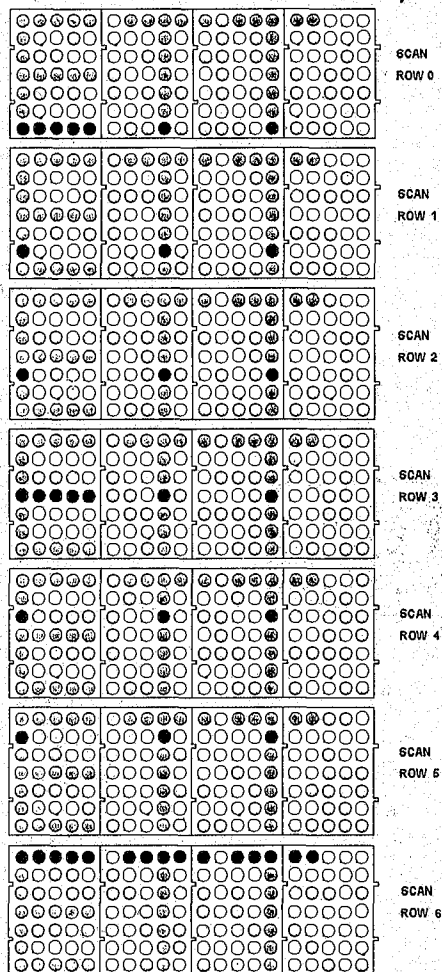


ET-DISPLAY 7x50

Message Display Board



ETT

บริษัท อีทีที จำกัด

1108/32 ถนนสุขุมวิท แขวงพระโขนง เขตคลองเตย กรุงเทพฯ 10110

<http://www.etteam.com>

1108/32 Sukhumvit Rd., Phrahanong Klongtoey BANGKOK 10110

<http://www.ett.co.th>

TEL 712 1120-1, 712 1123, 712 3786, 712 3789, 391 7215 FAX 391 7216

ET-DISPLAY 7X50

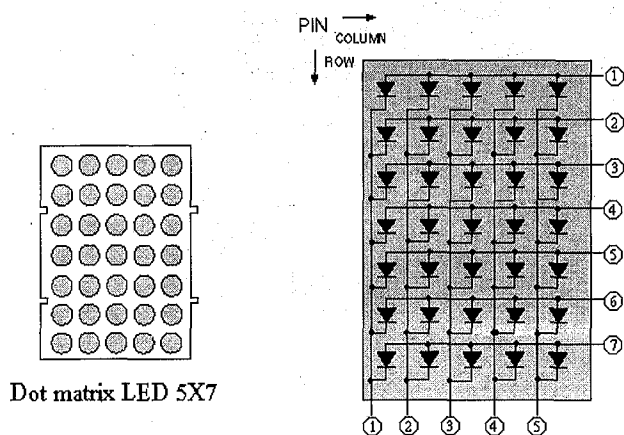


คุณสมบัติ

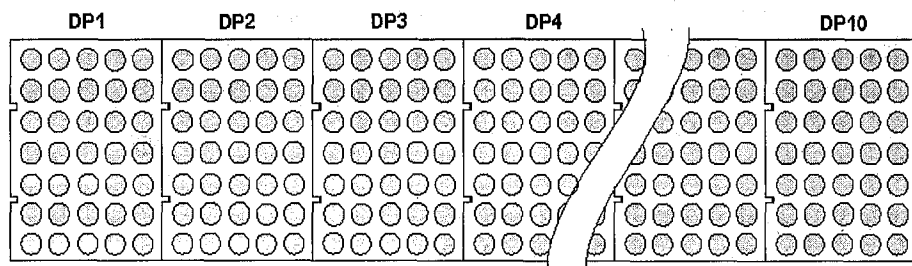
Microcontroller	: AT89S8252
รับข้อมูลผ่าน	: RS232 (Baud rate 19200,N,8,1)
จำนวนของ LED display	: 7 X 50 Dot
ความยาวของข้อความที่จะแสดง	: 64 ตัวอักษรต่อข้อความ (สูงสุด 30 ข้อความ)
จำนวนบรรทัด	: 1 บรรทัด
ภาษา	: อังกฤษ
รูปแบบของการแสดงตัวอักษร	: 1. หยุดนิ่ง : 2. วิ่งจากซ้ายไปขวา : 3. วิ่งจากขวาไปซ้าย : 4. เลื่อนจากบนลงล่าง : 5. เลื่อนจากล่างขึ้นบน
กระแสไฟ	: ไม่เกิน 500 mA
Power supply	: 9 Vac หรือ 5 Vdc (เลือกโดยจัมเปอร์)
อื่นๆ	: มี Switch เอนกประสงค์ 1 ตัว

ลักษณะโดยทั่วไป

ET-DISPLAY 7X50 เป็นบอร์ดแสดงผลตัวอักษรไฟวิ่ง โดยใช้ LED Matrix แบบ 5X7 ที่มีวงจรภายในของ LED ดังรูปที่ 1 จำนวน 10 ตัว มาต่อเรียงกันทางด้าน Column เพื่อให้เกิดเป็น LED DISPLAY ขนาด 7X50 ดังรูปที่ 2

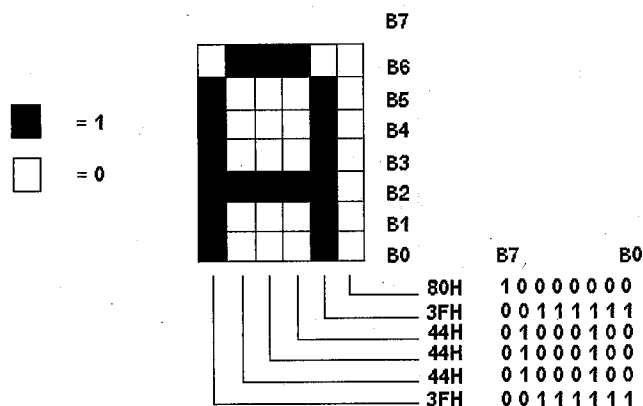


รูปที่ 1



รูปที่ 2

สาเหตุที่ต้องใช้ LED ในแนว Column ถึง 10 ตัวก็เพื่อที่จะต้องการให้แสดงตัวอักษรได้ 8 ตัว อย่างชัดเจนและเพียงพอต่อการแสดงผลข้อความยาวๆได้(สามารถนำหลักการนี้ไปสร้างขนาดที่มากกว่านี้ได้เองได้) โดย 1 ตัวอักษรจะใช้หน่วยความจำในการเก็บข้อมูล 6 ไบท์ ซึ่ง ET-DISPLAY 7X50 นี้สามารถแสดงรหัส ASCII ได้ตั้งแต่ 20H(Space) ถึง 7EH(~) เช่นตัว "A" จะมีข้อมูลเป็นดังรูปที่ 3 และตั้งแต่ 80H ถึง FFH สามารถกำหนดเองได้แต่ในที่นี้ได้ใช้เป็นเก็บตัวอักษรพิเศษและ ICON



รูปที่ 3 แสดงรูปแบบการติดของ LED ของตัว A

หลักการสแกน

การสแกนของ LED Matrix สามารถทำได้ 2 แบบโดยทั่วไปคือสแกนในแนว ROW และการสแกนในแนว Column โดยขึ้นอยู่กับความเหมาะสมของโครงสร้างทาง Hardware และความยากง่ายในการเขียนโปรแกรม(Software) ซึ่งในกรณีของ ET-DISPLAY 7X50 นี้จะใช้การสแกนในแนว ROW เพราะถ้ามีการเพิ่มจำนวนของ Column จะทำให้ความเร็วในการสแกนเกือบจะไม่เปลี่ยนแปลง(โดยหลักการ) การสแกนแบบนี้สามารถแสดงให้ดูได้ทีละสแต็ปได้ดังรูปที่ 4 ซึ่งเป็นการสแกนคำว่า ETT

จากรูปที่ 4 จะเห็นว่าเราเริ่มสแกนที่ ROW 0 ก่อนแล้ว แล้วไปสิ้นสุดการสแกนที่ ROW 6 แล้วกลับมาเริ่มการสแกนที่ ROW 0 อีกครั้ง โดยในการสแกนแต่ละ ROW นั้นจะให้ข้อมูลในแต่ละ ROW ติดช่วงชั่วขณะเพื่อให้ตาของคนเรามองเห็น ซึ่งแต่ละ ROW นั้นควรจะถูกสแกนอย่างน้อย 40-50 ครั้งต่อวินาที ข้อมูลจะถูกเปลี่ยนไปตามตัวอักษรที่ต้องการ โดยเมื่อเราเร่งความเร็วในการสแกนมากขึ้นให้เหมาะสมก็จะเห็นการติดของ LED เป็นตัวอักษรอย่างชัดเจน แต่อย่างไรก็ดีการเพิ่มความเร็วในการสแกนที่เร็วเกินไปก็จะทำให้ความสว่างของ LED ลดลง

สำหรับ ET-DISPLAY 7X50 นี้ จากวงจรในรูปวงจรจะเห็นว่า CPU IC1 เบอร์ AT89S8252 เป็นตัวควบคุมทั้งหมดทั้งการรับข้อมูลและการสแกน LED ซึ่งเป็นการสแกนทางด้าน ROW จะใช้ทรานซิสเตอร์ NPN เป็นตัวจ่ายกระแสไฟให้กับ LED DOT Matrix โดยใช้เบอร์ BC517 ซึ่งเป็นแบบดาร์ลิ่งตันซึ่งถูกถอดรหัสตำแหน่ง Row โดย 4017 เพื่อทำหน้าที่ในการสวิตช์ ให้ทรานซิสเตอร์ On/Off ในตำแหน่ง ROW ที่ถูกต้อง ซึ่ง 4017 นี้เป็นจะต้องไม่ใช่เอาพุทอันแรก(Q0)เพื่อไม่ให้ LED ใน ROW แรกติดหลังจากการรีเซ็ตตัวมัน

การนำข้อมูลออกจาก COLUMN

การนำข้อมูลออกจาก Column ของ ET-DISPLAY 7X50 นี้จะถูกส่งผ่าน Shift Register 74HC595 โดยข้อมูลจะถูกอ่านออกมาจากตารางแล้วนำมาหาบิตที่สัมพันธ์กับ Row และ Column ในขณะนั้น ซึ่งก็คือการหาค่า **Element** ของ Matrix ขนาด 7X6 (Pattern ของตัวอักษร) ที่เป็น 0 หรือ 1 นั้นเอง อย่างเช่น ตัว "A" มี Matrix ของข้อมูลดังนี้

Row	→	0 1 1 1 0 0
		1 0 0 0 1 0
		1 0 0 0 1 0
		1 0 0 0 1 0
		1 1 1 1 1 0
		1 0 0 0 1 0
		1 0 0 0 1 0
	↑	Column

เมื่อได้ค่า Element ที่เป็น 0 หรือ 1 แล้ว จากนั้นจึง Shift ออกไป ซึ่งในที่นี้ค่าที่ถูก Shift ออกไปจะเป็นดังนี้

1 0 0 0 1 0	1 0 0 0 1 0	1 1 1 1 1 0	1 0 0 0 1 0	1 0 0 0 1 0	1 0 0 0 1 0	0 1 1 1 0 0
Row 0	Row1	Row2	Row3	Row4	Row5	Row6

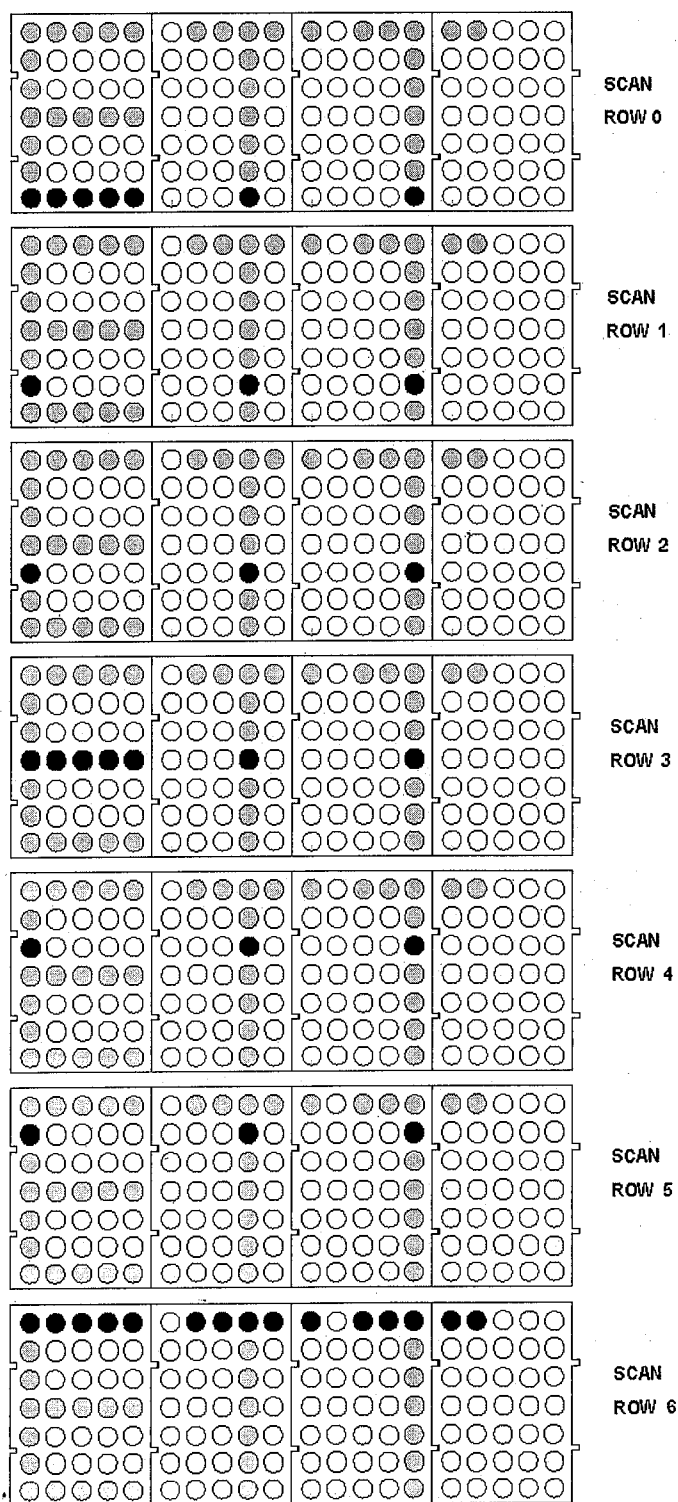
จากข้างบนจะเห็นว่า Row0 จะถูกส่งออกไปก่อน(Row0 เริ่มจากด้านล่าง ซึ่งจะตรงกันข้ามกับMatrix ทั่วไป) และผลของการทำงานอย่างซ้ำๆจะเป็นดังรูปที่ 4

Power Supply

ในส่วนของภาค Power supply นั้นเราสามารถเลือกได้ว่าจะใช้ +5 V จาก DB9 ตัวเมียหรือจาก 7805 ภายในคือถ้าใช้ +5V จากภายนอกก็ให้ขั้ว J3 โดยจะต้องเอาไฟ +5V เข้าที่ขา 9 ของ DB9 และ GND เข้าที่ขา 4 หรือขา 5 ส่วนกรณีจะใช้ +5V จากภายในก็ให้ขั้ว J2 แต่จะต้องไม่ขั้วต่อพร้อมกันเพราะอาจทำให้วงจรเสียหายได้ ซึ่งในกรณีที่ต้องการใช้ +5V จากภายในนี้เราต้องหาแหล่งจ่ายไฟ AC หรือ DC ตั้งแต่ 9V ถึง 16V โดยอาจจะเอาเข้าที่แจ็ค DCin หรือเอาเข้าที่ขา 8 และขา 6 ของ DB9 โดยไม่ต้องสนใจขั้ว ในขณะวงจรทำงานจะมี LED1 (LED POWER) และ LED2 (LED ON) แสดงการทำงานด้วย (ให้ดูรายละเอียดการเชื่อมต่อหน้าที 8)

Switch SW1

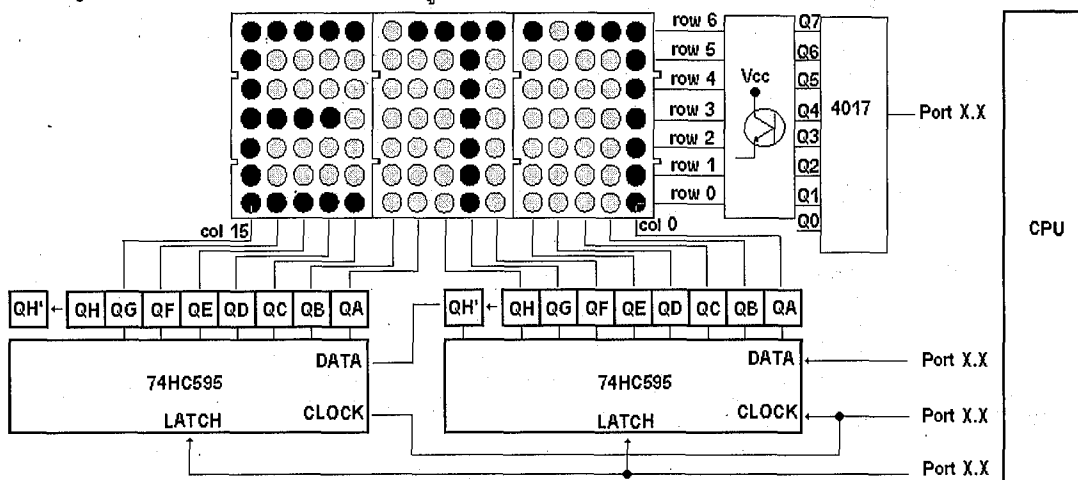
จากวงจรจะเห็นว่า SW1 ต่ออยู่ด้วยซึ่งเป็นสวิตช์เอนกประสงค์ตามแต่ละโปรแกรมให้ CPU ตอบสนอง ซึ่งในที่นี้เมื่อ SW1 ถูกกดจะเป็นการแสดงข้อความชุดถัดไป(ถ้ามีอยู่ใน EEPROM)



รูปที่ 4

การนำข้อมูลออกแสดงผล

การแสดงผลของบอร์ดตัวอักษรไฟวิ่งนั้นส่วนใหญ่ที่พบเห็นกันบ่อยๆ ก็จะทำการอ่านรูปแบบการติดของ LED มาเก็บไว้ในหน่วยความจำในส่วนที่เรียกว่า **Display Buffer** แล้วจากนั้นจึงทำการส่งข้อมูลออกไปที่ LED ซึ่งอาจจะเป็นการ Shift ดังที่กล่าวมาแล้วหรือวิธีอื่นๆ ซึ่งวิธีนี้จะมีประโยชน์ในกรณีที่ต้องการให้มีลูกเล่นในการแสดงผลได้มากมาย โดยเฉพาะเมื่อต้องการเลื่อนตัวอักษรด้วยแล้วยิ่งง่ายเข้าไปอีก เพราะสามารถใช้คำสั่งที่เกี่ยวข้องกับการเลื่อนข้อมูลในหน่วยความจำเท่านั้น จากนั้นโปรแกรมย่อย Scan ก็จะนำข้อมูลออกแสดงผล แต่ข้อเสียของวิธีนี้ก็คือต้องการหน่วยความจำค่อนข้างมากจึงเหมาะกับระบบใหญ่ๆ ที่มีทรัพยากรมากและต้องการความสามารถสูงๆ และผลที่ตามมาก็คือขนาดใหญ่ขึ้นและราคาที่สูงขึ้นตาม แต่ในบอร์ด ET-DISPLAY 7X50 นี้ต้องการให้มีขนาดเล็กและราคาถูกจึงมุ่งไปที่การใช้งานไมโครคอนโทรลเลอร์ในลักษณะ Single chip จึงทำให้ขนาดของหน่วยความจำถูกจำกัดไว้แค่หน่วยความจำภายในของ CPU เท่านั้น แต่ก็ไม่เป็นปัญหาในการทำงาน โดยข้อมูลที่ถูส่งออกไปที่ LED นั้นได้จากการเปิดตารางในส่วนของ Program memory แล้ว Shift ออกไปทันทีโดยไม่มีการเก็บไว้ที่ RAM ก่อน(ยกเว้นกรณีของการเลื่อนตัวอักษรขึ้นหรือลง) จึงทำให้ RAM ส่วนใหญ่ถูกใช้เป็นที่เก็บรหัส ASCII ของตัวอักษรที่จะแสดงผลเท่านั้นและใช้ในส่วนอื่นของโปรแกรมอีกเล็กน้อย โดยโครงสร้างที่สำคัญของ ET-DISPLAY 7X50 แสดงในรูปที่ 5



รูปที่ 5 แสดงบล็อกไดอะแกรมของ ET-DISPLAY 7X50

ขั้นตอนย่อยของการนำข้อมูลออกแสดงที่ LED

1. อ่านรหัส ASCII CODE ที่รับจาก RS232 ที่อยู่ใน Internal RAM ของ CPU
2. เปิดตาราง Pattern ของ ASCII ที่อยู่ใน Program memory
3. อ่านค่า Column Pattern อันแรกของตัวอักษร เช่นตัว "A" มี Pattern อันแรกเป็น 3FH ดังรูปที่ 3
4. หาบิตที่สัมพันธ์กับ Row และ Column ในขณะนั้นซึ่งก็คือการหาค่า Element ของ Matrix ดังที่กล่าวไว้แล้ว
5. Shift ข้อมูลออกไปทาง 74HC595 เท่ากับจำนวน Column
6. ให้ Clock กับ 4017 หนึ่งลูก

การเก็บตัวอักษรที่จะแสดงผล

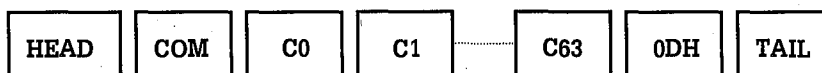
เนื่องจาก ET-DISPLAY 7X50 นี้เป็นบอร์ดแสดงผลที่ไม่มีความซับซ้อนมากจึงทำให้การจัดเก็บข้อมูลไม่มีความซับซ้อนเช่นกัน โดยข้อความที่จะแสดงผลนั้นจะถูกเก็บเป็นรหัส ASCII ใน Internal EEPROM ซึ่งมีได้มากที่สุด 64 ตัวอักษรต่อข้อความ(สูงสุด 30 ข้อความ) **ซึ่งขึ้นต้นด้วย 02H ปิดท้ายด้วย 0DH เสมอ** โดยข้อมูลของตัวอักษรที่เข้ามาทีหลังจะถูกต่อท้ายข้อความแรกและเป็นเช่นนี้ไปเรื่อยๆ เมื่อตัวอักษรที่เข้ามาเต็ม 2 Kb ของ EEPROM แล้วมีข้อความใหม่เข้ามาจะทำให้ข้อความแรกถูกเขียนทับด้วยข้อความใหม่ทันทีส่วนข้อความที่เหลือจะไม่ถูกทับนอกจากมีข้อความใหม่เข้ามาอีก อีกกรณีหนึ่งหลังจากรีเซ็ตหรือหลังจากจ่ายไฟจะทำให้ address ของ EEPROM ในส่วนของการเขียนจะชี้ที่ 000H ซึ่งนั่นก็หมายความว่า ถ้ามีข้อความใหม่เข้ามาจะทำให้ข้อความแรกที่มีอยู่แล้วถูกเขียนทับด้วยข้อความใหม่ แต่ถ้าต้องการลบข้อมูลใน EEPROM ทั้งหมดก็ให้ใช้คำสั่ง Z หรือ กดปุ่มเคลียร์ในโปรแกรม "DP750W.EXE" โดยจะใช้เวลาประมาณ 6 วินาทีซึ่งขณะลบ LED สีเขียวจะติดค้างและหลังจากลบเสร็จแล้ว LED สีเขียวจะกระพริบตามปกติ

ตัวอย่าง เมื่อต้องการแสดงคำว่า "ETT" ข้อมูลตัวอักษรที่เก็บใน EEPROM ของ ET-DISPLAY 7X50 จะเป็นดังนี้

E T T

02H	45H	54H	54H	0DH	← ASCII CODE
C0	C1	C2			← ตำแหน่งของ EEPROM

PROTOCOL หรือ COMMAND ที่ใช้ในการควบคุม ET-DISPLAY 7X50

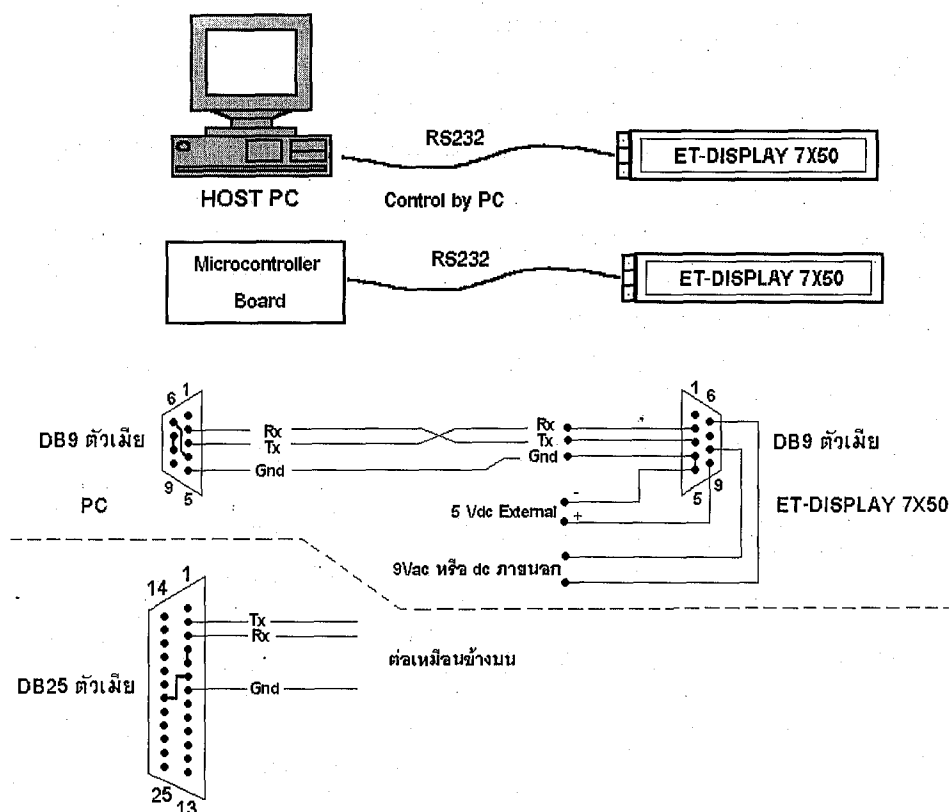


- เมื่อ **HEAD** = ส่วนหัวของเฟรมข้อมูลมีค่าเป็น **01H** เสมอ
- COM** = COMMAND ที่ใช้สั่งให้แสดงผลแบบต่างๆ(ต้องเป็นตัวใหญ่เสมอ)
- A : ตัวอักษรไม่วิ่ง (C0-C7 = 8 ตัว)
 - B : ตัวอักษรเริ่มวิ่งจากมุมขวาไปทางซ้าย (C0-C63 = 64 ตัว)
 - C : ตัวอักษรเริ่มวิ่งจากมุมซ้ายไปทางขวา (C0-C63 = 64 ตัว)
 - D : ตัวอักษรเริ่มวิ่งจากล่างขึ้นบน (C0-C7 = 8 ตัว)
 - E : ตัวอักษรเริ่มวิ่งจากบนลงล่าง (C0-C7 = 8 ตัว)
 - Z : ใช้ลบข้อมูลใน EEPROM (ส่งเพียง HEAD และตัว "Z" เท่านั้น ไม่ต้องส่ง TAIL)
- C0-C63** = เป็นตัวอักษรที่จะแสดง
- 0DH** = เป็นรหัสปิดท้ายข้อความ
- TAIL** = ส่วนหางของเฟรมข้อมูลมีค่าเป็น **04H** เสมอ

หมายเหตุ แต่ละไบต์ห่างกัน 1 mS

การใช้งาน ET-DISPLAY 7X50

การใช้งาน ET-DISPLAY 7X50 นั้นไม่ได้ยุ่งยากมากนัก เพียงแต่ต่อสายที่ใช้ในการส่งข้อมูลเข้าที่ DB9 ตัวเมียบนบอร์ดดังรูปข้างล่าง

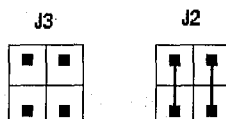


แหล่งจ่ายไฟ

สำหรับไฟที่จ่ายให้กับ ET-DISPLAY 7X50 นั้นสามารถเลือกได้ 3 กรณีโดยการเลือกที่จัมเปอร์ J2 และ J3 ดังนี้

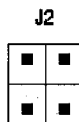
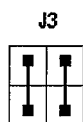
กรณีที่ 1 ใช้ ADAPTOR ภายนอกเสียบเข้าที่ แจ็ค DCin กลม ด้านข้างของตัวบอร์ดซึ่งติดกับ DB9 โดยไฟที่ใช้ ต้องไม่เกิน 16 Vac หรือ dc (แนะนำให้ใช้ 10VDC) แล้วให้ขอรท์ J2 ดังรูปข้างล่าง

กรณีที่ 2 ใช้แหล่งจ่ายภายนอกเข้าที่ ขา 8 กับขา 6 ของ DB9 ไม่เกิน 16 Vac หรือ dc (แนะนำให้ใช้ 10VDC) โดยไม่ต้องสนใจขั้วบวกลบ แล้วให้ขอรท์ J2 ดังรูปข้างล่าง



กรณีที่ 1 และ กรณีที่ 2

กรณีที่ 3 ใช้ไฟ 5 Vdc จากภายนอกผ่านหัว DB9 ตัวเมีย โดยให้ขอรท์ J3 ดังรูปข้างล่างซึ่งในกรณีที่ 2 นี้จะต้องใส่ไฟบวก 5 Vdc ที่ ขา 9 ของ DB9 และ Ground ที่ขา 4 หรือ 5 ของ DB9



กรณีที่ 3 ใช้ 5 V เท่านั้น

หมายเหตุ จะต้องไม่ขอรท์ทั้ง J2 และ J3 พร้อมกันเพราะอาจทำให้วงจรเสียหายได้

การใช้โปรแกรมสำหรับควบคุม ET-DISPLAY 7X50

โปรแกรมที่ใช้ควบคุมการทำงานเป็นโปรแกรมที่ใช้ส่งข้อความและกำหนดรูปแบบการแสดงผล โดยมีทั้งโปรแกรมที่รันบน DOS และ Windows95/98/NT4.0

1. โปรแกรมบน DOS มีชื่อว่า **DP750_1.EXE** สำหรับ ComPort 1 และ **DP750_2.EXE** สำหรับ ComPort 2 ซึ่งเป็นลักษณะ Command line โดยมีวิธีการใช้งานดังนี้

DP750_1 [COMMAND] หรือ DP750_2 [COMMAND]

เมื่อ **COMMAND** =

- A. หยุดนิ่ง
- B. วิ่งจากซ้ายไปขวา
- C. วิ่งจากขวาไปซ้าย
- D. เลื่อนจากบนลงล่าง
- E. เลื่อนจากล่างขึ้นบน
- Z. ลบข้อมูลใน EEPROM

เช่นต้องการ ให้แสดงคำว่า "Hello" และไม่ให้อักขรวิ่งซึ่งก็คือ Command "A" โดยส่งผ่าน COM 2 ทำได้โดยให้ไปที่ DOS แล้วเรียกโปรแกรมดังนี้

C:\DP750_2 A จากนั้น Enter

C:\DP750_2 A

ENTER Character (Max 8 char)

- Hello จากนั้น Enter

ถ้าทุกอย่างถูกต้องก็จะเห็น LED ติดเป็นคำว่า Hello

2. โปรแกรมบน Windows ซึ่งมีชื่อว่า **DP750W.EXE** สามารถทำงานได้บน Windows95/98/NT4.0 โดยมีหน้าต่างดังรูปที่ 8 การใช้งานก็เพียงแต่รันโปรแกรมนี้นี้ เมื่อโปรแกรมเปิดขึ้นมาแล้วก็เลือก COM PORT ที่ ET-DISPLAY 7X50 ต่ออยู่ จากนั้นก็พิมพ์ข้อความที่ต้องการแสดงลงไป จากนั้นเลือกรูปแบบการแสดงผลหรือ Command แล้วกดปุ่ม Send ถ้าทุกอย่างถูกต้องก็จะเห็น LED ติดเป็นตัวอักษรที่เราพิมพ์เข้าไปและตัว

อักขรยังถูกเก็บไว้ที่ EEPROM ภายในตัว CPU ด้วย นอกจากนั้นยังสามารถเปลี่ยนข้อความที่แสดงผลโดยการกด Switch ที่ ตัวบอร์ดได้ด้วย

หน้าที่ของปุ่มต่างๆ

ICON เป็นรูปอักขรพิเศษแต่ละจะมองไม่เห็นรูป ICON ในช่องพิมพ์ข้อความจนกว่าจะกดปุ่ม SEND รูปอักขรพิเศษจึงจะแสดงให้เห็นที่ ET-DISPLAY 7X50

SEND เป็นปุ่มที่ใช้ส่งข้อมูลหรือตัวอักขร

CLEAR มีไว้ Clear ข้อมูลที่อยู่ใน EEPROM ซึ่งจะใช้เวลาประมาณ 6 วินาที โดยขณะลบ LED สีเขียวจะติดค้างและเมื่อลบเสร็จ LED สีเขียวก็จะกระพริบตามปกติ

EXIT ออกจากโปรแกรม

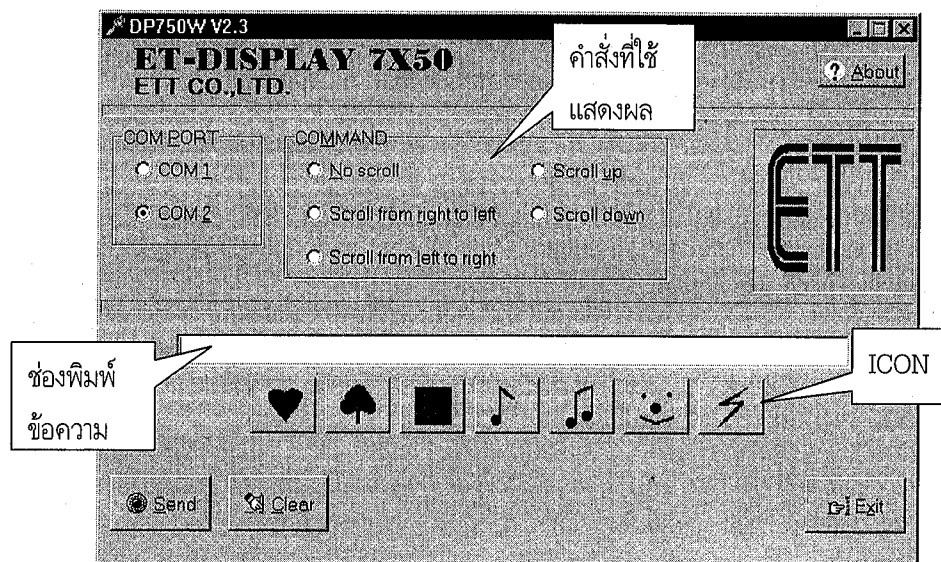
ABOUT เกี่ยวกับโปรแกรม

หมายเหตุ ในกรณีของคำสั่ง B และ C นั้นถ้าข้อความที่ส่งมีจำนวนตัวอักขรมากกว่า 64 ตัวจะไม่สามารถส่งได้ และจะมีข้อความเตือน ส่วนคำสั่ง A, D และ E นั้นถ้าข้อความที่ส่งมีจำนวนตัวอักขรมากกว่า 8 ตัวจะไม่สามารถส่งได้และจะมีข้อความเตือนเช่นกัน เพราะ 3 คำสั่งนี้จะมองเห็นเพียง 8 ตัวอักษรเท่านั้น

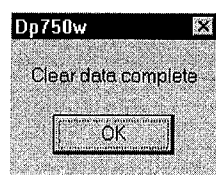
TIP การใช้คำสั่ง A, D และ E กับข้อความที่ยาวกว่า 8 ตัวอักษรสามารถทำได้โดยการแบ่งข้อความออกเป็นคำที่ไม่เกิน 8 ตัวอักษรและตัดตัวที่ไม่จำเป็นออกแต่ยังคงสามารถอ่านได้ แล้วนำมาแสดงต่อกันด้วยคำสั่ง A, D และ E ไปจนจบข้อความ เช่น ต้องการแสดงคำว่า "ET-DISPLAY 7X50 Dot FROM ETT" สามารถออกแบบการแสดงผลที่ต้องการใช้เป็นคำสั่ง A, D และ E ได้ใหม่เป็น

1	2	3	4	5	6	7	8	← ตำแหน่งของตัวอักษร
*		E	T		*			
D	I	S	P	L	A	Y		
7	X	5	0		D	o	t	
		F	R	O	M			
		E	T	T				

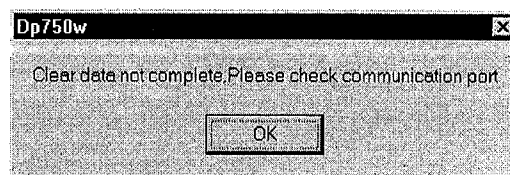
หากการจัดคำใหม่แล้วยังไม่สามารถอ่านได้ให้หลีกเลี่ยงไปใช้คำสั่ง C และ B หรือใช้ตัวย่อแทน ทั้งนี้ขึ้นกับความพอใจของผู้ใช้มากกว่า โดยไม่จำเป็นต้องยึดหลักตามนี้ก็ได



รูปที่ 8 แสดงหน้าต่างของโปรแกรม DP750W.EXE และส่วนหลักๆ



รูปที่ 9 แสดงข้อความเมื่อทำการลบข้อมูลใน EEPROM สำเร็จ



รูปที่ 10 แสดงข้อความเมื่อทำการลบข้อมูลใน EEPROM ไม่สำเร็จ

การประยุกต์ใช้งาน

- เป็นหน่วยแสดงผลทั่วไป
- ใช้แสดงข้อความในลิฟต์, โรงแรม, ห้างสรรพสินค้า, สถานีรถโดยสาร ฯลฯ
- ใช้แสดงยอดการผลิตของการผลิตและยังสามารถแสดงข้อความได้อีกด้วย
- ติดบนรถอย่างเช่น รถไฟฟ้า
- ประยุกต์ใช้กับระบบเข้าคิว
- ฯลฯ

การ TEST ET-DISPLAY 7X50

การ test นี้สามารถทำได้โดยการ กด Sw1 ค้างไว้แล้วจ่ายไฟเข้าตัวบอร์ด ถ้าทุกอย่างปกติจะเห็น LED ติดเป็นแถวยาววิ่งจากล่างขึ้นบนและจากบนลงล่าง จากนั้นก็จะติดหมดชั่วขณะ และสุดท้ายจะแสดงคำว่า "ETT" และจะแสดงรูปภาพกราฟฟิคคือ รูปผีเสื้อและรูป ICON ต่างๆ

แนวทางการพัฒนาต่อ

หากผู้ใช้งานต้องการที่จะพัฒนาขีดความสามารถของ ET-DISPLAY 7X50 ก็สามารถทำได้โดยการนำซอร์ทโค้ด(DP750V2.ASM สำหรับ 89S8252 และ DP750V1.ASM สำหรับ 89C52)ที่ให้ไปเปลี่ยนแปลงหรือแก้ไขได้ตามความต้องการ เช่น เพิ่มการแสดงผลภาษาไทยหรือฟลิตรูปต่างๆที่ต้องการตลอดจนเพิ่มฟังก์ชันการแสดงผล แต่ในที่นี้จะแนะนำการเปลี่ยน Font ของตัวอักษรหรือสัญลักษณ์อื่นๆ ซึ่งทำได้โดยการออกแบบด้วยมือหรือใช้โปรแกรม FONT DESIGNER V1.0 (FDV10.EXE) ซึ่งทำได้โดยเมื่อรันโปรแกรม ขึ้นมาแล้วจะมีหน้าต่างรูปที่ 11 จากนั้นให้ใช้ Mouse คลิกที่เซลล์ขนาด 7X6 ซึ่งเมื่อคลิกแล้วจะเป็นสีแดงและตัวเลขค่าฐานสิบหกด้านล่างจะเปลี่ยนไปด้วย เมื่อได้รูปแบบที่ต้องการแล้วให้เอาค่าเลขฐานสิบหกไปใส่ในโปรแกรม DP750Vx.ASM หลังตำแหน่งลาเบล ASCII_TABLE: ซึ่งของเดิมจะเริ่มจาก 20H(ASCII) ซึ่งก็คือ Space เช่นต้องการออกแบบตัว A ที่มีรูปร่างแปลกๆ สมมติว่าได้ค่าเป็น 22H,33H,44H,55H,66H,77H จากนั้นก็นำเอาไปใส่แทนที่ของตัว A ตัวเดิมซึ่งคือรหัส ASCII ที่ 41H จากนั้นให้ทำการ Assembler โปรแกรม DP750Vx.ASM อีกครั้งจากนั้นก็โปรแกรมลงตัว CPU เมื่อเราส่งข้อความที่มีตัว A ไปด้วยก็จะปรากฏตัว A ในแบบที่เราออกแบบไว้ สำหรับตัวอื่นๆถ้าต้องการเปลี่ยนก็ให้ทำเช่นเดียวกัน ที่กล่าวมานี้ให้ทำเฉพาะตั้งแต่ 20H-7EH เท่านั้น ส่วนที่ 80H-FFH นั้นสามารถใช้เป็นรูปแบบอย่างอื่นก็ได้ เช่นใช้ทำเป็นภาษาไทยโดยเราต้องรู้รหัส ASCII ภาษาไทยก่อน เช่นออกแบบตัว ก ได้ค่าเลขฐานสิบหกเป็น 2FH,50H,40H,40H,3FH,00H และอยากให้ตัว ก มีรหัส ASCII เป็น A1H ก็เอาค่าเลขฐานสิบหกไปใส่ที่ A1H ได้เลย แต่ในที่นี้เป็นอักขรพิเศษสำหรับ DEMO

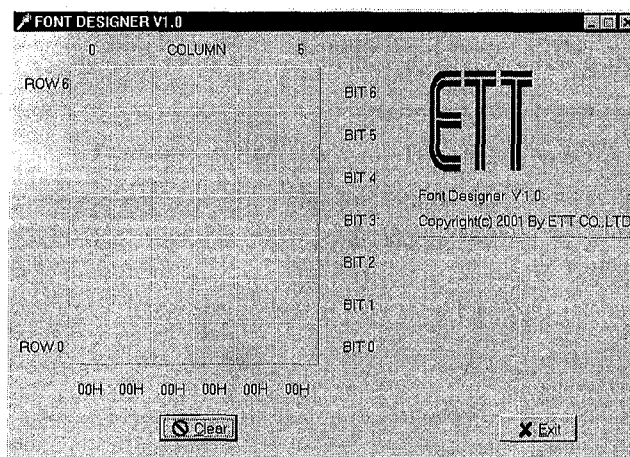
ASCII_TABLE:

20H	080H,080H,080H,080H,080H,080H

80H	000H,000H,000H,000H,000H,000H
81H	000H,000H,000H,000H,000H,000H

A1H	02FH,050H,040H,040H,03FH,000H ; ก

รูปที่ 11 หน้าตาของโปรแกรม FDV10.EXE



การเปลี่ยนความเร็วในการแสดงผล

การเปลี่ยนความเร็วในการแสดงผลให้เปลี่ยนที่โปรแกรม DP750Vx.ASM ในส่วนของ Constant variable ที่อยู่ตอนต้นของโปรแกรมโดยของเดิมกำหนดไว้ดังนี้

```
=====
; Constant variable for speed display .You can be change
=====
```

```
LOOP_F EQU 7 ; DELAY FOR FIX SCAN
LOOP_R EQU 6 ; SPEED FOR SCROLLING
LOOP_L EQU 7 ; SPEED FOR SCROLLING
LOOP_U EQU 10 ; SPEED FOR SCROLLING UP AND DONW
FIX_LOOP EQU 30 ; DELAY FOR NO SCROLL IN UP AND DONW SCAN
```

อธิบาย

LOOP_F เป็นค่าหน่วยเวลาในการแสดงข้อความขณะหยุดนิ่งก่อนที่จะแสดงข้อความถัดไป(ถ้ามี)
LOOP_R เป็นค่าหน่วยเวลาในการเลื่อนข้อความจากขวาไปซ้าย
LOOP_L เป็นค่าหน่วยเวลาในการเลื่อนข้อความจากซ้ายไปขวา
LOOP_U เป็นค่าหน่วยเวลาในการเลื่อนข้อความจากบนลงล่างหรือจากล่างขึ้นบน
FIX_LOOP เป็นค่าหน่วยเวลาในการแสดงข้อความขณะหยุดนิ่งก่อนที่จะเลื่อนข้อความขึ้นหรือลง

โดยค่าต่างๆเหล่านี้ถ้ามากจะช้าลงหรือนานขึ้นนั่นเอง

โดยเมื่อค่ามากขึ้นจะทำให้การเคลื่อนที่ของข้อความช้าลง แต่สำหรับ LOOP_F จะไม่เห็นผลเพราะตัวอักษรอยู่กับที่ เมื่อทำการเปลี่ยนแล้วให้ Assembler โปรแกรมใหม่ แล้วจึงนำไปโปรแกรมลง CPU

การเปลี่ยน Baud rate ของ Port RS232

การเปลี่ยนความเร็วของการสื่อสารนั้นสามารถทำได้โดยให้เปลี่ยนค่าของรีจิสเตอร์ TH1 และ PCON ในโปรแกรมย่อย INIT_SER

```
=====
; INIT UART
=====
INIT_SER: MOV TMOD,#00100001B ; time1 mode2, TIMER 0 MODE 1 FOR DIM
MOV SCON,#01010000B ; mode1 serial port
MOV A,#80H ; SMOD =1=Double baud rate
MOV PCON,A ; SMOD = Bit 7 in register PCON
MOV TH1,#0FBH ; BAUD RATE = 19200,N,8,1
```

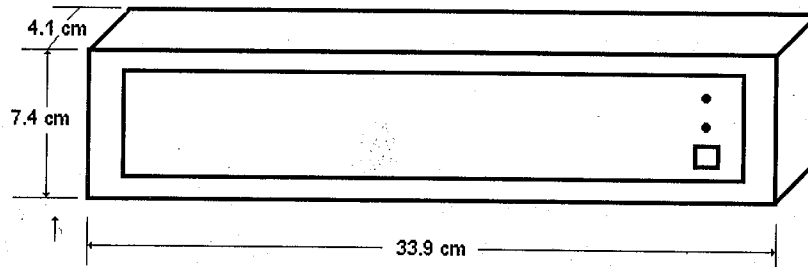
Baud rate	SMOD	TH1
19200	1	FBH
9600	0	FBH
4800	0	F6H
2400	0	ECH
1200	0	D8H

หลังจากเปลี่ยนแปลงที่ Source code ซึ่งทางอีทีที่ได้ให้ไปด้วยแล้ว โดยมีทั้งโปรแกรมที่ใช้กับ 89C52 (DP750V1.ASM) และ 89S8252(DP750V2.ASM) ซึ่งเมื่อแก้ไขเสร็จแล้วก็ให้ทำการแอสเซมเบลอร์โดยใช้โปรแกรม SXA51.EXE ดังนี้

C:\SXA51 -L DP750V2.ASM

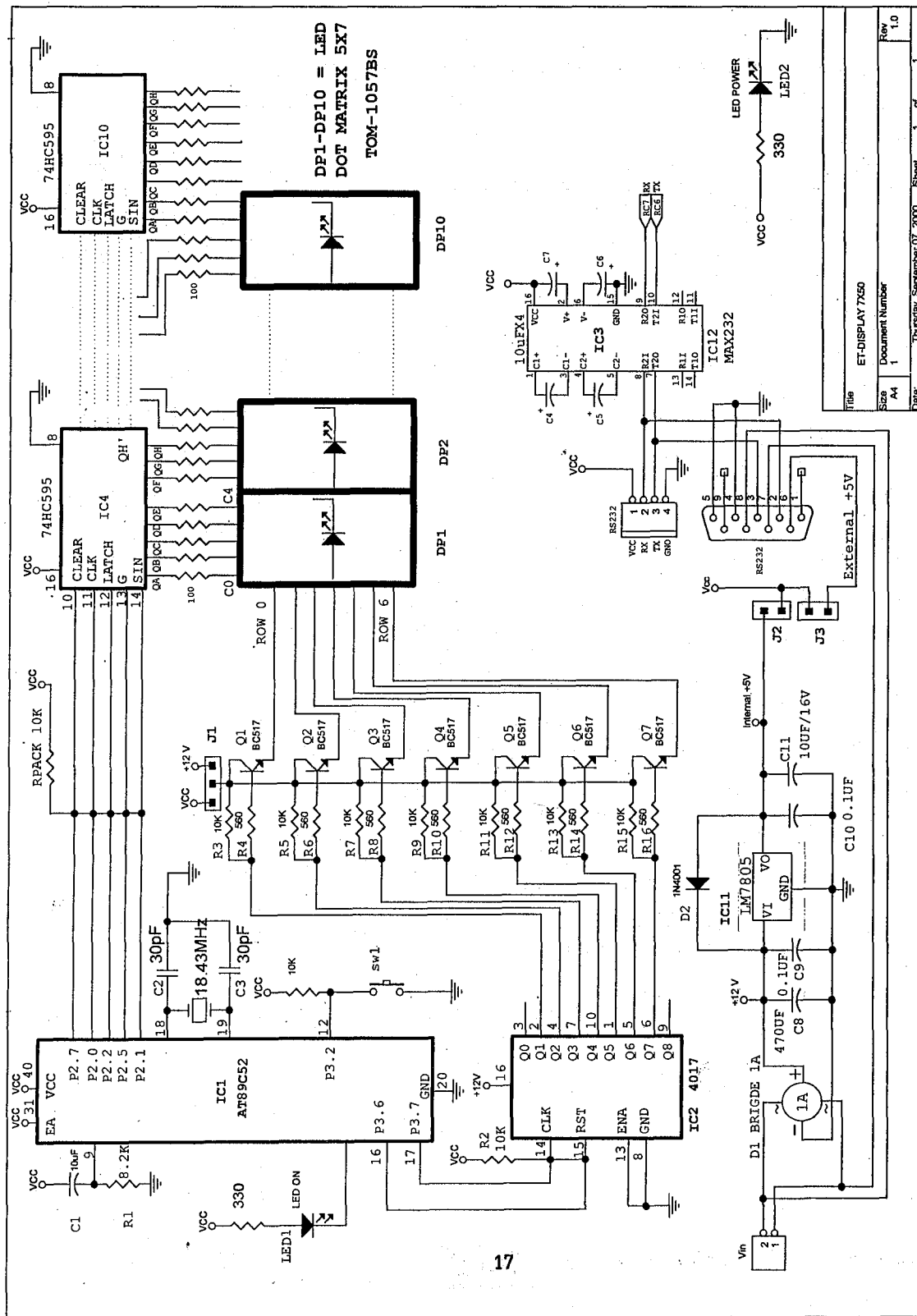
เมื่อแอสเซมเบลอร์ผ่านแล้วจะได้ไฟล์ 2 ไฟล์ดังนี้คือ DP750V2.LST และ DP750V2.HEX จากนั้นให้นำไฟล์ DP750V2.HEX ไปโปรแกรมลง CPU ซึ่งกรณีนี้จะต้องเป็นเบอร์ 89S8252 เท่านั้นเพราะเบอร์นี้มี EEPROM 2k ในตัวเองซึ่งเบอร์นี้เหมาะกับการใช้งานแบบ Stand alone เช่นป้ายโฆษณาสินค้า แต่สำหรับ 89C52 นั้นต้องใช้กับไฟล์ DP750V1.ASM เท่านั้นและไปสามารถเก็บข้อความได้ในขณะไฟดับซึ่งเหมาะสำหรับงานที่ต้อง การเปลี่ยนข้อความบ่อยมาก เช่นแสดงยอดการผลิต หรือหมายเลขเข้าคิว

DIMENSION



ตารางรหัส ASCII

DEC	HEX	CHAR	DEC	HEX	CHAR	DEC	HEX	CHAR	DEC	HEX	CHAR	DEC	HEX	CHAR	DEC	HEX	CHAR	DEC	HEX	CHAR	DEC	HEX	CHAR
0	0		32	20		64	40	@	96	60	'	128	80	Ç	160	A0	á	192	C0	À	224	E0	α
1	1	☉	33	21	!	65	41	A	97	61	a	129	81	ü	161	A1	í	193	C1	Á	225	E1	β
2	2	●	34	22	"	66	42	B	98	62	b	130	82	é	162	A2	ó	194	C2	Â	226	E2	Γ
3	3	▼	35	23	#	67	43	C	99	63	c	131	83	â	163	A3	ú	195	C3	Ã	227	E3	π
4	4	◆	36	24	\$	68	44	D	100	64	d	132	84	ä	164	A4	ñ	196	C4	Ä	228	E4	Σ
5	5	♣	37	25	%	69	45	E	101	65	e	133	85	å	165	A5	Ï	197	C5	Å	229	E5	σ
6	6	♠	38	26	&	70	46	F	102	66	f	134	86	ä	166	A6	■	198	C6	Æ	230	E6	μ
7	7	•	39	27	'	71	47	G	103	67	g	135	87	ç	167	A7	ø	199	C7	Ç	231	E7	τ
8	8	■	40	28	(72	48	H	104	68	h	136	88	ê	168	A8	ù	200	C8	È	232	E8	ϕ
9	9	○	41	29)	73	49	I	105	69	i	137	89	ë	169	A9	ü	201	C9	É	233	E9	θ
10	A	■	42	2A	*	74	4A	J	106	6A	j	138	8A	è	170	AA	¬	202	CA	Ê	234	EA	Ω
11	B	♠	43	2B	+	75	4B	K	107	6B	k	139	8B	ÿ	171	AB	½	203	CB	Ë	235	EB	δ
12	C	♀	44	2C	,	76	4C	L	108	6C	l	140	8C	í	172	AC	¾	204	CC	Ì	236	EC	∞
13	D	♫	45	2D	-	77	4D	M	109	6D	m	141	8D	ì	173	AD	ı	205	CD	Í	237	ED	ø
14	E	♫	46	2E	.	78	4E	N	110	6E	n	142	8E	Ï	174	AE	«	206	CE	Î	238	EE	€
15	F	♠	47	2F	/	79	4F	O	111	6F	o	143	8F	À	175	AF	»	207	CF	Ï	239	EF	η
16	10	▶	48	30	0	80	50	P	112	70	p	144	90	É	176	B0	☼	208	D0	Ĳ	240	F0	≡
17	11	◀	49	31	1	81	51	Q	113	71	q	145	91	æ	177	B1	☼	209	D1	Ŧ	241	F1	±
18	12	İ	50	32	2	82	52	R	114	72	r	146	92	Æ	178	B2	■	210	D2	Ŧ	242	F2	≥
19	13	İ	51	33	3	83	53	S	115	73	s	147	93	ø	179	B3		211	D3	Ŧ	243	F3	≤
20	14	¶	52	34	4	84	54	T	116	74	t	148	94	ö	180	B4	ı	212	D4	Ŧ	244	F4	ƒ
21	15	§	53	35	5	85	55	U	117	75	u	149	95	ö	181	B5	ı	213	D5	Ŧ	245	F5	ƒ
22	16	■	54	36	6	86	56	V	118	76	v	150	96	û	182	B6	ı	214	D6	Ŧ	246	F6	÷
23	17	İ	55	37	7	87	57	W	119	77	w	151	97	ü	183	B7	ı	215	D7	Ŧ	247	F7	≈
24	18	İ	56	38	8	88	58	X	120	78	x	152	98	ÿ	184	B8	ı	216	D8	Ŧ	248	F8	•
25	19	İ	57	39	9	89	59	Y	121	79	y	153	99	ÿ	185	B9	ı	217	D9	Ŧ	249	F9	•
26	1A	→	58	3A	:	90	5A	Z	122	7A	z	154	9A	ÿ	186	BA	ı	218	DA	Ŧ	250	FA	•
27	1B	→	59	3B	;	91	5B	[123	7B	(155	9B	ÿ	187	BB	ı	219	DB	Ŧ	251	FB	√
28	1C	→	60	3C	<	92	5C	\	124	7C		156	9C	ÿ	188	BC	ı	220	DC	Ŧ	252	FC	ˆ
29	1D	→	61	3D	=	93	5D]	125	7D)	157	9D	ÿ	189	BD	ı	221	DD	Ŧ	253	FD	ˆ
30	1E	→	62	3E	>	94	5E	^	126	7E	~	158	9E	ÿ	190	BE	ı	222	DE	Ŧ	254	FE	ˆ
31	1F	→	63	3F	?	95	5F	_	127	7F	Δ	159	9F	f	191	BF	ı	223	DF	Ŧ	255	FF	ˆ



Features

- Compatible with MCS-51™ Products
- 8K Bytes of In-System Reprogrammable Downloadable Flash Memory
 - SPI Serial Interface for Program Downloading
 - Endurance: 1,000 Write/Erase Cycles
- 2K Bytes EEPROM
 - Endurance: 100,000 Write/Erase Cycles
- 4.0V to 6V Operating Range
- Fully Static Operation: 0 Hz to 24 MHz
- Three-Level Program Memory Lock
- 256 x 8-bit Internal RAM
- 32 Programmable I/O Lines
- Three 16-bit Timer/Counters
- Nine Interrupt Sources
- Programmable UART Serial Channel
- SPI Serial Interface
- Low Power Idle and Power Down Modes
- Interrupt Recovery From Power Down
- Programmable Watchdog Timer
- Dual Data Pointer
- Power Off Flag

Description

The AT89S8252 is a low-power, high-performance CMOS 8-bit microcomputer with 8K bytes of Downloadable Flash programmable and erasable read only memory and 2K bytes of EEPROM. The device is manufactured using Atmel's high density nonvolatile memory technology and is compatible with the industry standard 80C51 instruction set and pinout. The on-chip Downloadable Flash allows the program memory to be reprogrammed in-system through an SPI serial interface or by a conventional non-volatile memory programmer. By combining a versatile 8-bit CPU with Downloadable Flash on a monolithic chip, the Atmel AT89S8252 is a powerful microcomputer which provides a highly flexible and cost effective solution to many embedded control applications.

The AT89S8252 provides the following standard features: 8K bytes of Downloadable Flash, 2K bytes of EEPROM, 256 bytes of RAM, 32 I/O lines, programmable watchdog timer, two Data Pointers, three 16-bit timer/counters, a six-vector two-level interrupt architecture, a full duplex serial port, on-chip oscillator, and clock circuitry. In addition, the AT89S8252 is designed with static logic for operation down to zero frequency and supports two software selectable power saving modes. The Idle Mode stops the CPU while allowing the RAM, timer/counters, serial port, and interrupt system to continue functioning. The Power Down Mode saves the RAM contents but freezes the oscillator, disabling all other chip functions until the next interrupt or hardware reset.

The Downloadable Flash can be changed a single byte at a time and is accessible through the SPI serial interface. Holding RESET active forces the SPI bus into a serial programming interface and allows the program memory to be written to or read from unless Lock Bit 2 has been activated.



8-Bit Microcontroller with 8K Bytes Flash

AT89S8252

0401D-A-12/97

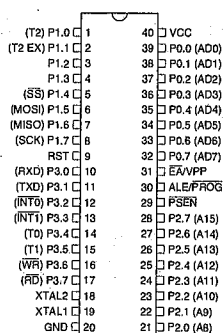


4-105

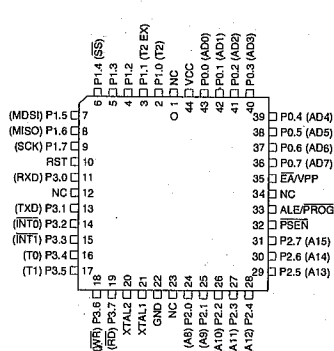


Pin Configurations

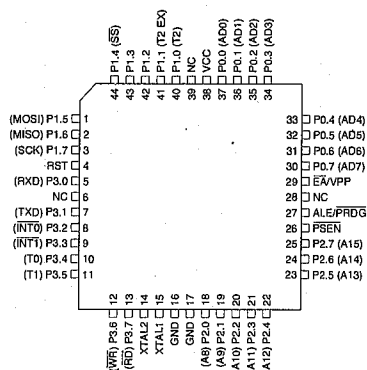
PDIP



PLCC



PQFP/TQFP



Pin Description

V_{CC}
Supply voltage.

GND
Ground.

Port 0

Port 0 is an 8-bit open drain bidirectional I/O port. As an output port, each pin can sink eight TTL inputs. When 1s are written to port 0 pins, the pins can be used as high-impedance inputs.

Port 0 can also be configured to be the multiplexed low-order address/data bus during accesses to external program and data memory. In this mode, P0 has internal pullups.

Port 0 also receives the code bytes during Flash programming and outputs the code bytes during program verification. External pullups are required during program verification.

Port 1

Port 1 is an 8-bit bidirectional I/O port with internal pullups. The Port 1 output buffers can sink/source four TTL inputs. When 1s are written to Port 1 pins, they are pulled high by the internal pullups and can be used as inputs. As inputs, Port 1 pins that are externally being pulled low will source current (I_{IL}) because of the internal pullups.

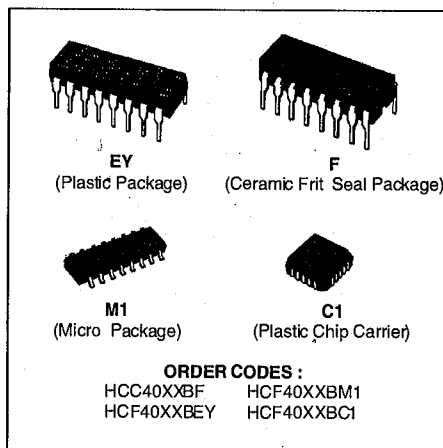
Some Port 1 pins provide additional functions. P1.0 and P1.1 can be configured to be the timer/counter 2 external count input (P1.0/T2) and the timer/counter 2 trigger input (P1.1/T2EX), respectively.

COUNTERS/DIVIDERS

**4017B DECADE COUNTER WITH 10
DECODED OUTPUTS**

**4022B OCTAL COUNTER WITH 8
DECODED OUTPUTS**

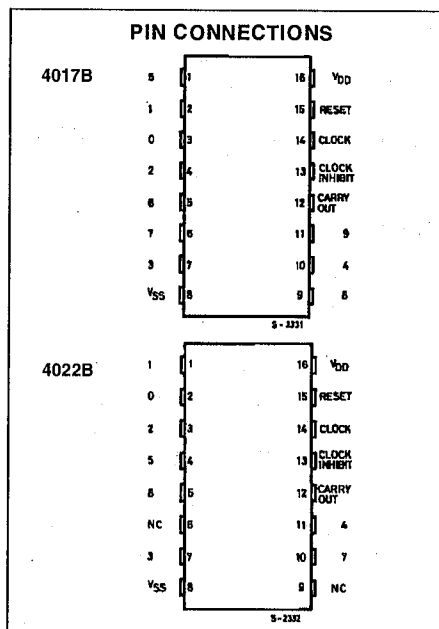
- FULLY STATIC OPERATION
- MEDIUM SPEED OPERATION-12MHz (typ.) AT $V_{DD} = 10V$
- STANDARDIZED SYMMETRICAL OUTPUT CHARACTERISTICS
- QUIESCENT CURRENT SPECIFIED TO 20V FOR HCC DEVICE
- INPUT CURRENT OF 100nA AT 18V AND 25°C FOR HCC DEVICE
- 100% TESTED FOR QUIESCENT CURRENT
- 5V, 10V, AND 15V PARAMETRIC RATINGS
- MEETS ALL REQUIREMENTS OF JEDEC TENTATIVE STANDARD N° 13A, "STANDARD SPECIFICATIONS FOR DESCRIPTION OF "B" SERIES CMOS DEVICES"



DESCRIPTION

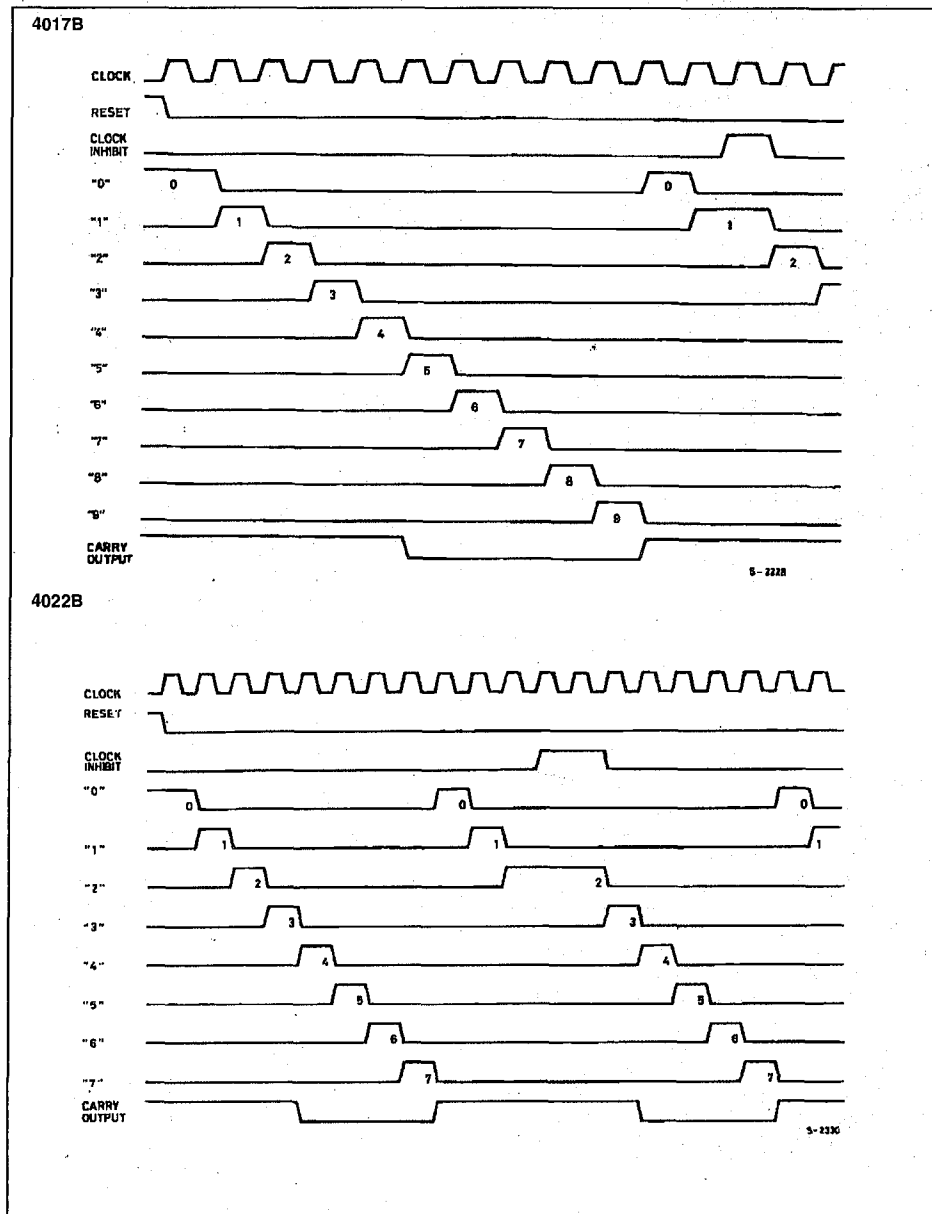
The **HCC4017B/4022B** (extended temperature range) and **HCF4017B/4022B** (intermediate temperature range) are monolithic integrated circuits, available in 16-lead dual in-line plastic or ceramic package and plastic micro package.

The **HCC/HCF4017B** and **HCC/HCF4022B** are 5-stage and 4-stage Johnson counters having 10 and 8 decoded outputs, respectively. Inputs include a **CLOCK**, a **RESET**, and a **CLOCK INHIBIT** signal. Schmitt trigger action in the **CLOCK** input circuit provides pulse shaping that allows unlimited clock input pulse rise and fall times. These counters are advanced one count at the positive clock signal transition if the **CLOCK INHIBIT** signal is low. Counter advancement via the clock line is inhibited when the **CLOCK INHIBIT** signal is high. A high **RESET** signal clears the counter to its zero count. Use of the Johnson decade-counter configuration permits high-speed operation, 2-input decimal-decode gating, and spike-free decoded outputs. Anti-lock gating is provided, thus assuring proper counting sequence. The decoded outputs are normally low and go high only at their respective decoded time slot. Each decoded output remains high for one full clock cycle. A **CARRY-OUT** signal completes one



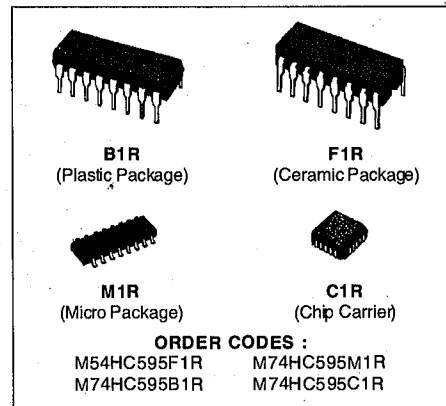
HCC/HCF4017B/4022B

TIMING DIAGRAMS



8 BIT SHIFT REGISTER WITH OUTPUT LATCHES (3 STATE)

- HIGH SPEED
 $f_{MAX} = 55 \text{ MHz (TYP.) AT } V_{CC} = 5 \text{ V}$
- LOW POWER DISSIPATION
 $I_{CC} = 4 \mu\text{A (MAX.) AT } T_A = 25^\circ\text{C}$
- HIGH NOISE IMMUNITY
 $V_{NIH} = V_{NIL} = 28\% V_{CC} \text{ (MIN.)}$
- OUTPUT DRIVE CAPABILITY
 15 LSTTL LOADS FOR QA TO QH
 10 LSTTL LOADS FOR QH'
- SYMMETRICAL OUTPUT IMPEDANCE
 $I_{OH} = I_{OL} = 6 \text{ mA (MIN.) FOR QA TO QH}$
 $I_{OH} = I_{OL} = 4 \text{ mA (MIN.) FOR QH'}$
- BALANCED PROPAGATION DELAYS
 $t_{PLH} = t_{PHL}$
- WIDE OPERATING VOLTAGE RANGE
 $V_{CC} \text{ (OPR)} = 2 \text{ V TO } 6 \text{ V}$
- PIN AND FUNCTION COMPATIBLE
 WITH LSTTL 54/74LS595



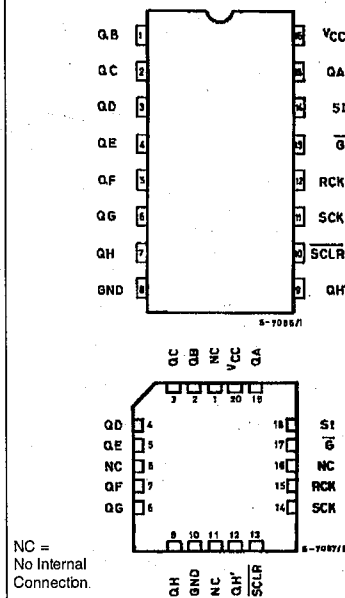
DESCRIPTION

The M54/74HC595 is a high speed CMOS 8-BIT SHIFT REGISTERS/OUTPUT LATCHES (3-STATE) fabricated in silicon C²MOS technology. It has the same high speed performance of LSTTL combined with true CMOS low power consumption. This device contains an 8-bit serial-in, parallel-out shift register that feeds an 8-bit D-type storage register. The storage register has 8 3-STATE outputs. Separate clocks are provided for both the shift register and the storage register.

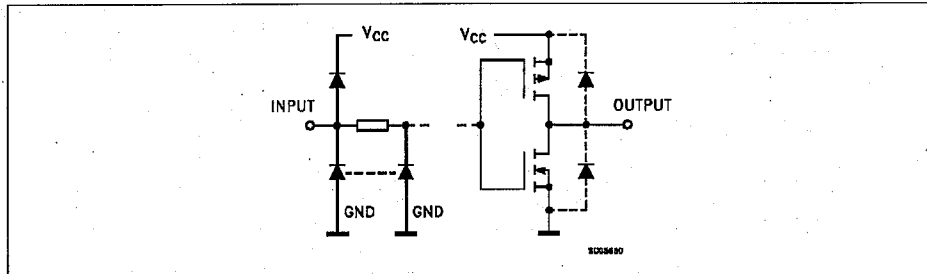
The shift register has a direct-overriding clear, serial input, and serial output (standard) pins for cascading. Both the shift register and storage register use positive-edge triggered clocks. If both clocks are connected together, the shift register state will always be one clock pulse ahead of the storage register.

All inputs are equipped with protection circuits against static discharge and transient excess voltage.

PIN CONNECTIONS (top view)



INPUT AND OUTPUT EQUIVALENT CIRCUIT

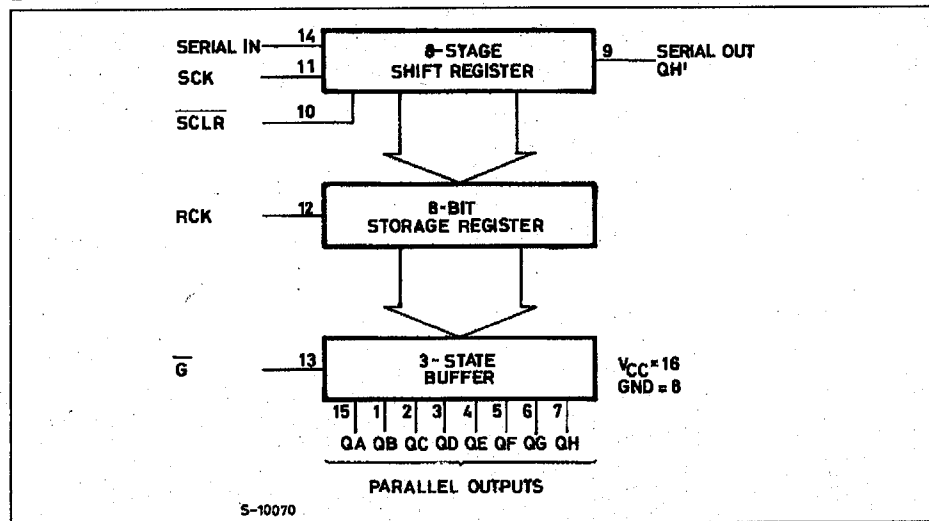


TRUTH TABLE

INPUTS					OUTPUT
SI	SCK	SCLR	RCK	\overline{G}	
X	X	X	X	H	QA THRU QH OUTPUTS DISABLE
X	X	X	X	L	QA THRU QH OUTPUTS ENABLE
X	X	L	X	X	SHIFT REGISTER IS CLEARED
L		H	X	X	FIRST STAGE OF S.R. BECOMES "L" OTHER STAGES STORE THE DATA OF PREVIOUS STAGE, RESPECTIVELY
H		H	X	X	FIRST STAGE OF S.R. BECOMES "H" OTHER STAGES STORE THE DATA OF PREVIOUS STAGE, RESPECTIVELY
X		H	X	X	STATE OF S.R IS NOT CHANGED
X	X	X		X	S.R. DATA IS STORED INTO STORAGE REGISTER
X	X	X		X	STORAGE REGISTER STATE IS NOT CHANGED

X: DON'T CARE

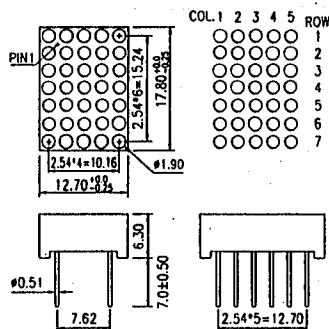
LOGIC DIAGRAM



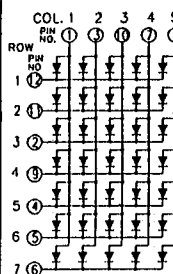


PACKAGE DIMENSION INTERNAL CIRCUIT DIAGRAM

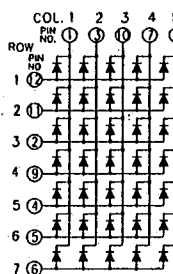
TOM-757Ax/Bx Series



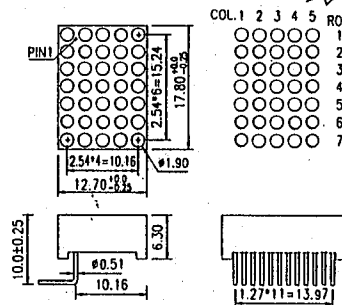
TOM-757Ax



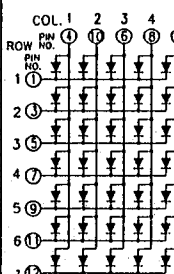
TOM-757Bx



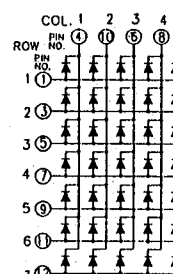
TOM-757Cx/Dx Series



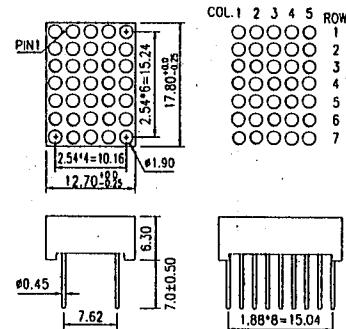
TOM-757Cx



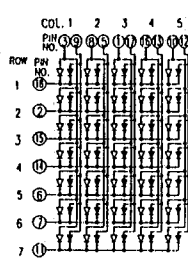
TOM-757Dx



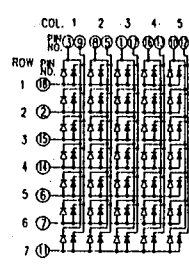
TOM-757AEG/BEG Series



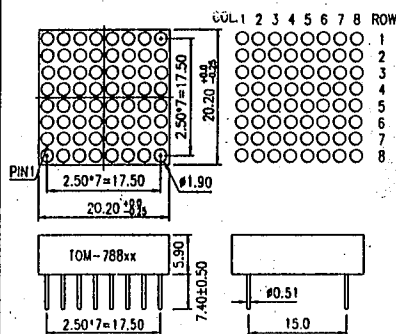
TOM-757AEG



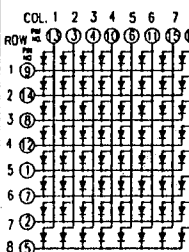
TOM-757BEG



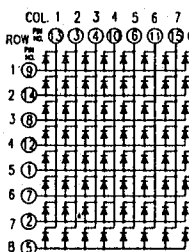
TOM-788Ax/Bx Series



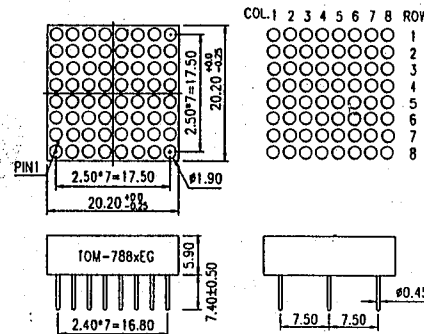
TOM-788Ax



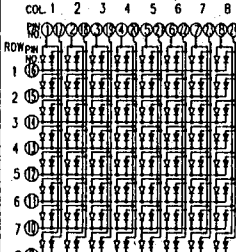
TOM-788Bx



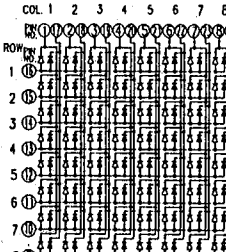
TOM-788AEG/BEG Series



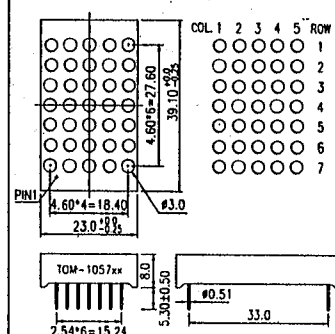
TOM-788AEG



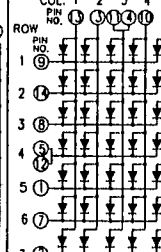
TOM-788BEG



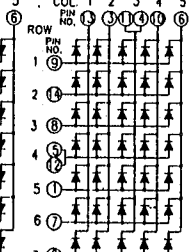
TOM-1057Ax/Bx Series



TOM-1057Ax



TOM-1057Bx



—X— GREEN
—Y— ORANGE

24

—X— GREEN
—Y— ORANGE

