

PROGRAMS THAT WRITE PROGRAMS
PART 6

by Jim Peterson

The first five parts of this series were written long ago, but since then I have found a new method to write programs that really do write programs. I must give Karl Romstedt credit for this idea.

To illustrate this technique I will use a program which writes and autoloader to display a diskfull of programs by their complete name rather than the abbreviated filename. This is the LOAD program which I put on all my TI-PD disks.

First, we key in the part which will always be a part of the LOAD program. Do not change the line numbers because there is a reason for them, and leave that REM in line 11 because something else will be plugged in there later.

```

10 CALL CLEAR :: DIM M$(127)
:: CALL SCREEN(5):: FOR S=0
TO 14 :: CALL COLOR(S,2,8)::
NEXT S :: CALL PEEK(B198,A)
:: IF A<>170 THEN CALL INIT
11 REM
12 ON WARNING NEXT
13 X=X+1 :: READ M$(X):: IF
M$(X)<>"END" THEN 13
14 R=3 :: FOR J=1 TO X-1 ::
READ X$ :: DISPLAY AT(R,1):S
TR$(J);TAB(4);X$ :: R=R+1 ::
IF R<23 THEN 17
15 DISPLAY AT(24,1):"Choice?
or 0 to continue 0" :: ACCE
PT AT(24,26)VALIDATE(DIGIT)S
IZE(-3):N :: IF N>X-1 THEN 1
5
16 IF N<>0 THEN 19 :: R=3
17 NEXT J
18 DISPLAY AT(24,1):"Choice?
" :: ACCEPT AT(24,9)VALIDATE
(DIGIT):N :: IF N=0 OR N>X-1

```

```

19 CALL CHARSET :: CALL CLEA
R :: CALL SCREEN(8):: CALL P
EEK(-31952,A,B):: CALL PEEK(
A*256+B-65534,A,B):: C=A*256
+B-65534 :: A$="DSK1."&M$(N)
:: CALL LOAD(C,LEN(A$))
20 FOR J=1 TO LEN(A$):: CALL
LOAD(C+J,ASC(SEG$(A$,J,1)))
:: NEXT J :: CALL LOAD(C+J,0)
):: GOTO 10000
10000 RUN "DSK1.1234567890"

```

Now, save that "source code" by SAVE DSK1.CAT/S, MERGE. Then key in this "assembler" which will convert the "source code" into an "object code."

```

100 OPEN #1:"DSK1.CAT/S",VAR
IABLE 163,INPUT
110 OPEN #2:"DSK1.CAT/O",VAR
IABLE 163,OUTPUT
120 FOR J=10 TO 21 :: LINPUT
#1:M$ :: PRINT #2:CHR$(0)&C
HR$(J)&CHR$(156)&CHR$(253)&C
HR$(200)&CHR$(1)&"2"&CHR$(18
1)&CHR$(199)&CHR$(LEN(M$))&M
$&CHR$(0):: NEXT J
130 PRINT #2:CHR$(255)&CHR$(
255):: CLOSE #1 :: CLOSE #2

```

Note what this routine does. It reads in each line of the tokenized CAT/S and prints it back out to CAT/O preceded by line numbers 10 to 21 in tokenized two-byte format followed by the tokens for PRINT #2, the tokens for a quoted string followed by the CAT/S record and the CHR\$(0) end-of-line indicator. Then it prints the double-255 end-of-file indicator and closes the files

Now key in the CATWRITER program.

```

1 CALL CLEAR :: CALL TITLE(1
6,"CATWRITER"):: CALL CHAR(1
27,"3C4299A1A199423C"):: DIS
PLAY AT(2,10):"Version 1.4":
;TAB(8); Tigercub Softwar
e"
2 DISPLAY AT(15,1):"For free
":"distribution":"but no pri
ce or":"copying fee":"to be

```

```

charged. :: FOR D=1 TO 300
:: NEXT D :: CALL DELSPRITE(
ALL)
3 DISPLAY AT(2,3)ERASE ALL:"
TIGERCUB CATWRITER V.1.4"::
" Will read a disk directory
,":"request an actual progra
m":"name for each program-ty
pe"
4 DISPLAY AT(7,1):"filename,
and create a merg-":"able Q
uickloader which dis-":"play
s full program names and":"r
uns a selected program."
5 DISPLAY AT(12,1):" Place d
isk to be cataloged":"in dri
ve 1 and press any key" :: C
ALL KEY(0,K,S):: IF S=0 THEN
5
9 OPEN #2:"DSK1.CAT",VARIABL
E 163,OUTPUT
100 OPEN #1:"DSK1.",INPUT ,R
ELATIVE,INTERNAL :: INPUT #1
:N$,A,J,K :: LN=1000 :: FN=1
100
110 DISPLAY AT(12,1):"Disk n
ame?":N$ :: ACCEPT AT(14,1
)SIZE(-28):N$ :: LX$=STR$(14
-LEN(N$)/2):: LXLEN=LEN(LX$)
120 PR$=CHR$(0)&CHR$(11)&CHR
$(162)&CHR$(240)&CHR$(183)&C
HR$(200)&CHR$(1)&"1"&CHR$(17
9)&CHR$(200)&CHR$(LXLEN)&LX$
130 PR$=PR$&CHR$(182)&CHR$(1
81)&CHR$(199)&CHR$(LEN(N$))&
N$&CHR$(0)
140 PRINT #2:PR$
145 DISPLAY AT(23,1):"To omi
t a file, press Enter"
150 X=X+1 :: INPUT #1:P$,A,J
,B :: IF LEN(P$)=0 THEN 190
:: IF ABS(A)=5 OR ABS(A)=4 A
ND B=254 THEN 160 ELSE X=X-1
:: GOTO 150
160 DISPLAY AT(12,1):P$;"
PROGRAM NAME?" :: ACCEPT AT
(14,1)SIZE(25):F$ :: IF F$="
" THEN X=X-1 :: GOTO 150
170 PRINT #2:CHR$(INT(FN/256
))&CHR$(FN-256*INT(FN/256))&
CHR$(147)&CHR$(200)&CHR$(LEN
(F$))&F$&CHR$(0):: FN=FN+1
180 M$=M$&CHR$(200)&CHR$(LEN
(P$))&P$&CHR$(179):: IF X<11
THEN 150
190 IF M$="" THEN 210
200 PRINT #2:CHR$(INT(LN/256
))&CHR$(LN-256*INT(LN/256))&
CHR$(147)&SEG$(M$,1,LEN(M$)-
1)&CHR$(0):: LN=LN+1 :: M$="

```

```

:: A=0 :: IF LEN(M$)/256 IN
EN 150
210 PRINT #2:CHR$(INT(LN/256
))&CHR$(LN-256*INT(LN/256))&
CHR$(147)&CHR$(200)&CHR$(3)&
"END"&CHR$(0)
220 PRINT #2:CHR$(255)&CHR$(
255):: CLOSE #1 :: CLOSE #2
230 DISPLAY AT(8,1)ERASE ALL
:"Enter -":" NEW":" ME
RGE DSK1.CAT*":" DELETE "
DSK1.CAT*""":" SAVE DSK1.L
OAD"
240 SUB TITLE(S,T$)
250 CALL SCREEN(S):: L=LEN(T
$):: CALL MAGNIFY(2)
260 FOR J=1 TO L :: CALL SPR
ITE(#J,ASC(SEG$(T$,J,1)),J+1
-(J+1=S)+(J+1=S+13)+(J>14)*1
3,J*(170/L),10+J*(200/L)::
NEXT J
270 SUBEND

```

Next, enter MERGE DSK1.CAT/O and that "object code" will pop into place right after line 9. If you list it, it will look like a blown file, because most of the token codes are unprintable, but don't worry. Save the program as CAT-WRITER.

When you run the program, it will open an output MERGE format file called CAT and write those merged lines from CAT/O in MERGE format. Then it will open the disk you are cataloging, read the directory sector, and ask you for a disk name with the existing diskname as default. You can select any disk name you want to title the menu screen, up to 28 characters long. Line 110 computes the position to center the title, and lines 120-140 write to the CAT file a tokenized line 11 (overwriting that REM line) to display your title at the top of the screen.

Line 150 reads each filename from the disk directory, skipping over anything

that is not a program (no one yet has been able to tell me how to distinguish an assembly image "program"!). For each filename, it will ask you for a complete program name. If you don't want a program on the menu (such as an XB program that is run from another program, or an image file), just press Enter. Otherwise the program name you select will be printed as DATA by line 170, in tokenized format in lines starting with 1100 (note the FN=1100 in line 100) and incremented by 1. Lines 180-200 assemble the filenames into DATA lines of up to ten names, and tokenize them in lines beginning with 1000.

When the last filename has been read, line 210 prints one last DATA item "END" to signal line 13 to stop reading, and then prints the double-255 end-of-file. Then you are given instructions to clear memory with NEW, merge in the CAT file, delete it because you don't need it any more, and save it back as LOAD.

When you list the LOAD program, you will find the original CAT/S restored in lines 10-19 and 1000, the line to display the title in line 11, the filenames in DATA lines starting with 1000 and the program names in DATA lines starting 1100.

When you run the program, it will display the disk name, and read the filenames into an array. Then it will display the program names, numbered, on as many screens as necessary, and ask you to select a program by number. The corresponding filename by number is selected from the array, and lines 19-20 rewrite line

10000 to RUN that filename. List the LOAD program after you have used it to load something, and you will see that it has changed.

That algorithm in lines 19-20 was published in one of the earliest 99'ER magazines, in a letter by A. Kludge. It has been the basis for every XBasic menu loader, and has saved us uncounted thousands of hours. The author had asked me not to reveal his identity, but I think I can tell you that "A. Kludge" was really the late Dr. Stefan-Romano, who passed away recently at the age of 57. He was a brilliant man who did much for the TI world, at first as editor of the IUG library, and then through the Amnion library and Amnion Helpline. He was of great help to me on several occasions.

Some of you may have obtained from me a copy of CATWRITER which wrote GOSUB 21 in line 12, and CALL LOADs in lines 21-25 to change the cursor to my Tigercub emblem. If you have begun to have problems with the resulting LOAD program or with my previous Tigercub Menuloader which used the same CALL LOADs, I have finally found out the cause. When my Horizon RamDisk is on, any program containing those CALL LOADs will lock up the second time it is run!

FOR SALE

Gary Pulda of 25 Barrett Ave., Worcester has a complete I.I. System for sale. This includes a black & silver console, Myarc disk drive, RS 232 and 32K mem. exp Axiom printer and 22 software cartridges or disks. All manuals are included. Gary will sell the complete system for \$250 or individually. Call him for individual prices or see Jim Cox at the next meeting.

THANK YOU

I want to take this opportunity to thank Jim Peterson, the Tigercub, for his outstanding contributions to the I.I. community. This newsletter and many of the others around the country would have a lot of blank pages if it weren't for the writings of the Tigercub. You will find a list of some of Jim's most recent disk offerings on another page of this newsletter. You should order some of these excellent disks and say "Thank You" to Jim. Ed.

The Complete TI ADVENTURE COMPENDIUM

Jam-packed disks of all the fairware and public domain adventures ever written for the TI-99/4A and Geneve

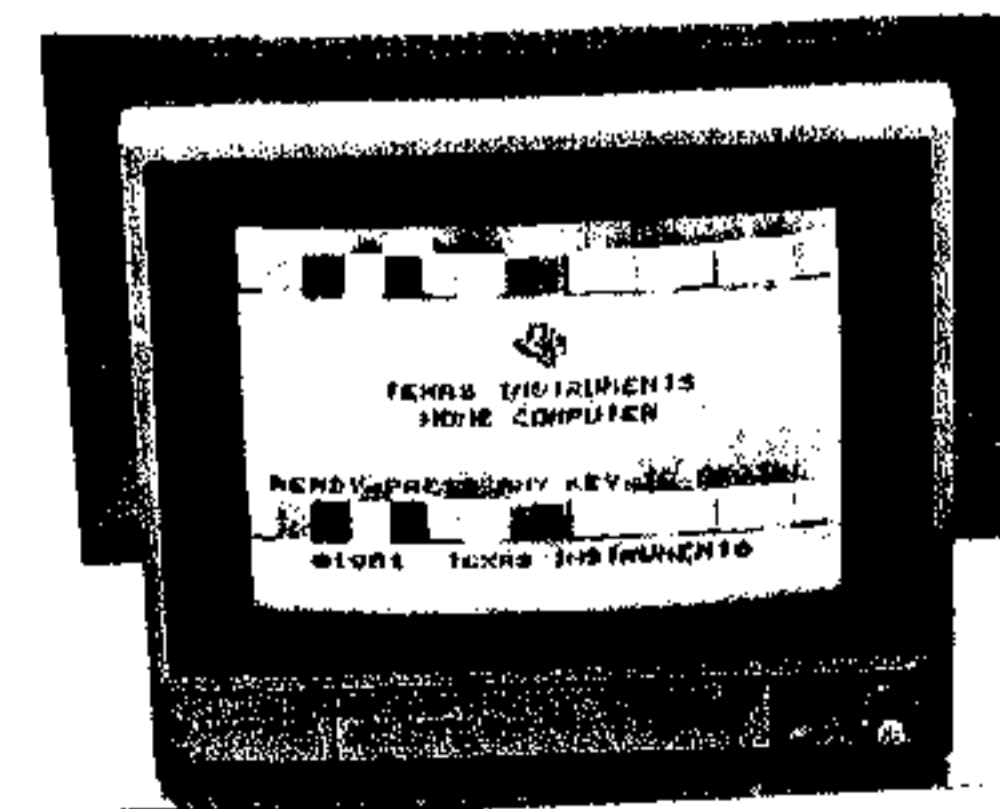
THE TI ADVENTURE COMPENIUM DISK SET:

Order directly today

Make out checks to MUNCH and mail to
M.U.N.C.H.
560 Lincoln St.
P.O. Box 7193
Worcester, MA 01605

\$11.95 - 3 full DSDD Disks: \$15.95 - 5 DSDD:
\$19.95 - 10 SSSD

MUNCH



THIS MONTH I'D LIKE TO CONTINUE WITH MORE COMMANDS TO THE PRINTER THAT MAY HELP THE USER GET MORE OUT OF THIS VERY VERSATILE PERIPHERAL. BELOW ARE THREE PRINT OUT EXAMPLES.
(1)

ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789abcdefghijklmnopqrstuvwxyz

132 COLUMN TEXT PRINTOUT FROM PROGRAM CONTROL IS INDICATED IN THE LINE ABOVE. THE PRINTER WAS FIRST PUT IN VARIABLE 132 MODE, FOLLOWED BY CONDENSED MODE (TO ALLOW FOR SMALL ENOUGH CHARACTERS TO FIT THE 132 COL. LINE) AND THEN THE RIGHT MARGIN WAS SET TO THE 132ND COL.

ABCDEFGHIJKLMNOPQRSTUVWXYZ01
23456789abcdefghijklmnopqrst
vwxyzABCDEFGHIJKLMNQR
STUVWXYZ12345678ABCDEFGHIJKL
MNOPQRSTUVWXYZ

(2)

(3)

ABCDEFGHIJKLMNOPQRSTUVWXYZ012
3456789abcdefghijklmnopqrstuv
wxyzABCDEFGHIJKLMNQR
STUVWXYZ012345678abcdefghijklmnop
mnopqrstuvwxyz

28 COLUMN PRINTOUT IS INDICATED IN THE LINES IMMEDIATELY ABOVE STARTING IN COLUMN 00 AND ENDING IN COLUMN 27 ON MY PRINTER. THE RIGHT MARGIN WAS SET TO 28, WHICH CAUSES ALL THE OUTPUT TO THE PRINTER TO AUTOMATICALLY GO TO A NEWLINE AND CARRIAGE RETURN WHEN THAT POINT IS REACHED.

(1)

```
100 OPEN #1:"PIO",VARIABLE 1
32
110 PRINT #1:CHR$(15);CHR$(27);CHR$(81);CHR$(132)
120 PRINT #1:"ABCDEFGHIJKLMN
OPQRSTUVWXYZ0123456789abcde
fghijklmnopqrstuvwxyzABCDEF
GHIJKLMNOPQRSTUVWXYZ012345678a
bcdefghijklmnopqrstuvwxyz"
```

29 COLUMN PRINTOUT STARTING IN COLUMN 48 AND ENDING IN COLUMN 76 WAS SHOWN IN THE PRINTOUT ATOP THE COLUMN OF TEXT ON THE RIGHT. THIS IS ACCOMPLISHED ON THE EPSON COMPATIBLE WITH JUST ONE COMMAND. YOURS MAY TAKE SEVERAL.

THE 132 COLUMN TEXT PRINTOUT MAY BE USEFUL TO YOU IN ANY NUMBER OF APPLICATIONS. ACTUALLY I CAN BY TWEAKING SOME NUMBERS INCREASE THAT A BIT TOO. IF WE LOOK AT THE PROGRAM (1) YOU CAN SEE THAT WE MUST FIRST OPEN THE PRINTER AS A VARIABLE 132 TO ALLOW FOR THE ADDITIONAL SIZE COLUMN WISE OF THE LINE. YOU MUST SET PRINTER IN THE CONDENSED MODE USING CHR\$(15) AND THEN SET THE RIGHT MARGIN TO 132 WITH CHR\$(81); CHR\$(132). I'LL BET YOU CAN TELL WHICH CHR\$ DOES THAT. NOW AS LONG AS YOU PRINT TEXT TO THE PRINTER YOU WILL BE IN THIS MODE. UNTIL YOU EITHER SEND A RESET TO THE PRINTER OR SHUT IT OFF.

(2)

```
100 OPEN #1:"PIO"
110 PRINT #1:CHR$(27);CHR$(81);CHR$(28)
120 PRINT #1:"ABCDEFGHIJKLMN
OPQRSTUVWXYZ0123456789abcde
fghijklmnopqrstuvwxyzABCDEF
GHIJKLMNOPQRSTUVWXYZ12345678A
BCDEFGHIJKLMNOPQRSTUVWXYZ"
```

(3)

```
100 OPEN #1:"PIO"
110 PRINT #1:CHR$(27);CHR$(88);CHR$(48);CHR$(76)
120 PRINT #1:"ABCDEFGHIJKLMN
OPQRSTUVWXYZ0123456789abcde
fghijklmnopqrstuvwxyzABCDEF
GHIJKLMNOPQRSTUVWXYZ012345678a
bcdefghijklmnopqrstuvwxyz"
```

THE 28 COLUMN PRINT (ACTUALLY ANY LENGTH TO 80 COLUMNS IN THIS EXAMPLE) IS USEFUL OBVIOUSLY FOR COLUMNIZING TEXT AS I AM DOING HERE. IN (2) PROGRAM YOU CAN SEE THAT ALL THAT I HAVE TO DO IS SET THE RIGHT MARGIN TO 28 USING CHR\$(28) OR ANY OTHER VALUE TO 80.

THE 29 COLUMN PRINT STARTING IN COLUMN 48 EXAMPLE NUMBER (3) IS A GREAT WAY TO PRINT OUT A SECOND OR THIRD COLUMN STARTING IN N1 COLUMN AND ENDING IN N2 COLUMN. IN THE PROGRAM YOU SEE CHR\$(88) WHICH ON MY PRINTER CAUSES IT TO ACCEPT TWO MORE COMMAND ATTRIBUTES, THE LEFT MARGIN [CHR\$(48)] AND THE RIGHT MARGIN [CHR\$(76)] SETTINGS THESE BEING FROM COLUMN 48 AND PRINTING THROUGH COLUMN 76, ALLOWING FOR A 4 COLUMN MARGINE ON MY PAPER. THIS MAY BE ELEMENTARY TO SOME OF YOU, BUT I KNOW THAT THERE ARE MANY EXTENDED BASIC PROGRAMS OUT THERE THAT I'VE ALWAYS WANTED TO CUSTOMIZE THE PRINTING ROUTINES IN, AND DIDN'T BECAUSE I DIDN'T WANT TO TAKE THE TIME TO LEARN WHAT MY PRINTER CAN DO. I MAY DEAL WITH THE PRINTER COMMANDS ONE MORE MONTH, PAPER SPACING, AND WE'LL GET BACK TO THE HARDWARE ASPECT. ML

SPEECH
By Andrei Derksen, Netherlands

This article ("SPEECH") talks about some experimenting I did with the speech synthesizer.

The files starting with "DEF..." are the merge files listed in the back of the Extended Basic manual about speech. They go with the program "SUFFIXES" (also in the back of the manual). Suffixes is the ready-to-use program, the merge-files are needed for other things you might want to do with speech. This is (more or less) explained in the article. As for "S--T", well, read the article and run the program. Even the title is self-explanatory.

"FRONTTRUNC" lets you remove bytes of the beginning of a speech word, More or less the reverse of the program Suffixes.

Those of you who own a Speech Synthesizer will know that although it might not always be very useful, it can be fun. Lately I have been playing around with it. In this article I would like to tell you about my findings.

As you might know, there are several ways to add res to your Speech Synthesizer. As far as TI is concerned, you have Extended Basic and the Terminal Emulator. In Extended Basic there is a built-in vocabulary. These words and phrases can be accessed by using the CALL SAY and CALL SPGET statements. If you use TE II, you can have the Speech Synthesizer pronounce every word you want it to. Even foreign languages, albeit with the restriction that you will always have to deal with the American accent built in to it.

However, if you have Extended Basic, 32K and a disk drive you don't need the TEII module to be able to have the TI say anything you like it to. There are programs available on disk which accomplish the same task as the TEII module.

I knew that with the TEII module, you can change the difference in pitch between the beginning of the sentence and the end. I also knew you can make the voice sound higher or lower. So I started wondering if it would also be possible to do the same with this Extended basic program. The best way to find out of course is to look at the basic listing. Most of the program is written in assembler, so that is of no help to us. But there is one very interesting line. In my version, this is line 150. You might have a different version, but it should be more or less the same. This line reads CALL LINK("SPEAK",B\$,43,128) This line hands over control to the assembler program called "SPEAK", and feeds it the string B\$ (the sentence you want it to say), and the values 43 and 128. These two values are particularly interesting. The first value (43) regulates how high the voice sounds. The second determines the difference in pitch between the beginning and the end of the sentence.

When you start changing these values, interesting things happen. If you increase the first value, the voice will become lower. If you decrease it, it will become higher. Setting it to zero will not, as you might expect, lock-up the computer. Try it. Ever heard a whispering computer before? As far as the second value is concerned, decreasing it will make the speech sound more and more like one would expect from a computer. If one sets it higher than 128, the pitch difference will start to become unnatural. By the way, you can run and break the computer without any problem. However it would be a good idea to skip the first line. This should read something like CALL LOAD(DSK1.xxx) The files loaded by this statement only need to be loaded once every time you use the program. Loading them every time you run the program is a waste of time.

Even in Extended Basic it is possible to have the computer say more than just the built-in words. Take the program Burt+Ernie. Whoever made that must have spent a lot of time on the matter. It isn't completely clear to me how it was done, but this much I do know. When listing the program, after a while you come across some kind of coding. Only, this is not coding. Type in the following line into Extended Basic: CALL SPGET("#TEXAS INSTRUMENTS#",A\$) :: PRINT A\$

Looks familiar? Also notice the length of the string. To check that, type PRINT LEN(A\$) (After having entered the CALL SPGET). The coding is in fact a long string of ascii chars which enable the Speech Synthesizer to pronounce the word. To see the actual ascii codes use this short program:

```
10 CALL SPGET("#TEXAS INSTRUMENTS#",A$) 20 A=LEN(A$)
30 FOR I=1 TO A 40 PRINT ASC(SEG$(A$,I,1)) 50 NEXT I
```

It takes many more characters for the Speech Synthesizer to pronounce the word than for us to spell it. What the programmer of Burt+Ernie did was to rearrange these chars in such a way as to make new words. The work this must have been is absolutely mind boggling. Another thing I don't understand yet, is how the programmer managed to get these speech strings into DATA statements.

One thing you can do yourself is use the program in the back of the Extended Basic manual. (If you never managed to take the time to enter it, it is included on this disk). What this program does is to truncate the end of any word in the vocabulary to the length you want it, and then add one of the provided endings. Using this program I made a small program which makes the computer pronounce a four-letter word not in the vocabulary (included on this disk). Another thing I discovered is that the first two bytes of any speech string are always 96 and 0. The third byte is always the length of the remaining string, i.e. the total length minus 3 (bytes). So, you could create your own speech strings. But don't expect anything meaningful. It will probably sound like a lot of burps and whistles. Have fun!

From N.H. 99 end'

TI-PD public domain software for the TI-99/4A computer, \$1.50 per disk postpaid (minimum 8 disks please); number of sectors filled is indicated in parentheses). For a 9-page catalog listing all titles and authors, send \$1 which is deductible from first order. (specify TI-PD catalog) Offered as a copying service only, without warranty other than that copies are equal to the original. Make checks payable to Tigercub Software (no credit card orders). Send to Tigercub Software, 156 Collingwood Ave., Columbus OH 43213.

600. Sam Moore Jr. Music #1 (341)
601. Sam Moore Jr. Music #2 (343)
602. Sam Moore Jr. Music #3 (348)
603. Sam Moore Jr. Music #4 (337)
604. Bill Knecht Hymns (334)
605. Christmas Music (318)
606. Holiday Music (339)
607. Great Songs by Bill Knecht (351)
608. Music by Bill Knecht (295)
609. March Music (329)
610. Tigercub Country Music (356)
611. Christmas Sing-Along (352)
612. J. Stephen Foster Music #1 (332)
613. J. Stephen Foster Music #2 (317)
614. Bach Music Programs (338)
615. Sing-Along Music (351)
616. Some of the Best Music (343)
617. Classical Music (340)
618. Assorted Music (346)
619. Chopin's Polonaise (280)
620. Assorted Music #2 (351)
621. Hamilton UG Music Package #1 (346)
622. Assorted Music #2 (349)
623. A Diskfull of J.S. Bach (345)
624. Assorted Music #4 (358)
625. Assorted Music #5 (347)
626. J.S. Bach Music (340)
627. Assorted Music #6 (354)
628. Some of the Very Best (349)
629. Ollie Hebert's Music (340)
630. Gregory Rashall Music Master (287)
631. Assorted Music #7 (352)
632. Sonata for Pianoforte (222)
633. Sonata for Pianoforte DS/SD
634. Strange Music (337)
635. Chuck Berry Tunes (218)
636. Christmas Songs w/Graphics (310)
637. Assorted Music #9 (337)
638. Classical Music #2 (338)
639. Assorted Music #10 (347)
640. Marches and College Songs (336)
641. Another Sing-Along (345)
642. Sing-Along Music #2 (236)
643. Christmas Music #2 (351)
644. Christmas Sing-Along #2 (340)
645. Classical Music #3 (352)
646. Assorted Music #11 (350)
647. Music Doodlers and Tinytunes (302)
648. Rhapsodie in Blue (287)
649. Assorted Music #8 (354)
650. Christmas Music w/Graphics 2 (357)
651. Sorgan II (145)
652. Christmas Music w/Graphics 3 (352)
653. Pop Deo VI.1 (225)
654. Christmas Music w/Graphics 4 (255)
655. Assorted Music #12 (349)
701. Musical Education (350)
702. Musical Education #2 (318)
703. Musical Education #3 (188)
710. American Flags (360)
711. Flags of the World (345)
712. Geography - U.S. States (341)
713. Geography - U.S. States #2 (212)
714. World Geography (179)
730. American History (48)
750. Alphabet w/Speech (343)
751. Children's Programs w/speech (357)
752. Alphabet for Preschool (329)
753. Children's Prog. w/Speech #2 (335)
755. Shapes, Colors, Directions (173)
760. Spelling (324)
770. Vocabulary and Reading (293)
780. Preschool Math (341)
790. Elementary Addition, Subtract (257)
791. Addition & Subtraction (337)
796. Multiplication, Division (348)
797. Multiplication, etc. (224)
800. Higher Math (355)
801. Higher Math #2 (228)
810. Typing Practice (223)
815. Morse Code Teacher (155)
820. Health (354)
821. Health #2 (145)
830. Physics (111)
840. Nature (277)
850. Chemistry (277)
860. Astronomy (342)
861. Astronomy #2 (304)
870. Religion (346)
871. Religion #2 (42)
890. Teacher's Helpers (203)
900. Home Utilities (351)
901. Home Utilities #2 (342)
902. Home Utilities #3 (350)
907. Screen Drawing, Doodling (160)
909. High-Resolution Drawing (287)
910. Charts & Graphs (178)
912. Calculators & Converters (345)
913. Calculators & Convert. #2 (147)
915. Financial Math (339)
916. Financial Programs (356)
918. Checkbook Programs (203)
920. Business Programs (146)
950. Genealogy
970. Astrology, Numerology etc. (171)
980. Radio Utilities (220)
990. Sports Programs (329)
1100. Character & Sprite Editors (254)
1101. Programmer's Utilities (346)
1102. Sorts, Scrables, Searches (228)
1105. Auto-loaders (217)
1106. Disk Catalogers (268)
1107. Character Sets etc. (353)
1110. Assembly Utilities (357)
1111. Assembly Utilities, Routines (328)
1112. New Horizon Assembly Util. (269)
1119. Hardware Utilities (169)
1120. Sound Effects (197)
1130. Disk Labels & Jackets (284)
1131. Gemini Printer Utilities (224)
1132. Word Processing Utilities (182)
1133. Banners, Graphs, etc. (203)
1135. Speech Utilities & Demos (355)
1140. Music Composers (288)
1141. Assembly Music Compiler (265)
1145. Telecommunications Aids (342)
1150. Programming Tutorials (348)
1160. Assembly Tutorials #1 (231)
1161. Assembly Tutorials #2 (289)
1162. Assembly Tutorials #3 (357)
1163. Assembly Tutorials #4 (358)
1164. Assembly Tutorials #5 (340)
1300. Mathematical Games (133)
1301. Brain Games #1 (344)
1302. Brain Games #2 (345)
1303. Brain Games #3 (352)
1304. Brain Games #4 (352)
1305. Two-Player Brain Games (335)
1306. Brain Games #5 (345)
1307. Master Mind (322)
1310. Memory Games (235)
1315. Sargon Chess (155)
1320. Mazes #1 (342)
1321. Maze Games #2 (346)
1322. Maze Games #3 (338)
1330. Hangman Games (335)
1331. Wheel of Fortune #1 (249)
1332. Wheel of Fortune #2 (248)
1333. Word Games (310)
1340. Games by Roland Trueman (333)
1350. Card Games #1 (352)
1351. Card Games #2 (348)
1352. Card Games #3 (94)
1356. Dice Games (354)
1360. Board Games (321)
1361. Bingo (73)
1362. Checkers (238)
1363. Board Games #2 (287)
1367. Gambling Games (237)
1381. Bowling (289)
1382. Golf (138)
1383. Billiards, Boxing, etc. (250)
1400. Adventure Disk #1 (360)
1401. Adventure Disk #2 (306)
1402. Adventure Disk #3 (329)
1403. Adventure Disk #4 (324)
1415. Hammurabi Games (268)
1416. Text Games #1 (313)
1417. Text Adventures (340)
1425. Graphics/Text Adventures (354)
1426. Graphics/Text Adv. #2 (322)
1427. Graphics/Text Adv. #3 (325)
1430. Road Race Games (356)
1431. Keyboard Maneuvering (349)
1432. Road Crossing Games (344)
1433. Road Crossing Games #2 (175)
1434. Keyboard Games (347)
1435. Keyboard Maneuvering #2 (354)
1436. Slot Machines (343)
1437. Keyboard Games #2 (353)
1438. Keyboard Games #3 (343)
1440. QtBert Games (288)
1445. King Kong Type Games (351)
1455. Assembly Games (231)
1456. Assembly Games #2 (346)
1460. Children's Programs (345)
1461. Fun Games for Kids (353)
1462. Easy Games for Kids (346)
1470. Great Games (342)
1471. Assorted Games #1 (348)
1472. Assorted Games #2 (343)
1473. Texas Games Medley w/speech (346)
1474. Sea Battle Games (329)
1475. Joystick Games (342)
1476. Joystick Games #2 (355)
1477. Joystick Games #3 (346)
1478. Joystick Games #4 (338)
1479. Two-Player Joystick Games (353)
1480. Two-Player Keyboard Games (353)
1481. Joystick Games #5 (345)
1500. Kaleidoscopes & Displays (262)
1501. Sprite Displays (200)
1505. Poetry, Prose & Nonsense (128)
R

TI-99/4A ERROR CODE REFERENCE CHART

The following courtesy BBBB BBS in Clinton, MD. (301-292-1482), thanks to Bob, the sysop there:

TI BASIC ERROR CODES PERTAINING TO THE DISK SYSTEM

#:	First Digit	Second Digit	Description
0:	OPEN		Can't find specified disk drive
1:	CLOSE		Disk or Program is write protected
2:	INPUT		Bad Open Attribute
3:	PRINT		Illegal Operation
4:	RESTORE		Disk Full or too many files open
5:	OLD		Attempt to read past EOF
6:	SAVE		Device Error
7:	DELETE		File Error
8:	NIL		
9:	EOF		

EXTENDED BASIC ERROR CODES

10	Numeric Overflow
14	Syntax Error
16	Illegal After Subroutine
19	Name Too Long
20	Unrecognized Character
24	\$/# Mismatch
28	Improperly Used Name
36	Image Error
39	Memory Full
40	Stack Overflow
43	NEXT Without FOR
44	FOR-NEXT Nesting
47	Must Be In Subroutine
48	Recursive Subroutine CALL
49	Missing SUBEND
51	RETURN without GOSUB
54	String Truncated
56	Speech \$ Too Long
57	Bad Subscript
60	Line Not Found
61	Bad Line #
62	Line Too Long
67	Can't CONTINUE
69	Command illegal in Program
70	Only Legal in Program
74	Bad Argument
78	No Program Present
79	Bad Value
80	Nil
81	Incorrect Argument List
82	Nil
83	Input Error
84	Data Error
97	Protection Violation
109	File Error
138	I/O Error
135	Subroutine Not Found

I/O ERRORS

#:	First Digit	Second Digit	Description
0:	OPEN		Device Not Found
1:	CLOSE		Write Protected
2:	INPUT		Bad Open Attribute
3:	PRINT		Invalid I/O Command
4:	RESTORE		Out of Space
5:	OLD		End Of File
6:	SAVE		Device Error
7:	DELETE		File/Data Mismatch

DISK MANAGER ERROR CODES

#:	First Digit	Second Digit	Description
1:	OTHER		Rec not found
2:	SEEK/STEP		Cyclic Redundancy
3:	INPUT		Lost Data
4:	PRINT		Write Protect
5:	NIL		Write Fault
6:	NIL		No Disk Drive
7:	NIL		Invalid Input
8:	NIL		
9:			Special error code for Comprehensive tests.

EDITOR/ASSEMBLER ERROR CODES X/B ERROR EQUATES

ERRNO	>0200	2	Numeric Overflow
ERRSYN	>0300	3	Syntax Error
ERRIBS	>0400	4	Illegal after Subprgm
ERRNOS	>0500	5	Unmatched Quotes
ERRNTL	>0600	6	Name too long
ERRSNM	>0700	7	\$/# Mismatch
ORROBE	>0800	8	Option Base Error
ERRMUV	>0900	9	Improperly Used Name
ERRIM	>0A00	10	Image Error
ERRMEM	>0B00	11	Memory Full
ERRSO	>0C00	12	Stack Overflow
ERRNVF	>0D00	13	NEXT without FOR
ERRFNN	>0E00	14	FOR-NEXT Nesting
ERRSNS	>0F00	15	Must be In Subprogram
ERRRSC	>1000	16	Recursive Subprogram
ERRMS	>1100	17	Missing SUBEND
ERRRWG	>1200	18	RETURN without GOSUB
ERRST	>1300	19	String Truncated
WRRRBS	>1400	20	Bad Subscript
ERRSSL	>1500	21	Speech \$ Too Long
ERRLNF	>1600	22	Line Not Found
ERRBLN	>1700	23	Bad Line Number
ERRLT	>1800	24	Line Too Long
ERRCC	>1900	25	Can't CONTINUE
ERRCIP	>1A00	26	Illegal in Program
ERROLP	>1B00	27	Only Legal in Program
ERRBA	>1C00	28	Bad Argument
ERRNPP	>1D00	29	No Program Present
ERRBV	>1E00	30	Bad Value
ERRIAL	>1F00	31	Incorrect ArgumentList
ERRINP	>2000	32	Input Error
ERRDAT	>2100	33	Data Error
ERRFE	>2200	34	File Error
ERRIO	>2400	36	I/O Error
ERRSNF	>2500	37	Subprogram Not Found
ERRPV	>2700	39	Protection Violation
ERRINV	>2800	40	Unrecognized Character
WRNNO	>2900	41	Numeric Overflow
WRNST	>2A00	42	String Truncated
WRNPP	>2B00	43	No Program Present
WRNINP	>2C00	44	Input Error
WRNIO	>2D00	45	I/O Error

TI-WRITER ERROR CODES

0:	Indicates Disk Controller not on; or that Diskette is not initialized
6:	No Disk in Drive; or disk is upside down; or Drive is Not on.
7:	No Disk in Drive
00:	Illegal use of LoadF, PrintF, or an error in using those commands
02:	No file on Diskette with Filename as Used
04:	Diskette is Full
06:	PrintF Command in progress was interrupted, or: Disk Door Was Opened while red light was on
07:	Invalid Filename (I.E., Name too long or using invalid characters)
15:	Invalid Disk Drive Number, or Device

LOADER ERROR CODES

#:	First Digit	Second Digit	Description
0:	OPEN		Device Not Found
1:	CLOSE		Write Protected
2:	INPUT		Bad Open Attribute
3:	PRINT		Invalid I/O Command
4:	RESTORE		Out of Space
5:	OLD		End Of File
6:	SAVE		Device Error
7:	DELETE		File/Data Mismatch
8:			Memory Overflow
9:			Not Used
10:			Illegal Tag
11:			Checksum Error
12:			Unresolved Reference

HARDWARE TIPS

The Extended Basic Cartridges. If you have an Extended Basic Cartridge that has gone bad for some reason, this info may help. I killed my Ex Basic recently. While soldering some new "pieces parts" on my console main board, I crossed some wires. "I'm always in a hurry." When I turned the power on I already had the Ex Basic in the Grom Port. "Mistake!!!" I fed some current directly back into the Cartridge and zapped it. After a severe anxiety attack, and real depression, I thought, "HEY, this is the beginning of another project." Well, the project has been about two months in the making and I'm ready to let you all know the results. Extended Basic Cartridges are fixable and the parts may not cost a lot, depending on how bad you crashed it and if you can solder. One other consideration is this. If you have to replace all the chips in the cartridge it will cost around thirty dollars. You can probably pick up a used cartridge for around twenty. Since I'm out to put some mileage on my new soldering iron these things never bothered me. So, lets go. NOTE: You're doing this at your own risk. If you have any problems arising from this article, I don't want to hear about it.

Try to open the cartridge as neatly as possible. You can glue it back together later, but it would be better if the original snaps worked. When you get the PC board out you'll see eight chips.

There are two piggyback chips at one end of the board you won't see unless you have to unsolder them. The next thing to do is check the PC board and every solder joint to make sure that all the connections are good. If you find a cracked line or a poorly soldered leg on one of the chips, repairing it may solve all your problems. If we haven't accidentally found the problem so far, we'll move on to the heavier stuff. There should be a 74LS00 and a 74LS74 at one end of the board. If you are lucky, replacing these will put you back in business. Just de-solder them,

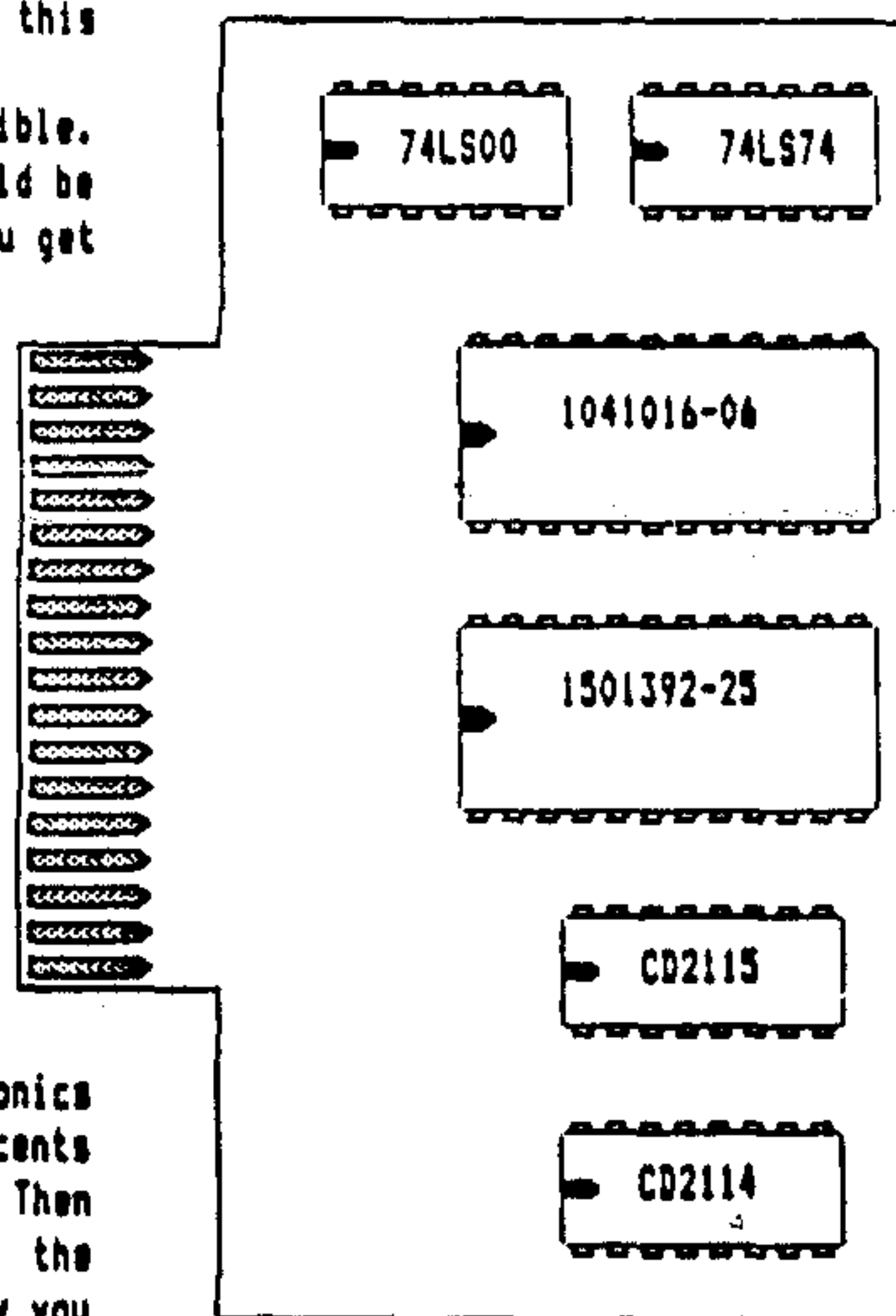
pick up two new ones at your local electronics supply, (they should cost less than fifty cents each), and solder the new chips back in. Then without bothering with the cartridge case, plug the board back into the console and see how lucky you are. In my case this was no help at all. The next step in this project is to replace the two large chips on the board. These are ROM chips and appear to be quick to fail in any adverse situation (static charge, etc.). The chip closest to the 74LS74 is listed as ROM,EXT.BASIC part number 1041016-0006, and the one next to it is ROM,EXT.BASIC part number 1501392-0025. Their prices are \$6.80 and \$5.60 respectively. These parts and others can be ordered from TI by calling (806)741-2265 or (806)741-2268. These are not toll free numbers. Replacing these two chips fixed my problem, and after doing a small amount of investigation, plus analyzing my own situation, it is my uneducated guess that replacing these four chips will fix the cartridge in at least 80% of the cases. I put a substantial charge of current back through the cartridge and did not hurt the piggyback ROM chips at the other end

of the board. So, replace the two large ROM chips and see if the cartridge works. I really hope it worked because we are not passing into the area where it would have been cheaper to pick up a good used Ex-Basic for \$20.00. If it still doesn't work, it's time to replace the GROMS. They are available from TI at the same phone number and they sell for \$3.60 each. The way they are tied together if one has been damaged they are probably all damaged. Toward the ROM, the top GROM is 2115, this is TI GROM,EXT.BASIC 1015960-3115. The one it is piggybacked over is 2122 (TI GROM,EXT.BASIC 1015960-1122). The top GROM next to it is 2114 (TI GROM,EXT.BASIC 1015960-1114), and the chip it is piggybacking is 2113 (TI GROM,EXT.BASIC 1015960-1113). If you are replacing the GROM chips, I recommend you do them all at once. If you try to add in one new chip at a time to isolate the problem, the soldering and desoldering could damage your new chips and you'll never find the problem. The resistor, an capacitors you see scattered around the printed circuit board will practically never fail, so don't worry about replacing that stuff. Here are some tips for electronic work. Use a low power soldering iron (15 Watt). Hold chips or a PC Board by the edges, like a photograph. Try to not put your fingerprints all

over the circuits or chip legs. Do not wear clothing that has caused you to get a static shock from the refrigerator door in the past. There is a notch or mark at one end of a chip to designate pin one, and the chip direction. Be sure you do not put a chip in backwards. Whenever you remove chips from a PC board use a vacuum type desoldering tool to remove all the solder from around the chip legs. "There have been times when I desoldered a leg, resoldered it, and then desoldered it again, in order to get a clean desolder job." Use long-nose pliers to wiggle and loosen every chip leg. If the legs are not all free and you pry the chip off the board, you will damage the board. The chip should be loose enough to almost pick it off with your fingers. When soldering any electronic part, do not heat the part with your iron and feed in the solder, this will over heat the chips. You should keep your iron clean. Hold the iron in one hand and the solder roll in

the other, with the item to be soldered on the table in front of you. Putting the end of the solder roll against the hot iron will accumulate a very small drop of molten solder on the end of the iron (don't do this directly over your project, place the iron against the part to be soldered for one or two seconds or until you see the molten solder flow around the wires or parts to be soldered. Do not hold the iron against the parts you are soldering any longer than necessary, and do not reheat a chip leg over and over. If you must replace the piggyback GROM chips, squeeze the legs of the top chip together until they fit tightly over the bottom chip, and then solder the chips together first. At that point solder the pair of chips to the board.

Have fun. Marty.



THE COMPUTER WIDOW

Getting Ready

Summertime and the livin' is easy. Right? Wrongo!!!! It's time once again for the Davies family annual migration. Between us Rich and I have about ten separate family units. No more than two of whom speak to any of the others at any given time. Each set of grandparents (there are four) feels that they must have an opportunity to spoil the kids. Since they all live within a hundred and fifty mile radius, it is our responsibility to bring the kids to them. Sooo off we go about once a year to make the rounds. Once we get there it's easy. The grandparents take over and aside from mediating a few minor squabbles over who gets us when, mommy is relieved of all duties. Getting us there is another story all together.

About a month before the trip, Rich starts thinking (out loud) about what needs to be done. Two weeks before he starts asking me why they aren't done. About a week before we leave, I start doing them. I must give him some credit though, any computer games he had started get finished. Any and all questions he's been asked over the year, get researched and answered. In the meantime, I make kennel reservations, wash and pack a months supply of clothes, pay all the bills, answer all correspondance, and generally go crazy. Then I look forward to thirty computer free days. Not this year though, my dad has been bitten by the computer bug. Should give my stepmom and I lots of conversation material. Right?

Since we are now in week zero, I've got to get packing. I'll write in September and let you know how it goes. Provided I can get Rich to come home from Daddy's.

Rich!!!!!! Put down that Micropendium and help me put this ninety pound suitcase on top of the car!!!!!!!!!!!!

Wish me luck?!

Annette Davies

JANUARY 9, 1990 HAPPY NEW YEAR !!!!!!!

MUNCH OFFICERS AND NUMBERS (all in 508 area unless noted)

President	W.C. Wyman	839-4134
Vice President	Bruce Willard	852/3250
Secretary	Jim Cox	
Treasurer	Jim Cox	869-2704
Acting Editor	Jim Cox	
Adv.Prog. Chair	Dan Rogers	248-5502
Library	Al/Lisa Cecchini	
Disk Librarian	Lou Holmes	617 965/3584
Tape Librarian	Walter Nowak	413 436/7675
+++++++	Jack Sughrue	476/7630

DECEMBER MEETING. The December meeting was probably the best one of the year. Lou Holmes brought Mike Wright from the B.C.S. and Mike gave a demonstration of some of the T.I. hardware and software which was never released. This included the CC40 computer and the hex peripheral system. Mike gave an excellent presentation, and I hope we can have him back later this year. Jack Sughrue demoed a new program called Cut-Throat Cribbage and it is excellent. We also had a demo of the new Gramulator which looks like it will be a popular piece of hardware. All in all this was an outstanding meeting.

JANUARY MEETING. This month's meeting should have two disks of the month demoed. Lou will demo last month's DOM which he didn't get a chance to do in December. Jack Sughrue is also expected to have the Disk of the Month for January which should include the Astronomy program he got from Mike Wright. The disks of the month are on sale for \$1.50 and I will give copies of the last three at the meeting.

~~BOYCOTT-COMPUTER SHOPPER.~~ I call on everyone in the TI community to boycott the Computer Shopper. Brian O'Brien talked to the new editor in New York and he reports that this man said that there will no longer be a TI Column in his magazine. This editor didn't seem to mind that his magazine will lose subscriptions. If you want to look up something in the magazine go to the library and read it for free.

RAFFLE. Every month we have a raffle to help defer the cost of the monthly hall rental. The number of prizes awarded depends on the number of tickets sold. This month we have a number of Norton Software games and utilities for prizes.

MONTHLY SALES. At each meeting you have the opportunity to buy and/or sell new or used hardware, software, books and original programs. Please have prices marked on any items you have to sell.

LIBRARY NOTICE. Please return any items borrowed from our library. If you can not come to a meeting or give these items to someone who will be at the meeting, please mail any library items to the group address which is listed on the cover of this newsletter. There are no late fees, we don't care how long they have been out, please return these items.

REPRINTS. Reprints of any items in this newsletter is permitted as long as credit is given to M.U.N.C.H.

ARTICLES. I am always looking for articles for this newsletter, anything which interests you will probably interest other members of the TI community, so please share your ideas and opinions with all of us.

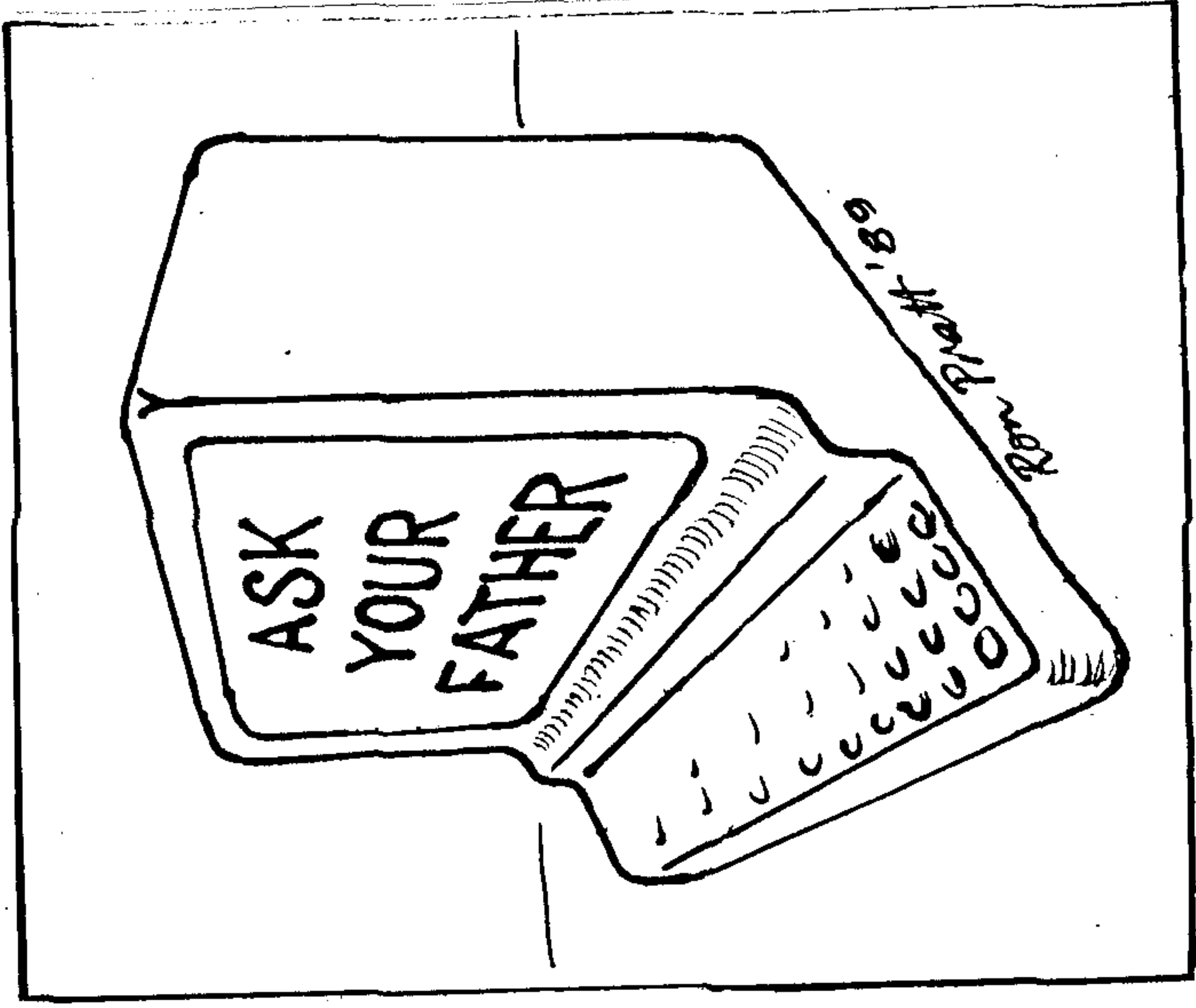
NEWSLETTER EXCHANGE EDITORS. Please note our corrected address on the front cover of this issue.

```

$ $ $ $ $ $ $ $ $ $ $ $
$ $ $ $ $ $ $ $ $ $ $ $
% % % % % % % % % % % %
+ + + + + + + + + + + +
# # # # # # # # # # # #
# # # # # # # # # # # #
* * * * * * * * * * * *
* * * * * * * * * * * *

```

Mass Users of the Ninety-nine and Computer Hobbyists
 JANUARY 1990 Monthly Newsletter Version 9.01



JOIN THE CROWD AT OUR JANUARY 9th MEETING

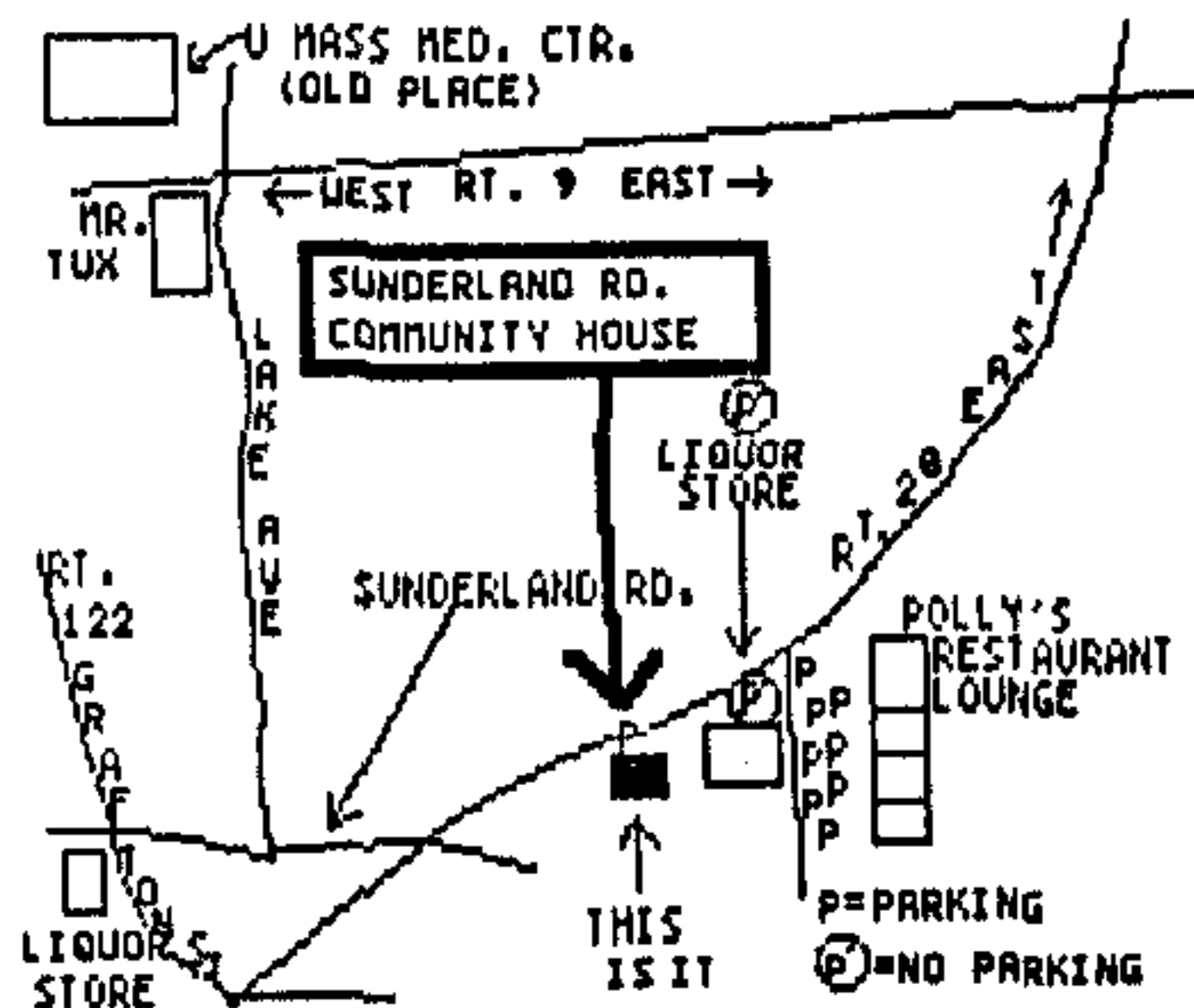
M.U.N.C.H.
 560 LINCOLN ST.
 P.O. BOX 7193
 WORCESTER, MA. 01605-7193



FIRST CLASS

!BOYCOTT COMPUTER SHOPPER!

Next Meeting JANUARY 9th.



POSTMASTER: Forwarding and Address Correction Requested.