สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

ระบบควบคุมแผงแสดงผลแบบด็อทเมตริกซ์

DOT MATRIX DISPLAY CONTROL SYSTEM

โดย

นายเทพฤทธิ์ ทาอามาตย์

นายธนบัตร ธัญญวิกัย

เลขหมู่.....
เลขทะเบียน 42530
วัน, เดือน, ปี 24 พ.ค. 2545

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

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมระบบควบคุม

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2543

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์ปีการศึกษา 2543

ภาควิชาวิศวกรรมระบบควบคุม

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง ระบบควบคุมแผงแสดงผลแบบด็อทเมตริกซ์ (Dot Matrix Display Control System)

จัดทำโดย

1. นายเทพฤทธิ์ ทาอามาตย์
2. นายธนบัตร ธัญญวิกัย

.....อาจารย์ที่ปรึกษา

(อาจารย์เกียรติวรรณ ทรงสัตย์)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระบบควบคุมแผงแสดงผลแบบด็อทเมตริกซ์

DOT MATRIX DISPLAY CONTROL SYSTEM

โดย

นายเทพฤทธิ์ ทาอามาตย์

นายธนบัตร ธัญญวิกัย

อาจารย์ที่ปรึกษา

อาจารย์เกียรติวรรณ ทรงสัตย์

บทคัดย่อ

ในวิทยานิพนธ์ฉบับนี้ เรียบเรียงขึ้นจากผลงานที่ได้พัฒนาระบบควบคุมแผงแสดงผลแบบด็อทเมตริกซ์โดยอาศัยการควบคุมของไมโครคอนโทรลเลอร์ เพื่อให้สามารถแสดงผลได้ทั้งข้อมูลที่เป็นรูปภาพและตัวอักษร โดยมีรูปแบบการแสดงผลทั้งภาพนิ่ง ข้อความต่อเนื่อง และภาพเคลื่อนไหว ผู้ใช้สามารถออกแบบรูปภาพหรือข้อความ รวมทั้งกำหนดรูปแบบการแสดงผลผ่านทางหน้าจอใช้งานของคอมพิวเตอร์ส่วนบุคคล ซึ่งข้อมูลที่สร้างนี้สามารถบันทึกไว้เพื่อใช้ในการแสดงผลครั้งต่อไปได้ และหลังจากสั่งงานเรียบร้อยแล้ว ผู้ใช้สามารถใช้คอมพิวเตอร์ส่วนบุคคลในการทำงานอื่นได้ โดยที่บอร์ดแสดงข้อมูลยังทำงานตามคำสั่งอยู่

ABSTRACT

This thesis is presented about the development of Dot matrix display controlled by microprocessor ,for propose to present both of graphic and character display in variable displays including fixed display, long-statement run and motion pictures. User can create picture or message ,select display mode by use interface on personal computer and the data can be saved for display in next times. After the commanding personal computer can be used for any task as display board still work under the command of personal computer.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

ปริญญานิพนธ์ฉบับนี้สำเร็จลุล่วงได้ด้วยดี เนื่องจากได้รับคำแนะนำ ความรู้ และความช่วยเหลือจากอาจารย์เกียรติวรรณ ทรงสัตย์ อาจารย์ที่ปรึกษาโครงงาน อีกทั้งยังได้คำแนะนำเพิ่มเติมจากอาจารย์ท่านอื่น ๆในภาควิชาระบบควบคุม จึงขอขอบพระคุณไว้ ณ. ที่นี้ด้วย

ขอขอบพระคุณ คุณพ่อ คุณแม่และอาจารย์ทุกท่านที่ได้ให้การอบรมสั่งสอนตลอดมา

ขอขอบคุณเพื่อน ๆ ทุกคน สำหรับความช่วยเหลือ กำลังใจ และความเป็นเพื่อนที่ดี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

	หน้า
บทคัดย่อ	i
กิตติกรรมประกาศ	ii
สารบัญ	iii
สารบัญตาราง	v
สารบัญรูปภาพ	vi
บทที่ 1 บทนำ	1
บทที่ 2 บอร์ดแสดงข้อมูล	3
2.1 หลักการออกแบบ	3
2.2 คุณลักษณะของส่วนประกอบต่าง ๆ ของบอร์ดแสดงข้อมูล	3
2.2.1 ส่วนแสดงผล	3
2.2.2 ส่วนตัวขับ	5
2.2.3 ส่วนควบคุมและพักข้อมูล	12
บทที่ 3 ไมโครคอนโทรลเลอร์	18
3.1 ส่วนประกอบ	18
3.1.1 หน่วยประมวลผลกลาง	18
3.1.2 หน่วยความจำ	20
3.1.3 พอร์ตอนุกรม	22
3.2 การทำงานของไมโครคอนโทรลเลอร์	24
3.2.1 การรับข้อมูลจากคอมพิวเตอร์ส่วนบุคคล	24
3.2.2 การควบคุมการแสดงผลของบอร์ดแสดงข้อมูล	24
3.2.2.1 การแสดงข้อมูลแบบภาพนิ่ง	24
3.2.2.2 การแสดงข้อมูลแบบภาพเคลื่อนไหว	25
บทที่ 4 คอมพิวเตอร์ส่วนบุคคล	32
4.1 หลักการออกแบบโปรแกรม	33
4.1.1 หลักการออกแบบโปรแกรมส่วนติดต่อกับผู้ใช้	33
4.1.2 หลักการออกแบบโปรแกรม สร้าง/แก้ไขตัวอักษร	35
4.2 รูปแบบของข้อมูลที่ได้จากโปรแกรมส่วนติดต่อกับผู้ใช้	36

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ
(ต่อ)

	หน้า
4.3 การส่งข้อมูลไปยังไมโครคอนโทรลเลอร์	37
4.4 ข้อมูลของคอนโทรลที่สำคัญที่ใช้ในการออกแบบโปรแกรม	37
บทที่ 5 การทดลองและผลการทดลอง	42
5.1 ขั้นตอนในการศึกษาและทดลอง	42
5.2 ผลการทดลอง	42
บทที่ 6 สรุปผลและวิจารณ์	44
หนังสืออ้างอิง	45
ภาคผนวก ก รีจิสเตอร์ที่สำคัญของไมโครคอนโทรลเลอร์ Z80180	
ภาคผนวก ข การใช้งานโปรแกรมควบคุมบอร์ดแสดงข้อมูล	
ภาคผนวก ค โปรแกรมในไมโครคอนโทรลเลอร์สำหรับควบคุมบอร์ดแสดงข้อมูล	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

หน้า

ตารางที่ ผ-1 แสดงรูปแบบข้อมูลจากการเซ็ต MOD0,1,2 50

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญภาพ

รูปที่	หน้า
1.1 แสดงส่วนประกอบที่สำคัญของโครงงาน	2
2.1 แสดงบล็อคไดอะแกรมของโครงสร้างทางฮาร์ดแวร์ และการส่งผ่านข้อมูลระหว่างส่วนประกอบต่าง ๆ ของโครงงาน	4
2.2 แสดงการแสดงการเชื่อมต่อของบอร์ดแสดงผลทางด้านแถว	7
2.3 แสดงการแสดงการเชื่อมต่อของบอร์ดแสดงผลทางด้านคอลัมน์	8
2.4 แสดงวงจรทางไฟฟ้าของตัวขับทางด้านแถวและคอลัมน์ต่อ 1 จุด	9
2.5 แสดงวงจรและการเชื่อมต่อของการทำงานทางด้านคอลัมน์	11
2.6 แสดงวงจร Decode Port Latch ทั้ง 24 ตัว โดยใช้ 74LS139 และ 74LS138	15
2.7 แสดงวงจรและการเชื่อมต่อพอร์ตพักข้อมูลของบอร์ดย่อย	16
2.8 แสดงวงจรทั้งหมดของส่วนวงจรควบคุมและพักข้อมูล	17
3.1 บล็อกไดอะแกรมโครงสร้างภายใน Z80180	19
3.2 แสดงตำแหน่งหน่วยความจำเรียกใช้ในทางโปรแกรม	21
3.3 แสดงตัวอย่างการอ้างอิงหน่วยความจำ	22
3.4 แสดงบล็อคไดอะแกรมของส่วนประกอบต่างๆของเอเอสซีไอ	23
3.5 แสดงผังงานของโปรแกรมบนไมโครคอนโทรลเลอร์ในการรับข้อมูลจากคอมพิวเตอร์	27
3.6 แสดงผังงานของโปรแกรมบนไมโครคอนโทรลเลอร์ในการส่งข้อมูลไปบอร์ดแสดงผล	28
3.7 แสดงผังงานของโปรแกรมบนไมโครคอนโทรลเลอร์ในการแสดงภาพนิ่งและภาพเลื่อนขนาด 1 เฟรม	29
3.8 แสดงผังงานของโปรแกรมบนไมโครคอนโทรลเลอร์ในการแสดงภาพเคลื่อนไหว	30
3.9 แสดงผังงานของโปรแกรมบนไมโครคอนโทรลเลอร์ในการแสดงข้อความต่อเนื่อง	31
4.1 บอร์ดแสดงผลที่จำลองบนหน้าจอคอมพิวเตอร์	34
4.2 ตัวเลือกสีของหลอดแอลอีดีจำลอง	34
4.3 โปรแกรม สร้าง/แก้ไขตัวอักษร	35

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญภาพ

(ต่อ)

รูปที่	หน้า
4.4 รูปแบบการส่งข้อมูลจากคอมพิวเตอร์ให้กับไมโครคอนโทรลเลอร์	36
4.5 ฟอร์ม (Form)	38
4.6 คอมมานด์บัตทอน (Command Button)	39
4.7 คอนโทรลเท็กบอกซ์ (TextBox)	39
4.8 คอนโทรลออปชั่นบัตทอนและคอนโทรลเฟรม	40
4.9 คอนโทรล MSComm	41
5.1 แสดงการทดลองควบคุมการแสดงผลของบอร์ดแสดงข้อมูล	43
5.2 แสดงไมโครคอนโทรลเลอร์ที่ใช้ควบคุมการแสดงผลของบอร์ดแสดงข้อมูล	43
ผ-1 แสดงบิตต่างๆของรีจิสเตอร์สถานะของเอเอสซีไอ	47
ผ-2 แสดงบิตต่างๆของรีจิสเตอร์ควบคุมของเอเอสซีไอ เอ	49
ผ-3 แสดงบิตต่างๆของรีจิสเตอร์ควบคุมของเอเอสซีอี บี	50
ผ-4 แสดงหน้าจอของโปรแกรมสร้างภาพและส่งข้อมูลออกแสดงผล	54
ผ-5 แสดงหน้าจอของโปรแกรมสร้างและแก้ไขตัวอักษร	55

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

<u>บทที่ 1</u>

บทนำ

บอร์ดแสดงข้อมูลแบบจุด (Dot Matrix Display) เป็นสื่อแสดงข้อมูลข่าวสารที่นิยมใช้กันอย่างแพร่หลายมากในปัจจุบันไม่ว่าจะเพื่อการโฆษณา การกีฬา เนื่องจากความสามารถในการดึงดูดความสนใจของบุคคลต่างๆได้เป็นอย่างดี โดยได้มีการพัฒนาความสามารถในการแสดงข้อมูลของบอร์ดแสดงข้อมูลให้มีความหลากหลายและใช้งานได้สะดวกยิ่งขึ้นเรื่อยมา

ในโครงงานนี้ได้ใช้ไมโครคอนโทรลเลอร์ควบคุมการแสดงข้อมูลของบอร์ดแสดงข้อมูลแบบจุดซึ่งใช้หลอดแอลอีดี (LED , Light Emitting Diode) ในการแสดงผลแต่ละจุด หลอดแอลอีดีแต่ละหลอดสามารถเปล่งแสงผลได้ 3 สี คือ แดง เขียวหรือ ส้ม ซึ่งถ้ารวมกับช่วงที่หลอดดับแล้วก็จะสามารถแสดงได้ทั้งหมด 4 สีโดยเพิ่มสีดำขึ้นมาอีก 1 สี สำหรับการควบคุมการแสดงผลของบอร์ดแสดงผลทำได้โดยใช้ไมโครคอนโทรลเลอร์ควบคุมการส่งข้อมูลและการสแกน(Scan) บอร์ดแสดงข้อมูล โดยข้อมูลเหล่านี้ไมโครคอนโทรลเลอร์ได้รับมาจากคอมพิวเตอร์ส่วนบุคคล(Personal Computer) ซึ่งทางคณะผู้จัดทำได้ออกแบบหน้าจอติดต่อกับผู้ใช้ (Interface) ให้สามารถใช้งานได้สะดวกและมีความหลากหลายในการใช้งานมากขึ้น การสั่งงานทั้งหมดจะทำได้บนคอมพิวเตอร์ส่วนบุคคล โดยผู้ใช้งานเพียงแค่สร้างข้อมูลและเลือกโหมดการแสดงผลเท่านั้น ส่วนงานอื่นๆ โปรแกรมจะทำหน้าจัดการให้จนกระทั่งเสร็จสิ้นการแสดงข้อมูล ซึ่งหลังจากส่งข้อมูลไปแสดงผลแล้วผู้ใช้สามารถใช้คอมพิวเตอร์ส่วนบุคคลในการทำงานอื่นต่อไปได้

ส่วนประกอบที่สำคัญของโครงงานนี้มีทั้งหมด 3 ส่วนคือ

1. <u>บอร์ดแสดงข้อมูล</u> (Dot Matrix Display) ประกอบด้วย

- ส่วนแสดงผล (Display Board)
  ประกอบด้วยแผงแอลอีดีจำนวน 1920 หลอดเรียงกัน 24 แถว 80 หลัก
- ส่วนตัวขับ (Driver Circuit Board)

ประกอบด้วย 2 ส่วน คือ วงจรตัวขับทางแถว (Row Driver Circuit) และวงจรตัวขับทางคอลัมน์ (Column Driver Circuit) ซึ่งแยกกันคนละวงจร โดยจะทำหน้าที่ขับกระแสไปให้แอลอีดีตามตำแหน่งแถว และคอลัมน์ที่ต้องการให้หลอดเปล่งแสง

- ส่วนควบคุมและพักข้อมูล (Control / Latch Data Circuit Board)
  ทำหน้าที่เก็บพักข้อมูลที่ส่งมาจากไมโครคอนโทรลเลอร์ เพื่อให้บอร์ดแสดงผลสามารถแสดงผลออกมาพร้อมกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. ไมโครคอนโทรลเลอร์ (Microcontroller)

ไมโครคอนโทรลเลอร์ ที่ใช้ในการควบคุมเป็นรุ่น Z80180 เป็นตัวควบคุมการทำงานของอุปกรณ์ต่างๆบนคอนโทรลบอร์ด ซึ่งอุปกรณ์ต่างๆเหล่านี้เป็นตัวช่วยให้การติดต่อระหว่างคอมพิวเตอร์ส่วนบุคคลกับไมโครคอนโทรลเลอร์ และระหว่างไมโครคอนโทรลเลอร์กับบอร์ดแสดงข้อมูลทำได้โดยสะดวก นอกจากนี้ข้อมูลที่จะใช้ในการแสดงผลยังถูกเก็บอยู่ในหน่วยความจำชั่วคราว(RAM) ของคอนโทรลบอร์ดอีกด้วย

3. คอมพิวเตอร์ส่วนบุคคล (Personal Computer)

คอมพิวเตอร์ส่วนบุคคลมีหน้าที่ในการติดต่อกับผู้ใช้งานซึ่งคณะผู้จัดทำได้ใช้โปรแกรมวิชวลเบสิก(Visual Basic) ในการออกแบบส่วนที่ติดต่อกับผู้ใช้งานโดยผู้ใช้สามารถใช้เมาส์(Mouse) หรือแป้นพิมพ์(Keyboard) ในการสร้างรูปภาพหรือข้อความและส่งข้อมูลไปแสดงผลยังบอร์ดแสดงข้อมูล นอกจากนี้คอมพิวเตอร์ส่วนบุคคลยังทำหน้าที่แปลงข้อมูลที่ผู้ใช้สร้างขึ้นให้อยู่ในรูปแบบที่กำหนดพร้อมทั้งใส่ข้อมูลที่จำเป็นในการรับข้อมูลแล้วส่งไปยังไมโครคอนโทรลเลอร์เพื่อส่งไปแสดงยังบอร์ดแสดงข้อมูลตามลำดับ

รูปที่1.1 แสดงส่วนประกอบที่สำคัญของโครงงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2

บอร์ดแสดงข้อมูล

(Dot Matrix Display Board)

2.1 หลักการออกแบบ

บอร์ดแสดงข้อมูลประกอบด้วยส่วนของวงจรที่ใช้ขับกระแสให้แก่แอลอีดี ซึ่งอยู่รวมกันเป็นแผงในส่วนแสดงผล ส่วนของวงจรควบคุมและส่วนพักข้อมูล ทั้ง 3 ส่วนแยกออกจากกันโดยอยู่บนบอร์ดเดียวกัน และมีการเชื่อมต่อถึงกันเพื่อส่งข้อมูล โดยใช้สายแพส่งข้อมูลจากไมโครคอนโทรลเลอร์มาที่ส่วนควบคุมและพักข้อมูล ซึ่งเมื่อข้อมูลถูกพักและพร้อมที่จะส่งออกไปแสดงผลแล้ว ก็จะถูกส่งออกไปยังส่วนตัวขับ ซึ่งเชื่อมต่ออยู่กับส่วนแสดงผลเพื่อแสดงผลข้อมูล เพื่อแสดงข้อมูลตามที่ต้องการ ตัวบอร์ดแยกเป็น 2 ชั้น ชั้นบนเป็นที่ติดตั้งส่วนแสดงผลโดยด้านหลังของชั้นนี้ได้ติดตั้งวงจรตัวขับทางแถวไว้และใช้สายแพในการส่งข้อมูลระหว่างส่วนตัวขับไปยังส่วนแสดงผลด้านหน้า ส่วนชั้นที่ 2 หรือชั้นหลังเป็นที่ติดตั้งของส่วนควบคุมและพักข้อมูล และวงจรตัวขับทางคอลัมน์โดยใช้สายแพเป็นสายส่งข้อมูลระหว่างแต่ละบอร์ดเช่นเดียวกัน

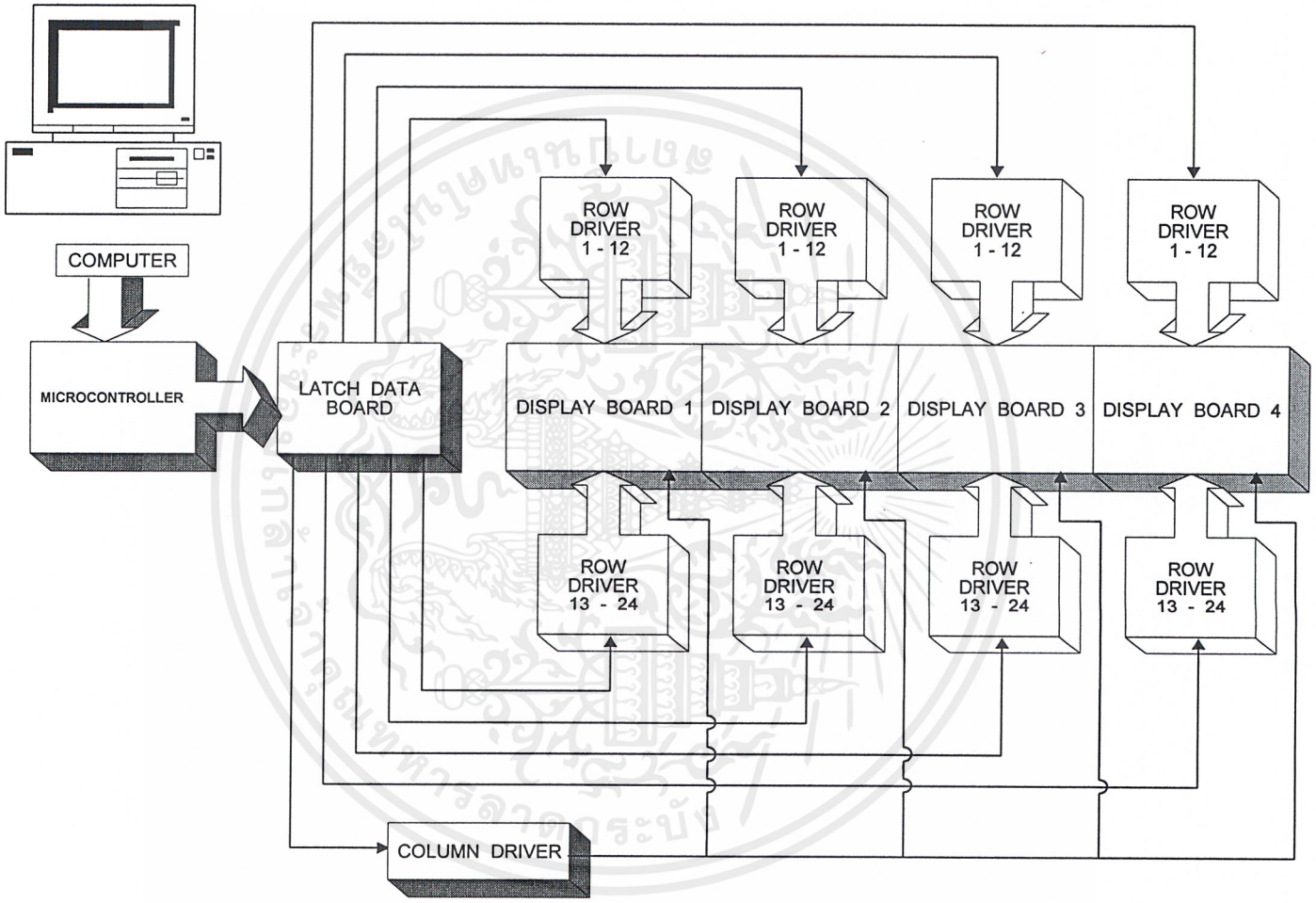
เนื่องจากการจ่ายไฟป้อนเข้าแผงแอลอีดีโดยตรงทุกดวงพร้อมกัน จะทำให้อายุการใช้งานแอลอีดีสั้นลงเพราะจะต้องทำงานตลอดเวลา ดังนั้นการออกแบบจึงต้องออกแบบการทำงานเป็นแบบมัลติเพลกซ์ (Multiplex) โดยให้มีการทำงานทีละคอลัมน์สลับเรียงกันไปเรื่อย ๆ ด้วยความเร็วสูงเพื่อให้เห็นว่าทุกคอลัมน์ทำงานพร้อมกันหมด โดยใช้การสแกน (Scan) ทางด้านคอลัมน์ด้วยความเร็วสูงโดยจะไม่มีการสแกนทางด้านแถวเลย การจ่ายไฟเลี้ยงขนาด 12 โวลท์ ให้กับแต่ละส่วนของบอร์ดแสดงผลต้องทำแบบขนานเพื่อให้แต่ละส่วนได้รับไฟเลี้ยงเท่ากัน เนื่องจากถ้าเราจ่ายไฟเลี้ยงให้เพียงส่วนเดียวแล้วต่ออนุกรมแต่ละส่วน จะทำให้แต่ละส่วนได้รับไฟเลี้ยงไม่เท่ากันเนื่องจากมีศักดาไฟฟ้าตกคร่อมในแต่ละส่วน

2.2 คุณลักษณะของส่วนประกอบต่าง ๆ ของบอร์ดแสดงข้อมูล (Dot Matrix Display Board)

2.2.1 ส่วนแสดงผล (Display Board)

ส่วนแสดงผลจะประกอบด้วยแอลอีดีเป็นหัวใจสำคัญของการแสดงข้อมูล ดังนั้นจึงต้องทำความเข้าใจเกี่ยวกับคุณสมบัติและการทำงานของแอลอีดีให้เข้าใจเพื่อจะนำไปใช้ในการกำหนดสีของแอลอีดี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.1 แสดงบล็อคไดอะแกรมของโครงสร้างทางฮาร์ดแวร์ และการส่งผ่านข้อมูลระหว่างส่วนต่าง ๆ ของโครงงาน

<u>คุณสมบัติของแอลอีดี หรือไดโอดปล่อยแสง</u> (LED หรือ Light Emitting Diode)

แอลอีดีเป็นไดโอดชนิดหนึ่งซึ่งสามารถปล่อยแสงได้ โดยโครงสร้างของแอลอีดีจะมีลักษณะเป็นสารกึ่งตัวนำชนิดพี (P-Type) และชนิดเอ็น (N-Type) ต่อกันเป็นพีเอ็นจังชั่น (P-N Junction) แอลอีดีจะปล่อยแสงออกมาได้ก็ต่อเมื่อมีการจ่ายกระแสฟคร์เวิร์ดไบอัส (Forward Bias) ให้กับมัน กระแสฟอร์เวิร์ดไบอัสนี้จะไปกระตุ้นอิเล็กตรอน (Electron)และโฮล (Hole) ข้ามพีเอ็นจังชั่นเพื่อมารวมกันในการรวมกันนี้จะมีการปลดปล่อยพลังงานออกมาในรูปของโฟตอน (Photon)ซึ่งเป็นอนุภาคแสง สำหรับสารกึ่งตัวนำที่นิยมนำมาสร้างแอลอีดีจะใช้สารแกลเลี่ยมอาร์เซไนด์ฟอสไฟด์ (Gallium Arsenidephosphide : GaAsP) ซึ่งสารชนิดนี้จะใช้กระแสไฟฟ้าไม่มากในการไบอัสให้ปลดปล่อยโฟตอนออกมา เนื่องจากการออกแบบได้แบ่งส่วนแสดงผลออกเป็น 4 ส่วน โดยแต่ละส่วนมีขนาดจำนวนจุดทางด้านแถวและด้านคอลัมน์เป็น 24 × 20 จุด ตามลำดับดังนั้นจะทำการอธิบายการออกแบบและการทำงานเพียงส่วนเดียวจาก 4 ส่วนเท่านั้น โดยวงจรในส่วนบอร์ดนั้นจะทำการเชื่อมขาของด็อทเมตริกซ์ (Dot Matrix) ที่ใช้ควบคุมด้านแถวเข้าด้วยกันในลักษณะขนานโดยขาที่ใช้ควบคุมแถวเดียวกันจะถูกเชื่อมต่อถึงกันหมด ดังนั้นเมื่อจ่ายไฟเข้าที่ตัวใดตัวหนึ่งของแถว ก็จะทำให้หลอดแอลอีดีบนแถวนั้นติดทุกดวง และในคอลัมน์เดียวกันของแต่ละแถวจะถูกเชื่อมถึงกันทั้งหมดเพื่อลงกราวน์ที่จุดเดียวกัน โดยการต่อเป็นวงจรเป็นดังรูปที่ 2.2 และ 2.3

ดังนั้นเมื่อต้องการให้แอลอีดีที่แถวใดและคอลัมน์ใด ก็เพียงทำการเลือกแถวและคอลัมน์นั้นแล้วส่งข้อมูลสีของแอลอีดีที่ต้องการ ข้อมูลก็จะพักที่แถวและคอลัมน์นั้น

2.2.2 <u>ส่วนตัวขับ</u> (Driver Circuit Board)

เนื่องจากจะต้องจ่ายไฟให้แก่แผงแอลอีดีในแต่ละหลอดเพื่อให้หลอดสว่างตามต้องการเพื่อแสดงข้อมูลเป็นตัวอักษร และรูปกราฟฟิกต่าง ๆ ดังนั้นเราจำเป็นต้องมีส่วนตัวขับที่ทำหน้าที่เป็นสวิตซ์ ปิด/เปิด (Off/On) โดยใช้ทรานซิสเตอร์ทำหน้าที่เป็นตัวควบคุมการจ่ายกระแสให้แก่แผงแอลอีดี

ส่วนตัวขับแบ่งออกเป็น 2 วงจร ได้แก่

- วงจรตัวขับทางแถว (Row Driver Circuit)
- วงจรตัวขับทางคอลัมน์ (Column Driver Circuit)

<u>วงจรตัวขับทางแถว</u> (Row Driver Circuit)

ทรานซิสเตอร์ที่ใช้ได้แก่ เบอร์ 4037 PNP และเบอร์ 9013 NPN ใช้ความต้านทานขนาด 1 กิโลโอห์ม , 10 กิโลโอห์ม และ 50 กิโลโอห์ม โดยจะแบ่งวงจรออกเป็น 2 ส่วน คือ

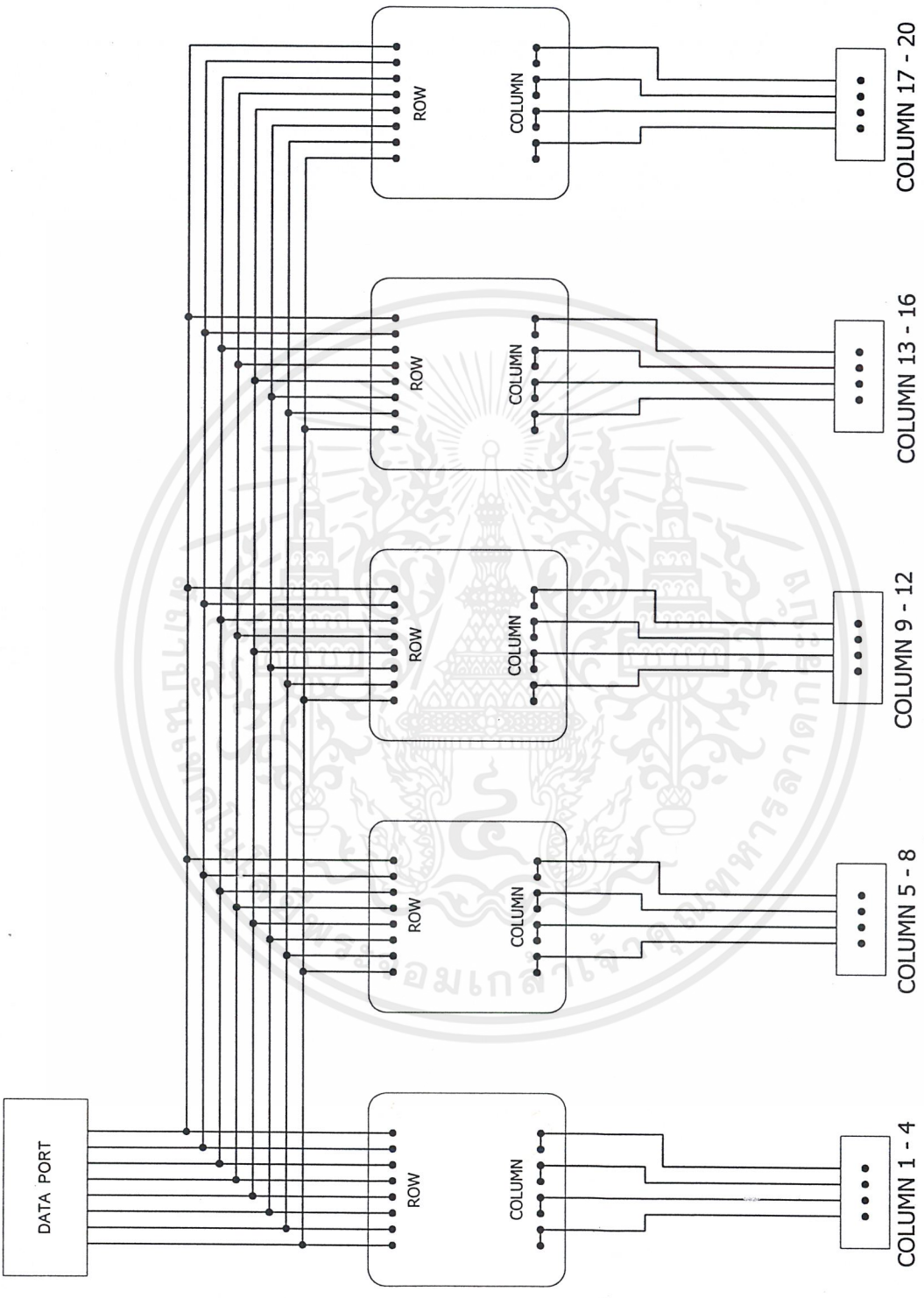
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ส่วนตัวขับกระแสให้แก่แอลอีดีสีแดง และส่วนขับกระแสให้แก่แอลอีดีสีเขียว โดยแต่ละส่วนมีการต่อวงจรแบบวงจรเสมือนสมมาตร (Quasi-Complementary Symmetry)ซึ่งเป็นวงจรดาร์ลิงตัน (Darlington) ชนิดหนึ่งดังรูปที่ 2.4

โดยใช้ความต้านทานขนาด 50 โอห์ม (R7 , R8) เป็นตัวจำกัดกระแสที่จะไหลเข้าสู่แอลอีดีแต่ละหลอด และใช้ความต้านทานขนาด 1 กิโลโอห์ม (R1 , R2) เป็นตัวจำกัดกระแสเบสที่จะไหลเข้าไปไบอัสทรานซิสเตอร์เบอร์ 9013 นอกจากนี้ยังใช้ความต้านทานขนาด 10 กิโลโอห์ม (R3 , R4) เป็นตัวจำกัดกระแสที่จะไหลเข้าสู่ทรานซิสเตอร์ เพราะถ้าไม่มีจะทำให้ทรานซิสเตอร์เบอร์ 9013 มีกระแสไหลมากจนเกินไปมีผลทำให้ทรานซิสเตอร์ร้อนจนเกินไปซึ่งจะทำให้ทรานซิสเตอร์เสียหาย

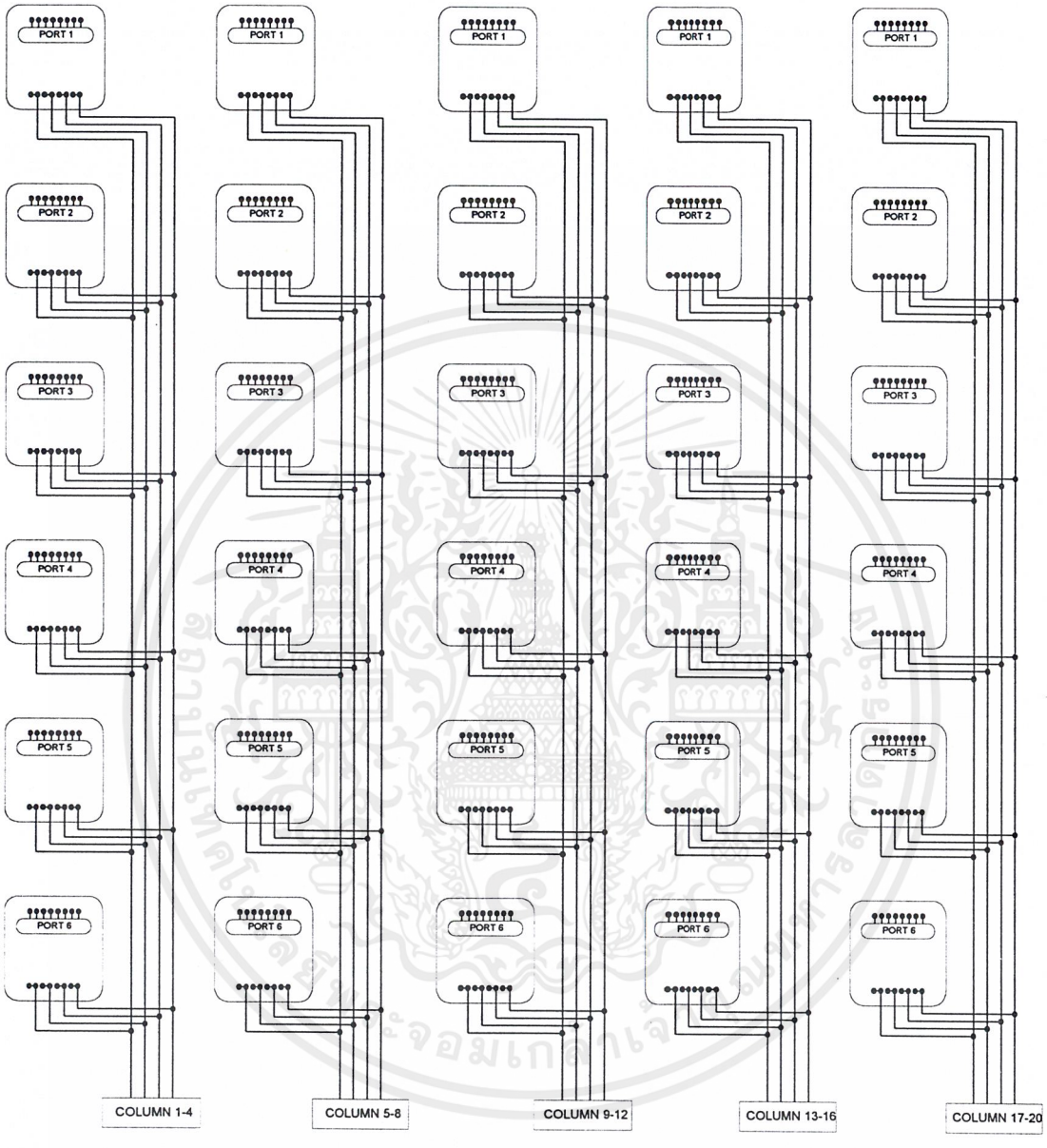
เมื่อเราต้องการให้หลอดแอลอีดีสีใด และตำแหน่งใดสว่างก็ต้องขับกระแสไปยังหลอดแอลอีดีนั้น ๆ โดยการส่งลอจิก (Logic) 1 ไปเปิดทาง (Enable) ที่ขาเบสของทรานซิสเตอร์ N2 และ/หรือ N3 โดยที่ลอจิก 1 ไปเปิดทางที่ขาเบสของทรานซิสเตอร์ N2 จะทำให้กระแสถูกขับไปยังแอลอีดี L2 ซึ่งจะให้แสงสีเขียว (Green) ออกมา ในขณะที่ถ้าส่งลอจิก 1 ไปเปิดทางที่ขาเบสของ N3 จะทำให้กระแสถูกขับไปยังแอลอีดี L1 ซึ่งจะให้แสงสีแดง (Red) ออกมา และถ้าส่งลอจิก 1 ไปเปิดทางที่ขาเบสของทรานซิสเตอร์ทั้งสองตัวพร้อมกัน ก็จะได้แสงสีส้ม (Orange) สว่างออกมา คือเป็นสีผสมระหว่างสีแดง และสีเขียว เราใช้วงจรขับทางแถวนี้เพื่อควบคุมกระแสที่จะจ่ายเข้าที่ขาควบคุมด้านแถว(แถวบน) คือตั้งแต่ RR1 -RR4 และ RG1 – RG4 ถ้าในส่วนของวงจรนี้ประกอบไปด้วยทรานซิสเตอร์เพียงสองตัวเท่านั้น คือใช้ขับกระแส คือใช้ขับกระแสให้แก่หลอดแอลอีดีแต่ละตัว จะทำให้มีประสิทธิภาพไม่ดีเพราะไม่สามารถคัทออฟ (Cut Off) ทรานซิสเตอร์ได้จึงต้องมีการพัฒนาวงจรขึ้นใหม่ดังกล่าว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



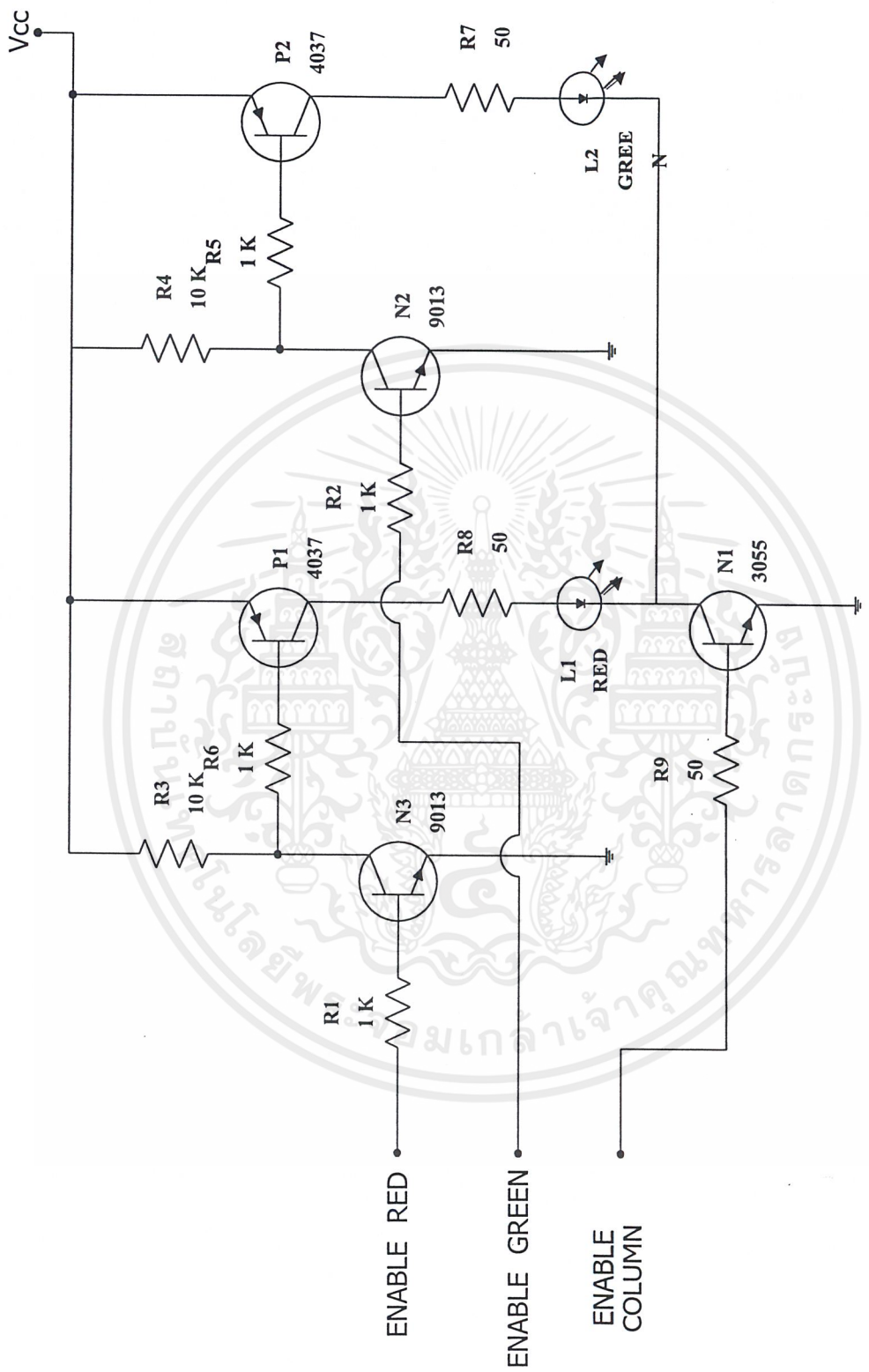
รูปที่ 2.2 แสดงการเชื่อมต่อของบอร์ดแสดงผลทางด้านแถว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.3 แสดงการเชื่อมต่อของบอร์ดแสดงผลทางด้านคอลัมน์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.4 แสดงวงจรทางไฟฟ้าของตัวขับทางด้านแถวและคอลัมน์ต่อ 1 จุด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

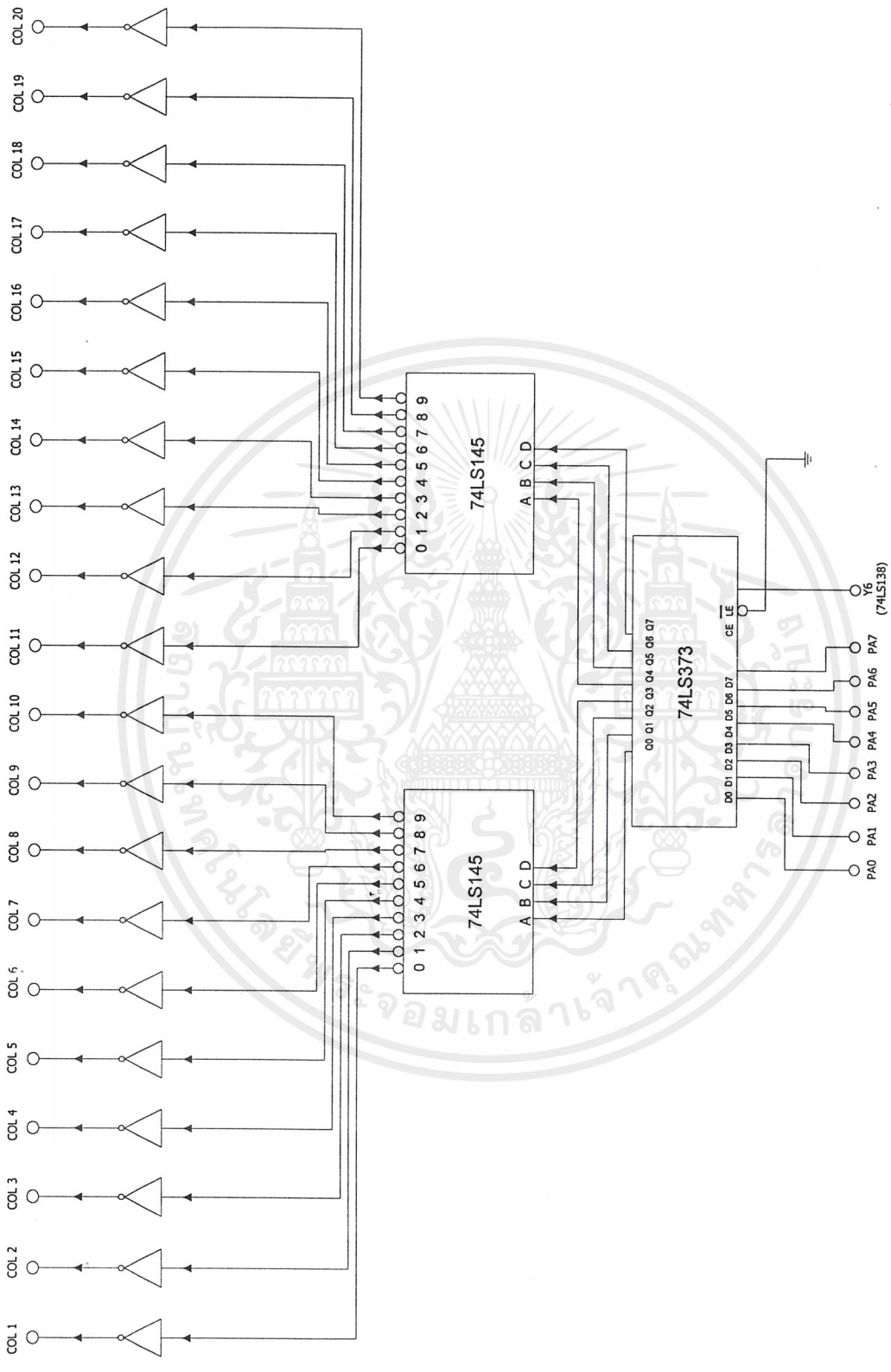
วงจรตัวขับทางคอลัมน์ (Column Driver Circuit)

ทรานซิสเตอร์ที่ใช้ ได้แก่ เบอร์ 3055 NPN และใช้ความต้านทานขนาด 50 โอห์ม เพื่อควบคุมกระแสที่ขาเบสของทรานซิสเตอร์ โดยทรานซิสเตอร์นี้จะควบคุมกระแสที่ไหลลงกราวน์ โดยจะต่อออกจากขาควบคุมด้านคอลัมน์ (แถวล่าง) คือต่อกับขา CR1 – CR4 และ CG1 – CG4 ซึ่งโดยปกติจะเชื่อมต่อขา CR และ CG ที่ควบคุมแต่ละคอลัมน์เข้าด้วยกันอยู่แล้ว การควบคุมจะสามารถทำได้โดยการส่งลอจิก 1 ไปเปิดทางที่ขาเบสของทรานซิสเตอร์ เนื่องจากทรานซิสเตอร์เบอร์ 4037 เป็นทรานซิสเตอร์ชนิด NPN

วงจรตัวขับทางคอลัมน์นี้จะอยู่รวมบนบอร์ดเพียงบอร์ดเดียว ใช้ควบคุมคอลัมน์ทั้งหมดทั้ง 80 คอลัมน์ โดยใช้ทรานซิสเตอร์ เบอร์ 3055 ทั้งหมดเพียง 20 ตัวเท่านั้น โดยใช้การต่อแบบขนานทั้ง 4 ส่วนของบอร์ด ซึ่งคอลัมน์เดียวกันของแต่ละชุดจะสแกนไปพร้อม ๆ กัน ดังนั้นเราจึงใช้วงจรตัวขับทางคอลัมน์เพียงชุดเดียวก็พอ

รายละเอียดการต่อวงจรในส่วนของวงจรตัวขับทางคอลัมน์แสดงไว้ดังรูปที่ 2.5

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.5 แสดงวงจรและการเชื่อมต่อของการทำงานทางด้านคอลัมน์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.3 ส่วนควบคุมและพักข้อมูล (Control / Latch Data CircuitBoard)

เนื่องจากส่วนไมโครโปรเซสเซอร์ในไมโครคอมพิวเตอร์ส่งข้อมูลแบบอนุกรมมาเก็บไว้ที่หน่วยความจำของไมโครคอนโทรลเลอร์ เมื่อทำการเรียก (Run) โปรแกรมที่เก็บไว้มาใช้ ไมโครคอนโทรลเลอร์ก็จะส่งข้อมูลออกมาเข้าส่วนควบคุมและพักข้อมูล ซึ่งจะทำหน้าที่รับข้อมูลที่ส่งมาแล้วทำการพัก (Latch)ข้อมูลเอาไว้ให้ข้อมูลที่ถูกส่งเข้ามาครบทุกแถวก่อน เนื่องจากข้อมูลที่ไมโครคอนโทรลเลอร์ส่งมานั้นส่งมาได้เพียงครั้งละ 8 บิตเท่านั้น เมื่อข้อมูลถูกส่งออกมาครบถ้วนแล้ว ตัวไมโครคอนโทรลเลอร์ก็จะส่งสัญญาณออกมาควบคุมให้ไอซีพักข้อมูล (IC Latch) ทุกตัวส่งข้อมูลออกมาขาเอาท์พุท (Output) พร้อมกันเพื่อส่งเข้าไปยังส่วนแสดงผลเพื่อทำให้หลอดสว่างตามข้อมูลที่ส่งมา แต่หลอดจะยังไม่สว่างจนกว่าไมโครคอนโทรลเลอร์จะส่งสัญญาณมาควบคุมให้วงจรขับทางคอลัมน์ทำงานด้วย โดยส่งสัญญาณมาควบคุมเปิดเกทที่คอลัมน์ใดก็จะทำให้แอลอีดีในคอลัมน์นั้นๆสว่างตามข้อมูลที่ส่งมาในแต่ละแถว

องค์ประกอบที่สำคัญของวงจรควบคุมและพักข้อมูล ได้แก่

1.ไอซีพักข้อมูล (IC Latch)ใช้ไอซีเบอร์ 74LS373 (เป็นแบบ Octal D-Type Transparent Latches and Edge-Triggered Flip-Flop)

2. ดีโคดเดอร์ควบคุมการพักข้อมูล (Latch Decoder) ใช้ไอซีเบอร์ 74LS138 (เป็นแบบ 3 Line-8 Line Decoder) ซึ่งจะมีสัญญาณเข้ามาควบคุม 3 ขา แล้วมีสัญญาณออกทั้งหมด 8 ขา โดยเราจะนำไปควบคุมการทำงานของไอซีพักข้อมูลทางด้านแถวทั้ง 6 ตัว แล้วที่เหลืออีก 2 ขา จะนำไปควบคุมการทำงานของไอซีพักข้อมูลทางด้านคอลัมน์

3. ดีโคดเดอร์ควบคุมส่วนตัวขับทางคอลัมน์ (Column Decoder) ใช้ไอซีเบอร์ 74LS145 (เป็นแบบ BCD-to-Decimal/Driver with Open-Collector Output) ซึ่งจะมีสัญญาณเข้าควบคุม 4 ขา และมีสัญญาณออกทั้งหมด 10 ขา เราจะนำสัญญาณนี้ออกทั้งหมด 10 ขาเพื่อไปควบคุมการทำงานของวงจรตัวขับทางด้านคอลัมน์โดยใช้ไอซีชนิดนี้ 2 ตัวเพื่อควบคุมทั้ง 20 คอลัมน์

4.อินเวอร์เตอร์ (Inverter) ใช้ไอซีเบอร์ 74LS04 (เป็นแบบ Hex Inverter)

5.ดีโคดเดอร์ควบคุมส่วนแสดงผล (Board Decoder) ใช้ไอซีเบอร์ 74LS139 (เป็น Dual 1-of-4 Decoder/Demultiplexer) ซึ่งมีสัญญาณเข้าทั้งหมด 2 ขา และมีสัญญาณออกทั้งหมด 4 ขา โดยสัญญาณที่นำเข้าจะนำมาจาก PB3 และ PB4 เพื่อถอดรหัสว่าจะให้เบอร์รับข้อมูลไปพักเอาไว้ โดยสัญญาณเอาท์พุทที่ออกมาจะนำไปเปิดทางให้ไอซี 74LS138 ทำงาน

เมื่อไมโครโปรเซสเซอร์บนไมโครคอนโทรลเลอร์ทำการประมวลผลข้อมูลเรียบร้อยแล้ว ก็จะส่งสัญญาณออกทางพอร์ต โดยเป็นข้อมูลขนาด 8 บิต (คือสัญญาณข้อมูลที่จะส่งไปยังวงจรตัวขับทางแถวเพื่อควบคุมการเปิดปิดของทรานซิสเตอร์เกท) ข้อมูลก็จะถูกส่งมายังส่วนวงจร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ควบคุมและพักข้อมูล โดยถูกส่งไปยังขาอินพุทของไอซีพักข้อมูลทุกตัวต่อขนานตามลำดับขา หลังจากนั้นไมโครโปรเซสเซอร์ก็จะส่งสัญญาณออกมาที่ดีโคดเดอร์ควบคุมการพักข้อมูล จำนวน 3 บิต เพื่อถอดรหัสว่าข้อมูลที่ส่งออกมาก่อนหน้านี้เป็นสัญญาณของข้อมูลแถวใด หรือต้องการให้พอร์ตใดรับสัญญาณข้อมูลเอาไปพักเอาไว้ เมื่อข้อมูลถูกพักไว้เรียบร้อยแล้วไมโครโปรเซสเซอร์ก็จะเริ่มส่งสัญญาณข้อมูล 8 บิตถัดไปเพื่อเป็นข้อมูลการติด-ดับ ของแผงแอลอีดีในส่วนถัดมา(ตามแนวตั้ง) ทั้ง 4 แถวต่อไป และก็จะส่งสัญญาณออกมาให้พักข้อมูลไว้เช่นเดียวกับที่กล่าวมาแล้วทั้ง 6 พอร์ต ซึ่งสัญญาณข้อมูลทั้ง 6 พอร์ตนี้เป็นสัญญาณข้อมูลที่ใช้ควบคุมการเปิด-ปิดทรานซิสเตอร์เกทในส่วนของวงจรขับทางแถว เพื่อควบคุมการติดดับของแอลอีดีในแต่ละแถวอีกที

เมื่อสัญญาณข้อมูลทั้ง 24 แถวถูกเก็บไว้เรียบร้อยแล้ว ไมโครโปรเซสเซอร์ก็จะส่งสัญญาณอีก 8 บิต ซึ่งเป็นสัญญาณข้อมูลที่ใช้ควบคุมการสแกนทางคอลัมน์ โดยเมื่อพอร์ตพักข้อมูลไว้เรียบร้อยแล้ว ไมโครโปรเซสเซอร์ก็จะส่งสัญญาณมาเปิดขาที่ Output Enable ของพอร์ตพักข้อมูลทุกตัวพร้อมกัน โดยใช้การต่อขนานขา Ouput Enable ของทุกตัวเข้าด้วยกัน ทำให้พอร์ตพักข้อมูลทุกตัวส่งสัญญาณออกมาทางขาเอาท์พุทพร้อมกัน เพื่อไปควบคุมส่วนวงจรขับทางแถวให้แสดงผลออกทางบอร์ดพร้อมกันทุกตัว และสัญญาณข้อมูลที่ออกจากพอร์ตพักข้อมูลพอร์ตที่ 7 (พอร์ตควบคุมการสแกนทางคอลัมน์) ก็จะถูกส่งไปยังดีโคดเดอร์ควบคุมส่วนตัวขับทางคอลัมน์ (Column Decoder) ซึ่งมีอยู่ 2 ตัวๆละ 4 บิต เพื่อไปทำการถอดรหัสโดยอินพุทจะเป็นสัญญาณข้อมูลขนาด 4 บิต และสัญญาณข้อมูลเอาท์พุทมีขนาด 10 บิต เพื่อทำการสแกนทางคอลัมน์

การสแกนคอลัมน์จะมีสัญญาณข้อมูล 1 หรือสถานะสูง(High)ออกมาเพียงขาเดียวเท่านั้น ไล่จากตัวแรกไปเรื่อยๆจนกระทั่งถึงคอลัมน์ที่ 20แล้วจึงวนกลับมาที่ตัวแรกเหมือนเดิม โดยจะวนไปเรื่อยๆ

สำหรับวงจรต่างๆของส่วนควบคุมและพักข้อมูล จะมีส่วนบอร์ดควบคุม (Control Board) ทำหน้าที่รับข้อมูลจากไมโครคอนโทรลเลอร์มาทำการพักข้อมูลไว้ที่ไอซีพักข้อมูลเบอร์ 74LS373 โดยในแต่ละบอร์ดจะต้องมีไอซีพักข้อมูลจำนวน 6 ตัว โดยที่ไอซีของพอร์ตที่ 1 ถึงพอร์ตที่ 6 ทำงานกับบอร์ดที่ 1 ส่วนไอซีของพอร์ตที่ 7 ถึงพอร์ตที่ 24 ก็จะทำงานกับบอร์ดที่ 2,3 และ 4 ตามลำดับ เพราะแต่ละตัวทำหน้าที่พักข้อมูลไว้ได้ 8 บิต ซึ่งเท่ากับว่าสามารถพักข้อมูลที่จะสั่งให้แอลอีดีติดเป็นสีใดหรือดับได้ทั้งสิ้น 4 แถวต่อไอซีพักข้อมูล 1 ตัว แต่บอร์ดแสดงผลมีความกว้างทั้งหมด 24 แถว ดังนั้นจึงต้องใช้ไอซีพักข้อมูลเบอร์ 74LS373 ทั้งสิ้น 6 ตัวต่อการพักข้อมูล 1 คอลัมน์

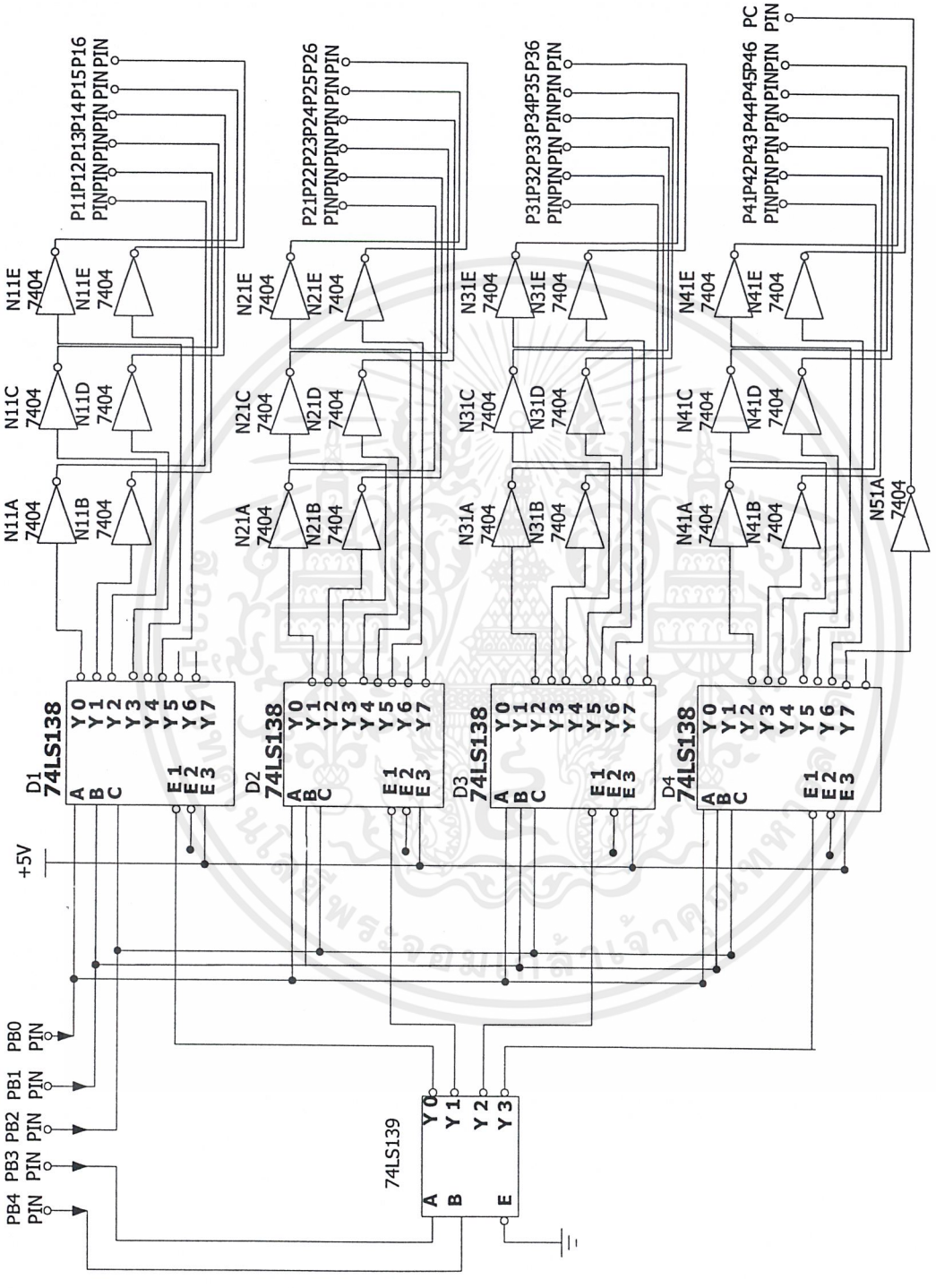
ดังนั้นเมื่อบอร์ดแสดงข้อมูลมีทั้งหมด 4 บอร์ดย่อย จึงต้องมีชุดพักข้อมูลทั้งหมด 4 ชุด นั่นคือต้องมีพอร์ตพักข้อมูล (Port Latch) 24 พอร์ต โดยมีการส่งข้อมูลเข้ามาควบคุมลำดับการพักข้อมูลให้ข้อมูลไปพักไว้ตามพอร์ตพักข้อมูลพอร์ตต่างๆให้ถูกต้องสอดคล้องกับแถวที่ต้องการ โดย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ใช้ไอซีเบอร์ 74LS138 เป็นตัวถอดรหัสข้อมูลที่ส่งเข้ามา โดยข้อมูลที่จะนำมาถอดรหัสนั้นได้รับมาจากพอร์ต B (PB0-PB2) ต่อเข้าที่ขา 1,2 และ 3 (ขา A,B และ C) ของไอซีเบอร์ 74LS138 แบบขนาน โดยต่อกับไอซีเบอร์ 74LS138 ทั้งหมด 4 ตัว เพราะมีพอร์ตพักข้อมูลทั้งหมด 4 ชุด โดยที่ขา Enable Input หรือขาที่ 4 และ 5(G2A , G2B) จะถูกทำให้ทำงานโดยใช้ไอซีเบอร์ 74LS139 เป็นดีโคดเดอร์เลือกว่าจะส่งข้อมูลเข้าไปพักที่บอร์ดใดใน 4 บอร์ดย่อย หรือพักข้อมูลในพอร์ตพักข้อมูลชุดใดนั่นเอง โดยจะเลือกว่าจะให้ไอซีเบอร์ 74LS138 ตัวใดทำงานก่อนหรือหลัง โดยรับข้อมูลเข้าที่ขา 2 และ 3 (หรือขา A และ B) แล้วส่งสัญญาณออกที่ขา Y0 ถึง Y3 หลังจากนั้นต้องนำสัญญาณไปคอมพลีเมนต์ (Complement) โดยใช้อินเวอร์เตอร์ (Inverter) เบอร์ 7404 เพราะเอาท์พุทของไอซีเบอร์ 74LS138 จะทำงานที่ค่าสัญญาณเท่ากับ 0 เมื่อไอซีเบอร์ 74LS138 (D1 ถึง D4) ตัวใดถูกเลือกให้ทำงาน ข้อมูลที่ถูกต่อที่ขา 1,2 และ 3 (หรือ A,B และ C) ก็จะถูกรับเข้าไปถอดรหัส แล้วส่งสัญญาณออกทางขา Y0 ถึง Y5 ไปเปิดทางที่ขาแลตซ์เอนาเบิ้ล (Latch Enable (ขา LE หรือขาที่ 11)) ให้พอร์ตพักข้อมูลแต่ละตัวรับข้อมูลที่อยู่ที่ขาอินพุต (ขา D0 ถึง D7) ไปพักไว้ตามลำดับจนครบทุกพอร์ตของบอร์ดแรก ต่อจากนั้นก็จะทำงานเช่นเดียวกันนี้กับบอร์ดที่ 2 บอร์ดที่ 3 และบอร์ดที่ 4

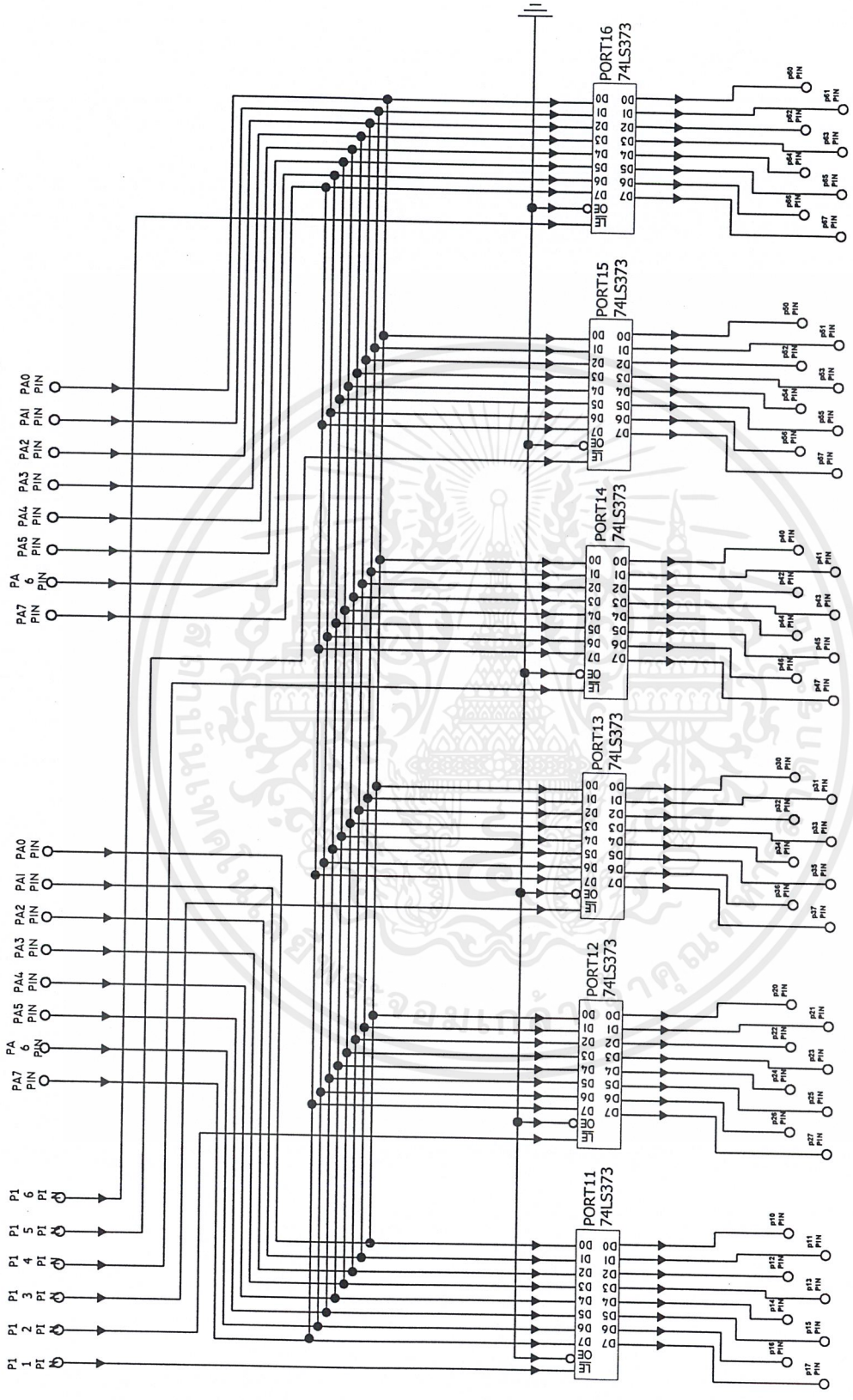
เมื่อส่งข้อมูลไปพักที่พอร์ตข้อมูลครบทั้ง 24 พอร์ตแล้ว ขา Y6 ของไอซีเบอร์ 74LS138 ก็จะส่งสัญญาณไปเปิดทางให้ไอซีเบอร์ 74LS373 COLPORT (Column Port) ทำการรับข้อมูลเข้าไปอีกชุดซึ่งเป็นข้อมูลที่สั่งให้คอลัมน์ใดทำงาน ไอซีที่คอลัมน์พอร์ตจะมีขาเอาท์พุท คือขา Q0 ถึง Q3 ต่อกับไอซีเบอร์ 74LS145 DEC-COL-1 และเอาท์พุทที่ขา Q4 ถึง Q7 DEC-COL-2 ซึ่งเป็นดีโคดเดอร์แบบเข้า 4 ออก 10 ไปทำการถอดรหัส แล้วส่งข้อมูลผ่านอินเวอร์เตอร์ (Inverter) เบอร์ 7404 เพื่อไปคอมพลีเมนต์ (Complement) สัญญาณ เนื่องจากเอาท์พุทของไอซีเบอร์ 74LS145 จะทำงานที่ค่าสัญญาณเป็น 0 สัญญาณที่ได้จะนำไปใช้ควบคุมคอลัมน์ทั้ง 20 คอลัมน์ของแต่ละบอร์ดให้บอร์ดสว่างตามคอลัมน์ที่ต้องการ โดยนำสัญญาณนี้ไปขับทรานซิสเตอร์ที่วงจรตัวขับทางคอลัมน์ให้กระแสสามารถลงกราวน์ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



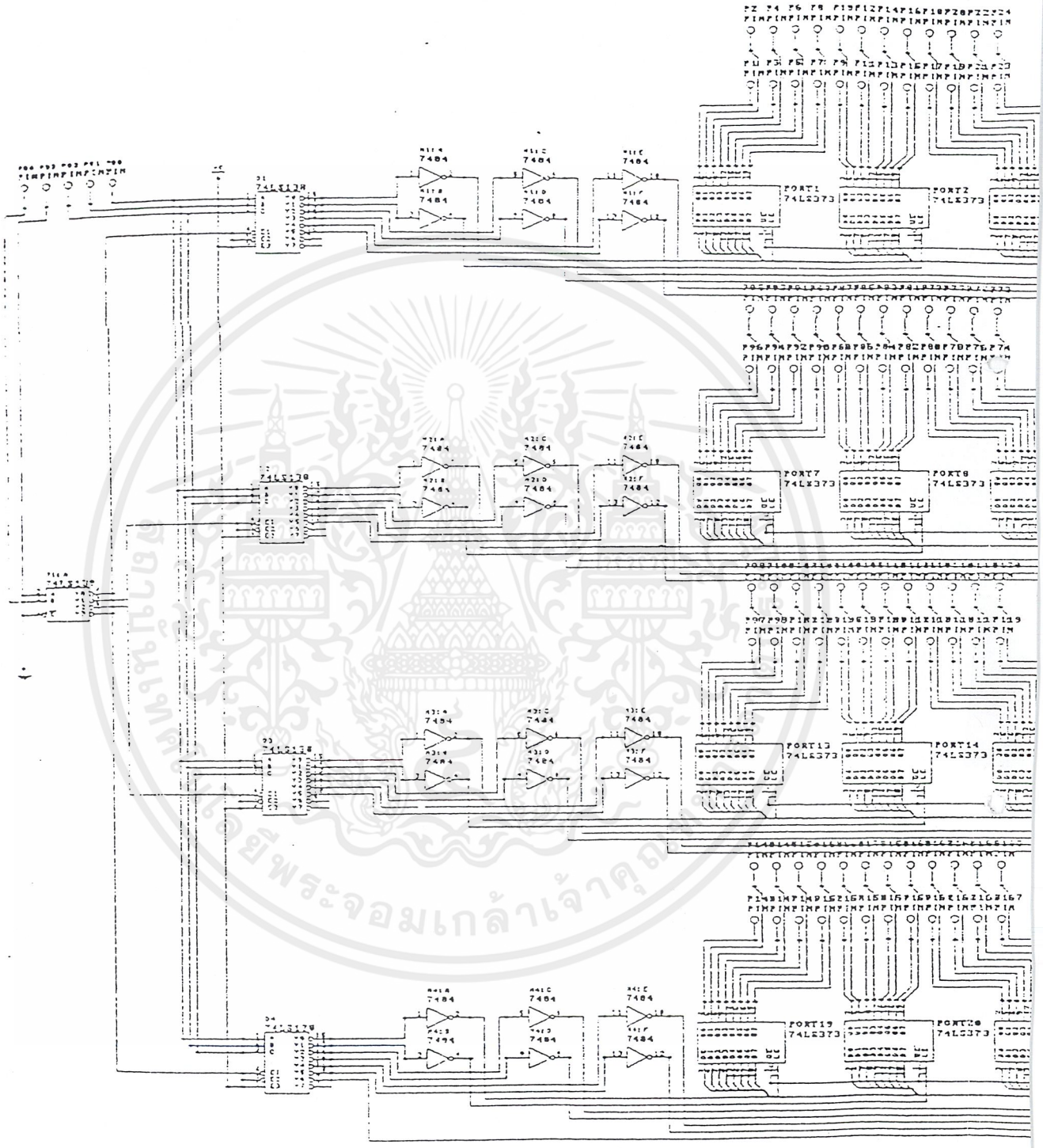
รูปที่ 2.6 แสดงวงจร Decode Port Latch ทั้ง 24 ตัว โดยใช้ 74LS139 และ 74LS138

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.7 แสดงวงจรและการเชื่อมต่อพอร์ตพักข้อมูลของบอร์ดย่อย

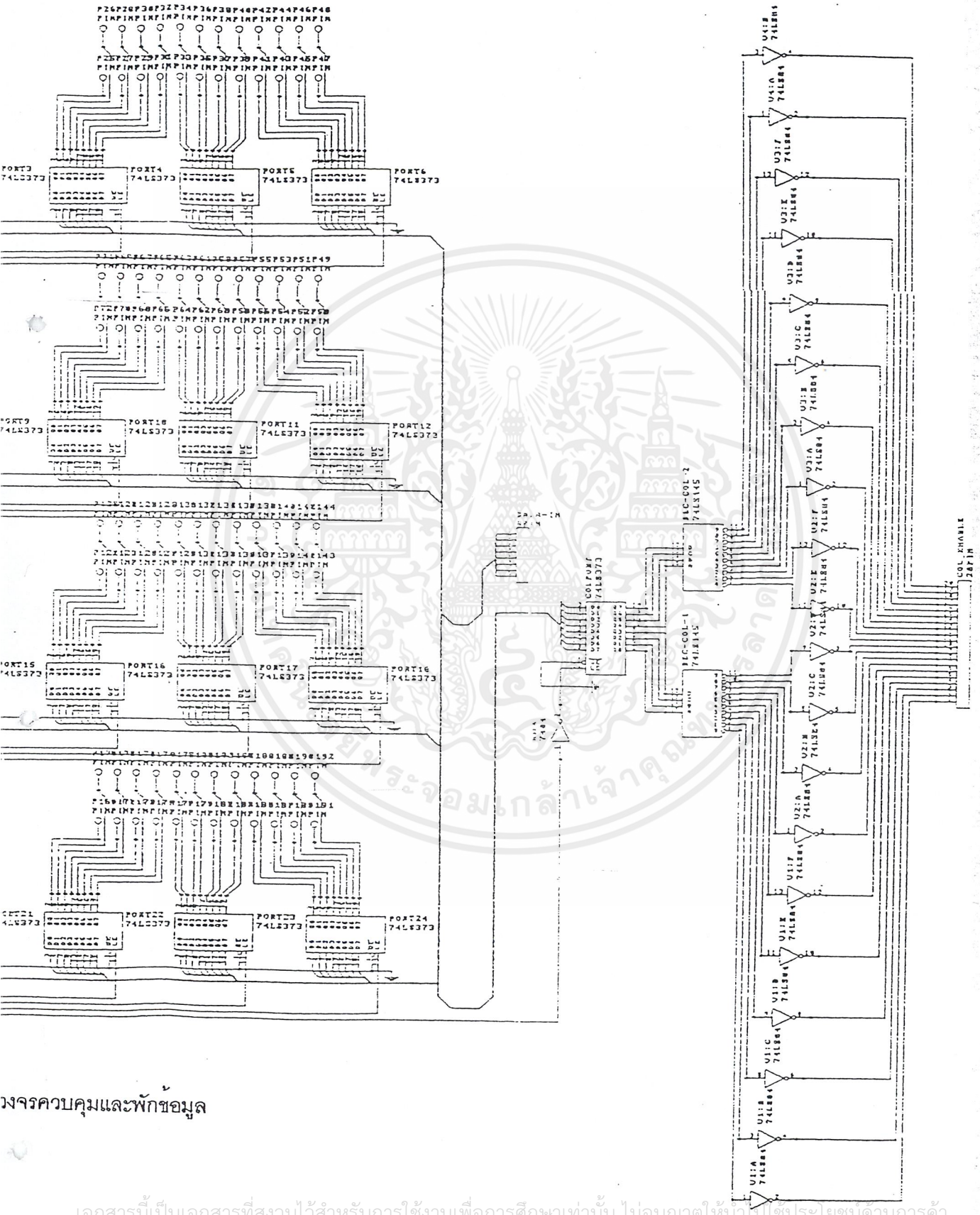
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.8 แสดงวงจรทั้งหมดของ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง



วงจรควบคุมและพักข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

42530

บทที่ 3
ไมโครคอนโทรลเลอร์
(Microcontroller)

3.1 ส่วนประกอบ

ไมโครคอนโทรลเลอร์เป็นตัวกลางระหว่างคอมพิวเตอร์ส่วนบุคคลกับบอร์ดแสดงข้อมูล และทำหน้าที่ควบคุมการแสดงผลของบอร์ดแสดงข้อมูล ในโครงงานนี้ได้เลือกใช้ไมโครคอนโทรลเลอร์ตระกูล Z80 ซึ่งต่ออยู่กับอุปกรณ์อื่นๆบนคอนโทรลบอร์ด โดยคอนโทรลบอร์ดที่เลือกมาใช้งาน คือ บอร์ด CP-JR180 ซึ่งมีส่วนประกอบต่างๆดังต่อไปนี้

- ใช้ซีพียู (CPU) เบอร์ Z80180 แบบ 68 ขา(PIN) ทำงานด้วยความเร็ว 6.144เมกกะเฮิตซ์ (MHZ)
- หน่วยความจำประกอบด้วยหน่วยความจำแบบอ่านอย่างเดียว (ROM) และหน่วยความจำที่สามารถโปรแกรมได้(ROM) ถึง 128 กิโลไบต์
- มีพอร์ท (PORT) 82C55 ให้ใช้งาน 1 ตัว โดยเป็นแบบซีมอส (CMOS) ขนาดเล็ก
- มีวงจรส่วนเวลาจริง (REAL TIME CLOCK) เป็นฐานให้กับระบบโดยใช้เบอร์ 6242
- ใช้กับแรงดัน 7-9 โวลท์ โดยมี ไอซีเร็กกูเรเตอร์ (REGULATOR) 7805 ในตัว
- มีคอนเน็กเตอร์มาตรฐาน สามารถต่อเข้ากับบอร์ดต่างๆได้เป็นอย่างดี
- มีพอร์ทอนุกรม (SERIAL PORT) 2 พอร์ทต่อใช้งานใช้ MAX 232

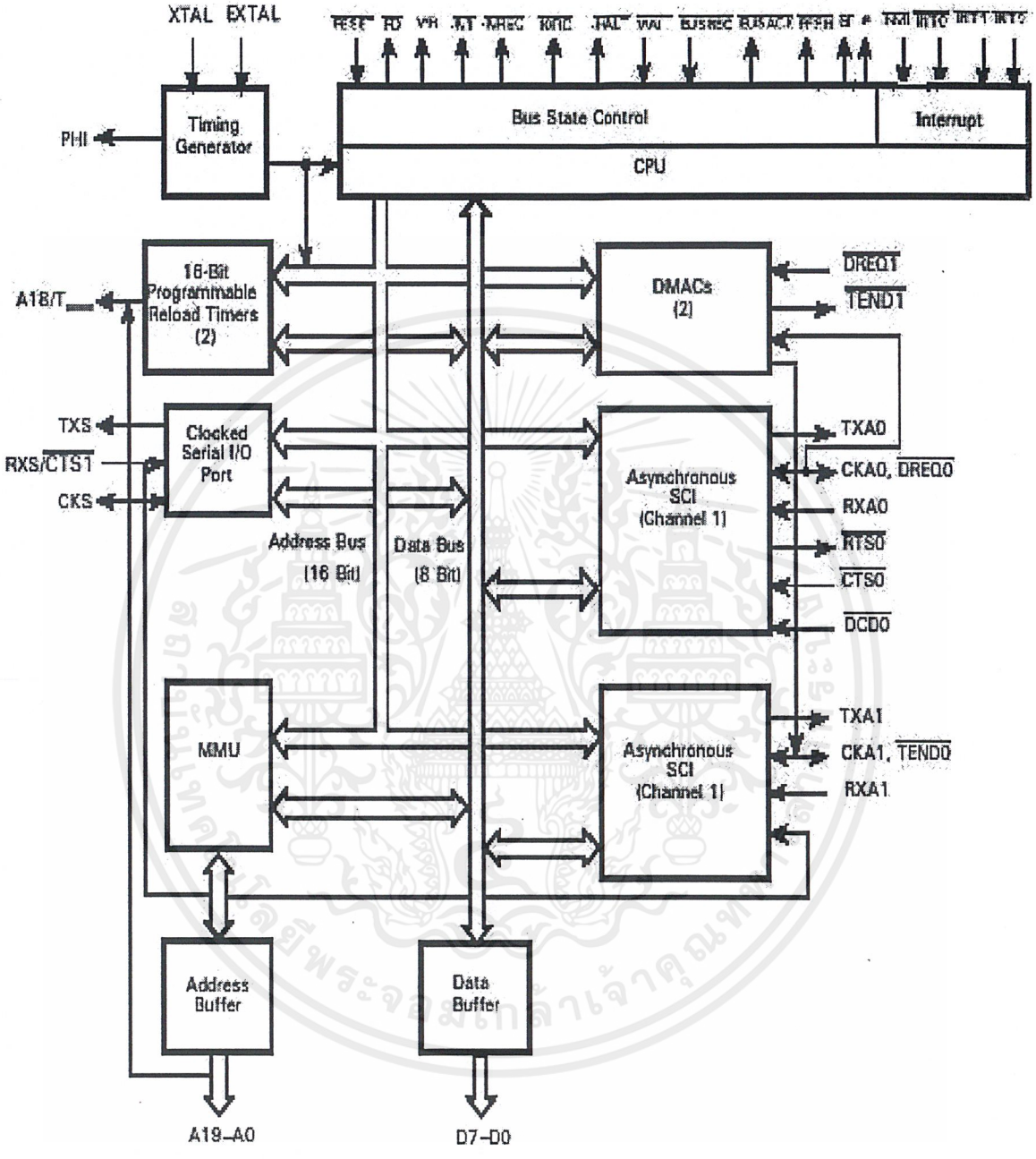
จากส่วนประกอบต่างๆจะเห็นได้ว่า CP-JR180 มีข้อดีที่เป็นประโยชน์ต่อการนำมาใช้งานดังนี้

3.1.1 หน่วยประมวลผลกลาง(CPU)

ใช้ซีพียู (CPU) Z80180 ของบริษัท ZILOG ซึ่งสามารถ ใช้คำสั่งของ Z80 ได้ทั้งหมดและยังเพิ่มอีก 12 ชุดคำสั่งใช้งาน ในบอร์ด JR180 นี้เลือกใช้ Z80180 ความเร็วขนาด 6 เมกกะเฮิตซ์(MHZ) แต่ก็สามารถใช้กับความถี่ 6.144 เมกกะเฮิตซ์(MHZ)ได้ด้วย ทำให้การทำงาน 1 คำสั่งใช้เวลาเพียง 0.48ไมโครวินาที(uSEC)นอกจากนี้Z80180ยังมีความคล่องตัวมากเนื่องจากZ80180 ประกอบด้วย

- (Memory Management Unit,MMU) อ้างอิงหน่วยความจำได้ถึง 1 เมกกะไบต์(Megabyte)
- ตัวควบคุมการถ่ายข้อมูล (Direct Memory Access Controller, DMAC) 2 ช่อง (Channel) ซึ่งช่วยเพิ่มความเร็วในการโอนถ่ายข้อมูลโดยไม่ต้องผ่านหน่วยประมวลผลกลาง(CPU)
- แหล่งสัญญาณอินเทอร์รัพท์(Interrupt Source) มีทั้งหมด 12 แหล่ง โดยแบ่งเป็น 4 อินเทอร์รัพท์ภายนอก และ 8 อินเทอร์รัพท์ภายใน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.1 บล็อกไดอะแกรมโครงสร้างภายใน Z80180

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.2 หน่วยความจำ (MEMORY)

CP-JR180 สามารถต่อใส่หน่วยความจำได้สูงสุด 128 กิโลไบต์ (ON BOARD) โดยใช้ไอซีดีโคดแบ่งหน่วยความจำเป็นช่วงๆได้ 8 ช่วง ช่วงละ 32 กิโลไบต์

ซอกเก็ต (SOCKET) U2 สามารถใส่ EPROM ขนาด 64 กิโลไบต์ (27512) หรือ 32 กิโลไบต์ (27256) โดยใช้จัมเปอร์(JUMPER) J1 เป็นตัวเลือกเบอร์หน่วยความจำ (EPROM) และใช้ไดโอด (DIODE) เบอร์ 1N4148 2 ตัวต่อในลักษณะแอนเกท (AND GATE) ให้ดีโคด (DECODE) ได้ 2 ช่วงแอดเดรส (ADDRESS) U2 นี้หน่วยความจำเริ่มจาก 0000H ถึง FFFFH

ซอกเก็ต(SOCKET) U3 สามารถใส่แรม (RAM) ขนาด 32 กิโลไบต์ (62256) หรือ 8 กิโลไบต์ (6264) ได้โดยใช้ จัมเปอร์(JUMPER) J4 เป็นตัวเลือกเบอร์แรม (RAM) ในวงจรส่วนนี้เราสามารถใส่ไฟสำรอง (BATTERY) ขนาด 3 โวลท์ เพื่อสำรอง (BACKUP) ข้อมูลในแรม (RAM) ได้ด้วย โดยใช้มอสเฟต (MOSFET) เบอร์ BS170 เป็นส่วนกันสัญญาณรบกวนจากการเปิดปิดระบบไฟไม่ให้เข้าไปรบกวนขา CS ของแรม (RAM) เพื่อผลการ สำรอง(BACKUP) ที่ดี U3 นี้หน่วยความจำเริ่มจาก 10000H ถึง 17FFFH

ซอกเก็ต(SOCKET) U4 เราสามารถใส่แรม (RAM) หรือรอม (ROM) ได้โดยใช้จัมเปอร์สวิตซ์ (JUMPER SW) เป็นตัวเลือกเบอร์ไอซีที่เราจะใส่ และใช้จัมเปอร์ (JUMPER) J3 เป็นตัวประกอบในกรณีที่จะใช้ แรม(RAM) และก็ต้องการสำรอง (BACKUP) ข้อมูลด้วย U4 นี้หน่วยความจำเริ่มจาก 18000H ถึง 1FFFFH

เนื่องจาก Z80180 สามารถอ้างอิงหน่วยความจำได้ 1024 กิโลไบต์ แต่ในชุดคำสั่งของ Z80 ไม่มีคำสั่งใดที่จะอ้างข้ามเกิน 64 กิโลไบต์ได้ ดังนั้นจึงใช้หน่วยจัดการหน่วยความจำ(MMU) มาเป็นตัวจัดการในการเข้าถึงตำแหน่งตำแหน่งทั้ง 1024 กิโลไบต์ ซึ่งในการปฏิบัติงาน จะถูกแยกเป็น 2 แบบคือ

- ส่วนของผู้ใช้ (User) เรียกว่าตำแหน่งการเรียกใช้ในทางโปรแกรม(Logical Address) (0000H-FFFFH) เป็นส่วนที่ถูกเรียกใช้ในโปรแกรมที่ผู้ใช้เขียนขึ้น มีขอบเขต64 กิโลไบต์
- ส่วนของหน่วยประมวลผลเรียกว่าตำแหน่งทางกายภาพ (Physical Address) (00000H-FFFFFH)เป็นส่วนที่หน่วยประมวลผลใช้ในการปฏิบัติงานจริง1024 กิโลไบต์

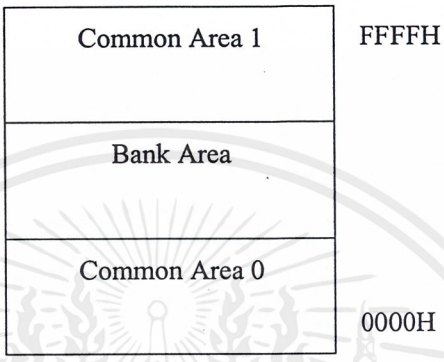
Logical Address 64 กิโลไบต์ จะถูกแยกเป็น 3 ส่วนดังนี้

- Common Area 0,1 ส่วนนี้เมื่อเรียกตำแหน่งทางตรรกะ ที่กำหนดไว้ไม่ว่าการทำงานของหน่วยประมวลผลกลางจะอยู่ในตำแหน่งทางกายภาพจริงที่ใดก็ตาม จะกลับมายังตำแหน่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ที่เป็น Common นั้น นั่นก็คือ Common จะตามหน่วยประมวลผลกลางไปทุกๆตำแหน่งที่กำลังปฏิบัติการอยู่

- Bank Area จะมีลักษณะการทำงานเป็นเพจ(Page) เมื่อย้ายตำแหน่งที่เกินเพจที่กำหนดไปยังเพจอื่น ก็จะไม่สามารถติดต่อเพจก่อนหน้านี้ได้



รูปที่3.2 แสดงตำแหน่งหน่วยความจำเรียกใช้ในทางโปรแกรม

<u>การกำหนดตำแหน่งในการใช้งานของตำแหน่งทางตรรกะ</u>

จะใช้รีจิสเตอร์ ชื่อ COMMON/BANK AREA REGISTER (CBAR:I/O ADDRESS =3AH) ซึ่งใช้กำหนดตำแหน่งการเรียกใช้ในทางโปรแกรม ขนาด 1 ไบต์โดยแบ่งออกเป็น 2 นิบเบิ้ล (Nibbles) คือ

HIGH NIBBLE ใช้กำหนดค่าของ CA (COMMON AREA 1 (D7-D4))

LOW NIBBLE ใช้กำหนดค่าของ BA (BANK AREA (D3-D0))

<u>ตัวอย่าง</u> กำหนดให้ CBAR = F8 ดังนั้น CA = F และ BA = 8 ตำแหน่งการเรียกใช้ทางโปรแกรมจะเป็นดังนี้

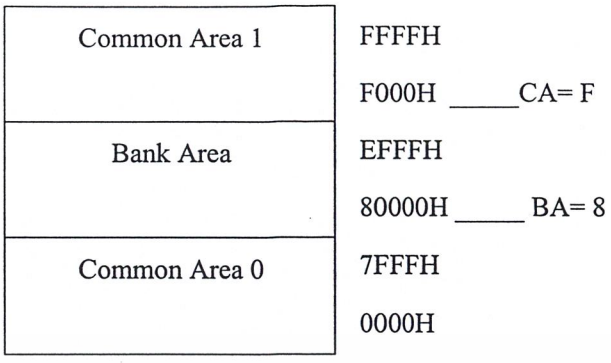
0000H -------- BA-1 BA--------- CA-1 CA---------- FFFFH

COMMON AREA0 เมื่อถูกเรียกใช้ทางโปรแกรมอยู่ในขอบเขต 32 กิโลไบต์ จากตำแหน่ง 0000H-7FFFH

BANK AREA จะมีขอบเขตในการเรียกใช้ทางโปรแกรม 28 กิโลไบต์ จากตำแหน่ง 8000H – EFFFH

COMMON AREA 1 จะมีขอบเขตในการเรียกใช้ทางโปรแกรม 4 กิโลไบต์ จากตำแหน่ง F000H – FFFFH

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่3.3 แสดงตัวอย่างการอ้างอิงหน่วยความจำ

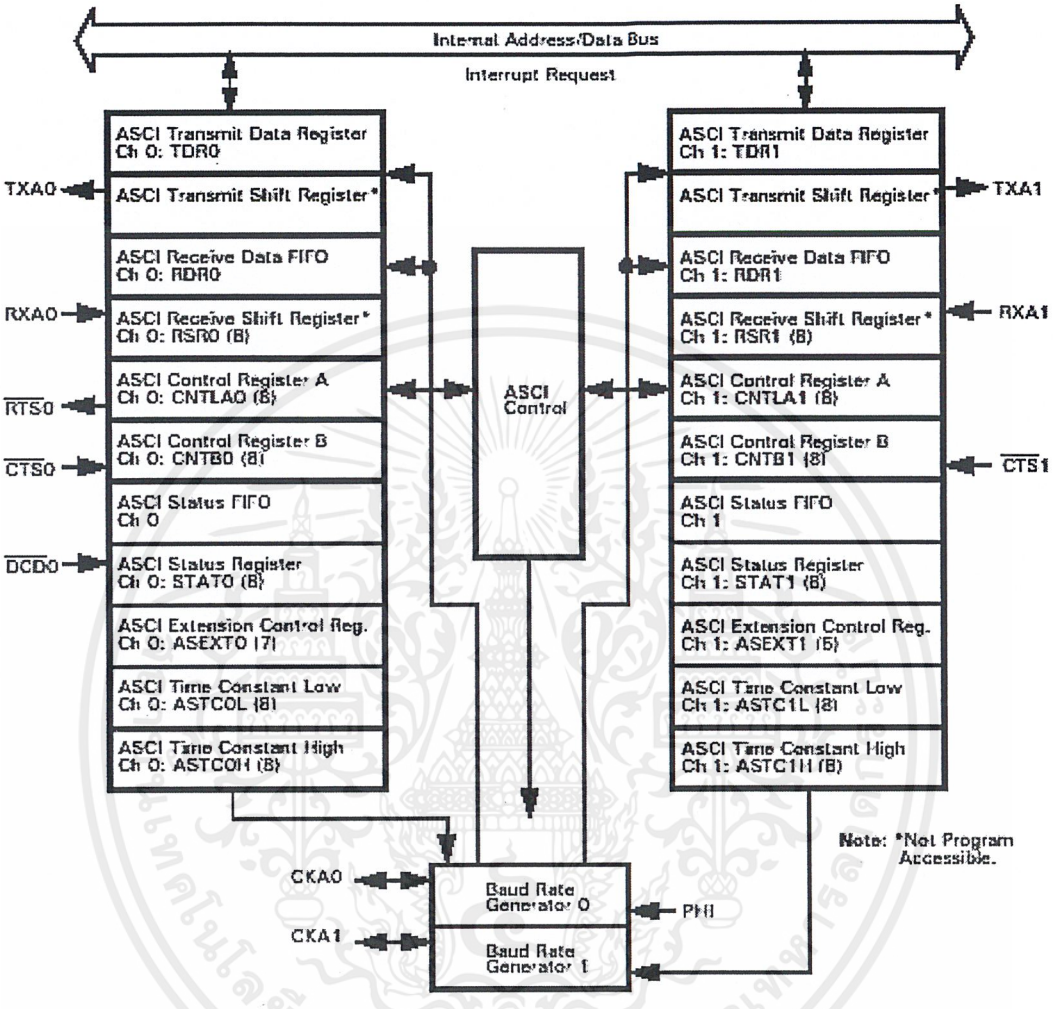
ในส่วนของBANK AREA และ COMMON AREA1 สามารถกำหนดตำแหน่งการใช้งานจริงว่าให้อยู่ส่วนใดของหน่วยความจำขนาด 1024 กิโลไบต์ ได้จากค่ารีจิสเตอร์ขนาด 8 บิต คือ BANK BASE REGISTER (BBR:ADDRESS 39H) และ COMMON BASE REGISTER(CBR:ADDRESS 38H) ตามลำดับโดย

ตำแหน่งใน 1024 กิโลไบต์ (Physical Address) = BANK AREA หรือ COMMON AREA + (BBR หรือ CBR*1000H)

3.1.3 พอร์ทอนุกรม (Serial Port) คือ ส่วนเอเอสซีไอ (ASCI ,Asynchronous Serial Communication Interface) มีด้วยกัน 2 ช่อง ดังแสดงในรูปที่ 3.2 สำหรับรายละเอียดและการใช้งานของรีจิสเตอร์แต่ละตัวจะอยู่ในส่วนภาคผนวก

หลักการทำงานของเอเอสซีไอเริ่มจากข้อมูลจะถูกส่งเข้ามาทางขา RXA ของเอเอสซีไอ แล้วถูกส่งเข้ามาเก็บในรีจิสเตอร์ RSR และเมื่อรับเข้ามาครบ 1 ไบต์แล้ว ข้อมูลทั้งหมดจะถูกส่งไปเก็บในรีจิสเตอร์ RDR โดยข้อมูลไบต์ถัดไปจะส่งเข้ามาได้ก็ต่อเมื่อข้อมูลไบต์ก่อนหน้านี้ถูกส่งไปเก็บในรีจิสเตอร์ RDR เรียบร้อยแล้ว โดยปกติจะไม่มีข้อมูลจากบัสข้อมูล (Data Bus) ส่งเข้ามาเก็บในรีจิสเตอร์ RDR เนื่องจากเป็นรีจิสเตอร์ที่ใช้อ่านได้อย่างเดียว (Read Only Register) แต่ในกรณีที่บิต RDRF (Receive Data Register) ในรีจิสเตอร์สถานะของเอเอสซีไอ (ASCI Status Register) มีค่าเป็น 0 เราก็สามารถเขียนข้อมูลลงในรีจิสเตอร์ RDR ได้เช่นกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.4 แสดงบล็อคไดอะแกรมของส่วนประกอบต่างๆของเอเอสซีไอ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2 การทำงานของไมโครคอนโทรลเลอร์

การทำงานของไมโครคอนโทรลเลอร์แบ่งออกเป็น 2 ส่วน คือ

3.2.1 การรับข้อมูลจากคอมพิวเตอร์ส่วนบุคคล

ข้อมูลที่ส่งมาจากคอมพิวเตอร์จะถูกส่งมาทางพอร์ตอนุกรมของคอมพิวเตอร์ด้วยการจัดการของโปรแกรมวิชวลเบสิก (Visual Basic) ไมโครคอนโทรลเลอร์จะรับข้อมูลเข้ามาทางพอร์ตอนุกรมคือส่วนที่เรียกว่า เอเอสซีไอ (รูปที่3.2) โดยคณะผู้จัดทำได้เลือกใช้เอเอสซีไอแชนเนล 1

จากหลักการการทำงานของเอเอสซีไอทำให้สามารถออกแบบการรับข้อมูลโดยหลังจากเซตรูปแบบของข้อมูลที่จะรับให้เป็น 1 BIT START+8 BIT DATA+1 BIT STOP และเซตอัตราการรับข้อมูลเป็น 19200 บิตต่อวินาทีแล้ว ก็จะเริ่มรับข้อมูลโดยการตรวจสอบรีจิสเตอร์ STAT1 ในบิตที่ 7 (RDRF) หากบิตนี้ถูกเซตเป็น 1 แสดงว่ามีข้อมูลเข้ามา ก็จะทำการอ่านข้อมูลจากรีจิสเตอร์ RDR 1 เพื่อนำข้อมูลไปเก็บไว้ในหน่วยความจำชั่วคราวของไมโครคอนโทรลเลอร์ หลังจากนั้นจึงทำการรับข้อมูลไบต์ถัดไปเข้ามาเรื่อยๆจนกว่าจะครบตามจำนวนที่โปรแกรมวิชวลเบสิกได้แปลงจากรูปภาพหรือข้อความไว้ ซึ่งจะมีข้อมูลส่วนที่บอกจำนวนของข้อมูลทั้งหมดอยู่ตอนต้นของข้อมูลที่ส่งมาด้วย ข้อมูลทั้งหมดที่รับเข้ามาจะถูกเก็บไว้ในหน่วยความจำตั้งแต่ตำแหน่ง 8000H โดยในโครงงานนี้ได้ต่อหน่วยความจำทั้งหมด 40 กิโลไบต์ทำให้สามารถเก็บข้อมูลในหน่วยความจำได้ถึงตำแหน่ง 18FFH

ในช่วงที่ไมโครคอนโทรลเลอร์ควบคุมการแสดงผลของบอร์ดแสดงข้อมูลนั้น จะมีการตรวจสอบบิต RDRF ของรีจิสเตอร์ STAT1 อยู่เสมอ หากมีการส่งคำสั่งและข้อมูลใหม่เข้ามาก็จะออกจากโปรแรกมส่วนควบคุมการแสดงผลเพื่อกลับไปทำงานในส่วนของโปรแกรมการรับข้อมูล

3.2.2 การควบคุมการแสดงผลของบอร์ดแสดงข้อมูล

เมื่อรับข้อมูลจากคอมพิวเตอร์ครบแล้ว คอนโทรลบอร์ดก็จะเริ่มการควบคุมการแสดงผลของบอร์ดแสดงข้อมูลโดยเริ่มจากการตรวจสอบข้อมูลบอกโหมดการแสดงผลของบอร์ดแสดงข้อมูลซึ่งส่งมาจากคอมพิวเตอร์พร้อมกับข้อมูลที่จะแสดง แล้วคอนโทรลบอร์ดก็จะเริ่มทำการส่งข้อมูลไปแสดงผลตามโปรแกรมของแต่ละโหมดแสดงผลที่ได้กำหนดไว้ ซึ่งมีรายละเอียดดังนี้

3.2.2.1 การแสดงข้อมูลแบบภาพนิ่ง

เป็นการส่งข้อมูลภาพ 1 ภาพ หรือข้อความที่มีขนาดไม่เกิน 1 หน้าจอของบอร์ดแสดงข้อมูล หรือขนาด 24 x 80 จุด โดยส่งไปทีละ 1 คอลัมน์จนครบทั้ง 4 บอร์ดแล้วจึงทำการสแกน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คอลัมน์นั้น หลังจากนั้นจึงส่งข้อมูลไปยังคอลัมน์ที่2 ต่อไปวนทำอย่างนี้ไปเรื่อยๆจนครบทั้ง 80 คอลัมน์ก็จะได้การแสดงรูปภาพ 1 เฟรม หากไม่มีการสั่งงานใดๆจากคอมพิวเตอร์เข้ามาก็จะวนแสดงผลข้อมูล 1 เฟรมนี้เรื่อยๆ

3.2.2.2 การแสดงข้อมูลแบบภาพเคลื่อนไหว

แบ่งออกเป็น 3 รูปแบบด้วยกันคือ

- การแสดงรูปภาพ หรือตัวอักษรใน 4 ทิศทาง

เป็นการแสดงข้อความ หรือรูปภาพขนาด 24 x 80 จุด(หรือเท่ากับขนาดของบอร์ดแสดงข้อมูล) และเคลื่อนที่ได้ 4 ทิศทาง คือ เลื่อนจากขวาไปซ้าย เลื่อนจากซ้ายไปขวา เลื่อนจากล่างขึ้นบน และเลื่อนจากบนลงล่าง

การแสดงข้อมูลของทั้ง 4 ทิศทาง จะต้องส่งข้อมูลขนาด 1 เฟรม (24 x 80 จุด) รวมกับข้อมูลที่กำหนดรูปแบบการแสดงภาพ (Control Word) ซึ่งจะเป็นตัวบอกทิศทางแก่ไมโครคอนโทรลเลอร์ เมื่อไมโครคอนโทรลเลอร์ทราบว่าจะต้องทำงานในลักษณะใด ก็จะเรียกใช้งานโปรแกรมย่อยที่เตรียมไว้ได้ถูกต้อง โดยหลักการในการแสดงแสดงภาพเลื่อนทั้ง 4 ทิศทางจะใช้การหมุนข้อมูลในหน่วยความจำ เช่น ถ้าเป็นการเลื่อนไปทางซ้ายก็จะทำการสแกนข้อมูลทั้งเฟรม1 รอบก่อน หลังจากนั้นจึงเลื่อนข้อมูลขึ้นมา 1 คอลัมน์ (06H)ในทุกๆแอดเดรส(Address) แล้วทำการสแกนทั้ง 80 คอลัมน์อีก 1 รอบ วนทำเช่นนี้ไปเรื่อยๆก็จะได้ภาพที่มีการเลื่อนไปทางซ้าย จนกว่าจะมีคำสั่งใหม่มาจากคอมพิวเตอร์ส่วนบุคคล

- การแสดงข้อความต่อเนื่อง

การแสดงข้อความต่อเนื่อง คือการส่งข้อมูลที่เป็นข้อความจำนวนมาก(มากกว่า 1 เฟรม) รวมกับข้อมูลที่บอกรูปแบบการแสดงภาพแบบข้อความต่อเนื่อง และจำนวนของข้อมูลที่ส่งมาทำการเก็บไว้ในหน่วยความจำหลักของไมโครคอนโทรลเลอร์ แล้วจึงทยอยนำข้อมูลออกแสดงผลที่บอร์ดแสดงข้อมูลจนครบตามจำนวนของข้อมูลที่ส่งมา โดยจะทำการเลื่อนตำแหน่งเริ่มของข้อมูลที่จะนำออกแสดงผลที่เก็บไว้ในหน่วยความจำที่เก็บข้อมูล ทีละ 6 ไบต์ แล้วทำการแสดงภาพเช่นเดียวกับการแสดงภาพนิ่ง ซึ่งก็คือการเลื่อนจากภาพขวาไปซ้ายทีละคอลัมน์นั่นเอง โดยข้อมูล 1 หน้า จะมีการเลื่อนตำแหน่งของหน่วยความจำทั้งหมด 80 ครั้ง ดังนั้นข้อมูลของข้อความต่อเนื่องจำนวน N หน้า จะต้องทำการเลื่อนทั้งหมด 80 x N ครั้ง จึงจะแสดงข้อมูลที่เป็นข้อความได้ครบ

เมื่อแสดงข้อมูลที่เป็นข้อความบนบอร์ดแสดงข้อมูลจนครบแล้ว และไม่มีข้อมูลใหม่เข้ามาก็จะวนแสดงข้อความเดิมไปเรื่อย ๆ จนกว่าจะมีข้อมูลใหม่ส่งเข้ามา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เนื่องจากข้อจำกัดของหน่วยความจำหลักของไมโครคอนโทรลเลอร์อยู่ที่ 40 กิโลไบต์ดังนั้นข้อมูลที่เป็นข้อความได้ 340 – 650 ตัวอักษร (ที่เป็นค่าประมาณเนื่องจากตัวอักษรแต่ละตัวใช้พื้นที่ในหน่วยความจำไม่เท่ากัน)

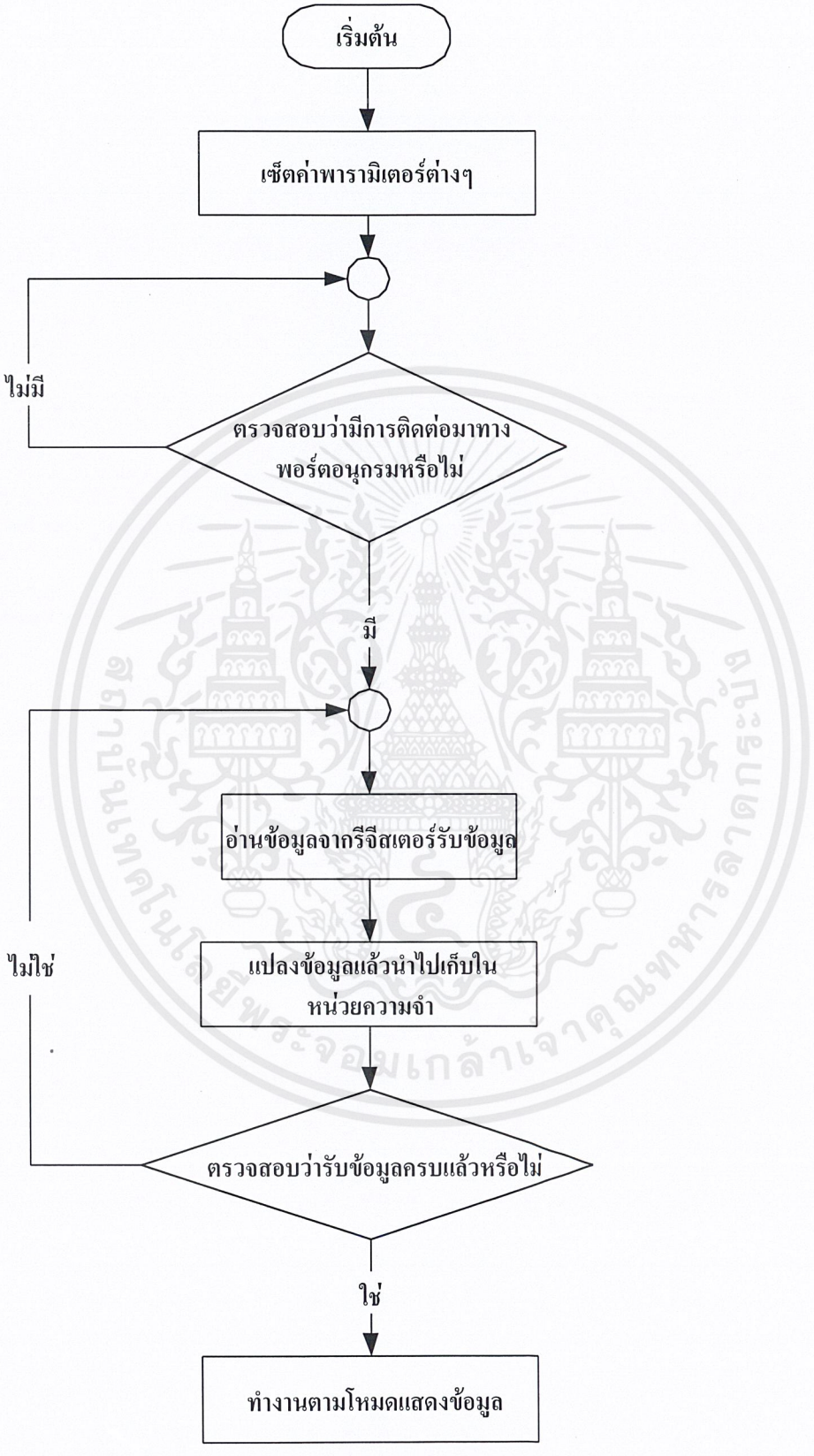
- <u>การแสดงภาพเคลื่อนไหว</u>

การแสดงภาพเคลื่อนไหวคือ การแสดงภาพนิ่งติดต่อกันหลาย ๆ เฟรม ซึ่งแต่ละเฟรมจะเป็นรูปภาพแสดงขั้นตอนการเปลี่ยนแปลงของภาพเคลื่อนไหว โดยแต่ละเฟรมสามารถสร้างได้จากโปรแกรมจำลองบอร์ดแสดงข้อมูลบนหน้าจอคอมพิวเตอร์ และนำแต่ละเฟรมมาบันทึกต่อ ๆ กันก็จะได้ข้อมูลของภาพเคลื่อนไหว บอร์ดแสดงข้อมูลก็จะนำข้อมูลออกแสดงผลทีละเฟรมทำให้เกิดเป็นภาพเคลื่อนไหวได้

เนื่องจากข้อจำกัดของหน่วยความจำหลักของไมโครคอนโทรลเลอร์อยู่ที่ 40 กิโลไบต์ดังนั้นข้อมูลที่เป็นภาพเคลื่อนไหวจึงจำกัดอยู่ที่ 85 เฟรม

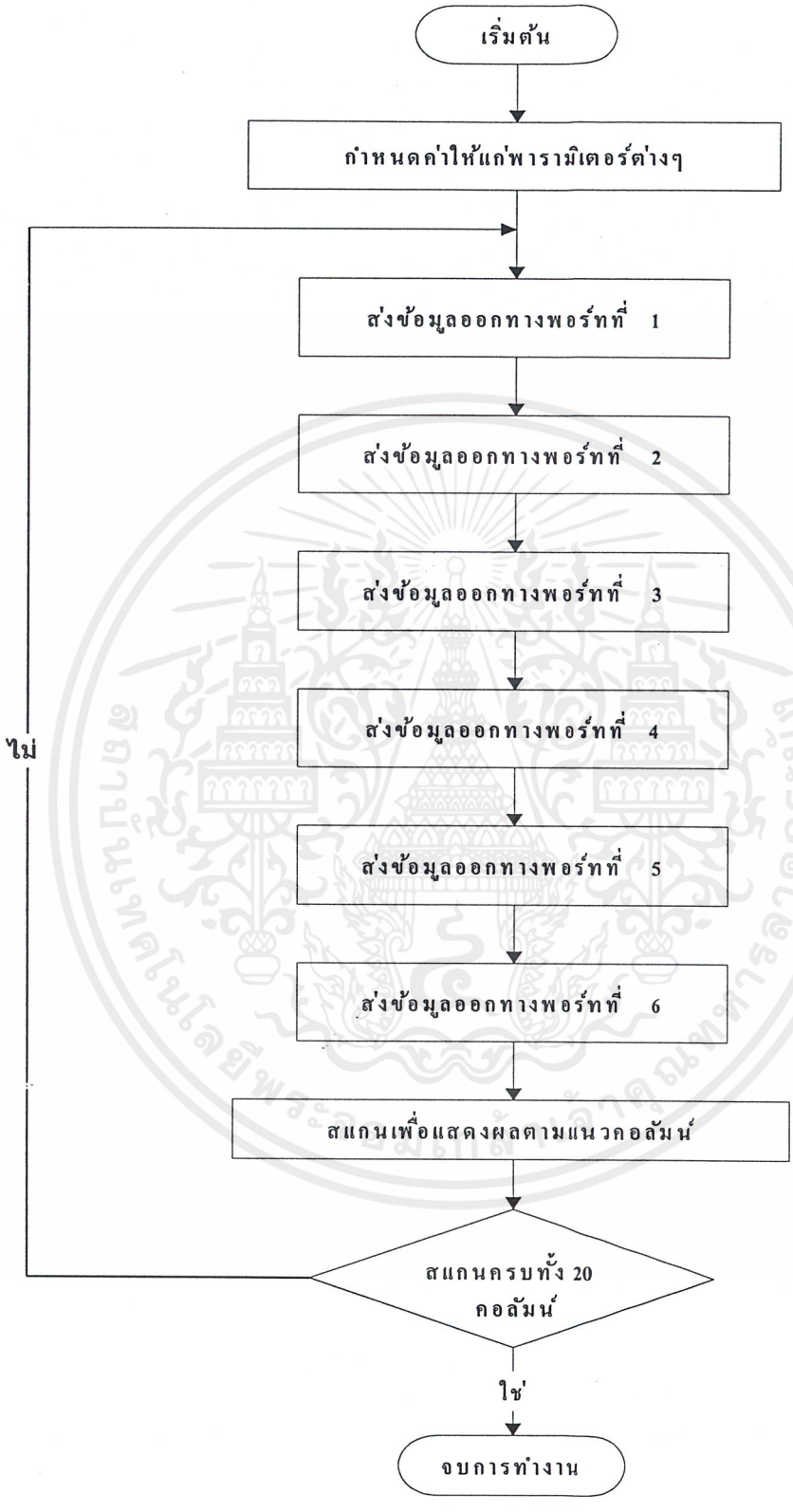
ส่วนโปรแกรมควบคุมการทำงานของไมโครคอนโทรลเลอร์จะมีรายละเอียดอยู่ในส่วนภาคผนวก ซึ่งจะมีลักษณะการทำงานเป็นไปตามผังงาน (Flowchart) ดังต่อไปนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



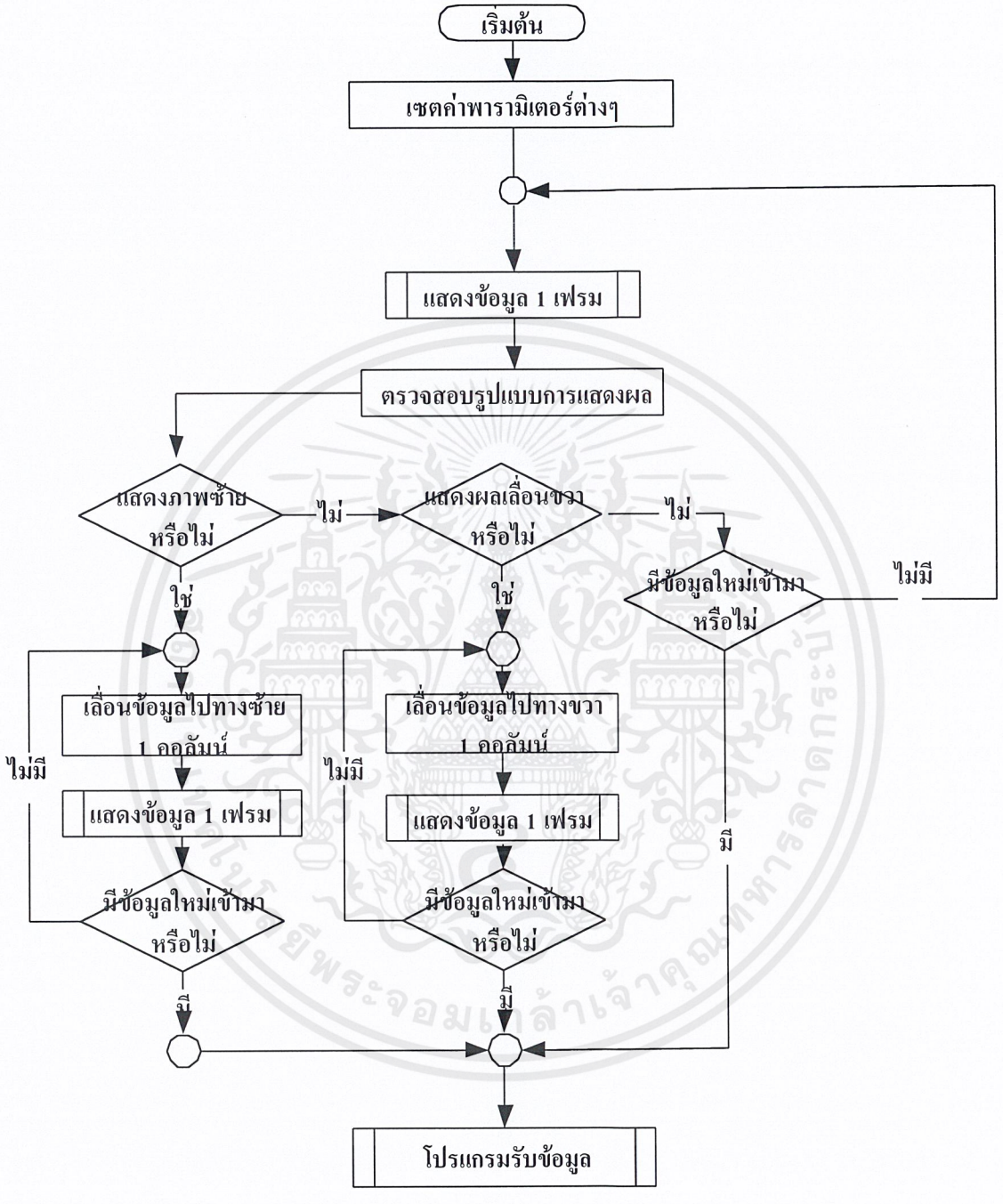
รูปที่3.5แสดงผังงานของโปรแกรมบนไมโครคอนโทรลเลอร์ในการรับข้อมูลจากคอมพิวเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



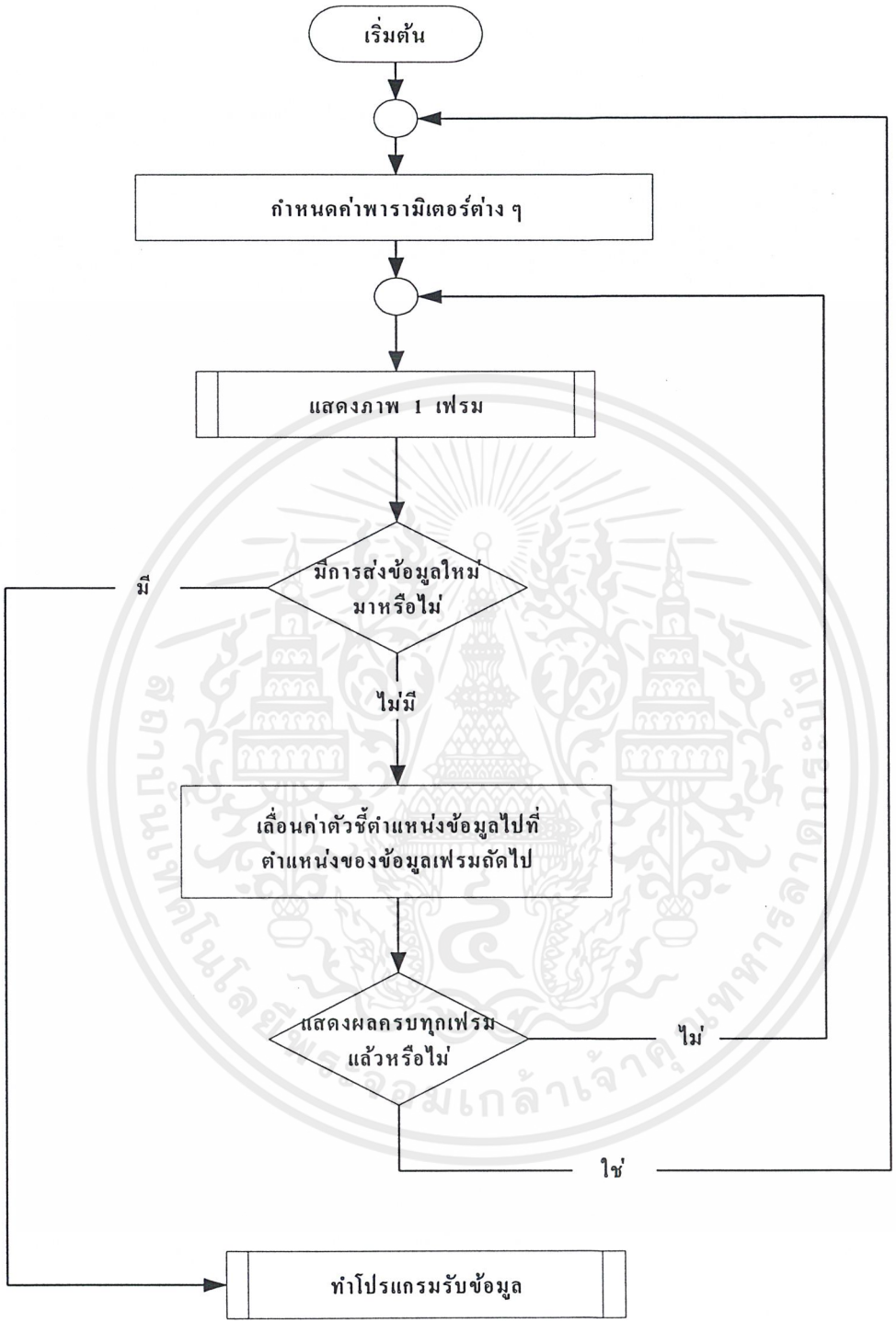
รูปที่ 3.6 แสดงผังงานของโปรแกรมบนไมโครคอนโทรลเลอร์ในการส่งข้อมูลไปบอร์ดแสดงผล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



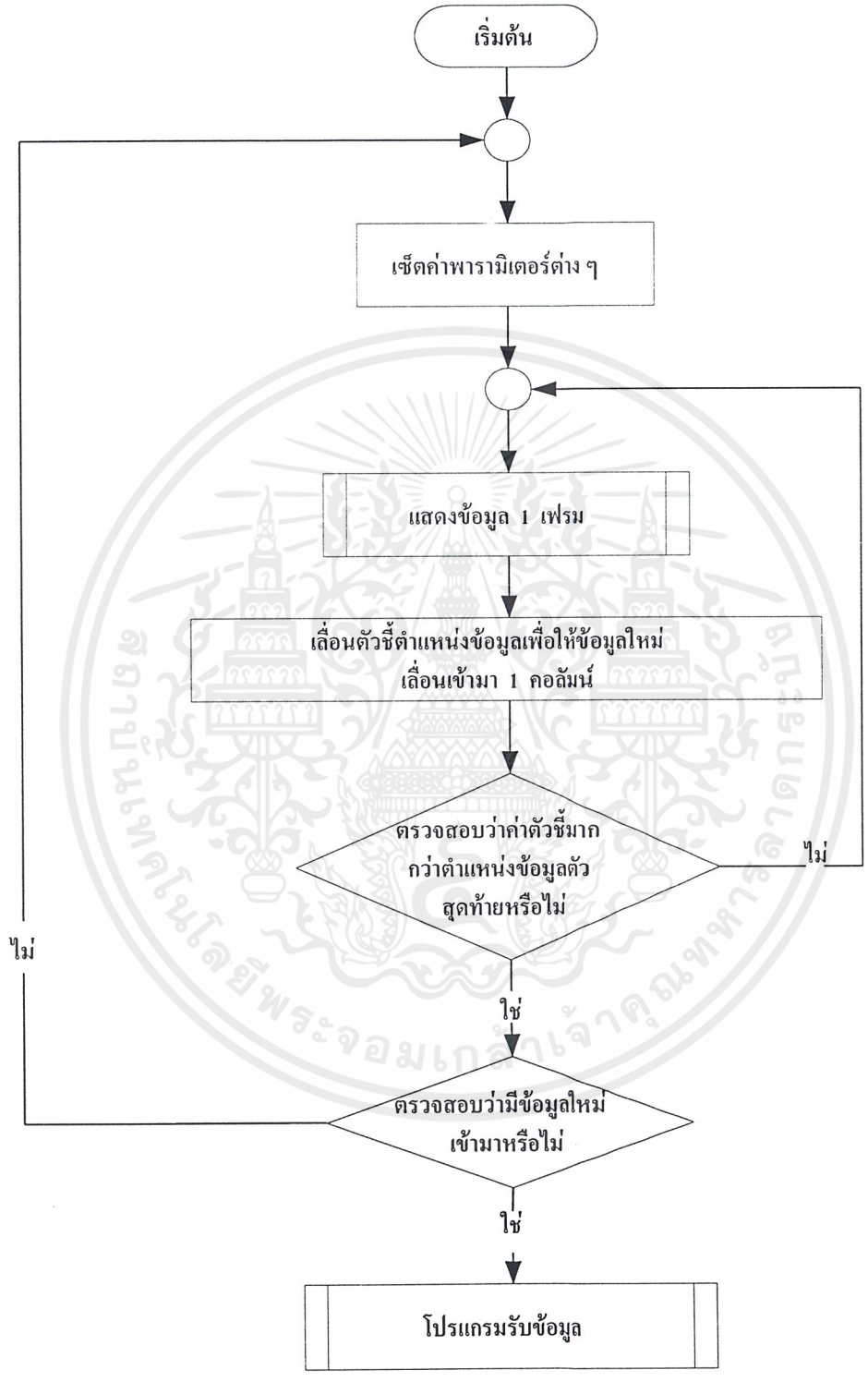
รูปที่ 3.7 แสดงผังงานของโปรแกรมบนไมโครคอนโทรลเลอร์ ในการแสดงภาพนิ่งและภาพเลื่อนขนาด 1 เฟรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.8 แสดงผังงานของโปรแกรมบนไมโครคอนโทรลเลอร์ในการแสดงภาพเคลื่อนไหว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.9 แสดงผังงานของโปรแกรมบนไมโครคอนโทรลเลอร์ในการแสดงข้อความต่อเนื่อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

คอมพิวเตอร์ส่วนบุคคล

(Personal Computer)

คอมพิวเตอร์ส่วนบุคคลจะเป็นส่วนที่ผู้ใช้งาน (User) ใช้สำหรับกำหนดให้หลอดแอลอีดี แต่ละหลอดบนบอร์ดแสดงผล (Display Board) ติดเป็นสีต่าง ๆ โดยจะมีโปรแกรมที่จำลองบอร์ดแสดงผลอยู่บนหน้าจอ และใช้เมาส์(Mouse)ในการเลือกสี ตำแหน่ง ของหลอดแอลอีดี ซึ่งโปรแกรมก็จะใช้ข้อมูลดังกล่าวในการสร้างข้อมูลที่จะส่งไปให้ไมโครคอนโทรลเลอร์ เพื่อให้ไมโครคอนโทรลเลอร์ทำการแสดงภาพที่บอร์ดแสดงผล ซึ่งจะได้ภาพที่แสดงเหมือนกับที่จำลองบนหน้าจอ (Monitor) ของคอมพิวเตอร์ส่วนบุคคล นอกจากข้อมูลที่เป็นรูปภาพแล้วส่วนของข้อมูลที่เป็นตัวอักษรนั้นเราสามารถที่จะกำหนดผ่านทางแป้นพิมพ์ (Keyboard) ได้เลย

โปรแกรมที่พัฒนาบนคอมพิวเตอร์ส่วนบุคคลได้ใช้โปรแกรมไมโครซอฟท์วิชวลเบสิก เวอร์ชัน 6 (Microsoft Visual Basic 6 , VB6)ในการพัฒนา ซึ่งโปรแกรมนี้เป็นเครื่องมือในการสร้างโปรแกรมบนระบบปฏิบัติการวินโดว์ (Windows) ที่ใช้งานง่าย คือเป็นโปรแกรมที่ใช้เทคนิคแบบวิชวลไลซ์(Visualize)ในรูปแบบของกราฟิก (Graphic User Interface) โดยการสร้างโปรแกรมบน VB6 นั้นจะเป็นการเลือกเครื่องมือต่าง ๆ ที่เหมาะสม มาออกแบบหน้าจอของโปรแกรมที่เราจะสร้าง ทำให้เราเห็นภาพโดยรวมว่าโปรแกรมของเราเป็นอย่างไรตั้งแต่ขั้นตอนออกแบบแล้ว ซึ่งเรียกการเขียนโปรแกรมลักษณะนี้ว่า วิชวลโปรแกรมมิ่ง (Visual Programming) การเขียนโปรแกรมลักษณะนี้เราไม่จำเป็นต้องเขียนคำสั่งต่าง ๆ มากนัก ก็สามารถสร้างโปรแกรมได้อย่างรวดเร็ว โดยอาศัยหลักการเขียนโปรแกรมแบบมีเหตุการณ์มากระตุ้น (Event Driven Programming) ซึ่งเป็นการเขียนโปรแกรมเพื่อกำหนดขั้นตอนการทำงานให้กับคอนโทรล (Control) ต่าง ๆ ที่ถูกสร้างขึ้น ตามเหตุการณ์ (Event) ต่าง ๆ ที่เกิดขึ้น เช่น การเลื่อนเมาส์ การคลิกเมาส์ เป็นต้น

นอกจากนี้โปรแกรมวิชวลเบสิกก็ยังได้พัฒนามาจากแนวความคิดแบบมุ่งที่วัตถุ (Object-Oriented Programming, OOP) ซึ่งแตกต่างจากการเขียนโปรแกรมในแบบเดิม ที่จะเป็นลักษณะของการมุ่งที่โปรแกรมย่อย (Procedural-Oriented) กล่าวคือจะพยายามแบ่งโปรแกรมที่มีขนาดใหญ่ให้เป็นโปรแกรมย่อยหลาย ๆ โปรแกรม (ตามแนวคิดของการออกแบบโปรแกรมจากบนลงล่าง (Top-Down Design)) แต่สำหรับแนวความคิดที่มุ่งวัตถุแล้ว แนวความคิดได้เปลี่ยนไปให้ความสนใจกับสิ่งต่าง ๆ ที่อยู่ในโปรแกรม ซึ่งเรียกว่าวัตถุ (Object) แทน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

วัตถุในโปรแกรมวิชวลเบสิกได้แก่ ส่วนของคอนโทรลต่าง ๆ ที่นำมาวางบนฟอร์ม ซึ่งมีคุณสมบัติบางอย่างเช่นเดียวกับวัตถุในแนวคิดแบบที่มุ่งวัตถุ กล่าวคือแต่ละวัตถุ (Object) จะประกอบไปด้วย

ข้อมูล (Data) เปรียบเสมือนข้อมูลของวัตถุ สำหรับในวิชวลเบสิก คำว่า "ข้อมูล" จะหมายถึง คุณสมบัติ (Property) เช่น ความยาว สี เป็นต้น

รหัสคำสั่ง (Code) เปรียบเสมือนกับพฤติกรรมของวัตถุ สำหรับในวิชวลเบสิก คำว่า "รหัสคำสั่ง" จะหมายถึง วิธีการ (Method) ประจำตัวของแต่ละวัตถุ เช่น เมธอดเคลื่อนที่ (Move) เมธอดเอนาเบิ้ล (Enable) เป็นต้น

ดังนั้นในการเขียนโปรแกรมแทนที่จะเขียนในเดิมที่เป็นลำดับขั้น ซึ่งเริ่มจากส่วนของโปรแกรมหลัก (Main Program) ที่ทำหน้าที่เรียกโปรแกรมย่อยต่าง ๆ ก็จะเปลี่ยนมาเป็นการเขียนโปรแกรมกับแต่ละวัตถุแทน โดยอาศัยคำสั่งและ วิธีการที่เกี่ยวข้องกับแต่ละวัตถุนั้น

จากคุณสมบัติ ๆ ที่กล่าวมาของโปรแกรมไมโครซอฟท์วิชวลเบสิก ก็เพียงพอต่อการพัฒนาโปรแกรมส่วนติดต่อกับผู้ใช้ (Display User Interface Program) ซึ่งมีหลักการออกแบบดังนี้

4.1 หลักการออกแบบโปรแกรม

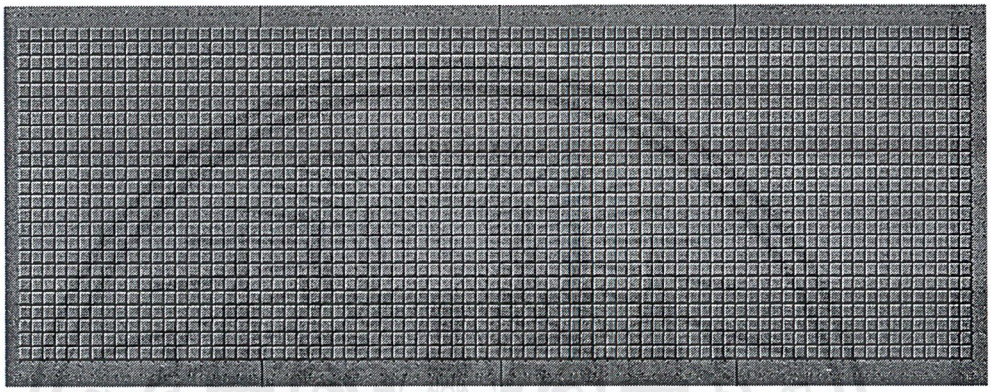
4.1.1 หลักการออกแบบโปรแกรมส่วนติดต่อกับผู้ใช้

จากสิ่งที่เราต้องการคือ การสามารถกำหนดตำแหน่ง และสีต่าง ๆ ของหลอดแอลอีดีที่อยู่บนบอร์ดแสดงผลโดยการกำหนดตำแหน่ง และสีของหลอดแอลอีดีจำลองบนหน้าจอคอมพิวเตอร์ ผู้จัดทำจึงได้ใช้คอนโทรลปุ่มคำสั่ง (Command Button) ซึ่งเป็นคอนโทรลชนิดหนึ่งของโปรแกรมวิชวลเบสิก สำหรับการจำลองหลอดแอลอีดีของบอร์ดแสดงผลไว้ที่หน้าจอคอมพิวเตอร์ และใช้เมาส์ (Mouse) ในการคลิ๊ก (Click) ที่ปุ่มคำสั่งนั้น โดยการคลิ๊กปุ่มคำสั่งจะทำให้เกิดเหตุการณ์ถูกคลิ๊ก (On Click) ก็จะเป็นเหตุการณ์ที่ทำให้สีพื้นของปุ่มคำสั่งนั้นเปลี่ยนเป็นสีที่เราได้เลือกไว้ เพื่อเลือกสีที่ต้องการสำหรับหลอดแอลอีดีนั้น ๆ เนื่องจากการที่มีหลอดแอลอีดีมาก ๆ (1920 หลอด) จึงทำให้การอ้างอิงคอนโทรลปุ่มคำสั่งนั้นยากลำบาก จึงต้องออกแบบปุ่มคำสั่งเป็นอาเรย์ (Array) เพื่อความสะดวกในการอ้างอิงในส่วนของโปรแกรม สำหรับการเลือกสีของหลอดแอลอีดีนั้น ก็ได้ใช้คอนโทรลเกี่ยวกับการเลือกคือ ออปชั่นบัตทอน (Option Button) ซึ่งสามารถเลือกได้เพียงอย่างเดียวจากหลาย ๆ อย่างภายในกรอบหรือเฟรม (Frame) เดียวกัน ซึ่งก็เหมาะในการเลือกสีของหลอดแอลอีดีซึ่งสามารถเลือกได้เพียงสีเดียวสำหรับแอลอีดีหลอดหนึ่ง ๆ

เมื่อเราได้เลือกสีสำหรับแอลอีดีทุก ๆ หลอดที่ต้องการแล้วเราจะต้องทำการแปลงข้อมูลของสี และตำแหน่งของหลอดแอลอีดีให้อยู่ในรูปแบบที่ไมโครคอนโทรลเลอร์เข้าใจ แล้วทำการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ส่งข้อมูลดังกล่าวให้กับไมโครคอนโทรลเลอร์เพื่อที่จะนำข้อมูลดังกล่าวออกแสดงผลที่บอร์ดแสดงผลต่อไป ซึ่งการส่งข้อมูลผ่านพอร์ทอนุกรม (Serial Port) ของคอมพิวเตอร์มาที่ไมโครคอนโทรลเลอร์สามารถทำได้โดยใช้คอนโทรล (MS Comm Control) ซึ่งสามารถกระตุ้นให้เกิดเหตุการณ์ส่งข้อมูลโดยการคลิ๊กเมาส์ที่คอนโทรลคอมมานบัตทอนที่สร้างมาโดยเฉพาะ



รูปที่ 4.1 บอร์ดแสดงผลที่จำลองบนหน้าจอคอมพิวเตอร์

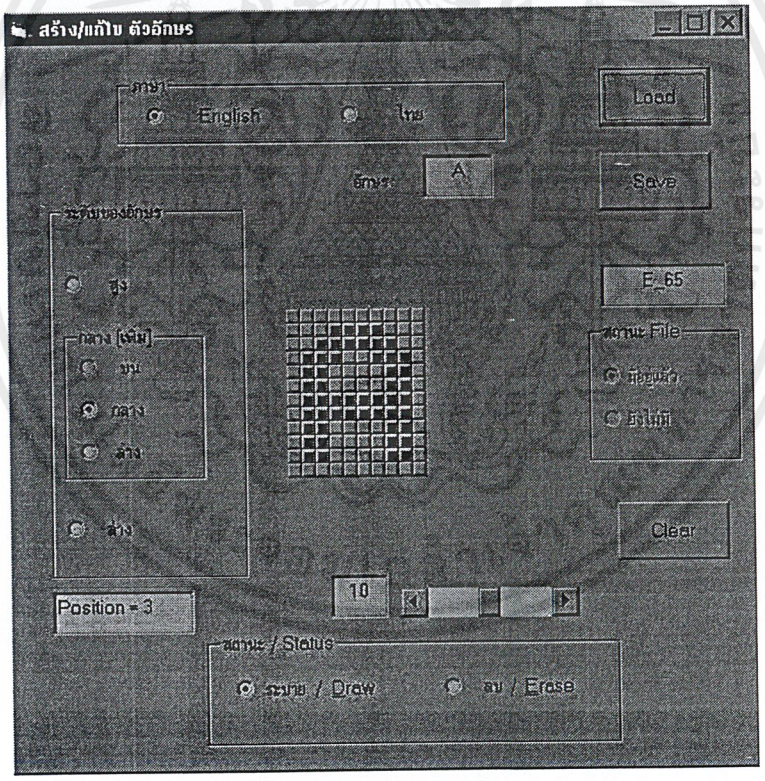
รูปที่ 4.2 ตัวเลือกสีของหลอดแอลอีดีจำลอง

สำหรับข้อมูลของตัวอักษรนั้นเราสามารถสร้างขึ้นใหม่หรือแก้ไขได้ด้วยโปรแกรม สร้าง/แก้ไขตัวอักษร(Font_Editor.EXE) ที่ผู้จัดทำได้สร้างขึ้นมาโดยเฉพาะซึ่งมีขั้นตอนการออกแบบดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.1.2 หลักการออกแบบโปรแกรม สร้าง/แก้ไขตัวอักษร

สำหรับการออกแบบเพื่อให้ได้ข้อมูลของตัวอักษรแต่ละตัว และให้ข้อมูลที่ได้มีรูปแบบที่สามารถเข้ากันได้กับส่วนของโปรแกรมติดต่อกับผู้ใช้ ดังนั้นการออกแบบจึงมีคอนโทรลปุ่มคำสั่งที่เพียงพอต่อการสร้างตัวอักษร 1 ตัว และเพื่อเป็นการประหยัดพื้นที่ในหน่วยความจำสำรองของคอมพิวเตอร์ จึงต้องสามารถที่จะปรับเปลี่ยนจำนวนของคอนโทรลปุ่มคำสั่งได้สะดวก และต้องบันทึกเป็นแฟ้มในรูปแบบที่สะดวกในการอ่านข้อมูลสำหรับโปรแกรมส่วนติดต่อกับผู้ใช้ด้วย รวมทั้งต้องเก็บข้อมูลที่จำเป็นอื่น ๆ เช่น เป็นข้อมูลของตัวอักษรใด ในภาษาใด ตำแหน่งที่จะแสดงตัวอักษร ของอักษรนั้น ๆ เพื่อที่โปรแกรมส่วนติดต่อกับผู้ใช้จะได้เข้าใจข้อมูลของตัวอักษรนั้นได้อย่างชัดเจนและนำข้อมูลของตัวอักษรนั้นไปใช้ได้ถูกต้อง



รูปที่ 4.3 โปรแกรม สร้าง/แก้ไขตัวอักษร

ในการเก็บข้อมูลของตัวอักษรนั้นจะใช้ชื่อแฟ้มที่เป็นค่ารหัสแอสกี (ASCII Code) ของตัวอักษรนั้นเป็นสำคัญ ดังนั้นจึงต้องตรวจสอบค่ารหัสแอสกีของตัวอักษรนั้น และเนื่องจากมีบางตัวอักษรที่ไม่สามารถพิมพ์อยู่โดด ๆ ได้ เช่น วรรณยุกต์ไทย และ สระไทยบางตัว จะทำให้โปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ไม่สามารถรู้ค่ารหัสแอสกีของตัวอักษรดังกล่าว จึงได้แก้ปัญหาโดย ใช้ค่ารหัสแอสกีของพยัญชนะของภาษาอังกฤษซึ่งสามารถพิมพ์โดด ๆ ได้ทุกตัวอักษร ตัวอย่างเช่น วรรณยุกต์ โท ไม่สามารถพิมพ์โดด ๆ ได้ก็ไม่สามารถหาชื่อแฟ้มที่จะเก็บได้ จึงอาศัยพยัญชนะของภาษาอังกฤษที่อยู่ตำแหน่งเดียวกับ วรรณยุกต์ โท ในแป้นพิมพ์ (Keyboard) คือ j แล้วทำการหาค่ารหัสของอักษร j แล้วทำการบันทึกเป็นชื่อแฟ้มว่า "T_<ค่ารหัสแอสกีของ j>" โดย "T" บ่งบอกว่าเป็นอักษรไทย และเพื่อไม่ให้ชื่อแฟ้มของอักษร j ซ้ำกับชื่อแฟ้มของวรรณยุกต์ โท จึงตั้งชื่อชื่อแฟ้มของอักษร j ว่า "E__<ค่ารหัสแอสกีของ j>" เท่านี้ก็จะทำให้สามารถบันทึกข้อมูลตัวอักษรทุกตัวได้โดยไม่ซ้ำซ้อนกัน

เหตุผลในการตั้งชื่อแฟ้มที่บันทึกที่รวมเอาค่ารหัสแอสกีด้วยก็เพราะตอนที่รับข้อมูลที่เป็นข้อความยาว ๆ เมื่อตรวจค่ารหัสแอสกีก็จะทำให้รู้ว่าเป็นอักษรตัวใด และจะได้อ่านเอาข้อมูลที่เก็บไว้ในแฟ้มที่ถูกต้อง โดยถ้าค่ารหัสแอสกี น้อยกว่าหรือเท่ากับ 127 ก็จะเป็นตัวอักษรภาษาอังกฤษ และชื่อแฟ้มที่จะทำการอ่านข้อมูลก็คือ "E_<ค่ารหัสแอสกีของตัวอักษรนั้น>" ส่วนเมื่อค่ารหัสแอสกี มากกว่า 127 ก็จะเป็นตัวอักษรภาษาไทยและชื่อแฟ้มที่จะทำการอ่านข้อมูลก็คือ "T_<XXX>" โดยค่า XXX ต้องนำไปเทียบตารางที่เขียนไว้ในโปรแกรมเพื่อที่จะสามารถอ่านเอาข้อมูลที่เก็บไว้ในแฟ้มที่ถูกต้อง

4.2 <u>รูปแบบของข้อมูลที่ได้จากโปรแกรมส่วนติดต่อกับผู้ใช้</u>

เพื่อที่จะทำให้ไมโครคอนโทรลเลอร์เข้าใจว่าข้อมูลที่รับมานั้นเป็นข้อมูลในโหมด (Mode) การทำงานโหมดใด และจำนวนข้อมูลที่ส่งมาเป็นจำนวนกี่เฟรม จึงจะสามารถแสดงภาพได้ถูกต้อง ดังนั้นจึงต้องใส่ข้อมูลส่วนหัว (Header) ที่บอกโหมดการทำงาน และจำนวนเฟรม รวมไปกับข้อมูลรูปภาพที่จะแสดงผลด้วยดังแสดงในรูปที่ 4.4

จำนวนเฟรม	โหมด 0	โหมด 1	สำรองไว้	สำรองไว้	ข้อมูล .....

รูปที่ 4.4 รูปแบบการส่งข้อมูลจากคอมพิวเตอร์ให้กับไมโครคอนโทรลเลอร์

โดยที่

จำนวนเฟรม จะบอกว่าข้อมูลที่ส่งตามมามีขนาดกี่เฟรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โหมด 0,1 จะบอกโหมดการทำงานแก่คอนโทรลเลอร์ว่าให้แสดงข้อมูลที่ส่งตามมาในโหมดการแสดงผลแบบใด

สำรองไว้ จะเป็นการสำรองเนื้อที่ไว้เพื่อใช้ในการอื่น

และเนื่องจากการที่กำหนดจำนวนเฟรมที่จะส่งออกไป เราจึงต้องตรวจดูว่าข้อมูลที่สร้างขึ้นมานั้นมีขนาดเต็มเฟรมหรือไม่ ถ้าไม่ก็จะต้องเติมข้อมูลที่บอกว่าเป็นหลอดแอลอีดีสีดำ(หรือดับ)ให้เป็นจำนวนเต็มเท่าของข้อมูล 1 เฟรม

4.3 <u>การส่งข้อมูลไปยังไมโครคอนโทรลเลอร์</u>

การส่งข้อมูลไปยังไมโครคอนโทรลเลอร์นั้น คอมพิวเตอร์จะส่งออกทางพอร์ตอนุกรมซึ่งในโปรแกรมวิชวลเบสิกมีคอนโทรลที่ช่วยในการติดต่อกับพอร์ตอนุกรมที่สำคัญ คือ MSComm (Microsoft Communication Control)

MSComm เป็นคอนโทรลที่ทำงานโดยมีการตอบสนองต่อเหตุการณ์แบบมีการกระตุ้น (Even-Driven) นั่นก็คือ คอนโทรลจะทำหน้าที่ตรวจสอบการเกิดขึ้นหรือการร้องขอให้เกิดเหตุการณ์ต่างๆกับพอร์ตอนุกรมโดยอัตโนมัติ และจะมีการแจ้งโปรแกรมผ่านทางโพรซีเยอร์เหตุการณ์ ดังนั้นเมื่อผู้ใช้จะสามารถทราบได้ว่ามีการส่งข้อมูลไปแล้วหรือไม่

4.4 <u>ข้อมูลของคอนโทรลที่สำคัญที่ใช้ในการออกแบบโปรแกรม</u>

- ฟอร์ม (Form)

เป็นคอนโทรลที่ใช้สำหรับกำหนดรูปร่างของโปรแกรมและให้คอนโทรลอื่นมาวาง

คุณสมบัติของฟอร์ม (Form Property)

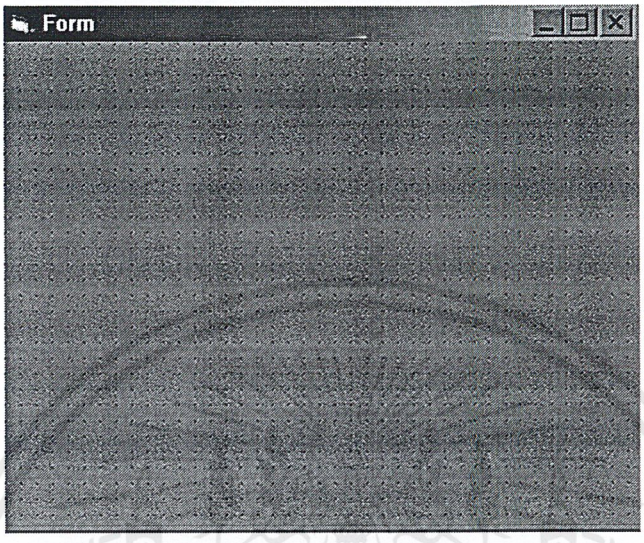
Caption	กำหนดข้อความที่แสดงบนไตเติ้ลบาร์ (Title Bar) ของฟอร์ม
Movable	กำหนดว่าฟอร์มสามารถเคลื่อนย้ายได้หรือไม่
Enable	กำหนดให้ฟอร์มสามารถตอบสนองต่ออีเวนต์ได้หรือไม่

เมดถอด และอีเวนต์ที่สำคัญของฟอร์ม

Load	เป็นอีเวนต์ที่เกิดขึ้นมาเมื่อเรียกฟอร์มขึ้นมาครั้งแรก
Active	เป็นอีเวนต์ที่เกิดขึ้นมาเมื่อฟอร์มนั้นเป็นฟอร์มที่เราทำงานในขณะนั้น
Deactive	เป็นอีเวนต์ที่เกิดขึ้นมาเมื่อฟอร์มอื่นเป็นฟอร์มแอคทีฟแทน
Show	เป็นเมดถอดที่ใช้แสดงฟอร์มขึ้นมา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Hide เป็นเมดถอดที่ใช้ซ่อนฟอร์มไว้



รูปที่ 4.5 ฟอร์ม (Form)

- คอนโทรลปุ่มคำสั่ง (Command Button)

ปุ่มคำสั่งเป็นคอนโทรลที่จะทำงานตามที่เรากำหนดเมื่อมีการกดปุ่มคำสั่ง โดยเราต้องกำหนดสิ่งที่จะตอบสนองในเหตุการณ์ถูกคลิ๊ก (Click)

นอกจากนี้คอนโทรลปุ่มคำสั่งยังมีคุณสมบัติ(Property) ที่ได้ใช้ประโยชน์ในโปรแกรมดังนี้

BackColor ใช้กำหนดสีพื้นของคอนโทรลปุ่มคำสั่ง

Caption ใช้กำหนดข้อความที่ปุ่มคำสั่งนั้น

Picture ใช้กำหนดรูปภาพบนปุ่มคำสั่งนั้น

Enable ใช้กำหนดให้ปุ่มคำสั่งใช้งานได้ หรือไม่ได้

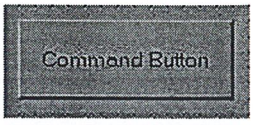
Visible ใช้กำหนดว่าให้สามารถมองเห็นหรือไม่ในขณะรันโปรแกรม

เมดถอด และอีเวนต์ที่สำคัญของปุ่มคำสั่ง

Click เป็นอีเวนต์ที่เกิดขึ้นเมื่อมีการมีการใช้เมาส์คลิ๊กที่ปุ่มคำสั่งนั้น

GotFocus เป็นอีเวนต์ที่เกิดขึ้นเมื่อโฟกัสเลื่อนมาอยู่ที่ปุ่มคำสั่งนั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.6 คอมมานด์บัตทอน (Command Button)

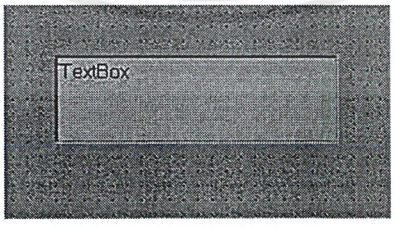
- คอนโทรลเท็กบอกซ์ (TextBox)

เป็นคอนโทรลที่ใช้เพื่อให้ผู้ใช้เติม แก้ไขข้อความโดยผ่านคีย์บอร์ดเข้าสู่โปรแกรม และแสดงข้อมูลต่าง ๆ จากโปรแกรมได้ โดยมีคุณสมบัติที่สำคัญดังนี้

Text	ใช้สำหรับรับข้อมูลจากคีย์บอร์ดและแสดงผล
Alignment	ใช้กำหนดรูปแบบที่แสดงในเท็กบอกซ์ เช่น จัดชิดซ้าย(Left Justify) จัดกึ่งกลาง(Center Justify)
MultiLine	กำหนดให้เท็กบอกซ์แสดงและรับข้อมูลแบบหลายบันทัดได้
ScrollBars	กำหนดให้เท็กบอกซ์แสดงสกอลบาร์ด้วยหรือไม่
MaxLength	กำหนดความยาวสูงสุดของข้อความที่สามารถรับได้ของเท็กบอกซ์เป็นเท่าไร
Locked	กำหนดให้ผู้ใช้แก้ไขข้อความในเท็กบอกซ์ได้หรือไม่

เมดถอด และอีเวนต์ที่สำคัญของเท็กบอกซ์

Change	เกิดขึ้นเมื่อข้อความไนเท็กบอกซ์มีการเปลี่ยนแปลง
GotFocus	เป็นอีเวนต์ที่เกิดขึ้นเมื่อโฟกัสเลื่อนมาอยู่ที่เท็กบอกซ์นั้น
LostFocus	เป็นอีเวนต์ที่เกิดขึ้นเมื่อโฟกัสเลื่อนไปอยู่ที่คอนโทรลตัวอื่นหลังจากที่ที่เท็กบอกซ์นั้นได้รับโฟกัสมาก่อน



รูปที่ 4.7 คอนโทรลเท็กบอกซ์ (TextBox)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- คอนโทรลออปชั่นบัตทอน (Option Button)

ใช้สำหรับเป็นตัวเลือกให้แก่ผู้ใช้ ถ้ามีหลายออปชั่นบัตทอนบนฟอร์ม ผู้ใช้จะเลือกออปชั่นบัตทอนได้ครั้งละ 1 ตัวเท่านั้น ซึ่งเมื่อเราเลือกออปชั่นบัตทอนหนึ่งจะทำให้ออปชั่นบัตทอนเดิมที่เราเลือกเปลี่ยนไปเป็นไม่ถูกเลือกโดยอัตโนมัติ โดยในการใช้งานจะใช้คอนโทรลเฟรมในการแบ่งกลุ่มของตัวเลือกต่าง ๆ โดยที่แต่ละกลุ่มผู้ใช้จะเลือกได้เพียงตัวเลือกเดียวเท่านั้น

คุณสมบัติที่สำคัญของออปชั่นบัตทอนคือ

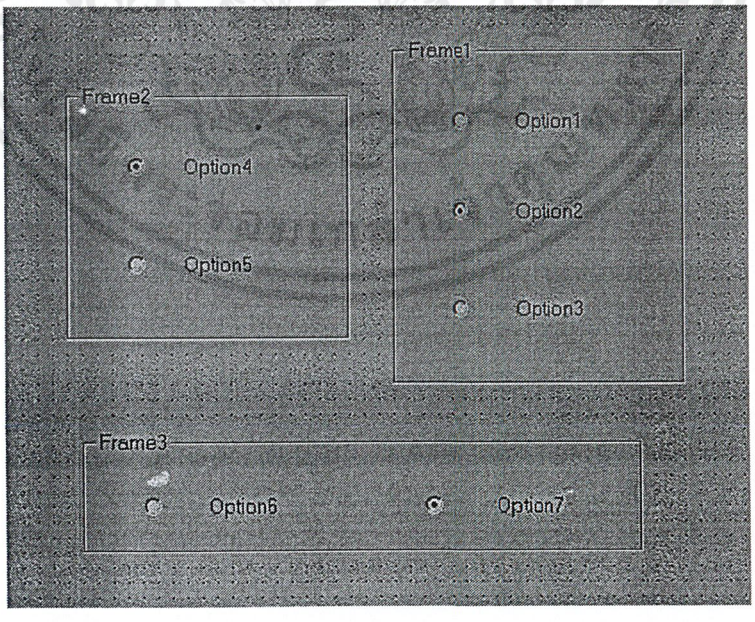
Value ใช้กำหนดสถานะของออปชั่นบัตทอนว่าถูกเลือกหรือไม่

เมดถอด และอีเวนต์ที่สำคัญของออปชั่นบัตทอน

Click เกิดขึ้นจากการคลิ๊กเมาส์ที่ออปชั่นบัตทอน หรือการใช้คีย์ลัด การกดปุ่ม<TAB> เลื่อนโฟกัสไปยังออปชั่นบัตทอนแล้วกดปุ่ม <SPACEBAR> หรือเกิดจากคำสั่งค่าให้กับคุณสมบัติ Value

- คอนโทรลเฟรม (Frame)

ใช้สำหรับจัดกลุ่มคอนโทรลที่ต้องใช้ร่วมกันเป็นกลุ่ม ๆ เช่นการแบ่งกลุ่มของออปชั่นบัตทอนเป็นกลุ่ม ๆ เป็นต้น ซึ่งโดยทั่วไปแล้วเฟรมเป็นคอนโทรลที่ช่วยเพิ่มความเรียบร้อย และความสวยงามของหน้าตาของโปรแกรมของเราให้มากขึ้น



รูปที่ 4.8 คอนโทรลออปชั่นบัตทอนและคอนโทรลเฟรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- คอนโทรล MSComm

คอนโทรลจะทำหน้าที่ตรวจสอบการเกิดขึ้นหรือการร้องขอให้เกิดเหตุการณ์ต่างๆกับพอร์ตอนุกรมโดยอัตโนมัติ และจะมีการแจ้งโปรแกรมผ่านทางโพรซีเยอร์เหตุการณ์ซึ่งสามารถตรวจสอบได้จากโปรแกรม

คุณสมบัติที่สำคัญของ MSComm

CommPort กำหนดหมายเลขของพอร์ตอนุกรม

Setting กำหนดพารามิเตอร์สำหรับการสื่อสาร เช่น อัตราการส่งข้อมูล , การตรวจสอบพาริตี้ , ความยาวของข้อมูล

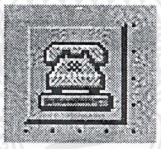
InBufferSize กำหนดขนาดของบัฟเฟอร์สำหรับด้านรับข้อมูล

OutBufferSize กำหนดขนาดของบัฟเฟอร์ด้านส่งออกข้อมูล

เมดถอด และอีเวนต์ที่สำคัญของออปชั่นบัตทอน

RTSEnable เมื่อมีการใช้สายขอร้องที่จะส่ง (Request To Send , RTS)

OnComm เกิดขึ้นเมื่อมีการเปลี่ยนแปลงคุณสมบัติเหตุการณ์ของคอนโทรลนี้



รูปที่ 4.9 คอนโทรล MSComm

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

การทดลองและผลการทดลอง

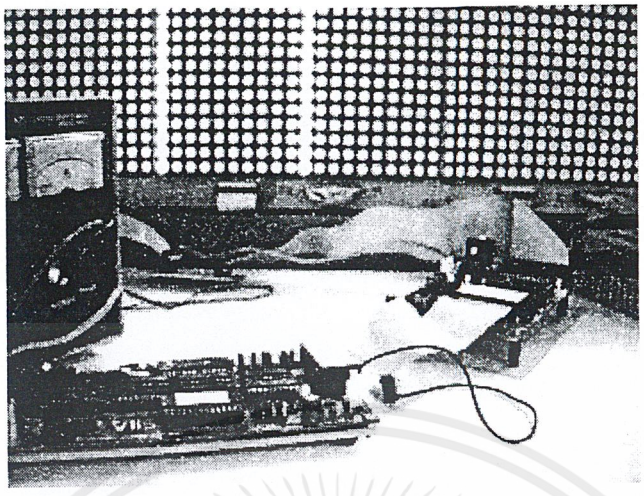
5.1 ขั้นตอนในการศึกษาและทดลอง

1. ศึกษาการทำงานของบอร์ดแสดงข้อมูลโดยการทดลองส่งข้อมูลออกทางพอร์ต 8255 ของไมโครคอนโทรลเลอร์ และตรวจสอบว่าการแสดงผลของบอร์ดแสดงข้อมูลตรงกับข้อมูลที่ส่งไปหรือไม่
2. ออกแบบโปรแกรมที่ใช้ในการแสดงรูปภาพ 1 เฟรมโดยการกำหนดข้อมูล (Define Byte) ขนาด 480 ไบต์ (1 เฟรม) เพื่อใช้แสดงภาพนิ่งและภาพเลื่อนซ้าย — ขวา
3. ศึกษาและทดลองการรับข้อมูลผ่านพอร์ตอนุกรมของไมโครคอนโทรลเลอร์โดยการส่งข้อมูลจากโปรแกรมวิชวลเบสิก
4. สร้างหน้าจอใช้งานบนคอมพิวเตอร์ส่วนบุคคลและทดลองสร้างข้อมูลจากภาพที่วาดบนหน้าจอจำลองบอร์ดแสดงข้อมูล หลังจากนั้นจึงส่งข้อมูลไปแสดงผลยังบอร์ดแสดงข้อมูล
5. ออกแบบและทดสอบโปรแกรมในการแสดงภาพเคลื่อนไหว และข้อความต่อเนื่อง
6. ทำการปรับปรุงโปรแกรมทุกส่วนเพื่อให้ได้การแสดงผลที่น่าพอใจ

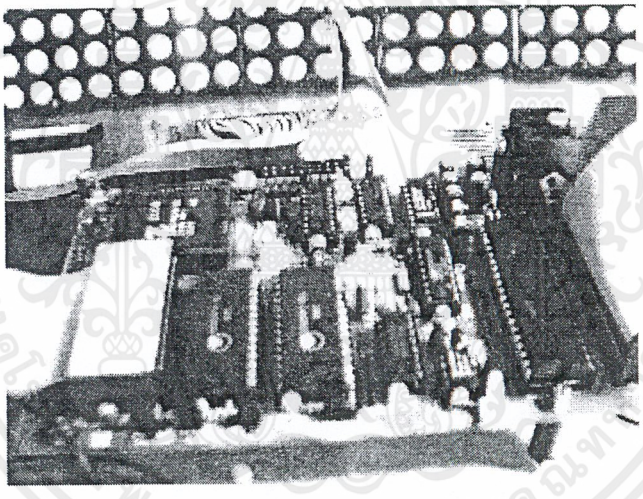
5.2 ผลการทดลอง

1. หลังจากการทดลองการรับส่งข้อมูลระหว่างคอมพิวเตอร์ส่วนบุคคลกับไมโครคอนโทรลเลอร์แล้วสามารถรับส่งข้อมูลได้ถูกต้องที่อัตราการรับส่ง 19200 บิตต่อวินาที
2. ในส่วนของการควบคุมการแสดงผลของภาพนิ่งและภาพเลื่อนซ้าย – ขวา สามารถแสดงใช้วิธีการในการแสดงข้อมูลขนาด 1 เฟรมแล้วได้การแสดงผลที่ดี
3. ภาพเลื่อนขึ้น – ลง และภาพเคลื่อนไหวต้องใช้การสร้างข้อมูลหลายเฟรมโดยโปรแกรมวิชวลเบสิกก่อนแล้วจึงส่งมาแสดงผลจึงจะได้การแสดงผลดี
4. การแสดงข้อความต่อเนื่องนั้นใช้การสร้างภาพหลายเฟรมที่มีความต่อเนื่องกันตามความยาวของข้อความที่ผู้ใช้พิมพ์ในกล่องข้อความซึ่งแสดงผลได้เป็นที่น่าพอใจ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5.1 แสดงการทดลองควบคุมการแสดงผลของบอร์ดแสดงข้อมูล



รูปที่ 5.2 แสดงไมโครคอนโทรลเลอร์ที่ใช้ควบคุมการแสดงผลของบอร์ดแสดงข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 6
สรุปผลและวิจารณ์

สรุปผล

โครงงานนี้ได้พัฒนาโปรแกรมในการควบคุมการทำงานของไมโครคอนโทรลเลอร์และโปรแกรมสำหรับผู้ใช้ติดต่อสั่งงานการแสดงข้อมูลผ่านทางคอมพิวเตอร์ส่วนบุคคล ทำให้ผู้ใช้สามารถควบคุมการแสดงข้อมูลของบอร์ดแสดงข้อมูลได้โดยสะดวกยิ่งขึ้น อีกทั้งยังสามารถเลือกรูปแบบในการแสดงข้อมูลได้ทั้ง

1. ภาพนิ่ง ซึ่งเป็นภาพหรือข้อความขนาด 1 เฟรม
2. ภาพเลื่อน สามารถเลือกที่ทางที่ต้องการให้รูปภาพเลื่อนไปได้ 4 ทิศทางคือ เลื่อนขึ้น ลง เลื่อนไปทางซ้าย และ ขวาเป็นภาพขนาด 1 เฟรม
3. ภาพเคลื่อนไหวและข้อความต่อเนื่อง เป็นการแสดงภาพหรือข้อความจำนวนหลายเฟรม ซึ่งแต่ละเฟรมมีความต่อเนื่องกัน

โดยที่ข้อมูลที่ผู้ใช้สร้างขึ้นสามารถบันทึกไว้ในรูปของแฟ้มอักษร (Text File) เพื่อนำมาแสดงผลในการใช้งานครั้งอื่นๆได้อีก หลังจากการสั่งงานในการแสดงผลเสร็จแล้วผู้ใช้ยังสามารถใช้คอมพิวเตอร์ส่วนบุคคลในการทำงานอื่นต่อไปได้โดยที่ยังสามารถกลับมาควบคุมการแสดงข้อมูลของบอร์ดแสดงข้อมูลได้ตลอดเวลา

ปัญหาและแนวทางในทางแก้ไข

แม้ว่าโปรแกรมในการควบคุมการแสดงผลผ่านทางคอมพิวเตอร์ส่วนบุคคลจะมีรูปแบบในการสั่งงานมาก แต่ก็ยังมีปัญหาอยู่ เนื่องจากขนาดของโปรแกรมที่ออกแบบมีขนาดใหญ่และต้องใช้ทรัพยากรในการทำงานมาก โดยเฉพาะการจำลองบอร์ดแสดงผลโดยใช้ตัวคอนโทรลคอมมานด์บัตทอนแทนหลอดแอลอีดีจำนวนมาก ทำให้เกิดความล่าช้าในการทำงานเมื่อใช้เป็นเวลานาน ดังนั้นแนวทางในการแก้ไขจึงควรศึกษาหาวิธีในการสร้างตัวคอนโทรลที่มีประสิทธิภาพ คือ สามารถแสดงคุณสมบัติของหลอดแอลอีดีได้ครบถ้วนและใช้ทรัพยากรในการทำงานน้อย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หนังสืออ้างอิง

1. ฉันทวุฒิ พืชผล , " คู่มือเรียน Visual Basic 6 " ,โปรวิชั่น , 438 หน้า , 2542
2. สุทธิศักดิ์ พงศ์ธนาพานิช , " Visual Basic 5.0 Professional การใช้คำสั่งและการคอนโทรล Active X " , ซีเอ็ดยูเคชั่น , 1144 หน้า , 2541
3. ยืน ภู่สุวรรณ , " ทฤษฎีและการประยุกต์ไมโครโปรเซสเซอร์ Z – 80 " , ซีเอ็ดยูเคชั่น , 292 หน้า , 2536
4. " Z80180 MICROPROCESSOR DATA BOOK TECHNICAL MANUAL" , Zilog , 120 p. , 1988

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ก.

รายละเอียดรีจิสเตอร์ของเอเอสซีไอ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การติดต่อรับ – ส่งข้อมูลผ่านทางเอเอสซีไอของ Z80180 มีรีจิสเตอร์สำคัญที่นำมาใช้ในกระบวนการรับส่งข้อมูลดังต่อไปนี้

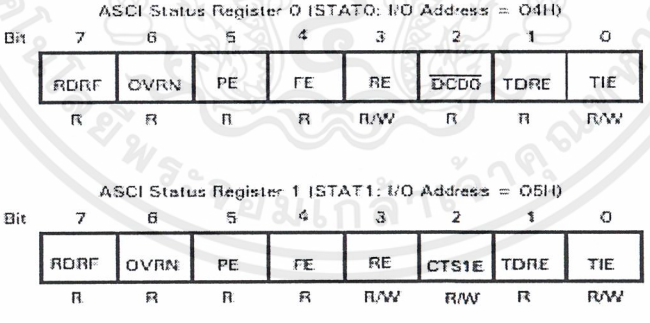
TSR0,1 เป็นรีจิสเตอร์ที่รับข้อมูลจากรีจิสเตอร์ส่งข้อมูล (Transmit Data Register , TDR) แล้วนำข้อมูลนั้นส่งออกที่ขา TXA

TDR0,1 เป็นรีจิสเตอร์ที่ใช้ส่งข้อมูลออกไปที่ขา TXA โดยการนำข้อมูลใน TDR ส่งไปที่ TSR เมื่อTSR ว่างลงและสามารถที่จะเขียนข้อมูลเข้าไปได้อีกในขณะที่ TSR กำลังเลื่อนข้อมูลออกไปที่ขา TXA

RSR0,1 เป็นรีจิสเตอร์ที่รับข้อมูลจากขา RXA เมื่อรับเต็มบัพเฟอร์ (BUFFER) แล้วก็เลื่อนไปที่ RDR ถ้า RSR ไม่ว่าง เมื่อมีการรับข้อมูลไบต์(Byte) ต่อไปเข้าไปอีกจะเกิดการทับซ้อนกันขึ้น ทำให้เกิดการผิดพลาดซึ่งจะมีการแสดงผลของการผิดพลาดที่รีจิสเตอร์สถานะ

RDR0,1(I/O Address 08H,09H) คือรีจิสเตอร์ที่ใช้เก็บข้อมูลที่รับมาจากขา RXA และในขณะที่กำลังบรรจุข้อมูลที่รับมาจาก RSR ข้อมูลไบต์ถัดไปสามารถรับเข้ามาต่อได้

STAT0,1(I/O Address 04H,05H) แต่ละช่อง(Channel) จะมีรีจิสเตอร์ใช้สำหรับตรวจสอบการสื่อสารเกี่ยวกับการผิดพลาดและสถานะสัญญาณ การอนุญาต(Enable)และไม่อนุญาต(Disable) เอเอสซีไอดังรูปที่ ผ-1



รูปที่ ผ-1 แสดงบิตต่างๆของรีจิสเตอร์สถานะของเอเอสซีไอ

RDRF : Receive Data Register Full

จะถูกเซ็ต (Set) เป็น 1 เมื่อข้อมูลที่รับเข้ามาถูกส่งเข้ามาที่ RDR เรียบร้อยแล้ว(ครบ1 ไบต์) แต่ถ้า การรับเกิดข้อผิดพลาดขึ้น RDRF ก็จะถูกเซ็ตค้างและข้อมูลที่ผิดนั้นก็จะถูกส่งมาที่ RDR และคงอยู่ ดังนั้นจะต้องทำการเคลียร์แฟลกแสดงการผิดพลาด

OVRN : Overrun Error

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จะเป็น 1 เมื่อ RDR เต็มและ RSR เต็มแล้วยังมีการรับข้อมูลเข้ามาอีก

PE : Parity Error

จะเป็น 1 เมื่อข้อมูลที่รับเข้ามาพาริตี้(Parity) ผิด

FE : Framing Error

จะเป็น 1 เมื่อข้อมูลที่รับเข้ามารูปแบบผิดไปจากรูปแบบที่กำหนด

RIE : Receive Interrupt Enable

เมื่อเป็น 1 จะอนุญาตให้เอเอสซีไอ ทำการขออินเทอร์รัพท์ได้เมื่อ RDRF,OVRN,PE หรือ FE ถูกเซ็ตเป็น 1 ด้วย เมื่อนั้นเอเอสซีไอก็จะให้สัญญาณอินเทอร์รัพท์

DCD0 : Data Carrier Detect Bit

จะเป็น 1 เมื่อเมื่อขาอินพุท DCD0 เป็น 1 และจะถูกเคลียร์เป็น 0 จากการอ่าน STAT0 ครั้งแรกนั้นขา INPUT DCD0 จะถูกเปลี่ยนจาก 1 เป็น 0 และระหว่างรีเซต(Reset) เมื่อ DCD0 เป็น 1 ส่วนของภาครับจะไม่ทำงาน

CTSIE : Channel 1 CTS Enable

ที่เชนแนล 1 มีขาอินพุท CTS1 ภายนอกซึ่งมัลติเพลกซ์ (Multiplex) กับ RXS เมื่อเซ็ตบิตนี้เป็น 1 จะถูกเลือกเป็นขา CTS1

TDRE : Transmit Data Register Empty

เป็นตัวบอกว่าข้อมูลพร้อมที่จะส่งได้หรือไม่ถ้าเป็น 1 คือพร้อมที่จะส่งข้อมูลแล้วให้เขียนข้อมูลเข้าไปที่ TDR ได้และเมื่อมีการเขียนข้อมูลเข้าไปที่ TDR ก็จะทำให้ TDRE เป็น 0 และข้อมูลใน TDR ก็จะถูกส่งให้ TSR จน TDR ว่างลงอีก TDRE ก็จะกลับเป็น 1 อีกครั้ง

TIE : Transmit Interrupt Enable

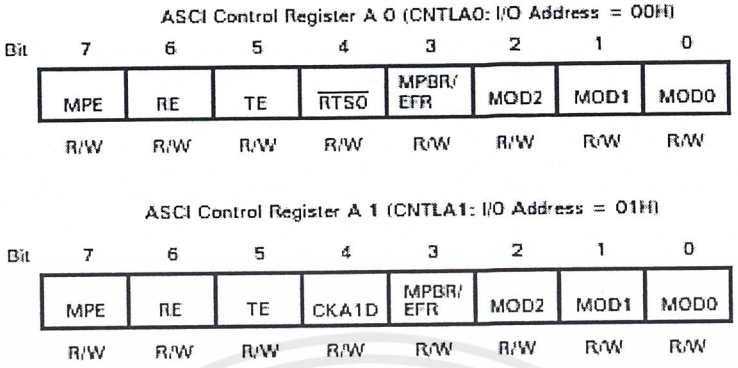
เมื่อเป็น 1 จะอนุญาตให้เอเอสซีไอใช้การส่งแบบอินเทอร์รัพท์ได้ โดยที่ TDRE ต้องเป็น 1 ด้วย TIE จะถูกเคลียร์เป็น 0 ระหว่างรีเซต

สำหรับการเซ็ตโหมดการทำงานต่างๆของเอเอสซีไอจะใช้ รีจิสเตอร์ CNTLA ซึ่งมีบิตสำหรับควบคุมการทำงานต่างๆดังรูปที่ ผ-2

MPE : Multiprocessor Mode Enable

ใช้อนุญาตการใช้ไมโครโปรเซสเซอร์ร่วมจากเมื่อมีการเลือกโหมดการสื่อสารแล้ว(MP = 1 ใน CNTLB) ในการสื่อสารแบบนี้รูปแบบของการสื่อสารจะมีบิตพิเศษเพิ่มเข้ามาเรียกว่าบิต MPE ซึ่งบิตนี้จะใช้ในการตรวจสอบหรือใช้งานเมื่อ MPE เป็น 1 และภาครับของไมโครโปรเซสเซอร์ร่วม (Multiprocessor) จะทำงาน คือ RDRF และแฟลกแสดงข้อมูลผิดพลาดจะทำงาน แต่ถ้า MPB = 0 เอเอสซีไอจะไม่สนใจข้อมูลไบต์นั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ ผ-2 แสดงบิตต่างๆของรีจิสเตอร์ควบคุมของเอเอสซีไอ เอ

RE : Receive Enable

ถ้าเป็น 1 จะ อนุญาตการรับของ เอเอสซีไอ แต่ถ้าเป็น 0 จะไม่อนุญาตให้มีการรับข้อมูล แต่ RDRF จะ FE ไม่ถูกรีเซตตาม

TE : Transmit Enable

เป็น 1 จะอนุญาตการส่ง ถ้าเป็น 0 จะไม่อนุญาตการส่ง แต่แฟลก TDRE จะถูกรีเซตตาม

RTS0 : Request To Send Channel 0

เป็นบิตที่ให้ผลเช่นเดียวกันกับขาเอาท์พุท RTS0 คือถ้าบิตนี้เป็น 1 ขาเอาท์พุท RTS0 ก็จะเป็น 1 ถ้าบิตนี้เป็น 0 ขาเอาท์พุท RTS0 ก็จะเป็น 0

CKA1D : CKA1 Clock Disable

ขา CKA1 จะมัลติเพลกซ์กับ TEND0 เมื่อบิตนี้เป็น 1 จะเลือกเป็นขา TEND0 แต่ถ้าเป็น 0 ก็จะเป็นสัญญาณนาฬิกา(Clock) ของเอเอสซีไอเชนแนล 1 บิตนี้เป็น 0 ระหว่างการรีเซต

MPBR/EFR : Multiprocessor Bit Receive/Error Flag

เมื่อโหมดหน่วยควบคุมหลายหน่วยถูกเลือก(MP ใน CNTLB = 1) เมื่ออ่านบิตนี้จะเป็นค่าในบิตของ MPB แต่ถ้าเขียน 0 ให้บิตนี้จะเป็นการรีเซตแฟลกแสดงความผิดพลาดในการรับ

MOD0,1,2

ใช้สำหรับเลือกรูปแบบของข้อมูลที่ใช้ในการรับส่ง ตารางที่ ผ-1

นอกจากนี้ยังมีรีจิสเตอร์ที่สำคัญอีกคือ รีจิสเตอร์ควบคุมเอเอสซีไอ B ซึ่งมีรายละเอียดบิตต่างๆดังรูปที่ ผ-3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

MOD2	MOD1	MOD0	DATA FORMAT
0	0	0	START +7 BIT DATA+1 STOP
0	0	1	START+7 BIT DATA+2 STOP
0	1	0	START+7 BIT DATA+PARITY+1 STOP
0	1	1	START+7 BIT DATA+PARITY+2 STOP
1	0	0	START+8 BIT DATA+1 STOP
1	0	1	START+8 BIT DATA+2 STOP
1	1	0	START+8 BIT DATA+PARITY+1 STOP
1	1	1	START+8 BIT DATA+PARITY+2 STOP

ตารางที่ ผ-1 แสดงรูปแบบข้อมูลจากการเซ็ต MOD0,1,2

ASCI Control Register B 0 (CNTLB0: I/O Address = 02H)
ASCI Control Register B 1 (CNTLB1: I/O Address = 03H)

Bit	7	6	5	4	3	2	1	0
	MPBT	MP	$\overline{\text{CTS}}$/PS	PEO	DR	SS2	SS1	SS0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

รูปที่ ผ-3 แสดงบิตต่างๆของรีจิสเตอร์ควบคุมของเอเอสซีอี บี

MPBT : Multiprocessor Bit Transmit

ใช้ส่งบิต MPB โดยถ้า MPBT = 1 เมื่อนั้นบิต MPB =1 และ MPBT = 0 บิต MPB = 0

MP : Multiprocessor Mode

ถ้าเป็น 1 จะเป็นการเซ็ตการติดต่อแบบไมโครโปรเซสเซอร์ร่วม โดยใช้รูปแบบของ MOD2 กับ MOD0 ดังนี้ START+7 OR 8 BIT DATA +1 OR 2 BIT STOP

CTS/PS : Clear To Send/Prescale

เมื่ออ่านบิตนี้จะใช้แสดงสถานะของขาอินพุท CTS ภายนอก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

PEO : Parity Even Odd Bit

ใช้เลือกว่าเมื่อมีการอนุญาตการตรวจสอบพาริตี้ใน MOD 1 จะใช้พาริตี้คู่หรือคี่ ถ้า PEO = 0 คือคู่ แต่ถ้า = 1 คือ คี่

DR : Divide Ratio

ใช้กำหนดบอร์ดเรท (Baud Rate) บิตนี้เป็น 0 เมื่อรีเซต

SS2,1,0 : Source/Speed Select 2,1,0

ใช้กำหนดสัญญาณนาฬิกาว่าจะให้เป็นภายในหรือภายนอกและกำหนดบอร์ดเรทด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ข

การใช้งานโปรแกรมควบคุมบอร์ดแสดงข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การใช้งานโปรแกรมควบคุมบอร์ดแสดงข้อมูล

จากหน้าจอแรกของโปรแกรมจะเป็นการเลือกว่าจะใช้โปรแกรมใด ระหว่าง โปรแกรมสร้างภาพและส่งข้อมูลออกแสดงผล และโปรแกรมสร้างและแก้ไขตัวอักษร

โปรแกรม สร้างภาพและส่งข้อมูลออกแสดงผล

การสร้างภาพ 1 เฟรม

อันดับแรกจะต้องเลือกรูปแบบการสร้างภาพเป็นแบบ Picture จากนั้นจึงใช้เมาส์เลือกสีที่ต้องการจาก ปุ่มเลือกสี แล้วใช้เมาส์คลิ๊กเลือกที่จุดแสดงผลที่ต้องการบนหน้าจอจำลองของบอร์ดแสดงผล เมื่อสร้างภาพเสร็จแล้วจึงกดที่ปุ่ม Generate เพื่อทำการสร้างข้อมูลของภาพที่ทำการสร้างขึ้นมา จากนั้นเมื่อต้องการส่งข้อมูลออกไปแสดงผลยังบอร์ดแสดงผลก็กดที่ปุ่มSend ข้อมูลภาพ ที่สร้างขึ้นมาก็จะถูกส่งออกไปแสดงผลที่บอร์ดแสดงผล

เมื่อผู้ใช้สร้างภาพขนาด 1 เฟรมเสร็จแล้วสามารถที่จะบันทึกข้อมูลภาพไว้ได้โดยกดปุ่ม Save แล้ว เลือกชื่อแฟ้มที่ต้องการบันทึก โปรแกรมก็จะทำการเก็บบันทึกข้อมูลของรูปภาพนั้นไว้ และถ้าต้องการเรียกเอารูปที่บันทึกไว้มาทำการแก้ไขก็กดที่ปุ่ม Load แล้วเลือกชื่อแฟ้มที่ต้องการ จากนั้นโปรแกรมก็จะเรียกเอาข้อมูลของภาพนั้นออกมาแสดงที่หน้าจอจำลองของบอร์ดผล และ ถ้าต้องการส่งข้อมูลออกไปแสดงผลยังบอร์ดแสดงผลก็ กดที่ปุ่ม Send

การสร้างภาพเคลื่อนไหว

อันดับแรก จะต้องเลือกรูปแบบในการสร้างภาพเป็น Motion ซึ่งจะมีขั้นตอนในการสร้างเช่นเดียวกับการสร้างภาพนิ่ง แต่ถ้าเราบันทึกข้อมูลเฟรมปัจจุบันในชื่อที่ซ้ำกับแฟ้มที่มีอยู่แล้วจะเป็นการสร้างภาพต่อเนื่องโดยสามารถที่จะบันทึกเป็นเฟรมแรกเฟรมสุดท้ายหรือจะแทรกที่ตำแหน่งไหนก็ได้โดยสามารถที่จะสร้างภาพเคลื่อนไหวได้อย่างมากที่สุด 85 เฟรม

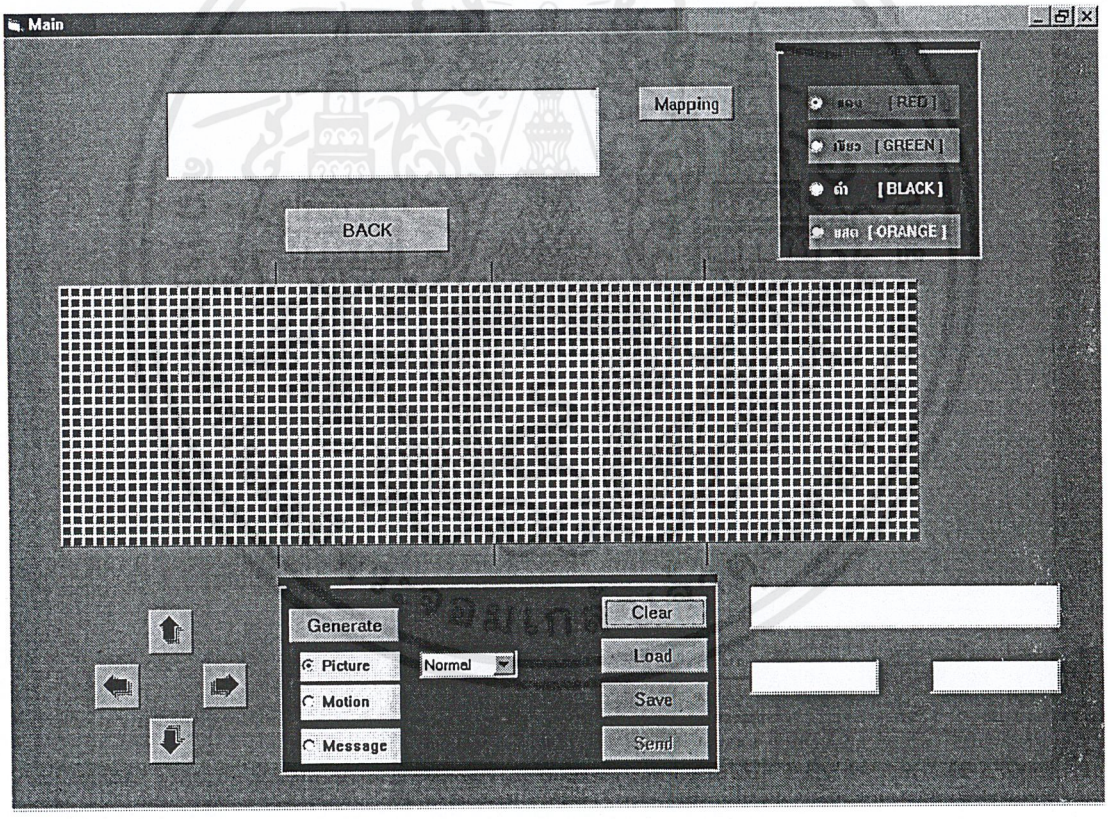
เมื่อผู้ใช้ต้องการที่จะนำเอาภาพเคลื่อนไหวนั้นออกไปแสดงที่บอร์ดแสดงข้อมูล ก็จะต้องทำการเรียกเอาแฟ้มข้อมูลที่บันทึกไว้โดยกดปุ่ม Load จากนั้นก็เลือกชื่อแฟ้มที่ต้องการ โปรแกรมก็จะทำการเรียกเอาข้อมูลของภาพเคลื่อนไหวนั้นมา และจะมีสกอลบาร์ (Scroll Bar) ใช้สำหรับเลื่อนดูเฟรมได้ทั้งหมด จากนั้นถ้าต้องการส่งข้อมูลออกไปแสดงผลยังบอร์ดแสดงผลก็ กดที่ปุ่ม Send เพื่อทำการส่งข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การสร้างข้อความต่อเนื่อง

สามารถทำได้โดยการพิมพ์ข้อความลงไปในกล่องข้อความ แล้วกดปุ่ม Mapping เพื่อสร้างข้อมูลสำหรับส่งไปแสดงผล และจะมีสกอลบาร์ที่สามารถเลื่อนดูภาพของข้อความที่พิมพ์เข้าไปทั้งหมดได้ จากนั้นก็สามารถส่งข้อมูลไปได้โดยกดปุ่ม Send ข้อความที่จะถูกแสดงบนบอร์ดแสดงข้อมูล

ผู้ใช้สามารถพิมพ์ข้อความได้ทั้งภาษาไทยและภาษาอังกฤษโดยสามารถเลือกสีของตัวอักษรที่จะแสดงได้ ซึ่งจะต้องเลือกที่ปุ่มเลือกสีก่อนที่จะทำการกดปุ่ม Mapping



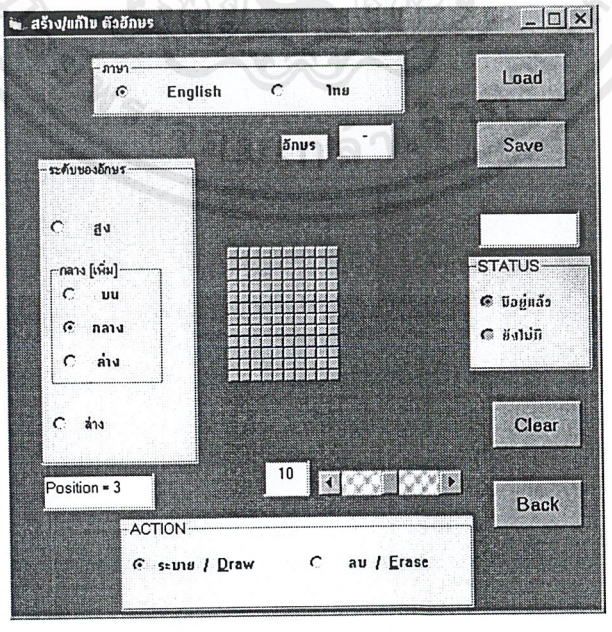
รูปที่ ผ-4 แสดงหน้าจอของโปรแกรมสร้างภาพและส่งข้อมูลออกแสดงผล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรมสร้างและแก้ไขตัวอักษร

จะเป็นโปรแกรมที่ใช้สำหรับสร้างตัวอักษรขึ้นมาใหม่ และแก้รูปแบบของตัวอักษรที่ต้องการจะแก้ โดยการใช้ขั้นแรกจะต้องเลือกระดับของตัวอักษรให้ถูกต้องและเหมาะสม เช่น อักษร A จะต้องเลือกระดับกลาง , วรรณยุกต์ไทยจะต้องเลือกระดับ สูง , สระ อุ สระ อู จะต้องเลือกระดับล่าง , อักษร ป จะต้องเลือกระดับ กลาง+สูง , อักษร ฤ จะต้องเลือกระดับ กลาง+ล่าง เป็นต้น ต่อไปก็ต้องทำการเลือกภาษา ถ้าเลือกภาษาอังกฤษแล้วพิมพ์ตัวอักษรภาษาอังกฤษในช่องตัวอักษร ตัวอักษรที่พิมพ์เข้าไปนั้นก็คือตัวอักษรที่จะทำการสร้างหรือแก้ไข แต่ถ้าเลือกภาษาไทยแล้วตัวอักษรที่พิมพ์จะต้องเป็นตัวอักษรภาษาอังกฤษที่ตรงกับตัวอักษรภาษาไทยในแป้นพิมพ์ที่ต้องการสร้างหรือแก้ไข เช่น ต้องการสร้างหรือแก้ไขตัวอักษร ก ก็จะต้องเลือกภาษาไทยและดูว่าตัวอักษร ก ในแป้นพิมพ์อยู่ตรงกันกับตัวอักษรภาษาอังกฤษตัวใดในแป้นพิมพ์ ซึ่งก็คือ ตัวอักษร d นั่นเอง และตัวอย่างอื่น ๆ เช่น ธ จะตรงกับอักษร T , วรรณยุกต์ โท จะตรงกับอักษร h เป็นต้น เมื่อเลือกค่าต่าง ๆ แล้วจึงทำการใช้เมาส์คลิ๊กในหน้าจอสำหรับสร้างและแก้ไขตัวอักษรเพื่อทำให้เกิดภาพของตัวอักษรที่ได้ทำการเลือกไว้

เมื่อทำการสร้างหรือแก้ไขตัวอักษรเสร็จแล้วจึงทำการบันทึกโดยกดที่ปุ่ม Save และสามารถเรียกเอาข้อมูลของตัวอักษรที่ได้ทำการสร้างแล้วมาทำการแก้ไขได้โดยการกดปุ่ม Load เมื่อผู้ใช้ได้ทำการแก้ไขตัวอักษรแล้ว ตัวอักษรนั้นก็จะปรากฏในภาพของข้อความต่อเนื่องเป็นแบบเดียวกับที่ได้ทำการแก้ไขไป



รูปที่ ผ-5 แสดงหน้าจอของโปรแกรมสร้างและแก้ไขตัวอักษร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ค
โปรแกรมในไมโครคอนโทรลเลอร์สำหรับควบคุมบอร์ดแสดงข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรมในไมโครคอนโทรลเลอร์สำหรับควบคุมบอร์ดแสดงข้อมูล

```
                ORG     1400H
DB   0F0H,0F1H,0F2H,0F3H,0F4H,0F5H,0F6H,0F7H,0F8H,0F9H
DB   00FH,01FH,02FH,03FH,04FH,05FH,06FH,07FH,08FH,09FH
                ORG     0000H
        PORTA   EQU     0080H
        PORTB   EQU     0081H
        PORTC   EQU     0082H
        PORTCON EQU     0083H
        DATA0   EQU     4005H
        DATA1   EQU     407DH
        DATA2   EQU     40F5H
        DATA3   EQU     416DH
        DCNTL   EQU     0032H
        RCR     EQU     0036H
        CBAR    EQU     003AH
        CBR     EQU     0038H
        BBR     EQU     0039H
        OMCR    EQU     003EH
        ICR     EQU     003FH
        STAT1   EQU     0005H   ;FOR CHECK STATUS
        CNTLA1  EQU     0001H   ;FOR SET MODE
        CNTLB1  EQU     0003H   ;FOR SET BAUD RATE
        RDR1    EQU     0009H   ;RECEIVE DATA REGISTER
        INDEX   EQU     1400H
        BUFFER  EQU     41E5H
        BUFFER1 EQU     41E6H
        SIZE    EQU     4000H
        MODE    EQU     4001H     ;MOTION OR NORMAL
        WAY     EQU     4002H     ;DIRECTION OF NORMAL
        FRAME   EQU     4003H     ;MOVE IX FOR MOTION
        COUNT   EQU     4004H

            LD          A,0C4H          ;BANK   = 4000H  LOGICAL
            OUT0        (CBAR),A        ;COMMON1 = C000H  LOGICAL
            LD          A,0CH
            OUT0        (BBR),A         ;BANK   = 10000H PHYSICAL
            LD          A,0EH
            OUT0        (CBR),A         ;COMMON1 = 18000H PHYSICAL

INIT:       LD          A,80H           ;INITIAL PORT
            OUT0        (PORTCON),A
            JP          HEAD
;********** SET REGISTER **************
            ORG   0100H
HEAD:       LD          A,08H
            OUT0        (STAT1),A       ;INTERRUPT ENABLE = 00001000
            LD          A,01H
            OUT0        (CNTLB1),A      ;BAUD RATE = 00000001 =19200
            LD          A,64H
            OUT0        (CNTLA1),A      ;SET MODE = 01100100
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                LD          A,40H
                OUT0        (OMCR),A
                IN0         A,(DCNTL)
                OR          70H
                OUT0        (DCNTL),A
                JP          HEAD1

;********** RECEIVE DATA (5 + 240*SIZE) BYTE***********

HEAD1:          LD          A,0FFH
                OUT0        (PORTA),A
                LD          A,1EH
                OUT0        (PORTB),A
                LD          A,0FFH
                OUT0        (PORTB),A
                LD          HL,4000H

                LD          E,05H       ;5 BYTE
LOP1:           IN0         A,(STAT1)     ;READ RECIEVE STATUS
                BIT         7,A
                JP          Z,LOP1
                IN0         A,(RDR1)      ;SEND DATA TO A
                LD          (HL),A      ;GET 1 BYTE

                IN0         A,(STAT1)
CHECK0:         BIT         6,A
                JP          NZ,CLEAR1
CHECK1:         BIT         5,A
                JP          NZ,CLEAR1
CHECK2:         BIT         4,A
                JP          NZ,CLEAR1
                JP          CONVERT1
CLEAR1:         IN0         A,(CNTLA1)
                AND         0F7H
                OUT0        (CNTLA1),A
                JP          LOP1

CONVERT1:       LD          A,(HL)
                INC         HL
                DEC         E           ;CHECK 5 BYTE
                JP          NZ,LOP1

                LD          A,(SIZE)    ;SIZE 02H = 1 FRAME
                LD          C,A
LOP2:           LD          E,0F0H      ;240 BYTE
LOP3:           IN0         A,(STAT1)     ;READ RECIEVE STATUS
                BIT         7,A
                JP          Z,LOP3
                IN0         A,(RDR1)      ;SEND DATA TO A
                LD          (HL),A

                IN0         A,(STAT1)
CHECK3:         BIT         6,A
                JP          NZ,CLEAR
CHECK4:         BIT         5,A
                JP          NZ,CLEAR
CHECK5:         BIT         4,A
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                JP          NZ,CLEAR
                JP          CONVERT
CLEAR:          IN0         A,(CNTLA1)
                AND         0F7H
                OUT0        (CNTLA1),A
                JP          LOP3

CONVERT:        LD          A,(HL)
                INC         HL
                DEC         E
                JP          NZ,LOP3         ;CHECK 240 BYTE
                DEC         C
                JP          NZ,LOP2         ;CHECK LOOP

                LD          A,(MODE)        ;CHECK MOTION OR NORMAL
                DEC         A
                JP          Z,MAIN1         ;MODE = 01 = NORMAL
                LD          A,(WAY)         ;MODE = 02 = MOTION OR TEXT
                DEC         A
                JP          Z,MAIN2         ;WAY = 01 = MOTION
                JP          MAIN3_1         ;WAY = 02 = TEXT

;***** MAIN1 = NORMAL 5 SUB MODE ******

MAIN1:          IN0         A,(STAT1)
                BIT         7,A
                JP          NZ,HEAD1

                LD          IY,INDEX        ;INDEX FOR COLUMN
                LD          E,00H           ;SHIFT ADDR EACH STEP = 6
                LD          D,20D           ;CHECK FINISH 20 COLUMN ?
BOARD10:        LD          IX,DATA0
                LD          C,E             ;
                LD          B,00H           ;INC ADDR = 6 PER ROUND
                ADD         IX,BC           ;
                LD          B,00H
                LD          C,06H

                IN0         A,(STAT1)       ;CHECK INTERRUPT
                BIT         7,A
                JP          NZ,HEAD1

LOOP10:         LD          A,(IX+0)        ;I/P DATA
                OUT0        (PORTA),A
                INC         IX
                LD          A,B             ;SELECT ROW (& BOARD)
                OUT0        (PORTB),A
                INC         B
                LD          A,07H           ;ENABLE SELECT (BOARD)
                OUT0        (PORTB),A
                LD          A,B             ;CHECK 6 PORT
                CP          C
                JP          NZ,LOOP10

BOARD11:        LD          IX,DATA1
                LD          C,E             ;
                LD          B,00H           ;INC ADDR = 6 PER ROUND
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                ADD     IX,BC       ;
                LD      B,08H
                LD      C,0EH

                IN0     A,(STAT1)   ;CHECK INTERRUPT
                BIT     7,A
                JP      NZ,HEAD1

LOOP11:         LD      A,(IX+0)    ;I/P DATA
                OUT0    (PORTA),A
                INC     IX
                LD      A,B         ;SELECT ROW (& BOARD)
                OUT0    (PORTB),A
                INC     B
                LD      A,0FH       ;ENABLE SELECT (BOARD)
                OUT0    (PORTB),A
                LD      A,B         ;CHECK 6 PORT
                CP      C
                JP      NZ,LOOP11

BOARD12:        LD      IX,DATA2
                LD      C,E         ;
                LD      B,00H       ;INC ADDR = 6 PER ROUND
                ADD     IX,BC       ;
                LD      B,10H
                LD      C,16H

                IN0     A,(STAT1)   ;CHECK INTERRUPT
                BIT     7,A
                JP      NZ,HEAD1

LOOP12:         LD      A,(IX+0)    ;I/P DATA
                OUT0    (PORTA),A
                INC     IX
                LD      A,B         ;SELECT ROW (& BOARD)
                OUT0    (PORTB),A
                INC     B
                LD      A,17H       ;ENABLE SELECT (BOARD)
                OUT0    (PORTB),A
                LD      A,B         ;CHECK 6 PORT
                CP      C
                JP      NZ,LOOP12

BOARD13:        LD      IX,DATA3
                LD      C,E         ;
                LD      B,00H       ;INC ADDR = 6 PER ROUND
                ADD     IX,BC       ;
                LD      B,18H
                LD      C,1EH

                IN0     A,(STAT1)   ;CHECK INTERRUPT
                BIT     7,A
                JP      NZ,HEAD1

LOOP13:         LD      A,(IX+0)    ;I/P DATA
                OUT0    (PORTA),A
                INC     IX
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                LD      A,B         ;SELECT ROW (& BOARD)
                OUT0    (PORTB),A
                INC     B
                LD      A,1FH        ;ENABLE SELECT (BOARD)
                OUT0    (PORTB),A
                LD      A,B         ;CHECK 6 PORT
                CP      C
                JP      NZ,LOOP13
                IN0     A,(STAT1)       ;CHECK INTERRUPT
                BIT     7,A
                JP      NZ,HEAD1

SCAN1:          LD      A,(IY)          ;D=0F0 =SELECT COLUMN
                OUT0    (PORTA),A
                INC     IY
                LD      A,1EH
                OUT0    (PORTB),A
                LD      A,1FH
                OUT0    (PORTB),A

DELAY1:         LD      A,05H
REPEAT1:        DEC     A
                JP      NZ,REPEAT1

OFF1:           LD      A,0FFH
                OUT0    (PORTA),A
                LD      A,1EH
                OUT0    (PORTB),A
                LD      A,0FFH
                OUT0    (PORTB),A

INC_E1:         LD      A,E         ;INC ADDR 6 FOR NEXT COLUMN
                ADD     A,6
                LD      E,A
                IN0     A,(STAT1)       ;CHECK INTERRUPT
                BIT     7,A
                JP      NZ,HEAD1

CHKCOL1:        LD      A,D         ;FINISH ALL 20 COLUMN
                DEC     A
                LD      D,A
                JP      NZ,BOARD10

DIRECTION:      LD      A,(WAY)
                DEC     A
                JP      Z,MAIN1         ;(8001H) = 01H
                DEC     A
                JP      Z,RIGHT         ;(8001H) = 02H
                DEC     A
                JP      Z,LEFT          ;(8001H) = 03H

;*************** 1. SHIFT RIGHT ************************
RIGHT:          LD      B,06H       ;BEGIN SHIFT DATA
                LD      IX,41DFH      ;FIRST MODULE OF LAST ROW
                LD      IY,BUFFER       ;BUFFER = 41E5H
                LD      D,50H       ;80 COLUMN
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
LOADDATA1:  LD      A,(IX)
            LD      (IY),A
            INC     IX
            INC     IY
            DEC     B
            JP      NZ,LOADDATA1    ;6 BYTE SHIFT ALREADY ?

            IN0     A,(STAT1)       ;CHECK INTERRUPT
            BIT     7,A
            JP      NZ,HEAD1

            LD      C,0F4H
            LD      B,0FFH
MOVIXIY1:   ADD     IX,BC
            ADD     IY,BC

            LD      B,06H
            DEC     D
            JP      NZ,LOADDATA1

            IN0     A,(STAT1)       ;CHECK INTERRUPT
            BIT     7,A
            JP      NZ,HEAD1

FIRSTCOL1:  LD      IX,4005H
            LD      IY,BUFFER
            LD      B,06H
FIRSTCOL6:  LD      A,(IY)
            LD      (IX),A
            INC     IX
            INC     IY
            DEC     B
            JP      NZ,FIRSTCOL6    ;6 MODULE OF FIRST COLUMN
            JP      MAIN1

;*************** 2. SHIFT LEFT ****************************
LEFT:       LD      B,06H           ;BEGIN SHIFT DATA
            LD      IX,4005H        ;FIRST MODULE OF FIRST ROW
            LD      IY,BUFFER       ;BUFFER = 41E5H

FIRSTTOBUF2:  LD    A,(IX)
            LD      (IY),A
            INC     IX
            INC     IY
            DEC     B
            JP      NZ,FIRSTTOBUF2  ;6 BYTE SHIFT ALREADY ?
            IN0     A,(STAT1)       ;CHECK INTERRUPT
            BIT     7,A
            JP      NZ,HEAD1


            LD      IY,4005H
            LD      B,06H
            LD      D,4FH           ;FOR CHECK 79 COLUMN
LOADDATA2:  LD      A,(IX)
            LD      (IY),A
            INC     IX
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                INC         IY
                DEC         B
                JP          NZ,LOADDATA2    ;6 BYTE SHIFT ALREADY ?

                IN0         A,(STAT1)       ;CHECK INTERRUPT
                BIT         7,A
                JP          NZ,HEAD1
                LD          B,06H
                DEC         D           ;79 COLUMN ALREADY ?
                JP          NZ,LOADDATA2

                IN0         A,(STAT1)       ;CHECK INTERRUPT
                BIT         7,A
                JP          NZ,HEAD1

BUFTOLAST2:     LD          A,(IX)
                LD          (IY),A
                INC         IX
                INC         IY
                DEC         B
                JP          NZ,BUFTOLAST2

                IN0         A,(STAT1)       ;CHECK INTERRUPT
                BIT         7,A
                JP          NZ,HEAD1
                JP          MAIN1

;********** MOTION PICTURE *********************

MAIN2:          LD          A,02H
                LD          (FRAME),A

                IN0         A,(STAT1)       ;CHECK INTERRUPT
                BIT         7,A
                JP          NZ,HEAD1

                LD          A,(SIZE)        ;SIZE = FRAME * 2
                LD          (COUNT),A
                LD          HL,0000H

NEXT2:          LD          IY,INDEX        ;INDEX FOR COLUMN
                LD          E,00H           ;SHIFT ADDR EACH STEP = 6
                LD          D,20D           ;CHECK FINISH 20 COLUMN ?
;************ 1 FRAME = 20 COLUMNS (ROUNDS) ************************
BOARD20:        LD          IX,DATA0
                LD          C,E         ;
                LD          B,00H           ;INC ADDR = 6 PER ROUND
                ADD         IX,BC           ;
                LD          B,H
                LD          C,L
                ADD         IX,BC           ;ADD 01E0H = 480 BYTE

CONT20:         LD          B,00H
                LD          C,06H
                IN0         A,(STAT1)       ;CHECK INTERRUPT
                BIT         7,A
                JP          NZ,HEAD1
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
LOOP20:     LD      A,(IX+0)        ;I/P DATA
            OUT0    (PORTA),A
            INC     IX
            LD      A,B             ;SELECT ROW (& BOARD)
            OUT0    (PORTB),A
            INC     B
            LD      A,07H           ;ENABLE SELECT (BOARD)
            OUT0    (PORTB),A
            LD      A,B             ;CHECK 6 PORT
            CP      C
            JP      NZ,LOOP20

BOARD21:    LD      IX,DATA1
            LD      C,E             ;
            LD      B,00H           ;INC ADDR = 6 PER ROUND
            ADD     IX,BC           ;
            LD      B,H
            LD      C,L
            ADD     IX,BC           ;ADD 01E0H = 480 BYTE

CONT21:     LD      B,08H
            LD      C,0EH
            IN0     A,(STAT1)       ;CHECK INTERRUPT
            BIT     7,A
            JP      NZ,HEAD1

LOOP21:     LD      A,(IX+0)        ;I/P DATA
            OUT0    (PORTA),A
            INC     IX
            LD      A,B             ;SELECT ROW (& BOARD)
            OUT0    (PORTB),A
            INC     B
            LD      A,0FH           ;ENABLE SELECT (BOARD)
            OUT0    (PORTB),A
            LD      A,B             ;CHECK 6 PORT
            CP      C
            JP      NZ,LOOP21

BOARD22:    LD      IX,DATA2
            LD      C,E             ;
            LD      B,00H           ;INC ADDR = 6 PER ROUND
            ADD     IX,BC           ;
            LD      B,H             ;MOVE POINTER TO NEXT FRAME
            LD      C,L
            ADD     IX,BC           ;ADD 01E0H = 480 BYTE

CONT22:     LD      B,10H
            LD      C,16H
            IN0     A,(STAT1)       ;CHECK INTERRUPT
            BIT     7,A
            JP      NZ,HEAD1

LOOP22:     LD      A,(IX+0)        ;I/P DATA
            OUT0    (PORTA),A
            INC     IX
            LD      A,B             ;SELECT ROW (& BOARD)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                OUT0        (PORTB),A
                INC         B
                LD          A,17H          ;ENABLE SELECT (BOARD)
                OUT0        (PORTB),A
                LD          A,B            ;CHECK 6 PORT
                CP          C
                JP          NZ,LOOP22

BOARD23:        LD          IX,DATA3
                LD          C,E            ;
                LD          B,00H          ;INC ADDR = 6 PER ROUND
                ADD         IX,BC          ;
                LD          B,H            ;MOVE POINTER TO NEXT FRAME
                LD          C,L
                ADD         IX,BC          ;ADD 01E0H = 480 BYTE

CONT23:         LD          B,18H
                LD          C,1EH
                IN0         A,(STAT1)      ;CHECK INTERRUPT
                BIT         7,A
                JP          NZ,HEAD1
LOOP23:         LD          A,(IX+0)       ;I/P DATA
                OUT0        (PORTA),A
                INC         IX
                LD          A,B            ;SELECT ROW (& BOARD)
                OUT0        (PORTB),A
                INC         B
                LD          A,1FH          ;ENABLE SELECT (BOARD)
                OUT0        (PORTB),A
                LD          A,B            ;CHECK 6 PORT
                CP          C
                JP          NZ,LOOP23
                IN0         A,(STAT1)      ;CHECK INTERRUPT
                BIT         7,A
                JP          NZ,HEAD1

SCAN2:          LD          A,(IY)         ;D=0F0 =SELECT COLUMN
                OUT0        (PORTA),A
                INC         IY
                LD          A,1EH
                OUT0        (PORTB),A
                LD          A,1FH
                OUT0        (PORTB),A

DELAY2:         LD          A,0FH
REPEAT2:        DEC         A
                JP          NZ,REPEAT2

OFF2:           LD          A,0FFH
                OUT0        (PORTA),A
                LD          A,1EH
                OUT0        (PORTB),A
                LD          A,1FH
                OUT0        (PORTB),A

INC_E2:         LD          A,E            ;INC ADDR 6 FOR NEXT COLUMN
                ADD         A,6
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                LD          E,A
                IN0         A,(STAT1)       ;CHECK INTERRUPT
                BIT         7,A
                JP          NZ,HEAD1

CHKCOL2:        LD          A,D             ;FINISH ALL 20 COLUMN
                DEC         A
                LD          D,A
                JP          NZ,BOARD20

;*********** END 1 FRAME *************************
                LD          A,(FRAME)
                DEC         A
                LD          (FRAME),A
                JP          NZ,CROSS

                LD          B,01H           ;MOVE POINTER TO NEXT FRAME
                LD          C,0E0H
                ADD         HL,BC
                LD          A,02H
                LD          (FRAME),A

CROSS:          IN0         A,(STAT1)       ;CHECK INTERRUPT
                BIT         7,A
                JP          NZ,HEAD1
                LD          A,(COUNT)       ;COUNT FOR CHECK LAST FRAME
                DEC         A
                LD          (COUNT),A
                JP          Z,MAIN2         ;GO TO START FRAME 1
                JP          NEXT2           ;GO TO DISPLAY NEXT FRAME

;********** MODE3 = MOVING TEXT ***********************

MAIN3:          IN0         A,(STAT1)
                BIT         7,A
                JP          NZ,HEAD1
MAIN3_1:        LD          A,50H
                LD          (MODE),A

                LD          A,(SIZE)
                SRA         A
                DEC         A
                LD          (COUNT),A
                LD          HL,0000H

HEY:            LD          IY,INDEX        ;INDEX FOR COLUMN
                LD          E,00H           ;SHIFT ADDR EACH STEP = 6
                LD          D,20D           ;CHECK FINISH 20 COLUMN ?

BOARD30:        LD          IX,DATA0
                LD          C,E             ;
                LD          B,00H           ;INC ADDR = 6 PER ROUND
                LD          A,L
                ADD         A,C
                LD          C,A
                LD          A,H
                ADC         A,B
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                LD      B,A
                ADD     IX,BC
                LD      B,00H
                LD      C,06H

                IN0     A,(STAT1)       ;CHECK INTERRUPT
                BIT     7,A
                JP      NZ,HEAD1

LOOP30:         LD      A,(IX+0)        ;I/P DATA
                OUT0    (PORTA),A
                INC     IX
                LD      A,B             ;SELECT ROW (& BOARD)
                OUT0    (PORTB),A
                INC     B
                LD      A,07H           ;ENABLE SELECT (BOARD)
                OUT0    (PORTB),A
                LD      A,B             ;CHECK 6 PORT
                CP      C
                JP      NZ,LOOP30

BOARD31:        LD      IX,DATA1
                LD      C,E             ;
                LD      B,00H           ;INC ADDR = 6 PER ROUND
                LD      A,L
                ADD     A,C
                LD      C,A
                LD      A,H
                ADC     A,B
                LD      B,A
                ADD     IX,BC
                LD      B,08H
                LD      C,0EH
                IN0     A,(STAT1)       ;CHECK INTERRUPT
                BIT     7,A
                JP      NZ,HEAD1

LOOP31:         LD      A,(IX+0)        ;I/P DATA
                OUT0    (PORTA),A
                INC     IX
                LD      A,B             ;SELECT ROW (& BOARD)
                OUT0    (PORTB),A
                INC     B
                LD      A,0FH           ;ENABLE SELECT (BOARD)
                OUT0    (PORTB),A
                LD      A,B             ;CHECK 6 PORT
                CP      C
                JP      NZ,LOOP31

BOARD32:        LD      IX,DATA2
                LD      C,E             ;
                LD      B,00H           ;INC ADDR = 6 PER ROUND
                LD      A,L
                ADD     A,C
                LD      C,A
                LD      A,H
                ADC     A,B
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                LD      B,A
                ADD     IX,BC
                LD      B,10H
                LD      C,16H
                IN0     A,(STAT1)       ;CHECK INTERRUPT
                BIT     7,A
                JP      NZ,HEAD1

LOOP32:         LD      A,(IX+0)        ;I/P DATA
                OUT0    (PORTA),A
                INC     IX
                LD      A,B             ;SELECT ROW (& BOARD)
                OUT0    (PORTB),A
                INC     B
                LD      A,17H           ;ENABLE SELECT (BOARD)
                OUT0    (PORTB),A
                LD      A,B             ;CHECK 6 PORT
                CP      C
                JP      NZ,LOOP32

BOARD33:        LD      IX,DATA3
                LD      C,E             ;
                LD      B,00H           ;INC ADDR = 6 PER ROUND
                LD      A,L
                ADD     A,C
                LD      C,A
                LD      A,H
                ADC     A,B
                LD      B,A
                ADD     IX,BC
                LD      B,18H
                LD      C,1EH
                IN0     A,(STAT1)       ;CHECK INTERRUPT
                BIT     7,A
                JP      NZ,HEAD1

LOOP33:         LD      A,(IX+0)        ;I/P DATA
                OUT0    (PORTA),A
                INC     IX
                LD      A,B             ;SELECT ROW (& BOARD)
                OUT0    (PORTB),A
                INC     B
                LD      A,1FH           ;ENABLE SELECT (BOARD)
                OUT0    (PORTB),A
                LD      A,B             ;CHECK 6 PORT
                CP      C
                JP      NZ,LOOP33
                IN0     A,(STAT1)       ;CHECK INTERRUPT
                BIT     7,A
                JP      NZ,HEAD1

SCAN3:          LD      A,(IY)          ;D=0F0 =SELECT COLUMN
                OUT0    (PORTA),A
                INC     IY
                LD      A,1EH
                OUT0    (PORTB),A
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
DELAY3:     LD      A,0AH
REPEAT3:    DEC     A
            JP      NZ,REPEAT3

OFF3:       LD      A,0FFH
            OUT0    (PORTA),A
            LD      A,1EH
            OUT0    (PORTB),A
            LD      A,0FFH
            OUT0    (PORTB),A

INC_E3:     LD      A,E         ;INC ADDR 6 FOR NEXT COLUMN
            ADD     A,6
            LD      E,A
            IN0     A,(STAT1)   ;CHECK INTERRUPT
            BIT     7,A
            JP      NZ,HEAD1

CHKCOL3:    LD      A,D         ;FOR FINISH ALL 20 COLUMN
            DEC     A
            LD      D,A
            JP      NZ,BOARD30

            LD      B,00H
            LD      C,06H
            ADD     HL,BC
            LD      A,(MODE)
            DEC     A
            LD      (MODE),A
            JP      NZ,HEY

            LD      A,50H
            LD      (MODE),A
            LD      A,(COUNT)
            DEC     A
            LD      (COUNT),A
            JP      NZ,HEY
            JP      MAIN3
END
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้