จอแสดงผล

SCORE BOARD

โดย

นางสาวภัทริน วิรารัตน์

นางสาวมณีสุดา ชะมด

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมโทรคมนาคม

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2541

เลขหมู่.....
เลขทะเบียน.....32601
วัน, เดือน, ปี 18 พ.ค. 2542

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งไม่อนุญาตให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จอแสดงผล

SCORE BOARD

โดย

นางสาวภัทริน วิรารัตน์ 38014368

นางสาวมณีสุดา ชะมด 38014386

อาจารย์ที่ปรึกษา

รศ.สมยศ จุณณะปิยะ

อาจารย์สมเกียรติ ฤกษ์วรัญญู

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมโทรคมนาคม

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2541

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์ปีการศึกษา 2541

ภาควิชาวิศวกรรมโทรคมนาคม

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง **จอแสดงผล**

SCORE BOARD

ผู้จัดทำ

1. นางสาวภัทริน วิรารัตน์ 38014368

2. นางสาวมณีสุดา ชะมด 38014386

..... อาจารย์ที่ปรึกษา

(รศ. สมยศ จุณณะปิยะ)

..... อาจารย์ที่ปรึกษา

(อาจารย์สมเกียรติ ฤกษ์วรัญญู)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จอแสดงผล

SCORE BOARD

โดย นางสาวภัทริน วิรารัตน์ 38014368
นางสาวมณีสุดา ชะมด 38014386

อาจารย์ที่ปรึกษา รศ.สมยศ จุณณะปิยะ
อาจารย์สมเกียรติ ฤกษ์วรัญญู

บทคัดย่อ

ปริญญานิพนธ์ฉบับนี้เป็นการศึกษาเกี่ยวกับจอแสดงผล ซึ่งเป็นการนำเสนอรูปแบบของสื่ออีกรูปแบบหนึ่งสามารถแสดงข้อมูลได้ในลักษณะตัวอักษรภาษาอังกฤษและรูปภาพ โดยทำการแสดงบนแผงแอลอีดีขนาด 32*32 (32 แถว 32 คอลัมน์) ในลักษณะเมตริกซ์จุด ผู้ใช้สามารถป้อนข้อความหรือทำการเลือกรูปภาพผ่านทางจอคอมพิวเตอร์ หลังจากนั้นข้อความหรือรูปภาพก็จะไปแสดงบนจอแสดงผลทันที

ABSTRACT

This project is studying in the scope of SCORE BOARD, it is a type of mass media. It can display in both English characters and graphics on LED dot matrix circuit board by user can key the message in or enter the chosen picture via computer and then that item will display on LED dot matrix circuit board, in this case we call “SCORE BOARD”.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

	หน้า
บทที่ 1 บทนำ	1
บทที่ 2 ทฤษฎีหรือหลักการ	2
2.1 แอลอีดี	2
2.2 หลักการทำงานของแผงแสดงผลทั่วไป	3
2.3 ไมโครคอนโทรเลอร์ตระกูล MCS-51	6
2.4 8255 PROGRAMMABLE PERIPHERAL INTERFACE	10
2.5 พอร์ทอนุกรม	17
บทที่ 3 การคำนวณและการสร้าง	19
3.1 บล็อกไดอะแกรมของจอแสดงผล	19
3.2 ส่วนควบคุม	19
3.3 ส่วนแสดงผล	22
3.4 โปรแกรม	26
บทที่ 4 การทดลองและผลการทดลอง	34
บทที่ 5 สรุปและบทวิจารณ์	38
ภาคผนวก	
หนังสืออ้างอิง	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป

หน้า

รูปที่ 2.1 แสดงสัญลักษณ์ของแอลอีดี 2
รูปที่ 2.2 แสดงลักษณะการไบแอสที่จะทำให้แอลอีดีทำงานได้ 2
รูปที่ 2.3 แสดงโครงสร้างหลักของแผงแสดงผลทั่วไป 4
รูปที่ 2.4 แสดงการสแกนทางคอลัมน์ 5
รูปที่ 2.5 แสดงการสแกนทางแถว 5
รูปที่ 2.6 แสดงโครงสร้างของ 8051 7
รูปที่ 2.7 แสดงแผนภูมิหน่วยความจำของ 8051 9
รูปที่ 2.8 แสดงโครงสร้างของ 8255 10
รูปที่ 2.9 รูปแบบการควบคุมของโหมดและบิท 13
รูปที่ 2.10 การให้เวิร์ดควบคุมและการกำหนดพอร์ท A,B และ C 15
รูปที่ 2.11 การกำหนดทางขาอินพุท 16
รูปที่ 2.12 การกำหนดขาเอาท์พุท 16
รูปที่ 2.13 ลักษณะของสัญญาณตามมาตราฐาน RS-232 18
รูปที่ 3.1 แสดงบล็อกไดอะแกรมของจอแสดงผล 19
รูปที่ 3.2 แสดงการต่อขาสัญญาณของ DB9 19
รูปที่ 3.3 แสดงวงจร MAX232 19
รูปที่ 3.4 แสดงการทำงานของ 8951 20
รูปที่ 3.5 แสดงหน่วยความจำสำหรับเก็บโปรแกรมภายในและภายนอก 20
รูปที่ 3.6 วงจรส่วนควบคุม 21
รูปที่ 3.7 แสดงวงจรแลตซ์ด้านแถว 22
รูปที่ 3.8 แสดงวงจรสวิตซ์ด้านแถว 22
รูปที่ 3.9 แสดงวงจรบัฟเฟอร์ 23
รูปที่ 3.10 แสดงวงจรแลตซ์ด้านคอลัมน์ 23
รูปที่ 3.11 แสดงวงจรสวิตซ์ด้านคอลัมน์ 23
รูปที่ 3.12 แสดงการวางแผงแอลอีดีขนาดชุดละ 8X8 จำนวน 16 ชุด 24
รูปที่ 3.13 วงจรส่วนแสดงผล 25
รูปที่ 3.14 โฟลว์ชาร์ทแสดงขั้นตอนการทำงานในการรับรูปแบบที่จะแสดงผลและส่งข้อมูลไปยังส่วนแสดงผล 28
รูปที่ 3.15 โฟลว์ชาร์ทแสดงขั้นตอนการทำงานของโหมด 1 (ตัวอักษร) 28
รูปที่ 3.16 โฟลว์ชาร์ทแสดงขั้นตอนการทำงานของโหมด 2 (รูปภาพ) 29
รูปที่ 3.17 แสดงรีจิสเตอร์ TMOD 30

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 3.18 แสดงรีจิสเตอร์ PCON 30
รูปที่ 3.19 แสดงรีจิสเตอร์ SCON 30
รูปที่ 3.20 แสดงรีจิสเตอร์ IE 30
รูปที่ 3.21 โฟลว์ชาร์ทแสดงขั้นตอนการทำงานในการรับข้อมูลและแยกไปตามโหมดการแสดงผลแบบต่าง ๆ 31
รูปที่ 3.22 โฟลว์ชาร์ทแสดงขั้นตอนการทำงานในการแสดงผลของโหมด 1 (ตัวอักษร) 32
รูปที่ 3.23 โฟลว์ชาร์ทแสดงขั้นตอนการทำงานในการแสดงผลของโหมด 2 (รูปภาพ) 33
รูปที่ 4.1 แสดงสัญญาณนาฬิกาในการเลือกชุดการแสดงผลทางคอลัมน์ 34
รูปที่ 4.2 แสดงข้อมูลทางด้านแถวที่ใช้ในการสแกนให้เกิดภาพจากไอซี 74LS377 34
รูปที่ 4.3 แสดงการเปรียบเทียบข้อมูลทางด้านอินพุทเอาท์พุทของไอซีเบอร์ 74HCT574 35
รูปที่ 4.4 วงจรแลตช์ข้อมูลและขับกระแส 35
รูปที่ 4.5 ภาพถ่ายจากจอแสดงผลแสดงตัวอักษรภาษาอังกฤษ A-R 36
รูปที่ 4.6 ภาพถ่ายจากคอมพิวเตอร์แสดงตัวอักษรภาษาอังกฤษ A-R 36
รูปที่ 4.7 ภาพถ่ายจากจอแสดงผลแสดงคำว่า GOOD MORNING 37
รูปที่ 4.8 ภาพถ่ายจากคอมพิวเตอร์แสดงคำว่า GOOD MORNING 37

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

	หน้า
ตารางที่ 2.1 แสดงโหมดต่างๆ ของ 8255	11
ตารางที่ 2.2 ตารางความจริงของ 8255	12
ตารางที่ 2.3 แสดงการโปรแกรม INTE ของพอร์ท A , B	17

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

ในปัจจุบันการโฆษณาต่างๆอาศัยจอแสดงผลในการดึงดูดความสนใจของผู้คนที่พบเห็น และเป็นที่นิยมกันอย่างแพร่หลาย สามารถพบเห็นได้ตามย่านธุรกิจต่างๆ , ศูนย์การค้า , โรงงาน , โรงพยาบาล หน่วยราชการบางแห่งหรือแม้แต่ริมถนนตามสี่แยกมีให้พบเห็นมากมาย การทำงานจะอาศัยการควบคุมจากส่วนประมวลผล ทำให้เกิดการแสดงข้อความหรือรูปภาพตามที่กำหนด แล้วให้ข้อความเคลื่อนที่ไป เช่น ซ้ายไปขวา , ขวาไปซ้าย , บนลงล่าง , และอื่นๆ

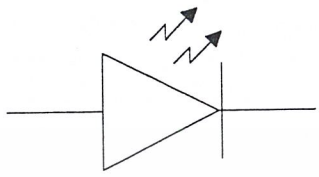
จากประโยชน์และจุดประสงค์การทำงานของแผงแสดงผลทำให้เกิดการศึกษาโครงงานชิ้นนี้ โดยแบ่งออกเป็น 2 ส่วน คือ ฮาร์ดแวร์ และ ซอฟท์แวร์ ส่วนของฮาร์ดแวร์ประกอบด้วยตัวจอแสดงผลแอลอีดี ขนาด 32X32 (32 แถว 32 คอลัมน์) ภาคควบคุมให้ชิพตระกูล MCS - 51 วงจรแลตซ์ข้อมูล และวงจรขับกระแส ส่วนที่ 2 เป็นซอฟท์แวร์ใช้ภาษาแอสแซมบลี ประกอบกับการแลตซ์ข้อมูลทางด้านคอลัมน์ และสแกนทางด้านแถว เหมือนกับการสแกนจอภาพโทรทัศน์ , การจัดการเกี่ยวกับตัวอักษร , การเลื่อนภาพบนจอแสดงผลจะจัดการกับข้อมูลในหน่วยความจำชั่วคราวที่เรียกว่า บัฟเฟอร์แสดงผล โดยนำข้อความถัดไปมาเก็บไว้ในบัฟเฟอร์แสดงผลสำรองก่อนที่จะนำมาแทนที่บัฟเฟอร์แสดงผลหลัก การแทนที่ของข้อมูลในแบบต่างๆก็จะได้การเลื่อนข้อความในทิศทางต่างๆ , การรับส่งข้อมูลทางพอร์ทอนุกรม ทั้งหมดนี้เป็นส่วนประกอบของซอฟท์แวร์ที่เขียนขึ้นในโครงงานนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

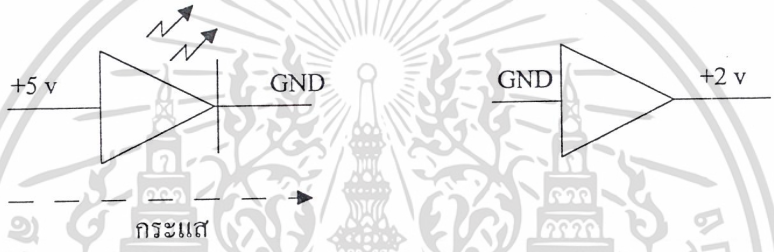
บทที่ 2
ทฤษฎีหรือหลักการ

2.1 แอลอีดี (Light Emitting Diode)

แอลอีดี คือ ไดโอดชนิดหนึ่งที่ถูกออกแบบมาเป็นพิเศษ ให้เปล่งแสงออกมาได้ในขณะที่ถูกไบแอสแบบฟอร์เวิร์ด (Forword Bias) สัญลักษณ์ของแอลอีดีจะมีแสงเปล่งออกมาดังรูปที่ 2.1



รูปที่ 2.1 แสดงสัญลักษณ์ของแอลอีดี



(1) ฟอร์เวิร์ด ไบแอส (Forward bias) (2) รีเวิร์ด ไบแอส (Reward bias)

รูปที่ 2.2 แสดงลักษณะการไบแอสที่จะทำให้แอลอีดีเปล่งแสงได้

การที่แอลอีดีเปล่งพลังงานออกมาได้ โดยอาศัยโครงสร้างการจัดเรียงตัวภายในของสารกึ่งตัวนำ คือ อิเลคตรอนในอะตอมของสารจะอยู่เป็นชั้นๆ (level) ซึ่งแต่ละชั้นของอิเลคตรอน ก็จะมีระดับพลังงานไม่เท่ากัน ชั้นที่อยู่ห่างจากศูนย์กลางมาก ก็จะมีพลังงานมากกว่าชั้นที่อยู่ใกล้กว่า สำหรับสารกึ่งตัวนำล้วนๆที่อุณหภูมิ 0 องศาเคลวินจะมีอิเลคตรอนอยู่ในชั้นพลังงานในๆเต็มและชั้นนอกๆ จะไม่มีอิเลคตรอนอยู่เลย ชั้นพลังงานนอกสุดของชั้นที่มีอิเลคตรอนเต็ม จะเรียกว่า วาเลนซ์แบนด์ (Valence Band) และชั้นในสุดของชั้นที่ไม่มีอิเลคตรอนอยู่ จะเรียกว่า คอนดักชั่นแบนด์ (Conduction Band) ซึ่งพลังงานที่แตกต่างระหว่างชั้นพลังงานทั้งสองนี้ เรียกว่า Eg (Energy gap) อิเลคตรอนจะไม่สามารถอยู่ใน Eg ได้เลยถ้าไม่อยู่ในคอนดักชั่นแบนด์ ก็จะอยู่ในวาเลนซ์แบนด์ การใส่สารไม่บริสุทธิ์ (Impurity) เข้าไปจะเป็นผลทำให้อิเลคตรอนสามารถที่จะอาศัยอยู่ใน Eg ได้ ถ้าอะตอมใดมีอิเลคตรอนอิสระอยู่ในคอนดักชั่นแบนด์ เราก็จะเรียกอะตอมนี้ว่า อะตอมผู้ให้ (Donor Atom) หรือเรียกอีกอย่างว่าเป็นสารกึ่งตัวนำชนิดเอ็น (N-type) ถ้าอะตอมใดมีอิเลคตรอนขาดหายไป (มี Hold เกิดขึ้น) ในวาเลนซ์แบนด์ เราจะเรียกอะตอมนี้ว่า อะตอมผู้รับ (Acceptor Atom) หรือเรียกอีกอย่างว่าเป็นสารกึ่งตัวนำชนิดพี(P-type) เมื่ออิเลคตรอนอิสระในอะตอมผู้ให้มารวมตัว(Recombine)กับโฮลในอะตอมผู้รับจะมีการเปลี่ยนแปลงพลังงานจากชั้นที่มีพลังงานโฟตรอน (Photon)ซึ่งความถี่ของโฟตรอนจะถูกกำหนดด้วยระดับพลังงานที่เปลี่ยนแปลง ยิ่งพลังงานมีมากก็ได้ความถี่สูงขึ้น

เอกสารนี้สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำหรับสารกึ่งตัวนำที่นำมาใช้ทำแอลอีดีนั้น นักวิทยาศาสตร์ได้ทำการคัดเลือกชนิดที่มีคุณสมบัติง่ายต่อการสร้างขึ้นและมีเสถียรภาพในสภาวะแวดล้อมต่างๆ สารเหล่านี้มีแกลเลียมอาร์ซีไนด์ (GaAs) และแกลเลียมอาร์ซีไนด์ฟอสไฟด์ (GaAsP) แอลอีดีที่ใช้ GaAs ทำจะมีความยาวคลื่น 9000 อังสตรอม ซึ่งในช่วงความยาวคลื่นนี้ตาคนเรามองไม่เห็น ส่วนแอลอีดีที่ใช้ GaAsP นั้นมีความยาวคลื่น 6600 อังสตรอม ซึ่งอยู่ในช่วงที่ตาคนมองเห็น ความถี่ที่ได้ออกมาค่อนข้างจะคงที่ (monochromatic) แต่อาจจะมีการเปลี่ยนแปลงไปบ้าง เนื่องจากคุณสมบัติของอะตอมซึ่งจะเคลื่อนที่อย่างสะเปะสะปะ

แอลอีดีที่ผลิตขึ้นจะทำโดยนำเอาสารกึ่งตัวนำชนิด P และ N มาประกบกัน ถ้าให้แรงดันเข้าไปแบบฟอร์เวิร์ดไบแอส จะทำให้อิเลคตรอนและโฮลไหลมารวมกัน ซึ่งจะเปล่งพลังงานแสงออกมาได้ โดยแอลอีดีจะนำกระแสที่ศักดาค่าหนึ่ง (ประมาณ 1 โวลท์สำหรับ GaAs) แล้วมีกระแสไหลมากขึ้นเมื่อเพิ่มศักดาเข้าไปพลังงานแสงที่ได้ออกมาจะแปรผันโดยตรงกับกระแสที่ไหล

แอลอีดีโดยทั่วๆไปจะไม่สามารถมองเห็นได้ทุกแง่ทุกมุม กล่าวคือ เราจะเห็นแสงที่เปล่งออกมาจากแอลอีดีในมุมแคบๆที่ทำกับแกนของมันเท่านั้น

2.1.1 ข้อดีของแอลอีดี

1. สามารถสว่างและดับได้ในเวลาที่สั้นมากเพียงไม่กี่นาโนเซค (nSec) ดังนั้นจึงสามารถให้แสงจากแอลอีดีเป็นตัวนำข่าวสารหรือข้อมูลไปใช้ในการสื่อสารที่มีความถี่สูงเป็นเมกกะเฮิร์ท (MHz)
2. ใช้แรงดันต่ำ จึงทำให้เหมาะที่จะใช้ควบคู่กับไอซี (IC) ได้
3. มีอายุการใช้งานมากกว่าหลอดไฟ
4. มีกำลังส่องสว่างแปรผันตามกระแสที่ไหลผ่าน ดังนั้นจึงสามารถส่งข้อมูลในรูปของกระแสจากจุดหนึ่งไปยังอีกจุดหนึ่งที่ห่างไกลออกไปโดยใช้แสงได้
5. แสงที่เปล่งออกไปมีช่วงความยาวคลื่นแคบ หรือพลังงานแสงที่เปล่งออกมาจากที่มีสีเดียว (Monochromatic)
6. สามารถผลิตให้แอลอีดีมีขนาดเล็กกระทัดรัดได้
7. ทนทานต่อการกระทบกระเทือน (พวกหลอดธรรมดาเวลาถูกกระเทือนไส้หลอดอาจจะขาดได้ง่ายๆ)
8. มีหลายสี
9. มีราคาถูก

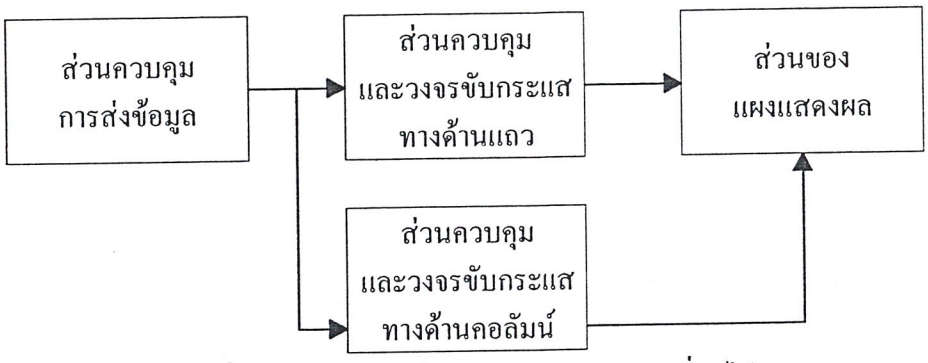
2.1.2 ข้อเสียของแอลอีดี

1. กำลังส่องสว่างขึ้นกับอุณหภูมิ สำหรับการใช้งานที่ต้องการกำลังส่องสว่างเต็มที่ต้องมีวงจรควบคุมกำลังส่องสว่างของแอลอีดี
2. เสียหายได้ง่าย ถ้าแรงดันและกระแสสูงกว่าที่ตัวมันจะทนได้

2.2 หลักการทำงานของแผงแสดงผลทั่วไป

โดยการพัฒนาแผงแสดงผลอิเล็กทรอนิกส์อย่างกว้างขวางและต่อเนื่องกันมาโดยตลอด ทำให้การแสดงผลใกล้เคียงใกล้เคียงและสามารถแสดงผลแทนจอภาพคอมพิวเตอร์หรือโทรทัศน์ได้ โครงสร้างต่างๆ ของแผงแสดงผลอิเล็กทรอนิกส์ทั่วๆ ไปจะมีลักษณะที่คล้ายๆ กัน ดังแสดงในรูปที่ 2.3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.3 แสดงโครงสร้างหลักของแผงแสดงผลทั่วๆไป

จากรูปที่ 2.3 จะเห็นได้ว่าโครงสร้างหลักของแผงแสดงผลอิเล็กทรอนิกส์จะประกอบไปด้วย 4 ส่วนหลักๆ คือ

ส่วนที่ 1 ส่วนควบคุมการส่งข้อมูล มีหน้าที่ในการจัดการเกี่ยวกับข้อมูลบนแผงแสดงผล และยังเป็นส่วนที่สร้างสัญญาณควบคุมต่างๆ ของระบบ เช่น สร้างสัญญาณควบคุมทางด้านแนวนอนและแนวตั้ง เพื่อให้ส่วนต่างๆ ทำงานประสานกันได้อย่างถูกต้อง

ส่วนที่ 2 ส่วนควบคุมและวงจรขับกระแสทางด้านแถว สำหรับจัดการเรื่องข้อมูลทางแนวนอนที่ได้รับมาจากส่วนที่ 1 ในการเปิดหรือปิดวงจรด้านแถวนั้นๆ เพื่อใช้ในการแสดงผลในลักษณะของการติดและดับก่อนที่จะส่งข้อมูลที่ได้รับมาแก่แผงแสดงผล และยังทำให้การแสดงผลที่ได้แน่นอนและสามารถควบคุมเป็นจุดๆ ได้

ส่วนที่ 3 ส่วนควบคุมและวงจรขับกระแสทางด้านคอลัมน์ สำหรับการจัดการข้อมูลในแต่ละหลัก ที่ได้รับมาจากส่วนที่ 1 การทำงานคล้ายคลึงกับส่วนควบคุมและวงจรขับกระแสทางแนวนอน และทั้ง 2 ส่วนนี้จะต้องมีการทำงานที่ประสานกันอย่างดี เพื่อให้การควบคุมแผงแสดงผลเป็นไปอย่างถูกต้องและแม่นยำ

ส่วนที่ 4 ส่วนของแผงแสดงผล เป็นส่วนที่ใช้ในการแสดงผลจากข้อมูลที่ได้รับมาทางด้านแนวตั้งและแนวนอน ซึ่งส่วนนี้จะใช้ไดโอดเปล่งแสง (LED : Light Emitting Diode) เป็นส่วนประกอบ

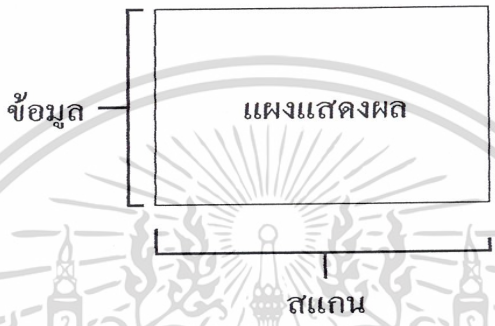
2.2.1 <u>หลักในการทำให้เกิดภาพบนแผงแสดงผล</u>

เราสามารถทำให้ภาพบนแผงแสดงผลแบบจุด(DOT MATRIX) ได้จากการอาศัยหลักการสแกนคล้ายๆ กับจอภาพของโทรทัศน์นั่นเอง คือ ภาพที่ปรากฏนั้นจะประกอบด้วยการรวมเส้นที่เกิดจากการสแกนครั้งละหนึ่งเส้น เป็นจำนวนเท่ากับคอลัมน์(COLUMN) หรือแถว(ROW) ของแผงแสดงผลแบบจุดแล้วแต่รูปแบบของการสแกน ที่ทำซ้ำกันหลายๆ รอบ ความเร็วที่ใช้นั้นต้องมากกว่าหรือเท่ากับ 25 รอบต่อวินาที ภาพที่ได้นั้นจะนิ่งไม่กะพริบหรือเราไม่สามารถจับการสแกนได้นั่นเอง โดยการสแกนนั้นจะใช้สวิตซ์ซึ่งถูกควบคุมด้วยซอฟท์แวร์ ซึ่งสามารถกำหนดความเร็วได้ โดยส่วนทางด้านการป้อนข้อมูลนั้นจะต้องสัมพันธ์กับการสแกนด้วย เช่น ข้อมูลคอลัมน์แรกเข้ามา ก็ต้องตรงกับตำแหน่งที่สวิตซ์สแกนแรกต่อกับวงจรในคอลัมน์แรกด้วย เป็นต้น ซึ่งสามารถนำหลักการนี้เองไปสร้างเป็นวงจรใช้งานจริง ซึ่งสามารถแบ่งการสแกนได้เป็น

เอกสารนี้สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.1.1 การสแกนทางคอลัมน์ (COLUMN)

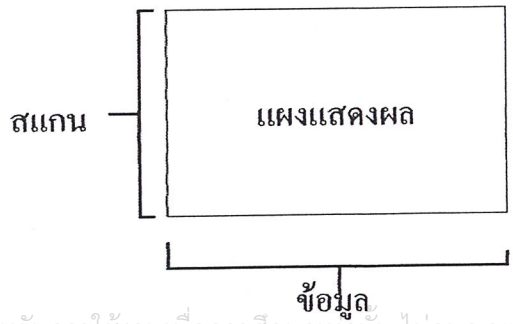
ในการสแกนทางคอลัมน์ จะทำการส่งข้อมูลออกไปทางแถว โดยส่งข้อมูลตัวที่ 1 ออกไป แล้วให้หลักที่ 1 ทำงาน จากนั้นก็ทำการส่งข้อมูลตัวที่ 2 ออกไปแล้วให้หลักที่ 2 ทำงาน ทำเช่นนี้ไปจนกระทั่งข้อมูลถูกส่งออกไปครบหมดทุกหลัก ก็จะเป็นการสแกนครบ 1 รอบ แต่จะสังเกตว่าการสแกนให้ครบหนึ่งภาพจะต้องใช้เวลายาวนานเพราะฉะนั้นกระแสเฉลี่ยที่ไหลผ่านแอลอีดี จึงมีค่าน้อย เพราะฉะนั้นจะทำให้แอลอีดีไม่สว่าง การสแกนช้าเกินไปภาพก็จะเกิดการกะพริบซึ่งเป็นผลเสียต่อแผงแสดงผล แต่มีข้อดีที่ว่าใช้อุปกรณ์ เช่น บัฟเฟอร์ต่ออยู่ที่เอาท์พุทจึงมีส่วนประกอบที่มากกว่าตัวสวิตช์ ดังนั้นอุปกรณ์ในส่วนนี้น้อยขึ้นทำให้เกิดการประหยัด



รูปที่ 2.4 แสดงการสแกนทางคอลัมน์

2.2.1.2 การสแกนทางแถว (ROW)

การทำงานหลักๆ เช่นเดียวกับการสแกนทางคอลัมน์ แต่จะสลับทิศทาง จะทำการส่งข้อมูลออกไปทางคอลัมน์ โดยจะส่งข้อมูลออกไปจนครบทุกหลักก่อน แล้วให้แถวที่ 1 ทำงาน จากนั้นก็ทำการส่งข้อมูลชุดถัดไปออกไปจนครบหมดทุกหลัก แล้วให้แถวที่ 2 ทำงาน ทำเช่นนี้จนกระทั่งข้อมูลถูกส่งออกไปครบหมดทุกแถวก็จะเป็นการสแกนครบ 1 รอบ เพราะฉะนั้นการต่อในลักษณะนี้จะทำให้ระยะเวลาในการสแกนครบหนึ่งภาพได้รวดเร็วหรือถ้าเวลาเท่าๆ กันแบบนี้จะสแกนได้หลายครั้งกว่าแบบสแกนทางคอลัมน์หรือจะมีผลต่อความสว่างของหลอดแอลอีดี จะสว่างกว่า เพราะกระแสเฉลี่ยที่ไหลผ่านจะมากกว่านั่นเอง ซึ่งหลักการของแผงแสดงผลแบบนี้เองที่เลือกมาใช้แก้ไขปัญหาความสว่างของหลอดแอลอีดี และถ้าจัดเวลาให้เหมาะสมแล้วเวลาทำการสแกนจะไม่ทำให้ภาพเกิดอาการพริ้ว แต่ก็มีข้อเสียอยู่ที่ว่าระบบฮาร์ดแวร์จะแพงกว่าแบบแรก และการเขียนโปรแกรมควบคุมให้ตัวอักษรเลื่อนทำได้ยากกว่าแบบแรก



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 2.5 แสดงการสแกนทางแถว

2.3 ไมโครคอนโทรลเลอร์ตระกูล MCS-51

ไมโครคอนโทรลเลอร์แบบชิพเดี่ยว (Single Chip Microcontroller) คือ ไมโครคอมพิวเตอร์แบบที่มีขนาดเล็กโดยบรรจุไว้ในแผงวงจรรวม (Integrated Circuit) เพียงชิพเดี่ยวเหมาะสำหรับงานควบคุมอุปกรณ์อื่นๆแบบอัตโนมัติ เพราะผู้ใช้สามารถเขียนโปรแกรมควบคุมการทำงานได้ตามต้องการ

MCS-51 ผลิตโดยบริษัทอินเทล มีการทำงานเป็นแบบ 8 บิท หมายความว่า ส่วนที่ทำหน้าที่ในการคำนวณ (Arithmetic Logic Unit, ALU) จะทำงานสูงสุดทีละ 8 บิท

2.3.1 ข้อดีของ MCS-51

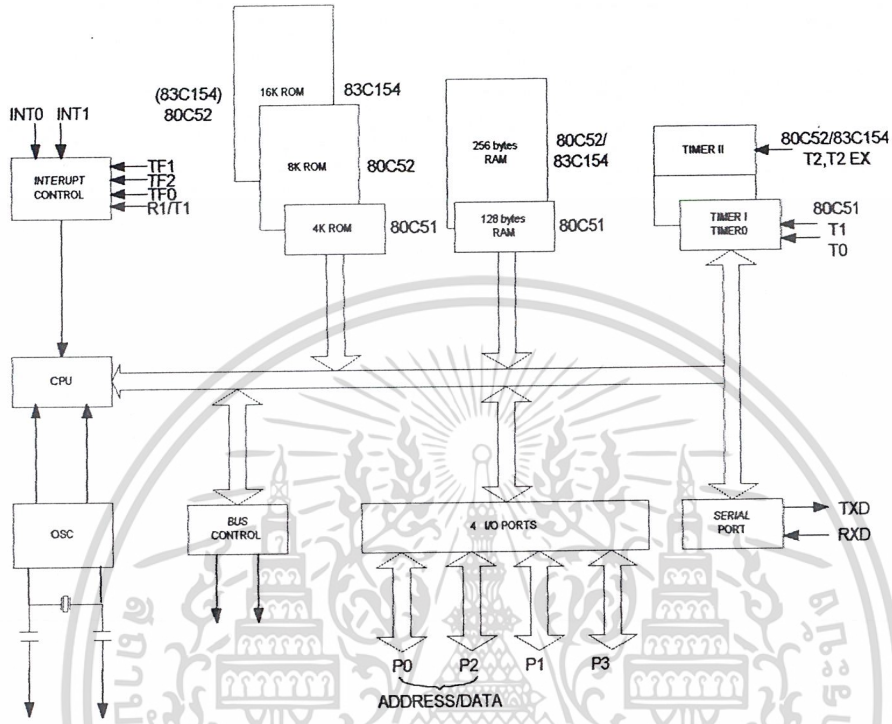
1. สามารถนำเอาข้อมูลมา AND, OR หรือทำ Complement ทั้งแบบทีละ 8 บิท และ 1 บิท
2. สามารถใช้กับหน่วยความจำสำหรับโปรแกรม (Program Memory) ซึ่งเป็นหน่วยความจำที่ใช้สำหรับเก็บชุดคำสั่งที่จะให้ MCS-51 ทำงานได้สูงสุด 64 กิโลไบท์ (Kilobyte) (64X1024 ไบท์) ทำให้เขียนโปรแกรมควบคุมการทำงานได้มาก
3. สามารถต่อกับหน่วยความจำสำหรับข้อมูล (Data Memory) ซึ่งเป็นหน่วยความจำสำหรับเก็บข้อมูลในระหว่างการทำงานของโปรแกรมได้สูงสุด 64 กิโลไบท์
4. ใน 8051 และ 8751 มีหน่วยความจำสำหรับโปรแกรมจำนวน 4 กิโลไบท์ (ใน 8052 และ 8752 มีหน่วยความจำสำหรับโปรแกรมจำนวน 8 กิโลไบท์) อยู่ภายในวงจรรวมทำให้ไม่ต้องต่อหน่วยความจำสำหรับโปรแกรมอยู่ภายนอก ระบบรวมทั้งหมดจึงมีขนาดเล็กและสัญญาณรบกวนจากภายนอกจะทำให้ MCS-51 ทำงานผิดพลาดได้ยาก
5. มีพอร์ทแบบขนาน (Parallel Port) สำหรับข้อมูลเข้าและออกจำนวน 32 บิทที่ข้อมูลแต่ละบิทเป็นอิสระต่อกัน
6. มีวงจร Timer/Counter ขนาด 16 บิท 2 ชุด (8052 มี 3 ชุด) ที่ทำงานในโหมดต่างๆได้ถึง 4 โหมด
7. มี Universal Asynchonous Receiver Transmitter (UART) สำหรับรับ-ส่งข้อมูลอนุกรม (Serial) แบบฟูลดูเฟลกซ์ (Full Duplex) ที่สามารถเลือกรูปแบบการรับ-ส่งข้อมูลได้ 4 แบบ
8. มีแหล่งกำเนิดสัญญาณขอขัดจังหวะการทำงานของโปรแกรม (Interrupt Request Signal) 6 แหล่ง ซึ่งสามารถทำกระโดดไปทำงานตอบสนองการขัดจังหวะ (Interrupt Service Routine) ได้ต่างๆกัน 5 ตำแหน่ง
9. สามารถเลือกการทำงานให้อยู่ในโหมดของ Idle และ Power Down ซึ่งจะประหยัดการใช้กำลังไฟในการทำงาน

ซึ่งจากข้อดีดังกล่าว จึงทำให้ MCS-51 เป็นที่นิยมนำมาใช้ในการควบคุมระบบอัตโนมัติมาก คุณสมบัติดังกล่าวบรรจุไว้ในวงจรรวมเดียว (Single Chip) ขนาด 40 ขา ดังนั้นจึงสามารถออกแบบให้ระบบทั้งหมดมีขนาดเล็ก และการที่ทั้งหมดบรรจุอยู่ภายในวงจรรวมเดียวจึงทำให้การตรวจสอบหาข้อผิดพลาดในระบบง่ายไม่สลับซับซ้อน รวมทั้งลดปัญหาเรื่องการที่มีสัญญาณรบกวนในระบบจนทำให้การทำงานผิดพลาดไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3.2 โครงสร้างของ 8051

ภายใน 8051 จะประกอบขึ้นด้วย GATE ต่างๆเช่น AND, OR, NOT ซึ่ง GATE เหล่านี้จะถูกนำเอามาออกแบบให้มีหน้าที่การทำงานต่างๆ เช่น วงจรถอดรหัสคำสั่ง (Intruction Decoder) , วงจรสร้างสัญญาณนาฬิกา (Clock Generator) โครงสร้างภายในของ 8051 จะประกอบด้วยส่วนย่อยๆ คือ



รูปที่ 2.6 แสดงโครงสร้างของ 8051

2.3.2.1 **หน่วยประมวลผล** (Central Processing Unit) ส่วนนี้จะมีวงจรที่ทำหน้าที่สร้างสัญญาณควบคุมในการติดต่อกับส่วนอื่นๆ เรียกว่าวงจรควบคุม (Control Unit) สัญญาณที่สร้างจากวงจรควบคุมได้แก่ สัญญาณสำหรับการติดต่อกับหน่วยความจำ, อุปกรณ์รับข้อมูลเข้าหรือส่งข้อมูลออกจากตัว 8051 ซึ่งส่วนควบคุมการขัดจังหวะ (Interrupt Control) และส่วนควบคุมบัส (Bus Control) ก็เป็นส่วนหนึ่งของวงจรควบคุมด้วย การสร้างสัญญาณควบคุมจากหน่วยประมวลผลนี้จะทำการสร้างสัญญาณโดยการถอดรหัสจากคำสั่ง (Instruction) ตามที่มีการกำหนดไว้ และสัญญาณที่สร้างขึ้นมาจะอ้างอิงกับสัญญาณนาฬิกาที่สร้างจากวงจรออสซิลเลเตอร์เพื่อให้ทุกๆส่วนในวงจรทำงานประสานกัน (Synchronize) อย่างถูกต้อง

ในหน่วยประมวลผลนี้ยังประกอบด้วยส่วนย่อยอีกส่วนที่เรียกว่า ส่วนประมวลผล ส่วนนี้จะทำหน้าที่ประมวลผลข้อมูล เช่น การบวก, ลบ, คูณ หรือหารข้อมูลแล้วนำผลลัพธ์ไปเก็บไว้ในรีจิสเตอร์หรือหน่วยความจำที่ต้องการ

2.3.2.2 **หน่วยความจำ** (Memory) มีไว้สำหรับจดจำข้อมูล การเอาข้อมูลไปเก็บในหน่วยความจำเรียกว่า การเขียน (Write) ข้อมูล และการเอาข้อมูลออกจากหน่วยความจำจะเรียกว่า การอ่าน (Read) ข้อมูล ซึ่งแต่ละตำแหน่งของหน่วยความจำจะเก็บข้อมูลได้เพียงค่าเดียวเท่านั้น ในไมโครโปรเซสเซอร์ทั่วไปรวมทั้ง 8051 นั้นข้อมูลในแต่ละตำแหน่งของหน่วยความจำจะมีค่าได้เพียง 8 หลักของเลขฐาน 2 (8

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บิทเท่ากับ 1 ไบท์) ดังนั้นแต่ละตำแหน่งของหน่วยความจำจะเก็บข้อมูลมีค่าได้ระหว่าง 0 ถึง 255 (00000000 ถึง 11111111 ในเลขฐาน 2) แต่จำนวนตำแหน่งที่จะเก็บข้อมูลได้ขึ้นกับไมโครโปรเซสเซอร์แต่ละเบอร์ การติดต่อกับหน่วยความจำจะต้องมีสัญญาณ 3 กลุ่ม คือ

- แอดเดรสหรือค่าตำแหน่งที่ต้องการติดต่อกับหน่วยความจำใน 8051 จะติดต่อกับหน่วยความจำประเภท Program Memory หรือ Dara Memory ได้สูงสุดชนิดละ 65536 ตำแหน่ง ดังนั้นการอ้างอิงแต่ละตำแหน่งของหน่วยความจำจะต้องใช้เส้นแสดงตำแหน่งในเลขฐาน 2 ทั้งหมด 16 เส้น (2^{16} เท่ากับ 64*1024 = 65536)

- ข้อมูลที่จะอ่านหรือเขียนกับหน่วยความจำที่ตำแหน่งในข้อ 1

- สัญญาณควบคุมที่จะส่งไปยังหน่วยความจำ เพื่อบอกกับหน่วยความจำว่าต้องการอ่านหรือเขียนข้อมูล

สัญญาณเหล่านี้จะถูกควบคุมภายใน 8051 สร้างมาจากวงจรถอดรหัสของคำสั่งที่ 8051 อ่านจากหน่วยความจำ Program Memory เข้าไปทำงานนั่นเอง

2.3.2.3 **อุปกรณ์อินพุทและเอาท์พุท** (Input/Output Device) เป็นส่วนที่จะใช้ส่งข้อมูลเข้าหรือออกจาก8051 ทำให้ 8051 ติดต่อกับภายนอกได้ ซึ่งประกอบด้วย

- พอร์ทอินพุท/เอาท์พุท 4 พอร์ท เป็นที่ใช้สำหรับรับ-ส่งข้อมูลซึ่งเป็นสัญญาณดิจิตอลเข้าหรือออกจากตัว MCS-51 โดยแต่ละพอร์ทจะรับ-ส่งข้อมูลได้ 8 บิท มีพอร์ท P0, P1, P2 และ P3 บางพอร์ทจะใช้ทำงานมากกว่า 1 อย่างก็ได้ เช่น พอร์ท P0 และ P2 จะใช้สำหรับการส่งค่าตำแหน่ง (Address) ของหน่วยความจำที่ต้องการติดต่อและพอร์ท P0 จะใช้รับ-ส่งข้อมูลเมื่อติดต่อกับสหน่วยความจำได้ด้วย แต่สิ่งเหล่านี้ไม่ได้เกิดขึ้นที่เวลาเดียวกัน แต่จะใช้วิธีทำงานตามลำดับโดยควบคุมจากสัญญาณควบคุมที่ถอดรหัสมาจากแต่ละคำสั่งที่ให้คอมพิวเตอร์ทำงานนั่นเอง และสัญญาณทั้งหมดจะอ้างอิงจากสัญญาณนาฬิกา

- Timer 0 และ Timer 1 เป็นวงจรนับที่สามารถกำหนดให้ทำการนับจำนวนไซเคิลของสัญญาณที่ต่อจากภายนอก 8051 หรือจำนวนไซเคิลของสัญญาณนาฬิกาภายใน 8051 ก็ได้ค่าจากการนับจะถูกอ่านหรือตั้งค่าเริ่มต้นของการนับได้โดยหน่วยประมวลผล

- พอร์ทอนุกรม (Serial Port) หน่วยประมลผลจะอ่านและเขียนข้อมูลกับพอร์ทอนุกรมเป็นแบบ 8บิท แต่ข้อมูลจะถูกส่งออกจาก 8051 เรียงไปทีละบิทออกจากขา TxD และในการรับข้อมูลเข้าก็จะรับเข้ามาทีละบิททางขา RxD แล้วจัดเรียงใหม่เป็น 8 บิท เพื่อให้หน่วยประมวลผลอ่านไปใช้งานต่อไป

2.3.3 <u>การจัดการหน่วยความจำของ 8051</u>

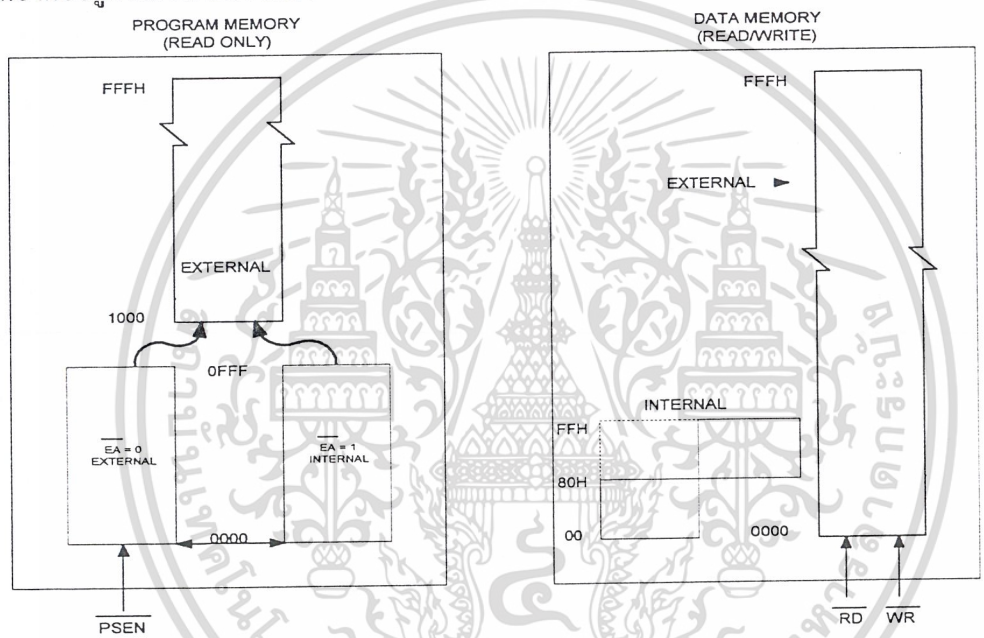
หน่วยความจำของ 8051 แบ่งออกได้เป็น 2 แบบ ตามลักษณะของการใช้งาน คือ

2.3.3.1 **หน่วยความจำโปรแกรม** (Program Memory) เป็นหน่วยความจำที่ใช้เก็บคำสั่งในรูปรหัสภาษาเครื่อง (Machine Language) ซึ่งต้องการให้ 8051 ทำงาน เมื่อ 8051 ทำงานก็จะอ่านข้อมูลที่เก็บในหน่วยความจำประเภทนี้เข้าไปถอดรหัสแล้วสร้างสัญญาณควบคุมส่วนอื่นๆ ตามการทำงานของแต่ละคำสั่งนั้น หน่วยความจำแบบนี้จะต้องเป็นแบบ Read Only Memory (ROM) และผู้ใช้ต้องเขียนข้อมูลในแต่ละตำแหน่งของหน่วยความจำเป็นรหัสภาษาเครื่องของ 8051 ตามลำดับการทำงานที่ต้องการ (หน่วยความจำแบบ ROM เป็นแบบ Non volatite ซึ่งเมื่อปิดไฟแล้วข้อมูลก็ไม่มีการสูญหาย) การเขียนข้อมูลไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บน ROM จะต้องใช้เครื่องมือพิเศษ ในระหว่างการทำงานของ 8051 ผู้ใช้จะไม่สามารถใช้คำสั่งทำการเขียนข้อมูลลงในหน่วยความจำแบบนี้ได้ จำนวนตำแหน่งสูงสุดของหน่วยความจำแบบนี้ที่ 8051 จะใช้งานได้คือ 65536 ตำแหน่ง ค่าของตำแหน่งจะเขียนเป็นเลขฐาน 16 ได้ตั้งแต่ 0000H ถึง FFFFHหน่วยความจำตำแหน่ง 0000H ถึง FFFFH จำนวน 4 กิโลไบท์ ผู้ใช้จะเลือกได้ว่าเป็นตำแหน่งของ ROM ที่อยู่ภายในหรือภายนอก 8051

ถ้าต้องการให้ 8051 ทำงานตามคำสั่งที่เก็บไว้ใน ROM ภายใน 8051 ก็ให้ป้อนสัญญาณสภาวะลอจิก High (1) เข้าที่ขา EA ของ 8051 แต่ถ้าต้องการให้ทำงานในโปรแกรมที่เก็บไว้ใน ROM ภายนอก 8051 ก็ให้ต่อลอจิก Low (0) เข้าที่ขา EA ของ 8051 ส่วนหน่วยความจำตำแหน่ง 1FFFH ถึง FFFFH จะต้องต่ออยู่ภายนอก 8051 เสมอ



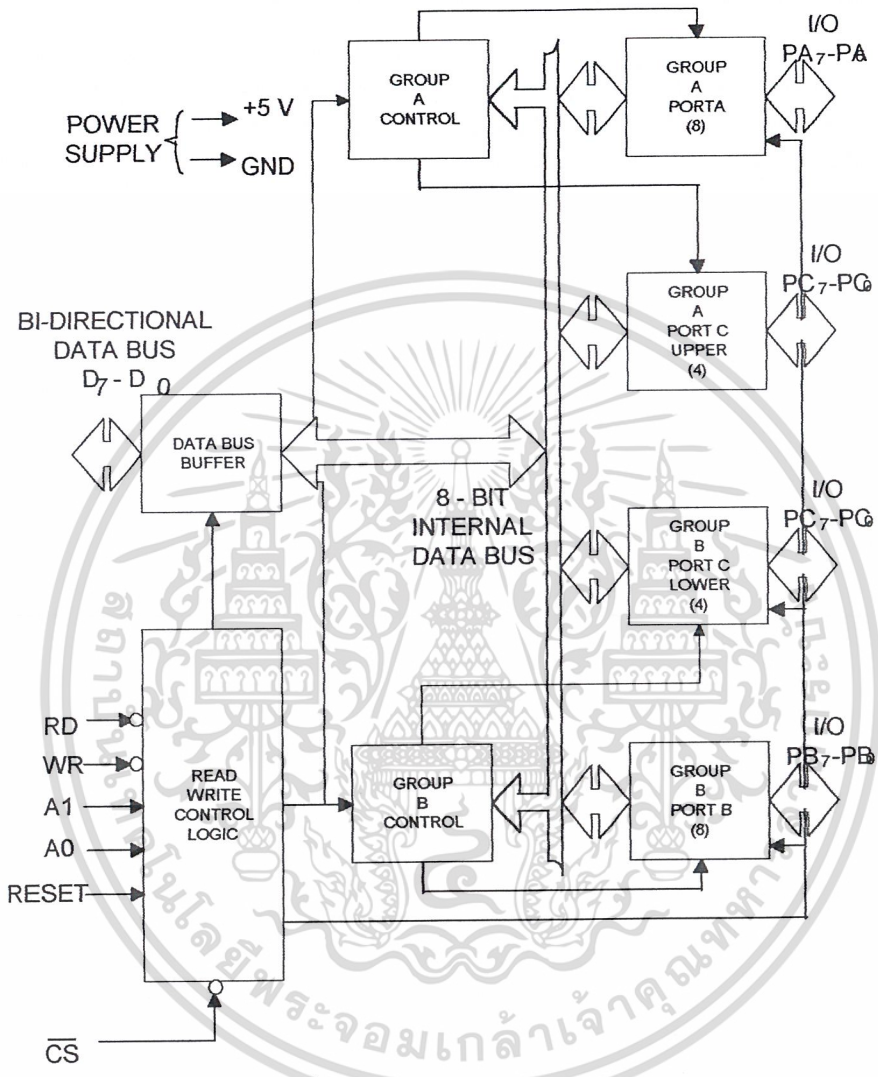
รูปที่ 2.7 แสดงแผนภูมิหน่วยความจำของ 8051

2.3.3.2 **หน่วยความจำข้อมูล** (Data Memory) เป็นหน่วยความจำที่ 8051 จะใช้สำหรับพัก เก็บข้อมูล แล้วเรียกใหม่ในระหว่างการทำงานของ 8051 การอ่านหรือเขียนข้อมูลจากหน่วยความจำจะกระทำโดยคำสั่งที่เก็บไว้ในหน่วยความจำโปรแกรม หน่วยความจำแบบนี้เป็นประเภท Random Access Memory (RAM) ถ้ามีไฟเลี้ยงอยู่ข้อมูลที่เก็บไว้จะไม่สูญหาย แต่ถ้าปิดเครื่องหรือไม่จ่ายไฟให้แก่ RAM แล้ว ข้อมูลใน RAM ก็จะสูญหายไป การสูญหายของข้อมูลไม่ได้หมายความว่าไม่มีอะไรอยู่เลยแต่เป็นการที่มีข้อมูลใหม่ซึ่งไม่ใช่ข้อมูลที่เก็บไว้เดิมเข้ามาอยู่แทนที่ หน่วยความจำข้อมูลของ 8051 จะมีอยู่ 2 ชุด ชุดหนึ่งอยู่ภายใน 8051 จำนวน 128 ไบท์ ที่ตำแหน่ง 00H ถึง 7FH และอีกชุดหนึ่งจะต้องต่ออยู่ภายนอกของวงจร รวม 8051 มีได้สูงสุด 65536 ไบท์ (64 กิโลไบท์) อยู่ที่ตำแหน่ง 0000H ถึง FFFFH หน่วยความจำข้อมูลภายใน 8051 ที่ตำแหน่ง 80H ถึง FFH นั้นไม่ได้มีอยู่ทุกตำแหน่ง จะมีเฉพาะในบางตำแหน่งซึ่งเรียกหน่วยความจำบางตำแหน่งนี้ว่า Special Function Register (SFR) เพราะจะใช้หน่วยความจำเหล่านี้สำหรับงานพิเศษเท่านั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4 8255 PROGRAMMABLE PERIPHERAL INTERFACE

เป็นชิพขนาด 40 ขา มี 3 พอร์ทคือ A, B, C เป็นพอร์ท 8 บิทที่สามารถโปรแกรมให้เป็นอินพุท หรือเอาท์พุทก็ได้ โดยที่พอร์ท C ยังแบ่งเป็น 4 บิทล่างและ 4 บิตบน โดยมีโครงสร้างตามรูปที่ 2.8



รูปที่ 2.8 แสดงโครงสร้างของ 8255

2.4.1 โหมดการทำงาน

โหมด 0 มีการทำงานแบบเบสิคอินพุท/เอาท์พุท และไม่มีการตรวจสอบสัญญาณ(handshake)

โหมด 1 โหมดนี้ใช้พอร์ท A, B ในการรับหรือส่งข้อมูล และใช้พอร์ท C ในการตรวจสอบสัญญาณ

โหมด 2 โหมดนี้ใช้พอร์ท A ในการรับส่งข้อมูล 2 ทิศทางและพอร์ท B ในการรับหรือส่งข้อมูล และใช้พอร์ท C บิท 1, 2 ในการรับส่งข้อมูลบิทและบิท 4, 5, 6 เป็นสัญญาณที่ใช้ในการตรวจสอบสัญญาณ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

	โหมด 0	โหมด0	โหมด 1	โหมด1	โหมด 2
	เข้า	ออก	เข้า	ออก	กลุ่ม A เท่านั้น
PA_0	เข้า	ออก	เข้า	ออก	←→
PA_1	เข้า	ออก	เข้า	ออก	←→
PA_2	เข้า	ออก	เข้า	ออก	←→
PA_3	เข้า	ออก	เข้า	ออก	←→
PA_4	เข้า	ออก	เข้า	ออก	←→
PA_5	เข้า	ออก	เข้า	ออก	←→
PA_6	เข้า	ออก	เข้า	ออก	←→
PA_7	เข้า	ออก	เข้า	ออก	←→
PB_0	เข้า	ออก	เข้า	ออก	———
PB_1	เข้า	ออก	เข้า	ออก	———
PB_2	เข้า	ออก	เข้า	ออก	———
PB_3	เข้า	ออก	เข้า	ออก	———
PB_4	เข้า	ออก	เข้า	ออก	———
PB_5	เข้า	ออก	เข้า	ออก	———
PB_6	เข้า	ออก	เข้า	ออก	———
PB_7	เข้า	ออก	เข้า	ออก	———
PC_0	เข้า	ออก	$INTR_B$	$INTR_B$	I/O
PC_1	เข้า	ออก	IBF_B	OBF_B	I/O
PC_2	เข้า	ออก	STB_B	ACK_B	I/O
PC_3	เข้า	ออก	$INTR_A$	$INTR_A$	INTRA
PC_4	เข้า	ออก	STB_A	I/O	STBA
PC_5	เข้า	ออก	IBF_A	I/O	IBFA
PC_6	เข้า	ออก	I/O	ACK_A	ACKA
PC_7	เข้า	ออก	I/O	OB_F	OBFA

ตารางที่ 2.1 แสดงโหมดต่าง ๆ ของ 8255

สัญญาณต่างๆ ของ 8255

D7 - D0 บัสข้อมูลเชื่อมโยงกับซีพียู

A1 - A0 ใช้เลือกพอร์ท A, B, C และพอร์ทควบคุม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

RESET เมื่อขานี้ได้รับสัญญาณกระตุ้นลอจิก 1 จะทำให้ 8255 ถูกรีเซทมีผลทำให้ทุกพอร์ทเป็นอินพุททันที

PA7 - PA0 เป็นพอร์ทขนาน 8 บิท

PB7 - PB0 เป็นพอร์ทขนาน 8 บิท

PC7 - PC0 เป็นพอร์ทขนาน 8 บิท

$\overline{RD}$ ในการอ่านข้อมูลที่พอร์ทของ 8255 ต้องทำให้ขานี้เป็นลอจิก 0 พร้อมกับ $\overline{CS}$

$\overline{WR}$ ในการเขียนข้อมูลหรือโปรแกรมลงบน 8255 ต้องทำให้ขานี้เป็นลอจิก 0 พร้อมกับ $\overline{CS}$

$\overline{CS}$ เป็นขาเลือกชิพ 8255 ได้ ขานี้มักจะต่อกับตัวถอดรหัสอินพุท/เอาท์พุท

$\overline{WR}$, $\overline{RD}$, A0, A1, $\overline{CS}$ ทำงานทั้ง 5 ขาจะมีฟังก์ชันการทำงานดังตารางที่ 2.2

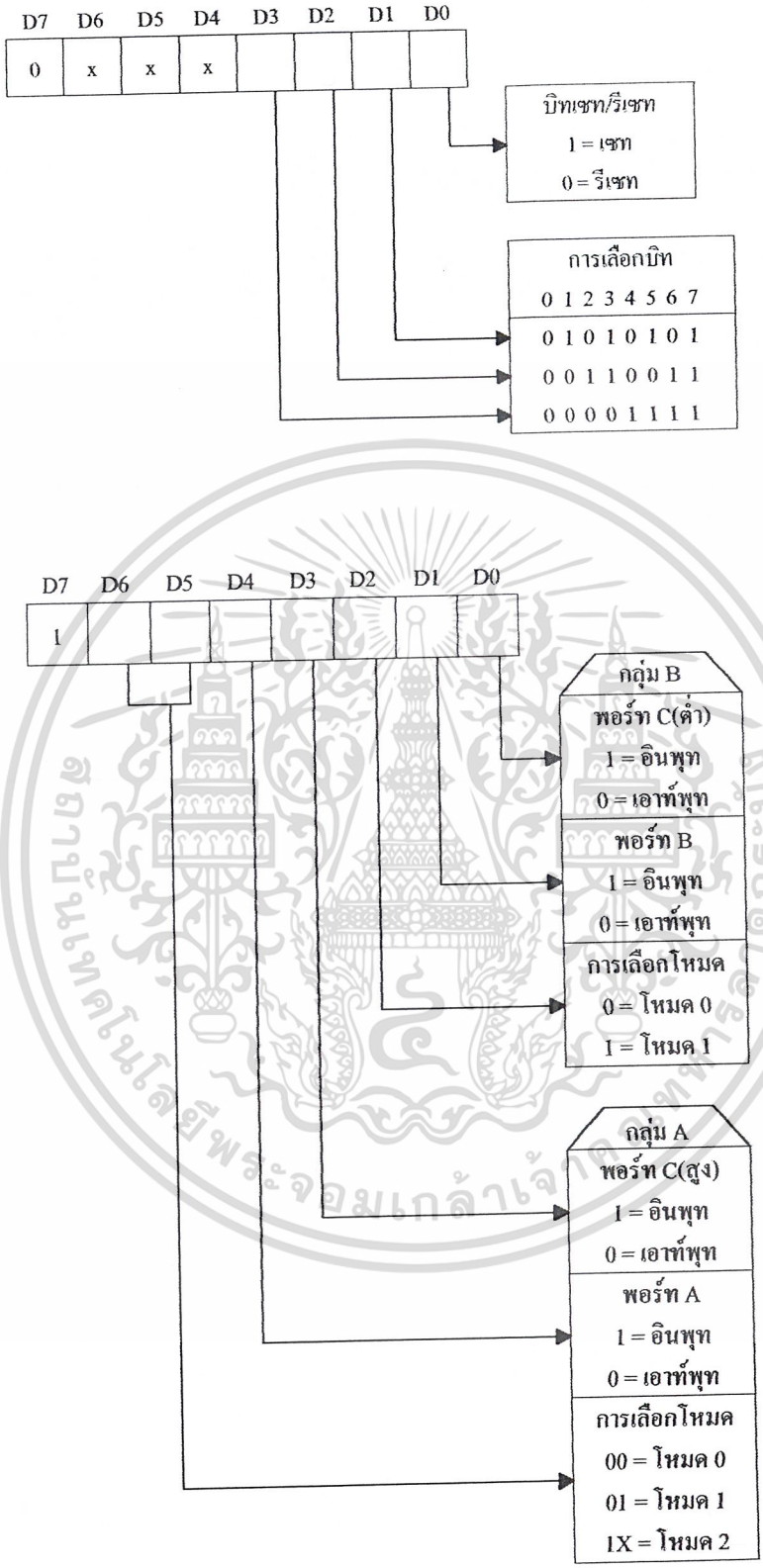
$\overline{A1}$	$\overline{A0}$	$\overline{RD}$	WR	CS	
					การเซทค่าเพื่อทำการอ่าน
0	0	0	1	0	พอร์ท A → บัสข้อมูล
0	1	0	1	0	พอร์ท B → บัสข้อมูล
1	0	0	1	0	พอร์ท C → บัสข้อมูล
					การเซทค่าเพื่อทำการเขียน
0	0	1	0	0	บัสข้อมูล→ พอร์ท A
0	1	1	0	0	บัสข้อมูล→ พอร์ท B
1	0	1	0	0	บัสข้อมูล→ พอร์ท C
1	1	1	0	0	บัสข้อมูล→ ควบคุม
					ฟังก์ชันที่ไม่สามารถทำงานได้
X	X	X	X	1	บัสข้อมูล→ 3 สเตจ
1	1	0	1	0	สภาวะผิดปกติ
X	X	1	1	0	บัสข้อมูล→ 3 สเตจ

ตารางที่ 2.2 ตารางความจริงของ 8255

2.4.2 <u>การ โปรแกรม 8255</u>

จะใช้ตารางการ โปรแกรมดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

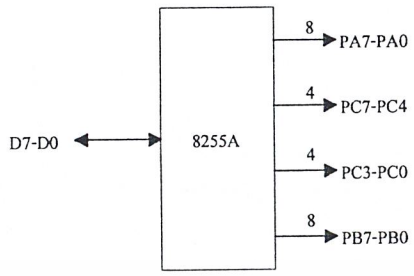


รูปที่ 2.9 รูปแบบการควบคุมของโหมดและบิท

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

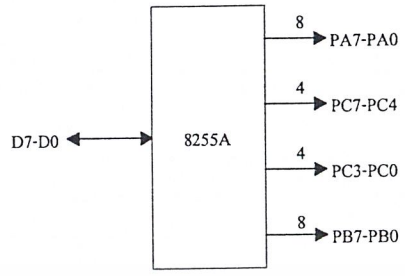
control word # 0

D7	D6	D5	D4	D3	D2	D1	D0
1	0	0	0	0	0	0	0



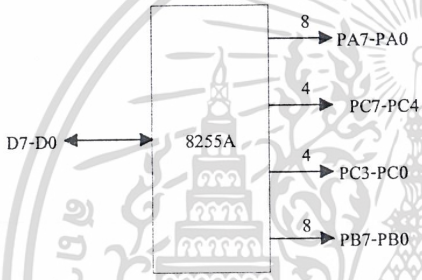
control word # 1

D7	D6	D5	D4	D3	D2	D1	D0
1	0	0	0	0	0	0	1



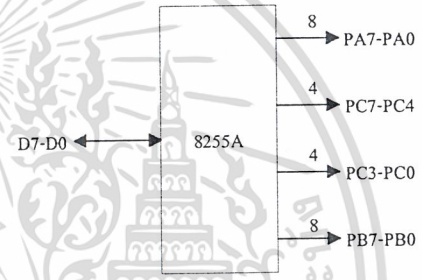
control word # 2

D7	D6	D5	D4	D3	D2	D1	D0
1	0	0	0	0	0	1	0



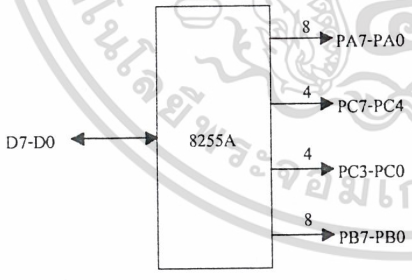
control word # 3

D7	D6	D5	D4	D3	D2	D1	D0
1	0	0	0	0	0	1	1



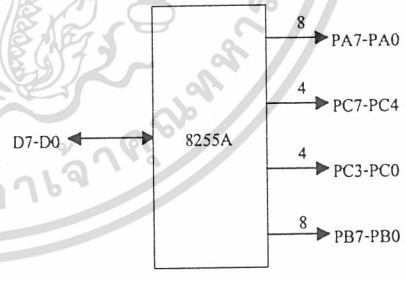
control word # 4

D7	D6	D5	D4	D3	D2	D1	D0
1	0	0	0	1	0	0	0



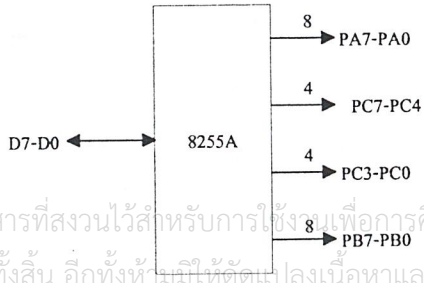
control word # 5

D7	D6	D5	D4	D3	D2	D1	D0
1	0	0	0	1	0	0	1



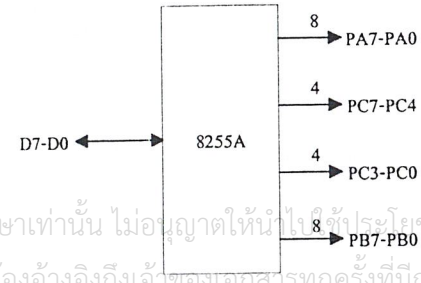
control word # 6

D7	D6	D5	D4	D3	D2	D1	D0
1	0	0	0	1	0	1	0

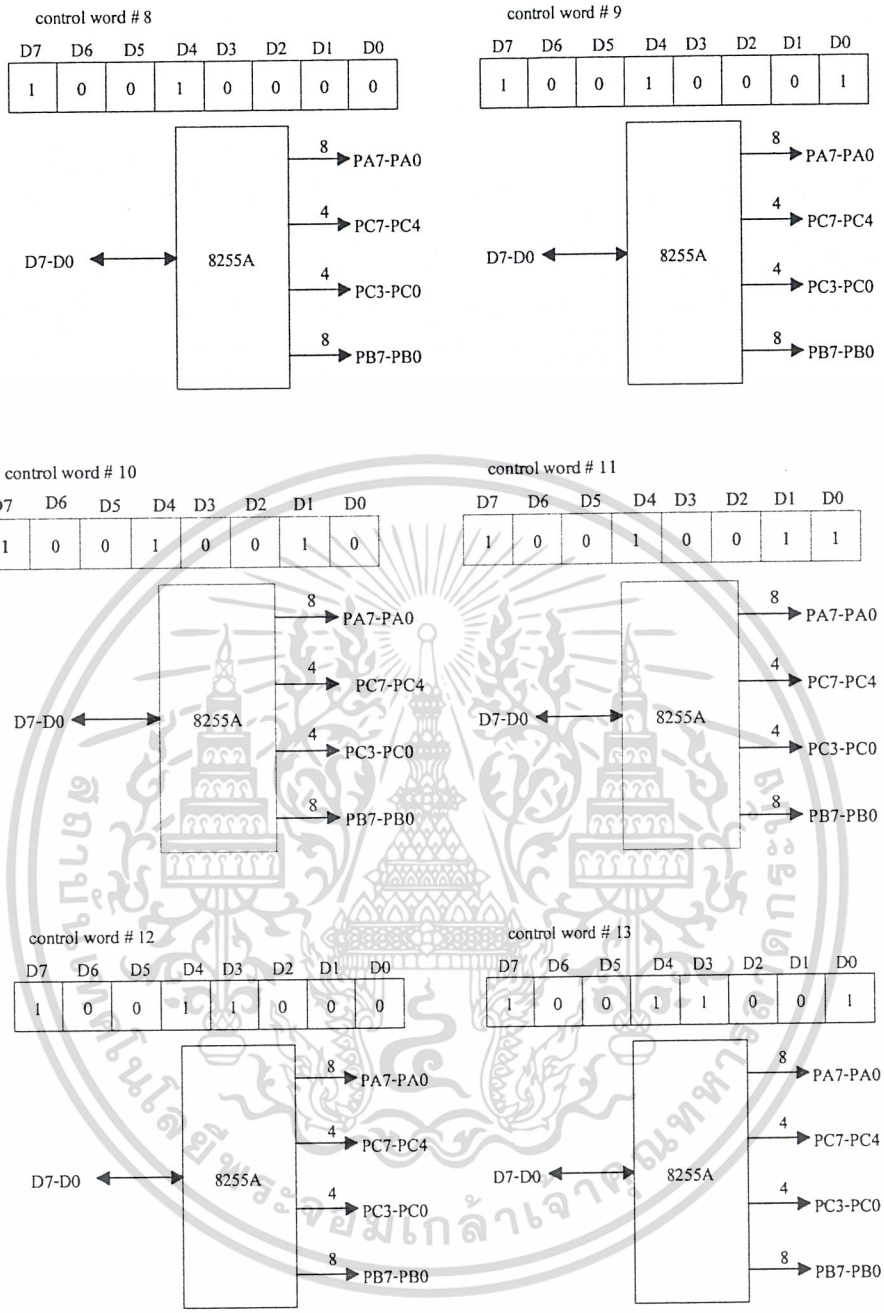


control word # 7

D7	D6	D5	D4	D3	D2	D1	D0
1	0	0	0	1	0	1	1



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 2.10 การให้ค่าเวิร์ดควบคุมและการกำหนดพอร์ท A, B และ C

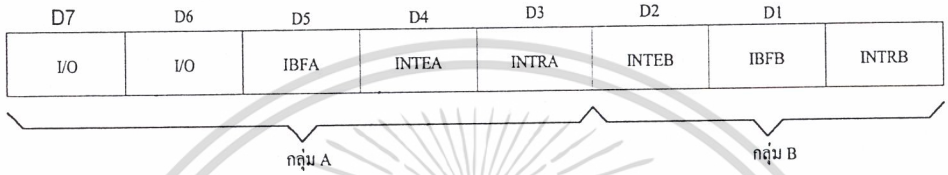
2.4.3 <u>โหมดการเซท/รีเซทบิท</u>

นอกจากเราใช้พอร์ท A,B,C ในการโปรแกรมให้เป็นอินพุท/เอาท์พุทแล้วเรายังสามารถที่จะโปรแกรมพอร์ท C บิต PC0-PC7 ให้เป็นลอจิก 0 หรือ 1 (ใช้งานเป็นเอาท์พุท)เพื่อใช้เป็นสัญญาณสโตรบ (Strobe) ได้

โหมด 1

โหมดนี้จะใช้พอร์ท A,B ในการส่งข้อมูล และใช้พอร์ท C ตรวจสอบความพร้อม เมื่อโปรมแกรม (โหมด 1) แล้ว พอร์ท C จะระบุเป็นขาสัญญาณดังต่อไปนี้

เมื่อเป็นอินพุทพอร์ท



รูปที่ 2.11 การกำหนดทางขาอินพุท

$PC0 = INTR_B$

$PC1 = IBF_B$

$PC2 = INTE_B$

$PC3 = INTR_A$

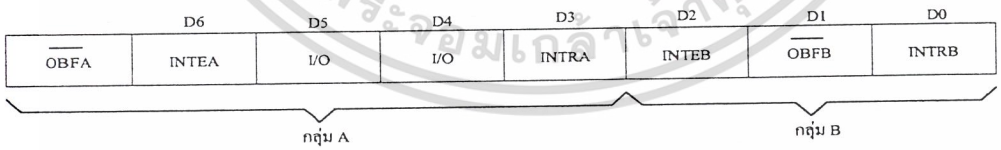
$PC4 = INTE_A$

$PC5 = IBF_A$

PC6 = I/O

PC7 = I/O

เมื่อเป็นเอาท์พุทพอร์ท



รูปที่ 2.12 การกำหนดขาเอาท์พุท

$PC0 = INTR_B$

$PC1 = OBF_B$

$PC2 = INTE_B$

$PC3 = INTR_A$

PC4 = I/O

PC5 = I/O

$PC6 = INTE_A$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$PC7 = OBF_A$

โดยที่ INTR คือ การร้องขออินเตอร์รัพท์ (Interrupt Request)

IBF คือ บัฟเฟอร์เต็ม (Input Buffer Full)

STB คือ สโตรบ (Strobe)

I/O คือ อินพุท/เอาท์พุท (Input/Output) โปรแกรมได้

INTE คือ อินเตอร์รัพท์ อีเนเบิล (Interrupt Enable)

ACK คือ แอคโนเลดจ์ (Acknowledge)

พอร์ท	ชนิด	ชื่อ	ควบคุมโดยบิท
A	อินพุท	INTEA	PC4
A	เอาท์พุท	INTEA	PC6
B	อินพุท	INTEB	PC2
B	เอาท์พุท	INTEB	PC2

ตารางที่ 2.3 แสดงการโปรแกรม INTE ของพอร์ท A, B

2.5 พอร์ทอนุกรม (Serial Port)

พอร์ทอนุกรม เป็นอุปกรณ์ที่ใช้รับส่งข้อมูลที่ถูกนำมาใช้งานอย่างมาก เพราะมีค่าใช้จ่ายถูกกว่าเมื่อเทียบกับพอร์ทขนาน (parallel port) แต่จะสร้างปัญหายุ่งยากให้แก่ผู้เขียนโปรแกรม เนื่องจากมีการทำงานและใช้งานที่ยากกว่าพอร์ทขนาน

2.5.1 <u>การรับส่งข้อมูลแบบอนุกรมอะซิงโครนัส</u>

ในการสื่อสารแบบอะซิงโครนัส ข้อมูลจะถูกส่งผ่านพอร์ทแบบอนุกรมครั้งละ 1 บิท ซึ่งจะแตกต่างจากการส่งแบบขนานที่จะส่งครั้งละ 1 ไบท์ และระยะเวลาที่ใช้ในการส่งข้อมูลแบบอะซิงโครนัสแต่ละไบท์นั้นไม่จำเป็นต้องเท่ากัน จึงเรียกว่า การรับส่งข้อมูลแบบอะซิงโครนัส

ในการส่งข้อมูลผ่านพอร์ทแบบอนุกรมนั้น ข้อมูลแต่ละไบท์จะประกอบด้วย

1. บิทเริ่มต้น (Start bit) 1 บิท
2. บิทข้อมูล (Data bit) 7 หรือ 8 บิท
3. พาริตี้บิท (Parity bit) จะมีหรือไม่มีก็ได้
4. บิทสิ้นสุด (Stop bit) 1 หรือ 2 บิท

สถานะของสายส่งในขณะที่ไม่มีข้อมูลจะมีสถานะเป็นสูง (สถานะทางดิจิตอลมี 2 สถานะ คือ สูง(high) และต่ำ (low)) ข้อมูลบิทใดมีค่า 0 จะทำให้สายส่งมีสถานะต่ำ ข้อมูลบิทใดมีค่า 1 ก็จะทำให้สายส่งมีสถานะสูงอยู่เช่นเดิม บิทเริ่มต้นใช้สำหรับบอกจุดเริ่มต้นของไบท์ของข้อมูล โดยการทำให้สถานะของสายส่งมีค่าต่ำ เป็นเวลา 1 รอบ (cycle) จากนั้นจะเป็นบิทของข้อมูล ตามด้วยพาริตี้บิท ซึ่งจะมีหรือไม่ก็ได้ สุดท้ายคือ บิทสิ้นสุด ซึ่งจะมี 1 หรือ 2 บิทก็ได้ ขึ้นกับว่าจะใช้เท่าใด

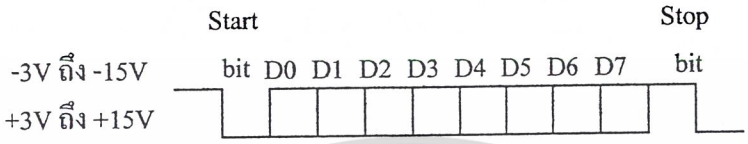
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

พาริตี้บิท ถ้าหากมีในไบท์ข้อมูล ก็จะทำหน้าที่ตรวจเช็คความผิดพลาดของข้อมูล พาริตี้มีค่า 2 อย่าง คือ เป็นคู่หรือคี่ (even or odd) ถ้าเป็นคู่ หมายความว่า เมื่อรวมพาริตี้บิทแล้ว จำนวนของบิทข้อมูลที่

อัตราการส่งข้อมูลมีหน่วนเป็นบอด (baud;bit per second) ค่าอัตราบอด (baud rate) ที่ต่ำที่สุดที่มีใช้กันคือ 300 บอด ซึ่งจะใช้กับโมเด็มรุ่นเก่า (โมเด็มรุ่นใหม่มักจะใช้ 1200-2400 บอด) ส่วนเครื่องคอมพิวเตอร์ระดับ IBM PC สามารถใช้ค่าอัตราบอดได้สูงถึง 9600 บอด

2.5.2 มาตรฐาน RS-232

การที่จะเข้าใจว่าปัญหามากมายที่เกิดกับพอร์ทแบบอนุกรมนั้นเกิดได้อย่างไร และทำไมถึงเกิดขึ้นได้ จะต้องเข้าใจมาตรฐานของการสื่อสารแบบอนุกรมอะซิงโครนัสของ RS-232 มากพอสมควร



รูปที่ 2.13 ลักษณะของสัญญาณตามมาตรฐาน RS-232

บอดเรท = 1200, 2400, 4800, 9600

ข้อมูล = 8 บิท

บิทสต็อป = 1 บิท

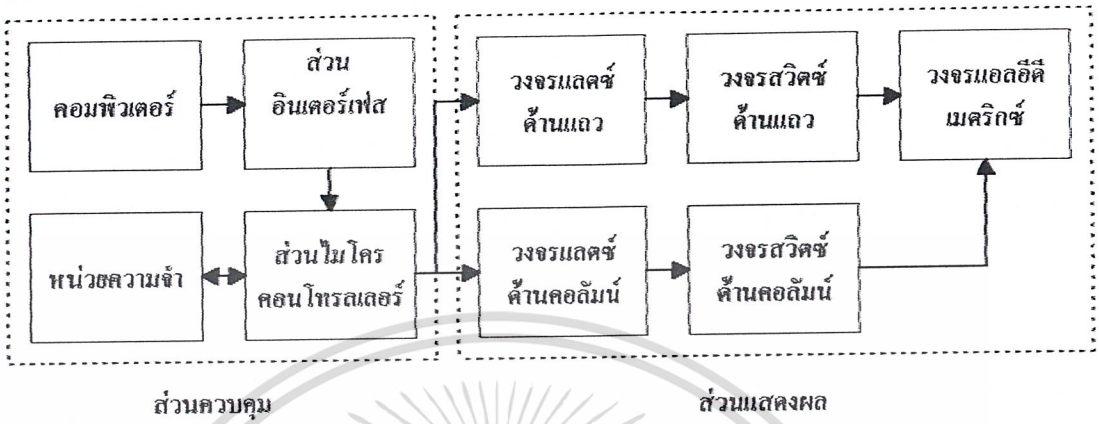
บิทพาริตี้ = ไม่มี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3

การคำนวณและการสร้าง

3.1 บล็อกไดอะแกรมของจอแสดงผล

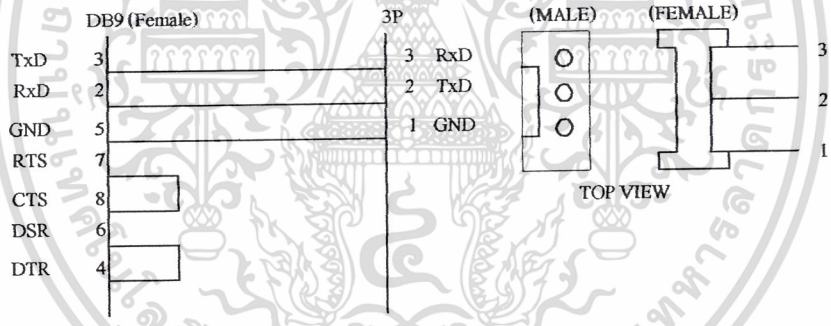


รูป ที่ 3.1 แสดงบล็อกไดอะแกรมของจอแสดงผล

3.2 ส่วนควบคุม (Control)

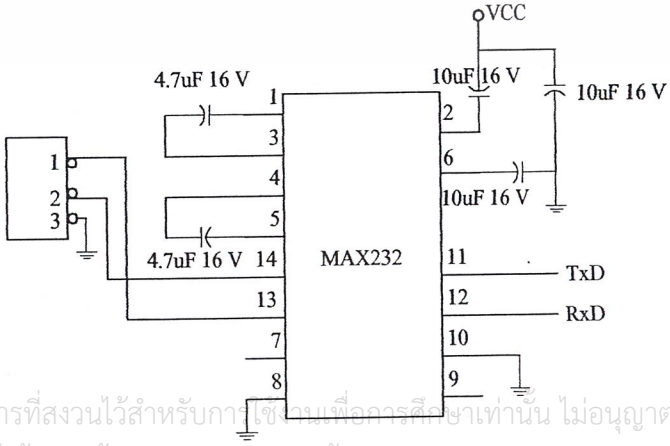
3.2.1 อุปกรณ์เชื่อมต่อคอมพิวเตอร์และไมโครคอนโทรเลอร์

ใช้การส่งข้อมูลผ่านทางพอร์ทอนุกรมของคอมพิวเตอร์ (dB9)ในรูปมาตรฐาน RS232



รูปที่ 3.2 แสดงการต่อขาสัญญาณของ dB9

มีวงจรดังรูปที่ 3.3 จะเห็นว่าขา RxD และ TxD จะต่อจากขาของ 8951 ผ่านไอซี MAX232 เพื่อปรับระดับแรงไฟ ในการรับส่งข้อมูลในโครงงานนี้จะใช้อัตราเร็ว 9600 บอดเรท

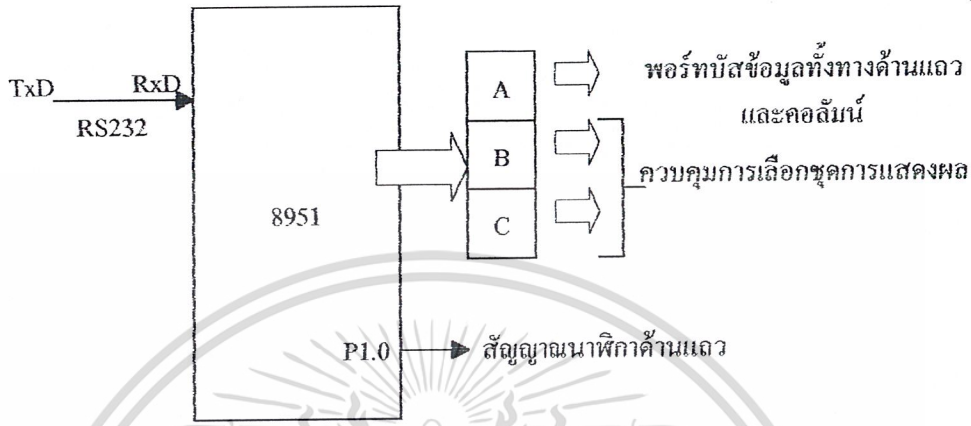


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 3.3 แสดงรูปวงจร MAX232

3.2.2 ส่วนประมวลผล

ในที่นี้จะใช้ไมโครคอนโทรลเลอร์ตระกูล MCS-51 เบอร์ 8951 ควบคุมการทำงานของส่วนขับกระแสและแสดงผล โดยผ่านทางพอร์ท 1 และพอร์ท A, B, C ของ 8255 ซึ่งช่วยในการขยายเอาท์พุทให้มากขึ้น การทำงานมีลักษณะดังนี้



รูปที่ 3.4 แสดงการทำงานของ 8951

3.2.3 หน่วยความจำ (Memory) ประกอบด้วย

3.2.3.1 หน่วยความจำสำหรับเก็บโปรแกรม (Program Memory)

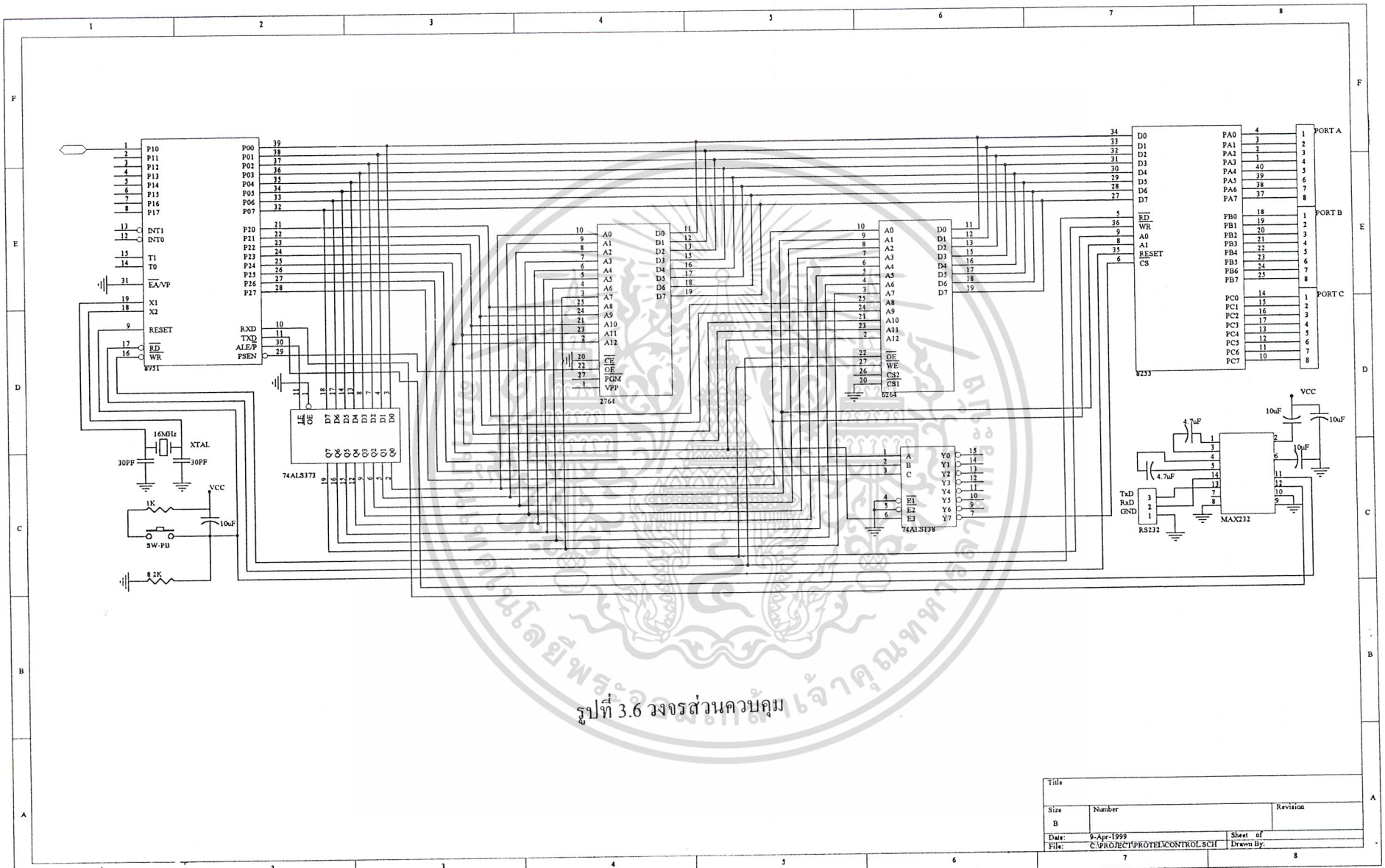
เป็นหน่วยความจำที่ใช้เก็บโปรแกรมสั่งงาน บรรจุอยู่ในไอซี 8951 ขนาด 4 กิโลไบท์ โปรแกรมที่ใช้งานนี้เขียนด้วยภาษาแอสแซมบลี้ (Assembly) เพื่อให้ไมโครคอนโทรลเลอร์ทำงานเป็นไปตามขั้นตอนในการแสดงผล

3.2.3.2 หน่วยความจำสำหรับเก็บข้อมูล (Data Memory)

หรือเรียกว่าแรม (RAM) แบ่งเป็น 2 ส่วน คือ หน่วยความจำข้อมูลภายในไอซีมีเพียง 128 ไบท์ และหน่วยความจำภายนอก มีความจุ 64 กิโลไบท์ แต่ในชิ้นงานนี้ใช้ขนาด 32K เพื่อเป็นหน่วยความจำชั่วคราว ได้แก่ เป็นบัฟเฟอร์ในการแสดงผล เป็นพื้นที่ที่เก็บรหัสตัวอักษรที่รับเข้ามา หรือไม่ว่าจะเป็นรหัสแสดงผลที่แปลงจากรหัสตัวอักษร และอื่นๆ ซึ่งในการเขียนโปรแกรมเพื่อติดต่อกับหน่วยความจำภายนอกนี้ต้องใช้คำสั่งแตกต่างจากหน่วยความจำภายใน ซึ่งรายละเอียดในการเขียนโปรแกรมควบคุมจอแสดงผลนี้จะกล่าวในหัวข้อถัดๆไป

รูปที่ 3.5 แสดงหน่วยความจำสำหรับเก็บโปรแกรมภายในและภายนอก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.6 วงจรส่วนควบคุม

3.2.4 การต่อขาเอาท์พุทจากตัวไมโครคอนโทรลเลอร์

3.2.4.1 ใช้พอร์ท 1 บิท 0 เป็น สัญญาณนาฬิกา(Clock) ในการแลตซ์ข้อมูลทางด้านแถว

3.2.4.2 ใช้ 8255 ในการขยายเอาท์พุท เป็นพอร์ท A, B และ C โดยมีตำแหน่ง 0F00H

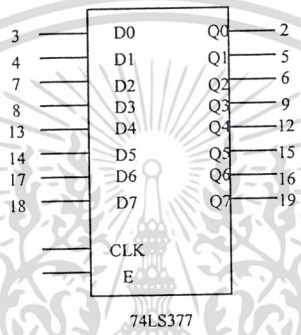
- พอร์ท A ใช้ในการส่งข้อมูลทั้งทางด้านแถวและคอลัมน์
- พอร์ท B และพอร์ท C ใช้เป็นสัญญาณนาฬิกาในการแลตซ์ข้อมูลทางด้านคอลัมน์

โดยแต่ละ 1 บิทจะควบคุมแต่ละกลุ่มจอแสดงผล (แอลอีดี ขนาด 8X8)

3.3 ส่วนแสดงผล

3.3.1 วงจรแลตซ์ด้านแถว

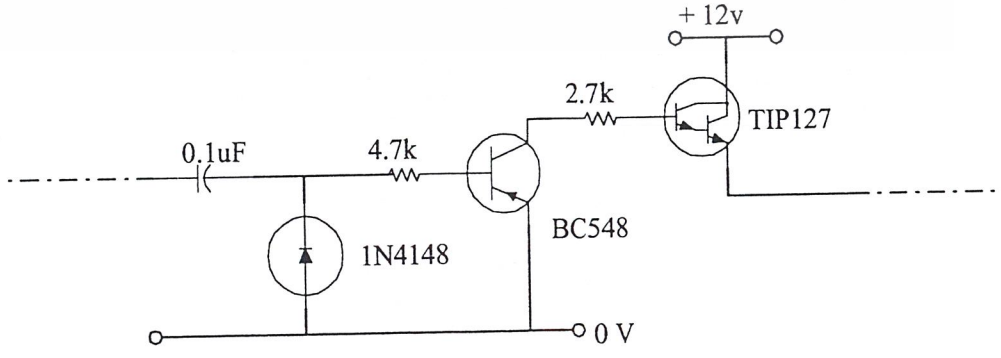
ทำหน้าที่รับข้อมูลควบคุมจากบัสข้อมูลของระบบ ใช้สำหรับควบคุมวงจรสวิตซ์ด้านแถว



รูปที่ 3.7 แสดงวงจรแลตซ์ด้านแถว

3.3.2 วงจรสวิตซ์ด้านแถว

ทำหน้าที่จ่ายกระแสไฟจากแหล่งจ่ายไฟ เพื่อส่งผ่านไปยังวงจรแอลอีดีเมตริกซ์ ให้หลอดแอลอีดีติด โดยมีวงจรดังรูปที่ 3.7 ประกอบด้วย 8 วงจรที่เหมือนกันทุกประการ เพื่อใช้ขับแอลอีดีแต่ละแถว โดยจะมีการทำงานดังนี้ สัญญาณเอาท์พุทจากวงจรแลตซ์จะถูกป้อนผ่าน ตัวเก็บประจุ 0.1uF และตัวต้านทาน 4.7k ไปยังขาเบสของ BC548 ซึ่งจะทำงานก็ต่อเมื่อเอาท์พุทมีสถานะเป็นไฮ (High) ที่ขาเบสของ BC548 จะคงสถานะอยู่ได้เพียง 5 มิลลิวินาทีเท่านั้น แล้วแรงดันที่ขาเบสจะลดลงเหลือเป็น 0 โวลท์อีกครั้ง ทำให้ BC548 หยุดทำงาน ส่งผลให้สวิตซ์ด้านแถวไม่ต้องทำงานตลอดเวลา เพื่อป้องกันไม่ให้แอลอีดีแสดงผลค้างจนเกิดการเสียหาย และใช้เพาเวอร์ทรานซิสเตอร์ TIP127 เป็นตัวขับกระแส

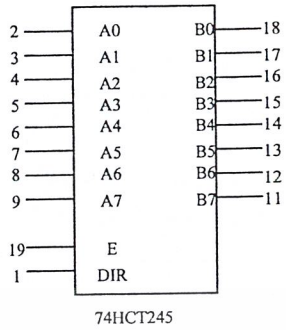


รูปที่ 3.8 แสดงวงจรสวิตซ์ด้านแถว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3.2.3 **วงจรบัฟเฟอร์**

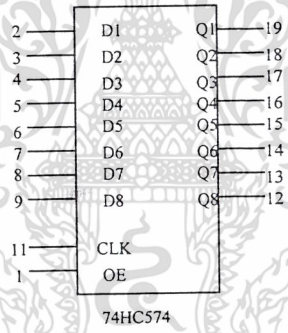
ทำหน้าที่เป็นบัฟเฟอร์สำหรับสัญญาณเลือกแลตซ์และบัสข้อมูล



รูปที่ 3.9 แสดงวงจรบัฟเฟอร์

3.3.2.4 **วงจรแลตซ์ด้านคอลัมน์**

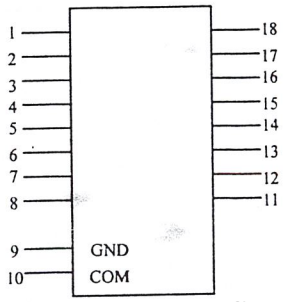
ทำหน้าที่รับข้อมูลจากบัสข้อมูลภายใต้สัญญาณควบคุมการเลือกกลุ่มวงจรแลตซ์ โดยทั้งสัญญาณควบคุมการเลือกและข้อมูลที่เชื่อมต่อกับวงจรส่วนนี้จะป้อนผ่านบัฟเฟอร์มาก่อนแล้ว ไอซีแลตซ์จะสามารถควบคุมวงจรสวิตซ์ด้านคอลัมน์ได้ 1 กลุ่มหรือ 8 คอลัมน์ แลตซ์จะถูกเลือกโดยได้รับพัลส์ที่ขา 11 เพื่อรับข้อมูลจากบัสข้อมูลมาทำการแลตซ์ไว้ โดยจะใช้ไอซีทั้งหมด 16 ตัว



รูปที่ 3.10 แสดงวงจรแลตซ์ด้านคอลัมน์

3.3.2.5 **วงจรสวิตซ์ด้านคอลัมน์**

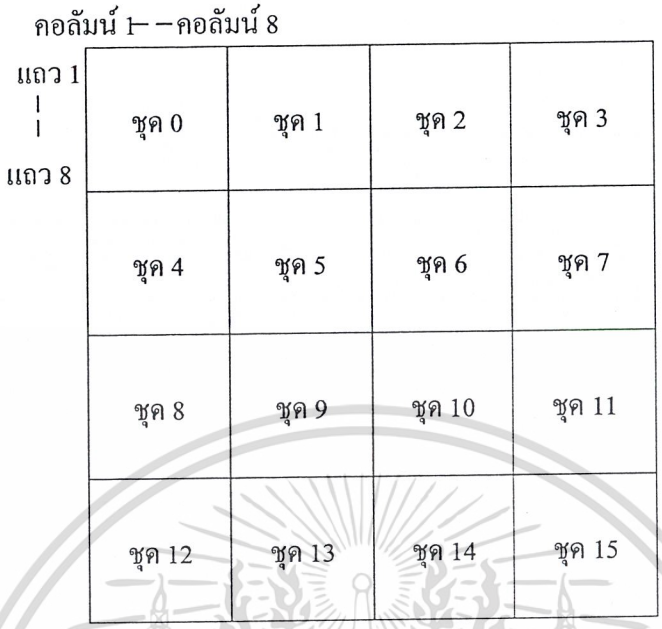
ทำหน้าที่ส่งผ่านกระแสไฟที่ไหลผ่านลงสู่กราวน์ ซึ่งสวิตซ์แต่ละตัวจะถูกควบคุมจากวงจรแลตซ์อีกครั้งหนึ่ง การควบคุมด้านคอลัมน์ต้องใช้สวิตซ์ถึง 128 ตัว ดังนั้นการจัดวงจรในส่วนแลตซ์จะถูกแบ่งออกเป็น 16 กลุ่ม แต่ละกลุ่มสามารถควบคุมสามารถสวิตซ์ด้านคอลัมน์ได้ 8 คอลัมน์



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 3.11 แสดงวงจรสวิตซ์ด้านคอลัมน์

3.3.2.6 วงจรแอลอีดีเมตริกซ์



รูปที่ 3.12 แสดงการวางแผงแอลอีดี ขนาดชุดละ 8X8 จำนวน16 ชุด

ประกอบด้วยแอลอีดีจำนวน 1,024 หลอด (32X32) โดยจะแบ่งเป็น 16 ชุด ชุดหนึ่งจะมีแอลอีดี 64 หลอด (8X8) โดยแอลอีดีที่เชื่อมต่อกันในแต่ละคอลัมน์จะถูกจำกัดกระแสที่ไหลผ่านโดยตัวต้านทาน การที่จะหาค่าความต้านทานเพื่อให้แอลอีดีติดสว่างชัดเจนจะต้องกำหนดให้กระแสไฟที่ไหลผ่านแอลอีดีขณะทำงานมีค่าประมาณ 100 มิลลิแอมป์ (mA) ซึ่งค่านี้ต้องมีค่ามากกว่าอัตรากระแสไฟสูงสุดขณะทำงานของแอลอีดี 3 เท่า จากวงจรแรงดันไฟเลี้ยงที่ป้อนให้เพื่อขับแอลอีดีมีค่า 12 โวลท์ แต่ขณะทำงานจะมีแรงดันไฟเลี้ยงที่จ่ายให้กับแอลอีดีจริง คือ 10 โวลท์ จากข้อมูลเหล่านี้สามารถคำนวณหาค่าความต้านทานได้จากกฎของโอห์มดังนี้

$$R = V/I$$

$$R = 10/0.1$$

$$R = 100 \text{ โอห์ม}$$

และคำนวณกำลังไฟที่เกิดจากการใช้ตัวต้านทาน 100 โอห์ม จะได้ดังนี้

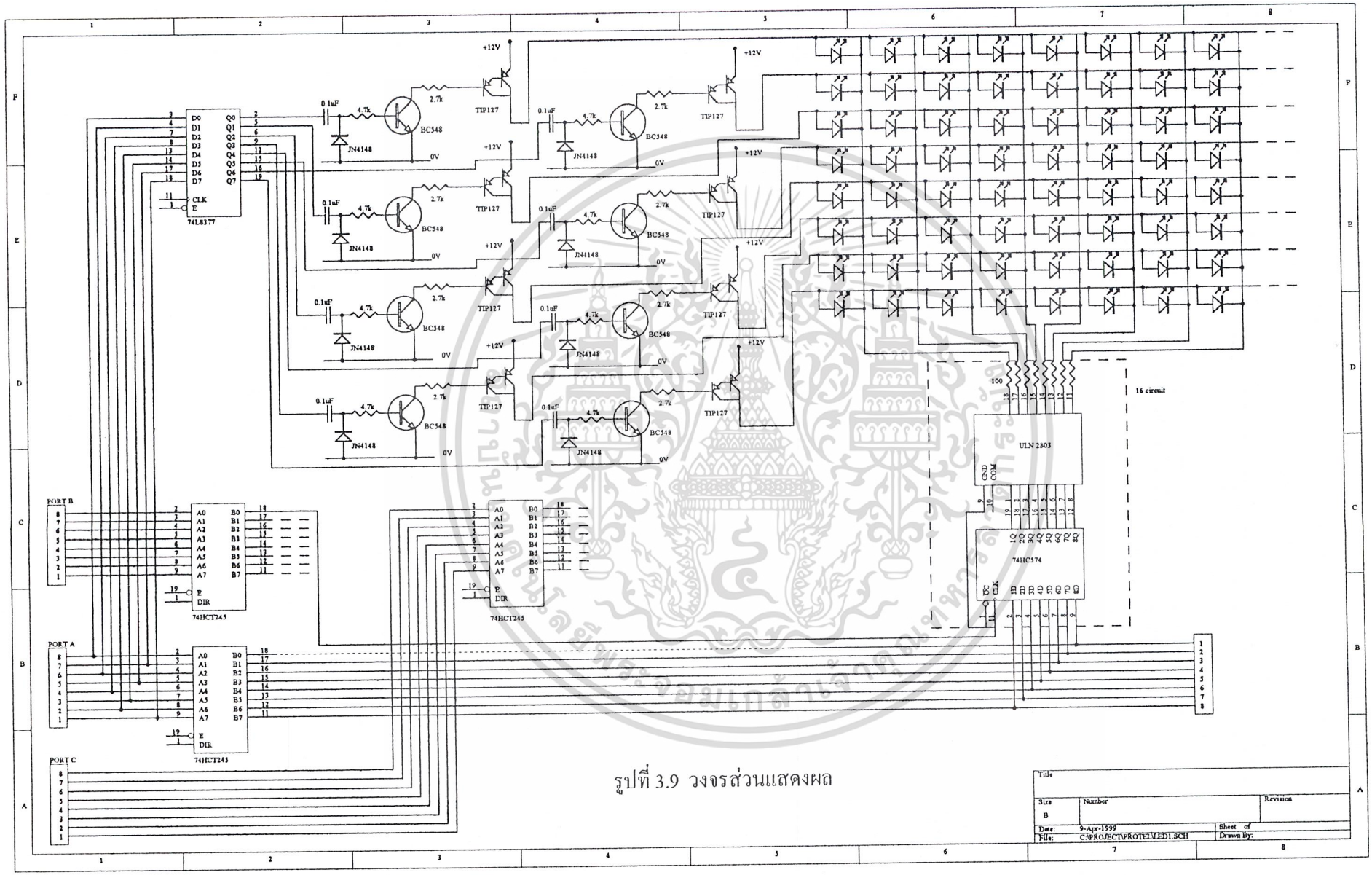
$$P = I^2R$$

$$P = (0.1)^2(100)$$

$$P = 1 \text{ วัตต์}$$

จากการคำนวณค่าความต้านทานใช้ค่า 100 โอห์ม และควรใช้ขนาด 1 วัตต์ขึ้นไปจึงจะเพียงพอ แต่วงจรแอลอีดีนี้เป็นการทำงานแบบมัลติเพล็กซ์ ซึ่งจะทำให้มีกระแสไหลผ่านตัวต้านทานเพียงชั่วขณะสั้นๆ จึงใช้ตัวต้านทานขนาด 0.5 วัตต์แทนได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.9 วงจรส่วนแสดงผล

3.4 โปรแกรม

3.4.1 การเขียนโปรแกรมภาษาซี

จะกล่าวถึงคำสั่งพิเศษที่ใช้จัดการกับพอร์ทอนุกรมและการวาดภาพให้มีลักษณะคล้ายจอแสดงผล คือ หลอดแอลอีดีขนาด 32X32

3.4.1.1 การจัดการกับพอร์ทอนุกรม คำสั่งที่ใช้ในการจัดการกับพอร์ทอนุกรม

Int bioscom (int cmd, char abyte, int port) ;

โดย cmd = 0 กำหนดสถานะให้พอร์ทอนุกรม

= 1 ส่งข้อมูลออกไปยังพอร์ทอนุกรม

= 2 รับข้อมูลจากพอร์ทอนุกรม

= 3 อ่านสถานะปัจจุบันของพอร์ทอนุกรม

port = 0 พอร์ทอนุกรมหมายเลข 1 (คอม 1)

= 1 พอร์ทอนุกรมหมายเลข 2 (คอม 2)

abyte ในกรณีที่ cmd = 0 ค่าใน abyte จะแบ่งออกเป็น 4 กลุ่ม ให้นำค่าใดค่าหนึ่งในแต่ละกลุ่มมากระทำ OR กัน เพื่อให้ได้สถานะที่ต้องการ

กลุ่มที่ 1 0X02 7 บิทข้อมูล

0X03 8 บิทข้อมูล

กลุ่มที่ 2 0X00 1 บิทสิ้นสุด

0X04 2 บิทสิ้นสุด

กลุ่มที่ 3 0X08 ตรวจสอบข้อมูลที่เป็นจำนวนคี่

0X18 ตรวจสอบข้อมูลที่เป็นจำนวนคู่

กลุ่มที่ 4 0X00 110 บอด

0X20 150 บอด

0X40 300 บอด

0X60 600 บอด

0X80 1200 บอด

0Xa0 2400 บอด

0Xe0 9600 บอด

ในกรณีที่ cmd = 1 ค่าใน abyte คือข้อมูลขนาด 8 บิท ที่ต้องการส่งออก

ค่าที่ส่งกลับ แบ่งออกเป็น 2 ส่วนคือ 8 บิทบน และ 8 บิทล่าง ค่าใน 8 บิทบน คือสถานะของพอร์ท มีดังนี้

บิท 15 หมดเวลา (Timeout)

บิท 14 รีจิสเตอร์เลื่อนการส่งว่าง (Transmit shift register empty)

บิท 13 รีจิสเตอร์คงค่าการส่งว่าง (Transmit holding register empty)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บิท 12 การป้องกันการหยุด (Break detect)

บิท 11 ความผิดพลาดของการจัดเฟรม (Framing error)

บิท 10 ความผิดพลาดของการตรวจสอบ (Parity error)

บิท 9 ความผิดพลาดของการประมวลผล (Overrun error)

บิท 8 ความพร้อมของข้อมูล (Data ready)

ในกรณีที่ cmd = 2 ค่า 8 บิทล่างจากเป็นค่าข้อมูลที่ได้รับจากพอร์ท

ในกรณีที่ cmd = 3 ค่า 8 บิทล่างจะเป็นสถานะของพอร์ท ดังนี้

บิท 7 การกู้สัญญาณที่รับได้ (Received line signal detect)

บิท 6 ตัวบอกการริง (Ring indicator)

บิท 5 ความพร้อมของการเซทข้อมูล (Data set ready)

บิท 4 ทำการเคลียร์ก่อนที่จะส่ง(Clear to send)

บิท 3 ตัวบอกการเปลี่ยนสายสัญญาณรับ (Change in receive line signal indicator)

บิท 2 การกู้ขอบสัญญาณการริง (Trailling edge ring detector)

บิท 1 การเปลี่ยนของความพร้อม (Change in data set ready)

บิท 0 การเปลี่ยนให้อยู่ในสถานะเคลียร์ก่อนการส่ง (Change in clear to send)

ในการที่จะส่งข้อมูลจะใช้คำสั่งดังนี้ คือ

bioscom (0, 227, 0) หมายถึง การกำหนดสถานะให้พอร์ทอนุกรมหมายเลข 1 มีอัตราบอด 9600

bioscom (1, data,0) หมายถึง การส่งข้อมูลออกไปยังพอร์อนุกรมหมายเลข 1 และข้อมูลที่ส่งคือ data

3.4.1.2 การวาดภาพให้มีลักษณะคล้ายกับจอแสดงผล

- คำสั่งที่ใช้ในการวาดรูปวงกลม

void far circle(int x, int y, int radius);

(x, y) จุดศูนย์กลาง

radius รัศมี

- คำสั่งที่ใช้ในการวาดรูปวงกลมและระบายสีภายใน ซึ่งจริงๆแล้วเป็นการวาดรูปวงรีและระบายสีภายใน จึงกำหนดให้ xradius และ yradius มีค่าเท่ากันจึงเป็นรูปวงกลม

void far fillellipse(int x, int y, int xradius, int yradius);

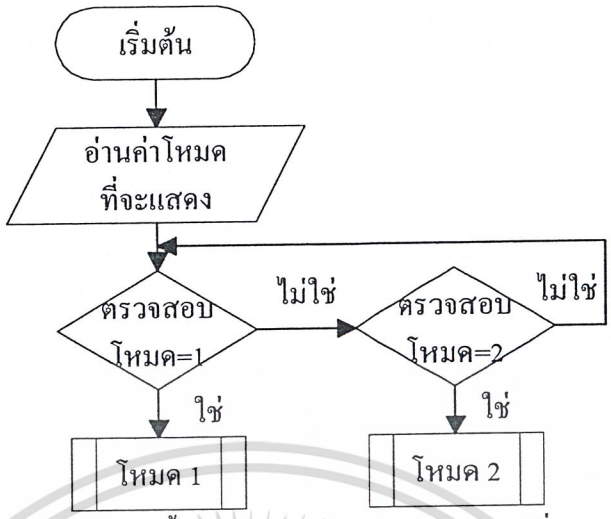
(x, y) จุดศูนย์กลาง

xradius ค่ารัศมีในแกนแนวนอน

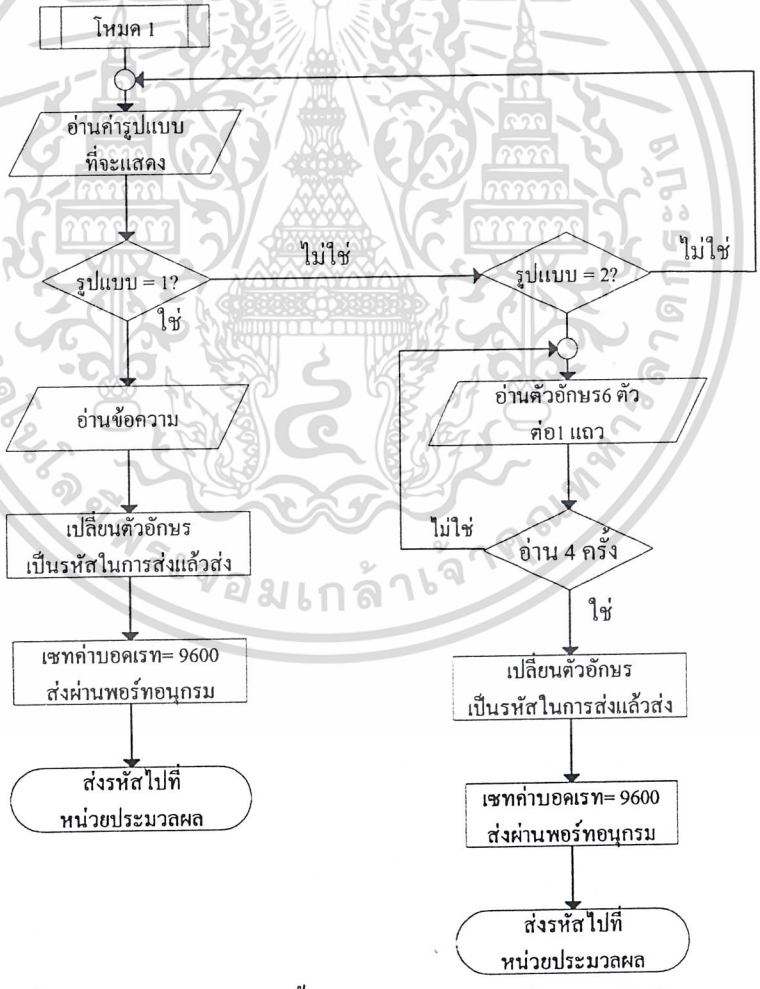
yradius ค่ารัศมีในแกนแนวตั้ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4.1.3 การเขียนโปรแกรมส่งข้อมูลจากคอมพิวเตอร์ผ่านพอร์ทอนุกรมไปยังส่วนควบคุม

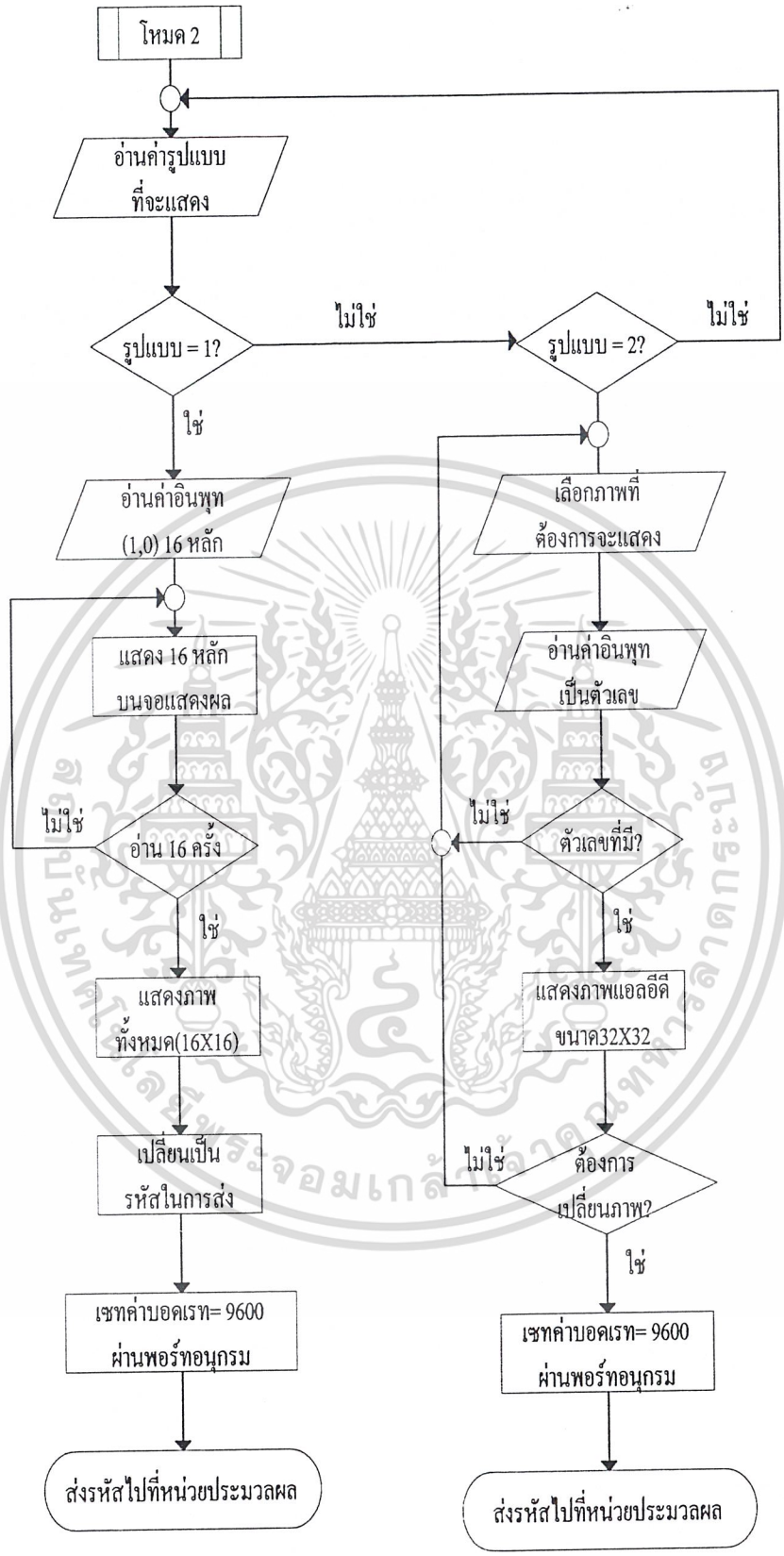


รูปที่ 3.14 โฟลว์ชาร์ทแสดงขั้นตอนการทำงานในการรับรูปแบบที่จะแสดงผล และส่ง ข้อมูลไปยังส่วนแสดงผล



รูปที่ 3.15 โฟลว์ชาร์ทแสดงขั้นตอนการทำงานของโหมด 1(ตัวอักษร)
โดยรูปแบบ = 1 คือ ตัวอักษรวิ่ง 16 ตัวอักษร
รูปแบบ = 2 คือ ตัวอักษร 4 แถวเต็มจอ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.16 โฟลว์ชาร์ทแสดงขั้นตอนการทำงานของโหมด 2 (รูปภาพ)

โดย รูปแบบ = 1 คือ รูปภาพที่สร้างขึ้นเอง ขนาด 16X16

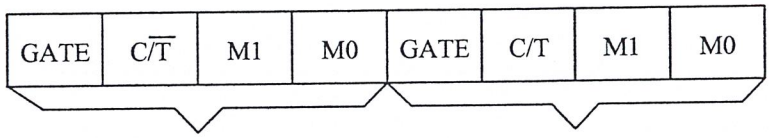
รูปแบบ = 2 คือ เลือกรูปภาพจากที่สร้างเก็บไว้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4.2 การเขียนโปรแกรมภาษาแอสเซมบลี

3.4.2.1 การเซทรีจิสเตอร์ต่างๆเพื่อทำการรับข้อมูลผ่านทางพอร์ทอนุกรม

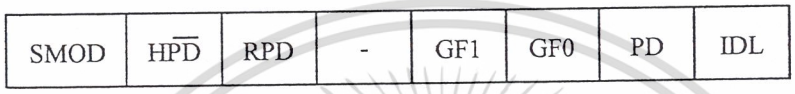
- รีจิสเตอร์ TMOD



รูปที่ 3.17 แสดงรีจิสเตอร์ TMOD

ให้ค่า 20H เพื่อใช้เป็นไทม์เมอร์ 1 โหมด 2 เซทบิท M1

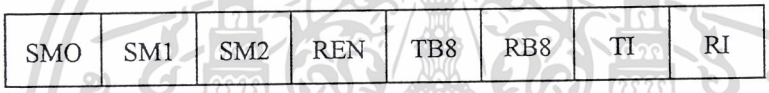
- รีจิสเตอร์ PCON



รูปที่ 3.18 แสดงรีจิสเตอร์ PCON

เซทให้บิท SMOD ใน PCON 0 (ถ้าเซทเป็น 1 อัตราเร็วในการรับข้อมูลจะเป็น 2 เท่าของ 9600)

- รีจิสเตอร์ SCON



รูปที่ 3.19 แสดงรีจิสเตอร์ SCON

ให้ค่า 50H หมายถึง ใช้โหมดการทำงานที่ 1 คือ มีข้อมูล 8 บิท UART อัตราเร็วในการรับข้อมูลกำหนดได้จากไทม์เมอร์ 1 และเซทบิท REN (Receive Enable) บิทควบคุมให้รับข้อมูลได้ ซึ่งถ้ามีการรับข้อมูลเสร็จ 1 ไบท์แล้ว บิท RS จะเป็น 1 และต้องทำการเคลียร์บิทนี้เอง

- TH1

ใช้ในการเซทอัตราเร็วในการรับข้อมูล ซึ่งต้องเซทให้ตรงกับทางด้านส่งด้วย คือมีอัตราเร็ว 9600 บอด เซทค่า TH1 เท่ากับ FDH สามารถคำนวณได้จาก

บอดเรท = (K X Oscillator Freq.)/(32 X 12 X (256 - TH1))

โดยถ้า SMOD = 0 จะใช้ K = 1

ถ้า SMOD = 1 จะใช้ K = 2

หรืออาจเขียนสมการได้เป็น

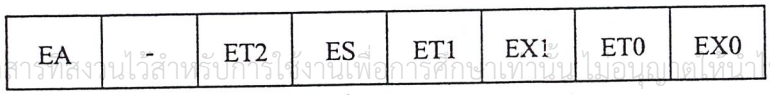
TH1 = 256 - ((K X Oscillator Freq.)/(32 X 12 X (256 - TH1)))

= 256 - ((1 X 11.059)/(384 X 9600))

= 253 = 0FDH

3.4.2.2 การเซทค่าในการให้ค่าอินเตอร์รัพท์เมื่อมีข้อมูลส่งเข้ามา (รับข้อมูล)

รีจิสเตอร์ IE(Interrupt Enable Register)



รูปที่ 3.20 แสดงรีจิสเตอร์ IE

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เซทค่า 90H คือหมายถึงให้เกิดการอินเตอร์รัพท์ที่บิท EA และให้เกิดการอินเตอร์รัพท์ทางพอร์ทอนุกรมที่บิท ES

3.4.2.3 การอ้างแอดเดรสกับ 8255

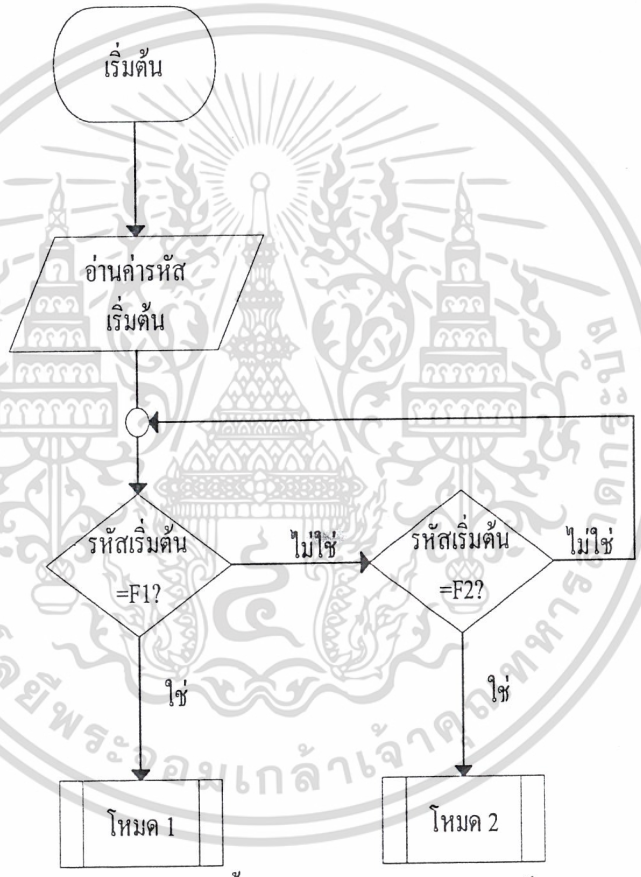
PORT A มีตำแหน่งแอดเดรส F000H

PORT B มีตำแหน่งแอดเดรส F001H

PORT C มีตำแหน่งแอดเดรส F002H

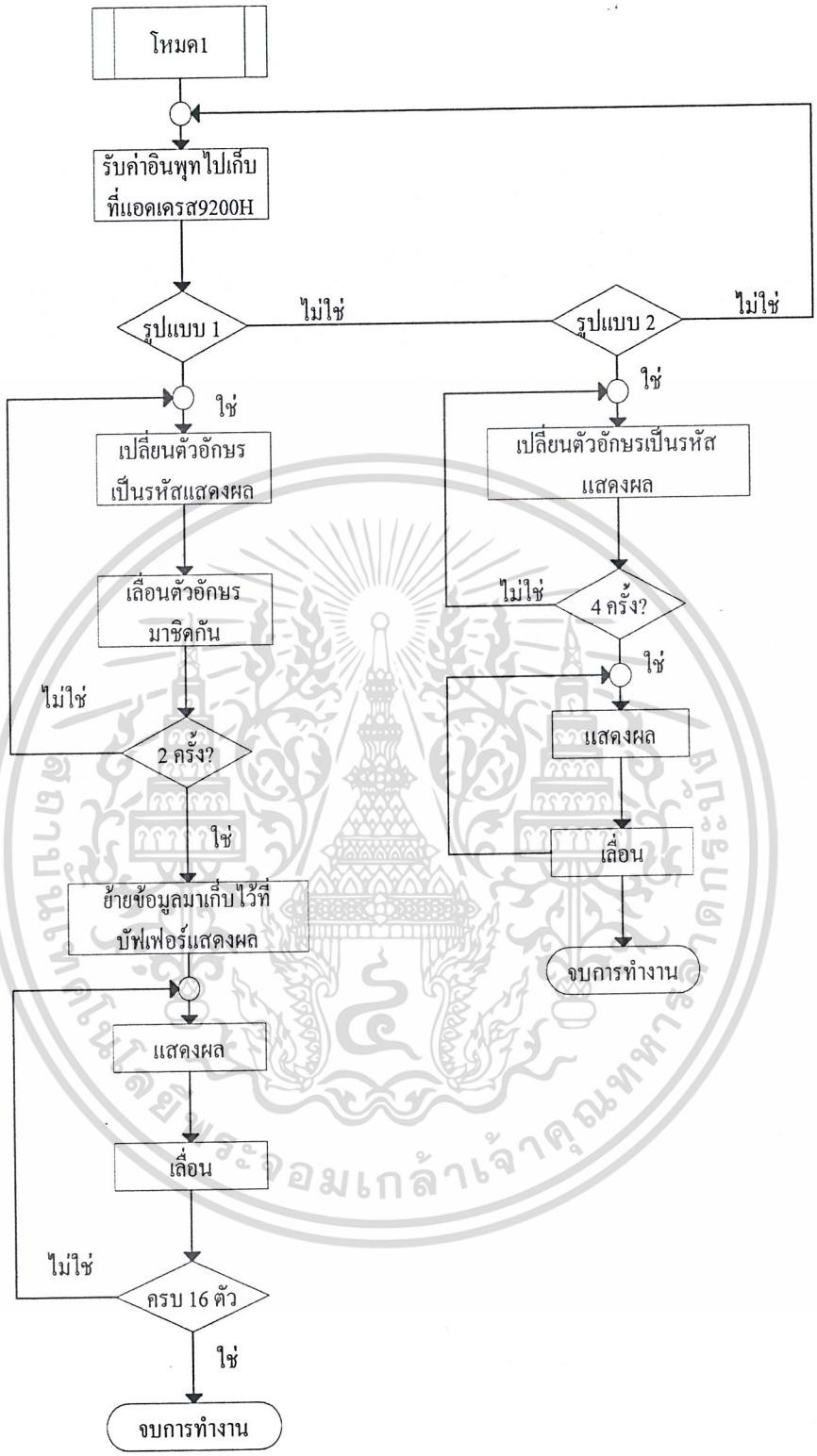
PORT CONTROL มีตำแหน่งแอดเดรส F003H

3.4.2.4 การเขียนโปรแกรมควบคุมส่วนควบคุมและแสดงผล



รูปที่ 3.21 โฟลว์ชาร์ทแสดงขั้นตอนการทำงานในการรับข้อมูลและแยกไปตามโหมดการแสดงผลแบบต่างๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

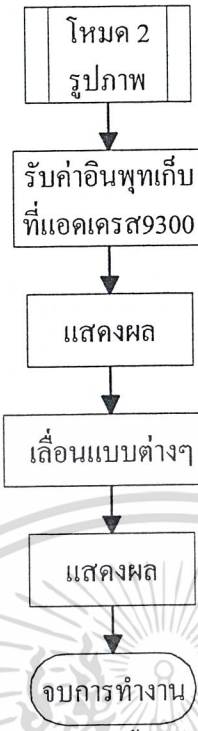


รูปที่ 3.22 โฟลว์ชาร์ทแสดงขั้นตอนการทำงานในการแสดงผลของโหมด 1 (ตัวอักษร)

โดย รูปแบบ = 1 คือ ตัวอักษรวิ่ง 16 ตัวอักษร

รูปแบบ = 2 คือ ตัวอักษร 4 แถวเต็มจอ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.23 โฟลว์ชาร์ทแสดงขั้นตอนการทำงานในการแสดงผลของโหมด 2

โดย รูปแบบ = 1 คือ รูปภาพที่สร้างขึ้นเอง ขนาด 16X16

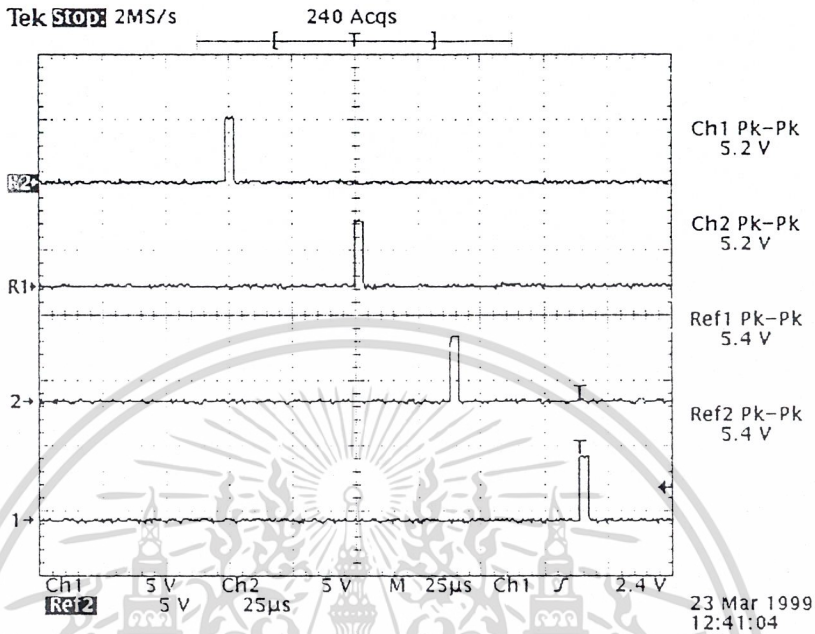
รูปแบบ = 2 คือ เลือกรูปภาพจากที่สร้างเก็บไว้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

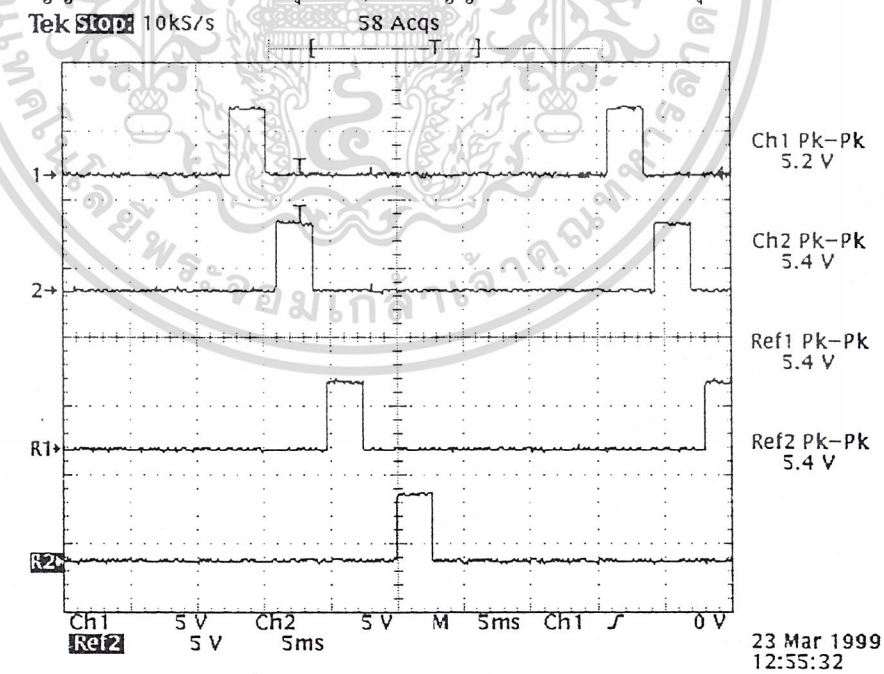
บทที่ 4

การทดลองและผลการทดลอง

4.1 วงจรแลตช์

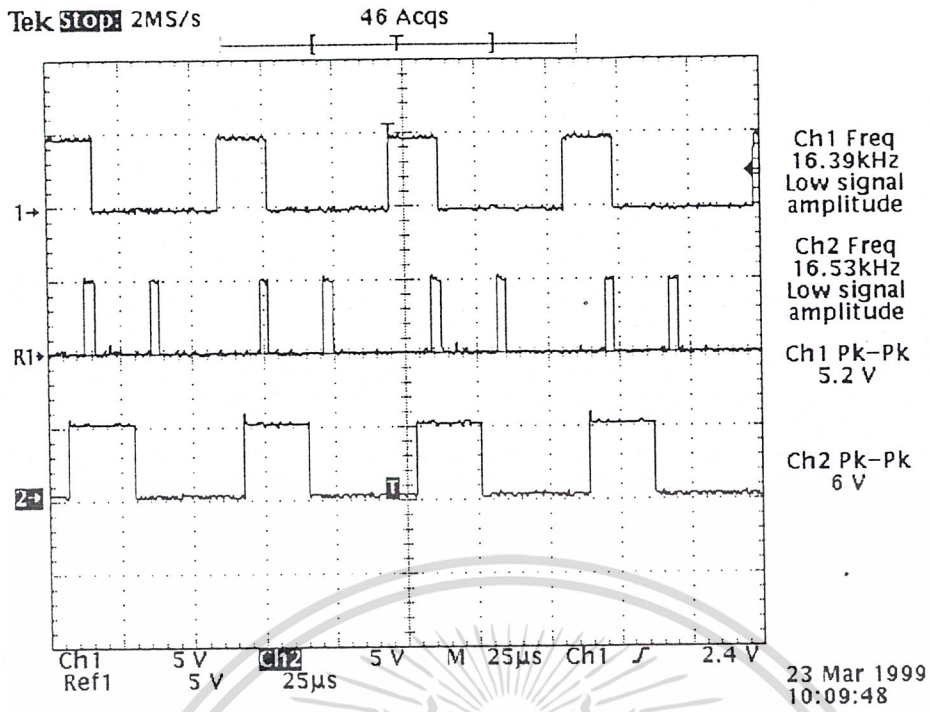


รูปที่ 4.1 แสดงสัญญาณนาฬิกาในการเลือกชุดการแสดงผลทางคอลัมน์โดย ช่องสัญญาณ R2 เป็นการเลือกชุดที่ 1 , ช่องสัญญาณ R1 เป็นการเลือกชุดที่ 2 , ช่องสัญญาณที่ 2 เป็นการเลือกชุดที่ 3 , ช่องสัญญาณที่ 1 เป็นการเลือกชุดที่ 4

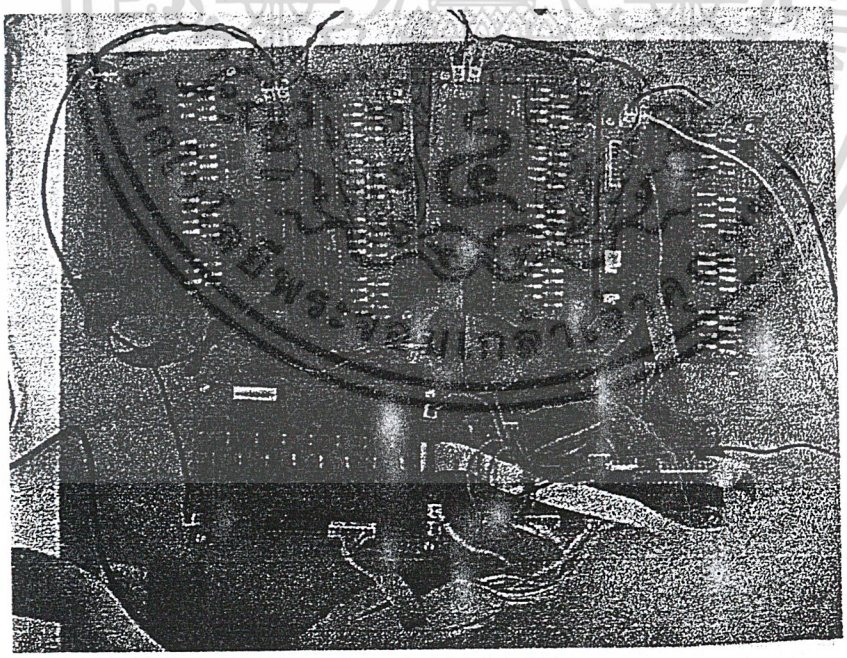


รูปที่ 4.2 แสดงข้อมูลทางด้านแถวที่ใช้ในการสแกนให้เกิดภาพจากไอซี 74LS377โดย ช่องสัญญาณที่ 1 เป็นข้อมูลของแถวที่ 1 , ช่องสัญญาณที่ 2 เป็นข้อมูลของแถวที่ 2 , ช่องสัญญาณ R1 เป็นข้อมูลของแถวที่ 3 และช่องสัญญาณ R2 เป็นข้อมูลของแถวที่ 4

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

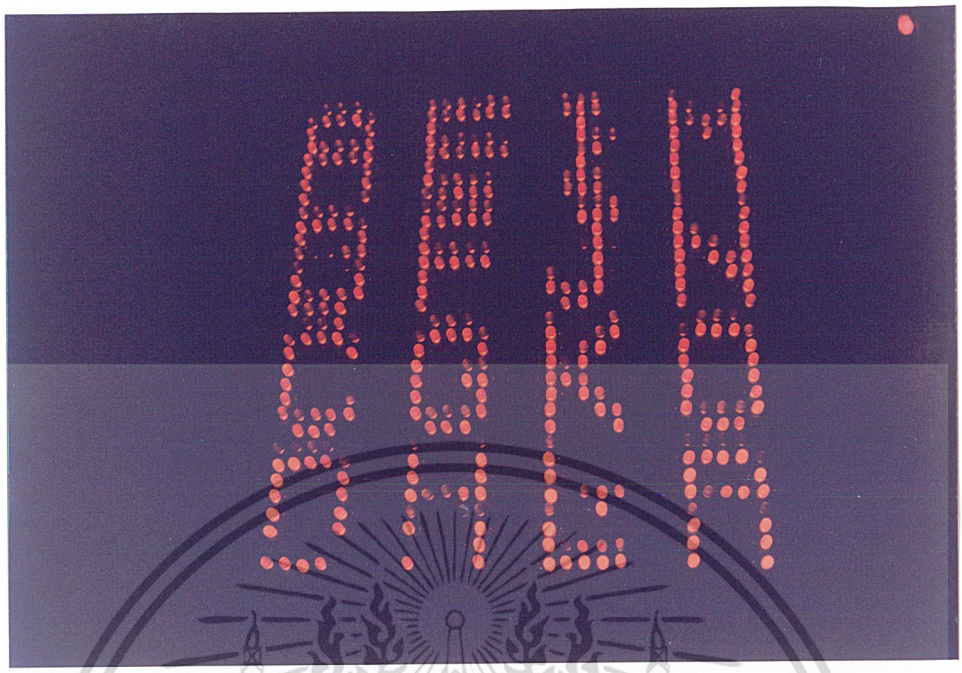


รูปที่ 4.3 แสดงการเปรียบเทียบข้อมูลทางด้านอินพุทและเอาท์พุทของไอซี 74HCT574 โดย ช่องสัญญาณที่ 1 เป็นข้อมูลทางด้านอินพุท , ช่องสัญญาณที่ 2 เป็นข้อมูลทางด้านเอาท์พุท , ช่องสัญญาณ R1 เป็นสัญญาณนาฬิกาในการแลตช์ข้อมูล

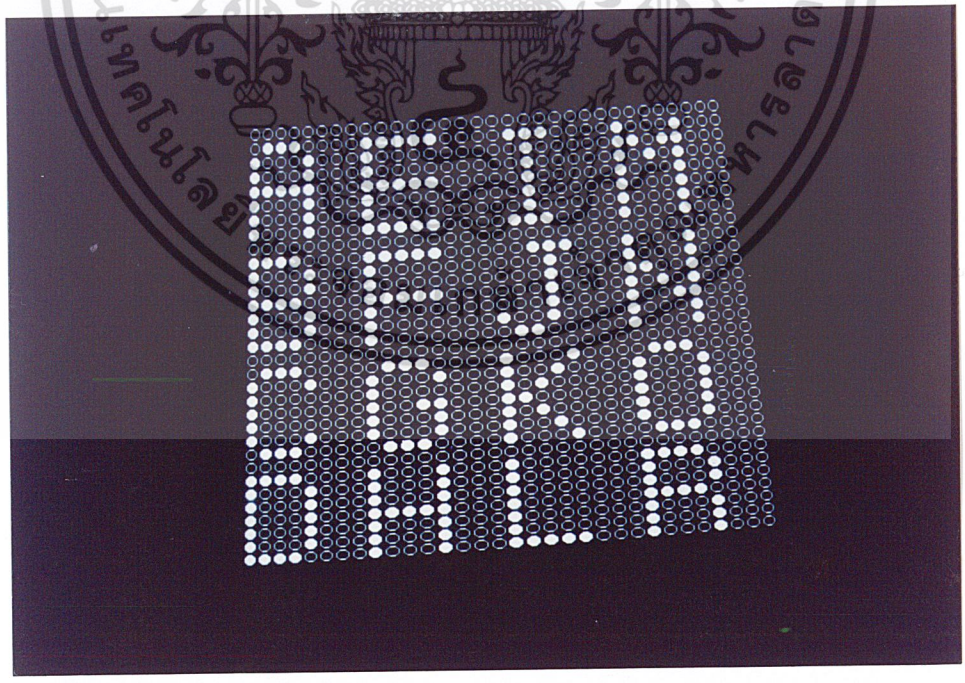


รูปที่ 4.4 แสดงวงจรแลตช์ข้อมูลและกระแส

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.5 ภาพถ่ายจากจอแสดงผลแสดงตัวอักษรภาษาอังกฤษ A-R



รูปที่ 4.6 ภาพถ่ายจากคอมพิวเตอร์แสดงตัวอักษรภาษาอังกฤษ A-R

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 4.7 ภาพถ่ายจากจอแสดงผลแสดงคำว่า GOOD MORNING

รูปที่ 4.8 ภาพถ่ายจากคอมพิวเตอร์แสดงคำว่า GOOD MORNING

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5
บทวิจารณ์และสรุป

5.1 สรุป

ในการสร้างจอแสดงผลนี้ จะเริ่มต้นจากการรับข้อมูลไม่ว่าจะเป็นข้อความหรือรูปภาพจากการป้อนทางคอมพิวเตอร์ผ่านทางพอร์ทอนุกรมไปยังส่วนควบคุมซึ่งใช้ไมโครคอนโทรลเลอร์เป็นหน่วยประมวลผลข้อมูลที่รับมาแปลงเป็นรหัสในการแสดงผลผ่านวงจรขับกระแสและแลตช์ข้อมูลเพื่อให้จอแสดงผลซึ่งเป็นหลอดแอลอีดีขนาด 32*32 ให้ติดและดับตามข้อมูลที่ผู้ใช้ต้องการ

หลังจากการศึกษาและทดลองจอแสดงผลแล้วปรากฏว่า สามารถแสดงผลได้เป็นที่น่าพอใจ แต่ยังเกิดปัญหา คือ เกิดเงาที่ตัวอักษรทุกตัวในการแสดงผล ซึ่งทำให้เกิดความผิดพลาดในการแสดงผลเป็นอย่างมาก อาจแก้ไขได้โดยการให้หลอดแอลอีดีมีความสว่างลดลงจากการโปรแกรม

5.2 บทวิจารณ์

การศึกษาการใช้งานจอแสดงผลแบบแอลอีดีนี้มีข้อดีทำให้ได้เรียนรู้หลักการทำงานของการสแกนและการเขียนโปรแกรมควบคุมระบบต่างๆ ให้ทำงานสัมพันธ์กัน

ส่วนข้อเสีย จะเห็นได้ว่า ในปัจจุบันนี้ จอแสดงผลแบบนี้ไม่เป็นที่นิยมเพราะให้ความละเอียดไม่เพียงพอ และยังมีผลของการแสดงผลที่ไม่ค่อยสวยงามและดึงดูดใจผู้พบเห็น

ชิ้นงานจากโครงงานนี้สามารถนำไปใช้งานได้จริง แต่ยังมีประสิทธิภาพต่ำ ต้องทำการปรับปรุงทั้งทางด้านฮาร์ดแวร์และซอฟท์แวร์เพิ่มเติมให้เหมาะสมกับงานในด้านต่างๆ นั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรมส่วนอินเตอร์เฟสกับผู้ใช้และส่งข้อมูลเข้าส่วนควบคุม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
#include<stdio.h>
#include<stdlib.h>
#include<conio.h>
#include<math.h>
#include<bios.h>
#include<graphics.h>
#include<dos.h>

FILE *muifile;
struct muitype{
       unsigned long int mum;
       }mui;

unsigned int ch_back(unsigned int num);
void bdraw(unsigned int num[132][8]);
void TextWindow(int x1,int y1,int x2,int y2,int fcolor,int bcolor);
unsigned int change(unsigned int num[8][8],int m);
void draw(char num[8],int m);
void adraw(unsigned int num[128][8]);
void sw_data(char dt[],int n);

int pic,mode,form,a,d,e,f,g,x,y;
unsigned int anu,str[128][8],num[8][8],b,dig;
char leg[1],rest[10],c[1],data[20],row1[10],row2[10],row3[10],row4[10];

void main(void)
{      clrscr();
       TextWindow(1,1,79,24,3,5);
       TextWindow(5,4,76,10,8,10);
       TextWindow(5,12,76,21,13,9);
       gotoxy(7,5);
       printf("Please enter mode display\n");
       gotoxy(7,6);
       printf("mode1 is character\n");
       gotoxy(7,7);
       printf("mode2 is picture\n");
       gotoxy(7,8);
       printf("you want to display mode:");
       gets(rest);
       mode=atoi(rest);
       /*printf("mode %d\n",mode);*/
aga:   if (mode==1)
       {       gotoxy(7,13);
               printf("mode1 (character)");
               gotoxy(7,14);
               printf("form1 : 16 character and run in left side\n");
               gotoxy(7,15);
               printf("form2 : full screen(4 rows and 5 character in 1
row\n");
               gotoxy(7,16);
               printf("please enter display form:");
               gets(rest);
               form=atoi(rest);
               /*printf("display form:%d",form);*/
       ag:     if (form==1)
               {       gotoxy(7,17);
                       printf("Please fill the message you want to display(only
16 characters).\n");
                       gotoxy(10,18);
                       gets(data);
                       bioscom(0,227,0);
                       bioscom(1,0xf1,0);
                       gotoxy(10,19);
                       printf("you want to sent message:%s",data);
                       sw_data(data,16);
                       getch();
                       bioscom(1,0xfa,0);
               }
               if (form==2)
               {       gotoxy(7,17);
                       printf("please fill message in each row(only 6 char in 1
row\n");
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของผลงานทุกครั้งที่มีการนำไปใช้

```
                    gotoxy(7,18);
                    printf("row1:");
                    gets(row1);

                    gotoxy(7,19);
                    printf("row2:");
                    gets(row2);

                    gotoxy(40,18);
                    printf("row3:");
                    gets(row3);

                    gotoxy(40,19);
                    printf("row4:");
                    gets(row4);

                    bioscom(0,227,0);
                    bioscom(1,0xf1,0);
                    sw_data(row1,6);
                    printf("\n");
                    sw_data(row2,6);
                    printf("\n");
                    sw_data(row3,6);
                    printf("\n");
                    sw_data(row4,6);
                    printf("\n");
                    bioscom(1,0xfb,0);
                }
                if((form!=1)&(form!=2))
                {       gotoxy(7,16);
                    printf("please enter form again:     ");
                    gotoxy(31,16);
                    scanf("%d",&form);
                    goto ag;
                }
        }
        if (mode==2)
        {       gotoxy(7,13);
                printf("please enter form display\n");
                gotoxy(7,14);
                printf("form1 : create picture by yourself\n");
                gotoxy(7,15);
                printf("form2 : choose the picture from libraly\n");
                gotoxy(7,16);
                printf("you want to display form:");
                gets(rest);
                form=atoi(rest);
        ag2:    if (form==1)
                {       gotoxy(7,17);
                        printf("size = 16*16 and you must put digit in each
8*8");
                        gotoxy(7,18);
                        printf("you can create by press 0 or 1");
                        nuifile=fopen("create.dat","wb");
                                for(a=0;a<128;a++)
                                {       nui.mum=0;
                                        fwrite(&nui,sizeof(struct
nuitype),1,nuifile);
                                }
                        for(a=41;a<49;a++)
                        {       clrscr();
                                TextWindow(4,19,76,22,13,9);
                                x=7;
                                y=20;
                                gotoxy(x,y);
                                printf("8*8 block1");
                                gotoxy(x,y+1);
                                printf("row %d, input number:",a-40);
                                dig=0;
                                gets(rest);
                                for(g=0;g<8;g++)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                        {       switch(rest[g])
                                {       case '0' : dig = dig+0;
                                                   break;
                                        case '1' : dig = dig+(pow(2,8-1-g));
                                                   break;
                                        default  : dig = dig+0;
                                }
                        }
                        draw(rest,8);
                        clrscr();
                        TextWindow(4,20,76,22,13,9);
                        gotoxy(7,21);

                        str[a][8]=ch_back(dig);
                        str[a][0]=dig/128;
                                nui.mum=dig;
                                fseek(nuifile,sizeof(struct
nuitype)*(a),(SEEK_SET));
                                fwrite(&nui,sizeof(struct
nuitype),1,nuifile);
                }
                adraw(str);
                for(a=73;a<81;a++)
                {       clrscr();
                        TextWindow(4,19,76,22,13,9);
                        x=7;
                        y=20;
                        gotoxy(x,y);
                        printf("8*8 block2");
                        gotoxy(x,y+1);
                        printf("row %d, input number:",a-72);
                        dig=0;
                        gets(rest);
                        for(g=0;g<8;g++)
                        {       switch(rest[g])
                                {       case '0' : dig = dig+0;
                                                   break;
                                        case '1' : dig = dig+(pow(2,8-1-g));
                                                   break;
                                        default  : dig = dig+0;
                                }
                        }
                        draw(rest,8);
                        clrscr();
                        TextWindow(4,20,76,22,13,9);
                        str[a][8]=ch_back(dig);
                        str[a][0]=dig/128;
                        nui.mum=dig;
                        fseek(nuifile,sizeof(struct
nuitype)*(a),(SEEK_SET));
                        fwrite(&nui,sizeof(struct nuitype),1,nuifile);
                }
                adraw(str);
                for(a=49;a<57;a++)
                {       clrscr();
                        TextWindow(4,19,76,22,13,9);
                        x=7;
                        y=20;
                        gotoxy(x,y);
                        printf("8*8 block3");
                        gotoxy(x,y+1);
                        printf("row %d, input number:",a-48);
                        dig=0;
                        gets(rest);
                        for(g=0;g<8;g++)
                        {       switch(rest[g])
                                {       case '0' : dig = dig+0;
                                                   break;
                                        case '1' : dig = dig+(pow(2,8-1-g));
                                                   break;
                                        default  : dig = dig+0;
                                }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                        }
                        draw(rest,8);
                        clrscr();
                        TextWindow(4,20,76,22,13,9);
                        str[a][8]=ch_back(dig);
                        str[a][0]=dig/128;
                        nui.mum=dig;
                        fseek(nuifile,sizeof(struct
nuitype)*(a),(SEEK_SET));
                        fwrite(&nui,sizeof(struct nuitype),1,nuifile);
                    }
                    adraw(str);
                    for(a=81;a<89;a++)
                    {       clrscr();
                        TextWindow(4,19,76,22,13,9);
                        x=7;
                        y=20;
                        gotoxy(x,y);
                        printf("8*8 block4");
                        gotoxy(x,y+1);
                        printf("row %d, input number:",a-80);
                        dig=0;
                        gets(rest);
                        for(g=0;g<8;g++)
                        {       switch(rest[g])
                            {       case '0' : dig = dig+0;
                                               break;
                                    case '1' : dig = dig+(pow(2,8-1-g));
                                               break;
                                    default  : dig = dig+0;
                            }
                        }
                        draw(rest,8);
                        clrscr();
                        TextWindow(4,20,76,22,13,9);
                        str[a][8]=ch_back(dig);
                        str[a][0]=dig/128;
                        nui.mum=dig;
                        fseek(nuifile,sizeof(struct
nuitype)*(a),(SEEK_SET));
                        fwrite(&nui,sizeof(struct nuitype),1,nuifile);
                    }
                    fclose(nuifile);
                    bioscom(0,227,0);
                    bioscom(1,0xf2,0);
                    nuifile=fopen("create.dat","rb");
                    rewind(nuifile);
                    while(!feof(nuifile))
                    {       if(fread(&nui,sizeof(struct nuitype),1,nuifile))
                            {       printf("%d ",nui.mum);
                                    bioscom(1,nui.mum,0);
                                    delay(1);
                            }
                    }
                    getch();
                    fclose(nuifile);
                    bioscom(1,0xfc,0);
                    adraw(str);
                    bdraw(str);
                }
                if (form==2)
                {       gotoxy(7,13);
                    printf("choose the picture you want to show");
                    gotoxy(7,14);
                    printf("nui  or  hello                                    ");
                    gotoxy(7,15);
                    printf("you choose picture :                ");
                    gotoxy(28,15);
                ch:     scanf("%d",&pic);
                    switch(pic)
                    {       case 1 : nuifile=fopen("nui.dat","rb");
                                     break;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                        case 2 : nuifile=fopen("hello.dat","rb");
                                 break;
                        default : printf("can't open this file.");
                                  printf("choose picture again");
                                  goto ch;
                    }
                    rewind(nuifile);
                    a=0;
                    bioscom(0,227,0);
                    bioscom(1,0xf2,0);
                    while(!feof(nuifile))
                    {       if(fread(&nui,sizeof(struct nuitype),1,nuifile))
                            {       anu=nui.mum;
                                    str[a][8]=ch_back(anu);
                                    str[a][0]=anu/128;
                                    bioscom(1,anu,0);
                                    delay(1);
                                    a++;
                            }
                    }
                    fclose(nuifile);
                    bdraw(str);
                    bioscom(1,0xfc,0);
                }
                if((form!=1)&(form!=2))
                {       gotoxy(7,16);
                        printf("please enter form again:      ");
                        gotoxy(31,16);
                        scanf("%d",&form);
                        goto ag2;
                }
        }
        if ((mode!=1) & (mode!=2))
        {       gotoxy(7,8);
                printf("please enter mode again:    ");
                gotoxy(31,8);
                scanf("%d",&mode);
                goto aga;
        }
}


void sw_data(char dt[],int n)
{       int a,byte;
        for(a=0;a<n;a++)
        {       switch(dt[a])
                {       case 'a': byte=0x01;
                                  break;
                        case 'b': byte=0x02;
                                  break;
                        case 'c': byte=0x03;
                                  break;
                        case 'd': byte=0x04;
                                  break;
                        case 'e': byte=0x05;
                                  break;
                        case 'f': byte=0x06;
                                  break;
                        case 'g': byte=0x07;
                                  break;
                        case 'h': byte=0x08;
                                  break;
                        case 'i': byte=0x09;
                                  break;
                        case 'j': byte=0x0a;
                                  break;
                        default : byte=0x00;
                                  break;
                }
                printf("%d",byte);
                bioscom(1,byte,0);
                delay(1);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งไม่อนุญาตให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        }
    return;
}

unsigned int change(unsigned int num[8][8],int m)
{   unsigned int b,c;
    int a;
    b=c=0;
    for (a=m-1;a>=0;a--)
    {     if ((num[g][a]!=0)&(num[g][a]!=1))
              return 0;
          b=num[g][a]*(pow(2,m-1-a));
          c=c+b;
    }
    return c;
}

void TextWindow(int x1,int y1,int x2,int y2,int fcolor,int bcolor)
{
    int x,y;
    textcolor(fcolor);
    textbackground(bcolor);
    for(y=y1+1;y<y2;y++)
    {
          for(x=x1+1;x<x2;x++)
          {
                gotoxy(x,y);
                putch(0x20);
          }
          gotoxy(x1,y);
          putch(0xb3);
          gotoxy(x2,y);
          putch(0xb3);
    }
    gotoxy(x1+1,y1);
    for(x=x1+1;x<x2;x++) putch(0xc4);
    gotoxy(x1+1,y2);
    for(x=x1+1;x<x2;x++) putch(0xc4);
    gotoxy(x1,y1); putch(0xda);
    gotoxy(x2,y1); putch(0xbf);
    gotoxy(x1,y2); putch(0xc0);
    gotoxy(x2,y2); putch(0xd9);
}
void draw(char num[8],int m)
{    int gdriver = DETECT,gmode;
    initgraph(&gdriver,&gmode,"c:\\borlandc\\bgi");
    int a,x,y;
    x=200;
    y=200;
    settextjustify(LEFT_TEXT,TOP_TEXT);
    settextstyle(DEFAULT_FONT,HORIZ_DIR,2);
    outtextxy(80,150,"column");
    outtextxy(195,150,"1");
    outtextxy(230,150,"2");
    outtextxy(265,150,"3");
    outtextxy(298,150,"4");
    outtextxy(335,150,"5");
    outtextxy(370,150,"6");
    outtextxy(405,150,"7");
    outtextxy(441,150,"8");
    for (a=0;a<m;a++)
    {     if(num[a]=='0')
          {     circle(x,y,15);
                x=x+35;
          }
          if(num[a]=='1')
          {     fillellipse(x,y,15,15);
                x=x+35;
          }
          if((num[a]!='0')&(num[a]!='1'))
                return;
    }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        delay(2000);
        closegraph();
}
void adraw(unsigned int num[128][8])
{       int gdriver = DETECT,gmode;
        initgraph(&gdriver,&gmode,"c:\\borlandc\\bgi");
        int a,b,x,y;
        x=120;
        y=40;
        for (b=41;b<49;b++)
        {       x=120;
                y=y+23;
                for (a=0;a<8;a++)
                {       if(num[b][a]==0)
                        {       circle(x,y,10);
                                x=x+23;
                        }
                        if(num[b][a]==1)
                        {       fillellipse(x,y,10,10);
                                x=x+23;
                        }
                        if((num[b][a]!=0)&(num[b][a]!=1))
                                return;
                }
                delay(1);
                }
        y=40;
        for (b=73;b<81;b++)
        {       x=303;
                y=y+23;
                for (a=0;a<8;a++)
                {       if(num[b][a]==0)
                        {       circle(x,y,10);
                                x=x+23;
                        }
                        if(num[b][a]==1)
                        {       fillellipse(x,y,10,10);
                                x=x+23;
                        }
                        if((num[b][a]!=0)&(num[b][a]!=1))
                                return;
                }
                delay(1);
        }
        for (b=49;b<57;b++)
        {       x=120;
                y=y+23;
                for (a=0;a<8;a++)
                {       if(num[b][a]==0)
                        {       circle(x,y,10);
                                x=x+23;
                        }
                        if(num[b][a]==1)
                        {       fillellipse(x,y,10,10);
                                x=x+23;
                        }
                        if((num[b][a]!=0)&(num[b][a]!=1))
                                return;
                }
                delay(1);
        }
        y=224;
        for (b=81;b<89;b++)
        {       x=303;
                y=y+23;
                for (a=0;a<8;a++)
                {       if(num[b][a]==0)
                        {       circle(x,y,10);
                                x=x+23;
                        }
                        if(num[b][a]==1)
                        {       fillellipse(x,y,10,10);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                        x=x+23;
                    }
                    if((num[b][a]!=0)&(num[b][a]!=1))
                        return;
                }
                delay(1);
        }
        delay(8000);
        closegraph();
}


unsigned int ch_back(unsigned int num)
{
        for(b=0;b<8;b++)
        {       str[a][8-1-b]=(num%2);
                num=(num/2);
        }
        return(str[a][8]);
}
void bdraw(unsigned int num[128][8])
{       int gdriver = DETECT,gmode;
        initgraph(&gdriver,&gmode,"c:\\borlandc\\bgi");
        int a,b,c,x,y;
        x=50;
        y=5;
        for (b=0;b<32;b++)
        {       x=80;
                y=y+14;
                for (a=0;a<8;a++)
                {       if(num[b][a]==0)
                        {       circle(x,y,5);
                                x=x+14;
                        }
                        if(num[b][a]==1)
                        {       fillellipse(x,y,5,5);
                                x=x+14;
                        }
                        if((num[b][a]!=0)&(num[b][a]!=1))
                                return;
                }
                delay(1);
                }
        y=5;
        for (b=32;b<64;b++)
        {       x=191;
                y=y+14;
                for (a=0;a<8;a++)
                {       if(num[b][a]==0)
                        {       circle(x,y,5);
                                x=x+14;
                        }
                        if(num[b][a]==1)
                        {       fillellipse(x,y,5,5);
                                x=x+14;
                        }
                        if((num[b][a]!=0)&(num[b][a]!=1))
                                return;
                }
                delay(1);
        }
        y=5;
        for (b=64;b<96;b++)
        {       x=302;
                y=y+14;
                for (a=0;a<8;a++)
                {       if(num[b][a]==0)
                        {       circle(x,y,5);
                                x=x+14;
                        }
                        if(num[b][a]==1)
                        {       fillellipse(x,y,5,5);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                        x=x+14;
                }
                if((num[b][a]!=0)&(num[b][a]!=1))
                        return;
        }
        delay(1);
}
y=5;
for (b=96;b<128;b++)
{       x=413;
        y=y+14;
        for (a=0;a<8;a++)
        {       if(num[b][a]==0)
                {       circle(x,y,5);
                        x=x+14;
                }
                if(num[b][a]==1)
                {       fillellipse(x,y,5,5);
                        x=x+14;
                }
                if((num[b][a]!=0)&(num[b][a]!=1))
                        return;
        }
        delay(1);
}
delay(8000);
closegraph();
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรมควบคุมการแสดงผล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        ORG 0000H
        LJMP MAIN

        ORG 0023H
        LJMP RxISR

        PORT1 EQU 0F800H
        PORT2 EQU 0F801H

        PORTA EQU 0F000H
        PORTB EQU 0F001H
        PORTC EQU 0F002H

MAIN:     MOV PSW,#00H
          MOV TMOD,#20H
          MOV TH1,#0FDH
          MOV PCON,#00H
          MOV SCON,#50H
          MOV IE,#90H
          SETB TR1
DISP:      MOV DPTR,#PORT1
          MOV A,#06H
          MOVX @DPTR,A
          MOV DPTR,#PORT2
          MOV A,#0E8H
          MOVX @DPTR,A
          CALL DELAY
          MOV DPTR,#PORT1
          MOV A,#00H
          MOVX @DPTR,A
          MOV DPTR,#PORT2
          MOV A,#0E8H
          MOVX @DPTR,A
          CALL DELAY
          SJMP DISP
DISP2:     MOV DPTR,#PORT1
          MOV A,#3FH
          MOVX @DPTR,A
          MOV DPTR,#PORT2
          MOV A,#0E9H
          MOVX @DPTR,A
          CALL DELAY
          MOV DPTR,#PORT1
          MOV A,#00H
          MOVX @DPTR,A
          MOV DPTR,#PORT2
          MOV A,#0E8H
          MOVX @DPTR,A
          CALL DELAY
          SJMP DISP2
;**************************PROGRAM INTERRUPT
RECEIVE*****************************
RxISR:     CLR RI
          PUSH ACC
          PUSH DPH
          PUSH DPL
          MOV A,SBUF
          CJNE A,#0F1H,TO1
          SJMP CHARMODE
    TO1: CJNE A,#0F2H,TO2
          SJMP PICMODE
    TO2: CJNE A,#0FAH,TO3
          LJMP LONG_CHAR
    TO3: CJNE A,#0FBH,TO4
          LJMP FULL_SCR
    TO4: CJNE A,#0FCH,WRITE
          LJMP PICTURE
CHARMODE: MOV R2,#92H
          MOV R3,#00H
          POP DPL
          POP DPH
          POP ACC
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        MOV A,R3
        SUBB A,#20H
        MOV R3,A
        SJMP ZBYTE
OBYTE:  CJNE A,#01H,ZBYTE
        MOV A,R3
        SUBB A,#10H
        MOV R3,A
ZBYTE:  MOV R0,0F0H
        MOV 0F0H,#00H
LEAN:   MOV A,0F0H
        RL A
        MOV 0F0H,A
        MOV DPH,R4
        MOV DPL,R5
        MOVX A,@DPTR
        CLR C
        RLC A
        MOVX @DPTR,A
        MOV 0F0H,C
        DJNZ R0,LEAN
        CLR C
        MOV DPH,R4
        MOV DPL,R5
        MOVX A,@DPTR
        MOV DPH,R2
        MOV DPL,R3
        MOVX @DPTR,A

        MOV A,R3
        SUBB A,#10H
        MOV R3,A
        MOV DPH,R2
        MOV DPL,R3
        MOVX A,@DPTR
        ORL A,0F0H
        MOVX @DPTR,A

        MOV A,R5
        ADD A,#10H
        MOV R5,A

        MOV A,R6
        ADD A,#03H
        MOV R6,A
        CJNE R6,#18H,NEXT_CH
        CLR C
        MOV A,R5
        SUBB A,#6FH
        MOV R5,A
        DJNZ R7,CH

        CLR C
        MOV A,R5
        ADD A,#30H
        MOV R5,A

        POP 01H
        POP 00H
        CJNE R1,#10H,TOB
        SJMP WRITE0
   TOB: LJMP MOVE8

WRITE0:     MOV R2,#04H
            MOV R3,#08H
            MOV R0,#93H       ;ROW1
            MOV R1,#00H
            CALL WRD
            MOV R0,#93H       ;ROW2
            MOV R1,#08H
            CALL WRD
            MOV R0,#93H       ;ROW3
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                MOV R1,#10H
                CALL WR0
                MOV R0,#93H         ;ROW4
                MOV R1,#18H
                CALL WR0
                SJMP MT5_16
          WR0:  MOV DPH,R0
                MOV DPL,R1
                CLR A
                MOVX @DPTR,A
                INC R1
                DJNZ R3,WR0
                MOV A,R1
                CLR C
                ADD A,#18H
                MOV R1,A
                MOV R3,#08H
                DJNZ R2,WR0
                MOV R2,#04H
                MOV R3,#08H
                RET

MOVTO5_16:   MOV R7,#0AH            ;TO MOVE DATA TO BUFFER 10
TIME(16 CHARS)
                MOV R0,#94H
                MOV R1,#00H
MT5_16:         MOV R2,#93H         ;MOVE DATA TO BUFFER (9388-
938FH => 16 BYTES)
                MOV R3,#88H
                MOV R4,#1DH
                MOV R6,#08H         ;SHIFT & SHOW 8 TIMES
                                    ;AND THEN MOVE ANOTHER CHAR
      TIME16:   MOV DPH,R0
                MOV DPL,R1
                MOVX A,@DPTR
                MOV DPH,R2
                MOV DPL,R3
                MOVX @DPTR,A
                INC R1
                INC R3
                DJNZ R4,TIME16

;?????????????? DON'T FORGET TO PUT SHOW2.ASM IN THIS AREA
??????????????????
                SHOW:   MOV A,#80H          ;INITIAL 8255 A,B,C=O/P
                        MOV DPTR,#0FC03H
                        MOVX @DPTR,A

                        MOV PSW,#08H        ;REGISTER BANK1
                SHO:    MOV R1,#80H         ;CLOCK ROW
                        MOV R2,#80H         ;CLOCK COLUMN
                        MOV R4,#04H         ;4 COLUMN
                        MOV R0,#04H         ;4 ROW OF COLUMN
                        MOV R5,#08H         ;8 ROW
                        MOV DPTR,#9300H
                        MOV R6,DPH
                        MOV R7,DPL
                        MOV R3,#30

                        CLR A
                        MOV DPTR,#PORTA
                        MOVX @DPTR,A

                        CLR 90H
                        SETB 90H
                        CLR 90H

                        CLR A
                        MOV DPTR,#PORTA
                        MOVX @DPTR,A

                        MOV A,#0FFH
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                MOV DPTR,#PORTB
                MOVX @DPTR,A

                INC DPTR
                MOVX @DPTR,A


COLUMN:         MOV DPH,R6
                MOV DPL,R7
                MOVX A,@DPTR            ;DATA AT COLUMN
                MOV DPTR,#PORTA
                MOVX @DPTR,A

                ;CJNE R4,#08H,TON
                CJNE R0,#02H,TON
                CJNE R4,#04H,TON
                MOV R2,#80H

TON:            CALL SETPORT
                MOV A,R2
                MOVX @DPTR,A

                CLR A
                MOVX @DPTR,A

                MOV A,R7                ;NEXT DATA
                ADD A,#20H
                MOV R7,A

                MOV A,R2
                RR A
                MOV A,R2

                DJNZ R4,COLUMN
                MOV R4,#04H
                ;MOV R2,#80H
                CLR C
                MOV A,R7
                SUBB A,#78H
                MOV R7,A
                DJNZ R0,COLUMN
                CLR C
                MOV A,R7
                SUBB A,#1FH
                MOV R7,A
                MOV R0,#04H
                MOV R4,#04H
                MOV R2,#80H

ROW:            MOV A,R1                ;DATA AT ROW
                MOV DPTR,#PORTA
                MOVX @DPTR,A

                SETB 90H
                CLR 90H

                MOV A,R1
                RR A
                MOV R1,A

                CALL DELAY

                DJNZ R5,COLUMN
                DJNZ R3,SHOW
                SJMP SHIFTLEFT
SETPORT:        CJNE R0,#04H,CP
                MOV DPTR,#PORTB
                RET
CP:             CJNE R0,#03H,CPORT
                MOV DPTR,#PORTB
                RET
CPORT:          MOV DPTR,#PORTC
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                    RET

SHIFTLEFT:  MOV PSW,#00H
            MOV R2,#93H          ;SHIFT LEFT DATA IN DISPLAY
BUFFER (ROW2,3)
            MOV R3,#0A8H
            MOV R4,#05H
            MOV R5,#10H
            MOV 0F0H,#00H
   SHIFTL:  CLR C
            MOV A,R3
            SUBB A,#20H
            MOV R3,A
            MOV DPH,R2
            MOV DPL,R3
            CLR C
            MOVX A,@DPTR
            RLC A
            ORL A,0F0H
            MOV 0F0H,C
            MOVX @DPTR,A
            DJNZ R4,SHIFTL
            MOV A,R3
            ADD A,#0A1H
            MOV R3,A
            DJNZ R5,ADDR4
            DJNZ R6,SHIFTLEFT  ; SHIFT & SHOW 8 TIMES THEN
MT5_16
            DJNZ R7,JMT5_16
            SJMP SH_AGAIN
  JMT5_16:  LJMP MT5_16
    ADDR4:  CLR C
            MOV 0F0H,#00H
            MOV R4,#05H
            SJMP SHIFTL

SH_AGAIN:   MOV R6,#08H
SH_AGA:     MOV R2,#93H
            MOV R3,#BBH
            MOV R4,#04H
            MOV R5,#10H
            MOV 0F0H,#00H
     SH_L:  CLR C
            MOV A,R3
            SUBB A,#20H
            MOV R3,A
            MOV DPH,R2
            MOV DPL,R3
            CLR C
            MOVX A,@DPTR
            RLC A
            ORL A,0F0H
            MOV 0F0H,C
            MOVX @DPTR,A
            DJNZ R4,SH_L
            MOV A,R3
            ADD A,#0A1H
            MOV R3,A
            DJNZ R5,ADDR3
            DJNZ R6,SH_AGA
            LJMP DISP
    ADDR3:  CLR C
            MOV 0F0H,#00H
            MOV R4,#05H
            SJMP SH_L
TABLE:      CJNE R7,#08H,CON     ;8 CHARACTERS
            LJMP CHIT
     CON:   INC R1
            INC R7
    ZERO:   CJNE A,#00H,ONE
            MOV DPTR,#CHARA
            RET
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
ONE:      CJNE A,#01H,TWO
          MOV DPTR,#CHARB
          RET
TWO:      CJNE A,#02H,THREE
          MOV DPTR,#CHARC
          RET
THREE:    CJNE A,#03H,FOUR
          MOV DPTR,#CHARD
          RET
FOUR:     CJNE A,#04H,FIVE
          MOV DPTR,#CHARE
          RET
FIVE:     CJNE A,#05H,SIX
          MOV DPTR,#CHARF
          RET
SIX:      CJNE A,#06H,SEVEN
          MOV DPTR,#CHARG
          RET
SEVEN:    CJNE A,#07H,EIGHT
          MOV DPTR,#CHARH
          RET
EIGHT:    CJNE A,#08H,NINE
          MOV DPTR,#CHARI
          RET
NINE:     CJNE A,#09H,TEN
          MOV DPTR,#CHARJ
          RET
TEN:      CJNE A,#0AH,ELEVEN
          MOV DPTR,#CHARK
          RET
ELEVEN:   CJNE A,#0BH,TWELVE
          MOV DPTR,#CHARL
          RET
TWELVE:   CJNE A,#0CH,THIRTEEN
          MOV DPTR,#CHARM
          RET
THIRTEEN: CJNE A,#0DH,FOURTEEN
          MOV DPTR,#CHARN
          RET
FOURTEEN: CJNE A,#0EH,FIFTEEN
          MOV DPTR,#CHARO
          RET
FIFTEEN:  CJNE A,#0FH,SIXTEEN
          MOV DPTR,#CHARP
          RET
SIXTEEN:  CJNE A,#10H,SEVENTEEN
          MOV DPTR,#CHARQ
          RET
SEVENTEEN: CJNE A,#11H,EIGHTEEN
          MOV DPTR,#CHARR
          RET
EIGHTEEN: CJNE A,#12H,NINETEEN
          MOV DPTR,#CHARS
          RET
NINETEEN: CJNE A,#13H,TWENTY
          MOV DPTR,#CHART
          RET
TWENTY:   CJNE A,#14H,T21
          MOV DPTR,#CHARU
          RET
T21:      CJNE A,#15H,T22
          MOV DPTR,#CHARV
          RET
T22:      CJNE A,#16H,T23
          MOV DPTR,#CHARW
          RET
T23:      CJNE A,#17H,T24
          MOV DPTR,#CHARX
          RET
T24:      CJNE A,#18H,T25
          MOV DPTR,#CHARY
          RET
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
T25: CJNE A,#19H,T26
     MOV DPTR,#CHARZ
     RET
T26: CJNE A,#1AH,T27
     MOV DPTR,#NO0
     RET
T27: CJNE A,#1BH,T28
     MOV DPTR,#NO1
     RET
T28: CJNE A,#1CH,T29
     MOV DPTR,#NO2
     RET
T29: CJNE A,#1DH,T30
     MOV DPTR,#NO3
     RET
T30: CJNE A,#1EH,T31
     MOV DPTR,#NO4
     RET
T31: CJNE A,#1FH,T32
     MOV DPTR,#NO5
     RET
T32: CJNE A,#20H,T33
     MOV DPTR,#NO6
     RET
T33: CJNE A,#21H,T34
     MOV DPTR,#NO7
     RET
T34: CJNE A,#22H,T35
     MOV DPTR,#NO8
     RET
T35: CJNE A,#23H,T36
     MOV DPTR,#NO9
     RET
T36: MOV DPTR,#SPACE
     RET
CHARA: DB 00H,70H,88H,88H,0F8H,88H,88H,88H
CHARB: DB 00H,0F0H,88H,88H,0F0H,88H,88H,0F0H
CHARC: DB 00H,70H,88H,80H,80H,80H,88H,70H
CHARD: DB 00H,0F0H,88H,88H,88H,88H,88H,0F0H
CHARE: DB 00H,0F8H,80H,80H,0F8H,80H,80H,0F8H
CHARF: DB 00H,0F8H,80H,80H,0F0H,80H,80H,80H
CHARG: DB 00H,70H,88H,80H,98H,88H,88H,70H
CHARH: DB 00H,88H,88H,88H,0F8H,88H,88H,88H
CHARI: DB 00H,70H,20H,20H,20H,20H,20H,70H
CHARJ: DB 00H,38H,10H,10H,10H,10H,90H,60H
CHARK: DB 00H,88H,90H,0A0H,0C0H,0A0H,90H,88H
CHARL: DB 00H,80H,80H,80H,80H,80H,80H,0F8H
CHARM: DB 00H,88H,0D8H,0A8H,88H,88H,88H,88H
CHARN: DB 00H,88H,88H,0C8H,0A8H,98H,88H,88H
CHARO: DB 00H,70H,88H,88H,88H,88H,88H,70H
CHARP: DB 00H,0F0H,88H,88H,0F0H,80H,80H,80H
CHARQ: DB 00H,70H,88H,88H,88H,0A8H,98H,68H
CHARR: DB 00H,0F0H,88H,88H,0F0H,0A0H,90H,88H
CHARS: DB 00H,70H,88H,80H,70H,08H,88H,70H
CHART: DB 00H,0F8H,20H,20H,20H,20H,20H,20H
CHARU: DB 00H,88H,88H,88H,88H,88H,88H,70H
CHARV: DB 00H,88H,88H,88H,88H,88H,50H,20H
CHARW: DB 00H,88H,88H,88H,88H,0A8H,0A8H,50H
CHARX: DB 00H,88H,88H,50H,20H,50H,88H,88H
CHARY: DB 00H,88H,88H,70H,20H,20H,20H,20H
CHARZ: DB 00H,0F8H,08H,10H,20H,40H,80H,0F8H
NO0: DB 00H,70H,88H,98H,0A8H,0C8H,88H,70H
NO1: DB 00H,20H,60H,20H,20H,20H,20H,70H
NO2: DB 00H,70H,88H,08H,10H,20H,40H,0F8H
NO3: DB 00H,0F8H,10H,20H,10H,08H,88H,70H
NO4: DB 00H,10H,30H,50H,90H,0F8H,10H,10H
NO5: DB 00H,0F8H,80H,0F0H,08H,08H,88H,70H
NO6: DB 00H,30H,40H,80H,0F0H,88H,88H,70H
NO7: DB 00H,0F8H,08H,10H,20H,40H,40H,40H
NO8: DB 00H,70H,88H,88H,70H,88H,88H,70H
NO9: DB 00H,70H,88H,88H,78H,08H,10H,60H
SPACE: DB 00H,00H,00H,00H,00H,00H,00H,00H
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการ[illegible]ะโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัด[illegible]ที่มีการนำไปใช้

```
FULL_SCR:   MOV A,#0FEH
            MOV DPH,R2
            MOV DPL,R3
            MOVX @DPTR,A
            MOV PSW,#08H
            MOV R0,#04H
            MOV PSW,#00H

            MOV R0,#92H
            MOV R1,#00H
   MOVE5:   MOV R4,#93H
            MOV R5,#0A0H
            MOV R7,#00H
   CHA:     MOV R6,#00H
            MOV DPH,R0
            MOV DPL,R1
            MOVX A,@DPTR
            CALL TABLE2
   MOV:     MOV DPH,R2
            MOV DPL,R3
            MOVX A,@DPTR
            MOV DPH,R4
            MOV DPL,R5
            MOVX @DPTR,A
            INC R3
            INC R5
            INC R6
            CJNE R6,#08H,MOV     ;1 CHAR = 8 DATA BYTES
            SJMP CHA
     CHI:   PUSH 00H
            MOV R2,#93H
            MOV R3,#0A8H
            MOV R0,#08H
      CA:   MOV R7,#00H
            MOV R6,#03H
NEXT_CHAR:  MOV A,R2
            MOV R4,A
            MOV A,R3
            MOV R5,A
            MOV A,R6
            MOV 0F0H,#08H
            DIV AB
            CJNE A,#02H,ONEBYTE
            MOV A,R5
            SUBB A,#10H
            MOV R5,A
            SJMP ZEROBYTE
 ONEBYTE:   CJNE A,#01H,ZEROBYTE
            MOV A,R5
            SUBB A,#08H
            MOV R5,A
ZEROBYTE:   MOV R7,0F0H
            MOV 0F0H,#00H
     LEOAN: MOV A,0F0H
            RL A
            MOV 0F0H,A
            MOV DPH,R2
            MOV DPL,R3
            MOVX A,@DPTR
            CLR C
            RLC A
            MOVX @DPTR,A
            MOV 0F0H,C
            DJNZ R7,LEOAN
            CLR C
            MOV DPH,R2
            MOV DPL,R3
            MOVX A,@DPTR
            MOV DPH,R4
            MOV DPL,R5
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
MOVX @DPTR,A
MOV A,R5
SUBB A,#08H
MOV R5,A
            MOV DPH,R4
            MOV DPL,R5
            MOVX A,@DPTR
            ORL A,0F0H
            MOVX @DPTR,A

            MOV A,R3
            ADD A,#08H
            MOV R3,A

            MOV A,R6
            ADD A,#03H
            MOV R6,A
            CJNE R6,#12H,NEXT_CHAR
            CLR C
            MOV A,R3
            SUBB A,#27H
            MOV R3,A
            DJNZ R0,CA
            POP 00H

SHIFTRIGHT: MOV R2,#93H
            MOV R3,#98H
            MOV R4,#05H
            MOV R5,#08H
            MOV 0F0H,#00H

 SHIFTR:    CLR C
            MOV A,R3
            ADD A,#08H
            MOV R3,A
            CLR C
            MOV DPH,R2
            MOV DPL,R3
            MOVX A,@DPTR
            RRC A
            ORL A,0F0H
            MOVX @DPTR,A
            MOV 0F0H,#00H
            MOV 0F0H,C
            MOV A,0F0H
            RRC A
            MOV 0F0H,#00H
            MOV 0F0H,A
            DJNZ R4,SHIFTR
            MOV R4,#05H
            MOV A,R3
            SUBB A,#27H
            MOV R3,A
            CLR C
            MOV 0F0H,#00H
            DJNZ R5,SHIFTR


  SETROW:   CJNE R1,#06H,RW2
            MOV R3,#00H
            SJMP MT_SHBUF
      RW2:  CJNE R1,#0CH,RW3
            MOV R3,#08H
            SJMP MT_SHBUF
      RW3:  CJNE R1,#12H,RW4
            MOV R3,#10H
            SJMP MT_SHBUF
      RW4:  MOV R3,#18H
 MT_SHBUF:  MOV R2,#93H
            MOV R4,#93H
            MOV R5,#0A0H
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                MOV R6,#04H
N_COL:          MOV R7,#08H
AGAIN:          MOV DPH,R4
                MOV DPL,R5
                MOVX A,@DPTR
                MOV DPH,R2
                MOV DPL,R3
                MOVX @DPTR,A
                INC R3
                INC R5
                DJNZ R7,AGAIN
                MOV A,R3
                ADD A,#18H
                MOV R3,A
                DJNZ R6,N_COL
                MOV PSW,#08H
                DJNZ R0,TOC
                LJMP DISP
TOC:            MOV PSW,#00H
                LJMP MOVE5

TABLE2:         CJNE R7,#06H,CON        ; 5 CHAR
                LJMP CHI
CON:            INC R1
                INC R7
                AJMP ZERO

                LJMP DISP2
PICTURE:        MOV A,#0FEH
                MOV DPH,R2
                MOV DPL,R3
                MOVX @DPTR,A
                LJMP DISP2

DELAY:          PUSH 00H
                PUSH 01H
                MOV R0,#64H
DELAY1:         MOV R1,#17H
DELAY2:         DJNZ R1,DELAY2
                DJNZ R0,DELAY1
                POP 01H
                POP 00H
                RET
                END
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

โครงงานจอแสดงผลสำเร็จลงไม่ได้ หากไม่ได้รับความช่วยเหลือและคำปรึกษาจากบุคคลเหล่านี้ ทั้ง รศ. สมยศ จุณณะปิยะ ที่ให้คำปรึกษาและความรู้ในเรื่องไมโครคอนโทรลเลอร์ , อาจารย์ สมเกียรติ ฤกษ์วรัญญู สำหรับความเอาใจใส่ ให้คำแนะนำ เครื่องมือ อุปกรณ์ต่างๆ และที่พักพิง ในการทำโครงงานนี้ รวมทั้งบุคคลดังต่อไปนี้ด้วย

นายชุมพล เหลืองรัตน์ ช่วยเหลือ และคอยอำนวยความสะดวกในการทำโครงงาน

นางสาวอุษณีย์ ศุภณีดิส ช่วยในการพิมพ์รายงาน และวาดรูปที่ใช้ในการแสดงผล

นางสาวมาลียา วิวิธวรรณ์ ช่วยในการวาดรูปต่างๆ มากมาย

และตลอดจนเพื่อนๆ ทุกคนที่ให้ความช่วยเหลือ และให้กำลังใจเสมอมา จนทำให้โครงงานนี้สำเร็จด้วยดี

คณะผู้จัดทำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หนังสืออ้างอิง

1. ธันวา ศรีประโมง, "การเขียนโปรแกรมภาษาซีสำหรับวิศวกรรม",มหาวิทยาลัยเทคโนโลยีมหานคร, พิมพ์ครั้งที่ 4,2538
2. รศ.สมยศ จุณณะปิยะ, "การประยุกต์ใช้งานไมโครคอนโทรลเลอร์ ตระกูล MCS-51, สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง, กรุงเทพฯ, พิมพ์ครั้งที่ 2, 2541
3. สุเจตน์ จันทรักษ์, "ไมโครคอนโทรลเลอร์ชิพเดี่ยว 8051", วิทยาลัยมหานคร, 2535
4. "เซมิคอนดัคเตอร์", พฤษภาคม 2539

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้