ระบบเฝ้ามองและติดตามผลของกระบวนการโดยผ่านเครือข่ายโทรศัพท์

(PROCESS MONITORING AND DATA ACUISITION WITH MODEM)

โดย

ทวีพล	ซื่อสัตย์	36013287
คมน์	สุอังคะวาทิน	36013279
สมชาย	นาคราช	36013308

วัน เดือน ปี 1 ต.ค. 2540
เลขทะเบียน 037162
เลขเรียกหนังสือ T 38255 ทพน ร

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิศวกรรมศาสตร์บัณฑิต

สาขาวิศวกรรมการวัดคุมทางอุตสาหกรรม

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2538

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

037162

ปริญญานิพนธ์ ปีการศึกษา 2538

ภาควิชา เทคโนโลยีการวัดคุมทางอุตสาหกรรม

สาขาวิชา วิศวกรรมการวัดคุม

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง ระบบเฝ้ามองและติดตามผลของกระบวนการโดยผ่านเครือข่ายโทรศัพท์
(PROCESS MONITORING AND DATA ACUSITION WITH MODEM)

ผู้จัดทำ

1.	นายทวีพล	ซื่อสัตย์	36013287
2.	นายคมน์	สุอังคะวาทิน	36013279
3.	นายสมชาย	นาคราช	36013308

.....อาจารย์ที่ปรึกษา

(อาจารย์ สุพรรณ กุลพาณิชย์)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระบบเฝ้ามองและติดตามผลของขบวนการโดยผ่านเครื่อข่ายโทรศัพท์
(PROCESS MONITORING AND DATA ACUISITION WITH MODEM)

โดย

ทวีพล ซื่อสัตย์ 36013287

คมน์ สุอังคะวาทิน 36013279

สมชาย นาคราช 36013308

อาจารย์ที่ปรึกษา

(ผศ. สุพรรณ กุลพาณิชย์)

บทคัดย่อ

ในปัจจุบัน เครื่องควบคุมแบบตรรกะ(Programmable logic control .PLC) มีใช้กันอย่างแพร่หลาย และได้มีการพัฒนาในรูปแบบที่ แตกต่างกัน ในปริญญานิพนธ์ฉบับนี้ได้กล่าวถึงการพัฒนาPLCในการควบคุมระยะไกล โดยมีโมเด็มและเครือข่ายโทรศัพท์ทำหน้าที่เชื่อมโยงสัญญาณ แต่การนำPLC มาต่อใช้งานรวมกับอุปกรณ์อื่นทางพอร์ทอนุกรม เช่น โมเด็มเป็นเรื่องยุ่งยาก ต้องมีอุปกรณ์ที่ทำให้โมเด็มต่อใช้งานรวมกับ PLC ได้คือ ASCII UNIT BASIC PROGRAMซึ่งเป็นอุปกรณ์พิเศษ ทำหน้าที่เสมือนคอมพิวเตอร์ตัวหนึ่ง ที่คอยจัดการกับหน่วยความจำของ PLC มีภาษาระดับสูงในการใช้งาน นอกจากนี้แล้วที่ปลายทางยังมีคอมพิวเตอร์ตัวแม่ ซึ่งคอยรับส่งข้อมูลกับ PLC เพื่อแสดงผล และควบคุมการทำงานของPLC ด้วย

Abstract.

Recently, To develop of process control is quickly . Programmable logic control (PLC) is the famous process control system . This thesis is the development of PLC. for remote control with modem and telephone network . But the communication between RS 232 C equipment and PLC. is difficult . so that we must have the ASCII UNIT BASIC PROGRAM for communicate

This project is explained the application of the monitoring and data ascuisition with modem:

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

ขอขอบคุณ บิดามารดาให้เราได้เกิดมา ขอขอบคุณท่านอาจารย์ที่ปรึกษาและประชาชน
ทุกท่านให้การสนับสนุนโครงงานนี้
ขอบคุณ การสื่อสารแห่งประเทศไทย
ขอบคุณ การท่าเรือแห่งประเทศไทย
ขอบคุณ คุณคธาชิต อทิฐิกานต์กุล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1

บทนำ

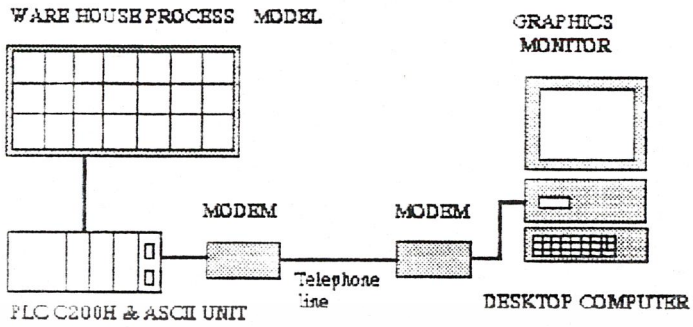
ในวงการอุตสาหกรรมปัจจุบัน ระบบควบคุมอัตโนมัติถูกนำมาใช้แทนแรงงานคนเป็นอันมาก เนื่องด้วยเหตุและผลหลายประการ ที่เครื่องจักรควบคุมอัตโนมัติ ถูกเลือกมาใช้งาน อาทิ เช่น ความละเอียด ความเที่ยงตรง ความรวดเร็ว และการทำงาน ภายใต้สิ่งแวดล้อมที่เป็นพิษ ซึ่งจะเป็นความได้เปรียบของเครื่องความคุมอัตโนมัติ ถึงอย่างไรก็ตามการพัฒนาของระบบควบคุมอัตโนมัติ ก็ยังไม่จบลง เพื่อขจัดปัญหาในการเดินทาง ความคิดที่จะมีการควบคุมและติดตามผลในระยะไกล จึงเกิดขึ้นซึ่งปัจจุบันก็มีอยู่มาก เช่น สกาด้า หรือรีโมต คอนโทรลยูนิต แต่ในโครงงานนี้ เราใช้ PLC เป็นตัวควบคุมและใช้ คู่สายโทรศัพท์เป็นสื่อกลาง

เครื่องควบคุมแบบโปรแกรมได้ (PROGRAMMABLE LOGIC CONTROL) เป็นอุปกรณ์ ที่ใช้กันแพร่หลาย และมีอุปกรณ์ที่ใช้งานร่วมกันเพื่อเพิ่มประสิทธิภาพอีกมาก เครื่อง PLC รุ่น C 200 H เป็นรุ่นที่ออกแบบเพื่อให้ใช้งานได้กว้างขวางมาก ASCII UNIT BASIC PROGRAM ซึ่งเป็นอุปกรณ์ตัวหนึ่งที่ทำหน้าที่จัดการค่าใน DATA MEMORY ของ PLC สามารถส่งและรับค่าแบบอนุกรมได้ เราจึงให้มันเป็นอุปกรณ์ ที่จะติดต่อกับ โมเด็ม เพื่อส่งข้อมูลเก็นสายโทรศัพท์

ที่เครื่องคอมพิวเตอร์ จะมีโมเด็มอีกตัวหนึ่งซึ่งทำหน้าที่รับข้อมูล จากเครื่องควบคุมกระบวนการ เราสามารถที่จะดูสภาวะการทำงานของกระบวนการ และควบคุมสั่งงานจากหน้าจอคอมพิวเตอร์ได้

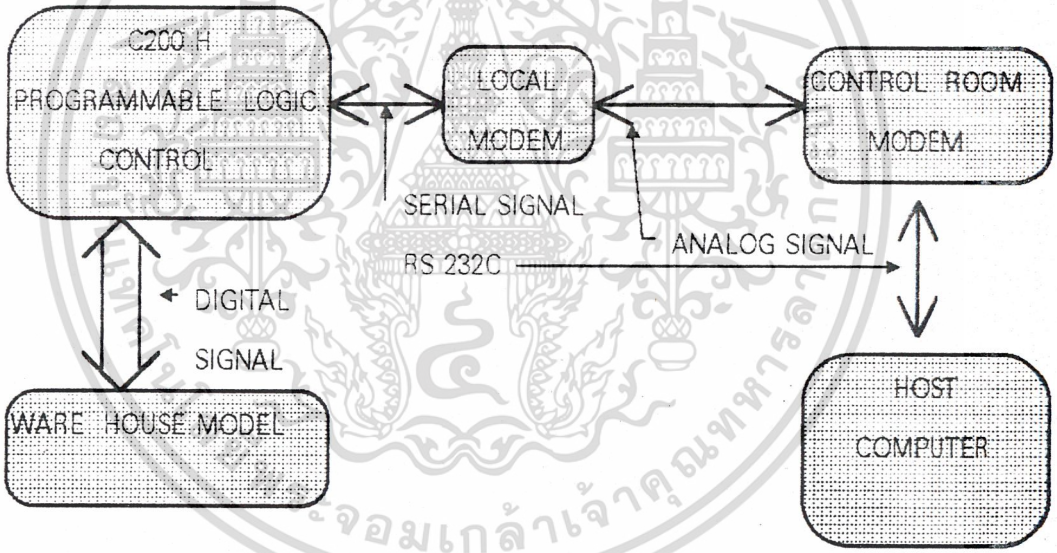
ในโครงงานนี้ เราทดลอง จาก ชั้นเก็บของ จำลอง (WARE HOUSE MODEL) เป็นกระบวนการที่เราควบคุม รายละเอียดต่าง ๆ อธิบายในรูปที่ 1.1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 1.1 แสดงการเชื่อมต่อของระบบ

BLOCK DIAGRAM แสดงทิศทาง และชนิดของสัญญาณ



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2

ทฤษฎี เนื้อหา ประกอบโครงงาน

2.1 PLC (PROGRAMMABLE LOGIC CONTROL)

2.2.1 INTRODUCTION TO PROGRAMMABLE LOGIC CONTROL

ระบบควบคุมแบบซีเควนซ์ (SEQUENCE CONTROL) แต่เดิมประกอบด้วยไฟฟ้า เชิงกล (ELECTROMECHANICAL DEVICE) ซึ่งได้แก่ replay,timer,counter แต่เดิมประกอบด้วยอุปกรณ์ ไฟฟ้าการทำงานของระบบควบคุมนี้จะมีอยู่ สองสภาวะด้วยกัน คือ เปิด กับ ปิด (ON,OFF) ระบบซีเควนซ์ที่ประกอบด้วยอุปกรณ์ ดังกล่าวนี้ มีขนาดใหญ่ ใช้กำลังงานสูง และเมื่อมีการเปลี่ยนแปลงการทำงานก็ต้องทำการสร้างวงจรควบคุมใหม่ ซึ่งไม่สามารถประยุกต์ใช้กับระบบควบคุมที่มีความยุ่งยากและซับซ้อน เหล่านี้เป็นต้น

เนื่องจากเทคโนโลยีทางด้าน microprocessor ในปัจจุบัน ได้เจริญก้าวหน้าไปมาก มีการนำ microcessor มาใช้กับการควบคุมแบบต่อเนื่อง ซึ่งก็คือเครื่องควบคุมแบบโปรแกรมได้ (Programmable sequence control) หรือเรียกสั้น ๆ ว่า PC ขึ้นมาใช้กับการควบคุมสำหรับงานด้านอุตสาหกรรมโดยเฉพาะ

PC จะมีส่วนที่เป็นอินพุทที่สามารถต่อใช้งานได้กับ ตัวตรวจจับต่าง ๆ (Sensor) และส่วนเอาท์พุทจะต่อไป ควบคุมการทำงานของอุปกรณ์ หรือเครื่องจักรกล โดยสามารถสร้างวงจรหรือเงื่อนไข การทำงานของเครื่องจักรเหล่านี้ได้จากการป้อนโปรแกรมสั่งงานเป็นภาษาที่เรียกว่า LADDER DIAGRAM เข้าไป โปรแกรมนี้จะทำหน้าที่เหมือนกับวงจรรีเลย์ ตัวตั้งเวลา ตัวนับ และอื่น ๆ อีก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

นอกจากนี้ PC ที่มีให้เลือกใช้อยู่ในปัจจุบัน มีการออกแบบระบบให้มีความยืดหยุ่นสูงในการที่จะต่อร่วมกับอุปกรณ์สมทบประเภทต่าง ๆ ทั้งนี้เพื่อเป็นการเพิ่มประสิทธิภาพการทำงานของ PC ให้มีสมรรถนะสูงขึ้น

อุปกรณ์สมทบต่าง ๆ มีดังนี้

1. I/O UNITE

   INPUT OUTPUT MODULE

   OPTICAL TRANSMITTING I/O UNIT

2. SPECIAL I/O UNITS

   ANALOG INPUT /OUTPUT UNITS

   PID UNIT

   TEMPERATURE SENSOR UNIT

   HIGH-SPEED COUNTER UNIT

   POSITION CONTROL UNIT

   ASCII UNIT

   LADDER PROGRAM I/O UNIT

   FILE MEMORY UNIT

   MCR UNIT (MAGNETIC CARD READER)

3. LINK SYSTEM

   REMOTE I/O SYSTER

   I/O LINKS

   PC. LINK SYSTEMS

   HOST LINK SYSTEMS

   SYSMAC NET LINK SYSTEMS

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. PERIPHERAL TOOLS

PROGRAMMING CONSOLE

EPROM WRITER

PRINTER INTERFACE UNIT

SYSMAC C - SERIES SUPPORT TOOLS

FIT : FACTORY INTELLIGENT TERMINAL

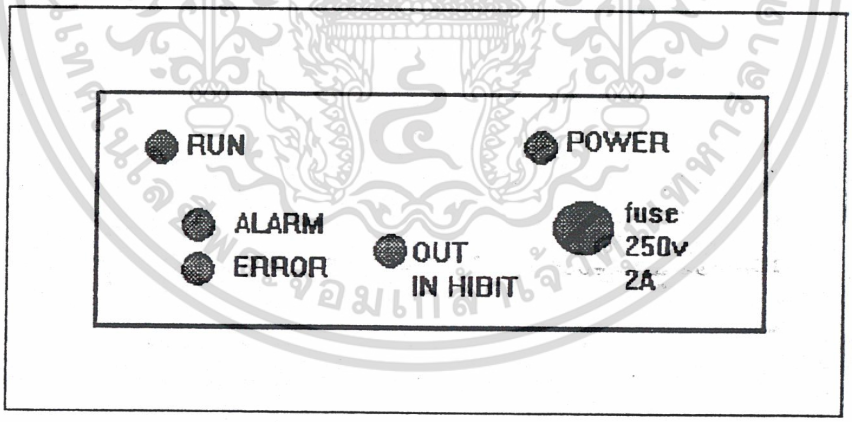
LSS: LADDER SUPPORT SOFTWARE

GPC: GRAPHICS PROGRAMMING CONSOLE

PLC C200 H เป็น PLC รุ่นที่สร้างขึ้นมา มีความยืดหยุ่นสูงมาก ใช้ LADDER PROGRAM ในการควบคุมกระบวนการซึ่งรายละเอียดทางเทคนิค มีดังต่อไปนี้

Programmable logic control C200 H

- ส่วนแสดงผลของ CPU จะเป็น LED ที่ใช้แสดงผลเหมือนกับ PC โดยทั่วไป ดังจะได้ทราบความหมายจากตาราง



รูปที่ 2.1.1 แสดง LED ในตำแหน่งต่าง ๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หลอดแสดงผล	ความหมาย
POWER	จะสร้างเมื่อจ่ายไฟให้กับ CPU
RUN	จะสร้างเมื่อ CPU ทำงาน
ALARM / ERROR	ALARM : จะสร้างขึ้นมาเมื่อไม่พบการ ERROR ของ PC
	ERROR : สร้างเมื่อเกิดการ ERROR และเมื่อ LED ติด LED ของRUN
	จะดับลง, CPU ก็หยุดการทำงาน และ OUTPUT จาก PC ก็จะปิดลง
OUT INHIBIT	จะสร้างเมื่อ Bit output off, SR 25215 จะกลับมา ON แต่ output
	from PC จะปิดตัวลง

<u>รูปร่างภายนอกของ PC -C200H</u>

พื้นฐานโครงสร้างของ PC ที่จะประกอบด้วย Lack 2 ส่วน คือ CPU and Expansion I/O

Expansion I/O เป็นส่วนที่ไม่ต้องการของระบบพื้นฐานคือในส่วนที่จะใช้ในการเพิ่มจำนวนของ I/O Points.

Rack ชนิดที่ 3 ซึ่งเป็นส่วนที่เพิ่มเข้ามา จะเป็น Rack ที่เรียกว่า Slave Rack จะใช้ต่อเมื่อ PC ถูกเตรียมมีการให้ใช้กับระบบ Remote I/O System

- CPU RACK

Rack ของ CPU C200H ประกอบด้วย ส่วน 4 ส่วน คือ

ส่วนที่ 1 CPU Rackplane ซึ่งประกอบด้วย CPU และ Unit อื่น ๆ ที่เพิ่มขึ้น

ส่วนที่ 2 CPU ที่ประกอบด้วยส่วนของการกระทำ Program และ Control PC

ส่วนที่ 3 ส่วนอื่น ๆ ซึ่งได้แก่ I/O unit,special I/O unit, และ Link unit ตามหลักการแล้ว I/o terminals ที่มีลักษณะเดียวกันกับ I/O points

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

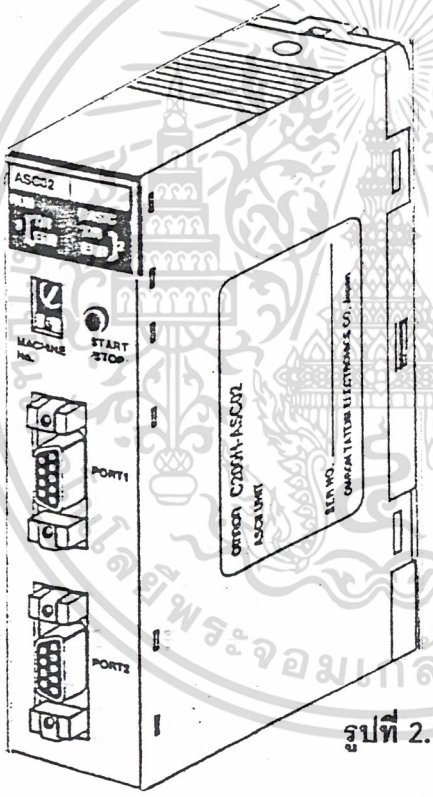
Rack CPU ของ C200H ที่สามารถที่จะใช้ เดี่ยว ๆ หรือ ต่อร่วมกับ Rack I/O points ที่เพิ่มขึ้นก็ได้ CPU Rack อาจจะมี 3, 5 หรือ 8 slot ก็ได้ ซึ่งเราก็สามารถเพิ่มในส่วนอื่นนี้เข้าไปบน book plane ที่ใช้

- Expansion I/O RACKS

Expansion I/O RACKS สามารถที่จะขยายออกไปได้ และจะต่อกับ CPU ON Bach planes, การที่จะทำการติดต่อระหว่าง 2 Rack Expansion I/o Rack จะต่อ Weries ไปที่ Rack

- Unit Mounting position

I/O unit and special I/O unit สามารถที่จะขยายไปที่ slave Racks ได้ I/O units, special I/O unit,remote I/O,pc และ Host link unit สมารถที่จะขยายไปที่ slot บน Rack ได้



รูปที่ 2.1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.2 ASCII UNIT BASIC PROGRAM

เนื่องจากเกิดปัญหาเมื่อทำการติดต่อระหว่าง PLC กับอุปกรณ์ทางพอร์ทอนุกรม RS232C ซึ่งจะทำได้ยาก เพราะต้องทำตามโปรโตคอลของ PLC ความยุ่งยากอยู่ที่ต้องมีการคำนวณและแปลงฐานเลข ในทางคณิตศาสตร์ ซึ่งตัว ASCII UNIT BASIC PROGRAM สามารถแก้ปัญหาตรงนี้ได้ ASCII UNIT สามารถที่จะติดต่อระหว่างหน่วยความจำในPLC เช่น DATA MEMMORY กับอุปกรณ์ภายนอก ได้โดยตรง

ASCII UNIT เป็นฮาร์ดแวร์ที่ทำหน้าที่เป็นคอมพิวเตอร์ที่ HOOK ลงไปในฮาร์ดแวร์ของPLC สามารถเขียนโปรแกรมภาษาระดับสูง(ภาษา BASIC) เพียงแต่มองคอมพิวเตอร์ภายนอกเป็นแต่ TERMINAL เพื่อนำมาแก้ไขโปรแกรมบน CPU ของ ASCII UNIT

PC จะเป็นตัวที่ใช้เตือนตลอด line input โดยเฉพาะ input ที่แสดงประสิทธิภาพในการนับเวลา อุณหภูมิ ตำแหน่ง ค่าของข้อมูล โดยตัวเลข ในค่า input เหล่านี้จำเป็นอย่างยิ่งที่ PC จำเป็นอย่างยิ่งที่ PC จำเป็นต้องส่ง สัญญาณที่เหมาะสม ไปที่อุปกรณ์ เพื่อที่จะปรับหรือรักษาสภาพการทำงาน ของ ระบบ Controlled

PC จะเป็นตัวตัดสินใจ บนพื้นฐานของค่าที่เก็บไว้ล่วงหน้าอย่างภาวนาในหน่วยความจำของมัน เช่น PC ที่จะแสดงผลของอุณหภูมิของระบบเครื่องกล มันจะทำการเปรียบเทียบค่าอยู่ตลอดเวลา และมีนจะแสดงคำว่า "อันตราย" ถ้าอุณหภูมิในระบบสูงเกินไป และ PC จะทำการปิดระบบลง จนกว่าระบบจะอยู่ในระดับที่ปลอดภัยแล้ว

นอกจากนี้แล้วยังมีตัวอย่างอีกมากมาย ในระบบที่ยุ่งยากมาก ๆ อาจจะต้องใช้ Process ปริมาณข้อมูลที่มากกว่า จากความแตกต่างของ input มาก ๆ พื้นฐานของผลิตภัณฑ์ ทางคณิตศาสตร์ ความสัมพันธ์ และ การคำนวณ logic มาทำการตัดสิน และ PC จำเป็นต้องทำรายละเอียดเหล่านั้นมาทำเป็นรายละเอียดมาแสดงด้วย ASCII UNIT

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

PC สามารถเป็นตัวแทนในการทำการปฏิบัติการทางด้านข้อมูล และตัดสินใจในการทำงานอย่างอื่น เพราะที่ ASCII unit คือ โปรแกรมใน Basic แทนที่ ของ Program Ladder

ASCII unit จะยอมให้ผู้ไปในข้อมูลที่ต้องการได้ง่าย ทาง printer or display terminal การใช้ ASCII Unit เพื่อให้ดูมีประสิทธิภาพมากขึ้น พื้นฐานระบบ control ของ PD จะต้องมีกำลังมาก เปลี่ยนแปลงได้ง่าย และเป็นเครื่องมือมี่มีประสิทธิภาพ ในส่วนนี้รูปพรรณภายนอกของ Hardware ของ ASCII Unit ด้านหน้าและด้านหลังจะประกอบด้วย

Switch,button,Connector ตัวที่แสดงสำหรับให้ผู้ใช้ set up, ควบคุมและเตือนการทำงานของ ASCII Unit

1. ส่วนหน้า

ส่วนหน้าจะประกอบด้วย port ติดต่อ RS 232 C 2 port, Switch Start,stop หน่วยให้เลือก Number และ หลอดไฟแสดงสภาวะชนิดต่าง ๆ ส่วนหลังจะประกอบด้วย Dip Switch 2 ชุด สำหรับการ SET ตัวแปรของ ASCII Unit และตัวติดต่อ PC

PORT

ส่วนข้างหน้าของ ASCII Unit จะประกอบด้วย RS 232C จำนวน 2 Port Port 1 หลักที่ใช้ต่อกับอุปกรณ์ I/O ที่อยู่ด้วยรอบ ๆ กับ ASCII Unit port ทั้งคู่สามารถใช้ติดต่อกับอุปกรณ์ printers,terminal and modem อย่างไรก็ตามใน port 1 ก็จะใช้ได้ทั้ง uploading and down loading ในโปรแกรม Basic รูปมาตรฐานที่ต่อในคอมพิวเตอร์ส่วนตัวจะใช้ port 1 and printer หรือ อุปกรณ์ I/O ต่าง ๆ ต่อ port2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

SWITCH

รายละเอียดเหมือนกับ Switch ASCII Unit โดยเหตุที่เป็นไปได้ที่จะมี ASCII Unit มากกว่า 1 ต่อไปที่ PC The machine No จะเหมือนกันเฉพาะ ASCII Unit มันจะไม่ยอมให้มี ASCII Unit 2 ตัว ใช้กับ machine No ที่เหมือนกัน Machine No สมารถ SET ได้ตั้งแต่ 1-9 จะต้องทำการ SET ก่อนที่จะนำไปใช้ประโยชน์

INDICATORS

จะมีตัวแสดงผลอยู่ด้านหน้า 4 หลอด รูปร่างแสดงอยู่หน้าถัดไป

LED DISPLAY

เป็นการแสดงการทำงานของ ASCII Unit

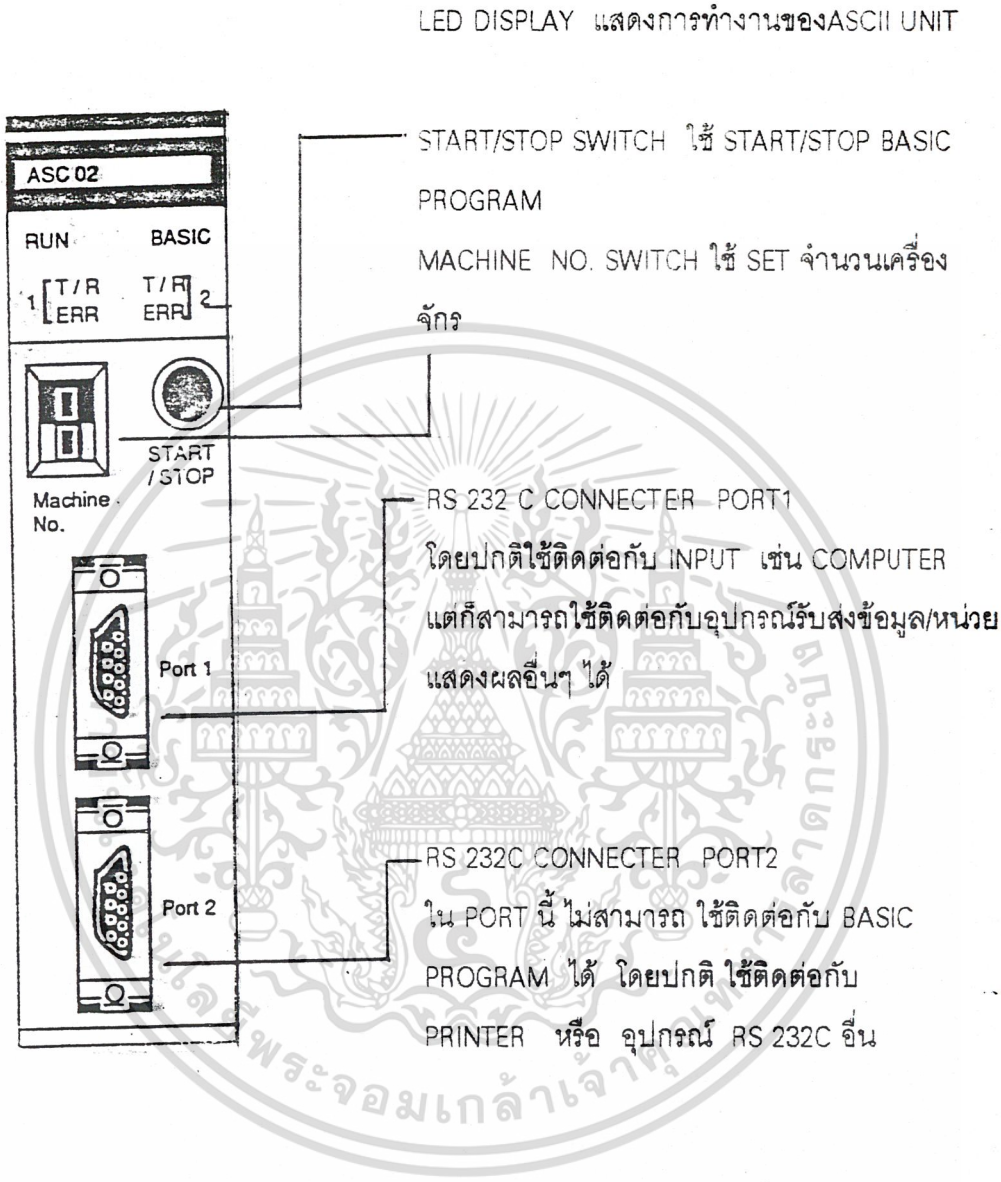
START/STOP SWITCH

ใช้ Start/Stop program basic

MACHINE No SWITCH

ใช้ SET Number the ASCII the ASCII machine

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.2 แสดงการใช้งานของ Port

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

RS-232C Connector port 1

ใช้ติดต่อกับอุปกรณ์รับข้อมูลและหน่วนแสงดผล โดยทั่วไปจะใช้ที่ input ที่ basic program แต่ก็สามารถที่จะใช้กับอุปกรณ์รับข้อมูล อื่น ๆ ได้ดี

RS 232 Connector port 2

ใช้ติดต่อกับอุปกรณ์รับข้อมูล/แสดงผล ไม่สามารถที่จะใช้ กับ input program basic ได้ โดยปกติจะใช้กับ printer หรือ RS232C

LED แสดงสภาวะ

Run  สีเขียว จะสร้างเมื่อ ASCII Unit ทำงานปกติ จนกว่าจะเกิด error

T/R for port 1 and 2  จะกระพริบระหว่างที่กำลังส่งข้อมูล (port 1 and port 2)

ERR1 (ข้อผิดพลาดสำหรับ port 1)  จะสร้างเมื่อเกิด error ขึ้น หรือเมื่อ ASCII Unit

ERR2 (ข้อผิดพลาดสำหับ port 2)  จะกระพริบเมื่อแรงดันของแบตเตอร์รี่ตกต่ำเกินกว่าค่าพิกัด หรือเมื่อแบตเตอร์รีไม่ได้รับทำการบรรจุในที่ถูกต้อง

BASIC  จะสร้างขณะที่ program basic ทำงาน

 จะกระพริบเมื่อ program basic หยุดทำงาน หรือเมื่อ ASCII Unit กำลังคอย input ขณะที่ program basic กำลังทำงาน ● จะดับใน monitor mode

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ส่วนข้างหลัง

ในส่วนที่จะอธิบาย การทำงานในส่วนข้างหลังของ ASCII Unit ในส่วนที่จะมี Dip Switch S pinอยู่ 2 ข้างหลังของ ASCII Unit เราจะทำการ SET ก่อนที่จะนำมา ASCII Unit เสียบไปที่แผงวงจร

ความหมายของ Dip Switch ด้านซ้าย

Pin 1 ใช้ในการเลือก start up mode ของ ASCII Unit program basic สามารถที่ Boot เครื่องได้อย่างอัตโนมัติ หรือกระตุ้นโดยการกด Switch star/stop

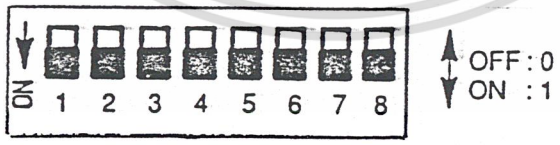
Pin 2 ยอมให้ ms load program Basic จาก EPPROM ไป RAM โดยอัตโนมัติ

Pin 3,4 เป็นการใช้เลือกว่า 3 program Basic อันไหนจะใช้ในการ boot program

Pin 5 ไม่ได้ใช้

Pin 6,7,8 ใช้ในการเลือกขนาดจอภาพของ Display terminal

รูปร่างของ Dip Switch และรายละเอียดอยู่ในหน้าต่อไป



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Pin 2 การสั่ง Program อัตโนมัติ จาก EPPROM ไป RAM

set ไว้ที่ "0" RAM จะถูกนำมาใช้

set ไว้ที่ "1" Program จะถูกสั่ง และ reset จาก EPPROM ไป RAM โดยอัตโนมัติ

Pin 1 Start Mode

set "0" เช่น start mode ใน mode Program Basic จะไป start up บน power application ว่าจะ start program ต้องกด Switch start/stop หรือขอคำสั่ง start จาก computer ส่วนตัวที่ต่อจาก port 1

set "1" start mode อัตโนมัติ ใน mode นี้ program basic จะถูก start อัตโนมัติบน power application

<u>การกำหนด Dip Switch ด้านขวา</u>

Pin 1,2,3 ใช้ในการ set baud rate ของ port 1

Pin 4 ไม่ได้ใช้

Pin 5,6,7 ใช้ในการ set baud rate ของ port 2

Pin 8 ไม่ได้ใช้

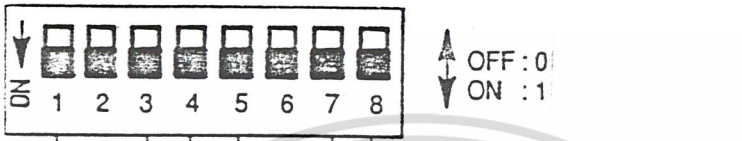
รูปร่างภายนอกของระบบ

ASCII Unit สามารถที่จะใช้เสียบเข้าใน slot CUP Back plane ได้ ยกเว้นสำหรับ 2 rightmost slots แต่ก่อนที่จะเก็บเพิ่มเติมหรือเสียบ ASCII Unit จะต้อง set Dip Switch เสียก่อน เพื่อความแน่นอน power Supply ของ PC จะต้องกด off เสียก่อนระหว่างที่ทำการติดตั้ง ASCII Unit,computer ที่จะใช้ในการเข้าสู่ program basic จะต้อง ต่อ port 1 และ อุปกรณ์ I/O เช่น printer

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

or display terminal สามารถต่อที่ port 2 (ดูได้จากรูป) สำหรับข้อมูลรายละเอียดเกี่ยวอุปกรณ์การติดต่อ และ เวลา ดูได้จากคู่มือที่มาพร้อมกับ 1 เครื่อง

การ set Dip Switch ด้านขวา



Not used (Always set these pins to OFF.)

Baud rate selection for port 2

Pin No.	4	5	6	Baud Rate
Setting	0	0	0	300 BPS
	1	0	0	600 BPS
	0	1	0	1200 BPS
	1	1	0	2400 BPS
	0	0	1	4800 BPS
	1	0	1	9600 BPS
	0	1	1	19,200 BPS
	1	1	1	

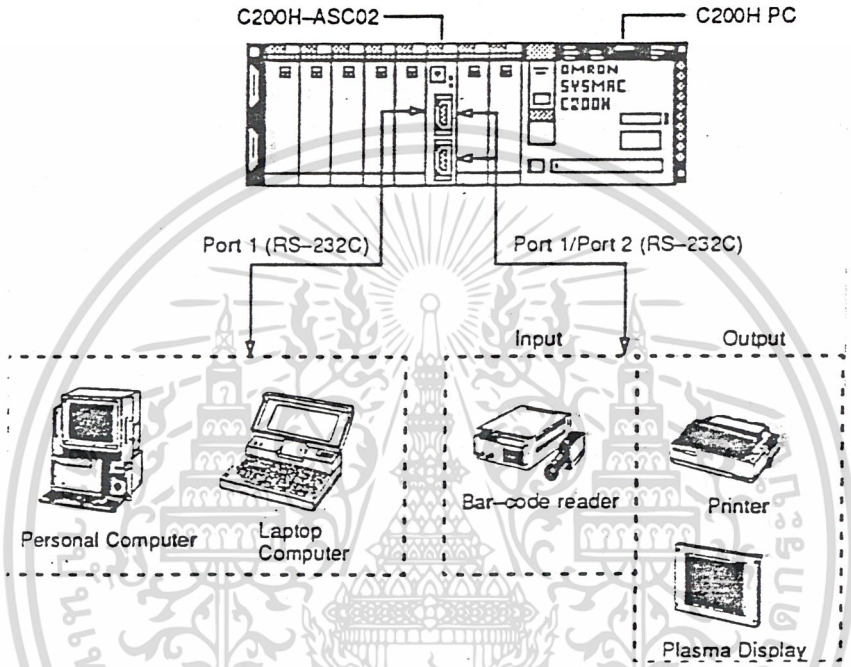
Not used (Always set these pins to OFF.)

Baud rate selection for port 1

Pin No.	1	2	3	Baud Rate
Setting	0	0	0	300 BPS
	1	0	0	600 BPS
	0	1	0	1200 BPS
	1	1	0	2400 BPS
	0	0	1	4800 BPS
	1	0	1	9600 BPS
	0	1	1	
	1	1	1	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การเชื่อมต่อกับอุปกรณ์ภายนอก



รูปที่ 2.3 การต่อใช้งานของ ASCII UNIT

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในส่วนที่จะเป็นการอธิบายในส่วนของข้อมูลของ PC หน่วยความจำพิเศษที่ใช้ในการติดต่อกับ ASCII Unit ในส่วนที่จะกล่าวและให้ความหมายสำหรับเทอมที่มีความสำคัญ ๆ หลายเทอม ซึ่งจะใช้ในคู่มือนี้ตลอด โดย เน้อหาสาระใน Second นี้ จะชัดเจนขึ้นเมื่อท่านเริ่มทำงานเกี่ยวกับ program ASCII Unit อย่างแท้จริง

Bits และ Bytes

หน่วยความจำของ PC ที่จะถูกแยกออกเป็นหลายส่วนซึ่งแต่ละส่วนจะมีชื่อและจุดมุ่งหลายของมันเอง ASCII Unit สามารถที่จะเข้าไปในหน่วยความจำ โดยกำหนด Basic Read and Write (รายละเอียดส่วนมากอยู่ใน Section 4) อย่างไรก็ตาม ข้อมูลIR ของ PC เป็นตัวกำหนด ASCII Unit แต่ละส่วน Machine No Switch บนส่วนข้างหน้าของ ASCII Unit จะใช้เลือกตั้งแต่ 1-9 ตำแหน่ง

หน่วยความจำที่รวมเป็น Unit เรียกว่า words ข้อมูลโดยปกติแล้วจะถูกเก็บอยู่ใน word หรือ กลุ่มของ words แต่ละ words จะมี address อยู่เดียวในหน่วยความจำ computer และจะสามารถเข้าถึงได้โดยระบุ address ของมัน

แต่ละ word จะประกอบด้วย 16 บิต บิตนี้ก็คือส่วนเล็ก ๆ ของข้อมูล ที่สมารถเก็บหรือเข้าถึงได้โดย computer bit นี้จะมีค่าเป็นๆ 1 และ 0 เสมอ แน่นอน บิต สามารถเข้าไปทีละตัวและใช้ได้มีเหมือนสัญญาณ สัญญาณนี้โดยปกติจะ set =1 และ clear = 0 โดย hardware จะเป็นตัวแสดงสภาวะของ computer ว่าจะยอมหรือห้ามการทำงาน บิต สามารถที่จะ set or clear โดย programmer ที่จะติดต่อว่าจะกำหนดค่าที่ตัวแปรหรือวางเงื่อนไขที่ CPU

ตัวอย่างโปรแกรม ASCII UNIT ข้อมูลส่งมาจาก PC โดยการใช้การกำหนด Basic get อย่างไรก็ตามในตอนนี้ PC ยังไม่สามารถติดต่อข้อมูลข่าวสารได้ สัญญาณของ PC จะถูก set ให้เป็น 0 การแสดงผลของ ASCII ต้องรอการแสดงผลเมื่อ PC ที่จะติดต่อกับข้อมูล มันจะตัดสัญญาณไปเป็น 1ในส่วนของข้อมูลการติดต่อด้วย PC word จะถูกแบ่งจาก address 100 to 199

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

037162

ของพื้นที่ หน่วยความจำ อย่างไรก็ตามข้อมูลที่ใช้นี้จะเข้าใจได้ดียิ่งขึ้นหลังจากท่านได้อ่านเกี่ยวกับภาษา Basic และตัวอย่างการใช้ ในคู่มือนี้ก่อน

เราจะได้รู้จากตารางสำหรับรายละเอียดและข้อมูลบนที่นั้น ความเสียหาย และจุดมุ่งหมายของ บิต แต่ละตัว ของข้อมูล

การกำหนด Bit

00	ไม่ได้ใช้	
01	เขียน PC to ASCII	Bit นี้จะใช้เป็นสัญญาณ เมื่อสัญญาณ set เป็น 1 PC จะผ่านคำสั่ง และเริ่มกระทำ คุณสมบัติพิเศษ ของข้อมูล คือ จะส่งจาก PC ไปที่ ASCII unit เริ่มจาก word พิเศษ เมื่อสัญญาณเป็น '0' การกระทำของ PC read ที่สิ้นสุดการกระทำ หมายเลขในการ interrupt เราจะใช้คำสั่ง ON PC Gosub
03	Start ใหม่	การเริ่มแรกในการจะ start ASCII unit ใหม่ สัญญาณที่ส่งผ่านจะเป็นลบ (from on to off)เมื่อสัญญาณ ถูก set ASCII unit ก็จกลับไปเริ่มต้นใหม่
04 - 07	ตัวเลข	4 บิตนี้ จะถูกใช้เป็นจำนวนเลข interrupt เมื่อมีคำสั่ง PC ออกมา บิต นี้จะถูกอ่านตัวเลขตั้งแต่ 01 - 15 และถือเอาจำนวนนี้เป็น interrupt number ขณะที่ bit 00 จะ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

			ถูกตัดทิ้ง
08 - 15		ข้อมูล output	bit นี้จะประกอบด้วยข้อมูลที่จะถูกเขียนไปที่ ASCII ด้วยคำสั่ง move และถูก read ด้วยคำสง PC get ใน basic program ข้อควรจำ ในการรวบรวมแบบหยาบ ๆ address data 8 bit สามารถที่จะส่งไปที่แขนงย่อยของ ASCII unit ได้สะดวกขึ้นด้วย Basic program
n+1	00 - 11	จำนวนของ data word ที่ทำการสั่งโอน	bit พิเศษ ที่จะถูกสั่งโดยคำสั่ง PC read หรือ PC write จำนวน word จะไม่มากไปกว่า 255 word
	12 - 15		ไม่ได้ใช้
n+2	00 - 12	Transfer base word No.	bit ที่จะเป็นการบรรจุ pc base word สำหรับสั่งข้อมูล
	13 - 15	หน่วยความจำ PC	bit นี้จะเป็นการบรรจุในส่วนของ PC memory จากข้อมูลที่ทำการส่ง ระหว่าง PC and ASCII Unit ด้วยคำสั่ง Pc read or Pc write

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

n+3	00 01 - 03 04	ASCII busy port 1 error	บิตที่จะใช้เป็นสัญญาณ set ระหว่างการส่งข้อมูลไม่ได้ใช้ บิตนี้จะถูกใช้เมื่อมีสัญญาณ error เกิดขึ้นใน port
	05 06 07	port 2 error Battery error Basic Run	บิตนี้จะถูกใช้งานขณะที่เกิดสัญญาณ error มันจะทำ set ทำการส่ง เกิด error ใน port2 บิตนี้จะถูกใช้ใน ขณะที่ supply Voltage ของBattery มีค่าต่ำกว่า rate ที่ set ไว้ หรือจะ ถูกใช้ขณะที่ battery ทำการต่อไม่ถูกต้อง สัญญาณที่จะถูก SET ขณะที่ Basic Program นี้กำลัง RUN
n+3	08-15	Input data	บิตนี้จะประกอบด้วย Data ที่ส่งจาก ASCII unit ไปที่ PC ข้อมูลจะถูกเขียนไปที่ PC ด้วย คำสั่ง ASCII Unit PC PUT และอ่านด้วยคำสั่ง MOV

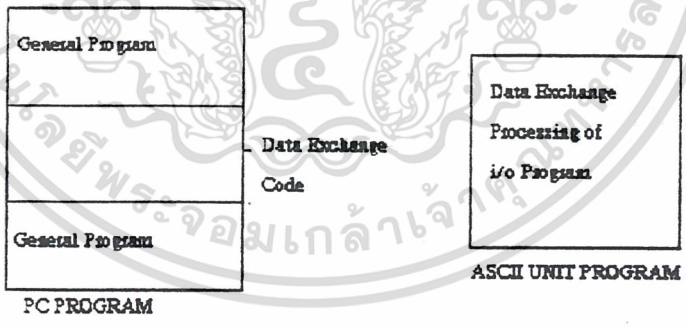
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

มี 2 วิธี ที่ ASCII UNIT สามารถถูกติดต่อสื่อสารโดย PC

วิธีที่ 1 PC จะคอยควบคุม เวลาในการส่งผ่าน ข้อมูลระหว่าง อุปกรณ์ 2 อย่าง ASCII UNIT จะเข้าถึงข้อมูลใน MEMORY PC ที่ใช้โดย คำสั่ง PC READ,PC WRITE ,PC GET,PC PUT หลังจากนั้น ASCII UNIT ก็จะคอยสำหรับ การตอบสนอง ต่อการอ่านหรือ การเขียน การส่งผ่านข้อมูลของ PC จะเป็นงานที่กระทำอยู่เสมอ ในขณะที่ PC ทำงาน, เมื่อ PC READY และ APPROPRIATE FLAG จะถูก เซ็ต

วิธีที่ 2 เมื่อไม่มีข้อมูลส่งผ่านพิเศษ (SPECIAL PC DATA EXCHANGE CODE) จะทำให้ง่ายในการติดต่อสื่อสาร ระหว่าง อุปกรณ์ 2 อย่าง ถ้าพื้นที่ของ MEMORY ถูกออกแบบเป็นพิเศษ สำหรับการอ่าน หรือ การเขียน ASCII UNIT ก็สามารถ ติดต่อได้ โดยตรงต่อพื้นที่ทำการ MEMORY ไว้ (SPECIFIC PC MEMORY AREA)

จากรูป จะอธิบาย ความสัมพันธ์ PC PROGRAM และ ASCII UNIT PROGRAM

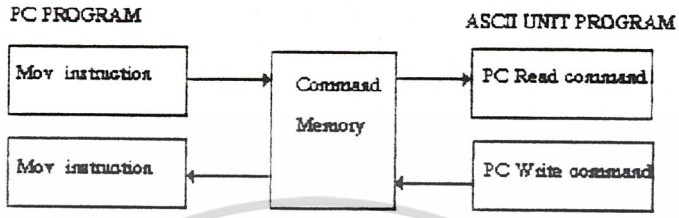


รูปที่ 2.4.1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จาก BLOCK DIAGRAM อธิบายถึง ความสัมพันธ์กันระหว่าง

PC DATA CODE และ ASCII UNIT PROGRAM



รูปที่ 2.4.2

การส่งผ่าน โปรแกรม (PROGRAM TRANSFER)

สำหรับ PC ที่จะทำการติดต่อกับ ASCII UNIT จะต้องมีการ SET SOFT WORD ที่จะทำการติดต่อ ดังนี้

BAUD RATE : เหมือนกับ ASCII UNIT

DATA LENGTH : 8 BIT

PARITY : NONE

NO STOP BIT : 2

ด้วยวิธี : FULL DUPLEX, NO ECHO , noXON/XOFF buffer busy control , no auto line feed.

: ตั้ง DIP SWITCHES ของ ASCII UNIT ตามรายการใช้งานที่ต้องการ

(อ้างถึงในส่วนที่ 1 ของการ SET DIP SWITCH)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การส่งผ่านโปรแกรม

ASCII UNIT BASIC PROGRAM จะต้องเขียน บน PERSONAL COMPUTER ที่ PERSONAL COMPUTER จะติดต่อกับ ASCII UNIT ทาง PORT โดยใช้ติดต่อตามมาตรฐาน RS 232C โปรแกรมสามารถถูกส่งผ่านไปยัง ASCII UNIT โดย PERSONAL COMPUTER หรือแหล่งเก็บข้อมูลอื่น ที่มี PORT การติดต่อสื่อสารแบบเดียวกัน โดยคำสั่ง LOAD ใน COMMAND ของภาษา BASIC โปรแกรม จะส่งผ่าน จาก ASCII UNIT EEROM ไปยัง ASCII UNIT RAM ที่ใช้โดยคำสั่ง LOAD

โปรแกรมจะถูกส่งจาก ASCII UNIT RAM ไปยัง EEPROM หรือ PERSONAL COMPUTER หรือแหล่งเก็บข้อมูลอื่น โดยการใช้การติดต่อสื่อสารที่ PORT1 เพื่อใช้สำหรับคำสั่ง SAVE

ASCII UNIT สามารถ เรียกใช้ (BOOSTER ON) POWER APPLICATION โดยโปรแกรมจะถูกเก็บอยู่ในตำแหน่ง ON ที่ด้านหลังของตัว ASCII UNIT

ในการส่งผ่านข้อมูล (DATA TRANSFER) จะเกิด OVERFLOW ถ้าความจุ BUFFER ของ BAUD RATE ระหว่าง COMPUTER และ ASCII UNIT ไม่ตรงกัน ถ้าเกิดปัญหาดังกล่าว จะต้องทำการ SET BAUD ที่โชว์ หรือ SET X ON (SPECIFY X ON) ด้วยคำสั่ง OPEN NOTE EEPROM มีอายุการใช้งาน เมื่อทำงานเขียน ลงประมาณ 5000 ครั้ง

การ RUN โปรแกรมภาษา BASIC

ASCII UNIT สามารถเก็บและเข้าถึง (STORE AND ACCESS) โปรแกรม ภาษา BASIC 3 วิธีแยกกัน แต่ละโปรแกรม สามาถทำงานโดยการ SET DIP SWITCH ที่ด้านหลังของ ASCII UNIT แต่จะต้องทำก่อนที่ ASCII UNIT จะเริ่มทำงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3 วิธีในการ RUN PROGRAM BASIC

1. กด RUN COMMAND จาก คีย์บอร์ด ของ PERSONAL COMPUTER โดยการกด CTRL + X โปรแกรมจะทำงาน (ABORT THE PROGRAM)

2. กด START/STOP บน ASCII UNIT มันจะเริ่ม START PROGRAM และเมื่อกด SWITCH อีครั้ง มันจะ STOP PROGRAM

3. ถ้า SWITCH ตำแหน่งที่ 1 (PIN1)ของ LEFT - SIDE DIP SWITCH ถูก SET ไว้ที่ตำแหน่ง ON โปรแกรมเฉพาะ (SPECIFIED PROGRAM) จะทำงานไปอย่างอัตโนมัติ เมื่อ ASCII UNIT TURN ON หรือเมื่อมันถูก RESET

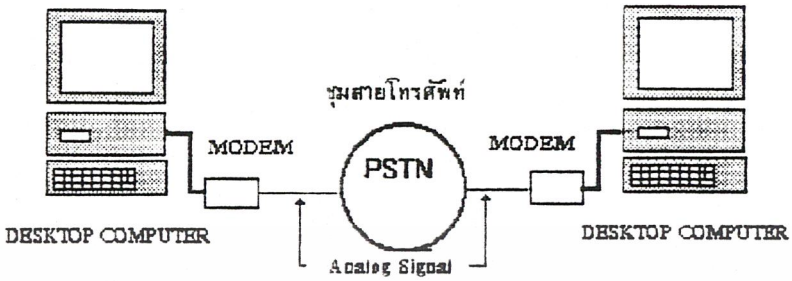
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2 โมเด็ม (Modem)

การส่งข้อมูลระหว่างเครื่องคอมพิวเตอร์ในระยะไม่ไกลนัก เราอาจใช้ในการรับส่งแบบอนุกรม (RS 232C) ซึ่งส่งข้อมูลดิจิตอล ของคอมพิวเตอร์ไปตามสาย จนถึงผู้รับได้ กรณีนี้เราสามารถรับส่งข้อมูลได้ไกลถึง 35 เมตร ตามคุณสมบัติของ RS232C หรือถ้าสายเคเบิลที่ใช้มีคุณภาพดี อาจส่งได้ไกลถึง 150 เมตร ที่ความเร็ว 9600 บิตต่อวินาทีสำหรับระยะทางที่ไกลมาก ๆ เช่น หลายสิบกิโลเมตร การส่งข้อมูลแบบดิจิตอล ออกไปโดยตรงจะไม่เหมาะสมหลายอย่าง ปัญหาที่สำคัญก็คือ คลื่นรูปสี่เหลี่ยมของสัญญาณดิจิตอลเมื่อส่งไปไกล ๆ จะเพี้ยนหรือ มีรูปร่างผิดเพี้ยนไป จากเดิมได้ง่ายทำให้สายส่งและวงจรรับส่งสัญญาณดิจิตอบต้องถูกออกแบบมาอย่างดีราคาของสายส่งสัญญาณแบบดิจิตอลจึงมีราคาแพงกว่าสายส่งสัญญาณแบบอะนาลอกมาก ในทางปฏิบัติเราอาจรับส่งข้อมูลระหว่างเครื่องคอมพิวเตอร์เครื่องโดยใช้สัญญาณดิจิตอล ผ่านสายส่งได้ ซึ่งทั้งสายส่งและวงจรเชื่อมต่อทั้งหมดเป็นแบบดิจิตอล แต่ว่าค่าใช้จ่ายจะมีราคาแพงจนกระทั่งไม่คุ้มค่า วิธีการที่จะหลีกเลี่ยงการทำเช่นนี้ โดยการส่งข้อมูลไปตามสายส่งในแบบอะนาลอกแทน การทำเช่นนี้เราจำเป็นต้องใช้อุปกรณ์ เข้าช่วย แปลงสัญญาณในการรับส่งข้อมูลทั้งสองด้าน ซึ่งเป็นที่มาของโมเด็มนั่นเอง

โมเด็ม (MODEM) จะทำหน้าที่แปลงสัญญาณ จากคอมพิวเตอร์ ที่ส่งมาทาง RS232C ให้กลายเป็นสัญญาณอะนาลอกแล้วส่งออกไปตามสายส่งกระบวนการนี้เรียกว่าการ Modulate สัญญาณ เมื่อถึงปลายทาง โมเด็มก็จะแปลงสัญญาณอะนาลอกที่ได้รับกลับมาสเป็นสัญญาณดิจิตอลแล้วส่งให้คอมพิวเตอร์ต่อในรูปของสัญญาณ ดิจิตอลผ่านทาง RS232C เช่นกัน กระบวนการแปลงสัญญาณกลับนี้เรียกว่า Demodulation ชื่อ ของโมเด็มก็ได้จากการ ทำงานทั้งสองแบบนั่นเอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.5 โมเด็มช่วยCOMPUTER รับส่งข้อมูลผ่านสายโทรศัพท์

โทรศัพท์ได้โดยเปลี่ยนสัญญาณให้เป็นเสียงก่อน

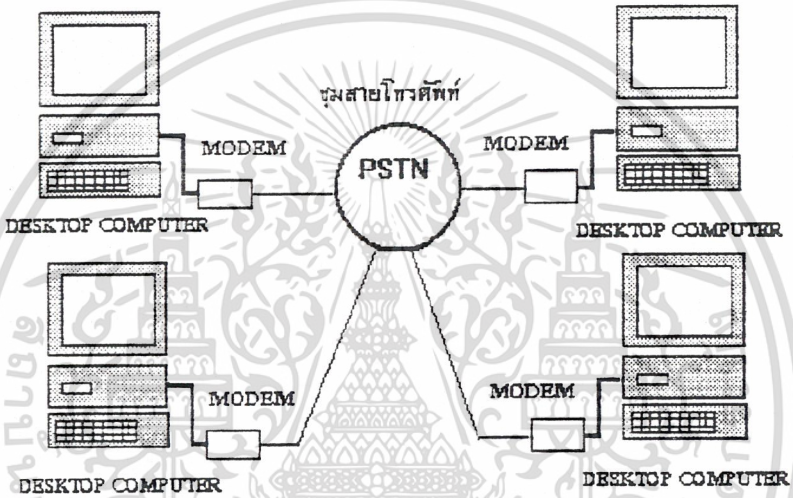
สัญญาณอะนาลอกมีคุณสมบัติเหมาะที่จะส่งไปไกล ๆ มากกว่าสัญญาณ แบบดิจิตอล เพราะว่าสัญญาณอะนาลอกจะเพี้ยนหรือมีรูปร่างผิดจากเดิมยากกว่า และสูญเสียกำลังในสายน้อยกว่าทำให้ส่งได้ระยะไกลมากขึ้น นอกจากนี้เรายังสามารถกรองสัญญาณรบกวนบางส่วนที่ไม่ต้องการ (FILTER) ออกได้อีกด้วย ราคาของสายส่งและอุปกรณ์เชื่อต่อก็มีราคาถูก เราจึงจำเป็นต้องใช้โมเด็มในการส่งข้อมูลบ คอมพิวเตอร์ระยะทางไกล ผ่านสายส่งอะนาลอก

ในการที่โมเด็มแปลสัญญาณจากคอมพิวเตอร์ให้กลายเป็นสัญญาณอะนาลอก ในการรับส่งข้อมูลนี้เอง ถ้าโมเด็มแปลงสัญญาณออกมาอยู่ในรูปของเสียงซึ่งเป็นสัญญาณอะนาลอกอยู่แบบหนึ่งเราก็สามารถรับส่งข้อมูลผ่านทางสายโทรศัพท์ได้ โมเด็มทั่ว ๆ ไป ที่เราใช้งานจะเป็นโมเด็มที่แปลงสัญญาณ จากคอมพิวเตอร์ให้อยู่ในรูปคลื่นเสียงทั้งนั้นมีโมเด็มบางชนิดที่แปลงสัญญาณดิจิตอลเป็นสัญญาณอะนาลอกความถี่สูง แต่โมเด็มแบบนี้มีใช้งานน้อยมากและส่งข้อมูลโดยใช้สายส่งพิเศษ จะส่งผ่านสายโทรศัพท์ธรรมดาไม่ได้ เราจึงเน้นเฉพาะโมเด็มที่ทำงานในย่านความถี่เสียงเท่านั้น ไม่ว่าโมเด็มจะเป็นแบบไหนก็ตาม เมื่อได้รับข้อมูลดิจิตอล จากคอมพิวเตอร์ มักจะเปลี่ยน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ให้กลายเป็นสัญญาณอะนาลอกจากนั้นก็นำสัญญาณ อะนาลอกได้นี้มารวมกับสัญญาณพาหะ (Carrier Wave) แล้วส่งออกไปทางสายส่งข้อมูลในรูปสัญญาณอะนาลอกไปจนถึงปลายทาง

โมเด็มแบบใช้กับสายโทรศัพท์ จะรับส่งข้อมูลด้วยความเร็วตั้งแต่ 300 BPS จนถึง 9600 BPS หรือมากกว่า เราสามารถรับส่งข้อมูลไปยังที่ต่าง ๆได้ตามต้องการ ตามจะหมายของเบอร์โทรศัพท์



รูปที่ 2.6 การรับส่งข้อมูลผ่านสายโทรศัพท์ เราสามารถรับส่งข้อมูลกับจุดต่าง ๆ ได้ หลายแห่งโดยหมุนเบอร์โทรศัพท์ ปลายทางผ่านชุมสาย

ความสะดวกเมื่อเราใช้สายโทรศัพท์ส่งข้อมูล ที่เห็นชัดก็คือสามารถส่งข้อมูลไปยังที่ต่าง ๆ ได้ตามต้องการโดยง่าย แต่ขีดจำกัดของ ระบบโทรศัพท์เป็นสิ่งที่หลีกเลี่ยงไม่ได้ คือ ช่วงความถี่ (BAND WIDTH) ที่แคบมากของสายโทรศัพท์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระบบการรับส่งของคนอยู่ในย่านความถี่ 30 Hz ถึง 20 Khz โดยประมาณ ความถี่สูงกว่านี้หรือต่ำกว่านี้เราจะไม่ได้ยิน เมื่อมีการคิดค้นระบบโทรศัพท์ขึ้นมาเทคโนโลยียังไม่สูงพอที่จะสร้างระบบโทรศัพท์ ให้รับส่งเสียงเทียบเท่าหูคนเราได้ยิน ระบบโทรศัพท์สามารถรับส่งสัญญาณเสียงได้นั้งแต่ ความถี่ 300 Hz จนถึงประมาณ 3400 Hz ซึ่งความถี่ช่วงดังกล่าวนี้มากพอสำหรับการพูดคุยของคนเราแต่ไม่มากพอที่จะใช้ในการรับส่งข้อมูล คอมพิวเตอร์ของในการรับส่งสัญญาณคอมพิวเตอร์ นั้น ยิ่งมี BAND WIDTH ยิ่งกว้างยิ่งรับส่งข้อมูลได้เร็วขึ้น

โมเด็มจำเป็นต้องมีมาตรฐานเช่นเดียวกับอุปกรณ์อื่น ๆ เพื่อให้ผู้ผลิตแต่ละบริษัทผลิตโมเด็มออกมาใช้รับส่งข้อมูลระหว่างกันได้ มาตรฐานของโมเด็มจะแบ่งออกเป็น 2 ส่วน คือ มาตรฐานในส่วนของฮาร์ดแวร์ที่โมเด็มใช้ และอีกส่วนหนึ่งคือมาตรฐานในส่วนของซอฟต์แวร์ที่ควบคุมการทำงานของโมเด็มหรือคำสั่งของโมเด็มในบทนี้จะกล่าวถึงมาตรฐานที่กำหนดขึ้นจากองค์การมาตรฐานสื่อสารสากล หรือ CCITT (INTERNATIONAL TALEPHONE AND TELEGRAPH CONSULTATIVE COMMITTED) ซึ่งมาตรฐานนี้จะครอบคลุมเกี่ยวกับความเร็วในการับส่งข้อมูล ความถี่ที่ใช้และเทคนิคการส่งสัญญาณในสาย

BIT RATE กับ BAUD RATE

ความเร็วในการรับส่งข้อมูลของโมเด็มนั้นกำหนดเป็นบิตต่อวินาที (BIT PER SECOND ; BPS) หรือ Bit Rate ซึ่งแตกต่างจากอัตราการเปลี่ยนแปลงของสัญญาณไฟฟ้าในสายส่งหรือที่เรียกกันกว่า Baud Rate

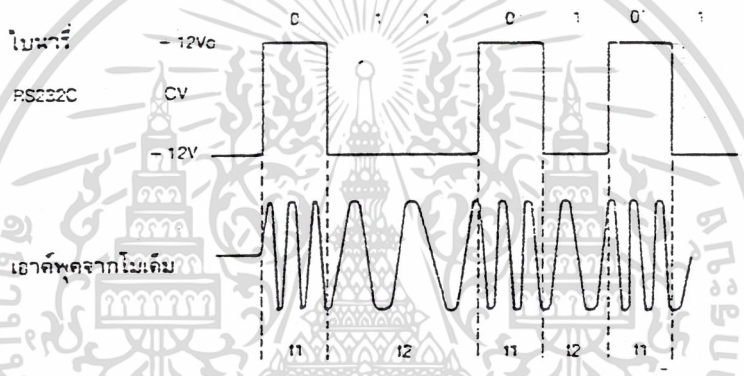
การผสมสัญญาณแบบ FSK และ PSK

นอกจากมาตรฐานจะกำหนดความเร็วในการับส่งข้อมูลให้ตรงกันแล้ว ยังกำหนดความถี่ของคลื่นพาหะที่ใช้ว่าผู้ส่งและผู้รับใช้ความถี่เท่าไร รวมทั้งมาตนฐานของการผสมสัญญาณอีกด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

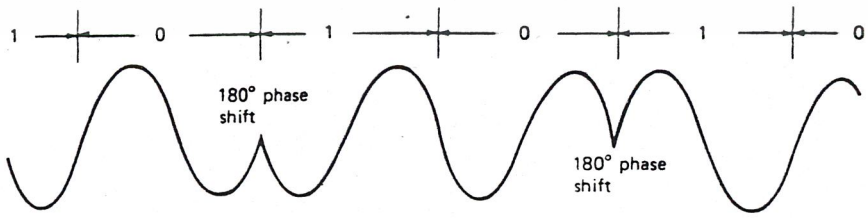
มาตรฐานการผสมสัญญาณที่ใช้กันมากในปัจจุบันคือ Frequency Shift Keying (FSK); Phase Shift Keying (PSK) และ QUADRATURE Amplitude Modulate (QAM)

1. Frequency Shift Keying นั้นเป็นระบบเก่าที่ใช้กับโมเด็ม ความเร็วต่ำ โดยแทน ' 0 ' และ '1' ด้วยความถี่ต่างกันผู้ส่งจะใช้ความถี่สองความถี่แทน '0' และ '1' ของด้านส่ง ส่วนผู้รับก็ใช้ความถี่อีกสองความถี่ แทน '0' และ '1' ของด้านรับเช่นกัน ทั้งหมดจึงใช้ความถี่รวม 4 ความถี่ด้วยกัน ความเร็วสูงสุดที่ได้จากการผสมสัญญาณแบบนี้คือ 1200 บิตต่อวินาที ซึ่งการผสมสัญญาณ แบบ FSK นี้ Bit Rate จะเท่ากับ BandRate เสมอ



รูปที่ 2.7 การผสมสัญญาณแบบ FSK.

2. Phase Shift Keying (PSK) นั้นใช้หลักการที่ว่าแทนข้อมูล '0' และ '1' ด้วยการเปลี่ยนแปลงมุมของช่วงส่งสัญญาณในสายส่ง (Phase) เช่นโดยเราอาจกำหนดว่า '0' แทนด้วยมุมต่อเนื่องกันไปเลย '1' แทนด้านมุมที่เปลี่ยนไปจากเดิม 180 องศา ถ้าเราส่งข้อมูล 010110 รูปร่างคลื่นในสายส่งจะเป็นดังนี้



รูปที่ 2.8 การผสมสัญญาณแบบ PSK

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. Quadrarture Amplitude Madulation (QAM)

เป็นการผสมสัญญาณที่ใช้ทั้งการเปลี่ยน Phase และขนาดของสัญญาณควบคู่กันไปสำหรับใช้กับโมเด็มความเร็วสูงซึ่งถ้าใช้การเปลี่ยน Phase เพียงอย่างเดียว มุมที่เปลี่ยนแปลงจะมีค่าน้อยเกินไปทำให้เกิดข้อผิดพลาดได้ถ้าเราใช้การเปลี่ยน Phase และขนาดของสัญญาณ ค่าของมุมจะอยู่ห่างกันมากขึ้น ดังนั้นความเร็วของการผสมสัญญาณ แบบนี้จึงมากขึ้น และเป็นที่นิยม

นอกจากนี้ยังมีการผสมสัญญาณสำหรับโมเด็มความเร็วสูงปลาย และที่กำลังมีการทดลองอยู่ แต่ยังไม่ได้รับการยอมรับให้เป็นมาตนฐานในปัจจุบัน

มาตรฐานของโมเด็มตาม CCITT V- Series

มาตรฐาน ของโมเด็มที่เราใช้อยู่เป็นไปตามที่องค์การมาตรฐานสื่อสารสากล หรือ CCITT เป็นผู้กำหนดขึ้น โดยมีชื่อเรียกมาตรฐานเหล่านี้ว่า V - Series ดังนี้

มาตรฐานโมเด็มของ CCITT

Series Number	Line Speed	Channel Separation	FDX or HDX	Modulation Technique	Switch Lines	Leased Lines
V.21	300	FD	FDX	FSK	Yes	0
V.22	1200	FD	FDX	TSK	Yes	PP2W
V.22	600	FD	FDX	PSK	Yes	PP2W
V.22 bis	2400	FD	FDX	QAM	Yes	PP2W
V.22 bis	1200	FD	FDX	QAM	Yes	PP2W
V.23	600	NA	HDX	FMK	Yes	0
V.23	1200	NA	HDX	FMK	Yes	0
V.26	2400	4-Wire	FDX	PSK	No	PP MP4W
V.26 bis	2400	NA	HDX	PSK	Yes	No
V.26 bis	1200	NA	HDX	PSK	Yes	No
V.26 ter	2400	EC	Either	PSK	Yes	PP 2W
V.26 ter	1200	EC	Either	PSK	Yes	PP 2W
V.27	4800	ND	Either	PSK	No	Yes
V.27 bis	4800	4-Wire	Either	PSK	No	2W 4W
V.27 bis	2400	4-Wire	Either	PSK	No	2W 4W
V.27 ter	4800	None	HDX	PSK	Yes	No
V.27 ter	2400	None	HDX	PSK	Yes	No
V.29	9600	4-Wire	Either	QAM	No	PP 4W
V.29	7200	4-Wire	Either	PSK	No	PP 4W
V.29	4800	4-Wire	Either	PSK	No	PP 4W
V.32	9600	EC	FDX	QAM	Yes	PP 2W
V.32	9600	EC	FDX	TCM	Yes	PP 2W
V.32	4800	EC	FDX	QAM	Yes	PP 2W
V.32 bis	14,400	EC	FDX	TCM	Yes	PP 2W
V.32 terbo	19,200	EC	FDX	TCM	Yes	PP 2W
V.fast	28,800	EC	FDX	TCM	Yes	PP 2W

* V.32 terbo is UNOFFICIAL STANDARD

ND = not defined
NA = not applicable
EC = echo cancellation
FD = frequency division
FDX = full duplex
HDX = half duplex
PP = point to point
MP = multipoint
FS = for further study

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

มาตรฐานคำสั่งโมเด็ม

ในยุคแรกการทำงานของโมเด็มต้องใช้คนเข้าช่วยเหลือเกือบทุกอย่างเนื่องจากโมเด็มทำหน้าที่รับส่งสัญญาณและเปลี่ยนแปลงสัญญาณที่ได้รับจากเครื่องคอมพิวเตอร์เท่านั้นเอง การหมุนโทรศัพท์เพื่อติดต่อกับคอมพิวเตอร์อีกจุดต้องใช้คนทั้งสิ้น ซึ่งเห็นได้ว่ายุ่งยากมาก

ในปัจจุบัน การสั่งงานโมเด็มด้วยคำสั่งจึงเกิดขึ้น ปกติโมเด็มจะติดต่อกับเครื่องคอมพิวเตอร์เพื่อทำหน้าที่รับส่งข้อมูลอยู่แล้ว ดังนี้ขณะที่โมเด็มยังไม่ได้ติดต่อกับปลายทางเพื่อส่งข้อมูลแต่อย่างใดจนกว่าจะติดต่อกับโมเด็มปลายทางได้

บริษัท Hayes Microcomputer products Inc เป็นผู้คิดค้นชุดคำสั่งชุดหนึ่งขึ้นมาเพื่อสั่งงานโมเด็ม และได้รับความนิยมจนถือเป็นมาตรฐาน เรียกว่า มาตรฐานคำสั่ง Hayes Command Set หรืออีกชื่อหนึ่งเรียกว่า AT Command ซึ่งคำสั่งต่าง ๆ มีดังนี้

- ATA เป็นคำสั่งให้โมเด็มตอบรับสัญญาณโทรศัพท์ที่เรียกเข้ามา
- ATB เป็นคำสั่งเลือกการทำงานของโมเด็มว่าจะใช้การผสมสัญญาณตาม CCITT หรือ Bell Standard โดยใช้ ATB และ ATB1 1 ตามลำดับ
- ATD เป็นคำสั่งให้โมเด็มหมุนโทรศัพท์ มี 2 แบบ คือ
  ATADT ตามด้วยเบอร์โทรศัพท์ เป็น การหมุนโทรศัพท์ระบบ Tone
  ATADP ตามด้วยเบอร์โทรศัพท์ เป็น การหมุนโทรศัพท์ระบบ Pulse
- ATE สั่งให้โมเด็ม Echo ข้อความกลับไปให้เครื่องคอมพิวเตอร์ โดย ATEO คือ ไม่ Echo และ ATE1 ให้ Echo
- ATH เป็นคำสั่งให้โมเด็ม วางสายโทรศัพท์
- ATL สำหรับปรับความดังของลำโพง ภายใน โมเด็ม ซึ่งเป็น ได้ 3 ระดับ ATLO, ATL2, ATL3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ATM ใช้สำหรับควบคุมการทำงานของลำโพง ATMO คือไม่ให้ลำโพงทำงาน ATM1 คือให้มีเสียงตอนที่ยังต่อกับปลายทางไม่ได้
- ATO เป็นคำสั่งให้โมเด็มอยู่ในโหมด Online คือเปลี่ยนจากการรับคำสั่งจากเครื่องเป็นการรับส่งข้อมูลผ่านสายส่ง
- ATV เป็นคำสั่งบอกให้โมเด็มแสดงผลลัพธ์ของคำสั่งต่าง ๆ เป็นรหัสตัวเลข (Digit code)
- ATX เป็นคำสั่งให้โมเด็มแสดงผลลัพธ์ในการหมุนโทรศัพท์แสดงค่าต่าง ๆ เช่น ATX0 และ ATX1 จะ

เป็นคำสั่งให้โมเด็มไม่ตรวจสอบ Dial Tone ก่อนหมุนโทรศัพท์ ATX2 ให้ตรวจสอบ Dial Tone ก่อน ATX3 ให้โมเด็มรับรู้ Busy Tone ส่วน ATX4 คือ ATX2 รวมกับ ATX3

- ATZ เป็นคำสั่งที่เซ็ตโมเด็มให้อยู่ในสภาวะปกติ
- ATX1D เป็นคำสั่งที่ให้โมเด็มส่งสัญญาณเมื่อใช้กับ ลีดสายน์ ในขณะที่โมเด็มปลายทางต้องเรียกคำสั่ง ATA
- ATS ใช้สำหรับเปลี่ยนแปลงค่าของ S - Register ซึ่งเป็นที่เก็บพารามิเตอร์ในการทำงานต่าง ๆ ของ Hayes ซึ่งค่าต่าง ๆ ใน S - Register มีดังนี้

Smartmodem S-Register Set

Register	Description	Units	Range	Default	
S0	Select ring to answer on	Rings	0-255	0	
S1	Ring count (incremented with each ring)	Rings	0-255	0	
S2	Escape-sequence character	ASCII	0-127	43(+)	
S3	Carriage-return character	ASCII	0-127	13	
S4	Line-feed character	ASCII	0-127	10	
S5	Backspace character	ASCII	0-32,127	8	
S6	Wait-time before blind dialing	Seconds	2-255	2	
S7	Wait-time for carrier/dial tone	Seconds	1-255	30	
S8	Length of pause for comma in number	Seconds	0-255	2	
S9	Carrier-detect response time	.1 second	1-255	6	
S10	Delay between carrier loss/hangup	.1 second	1-255	7/14	
S11	Duration/spacing of DTMF tones	.001 second	50-255	70/195	
S12	Escape-sequence guard time	.02 second	20-255	50	
S16	Test mode				

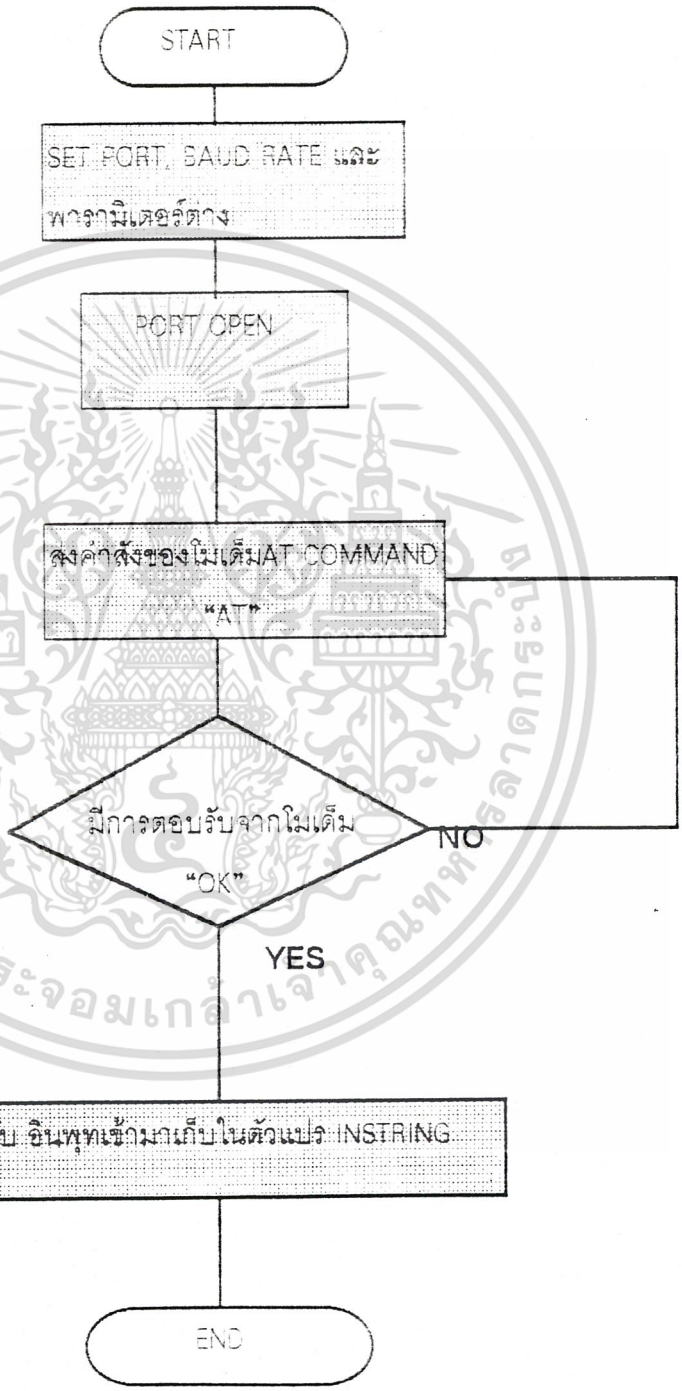
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวอย่าง โปรแกรมที่ใช้ติดต่อกับโมเด็ม โดยใช้ VISUAL BASIC FOR WINDOWS VERTION 3.00

```
Sub Form_Load ()
        ' Use COM1.
        Comm1.CommPort = 1
        ' 9600 baud, no parity, 8 data, and 1 stop bit.
        Comm1.Settings = "9600,N,8,1"
        ' Tell the control to read entire buffer when Input is used.
        Comm1.InputLen = 0
        ' Open the port.
        Comm1.PortOpen = True
        ' Send the attention command to the modem.
        Comm1.Output = "AT" + Chr$(13)
        ' Wait for data to come back to the serial port.
        Do
                Dummy = DoEvents()
        Loop Until Comm1.InBufferCount >= 2
        ' Read the "OK" response data in the serial port.
        InString$ = Comm1.Input
        ' Close the serial port.
        Comm1.PortOpen = False
End Sub
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

FLOW CHART แสดงการทำงานของโปรแกรม



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3

การสร้างและประกอบโครงงาน

ในโครงงานนี้แบ่งออกเป็น 2 ส่วน คือ

1 HARD WARE ซึ่งได้แก่ WARE HOUSE MODEL

- ELECTRICAL HARDWARE
- MECHANICAL HARDWARE

2 SOFT WARE มีอยู่ด้วยกัน 3 ส่วนคือ

- LADDER PROGRAM
- ASCII BASIC PROGRAM
- COMMUNICATION MODEM และ GRAPHICS DISPLAY

ซึ่งจะกล่าวดังต่อไปนี้

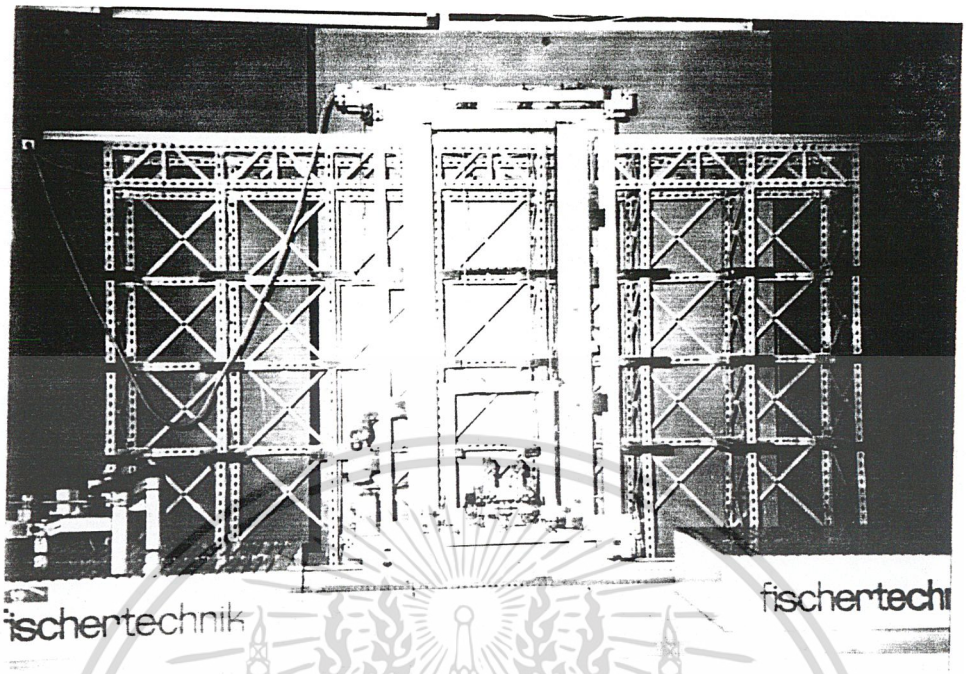
3.1 HARD WARE

- ELECTRICAL HARDWARE คือตัว PLC รุ่น C 200 H ซึ่งรายละเอียดกล่าวในบทที่ 2 แล้ว ในส่วนของ CONFIGULATION , I/O ASSIGNMENT , I/O TABLE ชนิดและประเภทของ I/P , O/P, จะกล่าวละเอียดในภาคผนวก

- MECHANICAL HARDWARE คือ WARE HOUSE ตามการใช้งานจริง จะเปรียบเสมือน เป็นที่เก็บ ของและวัสดุต่าง ๆ การเก็บจะเก็บลักษณะ การเรียงขึ้นตามแนวระดับในห้องต่าง ๆ ซึ่งจะทำให้ง่าย และสะดวกต่อการนำของและวัสดุเข้าไปเก็บ หรือนำออกมาใช้งาน

จากโครงงานนี้ได้ใช้ WARE HOUSE จำลอง ซึ่งจะมี จำนวน 3 ชั้น และ 7 หลัก จะมีสายพานลำเลียง คอยส่งวัสดุเข้าออก และมีคานลำเลียงคอยรับวัสดุเข้าเก็บตามชั้นและหลักต่าง ๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูป 3.1

ระบบขับเครื่อง

ระบบการขับเคลื่อนได้ใช้การขับเคลื่อนโดย PC MOTOR ส่งกำลังผ่านเฟืองและลูกรอก ทำให้สายพานและคานลำเลียงเคลื่อนที่ได้

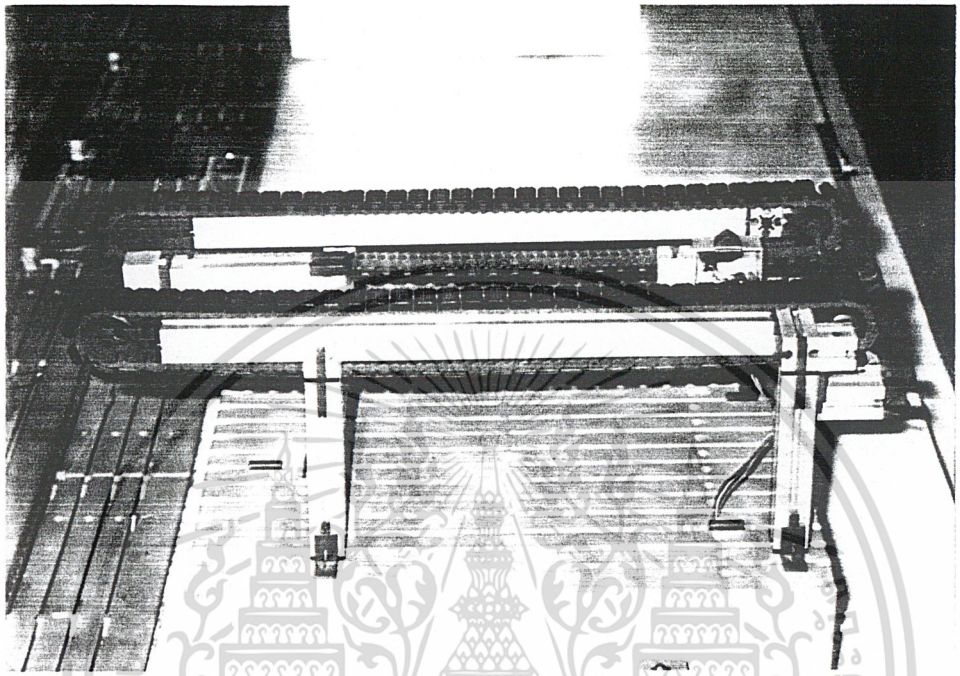
อุปกรณ์ในการขับเคลื่อน

1 สายพานลำเลียง

สายพานลำเลียงจะทำหน้าที่ลำเลียงวัสดุเข้าสู่ถาดลำเลียงโดยที่ตัวยสายพานลำเลียงจะมี LIMIT SW Z1 คอยตรวจสอบเมื่อมีวัสดุเข้าสู่สายพานลำเลียง และ PROXIMERY SW Z7 เป็น SWITEY คอยตรวจสอบเมื่อวัสดุมาอยู่ตำแหน่งสุดสายพานลำเลียง PROXIMIRY SWITCH Z7 จะ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เป็น SW IN PUT ของ PLC คอยสั่งงานให้ ถาดลำเลียง ยื่นมารับวัสดุในลำดับต่อไป การขับเคลื่อนให้สายพานลำเลียงเคลื่อนที่ จะขับด้วย moter m1 ผ่านเฟือง



รูปที่ 3.2

2 การลำเลียง

การลำเลียง จะเป็นตัวคอยส่งวัสดุเข้าสู่ตามชั้นและหลักต่าง ๆ โดยมีอุปกรณ์ ลำเลียงแบ่งเป็นส่วนประกอบทั้งหมด 3 ส่วนคือ

2.1 ถาดลำเลียงด้านแนวแกน Z

เมื่อวัสดุมาถึงสุดสายพานลำเลียง ถาดลำเลียงจะยื่นมารับวัสดุจากสายพานลำเลียงโดยมี Limit sw z3 z4 และ z5 คอยตรวจสอบตำแหน่ง ของถาดลำเลียง 3 ตำแหน่งคือ

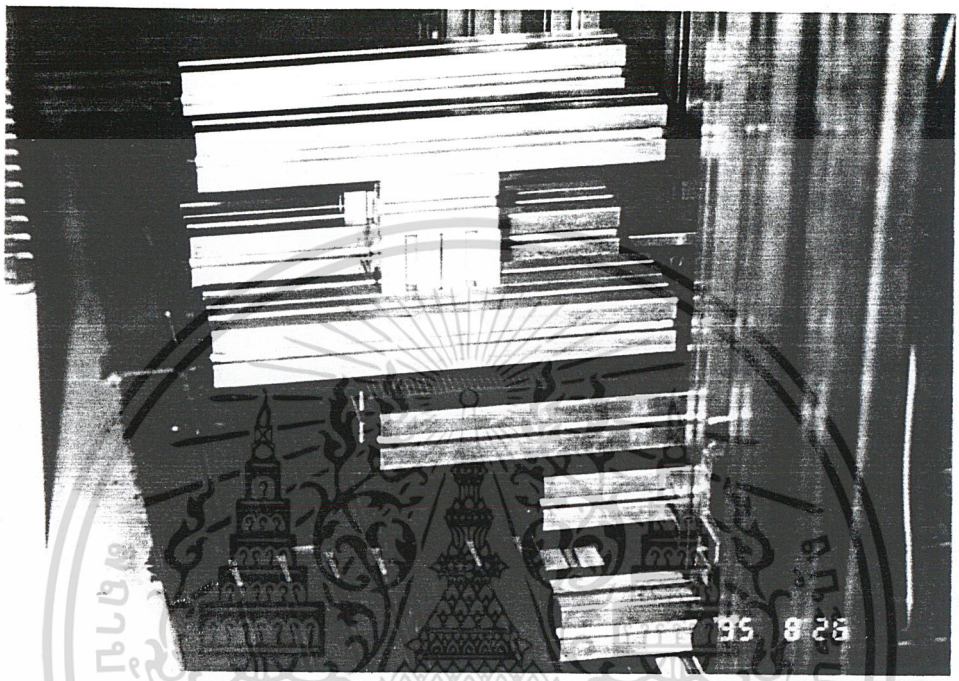
1. ยื่นออกมาสุด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. อยู่ในตำแหน่งตรงกลาง

3. เข้าไปสุดเพื่อส่งวัสดุ

การขับเคลื่อนให้ถาดลำเลียงเคลื่อนที่จะขับเคลื่อนด้วย motor mz ผ่านเฟืองตรง

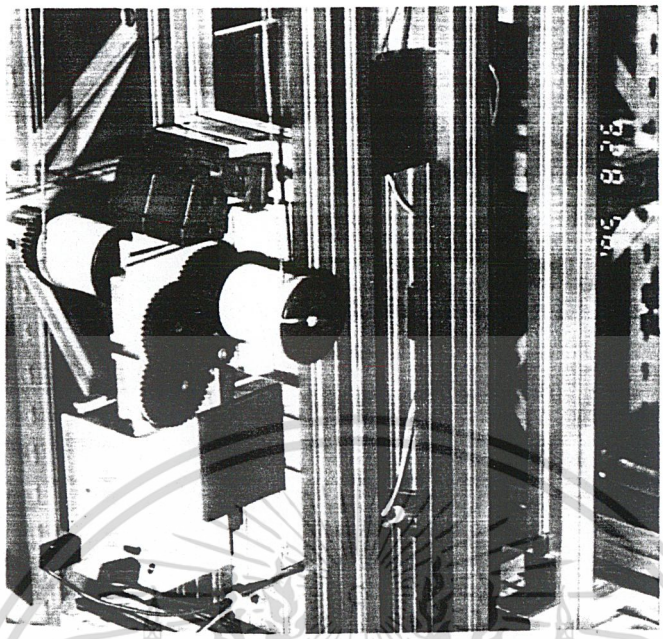


รูปที่ 3.3 แสดงถาดลำเลียง และทิศทางการเคลื่อนที่

2.2 รอกยกระดับด้านแกน Y

รอกยกระดับด้านแกน Y จะทำหน้าที่ยกถาดลำเลียงขึ้นและลงขึ้นสู่ตามชั้นต่าง ๆ โดยมี Limit sw ด้าน UP และด้าน DOWN คอยตรวจสอบคำแหน่งในการหาวัสดุรอกยกระดับจะถูกขับเคลื่อนด้วย motor m3 ผ่านระบบเฟืองและรอกอีกที

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

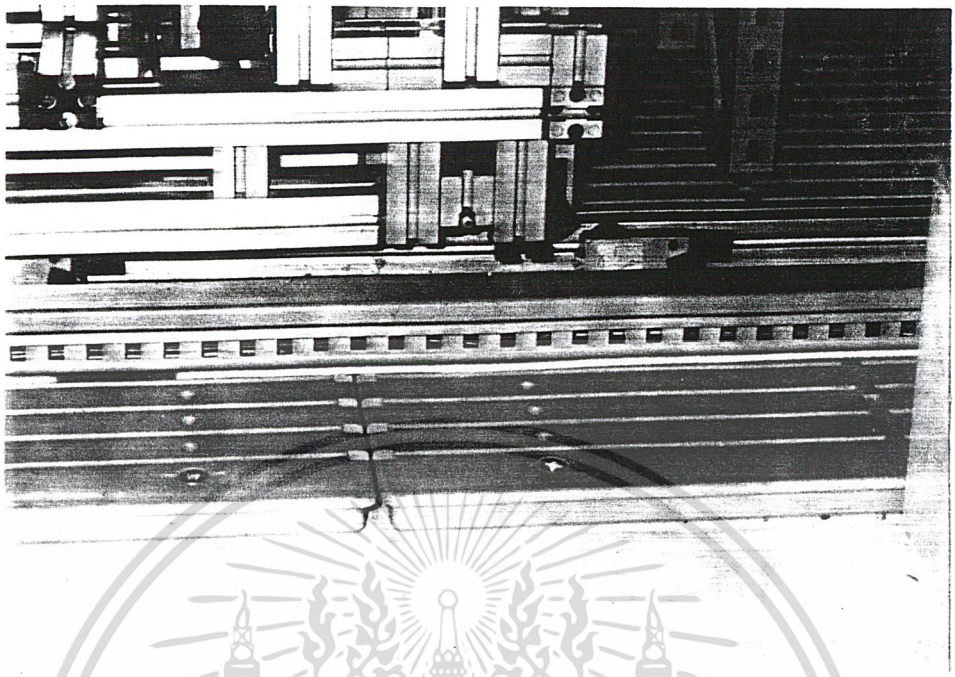


รูปที่ 3.4

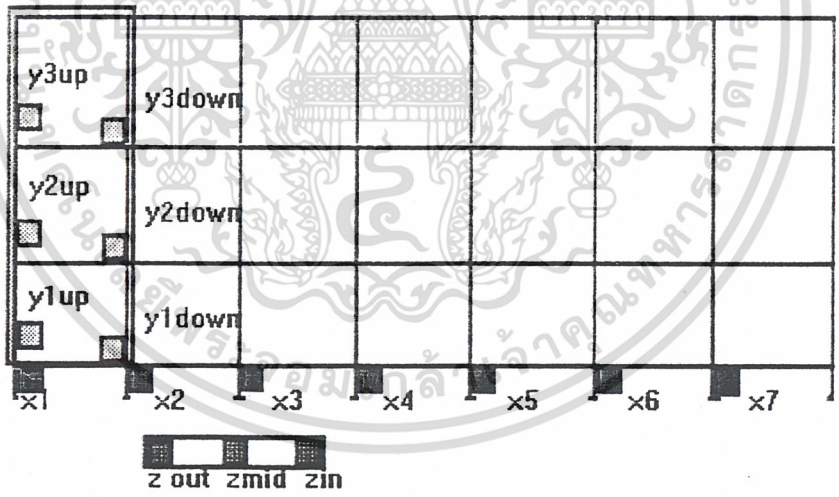
2.3 คานเลื่อนตามแนวแกน X

คานเลื่อนตัวนี้จะคอยเลื่อนตามถาดลำเลียง และรอกระดับซึ่งติดอยู่ด้วยกัน ให้เคลื่อนที่ไปตามแนวแกน X โดยมี Limit sw ด้านล่างคอยตรวจสอบตำแหน่งตามแนวแกน X คานเลื่อนเคลื่อนที่ได้โดยมี motor m4 ขับเคลื่อนผ่าน เฟืองทดและลงสู่ลูกล้อ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.5 แสดง การเคลื่อนที่ตามแนวแกน X และตำแหน่ง Limit switch ต่าง ๆ



รูปที่ 3.6 แสดงตำแหน่งของ LIMIT SWITCHS ตามตำแหน่งต่าง

SW ตรวจจับตำแหน่งต่าง ๆ ของ WARE HOUSE จะเป็น INPUT เพื่อส่งสถานะ ON และ OFF ให้แก่ PLC สามารถสรุป INPUT ที่เป็น limit switch และ Proximity switch ได้ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

X1 - X7 คอยตรวจจับตำแหน่งของคานเลื่อนให้ตรงตามหลักต่าง ๆ ทั้ง 7 หลัก

Z1 คอยตรวจจับเมื่อมี วัสดุ เข้าสู่สายพานลำเลียง

Z2, Z3, Z4 คอยตรวจสอบตำแหน่งของถาดลำเลียง 3 ตำแหน่ง คือ out, mid, in

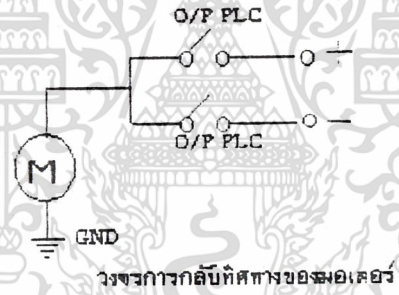
Y1 - Y3 Down คอยตรวจสอบตำแหน่งของถาดลำเลียงด้านต่ำ

Y1 - Y3 Up คอยตรวจสอบตำแหน่ง ของถาดลำเลียงด้านสูง

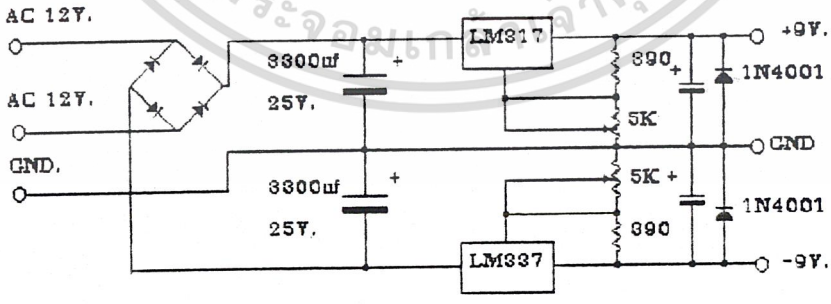
ระบบการขับเคลื่อน motor

motor ทั้ง 4 ตัว เป็นมอเตอร์กระแสตรง ขนาดเล็ก ทนแรงดันได้ไม่เกิน 9 โวลท์ แรงดันปกติ 5 โวลท์ ถึง 7 โวลท์

ในการทำงานมอเตอร์จะต้อง หมุน 2 ทิศทาง การขับเคลื่อนใช้แรงดันให้คงที่ บวกและลบ เพื่อเปลี่ยนทิศทางการหมุนของ motor



วงจรการกลับทิศทางของมอเตอร์



รูปที่ 3.7 แสดงภาคจ่ายไฟแรงดันคงที่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2 SOFT WARE3.2.1 LADDER PROGRAM

การควบคุมแบ่งออกเป็น 4 MODE ด้วยกันคือ

1 MODE รับ

2 MODE ย้าย

3 MODE ส่งของ

4 MODE AUTO (MODE นี้เป็นเพียง MODE ที่แสดงให้เห็นทั้งการเคลื่อนที่ใน MODE ย้าย เพื่อแสดงให้เป็นกราฟฟิกที่ personal computer ได้ชัดเจนขึ้น)

ดัง Flow Chart ที่ได้แสดงไปแล้วจะเห็นได้ถึงการ Control Hare House ด้วย PLC เพียงอย่างเดียวแต่ยังไม่ได้กล่าวถึงการดึงเอา ASCII UNIT เข้าไป READ และ WRITE memory ของ PLC ซึ่งเราจะขอกล่าวไปในทีละส่วนก่อน ซึ่งจุดนี้ขออธิบายการทำงานของแต่ละ MODE ให้เข้าใจเสียก่อน (ดู Flow Chart ประกอบ)

- MODE รับ

ตามโครงงานแล้ว เราจะนำเอาวัสดุที่จะทำการส่งไปวางบนสายพานเมื่อวัสดุที่จะทำการส่งวางบน SenSor ที่ทำการดักอยู่บนสายพานก็จะให้สายพานเลื่อนวัสดุนั้นไปจนสุดสายพาน ซึ่งมี Proximity ดักอยู่ เมื่อวัสดุไปตัดกับ Prox ทำให้สายพานอยู่พร้อมกับเลื่อนเอาลิฟท์วัสดุมารับเอาวัสดุนั้นไปจากสายพาน สั่งตามขั้นต่าง ๆที่ได้ป้อนไว้แล้วในตอนต้น เมื่อวัสดุส่งเสร็จเราก็ทำการ RESET เพื่อให้อยู่ในตำแหน่งเริ่มต้นใหม่ เพื่อรับค่าใหม่ต่อไป

- MODE ส่ง

ใน MODE นี้จะอยู่ระหว่างขั้นต่าง ของ WARE HOUSE เมื่อเราป้อนค่าเพื่อให้ลิฟท์รับของไปรับของที่ชั้นนั้น ๆ ลิฟท์รับของเมื่อได้รับค่าแล้วก็จะทำการประมวลผลและไปรับของชั้นนั้น ๆ ไปส่งที่แท่งรับของ และจะอยู่ที่แท่งรับของนั้นจนกว่าจะทำการ RESET ใหม่

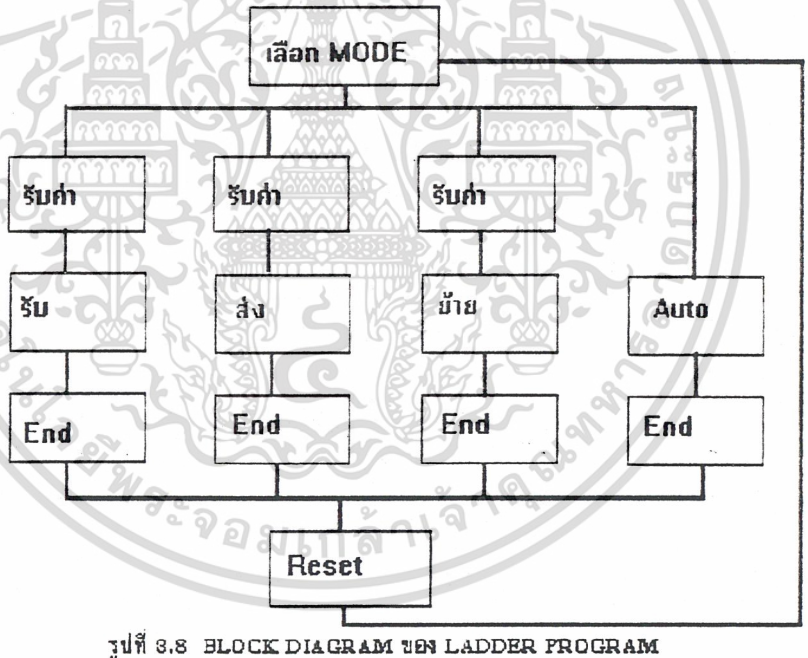
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- MODE ย้าย

MODE นี้จะมีการป้อนค่าถึง 4 ค่า คือ ป้อนค่าของชั้นและแถว เพื่อจะให้ลิฟท์รับของไปรับของชั้นนั้น ๆ เพื่อไปส่งยังชั้นและแถวของชั้นที่จะลงของ คือกล่าวอย่างง่าย ๆ คือ 2 ค่าแรกคือชั้นและแถวที่ลิฟท์จะไปรับของ 2 ค่าหลังคือ ชั้นและแถวที่ลิฟท์ขะไปส่งของ ซึ่งใน MODE นี้จะต้องมีFUNCTION COMPAREเข้ามาเกี่ยวข้องด้วย(ซึ่งศึกษาได้จาก LADDER DIAGRAM ที่ได้ให้มาในโครงงานในเลมนี้) เมื่อทำจนเสร็จสิ้นกระบวนการแล้วที่ทำการ RESET เพื่อกับสู่ตำแหน่งเดิม เพื่อรับค่าที่ทำการย้ายต่อไป

- MODE AUTO

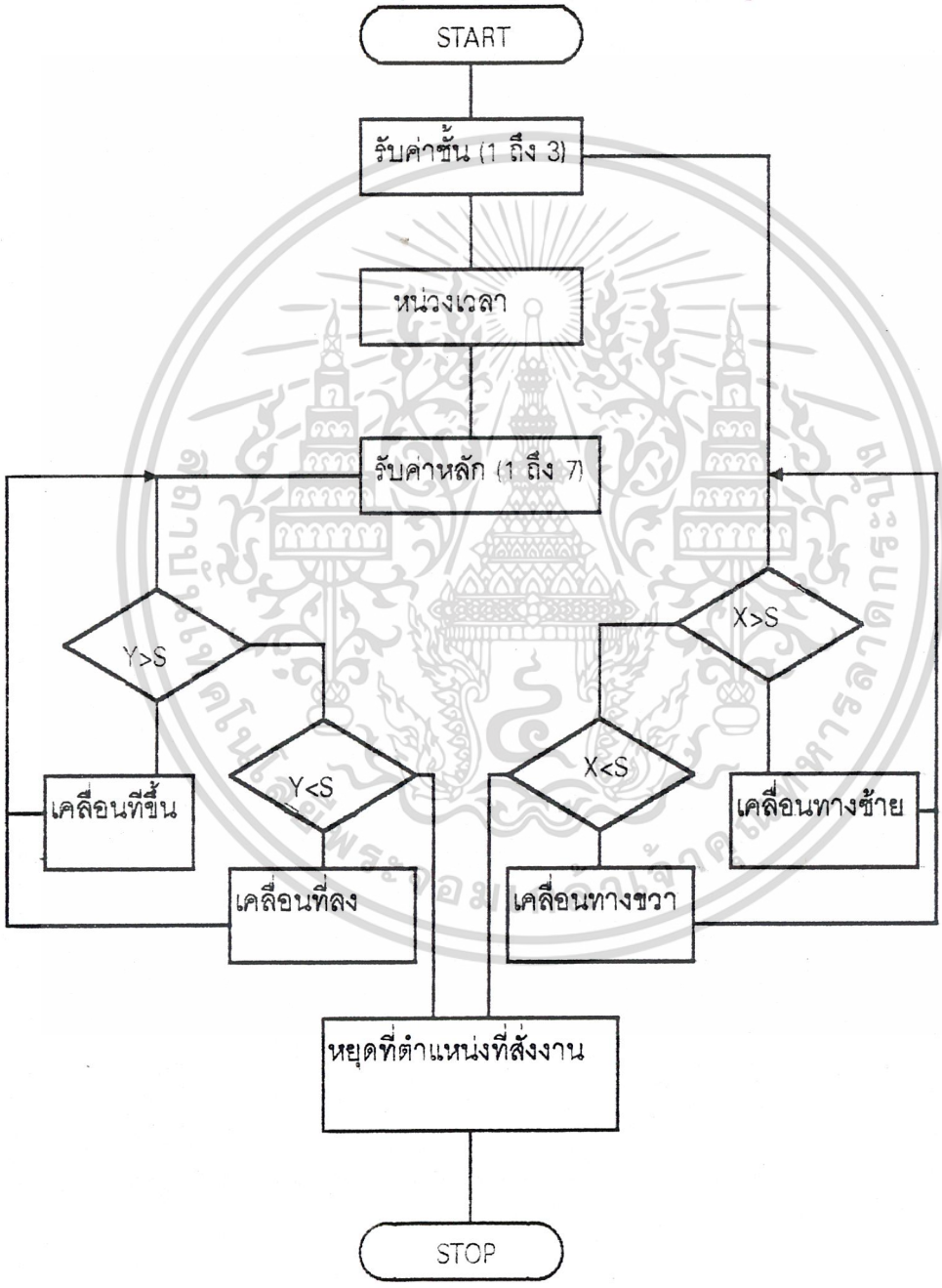
ใน MODE นี้อาจไปสำคัญในโครงงานนี้แต่ทำไว้เพื่อแสดงให้เห็นถึงการเคลื่อนที่ของลิฟท์ใน mode ย้าย เพื่อแสดงให้เห็น กราฟฟิกที่จอ computer ได้ชัดเจนขึ้น



รูปที่ 3.8 BLOCK DIAGRAM ของ LADDER PROGRAM

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวอย่าง FLOW CHART แสดงการทำงานในโหมดรับคือรับค่าจาก KEY BOARD เป็นตำแหน่งของชั้นและหลักที่ต้องการนำของเข้าไปเก็บ (Y คือ ตำแหน่งชั้นปัจจุบัน ,X คือ ตำแหน่งหลักปัจจุบัน, S คือ ค่าที่รับจากการสั่งงาน)



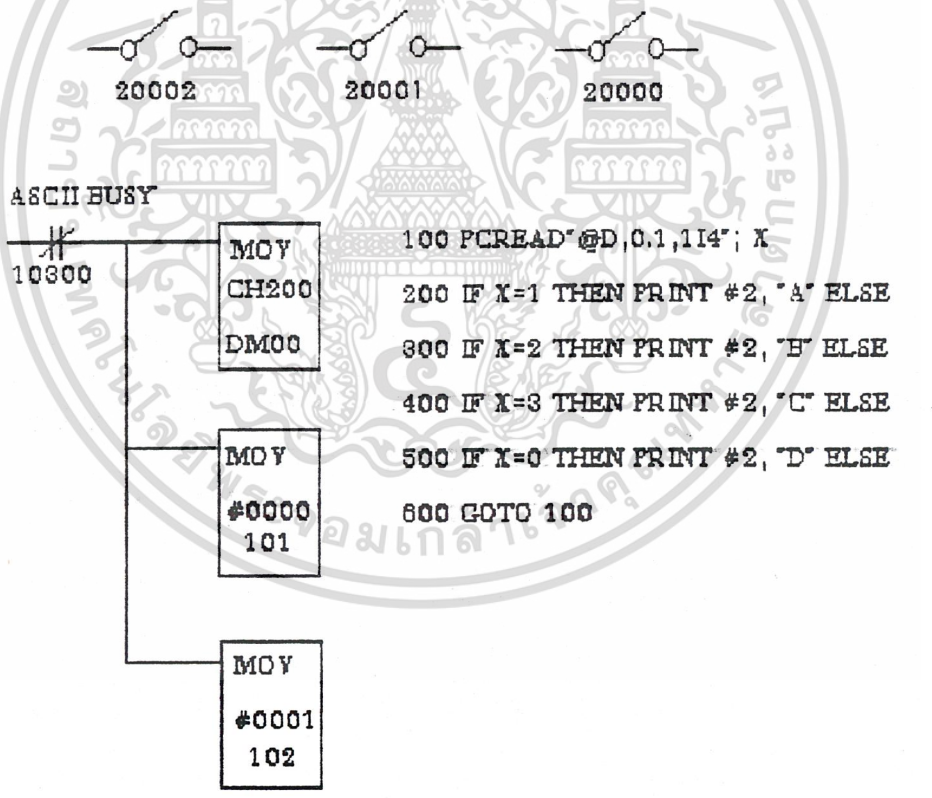
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.2 ASCII BASIC PROGRAM

เป็นโปรแกรมส่วนที่ทำหน้าที่เรียกค่าต่าง ๆ จากใน DATAMEMORY(DM) หรือ ป้อนค่าต่าง ๆ ลงใน DATA MEMORY เพื่อที่จะจัดการส่งออกยัง RS232C ในลักษณะของ ASCII CODE ไปยัง โมเด็ม เพื่อแสดงสถานะการทำงานและรับค่ากลับมาเพื่อควบคุมการทำงาน ของระบบ

ในการเขียนโปรแกรมนั้น BASIC PROGRAM จนต้องสัมพันธ์กับ LADDER PROGRAM ด้วย จากตัวอย่างดังต่อไปนี้ เมื่อเราต้องการจะนำสถานะการเปลี่ยนแปลงของ CH200 ซึ่งเป็น อินพุทจากสวิทช์ ส่งค่าออกไปยังปลายทางด้วย ASCII CODE ของคอมพิวเตอร์

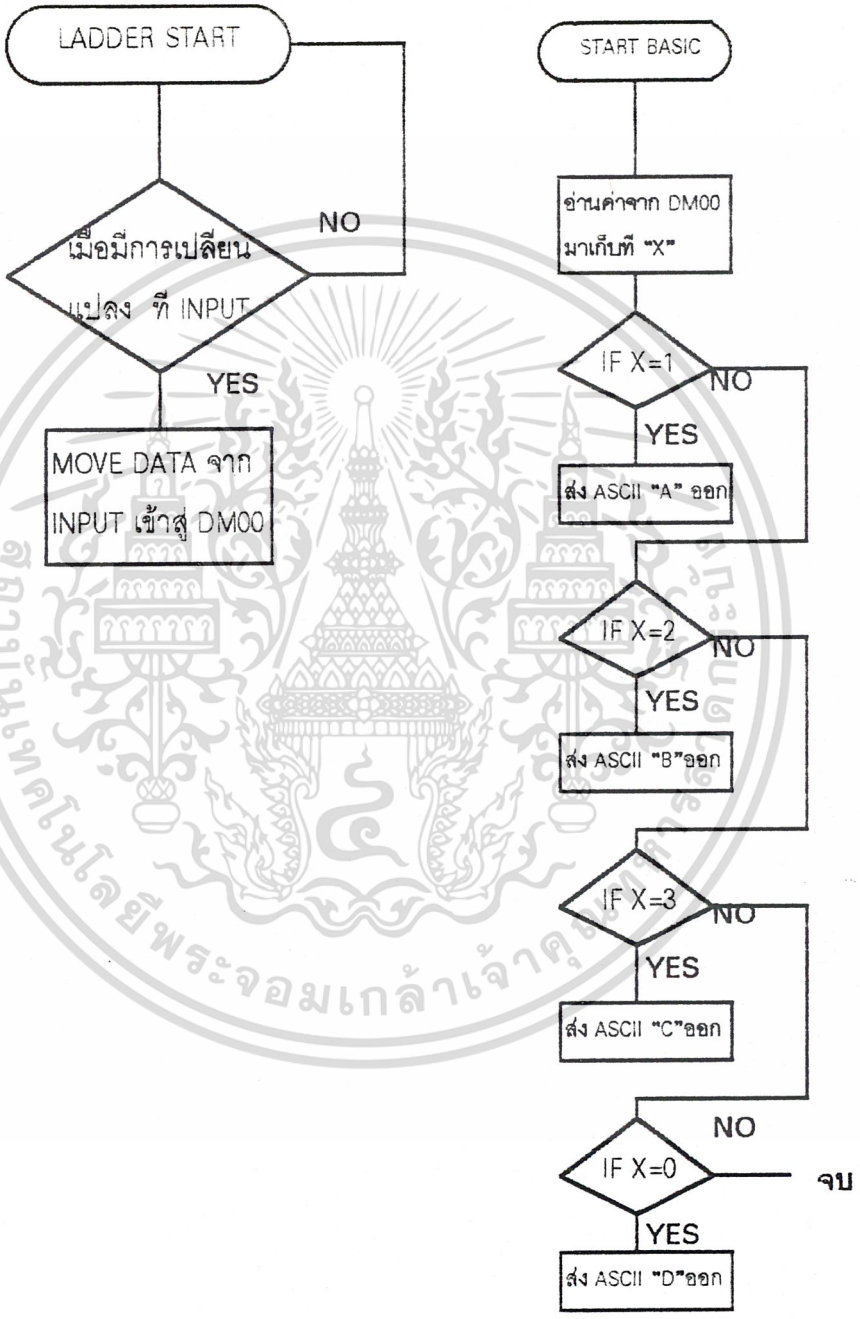
(สมมุติให้ SWITCH 20002,20001,20000 คือ LIMIT SWITCH ที่ใช้ควบคุมการทำงานของมอเตอร์แกน Z ซึ่งการ เปิดปิดของ LIMIT SWITCH จะเปลี่ยนแปลงตาม ตำแหน่งของการสั่งงาน)



รูปที่ 3.9

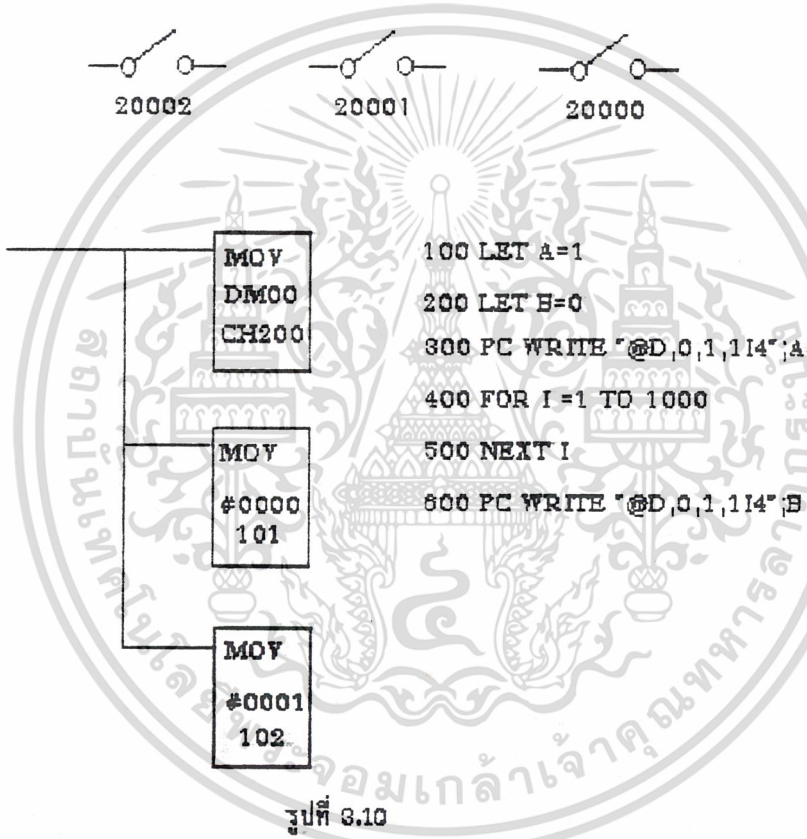
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

FLOW CHART **แสดงการทำงาน ของ**LADDER **และ** BASIC **ของ** ASCII UNIT **ที่สัมพันธ์กัน ของคำสั่ง** PC READ



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

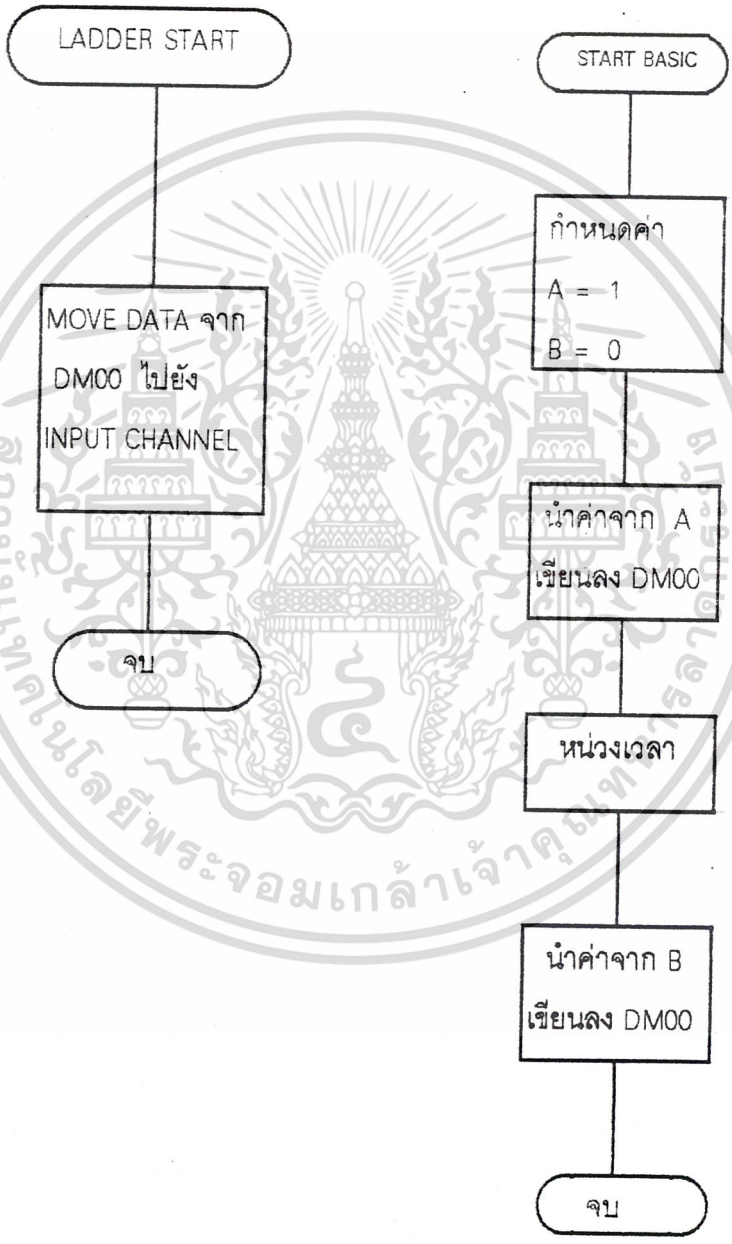
จากรูปที่ 3.9 เป็นโปรแกรมตัวอย่างแสดงความสัมพันธ์กันระหว่าง LADDER PROGRAM และ ASCII PROGRAM ในกรณีที่ทำการอ่านค่าจาก DATAMEMORY โดยที่ input 20000,20001,20002 เป็นเป้าหมายที่ต้องการแสดงผลซึ่งอยู่ใน CH200 เราย้ายค่าจาก CH200 ไปยัง DM00 จากนั้นอ่านค่าจาก DM00 มาเก็บไว้ในตัวแปร X เป็น BCD CODE แล้วเปรียบเทียบเทียบค่ากัน ให้ส่ง ASCII CODE ค่าต่างๆ ออกมายัง พอร์ท 2 ของ ASCII UNIT



รูปที่ 3.10

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

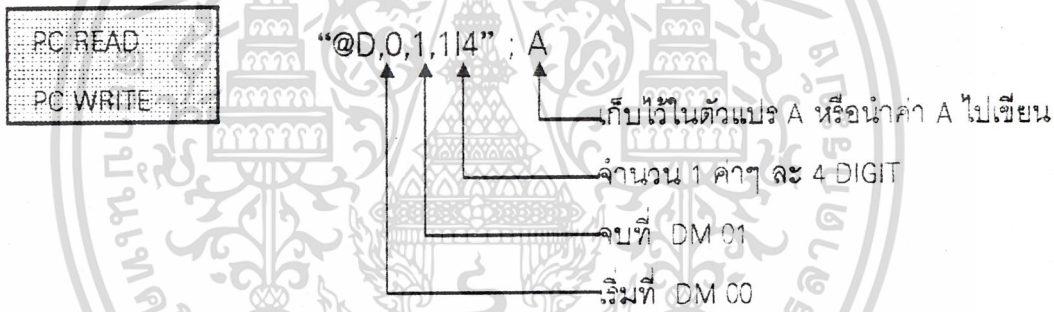
FLOW CHART แสดงการทำงาน ของLADDER และ BASIC ของ ASCII UNIT ที่สัมพันธ์กัน ของคำสั่ง PC WRITE



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่3.10 ในกรณีที่จะควบคุมสั่งงานเราจะใช้คำสั่ง PCWRITE โดยการเขียนค่าตัวเลข BCD ลงใน DATAMEMORY แล้วทำการย้ายค่าจาก DATA MEMORY ลงใน CH200 จากตัวอย่างเราก่อนหนดให้ A = 1 และ B = 0 แล้วเขียนค่า A ลงใน DM00 ขณะเดียวกันก็ย้ายค่าจาก DM00 ไปยัง CH200 ซึ่งในขณะนี้ INPUT 20000 ก็จะ ON เมื่อหน่วงเวลาลงไป แล้วเขียนค่า B ลงใน DM00 อีก INPUT 20000 ก็จะ off ดังนั้นจะเห็นว่าการกระทำเช่นนี้เปรียบเสมือนว่าเราได้กดสวิทช์ 20000 แล้วปล่อยนั่นเอง สำหรับในการติดต่อกับโมเด็ม เราส่งคำสั่ง ATA ให้กับโมเด็ม เมื่อมีการติดต่อจากคอมพิวเตอร์ต้นทาง

เนื่องจากคำสั่งต่างๆ มีค่อนข้างมาก ส่วนใหญ่แล้วจะคล้ายกับคำสั่งใน ภาษาBASIC ทั่วไปที่ทำงานบนPC สำหรับในโครงงานนี้ใช้เพียง PCREAD และ PCWRITE ส่วนในรายละเอียดศึกษาได้จากภาคผนวก



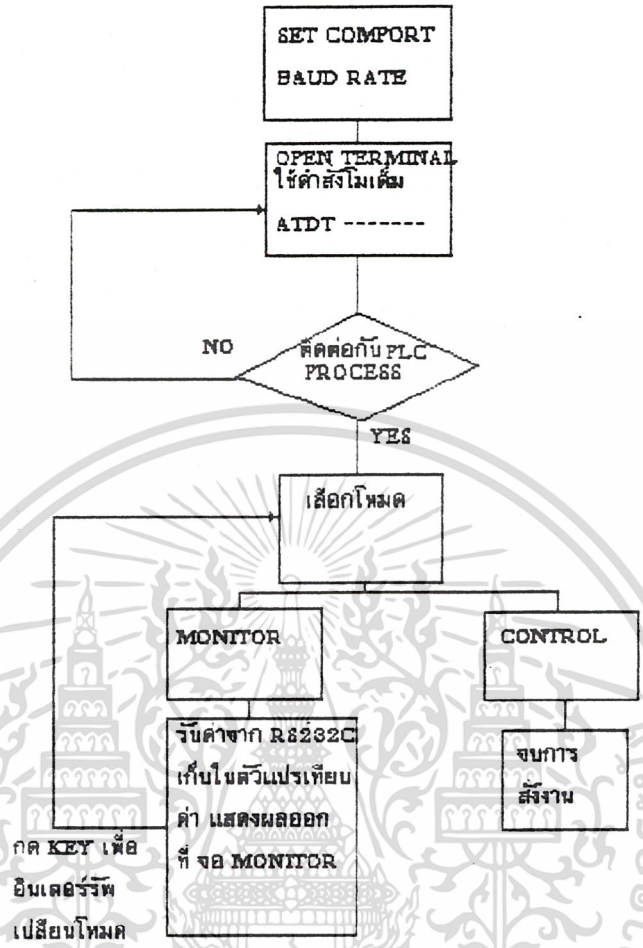
รูปแสดง FORMAT ของคำสั่งPCREAD,PCWRITE

2.3.2 COMMUNICATION PROGRAM และ GRAPHICS DISPLAY

เป็น Software ที่เขียนขึ้นจาก Microsoft Visual Basic for windows version 3.0 ทำหน้าที่ติดต่อกับโมเด็มและรับค่าที่ได้แสดงเป็นภาพออกหน้าจอมอนิเตอร์ ตามการทำงานจริง ๆ ลักษณะของโปรแกรมเป็นลักษณะของเทอมินัล เมื่อติดต่อกับปลายทางได้แล้ว ASCII UNIT จะส่งข้อมูลมายังคอมพิวเตอร์ ปลายทางโดยที่คอมพิวเตอร์ปลายทางจะนำข้อมูลขนาด 1 byte หรือ 1 อักขระ มาเก็บไว้ที่ตัวแปร 'INSTRING' แล้วทำการเปรียบเทียบส่งการแสดงผลออกเป็นรูปภาพที่ต่างกันออกไป

ในส่วนรายละเอียดของ SOFT WARE ได้แสดงไว้ในภาคผนวก แล้ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.11 แสดงการทำงานของโปรแกรมการติดต่อสื่อสาร

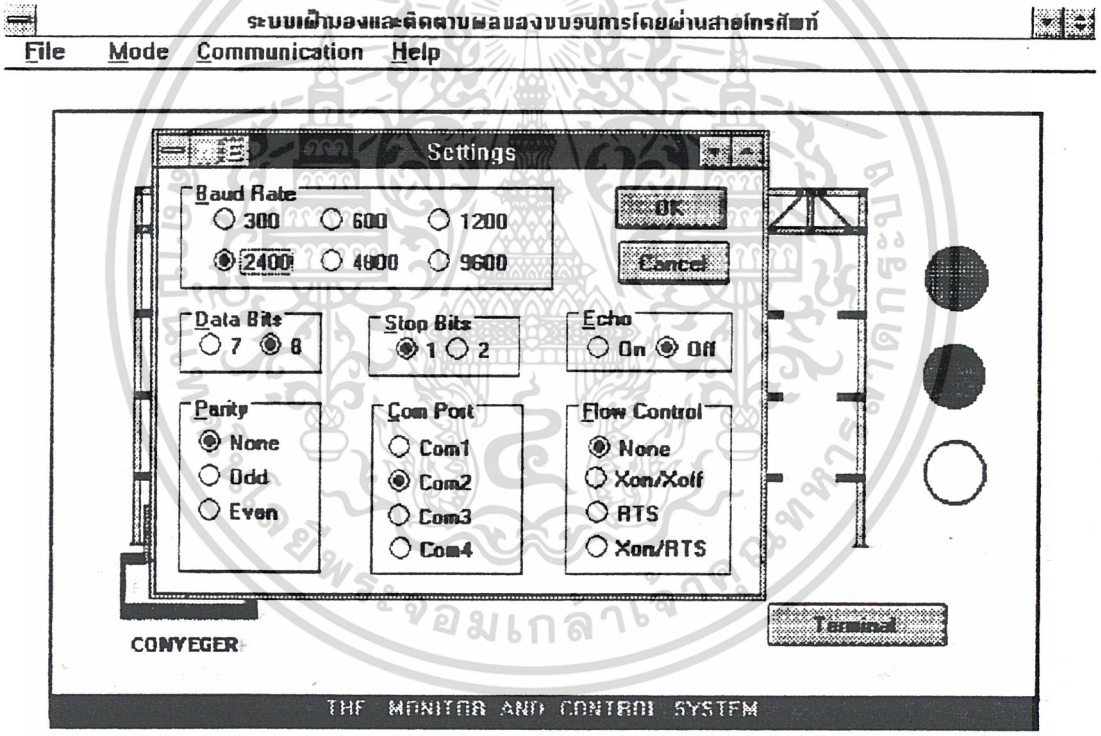
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

การทดลองโครงงาน

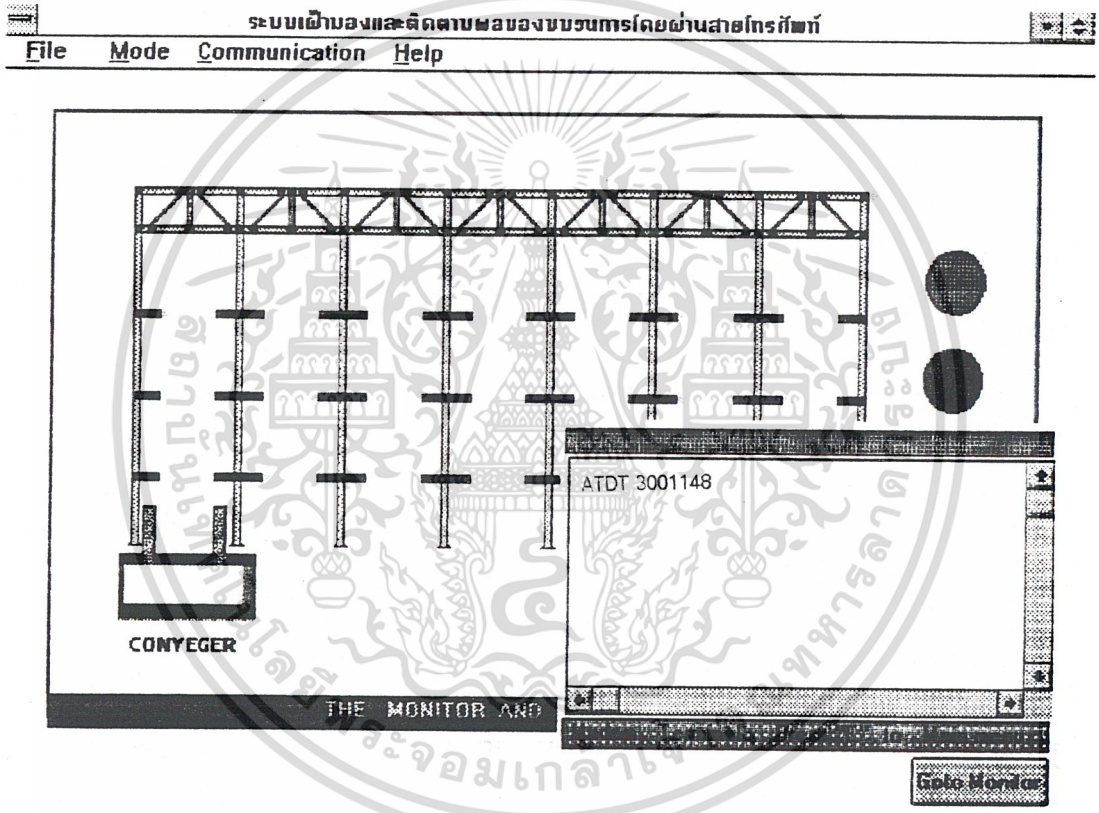
ในการทดลองระบบเฝ้ามองและติดตามผลของขบวนการโดยผ่านสายโทรศัพท์เป็นที่น่าพอใจ โดยสรุปเป็นขั้นการทำงานดังนี้

4.1 SET ค่าพื้นฐานต่าง ๆ จาก SOFTWARE COMMUNICATION



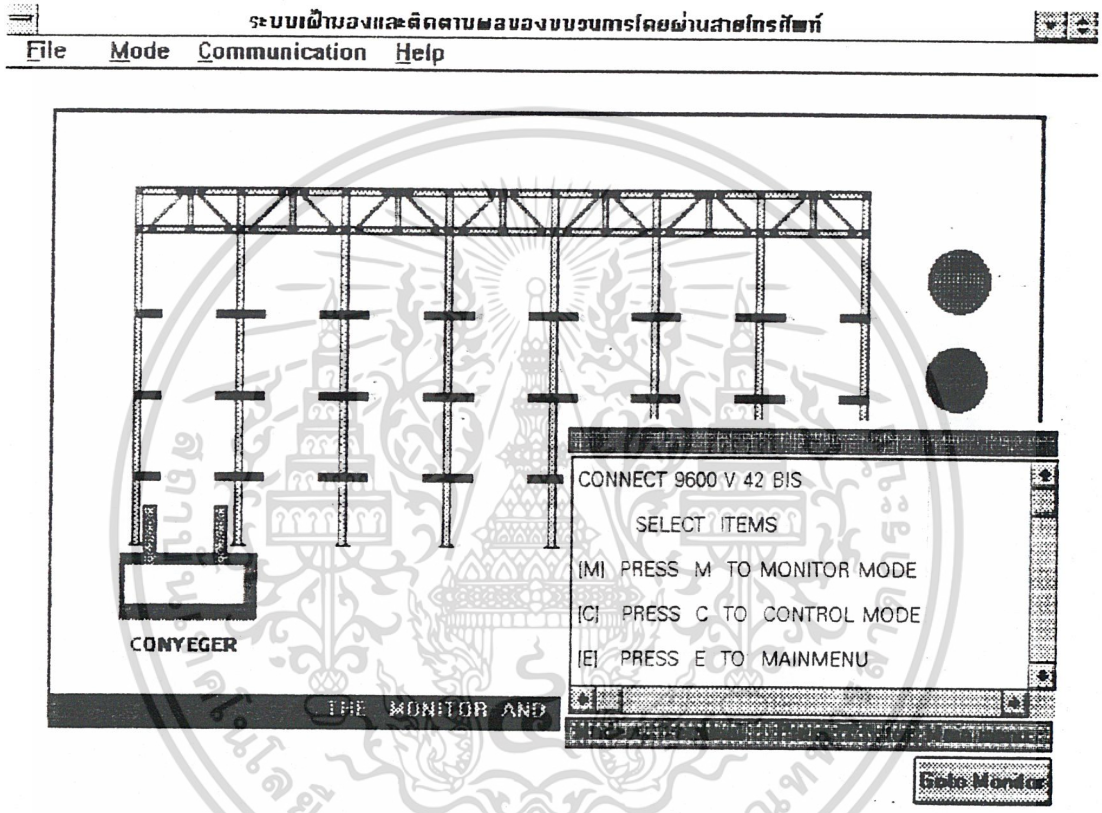
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 4.2 เปิดเทอมินัลของโปรแกรม เพื่อใช้คำสั่ง หมุนหมายเลขของโมเด็มโดยใช้คำสั่ง ATDT ตามด้วยเบอร์โทรศัพท์



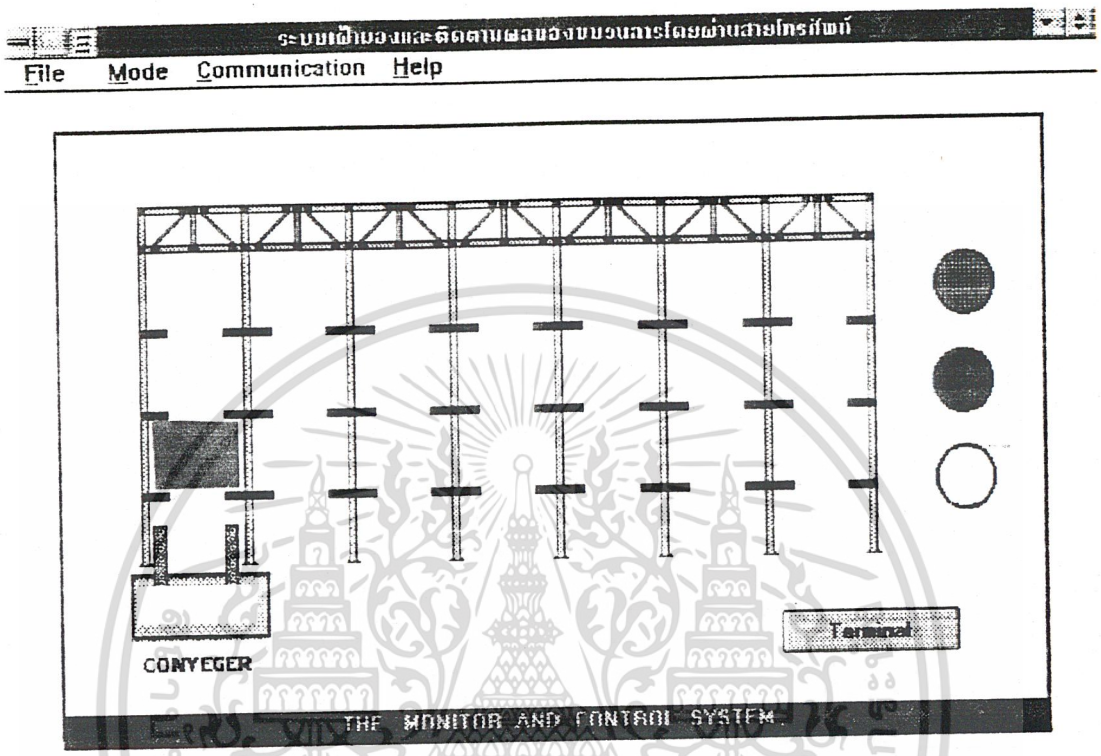
หรืออาจป้อน เบอร์โทรศัทพ์ เข้าที่เมนูก็ได้ จากนั้นรอการติดต่อจากโมเด็มปลายทาง และป้อนข้อมูล ตามข่าวสารที่ส่งมา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



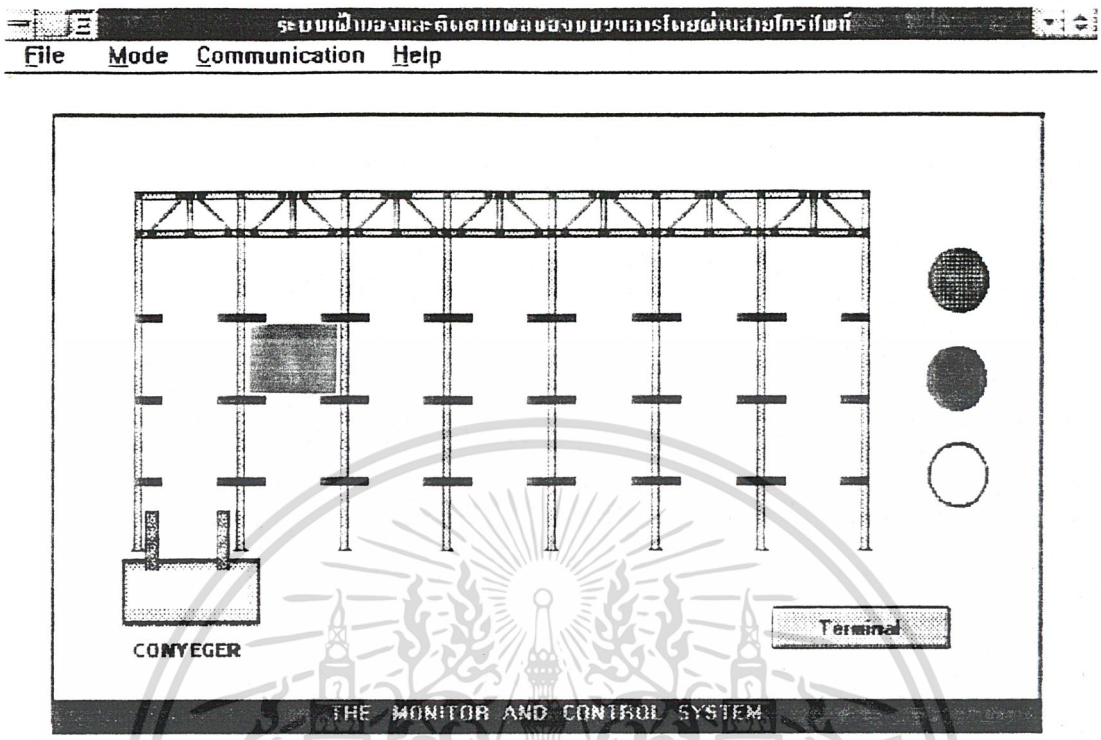
รูป4.3 เมื่อติดต่อได้แล้วให้ทำการเลือกโหมดการทำงาน ให้เป็นมอนิเตอร์ แล้วคลิกปุ่ม GOTO MONITOR เพื่อดูสภาวะการทำงานซึ่งในขณะนี้ไม่ว่าตัว WARE HOUSE จะอยู่ในสภาวะใดภาพที่ปรากฏก็จะตรงกัน ไม่ว่าอยู่กับที่หรือเคลื่อนไหว ในขณะนี้ที่ตัว PLC จะมีคนใช้งาน WARE HOUSE อยู่ด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



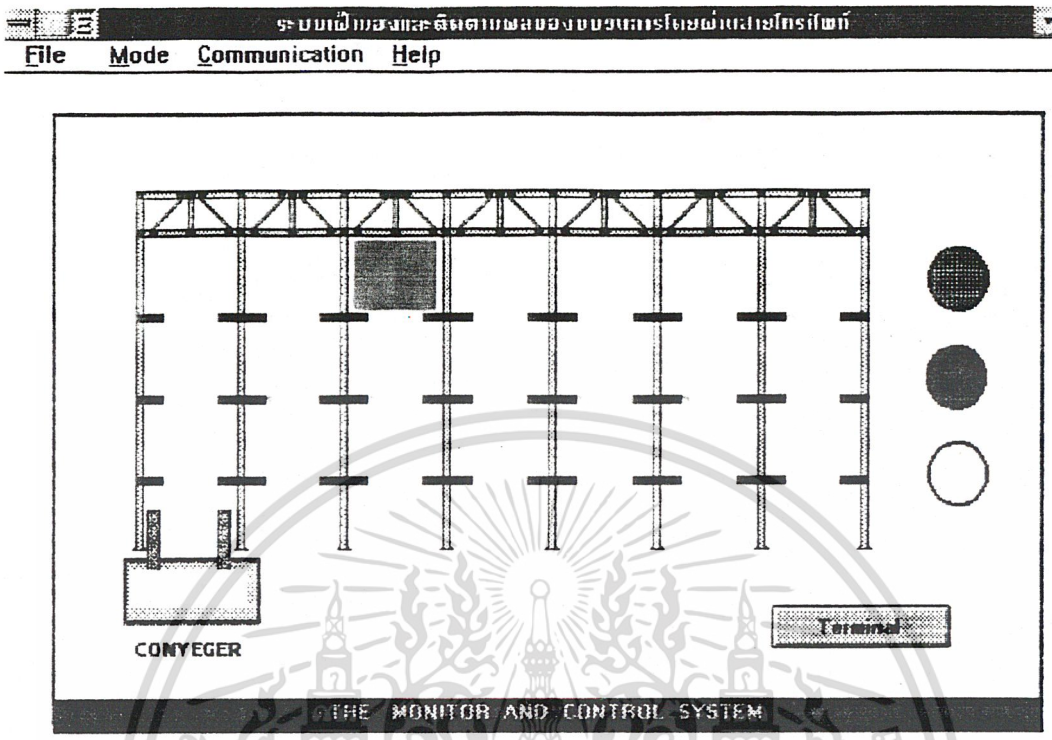
ชั้นจะเริ่มเคลื่อนที่ ยกตัวอย่าง จาก ตำแหน่ง 11 ไป 33 (11 หมายถึง ชั้นที่ 1 หลักที่ 1 และ 33 หมายถึง ชั้นที่ 3 หลักที่ 3)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



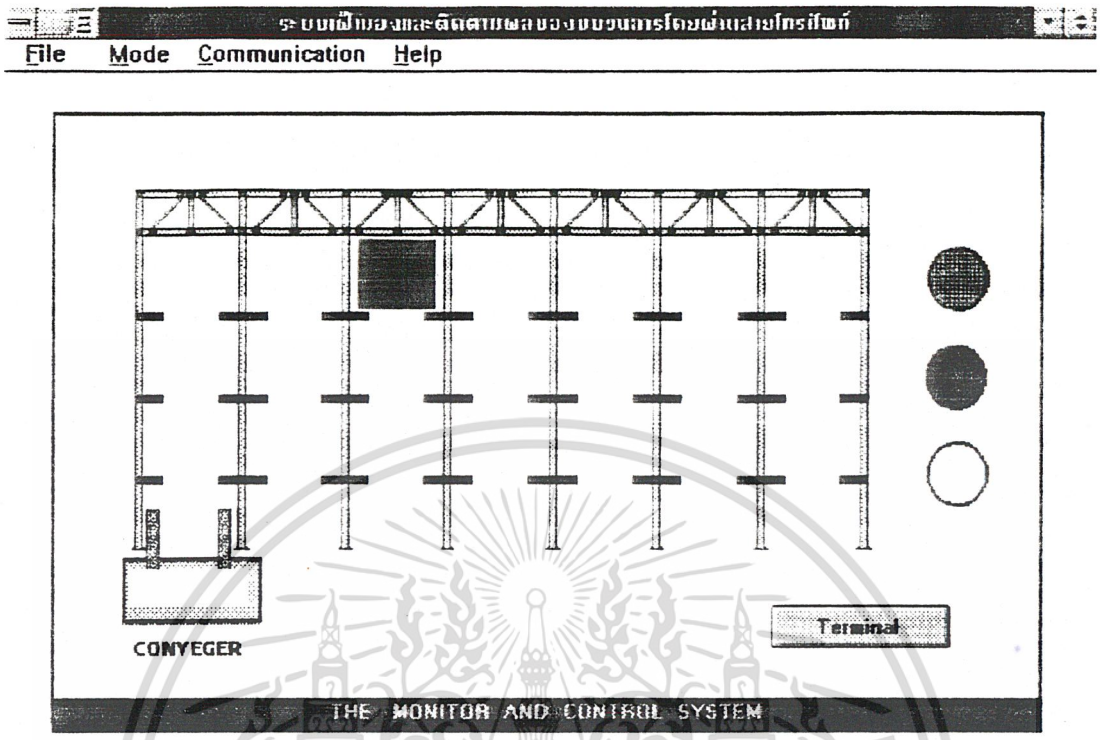
การเคลื่อนที่จะอิสระ ต่อกัน โดย แกนนอน และแกน ตั้งจะไปพร้อมๆ กัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



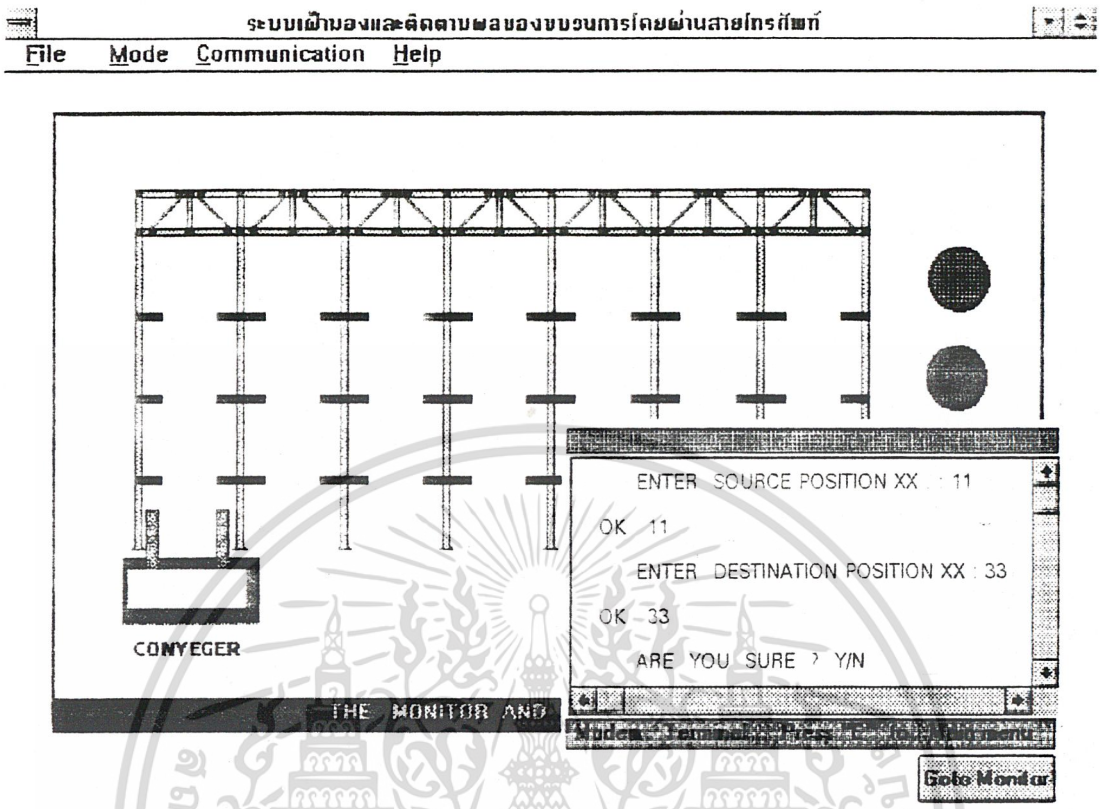
เมื่อของเคลื่อนที่มาถึงตำแหน่งที่กำหนด ก็จะทำการหยุดส่งของ ที่ ตำแหน่ง 33

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ในขณะที่ถาดใส่ของเคลื่อนที่เข้าข้างใน จะแสดงให้เห็นโดยการเปลี่ยนสีจากเขียวเป็นแดง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เมื่อจะทำการสั่งงานจากตัวคอมพิวเตอร์ ให้คลิกที่ Terminal ก่อนจากนั้น Key 'E' เพื่อทำการอินเตอร์รัพให้กลับสู่ Setlct items แล้วเลือกไปที่ Control system จะปรากฏเมนูขึ้นอีกให้เลือกโหมดการควบคุม ซึ่งมีให้เลือก 3 โหมด คือ Move, Send, Auto ในกรณีที่เลือกโหมดย้าย(Move) ต้องใส่ตำแหน่งซึ่งเป็นตัวเลข 2 หลัก ตัวแรกหมายถึงแถวในแนวแกน Y และหลักในแนวแกน X (Y มีตั้งแต่ 1 - 3) , X มีตั้งแต่ 1 - 7) การย้ายจะย้ายจากจุดหนึ่งไปยังอีกจุดหนึ่ง เมื่อป้อนค่าถูกต้องแล้วให้ตอบ YES ไปจากนั้นตัว WARE HOUSE ก็จะทำงานตามที่เราป้อนค่าลงไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

สรุปผลการทดลอง และแนวทางการพัฒนา

จากผลการทดลองสรุปว่า เป็นไปตามเป้าหมายที่ตั้งไว้ แต่โครงงานนี้ยังสามารถที่จะพัฒนาต่อไปได้อีกหลายด้าน และประยุกต์ใช้กับกระบวนการอื่น ๆ นอกจาก WARE HOUSE MODEL อีกมากมาย ในส่วนของโครงงานที่ใช้ WARE HOUSE MODEL นี้ยังมีฟังก์ชันต่าง ๆ ที่จะเพิ่มเติมได้อีกเช่น การตั้งโปรแกรม เป็น EDITTER ที่สามารถปรับเปลี่ยนล่วงหน้าได้ อาจจะเป็นเดือน ปี ตั้งเวลา หรือการเพิ่มการตรวจสอบของที่ใส่ในชั้นวาง

จากการทำโครงงานนี้ เป็นตัวอย่างหนึ่งที่เป็นระบบเฝ้ามองติดตามผล ในระยะทางไกล ๆ ในที่ที่ การสื่อสารและโทรคมนาคมเข้าถึง ไม่ใช่เพียงแต่กระบวนการใดกระบวนการหนึ่งเท่านั้น แต่ทุกกระบวนการสามารถที่จะนำมาประยุกต์เข้ากับระบบนี้ ได้อย่างไม่ยากนัก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก

A. LADDER PROGRAM

B. ASCII UNIT BASIC PROGRAM

C. COMMUNICATION PROGRAM

D. ASCII BASIC COMMAND

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก A

LADDER PROGRAM

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การกำหนดอินพุทและเอาท์พุท

Input CH 3,4

limit แถว 1 ch 3 Bit 04

limit แถว 2 ch 3 Bit 05

limit แถว 3 ch 3 Bit 06

limit แถว 4 ch 3 Bit 07

limit แถว 5 ch 3 Bit 08

limit แถว 6 ch 3 Bit 09

limit แถว 7 ch 3 Bit 10

limit Down ชั้น 3 ch 3 Bit 03

limit Down ชั้น 2 ch 3 Bit 02

limit Down ชั้น 1 ch 3 Bit 01

limit Up ชั้น 3 ch 3 Bit 02

limit Down ชั้น 2 ch 3 Bit 01

limit Down ชั้น 1 ch 3 Bit 00

limit ลิฟท์รับของ in ch 4 Bit 03

limit ลิฟท์รับของ out ch 4 Bit 05

limit ลิฟท์รับของ midium ch 4 Bit 06

Conveyer เข้า ch 4 Bit 06

Conveyer ออก ch 4 Bit 07

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Mode	ส่ง	CA 3	Bit 11
Mode	รับ	CA 3	Bit 12
Mode	ย้ายส่ง	CA 3	Bit 13
Mode	AUTO	CA 3	Bit 14

<u>OUTUPT CH 1, 2</u>

motor ขับ Conveyer out put ch 1 Bit 07

motor ขับ ลิฟท์ ทางด้านแกน X

X เลื่อนมาทางซ้ายมือ out put ch 2 Bit 08

X เลื่อนมาทางขวามือ out put ch 1 Bit 09

motor ขับ ลิฟท์ ทางด้านแกน Y

Y เลื่อน ลง out put ch 2 Bit 09

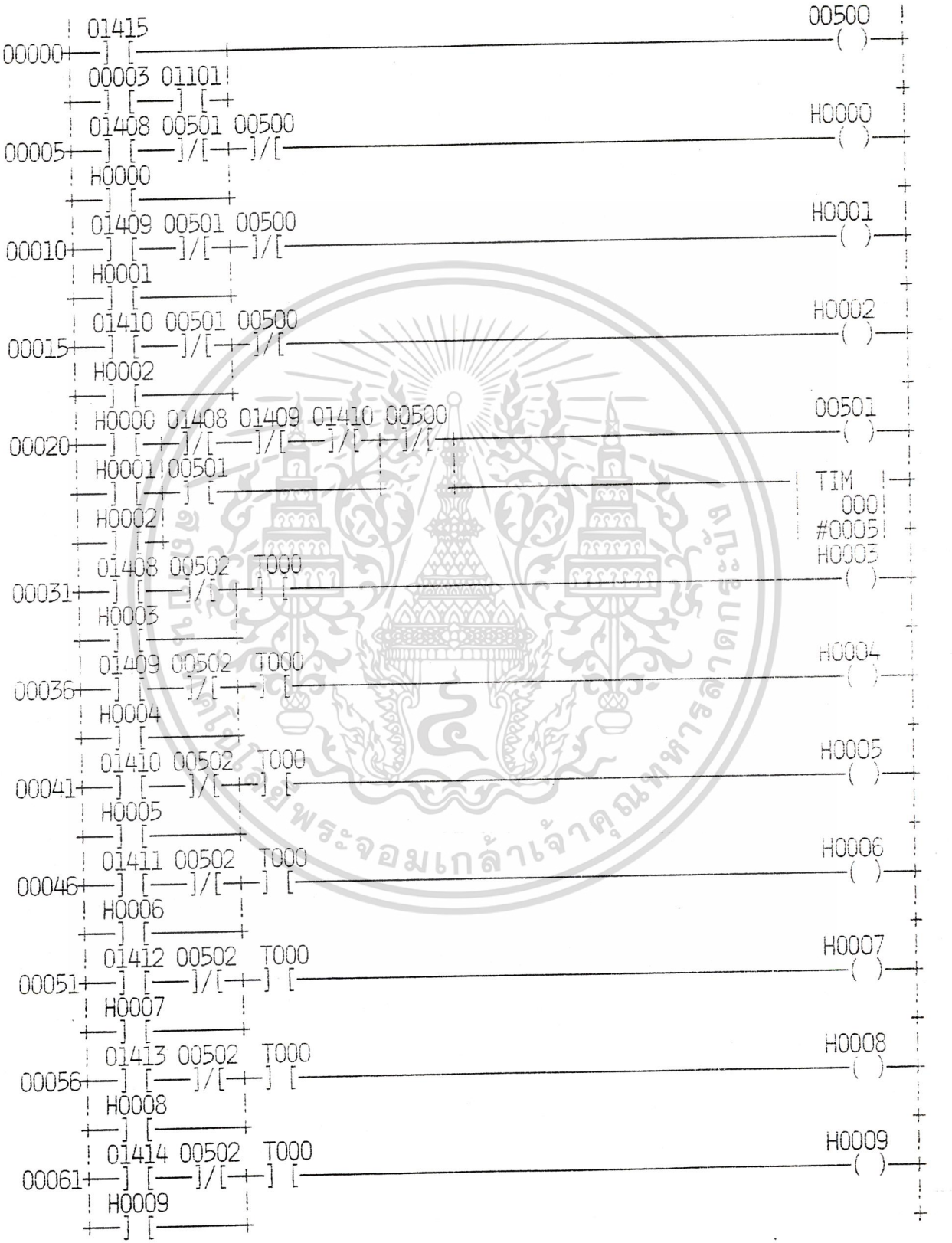
Y เลื่อน ขึ้น out put ch 1 Bit 05

motor ขับแกน Z

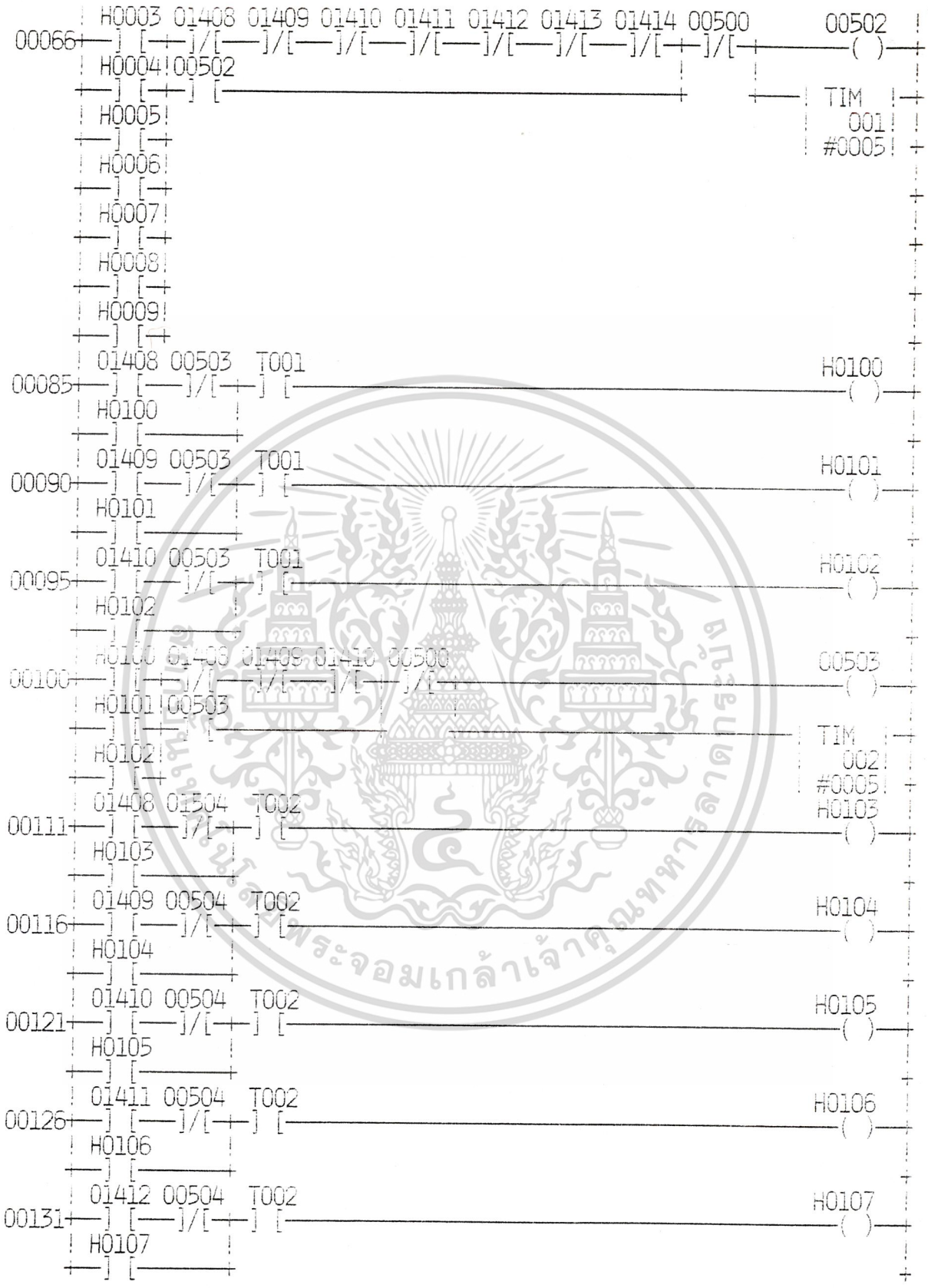
Z เลื่อน เข้า out put ch 1 Bit 06

Z เลื่อน ออก out put ch 2 Bit 10

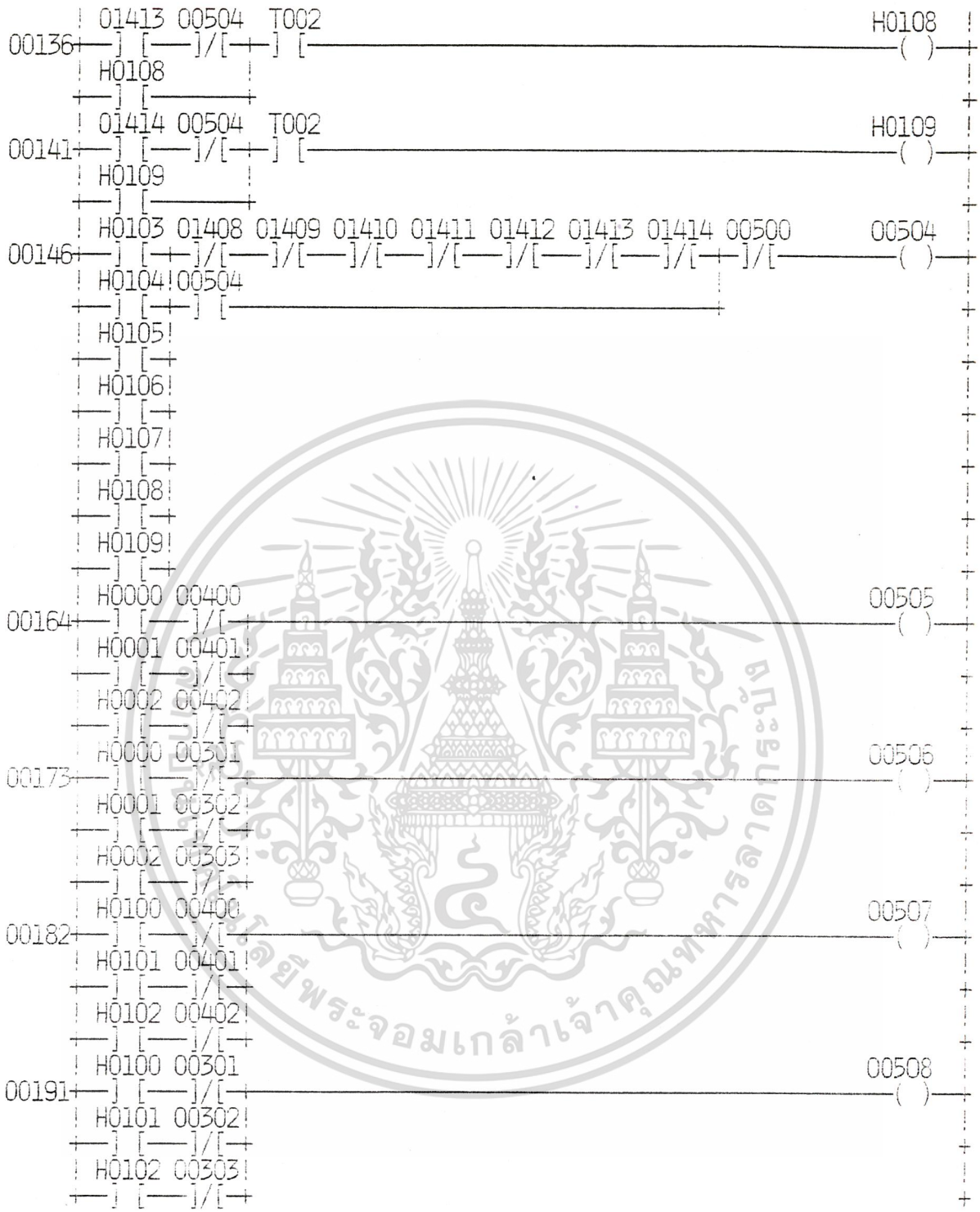
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



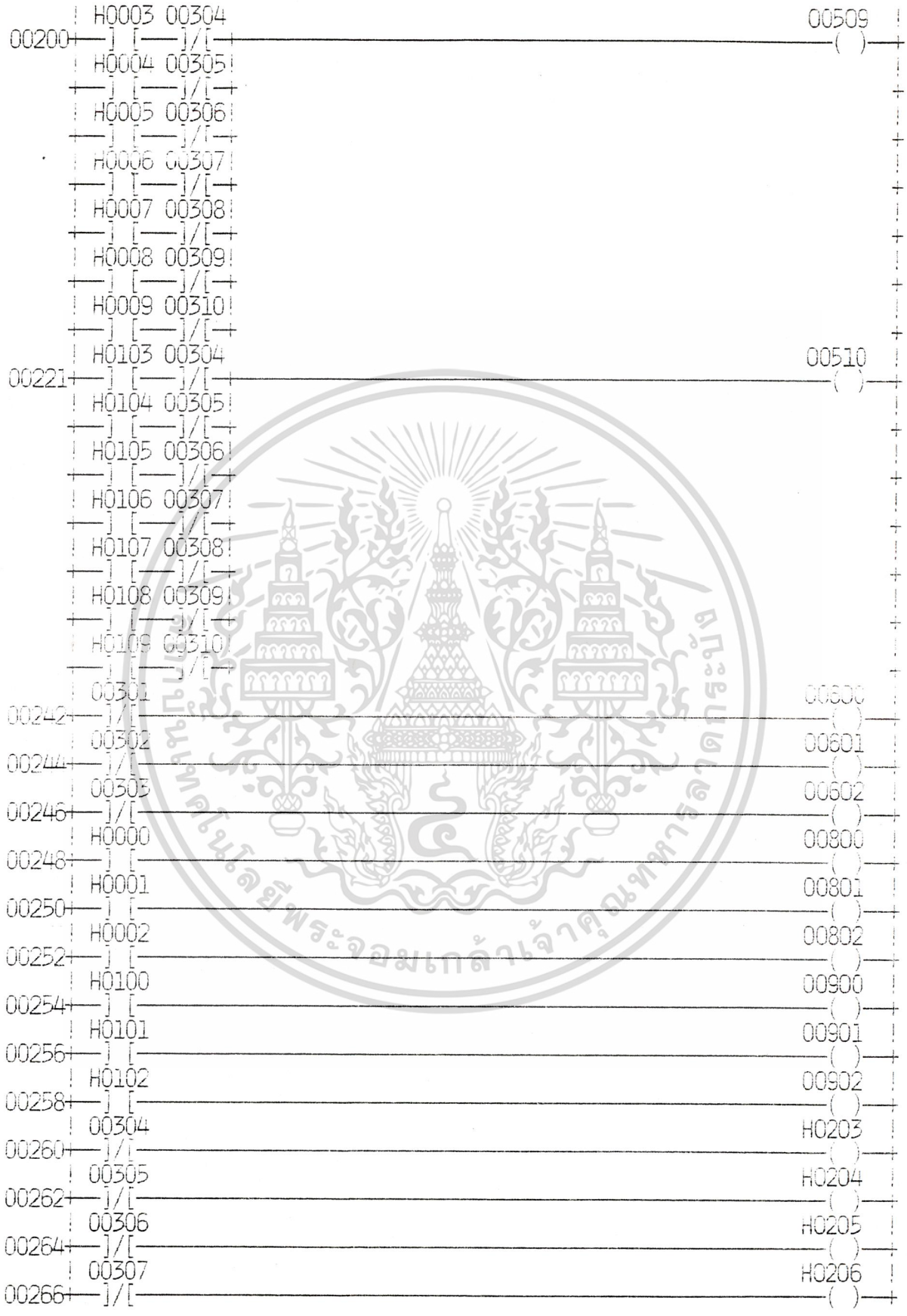
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



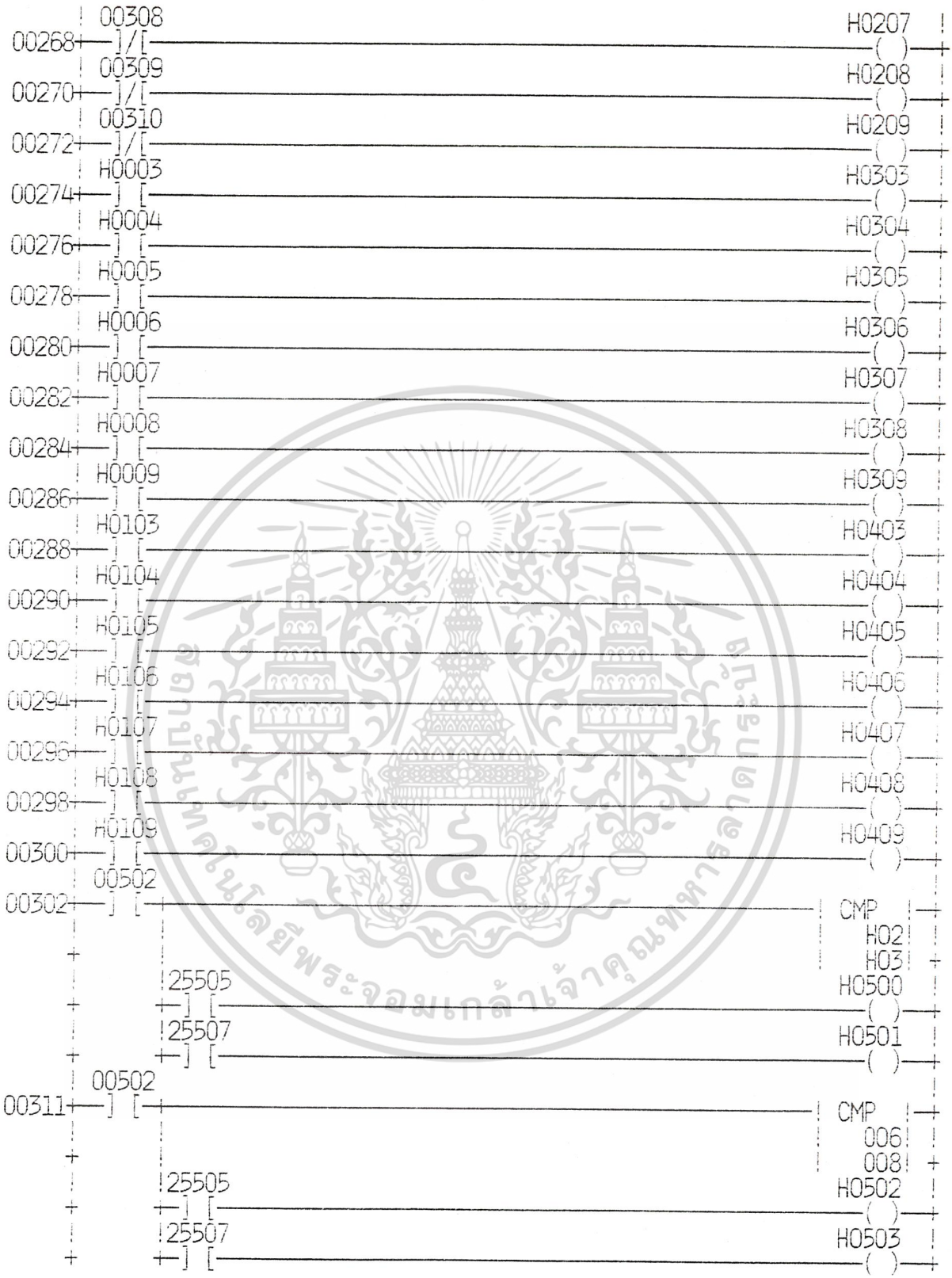
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



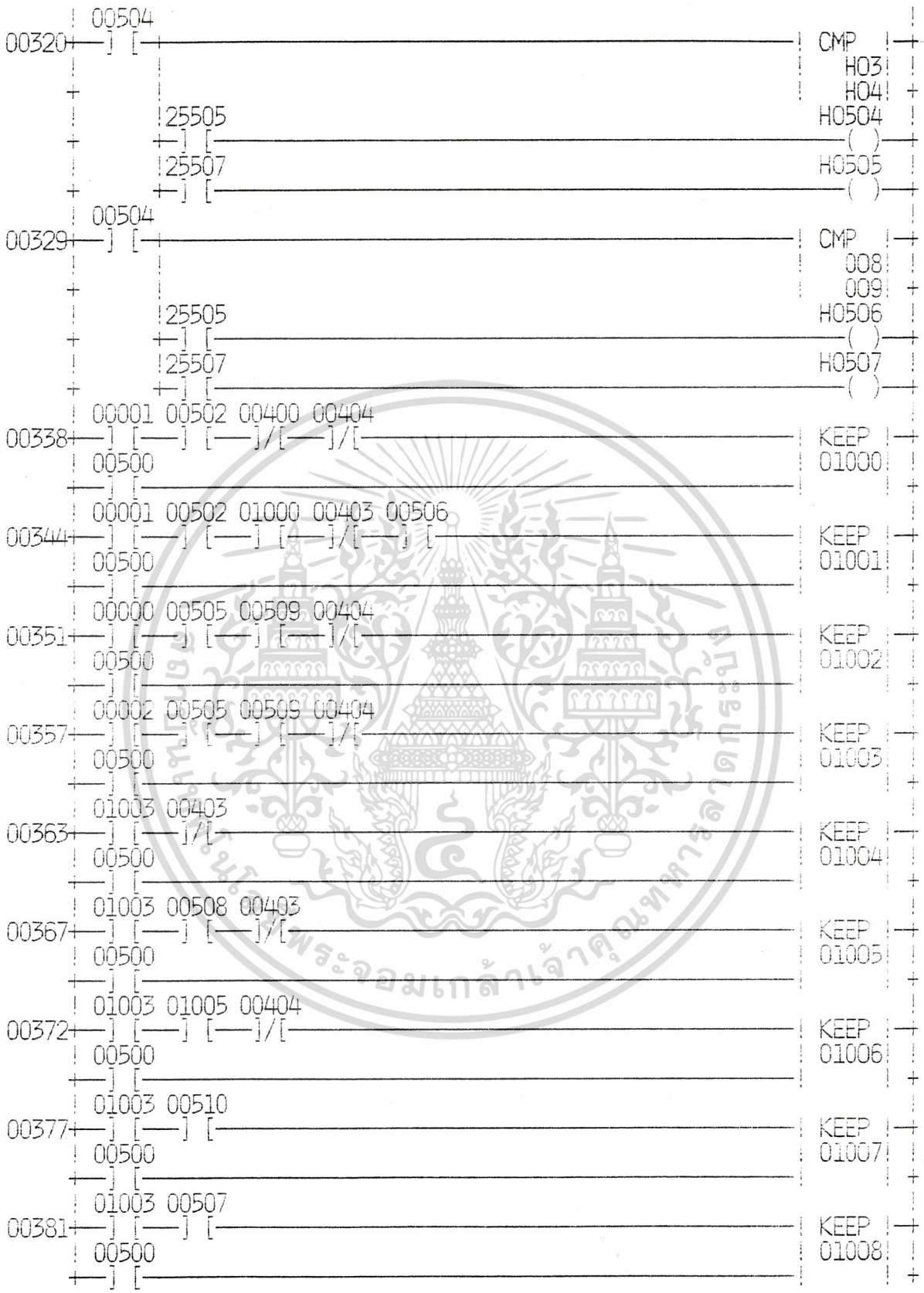
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



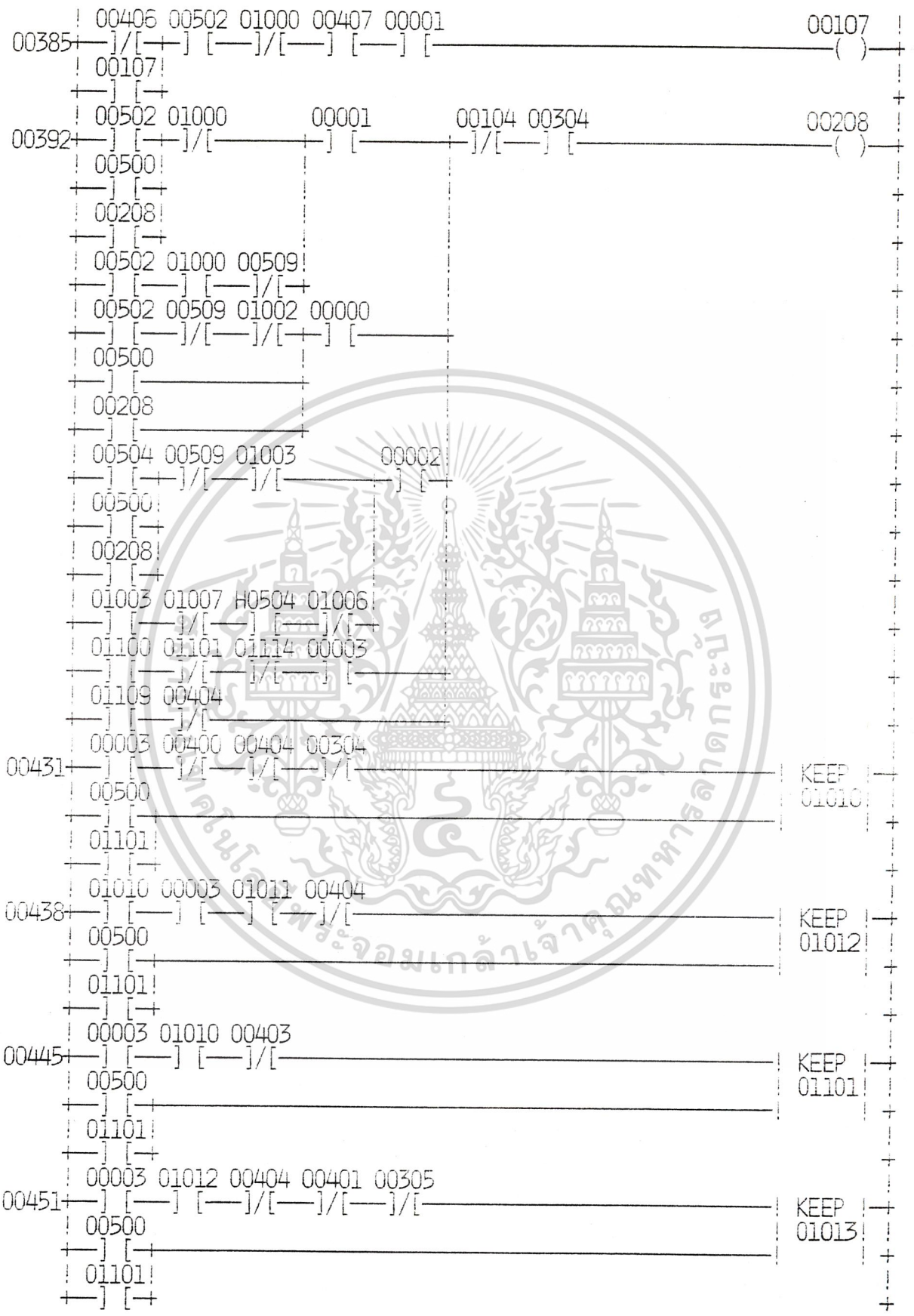
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



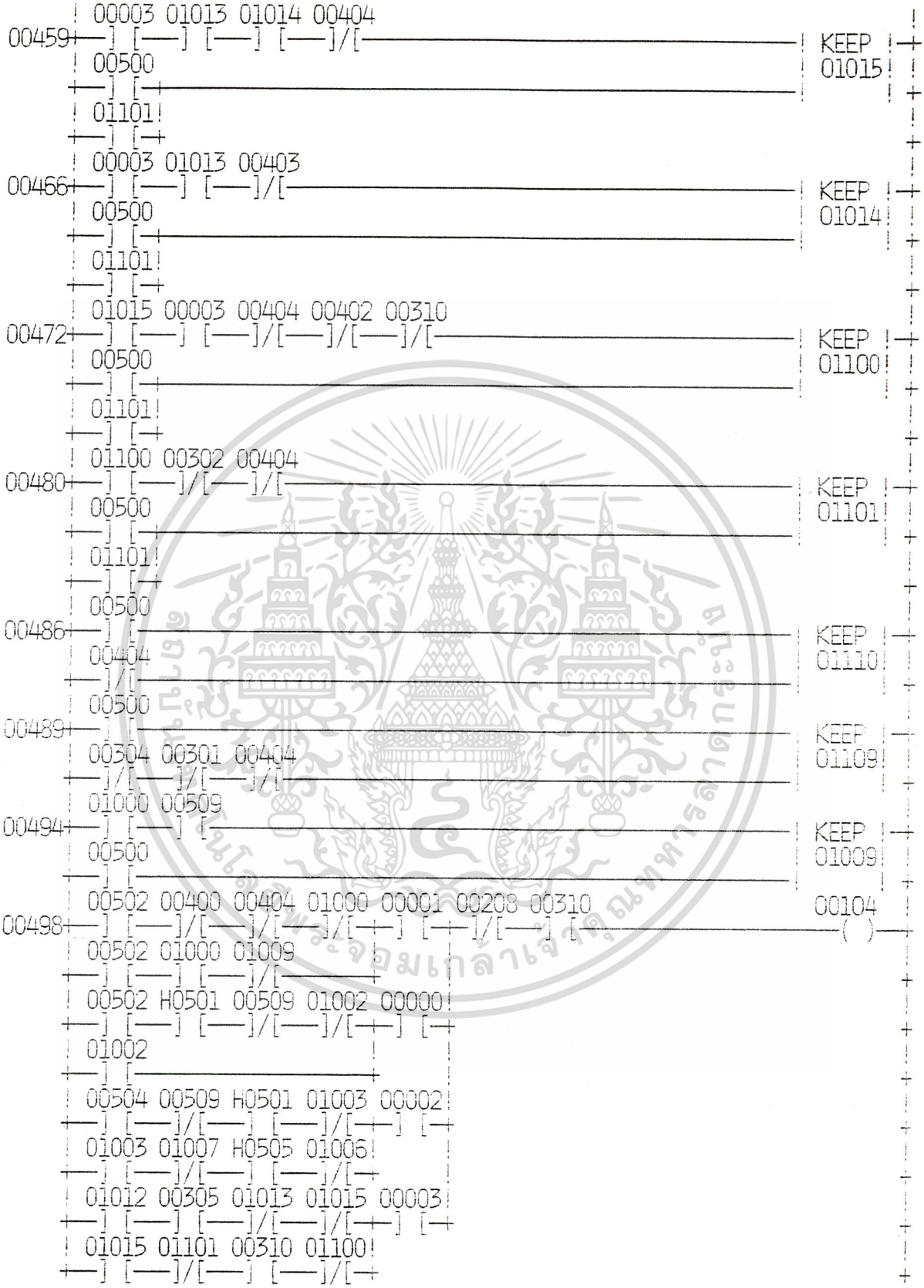
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



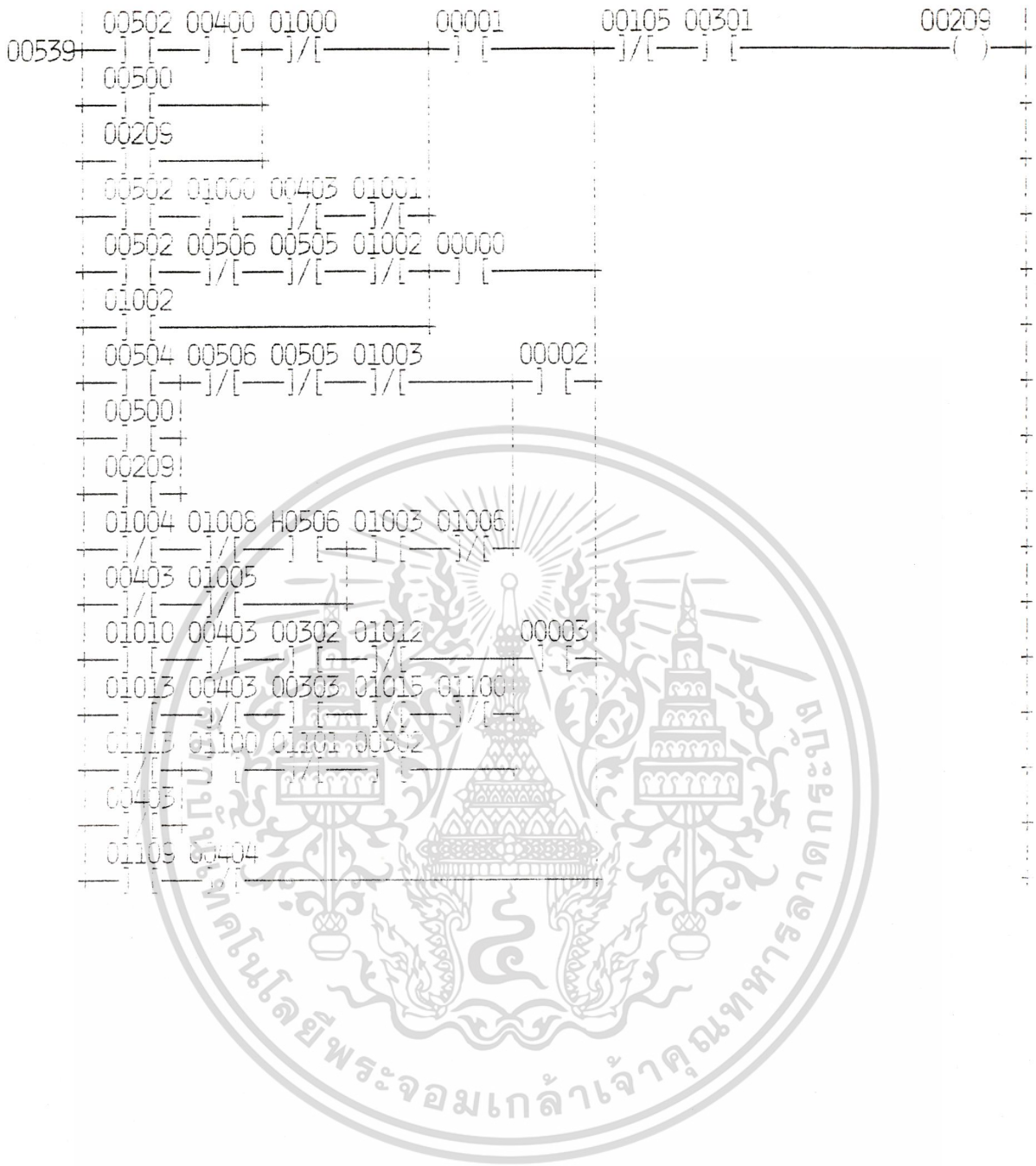
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



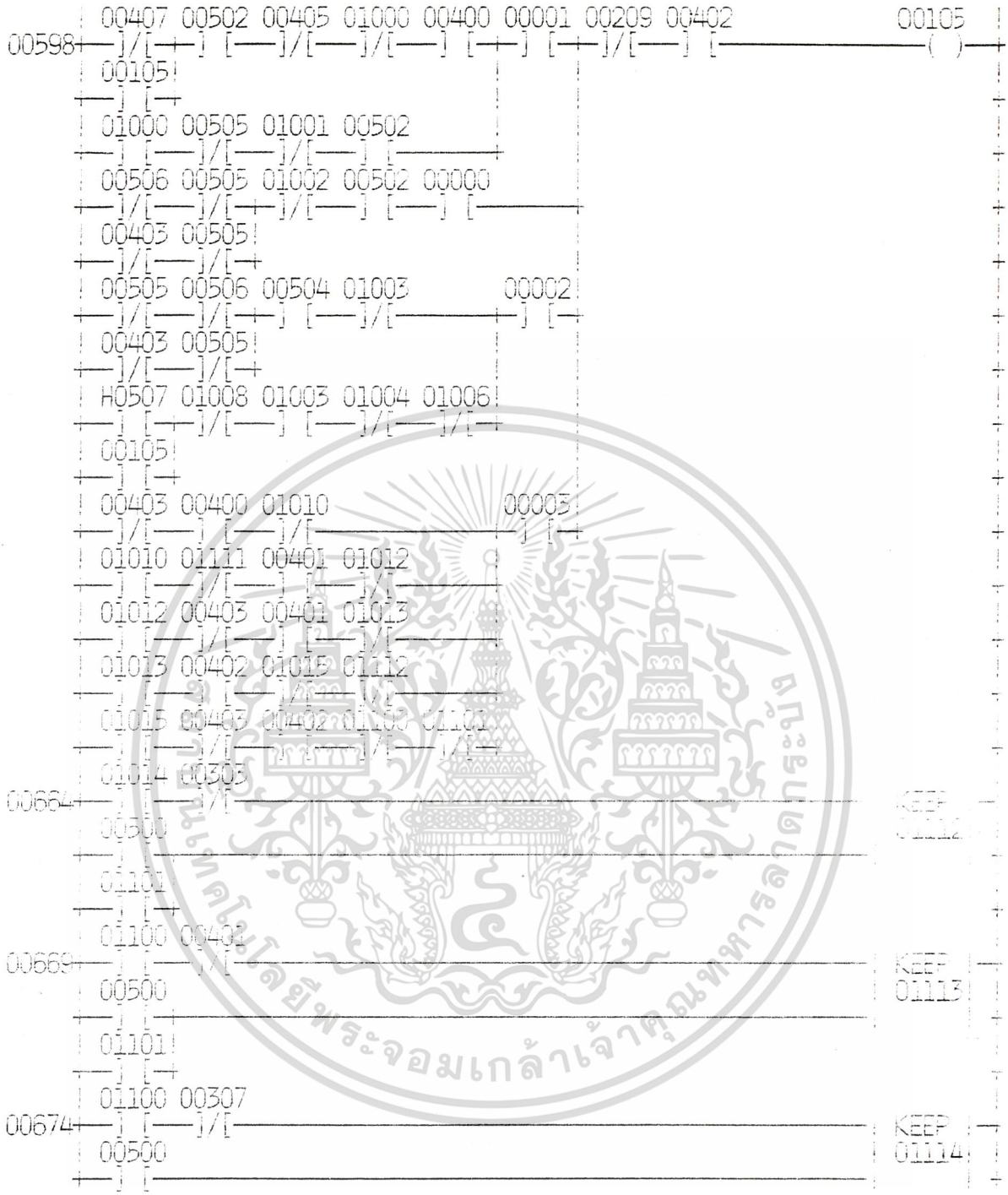
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



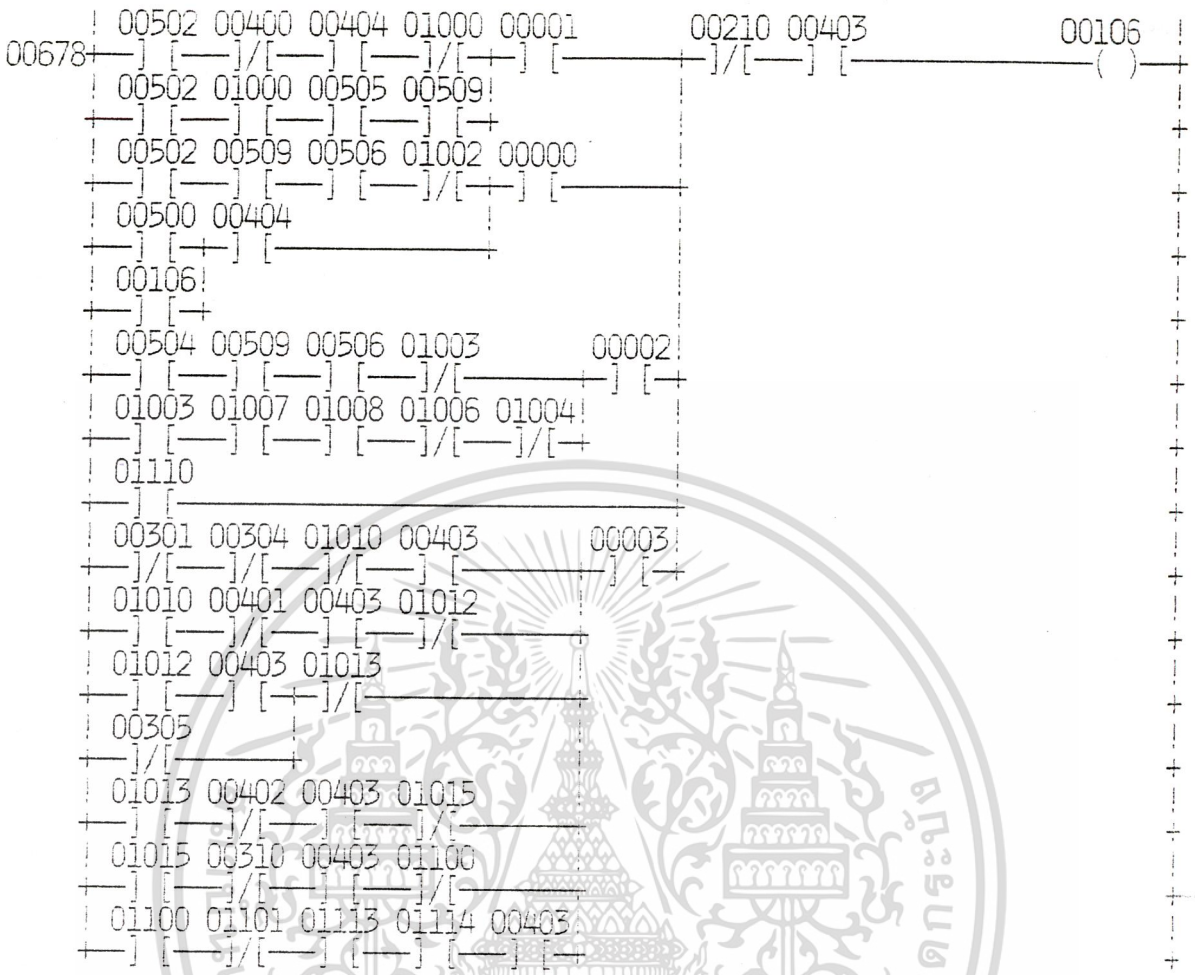
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



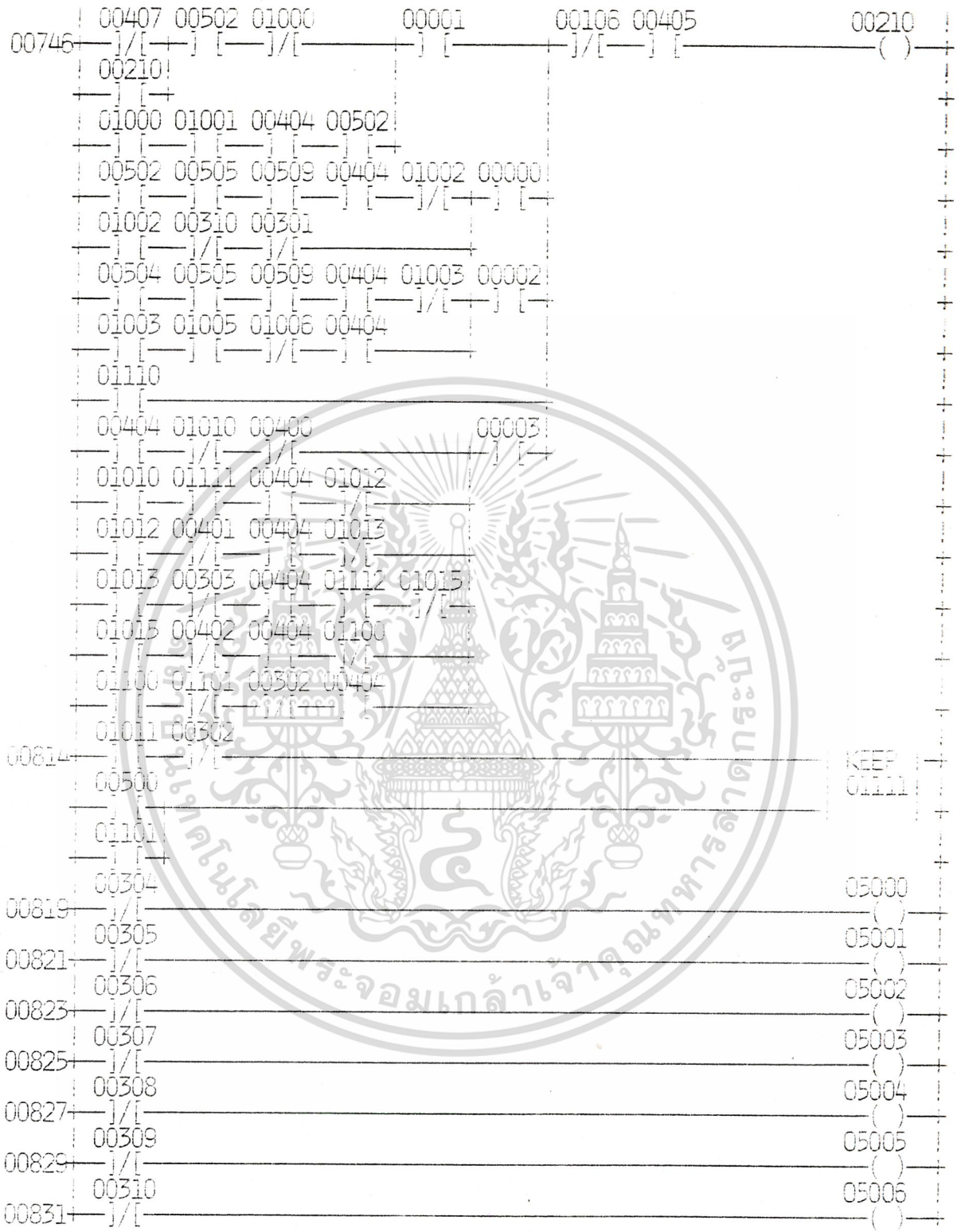
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



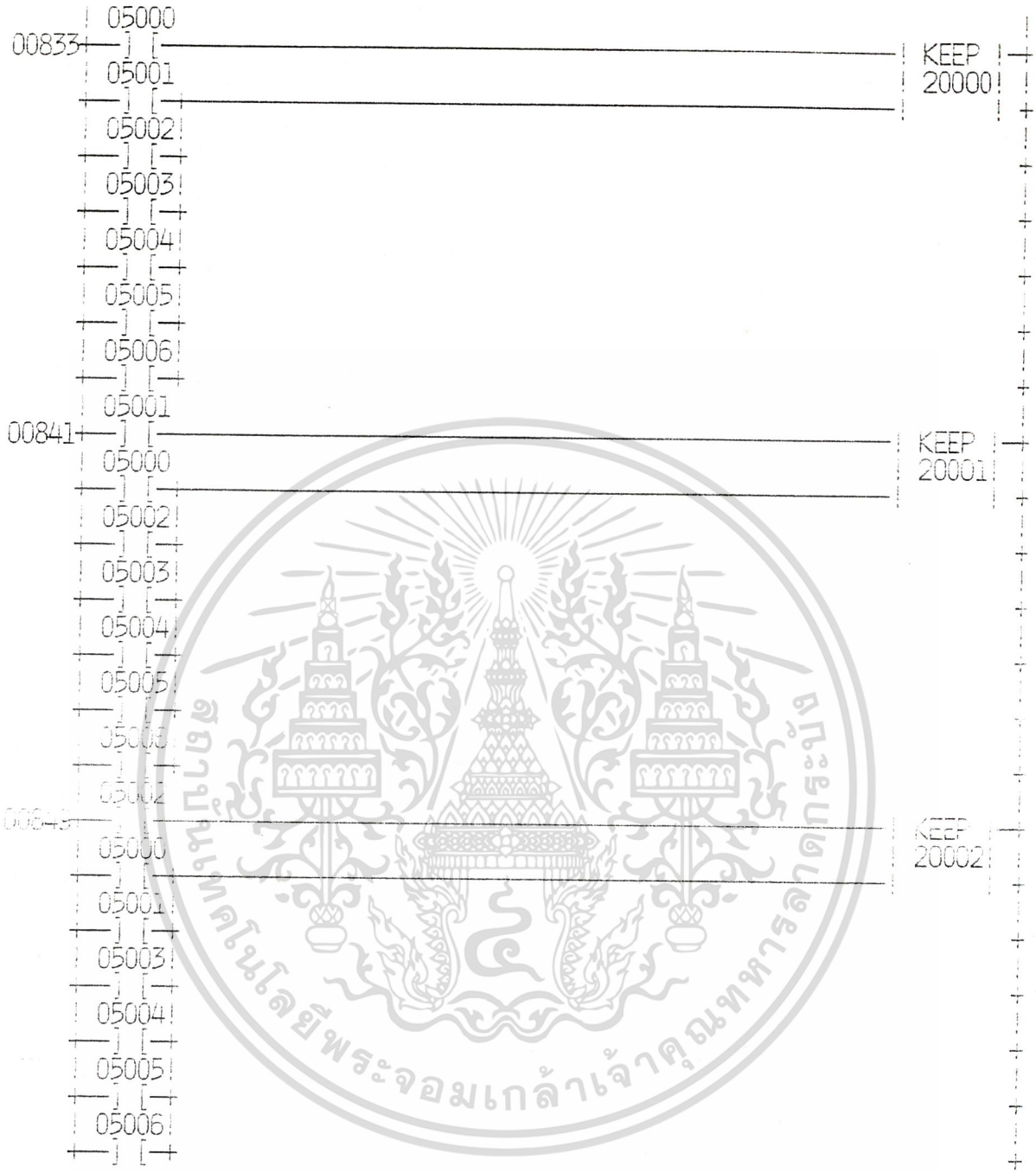
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



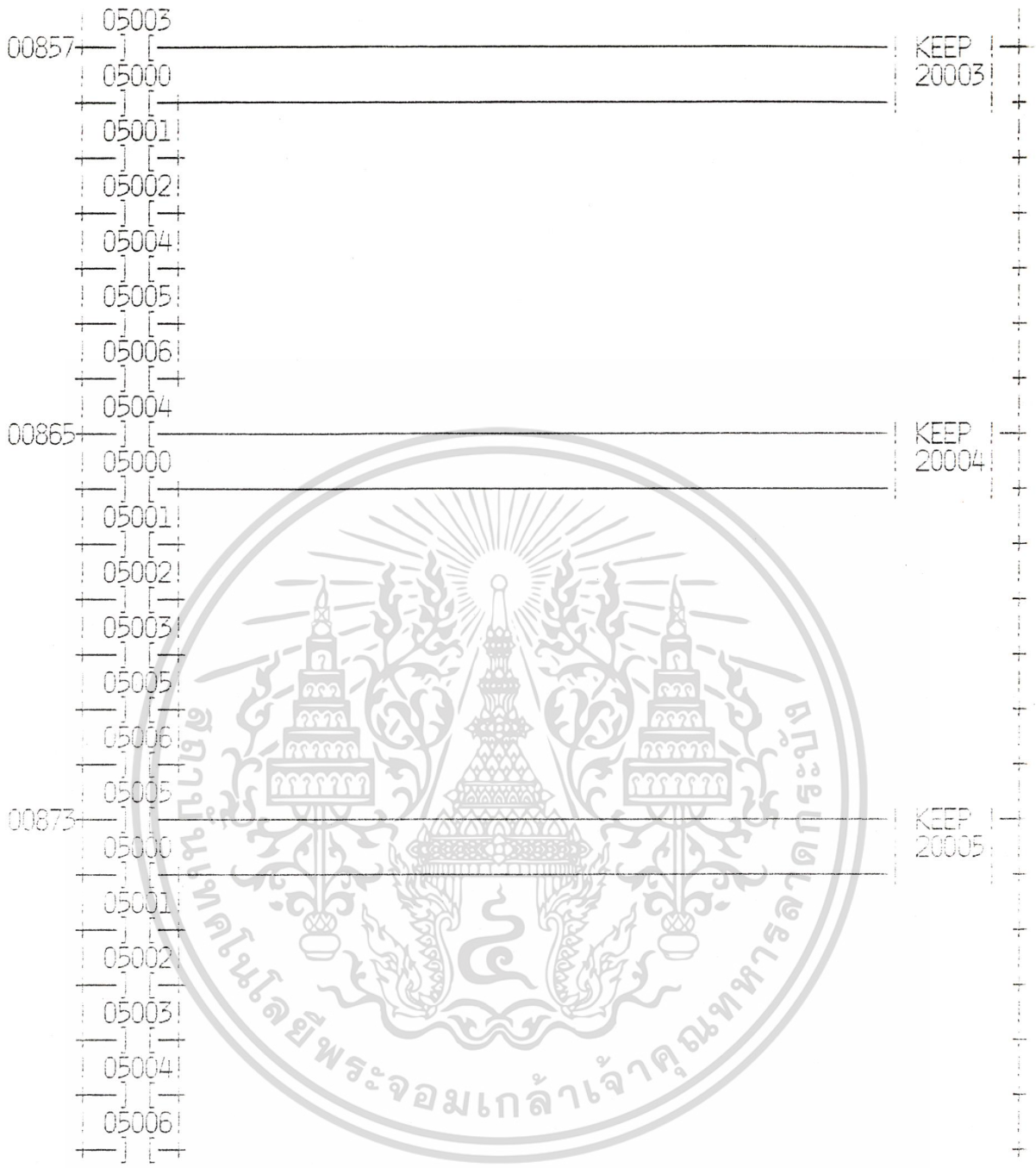
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



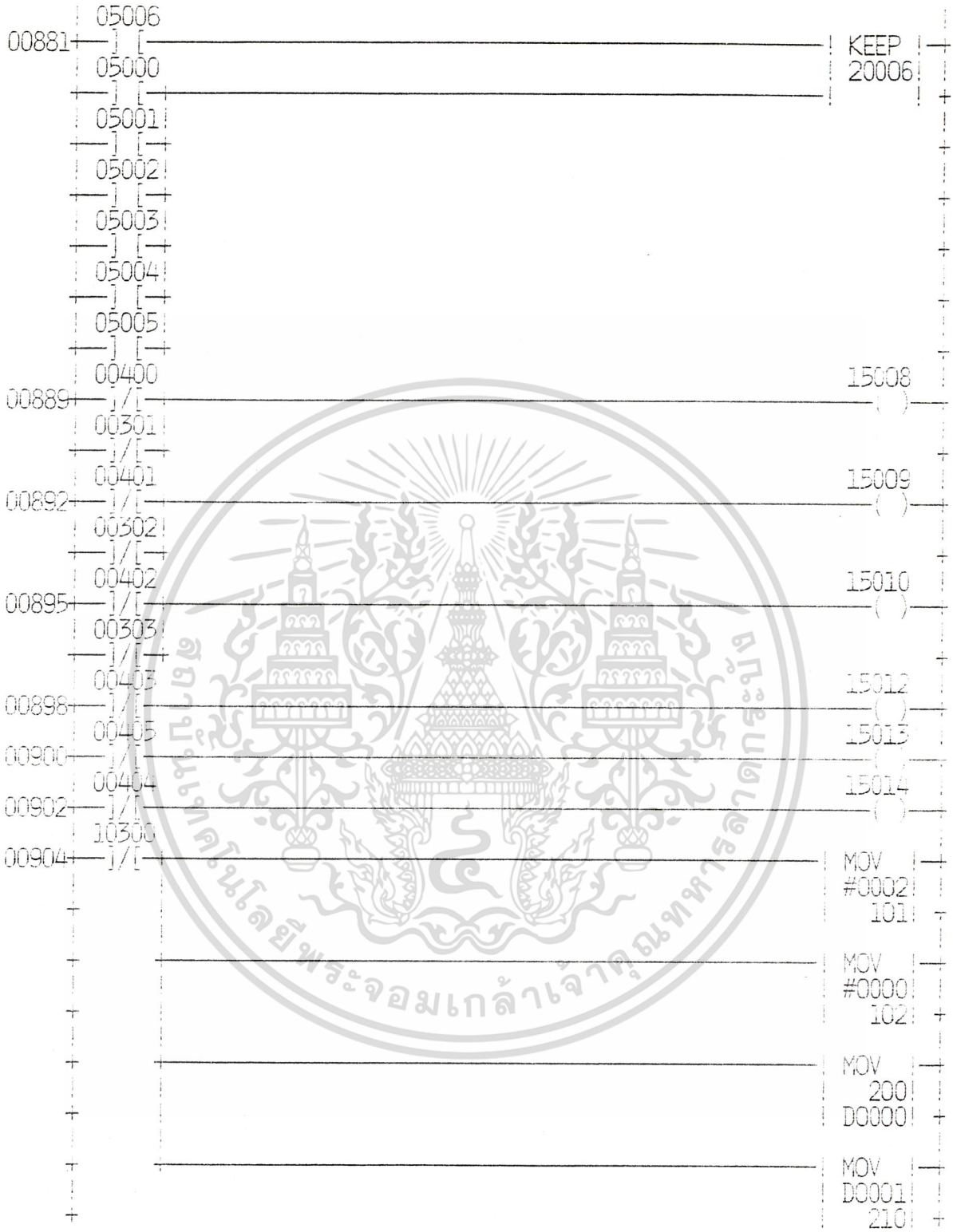
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



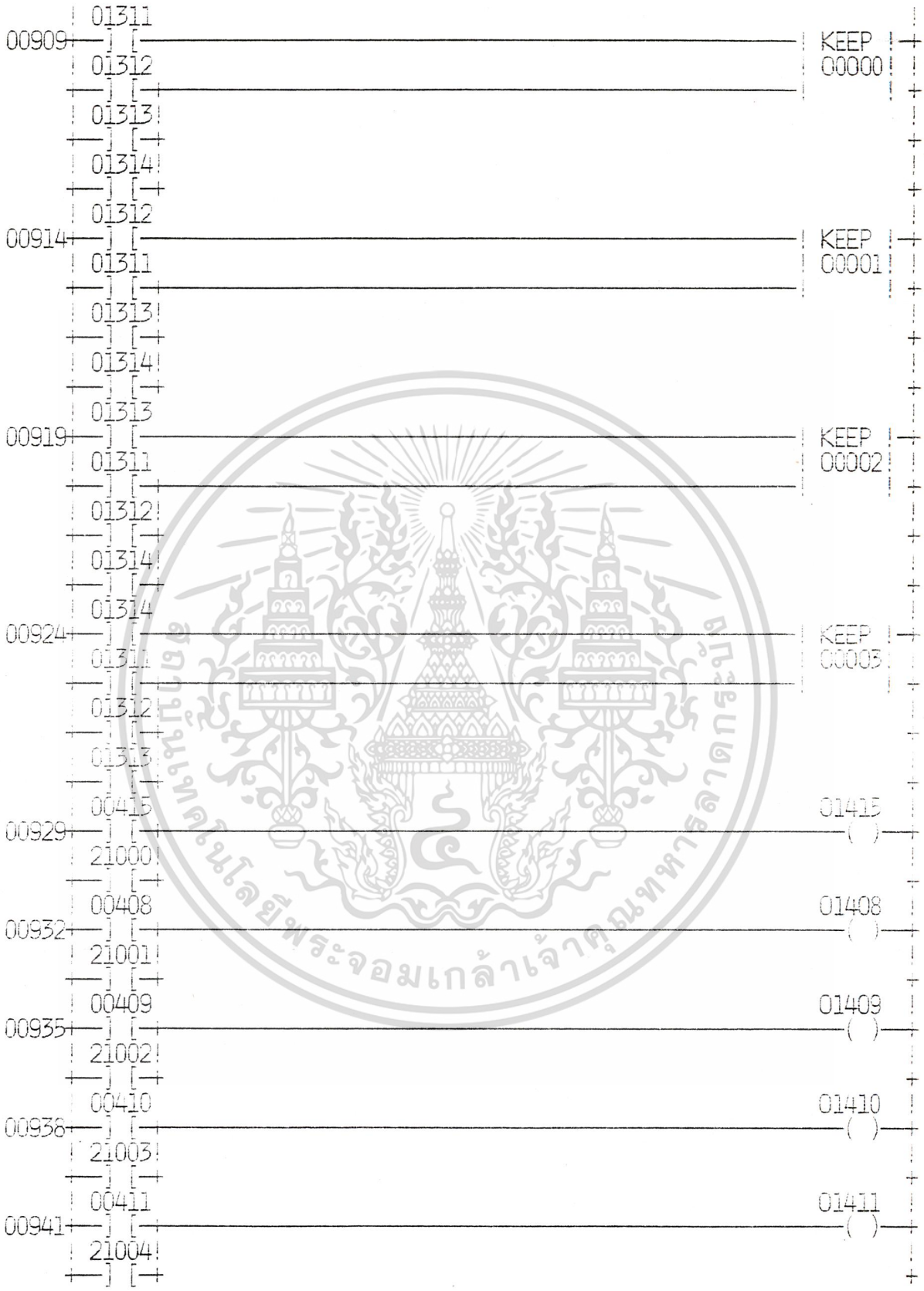
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



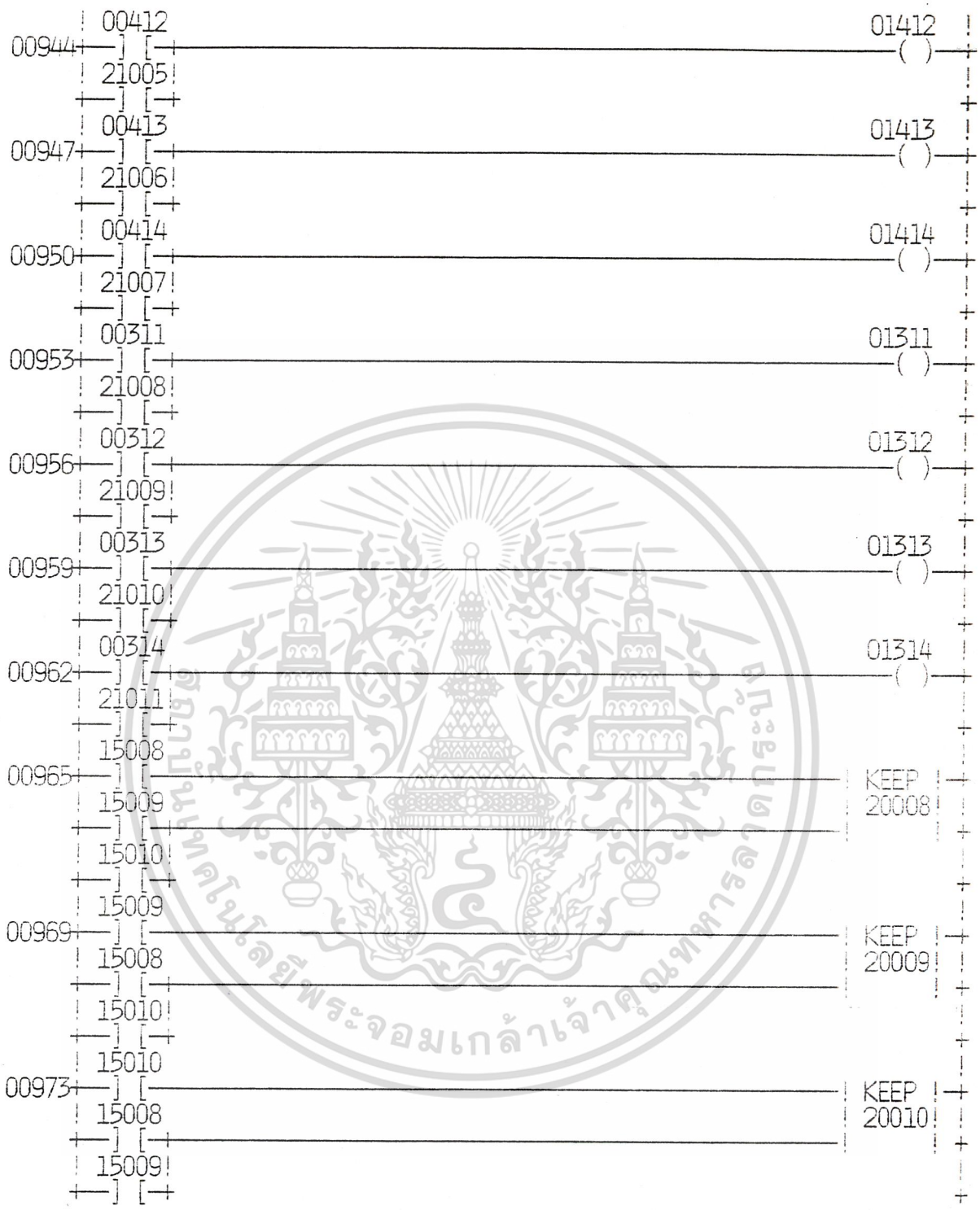
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



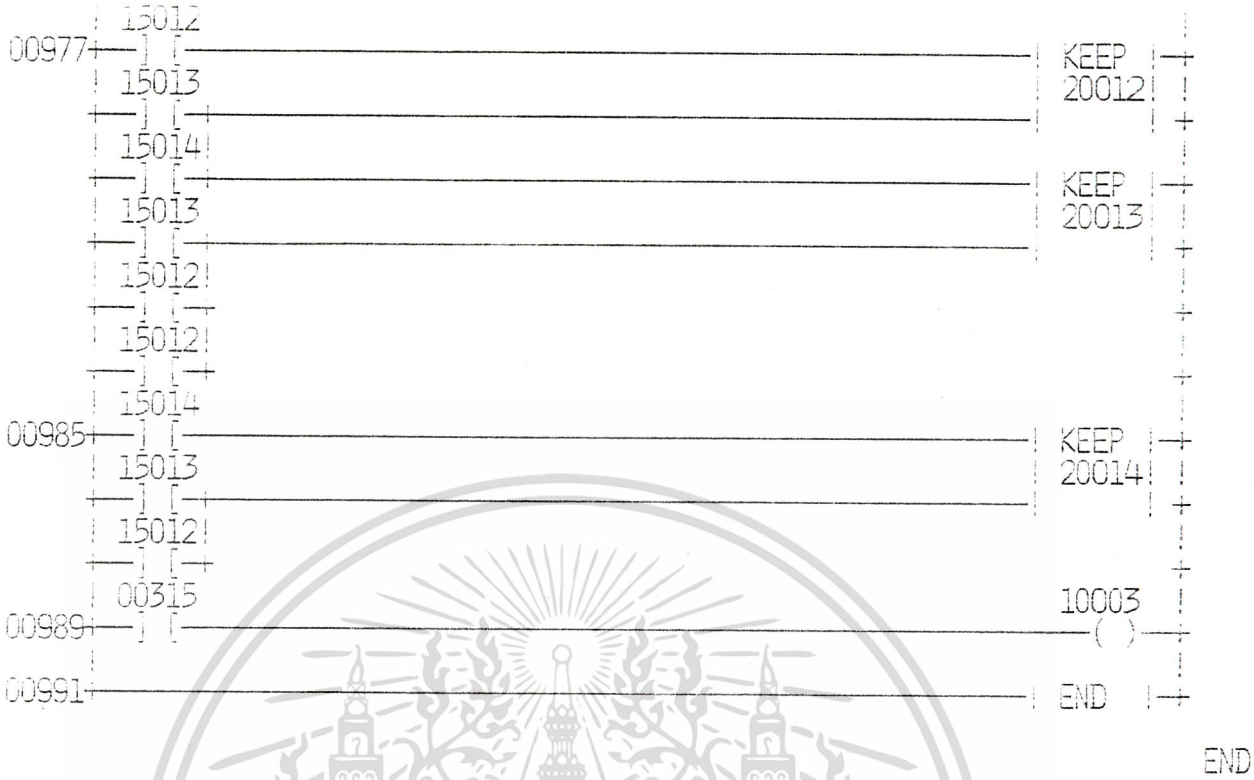
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



END

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก B

ASCII UNIT BASIC PROGRAM

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
LIST

10 OPEN #2,"COMU:(40)"
20 PRINT#2,"ATA"
30 FOR I =1 TO 15000
40 NEXT I
50 CLS #2
60 PRINT#2,"                THIS IS THE ASCII UNIT SYSTEM "
70 FOR I=1TO 2000
80 NEXT I
90 PRINT#2,"                CONTROL THE WAREHOUSE PROCESS "
100 FOR I=1 TO 2000
110 NEXT I
150 PRINT#2,"             THE PROJECT OF ENGINEERING FACULTY"
160 FOR I=1 TO 2000
170 NEXT I
180 PRINT#2,"             INSTRUMENTATION  TECHNOLOGY  DEPT."
190 FOR I=1 TO 2000
200 NEXT I
210 PRINT#2,"             KING MONGKUT'S INSTITUTE OF TECHNOLOGY"
220 FOR I=1TO 300
230 NEXT I
240 PRINT#2,"                        LADKRABANG "
250 FOR I=1 TO 2000
260 NEXT I
261 PRINT#2,""
262 PRINT#2,""
270 PRINT#2,""
271 PRINT#2,""
280 PRINT#2,"                        SELECT FUNCTION ITEMS "
290 PRINT#2,"                        ********************** "
300 PRINT#2,"                     1) PRESS [M] TO MONITOR MODE "
310 PRINT#2,"                     2) PRESS [C] TO CONTROL MODE "
320 PRINT#2,"                     3) PRESS [E] RETURN TO MAINMENU "
330 INPUT#2,"SELECT ITEMS";A$
340 IF A$="M" THEN GOTO 1000 ELSE 350
350 IF A$="C" THEN GOTO 2000 ELSE 360
360 GOTO 330
1000 CLS#2
1001 PRINT#2,""
1002 PRINT#2,"MONITOR MODE OK CLICK-GOTO MONITOR "
1005 IF "E"=INKEY$#2 THEN GOTO 261
1010 PC READ"@D,0,1,I4";A
1020 IF A=4101 THEN PRINT#2,"A" ELSE 1025
1025 IF A=1101 THEN PRINT#2,"A" ELSE 1030
1030 IF A=4102 THEN PRINT#2,"B" ELSE 1035
1035 IF A=1102 THEN PRINT#2,"B" ELSE 1040
1040 IF A=4104 THEN PRINT#2,"C" ELSE 1045
1045 IF A=1104 THEN PRINT#2,"C" ELSE 1050
1050 IF A=4108 THEN PRINT#2,"D" ELSE 1055
1055 IF A=1108 THEN PRINT#2,"D" ELSE 1060
1060 IF A=4110 THEN PRINT#2,"E" ELSE 1065
1065 IF A=1110 THEN PRINT#2,"E" ELSE 1070
1070 IF A=4120 THEN PRINT#2,"F" ELSE 1075
1075 IF A=1102 THEN PRINT#2,"F" ELSE 1080
1080 IF A=4140 THEN PRINT#2,"G" ELSE 1085
1085 IF A=1140 THEN PRINT#2,"G" ELSE 1090
1090 IF A=4201 THEN PRINT#2,"H" ELSE 1095
1095 IF A=1201 THEN PRINT#2,"H" ELSE 1100
1100 IF A=4202 THEN PRINT#2,"I" ELSE 1105
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น และไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
1105 IF A=1202 THEN PRINT#2,"I" ELSE 1110
1110 IF A=4204 THEN PRINT#2,"J" ELSE 1115
1115 IF A=1204 THEN PRINT#2,"J" ELSE 1120
1120 IF A=4208 THEN PRINT#2,"K" ELSE 1125
1125 IF A=1208 THEN PRINT#2,"K" ELSE 1130
1130 IF A=4210 THEN PRINT#2,"L" ELSE 1135
1135 IF A=1210 THEN PRINT#2,"L" ELSE 1140
1140 IF A=4220 THEN PRINT#2,"M" ELSE 1145
1145 IF A=1220 THEN PRINT#2,"M" ELSE 1150
1150 IF A=4240 THEN PRINT#2,"N" ELSE 1155
1155 IF A=1240 THEN PRINT#2,"N" ELSE 1160
1160 IF A=4401 THEN PRINT#2,"O" ELSE 1165
1165 IF A=1401 THEN PRINT#2,"O" ELSE 1170
1170 IF A=4402 THEN PRINT#2,"P" ELSE 1175
1175 IF A=1402 THEN PRINT#2,"P" ELSE 1180
1180 IF A=4404 THEN PRINT#2,"Q" ELSE 1185
1185 IF A=1404 THEN PRINT#2,"Q" ELSE 1190
1190 IF A=4408 THEN PRINT#2,"R" ELSE 1195
1195 IF A=1408 THEN PRINT#2,"R" ELSE 1200
1200 IF A=4410 THEN PRINT#2,"S" ELSE 1205
1205 IF A=1410 THEN PRINT#2,"S" ELSE 1210
1210 IF A=4420 THEN PRINT#2,"T" ELSE 1215
1215 IF A=1420 THEN PRINT#2,"T" ELSE 1220
1220 IF A=4440 THEN PRINT#2,"U" ELSE 1225
1225 IF A=1440 THEN PRINT#2,"U" ELSE 1230
1230 IF A=2101 THEN PRINT#2,"V" ELSE 1240
1240 GOTO 1005
1300 REM **********************END OF MONITOR ********
2000 PRINT#2,""
2010 PRINT#2,""
2020 PRINT#2,""
2030 PRINT#2,""
2040 PRINT#2,"                    SELECT MODE CONTROL "
2050 PRINT#2,"                    ************************"
2060 PRINT#2,"                  [1] MOVE MODE"
2070 PRINT#2,"                  [2] SEND MODE"
2080 PRINT#2,"                  [3] AUTO MODE"
2090 PRINT#2,"                  [4] RETRUN TO MAIN MENU"
2100 INPUT#2,"    SELECT NUMBER";B
2120 IF B=1 THEN GOTO 3000 ELSE 2125
2125 IF B=2 THEN GOTO 4000 ELSE 2130
2130 IF B=3 THEN GOTO 5000 ELSE 2135
2135 IF B=4 THEN GOTO 261  ELSE 2100
3000 PRINT#2,""
3001 R=1
3002 M=400
3003 O=0
3010 PRINT#2,""
3020 PRINT#2,"    ENTER 2 POSITION (SOUCE || DESTINATION) "
3030 INPUT#2,"ENTER SOUCE XX ";C
3031 GOSUB 13050
3040 INPUT#2,"ENTER DESTINATION XX";D
3041 GOSUB 23050
3050 PCWRITE"@D,1,2,I4";M
3060 FOR I=1 TO 500
3070 NEXT I
3080 PCWRITE"@D,1,2,I4";O
3090 FOR I=1 TO 500
3100 NEXT I
3110 PCWRITE"@D,1,2,I4";R
3120 FOR I=1 TO 500
3130 NEXT I
3140 PCWRITE"@D,1,2,I4";O
3150 FOR I=1 TO 500
3160 NEXT I
```

เอกสารนี้สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น การดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
3170 PCWRITE"@D,1,2,I4";E
3180 FOR I=1 TO 500
3190 NEXT I
3200 PCWRITE"@D,1,2,I4";O
3210 FOR I=1 TO 500
3220 NEXT I
3230 PCWRITE"@D,1,2,I4";F
3240 FOR I=1 TO 500
3250 NEXT I
3260 PCWRITE"@D,1,2,I4";O
3270 FOR I=1 TO 500
3280 NEXT I
3290 PCWRITE"@D,1,2,I4";G
3300 FOR I=1 TO 500
3302 NEXT I
3320 PCWRITE"@D,1,2,I4";O
3330 FOR I=1 TO 500
3340 NEXT I
3350 PCWRITE"@D,1,2,I4";H
3360 FOR I=1 TO 500
3370 NEXT I
3380 PCWRITE"@D,1,2,I4";O
3390 FOR I=1 TO 500
3400 NEXT I
3405 PRINT#2,"WRITE COMPLETE"
3410 GOTO 2000
3450
4000 PRINT#2,""
4001 R=1
4002 M=100
4003 O=0
4010 PRINT#2,""
4020 PRINT#2,"    ENTER POSITION + SENDING MODE"
4030 INPUT#2,"ENTER DESTINATION XX YY",
4042 GOSUB 13050
4050 PCWRITE"@D,1,2,I4";M
4060 FOR I=1 TO 500
4070 NEXT I
4080 PCWRITE"@D,1,2,I4";O
4090 FOR I=1 TO 500
4100 NEXT I
4110 PCWRITE"@D,1,2,I4";R
4120 FOR I=1 TO 500
4130 NEXT I
4140 PCWRITE"@D,1,2,I4";O
4150 FOR I=1 TO 500
4160 NEXT I
4170 PCWRITE"@D,1,2,I4";E
4180 FOR I=1 TO 500
4190 NEXT I
4200 PCWRITE"@D,1,2,I4";O
4210 FOR I=1 TO 500
4220 NEXT I
4230 PCWRITE"@D,1,2,I4";F
4240 FOR I=1 TO 500
4250 NEXT I
4260 PCWRITE"@D,1,2,I4";O
4270 FOR I=1 TO 500
4280 NEXT I
4281 PRINT#2,"WRITE COMPLETE"
4290 GOTO 2000
5000 PRINT#2,""
5001 M=800
5010 PRINT#2,""
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
5020 PRINT#2,"AUTO MODE RUNNING  OK!"
5050 PCWRITE"@D,1,2,I4";M
5060 FOR I=1 TO 500
5070 NEXT I
5080 PCWRITE"@D,1,2,I4";0
5090 FOR I=1 TO 500
5100 NEXT I
5105 PRINT#2,"  DATA OK!  "
5110 GOTO 2000
13000 REM ***********SOUCE DATA***************
13050 IF C=11 THEN GOTO 13051 ELSE 13055
13051 E=2
13052 F=2
13055 IF C=12 THEN GOTO 13056 ELSE 13060
13056 E=2
13057 F=4
13060 IF C=13 THEN GOTO 13061 ELSE 13065
13061 E=2
13062 F=8
13065 IF C=14 THEN GOTO 13066 ELSE 13070
13066 E=2
13067 F=4
13070 IF C=15 THEN GOTO 13071 ELSE 13075
13071 E=2
13072 F=20
13075 IF C=16 THEN GOTO 13076 ELSE 13080
13076 E=2
13077 F=40
13080 IF C=17 THEN GOTO 13081 ELSE 13085
13081 E=2
13082 F=80
13085 IF C=21 THEN GOTO 13086 ELSE 13090
13086 E=4
13087 F=2
13090 IF C=22 THEN GOTO 13091 ELSE 13095
13091 E=4
13092 F=4
13095 IF C=23 THEN GOTO 13096 ELSE 13100
13096 E=4
13097 F=8
13100 IF C=24 THEN GOTO 13101 ELSE 13105
13101 E=4
13102 F=10
13105 IF C=25 THEN GOTO 13106 ELSE 13110
13106 E=4
13107 F=20
13110 IF C=26 THEN GOTO 13111 ELSE 13115
13111 E=4
13112 F=40
13115 IF C=27 THEN GOTO 13116 ELSE 13120
13116 E=4
13117 F=80
13120 IF C=31 THEN GOTO 13121 ELSE 13125
13121 E=8
13122 F=2
13125 IF C=32 THEN GOTO 13126 ELSE 13130
13126 E=8
13127 F=4
13130 IF C=33 THEN GOTO 13131 ELSE 13135
13131 E=8
13132 F=8
13135 IF C=34 THEN GOTO 13136 ELSE 13140
13136 E=8
13137 F=10
13140 IF C=35 THEN GOTO 13141 ELSE 13145
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
13141 E=8
13142 F=20
13145 IF C=36 THEN GOTO 13146 ELSE 13150
13146 E=8
13147 F=40
13150 IF C=37 THEN GOTO 13151 ELSE 13155
13151 E=8
13152 F=80
13155 PRINT #2,"SOUCE DATA OK! "C
13157 RETURN
23000  REM**********DESTINATION DATA**********
23050 IF D=11 THEN GOTO 23051 ELSE 23055
23051 G=2
23052 H=2
23055 IF D=12 THEN GOTO 23056 ELSE 23060
23056 G=2
23057 H=4
23060 IF D=13 THEN GOTO 23061 ELSE 23065
23061 G=2
23062 H=8
23065 IF D=14 THEN GOTO 23066 ELSE 23070
23066 G=2
23067 H=10
23070 IF D=15 THEN GOTO 23071 ELSE 23075
23071 G=2
23072 H=20
23075 IF D=16 THEN GOTO 23076 ELSE 23080
23076 G=2
23077 H=40
23080 IF D=17 THEN GOTO 23081 ELSE 23085
23081 G=2
23082 H=80
23085 IF D=21 THEN GOTO 23086 ELSE 23090
23086 G=4
23087 H=2
23090 IF D=22 THEN GOTO 23091 ELSE 23095
23091 G=4
23092 H=4
23095 IF D=23 THEN GOTO 23096 ELSE 23100
23096 G=4
23097 H=8
23100 IF D=24 THEN GOTO 23101 ELSE 23105
23101 G=4
23102 H=10
23105 IF D=25 THEN GOTO 23106 ELSE 23110
23106 G=4
23107 H=20
23110 IF D=26 THEN GOTO 23111 ELSE 23115
23111 G=4
23112 H=40
23115 IF D=27 THEN GOTO 23116 ELSE 23120
23116 G=4
23117 H=80
23120 IF D=31 THEN GOTO 23121 ELSE 23125
23121 G=8
23122 H=2
23125 IF D=32 THEN GOTO 23126 ELSE 23130
23126 G=8
23127 H=4
23130 IF D=33 THEN GOTO 23131 ELSE 23135
23131 G=8
23132 H=8
23135 IF D=34 THEN GOTO 23136 ELSE 23140
23136 G=8
23137 H=10
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น และไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
23140 IF D=35 THEN GOTO 23141 ELSE 23145
23141 G=8
23142 H=20
23145 IF D=36 THEN GOTO 23146 ELSE 23150
23146 G=8
23147 H=40
23150 IF D=37 THEN GOTO 23151 ELSE 23155
23151 G=8
23152 H=80
23155 PRINT#2," DESTINATION DATA OK ! "  D
23157 RETURN
READY
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก C

COMMUNICATION PROGRAM

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
ON LOCAL ERROR RESUME NEXT
   IF NOT FORM5.MSCOMM1.PORTOPEN THEN
     FORM5.MSCOMM1.PORTOPEN = TRUE
    END IF

   FORM5.SHOW
   FORM5.COMMAND1.ENABLED = TRUE


END SUB
SUB IMAGE1_CLICK ()
 FORM2.SHOW

END SUB
SUB MCONTROL_CLICK ()
 ON LOCAL ERROR RESUME NEXT
   IF NOT FORM5.MSCOMM1.PORTOPEN THEN
     FORM5.MSCOMM1.PORTOPEN = TRUE
    END IF

   FORM5.SHOW
   FORM5.COMMAND1.ENABLED = TRUE


END SUB
SUB MDIAL_CLICK ()
   ON LOCAL ERROR RESUME NEXT
   STATIC NUM$
    NUM$ = INPUTBOX$("ENTER PHONE NUMBER:", "DIAL NUMBER", NUM$)
      IF NUM$ = "" THEN EXIT SUB
      IF NOT FORM5.MSCOMM1.PORTOPEN THEN
       FORM5.MSCOMM1.PORTOPEN = TRUE
       END IF
       FORM5.MSCOMM1.OUTPUT = "ATDT" + NUM$ + CHR$(13) + CHR$(10)

END SUB
SUB MEXIT_CLICK ()
   IF FORM5.MSCOMM1.PORTOPEN THEN
     FORM5.MSCOMM1.PORTOPEN = FALSE
    END IF

 UNLOAD FORM1
 UNLOAD FORM2
 UNLOAD FORM3
 UNLOAD FORM4
 UNLOAD FORM5

END SUB
SUB MHANG_CLICK ()
 RET = FORM5.MSCOMM1.DTRENABLE
 FORM5.MSCOMM1.DTRENABLE = TRUE
 FORM5.MSCOMM1.DTRENABLE = FALSE
 FORM5.MSCOMM1.DTRENABLE = RET
 END SUB
SUB MHELP_CLICK ()
 FORM3.SHOW

END SUB
SUB MSETTINGS_CLICK ()
FORM4.SHOW
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
END SUB
SUB IMAGE1_CLICK ()
 UNLOAD FORM2

END SUB
SUB MDEV_CLICK ()

    IMAGE1.VISIBLE = FALSE
    IMAGE2.VISIBLE = FALSE
    IMAGE3.VISIBLE = TRUE

END SUB
SUB MHEXIT_CLICK ()
UNLOAD FORM3

END SUB
SUB MPROG_CLICK ()

    IMAGE1.VISIBLE = FALSE
    IMAGE2.VISIBLE = TRUE
    IMAGE3.VISIBLE = FALSE
END SUB
SUB MPROJ_CLICK ()
    IMAGE1.VISIBLE = TRUE
    IMAGE2.VISIBLE = FALSE
    IMAGE3.VISIBLE = FALSE

END SUB
SUB BAUD12_CLICK ()

    NEWBAUD$ = "1200"

END SUB
SUB BAUD24_CLICK ()

    NEWBAUD$ = "2400"


END SUB
SUB BAUD3_CLICK ()

    NEWBAUD$ = "300"


END SUB
SUB BAUD48_CLICK ()

    NEWBAUD$ = "4800"


END SUB
SUB BAUD6_CLICK ()

    NEWBAUD$ = "600"


END SUB
SUB BAUD96_CLICK ()

    NEWBAUD$ = "9600"
END SUB
SUB BOTHFLOW_CLICK ()
NEWSHAKE = 3
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
END SUB
SUB CANCELBUTTON_CLICK ()
UNLOAD FORM4

END SUB
SUB COMPORT_CLICK (INDEX AS INTEGER)
NEWPORT = INDEX

END SUB
SUB DATA7_CLICK ()
NEWDATA$ = "7"


END SUB
SUB DATA8_CLICK ()
 NEWDATA$ = "8"

END SUB
SUB ECHOOFF_CLICK ()
ECHO = 0

END SUB
SUB ECHOON_CLICK ()
ECHO = TRUE

END SUB
SUB EVENPARITY_CLICK ()
NEWPARITY$ = "E"

END SUB
SUB FORM_LOAD ()

    '—— GET CURRENT PORT
    PORT = FORM5.MSCOMM1.COMMPORT
    FORM4.COMPORT(PORT).VALUE = TRUE        'SET OPTION BUTTON

    '—— GET CURRENT BAUD
    FIRSTCOMMA = INSTR(FORM5.MSCOMM1.SETTINGS, ",")
    BAUD$ = LEFT$(FORM5.MSCOMM1.SETTINGS, FIRSTCOMMA - 1)

    SELECT CASE VAL(BAUD$)                  'SELECT BAUD
    CASE 300                                'SET ACTIVE BAUD
        FORM4.BAUD3.VALUE = TRUE            'OPTION BUTTON
    CASE 600
        FORM4.BAUD6.VALUE = TRUE
    CASE 1200
        FORM4.BAUD12.VALUE = TRUE
    CASE 2400
        FORM4.BAUD24.VALUE = TRUE
    CASE 4800
        FORM4.BAUD48.VALUE = TRUE
    CASE 9600
        FORM4.BAUD96.VALUE = TRUE
    END SELECT

    '—— GET CURRENT PARITY
    PARITY$ = MID$(FORM5.MSCOMM1.SETTINGS, FIRSTCOMMA + 1, 1)

    SELECT CASE UCASE$(PARITY$)             'SELECT PARITY
    CASE "N"                                'SET ACTIVE PARITY
        FORM4.NOPARITY.VALUE = TRUE         'OPTION BUTTON
    CASE "E"
        FORM4.EVENPARITY.VALUE = TRUE
    CASE "O"
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        Form4.OddParity.Value = True
    End Select


    '—— Get data bits
    SecondComma = FirstComma + 2
    DBits$ = Mid$(Form5.MSComm1.Settings, SecondComma + 1, 1)
    Select Case Val(DBits$)                 'select data bits
    Case 7                                  'set active choice
        Form4.Data7.Value = True            'option button
    Case 8
        Form4.Data8.Value = True
    End Select

    '—— Get stop bits
    ThirdComma = SecondComma + 2
    SBits$ = Mid$(Form5.MSComm1.Settings, ThirdComma + 1, 1)
    Select Case Val(SBits$)                 'select stop bits
    Case 1                                  'set active choice
        Form4.Stop1.Value = True            'option button
    Case 2
        Form4.Stop2.Value = True
    End Select


    Select Case Form5.MSComm1.Handshaking
    Case 0                                  'set active choice
        Form4.NoFlow.Value = True           'option button
    Case 1
        Form4.XonFlow.Value = True
    Case 2
        Form4.RTSFlow.Value = True
    Case 3
        Form4.BothFlow.Value = True
    End Select

    If Echo Then
        Form4.EchoOn.Value = True
    Else
        Form4.EchoOff.Value = True
    End If

End Sub
Sub NoFlow_Click ()
NewShake = 0

End Sub
Sub NoParity_Click ()
NewParity$ = "N"

End Sub
Sub OddParity_Click ()
NewParity$ = "O"

End Sub
Sub OKButton_Click ()
    On Error Resume Next

    OldPort = Form5.MSComm1.CommPort
    If NewPort <> OldPort Then                  'if port num changing
        If Form5.MSComm1.PortOpen Then          'close the old port
            Form5.MSComm1.PortOpen = False
            ReOpen = True
        End If
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น ไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        Form5.MSComm1.CommPort = NEWPORT            'SET NEW PORT NUMBER

        If Err = 0 Then
           If REOpen Then
              Form5.MSComm1.PortOpen = True
           End If
        End If
        If Err Then
           MsgBox Error$, 48
           Form5.MSComm1.CommPort = OLDPort
           Exit Sub
        End If
     End If

     Form5.MSComm1.Settings = NEWBAUD$ + ","
           + NEWPARITY$ + "," + NEWDATA$ + "," + NEWSTOP$
     If Err Then
        MsgBox Error$, 48
        Exit Sub
     End If
     Form5.MSComm1.Handshaking = NEWSHAKE
     If Err Then
        MsgBox Error$, 48
        Exit Sub
     End If

     Unload Form4                               'REMOVE CONFIGURATION FORM


End Sub
Sub RTSFlow_Click ()
NEWSHAKE = 2

End Sub
Sub Stop1_Click ()
NEWSTOP$ = "1"

End Sub
Sub Stop2_Click ()
NEWSTOP$ = "2"

End Sub
Sub XonFlow_Click ()
NEWSHAKE = 1

End Sub
DefInt A-Z

Option Explicit

Dim RET                     'SCRATCH INTEGER
Dim TEMP$                   'SCRATCH STRING
Sub CLR ()
Form1.P11.BackColor = Form1.BackColor
Form1.P12.BackColor = Form1.BackColor
Form1.P13.BackColor = Form1.BackColor
Form1.P14.BackColor = Form1.BackColor
Form1.P15.BackColor = Form1.BackColor
Form1.P16.BackColor = Form1.BackColor
Form1.P17.BackColor = Form1.BackColor
Form1.P21.BackColor = Form1.BackColor
Form1.P22.BackColor = Form1.BackColor
Form1.P23.BackColor = Form1.BackColor
Form1.P24.BackColor = Form1.BackColor
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น และไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
FORM1.P25.BACKCOLOR = FORM1.BACKCOLOR
FORM1.P26.BACKCOLOR = FORM1.BACKCOLOR
FORM1.P27.BACKCOLOR = FORM1.BACKCOLOR
FORM1.P31.BACKCOLOR = FORM1.BACKCOLOR
FORM1.P32.BACKCOLOR = FORM1.BACKCOLOR
FORM1.P33.BACKCOLOR = FORM1.BACKCOLOR
FORM1.P34.BACKCOLOR = FORM1.BACKCOLOR
FORM1.P35.BACKCOLOR = FORM1.BACKCOLOR
FORM1.P36.BACKCOLOR = FORM1.BACKCOLOR
FORM1.P37.BACKCOLOR = FORM1.BACKCOLOR
FORM1.PCON.BACKCOLOR = FORM1.BACKCOLOR
 END SUB
STATIC SUB SHOWDATA (TERM AS CONTROL, DTA$)
    ON ERROR RESUME NEXT
    DIM ND, I

    '--- MAKE SURE THE EXISTING TEXT DOESN'T GET
    '    TOO LARGE.
    ND = LEN(TERM.TEXT)
    IF ND >= 16384 THEN
       TERM.TEXT = MID$(TERM.TEXT, 4097)
       ND = LEN(TERM.TEXT)
    END IF

    '--- POINT TO THE END OF TERM'S DATA
    TERM.SELSTART = ND

    '--- FILTER/HANDLE BACK SPACE CHARACTERS
    DO
       I = INSTR(DTA$, CHR$(8))
       IF I THEN
          IF I = 1 THEN
             TERM.SELSTART = ND - 1
             TERM.SELLENGTH = 1
             DTA$ = MID$(DTA$, I + 1)
          ELSE
             DTA$ = LEFT$(DTA$, I - 2) + MID$(DTA$, I + 1)
          END IF
       END IF
    LOOP WHILE I

    '--- ELLIMINATE LINE FEEDS (PUT BACK BELOW)
    DO
       I = INSTR(DTA$, CHR$(10))
       IF I THEN
          DTA$ = LEFT$(DTA$, I - 1) + MID$(DTA$, I + 1)
       END IF
    LOOP WHILE I

    '--- MAKE SURE ALL CHARRIAGE RETURNS HAVE A
    '    LINE FEED
    I = 1
    DO
       I = INSTR(I, DTA$, CHR$(13))
       IF I THEN
          DTA$ = LEFT$(DTA$, I) + CHR$(10) + MID$(DTA$, I + 1)
          I = I + 1
       END IF
    LOOP WHILE I

    '--- ADD THE FILTERED DATA TO .TEXT
    TERM.SELTEXT = DTA$
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
END SUB
```

```
SUB COMMAND1_CLICK ()
    COMMAND1.ENABLED = FALSE
    ECHO = TRUE
    FORM1.SHOW
  DIM INSTRING$
     'OPEN PORT


  ON LOCAL ERROR RESUME NEXT
     IF NOT FORM5.MSCOMM1.PORTOPEN THEN
       FORM5.MSCOMM1.PORTOPEN = TRUE
     END IF
   FORM1.X.CAPTION = "POSITION"

  DO WHILE FORM5.MSCOMM1.PORTOPEN

 RET = DOEVENTS()

     IF FORM5.MSCOMM1.INBUFFERCOUNT THEN
       INSTRING$ = FORM5.MSCOMM1.INPUT
    ' ================ 11

     IF INSTRING$ = "A" THEN
        CLR

         FORM1.Y.CAPTION = "1 1"
      FORM1.P11.BACKCOLOR = FORM1.GREEN.BACKCOLOR
        END IF
       ' P11.CAPTION = "MID"
     IF INSTRING$ = "A" THEN

       FORM1.P11.BACKCOLOR = FORM1.RED.BACKCOLOR
        END IF

    '================12
     IF INSTRING$ = "B" THEN
        CLR
        FORM1.Y.CAPTION = "1 2"
        FORM1.P12.BACKCOLOR = FORM1.GREEN.BACKCOLOR
          END IF
      IF INSTRING$ = "B" THEN
           ' P12.CAPTION = "MID"
        FORM1.P12.BACKCOLOR = FORM1.RED.BACKCOLOR
            'P12.CAPTION = "IN"
          END IF
     '================13

     IF 'INSTRING$ = "C" THEN
        CLR
          FORM1.Y.CAPTION = "1 3"
          FORM1.P13.BACKCOLOR = FORM1.GREEN.BACKCOLOR
          END IF
         'P13.CAPTION = "MID"
      IF INSTRING$ = "C" THEN
         FORM1.P13.BACKCOLOR = FORM1.RED.BACKCOLOR
          ' P13.CAPTION = "IN"
          END IF
      '================14
     IF INSTRING$ = "D" THEN
     CLR
          FORM1.Y.CAPTION = "1 4"
          FORM1.P14.BACKCOLOR = FORM1.GREEN.BACKCOLOR
          ' P14.CAPTION = "MID"
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น รวมทั้งไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเอกสารทุกครั้งที่มีการนำไปใช้

```
    END IF
IF INSTRING$ = "D" THEN
    FORM1.P14.BACKCOLOR = FORM1.RED.BACKCOLOR
  'P14.CAPTION = "IN"
    END IF

  '================15
IF INSTRING$ = "E" THEN
    CLR
    FORM1.Y.CAPTION = "1 5"
    FORM1.P15.BACKCOLOR = FORM1.GREEN.BACKCOLOR
  ' P15.CAPTION = "MID"
    END IF
 IF INSTRING$ = "E" THEN
    FORM1.P15.BACKCOLOR = FORM1.RED.BACKCOLOR
  'P15.CAPTION = "IN"
    END IF

  '================16
IF INSTRING$ = "F" THEN
  CLR
    FORM1.Y.CAPTION = "1 6"
    FORM1.P16.BACKCOLOR = FORM1.GREEN.BACKCOLOR
  ' P16.CAPTION = "MID"
    END IF
IF INSTRING$ = "F" THEN
    FORM1.P16.BACKCOLOR = FORM1.RED.BACKCOLOR
    ' P16.CAPTION = "IN"
    END IF
  '================17
IF INSTRING$ = "G" THEN
  CLR
    FORM1.Y.CAPTION = "1 7"
    FORM1.P17.BACKCOLOR = FORM1.GREEN.BACKCOLOR
   'P17.CAPTION = "MID"
    END IF
IF INSTRING$ = "G" THEN
    FORM1.P17.BACKCOLOR = FORM1.RED.BACKCOLOR
    END IF
  '================21
IF INSTRING$ = "H" THEN
  CLR
    FORM1.Y.CAPTION = "2 1"
    FORM1.P21.BACKCOLOR = FORM1.GREEN.BACKCOLOR
    END IF
IF INSTRING$ = "H" THEN
    FORM1.P21.BACKCOLOR = FORM1.RED.BACKCOLOR
  'P21.CAPTION = "IN"
    END IF

  '================22
IF INSTRING$ = "I" THEN
  CLR
    FORM1.Y.CAPTION = "2 2"
    FORM1.P22.BACKCOLOR = FORM1.GREEN.BACKCOLOR
    END IF
IF INSTRING$ = "I" THEN
    FORM1.P22.BACKCOLOR = FORM1.RED.BACKCOLOR
   'P22.CAPTION = "IN"
    END IF
  '================23
IF INSTRING$ = "J" THEN
  CLR
    FORM1.Y.CAPTION = "2 3"
    FORM1.P23.BACKCOLOR = FORM1.GREEN.BACKCOLOR
    P23.CAPTION = "MID"
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น

```
        End If
    If INSTRING$ = "J" Then
        Form1.P23.BackColor = Form1.RED.BackColor
    'P23.Caption = "IN"
        End If
    '=====================24
IF INSTRING$ = "K" Then
    CLR
        Form1.Y.Caption = "2 4"
        Form1.P24.BackColor = Form1.GREEN.BackColor
        End If
IF INSTRING$ = "K" Then
        Form1.P24.BackColor = Form1.RED.BackColor
    'P24.Caption = "IN"
        End If
    '=====================25
IF INSTRING$ = "L" Then
    CLR
        Form1.Y.Caption = "2 5"
        Form1.P25.BackColor = Form1.GREEN.BackColor
     'P25.Caption = "MID"
        End If
IF INSTRING$ = "L" Then
        Form1.P25.BackColor = Form1.RED.BackColor
    'P26.Caption = "IN"
        End If

    '=====================26
IF INSTRING$ = "M" Then
    CLR
        Form1.Y.Caption = "2 6"
        Form1.P26.BackColor = Form1.GREEN.BackColor
      P26.Caption = "MID"
        End If
    If INSTRING$ = "M" Then
        Form1.P26.BackColor = Form1.RED.BackColor
        End If
    '=====================27
IF INSTRING$ = "N" Then
    CLR
        Form1.Y.Caption = "2 7"
        Form1.P27.BackColor = Form1.GREEN.BackColor
        End If
    If INSTRING$ = "N" Then
        Form1.P27.BackColor = Form1.RED.BackColor
    'P27.Caption = "IN"
        End If
    '=====================31
IF INSTRING$ = "O" Then
    CLR
        Form1.Y.Caption = "3 1"
        Form1.P31.BackColor = Form1.GREEN.BackColor
        End If
IF INSTRING$ = "O" Then
        Form1.P31.BackColor = Form1.RED.BackColor
    'P31.Caption = "IN"
        End If
    '=====================32
IF INSTRING$ = "P" Then
    CLR
        Form1.Y.Caption = "3 2"
        Form1.P32.BackColor = Form1.GREEN.BackColor
      P32.Caption = "MID"
        End If
IF INSTRING$ = "P" Then
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        Form1.P32.BackColor = Form1.RED.BackColor
    'P32.Caption = "IN"
        End If
 '====================33
IF INSTRING$ = "Q" THEN
    CLR
        Form1.Y.Caption = "3 3"
        Form1.P33.BackColor = Form1.GREEN.BackColor
    ' P33.Caption = "MID"
        End If
 IF INSTRING$ = "q" THEN
        Form1.P33.BackColor = Form1.RED.BackColor
    'P33.Caption = "IN"
        End If
 '====================34
IF INSTRING$ = "R" THEN
    CLR
        Form1.Y.Caption = "3 4"
        Form1.P34.BackColor = Form1.GREEN.BackColor
    ' P34.Caption = "MID"
        End If
 IF INSTRING$ = "r" THEN
        Form1.P34.BackColor = Form1.RED.BackColor
    'P34.Caption = "IN"
        End If
 '====================35
IF INSTRING$ = "S" THEN
    CLR
         Form1.Y.Caption = "3 5"
         Form1.P35.BackColor = Form1.GREEN.BackColor
    ' P35.Caption = "MID"
        End If
 IF INSTRING$ = "s" THEN
 Form1.P35.BackColor = Form1.RED.BackColor
    'P35.Caption = "IN"
        End If
  '====================36
IF INSTRING$ = "T" THEN
     CLR
        Form1.Y.Caption = "3 6"
        Form1.P36.BackColor = Form1.GREEN.BackColor
     'P36.Caption = "MID"
        End If
IF INSTRING$ = "t" THEN
        Form1.P36.BackColor = Form1.RED.BackColor
    'P36.Caption = "IN"
        End If
 '====================37
IF INSTRING$ = "U" THEN
    CLR
        Form1.Y.Caption = "3 7"
        Form1.P37.BackColor = Form1.GREEN.BackColor
     'P37.Caption = "MID"
        End If
IF INSTRING$ = "u" THEN
         Form1.P37.BackColor = Form1.RED.BackColor
    'P37.Caption = "IN"
        End If
 '====================PCONVEGER
IF INSTRING$ = "V" THEN
    CLR
     Form1.Y.Caption = "CONVEGER"
     'PCON.Caption = "ON"
     Form1.P11.BackColor = Form1.RED.BackColor
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        END IF

    END IF
    LOOP

END SUB
 SUB MSCOMM1_ONCOMM ()
    DIM EVMSG$
    DIM ERMSG$
       IF COMMAND1.ENABLED = TRUE THEN


    '—— BRANCH ACCORDING TO THE COMMEVENT PROP.,
    SELECT CASE MSCOMM1.COMMEVENT
        '—— EVENT MESSAGES
        CASE MSCOMM_EV_RECEIVE
            SHOWDATA TERM, (MSCOMM1.INPUT)
        CASE MSCOMM_EV_SEND

        CASE MSCOMM_EV_CTS
            EVMSG$ = "CHANGE IN CTS DETECTED"
        CASE MSCOMM_EV_DSR
            EVMSG$ = "CHANGE IN DSR DETECTED"
        CASE MSCOMM_EV_CD
            EVMSG$ = "CHANGE IN CD DETECTED"
        CASE MSCOMM_EV_RING
            EVMSG$ = "THE PHONE IS RINGING"
        CASE MSCOMM_EV_EOF
            EVMSG$ = "END OF FILE DETECTED"

        '—— ERROR MESSAGES
        CASE MSCOMM_ER_BREAK
            EVMSG$ = "BREAK RECEIVED"
        CASE MSCOMM_ER_CTSTO
            ERMSG$ = "CTS TIMEOUT"
        CASE MSCOMM_ER_DSRTO
            ERMSG$ = "DSR TIMEOUT"
        CASE MSCOMM_ER_FRAME
            EVMSG$ = "FRAMING ERROR"
        CASE MSCOMM_ER_OVERRUN
            ERMSG$ = "OVERRUN ERROR"
        CASE MSCOMM_ER_CDTO
            ERMSG$ = "CARRIER DETECT TIMEOUT"
        CASE MSCOMM_ER_RXOVER
            ERMSG$ = "RECEIVE BUFFER OVERFLOW"
        CASE MSCOMM_ER_RXPARITY
            EVMSG$ = "PARITY ERROR"
        CASE MSCOMM_ER_TXFULL
            ERMSG$ = "TRANSMIT BUFFER FULL"
        CASE ELSE
            ERMSG$ = "UNKNOWN ERROR OR EVENT"
    END SELECT

    IF LEN(EVMSG$) THEN
        '—— DISPLAY EVENT MESSAGES IN LABEL
        LABEL1.CAPTION = EVMSG$
        EVMSG$ = ""
    ELSEIF LEN(ERMSG$) THEN
        '—— DISPLAY ERROR MESSAGES IN AN ALERT
        '     MESSAGE BOX.
        BEEP
        RET = MSGBOX(ERMSG$, 1, "PRESS CANCEL TO QUIT, OK TO IGNORE.")
        ERMSG$ = ""
        '—— IF CANCEL (2) WAS PRESSED
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น ทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        IF RET = 2 THEN
            MSComm1.PortOpen = 0     'CLOSE THE PORT AND QUIT
        END IF
    END IF

    END IF


END SUB
SUB Term_KeyPress (KeyAscii AS INTEGER)


    '--- IF THE PORT IS OPENNED,
    IF MSComm1.PortOpen THEN
        '--- SEND THE KEY STROKE TO THE PORT
        MSComm1.Output = CHR$(KeyAscii)
        '--- UNLESS ECHO IS ON, THERE IS NO NEED TO
        '    LET THE TEXT CONTROL DISPLAY THE KEY.
        IF NOT echo THEN KeyAscii = 0
    END IF

DEFINT A-Z

'--- MSComm EVENT CONSTANTS
GLOBAL CONST MSCOMM_EV_SEND = 1
GLOBAL CONST MSCOMM_EV_RECEIVE = 2
GLOBAL CONST MSCOMM_EV_CTS = 3
GLOBAL CONST MSCOMM_EV_DSR = 4
GLOBAL CONST MSCOMM_EV_CD = 5
GLOBAL CONST MSCOMM_EV_RING = 6
GLOBAL CONST MSCOMM_EV_EOF = 7

'--- MSComm ERROR CODE CONSTANTS
GLOBAL CONST MSCOMM_ER_BREAK = 1001
GLOBAL CONST MSCOMM_ER_CTSTO = 1002
GLOBAL CONST MSCOMM_ER_DSRTO = 1003
GLOBAL CONST MSCOMM_ER_FRAME = 1004
GLOBAL CONST MSCOMM_ER_OVERRUN = 1006
GLOBAL CONST MSCOMM_ER_CDTO = 1007
GLOBAL CONST MSCOMM_ER_RXOVER = 1008
GLOBAL CONST MSCOMM_ER_RXPARITY = 1009
GLOBAL CONST MSCOMM_ER_TXFULL = 1010

'--- COMMON DIALOG CONSTANTS
GLOBAL CONST CDERR_CANCEL = 32755
GLOBAL CONST CF_SCREENFONTS = &H1&

'--- GLOBAL VARIABLES
GLOBAL Echo                     'ECHO ON/OFF FLAG
GLOBAL CancelSend               'FLAG TO STOP SENDING
                                '   A TEXT FILE.
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก D

ASCII UNIT COMMAND

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Reference Tables

The following tables list the BASIC commands, statements, and functions alphabetically. A detailed explanation of each command, statement, and function may be found in *Section 4–2 Basic Language*.

The characters in the Command, Statement, and Function columns denote the following:

G: General statement
D: Device Control statement
A: Arithmetic Operation function
C: Character String function
S: Special function
✓: Command

Item	Description	Command	Statement	Function
ABS	Returns the absolute value of a number			A
ACOS	Returns the arc cosine of a number			A
ASC	Returns the value of the first character in a character string			C
ASIN	Returns the arc sine of a number			A
ATN	Returns the arc tangent of a number			A
AUTO	Automatically generates line numbers	✓		
CDBL	Rounds off a numeric value to make an integer			A
CHR$	Returns the character corresponding to the ASCII code given by the argument			C
CINT	Converts a numeric value into a double-precision real number			A
CLEAR	Initializes numeric and character variables		G	
CLOSE	Closes a port		D	
CLS	Clears the screen		D	
COM ON/OFF/ STOP	Enables, disables, or stops an interrupt from a communication port		G	
CONT	Resumes execution of a program that has been stopped	✓		
COS	Returns the cosine of a number			A
CSNG	Converts a numeric value into a single-precision real number			A
DATA	Defines numeric and character variables for subsequent READ statements		G	
DATE$	Sets or assigns the date			S
DAY	Sets or assigns the day (in numbers)			S
DEF FN	Defines and names a user-generated function		G	
LEN	Returns the total number of characters in a specified character string			C
LET	Assigns the result of the expression to the variable		G	
LINE INPUT	Reads one line of input from the keyboard and assigns it to a character string variable		G	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Item	Description	Command	Statement	Function
LIST/LLIST	Displays or prints a program	✔		
LOAD	Loads the program from the EEPROM or from a port	✔		
LOC	Returns the number of characters in the input queue waiting to be read			S
LOG	Returns the natural logarithm			A
MID$	Returns the specified number of characters starting from the specified character position		G	C
MON	Sets the terminal to monitor mode	✔		
MSET	Sets the address boundary for an assembly program	✔		
NEW	Clears the program and all currently defined variables	✔		
OCT$	Returns a string which represents the octal value of the decimal argument			C
ON COM GOSUB	Defines the branch destination of a subroutine invoked by an interrupt from a communication port		G	
ON ERROR GOTO	Causes branching to the specified line in the event of an error		G	
ON GOSUB GOTO	Causes branching to the specified line when "expression" is "true"		G	
ON KEY GOTO ON KEY GOSUB	Causes branching to the specified line when the specified key is input		G	
ON PC GOSUB	Defines an interrupt number and its associated subroutine branch line number		G	
OPEN	Opens a port		D	
PC GET	Reads data from the PC output area and assigns it to the specified variable		G	
PC ON/STOP	Enables or stops an interrupt invoked by the PC		G	
PC PUT	Writes the value of a numeric expression to the PC input data area		G	
PC READ (@)	Reads data from the specified PC memory area, converts it to the specified format, and assigns it to the specified variables		G	
PC WRITE (@)_	Converts data to the specified format and writes it to the specified PC memory area		G	
PEEK	Reads the contents of a specified memory address			S
PGEN	Sets the program memory area to be used	✔		
PINF	Displays the program area currently being used	✔		
PNAME	Names, or deletes the name, of the program selected	✔		
POKE	Writes data to a specified memory address		G	
PRINT/LPRINT	Displays or prints the value of an expression		G	
PRINT USING LPRINT USING	Displays or prints a character string in the specified format		G	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Item	Description	Command	Statement	Function
RANDOM	Reseeds the random number generator		G	
READ	Reads values from a data statement and assigns them to variables		G	
REM	Inserts a comment statement into the program		G	
RENUM	Reassigns line numbers in the program	✔		
RESTORE	Specifies which DATA statement will be used by the next READ statement		G	
RESUME	Specifies the line where execution will resume after error processing		G	
RIGHT$	Returns the number of characters in a string starting from the right			C
RND	Returns a random number between 0 and 1			A
RUN	Executes the program	✔		
SAVE	Saves the program to the EEPROM or to a device connected to a commuication port	✔		
SGN	Returns the sign of an argument			A
SIN	Returns the sine of a number			A
SPACE$	Returns an empty string of the specified number of characters			C
STOP	Stops program execution		G	
STR$	Converts a numeric value into a character string			C
STRING$	Returns a character string of the specified length			C
TAB	Outputs spaces up to the specified column position			C
TAN	Returns the tangent of a number			A
TIME$	Sets or gives the time			S
TRON/TROFF	Specifies or cancels a program trace	✔		
USR	Calls an assembly language function routine defined by a DEF USR statement			S
VAL	Converts a character string into a numeric value			C
VERIFY	Verifies the program and the EEPROM contents	✔		
VARPTR	Returns the memory address where the variable is stored			S
WAIT	Sets a delay before the next command is executed		G	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เอกสารอ้างอิง

(1) การเขียนโปรแกรมบนวินโดวส์ ด้วย VISAUL BASIC ภาคปฏิบัติ
โดย John Clark Craig

(2) วรวิทย์ ตันติโภคิน บริษัท เอซเอ็นกรุฟ 2537
นรดล ชาญธีระเดช

(3) Running Visual Basic for windows
ของ Ross Nelson
เรียบเรียงโดย ราบินเดอร์ ศรีกิจจาภรณ์
พิมพ์ที่ เม็ดทรายพริ้นติ้ง 2538

(4)) PC MAGAZINE VISUAL BASIC UTILITYS
'PAUL BONNER'
ZIFF DAUS PRESS EMERYU VILLE,CALIFORNIA 1993

(5) ADVANCED VISUAL BASIC
MARK S. BURGESS
สำนัก Addison - Wesley Publishing Compnay 1994

(6) ผศ. สุพรรณ กุลพาณิชย์ ' PROGRAMMABLE LOGIC CONTROL'
คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหาร ลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

(7) สุพจน์ ปุณณชัยยะ 'MODEM'
บริษัทด่านสุทธาการพิมพ์, 2521

(8) S.Andrew Silvert ' ASCII UNIT OPERATION MANUAL'
OMRON, 1990

(9) สมพงษ์ บุญธรรมจินดา 'คู่มือติดตั้งและใช้งาน โมเด็ม/แฟกซ์ โมเด็ม ด้วยตนเอง'
บริษัท เอซเอ็น กรุฟ จำกัด, 2537

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้