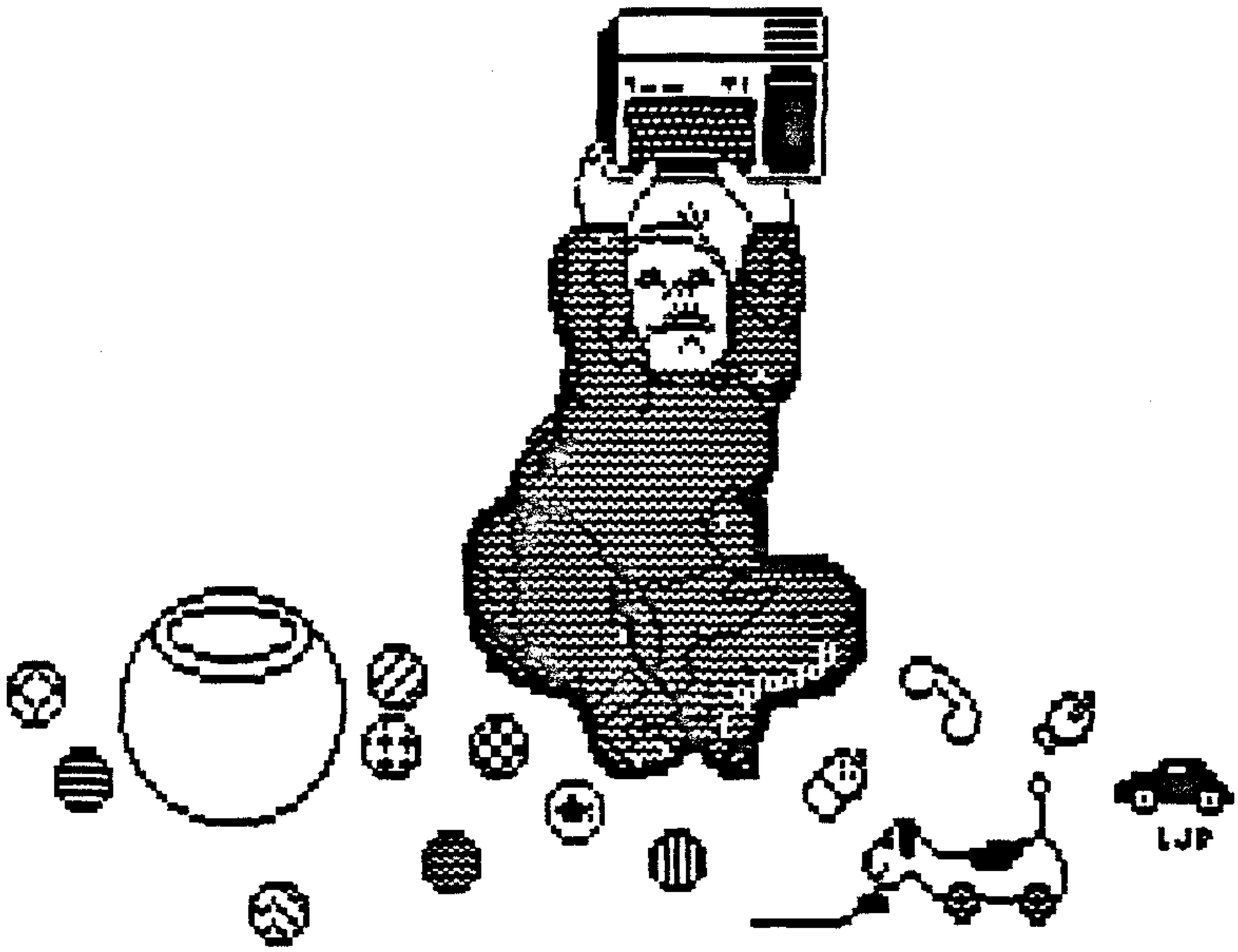
Texas Instrument 99/4A and Myarc 9640 Computers

Spirit of 99

THE OFFICIAL NEWSLETTER OF THE CENTRAL OHIO NINETY-NINERS INC.

PUBLISHED MONTHLY IN COLUMBUS OHIO

Blake loves his T.I.



$1.50 VOL. 10 NO. 11 NOV 1992

11/14/92

Editorial, address is:
181 HEISCHMAN AVE
WORTHINGION,OH 43085
Subscription rate (USA) $20.00 /1 year $25.00 /1 year outside continental U.S. Third class postage paid at Columbus, OH
CHANGE OF ADDRESS: Send both OLD and NEW address to address listed above. We assume no responsibility for manuscripts,programs (tape or disk) not accompanied by return postage. Letters to the Editor are property of Spirit of 99. If published,we reserve the right to edit at our discretion.

Central Ohio Ninety Niners Inc. is a non-profit organization comprised of MEMBERS who own or use the TI99/4A computer and it's related pro-ducts and have paid a yearly membership fee of $30 and whose main objectine is the exchange of Educational and Scientific information for the purpose of computer literacy.
C.O.N.N.I. meetings are held the 3rd sat-urday of each month at Chemical Abstract, 2540 Olentangy River Road Columbus, OH. Meeting time is 8:30 AM til 2:30PM, Meetings are open to the public. Membership dues ($20.00) are payable yearly to C.O.N.N.I. and cover the immediate family of the member. Please send check to our membership registerar and join C.O.N.N.I.
Please address it to:
John L. Parkins
2215 Bayfield Drive
Columbus, OH 43229

** INDEX **

C.O.N.N.I. MINUTES

C.O.N.N.I. MINUTES
Wednesday, October 28, 1992.

Once again the group was very small, but we made up in enthusiasm what we lacked in numbers.

President Parkins opened the meeting at 8:20p.m. First came a discussion of a printer problem being experienced by one of the members. The Star NX-1000, which had been working perfectly, suddenly would not advance the paper in either direction, although it sounded as if it were trying to do so. Harley Ryan agreed to try to locate the problem.

A variety of topics were discussed next: universal remote controls for TV, VCR, etc., including one that will work with satellite TV; hoped-for revisions to M-Dos for the Geneve; Mike Wright's emulator for T.I. on the PC; dealing with drive problems and too many files to fit on a disk; certain software producers that built in incompatibilities so that they would run only if other peripherals were purchased; various copiers, such as Mike Ballman's BACKUP, TRACK1, and COPYCAT; TI-Sort's apparent inability to sort truly in numerical order; the fact that many exchange newsletters no longer contain original articles; the reduced size of Micropendium, probably due to a loss of advertising; and the recent absence in Micro Center's catalog of the user-group listings, their schedules, and Ohio BBS listings. Apparently they have had staff cutbacks and cannot continue this service. Reportedly, Eagen Foster and the Columbus Computer Society have agreed to produce and keep updated these very popular sections in the catalog.

Also mentioned was an apparent "TI virus" (and we thought we were free of all that!). The virus mentioned erases files off disks, including the first and second sectors, then prints to the screen the message "I gotcha!". The feeling was that Bud Mills knew of a possible solution, so hopefully he will tell us all so we can avoid being "gotten".

John Parkins promised to write an article about this year's Chicago Faire, as he and Chuck Grimes were the only members to attend. At the November Saturday meeting, Jim Peterson will demonstrate a 40-column DV80 file reader, good for very long files, and assorted other goodies. At a not-too-distant future meeting, Harley Ryan will demonstrate the Harrison Word Processing program. Either Dave Truesdale or Bob DeVilbiss will be contacted and asked to demonstrate 4A-Flyer, which was recently released to the public domain.

Respectfully submitted,

Dick Beery, Co-Secretary

C.O.N.N.I. CALENDAR

November 1992

SUN	MON	TUE	WED	THU	FRI	SAT
1	2	3	4	5	6	7
8	9	10		12	13	14
15	16	17	18	19	20	21
22	23	24	CONNI MEETING		27	28
29	30					

SATURDAY MEETING 21 NOV 1992
Janis Center -- Columbus

8:30AM Setup, coffee, and doughnuts

9:00AM Disk of Month,
MICROpendium,
Beginners help,
Libraries open

9:30AM Question and
Answer Period

10:30AM Business
Meeting

11:15AM Demos:

40 Col DV80 File Reader
By Jim Peterson

1:30PM Tear down
and go home

WEDNESDAY MEETING -- 22 NOV 1992
McDONALD'S -- Cleveland and Main -- Westerville

7:30PM MEETING TIME

DUES ANNOUNCEMENT

Local dues are usually paid at or before the March meeting, and are $20 per year for full membership, library and voting privileges, plus the newsletter. You may also pay your dues in two installments if desired: $10 in March and $10 in Septeaber. Those who join during other months of the year pay a lesser, pro-rated amount:

MAR-20.00 APR-18.33 MAY-16.67 JUN-15.00 JUL-13.33 AUG-11.67 SEP-10.00 OCT-8.33 NOV-6.67 DEC-5.00 JAN-3.33 FEB-1.67

A DEAL
YOU CAN'T
PASS UP!
SUBSCRIBE TO BOTH!

Now you can have the best of both worlds-- Keep up to date on the latest news from the TI-99/4A world with a subscription to the Spirit of 99 Newsletter AND get an up-to-date collection of new public domain and shareware programs with the Disk of the Month--Both brought to you by the Central Ohio Ninety-Niners, Inc.-- No newsletter published in August.-- January newsletter is an index of all articles published during the previous year.-- 10-SSSD "flippy" DOM's published annually.-- At times, two diskettes depending on the availability of new material.--the NL is mailed 1st of the month-- DOM is mailed about the middle of the month.

SUBCRIPTION RATES

Annual membership including newsletter:
$20 (U.S.A.)
$25 (outside U.S.A.)
DOM $35 (U.S.A)
$40 (outside U.S.A.)

John Parkins, Membership
Central Ohio Ninety-Niners, Inc
2215 Bayfield Dr, Columbus, OH 43229
(614) 891-4965

Please enter my subscription for:
__ Newsletter only $20 (U.S.A.), $25 (outside U.S.A.)
__ D.O.M. $35 (U.S.A.), $40 (outside U.S.A.)
__ Newsletter and D.O.M. $55 (U.S.A.), $65 (outside U.S.A.)

Name _____

Address _____

State, Zip or Country _____

Type of disk drive:

___ SSSD

___ DSSD

___ DSDD

Make check payable to C.O.N.N.I.

CLEARING HOUSE

What:a means of sharing text files between clubs and to cut down on newsletter costs.
Who:Any T.I. users group (or individual) may participate.
Cost:$30 the first year; $15 each succeeding year.
Mail check to CONNI membership registrar (see page 3).
Free trial:For those who want to see what the service offers, call:
Spirit of '99 BBS
(614)263-3412 24 hrs.
8N1 300-1200-2400 baud.
(direct access or through Starlink or PC-Pursuit).

MIDI MASTER 99 and CASIO MT-240

by Jim Peterson

In my opinion, Midi Master 99 is one of the most interesting accessories ever developed for the TI-99/4A. It is very reasonably priced and, unlike many hardware developments, it offers no compatibility difficulties.

There are only two problems - obtaining it, and finding a low-priced MIDI-compatible keyboard to use it with.

Of all the TI suppliers with a poor reputation for filling orders, Crystal Software seems to have been the worst. Perhaps that has now changed, but the surest way to obtain the product would be to catch Mike Maksimik at a computer fair and walk away from his table with it firmly clutched in your hand.

Midi Master 99 was developed using the Casio MT-240 keyboard, which sold for about $80, and I was lucky enough to be able to find one for that price. Unfortunately, it is no longer on the market. The only MIDI-compatible keyboards in the 1992 Casio catalog are the CT-700 at $399, the CT-670 at $499 and the CT-770 at $599.

A local music store told me that Yamaha keyboards with the MIDI interface started at about $200, but I do not know the model numbers. A few people have been able to find them in discount stores for about $190, but those stores usually only stock them for the Christmas sales. The music stores only carry the professional keyboards in the $400 - $600 dollar range; they would probably order a cheaper model for you, but would certainly charge you full manufacturer's suggested retail price or more.

Many people are waiting to buy Midi Master 99, or to write any music for it, until Version 3 is released. I learned long ago not to hold my breath while waiting for a new version of any TI product.

Version 3 is supposed to allow you to play music on the keyboard, which will be converted into a MIDI file that the computer can play back, through MIDI, on the keyboard. Since I can only play a keyboard with even fewer fingers than the three I use for typing, that doesn't interest me.

Come to think of it, if you can play the keyboard, why would you want to convert your music to a MIDI file? Why not just tape it to a cassette, if you want to save it?

To me, the great thing about Midi Master 99 is that it allows me to create music even though I cannot play an instrument - just as I used to do in Extended Basic, using the three tone generators of the TI-99/4A. Also, it allows me to do things that no musician could do from the keyboard, such as playing two or more instruments simultaneously, or playing chords that no human hand could reach, or creating musical effects that would require two very nimble-fingered musicians.

Midi Master 99 consists of a cable, to connect your RS232 card to the keyboard, and a disk containing the necessary software, the documentation, and some sample music files. The documentation is adequate. It contains a good deal of technical material that is way over my head, but which is not necessary in order to use the program.

Music files are created by keying in an SNF file, from sheet music, using TI-Writer or Funnelweb or Editor/Assembler. If you use TI-Writer or Funnelweb, select the open cursor mode or else save the file by PF with the C option, because carriage returns will result in an error message.

If you have an elementary knowledge of reading music, keying in a selection is quite simple, although it does take time. The only thing I had to learn is that octaves start from C, not from A. The lowest note available, in octave 0, is the C which is 3 notes above Hertz 110 A, the lowest note available from the TI tone generators (other than the noise generator). This means that you may have to fudge on some notes in the bass clef.

You can key in all voices simultaneously or separately. That is, you can key in a melody note and its harmony notes, and then go on to the next, or you can key in the entire melody, and then the entire first note of the harmony, etc. Dolores Wertha of Harrison Software, who knows more about this than I ever will, recommends the second method, but I am stubbornly sticking to the first way.

One serious flaw is the lack of looping - a directive to repeat the melody over again as many times as you wish, which is so easily done in XBasic programming. You can only use the Copy function of Funnelweb to copy the fi. after itself, which doubles the tim required to load and compile it before playing. However, I understand that looping in this case is far more difficult than it would seem, and has only recently been implemented for MIDI or the PC.

According to the documentation, existing TI Basic music can be easily converted to the MIDI SNF format. In actual practice, it depends on how the music was originally programmed. That had best be the subject of another article.

The completed file can be saved in DV80 format, in which case it is loaded and compiled each time it is played, or in compiled image format which will load and play directly. The trouble is that the image file is stored in a very wasteful PC-style format of three 33-sector files. I have not done any comparative timing, but it seems that the additional loading time wipes out the time saved by not compiling - unless, of course, you have the file on a ramdisk or hard drive. Also, image files cannot be modified.

As a bonus for waiting so long for my Midi Master 99, Maksimik sent me a free copy of his Midi Album program. This requires the Mini Memory module or other device to provide extra memory, as Midi Master 99 itself uses all that is available. It will catalog a disk, allow you to select the files you want to play randomly or in sequence, and load and play them. It works very well. I did find that you must be sure to specify duration and instrumentation in the SNF file, if it is to be played through Midi Album; otherwise, it will carry through the values from the previous selection rather than using the program defaults.

For some reason, the documentation on my Midi Album disk was a DV254 file rather than a DV80 file, so it could not be printed with Funnelweb!

Different models of keyboards have different instruments available, and different numbers assigned to these instruments. Maksimik has provided a patch program, so that you can use your keyboard to play music written for a different keyboard. On the copy of Midi Master 99 he sent me, he had patched the percussion instrumentation into a couple of the other voices, which caused me great puzzlement for awhile.

If the music is in SNF format, it is

NEXT PAGE

probably more practical to just edit the file. I do hope that those who write MIDI music will include remarks in the SNF file, or separately with image files, to indicate what keyboard they programmed for and what instruments they assigned.

Regarding the Casio MT-240, it is a budget model which lacks some desirable features. For one thing, it does not allow MIDI to control the volume. It perhaps uses the same tone generators as a larger model because I found several instruments, numbered 21 through 29, beyond the 20 on the panel. There are also some additional percussion effects in the octave above the keyboard range.

I have found several problems which may be the fault of the keyboard, of Midi Master 99, or of Midi in general. Without having other keyboards to try out, I cannot tell.

Some instruments such as bells, are not practical to use because they continue to reverberate and create a dissonance. Others, such as chorus, drag out until they seem to affect the rhythm. Some, such as organ, are almost silent in the lowest octave, probably because they also sound in an octave lower. Some instruments sound harsh when programmed in all three voices although not when played from the keyboard. I have found it difficult to find pleasing combinations of two or more instruments. The best effects are generally obtained by giving all voices the default instrumentation of piano, and most existing TI MIDI music has been written for that instrument.

Dolores Werthe, the renowned music programmer of Harrison Software, is trying to organize a by-mail users group for those making music with Midi Master. If you are interested, write to her at 5705 40th Place, Hyattsville MD 20781.

****

MDOS BUYOUT COMPLETED

From MICROpendium

The September issue of MICROpendium reports that the MDOS buyout was sucessfully completed by Beery Miller who now retains exclusive rights to the source code for MDOS, ABASIC, and PSYSTEM for the Myarc Geneve 9640.

Paul Charlton also included the GPL Interpreter source code. Those people who made a financial contribution for the MDOS buyout to Beery will have the source code made available to them. According to Miller, the source code will not be abvailable on any network or BBs, and should not be distribited to anyone without Miller's direct authorization.

According to Miller's contract with Lou Phillips of Myarc, Miller will send final copies of MDOS directly to the owners. Miller is presently awaiting names and addresses of the owners from Phillips. Phillips required Miller to handle the estimated 2,200 Geveves and the final mailing of MDOS software in lieu of receiving immediate cash for the buyout.

Miller notes that he had anticipated mailing for 800 Geneves and is still short of money to handle the final mailing. He says contributions will be welcome, and persons mas still have the option of acquiring source code diskettes with a minimum $25 contribution.

Miller notes that suggestions for enhancements to MDOS will only be accepted from contributors; bug reports will, however, be accepted from anyone. He says current enhancements have included a three-fold speed increase for any floppy controller access and a two-fold increase in hard drive speed.

A hard and Floppy Disk Controller and hard drive are required to assemble the source code to these systems, as well as ownership of GenPROG by Paul Charlton, Miller is negotiating a contract with Charlton on the redistribution of this package.

Packages are availadbl in 5.25 inch DSSD (180K) format or 3.5 inch DSQD (720K) format.

Prices are MDOS/GPL, $10 for four 5.25 disks or $5 for one 3.5 disk; ABASIC/GPL, $7.50 for three 5.25 disks or $5 for one 3.5 disks; and PSYSTEM, $5 for two 5.25 disks or one 3.5 disk.

Write to Beery Miller at P.O. Box 752465, Memphis, TN 38175. END

****

MYARC PARTS

From MICROpendium

Don Walden of Cencure Electronics received a package of gate arrays for the Myarc Hard and Floppy Disk Controller and Geneve 9640 and some 9938 video chips from Lou Phillips Aug. 24.

"There was no note or anything," Walden says., "just the package."

Also, Walden notes, he recently received a letter from a user in Austria who had been referred to him by Somerset County, New Jersey, Consumer Affairs Division. The agency has recently investigated a number of complaints against Myarc.

The Consumer Affairs Division advised the Austrian customer that Phillips had told them that Walden would be supplying proprietary parts.

Walden noted that he still has not directly heard from Phillips himself for several months.

For repair or parts for Myarc products, write Cecur Electronics, 7759 So. Scepter Dr. 7, Franklin, WI 53132-2201 or call (414) 529-2173. END

TIPS FROM THE TIGERCUB

No. 70

Tigercub Software
156 Collingwood Ave.
Columbus, OH 43213

****

My three Nuts & Bolts disks, each containing 100 or more subprograms, have been reduced to $5.00 each. I am out of printed documentation so it will be supplied on disk.

My TI-PD library now has over 600 disks of fairware (by author's permission only) and public domain, all arranged by category and as full as possible, provided with loaders by full program name rather than filename, Basic programs converted to XBasic, etc. The price is just $1.50 per disk(!), post paid if at least eight are ordered. TI-PD catalog #6 is available for $1 which is deductible from the first order.

I still like to program "brain games". Here is one of the most devilish of all.

```
100 DISPLAY AT(2,3)ERASE ALL
:"THE FORE AND AFT PUZZLE":"
":" Try to get the numbers i
n  the lower half and the
   letters in the upper half
."
110 DISPLAY AT(8,1):" You ca
n move horizontally  or vert
ically to the vacant square
or jump over one     space t
o the vacant square,"
120 DISPLAY AT(12,1):"but nu
mbers can only move   right
and down, letters can only m
ove left and up!" !programme
d by Jim Peterson
130 DISPLAY AT(16,1):" Type
the number or letter  to mov
e or FCTN 8 to start  over o
r FCTN 7 for a demo."
140 DISPLAY AT(20,1):" It ca
n be done in 46 moves but pr
obably not in more    than 4
6 because you will getstuck.
"
150 DISPLAY AT(24,8):"PRESS
ANY KEY" :: DISPLAY AT(24,8)
:"press any key" :: CALL KEY
(0,K,S):: IF S=0 THEN 150
160 CALL CLEAR :: CALL COLOR
(0,16,16,3,16,5,4,16,5,5,16,
7,6,16,7,9,2,2,12,16,16):: C
ALL SCREEN(2)
170 A$=RPT$("~",9):: GOSUB 3
30 :: GOSUB 340 :: V$="12345
67BABCDEFGH"&CHR$(1)&CHR$(6)
180 CALL CALLKEY(24,1,V$,C$)
:: V=ASC(C$)<65 :: IF C$=CHR
$(6)THEN GOSUB 330 :: GOSUB
340 :: GOTO 180
190 IF C$<>CHR$(1)THEN GOSUB
 220 :: GOTO 180
200 GOSUB 330 :: GOSUB 340 :
: FOR W=1 TO 46 :: C$=SEG$("
A87AC63CF687FBE635F21ABEH87D
354BED21CD6546H21H",W,1):: V
=ASC(C$)<65 :: GOSUB 220 ::
NEXT W
210 FOR D=1 TO 500 :: NEXT D
 :: GOSUB 330 :: GOSUB 340 :
: GOTO 180
220 FOR J=3 TO 7 :: P=POS(M$
(J),C$,1):: IF P=0 THEN 230
ELSE X=J :: J=7 :: GOTO 240
230 NEXT J
240 IF V=-1 THEN 260 :: T=X-
1 :: GOSUB 290 :: IF F=1 THE
N F=0 :: RETURN ELSE T=X-2 :
: GOSUB 290 :: IF F=1 THEN F
=0 :: RETURN
250 T=P-1 :: GOSUB 310 :: IF
 F=1 THEN F=0 :: RETURN ELSE
 T=P-2 :: GOSUB 310 :: IF F=
1 THEN F=0 :: RETURN ELSE 28
0
260 T=X+1 :: GOSUB 290 :: IF
 F=1 THEN F=0 :: RETURN ELSE
 T=X+2 :: GOSUB 290 :: IF F=
1 THEN F=0 :: RETURN
270 T=P+1 :: GOSUB 310 :: IF
 F=1 THEN F=0 :: RETURN ELSE
 T=P+2 :: GOSUB 310 :: IF F=
1 THEN F=0 :: RETURN
280 CALL SOUND(500,110,0,-4,
0):: RETURN
290 IF SEG$(M$(T),P,1)<>"c"
THEN RETURN
300 M$(T)=SEG$(M$(T),1,P-1)&
C$&SEG$(M$(T),P+1,255):: M$(
X)=SEG$(M$(X),1,P-1)&"c"&SEG
$(M$(X),P+1,255):: GOSUB 340
 :: F=1 :: RETURN
310 IF SEG$(M$(X),T,1)<>"c"
THEN RETURN
320 M$(X)=SEG$(M$(X),1,T-1)&
C$&SEG$(M$(X),T+1,255):: M$(
X)=SEG$(M$(X),1,P-1)&"c"&SEG
$(M$(X),P+1,255):: GOSUB 340
 :: F=1 :: RETURN
330 M$(1),M$(2),M$(8),M$(9)=
A$ :: M$(3)="~~123~~~~~" :: M
$(4)="~~456~~~~~" :: M$(5)="~
~78cA8~~" :: M$(6)="~~~~CDE~
~" :: M$(7)="~~~~FGH~~" :: R
ETURN
340 FOR J=8 TO 16 :: DISPLAY
 AT(J,10):M$(J-7):: NEXT J :
: RETURN
350 SUB CALLKEY(R,C,V$,K$)
360 CALL HCHAR(R,C+2,30):: F
OR T=1 TO 3 :: CALL KEY(0,K,
S):: IF S<>0 THEN 390
370 NEXT T :: CALL HCHAR(R,C
+2,20):: FOR T=1 TO 3 :: CAL
L KEY(0,K,S):: IF S<>0 THEN
390
380 NEXT T :: GOTO 360
390 IF POS(V$,CHR$(K),1)=0 T
HEN 360 ELSE K$=CHR$(K)
400 SUBEND
```

I don't think this is very useful, but somebody asked me for it - it converts decimals to fractions.

```
100 CALL CLEAR :: CALL CHAR(
95,"000000FF")
110 DISPLAY AT(12,1):"Decima
l?" :: ACCEPT AT(12,10):D ::
 T=1
120 IF INT(D)<>D THEN D=D*10
 :: T=T*10 :: DISPLAY AT(14,
1):D :: DISPLAY AT(16,1):T :
: GOTO 120
130 DISPLAY AT(14,1):D :: DI
SPLAY AT(15,2):RPT$("_",LEN(
STR$(T))):: DISPLAY AT(16,1)
:T
140 FOR J=2 TO 5 STEP 3
150 IF D/J=INT(D/J)AND T/J=I
NT(T/J)THEN D=D/J :: T=T/J :
: DISPLAY AT(14,1):D :: DISP
LAY AT(16,1):T :: GOTO 150
160 NEXT J :: GOTO 110
```

Several years ago, John Hamilton wrote a program you could use to key in a program with TI-Writer, then merge it in, delete the "!" after each line number, and run it as a program. Its only problem was with lines of over 80 characters. Since then, better programs have been written - XLATE and TEXTLOADER - which do not require deleting anything but they still have some trouble with long lines and with missing spaces. This little version overcomes those faults but you do have to delete the "!".

Try keying in a program into the Funlweb Editor, be sure to put a carriage return at the end of each program line. When finished, check each program line which has wrapped around to two lines. If the first character in that second line should be preceded by a space, insert a space as its first character. Then save the file with the PF option and run this little program. Enter NEW, merge in the output file by MERGE DSKn.filename, go through it with FCTN X and FCTN 1 deleting the "!" after each line number, and it should run as a program.

```
100 DISPLAY AT(12,1)ERASE AL
L:"Input file? DSK":"":"Outp
ut file? DSK"
110 ACCEPT AT(12,16):A$ :: A
CCEPT AT(14,17):B$
120 OPEN #1:"DSK"&A$,INPUT :
: OPEN #2:"DSK"&B$,VARIABLE
163,OUTPUT
130 LINPUT #1:M$
140 IF POS(M$,CHR$(13),1)=0
THEN LINPUT #1:M2$ :: M$=M$&
M2$ :: GOTO 140 ELSE M$=SEG$
(M$,1,LEN(M$)-1)
150 X=POS(M$," ",1):: Y=VAL(
SEG$(M$,1,X-1))
160 PRINT #2:CHR$(INT(Y/256)
)&CHR$(Y-256*INT(Y/256))&"!"
&SEG$(M$,X+1,255)&CHR$(0)
170 IF EOF(1)<>1 THEN 130 EL
SE CLOSE #1 :: PRINT #2:CHR$
(255)&CHR$(255):: CLOSE #2
```

I had a question from a friend who wanted to key in some pieces of information in Funnelweb and then sort

NEXT PAGE

them. Trouble was, the data tended to be more than 80 characters long. Therefore it was saved as two or more separate records, which a sort scrambled into garbage.

So, how do you create and sort long records of varying length? The easiest way is to let the disk drive controller do it for you. Just type whatever you want, as long as you want, then save it as a separate file, using the first several letters of the text as the filename. Don't include any spaces or periods, of course. If you are using numbers as filenames, pad them with leading zeros to all the same length such as 001 to 999 or 0001 to 1000.

The drive controller will sort those files alphabetically, and this little program will print them in that sequence -

```
100 CALL CLEAR :: DIM F$(127
):: OPEN #1:"DSK1.",INPUT ,R
ELATIVE,INTERNAL :: INPUT #1
:D$,A,B,C
110 INPUT #1:M$,A,B,C :: IF
A=2 AND C=80 THEN X=X+1 :: F
$(X)=M$
120 IF LEN(M$)<>0 THEN 110 E
LSE CLOSE #1 :: OPEN #2:"PIO
"
130 FOR J=1 TO X :: OPEN #1:
"DSK1."&F$(J),INPUT
140 LINPUT #1:M$ :: IF ASC(M
$)<127 THEN PRINT #2:M$
150 IF EOF(1)<>1 THEN 140 EL
SE CLOSE #1
160 NEXT J :: STOP
```

This method is limited by the fact that you can only put 127 files on a disk, but if you have more than one drive you can have 127 on each one, and use this program -

```
100 DISPLAY AT(12,1)ERASE AL
L:"How many drives?" :: ACCE
PT AT(12,18)SIZE(1)VALIDATE(
NUMERIC):D :: DIM F$(510)
110 FOR J=1 TO D :: OPEN #1:
"DSK"&STR$(J)&".",INPUT ,REL
ATIVE,INTERNAL :: INPUT #1:D
$,A,B,C
120 INPUT #1:M$,A,B,C :: IF
A=2 AND C=80 THEN X=X+1 :: F
$(X)=M$&"¦"&STR$(J)
130 IF LEN(M$)<>0 THEN 120
140 CLOSE #1 :: NEXT J :: CA
LL LONGSHELL(X,F$()):: OPEN
#2:"PIO"
150 FOR J=1 TO X :: W=POS(F$
(J),"¦",1)
160 OPEN #1:"DSK"&SEG$(F$(J)
,W+1,1)&"."&SEG$(F$(J),1,W-1
)
170 LINPUT #1:M$ :: IF ASC(M
$)<127 THEN PRINT #2:M$
180 IF EOF(1)<>1 THEN 170
190 PRINT #2:"" :: CLOSE #1
:: NEXT J
200 SUB LONGSHELL(N,N$())
210 D=N
220 D=INT(D/3)+1 :: FOR I=1
TO N-D :: IF N$(I)<=N$(I+D)T
HEN 250 :: T$=N$(I+D):: J=I
230 N$(J+D)=N$(J):: J=J-D ::
 IF J<1 THEN 240 :: IF T$<N$
(J)THEN 230
240 N$(J+D)=T$
250 NEXT I
260 IF D>1 THEN 220
270 SUBEND
```

A recent article in a news letter reminded me of something I knew long ago but had forgotten. If you have been entering a lot of data into a disk file and the program crashes, all is not lost. Just enter CLOSE #1 in command mode and your data will be saved. If you get a FILE ERROR message, just try CLOSE #2 and so on until you hit the right one.

Many user group newsletter editors use a program that puts a code on the address label to indicate when membership expires. Trouble is, no one ever reads their address label!

This quick & dirty little program requires you to prepare your address file in TI-Writer or Funnelweb with name on first line, address on second, city and state on third, the fourth line blank or you can use it for additional address, number of expiration month on fifth line and year on sixth. Continue with other addresses, making sure you use six lines for each. Such a file is easy to update with TI-Writer. The program will read addresses from that file and print an address label for everyone whose membership has not expired. It will also optionally print a warning label, which you can slap conspicuously on the front page of the newsletter, if the subscription currently expires or expires next month. If you give a grace period for renewal, you can choose to print an address label and a warning label for those who are one month or two months overdue.

```
100 DISPLAY AT(1,4)ERASE ALL
:"REMINDER LABEL PRINTER"
110 DISPLAY AT(3,1):"Address
 file? DSK" :: ACCEPT AT(3,1
8):F$ :: OPEN #1:"DSK"&F$,IN
PUT
120 DISPLAY AT(5,1):"Printer
? PIO" :: ACCEPT AT(5,10)SIZ
E(-20):P$ :: OPEN #2:P$
130 DISPLAY AT(6,1):"Emphasi
zed print? (Y/N)" :: ACCEPT
AT(6,25)VALIDATE("YN")SIZE(1
):E$ :: IF E$="Y" THEN PRINT
 #2:CHR$(27)&"E";
140 DISPLAY AT(7,1):"Doubles
truck print? (Y/N)" :: ACCEP
T AT(7,27)VALIDATE("YN")SIZE
(1):D$ :: IF D$="Y" THEN PRI
NT #2:CHR$(27)&"6";
150 DISPLAY AT(9,1):"Print p
ending expiration    notice?
 (Y/N)" :: ACCEPT AT(10,15)S
IZE(1)VALIDATE("YN"):PEND$
160 DISPLAY AT(11,1):"Print
current expiration    notice
? (Y/N)" :: ACCEPT AT(12,15)
SIZE(1)VALIDATE("YN"):CUR$
170 DISPLAY AT(13,1):"Print
past    expiration    notice
? (Y/N)" :: ACCEPT AT(14,15)
SIZE(1)VALIDATE("YN"):PAST$
180 DISPLAY AT(15,1):"Print
two months past       expira
tion notice? (Y/N)" :: ACCEP
T AT(16,26)SIZE(1)VALIDATE("
YN"):PAST2$
190 DISPLAY AT(18,1):"Curren
t year?" :: ACCEPT AT(18,15)
:Y :: Y=Y+(Y>99)*1900 :: Y=Y
-92
200 DISPLAY AT(20,1):"Number
 of month?" :: ACCEPT AT(20,
18)VALIDATE(DIGIT):M :: X=M+
Y*12
210 IF EOF(1)=1 THEN 330 ::
LINPUT #1:A$ :: IF ASC(A$)=1
28 THEN 330
220 LINPUT #1:B$ :: LINPUT #
1:C$ :: LINPUT #1:D$ :: INPU
T #1:M,Y :: Y=Y+(Y>99)*1900
:: Y=Y-92 :: M=M+Y*12
230 IF M>=X THEN GOSUB 280
240 IF M=X AND CUR$="Y" THEN
 GOSUB 290 :: GOTO 210
250 IF M=X+1 AND PEND$="Y" T
HEN GOSUB 300 :: GOTO 210
260 IF M=X-1 AND PAST$="Y" T
HEN GOSUB 280 :: GOSUB 310 :
: GOTO 210
270 IF M=X-2 AND PAST2$="Y"
THEN GOSUB 280 :: GOSUB 320
:: GOTO 210 ELSE GOTO 210
280 PRINT #2:A$:B$:C$:D$:"":
"" :: RETURN
290 PRINT #2:A$:"YOUR SUBSCR
IPTION EXPIRES THIS":"MONTH.
 PLEASE RENEW NOW SO YOU":"W
ILL NOT MISS ANY ISSUES":" "
:" " :: RETURN
300 PRINT #2:A$:"YOUR SUBSCR
IPTION EXPIRES NEXT":"MONTH.
 PLEASE RENEW NOW SO YOU":"W
ILL NOT MISS ANY ISSUES":" *
:" * :: RETURN
310 PRINT #2:A$:"YOUR SUBSCR
IPTION EXPIRED LAST":"MONTH.
 PLEASE RENEW NOW SO YOU":"W
ILL NOT MISS ANY ISSUES":" "
:" " :: RETURN
320 PRINT #2:A$:"YOUR SUBSCR
IPTION EXPIRED":"TWO MONTHS
AGO":"THIS WILL BE YOUR LAST
 ISSUE":"UNLESS YOU RENEW PR
OMPTLY":" " :: RETURN
330 CLOSE #1 :: END
```

Memory just about full -

Jim Peterson

The above picture was taken at the C.O.N.N.I.'s 10th anniversary celebration on 17 October 1992 at the Janis Center in Columbus. Front row L-R Wesley Kincaid, Bud Mills, Everett Wade, Harley Ryan, Carole Parkins, Ken Marshall, Bud Wright and Dick Clark. Back Row L-R Kevin Noesner, Fred Deaner, Chuck Grimes, Ken Kuehnle, John Parkins, Dave Truesdale, Harold Timmons, John Broughman, Dick Beery, William Brandon, Irwin Hott, Jim Peterson, and Bill Sheppard. Others present were George Seibert, Sabrina Marshall, Larry Fairbanks, Jim Klein, Tom Roberts, Jack Montag and Eagan Foster. Picture courteny of Jack Montag.

TENTH ANNIVERSARY MEETING
By Dick Beery

The C.O.N.N.I. tenth anniversary meeting was held with gala and festivities on Saturday, October 17, 1992 at the Martin Janis Center, our former "home". Owing to scheduling difficulties with Chemical Abstracts, both the October and November 1992 meetings will be at the Janis center.

In addition to our regular attendees, we were honored by the presence of former president Ken Kuehnle, former newsletter editor Wes Kincaid, former graphics expert Bill Brandon and former members Fred Deaner and Tom Roberts. It was a special privilege to have present the president of the prestigious Columbus Computer Society, a group of devotees of the "other" computer. Mr. Foster expressed surprise and delight at the continued vigor of the T.I., the reported recent advances in both hardware and software, and the loyalty of the users of this "abandoned" computer.

A specially-decorated cake, mints, nuts and hot beverages of varying descriptions including coffee, were served to the guests by Carol Parkins, our "first lady". She was assisted in providing the refreshments and in setting up by Lorita Beery, wife of Disk Librarian and BBS co-sysop Dick Beery.

At the entrance, President John Parkins had provided a slideshow of many pictures that date back to the early days, some of which this author had never before seen. Nearby, Larry Fairbanks provided a convincing and well-received demonstration of the capabilites of the 4A in accessing a laser-jet printer.

Across the room in front of the picture windows, Bud Wright demonstrated on the Geneve his program that displays digitized pictures of group members and identifies each one in both speech and writing. The resolutional clarity of the graphics and the sound identification were extremely impressive.

In a classroom on the premises, Chuck Grimes had set up a demonstration of music for Midi and also the For-TI (Forth) quadriphonic music system. His display table offered a panorama of history with many early devices including :the T.I. impact printer; acoustic modems (300 baud); Super-Sketch drawing device; MBX controller and cartridges for the Milton Bradley voice-directed games; For-TI the Forth quadriphonic music system released by T.I. shortly before the pullout; MBP clock card; P-code card; A-D(analog to digital converter) card;and finally, the hard-floppy disk controller and hard drives.

Dave Truesdale presented a variety of games, many of them greatly appreciated by Sabrina Marshall, daughter of Ken Marshall, Jr., who will be taking over the Disk of the Month in July 1993.

The most special guest and highlight of the festivities was Bud Mills, who provided handouts and details regarding his company's new SCSI interface; 4A memex, Rambo and several other advances in the hardware field.

Altogether, a very satisfying morning. It was especially gratifying to this author to have gathered together so many who form a part of C.O.N.N.I.'s history. It seems very special that we are still together, nine years after the bailout by Texas Instruments. Who knows what the next three to five years will bring. We hope for the best.

CONSOLE DEBUGGING HELP
by John Guion / [illegible]
Home Computer Group via
Hocus 99 Newsletter
Part 1 of 2

PROBLEM AREAS:
1) Console will not power up
2) Keyboard errors
3) Intermittent computer lockup
4) Module errors
5) Joystick port errors
6) Video output difficulty
7) Sound problems

The TI-99/4A Console and Peripheral Expansion System Technical Data manual available from Texas Instruments' Dealer Parts Department [(806) 741-2265] will serve as an excellent source for schematics and part location guide.

The information contained herein is only intended for use as a reference for possible debugging procedures. It is not intended as a repair guide for the common user with little or no knowledge of digital electronics or the basic structure of the TI-99/4A system. The author assumes no responsibility for damages resulting from improper use of this information.

1) CONSOLE WILL NOT POWER UP

1.1 General information

Failure of the TI-99/4A to power up and produce the TI title screen is a common problem that is also the hardest to track down and fix since failure of nearly any component in the console or power supply can cause this. The following are not intended as solutions to the problem, but merely as points to check that may aid in finding the actual problem and fixing it.
Unless a particular part is suspected, replace any socketed chips possible with known working equivalents before desoldering any components. Since the socketed chips are common causes of lock up, eliminating them as possible problems first may save excess soldering on the board. The console will power up if the sound chip is removed entirely, but not if that chip is shorted internally.
A simple TTL logic probe can be used for tracing signals in the circuit. An oscilloscope may also be used and has [illegible] clock signals for proper frequency. When a signal should exist as an output from a particular device, be sure to check that device's input for proper signals before attempting to replace the component. When checking for locked up signals, try to trace all signals back through the circuit to the point of origin. A set of schematics (available from several sources, including TI) will help greatly in this part of debugging. Tracing locked signals can determine whether or not the signal is missing due to a faulty component that it must pass through or what power up operation was occurring during lock up.

1.2 Console power up procedure.

A. TMS9900 CPU resets and addresses low ROM locations.
B. TMS9900 initializes
C. TMS9900 sets up workspace registers in MCM6B10 RAM.
D. TMS9900 begins GROM read
E. TMS9900 enters delay loop for about 1/4 second.
F. TMS9919 sound chip is disabled.
G. TMS9918A VDP chip is initialized.
H. 4116 VDP RAM is initialized (requires about 1 second).
I. Title screen is loaded into VDP.
J. TMS9919 sound chip emits beep.
K. TMS9900 CPU enters keyboard scan.
L. System is ready for use.

1.3 Voltage/signal checklist.

A. Check power supply for +5V, +12V, and -5V. Lack of -5V often results in a gray flickering screen on power up. Check for +5V on chips throughout the board. Check TMS9900 for -5V at pin 1; +5V at pins 2, 33, 59, and 64; and +12V at pin 27. If any voltages are missing, check for shorts on main board. Replace Check TMS9900 for -5V at pin 1; +5V at pins 2, 33, 59, and 64; and +12V at pin 27. If any voltages are missing, check for shorts on main board. Replace power supply if necessary.

B. Check TMS9900 pins 8, 9, 25 and 28 for clock signal. If not found, check TIM9904 clock generator pins 1, 2, 3 and 4 for clock signal. If not found, check TIM9904 supply voltages (+5V at pin 20, +12V at pin 13) crystal, and tank circuit. If no external problem can be

C. Check TMS9918A pin 39 and pin 40 for the 10.73863 MHz clock. If missing, check crystal and oscillator circuit. Otherwise, check TMS9918A pin 36 and pin 37 for clock outputs. If not found, remove GROMS and sound processor (located next to GROMs) and test again for clock. If missing, possible TMS9918A failure. Reinsert GROMs and sound processor after tests.

D. Check TMS9918A pins 14 (-CSW) and 15 (-CSR) for lock up. If locked up, check memory enable from pin 6 of 74LS32 and pin 13 of 74LS138 located next to MCM6810. Trace signal to find possible failure.

E. Check TMS9918A pin 13 (MODE) for lock up. If locked up, trace signal back to TMS9900. Also check for other components that may be locking up this line (it is used as A14). If no other fault can be found on that line, possible TMS9918A failure.

F. Check TMS9918A pin 1 (-TAS), pin 2 (-CAS) and pin 11 (-R/W) for lock up. If locked up, possible TMS9918A failure.

G. Check TMS9918A pins 17 through 24 (data lines) for signals. If missing, trace to fault. Possible TMS9918A or TMS9900 failure.

H. Check TMS9918A pins 3 through 10 (RAM address/data lines) for signals. If missing, possible TMS9918A failure.

I. Check 4116 RAM pin 14 (DATA OUT) on each chip for signal. Each chip missing signal may be at fault as well as TMS9918A.

J. Check TMS9900 pin 62 (READY) for lockup. If locked up, check TMS9900 pin 6 (-RESET) for signal. If pin 6 is locked up low, possible TIM9904 failure. If high, possible TMS9900 failure. If TMS9900 pin 6 is not locked up, trace circuit back from pin 62 to find fault.

K. Check all three GROMs (CD2155, CD2156, and CD2157) at pin 10 (-CS) and pin 15 (GREADY) for signals. If either is missing, remove all three GROMs and test pin 10 again for signal. If the

signal at pin 10 does not exist, trace back through circuit to find failure. If signal exist, replace GROMs one at a time until GROM that causes lock up on pin 15 is found.

L. Check all three GROMs for signal on pin 11 (MO/A14) and pin 12 (M1/DBIN). If missing, trace circuit to find break in signal path.

M. Check each GROM for -5V at pin 14, +5V at pin 9, and -.8V to -.6V at pin 16. If missing, check for broken trace. If -.8V/-.6V is missing or at -5V, check diode connected to that line.

N. Remove sound generator. If console powers up, check pin 16 for +5V, pin 4 for clock from TMS9918A, pin 5 (-WE) for signal, and pin 6 (-CS) for signal from 74LS138 closest to MCM6810. If these signals exist, possible sound chip failure.

O. Check TMS9918A pin 36 for composite video output. If missing, check TMS9918S crystal and clock circuit and *pin 16 (-INT) for interrupt signal.* If signal exist, possible TMS9918A failure.

P. Check GROMs for clock on pin 13. If missing, check clock output on TMS9918A pin 37. If signal on TMS9918A exist, check for break in signal path. If not, check TMS9918A oscillator circuit. If oscillator operates, possible TMS9918A failure.

Q. Check pin 20 (-CS) of console ROMs for lockup. If locked up, trace circuit back to find fault.

R. Check pins 7 and 9 through 15 of 74LS138 nearest I/O port to determine memory area accessed during lock up. *Check pin 4 (-MEMEN) for lockup. If no* signal can be found on pin 7 or pins 9 through 15, possible 74LS138 failure.

S. Check pin 11 (-CS) of MCM6810 RAMs for lock up. If locked up, trace circuit back to find fault.

T. Check TMS9901 pin 5 (-CE) for lock up. If locked up, check 74LS138 nearest I/O port for failure. Check TMS9901 pin 11, 17, and 18 for lock up. If locked up, trace circuit back to find fault.

2) KEYBOARD ERRORS

2.1 General information

After much use, the keyboard will sometimes malfunction and key presses will not appear to have any effect or will only work part of the time (either single keys or groups of keys). Keys may also show multiple entries even though only one key was pressed.
The TI-99/4A's keyboard is set up using X-Y matrix to allow encoding of output signals from a 74LS156 to drive the interrupt input of the TMS9901. This method requires only 15 lines to encode all 48 keys. Keyboard failure is almost always a mechanical problem, but occasionally one of the computer's support components will fail and cause similar problems.

2.2 Possible causes and solutions.

A. If only one key works intermittently or not at all, that single keyswitch is probably dirty or damaged. Some keyboards may allow for disassembly and repair while others make single key repair less practical than replacement of the entire unit.

B. If a group of keys has suddenly failed to work properly, it is likely that the switches in the keyboard are good and that some common component has failed. Typically, this is a broken wire or faulty driver chip. Consult a keyboard schematic to determine control lines common to groups of keys. When a common line is found, check continuity from the keyboard to the main board. If continuity exists, check loading resistors on the control lines from the keyboard connector before replacing any chips.

*C. If the ALPHA LOCK key fails to oper*ate properly and the console has been modified with the addition of a diode in the ALPHA LOCK circuit, remove the diode and replace with a piece of wire. The diode is added by some users to allow use of joysticks with the ALPHA LOCK depressed, but it sometimes introduces a timing problem and is not reliable.

D. If some keys do not work at all and others result in improper entries, check the keyboard plug connector for proper alignment.

E. If a group of keys with a control line common to the 74LS156 fail to function and continuity exists to the main board, use a logic probe to check for output pulses on pins 4 through 7 and 9 through 12 of the 74LS156.

F. If the entire keyboard fails to function and continuity exists to the main board, replace the 74LS156 and test again.

G. If 74LS156 replacement has no effect, replace the TMS9901 and test again.

3) INTERMITTENT CONSOLE LOCK UP

3.1 General information.

Occasionally, a console will suffer from lock up during regular use of software. The may be caused by either a software error, a hardware error, or a disturbance of the system. Assuming that software errors and outside disturbance (such as bumping the flex-cable connector accidentally) have been eliminated as possible causes, several conditions may cause random locking up of the console during use. Causes for random console fall into three main categories: power supply, heat and poor connections.

3.2 Possible causes and solutions.

A. Check the computer's supply line voltage. The transformer input should be approximately 117VAC. The output of the consoles internal supply should be +5V, +12V, and -5V. Approximately 5% variations for each of these are tolerable.

*B. Test the computer in different sur*roundings. If the computer consistently works at another location or with different equipment attached, it may be affected by some components connected to it or by electrical interference from the 117VAC supply.

C. If lock up only occurs after some period of use, the problem may be heat related. Make sure the console's ventilation slots are not blocked. Check the heatsink of the TMS9918A for sufficient heat-conducting grease as

NEXT PAGE

well as the TMS9904 if it has a heatsink attached. A cooler switching power supply may be installed to help further lower operating temperature.

D. If the console fails to run certain modules reliably (such as Extended BASIC) and such modules often need reseating several times, the GROM port should be changed. This typically occurs with modules that use contacts on both sides of the edge connector in the module, as in the case with Extended BASIC. Cleaning the GROM port may also help, but the problem will probably appear again shortly unless a new port is installed. When replacing or cleaning the port, be sure to remove the felt or foam wiper from the clip-on cover to the port. The wiper may be removed with a small screwdriver and solvent. Replace the plastic cover once the wiper material has been removed. Do not use any type of lubricant since this will attract dirt and cause further problems.

E. A poor connection at the consoles I/O port on the side may also lead to occasional problems. If problems persist after checking for a secure connection, remove any device connected to the side port and use alcohol and a stiff piece of paper to clean the inner contacts on the device. Remove the main board from the console and clean the edge connector with a pencil eraser followed by wiping with an alcohol-treated pad. Only light rubbing is needed with eraser to remove surface oxidation and produce a clean surface. Excessive rubbing will not help and remove too much plating from the board, especially if this procedure is repeated multiple times.

F. Check the power plug at the back of the console for tightness. A loose connection may cause occasional power failure. If this plug is loose, use pliers or other suitable device to bend the pins in the receptacle SLIGHTLY towards each other. A small piece of electrician's tape around the plug will also help secure the connection.

4) MODULE ERRORS

4.1 General Information

Certain modules may fail to work in the console either intermittently or at any time. Although this is usually the fault of the GROM port, modules do sometimes fail or have other reasons for not working. Before attempting repair, however, it is important that the module in question be tried several times in the suspect console as well as another console. Be sure that other modules are also tested in the suspect console. This will help in finding the faulty component.
If the console does not offer the TI BASIC as an option from the module selection screen and displays an "INSERT CARTRIDGE" message when no module is inserted, the console cannot find TI BASIC in GROMs 1 and 2 (chip numbers CD2156 and CD2157, respectively). Replace GROM 1 and test again. If some of the commands in BASIC fail to function, replacement of GROM 2 may also be required.

4.2 Possible causes and solutions.

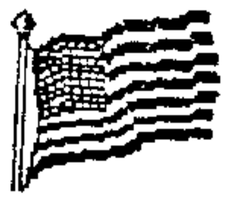
A. If various modules have intermittent problems and often require reseating the modules in the GROM port, the GROM port should be replaced or cleaned. This condition is usually noticed first with the modules that use contacts on both sides of their edge connectors, although it may occur with any module. Such a condition usually worsens with further use and is commonly a problem with consoles that have modules removed and inserted often.

B. If one particular module consistently fails to place a module selection of the TI menu, that module is most likely defective. This is also the case with modules that consistently lock up the computer when a specific function is attempted, indicating some portion of that module's program is damaged. If the module is a Texas Instruments produce device, the GROM's part numbers indicate their GROM addresses relative to each other. A memory editing or debugging program may be useful in determining a faulty GROM by allowing viewing of the GROM contents. Since only five 8K GROM address ranges may exist in a module (at GROM addresses >6000, >8000, >A000, >C000 and >E000) the number of 8K blocks locatable by the memory editor or debugger should be equal to the number of GROM chips in the module. If less blocks are found, GROM chips should be selectively removed until the faulty chip is found. If the module also has a ROM chip in it, The ROM contents may also be examined with a memory editor. The console allows for one 8K block of CPU memory at >6000

C. With the exception of some specialized GROM-emulating modules, nearly all non-TI produced modules contain only ROM and no GROM. This ROM is usually a single chip that may not be repaired. A few companies produced bank switched ROM modules (TI Extented BASIC also contained bank switched ROM in addition to GROM)> If a ROM module is determined to be at fault, it may contain bank switching components that have failed. Internal inspection of the module must be done to determine if repair is possible with replacement of bank switching components. Few, if any, aftermarket module manufactures will sell replacement ROM chips. Module replacement is usually necessary

D. If most non-TI modules will not run in the console, check the TI title screen for a 1983 copyright notice. If found, the console contains the TI operating system which prevented use of non-TI modules. This operating system may be bypassed with some software programs or aftermarket hardware adaptors. Replacement of GROM 0 in the console (chip number CD2155) with a chip from an earlier production console will replace the newer operating system and permanently solve the problem.

END



ASGARD SOFTWARE NEWS RELEASE

Asgard is pleased to announce the completion and eminent availability of the ASGARD 128K MEMORY SYSTEM.

The AMS is a product of a two-year research and development program focused on increasing TI-99/4A memory capacity. Designed by a team of hardware and software experts guided by experienced businessmen, and with the assistance and insight of a wide range of users, this device represents the begining of a new direction, as well as a blending of new proven technology.

The AMS combines flexibility with reliability and compatibility. It is the first advanced memory system for the 99/4A designed to be used exclusively as memory for programs and data.

When installed in your Peripheral Expansion Box it functions as a 32K card with standard TI-99/4A software. It is completely transparent to virtually every other TI-99/4A peripheral - it will not conflict with any floppy or hard drive controller, or even some RAM-disks. The card does not need to be configured - simply plug it in and turn on your computer. Beause it uses little power the AMS is highly reliable.

Programs designed to work with the card can access up to 128K of CPU memory simply and with a minimum of restrictions on program design. Memory can be banked in 4K increments, within a few clock cycles, anywhere within the standard 32K memory space available to TI-99/4A programs. The design used by AMS is similar to that used by TI in their TI-99/8 computer - and is currently readily accessible to programs written in Assembly and GPL.

To assist in programming for the AMS example programs with source code as well as extensive technical documentation is included with the device. All materials were prepared by software designers to be as clear and comprehensive as possible to programmers - and not just other hardware designers. The result is what we believe to be the easiest to program extended memory device for the TI-99/4A.

For non-programmers, AMS will open the door to a variety of new programs currently under development by some of the brightest programmers in the TI community today. With four times as much space available, AMS compatible programs will be more capable, faster, and have much more capacity for storing data. Types of programs can be written that would be impossible in 32K. Compatible languages under development will allow even casual programmers to write programs with access to memory.

AMS is not just a promise of new possibilities, it also represents a different way of doing things as well as a different approach to past problems.

While it may seem unusual that a software company would take the initiative in producing a new memory card, it's not so strange when you consider that you need software to make hardware useful, and a software company can insure that some of the software is written.

Further, to break with the long history of somer developers (including TI) of playing favorites and of secrecy, Asgard guarantees we will feely provide any and all software developers as much information needed to take advantage of the AMS. It's time to end the games that have hurt the community in the past, and to bury the hatchet somewhere other than each others backs.

Finally, since no one likes to buy something that becomes obsolete tomorrow - all users can be assured an investment in AMS will be protected by a company that has been serving the TI community for 10 years. Asgard will provide reasonably priced upgrades and even trade-in options as we continue to develop this technology. Further, any software written for AMS will be fully compatible with future developments with few if any changes.

AMS is not an end in itself, it is a beginning on a path to liberating the TI-99/4A from memory constraints. It also represents a new way to do business in the TI community.

The ASGARD 128 MEMORY SYSTEM requires a TI-99/4A with a Peripheral Expansion Box and a disk system. It is compatible with all disk controllers, all video cards, and some RAM disks and memory cards, as well as virtually all other cards for the TI-99/4A. It is not guaranteed to function with the Myarc or Corcomp RAM-disks, or the TI, Corcomp or Myarc 23K cards. No problems have been encountered with Horizon RAM-disks to date.

The suggested retail price of the AMS is $119.95. At this time all design and testing of the design has been completed, and it is expected to be in stock by the end of September.

To order, send a check or money order for $119.95, plus $10.00 S&H (in North America - $20.00 for Airmail shipping elsewhere) to:

Asgard Peripherals
P.O. Box 10697
Rockville, MD 20849-0697

COD and credit card orders are not accepted. All orders to U.S. customers will be shipped via UPS Ground - please allow 4-6 weeks for delivery.

Programmers may receive a free packet containing programming information by sending a post card to the above address. Again, please allow 4-6 weeks for delivery.

END

BITS AND BYTES
from Guilford 99'ers UG

RF MOD REPAIR

This article is for our friends who still may be using the RF modulator rather than a monitor.

Many TI owners are not as affluent as I am and use a TV instead of a monitor. Occasionally, the RF modulator needs to be fined tuned to eliminate annoying background noise, such as humming or buzzing. A simple internal adjustment on the modulator will often alleviate this problem. The following procedure is to be done when all equipment is on and operating. (If you have the old version of the TI900 Video Modulator, this procedure will not work.)

You will need a small, flat thin-bladed screwdriver.

1) Turn the volume of the TV all the way down, but do NOT turn it off.

2) Select the Master Title Screen on the computer.

3) Using the title screen color grid, fine tune the TV to the best color picture you can.

4) Using the screwdriver, pry off the lid of the modulator by lifting under one edge of the lid near the idntification holding it on.

5) lift off the lid and turn the TV volume up to half.

6) Insert the screwdriver blade into the slot of the small box labelled CV1 and turn it slightly until the background noise is at a minimum. (This should take less than 1/8th of a turn.)

This should take care of the any problems with background noise.

SCROLL DEMO

Here's a little demo which will allow you to scroll part of a screen and only takes up two lines of program code.

```
100 CALL SCREEN(15):: PRINT
: : : :"THIS PROGRAM WILL AC
CEPT ANY INPUT AND SCROLL UP
 1 LINE."
110 PRINT : :"BUT ONLY THE B
OTTOM HALF OF THE SCREEN WIL
L SCROLL. THE TOP HALF WILL
STAY INTACT."
120 PRINT :"_______________
__": : : : : : : : : :: B
$=RPT$("",252)
130 ACCEPT AT(24,1)SIZE(28):
A$ :: A$=A$&RPT$(" ",28-LEN(
A$)):: B$=SEG$(B$,29,224)&A$
 :: DISPLAY AT(15,1):B$ :: G
OTO 130
```

If you want to scroll down, change the "29" to a "1" in line 130. To change the location where scrolling occurs, change the DISPLAY AT. If it is higher than 15, it will split the screen so that the top and bottom will scroll and the middle stays the same. (You will also need to change the ACCEPT AT so that it lines up with the scrolling screen.)

You are limited to 9 lines because strings are limited to 255 characters and 9 lines takes up 252. Author of this is Edwin McFall

BASIC TIPS

1. HOW TO DISABLE THE "FUNCTION-QUIT" HARDWARE RESET: TI Basic and Extended Basic has two ways to exit, one by typing "BYE" which will properly close all files, or by pressing "Function=(QUIT)". The latter method really should not be used at all since files will not be closed and unpredictable things can happen if function quit is pressed while files are open. Unfortunately, many of us had the nasty experience of accidently hitting "Function Quit" with the result that everything in memory was lost and files were scrambled. If you have Extended Basic and 32K memory, the following will disable "Function Quit": CALL INIT::CALL LOAD(-31806,16). This can be typed in as a direct command, or could be the first line of an extended basic program.

2. HOW TO SPEED UP EXTENDED BASIC While XB offers faster execution speed for some applications compared to console basic, XB can be speeded up even further by disabling sprite graphics (naturally this works only if the program does not use sprite graphics). The program statement is: CALL INIT::CALL LOAD(-31878,0). There are several different releases or versions of Extended Basic and the speed-up effect will be more pronounced with some versions than with others. 32K memory is required.

3. HOW TO RECOVER MEMORY IN TI BASIC/EXTENDED BASIC WITh DISK DRIVE ATTACHED. The TI operating system automatically sets aside memory to serve three concurrent open files. A minimum of 534 bytes of memory are taken up by general expansion overhead plus 518 more bytes for each of the three files opened up by default, or a total of just about 2K. If you know that you will have only one file open, key in the following DIRECT COMMAND: CALL FILES(1) (Press ENTER) NEW (Press ENTER). This sequence will recover 1k of precious memory. Please note that this sequence can be keyed in as a command only and cannot be used as a program statement. Don't forget the NEW or results will be unpredictable. This procedure can be used with both TI Basic or Extended Basic. With TI Basic and attached disk this is more essential than every since TI Basic will only address 16K and you can ill afford to lose much of that.

EXPLORING BASIC PROGRAMS

by Tim MacEacEachern

The program listed here demonstrates how BASIC programs are stored in the 99/4A. The program as listed will work in Extended Basic with the Memory Expansion card or peripheral attached. A similar program can be run in normal BASIC with the Editor/Assembler or Mini Memory module inserted. To convert this program to normal BASIC simply change the calls to subroutine "PEEK" in lines 200, 240 and 260 into calls to subroutine "PEEKV". That is, add a "V" between the "PEEK" ans the "(" in each line. this program will not work properly in Extended Basic unless you have the memory expansion.

The techniques used in this program are intended to make it as easy to understand as possible, while still show-

ing how the DEF statement in BASIC can be used to do all the hard work for you. For instance, lines 100 to 130 of the program create a function HEX which will convert a string of hexadecimals (base 16) digits into a decimal number.

As can be seen in addresses as used by assembler language programmers.

Line 130 takes the string of hexadecimal digits given to it and pads *it with leading zeroes to make sure that* there are four hex digits.

Then function HEX4 is called to evaluate this four-digit hex number. In line 120, HEX4 splits the number into two two-digit hex numbers and combines them to get the proper decimal result. Similarly, line 110 splits a two-digit hex number into one-digit numbers. Line 100 then is used to figure out the value of each separate hexadecimal digit.

Using nested DEF statements as in this program can simplify development of a working program, but be warned that DEF statements take considerably longer to run than the exact same code put directly into your lines wherever needed . Still, you may find it convenient to write some programs that consist solely of DEF statements ! After such a program is RUN in normal BASIC (or in Extended BASIC without the memory expansion), the defined functions will be available to use in BASIC'S calculator mode. For instance, if your program consisted of lines 100 to 130 only, it would provide a conversion function from hex to decimal that you could use while in calculator or direct command mode.

Let's get back to the program. Line 140 defines a function that is used to convert a 16-bit unasigned number (from 0 to 65535) into a 16-bit signed number (from -32768 to 32767). For some strange reason BASIC insists on signed numbers for addresses passed to PEEK, PEEKV, LOAD and POKEV. So whenever an unsigned address is calculated function MA is used to convert it to a signed number. This function works by comparing its argument to the largest positive value allowed. If the number is too big the comparison yields a value of -1. The rest of the expression then caused 65536 to be subtracted from the argument value, giving the correct result. If *the origional number is okay (*from 0 to 32767) the comparison yields a result of 0 end the value of the function is the same as the value of its paremeter.

It seems complicated to write functions like this, but try to figure them out - you may find them fascinating.

BASIC stores your program in two sections. In the top of memory it stores each line of the program, not necessarily in the correct order. As a matter of fact, each time you edit a line, it becomes the last line in this area, with all other lines packed together above it. Each statement is made up of three parts. The first byte is the length of the rest of the statement in memory. The last byte is zero, and in between are bytes that represent the particular BASIC statement you have written. Basic keywords are translated into a single byte each (known as a token) while strings and numeric constants are represented as a leading token (199 or 200) followed by a length byte, followed by the ASCII character values of the string. By running this program you can determine how other elements of a BASIC program are stored.

Underneath the statements (that is, lower in memory) is a list of statement numbers and pointers to the first token in each statement. Each statement in your program has a four-byte entry in this list. The bottom two bytes store the statement number. The top two bytes are a pointer to the first token in the statement (the byte following the length byte). This program goes through this list and prints out each token in the statements of your program.

Pointers to the top byte in the statement pointer list and the bottom byte in the list are storedin the scratchpad RAM and read by lines 150 to 180. The loop that starts in line 190 examines each statement in the program.

If you have gotten this far in the article, you will understand how the rest of the lines in the program print out each token of eachline.

```
100 DEF HEX1(X$)=POS("123456
7B9ABCDEF",X$,1)
110 DEF HEX2(X$)=HEX1(SEG$(X
$,1,1))+HEX1(SEG$(X$,2,1))
120 DEF HEX4(X$)=HEX2(SEG$(X
$,1,2))6+HEX2(SEG$(X$,3,2))
130 DEF HEX(X$)=HEX4(SEG$("0
000"& X$,LEN(X$)+1,4))
140 DEF MA(X)=X+65536*(X>327
67)
150 CALL PEEK(MA(HEX("8332")
),A,B)
160 TOSL=MA(A6+B)
170 CALL PEEK(MA(HEX("8330")
),A,B)
180 BOSL=MA(A6+B)
190 FOR PTR=TOSL-3 TO BOSL S
TEP -4
200 CALL PEEK(PTR,A,B,C,D)
210 PRINT "STATEMENT #";A6+B
220 PRINT "TOKENS:"
230 SPTR=MA(C6+D)
240 CALL PEEK(SPTR-1,L)
250 FOR I=0 TO L-1
260 CALL PEEK(SPTR+I,X)
270 PRINT X;
280 NEXT I
290 PRINT : :
300 NEXT PTR
310 END
                        END
```

DON'T FORGET THE NEXT CONNI MEETING IS AT THE JANIS CENTER.

MEETING DATES FOR 1992 — 1993

3RD SATURDAY

21 NOV 1992 *
19 DEC 1992
16 JAN 1993
20 FEB 1993

* Meeting will be held at the Janis Center due to OSU football games.

4TH WEDNESDAY

24 NOV 1992
22 DEC 1992
27 JAN 1993
24 FEB 1993
24 MAR 1993
28 APR 1993

C.O.N.N.I. BOARD MEMBERS

Pres. - John Parkins	614/891-4965
Treas - Everett Wade	614/262-6346
Sec/Sat - Jim Peterson	614/235-3545
Sec/Wed - Dick Beery	614/459-3597
Membership - John Parkins	614/891-4965
Librarian - Chuck Grimes	614/268-8821
Disk - Dick Beery	614/459-3597
Cassette - Everett Wade	614/262-6346
Cartridge - Ken Marshall	614/876-1670
NL Exchange - Jean Hall	614/885-4223
TIABS BBS	614/851-0708
Vice Pres. - Chuck Grimes	614/268-8821
Spirit of 99 BBS	614/263-3412
Irwin Hott	614/263-5319
Dick Beery	614/459-3597
Co-Editors/Spirit of 99 Newsletter	
Jean Hall	614/885-4223
Bob DeVilbiss	614/891-0566

C.O.N.N.I.
181 HEISCHMAN AVE
WORTHINGTON, OH 43085

THEY FOUGHT
FOR FREEDOM