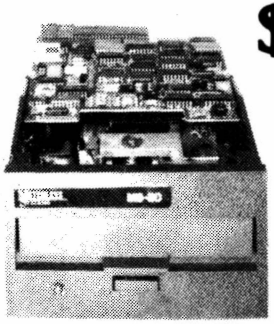
MONTHLY NEWSPAPER SUPPORTING THE TRS-80

S-80 COMPUTING

Vol 1, No. 5 March, 1980 20 pages $1.50

Cobol compiler for only $99

See page 6

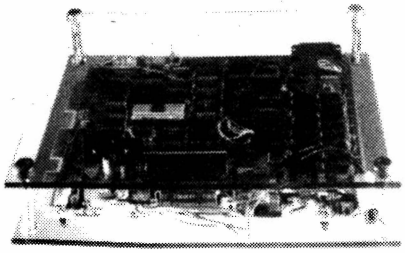


$69 module acts like floppy

See page 3

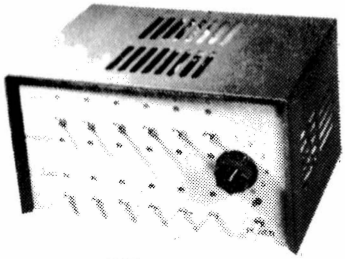
CIE tests $265 disk drive

See page 5



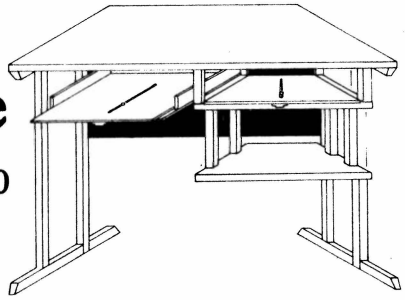
64K & greater mod 1 memory

See pages 2, 3, 6



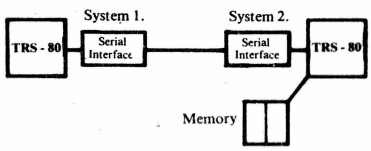
Connect '80s together for power

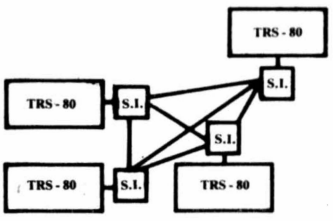
See pages 9, 10-11



Plans: desk gives work space

See pages 18, 19, 20





80K-bytes in your model 1

By GENE LU, partner
LNW Research, 8 Hollowglen st., Irvine CA 92714

The LNW Research **system expansion board** is designed to replace the Radio Shack expansion interface, in a low-cost, modular, and upward-compatible manner. Any section (i.e. 32K-byte memory, serial-RS232-20MA I-O, parallel-printer port, floppy controller, dual-cassette port, and four-voltage power supply) can be built individually, up to the full configuration of the board.

Now, with many S-80 users needing additional memory and two serial ports, it has been proposed to bank switch an additional 32K bytes of memory and an additional serial port.

Due to the low cost of the LNW Reserch system expansion board, a partially-populated board with RAM, serial port and the necessary banking logic can be added to an existing system with either the Radio Shack expansion interface or LNW Research's system expansion board, without modification to the existing system.

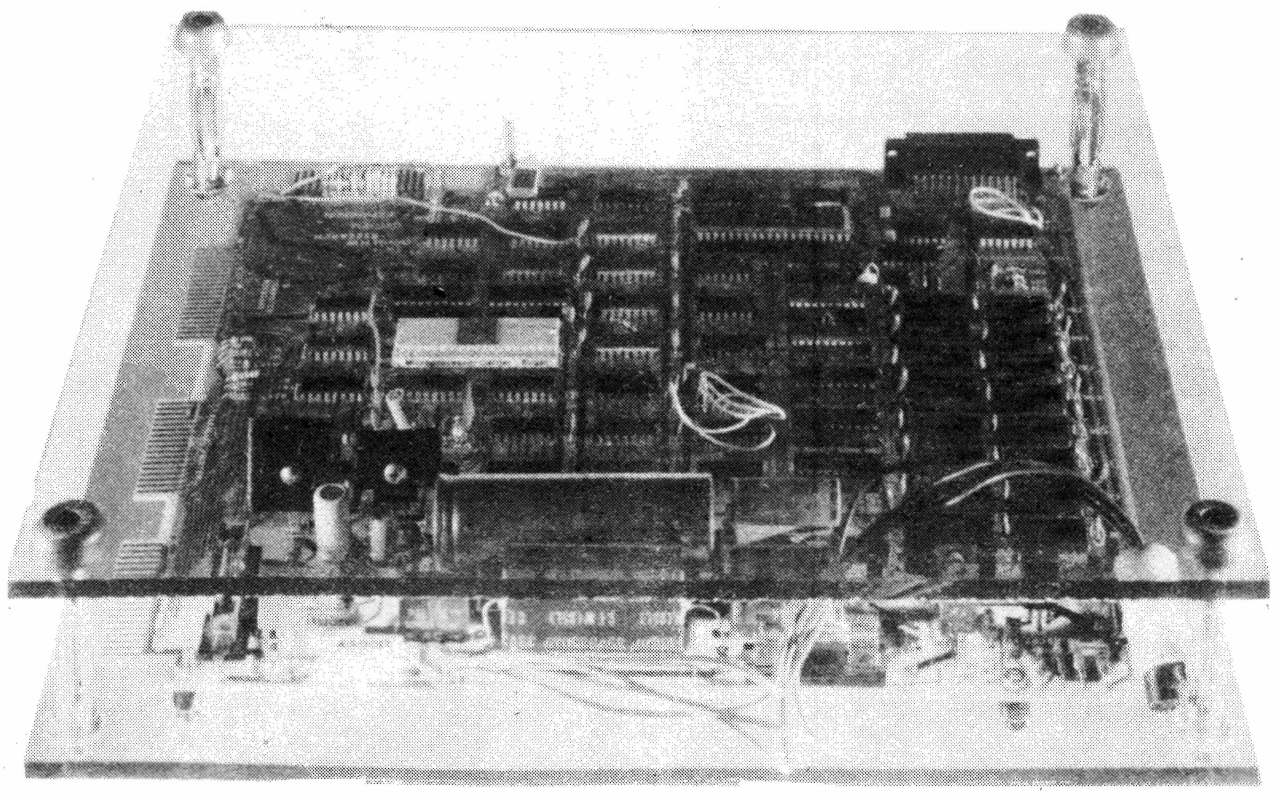
Software will control banking with a simple output command from Basic or machine-code programs. The serial ports will both respond to the existing software without any patches. Peripherals responding to port addresses other than E8-EF will not be affected.

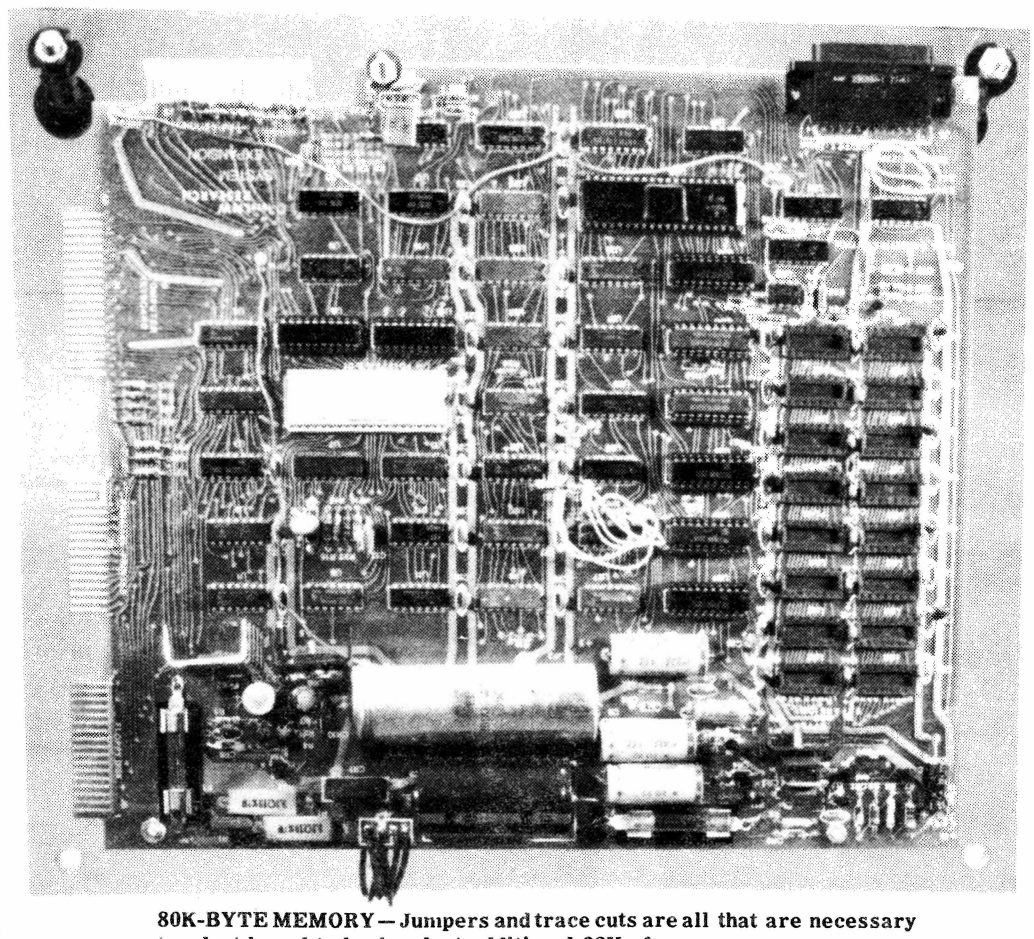
The proposed banking command in Basic would be:
OUT 236,1 (select the alternate 32K bytes of memory)
OUT 236,0 (restore to select the original 32K bytes of memory)
OUT 237,16 (select the alternate serial port)
OUT 237,0 (restore to select the original serial port)

On power-up the original memory and serial port are selected. **Remember: all modifications will be on the alternate system expansion board, with no modifications to the existing system.**

(Editor's note: Gene tells **Computing** the proposed banking logic all will be contained on the LNW Research **system expansion board**, and will be implemented with trace cuts and jumper wires. LNW Research is adding a section on this in its excellent documentation package.)

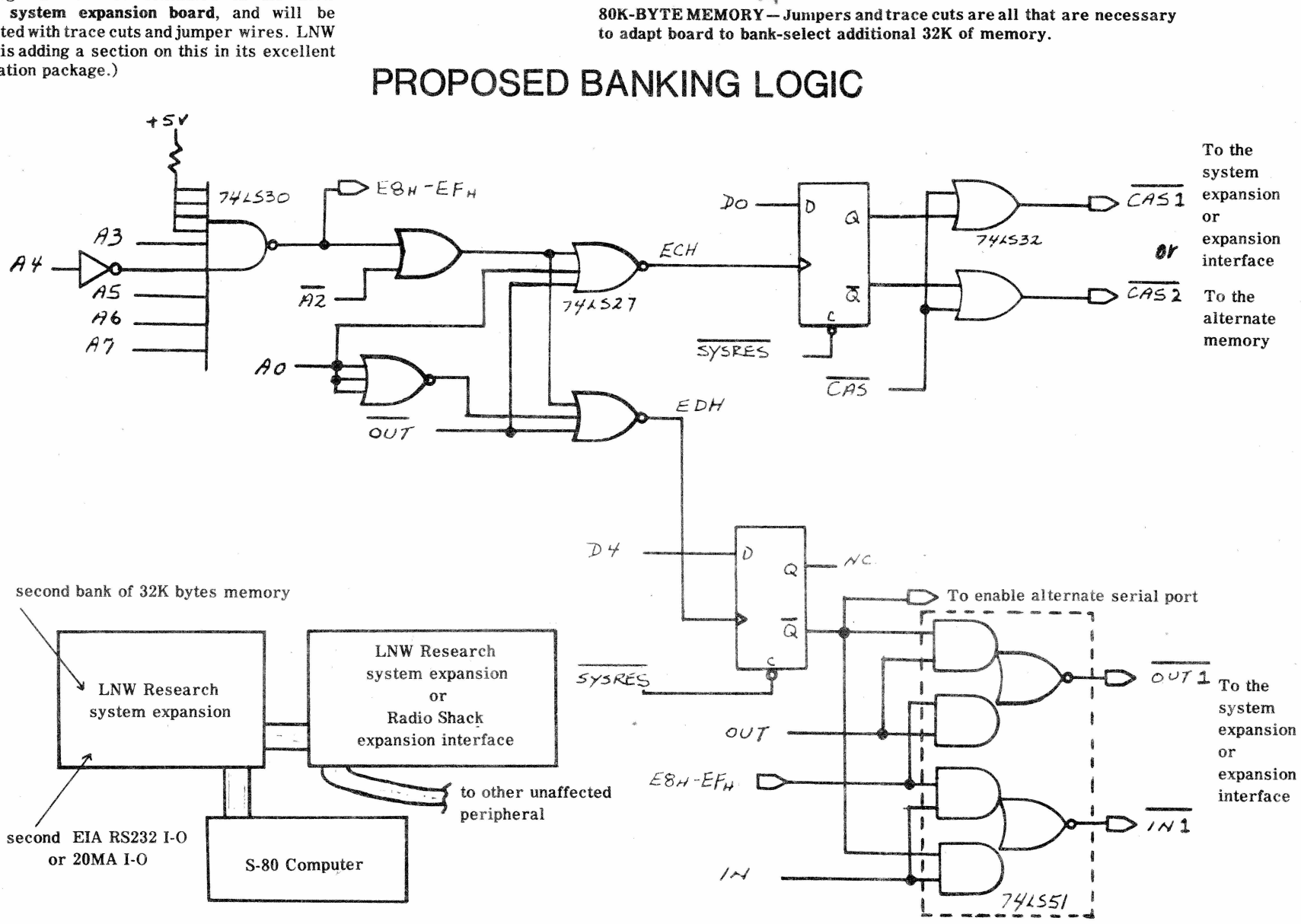
Case for LNW Research expansion memory board can simply be a sandwich of any material. We used smoke-colored plastic, held apart by simple spacers.





80K-BYTE MEMORY— Jumpers and trace cuts are all that are necessary to adapt board to bank-select additional 32K of memory.

PROPOSED BANKING LOGIC



Up to 40 megabyte storage

Lobo Drives Int'l, manufacturers of floppy-disk and hard-disk memory systems for the personal-and-small-business microcomputer market, has announced the addition of an enhanced expansion interface for TRS-80.

Identified as the Model LX80, the new expansion interface has been designed for the serious user who wants to improve and expand the performance and capabilities of his TRS-80. It offers improvements and new features not available on the standard Radio Shack part.

The LX80 stretches system performance by expanding memory **storage** capacity up to 40 megabytes, provides facilities for up to 32K of RAM and offers a second serial port.

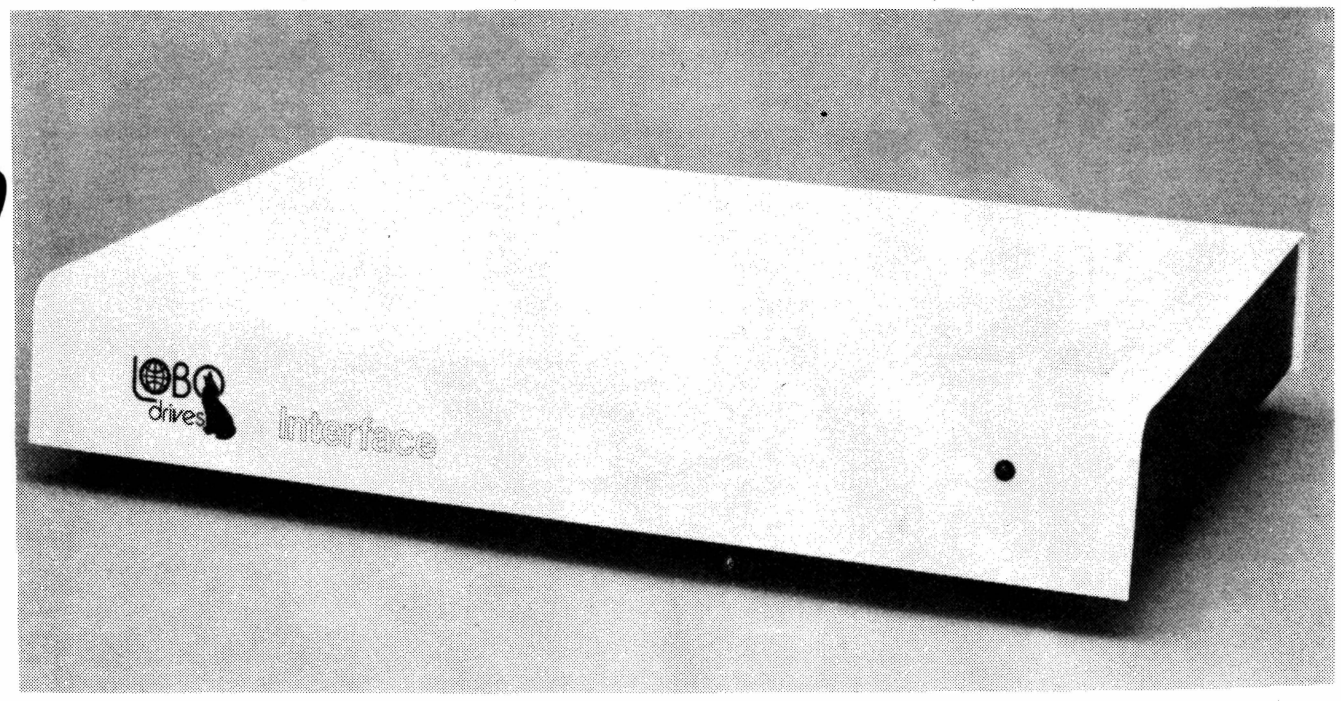
An easily-accessible switch permits overriding the keyboard ROM for booting-in diagnostics and customized operating systems.

Connectors for the 5¼ and 8'' floppy disk drives and other peripheral devices are located on the side and rear panels.

There is a separate, bidirectional parallel port exclusively for the Lobo Drives Model 7710T Winchester hard disk drive.

Other features include: a parallel "Centronics" printer port; screen printer port; two microprocessor-controlled bidirectional serial ports; a crystal-controlled "real-time" clock.

The Model LX80 expansion interface is priced at $525 in unit quantities. Dealer inquiries are invited. Shipment is quoted as 45 days after receiving order. For additional information contact Mike Mock, Lobo Drives, 935 Camino Del Sur, Goleta CA 93017 (805) 685-4546 or (714) 641-1436.



$99 Cobol running on TRS-80 Model 1, CP-M

The most widely-used programming language in the world is now available at a price that makes sense to anyone who can afford disk and CP-M.

Cobol—short for Common Business-oriented Language—has been used extensively world-wide in business applications since the early 1960s.

Cobol is an easy-to-learn, easy-to-program compiler language which features high-performance execution speed while at the same time maintaining highly-readable and easily-maintained program applications.

As its name implies, Cobol is designed for use by business people for business purposes. Cobol programs are written in English sentences and paragraphs, which can be structured in a logical sequence which makes sense to programmer and the businessman (it is self documenting).

An indication of just how common it is is the estimate nearly 300,000 Cobol programmers have written $10 billion worth of Cobol programs. Until now the Cobol language has been reserved for large-scale computer systems used by large businesses.

A practical, reasonably-priced Cobol compiler has been virtually nonexistent in the fast-growing microcomputing industry. That void now has been filled.

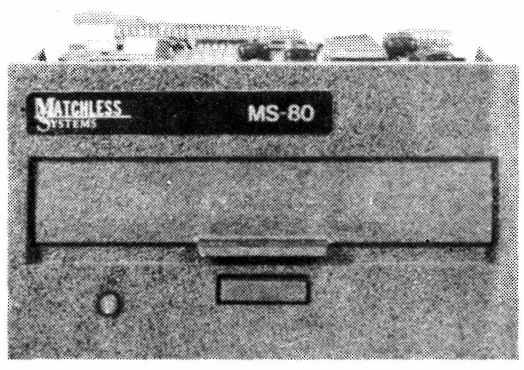
After three years of extensive development and testing, Ellis Computing has released the product that everyone has been waiting for: the super-fast, efficient, easy-to-use Cobol for under $100.

The design goals established from the beginning included the production of the Cobol standardized language, with low price, high-performance both in compilation and execution speed, programming and operating simplicity, and high-efficiency both in terms of memory and disk usage.

All of those goals are said to have been achieved resulting in what is called an outstanding product at an unbeatable price.

Features: Ellis' "Nevada" Cobol (NC) has been implemented in accordance with the American National Standards Institute (ANSI) "American National Standard Cobol 3.23-1974," commonly called ANSI-74. NC is an implementation of a subset of the Cobol language which is completely defined by that specification. The importance of that aspect of NC cannot be overstated, according to its developer. It means that there is a vast wealth of information available supporting the language.

(Continued on page 16)



MPI BARE DRIVES, uncased, no modification for TRS-80 ..... $265
You make modification, our instructions, or we do $10
You power, or buy assembled & tested supply $39.95
All drives tested, aligned, GUARANTEED 90 DAYS!
Buy our supply and we set speed to your power
Dual-drive cables $24.50 Four-drive cables $34.50

MATCHLESS SYSTEMS

TELEPHONE (213) 327-1010
18444 S. Broadway GARDENA, CALIFORNIA 90248

WE'RE HAPPY TO BE BACK

We're happy to be back, after essentially a year's absence.

Unlike Rip Van Winkle, we've been aware of the constant changes that have been going on in our world of computing, and we've been working from 8 a.m. until 11 p.m. trying to adapt to them.

We've accomplished a lot. When last seen, we weren't sure whether we were a newsletter or magazine. We didn't very well fit either discription, and usually tried to evade the issue by using the term "publication". Yet that did not mask the problem: we didn't know what we were.

We've solved our identity problem. We're none of the above.

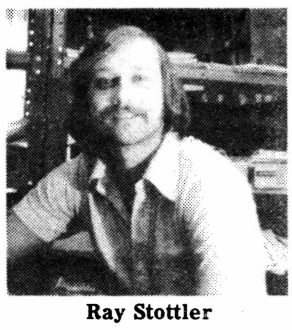
It may be a cliche, but important developments happen so rapidly in S-80 computing that it is nearly impossible to keep current. Why be spending $400 for disk, if a $265 system is what you really would have wanted, had you known about it? Why write that important program in Basic, just because you didn't know Cobol was inexpensively available? Why buy disk at all, if tape could as easily, or better, suit your needs. Maybe the $69 "Poor Man's Floppy" would suit you as well as more than $700 worth of disk. ETC!

Needed is a medium that offers fast turnaround of the latest news in S-80 computing. CIE has spent the past year preparing to bring out the first biweekly S-80 computing newspaper pair.

S-80 Computing subscribers will be getting the newspaper monthly, and will also be getting, between issues, S-80 Bulletin. Computing will carry the more important stories, the Bulletin smaller reports and late-breaking developments. We'll try and keep you never more than two weeks away from a fill-in on what is happening in S-80 computing.

All are full time, except for half-time Kadaw, Rawson and Abbott.

Bill McLaughlin
editor, chief snake handler



Ray Stottler
advertising, fulfillment

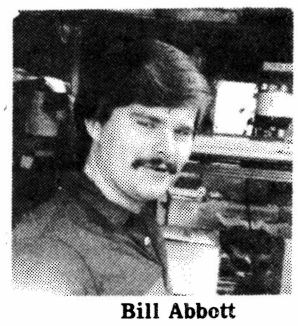
Julie Barrett
People's Software programmer

Paul Kadaw
systems programmer

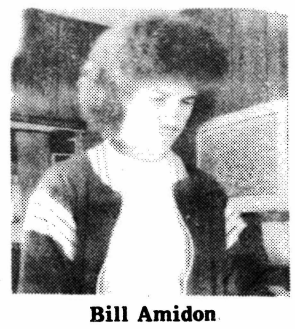
Bob Yocum
production, detail, fulfillment

Shirley Maddux
bookkeeper, fulfillment mgr.

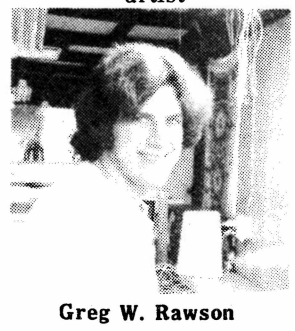
Margaret Mula
data entry



Bill Abbott
People's Software production



Bill Amidon
artist



Greg W. Rawson
technical assistant

SPEED disk

By STEPHEN GIBSON, copyright 1979
Gibson Engineering, 547 N. Beachwood dr.,
Los Angeles CA 90004

Remember when loading a Level I program took several minutes? The fast speed of disk I-O ended all that...or did it? Several seconds now seems an eternity. So...pressing forward to that day when we have everything resident in our $2-gigabyte RAM, here is a simple mod that will really soup-up your disk I-O speed to well beyond Warp 8. It's all in software, so you can put those soldering irons away.

After some research on the inner workings of the FD77B disk-controller chip (that big IC in the expansion interface), I made the discovery that you can adjust the stepping speed of your disk drive with software. Luckily, all this happens at one memory-mapped I-O location, 37EC (hex). It is here that your DOS or Basic stuffs speed values to program the disk controller chip. The lower two bits are all that are used to control speed. After some fooling around on several systems, I soon learned that not all drives can dance the F-A-S-T boogie. So you must experiment. Run this program in Basic:

```
2930 POKE10 '-- SPEEDISK --
20 ' A SIMPLE DISC SPEEDUP (C)1979
     STEPHEN GIBSON
30 M = 18007 : A = PEEK (M)
40 IF A = 30 THEN POKE M, 30 ELSE
     IF A = 27 THEN POKE M,26
50 CLS : PRINT "SUPERDISK OPERATION
     AL"
```

Then save this program to disk and note the I-O speed difference. The head should really buzz across disk during the save operation. Klinking sounds are a no-no. In fact, a klinking sound means this mod won't work and you should return to deep space shootouts at klingons. If all goes well you should try loading the program back in again to check for any loading problems. No errors? Great. You've just ended that "Shugart Shuffle". Let's go for a more lasting solution by DOS modification.

SPEEDISK for NEWDOS:

1) Remove the write protect tab thingy and load Basic.
2) Run "SUPERZAP".
3) Do "Display Disk Sectors" and select the disk you want to lobotomise.
4) Jump over to track zero "0", sector eight "8".
5) On the bottom line (about FE bytes down), the last 4 bytes of hex code should read "1BF5". If not, advance to the next sector (number 9) by hitting the ";" key and see "1BF5" in the top row. At either location we want to see that "1B" changed to "1A". So...
6) Type "MODFE" which should stick a + (plus) sign next to the "1BF5". If you found the "1B" over on sector 9, type "MODOF" and then type "1A" and see that it is displayed in place of the "1B".
8) Hit "enter" and exit to whatever after writing the disk.
9) You should have seen "1AF5" end up on that line. If not, it's time to read-up on SUPERZAP.

AND TRSDOS:

"Old DOS" users (2.1) can find the SPEEDISK value on track 0, sector 7, relative byte 52 (that's the sixth row down). Replace the 1B with a 1A as before. Now 2.1 runs rings around the new 2.2 (yuk!). What's that? You still want 2.2? Okay...go to track 0, sector 7, byte 52 again. There you'll see a "1F". Replace it with "1E".

This SPEEDISK mon also might be the great psychological fix of 1980 in as much as it "seems to reduce disk I-O errors." You know the ones. You're loading a program and you hear the head begin to scurry all over the disk for that last sector. Will it laod? Crossed fingers, arms, legs, and eyes don't help. But Speedisk does seem to reduce that "period of anticipation".

TO RECAP SPEEDISK PATCHES:

NEWDOS—Track 0, sector 8, byte FE.
Change the 1B to 1A
OR—Track 0, sector 9, byte 0F.

DOS 2.1—Track 0, sector 7, byte 52
change the 1B to 1A.

DOS 2.2—(Watch the value), track 0, sector 7, byte 52.
Change the 1F to 1E

If you know of any others, please let us know.

WARNING—As "factory specs" go, you are not supposed to increase the stepping rate of a Shugart (Radio shack's drive) without catastrophic results, or so I'm told. They specify 10 milliseconds. That 1B in the software stuffs a 11 into the two lowest bits of port 37EC to make sure that is exactly what will happen. Lowering the interval to eight milliseconds with 1A (sticking 10 into the two low bits) runs the risk of writing before you should. And we all know that laying bytes down in the wrong place is clearly a no-no. But, if your drive is rather new (whose isn't?), or if you are lucky enough to be running an MPI drive (or perhaps one of the others), you can use the SPEEDISK mod with little fear. I would be remiss in not telling you that you just "might" have a problem.

Still feeling adventurous? Okay, here are the bit positions needed to set the stepper speed even faster. Some drives willdo it. Others will let out a clink-clink sound which is a lead-pipe cinch you've gone too far.

Stepping Motor Bit Rate (bits 1-0)
00 - 6 ms between steps (patch 18h)
01 - 6ms between steps (patch 19h)
10 - 8ms between steps (patch 1Ah)
11 - 10ms between steps (patch 18h)

As we said before, not all drives can dance the F-A-S-T boogie. So Caveat Emptor!

Advertise in S-80 Computing

Program lets TRS-80 keys repeat

By HUBERT F. STURGES
519 E. Minnesota, Turlock CA 95380

Here are two short programs that demonstrate that the TRS-80 can repeat keys on command.

```
100  A$=INKEY$: IF A$="" THEN 100
110  FOR X=1 TO 128
120  PRINT A$;
130  IF INKEY$<>"" GOTO 100
140  FOR Y=1 TO 20: NEXT Y
150  NEXT
```

Line 100 rapidly scans the keyboard. When a character is entered the program drops to the FOR-NEXT loop for execution. Line 120 prints up to 128 of the characters entered. Line 130 is another keyboard scan which is ignored unless another character is hit. This loops the program back to 100, but requires you to hit the character a second time to execute repeat printing of the new character entered. Line 140 slows the repeat printing action so you can control it.

I chose to use a FOR-NEXT loop so to limit any repeat action to 128 X. By eliminating line 110, and changing 150 to read GOTO 120, the program will run indefinitely.

```
110  A$=INKEY$: IF A$="" THEN 110 ELSE
     140
120  A$=INKEY$: IF A$<>"" GOTO 140
125  FOR Y=1 TO 20: NEXT Y
130  IF B=1 GOTO 180
140  IF A$="0" THEN 110
150  A=1: B=0: PRINT A$;
160  B$=INKEY$; IF B$<>"" GOTO 180
165  FOR Y=1 TO 20: NEXT Y
170  IF A=1 GOTO 140
180  IF B$="0" THEN 110
190  A=0: B=1: PRINT B$;
200  GOTO 120
```

Line 110 rapidly scans the keyboard and is locked in an endless loop until the first character is entered. Execution then drops to line 140, which returns action to line 110 an "Ø" is entered.

Line 140 and line 180 are similar, and provide a way to stop repeat execution by entering "Ø".

If execution is not stopped, line 150 sets flag "A" to 1, and flag "B" to Ø; and prints ("stutters") out the character entered.

Line 160 scans the keyboard and gives you the opportunity to enter another character which will be repeat printed immediately, by passing action to the second GOTO loop. Assuming another character is not yet entered, line 165 is a short timing loop to slow execution of repeat action to a manageable speed. Line 170 completes the first GOTO loop, and sends action back to line 140.

Lines 120, 125, 130, 180, 190, and 200 make up the second GOTO loop and perform identical functions as similar lines in the first loop. The second loop allows you to enter another character at any time for immediate repeat printing.

The "flags" are probably unnecessary for the program as listed. However, they do permit some interesting effects:

Try these two lines:

```
145 IF B=1 PRINT "-HI-";
185 IF A=1 PRINT "--HELLO--";
```

These programs make a good demonstration, and you will have fun with them. . . Now if someone could put them into machine language. . . and maybe fix up a little red button. . .

Jumbleword solver

```
1 REM JUMBLEWORD SOLVER BY TOM ORR
5 REM 249 JUANITA WAY - PLACENTIA, CA 92670
6 REM  SEPTEMBER 78  TRS-80 USERS GROUP
10 CLS
15 DIM A(12)
20 PRINT"                JUMBLE-WORD SOLVER"
25 PRINT
30 PRINT"THIS PROGRAM WILL HELP YOU TO SOLVE THE "
35 PRINT"JUMBLE WORD PUZZLE THAT APPEARS IN MANY "
40 PRINT"NEWPAPERS.  JUST ENTER THE NUMBER OF "
45 PRINT"LETTERS IN THE JUMBLE-UP WORD FOLLOWED BY "
50 PRINT"THE ACTUAL LETTERS AND THE COMPUTER WILL "
55 PRINT"TRY TO STRAIGHTEN THEM OUT.  THE CORRECT "
56 PRINT"WORD WILL USUALLY COME UP IN LESS THAN A "
57 PRINT"COUPLE OF MINUTES."
60 INPUT"PRESS ENTER WHEN READY";Z$
70 CLS
80 INPUT"HOW MANY LETTERS";Y
90 IF (Y>12) OR (Y<5) THEN
     PRINT"5 TO 12 ONLY, PLEASE":GOTO 80
100 PRINT"WHAT ARE THEY
    (WITH COMMAS BETWEEN LETTERS,PLEASE)"
110 ON Y GOTO 90,90,90,90,120,130,140,
            150,151,152,153,154
120 INPUT A$,B$,C$,D$,E$:GOTO 160
130 INPUT A$,B$,C$,D$,E$,F$:GOTO 160
140 INPUT A$,B$,C$,D$,E$,F$,G$:GOTO 160
150 INPUT A$,B$,C$,D$,E$,F$,G$,H$:GOTO 160
151 INPUT A$,B$,C$,D$,E$,F$,G$,H$,I$:GOTO 160
152 INPUT A$,B$,C$,D$,E$,F$,G$,H$,I$,J$:GOTO 160
153 INPUT A$,B$,C$,D$,E$,F$,G$,H$,I$,J$,K$:GOTO 160
154 INPUT A$,B$,C$,D$,E$,F$,G$,H$,I$,J$,K$,L$:
      GOTO 160
160 CLS
170 A(1)=0:A(2)=0:A(3)=0:A(4)=0:A(5)=0:A(6)=0:
    A(7)=0:A(8)=0:A(9)=0:A(10)=0:A(11)=0:A(12)=0
180 FOR I=1 TO Y
190 X=RND(Y)
200 IF A(X)=9999 THEN 190
210 PRINT CHR$(23);
220 ON X GOTO 300,310,320,330,340,350,360,
            370,380,390,400,410
230 A(X)=9999
240 NEXT I
250 ON Y GOTO 90,90,90,90,260,270,280,
            290,291,292,293,294
260 PRINT"    ";:GOTO 170
270 PRINT"   ";:GOTO 170
280 PRINT"  ";:GOTO 170
290 PRINT"       ";:GOTO 170
291 PRINT"      ";:GOTO 170
292 PRINT"     ";:GOTO 170
293 PRINT"    ";:GOTO 170
294 PRINT"   ";:GOTO 170
300 PRINT A$;:GOTO 230
310 PRINT B$;:GOTO 230
320 PRINT C$;:GOTO 230
330 PRINT D$;:GOTO 230
340 PRINT E$;:GOTO 230
350 PRINT F$;:GOTO 230
360 PRINT G$;:GOTO 230
370 PRINT H$;:GOTO 230
380 PRINT I$;:GOTO 230
390 PRINT J$;:GOTO 230
400 PRINT K$;:GOTO 230
410 PRINT L$;:GOTO 230


 50  REM JUMBLE WORD VERSION 2
 70  REM SEPTEMBER 78 TRS-80 USERS GROUP
100 DIM A$(6)
200 CLS:PRINT"LETTERS"
250 PRINT"THIS PROGRAM PRINTS POSSIBLE COMBINATIONS"
260 PRINT"OF 2 TO 6 LETTERS AT RATES OF TEN WORDS"
270 PRINT"PER SECOND FOR SIX LETTERS"
300 I=0
350 PRINT"INPUT LETTERS ONE PER ? FOLLOWED BY ENTER"
375 PRINT"TYPE 'END' AFTER LAST LETTER"
400 INPUT A$(I)
500 IF A$(I)<>"END" GOTO600
550 I=I-1
560 INPUT"WORD RATE PER SECOND 1-10";W
565 IF (W>10) OR (W<1) GOTO 560
570 GOTO 1000
600 I=I+1
700 GOTO 400
1000 IF I<5 THEN 2000
1100 FOR E=5 TO 0 STEP -1
1200 J=5
1300 GOSUB 10000
2000 IF I<4 THEN 3000
2100 FOR D=4 TO 0 STEP -1
2200 J=4
2300 GOSUB 10000
3000 IF I<3 THEN 4000
3100 FOR C=3 TO 0 STEP -1
3200 J=3
3300 GOSUB 10000
4000 IF I<2 THEN 5000
4100 FOR B=2 TO 0 STEP -1
4200 J=2
4300 GOSUB 10000
5000 FOR A=1 TO 0 STEP -1
5100 J=1
5200 GOSUB 10000
5300 GOSUB 20000
5400 NEXT
5600 IF I>1 NEXT
5700 IF I>2 NEXT
5800 IF I>3 NEXT
5900 IF I>4 NEXT
6000 END
10000 B$=A$(0)
10100 FOR K=0 TO J-1
10200 A$(K)=A$(K+1)
10300 NEXT
10400 A$(K)=B$
10500 RETURN
20000 FOR K=I TO 0 STEP -1
20100 PRINT A$(K)
20200 NEXT
20400 PRINT
20450 FOR Z=0 TO 1/W[3*300
20460 NEXT Z
20500 RETURN
```

lowercase exercise

By DAN LIKINS
206½ S Birch, Santa Ana CA 92701

```
10  CLS:OUT 255,24
20  FOR C=15360 TO 16383 STEP 2
30  POKE C,0:A$=INKEY$:POKE C,32:IF A$=""THEN 30
35  X=ASC(A$)
40  IF X=8 THEN C=C-2:POKE C,32  :GOTO 30
50  IF X=31 THEN C=16383         :GOTO 80
60  IF X>64 AND X<91 THEN X=X+32 :GOTO 80
70  IF X>91 AND X<129 THEN X=X-32:GOTO 80
80  POKE C,X : NEXT              :GOTO 20
90  REM RUN AND START TYPING
```

CIE tries $265 disk from Matchless

When CIE heard about Matchless Systems selling bare MPI drives for $265, we immediately called the Gardena firm and ordered one. It was here within a day or two.

MPI drives are among the fastest in the industry, and they are super dependable. CIE has had four of them operating continually since July, when we started getting our 16,000 addresses on-system. No real problems.

Back in July "bare" drives would have scared us—we would have worried about the effects of dust and grime. Yes, we have dust in our office! But shortly after installing our first set of drives, we started worrying about overheating (I now don't think they were) and we took the cases off one pair. Another minor problem a few weeks ago and a case came off one on the other pair. So of our four MPI drives, only one now is cased and we eventually will have the cases off all our drives. Makes it easy to "borrow" a drive from one desk to make up a three-drive set on another work station.

The Matchless bare drives are fully checked out, and if you buy the power supply from them, the speed will be set with your power supply. If you supply the power, you must set speed. Here is how:

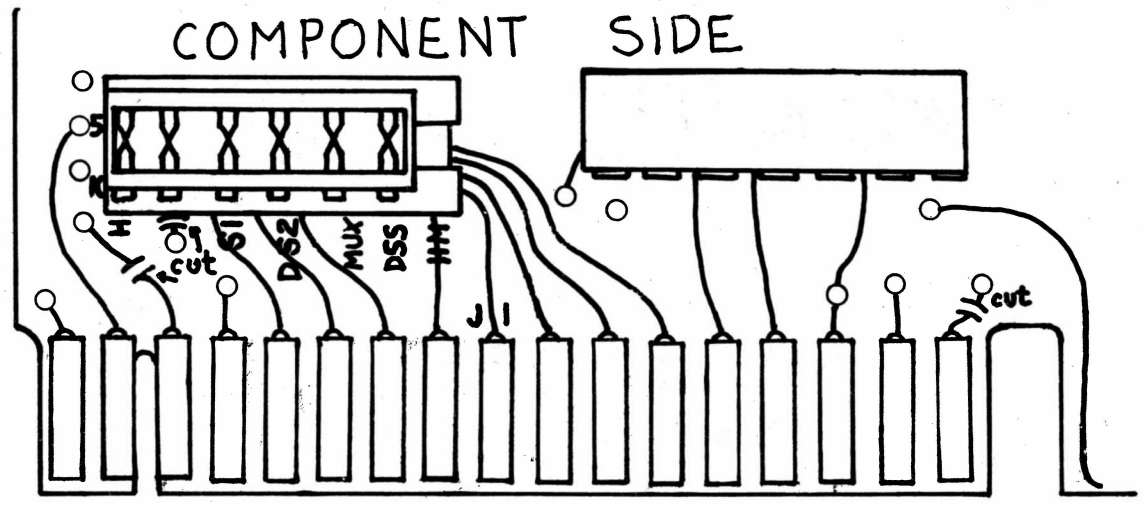
To properly set speed, the drive must be selected (red light on), head loaded and motor on. To get it to do that, plug it into the drive-0 position and hit reset on the TRS-80 keyboard unit. You or a helper, will have to continually hit reset until the job is done. Now put a blank disk in, turn the disk on its side, with the red LED up, and adjust the multiple-turn trim pot on the top of the circuit board until the stroboscope on the flywheel gives a stop-motion effect when viewed under fluorescent light. That is the outside strobe, when using 60 cycle power. The speed control pot is shown with an arrow in an accompanying picture. Lock it with fingernail polish or white glue when you finish.

To use with TRS-80, MPI drives must have three circuit traces cut, and a jumper wire soldered in place. The accompanying drawings show how to do that. Be careful that you don't cut any other traces! A razor will do. Matchless will do that modification for $10, which is a bargain!

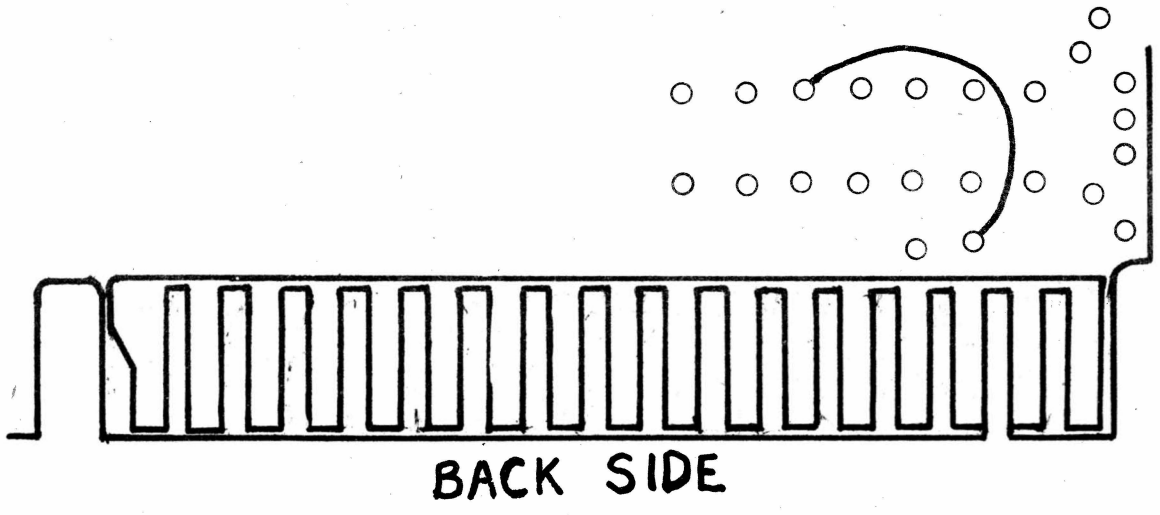
Power previously has been the catch that has kept people from taking advantage of bare-drive prices. You need 1.2A of +12V, and .7A of +5V, both within 5 percent. Throw together something sloppy and you at best have an undependable drive, at worst you've burned up $265. Matchless is selling assembled, tested and guaranteed power for $39.95. You have only to plug it in and you're up and running. We honestly feel this is a bargain.

The cables available from Matchless are compatible with Radio Shack TRS-80 drives. So either Radio Shack or Matchless cables can be used, and Matchless and TRS-80 drives can be mixed in the same system.

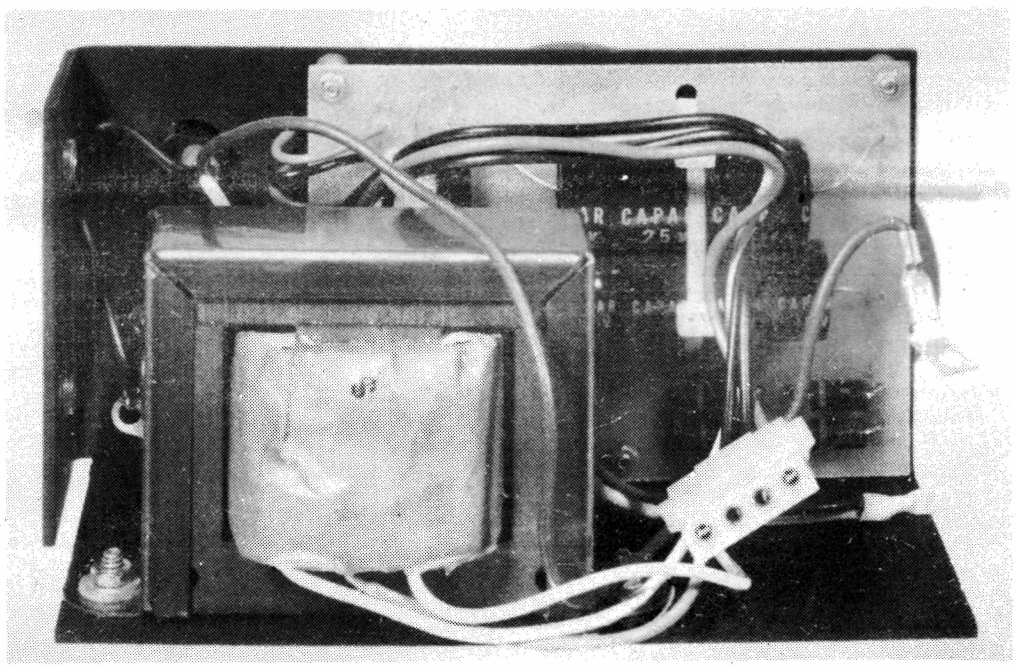
Matchless is at 18444 S. Broadway, Gardena CA 90248, phone (213) 327-1010.

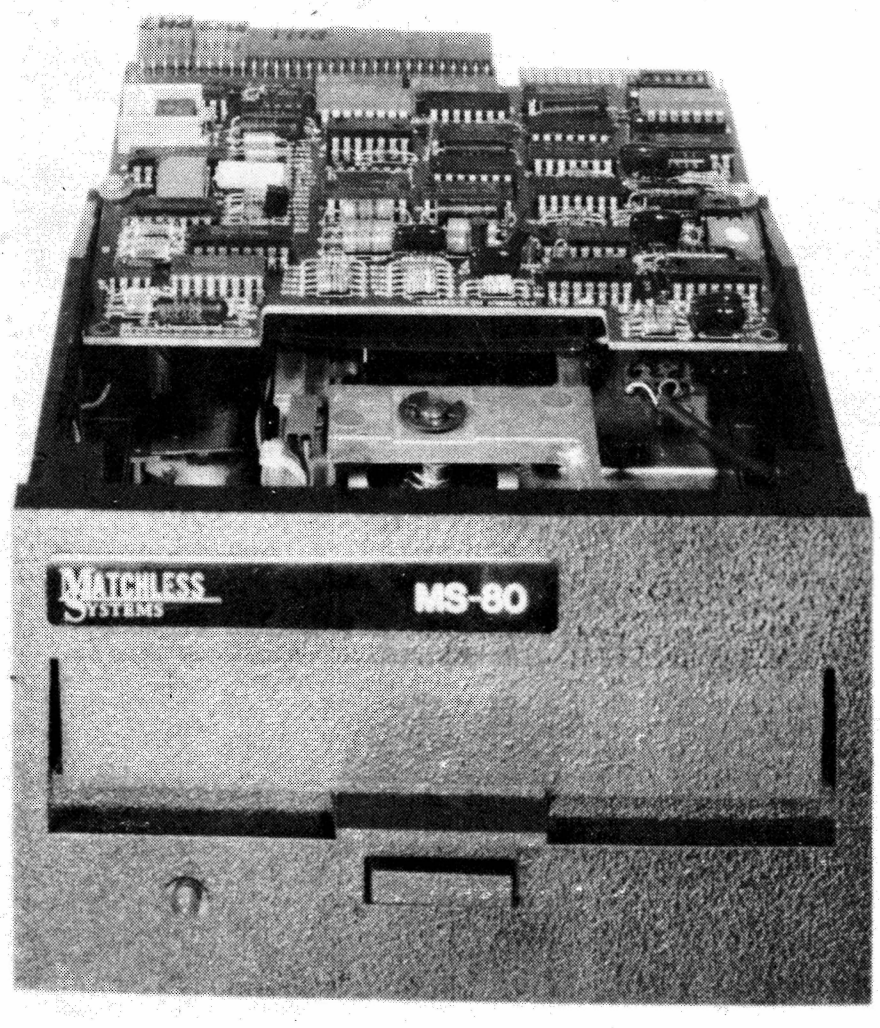


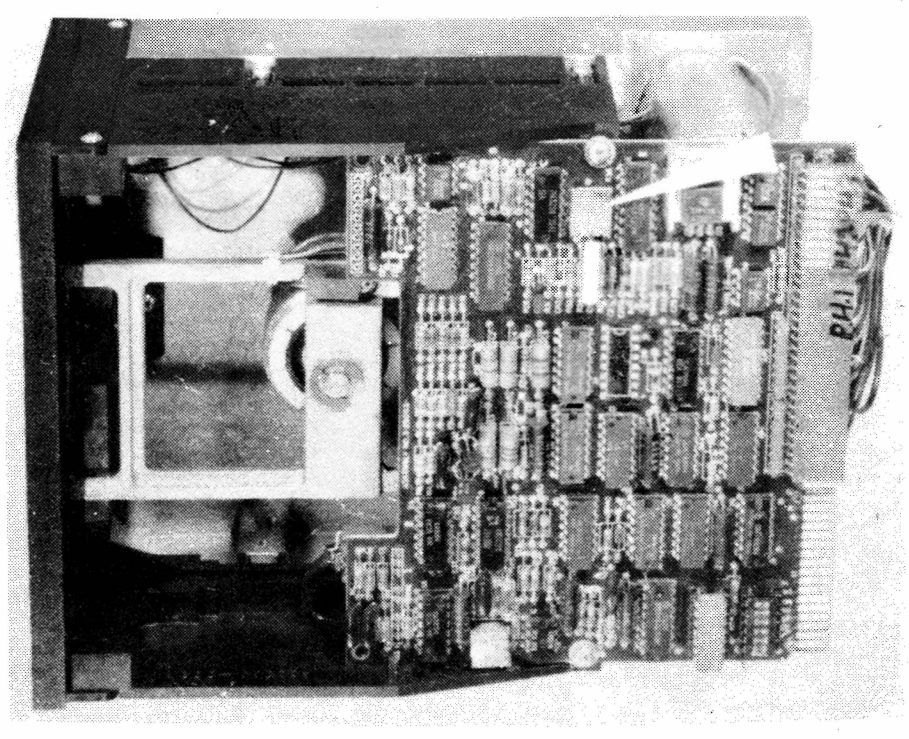
Three trace cuts, above, and one jumper, below, are all that are necessary to adapt MPI drive for TRS-80.



Power supply is just $39.95 from Matchless.







Two views of $265 Matchless MPI drive.

FFC boards remap, expand memory to 64K

Field Engineering Consultants, Ltd. of Woburn MA has announced memory remap and expansion circuits that allow use of standard CP-M and full 64K RAM.

The remap board and software (operating system) converts, on software command, the TRS-80 to a "real" CP-M type machine that loads programs at hex 100 and responds to Call 5.

The disk format is changed to a 128-byte, 18-sector or IBM format, instead of the 256-byte sector format of TRS-80. When switched to the Freedom Mode (FEC's version of CP-M), the TRS-80 really changes its personality.

Freedom Mode supports both serial (through RS232 board) and parallel printers. The operating system supports both 35 and 40 track disk drives. Freedom Mode permits compatability with other Z-80-based machines, such as the Cromemco Model 2, that use 5¼" disks. The change is pluggable and totally transparent while running in TRS-80 mode.

EXPANSION MEMORY—This board is used in the Freedom Mode, when the ROM is no longer used and there is room to address 14848 additional bytes. The expansion memory fills this gap and increases addressable RAM to a full 64K bytes. This memory is structured to account for the memory-mapped areas of the keyboard, screen and I-O functions.

In that first 16K of TRS-80 memory, taken up with ROM and memory-mapped functions, there are blank positions. These are from 3000 to 36FF and 3900 to 3BFF hex. This is wasted address space, when in TRS-80 mode. The **extended memory** option puts these address locations into operation, thus adding 2560 bytes of RAM under normal TRS-80 operation.

The real advantage of **extended memory** is in **freedom mode.** Then the machine is functioning on RAM only. The ROM which normally is addressable at 0000-DFFFh is inoperative. The **expansion memory** takes the place of the ROM and provides the capability of addressing RAM throughout the addressing range of 0000-FFFFh, except for the **reserved** (memory-mapped) areas. Memory mapping reserves F700-F8FF and FC00-FFFF for the I-O, keyboard and screen. The RAM is not contiguous, but the T8-OS operating system fills this area and allows for the reserved space. All of the space from 0080 to D5FFh is available for user programs.

The extended memory adds 14848 additional bytes of addressable RAM to the system, that makes the total RAM 64K (exclusive of the screen and memory-mapped areas).

The T8-OS loads into the highest RAM in the system automatically to provide the maximum user area. Even when the **freedom board** is not active, the **extended memory** adds 2560 bytes to the normal TRS-80 memory size The added memory is just above the ROM (3000-36FFh and 3900-3BFFh). This is an ideal place for programs such as RS232, debounce, lowercase, etc. These locations are unusable in the stock TRS-80, thus use of them totally avoids conflict with stock programs.

FREEDOM OPTION—Consists of circuit board, system disk and instructions. The board rotates memory to a more conventional configuration, with all RAM at the bottom and the special memory-mapped areas &disk, printer, keyboard and video) on top.

The disk has the T8-operating system and software to utilize the switch. It provides the necessary drivers to operate the system compatable with most CP-M-based software using the Call 5 convention.

Freedom change provides the ability to use either the TRS-80 256-byte disk sector standard or the widely-used IBM 128-byte convention. This opens the door for the TRS-80 owner to use software originally written for 8080 or other Z-80 systems.

FEC has run software from a number of sources on TRS-80 in **freedom mode** under T8-OS. These include: the Cromemco editor and assembler, Fortran, extended Basic and Cobol. The Cromemco Cobol requires **extended memory** to do any practical compilation, because it uses so much RAM. Also the REVAS Z80-8080 disassembler for CPM (distributed by A E Hawley, Los Angeles), the CP-M 1.4 editor and assembler, and the CP-M-compatable utilities distributed by InfoSoft. These all function as though written for TRS-80.

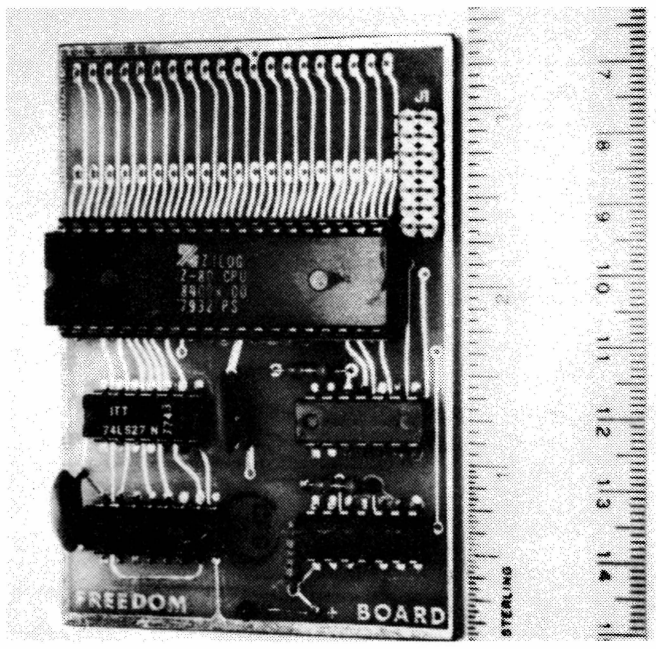
Use of **freedom change** allows use of the large amount of very powerful low-cost software on the market that will not run on a normal TRS-80. Installation of **freedom change** makes most of the CP-M type usable immediately.

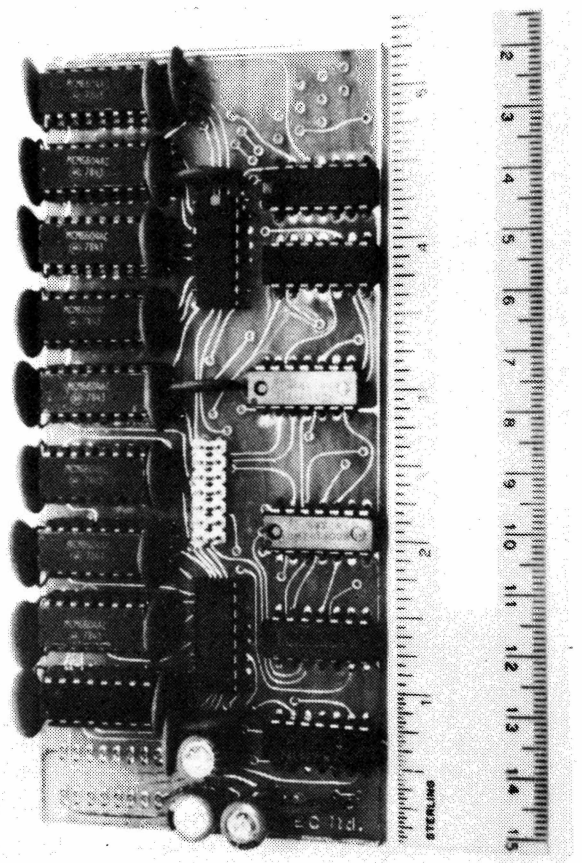
SYSTEM REQUIREMENTS—Both changes presume a 48K disk system. The **freedom board** is completely pluggable, and may be installed by a novice. The **extended memory** installation involves removal and replacement of an IC, cutting two circuit-board runs, and soldering several connections. Although these are minor changes, it is recommended that **extended memory** be installed by someone having skills in removal and replacement of printed-circuit-board components.

Freedom option is priced at $245 and **extended memory** is an additional $295. **Freedom board** may be installed alone, but **extended memory** must be installed with **freedom board,** since with **extended memory** the two boards interconnect. **Extended memory** can be added any time after **freedom board.** The hardware has a 6-mo. warranty.

Factory installation of **extended memory** is $50, with 10-day turnaround; call for reservations.

Field Engineering Consultants is at box 2368, Woburn MA 01888 (617) 944-5329—Master Charge and VISA orders accepted.





Freedom Board, above, and Expansion Memory, upper right.

$69 'Poor Man's Floppy', at right.

PRODUCT REVIEW:

'Poor Man's Floopy' so good it makes its $69 price a joke

Gerald Abear mentioned in a recent letter a "poor man's floppy" that sells for $69.95, plugs into the keyboard expansion port and "supposedly saves 50K bytes on a C-10 cassette and loads five times faster than TRS-80, with no critical volume level problems".

Gerald is the **great** guy who brings us that **fantastic** Basic 3 for just $24.95, and other neat software systems, and now has a lowercase instruction and software outfit for just $8.95 postpaid. (Personal Computer Service, 1515 Andrews dr., Las Cruces NM 88001)

What good is a $69.95 "floppy"?

Not expecting much, we called the manufacturer, JPC Products Co. of Albuquerque. We were right on deadline and there was no time to waste. Flaky product or boon to the TRS-80 user, there was a story here, so we were as persuasive as possible and got a unit air freighted here next day.

The trip to San Diego's beautiful Lindberg Field (It may be too close to the city to be safe, but, right on the bay, it is beautiful) was a rare delight to an editor who seldom gets farther from the office than just the UPS barn.

Back at the office, I plugged the thing in to a 16K computer and loaded their system software, at standard 500K baud, using our Acu-Data. That is a neat shortcut. Meant we got a load, first try. That was easy!

Now I had in my machine a "poor man's disk operating system" PMDOS? It gives you the following commands: SAVE, LOAD, LOAD? (verify), LOADN(directory command, prints list of programs on a cassette. Also a tape positioning command that positions tape to just after the named file), KILL (deletes part of operating system to create room for large Basic program), RSET ("recorder setup" command turns on tape deck so you can rewind or fast forward), RUN (loads and runs named program), PUT (binary or system equivalent of the SAVE command), GET (binary equivalent of LOAD?), GET? (binary verify command), GETN (binary directory command).

The part of the operating system deleted by the KILL command is called UTIL. The remainder is called BOOT. UTIL statements include OPEN (required in Basic programs before cassette input or output of data), CLOSE (required in Basic to end a data file), PRINT (similar to TRSDOS PRINT # statement, allows numerical and string data to be written into a cassette file), INPUT # (allows numerical and string data to be read from a cassette file).

We loaded the sample programs provided. They all came in first time, every time. And fast! It was a lot like disk. We CLOADed a People's Software program into the system and saved it onto another tape. Then we took the SAVED tape over to the tape duplicator, and ran a second generation copy. That copy could not be expected to run (load), because:

1) It was recorded at 3600 baud, and even 500 baud tapes are difficult (extremely!) to duplicate,

2) Nowhere in the chain did we take time to set levels. For duplication, an absolute must, here in this office, is to set recording level to optimin, and not use automatic level set on the recorder. We used automatic level set, did not check duplicator volume, did not set volume on replay.

3) We used junky tape that we've had and refused to use since we switched to Agfa 611 at least a year ago. We've had employes lose entire day's work, plus travel expenses, due to those tapes!

The duplicate tape loaded reliably, time after time. Rock solid.

We cannot recommend this system highly enough. You get that funny little box, power supply and ribbon cable. A dynamite manual and tape containing the operating system. Suddenly, your CTR-80 thinks it is a disk! Programs load almost that fast, and **at least** that reliably! Only $69.95 in kit form, or $99.95 assembled, tested and guaranteed. Add $3.50 postage and handling either version, and New Mexico residents add tax.

JPC Products says anyone that can spread toothpaste on a toothbrush can solder, and offers a "can't fail" guarantee: "If you build the TC-8 and for **any reason** it doesn't work, we will make it work at **no cost.** All you have to pay is the shipping." There are only about 30 parts on the board.

(Continued on page 11)

Mapper lets mod 1 use 8" drives, regular CP-M

By JOHN R. MARLER
6351 Almaden rd, San Jose CA 95120

The time for all TRS-80 Model 1 owners (who feel rejected from Radio Shack's apparent overlooking our needs for more disk storage) to step up to 8" disk drives has finally arrived. **Omikron Systems** of Berkeley CA has developed a modification for the Model 1 that allows the use of full-sized drives and **normal** CP-M.

Nearly everyone knows about CP-M and that it is an almost-universal operating system for Z80 and 8080 microprocessor systems. One of the **key** features of CP-M is the fact that it is "debugged" and is operating in many types and brands of computers throughout the world and the amount of software written for CP-M system operation would probably fill many catalogs.

CP-M version 2.0 is fast becoming the most acceptable operating system for the Model II. Two newsletters to which I subscribe have published the rumor that Radio Shack will soon announce CP-M instead of TRSDOS for the Model 2. Use of CP-M really protects software investment, because of the upward-compatibility of CP-M and other computers.

MAPPER 1 & 2—I purchased a system containing Mapper I and Mapper II with two 8" Shugart disk drives, which gives my Model 1 the same storage as the model 2, with more than a 1,000 savings! I have also found that double-disk operation is the only way to go, since it permits backup and other utility without the hassle of inserting and removing disks. I received delivery after only three weeks, although the company quotes six.

The hardware called Mapper I allows you to utilize your Random Access Memory as **low memory** and not have it located above ROM where machine activity begins under normal TRS-80 operation. This permits use of "normal" CP-M and for the first time makes the world of "real" CP-M available to us Model 1 users.

Mapper II is a board that, coupled with the disk controller chip, allows use of 8" drives and provides true data separation, which eliminates the cause of most disk I-O errors—that of poorly-separated data reading from the minidisk.

Mapper I also provides a way of using the graphics capability of the TRS-80 and adds cursor control in a manner similar to the Soroc 120 CRT terminal, which uses a very simple and easy-to-understand X-Y addressing of the cursor.

INSTALLING THE SYSTEM is simplicity itself. The Omikron instructions are so easy to read and understand that I was able to install the system without one problem.

The first step is to open the keyboard casing and locate the Z-80 chip. The instructions carefully outline which chip that is. Pull it out of its socket insert Mapper I into the now-vacent socket and put the Z-80 into a similar socket on Mapper I.

Two pinch connectors then "wire" the board with surprisingly-strong connection. This eliminates the need to solder anything! Pull a connector wire through the opening where the interface cable goes, close the keyboard case and Mapper I is ready.

Mapper II is like Mapper I, but is installed in the expansion interface. To install Mapper II, open the expansion interface and locate the disk controller chip. Omikron instructions clearly show how to do this. Remove the disk controller and replace with Mapper II. Put in two "pinch" connections. Pull the other end of the connector through the opening where the interface cable goes; close the expansion interface; fasten the Mapper connection wire, hook up the interface cable and you are ready to plug in disk drives.

In the system I purchased, I asked Omikron to designate drives A and B as minidrives, and C and D the 8"ers. Omikron will sell the disk drives, or sell the boards alone.

RUN IT—The CP-M disk is delivered configured to your specified memory size and drive configuration. Additional memory size or drive configurations are optionally priced at $10, and CP-M version 2.0 is available for an additional $100. Minidrives are not required—a system of four 8" drives can be ordered.

UTILITIES—Omikron has provided several, which are:

SETUP.COM—Allows customer selection of (a) deleted character echoed to screen, (b) automatic line feed with carriage return, (c) lowercase screen printing, (d) and graphics characters; (e) software-driven forms feed.
LFORMAT.COM—Formats the 8" disk.
MFORMAT.COM—Formats minidisk.
SERIAL.COM—Reads switch setting of optional Radio Shack RS232 board, displays it and sets all printer output to the RS232 board and serial printers. It also allows for two types of printers: one fast copy and other letter quality.
XDIR.COM—From CP-M user's library, but modified to read both size disks and show number of sectors allocated to each program, and number of extensions a large file has in the directory. Displays directory in three columns.

MEMTEST—The most extensive memory test I have ever used. Minimum time (unless defective chip detected) is 15 minutes for each bank of 16K. MEMTEST will not only write patterns to the memory chips but will check the relationship and interaction of the memory locations next to each location. This program has found a chip that all other memory tests overlooked, but which ruined data and programs until found.

OTHER GUY—It should be mentioned that there is another Berkeley firm that claims to have a similar system, but to my knowledge that other firm is not delivering.

MY SYSTEM—I now have my Omikron-modified TRS-80 Model 1 working. Due to the Mapper II data separator, used even in TRSDOS, I have eliminated my disk I-O errors and have found the disk work much easier and smoother. I have modified Wordstar from Micropro to operate on this system and have just completed modification to Wordmaster to work as a replacement for CP-M's ED.COM to allow full screen editing without the printing features of Wordstar.

I have been able to run the Osborne accounting systems in addition to other software programs that I have written. Conversion was not difficult. My company has its own business accounting software and, due to this modification, have markets in **three** areas: (1) TRS-80, (2) CP-M "normal" for both 8" and 5¼" drives and, (3) TRS-80-compatible CP-M, such as sold by Lifeboat Associates. This CP'-M is **not** compatible with "normal" CP-M programs and can only work with machine-language programs modified to work above the Radio Shack ROM. I have found that the disks formatted by my system are readable by a Model II using CP-M 2.0. An additional utility program now released by Omikron is a program to take any program off a TRSDOS-formatted disk and transfer it to a CP-M disk.

I know of no other computer on the market that provides the capability of using three different operating systems or using both mini and 8" drives AT THE SAME TIME!

TAPE 1 LEVEL 2

Mortgage calculations, Dow Jones Industrial, cash flow, inventory-change, California income tax, journal ledger (8K), loan amortization, perpetual calendar, bio rhythm, payroll, diet planning, speed reading, touch typing, sales receipt tally, decision maker, mail addressing, straight depreciation, double-declining depreciation, and revolving charge account.

Also, math problems, queen, Star Trek I, number guessing, wheel of fortune, World War II bomber, rock-scissors-paper, seek, Star Trek II, Red Baron, mini-Trek, strategy, pilot, battleship, "On A Snowy Evening", mastermind, tic-tac-toe, grand prix auto race, capitals, etch sketch, hangman.

TAPE 2 LEVEL 2

Fully documented in **Some Common Basic Programs** by Lon Poole & Mary Borchers (Osborne & Associates, 630 Bancroft way, Berkeley CA 94710—or from CIE—$12.50 postpaid from CIE, via UPS, CA residents add tax (to $13.25)):

Investment, future value regular deposits; regular withdrawals, initial minimum (for withdrawals); nominal interest, effective & earned-interest; depreciation rate, amount depreciation; salvage value; discount commercial paper; loan principal, regular and last payment, remaining balance, term-loan; mortgage amortization; greatest common denom.- integer prime factors; polygon area; triangle parts; analysis, operations two vectors; radian-degree., degree-radian conversion; coordinate, polar equation, functions plot; linear, curvilinear interpolation; Simpson's & trapezoidal rules, Gaussian quadrature integration; derivative.

Side 2—quadratic equation, polynomial (Newton) & half-interval-search roots; trig polynomial; simultaneous equations; linear programming; matrix addition, subtraction; scalar multiplication, inversion; permutations & combinations; Mann-Whitney U test; mean, variance, standard deviation; geometric mean & deviation; binomial, Poisson, normal, Chi-square distribution; Chi-sq., student's T-distribution test; F-distribution; linear correlation coefficient; linear, multiple-linear, Nth order, geometric, exponental regression; system reliability; future projections; Federal withholding taxes; tax depreciation schedule; check writer; recipe cost; map check; day of week; days between two dates; anglo to metric; alphabetize.

TAPE 4 LEVEL 1

Election returns, business percentage, ups and downs of business, index, inventory control, sales receipt tally, gas mileage, driving distance, mixed monthly sales report, payroll, annual earnings, speech recording aid, and double-declining depreciation.

Also, math problems, cash register, chase, snoopy, commander-in-chief, Christmas graphic, air raid, balance scale, stock market, tic-tac-toe and **On A Snowy Evening.**

TAPE 5 LEVEL 2

Memory test, mortgage payments, tension breaker, lineprinter-screen & vice-versa utilities, Federal income tax, election returns, business percentage, vacation planner, car pool(disk), diet planning 2, mailing list(disk) and first aid.

Also spelling bee, Star Trek 3, mind bender, tachistoscope, chase, common factor, klingon capture, spelling practice, Hamurabi, animals, Snoopy, cryptogram, starship, ants, **Yesterday**, and Pilot(disk). Pilot is the language of computer-aided instruction (CAI)..

TAPE 7 LEVEL 2

Disassembler, Pilot, roster, dropout, memory loader, memory sort, inventory control, graph, land surveying, mixed monthly sales report, shopping list, diet planning 3, loan progress chart, hex-decimal conversion.

Also Star Trek 4, states and capitals, battleships 2, spelling practice 2, number guessing, hangman 2, snark, slot machine, cipher, target, surround, adder, termites, lunar lander, multiplication exercise, five-in-a-row, Bastem, and write. A number after a program indicates there are other similar People's Software programs. Pilot is the same as the disk pilot on tape 5, except runs on 16K tape systems.

Announcing-

Have you heard about the $15 Pascal compiler?

TINY PASCAL FOR TRS-80

At last your TRS-80 can run Pascal too! The Chung/Yuen "tiny" Pascal is fully implemented for Level II TRS-80, 16 K and up. You no longer need be left out of the growing group of Pascal users, because People's Pascal gives you everything you need to write structured Pascal programs:

- tiny Pascal compiler
- complete text editor for writing your programs
- complete tiny Pascal monitor
- sample Pascal programs
- user's manual

(TRS-80 Computing issue 1:4)

People's Pascal is both a powerful, structured language and "CPU expeditor". People's Pascal programs execute at least four times faster than Basic, and often eight-times faster! Special functions open up the complete graphic capability of TRS-80. You now have the means to write those dazzling, impressive, high-speed graphics programs that are great for games, plotting, statistics, etc.

For the serious computerist, side two of People's Pascal II (tape 6) contains a larger compiler and complete source to the compiler, written in Pascal! This means you can re-compile the compiler, making changes, adding features, etc. (but this will take at least 36 K RAM and a solid knowledge of programming).

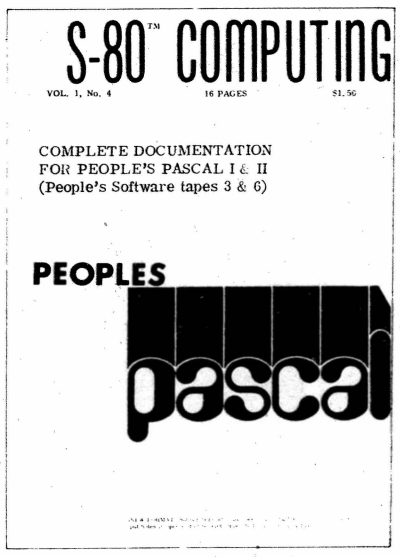
With the complete People's Pascal operating system, you can save and load both source (Pascal) programs, and compiled programs, to or from cassette tape. This means that once you have de-bugged a program, you can save the P-code (compiled program) and thereafter, to run the program, you need only load the super-fast P-code.

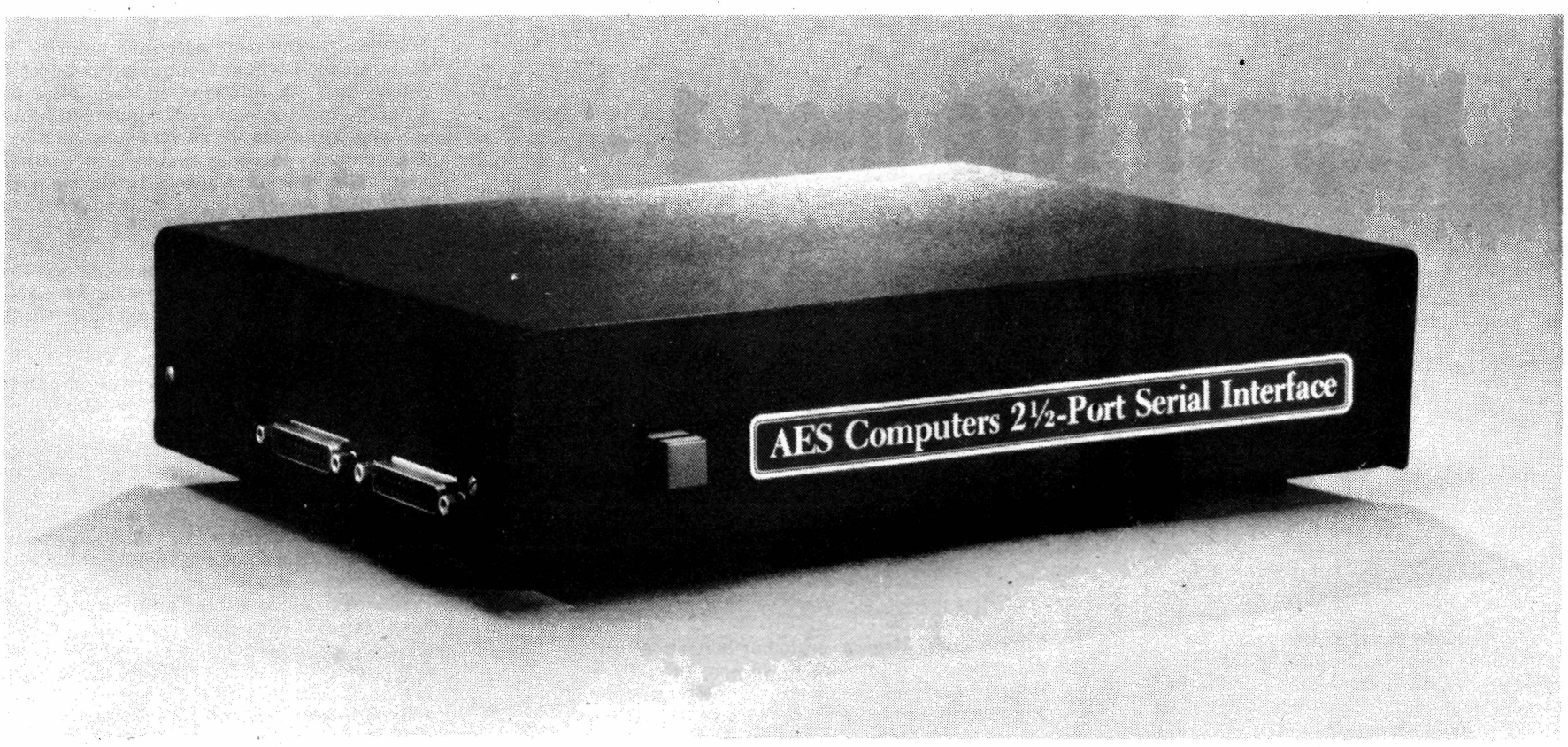
Here is a partial list of People's Pascal features: recursive procedure/functions • for (loop) • case if/then/else • one-dimensional arrays • write • read constant • repeat/until (loop) • "peek & poke" • plot (graphics for TRS-80)

—DEALER INQUIRIES INVITED—

People's Pascal I (tape 3), is written in Basic, implemented for TRS-80 by John Alexander of Berwick Australia. It compiles more slowly and is harder to use, but includes instructions for converting to disk. People's Pascal II (tape 6) is entirely by the Chung/Yuen team and is a further development stage of their concept. Full documentation included.

Other People's Software tapes $7.50





INTERCONNECT POWER—AES Computers' 2½-port serial interface allows parrallel-processing of two TRS-80s. It is priced at $395.95. A 4½-port version is $695.95. The device allows connection together of any number of TRS-80s. AES Computers is at 118 S. Loara, Anaheim CA 92802.

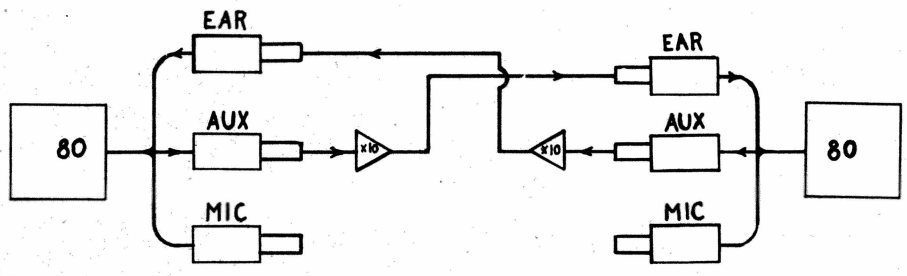
Interconnect

Model 1s via the cassette ports

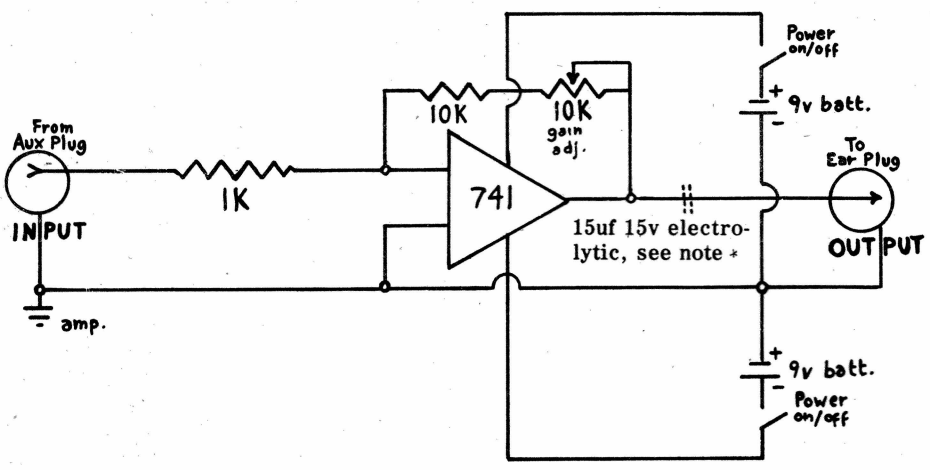
By Prof. JOHN V. KANE AB3C
Wynnewood House apt 402, Wynnewood PA 19096 (215) 642-2856

In connecting a pair of TRS-80s together the cassette I-O can be used but a gain-of-10 amplifier is needed between the cassette output of one and the cassette input of another.

That is because a TRS-80 only outputs 0.3 volts of signal but it needs 3 volts of drive. A typical connection is:



A typical booster amp for this purpose is the following:



When CLOAD"A" is executed on the receiving computer it waits for data from the sending computer. To send data it is only necessary to execute CSAVE"A" at the sending computer and the program in the sending computer will go to the receiver!

This same operation can be done over a phone line. Heathkit HD-15 phone patch is a rather nice convenience to use for this, and, an oscilloscope is a handy thing to use to check that the received data has not too little and not too much amplitude for the TRS-80 input.

Further, data transfer from one computer to another can be done using PRINT(number sign) to send and OUTPUT(number sign) to receive.

All of this is low cost and has many uses.

* Charles Riggs of Long Beach adds a 15uf 15V electrolytic capacitor in the output. Without that, only half the square wave is reproduced.

HEATH HD-15 PHONE PATCH—Makes it easier to connect TRS-80s over phone lines.

AMPLIFIER ALSO WORKS—This little speaker-amplifier from Radio Shack also can be used to interconnect TRS-80s via their cassette ports. CIE used it in its networking box. The device is $10.95, and stock number 277-1008.

NOBODY CAN BEAT OUR PRICE!

MS-204 PRINTER
INTRODUCTORY PRICE:
$749
CABLE: $34.50

Compatible with TRS-80, Apple, Pet or any other Centronics-type system

Features
- Documentation Included
- 80 Columns, 63 lines Per Minute, Bi-Directional, Nominal Thruput
- High Reliability - Heavy Duty Cycle, 100 Million Characters Print Head Life
- Adjustable Sprocket Feed (4½" to 9½")
- 5 x 7 Dot Matrix Character Font

Ask about our MS-80 Drives & Software

MATCHLESS SYSTEMS
18444 S. Broadway
Gardena, CA 90248
(213) 327-1010
© 1979 Matchless Systems & MarketPlan

Serially Interfaced Computer Networks

By Donald McWilliams, Ph.D.
AES Computers
118 S. Loara
Anaheim, CA 92802

Within the last several years a great change has occured in computer systems. As the volume of microprocessors and microprocessor-related equipment has grown, the cost has decreased. These circumstances have reduced the retail cost to less than $2000 per system and increased the numbers of computer users by a factor of 10. This proliferation of computers and users leads to thoughts of enormous networks of interactive computers in a common data base and with universal access. It is both feasible and possible for computer networks to operate to the benefit of the users and the TRS-80 and Apple are making the day of implementation imminent.

All the elements of the hardware are already on the market. The major challenge now is duo-fold. 1. The software requirements and theories are only superficially defined. 2. The logical environment and the physical potential of interactive systems are virtually unexplored. It is a solution looking for a problem. I will describe the solution and leave the discovery of the problems it fits to you. Now on to the solution.

Even in large systems the interactive computer system has been of very limited application. The great size and speed of government systems has lead to configurations of monolithic character with huge memory capacity. The most common utilization of multiple computers is that of one computer converting the language and another performing the calculations. Even this simple operation uses an intermediate tape as a storage device to separate the two computers so they can each work at their best pace and mode. Recent use of microprocessors to convert keyboard formats to computer formats is another application of the same type.

Huge military systems have been devoted to strategic games and, in some cases, several computers have been used by each side of the battle with smaller units devoted to the contributions of lesser nations in the global conflict. In this case, the interactive nature of the processors is clearly defined by the software which is strictly an attempt to simulate the human war ecology. This is conducted with a large human component in which intermediate steps are controlled largely through the observation of large video presentations combined with political inputs based solely on human intellect.

The intricacies of each country involved are usually built into each single computer when separate machines are used and these specific characteristics are not modifiable during active operation except to halt the on-line operation to modify the program. Only predictable computer reactions are utilized and those only in a very simple interaction basis.

The electrical power utilities have an interactive computer network which is primarily an A to D and D to A controller and alarm system. This is, truly, a large system both in cost and size as it reaches throughout the continent to control a very important part of every person's life.

The Bell system uses a very large interactive computer network to control traffic and allocate channels.

The railroads have simpler but effective interactive computers to control their traffic. A more important computer system is used to control the ebb and flow of railroad cars and railroad freight. This is really a large problem with lots of individual companies involved.

Certainly chemical plants, refineries, nuclear plants, and others use interactive systems. Also, large credit companies should be included.

In each of these cases, the systems are custom constructed to fit the user and the object is profit, service, or reduction of risk. They all cost $10,000,000 to $100,000,000 to install.

Now consider a small computer system: Two microcomputers utilizing two high speed serial interfaces in synchronous mode at 64,000 Baud. This capability is obtained with the 2½ Port Serial Interface made by AES Computers. Why should we need high speed serial mode? It is the only way to go fast at low cost. Parallel buses require a lot of hardware and to go any great distance at high speed is expensive. With the synchronous serial capabilities of the 8251A controllers in AES Computers' Interface, communication over a few hundred feet requires only a modest investment in hardware with less than 100 bytes of code for a complete full-duplex operation and error checking.

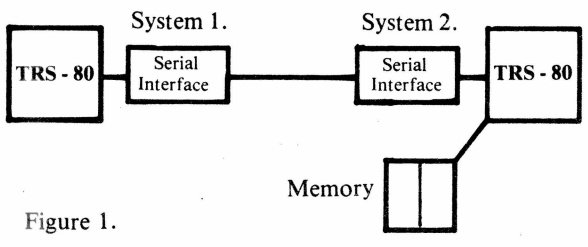


Figure 1.

1. The two memories (either disk or cassette) are accessible to both operations.

2. Disk 1 contains the game environment operating system which controls both Participants equally by defining the game rules.

3. The disk system for an advanced game would include each Participant's strategy disks, containing modifications where the individual Participant might include modifying the software which he devises from experience with the basic game.

4. Participant's Strategy disk or tape is not known to the opponent.

5. Each Participant is allowed unrestricted access to the common data of the Opponent's system through the serial interface software.

6. Defenses may be devised which detect probing and certain subterfuges may be employed.

7. The game may use cursors, joy sticks, sounds, bells, and graphics to implement the environment which each Participant observes.

8. Timed events can also be part of the system as well as scoring.

9. In a slower operation this configuration could be operated in asynchronous mode with modems.

A small business application could use the same configuration and operate System 1 as a minimum cost unit sharing the resources of the larger System 2.

Another application of this two-computer system would be an error elimination system with both computers running the same program at the same time in synchronous mode with the results compared at any point in the processing or at all points of the processing. The computers would detect a difference at predetermined parts of the program and either stop or ring a bell and continue on one of the systems. This is a very simple error detection system which would greatly increase the confidence in results of long programs. Consider that the entire two-computer system with four disks, a printer and ample RAM can be purchased for less than $5,000 including the serial interfaces!

Figure 2 is the next step.

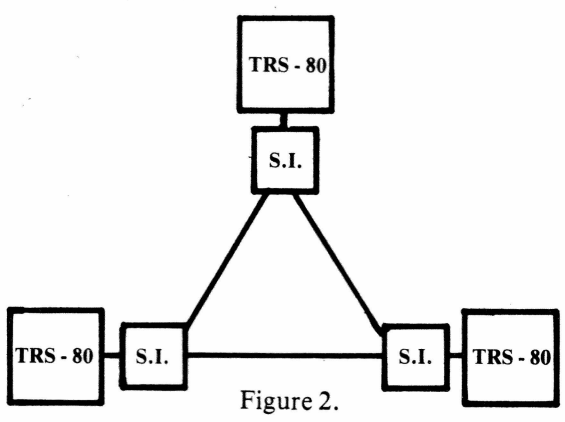


Figure 2.

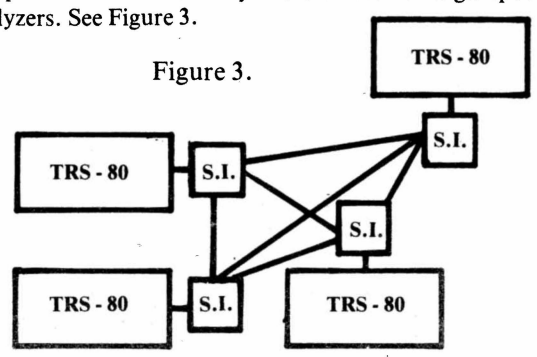
Because the AES Computers Serial Interface has 2 independent synchronous channels, this configuration is very easy to implement. It would take all of 10 minutes to connect the hardware. The serial interface has its own power supply and simple interfacing.

First consider the esoterics of the symmetry. Each operator looks into the same system configuration. This means that with a disk at each station that all positions can have the same program for start-up. The games philosophy for this system is much the same as for the two-computer system, but the business applications finds a substantial increase in processing power.

This 3 computer system can be used as an ultra-reliable majority logic computer that accepts an answer or intermediate result if the same result is presented by any two of the computers. It also corrects the errant computer at that step and continues with all three. This kind of computer is the kind used in "severe" environments and can be used for critical calculations as well as for the voting comparison of three external detectors during operation. This is very effective in threshold measurements and data/noise isolation. In addition, this can be a very powerful differential digital processor.

Extending this consideration to 4 computers in which #4 is used to control the detection modes of the other 3 computers leads to very cost-effective high-speed analyzers. See Figure 3.

Figure 3.



A straightforward way to describe serially connected interactive computer systems of n elements (computers) is to draw polygon of n sides and connect all the corners. Figure 4 is a system of n = 6 computers.

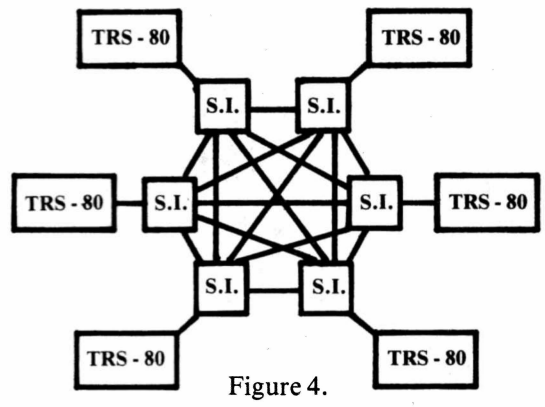


Figure 4.

There are 5 serial connections for each computer and all the observations previously made continue here. Particularly the symmetry. We also observe that high-speed, full-duplex handshaking serial communication will be required for any difficult or large problem.

Imagine the kind of game 6 members of a club could devise on this system. The opportunities for secret arrangements between participants become very prominent and the system begins to look a lot like a political system. It is entirely possible that each participant would be in a separate room using only his own screen as his source of information.

It might also be possible to incorporate verbal communication in the game as well as employing a referee. It could be used as a gambling device.

There are more compelling applications for this sort of system. Suppose that some complex interactive human system needed modeling in order to sort out the directions for further research.

Such a human system is the human endocrine system. These internal glands of the body are the controllers of metabolism and are very complex in their interactions. Suppose each computer of the hexagon was assigned the function of one of the endocrine glands. Interactions as defined in experiment would define that computer's program. The video would display the active function of the gland and changes made would be immediately apparent.

Some other features of this interactive system are:

1. The system can be increased to n + 1 or decreased to n − 1 without completely upsetting the software of the other nodes (node is another name for computer).

2. Your system is comprised of a lot of very simple elements. The computer with memory and the serial interfaces are the same throughout the system. The system can be repaired in minutes and only a few spare parts need be stocked. Indeed, you could limp along without spares if you go to n − 1.

An additional dimension can be added because a controlling computer can be added at each node of the hexagon. Figure 5 is a simple diagram of two hexagons in nodal communication. This double ring structure could well be two clubs in interactive communication.

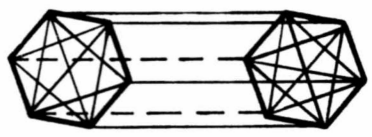


Figure 5.

A two-dimensional lattice can be arranged of several hexagons as shown in Figure 6. Needless to say, the programming would be challenging.

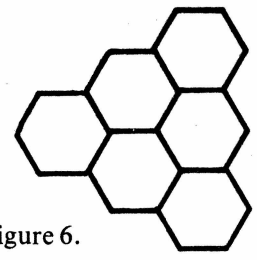


Figure 6.

As the technology grows and the number of participants increase, interactive computers are going to contribute. The contributions that are easiest to predict are in modeling systems and mathematical theory.

Microprocessor clubs have the greatest likelihood of implementing the software required for an early use of a large interactive system. Perhaps the clubs can pool their resources and commit themselves to leading the technology of these areas. It won't be easy, but it will be fun.

Connect TRS-80s via cassette ports

By GREG W. RAWSON
Staff Technical Assistant

Particularly in education, there are applications for TRS-80 which would be greatly enhanced if the computers are tied together so that any or all can be fed the same program or data.

in a classroom situation, the instructor could load his computer either from disk or TC-8 "poor man's floppy", then send it to student machines, either at the Level 2 500-baud standard or the ABS Suppliers' or Palomar Software speedups. If using the latter speedups, Palomar or ABS system software would be included in the student's initial load.

With a single load, all students could be up and running with their project for the day. Or, if all students were working on different things, all could be given the cassette speedup load, then their individual programs, one after the other.

Loading from machine to machine is simple and dependable. All that is needed is a 10X voltage gain, as Prof John Kane explains in his Interconnect article.

While even 500 Baud is unreliable on stock TRS-80, when going to and from tape, very high speed should be reliable when simply going from port to port, without introducing the distortions magnetic tape does.

Thus the instructor can quickly load all student machines, and not have any tape recorders present at the student stations. That should increase reliability and productivity by as much as a factor of ten.

And all the students can back-load their programs to the instructor's computer, where it can be put on floppy—either disk or "poor man's".

Or students can print listings or runs of their programs on the line printer. One student can be receiving a load from the instructor, while another is using the line printer.

LINEPRINTER CHOICE—It was assumed the classroom computer laboratory would have to be brought in on a rather tight budget, so we did some pretty careful shopping. The printer should be very fast, and reliable, as well as inexpensive. That is quite a combination!

One printer won hands down: Comprint 912. It supplies highly-readable 9x12 dot matrix characters, 80-character lines, 170 line-per-minute throughput and has a minimum of moving parts. Not even a ribbon! The price? Just $699 in serial configuration, which allows any student to print, simply by having the instructor flip one switch, and then typing LPRINT (or RUN).

That is made possible by an ingenious product, the Small System Hardware TRS-232 serial interface, which connects to the cassette port. Because of our interconnect scheme, only one of the inexpensive, $49.95 devices is needed. Were we to use the less-expensive ($660) parallel-interfaced Comprint 912, only one computer would be served by printer. By using the TRS-232 interface and our simple interconnect, any computer in the classroom (assuming they are all connected), including the nstructor's, has access to the printer.

Of course we get that versatility at a slight cost. At class beginning, each student machine must load the little TRS-232 driver program. They can all be loaded at the same time, and that one load should last until powerdown—possibly all day. The other cost, besides about 500 bytes of memory in each computer (the driver program), is speed. Program listings print out at about full speed, 170 lines-per-minute. Quite fast! That is because in listings, the lines are not even half full width (80 characters). The TRS-232 interface is unidirectional. It outputs only. It can not pick up handshakes from the computer, and tell the computer to hold of while the printer is busy. Thus you have to send data to the printer at maximum-or-less speed. The printer's maximum speed is 2250 Baud, and it ois strapped to accept data only at standard baud rates such as 1200 and 2400. No 2250! Thus, although you could get 2250 baud out of the TRS-232 interface, the printer couldn't read it. And you can't go out at 2400 baud (well, maybe you could—we simply didn't) If you settle for a 1200-baud speed, most work can be printed at 170 lines-minute, and you have very few problems. We'll get further into the Comprint in a future product review.

if we used the bi-directional Electronic Systems RS-232 serial interface, we would have to provide an interface and power supply on each computer station, as well as expensive cabling. So, in this application, the Small System Hardware unit offers tremendous savings.

HERE'S HOW WE MADE THE UNIT:

Make templates for panel layout: switches and LEDs on front, jacks, recepticles, instructors' controls on back.

Tape template to chassis, punch center holes and drill holes.

Mount hardware.

Build and mount the LED power supply.

Wire for 110V AC, including line cord, power switch, and inside and outside receptacles. Do not delete the outside receptacle. TRS-80 systems have too many plug-ins, and all accessories should provide at least a 2:1 replacement for every receptacle they use.

Mount the Small System Hardware RS-232 serial interface.

Wire jacks and switches together, using two-conductor microphone cable.

Put in 1K resistors for LEDs.

Install amplifiers. Make sure the right wires go to their respective amplifiers.

Install resistance circuit. This allows the instructor to dial-in the number of machines to which he is sending signal. It provides a uniform load to the amplifiers, since it offers a resistances which vary, according to the number of computers connected. Each TRS-80 offers 150 Ohms to the network. Figure load as you would resistances in parallel.

Label all switches and jacks.

Make cables with two-conductor microphone cable.

(See drawings and photograps on next page)

PARTS LIST

QTY	ITEM	PRICE EACH	PRICE	STOCK NUMBER
6	DPDT switch, center off	2.19	13.14	275-620
6	DPDT switch	2.19	13.14	275-614
2	SPST switch	1.69	3.38	275-612
13	red LEDs	2/79¢	5.53	276-033
6	green LEDs	2/89	2/89	276-034
1	6-position rotary switch	1.19	1.19	275-1386
1	knob	2/1.19	1.19	274-1407
20	1K Ohm resistors, 1/4W	5/39	1.56	271-1321
1	fuse block	.99	.99	270-365
1	.5 A fuse	.99	.99	270-1282
7	female sterio jacks	2/1.99	7.96	274-312
7	male stereo jacks	2/1.79	14.32	274-139
7	male DIN plugs	1.49	10.43	274-003
1	TRS-232 printer interface	49.95	49.95	**
2	mini amplifiers	10.95	21.90	277-1008
5	male miniature 1/8" jacks	2/1.29	3.87	274-286
1	cabinet	10.95	10.95	270-270
1	AC adapter	9.95	9.95	270-1551
4	AC-line recepticles	4/1.29	1.29	270-642
1	AC-line grounded plug	.89	.89	
1	6.3V transformer	2.49	2.49	273-1384
1	47 ufd capacitor	.69	.69	272-1027
1	diode	2/.39	.39	276-1101
		total	$178.86	

**from Small System Hardware, Box 366, Newbury Park CA 91320

Advertise in S-80 Computing

JOIN THE 1000'S WHO ARE ENJOYING

THE 80 NOTEBOOK

THE MONTHLY JOURNAL FOR ALL TRS-80 USERS

WITH THESE EXCITING FEATURES AND DEPARTMENTS: (AND MORE!!)

* SCIENTIFIC SOFTWARE
* LETTERS TO THE EDITOR
* PRACTICAL APPLICATIONS
* ENTERTAINMENT PROGRAMS
* WORD PROCESSING SYSTEMS
* ARTIFICIAL INTELLIGENCE
* SYSTEM UTILITY SOFTWARE
* RADIO SHACK NEWS RELEASES
* DATA BASE MANAGEMENT SYSTEMS
* EDUCATIONAL AND CAI PROGRAMMING
* SIMULATIONS AND COMPUTER MODELING
* GRAPHICS AND ANIMATION TECHNIQUES
* ASSEMBLY LANGUAGE PROGRAMMING LESSONS
* LEVEL I, II AND DISK BASIC PROGRAMMING LESSONS
* OPERATING SYSTEMS, LANGUAGES AND COMPILER DESIGN
* PROGRAM LISTINGS (UP TO 24 PAGES IN EVERY ISSUE!!)
* ARTICLES DEALING WITH UNUSUAL AND INTERESTING USES FOR YOUR TRS-80
* GAMES
* PUZZLES
* CONTESTS
* GAMBLING
* NEW PRODUCTS
* TRS-80 CLUB NEWS
* PERSONAL FINANCE
* SOFTWARE EXCHANGE
* BUSINESS SOFTWARE
* SORTING TECHNIQUES
* FREE CLASSIFIED ADS
* PRODUCT EVALUATIONS
* ARTICLES ON HARDWARE

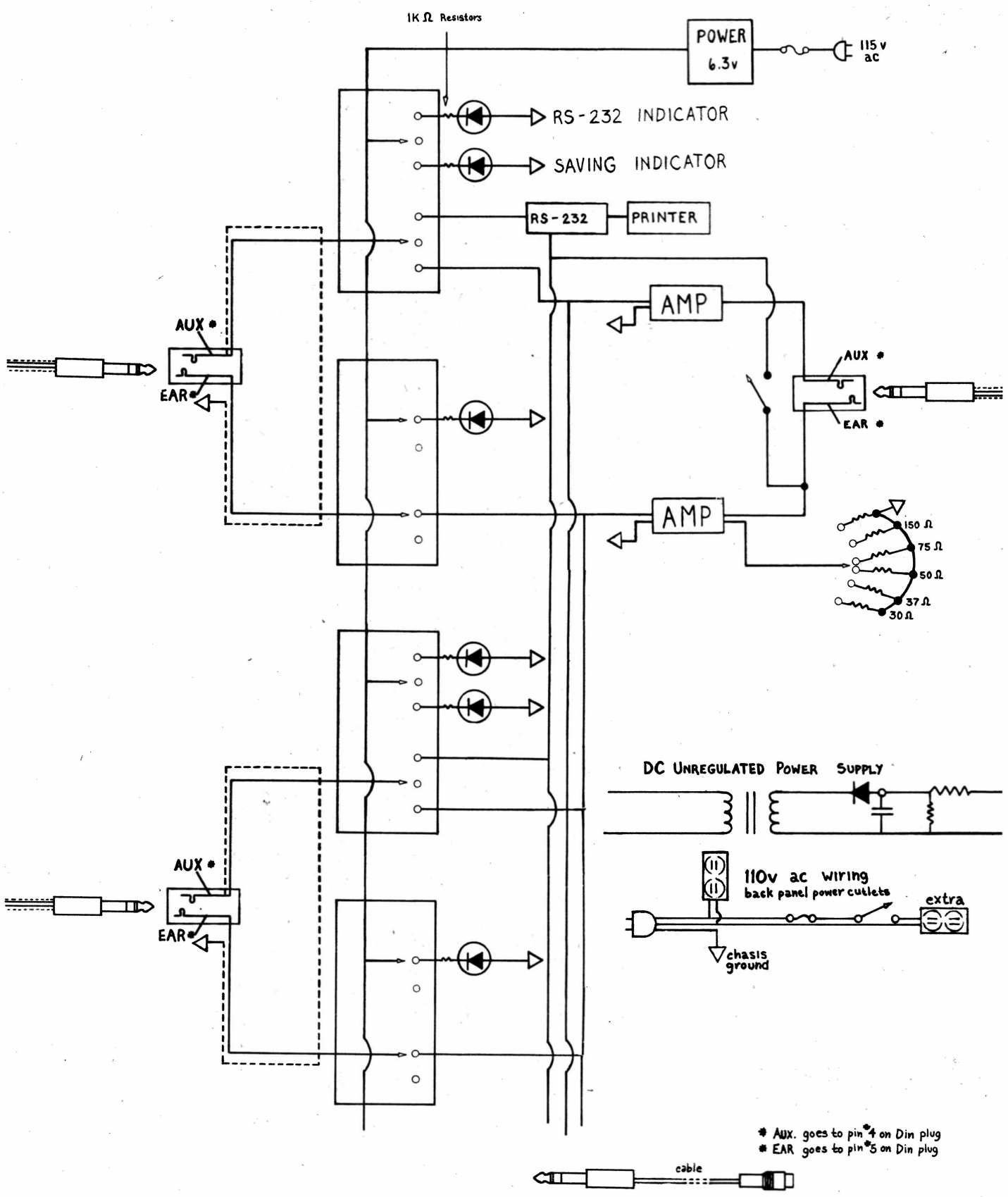
NOTE: The term "TRS-80" is a registered trademark of Radio Shack, a Division of Tandy Corporation, who is not affiliated with THE 80 NOTEBOOK in any way.

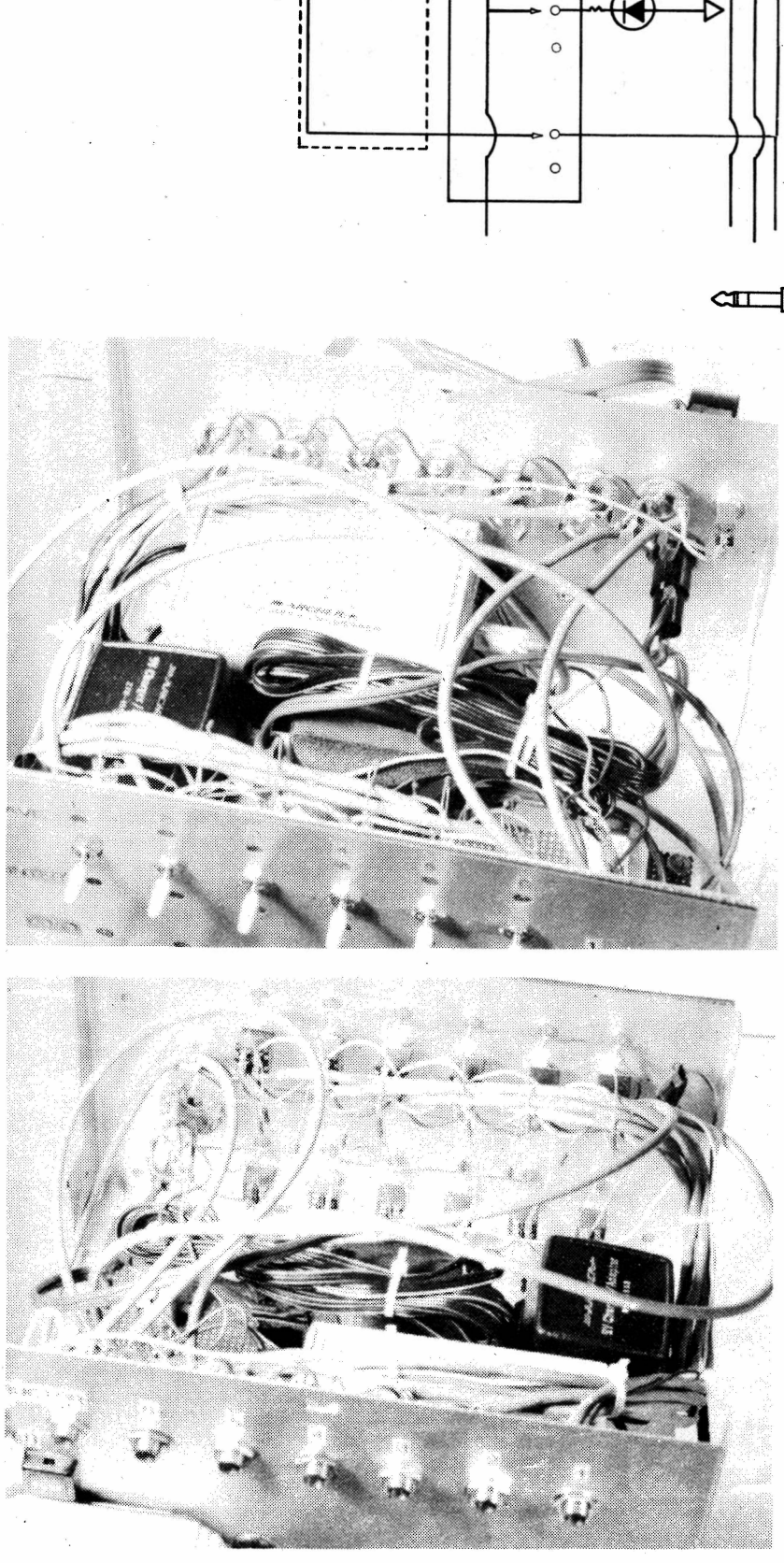
THE 80 NOTEBOOK
R. D. #3
Nazareth, PA 18064
Phone: 215-759-6873

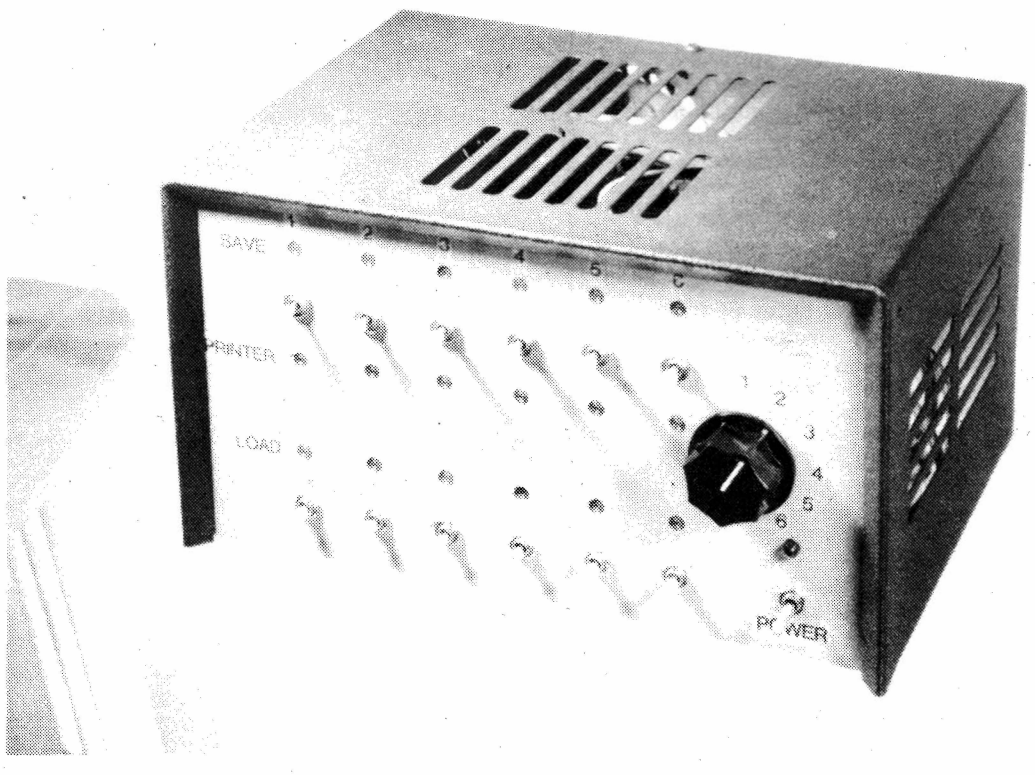
NAME _____
ADDRESS _____
_____

CHECK ONE: NEW _____ RENEWAL _____
1 YEAR SUBSCRIPTION: $14 _____
2 YEAR SUBSCRIPTION: $26 _____
3 YEAR SUBSCRIPTION: $36 _____
SAMPLE COPY: $2 _____ AIR MAIL $4 ___
CANADA/MEXICO: ADD $5/ YEAR
FOREIGN: ADD $12/ YEAR

AVOID THE 1980 PRICE INCREASES - SUBSCRIBE TODAY!







UNFLOPPY, cont.

WHAT OTHERS SAY—We talked to Dr. Richard Melville, an electrical engineer. he has used a prototype for 9 months, and not had a bad load in an estimated 10 to 100 megabytes, that was not due to tape fault (wrinkle, bubble, flaked emulsion, etc). It was always media, not system fault.

Dr. Melville is using the Radio Shack SCT-12 stereo tape deck, which is priced at $79.95. He records in stereo, but plays back in mono. If he gets a bad load, he switches to the other channel and about half the time can load on it. The error always has had to do with tape faults, but by changing channels he often gets around them. (How about putting mono heads in the stereo deck, and having: even better redundancy, or the ability to play both sides of the tape, without turning the cassette? How about that gang. Are you listening Radio Shack? Why not a CTR-82?)

User comments include: "Recommend it to all", "Excellent! A poor man's floppy", "Great. I can't believe it is so fast", "Great! Fast! Reliable! Worked first time!", "It is the answer to my prayers", and "The best through-the-mail device I have ever purchased".

JPC Products is not a new company, nor is their interface new. The firm has been making a version of the board for 6800-based systems for about three years. They, and their board, have built a solid reputation. They've been working on the TRS-80 version quite a while.

The 6800 version works at 4800 to 9600 baud. That is because the 6800 processor is better at certain instructions than the Z-80 is (how do you like that, Z-80 fans?), and the recommended tape deck is the $80 SCT-12, **not** the CTR-80 supplied with TRS-80.

Want to go faster than 3600 baud? Hook up the Bill Archbold clock speedup kit and go out at 7200 baud. The CTR-80 may not hold up its end, and you might have to substitute a better recorder. The SCT-12 is recommended. We will be trying **Poor Man's Floppy** with Archbold speedup next issue. The Archbold board (assembled, tested, guaranteed) is $24.95 ppd, from Archbold, 106 Snyder dr., Mather CA 95655.

Baud rates and quoted speedups are not as important as measured results. To see how fast the Poor Man's Floppy is, we loaded the fantastic Star Trek on People's Software tape 7—13K bytes in 210 seconds, at the regular TRS-80 rate. Next we saved it on Poor Man's Floppy, in 40 seconds. That's a better than 5X speedup. Archbald speedup would reduce that to 20 seconds. Next we loaded it from disk. Guess the speed. Twenty seconds, not counting the half minute or so it took to load Basic.

Watch out, Shugart! JPC Products Co. is at 12021 Paisano ct., Albuquerque NM 87112 (505) 294-4623. Master Charge and VISA accounts are honored.

SCHEMATIC for interconnect box is at upper left, and inside front and inside rear views are at left. Finished box is above, and will be demonstrated at the CIE booth at Computer Faire hourly, on the half hour.

Here's how you can hit Reset without

memory loss

By JACK M. MALONE
24671 Sadaba, Mission Viejo CA 92692

Are you tired of losing memory on a Reset with the expansion interface connected to your TRS-80? Well here's the fix.

Radio Shack in its design of the expansion interface figured everyone that bought one would be using disk drives with it. It designed the system so that on a system-reset, which sends the A-80 CPU to 0066 hex, that if an expansion interface was connected it would boot the TRSDOS into memory.

If no expansion interface was connected then it would send the CPU to the ready state.

This is fine except in the case of the expansion interface being used with no disk. You had to override the TRSDOS boot routine by holding down the Break key. This in turn dumped all your memory.

REMOVED CONTROLLER—I didn't have a disk system so this was a real inconvenience. When I reset everything went south.

So I removed the floppy disk controller chip and everything fine. The system reset just as before I added the expansion interface with no loss of memory and it returned to READY instead of MEMORY SIZE?

Then I received my KDFIX and real-time-clock tape from Radio Shack and lo and behold my real-time clock would not work with the FDC chip removed. Not a bad tradeoff. No clock, no problem with Reset.

Then I purchased the G2 Level II Basic and it just wouldn't work without the FDC chip in place, so back in went the chip, back to loosing my memory every time I reset. Totally not acceptable I told myself. I also told myself that I would never get any help from Radio Shack on this as the system is designed this way.

I purchased a service manual for the expansion interface, Radio Shack part number 26-1140 ($4.95) and there was the answer. On the pinout of the FDC chip it showed pin 3 as the chip select. Pin 3 was tied to ground (logic low). The instructions on the chip say a logic low on this pin 3 (CS) input selects the chip and enables computer communication with the device.

Using the old brain instead of the computer I figured if logic low selects the chip then a logic high would deselect it. Then it should react to the CPU as if there were no chip there. I also noticed Pin 18 (3PM) was tied into the +5V bus through a 10K resistor, giving it a logic high.

HERE'S FIX—Purchase a miniature SPDT switch from Radio Shack (part no. 275-662—$2.49) and mount it in the expansion interface right next to the power-on switch.

Take your shoes and socks off so you are well grounded to a tile or concrete floor.

Lift the FDC chip up and out of its socket.

Bend pin 3 (CS) out and re-insert the FDC chip back into its socket. Pin 3 is now protruding out the side and is not making contact with its socket.

Run a small wire from the center terminal of the switch to the pin 3 of the FDC chip that is sticking out and solder it.

Run 2 wires from the outer lugs of the switch to pin 18 (3PM) and pin 20 (VSS or ground) and wrap them around the FDC pins being careful not to short to pin 17 or 19.

Then quickly solder all connections.

Put the expansion interface back together. You have installed a switch that will either select the FDC chip or deselect it.

USING IT—There are lots of combinations on how to use this:

1) If not using disk, G2 Level III Basic or real-time clock, just leave the switch in the deselect position. Reset at any time without depressing the Break key and you will reset without loss of memory and go to a READY state.

2) If using disk, G2 Level III Basic or real-time clock, leave the switch in the select position. If your program bombs or something sends the CPU off into left field and it won't come back on track, then just throw the switch to the deselect position, reset (again do not hold Break key down), then throw the switch back to select position and you haven't lost any memory.

3) On powerup with no disk system, throw the switch to deselect position, turn on the CPU, then throw the switch to select position if you need the FDC.

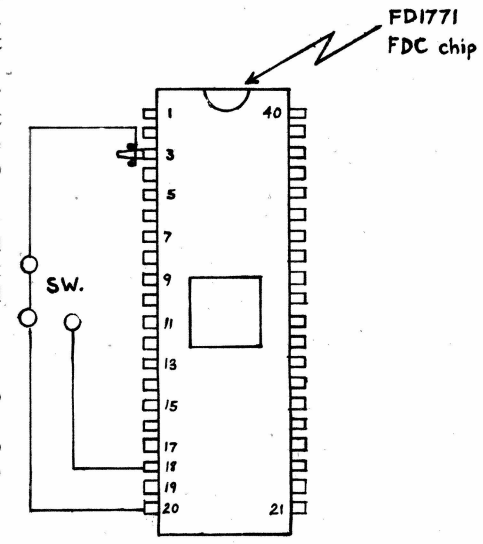
I might add it's still good practice to every so often dump to tape or disk the program that you are writing just in case the power should fail or something.

If using the real-time clock when you Reset it will stop and you will have to restart and reset it.

I hope this will help all TRS-80 owners that have expansion interfaces. I know it's made my system much easier to use and no worrying about Resets now.

If anybody has any questions, I'll be glad to answer them the best I can by mail or phone (714) 837-7734.

(Fantastic piece of work, Jack. I think everybody who has an expanded system should install this mod. Just one thing I would suggest: make that switch push button, so it returns to normal circuit when not needed. Then the procedure would be push deselect, holding down, push reset, release deselect. Shack has the item as part 275-1549, for $2.19. Bill)



SW. shown is the chip selectect mode

SW. is a SPDT

TRS-80 Level II ROM

COMMAND

statements & hex addresses

By GERALD J. ABEAR
Personal Computer Service

BASIC COMMAND		HEX ADDRESS
END	80	1DAE
FOR	81	1CA1
RESET	82	0138
SET	83	0135
CLS	84	01C9
CMD	85	4173
RANDOM	86	01D3
NEXT	87	22B6
DATA	88	1F05
INPUT	89	219A
DIM	8A	2608
READ	8B	21EF
LET	8C	1F21
GOTO	8D	1EC2
RUN	8E	1EA3
IF	8F	2039
RESTORE	90	1D91
GOSUB	91	1EB1
RETURN	92	1EDE
REM	93	1F07
STOP	94	1DA9
ELSE	95	1F07
TRON	96	1DF7
TROFF	97	1DF8
DEFSTR	98	1E00
DEFINT	99	1E03
DEFSNG	9A	1E06
DEFDBL	9B	1E09
LINE	9C	41A3
EDIT	9D	2E60
ERROR	9E	1FF4
RESUME	9F	1FAF
OUT	A0	2AFB
ON	A1	1F6C
OPEN	A2	4179
FIELD	A3	417C
GET	A4	417F
PUT	A5	4182
CLOSE	A6	4185
LOAD	A7	4183
MERGE	A8	418B
NAME	A9	418E
KILL	AA	4191
LSET	AB	4197
RESET	AC	419A
SAVE	AD	41A0
SYSTEM	AE	02B2
LPRINT	AF	2067
DEF	B0	415B
POKE	B1	2CB1
PRINT	B2	206F
CONT	B3	1DE4
LIST	B4	2E2E
LLIST	B5	2F29
DELETE	B6	2BC6
AUTO	B7	2008
CLEAR	B8	1E7A
CLOAD	B9	2C1F
CSAVE	BA	2BF5
NEW	BB	1B49
TAB	BC	
TO	BD	
USING	BF	
VARPTR	C0	24EB
USR	C1	27FE
ERL	C2	24DD
ERR	C3	24CF
STRING$	C4	2A2F
INSTR	C5	419D
POINT	C6	0132
TIME$	C7	4176
MEM	C8	27C9
INKEY$	C9	019D
THEN	CA	
NOT	CB	25C4
STEP	CC	
+	CD	249F
-	CE	2532
asterisk	CF	
slash	D0	
AND	D2	
OR	D3	
equals	D5	
SIGN	D7	
INT	D8	
ABS	D9	
FRE	DA	
INP	DB	
POS	DC	
SQR	DD	
RND	DE	
LOG	DF	
EXP	E0	
COS	E1	
SIN	E2	
TAN	E3	
ATN	E4	
CVI	E6	

TRS-80 Level II memory locations & starting

ADDRESSES

LOCATION HEX	LOCATION DECIMAL	DESCRIPTION
402D-402F	16429-31	—Return to TRSDOS. Holds C3 00 50
4040	16448	—Real-time-clock counter
4041-4043	16449-51	—Holds seconds, minutes, hours
4044-4046	16452-54	—Holds year, day, month
408E-408F	16526-27	—Starting address for SYSTEM program. USR (N)
40A0-40A1	16544-45	—Address of memory reserved for strings
40A4-40A5	16548-49	—Address of program storage space
40A6	16550	—Counter for video display column-cursor offset
40A7-40A8	16551-52	—Starting address for I-O buffer
40AB 40AC	16555-56	—Seed for random number generator
40B1-40B2	16561-62	—Highest memory address after the one given in answer to MEMORY SIZE?
40DF-40E0	16607-08	—Starting address for SYSTEM command
40E6-40E9	16614-17	—Initial address for stack pointer
40F7-40F8	16631-32	—Address of LINE number where Basic CONTinues after stop or break
40F9-40FA	16633-34	—Table address of simple variables & strings
40FB-40FC	16635-36	—Table of arrays
40FD-40FE	16637-38	—Starting address of free memory
40FF-4100	16639-40	—Block of 26 decimal addresses corresponding to A thru Z. All initialized to 4 for single precision. Changed to 2 by DEFINT; 3 by DEFSTR; 8 by DEFDBL; reset to 4 by DEFSNG
411B-	16667	—Trace: set to 0 by TRON, AF by TROFF
4200	16896	—Start of floppy disk loader
18C9-18F6	6345-90	—Level II ASCII error code abbreviations
40E1	16609	—Auto flag: if not 0 then auto command is on
1A7E	6782	—Parses input contained in buffer
1E5A	7770	—Converts decimal to binary (integer precision)
1BC0	7104	—Makes the hex code for reserved words found in the buffer
1B2C	6956	—Locates room in program storage area to place a line held in buffer
4090	16528	—Addresses-instructions
4096	16534	—OUT N,0
409B	16539	—Line printer position
409C	16540	—Output device type: 1 printer; 0 display; -1 tape
409D	16541	—Display line size (40H — 64 dec.)
40A2	16546	—Last executed line PTR — FFFF if a line has not been executed
40AF	16559	—Data type: 2 integer; 3 string; 4 single-precision; 8 double-precision
40D8	16600	—PTR to last executed byte of hex code (from Basic or buffer)
40DA	16602	—Last data-read-line number
40DE	16606	—Read-input flag
40E2	16610	—Current input line number
40E4	16612	—Auto line number increment
40EA	16618	—line number
40EC	16620	—Current or error line
4130	16688	—Work area. Used for converting binary line number to displayable ASCII
4152-41A6	16722-06	—DOS function calls to DOS exits.

These memory addresses and the above command statment addresses were contriubted by Gerald J. Abear of Personal Computer Service, 1515 Andrews dr., Las Cruces NM 88001. They sell Basic3, much like Level 3 but less expensive ($24.95 pp), and with more features (we think). Also MLCOPY, system copying tape that allows placing any number of System tapes on one cassette ($14.95 pp), and DMSONL, Basic six-part program: edit, monitor, disassemble and more ($14.95 pp).

MECA TAPE:

Get 250% more storage at one-third cost of disk

By JEFFREY R. FITZSIMMONS
Box 521 Star rt., Earleton FL 32631

Recent articles in CIE publications have pointed out the potential advantage of using digital tape drives for mass storage. Actually, several large computer companies have been using them with great success for years, but for some strange reason, in small computing, audio tape has been succeeded by mini-disks.

The disadvantages of audio tape are obvious to all, but the relative value of digital tape over disk storage has been obscured. A disk unit may cost $500 and require an interface costing $300, for a minimum of $800 worth of hardware. This may yield 90K of memory and 1-3-second access time. In addition, a second disk drive is often needed (+ $500) due to limitations on the operating system disk. To make things worse, the memory overhead goes up so that the operating system can reside in RAM (add 16K of memory, $100-$200). This can bring you quickly to $1,500 worth (?-ED) of hardware with less than 180K of storage capacity!

A TAPE DRIVE such as the Beta-1 from MECA, however, offers a minimum of 500K of storage on a single tape, file-handling capability and access times of 5-10 seconds for $400. The system has a built-in processor with its own instruction set and a very small demand on system main memory (an additional 16K is not needed). **We are talking about one-third the cost and two-and-a-half-times the storage capacity.**

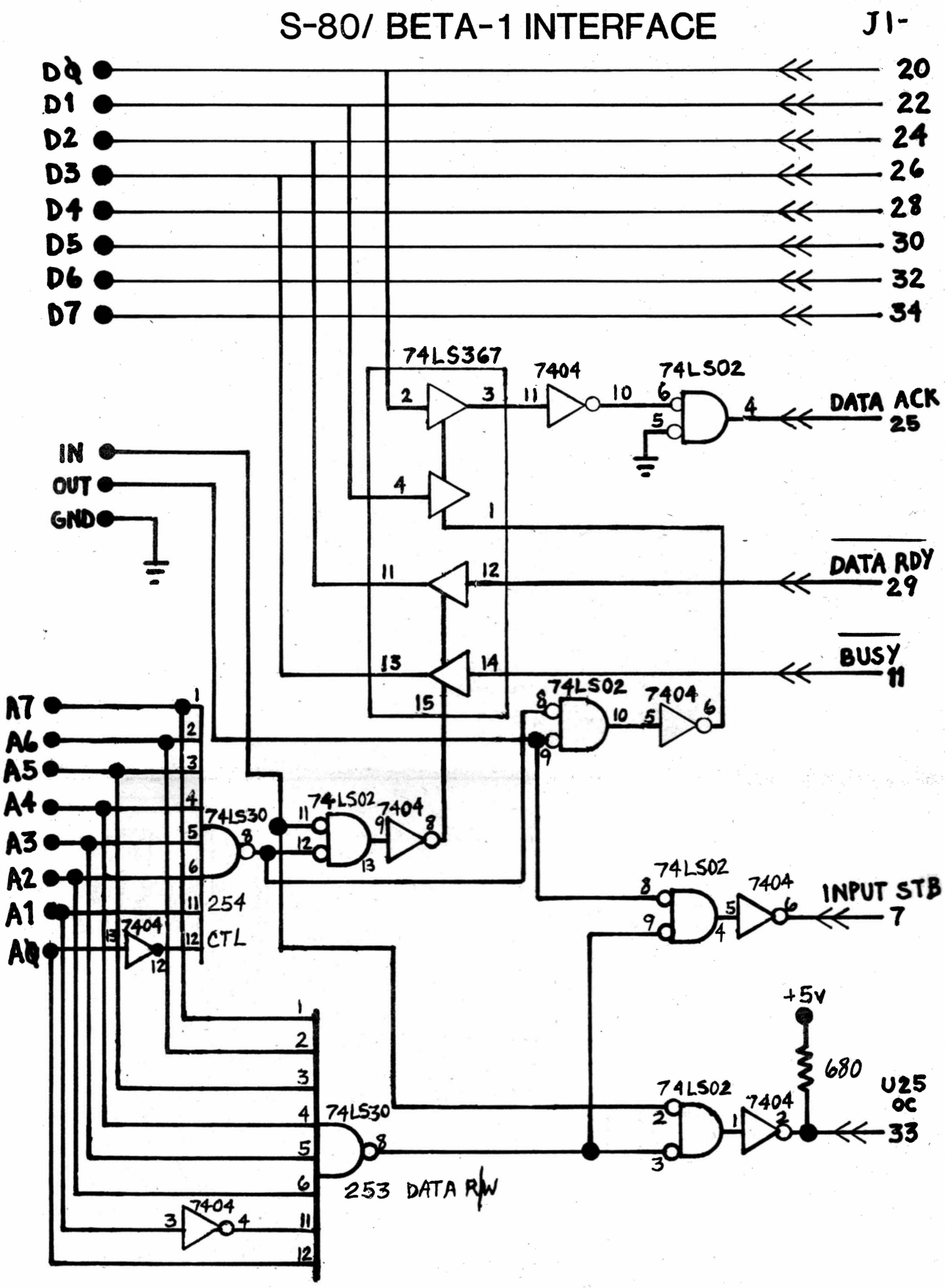
The tape drive has a minimum of control lines and can be interfaced with the TRS-80 quite easily. The cost of a minimal interface for parts alone is around $25 and all the parts can be obtained at your local Radio Shack store.

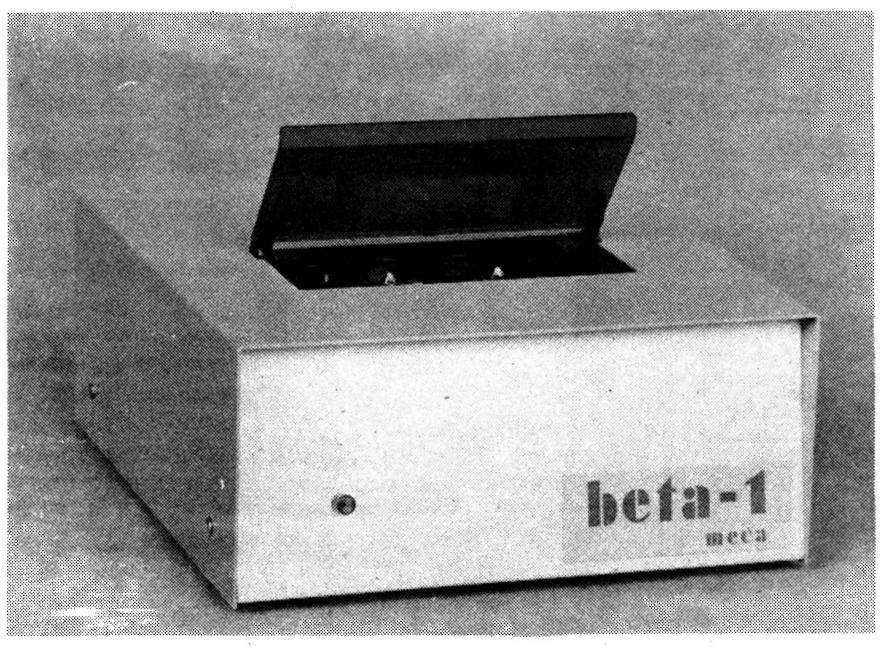
Basically, the Beta-1 and the TRS-80 must communicate over the eight-bit data bus for fast parallel operation. Parallel operation also greatly simplifies the minimum circuit requirements and has been found quite reliable. Only one minor modification is necessary to allow control over the Beta-1 data latches (output control). All other connections are as labeled in the accompanying schematic. Jumper W13 in the Beta-1 is removed and pin 1 of the 74LS374 is brought out on pin 33 of the ribbon cable. W5-W12 are jumpered allowing direct access to the TRS-80 system bus. A TRS-80 bus ribbon with appropriate connector is needed, and a common 44-pin edge connector can be used for mating to a small PC card. A single 5V regulator and DC supply also are required.

SOFTWARE REQUIREMENTS—The address lines utilize ports 253 and 254, which are easily controlled through Basic or machine language using single commands (INP and OUT). Port 254 is the control-status address, and 253 is the data port for reading and writing. Using Basic you can check the data-ready line with "IF NOT INP (254) AND 4 THEN NNN." A busy signal from the Beta can be checked with statements like "IF NOT INP (254) AND 8 THEN NNN." A data acknowledge can be sent to the Beta with "OUT 254, 0." Reading and writing data is simply a matter of designating a variable: "C equals INP (253)" (use equal sign) will place-transfer a byte of data from the Beta to the TRS-80. Output is simply the reverse "OUT 253, C" sends the byte contained in C to the Beta-1.

A Basic program using less than 2K can be written using the above control statements allowing the TRS-80 to move data to and from the Beta-1 in 256-byte blocks. A system buffer is established by answering the MEMORY? prompt with 32500 (16K systems). This will leave about 260 bytes protected in high memory. Data can be moved to and from this buffer by the controlling programs using the Beta-1 driver program as a subroutine. The Beta-1 then provides a fast and efficient means of storing large amounts of data with great reliability and low cost per byte.

A host of single-character commands are automatically decoded by the Beta-1's ROM and CPU. These commands allow you to read blocks, write blocks, seek, initialize, fill buffer, query status, verify, rewind, list the directory, open files, kill files, clear, select track and access the Beta-1 buffer. These functions are the essential abilities needed to construct almost any kind of data base the user might need.





Hardwire change manages lowercase

By RICHARD L. MAHN
2473 Draper, Ypsilanti MI 48197

I have modified my Level II TRS-80 for lowercase. Instead of providing a new display handler as Timothy P. Mann did in the November 1978 issue of TRS-80 Computing.

I have modified the original display handler. The original handler maps all alphabetic characters (40H - 5FH and 61H - 7AH) into control character values (00H - 1FH). This mapping is unnecessary and remains to me a mystery why this mapping was implemented. The only mapping really necessary for the original Level II is the folding of lowercase into uppercase.

To make the original display handler work with the display upper/lowercase modification, one bit of one byte in one ROM of the Level II ROMs must be changed from a one to a zero. Several ways are available to clear this bit i.e. a new ROM, UV ROM or ROM patch. Currently only the latter is practical.

The ROM patch consists of the following Location 0473H of the Level II ROM must be changed from a 38H (Jump Carry Relative) to a 28H (Jump Unconditional Relative). This instruction change removes all mapping of alphabetic characters. The enclosed figures detail the circuit changes for the ROM patch. The four (4) inexpensive 16 pin TTL chips detect address 0473H and force bit D5 (in D7 - D0 nomenclature) low. Thus changing the 38H data at that location to a 28H. The four chips can be added to a small perforated board on the Level II ROM board.

This software change by hardware is better than a switch (software or manual controlled) to select the operating mode and better than a RAM-resident patch to the display handler. My modification works from power up with no modifications or restrictions to software.

Note: By poking directly into the screen RAM the special display characters from OOH to 1FH can be displayed. It should be possible to incorporate an escape character into the normal control codes to allow these values to be transmitted to the display by normal means.

Eletelephony

```
ONCE THERE WAS AN ELEPHANT
WHO TRIED TO USE THE TELEPHANT.
OH NO - I MEAN AN ELEPHONE
WHO TRIED TO USE THE TELEPHONE

DEAR ME - I'M NOT CERTAIN, QUITE
THAT EVEN NOW I'VE GOT IT RIGHT.

HOWE'ER IT WAS, HE'D GOT HIS TRUNK
ENTANGLED IN THE TELEFUNK
THE MORE HE TRIED TO GET IT FREE,
THE LOUDER BUZZED THE TELEPHEE -
DEAR ME -
I FEAR I'D BETTER DROP THE SONG
OF ELEFOP AND TELEPHONG...


5   REM VIDEO PRINT BY JAMES GARON
7   REM SEPTEMBER 78 TRS-80 USERS GROUP
8   REM  OF ORANGE COUNTY
```

```
10 CLS:PRINT" E L E T E L E P H O N Y "
20 PRINT"ONCE THERE WAS AN ELEPHANT"
30 PRINT"WHO TRIED TO USE THE TELEPHANT."
40 PRINT"OH NO - I MEAN AN ELEPHONE"
50 PRINT"WHO TRIED TO USE THE TELEPHONE"
60 PRINT:PRINT"DEAR ME - I'M NOT CERTAIN, QUITE"
70 PRINT"THAT EVEN NOW I'VE GOT IT RIGHT."
80 PRINT:PRINT"HOWE'ER IT WAS, HE'D GOT HIS TRUNK"
90 PRINT"ENTANGLED IN THE TELEFUNK"
100 PRINT"THE MORE HE TRIED TO GET IT FREE,"
110 PRINT"THE LOUDER BUZZED THE TELEPHEE -"
120 PRINT"DEAR ME -"
130 PRINT"I FEAR I'D BETTER DROP THE SONG"
140 PRINT"OF ELEFOP AND TELEPHONG...";
150 GOSUB 30010
30000 END
30010 V=15360:LPRINTCHR$(29)
30020 FOR C=0 TO 15:FOR R=0 TO 63
30030 LPRINT CHR$(PEEK(V+64*C+R));
30040 NEXT:LPRINT" ":NEXT
30050 RETURN
```

Changes General Ledger

People's Software tape 1 is terrific. I have made the following corrections to the journal-ledger program, by Frank Heimisch. I have a TRS-80 16K Level II tape system.

1) CHANGED SEMICOLONS TO COMMA FOR LEVEL II.
```
2232 PRINT @522,'- - -
10090 PRINT @520,'- - -
12068 PRINT @520,'- - -
12080 PRINT @520,'- - -
12090 PRINT @520,'- - -
```
*** NOTE THE SINGLE QUOTE ARE MEANT TO BE THE DOUBLE QUOTES ***

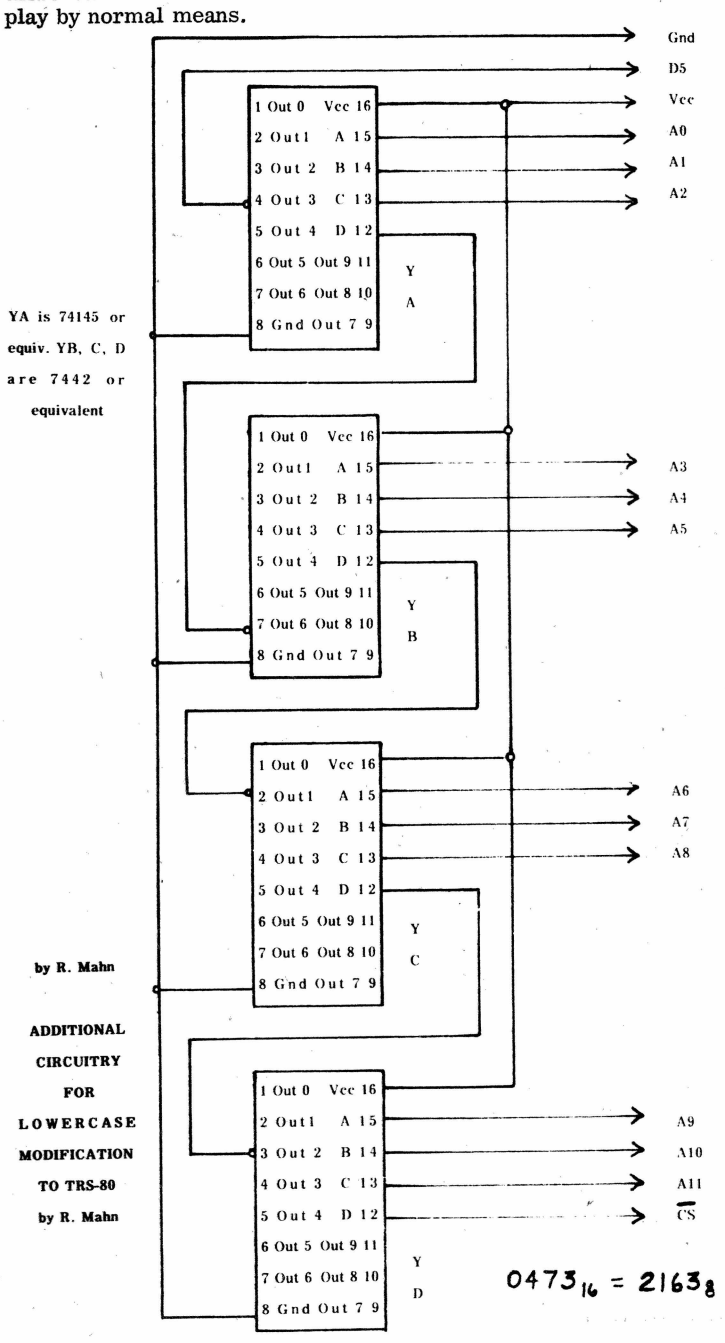
2) ADDED DIMEMSION STATEMENT FOR ARRAY.
```
23 DIM A(1600)
```

3) ADDED RESTORE STATEMENT
```
986 RESTORE
1276 RESTORE
2351 RESTORE
```

4) CORRECTION MADE
```
2140 IF INT((A(N*10+1))/1000)=D THEN 2160
12085 NEXT A:IF L=V-1 THEN 12093
```

5) CHANGES
LINES 119,715,716 CHANGE '+' TO 'OR'
LINE 940 CHANGE '*' TO 'AND'

CASEY OH



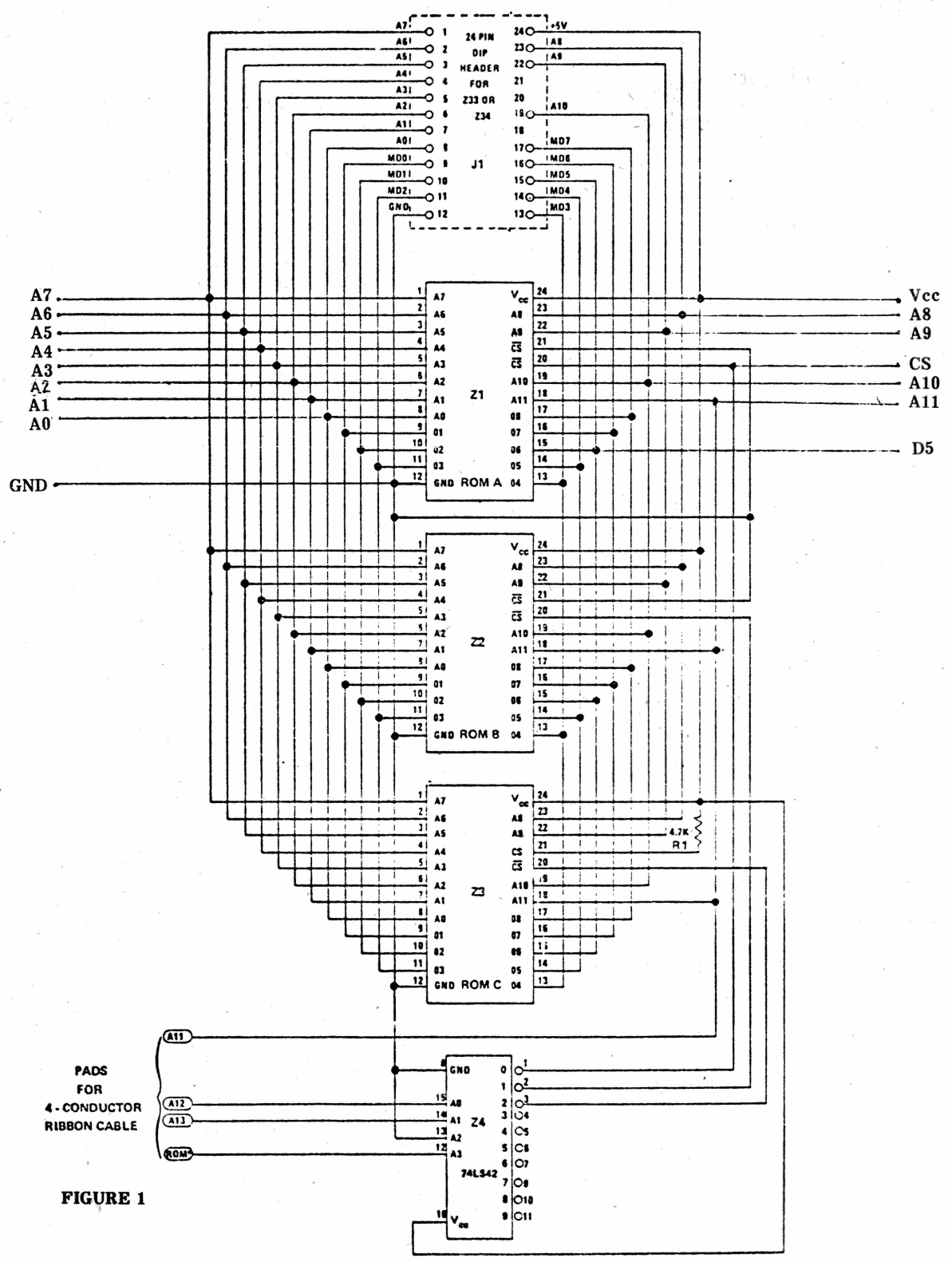


FIGURE 1

LEVEL II BASIC schematic showing tap points for lowercase mod by R. Mahn.

Electric typewritter

to store type on tape program makes it easy

By P. D. CAMPBELL
10126 Rittrell, Houston TX 77034

Lack of inexpensive hard copy is the biggest single shortcoming of the TRS-80. When I saw the Small System Hardware advertisement for its TRS-232 printer interface it seemed to be a solution.

When the TRS-232 arrived I took it and my Level II TRS-80 to the home of a lucky friend who owns a LA36 DECwriter. By following the simple and clear instructions we were soon producing hard copy in a very few minutes with absolutely no trouble or delay.

Returning home I faced the rather iszeable problem of determining which of the wires emerging from my old Model ASR-33 Teletype should be attached to the 20ma output of the TRS-232.

A little trial and error showed the green wire to be +22ma and the orange to be -20ma. The Teletype came to life ard has been producin, hard copy ever since.

The TRS-232 has performed flawlessly for over a month now. It is ected into the TRS-80 at all times and has not interfered with the cassette operation in any way.

```
10    REM TRS-80 ELECTRIC TYPEWRITER
20    REM LEVEL II 4K
30    REM TRS-232 SMALL SYSTEMS PRINTER INTERFACE
40    REM ASR MODEL 33 TELETYPE 20 MA CURRENT LOOP
50    REM WRITTEN BY P. D. CAMPBELL
60    REM 10126 KITTRELL, HOUSTON, TEXAS 77034
70    CLEAR 200
100   REM S=WRITE SCREEN AND CASSETTE
110   REM H=HARD COPY
150   REM SELECTION OF FUNCTION
160   INPUT "SCREEN & CASSETTE=S;  HARD COPY=H";F$
170   IF F$ = "S" THEN GOTO 200
180   IF F$ = "H" THEN GOTO 500
190   PRINT "INVALID FUNCTION"
199   GOTO 150
200   REM ***************************************
201   REM SCREEN AND CASSETE WRITER
210   GOSUB 800
220   INPUT "RECORD & PLAY KEYS DEPRESSED /
            HIT ENTER";X$
230   CLS
250   PRINT "MAXIMUM LINE LENGTH 72 CHARACTERS"
255   INPUT L$
256   IF LEN(L$) > 72 THEN GOTO 292
260   PRINT #-1,L$
270   IF L$ = "NO MORE" THEN GOTO 290
280   GOTO 255
290   GOTO 900
292   PRINT "LAST LINE IGNORED"
293   GOTO 250
500   REM ***************************************
501   REM HARD COPY
510   GOSUB 800
519   PRINT "TELETYPE ON LINE"
520   INPUT "PLAY KEY DOWN / HIT ENTER";X$
522   CLS
530   INPUT #-1,L$
535   IF L$="NO MORE" THEN GOTO 590
537   IF LEFT$(L$,2) = "//" THEN GOTO 600
540   LPRINT L$
550   PRINT L$
570   GOTO 530
590   LPRINT STRING$(5,10)
591   GOSUB 800
595   GOTO 900
600   REM ***************************************
601   REM PARAGRAPH INDENTATION
619   LPRINT STRING$(2," ");
620   LPRINT MID$(L$,3,70)
630   GOTO 550
800   REM ***************************************
801   REM THIS SECTION ENABLES MANUAL
            CASSETTE CONTROL
810   OUT 255,4
820   INPUT "REWIND AND POSITION CASSETTE /
            HIT ENTER";X$
830   OUT 255,0
850   RETURN
900   REM PROGRAM TERMINATION
990   END
```

```
10    REM SMALL SYSTEM HARDWARE TRS-232
        PRINTER INTERFACE LEVEL II 4K TRS-80
20    REM OUTPUT 20 MA CURRENT LOOP TO
        ASR-33 TELETYPE
30    REM WRITTEN BY P. D. CAMPBELL
        10126 KITTRELL, HOUSTON, TX 77034
40    REM A$ = ANY MESSAGE UP TO 72 CHARACTERS
50    INPUT "MESSAGE";A$
60    A = LEN(A$)
70    J = 1
80    B = (A/2)+1
90    FOR I=B TO 1 STEP -1
100   PRINT TAB(I)  MID$(A$,I,J)
110   LPRINT TAB(I); MID$(A$,I,J)
120   J = J+2
130   NEXT
150   FOR I=1 TO B
160   J = J-2
170   PRINT TAB(I)  MID$(A$,I,J)
180   LPRINT TAB(I); MID$(A$,I,J)
190   NEXT I
200   END
300   REM ******************************
310   REM EXAMPLE BY P. D. CAMPBELL
```

```
          T
         ETY
        LETYP
       ELETYPE
      TELETYPE
      TELETYPE O
    0 TELETYPE OU
   80 TELETYPE OUT
  -80 TELETYPE OUTP
 S-80 TELETYPE OUTPU
RS-80 TELETYPE OUTPUT
TRS-80 TELETYPE OUTPUT
TRS-80 TELETYPE OUTPUT
RS-80 TELETYPE OUTPUT
 S-80 TELETYPE OUTPU
  -80 TELETYPE OUTP
   80 TELETYPE OUT
    0 TELETYPE OU
      TELETYPE O
      TELETYPE
       ELETYPE
        LETYP
         ETY
          T
```

With this patch, you can use Electric Pencil with Apparat

Pencil - NewDOS

By RON LUCIA

(DOS compatibility is a growing problem in S-80 land. What if you want to use Electric Pencil with Apparat NewDOS? Ron, who for Int'l Jewelry Guild is author of the RADEX-10 data base management system, and sells it, and NewDOS and Electric Pencil, has written the following patch):

Modifying Electric Pencil for use with Apparat is an easy three-step process. First we will run Superzap, Apparat's disk display-modify routine.

superzap is a Basic program. Next we will find Pencil's location on the disk by looking on the directory (track 17 hex 11) of the diskette that contains Pencil (make sure that the file Pencil has not been killed). Then we will display the first sector of Pencil and modify three bytes, and the job will be done.

1(Load NewDOS disk in drive 0, get into Basic and run Superzap.

a. Select Display disk sectors.

b. Input relative disk, (the disk number that PENCIL-CMD is on.)

c. Input directory track (in hex), which is 11.

d. Input sector zero.

2. Look for PENCIL..CMD in right column of screen display.

$. If PENCIL..CMD is **not** in the right-hand column of the ASCII display **then** display the next sector by pressing the ";" key—and GOTO next instruction XX (please refer to figure 1) ELSE GOTO instruction 22a.

b. **If** PENCIL..CMD is in the right-hand column **then** look at the hex display to the left directly across from PENCIL..CMD and follow the bytes until you read 9642 9642 1800 (refer to figure 2).

c. The next two hex characters (following 9642 9642 1800) is the track (in hex) on which PENCIL will be located.

3. Press "K" key and reply to the track prompt with the hex characters from the above instruction (2c).

a. Input zero for sector.

b. Look for PENCIL in the top of the right-hand column of ASCII characters (see figure 3). **If** this sector appears exactly as it does in figure 3 **then go to** instruction 3c. **Else if** the sector does not appear exactly as it does in figure 3 **then** step to the next sector by pressing the ":" key until you get to sector 5). If the sector is not like figure 3 **then go to** instruction 3c. **If** the sector is not like figure 3 **then go to** instruction 1.

c. Type MODAF (which tells Superzap that we want to modify byte AF hex).

d. Change the six hex characters from 32 9B46 to 00 0000 by simply typing in six zeros. Press the "Enter" key.

```
011300  5D00 0000 0044 4952 2020 2020 2053 5953 .....DIR.....SYS
011310  A71D F9E5 0A00 1101 FFFF 0000 0000 0000 ................
011320  0000 0000 0000 0000 0000 0000 0000 0000 ................
011330  0000 0000 0000 0000 0000 0000 0000 0000 ................
011340  1000 0097 0050 454E 4D4F 4420 2050 434C .....PENMOD..PCL
011350  9642 9642 0D00 1B00 2021 FFFF FFFF FFFF .B.B.....!......
011360  0000 008D 004C 5631 4453 4B53 4C43 4D44 .....LV1DSKSLCMD
011370  9642 9642 0300 0D00 FFFF FFFF FFFF FFFF .B.B............
011380  1000 0000 0050 454E 4349 4C20 2043 4D44 .....PENCIL..CMD
011390  9642 9642 1800 0C00 0D00 1600 1701 FFFF .B.B............
0113A0  0000 0028 0054 5253 3233 3220 2053 5353 ...(.TRS232..SSS
0113B0  9642 9642 0700 1701 FFFF FFFF FFFF FFFF .B.B............
0113C0  1E00 00FD 0043 4F50 5920 2020 2043 4D44 .....COPY....CMD
0113D0  EB29 210E 0500 0620 FFFF FFFF FFFF FFFF .)!.............
0113E0  1000 0CAB 064D 454D 4D41 5020 2020 2020 .....MEMMAP.....
0113F06 9642 9642 0700 0B20 1620 FFFF FFFF FFFF .B.B............
```

This is the track in which the first sector of pencil starts.

This indicates that the program PENCIL/CMD is in the directory. This may or may not appear the same on your disk.

FIGURE 1—Pencil-NEWDOS patch by Ron Lucia.

```
   :-------- This will be different on your screen.
   :
<--:-þ
111380  1000 0000 0050 454E 4349 4C20 2043 2D44 .....PENCIL..CMD
111390  9642 9642 1800 0C00 0D00 1600 1701 FFFF .B.B............
(figure 1)              <:->
                         :
                         :
          0C = the track the program starts on.
          00 = the granule in which the program starts
               IF this byte is less than 20 THEN the program
               starts in sector zero
               IF this byte is greater than 20 THEN the program
               starts in sector five.
```

FIGURE 2—Pencil-NEWDOS patch.

```
00C000  0102 8955 C35F 5354 5253 444F 5332 2E31 ...U..STRSDOS2.1
00C010  5045 4E43 494C 3031 3232 3739 01E6 0611 PENCIL012279....
00C020  6E5C 211E 5AED B8C3 2D40 2197 5422 3062 ..!.Z...-@!.T"0.
00C030  3EC3 322F 6221 C300 22DB 6121 1C54 22DC >.2/.!..".."..!.T".
00C040  6121 C300 22DA 6621 8954 22DB 6621 C300 .!..".."..!.T"..!..
00C050  22E4 6621 9054 22E5 662A 1640 2230 5321 "..!.T"..*.@"0S!
00C060  0A54 2216 4021 FD59 11CD 6B01 0300 EDB0 .T".@!.Y........
00C070  2100 5A11 ED6B 0103 00ED B021 035A 119A !.Z........!.Z..
00C080  5D01 0300 EDB0 2109 5A11 B568 0103 00ED ......!.Z.......
00C090  B021 0D5A 1113 6601 0300 EDB0 3E20 322C .!.Z........>.2,
00C0A0  6621 BD59 22B1 5C21 EB69 22DF 613E F332 .!.Y"..!.."..>.2
00C0B0  9B46 C36F 5CE6 2FFE 20C0 3E1E C9F3 C5E5 .F..../...>.....
00C0C0  F5C3 D06B E1C1 FBC9 F53E 20CD AA6B F1CD .....>...*0S".@!.
00C0D0  AA6B C3B8 683E 00C9 2A30 5322 1640 2100 .....>..*0S".@!.
00C0E0  3C22 2040 C32D 403A 6964 FE00 2836 C5D5 <".@.-@:....(6..
00C0F0  E5ED 4BF5 5911 3253 2137 66ED B0ED 4BF5 ..K.Y.2S!7....K.
```

FIGURE 3—Pencil-NEWDOS patch.

Chart explains graphics patterns

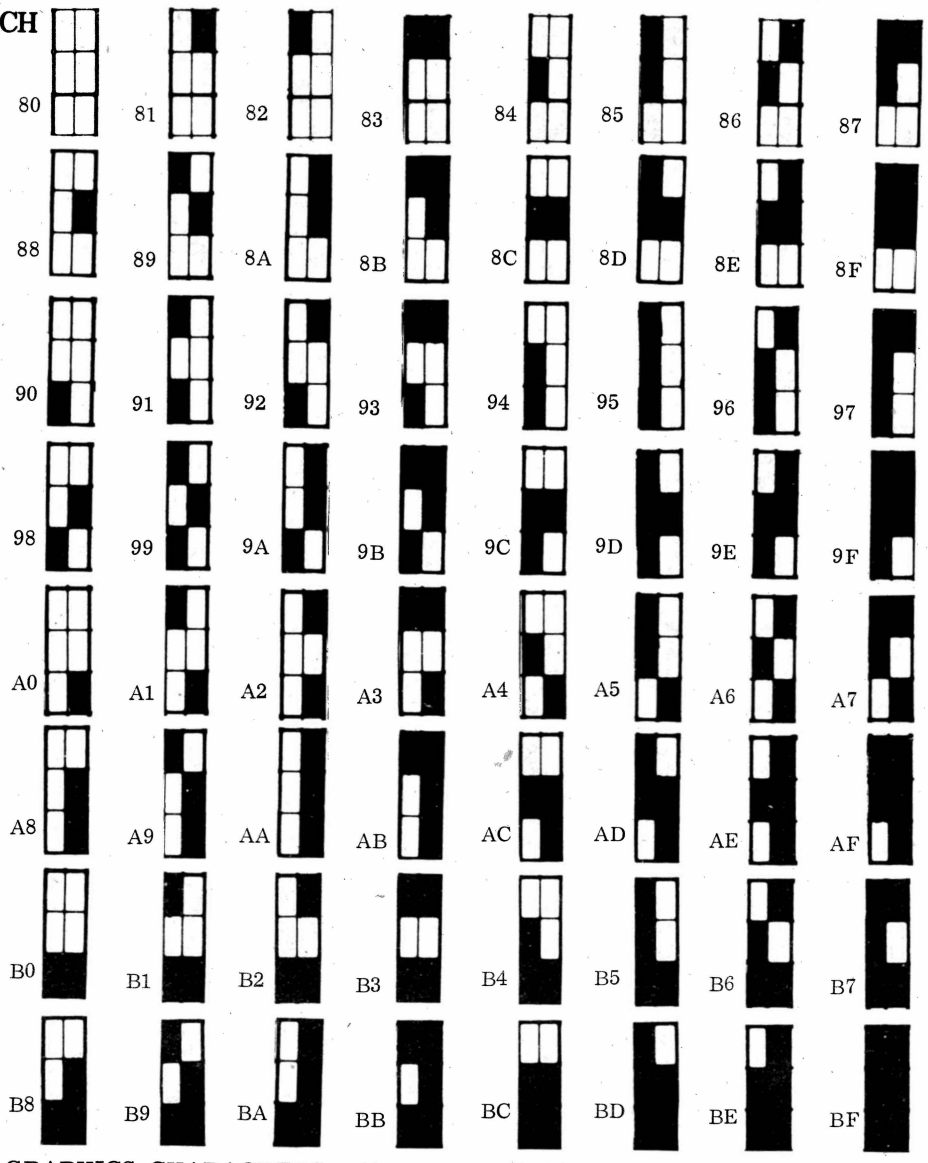
The graphics routine on appendix page C/2 of the TRS-80 Level II manual does not separate these graphics patterns very well.

This chart is a handy reference when formulating graphics displays.

The number below each bit pattern is decimal, starting with 129 at print location 450. Each bit pattern is stepped Four print Locations, ending with 191 at print Location 890.

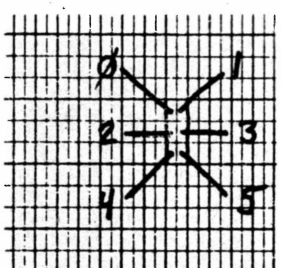
I used this sheet to create a graphics lettering program that produces an alphanumeric display that is three print lines high for each character.

—KENNETH G. BUSCH
12615 Higa Pl.
San Diego CA 92128

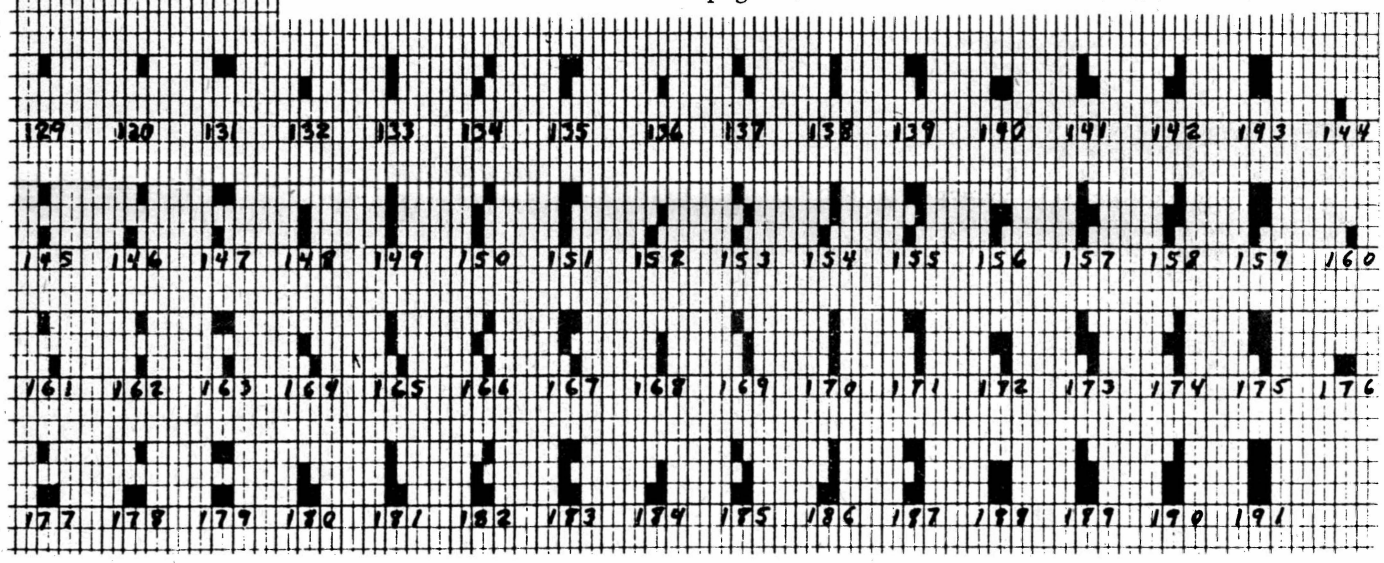


GRAPHICS CHARACTERS — Above are all 64 of them, courtesy of James Garon. Kenneth Busch arranged them numerically as in the ASCII set, skipping "blank" (Hex 80), below.

GRAPHICS BIT POSITION:
Bit six not used in graphics,
Bit seven is always set.



ASCII CHARACTER CODES from 129 to 198 for graphics — See page C/2 of Level II BASIC Manual.



Graphics faster than speeding bullet

By JAMES GARON

Have you ever caught yourself falling asleep while waiting for a BASIC graphics routine to finish? A short program to SET all 6144 graphics blocks;

```
1Ø FOR Y = Ø TO 47
2Ø FOR X = Ø TO 127
3Ø SET (X,Y)
4Ø NEXT X
5Ø NEXT Y
6Ø GOTO 6Ø
```

takes about 60 seconds to light the whole screen. Is there a better way?

The answer is "YES", but we must leave the comfort and security of BASIC behind us, and venture into the mysterious realm of Z-8Ø (Machine Language). It is not my purpose to explain how to program in Z-8Ø, but to share the flavor of Z-8Ø graphics, and perhaps motivate one or two readers to explore the subject further. (An excellent place to start is "Z-8Ø Microcomputer Handbook" by William Barden, Jr.) Howard W. Sams & Co., $8.95).

Let's begin by taking a closer look at a Print Position. Take a look at a video display worksheet (there's a copy in your manual). Notice that a print position is divided into six smaller blocks-two across and three down. When you tell the TRS-80:

```
NEW
1ØØ SET (Ø,Ø)
RUN
```

it does not look for a small block in the upper left corner of the screen and "turn it on"; it actually prints a special graphic symbol: (A) consisting of one lit block, and five dark (unlit) blocks; (of course the lines are not printed). If you now type;

```
2ØØ SET (1,Ø)
RUN
```

you have actually asked the machine to do something rather elaborate. The machine cannot simply print (in position Ø) the character: (B), since you do not want the dot at (Ø,Ø) to go out! So the machine must replace the previous: (A) with: (C)

If we now change line 1ØØ to,

```
1ØØ SET (Ø,2)
RUN
```

Then line 1ØØ prints (in position Ø) a: (D) and line 2ØØ causes it to be replaced by: (E).

Notice that we have not changed line 2ØØ, but the results of executing line 2ØØ are different.

If all of these graphics characters are available to the TRS-8Ø, why not to us, the users? The answer, excitingly, is that they are—and this brings us back to Z-8Ø language.

(By the way, you may wish to figure out how many different graphics characters there are, if any of the six blocks can be "on" or "off"— Remember to count the "all off" character! For those who don't like math, the answer will be given soon...)

All characters that the machine can print take up one byte of memory. A byte is eight BInary digiTS or "bits" which are either "on" (1) or "off" (Ø). To the machine, for example, the letter 'A' looks like this:

Ø1ØØ ØØØ1 ,

while the number "7" (as printed on the screen) looks like:

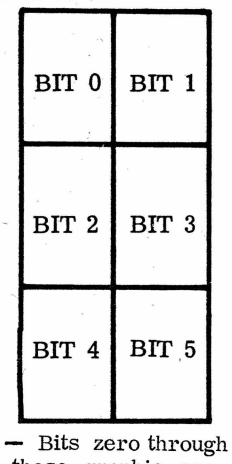
ØØ11 Ø111

For a complete list, see page 299 of Bardon's book, mentioned earlier. The bits in a byte are numbered as shown:

7,6,5,4,3,2,1,Ø

Notice that bit 7 is "off" in both letters and numbers (in fact, bit 7 is off in all characters entered thru the keyboard).

If bit seven is "on", this tells the machine to print a graphics character. For example, 1ØØØ ØØØ1 is our old friend (A) who was printed in the upper-left corner of the screen in line 1ØØ (before we changed that line). The six bits on the right —5,4,3,2,1,Ø— each correspond to one of the six small blocks in a graphics character. Since each block has two states (on or off), there are 2*2*2*2*2*2 (LEVEL 2'ers, that's 2↑6) or 64 different graphics characters. (Bit 6 is ignored). The diagram shows which bit goes with which block:



BIT 'MAP' — Bits zero through five control these graphic segments.

So, for instance, to print: (F), we use the byte: 1Ø1Ø Ø11Ø or—using the more compact HEX (adecimal) or base-16 number system—A6. Bit 7 says "This is a graphics character". Bits 5,2 and 1 light the three blocks we wanted. A complete chart of all 64 graphics characters and their hex equivalents is included in this article. Now VIDEO MEMORY lives from 3CØØ to 3FFF (hex), so here (in machine language) is a program to light the entire screen. (don't blink !!!):

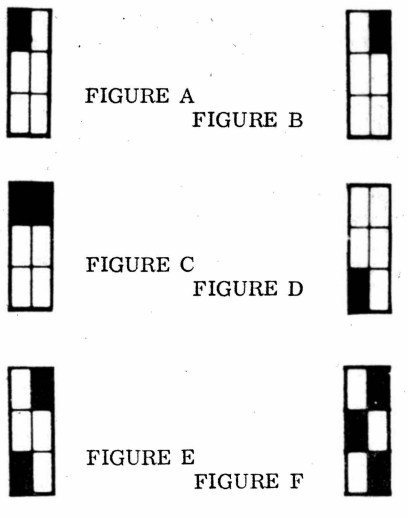
```
LITEON 3E 3B     LD A, 3B(HEX)
       21 FF 3F  LD HL, 3FFF      address of lower right print position (1Ø24)
LOOP   36 BF     LD (HL),BF(HEX)  place character on screen
       2B        DEC HL           point at next position
       BC        CP H             have we gone off the top of the screen yet?
       2Ø FA     JR NZ, LOOP      if not, go print in next position
       C9        RET              return to main program
```

If we wanted to REVERSE the screen (turn off all lit blocks & vice versa), we could use the next subroutine:

```
RÉVERS 21 FF 3F  LD HL, 3FFF
LOOP2  7E        LD A,(HL)        copy character into accumulator
       2F        CPL              change all bits
       CB FF     SET 7,A          reset graphic indicator
       77        LD (HL),A        put new character on screen
       2B        DEC HL
       3E 3B     LDA 3B
       BC        CP H             off screen?
       2Ø F5     JR NZ, LOOP 2
       C9        RET
```

When one considers the subroutines such as these execute in the wink of an eye, the true power of the Z-8Ø becomes apparent. If you have never worked in such a language, or indeed never gave it much thought—don't be put off by the strange-looking subroutines. Just set them aside, or share them with a friend who knows Z-8Ø, and who knows, some day you may decide to explore the subject further, and these subroutines may come in handy!

—Reprinted from Orange County TRS-80 Users' Group Newsletter, available by donation, $10 a year from user group, 24232 Tahoe ct., Laguna Niguel CA 92677.



Figures A-F, graphics symbols.

COBOL, cont.

Text books, self-teaching manuals, programming reference manuals and other support materials have been published on a large scale over the years, and much of it is available at the public and college library, or the local book store.

High-speed compilation, which exceeds initial design goals, has been made possible by use of unique state-of-the-art techniques. Comparisons made with existing Cobol compilers on large computer systems indicate performance improvements by as much as a factor of 10.

Cobol programs as large as 4300 statements have been compiled at the rate of over 700 per minute. In this respect alone, it is held, Nevada Cobol is a superior performer.

High-speed execution of machine code generated by Nevada Cobol was a major design objective during the three year development of the compiler. The most modern techniques available on the most widely-used microprocessor on the market today have been implemented, according to Ellis. A small benchmark program which appeared in Kilobuad magazine was coded in Nevada Cobol, Level II Basic and the Microsoft Basic compiler. The benchmark programs were executed on a standard TRS-80 mod. 1. Results of these tests will be reported in the Feb.-Mar. **S-80 Bulletin**.

Nevada Cobol for TRS-80 model 1 and 2, both running under CP-M, is $99.95 from Business Microproducts, Livermore Financial Center, 1838 Catalina ct., Livermore CA 94550 (415) 443-4876. Visa and Master Charge accepted. Immediate delivery; manual only is $25.

Check writer

By JOHN W. MELTON III
910 Saddleback ct, McLean VA 22102

As per your notation on the listing for Peoples' Software tape 2, the check writer program does indeed seem to have some bugs.

1. I HAVE ADDED A 'DEFDBL A'. IN SINGLE PRECISION, 'A' AS PRINTED ON THE SCREEN, WAS BEING ROUNDED WHEN A 7 DIGIT NUMBER SUCH AS $98556.21 WAS ENTERED TO $98556.2.

```
1  REM * CHECK WRITER *
2  REM * DEBUGGED *
3  REM * 5/31/79 *
5  CLEAR 100
6  DEFDBL A
7  CLS
10  PRINT @ 404, 'CHECK WRITER'
```

2. 'PRINT USING' STATEMENTS HAVE BEEN SUBSTITUTED FOR THE 'PRINT ,,,'$';A' STATEMENT TO PUT THE AMOUNT AT THE TOP OF THE 'CHECK'.

```
265  A$='**$##,##.##'
267  B$='**##,##.##'
270  PRINT ,,,USING A$;A
280  PRINT ,,,USING B$;A
```

3. IF AN AMOUNT LESSS THAN $2.00 IS ENTERED, THE SCREEN 'CHECK' NOW INDICATES 'ONE DOLLAR', NOT 'DOLLARS'.

```
492  IF A<=1.99 THEN PRINT 'DOLLAR ';
494  IF A>1.99 THEN PRINT 'DOLLARS ';
DELETE LINES 239,500,520
IN LINE 470 CHANGE 500 TO 492
```

4. $0.01 IS NOW PRINTED AS 'ONE CENT', NOT 'CENTS'. IF AN EVEN DOLLAR AMOUNT WAS ENTERED ON A 'CHECK' NO 'CENTS' NOTATION WAS MADE ON THE SCREEN. IT NOW APPEARS.

```
555  IF A1>1 THEN 560
556  IF A1=0 THEN PRINT ' NO CENTS'
557  IF A1=1 THEN PRINT ' 0';A1;'CENT'
558  GOTO 570
560  IF A1<10 THEN PRINT ' 0';A1;'CENTS'
        ELSE PRINT A1;'CENTS'
```

5. OTHER MODIFICATIONS

```
650  FOR X=1 TO 12+(N1-20)/10+20
751  GOTO 760
752  RESTORE
753  FOR X=1 TO 12+N1
754  READ X0$
755  NEXT X
756  PRINT X0$;' ';
```

* NOTE: ALL THE SINGLE QUOTE ARE MEANT TO BE DOUBLE QUOTES. *

AS THE PROGRAM NOW STANDS, IT WOULD BE SUITABLE FOR AN INDIVIDUAL TO REFORMAT THE 'CHECK', AND ADAPT IT FOR USE WITH A LINE PRINTER (AS I PLAN TO DO).

I HOPE THIS FIX AND MODIFICATION MAY BE OF USE TO FUTURE PURCHASERS OF THE PEOPLE'S SOFTWARE TAPES.

JOHN W. MELTON, III

Travelers checks, drawer building

By VERN L. BARNEY, D.C.
2441 E. 77th st., Long Beach CA 90804

Here are two short programs that might be of interest to readers:

Traveler's checks-

I have always been in a quandry when I pruchase them. The bank gives us one record to copy our numbers on, where do I put it so that it is in a safe place if the checks are lost, also if it is in a safe place how do I keep track of the ones that I cash?
The enclosed little program takes care of that. I make up several copies, keep one in my wallet, one with the checks, leave one at home and put another one in the car or motel room. I keep track of the ones cashed on the copy with the checks, when I get a chance I record the spent ones on the other copies.

```
0   REM  TRAVELER'S CHECK RECORD
1   CLS
2   PRINT "!!TURN ON PRINTER!!"
5   'RIGHTS CONCERNING ALL COMMERCIAL AND
       MACHINE-READABLE FORMS OF
6   'THE PROGRAM REMAIN WITH THE AUTHOR,
      VERN L. BARNEY, D. C.
7   '2441 E. 7TH STREET, LONG BEACH, CA. 90804
10  'TRAVELERS CHECKS
15  PRINT"TRAVELERS CHECKS"
20  INPUT"WHAT DENOMINATION";A$
30  INPUT"STARTING NUMBER";B$
35  INPUT"ENTER LAST THREE DIGITS";A
40  INPUT"HOW MANY CHECKS";B
45  LPRINT"TRAVELER'S CHECK RECORD"
50  LPRINT TAB(8) B$," DATE","WHERE SPENT":LPRINT
60  FOR N = 1 TO B
70  LPRINT A$".00",A;"    ---";"                -"
80  A=A+1
90  NEXT
100 LPRINT" ":LPRINT" "
```

RUN

```
TRAVELER'S CHECK RECORD
        365        DATE              WHERE SPENT

20.00               121      ---              -
20.00               122      ---              -
20.00               123      ---              -
20.00               124      ---              -
20.00               125      ---              -
20.00               126      ---              -
20.00               127      ---              -
20.00               128      ---              -
20.00               129      ---              -
20.00               130      ---              -
```

Drawer building-

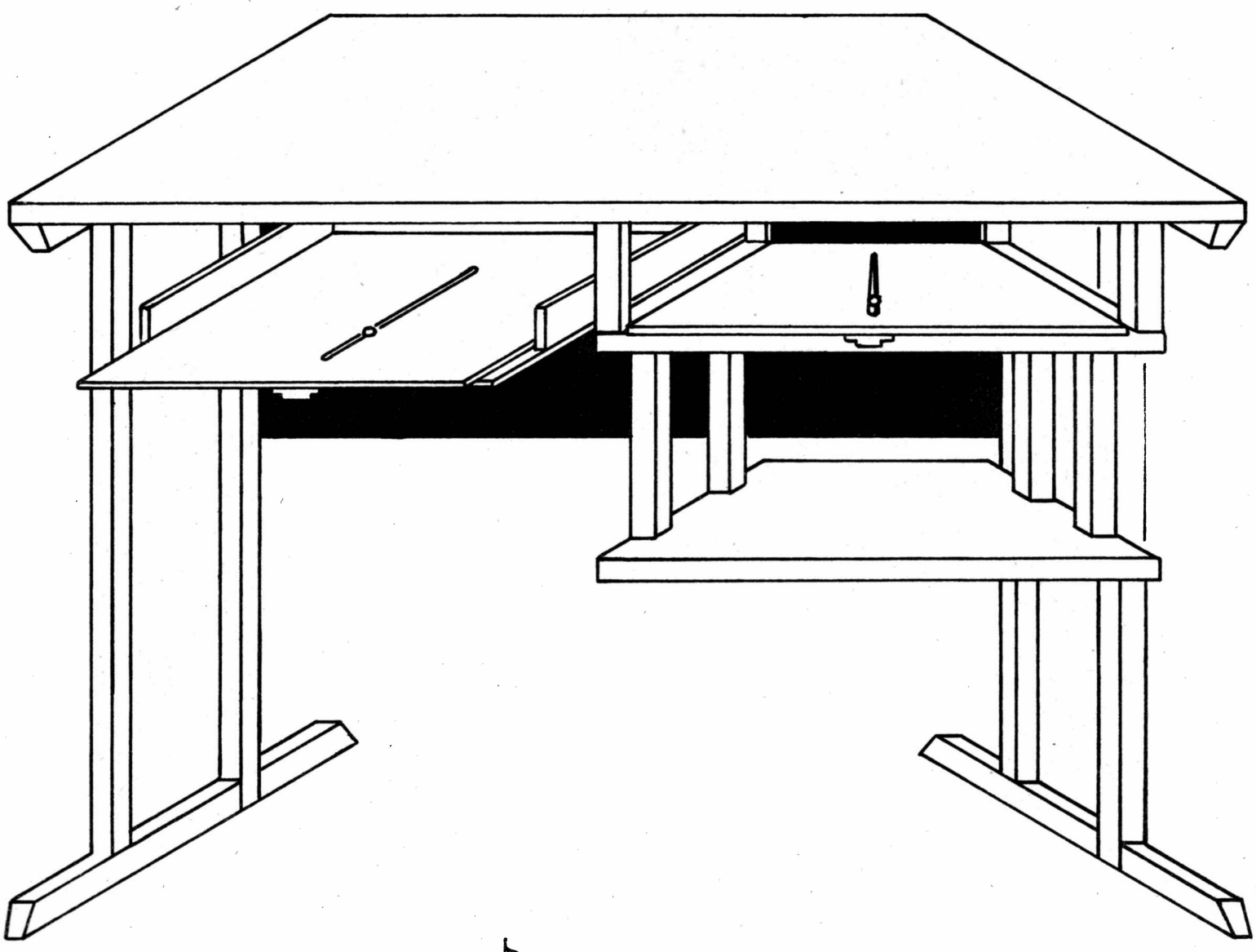
I have a hobby of woodworking and periodically make something that needs drawers. However, usually it has been so long since I last made a drawer I forget the formula for making one that will leave clearance for the opening. ing.
I use the dovetail template sold at Sears and a router with a dovetail bit.
The enclosed program gives me all of the dimensions to cut out the front, back, sides, and bottom for any drawer opening, either flush front or rabbeted.
I take the dimensions of the drawer opening, go to the computer, Run the program and all of the dimensions for the drawer parts are printed out on the line printer. Take the copy to my workshop and start cutting them all out at one time.
This program is designed for dovetail joints, but with a little modification it can be used for any type of joint. I wrote it so it will print on the printer as well as on the video display. Naturally if anyone tries to run it without a printer, without taking out all of the Lprint lines, the computer will hangup.

```
0   'DRAWER BUILDING FORMULA PROGRAM
1   'RIGHTS CONCERNING ALL COMMERCIAL AND
       MACHINE-READABLE FORMS OF
2   'THE PROGRAM REMAIN WITH THE AUTHOR,
       VERN L. BARNEY, D. C.
3   '2441 E. 7TH STREET, LONG BEACH, CA. 90804
5   CLS
6   PRINT"FOR FLUSH DRAWER: ALLOWS 1 5/8 IN.
     CLEARANCE AT BACK!"
7   PRINT:PRINT"FOR RABBET: ALLOWS 2 IN.
     CLEARANCE!":PRINT
20  PRINT"DIMENTIONS ARE BASED ON 3/4 IN.
     THICKNESS OF BACK & FRONT!"
30  INPUT"DEPTH OF CABINET OPENING";A
40  INPUT"THICKNESS OF DRAWER SIDES";K
50  INPUT"FOR FLUSH TYPE 0; RABBET TYPE 1";B
60  IF B=1 THEN 160
70  INPUT"(FOR FLUSH) WIDTH OF DRAWER OPENING";C
80  INPUT"HEIGHT OF DRAWER OPENING";D
85  LPRINT"     FOR DRAWER OPENING ";C;
     " BY ";D;" INCHES:"
90  LPRINT"SIDES WILL BE";A-2.5;" BY";
     D-.372;" INCHES"
91  PRINT"SIDES WILL BE ";A-2.5;" BY ";
     D-.375;" INCHES."
95  PRINT
100 LPRINT "BACK WILL BE ";C-.125;" BY ";
     D-.375;" INCHES"
101 PRINT"BACK WILL BE ";C-.125;" BY ";
     D-.375;" INCHES."
105 PRINT
110 LPRINT"FRONT WILL BE ";C-.125;" BY ";
     D-.125;" INCHES"
111 PRINT"FRONT WILL BE ";C-.125;" BY ";
     D-.125;" INCHES."
115 PRINT
120 PRINT"BOTTOM WILL BE ";A-2.75;" BY ";
     C+.375-(2*K);" INCHES."
121 LPRINT"BOTTOM WILL BE ";A-2.75;" BY ";
     C+.375-(2*K);" INCHES"
125 PRINT
140 INPUT"ANY MORE DRAWERS- IF YES (1); NO (0)";
     H : IF H=1 THEN 70
150 END
160 INPUT"(FOR RABBETED) WIDTH OF DRAWER OPENING";
     E
165 PRINT
170 INPUT"HEIGHT OF DRAWER OPENING";G
175 LPRINT"     FOR (RABBETED) DRAWER OPENING ";
     E;" BY ";G;" INCHES:"
180 LPRINT"SIDES WILL BE ";A-2.5;" BY ";
     G-.375;" INCHES."
185 PRINT:PRINT"SIDES WILL BE ";A-2.5;" BY ";
     G-.375;" INCHES."
190 PRINT"BACK WILL BE";E-.125;" IN. LONG AND";
     G-.375;" IN. HIGH."
191 LPRINT"BACK WILL BE ";E-.125;" BY ";
     G-.375;" INCHES."
195 PRINT
200 PRINT"FRONT WILL BE";E+.625;"INCHES LONG AND";
     G+.75;" INCHES HIGH."
201 LPRINT"FRONT WILL BE ";E+.625;" BY ";
     G+.75;" INCHES."
202 LPRINT
     "    (MAKE A 38 IN. RABBET ALL AROUND FRONT!)"
203 LPRINT
     "     (TAKE AN EXTRA 1/8 IN. RABBET AT TOP!)"
205 PRINT
210 PRINT"BOTTOM WILL BE";A=2.75;" BY ";
     E+.375-(2*K);" INCHES."
211 LPRINT"BOTTOM WILL BE ";A-2.75;" BY ";
     E+.375-(2*K);" INCHES."
230 INPUT"ANY MORE DRAWERS IF YES (1); NO (0)";
     J:IF J=1 THEN 160
```

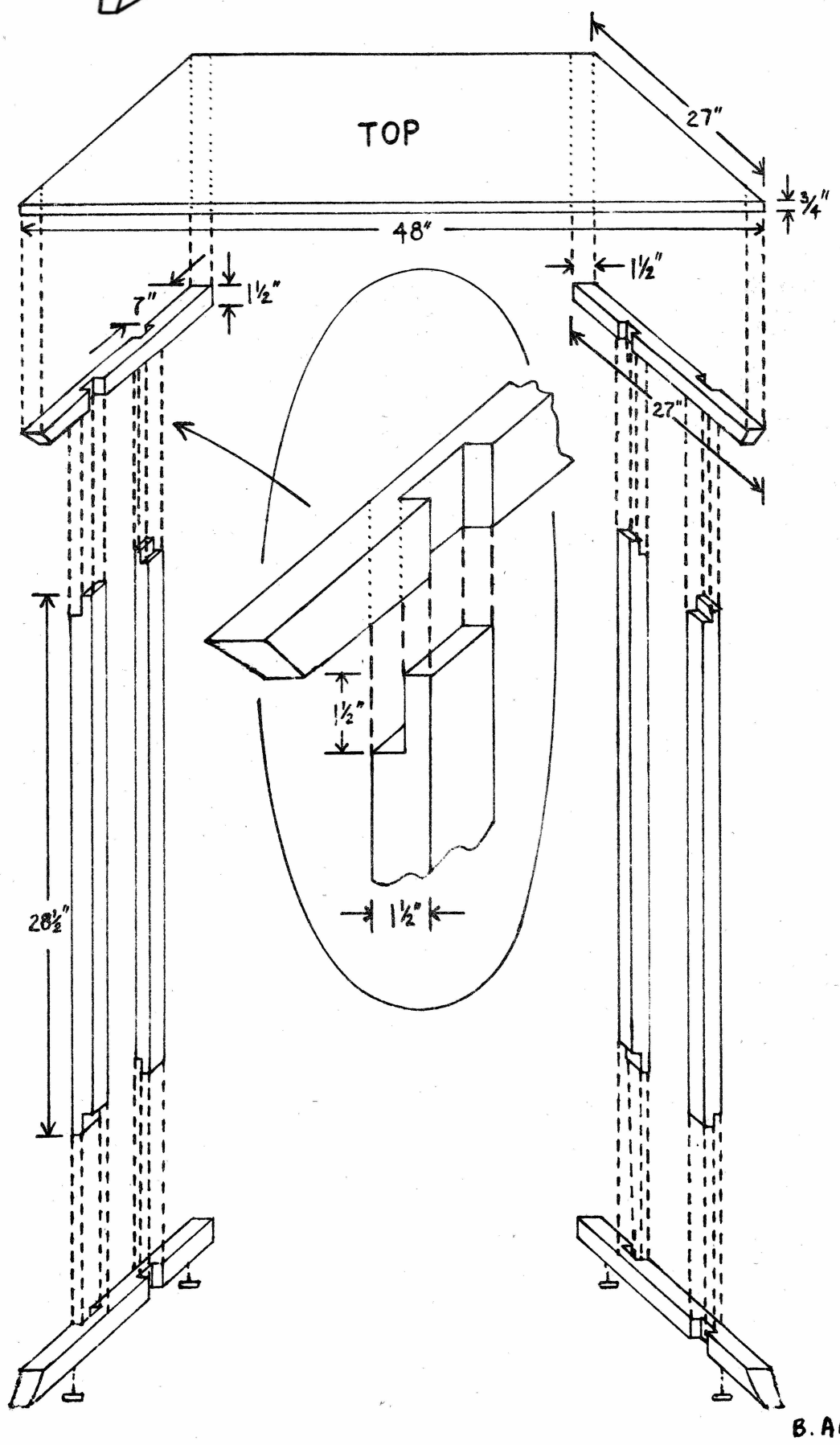
RUN

```
    FOR DRAWER OPENING  5  BY  5  INCHES:
SIDES WILL BE 4.5  BY 4.628  INCHES
BACK WILL BE  4.875  BY  4.625  INCHES
FRONT WILL BE  4.875  BY  4.875  INCHES
BOTTOM WILL BE  4.25  BY  5.375  INCHES

    FOR (RABBETED) DRAWER OPENING  5  BY 5  INCHES:
SIDES WILL BE  4.5  BY  4.625  INCHES.
BACK WILL BE  4.875  BY  4.625  INCHES.
FRONT WILL BE  5.625  BY  5.75  INCHES.
   (MAKE A 38 IN. RABBET ALL AROUND FRONT!)
    (TAKE AN EXTRA 1/8 IN. RABBET AT TOP!)
BOTTOM WILL BE  4.25  BY  4.375  INCHES.
```



B. Amidon



B. Amidon

Desk gives work space hides wire, cable tangle

One of the nice things about TRS-80 is that it is based on a keyboard unit, which can be conveniently placed on or near the user's lap.

In the history of keyboards, this has seldom been the case. In music, organs pianos and the like have had keys mechanically connected to strings or wind chests. Electrification of the organ still left the keyboard connected via heavy cables of hundreds of wires. With office machinery, keys were mechanically connected to type bars and other mechanisms.

So keyboards have been arbitrarily placed in positions that are convenient to mechanisms, and mechanichal designers, and operators have had to cope. Thus hands frequently have to be held rather high in the air, at considerable discomfort, even on electronic equipment where the keyboard is connected to the rest of the device by only a few wires, as in the Pet, most computer terminals, etc.

Although Radio Shack designed its computer so that it could be operated on or near a person's lap, the company apparently did so unintentionally, because when it designed a desk for TRS-80 (model 1 and 2), it put the keyboard **on top of it.**

That is a natural-enough error. In first grade, Miss Holtzer made us keep our hands on top of the desk, our pencils..., etc.

And Radio Shack sells—and thinks—equipment. It is not into **using** a computer. It is into **selling** them.

We are into **using** computers, and although we indeed need places to keep them, we also need desk space for our work clutter: printouts, books, manuals and magazines, paperwork containing data to be input. **We need the whole desk top for these goodies!**

The CIE desk was designed to place the keyboard on our laps, where our hands are most comfortable and least fatigued, and to provide the whole desk top for paper clutter, to meet operator needs. Even the monitor was raised off the desk, so paper could spread under it.

Cabeling was hidden under the desk. Six recepticles were provided for peripherals, in addition to an **Electronic Specialists'** isolator, with three recepticles. These are all on the back, and an additional duplex recepticle is mounted on the side, for easy operator access. The desk plugs into a switched recepticle on the wall. One switch turns everything on or off at once. Oh, I know, that is a no-no. Peripherals can generate voltage spikes that can zap sensitive computer parts. But our computers and expansion interfaces are protected by the isolator!

We have built four of these desks, and used them about nine months. They're pretty well designed, redesigned, debugged and thought-out. And we like them. We have only used them with disk and Radio Shack cassette tape—not the Meca tape shown in the picture. That drive was only put on that shelf to serve as a prop. It's connector cable would not reach the expansion interface, and it could not be used as shown. Soon, when we start using four Meca drives for storing mail addresses, we will have to make added provisions for them. This will probably be in the form of a stepped shelf similar to what was suggested by Bill Roach of Elliam Associates. This would have to be on the left side of the desk, convenient to the expansion interface bus connector.

Use the best 2x2'' stock you can get at reasonable cost. Some lumber yards have only very rough and extremely expensive 2x2s. Avoid both these extremes. Shop around and find some smooth, straight 2x2s at moderate cost.

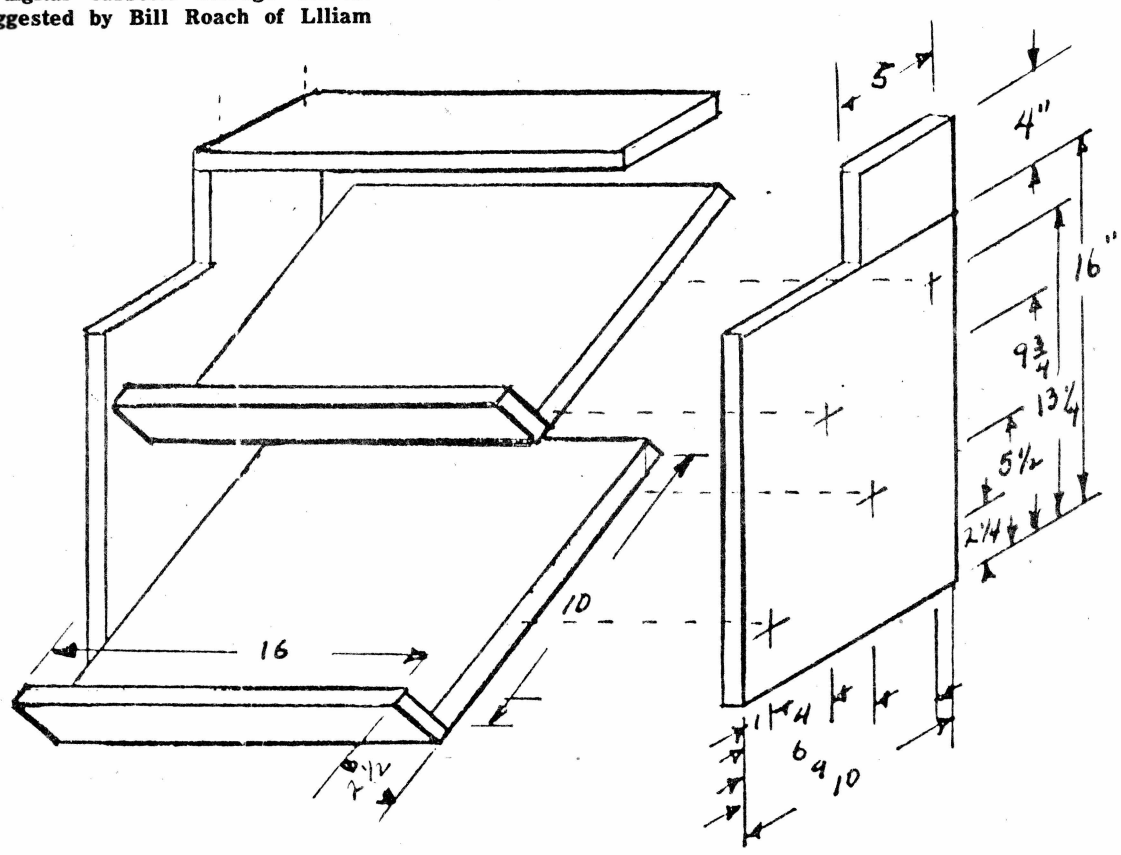
Don't skimp on top thickness. The desks we made with 5/8'' particle-board tops have sagged very badly, and we plan to re-top them. The 3/4'' top has held up well.

Similarly, don't skimp on top surfacing. First material we used was a vinyl fabric. This fake leather is hard to write on, and is not durable. Next we used Formica. Cost was not significantly greater. It probably was no more difficult to apply, and it really looks great! The ''butcher block'' pattern impresses everyone. This stuff is now available in thinner than normal stock, which applies with a hot iron. We plan to use that material to refinish the vinyl tops.

If the back of your desk is going to be visible, hang a fake back in front of the power cords. It could be hung simply with hooks and eyes, or more carefully, with hinge, and could be made of good-looking-yet-inexpensive plywood paneling.

Happy computing, on a desk that gives you twice as much work space as the not-inexpensive Tandy desk.

STEPPED SHELVES would be useful for mounting multiple Meca digital cassette storage drives. Design was suggested by Bill Roach of Lliam Associates.



Left Sliding Shelf

B. Amidon

Right Sliding Shelf

B. Amidon

B. Amidon

S-80 Computing is published by Computer Information Exchange, Inc., a nonprofit educational corporation, Box 158, San Luis Rey CA 92068.

Subscription rates in the U.S. are $15 for 12 issues of **Computing**, plus the monthly issues of **S-80 Bulletin**. To Canada and Mexico, subscriptions are $18US for 12 issues; all other countries $27US. Subscriptions outside the United States do **not** include the **Bulletin**, which circulates only in U.S.

Advertising is accepted, on the basis of a $200 magazine-size-page (7x10") rate.

In keeping with **S-80 Computing's** nonprofit educational aims, any amount of material from each issue may be reproduced by any not-for-profit educational or mutual-benefit group or institution, without prior request. Tearsheet is appreciated, as is credit: **S-80 Computing, box 158, San Luis Rey CA 92068; $15 for 12 issues.**"

Clubs having regular newsletters may receive **S-80 Computing** in exchange, providing CIE is also granted similar reprint privileges. Any number of **S-80 Bulletins** will be sent, as desired.

Organizations or anyone' having audio tapes of speakers whose talks might be of interest to CIE readers should submit them to the editor. If accepted, they will be typeset free, with type returned with the tape. Office phone is (714) 757-4849.

Editorial contributions of all kinds are requested. These include software (Basic, machine language, tiny Pascal, Fortran, Cobol, or any other language spoken by TRS-80), hardware and applications articles, letters, etc. and may be submitted either written or on machine-readable or spoken cassette. We are developing direct typesetting from cassette and disk, using Pencil, Scripsit, and Basic files.

computer information exchange, inc.

Box 158 San Luis Rey CA 92068

Nonprofit Org.
U.S. POSTAGE
PAID
San Luis Rey CA 92068
PERMIT 4

Dated material:
address correction requested