

K3 TI USERS GROUP
PO BOX 1941
KANKAKEE, IL 60901



USA
22



Fruited Dogwinkle

1 APR 86



TAKE A LOOK

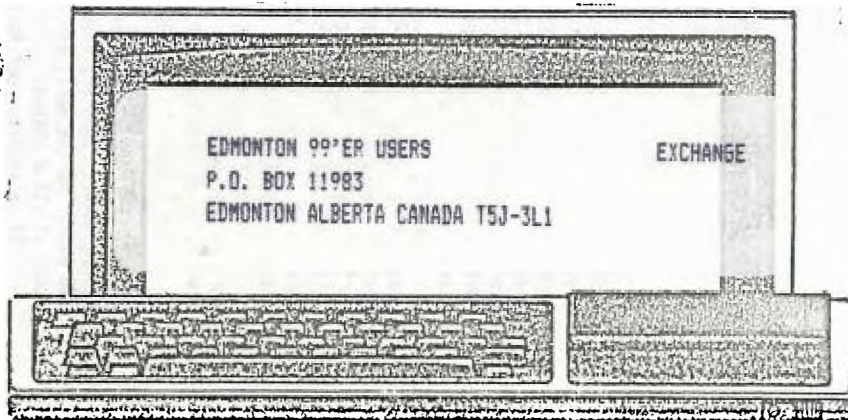
MEETING DATES

MARCH 15, 1986
APRIL 19, 1986

BOURBONNAIS
MUNICIPAL
CENTER

1 pm 'til 4 pm

K3 TI USERS GROUP NEWSLETTER



EDITOR'S PAGE
Beverly Cook, Editor
March/April 1986

Hello again. I hope the flu hasn't caught up with you. We've had quite a lot of it going around. Our Chairman wasn't feeling too good at our February meeting and ended up missing some work because of it. Schools have been closed because of it. If you missed it, you are one of the lucky ones. Its a real BUMMER! Let's hope the flu season is about over. I could'nt take another round of sick kids at home. I have 2 kids, (girl,12 and boy,10), and the only thing they share is their illnesses!

Down to business. Our March meeting will feature a demo of a Super Space Cartridge by DataBioTics. George Lempectis will give this demo. The Super Space Cartridge is sort of a poor man's GranCracker. This should prove to be quite interesting. They are not too difficult to make (as I understand). We will also try to have a quick demo of one of the Utility disks from the library.

The disk of the month for March will be FUNL-WRITER. For April, we will be giving away a disk full of games. As always, bring your own disk, please.

The library is selling some of its remaining modules. The are listed elsewhere in this newsletter. We will also be holding a raffle at the next 2 meetings. In March, we will raffle MINER 2049'er, a super game module. Chances will be .50 each. At the April meeting, we will raffle DONKEY KONG, again at .50 a chance.

If possible, our April meeting will feature a demo of PILQT by Rich De Roos. This is a programming language similar to LOGO. He's working on learning enough to dead it for us.

That's about all for this newsletter. Hope to see you at our meeting!

BEV

```
*****  
*                               *  
* Chairman                      Mark Harms *  
* Vice-Chairman                 Bruce Shearer *  
* Secretary                     George Lempectis *  
* Treasurer                     Glen Flowers *  
* Librarian                     Rich De Roos *  
* Newsletter Ed.                Beverly Cook *  
*                               *  
*****
```

K3 T1 USERS GROUP
BOARD MEETING MINUTES
FEBRUARY 23, 1986
by GEORGE LEMPEOTIS
SECRETARY

There were only four board members in attendance at the February board meeting. With only four members in attendance, all motions had to pass by unanimous vote. Four votes being a majority of seven board members.

Library sales for the January and February meeting were good, with good sales of blank disks and programs. The disk full of programs have been provided by the Chicago T1 99/4A Users Group.

I would like to thank the Chicago T1 99/4A Users Group especially their librarian John Behnke for providing some of their library programs for our club's library sales. The sale of these programs has kept the K3 T1 Users Group in the black and operating smoothly for the last four months. Again, Thank You Chicago T1 99/4A Users Group and John Behnke.

The board members decided to sell and raffel off the remaining library's game modules. The modules have not been renting over the past few months. Our librarian Richard DePoss will decide which modules to sell and their sale price, by the March meeting. Some of the modules will be raffled off at up coming meetings starting with the March main meeting. The board members also decided to purchase a MBX system for the library (at \$35.00 we could not pass it up), which will be offered for rent thru the library.

Unfortunately the board members could not get an unanimous vote to continue funding the BBS (the orphanage alias T1-K3) for the next two months. The newsletter will still be put out on a bimonthly basis.

The March 15, 1986 main meeting will feature a demo of a Super Space module by George Lempeotis. The Super Space module is made by DataBioTics of Palos Verdes Estates, California and is a extra 8K battery backed up Editor Assembler module that you can load with the programs of you choice. The free program of the month will be the fast loading FunlWriter V2.1, that gives you T1-Writer out of EX-Basic with Show Directory. The Miner 2049er module will be raffled at the meeting.

The April 19, 1986 main meeting will feature a demo of the Pilot program language by Richard DeRoos, and a demo of an utility programs from one of the library disk by George Leapeotis. The free program of the month will be a game disk loaded with programs. The Donkey-Kong module will be raffled at the meeting.

That is all we covered at the February board meeting. Hope to see you at the up coming meeting.



MEMBERSHIP RENEWALS DUE

The following people are due for renewal. Membership is \$7.50 per year. If your renewal is for 1/86 or 2/86, this will be the last newsletter you will receive. To receive the May/June newsletter, renewals must be received by April 15, 1986.

Don Duncan	2-86
Jeramie Messenbring	2-86
Duane Erwin	2-86
Brian Bolf	2-86
Fred Jacobazzi	2-86
Chris Kurtenback	3-86
Catherine Ross	3-86
Doug Sellers	3-86
Steve Bell	4-86
Paul Parkhill	4-86

NEWS

MUSICAL MOMENTS

100 REM RAINBOW	,B,587,A)
110 REM FROM THE CHICAGO TI USER'S GROUP LIBRARY	470 CALL SOUND(230,294,B,392 ,B,466,A)
120 REM BASIC OR EX-BASIC	480 CALL SOUND(230,294,B,440 ,B,523,A)
140 CALL SCREEN(2)	490 CALL SOUND(461,294,B,466 ,B,587,A)
150 CALL CLEAR	500 CALL SOUND(461,277,B,523 ,B,622,A)
160 FOR I=9 TO 14	510 CALL SOUND(922,208,B,262 ,B,311,A)
170 CALL COLOR(I,2,2)	520 CALL SOUND(461,156,B,392 ,B,523,A)
180 NEXT I	530 CALL SOUND(461,156,B,370 ,B,523,A)
190 CC=96	540 CALL SOUND(461,196,B,349 ,B,466,A)
200 FOR I=1 TO 24	550 CALL SOUND(461,196,B,311 ,B,466,A)
210 CALL HCHAR(I,1,CC,32)	560 CALL SOUND(461,175,B,294 ,B,466,A)
220 CC=CC+8	570 CALL SOUND(461,165,B,277 ,B,466,A)
230 IF CC<129 THEN 250	580 CALL SOUND(922,156,B,208 ,B,262,A)
240 CC=96	590 CALL SOUND(922,175,B,311 ,B,415,A)
250 NEXT I	600 CALL SOUND(461,233,B,311 ,B,392,A)
260 CALL COLOR(9,7,7)	610 CALL SOUND(230,233,B,262 ,B,311,A)
270 CALL COLOR(10,12,12)	620 CALL SOUND(230,233,B,294 ,B,349,A)
280 CALL COLOR(11,13,13)	630 CALL SOUND(461,233,B,330 ,B,392,A)
290 CALL COLOR(12,5,5)	640 CALL SOUND(461,233,B,277 ,B,415,A)
300 CALL COLOR(13,14,14)	650 CALL SOUND(461,220,B,294 ,B,349,A)
310 GOSUB 410	660 CALL SOUND(230,220,B,247
320 CALL SOUND(230,40000,30)	
330 GOSUB 410	
340 CALL SOUND(115,40000,30)	
350 GOSUB 720	
360 CALL SOUND(10,40000,30)	
370 GOSUB 410	
380 CALL SOUND(1000,40000,30)	
390 CALL CLEAR	
400 STOP	
410 A=0	
420 B=6	
430 C=9	
440 CALL SOUND(922,196,B,233 ,B,311,A)	
450 CALL SOUND(922,392,B,466 ,B,622,A)	
460 CALL SOUND(461,294,B,466	

,B,294,A)
670 CALL SOUND(230,220,B,262
,B,311,A)
680 CALL SOUND(461,208,B,294
,B,349,A)
690 CALL SOUND(461,208,B,294
,B,392,A)
700 CALL SOUND(1383,196,B,23
3,B,311,A)
710 RETURN
720 CALL SOUND(230,466,A)
730 CALL SOUND(230,156,C,311
,C,392,A)
740 CALL SOUND(230,156,C,311
,C,466,A)
750 CALL SOUND(230,156,C,233
,C,392,A)
760 CALL SOUND(230,156,C,233
,C,466,A)
770 CALL SOUND(230,156,C,262
,C,392,A)
780 CALL SOUND(230,156,C,262
,C,466,A)
790 CALL SOUND(230,156,C,233
,C,392,A)
800 CALL SOUND(230,156,C,233
,C,466,A)
810 CALL SOUND(230,262,C,311
,C,415,A)
820 CALL SOUND(230,262,C,311
,C,466,A)
830 CALL SOUND(230,262,C,311
,C,415,A)
840 CALL SOUND(230,262,C,311
,C,466,A)
850 CALL SOUND(230,233,C,294
,C,415,A)
860 CALL SOUND(230,233,C,294
,C,466,A)
870 CALL SOUND(230,233,C,294

,C,415,A)
880 CALL SOUND(230,233,C,294
,C,466,A)
890 CALL SOUND(922,156,4,392
,4,523,A)
900 CALL SOUND(1383,311,4,39
2,4,523,A)
910 CALL SOUND(115,40000,30)
920 CALL SOUND(230,466,A)
930 CALL SOUND(230,156,C,311
,C,392,A)
940 CALL SOUND(230,156,C,311
,C,466,A)
950 CALL SOUND(230,156,C,233
,C,392,A)
960 CALL SOUND(230,156,C,233
,C,466,A)
970 CALL SOUND(230,156,C,262
,C,392,A)
980 CALL SOUND(230,156,C,262
,C,466,A)
990 CALL SOUND(230,156,C,233
,C,392,A)
1000 CALL SOUND(230,156,C,23
3,C,466,A)
1010 CALL SOUND(230,262,C,31
1,C,440,A)
1020 CALL SOUND(230,262,C,31
1,C,523,A)
1030 CALL SOUND(230,262,C,31
1,C,440,A)
1040 CALL SOUND(230,262,C,31
1,C,523,A)
1050 CALL SOUND(230,185,C,31
1,C,440,A)
1060 CALL SOUND(230,185,C,31
1,C,523,A)
1070 CALL SOUND(230,185,C,31
1,C,440,A)
1080 CALL SOUND(230,185,C,31

1,C,523,A)
1090 CALL SOUND(922,349,4,46
6,4,587,A)
1100 CALL SOUND(922,311,4,37
0,4,587,A)

1110 CALL SOUND(922,349,4,52
3,4,698,A)
1120 CALL SOUND(922,294,4,37
0,4,523,A)
1130 RETURN

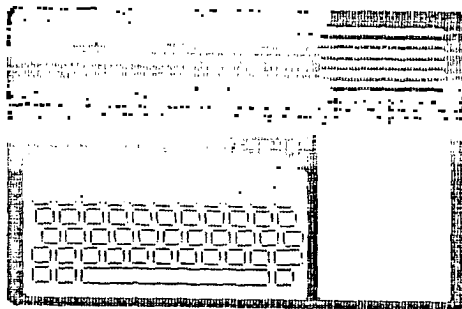
LIBRARY NEWS

The Chicago TI User's Group continues to provide us with quality programs. Come on in and have a look at the wide variety we have for sale.

We are selling the following modules. These are on a first come, first served basis as there is only one of each.

Pac Man \$4.00
Defender \$4.00
Centipede \$4.00
Tunnels of Doom \$4.00
Othello \$4.00
Burgertime \$3.00
Connect Four \$3.00
Henhouse \$3.00

THE TI-99/4A COMPUTER



CAR RALLY

```

50 REM CAR RALLY
60 REM FROM THE CHICAGO TI U
SER'S GROUP
100 RANDOMIZE
110 CALL CLEAR
120 PRINT TAB(10);"CAR RALLY
": : :TAB(7);"CHOICE OF CARS
": : : "MINI", "#1": "LOTUS", "#2
": : : "TRANS-AM", "#3": "FERRARI",
"#4":
130 PRINT : "ENTER NUMBER OF
YOUR CHOICE:"; "THE BETTER TH
E CAR THE MORE"; "GAS IT USES
140 INPUT "WHICH CAR? ":C1
150 C1=INT(C1)
160 IF C1>4 THEN 140
170 IF C1<1 THEN 140
180 CALL CLEAR
190 PRINT "CHOOSE THE COURSE
TO RACE.": : "1 IS EASIEST A
ND STRAIGHTEST"; "5 IS THE HA
RDEST AND MOSTLY"
200 PRINT "TURNS AND TWISTS.
": : :
210 PRINT "WHICH COURSE DO Y
OU WANT?": : : : : : : : :
220 INPUT "COURSE (1-5) ":C2
230 C2=INT(C2)
240 IF C2<1 THEN 220
250 IF C2>5 THEN 220
260 CALL CLEAR
270 PRINT "YOU WILL NEED TO
TRAVEL 5": "MILES WITH .5 GAL
LONS OF GAS": : "YOUR STATUS
WILL BE SHOWN"
280 PRINT "EVERY 10 SECONDS.
": : "YOU WILL BE ASKED FOR A
NEW": "RATE OF GAS.": : " 10
IS HARD ACCELERATION."
290 PRINT "-10 IS HARD BRAKI
NG.": : "ANY NUMBER BETWEEN 1
5": "ACCEPTABLE.": :
300 FOR I=1 TO C1
310 READ B,M,S
320 NEXT I
330 A1=.5
340 M1=0
350 C1=C1/2
360 V=0
370 R1=0
380 T=0
390 D=0
400 B1=0
410 PRINT "SEC =";T;"SPEED =
";INT(V00)/1000;"GAS =";I
NT(A10)/100;"MILES =";INT
(M1)/10; :
420 IF M1>=5 THEN 1040
430 INPUT "RATE OF GAS ":G
440 IF G<-10 THEN 430
450 IF G>10 THEN 430
460 IF G<9 THEN 520
470 Z=Z+1
480 IF Z<4 THEN 520
490 PRINT : "DUMMY!!": "YOU BL
EW YOUR ENGINE!!": :
500 Z=0
510 GOTO 910
520 V=INT(B*G-M*V+V)
530 T=T+10
540 PRINT : "STATUS ";
550 IF V>0 THEN 570
560 V=INT(RND*9)+1
570 M1=M1+V/460
580 IF G<0 THEN 620
590 A1=A1-(G*5)/5000
600 IF M1>=5 THEN 1040
610 IF INT(A10)/100<0 THE
N 1000
620 IF R1=1 THEN 760

```



```

630 IF Q1=1 THEN 700
640 Q=INT((C2+1)*RND)
650 R=INT((3.75-C2)*RND)
660 IF R>0 THEN 930
670 IF Q>0 THEN 970
680 PRINT "CLEAR STRAIGHT"
: :
690 GOTO 410
700 H=INT(35*RND)+15
710 M=H+5*C1
720 IF V>H THEN 1060
730 PRINT "THRU CURVE": :
740 Q1=0
750 GOTO 410
760 E=E-(V-D)*3
770 IF E<0 THEN 800
780 PRINT "VEHICLE";E;"FT AH
EAD": :
790 GOTO 410
800 IF V-D<5 THEN 860
810 PRINT "VEHICLE PASSED";
820 D=V-D
830 PRINT D;"MPH": :
840 R1=0
850 GOTO 410
860 PRINT "VEHICLE BEING PAS
SED": :
870 D=INT(40*RND)+25
880 PRINT "GRAYHOUND BUS IN
OTHER LANE.":"DOING";D;"MPH"
:
890 D=V+D
900 PRINT "CRASH VELOCITY ="
;D: :
910 PRINT "FANS WILL SEND YO
U FLOWERS!": :
920 GOTO 1100
930 PRINT "VEHICLE AHEAD 100

```

```

0 FT": :
740 D=INT((35*RND)+25
950 R1=1
960 GOTO 410
970 PRINT "CURVE AHEAD!": :
980 Q1=1
990 GOTO 410
1000 PRINT "EXCELLENT":"BUT
YOU RAN OUT OF GAS!!": "YOU
'RE A LEAD FOOTED MANIAC.":
:
1010 GOTO 1100
1020 PRINT "DON'T KNOW HOW,
BUT YOU MADE":"IT.": :
1030 GOTO 410
1040 PRINT "FINISH LINE": : "
YOU'RE LUCKY THIS YEAR!!": :
1050 GOTO 1100
1060 PRINT "ROAD IS TERRIBLE
": :
1070 H=H-5*C1
1080 PRINT H;"POSTED SPEED F
OR CURVE":V;"WAS YOUR SPEED.
": :
1090 GOTO 910
1100 PRINT "ANOTHER TRY? (PR
ESS Y OR N)"
1110 CALL KEY(0,V,S)
1120 IF S=0 THEN 1110
1130 IF V=78 THEN 1190
1140 IF V<>89 THEN 1110
1150 RESTORE
1160 I=0
1170 GOTO 110
1180 DATA 4.5,.53,10,6,.5,13
,7,.41,15,8,.39,18
1190 END

```



CALL LOADS FOR EXT BASIC

1. CALL INIT :: CALL
LOAD(-31931,0)--USED TO BREAK PROGRAM
PROTECTION ON TI/DOS YOU NEED 32K
CARD, AND X-BASIC OR E/A.

2. CALL INIT :: CALL
LOAD(-31806,16)--DISABLES THE QUIT
KEY

3. CALL INIT :: CALL
LOAD(-31806,0)--RE-ENABLES THE QUIT
KEY

4. CALL PEEK(2,A,B):: CALL
LOAD(-31804,A,B)--TO CAUSE A PROGRAM
TO RETURN TO MAIN TITLE
SCREEN

5. CALL LOAD(-31962,255)--TO RESTART
XBASIC, i.e. CHECK FOR LOAD PROGRAM,
ETC.

6. CALL INIT :: CALL
LOAD(-31888,63,255)...THEN...NEW--FREES
UP MEMORY ALLOCATED TO DISK DRIVES.
ANY CALL TO DRIVES WILL FREEZE UP
COMPUTER.

7. CALL INIT :: CALL
LOAD(-32630,128)--RETURNS YOU TO
TITLE SCREEN WITHOUT GRAPHICS. YOU
CAN USE COMMANDS AS NORMAL TO PICK
SCREEN CHOICES YOU JUST CAN'T SEE
THEM.

8. CALL INIT :: CALL
LOAD(-31961,149)..might have to
follow with END--WORKS MUCH LIKE #7
BUT GOES TO RESET POSITION. THEN
SEARCHES FOR A PROGRAM CALLED LOAD.
IF FOUND IT WILL LOAD AND RUN THAT
PROGRAM. IF NOT FOUND, WILL GO TO
XBASIC MODE.

9. CALL LOAD(-31961,51)::
END--RESETS TO TITLE SCREEN WITH
GRAPHICS.CAN BE USED FOR PROGRAM
PROTECTION FROM KIDS.

10. CALL LOAD(-32572,1)--PRODUCES A
"MUSHY" KEYBOARD, WITH IMPROPER
CHARACTERS PRINTED FROM THE ONES
BEING TYPED.

11. CALL LOAD(-32572,128)--COMPLETELY
DISABLES KEYBOARD

12. CALL LOAD(-31878,X)--WHERE X IS
THE HIGHEST NUMBER OF SPRITES YOU ARE
USING IN A PROGRAM.COMPUTER TRIES TO
UPDATE ALL 32 SPRITES WITH THE OLDER
VERSION OF X BASIC.THE WILL PERMIT IT
TO CHECK ONLY THE NUMBER YOU HAVE IN
THE PROGRAM. SUPPOSED TO BE FASTER
PROGRAM RESPONSE????

13. CALL LOAD(-31745,0)-- PRODUCES A
FROZEN SCREEN, THEN AFTER A FEW
SECONDS BLANKS ENTIRELY. RESTORE TO
MASTER SCREEN BY PRESSING FCNT -.

14. CALL LOAD(-31748,N)--CHANGES
RATE OF CURSOR FLASH

=====

SPECIAL

=====

REPORT

=====

TECH NOTES

This information is from John Clulow & Randy Gries of the New Horizons Users group. It originally appeared in R/D Computing newsletter by Ryte Data.

This article will show you how to add 8k of RAM to the E/A module! Questions may be directed to Ron at (419) 874-1414. Later a circuit to install battery backup will be printed. This is especially useful for E/A programming. NO RESPONSIBILITY CAN BE TAKEN FOR ANY LOSS OR DAMAGE YOU INFLICT. ATTEMPT THIS AT YOUR OWN RISK.

Parts: Hitachi COS HM6264P-15 (\$34.95) or LP-15 version (\$39.95), if you want battery backup later on.

These chips are available from JDR Micro-devices 1224 S. Bascom Ave. San Jose CA 95128 (800) 538-5000

You also need a TI game module which is foiled on both sides. Munchman was used for this one. (purchased cheap)

You will need an Edit/Assem module to transplant the E/A GROM chip. A 'spare' is a good idea, in case you destroy the chip! Buy a 1k resistor (R.S.# 271-023) and some wire wrap (#278-501). Vacuum solder remover - rosin core solder and a GROUNDED soldering iron (pencil type) are also needed. If you don't have experience handling CMOS devices and removing solder ASK FOR HELP.

First unscrew the game module shell & pull the case apart from the slotted side. Remove the PC board while holding the sliding door down. Note the spring is 'under' the PC board.

Next: Unsolder and remove the GROM and RAM chips. They are located as shown. The ROM chip is the larger of the two.

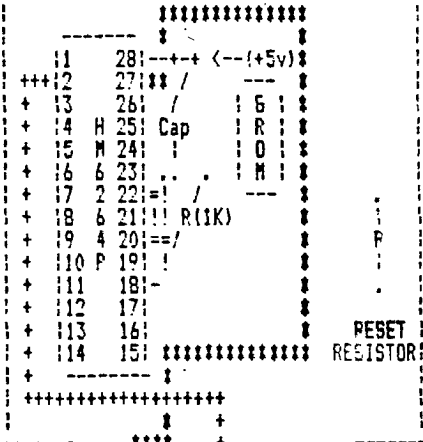


FIG 3

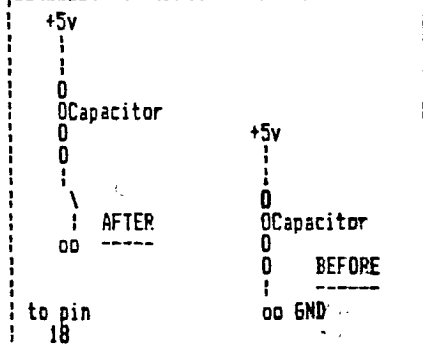
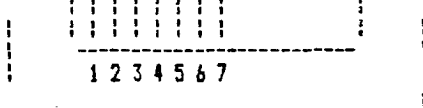


FIG 2

NOTE FIG 3
RAM pin 2 is wired to #7 edge line. 1k resistor is wired to pin 20 and 22. Line from capacitor to pin 18 does NOT connect to resistor. Ram pin 27 is wired to #3 edge line.

Look underside the board where the 8k ROM chip is located. Use the vacuum device to remove most of the solder. Gently pry up on one end of the chip while heating pins at the same time. (HINT: a rubber band set up to pull on the chip for you can help) A capacitor should be located next to pins 21-24 of the ROM. Desolder the ground end from its soldering pad leaving the +5v end (nearest the back of the board (away from the edge pins) attached. With a knife carefully break the foil between the two adjacent soldering pads where the capacitor WAS connected as shown in Figure 2:

Now resolder the ground end of the capacitor to the pad on the RIGHT. Solder one end of a short piece of wire to the pad on the LEFT (where the capacitor used to be) and the other end to hole 18 of the removed ROM. See figure 1 below for ROM pin numbering:

FIG 1

NC	11	28	VCC	
A12	12	8K RAM	27	WE
A7	13	26	CS2	
A6	14	25	A8	
A5	15	24	A5	
A4	16	23	A11	
A3	17	22	OE	
A2	18	21	A10	
A1	19	20	CS1	
A0	110	19	1/08	
1/01	111	18	1/07	
1/02	112	17	1/06	
1/03	113	16	1/05	
GND	114	15	1/04	

This will be the seventh hole from the BACK of the board on the side CLOSEST to the capacitor. When a command module is inserted, it normally resets the computer. If you

want to disable this auto-reset in your E/A module remove the resistor at the opposite end of the board. See Fig 3 above.

Figure 1 gives a pin diagram of the HM6264 RAM.

In handling the CMOS RAM chip take EXTRA precautions to eliminate static electricity. Don't work on a carpet. Touch a ground before handling the device. Handle it by the plastic body. Touch the pins as little as possible and work on a grounded surface, if at all possible. When soldering, hold the pencil iron on the pins for the LEAST amount of time required to make the connection -- not more than 1 to 2 seconds. If you clean the pins first, it helps. Remove the RAM from its anti-static tube. Place the device on its side on a table (hard flat surface) and move the body of the device to very gently bend the pins to closer to a right angle with respect to the body. Do both rows of pins. Check to make sure that the pins line up with the holes on the PC board. Orient the chip as in figure 1 and bend RAM pins 1, 2, 20, 27 & 28 straight out. This allows the chip to fit the old holes. Insert the RAM into the PC board so that the notched end is flush with the back of the board. RAM pin 3 goes into ROM hole 1 - RAM pin 26 into ROM hole 24 ect. ect.

With the RAM in place solder in one pin on each side to hold it. Connect a wire from RAM pin 27 (now bent straight out) to the Write Enable pin on the edge connector. It is the third from the left looking at the top of the board (see Fig 3) and it is NOT connected by foil to the PC board. Connect a wire from RAM pin 2 (also bent) to address line 12 on the edge card connector 7th pin from the left. This edge pin also doesn't have a foil connection to the board.

Solder a short wire from RAM pin 20 (bent out) to RAM pin 22. It will be relatively easy to solder one end of the wire to RAM pin 20 but RAM pin 22 is in a hole and a little more difficult to get at. Solder the wire as close to the board as possible using as little solder as feasible. Solder one lead of the 1k resistor to the soldering pad just above and to the right of the ground end of the capacitor. (Refer to Fig 3) The resistor lead can be pushed through the hole. Solder the other end of the resistor lead to RAM pin 20 (bent out). Solder a short wire from the soldering pad to the right of the +5v end of the capacitor to RAM pin 28 (bent out).

Now all that remains is to install the E/A GROM. Open the E/A module and remove the PC board. Install the E/A GROM in the holes left by the old GROM. Notched end goes towards the back of the board as well.

Finally solder all the IC pins in their respective pads for both the RAM and GROM. Place the spring in the BOTTOM of the E/A module case. Locate the sliding door properly. Put the new PC board in place and snap the case closed. Replace screw and you're done.

The first thing to do is make sure your E/A GROM still works OK. Then you can test out your RAM with the following program:

```
100 INPUT 'NUMBER 0-255? ':X
110 CALL LOAD(24576)
120 CALL PEEK(24576)
130 PRINT 'MEMORY HAS ':X
140 PRINT
150 GOTO 100
```

When you enter a number from 0 to 255 you should see the same number displayed on the screen, having been stored by 110 and reread by line 120. If the number the computer returns is different from the one you entered, the device is not working properly. Remove it and retrace all steps until you find the problem. I would check the soldering job first. The address 24576 is >6000. Your new RAM goes from >6000 to >7FFF or in decimal from 24576 to 32767. You may want to check out several addresses in this range to make sure they are all working right.

There are a number of things you can use the new RAM for. In assembly language programs you can use an AORG >6000 directive to have the loader place your object code in the new RAM. Alternatively, you can change the First Free Address in High Memory (FFAH) to >6000 with a CALL LOAD(8223) then load a program w/ CALL LOAD ('DSK1.NAME') as usual. If you plan to load other programs, you can change the FFAH to >A000 by CALL LOAD(8228). I have been using the new 8k of RAM to hold the DEBUG program when working on assembly language programs.



RANBLING IN FORTH
by TED ANDERSEN

Here are some FORTH routines which some of you may find helpful.

Printer code words on screen 92 make it easy to configure the printer in the normal FORTH command string: i.e. PRCMPR SWCH 0 180 INDEX UNSWCH produces a condensed disk index which fits in the disk sleeve neatly. PERMIT will send and number and combination of control characters needed for special occasions which may not be covered by the more common words on screen 92. Notice screens 91 and 92 were originally 72 and 73.

The disk I/O provided by TRIAX on screen 90 allows export of FORTH screens to DISVAR80 files mergeable with TI-WRITER or E-A editor files (uploadable to the BBS). As it stands it directs output to the Foundation 128k disk emulator, DSKX, but is easily redirected to floppy per the comment. I found this exercise very educational. My first attempt at adapting SWCH to redirect normal LIST, INDEX, etc output to a diskformat file was unsuccessful. SWCH, I found out, is limited to a single character output buffer which is okay for printers but produces an entire 80 byte record for each character when tried with disk.

Screen 91 here is just the PID conversion of 72. I include it to provoke a simple exercise for the FORTH class. Let's make a smarter INDEX function: one that identifies blank and binary loaded sectors for better and easier file management. I am thinking of listing just the first in a string of blank or binary sectors to avoid unnecessary output length. Let's see how neatly this can be done.

```
SCR #90
0 ( DISK FILE I/O AND 128K ENDATH DSKY 31MAR85 TSA) BASE->R
1 HEX 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
2 PABS @ 20 + BUFR 1700 FILE DISK ( FILE ATTRIBUTES TO CLEN )
3 : FILEVAR80 DISK SET-PAB SONTL .FE. DSPY UPDT 50 REC-LEN ;
4 : MEMINIT DISVAR80 F-D" MEMINIT" DLT ( F-D" DSKX.FORT" OPN ) ;
5
6 : TRIAX ( FROM TO --- ) DISVAR80 F-D" DSKX.FORT" OPN 1+ SWAP
7 DD 1 3 + 1 DO BUFR !" SCR # " <# I 0 #S #> BUFR 5 + SWAP
8 CMOVE 9 WRT CR BUFR 9 TYPE BUFR 50 BLANKS
9 I BLOCK 10 0 DD DUP (<# I 0 #S #> BUFR SWAP CMOVE 1 .
10 BUFR 3 + 40 CMOVE 44 WRT 40 + LOOP DROP 0D BUFR C!
11 1 WRT 1 WRT LOOP BUFR !" TI FORTH --- a fig-FORTH extension"
12 22 WRT 0C BUFR C! 0D BUFR !+ C! 2 WRT 3 +LOOP CLSE ;
13
14 ( CHANGE DSKY TO DSK2 etc FOR FLOPPY, MEMINIT UNIQUE TO 128K )
15 R->BASE
```

SCR #01

```

0 ( ALTERNATE I/O SUPPORT FOR PIO PWTP 12JUL82 LCT 10MAP84 TSA)
1 0 CLOAD INDEX      BASE->R DECIMAL 68 R->BASE CLOAD STAT
2 0 0 0 FILE >PASCII BASE->R HEX
3 : SWCH >RS232 PABS @ 10 + DUP PAB-ADDR ! 1- PAB-VBUF !
4 SET-PAB OUTPUT F-D" PIO.EC" ( RS232.BA=9600" )      OPN 3
5 PAB-ADDR @ VSWB 1 PAB-ADDR @ 5 + VSWB PAB-ADDR @ ALTOUT ! ;
6 : UNSWCH 0 ALTOUT ! CLSE ;
7 : PASCII ( BLOCK# --- FLAG )
8     BLOCK 0 SWAP DUP 400 + SWAP
9     DO I C0 20 > + I C0 DUP 20 < SWAP 7F > OR
10     IF DROP 0 LEAVE ENDF LOOP ;
11 : TRIAD 0 SWAP SWCH 3 / 3 + DUP 3 + SWAP
12 DO I PASCII IF 1+ I LIST CR ENDF LOOP
13 -DUP IF 3 SWAP - 14 * 0 DO CR LOOP
14 OF MESSAGE OC EMIT ENDF UNSWCH ;
15 R->BASE -->

```

SCR #02

```

0 ( SMART TRIADS AND INDEX 15SP82 LAD 18MP85 TA) BASE->R DECIMAL
1 : TRIADS ( FROM TO --- )
2 3 / 3 * 1 + SWAP 3 / 3 * DO I TRIAD 3 +LOOP ;
3 : INDEX ( FROM TO --- ) 1+ SWAP
4 DO I DUP PASCII IF CR 4 .R 2 SPACES I BLOCK 64 TYPE ELSE DROP
5 ENDF PASCII IF LEAVE ENDF LOOP ;
6 : PREMIT ( 0 CPM...COD1 --- ) ( EMIT PRNTR N CODES TIL 0 )
7 SWCH BEGIN DUP EMIT 0= UNTIL CR UNSWCH ;
8 : PRESET ( --- ) 0 64 27 PREMIT ; ( RESET TO 10 CPI, 6 LPI )
9 : PREMR ( --- ) 0 48 27 15 PREMIT ; ( 17 CPI, 8 LPI )
10 : PROBLM ( --- ) 0 14 PREMIT ; ( DOUBLE WIDE )
11 : PRITAL ( --- ) 0 52 27 PREMIT ; ( ITALICS )
12 : PRUNDL ( --- ) 0 1 45 27 PREMIT ; ( UNDERLINE )
13 : PRMEND ( --- ) 73 4 - 16 * DUP 15 + SWAP 6 + DO I MESSAGE CR
14 LOOP CR ;
15 R->BASE

```

```

DEF COLOR
:
: SET EQUATES FOR X BASIC
WMB EQU >2054
VMB EQU >2055
C DATA >F4F4,>F4F4,>F4F4,>F4F4 : SET COLORS TO WHITE ON BLUE
COLOR LI R0,20714 : SET THE SPACE CHARACTER
BLWP @VMBTR : EVERY SCREEN POSITION WITH
LI R0,2048
LI R1,C
LI R2,B
BLWP @VMBW
LI R0,2056 : ASSEMBLY LANGUAGE
BLWP @VMBW : SCREEN TO CHANGE
LI R0,2054 : THE TEXT AND SCREEN
BLWP @VMBW : COLORS IN X-BASIC
LI R0,2072 : FOR PROGRAMMING
BLWP @VMBW
B @R11 : BY LARRY BENTLEY
ADR0 >83CA : 9/29/84
DATA COLOR
ENC

```

ADJUSTING A TI ACOUSTIC MODEM

by George Leapeotis

I have had a slight problem with my TI Acoustic Modem since I bought it a year ago. I would get unexpectedly disconnected while calling a Bulletin Board, this would happen periodically and for no apparent reason.

About a month ago the problem got worse, now I could not get connected for more a than few minutes without getting disconnected. After talking to the board's Sysop and recalling a magazine letter about a similar problem, I thought the problem might be in the signal strength adjustment inside the modem.

I took apart the modem and found three adjustment pods, after some experimenting I found the pod under the speaker cup to be the signal strength adjustment. All I had to do to reach the pod was pry off the speaker cup (the cup that takes the part of the phone receiver that you talk into). With the cup off, I was able to reach the pod with a small screwdriver. The pod is located off to the center of the modem, turning the pod counterclock-wise seems to increase the signal. I turned the pod all the way to the left (increased the signal), and the modem worked fine. In a month of use I have not been unintentionally disconnected from a BBS.

I think the signal adjustment is there to compensate for noisy or weak phone lines (like mine), the worse the line the stronger the signal needed. On a cleaner and stronger phone line too high a signal could cause a feedback problem, the better the line the lower the signal needed. I would suggest experimenting with the adjustment, untill you find the setting that works best and corrects any problems you might have.

Happy Board Hunting.

