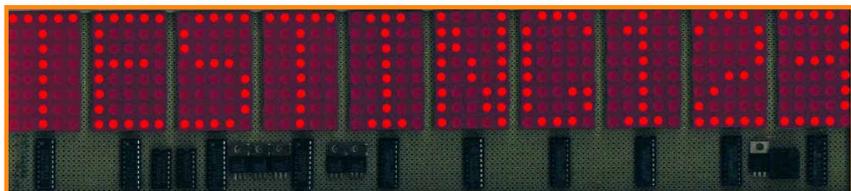


Make Your Own Printer Port

LED ARRAY SIGN

For Around \$50.00

by Robert Davis



I have always wanted to play with some LED arrays. So, when Electronics Goldmine advertised them at 10 for only \$18.00, I quickly bought some. Their part number G7295 gets you 40 bright red LEDs arranged as five columns by eight rows. These displays are over two inches tall by 1-1/2 inches wide. They can be used for all kinds of fun

projects and fun devices.

There were two things that I planned to make using the LED arrays. One was an "ON THE AIR" sign, and the other a multiple channel LED VU meter. The VU meter had already been made using a computer as the Printer Port VU meter. The

This article is about a programmable sign that connects to a computer's printer port. It is one of the neatest, most versatile, and most useful LED array projects you'll come across.

problems with it were that the monitor was taking up too much room in the sound control booth and that the computer seemed like overkill for a 16-channel VU meter.

Somehow the sign idea grew into two versions: a fixed one and a programmable one using a computer. The VU meter also grew into two versions: one that uses four of the five columns in each display and the other uses all five columns.

This article is about the programmable sign that connects to a computer's printer port. It is the neatest, most versatile, and most useful of the LED array projects.

Initially, the easiest method of lighting up the arrays that comes to mind would be to arrange them as one giant matrix and pass a rather large current through them. To see the problems that would arise, con-

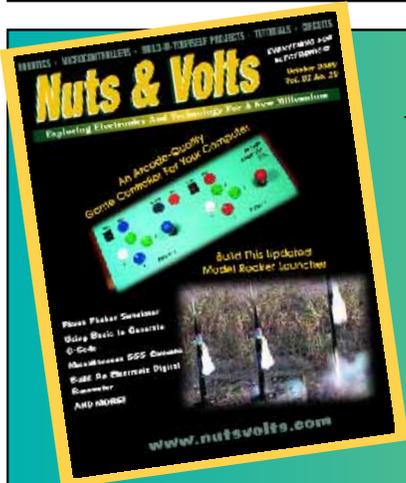
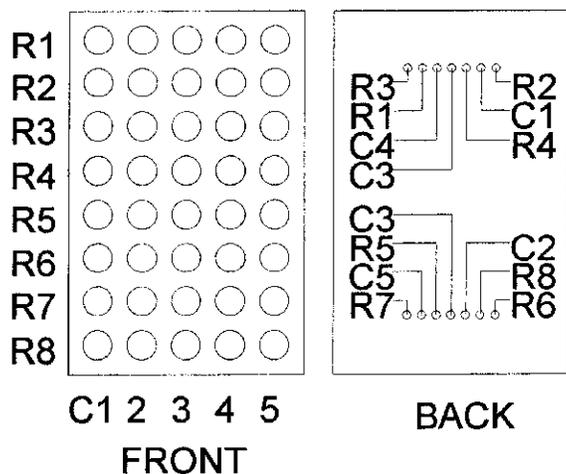
sider 10 of the eight by five arrays arranged as an eight row by 50 column matrix. A counter and decoder could rapidly select the 50 columns to make it appear like they are all lit. Since each is only lit 1/50 of the time, it would need 50 times the normal current during that time to reach normal brightness. Considering 10 mA per LED as sufficient, then we would need eight (LEDs in the column) times 10 milliamp times 50 or a total of four amps!

The current-limiting resistors would need to be large and, if the counter stopped, any lit LED would likely smoke. If it was instead broken down into 10 separate LED arrays, the same current would be 1/5 of four amps or a much more manageable 8/10 of an amp.

Even 8/10 of an amp exceeds the maximum rating of ordinary transistors like the 2N2222. I have tried using 2N2222s, but they get very hot. For the columns, my preference is to use a more powerful driver like the TIP120 or TIP125. These Darlington power transistors can easily drive 10 or more LED array columns with all LEDs lit and only get warm to the touch. If they do get too hot, a small heatsink can easily be attached, but none proved necessary in any of my designs.

Some designers will use individual drivers for

LED ARRAY LTP2058AHR-01



Shameless Plug

This article was reprinted from the May 2000 issue of *Nuts & Volts Magazine*. If you aren't familiar with *Nuts & Volts* and are interested in electronics, we invite you to stop by our website and see what it's all about. Heck, we'll even send you a sample copy for free if you haven't seen it before. And while you're there, why not subscribe, or buy electronics books, or visit our electronics forum, or read articles, or build a project?

(Even if you have seen it, stop by anyway, we update stuff every week!!)

www.nutsvolts.com



both the rows and columns for every LED array. Neither is required. On the rows, something like the ULN2803 driver would be used. But is that powerful of a driver really necessary? For instance, a 74LS373 can easily deliver 30 mA per output. If that is divided between five LEDs in the row, it gives 6 mA of current per LED. This will generally provide sufficient brightness with no high current driver.

My conclusion is that it is practical to share column drivers between multiple arrays, but not practical to share row drivers between multiple arrays. That means that a separate 74LS373 will be required for each LED array that is used. These drivers only deliver 30 mA, so there is little chance of damage to the LEDs if something goes wrong. The column drivers can be controlled by a counter that counts to 10 and then resets. A 4017 will work fine for this purpose. The 4017 delivers only about 1 mA, so a TIP120 driver transistor will be used to supply the needed current. The Darlington transistor reduces the load on the 4017 to where it can easily handle it.

Another problem is finding a circuit board big enough to hold 10 of the LED arrays.

Jameco has one that is 4-1/2 inches by 17 inches. That is enough room to fit 10 displays with about 2/10 of an inch between each of them. If you are into graphics, the space between them can be omitted, but the software to produce the graphics would be much more complicated than what I have provided.

The printer port sign is relatively simple to make. Most designs that I have seen use a latch to hold the next byte to display, then a latch for the currently displayed byte, and a current driver for each LED array. With some simple tricks, I was able to do all three of these things with just one IC — a 74LS373. I have tested 74F373, 74S373, 74HCT373, and the 74LS373, and did not notice any difference in brightness or performance. The 74XX374 eight-bit latches do create a timing difference that can be corrected in software.

The trick to doing all of these things with one

latch is to turn the display off while the latches are being loaded. This prevents a dim echo of the next or previous byte from appearing. A 4017 is used, but only every other output is connected to turn on the columns. This way the 10 latches can be loaded, then the columns turned on, then turned back off, and then load the latches with the next 10 bytes.

The current-limiting resistors for the printer port sign need to be smaller because there is only five volts to light up the LEDs instead of 10 or 12. This will prevent frying the 74LS373's output transistors. A value of 47 ohms works best,

Qty	Part	Number/Price	Total	Source
10	LED arrays	G7295 10/\$18	\$18.00	Electronic Goldmine 1-800-445-0697
1	Board 4x17	37604 10.95	10.95	Jameco 1-800-831-4242
10	74LS373	47600 .39	3.90	Jameco
2	CD4017	12749 .35	.70	Jameco
1	LM7805	51262 .29	.29	Jameco
5	TIP120	32993 .65	3.25	Jameco
1	1.5A bridge	145429 .59	.59	Jameco
1	1000uF/16V	30015 .19	.19	Jameco
1	470uF/16V	93817 .19	.19	Jameco
1	9 VAC	119204 4.95	4.95	Jameco
80	47 ohm 1/4W	29946 100/.89	.89	Jameco
3	2.2K 1/4W	29946 100/.89	.89	Jameco
10	20 pin sockets	112248 .12	1.20	Jameco
2	16 pin sockets	112221 .12	.24	Jameco
1	26 pin header	53495 .35	.35	Jameco
1	25 pin Male to header socket cable (from parts pile)			
Total cost of parts purchased			\$46.58	

Parts List for the Printer Port Sign

This is the software listing for a sign that will display four lines of text.

```
CLS : PRINT "Printer port sign, Copyright 1999 BY Robert J Davis"
'INPUT "Use Printer Port Number: ", lpt
'Hardcoded to lpt2 for higher speed - needed for a 486.
'Make these changes for using other printer ports;
IF lpt = 2 THEN dout = &H378: cout = &H37A: cin = &H379
IF lpt = 1 THEN dout = &H3BC: cout = &H3BE: cin = &H3BD
IF lpt = 3 THEN dout = &H278: cout = &H27A: cin = &H279
PRINT "Press q to quit displaying sign _____", ""
INPUT "What do you want line 1 to say? ", line1$
INPUT "What do you want line 2 to say? ", line2$
INPUT "What do you want line 3 to say? ", line3$
INPUT "What do you want line 4 to say? ", line4$
new = 1
getline:
IF new > 300 THEN new = 1
IF new = 1 THEN message$ = line1$
IF new = 100 THEN message$ = line2$
IF new = 200 THEN message$ = line3$
IF new = 300 THEN message$ = line4$
DIM array, byte(50)
FOR I = 1 TO 10
  letter$ = MID$(message$, I, 1)
  letter$ = UCASE$(letter$)
  SELECT CASE letter$
    CASE IS = "A": RESTORE a
    CASE IS = "B": RESTORE b
    CASE IS = "C": RESTORE c
    CASE IS = "D": RESTORE d
    CASE IS = "E": RESTORE e
    CASE IS = "F": RESTORE f
    CASE IS = "G": RESTORE g
    CASE IS = "H": RESTORE h
    CASE IS = "I": RESTORE i
    CASE IS = "J": RESTORE j
    CASE IS = "K": RESTORE k
    CASE IS = "L": RESTORE l
    CASE IS = "M": RESTORE m
    CASE IS = "N": RESTORE n
    CASE IS = "O": RESTORE o
    CASE IS = "P": RESTORE p
    CASE IS = "Q": RESTORE q
    CASE IS = "R": RESTORE r
    CASE IS = "S": RESTORE s
    CASE IS = "T": RESTORE t
    CASE IS = "U": RESTORE u
    CASE IS = "V": RESTORE v
    CASE IS = "W": RESTORE w
    CASE IS = "X": RESTORE x
    CASE IS = "Y": RESTORE y
    CASE IS = "Z": RESTORE z
    CASE IS = "1": RESTORE 1
    CASE IS = "2": RESTORE 2
    CASE IS = "3": RESTORE 3
    CASE IS = "4": RESTORE 4
    CASE IS = "5": RESTORE 5
```

```

CASE IS = "6": RESTORE 6
CASE IS = "7": RESTORE 7
CASE IS = "8": RESTORE 8
CASE IS = "9": RESTORE 9
CASE IS = "0": RESTORE 0
CASE IS = "-": RESTORE minus
CASE IS = "+": RESTORE plus
CASE IS = "=": RESTORE equal
CASE IS = "<": RESTORE less
CASE IS = ">": RESTORE great
CASE IS = "#": RESTORE num
CASE IS = "^": RESTORE up
CASE IS = "-": RESTORE down
CASE IS = "?": RESTORE ques
CASE IS = "*": RESTORE star
CASE IS = "!": RESTORE excl
CASE IS = " ": RESTORE blank
CASE IS = ":": RESTORE colon
CASE ELSE: RESTORE blank
END SELECT
  READ byte(0 + I)
  READ byte(10 + I)
  READ byte(20 + I)
  READ byte(30 + I)
  READ byte(40 + I)
NEXT I

start:
IF INKEY$ = "q" THEN END
OUT &H37A, 7: OUT &H37A, 3 'reset
FOR a = 1 TO 50
  OUT &H378, byte(a)
  IF a MOD 10 = 0 THEN
    OUT &H37A, 2: OUT &H37A, 0 'next column
    FOR t = 1 TO 50: NEXT t 'delay to display (50 = 486)
    OUT &H37A, 2: OUT &H37A, 0 'next column
  END IF
  OUT &H37A, 3: OUT &H37A, 2 'next byte
NEXT a
new = new + 1
IF new MOD 100 = 0 THEN GOTO getline
GOTO start
END

a: DATA 003,237,238,237,003
b: DATA 000,118,118,118,137
c: DATA 129,126,126,126,189
d: DATA 000,126,126,126,129
e: DATA 000,118,118,118,126
f: DATA 000,246,246,246,254
g: DATA 129,126,126,094,157
h: DATA 000,247,247,247,000
i: DATA 255,126,000,126,255
j: DATA 159,127,127,127,128
k: DATA 000,231,219,189,126
l: DATA 000,127,127,127,127
m: DATA 000,253,251,253,000
```

Continued

```

n: DATA 000,249,231,159,000
O: DATA 129,126,126,126,129
p: DATA 000,246,246,246,249
q: DATA 129,126,126,062,001
r: DATA 000,230,214,182,121
s: DATA 121,118,118,118,142
t: DATA 254,254,000,254,254
u: DATA 128,127,127,127,128
v: DATA 240,207,063,207,240
w: DATA 192,063,207,063,192
x: DATA 126,145,231,145,126
y: DATA 252,243,015,243,252
z: DATA 030,078,102,114,120
1 : DATA 255,125,000,127,255
2 : DATA 030,110,110,110,113
3 : DATA 118,118,118,118,137
4 : DATA 240,247,247,000,247
5 : DATA 112,118,118,118,142
6 : DATA 129,118,118,118,142
7 : DATA 254,254,254,014,240
8 : DATA 137,118,118,118,137
9 : DATA 249,246,246,054,193
star: DATA 213,227,128,227,213
excl: DATA 255,064,064,255,255
blank: DATA 255,255,255,255,255
great: DATA 255,190,221,235,247
less: DATA 255,247,235,221,190
equal: DATA 255,235,235,235,255
up: DATA 247,251,253,251,247
down: DATA 247,239,223,239,247
minus: DATA 247,247,247,247,247
plus: DATA 247,247,193,247,247
num: DATA 235,193,235,193,235
ques: DATA 254,078,238,238,241
colon: DATA 255,153,153,255,255

```

This is the software listing for a sign that displays a countdown to midnight. This version was written for and tested on New Years eve December 31 1999. Note that much of the code is the same as the previous version.

```

CLS : PRINT "Printer port sign, Copyright 1999 BY Robert J Davis"
PRINT "press q to quit"
'INPUT "Use Printer Port Number: ", lpt
'Hardcoded to lpt2 for higher speed - needed for a 486.
'Make these changes for using other printer ports;
IF lpt = 2 THEN dout = &H378: cout = &H37A: cin = &H379
IF lpt = 1 THEN dout = &H3BC: cout = &H3BE: cin = &H3BD
IF lpt = 3 THEN dout = &H278: cout = &H27A: cin = &H279
newtime:
new = 1
hours$ = STR$(23 - VAL(MID$(TIMES$, 1, 2)))
hours$ = MID$(hours$, 2, 2)
IF VAL(hours$) < 10 THEN hours$ = "0" + hours$
minutes$ = STR$(59 - VAL(MID$(TIMES$, 4, 2)))
minutes$ = MID$(minutes$, 2, 2)
IF VAL(minutes$) < 10 THEN minutes$ = "0" + minutes$
seconds$ = STR$(59 - VAL(MID$(TIMES$, 7, 2)))
seconds$ = MID$(seconds$, 2, 2)
IF VAL(seconds$) < 10 THEN seconds$ = "0" + seconds$
message$ = " " + hours$ + ":" + minutes$ + ":" + seconds$
IF hours$ = "00" AND minutes$ = "00" THEN
  IF seconds$ = "59" OR seconds$ = "58" THEN message$ = "*" HAPPY "*"
  IF seconds$ = "57" OR seconds$ = "56" THEN message$ = " NEW YEAR"
END IF
LOCATE 3, 2: PRINT "TIME: " + TIMES$
PRINT "COUNT:" + message$

```

```

seconds$ = MID$(TIMES$, 7, 2)
DIM array, byte(50)
FOR I = 1 TO 10
  letter$ = MID$(message$, I, 1)
  letter$ = UCASE$(letter$)
  SELECT CASE letter$
CASE IS = "A": RESTORE a
CASE IS = "B": RESTORE b
CASE IS = "C": RESTORE c
CASE IS = "D": RESTORE d
CASE IS = "E": RESTORE e
CASE IS = "F": RESTORE f
CASE IS = "G": RESTORE g
CASE IS = "H": RESTORE h
CASE IS = "I": RESTORE i
CASE IS = "J": RESTORE j
CASE IS = "K": RESTORE k
CASE IS = "L": RESTORE l
CASE IS = "M": RESTORE m
CASE IS = "N": RESTORE n
CASE IS = "O": RESTORE O
CASE IS = "P": RESTORE p
CASE IS = "Q": RESTORE q
CASE IS = "R": RESTORE r
CASE IS = "S": RESTORE s
CASE IS = "T": RESTORE t
CASE IS = "U": RESTORE u
CASE IS = "V": RESTORE v
CASE IS = "W": RESTORE w
CASE IS = "X": RESTORE x

```

```

CASE IS = "Y": RESTORE y
CASE IS = "Z": RESTORE z
CASE IS = "1": RESTORE 1
CASE IS = "2": RESTORE 2
CASE IS = "3": RESTORE 3
CASE IS = "4": RESTORE 4
CASE IS = "5": RESTORE 5
CASE IS = "6": RESTORE 6
CASE IS = "7": RESTORE 7
CASE IS = "8": RESTORE 8
CASE IS = "9": RESTORE 9
CASE IS = "0": RESTORE O
CASE IS = "-": RESTORE minus
CASE IS = "+": RESTORE plus
CASE IS = "=": RESTORE equal
CASE IS = "<": RESTORE less
CASE IS = ">": RESTORE great
CASE IS = "#": RESTORE num
CASE IS = "^": RESTORE up
CASE IS = "~": RESTORE down
CASE IS = "?": RESTORE ques
CASE IS = "*": RESTORE star
CASE IS = "!": RESTORE excl
CASE IS = " ": RESTORE blank
CASE IS = ":" RESTORE colon
CASE ELSE: RESTORE blank
END SELECT
READ byte(0 + I)
READ byte(10 + I)
READ byte(20 + I)
READ byte(30 + I)
READ byte(40 + I)
NEXT I
start:
IF INKEY$ = "q" THEN END
OUT &H37A, 7: OUT &H37A, 3 'reset
FOR a = 1 TO 50
  OUT &H378, byte(a)
  IF a MOD 10 = 0 THEN
    OUT &H37A, 2: OUT &H37A, 0 'next column
    FOR t = 1 TO 300: NEXT t 'delay to display (50 = 486)
    OUT &H37A, 2: OUT &H37A, 0 'next column
  END IF
  OUT &H37A, 3: OUT &H37A, 2 'next byte
NEXT a
IF seconds$ <> MID$(TIMES$, 7, 2) THEN GOTO newtime
GOTO start
END

```

```

a: DATA 003,237,238,237,003
b: DATA 000,118,118,118,137
c: DATA 129,126,126,126,189
d: DATA 000,126,126,126,129
e: DATA 000,118,118,118,126
f: DATA 000,246,246,246,254
g: DATA 129,126,126,094,157
h: DATA 000,247,247,247,000
i: DATA 255,126,000,126,255
j: DATA 159,127,127,127,128
k: DATA 000,231,219,189,126
l: DATA 000,127,127,127,127
m: DATA 000,253,251,253,000
n: DATA 000,249,231,159,000
O: DATA 129,126,126,126,129
p: DATA 000,246,246,246,249
q: DATA 129,126,126,062,001
r: DATA 000,230,214,182,121
s: DATA 121,118,118,118,142
t: DATA 254,254,000,254,254
u: DATA 128,127,127,127,128
v: DATA 240,207,063,207,240
w: DATA 192,063,207,063,192
x: DATA 126,145,231,145,126
y: DATA 252,243,015,243,252
z: DATA 030,078,102,114,120
1 : DATA 255,125,000,127,255
2 : DATA 030,110,110,110,113
3 : DATA 118,118,118,118,137
4 : DATA 240,247,247,000,247
5 : DATA 112,118,118,118,142
6 : DATA 129,118,118,118,142
7 : DATA 254,254,254,014,240
8 : DATA 137,118,118,118,137
9 : DATA 249,246,246,054,193
star: DATA 213,227,128,227,213
excl: DATA 255,064,064,255,255
blank: DATA 255,255,255,255,255
great: DATA 255,190,221,235,247
less: DATA 255,247,235,221,190
equal: DATA 255,235,235,235,255
up: DATA 247,251,253,251,247
down: DATA 247,239,223,239,247
minus: DATA 247,247,247,247,247
plus: DATA 247,247,193,247,247
num: DATA 235,193,235,193,235
ques: DATA 254,078,238,238,241
colon: DATA 255,153,153,255,255

```

but a smaller value might be used for more brightness. That is a limitation resulting from using the 74LS373; a higher voltage would damage the ICs. The 10-volt source does get connected to the TIP120s because the base of the transistors only rises to five volts and the transistor will not deliver any higher voltage in the emitter follower configuration. If the five-volt source is used instead, then the voltage regulator will get very hot due to the current that will be needed.

Assembly Hints

1. Mount electrolytic capacitors under the board laying flat against it.
2. Mount the TIP120's flat on the board and fasten them down with a copper wire or screws and nuts through their tabs.
3. The 47-ohm resistors fit best under the board soldered directly to LED arrays.
4. Pull-up resistors on the printer port control lines are not critical, they can be 1k to 10k.

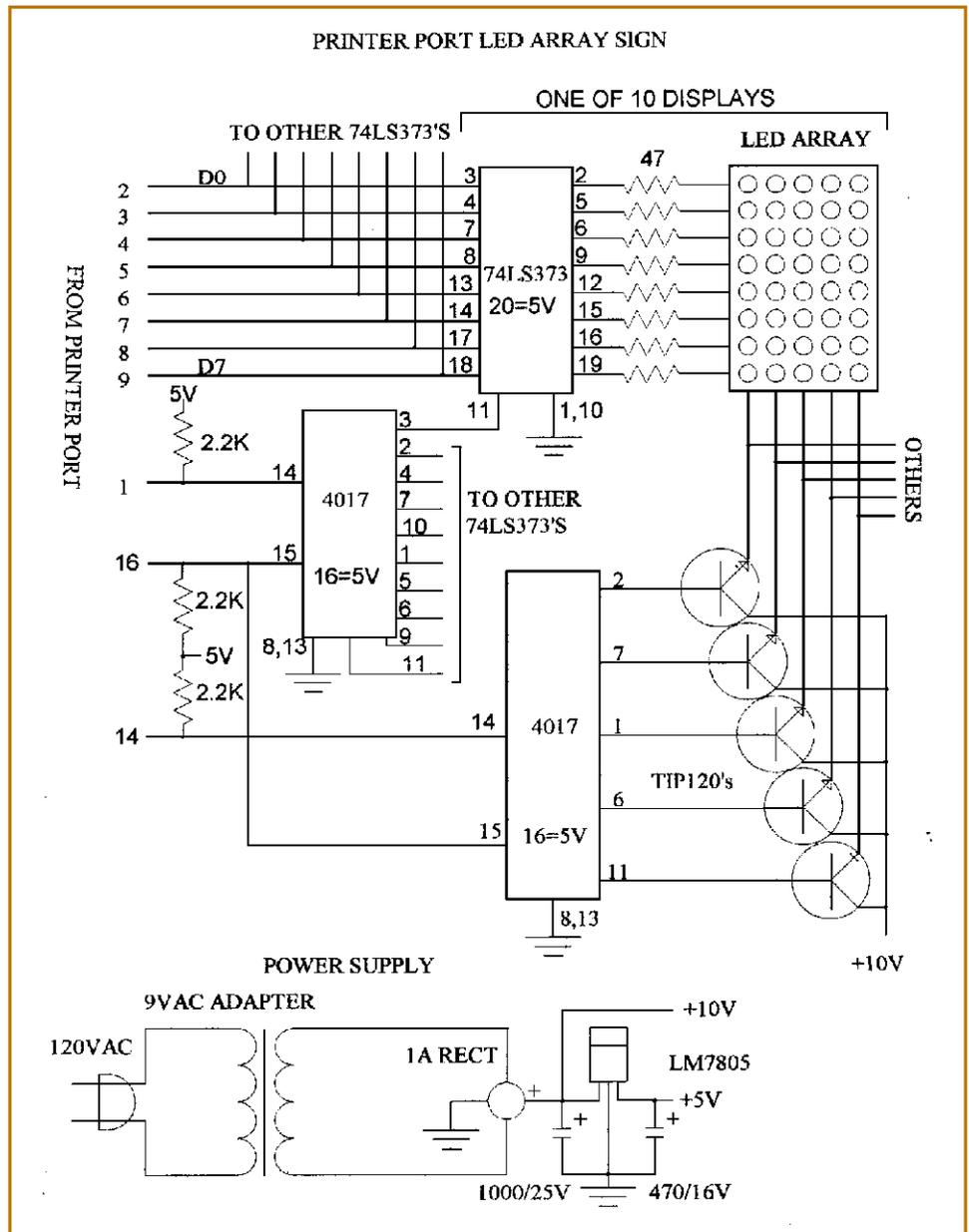
The frame for the sign is made out of a one by three inch board with a 3/8-inch groove cut into it. I did not miter the corners, but that might have helped the appearance. I had to chisel a notch into the back side of the frame for the header that is used to connect to a ribbon cable that has a 25-pin male connector on its other end. A 25-pin printer type extension cord then goes to the computer's printer port.

The software was used on an old 486 laptop at first. The problem is that the 486 takes too long to load the 10 latches and lacks display time afterwards. This results in a dimmer display or one that blinks in a well-lit room. The software works best on a 100 MHz or faster Pentium computer. On such a computer, the delay for display setting can be set to 500 or even 1,000 on a faster computer.

The program has many "case" statements that seem like they could be replaced with a loop. The problem is that line titles must be identified for the restore statements. It is all right to use letters of the alphabet for line titles, but you cannot use punctuation for line titles. Hence, the punctuation lines have to be spelled out to work properly.

Another long part of the code is the data statements. They define what LEDs to light in order to produce the desired character. The confusing thing about the bytes used is that they are actually inverted. To light an LED, you need a zero or a low on the row drivers. Hence, a zero is a lit LED and a one does not light the LED.

To create the data statements, I drew out the pattern on graph paper then converted the lit LEDs to binary, then octal, and subtracted the



value from 255.

After a while, you might be able to do it in your head, or remember that the same bit pattern is used elsewhere and just copy it. **NV**

If you would like to download the software listing, go to www.nutsvolts.com.