ระบบแสดงผลรอบทิศทาง

DISPLAY SURROUND SYSTEM

โดย

นายไชยยศ แสงนวล

นายยศ จันทร์สว่าง

เลขเรียกหนังสือ ปห ว888ร ร5ม1

เลขทะเบียน 019600

วัน เดือน ปี 19 ก.ค. 42

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิศวกรรมการวัดคุมทางอุตสาหกรรม

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2541

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์ปีการศึกษา 2541

ภาควิชา เทคโนโลยีการวัดคุมทางอุตสาหกรรม

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง ระบบแสดงผลรอบทิศทาง

ผู้จัดทำ

นายไชยยศ	แสงนวล	39013381
นายยศ	จันทร์สว่าง	39013396

.....อาจารย์ที่ปรึกษา

(ทรงชัย วีระทวีมาศ)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระบบแสดงผลรอบทิศทาง

ไชยยศ แสงนวล
ยศ จันทร์สว่าง

บทคัดย่อ

ในวิทยานิพนธ์ฉบับนี้ เป็นเรื่องเกี่ยวกับระบบแสดงผลโดยใช้หลอด LED เป็นตัวแสดงผล ซึ่งในการแสดงผลนั้นเราจะสามารถเห็นได้รอบทิศทาง โดยเราได้นำเอาไมโครคอนโทรลเลอร์ MCS – 51 มาใช้สำหรับการควบคุมการแสดงผล ข้อความที่เราต้องการให้ปรากฏบนตัวแสดงผลนั้นเราสามารถป้อนข้อความได้โดยผ่านคีย์บอร์ดคอมพิวเตอร์ ซึ่งจะเชื่อมต่ออยู่กับตัวไมโครคอรโทลเลอร์ MCS – 51 และในขณะที่เราป้อนข้อความอยู่นั้น ข้อความที่เราป้อนก็จะปรากฏขึ้นบนจอ LCD ซึ่งจะเชื่อมต่ออยู่กับตัวไมโครคอรโทลเลอร์ MCS – 51 ด้วยเช่นกัน

Abstract

This thesis is about Display Surround System by using LED as Display Surround System. To show the message on Display Surround System, we can key the message through computer's keyboard, which is connected with microcontroller MCS-51 and while keying the message, these message will show on LCD monitor which is connected with microcontroller MCS-51 as well.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่1
บทนำ

ระบบแสดงผลรอบทิศทาง

ความเป็นมา

ในสังคมปัจจุบันความเจริญก้าวหน้าทางอิเลคทรอนิคส์มีการพัฒนาไปเรื่อยๆ อย่างไม่หยุดยั้ง ไม่ว่าจะเป็นทางด้าน Hardware หรือ Software ความเจริญดังกล่าวล้วนเกิดขึ้นมาเพื่อสนองความต้องการของมนุษย์ทั้งสิ้น จากแนวความคิดของมนุษย์นี้เองทำให้การพัฒนาของ Microprocessor เป็นไปอย่างต่อเนื่อง ทำให้ได้เบอร์ CPU ใหม่ๆ ออกมาหลายเบอร์ ซึ่งในที่นี้ก็มี CPU ที่เรารู้จักกันดีในชื่อของ MCS – 51 ซึ่งผลิตโดยบริษัทอินเทล ผู้ผลิตชิพไอซีชั้นนำของโลก ตัว MCS – 51 นั้นสามารถใช้งานได้อย่างกว้างขวาง มีคำสั่งใช้งานที่ง่าย ตรงไปตรงมา มีคู่มือและตัวอย่างใช้งานมากมาย ดังนั้นเราจึงได้นำเอา MCS – 51 มาประยุกต์ใช้งานเกี่ยวกับระบบแสดงผลรอบทิศทางด้วยหลอด LED

วัตถุประสงค์ของ PROJECT

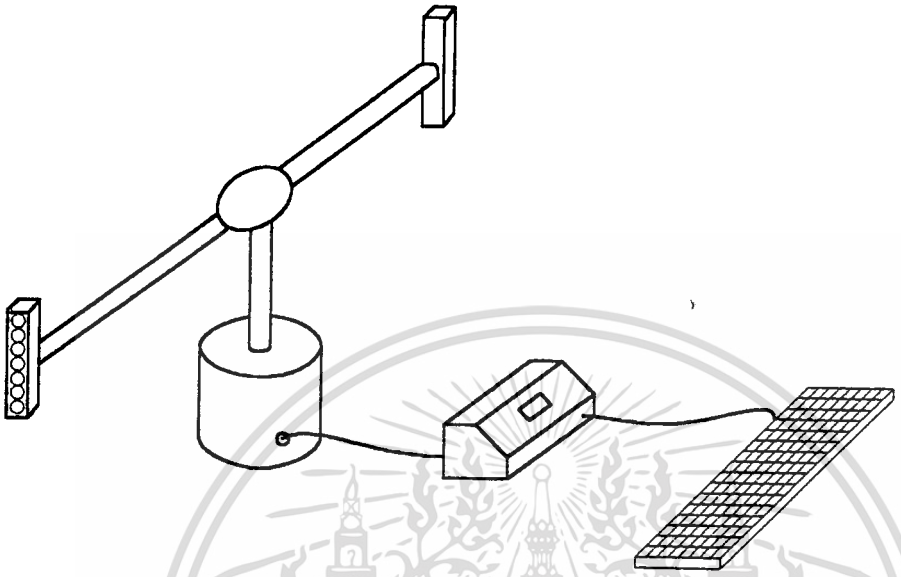
วัตถุประสงค์ของ PROJECT นี้คือ เพื่อความแปลกใหม่ในการแสดงผลด้วยหลอด LED กล่าวคือส่วนใหญ่แล้ว บอร์ดแสดงผล LED ที่พบเห็นโดยทั่วไปนั้น จะมีลักษณะเป็นหลอด LED เรียงเป็นแถวติดๆ กันหลายๆ แถว และแสดงผลโดยการเลื่อนไปทีละหลัก ซึ่งจำเป็นที่จะต้องใช้หลอด LED จำนวนมากในการแสดงผล และยังมองเห็นได้แต่เพียงด้านหน้าเท่านั้น ดังนั้นทางคณะผู้จัดทำ จึงได้ทำการคิดค้น และดัดแปลงรูปแบบการนำเสนอให้เป็นที่น่าสนใจแก่ผู้พบเห็น ทำให้ได้มาซึ่งระบบแสดงผลรอบทิศทาง

หลักการทำงานของระบบแสดงผลรอบทิศทาง

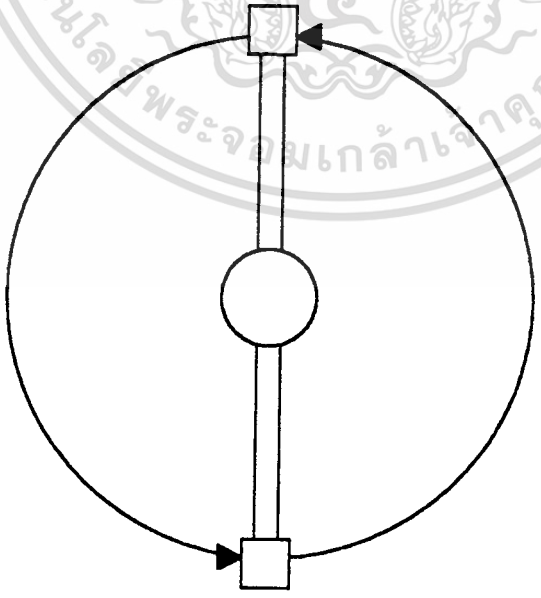
หลักการทำงานจะเริ่มจาก เมื่อมอเตอร์ที่โมดูล 1 (รูปที่1.1) หมุนก็จะมีแรงขับ ทำให้แกนเหล็กซึ่งมีชุดแสดงผลยึดติดอยู่ที่ปลายทั้ง 2 ด้านของแกนเหล็กหมุนตามไปด้วย (ชุดแสดงผลจะประกอบด้วยหลอด LED จำนวน 8 หลอด) การหมุนของชุดแสดงผลนั้นจะหมุนในลักษณะเป็นวงกลมดังรูปที่ 1.2 ด้วยความเร็วรอบในการหมุนที่คงที่ ซึ่งมีความสัมพันธ์กันกับความถี่ในการกระพริบของหลอด LED ในชุดแสดงผล ที่จะกระพริบอีกครั้งเมื่อชุดแสดงผลหมุนกลับมาอยู่ ณ. ตำแหน่งเดิมทุกครั้ง ทำให้การมองเห็นของมนุษย์นั้นไม่สามารถแยกแยะได้ว่า ในขณะที่เรากำลังมองชุดแสดงผลอยู่นั้น ชุดแสดงผลได้มีการกระพริบของหลอด LED อยู่ แต่เรากลับมองเห็นว่าหลอด LED ของชุดแสดงผลนั้น จะสว่างอยู่ที่ตำแหน่งเดิมอยู่ตลอดเวลา จากหลักการดังกล่าว เมื่อเราเพิ่มจำนวนตำแหน่งในการแสดงผลหลายๆ ตำแหน่ง มาประกอบเข้าด้วยกัน แล้วกำหนด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ลักษณะในการติดดับของหลอด LED ในตำแหน่งต่างๆ นั้น ก็จะทำให้เราสามารถกำหนดรูปแบบในการแสดงผลของชุดแสดงผลรอบทิศทางได้ เช่นการแสดงรูปภาพแบบง่ายๆ หรือ ตัวอักษร (รูปที่ 1.3)



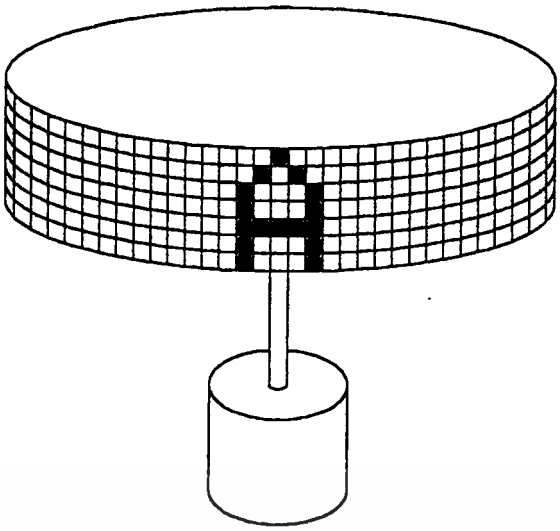
รูปที่ 1.1



Top View

รูปที่ 1.2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 1.3

หลักการและทฤษฎีที่เกี่ยวข้องที่จะนำมาใช้

1. ทฤษฎีเกี่ยวกับการใช้งาน MCS – 51
2. ทฤษฎีเกี่ยวกับจอแสดงผลแบบ LCD
3. ทฤษฎีเกี่ยวกับคีย์บอร์ดคอมพิวเตอร์
4. หลักการเกี่ยวกับการเชื่อมต่อ MCS – 51 กับจอแสดงผลแบบ LCD
5. หลักการเกี่ยวกับการเชื่อมต่อ MCS – 51 กับคีย์บอร์ดของคอมพิวเตอร์
6. หลักการเกี่ยวกับการส่งข้อมูลผ่านทางพอร์ทอนุกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2
ไมโครคอนโทรเลอร์ MCS-51

โครงสร้างของ MCS-51

ไมโครคอนโทรลเลอร์แบบชิพเดี่ยวตระกูล MCS-51 นี้ ผลิตโดยบริษัท อินเทล มีอยู่ด้วยกันหลายเบอร์ซึ่งมีรายละเอียดดังตารางที่ 2.1

Table 1. The MCS®-51 Family of Microcontrollers

Device	ROMless Version	EPROM Version	ROM Bytes	RAM Bytes	8-Bit I/O Ports	16-Bit Timer/ Counters	Programmable Counter Array (PCA)	UART	Serial Expansion Port (SEP)	Global Serial Channel (GSC)	DMA Channels	A/D Channels	Interrupt Sources/ Vectors	Power Down and Idle Modes
8051	8031	—	4K	128	4	2		✓					6/5	
8051AH	8031AH	8751H 8751BH	4K	128	4	2		✓					6/5	
8052AH	8032AH	8752BH	8K	256	4	3		✓					8/6	
80C31BH	80C31BH	87C51	4K	128	4	2		✓					6/5	✓
80C51FA	80C51FA	87C51FA	8K	256	4	3	✓	✓					14/7	✓
80C51FB	80C51FA	87C51FB	16K	256	4	3	✓	✓					14/7	✓
80C51GA	80C51GA	87C51GA	4K	128	4	2		✓	✓			8	8/7	✓
80C152JA	80C152JA	—	8K	256	5	2		✓		✓	2		19/11	✓
—	80C152JB	—	—	256	7	2		✓		✓	2		19/11	✓
83C152JC	80C152JC	—	8K	256	5	2		✓		✓	2		19/11	✓
—	80C152JD	—	—	256	7	2		✓		✓	2		19/11	✓
83C451	80C451	—	4K	128	7	2		✓					6/5	✓
83C452	80C452	87C452P	8K	256	5	2		✓					8/8	✓

ตารางที่ 2.1 แสดงความแตกต่างของสมาชิกไมโครคอนโทรลเลอร์

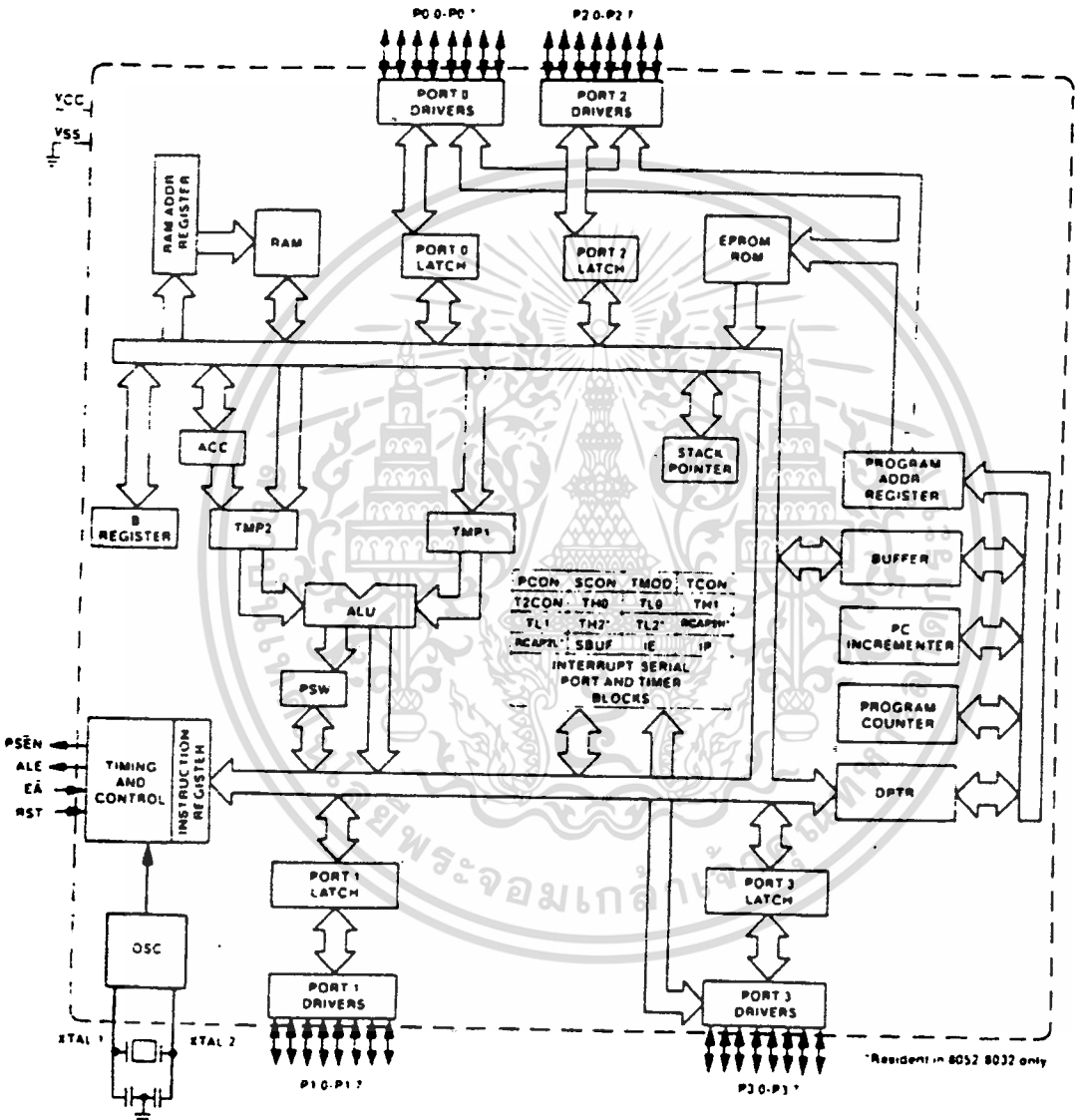
คุณสมบัติของไมโครคอนโทรลเลอร์ตระกูล MCS-51

- ต้องการแหล่งจ่ายไฟ +5V ชุดเดียว
- มีหน่วยความจำโปรแกรม (Program Memory) ขนาด 4 กิโลไบต์สำหรับเบอร์ 8051 และ 8031,8032 ไม่มีหน่วยความจำชุดนี้ ส่วน 8052 มีหน่วยความจำถึง 8 กิโลไบต์
- มีหน่วยความจำสำหรับเก็บข้อมูล (Data Memory) ขนาด 128 ไบต์ สำหรับ 8052 มีถึง 256 ไบต์
- หน่วยความจำสำหรับโปรแกรมและข้อมูล แยกจากกันอย่างละ 64 กิโลไบต์
- คำสั่งที่ใช้เวลาน้อยที่สุดใช้เวลาประมาณ 1 μs เมื่อทำงานที่ความถี่ 12 MHz
- มี Timer/Counter ขนาด 16 บิต 2 ชุด (สำหรับ 8052 มี 3 ชุด) ทำงานได้ 4 โหมด
- รับอินเตอร์รัพท์ได้ 6 แหล่ง 5 เวคเตอร์
- มีพอร์ทรับส่งข้อมูลอนุกรม (UART) ทั้งรับและส่งในเวลาเดียวกันได้ (Full Duplex) เลือกรูปแบบการรับส่งข้อมูลได้ 4 โหมด
- มีคำสั่งในการทำ AND, OR, หรือ COMPLEMENT ได้ทั้งแบบ 8 บิต และ 1 บิต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น พร้อมทั้งไม่อนุญาตให้ทำซ้ำดัดแปลงเอกสารนี้ก่อนได้รับอนุญาตเป็นลายลักษณ์อักษรครั้งที่มีการนำไปใช้

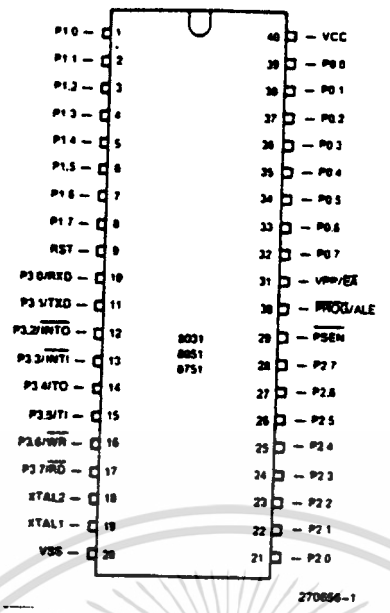
โครงสร้างภายในของ 8051

MCS-51 ใช้เทคโนโลยีในการผลิตแบบ NMOS และ CMOS เบอร์ 8032 และ 8052 จะมี ROM BASIC อยู่ภายใน จึงสะดวกต่อโปรแกรมเมอร์ที่จะเขียนโปรแกรมด้วยภาษา BASIC โครงสร้างภายในสำหรับเบอร์ 8051 ดังแสดงในรูป 2.1



รูป 2.1 8051 บล็อกไดอะแกรมของ MCS – 51

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

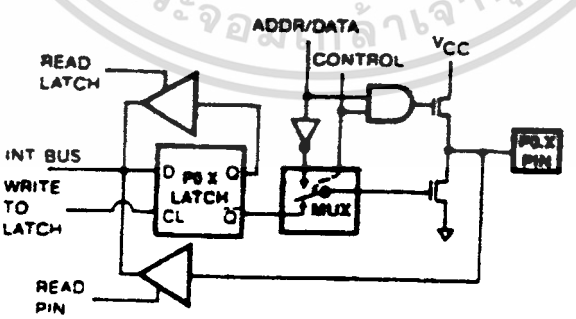


รูป 2.2 การจัดวางขาของ 8051

พอร์ทของ 8051

8051 เป็นไมโครคอนโทรเลอร์ขนาด 40 ขา ซึ่งมีขาต่างๆ ดังนี้

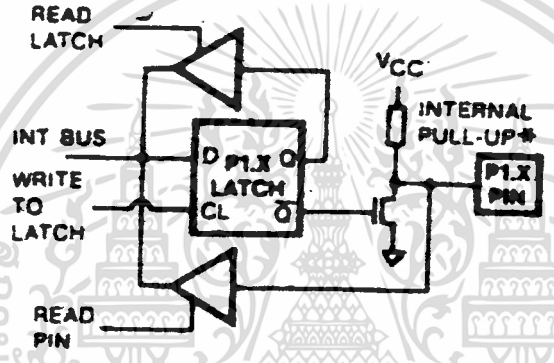
1. Vcc (ขา 40) ต่อกับ +5V
2. Vss (ขา 20) เป็นขา GND
3. PORT 0 (ขา 32 – 39) มีทั้งหมด 8 บิตคือ (P0.0 – P0.7) มีโครงสร้างแบบ Open Drain Bidirectional ดังรูป 2.3



รูป 2.3 แสดงโครงสร้างพอร์ท 0 (บิต)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

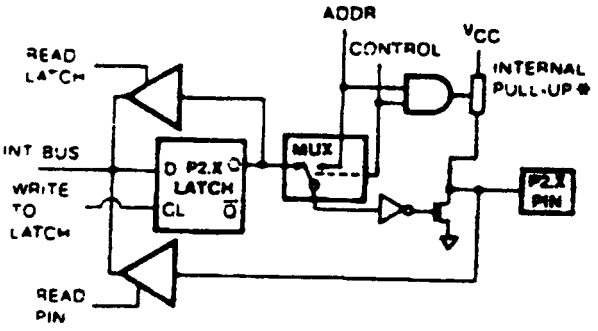
4. พอร์ท 0 (ขา 32 –39) มีทั้งหมด 8 บิตคือ (P0.0 – P0.7) ใช้งานได้ 2 หน้าที่คือ ส่งแอดเดรส และดาต้าออกไป ให้หน่วยความจำภายนอก เมื่อทำการเขียนข้อมูลลงในหน่วยความจำภายนอกควบคุมด้วยขา Control และอีกหน้าที่หนึ่งก็คือเป็นพอร์ท I/O ถ้าต้องการให้ทำงานเป็นอินพุทพอร์ท ต้องส่งลอจิก "1" ไปยังพอร์ทนี้ จะมีผลให้ $\overline{Q}$ ของ D – FF เป็น "0" ทำให้ FET ตัวล่างมีสถานะ OFF สัญญาณที่ใช้อ่านอินพุทพอร์ท PIN (พอร์ท P0.X PIN) จะใช้สัญญาณ READ LATCH เมื่อถูกกระตุ้นที่ Tri – State Buffer ตัวบน
5. พอร์ท 1 (ขา 1 – 8) มีทั้งหมด 8 บิตคือ (P1.0 –P1.7) มีโครงสร้างคล้ายพอร์ท 0 แต่จะใช้ความต้านทานภายในพูลอัพแทน (Internal Up Register) มีโครงสร้างดังรูป 2.4



รูป 2.4 โครงสร้างของพอร์ท 1 (บิต)

6. พอร์ท 2 (ขา 21 – 28) มีทั้งหมด 8 บิตคือขา (P2.0 – P2.7) มีโครงสร้างคล้ายพอร์ท 0 โดยมี FET ตัวล่างตัวเดียวส่วนด้านบนใช้ความต้านทานพูลอัพแทน (Internal pull – up) พอร์ทนี้ทำงาน 2 หน้าที่คือ สามารถใช้เป็นพอร์ทส่งแอดเดรส 8 บิตบน (A8 – A15) และเป็น I/O พอร์ทใช้งานทั่วไป เมื่อจะใช้งานเป็นอินพุทพอร์ท ต้องส่งลอจิก "1" มาที่พอร์ทนี้ก่อนเพื่อบังคับให้ FET อยู่ในสภาวะ OFF ดังแสดงในรูป 2.5

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

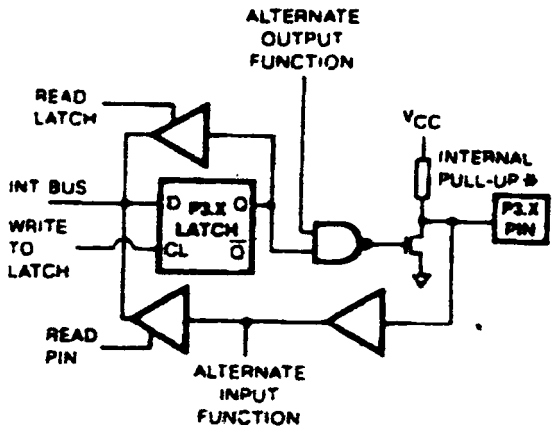


รูป 2.5 โครงสร้างของพอร์ท 2 (บิต)

7. พอร์ท 3 (ขา 10 –17) มีทั้งหมด 8 บิตคือขา (P3.0 –P3.7) มีโครงสร้างคล้ายพอร์ท 1 พอร์ทนี้ทำหน้าที่คือเป็น I/O พอร์ท ถ้าจะให้พอร์ทนี้เป็น อินพุทพอร์ท ก็ให้ส่งลอจิก "1" มาที่พอร์ทนี้ก่อน และอีกหน้าที่หนึ่งก็คือ ส่งสัญญาณควบคุมออกมา และรับสัญญาณเข้าไป สัญญาณต่างๆ มีดังนี้

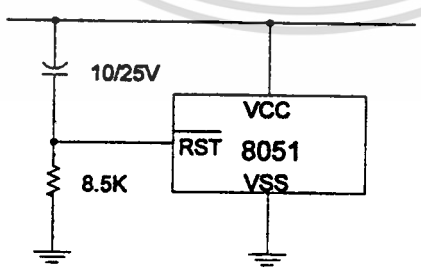
- P3.0/RxD (Serial Input Port) เป็นขาที่ใช้รับข้อมูลอนุกรม
- P3.1/RxD (Serial Output Port) เป็นขาที่ใช้ส่งข้อมูลอนุกรม
- P3.2/$\overline{\text{INT}}$ 0 (External Interrupt) ใช้รับสัญญาณขัดจังหวะจากภายนอก
- P3.3/$\overline{\text{INT}}$ 1 (External Interrupt) ใช้รับสัญญาณขัดจังหวะจากภายนอก
- P3.4/T0 (Timer/Counter0 External Input) ขารับสัญญาณเข้าไปยังวงจร Timer/Counter 0 ที่ทำหน้าที่นับจำนวนของไซเคิลสัญญาณ T1 นี้หรือสัญญาณนาฬิกาก็ได้
- P3.5/T1 (Timer/Counter 1 External Input) ขารับสัญญาณเข้าไปยัง Timer/Counter1 ซึ่งมีการทำงานเหมือนกับ T0
- P3.6/$\overline{\text{WR}}$ (External Data Memory Write Strobe) ขาสัญญาณควบคุมการเขียนข้อมูลไปยังหน่วยความจำสำหรับข้อมูลภายนอก 8051
- P3.7/$\overline{\text{RD}}$ (External Data Memory Read Strobe) ขาสัญญาณควบคุมการอ่านข้อมูลจากหน่วยความจำสำหรับข้อมูลภายนอก 8051

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูป 2.6 โครงสร้างของพอร์ท 3 (บิต)

8. ALE (ขา 30) เป็นขาส่งสโตรบสำหรับใช้ในการแลตช์แอดเดรสไบต์ต่ำ (A0 – A7) ที่ส่งออกมาจาก (พอร์ท 0) สัญญาณนี้จะแอคทีฟทุกๆ 2 ครั้งใน ๆ 1 แมชชีนไซเคิล
9. $\overline{\text{PSEN}}$ (ขา 29) เป็นขาที่ใช้สำหรับส่งสโตรบสำหรับอ่านข้อมูลจาก Program Memory ภายนอก (หน่วยความจำประเภท ROM EPROM) สัญญาณนี้จะส่งออกมา 2 ครั้ง ในแต่ละแมชชีนไซเคิล แต่ถ้าเป็นการอ่าน Internal Program Memory จะไม่มีสัญญาณออกที่ขานี้
10. $\overline{\text{EA}}$ (ขา 30) ถ้าป้อนลอจิก “0” เข้าที่ขานี้ CPU จะอ่านค่าจาก Program Memory ภายนอกชิพเท่านั้น แต่ถ้าถูกป้อนด้วยลอจิก “1” ก็จะอ่านโปรแกรมภายในชิพ
11. RST (ขา 9) เป็นขารีเซ็ต CPU จะรีเซ็ตได้ก็ต่อเมื่อป้อนลอจิก “1” เข้าที่ขานี้อย่างน้อย 2 แมชชีนไซเคิล เมื่อ CPU ถูกรีเซตค่าต่างๆ ในรีจิสเตอร์ใดๆ จะมีคำสั่งดังตารางที่ 2.4
12. XTAL1 (ขา 19) ใช้ต่อคริสตัลภายนอก โดยเป็นอินพุทเข้าสู่วงจรออสซิเลเตอร์
13. XTAL2 (ขา 18) ใช้ต่อคริสตัลภายนอก โดยเป็นอินพุทเข้าสู่วงจรออสซิเลเตอร์

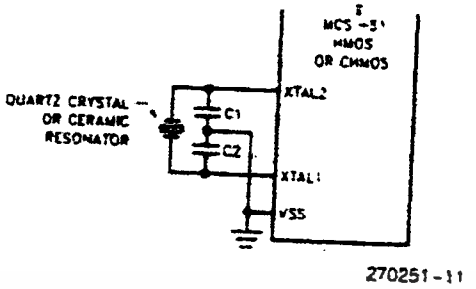


รูป 2.7 การต่อขารีเซตให้กับ 8051

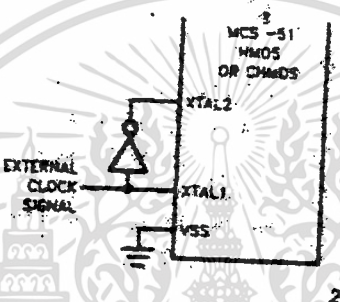
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

วงจร Clock ของ 8051

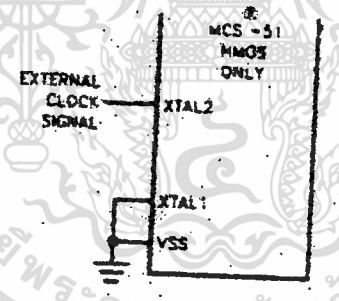
การต่อมือยู่ด้วยกัน 2 แบบคือ แบบ Clock ภายใน และ Clock ภายนอก มีรูปแบบการต่อดัง รูป 2.8



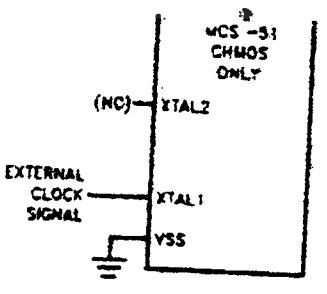
Using the On-Chip Oscillator



A. HMOS or CHMOS



B. HMOS Only

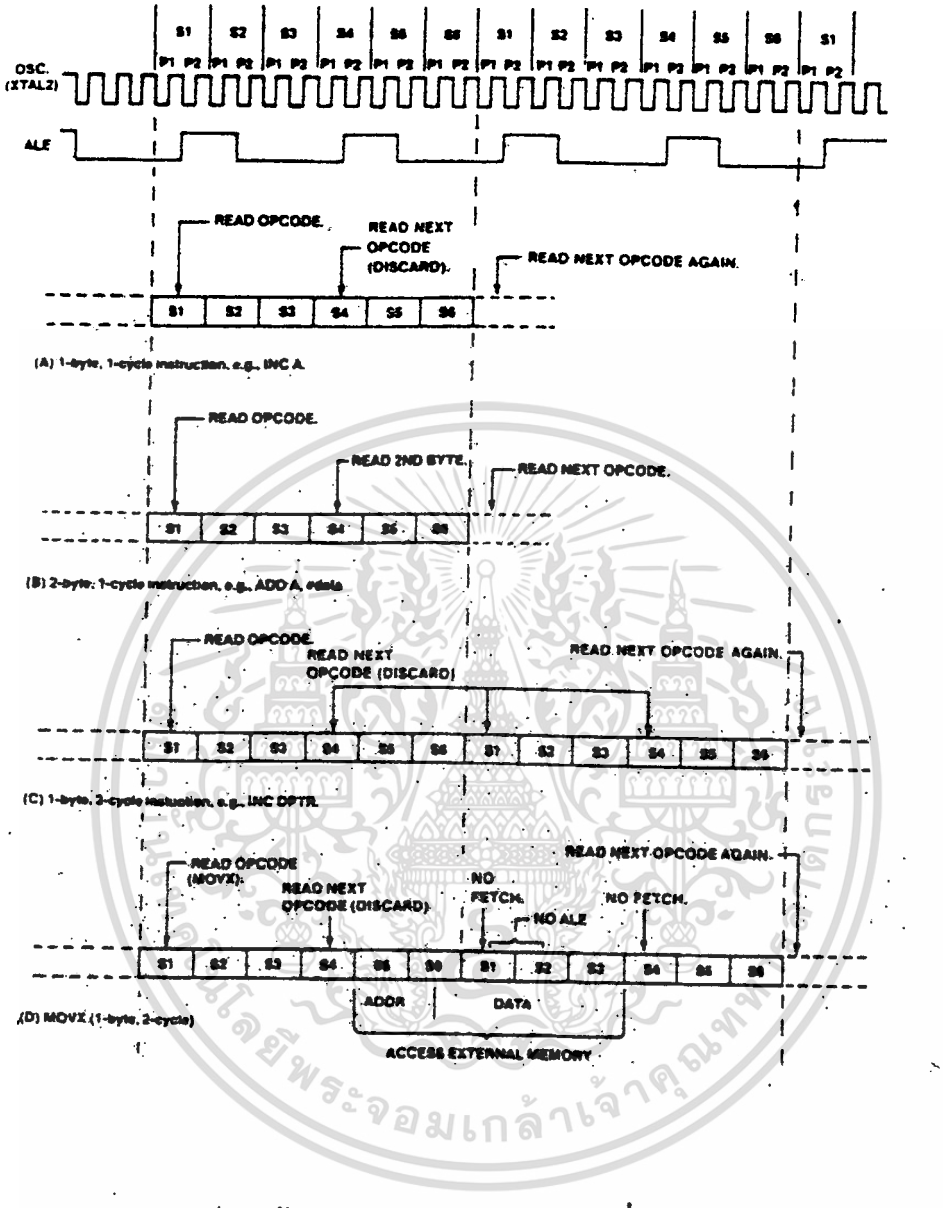


C. CHMOS Only

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูป 2.8 วงจรสร้าง CLOCK ของ 8051

ผังเวลาของ CPU



รูป 2.9 ผังเวลาการทำงานของแต่ละคำสั่ง

การทำงานใน 1 คำสั่งต่ำสุดจะกินเวลาเพียง 1 μS เช่นคำสั่ง INC A ซึ่งเป็นคำสั่ง 1 ไบต์ โดยใน 1 แมชชีนไซเคิล จะประกอบด้วย Clock 12 ลูก (ปกติแล้ว CPU จะ RUN ด้วยความเร็วเท่ากับ 12 MHz ดังนั้น 12 Clock จะกินเวลาเท่ากับ (1/12)*12 = 1μS

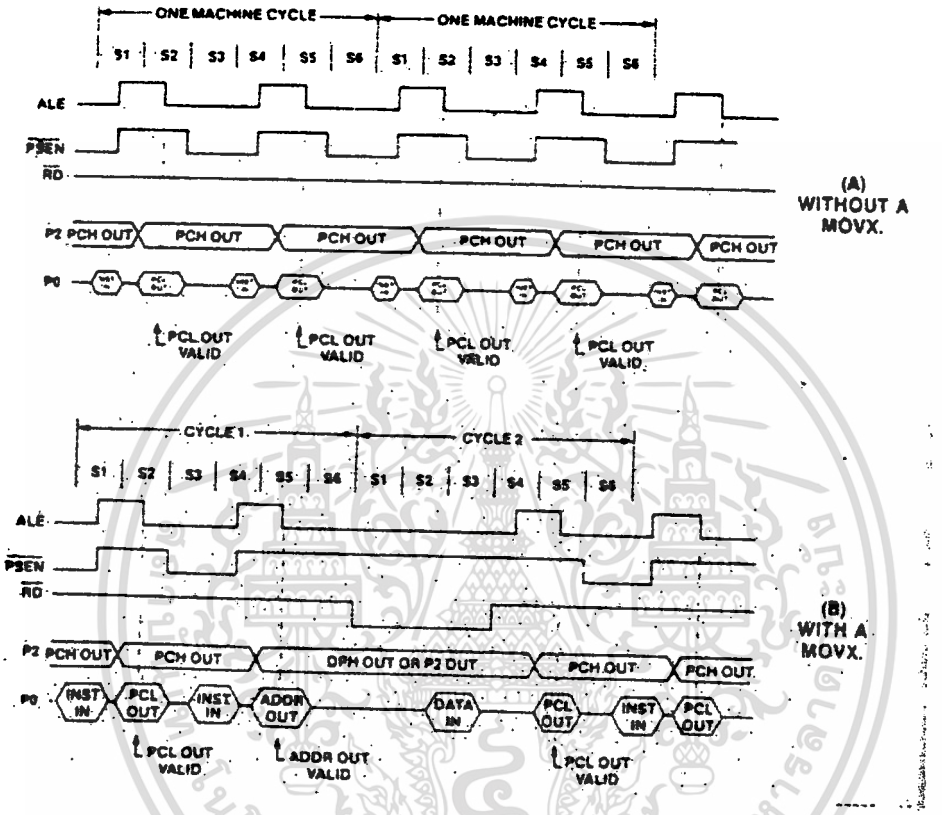
รูป 2.9 (A) แสดงการทำงานของคำสั่ง INC A ซึ่งเป็นคำสั่ง 1 ไบต์ ทำงานเสร็จภายใน 1 แมชชีนไซเคิล

รูป 2.9 (B) แสดงการทำงานของคำสั่ง ADD A,#DATA ซึ่งเป็นคำสั่ง 2 ไบต์ แต่ทำงานเสร็จภายใน 1 แมชชีนไซเคิล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น ไม่อนุญาตให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูป 2.9 (C) แสดงการทำงานของคำสั่ง INC DPTR ซึ่งเป็นคำสั่ง 1 ไบต์ แต่ทำงานเสร็จใน 2 แมชชีนไซเคิล

รูป 2.9 (D) แสดงการทำงานของคำสั่ง MOVX ซึ่งเป็นคำสั่ง 1 ไบต์ แต่ทำงานเสร็จใน 2 แมชชีนไซเคิล



รูป 2.10 แสดงผังเวลาการติดต่อกับหน่วยความจำภายนอก

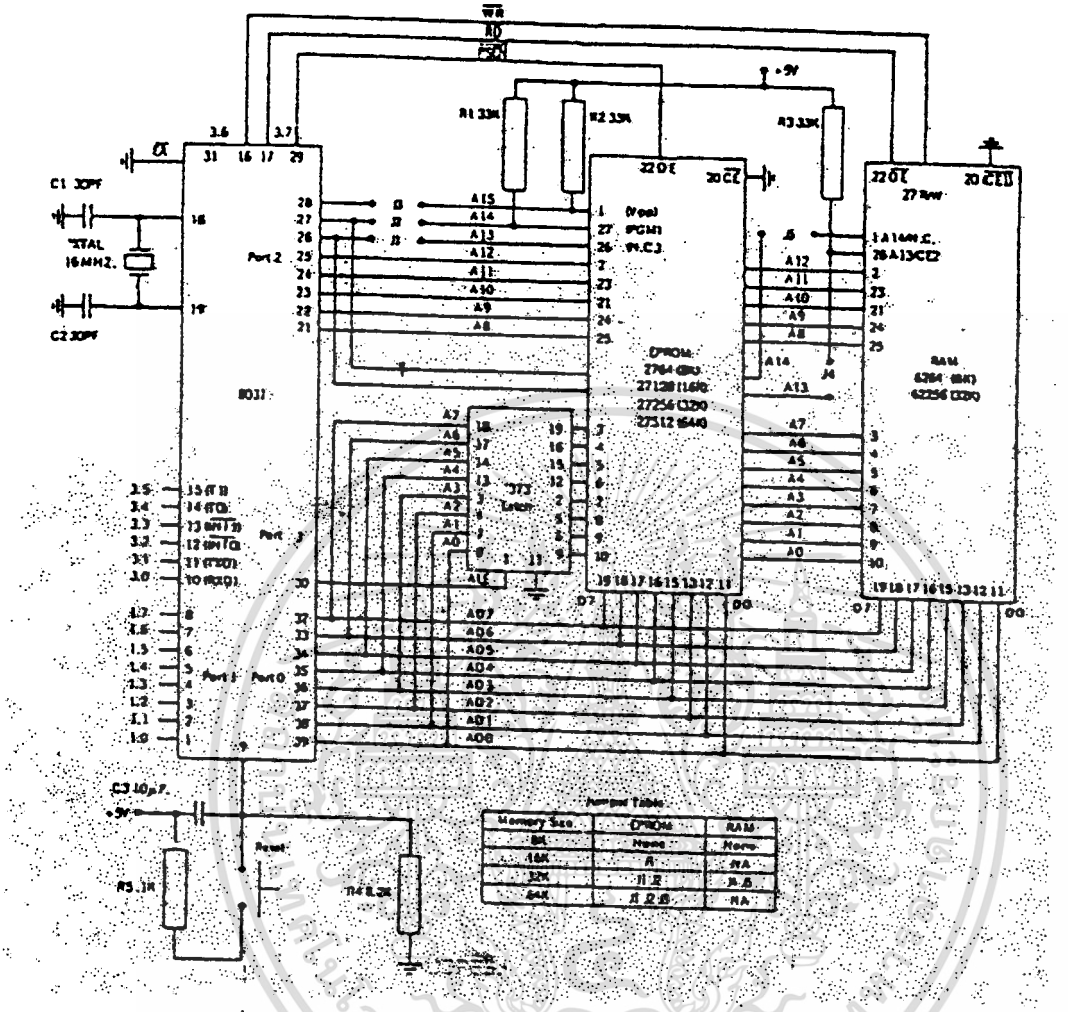
รูป 2.10 (A) เป็นผังเวลาของสัญญาณที่เกี่ยวข้องกับเฟิทซ์ เมื่อส่วนของ Program Memory อยู่ภายนอก ดังนั้นสัญญาณที่จะนำไปใช้อ่าน Op – Code จาก Program Memory ก็คือ $\overline{\text{PSEN}}$ ซึ่งจะแอคทีฟ 2 ครั้งใน 1 แมชชีนไซเคิล ดังนั้นสัญญาณที่ใช้อ่านข้อมูลจาก Program Memory จะใช้สัญญาณ $\overline{\text{PSEN}}$

รูป 2.10 (B) เป็นผังเวลาของสัญญาณของเวลาที่เกี่ยวข้องกับการอ่านข้อมูลจาก Data Memory สัญญาณ $\overline{\text{PSEN}}$ จะมีเพียง 1 ลูก เพราะช่วงเวลาถัดมาจะเป็นช่วงเวลาในการอ่านข้อมูลจาก Data Memory โดยใช้สัญญาณ $\overline{\text{RD}}$ (อาจสรุปได้ง่ายๆ ว่า การอ่านข้อมูลจาก Program Memory จะใช้สัญญาณ $\overline{\text{PSEN}}$ และการอ่านข้อมูลจาก Data Memory จะใช้สัญญาณ $\overline{\text{RD}}$ ส่วนสัญญาณ ALE คือ สัญญาณที่ใช้ในการ Latch Address A0 – A7 นั่นเอง)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การต่อหน่วยความจำ Program Memory และ Data Memory ภายนอกชิพ

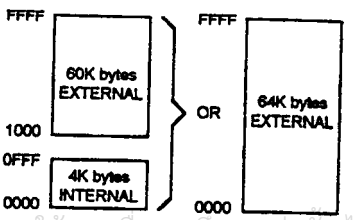
การต่อหน่วยความจำดังแสดงในรูป 2.11



รูป 2.11 การต่อหน่วยความจำโปรแกรม และดาต้าภายนอกชิพ

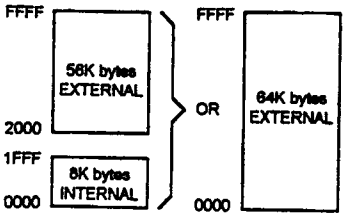
การแบ่งประเภทของหน่วยความจำ

หน่วยความจำที่ใช้กับ MCS – 51 มีอยู่ด้วยกัน 2 ชนิดคือ Program Memory และ Data Memory ซึ่งเป็นหน่วยความจำที่ใช้เก็บโปรแกรมสั่งงานบรรจุอยู่ภายในชิพ 8051 ส่วนที่เป็น Program Memory ก็คือ ROM ขนาด 4 KB นั่นเอง แต่ถ้าเป็นเบอร์ 8052 ก็คือ ROM ขนาด 8 KB ดังแสดงในรูป 2.12 และ 2.13



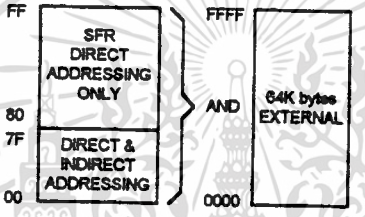
รูป 2.12 ผังหน่วยความจำสำหรับเก็บโปรแกรมสำหรับเบอร์ 8051

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

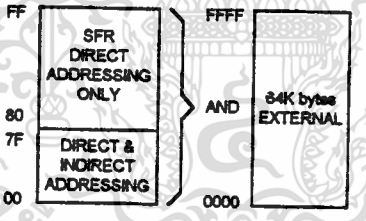


รูป 2.13 ผังหน่วยความจำสำหรับเก็บโปรแกรมสำหรับเบอร์ 8052

Data Memory เป็นหน่วยความจำที่ใช้เก็บข้อมูล หน่วยความจำนี้สามารถเขียนข้อมูลลงไปและอ่านข้อมูลออกมาได้ ซึ่งเป็นหน่วยความจำข้อมูลภายในชิพ มีเพียง 128 ไบต์ สำหรับเบอร์ 8051 และ 256 ไบต์สำหรับเบอร์ 8052 ส่วนหน่วยความจำภายนอกชิพมี 64 KB ดังแสดงในรูป 2.14 และ 2.15



รูป 2.14 ผังหน่วยความจำสำหรับ Data Memory เบอร์ 8051



รูป 2.15 ผังหน่วยความจำสำหรับ Data Memory เบอร์ 8052

บางครั้งอาจสงสัยว่าตำแหน่งของหน่วยความจำสำหรับโปรแกรมและข้อมูล มีตำแหน่งที่ซ้อนกัน CPU จะรู้ได้อย่าไรว่าจะติดต่อหน่วยความจำภายในหรือภายนอก บริษัทอินเทลได้ออกแบบแยกคำสั่งออกเป็น 3 ส่วนคือ

MOV ใช้ติดต่อกับ RAM ภายใน

MOVC ใช้ติดต่อ Program Memory

MOVX ใช้ติดต่อ Data Memory ภายนอกชิพ โดยระบุตำแหน่งผ่านทางรีจีสเตอร์ DPTR

ชิพเบอร์ 8052 จะมีพื้นที่บริเวณ 80H – FFH ซึ่งถ้าจะเขียนอ่านข้อมูล ณ. บริเวณนี้ จะต้องเข้าถึงข้อมูลโดยอ้อมเท่านั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

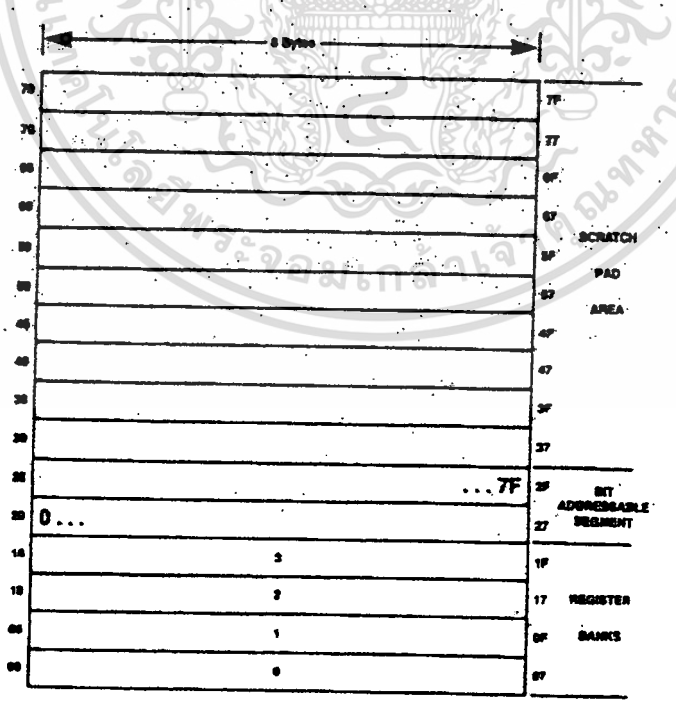
พื้นที่หน่วยความจำที่เข้าถึงข้อมูลโดยทางอ้อมเท่านั้น (Indirect Address Area)

พื้นที่หน่วยความจำบริเวณ (80H – FFH) เป็นพื้นที่ซ้อนกันอยู่อย่างละ 128 ไบต์ โดยส่วนแรกจะเป็น SFR Address และ Indirect Address Area ดังนั้นการเขียนโปรแกรมถ้าจะติดต่อกับ SFR จะต้องใช้คำสั่งแบบเข้าถึงข้อมูลโดยตรงเท่านั้น (Direct Address Area) ส่วนพื้นที่อีกส่วนหนึ่งจะเข้าถึงข้อมูลแบบทางอ้อมเท่านั้น (Indirect Address Only)

พื้นที่หน่วยความจำที่เข้าถึงข้อมูลโดยตรงและโดยอ้อม (Direct and Indirect Address Area)

พื้นที่ 128 ไบต์ล่างสุดจะแบ่งเป็น 3 ส่วนดังรูป 2.16

1. รีจีสเตอร์แบงค์ (Register Bank 0-3) ตั้งแต่ตำแหน่ง (00H – 1FH) จะเป็นส่วนของรีจีสเตอร์แบงค์ (0-3) โดยจะแบ่งเป็นแบงค์ละ 8 ไบต์ รวมแล้วได้ 32 ไบต์ ถ้า CPU ทำงานอยู่และถูกรีเซตก็จะกลับมาทำงานที่แบงค์ 0 เสมอ และ SP จะมาเริ่มต้นที่ตำแหน่ง 07H ทันที
2. บริเวณหน่วยความจำที่ใช้คำสั่งเขียน \ อ่านเกี่ยวกับบิตได้ (Bit Addressable Area) พื้นที่ตั้งแต่ Address (20H-2FH) จำนวน 16 ไบต์ หรือถ้านับเป็นบิตจะได้เท่ากับ 128 บิต
3. บริเวณหน่วยความจำที่ใช้งานทั่วไป (Scratch Pad Area) พื้นที่ตั้งแต่ (30H-7FH) จะเขียนข้อมูลได้ทีละไบต์เท่านั้น ไม่สามารถใช้คำสั่งที่เกี่ยวกับบิตได้



รูป 2.16 ไบต์ของ RAM ที่เข้าถึงข้อมูลแบบทางตรงและทางอ้อม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเอกสารทุกครั้งที่มีการนำไปใช้

รีจีสเตอร์หน้าที่พิเศษ (Special Function Register)

รายละเอียดดังตารางที่ 2.2

ตารางที่ 2.2 แสดงสัญลักษณ์ ชื่อ และตำแหน่งต่างๆ ที่มีอยู่ใน SFR

Symbol	Name	Address
*ACC	Accumulator	0E0H
*B	B Register	0F0H
*PSW	Program Status Word	0D0H
SP	Stack Pointer	81H
DPTR	Data Pointer 2 Bytes	
DPL	Low Byte	82H
DPH	High Byte	83H
*P0	Port 0	80H
*P1	Port 1	90H
*P2	Port 2	0A0H
*P3	Port 3	0B0H
*IP	Interrupt Priority Control	0B8H
*IE	Interrupt Enable Control	0A8H
TMOD	Timer/Counter Mode Control	89H
*TCON	Timer/Counter Control	88H
*T2CON	Timer/Counter 2 Control	0C8H
TH0	Timer/Counter 0 High Byte	8CH
TL0	Timer/Counter 0 Low Byte	8AH
TH1	Timer/Counter 1 High Byte	8DH
TL1	Timer/Counter 1 Low Byte	8BH
+TH2	Timer/Counter 2 High Byte	0CDH
+TL2	Timer/Counter 2 Low Byte	0CCH
+RCAP2H	T/C 2 Capture Reg. High Byte	0CBH
+RCAP2L	T/C 2 Capture Reg. Low Byte	0CAH
*SCON	Serial Control	98H
SBUF	Serial Data Buffer	99H
PCON	Power Control	87H

* = Bit addressable
+ = 8052 only

การติดต่อกับรีจีสเตอร์ SFR นี้ ต้องใช้คำสั่งเข้าถึงข้อมูลทางตรงเท่านั้น (Direct Addressing Mode) ทั้งแบบบิตหรือแบบไบต์ก็ได้เช่น ถ้าต้องการโหลดค่า 21 เข้าที่รีจีสเตอร์ A จะเขียนคำสั่งได้ดังนี้

MOV A,#21H แบบนี้เป็นการเขียนข้อมูลเข้ารีจีสเตอร์ที่ละไบต์ แต่ถ้าต้องการเขียนข้อมูลเข้าทีละบิต จะต้องใช้คำสั่งเกี่ยวกับการเซตบิตเช่น

SETB 0E0H เซตบิต D0 ของรีจีสเตอร์ A

SETB 0F3H เซตบิต D3 ของรีจีสเตอร์ B

CLR 0F0H เคลียร์บิต D0 ของรีจีสเตอร์ B

รายละเอียดเกี่ยวกับตำแหน่งบิตต่างๆ ของรีจีสเตอร์ใดๆ ดังแสดงในตารางที่ 2.3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.3 การจัดวางตำแหน่งบิตของ SFR

SFR MEMORY MAP

8 Bytes

F8								FF
F0	B							F7
E8								EF
E0	ACC							E7
D8								DF
D0	PSW							D7
C8	T2CON		RCAP2L	RCAP2H	TL2	TH2		CF
C0								C7
B8	IP							BF
B0	P3							B7
A8	IE							AF
A0	P2							A7
98	SCON	SBUF						9F
90	P1							97
88	TCON	TMOD	TL0	TL1	TH0	TH1		8F
80	P0	SP	DPL	DPH				PCON 87

Bit Addressable

Figure 5

จากตารางที่ 2.3 จะเห็นได้ว่า รีจิสเตอร์บางตัวไม่สามารถเข้าถึงข้อมูลแบบบิตได้ (Not Bit Addressable)

ตารางที่ 2.4 แสดงค่าต่างๆ ใน SFR หลังจากถูกรีเซต

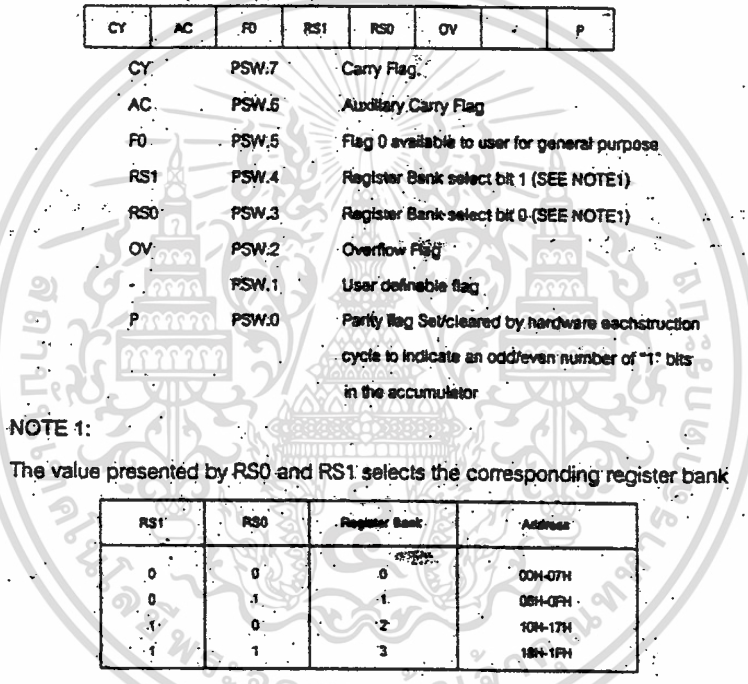
Register	Value in Binary
*ACC	00000000
*B	00000000
*PSW	00000000
SP	00000111
DPTR	
DPH	00000000
DPL	00000000
*P0	11111111
*P1	11111111
*P2	11111111
*P3	11111111
*IP	8051 XXX00000, 8052 XX000000
*IE	8051 0XX00000, 8052 0X000000
TMOD	00000000
*TCON	00000000
*T2CON	00000000
TH0	00000000
TL0	00000000
TH1	00000000
TL1	00000000
TH2	00000000
TL2	00000000
RCAP2H	00000000
RCAP2L	00000000
*SCON	00000000
SBUF	Indeterminate
PCON	HMOS 0XXXXXXX CHMOS 0XXX0000

X = Undefined
* = Bit Addressable
• = 8052 only

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รายละเอียดของรีจีสเตอร์ต่างๆ ใน SFR

- Accumulator (ACC) อยู่ที่ตำแหน่ง 0E0H เป็นรีจีสเตอร์ขนาด 8 บิตที่ใช้ประมวลผลทางคณิตศาสตร์ เช่น บวก ลบ คูณ หาร หรือ เป็นทางผ่านในการเคลื่อนย้ายข้อมูล
- B Register (B) อยู่ที่ตำแหน่ง 0F0H เป็นรีจีสเตอร์ขนาด 8 บิต ที่ใช้คำสั่ง คูณและหาร รวมกับรีจีสเตอร์ A
- Program Status Word (PSW) อยู่ที่ตำแหน่ง 0D0H เป็นรีจีสเตอร์ขนาด 8 บิต แต่ใช้งานเพียง 7 บิต โดย 5 บิตเป็นแฟลก และอีก 2 บิตเป็นบิตเลือกรีจีสเตอร์แบงค์ เลือกได้ 3 แบงค์ คือ แบงค์ 0-3 นั่นเอง รายละเอียดดังรูปที่ 2.17



CY	AC	F0	RS1	RS0	OV	-	P

CY	PSW.7	Carry Flag.
AC	PSW.6	Auxiliary Carry Flag
F0	PSW.5	Flag 0 available to user for general purpose
RS1	PSW.4	Register Bank select bit 1 (SEE NOTE1)
RS0	PSW.3	Register Bank select bit 0 (SEE NOTE1)
OV	PSW.2	Overflow Flag
-	PSW.1	User definable flag
P	PSW.0	Parity flag Set/cleared by hardware eachstruction cycle to indicate an odd/even number of "1" bits in the accumulator

NOTE 1:

The value presented by RS0 and RS1 selects the corresponding register bank

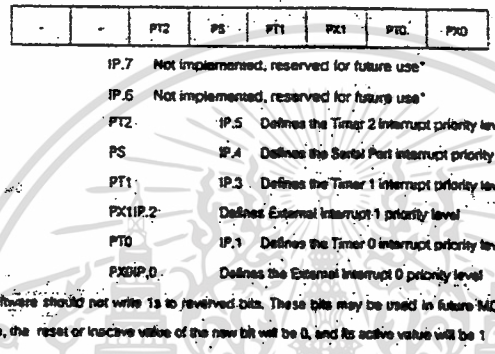
RS1	RS0	Register Bank	Address
0	0	0	00H-07H
0	1	1	08H-0FH
1	0	2	10H-17H
1	1	3	18H-1FH

รูป 2.17 รายละเอียดภายในรีจีสเตอร์ PSW

- Stack Pointer (SP) อยู่ที่ตำแหน่ง 081H เป็นรีจีสเตอร์ขนาด 8 บิตที่ใช้ชี้ตำแหน่งของแสตคทุกครั้งที่ถูกรีเซตก็จะมีค่าเท่ากับ 07H ทันที โดยจะเพิ่มขึ้น 1 เสมอเมื่อทำคำสั่ง CALL, PUSH และจะลดลงเมื่อทำคำสั่ง RET , RETI, POP
- Data Pointer Register อยู่ที่ตำแหน่ง 82H และ 83H เป็นรีจีสเตอร์ขนาด 16 บิต แบ่งเป็น 8 บิตล่าง และ 8 บิตบน มีชื่อว่า DPH และ DPL อยู่ที่ตำแหน่ง 83H และ 82H ตามลำดับ มีประโยชน์ในการชี้แอดเดรสของหน่วยความจำภายนอก ซึ่งจะเป็นการโอนย้ายข้อมูลโดยอ้อม

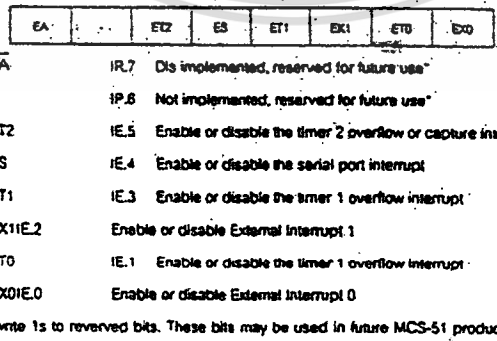
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- Port (0-3) อยู่ที่ตำแหน่ง 80H, 90H, 0A0H, 0B0H ตามลำดับ เป็นพอร์ท 8 บิตมีชื่อว่า P0, P1, P2, และ P3
- Serial Data Buffer (SBUF) อยู่ที่ตำแหน่ง 99H เป็นรีจีสเตอร์ขนาด 8 บิต ที่ใช้เก็บข้อมูลก่อนที่จะส่งข้อมูลอนุกรมออกไป หรือเป็นบัฟเฟอร์เก็บข้อมูลในช่วงที่รับข้อมูลเข้ามา SBUF จะมีอยู่ 2 ตัวคือ รับและส่งแยกกัน การรับส่งข้อมูลอนุกรมมีอยู่ด้วยกัน 4 โหมด ดังจะกล่าวต่อไป
- Interrupt Priority Register (IP) อยู่ตำแหน่ง 0B8H เป็นรีจีสเตอร์ควบคุมลำดับความสำคัญของอินเตอร์รัพท์



รูป 2.18 รายละเอียดภายในรีจีสเตอร์ IP

- Interrupt Enable Register (IE) อยู่ตำแหน่ง 0A8H เป็นรีจีสเตอร์ที่ใช้กำหนดให้ทำหรือไม่ทำอินเตอร์รัพท์ (Enable or Disable) จาก 6 แหล่ง 5 เวคเตอร์ เช่น ถ้าต้องการให้พอร์ทอนุกรมทำการอินเตอร์รัพท์ได้ก็ต้องเซตบิต ES (บิต D4 ใน IE) ด้วยคำสั่ง SETB ES หรือจะห้ามไม่ให้พอร์ทอนุกรมทำการอินเตอร์รัพท์ได้ก็ต้องใช้คำสั่ง CLR ES

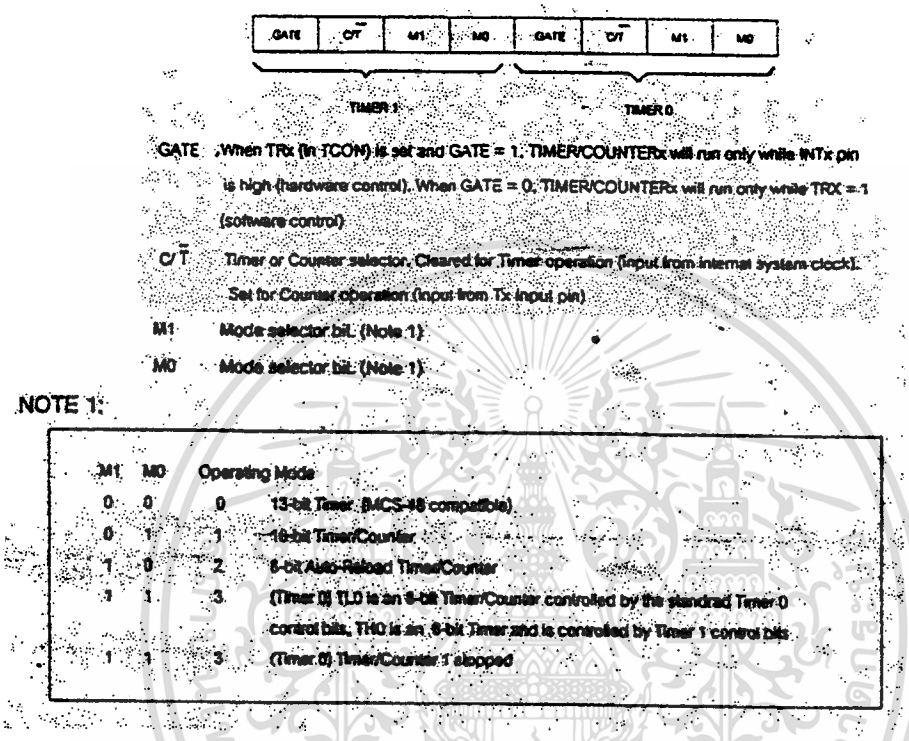


รูป 2.19 รายละเอียดภายในรีจีสเตอร์ IE

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

040600

- Timer / Counter Mode Control Register (TMOD) อยู่ที่ตำแหน่ง 089H เป็นรีจีสเตอร์ที่ใช้โหมดการทำงานเลือกได้ทั้ง 4 โหมด โดยเลือกที่บิต M0 และ M1 ส่วนบิต Gate เป็นบิตที่ใช้เลือกการ สตาร์ท Timer / Counter โดยควบคุมร่วมกับ TRx ใน TCON ส่วนบิต C/$\overline{T}$ ใช้เลือก Timer หรือ Counter



รูป 2.20 รายละเอียดภายในรีจีสเตอร์ TMOD

- Timre / Counter Control Register (TCON) อยู่ที่ตำแหน่ง 88H เป็นรีจีสเตอร์ที่ใช้ควบคุมให้ Timer / Counter เริ่มนับโดยเซตที่บิต TRx (TR0 หรือ TR1 นั่นเอง) อีกหน้าที่หนึ่งใช้แสดงผลการเกิดโอเวอร์โฟลว์ของ Timer / Counter โดยจะแสดงที่บิต TFx (TF0 หรือ TF1 นั่นเอง) อีกหน้าที่หนึ่งคือแสดงสถานะเมื่อเกิดอินเตอร์รัพท์จากภายนอก โดยแสดงที่บิต IE0 หรือ IE1 หน้าที่สุดท้ายคือบิตเลือกสัญญาณอินเตอร์รัพท์ว่าจะเอาแบบ Trig ที่ขอบขาลงหรือ Trig ที่ระดับศูนย์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

TF1	TR1	TF0	IE1	IT1	IE0	IT0

TF1	TCON.7	Timer 1 overflow flag. Set by hardware when the Timer/Counter 1 overflows. Cleard by hardware as processor vector to the interrupt savice routine
TR1	TCON.6	Timer 1 run control bit. Set/cleared by software to turn Timer/Counter 1 ON/OFF
TF0	TCON.5	Timer 0 overflow flag. Set by hardware when the Timer/Counter 0 overflows. Cleard by hardware as processor vector to the interrupt savice routine
TR0	TCON.4	Timer 0 run control bit. Set/cleared by software to turn Timer/Counter 0 ON/OFF
IE1	TCON.3	External interrupt 1 edge flag. Set by hardware when External interrupt edge is detect. Cleared by hardware when interrupt is processored.
IT1	TCON.2	Interrupt type 1 control bit. Set/cleared by hardware when External interrupt edge/low level triggered External interrupt.
IE0	TCON.1	External interrupt 0 edge flag. Set by hardware when External interrupt edge is detect. Cleared by hardware when interrupt is processored.
IT0	TCON.0	Interrupt type 0 control bit. Set/cleared by hardware when External interrupt edge/low level triggered External interrupt.

รูป 2.21 รายละเอียดภายในรีจีสเตอร์ TCON

- Power Control Register (PCON) อยู่ที่ตำแหน่ง 87H เป็นรีจีสเตอร์ที่ใช้ควบคุมการประหยัดพลังงาน มีการทำงาน 2 โหมดคือ Ideal Mode และ Power Down Mode และยังใช้เพิ่มอัตรา Baud Rate เป็น 1 เท่าหรือ 2 เท่าโดยเซตที่บิต SMOD

SMOD	-	-	-	GF1	GF0	PD	IDL

SMOD	Double baud rate bit. If Timer 1 is user to gaterate and SMOD =1, the baud- rate is doybled when the Serial Port is used in mode 1,2, or 3.
-	Not implemented, reserved for future use.*
-	Not implemented, reserved for future use.*
-	Not implemented, reserved for future use.*
GF1	General purpose flag bit
GF0	General purpose flag bit
PD	Power Down bit. Setting this bit activates Power Down operation in the 80C51BH. (Available only in CHMOS).
IDL	Idle Mode bit. Setting this bit activates Idle Mode operation in the 80C51BH. (Availeble only in CHMOS).

If 1s are written to PD and IDL at the same time, PD take precedance.

*User software should not write 1s to reserved bits. These bits may bi used in future MCS-51 products to invoke new features. in that case, the reset or inactive value of the new bit will be 0, and its active value will be 1

รูปที่ 2.22 รายละเอียดภายในรีจีสเตอร์ PCON

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การสื่อสารข้อมูลอนุกรม

การสื่อสารข้อมูลอนุกรมเป็นการรับหรือส่งข้อมูลในลักษณะของกลุ่มบิตคราวละ 1 บิต เรียงลำดับเรื่อยไปจนสิ้นสุด การสื่อสารแบบนี้จะมีข้อแตกต่างจากการสื่อสารแบบขนานเป็นอย่างมาก เนื่องจากข้อมูลมีการโอนย้ายมาพร้อมกัน จึงมีความจำเป็นต้องใช้จำนวนเส้นสัญญาณมากขึ้นตามจำนวนบิตของข้อมูลด้วย ในขณะที่การสื่อสารแบบอนุกรมนั้นต้องการเส้นสัญญาณเพียงสองหรือสามเส้นเท่านั้น ดังนั้นการสื่อสารแบบขนานจึงไม่เหมาะสมในการสื่อสารกับอุปกรณ์ภายนอกเป็นระยะทางไกลๆ

ความเร็วของการสื่อสารข้อมูลอนุกรม

เนื่องจากการสื่อสารข้อมูลแบบอนุกรมเป็นการรับส่งข้อมูลในลักษณะกลุ่มของบิตข้อมูล ดังนั้นจึงต้องให้ความสนใจในการพิจรณาถึงเรื่องอัตราความเร็วในการส่งบิตเหล่านี้เป็นลำดับแรก โดยทั่วไปมักจะระบุกันในหน่วยของจำนวนบิตของข้อมูลภายในเวลา 1 วินาทีเรียกว่า อัตราบอด ซึ่งได้แก่ 110, 150, 300, 1200, 2400, 4800, 9600, 19200 บอด ข้อมูลทั้ง 8 บิตนี้หากว่าถูกส่งออกมาด้วยอัตรา 2400 บอด จะใช้เวลาในการส่งข้อมูล 1 บิตเท่ากับ 1/2400 หรือ 416 μS และเวลาในการส่งข้อมูลทั้ง 8 บิต มีค่าเท่ากับ 8*416 หรือ 3328 μS

รูปแบบการส่งข้อมูลอนุกรม

การสื่อสารอนุกรมแบบอะซิงโครนัส จะใช้การแปลงข้อมูลให้เป็นอนุกรมแล้วเพิ่มเติมบิตบางอย่างร่วมไปกับการส่งข้อมูลจริงซึ่งได้แก่

1. บิตเริ่มต้น (Start bit) มีหน้าที่สำหรับบ่งบอกให้ทราบถึงตำแหน่งจุดเริ่มต้นก่อนบิตข้อมูล ตามปกติแล้วค่าของบิตเริ่มต้นจะเป็นระดับลอจิกต่ำ
2. บิตแสดงสภาวะความเป็นเลขคู่หรือเลขคี่ (Parity Bit) มีหน้าที่เพื่อตรวจสอบความถูกต้องของข้อมูล โดยทั่วไปมักเรียกว่าบิตพาริตี้ และจะนำไปต่อท้ายบิตข้อมูล ค่าของบิตนี้ขึ้นอยู่กับจำนวนค่าของบิตที่เป็น 1 ซึ่งจะเป็นได้สองลักษณะคือพาริตี้คู่ หรือพาริตี้คี่ ตัวอย่างเช่นระบบที่ติดต่อกันโดยระบุว่าจะใช้พาริตี้คี่ ทางด้านส่งจะนำค่าข้อมูลมาพิจรณาหาจำนวนบิตที่มีค่า 1 ถ้าเป็นเลขจำนวนคู่อยู่แล้วค่าของบิตพาริตี้จะมีค่าเป็น 0 แต่หากว่าจำนวนของบิตที่มีค่าเป็น 1 เป็นเลขจำนวนคี่ ค่าของบิตพาริตี้ก็จะมีค่าเป็น 1 การพิจรณาทางด้านรับเป็นการตรวจสอบจำนวนบิตที่มีค่า 1 ของข้อมูลที่ได้รับมาทั้งหมดรวมทั้งบิตพาริตี้ ถ้ามีค่าเป็นเลขจำนวนคู่ แสดงว่าข้อมูลที่ได้รับมานี้ถูกต้อง แต่หากไม่เป็นเลขจำนวนคู่แสดงว่าเกิดการผิพลาดของข้อมูลขึ้นเป็นต้น
3. บิตสุดท้าย (Stop Bit) เป็นบิตที่เพิ่มขึ้นเพื่อบอกถึงขอบเขตการสิ้นสุดของกลุ่มบิตข้อมูลบิตสุดท้ายนี้สามารถโปรแกรมบิตข้อมูลได้คือ 1 บิต 1 □ บิต และ 2 บิต ดังนั้นกรณี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ของการส่งข้อมูล 8 บิต หากข้อมูลถูกส่งออกไปด้วยอัตรา 2400 บอด เวลาโดยรวมในการส่งข้อมูล 1 ไบต์ จะมีค่าเป็น 12*416 หรือ 4.99 mS

การส่งข้อมูลอนุกรมของ 8051

พอร์ทอนุกรมของ 8051 มีโครงสร้างการทำงานในแบบที่เรียกว่า ฟูลดูเพล็กซ์ (Full Duplex) ในการรับและส่งข้อมูลอนุกรมได้ในเวลาเดียวกันโดยทางด้านวงจรของตัวส่ง (Transmitter) ประกอบด้วยข้อมูลออกไปยังพอร์ทอนุกรมทางขาสัญญาณ TxD (พอร์ท 3.1) ส่วนวงจรด้านตัวรับ (Receiver) ประกอบด้วย SBUF เช่นเดียวกับสัญญาณข้อมูลอนุกรมที่รับเข้ามาทางขาสัญญาณ RxD (พอร์ท 3.0)

พอร์ทอนุกรมของ 8051 สามารถโปรแกรมการทำงานได้หลายโหมดด้วยกัน โดยเลือกที่บิต SM0 และ SM1 ซึ่งอยู่ในรีจิสเตอร์ควบคุม SCON การทำงานทั้ง 4 โหมดของพอร์ทอนุกรมมีดังนี้

โหมด 0

ใช้รับส่งข้อมูล 8 บิต โดยการส่งจะเลื่อนออกทีละบิต โดยส่งบิต D0 ออกไปก่อนทางขา RxD และไม่มีการส่ง Start Bit แต่จะส่ง Shift Clock ทางขา TxD ความเร็ว 1/12 เท่าของ CPU Clock

โหมด 1

ใช้สำหรับการเชื่อมต่ออนุกรมแบบ UART (Universal Asynchronous Receiver / Transmitter) โดยส่งแบบ 10 บิต ประกอบด้วยข้อมูล 8 บิต Start 1 บิต และ Stop 1 บิต สามารถเปลี่ยนแปลงอัตราความเร็วในการส่งข้อมูลได้ โดยขึ้นกับบิต SMOD ใน PCON และอัตราโอเวอร์โฟล์วของ Timer 1

โหมด 2

ใช้สำหรับการเชื่อมต่ออนุกรมแบบ UART โดยส่งแบบ 11 บิต และกำหนดอัตราความเร็วในการส่งข้อมูลเท่ากับ 1/32 และ 1/64 ของ CPU Clock โดยโปรแกรมที่บิต SMOD ใน PCON

โหมด 3

ใช้สำหรับการเชื่อมต่ออนุกรมแบบ UART โดยส่งแบบ 11 บิต และกำหนดอัตราความเร็วในการส่งข้อมูลได้โดยควบคุมที่บิต SMOD และอัตราโอเวอร์โฟล์วของ Timer 1 นอกจากนี้โหมด 2 และ 3 ยังมีการดำเนินการอีกแบบหนึ่ง โดยสามารถนำมาใช้ประโยชน์ในการสื่อสารข้อมูล แบบที่มีไมโครโปรเซสเซอร์หลายตัวทำงานร่วมกันได้ซึ่งมีชื่อเรียกว่า Multiprocessor Mode

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Multiprocessor Mode ในโหมดนี้เราจะใช้ไมโครโปรเซสเซอร์ 1 ตัว สำหรับเป็น Master และอีก 0-256 ตัว เป็น Slave รีจิสเตอร์ที่ใช้ควบคุมการรับส่งข้อมูลอนุกรมมีรายละเอียดดังรูปที่ 2.23

SM0	SM1	SM2	REN	TB8	RB8	TI	RI

SM0	SCON.7	Serial Port mode specifier (NOTE 1)
SM1	SCON.6	Serial Port mode specifier (NOTE 1)
SM2	SCON.5	Enables the multiprocessor communication feature in mode 2&3. In mode 2 or 3, if SM2 is set to then RI will not be activated if the received 9th data bit (RB8) is 0.In mode 1, if SM2 = 1 then RI will not be activated if a valid stop bit was not received. In mode 0, SM2 should be 0.
REN	SCON.4	Set/Cleared by software to Enable/Disable reception.
TB8	SCON.3	The 9th bit that will be transmitted in mode 2&3. Set/Cleared by software.
RB8	SCON.2	In modes 2&3, is the 9th data bit that was received. In mode 1, if SM2 = 0, RB8 is the stop bit that was received. In mode 0,RB8 is not used.
TI	SCON.1	Transmit interrupt flag. Set by hardware at the end of the 8th bit time in mode 0, or at the beginning of the stop bit in the other modes.Must be clear by software.
RI	SCON.0	Receive interrupt flag. Set by hardware at the end of the 8th bit time in mode 0, or halfway through the stop bit in the other modes.Must be clear by software.

รูป 2.23 แสดงรายละเอียดใน SCON

อินเตอร์รัพท์ของการสื่อสารแบบอนุกรม

เนื่องจากการส่งหรือรับข้อมูลอนุกรมจะส่งทีละไบต์ 8051 จึงได้กำหนดให้บิตหรือแฟลกสถานะที่จัดรวมอยู่ในรีจิสเตอร์ SCON เช่น แฟลก T1 ซึ่งจะมีค่าเป็น 1 เมื่อข้อมูลได้ทำการส่งออกไปภายนอกเสร็จสิ้นแล้ว และแฟลก R1 ซึ่งจะมีค่าเป็น 1 เพื่อให้รับรู้ว่าได้รับข้อมูลผ่านเข้ามาทางพอร์ทอนุกรมเสร็จสิ้นแล้ว เมื่อแฟลก R1, T1 นี้มีค่าเป็น 1 จะมีผลทำให้เกิดการอินเตอร์รัพท์ขึ้น ดังนั้นภายในโปรแกรมรับหรือส่งข้อมูลจะต้องทำการตรวจสอบจากสภาวะของแฟลกเหล่านี้เอง ว่าเป็นการส่งข้อมูลหรือรับข้อมูล (Vector ของ T1 R1 อยู่ที่ 0023H)

กระบวนการรับและส่งข้อมูลอนุกรมของ 8051

การส่งข้อมูลออกทางพอร์ทอนุกรมของ 8051 จะเริ่มต้นขึ้นภายหลังเมื่อมีการเขียนข้อมูลลงใน SBUF ข้อมูลนี้จะถูกเลื่อนทีละบิต และส่งสัญญาณออกไปภายนอกโดยอัตโนมัติ เมื่อข้อมูลเหล่านี้ถูกส่งออกไปครบถ้วนแล้ว จะทำให้ค่าแฟลก T1 เป็น 1 เพื่อแจ้งให้ทราบว่าขณะนี้ SBUF ว่างและพร้อมที่จะส่งข้อมูลไบต์ต่อไปแล้ว ในกรณีที่เขียนข้อมูลใหม่ลงในรีจิสเตอร์ SBUF โดยไม่รอให้แฟลก T1 มีค่าเป็น 1 ก่อนจะมีผลทำให้ข้อมูลที่ส่งออกไปผิดพลาดได้

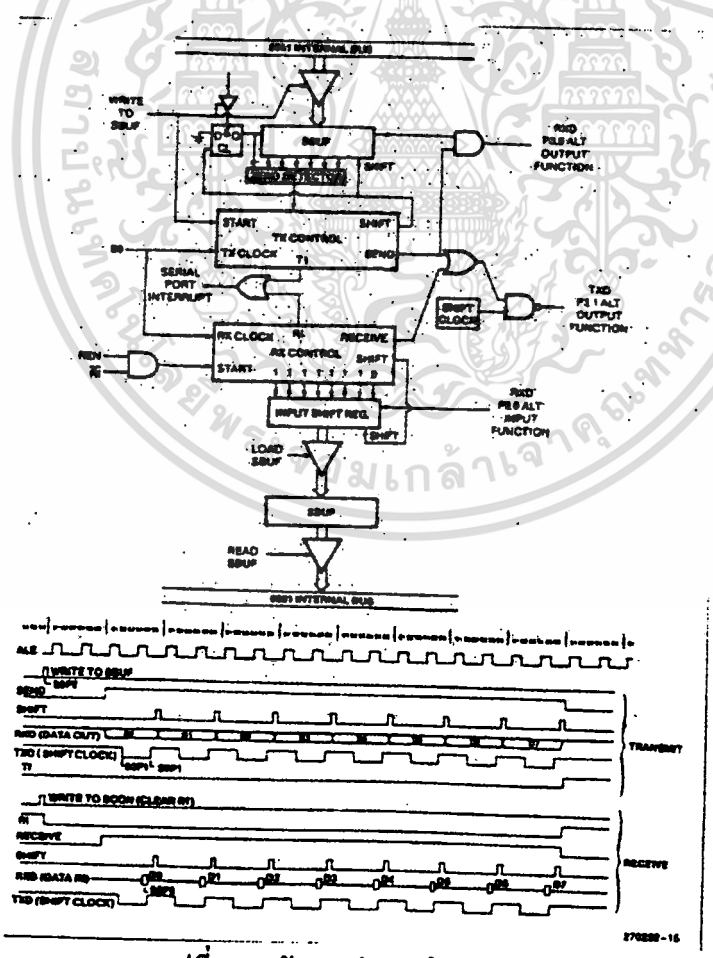
สำหรับการรับข้อมูลจากพอร์ทอนุกรมจะต้องเริ่มต้นโดยการเซตค่าดังนี้ (Receiver Enable) ให้มีค่าเป็น 1 ก่อน หลังจากนั้นเมื่อมีข้อมูลภายนอกถูกส่งเข้ามายัง 8051 ทีละบิตจนครบ และเมื่อ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารนี้ทุกครั้งที่มีการนำไปใช้

บิตสุดท้ายถูกเลื่อนเข้ามาเรียบร้อยแล้ว ข้อมูลนั้นจะถูกย้ายเข้ามาเก็บไว้ยังรีจีสเตอร์ SBUF และแฟลก R1 ก็จะมีค่าเป็น 1 หลังจากนั้นก็จะเกิดการอินเตอร์รัพท์ขึ้น

การรับส่งข้อมูลอนุกรมโหมด 0

การทำงานของพอร์ทอนุกรม (โหมด 0) เป็นการรับและส่งข้อมูลอนุกรมจำนวน 8 บิต โดยใช้เพียงขาสัญญาณ RxD เท่านั้น ส่วนขาสัญญาณ TxD จะนำไปใช้เพื่อขาสัญญาณนาฬิกาในการให้จังหวะการเลื่อนข้อมูลกับวงจร เลื่อนบิตภายนอกสำหรับอัตราความเร็ว จะถูกกำหนดไว้คงที่ที่ค่า 1/12 ของความถี่ออสซิเลเตอร์ จากรูปที่ 2.24 แสดงให้เห็นถึงแผนภาพของเวลาสัญญาณต่างๆ ในโหมด 0 เมื่อมีการรับและส่งข้อมูล 1 ไบต์ โดยสัญญาณนาฬิกาในการเลื่อนบิตนี้จะเกิดขึ้นภายในตัวของ 8051 เอง เนื่องจากโหมดนี้ไม่มีการส่ง Start Bit และ Stop Bit ดังนั้นจึงจำเป็นที่จะต้องส่งสัญญาณ Shift Clock ออกไปเพื่อใช้ Synchronize ระหว่างฝ่ายรับและฝ่ายส่ง โดยจะใช้ขา TxD ส่วนการรับข้อมูลจะรับข้อมูลเข้าทางขา RxD และรับ Shift Clock เข้าทางขา TxD ถ้าความถี่ออสซิเลเตอร์มีค่าเท่ากับ 12 MHz ก็จะส่งได้ถึง 1 ล้านบอด ซึ่งโหมด 0 เป็นโหมดที่ส่งข้อมูลได้เร็วที่สุด รายละเอียดของผังเวลาในการรับส่งดังแสดงในรูป 2.24

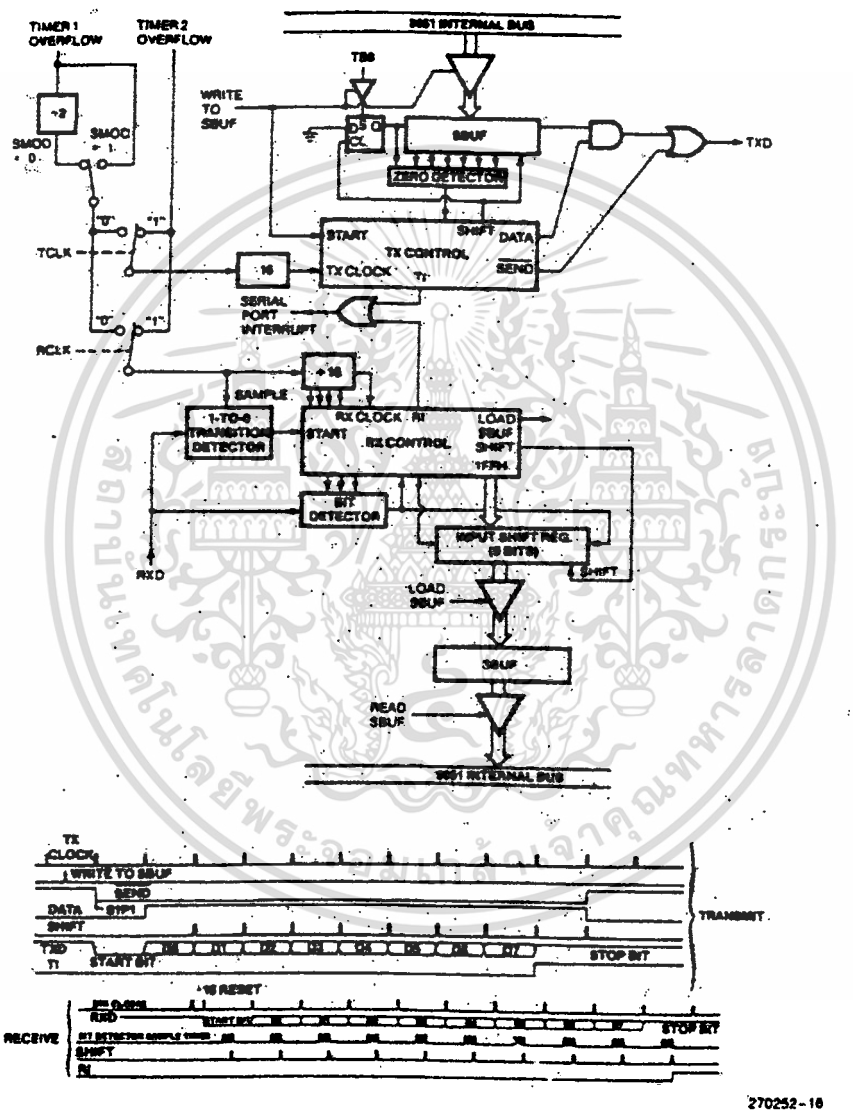


รูปที่ 2.24 ผังการทำงานโหมด 0

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การรับส่งข้อมูลอนุกรมโหมด 1

การทำงานในโหมด 1 เป็นการสื่อสารข้อมูลอนุกรมจำนวน 10 บิต ประกอบด้วยบิตเริ่มต้น 1 บิต บิตข้อมูล 8 บิต และบิตสุดท้ายอีก 1 บิต โดยข้อมูลจะถูกส่งออกมาทาง TxD และรับเข้ามาทางขาสัญญาณ RxD ในส่วนของข้อมูล 8 บิต ที่ได้รับหรือทำการส่งออก จะเป็นบิตนัยสำคัญต่ำเป็นลำดับแรกส่งทางฝ่ายรับ ค่าของ Stop Bit จะส่งเข้ามาจัดเก็บไว้ในบิต RB8 ภายในรีจีสเตอร์ SCON สำหรับอัตราความเร็วในการส่งข้อมูลของโหมด 1 นั้น สามารถกำหนดเลือกได้จาก Timer 1 ผังเวลาและการทำงานดังแสดงในรูปที่ 2.25



รูปที่ 2.25 ผังการทำงานโหมด 1

ดังได้กล่าวแล้วว่าการส่งข้อมูลอนุกรมโหมด 1 สามารถเปลี่ยนแปลงความเร็วได้โดยการใช้ Timer 1 ทำหน้าที่เป็นตัวกำเนิดอัตราการส่งข้อมูล และใช้แฟลกที่เกิดขึ้นจากการโอเวอร์ของ Timer 1 โดยโปรแกรม Timer 1 ทำงานในโหมด 2 8-Bit Automatic Reload

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$\text{ความถี่อัตราบอด} = \frac{(2)^{SMOD}}{(32)} \times \text{อัตราโอเวอร์โฟล์วของ Timer 1}$$

$$\text{หรือความถี่อัตราบอด} = \left|\frac{(2)^{SMOD}}{(32)}\right| \times \left|\frac{f_{osc}}{12 \times (256 - TH1)}\right|$$

โดย SMOD เป็นค่าของบิตภายในรีจีสเตอร์ PCON (มีค่าเป็น 0 หรือ 1) ค่าภายในรีจีสเตอร์ TH1 ซึ่งใช้เป็นค่าสำหรับโหลดซ้ำ

Baud Rate	fOSC	SMOD	TIMER 1		
			C/T	MODE	Reload Value
(MODE 0) Max : 1 MHz	12 MHz	X	X	X	X
(MODE 2) Max : 375 KHz	12 MHz	1	X	X	X
(MODE 2) Min : 187.5 kHz	12 MHz	0	X	X	X
MODE1,3 : 62.5 kHz	12 MHz	1	0	2	FFH
19.2 k	11.059 MHz	1	0	2	FDH
9.6 k	11.059 MHz	0	0	2	FDH
4.8 k	11.059 MHz	0	0	2	FAH
2.4 k	11.059 MHz	0	0	2	F4H
1.2 k	11.059 MHz	0	0	2	E8H
137.5 k	11.059 MHz	0	0	2	1DH
110	6 MHz	0	0	2	72H
110	12 MHz	0	0	1	FEEBH

การรับส่งข้อมูลอนุกรมโหมด 2

โหมดนี้ใช้ทั้งหมด 11 บิต โดยแบ่งเป็น 9 บิตข้อมูล 1 Start Bit 1 Stop Bit โดยบิตที่ 9 ผู้ใช้สามารถกำหนดค่าเองได้ว่าจะส่งค่าอะไรออกไป โดยจะต้องนำไปส่งไว้ในบิต TB8 ในรีจีสเตอร์ SCON ส่วนมากผู้ใช้จะนำบิตนี้มาใช้เป็น Parity Bit โดยโหลดค่ามาจาก Parity Flag ใน PSW ส่วนทางด้านรับบิตที่ 9 จะถูกนำมาเก็บไว้ใน RB8 อัตราความเร็วในการรับส่งข้อมูลขึ้นกับความถี่ออสซิเลเตอร์ของ CPU และค่า SMOD ซึ่งอยู่ในบิต 7 ของ PCON

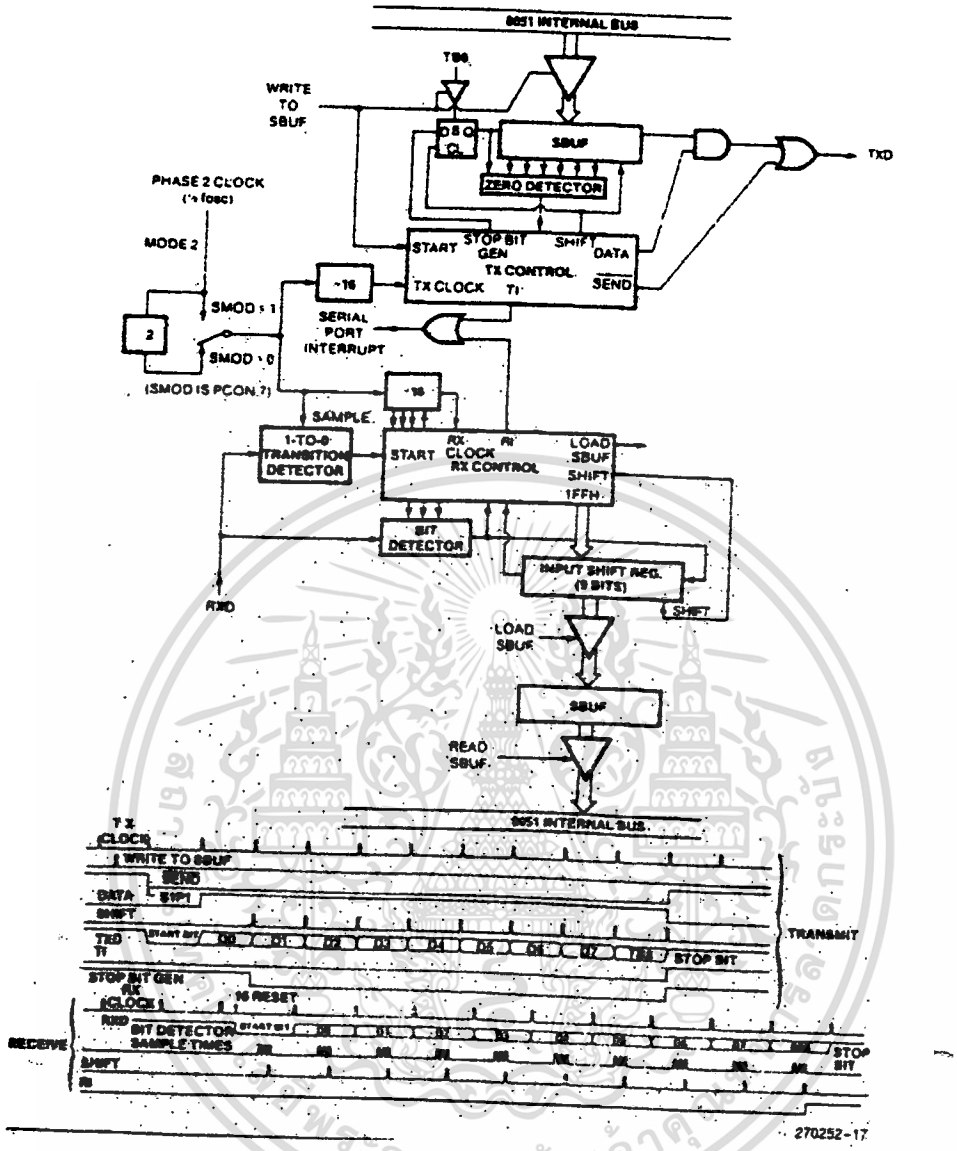
$$\text{MODE 2 BAUD RATE} = \frac{(2)^{SMOD} f_{osc}}{64}$$

ถ้า CPU รันที่ 12 MHz และ SMOD มีค่า 0 และ 1

$$\text{เมื่อ SMOD = 0 จะได้} = \frac{2^0(12)(10^6)}{64} = 187{,}500 \text{ บอด}$$

$$\text{เมื่อ SMOD = 1 จะได้} = \frac{2^1(12)(10^6)}{64} = 375{,}000 \text{ บอด}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

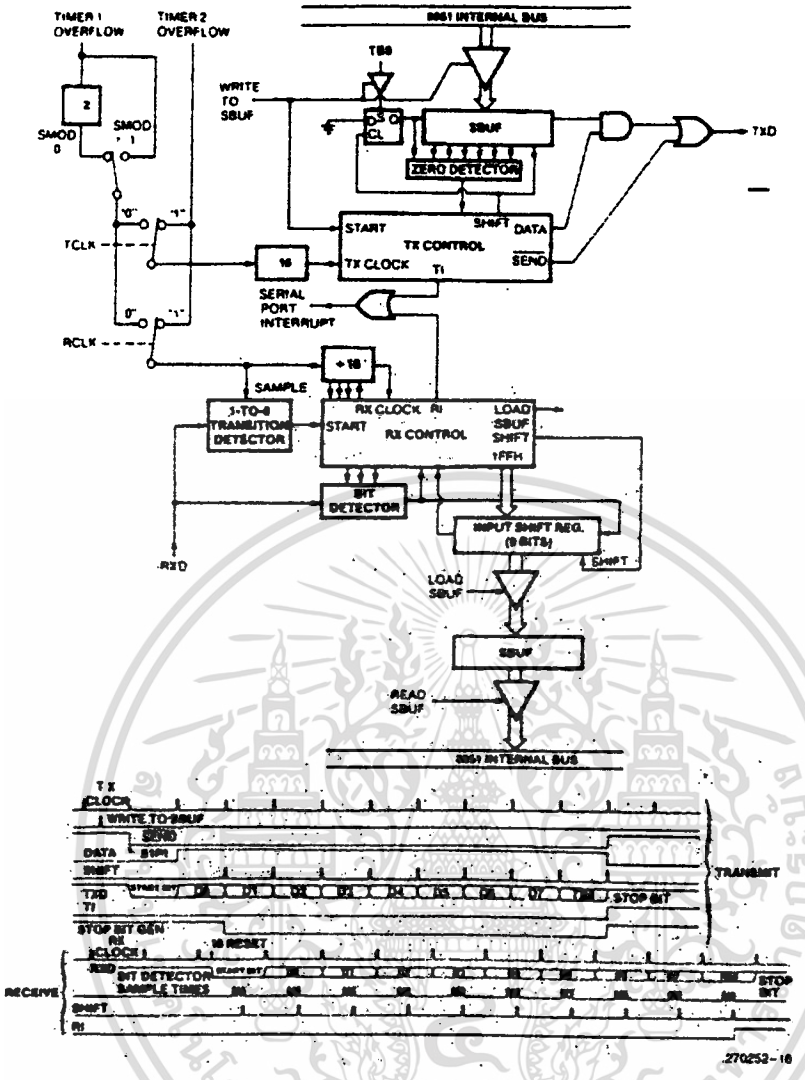


รูปที่ 2.26 ผังการทำงานโหมด 2

การรับส่งข้อมูลอนุกรมโหมด 3

เหมือนกับโหมด 2 ทุกอย่าง เว้นแต่ความเร็วในการรับส่งข้อมูลจะขึ้นกับอัตราโอเวอร์โฟล์วของ Timer 1 หรือ Timer 2 โดยมีผังการทำงานดังรูป 2.28 และการเปลี่ยนแปลงความเร็วดูได้จากตารางที่ 5 หรือคำนวนจากสูตรดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูป 2.27 ผังการทำงานโหมด 3

เมื่อกำหนดให้ Timer 1 ทำงานโหมด 2 Auto Reload

$$\text{MODE 1,3} = \frac{2^{SMOD}}{384(256-TH1)} f_{OSC}$$

$$\text{หรือ} = \frac{K\,(OSCfreg)}{384(256-TH1)}$$

ถ้า SMOD = 0 จะได้ K=1

SMOD = 1 จะได้ K=2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การเชื่อมโยง MCS-51 จำกับหน่วยความจำ

การจัดหน่วยความจำสำหรับ MCS-51 แบ่งได้เป็น 2 ส่วนใหญ่ๆ คือ

- Data Memory

เป็นหน่วยความจำที่ใช้เก็บข้อมูลที่มีความจุได้ถึง 64 KB เป็นหน่วยความจำประเภทอ่านและเขียนได้ (RAM) สัญญาณจาก MCS-51 ที่ใช้ในการอ่านข้อมูลจาก RAM คือ $\overline{RD}$ และสัญญาณที่ใช้ในการเขียนข้อมูลลง RAM คือ $\overline{WR}$

- RAM (Random Access Memory)

เป็นหน่วยความจำที่สามารถเขียนและอ่านข้อมูลได้ โดยข้อมูลจะสูญหายทันทีถ้าขาดไฟเลี้ยง แบ่งเป็น 2 ชนิดคือ Static RAM และ Dynamic RAM

Static RAM

เป็นหน่วยความจำชนิดหนึ่งที่ง่ายในการนำมาใช้งานแต่ราคาจะสูง มีใช้กันอย่างแพร่หลายมีอยู่ด้วยกันหลายเบอร์ตามขนาดความจุของหน่วยความจำ

Dynamic RAM

ราคาถูกกว่า Static RAM แต่ยุ่งยากขึ้นมาก็คือ ต้องการทำ Refresh หน่วยความจำตลอดเวลา

- Program Memory

เป็นหน่วยความจำที่ใช้กับโปรแกรม เป็นหน่วยความจำที่ใช้อ่านข้อมูลได้อย่างเดียว (ROM) สัญญาณจาก MCS-51 ที่ใช้ในการอ่าน ROM คือ $\overline{PSEN}$

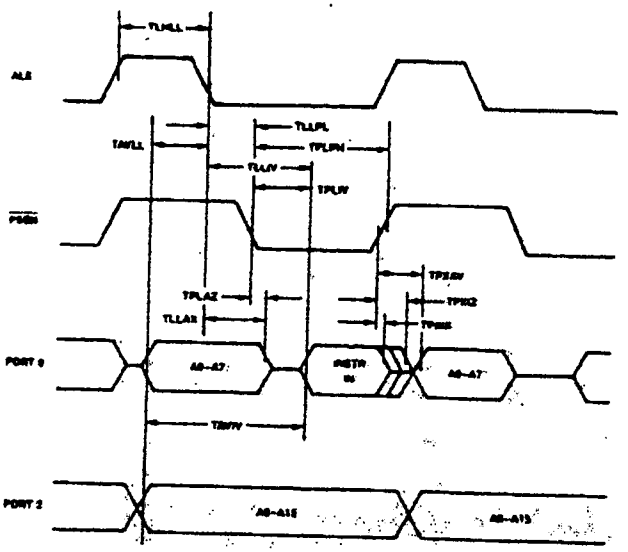
- ROM (Read Only Memory)

เป็นหน่วยความจำแบบถาวร เมื่อขาดไฟเลี้ยงข้อมูลไม่สูญหาย เป็นหน่วยความจำชนิดอ่านข้อมูลได้อย่างเดียว

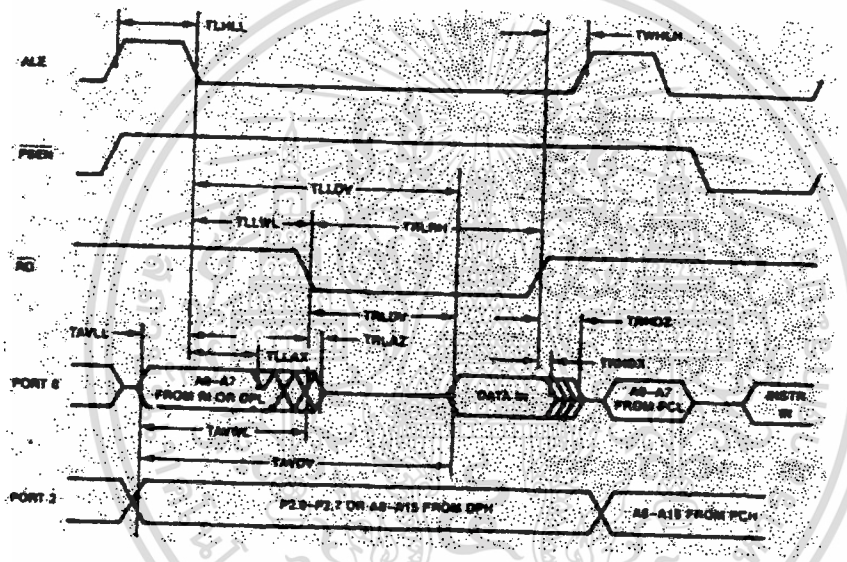
EPROM (Erasable Programmable Read Only Memory)

เป็นหน่วยความจำแบบถาวรชนิดหนึ่งซึ่งผู้ใช้นำมาโปรแกรมเองได้และสามารถใช้แสง UV ลบข้อมูลได้

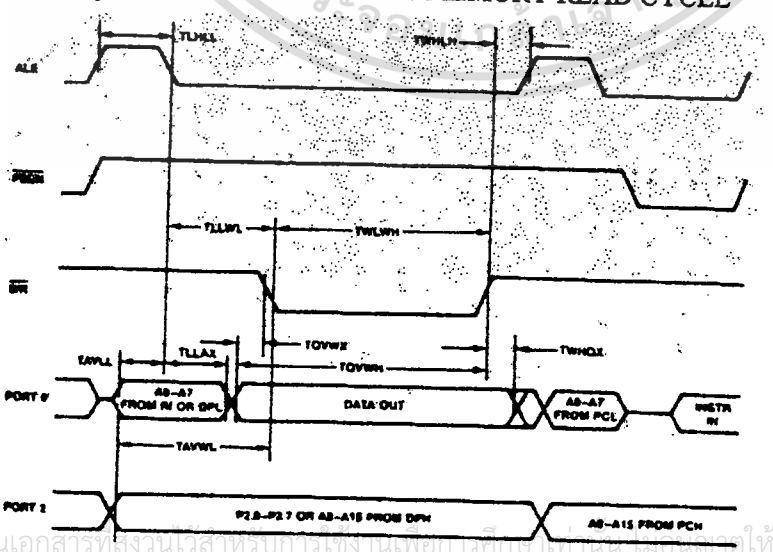
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูป 2.28 EXTERNAL PROGRAM MEMORY READ CYCLE



รูป 2.29 EXTERNAL DATA MEMORY READ CYCLE



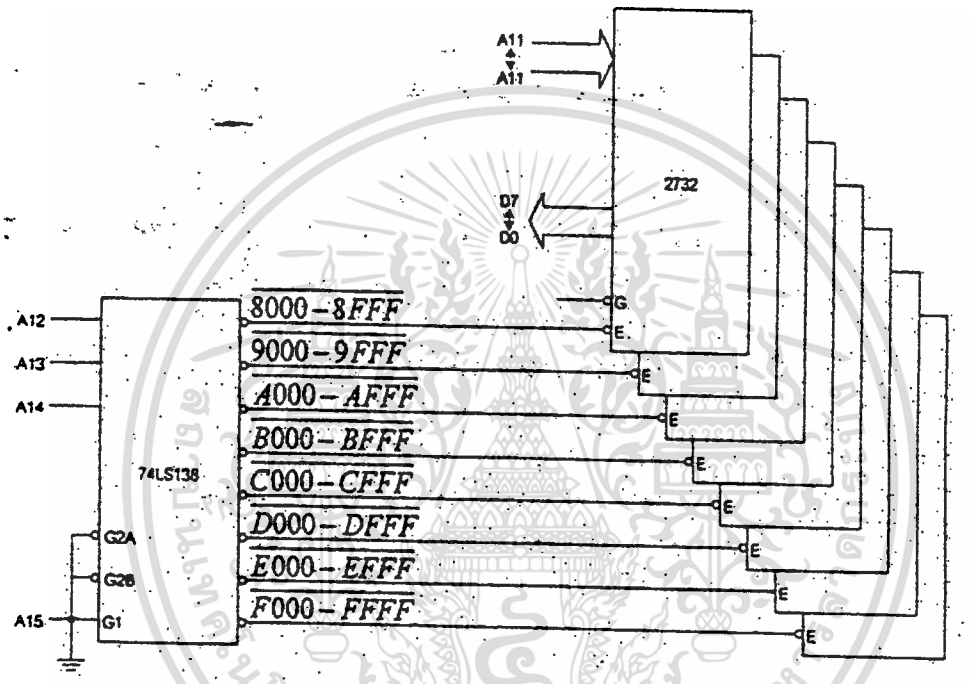
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูป 2.30 EXTERNAL DATA MEMORY WRITE CYCLE

การถอดรหัสตำแหน่งของหน่วยความจำ

การถอดรหัสของหน่วยความจำมีความจำเป็นอย่างยิ่งเพราะว่า หน่วยความจำที่ใช้มีขนาดเล็กกว่า 64 KB ดังนั้นจึงต้องนำมาเรียงกัน แล้วใช้วงจรถอดรหัสตำแหน่งตำแหน่งของหน่วยความจำแยกเป็นแต่ละช่วง ถ้าใช้หน่วยความจำขนาด 4 KB จำนวน 8 ตัว จะได้เท่ากับ 32 KB ถ้าให้ตัวแรกเริ่มที่ Address 8000H ตัวที่ 2 ก็จะเริ่ม Address 9000H

การถอดรหัสตำแหน่งของหน่วยความจำจะใช้ IC ถอดรหัส 3 to 8 line Decoder เบอร์ 74LS138 จะได้การต่อที่สมบูรณ์ดังแสดงในรูป 2.34



รูป 2.31 วงจรถอดรหัส

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3

จออักษร LCD

LCD เป็นชื่อย่อของ Liquid Crystal Display หรือที่เรียกกันโดยทั่วไปว่า ตัวแสดงผลแบบผลึกเหลว ซึ่งเป็นสารที่รวมตัวกันอยู่อย่างได้สัดส่วนระหว่างของเหลวกับผลึก จุดหลอมเหลวของสารชนิดนี้อยู่ในช่วงอุณหภูมิที่เรียกว่า เมโซเฟส (mesophase) ซึ่งโมเลกุลของสารสามารถเคลื่อนที่ได้เหมือนดังของเหลว แต่สารชนิดนี้ถูกจัดให้อยู่ในประเภทเดียวกันกับผลึกชนิดที่เป็นของแข็งทั่วๆไป

ในปี พ.ศ. 2513 ได้มีการค้นพบว่า แผ่นสารผลึกเหลวสามารถเปลี่ยนตัวเองจากใสกลายเป็นทึบแสง หรือจากทึบแสงกลายเป็นใสได้โดยการป้อนแรงดันเข้าไป คุณสมบัตินี้ก็คือหลักการพื้นฐานของ LCD ในปัจจุบันนั่นเอง

ข้อดีโดยทั่วไปของ LCD คือ

- บางเบา และพกพาสะดวก มีความหนาเพียงไม่กี่มิลลิเมตร
- ใช้พลังงานน้อยเนื่องจากต้องการพลังงานและแรงดันต่ำทำให้สามารถใช้งานได้นาน แม้จะมีเพียงแบตเตอรี่ขนาดเล็กเป็นตัวจ่ายกำลัง ทั้งยังร่วมใช้กับอุปกรณ์อิเลคทรอนิกส์ เช่น CMOS ได้
- เป็นตัวแสดงผลแบบพาสซีฟ เพราะว่า LCD ไม่ได้กำเนิดแสง การอ่านค่าที่ตัวแสดงผลต้องใช้แสงสว่างจากภายนอก แต่ความเข้มของการแสดงผลก็ไม่ได้จางลงเมื่อแสงสว่างจากภายนอกเพิ่มขึ้น หากต้องการอ่านค่าในที่มืด ทำได้โดยใช้แสงไฟส่องมาจากด้านหลังของแผง LCD
- เชื่อถือได้ ใช้งานได้ในช่วงอุณหภูมิที่กว้าง และมีอายุการใช้งานนาน
- ราคาถูก และใช้งานได้อย่างกว้างขวาง

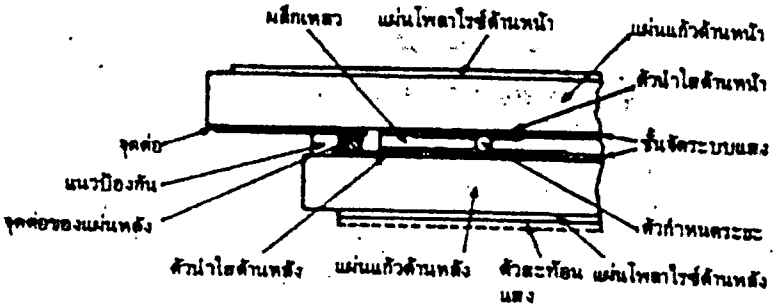
โดยทั่วไปแล้วเรามักจะเห็น LCD เป็นส่วนประกอบของนาฬิกา เครื่องคิดเลข และเครื่องมือวัด ในสมัยแรกๆ ตัวแสดงผลจะมี 7 ส่วน ที่มีจำนวนตัวแสดงผลไม่กี่หลักเท่านั้น การพัฒนาต่อมาทำให้ LCD ก้าวเข้าไปมีบทบาทในวิทยุเคลื่อนที่ คอมพิวเตอร์ และอุปกรณ์อิเลคทรอนิกส์เพื่อความบันเทิงต่างๆ ปัจจุบัน LCD ได้เข้ามาแทนที่ CRT (Cathode Ray Tube) ในการแสดงผลข้อความ และกราฟฟิก ตลอดจนนำเข้ามาแทนที่จอภาพโทรทัศน์ในที่สุด

โครงสร้างของ LCD

ประกอบด้วยแผ่นแก้ว 2 แผ่นประกบกัน โดยเว้นช่องกลางไว้ 6 ถึง 10 ไมโครเมตร (รูปที่ 3.1) ผิวด้านในของแผ่นแก้วเคลือบด้วยตัวนำไฟฟ้าชนิดใสที่ไว้แสดงตัวอักษร สัญลักษณ์หรือเครื่องหมายต่างๆ มักทำมาจากสาร อินเดียมทินออกไซด์ (Indium / Tin Oxide : ITO) ระหว่างตัวนำไฟฟ้าชนิดใสกับผลึกเหลวจะมีชั้นสารที่ทำให้โมเลกุลของผลึกรวมตัวกันในทิศทางของแสงที่ส่อง

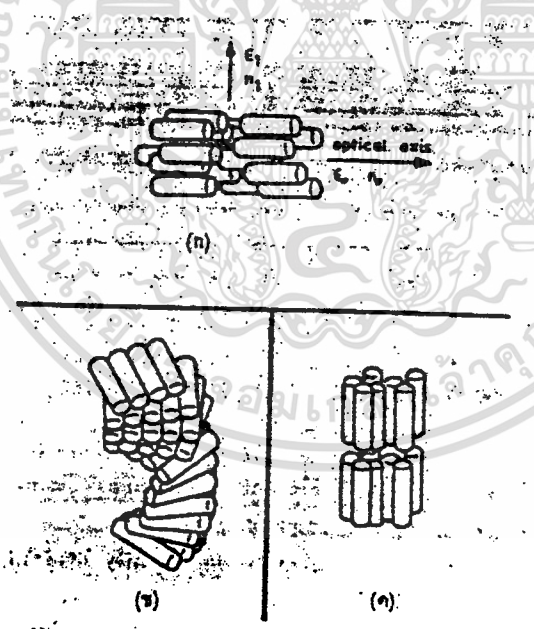
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

มา ชั้นสารนี้จึงเป็นที่รู้จักกันในนามของชั้นที่หันเหเข้าหาแสงหรือชั้นจัดระบบรับแสง (Alignment Layer)



รูปที่ 3.1 โครงสร้างของ LCD

ระยะห่างระหว่างแผ่นทั้งสองถูกจำกัดโดยตัวจัดระยะ (สังเกตดูรูปจะเห็นเป็นวงกลมอยู่ระหว่างชั้น) ชนิดของผลึกเหลวแบบที่ใช้โดยทั่วไปคือแบบนีเมติก (nematic) แสดงดังรูปที่ 3.2ก. โมเลกุลของผลึกเหลวแบบนีเมติกจะวางขนานเป็นแนวตรงคล้ายเส้นลวดยาว ถ้าหากวางกลับทิศจะทำให้คุณสมบัติของมันเปลี่ยนไป คริสตอลเหลวที่สามารถแสดงเครื่องหมายพิเศษต่างๆ ได้คือแบบโคเรสเทอริก (croresteric) รูปที่ 3.2ข. และแบบสเมกติก (smetic) รูปที่ 3.2ค.



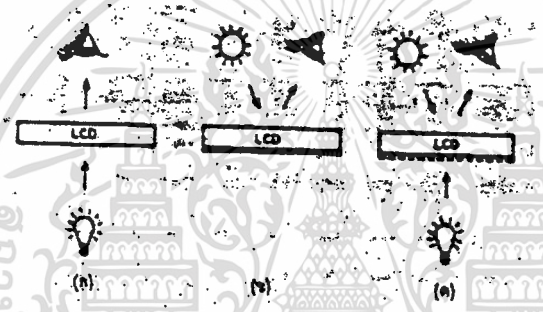
รูปที่ 3.2 แสดงโครงสร้างของโมเลกุลของผลึกเหลวทั้ง 3 แบบ

แบบต่างๆ ของการแสดงผล

LCD สามารถแสดงผลให้เราเห็นได้โดยมีหลักการ 3 แบบ จะเลือกใช้แบบใดก็ขึ้นอยู่กับแสงสว่างโดยรอบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. แบบสะท้อน (Reflective Mode) จะมีสารประเภทโลหะเคลือบอยู่ที่แผ่นหลังของ LCD เช่น เคลือบด้วยอลูมิเนียมฟอล์ยจะทำการสะท้อนแสงจากภายนอกผ่านตัวแสดงผลไปยังตาของเรา (รูปที่ 3.3ข.) แบบนี้จะเหมาะสำหรับที่ที่มีแสงสว่างเพียงพอ ข้อดีคือ ไม่ต้องการแหล่งจ่ายแรงดันป้อนให้กับหลอดไฟใดๆ อีก
2. แบบส่งผ่าน (Transmissive Mode) มักใช้กับ LCD ที่มีการแสดงผลเชิงลบ (ซึ่งสภาวะการทำงานของตัวแสดงผลจะโปร่งใสในขณะที่พื้นเป็นสีทึบ) แบบนี้จะวางหลอดไฟไว้ด้านหลังทำให้อ่านค่าแสดงผลได้ชัดเจน แสดงดังรูปที่ 3.3ก.
3. แบบส่งผ่าน / สะท้อน (Transflective Mode) เป็นการรวมระหว่าง 2 แบบที่กล่าวมาแล้ว ตัวแสดงผลอ่านได้จากการสะท้อนของแสงจากภายนอก และมีแสงสว่างส่องจากด้านหลังเมื่อต้องการอ่านค่าต่างๆ ในที่มืด แสดงดังรูปที่ 3.3ค.



รูปที่ 3.3 แบบต่างๆของการแสดงผลของ LCD

คุณสมบัติทางแสง

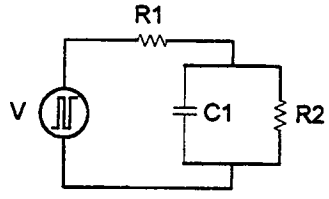
ความชัดเจนในการอ่านขึ้นอยู่กับตัวแปรสำคัญคือ ความสว่าง (Brightness) และความเข้มของแสง (Contrast) ของตัว LCD เอง ความเข้มของแสงหาได้จากความสว่างของบริเวณไร้สีหารด้วยความสว่างของบริเวณที่มีสีทึบ ในตัวแสดงผลแบบที่มีแสงส่องจากด้านหลังต้องมีอัตราส่วนความเข้มของแสงสูงเป็นพิเศษ เพราะเมื่อมองผ่านแสงที่สวนขึ้นมา ตาของเราจะจับความเข้มได้น้อยลง ทั้งความสว่างและความเข้มขึ้นอยู่กับชนิดของแผ่นโพลาไลซ์ สำหรับตัวแสดงผลแบบสะท้อนเชิงบวกแผ่นโพลาไลซ์ประสิทธิภาพต่ำจะให้ความสว่างมาก แต่ความเข้มน้อย ส่วนแผ่นโพลาไลซ์ประสิทธิภาพสูงจะให้ความเข้มสูงแต่ความสว่างลดลง

ความเร็วของการแสดงผล

เวลาตอบสนองการทำงานอยู่ในช่วง 50 ถึง 100 มิลลิวินาที (ในอุณหภูมิห้อง 25 °C) สิ่งสำคัญที่มีผลต่อช่วงเวลาตอบสนองคือ ความเหนียวของผลึกเหลว ซึ่งความเหนียว (Viscosity) ของมันจะเพิ่มขึ้นเมื่ออุณหภูมิลดลงเพราะโมเลกุลมีอิสระในการเคลื่อนที่น้อยลงทำให้การตอบสนองช้า

เอกสารนี้สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ลง เวลาตอบสนองยังขึ้นอยู่กับขนาดของแรงดันที่จ่ายให้ วิธีป้อนแรงดัน และความหนาของผลึกเหลวด้วย



รูปที่ 3.4 วงจรสมมูลย์ของ LCD

เมื่อ R1 เป็นความต้านทานอนุกรมที่เกิดจากตัวนำชนิดใส R2 และ C1 เป็นค่าที่เกิดจากผลึกเหลวที่อยู่ระหว่างตัวนำชนิดใส

LCD โมดูลชนิด Dot Metric

LCD โมดูลชนิด Dot Metric สามารถแบ่งออกได้เป็นพวกๆ คือ

- Character LCD Module
- Graphic LCD Module
- Segment Display Type LCD Module

โดยในแต่ละแบบนี้ก็จะมีส่วนประกอบใหญ่ๆ แบ่งได้เป็น

จอ LCD แบบ Dot Metric เป็นตัวแสดงผลให้เรามองเห็นในลักษณะการปิดและเปิดตัวเองกับแสงก็คือ ส่วนของที่เป็นตัวกระจกบรรจุผลึก

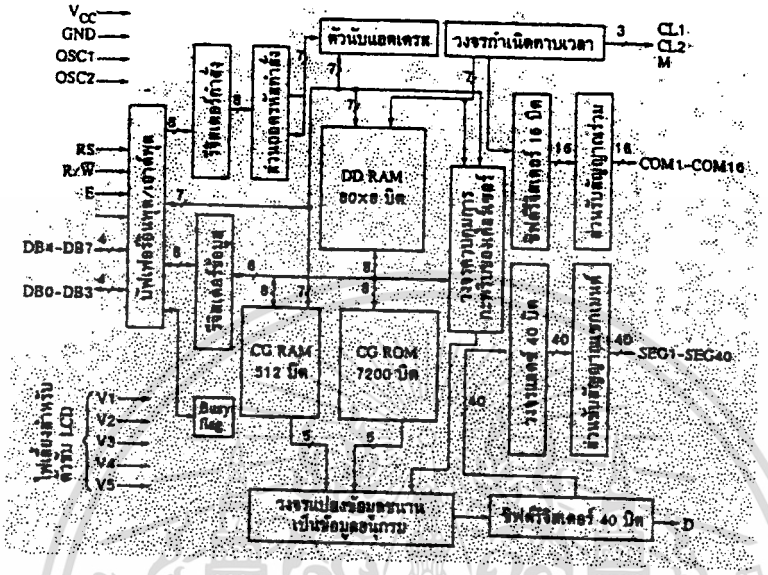
ไดรเวอร์ (Driver) เป็นตัวรับสัญญาณจากตัวควบคุมมาขับผลึก LCD อีกทีหนึ่งโดยมีเบอร์ที่นิยมใช้ใน LCD โมดูลเช่น HD4410H, MSM5259

คอนโทรลเลอร์ (Controller) เป็นตัวรับข้อมูลจากอุปกรณ์ภายนอกเข้ามา และจัดการควบคุม LCD โมดูลให้ทำงานแสดงผลต่างๆ เช่น การลบจอภาพ, การเกิดตัวอักษร เป็นต้น โดยมีเบอร์ไอซีที่นิยมใช้กันคือ HD44780 ซึ่งใช้ในแบบ Character LCD Module คือการแสดงเป็นตัวอักษรโดดๆ เป็นส่วนใหญ่ ส่วนเบอร์ HD61830 จะใช้ในแบบ Graphic LCD Module หรือแสดงในลักษณะกราฟฟิกได้

ในการศึกษาการทำงานและการใช้งาน LCD โมดูลนั้นไม่ใช่เรื่องยากเลยถ้าเราสามารถทำความเข้าใจในส่วนของคอนโทรลเลอร์ได้ก็เพียงพอแล้ว และโดยมาก LCD โมดูลในแต่ละบริษัทมักใช้ตัวคอนโทรลเลอร์ที่มีหลักการทำงานเหมือนๆ กันเป็นส่วนใหญ่ ใน LCD โมดูลแต่ละขนาดจำนวนตัวอักษรหรือจำนวนบรรทัดก็มีหลักการทำงานแบบเดียวกันทั้งหมด ไอซีที่นิยมมากที่สุดตัว

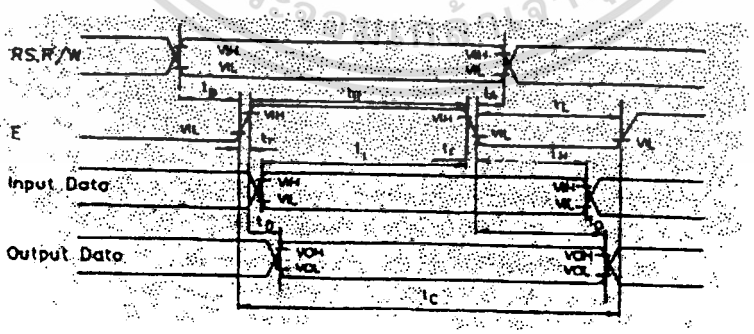
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หนึ่งที่เป็น LCD คอนโทรลเลอร์ก็คือ เบอร์ HD44780 โดยรูปแบบการทำงานของมันได้เป็นมาตรฐานให้กับ LCD คอนโทรลเลอร์ตัวอื่นๆ ด้วย



รูปที่ 3.5 โครงสร้างวงจรภายในของ HD44780 ซึ่งเป็น LCD คอนโทรลเลอร์

HD44780 เป็นไอซี LSI ตัวหนึ่ง ใช้ควบคุม LCD โดยแสดงผลในรูปตัวอักษรหรือสัญลักษณ์ต่างๆ ตัวมันเองสามารถต่อใช้งานแบบ 4 บิตหรือ 8 บิตได้ โดยถ้าเราต่อแบบ 4 บิต จะต่อใช้ขาที่ DB7 – DB4 เท่านั้น โดยข้อมูลแรกที่ส่งนั้น HD44780 จะถือเป็นข้อมูล 4 บิตบน และข้อมูลที่ส่งต่อมานั้นคือข้อมูล 4 บิตล่าง



รูปที่ 3.6 ลักษณะสัญญาณการควบคุมการทำงานของคอนโทรลเลอร์

โครงสร้างภายในของตัวควบคุมโมดูล LCD

ในการใช้งาน LCD จำเป็นต้องทำความเข้าใจเกี่ยวกับโครงสร้างและคำสั่งที่ใช้ในการควบคุมให้ดีเสียก่อน จากรูปโครงสร้างภายในของ HD 44780 ประกอบด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า แสวงหาผลประโยชน์ หรือนำไปเผยแพร่ในรูปแบบอื่นๆ ไม่ว่ากรณีใดๆ ทั้งสิ้น โดยไม่ได้รับอนุญาตจากเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บัฟเฟอร์อินพุทเอาท์พุท เป็นส่วนที่ใช้ในการติดต่อรับส่งข้อมูลกับอุปกรณ์ภายนอก เพื่อที่จะถ่ายทอดข้อมูลเข้าออกภายในตัวควบคุม

รีจีสเตอร์คำสั่ง (Instruction Registor : IR) เป็นรีจีสเตอร์ที่ใช้รับข้อมูลคำสั่งจากอุปกรณ์ภายนอก เพื่อนำไปทำการควบคุมการแสดงผล

รีจีสเตอร์ข้อมูล (Data Register : DR) เป็นรีจีสเตอร์ที่ใช้รับข้อมูลจากอุปกรณ์ภายนอก เพื่อถ่ายทอดต่อไปยัง RAM เก็บข้อมูลแสดงผล หรือนำข้อมูลนั้นไปเพื่อสร้างตัวอักษรเพิ่มเติมใน RAM เก็บตัวอักษร

แรมเก็บข้อมูลแสดงผล (Display Data Ram : DDRAM) เป็นหน่วยความจำแรมที่สามารถเขียนข้อมูลได้โดยผ่านทางรีจีสเตอร์ DR โดยตัวควบคุมจะนำข้อมูลใน DDRAM นี้ไปเปิดตาราง (Look up-table) ของตัวอักษรที่เก็บไว้ใน ROM และ RAM เก็บตัวอักษร เพื่อนำไปแสดงที่ตัวแสดงผล

รอมเก็บตัวอักษร (Character Generator ROM : CGROM) เป็นหน่วยความจำ ROM ที่ใช้เก็บข้อมูลตัวอักษรหรือสัญลักษณ์ที่สามารถดึงไปแสดงผลที่ตัวแสดงผลได้ มีขนาด 7200 บิต โดยจะถูกชี้โดยค่าของข้อมูลใน DDRAM

แรมเก็บตัวอักษร (Character Generator RAM : CGRAM) เป็นหน่วยความจำ RAM ที่ใช้เก็บตัวอักษรที่มีการสร้างขึ้นเพิ่มเติมขึ้นใหม่ ในกรณีที่ตัวอักษรใน CGROM ไม่เพียงพอ มีขนาด 512 บิต การดึงค่าไปใช้นั้นทำได้เช่นเดียวกับ CGROM คือเขียนข้อมูลลงใน DDRAM แล้วตัวควบคุมจะมาอ่านค่าจาก CGRAM เอง

แฟลก BUSY เป็นส่วนที่ทำหน้าที่แจ้งสถานะการทำงานของตัวควบคุมให้อุปกรณ์ภายนอกทราบว่า ตัวควบคุมพร้อมที่จะรับข้อมูลคำสั่งหรือไม่ ดังนั้นก่อนการส่งข้อมูลหรือคำสั่งมายังตัวควบคุม ต้องตรวจสอบสภาวะของแฟลก BUSY นี้เสียก่อน

ขาสัญญาณของโมดูล LCD แบบอักขระ

โมดูล LCD แบบอักขระที่นำมาเป็นตัวอย่างนี้เป็นแบบ 1 บรรทัด 16 ตัวอักษร มีขาต่อใช้งานทั้งสิ้น 14 ขาได้แก่

VSS (ขา 1) : ต่อกราวด์

VDD (ขา 2) : ต่อไฟเลี้ยง +5V

VO (ขา 3) : เป็นขาอินพุทสำหรับป้อนแรงดัน เพื่อปรับความเข้มของการแสดงผล

RS (ขา 4) : เป็นขาอินพุทใช้เลือกว่า ข้อมูลที่ทำการส่งในขณะนั้นเป็นข้อมูลคำสั่งของรีจีสเตอร์ IR หรือเป็นข้อมูลสำหรับรีจีสเตอร์ DR โดยถ้าขานี้เป็น 0 ข้อมูลที่ส่งมาจะเป็นข้อมูลคำสั่ง แต่ถ้าเป็น 1 ข้อมูลที่ส่งมาจะเป็นข้อมูลที่ใช้ในการแสดงผล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

R/W (ขา 5) : เป็นขาที่ใช้เลือกว่าจะอ่านหรือเขียนข้อมูลกับ LCD ถ้าหากเป็น 0 จะเป็นการเขียนข้อมูล แต่ถ้าเป็น 1 จะเป็นการอ่านข้อมูล

E (ขา 6) : เป็นขาอีนาเบิ้ล LCD ให้ทำงาน

DB0-DB7 (ขา 7-14) : เป็นขาที่ใช้เป็นทางผ่านของข้อมูลระหว่าง LCD กับอุปกรณ์ภายนอกขนาด 8 บิต

อนึ่งขา RS ,R/W และ E จะใช้งานร่วมกัน โดยมีลักษณะสัมพันธ์ดังในตารางที่ 3.1

ตารางที่ 3.1 ความสัมพันธ์ของสัญญาณ E ,RS และ R/W ของ LCD

Table 3 The relation between the operation and the combination of RS, R/W

RS	R/W	E	OPERATION
0	0		Write instruction code
0	1		Read busy flag and address counter
1	0		Write data
1	1		Read data

When performing data and instruction code by 4 bit, transfer RS, R/W every time.

เมื่อเราป้อนไฟให้ HD44780 นั้นก็จะทำการ Reset ตัวเองโดยจะใช้เวลาประมาณ 10 mS หลังจากไฟ VDD ถึง 4.5 V แล้ว โดยจะเซตตัวเองดังนี้

1. Display Clear จะทำการลบข้อมูล
2. Function Set โดยจะทำการเซตค่าภายใน

   DL = 1 เป็นการ Set ให้การติดต่อเป็นแบบ 8 บิต

   N = 0 เซตเป็น 1 บรรทัดการแสดงผล

   F = 0 ตัวหนังสือ 1 ตัวมีขนาด 5*7 จุด
3. Display ON/OFF

   D = Display OFF

   C = 0 Cursor OFF

   B = 0 Blink OFF
4. Entry Mode Set

   I/D = 1 (เพิ่มค่า Counter ขึ้น 1)

   S = 0 NO Shift

เมื่อเราเริ่มเปิดเครื่องทำงานแล้วก็จะต้องส่งคำสั่งควบคุมให้มันเริ่มทำงาน ซึ่งในการเขียนคำสั่งลงในตัวควบคุม จะต้องกำหนดให้ขา RS และ R/W เป็น 0 ทั้งคู่ แล้วเขียนข้อมูลคำสั่งตามไป คำสั่งควบคุม LCD ของชิพควบคุม HD44780 ที่สำคัญมีดังต่อไปนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. คำสั่งเคลียตัวแสดงผล (Clear Display)

RS	R/$\overline{W}$	DB7	--	--	--	--	--	--	DB0
0	0	0	0	0	0	0	0	0	1

มีข้อมูลคำสั่งเป็น $01 เป็นคำสั่งที่ใช้เขียนข้อมูลช่องว่าง หรือ Space เข้าไปใน DDRAM ทั้งหมด และจะทำให้ Address ของ DDRAM เป็น 0 และ Cursor จะกลับไปอยู่ตำแหน่งซ้ายสุดของจอแสดงผล และเซตบิต I/D ให้เป็น 1

2. คำสั่ง Return Home

RS	R/$\overline{W}$	DB7	--	--	--	--	--	--	DB0
0	0	0	0	0	0	0	0	1	x

ต้องกำหนดให้บิตที่ 1 ของข้อมูลเป็น 1 เป็นเป็นคำสั่งให้ Cursor เคลื่อนที่กลับไปยังตำแหน่งซ้ายสุดของจอแสดงผล แต่ข้อมูลบนจอแสดงผลไม่เปลี่ยนแปลง

3. คำสั่งเลือกโหมดการป้อนข้อมูล (Entry Mode Set)

RS	R/$\overline{W}$	DB7	--	--	--	--	--	--	DB0
0	0	0	0	0	0	0	1	I/D	S

บิต I/D เป็นบิตที่ใช้ในการกำหนดว่า เมื่อเขียนหรืออ่านข้อมูลแล้ว ทำให้แอดเดรสของ DDRAM เพิ่มขึ้น หรือลดลงหนึ่งแอดเดรส โดยถ้าบิตนี้เป็น 1 แอดเดรสของ DDRAM จะเพิ่มขึ้น แต่ถ้าเป็น 0 แอดเดรสจะลดลง

บิต S เป็นบิตที่ใช้ในการกำหนดลักษณะของการแสดงผล เมื่อมีการป้อนข้อมูลถ้าหากบิต S เป็น 1 เมื่อเกิดข้อมูลใหม่บนจอแสดงผล ตัว Cursor จะอยู่กับที่ แต่ตัวอักษรข้อมูลตัวเดิมจะถูกดันไปทางซ้าย แต่ถ้าหากบิตนี้เป็น 0 เมื่อเกิดข้อมูลตัวใหม่ตัว Cursor จะเลื่อนไปทางขวามือ

4. คำสั่งควบคุมการแสดงผล

RS	R/$\overline{W}$	DB7	--	--	--	--	--	--	DB0
0	0	0	0	0	0	1	D	C	B

บิต D ใช้ควบคุมการเปิดปิดจอแสดงผล ถ้าบิตนี้เป็น 1 จะเป็นการเปิดจอแสดงผล ถ้าเป็น 0 จะเป็นการปิดจอแสดงผล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บิต C ใช้ควบคุมการแสดงตัว Cursor บนจอแสดงผล ถ้าต้องการให้มี Cursor แสดงผลบนจอแสดงผล กำหนดให้บิตนี้เป็น 1 โดยตัวของ Cursor จะอยู่ที่ Line ที่ 8 ถ้าเป็นการแสดงผลแบบ 5*7 จุด และอยู่ที่ Line ที่ 11 ถ้าเป็รการแสดงผลแบบ 5*10 จุด

บิต B ใช้ควบคุมการกระพริบของ Cursor ถ้าบิตนี้เป็น 1 เคอร์เซอร์จะกระพริบ โดยมีอัตราการกระพริบประมาณ 379.2 mS

5. คำสั่งควบคุมการเลื่อน Cursor และลบข้อมูลตัวอักษร (Cursor or Display Shift)

RS	R/W	DB7	--	--	--	--	--	--	DB0
0	0	0	0	0	1	S/C	R/L	X	X

การควบคุมการเลื่อน Cursor และตัวอักษรบนตัวแสดงผล ขึ้นอยู่กับการกำหนดบิต S/C และ R/L ซึ่งสามารถสรุปได้ดังนี้

บิต S/C	บิต R/L	ลักษณะการเลื่อน
0	0	เลื่อน Cursor ไปทางซ้าย 1 ตำแหน่ง
0	1	เลื่อน Cursor ไปทางขวา 1 ตำแหน่ง
1	0	เลื่อนตัวอักษรที่เกิดขึ้นใหม่ไปทางซ้าย
1	1	เลื่อนตัวอักษรที่เกิดขึ้นใหม่ไปทางขวา

6. คำสั่งของฟังก์ชั่น (Function Set)

RS	R/W	DB7	--	--	--	--	--	--	DB0
0	0	0	0	1	DL	N	F	X	X

บิต DL ใช้กำหนดจำนวนบิตที่ใช้ติดต่อส่งผ่านข้อมูล ถ้าบิตนี้เป็น 0 จะเป็นการติดต่อแบบ 4 บิต แต่ถ้าเป็น 1 จะเป็นแบบ 8 บิต

บิต N ใช้กำหนดบรรทัดของการแสดงผล ถ้าเป็น 0 จะแสดงผล 1 บรรทัด ถ้าเป็น 1 จะแสดงเป็น 2 บรรทัด ในกรณีที่จอแสดงผลสามารถแสดงผลได้มากกว่า 2 บรรทัด และต้องการให้แสดงผลมากกว่า 2 บรรทัดก็กำหนดบิต N นี้ให้เป็น 1

บิต F ใช้เลือกความละเอียดของตัวอักษรในการแสดงผล ถ้าบิตนี้เป็น 0 จะเป็นการแสดงผลแบบ 5*7 จุด และถ้าเป็น 1 จะเป็นการแสดงผลแบบ 5*10 จุด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

7. คำสั่งเลือก Address ของ CGRAM (Set CGRAM Address)

RS	R/$\overline{W}$	DB7	--	--	--	--	--	--	DB0
0	0	0	1	A	A	A	A	A	A

เมื่อต้องการกำหนด Address ของ CGRAM ต้องกำหนดให้บิต 7 เป็น 0 บิต 6 เป็น 1 ส่วนอีก 6 บิตที่เหลือจะแทนด้วยค่าของ Address ของ CGRAM จะต้องทำการกำหนด Address ด้วยคำสั่งนี้ก่อนที่จะอ่าน หรือเขียนข้อมูลลงใน CGRAM โดย Address ของ CGRAM อยู่ระหว่าง $00-$3F

8. คำสั่งเลือก Address ของ DDRAM (Set DDRAM Address)

RS	R/$\overline{W}$	DB7	--	--	--	--	--	--	DB0
0	0	1	A	A	A	A	A	A	A

ใช้ในการเลือก Address ของ DDRAM ก่อนที่จะทำการอ่านหรือเขียนข้อมูลโดยบิต 7 ต้องเป็น 1 และข้อมูลอีก 7 บิตที่เหลือ จะเป็นค่า Address ของ DDRAM ซึ่ง Address ของ DDRAM จะอยู่ระหว่าง $00-$7F ทั้งนี้จำนวน Address ยังขึ้นกับการกำหนดสถานะที่บิต N ด้วย หากบิต N เป็น 0 Address ของ DDRAM จะอยู่ระหว่าง $00-$4F และถ้าบิต N เป็น 1 Address ของ DDRAM จะมี 2 ช่วงคือ $00-$27 สำหรับบรรทัดที่ 1 และ $40-$67 สำหรับบรรทัดที่ 2

9. คำสั่งอ่านแฟลก Busy และ Address (Read Busy Flag and Address)

RS	R/$\overline{W}$	DB7	--	--	--	--	--	--	DB0
0	1	BF	A	A	A	A	A	A	A

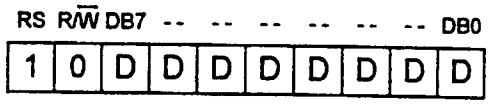
เป็นคำสั่งที่ใช้อ่านแฟลก Busy (BF) โดยแฟลกนี้จะเป็นตัวบอกสถานะของตัวควบคุม LCD ว่าพร้อมจะรับข้อมูลหรือไม่ ถ้าหากบิต BF เป็น 0 แสดงว่าตัวควบคุม LCD พร้อมรับข้อมูลหรือคำสั่ง แต่ถ้าเป็น 1 แสดงว่าขณะนี้ตัวควบคุม LCD ยังอยู่ในกระบวนการทำงานภายในหรือกำลังประมวลผลข้อมูลอยู่ ยังไม่พร้อมรับข้อมูลหรือคำสั่ง

เมื่อต้องการอ่านแฟลกต้องกำหนดให้ขา R/W เป็น 1 ด้วย แต่สัญญาณที่ RS ยังต้องเป็น 0 อยู่เพราะข้อมูลนี้เป็นข้อมูลคำสั่ง

นอกจากนี้ยังใช้เป็นคำสั่งอ่านข้อมูล Address ของ CGRAM และ DDRAM ด้วย โดยบิต 0 ถึงบิต 6 เป็นค่าข้อมูลของ Address ที่ต้องการอ่าน

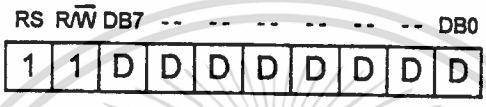
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

10. เมื่อต้องการเขียนข้อมูลลงใน CGRAM หรือ DDRAM



เริ่มแรกต้องกำหนด Address ที่ต้องการเขียนเสียก่อน จากนั้นกำหนดให้สัญญาณที่ขา RS เป็น 1 เพื่อบอกให้ตัวควบคุม LCD ทราบว่าข้อมูลที่ปรากฏต่อไปนี้เป็นข้อมูลปกติไม่ใช่ข้อมูลคำสั่งจากนั้นกำหนดให้ขา R/W เป็น 0 แล้วเขียนข้อมูล 8 บิตที่ต้องการตามไป

11. เมื่อต้องการอ่านข้อมูลจาก CGRAM หรือ DDRAM



เริ่มแรกต้องกำหนด Address ที่ต้องการอ่านเสียก่อน จากนั้นกำหนดให้สัญญาณที่ขา RS เป็น 1 เพื่อบอกให้ตัวควบคุม LCD ทราบว่าข้อมูลที่ปรากฏต่อไปนี้เป็นข้อมูลปกติไม่ใช่ข้อมูลคำสั่งจากนั้นกำหนดให้ขา R/W เป็น 1 ข้อมูล 8 บิตที่ปรากฏออกมาจะเป็นข้อมูลจาก Address CGRAM หรือ DDRAM ที่ต้องการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4
วงจรที่ใช้ในการทดลอง

โครงงานนี้ประกอบด้วย Hardware 4 ส่วน ได้แก่

1. Display Board
2. Control Board
3. DC-Motor Control Board
4. Power Supply Board

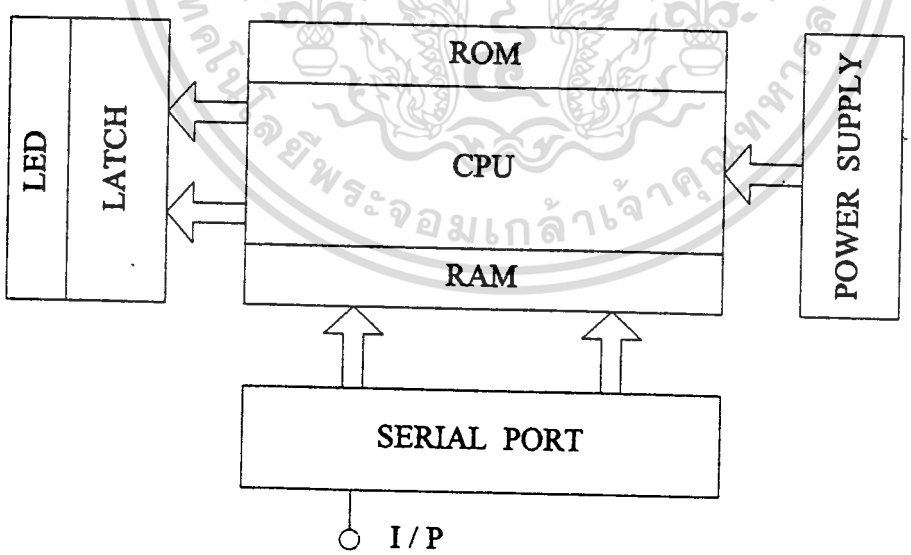
Display Board (2 ชุด)

หน้าที่ :

- ทำหน้าที่แสดงผลข้อมูลส่งออกทางหลอด LED 8 ดวง
- ทำหน้าที่รับข้อมูลที่ส่งมาทางพอร์ทอนุกรม

ส่วนประกอบหลัก :

- 89C51 WITH INTERNAL EPROM 4K
- RAM 8K (6264)
- DS – 275 SERIAL PORT
- LED 8 ดวง



รูปที่ 4.1 แผนผังการทำงานของ Display Board

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

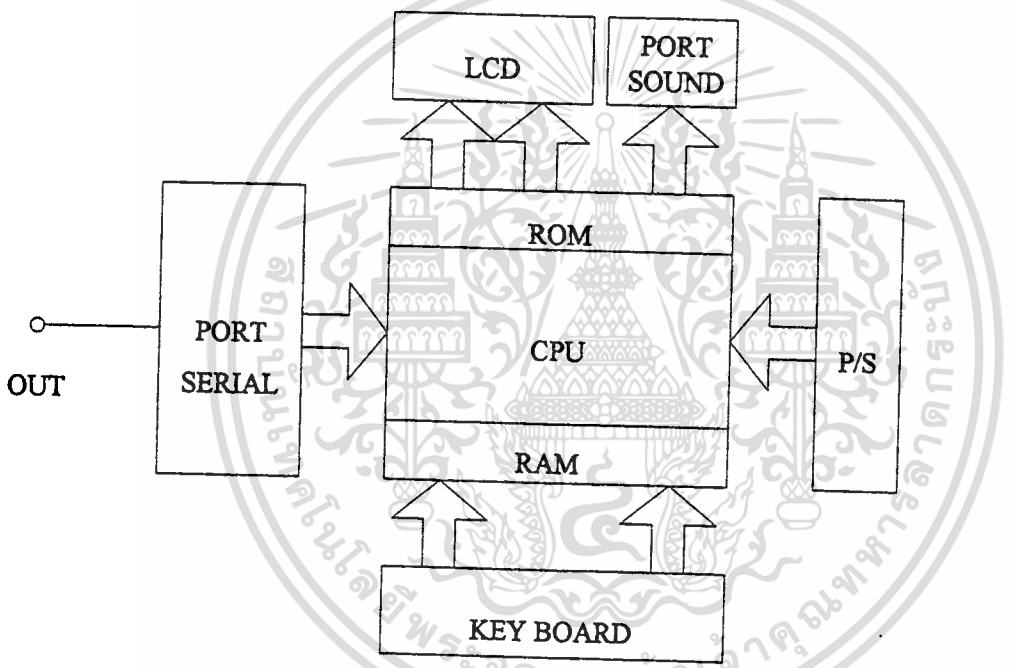
Control Board

หน้าที่ :

- ส่งข้อมูลไปให้ Display Board ทางพอร์ทอนุกรม โดยรับ Data จาก Keyboard
- นำข้อมูลจากการกด Keyboard ออกทางพอร์ทอนุกรม

ส่วนประกอบหลัก :

- 89C51 MICROCONTROLLER
- LCD MODULE
- KEYBOARD
- LM 386



รูปที่ 4.2 แผนผังการทำงานของ Control Board

DC – Motor Control Board

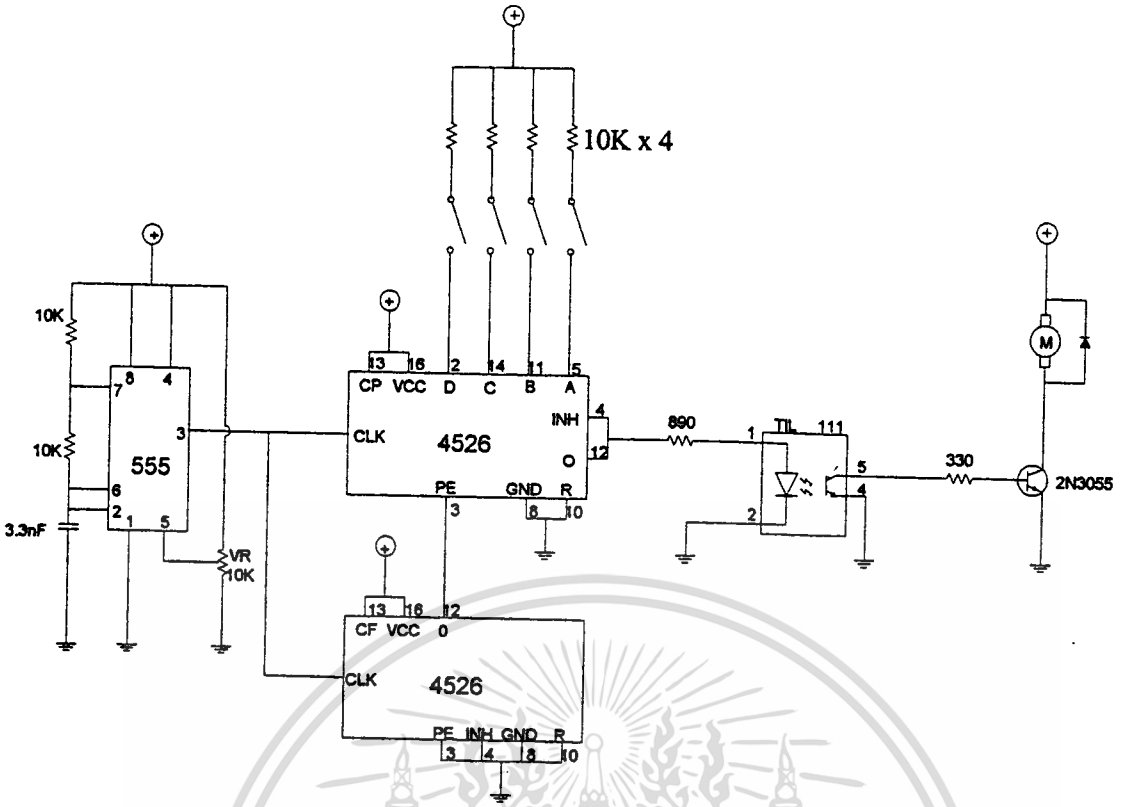
หน้าที่ :

- ทำหน้าที่ควบคุมความเร็วมอเตอร์

ส่วนประกอบหลัก :

- IC TIMER 555
- IC 4526

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.3 วงจรควบคุมมอเตอร์

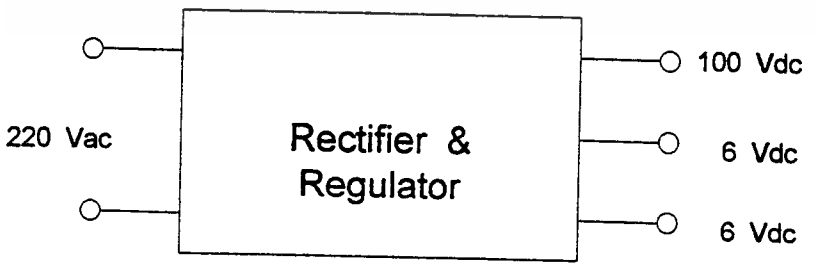
Power Supply Board

หน้าที่ :

- ทำหน้าที่จ่ายไฟให้กับ Display Board และ Control Board

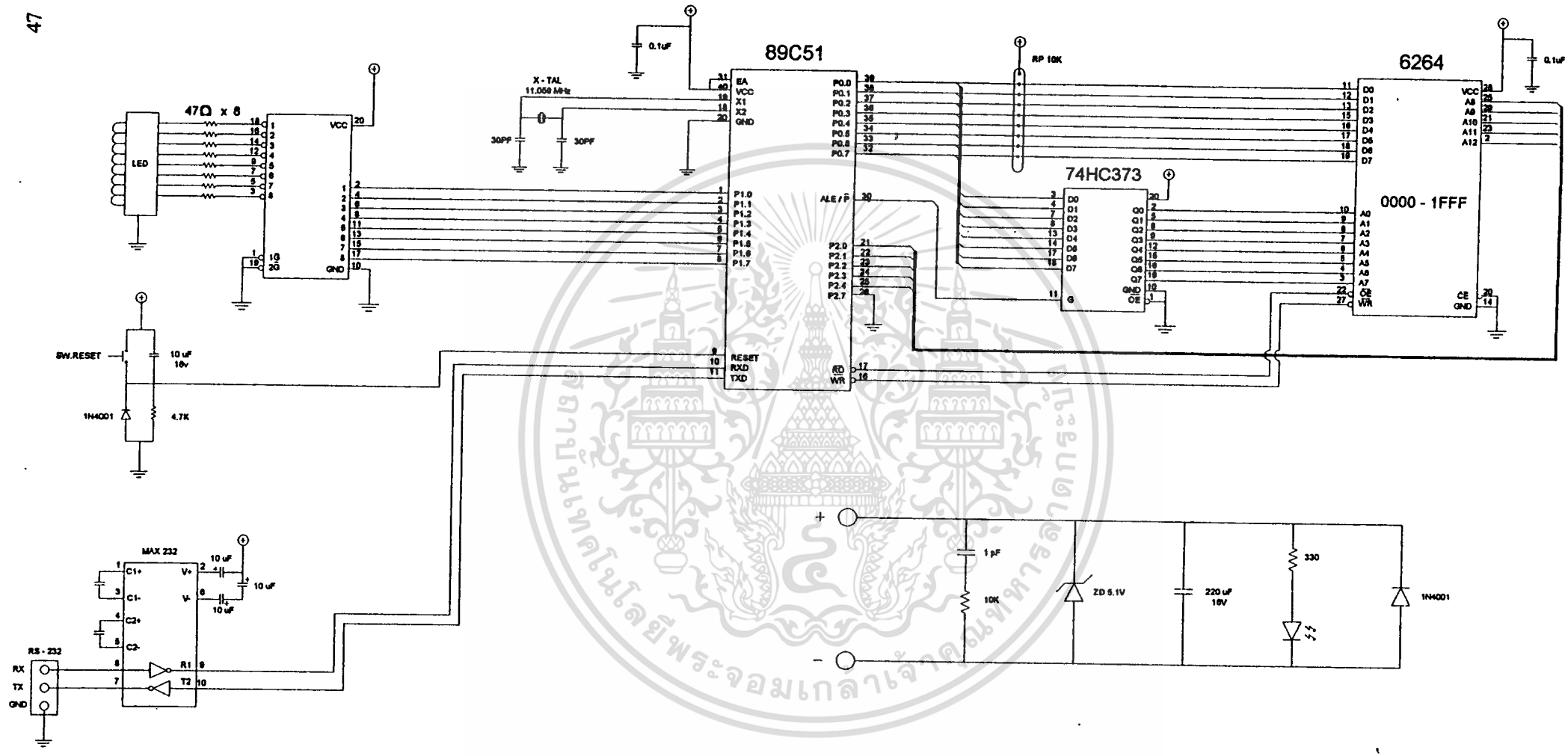
ส่วนประกอบหลัก :

- AUTO TRANSFORMER
- IC REGULATOR 7806

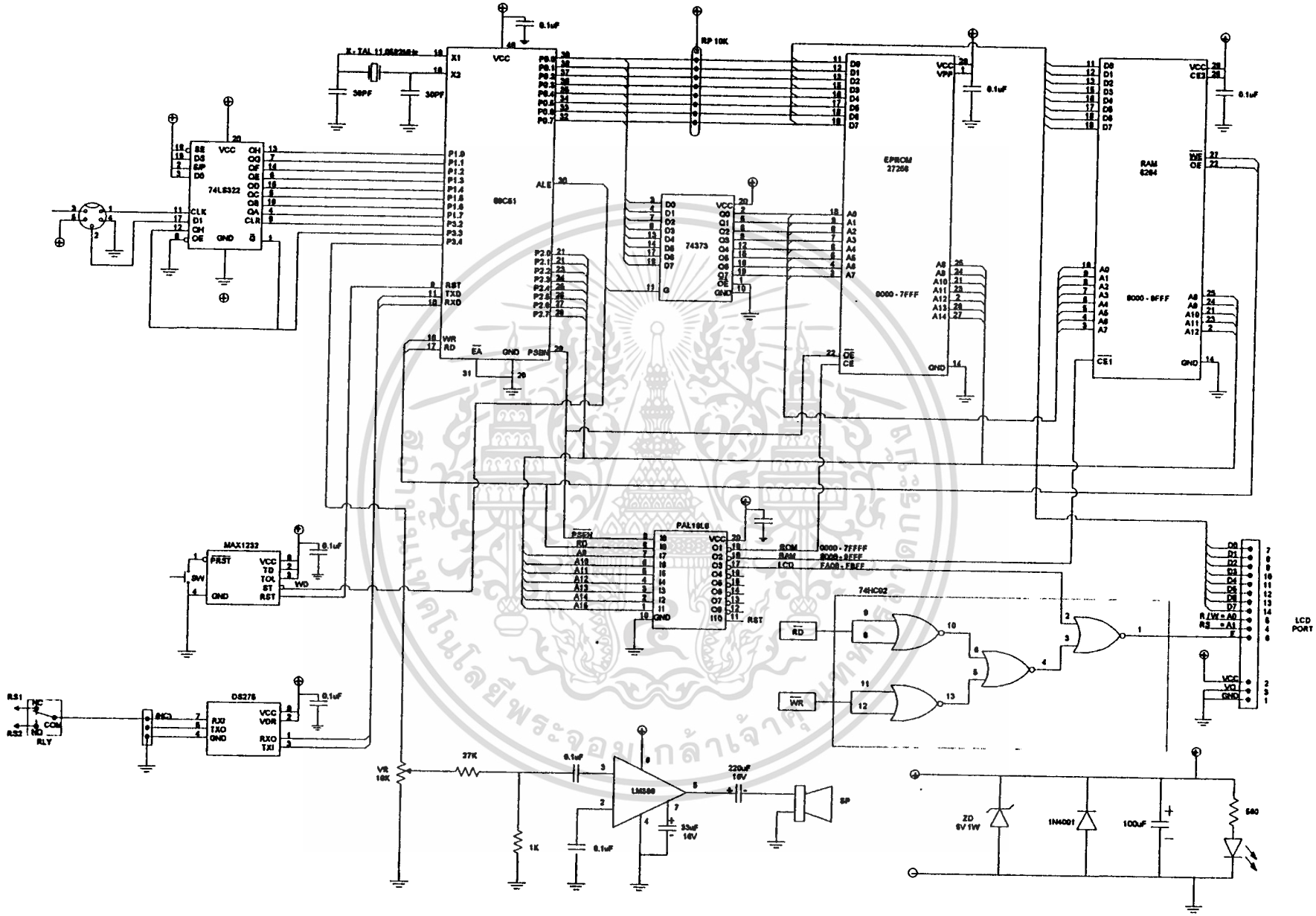


รูปที่ 4.4 แผนผังการทำงานของ Power Supply Board

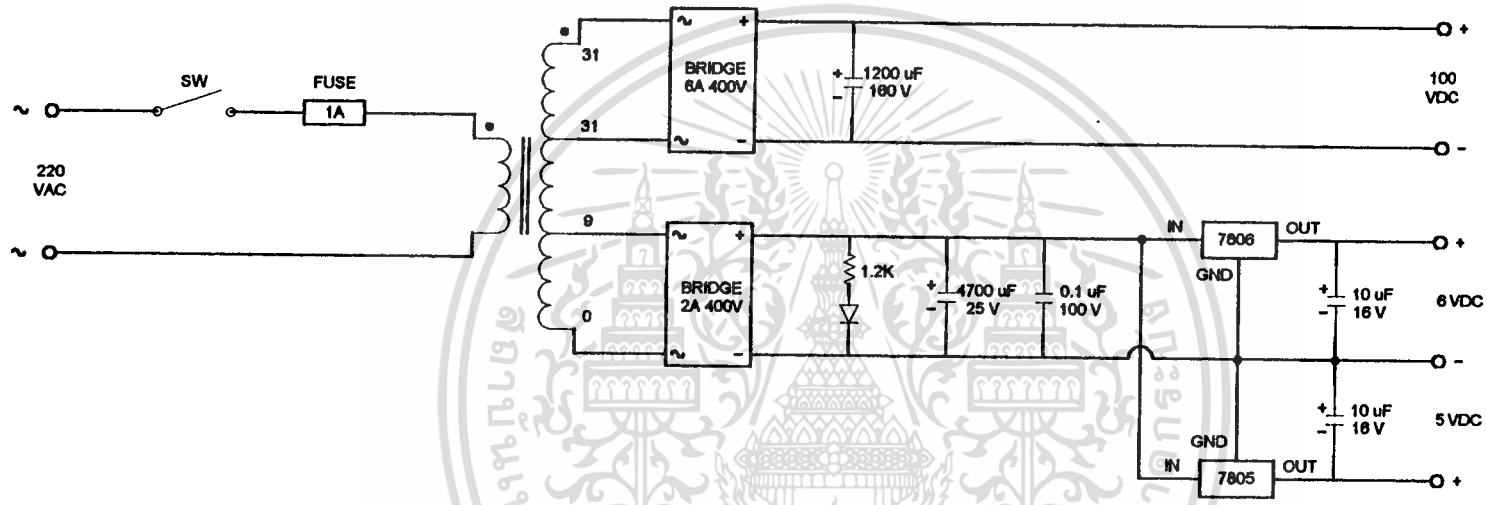
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



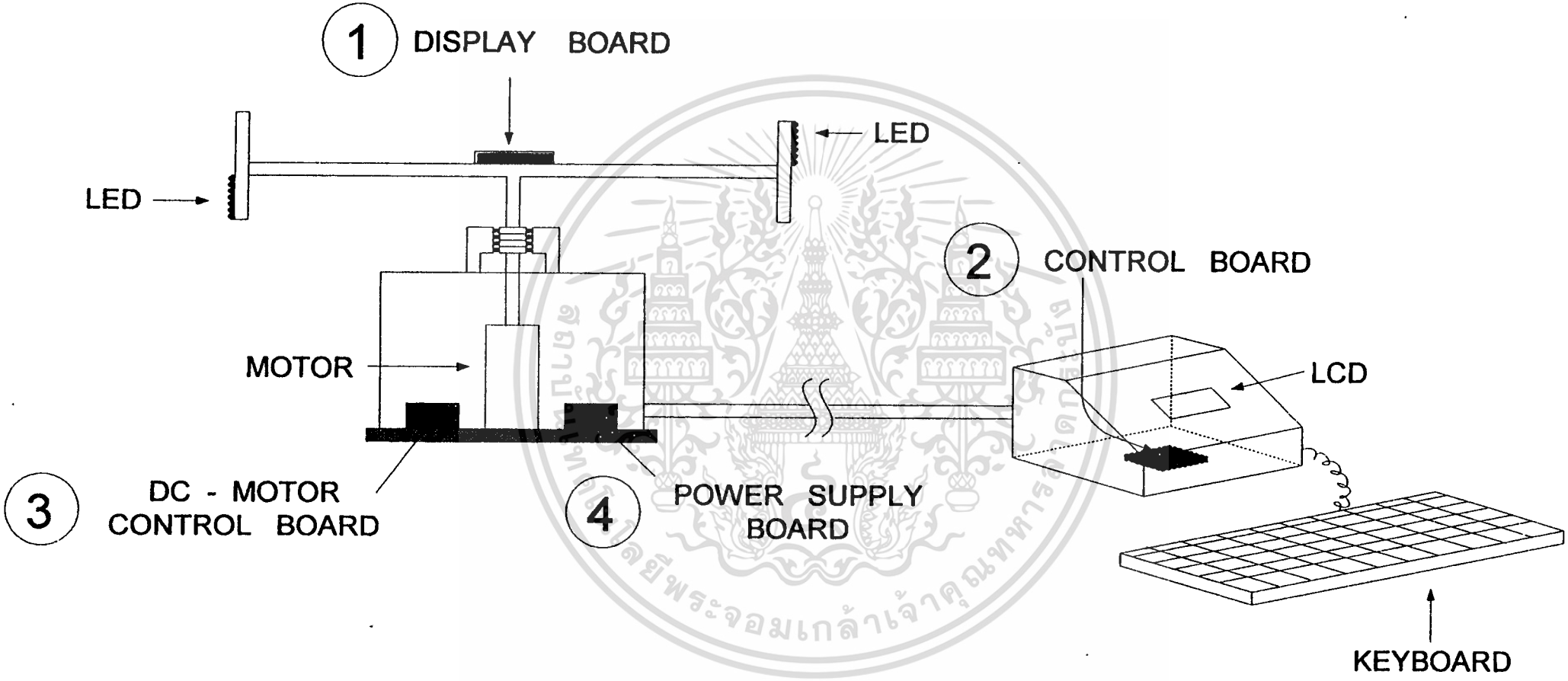
รูปที่ 4.5 วงจรของ Display Board



รูปที่ 4.6 วงจรของ Control Board



รูปที่ 4.7 วงจรของ Power Supply Board



รูปที่ 4.8 แสดงส่วนประกอบของโครงงาน

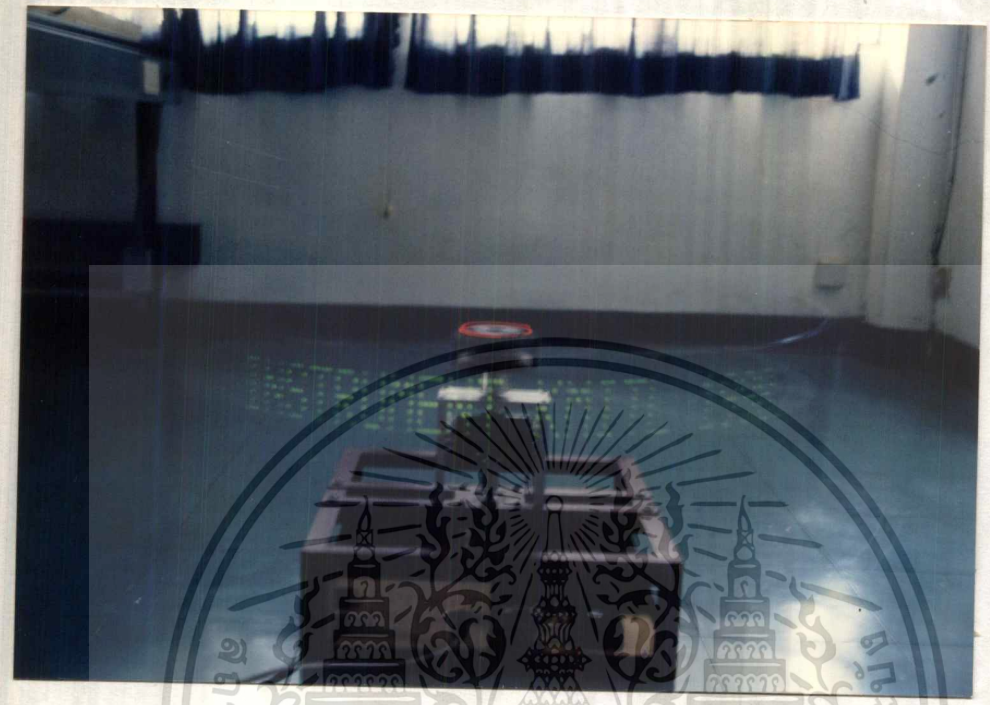
บทที่ 5

การทดลอง และ ผลการทดลอง

รูปที่ 5.1 แสดงส่วนประกอบทั้งหมดของโครงงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 5.2 แสดงผลข้อความที่ต้องการส่งออกบนจอ LCD



รูปที่ 5.3 แสดงผลการทำงานของโครงงานในการแสดงข้อความที่ต้องการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 6

บทวิจารณ์และสรุป

จากการทดลองป้อนข้อความที่ต้องการให้แสดงผลโดยผ่านทางคีย์บอร์ดคอมพิวเตอร์ พบว่า ตัวอักษรที่แสดงผลออกมาของระบบแสดงผลรอบทิศทางนั้นมีการกระพริบเกิดขึ้นเป็นช่วงๆ ทั้งนี้มีสาเหตุอันเนื่องมาจากความเร็วในการหมุนของตัวมอเตอร์น้อยเกินไป แต่ถ้าเราออกแบบวงจรควบคุมความเร็วมอเตอร์เสียใหม่ เพื่อให้มีความเร็วรอบในการหมุนเพิ่มมากขึ้น ก็จะมีผลต่อเนื่องไปถึงระยะเวลาที่ใช้ในการกำหนดการติดดับของหลอด LED ซึ่งจะต้องลดน้อยลงไปด้วยเพราะว่ามอเตอร์หมุนเร็วมากขึ้น อาจมีผลทำให้ความสว่างของหลอด LED ในการแสดงผลนั้นลดน้อยลง แต่ตัวอักษรที่แสดงผลออกมาของระบบแสดงผลรอบทิศทางนั้นจะกระพริบลดน้อยลงได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรมควบคุมการแสดงผล

```
;PROGRAM     : DISPLAY1.ASM
;DATE        : 1 JULY 1998  1:00PM
;ASSEMBLER   : T51 BY SONGCHAI WEERATHAWEEMAS  KMITL
;SOFTWARE    : BY YOT & JEE
;HARDWARE    : 89C51 WITH EEPROM 4K
;DESCRIPTION : SHOW 8 LED IN VERTICAL LINE FOR 300 COLUMN
;*************************************************************************************
;          INITIAL SUBROUTINE
;initial register of mcs-51
;*************************************************************************************
A           EQU     0E0H        ;ACC
TH1         EQU     8DH
R0          EQU     00H
R1          EQU     01H
R2          EQU     02H
R7          EQU     07H
;*************************************************************************************
            ORG     0000H
            SJMP    START       ;go to start program
            ORG     0023H
            LCALL   INT_SER     ;go to interrupt serial subroutine
START:      MOV     R0,#250     ;delay for 10 millisecond
DLY:        MOV     R1,#40      ;wait external device
            DJNZ    R1,$
            MOV     P1,#0FFH
            DJNZ    R0,DLY
            MOV     IE,#10010000B   ;set enable interrupt EA=1,ES=1
            MOV     IP,#00010000B   ;first priority serial interrupt PS=1
            MOV     PCON,#00H
            MOV     TMOD,#22H
            MOV     TL1,#0FDH       ;9600 bps reload
            MOV     TH1,#0FDH
            SETB    TCON.6          ;TR1=1 RUN_TIMER1
            MOV     SCON,#050H      ;MODE1 REN=1 TI=1 RI=0
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                SETB        P1.7            ;off led 8
MAIN:           AJMP        DISPLAY

;*****************************************************************************************
;           DISPLAY SUBROUTINE
;input: count,dptr
;*****************************************************************************************
DISPLAY:        MOV         DPTR,#0000H
                MOV         R0,#20
DIS1:           MOV         R1,#15
DIS:            MOVX        A,@DPTR
                LCALL       ON_LED
                LCALL       OFF_LED
                INC         DPTR
                DJNZ        R1,DIS
                DJNZ        R0,DIS1
                AJMP        DISPLAY

;*****************************************************************************************
;           ON LED SUBROUTINE
;input: acc
;*****************************************************************************************
ON_LED:         CPL         A
                SETB        P1.7
                MOV         P1,A            ;out led  1 m/c
                MOV         R2,#90          ;delay 94-4 microsecond 1 m/c
                DJNZ        R2,$            ;2 m/c
                RET
;*****************************************************************************************
;           OFF LED SUBROUTINE
;input: R7=TIC 10
;*****************************************************************************************
OFF_LED:        MOV         P1,#0FFH        ;off led  2 m/c
                MOV         R4,#115         ;125-10 1 m/c
                DJNZ        R4,$            ;2 m/c
                RET
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
;*****************************************************************************************
;           SERIAL INTERRUPT
;interrupt serial subroutine for receive data save in ram
;*****************************************************************************************
INT_SER:    CLR     IE.4
            MOV     P1,#0FFH
            MOV     DPTR,#0000H
            MOV     R0,#15
INN1:       MOV     R1,#20
INN:        JNB     SCON.0,$
            CLR     SCON.0
            MOV     A,SBUF
            MOVX    @DPTR,A
            INC     DPTR
            DJNZ    R1,INN
            DJNZ    R0,INN1
            SETB    IE.4
            RETI
            AJMP    MAIN

;*********************************************************************
            END
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรมควบคุม LCD ,ส่งข้อมูลออนุกรม ,ควบคุมคีย์บอร์ด

```
;*****************************************************************************************
;PROGRAM      : CONTROL.ASM
;DATE         : 15 JULY 1998  12:00 AM
;ASSEMBLER    : T51
;SOFTWARE     : BY  YOT & JEE
;HARDWARE     : 80C31
;DESCRIPTION  : CONTROL LCD AND SEND DATA TO DISPLAY , INPUT KEY BOARD
;*****************************************************************************************
;            INITIAL SUBRUTINE
;*****************************************************************************************
A           EQU     0E0H
TH1         EQU     08DH
R0          EQU     00H
R1          EQU     01H
R2          EQU     02H
RAM_BUFF    EQU     08000H      ;BUFFER DATA 40 CHAR
SEND_BUFF   EQU     09000H      ;SEND BUFF FOR LED 300
LCD_W_I     EQU     0FA00H      ;WRITE IR
LCD_R_I     EQU     0FA01H      ;READ IR
LCD_W_D     EQU     0FA02H      ;WRITE DATA
LCD_R_D     EQU     0FA03H      ;READ DATA
;*************************************************
            ORG     0000H
            AJMP    START
            ORG     0013H
            AJMP    INT_EX1
START:      MOV     R0,#250     ;delay 10 mSec
DLY:        MOV     R1,#40      ;wait external device
            DJNZ    R1,$
            DJNZ    R0,DLY
            MOV     IE,#10000000B   ;EA=1,EX1=0
            MOV     IP,#00000100B   ;first priority PX1=1
            MOV     PCON,#00H       ;SMOD=0
            MOV     TMOD,#22H       ;timer1 mode 2 8-bit auto-reload
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                MOV     TH1,#0FDH       ;for 9600 bps
                MOV     TL1,#0FDH       ;for 9600 bps
                SETB    TCON.6          ;set TR1=1 run timer1
                MOV     SCON,#52H       ;MODE1 REN=1 TI=1 RI=0
                MOV     P1,#0FFH        ;write modified read
                MOV     P3,#0FFH
                SETB    P3.3
                CLR     P3.2            ;clear data keyboard
                MOV     R3,#20
                DJNZ    R3,$
                SETB    P3.2            ;wait key press
MAIN:           LCALL   FUNC_SET
                LCALL   DIS_CON
BACK:           LCALL   CLR_LCD
                LCALL   STEP_1
MAIN1:          LCALL   TYPE_TEXT
                LCALL   STEP_2
                LCALL   CONFIRM
                LCALL   STEP_3
                LCALL   ENCODE
                LCALL   SEND
                AJMP    BACK

;*****************************************************************************
;       FUNCTION SETUP SUBRUTINE
;*****************************************************************************
FUNC_SET:       MOV     DPTR,#LCD_W_I   ;write IR
                MOV     A,#00111000B    ;set 8_bit,2 line,5*7
                MOVX    @DPTR,A
                LCALL   READY           ;delay and check BF
                RET

;*****************************************************************************
;       DISPLAY CONTROL SUBRUTINE
;*****************************************************************************
DIS_CON:        MOV     DPTR,#LCD_W_I
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                MOV     A,#00001111B    ;set on display,on cursor,cursor blink
                MOVX    @DPTR,A
                LCALL   READY
                RET

;*****************************************************************************************
;       CLEAR LCD SUBRUTINE
;*****************************************************************************************
CLR_LCD:        MOV     DPTR,#LCD_W_I
                MOV     A,#00000001B    ;clear LCD
                MOVX    @DPTR,A
                LCALL   READY
                RET

;*****************************************************************************************
;       SHIFT SUBRUTINE
;*****************************************************************************************
SHIFTR:         MOV     DPTR,#LCD_W_I
                MOV     A,#00011100B
                MOVX    @DPTR,A
                LCALL   BUSY
                LCALL   LATE
                RET
SHIFTL:         MOV     DPTR,#LCD_W_I
                MOV     A,#00011000B
                MOVX    @DPTR,A
                LCALL   BUSY
                LCALL   LATE
                RET
LATE:           PUSH    R1
                PUSH    R2
                SETB    P3.3
                CLR     P3.2
                SETB    P3.2
                MOV     R1,#00H
LATE2:          MOV     R2,#00H
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
LATE1:      JB      P3.3,AJP
            DJNZ    R2,$
            DJNZ    R1,LATE2
            POP     R2
            POP     R1
            RET
AJP:        POP     R2
            POP     R1
            LCALL   SOUND
            AJMP    MAIN1

;*****************************************************************************
;       ENTRY MODE SET SUBRUTINE
;*****************************************************************************
ENTRY_MOD:  MOV     DPTR,#LCD_W_I
            MOV     A,#00000110B    ;auto_inc_address,shift cursor right
            MOVX    @DPTR,A
            LCALL   READY
            RET

;*****************************************************************************
;       READY SUBRUTINE
;*****************************************************************************
READY:      PUSH    DPH
            PUSH    DPL
            MOV     DPTR,#LCD_R_I   ;read IR
WAIT:       MOVX    A,@DPTR         ;read BF
            MOV     R0,#20          ;delay 20 uS
            DJNZ    R0,$
            JB      A.7,WAIT        ;Check busy flag
            POP     DPL
            POP     DPH
            RET
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
;*****************************************************************************
;       STEP 1 SUBRUTINE
;*****************************************************************************
STEP_1:     MOV     DPTR,#LCD_W_I
            MOV     A,#00001100B    ;on display,off cursor
            MOVX    @DPTR,A
            LCALL   BUSY
            MOV     DPTR,#LCD_W_I
            MOV     A,#00010100B    ;cursor shift right
            MOVX    @DPTR,A
            LCALL   BUSY
            MOV     R3,#00H
            MOV     R1,#80H         ;DD-RAM LINE 1  R1=ADDRESS
            LCALL   LOOP
            MOV     R1,#0C0H        ;DD-RAM LINE 2
            LCALL   LOOP
            SETB    P3.3
            CLR     P3.2
            MOV     R1,#20
            DJNZ    R1,$
            SETB    P3.2
SHIF:       LCALL   SHIFTR
            LCALL   SHIFTR
            LCALL   SHIFTR
            LCALL   SHIFTR
            LCALL   SHIFTR
            LCALL   SHIFTR
            LCALL   SHIFTR
            LCALL   SHIFTR
            LCALL   SHIFTL
            LCALL   SHIFTL
            LCALL   SHIFTL
            LCALL   SHIFTL
            LCALL   SHIFTL
            LCALL   SHIFTL
            LCALL   SHIFTL
            LCALL   SHIFTL
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        LCALL   SHIFTL
        JNB     P3.3,SHIF
        LCALL   SOUND
        AJMP    MAIN1
LOOP:   MOV     R4,#20
LOOP1:  MOV     DPTR,#LCD_W_I
        MOV     A,R1
        MOVX    @DPTR,A         ;define address to DD-RAM
        LCALL   BUSY
        MOV     DPTR,#LCD_W_D
        LCALL   TEXT1
        MOV     A,R2            ;get data text1
        MOVX    @DPTR,A
        LCALL   BUSY
        INC     R1
        INC     R3
        DJNZ    R4,LOOP1
        RET
TEXT1:  PUSH    A
        PUSH    DPH
        PUSH    DPL
        MOV     DPTR,#D_TEXT1
        MOV     A,R3
        MOVC    A,@A+DPTR
        MOV     R2,A
        POP     DPL
        POP     DPH
        POP     A
        RET
BUSY:   PUSH    A
        PUSH    DPH
        PUSH    DPL
        MOV     DPTR,#LCD_R_I
BUSY1:  MOVX    A,@DPTR         ;read BF
        JB      A.7,BUSY1       ;check busy flag
        POP     DPL
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                POP     DPH
                POP     A
                RET

;*****************************************************************************
;       TYPE TEXT SUBRUTINE
;*****************************************************************************
TYPE_TEXT:      LCALL   CLR_LCD
                LCALL   DIS_CON
                MOV     R3,#80H
                MOV     R2,#20
                LCALL   ADR
LN1:            LCALL   KEYBOARD
                LCALL   SOUND
                CJNE    R5,#01H,DX1     ;ENTER
                RET
DX1:            CJNE    R5,#02H,WX1     ;DELETE
                LCALL   ERASE
WX1:            LCALL   WRITE1
                LCALL   S_RIGHT
                DJNZ    R2,LN1
                MOV     R3,#0C0H
                MOV     R2,#20
                LCALL   ADR
LN2:            LCALL   KEYBOARD
                LCALL   SOUND
                CJNE    R5,#01H,DX2     ;ENTER
                RET
DX2:            CJNE    R5,#02H,WX2     ;DELETE
                LCALL   ERASE
WX2:            LCALL   WRITE1
                LCALL   S_RIGHT
                DJNZ    R2,LN2
AGN:            LCALL   KEYBOARD
                LCALL   SOUND
                CJNE    R5,#01H,AGN
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                RET
S_RIGHT:        MOV     DPTR,#LCD_W_I
                MOV     A,#00000110B    ;SHIFT RIGHT, AUTO-INC ADDRESS
                MOVX    @DPTR,A
                LCALL   BUSY
                RET
WRITE1:         MOV     DPTR,#LCD_W_D
                MOV     A,R5
                MOVX    @DPTR,A
                LCALL   BUSY
                RET
ADR:            MOV     DPTR,#LCD_W_I
                MOV     A,R3
                MOVX    @DPTR,A
                LCALL   BUSY
                RET
ERASE:          INC     R2
                INC     R2
                MOV     R5,#20H
                LCALL   ERASE1
                RET
ERASE1:         MOV     DPTR,#LCD_W_I
                MOV     A,#00000100B    ;SHIFT LEFT
                MOVX    @DPTR,A
                LCALL   BUSY
                RET
LSH:            MOV     DPTR,#LCD_W_I
                MOV     A,#00010000B
                MOVX    @DPTR,A
                LCALL   BUSY
                RET
;*****************************************************************************************
;       KEYBOARD SETUP SUBRUTINE
;O/P=R5 SEND DATA FROM KEY PRESSED
;*****************************************************************************************
KEYBOARD:       PUSH    A
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        CLR     P3.2            ;clear data P3.2=1:CLEAR
        SETB    P3.3            ;for check keyboard press
        MOV     R7,#20          ;wait 20 uS
        DJNZ    R7,$
        SETB    P3.2            ;cancel reset
        JNB     P3.3,$          ;wait until key pressed=1, P3.3=1:PRESSED
        MOV     P1,#0FFH        ;write modify read set P1 to input port
        MOV     A,P1            ;READ DATA KEYBOARD
        LCALL   DECODE
        MOV     R5,A            ;R5=DATA FROM KEYBOARD
        POP     A
        RET

;*****************************************************************************
;       STEP 2 SUBRUTINE
;save data on ram
;*****************************************************************************
STEP_2: MOV     DPTR,#RAM_BUFF  ;RAM 8000H
        MOV     R0,#80H         ;line 1
        MOV     R1,#20
W1:     LCALL   L1              ;read data from dd-ram
        INC     R0
        MOVX    @DPTR,A         ;write data to ram buffer
        INC     DPTR
        DJNZ    R1,W1
        MOV     R0,#0C0H
        MOV     R1,#20
W2:     LCALL   L1
        INC     R0
        MOVX    @DPTR,A
        INC     DPTR
        DJNZ    R1,W2
        RET
L1:     PUSH    DPL
        PUSH    DPH
        MOV     DPTR,#LCD_W_I
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                MOV     A,R0
                MOVX    @DPTR,A
                LCALL   BUSY
                MOV     DPTR,#LCD_R_D
                MOVX    A,@DPTR         ;save in ACC
                LCALL   BUSY
                POP     DPH
                POP     DPL
                RET

;*****************************************************************************************
;       CONFIRM SUBRUTINE
;*****************************************************************************************
CONFIRM:        LCALL   CLR_LCD
                LCALL   DIS_CON
                PUSH    DPH
                PUSH    DPL
                MOV     R7,#00H
                MOV     DPTR,#D_TEXT3
                MOV     R3,#80H
                MOV     R1,#16
C1:             MOV     A,R7
                MOVC    A,@A+DPTR
                MOV     R5,A
                LCALL   WRITE
                INC     R3
                INC     R7
                DJNZ    R1,C1
UNDO:           LCALL   KEYBOARD
                LCALL   SOUND
                LCALL   WRITE
                CJNE    R5,#4EH,C3      ;NO
                POP     DPL
                POP     DPH
                AJMP    BACK
C3:             CJNE    R5,#59H,UNDO    ;YES
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        POP     DPL
        POP     DPH
        RET

;*****************************************************************************************
;       STEP 3  SUBRUTINE
;SHOW PLEASE WAITE
;*****************************************************************************************
STEP_3: LCALL   CLR_LCD
        LCALL   DIS_CON
        PUSH    DPH
        PUSH    DPL
        MOV     R7,#00H
        MOV     DPTR,#D_TEXT4
        MOV     R3,#80H
        MOV     R1,#20
S1:     MOV     A,R7
        MOVC    A,@A+DPTR
        MOV     R5,A
        LCALL   WRITE
        INC     R3
        INC     R7
        DJNZ    R1,S1
        POP     DPL
        POP     DPH
        RET

;*****************************************************************************************
;       ENCODE  SUBRUTINE
;*****************************************************************************************
ENCODE: PUSH    DPH
        PUSH    DPL
        PUSH    A
        MOV     R0,#00H         ;address counter00-300
        MOV     R1,#40          ;CHAR=40
        MOV     DPTR,#RAM_BUFF  ;LOAD data from RAM
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
E1:             MOVX        A,@DPTR
                LCALL       LED_DECODE
                INC         DPTR
                DJNZ        R1,E1
                MOV         DPTR,#SEND_BUFF;
                POP         A
                POP         DPL
                POP         DPH
                RET

;*****************************************************************************************
;       LED DECODE SUBRUTINE
;I/P= R0 counter address , A=data
;*****************************************************************************************
LED_DECODE:     PUSH        DPH
                PUSH        DPL
                CJNE        A,#20H,AAA      ;SPACE NULL
                MOV         DPTR,#D_NULL
                AJMP        REC
AAA:            CJNE        A,#41H,BBB      ;A
                MOV         DPTR,#D_A
                AJMP        REC
BBB:            CJNE        A,#42H,CCC
                MOV         DPTR,#D_B
                AJMP        REC
CCC:            CJNE        A,#43H,DDD
                MOV         DPTR,#D_C
                AJMP        REC
DDD:            CJNE        A,#44H,EEE
                MOV         DPTR,#D_D
                AJMP    REC
EEE:            CJNE        A,#45H,FFF
                MOV         DPTR,#D_E
                AJMP        REC
FFF:            CJNE        A,#46H,GGG
                MOV         DPTR,#D_F
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        AJMP    REC
GGG:    CJNE    A,#47H,HHH
        MOV     DPTR,#D_G
        AJMP    REC
HHH:    CJNE    A,#48H,III
        MOV     DPTR,#D_H
        AJMP    REC
III:    CJNE    A,#49H,JJJ
        MOV     DPTR,#D_I
        AJMP    REC
JJJ:    CJNE    A,#4AH,KKK
        MOV     DPTR,#D_J
        AJMP    REC
KKK:    CJNE    A,#4BH,LLL
        MOV     DPTR,#D_K
        AJMP    REC
LLL:    CJNE    A,#4CH,MMM
        MOV     DPTR,#D_L
        AJMP    REC
MMM:    CJNE    A,#4DH,NNN
        MOV     DPTR,#D_M
        AJMP    REC
NNN:    CJNE    A,#4EH,OOO
        MOV     DPTR,#D_N
        AJMP    REC
OOO:    CJNE    A,#4FH,PPP
        MOV     DPTR,#D_O
        AJMP    REC
PPP:    CJNE    A,#50H,QQQ
        MOV     DPTR,#D_P
        AJMP    REC
QQQ:    CJNE    A,#51H,RRR
        MOV     DPTR,#D_Q
        AJMP    REC
RRR:    CJNE    A,#52H,SSS
        MOV     DPTR,#D_R
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        AJMP    REC
SSS:    CJNE    A,#53H,TTT
        MOV     DPTR,#D_S
        AJMP    REC
TTT:    CJNE    A,#54H,UUU
        MOV     DPTR,#D_T
        AJMP    REC
UUU:    CJNE    A,#55H,VVV
        MOV     DPTR,#D_U
        AJMP    REC
VVV:    CJNE    A,#56H,WWW
        MOV     DPTR,#D_V
        AJMP    REC
WWW:    CJNE    A,#57H,XXX
        MOV     DPTR,#D_W
        AJMP    REC
XXX:    CJNE    A,#58H,YYY
        MOV     DPTR,#D_X
        AJMP    REC
YYY:    CJNE    A,#59H,ZZZ
        MOV     DPTR,#D_Y
        AJMP    REC
ZZZ:    CJNE    A,#5AH,D11
        MOV     DPTR,#D_Z
        AJMP    REC
D11:    CJNE    A,#31H,D22
        MOV     DPTR,#D_1
        AJMP    REC
D22:    CJNE    A,#32H,D33
        MOV     DPTR,#D_2
        AJMP    REC
D33:    CJNE    A,#33H,D44
        MOV     DPTR,#D_3
        AJMP    REC
D44:    CJNE    A,#34H,D55
        MOV     DPTR,#D_4
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        AJMP    REC
D55:    CJNE    A,#35H,D66
        MOV     DPTR,#D_5
        AJMP    REC
D66:    CJNE    A,#36H,D77
        MOV     DPTR,#D_6
        AJMP    REC
D77:    CJNE    A,#37H,D88
        MOV     DPTR,#D_7
        AJMP    REC
D88:    CJNE    A,#38H,D99
        MOV     DPTR,#D_8
        AJMP    REC
D99:    CJNE    A,#39H,D00
        MOV     DPTR,#D_9
        AJMP    REC
D00:    CJNE    A,#30H,PLUS1
        MOV     DPTR,#D_0
        AJMP    REC
PLUS1:  CJNE    A,#2BH,MINUS1
        MOV     DPTR,#D_PLUS
        AJMP    REC
MINUS1: CJNE    A,#2DH,MULTI1
        MOV     DPTR,#D_MINUS
        AJMP    REC
MULTI1: CJNE    A,#2AH,DOT1
        MOV     DPTR,#D_MULTI
        AJMP    REC
DOT1:   MOV     DPTR,#D_DOT
        AJMP    REC
REC:    MOV     R2,#06H
        MOV     R3,#00H
REC1:   MOV     A,R3
        MOVC    A,@A+DPTR
        MOV     R5,A
        PUSH    DPH
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                PUSH        DPL
                PUSH        A
                MOV         DPTR,#SEND_BUFF
                MOV         A,R0            ;COUNTER ADDRESS
                MOV         DPL,A
                MOV         A,R5            ;LOAD data
                MOVX        @DPTR,A
                INC         R0
                POP         A
                POP         DPL
                POP         DPH
                INC         R3
                DJNZ        R2,REC1
                POP         DPL
                POP         DPH
                RET

;*****************************************************************************
;       INTERRUPT EXTERNAL_1 SUBRUTINE
;*****************************************************************************
INT_EX1:        CLR         IE.2            ;EX1=1 non ex1-interrupt tempolary
                AJMP        MAIN1

;*****************************************************************************
;       SEND SUBRUTINE
;*****************************************************************************
SEND:           SETB        SCON.1          ;TI=1
                MOV         DPTR,#SEND_BUFF ;DATA 300 CHAR FROM RAM 8900H
                MOV         R6,#15
SS2:            MOV         R5,#20
SS1:            JNB         SCON.1,S        ;CHECK TI=1?
                CLR         SCON.1          ;clear TI=0
                MOVX        A,@DPTR
                MOV         SBUF,A          ;TRANSMIT
                INC         DPTR
                MOV         R7,#250         ;delay 250 uS
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                DJNZ        R7,$
                DJNZ        R5,SS1
                DJNZ        R6,SS2
                RET

;*****************************************************************************************
;       WRITE SUBRUTINE
;INPUT: R3=DD-address R5=data
;*****************************************************************************************
WRITE:          PUSH        A
                PUSH        DPH
                PUSH        DPL
                MOV         DPTR,#LCD_W_I
                MOV         A,R3
                MOVX        @DPTR,A
                LCALL       BUSY
                MOV         DPTR,#LCD_W_D
                MOV         A,R5
                MOVX        @DPTR,A
                LCALL       BUSY
                POP         DPL
                POP         DPH
                POP         A
                RET

;*****************************************************************************************
;       SOUND SUBRUTINE
;*****************************************************************************************
SOUND:          PUSH        R0
                PUSH        R1
                PUSH        R2
                MOV         R1,#70          ;PLAY TIME = 70 mS
SOUND1:         MOV         R2,#10
SOUND2:         SETB        P3.4            ;BIT SOUND= 10 KHz
                MOV         R0,#50
                DJNZ        R0,S
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        CLR     P3.4
        MOV     R0,#50
        DJNZ    R0,$
        DJNZ    R2,SOUND2
        DJNZ    R1,SOUND1
        POP     R2
        POP     R1
        POP     R0
        RET

;*****************************************************************************
;       DECODE SUBRUTINE
;*****************************************************************************
DECODE: CJNE    A,#0E1H,BB
        MOV     A,#41H
        RET
BB:     CJNE    A,#098H,CC
        MOV     A,#42H
        RET
CC:     CJNE    A,#084H,DD
        MOV     A,#43H
        RET
DD:     CJNE    A,#0C4H,EE
        MOV     A,#44H
        RET
EE:     CJNE    A,#093H,FF
        MOV     A,#45H
        RET
FF:     CJNE    A,#0D4H,GG
        MOV     A,#46H
        RET
GG:     CJNE    A,#0B1H,HH
        MOV     A,#47H
        RET
HH:     CJNE    A,#0CCH,II
        MOV     A,#48H
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        RET
II:     CJNE    A,#0C2H,JJ
        MOV     A,#49H
        RET
JJ:     CJNE    A,#0DCH,KK
        MOV     A,#4AH
        RET
KK:     CJNE    A,#085H,LL
        MOV     A,#4BH
        RET
LL:     CJNE    A,#0D2H,MM
        MOV     A,#4CH
        RET
MM:     CJNE    A,#0B9H,NN
        MOV     A,#4DH
        RET
NN:     CJNE    A,#08CH,OO
        MOV     A,#4EH
        RET
OO:     CJNE    A,#08BH,PP
        MOV     A,#4FH
        RET
PP:     CJNE    A,#0B2H,QQ
        MOV     A,#50H
        RET
QQ:     CJNE    A,#0A8H,RR
        MOV     A,#51H
        RET
RR:     CJNE    A,#0B4H,SS
        MOV     A,#52H
        RET
SS:     CJNE    A,#0D8H,TT
        MOV     A,#53H
        RET
TT:     CJNE    A,#0D1H,UU
        MOV     A,#54H
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        RET
UU:     CJNE    A,#0F3H,VV
        MOV     A,#55H
        RET
VV:     CJNE    A,#0A8H,WW
        MOV     A,#56H
        RET
WW:     CJNE    A,#0B8H,XX
        MOV     A,#57H
        RET
XX:     CJNE    A,#089H,YY
        MOV     A,#58H
        RET
YY:     CJNE    A,#0ACH,ZZ
        MOV     A,#59H
        RET
ZZ:     CJNE    A,#0B0H,_0
        MOV     A,#5AH
        RET
_0:     CJNE    A,#0A2H,_1
        MOV     A,#30H
        RET
_1:     CJNE    A,#0D0H,_2
        MOV     A,#31H
        RET
_2:     CJNE    A,#0F1H,_3
        MOV     A,#32H
        RET
_3:     CJNE    A,#0C8H,_4
        MOV     A,#33H
        RET
_4:     CJNE    A,#0A4H,_5
        MOV     A,#34H
        RET
_5:     CJNE    A,#0E9H,_6
        MOV     A,#35H
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        RET
_6:     CJNE    A,#0D9H,_7
        MOV     A,#36H
        RET
_7:     CJNE    A,#0BCH,_8
        MOV     A,#37H
        RET
_8:     CJNE    A,#0F8H,_9
        MOV     A,#38H
        RET
_9:     CJNE    A,#0C4H,ENTER
        MOV     A,#39H
        RET
ENTER:  CJNE    A,#0B5H,DEL
        MOV     A,#01H
        RET
DEL:    CJNE    A,#0CDH,SPACE
        MOV     A,#02H
        RET
SPACE:  CJNE    A,#094H,DOT
        MOV     A,#20H
        RET
DOT:    CJNE    A,#092H,PLUS
        MOV     A,#2EH
        RET
PLUS:   CJNE    A,#09EH,MINUS
        MOV     A,#2BH
        RET
MINUS:  CJNE    A,#0DEH,MULTI
        MOV     A,#2DH
        RET
MULTI:  CJNE    A,#0F9H,NON
        MOV     A,#2AH
        RET
NON:    CLR     P3.2            ;clear data
        SETB    P3.3            ;clear check key
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        MOV     R7,#20
        DJNZ    R7,$
        SETB    P3.2            ;cancel clear
        JNB     P3.3,$
        MOV     P1,#0FFH
        MOV     A,P1
        AJMP    DECODE

;*****************************************************************************************
;       DATA ZONE SUBRUTINE
;*****************************************************************************************
D_TEXT1:    DB 'PRESS ENTER     '
            DB ' TO EDIT        '

D_TEXT3:    DB 'CONFIRM? (Y/N):  '

D_TEXT4:    DB ' PLEASE WAIT...  '


D_NULL:     DB 00H,00H,00H,00H,00H,00H
D_A:        DB 00H,7EH,09H,09H,09H,7EH
D_B:        DB 00H,7FH,49H,49H,49H,36H
D_C:        DB 00H,3EH,41H,41H,41H,22H
D_D:        DB 00H,7FH,41H,41H,22H,1CH
D_E:        DB 00H,7FH,49H,49H,49H,41H
D_F:        DB 00H,7FH,09H,09H,09H,01H
D_G:        DB 00H,3EH,41H,41H,49H,38H
D_H:        DB 00H,7FH,08H,08H,08H,7FH
D_I:        DB 00H,00H,41H,7FH,41H,00H
D_J:        DB 00H,2OH,40H,41H,3FH,01H
D_K:        DB 00H,7FH,08H,14H,22H,41H
D_L:        DB 00H,7FH,40H,40H,40H,40H
D_M:        DB 00H,7FH,02H,04H,02H,7FH
D_N:        DB 00H,7FH,04H,08H,10H,7FH
D_O:        DB 00H,3EH,41H,41H,41H,3EH
D_P:        DB 00H,7FH,09H,09H,09H,06H
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
D_Q:        DB  00H,3EH,41H,51H,21H,5EH
D_R:        DB  00H,7FH,09H,19H,29H,46H
D_S:        DB  00H,26H,49H,49H,49H,32H
D_T:        DB  00H,01H,01H,7FH,01H,01H
D_U:        DB  00H,3FH,40H,40H,40H,3FH
D_V:        DB  00H,1FH,20H,40H,20H,1FH
D_W:        DB  00H,3FH,40H,30H,40H,3FH
D_X:        DB  00H,63H,14H,08H,14H,63H
D_Y:        DB  00H,03H,04H,78H,04H,03H
D_Z:        DB  00H,61H,51H,49H,45H,43H
D_1:        DB  00H,00H,42H,7FH,40H,00H
D_2:        DB  00H,42H,61H,51H,49H,69H
D_3:        DB  00H,21H,41H,45H,4BH,31H
D_4:        DB  00H,18H,14H,12H,7FH,10H
D_5:        DB  00H,27H,45H,45H,4DH,39H
D_6:        DB  00H,3CH,4AH,49H,49H,60H
D_7:        DB  00H,01H,71H,09H,05H,03H
D_8:        DB  00H,36H,49H,49H,49H,36H
D_9:        DB  00H,06H,49H,49H,29H,1EH
D_0:        DB  00H,3EH,51H,49H,45H,3EH
D_DOT:      DB  00H,00H,4OH,00H,00H,00H
D_PLUS:     DB  00H,08H,08H,7FH,08H,08H
D_MINUS:    DB  00H,08H,08H,08H,08H,08H
D_MULTI:    DB  00H,14H,08H,3EH,08H,14H

            END
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

สุนทร วิทูสุรพงษ์ **ไมโครคอนโทรลเลอร์ตระกูล 8051 :** ซีเอ็ดยูเคชั่นจำกัด (มหาชน)

วรสารเซมิคอนดัคเตอร์อิเลคทรอนิคส์ : ซีเอ็ดยูเคชั่นจำกัด (มหาชน)

The 8051 Ayala ,K.J. ,"**The 8051 Microcontroller Architecture ,Programming ,and Application**" West Publishing Company ,1991

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้