คู่มือการทดลองด้วย MCS-51 TRINING BOARD

EXPERIMENT WITH MCS-51 TRINING BOARD

โดย
นายชาตรี บัญญัตินพรัตน์

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตร์บัณฑิต
ภาควิชาเทคโนโลยีการวัดคุมทางอุตสาหกรรม
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2534

009626

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์ ปีการศึกษา 2534

ภาควิชา เทคโนโลยีการวัดคุมทางอุตสาหกรรม
สาขา เทคโนโลยีคอมพิวเตอร์อุตสาหกรรม
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้า เจ้าคุณทหารลาดกระบัง

ผู้จัดทำ
นาย ชาตรี บัญญัตินพรัตน์ รหัส 33162110

อาจารย์ที่ปรึกษา
(ผศ. พิพัฒน์ เลาหสงคราม)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Experiment With MCS-51 Trining Board

Mr. Chatree Banyatnoparat CODE 33.162110

Assist. Prof. PiPat Lalhasongkram ADVISOR

ABTRACT

THIS THESIS IS CONCERNING APPLIED MICRO CONTROLLER TO CONTROL EQUIPMENT. MCS-51 IS USED MICRO CONTROLLER. THIS PROJECT CONSIST WORK THEORY OF EQUIPMENTS (SUCH AS PRINTER , DOT MATRIX LCD , KEY BOARD) , THE PROGRAM IS USING FOR CONTROL EQUIPMENTS BY ASSAMBLY LANGUAGE , AND INTERFACE WITH EQUIPMENTS FOLLOW BY MICRO CONTROLLER.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

การทดลองที่ 1 หน่วยความจำ (MEMORY) และ DECODER	1
การทดลองที่ 2 PORT INPUT/OUTPUT	11
การทดลองที่ 3 DOT MATRIX LCD	24
การทดลองที่ 4 PRINTER PORT	41
การทดลองที่ 5 SERIAL PORT	53
การทดลองที่ 6 KEY BOARD	69
การทดลองที่ 7 สเต็ปปิ้งมอเตอร์ (STAPPING MOTER)	79
คู่มือแผงช่วยการทดลอง	86

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปีการศึกษา 2534

Experiment With MCS-51 Training Board

โดย

นายชาตรี บัญญัตินพรัตน์

อาจารย์ที่ปรึกษา

ผศ. พิพัฒน์ เลาหสงคราม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทคัดย่อ

ในปัจจุบันนี้การนำเอา Micro Controller มาใช้งานในการควบคุมอุปกรณ์ต่าง ๆ กันเป็นอย่างมาก ซึ่ง Micro Controller ที่กำลังและนำเข้ามาใช้งานในปัจจุบัน ก็คือ MCS-51 จึงเป็นส่วนสำคัญอย่างยิ่งที่เราควรที่จะศึกษาการทำงาน วงจร และ การเขียนโปรแกรม ของ Micro Controller ตัวนี้ Interface กับ อุปกรณ์ที่เราจะนำเอา Micrt Controller ไปควบคุม

จุดประสงค์ของโครงงานนี้คือ ทำให้ผู้ที่ต้องการที่จะศึกษา Micro controller MCS-51เบอร์เข้าใจ และ สามารถนำเอาไปใช้งานได้จริง ซึ่งในการทดลองนี้จะมีทั้งวงจร,โปรแกรม ภาษาแอสเซมบลี้ และแผงช่วยในการทดลองในการควบคุม อุปกรณ์ต่าง ๆ ด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทดลองที่ 1
หน่วยความจำ (MEMORY) และ Decoder

วัตถุประสงค์

1. เพื่อเข้าใจการทำงานของระบบ Micro Controller MCS-51 ในการติดต่อกับหน่วยความจำ
2. สามารถออกแบบเพื่อต่อ Micro Controller กับหน่วยความจำ
3. สามารถออกแบบวงจร Decoder ได้

ทฤษฎี

หน่วยความจำ (Memory) คือ ส่วนที่ใช้ในการเก็บข้อมูล (Data) หรือเก็บโปรแกรม (Program) ซึ่งอาจจะอยู่ในรูปของ Tape , Disk หรือ IC หน่วยความจำที่ใช้ในการติดต่อในการเก็บข้อมูล หรือ โปรแกรม ที่ติดต่อกับ Micro Controller โดยตรงจะอยู่ในรูปของ IC สามารถติดต่อข้อมูล และโปรแกรม ด้วยความเร็ว ใน Micro Controller MCS-51 สามารถแบ่งออกได้ 2 ชนิดคือ

INTETNAL MEMORY

Internal Memory คือหน่วยความจำภายใน ที่อยู่ภายในตัว Micro Controller MCS-51 แบ่งออกได้เป็น 2 แบบด้วยกันคือ

Program Memory เป็นหน่วยความจำถาวร ซึ่งใช้เก็บ โปรแกรม และข้อมูล ซึ่งอยู่ภายในของตัว CPU ที่จะใช้ในการทำงานของ Micro Controller MCS-51 โดยที่หน่วยความจำชนิดนี้สามารถเก็บโปรแกรม และข้อมูล อยู่ได้ แม้จะไม่มีไฟ การป้อนโปรแกรม หรือ ข้อมูลเข้าหน่วยความจำชนิดนี้ จะทำได้โดยทางโรงงาน หรือเครื่องเขียนข้อมูลโดยเฉพาะเท่านั้น

Internal Ram เป็นหน่วยความจำที่ไม่ถาวร ที่อยู่ภายในตัว CPU เช่นกัน หน่วยความจำชนิดนี้จะหายไปเมื่อไม่มีไฟเลี้ยงให้กับตัวมัน หน่วยความจำชนิดจะใช้เก็บเป็น พื้นที่ Stack,Register ต่าง ๆ,SFR (Spacial Funtion Register) และใช้เก็บข้อมูลชั่วคราว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Internal Memory จะมีขนาดความจุของหน่วยความจำขึ้นอยู่กับโรงงานที่ผลิตขึ้นมา ซึ่งในตระกูล MCS-51 ได้ออกแบบในการเพิ่มหน่วยตามเบอร์ที่ใช้ในท้องตลาด ดังจะแสดงดังตารางที่ 1

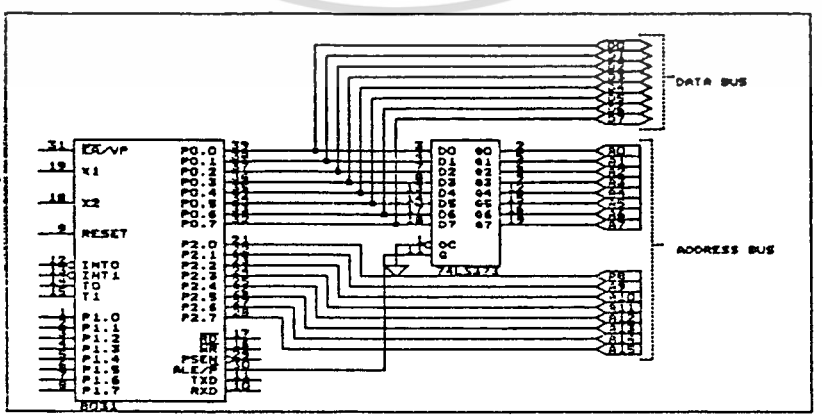
Table 1. The MCS®-51 Family of Microcontrollers

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		✓					5/5	
8051AH	8031AH	8751H 8751BH	4K	128	4	2		✓					6/5	
8052AH	8032AH	8752BH	8K	256	4	3		✓					6/6	
80C51BH	80C31BH	87C51	4K	128	4	2		✓					6/5	✓
83C51FA	80C51FA	87C51FA	8K	256	4	3	✓	✓					14/7	✓
83C51FB	80C51FA	87C51FB	16K	256	4	3	✓	✓					14/7	✓
83C51GA	80C51GA	87C51GA	4K	128	4	2	✓	✓	✓			8	8/7	✓
83C152JA	80C152JA	—	8K	256	5	2		✓		✓	✓		10/11	✓
—	80C152JB	—	—	256	7	2		✓		✓	2		10/11	✓
83C152JC	80C152JC	—	8K	256	5	2		✓		✓	2		10/11	✓
—	80C152JD	—	—	256	7	2		✓		✓	2		10/11	✓
83C451	80C451	—	4K	128	7	2		✓					6/5	✓
83C452	80C452	87C452P	8K	256	5	2		✓					6/6	✓

ตารางที่ 1

External Memory

เป็นหน่วยความจำซึ่งต่อเพิ่มเติมเข้าไป หรือ แทนหน่วยความจำถายในโดยการใช้หน่วยความจำนี้จะใช้ Port ของ Micro Controller จำนวน 2 Port ด้วยกันคือ P0 และ P2 เมื่อใช้ External Memory นี้จะต้องต่อสัญญาณ EA ลง GROUND เพื่อที่จะบอก Micro Control MCS-51 รู้ว่าต้องการที่จะใช้หน่วยความจำถายนอก และ เพิ่มวงจรในการแยกสัญญาณ P0 ซึ่งใช้เป็น Address Bus และ Data Bus ซึ่งจะใช้วงจร LATCH ในการดึงเอาค่า แอดเดรสออกมา โดยใช้ขา ALE เป็นตัว LATCH ที่ได้จากการแยกสัญญาณจะได้ค่าแอดเดรสทางต่ำมาก็คือ A0 - A7 โดยต่อวงจร ตามรูปที่ 1



รูปที่ 1 การแยกสัญญาณระหว่าง Address Bus กับ Data Bus

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การขยายหน่วยความจำถายนอกนี้จะขึ้นอยู่กับผู้ใช้ จะขยายหน่วยความจำเท่าไรก็ได้ แต่สามารถขยายได้สูงสุดไม่เกิน

Program Memory ไม่เกิน 64 Kbyte
Data Memory ไม่เกิน 64 Kbyte

ใน MCS-51 ได้แบ่งหน่วยความจำถายนอกออกเป็น 2 แบบด้วยกัน

หน่วยความจำที่ใช้เก็บโปรแกรม (External Program Memory) เป็นหน่วยความจำที่ถาวร ใช้ เก็บข้อมูล หรือ โปรแกรม หน่วยความจำชนิดนี้จะต้องคงอยู่แม้เราจะไม่ป้อน หรือ หยุดจ่ายไฟเลี้ยง หน่วยความจำชนิดนี้มีอยู่ด้วยกันหลายแบบด้วยกัน คือ

ROM (Read Only Memory)เป็นหน่วยความจำแบบถาวรซึ่งการเขียนทำได้เพียงครั้งเดียวซึ่งจะทำการโปรแกรมได้จากโรงงานเท่านั้น ไม่สามารถจะแก้ข้อมูลถายหลังได้

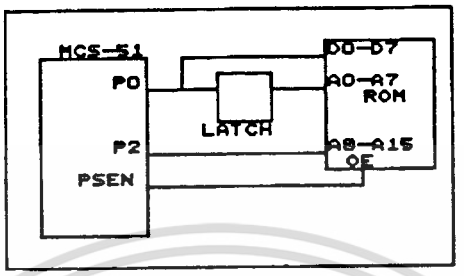
PROM (Program Read Only Memory)เป็นหน่วยความจำแบบถาวรการเขียนโปรแกรม หรือ ข้อมูล ทำได้เพียงครั้งเดียว จะโปรแกรมจากโรงงานหรือ จะโปรแกรมได้ด้วยเครื่องโปรแกรมโดยเฉพาะ การโปรแกรมทำได้โดยป้อนพัลส์ที่มีแรงดันสูง (High Voltage Pulsed) เมื่อโปรแกรมเข้าไปแล้วไม่สามารถจะแก้ข้อมูลถายหลังได้เช่นเดียวกับ ROM

EPROM (Erasable Programable ROM) เป็นหน่วยความแบบถาวรคล้ายกับ ROM,PROM แต่เมื่อโปรแกรมเข้าไปแล้วสามารถลบข้อมูลถายในโดยการฉายแสงอุลตร้าไวโอเลตเข้าไปในตัว ผ่านกระจกใสที่อยู่ในตัว IC เมื่อฉายแสงเข้าไประยะเวลาหนึ่งข้อมูล หรือโปรแกรมถายในจะถูกลบทิ้งไปโดยข้อมูลถายในจะเป็น "1" ทั้งหมด ส่วนของการโปรแกรมจะทำได้โดยป้อนสัญญาณที่มีแรงดันสูง (High Voltage Signal)เข้าไปในตัว IC คล้ายกับ PROM ซึ่งหน่วยความจำชนิดนี้จะนิยมมากเพราะสามารถแก้ไขโปรแกรมได้สะดวก และ ราคาไม่แพงมาก

EAROM (Electrically Alterable ROM)เป็นหน่วยความจำถาวรที่การโปรแกรมคล้ายกับ PROM และ EPROM แต่การลบข้อมูลของ EAROM จะลบด้วยทางไฟฟ้า โดยไม่ต้องฉายแสงอย่าง EPROM

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การต่อใช้งานของหน่วยความจำแบบถาวรนี้จำเป็นต้องใช้สัญญาณ PSEN เป็นสัญญาณที่บอกให้รู้ว่าต้องการจะใช้หน่วยความจำชนิดนี้ การต่อวงจรตามรูปที่ 2 โดย



รูปที่ 2 การต่อ External Program Memory

หน่วยความจำชนิดนี้ส่วนมากจะใช้ เก็บข้อมูลตารางของข้อมูล และ โปรแกรมที่จะใช้ในการทำงาน

การอ้างข้อมูลของ External Program Memory จะใช้คำสั่ง MOVC ในการอ้างข้อมูลในหน่วยความจำ หรือในการอ่านค่าจากตารางในหน่วยความจำ โดย เราจะใช้รีจิสเตอร์ DPTR (Data Pointer) ,รีจิสเตอร์ PC (Program Counter) ในการชี้ตำแหน่งของข้อมูล และใช้รีจิสเตอร์ A (Accumulator) เป็นตัว OffSet ในการบวกค่าของฐานข้อมูลคือ DPTR,PC และเป็นตัวเก็บข้อมูลที่ชี้นั้นไว้ ตัวอย่างเช่น

ตัวอย่างการใช้คำสั่ง MOVC

```
MOV DPTR,#1234H      ;DPTR <= 1234H
MOV A,#5             ;A <= 5
MOVC A,@DPTR+A       ;(DPTR + A) <= A
```

ในตัวอย่าง
คำสั่งแรกกำหนดตำแหน่งของข้อมูล โดยให้ DPTR = 1234H
คำสั่งที่สองกำหนดค่าของ Offset ของข้อมูล A = 5
คำสั่งที่สามจะนำค่าในรีจิสเตอร์มาบวกกัน DPTR + A

1234 + 5 = 1239H

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จะได้ค่าตำแหน่ง 1239H นำข้อมูลในตำแหน่งที่คำนวนได้ไปชี้ข้อมูลแล้วนำมาเก็บไว้ในรีจิสเตอร์ A

A = (1239)

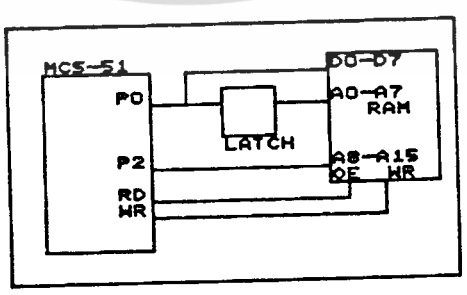
หน่วยความจำที่ใช้เก็บข้อมูล (External Data Memory) หน่วยความจำชนิดนี้ สามารถเขียนและอ่านข้อมูล ข้อมูลที่เขียนลงไปนั้น จะคงอยู่เมื่อมีไฟเลี้ยง แต่เมื่อเลิกจ่ายไฟเลี้ยงข้อมูลที่อยู่ในหน่วยความจำชนิดนี้จะหายไป หน่วยความจำชนิดนี้มีอยู่ด้วยกัน 2 ชนิดคือ

Dynamic RAM (DRAM) หน่วยความจำใช้หลักการเก็บประจุที่ขาเกตของ MOSFET เพื่อเป็นการเก็บข้อมูลและเมื่อมีการอ่านข้อมูลออกมา ประจุที่เก็บไว้จะถูกคายออกมา และ ข้อมูลที่เก็บไว้จะหายไป หน่วยความจำชนิดนี้ก็จะเป็นหน่วยความจำ DRAM จำเป็นจะต้องมีการรีเฟรช เพื่อทำให้ข้อมูลที่อยู่ภายในไม่สูญหายไป ข้อดีของหน่วยความจำชนิดนี้จะมีความจุในหน่วยความจำสูง ใช้แรงดันต่ำ และการเข้าถึงหน่วยความจำได้เร็ว แต่เราต้องเพิ่มวงจรในการสร้างสัญญาณรีเฟรช

Static RAM (SRAM) หน่วยความจำแต่ละหน่วย ใช้ลักษณะการทำงานของ ฟลิบฟลอบ ดังนั้นเมื่อมีไฟเลี้ยงอยู่ หรือไม่มีการเขียนข้อมูลเข้าไปในหน่วยความจำ ข้อมูลที่อยู่ในหน่วยความจำนี้จะไม่หายไป

การใช้หน่วยความจำแบบไม่ถาวรนี้จะใช้เป็นหน่วยความจำของข้อมูลต้องใช้สัญญาณ RD (Read) ในการอ่านข้อมูลจากหน่วยความจำ และสัญญาณ WR (Write) ในการเขียนข้อมูลลงในหน่วยความจำ

การต่อหน่วยความจำแบบไม่ถาวรเข้ากับตัว Micro Controller MCS-51 นี้จะต่อดังจะแสดงในรูปที่ 3



รูปที่ 3 แสดงการต่อ External Data Memory

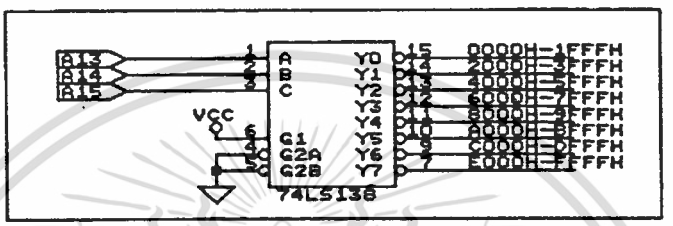
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

วงจรถอดรหัส (DECODER)

จำนวนของวงจรถอดรหัสของแอดเดรสที่จำเป็นต้องใช้ในแต่ละระบบขึ้นอยู่กับขนาดของหน่วยความจำนั้น ๆ ในบางระบบต้องใช้วงจรถอดรหัสเพิ่มเติม

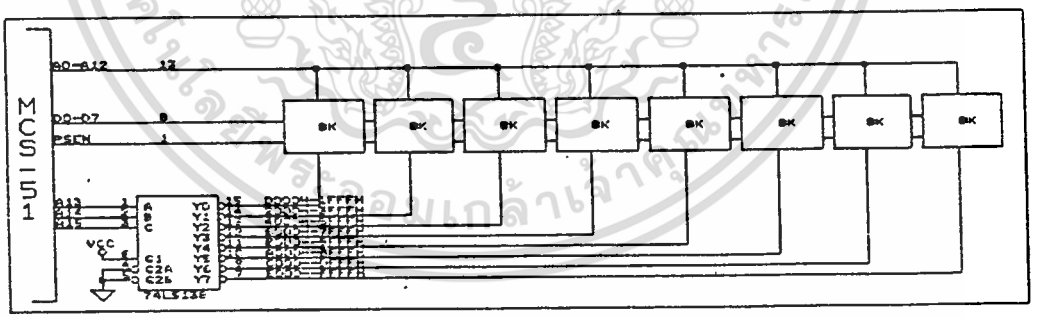
บางทีก็ไม่ต้องใช้ถ้าหน่วยความจำที่ใช้งานจริงๆ ไม่ใช่หน่วยความจำทั้งหมดเราสามารถต่อแอดเดรสเข้าวงจรหน่วยความจำได้เลย

ดังนั้นเราจำเป็นที่จะต้องแบ่งการมองหน่วยความจำของ MCS-51 ออกเป็นหลาย ๆ ช่วง คือการ Decode เช่นเราต้องการใช้ หน่วยความจำโดยแบ่งออกเป็น 2 ช่วงเราก็จำเป็นต้องหาวงจรเพิ่มเข้ามาให้ MCS-51 มองหน่วยความจำเป็นระยะโดยแบ่งระยะละ 8 Kbyte ต่อวงจรดังรูปที่ 4

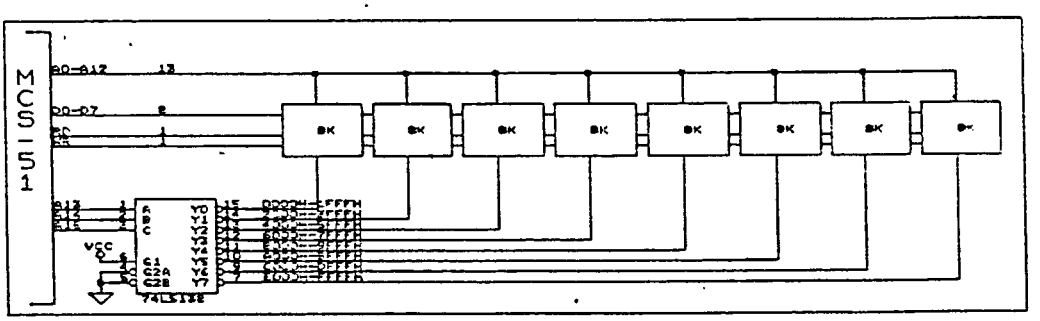


รูปที่ 4 วงจร Decode

จากรูปจะเป็นวงจร Decode โดยใช้เบอร์ 74LS138 3 to 8 Decoder เป็นไอซีที่นิยมใช้ในการ Decoder โดยต่อ A13,A14,A15 เข้าขา A,B,Cตามลำดับ จากรูปที่ 5,6 เป็นการต่อ MCS-51 ต่อหน่วยความจำของโปรแกรมและข้อมูล



รูปที่ 5 เป็นการต่อหน่วยความจำโปรแกรมถายนอก



รูปที่ 6 เป็นการต่อหน่วยความจำของข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อุปกรณ์

1.MCS-51 Traing Board หรือ JAZZ31

2.แผงช่วยฝึกการทดลอง

3.IC 6264

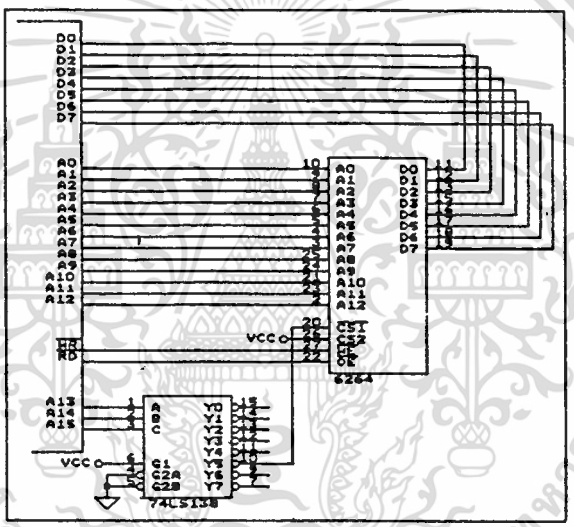
4.IC 74LS138

5.IC 74LS08

6.สายต่อวงจร และ แหล่งจ่ายไฟ

การทดลอง

1.ต่อวงจรตามรูปที่ 7



รูปที่ 7

2.ต่อ PORT1 เข้ากับ Logic Monitor

3.ป้อนโปรแกรมที่ 1 เข้าเครื่อง

```
                    1       ;program read form memory to port 1
                    2
8000                3               org 8000h
8000 90A000         4       start:  mov dptr,#table ;table
8003 E0             5               movx a,@dptr    ;input data form memory
8004 F590           6               mov p1,a        ;output data to port1
8006 80FE           7               sjmp $
                    8
A000                9               org 0a000h
A000 00010203      10       table:  db 00,01,02,03,04,05,06
A004 040506
A007 070B090A      11               db 07,0B,09,0ah,0bh
A00B 0B
A00C 0C0F          12               db 0ch,0fh
```

โปรแกรมที่ 1

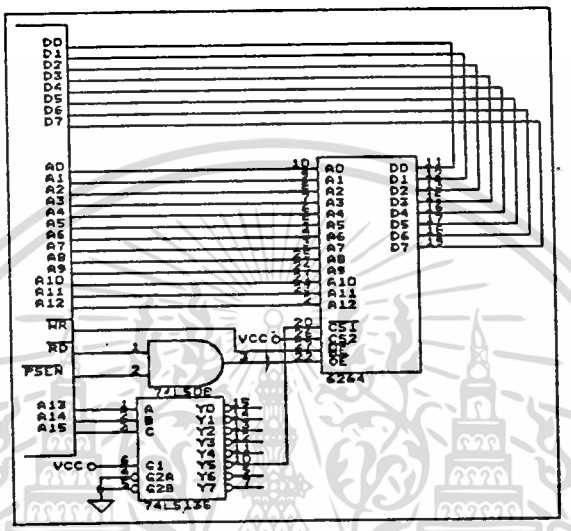
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.สั่งให้โปรแกรมทำงานที่แอดเดรสที่ 8000h

5.สังเกตุที่ Logic Monitor เปรียบเทียบค่าที่แสดงผล กับค่าที่ดูในหน่วยความจำโดยใช้ โปรแกรม Monitor

6.เพิ่มค่าของรีจิสเตอร์ dptr ทีละ 1 ทำตามข้อที่ 5

7.ต่อวงจรตามรูปที่ 8



รูปที่ 8

8.ป้อนโปรแกรมที่ 2 เข้าเครื่อง

```
                    1           ;program read form memory to port 1
                    2
8000                3                   org 8000h
8000 7400           4           start:  mov a,#0          ;offset
8002 900A00         5                   mov dptr,#table ;table
8005 93             6                   movc a,@a+dptr  ;input data form memory
8006 F590           7                   mov p1,a        ;output data to port1
8008 80FE           8                   sjmp $
0A00                9                   org 0a00h
0A00 00010203       10          table:  db 00,01,02,03,04,05,06
0A04 040506
0A07 0708090A       11                  db 07,08,09,0ah,0bh
0A0B 0B
0A0C 0C0F           12                  db 0ch,0fh
```

โปรแกรมที่ 2

10.สั่งให้โปรแกรมทำงานที่แอดเดรสที่ 8000h

11.สังเกตุที่ Logic Monitor เปรียบเทียบค่าที่แสดงผล กับค่าที่ดูในหน่วยความจำโดยใช้ โปรแกรม Monitor

12.เพิ่มค่าของรีจิสเตอร์ A ทีละ 1 ทำตามข้อที่ 11

13.ให้เปรียบเทียบระหว่างโปรมแกรมที่ 1,2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

009629

คำถาม

1.การหน่วยความจำภายในสามารถอ้างข้อมูลโดยท่างอ้อมได้หรือไม่ ถ้าได้ต้องผ่านรีจิสเตอร์อะไรบ้าง จงเขียนโปรแกรมตัวอย่าง

2.การอ้างข้อมูลจากหน่วยความจำภายนอกต้องสามารถชี้ข้อมูลโดยรีจิสเตอร์อะไรบ้าง และมีคำสั่งอะไรบ้างที่สามารถอ้างหน่วยความจำของข้อมูล จงยกตัวอย่างมา 1 คำสั่ง

3.จงออกแบบวงจรในการติดต่อกับหน่วยความจำภายนอกส่วนของ ข้อมูล โดยใช้ IC 6116 (SRAM 4 KByte) โดยกำหนดตำแหน่งเริ่มต้น 4000H-4FFFh จงเขียนวงจรมา และคำสั่งที่สามารถอ้างหน่วยความจำนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทดลองที่ 2
PORT INPUT/OUTPUT

วัตถุประสงค์

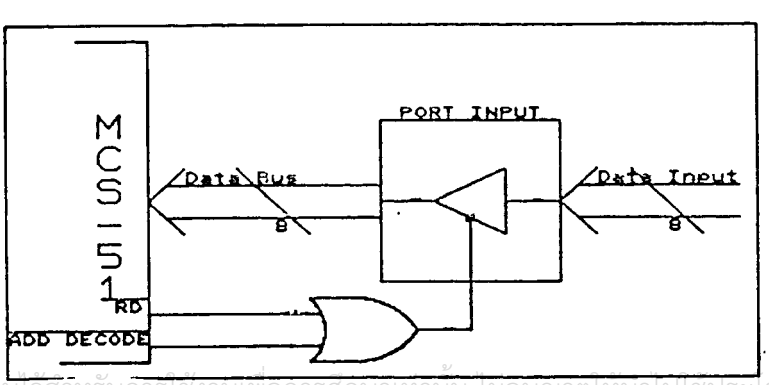
1. เพื่อเข้าใจหลักการในการส่งและรับข้อมูลของ Micro Controller MCS-51 กับอุปกรณ์ภายนอก
2. เพื่อเข้าใจหลักการทำงานของ 8255 I/O Port กับ Micro Controller
3. สามารถนำเอา 8255 I/O Port ไปใช้งาน

ทฤษฎี

ในขบวนการทำงานของระบบ Micro Controller จำเป็นอย่างยิ่งที่จะต้องทำการติดต่อกับอุปกรณ์อื่น ๆ เสมอ นอกจากหน่วยความจำแล้วในบางครั้ง Micro Controller ยังมีความจำเป็นที่จะต้องทำการอ่านข้อมูลจากอุปกรณ์อินพุท และส่งข้อมูลออกไปทางอุปกรณ์เอาท์พุท การเชื่อมต่อระหว่าง Micro Controller MCS-51 เข้ากับอุปกรณ์ I/O (Input/Output Device) จำเป็นต้องรู้จักการทำงานของ PORT ซะก่อน

Port คือการส่งถ่ายข้อมูลระหว่าง Micro Controller กับ อุปกรณ์ภายนอกเราเรียกอีกอย่างว่า อินพุท เอาท์พุท (input output Device)

อินพุท (Input Device)เป็นการรับข้อมูลจากภายนอกเข้ามายัง Micro Controller เพื่อที่จะนำข้อมูลที่ได้เข้ามาประมวลผลการรับข้อมูลจากภายนอกได้ จากอุปกรณ์ตรวจจับสัญญาณต่าง ๆ อุปกรณ์เหล่านี้ เช่น สวิทช์ อุปกรณ์เซนเซอร์ต่าง ๆ ดังที่จะแสดงในรูปที่ 1

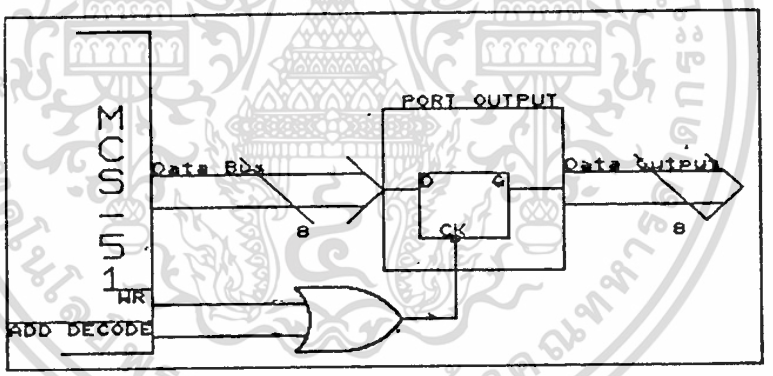


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 1 การรับข้อมูลอุปกรณ์จากภายนอก

การรับข้อมูลจากอุปกรณ์ภายนอกโดยที่มันต่อกับ Data Bus ต้องมี Buffer เป็นตัวกันระหว่างสัญญาณภายนอกกับ Data Bus ของตัว Micro Controller และ จะเป็น Tri-State เพื่อไม่ให้ข้อมูลจากภายนอกไปรบกวน กับ ข้อมูลใน Data Bus ของ Micro Controller และ ปรับระดับของสัญญาณที่เข้ามาให้เหมาะสมกับ Micro Controller สามารถรับข้อมูลได้แตกต่างกันระหว่าง "0" และ "1" ส่วนมากการรับข้อมูลจากภายนอกจะใช้ไอซีเบอร์ 74LS244 ภายใน ไอซีเป็น Buffer และ มีวงจรชมิททริกเกอร์ในการปรับระดับสัญญาณที่เข้ามาให้ได้ระดับที่เหมาะสม

เอาท์พุท (Output Device) เป็นการส่งข้อมูลจากตัว Micro Controller ไปยังอุปกรณ์ภายนอกเพื่อจะควบคุมการทำงานของอุปกรณ์นั้น เช่น จอภาพ เครื่องพิมพ์ หรือ อุปกรณ์ใช้งานอื่น ๆ เช่น รีเลย์ วงจรอิเล็คทรอนิคส์เพื่อควบคุม เป็นต้น ดังที่จะแสดงในรูปที่ 2

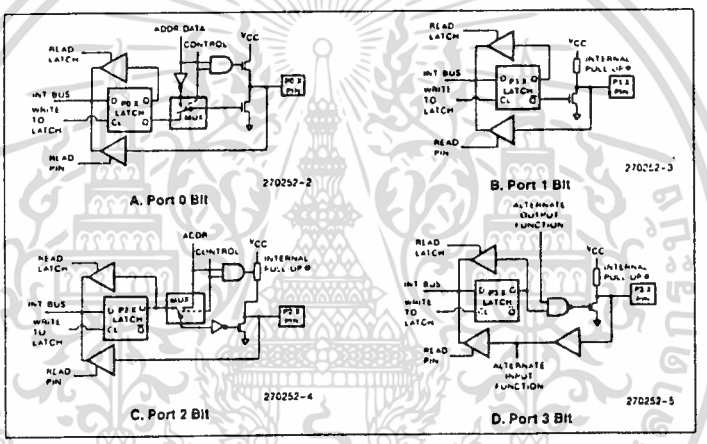


รูปที่ 2 การส่งข้อมูลให้อุปกรณ์จากภายนอก

การส่งข้อมูลไปยังอุปกรณ์ภายนอกนั้นเมื่อส่งข้อมูลออกไปยังอุปกรณ์นั้นจำเป็นอย่างยิ่งที่ต้องมีการเก็บค่าที่ส่งออกไปเพื่อให้อุปกรณ์ภายนอกที่มีความเร็วต่ำสามารถรับข้อมูลจาก Micro Controller ได้ทัน การเก็บค่าหรือการค้างสถานะนั้นเราเรียกอีกอย่างหนึ่งว่าการ แลทช์ (Latch) แม้ข้อมูลบน Data Bus จะเปลี่ยนแปลงไป จะเปลี่ยนข้อมูลได้ก็ต่อเมื่อส่งข้อมูลมาให้มัน ใช้หลักการทำงานของ D F/F ไอซีที่ใช้ในการส่งข้อมูลออกภายนอกจะใช้ โดยมากจะใช้ไอซีเบอร์ 74LS373,74LS374

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภายใน Micro Controller MCS-51 ได้มี PORT ไว้ให้ใช้ถึง 4 PORT เป็นพอร์ตวงจรการแลทช์และบัฟเฟอร์ ที่แสดงในรูปที่ 3 เป็นแบบของบิทของแต่ละพอร์ต ที่พอร์ตที่ 1,2 และ 3 จะมีพูลอัพอยู่ภายในตัว พอร์ต 0 เอาท์พุทเป็นแบบ Open Drain แต่ละเส้นของ I/O จะเป็นอิสระ ในการกำหนดเป็นอินพุท หรือ เอาท์พุท พอร์ต 0 และ 2 อาจใช้เป็น Input/Output ทั่วไปไม่ได้ ถ้าถูกกำหนดให้ใช้เป็น Address Bus และ Data Bus แล้วกรณีการใช้พอร์ตเป็นอินพุทนั้นจะสามารถรับการเปลี่ยนแปลงระดับ Logic จากระดับสูง "1" สู่ ระดับต่ำ "0" เท่านั้น



รูปที่ 3 แสดงวงจรภายในของพอร์ตของ MCS-51

การใช้คำสั่งในการอ้างพอร์ทภายในสามารถใช้คำสั่งได้หลายแบบ สามารถอ้างได้เร็วเพราะจะถือเป็น รีจิสเตอร์ตัวหนึ่ง และสามารถอ้างเป็นบิทก็ได้

การต่ออุปกรณ์ I/O เพิ่มเติม

ใน MCS-51 ไม่ได้ออกแบบในการต่อกับอุปกรณ์ภายนอก จำเป็นต้องใช้พื้นที่ของหน่วยความจำข้อมูล (External Data Memory) ส่วนหนึ่ง โดยถือว่าเป็นหน่วยความจำของข้อมูล ในการต่ออุปกรณ์ I/O เพื่อที่จะติดต่อกับอุปกรณ์ภายนอกนั้นจำเป็นต้องใช้สัญญาณ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

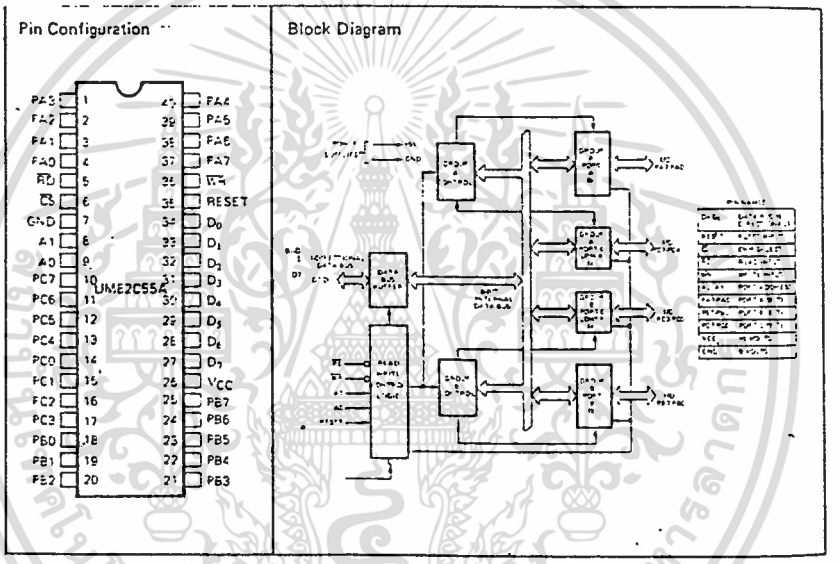
RD (Read) ในการอ่านข้อมูลจากอุปกรณ์ภายนอก

WR (Write) ในการเขียนข้อมูลไปที่อุปกรณ์ที่จะควบคุม

Data Bus เป็นทางผ่านของข้อมูลมีอยู่ 8 เส้นด้วยกันคือ D0 - D7

Address Decoder สัญญาณในการเลือกตำแหน่ง

อุปกรณ์นำมาติดต่อกับอุปกรณ์ภายนอกที่นิยมใช้ในปัจจุบัน ใช้ไอซี เบอร์ 8255 เป็นไอซีสามารถเป็นได้ทั้งอินพุท และ เอาท์พุท โดยที่เราสามารถกำหนดได้ ซึ่งตำแหน่งขาและโครงสร้างภายในดังจะแสดงในรูปที่ 4



รูปที่ 4 แสดงขาและโครงสร้างภายในของ 8255

จากรูปที่ 4 ภายใน 8255 มี พอร์ทPort อยู่ทั้งหมด 3 พอร์ทด้วยกันคือ Port A,Port B มีขนาด 8 Bit และ Port C โดยที่ Port C ยังสามารถแบ่งออกเป็น Port ละ 4 Bit คือ PC0-PC3 (Port C ล่าง) และ PC4-PC7 (Port C บน)

ขาต่าง ๆ ของ 8255

การต่อ 8255 นั้นจำเป็นต้องเข้าใจการทำงานของแต่ละขา

D0-D7 ต่อเข้ากับ Data Bus ของตัว MCS-51 เพื่อใช้สำหรับรับส่งข้อมูลกันระหว่าง Port กับ Micro Controller

CS เป็นสัญญาณในการชี้ตำแหน่งแอดเดรสของ Port ซึ่งใน 8255 จำเป็นต้องใช้ตำแหน่งของ Port ทั้งหมดด้วยกัน 4 ตำแหน่งด้วยกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

RD เป็นขบวนการรับข้อมูลจาก Port Input เมื่อ CS และ RD เป็น "0"

WR เป็นขบวนการส่งรับข้อมูลจาก Port Output เมื่อ CS และ WR เป็น "0"

RESET จะทำการ Clear สถานะของ 8255 ทั้งหมดเมื่อเป็น "1" Port ต่างๆ ของเป็น อินพุททั้งหมด

A0-A1 เป็นสัญญาณชี้ตำแหน่งแอดเดรสต่อจาก CS โดยเลือกตำแหน่งของ Port ภายใน 8255 โดยที่ 8255 แบ่งออกด้วยกัน 4 Port ด้วยกัน คือ Port A,Port B,Port C และ Control Port โดยการเลือกจะเรียงตามลำดับดังนี้

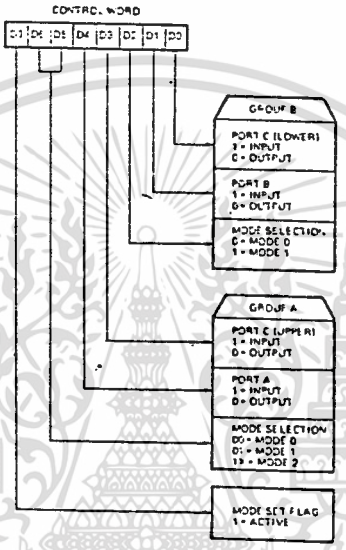
CS	A1	A0	ความหมาย
1	X	X	X
0	0	0	PORT A
0	0	1	PORT B
0	1	0	PORT C
0	1	1	CONTROL PORT

ตารางการเลือกตำแหน่งของ Port ภายในของ 8255

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การกำหนดการทำงานของ 8255

ก่อนการใช้งานของ 8255 จะต้องส่งรหัสควบคุม (Control Word) เข้าไปยัง Control Port ซึ่งเป็น Port ที่ 4 ของ 8255 เพื่อควบคุมการทำงานของ 8255 ว่าให้ทำงานใน Mode ไหน และให้แต่ละ Port (Port A,B, C) เป็น Input หรือ Output ความหมายของบิทต่าง ๆ ใน Control Port แสดงดังรูปที่ 5



รูปที่ 5 แสดงค่าบิทของ Control Port

การเปลี่ยนของแต่ละบิทมีความหมายดังต่อไปนี้

D7 เป็นบิทที่แสดงรหัสคำสั่งควบคุม ถ้าบิทนี้เป็น "1" หมายถึงรหัสควบคุมนี้มีผลต่อการเปลี่ยนแปลงการเซตโหมดต่าง ๆ ของ 8255

D6 และ D5 เป็นการเซ็ตโหมดการทำงานของ 8255 ซึ่งมี 3 โหมดด้วยกันคือ Mode 0,Mode 1 และ Mode 2 ซึ่งแสดงในรูปที่ 5

D4 เป็นการกำหนดทำงานของ Port A ว่าจะให้เป็น อิทพุท หรือ เอาท์พุท โดยถ้าเป็น "1" ก็กำหนดให้เป็น อินพุท และ ถ้าเป็น "0" ก็กำหนดให้เป็น เอาท์พุท

D3 เป็นการกำหนดทำงานของ Port C บน ว่าจะให้เป็น อิทพุท หรือ เอาท์พุท โดยถ้าเป็น "1" ก็กำหนดให้เป็น อินพุท และ ถ้าเป็น "0" ก็กำหนดให้ Port C บนเป็น เอาท์พุท

D2 เป็นการเลือกโหมดการทำงานของ Port B ซึ่งถ้าเป็น "0" เป็นการกำหนดให้ Port B ทำงานใน Mode 0 ถ้าเป็น "1" เป็นการกำหนดให้ Port B ทำงานใน Mode 1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

D1 เป็นการกำหนดทำงานของ Port B ว่าจะให้เป็น อิทพุท หรือ เอาท์พุท โดยถ้าเป็น "1" ก็กำหนดให้เป็น อินพุท และ ถ้าเป็น "0" ก็กำหนดให้เป็น เอาท์พุท

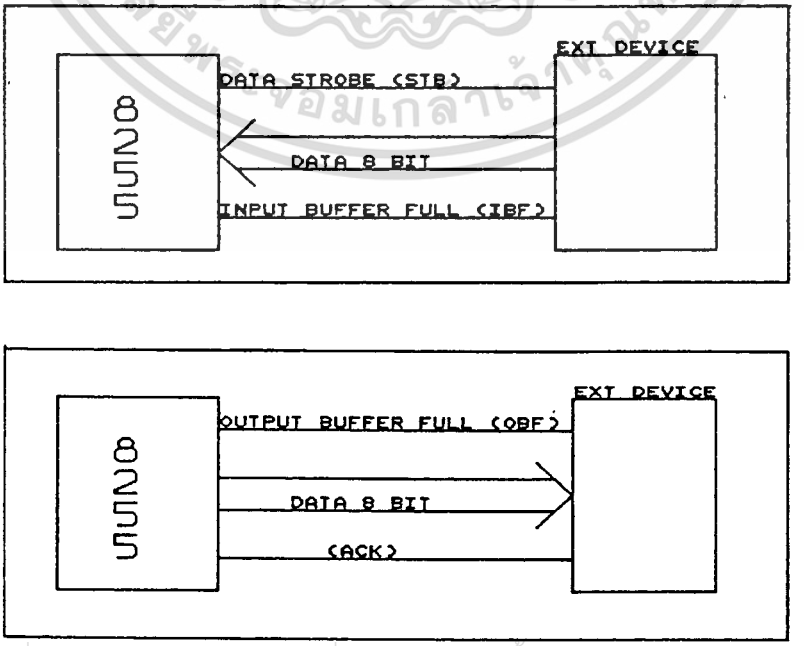
D0 เป็นการกำหนดทำงานของ Port C ล่าง ว่าจะให้เป็น อิทพุท หรือ เอาท์พุท โดยถ้าเป็น "1" ก็กำหนดให้เป็น อินพุท และ ถ้าเป็น "0" ก็กำหนดให้เป็น เอาท์พุท

การทำงานในโหมดต่าง ๆ ของ 8255

Mode 0 เป็นการกำหนดให้ 8255 ทำงานเป็น Port Input และ Port Output แบบพื้นฐาน

Mode 1 การทำงานของ 8255 ในโหมดนี้เป็นการส่งข้อมูล หรือ รับข้อมูล จากอุปกรณ์ ภายนอกโดยมีการตรวจสอบสัญญาณความพร้อมของอุปกรณ์ภายนอก (Hadn Shake) โดยใช้ Port A,Port B เป็นพอร์ทของข้อมูลส่วนพอร์ท C นี้จะถูกใช้เป็นสัญญาณ Hand Shake โดย 4 บิทบนจะเป็นสัญญาณ Hand Shake ของ Port A และ 4 บิทล่างจะเป็นสัญญาณ Hand Shake ของ Port B

หลักการรับส่งข้อมูลในวิธีการตรวจสอบความพร้อมของอุปกรณ์ภายนอกซึ่งมีความเร็วต่ำนั้น คือการให้อุปกรณ์ภายนอกทำการส่งสัญญาณ แสดงสภาวะความพร้อมของอุปกรณ์ให้กับ 8255 ดังที่จะแสดงในรูปที่ 6



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 6 แสดงการทำงานของ 8255 ในการทำงานในโหมดที่ 1

จากรูป a แสดงการส่งข้อมูลจากอุปกรณ์ภายนอกเข้าสู่ 8255 ก่อนที่อุปกรณ์ภายนอกจะทำการเขียนข้อมูลลงไปนั้นจะทำการตรวจสอบว่า 8255 พร้อมที่จะรับข้อมูล หรือไม่ โดยทำการเช็คที่สัญญาณ IBF ซึ่งอยู่ที่ Port C เป็นตัวที่จะบอกอุปกรณ์ภายนอกว่า 8255 พร้อมที่จะรับข้อมูลหรือยัง ถ้าพร้อมแล้วก็จะส่งสัญญาณ IBF นี้ออกมาเมื่ออุปกรณ์ภายนอกตรวจสอบว่า 8255 พร้อมที่จะรับข้อมูลแล้ว ก็จะส่งข้อมูลไปยัง 8255 แล้ว 8255 ก็ทำการส่งสัญญาณ STB (Strobe) เพื่ออ่านข้อมูลจากอุปกรณ์กภายนอกที่ส่งเข้าสู่ 8255

จากรูป b แสดงการส่งข้อมูลออกจาก 8255 ไปที่อุปกรณ์ภายนอก ก่อนที่ 8255 จะทำการส่งข้อมูลไปยัง อุปกรณ์ภายนอก นั้นจะทำการตรวจสอบว่าอุปกรณ์ ภายนอกนั้น พร้อมที่จะรับข้อมูล หรือไม่ โดยทำการเช็คที่สัญญาณ ACK (Acknowledge) ซึ่งอยู่ที่ Port C ซึ่งอุปกรณ์ภายนอกจะเป็นตัวส่งมา โดยเป็นตัวที่จะบอกเช็คว่า อุปกรณ์ภายนอกพร้อมที่จะรับข้อมูลหรือยัง หรือ ข้อมูลที่ส่งออกมา 8255 ได้ถูกทำการอ่านข้อมูลไปหรือยัง ถ้าอุปกรณ์ภายนอกส่งสัญญาณ ACK ว่าพร้อมที่จะรับข้อมูลแล้ว เมื่ออุปกรณ์ 8255 ตรวจสอบว่า อุปกรณ์ภายนอกพร้อมที่จะรับข้อมูลแล้ว ก็จะส่งข้อมูลจาก 8255 ไปยังอุปกรณ์ภายนอก แล้วก็ทำการทำให้ สัญญาณ OBF เป็น "1" เพื่อทำการบอกให้ อุปกรณ์ภายนอก รู้ว่าข้อมูลถูกส่งออกมาจาก 8255 แล้ว

ตารางที่ 1 เป็นตารางแสดงสัญญาณของ Port Cเมื่อกำหนดให้ 8255 ทำงานใน Mode 1

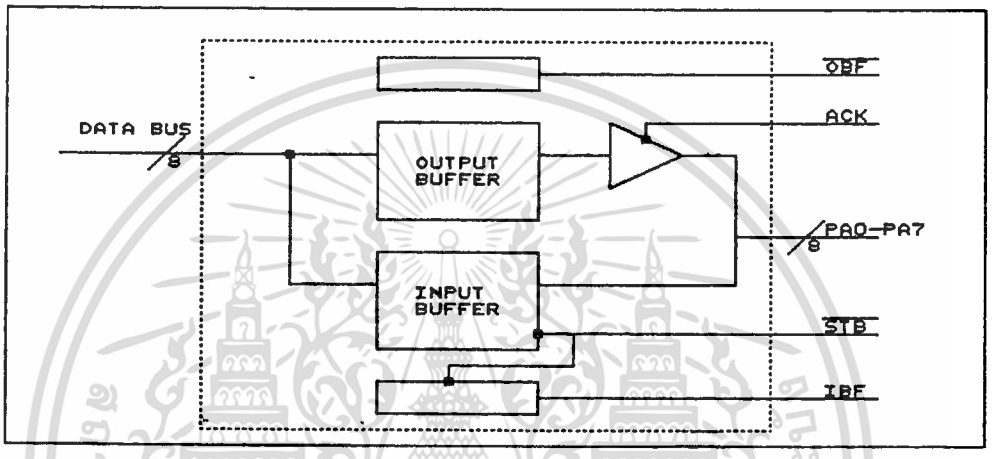
PORT C	OUT	IN
PC0	$INTR_B$	$INTR_B$
PC1	IBF_B	OBF_B
PC2	STB_B	ACK_B
PC3	$INTR_A$	$INTR_A$
PC4	STB_A	I/O
PC5	IBF_A	I/O
PC6	I/O	ACK_A
PC7	I/O	OBF_A

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 1 หน้าที่ต่างๆ ของ Port C ซึ่ง 8255 ทำงานใน Mode 1

Mode 2

ในโหมดนี้จะใช้ได้เฉพาะ Port A เท่านั้น ซึ่งจะทำหน้าที่เป็น Port 2 ทิศทาง คือสามารถเป็น Port Input และ Port Output โดยการส่งและรับข้อมูลของ Port A จะมีการตรวจสอบความพร้อมของอุปกรณ์ภายนอก (Hand Shake) ทั้งคู่ ส่วน Port C ทำหน้าที่สำหรับตรวจสอบ โครงสร้างของ 8255 ที่ทำงานใน Mode 2 ดังที่จะแสดงในรูปที่ 7



รูปที่ 7 โครงสร้างของ 8255 ทำงานใน Mode 2

จากรูปที่ 7 จะเห็นว่า Port A ของ 8255 สามารถเป็นได้ทั้ง Input และ Output โดยการทำงานของโหมดนี้ใช้ Port A เป็นทั้ง Input Latch และ Output Latch

Input Latch คือการเก็บข้อมูลที่ถูกส่งจากอุปกรณ์ภายนอกไว้เพื่อรอที่จะให้ CPU ทำการอ่านข้อมูลเข้าไป

Output Latch คือการเก็บข้อมูลไว้เพื่อรออุปกรณ์ภายนอกรับข้อมูลออกไป

ในการรับและส่งของข้อมูลระหว่าง 8255 กับอุปกรณ์ภายนอกเช็คสัญญาณที่ Port C ก่อนที่จะรับและส่งข้อมูลออกไป รายละเอียดของแต่ละขาของ Port C จะแสดงในตารางที่ 2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

PORT C LINE	ชื่อของสัญญาณ
PC0	I/O
PC1	I/O
PC2	I/O
PC3	INTR
PC4	STB
PC5	IBF
PC6	ACK
PC7	OBF

ตารางที่ 2 แสดงความหมายของ Port C ของ 8255 ใน Mode 2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อุปกรณ์

1.MCS-51 Traing Board หรือ JAZZ31

2.แผงช่วยฝึกการทดลอง

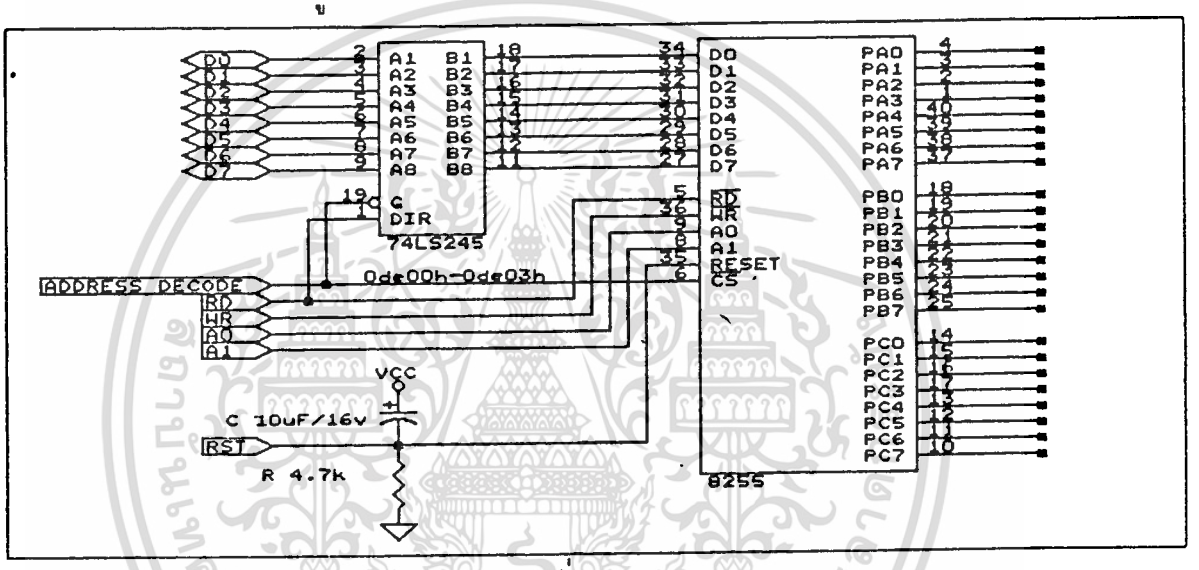
3.IC 8255

4.IC 74LS245

5.สายต่อวงจร และ แหล่งจ่ายไฟ

การทดลอง

1.ต่อวงจรตามรูปที่ 8



รูปที่ 8

2. ให้ต่อ Port A เข้า Logic Monitor และ ต่อ Port B เข้าที่ Logic Switch

3. ป้อนโปรแกรมที่ 1 เข้าเครื่อง แล้วสั่งให้โปรแกรมทำงาน

```
                    1  ;Program Test Port 8255
8000                2          org 8000h
DE00=               3          porta   equ  0de00h
DE01=               4          portb   equ  0de01h
DE02=               5          portc   equ  0de02h
DE03=               6          cont    equ  0de03h
8000 90DE03         7  start:  mov dptr,#cont  ;Select Control port
8003 7482           8          mov a,#10000010b;port a Output
                    9                          ;port b Input
                   10                          ;port c Output
8005 F0            11          movx @dptr,a
8006 7400          12          mov a,#0h       ;Clear Logic Monitor
8008 90DE00        13  st:     mov dptr,#porta ;Select Port A
800B F0            14          movx @dptr,a    ;Output Logic Monitor
800C 90DE01        15          mov dptr,#portb ;Select Port B
800F E0            16          movx a,@dptr    ;Input  Logic Switch
8010 80F6          17          sjmp st
```

โปรแกรมที่ 1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. ให้เปลี่ยนข้อมูลของ Logic Switch และ สังเกตุค่าที่เปลี่ยนแปลงที่ Logic Monitor

5. ให้เปลี่ยนย้าย Logic Switch มาที่ Port C

6. ให้เปลี่ยนค่าของ Control Port เป็น 88h

7. ทดลองตามข้อที่ 4

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คำถาม

1.เราสามารถกำหนดรูปแบบของการทำงานใน Mode 0 ได้กี่แบบ

2.จงเขียนโปรแกรมให้ 8255 ทำงานใน Mode 1 ให้ Port A เป็น Output และ Port B เป็น Output

3.เราจะใช้ Mode 1 ในการติดต่อกับอุปกรณ์อะไรบ้าง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทดลองที่ 3
DOT MATRIX LCD MODULE

วัตถุประสงค์

1. เพื่อที่จะศึกษาอุปกรณ์แสดงผลแบบ Dot Matrix LCD MODULE
2. สามารถเขียนโปรแกรมในการควบคุมการทำงานของ Dot Matrix LCD MODULE
3. สามารถประยุกค์นำเอา Dot Matrix LCD MODULE มาใช้งานในการแสดงผล

ทฤษฎี

อุปกรณ์ในปัจจุบันในส่วนแสดงผลนั้นจะใช้ LCD เสียส่วนใหญ่ไม่ว่าจะเป็นเครื่องคิดเลข เครื่องถ่ายเอกสาร เครื่องมือวัดต่าง ๆ ไปจนถึง คอมพิวเตอร์ สามารถแบ่งการแสดงผลแบบ Dot Matrix LCD ได้ 3 แบบด้วยกันคือ

1. Character LCD MODULE
2. Graphic LCD MODULE
3. Segment Display ชนิด LCD MODULE

โดยในแต่ละแบบนี้ก็จะมีส่วนประกอบใหญ่ ๆ แบ่งได้เป็น

1. DOT MATRIX LCD เป็นตัวแสดงผลให้เราได้มองเห็นในลักษณะการปิด และ เปิดตัวเอง กับ แสงภายนอก ส่วนของที่เป็นตัวกระจกบรรจุผลิก

2. DRIVER เป็นตัวรับสัญญาณจากตัวควบคุมมาขับผลิก LCD อีกทีหนึ่ง โดยมีเบอร์ที่นิยมใช้ใน LCD Module เช่น HD44100H,MSM5259

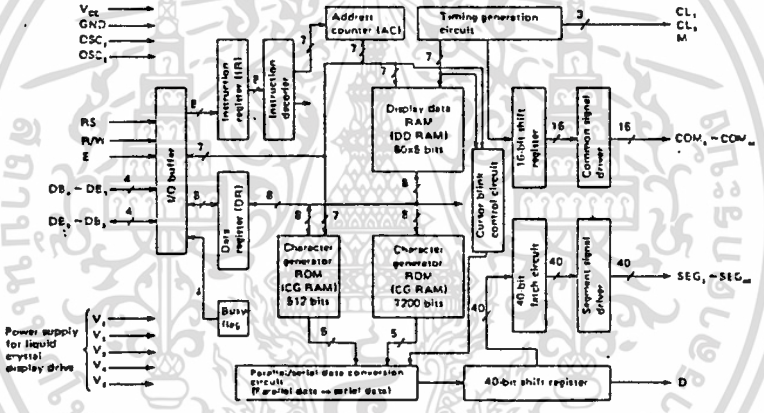
3. CONTROLLER เป็นตัวรับข้อมูลจากตัวอุปกรณ์ภายนอกมาและจัดการควบคุม LCD Module ให้ทำงานแสดงผลต่างๆ เช่น การลบจอถาพ,การเกิดตัวอักษร เป็นต้น โดยมีเบอร์ที่นิยมใช้กันคือ HD44780 ซึ่งจะใช้ในแบบ Character LCD Module เป็นส่วนใหญ่ และ เบอร์ HD61830 จะใช้ในแบบ Graphic LCD Module

การควบคุมตัว Controller ของ LCD นั้นต้องเข้าใจหลักการทำงานของ Controller ก่อน ซึ่ง LCD Module ของแต่ละบริษัทจะใช้ตัว Controller ที่มีหลักการทำงานเหมือน ๆ กันเป็นส่วนใหญ่และใน LCD

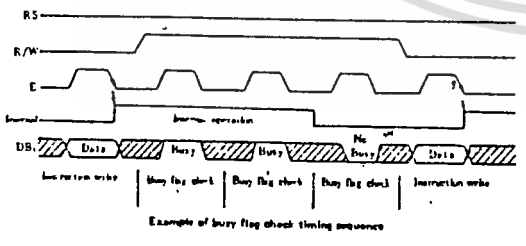
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

MODULE แต่ละขนาดจำนวนตัวอักษรหรือจำนวนบรรทัดก็มีหลักการทำงานแบบเดียวกันทั้งหมด IC ที่นิยมมากที่สุดตัวหนึ่งที่เป็น Controller LCD ก็คือเบอร์ HD44780 โดยรูปแบบการทำงานของมันได้เป็นมาตรฐานให้กับ Controller LCD ตัวอื่น ๆ ด้วย

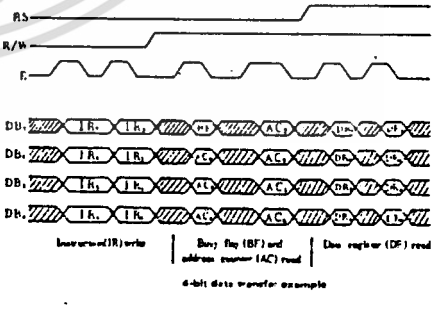
HD44780 เป็นไอซี LSI ตัวหนึ่งใช้ควบคุม LCD โดยแสดงผลในรูปตัวอักษรหรือสัญลักษณ์ต่างๆ ตัวมันเองสามารถต่อใช้งานได้ทั้งแบบ 4 Bit และ 8 Bit โดยถ้าเราต่อแบบ 4 Bit จะต่อใช้งานที่ DB7 - DB4 เท่านั้นโดยข้อมูลครั้งแรกที่ส่งนั้น HD44780 จะถือเป็นข้อมูล 4 Bit บน และ ข้อมูลที่ส่งต่อมานั้นเป็นข้อมูล 4 Bit ล่าง จากรูปที่ 1 จะแสดงโครงสร้าง และ การทำงานภายในตัว Controller และ Timing Diagram ในการรับข้อมูลของตัว Controller



โครงสร้างภายใน Controller



สัญญาณการติดต่อแบบ 8 Bit



สัญญาณการติดต่อแบบ 4 Bit

รูปที่ 1 แสดงโครงสร้างของ Controller และ Time Diagram ในการติดต่อข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 1 จะเห็นว่าในโครงสร้างจะเห็นว่าโครงสร้างภายในของ Controller จะประกอบหลายส่วนด้วยกัน ได้แก่

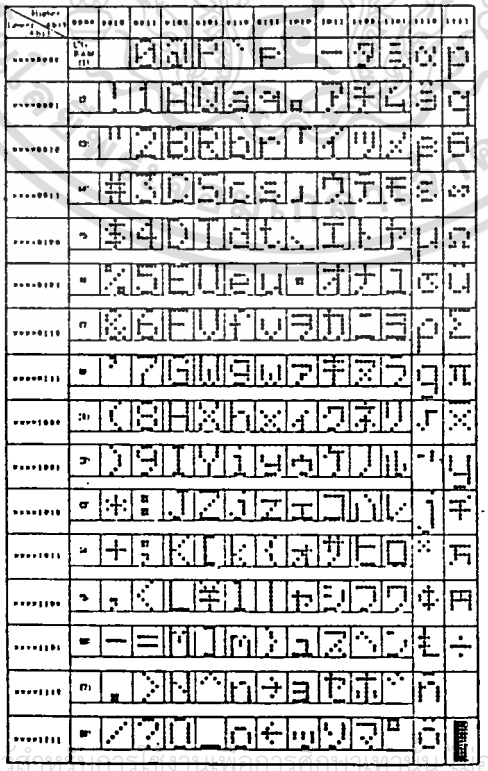
I/O Buffer เป็นวงจรในการติดต่อข้อมูลจากภายนอกเพื่อที่จะส่งข้อมูลเข้าไปยังภายในตัว Controller

Instruction Register (IR) เป็นรีจิสเตอร์ที่จะรับคำสั่งจากภายนอกเพื่อที่จะควบคุมการแสดงผล เช่น การจัดการแสดงผล,การ Clearจอ, กำหนดตำแหน่งของการแสดงผล เป็นต้น

Data Register (DR) เป็นรีจิสเตอร์ที่จะใช้รับข้อมูลจากภายนอกส่งไปให้ DD RAM หรือ สร้างตัวอักษรเพิ่มเติมให้กับ CG RAM

Display Data RAM (DD RAM) เป็นหน่วยความจำที่สามารถเขียนข้อมูลเข้าไปได้โดยผ่าน Data Register (DR) ซึ่งตัว Controller จะนำข้อมูลที่อยู่ใน DD RAM ไปชี้ค่าที่ตาราง หรือ ตำแหน่ง ของ CG ROM หรือ CG RAM เพื่อจะนำไปแสดงที่หน้าจอ LCD

Character Generater ROM (CG RAM) เป็นหน่วยความจำที่ถาวรใช้เก็บ Character ของตัวอักษร หรือ สัญลักษณ์ ที่จะนำไปแสดงที่หน้าจอ LCD และยังสามารถสร้าง Font ขึ้นมาใช้เองได้โดยการเขียนข้อมูลลงไปใน CG RAM และ ชี้ Font ได้โดยการกำหนดค่าที่ DD RAM ตารางของข้อมูลภายใน CG ROM จะแสดงในตารางที่ 1



ตารางที่ 1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Busy Flag เป็นตัวบอกสถานะการทำงานของ Controller ให้อุปกรณ์ภายนอกทราบว่า Controller พร้อมที่จะรับข้อมูล หรือ คำสั่ง ที่จะทำการส่งไปให้ตัว Controller หรือยัง โดยให้อุปกรณ์ภายนอกทำการอ่านสัญญาณ ตัวนี้งานเสร็จหรือยัง

เมื่อเราป้อนไฟให้ HD44780 นั้นก็จะทำการ Reset ตัวเองโดยจะใช้เวลาประมาณ 10 mS หลังจากไฟ VDD ถึง 4.5 V แล้ว โดยจะ SET ตัวเองดังนี้

1.Display Clear จะทำการลบข้อมูลจอภาพ LCD

2.Function Set โดยจะ SET ค่าภายใน

DL = 1 เป็นการ Set ให้การติดต่อเป็นแบบ 8 Bit

N = 0 Set เป็น 1 บรรทัดการแสดงผล

F = 0 5*7 Dot ต่อหนึ่งตัวอักษร

3.Display ON/OFF

D = 0 Display OFF

C = 0 Cursor OFF

B = 0 Blink OFF

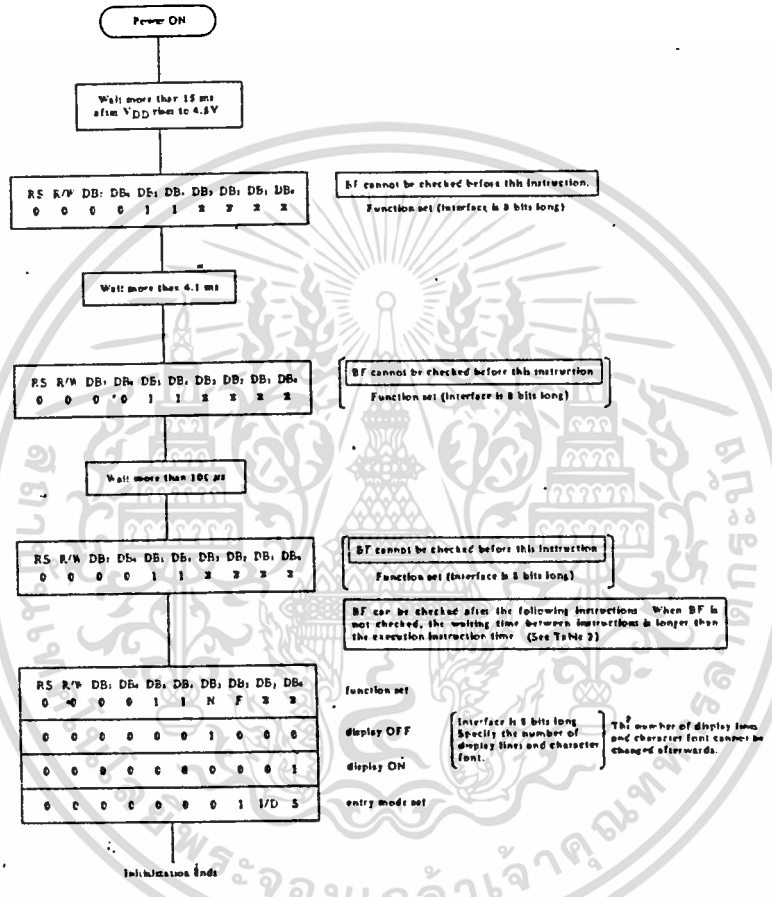
4.Entry Mode Set

I/D = 1 (เพิ่มค่า Counter ขึ้น 1)

S = 0 NO Shift

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อเราเริ่มเปิดเครื่องทำงานแล้วก็จะต้องส่งคำสั่งควบคุมให้มันเริ่มทำงานตาม Flow Chart ที่ 1 ดังนี้



Flow Chart ที่ 1 การติดตั้งระบบการแสดงของ LCD เมื่อเปิดเครื่อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รายละเอียดของคำสั่ง HD44780

Clear Display

RS R/W DB7 ---------- DB0

0	0	0	0	0	0	0	0	0	1

เป็นคำสั่งที่จะเขียนช่องว่าง หรือ Space (20h) เข้าไปใน DD RAM ทั้งหมด และ ทำการ Set DD RAM Aaddress เป็น 0 ตัว Cursor จะกลับไปอยู่ที่ตำแหน่งบนสุดทางซ้ายมือของ จอภาพ และทำการ Set I/D = 1, S ไม่มีการเปลี่ยนแปลง

Return Home

RS R/W DB7 ---------- DB0

0	0	0	0	0	0	0	0	1	*

เป็นคำสั่งที่จะทำการ Set DD RAM Address เป็น 0 ตัว Cursor จะกลับไปอยู่ตำแหน่งบนสุดซ้ายมือของจอภาพ ข้อมูลในจอภาพไม่เปลี่ยนแปลง

Entry Mode Set

RS R/W DB7 ---------- DB0

0	0	0	0	0	0	0	1	I/D	S

BIT I/D โดยเป็นตัวกำหนดให้ว่าเมื่อเขียนหรืออ่านข้อมูลแล้วจะทำให้ DD RAM Address เพิ่มขึ้นหนึ่งหรือลดลงหนึ่งโดย 1 = เพิ่ม, 0 = ลดลงหนึ่ง

Bit S เป็นตัวกำหนดการแสดงผลโดยถ้า S = 1 จะเป็นการใส่ข้อมูลแล้วตัว Cursor อยู่กับที่ข้อมูลจะถูกดันไปทางซ้าย ถ้า S = 0 ข้อมูลจะอยู่ที่ตัว Cursor จะถูกดันไปทางขวามือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Display on/off Control

RS R/W DB7 ---------- DB0

0	0	0	0	0	0	1	D	C	B

D เป็น Bit ให้เปิดปิดหน้าจอภายโดยถ้า D = 1 จะ ON และ D = 0 จะ OFF

C จะให้แสดง Cursor ให้ Bit C = 1 และถ้าไม่ต้องการแสดง Cursor Bit = 0 โดยตัว Cursor จะอยู่ที่ Line ที่ 8 ในแบบ 5*7 Dot และจะอยู่ Line ที่ 11 ในแบบ 5*10 Dot

B เป็น Bit Set การหระพริบของ Cursor โดย B = 1 มีการกระพริบ B = 0 ไม่มีการกระพริบ โดยมีระยะเวลาการกระพริบประมาณ 379.2 mS

Cursor or Displey Shift

RS R/W DB7 ---------- DB0

0	0	0	0	0	1	S/C	R/L	*	*

S/C	R/L	
0	0	ทำการย้าย Cursor ไปจากตำแหน่งเดินไปซ้ายมือ 1 ตำแหน่ง
0	1	ทำการย้าย Cursor ไปจากตำแหน่งเดินไปขวามือ 1 ตำแหน่ง
1	0	เป็นการดันตัวอักษรที่เกิดไปทางซ้าย
1	1	เป็นการดันตัวอักษรที่เกิดไปทางขวา

Function Set

RS R/W DB7 ---------- DB0

0	0	0	0	1	DL	N	F	*	*

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

DL เป็นการ SET การติดต่อว่าจะให้เป็นแบบ 8 Bit หรือ 4 Bit โดยถ้าต้องการติดต่อ 4 Bit DL = 0 และ 8 Bit DL = 1

N เป็นการ SET บรรทัดของการแสดงผล N = 0 แสดง 1 บรรทัด N = 1 แสดง 2 บรรทัด ในกรณีมากกว่าก็ให้ SET N = 1

F เป็นการ SET ขนาด DOT การแสดงผล 5*7 หรือ 5*10 โดย F = 0 เป็นแบบ 5*7 และ F = 1 เป็นแบบ 5*10

Set CG RAM Address

RS R/W DB7 ---------- DB0

0	0	0	1	A	A	A	A	A	A

<- High Low ->
Order Bit Order Bit

ใน HD44750 นั้นจะมีหน่วยความจำอยู่ 2 ชุด คือ Display DATA RAM (DD RAM) จำนวน 80*8 Bit และ Character Generater ROM CG RAM จะนวน 512 Bit และ 7200 Bit คำสั่งนี้เป็นการ Set Address ใน CG RAM โดยต้องทำการ Set Address ก่อนเขียนหรืออ่านข้อมูล CG RAM ด้วย

Set DD RAM Address

RS R/W DB7 ---------- DB0

0	0	1	A	A	A	A	A	A	A

<- High Low ->
Order Bit Order Bit

เป็นคำสั่ง SET ค่า Address ใน DD RAM ในการเขียนหรืออ่านค่าจาก DD RAM (DD RAM คือส่วนที่จะแสดงผลหน้าจอ LCD) โดยจำนวน Address ที่จะเกิดขึ้นบนจอ LCD จะอยู่กับ SET ค่า N ด้วย

ถ้า N = 0 (1 บรรทัด) Address จะอยู่ 00H-4fH

ถ้า N = 1 (2 บรรทัด) Address จะอยู่ 00H-27H สำหรับบรรทัดที่ 1 และ 40H-67H สำหรับบรรทัดที่ 2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Read Busy Falg And Address

RS R/W DB7 ---------- DB0

0	0	1	BF	A	A	A	A	A	A

<- High Low ->

Order Bit Order Bit

เป็นคำสั่งอ่านค่า Busy Flag ซึ่งจะเป็นตัวบอกว่าตัว HD44780 นี้อยู่ในขบวนการทำงานภายในอยู่หรืออยู่ในสภาพพร้อมจะรับข้อมูล โดย

BF = 1 อยู่ในขบวนการทำงานภายในไม่พร้อมจะรับข้อมูลหรือคำสั่ง

BF = 0 พร้อมจะรับข้อมูลหรือคำสั่งได้

และนอกจากนี้ยังเป็นคำสั่งอ่านค่าข้อมูล Address ของ CG RAM หรือ DD RAM ด้วย

Write Data to CG หรือ DD RAM

RS R/W DB7 ---------- DB0

1	0	D	D	D	D	D	D	D	D

<- High Low ->

Order Bit Order Bit

เป็นคำสั่งเขียนข้อมูลเข้าไปใน CG หรือ DD RAM โดยเมื่อเขียนข้อมูลและ Address จะเพิ่มหรือ ลด โดยอัตโนมัติตามคำสั่งที่ SET ใน Entry Mode ข้อกำหนดที่จะรู้ว่าเป็นการเขียนข้อมูลของ CG RAM หรือ DD RAM ทำได้โดยการ SET Address ของ CG RAM หรือ DD RAM ขึ้นมาก่อนจะเขียนข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Read Data From CG หรือ DD RAM

RS R/W DB7 ---------- DB0

1	1	D	D	D	D	D	D	D	D

<- High Low ->

Order Bit Order Bit

เป็นคำสั่งอ่านค่าข้อมูลจาก CG RAM หรือ DD RAM โดยก่อนอ่านค่าจาก DD RAM หรือ CG RAM นี้ควรจะใช้คำสั่ง SET Address ก่อนเพื่อให้รู้ว่าข้อมูลที่อ่านได้นั้นเป็น DD RAM หรือ CG RAM

การต่อ Dot Matrix LCD Module เข้ากับ Micro Controller MCS-51 นั้น จำเป็นต้องผ่าน Port ก่อนโดยทำการจำลองสัญญาณในการส่งข้อมูลที่จะไปควบคุมการทำงานของ Controller ของ LCD Module ใน Dot Matrix LCD Module มีขาสัญญาณในการต่อใช้งานดังนี้

RS (Registor Selcetion) จะเป็นขาเลือกรีจิสเตอร์ภายในตัวไอซี ซึ่งมีอยู่ 2 ตัวด้วยกันคือ Instruction Registor (IR) และ Data Registor (DR) โดยถ้าเป็น "0" เป็นการเลือกรีจิสเตอร์ของคำสั่ง "1" เป็นการเลือกรีจิสเตอร์ของข้อมูล

R/W (Read/Write) เป็นตัวเลือกว่าจะเขียนข้อมูลหรืออ่านข้อมูลจากตัวไอซีโดยถ้าเป็น "1" จะอ่านข้อมูล และ "0" เป็นการเขียนข้อมูลเข้าตัวไอซี

E (Enable Signal) เป็นขากำหนดสถาพการรับการเขียน และอ่านของข้อมูล

DB0-DB7 เป็นขาที่ใช้รับข้อมูล และ คำสั่งจากภายนอก

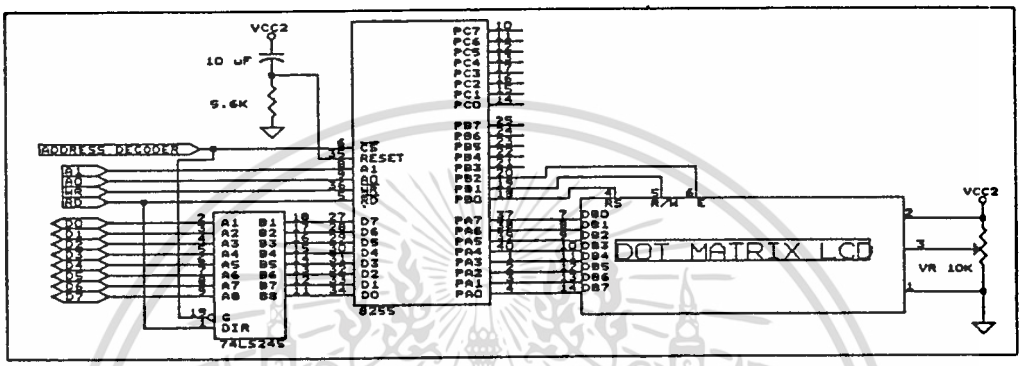
VDD เป็นขาไฟเลี้ยงตัววงจร

VSS เป็นขา GND

VO เป็นขาที่ใช้ปรับแรงดันขับในการแสดงของ LCD สว่างหรือมืด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การต่อจะใช้วงจรตามวงจรรูปที่ 1



วงจรที่ 1 การต่อ Dot Matrix LCD เข้ากับ Micro Controller

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อุปกรณ์

1.MCS-51 Traing Board หรือ JAZZ31

2.แผงช่วยฝึกการทดลอง

3.Dot Matrix LCD Modele DV1601

4.สายต่อวงจร และ แหล่งจ่ายไฟ

การทดลอง

1.ต่อวงจรตามวงจรรูปที่ 1

2.ป้อนโปรแกรมที่ 1 แล้วสั่งให้ทำงานตามโปรแกรมที่ป้อนเข้าไป

โปรแกรมที่ 1

```
                1  ;Program Test Charactor of Dot Matrix LCD
8000            2          ORG 8000h
DE00=           3          porta   equ  0de00h
DE01=           4          portb   equ  0de01h
DE02=           5          portc   equ  0de02h
DE03=           6          conf    equ  0de03h
                7
8000 90DE03     8  Start:  mov dptr,#conf
8003 7488       9          mov a,#10001000b ;Port A Output
                10                          ;Port B Output
                11                          ;Port C Input
8005 F0         12          movx @dptr,a
8006 12808B     13          lcall inti       ;PROGRAM intial display lcd
8009 7430       14          mov a,#'0'
800B 7D00       15  test0:  mov r5,#0
800D 04         16          inc a
800E F8         17  test1:  mov r0,a
800F 128055     18          lcall wrbyte
8012 0D         19          inc r5
8013 BD110A     20          cjne r5,#17,test2
8016 1280C4     21          lcall delay
8019 7E01       22          mov r6,#1
801B 128070     23          lcall code
801E 80EB       24          sjmp test0
8020 BD09EB     25  test2:  cjne r5,#9,test1
8023 7EC0       26          mov r6,#11000000b
8025 128070     27          lcall code
8028 80E4       28          sjmp test1
                29
                30  ;SUB Send Word DATA to Screen Display LCD
                31  ;code 0Ah is code NEW LINE
                32  ;REG dptr = Table
802A 7E01       33  sword:  mov r6,#1
802C 128070     34          lcall code       ;Clear Display LCD
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
802F 7901       35          mov r1,#1
8031 1280C4     36          lcall delay
8034 E0         37          movx a,@dptr
8035 7D00       38          mov r5,#0
8037 F8         39  s1:     mov r0,a
8038 128055     40          lcall wrbyte
803B 0D         41          inc r5
803C A3         42          inc dptr
803D E0         43          movx a,@dptr
803E BD1001     44          cjne r5,#16,s3
8041 22         45          ret
8042 B40A01     46  s3:     cjne a,#0ah,s4   ;if not  0ah then s3
8045 22         47          ret
8046 BD08EE     48  s4:     cjne r5,#8,s1
8049 7EC0       49          mov r6,#11000000b
804B 128070     50          lcall code
804E 7901       51          mov r1,#1
8050 1280C4     52          lcall delay
8053 80E2       53          sjmp s1
                54
                55  ;Send DATA 1 Byte to DD RAM
8055 C082       56  wrbyte: push dpl
8057 C083       57          push dph
8059 C0E0       58          push acc
805B 7401       59          mov a,#1
805D 90DE01     60          mov dptr,#portb
8060 F0         61          movx @dptr,a
8061 E8         62          mov a,r0
8062 90DE00     63          mov dptr,#porta
8065 F0         64          movx @dptr,a
8066 12B0AE     65          lcall pluse
8069 D0E0       66          pop acc
806B D083       67          pop dph
806D D082       68          pop dpl
806F 22         69          ret
                70
                71  ;Sent Code Control (IR)
8070 C083       72  code:   push dph
8072 C082       73          push dpl
8074 C0E0       74          push acc
8076 7400       75          mov a,#0
8078 90DE01     76          mov dptr,#portb
807B F0         77          movx @dptr,a
807C EE         78          mov a,r6
807D 90DE00     79          mov dptr,#porta
8080 F0         80          movx @dptr,a
8081 12B0AE     81          lcall pluse
8084 D0E0       82          pop acc
8086 D082       83          pop dpl
8088 D083       84          pop dph
808A 22         85          ret
                86
                87  ;Intial Dot Matrix LCD Module
808B 7400       88  Inti:   mov a,#0
808D 90DE01     89          mov dptr,#portb
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
8090 F0          90          movx @dptr,a
8091 7E38        91          mov r6,#00111000b;Funtion Set 30h
                 92                          ;DL=1 (8 BIT)
                 93                          ;N=1 (2 LINE)
                 94                          ;F=0 (5*7)
8093 128070      95          lcall code
8096 1280C4      96          lcall delay
8099 7E0D        97          mov r6,#00001101b;Display on/off Control
                 98                          ;D = 1 Display ON
                 99                          ;C = 0 Cursor  OFF
                100                          ;B = 1 Blink
809B 128070      101         lcall code
809E 7E06        102         mov r6,#00000110b;Entry Mode Set
                 103                         ;I/D = 1 Increment
                 104                         ;S = 0   Right
80A0 128070      105         lcall code
80A3 7E01        106         mov r6,#00000001b;Clear All Display
80A5 128070      107         lcall code
80A8 7901        108         mov r1,#1
80AA 1280C4      109         lcall delay
80AD 22          110         ret
                 111
                 112 ;Enable Pluse Sub
80AE             113 pluse:
80AE 90DE01      114         mov  dptr,#portb
80B1 E0          115         movx a,@dptr     ;Read Data PortB
80B2 D2E2        116         setb acc.2       ;Pin E (enable) = 1
80B4 90DE01      117         mov dptr,#portb
80B7 F0          118         movx @dptr,a
80B8 7901        119         mov r1,#1        ;Delay 1 mS
80BA 1280C4      120         lcall delay
80BD C2E2        121         clr acc.2        ;Pin E (enable) = 0
80BF 90DE01      122         mov dptr,#portb
80C2 F0          123         movx @dptr,a
80C3 22          124         ret
                 125
                 126 ;Delay Time Sub
80C4             127 delay:
80C4 7AF0        128 delay1: mov r2,#240      ;Delay 1 mS
80C6 00          129 de:     nop
80C7 00          130         nop
80C8 DAFC        131         djnz r2,de
80CA D9F8        132         djnz r1,delay1
80CC 22          133         ret
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.สังเกตุผลของการทำงานของ LCD

4. ลองเปลี่ยนค่าของ Delay ลอง RUN โปรแกรมสังเกตุผล

5.ให้ลองทำการเปลี่ยนค่าต่างๆในโปรแกรมของส่วน Inti เขียนโปรแกรมที่ 2 เพิ่มเติมเข้าไป และ สั่งให้โปรแกรมที่ 2 ทำงาน

โปรแกรมที่ 2

```
                        1  ;Program Test Charactor of Dot Matrix LCD
8000                    2          ORG 8000h
DE00=                   3          porta   equ  0de00h
DE01=                   4          portb   equ  0de01h
DE02=                   5          portc   equ  0de02h
DE03=                   6          conf    equ  0de03h
                        7
8000 900E03             8  Start:  mov dptr,#conf
8003 7488               9          mov a,#10001000b ;Port A Output
                       10                           ;Port B Output
                       11                           ;Port C Input
8005 F0                12          movx @dptr,a
8006 1280A7            13          lcall inti       ;PROGRAM intial display lcd
8009 908021            14          mov dptr,#table1
800C 128043            15          lcall sword
800F 7964              16          mov r1,#100
8011 1280E0            17          lcall delay
8014 908032            18          mov dptr,#table2
8017 128043            19          lcall sword
801A 7964              20          mov r1,#100
801C 1280E0            21          lcall delay
801F 80FE              22          sjmp $
                       23
8021 54657374          24  table1: db "Test Program LCD",0ah
8025 2050726F
8029 67726160
802D 204C4344
8031 0A
8032 5072696E          25  table2: db "Print Word      ",0ah
8036 7420576F
803A 72642020
803E 20202020
8042 0A
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คำถาม

1.จงอธิบายการทำงานของโปรแกรมที่ 1

2.ให้เขียนโปรแกรมในการส่งข้อมูลออกจอ LCD ออกเป็น Word

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทดลองที่ 4
PRINTER PORT

วัตถุประสงค์

1.เป็นการศึกษาการทำงานของ Printer

2.สามารถส่งข้อมูลออกไปแสดงผลที่ Printer ได้

ทฤษฎี

Printer เป็นอุปกรณ์ที่ใช้แสดงผลสำคัญตัวหนึ่ง ใช้เป็นพิมพ์ข่าวสารข้อมูลที่เราต้องการ โดยแบ่งตามลักษณะการใช้งานได้ 3 แบบ ด้วยกันคือ

1. แบบเดซี่วีล (Daisy Wheel) ก็คือเครื่อพิมพ์แบบอีเล็คทรอนิคส์ซึ่งจะมีตัวอักษรที่จะพิมพ์ เป็น แบบตายตัว ส่วนมากจะใช้ในการพิมพ์ตัวอักษรโดยเฉพาะ

ข้อดีของเครื่องพิมพ์ชนิดนี้จะให้ตัวอักษรที่คมชัด

ข้อเสีย ไม่สามารถพิมพ์รูปได้ และ มีตัวอักษรตายตัว

2. แบบดอทแมททริกซ์ (Dot Matrix) เหมาะสำหรับพิมพ์ข้อมูลทั้งในรูปอักษร หรือ รูปกราฟฟิก ต่าง ๆ การพิมพ์ใช้เป็นชุดของเข็มยิง ซึ่งเข็มแต่ละตัวสามารถจะควบคุมโดย Computer อย่างอิสระ แบ่งได้เป็น 2 แบบด้วยกันคือ

แบบใช้ความร้อน (Thermal) เป็นการใช้ความร้อนในการทำให้เกิดจุดที่กระดาษ ซึ่งเป็นกระดาษแบบ พิเศษ เมื่อถูกความร้อนจะเปลี่ยนสีที่ถูกความร้อนนั้นเท่านั้น ส่วนมากจะใช้ในเครื่องพิมพ์ของเครื่องคำนวน, เครื่อง Computer แบบพก เป็นต้น

ข้อดีของเครื่องพิมพ์นี้คือมีขนาดเล็กและเบา

ข้อเสีย คือต้องใช้กระดาษเฉพาะเท่านั้นจึงจะพิมพ์ได้

แบบใช้การตอกเข็ม (Impact) เป็นการใช้เข็มยิงไปผ้าหมึก เพื่อให้ผ้าหมึกส่วนที่ถูกยิงนั้นไปกระทบที่กระดาษที่รองรับอยู่ทำให้เกิดจุดที่กระดาษ เครื่องพิมพ์ชนิดนี้ส่วนมากจะใช้งานอยู่กับที่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ข้อดี ใช้งานได้ง่ายจะใช้ในงานพิมพ์ข้อมูล หรือ รูปกราฟฟิก ซะส่วนมาก
ข้อเสีย มีน้ำหนักมากกว่าเครื่องพิมพ์ชนิดความร้อน

การพิมพ์แบบ Dot Matrix ทั้งสองแบบจะมีข้อเสียคล้ายกันคือ ตัวอักษรที่พิมพ์ หรือรูปที่พิมพ์ออกมานั้นจะไม่คมชัด เพราะการพิมพ์แบบนี้จะอาศัยจุดแต่ละจุดต่อกันทำให้เป็นตัวอักษร หรือ รูปภาพ ขึ้นมา

3. แบบใช้การพลอท (Plotter) เครื่องพิมพ์แบบนี้จะอาศัยการลากเส้นของ หมึกพิมพ์ โดยมีแกนอ้างอิงเป็น แกน X และ แกน Y เครื่องพิมพ์ชนิดนี้เหมาะกับงานในการออกแบบ ไม่เหมาะกับงานพิมพ์รายงาน

ข้อดี เหมาะแก่การออกแบบต่าง ๆ เพราะมีความละเอียดมาก
ข้อเสียของเครื่องพิมพ์ชนิดนี้คือมีความเร็วในการพิมพ์ต่ำกว่าการพิมพ์ทั้งสองแบบที่กล่าวมาข้างต้น และ ราคาสูง

ในการทดลองนี้จะเป็นการทดลองเครื่องพิมพ์แบบ Dot Matrix ซึ่งนิยมใช้ในปัจจุบัน มีราคาถูก และ สามารถควบคุมการทำงานได้ง่าย

ส่วนประกอบของเครื่องพิมพ์แบบ Dot Matrix จะประกอบด้วยส่วนใหญ่ ๆ ด้วยกันคือ

1.หัวพิมพ์ (Print Head)

คือส่วนที่ทำให้เกิดตัวอักษรบนกระดาษ โดยมีลักษณะต่าง ๆ อาจเป็นก้านพิมพ์เหมือนเครื่องพิมพ์ดีด หรือ เป็นลวดสายไฟเส้นเล็ก ๆ โปรแกรมด้วยระบบอิเล็คทรอนิคส์

2.กลไกการพิมพ์ (Printing Mechamism)

ทำหน้าที่ควบคุมการทำงานของหัวพิมพ์กลไกบางแบบเป็นแบบ 2 ทิศทาง (Bidirectional) คือ สามารถพิมพ์จากซ้ายไปขวาและขวามาซ้ายได้พร้อมกัน

3.ล้อพิมพ์ (Platen)

คือ ล้อหมุนเหมือนกับที่ในเครื่องพิมพ์ดีดเป็นที่หัวพิมพ์เคาะบนกระดาษกับตัวนี้

4.กลไกป้อนกระดาษ (Paper-Feed Mechamism)

ทำหน้าที่เคลื่อนกระดาษไปข้างหน้าขณะที่กำลังพิมพ์ แบ่งเป็น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การป้อนกระดาษโดยอาศัยความฝืด (Friction Feed) เช่น ล้อพิมพ์

การป้อนกระดาษคล้ายแทรดเตอร์ (Tractor Feed) กระดาษเป็นแบบเจาะรูด้านข้าง ๆ โดยตลอดแล้วเครื่องพิมพ์ก็ดึงไปด้วยหนามเตยหรือสปรอคเกต (Sprocket)

การป้อนด้วยเป็ก (Peg Link Pin) เล็กๆ ซึ่งติดบนล้อพิมพ์และดึงกระดาษไปข้างหน้า วิธีนี้ทำให้กระดาษวางตัวแน่นอนกว่า

การอินเตอร์เฟสกับเครื่องพิมพ์
มี 2 แบบคือ

แบบอนุกรม (Serial)
เป็นการส่งข้อมูลไปทีละ Bit ข้อมูลโดยข้อมูล 1 ตัว 8 Bit ต้องส่งข้อมูลออกไป 10 Bit ด้วยกัน ข้อมูล 8 Bit จุดเริ่มต้นของข้อมูล (Start Bit) และ จุดสิ้นสุดของข้อมูล (Stop Bit) การส่งจากข้อมูลแบบอนุกรมนี้ส่วนมากจะผ่าน RS-232 ซึ่งเป็นการติดต่อข้อมูลมาตรฐานแบบอนุกรม

แบบขนาน (Parallel)
เป็นการส่งข้อมูลออกไปครั้งละ 8 Bit โดยมีสัญญาณ สโตรบ (Strob) เป็นสัญญาณในการส่งข้อมูล

ชุดตัวอักษร
เครื่องพิมพ์ส่วนมากใช้ชุดตัวอักษรตามมาตรฐาน ASCII จะประกอบด้วยอักษร 96 ตัว คือ ตัวอักษร ตัวเลข และ สัญลักษณ์ต่างๆ เป็นมาตรฐานสำหรับ Printer ทั่วไป

ทางส่วนควบคุม
Printer ก็เหมือน ไมโคร ตัวหนึ่งซึ่งภายในตัว Printer จะมีตัว CPU , ROM , RAM และ Port เครื่องจะถูกทำงานโดยโปรแกรมที่อยู่ใน ROM และนอกจาก ROM ที่ใช้เก็บโปรแกรมในการควบคุมการทำงานของ Printer ยังมีหน้าที่ในเก็บตัวอักษรในการพิมพ์ Character Pattern มาตรฐาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สัญญาณที่ใช้ในการติดต่อกับ Printer

STROBE (Input)
เป็นสัญญาณที่ส่งจากอุปกรณ์ภายนอก เพื่อบอกให้ Printer รับข้อมูล

ACKNLG (Output)
เป็นสัญญาณที่ส่งจาก Printer เพื่อบอกให้อุปกรณ์ภายสนอกรับรู้ว่าได้รับข้อมูลเรียบร้อยแล้ว

BUSY (Output)
เป็นสัญญาณที่ส่งจาก Printer เพื่อบอกให้อุปกรณ์ภายนอกรู้ว่า เครื่องพิมพ์ยังไม่พร้อมที่จะรับข้อมูล ซึ่งอาจจะมาจากสาเหตุ

- เครื่องพิมพ์อยู่ในระหว่างกระทำข้อมูลที่รับเข้ามา
- อยู่ในระหว่างพิมพ์ข้อมูล
- อยู่ในระหว่าง OFF-Line State
- ข้อมูลที่ส่งเข้ามาไม่สามารถตีความหมายได้

PE (Output)
เป็นสัญญาณที่ส่งจาก Printer เพื่อบอกให้อุปกรณ์ภายสนอกรับรู้ว่ากระดาษพิมพ์หมด

SELECT (Output)
เป็นสัญญาณที่ส่งจาก Printer เพื่อบอกให้อุปกรณ์ภายสนอกรับรู้ว่าเครื่องพิมพ์อยู่ในสถานะใด ถ้าเป็น "1" จะใช้งานอยู่ ถ้าเป็น "0" จะอยู่ในสถานะที่ไม่พร้อมที่จะใช้งาน หรือ ไม่สามารถรับข้อมูลได้โดยสถานะทั้ง 2 นี้เกิดจาก

1) สถานะที่ใช้งานได้ (Select State)
- มีการกดปุ่ม Select หรือ ON Line ในขณะที่เครื่องพิมพ์อยู่ในสถานะ Deselect หรือ OFF LINE

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2) สถานะที่ไม่ใช้งาน (Deselect State)

- มีการกดปุ่ม Select หรือ ON LINE ในขณะที่อยู่ในสถานะ Select หรือ ON LINE
- ได้รับรหัส DC3 (13H)
- เมื่อกระดาษพิมพ์หมด
- เมื่อเครื่องพิมพ์อยู่ในสถานะ FAULT

AUTO FEEDXT (Input)

เมื่อ Printer ได้รับสัญญาณนี้ ก็จะเลื่อนบรรทัดใหม่

INIT (Input)

เมื่อ Printer ได้รับสัญญาณนี้เครื่องพิมพ์จะเลื่อนหัวพิมพ์สู่จุดเริ่มต้นพิมพ์ ข้อมูลต่าง ๆ จะถูกละทิ้งหมด โดยทั่วไปจะมีความกว้างของ Pluse ประมาณ 50 uS

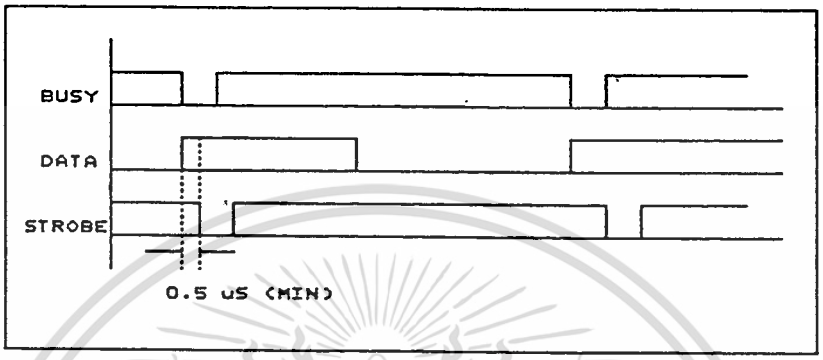
ERROR (Output)

เป็นสัญญาณที่ส่งจาก Printer เพื่อบอกให้อุปกรณ์ภายสนอกรับรู้ว่ามีข้อผิดผลาดเกิดขึ้นในการพิมพ์ เช่น

- กระดาษหมด
- อยู่ระหว่าง OFF LINE
- ข้อมูลที่รับเข้ามาผิดพลาดไม่สามารถตีความได้

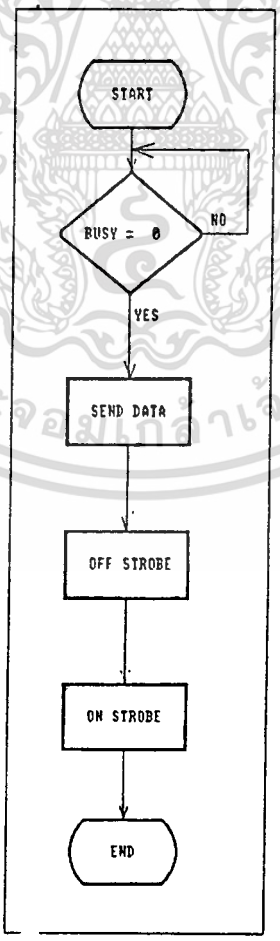
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในการติดต่อระหว่าง Micro Controller สามารถติดต่อกับ Printer โดยใช้สัญญาณของ Printer ในการติดต่อ Data (8 LINE), STORBE และ BUSY ซึ่งสัญญาณในการติดต่อจะแสดงใน Timing Diagram ต่อไปนี้



Timing Diagram

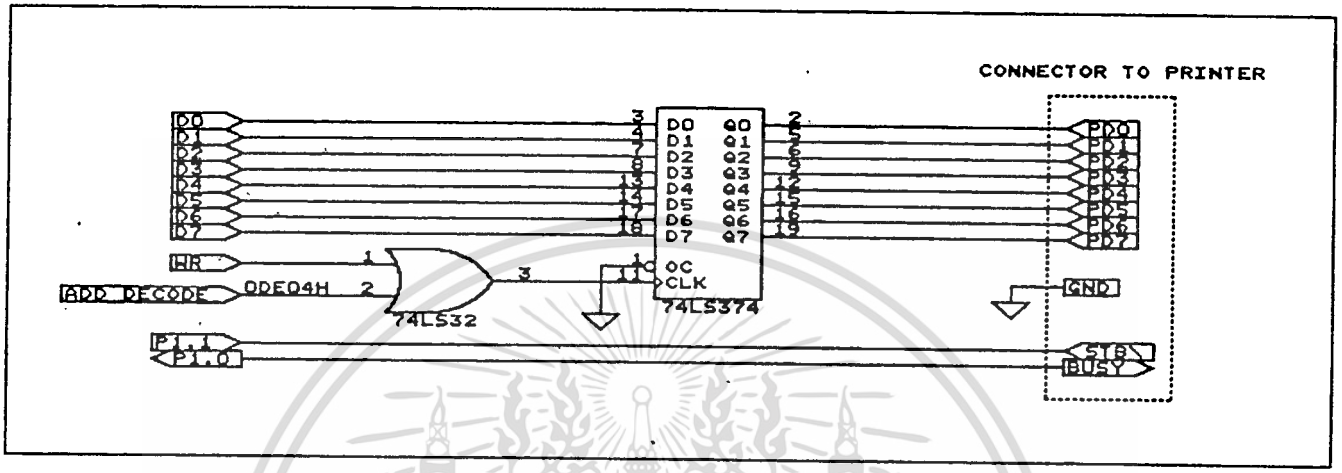
จาก Timing Diagram สามารถนำมาเขียนเป็น Flow Chart



Flow Chart

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

วงจรที่จะใช้การติดต่อกันระหว่าง Micro Controller MCS-51 ซึ่งจะแสดงในรูปที่ 1



รูปที่ 1 วงจรการติดต่อระหว่าง Printer กับ MCS-51

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อุปกรณ์

1.MCS-51 Traing Board หรือ JAZZ31 1

2.แผงช่วยฝึกการทดลอง 1

3.สายในการ Printer 1

4.Printer 1

5.74LS374 1

6.สายต่อวงจร และ แหล่งจ่ายไฟ

การทดลอง

1. ต่อวงจรตามรูปที่ 1
2. ให้ ป้อนโปรแกรมที่ 1 เข้าเครื่อง

โปรแกรมที่ 1

```
                1  ;Program Printer
8000            2          ORG   8000h
DE04=           3          portp   equ 0de04h
                4
8000 908024     5  start:  mov dptr,#table
8003 128039     6          lcall pword      ;Send Data to Printer
8006 740A       7          mov a,#0ah       ;LINE FEED
8008 12804D     8          lcall pbyte
800B 7420       9          mov a,#' '       ;A = ASCII (20h)
800D C0E0      10  st1:    push acc
800F 12804D    11          lcall pbyte
8012 D0E0      12          pop acc
8014 04        13          inc a            ;ASCII = ASCII + 1
8015 B47DF5    14          cjne a,#'}',st1 ;A = '}'
8018 740A      15          mov a,#0ah       ;LINE FEED
801A 12804D    16          lcall pbyte
801D 740D      17          mov a,#0dh
801F 12804D    18          lcall pbyte
8022 80FE      19          sjmp $
               20
8024 54657374  21  table:  db "Test Program Printer",0ah
8028 2050726F
802C 6772616D
8030 20507269
8034 6E746572
8038 0A
               22
               23  ;Send Data Word to Printer
               24  ;0ah = END Word
8039 E0        25  pword:  movx a,@dptr
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
803A 12804D     26          lcall pbyte      ;Send Data (ASCII Code) to Printer
803D A3         27          inc dptr         ;Table + 1
803E E0         28          movx a,@dptr
803F B40AF7     29          cjne a,#0ah,pword;If A = 0Ah End Charecter,Line Feed
8042 740A       30          mov a,#0ah
8044 12804D     31          lcall pbyte      ;Line Feed
8047 740D       32          mov a,#0dh
8049 12804D     33          lcall pbyte      ;Carrier Return
804C 22         34          ret
                35
                36
                37
                38  ;Send Data 1 Byte to Printer
804D C082       39  pbyte:  push dpl
804F C083       40          push dph
8051 128065     41          lcall busy       ;Check BUSY
8054 90DE04     42          mov dptr,#portp
8057 F0         43          movx @dptr,a     ;Send Data to Printer
8058 00         44          nop              ;delay 1 uS
8059 C291       45          clr p1.1         ;Strobe = "0"
805B 128069     46          lcall delay
805E D291       47          setb p1.1        ;Strobe = "1"
8060 D083       48          pop dph
8062 D082       49          pop dpl
8064 22         50          ret
                51
8065 2090FD     52  busy:   jb p1.0,busy
8068 22         53          ret
                54
8069 786E       55  delay:  mov r0,#110      ;delay 0.5 mS
806B D8FE       56          djnz r0,$
806D 22         57          ret
                58
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. ต่อวงจรเข้ากับ Printer

4. สั่งโปรแกรมให้ทำงาน

5. สังเกตุการแสดงผลที่ Printer

6. ให้ทดลองตัดส่วนของ โปรแกรมในการ Delay ออกโดยใส่ 00h เข้าไปแทน สั่งให้โปรแกรมทำงานสังเกตุดูผลการของการทำงานของโปรแกรม

7. ป้อนโปรแกรมที่ 2 เข้าเครื่อง แล้วให้ทำตามข้อ 4,5

โปรแกรมที่ 2

```
                    1  ;Program Printer
8000                2          ORG   8000h
DE04=               3          portp   equ 0de04h
                    4
8000 908019         5  start:  mov dptr,#table
8003 12803A         6          lcall pword     ;Send Data to Printer
8006 90802F         7          mov dptr,#table2
8009 12803A         8          lcall pword     ;Send Data to Printer
800C 908021         9          mov dptr,#table1
800F 12803A        10          lcall pword     ;Send Data to Printer
8012 740A          11  st:     mov a,#0ah      ;LINE FEED
8014 12804E        12          lcall pbyte
8017 80F9          13          sjmp st
                   14
8019 31323334      15  table:  db "1234567890",0ah
801D 35363738
8021 39300A
8024 30393837      16  table1: db "0987665431",0ah
8028 36363534
802C 33310A
802F 30303030      17  table2: db "0000000000",0ah
8033 30303030
8037 30300A
                   18
                   19  ;Send Data Word to Printer
                   20  ;0ah = END Word
803A E0            21  pword:  movx a,@dptr
803B 12804E        22          lcall pbyte     ;Send Data (ASCII Code) to Printer
803E A3            23          inc dptr        ;Table + 1
803F E0            24          movx a,@dptr
8040 B40AF7        25          cjne a,#0ah,pword;If A = 0Ah End Charecter,Line Feed
8043 740A          26          mov a,#0ah
8045 12804E        27          lcall pbyte     ;Line Feed
8048 740D          28          mov a,#0dh
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
804A 12804E     29          lcall pbyte      ;Carrier Return
804D 22         30          ret
                31
                32
                33
                34  ;Send Data 1 Byte to Printer
804E C082       35  pbyte:  push dpl
8050 C083       36          push dph
8052 128066     37          lcall busy       ;Check BUSY
8055 90DE04     38          mov dptr,#portp
8058 F0         39          movx @dptr,a     ;Send Data to Printer
8059 00         40          nop              ;delay 1 uS
805A C291       41          clr p1.1         ;Strobe = '0'
805C 12806A     42          lcall delay
805F D291       43          setb p1.1        ;Strobe = '1'
8061 D083       44          pop dph
8063 D082       45          pop dpl
8065 22         46          ret
                47
8066 2090FD     48  busy:   jb p1.0,busy
8069 22         49          ret
                50
806A 786E       51  delay:  mov r0,#110      ;delay 0.5 mS
806C D8FE       52          djnz r0,$
806E 22         53          ret
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คำถาม

1. ถ้าไม่มีการเช็ค BUSY เครื่องพิมพ์จะทำงานได้หรือไม่ เพราะเหตุใด

2. ให้เขียนโปรแกรมในการส่งข้อมูลออก Printer โดยใช้โปรแกรมย่อยใน โปรแกรมที่ 1 ส่งข้อมูลต่อไปนี้

```
**********************************
*    Program Example Printer     *
**********************************
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทดลองที่ 5

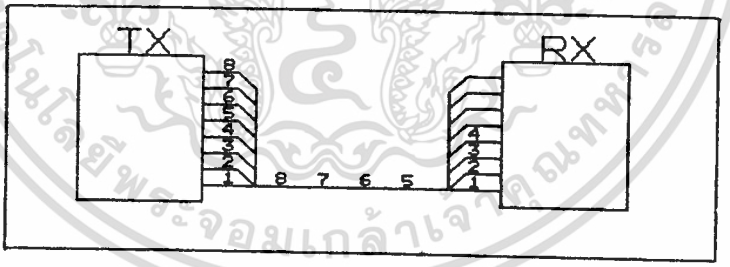
SERIAL PORT RS-232

วัตถุประสงค์

1. เพื่อเข้าใจในระบบการติดต่อสื่อสารมาตรฐาน RS-232
2. สามารถสื่อสารข้อมูลระหว่าง MCS-51 กับ PC
3. ในเอาการสื่อสารชนิดนี้ไปใช้งานในการควบคุมอุปกรณ์ที่อยู่ระยะไกล

ทฤษฎี

การสื่อสารข้อมูลแบบ อนุกรม เป็นการส่ง หรือ รับข้อมูลครั้งละ 1 บิท การส่งข้อมูล หรือ รับข้อมูลแบบอนุกรมจะทำโดยการนำข้อมูลที่เป็นแบบขนานมาทำการเลื่อน (Shift) ข้อมูลไปทางซ้าย หรือ ขวา รูปที่ 1 จะแสดงการรับส่งข้อมูลแบบอนุกรม



รูปที่ 1

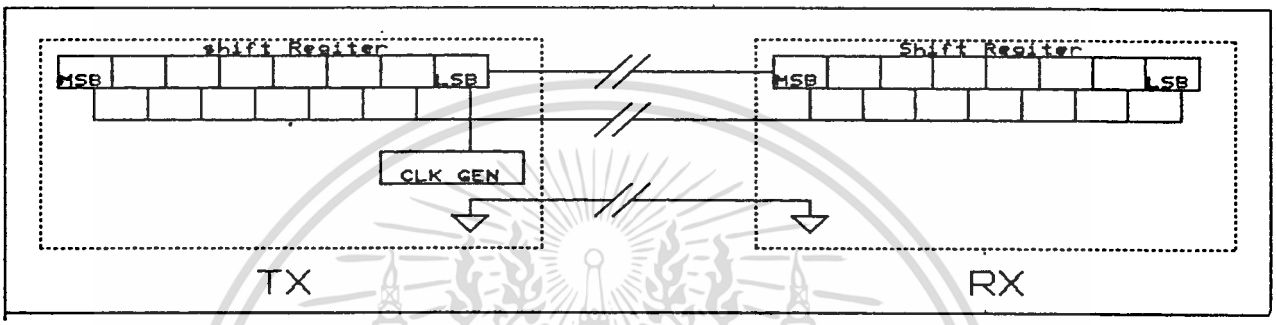
จากรูปที่ 1 แสดงให้เห็นการส่งข้อมูลแบบอนุกรม ข้อมูลจากจุดส่งจะถูกเปลี่ยนให้เป็นแบบ อนุกรมซะก่อนที่จะถูกส่งออกไป จะทยอยส่งออกทีละ 1 บิท จนครบทุกบิทที่จะทำการส่งถือว่าจบการส่งข้อมูลแบบ อนุกรม ส่วนของการรับข้อมูลแบบอนุกรมนั้นจะมีกลไกในการรับข้อมูลแบบอนุกรม โดยการรับนั้นจะถูกเลื่อนข้อมูลเข้ามาทีละ 1 บิทจนครบทุกบิทที่ถูกส่งมาถือว่าจบการรับข้อมูลแบบ อนุกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การส่ง และ รับข้อมูลแบบอนุกรม นั้นมีอยู่ด้วยกัน 2 วิธีคือ

แบบ ซิงโครนัส (Synchronous)

เป็นการส่งข้อมูลแบบอนุกรม ซึ่งจะอาศัยสัญญาณความถี่ที่จุดส่งมาทำการเลื่อนข้อมูลที่ส่งออกมาดังจะแสดงดังรูปที่ 2



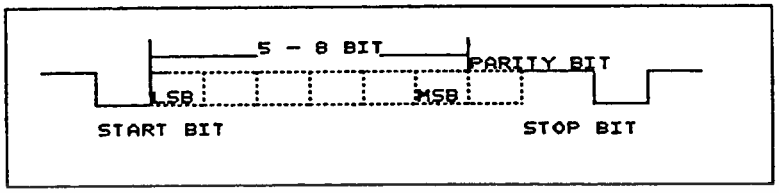
รูปที่ 2 การส่งข้อมูลอนุกรมแบบ Synchronous

จากรูปที่ 2 จะเห็นการส่งข้อมูลอนุกรม แบบ ซิงโครนัส (Synchronous) นั้นจะต้องใช้สายสัญญาณในการติดต่อ 3 สายด้วยกัน คือ สายของข้อมูล,สาย Clock และ สาย GND ซึ่งในการส่ง หรือ รับข้อมูลแบบนี้จะถูกควบคุมการส่ง และ การรับ โดยสัญญาณ Clock

แบบ อะซิงโครนัส (ASynchronous)

เป็นการส่งข้อมูลที่ไม่ต้องใช้สัญญาณ Clock ในการควบคุมในการส่ง หรือรับข้อมูลแบบ อนุกรมเพียงแต่ต้องเพิ่มข้อมูลเข้าไปอีก 2 - 3 บิท โดยที่บิทที่ถูกเพิ่มมานั้นจะเป็นสัญญาณการควบคุมการส่ง และรับข้อมูลแบบ อนุกรม ซึ่งบิทที่เพิ่มเข้ามานั้นได้แก่ Start Bit,Stop Bit และ Parity Bit ความหมายของสัญญาณดังกล่าวนั้นจะเป็นตัวกำหนดจุดเริ่มต้นของข้อมูล (Start Bit) กำหนดจุดจบของข้อมูล (Stop Bit) และ ตรวจสอบข้อมูล (Parity Bit) จะแสดงในรูปที่ 3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3 การส่ง และ รับข้อมูลแบบ อนุกรม แบบ ASynChronous

จากรูปที่ 3 เป็นการส่งข้อมูลขนาด 8 บิท โดยจะมีบิทที่เพิ่มเติมเพื่อเป็นตัวในการ ซิงกันระหว่าง ตัวส่ง และ ตัวรับ

BAUD RATE คือ อัตราของการส่งข้อมูลแบบอนุกรม เทียบกับเวลา 1 วินาที เช่น ส่งข้อมูลแบบอนุกรม BAUD RATE 9600 หมายถึงการส่งข้อมูลทั้งหมด 9600 Bit ต่อ 1 วินาที ซึ่งในการส่งข้อมูลแบบอนุกรมจะส่ง BAUD RATE ได้ 110,1200,2400,4800 และ 9600 เป็นต้น

การรับ และส่งข้อมูลแบบอนุกรมยังแบบตามลักษณะการส่งและรับข้อมูลได้ 3 แบบด้วยกันคือ

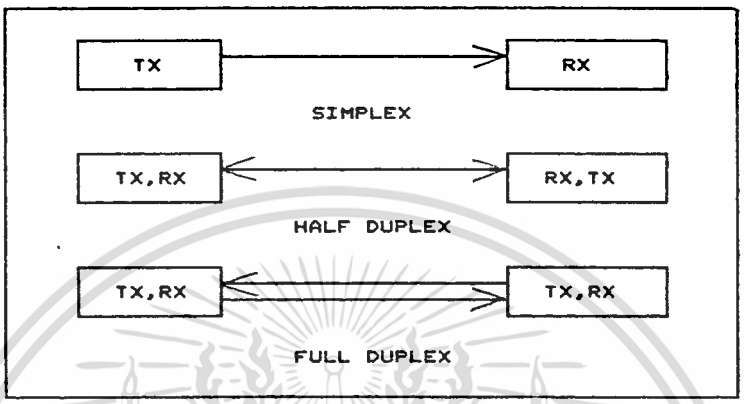
1). แบบซิมเพลกซ์ (Simplex) เป็นการส่งข้อมูลแบบทางเดียวเท่านั้น บางครั้งก็เรียกการส่งแบบนี้ว่าการส่งทิศทางเดียว

2) แบบ ฮาล์ฟดูเพลกซ์ (Half Duplex) เป็นการส่งและรับข้อมูลได้ทั้ง 2 สถานี ในสายส่ง และรับข้อมูลในเส้นทางเดียว ซึ่งการรับ และ ส่งของข้อมูลนั้นจะต้องสลับกัน คือ ถ้าด้านหนึ่งส่ง อีกด้านหนึ่งต้องเป็นตัวรับ

3) แบบฟลูดูเพลกซ์ (Full Duplex) เป็นการส่ง และ รับข้อมูลในเวลาเดียวกัน ซึ่งจำเป็นในการใช้สายของข้อมูล 2 สายในการส่งและรับข้อมูลแบบนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในการรับส่งข้อมูลแบบต่างๆ จะแสดงในรูปที่ 4



รูปที่ 4 การส่ง และ รับข้อมูลที่แบบต่าง ๆ

ในตัว Micro Controller MCS-51 มีพอร์ทอนุกรมให้ใช้ เป็นแบบ Full Duplex คือ สามารถส่งและรับข้อมูลแบบ อนุกรมในเวลาเดียวกัน ซึ่งในพอร์ทอนุกรมของ Micro Controller MCS-51 มีรีจิสเตอร์อยู่ด้วย 2 ตัว ที่ใช้ในการติดต่อข้อมูลแบบอนุกรมคือ

SBUF เป็นรีจิสเตอร์ที่ใช้เป็นตัวเก็บข้อมูลที่จะทำการส่ง หรือ รับข้อมูลแบบอนุกรม ซึ่งภายในตัว MCS-51 นี้จะแยกส่วนข้อ SBUF เป็นส่วนของด้านรับและด้านส่ง ออกจากกัน ข้อควรระวังในการใช้การอ่านข้อมูลจาก SBUF นี้จะต้องอ่านก่อนที่ข้อมูล ถัดไปจะถูกส่งเพราะถ้าไม่ทำการอ่านข้อมูลที่ เข้ามาก่อนนั้นจะทำให้ข้อมูลที่ส่งต่อมาจะทับข้อมูลตัวที่เข้ามาก่อนนั้นทำให้ข้อมูลนั้น สูญหายไป

SCON เป็นรีจิสเตอร์ที่ใช้ในการกำหนดควบคุม การทำงานของการส่งข้อมูลแบบ อนุกรม ซึ่งความหมายของแต่ละ บิทใน SCON มีดังต่อไปนี้

SM0	SM1	SM2	REN	TB8	RB8	TI	RI

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

SMO,SM1 เป็นบิทที่จะใช้ในการเลือกโหมดของการทำงานของ การรับและ ส่งข้อมูลแบบอนุกรม ดังนี้

SMO	SM1	MODE	ลักษณะการทำงาน	อัตราของ BAUD RATE
0	0	0	ซิงโครนัส	Fosc/12
0	1	1	8 บิท อะซิงโครนัส	เปลี่ยนไปตามตัวจับเวลา
1	0	2	9 บิท อะซิงโครนัส	Fosc/64 fosc/32
1	1	3	9 บิท อะซิงโครนัส	แปรผัน

REN เป็นตัวอีนาเบิ้ลการรับข้อมูลแบบอนุกรม
1 = อีนาเบิ้ลการรับข้อมูลแบบอนุกรม
0 = ดิสเอเบิ้ลการรับข้อมูลแบบอนุกรม

TR8 เป็นข้อมูลบิทที่ 9 ซึ่งจะถูกส่งในโหมด 2 และ 3 ซึ่งจะให้เป็น "1" หรือ "0" สามารถเปลี่ยนด้วยโปรแกรม

RB8 เป็นข้อมูลบิทที่ 9 ที่ถูกรับเข้ามา ในโหมด 2 และ 3
โหมด 0 จะไม่ใช้
โหมด 1 ถ้า SM2 = 0 จะกลายเป็น STOP BIT ที่ถูกรับไป

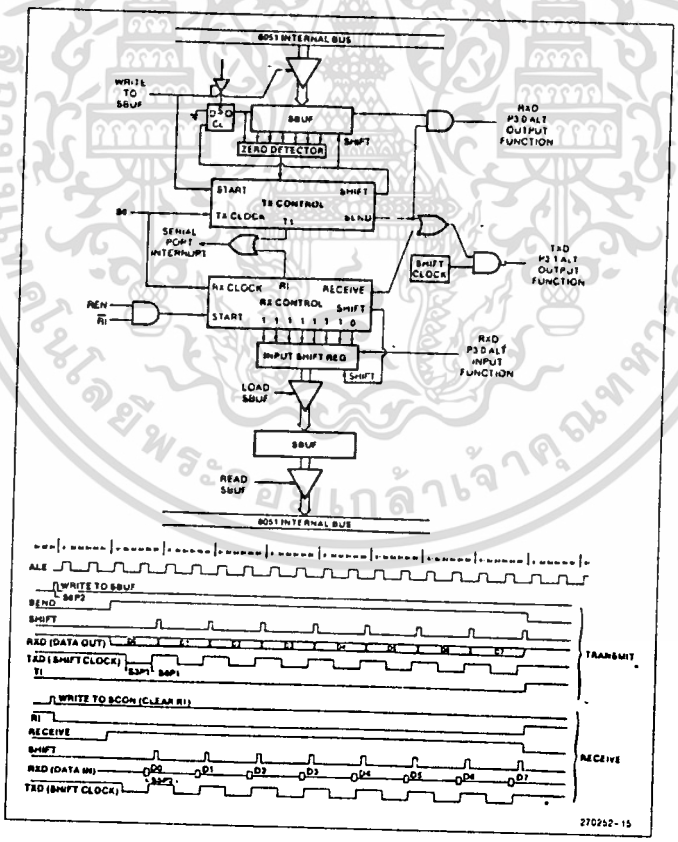
TI เป็นแฟลกอินเตอร์รัพต์ในการส่ง จะ SET ด้วยฮาร์แวร์ คือสัญญาณที่ปลายช่วงเวลาของบิทที่ 8 ในโหมด 0 หรือจุดเริ่มต้นของ STOR BIT ในโหมดอื่น ในการส่งข้อมูลแบบอนุกรมของทุกโหมดจะต้องทำการ เคลียร์ บิทนี้ด้วยโปรแกรมหลังจากการส่ง

RI เป็นแฟลกอินเตอร์รัพท์ในการรับ จะ SET ด้วยฮาร์แวร์ คือสัญญาณที่ปลายช่วงเวลาของบิทที่ 8 ในโหมด 0 หรือจุดครึ่งทางของ STOR BIT ในโหมดอื่น ในการรับข้อมูลแบบอนุกรมของทุกโหมดจะต้องทำการ เคลียร์ บิทนี้ด้วยโปรแกรมหลังจากการรับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในพอร์ทอนุกรมของ MSC-51 นั้นสามารถกำหนด การทำงาน ของการรับ และส่งข้อมูลแบบ อนุกรมได้ 4 Mode ด้วยกัน

MODE 0 เป็นการรับส่งข้อมูลแบบ ซิงโครนัส ซึ่งจะรับ และ ส่งข้อมูลขนาด 8 บิท โดยผ่านขา RXD และ ใช้สัญญาณ TXD เป็นสัญญาณในการเลื่อนข้อมูลในการรับ โดยจะทำการเลื่อน บิท LSB (D0) เป็นตัวแรก อัตรา BUAD RATE คงที่ 1/12 ของความถี่ที่ป้อนให้กับตัว Micro Controller MCS-51 การทำงานของการส่งและรับข้อมูล ในโหมด 1 จะแสดงในรูปที่ 5

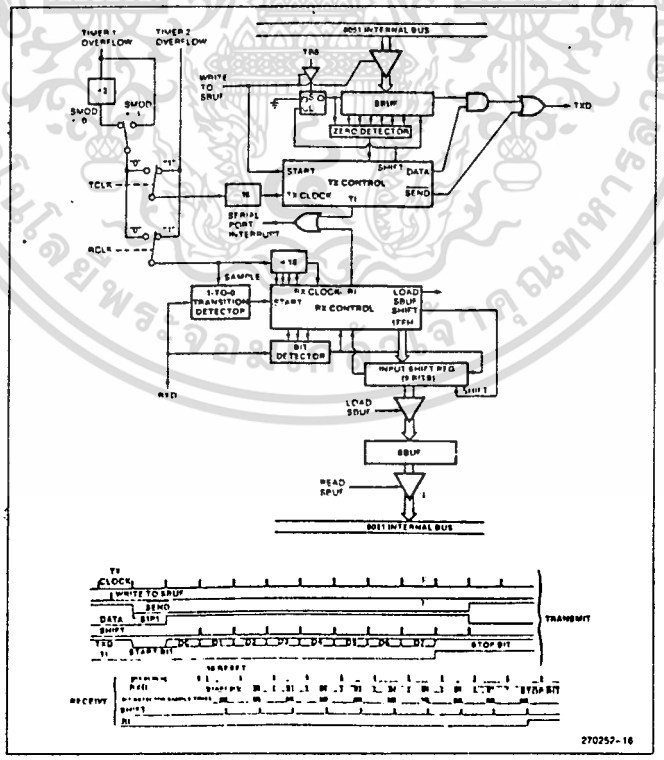


รูปที่ 5

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 5 จะเห็นว่าการรับและการส่งข้อมูลแบบอนุกรมในโหมดนี้จะใช้ขา RXD เป็นตัวรับข้อมูล และ ส่งข้อมูล ส่วน TXD จะใช้เป็น Clock ในการควบคุมการรับส่ง

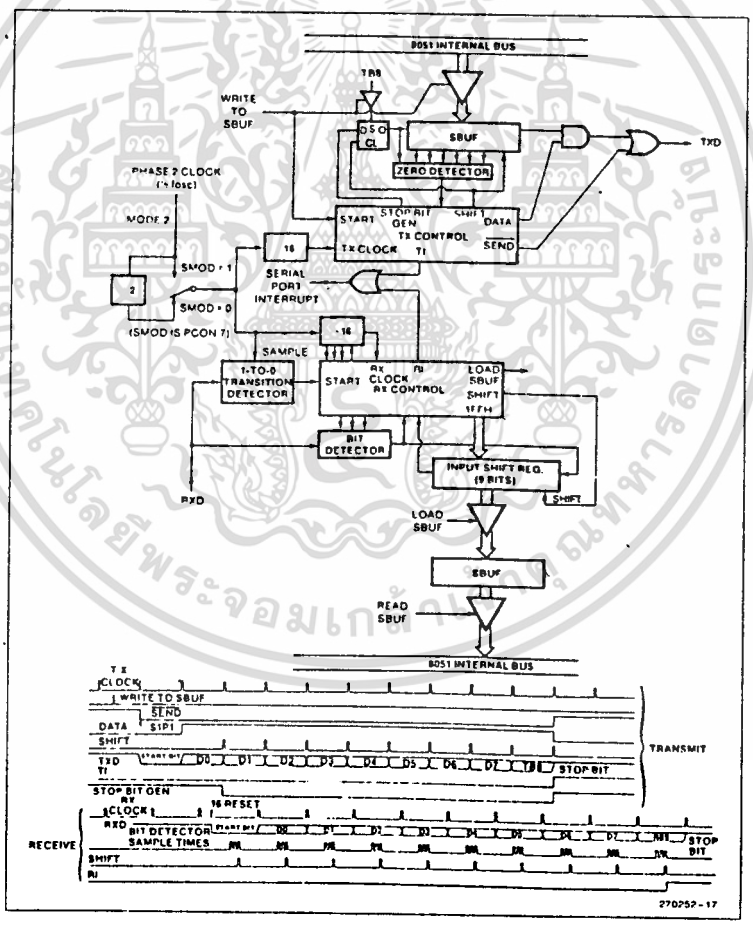
MODE 1 เป็นการรับ และ ส่งข้อมูลแบบ อะซิงโครนัส ซึ่งจะรับและส่งข้อมูลขนาด 8 บิท โดยจะมี 2 บิทเพิ่มเติมคือ Start Bit และ Stop Bit ซึ่งรวมกับบิทของข้อมูลก็ได้ทั้งหมด 10 บิท การับ Stop Bit จะถูกส่งเข้าไปยัง RB2 ของ SCON การตั้ง BAUD RATE จะขึ้นอยู่กับการตั้งตัวจับเวลาที่ 1 หรือ 2 อาจใช้สัญญาณ Clock การทำงานของการรับ และ ส่งข้อมูลแบบอนุกรมในโหมด 1 จะแสดงในรูปที่ 6



รูปที่ 6

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

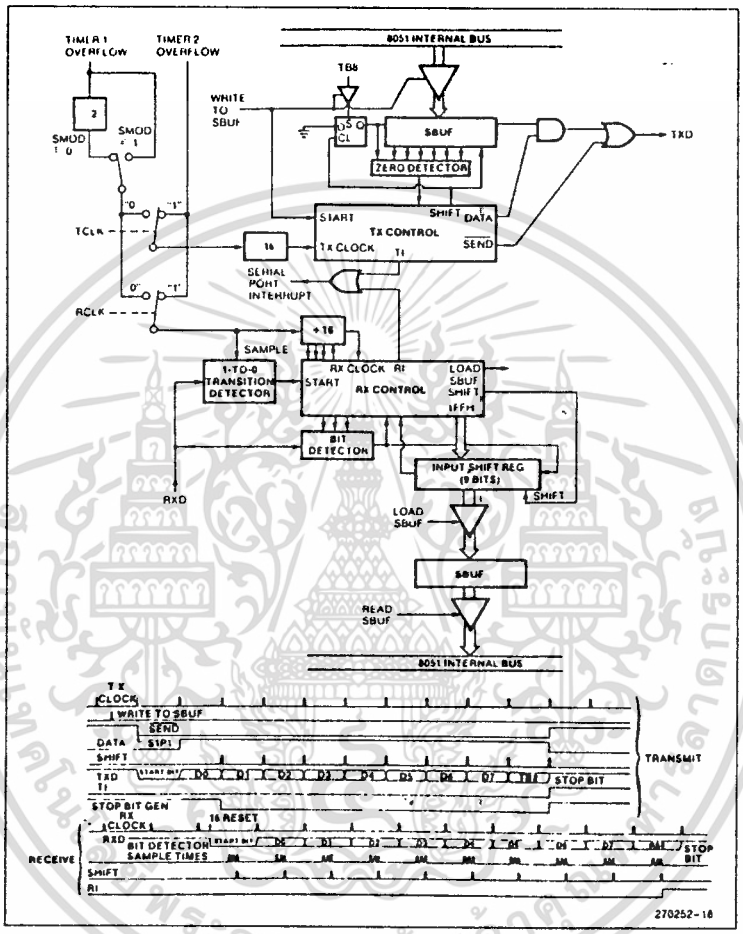
MODE 2 เป็นการรับ และ ส่งข้อมูลแบบ อะซิงโครนัส ซึ่งจะรับและส่งข้อมูลขนาด 8 บิท โดยจะมี 3 บิทเพิ่มเติมคือ บิทที่ 9 (TB8 ในการส่งข้อมูล และ RB8 ในการรับข้อมูล) Start Bit และ Stop Bit รวมกับบิทของข้อมูลก็ได้ทั้งหมด 11 บิท การรับ Stop Bit การตั้ง BAUD RATE สามารถ โปรแกรมได้ทั้งแบบ 1/32 หรือ 1/64 ของความถี่ออสซิลเลเตอร์ในโหมด 2 การทำงานของการรับ และ ส่งข้อมูลแบบ อนุกรมจะแสดงในรูปที่ 7



รูปที่ 7

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

MODE 3 การทำงานจะคล้ายโหมด 2 แต่สามารถตั้ง BAUD RATE โดยจะใช้ตัวแปรหลายค่าของ BAUD RATE ให้ตัวจับเวลาที่ 1 หรือ 2 ขึ้นอยู่กับสถานะ TCLK และ RCLK การทำงานในโหมด 3 แสดงในรูปที่ 8



รูปที่ 8

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

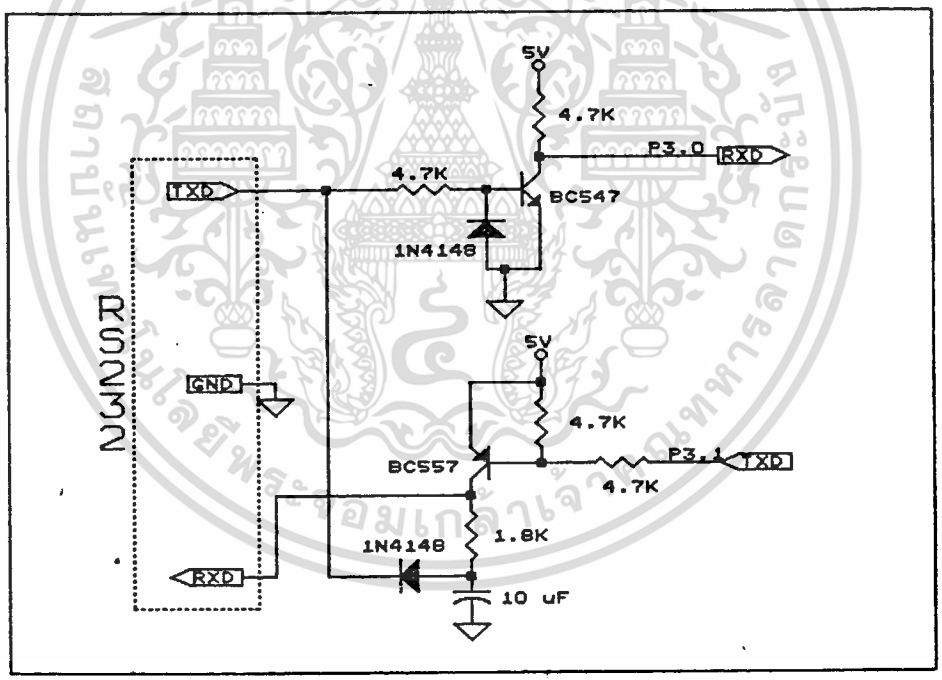
การใช้ตัวจับเวลา (Timer/Counter)เป็นตัวสร้าง BAUD RATE สามารถใช้ตารางในการสร้าง BAUD RATE

BAUD RATE Hz	Fosc MHz	SMOD	TIMER 1		
			C/T	MODE	TH1
Mode 0 1M	12	X	X	X	X
Mode 2 375K	12	1	X	X	X
Mode 1,3					
62.5 K	12	1	0	2	0FFH
19.2 KHz	11.059	1	0	2	0FDH
9.6 KHZ	11.059	0	0	2	0FDH
4.8 KHz	11.059	0	0	2	0FAH
2.4 KHz	11.059	0	0	2	0F4H
1.2 KHz	11.059	0	0	2	0F8H
137.5 Hz	11.059	0	0	2	1DH
110 Hz	6	0	0	2	72H
110 Hz	12	0	0	1	0FEEBH

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การใช้วงจรในการ Drivers ในการติดต่อกับ RS-232

ในการต่อกันระหว่าง พอร์ท แบบอนุกรมของ MCS-51 จำเป็นต้องวงจรในการปรับระดับของแรงไฟมาตรฐานของการสื่อสาร คือ ± 12 ถึง ± 15 V มี Buffer ที่จะใช้นิยมในการต่อได้แก่ ไอซีเบอร์ MC1488 ในการส่งข้อมูลไปที่ RS-232 และ เบอร์ MC1489 ในการแปลงระดับของสัญญาณจาก RS-232 ซึ่งการใช้ Buffer นี้จำเป็นต้องใช้แรงดันไฟในการขับ Buffer เบอร์นี้ดังนั้น เราสามารถต่อด้วยวงจรที่ใช้ Transistor ในการแปลงสัญญาณโดยเราจะอาศัยแรงไฟจาก RS-232 มาใช้งาน ซึ่งการวงจรตามวงจรรูปที่ 9



รูปที่ 9 วงจรในการแปลงสัญญาณจาก RS-232 โดยใช้ Transistor

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อุปกรณ์

1.MCS-51 Traning Board หรือ JAZZ31 1

2.PC AT/XT COMPAT 1

3.ซอฟแวร์ในการติดต่อข้อมูลแบบอนุกรม

5. BC 547 1

6. BC 557 1

7. DIODE 1N4148 2

8. R 4.7K 4

9. R 1.8K 1

10.C 10 uF 1

11.สายต่อ Serial Port (RS-232) 1

12.แผงช่วยในการทดลอง 1

13.สายต่อวงจรแหล่งจ่ายไฟ

การทดลอง

1. ให้ต่อวงจรตามรูปที่ 9
2. ให้ทำการถอด IC MAX232 หรือวงจร Buffer ภายใน ตัว JAZZ-31 หรือ MCS-51 Traning Board
3. ป้อนโปรแกรมที่ 1 แล้วสั่งให้โปรแกรมทำงาน

โปรแกรมที่ 1

```
8000             1              org 8000h
8000 759852      2         mov scon,#01010010b ;Set Mode 1
8003 758920      3         mov tmod,#00100000b ;Set Timmer 1 = Mode
8006 7588FD      4         mov th1,#11111101b  ;Set Time
8009 D28E        5         setb TR1            ;Start Timmer
800B 3098FD      6  rx:    jnb RI,$            ;If RI = 0 ,Data In form RS-232
800E C298        7         clr RI
8010 E599        8         mov a,SBUF          ;Read Data
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
8012 F590        9          mov p1,a            ;Send Data To Logic Monitor
8014 12801E      10         lcall tx1           ;Send Data To RS-232
8017 7420        11         mov a,#' '          ;Data = 20H
8019 12801E      12         lcall tx1           ;Send Data To RS-232
801C 80ED        13         sjmp rx
                 14
                 15  ;SUB Send Data To RS-232
801E 3099FD      16  tx1:    jnb TI,$
8021 C299        17         clr TI
8023 F599        18         mov SBUF,a
8025 22          19         ret
```


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5. ทำการ RUN โปรแกรมในการติดต่อข้อมูล ผ่าน RS-232 ตั้ง BAUD RATE 9600

6. ให้ทำการกด Key Board ที่ PC สังเกตุผลที่เกิดขึ้นที่ Logic Monitor

7. ให้ป้อนโปรแกรมที่ 2 เข้าเครื่อง

โปรแกรมที่ 2

```
8000              1          org 8000h
8000 759852       2          mov scon,#01010010b ;Set Mode 1
8003 758920       3          mov tmod,#00100000b ;Set Timmer 1 = Mode
8006 758DFD       4          mov th1,#11111101b  ;Set Time
8009 D28E         5          setb TR1            ;Start Timmer
800B 3098FD       6  rx:     jnb RI,$            ;If RI = 0 ,Data In form RS-232
800E C298         7          clr RI
8010 E599         8          mov a,SBUF          ;Read Data
8012 B40D0D       9          cjne a,#0dh,tx0
8015 90802C      10          mov dptr,#table
8018 E0          11  tx:     movx a,@dptr
8019 12804C      12          lcall tx1
801C A3          13          inc dptr
801D B40AF8      14          cjne a,#0ah,tx
8020 80E9        15          sjmp rx
8022 12804C      16  tx0:    lcall tx1           ;Send Data To RS-232
8025 7420        17          mov a,#' '          ;Data = 20H
8027 12804C      18          lcall tx1           ;Send Data To RS-232
802A 80DF        19          sjmp rx
802C 54657374    20  table:  db "Test Program Serial Port RS-232",0ah
8030 2050726F6772616D2053657269616C20
8040 506F72742052532D3233320A

                 21
                 22  ;SUB Send Data To RS-232
804C 3099FD      23  tx1:    jnb TI,$
804F C299        24          clr TI
8051 F599        25          mov SBUF,a
8053 22          26          ret
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

8. ทำตามข้อ 5 และข้อ 6 ให้สังเกตุผลที่หน้าจอของ PC

9. ให้ลองเปลี่ยนค่า BAUD RATE ในการส่งข้อมูลแบบอนุกรมเป็น 2400 BAUD RATE โดยการเปลี่ยนค่า ของ TH1 ตามตาราง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทดลองที่ 6

KEY BOARD

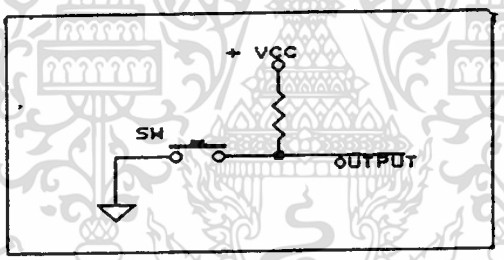
วัตถุประสงค์

1. เพื่อศึกษาการทำงานของ Micro Controller MCS-51 ในการรับข้อมูลจาก Key Board

2. เพื่อศึกษาปัญหาจากการรับข้อมูลจาก Key Board และสามารถเขียนโปรแกรมในการแก้ปัญหาที่เกิดขึ้นในการรับข้อมูลจาก Key Board

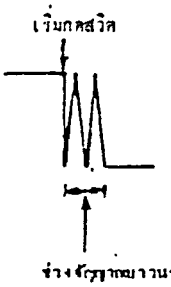
ทฤษฎี

สวิทช์เป็นอุปกรณ์พื้นฐานอย่างหนึ่งสำหรับระบบไมโคร เราใช้สวิทช์เพื่อเป็นอุปกรณ์ในการตรวจจับสัญญาณบางอย่างที่ต้องการ เช่น การเข้ารหัส ลิมิตสวิทช์ เป็นต้น โดยจะทำการต่อสวิทช์ เข้ากับบิท ใดของ พอร์ท อินพุท ลักษณะการต่อสวิทช์ เข้ากับพอร์ท อินพุท จะต่อวงจรตามรูปที่ 1



รูปที่ 1 วงจรในต่อสวิทช์ เข้ากับพอร์ท อินพุท

จากรูปที่ 1 D0 - D7 ของพอร์ทอินพุทต่อกับสวิทช์ และมีรีจิสเตอร์ Pull up ในการยกระดับของอินพุท เมื่อไม่มีการกดสวิทช์ ข้อมูลที่ที่รับเข้าไปจะเป็น "1" ถ้ามีการกดสวิทช์โดยทั่วไปจะปรากฏ พัลส์ที่ไม่ต้องการขึ้นมา ซึ่งเกิดจากการกระเด้งของหน้าสัมผัสปรากฏการณ์นี้เรียกว่า การ บาวนซ์ (Bounce) ดังรูปที่ 2

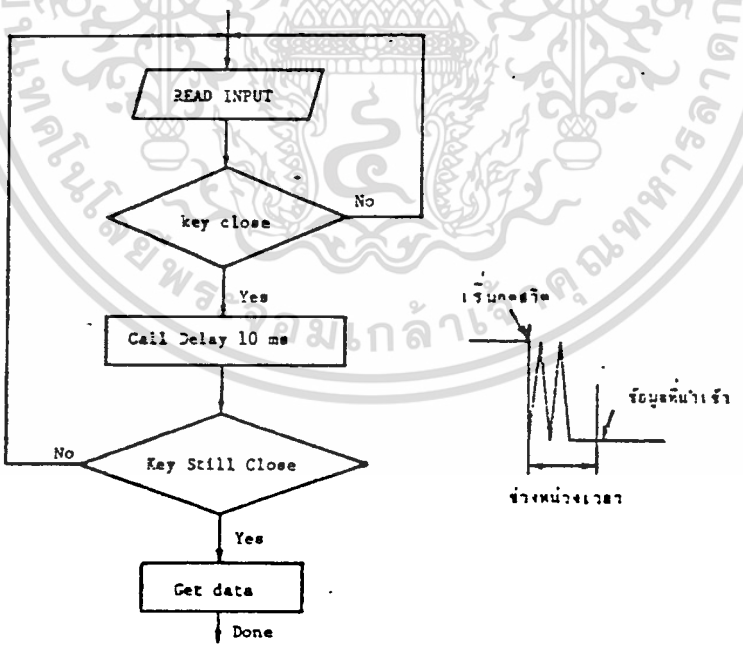


รูปที่ 2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

วิธีการแก้ บราวซ์ (Bounce) สามารถทำได้โดยวิธีทาง Hard ware หรือ ซอฟแวร์ ก็ได้วิธีการทาง Hard ware โดยทั่วไปจะใช้วงจร Mono stable เพื่อที่จะหน่วงเวลาของ สัญญาณ ในลักษณะการเกิดการ บาวนซ์ มีผลต่อการการป้อนสัญญาณในวงจร Logic เป็นอย่างมาก คือเมื่อเราต้องการกดสวิทช์ 1 ครั้งเพื่อที่จะได้ 1 พัลส์ แต่ถ้ามีการเกิด บราวซ์ สัญญาณจะได้มีมากกว่า 1 พัลส์ ค่าที่จะทำการตรวจสอบที่ไม่แน่นอน อาจจะเป็น 0 หรือ 1 ก็ได้ ดังนั้นการกำจัดสัญญาณ บาวนซ์ อาจจะทำได้โดยวิธีการทางด้าน อาร์แวร์ หรือ ซอฟแวร์ แต่ที่นิยมใช้ในระบบไมโครคอมพิวเตอร์ก็คือ ด้วย ซอฟแวร์ เพราะ ไม่ต้องเพิ่มอุปกรณ์เข้าไป

การแก้สัญญาณ บาวนซ์โดยใช้ ซอฟแวร์ โปรแกรมจะทำงาน โดยการตรวจสอบ การกดสวิทช์ครั้งแรก แล้วรอสักระยะเวลาช่วงหนึ่ง เพื่อที่จะให้ผ่านช่วงเวลาที่เกิดการ บาวนซ์ ผ่านไป แล้วก็ทำการอ่านข้อมูลจากสวิทช์อีกครั้งหนึ่งเพื่อจะนำไปเป็นข้อมูล การแก้สัญญาณ บาวนซ์ นั้นสามารถนำมาเขียนเป็น Flow Chart ดังต่อไปนี้



Flow Chart ในการแก้สัญญาณบาวนซ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การต่อสวิทช์จำนวนมาก (Key Board)

จากรูปที่ 1 จะเห็นว่าการรับข้อมูลจากสวิทช์จะใช้ 1 Bit ต่อ สวิทช์ 1 ตัว แต่ในการใช้งานใน Micro Controller ต้องการใช้สวิทช์หรือ Key Board จำนวนมาก ดังนั้นถ้าใช้ 1 สวิทช์ต่อ 1 Bit จะทำให้สิ้นเปลืองพอร์ทเป็นจำนวนมาก เช่นถ้าต้องการรับข้อมูลจาก สวิทช์ 32 ตัว ต้องใช้พอร์ทแบบ 8 Bit จำนวนถึง 4 พอร์ทด้วยกัน

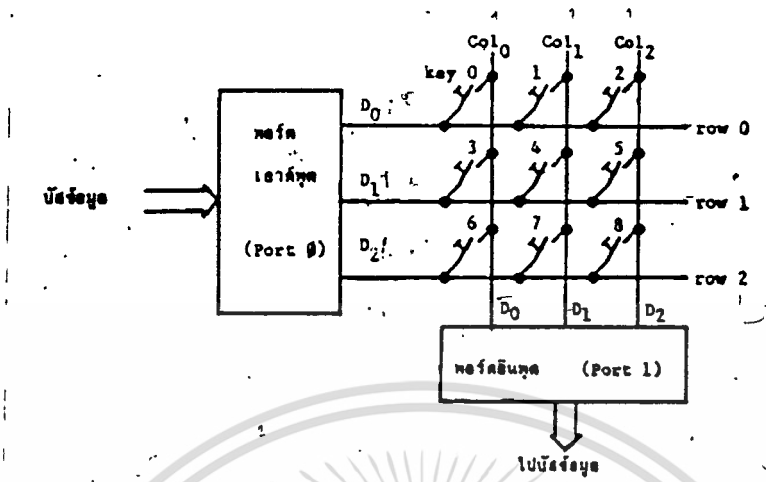
การต่อสวิทช์แบบที่ประหยัดพอร์ท และนิยมใช้ในปัจจุบัน คือ การต่อแบบแมทริกซ์ การต่อด้วยวิธีนี้จะลดจำนวนการใช้พอร์ทลง และจำนวนของสายในการต่อลงด้วยจากตารางที่ 1 เป็นการแสดงการใช้จำนวนของสายในการรับข้อมูลจากสวิทช์แบบธรรมดา และ แบบ แมทริกซ์

จำนวนของสวิทช์	จำนวนของสายที่ใช้ในการต่อสวิทช์	
	แบบธรรมดา	แบบแมทริกซ์
1*1	1	1
2*2	4	4
4*4	16	8
5*5	25	10
8*8	64	16

ตารางที่ 1

จากตารางที่ 1 จะว่าการต่อสวิทช์แบบแมทริกซ์จะประหยัดสายได้มากกว่าการต่อสวิทช์แบบธรรมดาสะดวกในการออกแบบทาง อาร์ดแวร์ และการตรวจสอบสวิทช์ที่กด จะใช้ ซอฟท์แวร์เข้ามาช่วยในการรับข้อมูลจากสวิทช์ โดยซอฟท์แวร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

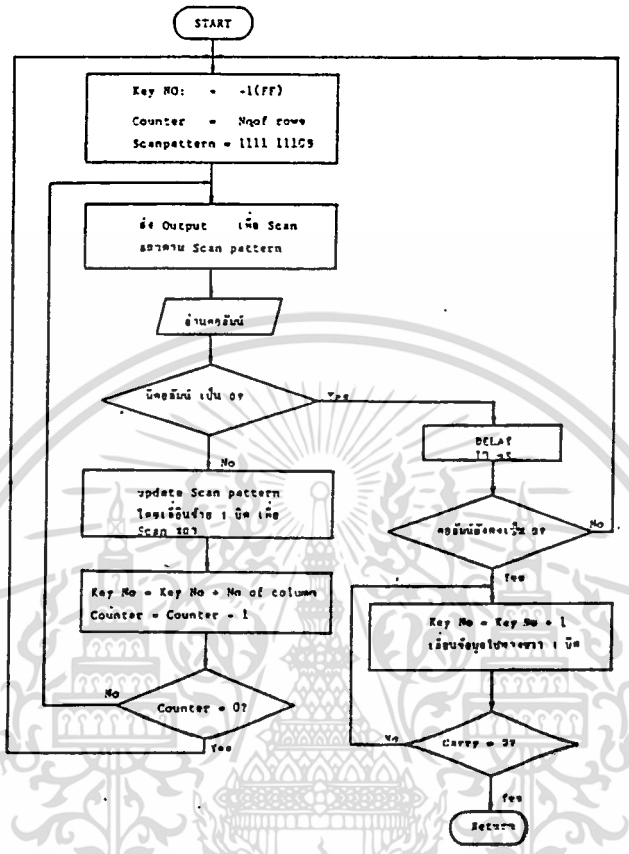


รูปที่ 3 การต่อสวิทช์แบบ แมทริกซ์ 4*4

หลักการทำงานของโปรแกรมจะใช้หลักการในการกวาด (Scanning) เพื่อหาว่าสวิทช์ใดที่ถูกกดโดยอาจเริ่มต้นที่ แถวที่ 0 โดยกำหนดให้แถวที่ 0 มีระดับเป็น 0 ส่วนแถวอื่นเป็น 1 แล้วทำการตรวจสอบทีละ Column โดยเริ่มจากคอลัมน์ 0 ตรวจสอบว่าทาง Row ว่าเป็น 0 หรือไม่ ถ้าไม่ก็ให้ทำการเพิ่มค่าของการ Scan ทาง Column แล้วทำการเช็คทาง Row ทำไปเรื่อย ๆ จนกว่าครบการ Scan ทาง Column ทุก Column ถ้ายังตรวจสอบไม่เจอว่าทาง Row ไม่เป็น 0 ก็แสดงว่าไม่มีการกด Key แต่ถ้ามีถ้าเช็คว่ามี 0 เกิดขึ้นที่ Row ก็ให้ทำการ หน่วงระยะเวลา 10 mS แล้วทำการเช็คข้อมูลทาง Row อีกครั้ง นำเอามา ทำให้เป็น ตำแหน่งของ Key ทาง Row และ Column ดังแสดงใน Flow Chart ที่ 2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Flow Chatr ที่ 2 แสดงการรับข้อมูลจาก Key 4*4



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

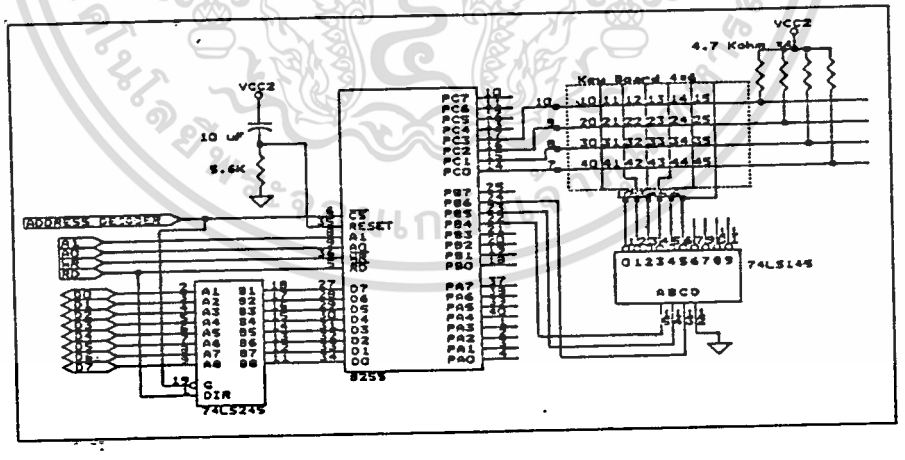
อุปกรณ์

1.MCS-51 Traing Board หรือ JAZZ31	1
2.แผงช่วยฝึกการทดลอง	1
3.KEY BOARD 6*4 (24 KEY)	1
4. R 4.7 K	4
5. 8255	1
6. 74LS245	1
7. 74LS145	1

8.สายต่อวงจร และ แหล่งจ่ายไฟ

การทดลอง

1.ต่อวงจรตามรูปที่ 4



รูปที่ 4

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. ป้อนโปรแกรมที่ 1

โปรแกรมที่ 1

```
                    1  ;program key
                    2  ;keyboard 24 key
                    3  ;portb (pb4-pb6) is scan
                    4  ;portc (pc0-pc3) is input key
                    5
8000                6          org 8000h
DE03=               7          conf equ 0de03h
DE00=               8          porta   equ 0de00h
DE01=               9          portb   equ 0de01h
DE02=              10          portc   equ 0de02h
8500=              11          wrbyte  equ 8500h
8200=              12          inlcd   equ 8200h
                   13          ;intial 8255
                   14
8000 90DE03        15  start:  mov dptr,#conf
8003 7489          16          mov a,#10001001b ;port a o/p ,portb o/p, portc i/o
8005 F0            17          movx @dptr,a
8006 128246        18  st:     icall key
8009 908210        19          mov dptr,#table
800C F582          20          mov dpl,a
800E E0            21          movx a,@dptr
800F F590          22          mov p1,a
8011 80F3          23          sjmp st
                   24
8210               25          org 8210h
8210 30343843      26  table:  db '0','4','8','C',01,02
8214 0102
8220               27          org 8220h
8220 31353944      28          db '1','5','9','D',11h,12h
8224 1112
8230               29          org 8230h
8230 32364145      30          db '2','6','A','E',21h,22h
8234 2122
8240               31          org 8240h
8240 33374246      32          db '3','7','B','F',31h,32h
8244 3132
                   33  ;use reg acc,r2,dptr,b
8246               34  key:
8246 7A00          35  key0:   mov r2,#0
8248 90DE01        36  key1:   mov dptr,#portb
824B EA            37          mov a,r2
824C C4            38          swap a
824D F0            39          movx @dptr,a
824E 90DE02        40          mov dptr,#portc
8251 E0            41          movx a,@dptr
8252 540F          42          anl a,#0fh
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
8254 B40F06      43          cjne a,#0fh,key3  ;if acc.7 ="0" close key
8257 0A          44  key2:   inc r2
8258 BA06ED      45          cjne r2,#5,key1
825B 80E9        46          sjmp key0
825D 128270      47  key3:   lcall delay
8260 540F        48          anl a,#0fh
8262 F4          49          cpl a
8263 75F000      50          mov b,#0
8266 05F0        51  key4:   inc b            ;table colum
8268 13          52          rrc a
8269 50FB        53          jnc key4
826B E5F0        54          mov a,b
826D C4          55          swap a
826E 4A          56          orl a,r2        ;out colum = msb ,low = lsb
826F 22          57          ret
                 58
                 59  ;delay time 10 mS
8270 780A        60  delay:  mov r0,#10
8272 79FA        61  delay1: mov r1,#250
8274 00          62  de:     nop
8275 00          63          nop
8276 00          64          nop
8277 D9FB        65          djnz r1,de
8279 D8F7        66          djnz r0,delay1
827B 22          67          ret
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. ต่อ Port 1 เข้ากับ Login Monitor

4. สั่งให้เครื่องทำงานตามโปรแกรมที่ 1

5. ให้ทำการกด Key แล้วสังเกตุการเปลี่ยนแปลงที่ Logic Monitor

6. ให้เปลี่ยนค่าของ Delay Time เป็น 1 mS สังเกตุผลที่เปลี่ยนแปลง

7. ให้เปลี่ยนค่าในตาราง TABLE สลับตำแหน่งของรหัส แล้วสั่งให้โปรแกรมทำงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คำถาม

1. ถ้าเราไม่มีการ หน่วงเวลาในช่วงของการรับ Key จะเกิดอะไรขึ้นกับข้อมูลที่ได้รับ

2. จงเขียนโปรแกรมในการควบคุม Key Board 4*4 โดยใช้วงจรที่รูปที่ 4

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทดลองที่ 7
สเต็ปปิ้งมอเตอร์ (STAPPING MOTER)

วัตถุประสงค์

1. เพื่อศึกษาการทำงานของ สเต็ปปิ้งมอเตอร์ (Stapping Moter)
2. สามารถเขียนโปรแกรมในการควบคุมการทำงานของ สเต็ปปิ้งมอเตอร์ (STAPPING MOTER)

ทฤษฎี

Stapping Motor คือ Motor ชนิดหนึ่งแต่สามารถควบคุมการหมุนเป็น Stap โดยใช้ Pluse เป็นตัวควบคุมการหมุน ภายในจะประกอบด้วย โรเตอร์เป็นเหล็กอ่อน ซึ่งมีคุณสมบัติพยายามปรับตัวเองให้อยู่ในแนวที่เส้นแรงแม่เหล็กผ่านมากที่สุด โดยการหมุนตัวเองทำให้เกิดมุมของการหมุนขึ้น และ มอเตอร์จะหยุดหมุน เมื่อเส้นแรงแม่เหล็กที่ตัดผ่านตัวมันถึงจุดที่มากที่สุด ซึ่งจะแสดงในรูปที่ 1

รูปที่ 1 เส้นแรงแม่เหล็กที่ทำให้เกิดแรงบิด

การทำให้ Stepping Motor หมุนก็ทำได้โดยอาศัยการทำให้เกิดเส้นแรงแม่เหล็กเกิดขึ้นเป็นช่วงต่อกันไปเรื่อย การหมุนของ Stepping Motor จะขึ้นอยู่กับการขับกระแสเข้าขดลวดว่าจะให้ไปในทางไหน และเมื่อต้องการให้มอเตอร์หยุดก็หยุดการจ่ายกระแสให้กับขดลวด ของ Stepping Motor ก็จะหยุดหมุน ที่ตำแหน่งสุดท้ายที่มีการับที่ Stepping Moter เราก็สามารถรู้ตำแหน่งของมอเตอร์ได้ โดยการนับจำนวนพัลซ์ที่ป้อนให้กับตัว Stepping Motorเราสามารถคำนวนหาตำแหน่งของ Stepping Motor ตามสูตร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

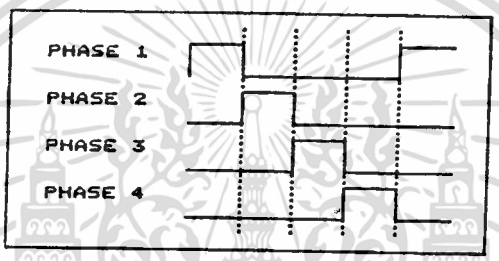
มุมที่เปลี่ยนไป = ค่ามุมต่อสเต็ป * จำนวนพัลส์ที่ป้อนให้

การกระตุ้นเฟสของ Stepping Motor

การทำให้ Stepping Motor หมุนนั้น จะต้องกระตุ้นเฟสของขดลวดของ Stepping Motor ให้เรียงกันไปเรื่อย ๆทางใดทางหนึ่ง ถ้าต้องการให้หมุนกลับก็กระตุ้นเฟสในทิศทางกลับกัน ซึ่งการกระตุ้นเฟสของ Stepping Motor มีอยู่ด้วยกัน 3แบบคือ

1.การกระตุ้นแบบเฟสเดียว Single Phase Excitation

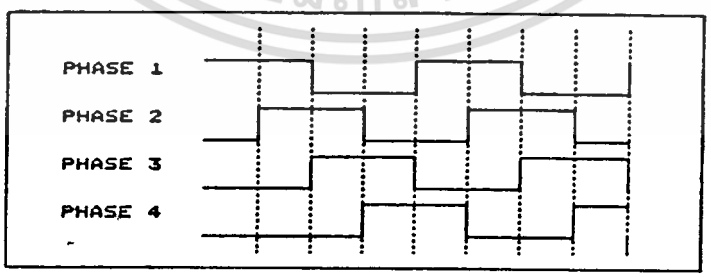
เป็นการกระตุ้นกระแสของ Stepping Motor ทีละขด แสดงในรูปที่ 2



รูปที่ 2

2.การกระตุ้นแบบสองเฟส Two Phase Excitation

เป็นการกระตุ้นกระแสของ Stepping Motor ทีละ 2 ขด ต่อกันไปเรื่อย ๆ ดังจะแสดงในรูปที่ 3

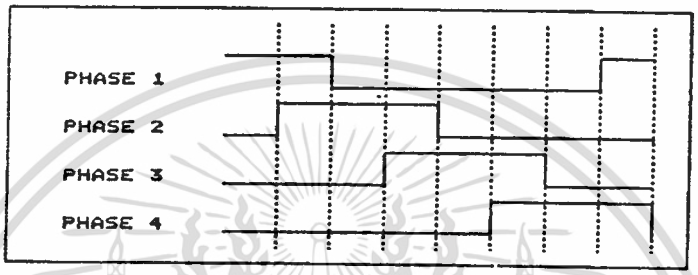


รูปที่ 3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.การกระตุ้นโดยใช้แบบที่ 1 และ 2 สลับกันเรียกว่า Phase Excitation หรือ แบบ Half Step Operation

เป็นการกระตุ้นกระแสของ Stepping Motor ทีละขด กับ 2 ขดสลับกัน ดังจะแสดงในรูปที่ 4



รูปที่ 4

จากรูปที่ 2,3,4 เราสามารถนำมาเขียนเป็นโปรแกรมได้โดยการนำเอาข้อมูล มาทำการ Shift โดยเราเปลี่ยนทิศทางในการหมุนของ Stepping Motor โดยการเป็นทิศทางในการ Shift ของข้อมูลในโปรแกรม เมื่อนำข้อมูลมาทำการ Shift แล้วก็ส่งข้อมูลนี้ออก พอร์ทเพื่อไปส่งสัญญาณให้วงจร Driver กระตุ้นการทำงานของ Stepping Motor

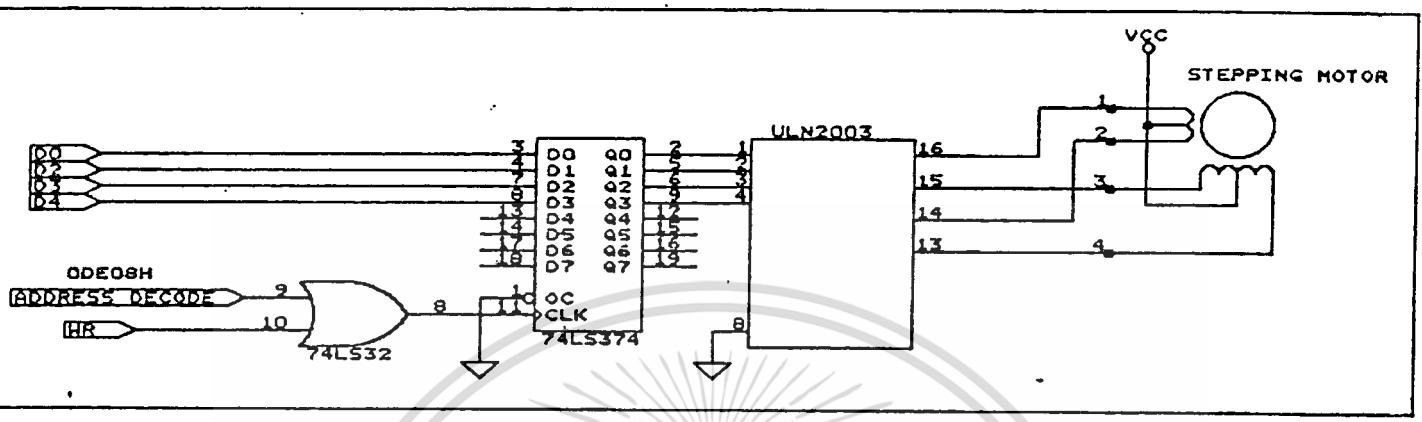
ตัวอย่างของการ Shift เพื่อไปควบคุมการทำงานของ Stepping Motor ในการหมุนแบบ การกระตุ้นแบบเฟสเดียว Single Phase Excitation

DATA = 0001 0001

RR1 DATA = 1000 1000
RR2 DATA = 0100 0100
RR3 DATA = 0010 0010
RR4 DATA = 0001 0001

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากตัวอย่างเราสามารถนำมาต่อกับวงจรในการส่งข้อมูลไปยัง Stepping Motor ได้ดังจะแสดงในรูปที่ 5



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อุปกรณ์

1.MCS-51 Traing Board หรือ JAZZ31 1

2.แผงช่วยฝึกการทดลอง 1

3.Staping Motor 1

4. ULN 2003 1

6.สายต่อวงจร และ แหล่งจ่ายไฟ

การทดลอง

1.ต่อวงจรตามรูปที่ 5

2.ป้อนโปรแกรมที่ 1 แล้วสั่งโปรแกรมเริ่มทำงาน

โปรแกรมที่ 1

```
                    1  ;Program Stepping Motor
8000                2          org 8000h
DE08=               3          ports equ 0de08h
                    4
8000 7411           5          mov a,#11h      ;Data 11h
                    6                          ;Rotate Single Phase Excitation
8002                7  Start:
8002 90DE08         8          mov dptr,#ports
8005 F0             9          movx @dptr,a    ;Send Data To Stepping Motor
8006 23            10          rl a            ;Rotate Left
8007 7864          11          mov r0,#100     ;Delay 100 mS
8009 12800E        12          lcall delay
800C 80F4          13          sjmp Start
                   14
800E               15  delay:
800E 79FA          16  de0:    mov r1,#250     ;Delay Time 1 mS
8010 00            17  de1:    nop
8011 00            18          nop
8012 D9FC          19          djnz r1,de1
8014 D8F8          20          djnz r0,de0
8016 22            21          ret
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.สังเกตุการหมุนของ Stepping Motor

4.ลองเปลี่ยนคำสั่ง RL A เป็น RR A แล้วทดลอง สั่งให้โปรแกรมทำงานสังเกตุการหมุนของ Stapping Motor

5.ให้เปลี่ยนค่า Data จาก 11h เป็น 33h ทำตามข้อที่ 3,4

6.ให้เปลี่ยนค่าของ Delay Time ทำตามข้อ 3,4

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คำถาม

1. ถ้าเปลี่ยนค่า Delay Time เป็น 1 mS มีผลต่อการหมุนของ Stepping Motor เพราะ

2. จงเขียนโปรแกรมควบคุมการหมุนของ Stepping Motor แบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

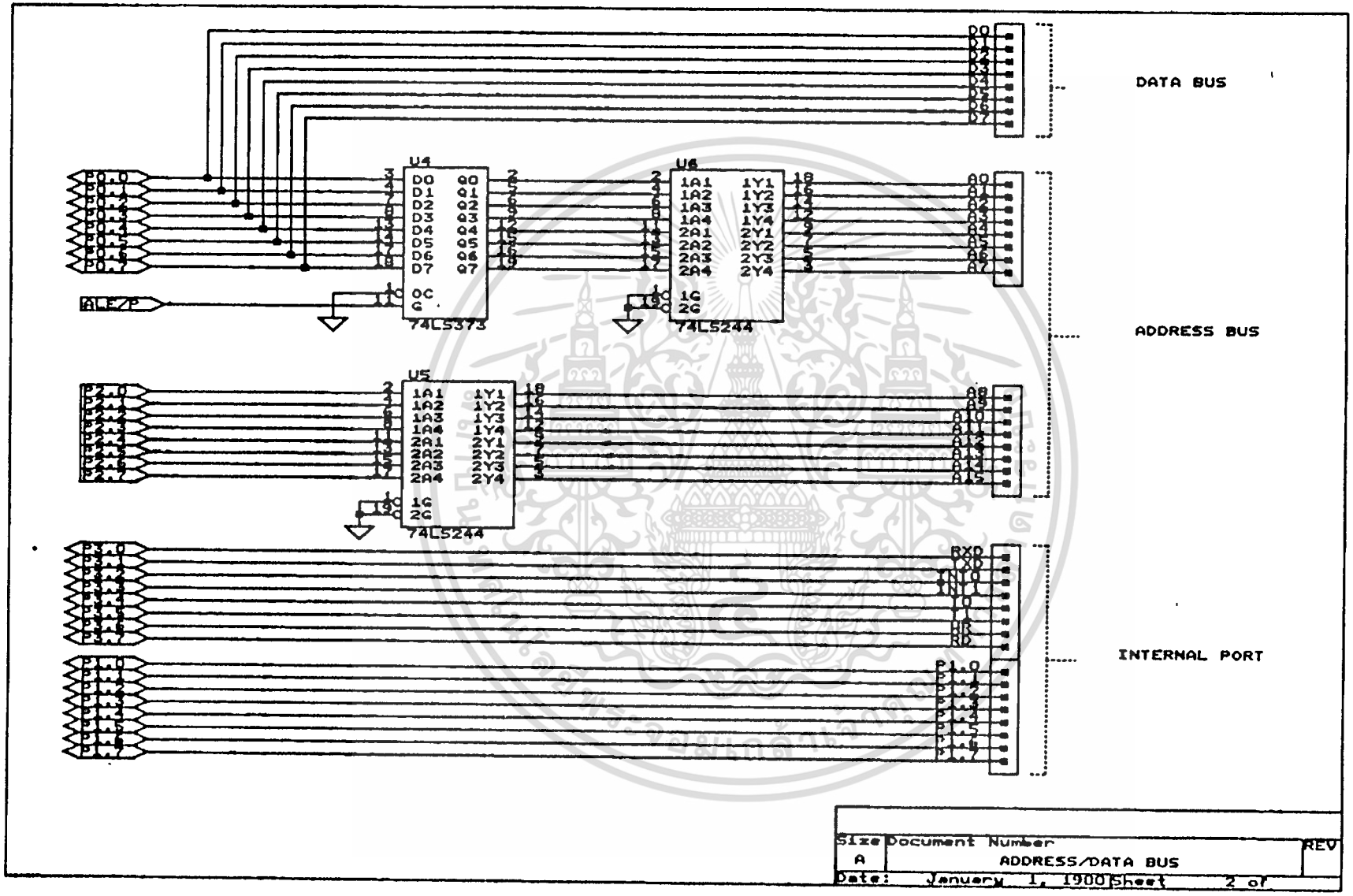
คู่มือแผงช่วยการทดลอง

จากโครงงานนี้ผู้จัดทำเห็นว่าจะเป็นการสะดวกถ้าได้ทำแผงช่วยในการทดลองเพื่อที่จะสะดวกในการทดลอง เพราะถายในแผงช่วยการทดลองนี้สามารถใช้คู่กับ MCS-51 Trining Board และ JAZZ31 ซึ่งได้ออกแบบไว้อุปกรณ์ถายในเพื่อช่วยในการทดลองได้เร็วยิ่งขึ้น ถายในแผงช่วยการทดลองนี้ได้ออกแบบวงจรในการ ติดต่อกับตัว MCS-51 โดยมีสัญญาณ Address Bus, Data Bus,Address Decoder,Logic Monitor,Logic Switch และแหล่งจ่ายไฟ 5 V บนตัวแผงช่วยการทดลองจะมี Foto Board ที่ใช้ในการต่อวงจรเพิ่มเติม จะมีวงจรดังนี้

1.Address Bus และ DATA Bus

เป็นวงจรที่ใช้แยกสัญญาณของ Address Bus A0-A7 กับ Data Bus โดยใช้ไอซีเบอร์ 741s373 ในการแยกสัญญาณ และ สัญญาณ Address Bus จะมี Buffer ในการขับสัญญาณ Address Bus ใช้ไอซีเบอร์ 741s244 เป็นตัว Buffer เพื่อที่จะ สามารถขับสัญญาณถายนอกได้เพียงพอ กันอุปกรณ์ที่ต่อจากภายนอกทำความเสียหายกับ ภายใน Board

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้


DATA BUS
ADDRESS BUS
INTERNAL PORT
U4
74LS373
U6
74LS244
U5
74LS244
ALE/P
Size
A
Document Number
ADDRESS/DATA BUS
REV
Date: January 1, 1900
Sheet 2 of

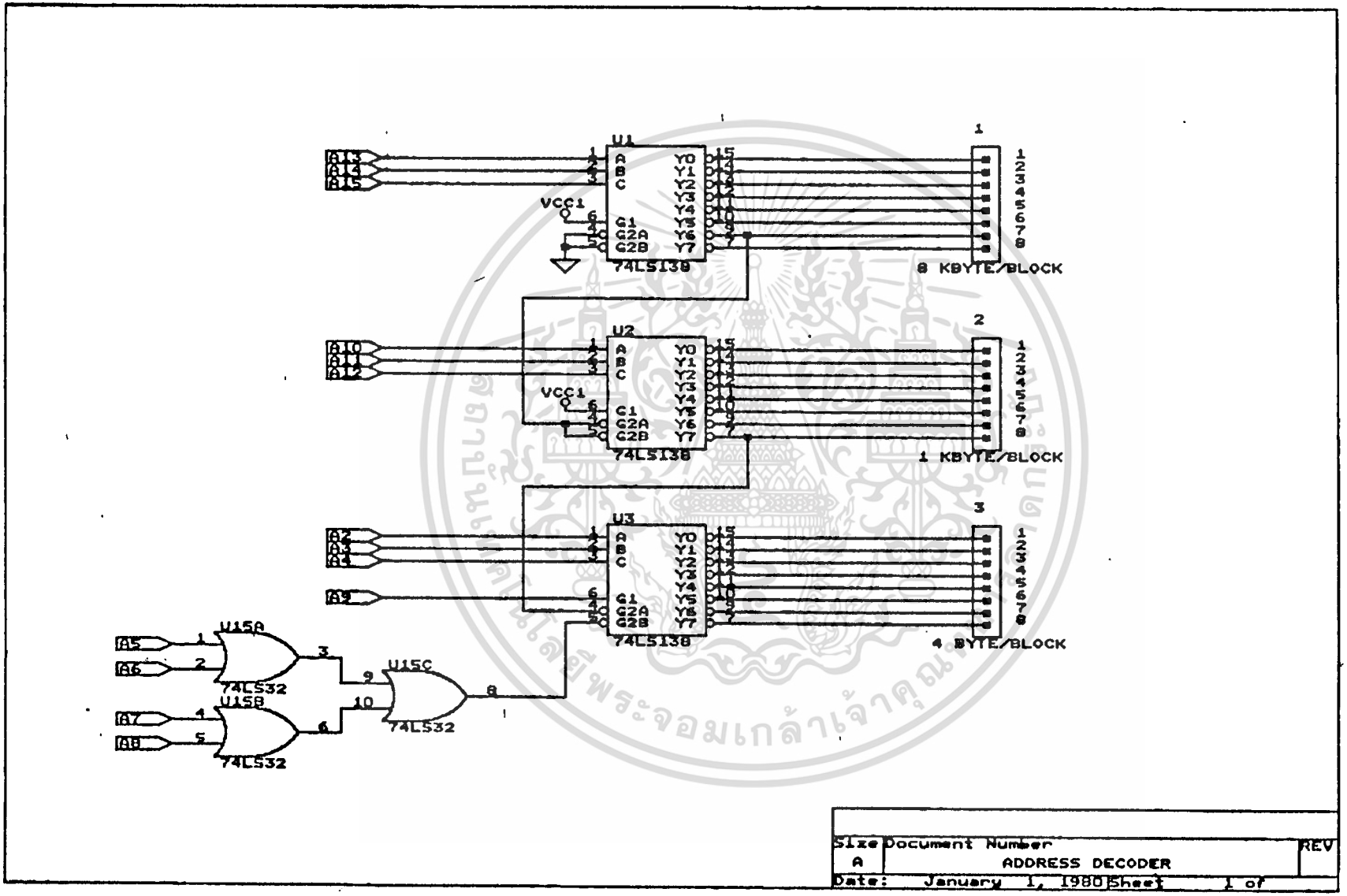
2. Address Decoder

เป็นวงจรที่ทำการ Decode จาก Address Bus เพื่อเป็นสัญญาณใช้เลือกอุปกรณ์ที่จะต่อเพิ่มเติมเข้ามา จะใช้ ไอซีเบอร์ 74ls138 ในการ Decode สัญญาณจาก Address Bus โดยจะ Decode ออกเป็น 3 ช่วงด้วยกัน คือ 8 Kbyte/Block ,1Kbyte/Block และ 4 Byte/Block ซึ่งตำแหน่งของ Address ที่ Decode มาได้จะแสดงตามตารางต่อไปนี้

	1	2	3
1	0000H-1FFFH*	C000H-C3FFH	DE00H-DE03H
2	2000H-3FFFH	C400H-C7FFH	DE04H-DE07H
3	4000H-5FFFH	C800H-CBFFH	DE08H-DE0BH
4	6000H-7FFFH	CC00H-CBFFH	DE0CH-DE0FH
5	8000H-9FFFH*	D000H-D3FFH	DE10H-DE13H
6	A000H-BFFFH	D400H-D7FFH	DE14H-DE17H
7	C000H-DFFFH	D800H-DBFFH	DE18H-DE1BH
8	E000H-FFFFH*	DC00H-DFFFH	DE1CH-DE1FH

TABLE ADDRESS DECODER

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้


U1
A13
A14
A15
A
B
C
G1
G2A
G2B
VCC1
74LS138
Y0
Y1
Y2
Y3
Y4
Y5
Y6
Y7
1
8 KBYTE/BLOCK
U2
A10
A11
A12
2
1 KBYTE/BLOCK
U3
A2
A3
A4
A9
3
4 BYTE/BLOCK
U15A
A5
A6
74LS32
U15B
A7
A8
U15C
Size
A
Document Number
ADDRESS DECODER
REV
Date: January 1, 1980
Sheet 1 of

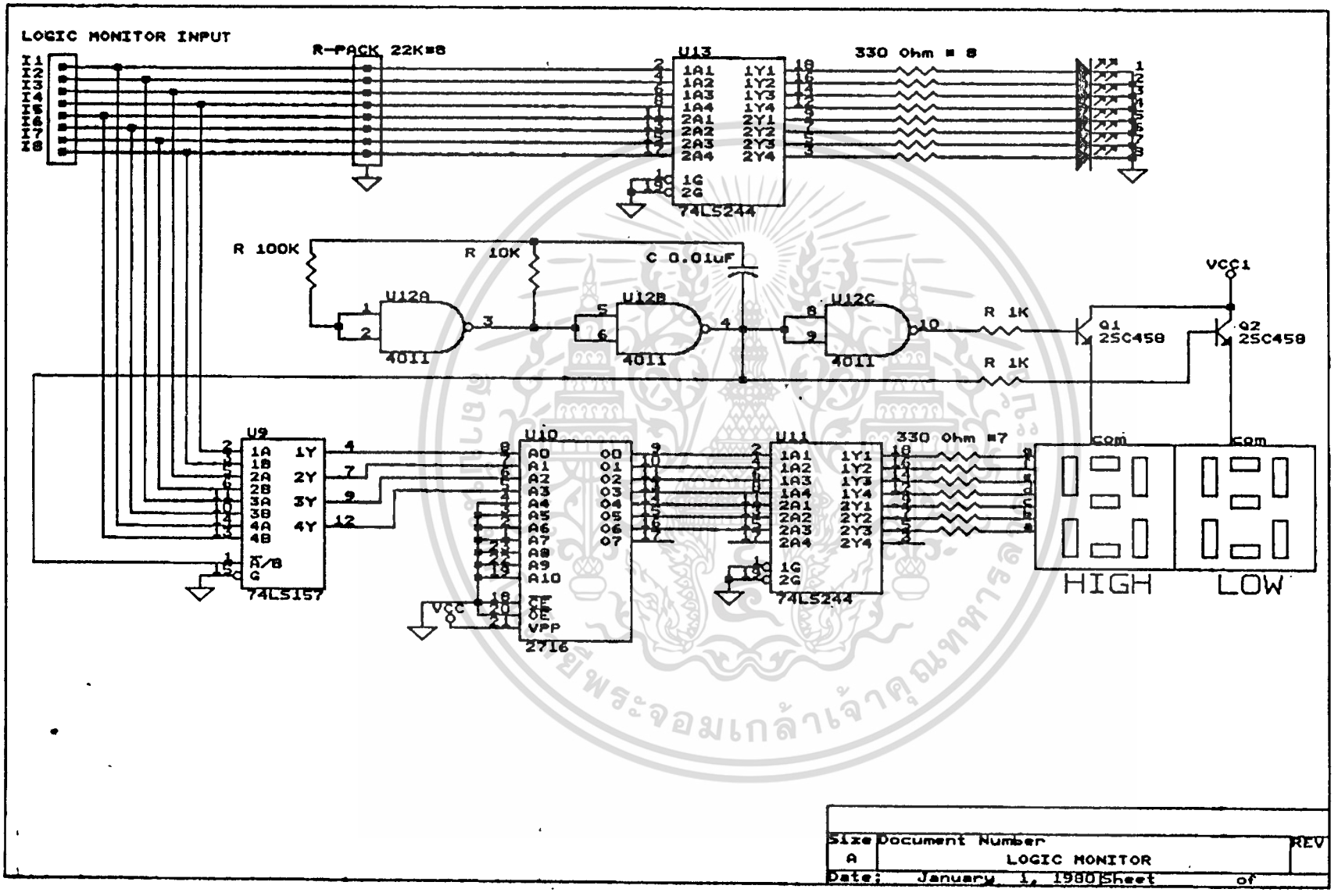
3. Logic Monitor

เป็นวงจรที่ใช้แสดงข้อมูลที่ต้องการจะตรวจสอบ โดยจะแสดงเป็น 2 แบบด้วยกันคือ

Binary จะแสดงผลที่ LED โดยที่ถ้า สว่าง Logic ที่วัดจะเป็น 1, ดับ Logic ที่วัดจะเป็น 0

Hex จะแสดงผลเป็นตัวเลข 0-9 ตัวอักษร A-F โดยการแสดงผล 2 หลัก คือ High I8-I5 และ Low I4-I1 การทำงานของการแสดงผลจะเป็นแบบ มัลติเพล็ก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

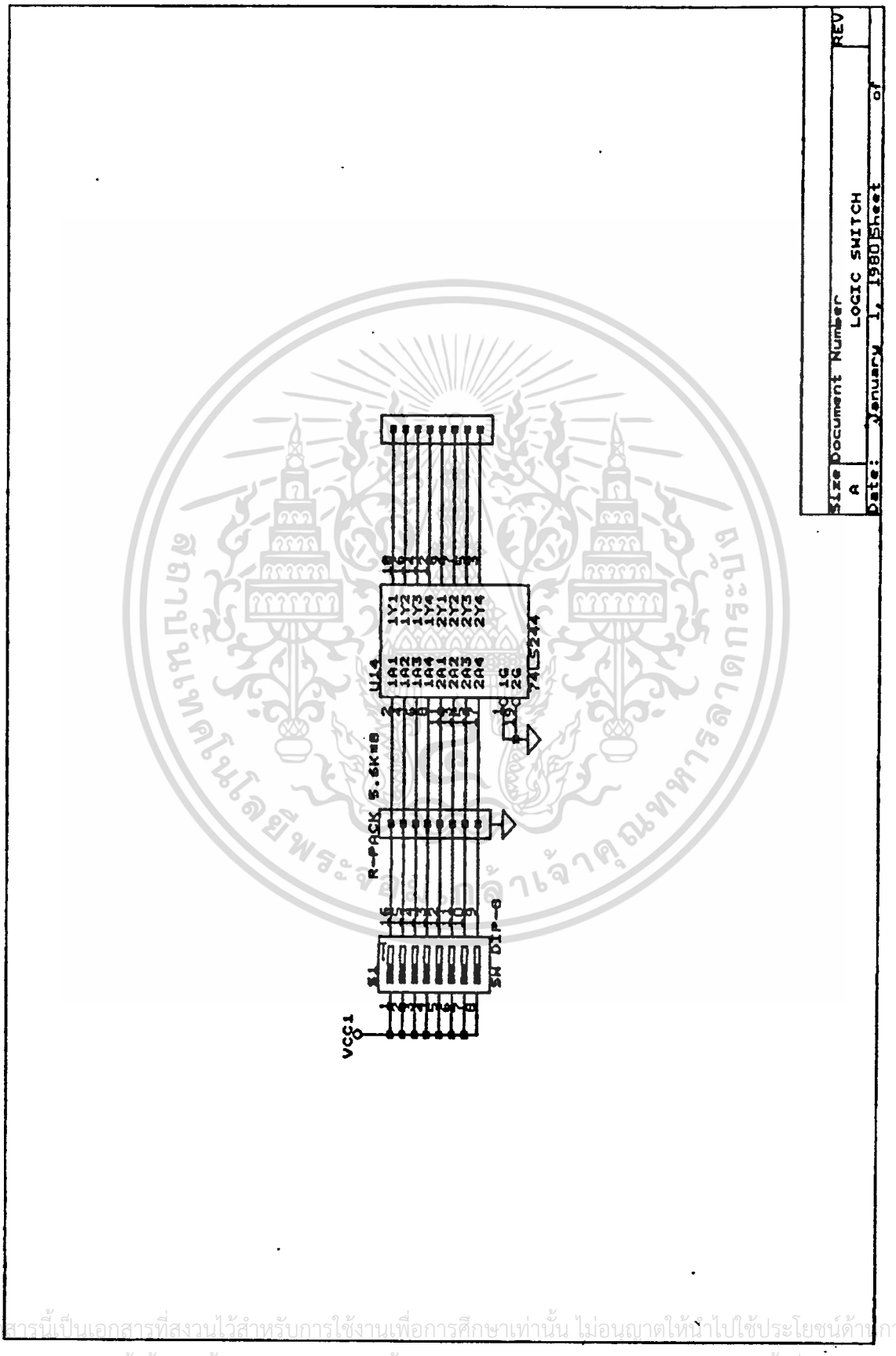


ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

4. Logic Switch

เป็นวงจรในการสร้างสัญญาณ Logic 0,1 ที่ใช้ Dip Switch เป็นตัวเปลี่ยนค่า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

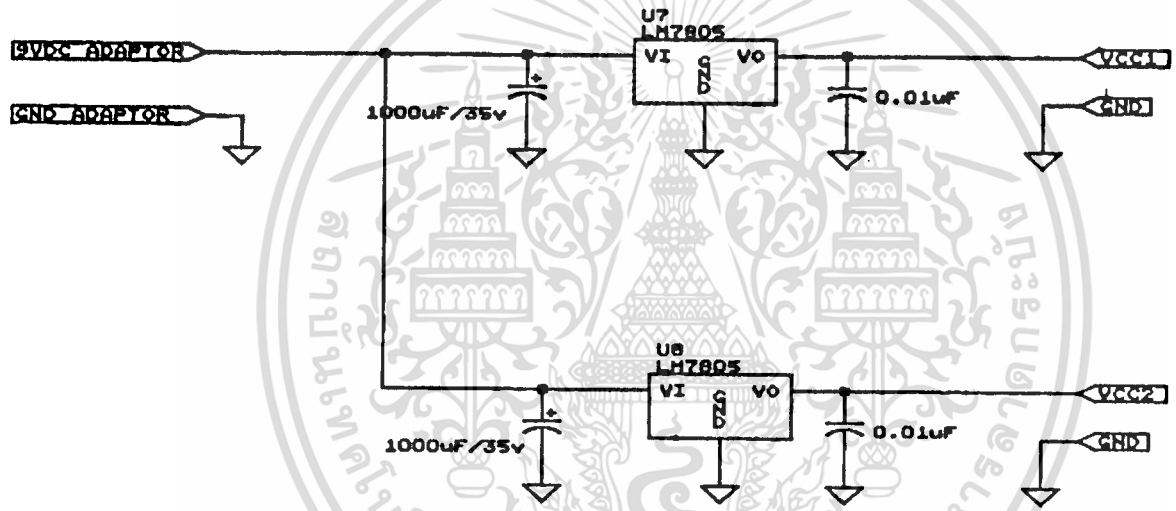


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5. Power

เป็นวงจรในการแปลงแรงดันจาก 9VDC เป็น 5 VDC ซึ่งมีวงจรในการแปลงแรงดัน 2 ตัวด้วยกันคือจ่ายให้กับวงจรในแผงช่วยการทดลอง และ จ่ายให้กับอุปกรณ์ที่จะต่อเพิ่มเติม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



Size	Document Number	REV
A	POWER SUPPLY	3
Date: January 1, 1980	Sheet 3 of	

หนังสืออ้างอิง

[1]. Kenneth J. Ayala, **The 8051 Microcontroller** ARCHITECTURE, PROGRAMMING and APPLICATIONS, 1991

[2]. **intel** คู่มือไอซี ไมโครโปรเซสเซอร์, MICROPROCESSOR DATA BOOK MSC-51 MICROCONTROLLERS

[3]. Signetics Microcontroller User's Guide, 1989

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

ปริญญานิพนธ์เล่มนี้สำเร็จ ลุล่วงไปได้ด้วยดี เพราะได้รับความช่วยเหลือชี้แนะข้อคิดเห็นต่างๆ ได้เรียนคอร์สหนึ่งจากอาจารย์ที่ปรึกษา จึงกราบขอบคุณ อาจารย์ ผศ. พิพัฒน์ เลาหสงคราม

หากปริญญานิพนธ์นี้พอมีประโยชน์อยู่บ้าง ขอมอบความดี ให้ผู้มีพระคุณของ อาจารย์ ผศ. พิพัฒน์ เลาหสงคราม ผู้ที่ให้ คำปรึกษาคำปรึกษา มาโดยตลอด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้