การควบคุมหน่วยการผลิตอัตโนมัติขนาดเล็กและกลางผ่านเครือข่าย

นายราชิต ลิมป์จันทรา

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต
สาขาวิชาวิศวกรรมเครื่องกล ภาควิชาวิศวกรรมเครื่องกล
คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย
ปีการศึกษา 2542

ISBN 974-334-634-1

ลิขสิทธิ์ของจุฬาลงกรณ์มหาวิทยาลัย

I19282795

SMALL AND MEDIUM SIZE AUTOMATION-CELL CONTROL THROUGH A NETWORK

Mr. Rachit Limchantra

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

A Thesis Submitted in Partial Fulfillment of the Requirements
for the Degree of Master of Engineering in Mechanical Engineering
Department of Mechanical Engineering
Faculty of Engineering
Chulalongkorn University
Academic Year 1999

ISBN 974-334-634-1

หัวข้อวิทยานิพนธ์	การควบคุมหน่วยการผลิตอัตโนมัติขนาดเล็กและกลางผ่านเครือข่าย
โดย	นาย ราชิต ลิมป์จันทรา
ภาควิชา	วิศวกรรมเครื่องกล
อาจารย์ที่ปรึกษา	รองศาสตราจารย์ ดร. วิบูลย์ แสงวีระพันธุ์ศิริ

คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย อนุมัติให้นับวิทยานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญามหาบัณฑิต

..... คณบดีคณะวิศวกรรมศาสตร์
(ศาสตราจารย์ ดร. สมศักดิ์ ปัญญาแก้ว)

คณะกรรมการสอบวิทยานิพนธ์

..... ประธานกรรมการ
(รองศาสตราจารย์ ดร. อิทธิพล ปานงาม)

..... อาจารย์ที่ปรึกษา
(รองศาสตราจารย์ ดร. วิบูลย์ แสงวีระพันธุ์ศิริ)

..... กรรมการ
(รองศาสตราจารย์ ดร. วิทยา ยงเจริญ)

..... กรรมการ
(ผู้ช่วยศาสตราจารย์ ดร. ชัยโรจน์ คุณพนิชกิจ)

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

ราชิต ลิมป์จันทรา : การควบคุมหน่วยการผลิตอัตโนมัติขนาดเล็กและกลางผ่านเครือข่าย. (SMALL AND MEDIUM SIZE AUTOMATION-CELL CONTROL THROUGH A NETWORK) อ.ที่ปรึกษา : รศ. ดร. วิบูลย์ แสงวีระพันธุ์ศิริ, 90 หน้า.
ISBN 974-334-634-1.

วิทยานิพนธ์ฉบับนี้ แสดงถึงการนำเอาระบบเครือข่ายการควบคุมมาใช้กับระบบการผลิตอัตโนมัติแบบโมดูล ซึ่งสามารถปรับเปลี่ยนรูปแบบของสายการผลิตได้หลากหลายรูปแบบ ระบบเครือข่ายที่นำมาใช้จะครอบคลุมทุกระดับของการสื่อสาร ตั้งแต่ระดับล่างสุดเป็นเครือข่ายของอุปกรณ์ตรวจรู้ (Sensor) และอุปกรณ์ขับ (Actuator) ระดับกลางเป็นเครือข่ายของพีแอลซี (PLC) และระดับบนสุดเป็นเครือข่ายโรงงาน นอกจากนั้นยังได้พัฒนาโปรแกรมสำหรับตรวจดู (Monitor) สายการผลิตในรูปแบบของรูปภาพสามารถแสดงการเคลื่อนไหวได้ตามการการทำงานของสายการผลิตจริง

ผลที่ได้จากการวิจัย แสดงให้เห็นถึงความยืดหยุ่น (Flexibility) ของสายการผลิต สามารถปรับเปลี่ยนรูปแบบของสายการผลิตได้ง่าย และรวดเร็ว รวมทั้งสามารถตรวจดูการทำงาน และข้อมูลชิ้นงานที่ผลิตได้จากทุกที่ภายในโรงงาน ซึ่งเป็นการเพิ่มความสามารถในการแข่งขันในอุตสาหกรรมการผลิตปัจจุบัน

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

ภาควิชา วิศวกรรมเครื่องกล	ลายมือชื่อนิสิต ราชิต ลิมป์จันทรา
สาขาวิชา วิศวกรรมเครื่องกล	ลายมือชื่ออาจารย์ที่ปรึกษา
ปีการศึกษา 2542	ลายมือชื่ออาจารย์ที่ปรึกษาร่วม

4070393221 : MAJOR MECHANICAL ENGINEERING
KEY WORD: AUTOMATION / NETWORK / PLC / FIELDBUS / HUMAN MACHINE INTERFACE

RACHIT LIMCHANTRA : SMALL AND MEDIUM SIZE AUTOMATION-CELL CONTROL THROUGH A NETWORK. THESIS ADVISOR : ASSO. PROF. VIBOON SANGVERAPHUNSIRI, Ph.D. 90 pp. ISBN 974-334-634-1.

This thesis studies the advantage of using control network in modular production system that can be adjusted or changed the combination of individual modules. This network includes all level of communication between software and hardware components. Low level is sensor and actuator network, middle level is PLC network and top level is factory network. The research shows development of monitor program of the production line. The monitor program will obtain the sensor and actuator status through high level network and link the information with graphic objects for displaying the animated result on the client screen.

The research also shown, when incorporation the high-level network to the controller, we can improve the flexibility of production line. The production line can be adjusted to a new possible combination with simpler and faster. Furthermore, Operators can real-time monitor the product and status of production line from everywhere in the factory through network client.

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

ภาควิชา วิศวกรรมเครื่องกล	ลายมือชื่อนิสิต .....
สาขาวิชา วิศวกรรมเครื่องกล	ลายมือชื่ออาจารย์ที่ปรึกษา .....
ปีการศึกษา 2542	ลายมือชื่ออาจารย์ที่ปรึกษาร่วม .....

หอสมุดกลาง สถาบันวิทยบริการ จุฬาลงกรณ์มหาวิทยาลัย

กิตติกรรมประกาศ

วิทยานิพนธ์ฉบับนี้สำเร็จลุล่วงไปได้ด้วยความช่วยเหลืออย่างดียิ่งของ รศ.ดร. วิบูลย์ แสงวีระพันธุ์ศิริ อาจารย์ที่ปรึกษาวิทยานิพนธ์ ซึ่งท่านได้เสียสละเวลาคอยให้คำแนะนำปรึกษา และให้ข้อคิดเห็นต่าง ๆ ที่ใช้ในการทำวิจัยครั้งนี้ พร้อมทั้งจัดหาอุปกรณ์ที่ใช้ในการทำวิจัย วิทยานิพนธ์นี้มาด้วยดีโดยตลอด ขอขอบคุณบัณฑิตวิทยาลัยที่ได้ให้ทุนอุดหนุนการวิจัย และขอบใจเพื่อน ๆ พี่ ๆ นิสิตปริญญาโทด้วยกันที่ได้ให้ข้อคิดเห็นและกำลังใจตลอดมา

ท้ายนี้ ผู้วิจัยใคร่ขอกราบขอบพระคุณ บิดา-มารดา ซึ่งสนับสนุนทางด้านทุนทรัพย์และให้กำลังใจแก่ผู้วิจัยเสมอมาจนสำเร็จการศึกษา

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

สารบัญ

หน้า

บทคัดย่อภาษาไทย ..... ง

บทคัดย่อภาษาอังกฤษ ..... จ

กิตติกรรมประกาศ ..... ฉ

สารบัญ ..... ช

สารบัญตาราง ..... ฌ

สารบัญภาพ ..... ญ

บทที่

1. บทนำ 1

1.1 ความสำคัญและที่มาของวิทยานิพนธ์ 1

1.2 วัตถุประสงค์ของวิทยานิพนธ์ 1

1.3 ขอบเขตของวิทยานิพนธ์ 1

1.4 ขั้นตอนในการทำวิทยานิพนธ์ 2

1.5 ประโยชน์ที่คาดว่าจะได้รับจากการทำวิทยานิพนธ์ 2

2. ระบบการผลิตอัตโนมัติ 3

2.1 ระบบการผลิตแบบโมดูลาร์ 3

2.2 อุปกรณ์และเซนเซอร์ที่ใช้ในโมดูลต่าง ๆ 5

2.3 ชิ้นงานที่ใช้ผลิตในระบบการผลิตแบบโมดูลาร์ 9

2.4 โปรแกรมเมเบิลลอจิกคอนโทรลเลอร์ 11

2.5 PLC ของ GE FANUC ที่นำมาใช้กับโมดูลการผลิตต่าง ๆ 15

3. เครือข่ายการสื่อสารของระบบการผลิตอัตโนมัติ 19

3.1 เครือข่ายการสื่อสารระดับต่าง ๆ ในโรงงานอัตโนมัติ 19

3.2 เทคโนโลยีฟิลด์บัส 19

3.3 การสื่อสารอนุกรม 21

3.4 อีเทอร์เน็ต 25

3.5 โปรโตคอลจีเนียสบัส 26

3.6 โปรโตคอล SNP/SNP-X 30

3.7 โปรโตคอล TCP/IP 33

3.8 ซอฟต์แวร์ CIMPLICITY HMI 34

4. การเขียนโปรแกรมการทำงานของ 7 สถานี 35

4.1 การเขียนโปรแกรมด้วยภาษาแลดเดอร์ 35

4.2 โปรแกรมการทำงานของสถานีจ่ายชิ้นงาน 36

4.3 โปรแกรมการทำงานของสถานีทดสอบชิ้นงาน 37

บทที่ หน้า

4.4 โปรแกรมการทำงานของสถานีควบคุมคุณภาพ 38
4.5 โปรแกรมการทำงานของสถานีพักชิ้นงาน 40
4.6 โปรแกรมการทำงานของสถานีผลิตชิ้นงาน 41
4.7 โปรแกรมการทำงานของสถานีแขนกลเพื่อเคลื่อนย้ายหรือเก็บชิ้นงาน 42
4.8 โปรแกรมการทำงานของสถานีแยกชิ้นงาน 44
5. การจัดเครือข่ายการควบคุมในระบบการผลิตอัตโนมัติแบบ 7 โมดูล 46
5.1 การจัดเครือข่ายการสื่อสาร เมื่อใช้ 7 โมดูล 46
5.2 การรับ-ส่งข้อมูลระหว่างสถานี 46
5.3 การเขียนโปรแกรมรับ-ส่งข้อมูล 49
5.4 ลักษณะการทำงานของสายการผลิต 50
6. การพัฒนาโปรแกรมสำหรับการตรวจดูระบบโดยใช้ซอฟต์แวร์ CIMPLICITY HMI 52
7. การทดลองปรับเปลี่ยนสายการผลิตในอีก 6 รูปแบบ 63
7.1 การปรับเปลี่ยนรูปแบบของสายการผลิต 63
7.2 เครือข่ายการควบคุมในสายการผลิตอีก 6 รูปแบบ 63
8. สรุปผลและข้อเสนอแนะ 75
สรุปผลการทำวิทยานิพนธ์ 75
ข้อเสนอแนะ 76
รายการอ้างอิง 77
ภาคผนวก 78
ภาคผนวก ก โปรแกรมการทำงานของสถานีจ่ายชิ้นงาน 79
ภาคผนวก ข โปรแกรมสำหรับรับ-ส่งข้อมูลระหว่างสถานีต่าง ๆ 88
ประวัติผู้เขียน 90

สารบัญตาราง

ตารางที่ หน้า

2.1 แสดงข้อมูลบิต D0 10

2.2 แสดงข้อมูลบิต D1 และ D2 10

2.3 แสดงข้อมูลบิต D3 และ D4 10

2.4 แสดงข้อมูลบิต D5 10

2.5 แสดงตัวอย่างการเขียนโปรแกรมคำสั่งบูลลีน 14

3.1 แสดงมาตรฐานของหัวต่อ RS-232 25

4.1 แสดงการตรวจสอบชิ้นงาน โดยเซนเซอร์ทั้ง 3 ตัว 38

5.1 แสดงข้อมูลอินพุตขนาด 16 บิต ของสถานีต่าง ๆ 48

5.2 แสดงข้อมูลเอาต์พุตขนาด 16 บิต ของสถานีต่าง ๆ 48

5.3 แสดงหน่วยความจำ %G ที่ใช้เก็บค่าอินพุตและเอาต์พุตของสถานีต่างๆ 49

สารบัญภาพ

รูปที่ หน้า

2.1 แสดงสายการผลิตในรูปแบบต่าง ๆ 4
2.2 แสดงสายการผลิตที่ประกอบด้วย 7 สถานี 4
2.3 แสดงเครื่องอัดลม 6
2.4 แสดงชุดปรับปรุงคุณภาพลมอัด 6
2.5 แสดงโซลินอยด์วาล์ว 7
2.6 แสดงกระบอกสูบลม 7
2.7 แสดง Semi-rotary actuator 7
2.8 แสดง Pneumatic / Electronic Switch 7
2.9 แสดง Suction cup 8
2.10 แสดง Position sensor 8
2.11 แสดง Inductive, Capacitive และ Optical proximity sensor ในสถานีทดสอบชิ้นงาน 8
2.12 แสดงชิ้นงานในแบบต่าง ๆ 9
2.13 แสดงโครงสร้างของ PLC 11
2.14 แสดงตัวอย่างการเขียนโปรแกรมภาษาแลดเดอร์ 13
2.15 แสดงตัวอย่างภาษา SFC 14
2.16 แสดง PLC รุ่น 90-30 15
2.17 แสดง PLC รุ่น 90-70 16
2.18 แสดง 28-Point Micro PLC 17
2.19 แสดง Field Control 18
3.1 แสดงเครือข่ายการสื่อสารระดับต่าง ๆ ในโรงงานอัตโนมัติ 20
3.2 แสดงเลเยอร์ของสถาปัตยกรรม ISO/OSI Communication Model และ Fieldbus 21
3.3 แสดงรหัส ASCII ที่ใช้แทนตัวอักษรหนึ่งตัว 23
3.4 แสดงสัญญาณทางไฟฟ้าของการส่งข้อมูลแบบไม่ต่อเนื่อง 23
3.5 แสดงค่าของบิตพาริตี้ ในแบบ odd และ even 23
3.6 แสดงการคำนวณและใส่ค่า LRC 23
3.7 แสดงการรับ-ส่งข้อมูลระหว่างอุปกรณ์ด้วยมาตรฐาน RS-232 (ไม่ใช้ Flow control) 26
3.8 แสดงโครงสร้างของเครือข่ายอีเทอร์เน็ต 26
3.9 แสดงการใช้งาน Genius Bus 28
3.10 แสดงวิธีการสื่อสารโดยใช้ Token passing 28
3.11 แสดงการส่งค่าอินพุตของ I/O Block 29
3.12 แสดงการส่งค่าเอาต์พุตจาก CPU 29

สารบัญภาพ (ต่อ)

รูปที่ หน้า

3.13 แสดงบัสสำหรับสื่อสาร และบัสสำหรับควบคุม I/O 29
3.14 แสดงการสร้างส่วนติดต่อสื่อสารระหว่าง Master กับ Slave ของ SNP 31
3.15 แสดงการส่งการร้องขอข้อมูลจาก Master และ Slave ส่งข้อมูลตอบกลับของ SNP ... 31
3.16 แสดงการสร้างส่วนติดต่อสื่อสารของ SNP-X 32
3.17 แสดงส่วนการรับ-ส่งข้อมูลของ SNP-X 32
3.18 แสดงการรับ-ส่งข้อมูลมากกว่า 2 bytes ของ SNP-X 32
3.19 แสดงการส่งข้อมูลแบบ Broadcast ไปยัง Slave ทุกตัวของ SNP-X 33
3.20 แสดงสถาปัตยกรรมของระบบโดยใช้ซอฟต์แวร์ CIMPLICITY HMI 34
4.1 แสดงตัวอย่างหน้าจอของการเขียนโปรแกรมด้วยซอฟต์แวร์ Logicmaster 35
4.2 แสดงสถานีจ่ายชิ้นงาน 36
4.3 แสดงสถานีทดสอบชิ้นงาน 38
4.4 แสดงสถานีควบคุมคุณภาพ 39
4.5 แสดงสถานีพักชิ้นงาน 40
4.6 แสดงสถานีผลิตชิ้นงาน 41
4.7 แสดงสถานีแขนกล 43
4.8 แสดงสถานีแยกชิ้นงาน 44
5.1 แสดงเครือข่ายการสื่อสารแบบใช้ 7 โมดูล 47
5.2 แสดงบิดข้อมูลที่รับ-ส่งระหว่างสถานี 47
5.3 แสดงการรับ-ส่งข้อมูลระหว่างสถานีต่าง ๆ 50
5.4 แสดงสายการผลิตอัตโนมัติ 7 โมดูล 50
6.1 ตัวอย่างหน้าจอการทำงานในส่วนของการสร้างกลุ่มรูปภาพ 52
6.2 แสดงภาพบนจอคอมพิวเตอร์ในหน้าจอหลัก 53
6.3 แสดงภาพบนจอคอมพิวเตอร์ในการตรวจดูสถานีจ่ายชิ้นงาน 54
6.4 แสดงภาพบนจอคอมพิวเตอร์ในการตรวจดูสถานีทดสอบชิ้นงาน 55
6.5 แสดงภาพบนจอคอมพิวเตอร์ในการตรวจดูสถานีควบคุมคุณภาพ 56
6.6 แสดงภาพบนจอคอมพิวเตอร์ในการตรวจดูสถานีพักชิ้นงาน 57
6.7 แสดงภาพบนจอคอมพิวเตอร์ในการตรวจดูสถานีผลิตชิ้นงาน 58
6.8 แสดงภาพบนจอคอมพิวเตอร์ในการตรวจดูสถานีแขนกล 59
6.9 แสดงภาพบนจอคอมพิวเตอร์ในการตรวจดูสถานีแยกชิ้นงาน 60
7.1 แสดงสายการผลิตในอีก 6 รูปแบบ 63
7.2 แสดงสายการผลิตแบบ 2 โมดูล 64

สารบัญภาพ (ต่อ)

รูปที่ หน้า

7.3 แสดงเครือข่ายการสื่อสารและการรับ-ส่งข้อมูลระหว่างสถานีของสายการผลิตแบบ 2 โมดูล 65

7.4 แสดงสายการผลิตแบบ 3 โมดูล 66

7.5 แสดงเครือข่ายการสื่อสารและการรับ-ส่งข้อมูลระหว่างสถานีของสายการผลิตแบบ 3 โมดูล 67

7.6 แสดงสายการผลิตแบบ 4 โมดูลต่อเป็นรูปตัว O 68

7.7 แสดงเครือข่ายการสื่อสารและการรับ-ส่งข้อมูลระหว่างสถานีของสายการผลิตแบบ 4 โมดูลต่อเป็นรูปตัว O 69

7.8 แสดงสายการผลิตแบบ 4 โมดูล ต่อเป็นรูปตัว L 68

7.9 แสดงเครือข่ายการสื่อสารและการรับ-ส่งข้อมูลระหว่างสถานีของสายการผลิตแบบ 4 โมดูล ต่อเป็นรูปตัว L 70

7.10 แสดงสายการผลิตแบบ 5 โมดูล 72

7.11 แสดงเครือข่ายการสื่อสารและการรับ-ส่งข้อมูลระหว่างสถานีของสายการผลิตแบบ 5 โมดูล 73

7.12 แสดงสายการผลิตแบบ 6 โมดูล 72

7.13 แสดงเครือข่ายการสื่อสารและการรับ-ส่งข้อมูลระหว่างสถานีของสายการผลิตแบบ 6 โมดูล 74

บทที่ 1
บทนำ

1.1 ความสำคัญและที่มาของวิทยานิพนธ์

เป็นที่ทราบกันดีอยู่แล้วว่า ระบบการผลิตแบบอัตโนมัติสามารถเพิ่มประสิทธิภาพในการผลิต ผลผลิตที่ได้มีคุณภาพและมาตรฐานเท่ากัน ลดความเสียหายของผลิตภัณฑ์ รวมทั้งเพิ่มความปลอดภัยในการทำงาน ซึ่งในอุตสาหกรรมการผลิตปัจจุบันที่มีการแข่งขันสูงขึ้นในยุคโลกาภิวัฒน์ ตลาดต้องการผลิตภัณฑ์ที่หลากหลายและต้องมีการปรับปรุงเปลี่ยนแปลงผลิตภัณฑ์อยู่ตลอดเวลาทำให้หลาย ๆ บริษัทต้องปรับระบบการผลิตให้สามารถผลิต ผลิตภัณฑ์ที่หลากหลายได้ในเวลาที่รวดเร็วขึ้น เพื่อเพิ่มความสามารถในการแข่งขัน ซึ่งสิ่งที่สำคัญที่ต้องคำนึงถึงในการเพิ่มความสามารถในการผลิต ก็คือ การใช้เทคโนโลยีของระบบควบคุมแบบอัตโนมัติ ในสายการผลิตแบบอัตโนมัติ และ โปรแกรมเมเบิลลอจิกคอนโทรลเลอร์ (PLC)

การนำเทคโนโลยีของระบบควบคุมแบบอัตโนมัติมาใช้ เช่น ระบบการผลิตแบบโมดูลาร์ (Modular) เทคโนโลยีเซนเซอร์ (Sensor) ระบบตรวจสอบโดยใช้กล้อง (Vision system) การใช้หุ่นยนต์ (Robot) การใช้เครือข่ายการควบคุม (Network) รวมไปถึงซอฟต์แวร์ที่ใช้ตรวจดูแบบ Human Machine Interface (HMI) ซึ่งทั้งหมดนี้สามารถนำมาช่วยเพิ่มประสิทธิภาพในการผลิตเปลี่ยนรูปแบบของโรงงาน ให้เป็นโรงงานอัตโนมัติ (Factory Automation)

วิทยานิพนธ์ฉบับนี้ต้องการแสดงให้เห็นถึงการใช้เครือข่ายการควบคุมในระดับต่าง ๆ ในโรงงานอัตโนมัติ ความยืดหยุ่นของการใช้เครือข่ายการควบคุม โดยการเปลี่ยนรูปแบบของสายการผลิตในแบบต่าง ๆ

1.2 วัตถุประสงค์ของวิทยานิพนธ์

1. จัดระบบเครือข่ายการควบคุม ระดับต่าง ๆ ในสายการผลิตอัตโนมัติแบบโมดูล
2. เพื่อศึกษาเทคโนโลยีการสื่อสารฟิลด์บัส (Fieldbus)

1.3 ขอบเขตของวิทยานิพนธ์

1. จัดระบบเครือข่ายการควบคุมในสายการผลิตอัตโนมัติแบบโมดูลโดยใช้โปรโตคอลฟิลด์บัส GENIUS BUS และ SNP/SNP-X และใช้ซอฟต์แวร์ CIMPLICITY HMI ในการมอนิเตอร์ระบบการผลิต
2. ทดลองปรับเปลี่ยนสายการผลิตอัตโนมัติในรูปแบบต่าง ๆ ทั้งหมด 7 รูปแบบ

1.4 ขั้นตอนในการทำวิทยานิพนธ์

1. ศึกษาการทำงานและโปรแกรมใน PLC ของโมดูลต่าง ๆ ทั้ง 7 โมดูลในชุดการผลิตอัตโนมัติ

2. ศึกษาการเขียนโปรแกรมแลดเดอร์ใน PLC ของ GE FANUC โปรโตคอลฟิลด์บัส GENIUS BUS และ SNP/SNP-X และซอฟต์แวร์ CIMPLICITY HMI

3. ติดตั้ง PLC ของ GE FANUC ลงในโมดูลต่าง ๆ ตามความเหมาะสม พร้อมทั้งเขียนโปรแกรมและทดสอบการทำงานในโมดูลต่าง ๆ

4. จัดเครือข่ายการสื่อสารของ GENIUS BUS และ SNP/SNP-X เดินสายสื่อสารสำหรับโมดูลต่าง ๆ และต่อเชื่อม PLC เข้ากับเครือข่ายอีเทอร์เน็ต

5. จัดสายการผลิตโดยใช้ทั้ง 7 โมดูล เขียนโปรแกรมรับ-ส่งข้อมูลระหว่างโมดูล จากนั้นทดสอบการทำงานของสายการผลิตอัตโนมัติ

6. เขียนโปรแกรมเพื่อใช้ตรวจดูระบบ โดยแสดงการทำงานในลักษณะรูปภาพบนจอมอนิเตอร์โดยใช้โปรแกรม CIMPLICITY HMI

7. ทดลองปรับเปลี่ยนรูปแบบสายการผลิต แบบอื่น ๆ อีก 6 รูปแบบ ให้ทำงานได้ตามลักษณะสายการผลิตแบบนั้น ๆ

8. รวบรวมข้อมูลและสรุปผลการทำวิทยานิพนธ์

1.5 ประโยชน์ที่คาดว่าจะได้รับจากการทำวิทยานิพนธ์

1. ได้โปรแกรมแลดเดอร์ เพื่อควบคุมการทำงานในโมดูลต่าง ๆ เครือข่ายการสื่อสารและโปรแกรมตรวจดูระบบ สำหรับระบบการผลิตอัตโนมัติแบบโมดูล

2. ได้เข้าใจถึงประโยชน์ของการนำเอาเครือข่ายการควบคุมมาใช้ในระบบการผลิตอัตโนมัติแบบโมดูล

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

บทที่ 2
ระบบการผลิตอัตโนมัติ

2.1 ระบบการผลิตแบบโมดูลาร์ (Modular production system)

ระบบการผลิตแบบโมดูลาร์ คือระบบการผลิตที่ประกอบไปด้วยโมดูลการผลิตย่อย ๆ ตามลักษณะหน้าที่ในการทำงาน เช่น โมดูลทดสอบชิ้นงาน โมดูลแยกชิ้นงาน เป็นต้น ระบบการผลิตแบบนี้สามารถเพิ่มความยืดหยุ่น (Flexibility) ให้กับการผลิต การปรับเปลี่ยนสายการผลิตเมื่อต้องการเปลี่ยนผลิตภัณฑ์ ก็เพียงแต่เปลี่ยนโมดูลเครื่องจักร ซึ่งทำได้ง่ายและรวดเร็ว

ระบบการผลิตที่ใช้ในการทำวิทยานิพนธ์ ประกอบไปด้วยโมดูลทั้งหมด 7 โมดูล หรือ 7 สถานีดังนี้

1. สถานีจ่ายชิ้นงาน (Distribution station) หรือ D+ ทำหน้าที่จ่ายชิ้นงานจากกระบอกบรรจุชิ้นงานเข้าสู่สายการผลิต

2. สถานีทดสอบชิ้นงาน (Testing station) หรือ T+ ทำหน้าที่ตรวจสอบชิ้นงานโดยสามารถตรวจสอบความเป็นโลหะ, ตรวจสอบการสะท้อนแสง และตรวจสอบความสูงของชิ้นงานได้ ซึ่งสามารถใช้แยกชิ้นงานโลหะ, ชิ้นงานสีแดง และชิ้นงานสีดำ ที่มีความสูงใช้ได้ และใช้ไม่ได้

3. สถานีควบคุมคุณภาพ (Quality station) หรือ Q+ ทำหน้าที่ตรวจสอบคุณภาพชิ้นงานโดยใช้กล้อง ซึ่งสามารถใช้วัดขนาดเส้นผ่าศูนย์กลางวงนอก และความหนาของร่องวงแหวนบนชิ้นงาน

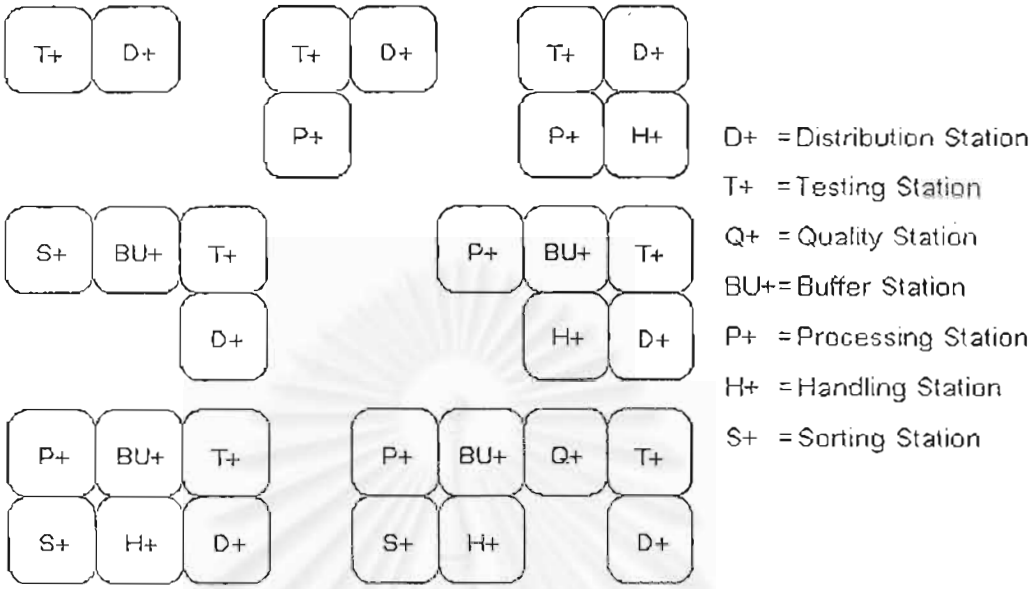
4. สถานีพักชิ้นงาน (Buffer station) หรือ BU+ ทำหน้าที่พักชิ้นงานชั่วคราว ซึ่งสามารถพักชิ้นงานไว้ได้สูงสุด 3 ชิ้น

5. สถานีผลิตชิ้นงาน (Processing station) หรือ P+ ทำหน้าที่จำลองการผลิตชิ้นงาน โดยจะมีหัวสว่าน ทำหน้าที่เจาะรูชิ้นงาน และหัวทดสอบรูเจาะ ทำหน้าที่ตรวจสอบรูเจาะบนชิ้นงาน

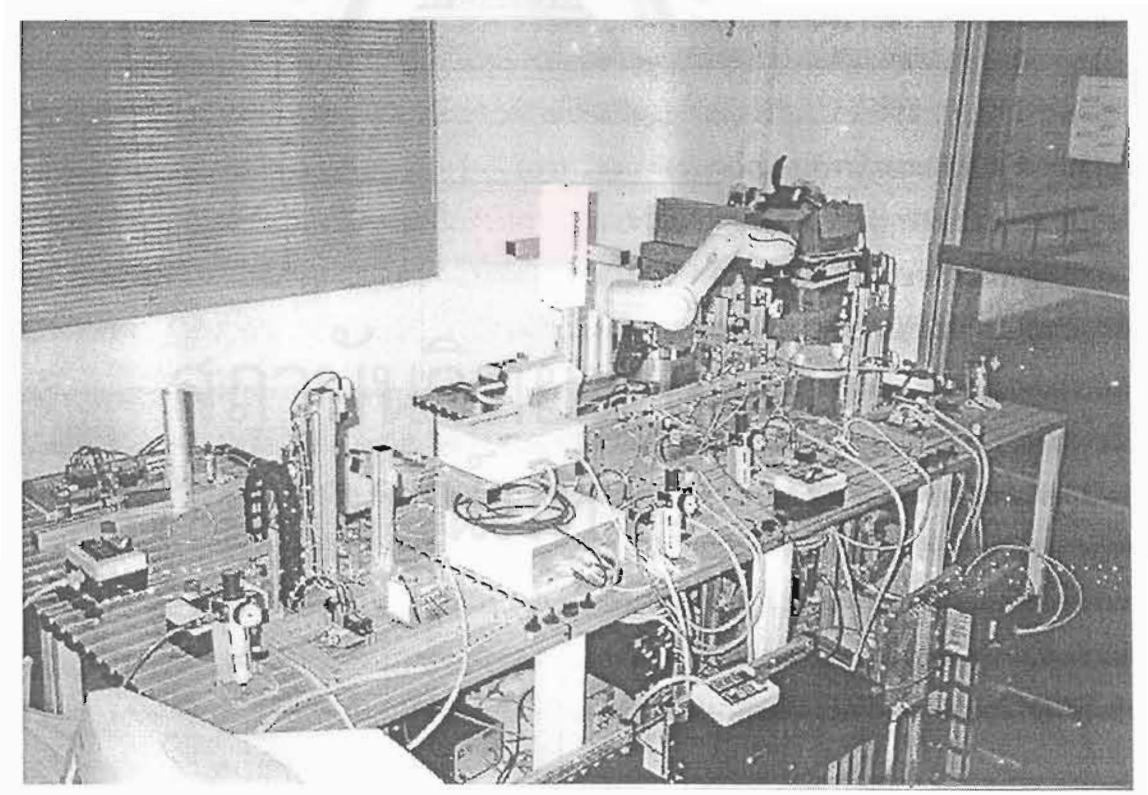
6. สถานีแขนกลเพื่อเคลื่อนย้ายหรือจัดเก็บชิ้นงาน (Handling station) หรือ H+ ทำหน้าที่เคลื่อนย้ายชิ้นงาน จากสถานีหนึ่งไปอีกจากสถานีหนึ่ง และทำหน้าที่จัดเก็บชิ้นงานที่ผลิตเสร็จแล้ว โดยแยกใส่กระบอกใส่ชิ้นงาน (Magazine) ตามลักษณะชิ้นงานนั้น

7. สถานีแยกชิ้นงาน (Sorting station) หรือ S+ ทำหน้าที่แยกชิ้นงาน ให้ไหลลงในรางรับชิ้นงาน ตามลักษณะชิ้นงานนั้น

โมดูล หรือสถานีทั้งเจ็ดสถานีนี้ สามารถนำมาประกอบเป็นสายการผลิต ได้หลากหลายรูปแบบ ตามลักษณะของชิ้นงานที่จะผลิต ซึ่งการปรับเปลี่ยนรูปแบบของสายการผลิตสามารถทำได้ง่าย และรวดเร็ว ดังรูปที่ 2.1 และรูปที่ 2.2 แสดงแบบหนึ่งของสายการผลิตที่ใช้สถานีเจ็ดสถานีประกอบกัน



รูปที่ 2.1 แสดงสายการผลิตในรูปแบบต่าง ๆ



รูปที่ 2.2 แสดงสายการผลิตที่ประกอบด้วย 7 สถานี

หอสมุดกลาง สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

2.2 อุปกรณ์และเซนเซอร์ที่ใช้ในโมดูลต่าง ๆ

เครื่องอัดลม (Compressor) มีหน้าที่ดูดอากาศเข้า แล้วอัดอากาศให้มีความดันสูงขึ้นกว่าเดิมจากนั้นจึงส่งอากาศที่ถูกอัดตัวแล้วไปยังถังพักลม ก่อนถูกส่งไปใช้งานในการควบคุมระบบนิวแมติก ดังรูปที่ 2.3

ชุดปรับปรุงคุณภาพลมอัด (Service unit) ทำหน้าที่กรองฝุ่นและน้ำ ออกจากลมอัดก่อนที่จะส่งเข้าเครื่องจักร ประกอบด้วย ฟิลเตอร์ (Filter) ใช้กรองเศษฝุ่นผงและน้ำ เร็กกูเรเตอร์ (Regulator) ใช้ปรับหรือควบคุมความดันลมในระบบดังรูปที่ 2.4

โซลินอยด์วาล์ว (Solenoid valve) ทำหน้าที่ควบคุมการไหลของลมอัดให้ไปตามทิศทางที่ต้องการ โดยใช้ไฟฟ้าเข้ามาควบคุมการทำงานของวาล์วให้เปิด-ปิดการจ่ายลมสู่อุปกรณ์ เช่น กระบอกสูบ ดังรูปที่ 2.5

กระบอกสูบลม (Cylinder) ทำหน้าที่เปลี่ยนพลังงานลมอัดให้เป็นการเคลื่อนที่ ประกอบด้วยลูกสูบ ซึ่งเคลื่อนที่ไป-กลับในกระบอกสูบตามทิศทางของลมที่เข้ามาดังรูปที่ 2.6

Semi-rotary actuator ลักษณะการทำงานคล้ายกับกระบอกสูบแต่การเคลื่อนที่จะเป็นการหมุนไปและหมุนกลับจาก 0 องศา ถึง 180 องศา ดังรูปที่ 2.7

Pneumatic / Electronic switch ทำหน้าที่เป็นสวิตช์ความดันสุญญากาศ เมื่อเกิดแรงดันสุญญากาศ เช่น ในกรณีที่หัวดูด ดูดชิ้นงานแล้ว สวิตช์ความดันจะเปิด (ON) และเมื่อไม่มีแรงดันสุญญากาศ สวิตช์ความดันจะปิด (OFF) ดังรูปที่ 2.8

Suction cup ทำหน้าที่เป็นหัวดูด ดูดจับชิ้นงาน ดังรูปที่ 2.9

Position sensor ใช้ทำหน้าที่ตรวจสอบความสูงของชิ้นงานโดยค่าความสูงที่วัดได้ถูกแปลงเป็นสัญญาณแอนะลอก 0-10 โวลต์ ดังรูปที่ 2.10

Proximity Sensors เป็นอุปกรณ์ใช้สำหรับตรวจจับวัตถุทำหน้าที่เหมือนกับลิมิตสวิตช์ แต่สามารถใช้งานได้กว้างกว่าและเป็นอุปกรณ์ตรวจจับอิเล็กทรอนิกส์ ไม่มีส่วนเคลื่อนที่ทำให้มีอายุการใช้งานนานกว่าลิมิตสวิตช์ การใช้งานสามารถเลือกใช้พล็อกซิมิตีเซนเซอร์ (Proximity Sensor) ชนิดต่าง ๆ ตามลักษณะการใช้งาน หรือลักษณะของวัตถุที่ตรวจจับ ดังนี้

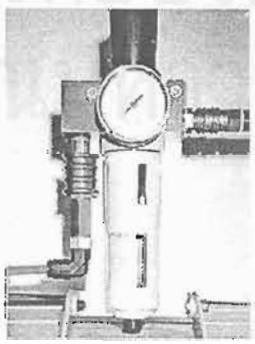
- *Inductive proximity sensor* ใช้หลักการของการเปลี่ยนแปลงของค่าความเหนี่ยวนำ (Inductive) โดยสามารถใช้ตรวจจับวัตถุที่เป็นตัวนำชนิดต่าง ๆ ตัวตรวจจับประกอบด้วยขดลวดพันเป็นขด อยู่บริเวณผิวด้านหน้า ขดลวดจะสร้างสนามแม่เหล็กออกมา เมื่อมีวัตถุที่เป็นตัวนำผ่านเข้ามาในบริเวณสนามแม่เหล็กจะเหนี่ยวนำให้เกิดกระแสขึ้นที่ผิวของวัตถุ ซึ่งถือเป็นโหลดของขดลวด ทำให้การสั่นของคลื่นแม่เหล็กเปลี่ยนแปลงไป และค่าของการสั่นจะถูกเปรียบเทียบกับระดับอ้างอิง เพื่อกำหนดสถานะของตัวตรวจจับให้เปิด (ON) หรือปิด (OFF)
- *Capacitive proximity sensor* ใช้หลักการของการเปลี่ยนแปลงของค่าความเก็บประจุ (Capacitance) โดยสามารถใช้ตรวจจับวัตถุที่เป็นตัวนำและ

ไม่เป็นตัวนำได้ เมื่อมีวัตถุเคลื่อนที่เข้ามาในสนามไฟฟ้าด้านหน้าของอุปกรณ์ตรวจจับ จะทำให้ค่าความเก็บประจุเปลี่ยนไป เนื่องจากวัตถุมีค่า Permittivity มากกว่าอากาศที่มีค่าเท่ากับ 1 และค่าความเก็บประจุจะถูกเปรียบเทียบกับค่าอ้างอิง เพื่อกำหนดสถานะของตัวตรวจจับให้เปิด (ON) หรือปิด (OFF)

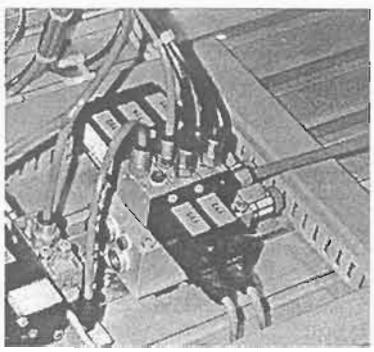
- *Optical proximity sensor* ใช้หลักการของการส่ง-รับแสง ประกอบด้วยตัวส่งลำแสง ปกติแล้วจะใช้แสงอินฟาเรดซึ่งคนมองไม่เห็น และตัวรับลำแสง การใช้งานจะติดตั้งให้ตัวรับและส่งอยู่ตรงกัน ในกรณีที่ไม่มีวัตถุแสงจะสามารถส่งจากตัวส่งไปสู่ตัวรับได้ แต่ในกรณีที่มีวัตถุเคลื่อนที่เข้ามาขวางลำแสงจะทำให้ลำแสงจากตัวส่งไปไม่ถึงตัวรับ ซึ่งจะเป็นการเปลี่ยนสถานะของตัวตรวจจับให้เปิด (ON) หรือปิด (OFF)

ตัวอย่างการใช้งานพล็อกซิมิตีเซนเซอร์ เช่น ในสถานีทดสอบชิ้นงาน มีการใช้ Inductive, Capacitive และ Optical proximity sensor ในการตรวจสอบวัสดุ และสีของชิ้นงาน ดังรูปที่ 2.11

รูปที่ 2.3 แสดงเครื่องอัดลม

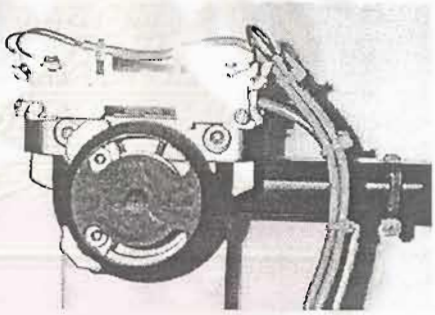


รูปที่ 2.4 แสดงชุดปรับปรุงคุณภาพลมอัด

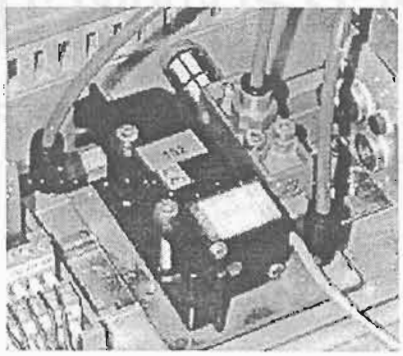


รูปที่ 2.5 แสดงโซลินอยด์วาล์ว

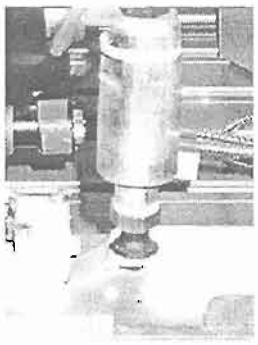
รูปที่ 2.6 แสดงกระบอกสูบลม



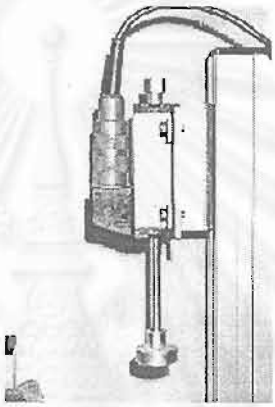
รูปที่ 2.7 แสดง Semi-rotary actuator



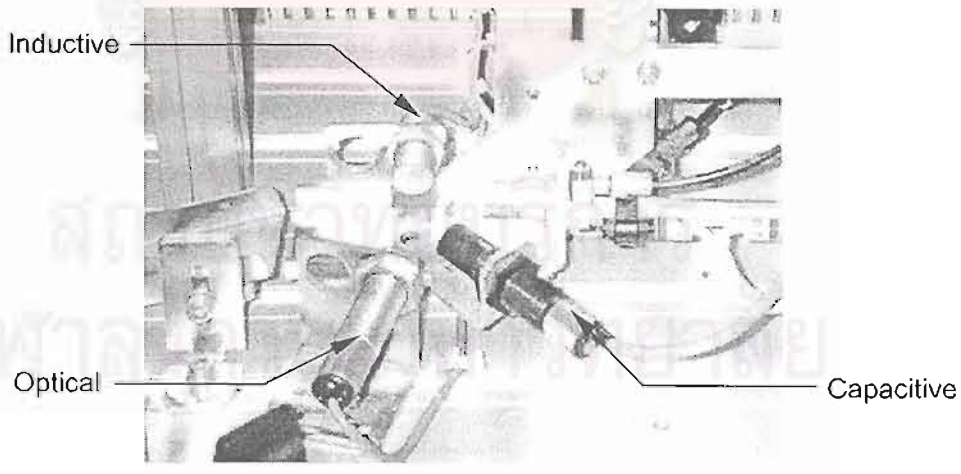
รูปที่ 2.8 แสดง Pneumatic / Electronic Switch



รูปที่ 2.9 แสดง Suction cup



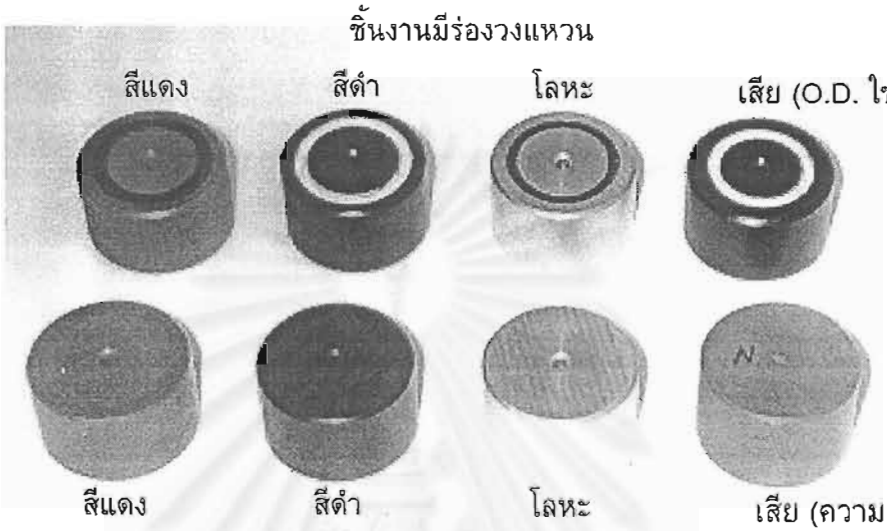
รูปที่ 2.10 แสดง Position sensor



รูปที่ 2.11 แสดง Inductive, Capacitive และ Optical proximity sensor ในสถานีทดสอบชิ้นงาน

2.3 ชิ้นงานที่ใช้ผลิตในระบบการผลิตแบบโมดูลาร์

ลักษณะชิ้นงานที่ใช้ผลิตเป็นชิ้นงานทรงกระบอกในแบบต่าง ๆ คือ ชิ้นงานโลหะ, ชิ้นงานพลาสติกสีแดง และชิ้นงานพลาสติกสีดำ ซึ่งบางชิ้นงานจะมีร่องวงแหวนด้านบน ชิ้นงานในแบบต่าง ๆ สามารถผลิตในสายการผลิตเดียวกันดังรูปที่ 2.12



รูปที่ 2.12 แสดงชิ้นงานในแบบต่าง ๆ

เมื่อชิ้นงานถูกส่งต่อ จากสถานีหนึ่งไปยังสถานีถัดไปจะมีการรับ-ส่งข้อมูลกันระหว่างสถานี ซึ่งข้อมูลที่รับ-ส่งสามารถแบ่งออกได้เป็น 2 ส่วน คือ

1. <u>ข้อมูลของสถานี</u> เป็นข้อมูลที่แสดงถึงสถานะของสถานี ซึ่งจะถูกส่งไปที่สถานีก่อนหน้า ประกอบด้วยบิตข้อมูลดังนี้

- *บิตข้อมูล S_RDY* ใช้แสดงว่าสถานีเปิดทำงานและพร้อมทำงานแล้ว
- *บิตข้อมูล P_RDY* ใช้แสดงว่าสถานีพร้อมรับชิ้นงาน
- *บิตข้อมูล D_REQ* ใช้แสดงว่าสถานีร้องขอข้อมูล

2. <u>ข้อมูลของชิ้นงาน</u> เป็นข้อมูลที่แสดงถึงลักษณะของชิ้นงาน ซึ่งข้อมูลของชิ้นงานนี้จะถูกใส่ข้อมูลเพิ่มมากขึ้นที่สถานีต่าง ๆ ระหว่างที่ชิ้นงานไหลผ่านไป สถานีที่จะใส่ข้อมูลชิ้นงานเพิ่มขึ้น มีดังนี้

- *Testing station* จะทำการใส่ข้อมูลชิ้นงานใน 3 บิต คือ บิต D0, D1 และ D2 ซึ่งเป็นข้อมูลของวัสดุและสี รวมทั้งความสูงของชิ้นงาน ดังตารางที่ 2.1 และตารางที่ 2.2
- *Quality station* จะทำการใส่ข้อมูลชิ้นงานใน 2 บิต คือ บิต D3 และ D4 ซึ่งเป็นข้อมูลของร่องวงแหวนบนชิ้นงาน ดังตารางที่ 2.3

- *Processing station* จะทำการใส่ข้อมูลชิ้นงานใน 1 บิต คือบิต D5 ซึ่งเป็นข้อมูลการทดสอบรูเจาะของชิ้นงาน ดังตารางที่ 2.4

และยังมีข้อมูลเพิ่มเติมที่ถูกส่งไปพร้อมกับข้อมูลของชิ้นงานอีก 2 บิดดังนี้

- *บิตข้อมูล EN* ใช้แสดงว่ามีข้อมูลพร้อมส่งแล้ว
- *บิตข้อมูล HS* ใช้แสดงการร้องขอให้แขนกลหยิบชิ้นงานมาให้

ตารางที่ 2.1 แสดงข้อมูลบิต D0

D0	ลักษณะชิ้นงาน
0	ชิ้นงานสูงไปหรือต่ำไป
1	ชิ้นงานสูงใช้ได้

ตารางที่ 2.2 แสดงข้อมูลบิต D1 และ D2

D1	D2	ลักษณะชิ้นงาน
0	0	ชิ้นงานเสีย
1	0	ชิ้นงานสีแดง
0	1	ชิ้นงานสีดำ
1	1	ชิ้นงานโลหะ

ตารางที่ 2.3 แสดงข้อมูลบิต D3 และ D4

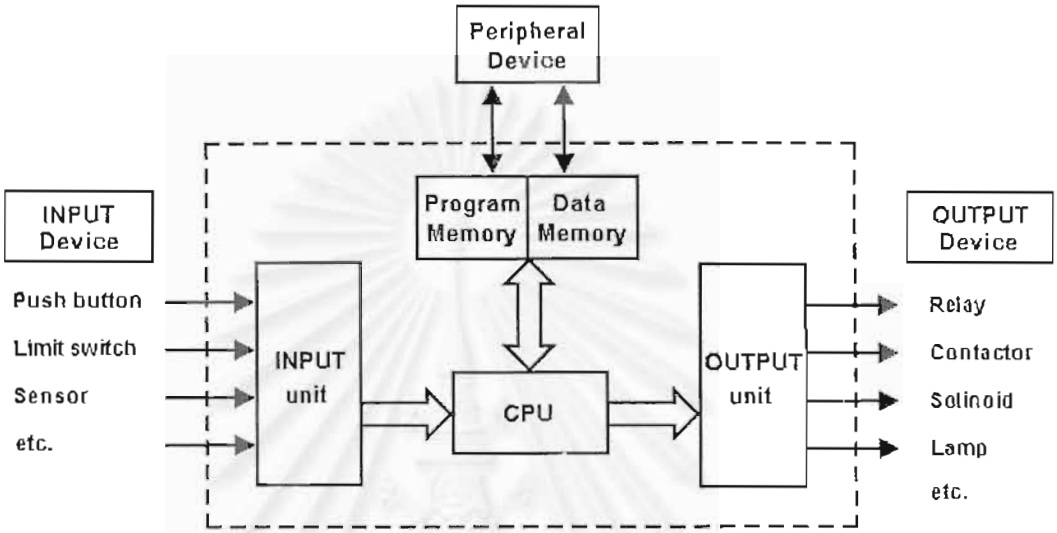
D3	D4	ลักษณะชิ้นงาน
0	0	O.D. และความหนาร่องวงแหวนใช้ไม่ได้
1	0	O.D. ใช้ได้
0	1	ความหนาร่องวงแหวนใช้ได้
1	1	O.D. และความหนาร่องวงแหวนใช้ได้

ตารางที่ 2.4 แสดงข้อมูลบิต D5

D5	ลักษณะชิ้นงาน
0	ไม่มีรูเจาะหรือรูเจาะใช้ไม่ได้
1	รูเจาะบนชิ้นงานใช้ได้

2.4 โปรแกรมเมเบิลลอจิกคอนโทรลเลอร์ (PLC)

PLC เป็นอุปกรณ์ควบคุมการทำงานของเครื่องจักรหรือกระบวนการต่าง ๆ ที่มีลักษณะการทำงานเป็นซีเควันซ์ตามโปรแกรมคำสั่งของผู้ใช้ โครงสร้างของ PLC โดยทั่ว ๆ ไป แบ่งเป็น 3 ส่วน คือ หน่วยประมวลผลกลาง (Central Processing Unit) หน่วยความจำ (Memory) และหน่วยอินพุต/เอาต์พุต (Input/Output) ดังรูปที่ 2.13



รูปที่ 2.13 แสดงโครงสร้างของ PLC

1. หน่วยประมวลผลกลาง หรือ CPU เป็นตัวประมวลผลและควบคุมการทำงานของ PLC โดยทั่วไปจะใช้ ไมโครโปรเซสเซอร์ เป็นตัวประมวลผล หน้าที่ของ CPU ก็คือ เมื่อรับอินพุตเข้ามาจะทำการประมวลผลตามโปรแกรมการทำงานที่บันทึกไว้ จากนั้นจะส่งผลที่ได้ออกไปยังเอาต์พุต และจะวนกลับไปรับข้อมูลอินพุตเข้ามาอีก แล้วทำซ้ำเดิมในลักษณะนี้เรื่อย ๆ ซึ่งเรียกว่า การสแกน (Scan)

2. หน่วยความจำ (Memory) เป็นหน่วยความจำที่ใช้เก็บโปรแกรมและข้อมูลแบ่งออกเป็น 2 ส่วน คือ

1. *หน่วยความจำระบบ* ทำหน้าที่เก็บโปรแกรมระบบและข้อมูลระบบของ PLC
2. *หน่วยความจำผู้ใช้* ทำหน้าที่เก็บโปรแกรมการทำงานที่ผู้ใช้เขียนขึ้น ข้อมูลของหน่วยอินพุต/เอาต์พุต และข้อมูลของอุปกรณ์ภายใน เช่น รีจิสเตอร์ รีเลย์ภายใน

หน่วยความจำมีหลายชนิดแล้วแต่การใช้งาน และชนิดของข้อมูลที่จัดเก็บ ดังนี้

- *ROM (Read Only Memory)* เป็นหน่วยความจำที่ไม่อนุญาตให้ผู้ใช้เปลี่ยนแปลงหรือแก้ไขข้อมูลภายใน สามารถเก็บรักษาข้อมูลได้โดยไม่มีกระแสไฟ ส่วนใหญ่ใช้เก็บโปรแกรมและข้อมูลระบบของตัว PLC

- *RAM (Random Access Memory)* เป็นหน่วยความจำที่ผู้ใช้สามารถเปลี่ยนแปลงแก้ไขข้อมูลได้ตลอดเวลา สามารถเก็บข้อมูลได้โดยต้องมีกระแสไฟ ในการนำไปใช้งานจะต้องมีการต่อแหล่งจ่ายไฟสำรองไว้เพื่อป้องกัน ไม่ให้ข้อมูลสูญหาย ส่วนใหญ่ใช้เก็บข้อมูลอุปกรณ์ภายใน
- *EPROM (Erasable Programmable Read Only Memory)* เป็นหน่วยความจำคล้ายกับ ROM แต่สามารถล้างโปรแกรม และเขียนโปรแกรมใหม่ได้โดยใช้เครื่องเขียนชนิดพิเศษ ส่วนใหญ่ใช้สำหรับเก็บโปรแกรมการทำงานที่ได้พัฒนาจนใช้งานได้ดีแล้ว
- *EEPROM (Electrically Erasable Programmable Read Only Memory)* เป็นหน่วยความจำคล้ายกับ EPROM แต่สามารถลบและเขียนได้ด้วยไฟฟ้าและข้อมูลจะคงอยู่แม้ไม่มีกระแสไฟ ส่วนใหญ่ใช้สำหรับเก็บโปรแกรมการทำงานที่เสร็จสมบูรณ์แล้ว

3. หน่วยอินพุต/เอาต์พุต (Input/Output) ทำหน้าที่เชื่อมต่ออุปกรณ์ภายนอก เข้ากับ CPU ซึ่งจะทำการแปลงระดับสัญญาณให้เป็นระดับที่เหมาะสมกับตัว CPU และแยกสัญญาณภายนอกและสัญญาณภายในให้ออกจากกันเพื่อป้องกันไม่ให้ CPU ได้รับความเสียหายในกรณีเกิดการลัดวงจร การใช้งานสามารถแบ่งได้เป็น 2 ส่วน คือ

- *อินพุต (Input)* ทำหน้าที่รับสภาวะหรือปริมาณทางกายภาพ จากอุปกรณ์ตรวจเช็ค เช่น พร็อกซิมิตีเซนเซอร์ ลิมิตสวิตช์ ปุ่มกด อุณหภูมิ และแรงดันกระแสอื่น ๆ แล้วส่งไปยัง CPU เพื่อประมวลผลตามโปรแกรมคำสั่งของผู้ใช้
- *เอาต์พุต (Output)* ทำหน้าที่รับค่าสัญญาณที่ได้จากการประมวลผลของ CPU ไปขยายให้สัญญาณออกมีขนาดใหญ่พอที่จะขับอุปกรณ์ภายนอกเช่น มอเตอร์ วาล์ว ปั๊ม โซลินอยด์ หลอดไฟ และอื่น ๆ

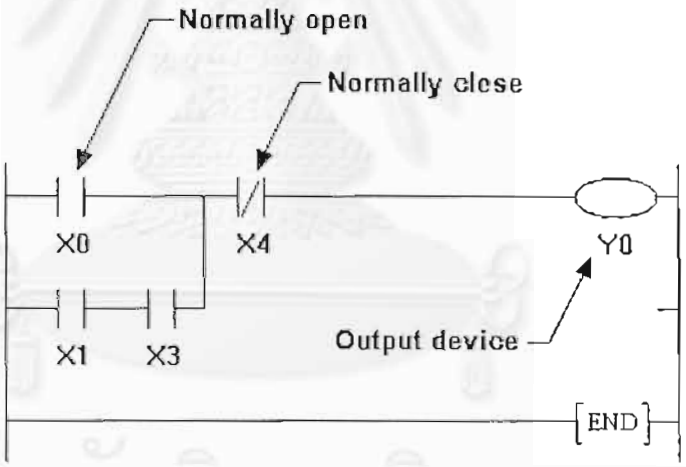
หน่วยอินพุต/เอาต์พุตที่ใช้งานทั่วไป สามารถแบ่งออกได้เป็น 3 ประเภท คือ

1. *ดิจิตอล* หรือแบบลอจิก สัญญาณที่เข้าออกจะอยู่ในรูป 0 หรือ 1 เท่านั้น
2. *แอนะลอก* สัญญาณที่เข้า-ออกจะเปลี่ยนค่าได้ในช่วงที่กำหนด มีทั้งแบบปรับเปลี่ยนค่ากระแส และแรงดัน ซึ่งโดยปกติจะอยู่ในช่วงกระแส 4-20 mA และแรงดัน 0-10 V
3. *แบบพิเศษ* สัญญาณเข้า-ออกจะอยู่ในรูปแบบอื่น ๆ ตามลักษณะของอินพุต/เอาต์พุต นั้น เช่นอยู่ในรูปของพัลส์ (Pulse)

การเขียนโปรแกรมการทำงานใน PLC

ภาษาหรือคำสั่งที่ใช้เขียนโปรแกรม PLC มีอยู่หลายภาษา หรือหลายแบบด้วยกัน แต่ที่นิยมใช้กัน มีดังนี้

- *ภาษาแลดเดอร์* ประกอบไปด้วยสัญลักษณ์หน้าสัมผัส มีลักษณะคล้ายวงจรรีเลย์ การเขียนโปรแกรมภาษาแลดเดอร์ทำได้ง่าย และเป็นภาษาพื้นฐานสำหรับ PLC คำสั่งภาษาแลดเดอร์ประกอบด้วยสัญลักษณ์ของหน้าสัมผัส และขดลวด เพื่อแสดงเงื่อนไขในการควบคุมระหว่างอุปกรณ์ หน่วยอินพุต/เอาต์พุต และอุปกรณ์ภายใน ดังรูปที่ 2.14
- *ภาษาบูลลีน* เป็นภาษาพื้นฐานของ PLC เช่นเดียวกับภาษาแลดเดอร์ คำสั่งในภาษาบูลลีนมีลักษณะคล้ายกับพีชคณิตบูลลีน ประกอบไปด้วยกลุ่มคำสั่งเช่นเดียวกับ ภาษาแลดเดอร์ ดังตารางที่ 2.5
- *ฟังก์ชันชาร์ต* เรียกย่อ ๆ ว่า SFC เป็นภาษาที่ใช้บรรยายการควบคุม ซีเควันซ์ โดยใช้แผนภาพสถานะ ซึ่งเข้าใจได้ง่าย และสามารถเขียนได้ทันทีจากการทำงานของเครื่องจักร ดังรูปที่ 2.15

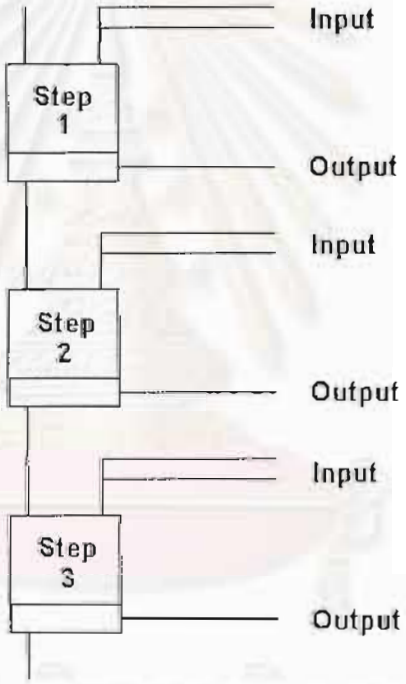


รูปที่ 2.14 แสดงตัวอย่างการเขียนโปรแกรมภาษาแลดเดอร์

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

ตารางที่ 2.5 แสดงตัวอย่างการเขียนโปรแกรมคำสั่งบูลลีน

ADDRESS	INSTRUCTION	DATA
0	LD	X1
1	AND	X3
2	OR	X0
3	AND NOT	X4
4	OUT	Y0
5	END	



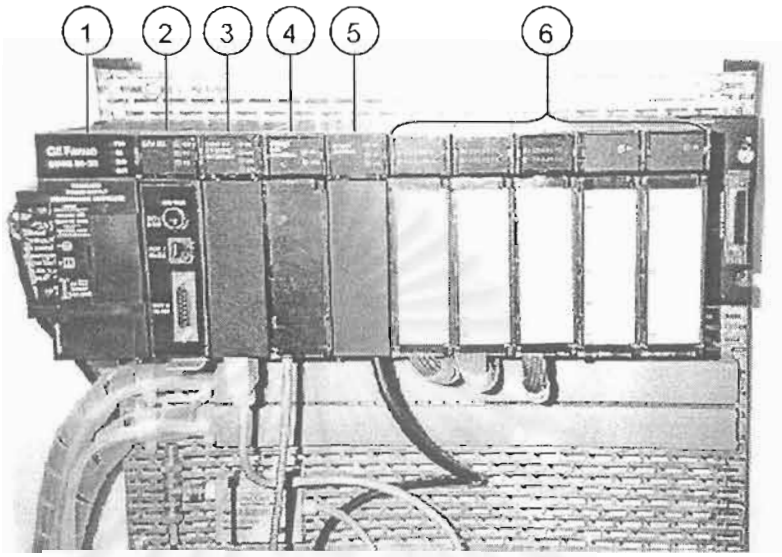
รูปที่ 2.15 แสดงตัวอย่างภาษา SFC

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

2.5 PLC ของ GE FANUC ที่นำมาใช้กับโมดูลการผลิตต่าง ๆ

PLC ของ GE FANUC ที่นำมาใช้ประกอบไปด้วยรุ่นต่าง ๆ ดังนี้

PLC รุ่น 90-30 นำมาใช้กับสถานีจ่ายชิ้นงานและสถานีทดสอบชิ้นงาน ทำหน้าที่เป็นตัวมาสเตอร์ (Master) รับ-ส่งข้อมูลกับตัว PLC อื่นประกอบด้วยโมดูลใช้งาน ดังรูปที่ 2.16

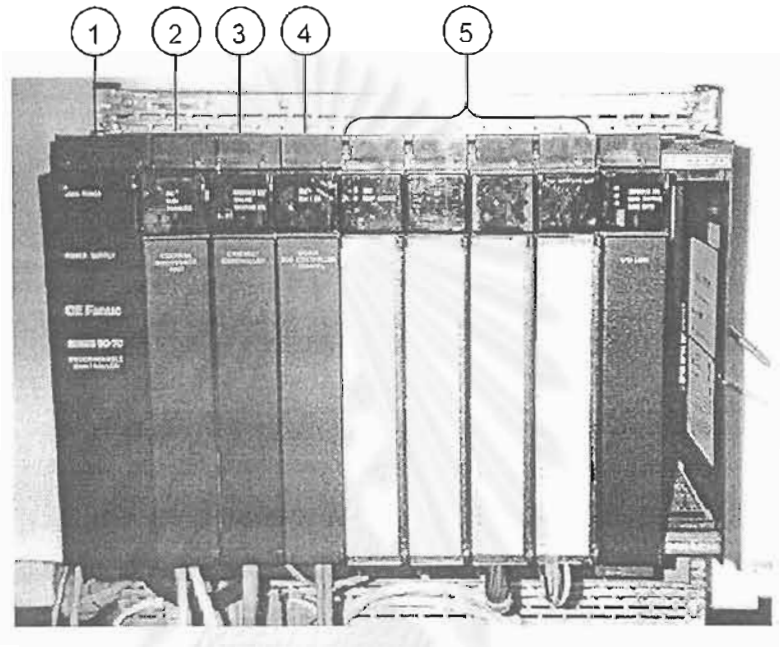


รูปที่ 2.16 แสดง PLC รุ่น 90-30

1. *POWER SUPPLY* ใช้จ่ายไฟให้กับตัว PLC สามารถจ่ายไฟได้สูงสุด 30 watts
2. *CPU 351* ใช้ไมโครโปรเซสเซอร์ รุ่น 80386 EX ทำงานที่ความเร็ว 25 MHZ สามารถกำหนดค่าอินพุตและเอาต์พุตได้สูงสุดอย่างละ 2048 บิต และสามารถเขียนโปรแกรมได้สูงสุด 80 Kbytes
3. *CMM 321 ETHERNET INTERFACE* ใช้เชื่อมต่อ PLC เข้ากับ Ethernet LAN โดยต่อโมดูลเข้ากับตัว Transceiver ผ่านทางสาย AUI เข้าสู่เครือข่ายอีเทอร์เน็ต (Ethernet) เพื่อใช้สื่อสารกับ Host หรือ อุปกรณ์ควบคุมอื่น ๆ บนเครือข่าย
4. *BEM 331 GENIUS BUS CONTROLLER* ใช้เชื่อมต่อ PLC เข้ากับ Genius I/O serial bus โดยสามารถใช้รับ-ส่งข้อมูลได้สูงสุดครั้งละ 128 bytes และต่อกับอุปกรณ์ได้สูงสุด 31 ตัว
5. *CMM 311 COMM COPROC* เป็นโมดูลเพื่อใช้สื่อสารแบบอนุกรมผ่านทางพอร์ตอนุกรม (Serial port) ซึ่งจะมีให้ 2 พอร์ต คือ พอร์ต 1 สำหรับสื่อสารโดยใช้มาตรฐาน RS-232 และ พอร์ต 2 สำหรับสื่อสารได้ทั้งมาตรฐาน RS-232 และ RS-485 ในส่วนของโปรโตคอลสื่อสารสามารถเลือกใช้ได้ 3 แบบ คือ CCM Communication Protocol, The RTU (Modbus) Slave Communications Protocol และ SNP Protocol

6. *I/O Modules* เป็นโมดูลสำหรับอินพุตและเอาต์พุต มีทั้งหมด 5 โมดูล ดังนี้ 24V 16Pt Input 2 โมดูล, 24V 16 Pt Output, 4 Pt Analog input (V&I) และ 2Pt Analog Output (V&I)

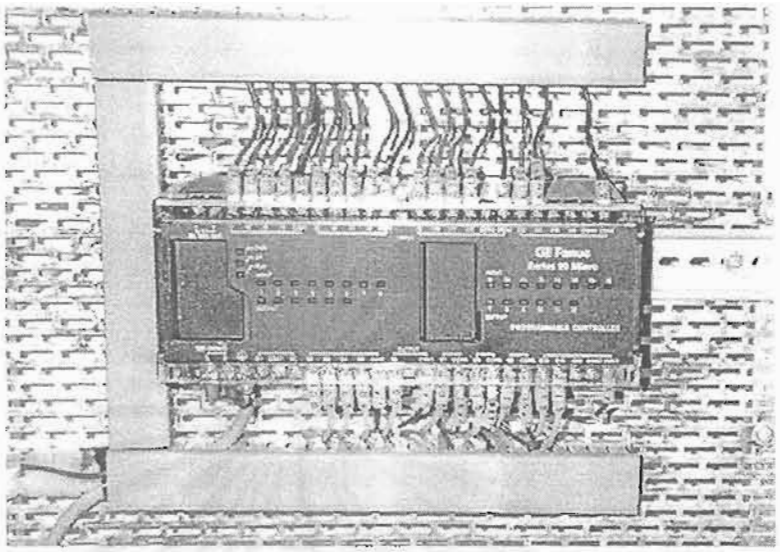
PLC รุ่น 90-70 นำมาใช้กับสถานีทดสอบชิ้นงาน ประกอบไปด้วยโมดูลใช้งาน ดังรูปที่ 2.17



รูปที่ 2.17 แสดง PLC รุ่น 90-70

1. *PWR 710 POWER SUPPLY* ใช้จ่ายไฟให้กับตัว PLC สามารถจ่ายไฟได้สูงสุด 55 watts
2. *CPU 772* ใช้ไมโครโปรเซสเซอร์ 80C186 และมีบอร์ดเพิ่มหน่วยความจำจำนวน 64 KB สามารถกำหนดค่าอินพุตและเอาต์พุตได้สูงสุดอย่างละ 2048 บิต
3. *CMM 741 ETHERNET INTERFACE* ใช้เชื่อมต่อ PLC เข้ากับ Ethernet LAN
4. *BEM 731 GENIUS BUS CONTROLLER* ใช้เชื่อมต่อ PLC เข้ากับ Genius I/O serial bus
5. *I/O Modules* เป็นโมดูลสำหรับอินพุตและเอาต์พุตมีทั้งหมด 4 โมดูล ดังนี้ 8Pt Analog Input (I), 4Pt Analog Output (V&I), 24V 32pt Input และ 24V 16pt Output

PLC รุ่น MICRO 28-Point เป็น PLC ขนาดเล็ก นำมาใช้กับสถานีผลิตชิ้นงานและสถานีแขนกล ดังรูปที่ 2.18

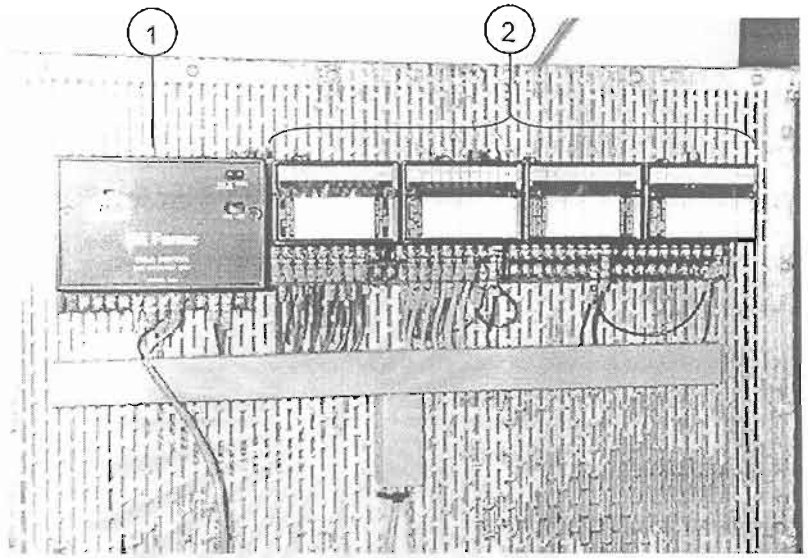


รูปที่ 2.18 แสดง 28-Point Micro PLC

ภายในตัว PLC จะประกอบไปด้วย 3 ส่วนคือ Power Supply board, CPU circuit board และ I/O board มีคุณสมบัติต่าง ๆ ดังนี้

- CPU ใช้ไมโครโปรเซสเซอร์ H8/3003 ทำงานที่ความเร็ว 9.84 MHz
- 256 kX16 flash memory สำหรับระบบปฏิบัติการภายใน และ 64 Kbyte RAM
- สามารถเขียนโปรแกรมใช้งานได้สูงสุด 6K words
- มี 24 V Input จำนวน 16pt และ Relay Output 8pt
- มี 2 พอร์ต สื่อสารอนุกรม สามารถใช้สื่อสาร SNP/SNP-X ได้

Field control รุ่น Genius Bus Interface Unit เป็น I/O Module ที่ใช้กับ Genius Bus นำมาใช้กับสถานีพักชิ้นงานและสถานีแยกชิ้นงานประกอบด้วนส่วนต่าง ๆ ดังรูปที่ 2.19



รูปที่ 2.19 แสดง Field Control

1. *Genius Bus Interface Unit* ใช้เชื่อมต่อการควบคุม I/O Modules เข้ากับ Genius I/O serial bus สามารถรองรับการเชื่อมต่อได้ 2 บัส
2. *I/O Modules* เป็นโมดูลสำหรับอินพุตและเอาต์พุต ประกอบไปด้วย 4 โมดูล ดังนี้ 24V 16pt Input, Relay 8pt Output, 4pt Analog Input (I) และ 8pt Analog Output (I)

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

บทที่ 3
เครือข่ายการสื่อสารของระบบการผลิตอัตโนมัติ

3.1 เครือข่ายการสื่อสารระดับต่าง ๆ ในโรงงานอัตโนมัติ

เครือข่ายการสื่อสารในโรงงานอัตโนมัติแบ่งเป็นระดับต่าง ๆ ดังรูปที่ 3.1 จะเห็นว่ามีการแบ่งระดับการสื่อสารออกเป็น 3 ระดับ ตามลักษณะและจำนวนของข้อมูลดังนี้

Field level เป็นเครือข่ายในระดับต่ำที่สุด ใช้สื่อสารระหว่างอุปกรณ์สนาม (Field device) ที่ต้องทำงานสัมพันธ์กันเช่น ตัวตรวจรู้ (Sensor) และตัวขับ (Actuator) จำนวนข้อมูลในการสื่อสารไม่มาก (Bit level) และต้องการความเร็วในการสื่อสารสูงหรือเป็นการสื่อสารแบบเวลาจริง (Real time)

Cell level เป็นเครือข่ายการสื่อสารที่ใช้รับ-ส่งข้อมูลระหว่างหน่วยหรือใช้สื่อสารระหว่าง PLC จำนวนข้อมูลในการสื่อสารปานกลาง (Byte level) และต้องการความเร็วในการสื่อสารสูง

Factory level เป็นเครือข่ายในระดับบนสุด ใช้เป็นเครือข่ายการสื่อสารเพื่อควบคุมการทำงานของระบบโดยรวมทั้งหมด รวมทั้งสามารถรวบรวม เรียงลำดับ และจัดเก็บข้อมูลจากเครือข่ายต่ำกว่า

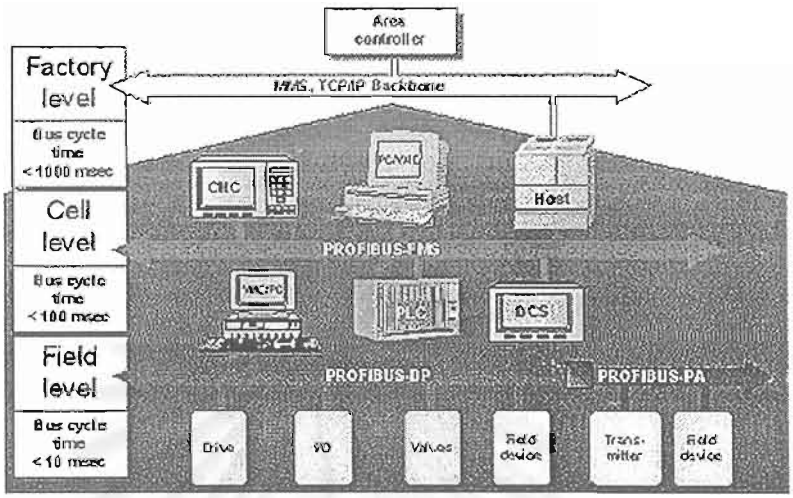
3.2 เทคโนโลยีฟิลด์บัส

ในช่วงปี 1940 เครื่องมือควบคุมกระบวนการอาศัยสัญญาณความดัน 3-15 psi สำหรับตรวจดูอุปกรณ์ควบคุม ต่อมาในช่วงปี 1960 ได้มีการนำมาตรฐานสัญญาณแอนะลอก 4-20 mA มาใช้กับเครื่องมือต่าง ๆ และเนื่องจากการพัฒนาตัวประมวลผลดิจิตอลในช่วงปี 1970 จึงเริ่มมีการใช้คอมพิวเตอร์มาใช้ควบคุมและตรวจดูระบบจากศูนย์ควบคุมกลาง ทำให้เกิดระบบการควบคุมแบบรวมศูนย์ หลังจากนั้นในช่วงปี 1980 ได้มีการพัฒนาอุปกรณ์ควบคุมแบบดิจิตอลโดยใช้ไมโครโปรเซสเซอร์ ซึ่งเรียกว่าอุปกรณ์สมาร์ต (Smart) จึงกระตุ้นให้มีการพัฒนาการส่งสัญญาณแบบใหม่คือฟิลด์บัส เพื่อให้เป็นการสื่อสารอนุกรมดิจิตอลแบบเปิด ไม่ขึ้นอยู่กับผู้ผลิต สามารถสื่อสารแบบ Multi-drop และเป็นระบบการสื่อสารดิจิตอลแบบสองทาง

โปรโตคอล (Protocol) เป็นข้อตกลงระหว่างส่วนต่าง ๆ ของเครือข่ายเกี่ยวกับวิธีการถ่ายโอนข้อมูล องค์ประกอบสำคัญของโปรโตคอลคือ วิธีการสร้างประโยค (Syntax) ความหมายของคำ (Semantic) และจังหวะเวลา (Timing) ซอฟต์แวร์ที่ทำตามโปรโตคอลมาตรฐานสามารถทำให้การสื่อสารระหว่างระบบที่แตกต่างกันเป็นไปได้

มาตรฐานและการทำงานของ ฟิลด์บัส

มาตรฐานฟิลด์บัสเกิดขึ้นโดยองค์กรระหว่างประเทศคือ Instrument Society of America (ISA) และ International Electrotechnical Commission (IEC) มีการประชุมร่วมกันเพื่อกำหนดมาตรฐานเรียกว่า IEC/ISA SP50



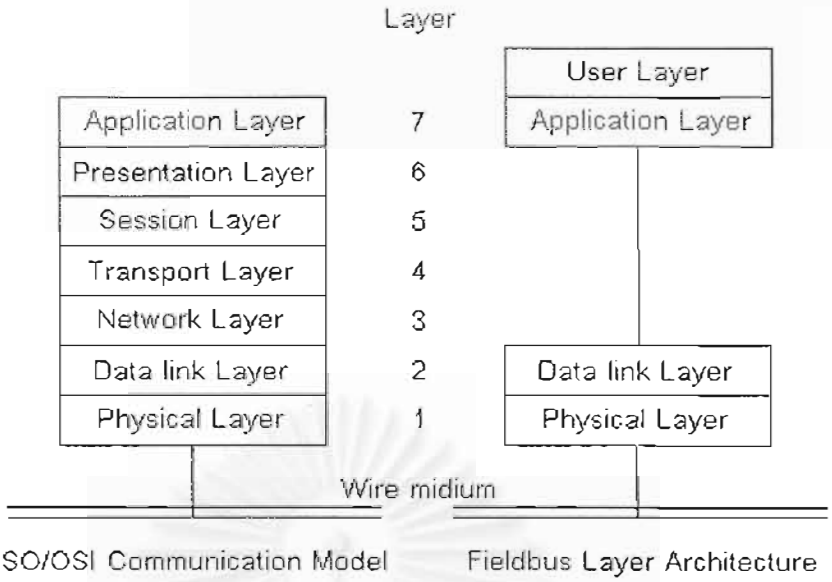
รูปที่ 3.1 แสดงเครือข่ายการสื่อสารระดับต่าง ๆ ในโรงงานอัตโนมัติ

การกำหนดมาตรฐานการสื่อสารโปรโตคอลฟิลด์บัสเป็นไปตาม ISO/OSI Communication Model ซึ่งประกอบไปด้วย 7 Layers แต่สำหรับฟิลด์บัส จะใช้เพียง 3 เลเยอร์ คือ Physical Layer, Data Link Layer และ Application Layer (Layer1, 2 และ 7) Layer3 ถึง Layer7 ไม่นำมาใช้งาน เพื่อให้การสื่อสารเป็นแบบเวลาจริง (Real time) และได้มีการเพิ่มเลเยอร์ใหม่คือ User Layer เข้าไปชั้นบนสุดต่อจาก Application Layer ดังรูปที่ 3.2

เลเยอร์ทั้งสี่มีหน้าที่ดังนี้

- *Physical Layer* ทำหน้าที่ในการรับและแปลงข้อมูลจากเลเยอร์ที่สูงกว่าให้อยู่ในรูปของสัญญาณทางไฟฟ้า เพื่อส่งไปตามสายส่งข้อมูล รวมทั้งสามารถแปลงสัญญาณทางไฟกลับมาเป็นข้อมูลเพื่อส่งต่อให้เลเยอร์สูงขึ้นไป
- *Data Link Layer* ทำหน้าที่จัดเรียงข้อมูลและควบคุมการส่งชุดข้อมูลบนฟิลด์บัสว่าจะส่งไปที่ใด รวมทั้งตรวจสอบความผิดพลาดที่เกิดขึ้น
- *Application Layer* ทำหน้าที่จัดการการแปลงการร้องขอการบริการผู้ใช้จาก User Layer ไปเป็นข้อความหรือชุดข้อมูลที่อุปกรณ์ทุกตัวในเครือข่ายสามารถเข้าใจได้ แล้วส่งผ่านไปยัง Data Link Layer และสามารถทำในทางตรงข้ามได้
- *User Layer* เป็นส่วนของซอฟต์แวร์ที่กำหนดการติดต่อภายใน ทำหน้าที่เก็บฟังก์ชันการควบคุม (Control Function) และฐานข้อมูลของอุปกรณ์วัดและควบคุม เพื่อที่จะจัดสภาวะแวดล้อมในการใช้งานหรือจัดตารางในการปฏิบัติงานและการสื่อสารของ Function Block บนฟิลด์บัส นอกจากนี้ยังดูแลความเรียบร้อยของระบบโดยรวม

หอสมุดกลาง สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย



รูปที่ 3.2 แสดงเลเยอร์ของสถาปัตยกรรม ISO/OSI Communication Model และ Fieldbus

3.3 การสื่อสารอนุกรม

การส่งข้อมูลไปตามสายสื่อสารอนุกรมจะต้องมีการแปลงข้อความ ให้อยู่ในรูปของรหัสของเลขฐานสองเพื่อส่งไปตามสายส่งข้อมูลในรูปของสัญญาณทางไฟฟ้า ทางด้านผู้รับจะต้องมีการตรวจสอบข้อมูลที่ได้รับ ซึ่งจะใช้วิธีการต่าง ๆ แล้วแปลงกลับมาเป็นข้อความได้ การสื่อสารอนุกรมจะประกอบไปด้วย ส่วนต่าง ๆ ดังนี้

การแปลงรหัสข้อมูล

รหัสข้อมูลถูกนำมาใช้เมื่อต้องการส่งตัวเลข ข้อความ สัญลักษณ์ และตัวอักษรควบคุมไปตามสายสื่อสารอนุกรม ซึ่งในปัจจุบันมีหลายวิธีที่จะแปลงข้อความเป็นรหัสข้อมูล แต่มีวิธีหนึ่งที่ใช้กันแพร่หลายมากก็คือ การแปลงเป็น ASCII Code ซึ่งจะแปลงตัวอักษรแต่ละตัวให้อยู่ในรูปของข้อมูลฐานสอง 8 บิต และจะมีการเพิ่มเติมบิตเริ่มต้น และบิตจบ บางครั้งอาจจะมีบิตตรวจสอบหรือพาริตีบิต (Parity bit) เพิ่มขึ้นมาอีกดังรูปที่ 3.3

การส่งข้อมูลแบบไม่ต่อเนื่อง (Asynchronous)

การส่งข้อมูลแบบนี้จะเป็นการส่งข้อมูลแบบไม่ต่อเนื่อง จะส่งเมื่อไรก็ได้ จึงต้องมีบิตเริ่มต้นและบิตจบ ตัวส่งและตัวรับข้อมูล ต้องทำงานที่อัตราการรับ-ส่งข้อมูลเดียวกัน หรือ Baud rate เดียวกัน เมื่อตัวรับตรวจสอบพบบิตเริ่มต้น (Start bit) ซึ่งปกติจะเป็น Logic 0 จะเริ่มมีการเข้าจังหวะตามอัตราการส่งข้อมูลเพื่อตรวจจับบิตข้อมูลที่ตามมา จนกระทั่งตรวจสอบพบบิตจบ (Stop bit) ซึ่งปกติเป็น Logic 1 อย่างน้อยคงอยู่ 1 บิต จึงจะครบการส่งข้อมูลหนึ่งตัวอักษร ดังรูปที่ 3.4

การตรวจสอบความผิดพลาดของการรับ-ส่งข้อมูล

การตรวจสอบความผิดพลาดเป็นการลดความผิดพลาดในการรับ-ส่งข้อมูล และเป็นการยืนยันความน่าเชื่อถือของข้อมูล ซึ่งมีวิธีที่จะตรวจสอบได้หลายวิธี ดังนี้

- *Parity Checking* มีการกำหนดการตรวจสอบคือ odd, even หรือไม่ตรวจสอบบิตพาริตี ถูกกำหนดขึ้นโดยผู้ส่งและตรวจสอบโดยผู้รับ ถ้ามีการกำหนดการตรวจสอบแบบ odd จำนวนรวมของบิตที่เป็น 1 (รวมบิตพาริตีด้วย) ต้องเป็นเลขคี่ แต่ถ้ากำหนดการตรวจสอบแบบ even จำนวนรวมของบิตที่เป็น 1 (รวมบิตพาริตี) ต้องเป็นเลขคู่ และถ้าไม่ตรวจสอบบิตพาริตีก็จะไม่มีการใส่ค่าในบิตพาริตี ดังรูปที่ 3.5

- *Block Check Code Checking (BCC)* ใช้ตรวจสอบความผิดพลาดในการรับ-ส่งข้อมูลเมื่อใช้ SNP/SNP-X Protocol โดยจะมีการเพิ่ม BCC ความยาว 1 ไบต์ เข้าไปท้ายสุดของ SNP message เพื่อใช้ตรวจสอบข้อความภายใน message การคำนวณค่าของ BCC จะถูกคำนวณจากไบต์ทั้งหมดใน message ยกเว้นไบต์สุดท้ายซึ่งเป็นไบต์ของ BCC เอง ซึ่งแบบแผนของการคำนวณค่า BCC เป็นดังนี้

```
Initialize Temp BCC byte to zero
For first to last byte in message range to be checked
        Exclusive – OR Temp BCC byte with message byte
        Rotate Temp BCC byte left by 1 bit (with warp around)
        Bump to next message byte
End for
Temp BCC byte now contains final BCC valve
```

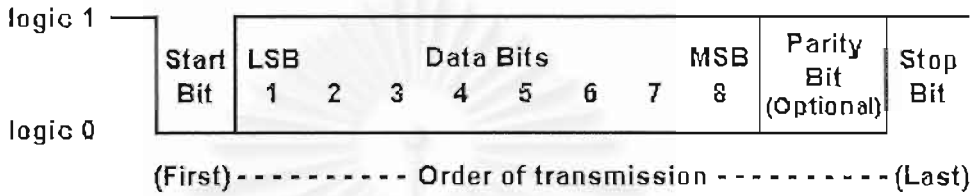
- *Longitudinal Redundancy Checking (LRC)* เป็นวิธีตรวจสอบความผิดพลาดตลอดทั้งบล็อกข้อมูลโดยจะเพิ่ม LRC เข้าไปตอนท้ายของบล็อกข้อมูล เพื่อตรวจสอบ พาริตีของตัวอักษรทุกตัวภายในเฟรม การคำนวณค่าของ LRC จะใช้การ exclusive OR (XOR) ในแต่ละไบต์ของข้อมูลในบล็อก ดังรูปที่ 3.6

- *Cyclic Redundancy Checking (CRC)* เป็นการตรวจสอบความผิดพลาด โดยจะคำนวณและเติม CRC ไว้ตอนท้ายของบล็อกข้อมูล วิธีการคำนวณ CRC เป็นดังนี้

1. บิตข้อมูลที่ต้องการส่งจะถูกคูณด้วย จำนวนของบิตใน CRC
2. นำผลจากข้อ 1 ไปหารด้วย Generating polynomial โดยใช้ Modulo 2 with no carries เศษเหลือที่ได้จากการหาร คือ CRC
3. ผลลัพธ์จากการหารจะไม่นำมาใช้ โดยจะใช้เศษเหลือ หรือ CRC เพิ่มต่อเข้าไปในบิตข้อมูลที่จะส่งและส่งข้อมูลไปพร้อม CRC
4. ผู้รับจะหารข้อมูลที่ได้รับ (รวม CRC) ด้วย Generating polynomial ซึ่งถ้าเศษมีค่าเป็น 0 แสดงว่าการส่งข้อมูลไม่มีความผิดพลาด

		MSB			Data Bits				LSB	
10	9	8	7	6	5	4	3	2	1	0
Stop	Parity (Optional)									Start

รูปที่ 3.3 แสดงรหัส ASCII ที่ใช้แทนตัวอักษรหนึ่งตัว



รูปที่ 3.4 แสดงสัญญาณทางไฟฟ้าของการส่งข้อมูลแบบไม่ต่อเนื่อง

Parity Bit (odd)	Received Data Byte 8	7	6	5	4	3	2	1
1	0	1	0	0	0	0	0	1

Parity Bit (even)	Received Data Byte 8	7	6	5	4	3	2	1
0	0	1	0	0	0	0	0	1

รูปที่ 3.5 แสดงค่าของบิตพาริตี ในแบบ odd และ even

Parity Bit (odd)	Received Data Byte 8	7	6	5	4	3	2	1	
0	0	0	0	0	0	0	0	1	- 1st data char. transmitted
0	0	0	0	0	0	0	1	0	- 2nd data char. transmitted
	0	0	0	0	0	0	1	1	- XOR result of 1st and 2nd data charactors
1	0	0	0	0	0	1	0	1	- 3rd data char. transmitted
	0	0	0	0	0	1	1	0	- LRC=XOR of previous XOR and 3rd data char.

รูปที่ 3.6 แสดงการคำนวณและใส่ค่า LRC

สายสื่อสารอนุกรม

สายสื่อสารเป็นตัวกลางในการรับ-ส่งข้อมูล ซึ่งคุณสมบัติของสายสื่อสารจะขึ้นอยู่กับความต้องการของผู้ใช้ และมาตรฐานการเชื่อมต่อสายสัญญาณของสายสื่อสารนั้น ๆ รูปแบบของการสื่อสาร แบ่งออกได้เป็น 3 รูปแบบ คือ

1. *Simplex* ข้อมูลสามารถส่งไปได้ในทิศทางเดียวจากผู้ส่งไปยังผู้รับ
2. *Half-duplex* ข้อมูลสามารถส่งได้ทั้งสองทิศทาง (ไป-กลับ) แต่คนละเวลากัน ไม่สามารถส่งข้อมูลสองทิศทางในเวลาเดียวกันได้
3. *Full-duplex* ข้อมูลสามารถส่งได้ทั้งสองทิศทางพร้อม ๆ กัน

รูปแบบของการสื่อสารจะขึ้นอยู่กับมาตรฐานของการเชื่อมต่อ ซึ่งจะกำหนดลักษณะของสัญญาณ สายเชื่อมต่อสัญญาณและหัวต่อสายสัญญาณ การกำหนดเข็มของหัวต่อสายสัญญาณ และการควบคุมลำดับในการส่งสัญญาณระหว่างอุปกรณ์ มาตรฐานของการเชื่อมต่อมีอยู่หลายมาตรฐาน เช่น

- *RS-232* เป็นมาตรฐานการต่อถึงกันระหว่าง Data Terminal Equipment (DTE) เช่น เครื่องพิมพ์ กับ Data Communications Equipment (DCE) เช่น โมเด็ม เพื่อส่งข้อมูลผ่านโครงข่ายโทรศัพท์ ซึ่งเราสามารถนำมาใช้สื่อสารในระยะทางสั้น ๆ ได้โดยไม่ใช้โมเด็ม การส่งสัญญาณทางไฟฟ้าของมาตรฐาน RS-232 จะเป็นแบบไม่สมดุลย์ (Unbalanced) ก็คือ แรงดันของวงจรรับและส่งสัญญาณจะถูกวัดเทียบกับคอมมอนกราวนด์ (Common ground) หรือใช้กราวนด์ร่วมกัน การนำไปใช้งานสามารถใช้ส่งข้อมูลด้วยอัตราการส่งสัญญาณสูงสุด 20 kilobits/sec ในระยะทางสูงสุดไม่เกิน 15 เมตร หัวต่อ RS-232 จะใช้แบบ 25-pin-D-type และมีการกำหนดสัญญาณทางไฟฟ้าดังตารางที่ 3.1 และการรับ-ส่งข้อมูลระหว่างอุปกรณ์เป็นดังรูปที่ 3.7
- *RS-449, RS-422 และ RS-485* เป็นกลุ่มของมาตรฐานที่เกิดขึ้นใหม่ ซึ่งสามารถให้อัตราส่งข้อมูลและระยะทางในการส่งข้อมูล สูงขึ้น RS-422 และ RS-485 เป็นมาตรฐานที่กำหนดลักษณะของการเชื่อมต่อทางไฟฟ้า การกำหนดเข็มของหัวต่อสาย สายสัญญาณและหัวต่อสาย และการควบคุมลำดับการส่งสัญญาณ ส่วนมาตรฐาน RS-449 จะใช้ร่วมกับ RS-422 และ RS-485 ลักษณะของการส่งสัญญาณของ RS-422 และ RS-485 จะเป็นแบบสมดุลย์ (Balanced) หรือใช้ค่าความแตกต่างของแรงดันในการส่งสัญญาณ ซึ่งจะแยกจากกราวนด์ทำให้สามารถส่งข้อมูลได้เร็วขึ้นในระยะทางที่ไกลขึ้น ตัวอย่างลักษณะของการสื่อสารโดย RS-485 เป็นดังนี้
    - ความยาวของสายส่งสัญญาณได้สูงสุดเท่ากับ 1200 เมตร
    - อัตราการส่งข้อมูลสูงสุดเท่ากับ 1 Megabits/sec ที่ความยาว 1200 เมตร และ 10 Megabits/sec ที่ความยาว 12 เมตร
    - ใช้หัวต่อแบบ 25-pin-D-type

- การกำหนด Logic 0 หรือ 1 จะใช้ความแตกต่างของแรงดันดังนี้

  Logic 0: วงจร A มีแรงดัน + 200 mV ถึง 6 V เมื่อเทียบกับวงจร B

  Logic 1: วงจร A มีแรงดัน – 200 mV ถึง –6 V เมื่อเทียบกับวงจร B

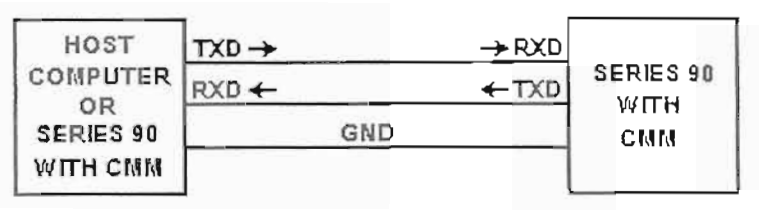
ตารางที่ 3.1 แสดงมาตรฐานของหัวต่อ RS-232

Pin #	Function	Abbrev.	Type	Direction
1	Protective Ground	PROT GND	-	-
2	Transmit data	TXD	Data	From DTE
3	Receive data	RXD	Data	To DTE
4	Request to send	RTS	Control	From DTE
5	Clear to send	CTS	Control	To DTE
6	Data Set Ready	DSR	Control	To DTE
7	Signal Ground	GND	-	-
8	Recvd Line Signal Detect or Data Carrier Detect	RLSD or DCD	Control	To DTE
20	Data Terminal Ready	DTR	Control	From DTE

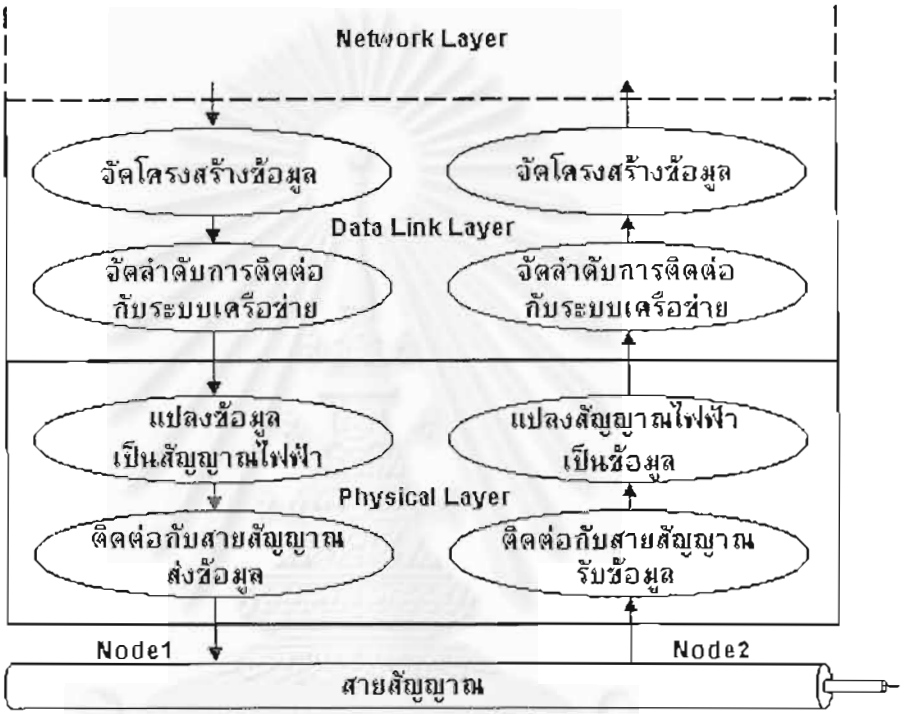
3.4 อีเทอร์เน็ต (Ethernet)

เครือข่ายอีเทอร์เน็ต เป็นแบบการเชื่อมต่อสำหรับคอมพิวเตอร์ ที่ได้รับความนิยมมากในปัจจุบัน เนื่องจากให้การส่งข้อมูลความเร็วสูงในราคาประหยัด การพัฒนาแบบการเชื่อมต่อเครือข่ายอีเทอร์เน็ตเริ่มจากบริษัท Xerox เป็นการกำหนดมาตราฐานของระบบเครือข่ายคมนาคมสำหรับเครื่องคอมพิวเตอร์จำนวนมากเชื่อมต่อกันและใช้สายสัญญาณร่วมกัน อีเทอร์เน็ตครอบคลุมฟิสิคอล และดาต้าลิงก์เลเยอร์ ตามแบบจำลองของ ISO

การสื่อสารข้อมูลของอีเทอร์เน็ต จะใช้วิธีการ CSMA/CD (Carrier Sense Multiple Access/Collision Detect) ซึ่งนิยมใช้สำหรับการต่อระบบเครือข่ายในลักษณะบัสโดยทุก ๆ โหนด (Node) ที่ต้องการส่งข้อมูลจะต้องเช็คดูก่อนว่าสัญญาณว่างหรือเปล่าหรือว่ากำลังอยู่ในระหว่างการใช้งาน และถ้าพบว่าช่องสัญญาณว่างก็จะเริ่มส่งสัญญาณทันที แต่เนื่องจากว่าสัญญาณเคลื่อนที่จากจุดหนึ่งไปยังอีกจุดหนึ่งจะใช้เวลา ดังนั้นหากจุดต่อจุดหนึ่งเริ่มส่งสัญญาณในขณะที่อีกจุดหนึ่ง ซึ่งอยู่ห่างออกไปยังไม่สามารถเช็คได้ว่ามีผู้ใช้ช่องสัญญาณอยู่ ก็จะคิดว่าช่องสัญญาณว่าง ก็เริ่มส่งสัญญาณออกมาทำให้เกิดการชนกันของสัญญาณ (Collision) ซึ่งเมื่อจุดต่อที่ส่งสัญญาณออกมาพบว่ามีการชนกันของสัญญาณเกิดขึ้น ก็จะหยุดส่งสัญญาณแล้วรอเช็คใหม่ว่าช่องสัญญาณว่างหรือยังถ้าว่างก็จะส่งใหม่อีกครั้ง การรอเพื่อส่งสัญญาณใหม่ในแต่ละจุดจะไม่เท่ากันเพื่อป้องกันการชนซ้ำอีก โครงสร้างการสื่อสารของอีเทอร์เน็ตเป็นดังรูปที่ 3.8



รูปที่ 3.7 แสดงการรับ-ส่งข้อมูลระหว่างอุปกรณ์ด้วยมาตรฐาน RS-232 (ไม่ใช้ Flow control)



รูปที่ 3.8 แสดงโครงสร้างของเครือข่ายอีเทอร์เน็ต

3.5 โปรโตคอลจีเนียสบัส (Genius Bus Protocol)

จีเนียสบัสถูกพัฒนาขึ้นโดยบริษัท GE FANUC เพื่อให้เป็นการสื่อสารที่ทนทาน ในสภาวะแวดล้อมภายในโรงงาน สายบัสจะใช้สายคู่ดีเกลียวหุ้มฉนวน (Single shielded twisted pair) สามารถต่อกับอุปกรณ์ได้สูงสุด 32 ตัว ในหนึ่งบัส อัตราการรับ-ส่งข้อมูลสูงสุด 153.6 Kbits/s ข้อมูลที่รับ-ส่งจะมีการตรวจสอบความผิดพลาดโดยวิธี Cyclic Redundancy Checksum การสื่อสารบนจีเนียสบัส สามารถแบ่งออกได้เป็น 3 ลักษณะ คือ

1. I/O Service
   - Input จะถูก Broadcast ไปที่ CPU ทุกตัวบนบัสทุก ๆ Bus scan
   - Output จะถูกส่งจาก CPU ไปที่ I/O Block ที่กำหนด ทุก ๆ Bus scan

2. Datagrams
   - ในหนึ่ง Bus scan สามารถส่งได้หนึ่ง Datagram
   - มีการตรวจสอบผลของการส่ง และส่งข้อมูลซ้ำได้เอง
   - สามารถส่งอัตโนมัติจากอุปกรณ์หนึ่งไปยังอุปกรณ์อื่น
   - สามารถส่งจากโปรแกรมบนคอมพิวเตอร์ไปยัง I/O Block หรือ CPU อื่น ๆ บนบัส
   - สามารถส่งข้อมูลได้สูงสุดครั้งละ 128 bytes ไปยัง CPU ตัวอื่น ๆ
3. Global data
   - ไม่ต้องการการเขียนโปรแกรมเพื่อรับ-ส่งข้อมูล
   - ข้อมูลจะถูก Broadcast โดยอัตโนมัติทุก ๆ Bas scan
   - ในแต่ละ CPU สามารถส่งข้อมูลได้สูงสุด 128 bytes
   - CPU ทุกตัวสามารถรับข้อมูลที่ Broadcast อยู่บนบัส

วิธีการสื่อสารระหว่างอุปกรณ์ด้วยจีเนียสบัส

ในแต่ละบัส (Bus) สามารถกำหนดอุปกรณ์ได้ทั้งหมด 32 ตัว การกำหนดจะใช้ Block numbers หรือ Serial bus address ซึ่งจะถูกตั้งค่าลงในอุปกรณ์ทุกตัว ดังรูปที่ 3.9

ในเครือข่ายจีเนียสบัส อุปกรณ์ที่เป็น I/O Block จะส่งข้อมูลแบบ Broadcast ไปยัง CPU ทุกตัวบนบัส และอุปกรณ์ที่เป็น Bus controller จะสามารถรับข้อมูลที่ Broadcast และสามารถส่งข้อมูลไปยัง I/O Block แบบเจาะจงได้

การติดต่อสื่อสารบนบัสจะใช้วิธีการที่เรียกว่า Token passing โดยอุปกรณ์แต่ละตัวจะได้รับโทเก็น (Token) หรือ สิทธิในการส่งข้อมูล และเมื่อส่งข้อมูลเสร็จแล้วจะผ่านโทเก็น ไปยังอุปกรณ์ตัวอื่น ๆ บนบัสตามลำดับจาก Address 0 ไปจนถึง 31 แล้วจึงวนกลับไปเริ่มที่ 0 ใหม่ ดังรูปที่ 3.10 ในกรณีของการส่งข้อมูลอินพุตจาก I/O Block เมื่อ I/O Block นั้นได้รับโทเก็น จะ Broadcast ค่าของอินพุต ทั้งหมดไปยังทุก ๆ CPU บนบัสจากนั้นจะส่งค่าโทเก็นต่อไปยังอุปกรณ์ถัดไป ดังรูปที่ 3.11 ในกรณีของการส่งข้อมูลเอาต์พุตจาก CPU ผ่านทาง Bus controller เมื่อ Bus controller ได้รับโทเก็น จะส่งค่าเอาต์พุต ไปยัง I/O Block โดยตรง โดยไม่ Broadcast จากนั้นจะส่งค่าโทเก็นต่อไปยังอุปกรณ์ถัดไป ดังรูปที่ 3.12

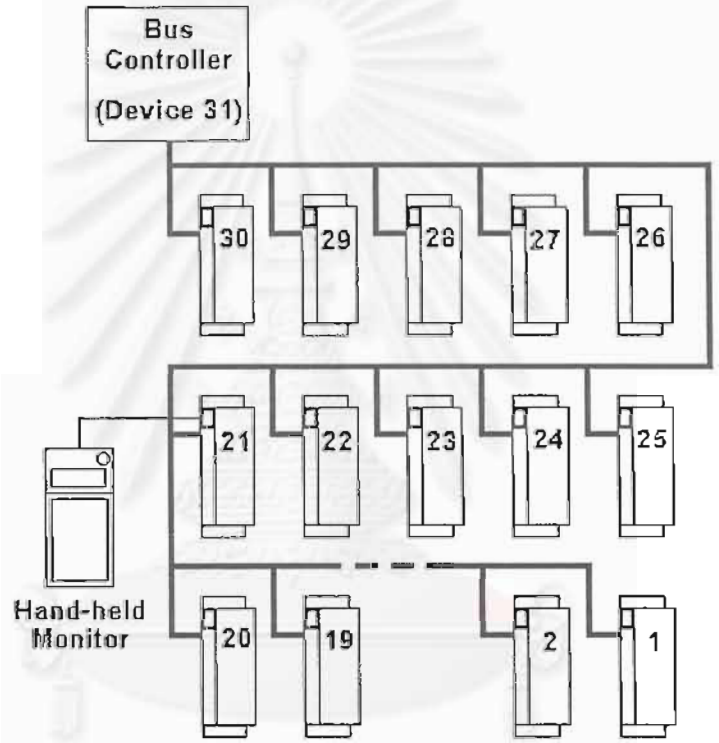
การใช้งานจีเนียสบัส

จีเนียสบัสสามารถใช้งานเป็นได้ทั้งบัสสำหรับสื่อสาร และบัสสำหรับควบคุม I/O ผ่านอุปกรณ์ I/O Block โดยบัสทั้งสองชนิดอาจรวมอยู่บนบัสเดียวกัน หรือแยกบัสกันก็ได้ ดังรูปที่ 3.13

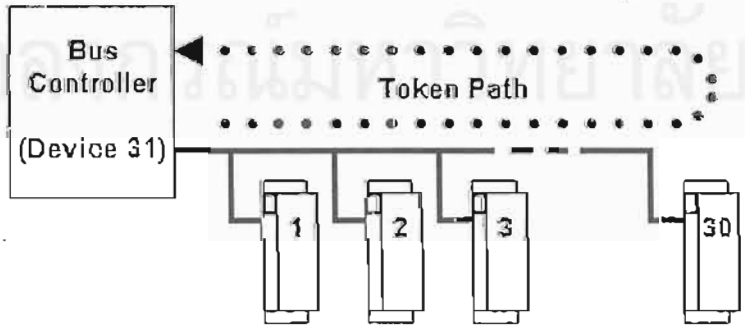
การใช้จีเนียสบัสในการรับ-ส่งข้อมูล สามารถรับ-ส่งได้ 2 แบบ คือ

1. *Individual Datagram* สามารถส่งจาก PLC หรือคอมพิวเตอร์ ไปยัง I/O Block หรือ CPU อื่น ๆ บนบัส ตัวอย่างเช่น ส่ง Datagram เพื่อเปลี่ยนแปลงค่า Configuration ของ I/O Block, ตรวจสอบสถานะของ I/O Block หรือใช้อ่านค่าข้อมูลของอินพุตบน I/O Block ซึ่งในหนึ่ง Datagram สามารถรับ-ส่งข้อมูลได้สูงสุด 128 bytes

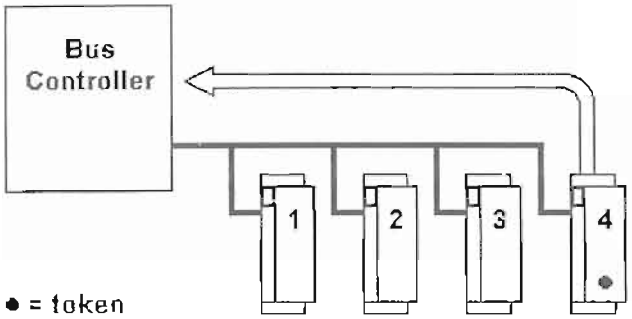
2. *Global Data* ใช้รับ-ส่งข้อมูลอินพุตและเอาต์พุตระหว่าง CPU กับ I/O Block หรือใช้รับ-ส่งข้อมูลในหน่วยความจำระหว่าง CPU ด้วยกัน Global Data ต่างจาก Datagram ตรงที่ข้อมูลสามารถรับ-ส่งได้โดยอัตโนมัติในทุก ๆ Bus scan สามารถรับ-ส่งข้อมูลได้สูงสุด 128 bytes หรือสามารถกำหนดค่า Global Data ได้สูงสุด 128 bytes



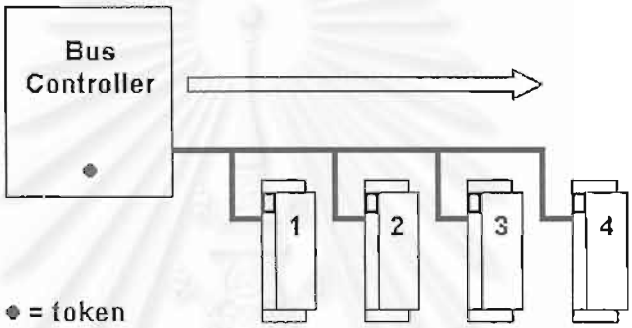
รูปที่ 3.9 แสดงการใช้งาน Genius Bus



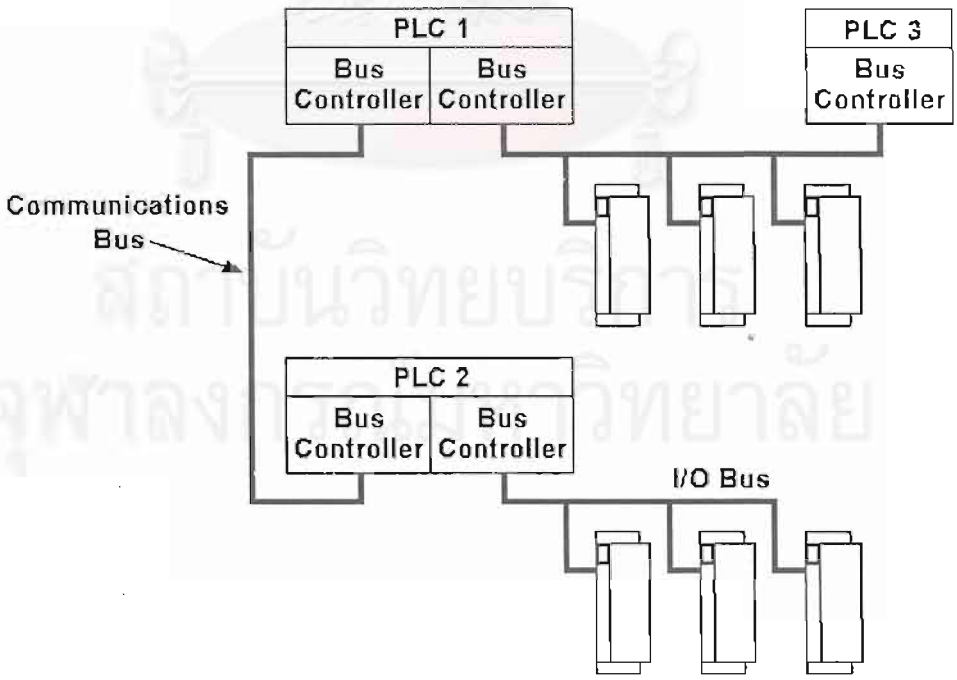
รูปที่ 3.10 แสดงวิธีการสื่อสารโดยใช้ Token passing



รูปที่ 3.11 แสดงการส่งค่าอินพุตของ I/O Block



รูปที่ 3.12 แสดงการส่งค่าเอาต์พุตจาก CPU



รูปที่ 3.13 แสดงบัสสำหรับสื่อสาร และบัสสำหรับควบคุม I/O

3.6 โปรโตคอล SNP/SNP-X

SNP ย่อมาจาก Serial Ninety Protocol เป็นโปรโตคอลสำหรับการสื่อสารอนุกรม ซึ่งถูกพัฒนาโดย GE FANUC เพื่อใช้สื่อสารระหว่าง PLC Series 90 โดยจะบรรจุ SNP Protocol ไว้ใน EPROM และใช้เป็นโปรโตคอลมาตรฐาน สำหรับทุก ๆ โมเดลใน PLC Series 90 สำหรับ SNP-X Protocol เป็นโปรโตคอลเพิ่มเติมจาก SNP Protocol เพื่อให้ใช้งานได้ง่าย และมีประสิทธิภาพเพิ่มมากขึ้น แต่ว่าจะมีคำสั่งใช้งานน้อยกว่า ซึ่ง SNP-X Protocol เป็นส่วนเพิ่มเติมของ SNP Protocol และจะใช้งานได้เมื่อ SNP ได้ถูกตั้งค่าและใช้งานอยู่แล้ว

การรับ-ส่งข้อมูลโดยใช้ SNP และ SNP-X Protocol จะใช้ระบบ Master/Slave ซึ่งการรับ-ส่งจะเป็นแบบ Half-duplex และใช้มาตรฐาน RS-485 และ RS-232 ในการเชื่อมต่อสัญญาณทางไฟฟ้า การตรวจสอบความผิดพลาดจะใช้วิธี Block Check Code Checking อุปกรณ์บนบัสแบ่งออกเป็น 2 ชนิด คือ

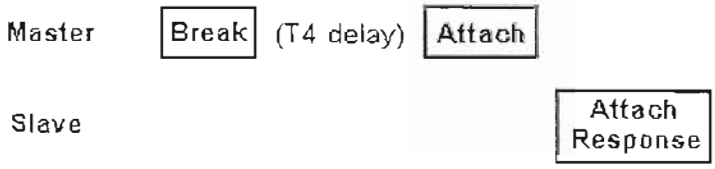
- *Master* เป็นตัวกำหนดการสื่อสารบนบัส โดย Master สามารถส่งข้อความสื่อสารไปยังอุปกรณ์ Slave ที่อยู่บนบัสเพื่อร้องขอข้อมูลจาก Slave ได้
- *Slave* เป็นอุปกรณ์ที่เชื่อมต่อกับ Master ทำหน้าที่รับและตอบสนองการร้องขอจาก Master ซึ่งโดยปกติพอร์ต (port) สื่อสาร SNP บนตัว PLC จะเป็น Slave และใช้มาตรฐาน RS 485

การสื่อสารด้วย SNP

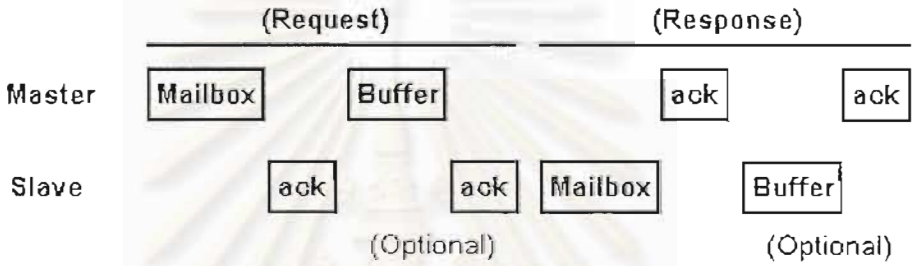
ข้อมูลที่สื่อสารแบ่งออกเป็น 2 ส่วนคือส่วนแรกคือ การสร้างส่วนติดต่อสื่อสาร โดยใช้คำสั่ง Attach หรือ Long Attach ส่วนที่สองคือ ส่วนรับ-ส่งข้อมูลระหว่าง Master และ Slave จะใช้คำสั่งอ่าน (Read) หรือเขียน (Write) ซึ่งมีหลายคำสั่งตามชนิดของข้อมูลที่รับ-ส่ง

ในการสร้างส่วนติดต่อสื่อสาร อุปกรณ์ Master จะส่ง Break sequence ไปยังอุปกรณ์ทุกตัวตลอดบัส เมื่ออุปกรณ์ Slave ได้รับ Break sequence จะทำการยกเลิกการสื่อสาร SNP ที่มีอยู่และเตรียมรับ Attach massage จาก Master หลังจากที่ Master ส่ง Break sequence ไปแล้ว จะรอเป็นเวลา T4 (สามารถตั้งค่า T4 ได้) แล้วจึงส่ง Attach massage ซึ่งประกอบด้วย SNP ID ของอุปกรณ์ Slave ที่จะสร้างส่วนติดต่อสื่อสารด้วย จากนั้น Slave ที่ถูกระบุตาม SNP ID จะส่งสัญญาณตอบสนอง และ Attach Response message กลับไปยัง Master ดังรูปที่ 3.14เมื่อสร้างส่วนติดต่อสื่อสารระหว่าง Master กับ Slave ที่ต้องการแล้ว Master จะสามารถส่งการร้องขอข้อมูลไปยัง Slave ซึ่ง Slave จะส่งข้อมูลตอบกลับไปยัง Master การส่งการร้องขอข้อมูลหรือการตอบกลับจะประกอบไปด้วย Mailbox massage และอาจจะตามด้วย Buffer massage (ในกรณีที่ข้อมูลมีจำนวนมาก) ทั้ง Master และ Slave เมื่อได้รับข้อมูลจะตรวจสอบความผิดพลาดและจะส่งสัญญาณ Acknowledge กลับไปยังผู้ส่ง ดังรูปที่ 3.15 ในกรณี เมื่อตรวจสอบพบความผิดพลาดของข้อมูลที่ได้รับ ผู้รับจะส่ง Error Code กลับไปยังผู้ส่ง และผู้ส่งจะพยายามส่งข้อมูลใหม่อีกครั้ง สำหรับ SNP master จะอนุญาตให้มีการส่งข้อมูลซ้ำได้สูงสุด

สองครั้ง ถ้าข้อมูลที่ส่งหรือรับยังตรวจสอบพบความผิดพลาด Master จะยกเลิกการสื่อสารนั้น และทำงานข้ามไป โดยจะแสดงค่าความผิดพลาดใน Major Error Code 12 และ Minor Error Code 32



รูปที่ 3.14 แสดงการสร้างส่วนติดต่อสื่อสารระหว่าง Master กับ Slave ของ SNP



รูปที่ 3.15 แสดงการส่งการร้องขอข้อมูลจาก Master และ Slave ส่งข้อมูลตอบกลับของ SNP

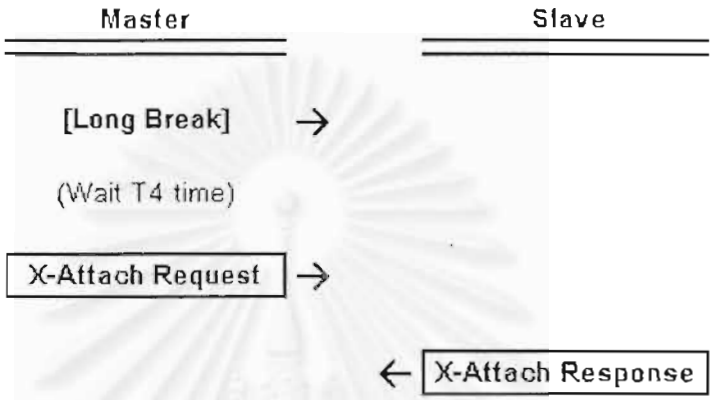
การสื่อสารด้วย SNP-X

ลักษณะของการสื่อสารจะคล้ายกับ SNP คือต้องสร้างส่วนติดต่อสื่อสาร หรือ SNP-X Communication Session ระหว่าง Master กับ Slave ที่ต้องการรับ-ส่งข้อมูล จากนั้นจึงจะสามารถรับ-ส่งข้อมูลกันได้ แต่ในกรณีของ SNP-X จะรวมส่วนติดต่อสื่อสารและส่วนรับ-ส่งข้อมูลอยู่ในคำสั่งเดียวกัน เช่นในกรณีของคำสั่งสำหรับอ่านข้อมูลจาก Slave ใน SNP จะต้องใช้คำสั่ง Attach เพื่อสร้างส่วนติดต่อสื่อสารจากนั้นจึงใช้คำสั่ง Read เพื่ออ่านข้อมูลจาก Slave แต่ใน SNP-X เราสามารถใช้คำสั่ง X-Read คำสั่งเดียวในการสร้างการติดต่อสื่อสารและอ่านข้อมูลจาก Slave ความสามารถที่เพิ่มขึ้นอีกอย่างของ SNP-X คือสามารถส่งข้อมูลแบบ Broadcast ได้เมื่อต้องการส่งข้อมูลไปยังทุก ๆ Slave บนบัส

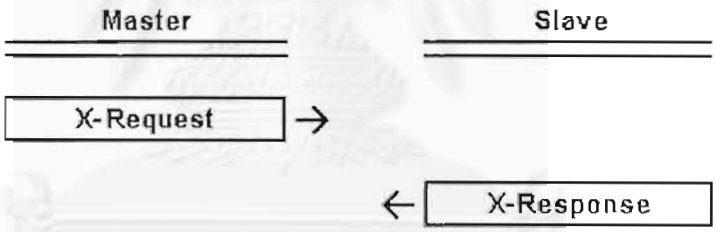
ในส่วนของการสร้างส่วนติดต่อสื่อสาร Master จะส่งสัญญาณ Long Break ไปบนบัส เพื่อให้ Slave ทุกตัวยกเลิกการสื่อสารที่มีอยู่และเตรียมรับคำสั่ง X-Attach โดย Master จะรอเป็นเวลา T4 จึงส่ง X-Attach message ไปยัง Slave ที่ถูกระบุตาม SNP ID ซึ่ง Slave จะส่ง X-Attach Response กลับไปยัง Master ดังรูปที่ 3.16

ในส่วนของการรับ-ส่งข้อมูล เมื่อสร้างส่วนติดต่อสื่อสารระหว่าง Master กับ Slave แล้ว Master สามารถส่งการร้องขอข้อมูล หรือ X-Request ไปยัง Salve ซึ่ง Salve จะส่งข้อมูลตอบกลับ หรือ X-Response กลับไปยัง Master ดังรูปที่ 3.17 ในกรณีที่ข้อมูลที่รับ-ส่งมากกว่า

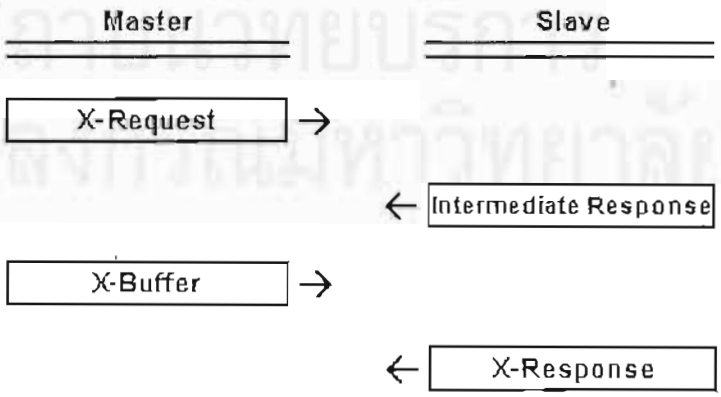
2 bytes จะไม่สามารถรับ-ส่งให้เสร็จเรียบร้อยได้ในครั้งเดียวจึงต้องมีการเพิ่มการส่ง X-Buffer ต่อเนื่องจาก X-Request เพื่อให้สามารถรับ-ส่งข้อมูลได้ทั้งหมด ดังรูปที่ 3.18 การตรวจสอบความผิดพลาดของข้อมูลที่ได้รับ เมื่อตรวจสอบพบความผิดพลาดของการรับข้อมูล ผู้รับจะส่ง Error Code กลับมาให้ผู้ส่ง และผู้ส่งจะยกเลิกการสื่อสารนั้น และทำงานข้ามไป โดยจะแสดงความผิดพลาดผ่าน Major Error Code 15 และ Minor Error Codes 32-35



รูปที่ 3.16 แสดงการสร้างส่วนติดต่อสื่อสารของ SNP-X

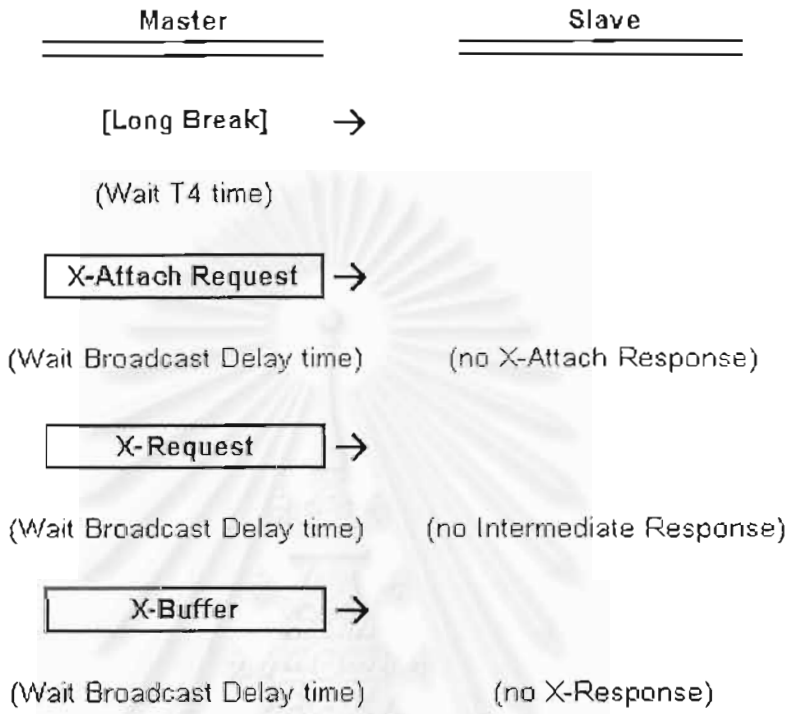


รูปที่ 3.17 แสดงส่วนการรับ-ส่งข้อมูลของ SNP-X



รูปที่ 3.18 แสดงการรับ-ส่งข้อมูลมากกว่า 2 bytes ของ SNP-X

ในการส่งข้อมูลแบบ Broadcast จะใช้ได้กับคำสั่ง X-Write หรือส่งข้อมูลออกได้เท่านั้น ซึ่งเมื่อ Master ส่งข้อมูลไปแล้วจะไม่มีการ Response จาก Slave ซึ่ง Slave ทุกตัวจะได้รับข้อมูลโดย Master จะรอเป็นเวลาเท่ากับ Broadcast Delay Time หลังจากนั้นจะทำงานต่อไป ดังรูปที่ 3.19



รูปที่ 3.19 แสดงการส่งข้อมูลแบบ Broadcast ไปยัง Slave ทุกตัวของ SNP-X

3.7 โปรโตคอล TCP/IP

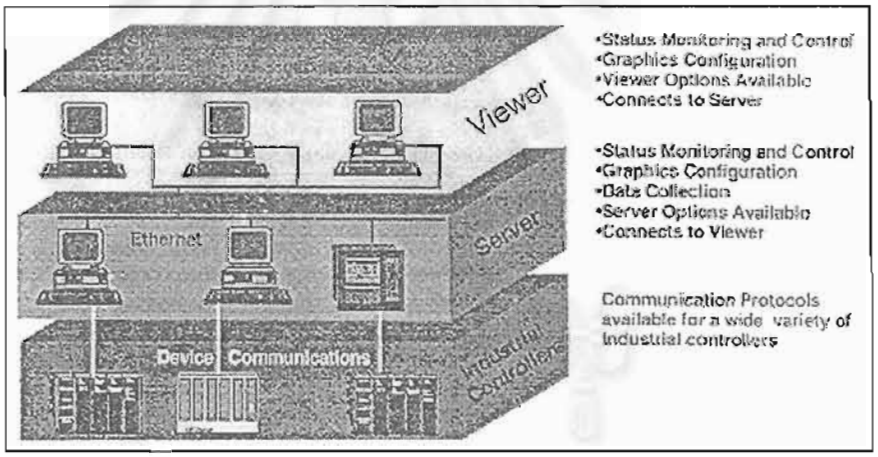
TCP/IP ย่อมาจาก Transmission Control Protocol/Internet Protocol เป็นโปรโตคอลที่เกิดขึ้น เพื่อใช้แก้ปัญหาการเชื่อมโยงคอมพิวเตอร์ จำนวนมาก ๆ ที่มีระบบต่างกัน สามารถแบ่งออกได้เป็น 2 ส่วนคือ ส่วน IP ใช้แนวความคิดที่เรียกว่า อินเตอร์เน็ตแอดเดรส (Internet address) ซึ่งเป็นตัวเลข 32 บิต ที่ถูกกำหนดให้กับทุกโหนดบนเครือข่าย แอดเดรส แบ่งออกได้เป็นหลายประเภทตามขนาดของเครือข่าย ในการอ้างถึงแอดเดรส สามารถเขียนให้อยู่ในรูปตัวเลขฐานสิบ 4 ตัว ดังตัวอย่างเช่น 128.22.5.13 ตัวเลขแต่ละตัวจะบอกถึงเครือข่ายหลักและเครือข่ายย่อยที่โหนดอยู่ การสื่อสารจะต้องรู้อินเตอร์เน็ตแอดเดรสของแต่ละโหนดเพื่อให้สื่อสารกันได้ ส่วน TCP จะทำงานเมื่อแพกเก็ตถูกจัดส่งไปยังอินเตอร์เน็ตแอดเดรสที่ถูกต้อง ซอฟต์แวร์ที่ทำตามมาตรฐาน TCP จะสร้างการเชื่อมต่อซึ่งกันและกัน และจัดการกับการสื่อสารในการส่งข้อมูล TCP จะรอคอยการตอบสนอง และสามารถส่งข้อมูลใหม่แทนบล็อกที่เสีย หรือสูญหายไป

โดยปกติแล้ว TCP/IP จะใช้งานอยู่บนเครือข่ายอีเทอร์เน็ต ข้อมูลที่รับ-ส่งจะถูกหุ้มไว้ในแพกเก็ตอีเทอร์เน็ต ไปยังแต่ละเครื่อง ซึ่งเครื่องที่ได้รับจะจัดการกับแพกเก็ตตามมาตรฐาน TCP/IP และทำให้ข้อมูลนั้นนำไปใช้ได้โดยโปรแกรมระดับสูงกว่า เช่น โปรแกรมสำหรับรับ-ส่งไปรษณีย์อีเล็กทรอนิกส์

3.8 ซอฟต์แวร์ CIMPLICITY HMI

CIMPLICITY HMI เป็นซอฟต์แวร์ที่ใช้ในการตรวจดู (Monitor) และควบคุม (Control) ระบบ สามารถใช้ควบคุมเครือข่ายในทุกระดับในสายการผลิต และมีพื้นฐานการทำงานแบบไคลแอนต์/เซริฟเวอร์ ทำให้ง่ายต่อการขยายระบบ ดังรูปที่ 3.20

- SERVER ทำหน้าที่ในการเก็บรวบรวมข้อมูลของจุดต่าง ๆ และกระจายข้อมูลไปยัง Viewer
- VIEWER (Client) ทำหน้าที่ติดต่อกับ Server เพื่อเข้าถึงข้อมูลสำหรับการ Viewing และ Control actions



รูปที่ 3.20 แสดงสถาปัตยกรรมของระบบโดยใช้ซอฟต์แวร์ CIMPLICITY HMI

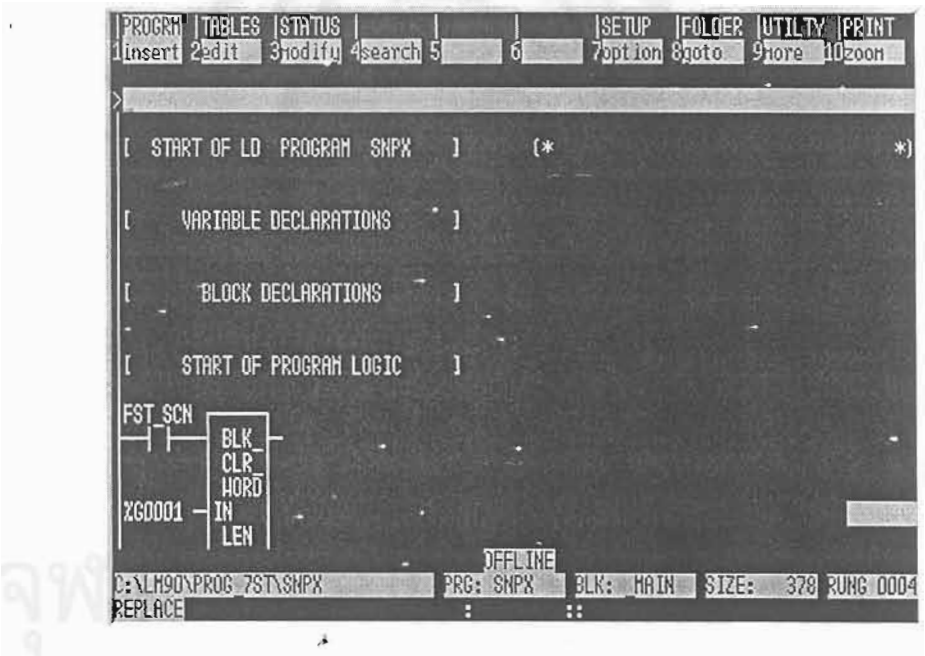
บทที่ 4
การเขียนโปรแกรมการทำงานของ 7 สถานี

4.1 การเขียนโปรแกรมด้วยภาษาแลดเดอร์

ในการเขียนโปรแกรมการทำงานจะใช้ซอฟต์แวร์ Logicmaster ซึ่งมีตัวอย่างหน้าจอการทำงานในส่วนของการเขียนโปรแกรมเป็นดังรูปที่ 4.1 มีขั้นตอนหรือวิธีโดยสังเขปดังนี้

1. เลือกการเขียนโปรแกรมสำหรับรุ่นของ PLC ที่ต้องการใช้
2. เข้าไปในส่วนของโครงแบบ (Configuration) เพื่อตั้งค่ารายละเอียดของ PLC ที่ใช้
3. เข้าไปในส่วนของการโปรแกรม (Program) เพื่อเริ่มต้นการเขียนโปรแกรม
4. ก่อนการเขียนโปรแกรม อาจจะกำหนดชื่อให้กับข้อมูลในหน่วยความจำเพื่อช่วยให้การอ้างถึงทำได้ง่าย
5. เขียนโปรแกรมการทำงาน
6. ใส่โปรแกรมการทำงานลงในตัว PLC เพื่อให้ทำงานตามโปรแกรมที่เขียนขึ้น

ตัวอย่างการเขียนโปรแกรมการทำงานแสดงอยู่ในภาคผนวก ก



รูปที่ 4.1 แสดงตัวอย่างหน้าจอของการเขียนโปรแกรมด้วยซอฟต์แวร์ Logicmaster

I19282795

4.2 โปรแกรมการทำงานของสถานีจ่ายชิ้นงาน

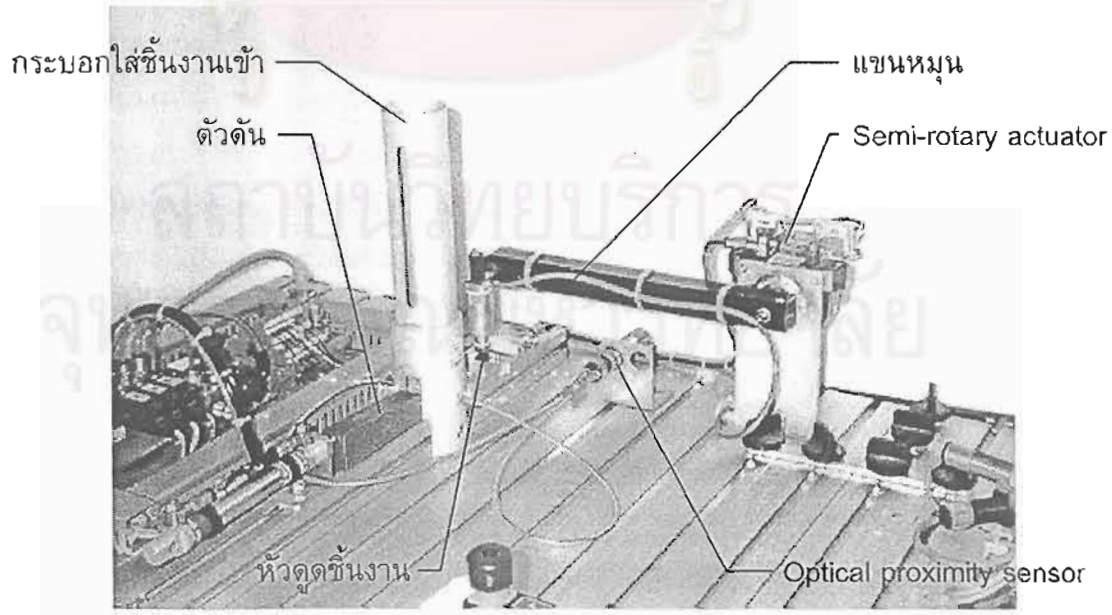
ในการเขียนโปรแกรมเพื่อให้สถานีสามารถทำงานได้มีขั้นตอนโดยสังเขปดังนี้

1. ศึกษาลำดับการทำงาน
2. เขียนโปรแกรมการทำงาน
3. ทดสอบการทำงานจริง

โปรแกรมการทำงานของสถานีจ่ายชิ้นงานแสดงอยู่ในภาคผนวก ซึ่งแบ่งออกเป็นส่วนย่อย ๆ ดังนี้

- *Mode of operation* เป็นส่วนของการรีเซตสถานี พร้อมทั้งสร้างแฟล็ก (Flag) สำหรับการทำงานในแบบขั้นเดียว (Single step mode) และแบบอัตโนมัติ (Automatic mode)
- *Auxiliary flag* เป็นส่วนของการสร้างแฟล็ก (Flag) สำหรับช่วยในการเขียนโปรแกรมการทำงาน
- *Step sequence module* เป็นส่วนของการสร้างลำดับขั้นในการทำงาน
- *Output assignment module* เป็นส่วนของการให้อุปกรณ์ขับทำงานตามลำดับขั้นที่ได้สร้างไว้
- *Flashing pulse module* เป็นส่วนของการสร้างพัลส์ (Pulse) เพื่อใช้งานในโปรแกรม
- *Communication module* เป็นส่วนของการจัดการเกี่ยวกับการสื่อสารเมื่อต้องมีการรับ-ส่งข้อมูลกับสถานีอื่น

ส่วนประกอบต่าง ๆ ของสถานีจ่ายชิ้นงานเป็นดังรูปที่ 4.2 และมีลักษณะการทำงานจากโปรแกรมที่ได้เขียนขึ้นมาเป็นดังนี้



รูปที่ 4.2 แสดงสถานีจ่ายชิ้นงาน

ก่อนที่จะเริ่มทำงานจะต้องทำการรีเซตสถานีโดยหยิบชิ้นงานที่ค้างอยู่ออกให้หมดแล้วกดปุ่ม RESET ที่กล่องควบคุมเพื่อให้อุปกรณ์ต่าง ๆ กลับมาอยู่ในสภาวะเริ่มต้น ก็คือ ตัวดันหด และแขนหมุนอยู่ด้านกระบอกใส่ชิ้นงาน หลังจากนั้นจะเห็นว่าไฟปุ่ม MAGAZINE EMPTY กระพริบในกรณีที่ไม่มีชิ้นงานอยู่ในกระบอกบรรจุชิ้นงาน ให้นำชิ้นงานวัตถุดิบใส่ลงไป หรือถ้ามีชิ้นงานอยู่แล้วก็อาจจะใส่เพิ่มให้เต็มหรือไม่ใส่ก็ได้ แล้วจึงกดปุ่ม MAGAZINE EMPTY ในตอนนี้ถ้ามีชิ้นงานอยู่ในกระบอก ซึ่งถูกตรวจจับด้วยเซ็นเซอร์แบบใช้แสง (Optical proximity sensor) ไฟที่กระพริบอยู่ก็จะดับลง ไฟปุ่ม START ก็จะสว่างขึ้นแสดงว่าสถานีพร้อมทำงานแล้ว และจะเริ่มทำงานเมื่อกดปุ่ม START

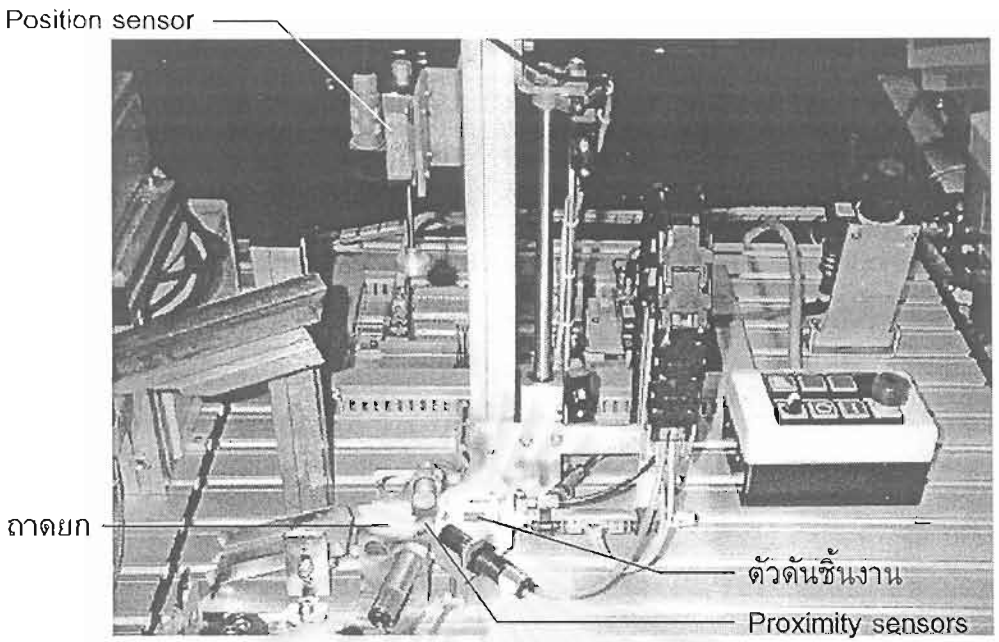
การทำงานเริ่มจากตรวจสอบข้อมูลที่ได้รับจากสถานีถัดไปว่าพร้อมรับชิ้นงานหรือไม่ เพื่อป้องกันไม่ให้แขนหมุน หมุนไปชนกับอุปกรณ์ของสถานีถัดไป ถ้าพร้อมรับชิ้นงาน แขนหมุนก็จะหมุนไปที่ถาดรับชิ้นงานของสถานีถัดไป จากนั้นตัวดันก็จะดันชิ้นงานออกมา แขนหมุนก็จะหมุนกลับมาที่กระบอกบรรจุชิ้นงาน ระบบสุญญากาศจะทำงานให้หัวดูด ดูดชิ้นงานเอาไว้ แล้วจึงหมุนแขนพร้อมชิ้นงานไปวางที่ถาดรับชิ้นงานของสถานีถัดไป เมื่อวางชิ้นงานเรียบร้อยแล้ว ระบบสุญญากาศจะปิดเพื่อปล่อยชิ้นงานไว้ จากนั้นจึงหมุนแขนกลับมาที่กระบอกบรรจุชิ้นงาน เพื่อให้สถานีถัดไปนำชิ้นงานไปดำเนินการต่อไป

4.3 โปรแกรมการทำงานของสถานีทดสอบชิ้นงาน

การเขียนโปรแกรมการทำงานจะมีวิธีการเช่นเดียวกับสถานีจ่ายชิ้นงาน ซึ่งส่วนประกอบต่าง ๆ ของสถานีทดสอบชิ้นงานเป็นดังรูปที่ 4.3 และมีลักษณะการทำงานจากโปรแกรมที่ได้เขียนขึ้นมาเป็นดังนี้

การรีเซตสถานีให้นำชิ้นงานที่ค้างอยู่ออกแล้วจึงกดปุ่ม RESET อุปกรณ์บนสถานีจะกลับสู่สภาวะเริ่มต้น ก็คือ ตัวดันหดกลับ และถาดยกอยู่ในตำแหน่งล่าง ไฟปุ่ม START จะสว่างขึ้นแสดงว่าสถานีพร้อมทำงานและจะทำงานเมื่อกดปุ่ม START

การทำงานเริ่มจากเมื่อได้รับชิ้นงานต่อจากสถานีก่อนหน้าแล้ว จะทำการตรวจสอบชิ้นงาน โดยใช้เซนเซอร์ทั้ง 3 ตัว คือ Inductive, Capacitive และ Optical proximity sensor ดังตารางที่ 2.6 เมื่อตรวจสอบเสร็จแล้วจะบันทึกข้อมูลชิ้นงานในบิต D1 และ D2 จากนั้นถาดยกถูกยกขึ้นไปตำแหน่งบนเพื่อทดสอบความสูงของชิ้นงาน โดยเมื่อถาดยกถูกยกขึ้นจนสุดแล้ว จะอ่านค่าความสูงจากแอนะลอกอินพุต (Analog Input) ถ้าความสูงของชิ้นงานอยู่ในช่วงความสูงที่ใช้ได้ บิต D0 จะถูกบันทึกเป็น 1 และถ้าความสูงของชิ้นงานสูงมากเกินไปหรือต่ำเกินไป บิต D0 จะถูกบันทึกเป็น 0 เมื่อตรวจสอบความสูงเสร็จแล้ว ชิ้นงานก็พร้อมที่จะถูกส่งไปสถานีถัดไป ถ้าสถานีถัดไปส่งข้อมูลว่าพร้อมรับชิ้นงาน ตัวดันจะดันชิ้นงานให้ไหลไปสู่สถานีถัดไปแล้วจะหดกลับ จากนั้นถาดยกจะเลื่อนลงไปตำแหน่งต่ำสุด และพร้อมรับชิ้นงานชิ้นต่อไป



รูปที่ 4.3 แสดงสถานีทดสอบชิ้นงาน

ตารางที่ 4.1 แสดงการตรวจสอบชิ้นงาน โดยเซนเซอร์ทั้ง 3 ตัว

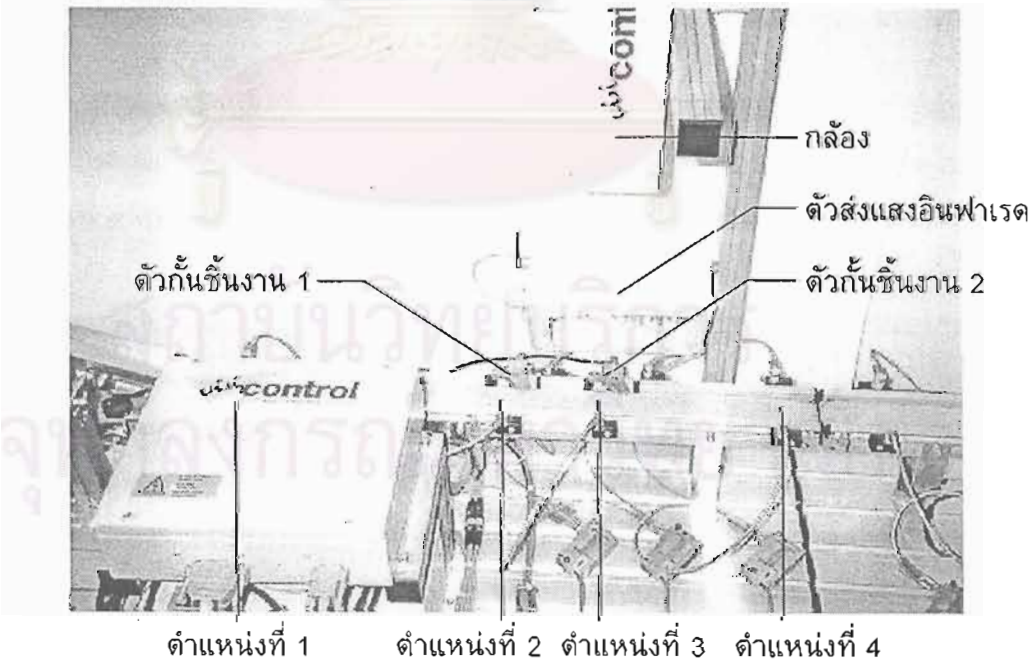
Capacitive	Inductive	Optical	ผลการตรวจสอบ
0	-	-	ไม่มีชิ้นงานอยู่บนถาดยก
1	1	0	ชิ้นงานเสีย
1	1	1	ชิ้นงานโลหะ
1	0	1	ชิ้นงานสีแดง
1	0	0	ชิ้นงานสีดำ

4.4 โปรแกรมการทำงานของสถานีควบคุมคุณภาพ

การเขียนโปรแกรมการทำงานจะมีวิธีการเช่นเดียวกับสถานีจ่ายชิ้นงาน ซึ่งส่วนประกอบต่าง ๆ ของสถานีควบคุมคุณภาพเป็นดังรูปที่ 4.4 และมีลักษณะการทำงานจากโปรแกรมที่ได้เขียนขึ้นมาเป็นดังนี้

การรีเซตสถานีให้นำชิ้นงานที่ค้างอยู่ออก แล้วจึงกดปุ่ม RESET ในขณะที่สถานีทำการรีเซต สายพานลำเลียงจะหมุน ถ้ายังมีชิ้นงานค้างอยู่ ชิ้นงานจะถูกปล่อยให้ไหลออกจากสายพานลำเลียง การรีเซตนี้จะใช้เวลา 12 วินาที ถ้าไม่พบชิ้นงานค้างอยู่บนสายพาน ไปปุ่ม START จะสว่างขึ้น และสถานีพร้อมทำงาน

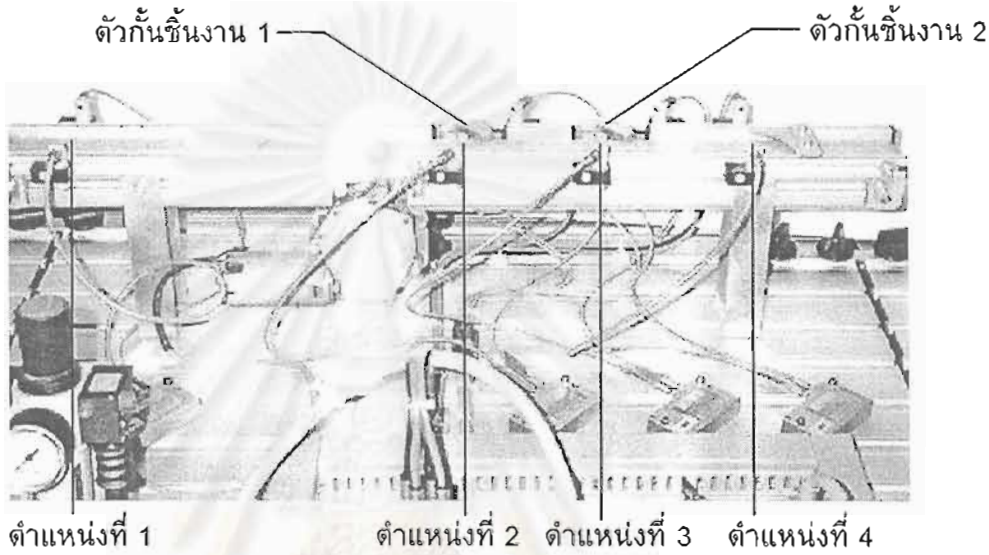
เมื่อกดปุ่ม START ให้เริ่มทำงานสายพานลำเลียงจะหมุนและถ้าได้รับชิ้นงานจากสถานีก่อนหน้าแล้ว ชิ้นงานจะผ่านตำแหน่งที่ 1 ซึ่งตำแหน่งต่าง ๆ จะถูกตรวจจับได้โดยเซนเซอร์แบบใช้แสง (Optical proximity sensor) ขณะนี้เองข้อมูลของชิ้นงานที่ได้รับจะถูกเก็บลงในหน่วยความจำชั่วคราวสำหรับตำแหน่งที่ 1 และเมื่อชิ้นงานไหลผ่านมาจนถึงตำแหน่งที่ 2 ซึ่งชิ้นงานจะถูกกั้นไว้โดยตัวกั้นชิ้นงาน 1 ให้ค้างไว้ที่ตำแหน่งที่ 2 ข้อมูลของชิ้นงานจะถูกคัดลอกจากหน่วยความจำชั่วคราวสำหรับตำแหน่งที่ 1 มาที่หน่วยความจำชั่วคราวสำหรับตำแหน่งที่ 2 จากนั้นถ้าไม่มีชิ้นงานค้างอยู่ที่ตำแหน่งที่ 3 ตัวกั้นชิ้นงาน 1 จะปล่อยให้ชิ้นงานไหลผ่านไปจนเมื่อชิ้นงานไหลมาถึงตำแหน่งที่ 3 ซึ่งชิ้นงานจะถูกกั้นไว้โดยตัวกั้นชิ้นงาน 2 ให้ค้างไว้ที่ตำแหน่งที่ 3 ข้อมูลของชิ้นงานจะถูกคัดลอกจากหน่วยความจำชั่วคราวสำหรับตำแหน่งที่ 2 มาที่หน่วยความจำชั่วคราวสำหรับตำแหน่งที่ 3 ณ.ตำแหน่งที่ 3 นี้ชิ้นงานจะถูกตรวจสอบคุณภาพโดยตัวส่งแสงอินฟาเรด จะเปิดแสงอินฟ่าเรดและกล้องจะบันทึกภาพของชิ้นงาน เพื่อวัดขนาดเส้นผ่าศูนย์กลางวงนอก และความหนาของร่องวงแหวนซึ่งภาพและผลการตรวจสอบจะแสดงที่จอมอนิเตอร์ และเมื่อตรวจสอบขนาดและความหนาเสร็จแล้วข้อมูลจะถูกบันทึกลงในบิตข้อมูล D3 และ D4 จากนั้นข้อมูลของชิ้นงานจะถูกคัดลอกจากหน่วยความจำชั่วคราวสำหรับตำแหน่งที่ 3 มาที่หน่วยความจำสำหรับข้อมูลขาออก เพื่อส่งให้สถานีถัดไป ซึ่งถ้าสถานีถัดไปพร้อมรับชิ้นงาน ตัวกั้นที่ 2 จะปล่อยให้ชิ้นงานไหลผ่านไปสู่สถานีถัดไปได้ ในกรณีของชิ้นงานจากสถานีก่อนหน้าจะไหลเข้าสู่สถานีได้ก็ต่อเมื่อไม่มีชิ้นงานค้างอยู่ในตำแหน่งที่ 2 และจะมีชิ้นงานค้างอยู่ในสถานีนี้ได้ทั้งหมดไม่เกิน 2 ชิ้น



รูปที่ 4.4 แสดงสถานีควบคุมคุณภาพ

4.5 โปรแกรมการทำงานของสถานีพักชิ้นงาน

การเขียนโปรแกรมการทำงานจะมีวิธีการเช่นเดียวกับสถานีจ่ายชิ้นงาน ซึ่งส่วนประกอบต่าง ๆ ของสถานีทดสอบพักชิ้นงานเป็นดังรูปที่ 4.5 และมีลักษณะการทำงานจากโปรแกรมที่ได้เขียนขึ้นมาเป็นดังนี้



รูปที่ 4.5 แสดงสถานีพักชิ้นงาน

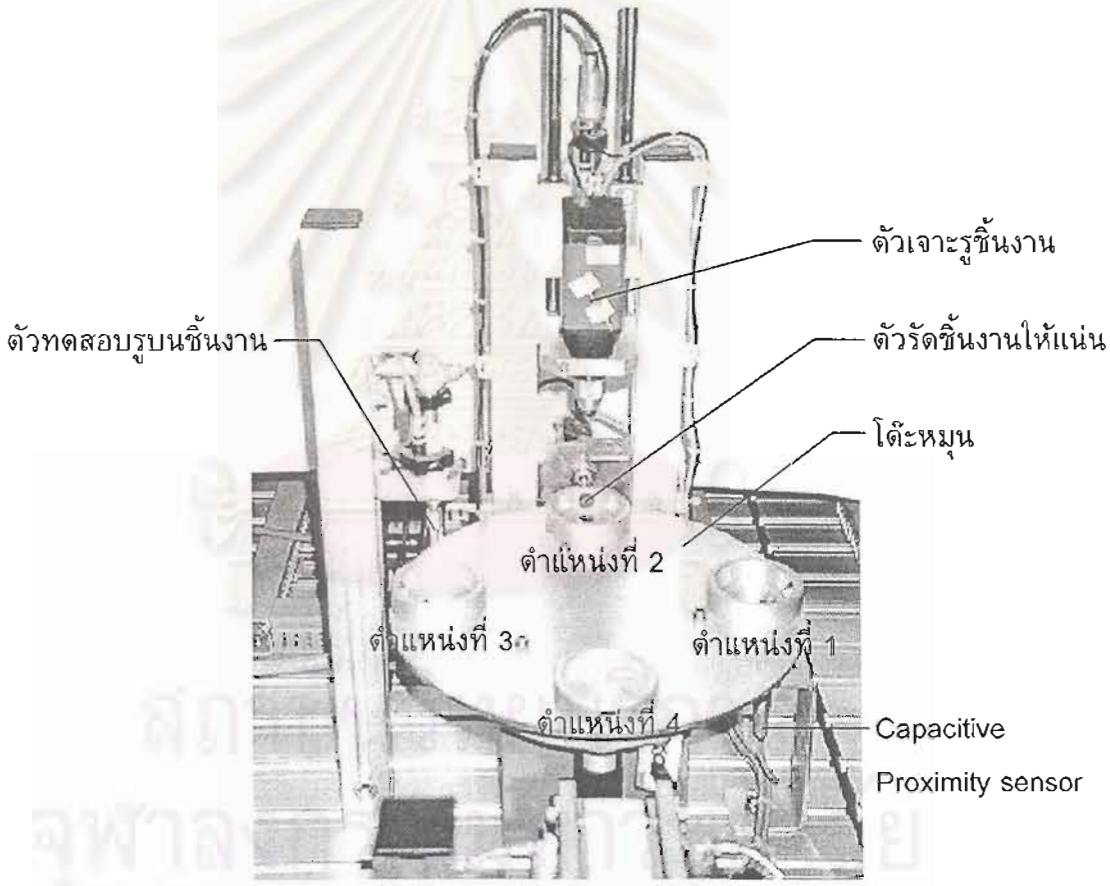
การรีเซตสถานีให้นำชิ้นงานที่ค้างอยู่ออกแล้วจึงกดปุ่ม RESET ขณะที่สถานีทำการรีเซต สายพานลำเลียงจะหมุน ถ้ายังมีชิ้นงานค้างอยู่ชิ้นงานจะถูกปล่อยให้ไหลไปจนสุดถึงตำแหน่งที่ 4 ให้นำชิ้นงานที่ค้างอยู่ออกให้หมด ซึ่งการรีเซตจะใช้เวลา 12 วินาที ถ้าไม่พบชิ้นงานค้างอยู่บนสายพาน ไฟปุ่ม START จะสว่างขึ้น และสถานีพร้อมทำงาน

การทำงานเริ่มจาก เมื่อกดปุ่ม START สายพานลำเลียงจะหมุนชิ้นงานจากสถานีก่อนหน้าจะไหลผ่านตำแหน่งที่ 1 ซึ่งตำแหน่งต่าง ๆ จะถูกตรวจจับได้โดยเซนเซอร์แบบใช้แสง (Optical proximity sensor) ขณะที่ชิ้นงานผ่านตำแหน่งที่ 1 ข้อมูลของชิ้นงานที่ได้รับจะถูกเก็บลงในหน่วยความจำชั่วคราวสำหรับตำแหน่งที่ 1 และเมื่อชิ้นงานไหลผ่านมาจนถึงตำแหน่งที่ 2 ซึ่งชิ้นงานจะถูกกั้นไว้โดยตัวกั้นที่ 1 ให้ค้างไว้ที่ตำแหน่งที่ 2 ข้อมูลของชิ้นงานจะถูกคัดลอกจากหน่วยความจำชั่วคราวสำหรับตำแหน่งที่ 1 มาที่หน่วยความจำชั่วคราวสำหรับตำแหน่งที่ 2 จากนั้นถ้าไม่มีชิ้นงานค้างอยู่ที่ตำแหน่งที่ 3 ตัวกั้นชิ้นงาน 1 จะปล่อยให้ชิ้นงานไหลผ่านไปได้ และเมื่อชิ้นงานไหลไปจนถึงตำแหน่งที่ 3 ซึ่งชิ้นงานจะถูกกั้นไว้โดยตัวกั้นชิ้นงาน 2 ให้ค้างไว้ที่ตำแหน่งที่ 3 ข้อมูลของชิ้นงานจะถูกคัดลอกจากหน่วยความจำชั่วคราวสำหรับตำแหน่งที่ 2 มาที่หน่วยความจำชั่วคราวสำหรับตำแหน่งที่ 3 จากนั้นถ้าไม่มีชิ้นงานค้างอยู่ที่ตำแหน่งที่ 4 ตัวกั้น

ชิ้นงานที่ 2 จะปล่อยให้ชิ้นงานไหลผ่านไปได้และเมื่อชิ้นงานไหลไปจนถึงตำแหน่งที่ 4 ข้อมูลของชิ้นงานจะถูกคัดลอกจากหน่วยความจำชั่วคราวสำหรับตำแหน่งที่ 3 มาที่หน่วยความจำสำหรับข้อมูลขาออกเพื่อส่งให้สถานีถัดไป ในกรณีของชิ้นงานจากสถานีก่อนหน้า จะไหลเข้าสู่สถานีได้ก็ต่อเมื่อไม่มีชิ้นงานค้างอยู่ในตำแหน่งที่ 2 และจะมีชิ้นงานค้างอยู่ในสถานีนี้ได้ทั้งหมดไม่เกิน 3 ชิ้น

4.6 โปรแกรมการทำงานของสถานีผลิตชิ้นงาน

การเขียนโปรแกรมการทำงานจะมีวิธีการเช่นเดียวกับสถานีจ่ายชิ้นงาน ซึ่งส่วนประกอบต่าง ๆ ของสถานีผลิตชิ้นงานเป็นดังรูปที่ 4.6 และมีลักษณะการทำงานจากโปรแกรมที่ได้เขียนขึ้นมาเป็นดังนี้



รูปที่ 4.6 แสดงสถานีผลิตชิ้นงาน

การรีเซตสถานีให้นำชิ้นงานที่ค้างอยู่ออกแล้วจึงกดปุ่ม RESET โต๊ะหมุนจะหมุนไปจนตรงตำแหน่ง ซึ่งจะถูกตรวจจับโดย Inductive proximity sensor จากนั้นโต๊ะหมุนจะหมุนไปจนตรงตำแหน่งอีกครั้งซึ่งเท่ากับ 90 องศา ที่ตำแหน่งที่ 1 จะใช้ Capacitive proximity sensor ตรวจจับชิ้นงานว่ามีอยู่หรือไม่ ถ้ามีชิ้นงานอยู่ที่ตำแหน่งที่ 1 จะถูกบันทึกไว้ ขั้นตอนการตรวจ

สอบนี้ใช้เวลา 1 วินาที จากนั้นโต๊ะหมุนจะหมุนไปจนตรงตำแหน่งอีก และทำเช่นเดิม คือ ตรวจสอบชิ้นงานที่ตำแหน่งที่ 1 แล้วบันทึกผลไว้ โต๊ะหมุนจะหมุนไปจนตรงตำแหน่งอีกจนครบ 4 ครั้ง ก็คือตรวจสอบชิ้นงานจนครบ 4 ตำแหน่ง ถ้ายังคงมีชิ้นงานค้างอยู่อีกไฟปุ่ม START จะไม่สว่างขึ้น ให้นำชิ้นงานที่ค้างอยู่ออกแล้วทำการรีเซตใหม่ แต่ถ้าไม่มีชิ้นงานค้างอยู่เลยไฟปุ่ม START จะสว่างขึ้นและสถานีพร้อมทำงาน

การทำงานเริ่มจากเมื่อได้รับข้อมูลจากสถานีก่อนหน้า ข้อมูลที่ได้รับจะถูกเก็บลงในหน่วยความจำชั่วคราวสำหรับตำแหน่งที่ 1 จากนั้นจะใส่ค่า 1 ลงในบิตข้อมูล O_HS ในข้อมูลขาออกเพื่อสั่งให้สถานีแขนกลนำชิ้นงานจากสถานีก่อนหน้ามาใส่ไว้ที่ตำแหน่งที่ 1 เมื่อชิ้นงานมาถึงตำแหน่งที่ 1 แล้ว บิตข้อมูล O_HS จะถูกใส่ค่า 0 และถ้าตำแหน่งที่ 4 ไม่มีชิ้นงานค้างอยู่ โต๊ะหมุนจะหมุนไปจนตรงตำแหน่งอีกครั้งระหว่างที่โต๊ะหมุนหมุน ข้อมูลของชิ้นงานจะถูกคัดลอกจากหน่วยความจำชั่วคราวสำหรับตำแหน่งที่ 3 มาที่หน่วยความจำสำหรับข้อมูลขาออกเพื่อส่งให้สถานีถัดไปจากนั้นจะคัดลอกข้อมูลชิ้นงานจากหน่วยความจำชั่วคราวสำหรับตำแหน่งที่ 2 มาที่หน่วยความจำชั่วคราวสำหรับตำแหน่งที่ 3 และคัดลอกข้อมูลชิ้นงานจากหน่วยความจำชั่วคราวสำหรับตำแหน่งที่ 1 มาที่หน่วยความจำชั่วคราวสำหรับตำแหน่งที่ 2 ในกรณีถ้ามีชิ้นงานอยู่ในตำแหน่งที่ 2 และตำแหน่งที่ 3 และบิตข้อมูล D0 ของชิ้นงานเป็น 1 นั่นคือความสูงของชิ้นงานใช้ได้ ที่ตำแหน่งที่ 2 ตัวรัดชิ้นงานจะยืดออกเพื่อรัดชิ้นงานให้แน่น สว่านของตัวเจาะรูชิ้นงานจะหมุนและเลื่อนลงมาเจาะชิ้นงานจนสุดจึงเลื่อนขึ้นไป สว่านจะหยุดหมุนและตัวรัดชิ้นงานหดกลับ จากนั้นที่ตำแหน่งที่ 3 ตัวทดสอบรูของชิ้นงานจะยืดออกเพื่อทดสอบรู และจะบันทึกค่าที่ตรวจสอบได้ลงในบิตข้อมูล D5 จากนั้น ตัวทดสอบรูจะหดกลับเข้าที่ ชิ้นงานที่ตำแหน่งที่ 4 เป็นชิ้นงานที่เสร็จเรียบร้อยแล้วพร้อมส่งต่อให้สถานีถัดไปและถ้าสถานีถัดไปนำชิ้นงานที่ตำแหน่งที่ 4 ไปแล้วสถานีก็จะพร้อมรับชิ้นงานถัดไป

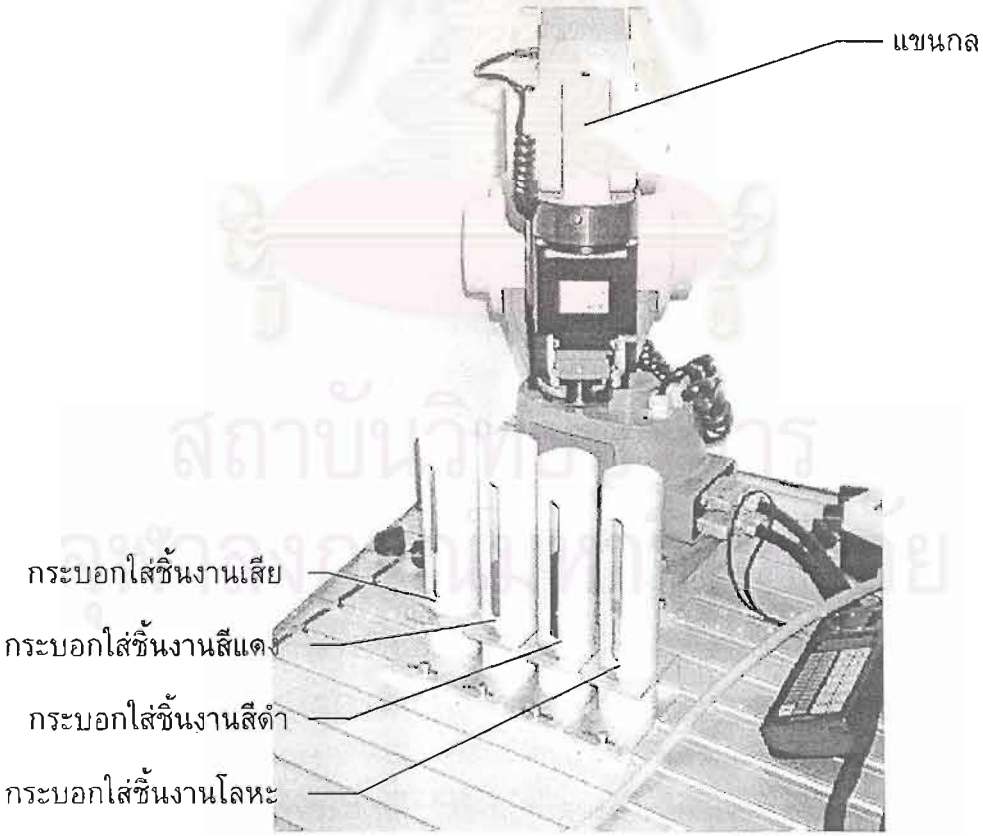
4.7 โปรแกรมการทำงานของสถานีแขนกลเพื่อเคลื่อนย้ายหรือเก็บชิ้นงาน

การเขียนโปรแกรมการทำงานจะมีวิธีการเช่นเดียวกับสถานีจ่ายชิ้นงาน ซึ่งส่วนประกอบต่าง ๆ ของสถานีแขนกลเป็นดังรูปที่ 4.7 และมีลักษณะการทำงานจากโปรแกรมที่ได้เขียนขึ้นมาเป็นดังนี้

การรีเซตสถานี ให้กดปุ่ม RESET ซึ่งจะเป็นการสั่งให้แขนกลเคลื่อนที่ไปที่จุดออริจิน (Origin) เพื่อเป็นการรีเซตค่า 0 ให้กับข้อต่อต่าง ๆ จากนั้นแขนกลจะเคลื่อนที่ไปที่จุดเริ่มต้น ไฟปุ่ม START จะสว่างขึ้น และสถานีพร้อมทำงาน

การทำงานจะแบ่งออกเป็นสองกรณี ในกรณีที่หนึ่งคือ ได้รับการร้องขอให้เคลื่อนย้ายชิ้นงาน ผ่านทางบิตข้อมูล I_HS ที่ได้รับจากสถานีก่อนหน้า เมื่อบิต I_HS เป็น 1 แขนกล จะเคลื่อนที่ไปหยิบชิ้นงานจากสถานีพักชิ้นงานที่ตำแหน่งที่ 4 ไปยังสถานีผลิตชิ้นงาน ที่ตำแหน่งที่ 1 จากนั้นจะเคลื่อนที่กลับไปจุดเริ่มต้นพร้อมทำงานต่อไป ส่วนในกรณีที่สองคือ เมื่อสถานีก่อนหน้าพร้อมส่งชิ้นงานต่อมาให้ แขนกลจะทำการหยิบชิ้นงานจากสถานีผลิตชิ้นงานที่ตำแหน่งที่ 4

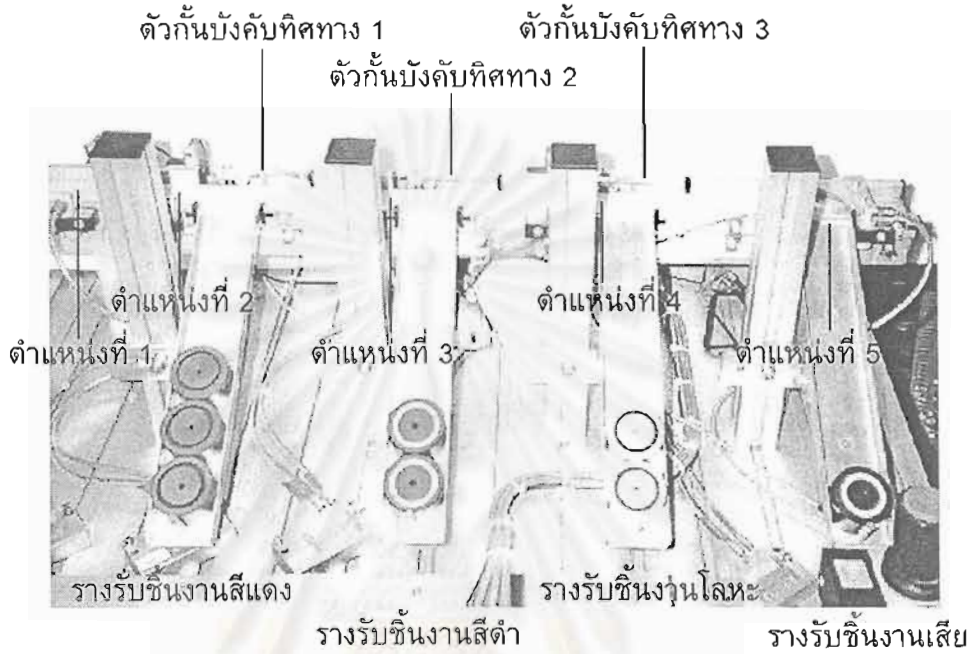
จากนั้นจะเคลื่อนที่ไปจุดเริ่มต้นถ้าข้อมูลของชิ้นงานที่ได้รับ เป็นชิ้นงานที่มีร่องวงแหวน ซึ่งบิต D3 เป็น 1 หรือ บิต D4 เป็น 1 หรือบิต D3 และ D4 เป็น 1 ทั้งคู่ แขนกล จะนำชิ้นงานส่งต่อให้สถานีแยกชิ้นงานโดยนำชิ้นงานไปวางที่ตำแหน่งที่ 1 ของสถานีแยกชิ้นงาน และถ้าเป็นชิ้นงานที่ไม่มีร่องวงแหวน ซึ่งบิต D3 และ D4 จะเป็น 0 และเป็นชิ้นงานที่มีความสูงและเจารูใช้ได้ ชิ้นงานจะถูกแยกไปเก็บในกระบอกใส่ชิ้นงานตามชนิดและสีของชิ้นงาน โดยถ้าเป็นชิ้นงานสีแดง แขนกลจะไปปล่อยชิ้นงานลงที่กระบอกใส่ชิ้นงานสีแดง และถ้าเป็นชิ้นงานสีดำ แขนกลจะไปปล่อยชิ้นงานลงที่กระบอกใส่ชิ้นงานสีดำ และถ้าเป็นชิ้นงานโลหะ แขนกลจะไปปล่อยชิ้นงานลงที่กระบอกใส่ชิ้นงานโลหะ ถ้าเป็นชิ้นงานที่ความสูงหรือรูเจาะใช้ไม่ได้ แขนกลจะไปปล่อยชิ้นงานลงที่กระบอกใส่ชิ้นงานเสีย ในขณะที่แขนกลนำชิ้นงานไปใส่ลงในกระบอกใส่ชิ้นงานใด ๆ จะมีตัวนับซึ่งจะนับจำนวนชิ้นงานที่มีอยู่ในกระบอกใส่ชิ้นงานนั้น ๆ และถ้ามีชิ้นงานอยู่ในกระบอกใส่ชิ้นงาน 6 ชิ้น ไฟปุ่ม MAGAZINE FULL จะกระพริบแสดงว่าชิ้นงานเต็ม ให้นำชิ้นงานออกจากกระบอกใส่ชิ้นงานที่เต็ม เสร็จแล้วกดปุ่ม MAGAZINE FULL ค่าตัวนับของกระบอกใส่ชิ้นงานนั้นจะถูกรีเซตให้เป็น 0 เมื่อนำชิ้นงานไปวางที่สถานีถัดไป หรือนำชิ้นงานไปแยกใส่กระบอกรับชิ้นงานเสร็จแล้ว แขนกลจะเคลื่อนที่กลับไปจุดเริ่มต้น และสถานีพร้อมรับชิ้นงานถัดไป



รูปที่ 4.7 แสดงสถานีแขนกล

4.8 โปรแกรมการทำงานของสถานีแยกชิ้นงาน

การเขียนโปรแกรมการทำงานจะมีวิธีการเช่นเดียวกับสถานีจ่ายชิ้นงาน ซึ่งส่วนประกอบต่าง ๆ ของสถานีแยกชิ้นงานเป็นดังรูปที่ 4.8 และมีลักษณะการทำงานจากโปรแกรมที่ได้เขียนขึ้นมาเป็นดังนี้



รูปที่ 4.8 แสดงสถานีแยกชิ้นงาน

การรีเซตสถานีให้นำชิ้นงานที่ค้างอยู่บนสายพานออก แล้วจึงกดปุ่ม RESET ขณะที่สถานีทำการรีเซตสายพานลำเลียงจะหมุน ถ้ามีชิ้นงานค้างอยู่ชิ้นงานจะถูกปล่อยให้ไหลไปจนสุด ชิ้นงานจะไหลไปลงที่รางรับชิ้นงานเสีย การรีเซตจะใช้เวลา 12 วินาที ถ้าไม่มีชิ้นงานค้างอยู่บนสายพานลำเลียงแล้ว ไฟปุ่ม START จะสว่างขึ้น และสถานีพร้อมทำงาน

การทำงานเริ่มจากเมื่อได้รับชิ้นงานจากสถานีก่อนหน้าและชิ้นงานมาถึงตำแหน่งที่ 1 ซึ่งตำแหน่งต่าง ๆ จะถูกตรวจจับได้โดยเซนเซอร์แบบใช้แสง (Optical proximity sensor) มอเตอร์สายพานลำเลียงจะหมุน และข้อมูลของชิ้นงานที่ได้รับจะถูกเก็บลงในหน่วยความจำชั่วคราว ในกรณีที่ใช้แยกชิ้นงานที่มีร่องวงแหวน ถ้าชิ้นงานที่ได้รับมีขนาดของเส้นผ่าศูนย์วงนอก, ความหนาของร่องวงแหวน, ความสูง และรูเจาะใช้ได้ชิ้นงานจะแยกลงตามชนิดและสีของชิ้นงาน โดยถ้าเป็นชิ้นงานสีแดงตัวบังคับทิศทาง 1 จะเปิดกั้นบังคับให้ชิ้นงานที่ไหลมาลงในรางรับชิ้นงานสีแดง และถ้าเป็นชิ้นงานสีดำตัวบังคับทิศทาง 2 จะเปิดกั้นบังคับให้ชิ้นงานที่ไหลมาลงในรางรับชิ้นงานสีดำ และถ้าเป็นชิ้นงานโลหะตัวบังคับทิศทาง 3 จะเปิดกั้นบังคับให้ชิ้นงานที่ไหลมาลงในรางรับชิ้นงานโลหะ ถ้าชิ้นงานที่ได้รับมีขนาดของเส้นผ่าศูนย์กลางวงนอก และความหนาของร่องวงแหวนใช้ได้อย่างใดอย่างหนึ่ง หรือชิ้นงานสูงหรือต่ำเกินไป หรือชิ้นงานที่

รูเจาะใช้ไม่ได้ชิ้นงานจะถูกปล่อยให้ไหลไปจนสุดและไปลงที่รางรับชิ้นงานเสีย ในขณะที่ชิ้นงานไหลไปลงที่รางรับชิ้นงานใดจะมีตัวนับจำนวนชิ้นงานที่ไหลลงรางนั้น ๆ และถ้ามีชิ้นงานอยู่ในรางครบ 5 ชิ้น ไฟปุ่ม MAGAZINE FULL จะกระพริบแสดงว่า ชิ้นงานเต็มให้นำชิ้นงานออกจากรางที่เต็ม เสร็จแล้วกดปุ่ม MAGAZINE FULL ค่าตัวนับในรางนั้นจะถูกรีเซตให้เป็น 0 และเมื่อชิ้นงานไหลจนไปลงที่รางรับชิ้นงานเรียบร้อยแล้ว ซึ่งตรวจสอบได้โดยเซนเซอร์ตำแหน่งที่ 2-5 ตัวบังคับทิศทางที่เปิดอยู่จะปิด มอเตอร์สายพานจะหยุดหมุน และสถานีพร้อมรับชิ้นงานถัดไป

บทที่ 5
การจัดเครือข่ายการควบคุมในระบบการผลิตอัตโนมัติแบบ 7 โมดูล

5.1 การจัดเครือข่ายการสื่อสาร เมื่อใช้ 7 โมดูล

ในการจัดเครือข่ายการสื่อสารของทั้ง 7 สถานี จะต้องพิจารณาถึงความสามารถในการสื่อสารของตัว PLC ที่ใช้ควบคุมในสถานีนั้นๆ ซึ่งทั้ง 7 สถานีสามารถใช้โปรโตคอลสื่อสาร ได้ดังนี้

สถานีจ่ายชิ้นงานและสถานีทดสอบชิ้นงาน สถานีทั้งสองนี้ใช้ตัว PLC ตัวเดียวกันในการควบคุม คือ PLC รุ่น 90-30 แต่การเขียนโปรแกรมแลดเดอร์ในการทำงานจะเขียนแยกกัน เป็นบล็อกย่อยๆ ในหนึ่งโปรแกรม ซึ่งตัว PLC รุ่น 90-30 นี้จะใช้เป็นตัว Master ในการรับ-ส่งข้อมูล สามารถสื่อสารโดยใช้โปรโตคอล Genius Bus, SNP/SNP-X และ Ethernet โดยจะมีโมดูล Genius Bus Controller สามารถควบคุม I/O Block ที่อยู่ใน สถานีพักชิ้นงาน และสถานีแยกชิ้นงาน และมีโมดูล Communication Coprocessor ซึ่งใช้เป็นตัว Master ในการสื่อสารโปรโตคอล SNP/SNP-X

สถานีควบคุมคุณภาพ สถานีนี้ใช้ตัวควบคุม คือ PLC รุ่น 90-70 สามารถสื่อสารโดยใช้โปรโตคอล Genius Bus, SNP/SNP-X (slave) และ Ethernet ซึ่งจะมีโมดูล Genius Bus Controller ใช้รับ-ส่งข้อมูลกับตัว PLC รุ่น 90-30

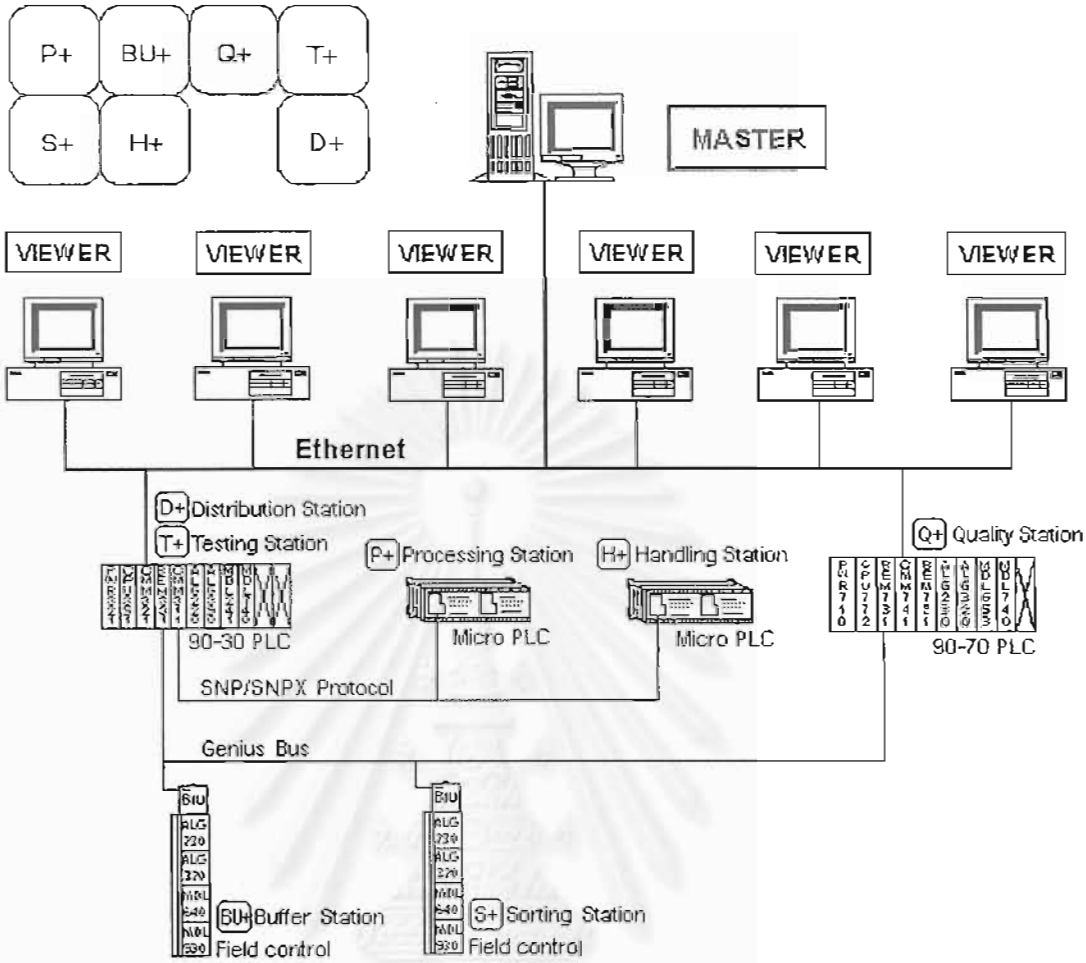
สถานีพักชิ้นงานและสถานีแยกชิ้นงาน สถานีทั้งสองนี้ใช้ตัว I/O Block คือ Field Control ในการควบคุมการทำงานผ่าน Genius Bus ซึ่งโปรแกรมแลดเดอร์ในการทำงาน จะเป็นบล็อกหนึ่งในโปรแกรมของตัว PLC รุ่น 90-30

สถานีผลิตชิ้นงานและสถานีแขนกล สถานีทั้งสองนี้ใช้ตัวควบคุม คือ PLC รุ่น Micro สามารถสื่อสารโดยใช้ โปรโตคอล SNP/SNP-X (slave) และมีการรับ-ส่งข้อมูลกับตัว PLC รุ่น 90-30 ผ่านทางสายสื่อสาร SNP/SNP-X

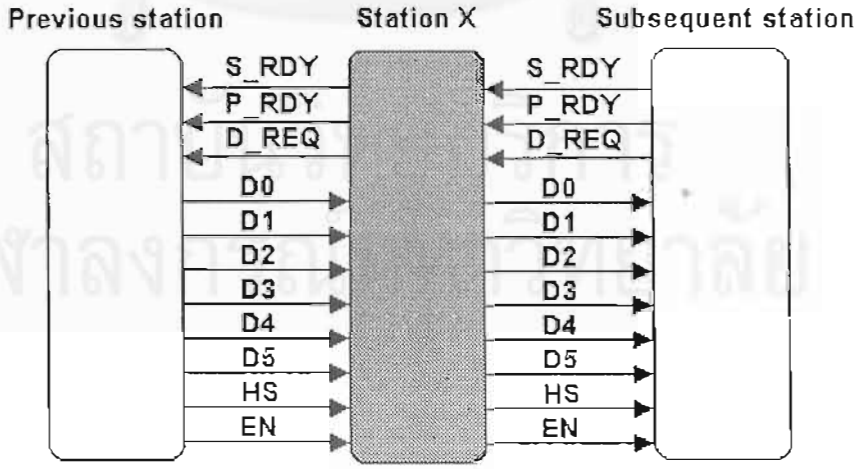
เครือข่ายการสื่อสารระดับล่างจะใช้โปรโตคอล Genius Bus ระดับกลาง จะใช้โปรโตคอล SNP/SNP-X ส่วนเครือข่ายระดับบนจะใช้ เครือข่าย Ethernet ดังรูปที่ 5.1

5.2 การรับ-ส่งข้อมูลระหว่างสถานี

ข้อมูลที่รับ-ส่งระหว่างสถานีเป็นข้อมูลขนาด 16 บิต แบ่งออกเป็น 2 ส่วน คือ 8 บิตแรก จะเป็นข้อมูลของสถานี ซึ่งจะถูกส่งให้สถานีก่อนหน้า และ 8 บิตหลัง จะเป็นข้อมูลของชิ้นงานซึ่งจะถูกส่งให้สถานีถัดไป ดังตารางที่ 5.1, 5.2 และรูปที่ 5.2



รูปที่ 5.1 แสดงเครือข่ายการสื่อสารแบบใช้ 7 โมดูล



รูปที่ 5.2 แสดงบิตข้อมูลที่รับ-ส่งระหว่างสถานี

ตารางที่ 5.1 แสดงข้อมูลอินพุตขนาด 16 บิต ของสถานีต่าง ๆ

BIT	FLAG		DATA SOURCE
0	I_S_RDY	Subsequent station ready	Subsequent station
1	I_P_RDY	Subsequent station ready for part	Subsequent station
2	I_D_REQ	Data request	Subsequent station
3	-		
4	-		
5	-		
6	-		
7	-		
8	I_D0	Data bit 0	Previous station
9	I_D1	Data bit 1	Previous station
10	I_D2	Data bit 2	Previous station
11	I_D3	Data bit 3	Previous station
12	I_D4	Data bit 4	Previous station
13	I_D5	Data bit 5	Previous station
14	I_HS	Handling slave request	Previous station
15	I_EN	Data/Part ready	Previous station

ตารางที่ 5.2 แสดงข้อมูลเอาต์พุตขนาด 16 บิต ของสถานีต่าง ๆ

BIT	FLAG		TARGET
0	O_S_RDY	Station ready	Previous station
1	O_P_RDY	Station ready for part	Previous station
2	O_D_REQ	Data request	Previous station
3	-		
4	-		
5	-		
6	-		
7	-		
8	O_D0	Data bit 0	Subsequent station
9	O_D1	Data bit 1	Subsequent station
10	O_D2	Data bit 2	Subsequent station
11	O_D3	Data bit 3	Subsequent station
12	O_D4	Data bit 4	Subsequent station
13	O_D5	Data bit 5	Subsequent station
14	O_HS	Handling slave request	Subsequent station
15	O_EN	Data/Part ready	Subsequent station

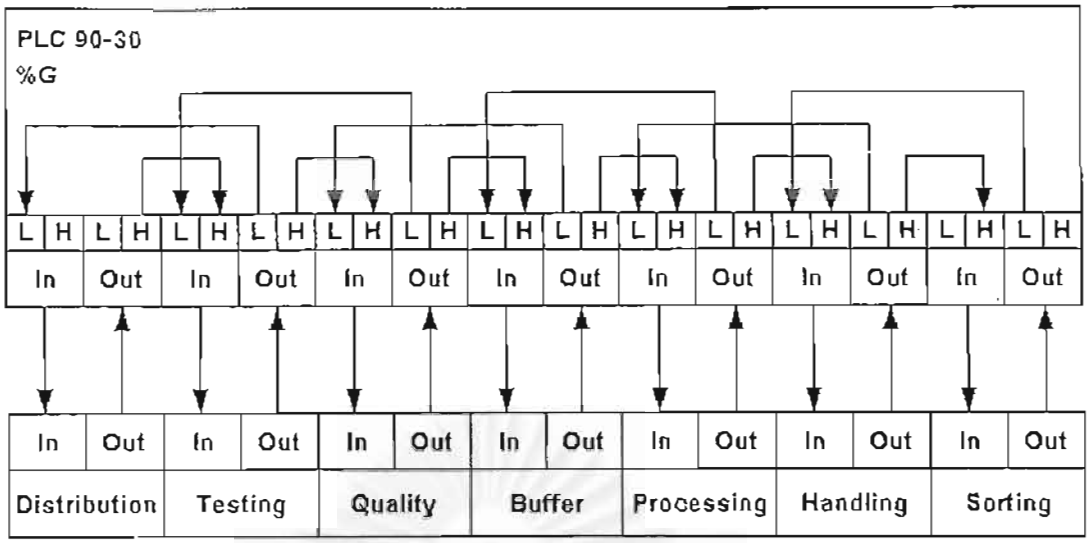
5.3 การเขียนโปรแกรมรับ-ส่งข้อมูล

การเขียนโปรแกรมแลดเดอร์รับ-ส่งข้อมูลระหว่างสถานีจะเขียนและเก็บอยู่ใน PLC รุ่น 90-30 โดยจะกำหนดหน่วยความจำส่วน %G สำหรับข้อมูลอินพุตและเอาต์พุตของสถานีต่าง ๆ ดังตารางที่ 5.3 โดย PLC รุ่น 90-30 จะเป็นตัวกลางในการรับ-ส่งข้อมูล โดยข้อมูลเอาต์พุตจากทุกสถานีจะถูกส่งเข้าไปยังหน่วยความจำ %G ของ PLC รุ่น 90-30 ในส่วนเอาต์พุตของสถานีต่างๆ จากนั้นจะทำการเคลื่อนย้ายข้อมูลไปใส่ยังอินพุตของสถานีต่างๆ และจะส่งค่าอินพุตกลับไปยังสถานีนั้น ๆ ดังรูปที่ 5.3 ตัวอย่างการเขียนโปรแกรมสำหรับรับ-ส่งข้อมูลระหว่างสถานีต่าง ๆ แสดงอยู่ในภาคผนวก ข

ตารางที่ 5.3 แสดงหน่วยความจำ %G ที่ใช้เก็บค่าอินพุตและเอาต์พุตของสถานีต่าง ๆ

%G	INPUT/OUTPUT	STATION
1-16	INPUT	DISTRIBUTION
17-32	OUTPUT	STATION
33-48	INPUT	TESTING
49-64	OUTPUT	STATION
65-80	INPUT	QUALITY
81-96	OUTPUT	STATION
97-112	INPUT	BUFFER
113-128	OUTPUT	STATION
129-144	INPUT	PROCESSING
145-160	OUTPUT	STATION
161-176	INPUT	HANDLING
177-192	OUTPUT	STATION
193-208	INPUT	SORTING
209-224	OUTPUT	STATION

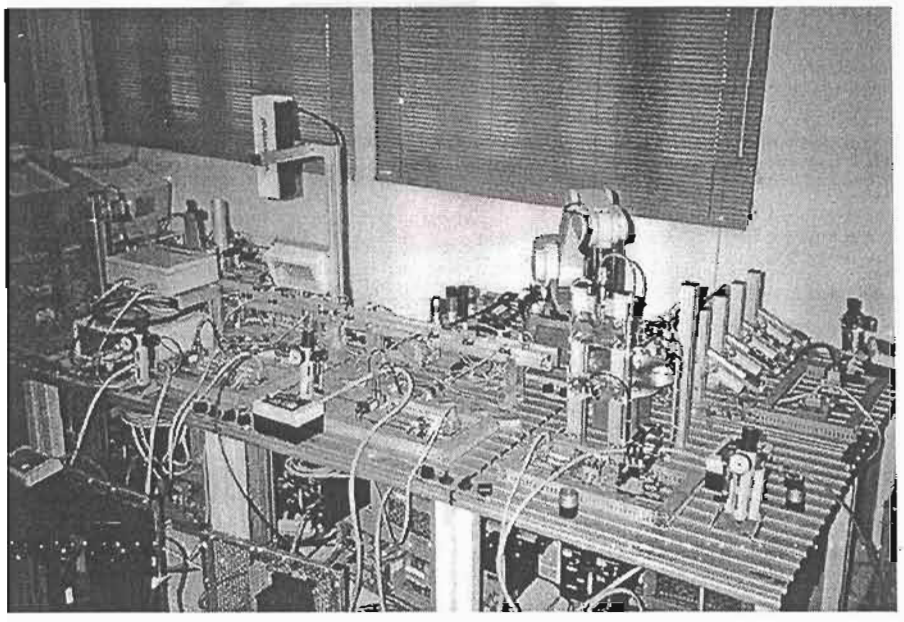
สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย



รูปที่ 5.3 แสดงการรับ-ส่งข้อมูลระหว่างสถานีต่าง ๆ

5.4 ลักษณะการทำงานของสายการผลิต

สายการผลิตเป็นดังรูปที่ 5.4 และมีลักษณะการทำงานจากโปรแกรมที่ได้เขียนขึ้นมาเป็นดังนี้



รูปที่ 5.4 แสดงสายการผลิตอัตโนมัติ 7 โมดูล

การทำงานเริ่มจากสถานีจ่ายชิ้นงาน ส่งชิ้นงานจากกระบอกใส่ชิ้นงานเข้าไปยังถาดรับชิ้นงานในสถานีทดสอบชิ้นงาน ชิ้นงานจะถูกตรวจสอบความเป็นโลหะ, สี และความสูงของชิ้นงาน และจะบันทึกข้อมูลชิ้นงานลงในบิต D0, D1, และ D2 เมื่อตรวจสอบเสร็จแล้ว ชิ้นงานจะถูกส่งต่อให้กับสถานีควบคุมคุณภาพ ซึ่งจะใช้ระบบการตรวจสอบด้วยกล้อง ตรวจสอบร่องวงแหวนบนชิ้นงานและจะบันทึกข้อมูลชิ้นงานลงในบิต D3 และ D4 จากนั้นชิ้นงานจะถูกส่งต่อไปยังสถานีพักชิ้นงาน ซึ่งทำหน้าที่พักชิ้นงานไว้ชั่วคราวได้สูงสุด 3 ชิ้น และเมื่อสถานีผลิตชิ้นงานพร้อมรับชิ้นงาน ก็จะส่งบิตข้อมูลร้องขอให้แขนกลทำการหยิบชิ้นงานจากสถานีพักชิ้นงาน มายังสถานีผลิตชิ้นงาน แล้วจะทำการเจาะรูชิ้นงานที่มีความสูงใช้ได้ รวมทั้งตรวจสอบรูเจาะบนชิ้นงานและบันทึกข้อมูลชิ้นงานลงในบิต D5 จากนั้นสถานีแขนกลจะรับชิ้นงานต่อ โดยให้แขนกลหยิบชิ้นงานจากสถานีผลิตชิ้นงาน ในกรณีที่ชิ้นงานไม่มีร่องวงแหวนแขนกลจะนำชิ้นงานไปแยกใส่กระบอกใส่ชิ้นงาน ซึ่งจะมี 4 กระบอก คือ ชิ้นงานสีแดง ชิ้นงานสีดำ ชิ้นงานโลหะ และชิ้นงานเสีย (ความสูงหรือรูเจาะใช้ไม่ได้) ส่วนในกรณีที่ชิ้นงานมีร่องวงแหวน แขนกลจะไปวางชิ้นงานที่สถานีแยกชิ้นงาน จากนั้นที่สถานีแยกชิ้นงานสายพานจะหมุน และเปิดตัวกั้นบังคับให้ชิ้นงานไหลลงในรางรับชิ้นงาน ตามชนิดและสีของชิ้นงาน ซึ่งจะมีรางรับชิ้นงานทั้งหมด 4 ราง คือ ชิ้นงานสีแดง ชิ้นงานสีดำ ชิ้นงานโลหะ และชิ้นงานเสีย (ขนาดหรือความหนาของร่องวงแหวนใช้ไม่ได้อย่างใดอย่างหนึ่ง หรือความสูงใช้ไม่ได้ หรือรูเจาะใช้ไม่ได้)

บทที่ 6
พัฒนาโปรแกรมสำหรับการตรวจดูระบบโดยใช้ซอฟต์แวร์ CIMPLICITY HMI

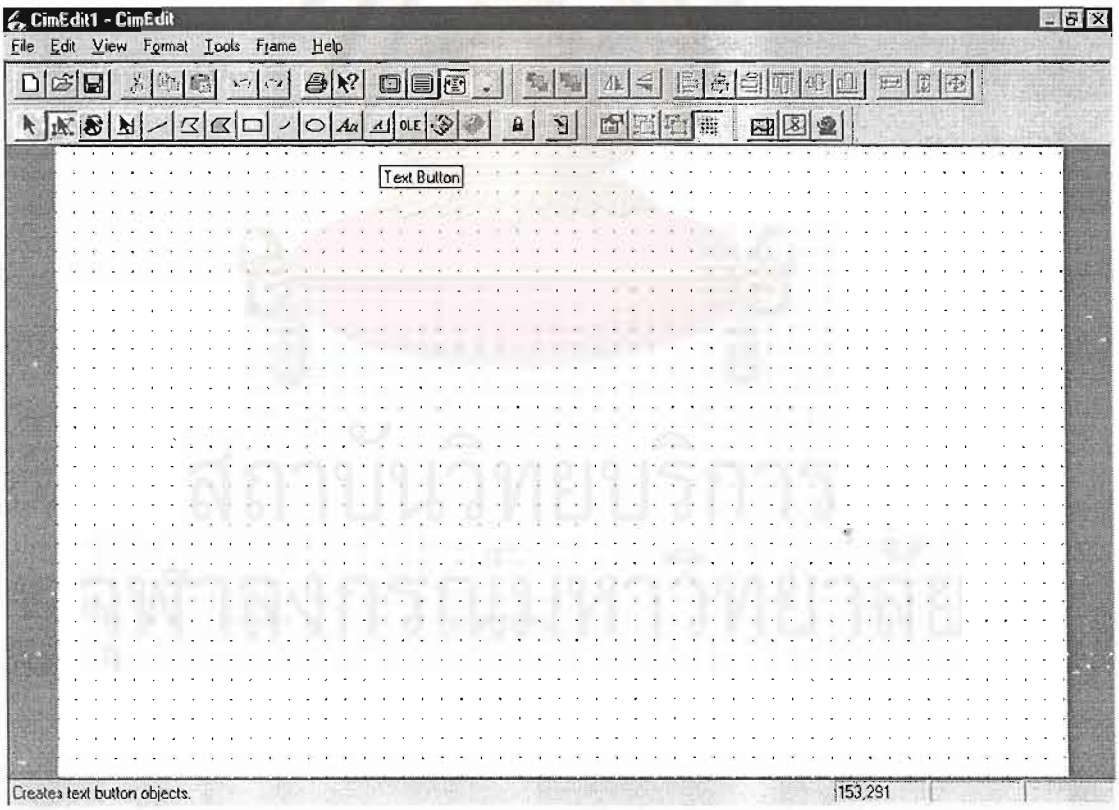
ในการตรวจดูระบบจะใช้คอมพิวเตอร์หลักทำหน้าที่เป็นเซริฟเวอร์ ทำการติดต่อสื่อสารกับตัว PLC รุ่น 90-30 และ PLC รุ่น 90-70 ผ่านโมดูล Ethernet Interface ข้อมูลที่ใช้ในการตรวจดูสถานีต่างๆ จะถูกส่งเข้าสู่เครือข่ายอีเทอร์เน็ต และเข้าสู่เซริฟเวอร์ โดยผ่าน PLC รุ่น 90-30 ยกเว้นสถานีควบคุมคุณภาพสามารถส่งข้อมูลการตรวจดูไปยังเซริฟเวอร์ได้โดยตรง

การเขียนโปรแกรมสำหรับตรวจดูระบบโดยใช้ซอฟต์แวร์ CIMPLICITY HMI ซึ่งมีตัวอย่างหน้าจอการทำงานในส่วนของการสร้างกลุ่มรูปภาพเป็นดังรูปที่ 6.1 มีขั้นตอนหรือวิธีโดยสังเขปดังนี้

1. กำหนดจุดข้อมูล (Data Point) ขึ้นมา โดยแต่ละจุดข้อมูลเป็นข้อมูลจากสถานีต่าง ๆ ซึ่งจะรวบรวมไว้เป็นกลุ่มตามสถานีนั้น ๆ

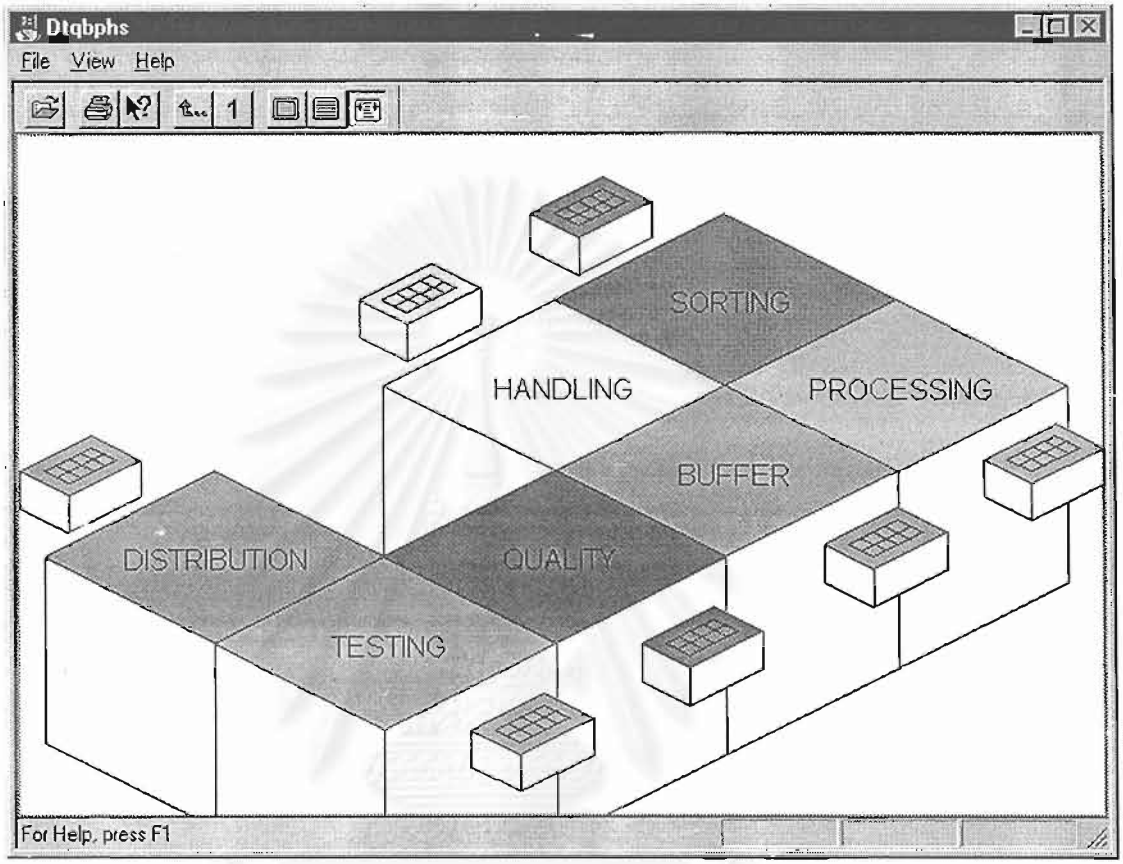
2. ทำการสร้างกลุ่มรูปภาพ แทนลักษณะทางกายภาพของสถานีนั้น ๆ

3. เชื่อมโยงรูปภาพเข้ากับจุดข้อมูล พร้อมทั้งกำหนดการเคลื่อนไหวหรือลักษณะการเปลี่ยนแปลงต่าง ๆ เพื่อให้รูปภาพสามารถเคลื่อนไหวได้ตามการทำงานจริงในสถานีนั้น ๆ

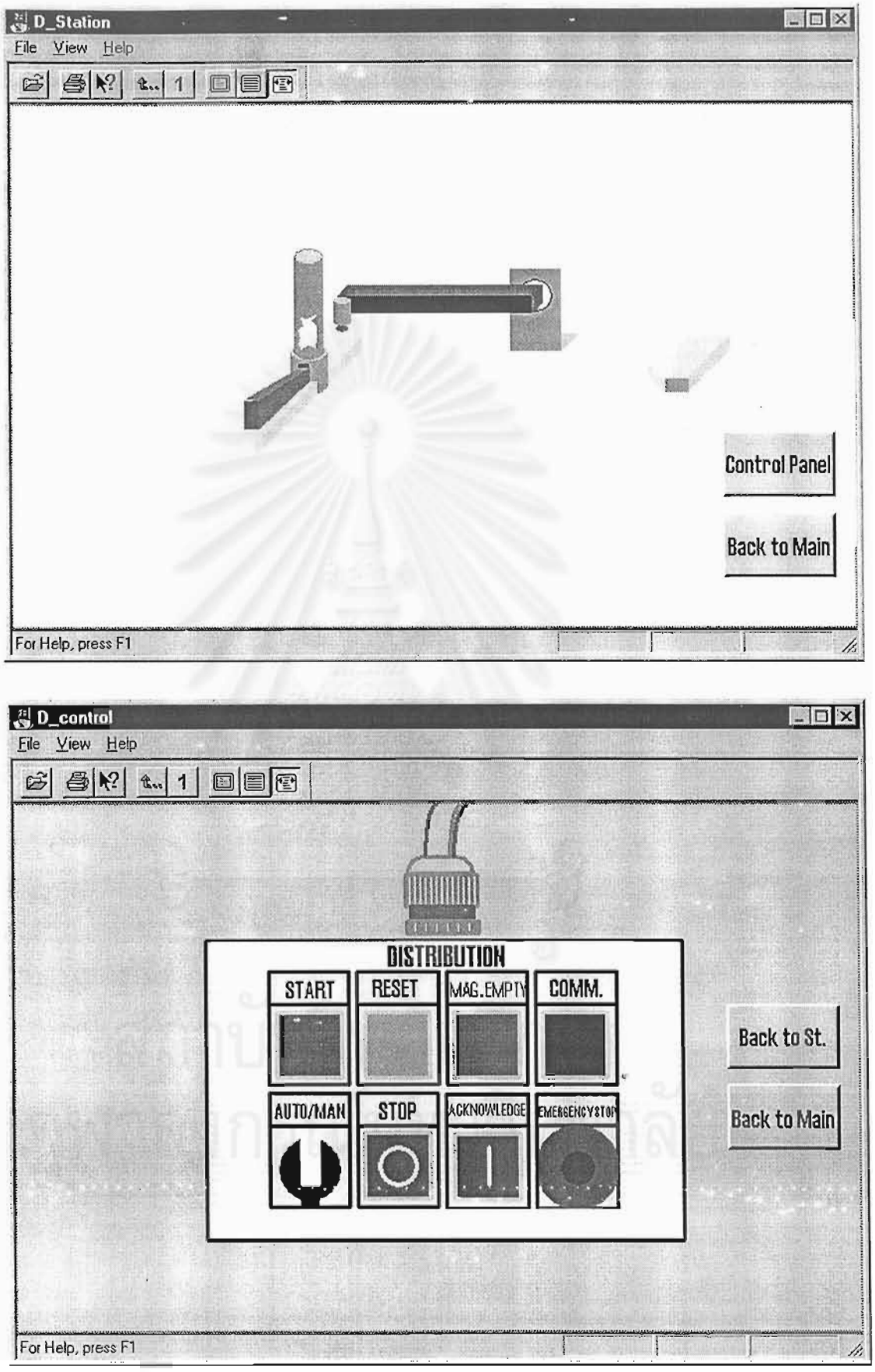


รูปที่ 6.1 ตัวอย่างหน้าจอการทำงานในส่วนของการสร้างกลุ่มรูปภาพ

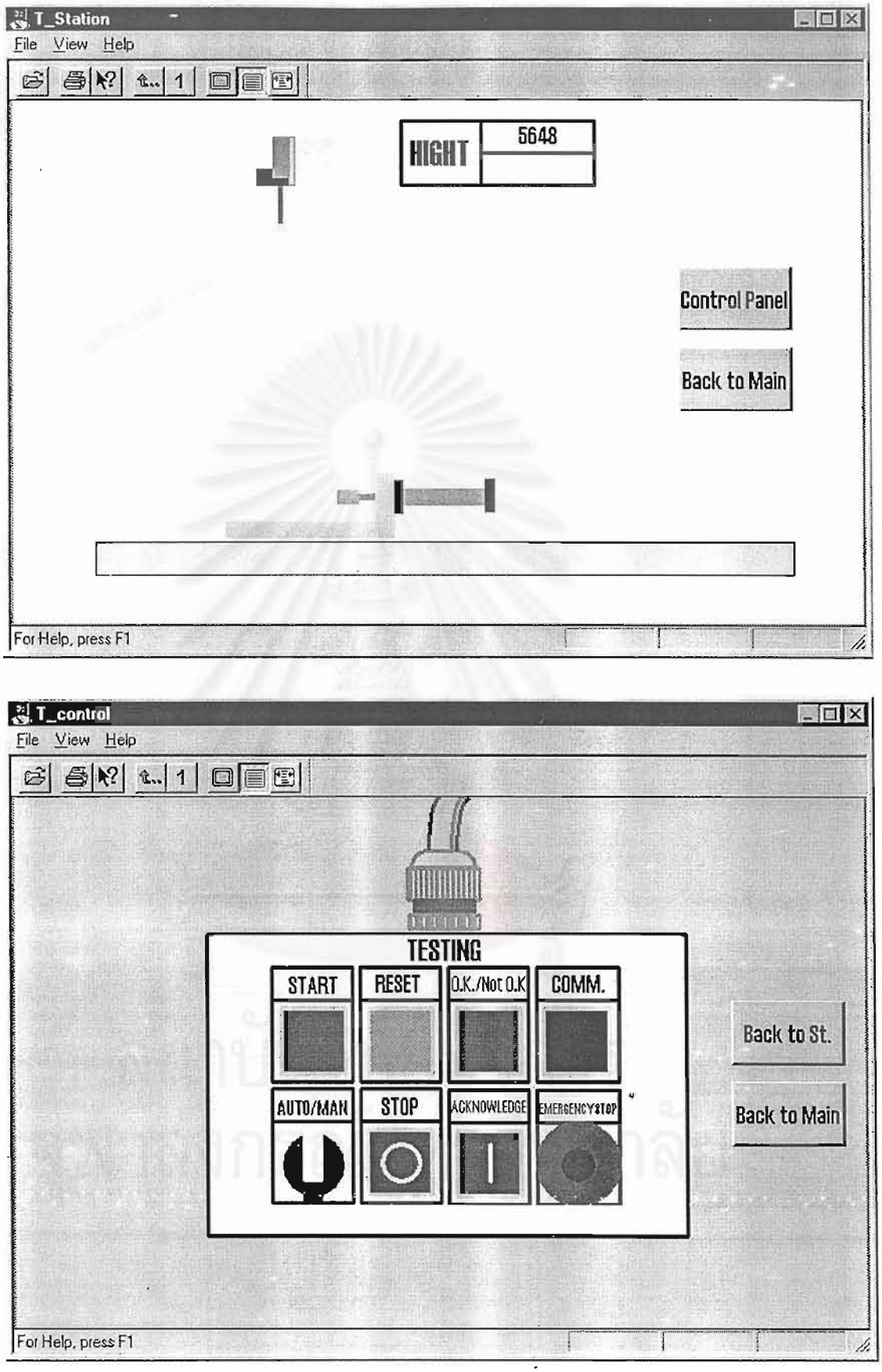
โปรแกรมที่ได้พัฒนาสามารถแสดงหน้าจอการตรวจดู โดยมีหน้าจอหลักสามารถคลิกเข้าไปตรวจดูสถานี และกล่องควบคุมของสถานีทั้ง 7 สถานี ดังรูปที่ 6.2 และมีหน้าจอของสถานีต่าง ๆ ซึ่งแสดงลักษณะและการทำงานในสถานีนั้น ๆ ดังรูปที่ 6.3, 6.4, 6.5, 6.6, 6.7, 6.8 และ 6.9



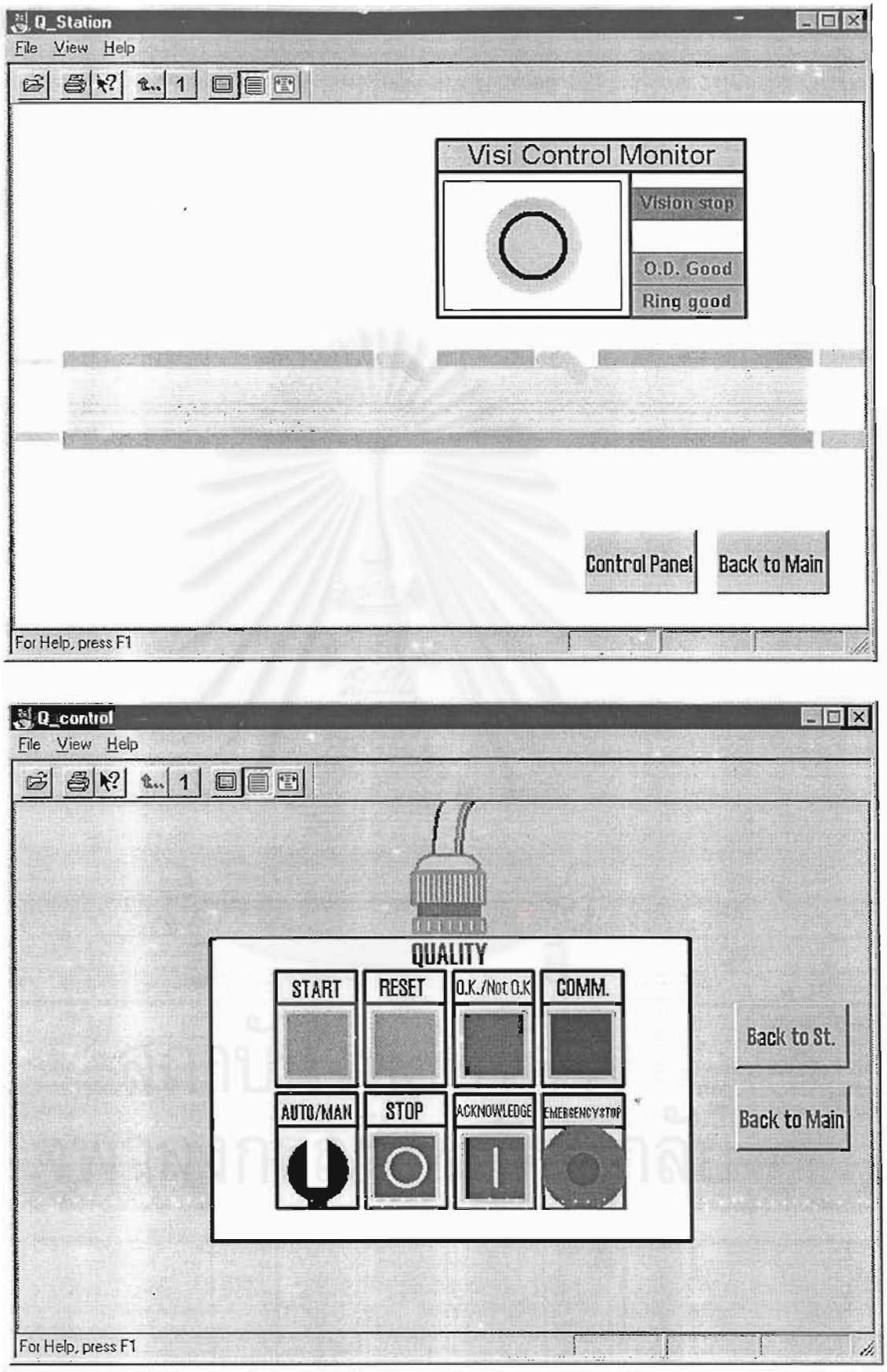
รูปที่ 6.2 แสดงภาพบนจอคอมพิวเตอร์ในหน้าจอหลัก



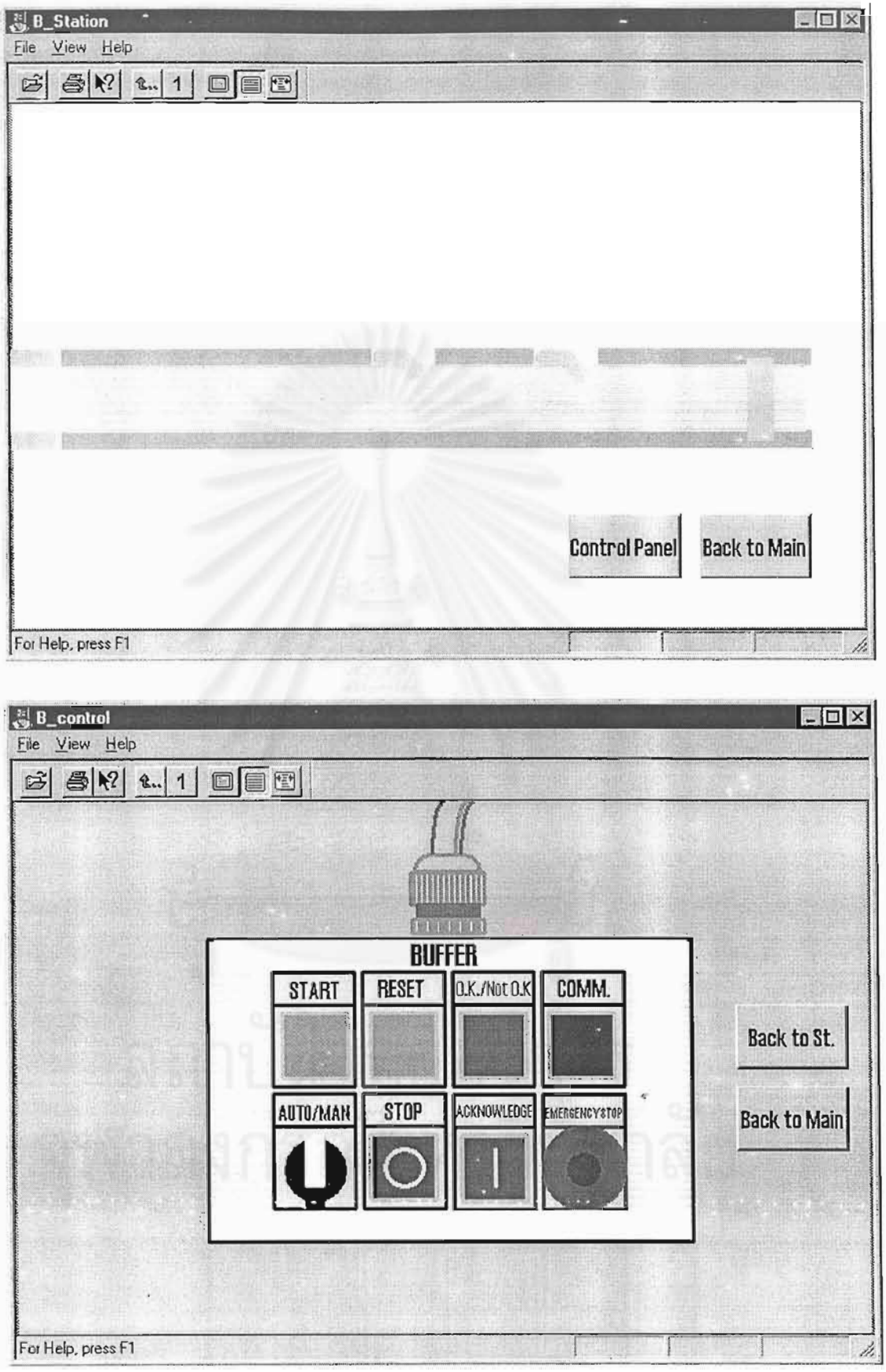
รูปที่ 6.3 แสดงภาพบนจอคอมพิวเตอร์ในการตรวจดูสถานีจ่ายชิ้นงาน



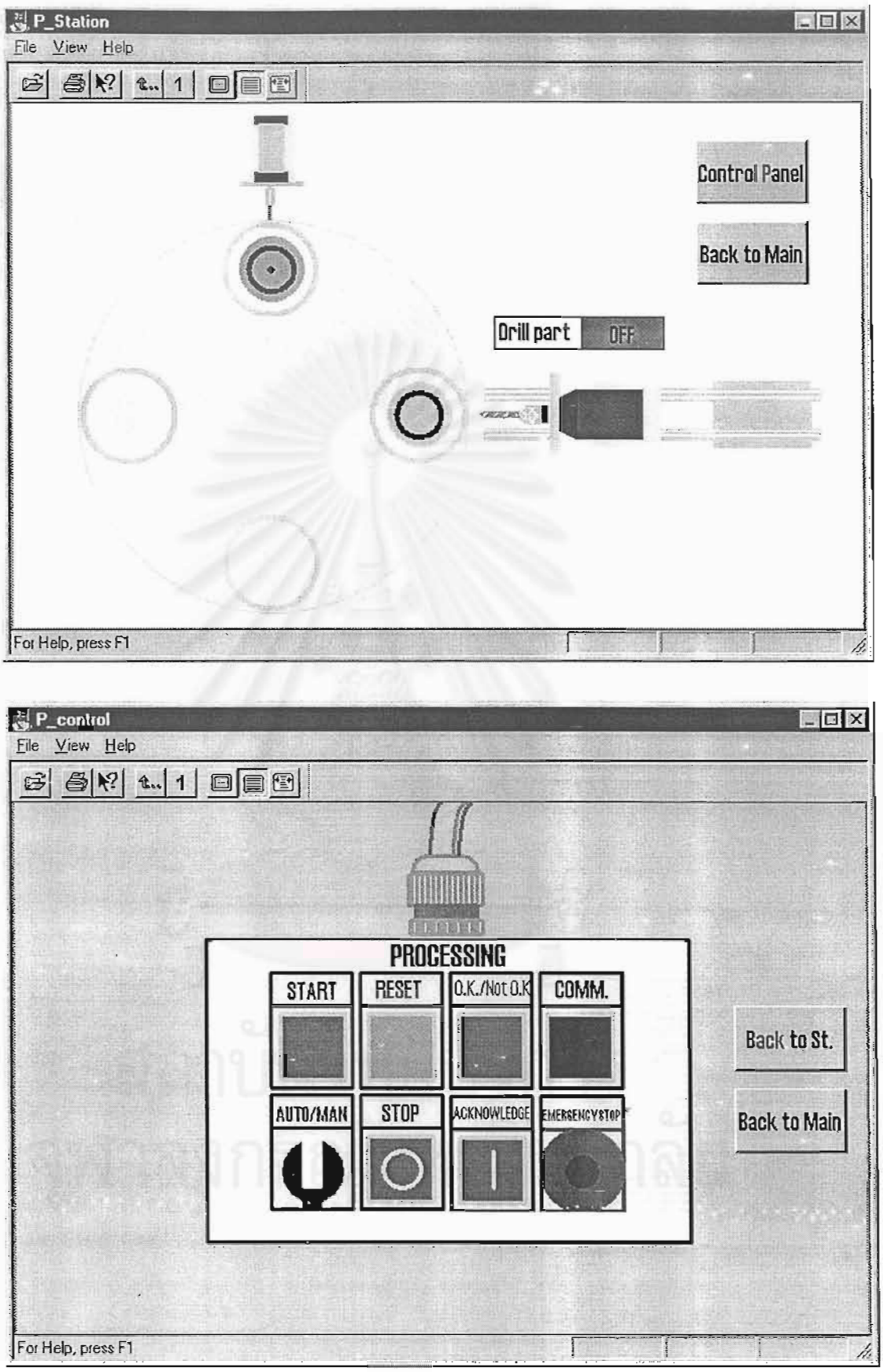
รูปที่ 6.4 แสดงภาพบนจอคอมพิวเตอร์ในการตรวจดูสถานีทดสอบชิ้นงาน



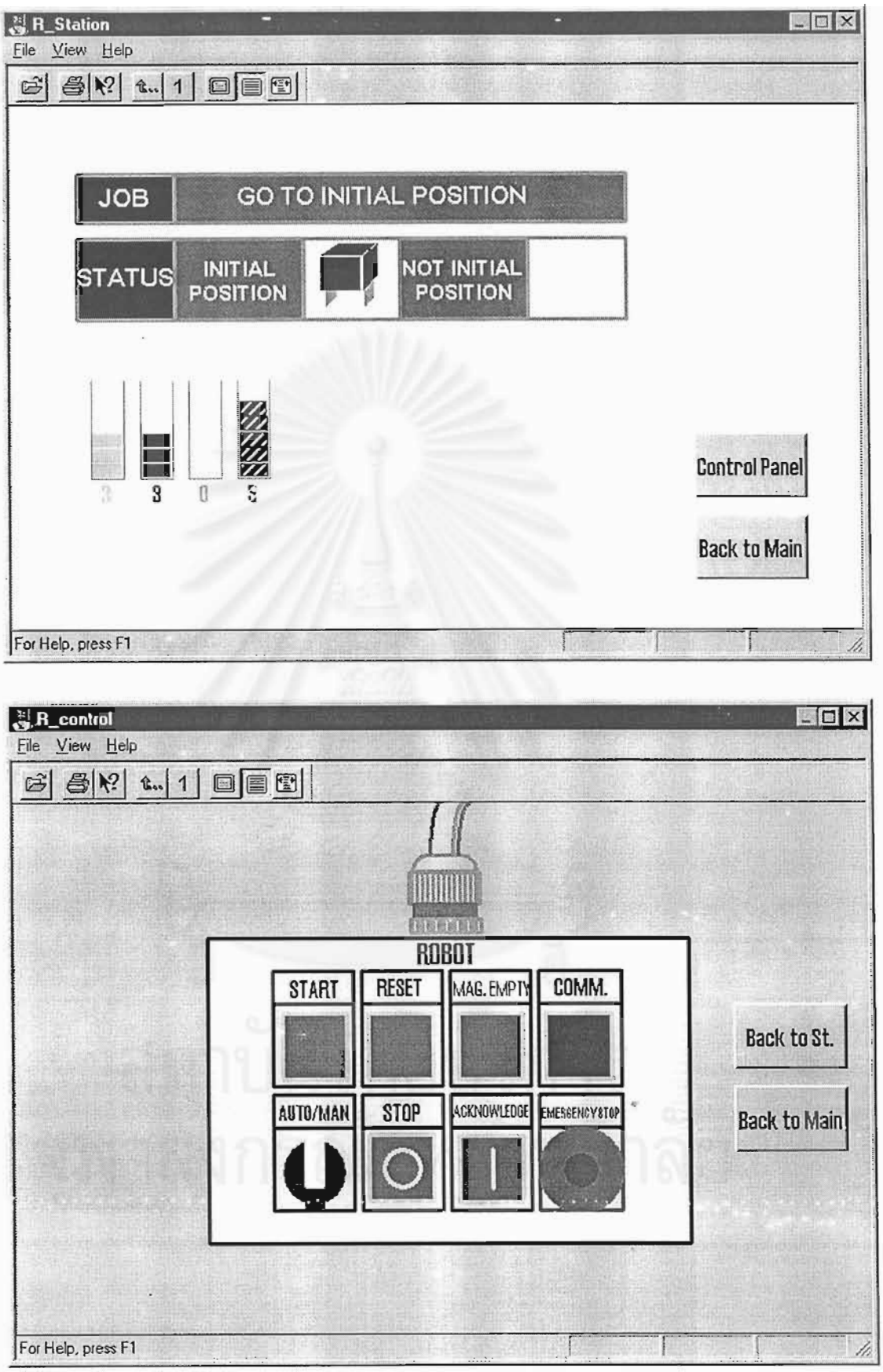
รูปที่ 6.5 แสดงภาพบนจอคอมพิวเตอร์ในการตรวจดูสถานีควบคุมคุณภาพ



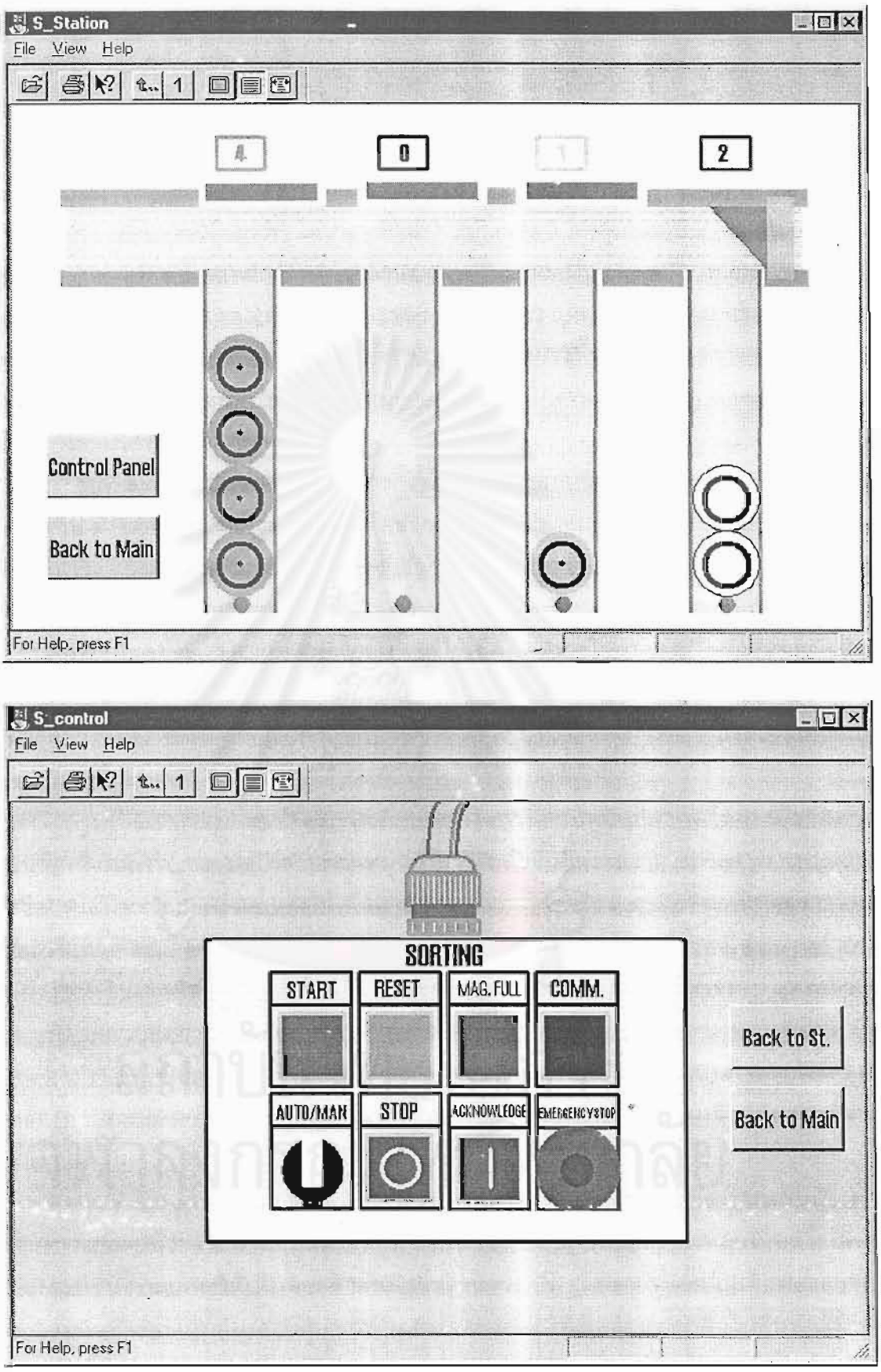
รูปที่ 6.6 แสดงภาพบนจอคอมพิวเตอร์ในการตรวจดูสถานีพักชิ้นงาน



รูปที่ 6.7 แสดงภาพบนจอคอมพิวเตอร์ในการตรวจดูสถานีผลิตชิ้นงาน



รูปที่ 6.8 แสดงภาพบนจอคอมพิวเตอร์ในการตรวจดูสถานีแขนกล



รูปที่ 6.9 แสดงภาพบนจอคอมพิวเตอร์ในการตรวจดูสถานีแยกชิ้นงาน

รูปที่ 6.2 แสดงหน้าจอหลัก ซึ่งสามารถคลิกเข้าไปดรวจดูสถานี และกล่องควบคุมของสถานีทั้ง 7 สถานี

รูปที่ 6.3 ด้านบนจะแสดงภาพจำลองของสถานีจ่ายชิ้นงาน โดยจะมีกระบอกใส่ชิ้นงาน ซึ่งสามารถแสดงให้เห็นว่ามีชิ้นงานอยู่หรือไม่ ตัวดันชิ้นงานสามารถเคลื่อนที่แสดงการทำงานได้ แขนหมุนสามารถหมุนไป-กลับ และจะมีชิ้นงานแสดงให้เห็นตามตำแหน่งชิ้นงานจริง ปุ่ม Control Panel ใช้เมื่อต้องการไปตรวจดูที่หน้าจอของกล่องควบคุม ปุ่ม Back to Main ใช้เมื่อต้องการกลับไปที่หน้าจอหลัก ส่วนภาพด้านล่างจะเป็นภาพหน้าจอของกล่องควบคุมของสถานีจ่ายชิ้นงาน สามารถแสดงให้เห็นสถานะของปุ่มต่าง ๆ ว่าถูกกดอยู่หรือไม่ หรือสวิตช์หมุน AUTO/MAN อยู่ที่ตำแหน่งใด และสามารถแสดงไฟที่สว่างขึ้นในปุ่มกดต่าง ๆ สำหรับปุ่ม Back to St. ใช้เมื่อต้องการไปดรวจดูที่หน้าจอสถานีจ่ายชิ้นงาน และปุ่ม Back to Main ใช้เมื่อต้องการกลับไปที่หน้าจอหลัก

รูปที่ 6.4 ด้านบนจะแสดงภาพจำลองของสถานีทดสอบชิ้นงาน โดยจะมีถาดยกสามารถเลื่อนขึ้น-ลงได้ ตัวดันชิ้นงานสามารถยืดหรือหด และจะมีชิ้นงานแสดงให้เห็นตามตำแหน่งชิ้นงานจริง และเมื่อชิ้นงานถูกวัดความสูงแล้ว จะแสดงผลการวัดในตาราง HIGHT โดยช่องบนจะเป็นค่าที่วัดได้จาก เซนเซอร์วัดความสูง และช่องล่างจะเป็นผลการตรวจสอบ ถ้าความสูงใช้ได้จะแสดงข้อความ O.K. ปุ่ม Control Panel ใช้เมื่อต้องการไปตรวจดูที่หน้าจอของกล่องควบคุม ปุ่ม Back to Main ใช้เมื่อต้องการกลับไปที่หน้าจอหลัก ส่วนภาพด้านล่างจะเป็นภาพหน้าจอของกล่องควบคุมของสถานีทดสอบชิ้นงาน สามารถแสดงการทำงานได้เช่นเดียวกับสถานีจ่ายชิ้นงาน

รูปที่ 6.5 ด้านบนจะแสดงภาพจำลองของสถานีควบคุมคุณภาพ โดยจะมีสายพานลำเลียงซึ่งสามารถแสดงได้ว่าสายพานหมุนหรือหยุด ตัวกั้นชิ้นงานสามารถแสดงการหมุนเปิดหรือปิดได้ และจะมีชิ้นงานแสดงให้เห็นตามตำแหน่งชิ้นงานจริง และเมื่อชิ้นงานถูกตรวจสอบร่องวงแหวน จะแสดงผลการตรวจสอบในตาราง Visi Control Monitor ซึ่งช่องใหญ่ด้านซ้ายจะแสดงภาพชิ้นงานที่ตรวจสอบเสร็จแล้ว ส่วนช่องเล็กด้านขวาเป็นผลการตรวจสอบ และสถานะของระบบตรวจสอบ ปุ่ม Control Panel ใช้เมื่อต้องการไปตรวจดูที่หน้าจอของกล่องควบคุม ปุ่ม Back to Main ใช้เมื่อต้องการกลับไปที่หน้าจอหลัก ส่วนภาพด้านล่างจะเป็นภาพหน้าจอของกล่องควบคุมของสถานีควบคุมคุณภาพ สามารถแสดงการทำงานได้เช่นเดียวกับสถานีจ่ายชิ้นงาน

รูปที่ 6.6 ด้านบนจะแสดงภาพจำลองของสถานีพักชิ้นงาน โดยจะมีสายพานลำเลียงซึ่งสามารถแสดงได้ว่าสายพานหมุนหรือหยุด ตัวกั้นชิ้นงานสามารถแสดงการหมุนเปิดหรือปิดได้ และจะมีชิ้นงานแสดงให้เห็นตามตำแหน่งชิ้นงานจริง ปุ่ม Control Panel ใช้เมื่อต้องการไปตรวจดูที่หน้าจอของกล่องควบคุม และปุ่ม Back to Main ใช้เมื่อต้องการกลับไปที่หน้าจอหลัก ส่วนภาพด้านล่างจะเป็นภาพหน้าจอของกล่องควบคุมของสถานีพักชิ้นงาน สามารถแสดงการทำงานได้เช่นเดียวกับสถานีจ่ายชิ้นงาน

รูปที่ 6.7 ด้านบนจะแสดงภาพจำลองของสถานีผลิตชิ้นงาน โดยจะมีโต๊ะหมุนสามารถแสดงได้ว่ากำลังหมุนหรือหยุด และจะมีชิ้นงานแสดงให้เห็นตามตำแหน่งชิ้นงานจริง ที่ตำแหน่งที่ 2 เมื่อชิ้นงานถูกรัดแน่นแล้วจะมีข้อความ Clamped แสดงขึ้นมาด้านบนชิ้นงาน และจะมีภาพจำลองของหัวเจาะรู ซึ่งสามารถเลื่อนแสดงการเจาะรูได้ พร้อมทั้งแสดงสถานะ การหมุนของดอกสว่านในตาราง Drill part ที่ตำแหน่งที่ 3 จะแสดงการตรวจสอบรูเจาะบนชิ้นงาน โดยก้านทดสอบรูสามารถเลื่อนขึ้น-ลงแสดงการทดสอบรูเจาะ และถ้ารูเจาะใช้ได้จะแสดงให้เห็นเป็นจุดบนชิ้นงาน ปุ่ม Control Panel ใช้เมื่อต้องการไปตรวจดูที่หน้าจอของกล่องควบคุม ปุ่ม Back to Main ใช้เมื่อต้องการกลับไปที่หน้าจอหลัก ส่วนภาพด้านล่างจะเป็นภาพหน้าจอของกล่องควบคุมของสถานีผลิตชิ้นงาน สามารถแสดงการทำงานได้เช่นเดียวกับสถานีจ่ายชิ้นงาน

รูปที่ 6.8 ด้านบนจะแสดงภาพจำลองของสถานีแขนกล ซึ่งจะมีตาราง JOB แสดงคำสั่งการทำงานของแขนกลขณะนั้น และตาราง STATUS แสดงสถานะของแขนกล โดยจะมีช่องแสดงมือจับของแขนกล 2 ช่อง ถ้าแขนกลอยู่ที่ตำแหน่งเริ่มต้นจะแสดงให้เห็นภาพมือจับในช่อง INITIAL POSITION แต่ถ้าแขนกลไม่ได้อยู่ที่ตำแหน่งเริ่มต้นจะแสดงให้เห็นภาพมือจับในช่อง NOT INITIAL POSITION ในแต่ละช่อง ถ้าขณะนั้นแขนกลกำลังหยิบจับชิ้นงานอยู่จะแสดงให้เห็นภาพชิ้นงานที่มือจับของแขนกล และถัดลงมาจะมีภาพแสดงกระบอกใส่ชิ้นงาน 4 กระบอก ซึ่งสามารถแสดงภาพชนิดและจำนวนชิ้นงานที่มีอยู่ในกระบอกนั้น ๆ และเมื่อชิ้นงานเต็มจะแสดงข้อความ FULL ปุ่ม Control Panel ใช้เมื่อต้องการไปตรวจดูที่หน้าจอของกล่องควบคุม ปุ่ม Back to Main ใช้เมื่อต้องการกลับไปที่หน้าจอหลัก ส่วนภาพด้านล่างจะเป็นภาพหน้าจอของกล่องควบคุมของสถานีแขนกล สามารถแสดงการทำงานได้เช่นเดียวกับสถานีจ่ายชิ้นงาน

รูปที่ 6.9 ด้านบนจะแสดงภาพจำลองของสถานีแยกชิ้นงาน โดยจะมีสายพานลำเลียงซึ่งสามารถแสดงได้ว่าสายพานหมุนหรือหยุด ตัวกั้นบังคับชิ้นงานสามารถแสดงการเปิดกั้นหรือปิดได้ และจะมีชิ้นงานแสดงให้เห็นตามตำแหน่งชิ้นงานจริง รางรับชิ้นงานจะมี 4 รางซึ่งสามารถแสดงภาพชนิดและจำนวนชิ้นงานที่มีอยู่บนรางนั้น เมื่อชิ้นงานเต็มจะแสดงข้อความ FULL ปุ่ม Control Panel ใช้เมื่อต้องการไปตรวจดูที่หน้าจอของกล่องควบคุม ปุ่ม Back to Main ใช้เมื่อต้องการกลับไปที่หน้าจอหลัก ส่วนภาพด้านล่างจะเป็นภาพหน้าจอของกล่องควบคุมของสถานีแยกชิ้นงาน สามารถแสดงการทำงานได้เช่นเดียวกับสถานีจ่ายชิ้นงาน

จุฬาลงกรณ์มหาวิทยาลัย

บทที่ 7
การทดลองปรับเปลี่ยนสายการผลิตในอีก 6 รูปแบบ

7.1 การปรับเปลี่ยนรูปแบบของสายการผลิต

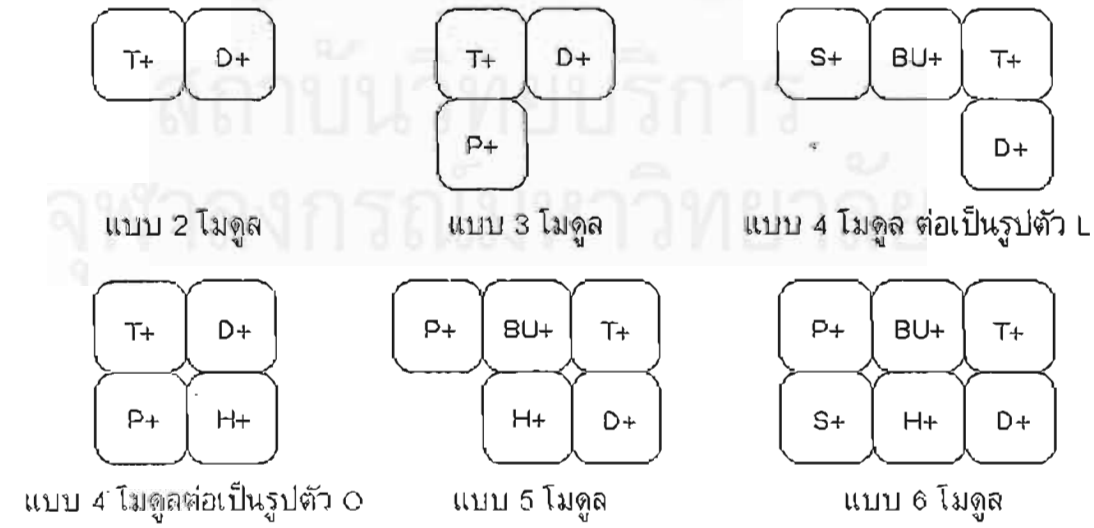
ในการปรับเปลี่ยนรูปแบบของสายการผลิต จะต้องมีการปรับเปลี่ยนทั้งทางด้านซอฟต์แวร์ และฮาร์ดแวร์ ในส่วนของซอฟต์แวร์ หรือโปรแกรมในตัว PLC จะมีการปรับเปลี่ยนในส่วนของการรับ-ส่งข้อมูลระหว่างสถานี และอาจจะมีการปรับเปลี่ยนในส่วนของโปรแกรมการทำงานด้วย เพื่อให้เหมาะสมกับรูปแบบของสายการผลิตใหม่ ซึ่งในการปรับเปลี่ยนโปรแกรมในตัว PLC เราอาจจะมีการเตรียมโปรแกรมไว้ล่วงหน้า เพื่อให้เวลาปรับรูปแบบสายการผลิตสามารถทำได้รวดเร็วมากขึ้น

ขั้นตอนในการปรับเปลี่ยนรูปแบบของสายการผลิต เป็นดังนี้

1. ปิดการทำงานทั้งหมด ปลดสายลม, สายไฟ รวมทั้งสายเครือข่ายการสื่อสารออก
2. จัดเรียงโมดูลใหม่ตามรูปแบบของสายการผลิตที่ต้องการเปลี่ยน และปรับอุปกรณ์บนโมดูลให้สายการผลิตมีความต่อเนื่อง
3. ต่อสายลม, สายไฟ และสายเครือข่ายการสื่อสาร
4. ใส่โปรแกรมการทำงานใหม่ที่ได้เตรียมไว้ลงใน PLC ทุกตัว
5. เปิดการทำงานทั้งหมด เริ่มทำงานในสายการผลิตใหม่

7.2 เครือข่ายการควบคุมในสายการผลิตอีก 6 รูปแบบ

สายการผลิตสามารถปรับเปลี่ยนไปได้อีก 6 รูปแบบ ดังรูปที่ 7.1 โดยในแต่ละรูปแบบมีเครือข่ายการสื่อสารและลักษณะการทำงานดังต่อไปนี้



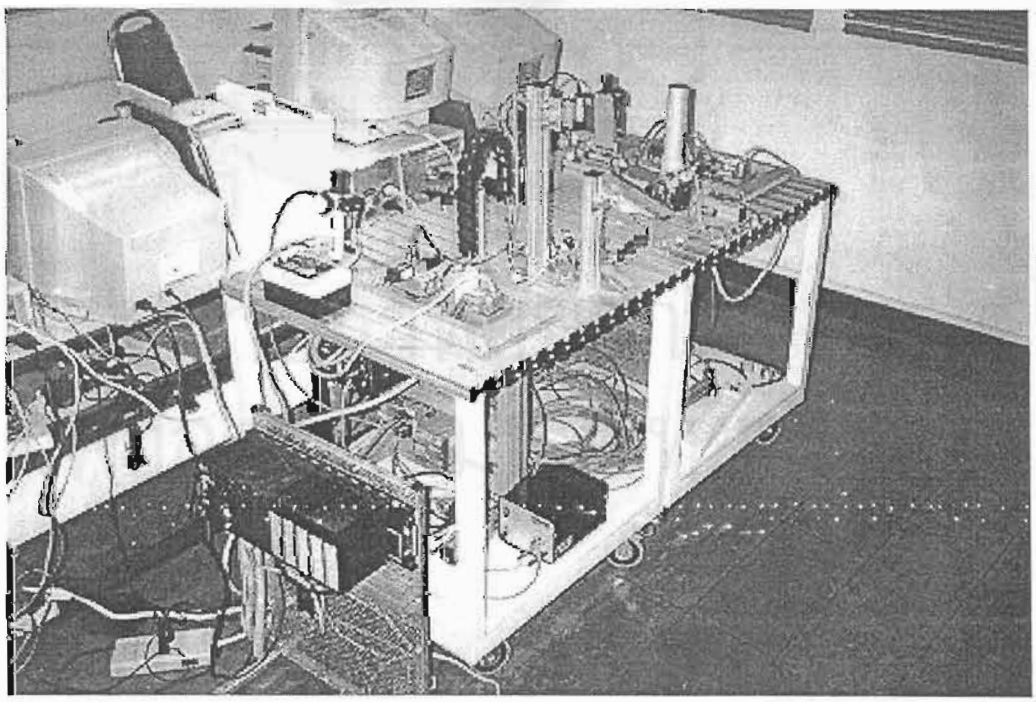
รูปที่ 7.1 แสดงสายการผลิตในอีก 6 รูปแบบ

1. <u>สายการผลิตแบบ 2 โมดูล</u> ประกอบไปด้วยสถานีจ่ายชิ้นงาน และสถานีทดสอบชิ้นงาน ดังรูปที่ 7.2 เครือข่ายการสื่อสารที่ใช้ และการรับ-ส่งข้อมูลระหว่างสถานีเป็นดังรูปที่ 7.3 และมีลักษณะการทำงานของสายการผลิตจากโปรแกรมที่ได้เขียนขึ้นมาเป็นดังนี้

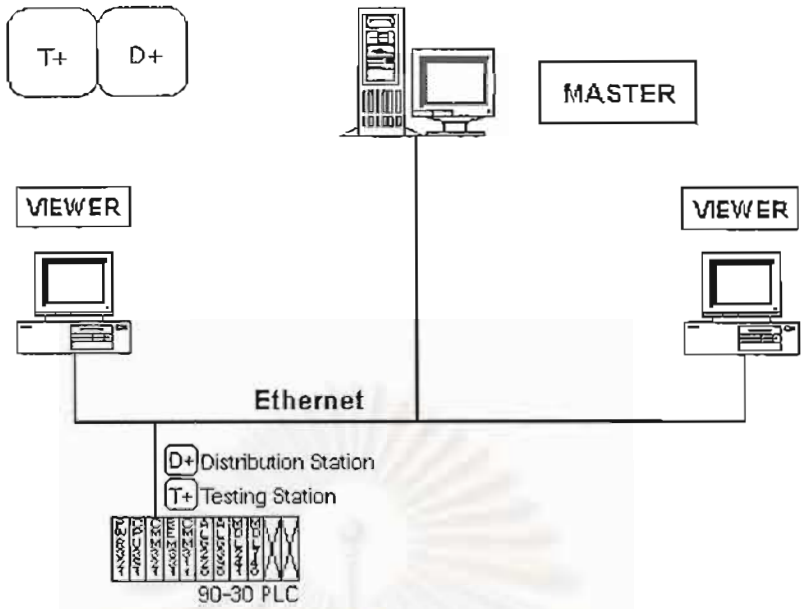
การทำงานจะเริ่มจากสถานีจ่ายชิ้นงาน ส่งชิ้นงานจากกระบอกใส่ชิ้นงานเข้าไปยังถาดรับชิ้นงานในสถานีทดสอบชิ้นงาน จากนั้นชิ้นงานจะถูกยกขึ้นไปเพื่อทดสอบความสูง ถ้าชิ้นงานมีความสูงใช้ได้หรืออยู่ในช่วงที่กำหนด ชิ้นงานจะถูกดันออกที่ด้านบน แต่ถ้าชิ้นงานสูงไปหรือต่ำไป ถาดยกจะเลื่อนลงมาแล้วดันชิ้นงานออกที่ด้านล่าง

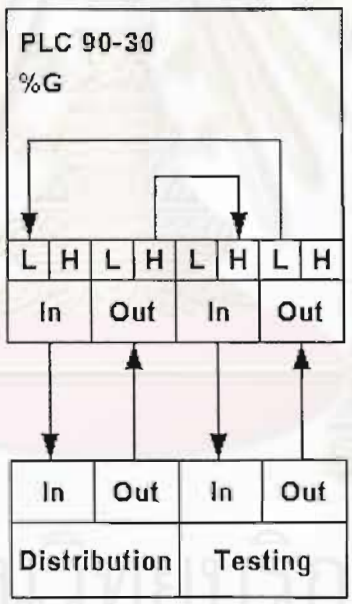
2. <u>สายการผลิตแบบ 3 โมดูล</u> ประกอบไปด้วยสถานีจ่ายชิ้นงาน, สถานีทดสอบชิ้นงาน และสถานีผลิตชิ้นงาน ดังรูปที่ 7.4 เครือข่ายการสื่อสารที่ใช้และการรับ-ส่งข้อมูลระหว่างสถานีเป็นดังรูปที่ 7.5 และมีลักษณะการทำงานของสายการผลิตจากโปรแกรมที่ได้เขียนขึ้นมาเป็นดังนี้

การทำงานจะเริ่มจากสถานีจ่ายชิ้นงาน ส่งชิ้นงานจากกระบอกใส่ชิ้นงานเข้า ไปยังถาดรับชิ้นงาน ในสถานีทดสอบชิ้นงาน จากนั้นชิ้นงานจะถูกยกขึ้นไปเพื่อทดสอบความสูง ถ้าชิ้นงานมีความสูงใช้ได้หรืออยู่ในช่วงที่กำหนด ชิ้นงานจะถูกดันออกที่ด้านบน แต่ถ้าชิ้นงานสูงไปหรือต่ำไป ถาดยกจะเลื่อนลงมาแล้วดันชิ้นงานออกที่ด้านล่าง ในกรณีที่ชิ้นงานถูกดันออกที่ด้านบนชิ้นงานจะไหลเข้าสู่สถานีผลิตชิ้นงานซึ่งจะทำการเจาะรูบนชิ้นงาน และหมุนออกมาเพื่อให้หยิบชิ้นงานออก และเมื่อหยิบชิ้นงานออกแล้วให้กดปุ่ม O.K./NOT O.K. เพื่อให้ทำงานต่อไป



รูปที่ 7.2 แสดงสายการผลิตแบบ 2 โมดูล





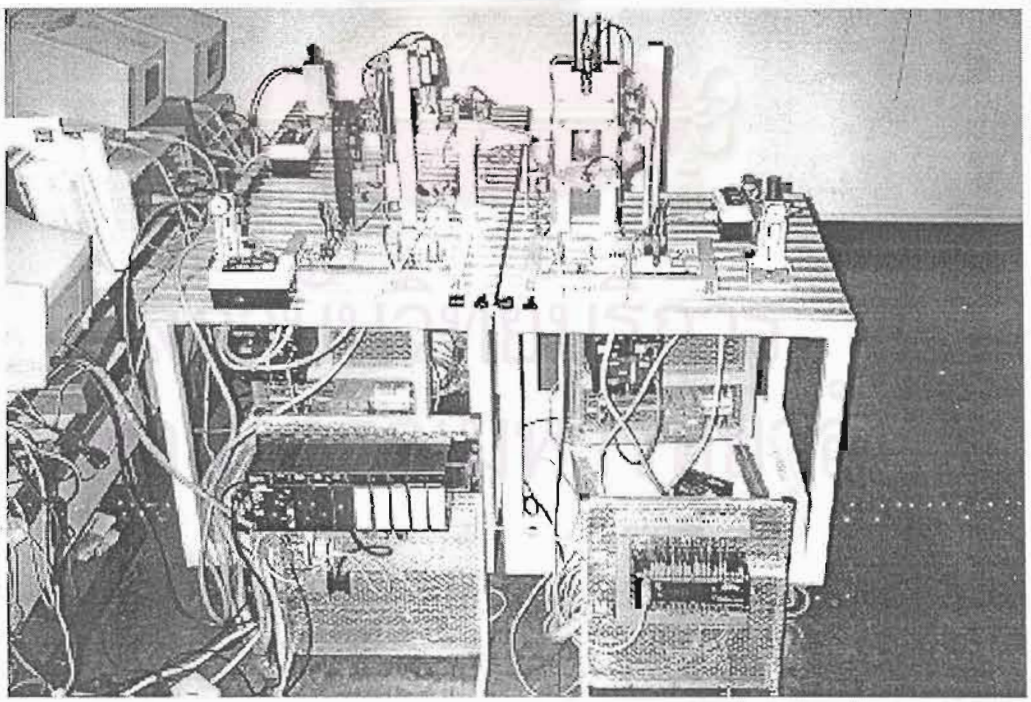
รูปที่ 7.3 แสดงเครือข่ายการสื่อสารและการรับ-ส่งข้อมูลระหว่างสถานีของสายการผลิตแบบ 2 โมดูล

3. <u>สายการผลิตแบบ 4 โมดูล ต่อเป็นรูปตัว O</u> ประกอบไปด้วยสถานีจ่ายชิ้นงาน, สถานีทดสอบชิ้นงาน, สถานีผลิตชิ้นงาน และสถานีแขนกล ดังรูปที่ 7.6 เครือข่ายการสื่อสารที่ใช้และการรับ-ส่งข้อมูลระหว่างสถานีเป็นดังรูปที่ 7.7 และมีลักษณะการทำงานของสายการผลิตจากโปรแกรมที่ได้เขียนขึ้นมาเป็นดังนี้

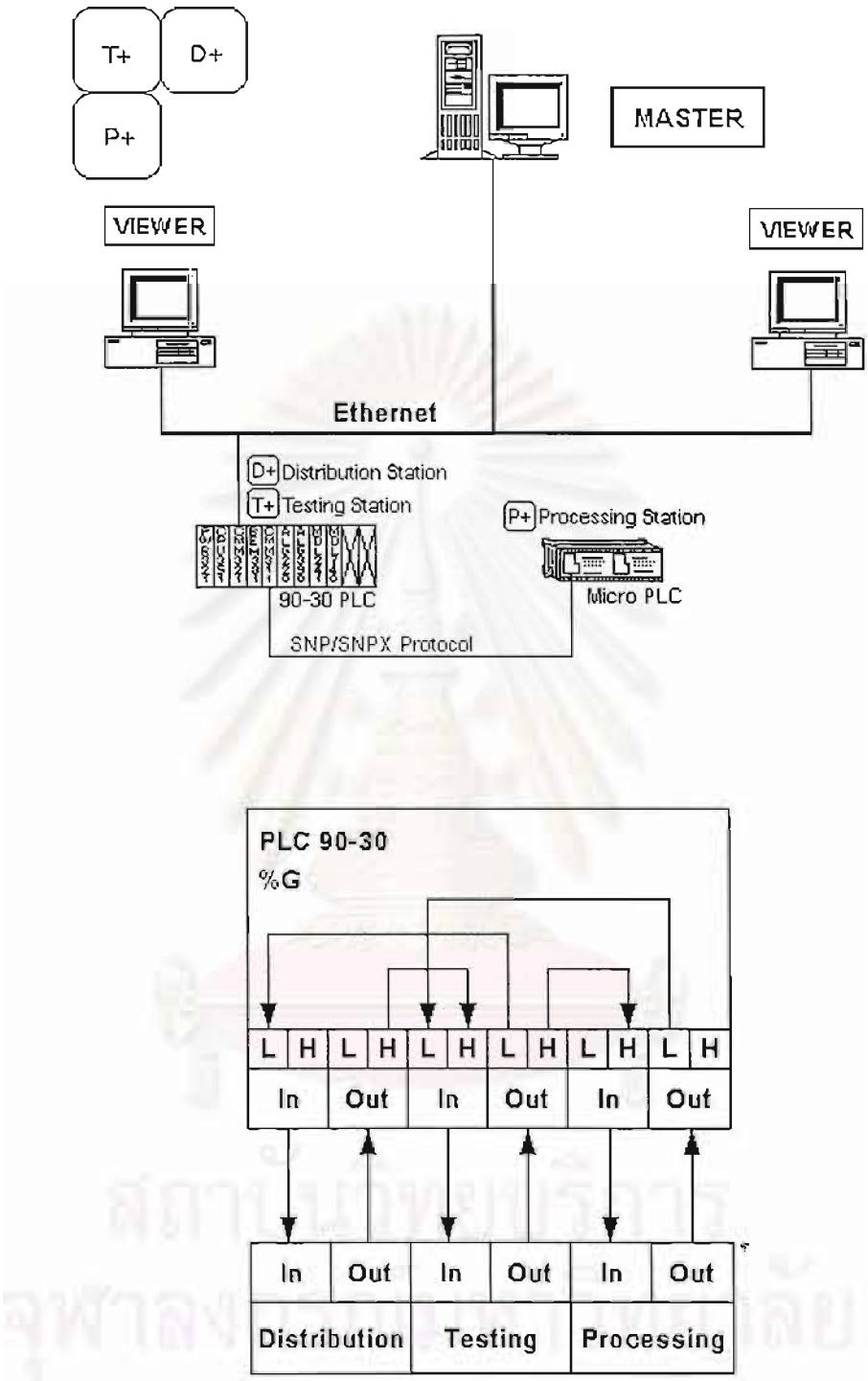
การทำงานจะเริ่มจากสถานีจ่ายชิ้นงาน ส่งชิ้นงานจากกระบอกใส่ชิ้นงาน เข้าไปยังถาดรับชิ้นงานในสถานีทดสอบชิ้นงาน ชิ้นงานจะถูกตรวจสอบความเป็นโลหะ, สี และความสูง และจะบันทึกข้อมูลชิ้นงานลงในบิต D0, D1 และ D2 เมื่อตรวจสอบเสร็จแล้วชิ้นงานจะถูกดันออกที่ด้านบนไหลเข้าสู่สถานีผลิตชิ้นงาน ซึ่งจะทำการเจาะรูบนชิ้นงานที่มีความสูงใช้ได้ รวมทั้งตรวจสอบรูเจาะบนชิ้นงาน และบันทึกข้อมูลลงในบิต D5 จากนั้นสถานีแขนกลจะรับชิ้นงานต่อ โดยให้แขนกลหยิบชิ้นงานจากสถานีผลิตชิ้นงานไปแยกใส่กระบอกใส่ชิ้นงาน ซึ่งจะมี 4 กระบอก คือ ชิ้นงานสีแดง, ชิ้นงานสีดำ, ชิ้นงานโลหะ และชิ้นงานเสีย (ความสูงหรือรูเจาะใช้ไม่ได้)

4. <u>สายการผลิตแบบ 4 โมดูล ต่อเป็นรูปตัว L</u> ประกอบไปด้วยสถานีจ่ายชิ้นงาน, สถานีทดสอบชิ้นงาน, สถานีพักชิ้นงาน และสถานีแยกชิ้นงาน ดังรูปที่ 7.8 เครือข่ายการสื่อสารที่ใช้และการรับ-ส่งข้อมูลระหว่างสถานีเป็นดังรูปที่ 7.9 และมีลักษณะการทำงานของสายการผลิตจากโปรแกรมที่ได้เขียนขึ้นมาเป็นดังนี้

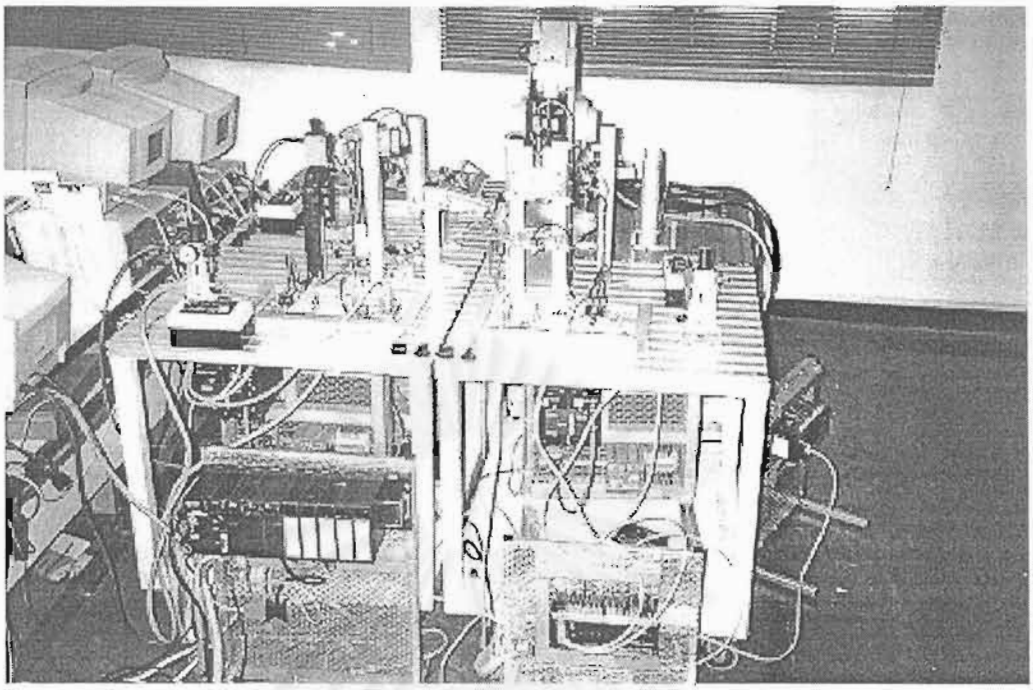
การทำงานจะเริ่มจากสถานีจ่ายชิ้นงาน ส่งชิ้นงานจากกระบอกใส่ชิ้นงานเข้า ไปยังถาดรับชิ้นงานในสถานีทดสอบชิ้นงาน ชิ้นงานจะถูกตรวจสอบความเป็นโลหะ, สี และความสูง และจะบันทึกข้อมูลชิ้นงานลงในบิต D0, D1 และ D2 เมื่อตรวจสอบเสร็จแล้ว ชิ้นงานจะถูกดันออกที่ด้านบนไหลเข้าสู่สถานีพักชิ้นงาน ซึ่งจะพักชิ้นงานไว้ชั่วคราวได้สูงสุด 2 ชิ้น จากนั้นชิ้นงานจะไหลต่อไปยังสถานีแยกชิ้นงาน ที่สถานีแยกชิ้นงานสายพานจะหมุน และเปิดตัวกั้นบังคับให้ชิ้นงานไหลลงในรางรับชิ้นงาน ตามชนิดและสีของชิ้นงาน โดยจะมีรางรับชิ้นงานทั้งหมด 4 ราง คือ ชิ้นงานสีแดง, ชิ้นงานสีดำ, ชิ้นงานโลหะ และชิ้นงานเสีย (ความสูงใช้ไม่ได้)



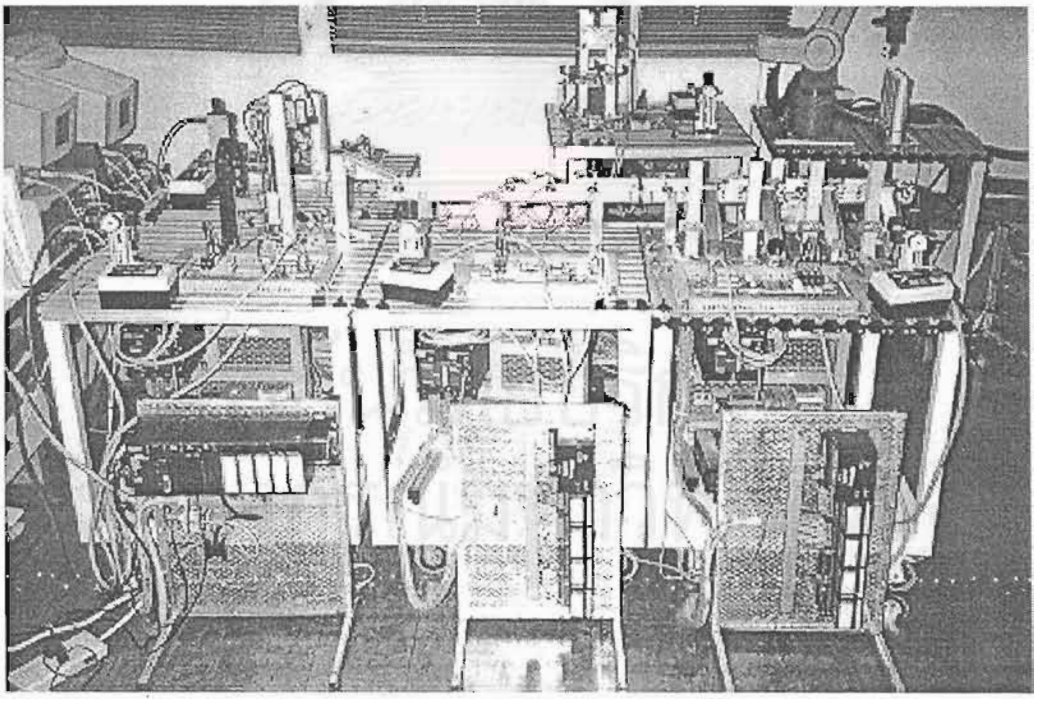
รูปที่ 7.4 แสดงสายการผลิตแบบ 3 โมดูล



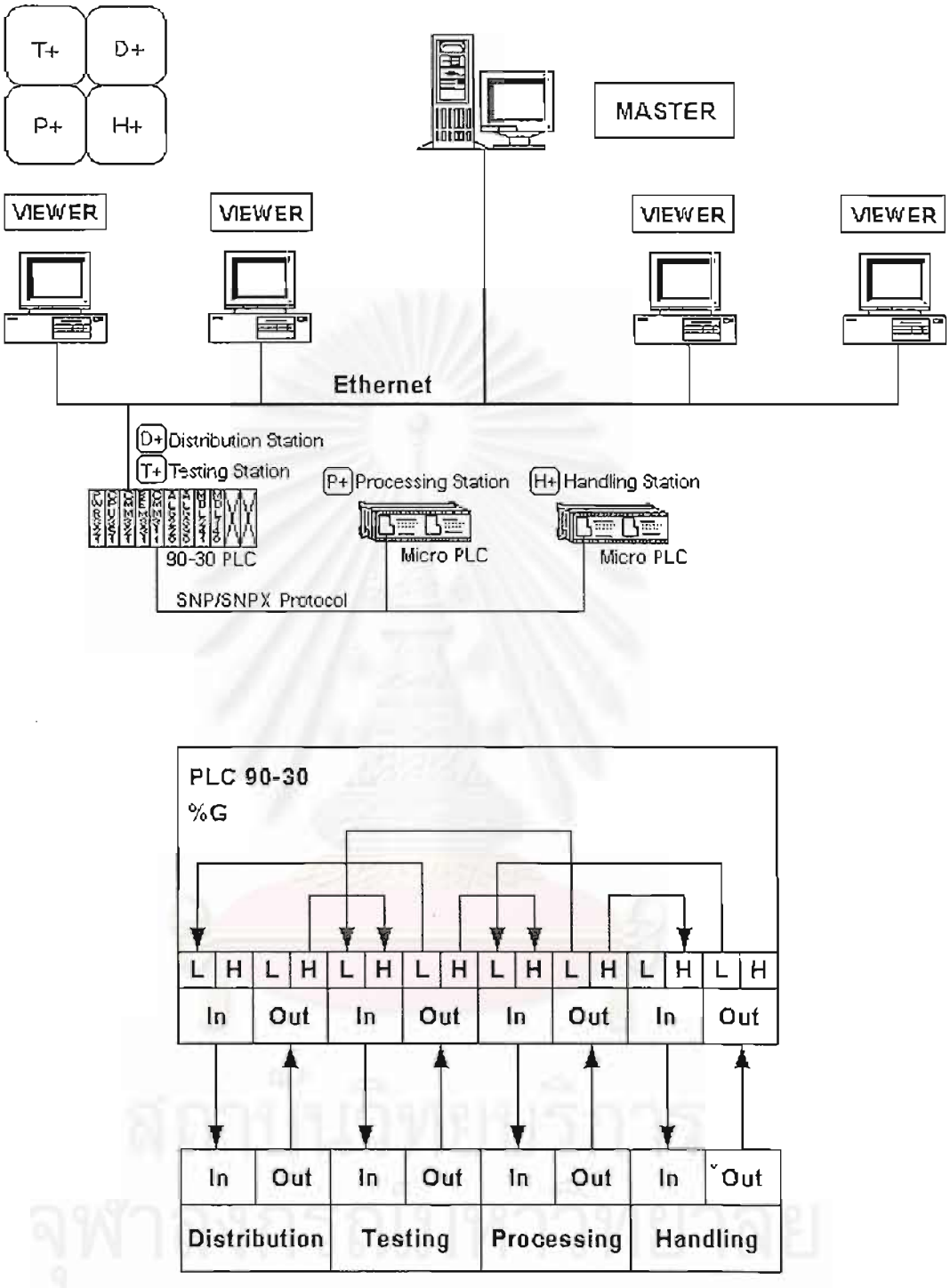
รูปที่ 7.5 แสดงเครือข่ายการสื่อสารและการรับ-ส่งข้อมูลระหว่างสถานีของสายการผลิตแบบ 3 โมดูล



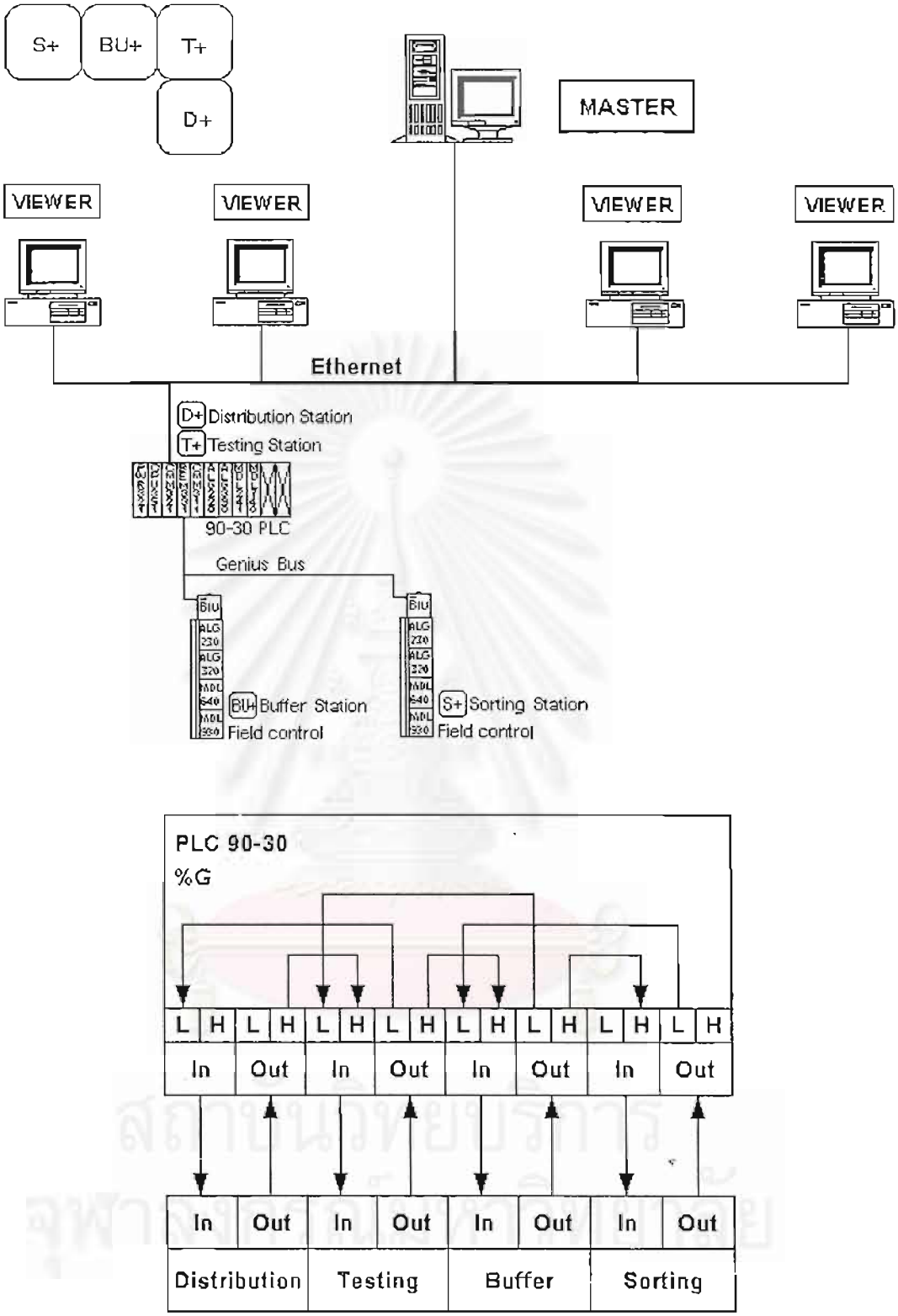
รูปที่ 7.6 แสดงสายการผลิตแบบ 4 โมดูลต่อเป็นรูปตัว O



รูปที่ 7.8 แสดงสายการผลิตแบบ 4 โมดูล ต่อเป็นรูปตัว L



รูปที่ 7.7 แสดงเครือข่ายการสื่อสารและการรับ-ส่งข้อมูลระหว่างสถานีของสายการผลิตแบบ 4 โมดูลต่อเป็นรูปตัว O



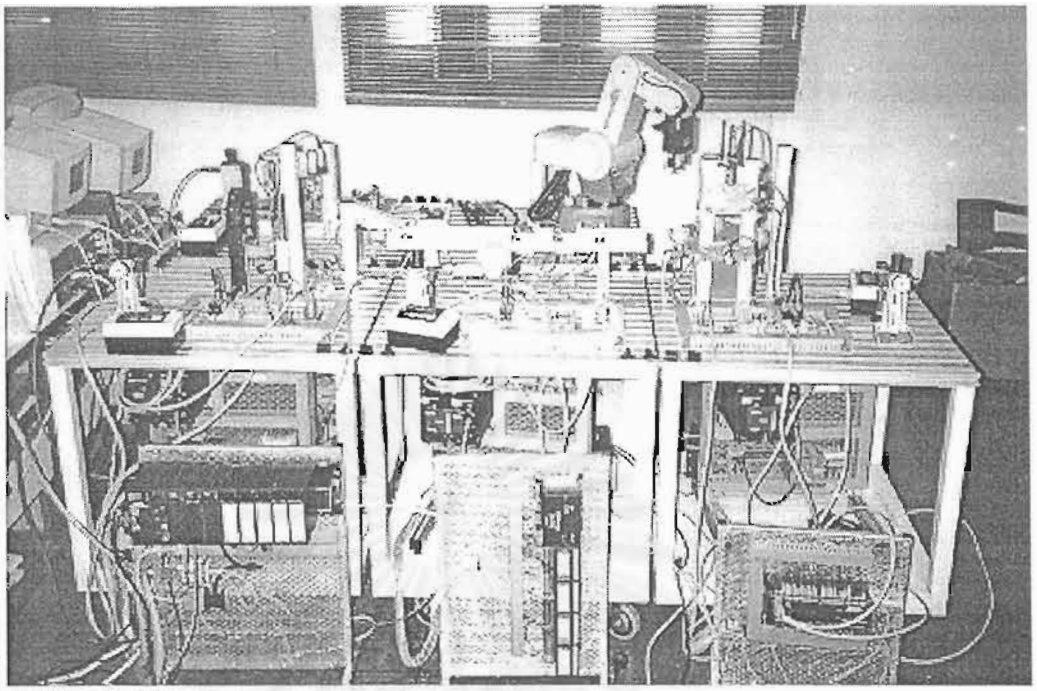
รูปที่ 7.9 แสดงเครือข่ายการสื่อสารและการรับ-ส่งข้อมูลระหว่างสถานีของสายการผลิตแบบ 4 โมดูล ต่อเป็นรูปตัว L

5. สายการผลิตแบบ 5 โมดูล ประกอบไปด้วยสถานีจ่ายชิ้นงาน, สถานีทดสอบชิ้นงาน สถานีพักชิ้นงาน, สถานีผลิตชิ้นงาน และสถานีแขนกล ดังรูปที่ 7.10 เครือข่ายการสื่อสารที่ใช้และการรับ-ส่งข้อมูลระหว่างสถานีเป็นดังรูปที่ 7.11 และมีลักษณะการทำงานของสายการผลิตจากโปรแกรมที่ได้เขียนขึ้นมาเป็นดังนี้

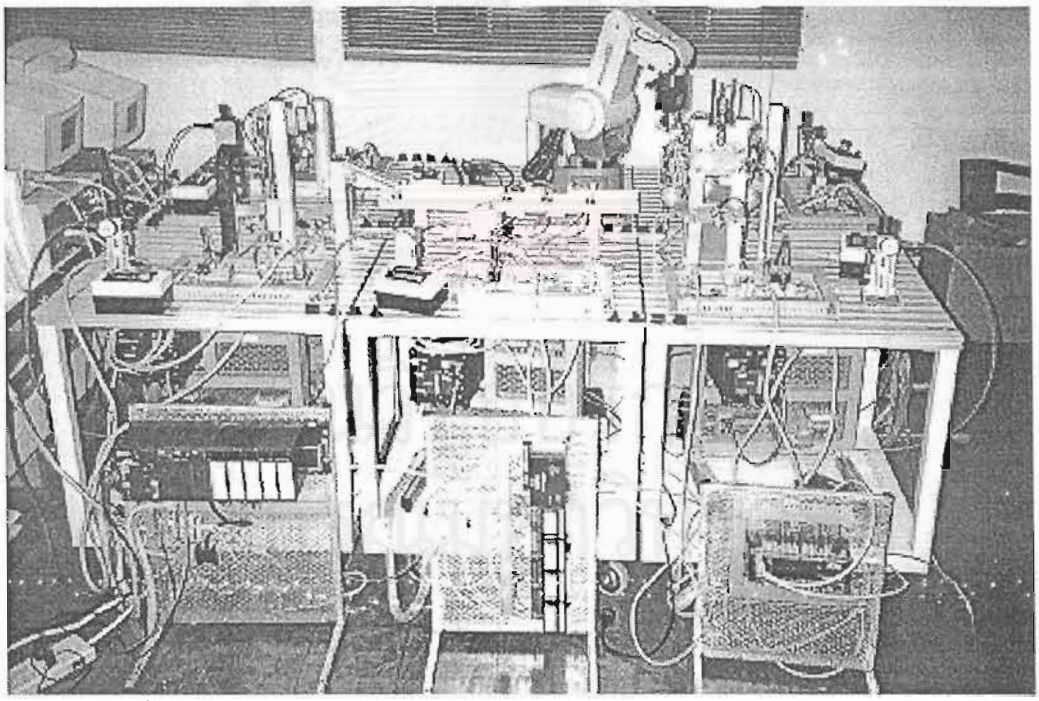
การทำงานจะเริ่มจากสถานีจ่ายชิ้นงาน ส่งชิ้นงานจากกระบอกใส่ชิ้นงานเข้า ไปยังถาดรับชิ้นงานในสถานีทดสอบชิ้นงาน ชิ้นงานจะถูกตรวจสอบความเป็นโลหะ, สี และความสูง และจะบันทึกข้อมูลชิ้นงานลงในบิต D0, D1 และ D2 เมื่อตรวจสอบเสร็จแล้ว ชิ้นงานจะถูกดันออกที่ด้านบนไหลเข้าสู่สถานีพักชิ้นงาน ซึ่งจะพักชิ้นงานไว้ชั่วคราวได้สูงสุด 3 ชิ้น และเมื่อสถานีผลิตชิ้นงานพร้อมรับชิ้นงาน ก็จะส่งบิตข้อมูลร้องขอให้แขนกลทำการหยิบชิ้นงานจากสถานีพักชิ้นงานมายังสถานีผลิตชิ้นงาน ซึ่งจะทำการเจาะรูบนชิ้นงานที่มีความสูงใช้ได้ รวมทั้งตรวจสอบรูเจาะบนชิ้นงาน และบันทึกข้อมูลชิ้นงานลงในบิต D5 จากนั้นสถานีแขนกลจะรับชิ้นงานต่อโดยให้แขนกลหยิบชิ้นงานจากสถานีผลิตชิ้นงาน ไปแยกใส่กระบอกใส่ชิ้นงาน ซึ่งจะมี 4 กระบอก คือ ชิ้นงานสีแดง, ชิ้นงานสีดำ, ชิ้นงานโลหะ และชิ้นงานเสีย (ความสูงหรือรูเจาะใช้ไม่ได้)

6. สายการผลิตแบบ 6 โมดูล ประกอบไปด้วยสถานีจ่ายชิ้นงาน, สถานีทดสอบชิ้นงาน สถานีพักชิ้นงาน, สถานีผลิตชิ้นงาน, สถานีแขนกล และสถานีแยกชิ้นงาน ดังรูปที่ 7.12 เครือข่ายการสื่อสารที่ใช้และการรับ-ส่งข้อมูลระหว่างสถานีเป็นดังรูปที่ 7.13 และมีลักษณะการทำงานของสายการผลิตจากโปรแกรมที่ได้เขียนขึ้นมาเป็นดังนี้

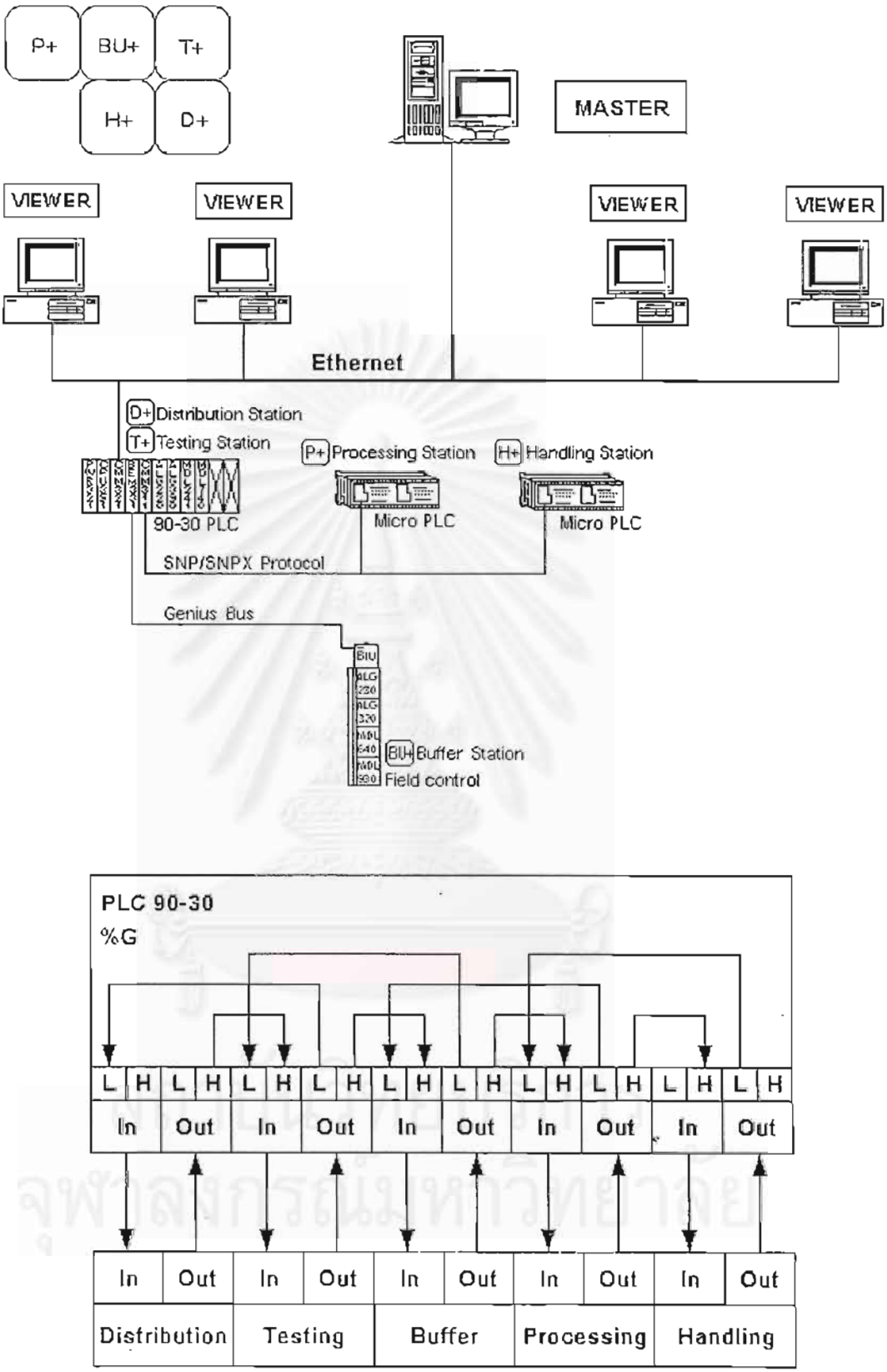
การทำงานจะเริ่มจากสถานีจ่ายชิ้นงาน ส่งชิ้นงานจากกระบอกใส่ชิ้นงานเข้า ไปยังถาดรับชิ้นงานในสถานีทดสอบชิ้นงาน ชิ้นงานจะถูกตรวจสอบความเป็นโลหะ, สี และความสูง และจะบันทึกข้อมูลชิ้นงานลงในบิต D0, D1 และ D2 เมื่อตรวจสอบเสร็จแล้วชิ้นงานจะถูกดันออกที่ด้านบนไหลเข้าสู่สถานีพักชิ้นงาน ซึ่งจะพักชิ้นงานไว้ชั่วคราวได้สูงสุด 3 ชิ้น และเมื่อสถานีผลิตชิ้นงานพร้อมรับชิ้นงาน ก็จะส่งบิตข้อมูลร้องขอให้แขนกลทำการหยิบชิ้นงาน จากสถานีพักชิ้นงานมายังสถานีผลิตชิ้นงาน ซึ่งจะทำการเจาะรูบนชิ้นงานที่มีความสูงใช้ได้ รวมทั้งตรวจสอบรูเจาะบนชิ้นงาน และบันทึกข้อมูลชิ้นงานลงในบิต D5 จากนั้นสถานีแขนกลจะรับชิ้นงานต่อ โดยให้แขนกลหยิบชิ้นงานจากสถานีผลิตชิ้นงาน ในกรณีที่ชิ้นงานไม่มีรูเจาะ แขนกลจะนำชิ้นงานไปแยกใส่กระบอกใส่ชิ้นงาน ซึ่งจะมี 4 กระบอก คือ ชิ้นงานสีแดง, ชิ้นงานสีดำ, ชิ้นงานโลหะ และชิ้นงานเสีย (ความสูงใช้ไม่ได้) ส่วนในกรณีที่ชิ้นงานมีรูเจาะ แขนกลจะนำชิ้นงานไปวางที่สถานีแยกชิ้นงาน จากนั้นที่สถานีแยกชิ้นงานสายพานจะหมุน และเปิดตัวกั้นบังคับ ให้ชิ้นงานไหลลงในรางรับชิ้นงาน ตามชนิดและสีของชิ้นงาน ซึ่งจะมีรางรับชิ้นงานทั้งหมด 4 ราง คือ ชิ้นงานสีแดง, ชิ้นงานสีดำ, ชิ้นงานโลหะ และชิ้นงานเสีย (ความสูงใช้ไม่ได้)



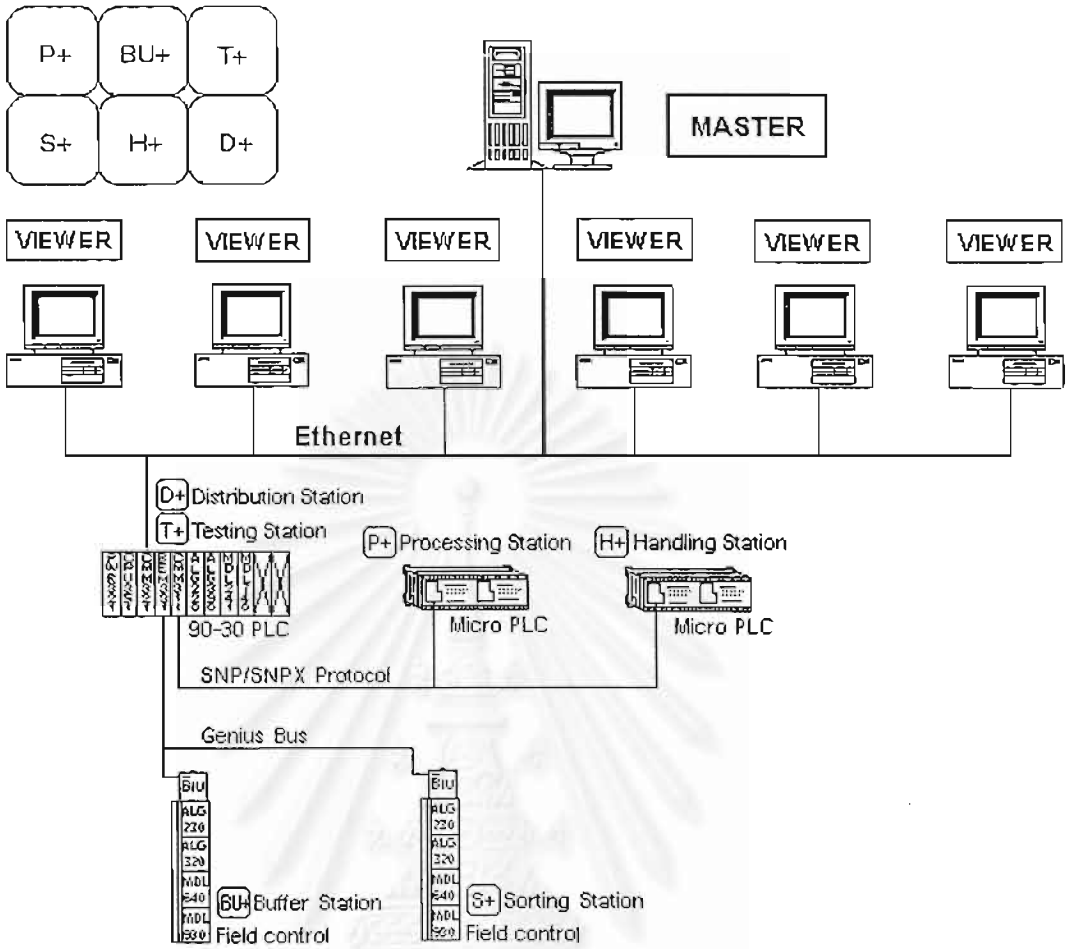
รูปที่ 7.10 แสดงสายการผลิตแบบ 5 โมดูล

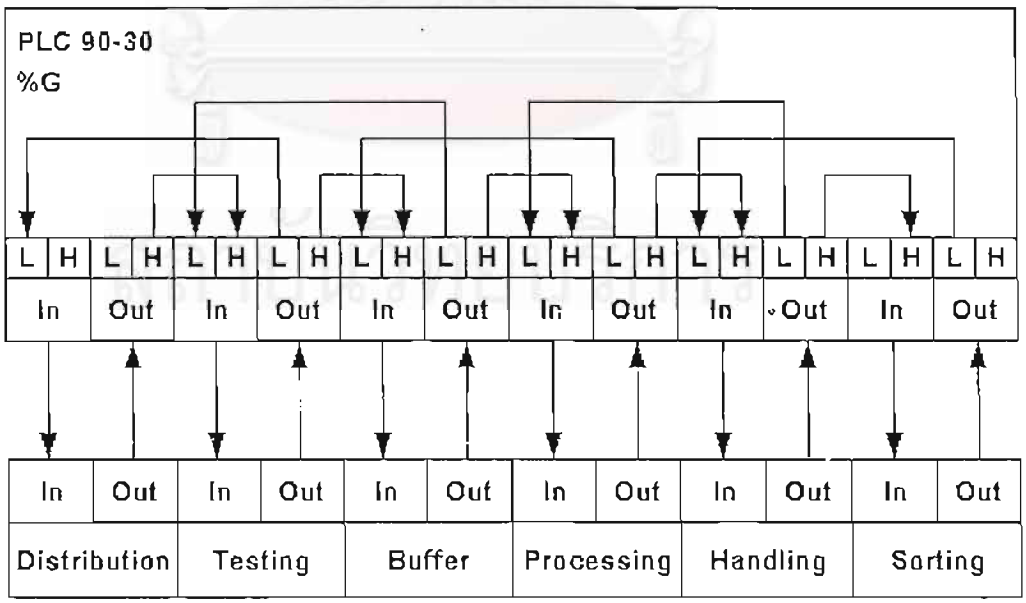


รูปที่ 7.12 แสดงสายการผลิตแบบ 6 โมดูล



รูปที่ 7.11 แสดงเครือข่ายการสื่อสารและการรับ-ส่งข้อมูลระหว่างสถานีของสายการผลิตแบบ 5 โมดูล





รูปที่ 7.13 แสดงเครือข่ายการสื่อสารและการรับ-ส่งข้อมูลระหว่างสถานีของสายการผลิตแบบ 6 โมดูล

บทที่ 8
สรุปผลและข้อเสนอแนะ

สรุปผลการทำวิทยานิพนธ์

วิทยานิพนธ์ฉบับนี้แสดงให้เห็นถึง การนำเอาระบบเครือข่ายการควบคุมมาใช้กับระบบการผลิตอัตโนมัติแบบโมดูล ที่สามารถปรับเปลี่ยนรูปแบบของสายการผลิตได้หลากหลายรูปแบบ ระบบเครือข่ายที่นำมาใช้จะครอบคลุมทุกระดับของการสื่อสาร ตั้งแต่ระดับล่างสุด (เครือข่าย I/O) ไปจนถึงระดับบนสุด (เครือข่ายโรงงาน) ทำให้สามารถควบคุมสายการผลิต และเก็บรวบรวมข้อมูลการผลิต เพื่อใช้ในการวางแผนการผลิตต่อไปได้ โดยมีรายละเอียดเป็นขั้นตอนดังนี้

1) ขั้นตอนการศึกษาการทำงาน และวางแผนการจัดการเครือข่ายการควบคุม เนื่องจากเรามีระบบการผลิตแบบโมดูลอยู่ 7 โมดูล ซึ่งมีหน้าที่การทำงานต่าง ๆ กัน และมีอุปกรณ์ควบคุมคือ PLC, I/O Block ขนาดต่าง ๆ กัน ดังนั้นในการวิจัยจะแสดงให้เห็นถึงการเลือกใช้อุปกรณ์ควบคุมที่มีอยู่ให้เหมาะสมกับโมดูล ตามลักษณะการทำงานและจำนวน I/O ในโมดูลนั้น ๆ รวมไปถึงการวางแผนการจัดการเครือข่ายการควบคุม สำหรับอุปกรณ์ควบคุมที่อยู่ในโมดูลต่าง ๆ ให้สามารถรองรับการปรับเปลี่ยนรูปแบบของสายการผลิต

2) ขั้นตอนการเขียนโปรแกรมการทำงาน โปรแกรมสำหรับการสื่อสาร และโปรแกรมสำหรับตรวจดูระบบการผลิต ขั้นตอนนี้แสดงให้เห็นถึงการเขียนโปรแกรมการทำงานสำหรับโมดูลต่าง ๆ และโปรแกรมสำหรับการสื่อสารข้อมูลระหว่างโมดูล โดยใช้ภาษาแลดเดอร์ในการเขียนโปรแกรมซึ่งเป็นภาษาพื้นฐานที่เข้าใจได้ง่าย ในส่วนของโปรแกรมสำหรับตรวจดูระบบการผลิต ได้ใช้ซอฟต์แวร์ CIMPLICITY HMI ซึ่งเป็นซอฟต์แวร์ที่ใช้งานได้ง่าย และทำให้สามารถแสดงรูปภาพเคลื่อนไหวแทนการทำงานในโมดูลนั้นได้

3) ขั้นตอนการทดลองปรับเปลี่ยนรูปแบบสายการผลิต ในขั้นตอนนี้แสดงให้เห็นถึงความยืดหยุ่น (Flexibility) ของเครือข่ายการควบคุมและสายการผลิตแบบโมดูล โดยในการวิจัยได้ปรับเปลี่ยนรูปแบบของสายการผลิตไปทั้งหมด 7 รูปแบบ ซึ่งการปรับเปลี่ยนในแต่ละรูปแบบสามารถทำได้ง่ายและรวดเร็ว และสามารถทำงานได้ตามลักษณะของสายการผลิตนั้น

ข้อเสนอแนะ

1) เนื่องจากอุปกรณ์ควบคุมที่มีอยู่ มีหลายขนาด ทำให้ในบางโมดูลมีการใช้อุปกรณ์ควบคุมที่มีขนาดความสามารถเกินความต้องการในโมดูลนั้น ๆ

2) ในการเขียนโปรแกรมใน PLC ด้วยภาษาแลดเดอร์ สำหรับโปรแกรมที่มีความซับซ้อน เช่น โปรแกรมการทำงานในโมดูลต่าง ๆ ทำให้ต้องเขียนโปรแกรมที่ยาวมาก ซึ่งการตรวจสอบโปรแกรมทำได้ยาก ซึ่งถ้าใช้ภาษาที่สูงกว่า จะทำให้สามารถตรวจสอบโปรแกรมได้ง่ายขึ้น

3) ในส่วนของโปรแกรมตรวจดูระบบการผลิต สามารถพัฒนาเพิ่มเติมให้สามารถควบคุมการทำงานของโมดูลผ่านทางหน้าจอคอมพิวเตอร์ เช่น ในส่วนหน้าจอของกล่องควบคุมสามารถสามารถพัฒนา ให้สามารถกดปุ่มเพื่อควบคุมการทำงานของโมดูลนั้น ได้จากบนหน้าจอ

4) ในส่วนของเครือข่ายระดับโรงงาน ที่ใช้เครือข่ายอีเทอร์เน็ต และโปรโตคอล TCP/IP ถ้าสามารถเชื่อมต่อเครือข่ายของโรงงาน เข้ากับเครือข่ายสาธารณะ จะทำให้สามารถตรวจดูสายการผลิต ได้จากทุกที่ทั่วโลก หรืออาจจะสร้างเป็นเวบเพจ (Web Page) แสดงการทำงานของสายการผลิต

รายการอ้างอิง

ภาษาไทย

กองบรรณาธิการ. ระบบควบคุมอัตโนมัติ. Industrial Technology Review ฉบับที่ 66 (มกราคม 2543) : 60-63.

ปานเพชร ชินินทร, ขวัญชัย สินทิพย์สมบูรณ์. นิวแมติกอุตสาหกรรม. กรุงเทพมหานคร : ซีเอ็ดยูเคชั่น, 2541.

พรจิต ประทุมสุวรรณ และคณะ. PROGRAMMABLE CONTROLLER. กรุงเทพมหานคร : โรงพิมพ์เรือนแก้วการพิมพ์, 2536.

พูลพงษ์ บุญพราหมณ์. คอมพิวเตอร์ช่วยงานอุตสาหกรรม. พิมพ์ครั้งที่ 3. กรุงเทพมหานคร : สมาคมสงเสริมเทคโนโลยี (ไทย-ญี่ปุ่น), 2537.

ภาสิต อนุกุลอนันต์ชัย. การพัฒนาโพสต์โปรเซสเซอร์สำหรับระบบผลิตอัตโนมัติ. วิทยานิพนธ์ปริญญามหาบัณฑิต สาขาวิชาวิศวกรรมเครื่องกล ภาควิชาวิศวกรรมเครื่องกล บัณฑิตวิทยาลัย จุฬาลงกรณ์มหาวิทยาลัย.

วาทิต เบญจพลกุล. การสื่อสารข้อมูล. พิมพ์ครั้งที่ 1. กรุงเทพมหานคร : สัมฤทธิ์การพิมพ์, 2541.

สุภัทสรณ์ จุดิสงขลา. Fieldbus เทคโนโลยีระบบควบคุมในศตวรรษที่ 21. Industrial Technology Review ฉบับที่ 45 (พฤษภาคม 2541) : 120-124.

สุเธียร เกียรติสุนทร. ดีซีเอส. พิมพ์ครั้งที่ 1. กรุงเทพมหานคร : สมาคมสงเสริมเทคโนโลยี (ไทย-ญี่ปุ่น), 2542.

ภาษาอังกฤษ

Festo Didactic. Communication at field level Profibus. August 1995.

Fieldbus Forum. http://hallen.ele.kth.se/~willi/Fieldbus

GE Fanuc Automation. CIMPLICITY® HMI for Windows NT® and Windows 95® Base System User's Manual. GFK-1180D November 1996.

GE Fanuc Automation. Genius™ I/O System and Communications User's Manual. GEK-90486F-1 November 1994.

GE Fanuc Automation. Series 90™ PLC Serial Communication User's Manual. GEK-0582B May 1993.

Technical Overview Profibus. http://www.profibus.com

ภาคผนวก

ภาคผนวก ก

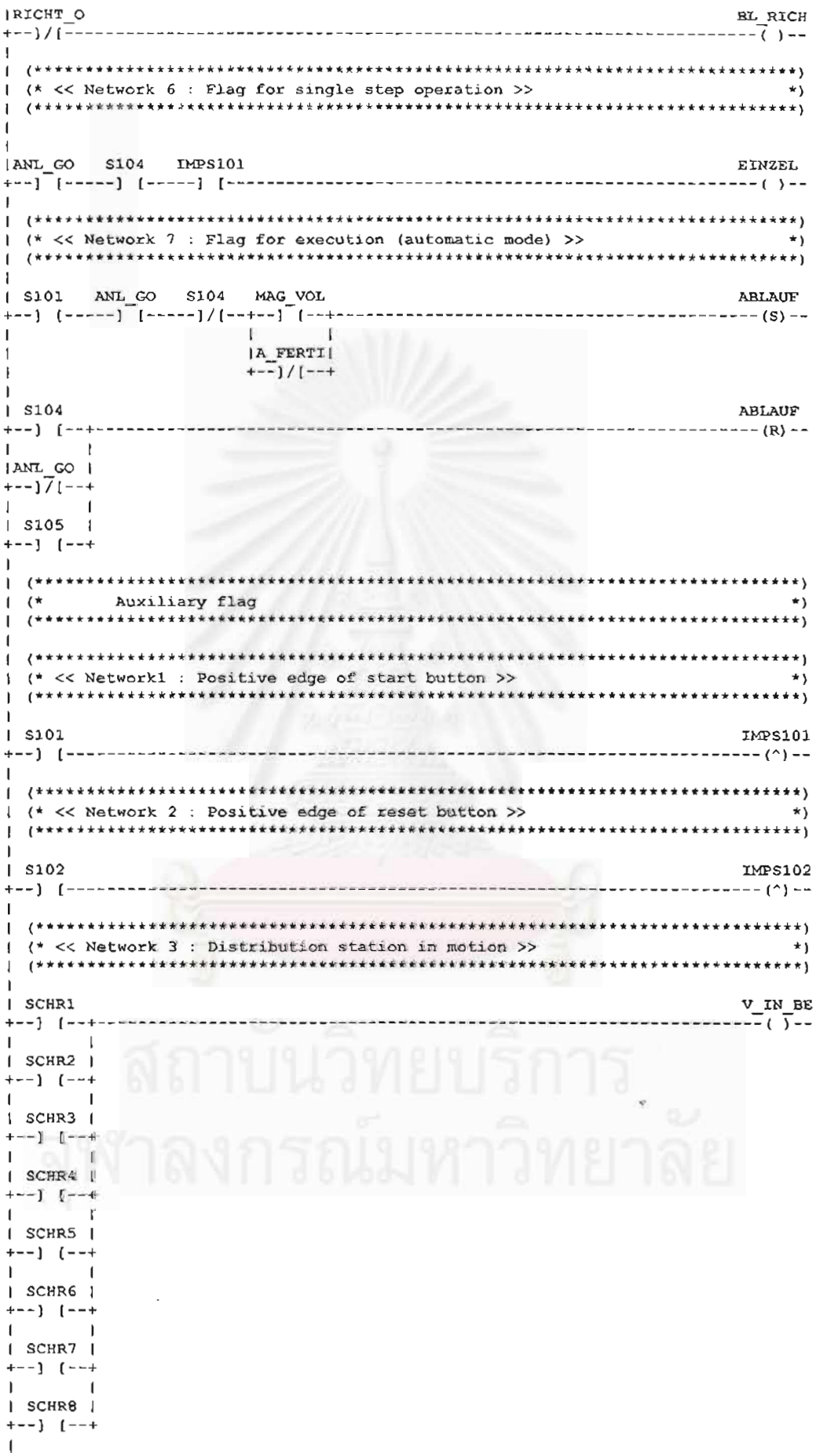
โปรแกรมการทำงานของสถานีจ่ายชิ้นงาน

```
||  START LD SUBROUTINE  DISTRIB  )
|
||     VARIABLE DECLARATIONS      )
```

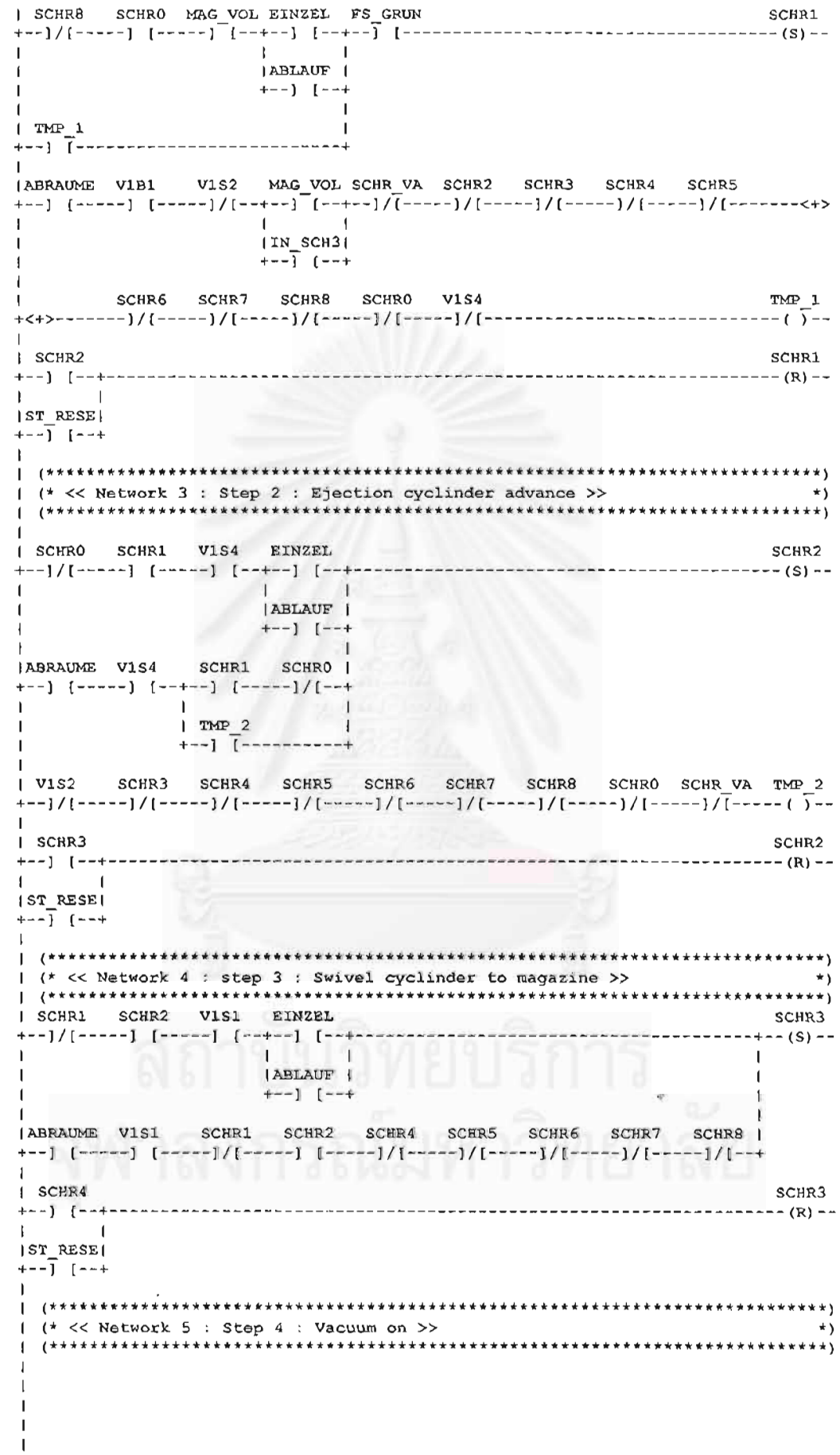
V A R I A B L E D E C L A R A T I O N T A B L E

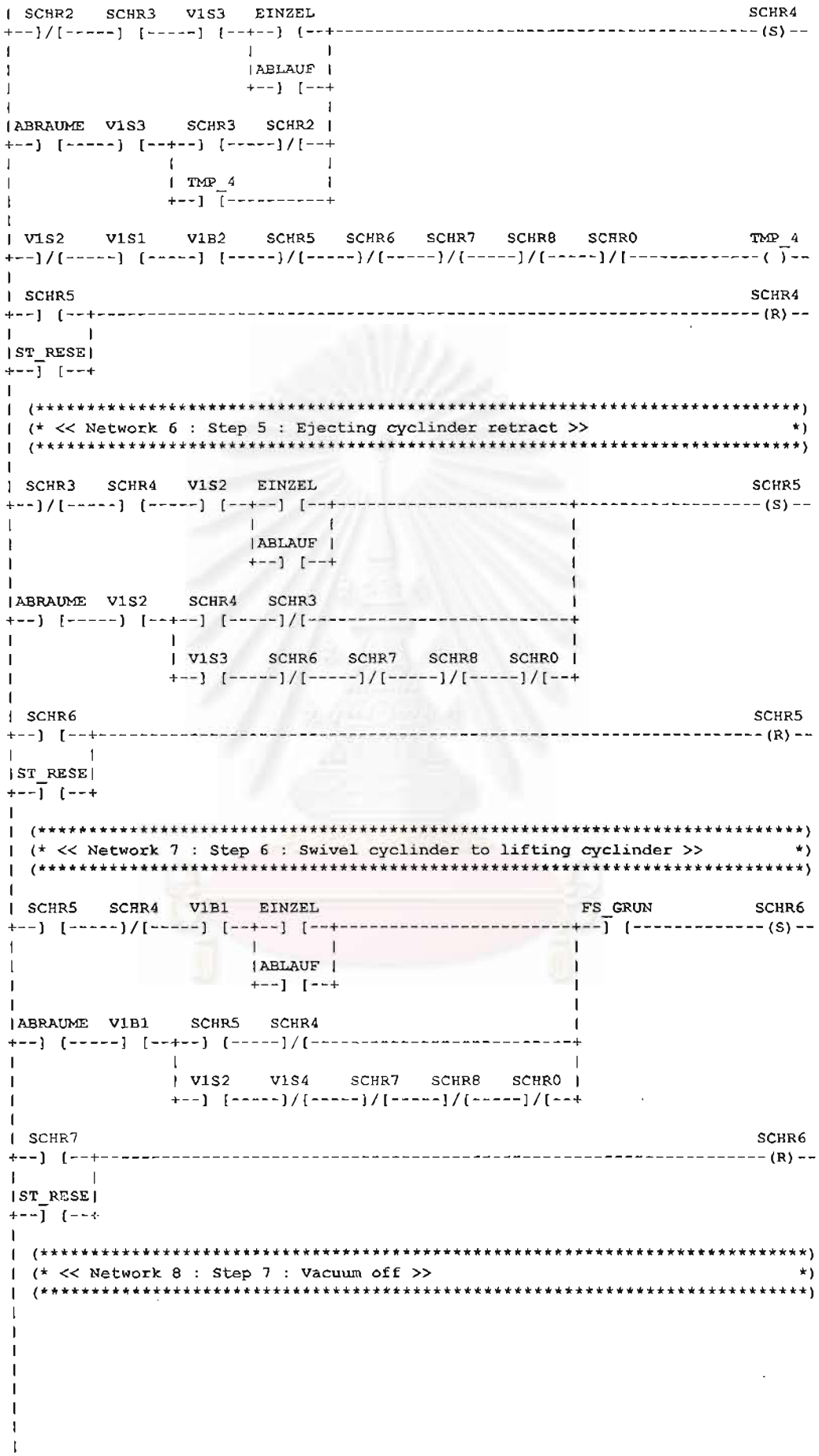
REFERENCE	NICKNAME	REFERENCE DESCRIPTION
%I0001	V1B1	Ejecting cyclinder retracted
%I0002	V1B2	Ejecting cyclinder extended
%I0003	V1S3	Swivel cyclinder at magazine
%I0004	V1S4	Swivel cyclinder at lift. cyc.
%I0005	V1S1	Workpiece ejected
%I0006	V1S2	Workpiece retained
%I0007	V1B3	Part in magazine
%I0009	S101	Start Button
%I0010	S102	Reset button
%I0011	S103	reset magazine empty
%I0012	S104	Sequence/Stop
%I0013	S105	Stop at end of cycle
%I0016	NOTAUS	Emergency-Stop function
%Q0001	V1Y1	Ejecting cyclinder advance
%Q0002	V1Y5	Swivel cyclinder to magazine
%Q0003	V1Y4	Swivel cyclinder to lifting cyc.
%Q0004	V1Y3	Vacuum off
%Q0005	V1Y2	Vacuum on
%Q0006	AUTO_LA	Light for station in auto mode
%Q0007	LA_GRST	Light fir station in init. pos.
%Q0008	LAMPE_M	Light for magazine empty
%Q0009	VERNET2	Light for station in networked
%M0001	ST_RESE	Reset ststion
%M0002	EINZEL	Single step
%M0003	ABLAUF	Flag for sequence
%M0004	A_FERTI	Distribution station ready
%M0005	R_FERTI	Initial position after reset
%M0006	ABRAUME	Clear starion
%M0007	ANL_GO	Starting of installation
%M0008	BL_RICH	Reset light flashing
%M0009	V_IN_BE	Distribution station in motion
%M0010	R_VAKUU	Is part retained during reset?
%M0011	MAG_VOL	Magazine is full
%M0012	RICHT_O	Station being reset
%M0101	IMPS101	Pulse flag of start button
%M0103	IMPS102	Pulse flag of reset button
%M0104	SCHR_VA	Flag whether step 4 was active
%M0105	IN_SCH3	Flag whether step 3 was active
%M0107	MERK_VE	Operation of station networked
%M0200	SCHR0	Initialising step
%M0201	SCHR1	Swivel cyclinder to lifting cycr
%M0202	SCHR2	Ejecting cyclinder advance
%M0203	SCHR3	Swivel cyclinder to magazine
%M0204	SCHR4	Vacuum on
%M0205	SCHR5	Ejecting cyclinder retract
%M0206	SCHR6	Swivel cyclinder to lifting cycr
%M0207	SCHR7	Vacuum off
%M0208	SCHR8	Swivel cyclinder to magazine
%M0300	FS_GRUN	Subsequent station is in initian
%M0500	BLINK2H	Flashing pulse 2 Hz
%M0503	BLINKRE	Reset of flashing counter
%G0001	I_S_RDY	Subseq. station ready
%G0002	I_P_RDY	Subseq. station ready for p.
%G0003	I_D_REQ	Data Request
%G0009	I_D0	Data bit 0
%G0010	I_D1	Data bit 1
%G0011	I_D2	Data bit 2
%G0012	I_D3	Data bit 3
%G0013	I_D4	Data bit 4
%G0014	I_D5	Data bit 5

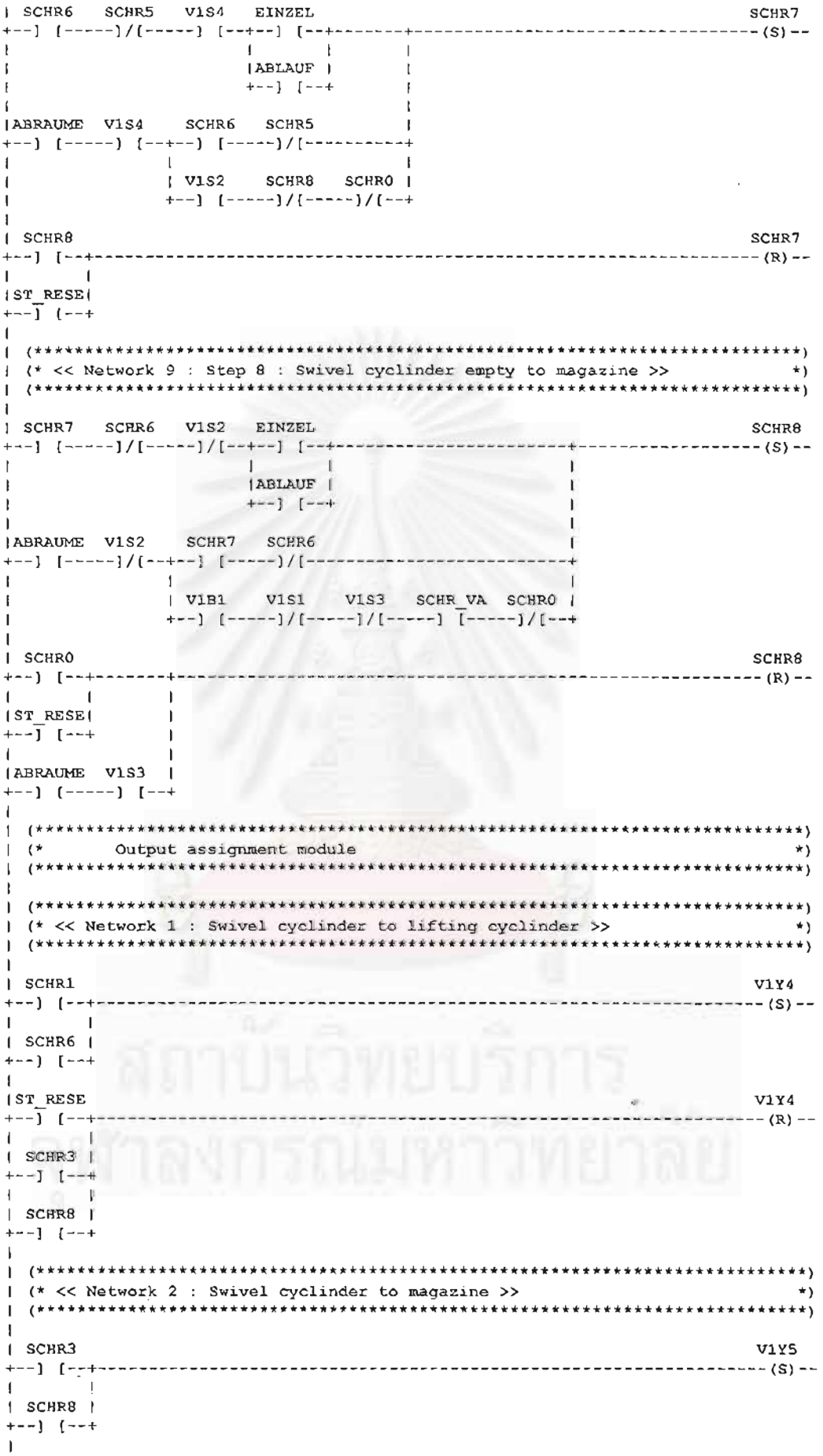
```
          %G0015        I_HS          Handling slave request
          %G0016        I_EN          Data/ Part ready
          %G0017        O_S_RDY
          %G0018        O_P_RDY
          %G0019        O_D_REQ
          %G0025        O_D0
          %G0026        O_D1
          %G0027        O_D2
          %G0028        O_D3
          %G0029        O_D4
          %G0030        O_D5
          %G0031        O_HS
          %G0032        O_EN
          %T0001        TMP_1         Temp. flag 1
          %T0002        TMP_2         Temp. flag 2
          %T0004        TMP_4         Temp. flag 4
          %T0010        TMP_B2H       Temp. flag for BLINK2H
          %R0001        R_TMP1        Temp. register for BLINK2H
          %R0011        R_TMP2        Temp. register for BLINK2H

|[   START OF SUBROUTINE LOGIC    ]
|
| (*****************************************************************************)
| (*      MODE OF OPERATION                                                    *)
| (*****************************************************************************)
|
| (*****************************************************************************)
| (* << Network 1 : Reset_OK is set, if station is reset >>                    *)
| (*****************************************************************************)
|
|R_FERTI                                                              RICHT_O
+--] [-------------------------------------------------------------------(S)--
|
|IMPS102                                                              RICHT_O
+--] [--+----------------------------------------------------------------(R)--
|       |
|NOTAUS |
+--]/[--+
|
| (*****************************************************************************)
| (* << Network 2 : Reset flag for station >>                                  *)
| (*****************************************************************************)
|
|RICHT_O                                                              ST_RESE
+--]/[-------------------------------------------------------------------(S)--
|
|ABRAUME                                                              ST_RESE
+--] [-------------------------------------------------------------------(R)--
|
| (*****************************************************************************)
| (* << Network 3 : Clear station during reset >>                             *)
| (*****************************************************************************)
|
| S102   RICHT_O                                                      ABRAUME
+--] [-----]/[-----------------------------------------------------------(S)--
|
|R_FERTI                                                              ABRAUME
+--] [--+----------------------------------------------------------------(R)--
|       |
|NOTAUS |
+--]/[--+
|
| (*****************************************************************************)
| (* << Network 4 : Station is reset and can be started >>                     *)
| (*****************************************************************************)
|
|ABRAUME RICHT_O                                                       ANL_GO
+--]/[-----] [-----------------------------------------------------------(S)--
|
|RICHT_O                                                               ANL_GO
+--]/[-------------------------------------------------------------------(R)--
|
| (*****************************************************************************)
| (* << Network 5 : Flag for flashing of initial position light >>            *)
| (*****************************************************************************)
|
```

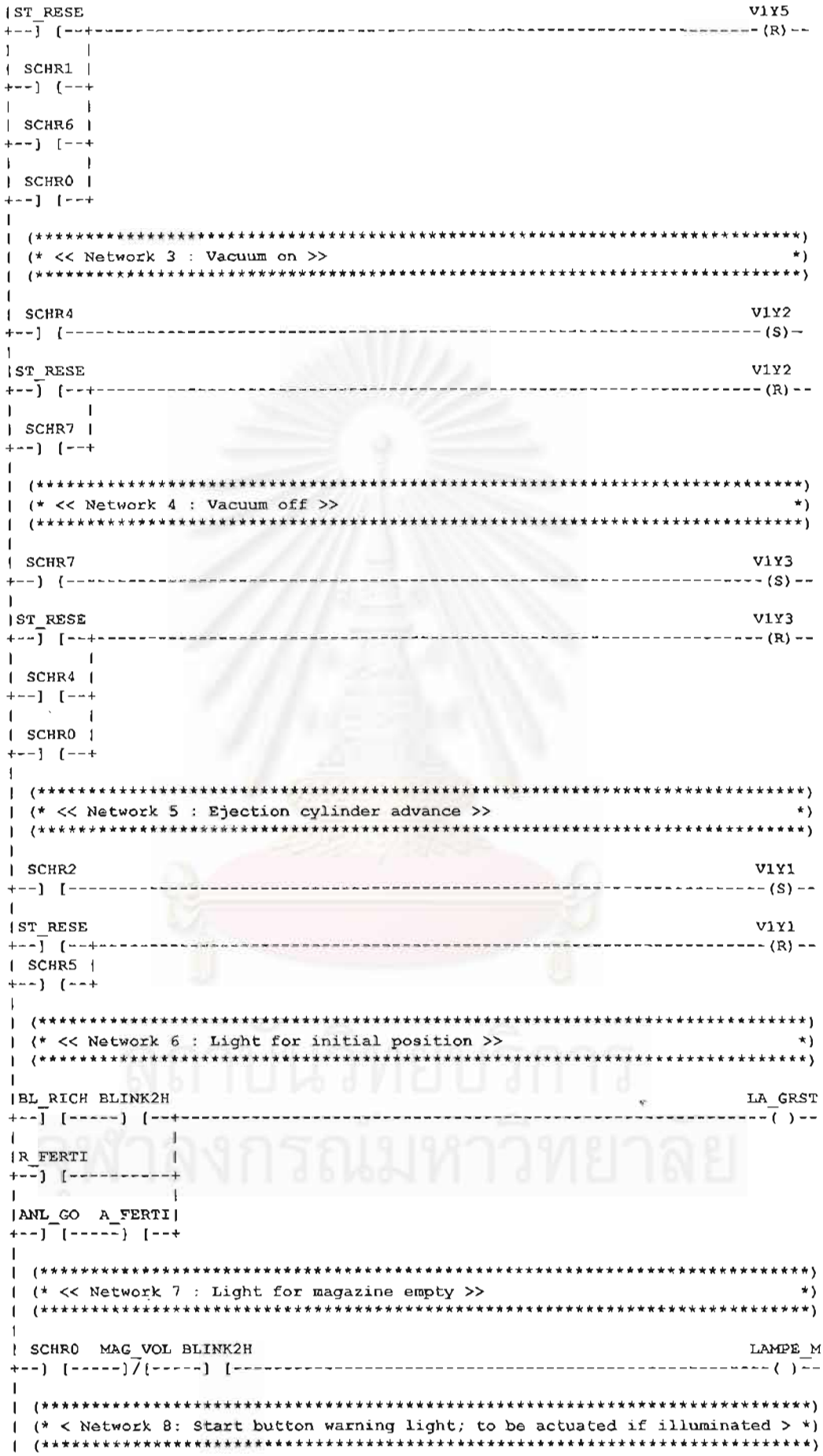

RICHT_O
BL_RICH
(* << Network 6 : Flag for single step operation >> *)
ANL_GO
S104
IMPS101
EINZEL
(* << Network 7 : Flag for execution (automatic mode) >> *)
S101
ANL_GO
S104
MAG_VOL
A_FERTI
ABLAUF
S104
ABLAUF
ANL_GO
S105
(* Auxiliary flag *)
(* << Network1 : Positive edge of start button >> *)
S101
IMPS101
(* << Network 2 : Positive edge of reset button >> *)
S102
IMPS102
(* << Network 3 : Distribution station in motion >> *)
SCHR1
V_IN_BE
SCHR2
SCHR3
SCHR4
SCHR5
SCHR6
SCHR7
SCHR8

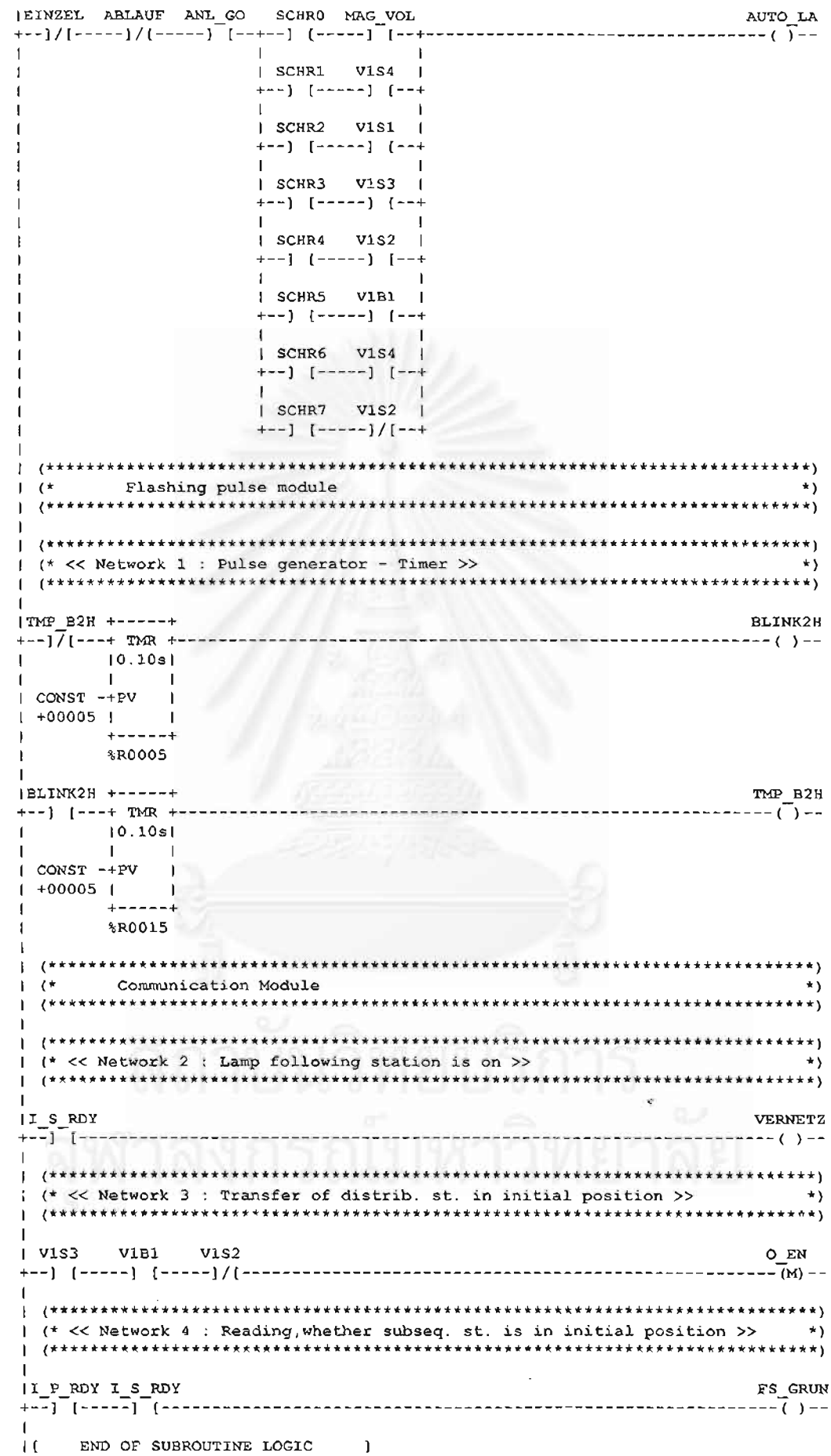
```
|  (*************************************************************************)
|  (* << Network 4 : Distribution station execution completed >>            *)
|  (*************************************************************************)
|
|  V1B1     V1S1     V1S3     V1S2    V_IN_BE                          A_FERTI
+--] [-----]/[-----] [-----]/[-----]/[-----------------------------------( )--
|
|
|  (*************************************************************************)
|  (* << Network 5 : Reset of distribution station ready >>                 *)
|  (*************************************************************************)
|
|A_FERTI ABRAUME IN_SCH3                                                R_FERTI
+--] [-----] [-----]/[-------------------------------------------------(S)--
|
|ST_RESE                                                                R_FERTI
+--] [--+-------------------------------------------------------------(R)--
|       |
|  SCHR0 |
+--] [--+
|
|  (*************************************************************************)
|  (* << Network 6 : Flag whether step 4 was active >>                      *)
|  (*************************************************************************)
|
|  SCHR4                                                                SCHR_VA
+--] [-------------------------------------------------------------------(S)--
|
|  SCHR0                                                                SCHR_VA
+--] [-------------------------------------------------------------------(R)--
|
|  (*************************************************************************)
|  (* << Network 7 : Flag whether step 3 was active >>                      *)
|  (*************************************************************************)
|
|  SCHR3                                                                IN_SCH3
+--] [-------------------------------------------------------------------(S)--
|
|  SCHR8                                                                IN_SCH3
+--] [-------------------------------------------------------------------(R)--
|
|  (*************************************************************************)
|  (* << Network 8 : Establish whether a part is still in the magazine >>   *)
|  (*************************************************************************)
|
|  SCHR0    V1B3    MAG_VOL   S103                                      MAG_VOL
+--] [-----]/[-----]/[-----] [-----------------------------------------(S)--
|
|  V1B3     SCHR0                                                       MAG_VOL
+--] [-----] [-----------------------------------------------------------(R)--
|
|  (*************************************************************************)
|  (*       Step sequence module                                            *)
|  (*************************************************************************)
|
|  (*************************************************************************)
|  (* << Network 1 : step 0 : Initialising step >>                          *)
|  (*************************************************************************)
|
|  SCHR7    SCHR8    V1S3    ANL_GO                                     SCHR0
+--]/[-----] [-----] [--+--] [-----------------------------------------(S)--
|                       |
|R_FERTI                |
+--] [------------------+
|
|  SCHR1                                                                SCHR0
+--] [--+-------------------------------------------------------------(R)--
|       |
|ST_RESE|
+--] [--+
|
|  (*************************************************************************)
|  (* << Network 2 : Step 1 : Swivel cyclinder to lifting cyclinder >>      *)
|  (*************************************************************************)
|
|
```


SCHR8 SCHR0 MAG_VOL EINZEL FS_GRUN SCHR1
ABLAUF
TMP_1
ABRAUME V1B1 V1S2 MAG_VOL SCHR_VA SCHR2 SCHR3 SCHR4 SCHR5
IN_SCH3
SCHR6 SCHR7 SCHR8 SCHR0 V1S4 TMP_1
SCHR2 SCHR1
ST_RESE
(* << Network 3 : Step 2 : Ejection cyclinder advance >> *)
SCHR0 SCHR1 V1S4 EINZEL SCHR2
ABLAUF
ABRAUME V1S4 SCHR1 SCHR0
TMP_2
V1S2 SCHR3 SCHR4 SCHR5 SCHR6 SCHR7 SCHR8 SCHR0 SCHR_VA TMP_2
SCHR3 SCHR2
ST_RESE
(* << Network 4 : step 3 : Swivel cyclinder to magazine >> *)
SCHR1 SCHR2 V1S1 EINZEL SCHR3
ABLAUF
ABRAUME V1S1 SCHR1 SCHR2 SCHR4 SCHR5 SCHR6 SCHR7 SCHR8
SCHR4 SCHR3
ST_RESE
(* << Network 5 : Step 4 : Vacuum on >> *)


SCHR2 SCHR3 V1S3 EINZEL SCHR4
ABLAUF
ABRAUME V1S3 SCHR3 SCHR2
TMP_4
V1S2 V1S1 V1B2 SCHR5 SCHR6 SCHR7 SCHR8 SCHR0 TMP_4
SCHR5 SCHR4
ST_RESE
(* << Network 6 : Step 5 : Ejecting cyclinder retract >> *)
SCHR3 SCHR4 V1S2 EINZEL SCHR5
ABLAUF
ABRAUME V1S2 SCHR4 SCHR3
V1S3 SCHR6 SCHR7 SCHR8 SCHR0
SCHR6 SCHR5
ST_RESE
(* << Network 7 : Step 6 : Swivel cyclinder to lifting cyclinder >> *)
SCHR5 SCHR4 V1B1 EINZEL FS_GRUN SCHR6
ABLAUF
ABRAUME V1B1 SCHR5 SCHR4
V1S2 V1S4 SCHR7 SCHR8 SCHR0
SCHR7 SCHR6
ST_RESE
(* << Network 8 : Step 7 : Vacuum off >> *)


SCHR6 SCHR5 V1S4 EINZEL SCHR7
ABLAUF
ABRAUME V1S4 SCHR6 SCHR5
V1S2 SCHR8 SCHR0
SCHR8 SCHR7
ST_RESE
(* << Network 9 : Step 8 : Swivel cyclinder empty to magazine >> *)
SCHR7 SCHR6 V1S2 EINZEL SCHR8
ABLAUF
ABRAUME V1S2 SCHR7 SCHR6
V1B1 V1S1 V1S3 SCHR_VA SCHR0
SCHR0 SCHR8
ST_RESE
ABRAUME V1S3
(* Output assignment module *)
(* << Network 1 : Swivel cyclinder to lifting cyclinder >> *)
SCHR1 V1Y4
SCHR6
ST_RESE V1Y4
SCHR3
SCHR8
(* << Network 2 : Swivel cyclinder to magazine >> *)
SCHR3 V1Y5
SCHR8


ST_RESE
V1Y5
(R)
SCHR1
SCHR6
SCHR0
(* << Network 3 : Vacuum on >> *)
SCHR4
V1Y2
(S)
ST_RESE
V1Y2
(R)
SCHR7
(* << Network 4 : Vacuum off >> *)
SCHR7
V1Y3
(S)
ST_RESE
V1Y3
(R)
SCHR4
SCHR0
(* << Network 5 : Ejection cylinder advance >> *)
SCHR2
V1Y1
(S)
ST_RESE
V1Y1
(R)
SCHR5
(* << Network 6 : Light for initial position >> *)
BL_RICH BLINK2H
LA_GRST
( )
R_FERTI
ANL_GO A_FERTI
(* << Network 7 : Light for magazine empty >> *)
SCHR0 MAG_VOL BLINK2H
LAMPE_M
( )
(* < Network 8: Start button warning light; to be actuated if illuminated > *)


EINZEL ABLAUF ANL_GO SCHR0 MAG_VOL
AUTO_LA
SCHR1 V1S4
SCHR2 V1S1
SCHR3 V1S3
SCHR4 V1S2
SCHR5 V1B1
SCHR6 V1S4
SCHR7 V1S2
Flashing pulse module
<< Network 1 : Pulse generator - Timer >>
TMP_B2H
BLINK2H
TMR
0.10s
CONST
PV
+00005
%R0005
BLINK2H
TMP_B2H
TMR
0.10s
CONST
PV
+00005
%R0015
Communication Module
<< Network 2 : Lamp following station is on >>
I_S_RDY
VERNETZ
<< Network 3 : Transfer of distrib. st. in initial position >>
V1S3 V1B1 V1S2
O_EN
(M)
<< Network 4 : Reading,whether subseq. st. is in initial position >>
I_P_RDY I_S_RDY
FS_GRUN
END OF SUBROUTINE LOGIC

โปรแกรมสำหรับเคลื่อนย้ายข้อมูล

```
|[  START OF LD  PROGRAM  SNPX    ]       (*                                  *)
|
|[     VARIABLE DECLARATIONS      ]

|[       BLOCK DECLARATIONS       ]

                +--------+
        SUBR  1 |DISTRIB|       LANG: LD  (* Distribution Station             *)
                +--------+

                +--------+
        SUBR  2 |TESTING|       LANG: LD  (* Testing Station                  *)
                +--------+

                +--------+
        SUBR  4 |BUFFER |       LANG: LD  (* Buffer Station                   *)
                +--------+

                +--------+
        SUBR  3 |SORTING|       LANG: LD  (* Sorting Station                  *)
                +--------+

                +--------+
        SUBR  5 |COM_PLC|       LANG: LD  (* Communicate MicroPLC on P+ H+    *)
                +--------+

|[     START OF PROGRAM LOGIC     ]
|
|FST_SCN +-----+
+--] [---+ BLK_+-
|        | CLR_|
|        | WORD|
|%G0001 -+IN   |
|        | LEN |
|        |00016|
|        +-----+
|
|ALW_ON  +-------------+
+--] [---+ CALL DISTRIB+
|        | (SUBROUTINE)|
|        +-------------+
|
|ALW_ON  +-----+                  +-----+
+--] [---+MOVE_+------------------+MOVE_+-
|        | BIT |                  | BIT |
|        |     |                  |     |
|%G0025 -+IN  Q+-%G0041  %G0049 -+IN  Q+-%G0001
|        | LEN |                  | LEN |
|        |00008|                  |00008|
|        +-----+                  +-----+
|
|ALW_ON  +-------------+
+--] [---+ CALL TESTING+
|        | (SUBROUTINE)|
|        +-------------+
|
|ALW_ON  +-----+                  +-----+
+--] [---+MOVE_+------------------+MOVE_+-
|        | BIT |                  | BIT |
|        |     |                  |     |
|%G0057 -+IN  Q+-%G0073  %G0081 -+IN  Q+-%G0033
|        | LEN |                  | LEN |
|        |00008|                  |00008|
|        +-----+                  +-----+
|
| (*************************************************************************)
| (*      Quality Station on PLC90-70                                      *)
| (*************************************************************************)
```

```
|ALW_ON  +-----+                  +-----+
+--] [---+MOVE_+------------------+MOVE_+-
|        | BIT |                  | BIT |
|        |     |                  |     |
|%G0089 -+IN  Q+-%G0105  %G0113 -+IN  Q+-%G0065
|        | LEN |                  | LEN |
|        |00008|                  |00008|
|        +-----+                  +-----+
|
|ALW_ON  +-------------+
+--] [---+ CALL BUFFER +
|        | (SUBROUTINE)|
|        +-------------+
|
|ALW_ON  +-----+                  +-----+
+--] [---+MOVE_+------------------+MOVE_+-
|        | BIT |                  | BIT |
|        |     |                  |     |
|%G0121 -+IN  Q+-%G0137  %G0145 -+IN  Q+-%G0097
|        | LEN |                  | LEN |
|        |00008|                  |00008|
|        +-----+                  +-----+
|
|  (*****************************************************************************)
|  (*      PROCESSING STATION on Micro PLC                                      *)
|  (*****************************************************************************)
|
|ALW_ON  +-------------+
+--] [---+ CALL COM_PLC+
|        | (SUBROUTINE)|
|        +-------------+
|
|ALW_ON  +-----+                  +-----+
+--] [---+MOVE_+------------------+MOVE_+-
|        | BIT |                  | BIT |
|        |     |                  |     |
|%G0153 -+IN  Q+-%G0169  %G0177 -+IN  Q+-%G0129
|        | LEN |                  | LEN |
|        |00008|                  |00008|
|        +-----+                  +-----+
|
|  (*****************************************************************************)
|  (*      ROBOT STATION on micro PLC                                           *)
|  (*****************************************************************************)
|
|ALW_ON  +-----+                  +-----+
+--] [---+MOVE_+------------------+MOVE_+-
|        | BIT |                  | BIT |
|        |     |                  |     |
|%G0185 -+IN  Q+-%G0201  %G0209 -+IN  Q+-%G0161
|        | LEN |                  | LEN |
|        |00008|                  |00008|
|        +-----+                  +-----+
|
|ALW_ON  +-------------+
+--] [---+ CALL SORTING+
|        | (SUBROUTINE)|
|        +-------------+
|
|[      END OF PROGRAM LOGIC      ]
```

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

ประวัติผู้เขียนวิทยานิพนธ์

นาย ราชิต ลิมป์จันทรา เกิดเมื่อวันที่ 21 พฤษภาคม พ.ศ. 2517 ที่จังหวัดกรุงเทพมหานคร สำเร็จการศึกษาปริญญาตรี วิศวกรรมศาสตร์บัณฑิต ภาควิชาวิศวกรรมเครื่องกล คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย ในปีการศึกษา 2539 และได้เข้าศึกษาต่อในหลักสูตรวิศวกรรมศาสตร์มหาบัณฑิต ภาควิชาวิศวกรรมเครื่องกล คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย ในปี พ.ศ. 2540