที่ต้องการส่ง ( 0-1000,4-digits)	
C+1	ขีดจำกัดของเวลาตอบสนองในหน่วยวินาที ถ้าตั้งค่าเป็น 0(hex)เวลาตอบสนองจะเป็น 2 วินาที ถ้าตั้งค่าเป็น FF(hex)เวลาตอบสนองจะไม่มีสิ้นสุด	Bit 12 ตั้งค่าเป็น(0) Bit 13 ตั้งค่าเป็น(0) Bit 14 ON:ปฏิบัติการที่ระดับ 0 OFF:ปฏิบัติการที่ระดับ 1 Bit 15 ตั้งค่าเป็น 1
C+2	หมายเลขของ Node ต้นทาง	ตั้งค่าเป็น(0)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 ตาราง 3.6 แสดงส่วนควบคุมการรับของคำสั่ง RECV(98) ภายในโครงข่าย  
 ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.2 การติดตั้ง Network Support Board

Network Support Board คือ อุปกรณ์ที่ต่อเข้าไปในเครื่อง AT Computer เพื่อใช้เป็นตัวเชื่อมต่อข้อมูลระหว่าง AT Computer กับระบบโครงข่าย ในรูปแบบการเชื่อมต่อโดยใช้เส้นใยแสงของ S3200 optical local area network

Network Support Board จะใช้ติดตั้งกับ PLC-DOS versions 3.1 ถึง 3.3 หน่วยความจำอย่างน้อย 256 กิโลไบต์



รูปที่ 3.2 แสดง Network Support Board

#### 3.2.1 การตั้งค่า I/O ADDRESS

RSW4 และ RSW5 คือ rotary switch ที่ใช้ติดตั้งเพื่อเป็นตัวบอกตำแหน่ง I/O address ของ Network support board เองกับ CPU โดยทั่วไปจะตั้งค่าไว้ที่ 0300H ถึง 030FH

### 3.2.2 การ INSTALL NETWORK SUPPORT BORD

Content parameter of sysmacnet driver I/O Address

Parameter of config.sys

Buffer	20
Files	20
I/O Address	300
Interrupt	00
Network Address	01
Output message	English
Connection with F.C	****
Automatic fixture of routing table	****
Using service	
Datagram	USE
Buffer size	2000
Number of Datagram Buffer	3
Number of Datagram With ACK Buffer	0
Transmitting File (TFTP)	****
Auto Restart	****
File Server (Server)	****
Auto Restart	****
File server (Requester)	****
Name Service	****
Auto Restart	****
TCP	****

ARE YOU SURE?[Y/N]

### 3.3 การติดตั้ง Bridge

บริดจ์ทำหน้าที่รับและตรวจสอบแพคเกจที่ส่งผ่านเข้ามาในเน็ตเวิร์ค ยิ่งกว่านั้นฟังก์ชันของบริดจ์จะเปรียบเทียบแอดเดรสของแพคเกจที่รับได้กับตารางแอดเดรสเพื่อตัดสินใจว่าควรที่จะทิ้งหรือส่งต่อแพคเกจนั้นเมื่อบริดจ์ตรวจพบแพคเกจที่ถูกส่งไปยัง LAN อื่น บริดจ์จะเก็บแพคเกจนั้นและเตรียมส่งต่อด้วยการแอดเดรส LAN ค้างส่งออก

#### 3.3.1 Software Setting ของ Bridge

<Node Configuration Parameter Configuration>		
Net :1 Node :1 Node type :SBE		
NO.	Item	contents
1.	Lan 1 Network Address	:1
2.	Lan 2 Network Address	:2
3.	Gateway TTL Initial Value	:15
4.	Gateway TTL Decrease Value	:1
5.	Gateway TTL Limit Value	:1
6.	IP Trace Switch	:don't log
7.	Broadcast Packet Pass	:don't pass
8.	Netware Packet Pass	:don't pass
----- Status :Enter command> -----		
ESC ;	terminal	1.Editing
CTRL-J:	next entry	2.Load from file
CTRL-K:	previous entry	3.Read from node
P:	print	4.Save to disk
		5.Write to node

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้ในการใช้งานเพื่อการศึกษานี้ ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
**ตารางที่ 3.8 พารามิเตอร์ ในการติดตั้ง Bridge**  
 ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- การตั้งเส้นทางสื่อสารของ Lan1 ในระบบโครงข่าย

SYSNET IP Routing Table Configuration

<- **Lan1** Lan 2 ->

No.	Net-addr	1 <sup>st</sup> -hop-addr	No-entry	No-entry
1.	02000000	01000111		

-----Status : Entry Command >

ESC: terminal	1.Editing
CTRL-J: next entry	2. Load from file
CTRL-K: previous entry	3. Read from node
CTRL-H/L: enter to switch	4.save to disk
	5.Write to node

### ตารางที่ 3.9 พารามิเตอร์ในการตั้งเส้นทางสื่อสารของ Lan1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

● การตั้งเส้นทางสื่อสารของ Lan2 ในระบบโครงข่าย

SYSNET IP Routing Table Configuration

<- Lan1 Lan2 ->

No.	Net-addr	1 <sup>st</sup> -hop-addr	No-entry	No-entry
1.	01000000	02000111		

-----Status : Entry Command >

ESC: terminal	1.Editing
CTRL-J: next entry	2. Load from file
CTRL-K: previous entry	3. Read from node
CTRL-H/L: enter to switch	4.save to disk
	5. Write to node

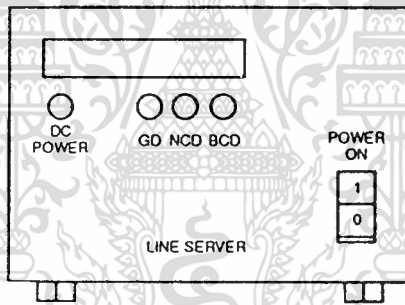
ตารางที่ 3.10 พารามิเตอร์ในการตั้งเส้นทางสื่อสารของ Lan2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.4 การติดตั้ง Line Server ( S3200 – LSU03 –01E)

เป็นอุปกรณ์ที่สร้างและเฝ้าคุม (issue and monitoring) Token ของระบบ ถ้ามันตรวจจับได้ว่า Token ไม่วนกลับมาในเวลาที่กำหนด มันจะสร้าง Token ใหม่ออกมา ในแต่ละโครงข่ายจะต้องมี LINE SERVER อย่างน้อย 1 ตัว ซึ่งถ้า Line Server เสียหรือไม่ทำงาน การสื่อสารภายในโครงข่ายก็จะไม่สามารถเกิดขึ้นได้

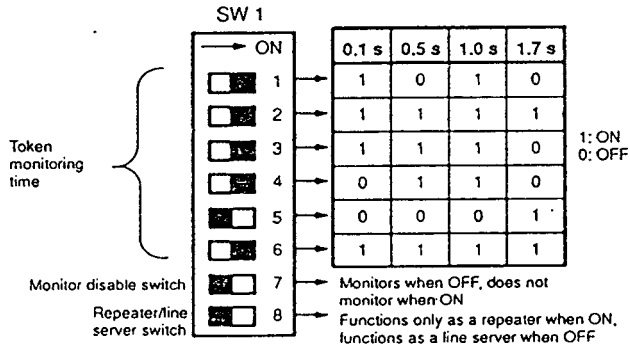
นอกจากนี้ถ้าการเชื่อมต่อภายในโครงข่ายมีระยะทางมากกว่า 1 กม. ต้องเพิ่ม Line Server เข้าไปอีก 1 ตัวในระยะทางที่เพิ่มขึ้น 1 กม. โดย Line Server ที่เพิ่มเข้าไปจะทำหน้าที่เป็น Repeater แต่มีข้อจำกัดว่าใน 1 โครงข่ายสามารถมี Repeater ได้ไม่เกิน 2 ตัว ดังนั้นระยะทางการเชื่อมต่อภายในโครงข่ายต้องไม่เกิน 3 กม. ( หากต้องการเชื่อมต่อในระยะทางที่มากกว่า 3 กม. ต้องแยกออกไปเป็นโครงข่ายใหม่)



รูปที่3.3 แสดง Line Server

DC Power Indicator (green)	: ไฟสว่างเมื่อเปิดเครื่อง
GD Indicator ( red )	: ไฟสว่างเมื่อมี Token ผ่าน Line Server
NCD Indicator(red)	: Normal Carrier Detect ไฟสว่างเมื่อการเดินทางของ Token เป็นแบบ Normal Loop
BCD Indicator( red )	: Back Loop Carrier Detect ไฟสว่างเมื่อการเดินทางของToken เป็นแบบ Back Loop

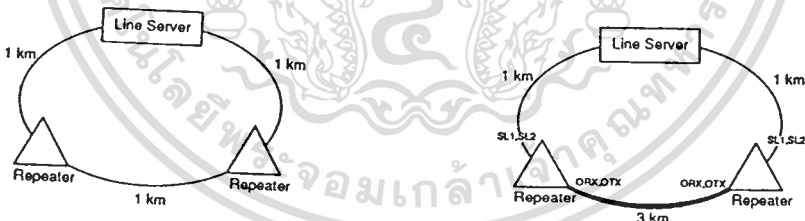
การตั้งค่าเวลาในการเฝ้าคุม Token ของ Line Server รวมถึงการกำหนดหน้าที่ว่าทำหน้าที่เป็น Line Server หรือ Repeater สามารถตั้งได้ที่ Dip switch ซึ่งอยู่ด้านหลังของ Line Server ดังรูป เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่3.4 แสดง Dip switch ของ Line Server

สำหรับ pin 7 และ 8 จะตั้งค่าเป็น OFF, OFF เมื่อ Line Server ตัวนั้นทำหน้าที่ปกติ และจะตั้งเป็น ON, ON เมื่อ Line Server นั้นทำหน้าที่เป็น Repeater

สายใยแก้วนำแสงสามารถใช้ได้ 2 อย่าง คือ แบบมาตรฐาน HC-PLCF (hard core plastic clad fiber) หรือแบบ AGF (all glass fiber) quartz crystal แบบหลังจะใช้เมื่อต้องการให้ระยะห่างระหว่าง Repeater เป็น 3 กม. ดังแสดงในรูป



รูปที่3.5 แสดงการขยายระยะทางภายในโครงข่ายโดยใช้ Repeater

### 3.5 การติดตั้ง Host Link Unit

โดยมีสวิตช์ที่สำคัญอยู่สองสวิตช์คือ

สวิตช์ 1 (SW1) เป็นกลุ่มสวิตช์ที่ใช้ในการตั้งค่ายูนิต (unit) ของตัว PLC และใช้สำหรับในเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า การเลือก Mode การทำงานของ PLC นอกจากนี้ยังใช้เป็นตัวกำหนดรูปแบบในการติดต่อว่าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้เป็นแบบใด

ตามรูป



รูปที่ 3.6 แสดง Dip switch1 ของ Host Link Unit

หมายเลขหน่วย(Unit No.)

1	2	3	4	5	Unit No.	1	2	3	4	5	Unit No.
ปิด	ปิด	ปิด	ปิด	ปิด	0	เปิด	ปิด	ปิด	ปิด	ปิด	16
ปิด	ปิด	ปิด	ปิด	เปิด	1	เปิด	ปิด	ปิด	ปิด	เปิด	17
ปิด	ปิด	ปิด	เปิด	ปิด	2	เปิด	ปิด	ปิด	เปิด	ปิด	18
ปิด	ปิด	ปิด	เปิด	เปิด	3	เปิด	ปิด	ปิด	เปิด	เปิด	19
ปิด	ปิด	เปิด	ปิด	ปิด	4	เปิด	ปิด	เปิด	ปิด	ปิด	20
ปิด	ปิด	เปิด	ปิด	เปิด	5	เปิด	ปิด	เปิด	ปิด	เปิด	21
ปิด	ปิด	เปิด	เปิด	ปิด	6	เปิด	ปิด	เปิด	เปิด	ปิด	22
ปิด	ปิด	เปิด	เปิด	เปิด	7	เปิด	ปิด	เปิด	เปิด	เปิด	23
ปิด	เปิด	ปิด	ปิด	ปิด	8	เปิด	เปิด	ปิด	ปิด	ปิด	24
ปิด	เปิด	ปิด	ปิด	เปิด	9	เปิด	เปิด	ปิด	ปิด	เปิด	25
ปิด	เปิด	ปิด	เปิด	ปิด	10	เปิด	เปิด	ปิด	เปิด	ปิด	26
ปิด	เปิด	ปิด	เปิด	เปิด	11	เปิด	เปิด	ปิด	เปิด	เปิด	27
ปิด	เปิด	เปิด	ปิด	ปิด	12	เปิด	เปิด	เปิด	ปิด	ปิด	28
ปิด	เปิด	เปิด	ปิด	เปิด	13	เปิด	เปิด	เปิด	ปิด	เปิด	29
ปิด	เปิด	เปิด	เปิด	ปิด	14	เปิด	เปิด	เปิด	เปิด	ปิด	30
ปิด	เปิด	เปิด	เปิด	เปิด	15	เปิด	เปิด	เปิด	เปิด	เปิด	31

Monitor Mode/Normal Mode(สวิตช์8)

เปิด	Monitor Mode
ปิด	Normal Mode

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้ภายในเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามเผยแพร่ต่อสาธารณะ และต้องขึ้นไปถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สวิตช์ที่ใช้ในการเลือก Parity / Transmission code

6	7	Parity	Transmission Code
ปิด	ปิด	Even	ASCII 7-bit
ปิด	เปิด	Odd	2 Stop bits
เปิด	ปิด	Even	JIS 8 bit
เปิด	เปิด	Odd	1 stop bit

ตัวเลือกระหว่าง Monitor mode กับ Normal mode

เลือกการทำงานของ PLC โดยการตั้งค่าปุ่มที่ 8 ให้เปิดหรือปิดถ้าเปิดอยู่ก็จะเลือก mode การทำงานเป็นแบบ Monitor Mode ถ้าปิดmodeการทำงานของ PLCจะเป็น Normal Mode

สวิตช์ 2 (SW2) เป็นกลุ่มสวิตช์ที่ใช้กำหนดอัตราเร็วในการรับส่งระหว่างmodeต่างๆในระบบซึ่งอาจจะเป็นการติดต่อระหว่าง PLC กับ PLC หรือระหว่าง คอมพิวเตอร์กับ PLC ก็ได้และยังกำหนดว่าเป็นการติดต่อใน PLC หลายตัวหรือตัวเดียวนอกจากนี้ยังสามารถกำหนดระดับของความต้อองการ(Command Level)ในการติดต่อกับPLC ตามรูป



รูปที่ 3.7 แสดงกลุ่ม Dip Switch 2 ของ Host Link Unit

1	2	3	4	Baud Rate
ปิด	เปิด	ปิด	เปิด	300bps
เปิด	ปิด	ปิด	เปิด	600bps
ปิด	ปิด	ปิด	เปิด	1200bps
เปิด	เปิด	เปิด	ปิด	2400bps
ปิด	เปิด	เปิด	ปิด	4800bps

เปิด ปิด เปิด ปิด	9600bps
ปิด ปิด เปิด ปิด	19200bps

### ระดับการปฏิบัติการ

เปิด	ระดับ0
ปิด	ระดับ1

เลือกประเภทของการเชื่อมต่อ

ปิด	Multiple-link
เปิด	Single-link

### ระดับความต้องการในการติดต่อ(command level)

7	8	Valid levels
ปิด	ปิด	....
เปิด	ปิด	1
ปิด	เปิด	1และ2
เปิด	เปิด	1,2และ3

### ระดับปฏิบัติการ

โดยการกำหนดปุ่มที่5ให้เป็นปิดหรือเปิดถ้าปิดจะถือเป็นระดับปฏิบัติการ1 ถ้าเปิดระดับปฏิบัติการจะเป็น0 โดยจะใช้เมื่อ Host Link Unit มีการใช้งานร่วมกับSysmac Net Link Unit หรือกรณีที่ Host Link Unitsสองตัวมีการทำงานร่วมกันจะต้องกำหนดค่าให้ตัวหนึ่งเปิด(ระดับปฏิบัติการ0)และอีกตัวหนึ่งปิด(ระดับปฏิบัติการ1)

ข้อควรระวัง ถ้ากำหนดค่าปุ่มที่5ไม่ถูกต้องอาจทำให้เกิดความเสียหายในระบบป้องกันได้

### การเลือกชนิดของการเชื่อมต่อ(Link Type Selector)

ถ้าให้ปุ่มที่6ของสวิทช์กลุ่มที่2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**เปิด**

จะเป็นการเชื่อมต่อ Host Link แบบตัวเดียว(single link) โดยที่ไม่จำเป็นต้องกำหนดหมายเลขหน่วย(Unit No.)และFrame CheckSum (FCS) ในการกำหนดInternet Protocol (IP)ก็สามารถติดต่อได้เลย

**ปิด**

จะใช้ในการเชื่อมต่อ Host Link แบบหลายๆตัว(Multiple Link)สามารถใช้ได้ถึง32 Host link Unit ต่อหนึ่ง host computer ถ้าต้องการใช้ โปรแกรม Ladder ปุ่ม6ต้องปิด

**ระดับความต้องการในการติดต่อกับ PLC (Command Levels)**

แบ่งออกเป็นสามระดับด้วยกันคือ

- 1.สามารถอ่านและเขียนข้อมูลลงไปในพื้นที่ซึ่งไม่ใช่พื้นที่สงวนได้
- 2.สามารถส่งโปรแกรมเพื่อทำการอ่านเขียนได้
- 3.สามารถอ่านและลงตารางI/O สามารถปฏิบัติบิบัติการ โปรแกรม LSS ได้และสามารถ

Monitorข้อมูลผ่านโปรแกรมประยุกต์ได้

**รีเซ็ต สวิตช์**

ใช้สำหรับเมื่อต้องการนำข้อมูลเข้าไปใหม่

**CTS Selector**

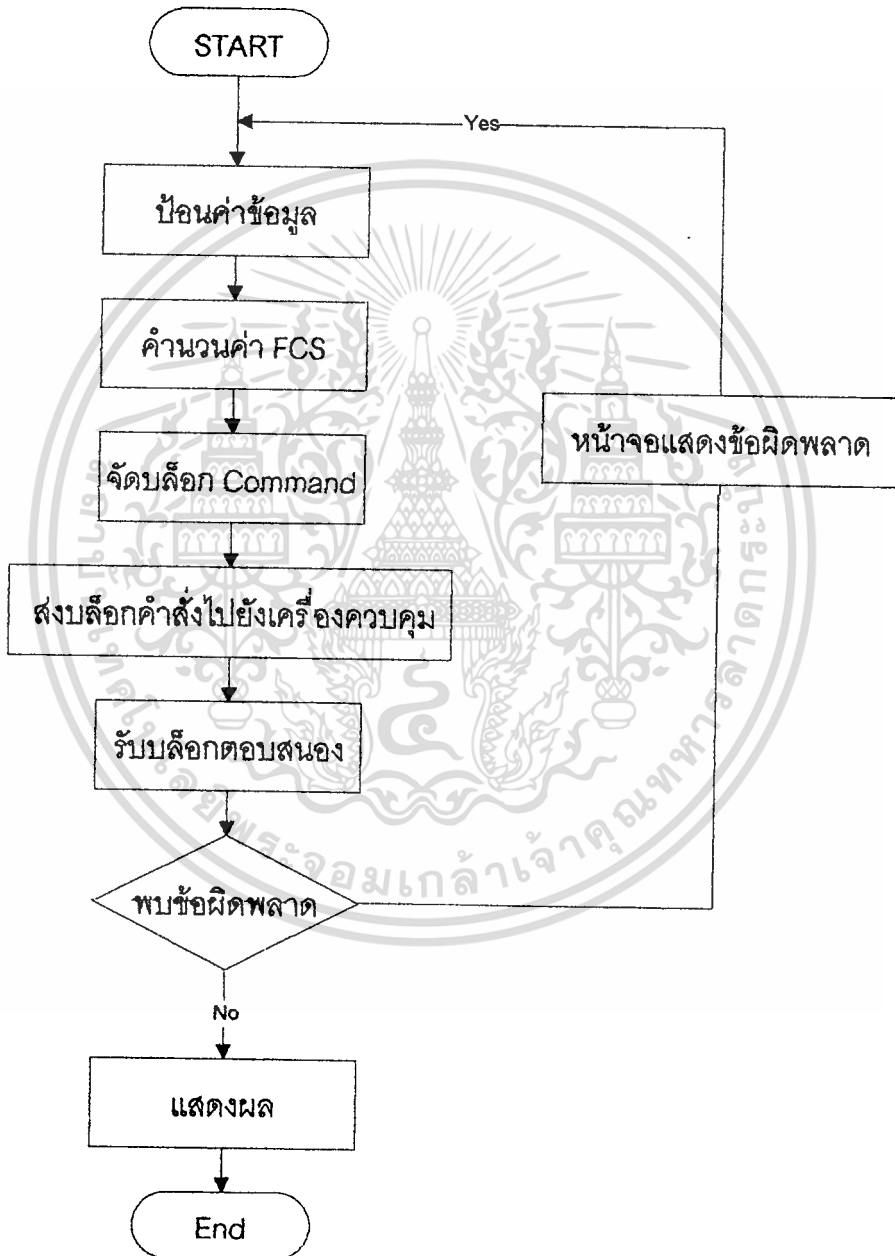
ตั้งค่าให้เป็น"0" เมื่อต้องการลบข้อมูลที่ค้างก่อนหน้านี้ออก( clear to send)โดยทั่วไปมักจะตั้งค่านี้ให้เปิดอยู่เสมอ

## บทที่ 4

### โปรแกรมที่พัฒนาขึ้น

โปรแกรมที่ทำการพัฒนาเพื่อใช้ในการเข้าถึงระบบโครงข่ายเขียนจากโปรแกรม Microsoft Visual Basic 5.0 เนื่องจากมีการใช้งานง่าย เอื้ออำนวยประโยชน์ต่อการใช้งาน

#### 4.1 ขั้นตอนการทำงานของโปรแกรม



รูปที่ 4.1 แสดงขั้นตอนการทำงานของโปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับกรรมการแข่งขันเพื่อการศึกษาเท่านั้น เมื่ออนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 4.2 โปรแกรม

### Private Sub Combo1\_Change()

```

OK.Enabled = True
If Combo1.Text = "" Then
OK.Enabled = False
End If

```

End Sub

### Private Sub Command1\_Click()

```

'Other Command
List3.Clear

OK.Enabled = True
List2.Visible = False
List2.Clear
Label7.Visible = True
Label8.Visible = False
Combo1.Text = ""
Combo1.Visible = True
Combo1.SetFocus
List1.Selected(0) = False
List1.Selected(1) = False
List1.Selected(2) = False
List1.Selected(3) = False
List1.Selected(4) = False
List1.Selected(5) = False
List1.Visible = False
Text1.Visible = False
Text1.Text = ""
Text2.Visible = False
Text2.Text = ""
Text5.Visible = False
Text5.Text = ""
Text6.Visible = False

```

**Text8.Text = ""**

**Text8.Visible = False**

**Option1.Visible = False**

**Option2.Visible = False**

**Option1.Value = False**

**Option2.Value = False**

**Label7.Visible = True**

**OK.Enabled = True**

**Combo1.SetFocus**

**End Sub**

**Private Sub Command2\_Click()**

**'CLEAR**

**List3.Clear**

**OK.Enabled = True**

**List2.Visible = False**

**List2.Clear**

**List3.Visible = False**

**List3.Clear**

**Combo1.Visible = False**

**List1.Selected(0) = False**

**List1.Selected(1) = False**

**List1.Selected(2) = False**

**List1.Selected(3) = False**

**List1.Selected(4) = False**

**List1.Selected(5) = False**

**Label7.Visible = False**

**List1.Visible = True**

**Text1.Visible = True**

**Text2.Visible = True**

**Text5.Visible = True**

**Text6.Visible = True**

**Option1.Visible = True**

**Option2.Visible = True**

เอกสารนี้เป็นเอกสารลิขสิทธิ์สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Text1.Text = ""
Text2.Text = ""
Text5.Text = ""
Text6.Text = ""
Combo1.Text = ""
Option1.Value = False
Option2.Value = False
OK.Enabled = True

```

**End Sub**

**Private Sub Command3\_Click()**

End

**End Sub**

**Private Sub Form\_Load()**

```

Text8.Visible = False
Label8.Visible = False
ProgressBar1.Visible = False
Label7.Visible = False
Combo1.Visible = False
List2.Visible = False
OK.Enabled = True

```

**End Sub**

**Private Sub List1\_Click()**

```

If List1.Selected(0) = True Then
Text2.ToolTipText = "(00-27) valid value"
ElseIf List1.Selected(1) = True Then
Text2.ToolTipText = "(0000-4095) valid value"
ElseIf List1.Selected(2) = True Then
Text2.ToolTipText = "(00-99) valid value"

```

```

ElseIf List1.Selected(3) = True Then

```

```

Text2.ToolTipText = "(00-236) valid value"

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามแก้ไขเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Elseif List1.Selected(4) = True Then
Text2.ToolTipText = "(00-64) valid value"
Elseif List1.Selected(5) = True Then
Text2.ToolTipText = "(237-255) valid value"
End If
Text1.Visible = True
Text1.Text = ""
Text1.SetFocus
Text2.Visible = True
Text2.Text = ""
Text5.Visible = False
Text5.Text = ""
Text6.Visible = False
Text6.Text = ""

Text8.Text = ""
Text8.Visible = False
Option1.Value = False
Option2.Value = False
List2.Clear
List2.Visible = False

End Sub

Private Sub OK_Click()

Dim command As String, area As String
Dim r As String, rec As String
Dim m As Integer, w As Integer, w1 As Integer, y As Long, q As Integer, q1 As Integer,
    p As Integer, line As String
Dim g(1 To 4) As String
Dim t(5000) As String
Dim rs As String, c As String

unit$ = Text1.Text
firstadd$ = Text2.Text
Data$ = Text6.Text

```

```

z = Len(Text5.Text)
Do While z < 4 And z > 0
    Text5.Text = "0" & Text5.Text
    z = z + 1
Loop

noword$ = Text5.Text

'Open Port COM1
If MSCComm1.PortOpen = False Then
    MSCComm1.PortOpen = True
End If

If Option1.Value = True Then
    Data$ = ""
ElseIf Option2.Value = True Then
    noword$ = ""
End If

'case Read command
If Option1.Value And List1.Selected(0) = True Then
    code$ = "RJ"
    area$ = "AR"
    Text1.Visible = True
    Text2.Visible = True

    q = Val(Right$(Text2.Text, 2))
    c$ = Right$(Text2.Text, 2)

ElseIf Option1.Value And List1.Selected(1) = True Then
    code$ = "RD"
    area = "DM"
    Text1.Visible = True
    Text2.Visible = True

    q = Val(Text2.Text)
    c$ = Text2.Text

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

ElseIf Option1.Value And List1.Selected(2) = True Then
    code$ = "RH"
    area$ = "HR"
Text1.Visible = True
    Text2.Visible = True

    q = Val(Right$(Text2.Text, 2))
    c$ = Right$(Text2.Text, 2)

ElseIf Option1.Value And List1.Selected(3) = True Then
    code$ = "RR"
    area$ = "IR"
Text1.Visible = True
    Text2.Visible = True

    q = Val(Right$(Text2.Text, 3))
    c$ = Right$(Text2.Text, 3)
ElseIf Option1.Value And List1.Selected(4) = True Then
    code$ = "RL"
    area$ = "LR"
Text1.Visible = True
    Text2.Visible = True

    q = Val(Right$(Text2.Text, 2))
    c$ = Right$(Text2.Text, 2)
ElseIf Option1.Value And List1.Selected(5) = True Then
    code$ = "RR"
    area$ = "SR"
Text1.Visible = True
    Text2.Visible = True

    q = Val(Right$(Text2.Text, 3))
    c$ = Right$(Text2.Text, 3)
End If

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

'case Write command
If Option2.Value And List1.Selected(0) = True Then
    code$ = "WJ"
ElseIf Option2.Value And List1.Selected(1) = True Then
    code$ = "WD"
ElseIf Option2.Value And List1.Selected(2) = True Then
    code$ = "WH"
ElseIf Option2.Value And List1.Selected(3) = True Then
    code$ = "WR"
ElseIf Option2.Value And List1.Selected(4) = True Then
    code$ = "WL"
ElseIf Option2.Value And List1.Selected(5) = True Then
    code$ = "WR"
End If

```

```

If List1.Visible = True Then
    i$ = "@" & unit$ & code$ & firstadd$ & noword$ & Data$
ElseIf List1.Visible = False Then
    i$ = Combo1.Text
End If

```

FCS

```

n% = 0
For x = 1 To Len(i$)
    n% = n% Xor Asc(Mid(i$, x, 1))
fcs = Hex(n%)
Next x

command$ = i$ & fcs & "*"

```

```

If Right$(command$, 1) = "*" Then

```

```

    OK.Enabled = True

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

End If

'Output or Send Command

MSComm1.Output = command\$ & Chr\$(13) & Chr\$(10)

'Delay Time

ret = DoEvents()

For am = 0 To 2000

  ProgressBar1.Value = am

Next am

'Input or receive

r1\$ = MSComm1.Input

If Asc(Right\$(r1\$, 2)) <> 42 Then

  r1\$ = Left\$(r1\$, Len(r1\$) - 3)

  List3.AddItem (r1\$)

  OK.Enabled = True

  OK.Caption = "Next"

ElseIf Asc(Right\$(r1\$, 2)) = 42 Then

  List3.AddItem (r1\$)

  r\$ = Trim\$(List3.List(0) & List3.List(1) & List3.List(2) & List3.List(3) & List3.List(4)

    & List3.List(5) & List3.List(6) & List3.List(7))

    OK.Caption = "OK"

  OK.Enabled = False

  Label7.Visible = False

End If

l = Len(r\$)

rec\$ = Mid\$(r\$, 6, 2)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
If rec$ = "00" Then
```

```
w = 1
```

```
w1 = 1
```

```
y = 0
```

```
rs$ = Mid$(r$, 8, 1 - 10)
```

```
line = Fix(Len(rs$) / 4)
```

```
If Option2.Value = True And rec$ = "00" Then
```

```
List2.Visible = False
```

```
List2.Text = ""
```

```
MsgBox " SUCCESS", vbOKOnly, " Response Code" "Write Sucessly
```

```
Form1.Show
```

```
Elseif Option1.Value = fales Then
```

```
List2.Visible = True
```

```
For q1 = 1 To (Len(r$) / 40)
```

```
If Len(rs$) < 40 Then
```

```
List2.AddItem (r$)
```

```
End If
```

```
List2.AddItem (" " & (Mid$(rs$, w1, 40)))
```

```
w1 = w1 + 40
```

```
Next
```

```
Combo1.Visible = False
```

```
Elseif Option1.Value = True Then
```

```
Combo1.Visible = False
```

```
For q = q To (Val(Text2.Text) + (line - 1))
```

```
t(y) = Mid$(rs$, w, 4)
```

```
w = w + 4
```

```
If List1.Selected(0) And Val(q) < 10 Then
```

```
c$ = "0" & q
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับบริการเชิงงานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
ElseIf List1.Selected(4) And Val(q) > 9 Then
```

```
    c$ = q
```

```
    Text2.ToolTipText = "(00-64) valid value"
```

```
End If
```

```
If List1.Selected(5) Then
```

```
    Text2.ToolTipText = "(237-255) valid value"
```

```
    c$ = q
```

```
End If
```

```
For m = 1 To 4
```

```
    g(m) = Mid$(t(y), m, 1)
```

```
    If g(m) = "0" Then
```

```
        g(m) = "0 0 0 0"
```

```
    ElseIf g(m) = "1" Then
```

```
        g(m) = "0 0 0 1"
```

```
    ElseIf g(m) = "2" Then
```

```
        g(m) = "0 0 1 0"
```

```
    ElseIf g(m) = "3" Then
```

```
        g(m) = "0 0 1 1"
```

```
    ElseIf g(m) = "4" Then
```

```
        g(m) = "0 1 0 0"
```

```
    ElseIf g(m) = "5" Then
```

```
        g(m) = "0 1 0 1"
```

```
    ElseIf g(m) = "6" Then
```

```
        g(m) = "0 1 1 0"
```

```
    ElseIf g(m) = "7" Then
```

```
        g(m) = "0 1 1 1"
```

```
    ElseIf g(m) = "8" Then
```

```
        g(m) = "1 0 0 0"
```

```
    ElseIf g(m) = "9" Then
```

```
        g(m) = "1 0 0 1"
```

```
    ElseIf g(m) = "A" Then
```

```
        g(m) = "1 0 1 0"
```

```
    ElseIf g(m) = "B" Then
```

```
        g(m) = "1 0 1 1"
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Text2.ToolTipText = "(00-27) valid value"
ElseIf List1.Selected(0) And Val(q) > 9 Then
    c$ = q
    Text2.ToolTipText = "(00-27) valid value"
End If
If List1.Selected(1) And Val(q) < 10 Then
    c$ = "000" & q
    Text2.ToolTipText = "(0000-4095) valid value"
ElseIf List1.Selected(1) And Val(q) > 9 And Val(q) < 100 Then
    c$ = "00" & q
    Text2.ToolTipText = "(0000-4095) valid value"
ElseIf List1.Selected(1) And Val(q) > 99 And Val(q) < 1000 Then
    c$ = "0" & q
    Text2.ToolTipText = "(0000-4095) valid value"
ElseIf List1.Selected(1) And Val(q) > 999 And Val(q) < 10000 Then
    c$ = q
    Text2.ToolTipText = "(0000-4095) valid value"
End If
If List1.Selected(2) And Val(q) < 10 Then
    c$ = "0" & q
ElseIf List1.Selected(2) And Val(q) > 9 Then
    c$ = q
End If
If List1.Selected(3) And Val(q) < 10 Then
    c$ = "00" & q
    Text2.ToolTipText = "(000-236) valid value"
ElseIf List1.Selected(3) And Val(q) > 9 And Val(q) < 100 Then
    c$ = "0" & q
    Text2.ToolTipText = "(000-236) valid value"
ElseIf List1.Selected(3) And Val(q) > 99 And Val(q) < 1000 Then
    c$ = q
    Text2.ToolTipText = "(000-236) valid value"
End If
If List1.Selected(4) And Val(q) < 10 Then
    c$ = "0" & q
    Text2.ToolTipText = "(00-64) valid value"

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมีเหตุใดแต่สิ่งเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

ElseIf g(m) = "C" Then
    g(m) = "1 1 0 0"
ElseIf g(m) = "D" Then
    g(m) = "1 1 0 0"
ElseIf g(m) = "E" Then
    g(m) = "1 1 1 0"
ElseIf g(m) = "F" Then
    g(m) = "1 1 1 1"
End If

```

```

If m = 4 Then
    g(m) = area$ & c$ & " " & g(1) & " " & g(2) & " " & g(3) & " " & g(4) & " " &
    t(y)
    If Option1.Value = True Then
        List2.Visible = True
        List2.AddItem (g(m))
    End If
End If
Next
y = y + 1
Next
End If

```

'Check Error

```

ElseIf rec$ = "01" Then
    MsgBox "Can't able to Exacution", vbOKOnly, "Error 01"
    Form1.Show

```

```

ElseIf rec$ = "02" Then
    MsgBox "Can't able to Exacution", vbOKOnly, "Error 02"
    Form1.Show

```

```

ElseIf rec$ = "03" Then
    MsgBox "Can't able to Exacution", vbOKOnly, "Error 03"
    Form1.Show

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับวิชาการแข่งขันเพื่อการศึกษาเท่านั้น เมื่ออนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

ElseIf rec$ = "04" Then
    MsgBox "Data Overflow", vbOKOnly, "Error 04"
    Form1.Show

```

```

ElseIf rec$ = "0B" Then
    MsgBox "Can't able to Exacution", vbOKOnly, "Error 0B"
    Form1.Show

```

```

ElseIf rec$ = "0C" Then
    MsgBox "Can't able to Exacution", vbOKOnly, "Error 0C"
    Form1.Show

```

```

ElseIf rec$ = "0D" Then
    MsgBox "Can't able to Exacution", vbOKOnly, "Error 0D"
    Form1.Show

```

```

ElseIf rec$ = "10" Then
    MsgBox "Can't able to Exacution", vbOKOnly, "Error 10"
    Form1.Show

```

```

ElseIf rec$ = "11" Then
    MsgBox "Stop bit not detecteed", vbOKOnly, "Error 11"
    Form1.Show

```

```

ElseIf rec$ = "12" Then
    MsgBox "Overrun", vbOKOnly, "Error 12"
    Form1.Show

```

```

ElseIf rec$ = "13" Then
    MsgBox "Chack sum Error", vbOKOnly, "Error 13"
    Form1.Show

```

```

ElseIf rec$ = "14" Then
    MsgBox "Commamd Format Error", vbOKOnly, "Error 14"
    Form1.Show

```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**ElseIf rec\$ = "15" Then**

**MsgBox "Incorrect Data", vbOKOnly, "Error 15"**

**Form1.Show**

**ElseIf rec\$ = "16" Then**

**MsgBox "Struction Not Found!", vbOKOnly, "Error 16"**

**Form1.Show**

**ElseIf rec\$ = "17" Then**

**MsgBox "Can't able to execution", vbOKOnly, "Error 17"**

**Form1.Show**

**ElseIf rec\$ = "18" Then**

**MsgBox "Can't able to Exacution", vbOKOnly, "Error 18"**

**Form1.Show**

**ElseIf rec\$ = "20" Then**

**MsgBox "Can't able to generation I/O table", vbOKOnly, "Error 20"**

**Form1.Show**

**ElseIf rec\$ = "21" Then**

**MsgBox "Error in PC's CPU", vbOKOnly, "Error 21"**

**Form1.Show**

**ElseIf rec\$ = "22" Then**

**MsgBox "Specified memory Unit not Exist", vbOKOnly, "Error 22"**

**Form1.Show**

**ElseIf rec\$ = "23" Then**

**MsgBox "Can't able to Exacution", vbOKOnly, "Error 23"**

**Form1.Show**

**ElseIf rec\$ = "A0" Then**

**MsgBox "Parity Error in transmit Data", vbOKOnly, "Error A0"**

**Form1.Show**

**ElseIf rec\$ = "A1" Then**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งยังมีให้เปลี่ยนแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
MsgBox "Framing Error in transmit Data", vbOKOnly, "Error A1"
```

```
Form1.Show
```

```
ElseIf rec$ = "A2" Then
```

```
MsgBox "Overrun in transmit Data", vbOKOnly, "Error A2"
```

```
Form1.Show
```

```
ElseIf rec$ = "A3" Then
```

```
MsgBox "Checksum Error in transmit Data", vbOKOnly, "Error A3"
```

```
Form1.Show
```

```
ElseIf rec$ = "A4" Then
```

```
MsgBox "Format Error in transmit Data", vbOKOnly, "Error A4"
```

```
Form1.Show
```

```
ElseIf rec$ = "A5" Then
```

```
MsgBox "Data number Error in transmit Data", vbOKOnly, "Error A5"
```

```
Form1.Show
```

```
ElseIf rec$ = "A8" Then
```

```
MsgBox "Frame length Error in transmit Data", vbOKOnly, "Error 01"
```

```
Form1.Show
```

```
ElseIf rec$ = "B0" Then
```

```
MsgBox "Not execable", vbOKOnly, "Error 01"
```

```
Form1.Show
```

```
End If
```

```
End Sub
```

```
Private Sub Option1_Click()
```

```
'read click
```

เอกสารนี้เป็นเอกสารที่จัดทำขึ้นเพื่อใช้ในการเรียนการสอนเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น  $y = \text{Len}(\text{Text2.Text})$  วัตถุประสงค์เพื่อเผยแพร่เนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Do While y < 4 And y > 0
    Text2.Text = "0" & Text2.Text
    y = y + 1

```

```

Loop

```

```

firstadd$ = Text2.Text

```

```

Text5.Text = ""

```

```

Text5.Visible = True

```

```

Text5.SetFocus

```

```

Text6.Text = ""

```

```

Text6.Visible = False

```

```

Text8.Text = ""

```

```

Text8.Visible = False

```

```

Option2.Value = False

```

```

Combo1.Text = ""

```

```

Combo1.Visible = False

```

```

List3.Clear

```

```

List3.Visible = False

```

```

OK.Enabled = True

```

```

Label8.Visible = False

```

```

End Sub

```

```

Private Sub Option2_Click()

```

```

    'write click

```

```

        y = Len(Text2.Text)

```

```

    Do While y < 4 And y > 0

```

```

        Text2.Text = "0" & Text2.Text

```

```

        y = y + 1

```

```

    Loop

```

```

        Text6.Visible = True

```

```

        Text8.Visible = True

```

```

        Text5.Text = ""

```

```

        Text5.Visible = False

```

```

        Text6.Text = ""

```

```

        Text6.SetFocus

```

```

        Option1.Value = False

```

เอกสารนี้เป็นเอกสารที่ให้บริการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาติให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น

```

Combo1.Text = ""
Combo1.Visible = False
OK.Enabled = True
Label8.Visible = True
List2.Clear
List3.Clear
List3.Visible = False
List2.Visible = False

```

**End Sub**

**Private Sub Text1\_Change()**

```

If Len(Text1.Text) = 2 Then
    Text2.SetFocus
End If

```

**End Sub**

**Private Sub Text2\_Change()**

```

m = Len(Text1.Text)
Do While m < 2 And m > 0
    Text1.Text = "0" & Text1.Text
    m = m + 1
Loop
unit$ = Text1.Text

```

**End Sub**

**Private Sub Text3\_Change()**

```

If Len(Text3.Text) = 2 Then
    Text4.SetFocus
End If

```

**End Sub**

**Private Sub Text6\_Change()**

```
Data$ = Text6.Text
```

```
If (Len(Text6.Text)) Mod 4 = 0 Then
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามเผยแพร่เปลี่ยนแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Text8.Text = " " & Fix(Len(Text6.Text) / 4) & " "
```

```
End If
```

```
End Sub
```

```
Private Sub Text7_Change()
```

```
Label7.Visible = False
```

```
End Sub
```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทที่ 5 ผลการทดลอง

จากการทดลองโดยผ่านคำสั่งจากโปรแกรม Microsoft Visual Basic ที่ได้พัฒนาขึ้นทางหน้าจอคอมพิวเตอร์ไปยัง PLC ที่อยู่ในระบบ LAN ที่มีโทโปโลยีแบบ Token Ring โดยใช้การติดต่อสื่อสารกับคอมพิวเตอร์ที่ทำหน้าที่ควบคุมระบบแบบ Host Link

### 5.1 อธิบายการใช้งานโปรแกรมที่พัฒนาขึ้น

#### 5.1.1 เลือกการอ่านหรือเขียนข้อมูลในหน่วยความจำ PLC

##### 1. เลือกพื้นที่หน่วยความจำ

พื้นที่หน่วยความจำที่ใ้รับย่อยๆ โดยทั่วไปใน PLC ได้แก่

- AR AREA คือ Auxialary Relay Area หรือ หน่วยความจำเสริม
- DM AREA คือ Data Memory Area เป็นพื้นที่หน่วยความจำหลักของ PLC
- HR AREA คือ พื้นที่ที่ยังคงค่าสถานะต่างๆ ไว้แม้ยามปิดสวิตซ์ไฟ
- IR AREA คือ Internal Relay Area เป็นพื้นที่หน่วยความจำภายในของ PLC
- LR AREA คือ Link Relay Area ส่วนมากนิยมใช้เป็นที่ติดต่อสื่อสารกับ PLC ตัวอื่นๆ
- SR AREA คือ Spacial Relay Area เป็นพื้นที่หน่วยความจำพิเศษ

##### 2. เลือก Unit ของ PLC ที่ต้องการติดต่อ

##### 3.เลือก First Address

First Address คือตำแหน่งพื้นที่หน่วยความจำแรกที่ต้องการทำการอ่านหรือเขียน

##### 4.เลือกที่จะทำการอ่านหรือเขียนข้อมูล

- กรณีถ้าเลือกการอ่านจะปรากฏช่องให้เติมจำนวนของ word ที่ต้องการอ่าน
- กรณีถ้าเลือกการเขียนจะปรากฏช่องให้เติมจำนวนข้อมูลซึ่งสามารถเขียนได้มากที่สุด 29 ช่อง

#### 5.1.2 แสดงการเรียก Command อื่นๆ

การเรียกใช้คำสั่งอื่นๆ จำเป็นที่จะต้องทราบรูปแบบโปรโตคอลที่ใช้กับเครื่อง PLC นั้นๆ ซึ่งจะแตกต่างกันออกไปตามชนิดและรุ่นของ PLC

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 5.2 ผลการอ่านข้อมูลจากพื้นที่หน่วยความจำ PLC

PLC AREA OPERATION

**AREA**

AR AREA ▲  
**DM AREA**  
 HR AREA  
 IR AREA ▼

UNIT

FIRSTADDRESS

OK

OTHER

CLEAR

CANCEL

**OPERATION**

READ     WRITE

NO.WORD

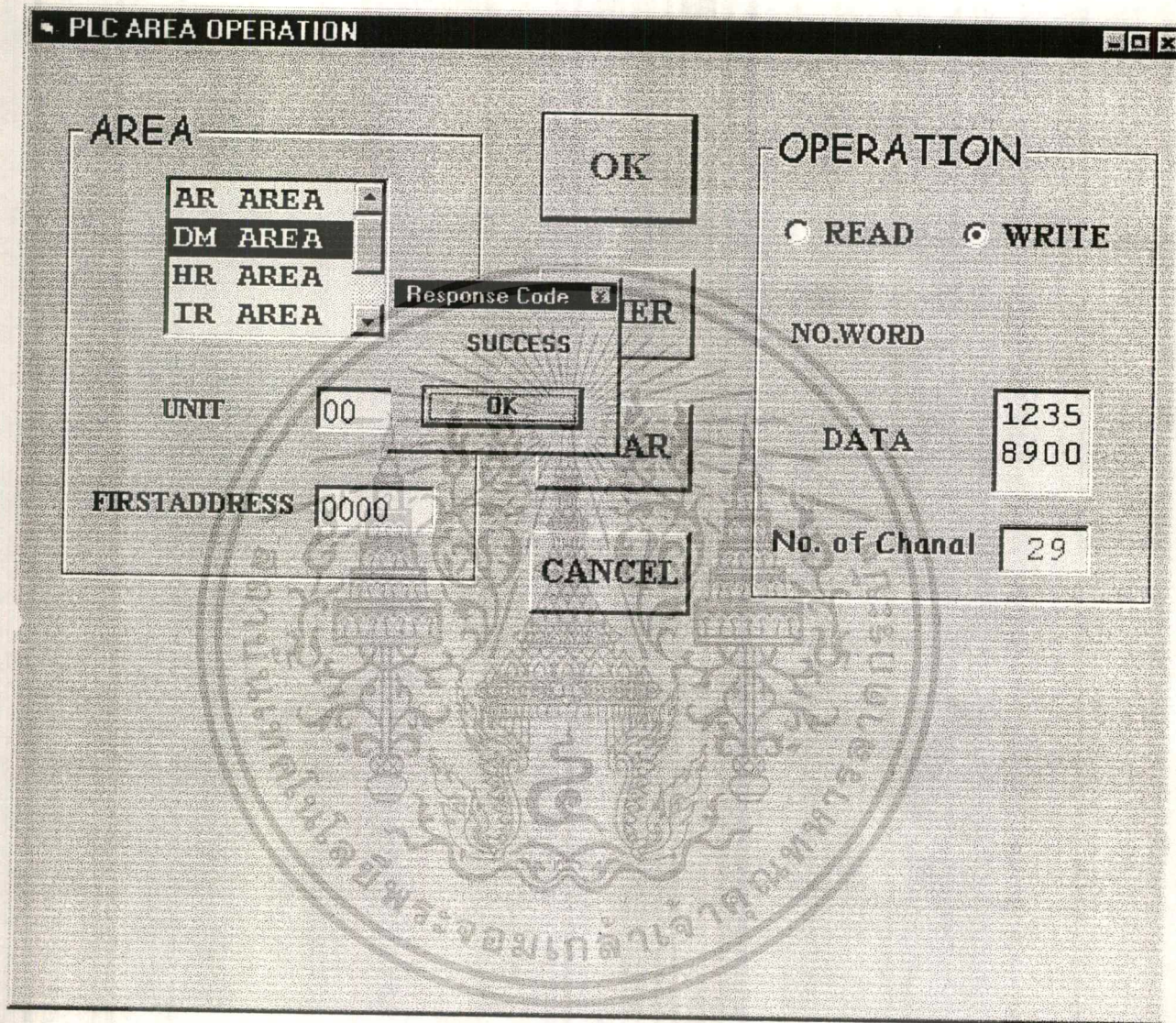
DATA

DM0108	0	1	0	1	1	0	0	0	0	1	0	1	0	1	1	0	5856	▲
DM0109	1	0	0	1	0	0	1	1	0	0	1	1	0	0	1	1	9333	
DM0110	0	0	1	1	0	0	1	0	0	0	1	0	0	0	0	1	3221	
DM0111	0	0	0	1	0	0	0	1	0	0	1	0	0	0	0	1	1121	
DM0112	0	0	0	1	0	1	0	1	0	1	0	0	0	1	0	1	1545	▼
DM0113	0	1	0	0	0	1	1	0	0	1	0	1	0	1	0	0	4654	
DM0114	0	1	1	0	0	0	1	1	0	0	0	1	0	0	1	1	6313	
DM0115	0	0	0	1	0	0	1	1	0	0	0	1	0	0	1	1	1313	▼

รูปที่ 5.1 แสดงการอ่านข้อมูลจากพื้นที่หน่วยความจำของ PLC

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

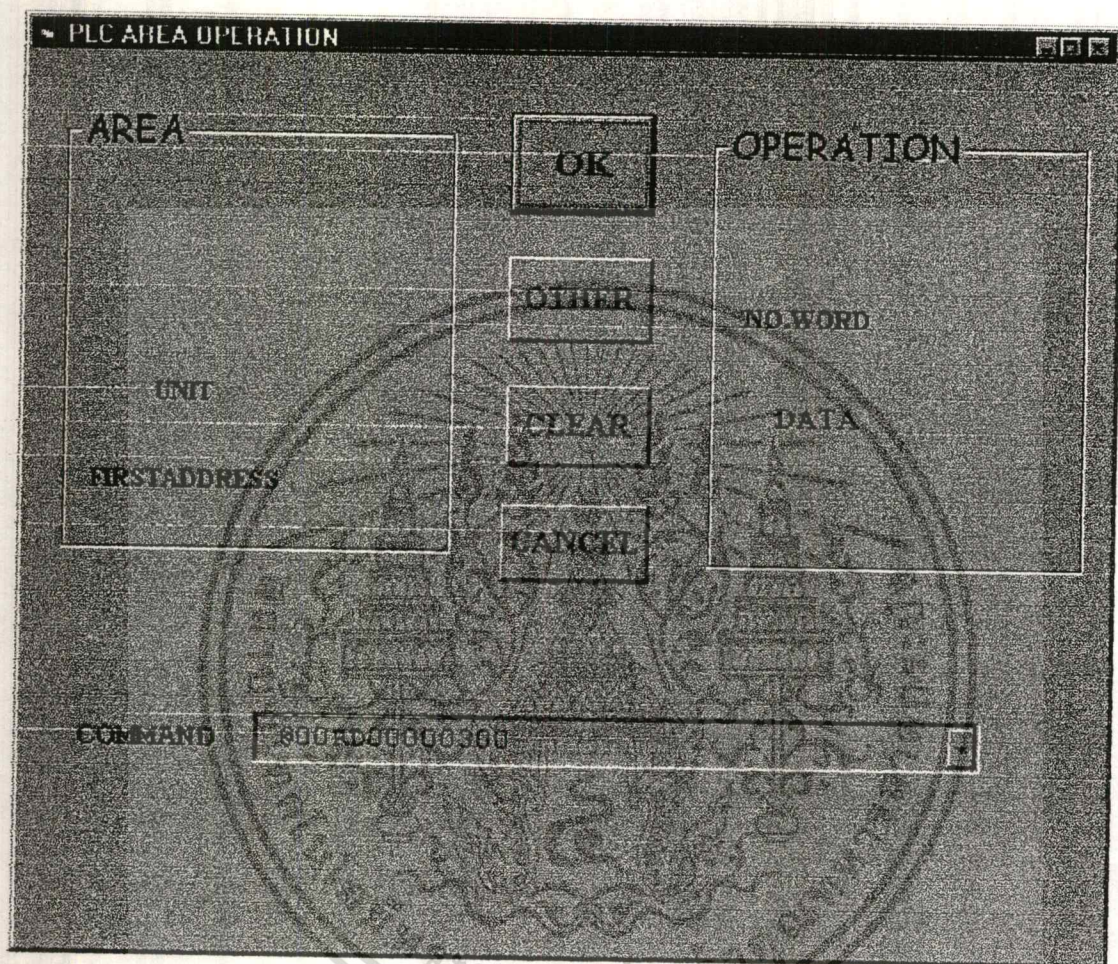
### 5.3 ผลการเขียนข้อมูลลงในพื้นที่หน่วยความจำ PLC



รูปที่ 5.2 แสดงการเขียนข้อมูลลงในพื้นที่หน่วยความจำของ PLC

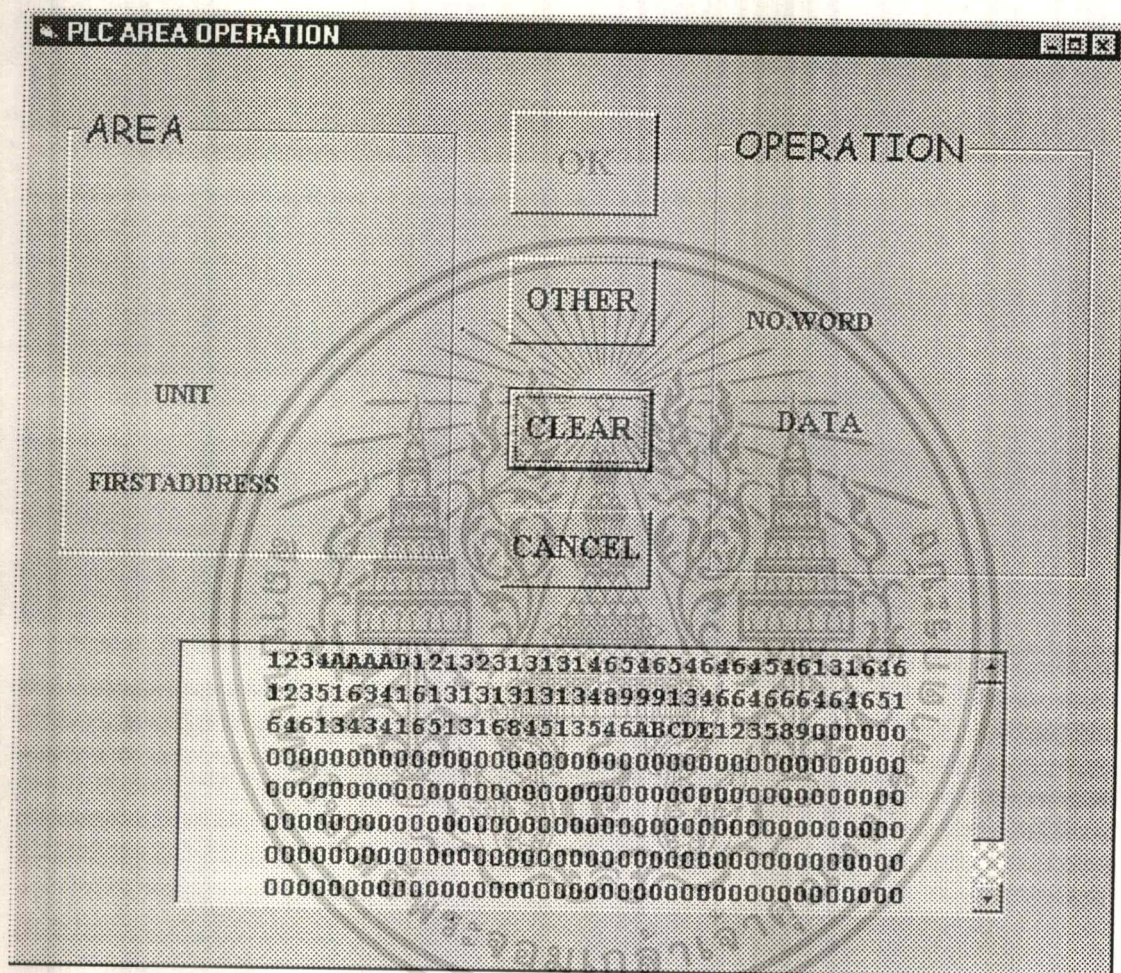
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 5.4 ผลการเรียก Command อื่นๆ



รูปที่ 5.3 แสดงการเรียก Command อื่นๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5.4 แสดงผลทางหน้าจอจากการเรียก Command อื่นๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทที่ 6 บทสรุป

### แนวทางการพัฒนาต่อ

- สามารถขยายขนาดของโครงข่ายโดยเพิ่มจำนวน Node ได้มากถึง 126 Node ต่อ 1 Loop และสามารถสื่อสารระหว่าง Loop โดยใช้ Bridge เป็นตัวเชื่อมต่อได้
- สามารถเชื่อมต่อโครงข่ายที่สร้างขึ้นกับอุปกรณ์ I/O เพื่อใช้ควบคุมกระบวนการต่างๆ ผ่านโปรแกรมที่พัฒนาขึ้นได้
- โปรแกรมที่พัฒนาขึ้นนี้ สามารถนำมาประยุกต์ใช้งานกับ PLC รุ่นอื่นๆ ได้ ถ้ารู้รูปแบบโปรโตคอล ของ PLC รุ่นนั้นๆ

### ปัญหาที่พบในการทำโครงการ

- เนื่องจากเทคโนโลยี LAN สำหรับการควบคุมโรงงานให้เป็นแบบอัตโนมัตินี้ยังถือว่าเป็นระบบที่ค่อนข้างใหม่ในเมืองไทยจึงหาแหล่งข้อมูลได้ยาก
- Hardware ที่ใช้ในการติดต่อผ่านใยแก้วนำแสงเป็น Hardware ใหม่จึงอาศัยเวลาในการศึกษานานและเมื่อเกิดปัญหาจะต้องติดต่อโดยตรงกับซึ่งใช้เวลานาน
- Hardware ไม่มีตัว Driver สำหรับการทำงานบน Windows
- Microsoft Visual Basic มีข้อจำกัดในจำนวนข้อมูลที่ใช้ในการติดต่อสื่อสาร

### สรุปโครงการ

โครงการนี้เป็นการเชื่อมต่อ LAN อย่างง่ายของ PLC ซึ่งเป็น LAN แบบ Token Ring ใช้เส้นใยแก้วนำแสงเป็นตัวนำสัญญาณ และได้ทำการพัฒนาโปรแกรมเพื่อให้ผู้ใช้สามารถอ่านหรือเขียนข้อมูลลงบน PLC ผ่านระบบปฏิบัติการ Microsoft Windows ซึ่งมีความสะดวกกว่าการใช้งานบน DOS โดยอาศัยโปรโตคอลในระบบ Host Link ในการสื่อสารระหว่างคอมพิวเตอร์กับ PLC และเลือกใช้โปรแกรม Microsoft Visual Basic เนื่องจากมีขั้นตอนน้อยและสะดวกในการใช้งาน

โครงการนี้มีประโยชน์โดยสามารถนำไปประยุกต์ใช้งานกับระบบควบคุมแบบอัตโนมัติและการพัฒนาอื่นๆ อีกทั้งยังเป็นการเพิ่มเติมความรู้และประสบการณ์แก่ผู้จัดทำอีกด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บรรณานุกรม

1. กิตติ ภัคดีวัฒนะกุล และ จำลอง ทรูอุตสาหะ, "Visual Basic 5 ฉบับโปรแกรมเมอร์", พิมพ์ครั้งที่ 2, กรุงเทพมหานคร, สำนักพิมพ์ไทยเจริญการพิมพ์, 2541.
2. ณรงค์ ต้นชีวะวงศ์, "ระบบ PLC (Programmable Logic Controller)", พิมพ์ครั้งที่ 1, กรุงเทพมหานคร, สำนักพิมพ์บริษัทประชาชนจำกัด
3. สุวิมล สิริชีวะภาค, ผศ.ดร., "โลคัลแอเรียเน็ตเวิร์ค", พิมพ์ครั้งที่ 1, ภาควิชาโทรคมนาคม, คณะวิศวกรรมศาสตร์, สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
4. สุทธิศักดิ์ พงศ์ธนาพานิช, "Visual Basic 5.0 Professional การใช้คำสั่งและคอนโทรล Active X", กรุงเทพมหานคร, สำนักพิมพ์ซีเอ็ดยูเคชั่น, 2541
5. OMRON Corporation SYSMAC (C1000H/C2000H) Programmable Controllers
6. OMRON Corporation SYSMAC (S3200-NSB11-E) SYSMAC NET Support Board
7. OMRON Corporation SYSMAC (C500-SNT-31) SYSMAC NET Link Unit
8. OMRON Corporation SYSMAC (S3200-NSUG4-00E) Local Bridge
9. OMRON Corporation SYSMAC (S3200-LSU03-01E) Line Server
10. OMRON Corporation SYSMAC (C-Series Rack PCs) Host Link Unit