สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

# ระบบทันเวลาพอดี

# JUST IN TIME SYSTEM

นายณัฐวุฒิ ศรีธรรมมา

นางสาวนันทิกา เกษมวัฒนา

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เลขหมู่.....
เลขทะเบียน 62131
วัน,เดือน,ปี 31 ก.ค. 2549

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

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมการวัดคุม

ภาควิชาวิศวกรรมการวัดคุม คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2547

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# JUST IN TIME SYSTEM

NATTAWUT SITUMMA

NANTIKA KASAMWATTANA

A THESIS SUBMITTED IN PARTIAL FULFILLMENT

OF THE REQUIREMENT FOR THE DEGREE OF

BACHELOR OF ENGINEERING IN INSTRUMENTATION ENGINEERING

DEPARTMENT OF INSTRUMENTATION ENGINEERING

FACULTY OF ENGINEERING

KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

2004

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**ภาควิชาวิศวกรรมการวัดคุม**

**คณะวิศวกรรมศาสตร์**

**สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง**

**ใบรับรองปริญญานิพนธ์**

.....

**หัวข้อปริญญานิพนธ์** ระบบทันเวลาพอดี

JUST IN TIME SYSTEM

**นักศึกษาผู้จัดทำ** นายณัฐวุฒิ ศรีธรรมมา **รหัสประจำตัว** 45045550

นางสาวนันทิกา เกษมวัฒนา **รหัสประจำตัว** 45015557

**ปริญญา** วิศวกรรมศาสตร์บัณฑิต

**สาขาวิชา** วิศวกรรมการวัดคุม

**ปีการศึกษา** 2547

| อาจารย์ผู้ควบคุมปริญญานิพนธ์ | ลายมือชื่อ |
|---|---|
| ผศ. ทวีผล ซื่อสัตย์ | |
| ดร. พงศ์ชัย นิลาศ | |

**วัน/เดือน/ปี ที่สอบ** วันจันทร์ที่ 23 มีนาคม พ.ศ.2548

**สถานที่สอบ** ณ ห้องสอบปริญญานิพนธ์ ภาควิชาวิศวกรรมการวัดคุม

**ภาควิชารับรองแล้ว**

**(รศ.ประสิทธิ์ จุลเสรีวงค์)**

**หัวหน้าภาควิชาวิศวกรรมการวัดคุม**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

| | | |
|---|---|---|
| **หัวข้อปริญญานิพนธ์** | ระบบทันเวลาพอดี | |
| | JUST IN TIME SYSTEM | |
| **นักศึกษาผู้จัดทำ** | นายณัฐวุฒิ | ศรีธรรมมา |
| | นางสาวนันทิกา | เกษมวัฒนา |
| **อาจารย์ที่ปรึกษา** | ผศ.ทวีพล | ซื่อสัตย์ |
| | ดร.พงศ์ชัย | นิลาศ |
| **ปีการศึกษา** | 2547 | |

## บทคัดย่อ

ในสภาวะปัจจุบันเราจะเห็นได้ว่าเวลาเป็นสิ่งสำคัญมากอย่างยิ่งต่อการดำเนินธุรกิจทางด้านอุตสาหกรรม การคิด การปรับปรุงกระบวนการต่างๆที่อยู่ในโรงงานอุตสาหกรรม ก็เพื่อ ให้ได้กำไรสูงสุด ผลิตภัณฑ์ที่ผลิตออกมาได้ตรงตามความต้องการของลูกค้า ในปริญญานิพนธ์นี้จึงได้นำเสนอระบบการผลิตแบบระบบ Just In Time ซึ่งระบบนี้ได้เข้ามามีบทบาทต่อการดำเนินงาน ในทางด้านอุตสาหกรรม เพื่อทดสอบให้เห็นว่าระบบนี้ สามารถลดต้นทุนในการผลิตได้ เช่น ประหยัดค่าใช้จ่ายในการสร้างโรงเก็บวัตถุหรือผลิตภัณฑ์ต่างๆ การดึงวัตถุที่จะต้องใช้ในช่วงเวลาที่พอดี การใช้พนักงาน 1 คนต่องานหลายอย่าง เป็นต้น สำหรับปริญญานิพนธ์นี้ได้มีการจัดทำการควบคุมแบบ Just In Time โดยการสมมุติการทำงานจากโปรแกรมควบคุมสายการผลิต หนึ่งๆ คือ มีการสั่งวัตถุผ่านระบบเครือค่าย โปรแกรมการสั่งวัตถุ ป้อนวัตถุที่ใช้ในการะบวนการ สร้างวัตถุจากการคัดแยกวัตถุตามการสั่ง ซึ่งจะช่วยให้เรามองเห็นลักษณะการทำงานได้ง่ายขึ้นว่า ในระบบ Just In Time มีกระบวนการอย่างไร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

| | |
|---|---|
| **Thesis Title** | Automatic PID Controller Parameter Analyzer |
| **Authors** | Mr.Nattawut Situmma |
| | Miss.Nantika Kasamwattana |
| **Thesis Advisor** | Asst.Prof. Taweepol Suesat |
| | Dr.Pongchai Nilas |
| **Year** | 2004 |


## ABSTRACT

Nowadays, the operating time is very important for manufacturing process business. To gain success in profit, productivity and demand of customer. This thesis presents the Just – In – Time system that can reduce operating time. Furthermore, this system can economize the inventory cost and also worth for resource planning. The case study of this system consists of the conveyer and color object detection system which linked to the automatic warehouse through an industrial network and Ethernet LAN.



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# กิตติกรรมประกาศ

ปริญญานิพนธ์ฉบับนี้สำเร็จลุล่วงได้ด้วยดีเพราะได้รับความเมตตา จากอาจารย์ทวีพล ซื่อสัตย์ และอาจารย์พงศ์ชัย นิลาศ ที่ได้ให้คำแนะนำแก่ผู้วิจัยตลอดมา อีกทั้งยังเอื้อเฟื้ออุปกรณ์และเครื่องมือต่าง ๆ ในการทำปริญญานิพนธ์นี้ ผู้วิจัยรู้สึกซาบซึ้งและขอกราบขอบพระคุณเป็นอย่างสูง

ขอขอบพระคุณ อาจารย์ภาควิชาวิศวกรรมการวัดคุมทุกท่าน ที่ให้คำแนะนำอันเป็นประโยชน์ต่อทำปริญญานิพนธ์ฉบับนี้

และที่ลืมเสียมิได้คือ ขอกราบขอบพระคุณคุณแม่ อันเป็นที่รักยิ่ง ที่สนับสนุนและเป็นแรงบันดาลใจในการทำปริญญานิพนธ์ฉบับนี้

คุณค่าและประโยชน์อันพึงมีจากปริญญานิพนธ์ฉบับนี้ ผู้วิจัยขอมอบแด่ผู้มีพระคุณทุกท่าน

คณะผู้จัดทำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# สารบัญ


หน้า

**บทคัดย่อภาษาไทย** ..... I

**บทคัดย่อภาษาอังกฤษ** ..... II

**กิตติกรรมประกาศ** ..... III

**สารบัญ** ..... IV

**สารบันตาราง** ..... VII

**สารบัญภาพ** ..... VIII

**บทที่ 1 บทนำ** ..... 1

1.1 ความเป็นมาและเหตุจูงใจของการวิจัย ..... 1

1.2 วัตถุประสงค์ของปริญญานิพนธ์ ..... 1

1.3 ขอบเขตของปริญญานิพนธ์ ..... 1

1.4 รายละเอียดของปริญญานิพนธ์ ..... 2

**บทที่ 2 ทฤษฎีระบบทันเวลาพอดี** ..... 3

2.1 กล่าวนำ ..... 3

2.2 วิธีการผลิตแบบ Just In Time ..... 3

2.3 การลดขนาดของการผลิต ..... 5

2.4 การลดเวลาเตรียมการผลิต ..... 7

2.5 ผลกระทบของวิธีการผลิตแบบ Just In Time ..... 8

2.6 การลดความสูญเปล่าของระบบ Just In Time ..... 13

2.7 การขจัดความสูญเปล่าให้หมดไป ..... 16

**บทที่ 3 ทฤษฎีการติดต่อสื่อสารและฐานข้อมูล** ..... 17

3.1 กล่าวนำ ..... 17

3.2 การติดต่อสื่อสารและส่งผ่านข้อมูล ..... 17

3.3 การติดต่อสื่อสารข้อมูลทั่วไป ..... 18

3.4 รูปแบบการสื่อสารข้อมูล ..... 18



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# สารบัญ (ต่อ)


หน้า

3.5 มาตรฐานในการสื่อสารข้อมูลแบบอุนกรม....21

3.6 Wring Ports....24

3.7 รูปแบบการสื่อสารข้อมูลของเครื่องควบคุมแบบตรรกะที่โปรแกรมได้....25

3.8 การกำหนดตัวพารามิเตอร์ของ Host Iing....30

3.9 การส่งสัญญาณเพื่อควบคุมการทำงาน....31

3.10 รูปแบบของบล็อก (BLOCK FORMAT)....31

3.11 ระบบเน็ทเวิร์ก....33

3.12 ลักษณะพิเศษของวิธีการส่งแบบเบสแบนด์นั้นจะเป็นดังต่อไปนี้....38

3.13 ระบบไคลเอนต์ / เซอร์ฟเวอร์....39

3.14 การสร้างฐานข้อมูลด้วยโปรแกรม Access....44

3.15 รูปแบบการติดต่อกับฐานข้อมูลด้วย Visual Basic....48

3.16 Visual Basic 6....50

3.17 ฐานข้อมูล Data Base....54

**บทที่ 4 ขั้นตอนและวิธีการดำเนินงาน....63**

4.1 กล่าวนำ....63

4.2 ศึกษาหลักการผลิตแบบ Just In Time และออกแบบจำลองกระบวนการผลิต....63

4.3 ศึกษาหลักการทำงาน วิธีใช้และตรวจสอบการทำงานเครื่อง Warehouse....68

4.4 ศึกษาหลักการทำงาน วิธีการใช้งานชุดสายพานลำเลียง....68

4.5 ศึกษาความสัมพันธ์ของอุปกรณ์ต่างๆ ที่ทำงานกันในระบบ....70

4.6 ศึกษาเกี่ยวกับ Visual Basic....70

**บทที่ 5 การทดสอบโครงงาน....71**

5.1 กล่าวนำ....71

5.2 การทดสอบในส่วนสายการผลิตย่อย....72

5.3 การทดลองในส่วนสายการผลิตหลัก....78



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# สารบัญ (ต่อ)


หน้า

**บทที่ 6 สรุปโครงงาน.....86**

6.1 สรุปโครงงาน.....86

6.2 ปัญหาที่พบ.....86

6.3 ข้อเสนอแนะ.....86

**บรรณานุกรม.....87**

**ภาคผนวก.....88**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# สารบัญตาราง


**ตารางที่** **หน้า**

3.1 แสดงค่าเปรียบเทียบคุณสมบัติทางไฟฟ้าของ RS232C, RS-422A, RS-485.....23

3.2 แสดงตารางของ ASCII.....33

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# สารบัญภาพ

**ภาพที่** **หน้า**


2.1 แสดงผังการผลิตรวม....3

2.2 แสดงการทำงานของระบบ Just In Time....4

2.3 แสดงการเปรียบเทียบการทำงานระบบ Just In Time กับระบบอัตโนมัติด้วยสัมผัสมนุษย์....5

2.4 แสดงขนาดของการผลิตหรือสั่งซื้อที่ประหยัดค่าใช้จ่าย (EOQ)....6

2.5 แสดงปริมาณการผลิตต่อครั้งที่ประหยัดค่าใช้จ่าย (EOQ) มีขนาดลดลง เมื่อเวลาเตรียมการผลิต....7

2.6 แสดงผลกระทบของวิธีการผลิตแบบ Just In Time....8

2.7 แสดงความสูญเปล่าอันเกิดมาจากมีเวลาเหลือ....14

2.8 แสดงการขจัดความสูญเปล่าอันเนื่องมาจากเวลาที่เหลืออยู่....15

2.9 แสดงการขจัดความสูญเปล่าที่เกิดมาจากการขนส่ง....16

3.1 แสดงการติดต่อสื่อสารแบบ Host Link....17

3.2 แสดงความสัมพันธ์ของส่วนประกอบหลักในการสื่อสารข้อมูล....18

3.3 แสดงการสื่อสารข้อมูลอนุกรมแบบ Simplex....19

3.4 แสดงการสื่อสารข้อมูลอนุกรมแบบ Half Duplex....19

3.5 แสดงการสื่อสารข้อมูลอนุกรมแบบ Full Duplex....19

3.6 แสดงการติดต่อสื่อสารข้อมูลแบบอนุกรม....20

3.7 แสดงการสื่อสารข้อมูลแบบขนาน....20

3.8 แสดงแสดงลักษณะสมบัติทางไฟฟ้าของการอินเทอร์เฟสแบบ RS232C....21

3.9 แสดงแสดงโครงสร้างของการสื่อสารข้อมูลแบบอนุกรมตามมาตรฐาน RS-422A....22

3.10 แสดงโครงสร้างของการสื่อสารข้อมูลแบบอนุกรมตามมาตรฐาน RS-485....23

3.11 แสดง Port และการเชื่อมต่อ Port ระหว่าง PLC กับคอมพิวเตอร์ส่วนบุคคล....25

3.12 แสดงระบบการติดต่อแบบจุดต่อจุด....26

3.13 แสดงการเชื่อมโยงแบบอินพุทและเอาท์พุทระยะไกล....26

3.14 แสดงการจำลองพื้นที่ของตัวควบคุมหลักขณะทำงานปกติ....27

3.15 แสดงการจำลองพื้นที่ของตัวควบคุมหลักขณะเมื่อมีการทำงานร่วมกับหน่วยอินพุทเอาท์พุทระยะไกล....27

3.16 แสดงการต่อแบบ Serial link....28

3.17 แสดงการต่อแบบ Parallel link....28



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# สารบัญภาพ (ต่อ)

**ภาพที่** **หน้า**


3.18 แสดงการแบ่งพื้นที่สำหรับ PC Link System....29
3.19 แสดงระบบการติดต่อสื่อสารแบบเครือข่ายท้องถิ่น....29
3.20 แสดงแบบมาตรฐานการสื่อสารข้อมูลแบบอะซิงโครนัส....31
3.21 แสดงการส่งสัญญาณควบคุมผ่านพอร์ทอนุกรม RS-232....31
3.22 แสดงแบบของบล็อกคำสั่ง....31
3.23 แสดงการคำนวณ FCS....32
3.24 แสดงโครงสร้าง Local Area Network ของระบบ....34
3.25 แสดงเพียร์ทูเพียร์เน็ตเวิร์ก....34
3.26 แสดงไคลเอ็นต์ – เซิร์ฟเวอร์....35
3.27 แสดงBus Network....36
3.28 แสดงRing Network....37
3.29 แสดงStar Network....37
3.30 แสดงระบบไคลเอนต์ / เซอร์ฟเวอร์....40
3.31 แสดงรูปแบบการให้บริการฐานข้อมูล....42
3.32 แสดงการให้บริการฐานข้อมูลแบบกระจาย....43
3.33 แสดงการให้บริการฐานข้อมูลขนาดใหญ่....44
3.34 แสดงการแปลงฐานข้อมูลต่างๆ เป็นเทกซ์ไฟล์....46
3.35 แสดงการร่วมกันใช้แฟ้มข้อมูลของโปรแกรม....47
3.36 แสดงตัวอย่างการสร้างตาราง....48
4.1 แสดงแบบจำลองกระบวนการผลิต....67
4.2 แสดงสายการผลิตหลัก....67
4.3 แสดงสายการผลิตย่อย....67
4.4 แสดงWarehouse....68
4.5 แสดงชุดสายพานลำเรียง....69
5.1 แสดงแบบจะลองการผลิต....71
5.2 แสดงรูปแบบของโปรแกรม....72
5.3 แสดงวิธีการคลิก Mouse เลือกพอร์ตในการติดต่อ....73



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# สารบัญภาพ (ต่อ)

ภาพที่ หน้า


5.4 แสดงการสั่งให้กล้องเริ่มทำงาน....73
5.5 แสดงเมื่อสั่งการทำงานให้กล้องเริ่มทำงาน....74
5.6 แสดงการถ่ายภาพไปวิเคราะห์สี....74
5.7 แสดงการคลิก Mouse เพื่อดูจำนวนอุปกรณ์ที่จะผลิต....75
5.8 แสดงจำนวนและชนิดของอุปกรณ์ที่จะผลิต....75
5.9 แสดงการคลิก Mouse เพื่อดูว่าขณะนี้กำลังผลิตอุปกรณ์ชนิดใดอยู่....76
5.10 แสดงให้ดูว่ากำลังผลิตอุปกรณ์ชนิดอยู่และจะผลิตอะไรต่อไป....76
5.11 แสดงการคลิก Mouse เพื่อติดต่อสื่อสารระหว่างเครื่องแม่และเครื่องลูก....77
5.12 แสดงโปรแกรมการติดต่อสื่อสาร....77
5.13 แสดงรูปตัวโปรแกรมในส่วนสายการผลิตหลัก....78
5.14 แสดงวิธีการเลือกพอร์ต....79
5.15 แสดงวิธีการ Connect....79
5.16 แสดงการที่จะเข้าไปสั่ง Order....80
5.17 แสดงหน้าจอที่ให้สั่ง Order....80
5.18 แสดงรายงานการผลิต....81
5.19 แสดงการกดปุ่มเริ่มการทำงานสายการผลิตหลัก....81
5.20 แสดงการเลือกการผลิตแบบ Auto....82
5.21 แสดงการเลือกการผลิตแบบ Manual....82
5.22 แสดงรูปที่เกิดการฟ้องให้เราไปปรับ Switch ที่ตัวสายการผลิต....83
5.23 แสดงหน้าต่างการทำงานในโหมด Manual....83
5.24 แสดงวิธีเข้าโปรแกรม....84
5.25 แสดงการใส่ค่า IP....84
5.26 แสดงหน้าจอที่ใส่ค่า IP เสร็จพร้อมพูดคุยได้....85



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# บทที่ 1
# บทนำ

## 1.1 ความเป็นมาและเหตุจูงใจของการวิจัย

กระบวนการผลิตของโรงงานอุตสาหกรรมนั้นทุก ๆ โรงงานต้องการผลกำไร เพื่อให้คุ้มค่ากับการลงทุนจึงจำเป็นต้องมีการออกแบบกระบวนการผลิตที่เหมาะสม เพื่อให้ได้สินค้าที่มีคุณภาพและได้กำไรสูงสุด การใช้ระบบการผลิตแบบ Just In Time เข้ามาใช้ในกระบวนการผลิตนั้นจะช่วยประหยัดค่าใช้จ่ายในการผลิตให้ต่ำที่สุดเมื่อใช้ต้นทุนในการผลิตน้อยจะทำได้ผลกำไรมากยิ่งขึ้น จึงได้ศึกษาและทำ Project เพื่อความเข้าใจในหลักการผลิตแบบ Just In Time อย่างท่องแท้

## 1.2 วัตถุประสงค์ของปริญญานิพนธ์

1. ศึกษาทำความเข้าใจหลักการผลิตซึ่งใช้ระบบ Just In Time
2. เขียนโปรแกรมเพื่อติดต่อสื่อสารผ่านระบบ LAN ได้
3. เขียนโปรแกรมเพื่อสั่งการทำงานระหว่าง Computer กับ Ware House ได้
4. เขียนโปรแกรมเพื่อสั่งการทำงานระหว่าง Computer กับชุดสายพานลำเรียงได้
5. พัฒนาให้ชุดสายพานลำเรียงสามารถขัดแยกวัตถุได้
7. เขียนโปรแกรมติดต่อข้อมูลกับData Base ผ่านระบบ LAN
8. ศึกษาการตรวจจับสีด้วยระบบ Fuzzy

## 1.3 ขอบเขตของปริญญานิพนธ์

1. สามารถสั่งการทำงานชุด Ware House จาก Computer ได้
2. สามารถติดต่อสื่อสารข้อมูล ผ่านระบบ LAN ได้
3. สามารถสั่งการทำงานชุด Ware House ผ่านระบบ LAN ได้
4. สามารถสั่งการทำงานชุดสายพานลำเรียงได้
5. ทำให้ชุดสายพานลำเรียงสามารถขัดแยกวัตถุได้
6. ชุด Ware House สามารถสั่งการทำงานไปยังชุดสายพานลำเรียงแบบอัตโนมัติได้ เมื่อมีการสั่งการทำงานมายังชุด Ware House
7. สามารถติดต่อสื่อสารข้อมูลกับData Base ผ่านระบบ LAN

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 1.4 รายละเอียดของปริญญานิพนธ์

ปริญญานิพนธ์ฉบับนี้ได้จัดทำออกเป็นทั้งหมด 6 บทด้วยกัน โดยแต่ละบทมีรายละเอียดดังนี้

บทที่ 1 เป็นการกล่าวนำวัตถุประสงค์และขอบเขตของปริญญานิพนธ์แบบนี้

บทที่ 2 เป็นการกล่าวถึงทฤษฎีระบบทันเวลาพอดี

บทที่ 3 เป็นการกล่าวถึงทฤษฎีการติดต่อสื่อสารข้อมูล

บทที่ 4 เป็นการกล่าวถึงขั้นตอนและวิธีดำเนินงาน

บทที่ 5 เป็นการกล่าวถึงการทดสอบโครงงาน

บทที่ 6 เป็นการกล่าวถึงสรุปผลการวิจัย และข้อเสนอแนะ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# บทที่ 2
# ทฤษฎีระบบทันเวลาพอดี

## 2.1 กล่าวนำ

ระบบแบบทันเวลาพอดีเป็นการผลิตสินค้าในเวลาที่ต้องการเท่านั้น ด้วยเหตุนี้จึงไม่จำเป็นต้องมีเครื่องคอมพิวเตอร์ขนาดใหญ่มาช่วยควบคุมวัสดุคงคลัง นอกจากนี้การผลิตแบบ Just in time ยังช่วยทำให้คุณภาพสินค้าดีขึ้น เป็นการเพิ่มผลผลิต และที่สำคัญคือทำให้คนงานมีความรับผิดชอบต่องานที่ทำมากขึ้นกว่าเดิม

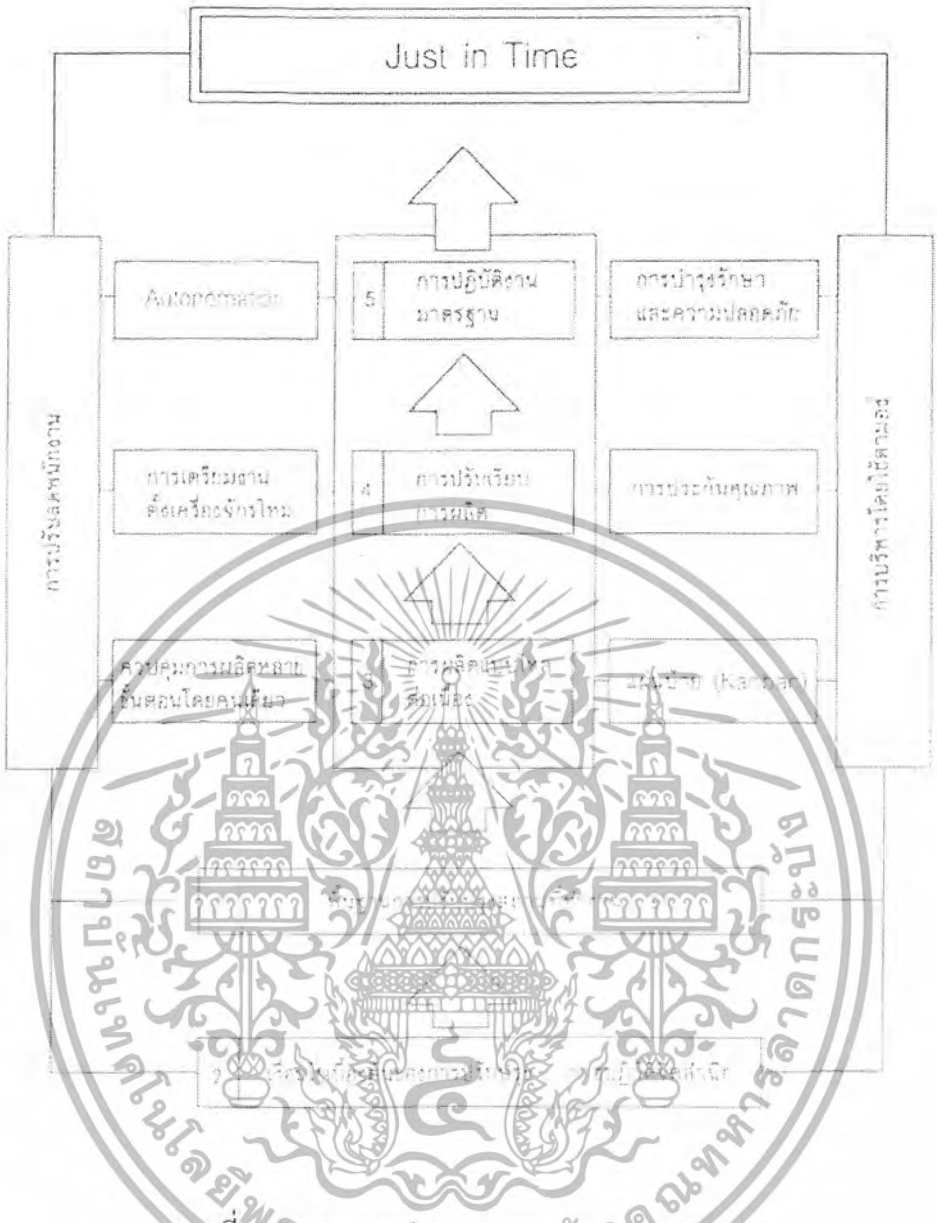
## 2.2 วิธีการผลิตแบบ Just In Time

วิธีการผลิตแบบ "Just In Time" เป็นหัวใจสำคัญของการบริหารการผลิต และเพิ่มผลผลิตของโรงงาน หลักการของ Just In Time คือ โรงงานจะทำการผลิตสินค้าสำเร็จ และจัดส่งออกไปเมื่อมีการขายเกิดขึ้นเท่านั้น และส่วนประกอบต่าง ๆ การขึ้นผลิตชิ้นส่วนต่าง ๆ จะทำในเวลาที่การผลิตส่วนประกอบมีความต้องการชิ้นส่วน วัสดุก็จะถูกสั่งซื้อก็ต่อเมื่อมีความต้องการใช้วัสดุเท่านั้น

หลักการของ Just In Time จะทำให้ค่าใช้จ่ายของการเก็บรักษาวัสดุต่ำที่สุด ไม่ว่าจะเป็นวัตถุดิบ งานระหว่างผลิต และระหว่างผลิตและสินค้าสำเร็จรูป ขนาดของการผลิตที่ประหยัดที่สุดก็มีค่าเข้าใกล้ 1 หน่วย ดังนั้น จึงทำให้โรงงานสามารถผลิตสินค้าตามความต้องการได้อย่างแท้จริง และสามารถผลิตสินค้าได้หลายอย่างในสายการผลิตเดียวกันด้วย Just In Time จะช่วยลดต้นทุนการผลิต โดยเฉพาะอย่างยิ่งค่าเก็บรักษาวัสดุ

**ภาพที่ 2.1** แสดงผังการผลิตรวม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้




ภาพที่ 2.2 แสดงการทำงานของระบบ Just In Time

การผลิตแบบ Just In Time เป็นการบริหารการผลิตที่ใช้ในการควบคุมวัสดุคุณภาพสินค้าและของเสียที่เกิดจากการผลิต เกี่ยวข้องกับการจัดโรงงานเพื่อเพิ่มผลผลิตให้สูงขึ้น ซึ่งต้องอาศัยการมีส่วนร่วมจากคนงานทุกคน และการกระตุ้นจากฝ่ายบริหาร เริ่มต้นตั้งแต่การจัดซื้อ การผลิตในแต่ละแผนก และการจัดส่งสินค้า ในที่นี้จะกล่าวถึงกระบวนการที่เกิดขึ้นเมื่อนำเอาวิธีการ Just In Time มาใช้ในการผลิต ซึ่งแบ่งออกเป็น 3 ขั้นตอนใหญ่ ๆ ดังนี้ คือ

1. การลดขนาดการผลิตที่ประหยัดค่าใช้จ่ายที่สุดของสินค้า แต่ละชนิด
2. ผลกระทบที่เกิดขึ้นเมื่อนำเอาวิธีการ Just In Time มาใช้
3. การผสมผสานระหว่างการควบคุมคุณภาพสมบูรณ์แบบกับการผลิตแบบทันเวลาพอดี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**ภาพที่ 2.3** แสดงการเปรียบเทียบการทำงานระบบ Just In Time กับระบบอัตโนมัติด้วยสัมผัสมนุษย์

## 2.3 การลดขนาดของการผลิต

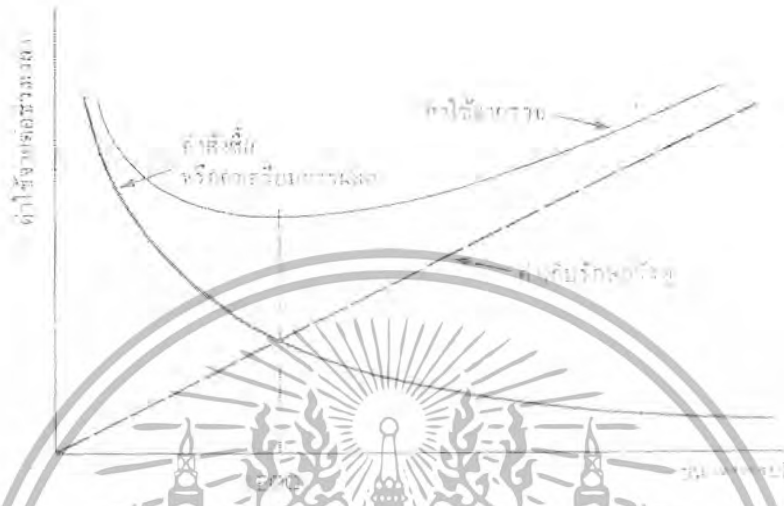
เมื่อผลิตสินค้าหรือสั่งซื้อสินค้าครั้งหนึ่ง ๆ เป็นจำนวนมาก ปริมาณของวัสดุคงคลังก็จะมีจำนวนมากตามไปด้วย ซึ่งทำให้ต้องใช้เงินลงทุนเป็นจำนวนมาก โดยจ่ายไปในรูปของดอกเบี้ยของเงินที่ลงทุนไปในวัสดุคงคลังนั้น รวมทั้งค่ารักษา ดังนั้นถ้าต้องการลดค่าใช้จ่ายตัวนี้ลงมาก็จะต้องสั่งให้ผลิตหรือออกคำสั่งซื้อสินค้าครั้งละน้อย ๆ บ่อย ๆ ครั้ง

อย่างไรก็ตามการสั่งให้ผลิตหรือสั่งซื้อสินค้าครั้งละน้อย ๆ บ่อย ๆ ครั้งมากเกินไป ก็เป็นการเพิ่มค่าใช้จ่ายได้เหมือนกัน ตัวอย่างเช่น ในโรงงานผลิตชิ้นส่วนเครื่องยนต์ ทุกครั้งที่มีคำสั่งให้ผลิตชิ้นส่วน ฝ่ายผลิตที่เกี่ยวข้องก็จำเป็นต้องปรับเครื่องจักรเพื่อผลิตสินค้าให้ได้ตามข้อกำหนดที่ต้องการ ในการนี้จะเกิดค่าใช้จ่ายขึ้นชนิดหนึ่งเรียกว่าค่าเตรียมการผลิต ซึ่งได้แก่ ค่าเสียเวลาเครื่องจักรในการรอคอยการเปลี่ยนแบบสำหรับการขึ้นรูปชิ้นงาน และเมื่อทำการเปลี่ยนแบบเสร็จแล้ว ก็จำเป็นต้องทดลองผลิตชิ้นงานขึ้นดูก่อนอย่างน้อยหนึ่งชิ้น เพื่อตรวจสอบขนาดว่าได้ตามที่กำหนดหรือไม่ ปกติแล้วชิ้นงานแรก ๆ มักจะมีข้อบกพร่องไม่มากก็น้อย ทำให้ต้องเสียเวลาปรับแต่งกันนานซึ่งบางครั้งใช้เวลานานหลายชั่วโมงกว่าจะได้ชิ้นงานที่มีขนาดตามต้องการ

ทางแก้ไขคือ การคำนวณหาขนาดของการผลิตหรือสั่งซื้อที่ประหยัดค่าใช้จ่ายที่สุดคือไม่เสียค่าเตรียมการผลิตและค่าเก็บรักษา ค่าใดค่าหนึ่งมากเกินไป ดังแสดงไว้ในรูปที่ 2 จากรูปจะเห็น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ได้ว่าถ้าขนาดของการผลิตหรือสั่งซื้อครั้งหนึ่ง ๆ มาก ค่าเก็บรักษาก็จะมากขึ้นไปด้วย แต่ค่าเตรียมการผลิตหรือค่าสั่งซื้อกลับมีค่าน้อยลงในทางตรงกันข้ามถ้าขนาดของการผลิตหรือสั่งซื้อครั้งหนึ่ง ๆ น้อย ค่าเก็บรักษาก็จะน้อยลง แต่ค่าเตรียมการผลิตสูงขึ้น



**ภาพที่ 2.4** แสดงขนาดของการผลิตหรือสั่งซื้อที่ประหยัดค่าใช้จ่าย (EOQ)

สูตรสำหรับการคำนวณหาขนาดของการผลิตหรือสั่งซื้อที่ประหยัดค่าใช้จ่ายที่สุด หรือที่รู้จักกันดีว่า EOQ (Economic Order Quantity) อย่างไรก็ตามญี่ปุ่นจะไม่ยอมรับขนาดของการผลิตหรือสั่งซื้อที่ประหยัดค่าใช้จ่ายที่สุดจากการคำนวณตามสูตรของฟอร์ด แฮร์ริส และคาร์ เฮช วินสัน ทั้งนี้เป็นเพราะมีเหตุผล 2 ประการคือ

- ตามสูตรของการคำนวณหาค่า EOQ นั้น มีค่าใช้จ่ายที่เกี่ยวข้องเพียง 2 ตัวเท่านั้น คือ ค่าเตรียมการผลิตหรือค่าสั่งซื้อ และค่าเก็บรักษาวัสดุ เพราะจะมันเล็กเกินไปที่ไม่เป็นการเพียงพอเพราะมีปัจจัยอื่น ๆ อีกที่ควรนำมาเกี่ยวข้องกับการกำหนดค่า EOQ ด้วย เช่น คุณภาพ เศษเล็กเศษน้อยของการผลิต ความรับชอบและพลังการกระตุ้นการทำงานของคนงาน เป็นต้น

- ค่าเตรียมการผลิตมีอยู่จริง และเป็นตัวที่มีความสำคัญมาก ต่อการกำหนดขนาดของการผลิตอย่างไรก็ตามค่านี้ ไม่ใช่ค่าคงที่ที่ไม่สามารถเปลี่ยนแปลงได้คนส่วนใหญ่แล้วมักจะให้ความสำคัญกับค่าเก็บรักษาวัสดุมากกว่าค่าเตรียมการผลิต

จุดสำคัญของการเพิ่มผลผลิตของญี่ปุ่นอยู่ที่การเอาชนะเวลาที่ต้องเสียไป ในการเตรียมการผลิต โดยการรณรงค์กันทั้งโรงงาน ทุกระดับของการผลิต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 2.4 การลดเวลาเตรียมการผลิต

ตัวอย่างของโรงงานอุตสาหกรรมญี่ปุ่น ที่ประสบความสำเร็จเป็นอย่างดี ในการลดเวลาเตรียมการผลิตก็คือบริษัทโตโยต้า บริษัทนี้เริ่มรณรงค์เรื่องการลดเวลาเตรียมการผลิตตั้งแต่ปี พ.ศ. 2514 ในขณะนั้นเครื่องปั๊มแผ่นเหล็กขนาด 800 ตันสำหรับการผลิตหลังคาและยังโคลนรถยนต์ต้องใช้เวลาเตรียมการผลิตต่อครั้งประมาณ 1 ชั่วโมงหลังจากนั้นอีก 5 ปี สามารถลดเวลาเตรียมการผลิตนี้ลงได้เหลือเพียง 12 นาที และต่อมา ได้ตั้งนโยบายที่จะทำให้เวลาเตรียมการผลิตน้อยกว่า 10 นาที จึงทำให้งานบางงานสามารถเตรียมการผลิตได้โดยใช้วิธีกดปุ่มเพียงปุ่มเดียวเท่านั้น ดังนั้นด้วยเวลาเพียงไม่กี่วินาทีก็สามารถผลิตได้แล้ว นับว่าเป็นประโยชน์ต่อการผลิตมากทีเดียวเพราะทำให้โรงงานไม่ต้องผลิตสินค้าสำรองเอาไว้มาก แนวความคิดนี้จึงเป็นที่รู้จักและใช้กันอย่างแพร่หลายในโรงงานอุตสาหกรรม

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

**ภาพที่ 2.5** แสดงปริมาณการผลิตต่อครั้งที่ประหยัดค่าใช้จ่าย (EOQ) มีขนาดลดลงเมื่อเวลาเตรียมการผลิตน้อย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การดัดแปลงเครื่องจักรเพื่อลดเวลาเตรียมการผลิตเป็นไปอย่างกว้างขวาง ไม่เพียงแต่เท่านั้นโรงงานหลายแห่งเริ่มใช้วิธีการสร้างเครื่องจักรขึ้นมาใช้งานเองเพื่อแก้ไขปัญหาเรื่องเวลาเตรียมการผลิต ผลที่ได้ก็คือโรงงานได้เครื่องจักรที่เหมาะสมกับงาน มีน้ำหนักเบา สามารถเคลื่อนย้ายได้ง่าย เสียค่าใช้จ่ายน้อยและที่สำคัญที่สุดก็คือเวลาเตรียมการผลิตลดลงเหลือ 0 ที่เป็นเช่นนี้ได้เพราะเหตุว่าโรงงานใช้เครื่องจักรเฉพาะงาน ข้อแตกต่างระหว่างการผลิตแบบตะวันตก และแบบญี่ปุ่นพิจารณาได้จากภาพที่ 2.5

## 2.5 ผลกระทบของวิธีการผลิตแบบ Just In Time

**ภาพที่ 2.6** แสดงผลกระทบของวิธีการผลิตแบบ Just In Time

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ผลกระทบจากการผลิตแบบ Just In Time แสดงไว้ในรูปที่ 4 จากรูปนี้จะเห็นได้ว่าผลของการลดขนาดของการผลิตลงมาทำให้เกิดการผลิตแบบ Just In Time ซึ่งสนองต่อความต้องการของตลาดได้ดี วัสดุคงคลังในระบบการผลิตลดลงเป็นการลดค่าใช้จ่ายทางอ้อมลงมา การสูญเปล่าของเวลาทำงานคนงาน เครื่องจักรและงานที่นำมาแก้ไขใหม่ลดลง นอกจากนี้ยังเข้ากันได้ดีกับการควบคุมคุณภาพสมบูรณ์แบบอีกด้วย คือสามารถตรวจสอบหาข้อบกพร่องและควบคุมคุณภาพสินค้าได้อย่างรวดเร็วและทั่วถึง ทำให้ส่งข้อมูลย้อนกลับไปที่คนงานหรือผู้ผลิตได้ทันที และเมื่อผู้ผลิตทราบก็จะทำการค้นหาสาเหตุและหาทางแก้ไขปัญหาเพื่อไม่ให้เกิดขึ้นมาอีกโดยเร็วในบาง ครั้ง อาจมีการหยุดสายการผลิตเพื่อแก้ไขปัญหา จนกว่าจะเรียบร้อยแล้วจึงทำการผลิตต่อไป กล่าวได้ว่าการผลิตแบบ Just In Time ทำให้ได้สินค้าที่มีคุณภาพสูงและมีอัตราการผลิตที่สม่ำเสมอ และถ้าโรงงานมีเป้าหมายที่จะไม่มีวัสดุคงคลังสำรองหรือลดจำนวนคนงานลงมา จุดที่เป็นปัญหาก็แสดงออกมาให้เห็นทันทีทำให้รู้ว่าระบบการผลิตของโรงงานมีจุดอ่อนอยู่ที่ใด ซึ่งจะได้ทำการแก้ไขปรับ ปรุงจุดนี้ต่อไป รายละเอียดของผลดีที่เกิดขึ้นเมื่อนำเอาวิธีการผลิตแบบ Just In Time มาใช้ พอ สรุปเป็นข้อ ๆ ดังนี้คือ

**2.5.1 เป็นการยกระดับคุณภาพสินค้าให้สูงขึ้นและลดของเสียจากการผลิตให้น้อยลง**

เหตุผลก็คือเมื่อคนงานผลิตชิ้นส่วนชิ้นหนึ่งเสร็จก็จะส่งต่อไปให้กับคนงานคนต่อไปทัน ที ถ้าหากปรากฏว่าชิ้นส่วนมีข้อบกพร่อง คนงานที่รับชิ้นส่วนนั้นมาก็จะรีบแจ้งให้กับคนงานที่ผลิตรับทราบทันที เพื่อให้หาสาเหตุ และแก้ไขให้ถูกต้อง ดังนั้นจึงทำให้การผลิตเต็มไปด้วยความระมัดระวังข้อบกพร่องต่าง ๆจึงน้อยลง คุณภาพสินค้าจึงดีขึ้น ซึ่งต่างจากการผลิตครั้งละมาก ๆ คนงานที่รับชิ้นส่วนมาผลิตต่อนั้นไม่สนใจว่าชิ้นส่วนที่รับมานั้นมีข้อบกพร่องอยู่มากหรือไม่ แต่จะรีบทำการผลิตต่อทันทีเพราะเขามีชิ้นส่วนที่จะต้องผลิตรออยู่เป็นร้อยเป็นพันชิ้นอยู่ในกระบะข้างหน้าเขา ข้อบกพร่องต่าง ๆไม่ได้รับการแก้ไข คุณภาพสินค้าจึงไม่ดีเท่าอับการผลิตแบบ Just In Time

**2.5.2 ช่วยกระตุ้นการทำงานของคนงาน**

จากหลักการของนักจิตวิทยาท่านหนึ่งคือ บี ดับบลิว สกินเนอร์ ซึ่งกล่าวว่า "คนงานจะเกิดแรงกระตุ้นตามธรรมชาติที่จะพัฒนาตนเองให้ดีขึ้น ถ้าหากเขาได้รับรู้ทันทีว่าผลงานของเขาเป็นอย่างไร" หลักการนี้ใช้ได้ดีในโรงงานที่มีการผลิตแบบ Just In Time เพราะคนงานจะเกิดการตื่นตัวและจะมีความคิดริเริ่มใหม่ ๆ ที่จะปรับปรุงงานที่ตนเองรับผิดชอบอยู่ให้ดียิ่งขึ้นโดยไม่ต้องมีผู้ใดมาสั่งให้ทำ เช่น เมื่อทราบว่าชิ้นงานที่ตนเองทำและส่งไปให้ผู้อื่นมีข้อบกพร่องเกิดขึ้น คนงานที่ทำชิ้นงานนั้นจะเกิดความรู้สึกว่าตัวเองบกพร่องและจะระมัดระวังไม่ให้เกิดขึ้นอีก การผลิตครั้งละมาก ๆ ปัญหาต่าง ๆจะถูกปิด ส่วนการผลิตแบบ Just In Timeนั้น จะผลิตทีละน้อย ๆ การผลิตแบบ Just

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

In Time จะทำให้คนงานทุกคนมีความตระหนักในการผลิตและมีคุณภาพสูง และถ้ามีกิจกรรมกลุ่มคุณภาพร่วมด้วยแล้วกลุ่มจะช่วยกันระดมสมอง เสนอความคิดเห็นที่จะปรับปรุงงานให้ดีขึ้นดังนี้

- ความคิดเห็นสำหรับการควบคุมข้อบกพร่องซึ่งจะส่งผลทำให้คุณภาพดีขึ้น ของเสียลดลง
- ความคิดเห็นในการปรับปรุงการจัดส่งวัสดุเพื่อลดความล่าช้าทำให้การผลิตสินค้าสะดวกและรวดเร็ว
- ความคิดเห็นที่จะลดเวลาเตรียมการผลิต ซึ่งจะส่งผลทำให้ขนาดของการผลิตลดลง

### 2.5.3 ผลกระทบต่อความรับผิดชอบ

ในสายการผลิตแบบ Just In Time คนงานจะมีความรับผิดชอบต่องานของตนเองและงานของส่วนรวมสูงมาก กล่าวคือความรับผิดชอบต่อตนเองก็คือจะต้องผลิตสินค้าที่ดี มีคุณภาพสูง ส่งต่อไปให้กับคนงานตนต่อไปโดยถือเหมือนว่าเป็นลูกค้า ด้านความรับผิดชอบต่อส่วนรวมนั้นก็คือคนงานทุกคนจะต้องช่วยกันแก้ปัญหาเมื่อมีปัญหาเกิดขึ้นในการผลิตที่ผู้หนึ่งผู้ใด เพื่อไม่ให้การผลิตหยุดชะงักเป็นเวลานาน เช่น เมื่อสายพานเครื่องจักรขาด คนงานที่อยู่ใกล้เคียงจะเข้ามาช่วยกันเปลี่ยนทันที โดยไม่ต้องออกปากขอร้อง หรือเมื่อมีงานคั่งค้างอยู่ที่คนใดคนหนึ่งมาก คนงานที่มีเวลาว่างพอก็จะเดินเข้ามาช่วยกันทันทีเพื่อให้งานคั่งค้างในสายการผลิตนั้นน้อยลง ฯลฯ เรื่องการช่วยกันทำงานนี้เป็นเรื่องธรรมดาของญี่ปุ่น แต่เป็นเรื่องยากสำหรับตะวันตกเพราะมักจะทำงานตามหน้าที่ที่กำหนดไว้ในใบกำหนดงานอย่างเคร่งครัด นอกจากนี้ยังมีกฎเกณฑ์จากสหภาพแรงงานหรือสมาคมวิชาชีพไม่ให้คนงานไปทำงานอื่นอีกด้วย

### 2.5.4 เกิดกิจกรรมกลุ่มย่อยเพื่อการปรับปรุงงาน

หรือรู้จักกันดีว่ากลุ่มคุณภาพ หรือ QCC กิจกรรมนี้ประสบความสำเร็จเป็นอย่างดีในประเทศญี่ปุ่นและแพร่หลายออกไปทั่วโลก ปรัชญาของกลุ่มคุณภาพก็คือการอาศัยคนงานที่อยู่ในหน่วยงานเดียวกัน ที่อาสาสมัครกันเข้ามาช่วยกันพัฒนาคุณภาพงานในหน่วยงานของตนอย่างต่อเนื่อง โดยมีเป้าหมายที่แน่นอน จากที่ได้กล่าวมาแล้วว่าการผลิตแบบ Just In Time ทำให้คนงานมีความรับผิดชอบต่อตนเองและส่วนรวมดีขึ้นนั้น สาเหตุเป็นเพราะว่าเมื่อขนาดการผลิตเล็กลง การทำงานของแต่ละคนก็จะมีความสัมพันธ์กันอย่างใกล้ชิดขึ้น ผลผลิตของคนหนึ่งเป็นสินค้าของอีกคนหนึ่ง ต่อเนื่องกันไปทั้งโรงงาน ดังนั้นเมื่อมีปัญหาเกิดขึ้นที่จุดใดจุดหนึ่งหัวหน้างานและคนงานที่เกี่ยวข้องอีกคนหนึ่ง ต่อเนื่องกันไปทั้งโรงงาน ดังนั้นเมื่อมีปัญหาเกิดขึ้นที่จุดใดจุดหนึ่ง หัวหน้าคนงานและคนงานที่เกี่ยวข้องก็จะเข้ามาช่วยกันแก้ไขปัญหานั้นทันที โดยถือว่าเป็นกิจกรรมของกลุ่มคุณภาพซึ่งมีการกำหนดปัญหาการระดมสมอง การเก็บข้อมูล การกำหนดรูปแบบข้อมูล การใช้กราฟและแผนภูมิควบคุม การวิเคราะห์การตัดสินใจด้วยไดอะแกรมพาเรโต การวิเคราะห์เหตุและผล และวิธีเสนอผลงานต่อผู้บริหารโครงการของกลุ่มคุณภาพในโรงงานมักจะเกี่ยวข้องกับการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ป้องกันและแก้ไขปัญหาจากกระบวนการผลิต เช่น ข้อบกพร่อง การคับคั่งของงานที่จุดใดจุดหนึ่ง การทำงานช้ากว่าปกติ เครื่องจักรเสียบ่อย ๆ ฯลฯ กิจกรรมของกลุ่มคุณภาพจะมีความต่อเนื่องและสอดคล้องกันทั้งโรงงานโดยมีคณะกรรมการอำนวยการกำหนดนโยบาย แบะมีผู้แระสานงานติดตามผลงานของแต่ละกลุ่มอย่างใกล้ชิด เมื่อกลุ่มคุณภาพแก้ไขปัญหาหนึ่งได้แล้วก็จะคัดเลือกปัญหาที่มีความสำคัญต่อหน่วยงานของตนมาทำต่อไปจากปัญหาหนึ่งไปสู่อีกปัญหาหนึ่ง ทำให้คนงานได้ใช้ความรู้ความสามารถอย่างเต็มที่ เป็นการทำงานร่วมกัน และยังเป็นการใช้เวลาว่างของคนงานให้เกิดประโยชน์อีกด้วย

### 2.5.5 การไม่มีวัสดุคงคลังสำรอง

ในระบบการผลิตแบบ Just In Time ถือว่าการมีวัสดุคงคลังสำรองนั้นเป็นการสูญเปล่าที่จำเป็นต้องขจัดออกไปให้หมด เหตุผลที่จำเป็นต้องมีวัสดุคงคลังสำรองเกิดจากความไม่แน่นอน ไม่สม่ำเสมอที่เกิดขึ้นระหว่างกระบวนการผลิต ยิ่งมีความไม่สม่ำเสมอมากเท่าไหร่ก็ยิ่งต้องมีวัสดุคงคลังสำรองมากขึ้นเท่านั้นโดยการกำหนดปริมาณของวัสดุคงคลังสำรองที่เหมาะสมนั้นต้องอาศัยสูตรทางสถิติและวิธีการทางคณิตศาสตร์คำนวณหาออกมา ที่กล่าวมานี้เป็นหลักการบริหารวัสดุคงคลังที่รู้จักกันดี อย่างไรก็ตามญี่ปุ่นไม่ยอมรับเรื่องนี้เพราะถือว่าเป็นการปิดบังปัญหาเอาไว้ และไม่ได้แก้ปัญหาที่ต้นเหตุที่ทำให้เกิดความไม่สม่ำเสมอนั้น หลักการของ Just In Time จะให้คนงานเผชิญกับปัญหานั้นโดยตรงและมีนโยบายที่จะขจัดวัสดุคงคลังสำรองออกไปจากกระบวนการผลิตให้หมดโดยให้คนงานช่วยกันแก้ไขปัญหาความไม่สม่ำเสมอที่เกิดขึ้นนั้น เช่น เครื่องจักรเกิดขัดข้องบ่อย ๆ หรือเครื่องมือจับยึดชิ้นงานไม่ค่อยดี ต้องปรับต้องแต่งกันบ่อย ๆ และเมื่อคนงานสามารถแก้ไขปัญหาความไม่สม่ำเสมอได้แล้ว ผู้จัดการก็จะกล้าที่จะลดวัสดุคงคลังสำรองที่มีอยู่ออกไปอีก บริษัทโตโยต้าถือว่าสายการผลิตที่มีการหยุดงานบ่อย ๆ เป็นสายงานที่ใช้ไม่ดีเช่นกัน เพราะอาจมีคนงานหรือมีการสำรองวัสดุคงคลังไว้มากเกินไป โตโยต้ากำหนดไว้ว่าการหยุดสายการผลิตอย่างมีเหตุผลควรใช้เวลาอยู่ในช่วง 20 - 30 นาที ต่อ 1 กะ (เวลาที่สายการผลิตหยุดจะมีนาฬิกาจับเวลาโดยอัตโนมัติ โดยจะบอกเวลาหยุดสะสมเพื่อให้หัวหน้างานและคนงานมองเห็นได้ชัดเจน) จากรูปที่ 4 จะเห็นได้ว่าเมื่อการผลิตมีอัตราที่สม่ำเสมอ ฝ่ายบริหารจะให้ลดวัสดุคงคลังสำรองหรือคนงานที่มีอยู่ เพื่อจะให้จุดอ่อนหรือปัญหาของสายการผลิตที่แฝงปรากฏขึ้น ซึ่งการแก้ไขเรื่องนี้จะเป็นหน้าที่ของกลุ่มคุณภาพ ที่จะหาทางขจัดปัญหานั้นออกไป

### 2.5.6 การลดแรงงานทางอ้อม

เมื่อวัสดุคงคลังในระบบการผลิตลดลงอันเนื่องมาจากการผลิตแบบ Just In Time ผลกระทบที่จะเกิดขึ้นอีกได้แก่คนงานและผลผลิต กล่าวคือเมื่อวัสดุคงคลังน้อยลง ค่าดอกเบี้ยของเงินทุนที่จมอยู่ก็จะน้อยลงไปด้วย ห้องเก็บของก็ไม่ต้องมีขนาดใหญ่ และจะทำให้โรงงานดู

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กว้างขวางและมีพื้นที่ใช้สอยมากขึ้น เพราะงานระหว่างผลิตน้อยลงหรือไม่มีคั่งค้างอยู่เลย พนักงานประจำห้องเก็บของลดลง งานบัญชีวัสดุลดลง การควบคุมดูแลรักษาวัสดุคงคลังลดลง เพราะฉะนั้นแรงงานทางอ้อมจึงสามารถลดลงได้มากในประเทศญี่ปุ่นโรงงานส่วนใหญ่นิยมให้ผู้ผลิตส่งวัสดุมาให้มีสายการผลิตที่ต้องใช้วัสดุนั้นโดยตรงอีกด้วย ดังนั้นการผลิตจึงไม่จำเป็นต้องมีวัสดุคงคลังสำรองและการขนถ่ายวัสดุภายมนโรงงานก็ลดลงด้วย ในระบบการผลิตของโตโยต้า การขจัดปัญหา และการลดจำนวนคนงานจะกำเนินไปเรื่อย ๆ โดยไม่มีการหยุดยั้ง ซึ่งในบางกรณีจำเป็นต้องลดคนงานลงเพื่อให้เกิดปัญหาและจะเข้าไปแก้ปัญหาที่จุดนั้น ผู้จัดการโรงงานได้สรุปขั้นตอนของวิธีการปรับปรุงสายการผลิตเอาไว้ดังนี้

- เอาคนงานบางคนจากสายการผลิต
- ดูว่ามีปัญหาเกิดขึ้นหรือไม่ ถ้ามี เกิดที่ใด
- ทำการวิจัยปัญหาที่เกิดขึ้น เพื่อหาทางขจัดปัญหานั้น
- เมื่อวิจัยและขจัดปัญหาได้แล้วจึงกำหนดคนงานลงไปในสายการผลิต
- ให้สายการผลิตนั้นดำเนินงานไปสักระยะเวลาหนึ่ง (2 – 3 อาทิตย์ หรือ 2 – 3 เดือน๗)
- เมื่อพิจารณาแล้วเห็นว่างานดำเนินไปได้เรียบร้อยดีและปราศจากปัญหาใด ๆ เราจะต้องคิดว่ามีส่วนเกินในสายการผลิตนั้นแล้ว ดังนั้นจึงต้องทำการปรับปรุงสายการผลิตนั้น

**2.5.7 การเพิ่มผลผลิตและตอบสนองความต้องการของตลาด**

เป็นที่ทราบกันดีอยู่แล้วว่าคนงานเป็นทรัพยากรที่มีค่าที่สุดของระบบการผลิต ดังนั้นคนงานจะต้องได้รับการจูงใจให้ร่วมกิจกรรมในการปรับปรุงงานอย่างสม่ำเสมอ ทั้งนี้เพราะเป็นสิ่งที่เกี่ยวข้องกับการทำงานของตัวเขาเองของส่วนรวมและสิ่งแวดล้อมอื่น ๆ การปรับปรุงงานโดยมากจะเกี่ยงข้องกับงานดังต่อไปนี้คือ การลดต้นทุนการผลิต การลดเวลาเตรียมการผลิต และเวลาผลิตต่อชิ้น การลดการสูญเสีย การปรับปรุงอุปกรณ์เครื่องมือเครื่องใช้ให้สะดวกและเหมาะสมกับงาน การปรับปรุงสถานที่ทำงาน การปรับปรุงด้านความปลอดภัย และสิ่งแวดล้อมต่าง ๆ ให้ดีขึ้น คนงานเป็นองค์ประกอบที่สำคัญของกิจกรรมกลุ่มคุณภาพ ซึ่งมีการประชุมร่วมกันอย่างสม่ำเสมอ เพื่อที่จะปรับปรุงงาน โดยช่วยกันเสนอความคิด ให้ความรู้ และฝึกอบรมแก่คนงานในกลุ่ม ฝ่ายบริหารจะต้องมีความกระตือรือร้นเกี่ยงกับข้อเสนอแนะที่ได้มาจากกลุ่มคุณภาพและเมื่อเห็นด้วยแล้วก็ต้องให้คนงานดำเนินงานตามความคิดเห็นนั้นทันที จากรูปที่ 4 ความคิดเห็นจากคนงาน และกลุ่มคุณภาพเมื่อนำมาปฏิบัติอย่างจริงจังและต่อเนื่องจะส่งผลทำให้ต้นทุนการผลิตลดลง นอกจากนี้ยังทำให้ฝ่ายผลิตตอบสนองความต้องการของฝ่ายขายหรือตลาดได้เร็ว ไม่ต้องพยากรณ์ล่วงหน้าเป็นเวลานาน ๆ และผลิตสำรองไว้เป็นจำนวนมาก เพราะการผลิตมีความคล่องตัวสูง การเตรียมการผลิตใช้เวลาไม่ถึงนาที และสายการผลิตก็สามารถผลิตสินค้าได้หลายอย่างในเวลาเดียวกัน ซึ่งทำให้สินค้าสำเร็จรูปคงคลังเหลืออยู่ในโรงงานน้อยมาก เพราะการผลิต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แบบนี้เป็นไปตามความต้องการของตลาดอย่างแท้จริง ลูกค้าไม่ต้องเสียเวลารอคอย การพยากรณ์การผลิตแม้นยำขึ้นเพราะเป็นการพยากรณ์ระยะสั้น ผู้บริหารไม่ต้องเสียเวลาในการแก้ไขปัญหาในโรงงานต่างๆ ทำให้มีเวลาสำหรับการกำหนดนโยบาย และวางแผนการตลาด และเรื่องอื่น ๆ ได้มาขึ้น

## 2.6 การลดความสูญเปล่าของระบบ Just In Time

ระบบการผลิตแบบ Just In Time มีแนวความคิดที่จะขจัดความสูญเปล่าให้หมดไปเรากล่าวไว้ว่า "กำไรของผู้ผลิตนั้น จะพบได้จากวิธีการผลิต" คำกล่าวนี้ จะสะท้อนให้เห็นถึงปรัชญาในการลดต้นทุน โดยการขจัดกระบวนการที่มีความสูญเปล่า มีความสูญเปล่าอยู่หลายประเภทในการที่จะลดชั่วโมงการทำงานลงนั้นเราจำแนกประเภทของความสูญเปล่าไว้ดังต่อไปนี้

1. ความสูญเปล่าอันมีสาเหตุมาจากการผลิตมากเกินความจำเป็น
2. ความสูญเปล่าอันมีสาเหตุมาจากเวลาการรอคอย
3. ความสูญเปล่าอันมีสาเหตุมาจากการขนส่ง
4. ความสูญเปล่าที่เกิดจากกระบวนการผลิต
5. ความสูญเปล่าที่เกิดจากมีสินค้าคงคลังที่ไม่จำเป็น
6. ความสูญเปล่าที่มีสาเหตุมาจากการเคลื่อนไหวที่ไม่จำเป็น
7. ความสูญเปล่าที่เกิดมาจากการผลิตของเสีย

ภาพที่ปรากฏอยู่ทั่วๆ ไปในสถานที่ทำงานหลายๆ แห่ง คือ งานที่มีความก้าวหน้ามากเกินไป โดยปกติแล้วจะต้องมีการรอคอยอยู่บ้าง แต่เป็นเพราะว่าคนงานนั้นทำงานต่อไป ดังนั้น เวลาที่ควรจะเป็นช่วงของการรอคอย ก็จะถูกซ่อนไว้ทำให้ไม่สามารถมองเห็นได้เมื่อกรรมวิธีการผลิตนั้นมีการกระทำแบบซ้ำๆ กัน วัตถุดิบหรือชิ้นส่วนก็จะสะสมกันอยู่ในช่วงเวลาหรือที่จุดท้ายของสายการ ผลิตทำให้เกิดการสต๊อกของชิ้นส่วนโดยไม่จำเป็นการขนย้ายชิ้นส่วน หรือความจำเป็นที่จะต้องหาที่จัดเก็บก็เกิดขึ้นซึ่งเป็นงานที่ไม่จำเป็น ก็เป็นเรื่องยากขึ้นไปอีก ที่จะทำให้เราค้นหาความสูญเปล่าว่าเกิดขึ้นที่ใด

ภายใต้ระบบการผลิตแบบ Just In Time เราเรียกปรากฏการณ์อย่างนี้ว่า "ความสูญเปล่าอันเนื่องมาจากการผลิตมากเกินความจำเป็น" และปรากฏการณ์อย่างนี้นับได้ว่าเป็นสิ่งที่เลวร้ายมากที่สุดในบรรดาความสูญเปล่าทั้งหมดที่กล่าวมาข้างต้น

ความสูญเปล่าอันเนื่องมาจากการผลิตมากเกินความจำเป็นนั้น เป็นสิ่งที่แตกต่างกับความสูญเปล่าประเภทอื่น เพราะว่าจะสะท้อนให้เห็นถึงความสูญเปล่าทั้งหมด ความสูญเปล่าประเภทอื่นนั้นจะทิ้งร่องรอยไว้ให้เห็นเพื่อการแก้ไข แต่ความสูญเปล่าประเภทนี้ เปรียบเสมือนผ้าห่มที่คลุมทั้งหมดไว้ และป้องกันไม่ให้เราทำการแก้ไขหรือปรับปรุง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ดังนั้น ขั้นตอนแรกของการลดชั่วโมงการทำงานก็คือ การขจัดความสูญเปล่าประเภทนี้เสียก่อน ในการที่จะทำเช่นนั้นจะต้องมีการจัดเปลี่ยนสายการผลิตใหม่ มีการตั้งกฎเกณฑ์เพื่อป้องกันไม่ให้เกิดการผลิตมากไป และจะต้องมีการควบคุมไม่ให้เกิดเหตุการณ์อย่างนั้น ต้องมีข้อกำหนดต่าง ๆ ของเครื่องจักรในที่ทำงานนั้น ๆ

เมื่อเราได้ปฏิบัติตามขั้นตอนเหล่านั้นแล้ว การเคลื่อนย้ายสิ่งของต่างๆ ก็จะกลับเข้าสู่สภาพปกติสายการผลิตนั้นก็จะผลิตชิ้นส่วนตามปริมาณ และเวลาที่ต้องการถ้ามีความสูญเปล่าเกิดขึ้นก็จะพบได้ทันทีเพราะจะเกิดขึ้นในเวลานั้น เมื่อมีการจัดสายการผลิตโดยวิธีนี้แล้ว จะทำให้เป็นการง่ายที่จะร่วมมือกันทำกิจกรรมเพื่อ ขจัดความสูญเปล่า เปลี่ยนแปลงงาน ลดจำนวนคนงาน

ความสูญเปล่าอันมีสาเหตุมาจากการเกิดการรอคอยจะเกิดขึ้นเมื่อคนงานต้องยืนรออยู่เฉยๆ ในบริเวณใกล้เครื่องจักรอัตโนมัติ ซึ่งเปรียบเสมือนคนเฝ้า หรือเมื่อเขาไม่สามารถที่จะทำประโยชน์อะไรด้วยมือได้ เพราะว่าเครื่องจักรกำลังทำงานอยู่

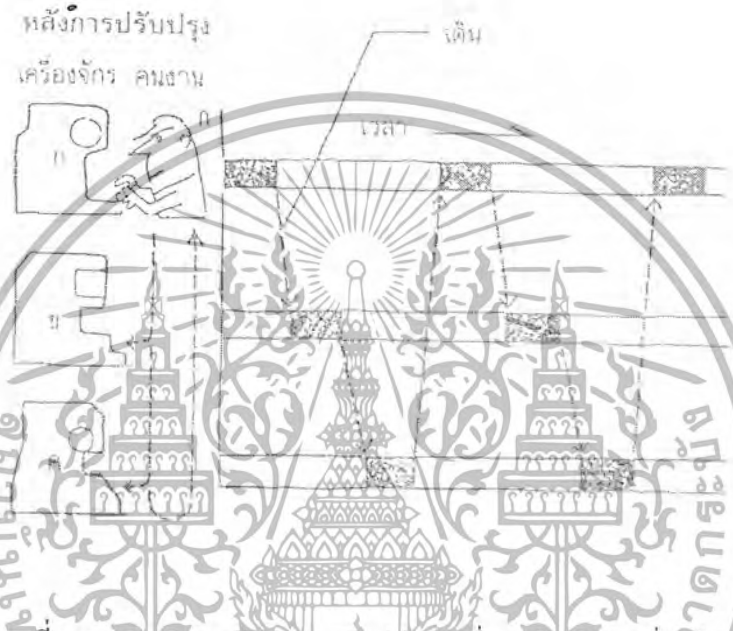
ความสูญเปล่าประเภทนี้จะเกิดขึ้นเมื่อกระบวนการผลิตขั้นก่อนหน้าไม่อาจที่จะจัดส่งชิ้นส่วนที่ต้องการมาให้ ทำให้คนต่อไปไม่มีงานทำ

จากภาพข้างล่างนี้ คนงานได้รับคำสั่งให้ทำงานประจำเครื่องจักร ก ข และ ค ในขั้นตอนนี้คนงานจะยืนอยู่เฉยๆ ในขณะที่เครื่องจักรทำงานอยู่ เขาไม่อาจที่จะทำอะไรได้ถึงแม้ว่าเขาต้องการที่จะทำ จึงทำให้เกิดความสูญเปล่าเนื่องจากการรอคอยขึ้น

**ภาพที่ 2.7** แสดงความสูญเปล่าอันเกิดมาจากมีเวลาเหลือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เพื่อที่จะขจัดความสูญเปล่าประเภทนี้ คนงาน ก จะต้องทำงานประจำเครื่องจักรทั้ง 3 เครื่อง เพื่อทำการป้อนชิ้นงานเข้าเครื่องอย่างต่อเนื่อง การจัดลำดับการทำงานแบบนี้ คนงานจะต้องนำวัตถุดิบในใสเข้าเครื่อง ก แล้งกดปุ่มให้เครื่องจักรทำงานหลังจากนั้นจึงเดินไปที่เครื่องจักร ข นำชิ้นส่วนใส่เครื่องแล้วเดินเครื่อง แล้วจึงเดินไปที่เครื่องจักร ค และทำวิธีการเดิมเหมือนกับเครื่อง จักรทั้งสองเครื่อง หลังจากที่ได้เดินเครื่องจักร ค แล้ว คนงานนั้นก็จะกลับไปที่เครื่องจักร ก งานที่เครื่องจักร ก ก็จะเสร็จพอดี และก็พร้อมที่จะเริ่มงานต่อไปได้ คนงานก็จะสามารถเริ่มงานในรอบต่อไปได้




ภาพที่ 2.8 แสดงการขจัดความสูญเปล่าอันเนื่องมาจากเวลาที่เหลืออยู่

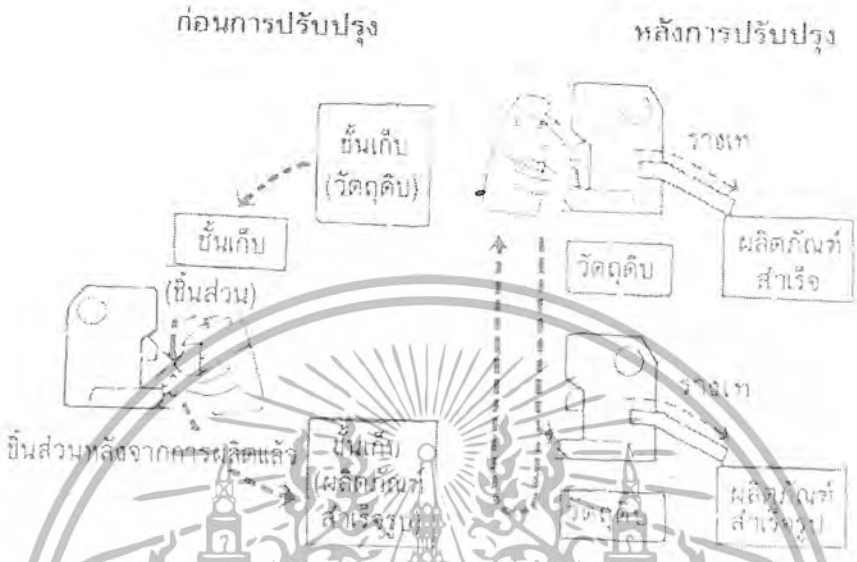
การขจัดความสูญเปล่าอันเนื่องมาจากมีเวลาเหลืออยู่นี้ เราสามารถที่จะนำคนงานออกจากสายการผลิตนั้นได้ถึงสองคน ด้วยวิธีการกล้ายๆ กันนี้ เราสามารจะพิจารณาขจัดความเคลื่อนไหวที่ไม่จำเป็นออกได้อีกด้วย การเคลื่อนไหวดังกล่าวนั้นไม่มีส่วนช่วยในการทำงานเลย

ความสูญเปล่าจากการขนส่ง หมายถึงความสูญเปล่าซึ่ง จะเกิดขึ้นเนื่องจากของชิ้นหนึ่งถูกเคลื่อนย้ายไปโดยไม่มีความจำเป็น ถูกนำไปจัดเก็บไว้ชั่วคราว หรือถูกจัดเรียงใหม่ตัวอย่างเช่นตามที่เคยปฏิบัติกันมานั้น มีการเคลื่อนย้ายชิ้นส่วนจากชั้นเก็บของอันใหญ่ไปยังชั้นเก็บของอันเล็ก แล้วจึงนำไปวางไว้ที่เครื่องจักรชั่วคราวซึ่งจะเห็นได้ว่ามีการขนย้ายเกิดขึ้นหลาย ๆ ครั้งก่อนที่จะนำไปผ่านกระบวนการผลิตจริงๆ การปรับปรุงชั้นเก็บของนั้นจะช่วยให้สามารถขจัดการจัดหาที่วางชั่วคราวออกไปได้ พร้อมทั้งสามารถให้คนงานคนเดียวสามารถเดินเครื่องจักรพร้อมกันสองเครื่องได้

อีกตัวอย่างหนึ่งของความสูญเปล่าอันเนื่องมาจากการขนส่งเกิดขึ้นเมื่อมีการเคลื่อนย้ายชิ้นส่วนจากโกดังไปยังโรงงาน จากโรงงานไปยังเครื่องจักร และจากเครื่องจักรไปยังมือของคนงาน ในแต่ละขั้นตอนเหล่านี้ จะต้องมีการจัดเปลี่ยน และการเคลื่อนย้ายชิ้นส่วนอยู่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ความสูญเปล่าที่เกิดกับกระบวนการผลิตเองเกิดขึ้นเมื่อสิ่งที่บอกตำแหน่งบนแม่แบบจับชิ้นงาน (JIG) มีความคลาดเคลื่อน คนงานจะต้องใช้มือช้ายช่วยจับชิ้นงานไว้ จึงทำให้กระบวนการผลิตเป็นไปอย่างไม่ค่อยจะราบรื่นเท่าที่ควร และทำให้ต้องสูญเสียเวลาไป




ภาพที่ 2.9 แสดงการขจัดความสูญเปล่าที่เกิดมาจากการขนส่ง

นอกจากนี้ ยังมีความสูญเปล่าอันที่เกิดมาจากการที่มีสินค้าคลังมากเกินความจำเป็นจากการเคลื่อนไหวที่ไม่จำเป็น และจากการผลิตของเสีย

## 2.7 การขจัดความสูญเปล่าให้หมดไป

มีหัวหน้างานและผู้จัดการมากมาย ที่ยอมให้คนงาน หรือลูกน้องของตนทำงานที่เขาสงสัยว่าอาจจะเป็นความสูญเปล่าหลายคนก็คิดว่าการกระทำดังกล่าวเป็นส่วนที่จำเป็นสำหรับงานเขาและบ่อยครั้งที่เขาเหล่านั้นไม่เข้าใจถึงพื้นฐานของความสูญเปล่าเหล่านั้น

ไม่ว่าใครก็ตามที่มีความตั้งใจจริงในการขจัดความสูญเปล่ามากเพียงใดถ้าหากเขาขาดความเข้าใจว่าความสูญเปล่านั้นจะประกอบไปด้วยอะไรบ้าง เขาก็จะไม่มีทางที่จะขจัดมันได้ ดังนั้นงานสำคัญที่รอเราอยู่ก็คือการทำให้แน่ใจเสียก่อนว่ามีความสูญเปล่าให้เห็นอย่างเด่นชัดว่าเป็นความสูญเปล่าสำหรับทุกคน และนี่ก็คือขั้นตอนแรกที่จะนำมาซึ่งการปรับปรุงประสิทธิภาพ

ในความสูญเปล่าหลายประเภทนั้น บางอย่างก็มองเห็นได้โดยง่าย แต่บางอย่างก็ยากที่จะมองเห็น ในประเภทต่างๆ เหล่านั้น ประเภทที่มองเห็นได้โดยง่ายก็คือ ความสูญเปล่าอันเกิดมาจากการรอคอย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

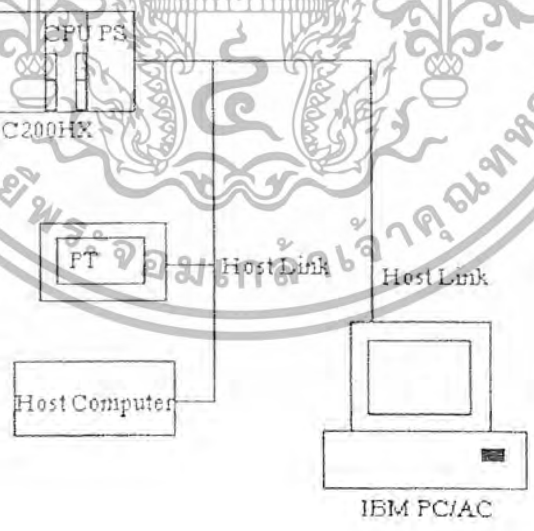
# บทที่ 3
# ทฤษฎีการติดต่อสื่อสารและฐานข้อมูล

## 3.1 กล่าวนำ

ในบทนี้เราจะกล่าวถึงทฤษฎีที่เกี่ยวข้องกับการส่งข้อมูล และการติดต่อสื่อสารกันระหว่าง COMPUTER กับ COMPUTER โดยการติดต่อสื่อสารและส่งข้อมูลนั้นจะใช้สาย UTP ต่อกับการ์ด LAN เป็นตัวกลางในการส่งผ่านและการสื่อสารระหว่าง COMPUTER กับ PLC จะใช้พอร์ตอนุกรม RS 232 เป็นตัวกลางในการส่งข้อมูล

## 3.2 การติดต่อสื่อสารและส่งผ่านข้อมูล

เป็นการถ่ายโอนข้อมูลระหว่าง PLC กับคอมพิวเตอร์ เพื่อเป็นการดูสถานการณ์ทำงานของ PLC ผ่านหน้าจอคอมพิวเตอร์และยังมีไว้เก็บโปรแกรมที่เราสร้างขึ้นมาด้วย ส่วนมากจะนิยมใช้มาตรฐานการส่งอบบอนุกรม เพื่อให้สามารถควบคุม PLC จากคอมพิวเตอร์ได้ สำหรับคอมพิวเตอร์ 1 เครื่องสามารถต่อเข้ากับ PLC ได้จำนวนมาก โดยใช้คำสั่งจาก Host Link




**ภาพที่ 3.1** แสดงการติดต่อสื่อสารแบบ Host Link

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
62131

## 3.3 การติดต่อสื่อสารข้อมูลทั่วไป

ส่วนประกอบเบื้องต้นในการสื่อสารข้อมูลแบ่งได้ออกเป็น 3 ส่วน คือ

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

ซึ่งในแต่ละส่วนมีความสัมพันธ์กันดังรูป

ภาพที่ 3.2 แสดงความสัมพันธ์ของส่วนประกอบหลักในการสื่อสารข้อมูล

การจำแนกรูปแบบในการสื่อสารข้อมูลอาจจัดแบ่งได้หลายแบบ เช่น แบ่งตามชนิดของสัญญาณของข้อมูล คือ การสื่อสารข้อมูล อนาล็อก (Analog) และ การสื่อสารข้อมูลดิจิตอล (Digital) หรือแบ่งตามวิธีการส่งข้อมูลในบทนี้จะกล่าวถึงการสื่อสารข้อมูลดิจิตอลซึ่งเป็นข้อมูลชนิดที่เกี่ยวข้องกับเครื่องคอมพิวเตอร์และเครื่องควบคุม PLC/PC

## 3.4 รูปแบบการสื่อสารข้อมูล

มีรูปแบบการสื่อสารข้อมูลอยู่ 2 ประเภท คือ

1. แบบอนุกรม
2. แบบขนาน

### 3.4.1 แบบอนุกรม

การส่งข้อมูลแบบอนุกรมนั้น ข้อมูลก็จะถูกทยอยส่งออกไปทีละบิตจนครบทั้งเวิร์ดในสัญญาณเพียงเส้นเดียว แต่ในการใช้งานจริงจะต้องมีสัญญาณอีกเส้นเป็นสัญญาณกราวด์(Ground) ดังนั้นเมื่อเราส่งข้อมูลในแบบอนุกรมเราจะสามารถใช้สายสัญญาณอย่างน้อยที่สุดเพียง 2 เส้นในขณะที่ส่งข้อมูลแบบขนานจะต้องใช้อย่างน้อยเท่ากับจำนวนบิตบวกกับสายสัญญาณอย่างน้อยที่สุด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เพียง 2 เส้น การส่งข้อมูลแบบอนุกรมนั้นจะสามารถส่งข้อมูลได้ไกลกว่ามาก เช่น ถ้าส่งตามมาตรฐานของ RS-232 ที่จะกล่าวต่อไปในภายหลังจะสามารถส่งได้ไกลถึง 30 ถึง 40 ฟุต

รูปแบบการติดต่อสื่อสารแบบอนุกรมแบ่งออกเป็น 3 แบบ คือ

**3.4.1.1 แบบ Simplex**

เป็นการสื่อสารไปในลักษณะทิศทางเดียวตลอดเวลา โดยมีอุปกรณ์ที่มีผู้รับและผู้ส่งตายตัวนั้น ซึ่งจะมีที่ใช้ไม่มากนัก เช่น การสื่อสารข้อมูลระหว่างเครื่องคอมพิวเตอร์กับเครื่องพิมพ์ เป็นต้น

**ภาพที่ 3.3** แสดงการสื่อสารข้อมูลอนุกรมแบบ Simplex

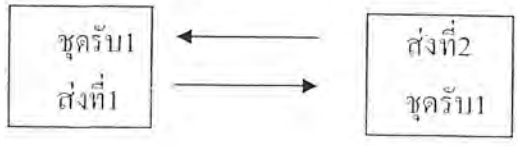
**3.4.1.2 แบบ Half Duplex**

มีทิศทางการสื่อสารในลักษณะที่สลับกันเป็นผู้ส่งและผู้รับพร้อมกันไป

**ภาพที่ 3.4** แสดงการสื่อสารข้อมูลอนุกรมแบบ Half Duplex

**3.4.1.3 แบบ Full Duplex**

มีทิศทางการสื่อสารในลักษณะสัญญาณรับทิศทางหนึ่งสัญญาณส่งอีกทิศทางหนึ่ง หรือกล่าวได้ว่าสัญญาณรับและส่งจะมีสายตัวนำสัญญาณแยกออกจากกันโดยเด็ดขาด




**ภาพที่ 3.5** แสดงการสื่อสารข้อมูลอนุกรมแบบ Full Duplex

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**ภาพที่ 3.6** แสดงการติดต่อสื่อสารข้อมูลแบบอนุกรม

**3.4.2 แบบขนาน**

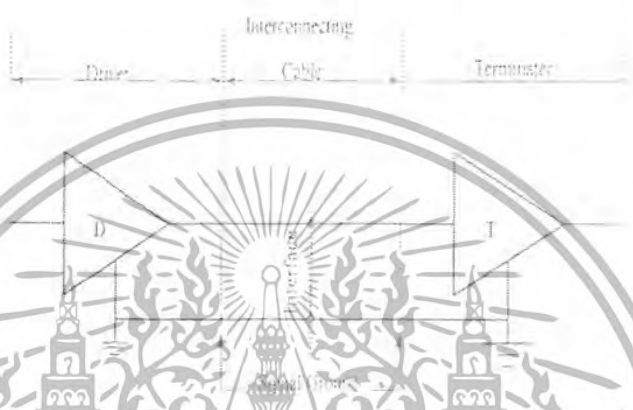
สื่อสารแบบขนานก็คือข้อมูลทุก ๆ บิตในแต่ละเวิร์ดจะถูกส่งออกไปพร้อม ๆ กันขึ้นอยู่กับว่าเวิร์ดดังกล่าวมีขนาดเท่าไร โดยทั่วไปก็คือ 1 ไบท์ หรือ 8 บิต นั่นเอง การส่งข้อมูลแบบขนานนี้จะมี ข้อจำกัดทางด้านระยะทางโดยทั่วไปจะส่งได้ในระยะไม่เกิน 3-5 ฟุต เท่านั้น ทั้งนี้ขึ้นอยู่กับอัตราความเร็วที่ใช้ในการส่งข้อมูลด้วยอัตราการส่งสูงก็จะได้ระยะทางที่สั้นลง การส่งข้อมูลแบบนั้นนิยมในระบบที่ต้องการความเร็วสูงมาก ๆ แต่อุปกรณ์อยู่ไม่ห่างกันมากนักการส่งข้อมูลแบบขนานจะต้องใช้อย่างน้อยเท่ากับจำนวนเท่ากับจำนวนบิตบวกกับสายสัญญาณระดับแรงดัน Ground อีก 1 เส้น

**ภาพที่ 3.7** แสดงการสื่อสารข้อมูลแบบขนาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 3.5 มาตรฐานในการสื่อสารข้อมูลแบบอุนกรม

**3.5.1 มาตรฐานสัญญาณอุนกรมแบบ RS-232C** มาตรฐานการสื่อสารข้อมูลแบบอนุกรมที่กำหนดโดย EIA (Electronics Industries Association) มาตรฐาน RS-232C ได้ถูกตีพิมพ์ในปี คศ 1969 ตัวอักษร RS แทน Recommend Standard 232 แทนหมายเลขของมาตรฐาน ส่วนอักษร C แสดงให้รู้ว่ามาตรฐานได้รับการแก้ไขกี่ครั้ง




ภาพที่ 3.8 แสดงลักษณะสมบัติทางไฟฟ้าของการอินเทอร์เฟสแบบ RS232C

**3.5.2 ลักษณะสมบัติทางไฟฟ้าของการอินเทอร์เฟสแบบ RS232C**

1. ถูกออกแบบให้ใช้กับอุปกรณ์พวกสัญญาณ Discrete
2. ใช้การอินเทอร์เฟสแบบ Unbalanced
3. ในแต่ละวงจรใช้ลวดนำในการนำสัญญาณ 1 เส้น และมีสายรวมของทุกวงจรอีกหนึ่งเส้น
4. อัตราเร็วในการส่งข้อมูลมีค่า $<$ 20 กิโลบิทต่อวินาที (Kbps)
5. ระยะสูงสุดที่ใช้ในการส่งข้อมูลามีค่า $<$ 15 เมตร
6. ทำให้เกิด Crosstalk ที่มีค่ามาก

**3.5.3การสื่อสารข้อมูลแบบอนุกรมมาตรฐาน RS-422A**

ในการออกแบบระบบสื่อสารข้อมูลจากที่ผ่านมาจนถึงปัจจุบันได้มีการพยายามที่จะออกแบบให้การสื่อสารข้อมูลได้รวดเร็วขึ้นและมีระยะในการสื่อสารที่มากขึ้นด้วย ซึ่งที่ผ่านมา การสื่อสารข้อมูลตามมาตรฐาน RS-232C ได้ออกแบบเพื่อใช้เชื่อมต่อกับโมเด็มเท่านั้น จึงไม่ได้คำนึงถึงความเร็วและระยะทางในการสื่อสาร ต่อมาได้มีมาตรฐานในการสื่อสารข้อมูลใหม่ที่ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ออกแบบมาเพื่อรองรับความต้องการของผู้ใช้งานที่ต้องการให้การรับส่งข้อมูลได้ระยะทางไกลและรวดเร็ว มาตรฐานนี้คือ RS-422A ซึ่งการที่มาตรฐานสื่อสารนี้สามารถรับส่งข้อมูลได้ไกลและรวดเร็วขึ้นเนื่องมาจาก หลักการที่ใช้สัญญาณเป็นแบบดิฟเฟอเรนเชียลดังแสดงในรูปที่ 5-8 ซึ่งหลักการก็คือสัญญาณที่รับส่งจะเป็นการเปรียบเทียบระหว่างสัญญาณ 2 เส้นเทียบกับมาตรฐาน RS-232C ที่สัญญาณทุกสัญญาณจะเทียบกับกราวด์ ซึ่งในการสื่อสารในระยะทางไกลๆ แล้วสัญญาณจะถูกลดทอนไปและเมื่อสัญญาณจะถูกลดทอนถึงจุดๆ หนึ่งสัญญาณนั้นก็จะผิดพลาดไปจากความเป็นจริง ทำให้การรับส่งข้อมูลเกิดผิดพลาดขึ้น แต่สำหรับสัญญาณแบบดิฟเฟอเรนเชียล การลดทอนของสัญญาณก็จะไปลดทอนทั้งสองสายด้วยค่าที่เท่ากันหรือใกล้เคียงกันและความแตกต่างของสัญญาณทั้ง 2 เส้น จากตัวส่งไปยังตัวรับก็จะยังคงมีค่าเดิมหรือเปลี่ยนแปลงน้อย จึงทำให้ผลของการติดต่อกับตัวรับได้ถึง 10 ตัว ดังแสดงตารางในตารางที่ 3.1 การเปรียบเทียบมาตรฐานการสื่อสารข้อมูล

ภาพที่ 3.9 แสดงโครงสร้างของการสื่อสารข้อมูลแบบอนุกรมตามมาตรฐาน RS-422A

### 3.5.4 การสื่อสารข้อมูลแบบอนุกรมตามมาตรฐาน RS-485

การสื่อสารข้อมูลตามมาตรฐานที่กล่าวมาข้างต้นคือ RS-232C นั้นจะเป็นมาตรฐานการสื่อสารข้อมูลในแบบที่ใช้สื่อสารระหว่างอุปกรณ์ต่ออุปกรณ์ หรือ จุดต่อจุด (Point-to-Point) ส่วน RS-422A นั้นเป็นมาตรฐานที่พัฒนามาจาก RS-232C ให้ได้ระยะทางไกลขึ้นและอัตราการสื่อสารเพิ่มขึ้น แต่ก็ยังเป็นการสื่อสารข้อมูลจากอุปกรณ์หนึ่งตัวไปยังอุปกรณ์อื่น ๆ ได้สูงสุด 10 ตัวเท่านั้นไม่สามารถส่งย้อนกลับจากอุปกรณ์ตัวรับมาตัวส่งได้ หรือกล่าวได้ว่าการสื่อสารข้อมูลตามมาตรฐาน RS-422A นั้นเป็นการสื่อสารข้อมูลแบบ Simplex คือทิศทางของข้อมูลเป็นแบบทางเดียวตลอดเวลา ดังนั้นถ้าต้องการออกแบบระบบให้เป็นลักษณะโครงข่ายข้อมูลก็จะไม่สามารถทำได้ จึงมีการพัฒนามาตรฐานการสื่อสารข้อมูลขึ้นใหม่เพื่อรองรับความต้องการนี้ คือ มาตรฐาน RS-422A แต่สามารถ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สื่อสารข้อมูลได้ทั้ง 2 ทิศทางในสายสัญญาณเพียงคู่เดียว ซึ่งก็คือการสื่อสารข้อมูลแบบ Half-Duplex จากผลของการใช้สัญญาณในลักษณะดิฟเฟอเรนเชียลนี้ ทำให้ระยะทางและความเร็วในการสื่อสารข้อมูลมีค่าสูง เช่นเดียวกับมาตรฐานการสื่อสารข้อมูล RS-422S แต่มาตรฐาน RS-485 แต่สามารถที่จะสื่อสารระหว่างอุปกรณ์ทั้งการส่งของอุปกรณ์ได้สูงสุด 32 ตัวหรืออาจกล่าวได้ว่าการสื่อสารค่าเปรียบเทียบในตารางที่ 3.1 และแสดงโครงสร้างในรูป

ภาพที่ 3.10 แสดงโครงสร้างของการสื่อสารข้อมูลแบบอนุกรมตามมาตรฐาน RS-485

ตารางที่ 3.1 แสดงค่าเปรียบเทียบคุณสมบัติทางไฟฟ้าของ RS232C, RS-422A, RS-485

| พารามิเตอร์ | RS-232-C | RS-422-A | RS-485 |
|---|---|---|---|
| โหมดการทำงาน | Single-ended | Differential | Differential |
| จำนวนของตัวรับและส่งที่ยอมรับได้ | 1 ตัวส่ง<br>1 ตัวรับ | 1 ตัวส่ง<br>10 ตัวรับ | 32 ตัวรับ<br>32 ตัวส่ง |
| ความยาวของคู่สายสูงสุด | 50 ฟุต | 4000 ฟุต | 4000 ฟุต |
| อัตราการส่งข้อมูลสูงสุด | 20k bps | 10 M bps | 10 M bps |
| Maximum common mode voltage | 2.5 v | 6v<br>-2.5 v | 12 v<br>-7 v |

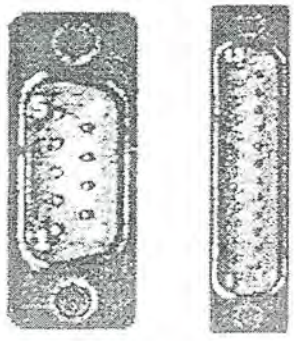
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

| | | | |
|---|---|---|---|
| Driver output | 5 v ต่ำสุด<br>15 v สูงสุด | 2 v ต่ำสุด | 1.5 v ต่ำสุด |
| Driver load(ohm) | 3k ถึง 7k | 100 ต่ำสุด | 60 ต่ำสุด |
| Driver slew rate | 30v/us สูงสุด | NA | NA |
| กระแสลิมิตเมื่อเอาท์พุทลัดวงจร | 500mA ลัดวงจรกับ VCC หรือ GND | 150mA ลัดวงจรกับ GND | 150mA ลัดกับ GND 250mA ลัดกับ12v |
| ค่าความต้านทานเอาท์พุทขององตัวส่ง (โอห์ม) | NA(power on)<br>300 (power off) | NA(power on)<br>60k (power off) | 120k (power on)<br>120k (power off) |
| ค่าความต้านทานอินพุทตัวรับ | 3k ถึง 7k ohm | 4k ohm | 120 ohm |
| ความไวของตัวรับ | 3v | 200mV | 200mV |

### 3.6 Wring Ports

เราใช้พอร์ทอนุกรม ขนาด 9 Pin เป็นรูปสี่เหลี่ยมคางหมูหรือ 25 pin ก็ใช้เชื่อมต่อระหว่าง PLC กับคอมพิวเตอร์ส่วนบุคคล แต่ในที่นี้ขอยกตัวอย่าง ขนาด 9 pin

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 3.11 แสดง Port และการเชื่อมต่อ Port ระหว่าง PLC กับคอมพิวเตอร์ส่วนบุคคล

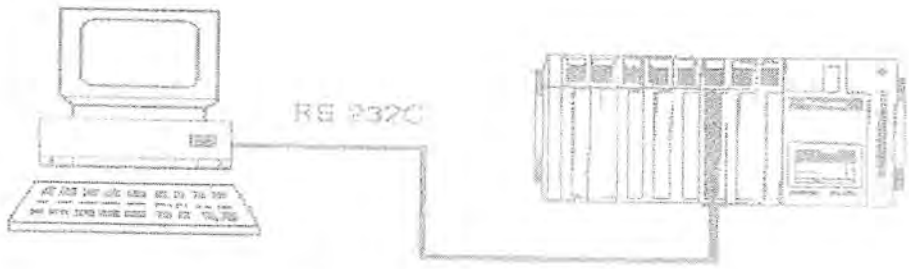
## 3.7 รูปแบบการสื่อสารข้อมูลของเครื่องควบคุมแบบตรรกะที่โปรแกรมได้

การต่อวงจรในการติดต่อสื่อสารข้อมูลของเครื่องควบคุม PLC/PC อาจแบ่งออกได้ 3 แบบด้วยกันคือ

### 3.7.1 การต่อวงจรแบบจุดต่อจุด

ในการติดต่อสื่อสารข้อมูลแบบจุดต่อจุด จะใช้มาตรฐานการสื่อสารแบบ RS 232 ซึ่งเป็นการติดต่อในระยะทางที่ไม่ไกลมาก เช่น การติดต่อระหว่างเครื่องควบคุม PLC/PC กับเครื่องคอมพิวเตอร์ เพื่อทำความการตรวจสอบสภาวะการทำงาน หรือ การควบคุมจากคอมพิวเตอร์หรือสนับสนุนการทำงานอื่นๆ การติดต่อระหว่างเครื่องควบคุม PLC/PC กับอุปกรณ์อื่น เช่น โมเด็ม (MODEM) อุปกรณ์ควบคุมสัมผัสหน้าจอ ( Touch Screen) คีย์บอร์ด (Key board) เครื่องอ่านรหัสแถบ (Bar-code Reader) เครื่องพิมพ์ เครื่องบันทึก(Recorder) เครื่องรายงานผล (Reporter) เป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้




**ภาพที่ 3.12** แสดงระบบการติดต่อแบบจุดต่อจุด

### 3.7.2 การต่อวงจรแบบหลายจุด (Multi point)

#### 3.7.2.1 เป็นการติดต่อแบบอินพุทและเอาท์พุทระยะไกล (Remote I/O System)

เป็นการเชื่อมโยงสื่อสารข้อมูลในลักษณะ ที่ตัวประมวลผลของเครื่องควบคุมหลักกับปลายทางอุปกรณ์ที่ถูกควบคุมอยู่ไกลกัน และอาจมีมากกว่า 1 จุด ดังนั้นจำเป็นจึงมีความจำเป็นที่จะมีการเชื่อมโยงในระยะไกล ในการติดต่อลักษณะนี้จะใช้มาตรฐานการสื่อสารแบบ RS-485 บนตัวนำสองเส้น โดยที่ตัวควบคุมหลัก (Master) จะทำการรับอินพุททั้งจากพื้นที่ของตัวควบคุม และหน่วยระยะไกล (Slave) จากนั้นจะทำการประมวลผล แล้วส่งเอาท์พุทไปยังส่วนต่างๆ โดยที่อินพุทและเอาท์พุทของหน่วยระยะไกลจะจองพื้นที่ของรีเลย์ภายใน (Internal Relay) ของตัวควบคุมหลัก โดยการเซทระบบนั้นจะแตกต่างกันไปตามบริษัทผู้ผลิต

**ภาพที่ 3.13** แสดงการเชื่อมโยงแบบอินพุทและเอาท์พุทระยะไกล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**ภาพที่ 3.14** แสดงการจำลองพื้นที่ของตัวควบคุมหลักขณะทำงานปกติ

**ภาพที่ 3.15** แสดงการจำลองพื้นที่ของตัวควบคุมหลักขณะเมื่อมีการทำงานร่วมกับหน่วยอินพุทเอาท์พุทระยะไกล

**3.7.2.2 การติดต่อแบบพีซีลิงค์ (PC Link System)**

เป็นการเชื่อมโยงสื่อสารข้อมูลในลักษณะของการแบ่งพื้นที่ในการเขียนระหว่างเครื่องควบคุม PLC/PC แต่ละตัว เครื่องควบคุม PLC/PC แต่ละตัวสามารถรับรู้ข่าวสาร ซึ่งกันและกันได้ มาตรฐานในการสื่อสารข้อมูลชนิดนี้มักจะเป็นแบบ RS422A หรือ RS485 หรือเป็นรูปแบบการสื่อสารผ่าน สายใยแก้วนำแสง (Fiber optic) หรือผ่านสายโคแอกเชียล หรือสายคู่ตีเกลียว โดยจะมีตัว Link Adapter เป็นตัวแปลงรูปแบบข้อมูลให้เป็นมาตรฐานตรงกันอาจจะมีการเชื่อมโยงกับเครื่องคอมพิวเตอร์ เข้ามาในระบบการสิงค์ เพื่อสนับสนุนการทำงานของเครื่องควบคุม PLC/PC ซึ่งจะเรียกว่าโฮสคอมพิวเตอร์ ( Host Computer) มีลักษณะการต่อวงจรดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**ภาพที่ 3.16** แสดงการต่อแบบ Serial link

**ภาพที่ 3.17** แสดงการต่อแบบ Parallel Link

ตัวอย่างการจัดแบ่งพื้นที่ของเครื่องควบคุม PLC/PC กำหนดให้มีเครื่องควบคุมจำนวน 4 ชุด เชื่อมโยงในเครื่องควบคุมแต่ละเครื่องจะมีพื้นที่ของรีเลย์ที่ทำหน้าที่ในการเชื่อมโยงข้อมูล (Link –Relay) ถ้ามีเครื่องควบคุมที่เชื่อมต่อในระบบมากพื้นที่ก็จะถูกแบ่งออกไปตามจำนวน เช่น 4 ชุด พื้นที่จะถูกแบ่งออกเป็น 4 ส่วน เช่นเครื่องควบคุมมีพื้นที่ของลิงค์รีเลย์จำนวน 64 Word ก็จะถูกแบ่งเป็นส่วนละ 16 Word ส่วนแรกจะเป็นพื้นที่ของเครื่องควบคุมตัวที่ 1 ส่วนที่สองจะเป็นของเครื่องควบคุมตัวที่ 2 และสามตามลำดับ โดยพื้นที่ส่วนอื่นจะถูกกันไว้ไม่ให้สามารถเขียนข้อมูลลงไปได้ (Read Only) ยกเว้นพื้นที่ของตัวเอง ดังนั้นถ้า PLC/PC เครื่องที่ 1 ต้องการรับข้อมูลของ PLC เครื่องที่ 3 ก็สามารถอ่านได้จากพื้นที่ของเครื่องควบคุมส่วนที่สาม และถ้าต้องการส่งข้อมูลให้ก็เขียนลงในพื้นที่ของส่วนแรก แล้ว PLC ตัวที่ 3 ก็จะทำการอ่านข้อมูลในส่วนแรก ก็จะสามารถทราบข้อมูลของเครื่องควบคุม PLC ตัวที่ 1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**ภาพที่ 3.18** การแบ่งพื้นที่สำหรับ PC Link System

**3.7.2.3 การต่อวงจรสื่อสารแบบเครือข่ายท้องถิ่น (Local area Network for PLC/PC)**

การเชื่อมโยงในลักษณะของเครือข่ายท้องถิ่นนี้เป็นระบบขนาดใหญ่ประกอบด้วยอุปกรณ์ หลายชนิดและหลายวงรอบด้วยกัน ซึ่งจะเป็นลักษณะเครือข่ายแบบวงแหวน (Ring Network) ในแต่ละวงรอบ สามารถที่จะเชื่อมโยงไปยังวงรอบอื่นๆ ได ซึ่งจะมีความซับซ้อนในระบบการติดต่อสื่อสารมากขึ้น รูปแบบสายนำสัญญาณจะมีทั้งสายใยแก้วนำแสง (Fiber optic) และสายตัวนำในแบบต่างๆ ในระบบโครงข่ายก็จะมีเครื่องคอมพิวเตอร์ ส่วนบุคคลทำหน้าที่เป็นตัวจัดการข้อมูลเช่นเดียวกันกับเครื่องเซิฟเวอร์ (Computer Server) บนเครือข่ายของคอมพิวเตอร์ ในปัจจุบันได้มีการพัฒนาระบบเครือข่ายเครื่องควบคุมอย่างต่อเนื่องจนความสามารถใกล้เคียงกับระบบเครือข่ายคอมพิวเตอร์ เช่น การเชื่อมต่อกับระบบอินเตอร์เน็ท เป็นต้น

**ภาพที่ 3.19** แสดงระบบการติดต่อสื่อสารแบบเครือข่ายท้องถิ่น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.8 การกำหนดตัวพารามิเตอร์ของ Host ling

เมื่อพิจารณาการส่งข้อมูลในแบบอนุกรมให้ดีจะพบว่า ปัญหาหนึ่งที่ควรจะเกิดขึ้นอยู่เสมอ ก็คือการตัดสินใจว่าข้อมูลที่ได้รับนั้นมีจะเริ่มต้นที่ใด ดังนั้นจึงมีการกำหนดข้อตกลงในการสื่อสารขึ้นเพื่อแก้ปัญหานี้ ข้อตกลงดังกล่าวเราเรียกว่า โปรโตคอล (Protocol) ของการสื่อสารข้อมูลแบบอนุกรม สามารถแบ่งออกเป็น 2 ประเภทใหญ่ๆ คือ โปรโตคอลสำหรับการสื่อสารข้อมูล

* แบบซิงโครนัส (Synchronous) ข้อมูลจะถูกส่งออกไปอย่างสม่ำเสมอช่วงเวลาระหว่างบิต และระหว่างเวิร์ดก็จะมีค่าเท่ากันเสมอ ดังนั้นในการสื่อสารข้อมูลอนุกรมในแบบซิงโครนัสจึงต้อง มีสัญญาณเพิ่มเติมเพื่อกำกับการส่งว่าควรจะส่งเมื่อใดและควรจะหยุดเมื่อใดระบบที่เป็นซิงโคร นัสจะเป็นระบบที่มีความเร็วสูงแต่ก็ยังต่ำกว่าการสื่อสารแบบขนาน

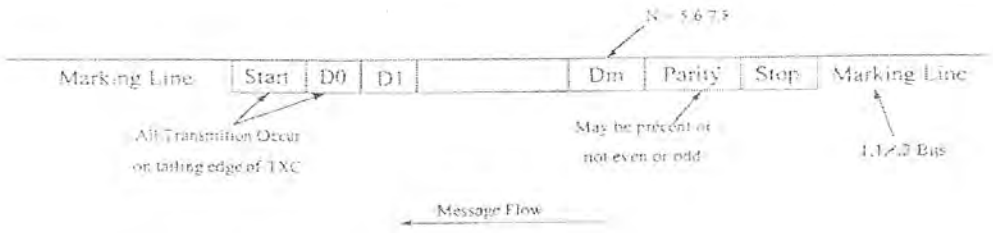
* แบบอะซิงโครนัส (Asynchronous) ถือเป็นหัวใจของการสื่อสารข้อมูลผ่านทางสายโทรศัพท์ในปัจจุบัน การสื่อสารแบบนี้ช่วงระยะเวลาระหว่างบิตจะมีค่าเท่ากันเช่นเดียวกับซิงโครนัส แต่จะมีระยะห่างระหว่างเวิร์ดนั้นแตกต่างกันออกไปเป็นกี่วินาที นาที ชั่วโมง หรือวัน เป็นต้น ขึ้นอยู่กับทางฝ่ายรับสามารถรอได้หรือไม่เท่านั้น เมื่อไม่มีข้อกำหนดทางด้านระยะเวลาระหว่างเวิร์ดแล้ว ทางผู้ส่งและผู้รับจะเข้าใจตรงกันได้อย่างไรว่าที่ใดคือจุดเริ่มต้นและจุดสิ้นสุดของแต่ละเวิร์ด

เพื่อแก้ปัญหานี้จึงมีการกำหนดข้อตกลงเกี่ยวกับรูปแบบของข้อมูลที่จะส่งให้ทางผู้รับสามารถเข้าใจว่าจุดใดเป็นจุดเริ่มของเวิร์ด จึงมีข้อกำหนดดังนี้

| | |
|---|---|
| Start Bits | บิตเริ่มต้น แต่ละเวิร์ดจะต้องขึ้นต้นด้วยบิตนี้ ซึ่งจะต้องมีข้อมูลเป็นลอจิก 0 เสมอ |
| Data Length | บิตข้อมูล มีความยาว 5 ถึง 8 บิต |
| Stop Bits | ทำหน้าที่บอกทางผู้รับว่า ขณะนี้ข้อมูลที่ทางผู้รับได้รับนั้นครบเวิร์ดแล้วบิตสุดท้ายนี้ถูกกำหนดให้เป็นลอจิก 1 เสมอเพื่อให้ระบบสามารถตรวจสอบบิตเริ่มต้นได้ |
| Parity | ทำหน้าที่เป็นบิตสำหรับตรวจสอบความถูกต้อง |
| Baud rate | ความไว เช่น 9,600 bps |

ในการใช้งานทั่วไปเรานิยมใช้กันอยู่เพียง 2 รูปแบบคือ 7E1 (7Data bit, Even Parity, 1 Shop Bit) เหมาะกับสภาพที่มีสัญญาณรบกวนมาก และ 8N1 (8Data bit, No Parity, 1 Shop Bit) เหมาะกับสภาพที่มีการรบกวนของสัญญาณต่ำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้




**ภาพที่ 3.20** แสดงแบบมาตรฐานการสื่อสารข้อมูลแบบอะซิงโครนัส

## 3.9 การส่งสัญญาณเพื่อควบคุมการทำงาน

การส่งสัญญาณการควบคุมผ่านพอร์ทอนุกรม RS-232 จะอาศัยแบบข้อตกลงในการสื่อสาร (Protocol) ของเครื่องควบคุมพีแอลซี โดยทั่วไปจะเป็นลักษณะการตอบกันระหว่างเครื่องควบคุมกับอุปกรณ์ภายนอก ซึ่งอุปกรณ์ภายนอกซึ่งเป็นเครื่องคอมพิวเตอร์จะเป็นฝ่ายถามก่อนแล้วจะส่งบล็อกคำสั่ง (Command Block) ออกไป จากนั้นเครื่องควบคุมก็จะทำการตรวจสอบแล้วส่งบล็อกตอบสนองกลับมา (Response Block) ตัวอย่าง การสื่อสารข้อมูลกันระหว่างเครื่องคอมพิวเตอร์กับเครื่องควบคุมพีแอลซี

**ภาพที่ 3.21** แสดงการส่งสัญญาณควบคุมผ่านพอร์ทอนุกรม RS-232

## 3.10 รูปแบบของบล็อก (BLOCK FORMAT)

ลักษณะของบล็อกของเครื่องควบคุมพีแอลซี แต่ละผู้ผลิตจะแตกต่างกันไป แต่จะมีพื้นฐานเดียวกัน ซึ่งบล็อกคำสั่งที่ใช้มีลักษณะดังนี้

**ภาพที่ 3.22** แสดงแบบของบล็อกคำสั่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. หมายเลขเครื่องในการเชื่อมต่อเป็น โครงข่ายแบบหลายจุดนั้น เครื่องควบคุมที่เชื่อมต่ออยู่ในระบบจะมีมากกว่า 1 เครื่อง การกำหนดว่าต้องการส่งฐานข้อมูลให้กับเครื่องควบคุมตัวใดเช่น @ 05 คือหมายเลขที่ 05

2. ส่วน HEADER เป็นส่วนของคำสั่งหลักที่จะกำหนดว่าต้องการกระทำกับข้อมูลส่วนใด เช่น ต้องการอ่านข้อมูลอินพุตจากเครื่องควบคุมจะใช้ตัวอักษรพิมพ์ใหญ่ " RR" ต้องการเขียนข้อมูลให้กับเครื่องควบคุมจะใช้ "WR" เป็นต้น

3. ส่วน TAXT เป็นส่วนของข้อมูล เช่น คำที่อ่านได้จากอินพุต หรือ ค่าที่จะต้องเขียนลงในพื้นที่ต่างๆ

4. ส่วนของ FCS เป็นส่วนของการควบคุมความผิดพลาดของข้อมูล จะได้จากการคำนวณ

5. ส่วนของ TERMINAL (*) ซึ่งเป็นส่วนที่ใช้ในการตรวจสอบ ความผิดพลาดของการสื่อสารเป็นการเปลี่ยนข้อมูล 8 บิต เป็น 2 ตัวอักษรข้อมูล ASCII นำมาทำการ Exclusive - Or (XOR) โดยเริ่มจาก @ จนถึงตัวอักษรตัวสุดท้ายของ TEXT เช่น

**ภาพที่ 3.23** แสดงการคำนวณ FCS

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# Appendix G
## Extended ASCII

### Programming Console Displays

| Bits 0 to 3 | | Bits 4 to 7 | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| BIN | | 0000 | 0001 | 0010 | 0011 | 0100 | 0101 | 0110 | 0111 | 1010 | 1011 | 1100 | 1101 | 1110 | 1111 |
| | HEX | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | A | B | C | D | E | F |
| 0000 | 0 | NUL | DLE | SPACE | 0 | @ | P | ` | p | | 0 | @ | P | ` | p |
| 0001 | 1 | SOH | $DC_1$ | ! | 1 | A | Q | a | q | | 1 | A | Q | a | q |
| 0010 | 2 | STX | $DC_2$ | " | 2 | B | R | b | r | " | 2 | B | R | b | r |
| 0011 | 3 | ETX | $DC_3$ | # | 3 | C | S | c | s | # | 3 | C | S | c | s |
| 0100 | 4 | EOT | $DC_4$ | $ | 4 | D | T | d | t | $ | 4 | D | T | d | t |
| 0101 | 5 | ENQ | NAK | % | 5 | E | U | e | u | % | 5 | E | U | e | u |
| 0110 | 6 | ACK | SYN | & | 6 | F | V | f | v | & | 6 | F | V | f | v |
| 0111 | 7 | BEL | ETB | ' | 7 | G | W | g | w | ' | 7 | G | W | g | w |
| 1000 | 8 | BS | CAN | ( | 8 | H | X | h | x | ( | 8 | H | X | h | x |
| 1001 | 9 | HT | EM | ) | 9 | I | Y | i | y | ) | 9 | I | Y | i | y |
| 1010 | A | LF | SUB | * | : | J | Z | j | z | * | : | J | Z | j | z |
| 1011 | B | VT | ESC | + | ; | K | [ | k | { | + | ; | K | [ | k | { |
| 1100 | C | FF | FS | , | < | L | \ | l | \| | , | < | L | \ | l | \| |
| 1101 | D | CR | GS | - | = | M | ] | m | } | - | = | M | ] | m | } |
| 1110 | E | SO | RS | . | > | N | ^ | n | ~ | . | > | N | ^ | n | ~ |
| 1111 | F | SI | US | / | ? | O | _ | o | DEL | / | ? | O | _ | o | DEL |

ตารางที่ 3.2 แสดงตารางของ ASCII

## 3.11 ระบบเน็ทเวิร์ก

ประเภทของเน็ทเวิร์ก

1. LAN (Local Area Network) เป็นกลุ่มของ PC ที่เชื่อมต่อกันในพื้นที่จำกัด

2. Man (Metropolitan Area Network) เป็นการนำระบบ LAN หลายๆ LAN ที่มีพื้นที่อยู่ใกล้เคียงกันมาเชื่อมต่อกันให้มีขนาดใหญ่ขึ้น

3. Wan (Wide Area Network) เป็นกลุ่มของ PC ที่เชื่อมต่อกันแบบกว้างขวางอาจเป็นภายในประเทศหรือระหว่างประเทศเป็นการใช้หลาย ๆ LAN และหลายๆ Man ซึ่งอยู่พื้นที่เชื่อมต่อเข้าหากันตามหน้าที่

### 3.11.1 ความหมายและความสำคัญของ Local Area Network

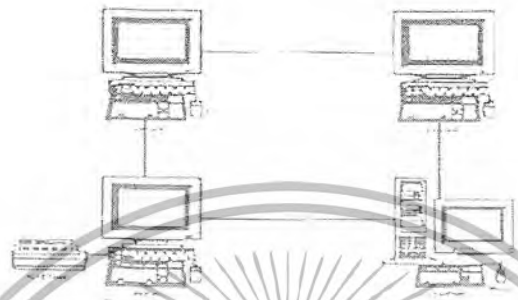
โลคัสแอเรียเน็ตเวิร์ก หรือ LAN มีความหมายตามคำนิยามของ IEEE หรือ ISO ก็คือโครงข่ายที่เป็นไปได้สำหรับการส่งข้อมูลที่มีความเร็วขนาดกลาง ไปจนกระทั่งความเร็วสูงซึ่งถูกจำกัดขอบเขตอยู่ภายในสำนักงาน หรืออาคารหนึ่ง ๆ

สำหรับโครงสร้างของ LAN นั้นจะเป็นโครงข่ายที่เชื่อมต่อโฮสคอมพิวเตอร์เวิร์คโปรเซสเซอร์ไมโครคอมพิวเตอร์ หรือโทรสาร ที่กระจายภายในอาคารหรืออาเขตเดียวกันของสำนักงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หรือโรงงานด้วยความเร็วในการส่งประมาณ 1Mbps – 100Mbps โดยมีจุดมุ่งหมายเพื่อการรีสอร์ต (Resource) เช่น ข่าวสารหรือ อุปกรณ์ร่วมกัน

LAN จะมีพื้นที่เป้าหมายของโครงข่ายค่อนข้างแคบ นอกจากนี้รูปแบบการติดตั้งโดยทั่วไปจะคล้ายคลึงกันและมีความเสถียรภาพ LAN มีบทบาทเพื่อเพิ่มประสิทธิภาพของงานในลักษณะของโรงงานอัตโนมัติ



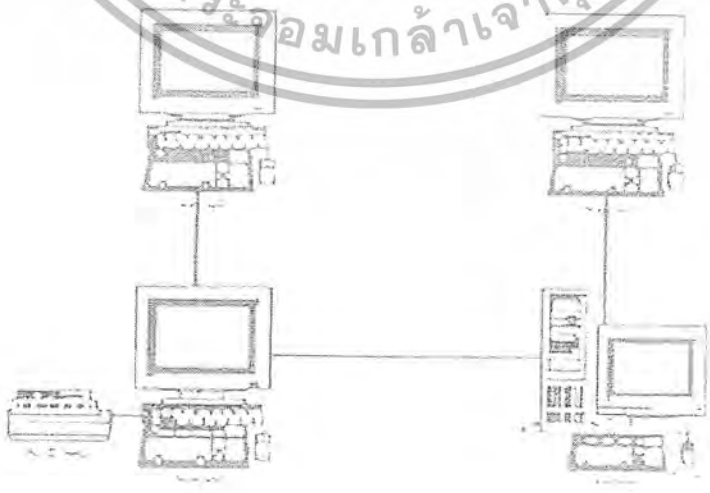
ภาพที่ 3.24 แสดงโครงสร้าง Local Area Network ของระบบ

**3.11.2 รูปแบบของระบบ**

แบ่งออกเป็น 2 รูปแบบ คือ

**3.11.2.1 เพียร์ทูเพียร์เน็ตเวิร์ก (Peer to Peer Network)**

เป็นลักษณะของกลุ่ม PC ที่ทุกเครื่องมีสิทธิเท่าเทียมกันหมด (Peer) ไม่มีเครื่องไหนทำหน้าที่เป็นศูนย์กลางของเน็ตเวิร์กทุกเครื่องเป็นทั้งผู้ใช้และผู้ถูกใช้ สามารถเป็นได้ทั้ง Client และ Server ไม่มีเครื่องไหนมีหน้าที่ดูแลจัดการระบบทั้งหมดแต่ละเครื่องจะเป็นผู้ดูแลข้อมูลและทรัพยากรของตัวเอง



ภาพที่ 3.25 แสดงเพียร์ทูเพียร์เน็ตเวิร์ก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**3.11.2.2 ไคลเอ็นต์ – เซิร์ฟเวอร์ (Client – Sever)**

ในกรณีที่องค์กรของเรามีผู้ใช้มากกว่า 15 – 20 เครื่อง ระบบเน็ตเวิร์กแบบเพียร์ทูเพียร์ดูจะไม่เหมาะสมเสียแล้ว ระบบไคลเอ็นต์ – เซิร์ฟเวอร์จะเป็นระบบที่เหมาะสมกว่า เพราะมีความสามารถของการดูแลควบคุมการใช้งานระบบเน็ตเวิร์กที่มีผู้ใช้งานจำนวนมากได้ดีกว่าระบบเน็ต เวิร์กแบบนี้จะเป็นระบบที่มีศูนย์กลางมีคอมพิวเตอร์ที่ทำหน้าที่ดูแลระบบ อำนวยความสะดวกจัดเก็บข้อมูลรักษาความปลอดภัยให้กับคอมพิวเตอร์ทั้งระบบมีหน้าที่คล้ายๆกับหัวหน้ากลุ่ม เราจะเรียกว่าคอมพิวเตอร์ที่ทำหน้าที่นี้ว่า เซิร์ฟเวอร์ (Sever) ส่วนเครื่องที่เหลือในระบบที่ไม่ได้ทำหน้าที่นี้จะเรียกว่า ไคลเอ็นต์ (Client) หรือ เวิร์กสเตชัน (Workstation) ซึ่งเป็นกลุ่มควบคุมการใช้งานทุกอย่างของ เน็ตเวิร์ก เช่น ข้อมูล เครื่องพิมพ์จะถูกดูแล และแชร์โดยเครื่องเซิร์ฟเวอร์อุปกรณ์ และทรัพยากรทุกอย่างก็จะเชื่อมต่อกับเครื่องเซิร์ฟเวอร์โดยตรง คือเครื่องไคลเอ็นต์ทุกเครื่องจะใช้งานทรัพยากรต่างๆ โดยผ่านทางเซิร์ฟเวอร์

**ภาพที่ 3.26** แสดงไคลเอ็นต์ – เซิร์ฟเวอร์

**3.11.3 โครงสร้างของระบบเน็ทเวิร์ก (Network Topology)**

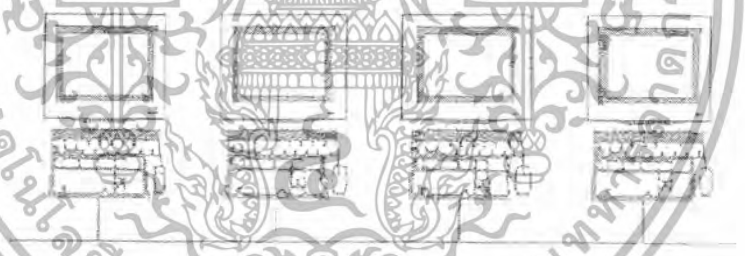
หมายถึงรูปแบบในการจัดวางตำแหน่งของคอมพิวเตอร์ สายเคเบิ้ล และอุปกรณ์อื่น ๆ เพื่อที่จะให้ข้อมูลได้ทำงานตามทิศทางที่เราได้กำหนดไว้ โครงสร้างเน็ทเวิร์กที่ต่างกันมีความต้องการด้านอุปกรณ์ต่างๆ เช่น ฮาร์ดแวร์ ซอฟต์แวร์ ที่แตกต่างกันไปด้วยโดยทั่วไปสามารถแบ่งออกได้เป็น 3 แบบ

- แบบ Bus
- แบบ Ring
- แบบ Star

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**3.11.3.1 แบบ Bus**

เป็นเน็ทเวิร์กที่ง่ายที่สุดและเป็นที่รู้จักอย่างกว้างขวางโดยการเชื่อมต่อ Computer ทุกเครื่องด้วยการใช้สายเคเบิ้ลเป็นสายหลัก เชื่อมต่อคอมพิวเตอร์ทุกเครื่องโดยมีเน็ทเวิร์กการ์ดเป็นตัวเชื่อมระหว่างสายเคเบิ้ลกับคอมพิวเตอร์ ข้อมูลจะถูกส่งออกไปตามสายไปยังคอมพิวเตอร์ทุกๆ เครื่อง ไม่สนใจว่าเครื่องไหนเป็นเครื่องรับ คอมพิวเตอร์แต่ละเครื่องจะตรวจสอบเองว่าข้อมูลที่ถูกส่งออกมานั้นเป็นของตนหรือไม่ ถ้าไม่ จะปล่อยข้อมูลผ่านไป แต่ถ้าใช่ก็จะรับข้อมูลนำไปใช้ ณ เวลาใดเวลาหนึ่ง คอมพิวเตอร์เครื่องเดียวเท่านั้นที่สามารถส่งข้อมูลได้ ในระบบนี้คอมพิวเตอร์ไม่สามารถส่งข้อมูลได้พร้อมๆกันหลายเครื่องในเวลาเดียวกัน เป็นสาเหตุให้ประสิทธิภาพของเน็ทเวิร์กจะน้อยลงเมื่อมีจำนวนคอมพิวเตอร์มากขึ้น สายเคเบิ้ลที่เป็นสายกลางหรือสายหลักที่คอมพิวเตอร์ใช้รับส่งข้อมูลเรียกว่า Backbone สายที่ใช้ส่วนมากจะใช้สายโคแอ็กเชียล Coaxial) แบ่งเป็น 2 ประเภทคือ Thicknet และ Thinet มีลักษณะคล้ายๆ กับสายเคเบิ้ลทีวี (UBC) การใช้จะต้องมีอุปกรณ์ที่ปิดหัวและท้ายของสายเคเบิ้ลด้วยเรียกว่า เทอร์มิเนเตอร์ (Therminater) คอยรับสัญญาณไม่ให้สะท้อนกลับไปซึ่งอาจจะเป็นการรบกวนสัญญาณได้ นับเป็นวิธีที่ง่ายและสะดวกที่ใดในการติดตั้งเน็ตเวิร์กไม่ต้องมีฮาร์ดแวร์มากมาย มีเพียงแค่สายเคเบิ้ลเน็ทเวิร์กกับเทอร์นิเนเตอร์เท่านั้นก็พอแล้ว มักใช้กับเน็ทเวิร์กขนาดเล็ก ซึ่งมีคอมพิวเตอร์ไม่มากนัก



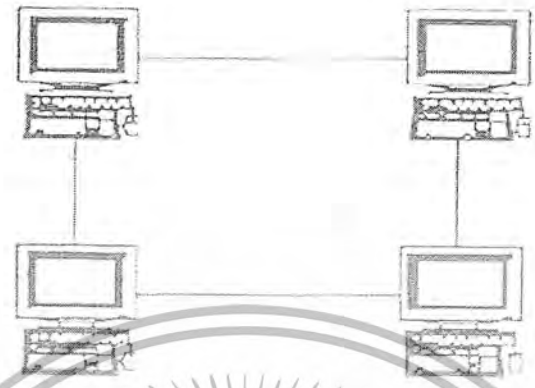
ภาพที่ 3.27 แสดง Bus Network

**3.11.3.2 แบบวงแหวน (Ring)**

เป็นการเชื่อมต่อคอมพิวเตอร์เข้าด้วยกันในลักษณะของรูปวงแหวนโดยจะใช้สายเคเบิ้ล การต่อลักษณะนี้ จะไม่มีจุดเริ่มต้น หรือจุดสุดท้าย ซึ่งการส่งข้อมูลจะวิ่งผ่านคอมพิวเตอร์ทุกเครื่องเป็นรูปวงแหวนในทิศทางเดียวกัน เมื่อคอมพิวเตอร์เครื่องหนึ่งต้องการส่งข้อมูล มันจะทำการใส่ข้อมูล ตำแหน่งและที่อยู่ของเครื่องที่มันต้องการจะส่งข้อมูลไปให้ คอมพิวเตอร์ทุกเครื่องจะรับสัญญาณมาเช็คว่าเป็นข้อมูลของตนเองหรือเปล่า ถ้าไม่ จะส่งไปให้เครื่องต่อๆไป สัญญาณจะวิ่งไปจนกระทั่ง คอมพิวเตอร์เครื่องนั้น และรับข้อมูลนำไปใช้ เนื่องจากสัญญาณจะวิ่งไปเรื่อยๆเป็นวงกลม จึงไม่ต้องการอุปกรณ์ปิดหัวปิดท้าย มักจะใช้กับเน็ทเวิร์กที่มีคอมพิวเตอร์อยู่ไม่ใกล้กันมากนัก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ใช้โทเค็นเป็นสื่อในการนำสัญญาณ โทเค็นจะถูกวิ่งผ่านไปทุกๆ เครื่องเรื่อยๆ จนกว่าเครื่องที่ต้องการส่งข้อมูลจะดึงโทเค็นไปใช้และส่งสัญญาณออกมาเครื่องที่มีโทเค็นเท่านั้นที่สามารถส่งข้อมูลได้



ภาพที่ 3.28 แสดงRing Network

**3.11.3.3 แบบดาว (Star)**

เป็นลักษณะการเชื่อมต่อโดยเชื่อมต่อคอมพิวเตอร์ทุกเครื่อง เข้าสู่อุปกรณ์วนกลางที่เรียกว่า ฮับ (Hub) ข้อมูลหรือสัญญาณจะเดินทางจากเครื่องส่งไปสู่ผู้รับโดยผ่าน ฮับ

**ภาพที่ 3.29** Star Network

**3.11.4 ตัวกลางในการส่ง**

ตัวกลางในการส่งสำหรับ LAN นั้นจะใช้ สายคู่ตีเกลียว โคแอกเชียลเคเบิ้ล หรือ เส้นใยนำแสง ในที่นี้จะกล่าวถึงเฉพาะเส้นใยนำแสง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**3.11.4.1 เคเบิ้ลเส้นใยนำแสง (Fiber Optic)**

โดยทั่วไปแล้วจะเป็นเคเบิ้ลที่มีซิลิก้าออฟ-ติกไฟเบอร์ เป็นตัวนำขนาดเส้นผ่านศูนย์กลาง 0.05-0.1 mm. ในการนำเอาสายเคเบิ้ลใยแสงมาเป็นเส้นทางส่งนั้นจำเป็นต้องมีอุปกรณ์ส่งและรับซึ่งทำหน้าที่แปลงสัญญาณไฟฟ้าจากสถานีให้เป็นสัญญาณแสง สำหรับอุปกรณ์กำเนิดแสง ได้แก่ LED หรือ LD (Laser Diode) ส่วนอุปกรณ์รับแสงนั้นได้แก่ PD (Photo Diode) หรือ APD (Avalanche Photo Diode) การใช้เส้นใยนำแสงเป็นตัวกลางในการส่งนั้น ถ้าเปรียบเทียบกับตัวกลางอื่นๆ แล้วจะมีราคาแพงกว่า แต่จะมีข้อดีตรงที่เป็นเคเบิ้ลที่มีขนาดเล็กและเบา รวมทั้งสามารถใช้ในการส่งย่านความถี่กว้างความเร็วสูง นอกจากจะใช้ในการส่งสัญญาณข้อมูลแล้วยังสามารถใช้ในการประกอบ LAN ที่รวมเอามีเดียรแต่แต่ละชนิด เช่น เสียงและภาพ เป็นต้น และเนื่องจากจะไม่รับอิทธิพลของสัญญาณของสนามแม่เหล็กไฟฟ้าจากภายนอก ดังนั้นคุณภาพในการส่งจะคงที่จึงเหมาะที่จะใช้ในสถานที่ที่มีสัญญาณรับกวนอยู่มาก เช่น ในโรงงาน เป็นต้น แต่ในทางตรงข้ามการเชื่อมต่อโหนดกับสายเคเบิ้ลจะเป็นเรื่องยุ่งยาก ดังนั้นการเพิ่มหรือเคลื่อนย้ายโหนดจะลำบาก ซึ่งเป็นข้อจำกัดของระบบ

วิธีการส่งสัญญาณ LAN นั้นได้แก่วิธีการส่งแบบเบสแบนด์ซึ่งสัญญาณดังเดิมจากสถานี (โดยทั่วไปจะเป็นดิจิตอล) จะถูกส่งในลักษณะเช่นนั้นเข้าไปในแชนเนล ส่วนวิธีการส่งแบบบรอดแบนด์ สัญญาณดังเดิมจะถูกโมดเดท แล้วข้าฟรีเควันซี่มัลติเพลกซ์ (TDM) เป็นกลุ่มสัญญาณเข้าไปในแชนเนล

วิธีการส่งแบบเบสแบนด์ วิธีการส่งสัญญาณแบบเบสแบนด์ จะมีแชนเนลเพียงแชนเนลเดียวเท่านั้นที่สามารถส่งข่าวสารได้ ดังนั้นเพื่อที่จะให้บริการ สถานีสามารถใช้แชนเนลร่วมกันได้ก็ต้องเพิ่มความเร็วในการส่งให้สูงขึ้น และแบ่งเวลาใช้งานให้กับแต่ละสถานี

## 3.12 ลักษณะพิเศษของวิธีการส่งแบบเบสแบนด์นั้นจะเป็นดังต่อไปนี้

1. วงจรรับส่งจะง่ายเนื่องจากไม่ต้องใช้โมเด็ม

2. เนื่องจากรูปแบบการส่งจะใช้สัญญาณดิจิตอล ดังนั้นความกลมกลืนกับการสื่อสารข้อมูลจะมีมาก LAN ที่จะใช้วิธีการส่งแบบเบสแบนด์นั้นโดยทั่วไปจะมีวิธีการซิงโครนัสด้วยวิธีการดึงเอาไทม์มิงจากสัญญาณที่รับเข้ามา ดังนั้นรหัสสัญญาณข่าวสารนั้นจำเป็นต้องมีรูปแบบง่ายสำหรับให้ทางด้านรับสามารถ ดึงเอาไทมมิงออกมา และสามารถที่จะตรวจตราคุณภาพในการส่ง การตรวจหาความผิพลาดของสัญญาณส่งซึ่งโดยทั่วไปแล้วจะใช้รหัสแมนเชสเตอร์และดิฟเฟอร์เรน เทียลแมนเชสเตอร์ เป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 3.13 ระบบไคลเอนต์ / เซอร์ฟเวอร์

ระบบไคลเอนต์ / เซอร์ฟเวอร์ ซึ่งเป็นสถาปัตยกรรมของทางด้าน ระบบคอมพิวเตอร์แบบ Distribute แบบหนึ่งที่นำเอาเครื่องคอมพิวเตอร์มากกว่า 1 เครื่อง มาเชื่อมต่อกันด้วยระบบเครือข่าย (Computer Network) โดยมีจุดประสงค์เพื่อต้องการ ให้เกิดการใช้ข้อมูลร่วมกันระหว่างเครื่องคอมพิวเตอร์ในเครือข่ายนั้น ภายใต้คุณสมบัติต่าง ๆ ดังนี้

เครื่องคอมพิวเตอร์ที่เชื่อมต่อกันเป็นระบบซึ่งจะต้องประกอบด้วยเครื่องคอมพิวเตอร์ที่ทำหน้า ที่เป็น ระบบไคลเอนต์ และ เซอร์ฟเวอร์

ฐานข้อมูลจะถูกจัดเก็บอยู่ในคอมพิวเตอร์ที่ทำหน้าที่เซิร์ฟเวอร์ โปรแกรมที่ทำหน้าที่ในในการเรียกใช้ข้อมูลจากฐานข้อมูล จะทำงานอยู่บนเครื่องคอมพิวเตอร์ที่ทำหน้าที่เป็น ไคลเอนต์ภายในระบบคอมพิวเตอร์แบบ ระบบไคลเอนต์ / เซอร์ฟเวอร์ โดยทั่วไปมักจะประกอบด้วยเครื่องคอมพิวเตอร์ที่ทำหน้าที่เป็นเซิร์ฟเวอร์ 1 เครื่องที่เชื่อมต่อกับเครื่องคอมพิวเตอร์ที่ทำหน้าที่เป็นไคลเอนต์มากกว่า 1 เครื่อง ซึ่งมีจำนวนเท่าใดจะขึ้นออยู่กับปริมาณงานภายในระบบ เครื่องคอมพิวเตอร์ซึ่งทำหน้าที่เป็นเซิร์ฟเวอร์ จะใช้จัดเก็บฐานข้อมูล และโปรแกรม DBMS ที่มีหน้าที่ในการรับคำสั่งจากไคลเอนต์ มาประมวลผล และ นำข้อมูลข่าวสารที่ ไคลเอนต์ ต้องการส่งกลับไปยัง ไคลเอนต์ ดังนั้นจึงมักเรียกส่วนเซิร์ฟเวอร์นี้ว่า ส่วนของโปรแกรม Backend ส่วนเครื่องคอมพิวเตอร์ที่ทำหน้าที่เป็นไคลเอนต์ แต่ละเครื่อง จะใช้สำหรับ Run โปรแกรมที่ทำหน้าที่ในการส่งคำสั่งสำหรับเรียกใช้ข้อมูลจากฐานข้อมูล ไปยัง เซิร์ฟเวอร์ เพื่อให้ เซิร์ฟเวอร์ ส่งข้อมูลข่าวสารที่ต้องการกลับมาและนำไปแสดงผล ให้กับผู้ใช้ ต่อไปดังนั้นจึงมักเรียกส่วน ไคลเอนต์ นี้ว่า ส่วนของโปรแกรม Front-end

ในบางระบบ อาจมีการใช้เซิร์ฟเวอร์ มากกว่า 1 ตัว ซึ่งอาจเนื่องมาจากมีการใช้ฐานข้อมูลหลายฐานข้อมูล และจำเป็นที่จะต้องแยกเก็บใน เซิร์ฟเวอร์ ที่ต่างเครื่องกัน ในกรณีนี้ไคลเอนต์จะต้องสามารถที่จะเชื่อมต่อกับ เซิร์ฟเวอร์ แต่ละ เซิร์ฟเวอร์ ได้ แต่ในการเชื่อมต่อของไคลเอนต์ ไปยังเซิร์ฟเวอร์ ในแต่ละครั้ง จะเกิดขึ้นได้เพียง เซิร์ฟเวอร์เดียว ดังนั้น ผู้ใช้ของระบบในลักษณะนี้ จึงต้องทราบถึง เซิร์ฟเวอร์ ที่จัดเก็บข้อมูลที่ตนต้องการ แต่ก็มีบางกรณี ที่ ไคลเอนต์ก็สามารถใช้งานหลาย ๆ เซิร์ฟเวอร์ ในแบบต่อเนื่องได้ ซึ่งมักเกิดกับฐานข้อมูล ที่มีการแยกข้อมูลในฐานข้อมูลออกเป็นส่วน ๆ และจัดเก็บลงในแต่ละ เซิร์ฟเวอร์ แต่การใช้งานในกรณีนี้จะไม่จัดเป็นระบบคอมพิวเตอร์ในแบบ ไคลเอนต์ / เซอร์ฟเวอร์ เนื่องจาก ระบบมีการทำงานที่ตรงข้ามกับระบบคอมพิวเตอร์ในแบบ ไคลเอนต์ / เซอร์ฟเวอร์ กล่าวคือแต่ละเซิร์ฟเวอร์ จะมอง ไคลเอนต์ เป็นเซิร์ฟเวอร์เดียวแทน เนื่องจากจะต้องทยอยส่งข้อมูลไปให้กับ ไคลเอนต์ แทน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.13.1 ระบบไคลเอนต์ / เซอร์ฟเวอร์ กับฐานข้อมูล

ด้วยมุมมองตามสถาปัตยกรรม ANSI / SPARC ของระบบฐานข้อมูลอาจสามารถแบ่งระบบฐานข้อมูลออกเป็น 2 ส่วนคือ ส่วนของ เซิร์ฟเวอร์ ซึ่งโดยทั่วไปมักเรียกว่า โปรแกรม Backend และส่วนของ ไคลเอนต์ ซึ่งโดยทั่วไปมักเรียกว่า โปรแกรม Front-end ดังรูป

ภาพที่ 3.30 แสดงระบบไคลเอนต์ / เซอร์ฟเวอร์

ส่วนเซิร์ฟเวอร์ ในแง่ของฐานข้อมูล ได้แก่โปรแกรม DBMS ที่ทำหน้าที่สนับสนุนการทำงานต่างๆ ของฐานข้อมูล เช่น การสร้างและกำหนดโครงสร้างของฐานข้อมูล การจัดการข้อมูลภายในฐานข้อมูล การรักษาความปลอดภัยให้กับข้อมูล การรักษาความถูกต้องของข้อมูล ฯลฯ เป็นต้น

ส่วนไคลเอนต์ ในแง่ของฐานข้อมูล ได้แก่โปรแกรมต่างๆ ที่ทำงานร่วมกับโปรแกรม DBMS เพื่อใช้งานฐานข้อมูล โปรแกรมเหล่านี้ อาจพัฒนาขึ้นด้วยภาษาคอมพิวเตอร์ภาษาใดภาษาหนึ่ง เช่นภาษา C , COBOL , FOCUS ฯลฯ เป็นต้น หรืออาจพัฒนาขึ้นด้วย Tool ของตัว DBMS เอง เช่น Tool ที่ใช้ในการสร้างรายงานต่างๆ ภาษา Query Language ตารางที่ใช้แสดงข้อมูลในรูป Spreadsheet ภาษาที่ใช้ในการสร้าง From สำหรับจัดการกับข้อมูลหรือที่เรียกว่า 4GI รูป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาพกราฟิกที่ใช้แสดงข้อมูลทางสถิติ หรือในบาง Tool อาจมีส่วนที่ใช้ในการสร้างโปรแกรมขึ้นใช้งานหรือที่เรียกว่า CASE (Computer-aided Software Engineering) ฯลฯ เป็นต้น

ทั้ง 2 ส่วนของระบบฐานข้อมูลนี้ จะถูกนำไปใช้บนเครื่องคอมพิวเตอร์ที่ต่างกัน เพื่อสนับสนุนการประมวลผลในแบบกระจาย (Distributed Processing) เช่น ระบบคอมพิวเตอร์ในแบบไคลเอนต์ / เซิร์ฟเวอร์ ที่ซึ่งเครื่องคอมพิวเตอร์แต่ละเครื่องถูกเชื่อมโยงกันด้วยระบบเครือข่ายซึ่งส่งผลให้ข้อมูลชุดเดียวกันสามารถส่งถ่ายไปมาระหว่างเครื่องคอมพิวเตอร์เครื่องต่างๆ ที่เชื่อมต่อกันอยู่ภายในเครือข่ายนั้นได้

### 3.13.2 การเขียนโปรแกรมกับระบบไคลเอนต์ / เซอร์ฟเวอร์

ส่งที่โปรแกรมเมอร์ จะต้องคำนึงถึงในการพัฒนาโปแกรมขึ้นใช้ที่อยู่ในระบบคอมพิวเตอร์แบบ ไคลเอนต์ / เซอร์ฟเวอร์ ได้แก่ การมอง เซิร์ฟเวอร์ เป็นเครื่องมือในการเข้าถึงข้อมูล และการมองข้อมูลในระดับ Record เนื่องจาก Request ที่ ไคลเอนต์ ส่งไปให้กับ เซิร์ฟเวอร์ เป็นคำสั่งที่ต้องการได้ผลลัพธ์ในระดับ Record และซึ่งจะต้องอยู่ในรูปของคำสั่งในแบบ Non-rocedural เช่น คำสั่ง SQL ดังนั้น จึงไม่มีการใช้คำสั่งที่ก่อให้เกิดการทำซ้ำ (Loop) หรือเป็นคำสั่งที่ใช้ในการแก้ไข (Update) หรือลบ (Delete) ข้อมูล แต่เมื่อต้องการลดจำนวนของการติดต่อระหว่าง ไคลเอนต์ และเซิร์ฟเวอร์ ให้นำเอา Stored Procedure มาใช้แทนคำสั่ง SQL (Stored Procedure) เป็นคำสั่งที่เรียกใช้ข้อมูลจากฐานข้อมูลเช่นเดียวกับ SQL แต่สามารถนำมาเขียนคำสั่งมารวมเป็นชุดคำสั่งได้รวมทั้งจะถูกจัดเก็บและทำงานอยู่บน เซิร์ฟเวอร์ และจะถูกเรียกใช้โดย ไคลเอนต์

### 3.13.3 สำหรับข้อดีของการใช้ Stored Procedure ประกอบด้วย

- Stored Procedure สามารถปิดบังความซับซ้อนทางด้านการใช้งานส่งผลให้ ผู้ใช้สามารถเรียกใช้ข้อมูลในฐานข้อมูลได้ง่าย ซึ่งเป็นไปตามคุณสมบัติทางด้าน Data Independence
- แต่ละ Stored Procedure สามารถเรียกใช้ร่วมกันหลายๆ ไคลเอนต์
- ผู้ใช้สามารถตรวจสอบความถูกต้องของ Stored Procedure ก่อนนำไปใช้งาน ซึ่งต่างจากคำสั่ง SQL ที่จะไม่สามารถตรวจสอบความถูกต้องก่อนนำไปใช้งานได้ เนื่องจาก เป็นคำสั่งที่อยู่ในรูปของ Request ที่ ไคลเอนต์ ส่งไปยัง เซิร์ฟเวอร์ ให้ทำงาน ณ ขณะนั้น
- Stored Procedure สามารถกำหนดความปลอดภัยได้ดีกว่าคำสั่ง SQL เนื่องจากสามารถกำหนดผู้ที่มีสิทธิ์ในการใช้ Stored Procedure นั้นได้โดยตรง

แต่อย่างไรข้อเสียของการใช้ Stored Procedure ได้แก่ ความแตกต่างในการใช้งาน Stored Procedure ของแต่ละผลิตภัณฑ์ เนื่องจากยังไม่มีการกำหนดมาตรฐานของ Stored Procedure

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.13.4 ระบบไคลเอนต์ / เซิร์ฟเวอร์ ที่สนับสนุนการประมวลผลแบบกระจาย

เนื่องจากการประมวลผลแบบกระจาย เป็นการนำเอาคอมพิวเตอร์หลายๆ เครื่องมาเชื่อมต่อกันด้วยระบบเครือข่าย ด้วยจุดประสงค์ ที่ต้องการใช้ข้อมูลร่วมกัน ดังนั้นระบบฐานข้อมูลที่นำมาใช้ในการประมวลผลแบบกระจายจึงต้องแบ่งออกเป็นส่วน เซิร์ฟเวอร์ (โปรแกรม Backend) และส่วน ไคลเอนต์ (โปรแกรม Front-end) ไว้ในคอมพิวเตอร์ที่ต่างเครื่องกัน เพื่อให้คอมพิวเตอร์แต่ละเครื่องที่ทำหน้าที่เป็น ไคลเอนต์ สามารถเรียกใช้ข้อมูลจากฐานข้อมูลที่เก็บอยู่ในเครื่องคอมพิวเตอร์ที่ทำหน้าที่เป็น เซิร์ฟเวอร์ ได้ร่วมกัน ดังรูป

ภาพที่ 3.31 แสดงรูปแบบการให้บริการฐานข้อมูล

### 3.13.5 การนำเอาระบบไคลเอนต์ / เซิร์ฟเวอร์ มาใช้ในการประมวลผลแบบกระจายนี้จะส่งผลให้

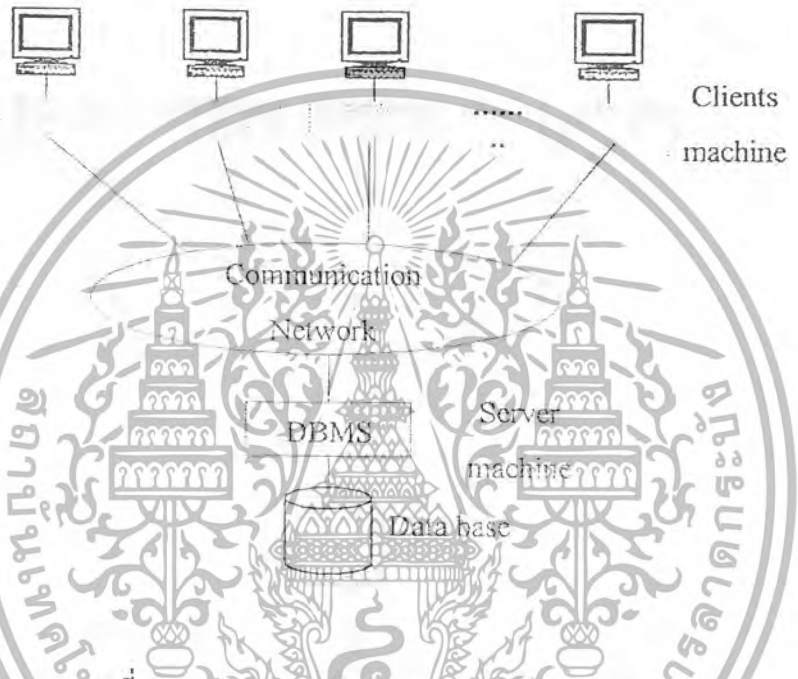
1. การประมวลผลของ ไคลเอนต์ / เซิร์ฟเวอร์ จะอยู่ในรูปแบบขนาน ดังนั้นเวลาในการตอบสนอง (Response Time) ของระบบที่มีต่อการทำงานจะใช้เวลาน้อยลง รวมทั้งปริมาณผลลัพธ์ที่ได้จากการประมวล (Throughput) จะได้ปริมาณมากขึ้นด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. ประสิทธิภาพการทำงานของ DBMS จะดีขึ้น โดยเฉพาะอย่างยิ่ง ในกรณีที่คอมพิวเตอร์ที่เป็นเซิร์ฟเวอร์ มีความสามารถที่สนับสนุนฟังก์ชันการทำงานต่างๆ ของ DBMS

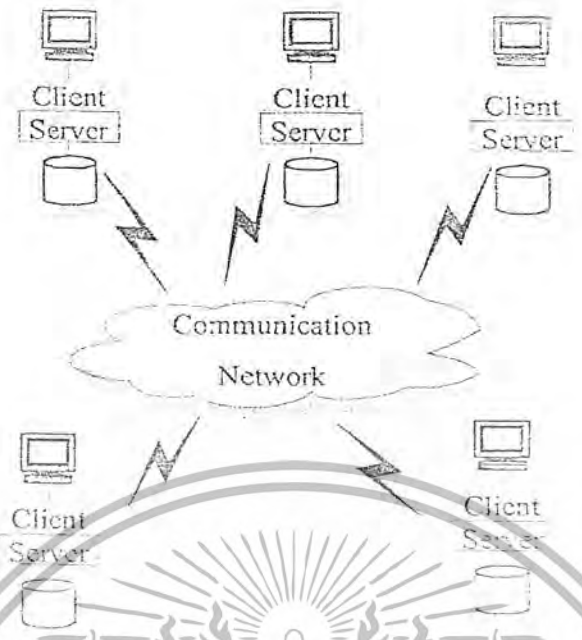
3. การติดต่อกับผู้ใช้ (User Interface) จะมีความรวดเร็ว และมีประสิทธิภาพมากขึ้นเนื่องจากหน้าที่หลักของคอมพิวเตอร์ที่ทำหน้าที่เป็น ไคลเอนต์ ได้แก่ การแสดงผล

4. คอมพิวเตอร์ที่ทำหน้าที่เป็น ไคลเอนต์สามารถใช้ข้อมูลร่วมกันได้ซึ่งสามารถแสดงด้วยแผนภาพดังรูปที่




ภาพที่ 3.32 แสดงการให้บริการฐานข้อมูลแบบกระจาย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้




**ภาพที่ 3.33** แสดงการให้บริการฐานข้อมูลขนาดใหญ่

ในกรณีที่องค์กรที่มีขนาดใหญ่ ที่มีการจัดเก็บข้อมูลไว้หลายๆ ที่ เช่น ธนาคาร ที่มีการแยกจัดเก็บข้อมูลตามสาขาต่างๆ อาจมีการกำหนดให้คอมพิวเตอร์ที่ทำหน้าที่เป็น ไคลเอนต์ มีการจัดเก็บข้อมูลของตนเองไว้และในขณะเดียวกัน คอมพิวเตอร์ที่ทำหน้าที่เป็น เซิร์ฟเวอร์ จะคอยส่งข้อมูลให้กับคอมพิวเตอร์เครื่องอื่น และในขณะเดียวกันก็สามารถทำหน้าที่เป็น ไคลเอนต์เพื่อเรียกใช้ข้อมูลจากฐานข้อมูลในคอมพิวเตอร์เครื่องอื่นได้เช่นเดียวกัน สำหรับรูปแบบของระบบในลักษณะนี้เรียกว่า Entire Database System ดังรูปที่ 2.18

ในกรณีที่มีการจัดเก็บฐานข้อมูลใน เซิร์ฟเวอร์ หลายๆ ตัว ไคลเอนต์ จะต้องมีความสามารถเพิ่มเติมในการเชื่อมต่อไปยัง เซิร์ฟเวอร์ แต่ละตัว เพื่อที่จะเรียกใช้ข้อมูลในฐานข้อมูลของแต่ละ เซิร์ฟเวอร์ ได้ แต่การเรียกใช้ข้อมูลในฐานข้อมูลของแต่ละเซิร์ฟเวอร์ นี้ จะเกิดขึ้นได้เพียงเซิร์ฟเวอร์เดียว ในแต่ละครั้งของการติดต่อระหว่าง ไคลเอนต์ / เซิร์ฟเวอร์

## 3.14 การสร้างฐานข้อมูลด้วยโปรแกรม Access

โปรแกรม Access เป็นโปรแกรมหนึ่งในกลุ่มของ Microsoft office ซึ่งจะช่วยงานด้าน "ฐานข้อมูล" (Database) นั่นคือการนำมาเก็บรายละเอียดของข้อมูลต่างๆ ให้เป็นหมวดหมู่จากนั้นจึงนำข้อมูลเหล่านั้นมาสร้าง รายงาน (Record) , แบบฟอร์ม (Form) , ค้นหาข้อมูล (Query) ในฐานข้อมูลตามเงื่อนไขและอื่นๆ อีกมากมาย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.14.1 ความหมายของ Database

Database หรือ ฐานข้อมูล หมายถึง ข้อมูลที่ถูกรวบรวมไว้อย่างมีระเบียบแบบแผน โดยข้อมูลเหล่านี้จะต้องมีความสัมพันธ์กันด้วย ตัวอย่างของ Database เช่น

- ข้อมูลในสมุดโทรศัพท์ขององค์การโทรศัพท์ ซึ่งจะมีชื่อ , ที่อยู่ และหมายเลขโทรศัพท์
- ข้อมูลของนักศึกษาในโรงเรียนหรือมหาลัย โดยจะมีชื่อ , นามสกุล , ที่อยู่ ฯลฯรวบรวมไว้

### 3.14.2 การสร้างฐานข้อมูล

เมื่อเราต้องการสร้างฐานข้อมูลในเครื่องคอมพิวเตอร์เราจะสามารถสร้างได้จาก 2 วิธี ดังต่อไปนี้

#### 3.14.2.1 สร้างจากโปรแกรมสำเร็จรูป

ในกรณีที่เราต้องการความสะดวกรวดเร็ว ในการสร้างฐานข้อมูลแล้ว ก็สามารถสร้างจากโปรแกรมสำเร็จรูปที่ถูกออกแบบมาเพื่อทำงานด้านฐานข้อมูลโดยเฉพาะ ซึ่งใช้โปรแกรมสำเร็จรูปนี้ จะมีมาตั้งแต่ยุคแรกๆ เลยทีเดียว ที่เรารู้จักกันดี ได้แก่ Dbase, FoxPro ซึ่งจะสามารถบันทึกแต่เพียงตัวหนังสือเท่านั้น จนในปัจจุบันนี้ จะมีโปรแกรมที่มาจัดการกับฐานข้อมูลที่เรารู้จักกันดีได้แก่ Microsoft Access ของบริษัทไมโครซอฟต์ ที่มีความสามารถบันทึกข้อมูลแปลกๆ ที่ได้กล่าวไปแล้วนั้น นอกจากนี้แล้วยังมีอีกหลาย ๆ โปรแกรมที่สร้างมา เพื่อฐานข้อมูล โดยเฉพาะคือ Oracle หรือ Power Builder

การสร้างฐานข้อมูล โดยโปรแกรมสำเร็จรูปนี้ จะทำให้เราสร้างฐานข้อมูลได้รวดเร็วแต่จะมีข้อมูลจำกัดอยู่มาก เมื่อต้องการสร้างโปรแกรม เพื่อทำงานกับฐานข้อมูล และเพื่อป้องกันปัญหาเหล่านี้ เราจะใช้โปรแกรมสำเร็จรูปสร้างฐานข้อมูลแต่เพียงอย่างเดียว และใช้วิธีเขียนโปรแกรมอีกภาษาหนึ่งเพื่อเปิดดูฐานข้อมูลนี้อีกทีหนึ่ง

นอกจากนี้แล้ว การใช้โปรแกรมสำเร็จรูปเพื่อสร้างฐานข้อมูลจะช่วยให้การพัฒนาโปรแกรมทำงานได้เร็วขึ้น ให้มีการสร้างฐานข้อมูลด้วยโปรแกรมสำเร็จรูปไปก่อน โดยไม่ต้องรอจนกว่าโปรแกรมที่พัฒนาสำเร็จ แล้วค่อยเปลี่ยนจากฐานข้อมูลที่สร้างแล้วนี้ ให้รับกับโปรแกรมที่พัฒนานี้ โปรแกรมสำเร็จรูปบางโปรแกรมก็สามารถสร้างฐานข้อมูลได้เช่นเดียวกัน เช่น ในโปรแกรมทางด้าน Spread Sheet เช่น Microsoft Excel หรือ Lotus 123 เป็นต้น เนื่องจากเป็นโปรแกรม ที่มีพื้นที่ที่ทำงานเป็นตาราง ที่เราสามารถบันทึกข้อมูลนี้ลงในตารางได้

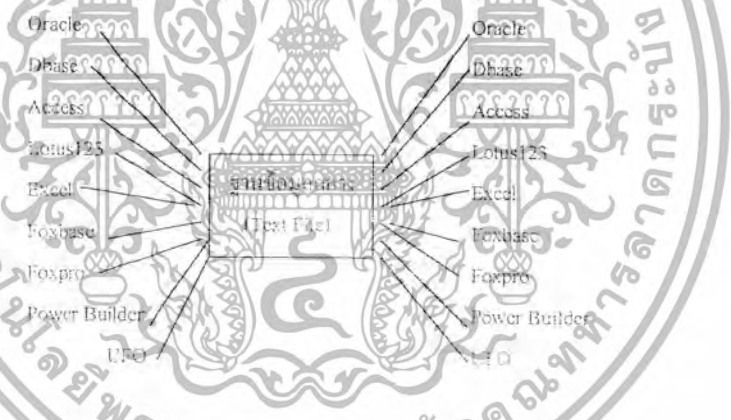
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.14.2.2 การสร้างฐานข้อมูลเอง

ถ้าหากเราต้องการสร้างโปรแกรม และควบคุมการทำงานของฐานข้อมูลได้ง่ายก็ควรที่จะสร้างฐานข้อมูลของเราเอง ซึ่งตามปกติฐานข้อมูลที่สามารถสร้างขึ้นได้เองนี้จะมีอยู่ 3 ชนิดด้วยกันคือ Sequential, Random และ Binary ซึ่งแต่ละชนิด จะมีคุณสมบัติที่แตกต่างกันออกไปแล้วแต่เราจะเลือก และเรากล่าวอีกทีหนึ่งในหนังสือเล่มนี้ สำหรับภาษาที่เราจะสร้างฐานข้อมูลของเราเองนี้ จะเป็นภาษา Visual Basic ของบริษัทไมโครซอฟต์

### 3.14.3 การย้ายฐานข้อมูล

ในการย้ายฐานข้อมูลไปมานี้ บางครั้งจะให้มีการย้ายข้อมูลกลางอันหนึ่ง ซึ่งสามารถติดต่อกับฐานข้อมูลทุกชนิดได้ ฐานข้อมูลกลางนี้ก็คือ ฐานข้อมูลที่เป็นเท็กซ์ไฟล์ธรรมดาๆ นี่เองซึ่งฐานข้อมูลเท็กซ์ไฟล์นี้จะเป็นที่ยอมรับของฐานข้อมูลเกือบทุกๆ ชนิดของข้อมูลเลย ดังนั้นถ้าเราต้องการย้ายฐานข้อมูลจากโปรแกรมที่พิสดาร มีคนใช้น้อยๆ แล้วก็ควบแปรงฐานข้อมูลเป็นฐานข้อมูลที่เท็กซ์ไฟล์ก่อน จากนั้นจึงค่อยๆ ย้ายเข้าไปในโปรแกรมที่เราต้องการ

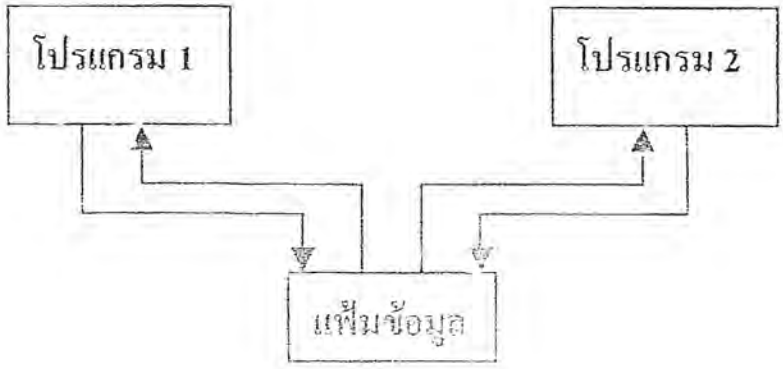



**ภาพที่3.34** การแปลงฐานข้อมูลต่างๆ เป็นเท็กซ์ไฟล์

### 3.14.4 การทำงานของโปรแกรมร่วมกับฐานข้อมูล

ฐานข้อมูลนั้นจะอยู่คนละส่วนกับโปรแกรมดังนั้นการพัฒนา สองส่วนนี้จึงแยกกันพัฒนาได้ฐานข้อมูลจึงทำงานได้อิสระไม่ยึดติดกับโปรแกรมหนึ่งโปรแกรมใด คือ ฐานข้อมูลหนึ่ง ๆ อาจจะทำงานร่วมกับโปรแกรมหลายโปรแกรมก็ได้ ดังแสดงในรูป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้




ภาพที่ 3.35 แสดงการร่วมกันใช้แฟ้มข้อมูลของโปรแกรม

**3.14.5 ศัพท์ต่างๆ ที่ควรทราบเกี่ยวกับ Database สำหรับโปรแกรม Access**

**Field (ฟิลด์)**

หมายถึง รายละเอียดย่อยๆ เช่น ชื่อคน,นับเป็น 1 Field, หมายเลยโทรศัพท์ก็นับเป็น 1 Field เช่นกัน (ใน Relational Database จะใช้คำว่า Attribute แทน ซึ่งมีความหมายที่ตรงมากกว่า)

**Record (เรคคอร์ด)**

ชื่อคนนับเป็น 1 Field, ที่อยู่นับเป็น 1 Field และหมายเลขโทรศัพท์ก็นับเป็น 1 Field ถ้า นำ 3 Field นี้มารวมกันจะนับเป็น 1 Record ดังนั้นรายละเอียดของคน 1 คนในสมุดโทรศัพท์ ก็คือ 1 Record นั่นเอง (ใน Relational Database จะใช้คำว่า Tuple แทนคำว่า Record)

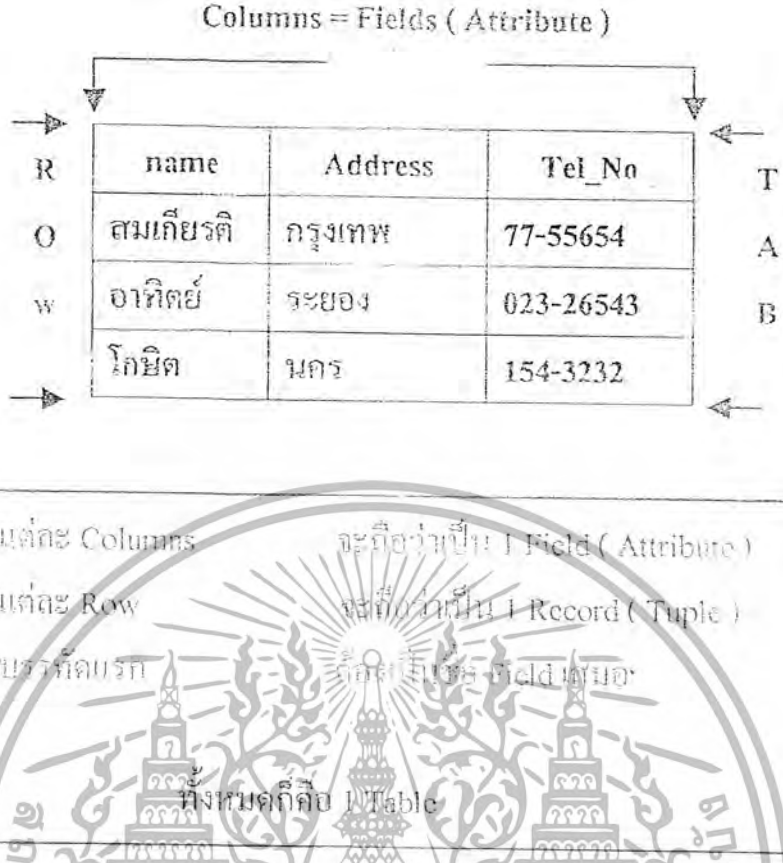
**Table (ตาราง)**

หมายถึงรายละเอียดของทุกคนที่อยู่ในสมุดโทรศัพท์ 1 เล่มคือ 1 Table

**Relational Database**

เป็น Database ชนิดหนึ่ง ซึ่ง Access นำมาใช้เป็นพื้นฐานในการจัดการฐานข้อมูล ของโปรแกรม โดยใน Database หนึ่งนั้นอาจประกอบไปด้วย Table หลายๆ Table และในแต่ละ Table ก็จะมีความสัมพันธ์ (Relation) ซึ่งกันและกันอยู่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้




**ภาพที่ 3.36** แสดงตัวอย่างการสร้างตาราง

## 3.15 รูปแบบการติดต่อกับฐานข้อมูลด้วย Visual Basic

ในการติดต่อกับฐานข้อมูลโดยปกติแล้ว VB จะเชื่อมโยงผ่านทาง Database Engine ที่เรียกว่า JET Engine จึงอาจกล่าวได้อีกนัยหนึ่งว่า JET Engine เป็นไดรเวอร์ชนิดหนึ่ง ซึ่งทำหน้าที่เป็นตัวเชื่อมโยงให้ VB สามารถติดต่อกับฐานข้อมูลได้นั่นเอง โดยที่ฐานข้อมูลหลัก (Default) ที่ VB รู้จักเป็นอย่างดีคือ MS Access แต่ Visual Basic สามารถติดต่อกับฐานข้อมูลได้ทุกชนิด เช่นกัน โดยอาศัยเทคโนโลยีหลายๆ อย่าง

สำหรับฐานข้อมูลที่นำเสนอ จะใช้ฐานข้อมูลของ MS Access 2000 Thai Edition เป็นหลักการติดต่อฐานข้อมูลใน Visual Basic จะแยกออกเป็น 4 ประเภทใหญ่ คือ

### 3.15.1 แบบที่ 1 ติดต่อโดยอาศัยคอนโทรลด้านฐานข้อมูล

มีศัพท์เรียกคอนโทรลกลุ่มนี้โดยเฉพาะว่า Bound Control ส่วนใหญ่แล้วก็คือ กลุ่มของคอนโทรลมาตรฐานที่ใช้กันโดยทั่วไปนั่นเอง เช่น คอนโทรล Textbox, Picture, Image, List Box,

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

combo Box เป็นต้นโดยใช้คอนโทรล Data (Data Control) เป็นตัวเชื่อมโยงระหว่างฐานข้อมูล กับกลุ่ม Bound Controls

คุณสามารถตรวจสอบได้ว่า คอนโทรลตัวใดบ้างถูกจัดอยู่ในกลุ่มของ Bound Controls โดยการตรวจสอบว่าคอนโทรลตัวนั้นมีคุณสมบัติที่ขึ้นต้นด้วยคำว่า Data...เช่น Data field, Data Format, Data Source หรือไม่ ถ้ามีหมายถึง คอนโทรลตัวดังกล่าวถูกจัดอยู่ในกลุ่มของ Bound Controls ด้วยเช่นกัน

ยังมีคอนโทรลอีกกลุ่มหนึ่งเรียกว่า ActiveX Bound Controls หมายถึง กลุ่มของคอนโทรลที่มีคุณสมบัติที่ขึ้นต้นด้วยคำว่า Data... เช่นกัน แต่มีข้อแตกต่างจากคอนโทรลในกลุ่ม Bound Controls ก็คือ ActiveX Bound Controls อาจมาจากผู้ผลิตรายอื่นๆ (Thirds Party) ซึ่งเป็นคอนโทรลที่ไม่ได้อยู่ในแถบเครื่องมือมาตรฐานของ Visual Basic โดยที่คุณต้องเพิ่มเติมคอนโทรลกลุ่มนี้เข้ามาในแถบเครื่องมือเอง ดังนั้นจึงเรียกคอนโทรลกลุ่มนี้ว่า ActiveX Bound Controls เช่น คอนโทรล DBGrid, คอนโทรล Dblist หรือคอนโทรล DBCombo ฯลฯ เป็นต้น

### 3.15.2 แบบที่ 2 ติดต่อโดยใช้ออบเจ็กต์ Data Access Object (DAO)

ถือว่าเป็นวิธีที่ถ้าสมัยแล้ว โดยมีแนวคิดในการติดต่อหรือเข้าถึงข้อมูลในฐาน ข้อมูลผ่านทางองค์ประกอบต่างๆ ในฐานข้อมูล เช่น ฟิลด์ (Field), เร็กคอร์ด (Record), ความสัมพันธ์ระหว่างตาราง (Relation) เป็นต้น โดยจะแทนแต่ละองค์ประกอบเหล่านั้นด้วยออบเจ็คต์ (Object) และควบคุมออบเจ็คต์ต่างๆ เหล่านี้โดยการเขียนโค้ด

แม้จะทำงานได้ดีกว่า อิสระกว่า แต่มีความยุ่งยากในการเขียนโปรแกรมด้วยเช่นกัน อีกทั้งเป็นเทคโนโลยีที่เก่าแล้วคือ เน้นเฉพาะระบบฐานข้อมูลที่เป็นตาราง (โดยเฉพาะ Access รุ่นเก่าๆ) แต่ว่าการเก็บข้อมูลในปัจจุบัน ถูกจัดเก็บในสภาพแวดล้อมแตกต่างกันมากมาย เช่น รูปภาพ (Imager), ข้อความ (Text) และรูปแบบอื่นๆ อีกมากมายทำให้ต้องสร้างออบเจ็คต์ใหม่ๆ ขึ้นมาเรื่อยๆ แต่นั่นไม่ใช่ที่ทำได้ง่าย และกลายเป็นข้อความที่สำคัญของ DAO ตัวอย่างออบเจ็คต์ในกลุ่มนี้ เช่น ออบเจ็คต์ Record Set, ออบเจ็คต์ Table Def, คอนเล็กชัน Fields เป็นต้น

### 3.15.3 แบบที่ 3 ติดต่อผ่าน ODBC โดยตรง (ODBC Direct)

เป็นการติดต่อกับฐานข้อมูลแบบ 32 บิตที่สนับสนุนมาตรฐาน ODBC (Open Data Base Connectivity) ที่ JET Engine (กลไกการติดต่อฐานข้อมูลของไมโครซอฟต์ ซึ่งเน้นที่ Access) ไม่สามารถจัดเก็บได้ เช่น ฐานข้อมูลของ Oracle, ฐานข้อมูลของ Microsoft SQL Server, DB2 ฯลฯ) ซึ่งเป็นไปตามสถาปัตยกรรม n-Tier (n เทียร์) เช่น ไคลเอนต์ / เซิร์ฟเวอร์ (2-Tier ทูเทียร์) หรือ Middle Tier (3-Tier ทรีเทียร์) โดยอาศัยมาตรฐาน ODBC ในการเชื่อมโยง ที่เกี่ยวข้องกับ Visual Basic คือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- คอนโทรล Remote Data (Remote Data Control – RDC)
- ออบเจ็คต์ Remote Data (Remote Data Object – RDO)

### 3.15.4 แบบที่ 4 เข้าถึงข้อมูลโดยอาศัยเทคโนโลยี OLEDB

เป็นรูปแบบการติดต่อกับฐานข้อมูลผ่านทางกลุ่มออบเจ็คต์ในโมเดล ADO ซึ่งใช้ OLEDB Provider เป็นกลไกในการเข้าถึงข้อมูลในฐานข้อมูลแทน JET Engine โดยเปลี่ยนจากมุมมองการติดต่อฐานข้อมูลแบบเดิมๆ ที่ต้องกำหนดชนิดของฐานข้อมูลมาเป็นมองที่รูปแบบของการเชื่อมต่อ (Connection) เข้ากับฐานข้อมูล

OLEDB เป็นเทคโนโลยีล่าสุดที่ไมโครซอฟต์วางตำแหน่งให้เป็นยุคต่อไปในการเข้าถึงข้อมูลในฐานข้อมูลที่มีรูปแบบการเก็บข้อมูลสารพัดรูปแบบ ซึ่งจะไม่ได้จำกัดตัวเองเฉพาะตารางโดย OLEDB เป็นส่วนหนึ่งของสถาปัตยกรรม Universal Data Access (UDA) ของไมโครซอฟต์

ใน Visual Basic สามารถใช้ OLEDB ได้ 2 แบบ โดยการใช้

1. คอนโทรล ADO Data (ADO Data Control) ร่วมกับกลุ่มของ ActiveX Bound Control) ซึ่งสังเกตจากชื่อของคอนโทรล Data Glide, Data list, Hierarchical Flex Grid เป็นต้น

2. ชุดออบเจ็กต์ ActiveX Data (ActiveX Data Object – ADO) โดยการเขียนโปรแกรมควบคุม

## 3.16 Visual Basic 6

Visual Basic ถือได้ว่าเป็นเครื่องมือที่ใช้เขียน Windows ที่ได้รับความนิยมสูงสุด ทั้งนี้เพราะผ่านการพัฒนามาอย่างต่อเนื่องจากเวอร์ชั่นแรกที่ทำงานบนดอส แล้วมาดังสุดขีดที่เวอร์ชั่น 3.0 ที่ทำงานบน Windows 3.1 จนก้าวมาถึงเวอร์ชั่นล่าสุดคือ เวอร์ชั่น 6.0 ที่ได้รับความนิยมทั้งเมืองไทยและทั่วโลก

แต่ก่อนนั้นเราจะเรียกการสร้างแอพพลิเคชั่น (application) ว่า การเขียนโปรแกรม โดยในอดีตนั้นผู้เขียนโปรแกรม หรือโปรแกรมเมอร์ต่างก็ผ่านความยุ่งยากซับซ้อนของการเขียนโปรแกรม ทำให้มองกันว่าการเขียนโปรแกรมเป็นเรื่องซับซ้อน

บริษัทไมโครซอฟท์ได้เล็งเห็นว่า ถ้าขืนทำให้การสร้างแอพพลิเคชั่นเป็นเรื่องยากแล้ว ก็เปรียบเสมือนฆ่าตัว Windows ทางอ้อมนั้นเอง ทำให้มีการสร้างการเขียนโปรแกรมแบบ Visual programming ขึ้นมา ซึ่งรูปแบบนี้ก็คือ การเขียนโปรแกรมไปพร้อมๆกับ การเห็นผลลัพธ์ที่เกิดขึ้นมีเครื่องไม้เครื่องมือคอยอำนวยความสะดวกในการเขียนโปรแกรมมากขึ้น ลดความซับซ้อนยุ่งยากต่างๆ ลงเหลือเฉพาะสิ่งที่ต้องเขียนโปรแกรมลงจริงๆ เท่านั้น

Visual Basic ก็คือหนึ่งใน Visual Programming ที่ไมโครซอฟท์สร้างขึ้นมา และด้วยความเรียบง่ายของภาษาและการเขียนโปรแกรมที่รวดเร็วทำให้ได้รับความนิยมสูงสุดในปัจจุบัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.16.1 จุดเด่นของ Visual Basic

#### 3.16.1.1 สร้างแอพพลิเคชั่นได้ง่ายและรวดเร็ว

Visual Basic ได้รับการวางตัวให้เป็นเครื่องมือที่ช่วยให้สร้างแอพพลิเคชั่นได้อย่างรวดเร็วและง่ายดาย เพื่อลดในการสร้างแอพพลิเคชั่นให้สั้นลง ซึ่งเรียกรูปแบบนี้ว่า Rapid Application Development หรือ RAC ทั้งนี้เพราะมีการขจัดงานที่โปรแกรมเมอร์ต้องทำซ้ำ ๆ ซาก ๆ ออกไป เหลือเฉพาะที่ต้องโฟกัสเกี่ยวกับปัญหาของงานจริง ๆ แล้วเขียนโปรแกรมจัดการปัญหานั้น ๆ ส่วนเรื่องอื่น ๆ เหลือให้ Visual Basic จัดการ

#### 3.16.1.2 การเขียนโปรแกรมที่ง่ายต่อการเรียนรู้

ถ้าได้มีโอกาสเขียนโปรแกรมด้วย Visual Basic แล้วจะเห็นว่า ภาษา Basic ใน Visual Basic นั้นอ่านง่ายคือ อ่านแล้วใกล้เคียงกับภาษาที่เราใช้งานปกติ อ่านแล้วสื่อความหมายเข้าใจได้ง่ายกว่าภาษาโปรแกรมอื่น ๆ ทำให้ผู้ที่เพิ่มเริ่มต้นเขียนโปรแกรมทำความเข้าใจกับการเขียนโปรแกรมได้อย่างรวดเร็ว

#### 3.16.1.3 รวมเครื่องมืออำนวยความสะดวกในการเขียนโปรแกรม

นอกจากจะง่ายต่อการเรียนรู้แล้ว Visual Basic ก็ยังมีเครื่องมือที่ช่วยให้การเขียนโปรแกรมเป็นเรื่องที่ไม่ยุ่งยากเพราะจะมีเครื่องมือที่ช่วยให้ไม่ต้องจดจำไวยากรณ์ภาษาที่ยุ่งยากในการตรวจสอบอัตโนมัติว่าโปรแกรมที่เขียนนั้นถูกต้องตามหลักภาษาหรือไม่, มีการแยกแยะส่วนของโปรแกรมอย่างเป็นระเบียบ ทำให้งานของโปรแกรมเมอร์นั้นลดลงได้มาก

นอกจากจะมีเครื่องมือช่วยในการเขียนโปรแกรมแล้ว ยังมีเครื่องมือที่ใช้ทดสอบแก้ไขโปรแกรม (Debugger) ที่เขียนขึ้นมาว่าทำงานได้ถูกต้องหรือไม่ มีระบบขอความช่วยเหลือ (Online Help) ไว้อ้างอิง และขอความช่วยเหลือในจุดที่เราสงสัยข้องใจได้

เครื่องมือทั้งหมดที่กล่าวมาถูจัดรวมไว้ในสภาพแวดล้อมการทำงานเดียวกัน (เรียกย่อๆว่า IDE ซึ่งย่อมาจาก Integratd Development Environment) ทำให้เรียกใช้งานได้สะดวกตั้งแต่เขียนโปรแกรม, ทดสอบ, แก้ไข, สร้างชุดติดตั้ง รวมทั้งระบบขอความช่วยเหลือ ซึ่งเราสามารถเพิ่มเติมเครื่องมือชนิดใหม่ๆ เข้าไปได้เรื่อยๆ หรือถอดเครื่องมือที่ไม่จำเป็นต้องใช้เพื่อประหยัดพื้นที่ฮาร์ดดิสก์ก็ได้เช่นกัน

### 3.16.2 รูปแบบการพัฒนาแอพพลิเคชั่นกับ Visual Basic

เมื่อเรามองเห็นว่า Visual Basic จะช่วยให้เราสร้างแอพพลิเคชั่นบน Windows ได้ง่าย และรวดเร็วแล้วยังมีรูปแบบที่ Visual Basic สามารถสร้างขึ้นมาได้อีก คือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**3.16.2.1 พัฒนาแอพพลิเคชั่นกับ ActiveX Control**

เทคโนโลยีที่มีชื่อว่า ActiveX เป็นตัวอยู่เบื้องหลังความสำเร็จของ Visual Basic ซึ่งช่วยลดงานที่ซ้ำซ้อนของการเขียนโปรแกรมลงไปมาก

ตัวอย่างเช่น การเขียนโปรแกรมเพื่อรับค่าข้อมูลจากผู้ใช้ แต่เดิมเราต้องมาเขียนโปรแกรมเพื่อวาดหน้าจอ, เขียนโปรแกรมวาดรูปปุ่ม และช่องรับข้อความ รวมทั้งเขียนโปรแกรมเพื่อจัดการกับข้อมูลที่ผู้ใช้ป้อมเข้ามา แต่ ActiveX จะทำให้เราสนใจเฉพาะการจัดการกับข้อมูลที่ผู้ใช้ป้อนเข้ามาเท่านั้น ที่เหลือจัดการให้เอง โดยช่องรับข้อความปุ่มต่างๆ จะใช้ ActiveX Control จัดการ

นอกจากจะลดความซับซ้อนลงแล้วการที่เราใช้ ActiveX Control ในการเขียนโปรแกรมยังช่วยให้โปรกแกรมที่เราเขียนกับโปรแกรมที่คนอื่นๆเขียนนั้น ตั้งอยู่บนมาตรฐานเดียวกัน ทำให้การบำรุงรักษาง่าย เพราะใครๆ ก็เข้าใจมาตรฐานของ ActiveX Control นี้ ทำให้ไม่ต้องกังวลใจว่าโปรแกรมที่เขียนขึ้นจะมีเฉพาะคนเขียนเท่านั้นที่เข้าใจ

**3.16.2.2 สร้างแอพพลิเคชั่นที่ใช้งานกับฐานข้อมูล**

เป็นแอพพลิเคชั่นที่มีการใช้งานมากที่สุดเพราะระบบร้านค้า, คลังสินค้า, ระบบบัญชี, ระบบบริหาร งานบุคคลหรือแม้แต่ e-Commerce ทั้งหมดต่างๆ ต้องมีส่วนที่ติดต่อกับฐานข้อมูลที่แน่นอน

Visual Basic ได้ช่วยในการสร้างแอพพลิเคชั่นกับฐานข้อมูลเป็นเรื่องที่ทำได้ง่ายดาย เพราะมีเครื่องมือต่างๆ ที่อำนวยความสะดวกในการเขียนโปรแกรมเพื่อใช้งานข้อมูลจากฐานข้อมูล ซึ่งไม่จำกัดด้วยว่าจะเป็นฐานข้อมูลแบบใดทั้งฐานข้อมูลส่วนบุคคล, ฐานข้อมูลผ่านเครือข่าย หรือฐานข้อมูลผ่านอินเตอร์เน็ต จากความสามารถที่หลากหลายนี้ Visual Basic จึงเป็นตัวเลือกอันดับต้นๆ ของการสร้างแอพพลิเคชั่นที่เกี่ยวกับฐานข้อมูลในโลกธุรกิจ

**3.16.2.3 สร้างแอพพลิเคชั่นแบบใหม่กับอินเตอร์เน็ต**

อินเตอร์เน็ตนับว่ามีความสำคัญกับชีวิตของคนที่ใช้ไอทีมากขึ้นทุกวัน ซึ่ง Visual Basic เปิดโอกาสให้เราสามารถสร้างแอพพลิเคชั่นเพื่อรองรับการทำงานร่วมกับอินเตอร์เน็ตได้ด้วยการใช้ความรู้เดิมที่เรามีอยู่แล้วจากการสร้างแอพพลิเคชั่นปกติ รวมทั้งมีเครื่องมือเสริมการทำงานต่างๆ อย่างมากมาย

จุดเด่นของ Visual Basic อีกข้อหนึ่งคือ เปิดโอกาสให้เรานำแอพพลิเคชั่นปกติที่เดิมทำงานกับคอมพิวเตอร์ส่วนบุคคล สามารถดัดแปลงใช้กับงานอินเตอร์เน็ตได้อย่างไม่ยากเย็นนักทำให้ไม่ต้องโละงานเดิมๆทิ้ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**3.17.1 เริ่มสร้างแอพพลิเคชั่นด้วย Visual Basic**

ขั้นแรก : ออกแบบแอพพลิเคชั่น

ต้องทราบให้แน่ชัดก่อนว่าแอพพลิเคชั่นที่เราจะสร้างนั้นใช้ประโยชน์อะไร ต้องมีความสามารถอะไรบ้าง ต้องมีรูปร่างหน้าตาเป็นอย่างไร ซึ่งจำเป็นอย่างยิ่งที่ต้องคิดให้รอบคอบและเขียนออกมาให้ชัดเจน โดยอาจจะร่างง่ายๆ ในกระดาษ

ขั้นที่ 2: ตกแต่งหน้าตาแอพพลิเคชั่น

เป็นการตกแต่งรูปร่างแอพพลิเคชั่นตามที่ได้ออกแบบไว้ พร้อมๆกับการกำหนดค่า พร็อพเพอร์ตี้ต่างๆ ให้กับคอนโทรลแต่ละตัวในแอพพลิเคชั่น

ขั้นที่ 3: เขียนโค้ดกำกับแอพพลิเคชั่น

หลังจากตกแต่งหน้าตาเสร็จแล้ว ขั้นต่อไปคือ การเขียนโค้ดหรือการเขียนโปรแกรมเพื่อควบคุมการทำงานต่างๆ ซึ่งเราจะใช้การเขียนโปรแกรมแบบ Event Driven Programming ซึ่งจะเป็นการเขียนโค้ดเพื่อรอบรับกับเหตุการณ์ต่างๆ ที่จะเกิดขึ้นกับคอนโทรลต่างๆ ในแอพพลิเคชั่นของเรา

ขั้นที่ 4: ทดสอบการทำงานของแอพพลิเคชั่น

เมื่อเขียนโค้ดเสร็จแล้ว ก็ถึงเวลาที่จะทดสอบการทำงานของแอพพลิเคชั่นที่เราสร้างขึ้น ซึ่งประกอบไปด้วยคอนโทรลต่างๆ ที่ปรับแต่งไว้ แต่โค้ดที่เขียนเพื่อจัดการกับเหตุการณ์ต่างๆ

ขั้นที่ 5: บันทึกเก็บไว้ในคอมพิวเตอร์

หลังจากทดสอบจนแน่ใจว่าแอพพลิเคชั่นที่สร้างนั้นทำงานได้ถูกต้อง เราจึงบันทึกเก็บไว้ซึ่งสามารถแก้ไขและเพิ่มเติมความสามารถอื่นๆ ได้ในภายหลัง

ขั้นที่ 6: สร้างไฟล์ .EXE(Make)

เมื่อเราสร้างแอพพลิเคชั่นเสร็จแล้ว เราอาจจะต้องการนำแอพพลิเคชั่นที่สร้างขึ้นเรียกใช้งานได้เอง โดยไม่ต้องเรียกผ่าน Visual Basic หรือต้องการนำไปใช้งานในคอมพิวเตอร์เครื่องอื่นๆ ซึ่งเราจะทำให้ได้โดยการสร้างไฟล์เอ็กซ์คิวต์ (ไฟล์ที่มีนามสกุลเป็น.EXE)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 3.17 ฐานข้อมูล Data Base

ฐานข้อมูล (Data Base) คือ วิธีการจัดเก็บข้อมูลที่สัมพันธ์กันอย่างมีระเบียบ ซึ่งจะทำให้ง่ายต่อการงาน และค้นหาข้อมูล ซึ่งฐานข้อมูลทีคนส่วนใหญ่คุ้นเคยคือ ฐานข้อมูลเชิงสัมพันธ์ เป็นรูปแบบการจัดเก็บฐานข้อมูลที่สัมพันธ์กัน โดยมองข้อมูลในลักษณะของตารางต่างๆ ทีมีความสัมพันธ์กัน

### 3.17.1 องค์ประกอบของฐานข้อมูล

การใช้งานฐานข้อมูลจะต้องประกอบด้วยองค์ประกอบต่อไปนี้

#### 3.17.1.1 แอพพลิเคชั่นฐานข้อมูล (Database Application)

แอพพลิเคชั่นฐานข้อมูลเป็นแอพพลิเคชั่นที่สร้างไว้ให้ผู้ใช้งานสามารถติดต่อกับฐานข้อมูลได้ อย่างสะดวก ซึ่งมีรูปแบบการติต่อกับฐานข้อมูลแบบเมนู หรือกราฟิก โดยไม่จำเป็นต้องมีความรู้เลยก็สามารถเรียกใช้งานฐานข้อมูลได้

#### 3.17.1.2 ระบบจัดการฐานข้อมูล (DBMS)

ระบบจัดการฐานข้อมูล เป็นซอฟต์แวร์ที่ทำหน้าที่จัดการข้อมูลในฐานข้อมูล ทั้งการจัดเก็บ การแสดงผล การค้นหา การสำรองข้อมูล เป็นต้น โดยจะเป็นเครื่องมือในการทำงานของผู้บริหารฐานข้อมูล และเป็นตัวกลางที่เชื่อมผ่านระหว่างแอพพลิเคชั่นฐานข้อมูลที่สร้างขึ้นกับตัวข้อมูลในฐานข้อมูล เช่น Microsoft Access, FoxPro เป็นต้น

#### 3.17.1.3 ดาต้าเบสเซิร์ฟเวอร์ (Database Server)

ดาต้าเบสเซิร์ฟเวอร์ เป็นคอมพิวเตอร์ที่คอยให้บริการในการจัดการฐานข้อมูล ซึ่งก็คือคอมพิวเตอร์ที่ระบบจัดการฐานข้อมูลทำงานนั่นเอง เพราะฉะนั้นจึงมักจะเป็นคอมพิวเตอร์ที่มีประสิทธิภาพการทำงานสูงกว่าคอมพิวเตอร์ที่ใช้งานทั่วไป

#### 3.17.1.4 ข้อมูล (Data)

ข้อมูล คือ ตัวเนื้อของข้อมูลที่เราใช้งาน ซึ่งจะถูกเก็บในหน่วยความจำของดาต้าเบสเซิร์ฟเวอร์ โดยจะถูกเรียกมาใช้งานจากระบบจัดการฐานข้อมูล

#### 3.17.1.5 ผู้บริหารฐานข้อมูล (Database Administrator)

ผู้บริหารฐานข้อมูล เป็นคนที่ทำหน้าที่ดูแลข้อมูลในฐานข้อมูลผ่านระบบจัดการฐานข้อมูล ซึ่งจะควบคุมให้การทำงานเป็นไปอย่างราบรื่น นอกจากนี้ยังทำหน้าที่กำหนดผู้ที่จะมี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สิทธิ์ใช้งานฐานข้อมูล กำหนดในเรื่องความปลอดภัยของการใช้งาน พร้อมทั้งดูแลดาต้าเบสเซิร์ฟเวอร์ให้ทำงานอย่างเป็นปกติด้วย

ฐานข้อมูลมีหลักการพื้นฐานสำคัญมาจากคณิตศาสตร์ในเรื่องทฤษฎีบทของเซท (Theory of Set) โดยที่การจัดเก็บหรือแสดงข้อมูลให้ผู้ใช้เห็นจะเป็นแบบตาราง

ข้อมูลต่างๆจะแสดงในรูปของตาราง โดยในแต่ละตารางจะเป็นการจัดรวบรวมข้อมูลประเภทเดียวกันไว้ด้วยกัน โดยแต่ละแถวที่ประกอบขึ้นเป็นตารางนั้น ก็คือ เรคคอร์ด ซึ่งจะเป็นที่เก็บข้อมูลแต่ละชุดของตารางนั้น และในแต่ละแถวก็จะประกอบด้วยคอลัมน์ซึ่งเป็นหน่วยย่อยที่แสดงคุณสมบัติของข้อมูลแต่ละแถว

ในแต่ละตารางมักจะมีบางคอลัมน์หรืออาจจะหลายคอลัมน์ประกอบกันที่สามารถบอกถึงความแตกต่างของฐานข้อมูลในแต่ละแถวได้ คอลัมน์ดังกล่าวเรียกว่า Primary Key เช่น คอลัมน์เลขประจำตัวพนักงานที่แตกต่างกันทุกคน และในการเชื่อมโยงตารางฐานข้อมูลที่มีหลายตารางนั้นจะกระทำได้ก็ต่อเมื่อแต่ละตารางมีคอลัมน์เกี่ยวโยงกัน ซึ่งคอลัมน์ที่เกี่ยวโยงนี้จเรียกว่า Foreign key

### 3.17.2 ชุดคำสั่งที่ใช้จัดการกับข้อมูล

แบ่งออกเป็น 2 ประเภทหลักๆคือ

1. จัดการกับโครงสร้างฐานข้อมูล เป็นการสร้าง ลบ หรือแก้ไขฐานข้อมูลและตาราง เช่น สร้างตารางฐานข้อมูลพนักงานขึ้นมา ชุดคำสั่งต่างๆ ที่เกี่ยวข้องกับการสร้างหรือเปลี่ยนแปลงโครงสร้างของฐานข้อมูลมีศัพท์เรียกว่า Data Management Language หรือ DDL

2. จัดการกับฐานข้อมูลที่มีอยู่ในตารางต่างๆ ในฐานข้อมูล เป็นการทำงานในเรื่องของการเพิ่ม ลบ แก้ไขข้อมูลในตารางต่างๆ เช่น การเพิ่มข้อมูลพนักงานใหม่เข้าไปในตารางข้อมูลพนักงาน คำสั่งประเภทนี้เรียกว่า Data Management Language หรือ DML

### 3.17.3 โครงสร้างของภาษา SQL

Structure Query Language หรือ SQL เป็นภาษามาตรฐานที่ใช้ในการจัดการฐานข้อมูลโดยเฉพาะที่เป็นแบบแอพพลิเคชั่นที่ใช้งานกับฐานข้อมูล SQL เป็นภาษาที่มีประสิทธิภาพสูงสามารถเรียนรู้และใช้งานอย่างง่ายดาย จึงทำให้เป็นที่นิยมได้อย่างรวดเร็ว นอกจากนี้ยังสามารถใช้กันการจัดการฐานข้อมูลโดยผ่านโปรแกรมก็ได้ เช่น หากเราใช้ MS-Access เราก็สามารถพิมพ์คำสั่ง SQL ได้โดยตรงในโปรแกรม Access หรือ จะใช้คำสั่ง SQL ผ่านVisual Basic ก็ได้ เราจะต้องใช้ SQL เพื่อทำการจัดการข้อมูล ในฐานข้อมูลได้หลายอย่าง เช่น การแสดงข้อมูลจากฐานข้อมูลแบบมีเงื่อนไข การเพิ่ม การลบ และการนำข้อมูลจากตารางหลาย ๆ ตารางมาแสดงร่วมกันได้ เป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาษา SQL ประกอบไปด้วย 3 ส่วนใหญ่ ๆ คือ

1. Data Definition Languge (DDL) เป็นกลุ่มคำสั่งในภาษา SQL ที่ใช้สำหรับโครงสร้างของฐานข้อมูล เช่น การสร้างฐานข้อมูล ปรับปรุงโครงสร้างของฐานข้อมูล เป็นต้น
ตัวอย่างการใช้งานของกลุ่มคำสั่ง DDL ก็คือ การสร้างฐานข้อมูลด้วย MS-SQL Sever ก็จะมีการใช้งานคำสั่งในกลัง DDL เป็นหลัก

2. Data Mainpulation Language (DML) เป็นกลุ่มคำสั่งในภาษา SQL ที่ใช้สำหรับจัดการข้อมูลในฐานข้อมูล เช่น การแสดงข้อมูลแบบมีเงื่อนไข การลบข้อมูล การเพิ่มข้อมูล และการแสดงที่มาจากตารางหลายตาราง เป็นต้น

3. กลุ่มฟังก์ชั่น Aggregate Funtion เป็นฟังก์ชันพิเศษของภาษา SQL ที่ทำหน้าที่เฉพาะอย่าง เช่น หาผลรวมเร็คคอร์ด ค่าสูงสุด ค่าต่ำสุด เป็นต้น เป็นกลุ่มฟังก์ชันที่เป็นประโยชน์มาก เพราะจะช่วยลดภาระให้คุณไม่ต้องเขียนโค้ดจัดการเอง

สำหรับการใช้งานภาษา SQL ร่วมกับ Visual Basic เพื่อจัดการฐานข้อมูล จะใช้งานกลุ่มคำสั่ง DML ร่วมกับกลุ่มฟังก์ชัน Aggregate และกำหนดเงื่อนไขโดยใช้ตัวดำเนินการด้านต่างๆ

คำสั่งในกลุ่มของ DML จะมีคำสั่งพื้นฐานอยู่ 4 คำสั่งคือ

1. DELETE เป็นคำสั่งที่ใช้สำหรับลบข้อมูลหรือลบเร็คคอร์ดใดๆในตาราง
2. INSERT เป็นคำสั่งที่ใช้สำหรับเพิ่มข้อมูลหรือเพิ่มเร็คคอร์ดใดๆเข้าไปในตาราง
3. SELECT เป็นคำสั่งที่ใช้สำหรับเลือกข้อมูลหรือแสดงเร็คคอร์ดใดๆที่ต้องการจากตารางอาจจะมาจากตารางเดียว หรือหลายตารางก็ได้
4. UPDATE ใช้สำหรับแก้ไขข้อมูลหรือแก้ไขเร็คคอร์ดใดๆในตาราง

**3.17.3.1 ตัวดำเนินการ(Operator)**

ตัวดำเนินการเปรียบเทียบที่น่าสนใจ ได้แก่

| | |
|---|---|
| = | คือ เท่ากับ (Equal) |
| <> | คือ ไม่เท่ากับ (Not Equal) |
| < | คือ น้อยกว่า (Less Than) |
| > | คือ มากกว่า (Greater Than) |
| <= | คือ น้อยกว่าห รือเท่ากับ (Less Than or TO) |
| >= | คือ มากกว่าหรือเท่ากับ(Greater Than or Equal To) |
| Like | เข้าการเปรียบเทียบโดยใช้ตัวอักษรพิเศษ(Wild Card Character) เข้าร่วม |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**3.17.3.2 ตัวดำเนินการด้านตรรกะ(Logical Operater)**

ตัวดำเนินการด้านตรรกะที่นิยมใช้จะมีอยู่ 3 คือ And, Or, Not

**3.17.3.3 กลุ่มฟังก์ชัน Aggregate**

กลุ่มฟังก์ชัน Aggregate เป็นฟังก์ชันที่ใช้คำนวณทางคณิตศาสตร์ โดยช่วยให้การนำเสนอผลการค้นหาเป็นไปอย่างมีประสิทธิภาพ ซึ่งมีฟังก์ชันพื้นฐานดังต่อไปนี้

| | | |
|---|---|---|
| AGV( ) | หน้าที่ | หาค่าเฉลี่ยของฟิลด์จากเร็คคอร์ดทั้งหมด |
| COUNT( ) | หน้าที่ | นับจำนวนเร็คคอร์ด |
| FIRST | หน้าที่ | หาค่าแรกในฟิลด์ |
| LAST | หน้าที่ | หาค่าที่สุดท้ายในฟิลด์ |
| MAX | หน้าที่ | หาค่ามากที่สุด หรือค่าสูงสุด |
| MIN | หน้าที่ | หาค่าน้อยสุด หรือหาค่าต่ำสุด |
| SUM | หน้าที่ | หาผลรวมทั้งหมดของฟิลด์ |

**3.17.4 ลักษณะการใช้งานของกลุ่มคำสั่ง DML**

เป็นคำสั่งที่ใช้สำหรับ ลบหรือเร็คคอร์ดใดๆออกจากตาราง มีรูปแบบการใช้งาน 2 ลักษณะใหญ่ๆ คือ

| | | |
|---|---|---|
| รูปแบบที่ 1 | DELETE FORM | ชื่อตาราง WHERE เงื่อนไข |
| รูปแบบที่ 2 | DELETE FORM | ชื่อตาราง |

ชื่อตารางนี้ หมายถึงชื่อตารางที่ต้องการลบ ส่วนเงื่อนไขจะหมายถึง เงื่อนไขในการลบข้อมูล หรือลบเร็คคอร์ดในตารางนั้น

เครื่องหมาย * หมายถึง ข้อมูลใดๆ หรือข้อมูลทุกเร็คคอร์ด

**คำสั่ง INSERT**

เป็นคำสั่งที่ใช้สำหรับเพิ่มข้อมูล หรือเพิ่มเร็คคอร์ดเข้าไปในตารางในกรณีที่ฟิลด์เป็นข้อมูลชนิด Text ต้องใช้เครื่องหมาย “กำกับด้วยฟิลด์นั้นด้วย มีรูปแบบการใช้งาน 2 ลักษณะ

| |
|---|
| รูปแบบที่ 1 INSERT INTO tablename(field1.field2,…)VALUES(value1.value2,…) |
| รูปแบบที่ 2 INSERT INTO table name 1 |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

| | |
|---|---|
| ตัวแปรtablename | หมายถึง ชื่อตารางที่ต้องการเพิ่มเร็คคอร์ดเข้าไป |
| ตัวแปรTablename1 | หมายถึง เลือกข้อมูลจากตารางที่ชื่อ tablename1ตามเงื่อนไขที่ระบุไว้ในตัวแปร criteria แล้วนำมาเพิ่มที่ตาราง tablename2 |
| ตัวแปร Field | หมายถึง ชื่อของฟิลด์ต่างๆที่อยู่ในตารางที่ต้องการเพิ่มข้อมูล |
| ตัวแปร Value | หมายถึง ค่าของฟิลด์ที่จะเพิ่มเข้าไป โดยที่ต้องระบุค่าให้ตรงกับฟิลด์ |

- **คำสั่ง SELECT**

สำหรับเลือกหรือดึงข้อมูล(Retrieve Data) ที่เราต้องการจากตารางที่ระบุไว้คำสั่งที่มีความยืด
หยุ่นสูงมาก เพราะว่าเงื่อนไขในการนำข้อมูลออกมาจากตารางมีมากหลายลักษณะแต่มีรูปแบบการใช้งานพื้นฐานอยู่ 2 ลักษณะ คือ

```
SELECT*FORM ชื่อตาราง
```

หรือ

```
Select ฟิลด์ที่1,ฟิลด์ที่2,...ฟิลด์ที่n FORM ชื่อตาราง WHERE เงื่อนไข
```

| | |
|---|---|
| ชื่อตาราง | หมายถึง ชื่อตารางที่ต้องการดึงข้อมูล |
| ตัวแปร ฟิลด์ที่1,ฟิลด์ที่2,...ฟิลด์ที่ n | หมายถึง ชื่อฟิลด์ที่ต้องการดึงข้อมูลถ้ามีมากกว่า 1 ฟิลด์จะใช้เครื่องหมาย , คั่นระหว่างฟิลด์ |
| เงื่อนไข | หมายถึง เงื่อนไขในการดึงข้อมูลอาจเป็นคำสั่งทางคณิตศาสตร์ หรือเป็นคำสั่ง SELECT ซ้อนอยู่ข้างในก็ได้ |

**คำสั่ง UPDATE**

เป็นคำสั่งที่ใช้สำหรับแก้ไข หรือเปลี่ยนแปลงข้อมูลในเร็คคอร์ดที่มีอยู่ในตาราง มีรูปแบบการใช้งานดังต่อไปนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
UPDATE ชื่อตาราง SET ชื่อฟิลด์ = ค่าที่กำหนด WHERE เงื่อนไข
```

### 3.17.5 การเชื่อมโยงตาราง

**การเชื่อมโยงตารางโดยใช้คำสั่ง SELECT**

วิธีนี้เป็นการเชื่อมโยงตาราง โดยระบุฟิลด์ที่มีความสัมพันธ์กัน เช่น

```
SELECT Student,FirstName,LastName,MajorDesc
FROM student,major
WHERE student.MajorID = majorID
```

**การเชื่อมโยงตารางโดยใช้คำสั่ง INNER JOIN**

เป็นการเชื่อมโยงข้อมูลจาก 2 ตารางเข้าด้วยกัน โดยระบุชื่อฟิลด์ที่มีความสำคัญกันหลังคำสั่ง ON เป็นเงื่อนไขในการ JOIN ตาราง โดยที่ฟิลด์ดังกล่าว จะต้องเป็นข้อมูลชนิดเดียวกัน มีลักษณะเช่นเดียวกับการใช้คำสั่ง SELECT มีรูปแบบการใช้งานดังนี้

```
SELECT fieldname
FORM tablename 1
INNER JOIN tablename 2
ON tablename1.fieldname1 OPERATION tablename2.fieldname 2
```

| | |
|---|---|
| ตัวแปรfieldname | หมายถึงชื่อฟิลด์ที่อยู่ในตารางที่1 หรือ 2 |
| ตัวแปรtablename1 | หมายถึงชื่อตารางที่1 |
| ตัวแปรtablename2 | หมายถึง ชื่อตารางที่ 2 |
| ตัวแปรfieldname1 | หมายถึงชื่อฟิลด์ที่อยู่ตารางที่ 1 |
| ตัวแปรfieldname2 | หมายถึงชื่อฟิลด์ที่อยู่ตารางที่ 2 |
| ตัวแปรoperation | หมายถึงเงื่อนไขหรือตัวดำเนินการต่างๆที่ใช้ในการเชื่อมโยง |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**การเชื่อมโยงคำสั่งโดยให้คำสั่ง LEFT JOIN**

การเชื่อมโยง 2 ตารางเข้าด้วยกัน โดยกำหนดให้ตารางแรก(ตารางทางซ้าย ของคำสั่ง LEFT JOIN) เป็นหลัก แล้วนำตารางที 2 เข้ามาเชื่อมโยง ตามเงื่อนไขที่ได้ระบุไว้ มีรูปแบบการใช้งานได้ดังนี้

```
SELECT fieldname
FORM tablename 1 LEFT JOIN tablename 2
ON tablename1.fieldname1 OPERATION tablename2.fieldname 2
```

| | |
|---|---|
| ตัวแปร fieldname | หมายถึงชื่อฟิลด์ที่อยู่ในตารางที่1 หรือ 2 |
| ตัวแปร tablename1 | หมายถึงชื่อตารางที่1 |
| ตัวแปร tablename2 | หมายถึง ชื่อตารางที่ 2 |
| ตัวแปร fieldname1 | หมายถึงชื่อฟิลด์ที่อยู่ตารางที่ 1 |
| ตัวแปร fieldname2 | หมายถึงชื่อฟิลด์ที่อยู่ตารางที่ 2 |
| ตัวแปร operation | หมายถึงเงื่อน ไขหรือตัวดำเนินการต่างๆที่ใช้ในการเชื่อมโยง |

**การเชื่อมโยงคำสั่งโดยให้คำสั่ง RIGHT JOIN**

เป็นการเชื่อมโยง 2 ตารางเข้าด้วยกัน แต่กำหนดให้ตารางที่อยู่ทางด้านขวาของคำสั่ง RIGHT JOIN เป็นหลัก แล้วนำเร็คคอร์ดของตารางที่อยู่ทางด้านซ้ายของคำสั่ง RIGHT JOIN เข้ามาร่วมแสดง เฉพาะเร็คคอร์ดที่ตรงตามเงื่อนไขที่ระบุไว้มีรูปแบบการใช้งานดังนี้

```
SELECT fieldname
FORM tablename 1 RIGHT JOIN tablename 2
ON tablename1.fieldname1 OPERATION tablename2.fieldname 2
```

| | |
|---|---|
| ตัวแปร fieldname | หมายถึงชื่อฟิลด์ที่อยู่ในตารางที่1 หรือ 2 |
| ตัวแปร tablename1 | หมายถึงชื่อตารางที่1 |
| ตัวแปร tablename2 | หมายถึง ชื่อตารางที่ 2 |
| ตัวแปร fieldname1 | หมายถึงชื่อฟิลด์ที่อยู่ตารางที่ 1 |
| ตัวแปร fieldname2 | หมายถึงชื่อฟิลด์ที่อยู่ตารางที่ 2 |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

| | |
|---|---|
| ตัวแปร operation | หมายถึงเงื่อนไขหรือตัวดำเนินการต่างๆที่ใช้ในการเชื่อมโยง |

**3.17.6 เทคโนโลยี ActiveX Data Object**

ADO เป็นเทคโนโลยีที่มีแนวความคิดจาก DAO บางส่วน กล่าวคือ จะมองฐานข้อมูลเป็นออบเจ็กต์เช่นกัน แต่จะใช้ OLEDB Provider เป็นตัวจัดการ ข้อมูลและโครงสร้างของฐานข้อมูลแทน โดยที่จะไม่นิยาอออบเจ็กต์ขึ้นมา เพื่อแทนโครงสร้างของฐานข้อมูลแต่ละส่วนเหมือนกับ DAO แต่จะใช้วิธีการสร้างวิธีการสร้าง OLEDB Provider ให้กับ RDBMS แต่ละชนิดแทน เช่น เมื่อฐานข้อมูลเป็นชนิด JET ก็จะใช้ Microsoft Jet OLEDB Provider ถ้าเป็น Oracle ก็จะใช้ Microsoft OLEDB Provider for Oracle เป็นต้น ทำให้ออบเจ็กต์ในโมเดล ADO ไม่ยุ่งยากซับซ้อนเหมือนออบเจ็กต์ DAO

ใน Visual Basic สามารถเรียกใช้ OLEDB Provider ได้ 2 วิธีคือ

1. อาศัยคอนโทรล ADO Data (ADO Data Control) ร่วมกับกลุ่มคอนโทรล Bound & ActiveX Bound Control ที่มีคำว่า OLEDB ต่อท้าย
2. โดยเรียกใช้งานกลุ่มออบเจ็กต์ ADO โดยตรง

**ชุดออบเจ็กต์ใน ADO 2.5**

สำหรับในโมเดลของ ADO 2.5 จะประกอบด้วยออบเจ็กต์ 9 ตัวคือ

| | |
|---|---|
| ออบเจ็กต์ Command | เป็นออบเจ็กต์ตัวกลางที่ใช้สำหรับส่งคำสั่ง (Execute) ไปยังฐานข้อมูลเป้าหมายที่ต้องการติดต่อ มีสิทธิ์เรียกฐานข้อมูลดังกล่าวว่า Data Sources |
| ออบเจ็กต์ Command | ใช้สำหรับเริ่มต้นเปิดการเชื่อมต่อกับฐานข้อมูล |
| ออบเจ็กต์ Error | เป็นออบเจ็กต์ที่เก็บข้อมูลเกี่ยวกับ ความผิดพลาดของการเข้าถึงข้อมูลของ OLEDB Provider |
| ออบเจ็กต์ Field | ใช้สำหรับเก็บฟิลด์ใดฟิลด์หนึ่งที่อยู่ในฐานข้อมูล |
| ออบเจ็กต์ Parameter | ใช้สำหรับรับ-ส่ง ตัวแปร หรืออาร์กิวเมนต์ระหว่างการทำคิวรี่หรือ Stored Procedure ของออบเจ็กต์ Command |
| ออบเจ็กต์ Property | ใช้สำหรับกำหนดคุณสมบัติต่างๆ ของ ออบเจ็กต์ Recordset |
| ออบเจ็กต์ Recode | ใช้สำหรับเก็บเร็กกอร์ดใดเร็กกอร์ดหนึ่ง ที่อยู่ในออบเจ็กต์ Recordset |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

| | |
|---|---|
| ออบเจ็กต์ Recordset | เป็นผลลัพธ์ที่ได้มาจากฐานข้อมูล ประกอบไปด้วยกลุ่มของเร็คคอร์ด และ ฟิลด์ที่มาจากตารางที่เชื่อมต่อเป็นออบเจ็กต์ที่มีความสำคัญมากที่สุด |

### 3.17.7 วิธีติดต่อกับฐานข้อมูลของ MS-Visual Basic 6.0

โปรแกรม MS-Visual Basic 6.0 มีความสามารถในการจัดการกับฐานข้อมูลได้ โดยจะมีวิธีการจัดการอยู่ 2 แบบ คือ

1. การใช้ Data Control เป็นวิธีการที่ง่ายและสะดวกในการติดต่อกับฐานข้อมูลเนื่องจาก Data Control จะติดต่อกับฐานข้อมูลและจัดการกับฐานข้อมูลในตารางอัตโนมัติเช่น การเปิดฐานข้อมูลการแสดงและแก้ไขข้อมูลในตารางอย่างไรก็ดีการใช้ Data Control ก็มีข้อจำกัดอยู่พอสมควร เช่น ไม่มีฟังค์ชันในการลบข้อมูล

2. การใช้ Data Object วิธีนี้ต้องเขียนโปรแกรมเพื่อติดต่อกับฐานข้อมูล โดยการใช้ Object ต่างๆ ที่ VB6 มีมาให้โดยการเพิ่ม ลบ หรือแก้ไขข้อมูล จะต้องเขียนโปรแกรมเอง แต่ข้อดีของวิธีนี้คือสามารถติดต่อกับข้อมูลจากหลายๆ ตารางพร้อมกัน สามารถสร้างคิวรีตอนรันโปรแกรมได้ และสามารถควบคุมความผิดพลาดต่างๆ ได้ดีกว่าใช้ Data Control รวมทั้งสามารถใช้ภาษาSQL เพื่อจัดการกับฐานข้อมูลได้อีกด้วย

#### 3.17.7.1 การใช้งานดาต้าคอนโทรล (Data Control)

ดาต้าคอนโทรลซึ่งเป็นเครื่องมือที่ช่วยให้สามารถจัดการกับข้อมูลได้ในตารางเช่น การเพิ่ม ลบ แก้ไข หรือการค้นหาข้อมูลที่ถูกต้องเป็นต้น

คุณสมบัติที่สำคัญสำหรับดาต้าคอนโทรล

| | |
|---|---|
| DataSource | เป็นการกำหนดชื่อของดาต้าคอนโทรลที่ต้องการเชื่อมโยง |
| DataFeild | เป็นการกำหนดว่าเราต้องการให้แสดงหรือแก้ไขข้อมูลฟิลด์ใดในตาราง |
| DatabaseName | เป็นที่อยู่ของไฟล์ฐานข้อมูลที่ต้องการ โดยมีการระบุชื่อฟิลด์นั้น |
| Connect | เป็นคุณสมบัติที่ใช้บอกว่าต้องการติดต่อกับฐานข้อมูลประเภทใด เช่น MS-Access |
| RecordSetType | เป็นชนิดของเร็กคอร์ดเซตที่ต้องการเข้าถึงฐานข้อมูล |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# บทที่ 4
# ขั้นตอนและวิธีการดำเนินงาน

## 4.1 กล่าวนำ

ขั้นตอนและวิธีการดำเนินงานต่างๆในโครงงานนี้ จะเริ่มตั้งแต่การศึกษาหลักการผลิตแบบ Just In Time ออกแบบการผลิตแบบ Just In Time ศึกษาหลักการทำงานของชุด Ware House ศึกษาหลักการทำงานของชุดสายพานลำเรียงโดยใช้ระบบ Fuzzy ในการวิเคราะห์สีของวัตถุ และศึกษาอุปกรณ์ต่าง ๆ ที่จะนำมาใช้ประกอบเป็นชุดทดลองจนถึงการเขียนโปรแกรมเพื่อให้ระบบสามารถดำเนินงานตามคำสั่งของผู้ใช้ได้ ในรายละเอียดต่างๆทั้งหมดนี้จะถูกนำมากล่าวในบทนี้ซึ่งขั้นตอนและวิธีการดำเนินงานดังกล่าวได้ดำเนินการตามขั้นต่อดังต่อไปนี้

## 4.2 ศึกษาหลักการผลิต Just In Time และออกแบบจำลองกระบวนการผลิต

ระบบการผลิตแบบ Just In Time เป็นกระบวนการผลิตที่พยายามจะทำให้ได้ผลกำไรมากขึ้น หรือมากที่สุดก็ว่าได้ โดยที่ไม่ได้มีการเพิ่มราคาในการขาย แต่เป็นการลดต้นทุนในการผลิตให้เหลือน้อยที่สุด โดยการตัดหรือลอกกระบวนการที่สูญเปล่าออกหรือให้ลดน้อยลงเท่าที่ทำได้ เช่น จากการที่เราผลิตของในครั้งหนึ่งเราจะผลิตจำนวนมากแล้วสต๊อกเก็บไว้ทำให้เกิดการเพิ่มต้นทุนในกระบวนการผลิตจากการดูแลรักษาสินค้า และพื้นที่ในการเก็บสินค้า ทำให้ผลกำไรที่ได้ลดน้อยลงจากสมการ

**ราคาขาย – ต้นทุน = ผลกำไร**

เมื่อเปลี่ยนกระบวนการการผลิตมาเป็นแบบ Just In Time จะเป็นการผลิตสินค้า เมื่อมีการสั่งสินค้าเข้ามาจึงทำให้ลดต้นทุนในการสต๊อกสินค้า ทำให้เกิดผลกำไรมากขึ้น

เมื่อเข้าใจหลักการของกระบวนการผลิตแบบ Just In Time แล้ว จึงได้มีการออกแบบจำลองกระบวนการผลิตแบบ Just In Time โดยใน Project นี้เราได้แยกแบบการจำลองออก 2 ส่วนด้วยกัน คือ

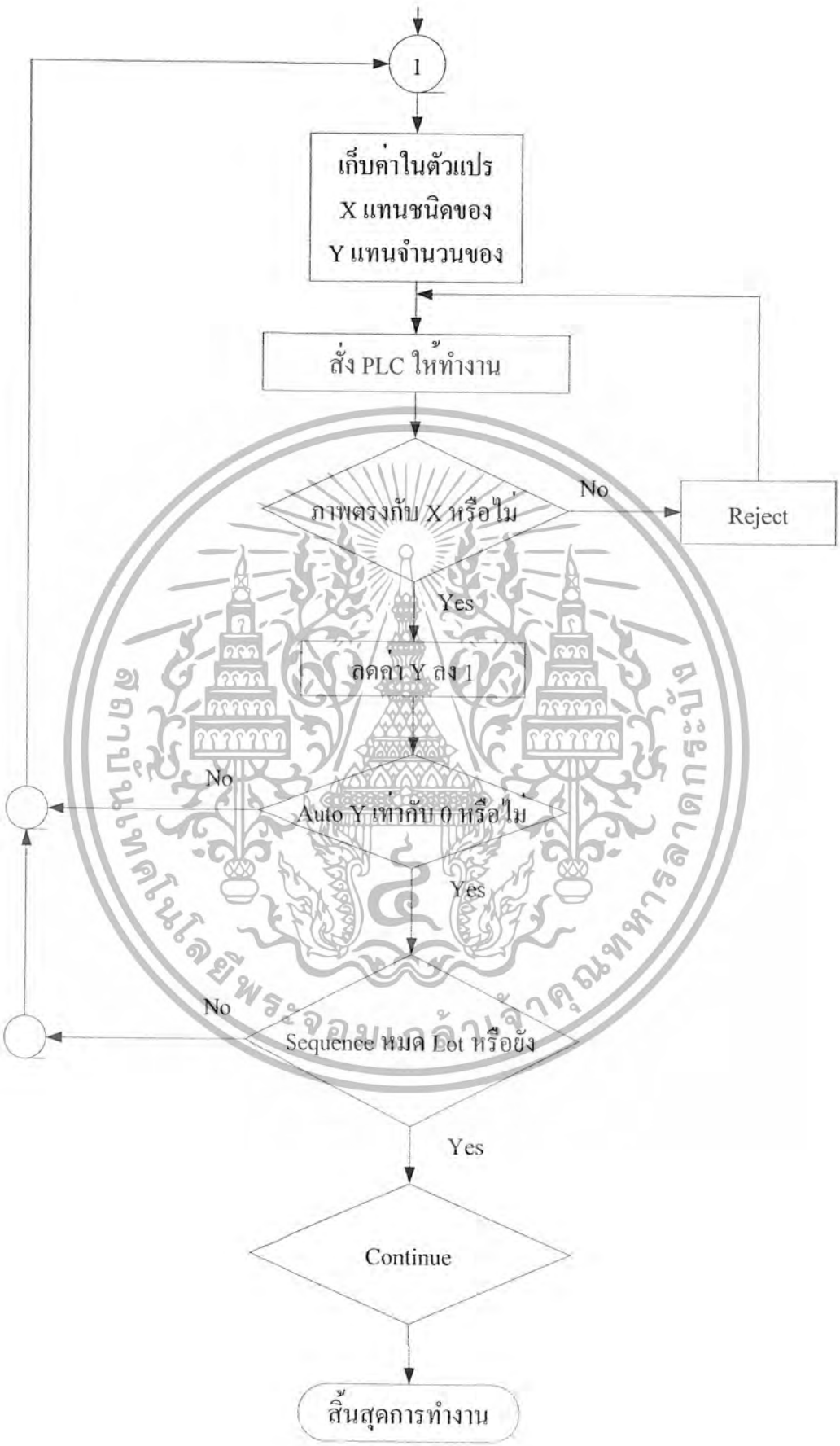
1. ส่วนของสายการผลิตหลัก ซึ่งในที่นี้ได้นำเอาเครื่อง Warehouse มาสมมุติใช้แทน
2. ส่วนของสายการผลิตย่อย ซึ่งในที่นี้ได้นำเอาชุดสายพานลำเลียง มาสมมุติใช้แทน

ลำดับขั้นการทำงานจะแสดงโดย Flow Chart ที่ 1 และ 2

**Flow Chart ที่1 สายการผลิตย่อย**

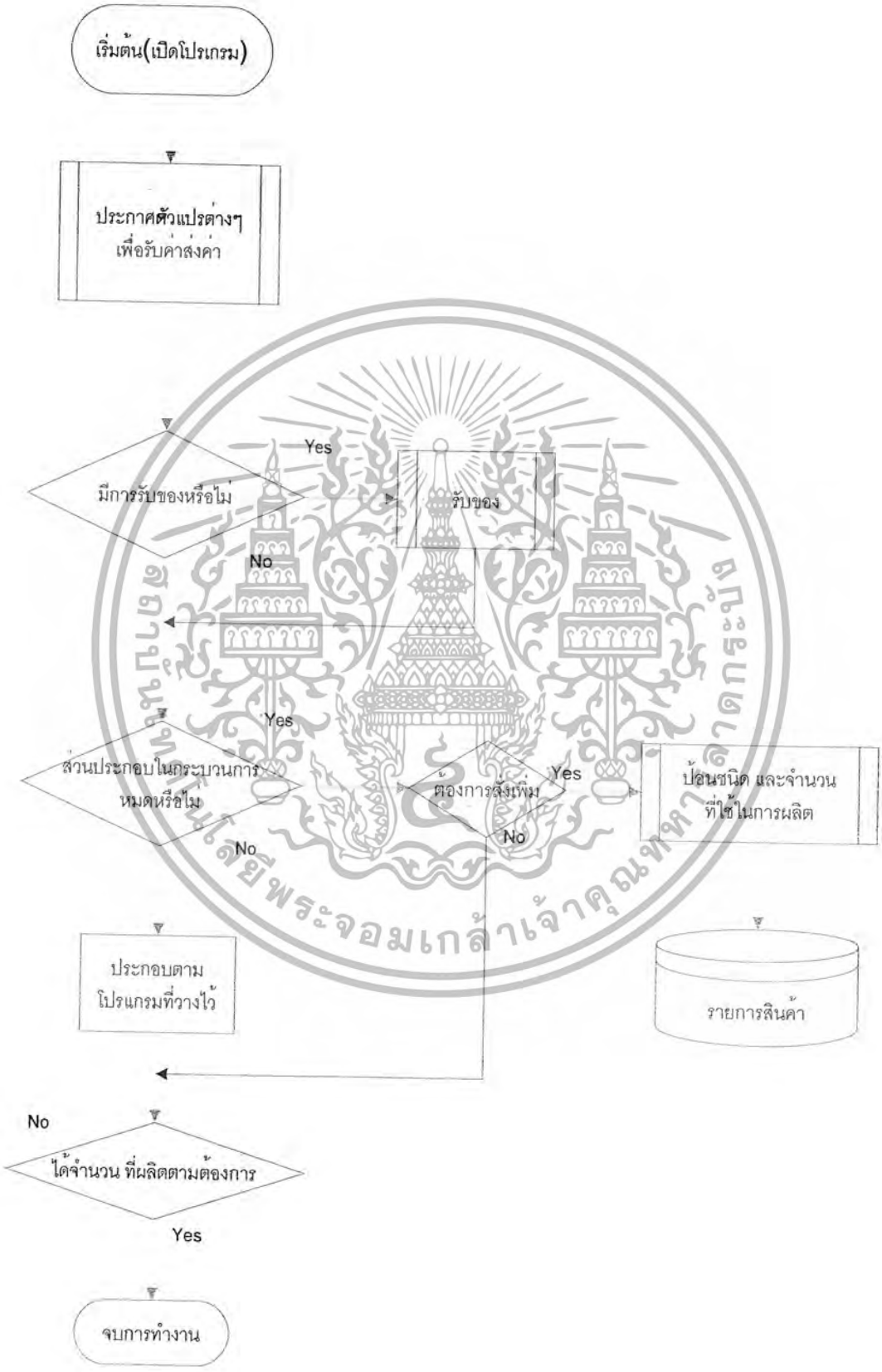
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

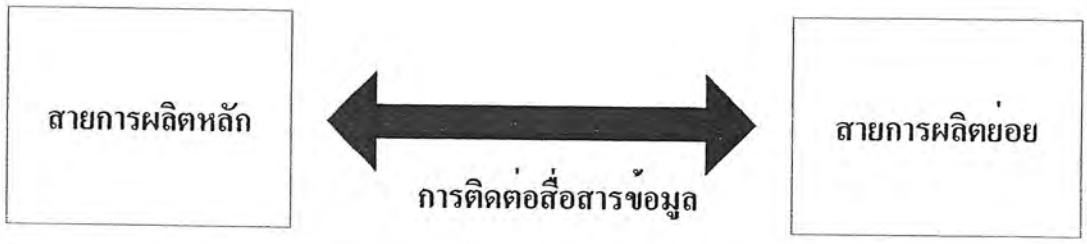



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**Flow Chart ที่2 สายการผลิตหลัก**




เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้




**ภาพที่ 4.1** แสดงแบบจำลองกระบวนการผลิต

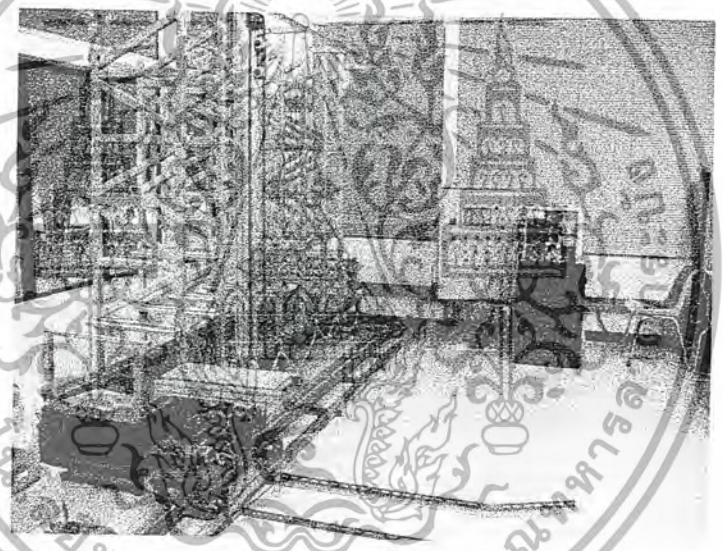
**ภาพที่4.2** สายการผลิตหลัก

**ภาพที่ 4.3** แสดงสายการผลิตย่อย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 4.3 ศึกษาหลักการทำงาน วิธีใช้และตรวจสอบการทำงานเครื่อง Warehouse

หลังจากสรุปได้ว่าจะใช้เครื่อง Warehouse มาสมมุติใช้เป็นสายการผลิตหลัก จึงได้นำปริญญานิพนธ์ของรุ่นพี่ เรื่องการออกแบบจัดระบบจัดเก็บผลิตภัณฑ์ผ่านเครือข่ายอินเตอร์เน็ท เพื่อทำความเข้าใจวิธีการใช้ และหลักการทำงานของเครื่อง Warehouse เมื่อเข้าใจวิธีการใช้เครื่อง Warehouse แล้วจึงได้มีการทดลองใช้ ทั้งในโหมด Auto and Manual และเป็นการตรวจสอบไปด้วยในตัวว่าเครื่องมีการทำงานเป็นปกติหรือไม่ และถ้ามีความไม่ปกติเกิดขึ้น จะได้รู้ว่ามีความไม่ปกติ ณ จุดใด แล้วจึงทำการแก้ไขต่อไป ผลจากการตรวจเช็ค Warehouse มีปัญหาในการวัดระยะ คือ ระยะในแต่ละหลักมีการคลาดเคลื่อน จากที่ตั้งค่าของระยะไว้ จึงได้ทำการตั้งระยะห่างใหม่ และทดสอบการทำงานอีกครั้ง เพื่อเช็คดูว่า Warehouse ทำงานปกติดีหรือไม่ ผลการทดลองในครั้งนี้ Warehouse มีการทำงานปกติดี



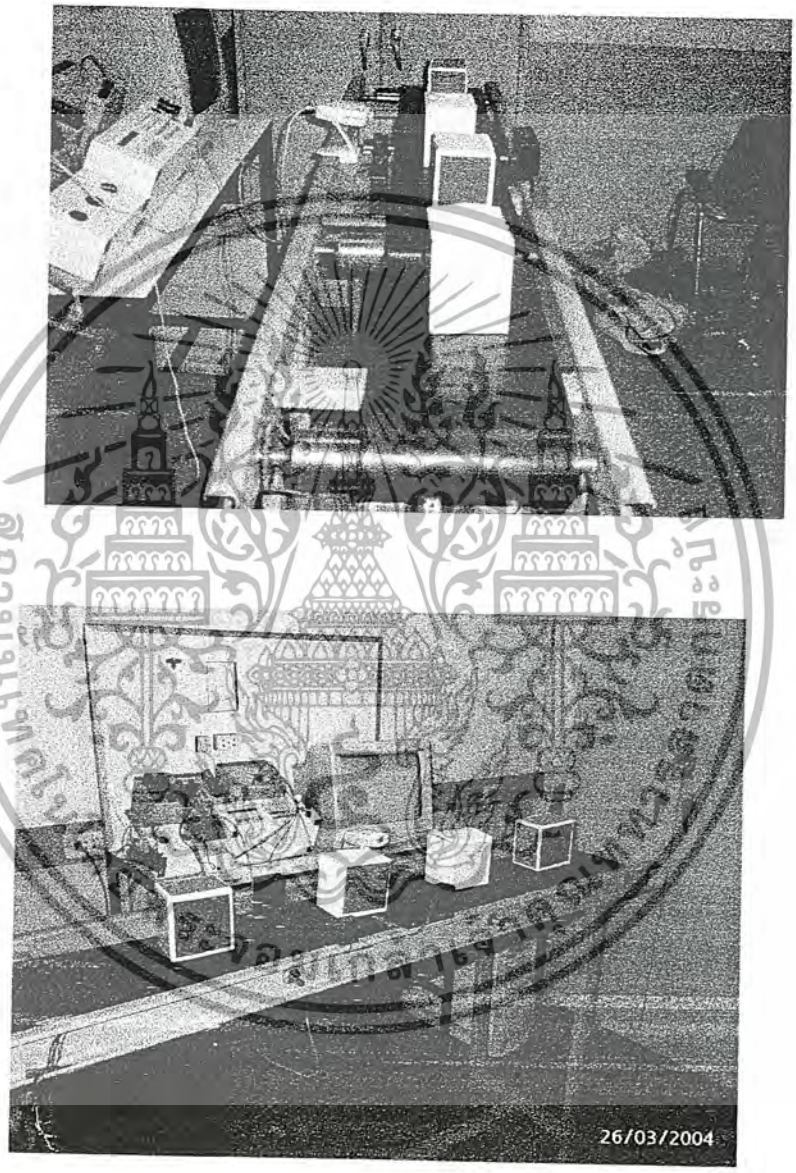
ภาพที่ 4.4 แสดงWarehouse

## 4.4 ศึกษาหลักการทำงาน วิธีการใช้งานชุดสายพานลำเลียง

เมื่อสรุปแล้วว่าเราจะใช้ชุดสายพานลำเลียงมาประยุกต์ใช้ เป็นสายการผลิตย่อย จึงได้มีการนำปริญญานิพนธ์ของรุ่นพี่ เรื่องการออกแบบจัดระบบจัดเก็บผลิตภัณฑ์ผ่านเครือข่ายอินเตอร์เน็ท มาศึกษาวิธีการใช้งาน และหลักการทำงานต่างๆ ของจุดสายพานลำเลียง และทดลองการทำงานของชุดสายพานลำเลียงว่าทำงานได้ตามปกติหรือไม่ และมีการทำงานตรงกับหลักการที่ต้องการหรือไม่ ผลจากการทดลองปรากฏว่าชุดสายพานลำเลียงทำงานได้อย่างปกติ แต่หลักการทำงานของชุดสายพานลำเลียงไม่ตรงตามความต้องการที่เราจะใช้งาน จึงได้มีการปรับเปลี่ยนหลักการทำงานใหม่และ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ได้ทำตัวคัดแยกวัตถุเข้าไปอีกเพื่อใช้ในการคัดแยกวัตถุที่ไม่ต้องการออกจาสายพานลำเลียงและต้องเขียนโปรแกรม Ladder ป้อนเข้าไปให้กับ PLC ใหม่ และทำการทดลองการทำงาน ผลที่ได้จากการทดลองการทำงาน ชุดสายพานลำเลียงที่ได้แก้ไขและเขียนโปรแกรมใหม่นั้น สามารถทำงานได้ตามความต้องการ




**ภาพที่ 4.5** แสดงชุดสายพานลำเรียง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 4.5 ศึกษาความสัมพันธ์ของอุปกรณ์ต่างๆ ที่ทำงานกันในระบบ

หลังจากที่ได้ทำการตรวจสอบโครงสร้าง และชุดอุปกรณ์เรียบร้อยแล้ว ในขั้นตอนนี้เราจึงทำการศึกษาการทำงานของอุปกรณ์ต่างๆ ว่าทำงานสัมพันธ์กันอย่างไร และทำการออกแบบการใช้งานในแต่ละชุด พร้อมทั้งทำการออกแบบการต่อสายเพื่อทำการใช้งาน เพื่อใช้ในการสั่งการทำงานของ Warehouse และชุดสายพานลำเลียง ให้เหมาะสม และในที่นี้ได้เลือกการติดต่อสื่อสารเป็น 2 ประเภท คือ การติดต่อสื่อสารระหว่าง Computer กับ Computer โดยผ่านระบบ Network และการติดต่อระหว่าง Computer กับ PLC โดยใช้พอร์ตอนุกรม RS 232 ในการติดต่อสื่อสาร

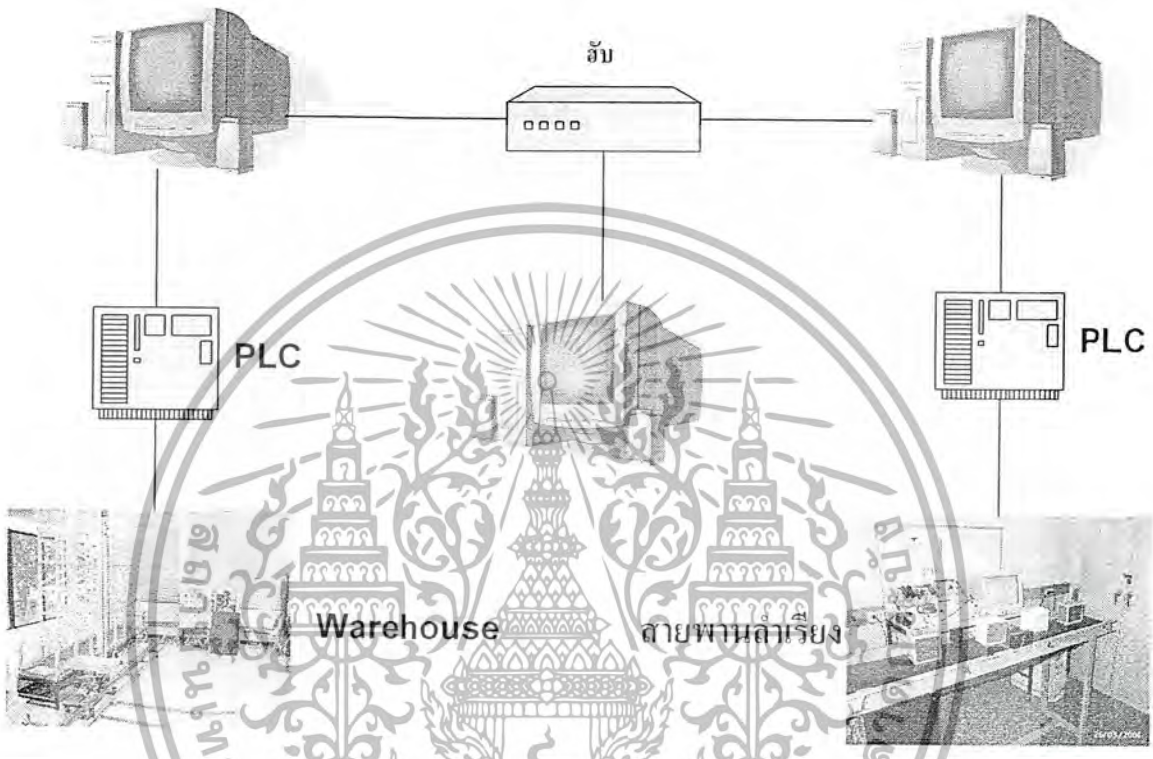
## 4.6 ศึกษาเกี่ยวกับ Visual Basic

หลังจากที่ทุกอย่างสำเร็จลุล่วงไปด้วยดีแล้ว ในขั้นตอนต่อไปนี้ เราได้ทำการศึกษาเกี่ยวกับโปรแกรม Visual Basic ซึ่งเราจะนำมาใช้ในการเขียนโปรแกรม เพื่อควบคุมPLC สามารถติดต่อกับ Computer ได้ และเขียนโปรแกรมเพื่อติดต่อสื่อสารส่งข้อมูลระหว่าง Computer กับ Computer ได้ ซึ่งในโครงงานนี้เราจะใช้ Computer เป็นตัวเก็บฐานข้อมูลต่างๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# บทที่ 5
# การทดสอบโครงงาน

## 5.1 กล่าวนำ




ภาพที่ 5.1 แสดงแบบจำลองการผลิต

จากแบบจำลองกระบวนการผลิตนี้เป็นจำลองกระบวนการผลิตแบบ Just In Time System โดยมีขั้นตอนการทำงานดังนี้ เมื่อมี Order มายังสายการผลิตหลักในที่นี้ได้สมมุติใช้เครื่อง Warehouse เป็นสายการผลิตหลัก ชุดสายการผลิตหลักจะทำการสั่ง Order อุปกรณ์ที่ต้องการนำมาประกอบไปยังสายการผลิตย่อยโดยสั่งผ่านระบบฐานข้อมูล โดยสายการผลิตย่อยนั้นเราได้สมมุติใช้ชุดสายพานลำเรียง เมื่อชุดกระบวนการผลิตย่อยได้รับว่ามี Order เข้ามาจะทำการอ่านข้อมูลจากระบบฐานข้อมูลว่าชุดสายการผลิตหลักต้องการอุปกรณ์ชนิดใดบ้าง เมื่อทราบว่าสายการผลิตหลักต้องการอุปกรณ์ชนิดใดและจำนวนอุปกรณ์ที่ต้องการ ชุดสายการผลิตย่อยจะทำการผลิตอุปกรณ์ โดยขั้นตอนการผลิตนี้มีหลักการทำงานดังนี้ เครื่องจะทำการลำเรียงอุปกรณ์มาเมื่อถึงตัวเซนเซอร์ เครื่องจะทำการถ่ายภายด้วยกล้อง Webcam และทำภาพที่ถ่ายไปวิเคราะห์สีเพื่อเปรียบเทียบว่าตรงตามที่ต้องการหรือไม่ โดยการวิเคราะห์สีนี้เราได้ใช้ระบบ Fuzzy เป็นตัววิเคราะห์ หลังจากนั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เครื่องจะทำการลำเรียงต่อมาตัวเซนเซอร์ตัวที่สองเมื่อมาถึงจุดนี้ ถ้าหากเป็นอุปกรณ์ที่ไม่ตรงตามความต้องการเครื่องจะทำการคัดแยกอุปกรณ์นั้นออกโดยกระบอกสูบ และถ้าหากเป็นอุปกรณ์ที่ตรงตามความต้องการเครื่องจะลำเรียงต่อมายังถาดรับอุปกรณ์ และจะกระทำซ้ำจนครบตามจำนวนที่มีการสั่งเข้ามาเครื่องจะหยุดทำงานแบบอัตโนมัติ เมื่อสายการผลิตย่อยผลิตอุปกรณ์ได้จะทำการส่งมอบให้ชุดสายการผลิตหลักเพ่อให้ชุดสายการผลิตหลักทำการประกอบและส่งให้แก่ลูกค้า ซึ่งตรงตามหลักการในระบบ Just In Time System ที่ว่าไม่มีการเก็บอุปกรณ์คงคลัง

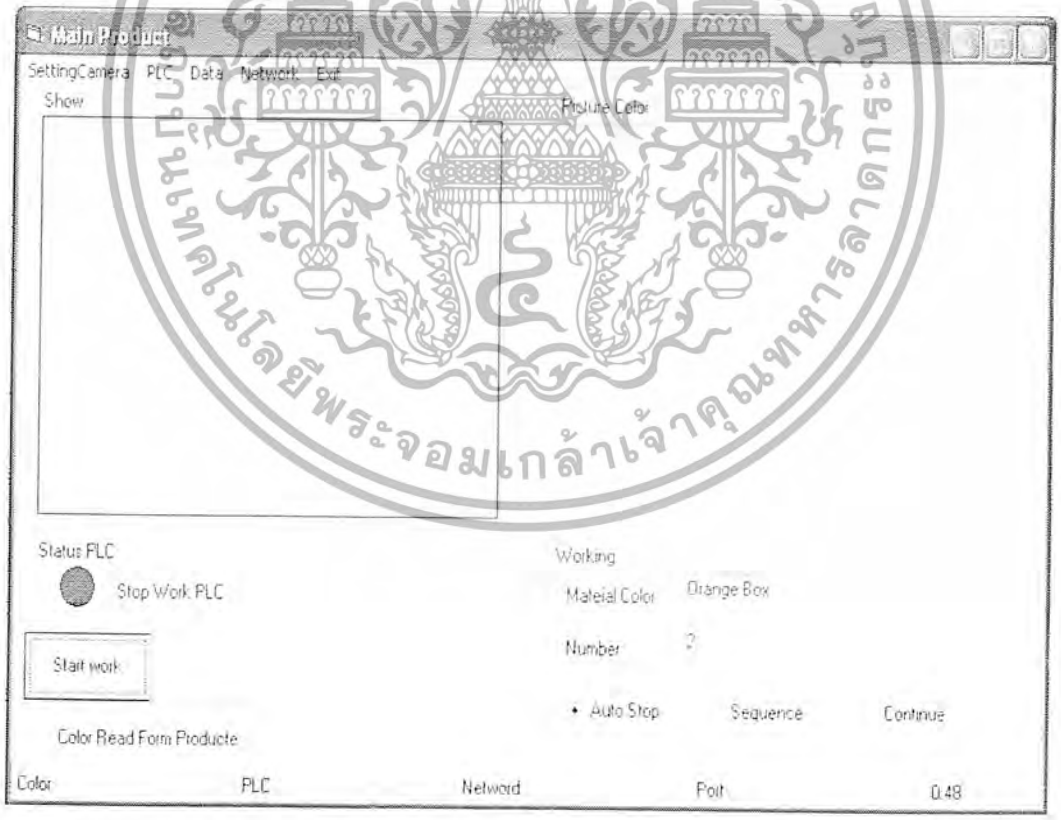
ในโครงงานนี้ได้แบ่งกระบวนการผลิตออกเป็น 2 ส่วนด้วยกัน คือ

1. การทดสอบในส่วนสายการผลิตหลัก
2. การทดสอบในส่วนสายการผลิตย่อย

## 5.2 การทดสอบในส่วนสายการผลิตย่อย

### 5.2.1 ลำดับขั้นการทดลองในส่วนสายการผลิตย่อย

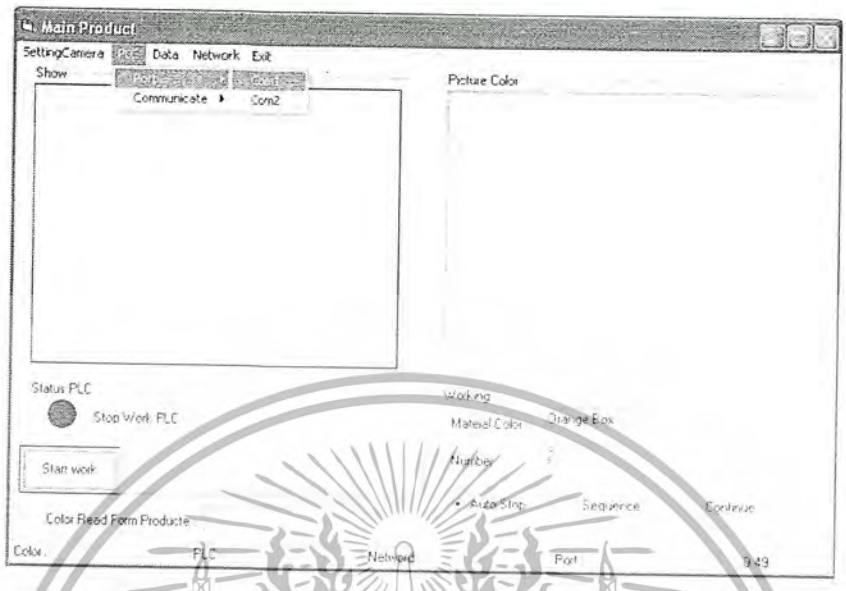
1. ทำการเปิดโปรแกรมซึ่งใช้ในการทำงาน ซึ่งแสดงรูปแบบดังภาพที่ 5.2




**ภาพที่ 5.2** แสดงรูปแบบของโปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

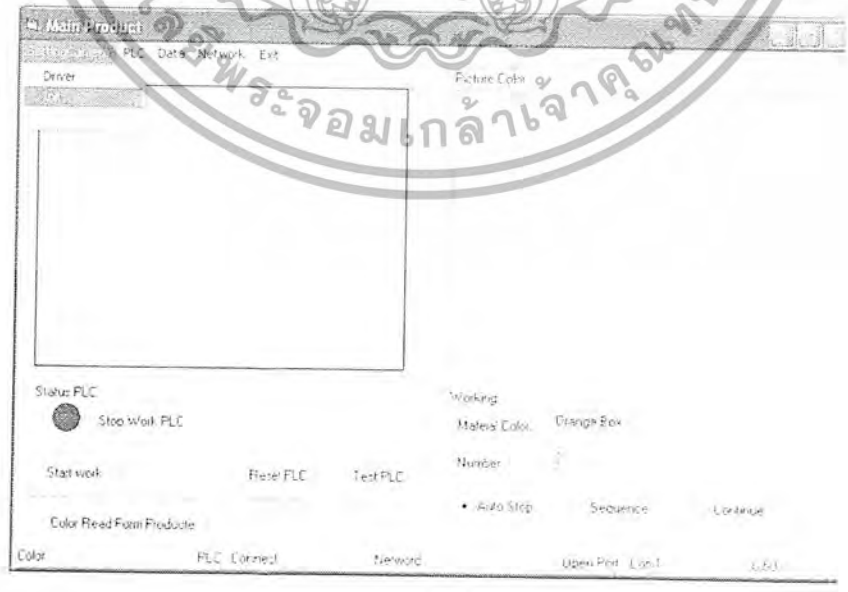
2. ทำการเลือกพอร์ตในการต่อใช้งาน โดยการคลิก Mouse ไปที่ PLC → Port → Com1 ดังรูป




**ภาพที่ 5.3** แสดงวิธีการคลิก Mouse เลือกพอร์ตในการติดต่อ

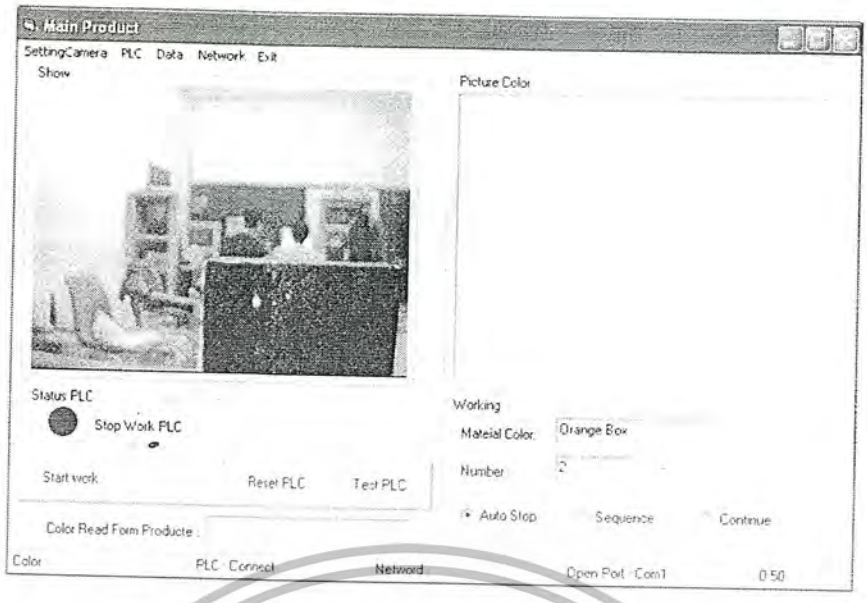
3. ทำการติดต่อสื่อสารระหว่าง Computer กับ PLC โดยการคลิก Mouse ไปที่ PLC → Communicate → Connect

4. สั่งการทำงานให้กล้องเริ่มทำงานโดยการคลิก Mouse ไปที่ Setting Camera → Init ดังรูป



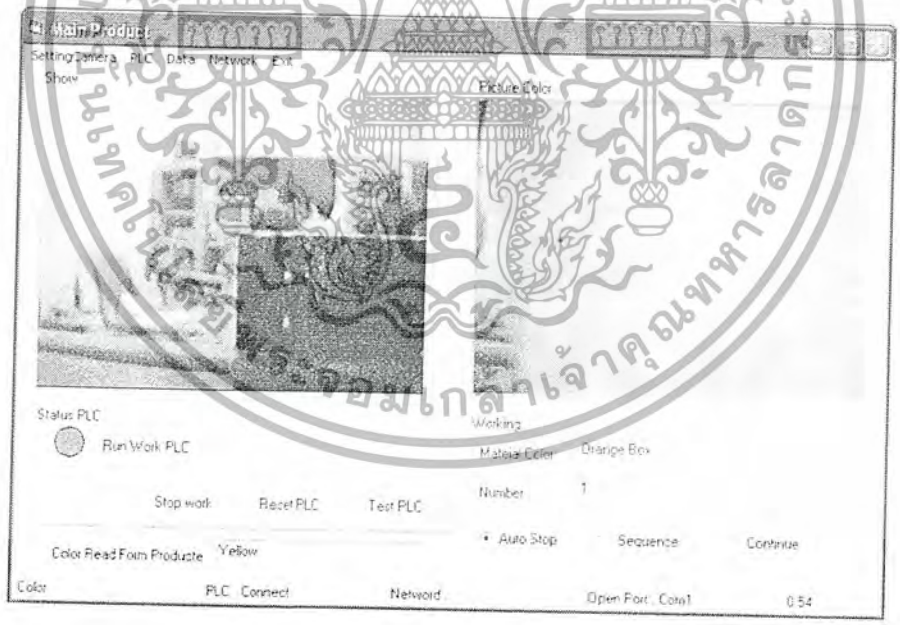

**ภาพที่ 5.4** แสดงการสั่งให้กล้องเริ่มทำงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้




**ภาพที่ 5.5** แสดงเมื่อสั่งการทำงานให้กล้องเริ่มทำงาน

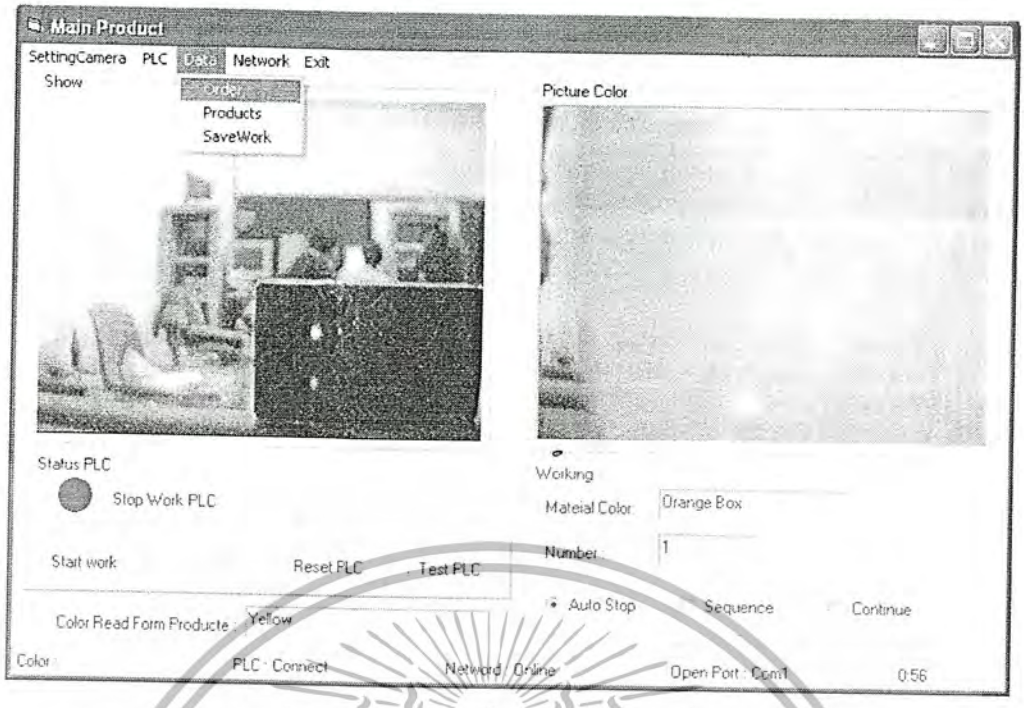
5. คลิก Mouse ไปที่ Start Work ชุดสายพานลำเลียงจะเริ่มการทำงาน และมีการขัดแยกอุปกรณ์ที่ไม่ต้องการออก โดยการตรวจเช็กสีของอุปกรณ์




**ภาพที่ 5.6** แสดงการถ่ายภาพไปวิเคราะห์สี

6. เมื่อต้องการทราบถึงจำนวนและชนิดของอุปกรณ์ที่จะต้องผลิตให้คลิก Mouse ไปที่ DATA ▸ Order ดังรูป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

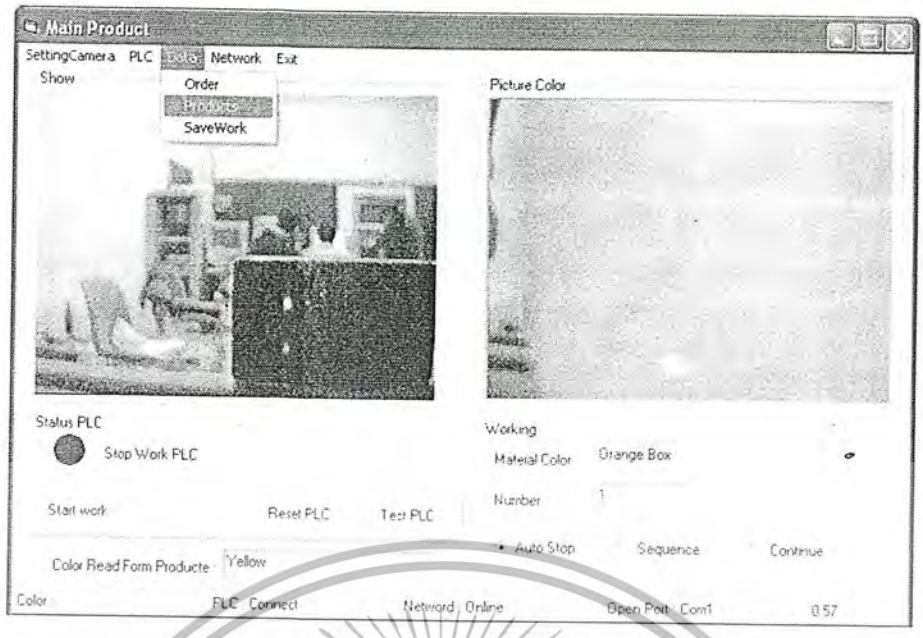



ภาพที่ 5.7 แสดงการคลิก Mouse เพื่อดูจำนวนอุปกรณ์ที่จะผลิต

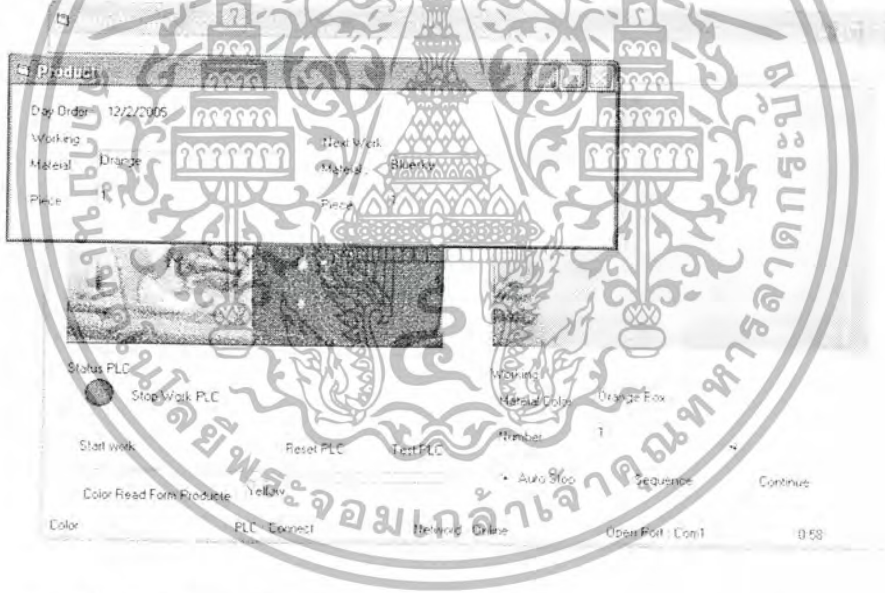
**ภาพที่ 5.8** แสดงจำนวนและชนิดของอุปกรณ์ที่จะผลิต

7. เมื่อต้องการทราบว่าสายการผลิตกำลังผลิตอุปกรณ์ชนิดใดอยู่และจะผลิตอุปกรณ์ชนิดใดต่อไปให้คลิก Mouse ไปที่ DATA → Products ดังรูป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



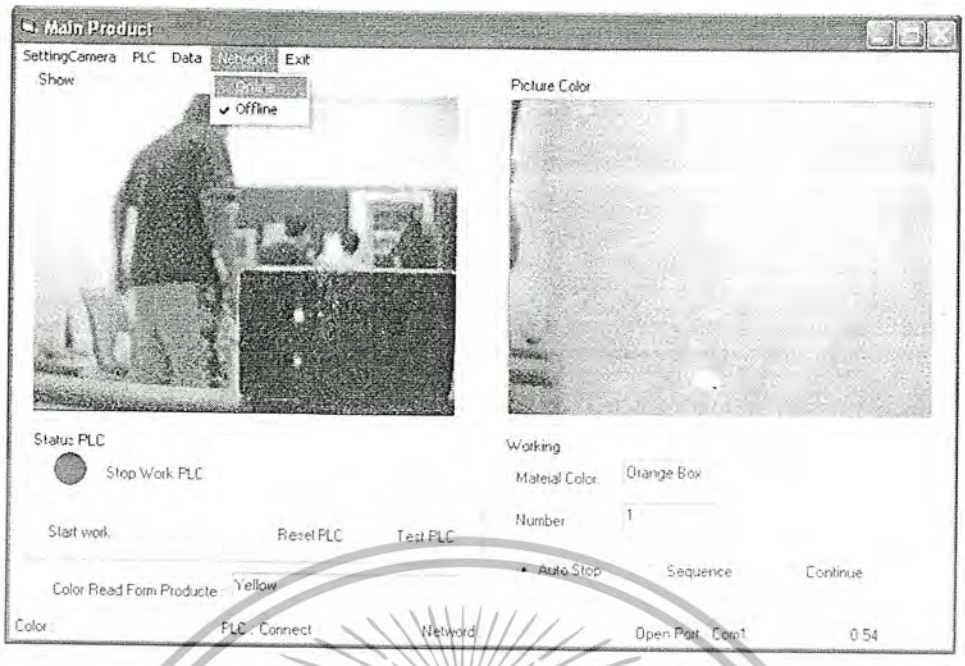

**ภาพที่ 5.9** แสดงการคลิก Mouse เพื่อดูว่าขณะนี้กำลังผลิตอุปกรณ์ชนิดใดอยู่



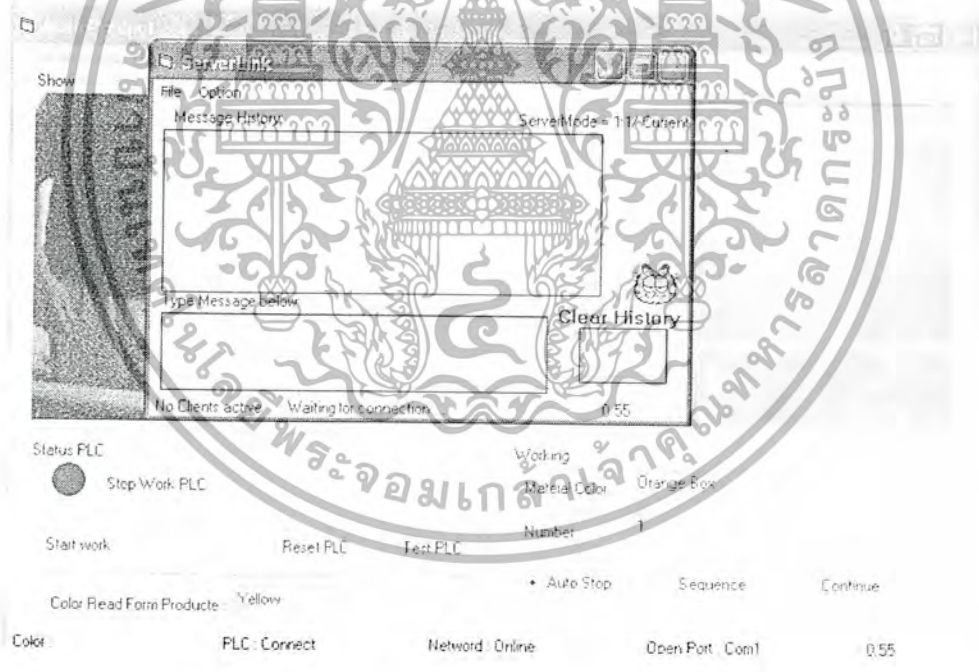

**ภาพที่ 5.10** แสดงให้ดูว่ากำลังผลิตอุปกรณ์ชนิดอยู่และจะผลิตอะไรต่อไป

8. ในโปรแกรมตัวนี้ได้ติดตั้งโปรแกรมติดต่อสื่อสารเพื่อใช้ในการติดต่อถามจำนวนการผลิตและอีกหลายๆอย่าง โดยการคลิก Mouse ไปที่ Network ▸ Online ในการเปิดการติดต่อ และ คลิก Mouse ไปที่ Network ▸ Offline ในการปิดการติดต่อดังรูป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้




**ภาพที่ 5.11** แสดงการคลิก Mouse เพื่อติดต่อสื่อสารระหว่างเครื่องแม่และเครื่องลูก




**ภาพที่ 5.12** แสดงรูปโปรแกรมการติดต่อสื่อสาร

### 5.1.2 สรุปผลการทดลองในส่วนสายการผลิตย่อย

ในโปรแกรมนี้สามารถติดต่อกันระหว่างผู้สั่ง Order และผู้รับ Order เป็นการพูดคุยก่อนการสั่ง Order โดยปริมาณสามารถเช็กได้จากฐานข้อมูล เมื่อเราเริ่มการทำงานชุดสายพานลำเรียงจะ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เริ่มทำกระบวนการผลิต ในที่นี้จะสมมุติให้กล้องที่มีแถบสีติดอยู่แทนอุปกรณ์ที่ผลิต จะไหลตามสายพานลำเรียง เมื่ออุปกรณ์ไหลผ่านเซนเซอร์ตัวแรกกล้องจะทำการถ่ายภาพเพื่อนำภาพที่ได้ไปเปรียบเทียบโดยระบบ Fuzzy ว่าอุปกรณ์นั้นตรงตามที่ต้องการหรือไม่จากแถบสี โดยสีที่เป็นตัวอ้างอิงว่าเป็นอุปกรณ์ที่ต้องการนั้นมาจากฐานข้อมูล เมื่อทำการวิเคราะห์ภาพโดยระบบ Fuzzy เสร็จแล้ว จะระบุว่าเป็นอุปกรณ์ที่ต้องการหรือไม่ หากเป็นอุปกรณ์ที่ไม่ต้องการเมื่อไหลผ่านเซนเซอร์ตัวที่สองจะทำการคัดออกจากสายการผลิต และถ้าหากเป็นอุปกรณ์ที่ต้องการจะไหลไปจนกระทั่งถึงถาดรับอุปกรณ์และจะทำเช่นนี้ไปเรื่อยจนกว่าจะได้รับอุปกรณ์ที่ตรงตามความต้องการครบตามจำนวนที่มีการ Order มา

## 5.3 การทดลองในส่วนสายการผลิตหลัก

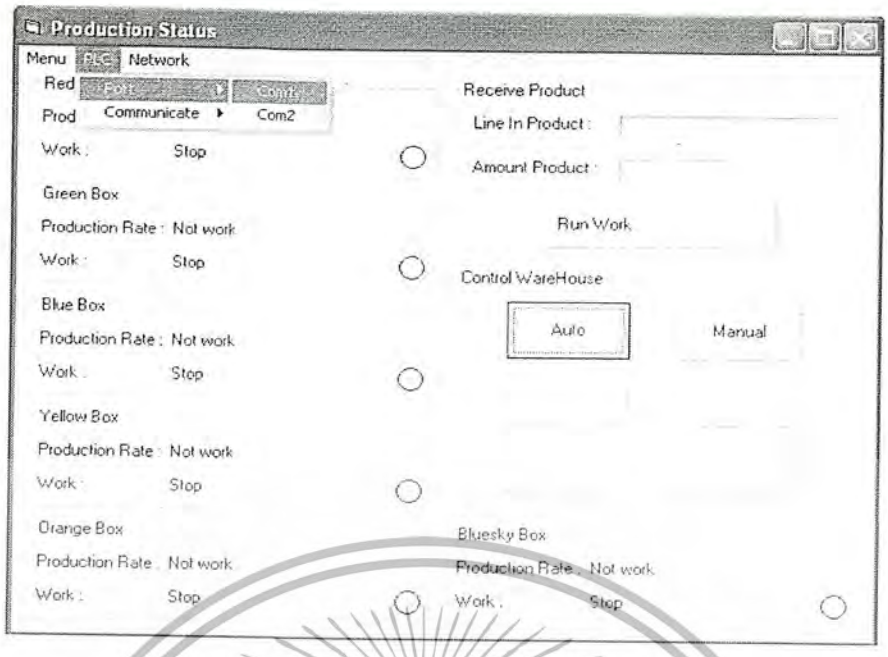
### 5.3.1 ลำดับขั้นการทดลองในส่วนสายการผลิตหลัก

1. เมื่อทำการเปิดโปรแกรมในส่วนสายการผลิตหลักจะพบกับหน้าต่างดังภาพที่ 5.12

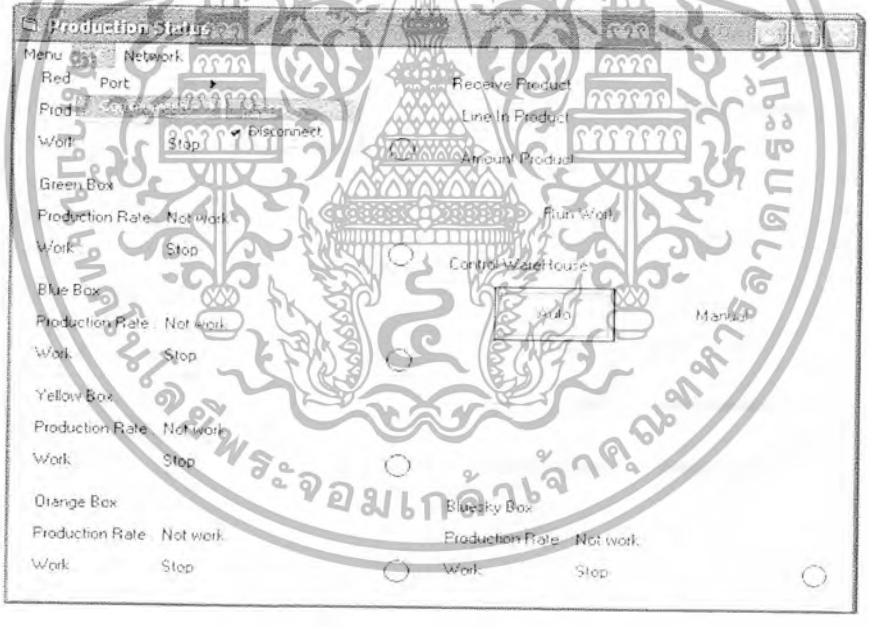
**ภาพที่ 5.13** แสดงรูปตัวโปรแกรมในส่วนสายการผลิตหลัก

2. เลือกพอร์ตในการติดต่อกับ PLC โดยการคลิก Mouse ไปที่ PLC → Port → Com1 ดังภาพที่ 5.13

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้




**ภาพที่ 5.14** แสดงวิธีการเลือกพอร์ต

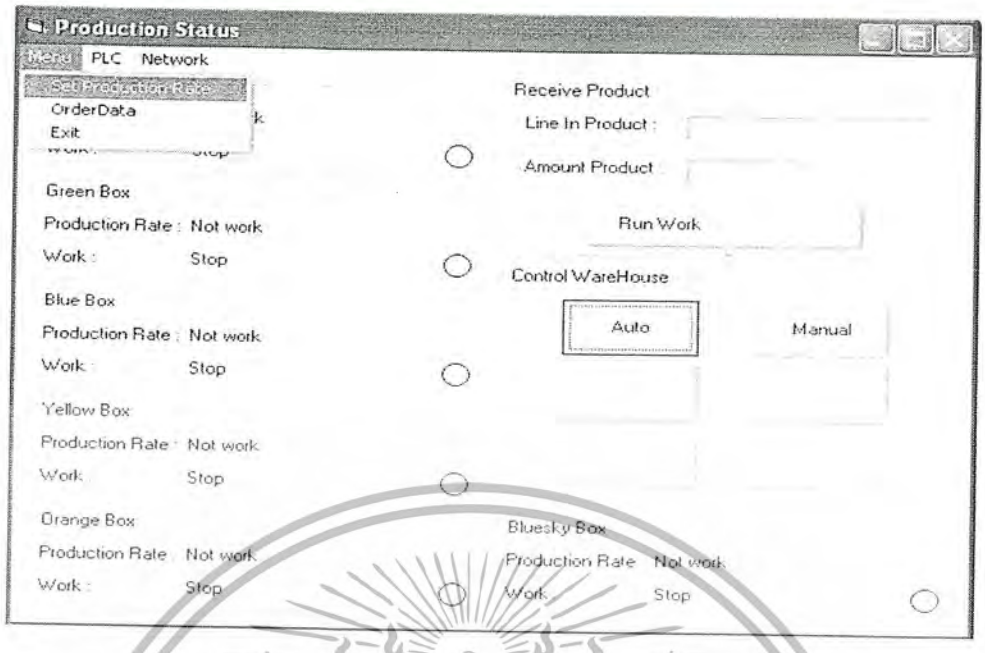



**ภาพที่ 5.15** แสดงวิธีการ Connect

3. ทำการ Connect โดยการคลิก Mouse ไปที่ PLC ▸ Communicate ▸ Connect ดังภาพที่ 5.14

4. เมื่อต้องการที่จะสั่ง Order ไปยังชุดสายการผลิตย่อยให้คลิก Mouse ไปที่ Menu ▸ Set Production Rate ดังภาพที่ 5.15

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

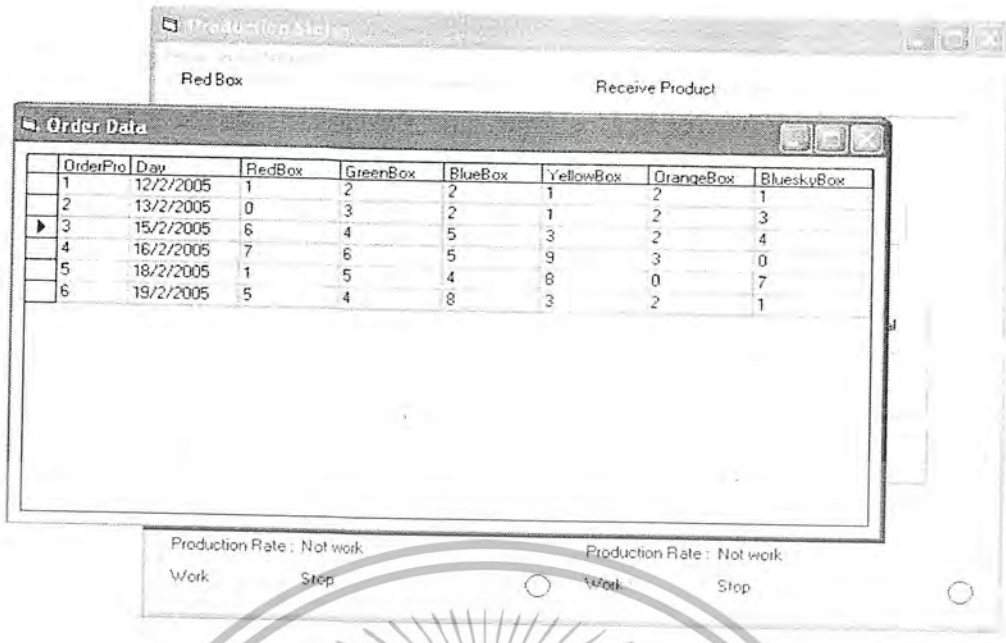



**ภาพที่ 5.16** แสดงการที่จะเข้าไปสั่ง Order

**ภาพที่ 5.17** แสดงหน้าจอที่ให้สั่ง Order

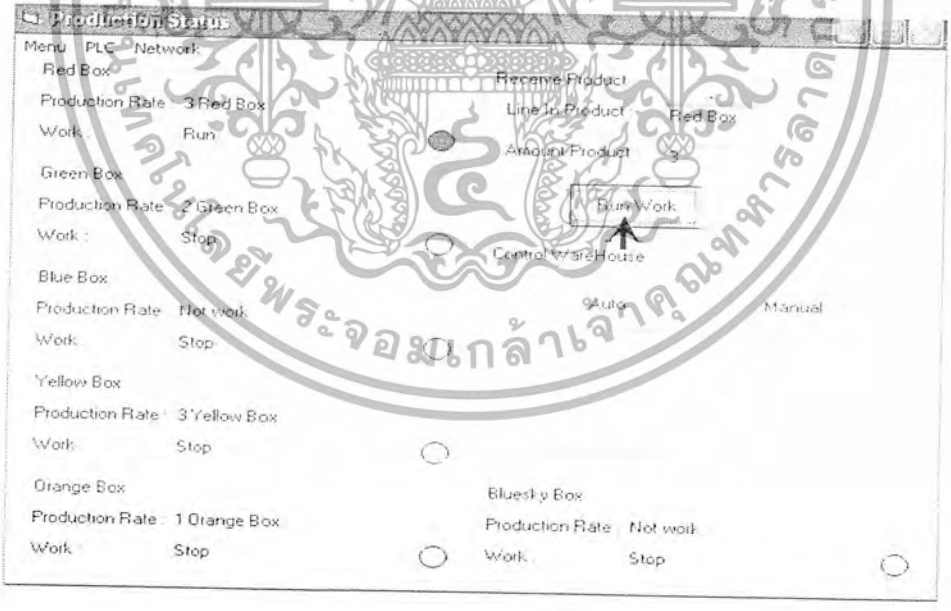
5. เมื่อต้องการดูรายงานการผลิตให้เราคลิก Mouse ไปที่ Menu • Order Data จะแสดงหน้าจอขึ้นดังรูปที่5.17

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้




**ภาพที่ 5.18** แสดงรายงานการผลิต

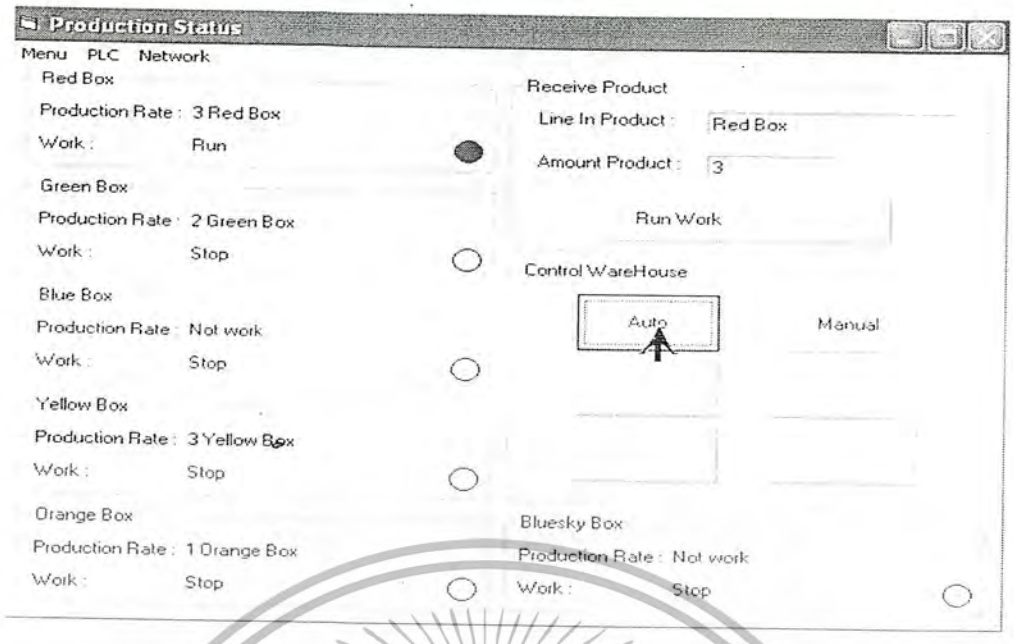
6. ให้กดปุ่มเริ่มการทำงานของสายการผลิตหลักโดยการคลิก Mouse ไปที่ Run Work ดังภาพที่ 5.18




**ภาพที่ 5.19** แสดงการกดปุ่มเริ่มการทำงานสายการผลิตหลัก

7. กดเลือกการทำงานในชุดสายการผลิตหลักว่าจะให้ทำงานในโหมด Auto หรือ Manual ดังภาพที่ 5.19 และภาพที่ 5.20

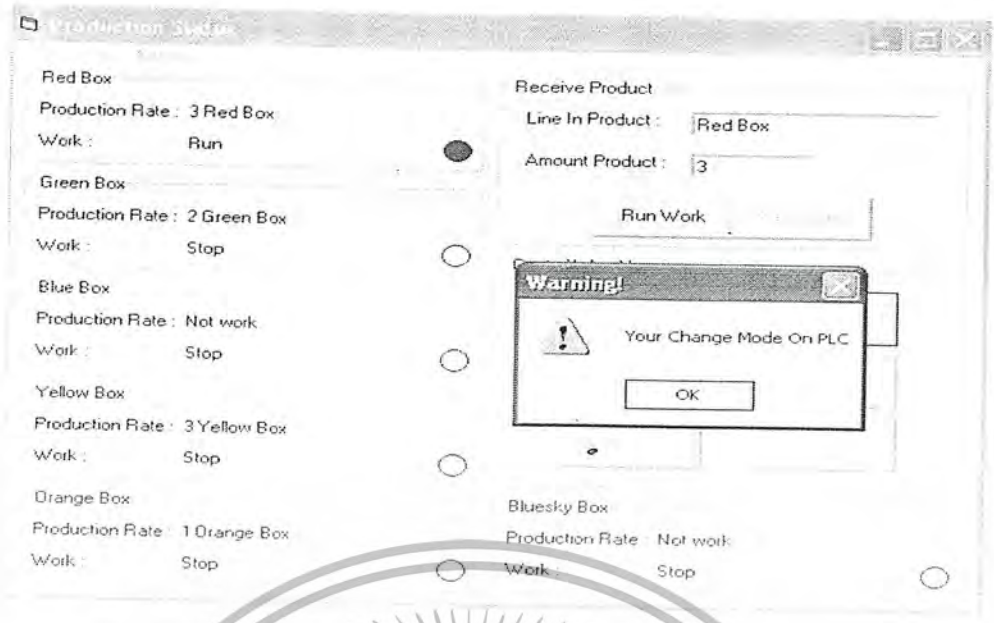
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้




**ภาพที่ 5.20** แสดงการเลือกการผลิตแบบ Auto

**ภาพที่ 5.21** แสดงการเลือกการผลิตแบบ Manual

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้




**ภาพที่ 5.22** แสดงรูปที่เกิดการฟ้องให้เราไปปรับ Switch ที่ตัวสายการผลิต

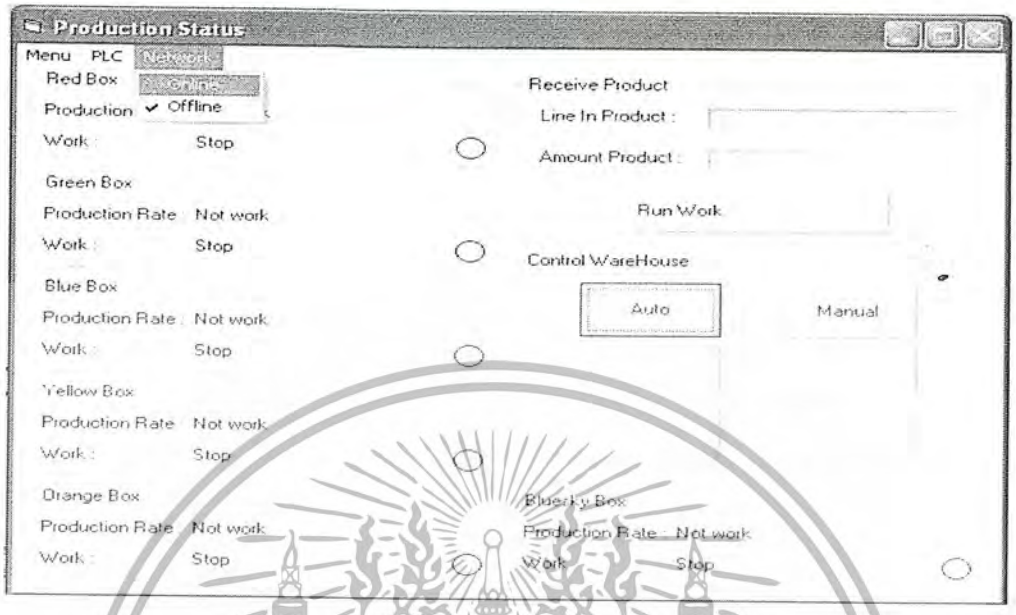
**ภาพที่ 5.23** แสดงหน้าต่างการทำงานในโหมด Manual

8. เมื่อมีการฟ้องของโปรแกรมดังรูปที่5.21 หลังจากการเลือกโหมดการทำงาน ให้เราทำการปรับ Switch ที่ชุดสายการผลิตหลักใหม่

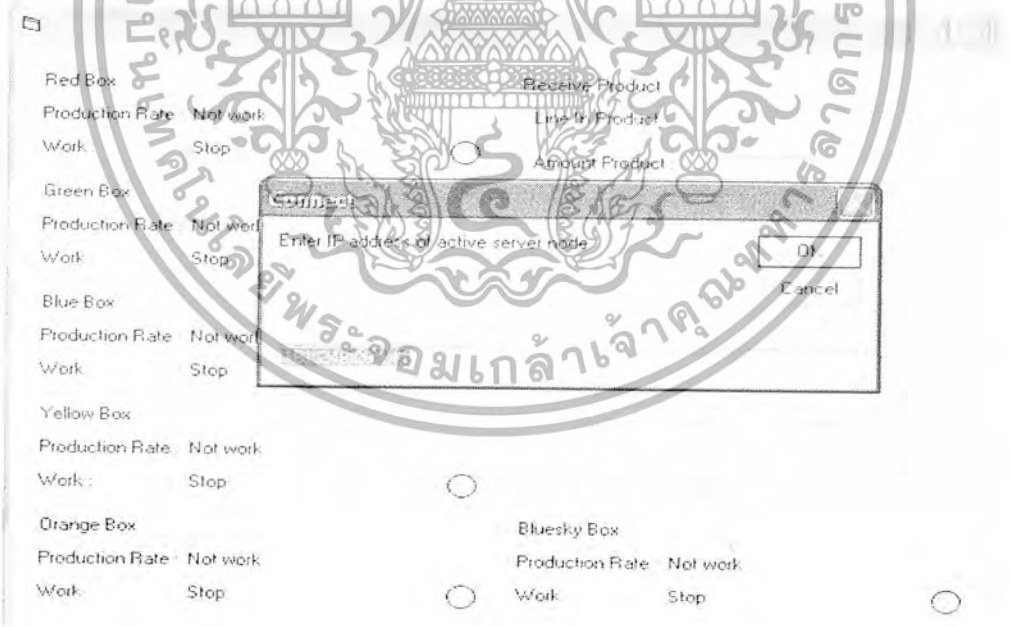
9. และในโปรแกรมนี้ยังสามารถสอบถามสถานะของการผลิตของแต่ระยะ ส่วนด้วยตัวโปรแกรมที่เพิ่มเข้าไปใช้ในการพูดคุย โดยการ เข้าไปที่ Network และไปที่ Online ดังภาพที่ 5.23

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หลังจากนั้นใส่ IP ของเครื่องที่เราต้องการจะติดต่อดังภาพที่ 5.24 และเมื่อใส่ IPเสร็จเรียบร้อยแล้ว จะปรากฏหน้าจอดังภาพที่ 5.25 เมื่อปรากฏหน้าจอดังภาพที่ 5.25 แล้วก็สามารถพูดคุยกับอีกฝ่ายได้

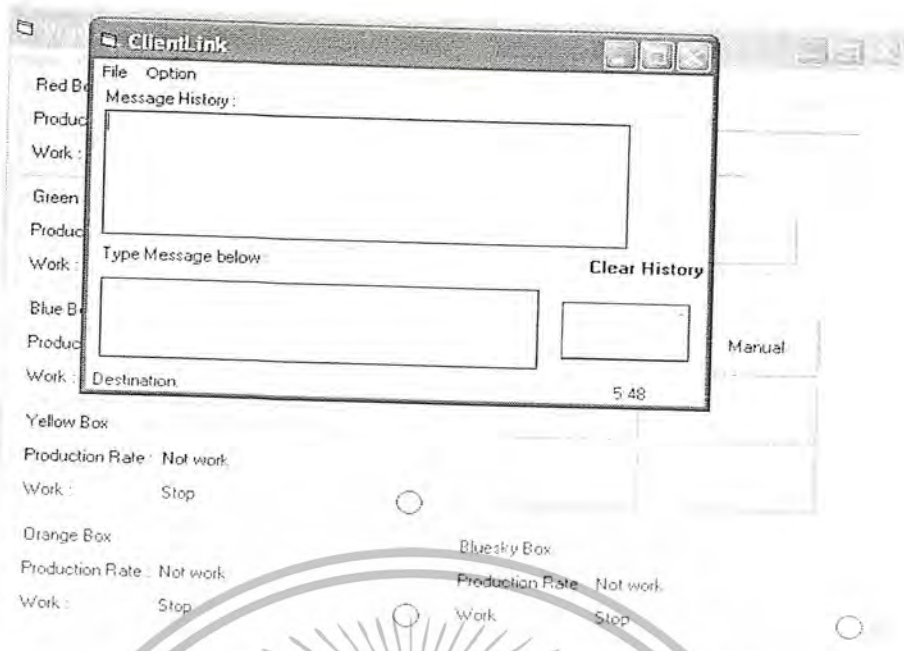



ภาพที่ 5.24 แสดงวิธีเข้าโปรแกรม




ภาพที่ 5.25 แสดงการใส่ค่า IP

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้




**ภาพที่ 5.26** แสดงหน้าจอที่ใส่ค่า IP เสร็จพร้อมพูดคุยได้

### 5.3.2 สรุปผลการทดลองในส่วนสายการผลิตหลัก

เมื่อเริ่มการทำงานในกระบวนการแรกนั้นจะไม่มีอุปกรณ์ที่จะประกอบจะรอจนกว่าอุปกรณ์มาถึงและจะทำการประกอบในทันที และในการประกอบนี้ จะมีอยู่สองโหมดการทำงานให้เลือก คือ การทำงานในโหมด Auto และการทำงานในโหมด Manual ซึ่งในโหมด Auto นั้นเครื่องจะทำงานเองทั้งหมด ในส่วนของโหมด Manual นั้นเราจะต้องกดปุ่มสั่งการทำงานทุกขั้นตอน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# บทที่6
# สรุปโครงงาน

## 6.1 สรุปโครงงาน

ปริญญานิพนธ์ฉบับนี้ เป็นการศึกษา และนำอุปกรณ์ที่มีอยู่แล้วภายในห้องทดลองมาใช้งานร่วมกันเป็นระบบ Just In Time ซึ่งเป็นการรวมระหว่าง Ware House และสายพานลำเลียงการผลิต การผลิต เข้าด้วยกันเป็นระบบ โดยการควบคุมจะผ่านระบบเครือข่ายคอมพิวเตอร์ (Network) สำหรับระบบ Just In Time ที่กล่าวถึงนี้ในการทดลองนี้ ได้มีการจำลองระบบการสั่งวัตถุผ่าน Network ซึ่งจำนวนวัตถุและชนิดของวัตถุที่สั่งจะเป็นการเก็บไว้ในระบบฐานข้อมูล เมื่อมีการสั่งผลิตภัณฑ์เข้ามา โปรแกรมก็จะทำการดึงค่าจากฐานข้อมูลแล้ว จึงสั่งให้สายพานลำเลียงวัตถุเข้ามาผ่านตัวเซนเซอร์และใช้กล้องดิจิตอลเป็นตัวตรวจจับสีของวัตถุ เพื่อ นำชนิดของวัตถุ (สีของวัตถุ) ที่สั่งมาเปรียบเทียบกับข้อมูลที่อยู่ในฐานข้อมูล ถ้าวัตถุที่สั่งตรงกัน ก็จะรับของชิ้นนั้นเข้ามาในระบบ แต่ถ้าวัตถุไม่ตรงกัน ก็จะคัดทิ้งออกไป ซึ่งระบบนี้จะได้ของครบตามจำนวนที่สั่งไป พร้อมกับเวลาในการรับของ ก็จะเป็นไปตามที่เรากำหนด

## 6.2 ปัญหาที่พบ

เนื่องจากนำอุปกรณ์ที่มีอยู่เดิมมาประยุกต์ใช้คือ Ware House ในส่วนนี้จะมีปัญหาที่เกิดจากการบิดเบี้ยวของช่องใส่ของ ทำให้เวลาทดลองมีการบิดพลาด ส่งผลให้อุปกรณ์เกิดความเสียหาย ส่วนการเขียนควบคุมสายพานที่คัดแยกสีของวัตถุ ด้วยระบบวิเคราะห์สีแบบ Fuzzy ทำให้แสงมีผลกับการวิเคราะห์สีอย่างมาก ถ้าหากแสงไม่พอ จะทำให้การวิเคราะห์สีผิดพลาด

## 6.3 ข้อเสนอแนะ

ระบบ Just In Time เป็นระบบที่ช่วยในการลดต้นทุนในการผลิตโดยจะลดการทำงานที่สูนย์เปล่าออกไปอย่างเช่น พนักงานหนึ่งคนสามารถควบคุมได้หลายเครื่องจักร ลดค่าใช้จ่ายในการเก็บรักษาวัสดุ เป็นต้น แต่ในการนำไปปฏิบัติจริงไม่สามารถนำไปใช้ได้ทั้งกระบวนการเนื่องจากเมื่อมีเครื่องจักรเสียหายจะทำให้หยุดการผลิตส่งผลให้มีความล่าช้าในการส่งมอบสินค้าให้แก่ลูกค้า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# บรรณานุกรม


อ. ทวีพล ซื่อสัตย์, **ระบบโรงงานอัตโนมัติ**,ภาควิชาวิศวกรรมการวัดคุม คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง.

Operation Manual, **Programmable Controller**, C200HX/C200HG/200HE, Omron, 1996

Operation Manual, **Sysmac C200H-CT001-V1/CT002**, High Speed Counter Unit, Omron, 2000

สัจจะ จรัสรุ่งรวีวร, **คู่มือเขียนโปรแกรมและการใช้งาน Visual Basic 6.0**, อินโฟเพรส, กรุงเทพฯ,พิมพ์ครั้งที่2, 2544, 392 หน้า

สัจจะ จรัสรุ่งรวีวร, **คู่มือการสร้างแอบพลิเคชั่นด้วย Visual Basic 6.0 ฉบับสมบูรณ์**, อินโฟเพรส, กรุงเทพฯ,พิมพ์ครั้งที่ 1, 422 หน้า

วิฑูรย์ สิมะโชคดี, **ระบบคัมบัง**, กระทรวงอุตสาหกรรม, กรุงเทพฯ, พิมพ์ครั้งที่ 1, 2533, 174 หน้า

สุรชัย ธรรมทวีธีกุล, **ระบบการผลิต JIT จากหลักการสู่ภาคปฏิบัติจริง**, ซีเอ็ดยูเคชั่น, กรุงเทพฯ, พิมพ์ครั้งที่1, 170 หน้า

ปริญญานิพนธ์, **การออกแบบระบบจัดเก็บผลิตภัณฑ์อัตโนมัติผ่านเครือข่ายอินเตอร์เน็ต**, สาขาวิชาวิศวกรรมการวัดคุม ภาควิชาวิศวกรรมการวัดคุม คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปริญญานิพนธ์, **การตรวจจับสีวัตถุด้วยวิธีการของฟัซซีล**, สาขาวิชาวิศวกรรมการวัดคุม ภาควิชาวิศวกรรมการวัดคุม คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

อภิชาต ภู่พลับ, **เริ่มต้นการเขียนโปรแกรมติดต่อและควบคุมฮาร์ดแวร์ด้วย Visual Basic 6.0**, อินโฟ เพรส, นนทบุรี, พิมพ์ครั้งที่ 1,2546, 240 หน้า



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# ภาคผนวก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# Data Sheet

# RS Omron CK range programmable logic controllers (PLCs)

The Omron CK PLCs are a high specification, cost effective, compact range of programmable logic controllers (PLCs) which enable very flexible systems to be configured due to their wide array of CPU and extension I/O modules. Using a 'building block' principle, the user can easily design his initial system as well as being able to add any I/O block to any existing CPU or extension modules in any order, up to 148 I/O. The programming language adopted is an easy to use standard ladder diagram which includes instructions and features normally associated with much larger controllers and is common to the CH range and other C-series PLCs (not supplied by RS).

## Main system features:

- Very flexible I/O system structure
- All CPUs have in-built 2kHz counters
- Easy to program with ladder diagram software
- Analogue input and output capability
- CPU's source supply for inputs
- Programmable with PC based software
- NPN and PNP inputs as standard
- Individual outputs on a CPU or expansion can be changed from relay to transistor or triac as applications require.

## Section index


1. What is a PLC?
2. System configuration
3. How to choose CK range
   - 3.1 System value (SV)
   - 3.2 Range overview
   - 3.3 Input and output specifications
4. Main and expansion units
   - 4.1 Main units
   - 4.2 Expansion units
     - 4.2.1 Digital input/output units
     - 4.2.2 Analogue units
     - 4.2.3 Analogue timer unit
5. Programmers
   - 5.1 'LSS' programming documentation and monitoring software
   - 5.2 Programming console
   - 5.3 Data access console
6. Accessories
   - 6.1 EPROM writer
   - 6.2 Program memory chips
   - 6.3 Communications modules
   - 6.4 Replacement outputs
   - 6.5 Input simulator and training system
   - 6.6 CK operation manuals
7. Dimensions
8. Programming
   - 8.1 Ladder diagram programming
   - 8.2 Symbols and what they mean
   - 8.3 Example circuits
   - 8.4 High speed drum (counter)
   - 8.5 Reversible drum counter
   - 8.6 Interlocks and jumps
   - 8.7 Shift register
   - 8.8 Subroutines
   - 8.9 Step sequence
   - 8.10 I/O refresh
   - 8.11 Failure alarm
9. Programming instructions
10. Glossary

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 1. What is a PLC?

A programmable logic controller is a system for controlling industrial and production processes, electromechanical equipment and automated assembly operations. This control function is performed by continuously monitoring the states of all the input devices connected to the controller (eg. proximity switches and mechanical contacts). Following the user instructions stored in the controller (known as the program), the states of all its output devices are set as appropriate (eg. relays, solid state relays or transistors). These output elements can in turn control the operation of an external system.

Due to the simplicity of entering and modifying the programmed instructions to suit the requirements of the process under control, the PLC is a truly versatile and flexible device that can be employed easily and efficiently to repeatedly control tasks that vary in nature and complexity.

A schematic diagram of the basic PLC control system is shown in Figure 1.

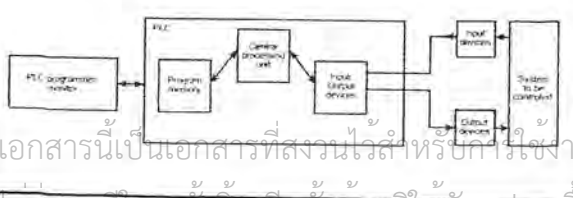
In this representation the central processing unit controls the overall operation of the system. Input devices may be switches, relay contacts, timers, solid state switches, proximity and limit switches, sensors, analogue to digital converters or other electronic circuits. The output devices may be external relays, heating elements, lights, alarms, sub-assemblies, electronic and electrical circuits, motors or other electromechanical devices.

To illustrate the advantages of using a PLC over a traditional electromechanical system consider a control system with 20 input/output points. This assembly could comprise 60-80 relays, some counters and/or timers and a great deal of wiring. This assembly would be cumbersome with a power consumption of 30-40VA. A considerable time would be required to design, build, test and commission the assembly and once it is in full working order any desired modifications, even of a minor nature, could require major hardware changes.

Alternatively a PLC can be used. A programmer is used to set up the control sequence as a software program with relays, counters and timers together with input and output devices. Programming and testing may be done off site before the PLC is connected up to actual input and output devices.

Programming will typically take 1-2 hours, any alterations to the control function of the system can be simply implemented within minutes by modifying the stored program. Other advantages over more conventional systems include compact size, low power consumption, fast response, versatility and lower overall cost. As the input/output requirements of a system increase, the advantage of using a PLC becomes much more significant due to the complexity of the equivalent hardwired system and its labour and material costs.

Figure 1 Schematic diagram



## 2. System configuration

A system is principally made up of the following parts:

- A 110/240Vac powered main unit which contains Central Processor Unit, program memory and a fixed number of inputs and outputs
- Expansion units can be added as required up to a system value of 5 (see below)
- Programming may be performed by LSS, PC based software or clip on programmer.

Unit type programmable controllers are popular because of their simple, all-in-one construction, but suffer from some limitations on I/O expansion. Modular types, on the other hand, allow flexibility but are cumbersome to handle and can involve a lengthy and tiresome selection process.

The CK range eliminates these problems by combining the simplicity of the unit type with the expandability of the modular type.

The availability of a wide range of model types and I/O sizes greatly facilitates the selection of the most appropriate, most economical configuration for a particular application.

## 3. How to choose CK range

1. Assess the requirement for input and output points in the chosen application and whether outputs should be standard relay output, solid state relay (triac) or transistor output.
2. Choose main unit and expansion units to optimise the number of inputs and outputs. However, extra redundant I/O are often included for system additions and alterations. Any combination of any main unit, digital and analogue expansion units can be used up to a System Value of 5 (see below). Relay supplied units can be changed to transistor, triac or any combination using the replacement outputs available.
3. Choose the most suitable programming method, clip-on programmer, advanced hand held programmer or LSS programming and monitoring PC based software.
4. Decide if the program should be stored on EPROM memory. A dedicated EPPOM writer is available.

   Note: Program storage is recommended, but is not essential.
5. Decide if information needs to be displayed via the Omron data access module or a message display (see current RS Catalogue).

   A separate adaptor is required when using either the data access module or advanced hand held programmer.
6. Refer to CK Manuals for a fuller explanation of CK devices and programming.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.1 System Value (SV)

Having established the I/O requirements, it is usual to choose a main CPU unit with an I/O count which is closest to the desired number – there is a choice of 4 units – 60, 40, 28 and 20, each available with either transistor or relay outputs (relay output versions are all plug-in and allow the mixture of output types as mentioned earlier). If the application needs more I/O than is available on the CPU main unit, any expansion module can be added until the requirement is met.

The number of expansion units which can be added to a system is controlled by this simple rule:

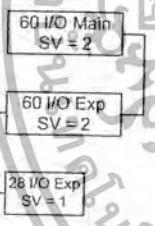
Each CPU and expansion unit has a 'System Value' thus:

| CPU | System value |
|---|---|
| C60K | 2 |
| C40K | 2 |
| C28K | 1 |
| C20K | 1 |
| **Expansion** | |
| C60P | 2 |
| C40P | 2 |
| C28P | 1 |
| C20P | 1 |
| C16P | 1 |
| C4K | 1 |

When connected, the System Values of all the modules added together must not exceed '5' – this is the only restriction.

### Examples

1.

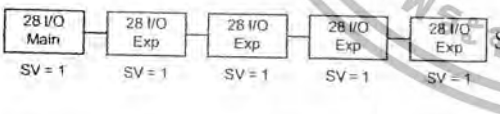



Total
SV = 5

2.

SV = 5

3.




SV = 5

This last example would not normally occur at initial design stage, but could be the result of an original system being added to over a period of time due to modifications; however caused, it illustrates the flexibility of building a CK system.

Having decided upon the I/O configuration, the next step is to choose the necessary programming and peripheral equipment; this is outlined later in this section.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Table 3 General ratings

| | |
|---|---|
| Supply voltage | 100 to 240Vac 50/60Hz |
| Operating voltage range | 85 to 264Vac |
| Power consumption | 60VA max. |
| 24Vdc output | 0.3A 24Vdc ±10% (use as dc power supply) |
| Insulation resistance | 10MΩ min. (at 500Vdc) between ac terminals and housing |
| Dielectric strength | 2000Vac 50/60Hz for 1 minute (between ac terminals and housing)<br>500vac 50/60Hz for 1 minute (between dc terminals and housing) |
| Noise immunity | 1000 Vp-p , pulse-width: 100ns to 1μs , rise time 1ns |
| Vibration | 10 to 35Hz, 2mm double amplitude, in X, Y, and Z directions; 2 hours each (when mounted on a DIN rail: 16.7Hz, 1mm double amplitude, in X, Y, and Z directions, 1 hour each) |
| Shock | 10G in X, Y, and Z directions, 3 times each |
| Ambient temperature | Operating: 0° to 55°C<br>Storage: –20° to 65°C |
| Humidity | 35% to 85% RH (without compensation) |
| Grounding | Less than 100Ω |
| Structure | IEC IP-30 (mounted in a panel) |
| Weight CPUs<br>Expan. I/O units | C20K, C28K: 1.9kg max.; C40K: 2.2kg max.; C60K: 2.6kg max.<br>C20P, C28P: 1.7kg max.; C40P: 2.0kg max.; C60P: 2.4kg max. |
| Dimensions (CPUs and Expan. I/O units) | C20K, C28K: 250(w) × 110(h) × 100(d); C40K: 300(w) × 110(h) × 100(d);<br>C60K: 350(w) × 140(h) × 100(d) |

Table 4 CPU characteristics

| | |
|---|---|
| Main control elements | MPU, C-MOS, LS-TTL |
| Programming method | Ladder diagram |
| Instruction length | 1 address/instruction,6 bytes/instruction |
| Number of instructions | 49 |
| Execution time | 10μs/instruction (average) |
| Memory capacity | 1194 addresses |
| IR bits | 136 (1000 to 1807)<br>1804 to 1806 are reserved for RDM: FUN60 if it is used.<br>1807 is reserved as software reset input for HDM: FUN 98 if it is used. |
| SR bits | 16 (1808 to 1907)<br>Normally-ON, normally-OFF, battery failure, initial scan ON, 0.1-s pulse, 0.2-s pulse, 1.0-s pulse etc. |
| HR bits | 160 (HR 000 to 915) |
| TM bits | 8 (TR0 to 7) |
| DM words | 64 (DM words 00 to 63)<br>DM words 00 to 31 are reserved as upper and lower limit setting areas for RDM: FUN 60 if it is used. DM words 32 to 63 are reserved as upper and lower limit setting areas for HDM: FUN 61 if it is used. |
| Timer/Counters | 48 (total of TIMs, CNTs, and CNTRs)<br>TIM 00 to 47 (0 to 999.9 s)<br>TIMH 00 to 47 (0 to 99.99 s)<br>CNT 00 to 47 (0 to 9999 counts)<br>CNTR 00 to 47 (0 to 9999 counts)<br>CNT 46 serves as RDM: FUN 60. Likewise CNT 47 serves as HDM: FUN 61. When these instructions are not used, CNT 46 and 47 can be used for other purposes. |
| High-speed counter | Count input: 0000<br>Hardware reset input: 0001<br>Software inset: 1807<br>Maximum response frequency: 2kHz<br>Preset count range: 0000 to 9999<br>Number of outputs: 16 |
| Reversible drum counter | Reset: 1804<br>Count input: 1805<br>Reverse input: 1806<br>Preset count range: 0000 to 9999<br>Number of outputs: 16 |
| Memory protection | Status of HR bits, present value of counters, and contents of DM bits are retained during power failure. |
| Battery life | 5 years at 25°C<br>Battery life is shortened at temperatures higher than 25°C. Replace battery with new one within 1 week when ALARM indicator blinks. |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 3.2 Range overview

Table 1 **RS** Omron CK PLC range overview

| Product | RS stock no. | Omron no. | Description and features | System value |
|---|---|---|---|---|
| Main units | 731-704 | C20K CDRA | CPU with 240/110Vac power supply, 12 24Vdc inputs, 8 relay contact outputs | 1 |
| | 731-710 | C28K CDRA | CPU with 240/110Vac power supply, 16 24Vdc inputs, 12 relay contact outputs | 1 |
| | 731-726 | C40K CDRA | CPU with 240/110Vac power supply, 24 24Vdc inputs, 16 relay contact outputs | 2 |
| | 731-732 | C60K CDRA | CPU with 240/110Vac power supply, 32 24Vdc inputs, 28 relay contact outputs | 2 |
| | 731-748 | C20K CDT1A | CPU with 240/110Vac power supply, 12 24Vdc inputs, 8 transistor outputs | 1 |
| | 731-754 | C28K CDT1A | CPU with 240/110Vac power supply, 16 24Vdc inputs, 12 transistor outputs | 1 |
| | 731-760 | C40K CDT1A | CPU with 240/110Vac power supply, 24 24Vdc inputs, 16 transistor outputs | 2 |
| | 731-776 | C60K CDT1A | CPU with 240/110Vac power supply, 33 24Vdc inputs, 28 transistor outputs | 2 |
| Expansion units | 731-805 | C20P EDRA | Expansion with 240/110Vac power supply, 12 24Vdc inputs, 8 relay contact outputs | 1 |
| | 731-811 | C28P EDRA | Expansion with 240/110Vac power supply, 16 24Vdc inputs, 12 relay contact outputs | 1 |
| | 731-827 | C40P EDRA | Expansion with 240/110Vac power supply, 24 24Vdc inputs, 16 relay contact outputs | 2 |
| | 731-833 | C60P EDRA | Expansion with 240/110Vac power supply, 32 24Vdc inputs, 28 relay contact outputs | 2 |
| | 731-855 | C20P EDT1A | Expansion with 240/110Vac power supply, 12 24Vdc inputs, 8 transistor outputs | 1 |
| | 731-861 | C28P EDT1A | Expansion with 240/110Vac power supply, 16 24Vdc inputs, 12 transistor outputs | 1 |
| | 731-877 | C40P EDT1A | Expansion with 240/110Vac power supply, 24 24Vdc inputs, 16 transistor outputs | 2 |
| | 731-883 | C60P EDT1A | Expansion with 240/110Vac power supply, 32 24Vdc inputs, 28 transistor outputs | 2 |
| | 731-798 | C16P ORA | Expansion with 240/110Vac power supply, 16 relay contact outputs | 1 |
| | 731-782 | C16P 1DA | Expansion with 240/110Vac power supply, 16 24Vdc inputs | 1 |
| | 731-849 | C16P OT1A | Expansion with 240/110Vac power supply, 16 transistor outputs | 1 |
| | 731-906 | C4K OR2 | Expansion with 4 relay contact outputs | 1 |
| | 731-912 | C4K OT2 | Expansion with 4 transistor outputs | 1 |
| | 731-899 | C4K 1D | Expansion with 4 24Vdc inputs | 1 |
| Analogue units | 731-934 | C4K AD | Expansion with 4 analogue inputs, 1-5V and 4-20mA | 1 |
| | 731-928 | C1K AD | Expansion with 1 analogue input, 1-5V and 4-20mA | 1 |
| | 731-940 | C1K DA | Expansion with 1 analogue output, 1-5V and 4-20mA | 1 |
| | 731-956 | C4K TM | Analogue timer module | 1 |
| | 731-962 | C4K CN223 | Connector for C4K TM for external potentiometers | – |
| Programming equipment | 734-488 | 3G2A6 PRO15E | Programming console for CK range | – |
| | 731-653 | C200H PRO27ENL | Advanced programming console for CK and CH ranges | – |
| | 731-675 | C200H CN222 | 2m connecting cable for C200H DAC01 and C200H PRO27 | – |
| | 734-551 | 3G2A5 AP003 | Adaptor for connecting C200H DAC01 or C200H PRO27 to CK PLCs | – |
| | 731-647 | LSS 3.5/5.25 | PC based programming software for CK, CH and all other C series PLCs | – |
| Communications modules | 731-978 | 3G2C7 LK201EV1TWN | RS-232 communications module for CK range | – |
| | 731-984 | 3G2C7 LK202EV1TWN | RS-422 communications module for CK range | – |
| Accessories | 734-539 | C500 PRW06 | EPROM writer for CK range | – |
| | 734-545 | ROMH | EPROM 2764 | – |
| | 731-697 | C200H DAC01ENL | Data access console | – |
| | 734-494 | G3SDZ01PDPUS24DC | Transistor module (NPN) for CK and CH ranges | – |
| | 734-501 | G3SDZ01PPE24DC | Transistor module (PNP) for CK and CH ranges | – |
| | 734-517 | G3S201PLPDUS24DC | Triac module for CK and CH | – |
| | 734-567 | C20K ETL01 | Simulator switches for C20K, C28K, C20H and C28H | – |
| | 731-619 | EEPROMH | EEPROM 28C64 | – |
| Manuals | 734-573 | CKMANUAL | CK CPU manual and analogue modules manual | – |

Table 2 Comparison of CK and CH CPU characteristics

| | CK | CH |
|---|---|---|
| Supply voltage | 100 to 240Vac −15 to +10% 50/60Hz | 24Vdc −15 to +10% |
| dc output | 0.3A 24Vdc ±10% | – |
| Operative ambient temperature | 0° to 55°C | 0° to 55°C |
| I/O control method | Cyclical scan | Cyclical scan |
| Programming method | Ladder diagram | Ladder diagram |
| Instruction length | 1 address/ instruction, 6 bytes/instructions | 1 address/ instruction, 1 to 4 words/instruction |
| Number of instructions | 49 | 142 (12 basic 130 special) |
| Execution time | 10μs/instruction (average) | 0.75 to 2.25μs |
| Memory capacity | 1194 address (≈ 3K bytes) | 2878 words |
| Memory type | RAM/EPROM | RAM/EPROM/ EEPROM |

| | CK | CH |
|---|---|---|
| Internal relays | 136 | 3472 |
| Special relays | 16 | 136 |
| Holding relays (battery backed) | 160 | 1600 |
| Temporary relays | 8 | 8 |
| Data memory (registers) | 464 | 1000 read/write 1000 processor generated |
| Auxiliary relays | – | 448 |
| Link relays | – | 1024 |
| Timers/Counters | 48 | 512 |
| High speed counter | 2kHz | 2kHz |
| Battery life | 5 years at 25°C | 5 years at 25°C |
| Features | Self diagnostics program check | Self diagnostics program check |
| Maximum number of inputs/outputs | 148 | 240 |

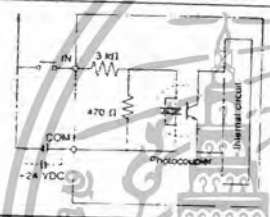
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Table 4 CPU characteristics (continued)

| | |
|---|---|
| Self-diagnostic features | CPU failure (watchdog timer)<br>Memory failure<br>I/O bus failure<br>Battery failure, etc. |
| Program check | Program check (executed on start of RUN operation.)<br>END instruction missing<br>JMP-JME error<br>Coil duplication<br>Circuit error<br>DIFU/DIFD over error<br>IL/ILC error |

## 3.3 Input and output specifications

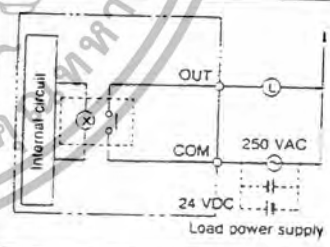
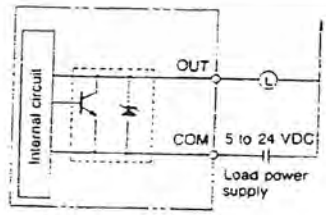
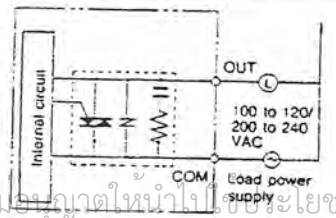
Table 5 Input specifications

| | dc input (photocoupler-isolated) |
|---|---|
| Supply voltage | 24Vdc ±10% |
| Input impedance | 3kΩ |
| ON voltage | 7mA at 24Vdc |
| ON voltage | 15Vdc min. |
| OFF voltage | 5Vdc max. |
| ON delay time | 2.5ms max. (input 0000 and 0001: 0.15ms) |
| OFF delay time | 2.5ms max. (input 0000 and 0001: 0.15ms) |
| Circuit configuration |  |



Note: The 24Vdc power source can be connected to either the positive or the negative terminal. Therefore, both PNP input (negative common) and NPN input (positive common) can be used.

Table 6 Output specifications

| | ON-delay | OFF-delay | Switching capacity Max. | Switching capacity Min. | Circuit configuration |
|---|---|---|---|---|---|
| Relay (photocoupler-isolated) | 15ms max. | 15ms max. | 2A at 250Vac<br>2A at 24Vdc<br>(cosφ = 1)<br>0.5A at 250Vac<br>(cosφ = 0.4)<br>4/4 points | 10mA at 5Vdc |  |
| Transistor* (photocoupler-isolated) | 1.5ms max. | 1.5ms max. | 1 A/point at 5 to 24Vdc, 1.6 to 4 A/4 points | 10mA at 5Vdc, saturation voltage: 1.5V max. |  |
| Triac** (photocoupler-isolated) | 1.5ms max. | ½ of load frequency +1ms max. | 1 A/point at 85 to 250Vac, 1.6 to 4 A/4 points | 10mA at 100Vac, 20mA at 200Vac |  |





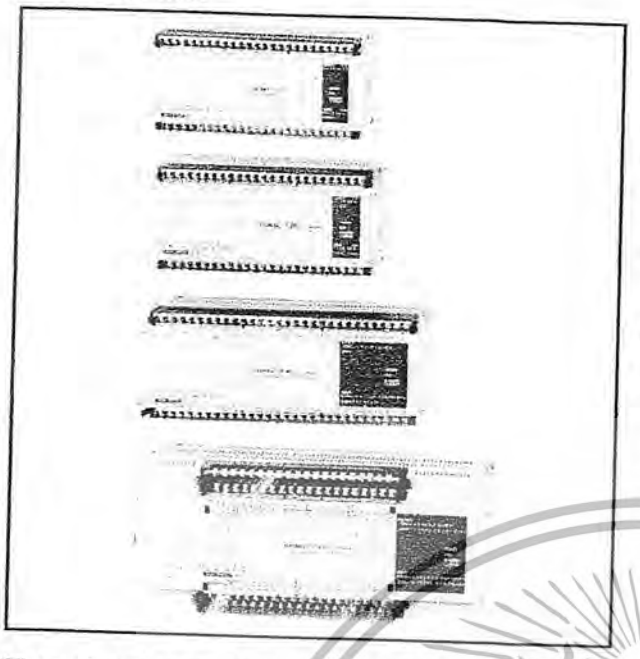


*Leakage current: 100μA max. at 24Vdc. Residual voltage: 1.2V max. **Leakage current: 20mA max. at 100Vac, 5mA max. at 200Vac. Residual voltage: 1.5V max.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงลิขสิทธิ์ของมหาวิทยาลัยทุกครั้งที่มีการนำไปใช้

# 4. Main and expansion units

## 4.1 Main unit



The main units come in a choice of 60, 40, 28 and 20 I/O versions and are all complete with the same amount of RAM – 1194 actual instructions – this means that there are guaranteed 1194 instructions regardless of individual instructions complexity. For example a normally open contact will only occupy one memory location, as will a shift-register.

Each CPU is supplied with 240/110Vac and will in turn generate a 24Vdc power supply for use with input devices; these input devices can be voltage free contacts, PNP or NPN inputs. The outputs can be relay or transistor or, by using the unique structure of the output circuits, can be a mixture of relay contact, NPN transistor, PNP transistor or triac – this is because on the relay output version, all outputs are 'plug-in' and the arrangement of the common terminals means that such outputs can be mixed (refer to replacement relay section).

The CK CPUs have been designed so that the maintainable parts (fuses, outputs and battery if using RAM) are easily accessed without any interference with non-maintainable parts such as the processor and peripheral circuits. Removable terminal strips allow quick and easy access, with the maintainable parts being visible by removing the main units' cover (which is mounted with 4 sprung retained screws). On the relay outputs versions, a tool is mounted inside the main unit for allowing removal of the 'plug-in' output modules. All maintenance and changing of outputs is accomplished without any hindrance to the main circuits which are securely protected and shielded in the lid of the controller. Access for EPROM changing is through a small cover on the front of the CPU and does not need the main cover to be removed.

Connection of other hardware devices to the main unit is covered in the following sections of this data sheet, as is the full software capability of the processors.

### Features

- 110/240Vac power supply which sources 24Vdc for inputs
- NPN or PNP inputs accepted as standard
- Outputs are 'plug-in' on the relay output version and can be individually replaced with transistors (NPN or PNP) and/or triacs; thus different output types can be configured in the same main unit
- 1194 RAM memory program addresses with five year battery back-up
- Removable terminal strips
- Analogue input and output modules available for all CPUs
- All modules are DIN rail or screw fixing (screws supplied)
- LED indication of inputs, outputs and CPU status
- Changeable parts (relays, batteries, fuses) are easily accessible
- Built-in 2kHz high speed counters on all models with hardware and software resets
- Any expansion module can be connected to any CPU
- EPROM socket is under the front of the unit for easy access
- 140 input/output points maximum capability
- 48 timers and counters including timers settable to 1/100th second
- 64 16 bit data registers
- Maths instructions including +, −, ÷, × and data manipulation instructions including Move, Compare, Word shift
- Programming features include:
  - Input/Output refresh
  - Interlock and jump (including jump to number)
  - Step sequencing
  - Shift register
  - High speed and reversible drum counters
  - Differential instructions
  - Binary and BCD data manipulation
  - Subroutines
- Full diagnostic functions and flags
- All CPUs have comprehensive communications capability whereby they can be connected to a host computer on either RS-232 or RS-485 for programming and/or graphics monitoring. This communication is performed transparent to the main user program
- Programs can be written and stored by Omron PC based software package (Ladder Support Software LSS) and/or by a programming console
- Programs can be transferred to EPROM via an Omron EPROM writer RS stock no. 734-539 or via LSS to an RS-232 writer
- Timers, counters and Data registers are accessible during Run operation
- Simulator switches available for inputs.

See Overview table for equivalent Omron part no.

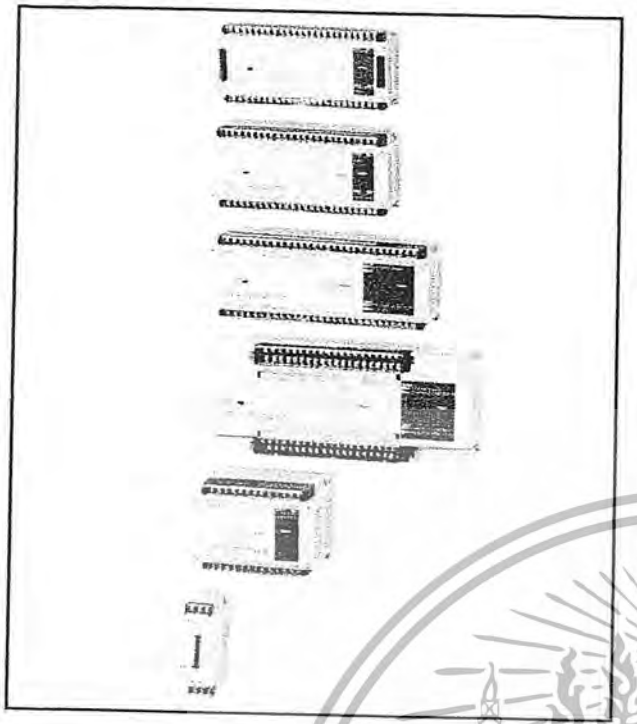
Supplied with comprehensive instruction manual

| **RS** stock no. | Type |
|---|---|
| 731-704 | 20 I/O relay |
| 731-710 | 28 I/O relay |
| 731-726 | 40 I/O relay |
| 731-732 | 60 I/O relay |
| 731-748 | 20 I/O transistor |
| 731-754 | 28 I/O transistor |
| 731-760 | 40 I/O transistor |
| 731-776 | 60 I/O transistor |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 4.2 Expansion units

### 4.2.1 Digital input/output units



Expansion units provide additional I/0 points to those provided by the CPU. As described in the selection guide, by adhering to a simple rule, any expansion can be connected to any CPU in any order. As with the CPUs, all expansions have removable terminal strips (with the exception of 4 way modules) and selectable outputs – the flexibility of the CK range is therefore maintained throughout the system.

When selecting expansion units, there is no need to allow for connecting cables as these are included with the module itself. With the exception of the 4 way modules, all expansions can be connected either to the right, to the left, above or below a previous CPU or expansion unit – this allows for very flexible panel layouts and can be especially useful where expansions are being added to a system where space may not originally have been reserved alongside existing units.

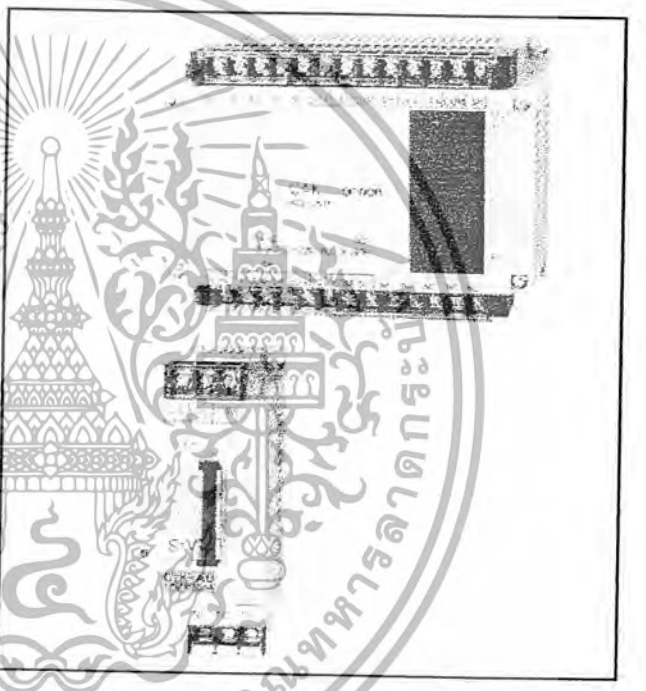
**Features**

- 4 way and 16 way input and output units, 20, 28, 40 and 60 way mixed input/output units
- 110/240Vac supply which sources 24Vdc for inputs (except 4 way modules)
- NPN or PNP inputs accepted as standard
- 'Plug-in' outputs enabling relays to be individually changed for transistors or triacs within the same module. (Relay output supplied versions only)
- Expansion modules can be mounted alongside (left or right) or above or below another expansion or CPU
- All connecting cables are included
- LED indication of all I/O and CPU status
- All units are DIN rail mountable
- Simulator switches available
- Removable terminal strips on all models excluding 4 way.

See Overview table for equivalent Omron part no.

| **RS** stock no. | Type |
|---|---|
| 731-805 | 20 I/O relay |
| 731-811 | 28 I/O relay |
| 731-827 | 40 I/O relay |
| 731-833 | 60 I/O relay |
| 731-855 | 20 I/O transistor |
| 731-861 | 28 I/O transistor |
| 731-877 | 40 I/O transistor |
| 731-883 | 60 I/O transistor |
| 731-782 | 16 I/P |
| 731-798 | 16 O/P relay |
| 731-849 | 16 O/P transistor |
| 731-899 | 4 I/P |
| 731-906 | 4 O/P relay |
| 731-912 | 4 O/P transistor |

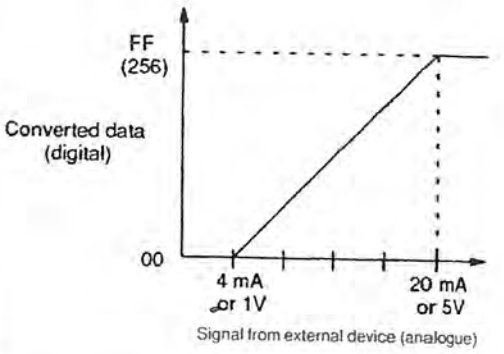
### 4.2.2 Analogue units



Analogue units are mounted close to the proceeding unit (they are not permitted to be connected via longer cables as in digital expansion units). When connected to a system, they are simply assigned the next word available in the I/O Map. For analogue inputs, the value which appears as an analogue voltage or current to the terminals of the module is presented to the word to which the module has been assigned as a Binary value (see the diagram); there is an instruction in the CPU which allows the user to convert this binary value to Binary Coded Decimal (BCD) so that a signal varying from 4 to 20mA for example, would be converted to a number between 1 and 256 by the CPU – this number being available to the user throughout the program.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

The analogue input signal is converted in a straight-line relationship to binary values between 00000000 and 11111111 (00 and FF in hexadecimal). These are related to the input voltage and current ranges as shown in the following graph.




Analogue outputs work in exactly the opposite way, a BCD value between 1 and 256 is converted to binary by the CPU and moved to the appropriate word which has been assigned to the module – this is then converted to an analogue signal between 4 and 20mA or 1 to 5V.

## Features

- 4 channel analogue input, 1 channel analogue input and 1 channel analogue output
- Connecting cables included
- BCD to Binary conversion instruction in CPU simplifies programming
- DIN rail mounting
- Current or voltage on the same module
- Typical connectable devices include transducers and positioning devices
- Each analogue module must be connected to the right of the main unit.

## Analogue input units

| Item | Specifications | |
|---|---|---|
| Number of analogue inputs | 1 | 4 |
| Input signal range | Voltage input: 1 to 5V<br>Current input: 4 to 20mA | |
| Resolution | Voltage input: 15.6mV (1/256)<br>Current input: 62.5μA (1/256) | |
| Accuracy | 25°C: ±1.0% max. of full scale<br>0° to 55°C: ±2.0% max. of full scale | |
| Conversion time | 20ms max. (per bit) | |
| External input impedance | Voltage input: 1MΩ min.<br>Current input: 250Ω | |
| Code input to CPU | 8-bit binary | |
| Number of words allocated | 2 | |
| Number of terminals on terminal block | 6 (fixed) | 26 (removable) |
| Insulation | Non-insulated | |
| Internal power supply | 250mA max. at 5V | |
| Weight | 250g max. | 750g max. |

## Analogue output unit

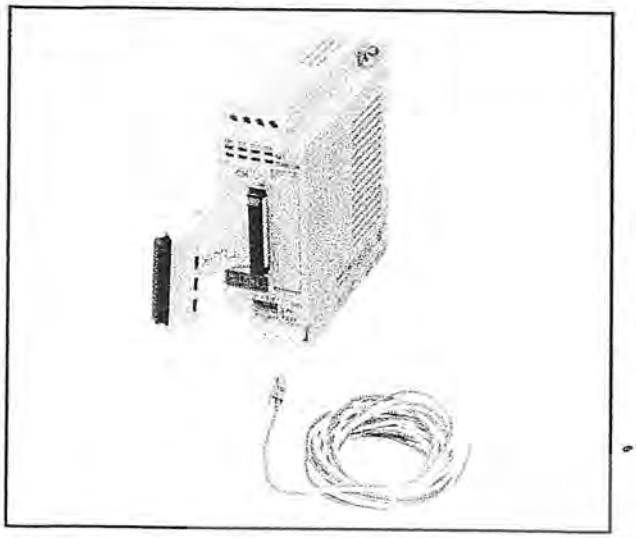
| Item | Specifications |
|---|---|
| Number of analogue outputs | 1 |
| Output signal range | Voltage output: 1 to 5V<br>Current output: 4 to 20mA |
| Resolution | Voltage output: 15.6mV (1/256)<br>Current output: 62.5μA (1/256) |
| Accuracy | 25°C: ±1.0% max. of full scale<br>0° to 55°C: ±2.0% max. of full scale |
| Conversion time | 20ms max. (per bit) |
| External input impedance | Voltage output: 0.5Ω max. |
| Maximum current for external output | Voltage output: 15mA max. |
| Permissible load resistance for external output | Current output: 250Ωmax. |
| Code required from CPU | 8-bit binary |
| Number of words allocated | 1 |
| Number of terminals on terminal block | 6 (fixed) |
| Insulation | Non-insulated |
| Internal power supply | 250mA max. at 5V |
| Weight | 250g max. |

See overview table for equivalent Omron part no.

| **RS** stock no. | Type |
|---|---|
| 731-928 | 1 channel input |
| 731-934 | 4 channel input |
| 731-940 | 1 channel input |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 4.2.3 Analogue timer unit

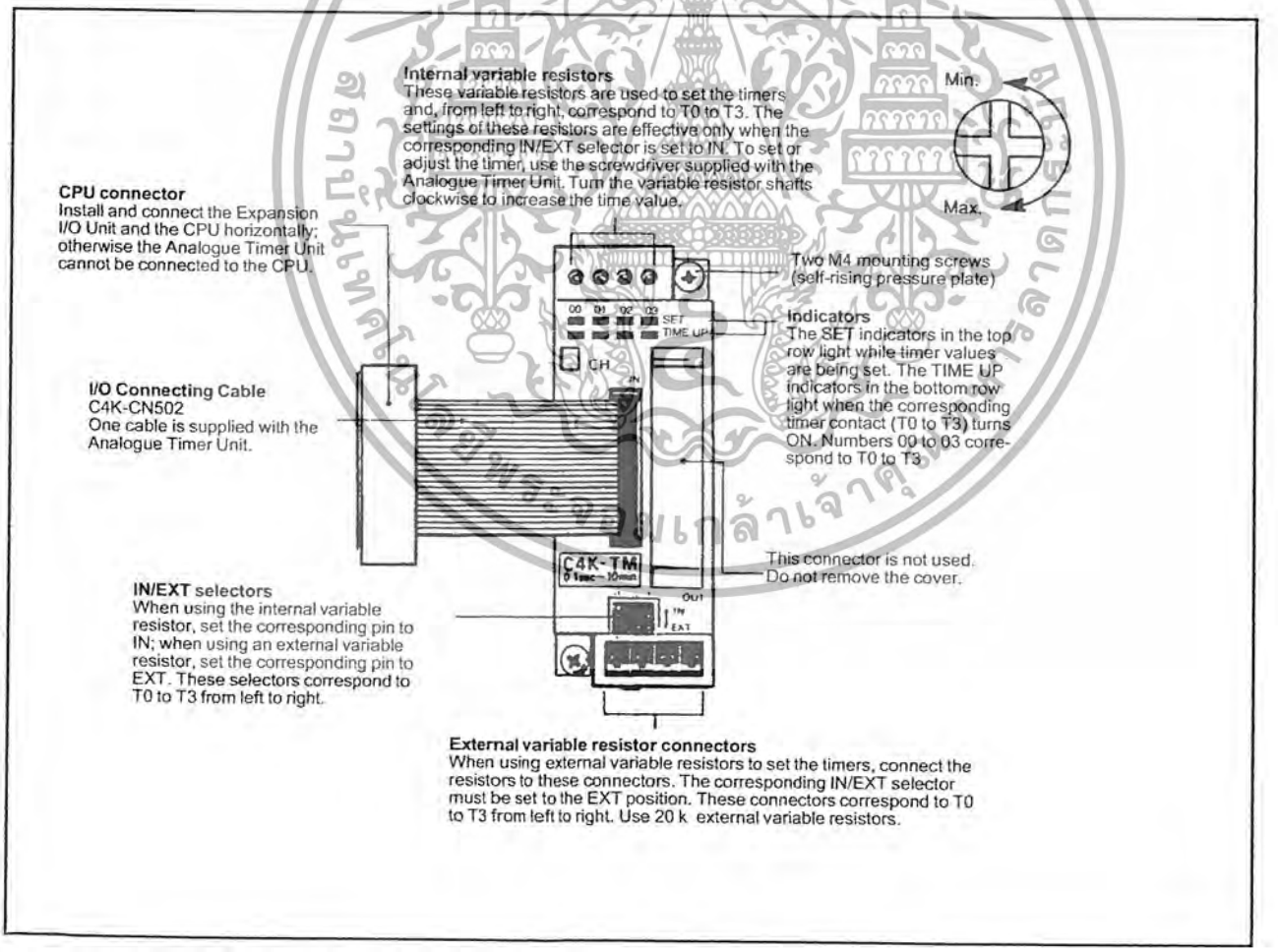


This module enables the set values of 4 timers to be adjusted either on the module or externally via a potentiometer which can easily be connected to terminals on the module.

It is ideal for instances where particular time values require to be changed but access to the CPU via programming or data access consoles does not want to be warranted. An operator who knows nothing about the PLC, can easily adjust the screws on the front of the analogue timer unit until he gets the control he requires; further, the time values which are being adjusted by the screws on the front of the module can be selected to be adjustable from external potentiometers (mounted on a panel door for instance) – thereby avoiding the need for the operator to open the panel door at all. The 4 timers can be individually set to be adjusted via the screws on the unit front or on external potentiometer.

### Features

- Time ranges 0.1 to 1.0s
  1.0 to 10s
  10 to 60s
  1 to 10m
- Timers can be paused by user program
- Timer 'coils' are regarded as outputs by the CPU, and timer 'contacts' appear as inputs to the CPU
- Timer set and 'time up' LED indication
- Can be used as cumulative timers
- DIN rail mountable
- Connecting cables for module included
- 2m connecting cable available to connect external potentiometer.



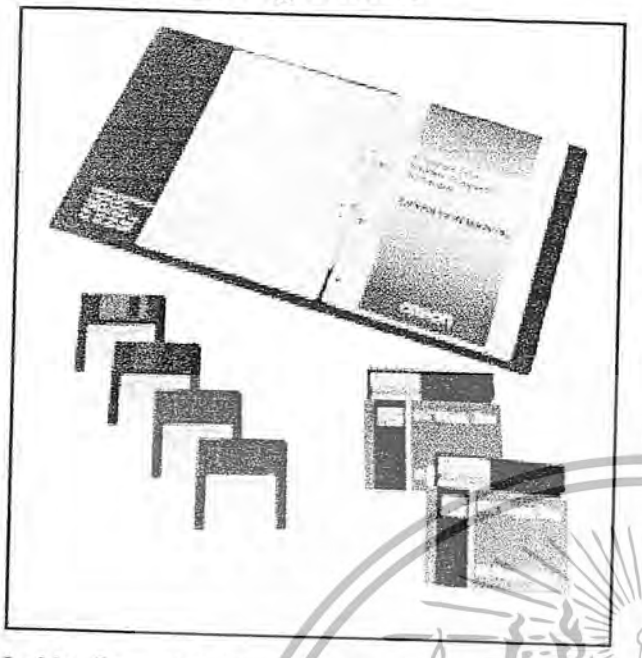

See overview table for equivalent Omron part no.

| **RS** stock no. | Type |
|---|---|
| 731-956 | Timer module |
| 731-962 | 2m cable |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# 5. Programmers

## 5.1 'LSS' programming, documentation and monitoring software

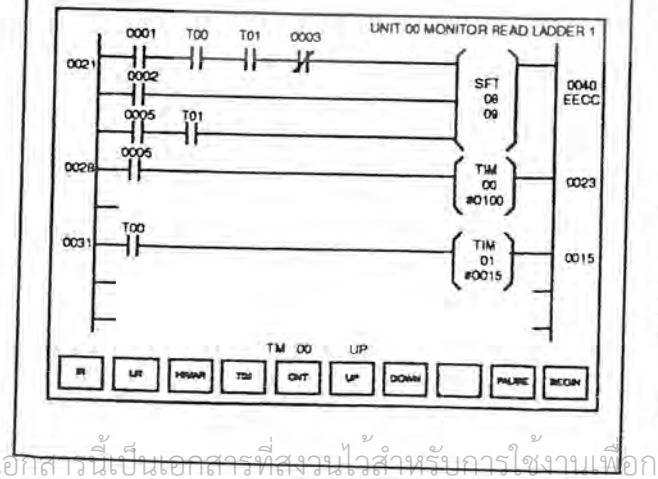


Ladder Support Software (LSS) provides the ability to program CK, CH and all other Omron C-series PLCs (not available through RS) using a personal computer.

It has the advantage of enabling programs to be written on a computer in either ladder diagram or mnemonic code and offers extensive documentation facilities so that once a program has been written and commissioned, it can be documented so that others looking at a machine or process for the first time have a clear indication of procedures, etc.

To connect the computer which is running LSS to the relevant CK PLC, a host link unit is required and mounted onto the front of the PLC; a host link is a device which enables RS-232 or RS-422 connection (depending on which type is ordered) to a host computer for programming or monitoring; refer to the communication modules section for other features of these host link modules.

Typical screen layout




### Features

#### Software library for easy management

- Each data diskette can contain a library of up to 255 program files with a file name, directory, serving to enable standardisation and program development.

#### Finger-tip switching between ladder diagrams and mnemonic code

- Switch quickly between ladder diagrams and the mnemonic code
- Display of up to 50 instruction lines and line comments
- Immediate programming (deletion or addition) capability.

#### ASCII-based DM area editor

- Display of up to 160 words at once
- ASCII-code input of messages word copying, and input of same contents to designated word range
- Transfer of DM area contents to PC.

#### Monitoring

- Any instruction instantly accessed and monitored
- ON bits displayed in yellow; OFF bits, in blue; and comments, in green
- Change between modes and change set values.

#### Multibit monitoring

- Monitor any PC data by bit or word with a 20 bit maximum
- Set or reset bits, or change present values.

#### Ladder diagram output

Printing out ladder diagrams or reference lists greatly increases programming and debugging efficiency. I/O cross references are provided with ladder diagrams to quickly show the application of bits, data words, timers, and counters.

#### Programming in ladder diagram symbols

- Up to 22 lines each containing a maximum of nine bits and one output can be written at one time
- Display of up to 11 lines without comments or six lines with comments
- Deleting, moving, or copying designated program blocks
- Program section save
- Copying comments from other programs.

#### Check program syntax

Three checking levels are available for new or altered programs.

#### Comment editing on I/O list

- On-screen editing of comments for all bits, including overwrites and searches
- I/O status readouts
- Advanced comment input to facilitate programming (comments provided when bits are accessed).

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

PLC to PC connections

Programs are transferred to the PLC through the RS-232C communications module RS stock no. 731-978, ordered separately.

Package consists of

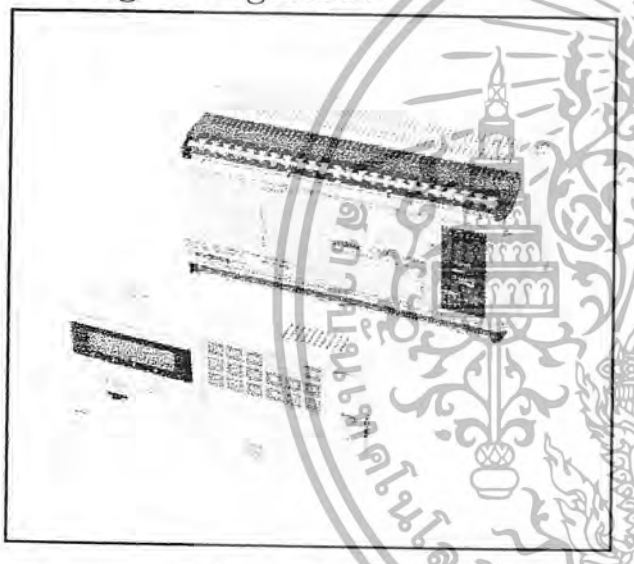
- Manual
- 3½in and 5¼in double sided, double density disks.

Minimum computer requirements

- 100% IBM XT, AT compatibility
- DOS 3.2 or above
- 640Kbyte RAM
- 3Mbyte hard disk
- Serial communications port
- Monochrome CGA, EGA or VGA display
- Floppy disk drive.

| **RS** stock no. | Type |
|---|---|
| 731-647 | LSS software |

## 5.2 Programming console



The console is ideal for inputting programs in mnemonic code and for very fast monitoring and fault finding during commissioning or operation stages.

The programming console offers full programming features on a back-lit, two line LCD display. The keypad is divided into clearly labelled and coloured areas with differing functions – white keys are used to input numeric values such as program address, data area address and operand values. These keys are also used for inputting function key numbers. Yellow keys are for actually writing or modifying programs. The grey keys are primarily used to input instructions and designate the type of data area being chosen for modification or data changing. Features available with the console which do not have an individual key are accessed by either the shift or FUN keys.

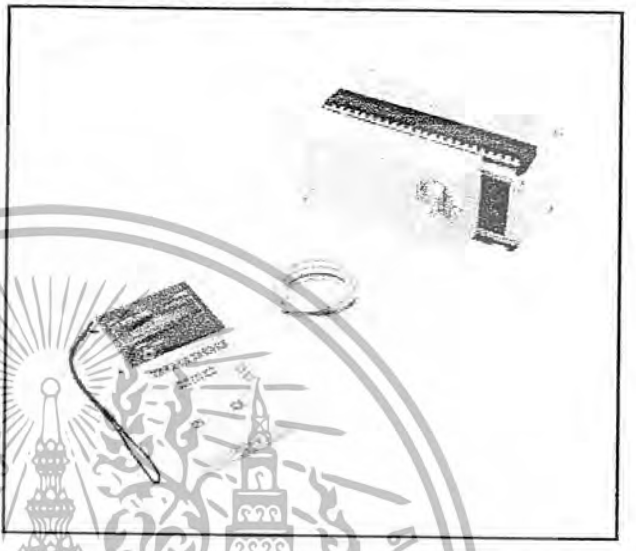
Features

- Password protected
- Can monitor every data register and channel in the CPU
- All CPU self diagnostics are displayed on console
- Programs can be stored on to a cassette tape
- 1m extension cable for easier use
- LED screen angle is adjustable
- Program check will return display messages, indicating errors in program
- Clips directly onto main unit.

Equivalent to Omron part no. 3 G2A6-PRO15E.

| **RS** stock no. | Type |
|---|---|
| 734-488 | Clip programmer |

## 5.3 Data Access Console



The Data Access Console is a device which only allows access to timers, counters and data registers for modification; it will not facilitate modification of the user program. It is ideal for use as an operator setting console where certain timers, counters and data registers on a machine or process may need to be changed periodically, but where program alteration (even by mistake) is not possible.

In addition to its data modification, the Data Access Console (DAC) can be used as a basic terminal rather like the programming console operating in terminal mode, allowing keyboard mapping to be displayed. As with the programmer, an overlay can be placed on the DAC to customise its operation.

A very useful feature on the DAC is its three level security operations; there are two different security levels which are entered via a keyswitch which allow varying levels of alteration; within these two security levels, there is a third level predetermined in software, which only allows only certain timers and counters etc to be modified. For example, on a particular operation it may be required for the operator to have access to timers and counters, but only to maybe three or four in each case – this is where the third security level can be used to 'block' out all other timers and counters.

The connecting cable is required to join the console to the adaptor.

Supplied with Data Access Console Operation Guide.

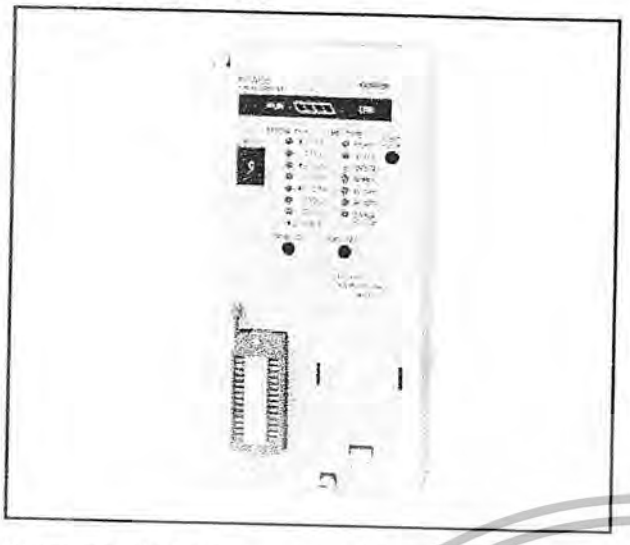
See overview table for equivalent Omron part no.

| **RS** stock no. | Type |
|---|---|
| 731-697 | Data access console |
| 731-675 | 2m cable |
| 734-551 | Adaptor |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# 6. Accessories

## 6.1 EPROM writer



Programs which have previously been stored in the battery backed RAM of the CPU can be easily transferred to EPROM via the EPROM writer. This module plugs directly onto the CPU and enables programs to be stored onto the standard EPROM, or read from EPROM. An EPROM can be tested by the writer to see if it is 'clean' and once downloaded onto EPROM, the writer performs a full verification check.

Equivalent to Omron part no. C500 PRW06.

| **RS** stock no. | Type |
|---|---|
| 734-539 | Writer |

## 6.2 Program memory chips

The program memory is stored either in inbuilt RAM memory or in a plug in EPROM memory.

RAM memory is useful for short term retention but the program will be lost on power off and battery fail.

EPROM memory provides long term retention and program protection. The program can only be altered by erasing the EPROM using an EPROM eraser and writing to it with a special EPROM writer. All chips have 8K word memories.

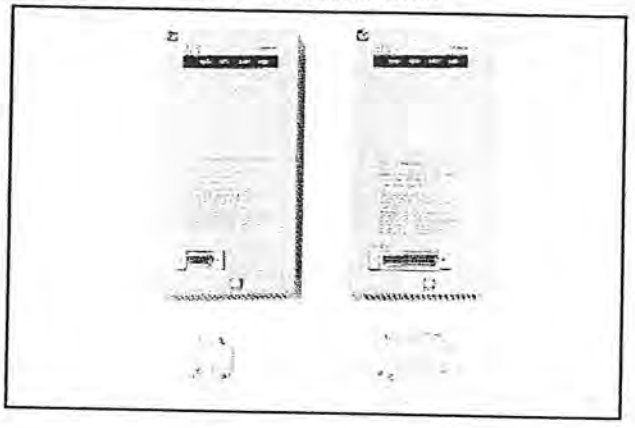
### Chip installation

The memory chip is installed in a socket beneath a cover on the front of the main unit.

See overview table for equivalent Omron part no.

| **RS** stock no. | Type |
|---|---|
| 734-545 | EPROM |
| 731-625 | RAM |

## 6.3 Communications modules



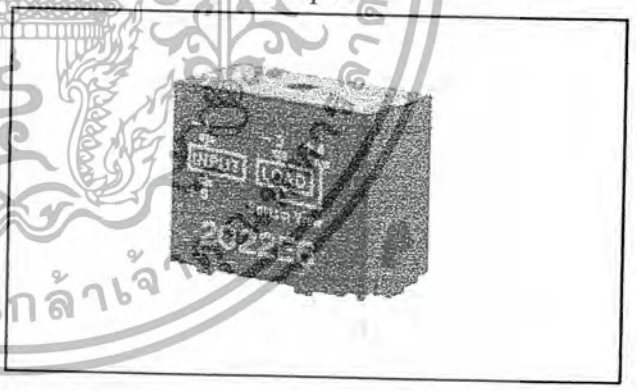
The CK CPUs can communicate to host computers with the addition of either RS-232 or RS-422 communications modules; these modules enable the CKs to be programmed by LSS via a computer or connection to a computer for SCADA applications. If desired, the user can write his own protocol for these modules and therefore perform communications to and from a computer with his own software. When communication is performed, it is done so without any interference with the user program – no allowance has to be made for a host unit being connected.

Supplied with an instruction leaflet.

See overview table for equivalent Omron part no.

| **RS** stock no. | Type |
|---|---|
| 731-978 | RS-232 |
| 731-984 | RS-422 |

## 6.4 Replacement outputs

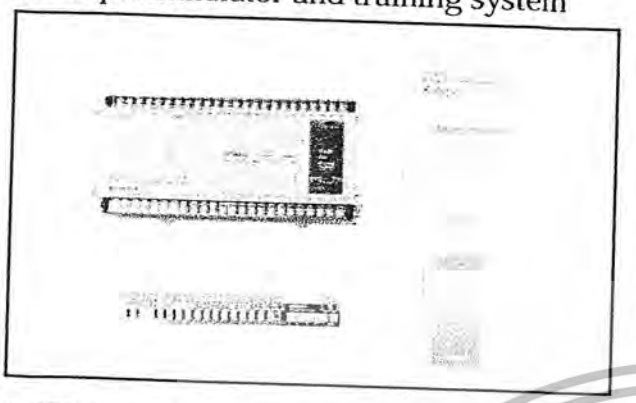



All relay output versions of the CK range have their relays mounted into sockets, this not only allows for easy maintenance should an output become faulty, but also enables individual outputs to be removed and exchanged for a transistor or a triac which may have a more suitable switching characteristic for a particular application (there is a tool for removing the outputs conveniently housed inside the cover of the CK CPU). Now, because the outputs terminals of the CK CPUs are arranged so that they are served by various common terminals (some output terminals being served by individual commons – see wiring details), it is possible to have the majority of outputs being switched by relay contacts, some being switched at 24Vdc via transistors (NPN or PNP) and others being switched at 110Vac by triacs. This tailoring of output circuits allows for very flexible system design and can negate the need for future maintenance of outputs because loads are being switched by correctly matched devices.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

| RS stock no. | Type |
|---|---|
| 734-494 | NPN transistor |
| 734-501 | PNP transistor |
| 734-517 | Triac |

## 6.5 Input simulator and training system



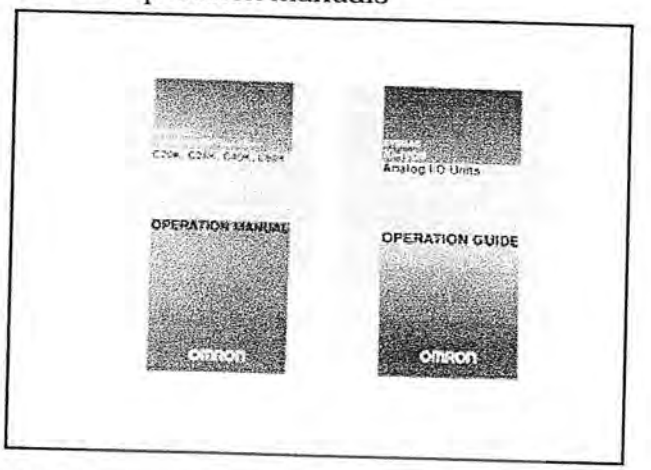
A PCB carrying an extra array of toggle switches for energising the inputs to the PLC using the on-board 24Vdc supply. Designed for use with 20 I/O or 28 I/O CK main unit, used for program monitoring and testing.

Also supplied is a 60 minute audio cassette tape and booklet for a basic step by step introduction to programming the CK range using the CK programming console, RS stock no. 734-488.

Equivalent to Omron part no. C20K-ETL01.

| RS stock no. | Type |
|---|---|
| 734-567 | Input simulator and training system |

## 6.6 CK operation manuals




The CK manuals include the CK Operation Manual and the CK Analogue I/O Manual.

The Operating Manual gives a full overview on PLC hardware, memory areas, programming, instruction set, program execution timing, debugging and execution and troubleshooting.

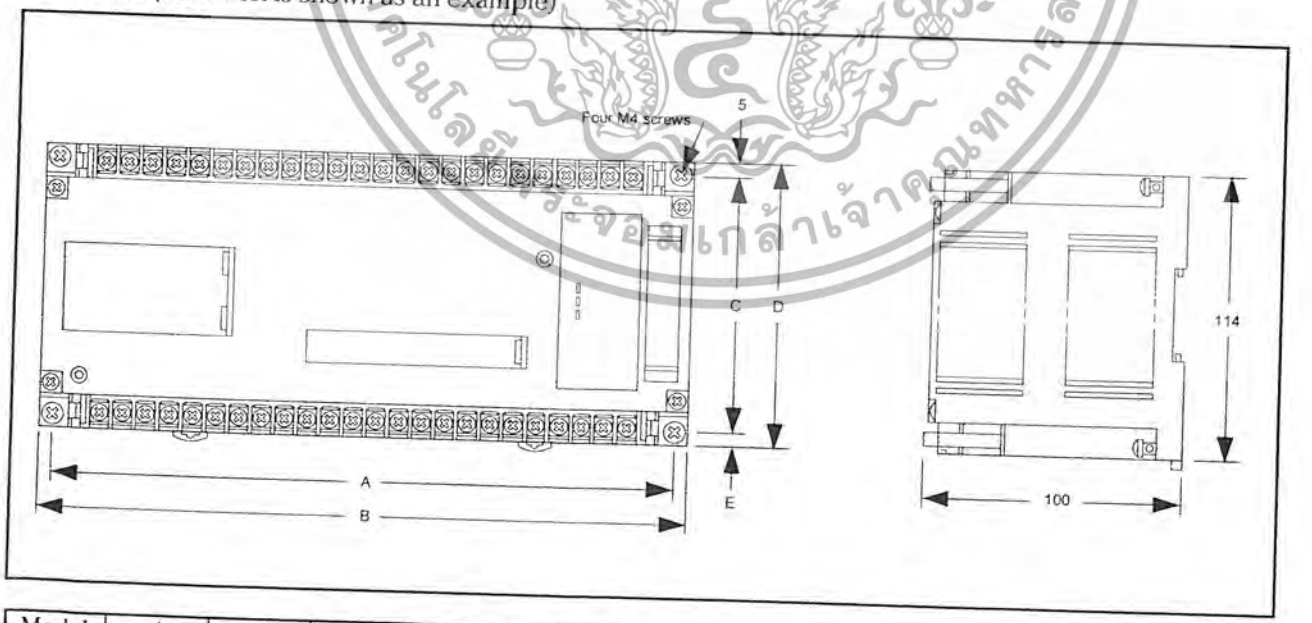
The Analogue Manual gives an overview of analogue to digital conversion, and how the unit should be used, giving examples of ladder diagram programs.

Omron part no. operation manuals

| RS stock no. | Type |
|---|---|
| 734-573 | CK manuals |

## 7. Dimensions

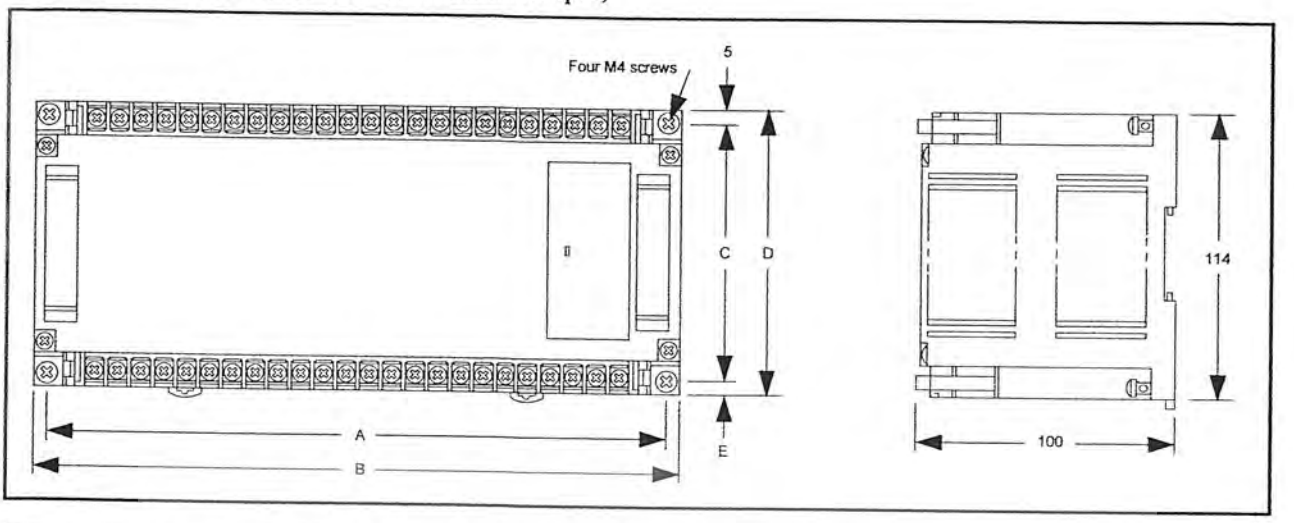
Main units (the C20K is shown as an example)




| Model | A | B | C | D | E |
|---|---|---|---|---|---|
| C20K | 240 | 250 | 100 | 110 | 5 |
| C28K | 240 | 250 | 100 | 110 | 5 |
| C40K | 290 | 300 | 100 | 110 | 5 |
| C60K | 340 | 350 | 120 | 140 | 15 |

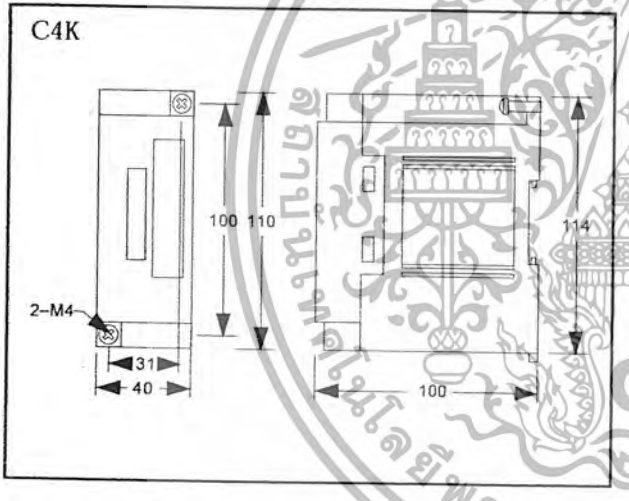
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Expansion units (The C20P is shown as an example)

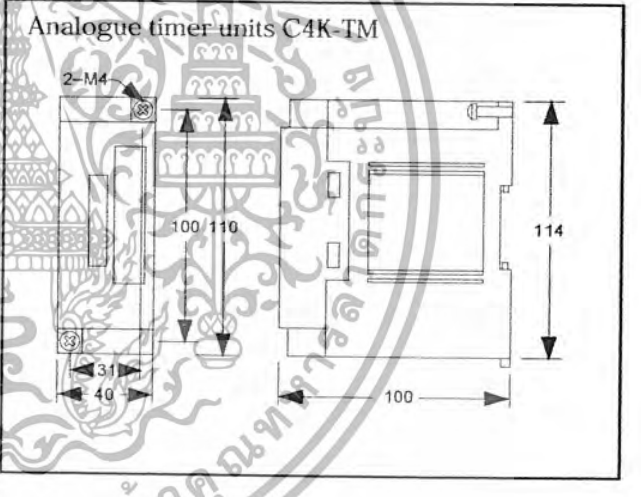



| Model | A | B | C | D | E |
|---|---|---|---|---|---|
| C16P | 145 | 155 | 100 | 110 | 5 |
| C20P | 240 | 250 | 100 | 110 | 5 |
| C28P | 240 | 250 | 100 | 110 | 5 |
| C40P | 290 | 300 | 100 | 110 | 5 |
| C60P | 340 | 350 | 120 | 140 | 15 |

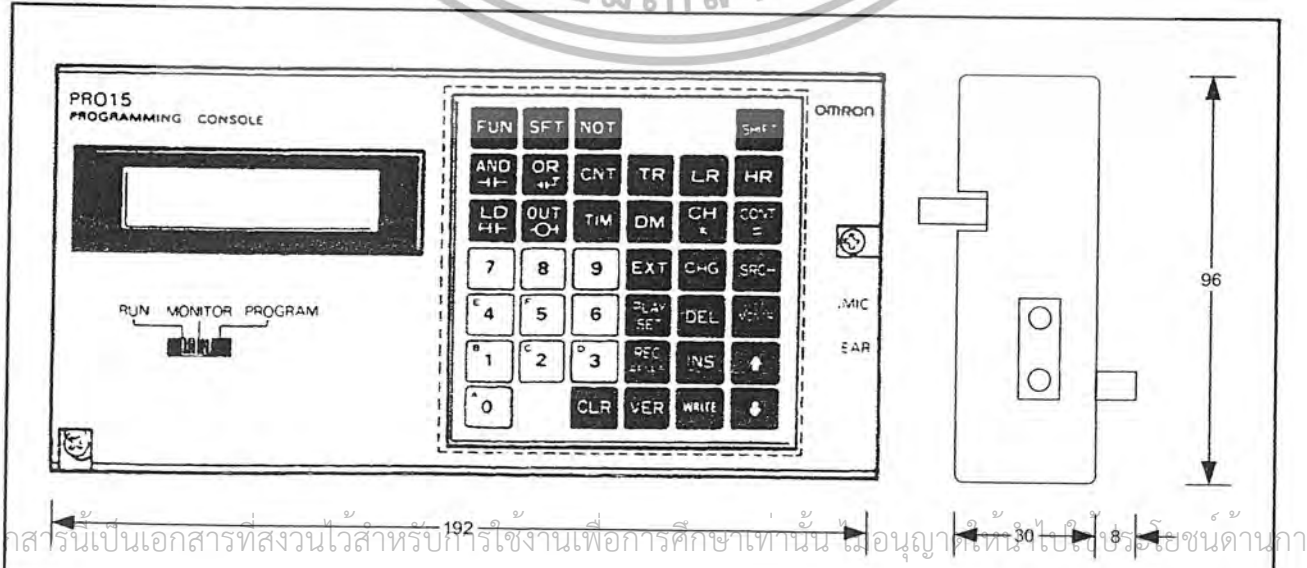
C4K




Analogue timer units C4K-TM

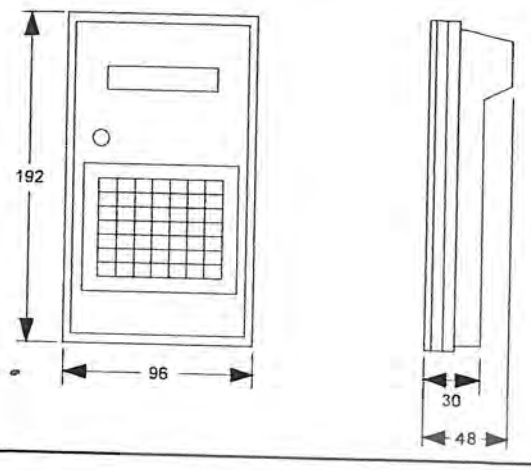



Programming console




เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Data access unit and advanced programming console




Other dimensions

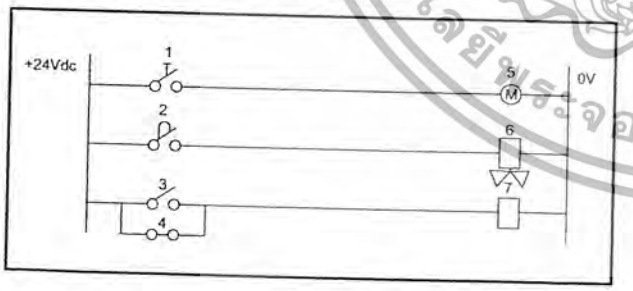
| Type | W | D | M |
|---|---|---|---|
| Programming adaptor | 192 | 54 | 96 |
| Communications modules | 192 | 54 | 96 |
| Transistor/Triac module | 20 | 10 | 20 |
| CK analogue manual | 210 | 2 | 297 |
| CK operation manual | 210 | 9 | 297 |
| LSS programming software | 261 | 318 | 48 |
| EPROM writer | 192 | 59 | 96 |

# 8. Programming

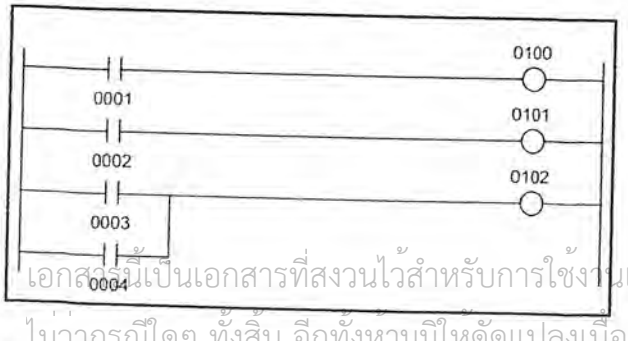
## 8.1 Ladder diagram programming

There are only a few simple instructions which are needed to create sophisticated circuit diagrams; these are outlined below:

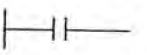
Conventional electrical diagrams have positive and negative control lines as follows:



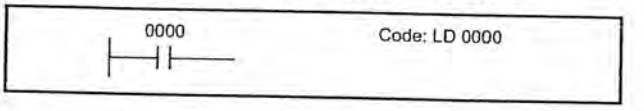

In essence, ladder diagram is no different; a typical circuit diagram would have a Start line or BUS and an End line or BUS:

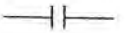



## 8.2 Symbols and what they mean

LD symbol 

An 'LD' instruction is always used when beginning a new 'network' of contacts and outputs within a circuit – it is normally the first contact from the start line.

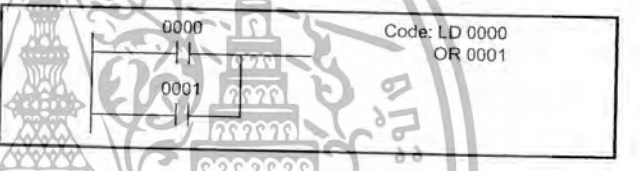



AND symbol 

An 'AND' instruction is used to put a contact directly in series with another contact.

OR symbol

An 'OR' instruction is used to put a contact directly in parallel with another contact.

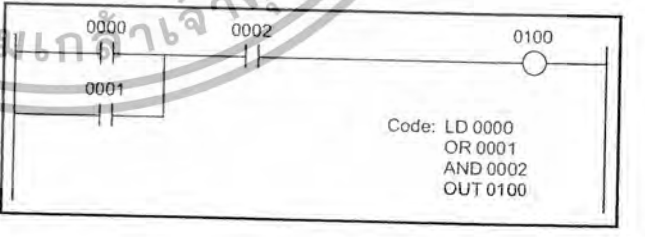



OUT symbol

An 'OUT' instruction designates which bit is the output for that particular network.

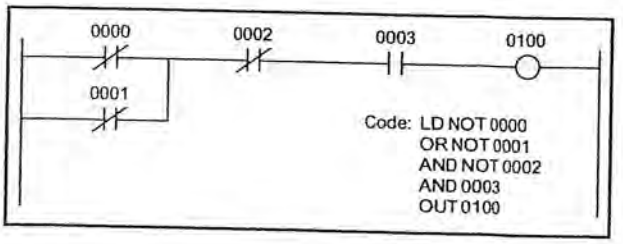
## 8.3 Example circuits

So, these basic instructions can program most required circuits, for example:



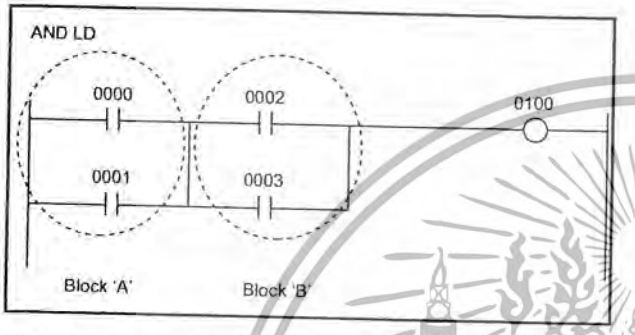

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

All the contacts shown so far are 'normally open'; to program a 'normally closed' contact, NOT is placed after the contact, eg.




More complex circuits can be drawn with two more instructions:

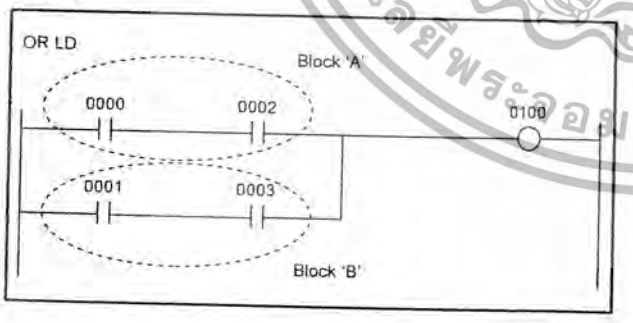
AND LD




This circuit is divided into two 'blocks' – 'A' and 'B'. The load block 'A' followed by block 'B'; and because block 'A' AND block 'B' need to be valid to operate the output, AND LD is required to join the two blocks. The code becomes:

LD 0000
OR 0001
LDF 0002
OR 0003
AND LD
OUT 0100

OR LD




This circuit needs a similar principle as applied to the circuit above with the exception that we need to OR block 'A' and 'B' together. The coding for this circuit then becomes:

LD 0000
AND 0002
LD 0001
AND 0003
OR LD
OUT 0100

Complicated control circuits can now be programmed. Obviously, if the outputs are combined with timers, counters, shift-registers, etc, this begins to develop a powerful control system. Some of these facilities are explained in the next section.

## 8.4 High Speed Drum (HDM) (High speed counter)

There is a high speed counter available on all of the CK PLCs which has a guaranteed response time of 2kHz; the input marked 0000 on the CPU is always the input used if this facility is required (see wiring details); in addition to the high speed input, there is a high speed reset which will respond to high speed pulses from external devices to reset the high speed counter.

The HDM is essentially a 'Drum Counter' which rotates every time a pulse is applied to the high speed input (number 0000); the cumulative count value of the drum is constantly compared to a 'Drum setting table' which permits up to 32 outputs to be generated depending on the value of the Drum cumulative count; the table is set similar to that shown on next page.

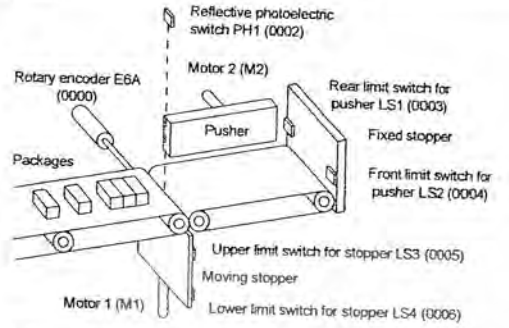
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

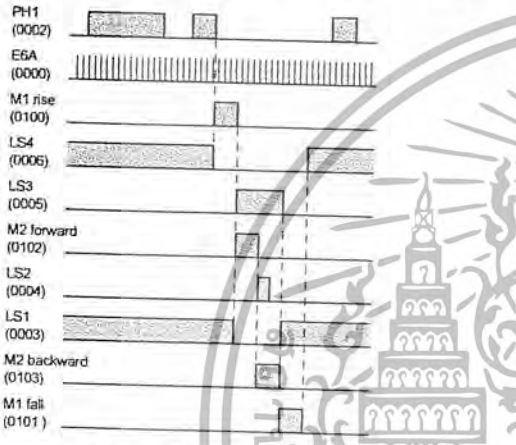
**Example: Packaging machine**

The high-speed counter is very useful this application. Here, packages are being carried on a conveyor belt at random intervals. Some of them are spaced far apart and others are clustered together, making it impossible to accurately detect and count them with photoelectric switches alone.

By presetting the number of pulses generated when a single package is detected and by counting those pulses, the number of packages can be accurately counted, regardless of whether the packages are spaced or clustered.

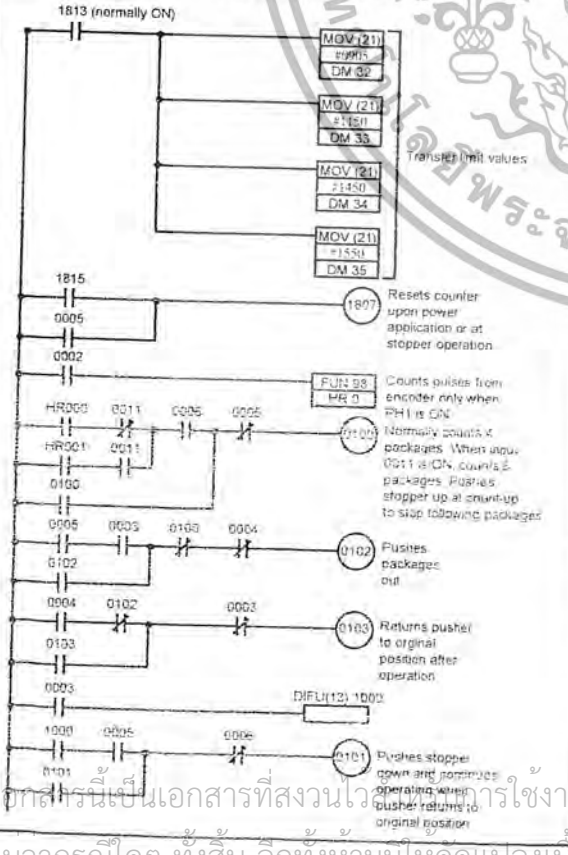
The following diagram shows the packaging system and the corresponding timing chart.




In this example, 'x' is the number of pulses per package. To detect four packages, therefore 4x must be set as the preset value of the high-speed counter.

Here is the program example for the application:




## 8.5 Reversible drum counter (RDM)

The RDM is similar in operation to the HDM in that it is a Drum counter which 'spins' when a count pulse is applied; the main differences with the RDM are that it can count in both a forward and reverse direction depending on which signal is applied, and that it operates from normal digital input signals and does not count at the speed of the HDM.

Typical applications for the RDM are the control of objects with a constantly changing direction, such as a carousel table.

## 8.6 Interlocks and jumps

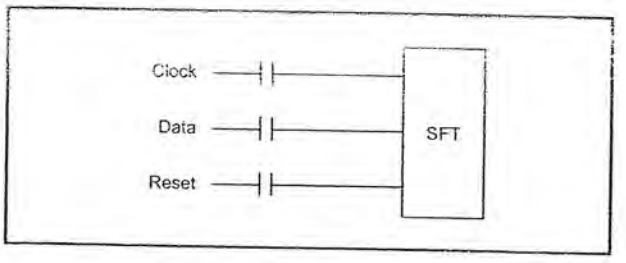
Interlocks are used where groups or blocks of circuit diagram may need to be 'skipped' by the program because they are not relevant for the application which is currently being run. An example may be a machine which has a manual operation circuit and an automatic operation circuit – when one of these operations is chosen the other operations circuit is ignored.

The principle of 'Jumps' is similar to that of interlocks in that groups of programs can be 'skipped' or 'jumped' depending on whether they are necessary at a particular point of an applications control cycle. Jumps have the additional feature of being able to jump to numbers. This means that the exact line can be designated exactly where to jump to in the program. (Interlocks will 'skip' the circuit concerned until it finds the end of that interlock.) Another point is the condition of outputs, timers etc within the two instructions blocks; interlocks will reset most output types whereas jumps will not.

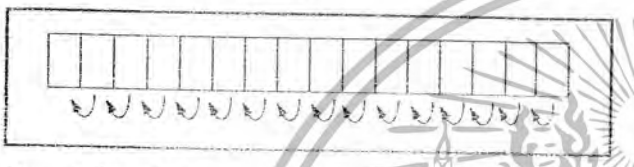
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 8.7 Shift register

The shift register is a very useful and versatile instruction which can be used for many different applications – especially those relating to sequencing events or tracking products in a process. The instruction is programmed as follows:




Essentially the shift register is controlled by the clock and data inputs (with the reset resetting the complete instruction); the idea is that a group of 16 bits (a word or channel of data) is shifted together as shown below each time a clock is operated.



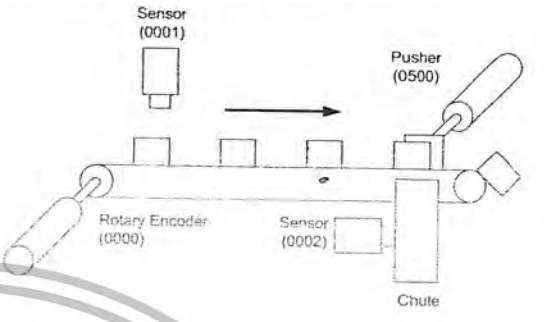
Now, if the clock is present, and the data is also present, then not only is the complete word shifted one place, but the data signal is loaded into the beginning of the channel.

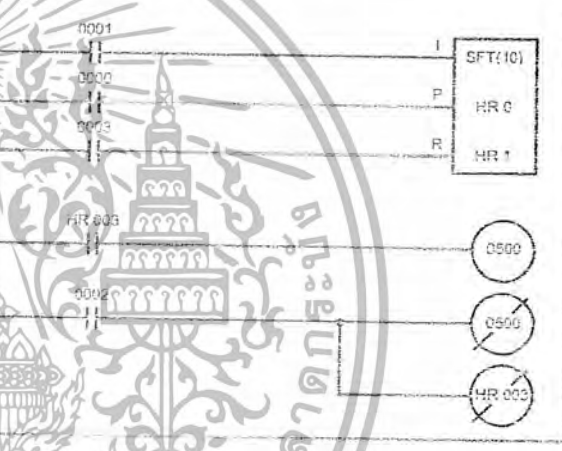
The use of the shift register can be seen when you recognise that as the '1' is shifted along a word it is actually turning on individual bits within that word in sequence, eg if the word is comprised of [illegible] shifted along word [illegible] sequence (the shifting being [illegible] to be the clock).

**Example: Conveyor line**

This program controls the conveyor line shown below so that faulty products detected at the sensor are pushed down a chute. To do this, the execution condition determined by inputs from the first sensor (0001) are stored in a shift register: ON for good products; OFF for faulty ones. Conveyor speed has been adjusted so that HR 003 of the shift register can be used to activate a pusher (0500) when a faulty product reaches it; ie, when HR 003 turns ON, 0500 is turned ON to activate the pusher.

The program is set up so that a rotary encoder (0000) controls execution of SFT(10) through a DIFU(13); the rotary encoder is set up to turn ON and OFF each time a product passes the first sensor. Another sensor (0002) is used to detect faulty products in the chute so that the pusher output and HR 003 of the shift register can be reset as required.



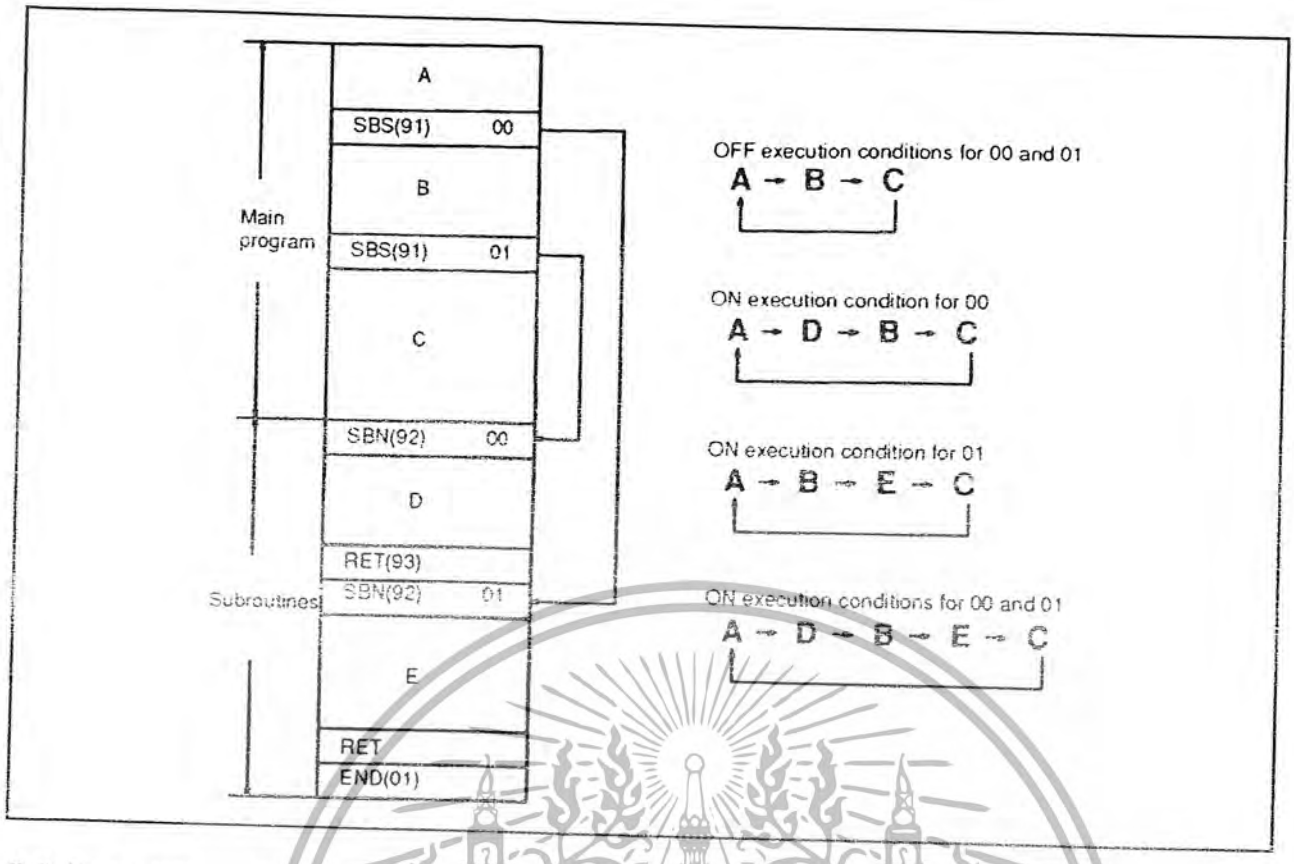




A powerful extension to the shift register is the reversible shift register which operates in a similar manner to the standard shift register but will operate in forward or reverse. This instruction is very useful for handling products on a conveyor with a reversible type of conveyor which can operate in clockwise/counter-clockwise or forward/reverse direction.

## 8.8 Subroutines

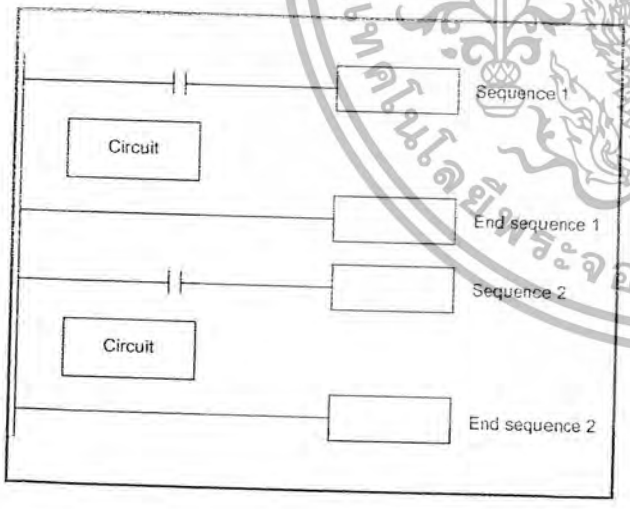
Subroutines are used when particular processes or actions are required to operate upon demand from a main system process or action. For example, a packaging machine might have conventional packaging control hardware and circuits and this would be catered for in the main PLC program. In addition, the machine may have a small conveyor and pallet circuit for handling products which the main control circuit and equipment have deemed to be faulty. Now, the PLC circuit to control this ancillary conveyor system only needs to be run when there is a faulty product detected by the main circuit – this makes it an ideal candidate for a subroutine circuit. It is only scanned and processed when called for by the main circuit. Subroutines ease program writing and can optimise the PLC scan time and enable easier fault finding and diagnostics.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้




## 8.9 Step sequence

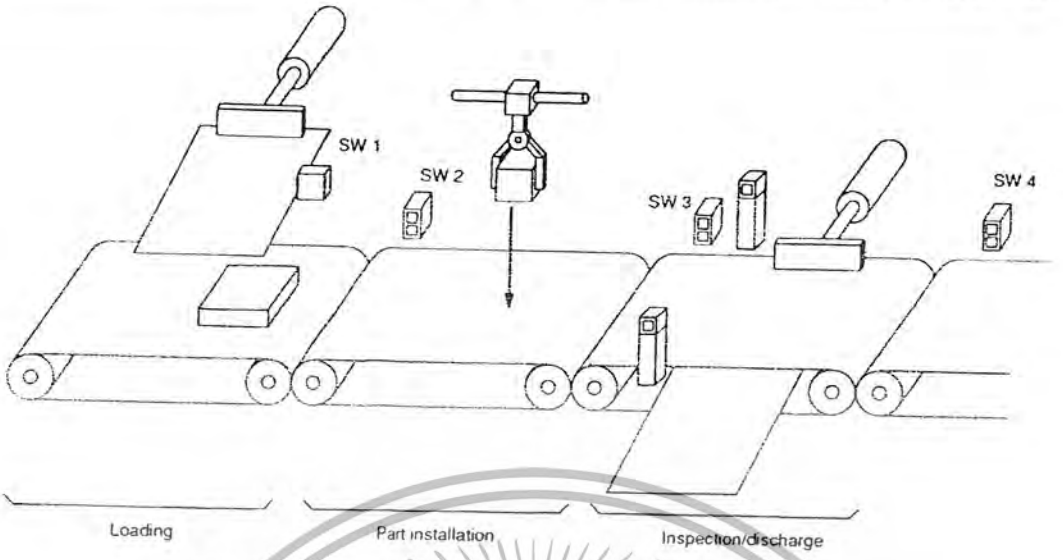
The step sequence instruction is very useful where an application requires one process to be completed before another can start (rather like a baton has to be passed before the next runner can begin). The example below shows how a step sequence would be programmed.




เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

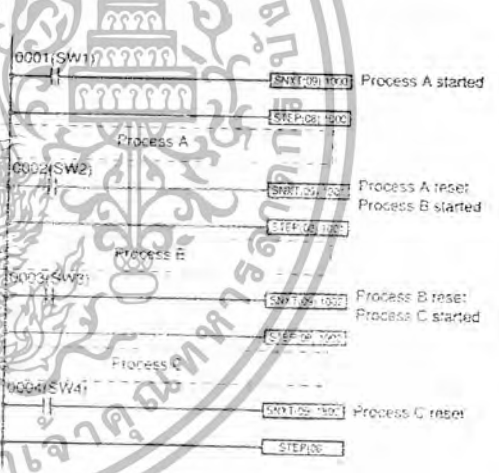
**Example: Conveyor line**

The following process requires that three processes, loading, part installation, and inspection/discharge, be executed in sequence with each process being reset before continuing on to the next process. Various sensors (SW1, SW2, SW3 and SW4) are positioned to signal when processes are to start and end.




The following diagram demonstrates the flow of processing and the switches that are used for execution control.

The program for this process, shown below, utilises the most basic type of step programming: each step is completed by a unique SNXT(09) that starts the next step. Each step starts when the switch that indicates the previous step has been completed turns ON.




เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 8.10 I/O refresh

A PLC's CPU has a number of tasks to perform – it reads input signals, checks hardware such as host links and peripheral ports, sets outputs etc. – all this in addition to running the user program. the combination of all these tasks is called the scan time. If the CPU is performing its normal scan time routines, and a very fast input were to come on and off before the CPU could read in, that signal will not be registered to the CPU. In cases like this, I/O refresh can instruct the CPU to periodically halt its current operation, refresh the I/O designated by the command and then return to exactly where it left off. This instruction therefore enables certain inputs and outputs to be read or set quicker than would normally be possible.

## 8.11 Failure alarm (FAL)

The FAL command allows the user to program up to 100 alarm conditions unique to the application. whenever one of these alarms is activated in the program, a unique code relating to that alarm is produced in the FAL area in the memory map. This means that whenever there is a fault or alarm in a process, only one FAL code need be monitored to identify what the problem is. For example, [illegible] designate the even 10 as [illegible] codes, now. If the machine [illegible] guard switch, and its FAL area was checked, it would read code 10. Which is [illegible] the RS232 [illegible] FAL [illegible] of line because the [illegible] easily fixed. [illegible]

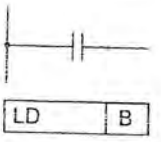
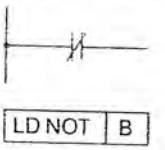
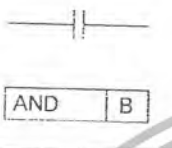
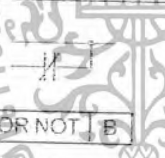
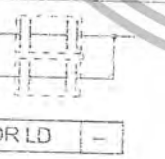
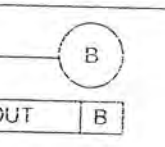
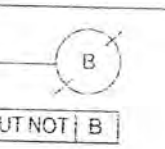
What makes this [illegible] all system [illegible] also see to the [illegible] number greater than [illegible] guarantee [illegible] diagnostics

## 9. Programming instructions

| Function code | Name | Mnemonic |
|---|---|---|
| – | Load | LD |
| – | Load Inverse | LD NOT |
| – | AND | AND |
| – | AND Inverse | AND NOT |
| – | OR | OR |
| – | OR Inverse | OR-NOT |
| – | AND Load | AND LD |
| – | OR Load | OR-LD |
| – | Output | OUT |
| – | Output Inverse | OUT-GNT |
| – | Timer | TIM |
| – | Counter | CNT |
| 00 | No Operation | NOP |
| 01 | End | END |
| 02 | Interlock | IL |
| 03 | Interlock Clear | ILC |
| 04 | Jump | JMP |
| 05 | Jump End | JME |
| 06 | Single Step | STEP |
| 07 | Step Advance | SNXT |
| 10 | Shift Register | SFT |
| [illegible] | Keep | KEEP |
| [illegible] | Reversible Counter | CNTR |
| [illegible] | Differentiate Up | DIFU |
| [illegible] | Differentiate Down | DIFD |
| [illegible] | [illegible] | TIMH |
| [illegible] | Word Shift | WSFT |
| [illegible] | [illegible] | CMP |
| [illegible] | Move | MOV |
| [illegible] | Move Inverse | MVN |
| [illegible] | BCD → Binary | BIN |
| [illegible] | Binary → BCD | BCD |
| [illegible] | BCD Add | ADD |
| [illegible] | BCD Subtract | SUB |
| [illegible] | BCD Multiply | MUL |
| 33 | BCD Divide | DIV |
| [illegible] | Set Carry | STC |
| 41 | Clear Carry | CLC |
| [illegible] | Reversible Drum Counter | RDM |
| [illegible] | High-Speed Counter | HSCH |
| [illegible] | End Wait | ENDW |
| [illegible] | Network Identifier | NETW |
| 76 | 7 to 12 Decoder | MLPX |
| 77 | Encoder | DMPX |
| 84 | Reversible Shift Register | SFTR |
| 91 | Subroutine Entry | SBS |
| [illegible] | Subroutine Definition Start | SBN |
| 93 | Subroutine Definition End | RET |
| 97 | I/O Refresh | IORF |

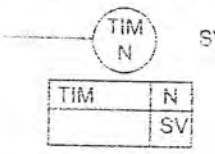
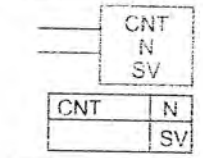
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## Ladder diagram instructions

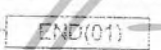
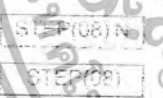
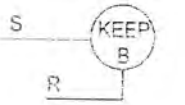
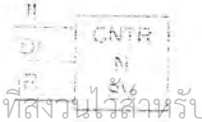
| Name<br>mnemonic | Symbol | Function | Operands |
|---|---|---|---|
| Load<br>LD |  | Used to start instruction block with status of designated bit. | B:<br>IR<br>SR<br>HR<br>TC<br>TR |
| Load Inverse<br>LD NOT |  | Used to start instruction block with inverse of designated bit. | B:<br>IR<br>SR<br>HR<br>TC<br>TR |
| AND<br>AND |  | Logically ANDs status of designated bit with execution condition. | B:<br>IR<br>SR<br>HR<br>TC<br>TR |
| AND Inverse<br>AND NOT | | Logically ANDs inverse of designated bit with execution condition. | B:<br>IR<br>SR<br>HR<br>TC<br>TR |
| OR<br>OR | | Logically ORs status of designated bit with execution condition. | B:<br>IR<br>SR<br>HR<br>TC<br>TR |
| OR Inverse<br>OR NOT |  | Logically ORs inverse of designated bit with execution condition. | B:<br>IR<br>SR<br>HR<br>TC<br>TR |
| Block AND<br>AND LD | | Logically ANDs results of preceding blocks. | None |
| Block OR<br>OR LD |  | Logically ORs results of preceding blocks. | None |
| Output<br>OUT |  | Turns ON designated bit. | B:<br>IR<br>HR<br>TR |
| Output Inverse<br>OUT NOT |  | Turns OFF designated bit. | B:<br>IR<br>HR<br>TR |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## Ladder diagram instructions (continued)

| Name<br>mnemonic | Symbol | Function | Operands |
|---|---|---|---|
| Timer<br>TIM |  | ON-delay (decrementing) timer operation. Set value: 999.9 s; accuracy: +0.0/−0.1 s. Same TC bit cannot be assigned to more than one timer/counter. The TC bit is input as a constant. | N: TC<br>SV:<br>IR<br>HR<br># |
| Counter<br>CNT |  | A decrementing counter. SV: 0 to 9999; CP: count pulse; R: reset input. The TC bit is input as a constant. | N: TC<br>SV:<br>IR<br>HR<br># |

## Special instructions

| Name<br>mnemonic | Symbol | Function | Operands |
|---|---|---|---|
| No Operation<br>NOP (00) | None | Nothing is executed and next instruction is [illegible] | None |
| End<br>END(01) |  | Required at the end of the program. | None |
| Interlock<br>IL(02)<br>Interlock Clear<br>ILC(03) | | [illegible] | None |
| Jump<br>JMP(04)<br>Jump End<br>JME(05) | | [illegible] | [illegible] |
| Single Step<br>STEP(08) |  | [illegible] | [illegible] |
| Step Advance<br>SNXT(09) | | [illegible] | [illegible] |
| Shift Register<br>SFT(10) | | [illegible] | St, E:<br>IR<br>HR |
| Latching Relay<br>KEEP (11) |  | Defines a bit (B) as a latch controlled by set (S) and reset (R) inputs. | B:<br>IR<br>HR |
| Reversible Counter<br>CNTR (12) |  | Increases or decreases PV by one when ever the increment input (I) or decrement input (D) signal goes from OFF to ON. Set to reset input. Must not access the [illegible] | N:<br>TC<br>SV:<br>IR<br>HR |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

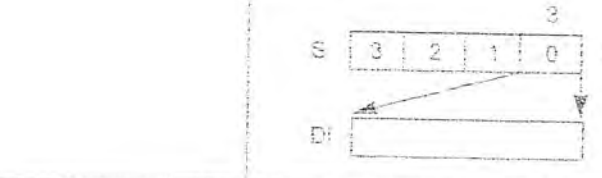
Special instructions (continued)

| Name<br>mnemonic | Symbol | Function | Operands |
|---|---|---|---|
| Differentiate Up<br>DIFU(13)<br>Differentiate Down<br>DIFD(14) | DIFU(13) \| B<br>DIFD(14) \| B | DIFU turns ON the designated bit (B) for one scan on the rising edge of the input signal; DIFD turns ON the bit for one scan on the trailing edge. The maximum number of DIFU/DIFDs is 48. | B:<br>IR<br>HR |
| High-speed Timer<br>TIMH(15) | TI — TIMH N — SV | A high-speed ON-delay (decrementing) timer. SV: 0.01 to 99.99s, accuracy +0.00/–0.01 s. Must not be assigned the same TC bit as another timer/counter. The TC bit is input as a constant. | N: SV:<br>TC: IR<br>HR<br>#<br>DM |
| Word Shift<br>WSFT(16) | WSFT(16)<br>St<br>E | Shifts data between the start and end words in word units. | St/E:<br>IR<br>HR<br>DM |
| Compare<br>CMP(20) | CMP(20)<br>Cp1<br>Cp2 | Compares two sets of four-digit hexadecimal data (Cp1 and Cp2) and outputs results to GR, EQ and LE. | Cp1/Cp2:<br>IR<br>SR<br>HR<br>TC<br>DM<br># |
| Move<br>MOV(21) | MOV(21)<br>S<br>D | Transfers source data (S) (word or four-digit constant) to destination word (D). | S: D:<br>IR IR<br>SR HR<br>HR DM<br>TC<br>DM<br># |
| Move Inverse<br>MVN(22) | MVN(22)<br>S<br>D | Inverts source data (S) (word or [illegible]) and transfers it to destination word (D). | S: D:<br>IR<br>SR HR<br>HR DM<br>TC<br>DM |
| BCD to Binary<br>BIN(23) | BIN(23)<br>S<br>R | Converts four-digit BCD data in source word (S) into 16-bit binary data and outputs converted data to result word (R). | S: D:<br>IR<br>SR<br>HR DM<br>TC<br>DM |
| Binary to BCD<br>BCD(24) | BCD(24)<br>S<br>R | Converts binary data in source word (S) into BCD and outputs converted data to result word (R). | S: R:<br>IR IR<br>SR HR<br>HR DM<br>DM |
| BCD Add<br>ADD(30) | ADD(30)<br>Au<br>Ad<br>R | Adds two four-digit BCD values (Au and Ad) and content of CY and outputs result to specified result word (R). | Au/Ad: R:<br>IR IR<br>SR HR<br>HR DM<br>DM<br># |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

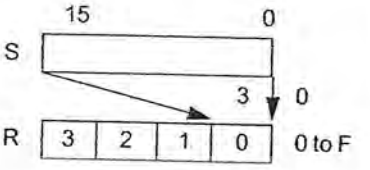
Special instructions (continued)

| Name<br>mnemonic | Symbol | Function | Operands |
|---|---|---|---|
| BCD<br>Subtract<br>SUB(31) | SUB(31)<br>Mi<br>Su<br>R | Subtracts both four-digit BCD subtrahend (Su) and content of CY from four-digit BCD minuend (Mi) and outputs result to specified result word (R).<br>Mi − Su → CY → R CY | Mi/Su: R:<br>IR IR<br>SR HR<br>HR DM<br>TC<br>DM<br># |
| BCD<br>Multiply<br>MUL(32) | MUL(32)<br>Md<br>Mr<br>R | Multiplies a words data or a four-digit BCD value (Md) and another word's data (Mr) and outputs the result to a specified result word (R).<br>Md × Mr → R R + 1 | Md/Mr: R:<br>IR IR<br>SR HR<br>HR DM<br>TC<br>DM<br># |
| BCD<br>Divide<br>DIV(33) | DIV(33)<br>Dd<br>Dr<br>R | Divides a words data or a four-digit BCD dividend (Dd) and another word's data (Dr) and outputs result to specified result word (R).<br>R / R + 1 | Dd/Dr: R:<br>IR IR<br>SR HR<br>HR DM<br>TC<br>DM<br># |
| Set Carry<br>STC(40) | STC(40) | Sets carry flag (CY) to 1. | None |
| Clear Carry<br>CLC(41) | CLC(41) | Clears carry flag (CY) to 0. | None |
| Reversible<br>Drum Counter<br>RDM(83) | RDM(83)<br>C<br>R | Reversible drum counter (R to R + 2) | C:<br>R<br>DM |
| High-speed<br>Drum Counter<br>HDM(84) | HDM(84)<br>C<br>R | Accepts high-speed pulse input | C:<br>R<br>DM |
| End Wait<br>ENDW(62) | ENDW(62)<br>Bit | Used to force a scan time wait | Bit: |
| Network<br>Identifier<br>NETW(63) | NETW(63)<br># | Used to leave comments in the program. | # |
| 4 to 16<br>Decoder<br>MLPX(76) | MLPX(76)<br>S<br>Di<br>R | Converts data in four-bit (1 digit) units in specified word to 16-bit data (1 word) and outputs it to specified result word. Conversion is performed in BCD or hexadecimal. Bits from 0 to 3 of S are converted. Next, Bit to Bit number of digits to be converted minus 1. | S: Di: R:<br>IR IR IR<br>SR HR HR<br>HR DM<br>TC<br>DM<br># |




เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Special instructions (continued)

| Name<br>mnemonic | Symbol | Function | Operands |
|---|---|---|---|
| 16 to 4 Encoder<br>DMPX(77) | DMPX(77)<br>S<br>R<br>Di | Determines position of highest ON bit in source word(s) (starting word: S) and turns ON corresponding bit(s) in result word (R). Digit designations made with Di digits (rightmost digit: first digit to receive converted value; next digit to left: number of words to be converted minus 1).<br>  | S:<br>IR<br>SR<br>HR<br>TC<br>DM<br><br>R:<br>IR<br>IR<br>HR<br>DM<br><br>Di:<br>IR<br>HR<br>TC<br>DM<br># |
| Reversible Shift Register<br>SFT(84) | SFTR(84)<br>C<br>St<br>E | Shifts data in a specified word or series of words one bit to either the left or the right. | St/E:<br>IR<br>HR<br>DM<br><br>C:<br>IR<br>HR<br>DM |
| Subroutine Entry<br>SBS(91) | SBS(91) N | Transfers control of a program over to a subroutine N. | N:<br>00 to 15 |
| Subroutine Definition Start<br>SBN(92) | SBN(92) N | Indicates the beginning of a subroutine definition. | N:<br>00 to 15 |
| Subroutine Definition End<br>RET(93) | RET(93) | Indicates the end of a subroutine definition. | None |
| I/O Refresh<br>IORF(97) | IORF(97)<br>St<br>E | Refreshes I/O words between a specified range. Refreshes words in word units. | St/E:<br>00 to 99 |

| Area | IR | SR | HR | TR | TC | DM | # |
|---|---|---|---|---|---|---|---|
| Bits/Words | 0000 to 1807 | 1808 to 1907 | HR 000 to 915 | TR 0 to 7 | TC 00 to 47 | Dm 00 to DM 63 | 0000 to 9999<br>or 0000 to FFFF |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 10. Glossary

**address** – The location in memory where data is stored. For data areas, an address consists of a two-letter data area designation and a number that designates the word and/or bit location. For the UM area, an address designates the instruction location (UM area); for the FM area, the block location (FM area), etc.

**allocation** – The process by which the PC assigns certain bits or words in memory for various functions. This includes pairing I/O bits to I/O points on units.

**Analogue Timer Unit** – A dedicated timer that interfaces through analogue signal externally and digital signals internally.

**AND** – A logic operation whereby the result is true if and only if both premises are true. In ladder-diagram programming the premises are usually ON/OFF states of bits or the logical combination of such states called execution conditions.

**BCD** – An acronym for binary-coded decimal.

**BCD calculation** – An arithmetic calculation that uses numbers expressed in binary-coded decimal.

**binary** – A number system where all numbers are expressed to the base 2. Although in a PC all data is ultimately stored in binary form, binary is used to refer to data that is numerically equivalent to the binary value. It is not used to refer to binary-coded decimal. Each four binary bits is equivalent to one hexadecimal digit.

**binary-coded decimal** – A system used to represent numbers so that each four binary bits is numerically equivalent to one decimal digit.

**bit** – The smallest unit of storage in a PC. The status of a bit is either ON or OFF. Four bits equal one digit, sixteen bits, one word. Different bits are allocated to special purposes, such as holding the status input from external devices, while other bits are available for general use in programming.

**bit address** – The location in memory where a bit of data is stored. A bit address must specify (sometimes by default) the data area and word that is being addressed as well as the number of the bit.

**bit designator** – An operand that is used to designate the bit or bits of a word to be used by an instruction.

**bit number** – A number that indicates the location of a bit within a word. Bit 00 is the rightmost (least significant) bit; bit 15 is the leftmost (most significant) bit.

**buffer** – A temporary storage space for data in a computerised device.

**bus bar** – The line leading down the left and sometimes right side of a ladder diagram. Instruction execution follows down the bus bar, which is the starting point for all instruction lines.

**call** – A process by which instruction execution shifts from the main program to a subroutine. The subroutine may be called by an instruction or by an interrupt.

**carry flag** – A flag that is used with arithmetic operations to hold a carry from an addition or multiplication operation or to indicate that the result is negative in a subtraction operation. The carry flag is also used with certain types of shift operation.

**clock pulse** – A pulse available at a certain bit in memory for use in timing operations. Various clock pulses are available with different pulse widths.

**clock pulse bit** – A bit in memory that supplies a pulse that can be used to time operations. Various clock pulse bits are available with different pulse widths.

**condition** – An instruction placed along an instruction line to determine how terminal instructions on the right side are to be executed. Each condition is assigned to a bit in memory that determines its status. The status of the bit assigned to each condition determines, in turn, the execution condition for each instruction up to a terminal instruction on the right side of the ladder diagram.

**constant** – An operand for which the actual numeric value is input directly and in place of a data memory address would hold the value to be used.

**control bit** – A bit in a memory area that is set either from the program or from a programming device to achieve a specific purpose, eg., a restart bit is turned ON and OFF to restart a Unit.

**Control System** – All of the hardware and software components used to control other devices. A Control System includes the PC system, the PC programs, and all I/O devices that are used to control or obtain feedback from the controlled system.

**controlled system** – The devices that are being controlled by a PC system.

**control signal** – A signal sent from the PC to affect the operation of the controlled system.

**counter** – Either a dedicated number of digits or words in memory used to count the number of times a specific process has occurred or a location in memory accessed through a TC bit and used to count the number of times the status of a bit or an execution condition has changed from OFF to ON.

**CPU** – An acronym for central processing unit. In a PC system, the CPU executes the program, processes I/O signals, communicates with external devices, etc.

**CPU Unit** – the CPU Unit contains the CPU and provides a certain number of I/O points.

**data area** – An area in the PC's memory that is designed to hold a specific type of data, eg., the SR area is designed to hold flags and control bits. Memory areas that hold programs are not considered data areas.

**data area boundary** – The highest address available in a data area. When designating an operand that requires multiple words, it is necessary that the highest address in the data area is not exceeded.

**debug** – A process by which a draft program is corrected until it operates as intended. Debugging includes both removal of syntax errors as well as fine-tuning of timing and co-ordination of control operations.

**decimal** – A number system where all numbers are expressed to the base 10. Although in a PC all data is ultimately stored in binary form, four binary bits are often used to represent one decimal digit, a system called binary-coded decimal.

**decrement** – Decreasing a numeric value by 1.

**default** – A value assumed and automatically set by the PC when a specific value is not input by the user.

**definer** – A number used as an operand for an instruction but that serves to define the instruction itself rather than the data on which the instruction is to operate. Definers include jump numbers, subroutine numbers, etc.

**delay** – In tracing, a value that specifies where tracing to begin in relationship to the trigger. A delay can be either positive or negative, ie., can designate an off- set on either side of the trigger.

**destination** – The location where data of some sort in an instruction is to be placed as opposed to the location from which data is to be taken for use in the instruction. The location from which data is to be taken is called the "source".

เอกสารนี้ใช้เพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**differentiation instruction** – An instruction used to ensure that the operand bit is never turned ON for more than one scan after the execution condition goes either from OFF to ON for a differentiate up instruction or from ON to OFF for a differentiate down instruction.

**digit** – A unit of storage in memory that consists of four bits.

**digit designator** – An operand that is used to designate the digit or digits of a word to be used by an instruction.

**distributed control** – An automation concept in which control of each portion of an automated system is located near the devices actually being controlled, ie., control is decentralised and 'distributed' over the system. Distributed control is a concept basic to PC systems.

**DM area** – A data area used to hold word data. A word in the DM area cannot be accessed by bit.

**download** – The process of transferring a program or data from a higher-level computer to a lower-level computer or PC.

**electrical noise** – Electrical 'static' that can disturb electronic communications. The 'snow' that can appear on a TV screen is an example of the effects of electrical noise.

**error code** – A numeric code output to indicate the existence of and something about the nature of an error. Some error codes are generated by the system, others are defined in the program by the operator.

**execution condition** – The ON or OFF status under which an instruction is executed. The execution condition is determined by the logical combination of conditions on the same instruction line and up to the instruction being executed.

**execution time** – The time required for the CPU to execute either an individual instruction or an entire program.

**Expansion I/O Unit** – An Expansion I/O Unit is connected to increase the number of I/O points available.

**extended counter** – A counter created in a program that counts higher than any of the standard counters provided by the individual instructions.

**extended timer** – A timer created in a program that times longer than any of the standard timers provided by the individual instructions.

**fatal error** – An error that will stop PC operation and require correction before operation can be continued.

**flag** – A dedicated bit in memory that is set by the system to indicate some type of operating status. Some flags, such as the carry flag, can also be set by the operator or program.

**flicker bit** – A bit that is programmed to turn ON and OFF at a specific interval.

**force reset** – The process of artificially turning OFF a bit from a programming device. Bits are usually turned OFF as a result of program execution.

**force set** – The process of artificially turning ON a bit from a programming device. Bits are usually turned ON as a result of program execution.

**function code** – A two-digit number used to input an instruction into the PC.

**GPC** – An acronym for Graphic Programming Console.

**Graphic Programming Console** – A programming device provided with advanced programming and debugging capabilities to facilitate PC operation. A Graphic Programming Console is provided with a large display onto which ladder-diagram programs can be written directly in ladder-diagram symbols for input into the PC without conversion to mnemonic form.

**hardware error** – An error originating in the hardware structure of the PC, as opposed to a software error, which ordinates in software (ie., programs).

**hexadecimal** – A number system where all numbers are expressed to the base 16. Although in a PC all data is ultimately stored in binary form, displays on and inputs through programming devices are often expressed in hexadecimal to facilitate operation. Each four binary bits is numerically equivalent to one hexadecimal digit.

**HR area** – A data area used to store and manipulate data and to preserve data when power to the PC is turned OFF.

**I/O capacity** – The number of inputs and outputs that a PC is able to handle. This number ranges from around one-hundred for smaller PCs to two thousand for the largest ones.

**I/O devices** – The devices to which terminals on I/O Units, Special I/O Units, or Intelligent I/O Units are connected. I/O devices may be either part of the control system, if they function to help control other devices, or they may be part of the controlled system.

**I/O point** – The place at which an input signal enters the PC system or an output signal leaves the PC system. In physical terms, an I/O point corresponds to terminals or connector pins on a Unit; in terms of programming, an I/O point corresponds to an I/O bit in the IR area.

**I/O response time** – The time required for an output signal to be sent from the PC in response to an input signal received from an external device.

**I/O unit** – The most basic type of Unit mounted to a backplane to create a Rack. I/O Units include Input Units and output Units, each of which is available in a range of specifications. I/O Units do not include special I/O Units, link Units, etc.

**I/O word** – A word in the IR area that is allocated to a Unit in the PC system.

**increment** – Increasing a numeric value by 1.

**initialisation error** – An error that occurs either in hardware or software before the PC system has actually begun operation, ie., during initialisation.

**initialise** – Part of the startup process whereby some memory areas are cleared, system setup is checked, and default values are set.

**input** – The signal coming from an external device into the PC. Input often is used abstractly or collectively to refer to incoming signals.

**input bit** – A bit in the IR area that is allocated to hold the status of an input.

**input device** – An external device that sends signal(s) into the PC system.

**input point** – The point at which an input enters the PC system. An input point physically corresponds to terminals or connector pin(s).

**input signal** – A change in the status of a connection entering the PC. Generally an input signal is said to exist when, for example, a connection point goes from low to high voltage or from a nonconductive to a conductive state.

**instruction** – A direction given in the program that tells the PC an action to be carried out and the data to be used in carrying out the action. Instructions can simply turn a bit ON or OFF, or they can perform much more complex actions, such as converting and/or transferring large blocks of data.

เอกสารนี้ใช้เพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**instruction block** – A group of instructions that is logically related in a ladder-diagram program. Although any logically related group of instructions could be called an instruction block, the term is generally used to refer to blocks of instructions called logic blocks that require logic block instructions to relate them to other instructions or logic blocks.

**instruction execution time** – The time required to execute an instruction. The execution time for any one instruction can vary with the execution condition for the instruction and the operands used in it.

**instruction line** – A group of conditions that lies together on the same horizontal line of a ladder diagram. Instruction lines can branch apart or pin together to form instruction blocks.

**interface** – An interface is the conceptual boundary between systems or devices and usually involves changes in the way the communicated data is represented. Interface devices perform operations such as changing the coding, format, or speed of the data.

**interlock** – A programming method used to treat a number of instructions as a group so that the entire group can be reset together when individual execution is not required. An interlocked program section is executed normally for an ON execution condition and partially reset for an OFF execution condition.

**inverse condition** – A condition that produces an ON execution condition when the bit assigned to it is OFF and an OFF execution condition when the bit assigned to it is ON.

**IR area** – A data area whose principal function is to hold the status of inputs coming into the system and outputs that are to be set out of the system. Bits and words in the IR that are used this way are called I/O bits and I/O words. The remaining bits in the IR area are work bits.

**jump** – A type of programming where execution moves directly from one point in a program to a separate point in the program without sequentially executing the instruction in between. Jumps are usually conditional on an execution condition.

**jump number** – A definer used with a jump that defines the points from which and to which a jump is to be made.

**ladder diagram (program)** – A form of program arising out of relay-based control systems that uses circuit type diagrams to represent the logic flow of programming instructions. The appearance of the program suggests a ladder, and thus the name.

**ladder diagram symbol** – A symbol used in a ladder diagram program.

**ladder instruction** – An instruction that represents the 'rung' portion of a ladder-diagram program. The other instructions in a ladder diagram fall along the right side of the diagram and are called terminal instructions.

**leftmost (bit/word)** – The highest numbered bits of a group of bits, generally of an entire word, or the highest numbered words of a group of words. These bits/words are often called most significant bits/words.

**link** – A hardware or software connection formed between two Units. 'Link' can refer either to a part of the physical connection between two Units or a software connection created to data existing at another location (I/O links).

**load** – The processes of copying data either from an external device or from a storage area to an active portion of the system such as a display buffer. Also, an output device connected to the PC is called a load.

**logic block** – A group of instructions that is logically related in a ladder-diagram program and that requires logic block instructions to relate it to other instructions or logic blocks.

**logic block instruction** – An instruction used to locally combine the execution condition resulting from a logic block with a current execution condition. The current execution condition could be the result of a single condition or of another logic block. AND Load and OR Load are the two logic block instructions.

**LR area** – A data area that is used in a PC link system so that data can be transferred between two or more PCs. If a PC link system is not used, the LR area is available for use as work bits.

**main program** – All of a program except for the subroutines.

**memory area** – Any of the areas in the PC used to hold data or programs.

**mnemonic code** – A form of a ladder-diagram program that consists of a sequential list of the instructions without using a ladder diagram. Mnemonic code is required to input a program into a PC when using a programming console.

**MONITOR mode** – A mode of PC operation in which normal program execution is possible but in which modification of data held in memory is still possible. Used for monitoring or debugging the PC.

**NC input** – An input that is normally closed, ie., the input signal is considered to be present when the circuit connected to the input opens.

**nest** – Programming one jump within another jump, programming a call to a subroutine from within another subroutine, etc.

**NO input** – An input that is normally open, ie., the input signal is considered to be present when the circuit connected to the input closes.

**noise interference** – Disturbances in signals caused by electrical noise.

**nonfatal error** – A hardware or software error that produces a warning but does not stop the PC from operating.

**normal condition** – A condition that produces an ON execution condition when the bit assigned to it is ON and an OFF execution condition when the bit assigned to it is OFF.

**NOT** – A logic operation which inverts the status of the operand. For example, AND NOT indicates an AND with the opposite of the actual status of the operand bit.

**OFF** – The status of an input or output when a signal is said not to be present. The OFF state is generally low voltage or non-conductivity, but can be defined as the opposite of either.

**OFF delay** – The delay produced between the time turning OFF a signal is initiated (eg., by an input device or PC) and the time the signal reaches a state readable as an OFF signal (ie., as no signal) by a receiving party (eg., output device or PC).

**ON** – The status of an input or output when a signal is said to be present. The ON state is generally high voltage or conductivity, but can be defined as the opposite of either.

**ON delay** – The delay produced between the time a signal is initiated (eg., by an input device or PC) and the time the signal reaches a state readable as an ON signal by a receiving party (eg., output device or PC).

**one-shot bit** – A bit that is turned ON or OFF for a specified interval of time longer than one scan.

เอกสารนี้จัดทำขึ้นเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**operand** – A bit(s) or word(s) designated as the data to be used for an instruction. An operand can be input as a constant expressing the actual numeric value to be used or as an address to express the location in memory of the data to be used.

**operand bit** – A bit designated as an operand for an instruction.

**operand word** – A word designated as an operand for an instruction.

**operating error** – An error that occurs during actual PC operation as opposed to an initialisation error, which occurs before actual operations can begin.

**OR** – A logic operation whereby the result is true if either one or both of the premises is true. In ladder-diagram programming the premises are usually ON/OFF states of bits or the logical combination of such states called execution conditions.

**output** – The signal sent from the PC to an external device. Output often is used abstractly or collectively to refer to outgoing signals.

**output bit** – A bit in the IR area that is allocated to hold the status to be sent to an output device.

**output device** – An external device that receives a signal(s) from the PC system.

**output point** – The point at which an output leaves the PC system. An output point physically corresponds to terminals or connector pin(s).

**output signal** – A change in the status of a connection leaving the PC. Generally an output signal is said to exist when, for example, a connection point goes from low to high voltage or from a nonconductive to a conductive state.

**overseeing** – Part of the processing performed by the CPU that includes general tasks required to operate the PC.

**overwrite** – Changing the content of a memory location so that the previous content is lost.

**PC** – An acronym for programmable controller.

**PC system** – All of the Units connected to the CPU Unit up to, but not including the I/O devices. The limits of the PC System on the upper end is the PC and the program in its CPU and on the lower end, I/O Units, etc.

**PCB** – An acronym for printed circuit board.

**Peripheral Device** – Devices connected to a PC system to aid in system operation. Peripheral devices include printers, programming devices, external storage media, etc.

**present value** – The current time left on a timer or the current count of a counter. Present value is abbreviated PV.

**printed circuit board** – A board onto which electrical circuits are printed for mounting into a computer or electrical device.

**program** – The list of instructions that tells the PC the sequence of control actions to be carried out.

**Programmable Controller** – A computerised device that can accept inputs from external devices and generate outputs to external devices according to a program held in memory. Programmable controllers are used to automate control of external devices. Often referred to as Programmable Logic Controller (PLCs).

**programmed alarm** – An alarm given as a result of execution of an instruction designed to generate the alarm in the program as opposed to one generated by the system.

**programmed error** – An error arising as a result of execution of an instruction designed to generate the error in the program as opposed to one generated by the system.

**programmed message** – A message generated as a result of execution of an instruction designed to generate the message in the program as opposed to one generated by the system.

**Programming Console** – The simplest form or programming device available for a PC. Programming consoles are available both as hand-held models and as CPU-mounting models.

**Programming Device** – A peripheral device used to input a program into a PC or to alter or monitor a program already held in the PC. There are dedicated programming devices, such as programming consoles, and there are non-dedicated devices, such as a host computer.

**PROGRAM mode** – A mode of operation that allows for inputting and debugging programs but that does not permit normal execution of the program.

**PROM Writer** – A peripheral device used to write programs and other data into a ROM for permanent storage and application.

**prompt** – A message or symbol that appears on a display to request input from the operator.

**PV** – An acronym for present value.

**refresh** – The process of updating output status sent to external devices so that it agrees with the status of output bits held in memory and of updating input bits in memory so that they agree with the status of inputs from external devices.

**relay-based control** – The forerunner of PCs. In relay-based control, groups of relays are wired to each other to form control circuits. In a PC, these are replaced by programmed circuits.

**reset** – The process of turning a bit or signal OFF or of changing the present value of a timer or counter to its set value or to zero.

**return** – The process by which instruction execution shifts from a subroutine back to the point from which the subroutine was called. A return is automatic upon completion of the subroutine.

**reversible counter** – A counter that can be both incremented and decremented depending on a specified condition(s).

**reversible shift register** – A shift register that can shift data in either direction depending on a specified condition(s).

**right-hand instruction** – Another term for terminal instruction.

**rightmost (bit/word)** – The lowest numbered bits of a group of bits, generally of an entire word or the lowest numbered words of a group of words. These bits/words are often called least significant bits/words.

**RUN mode** – The operating mode used by the PC for normal control operations.

**scan** – The process used to execute a ladder-diagram program. The program is examined sequentially from start to finish and each instruction is executed in turn based on execution conditions.

**scan time** – The time required for a single scan of the ladder-diagram program.

**self diagnosis** – A process whereby the system checks its own operation and generates a warning or error if an abnormality is discovered.

**self-maintaining bit** – A bit that is programmed to maintain either an OFF or ON status until set or reset by a specific condition(s) different from the one that originally caused the bit to turn OFF or ON.

**servicing** – The process whereby the PC provides data to or receives data from external devices.

เอกสารนี้ใช้เพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**set** – The process of turning a bit or signal ON.

**set value** – The count from which a counter starts counting down (or, in the case of a reversible counter, the maximum count) or the time from which a timer starts timing. Set value is abbreviated SV.

**shift register** – One or more words in which data is shifted in bit, digit, or word units a specified number of units to the right or left.

**software error** – An error that occurs in the execution of a program.

**software protects** – A software means of protecting data from being changed as opposed to a physical switch or other hardware setting.

**source** – The location from which data is taken for use in an instruction as opposed to the location to which the result of an instruction is to be written. The location to which the result of an instruction is to be written is called the destination.

**SR area** – A data area in a PC used mainly for flags, control bits, and other information provided about PC operation. The status of only certain SR bits may be controlled by the operator, ie., most SR bits can only be read.

**subroutine** – A group of instructions placed after the main program and executed only if called from the main program or activated by an interrupt.

**subroutine number** – A definer used to identify the subroutine that a subroutine call or interrupt activates.

**SV** – An acronym for set value.

**switching capacity** – The voltage/current that a relay can switch on and off.

**syntax error** – An error in the way in which a program is written. Syntax errors can include 'spelling' mistakes (ie., a function code that does not exist), mistakes in specifying operands within acceptable parameters (eg., specifying unwritable SR bits as a destination), and mistakes in actual application of instructions (eg., a call to a subroutine that does not exist).

**system error** – An error generated by the system as opposed to one resulting from execution of an instruction designed to generate an error.

**system error message** – An error message generated by the system as opposed to one resulting from execution of an instruction designed to generate a message.

**TC area** – A data area that can be used only for timers and counters. Each bit in the TC area serves as the access point for the SV, PV, and completion flag for the timer or counter defined with that bit.

**TC number** – A number that corresponds to a bit in the TC area and used to define the bit as either a timer or a counter.

**terminal instruction** – An instruction placed on the right side of a ladder diagram that uses the final execution condition on an instruction line(s).

**timer** – A location in memory accessed through a TC bit and used to time down from the timer's set value. Timers are turned ON and reset according to their execution conditions.

**TM area** – A memory area used to store the results of a trace.

**transmission distance** – The distance that a signal can be transmitted.

**TR area** – A data area used to store execution conditions so that they can be reloaded later for use with other instructions.

**transfer** – The process of moving data from one location to another within the PC or between the PC and external devices. When data is transferred, generally a copy of the data is sent to the destination, ie., the content of the source of the transfer is not changed.

**UM area** – The memory area used to hold the active program, ie., the program that is being currently executed.

**Unit** – In OMRON PC terminology, the word Unit is capitalised to indicate any product sold for a PC system. Though most of the names of these products end with the word Unit, not all do, eg., a remote terminal is referred to in a collective sense as a Unit. Context generally makes any limitations of this word clear.

**unit number** – A number assigned to special I/O Units to assign words and sometimes other operating parameters to it.

**watchdog timer** – A timer within the system that ensures that the scan time stays within specified limits. When limits are reached, either warnings are given or PC operation is stopped depending on the particular limit that is reached. Although a default value of 130ms is automatically set for the basic time limit, this value can be extended by the program.

**word** – A unit of storage in memory that consists of 16 bits. All data areas consist of words. Some data areas can be accessed by words; others, by either words or bits.

**word address** – The location in memory where a word or data is stored. A word address must specify (sometimes by default) the data area and the number of the word that is being addressed.

**work bit** – A bit in a work word.

**work word** – A word that can be used for data calculation or other manipulation in programming, ie., a 'work space' in memory. A large portion of the IR area is always reserved for work words. Parts of other areas not required for special purposes may also be used as work words, eg. I/O words not allocated to I/O Units.

The information provided in RS technical literature is believed to be accurate and reliable, however, RS Components assumes no responsibility for inaccuracies or omissions, or for the use of this information, and all use of such information shall be entirely at the user's own risk.
No responsibility is assumed by RS Components for any infringements of patents or other rights of third parties which may result from its use.
Specifications as given in RS Components technical literature are subject to change without notice.



เอกสารนี้ใช้ได้ ทั้งสิ้น ยกเว้นการนำไปใช้ในเชิงพาณิชย์ โดยไม่ได้รับอนุญาตเป็นลายลักษณ์อักษรจากเจ้าของลิขสิทธิ์ ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ทั้งสิ้น ยกเว้นการนำไปใช้ในเชิงพาณิชย์ โดยไม่ได้รับอนุญาตเป็นลายลักษณ์อักษรจากเจ้าของลิขสิทธิ์ ไม่อนุญาตให้นำไปใช้