QUICK REFERENCE HANDBOOK

TI USERS GROUP OF WILL COUNTY

Table of Contents

Note: As with any project, you assume complete liability. If you do not know what you are doing, do not attempt these projects. We cannot assume liability for your system should you perform any of these modifications.

The information contained in the QUICK REFERENCE HANDBOOK was gathered from various user groups newsletters and compiled by the TI USERS GROUP OF WILL COUNTY.

```
4A/TALK
DELETE FILE!   TE-II   !            !           !            !            !   HELP   !             !             !           !
 PRINT ON ! PRINT OFF !HALF/FULL DUP!BUFF ON/OFF!SAV/CLR BUFF! SEND ASCII !  CATALOG  !  CONFIGURE ! XMODEM XFER!AUTO-DIALER!    QUIT

CA8I-I200
 MODEM BAUD!PRNT ON/OFF! MODEM PARITY! MODEM PORT!PRINT PARITY! PRINT PORT ! PRINT BAUD!           !           !SCR'N WIDTH!  'QUIT
N,=SND FILE!B=DUMP BUFF! Y=CLEAR LOG !   CANCEL  !WINDOW RIGHT! K=SET TIMER! TEXT COLOR!SCREEN COLOR!,=LOG ON/OFF!FREEZE TOGL!   QUIT

F-TERM
TOGGL PRNTR!LOAD BUFFER!XMIT NXT LINE!XMT X/ONOFF!RESET BUFFER! DUMP BUFFER!RESTRT PROG!           !           !   QUIT   !
E=TOGL ECHO!           !             !           !            !            !SCR'N COLOR!           !           !          !

TE-II
 SPEAK!OUTPUT! CANCEL! TRANS!  WRAP  ! CASE  ! PAGE  !          !         ! EXIT  !
      !      !       !       !       !       !       !          !         !       ! QUIT

BASIC
      !        !        !       !       !        !       !         !       !       !
DELETE!INSERT! ERASE ! CLEAR! BEGIN !PROC'D!  AID  !  REDO  !  BACK  !        ! QUIT

TI-WRITER
 OOPS!     ! REFORMAT ! SCREEN COLOR! NEXT PARA !  DUPE LINE !  LAST PARA ! WORD TAB !  NEW PARA  !  NEW PAGE  ! WORD WRAP !
DELETE CHAR!INSERT CHAR! DELETE LINE ! ROLL DOWN ! NEXT WINDOW!   ROLL UP  !    TAB    ! INSERT LINE! COMMAND/ESC!  LINE #'S !   QUIT

MULTIPLAN
   HOME    !    TAB     !NEXT UNL CELL!FORW'D CHAR! FORW'D WORD! CHNG WINDOW!REL/ABS REF!           !           !           ! CANCEL
LOWER RIGHT!            !             ! BACK CHAR ! BACK WORD  !            !   HELP    !   RECALC  ! BACKSPACE !DEL FORWARD!

RAVE KYBD
  OOPS     ! REFORMAT ! SCREEN COLOR! NEXT PARA !  DUPE LINE !  LAST PARA ! WORD TAB !  NEW PARA  !  NEW PAGE  ! WORD WRAP ! LINE #'S
LOWER RIGHT!REL/ABS REF! FORWARD CHAR! BACK CHAR ! BACK WORD !FORWARD WORD!   HELP   ! RECALCULATE! NEXT WINDOW!NXT UNL CEL!RAVE 99/101

EDIT/ASM
DELETE CHAR!INSERT CHAR! DELETE LINE !  ROLL UP  ! NEXT SCREEN!  ROLL DOWN !    TAB    ! INSERT LINE!   ESCAPE   !           !   QUIT
  DELETE   !   INSERT   !    ERASE    !   CLEAR   !   BEGIN    !  PROCEED   !    AID    !    REDO    !    BACK    !           !   QUIT

FORTH
           !            !             !           !            !            !           ! INSERT LINE!            !           ! FORTH
  DELETE   !   INSERT   !    ERASE    !NEXT SCREEN! NEXT WINDOW! LAST SCREEN!DEL>ENDOLIN!  COPY LINE ! EXIT EDITOR!           !

DISKO
           !            !             ! FCTN E=UP ! FCTN X=DOWN! FCTN S=LEFT!FCTN D=RGHT!           !            !           ! DISKO
HEX DISPLAY!ASCII DISPL!LEAVE PROGRAM!BACK ! SCTR!RSTRT SUBPRG!FRWRD ! SCTR!           !REWRITE SCTR!RSTRT PROGRM!           ! QUIT

ADV/DIAG
E=PG/BUF UP!X=PG/BUF DN!             !           !            !            !           !            !            !           !ADV DIAGNOS
  DELETE   !   INSERT   !    ERASE    !NXT SCTR/TK! EXIT PROGRM!PREV SCTR/TK!   HELP    !RETES BADMEM!   ESCAPE   !SCREEN DUMP! ASCII/HEX

GRAPHX
       !       !        !        !         !        !        !         !         !        !GRAPHX
SLOWER!FASTER!  DRAW  ! ERASE!NO HELP! ZOOM  !COLORS! LINES  !CIRCLES! COPY   ! MENU

CHESS
 NO CONTROL! KEY PRESS !  REQUIRED  !           !1=MODE CHANG!           ! <=HELP !            !            !           ! CHESS
WITH FUNCTN!D=OFFR DRAW! 3=TIME OUT ! 4=ERASE   ! E=RESIGN  !P=ARANG POSN! S=SWITCH ! 8=REPLAY ! X=BACK UP !X=SAVE GAME! ==QUIT

CREDITS
COMPILED BY!            ! ED MACHONIS !           !            !           !            !           !            !           !
  PRINTED  !   USING    ! "PRINTSTRIP"!WRITTEN BY!    MIKE    !  MACHONIS  !     AS     !PUBLISHED IN!MICROPENDIUM!AUGUST 1986!
```

TI-WRITER COMMANDS

The following is a condensed listing of the TI-WRITER Text Editor commands. This was from Evy and Allen Anderson (Valley Chapter), published in K*3 Users Group Newsletter.

EDITOR COMMAND	FCTN	CTRL	EDITOR COMMAND	FCTN	CTRL	EDITOR COMMAND	FCTN	CTRL
BACK TAB		T	INS. BLANK LINE	8	O	QUIT	=	
BEGINNING / LINE		V	INSERT CHARACTER	2	G	REFORMAT		2orR
COMMAND/ESCAPE	9	C	LAST PARAGRAPH		6orH	RIGHT ARROW	D	D
DELETE CHARACTER	1	F	LEFT ARROW	S	S	ROLL DOWN	4	A
DEL. END OF LINE		K	LEFT MARGIN REL.		Y	ROLL UP	6	B
DELETE LINE	3	N	NEW PAGE		9orP	SCREEN COLOR		3
LINE #'s(on/off)	O		NEW PARAGRAPH		8orM	TAB	7	I
DOWN ARROW	X	A	NEXT PARAGRAPH		4orJ	UP ARROW	E	E
DUPLICATE LINE		5	NEXT WINDOW	5		WORD TAB		7orW
HOME CURSOR		L	OOPS!		1orZ	WORD WRAP/FIXED		O

<u>LOAD_FILES</u> = LF (enter) DSK1.FILENAME (load entire file)
LF (enter) 3 DSK1.FILENAME (merges filename with data in memory after line 3)
LF (enter) 3 1 10 DSK1.FILENAME (lines 1 thru 10 of filename are merged after line 3 in memory)
LF (enter) 1 10 DSK1.FILENAME (loads 1 thru 10 of filename)

<u>SAVE_FILES</u> = SF (enter) DSK1.FILENAME (save entire file)
SF (enter) 1 10 DSK1.FILENAME (saves lines 1 thru 10)

<u>PRINT_FILES</u>= PF (enter) PIO (prints control character and line numbers)
PF (enter) C PIO (prints with no control characters)
PF (enter) L PIO (prints 74 characters with line numbers)
PF (enter) F PIO (fixed 80 format)
PF (enter) 1 10 PIO (prints lines 1 thru 10)

Note: If your printer uses RS232 switch PIO with RS232.
To cancel the print command press FCTN 4.

<u>DELETE_FILE</u>= DF (enter) DSK1.FILENAME

<u>SETTING_MARGINS_AND_TABS</u> (16 tabs maximum)
L - Left margin R - Right margin I - Indent T - Tab
Use ENTER to execute or COMMAND/ESCAPE to terminate command.

<u>RECOVER_EDIT</u>= RE (enter) Y or N

<u>EDIT</u> = E (enter) (enter edit mode)

<u>LINE_MOVE</u> = M (enter) 2 6 10 (moves lines 2 thru 3 after line 10)
M (enter) 2 2 10 (moves line 2 after line 10)

<u>COPY</u> = Same as move except use C instead of M.

<u>FIND_STRING</u> = FS (enter) /string/ (will find string)
FS (enter) 1 15 /string/ (will find string in lines 2 thru 15)

<u>DELETE</u> = D (enter) 10 15 (deletes line 10 thru 15)

PRINTER COMMANDS

(energizes or turns on)

	10X	SG-10	MX-80	FX-80	KX-P1091	OKIDATA
ITALICS	27 52	27 52	##########	27 52	27 52	##########
ELITE	27 66 2	27 66 2	##########	27 77	27 77	28
CONDENSED	27 15	27 15	27 15	27 15	27 15	29
PICA	27 66 1	27 66 1	##########	##########	27 80	30
EXPANDED	27 87 1	27 87 1	27 14	27 87 1	27 87 1	31
SUPERSCRIPT	27 83 0	27 83 0	##########	27 83 0	27 83 0	27 74
SUBSCRIPT	27 83 1	27 83 1	##########	27 83 1	27 83 1	27 76
NEAR LETTER	##########	27 65 4	##########	27 120 1	27 110	27 49
EMPHASIZED	27 69	27 69	##########	27 69	27 69	27 84
UNDERLINE	27 45 1	27 45 1	##########	27 45 1	27 45 1	27 67
DOUBLE STRIKE	27 71	27 71	27 71	27 71	27 71	27 72
SLASHED ZERO	##########	27 92 1	##########	##########	##########	##########
1/8 LINE SP.	27 48	27 48	27 48	27 48	27 48	27 56
1/6 LINE SP.	27 50	27 50	27 50	27 50	27 50	27 54
7/72 LINE SP.	27 49	27 49	27 49	27 49	27 49	##########
n/72 LINE SP.	27 65 n	27 65 n	27 65 n	27 65 n	##########	##########
n/144 LINE SP.	27 51 n	27 51 n	##########	##########	##########	27 37 57 n
n/216 LINE SP.	##########	##########	##########	27 51 n	##########	##########
TOP MARGIN	27 82 n	27 82 n	##########	##########	##########	##########
BOTTOM MARGIN	27 78 n	27 78 n	27 78 n	27 78 n	##########	##########
LEFT MARGIN	27 77 n	27 77 n	##########	27 108 n	##########	##########
RIGHT MARGIN	27 81 n	27 81 n	##########	27 81 n	##########	##########
COLUMN WIDTH	##########	##########	27 81 n	##########	##########	##########
PAGE LTH. LINES	27 67 n	27 67 n	27 67 n	27 67 n	##########	##########
PAGE LTH. INCHES	27 67 0 n	27 67 0 n	##########	27 67 0 n	##########	##########
PAPER OUT "OFF"	27 56	27 56	27 56	27 56	27 56	##########
PROPORTIONAL	##########	27 112	##########	27 112	27 111	##########
RESET PRINTER	27 64	27 64	##########	27 64	##########	24

- GRAPHICS COMPATABILITY -

by Don Macclellan
BLUEGRASS 99 COMPUTER SOCIETY, INC.

The multitude of Graphics programs available for use with the TI-99/4A computer and their compatability with word processing programs has prompted a request for some description of the available programs. This will be an attempt to clarify compatability among most of the later programs. The diagram on the right does not cover everything available, but does cover all of the programs which have been sold through the Society or which are available in the Society library. This discussion is not an attempt to provide a tutorial or review of such a multitude of software, only to give you a perspective and hint of what might suit your needs.

TI-WRITER is the only prudent choice for a word processing program for those having 32K & Disk drive. There are no others which come close to providing the features and versatility. 'Companion' is probably the logical alternate if WRITER were not so available. All versions currently available (and there are several) still use TI's WRITER program files. The best and least costly is FUNNELWEB's version which in addition to freeing E/A & WRITER from their respective modules, includes -C, DISKO, FORTH, FASTTERM, and your choice of any others in a disk-based program which really begins to shine when installed on a RamDisk. Almost all of the programs which we will discuss will be used either with Text from TI-WRITER or through TI-WRITER.

The CSGD Series of Graphics programs written by Dave Rose has been more widely accepted by our members and is shown at the top of the diagram. It is compatable with both Prowriter and Epson-Gemini printers but you must purchase the correct version for your printer. The Keyboard or the JoyStick may be used in any of the Editors. The Message printing program is common to CSGD-I,II, & III.

CSGD-I contains the EDITOR programs for creating your own (1) Character Sets, (2) 5x5 Graphics, and (3) Pictures. The Editors, primarily the Character Editor, have undergone 4 revisions including the most recent which was in response to suggestions by our members. Provisions are also made to jockey graphics around to convert between alternate printer types.

CSGD-II is basically a banner program which prints the message sideways and 8 times magnified. It also contains the Graphic Editor but not the other editors. It has gone through three revisions including the latest which now allows printing lower case in the Banner mode.

CSGD-III is primarily a Label program which produces multi-width labels in 3 heights. It also contains the Message program and a 'easier to operate' Letterhead program. ALL OF THE FILES CREATED IN CSGD ARE I/V 254 and are not compatable with TI-WRITER. The Docuprinter is compatable with WRITER and will produce 1 or 2 column texts using a choice of 6 type fonts and D/V 80 files through the Formatter. It will not presently handle fonts of greater height than 1 row.

The CSGD programs are supported by a multitude of graphics, pictures, and over 100 fonts written by Dave Rose and contributors who use and enjoy his programs. Three of our members have contributed Charater Fonts. These have been issued as **UD1**(1), **UD2**(2), **UD3**(2). **UD4**(2) will be available at the meeting !!!

TI-ARTIST is an extremety versatile drawing program written by a talented young man, Chris Faherty, which many of the Society members have purchased. It allows creating, loading and modifying, size changes and many other features using the Keyboard or a combination of the Keyboard and Joystick.

NONE OF THE PROGRAMS DESCRIBED IN THIS DISCUSSION CAN BE LEARNED READING THE MANUAL. You must use them and make your mistakes: It helps you to remember. TI-ARTIST is in its second revision which now includes Prowriter as well as Epson-Gemini versions. The 2.01 RamDisk configured version is also now available.

The only Files that were available with ARTIST were **ART-EXTRAS** however, Dave Rose has converted his I/V 254 files to D/V 80 files which can be loaded as 'Instances' and they have been released as the **COMPANION I-III** series. IV should follow soon since UD4 has now been released. In addition, TI-ARTIST allows the loading of GRAPHX files so that a rather large base of graphics is available to use and modify or create you own.

GRAPHX is a program with quite a few similarities to ARTIST; was written in Australia and introduced in the USA in 1984. It allows use of the Joystick ONLY, and is not compatible with any but the Epson printer. It is a quite versatile drawing board in the hands of an experienced user.

The support graphics available consist of **GRAPHX COMPANION-I thru III** and **GRAPHX PICTURES**. They are all well done though I personally have not purchased the program because of the printer limitation and my distaste for the Joystick. The availability of additional graphics is made possible through TI-ARTIST which will load and save to the Image (PGM) format. Several of our members prefer to do their creative work with GRAPHX and use ARTIST only as a transport.

JOYPAINT & JP PAL are Graphics Drawing Boards well spoken of by their owners. I am not aware of any Society members who own this 'Joystick' only Program. It does have options which allow loading from and saving to ARTIST &/or GRAPHX filetypes. It is currently only compatible with the EPSON printer.

ARTCONVERT is a Program which we purchased in Chicago that converts TI-ARTIST files to TI-WRITER files. It is supported by 4 diskettes of files; **ARTDATA-I** thru **IV**. This permits anyone with TI-WRITER to have the ability to print graphics. It will also merge and print two graphics but there are no provisions to include text in the graphic file to complete a document.

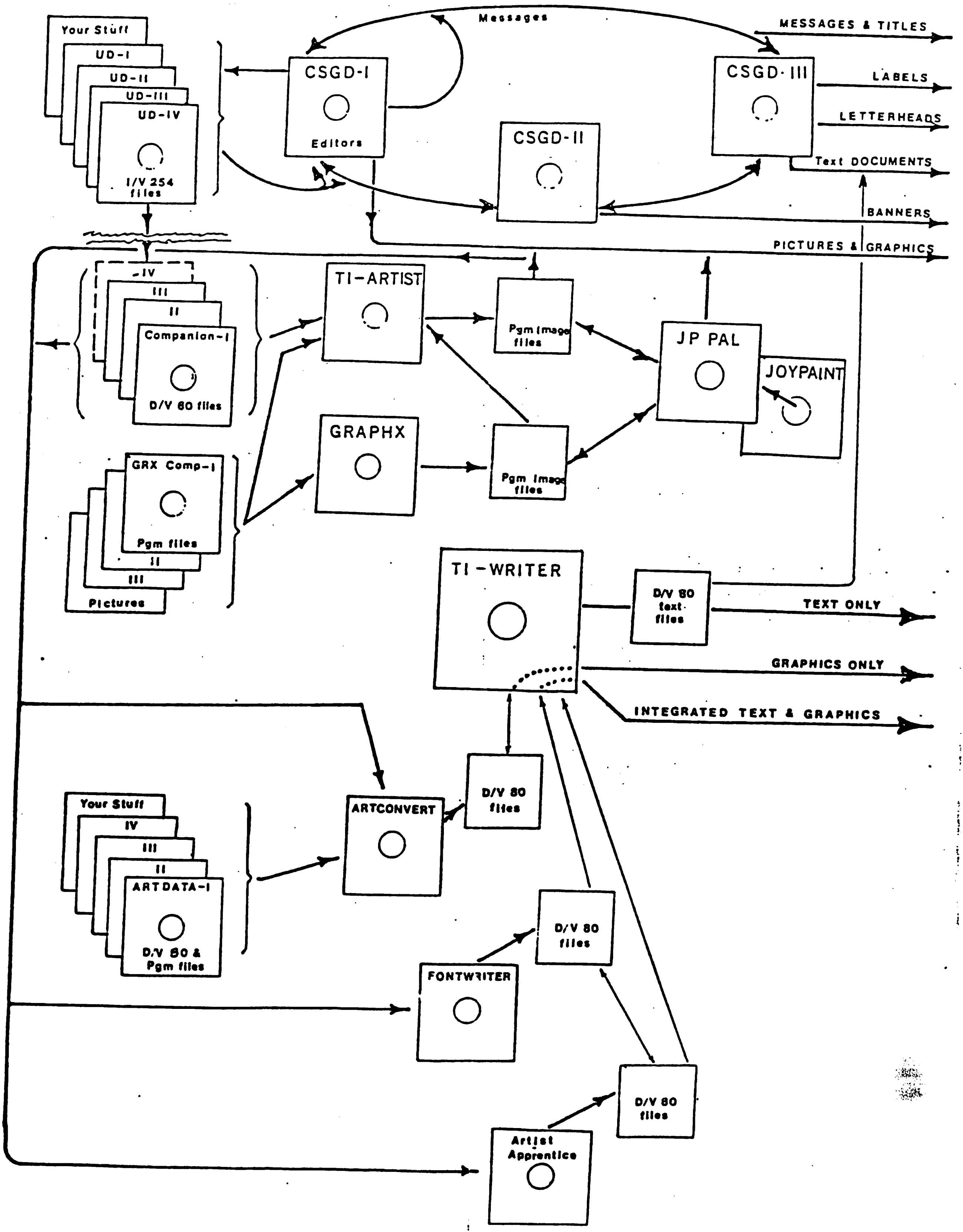
One unique feature of ARTCONVERT allows the user to convert one row high TI-ARTIST fonts for use thru TI-WRITER. This would allow conversion of all of Rose's 1 row high fonts in Companion I-III to be used though they contain only 78 of 96 Typewriter keys if you do not have the COMPANION series. Compatible with Prowriter and Epson-Gemini.

FONTWRITER by Peter Hoddie is really the first graphics program which is compatable with TI-ARTIST and TI-WRITER. With this program you can create your own type fonts, revise TI-ARTIST and CSGD fonts and use TI-ARTIST Instances mixed in printed documents even on the same line.

Although I bought it in Chicago based on Asgard's assurance that it would work on the Prowriter, the second paragraph of the documentation states that it strictly for Epson-Gemini owners. In a late note Hoddie added a brief text to the diskette which told of the location of the printer codes and I think it can easily be made to work with the better printers. The resources for this program are as broad as all of the ARTIST files plus all of those I have indicated earlier which can be converted through ARTIST.

ARTIST APPRENTICE is similar in several ways to FONTWRITER. It allows use of TI-ARTIST type files, fonts and graphics to produce files which are printed according to a 'Scheduler'. It is limited in being compatable only with the EPSON printer.

SOME POPULAR GRAPHICS PROGRAMS



THIS QUICK REFERENCE CHART SHOWS CSGD CHARACTER SET DATA, INCLUDING: HEIGHT, UPPER AND LOWER CASE AVAILABILITY, NUMBER AVAILABILITY, AND PUNCTUATION AVAILABILITY. ITWAS CREATED BY MR. ALLAN COX, TARRANT, AL.

CSGD CHAR SET DATA

FILE	HT.	UP.	LO.	NO.	.	,	"	!	@	-	:
3D	4	X		X	X	X	X	X	X	X	X
ABC	3	X									
AMBROSA	4	X	X	X	X	X	X	X	X	X	X
ATHEN	2	X	X	X	X	X	X	X	X	X	X
BANNER	5	X		X	X	X	X	X	X	X	X
BOLD	2	X		X	X	X	X	X	X	X	X
BSBOLD	4	X									
BWAY	2	X		X	X	X	X	X	X	X	X
CASLON	3	X		X	X	X	X	X	X	X	X
CHOCHO	3	X		X	X	X	X	X	X	X	X
CURSIVE	4	X	X	X	X	X	X	X	X	X	X
FANCY	3	X		X	X	X	X	X	X	X	X
FAREAST	4	X		X	X	X	X	X	X	X	X
FAST	2	X		X	X	X	X	X	X	X	X
FAT	4	X		X	X	X	X	X	X	X	X
FBLOCK	3	X									
FILM	4	X		X	X	X	X	X	X	X	X
FROZEN	4	X									
GOTHIC	4	X									
GOTHIC2	4	X		X	X	X	X	X	X	X	X
HEBREW	4	X		X	X	X	X	X	X	X	X
HOLLOW	5	X	X	X	X	X	X	X	X	X	X
HOLLOW2	3	X		X	X	X	X	X	X	X	X
IBM	3	X		X	X	X	X	X	X	X	X
LBLOCK	3	X		X	X	X	X	X	X	X	X
LED	4	X									
LEDGER	4	X		X	X	X	X	X	X	X	X
MACBETH	4	X									
MONOGRM	3	X		X	X	X	X	X	X	X	X
NBOAT	3	X		X	X	X	X	X	X	X	X
NINETY	4	X		X	X	X	X	X	X	X	X
OBLONG	5	X		X	X	X	X	X	X	X	X
OFFBEAT	1	X		X	X	X	X	X	X	X	X
OLDENG	4	X	X	X	X	X	X	X	X	X	X
OLDENG2	4	X		X	X	X	X	X	X	X	X
OLDENG3	4	X	X	X	X	X	X	X	X	X	X
PCSET	1	X									
PCSET1	1	X		X	X	X	X	X	X	X	X
PCSET2	2	X		X	X	X	X	X	X	X	X
PCSET3	3	X		X	X	X	X	X	X	X	X
PCSET4	2	X	X	X	X	X	X	X	X	X	X
PLAIN	2	X		X	X	X	X	X	X	X	X
PLANE	3	X		X	X	X	X	X	X	X	X
PLAYBIL	4	X									
PSET	3	X									
RCSET1	1	X	X	X	X	X	X	X	X	X	X
RCSET2	1	X	X	X	X	X	X	X	X	X	X
RCSET3	1	X	X	X	X	X	X	X	X	X	X
RCSET4	1	X	X	X	X	X	X	X	X	X	X
RCSET5	1	X	X	X	X	X	X	X	X	X	X
RCSET6	1	X	X	X	X	X	X	X	X	X	X
RCSET7	1	X	X	X	X	X	X	X	X	X	X
RODEO	2	X		X	X	X	X	X	X	X	X
RODEO2	3	X		X	X	X	X	X	X	X	X
ROMAN	1	X		X	X	X	X	X	X	X	X
ROUND	1	X		X	X	X	X	X	X	X	X
SBLOCK	3	X		X	X	X	X	X	X	X	X
SCRIPT1	3	X		X	X	X	X	X	X	X	X
SCRIPT2	2	X		X	X	X	X	X	X	X	X
SCRIPT3	2	X		X	X	X	X	X	X	X	X
SCRIPT4	4	X	X	X	X	X	X	X	X	X	X
SCRIPT5	3	X		X	X	X	X	X	X	X	X
SHADOW	5	X		X	X	X	X	X	X	X	X
SHADOW2	4	X		X	X	X	X	X	X	X	X
SHELBY	4	X									
SLANT	1	X		X	X	X	X	X	X	X	X
SQUARE	1	X		X	X	X	X	X	X	X	X
STENCIL	3	X		X	X	X	X	X	X	X	X
STRIPE2	4	X		X	X	X	X	X	X	X	X
TECH	3	X		X	X	X	X	X	X	X	X
TECH2	1	X		X	X	X	X	X	X	X	X
VETICA2	3	X	X	X	X	X	X	X	X	X	X

ERROR CODE REFERENCE SHEET...

The following is reprinted from the Central Texas 99-4A users groups newsletter who reprinted it from the newsletter of the Milwaukee Area Users group.

EXTENDED BASIC

10	Numeric Overflow
14	Syntax Error
16	Illegal after Sbrtn
19	Name too long
20	Unrecognized Char
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 Sbrtn
48	Recursive Sbrtn 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 prgrm
70	Only legal in prgrm
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
130	I/O Error
135	Sbrtn not found

DISK MANAGER ERROR CODES

#:	First #	Second #
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 Test	

I/O ERRORS

#:	FIRST #	SECOND #
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	EOF
6:	SAVE	Device Error
7:	DELETE	File/Data Mismatch

TI BASIC ERROR CODES PERTAINING TO DISK SYSTEM

#:	FIRST #	SECOND #
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 opened
5:	OLD	Attempt to read past EOF
6:	SAVE	Device Error
7:	DELETE	File Error
9:	EOF	

EDITOR/ASSEMBLER ERROR CODES

X.B. ERROR EQUATES

ERRNO	)0200	2	Numeric Overflow
ERRSYN	)0300	3	Syntax Error
ERRIBS	)0400	4	Ill. after Sbprgm
ERRNQS	)0500	5	Unmatched Quotes
ERRNTL	)0600	6	Name too long
ERRSNM	)0700	7	$/# Mismatch
ERROBE	)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
ERRNWF	)0D00	13	NEXT without FOR
ERRFNN	)0E00	14	FOR-NEXT nesting
ERRSNS	)0F00	15	Must be in Sbprgm
ERRRSC	)1000	16	Recursive Sbprgm
ERRMS	)1100	17	Missing SUBEND
ERRRWG	)1200	18	RETURN without GOSUB
ERRST	)1300	19	String truncated
ERRRBS	)1400	20	Bad subscript
ERRSSL	)1500	21	Speech $ too long
ERRLNF	)1600	22	Line not found
ERRBLN	)1700	23	Bad line number
ERRLTL	)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 argument list
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
WRNNPP	)2B00	43	No program present
WRNINP	)2C00	44	Input error
WRNIO	)2D00	45	I/O error

LOADER ERROR CODES

0-7	Standard I/O
8	Memory overflow
9	Not used
10	Illegal tag
11	Checksum error
12	Unresolved ref.

EXECUTION ERRORS

0-7	Standard I/O
08	Memory Full
09	Incorrect Statement
0A	Illegal Tag
0B	Checksum Error
0C	Dup. Definition
0D	Unresolved Ref.
0E	Incorrect Statement
0F	Program not found
10	Incorrect Statement
11	Bad Name
12	Can't CONtinue
13	Bad Value
14	Number too big
15	String-Number
16	Bad Argument
17	Bad Subscript
18	Name Conflict
19	Can't do that
1A	Bad Line Number
1B	FOR NEXT Error
1C	I/O Error
1D	File Error
1E	Input Error
1F	Data Error
20	Line too long
21	Memory Full
22-	Unknown Error Code

RS232c ERRORS

OPEN:	00	Device cannot be opened
	02	Software Switch Error
	06	Hardware Error
INPUT:	24	Internal Data too large for buffer
	26	'CLEAR' pressed or Hardware Error
PRINT:	36	'CLEAR' pressed or Hardware Error
OLD:	50	Can't load from specified device
	52	Can't use software switch with 'OLD'
	54	Program too large to load
	56	'CLEAR' pressed or Hardware Error
SAVE:	60	Can't save to specified device
	62	See 02. Can't use with SAVE
	66	'CLEAR' pressed or Hardware Error
MISC:	43,73,83,93,	Executing Illegal Command

TI WRITER ERROR CODES

0	- Indicates Disk Controller not on; OR: Diskette not Initialized
6	- No Disk in Drive; OR: Is upside down; OR: Drive is not turned on
7	- No Disk in Drive
00	- Illegal use of LoadF, PrintF; OR: Error in using those commands
02	- No file in Diskette with Filename used
04	- Disk 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

PEEKS, POKES, AND LOADS....

Submitted by: Jerry Frattini

Here is a rather extensive listing of the PEEKS, POKES, and LOADS for the 99/4A. Jerry picked this up from a BBS in Boston so I don't know who gets the credit for compiling this list. These require X-BASIC and 32K memory Expansion. Remember to do a CALL INIT first. The P and Q variables are used for "PEEK" - the numbers are for "POKE" or "LOAD".

ADDRESS	VALUE(S)	MEANING IN EXTENDED BASIC
8192	P	USE (PEEK,P) IF P<> 70 OR <>121 THEN DO A CALL INIT
8194		FIRST FREE ADDRESS IN LOW MEMORY
8196		LAST FREE ADDRESS IN LOW MEMORY
-28672	P	P=0 SPEECH NOT ATTACHED P=96 OR P=255 SPEECH IS ATTACHED
-31572	0 TO 255	VARY KEYBOARD RESPONSE
-31740	P , Q	PUT IN DIFFERENT TO CHANGE BEEPS,WARNINGS, ETC
-31744	0 TO 15	CONTINUATION OF LAST SOUND (0=LOUD AND 15=SOFT)
-31748	0 TO 255	CHANGE THE CURSOR FLASHING AND RESPONSE TONE RATES
-31788	160	BLANK OUT THE SCREEN (MUST PUSH A KEY TO ACTIVATE)
	192	NO AUTOMATIC SPRITE MOTION OR SOUND
	224	NORMAL OPERATION
	225	MAGNIFIED SPRITES
	226	DOUBLE SIZE SPRITES
	227	MAGNIFIED DOUBLE SIZED SPRITES
	232	MULTICOLOR MODE (48 BY 64 SQUARES)
-31794	P	TIMER FOR CALL SOUND (COUNTS FROM 255 TO 0)
-31804	X , Y	RETURN TO THE TITLE SCREEN (USE "PEEK (2,X,Y)")
	P	CHANGE THE CURSOR FLASH RATE (0 TO 255)
-31806	0	NORMAL OPERATION
	16	DISABLE QUIT KEY (FCTN =)
	32	DISABLE SOUND (USE NEG DUR FOR CONTINOUS SOUND)
	48	DISABLE SOUND QUIT KEY
	64	DISABLE AUTO SPRITE MOTION
	80	DISABLE SPRITES QUIT KEY
	96	DISABLE SPRITES AND SOUND
	128	DISABLE ALL THREE
-31808	P , Q	DOUBLE RANDOM NUMBERS (0 TO 255) NEED "RANDOMIZE"
-31860	4	GO FROM EX-BASIC TO CONSOLE BASIC (NEED "NEW")
	8	AUTO RUN OF DSK1.LOAD
-31866	P , Q	END OF CPU PROGRAM ADDRESS (P6+Q)
-31868	0	NO "RUN" OR "LIST" AFTER "BREAK" IS USED
	0 , 0	TURNS OFF THE 32K MEMORY EXPANSION
	255 , 231	TURNS ON THE 32K MEMORY EXPANSION
-31873	3 TO 30	SCREEN COLUMN TO START AT WITH A "PRINT"
-31877	P	P/32 = SPRITE COINCIDENCE P/64 = 5 SPRITES ON A LINE
-31878	P	HIGHEST NUMBER SPRITE IN MOTION (0 STOPS ALL)
-31879	P	TIMER FOR VDP INTERRUPTS EVERY 1/60 OF A SEC (0 TOP 255)
-31880	P	RANDOM NUMBER (0 TO 99) NEED "RANDOMIZE"
-31884	0 TO 5	CHANGE KEYBOARD MODE (LIKE "CALL KEY(K,...)")
-31888	63 , 255	DISABLE ALL DISK DRIVES (USE "NEW" TO FREE MEMORY)
	55 , 215	ENABLE ALL DISK DRIVES (USE "NEW" TO FREE DRIVES)

*** CONTINUED ON NEXT PAGE ***

PEEKS, POKES AND LOADS (cont.)....

ADDRESS	VALUE(S)	MEANING
-31931	0	UNPROTECT X-B PROTECTION
	2	SET "ON WARNING NEXT" COMMAND
	4	SET "ON WARNING STOP" COMMAND
	14	SET "UNTRACE" COMMAND
	15	SET "UNTRACE" COMMAND "NUM" COMMAND
	16	SET "TRACE" COMMAND
	64	SET "ON BREAK NEXT" COMMAND
	128	PROTECT X/B PROGRAM
-31952	P	PEEK P=55 THEN 32K EXPANSION MEMORY IS OFF <>55 MEANS ON
-31962	32	RETURN TO THE TITLE SCREEN
	255	RESTART X/B W/DSK1.LOAD
-31974	P , Q	END OF VDP STACK ADDRESS (P6+Q)
-32112	8	SEARCHES DISK FOR ?
-32114	2	RANDOM GARBAGE
	13	SCREEN GOES WILD
	119	PRODUCE LINES
-32116	2	RANDOM CHARACTERS ON SCREEN
	4	GO FROM X/BASIC TO BASIC
-32187	0	UNPROTECT XB PROGRAM
	2	SET "ON WARNING NEXT" COMMAND
	4	SET "ON WARNING STOP" COMMAND
	9	SET 0 LINE NUMBER
	14	SET "UNTRACE" COMMAND
	15	SET "UNTRACE" COMMAND "NUM" COMMAND
	16	SET "TRACE" COMMAND
	64	SET "ON BREAK NEXT" COMMAND
	128	PROTECT XB PROGRAM
-32188	1	CHANGE COLOR AND RECEIVE SYNTAX ERROR
	127	CHANGE COLOR AND RECEIVE BREAKPOINT
-32630	128	RESET TO TITLE SCREEN
-32699	0	UNPROTECT XB PROGRAM
	2	SET "ON WARNING NEXT" COMMAND
	4	SET "ON WARNING STOP" COMMAND
	14	SET "UNTRACE" COMMAND
	15	SET "UNTRACE" "NUM" COMMAND
	16	SET "TRACE" COMMAND
	64	SET "ON BREAK NEXT"
	128	PROTECT XB PROGRAM
-32700	0	CLEARS CREEN FOR AN INSTANT
-32729	0	RUN "DSK1.LOAD"
-32730	32	RESET TO TITLE SCREEN
-32961	51	RESET TO TITLE SCREEN
	149	SETS "ON BREAK GOTO" LOCKS SYSTEM

Keyboard Conversion

courtesy TOM FREEMAN

49	50	51	52	52	54	55	56	57	48	61	SHIFT UP
33	64	35	36	37	94	38	42	40	41	43	SHIFT DOWN
1	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*	*0*	*=*	
3	4	7	2	14	12	1	6	15	188	5	FTCN
177	178	179	180	181	182	183	158	159	176	157	CTRL

113	119	101	114	116	121	117	105	111	112	47
81	87	69	82	84	89	85	73	79	80	45
Q	*W*	*E*	*R*	*T*	*Y*	*U*	*I*	*O*	*P*	*/*
197	126	11	91	93	198	95	63	39	34	186
145	151	133	146	148	153	149	137	143	144	187

97	115	100	102	103	104	106	107	108	59	13
65	83	68	70	71	72	74	75	76	58	13
A	*S*	*D*	*F*	*G*	*H*	*J*	*K*	*L*	*;*	*ENTER*
124	8	9	123	125	191	192	193	194	189	13
129	147	132	134	135	136	138	139	140	156	13

	122	120	99	118	98	110	109	44	46	
	90	88	67	86	66	78	77	60	62	
SHIFT	*Z*	*X*	*C*	*V*	*B*	*N*	*M*	*,*	*.*	*SHIFT*
	92	10	96	127	190	196	195	184	185	
	154	152	131	150	130	142	141	128	155	

	32	
	32	
CTRL	*SPACE*	*FCTN*
	32	
	32	

** DISK MAPPING **

Ever wonder how your computer knows where to find a program on a disk or how and where that program is put? Well, I don't have all the answers but listing below can certainly help you arrive at some answers. I don't know just where all this data came from (I got it out of Florida) but suspect the original source was a T.I. data book. If you have DPATCH, DISK FIXER, SUPERBUGGER, or any other disk repair/fix program, then this should be a useful road map to your disk.

```
HEX       DEC       CONTENTS
--------- --------- ----------------------

SECTOR_0_-_Disk_Information

0000-0009 0000-0009 Disk Name - up to 10 characters
000A-000B 0010-0011 Total number of sectors on disk
                      >0168=360, >02D0=720
```

000C	0012	Number of sectors per track >09=9
000D-000F	0013-0015	>44534B="DSK"
0010	0016	>50="P" (backup protected) >20=" " (not protected)
0011	0017	Number of tracks >28=40
0012	0018	Number of sides >01=1 (single) >02=2 (double)
0013	0019	Disk density >01=1 (single) >02=2 (double)
0014-0038	0020-0055	Not used
0038-0064	0056-0100	Bit map of all disk sectors (see note)
0065	0101	Not used
0066-0092	0102-0146	Bit map of all disk sectors (see note)
0093	0147	Not used 0094-00C0 0148-0192 Bit map of all disk sectors (see note)
00C1	0193	Not used
00C2-00EE	0194-0238	Bit map of all disk sectors (see note)
00F0-00FF	0240-0255	Not used

NOTE: The usage of these maps depends on if the disk is SS, DS, SD, or DD. Take one word at a time, and split it into bytes. Take a byte at a time and split it into bits. Reverse the order of the bits in this byte. 1=sector used, 0=sector not used. e.g. >0038=F300 this converts to 1111 0011 0000 0000. Take the first byte (1111 0011), and reverse the order (1100 1111). This means that sectors 0, 1, 4, 5, 6, and 7 are used; sectors 2, and 3 are not.

SECTOR 1 - Directory Link

0000-0001	0000-0001	The sector of the 1st directory in alphabetical order
0002-0003	0002-0003	The sector of the 2nd directory in alphabetical order
"	"	
"	"	
00FC-00FD	0252-0253	The sector of the 127th directory in alphabetical order
00FE-00FF	0254-0255	The sector of the 128th directory in alphabetical order

SECTORS 2-22 - File Header

Headers for the first 19 files are placed here by default. If the disk contains more then 19 files, additional file headers are placed in the first available sector.If there are fewer then 19 files but they fill the disk, the sectors in this area may be used to contain last few sectors of the last file.

0000-0009	0000-0009	File name		
000A-000B	0010-0011	Not used		
000C	0012	File type		
		bit 0	0=fixed	1=variable
		bit 4	0=none	1=write protected

		bit 6 0=display 1=internal bit 7 0=data 1=program file
000D	0013	Number of records per sector (n/a for program)
000E-000F	0014-0015	Number of sectors per file
0010	0016	End of file offset in last sector (n/a for fixed file)
0011	0017	Record length of files
0012-0013	0018-0019	Number of records per file note the bytes are reversed >0102=>0201 (n/a for program)
0014-001B	0020-0025	Not used
001C-00FF	0026-0255	Block cluster linkage (see note)

NOTE: Files are placed on disk in first-come first-served manner. The first file written will start at sector >0022, and each subsequent file will be placed after it. If a file deleted , the next file written will start in this hole. If the hole is not long enough for the file the file will be fractured, and the remainder will be placed in the next available block of sectors . the block cluster link map keeps track of this fracturing. Each block cluster link is 3 bytes long. Byte 2 is divided into 2 nybbles (4 bits). The rightmost nybble is appended to the left end of byte 1. The leftmost nybble is appended to the right end of byte 3. the result is 2 12-bit numbers. The left number represents the starting sector for this cluster. The right represent the number of sectors within this cluster. e.g. >001C=>22 20 01 this equals >022 and >012 where >022 equals starting sector >022 with >012 sectors in the cluster.

SECTORS 22-168 - Data Files

These sectors contain file data and file headers. The first byte of the first sector of each file is where the proprietary protection invoked by Extended Basic is flagged. to unprotect this file, change >0000 to its 2s complement, leaving the LSB=1.

HOW TO FIX DISKS...

By Niraj N. Shah Mike Ballman

Editor: I think that this file may be of interest to many members in our group that have had a sector 'blow' on them on a disk that had important files on them. This article is written to use DISK FIXER program to correct blown sectors, but any sector editor program could be used and applying these concepts. Maybe someone could write a addendum to this article on how to use other sector editors???

Did you ever try to catalog a disk and find out the Disk Controller thinks the disk is NOT initialized? But you know better! What do you usually do with the blown disk? Most people Delete the file giving them the problem. Usually that does correct the problem, but it also gets rid of that file forever. The ulti mate solution is to use DISK FIXER by Navarone Industries.

The DISK FIXER enables one to examine and change the contents of any disk on a a sector by sector basis. I think it is worth its forty-dollar list price. It is available from some TI retailers or directly from Navarone Industries.

Here is the process to fix a blown-up disk...

First acquire a DISK FIXER from a friend buy one; they're worth it. Get a hardcopy catalog of the blown disk, or even better, get a complete (old) catalog of what should be on the disk. If a complete catalog is not available try to remember what should be on the disk and write those names down on paper. Once you have a catalog of the disk, you are ready to start using DISK FIXER.

Insert the DISK FIXER cartridge and select option 2 from the Title Screen. Upon doing so you should see the DISK FIXER menu. Do the following if the most recent catalog of the bad disk tells you there are more sectors used/free than is logically posssible: 358 for single side d 718 for double sided disks. For example, IF the catalog lists 500 sectors used/free on a single-sided disk THEN do the following ELSE GOTO the paragraph on "SECTOR ONE".

This part tells you how to fix up Sector 0; which is the sector containing the in formation concerning the disk name and number of sectors used/free on the disk. If the disk catalog tells you the used/ free sector information is in error then Sector 0 needs to be fixed. The easiest way to do this is to copy a good Sector 0 from another disk to the blown disk. Here is how to do that:

1) Insert a good disk in drive
2) Read Sector 0 of that disk:
```
^^^^^R 0,1 [ENTER]
```
3) Put the blown disk in drive
4) Write good Sector 0 to disk:
```
^^^^^W 0,1 [ENTER]
```

If you catalog the bad disk, you will see that the diskname and the used/free information is the same as the good disk But do not let that alarm you. We did that to fool the Disk Controller into thinking the bad disk is at least partially restored to normalcy. Now we need to fix up the blown disk as much as we can. This is done by changing Sector 1.

Here is how to fix Sector 1. First, get the most complete catalog and the most re cent catalog of the bad disk in front of you. Then compare the two catalogs to see which filenames are missing. Next, compile an alphabetical list of all the filenames which are and should be in the catalog.

HOW TO FIX DISKS (cont.)

Then you need to find the corresponding sector for each filename. This is done by using the Find String function of the DISK FIXER

1) Put the bad disk in drive
2) Find a filename by:

```
 F 0,2D0,1 [ENTER]
```

type in the filename [ENTER]
3) Ignore the "ERROR IN SECTOR" message
4) Write down the sector number for that filename
5) If that filename could not be found make sure you typed it in correctly and try again; otherwise that file does not exist on the disk.
6) Repeat the process from step two for all of the filenames

You should now have an alphabetical list consisting of two columns: filenames and sectors. With that information in hand you are ready to begin fixing up the bad disk. This is done by modifying Sector 1 of the blown disk. First you have to read Sector 1 from the bad disk by doing this:

1) Put the bad disk in drive
2) Read Sector 1 of disk by:

```
    R 1,1 [ENTER]
```

Then you want to alter the contents of sector 1. This is done by using the alter function of the DISK FIXER. This process is best learned by observing a concrete example.

Lets say the blown disk has 14 files (filenames) on it. Thus there should be 14 entries on sector 1; one entry for each file. The rest of the sector should be all zeros. Lets alter Sector 1:

1) Keep the bad disk in drive
2) Enter the Alter function:

```
    A 0 [ENTER]
```

3) Type in the following just as shown, including the spaces:

```
 1 2 3 4 5 6 7 8 9 A B C D E
```

4) Do not press [ENTER] yet!
5) If you saw a non-zero entry after the E entry in the first column then type in [0] and a [SPACE] and repeat until the first column shows a zero.
6) Press [ENTER]
7) Write the revised Sector 1 to the bad disk:

```
    W 1,1 [ENTER]
```

You have just entered a table of pointers to the files on the disk. The table points to the corresponding sector for each file name. This is the table that is updated and sorted if you add/delete files to the disk.

Leave the DISK FIXER by typing [Q] for QUIT and press [ENTER]. Then catalog the disk. Lets call this new catalog the mixed catalog. You will see the reason once the disk has been cataloged. Notice how the catalog is NOT in alphabetical order It does however contain all of the file names that you hoped and prayed would be on the disk! The next step is to alphabe tize the catalog. This is done by first alphabetizing the catalog on paper and carrying along the appropriate sector number of each filename. Here is an example of a Mixed Catalog:

HOW TO FIX DISKS (cont.)

MIXED CATALOG		SORTED CATALOG	
FILENAME	SECTOR	FILENAME	SECTOR
CAT	1	APPLE	E
SCREEN	5	CAT	1
VOTE	2	DEMO	7
FIRE	6	FIRE	6
APPLE	E	HELLO	9
HELLO	9	JUSTIFY	D
SCROLL	C	LOAD	3
LOAD	3	LOGO	A
TIME	8	PLOT	B
DEMO	7	QUICK	4
QUICK	4	SCREEN	5
JUSTIFY	D	SCROLL	C
PLOT	B	TIME	8
LOGO	A	VOTE	2

The above example shows how you should alphabetize the filenames and the corresponding sector numbers on paper. If you are unsure when dealing with funny characters, the system alphabetizes by lower to higher ASCII values. These values can be found on your TI Basic reference card Once you have done this you are ready to enter this information into Sector 1. You do not have to enter the filenames, just the sector numbers.

Here is how to do that:

1) Put the blown disk in drive
2) Read Sector 1 by entering:
^^^^^^^R 1,1 [ENTER]
3) Enter the Alter function:
^^^^^^^^^A 0 [ENTER]
4) Type in the sector numbers in the order as shown for the above sorted example catalog. Separate each number by a space:
^^^E 1 7 6 9 D 3 A B 4 5 C 8 2
5) Then press [ENTER]
6) Write revised sector to disk:
^^^^^^^^^W 1,1 [ENTER]
7) Put a Write-Protect tab on the disk!

You have now fixed up the disk. For verification quit the DISK FIXER program and catalog the disk. You should have no problems during the cataloging process. But you are not completely done yet! DO NOT add/delete any files or programs to this disk!

Get a fresh disk and inititalize it to the same configuration as the blown disk Then backup the blown disk to the fresh disk. Then catalog the fresh disk and you will see that the used/free sector information is now correct. Thus, the fresh disk is now your working disk and the blown disk is now a disk for your archives.

Keep the blown disk in a safe place just in case you remember a file that was not previously recovered from the blown disk Go through the above procedures to recover that new-but-old file.

LOADING FROM DISK...

There have been several users in the group who have asked questions regarding how to run a program. Program files can be in several forms from BASIC programs to assembler programs. This article will hopefully answer some, maybe all the questions you have about program files. This is reprinted from the BITS, BYTES, and PIXELS, the newsletter of the LIMA 99/4 Users Group.

Disk files that can be loaded directly into the computer are in the following forms:

o PROGRAM
o INT/VAR 254
o DIS/VAR 163
o DIS/VAR 80
o DIS/FIX 80

Any other format type represents a data file and cannot be loaded directly into the computer, but instead are used by another program.

PROGRAM - These files are the most common and the vast majority represent TI BASIC or EXTENDED BASIC programs. Many TI BASIC programs load and run correctly in EXTENDED BASIC (but not visa versa). However if you load a program into EXTENDED BASIC and get a BAD VALUE IN XXX error when you run the program, the program may only be run in BASIC. The BAD VALUE error is caused by the use of chars above 143 which is not allowed in EXTENDED BASIC.

(Editors note: The BAD VALUE may also be caused by a CALL COLOR statement as BASIC allows color sets 1-16 to be addressed, while EXTENDED BASIC only allows the first 14 color sets to be changed. There is a utility program that may be used to allow a BASIC only program to be run from EXTENDED BASIC. This utility somehow converts these bad values into the proper values to run in EXTENDED BASIC. The utility though requires that the user also have 32K memory expansion. You can find the program listing in this newsletter on page 6.)

If you attempt to load an EXTENDED BASIC program with BASIC you will be given a error message FOR-NEXT ERROR IN XXX. Attempting to list line XXX causes the system to lock-up. You cannot use TI BASIC to work with EXTENDED BASIC programs.

If a program file occupies more than 45 sectors and won't load in either version of BASIC, you have to open up extra memory. Do this by typing the following:

```
CALL FILES(1) (enter)
NEW (enter)
OLD DSK1.filename (enter)
```

The program will now probably load.

Occasionally a PROGRAM file will not load from either version of BASIC, giving an I/O ERROR 50 when you attempt to do so. These files are more than likely assembly program that need the Editor/Assembler module to run. After calling up the first menu, press "2" for EDITOR/ASSEMBLER, then press "5" for RUN PROGRAM FILE. When prompted, type DSK1.filename, hit enter and the program should begin to load and run. Some assembly files of this type may also be run from the TI-Writer option #3, UTILITY.

(Editors note: If you have the Funnl-Web Farm, TI-Writer loader version #3.1, you may also use the utility option to run assembly programs.)

LOADING FROM DISK (Cont.)...

Finally, some specialized PROGRAM files can only be loaded from the ADVENTURE, PERSONAL RECORD KEEPING, STATISTICS or other specialized module. The files are actually data bases that can be used with their particular module.

INT/VAR 254 - These files are large EXTENDED BASIC programs most of the time. To run them use OLD and RUN just as you would for any other EXTENDED BASIC file. These files do require 32K memory expansion. These files will be larger than 45 sectors on a disk and do not require a CALLS FILES(1) to load. Once loaded these programs can not be saved to tape without special techniques. You cannot OLD any INT/VAR 254 program from BASIC.

DIS/VAR 163 - This type of file represents a EXTENDED BASIC program that has bee saved in MERGE format. Usually these will be subroutines that can be merged into another EXTENDED BASIC program. To load, type MERGE DSK1.filename and hit enter. You cannot use OLD with these types of files. To save a program in MERGE format, type SAVE DSK1.filename, MERGE. The MERGE option is not available from BASIC.

DIS/VAR 80 - These are text files which can be read, edited and printed using TI-Writer. Editor/Assembler may also be used by selecting E/A option #1. Many of our more complicated programs have documentation files on the same disk as the program, and usually have the letters DOC in the filename.

DIS/FIX 80 - These are assembly programs which must be loaded with Editor/Assembler or Mini-Memory modules. Press #2 from E/A or #3 for Mini-Memory. Then press the number which corresponds to LOAD AND RUN. When asked for the filename enter DSK1.filename, then hit enter. The file will then load and may begin to start, if not press enter you'll be asked for the PROGRAM NAME. The name can be found in the documentation file usually, but if you have no doc file try enter START, START1, or BEGIN. Sometime the name will be the same, or similar to the filename itself.

(Editors note: The above file may also be an assembler program to be run using EXTENDED BASIC. These are usually subprograms or routines which may or may not be addressed from the command mode and then wait in memory to be used. One such program is Danny Michaels program NEATLIST. To load these programs type CALL INIT::CALL LOAD("DSK1.filename"). The program will then load and return you to EXTENDED BASIC. In the case of NEATLIST, you simply enter CALL LINK("NEATLIST") and the assembler program begins to run.)

FINAL NOTES - Any of the above file types may also be used as a data file and can only be loaded from another program. The computer recognizes that this file is not a program file and will not load it.

With the above information, you should have no trouble loading anything from disk. Unless you know how to load a file, it may just take a little trial and error.

D/F 80 to ASSEMBLY PROGRAM (Cont.)....

HOW TO CONVERT ASSEMBLY PROGRAMS TO PROGRAM FORM FOR FASTER LOADING AND LESS DISK SPACE.

Written by Darren Leonard PUG
on an idea by Marty Kroll Jr.

If you have ever loaded an assembly program with editor/assembler option #3 you may have noticed that it takes quite a while to load. With some programs this can take over 2 minutes. These types of programs are in Dispay/Fixed 80 format which we are going to change to PROGRAM format to load with OPTION #5. In addition to loading 3 to 5 times faster, programs stored in program format,ie Memory Image, take as little as 1/4 the disk space of D/F 80 files.

The method outlined in this article will work on 95% of all Assembly D/F 80 programs. Prior to writing this, I tried it on 20 programs and it worked on 19 of them. It will even allow you to save a ASSEMBLY program to cassette. Thus people with and E/A and 32K can run assembly programs!

To begin with read page 420 of the Editor/Assembler manual. Try your program the way they outline it. If you get an error then read on and I will explain in detail how to get around it.

This section describes the procedure for D/F 80 files that DO NOT AUTOSTART!, if your program does autostart read down a few paragraphs on how to remove it with DISKO.

1) Plug in your E/A and call up TI-BASIC, your E/A must be plugged in!

2) Type " CALL INIT"
   " CALL LOAD("DSK1.FILENAME")

3) If your program has more that one file type in all the remaining files in order as follows:
   " CALL LOAD("DSK1.GAME*1")
   " CALL LOAD("DSK1.GAME*2")
   " CALL LOAD("DSK1.GAME*3")
   get the idea?

4) Type "CALL PEEK(8228,A,B)"
   PRINT A,B

5) Now 2 numbers will appear on the screen , one on the left and one in the middle of the screen. This number corresponds to the first free address in the memory which is also the last address of your program.

6) Convert this numbers to hex and add A+B to come up with a 4 digit hexadecimal number. Since you program is normally loaded in memory from addresses >A000->FFD7 if you get A000 for A+B then you program has an Absolute Origin statement (AORG) and you will not be able to convert it with this method. Similiarly, if A+B is A780 or smaller then the program is loaded in a unusual manner since it cannot fit in the small area from >A000-A780. But if you come up with A+B=B000 or greater then this method will work 99% of the time.

7) Type " BYE " and call up the editor. Now type in the small assembly program listed here:

```
       DEF SFIRST,SLAST,SLOAD
SFIRST EQU >A000
SLOAD  EQU >A000
SLAST  EQU >A780 (the value of A+B)
       END
```

NOTE!! PUT THE HEX NUMBER OF A+B IN THE PLACE WHERE A780 IS!!!!!

Hit Fctn 9 twice and save to disk.

8) Load the Assembler.

For source file enter what you save in step 7.

For object file type DSK1.GAME*4 or what you want.

Hit return for the printer output.

TYPE "RC" when it prompts for assembler directives.

It will then assemble the program. You shouldn't get any errors.

9) Now load E/A option 3.
Enter your filename DSK1.GAME*1
DSK1.GAME*2

Then enter the assembled filename from step 8. DSK1.GAME*4

10) Insert E/A disk #2 into drive one and load file " DSK1.SAVE".
Hit enter and type "SAVE" for the program name. Follow the screen input prompts.

11) Now hit FCN + and call up E/A option #5 and type DSK1.YOURFILE and wala!

D/F 80 to ASSEMBLY PROGRAM (Cont.)....

HOW TO CONVERT FROM DECIMAL TO HEXADECIMAL.

This might appear quite intimidating but assure you that it is very simple. I will not go over the priciples of HEXIDECIMAL numbering systems because that is beyond the scope of this aticle, I will how ever show you how to convert to it.

Decimal	Hexadecimal	Binary	Octal
0	0	0	0
1	1	1	1
2	2	10	2
3	3	11	3
4	4	100	4
5	5	101	5
6	6	110	6
7	7	111	7
8	8	1000	10
9	9	1001	11
10	A	1010	12
11	B	1011	13
12	C	1100	14
13	D	1101	15
14	E	1110	16
15	F	1111	17
16	10	10000	20
17	11	10001	21
18	12	10010	22

AD INFINITUM ....

The number in the left column represents the numbers you are familiar with. In the second ,third and forth columns are the equivalent numbering systems.

Take A from step 4 above .
say it is 213 which is in decimal.

Divide by 16   213/16=13.3125
Take the part to the left of the decimal point, which in this case is 13 and convert to Hex from above chart 13=D.

Now take 213- (13*16)=5 and this =5 in hex. Therefore your hex number is D5 which equals 213 decimal.

Do the same for B and add the D5 to what you obtain for B. If the first didgit is not a A,B,C,D,E or F you have and invalid address or you have incorrectly converted to hex.

By doing the exact reverse of the above you can go from HEX to DECimal.

TROBLESHOOTING THE PROCEDURE

If you encountered and error in steps 1-11 above there is still hope!

If you received an error in step 9 when you attempted to load your assembled program, and that error was a "DUPLICATE DEF" you may attempt to figure which is the duplicate:SFIRST,SLAST or SLOAD by two ways.

1) If you have DISKO load it up and search your program file for SFIRST,SLAST or SLOAD on you disk and change them to TLOAD,TLAST or TFIRST AT EVERY PLACE they occur!! BE SURE TO CHECK THE LAST 3 SECTORS OF THE PROGRAM THOROUGHLY!!!

Then go back and try STEP 9 AGAIN.

2) Change the Assembly program in 7 to allow all combinations.

```
       DEF SFIRST,SLAST
SFIRST EQU >A000
SLAST  EQU >A+B
       END
```

Try it eliminall three one at a time.

if that doesn't help try eliminating 2 of the words:

```
       DEF SLAST
SLAST  EQU >A+B
       END
```

Then try using only SFIRST then SLOAD.

If this doesn't work you will have to wait untill part 2 of this article comes out.

HOW TO ELIMINATE AUTOSTART FUNCTION ON D/F 80 PROGRAMS.

If your program autostarts, you cannot use the above procedure because it will take over controll of the machine. You can remove that feature if you have DISKO.

Load up DISKO and examine the last 3 sectors of your program for the following

```
(in hex mode)  20314523462020
                 ^^    ^^
```

They thing to look for is the 31 and the 46 with and address between them. Change the 31 to a 40 or change the entire sequense to 20

after change it should look like this:

```
20202020202020  or  20404552462020
  ^^    ^^            ^^    ^^
```

D/F 80 to ASSEMBLY PROGRAM (Cont.)....

Important!!! MAKE SURE THAT YOU DO THIS ON A BACKUP COPY OF YOUR PROGRAM ONLY!!

You may need to look back a few more sectors if you are unable to find the last 3. Also the 31xxxx46 must come before the :99/4 code on the last sector of the program.

I hope this article is useful to you and if I get around to it I will write another article on how to convert those unusual programs that cannot be done with this method.

Danny Leonard Editor PUG, 1218 Michael Drive, Pittsburgh, PA 15227

==

LOAD INTERUPT SWITCH...

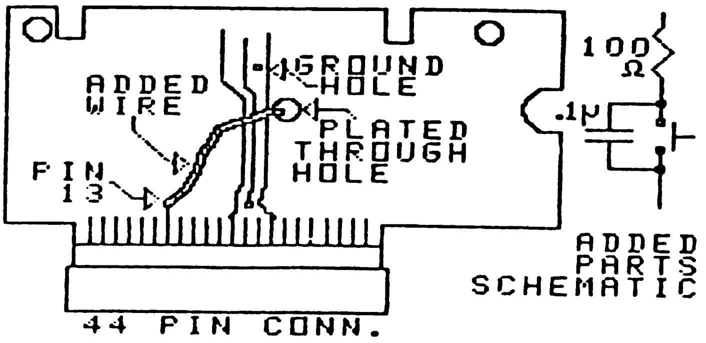
By: Tom Rhodes, Reprinted from BYTEMONGER, newsletter of BLUEGRASS AREA 99'er

There is an excellent screen dump program by Danny Michaels which uses a load interupt switch which can be installed in your speech sythesizer or in your console. I have installed mine and the results are great! Screens from modules, even TI Title screen can be dumped to your printer. How do you do it? Here is a summary from two versions which appear in other user group newsletters.

1. Buy the parts at Radio Shack or elsewhere.
   a) A subminature switch, push button momentary contact.
   b) A 100-500 ohm resistor
   c) a .01-.1mfd capacitor
2. Dismantle sythesizer. Note how shield slides together.
3. Locate a plated through hole about in the center of the circuit board. Clear the solder away from this hole as there is nothing attached to this hole on eithr side (make sure).
4. On the bottom side of the board, solder a solid strand wire to pin 13 of the 44 pin connector. Pin 13 can be located by counting from left to right on the bottom of the board, with pin 44 facing south as you look at the circuit board. It is the 7th pin from the left. All other parts go on top of the cicuit board. 5. Solder one end of the 100 ohm resistor in ground hole. Looking straight into he I/O port, pin 1 is on the bottom left, pin 2 is top left, pin 3 is to the right of pin 1, and so on. Pin 21 is the ground. When you solder use only a smal pencil type iron, not a gun.
6. Solder 2 or 3 inch piece of wire to other end of resistor.
7. Solder other side of resistor to one side of switch with one lead from capacitor. Other capacitor lead is soldered to other side of switch with the wire from pin 13 (runs through through hole).
8. Drill hole in middle top of shield for switch.
9. Mount switch, check leads and make sure everything fits.
10. Reassemble unit, carefully make sure nothing shorts.

Follow instructions on loading the screen dump program to check operation of the switch.

Please note that if you do not know what you are doing, you should not attempt this project. We cannot assume liability for your computer should you any of these modifications.

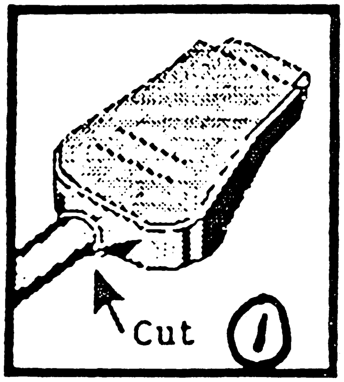


BOTTOM VIEW OF SPEECH SYNTHESIZER BOARD

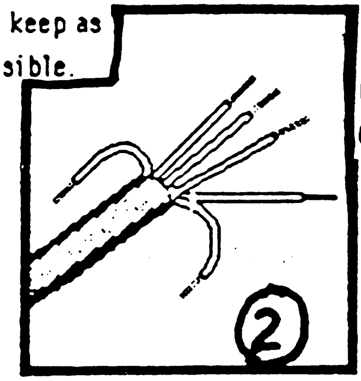
For those of you who don't have Wico Atari/Commodore to TI-99/4A adapter; here is a useful joystick conversion which recently appeared in R/D Computing. (Do it at your own risk! Also, where applicable, it likely to void the warranty on your joystick -Ed.):

Atari to TI-99/4a Joy-stick Conversion

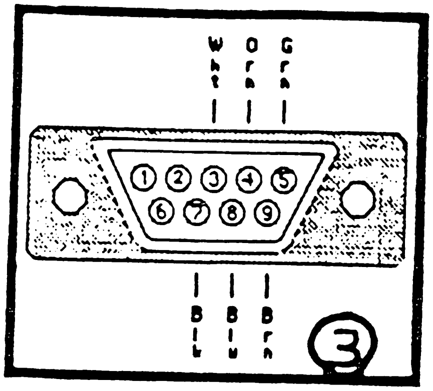
- by Mark G. Webb



Cut the cable as close to the plug as possible to keep as much length as possible.

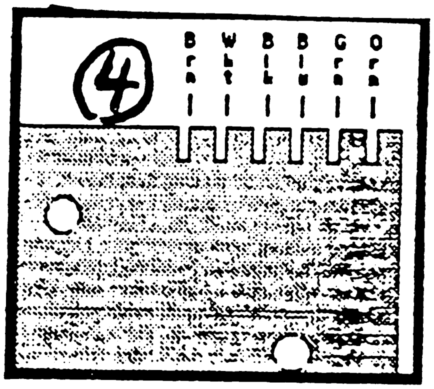


Trim back the outer insulation 3/4 to 7/8 of an inch. Then trim back the insulation on each wire 3/16 of an inch.



Tin the wire tips and solder in the following manner:

White to pin 3 Black to pin 7
Orange to pin 4 Blue to pin 8
Green to pin 5 Brown to pin 9



It is not necessary to open the joystick to complete the process but if you do these are the wire connections on the board inside.

Triple Tech Tip

-Gil Tennant

For those of who use the Corcomp Triple Tech(TM) card; there have been reports of exploding lithium batteries due to power from the Peripheral Expansion Box "charging" this non-chargible battery. A fix comes from Gil Tennant:

I have developed a fix for the exploding battery problem on the Triple Tech Clock Card. For those of you with the card and is, or know of someone who is electronically inclined, this is the fix to prevent battery charge during P-Box operation:

1) Locate resistor #7 (to the lower right of the speech synthesizer slot), near the clock IC 5832

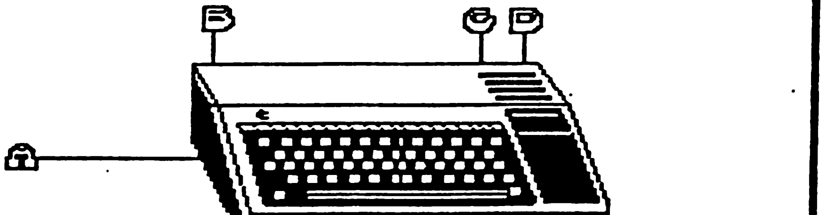
2) With a 15W pencil-type soldering iron or desoldering tool, remove the #7 resistor,(labelled R7)

3) Replace R7 with a 1N914 diode(Radio Shack sells them as cat.#276-162Ø in packages of 5Ø for $2.99 -Ed.)with the anode-end towards the battery

4) Solder in-place and do a volt meter test on the battery's contact (leads) with the battery removed and card in-place with P-Box running. Presto! No voltage at the leads!

5) Put the battery in and set the clock; turn off the system for 1Ø minutes. Power-up the system and reboot the clock program. Voila! You have just prevented the battery from being charged! This fix has been confirmed by Corcomp to work with no loss of function to the card. NOTE: As usual you assume all resposibility for damage to you system in this project; it is done at your own risk!

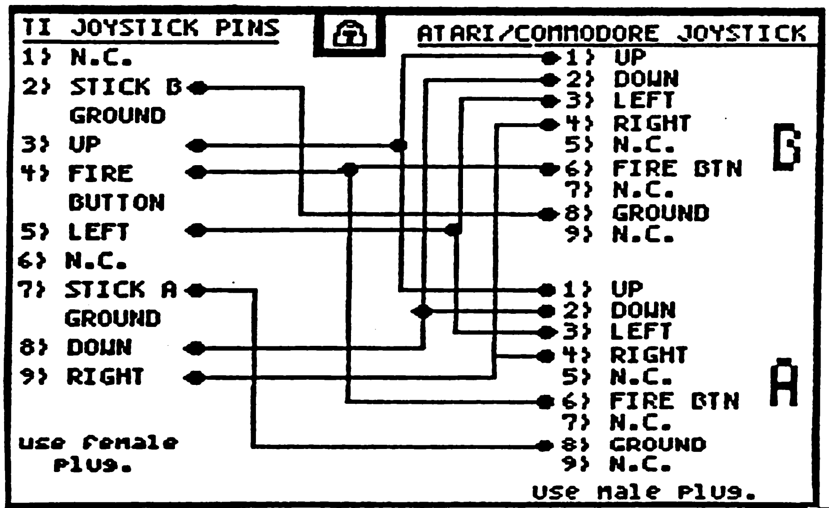
PIN ASSIGNMENTS



PIN ASSIGNMENTS APPEAR AS THEY WOULD IF YOU LOOKED DIRECTLY INTO THE PLUGS.

FORT USER GROUP

FEBRUARY, 1987



TI JOYSTICK PINS (A)

1) N.C.
2) STICK B GROUND
3) UP
4) FIRE BUTTON
5) LEFT
6) N.C.
7) STICK A GROUND
8) DOWN
9) RIGHT

use female plug.

ATARI/COMMODORE JOYSTICK

B

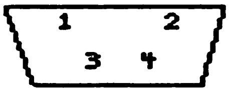
1) UP
2) DOWN
3) LEFT
4) RIGHT
5) N.C.
6) FIRE BTN
7) N.C.
8) GROUND
9) N.C.

A

1) UP
2) DOWN
3) LEFT
4) RIGHT
5) N.C.
6) FIRE BTN
7) N.C.
8) GROUND
9) N.C.

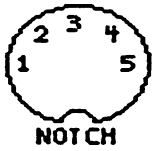
use male plug.

POWER SUPPLY PLUG (C)



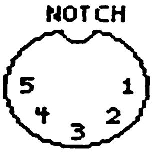
1) NOT USED
2) 8 VOLTS AC
3) 16 VOLTS AC
4) GROUND

MODULATOR PLUG (B)



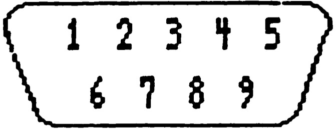
1) RED
2) TRANSPARENT
3) WIRE WRAP (not a ground)
4) BLACK (ground)
5) WHITE or YELLOW

VIDEO/SOUND PLUG

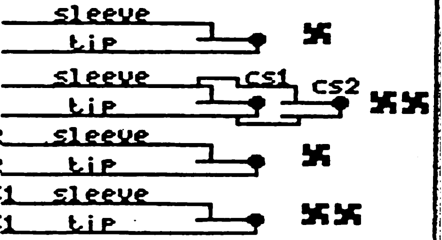


1) +12 volt
2) VIDEO OUT
3) SHIELD
4) GROUND
5) SOUND OUT

CASSETTE PORT IN CONSOLE (D)



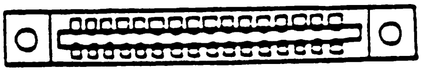
CASSETTE CABLE BELOW



2) NEGATIVE REMOTE CS1 sleeve
1) POSITIVE REMOTE CS1 tip 卐
3) NEGATIVE MIC 1 & 2 sleeve
5) POSITIVE MIC 1 & 2 tip 卐卐
7) NEGATIVE REMOTE CS2 sleeve
6) POSITIVE REMOTE CS2 tip 卐
9) NEGATIVE SPEAKER CS1 sleeve
8) POSITIVE SPEAKER CS1 tip 卐卐
4) NOT USED

卐 SMALL REMOTE PLUG 3/32"
卐卐 PHONE PLUG 1/8"

I/O PORT PIN ASSIGNMENTS
GROM PORT



PIN	DESCRIPTION	PIN	DESCRIPTION
1	RESET	2	GND (SYSTEM)
3	D7	4	CRU CLK
5	D6	6	CRU IN
7	D5	8	A15/CRU OUT
9	D4	10	A13
11	D3	12	A12
13	D2	14	A11
15	D1	16	A10
17	D0	18	A9
19	+5 VOLT	20	A8
21	GS (GROM SELECT)	22	A7
23	M0/A14	24	A3
25	M1 (DBIN)	26	A6
27	GROM CLOCK	28	A5
29	-5 VOLT	30	A4
31	GR (GROM READY)	32	WE
33	GND (GROM)	34	ROM G
35	GND (SYSTEM)	36	GND (SYSTEM)

I/O PORT PIN ASSIGNMENT
PERIPHERAL I/O port

PIN	DESCRIPTION	PIN	DESCRIPTION
1	+5 VOLT	2	SBE (SPEECH SELECT)
3	RESET	4	EXT INT
5	A5	6	A10
7	A4	8	A11
9	DBIN	10	A3
11	A12	12	READY/HOLD
13	LOAD	14	A8
15	A13	16	A14
17	A7	18	A9
19	A15	20	A2
21	GND	22	CRU CLK
23	GND	24	Ø3
25	GND	26	WE
27	GND	28	MBE
29	A6	30	A1
31	A0	32	MEMEN
33	CRU IN	34	D7
35	D4	36	D6
37	D0	38	D5
39	D2	40	D1
41	HOLD/IAQ	42	D3
43	-5 VOLT	44	SPEECH

HAMAI'S hard WARE #5

WARNINGWARNING***

This project requires some skills and knowledge in electronics assembly. Incorrect assembly could result in burning up your disk drive. If you are not sure how to connect the parts, contact me or somebody who can help. In any event, neither I or the ROM will cover you for any damages or losses resulting from the use of this power supply as suggested by this article and you are using it solely at your own risk.

This power supply was only tested with the Shugart 400, 400L and 450 disk drives. These drives have a low power requirement and as a result work well with this supply. Disk drives which require stronger power supplies will not work.

PARTS LIST

1. Radio Shack 277-1016 power supply chassis
2. Radio Shack 273-1511 12.6volt, 3amp transformer
3. 1/2 amp fuse and holder
4. SPST bat switch
5. 5 ft. of lamp cord and plug
6. Three 6 inch lengths of 20-22ga stranded wire, different colors
7. Male plug for disk drive power connector
8. Small piece of heat shrink tubing 1/16th inch size

All of the above items except for item 7 are available at Radio Shack stores. The disk drive connector plug is available from R&D Electronic Supply, 100 E. Orangethrope Ave. Anaheim, CA 92801.

If you plan to use the same thing I used on the demo, then you will need to have a TI computer transformer. If you have the TI transformer then you would not need to purchase items 2, 3, 4, and 5 on the parts list.

When you purchase the power supply board, you will note that it comes with instructions on a suggested wiring scheme. These instructions also recommend the use of an 18volt transformer and 2amp fuse. The reason I recommend the other transformer is because the power supply does not have to work so hard to regulate the output voltages and the lower amperage fuse gives quicker response to an overload. An added plus is that the 12.6volt transformer costs less.

Step 1. - See fig. 1 for suggested hookup for the transformer and the power supply. You will note this is the same as shown in the Radio Shack diagram. I have included some notes for clarity.

Step 2. - Double check your connections and then plug in your supply to an outlet to test it out. Be sure to turn on the power supply switch located on the board (see fig. 3). Using a suitable volt meter and fig. 2, check that you get the indicated output voltages when you test the +5 and +12 pins and ground. The voltages MUST be pretty close. DO NOT use the power supply if you find it is off by 1/2 volt or more, especially on the +5 volt pin. If the voltages are way off, I suggest you return the board and get another one.

Step 3. - Disconnect the power to the supply and carefully bend the -5 volt pin out of the way or cut it completely off. Then solder one of the 6 inch lengths of 20-22ga wire to each of the remaining pins. Use a piece of heat shrink tubing over each soldered connection for insulation. Use yours or a friend's blow dryer at the High setting to shrink the tubing.

Step 4. - Referring to fig. 4, assemble the three wires you solderud to the pins into the power connector for the disk drive. Double check your wiring and test the connector with your voltmeter to be sure that you have the wires in the correct socket positions.

That's all there is to the wiring. If you connected up your disk drive now, it should work.

One more thing, I have not included plans for a cabinet for the disk drive and power supply. You will need to build one to hold your components together. For my demo model, I used the TI computer power supply transformer and eliminated the extra transformer, fuse, switch and lamp cord. And mounted the disk drive to a piece of plywood and covered the whole thing with a cardboard to keep the fingers and dust out. I suggest you take your own measurements for the design of the cabinet. See fig. 5 for a suggested configuration.

Using the above and one of the TI/Shugart 400L disk drives you should be able to set up a second or third disk drive drive for your system AND this power supply for less than $40.00, including the cardboard and nails.

Bye for now. Be especially careful out there and tell them all you saw it FIRST!...in the ROM...Ken H.

THE R O M NEWSLETTER
USERS GROUP OF ORANGE COUNTY
17301 SANTA ISABEL STREET
FOUNTAIN VALLEY, CA 92708

This article was originally published in the September 1985 issue.

DISK DRIVE POWER SUPPLY

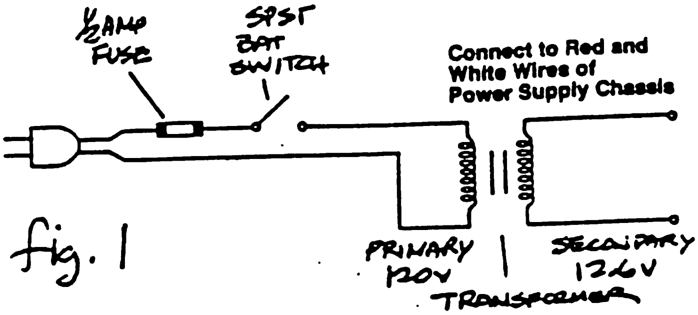
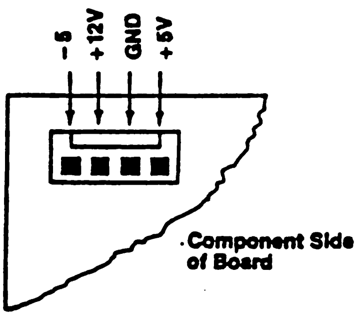


fig. 1

Output Connection



-5V NOT USED

fig. 2

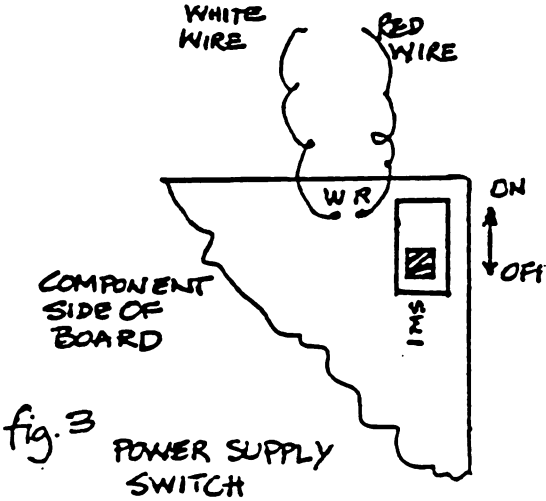
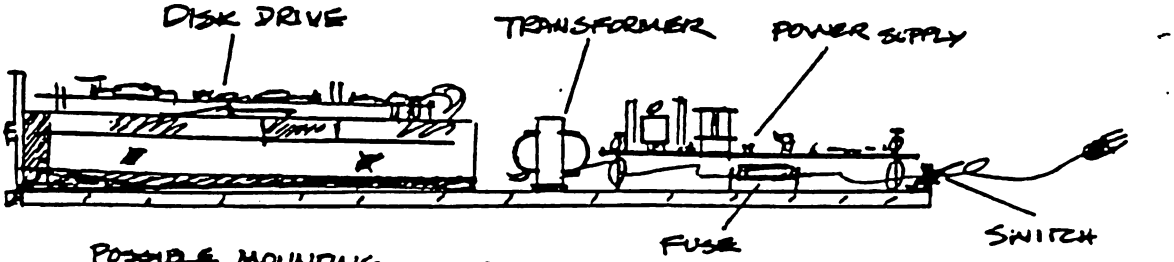


fig. 3 POWER SUPPLY SWITCH



POSSIBLE MOUNTING CONFIGURATION fig. 5

-NOTES-

-NOTES-

TI Users Group of Will County
P.O. Box 216R
Romeoville, IL 60441