QB MONITOR

101

448C9

QB-99'ERS U.G. NEWSLETTER

SEPTEMBER 1988

The QB MONITOR is the Newsletter of the QB-99'ers User Group, is printed Sept. thru June and sent in exchange for other User Group Newsletters. Send Exchange Newsletter to Frank Cotty, Queensborough Community College, Bayside, NY 11364. Credit original sources.

The QB 99'ers meets the second Saturday of each month September through May, at Queensborough Community College, Bayside New York, room S225, at 2 P.M. Calendar at right shows dates

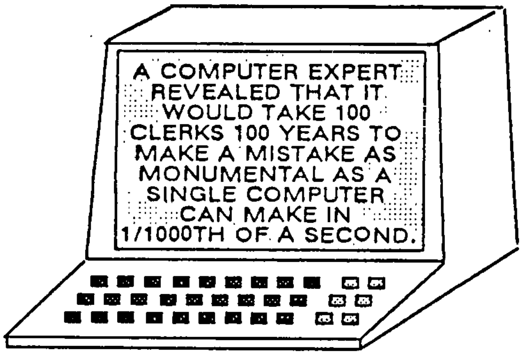
September						1988
S	M	T	W	T	F	S
				1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	

October						1988
S	M	T	W	T	F	S
						1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29
30	31					

November						1988
S	M	T	W	T	F	S
		1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30			

The QB 99'ers Users Group is a not for profit group of individuals interested in furthering their knowledge of the Orphaned TI-99/4A Home Computer. There is no connection between the QB 99'ers and Texas Instruments. Meeting location, notices and newsletter printing and mailings are graciously supplied as a public service by Queensborough Community College, Bayside, New York 11545. Those who choose to be members of the QB 99'ers pay a modest fee used for obtaining new equipment, for software and for refreshments available at any meeting.

To join the QB 99'ers Users Group simply come to meetings, or send for more information: Frank Cotty, Biology Department



A COMPUTER EXPERT REVEALED THAT IT WOULD TAKE 100 CLERKS 100 YEARS TO MAKE A MISTAKE AS MONUMENTAL AS A SINGLE COMPUTER CAN MAKE IN 1/1000TH OF A SECOND.

Contents **Page**

QB MONITOR ~ QB-99'er NEWSLETTER .

Welcome back! I hope each of you had a teriffic summer. It's been a long time since the last QB Monitor was published. I hope to be able to return to the regular schedule we had established for ourselves the past year. Much of the credit for the regularity of this publication must go to Ed Machonis. Ed never *lets* a month go by without contributing some of the best writing seen in this publication.

You are, of course, waiting to see what wonders Ed Machonis has performed this month. I have this secret desire to be a millionaire. At least that's all it has amounted to. If you are like me you have played the local state lottery at least once. Well Ed has made it easier to gamble our money away. I always wait till the last minute before buying the tickets and usually don't get to select my numbers before I enter the store. sometimes the store closes before I can select the six numbers in two games. With Ed's TIny LOTTO program I can select my winners months in advance.

It is my pleasure to add to this month's contents another fine example of writing by our members. Jack Youngs has provided us with a modification of the great LA Styler program written by Tom Freeman. This modification permits those with color printing capability to select color as well as type style.

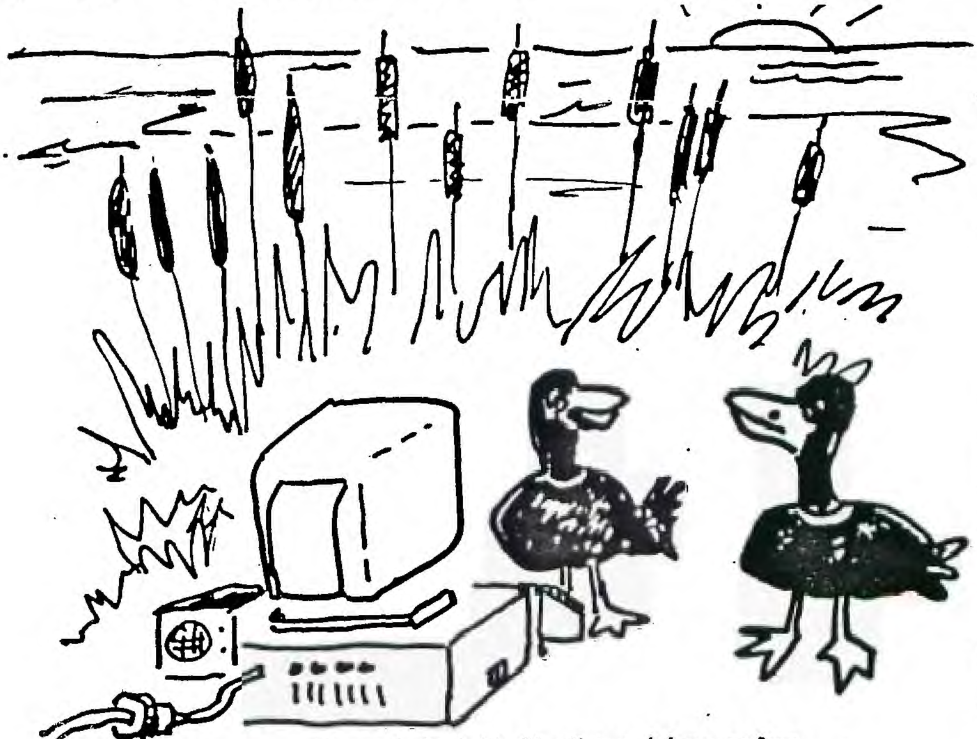
Our children may come second to our computers, but we still think of them. To show we are all heart I include Kidkeys (no not KIDNEYS) from BYTEMONGER newsletter of the Lexington, Kentucky BLUEGRASS 99'er Computer Society. This is a neat little program to turn your TI into a piano.

Is LOGO your cup of tea? You want to try it out? Well this article by Rick Felzien on recursion (?) in LOGO will help.

John Willforth continued his Let's Talk RAM disks series in the West Penn 99'ers Club so here is Part V (Part VI next month)

As much as we like or CATLIB program, we still need hard copy records of our disks' contents. The 3X5 card catalog utility prints your disks' catalogs to an index card. It's necessary! An updated version by Ed York.

And finally now you can have a 32K memory expansion without the expansion box. And you only pay $25! By Joe Spiegel of the Airport Area Computer Club.



'Say each shot holds 48 pellets and there are four guys with four guns ... what pattern should we fly?'

LA STYLER was writtn by Tom Freeman of LA 99 er's. I don't believe I saw the original but I did see and use the revision by Ed Machonis of QB 99er's.

I have a JX-80 color printer, and I thought this would be a very good program for me if I could add the color codes to the program. I did it before for Ed's label and tag programs and others. I also have a merge program to use when needed. Anyway the program is as follows for anyone out there that has a JX-80 color printer.

This is the screen you see when the program loads

```
MODE            1=ON,0=OFF
1  ELITE            0
2  COMPRESSED       0
3  EMPHASIZED       0
4  DOUBLE STRIKE    0
5  EXPANDED         0
6  ITALICS          0
7  UNDERLINE        0
8  SUPERSCRIPT      0
9  SUBSCRIPT        0
10 X/72 IN. LF    X=12
11 L MARGIN       X=0
12 R MARGIN       X=80
13 SKIP X LINES   X=0
14 COLOR FOR EPSON JX80
15 TEST
16 RESET/PICA
17 INPUT TEXT
18 EXIT
```

Some of the control codes of the JX80 are different so they were changed also. for instance, there is no Elite with Emphasized and no Compressed with Emphasized and a few others.

Most important in typing in this program, the 3rd and 4th DATA items in line 360 are not blank spaces but CHR$(18) (Type CONTROL plus R) and CHR$(15) (CONTROL O),respectively. Similarly ,the apparent two blank spaces at the end of EACH quoted string in line 370 are actually CHR$(10) (CONTROL J) and CHR$(13) (CONTROL M). The blank space at the beginning of EACH quoted string is a true blank space and required in this program (as it saves sending additional Escape code to the printer).

```
100 ! *** JX80 STYLER ***
by Jack Youngs QB 99ers-a
revision of program by Ed
Machonis QB 99ers-based on a
 program by Tom Freeman

110 DIM P$(16,2),N(18):: M(1
)=12 :: M(2),M(4)=0 :: M(3)=
80

120 FOR X=1 TO 16 :: FOR J=0
 TO 1 :: READ P$(X,J):: NEXT
 J :: NEXT X :: FOR X=1 TO 4
 :: READ T$(X):: NEXT X :: O
PEN #1:"PIO.CR" :: PRINT #1:
CHR$(27)&"@"

130 DISPLAY AT(3,1)ERASE ALL
:"MODE","1=ON,0=OFF","1  ELI
TE":"2  COMPRESSED":"3  EMPH
ASIZED":"4  DOUBLE STRIKE":"
5  EXPANDED":"6  ITALICS":"7
  UNDERLINE"

140 DISPLAY AT(11,1):"8  SUP
ERSCRIPT":"9  SUBSCRIPT"

150 IF I=8 THEN N(9)=0 ELSE
IF I=9 THEN N(8)=0

160 IF N(1)=1 THEN N(3)=0 EL
SE IF N(3)=1 THEN N(1)=0

170 IF N(2)=1 THEN N(3)=0 EL
SE IF N(3)=1 THEN N(2)=0

180 FOR X=1 TO 9 :: DISPLAY
AT(X+3,18):N(X):: NEXT X

190 FOR X=1 TO 4 :: DISPLAY
AT(X+12,1):T$(X);STR$(M(X)):
: NEXT X

200 DISPLAY AT(17,1)SIZE(23)
:"14 COLOR FOR EPSON JX80"

210 DISPLAY AT(18,1):"15 TES
T":"16 RESET/PICA":"17 INPUT
 TEXT":"18 EXIT"

220 ACCEPT AT(22,1)VALIDATE(
DIGIT," ")SIZE(-2)BEEP:I

230 IF I>18 THEN 220 ELSE N(
I)=N(I)XOR 1 :: ON I GOTO 33
0,330,330,330,330,330,330,33
0,330,310,310,310,310,240,33
0,300,340,350

240 CALL CLEAR :: DISPLAY AT
(3,2):"0 BLACK  4 YELLOW
       1 RED    5 ORANGE
       2 BLUE   6 GREEN
       3 VIOLET 7 STYLER"

250 INPUT "COLOR OR STYLER?"
:I

260 IF (I<0)+(I>7)THEN 240

270 PRINT #1:CHR$(27);"r";CH
R$(I);

280 IF (I<>7)THEN 240

290 IF I=7 THEN 130

300 FOR X=1 TO 14 :: N(X)=0
:: NEXT X :: GOTO 320

310 ACCEPT AT(I+3,19)VALIDAT
E(DIGIT," ")SIZE(-2)BEEP:M(I
-9)

320 PRINT #1:CHR$(27)&P$(I,N
(I))&CHR$(M(I-9)):: GOTO 150

330 PRINT #1:CHR$(27)&P$(I,N
(I)):: IF I=16 THEN M(1)=12
:: M(2),M(4)=0 :: M(3)=8 0 :
: GOTO 150 :: ELSE 150

340 PRINT "INPUT A LINE OF T
EXT":"(ZZZ RETURNS TO MENU)"
 :: LINPUT A$ :: IF A$="ZZZ"
 OR A$="zzz" THEN 130 ELSE P
RINT #1:A$&CHR$(10)&CHR$(13)
:: GOTO 340

350 CLOSE #1

360 DATA P,M, , ,F,E,H,8,W0,
W1,5,4,-0,-1,T,S0,T,S1,A,A,1
,1,Q,Q,N,N,x0,x1

370 DATA " QUICK BROWN FOX J
UMPS OVER THE LAZY RED DOG 1
234567890  "," QUICK BROWN F
OX JUMPS OVER THE LAZY RED D
OG 1234567890  ",@,@

380 DATA 10 X/72 IN. LF  X=,
11 L MARGIN     X=,12 R MARG
IN     X=,13 SKIP X LINES X=

390 DATA BLACK,RED BLUE,VIOL
ET,YELLOW,ORANGE,GREEN
```

TIny LOTTO

A TIny GRAM

by Ed Machonis

The first program I ever wrote on my newly purchased TI-99/4A was a random number generator for NY State Lotto games. When I finally got the program to generate 6 numbers between 1 and 40, I was elated. Not *having* any way to save the program, I copied the code onto paper with a pencil.

Knowing I was on my way to my first million, I decided to splurge on a cassette recorder. Leaving the computer turned on, I embarked on a two hour trip into the city to purchase a recorder. Upon my return, about five hours later, I saved the program. That first number generator eventually grew to 46 sectors, with nearly every conceivable bell and whistle, and took over 3 minutes to load from cassette.

Here I am, 5 years later, still writing Lotto programs and still chasing that first million. Hope also springs eternal in the compulsive programmer's [Gambler's?] breast! If nothing else, I did learn to write them smaller.

But don't let the small size fool you. This screenfull of code does a lot of work and once again proves the power of the TI-99/4A. It will generate random numbers for any of the popular lottery games, WIN 3, WIN 4, Pick 6 LOTTO, and WIN 10 or Keno. The low number can be a zero or a one. The high number can be whatever is being used, 40, 48, 54, 80, 999 or 9999. It should work in any state.

The same RND statement, in Line 4, borrowed from son Michael's Basic 10 liner LUCKY LOTTO, is used to generate the random numbers for all games. A clever piece of code well worth your study.

Where multiple numbers are generated for a game, as in Lotto or Keno, duplicate numbers are discarded and the numbers are sorted in ascending order to make it easier to fill out your bet *slip*. Output can be directed to screen or printer. When several games are played, the hard copy is easier to check for winners than the individual tickets.

Leading zeroes are inserted where required to keep the columns neatly aligned and to reduce the possibility of transcription errors. A total of 10 Lotto games (the bet slip capacity) can be displayed on the screen without any scrolling off.

The N.Y. State Lottery states "...if you are playing Lotto for the big prize, pick your numbers randomly." Early on, I distrusted the randomness of TI's RND function, and in my niavety visualized having to split that first million with half the TI owners in NY State. Despite a RANDOMIZE statement, the computer often generated identical series of numbers.

TI's User's Reference Guide states on page II-96, "The random number function gives you the NEXT PSEUDO-RANDOM number in the current SEQUENCE of pseudo-random numbers." Page II-95 states: "When the RANDOMIZE statement is used ..... a different and unpredictable SEQUENCE of random numbers is generated .... each time the program is run." RND generates numbers in accordance with a built-in sequence. The RANDOMIZE statement merely insures that a program does not ALWAYS start with the same sequence. But it can, HAS and will.

The RANDOMIZE statement in Line 3 can be placed in three different positions. Placing it before the start of the G loop will cause an unpredictable sequence to be selected each time the program is RUN. Placed before the start of the K loop, a new sequence is used for each game. Placing it after the start of the K loop, as it is, causes an unpredictable sequence to be selected for each number that is generated. As only one number is used from each sequence, we are no longer governed by the built-in sequence and the program generates truly random numbers.

WIN 3 numbers can be selected with TIny LOTTO in one of two ways. We can use a Low Number of 0, a High Number of 999, and 1 number per game. Or one can use a Low Number of 0, a High Number of 9, and three numbers per game. The same two methods are available for four digit numbers, using 9999 and 1, or 9 and 4, as required. In the first case, a three digit number is selected, in the second case each digit of the three digit number is separately selected. Just a little user friendliness to conform to the way the user thinks of the numbers.

If you find you only play one type of game, and are always entering the same information in response to the prompts, Line 2 can be changed to permanently assign values to the variables. Suppose your regular selection is for 10 games of Pick 6 Lotto, with a low number of 1 and a high number of 54, with output to a printer. Line 2 would read:

```
2 L=1 :: H=54 :: T=6 :: Q=10
 :: P=1 :: OPEN #P:"PIO"
```

Due to sale of public domain programs by some software distributors, a copyright notice has been placed on this program. It may be freely distributed provided no fee of any kind is charged. This article and/or the program listing may be published in Newsletters of non profit user groups.

```
1 !$$$$    TIny LOTTO   $$$$
   $     Copyright 1988    $
   $     by Ed Machonis    $
   $ QB-99'ers, Bayside NY $
   $$$$$$$$$$$$$$$$$$$$$$$$$

2 CALL CLEAR :: INPUT "LOW N
UMBER? ":L :: INPUT "HIGH NU
MBER? ":H :: INPUT "NUMBERS
PER GAME? ":T :: INPUT "HOW
MANY GAMES? ":Q :: INPUT "SC
REEN=0 - PRINTER=1 (0/1)?":P
 :: IF P THEN OPEN #P:"PIO"

3 H$=STR$(H):: FOR G=1 TO Q
:: FOR K=1 TO T :: RANDOMIZE

4 N(K)=INT(RND*(H+ABS(L=0)))
+L :: FOR D=1 TO K-1 :: IF N
(K)=N(D)AND H>9 THEN 4

5 NEXT D :: NEXT K :: U=T-1
:: IF H<10 THEN 9

6 F=0 :: FOR K=1 TO U :: IF
N(K)<N(K+1)THEN 8

7 M=N(K):: N(K)=N(K+1):: N(K
+1)=M :: F=1 :: U=K

8 NEXT K :: IF F=1 THEN 6

9 FOR K=1 TO T :: N$(K)=STR$
(N(K)):: PRINT #P:RPT$("0",L
EN(H$)-LEN(N$(K)))&N$(K)&" "
;:: NEXT K :: PRINT #P:;:;:;
:: NEXT G ! GOOD LUCK!
```

KIDKEYS

Do you want your children to learn to play the piano, but can't afford one right now? Here is an inexpensive substitute: the TI-99/4A PIANO! Only the bottom three rows constitute the (PIANO), essentially all of the letter keys. Bear in mind that the keys react differently when the shift key is up or down. In one case whole notes are played, in the other half notes are played; notes repeat when the key is held down. The very bottom row plays noise tones when the shift key is locked: very amusing to little children. To keep the program simple, the screen shows only the upper and lower case options, but that should not impede all you budding programmers; create some nice graphics to enliven this music program and make it even more attractive to your children. This program was originally *published in* "Nittinian", the Swedish newsletter for 99/ers, by an unknown author. The transliteration was added by Maurice E. T. Swinnen of the Washington DC Area 99/er Computer Club.

```
 6Ø REM PIANO, NITTINIAN 84-2
 7Ø REM TI-99/4A EXTENDED BASIC
 8Ø CALL CLEAR :: DISPLAY AT(6,13):"PIA
    NO" :: DISPLAY AT(1Ø,2):"UPPER CASE
    =HALF TONES+NOISE"
 9Ø DISPLAY AT(12,2):"LOWER CASE=WHOLE
    TONES" :: DISPLAY AT(16,8):"PRESS A
    NY KEYS"
1ØØ CALL KEY(Ø,K,S):: IF S=Ø THEN 1ØØ
11Ø IF K=45 THEN 1ØØ
12Ø IF K<44 THEN 1ØØ ELSE IF K>46 AND K
    <58 THEN 1ØØ ELSE IF K>6Ø AND K<65
    THEN 1ØØ
13Ø IF K>9Ø ANO K<96 THEN 1ØØ ELSE IF K
    >96 THEN 21Ø !CHECK IF LOWER OR UPP
    ER CASE LETTER HAS BEEN PRESSEO
14Ø IF K=44 THEN CALL SOUND(-1ØØ,1568,Ø
    ):: GOTO 1ØØ
15Ø IF K=46 THEN CALL SOUND(-1ØØ,176Ø,Ø
    ):: GOTO 1ØØ
16Ø IF K=59 THEN CALL SOUNO(-1ØØ,698,Ø)
    :: GDTO 1ØØ
17Ø IF K=58 THEN CALL SOUND(-1ØØ,1661,Ø
    ):: GOTO 1ØØ
18Ø IF K=6Ø THEN CALL SOUNO(-12Ø,-8,Ø):
    : GOTO 1ØØ
19Ø REM UPPER CASE LETTERS ASCII=65->9Ø
2ØØ ON K-64 GOTO 23Ø,24Ø,25Ø,26Ø,27Ø,28
    Ø,29Ø,3ØØ,31Ø,32Ø,33Ø,34Ø,35Ø,36Ø,3
    7Ø,38Ø,39Ø,4ØØ,41Ø,42Ø,43Ø,44Ø,45Ø,
    46Ø,47Ø,48Ø
21Ø ON K-96 GOTO 5ØØ,51Ø,52Ø,53Ø,54Ø,55
    Ø,56Ø,57Ø,58Ø,59Ø,6ØØ,61Ø,62Ø,63Ø,6
    4Ø,65Ø,66Ø,67Ø,68Ø,69Ø,7ØØ,71Ø,72Ø,
    73Ø,74Ø,75Ø
22Ø REM UPPER CASE LETTERS=HALF TONES+N
    OISE
23Ø CALL SOUNO(-12Ø,466,Ø):: GOTO 1ØØ
24Ø CALL SOUNO(-12Ø,-6,Ø):: GOTO 1ØØ
25Ø CALL SOUND(-12Ø,-1,Ø):: GOTO 1ØØ
26Ø CALL SOUNO(-12Ø,622,Ø):: GOTO 1ØØ
27Ø CALL SOUND(-12Ø,177,Ø):: GDTO 1ØØ
28Ø CALL SOUND(-12Ø,74Ø,Ø):: GDTO 1ØØ
29Ø CALL SOUNO(-12Ø,831,Ø):: GOTO 1ØØ
3ØØ CALL SOUNO(-12Ø,932,Ø):: GOTO 1ØØ
31Ø CALL SOUND(-12Ø,311,Ø):: GOTO 1ØØ
32Ø CALL SOUND(-12Ø,11Ø9,Ø):: GOTO 1ØØ
33Ø CALL SOUNO(-12Ø,1245,Ø):: GOTO 1ØØ
34Ø CALL SOUNO(-12Ø,148Ø,Ø):: GOTO 1ØØ
35Ø CALL SOUND(-12Ø,-4,Ø):: GOTO 1ØØ
36Ø CALL SOUNO(-12Ø,-5,Ø):: GOTO 1ØØ
37Ø CALL SOUNO(-12Ø,37Ø,Ø):: GOTO 1ØØ
38Ø CALL SOUNO(-12Ø,415,Ø):: GOTO 1ØØ
39Ø CALL SOUND(-12Ø,139,Ø):: GOTO 1ØØ
4ØØ CALL SOUND(-12Ø,185,Ø):: GOTO 1ØØ
41Ø CALL SOUND(-12Ø,554,Ø):: GOTO 1ØØ
42Ø CALL SOUND(-12Ø,2Ø8,Ø):: GOTO 1ØØ
43Ø CALL SOUND(-12Ø,277,Ø):: GOTO 1ØØ
44Ø CALL SOUND(-12Ø,-7,Ø):: GOTO 1ØØ
45Ø CALL SOUND(-12Ø,156,Ø):: GOTO 1ØØ
46Ø CALL SOUND(-12Ø,-2,Ø):: GOTO 1ØØ
47Ø CALL SOUNO(-12Ø,233,Ø):: GOTO 1ØØ
48Ø CALL SOUND(-12Ø,-3,Ø):: GOTO 1ØØ
49Ø REM LOWER CASE LETTERS=WHOLE TONES
5ØØ CALL SOUNO(-1ØØ,294,Ø):: GOTO 1ØØ
51Ø CALL SOUNO(-1ØØ,1175,Ø):: GOTO 1ØØ
52Ø CALL SOUND(-1ØØ,988,Ø):: GOTO 1ØØ
53Ø CALL SOUND(-1ØØ,349,Ø):: GOTO 1ØØ
54Ø CALL SOUNO(-1ØØ,131,Ø):: GOTO 1ØØ
55Ø CALL SOUNO(-1ØØ,392,Ø):: GOTO 1ØØ
56Ø CALL SOUNO(-1ØØ,44Ø,Ø):: GOTO 1ØØ
57Ø CALL SOUNO(-1ØØ,494,Ø):: GOTO 1ØØ
58Ø CALL SOUND(-1ØØ,22Ø,Ø):: GOTO 1ØØ
59Ø CALL SOUND(-1ØØ,523,Ø):: GOTO 1ØØ
6ØØ CALL SOUND(-1ØØ,587,Ø):: GOTO 1ØØ
61Ø CALL SOUND(-1ØØ,659,Ø):: GOTO 1ØØ
62Ø CALL SOUNO(-1ØØ,1397,Ø):: GOTO 1ØØ
63Ø CALL SOUNO(-1ØØ,1319,Ø):: GOTO 1ØØ
64Ø CALL SOUNO(-1ØØ,247,Ø):: GOTO 1ØØ
65Ø CALL SOUNO(-1ØØ,262,Ø):: GOTO 1ØØ
66Ø CALL SOUND(-1ØØ,11Ø,Ø):: GOTO 1ØØ
67Ø CALL SOUNO(-1ØØ,147,Ø):: GOTO 1ØØ
68Ø CALL SOUNO(-1ØØ,33Ø,Ø):: GOTO 1ØØ
69Ø CALL SOUNO(-1ØØ,165,Ø):: GOTO 1ØØ
7ØØ CALL SOUND(-1ØØ,196,Ø):: GOTO 1ØØ
71Ø CALL SOUNO(-1ØØ,1Ø47,Ø):: GOTO 1ØØ
72Ø CALL SOUND(-1ØØ,123,Ø):: GOTO 1ØØ
73Ø CALL SOUND(-1ØØ,88Ø,Ø):: GOTO 1ØØ
74Ø CALL SOUNO(-1ØØ,175,Ø):: GOTO 1ØØ
75Ø CALL SOUND(-1ØØ,784,Ø):: GOTO 1ØØ
```

The Logo Program

by
Rick Felzien

I have been asked on several occasions to give an explanation of what is meant by recursion. This term is used extensively in literature pertaining to Logo. .

As we have seen before, the REPEAT command in Logo is the equivelant to a FOR-NEXT loop. In Basic this is a controlled loop because the number 5 controls the number of repetitions to be performed, as would the Repeat command.

```
MAKE "B :5                                  B=5
REPEAT 5 [MAKE "C :A+B-1 ]                  FOR I=1 TO 5
                                            C=A+B-1
                                            NEXT I
```

An essentially endless loop in Basic is the same as recursion in Logo. In other words the control is not necessarily a numeric constant. Let's look at a short proceedure using recursion and the equivelent Basic loop.

```
TO COUNTDOWN :NUMBER           INPUT NUMBER
PRINT :NUMBER                  PRINT NUMBER
COUNTDOWN NUMBER - 1           NUMBER=NUMBER-1
IF :NUMBER = 0 STOP            IF NUMBER=0 THEN 150
END                            GOTO 100
                               END
```

In Basic you send control to the first line of the sequence, whereas in Logo the procedure in effect calls itself over and over. The following is an example of recursion.

```
TO COUNTDOWN :NUMBER
IF :NUMBER = 0 STOP
PRINT NUMBER
COUNTDOWN :NUMBER - 1
END
```

The if statement is used in Logo to perform tests, in this case to test whether the value of NUMBER is zero. If so, the COUNTDOWN procedure STOP's. That is rather than continuing with the next line in the procedure, it returns control to wherever the procedure was called from. So in response to the command COUNTDOWN 5, the computer prints 5, 4, 3, 2, 1 and prompts for a new command. Keep in mind that the idea of STOP is that when a procedure stops, the next command that gets executed is the one after the one that called the procedure. for example:

```
TO BLASTOFF
COUNTDOWN 10
FORWARD 100
END
```

counts down from 10 to 1 and then moves the turtle.
The IF statement is called a conditional expression.

IF {come condition is true} {do some action} THEN is understood.

following is a simple method of drawing a binary tree. on the and on the right a more complex, yet more simply written version.

```
TO VEE
LEFT 45
FORWARD 10
BACK 10
RIGHT 90
FORWARD 10
BACK 10
LEFT 45
END

TO BRANCH
FORWWARD 15
VEE
FORWARD 15
VEE
FORWARD 10
BACK 40
END

TO BUSH
LEFT 60
REPEAT 6 [BRANCH RIGHT 20 ]
BRANCH
LEFT 60
END

TO GREENTREE
FORWARD 50
BUSH
BACK 50
END

TO MOVE
PENUP
RIGHT 90
FORWARD 80
LEFT 90
PENDOWN
END

TO TREES
REPEAT 3 [GREENTREE MOVE ]
```

```
TO NEW.TREE :LENGTH :ANGLE :DEPTH
IF :DEPTH = 0 THEN STOP
LEFT ANGLE
FORWARD 2 * :LENGTH
NEW.TREE :LENGTH :ANGLE :DEPTH - 1
BACK 2 * :LENGTH
RIGHT 2 * :ANGLE
FORWARD :LENGTH
NEW.TREE :LENGTH :ANGLE :DEPTH - 1
BACK :LENGTH
LEFT :ANGLE
END
```

As you can see, Logo's procedures build on one another to become bigger and more powerful procedures.

LET'S TALK RAM DISKS PART V

By John F. Willforth (April 1988)

The MEMORY PLUS card from CORCOMP appeared on the market almost two years ago, and to date I have not seen one. I therefore was hesitant to write this article. I had to depend on an article by Scott Darling as well as information provided by Willis Richardson and the technical support at CORCOMP. I hope that it will be complete and accurate enough to merrit your consideration. I have tried to be as objective as possible on all the Ram Disks reviewed.

The MEMORY PLUS comes in both a PEB unit and a stand alone unit. The stand alone unit is more flexable in that it can be used in conjunction with your 32K expansion memory, while the PEB version cannot. They both come in 256K as well as 512K sizes, again the stand alone can be configured with an added unit (two 512K units for example) and the PEB version cannot. Both units are supported by a 9V. power supply to the ram disk card to support memory when a system is powered down normally. If a total failure of the AC occurs, you will lose all files on the MEMORY PLUS. This is a common failure of any Dynamic Ram based RAM DISK.

The MEMORY PLUS comes with the Disk Manager resident on the card, this is good for two reasons, one is that you don't have to load it from a diskette, and two, it is the only one that gives you full use of the disk. The manager is called with "CALL RAMGR" for units with the newest PROM installed V. 3.1 or "CALL RMGR" with lower versions, a good way to tell what PROM you have in your MEMORY PLUS. The disk manager can initialize the disk, handle all disk and file functions as well as test the entire ram disk memory. The manager is very similar to the disk manager that comes with the Corcomp disk controller. It has some nice features, among them pressing a "T" when selecting to copy a file that is protected, will Temporarily unprotect that file until the file has been copied. The resident disk manager will also work with other disk units in the system. A total of 2048 sectors is the default for a 512K and 1920 will be the limit if you wish for the 32K expansion memory to reside in this unit (required on a PEB only set-up).

A major draw back with the ram disk is in the fact that the entire disk is called as one volume. In other words if you intend to use "TIMP" for MultiPlan, that is the only name that can be used for that entire unit. You will have to take this into consideration if you are a user of software that is dependent on specific volume (disk) names. Many of the other ram disks do allow for multiple volume names within a single ram disk unit.

The MEMORY PLUS, according to Scott, is able to work in the system with a different ram disk present. This could be a saving grace to compensate for it not accepting more volume names. You will have to set up CRU addresses for your card, which by the way are >1000 and >1400 for the MEMORY PLUS.

The drive number can be set with the disk manager or under basic using a DELETE "SDx", where x is the drive # selected.

A lowercase with desenders is available for use by basic/xbasic simply by using a DELETE "LOWER".

There is a switch on the MEMORY PLUS which is of course located at the rear of the card (but has pins available for a remote connection) whose purpose is to assure an orderly power down of the PEB without glitching the ram disk and this switch should be used each time the PEB is powered down. A switch over of clock and flag settings will be done if this switch is pressed . I think this is a bothersome drawback.

I spoke of the Prom V. 3.1 which is available. Corcomp has corrected some problems such as a density identification problem in sector 0, and added the ability to catalog the disks to a serial, parallel port or to a disk.

The stand alone units are built by Corcomp as ordered, and any pricing should be checked with your CORCOMP dealer. CORCOMP has a good attitude of support for their products. Call them at (714) 630-2903 or write: CorComp Inc., 2211-G East Winston Road, Anaheim, CA 92807

By the way if you are still under a warranty CORCOMP will send you a new Prom and if your warranty has expired $15 will update your MEMORY PLUS. I have used all the space available this month, so check back next month for a review of the GRAND RAM. (Hopefully).

QB MONITOR ~ QB-99'er NEWSLETTER

Reprinted from CIN-DAY USERS GROUP NEWSLETTER - April, 1987

```
100 ! ########################
110 ! # 3 x 5 CARD CATALOG #
120 ! ########################
130 ! Original: Ed York
      Revamped: Rick Kellogg
      Revision: Ed York
140 !
150 DIM A$(12):: FOR A=1 TO
12 :: READ A$(A):: NEXT A ::
 CALL CLEAR
160 DATA JAN,FEB,MAR,APR,MAY
,JUN,JUL,AUG,SEP,OCT,NOV,DEC
170 DISPLAY AT(9,9):"DISK LA
BELER": :"     ENTER TODAYS
DATE:": :TAB(11);"MM/DD/YY":
 : :TAB(11);"__/__/8_"
180 ACCEPT AT(16,11)BEEP SIZ
E(-2)VALIDATE(DIGIT):B$ :: I
F B$="_" THEN 180 ELSE IF (
VAL(B$)>12)+(VAL(B$)<1)THEN
170
190 ACCEPT AT(16,14)SIZE(-2)
VALIDATE(DIGIT)BEEP:C$ :: IF
 C$="_" THEN 190 ELSE IF (V
AL(C$)>31)+(VAL(C$)<1)THEN 1
90
200 ACCEPT AT(16,19)SIZE(-2)
VALIDATE(DIGIT)BEEP:D$ :: D$
="8"&D$ :: GOSUB 610
210 E$=A$(VAL(B$))&" "&C$&",
 19"&D$ :: F$(1)="D/F" :: F$
(2)="D/V" :: F$(3)="I/F" ::
F$(4)="I/V" :: F$(5)="PROGRM
"
220 IMAGE "DISKNAME: #######
### FREE:####  USED:####"
230 IMAGE "###### SIDED/####
## DENSITY   ############"
240 IMAGE "########## #### #
##### #       ########## ###
# ###### #       ##########
#### ###### #"
250 IMAGE "########## #### #
##### #       ########## ###
# ###### #"
260 IMAGE "########## #### #
##### #"
270 OPEN #1:"PIO",VARIABLE 1
00 :: PRINT #1:CHR$(15);CHR$
(27);"S";CHR$(0);CHR$(27);"A
";CHR$(5)
280 OPEN #2:"DSK"&STR$(B)&".
",INPUT ,RELATIVE,INTERNAL :
: INPUT #2:G$,C,C,D
290 PRINT #1:RPT$("*",86);CH
R$(27);"A";CHR$(3):: E=0
300 PRINT #1:CHR$(27);"W";CH
R$(1):: PRINT #1,USING 220:G
$,D,C-D
310 IF C>720 AND C<1441 THEN
 H$="DOUBLE" :: I$="DOUBLE"
320 IF C>360 AND C<721 THEN
H$="SINGLE" :: I$="DOUBLE"
330 IF C<361 THEN H$="SINGLE
" :: I$="SINGLE"
340 PRINT #1:CHR$(14):: PRIN
T #1,USING 230:H$,I$,E$ :: P
RINT #1:CHR$(27);"W";CHR$(0)
350 PRINT #1:RPT$("=",86);CH
R$(27);"A";CHR$(5)
360 PRINT #1,USING 240:" Fil
ename ","Size"," Type ","P",
" Filename ","Size"," Type "
,"P"," Filename ","Size"," T
ype ","P"
370 PRINT #1,USING 240:"----
------","----","------","-",
"----------","----","------"
,"-","----------","----","--
----","-"
380 IF E=126 THEN 620
390 FOR F=1 TO 3 :: INPUT #2
:J$(F),G(F),H(F),I(F):: NEXT
 F :: IF LEN(J$(1))=0 THEN 4
40 ELSE IF LEN(J$(2))=0 THEN
 GOSUB 490 ELSE 410
400 PRINT #1,USING 260:J$(1)
,H(1),K$(1),L$(1):: E=E+1 ::
 GOTO 380
410 IF LEN(J$(3))=0 THEN GOS
UB 490 :: GOSUB 530 :: PRINT
 #1,USING 250:J$(1),H(1),K$(
1),L$(1),J$(2),H(2),K$(2),L$
(2):: E=E+2 :: GOTO 380
420 GOSUB 490 :: GOSUB 530 :
: GOSUB 570
430 PRINT #1,USING 240:J$(1)
,H(1),K$(1),L$(1),J$(2),H(2)
,K$(2),L$(2),J$(3),H(3),K$(3
),L$(3):: E=E+3 :: GOTO 380
440 PRINT #1: :: PRINT #1:CH
R$(18);CHR$(27);CHR$(50):: C
LOSE #2 :: CLOSE #1 :: DISPL
AY AT(20,1):"Want another Co
py or Disk? Y": :""
450 ACCEPT AT(20,28)SIZE(-1)
VALIDATE("YN")BEEP:M$ :: IF
M$="Y" THEN GOSUB 610 :: GOT
O 470 ELSE CALL CLEAR
460 OPEN #1:"PIO" :: PRINT #
1:CHR$(27);"T" :: CLOSE #1 :
: STOP
470 DISPLAY AT(20,1)BEEP:"
INSERT DISK INTO DRIVE "&STR
$(B): :"  PRESS ANY KEY TO
BEGIN  "
480 CALL KEY(0,J,K):: IF K<1
 THEN 480 ELSE 270
490 IF G(1)>0 THEN L$(1)=" "
 ELSE L$(1)="Y"
500 IF ABS(G(1))=5 THEN K$(1
)=F$(5):: RETURN ELSE A=LEN(
STR$(I(1)))
510 IF A=1 THEN K$(1)=F$(ABS
(G(1)))&"  "&STR$(I(1)):: RE
TURN
520 IF A=2 THEN K$(1)=F$(ABS
(G(1)))&" "&STR$(I(1)):: RET
URN ELSE K$(1)=F$(ABS(G(1)))
&STR$(I(1)):: RETURN
530 IF G(2)>0 THEN L$(2)=" "
 ELSE L$(2)="Y"
540 IF ABS(G(2))=5 THEN K$(2
)=F$(5):: RETURN ELSE A=LEN(
STR$(I(2)))
550 IF A=1 THEN K$(2)=F$(ABS
(G(2)))&"  "&STR$(I(2)):: RE
TURN
560 IF A=2 THEN K$(2)=F$(ABS
(G(2)))&" "&STR$(I(2)):: RET
URN ELSE IF A=3 THEN K$(2)=F
$(ABS(G(2)))&STR$(I(2)):: RE
TURN
570 IF G(3)>0 THEN L$(3)=" "
 ELSE L$(3)="Y"
580 IF ABS(G(3))=5 THEN K$(3
)=F$(5):: RETURN ELSE A=LEN(
STR$(I(3)))
590 IF A=1 THEN K$(3)=F$(ABS
(G(3)))&"  "&STR$(I(3)):: RE
TURN
600 IF A=2 THEN K$(3)=F$(ABS
(G(3)))&" "&STR$(I(3)):: RET
URN ELSE IF A=3 THEN K$(3)=F
$(ABS(G(3)))&STR$(I(3)):: RE
TURN
610 DISPLAY AT(24,7):"CATALO
G DRIVE: 1" :: ACCEPT AT(24
,23)BEEP SIZE(-1)VALIDATE("1
234"):B :: RETURN
620 INPUT #2:J$(1),G(1),H(1)
,I(1):: GOSUB 490 :: PRINT #
1,USING 250:J$(1),H(1),K$(1)
,L$(1):: GOTO 440
630 END
```

```
**************************************************************************************
DISKNAME: NUTS&BOLTS   FREE:   12   USED: 346
SINGLE SIDED/SINGLE DENSITY          FEB 14, 1987
======================================================================================
 Filename   Size  Type   P    Filename    Size  Type   P    Filename    Size  Type   P
----------  ----  ------ -   ----------   ----  ------ -   ----------   ----  ------ -
ALTERNATE      6  D/V163     AVERAGE         2  D/V163     BELL            2  D/V163
BIGCHAR        3  D/V163     BIGCHAR2        3  D/V163     BINARYDEC       3  D/V163
BORDER         2  D/V163     CALENDAR        2  D/V163     CALLBELL        3  D/V163
CARDS          4  D/V163     CHAMELEON       3  D/V163     CHARFACE        4  D/V163
CHARSAVE       2  D/V163     CHARSAVE2       3  D/V163     CHARSET2        8  D/V163
CHECK          2  D/V163     CHECKSAY        2  D/V163     CLOCK           2  D/V163
COLORTEXT      2  D/V163     COLUMNIZER      2  D/V163     COUNTER         2  D/V163
CRAWL          2  D/V163     CURSOR          4  D/V163     CURSORCONT      3  D/V163
CURTAIN        2  D/V163     CURTAIN2        3  D/V163     DIFFERENT       2  D/V163
FACTORIAL      2  D/V163     FADE-IN         3  D/V163     FADE-OUT        3  D/V163
FAILSAFE       2  D/V163     FASTJOY         2  D/V163     FLAG   [illegible]  D/V163
FLASH          2  D/V163     FLASH-ON        2  D/V163     FORMATTER [illegible]  D/V163
HEX-DEC        2  D/V163     HIGHLOW         2  D/V163     HIGHSCORE [illegible]  D/V163
HIGHSCRAM      3  D/V163     HOLD            7  D/V163     INTERLUDE [illegible]  D/V163
INVERSE        3  D/V163     JIGGLE          2  D/V163     JOYSELECT [illegible]  D/V163
JOYWRAP        2  D/V163     KEYBOARD        2  D/V163     KILLQUIT [illegible]  D/V163
LARGECHAR      9  D/V163     LOAD           21  PROGRM     LONGSHELL [illegible]  D/V163
LONGSHELLN     2  D/V163     LOWERCASE       8  D/V163     MAJORSCALE [illegible]  D/V163
MINORSCALE     2  D/V163     MIRROR          3  D/V163     MONGOLIAN [illegible]  D/V163
MONTH          2  D/V163     MUSIC           3  D/V163     NO-REPEAT [illegible]  D/V163
NOTICE         7  PROGRM     OUTSIDE-IN      2  D/V163     PACKING [illegible]  D/V163
PACKMUN        4  D/V163     PASSWORD        2  D/V163     PAUSE [illegible]  D/V163
PLAY           2  D/V163     PLAYORGAN       2  D/V163     PRICE [illegible]  D/V163
QUICKSORT      2  D/V163     QUICKSORTN      2  D/V163     READCHAR [illegible]  D/V163
READNUM        2  D/V163     READPACK        2  D/V163     READSCREEN [illegible]  D/V163
RESORTNUM      2  D/V163     RESORTSORT      2  D/V163     RUSSIAN [illegible]  D/V163
SAVESTRING     2  D/V163     SCALE           2  D/V163     SCREENGRID [illegible]  D/V163
SCREENSAVE     3  D/V163     SCRUNCH         3  D/V163     SECTIONS [illegible]  D/V163
SHOEHORN       2  D/V163     SHOEHORN-N      2  D/V163     SHUTOFF [illegible]  D/V163
SLANT          9  D/V163     SLASH           2  D/V163     SLASHIERD [illegible]  D/V163
SPRITESHOW     3  D/V163     TAKETURNS       3  D/V163     TIGER [illegible]  D/V163
TITLE          2  D/V163     TUTORIAL       44  PROGRAM    TWODIE [illegible]  D/V163
TWOWAY         2  D/V163     UPDOWNWIPE      2  D/V163     UPSIDEDOWN      2  D/V163
WAIT           2  D/V163     WAITING         3  D/V163
```

SINGLE CHIP 32K MEMORY EXPANSION..

By Joe Spiegel of the Airport Area Computer Club

Joe is still doing it. He has decoded around the ROM in the console using diodes in order to make the use of the new 32K BYTE (single) Chip, the 62256 He has also designed a single sided board that can be etched by you, in order to build this project.

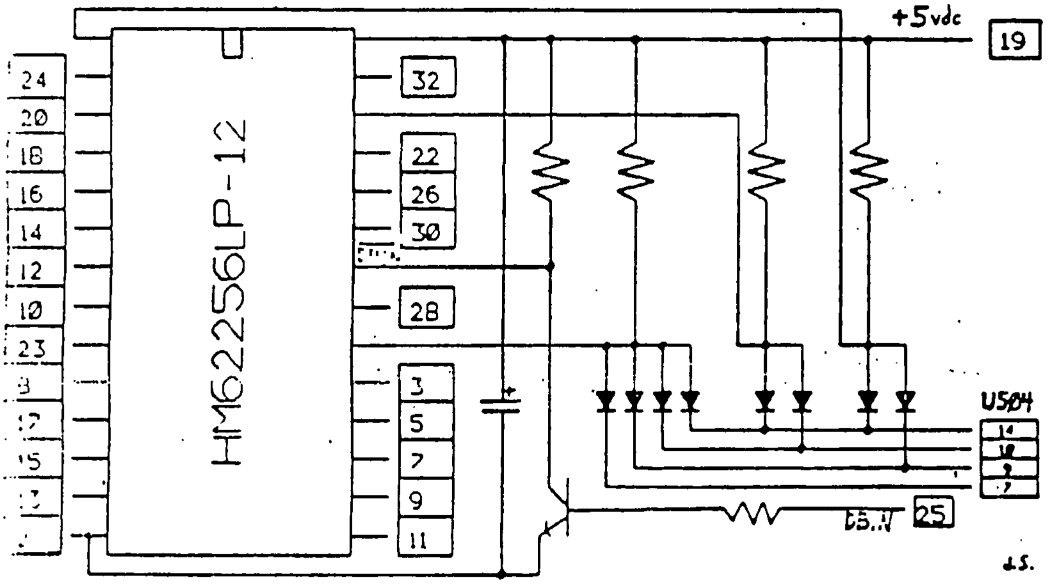
Joe will either send you an etched board for $3.00 (unbelievable), or a complete unit ready to solder (4 wires to the U504 chip in your console), for and get this, $25.00. Hey Joe! Your ruining the neighborhood. You know that for him to do it at these prices, Joe is " doing it for YOU ". He will need your old GROM connector back after you install the unit he builds for you.

Joe, does your wife know that the family will not see you for the next two years. Send inquiries to the :

AIRPORT AREA COMPUTER CLUB
% JOE SPIEGEL
P.O. BOX 710
CORAOPOLIS, PA 15108

OR CALL JOE AT: 412 457-8284

SINGLE RAM CHIP 32K Expansion

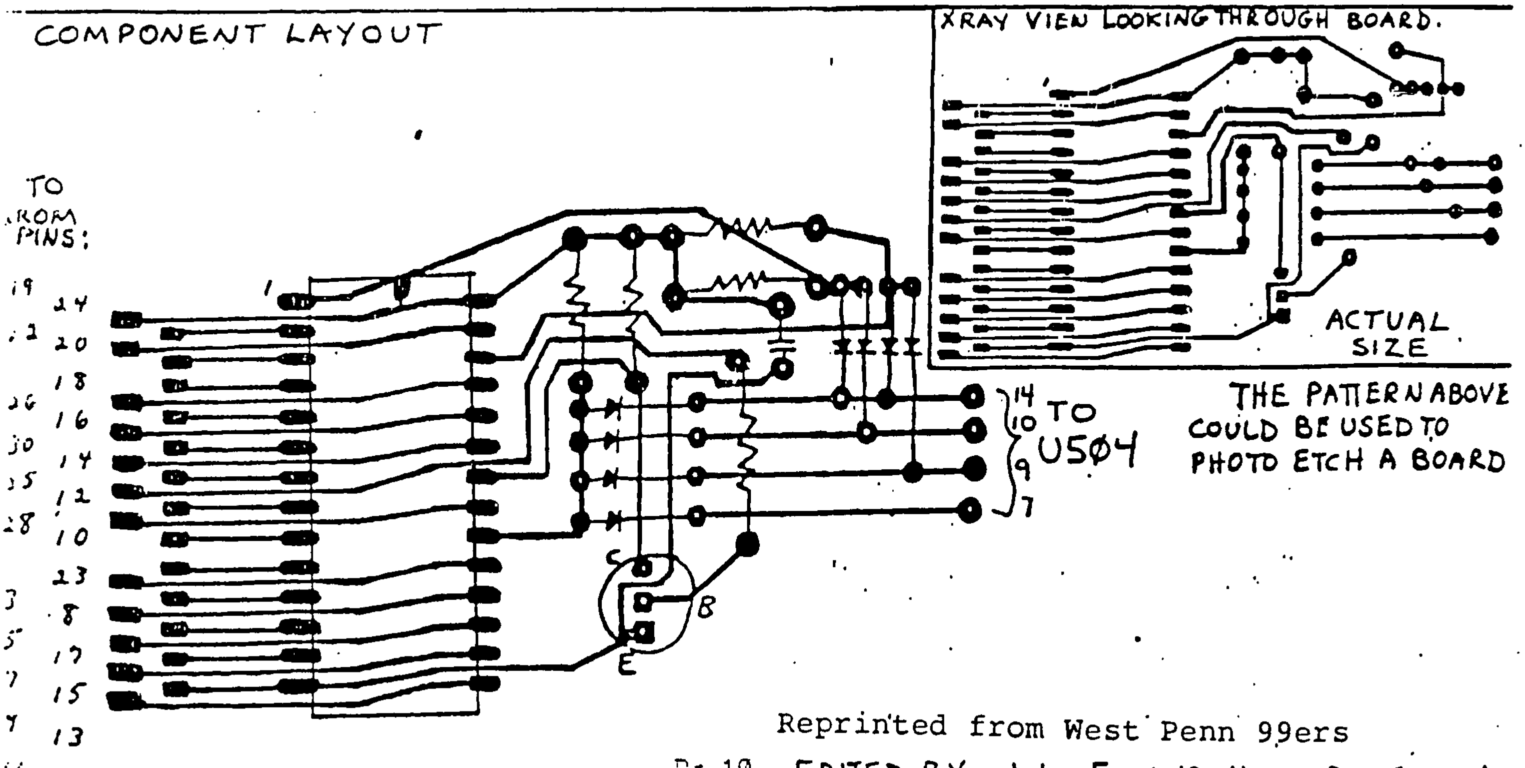


Notes:

All resistors - 1K
All diodes - 1N914 or 1N34
Transistor - 2N2222 or 2N3904
Capacitor - 22 mfd Tantalum

All pin numbers refer to connections on the GROM port except the four lines going to U504 on the motherboard.

The transistor and its base and collector resistors may be omitted if $\overline{\text{DBIN}}$ is obtained from U508, pin 9, on the motherboard.



THE PATTERN ABOVE COULD BE USED TO PHOTO ETCH A BOARD

Reprinted from West Penn 99ers