Ogdens 99'er User Group
Mail Address: 1396 LINCOLN Ave. Apt.B Ogden Utah 84404

GROUP OFFICERS

President: JimBuck 773-2552
Vice President: Harold Bingham 394-6382
Secretary-Treasurer: Richard Scott 776-2551
Librarian: Harold Hilburn 773-0622
Asst. Librarian: Mel Bragg 393-9605
Newsletter Editor: Mel Bragg 393-9605
Associate Editor: Harold Hilburn 773-0622

FEBRUARY 1990 NEWSLETTER
OUR NEXT MEETINGS ARE:

SATURDAY: FEBRUARY 03 TIME: 0900 hrs.
TUESDAY: FEBRUARY 06 TIME: 1800 hrs.
TUESDAY: FEBRUARY 20 TIME: 1900 hrs.

We will be meeting in the CIVIL AIR PATROL building at the OGDEN MUNICIPAL AIRPORT AIRPORT ROAD. On the 03rd.

We have started a BASIC class for those interested. The people taking the BASIC class will meet on the FIRST and THIRD Tuesday of the Month at times above. Jim Buck is the teacher.. These meetings are held at the Hill Side Center in the basement. ROY CITY CENTER.

OGDEN TI USERS GROUP
1396 LINCOLN APT #B
OGDEN, UTAH 84404

HAPPY VALENTINE

A publication of the TI 99/4 and 9640
Ogden users group inc.

TI 99-4A

OGDENS USERS GROUP

THE CLUB THAT REFUSES TO SAY GOODBYE

February 1990

STOP PRESS NOTE ADDED TO THE FOLLOWING ARTICLE:

We have just received a letter dated Dec 04/89 from Tony McGovern in response to a letter we sent him about a few bugs in the initial release of Funnelweb v4.2. Tony says "You are right, don't circulate widely on 4.20 until 4.21 is received.....I'll send 4.21 very soon." It was initially our intention to include disks with FWB v4.2 to all exchange user groups and out of town paid members who receive this newsletter. We will now wait on this. When v4.21 is received, we will mail it. The major bug in the current v4.2 is that you cannot run XBASIC software from the XB user list. You are instead returned to XB command mode. This really is no problem, because you can run XB software from the DISK REVIEW part of Funnelweb. Those user groups and out of town members who are super anxious to get their hands on v4.20 as it now exists can do so by sending TWO DISKS and a paid return mailer. We will send out v4.21 when it is available.

FUNNELWEB VERSION 4.2, IN 40 AND 80 COLUMNS
new features report by Charles Good
Lima Ohio User Group

Dated Nov 14, the first release of Funnelweb v4.2 has been sent by Tony McGovern to his regular correspondants and to those who recently sent him a fairware donation. Originally, Tony was planning to completely rewrite an 80 column version of FUNNELWEB from scratch, making its coding more "elegant" and compact and giving it the new name of WHIPBIRDS. However, such a rewrite would have made WHIPBIRDS largely incompatible with previous versions of FUNNELWEB with respect to previously created user lists and system configuration (SYSCON) data. So Tony has put WHIPBIRDS on hold and has incorporated some of the new features he planned for WHIPBIRDS into the latest FUNNELWEB.

DISK REVIEW:

The enhanced disk directory available to 80 column users (Geneve, AVPC card, or Mechatronics 80 column peripheral) since last summer as file GD4V has been renamed DISK REVIEW. It comes preconfigured in one of the FWB central menus and now comes in both 80 and 40 column versions. Yes, now 40 column users can make use of this new VERY POWERFUL utility. With the 40 or 80 column DISK REVIEW you can bring up a disk directory and move the cursor next to a file name. You can then press "R" and run the software' This includes assembly language software that reads as PROGRAM or DF80 on the disk directory and extended basic software (if you have the XB module plugged in) that reads as PROGRAM or IV254 on the disk directory. The only thing you can't run this way is TI BASIC software that won't run from extended basic, such as PROGRAMs using TI2 speech.

Using either 40 or 80 column DISK REVIEW, you can protect, unprotect, delete, rename, and copy (a feature not previously found in GDAV) the disk directory file next to the cursor. File copy requires more than one drive. You can examine the contents of the disk header sector (sector 0), and you can examine each of the file header sectors. You can print a disk directory to a printer, or to a DV80 disk file name. When printing to a disk file, the file is opened in APPEND mode. This allows you to chain successive disk directories into one large text file which can serve as a reference data base to quickly show you the contents of your entire disk library. These features allow you to use DISK REVIEW for many of the things you would usually have to do from a disk manager.

With the 40 column DISK REVIEW you can load ANY KIND OF FILE into a 17K buffer for VIEWing on the screen. If the file fills the buffer, the file continues to load and overwrites the contents at the beginning of the buffer. You can load the whole file in at once, or page it in one screen of text at a time, or one file record at a time. PROGRAM files loaded this way are displayed simultaneously in both ASCII and Hex. The display resembles what you see with a sector editor. The most useful use of this VIEW feature of DISK REVIEW is the viewing of DV80 text files. You can take the DOC file of a piece of software and read it on the screen without having to make a hardcopy. Then you can go back to the DISK REVIEW disk directory and R(un) the software. The 17K buffer will accomodate about 68 disk sectors of text before it starts to overwrite itself. Once in memory, you can print the file to a printer in whole or in part. You can mark the buffer text in two places and print to a printer or disk file only that part of the buffer contents between the marks. This allows you to make a small DV80 file containing only part of the text of a larger DV80 file.

With the 80 column version of DISK REVIEW there are two 64K VIEW buffers, the second available only if you have 192K of video RAM installed. This is an easy option with the AVPC card. It can be done, but not easily, on a Geneve. You can put text in each of these buffers and display either buffer on the screen, alternating back and forth between either buffer and the disk directory. You can also display both buffers simultaneously, scrolling them both up and down and printing all or parts of either buffer. Text in each buffer stays in memory for instant recall without disk activity until you exit DISK REVIEW. The 40 column DISK REVIEW has only one text buffer.

When

REVISED EDITOR:

The text and assembly program 40 and 80 column editors have been rewritten internally to allow much faster move

NEXT PAGE

lines, copy lines, delete lines, and reformat. MOVE LINES no longer causes a TEXT BUFFER FULL condition. Because of the increased speed of reformat, it is now unlikely that characters will be dropped at the end of lines using the 40 column editor, slightly more likely with the 80 column editor. Text buffer capacity has been increased slightly. When editing, you can alternate back and forth between two different sets of tabs using ST (swap tabs). T displays the current set of tabs, ST displays the alternate set. Both sets of tabs are saved on disk when the file is saved with SF. In the 80 column editor the tabs can optionally be displayed on the bottom ruler.

A new feature has been added to the assembly source code editor (the E/A editor) that is useful for creating new assembly code, or typing in an assembly listing from a newsletter or Micropendium. You can, optionally, keep the alpha lock off and type everything in using lower case, using SHIFT for upper case as desired in the comment portion of each line of code. When you press (enter) after typing a line of code and its comment the code automatically becomes upper case letters as is required for source code, while the comment remains just as you typed it in a mixture of upper and lower case. This makes comments much easier to read and to distinguish from the actual source code when viewing a source code listing.

In the 80 column ShowDirectory, you can bring up a second directory without having to exit SD to the editor and then reboot SD. In the 40 column SD you still have to back out and then reboot SD to see another directory.

PATH NAME CONFIGURATION FOR HARD DRIVES ETC:

When configuring FWB v4.2 you have the option of designating a path name for all files. When set, path name configuration deactivates boot disk tracking and whatever drive numbers you choose to designate for the TIWriter and EA central menus. Path name choices are WDS1.FWB. or RD. or DSKR. or HD. if you choose this option. If you are using a hard disk WDS1.FWB. allows you to have all the Funnelweb files, including user configured software, as files in the same subdirectory. This leaves DSK1. emulation available for other purposes. Using RD. you can put all your Funnelweb files on a Ayarc ramdisk. I think DSKR. works with a Foundation card. Once Boot path is set, the rest of the configuration process just needs file names. Funnelweb will automatically insert the boot path name in front of the file name when searching for the file.

"ON THE FLY" RECONFIGURATION:

When exiting Funnelweb from either of the central menus you press FCTN/9 (BACK). You need to know this, since there is no screen prompt to suggest that FCTN/9 does anything here. If you answer "N" to the resulting QUIT? prompt you are then given the opportunity to temporarily load a different character set into memory, change the editor printer default name, and change the TI Writer and EA central menu drive numbers. These changes are not permanently read back to disk but they do remain in effect until you exit Funnelweb.

OTHER SMALL CHANGES:

Files in all disk directories are no longer marked by pressing a number. Instead you move the cursor next to the file name with the arrow keys (E? or FCTN/E?) to perform some action on that file. It is only really necessary to mark a file (with space bar or "M" or (enter) depending on where you are within Funnelweb) if you want a DV80 or DF80 file to be the workfile name that appears in the editor when you LF or SF.

You can now protect and unprotect with QUICK DIRECTORY, which you can access from most places in Funnelweb with FCTN/7 (AID).

When booting DF80 software from the LOADERS menu, or with R(un) from DISK REVIEW and you have put all the files into memory, you have to blank the next DSK? prompt off the screen with spaces or with ERASE before you press (enter) in order to display the START name. If you press (enter) without doing this you get an error. You can recover from this error with REDO. Then press ERASE and then (enter) to display the START name.

The FORMATTER is now an independent LOADER 62 file not linked directly to the rest of the Funnelweb environment. This means you can use other formatters, such as the one that comes with Art Green's "TI-Writer v4.2" instead of the formatter that comes with Funnelweb. Rename the alternate formatter files FD, FP, etc and put them on the funnelweb disk or subdirectory.

Screen layouts for the 80 column DISK REVIEW and SHOW DIRECTORY have been improved, compared previous versions of 80 column funnelweb.

You can no longer delete files from the EDITOR command line, although the prompts on the FILES submenu on the command line say that you can. File deletion is now done from within the EDITOR using SHOW DIRECTORY.

CONCLUSIONS:

New features are found throughout the new Funnelweb. 40 column users especially will love the new very powerful features available to them in DISK REVIEW. DISK REVIEW is a major new piece of software, not just a minor update. If you use it you should send Tony an additional fairware payment over and above what you have already paid for other parts of Funnelweb. How much additional? Well, DISK REVIEW is comparable to the latest versions of John Johnson's BOOT, maybe better. You can consult ads in recent MICROPENDIUMs for the asking price of BOOT.

DONE

IMPACT/99
by Jack Sughrue

IFFING TIW

To what state have we arrived, jargonwise, when such a title as IFFING TIW has some meaning?

If you use T.I. WRITER or any of its improved versions (T.X. WRITER, D.A. WRITER, or - by far the most superior - FUNNELWEB), you may have become familiar with the Include File structure. This Include-Filing process (known as IFfing) allows some extraordinary things to happen to your word processor. It automatically brings up a file into your text that will let you access, simply, some very complex things.

Like what?

Like that, for example! But let's say you'd like to have a term-paper style format (centered heading, right justification, wide margins, double spacing, along with an active transliteration key to underline, double strike, super/sub script and so on at will WITHIN YOUR DOCUMENT AND OPERATED FROM KEY PRESSES!), then IFfing will give it to you. Then lets say that part way through this term paper you need a large indentation and condensed type to offset that piece from the rest of the paper. And then go back to the original structure. IFfing does that by just adding a few characters before the offset piece and a few characters to return it, right in the text as you are typing.

Or say you have a series of sections to a long piece of writing: a novel perhaps. You have six chapter written, you want to load up the disk and print all six chapters saved under different filenames. With the flick of an IFfing switch you can print them ALL while you're off having dinner or taking your dog to the cheese factory. When you return your novel (to this stage) is printed out. That's IFfing.

First, you must build a file to be IFfed. Ninth easiest thing in the world. (We'll discuss the other eight in another column some time, but I will mention that breathing is Number One. So you can see the competition.)

```
.FI:AD:LM 4:RM 75:IN +3
.TL 1:27,52
.TL 2:27,53
.TL 3:27,83,0
.TL 4:27,83,1
.TL 5:27,84
.TL 6:27,66,1
.TL 7:18
.TL 8:27,87,1
.TL 9:27,87,0
.TL 10:27,66,1
.TL 20:18
.TL 12:7
.TL 21:27,45,1
.TL 22:27,45,0
.TL 15:27,71
.TL 16:27,72
.TL 17:27,69
.TL 18:27,70
```

Above, for example, is a code template I use in the PLUS! companion disk I wrote and edited for the Fairware market. As template codes go it is fairly simple. The first line FIlls, ADjusts (for right justification), Left Margins (in 4), Right Margins (in 75), and INdents (3 in from ANY Left Margin I establish during my document). This, I would assume, would be a reasonable standard for most text. I could have added (and DID on other templates) double spacing, automatic page numbering, a pre-designed HEader, or many other things. The template above is primarily to activate the TL key. After this template is SAVEd as a file (Do not type in the > sign. Begin each line with the period.), it is best to keep that tiny file on your main FUNNELWEB (or whatever) disk under a quick title. I refer to it as C3 because it is the third template I created. C2, for example, will automatically let me type out in condensed at 132 columns wide with all the TL keys intact. C4 will give me the term-paper structure I mentioned above. C1 is

strict TL. My margins and indents will be my own problem. And so on.

Okay. Let's say you've typed in and SAVEd that exact file under the filename C3 on your FUNNELWEB or DSK1.

Now, whenever you load your wordprocessor and the cursor awaits your initial command, type T (ENTER). Then put in L on 1, and I on 5, and an R on 38. This will set your screen margins within the width of the screen and will automatically create an indentation at the start of each paragraph. You'll be able to read everything ON your screen - no more windowing.

Next, press FCTN/0. This will rid the screen of line numbers and let you view your literary masterpiece unfettered.

Now your cursor is sitting in the upper left corner of your screen rarin' to go. Type the following without the parentheses: (.IF DSK1.C3) and press ENTER.

Type whatever text you want, viewing it perfectly on your screen. Feel free to use the TL keys to underline, enlarge, condense, doublestrike, superscript, italicize, letter quality, elite, whatever, whenever you wish.

The code is sitting there to automatically FORMAT this text into the original Fill/Adjust etc. you wanted. And, even though the template is tiny, you have the most extraordinary other things built in.

The TL chart is mnemonic. I built it that way when I first began to use the TL key in 1981 to help myself remember. It has been very easy, very faithful.

With that C3 sitting in DSK1., all you do to automatically call up any of this stuff is type the following: CTRL/U, SHIFT/n, CTRL/U. n is the letter that turns ON the desired printer code. The following ALPHABETICAL letter, using the same CTRL/U, SHIFT/n, CTRL/U will turn off the code.

An example would be if you wanted some words doublestruck for emphasis. You would type along regularly. Then when you came to the word or words you wanted doublestruck you would type CTRL/U, SHIFT/D, CTRL/U. Then type all the things you want in dark type. Then type CTRL/U, SHIFT/P, CTRL/U. That will shut it off, and you can go on typing to your heart's content. If D turns it on, P turns it off. Italics is turned ON by A, so S turns it off. Underlined is turned on by U, so V turns it off. If you wanted some words underlined and doublestuck and in italics with the above template, you would type CTRL/U, SHIFT/UDA, CTRL/U. To turn this batch off type CTRL/U, SHIFT/VPS, CTRL/U.

Neat, eh?

(At the end of this article is the complete mnemonic code Quick Reference Chart for the FUNL/PLUS! Template C3 shown above.)

(The TL key, by the way, can do MUCH, MUCH more than just activate printer codes in this way. But that will have to be another column.)

Now back to our text. You've type all you wanted (let's say a two-page letter) and you're ready to print. Type SF. Type DSK2.WHATEVER. After the file is SAVEd to another disk, go back to the Command line (FCTN/9) and type 3 (ENTER) and E (ENTER). If you are using FUNNELWEB you are back to the menu. Type 2 (FORMATTER). When the file comes up it'll say DSK2.WHATEVER. Turn on your printer. Press the keys all the way down and watch your printer PRINT out full-width, right-justified, fully-coded text!

But what if you wanted to print lots of files? There are lots of ways of doing this. Whatever is convenient for you.

I took the C3 template above and added the following for a special project:

```
.HE Poetry Book:THE LINK by Jack Sughrue
.FO Page I
.IF DSK2.THE/LINK1
.IF DSK2.THE/LINK2
.IF DSK2.THE/LINK3
.IF DSK2.THE/LINK4,
```

This printed out each file of my book after first going back to DSK1.C3 to see what was expected of the FORMATTER. But it would go to DSK2 to get each file itself.

I could even add

```
>.IF DSK3.THE/LINK5
>.IF DSK1.LAST/POEM
>.IF DSK.BOOK.CREDITS
```

if I wanted to, because the original IFfer will sort it all out for me. If one file is on DSK3 and another on DSK1 and another in ANY drive as long as the disk name is BOOK, it will find the file and PRINT it out.

Isn't that extraordinary?

With IFfing you can expand the horizons of your FUNNELWEB (or whatever TIW processor you're using) to exciting new dimensions. I have used the IFfing so much over the years, I have even created LF templates that make the LOADing even faster. I have a file called 2, for example, that automatically loads up the C2 code and the first few lines including condensed FORMAT codes which can't be replicated in this article. But you could actually create the following file and call it 3 (because it will draw up C3) in the automatic process:

```
.IF DSK1.C3
.CE 4
FUNLPLUS' v. 4.4
***************

by Jack Sughrue
***************
```

This automatically loads the IF and CEnters the next four lines which act as a quick heading for letters about Version 4.4. From there I can type the notes or comments or letters or article and SAVE it by its new name. When I print it out, it will call up C3 and PRINT out all my text within the C3 structure. This is great because you are automatically at the Command Mode when you enter FUNNELWEB. Just typing DSK1.3 loads up everything shown above AND the Tabs I had previously set. Thus, no more TABbing; no more typing the IF info, no more setting up the text structure. It's all in one swoop - 3. And all the FORMATting code is in C3. Beautiful.

I don't know of another wordprocessor that allows such wonderful flexibility and speed.

This system of templates would be even greater with RAMdisk and DSDD drives, of course, but all my system had when I first created and used these codes was one SSSD drive, and it sure is easy and fun.

(Jack Sughrue, Box 459, E.Douglas MA 01516)

If any newsletter editor prints these IMPACT/99 articles, please pu
me on your mailing list. Thanks - JS

OGDEN TI USERS GROUP

ANNUAL FINANCIAL REPORT
AS OF DEC. 1989

BY
RICHARD SCOTT
SECRETARY/TREASUERER

CASH ON HAND AT BEGINING	515.50
CASH DISBURSED	493.77
CASH ON HAND AT THE END	21.73

WELCOME!

TO OUR NEWEST MEMBERS

DANIEL BARKER DOUGLAS J FOWERS
WALTER LAMBERT

DOCUMENTATION FOR CASSETTE INDEX UTILITY VERSION 1.0 (XB)
CREATED BY HARRY BRASHEAR OF THE WESTERN NY 99'ERS
MODIFIED (V.2.0) BY MICKEY SCHMITT OF THE WEST PENN 99'ERS

PROGRAM NAME: CSI*INDEX (SEE LIBRARIAN)
PROGRAM DOCS: CSI*DOCS (DOCUMENTATION)
PROGRAM FILES: ROCKFILE1 AND ROCKFILE2 (BASIC SET-UPS)

THIS EXTENDED BASIC PROGRAM WILL ALLOW YOU TO PRINT OUT A NEAT LITTLE INSERT FOR YOUR CASSETTE TAPES, JUST LIKE THE ONES THAT COME WITH THEM.

ALL PRINTER CODES ARE EPSON COMPATIBLE. YOU MAY WISH TO MODIFY PROGRAM LINE 570 TO OPEN YOUR PARTICULAR PRINTER. CURRENTLY THIS PROGRAM OPENS THE FOLLOWING PRINTER DEVICE: (RS232/2.DA=8.BA=9600).

WHEN YOU FIRST BOOT UP THE PROGRAM, YOU'LL BE ASKED IF YOU WANT TO USE FILES OR INPUT THE INFORMATION DIRECTLY. (MORE ON THIS LATER). YOU WILL THEN BE ASKED FOR YOUR FULL NAME, FOR GOOD REASON. THAT LITTLE FOLD OVER THE OTHER SIDE OF YOUR TAPE WILL BE PUT TO GOOD USE BY STATING WHO THE TAPE BELONGS TO. (THE DEFAULT SPACE LIMIT IS 20 CHARACTERS).

NEIT YOU WILL BE ASKED IF YOU WANT ONE OR BOTH SIDES OF THE TAPE INDEXED. (THE DEFAULT IS 2 FOR BOTH SIDES).

YOU WILL THEN BE ASKED FOR THE CASSETTE TITLE (THE DEFAULT SPACE LIMIT IS 25 CHARACTERS), AND ANY SPECIAL INFORMATION THAT YOU MAY WISH TO RECORD. (AGAIN, THE DEFAULT SPACE LIMIT IS 25 CHARACTERS).

FINALLY, YOU WILL BE ABLE TO INPUT YOUR CASSETTE PROGRAMS, USING ANY CODING SYSTEM YOU MAY PREFER. (THE DEFAULT SPACE LIMIT IS 27 CHARACTERS).

SHOULD YOU MAKE A MISTAKE WHILE ENTERING YOUR PROGRAM DATA JUST HIT THE (ENTER KEY) AT A BLANK LINE AND YOU WILL THEN BE PROMPTED BACK TO YOUR PREVIOUS ENTRY.

WHEN YOU HAVE COMPLETED ALL OF YOUR PROGRAM DATA INPUTS, JUST TYPE (0) AND HIT THE (ENTER KEY). (NOTE: THE (0) WILL NOT APPEAR ON YOUR PRINTED INSERT).

LAST, YOU WILL BE ASKED IF YOU WOULD LIKE TO FILE OR PRINT YOUR INPUTS. (THIS SPECIAL FEATURE WAS INCLUDED BY HARRY SO THAT PEOPLE WITHOUT PRINTERS COULD FILE THE INFORMATION TILL THEY COULD BORROW A PRINTER OR CON A FRIEND INTO PRINTING THE INSERTS).

NOTE: IF YOU HAVE FILED INFORMATION STORED ON YOUR DISK YOU COULD JUST TYPE (F) AT THE FIRST PROMPT, WHICH WOULD PROMPT YOU FOR THE FILENAME YOU HAVE YOUR DATA STORED UNDER. ONCE YOU HAVE ENTERED YOUR FILENAME, JUST SIT BACK AND WATCH YOUR PRINTER GO INTO ACTION. (PROVIDED YOU DID REMEMBER TO TURN YOUR PRINTER ON).

YOU MAY ALSO MAKE MULTIPLE COPIES BY SIMPLY REPRINTING. THE "BUTTONS" ARE JUST FOR FUN, AND YOU WILL HAVE TO HOLD YOUR KEY PRESSES LONGER BECAUSE OF THEM.

(Thanks to West Penn 99'ers
US and Mickey Schmitt

SIDE 1	SIDE 2
1234567890123456789012345678	1234567890123456789012345678
2	2
3	3
4	4
5	5
6	6
7	7
8	8
9	9
0	0
1	1
2	2
3	3
4	4
5234567890123456789012345678	5234567890123456789012345678

SIDE 1: 123456789012345678901234 123456789012345678901234
SIDE 2: 123456789012345678901234 123456789012345678901234

THIS TAPE IS FROM THE LIBRARY OF 1234567890123456789012345678

ROCKFILE1

SIDE 1	SIDE 2
PROGRAM 001 - COUNTER 000 B	PROGRAM 016 - COUNTER 000 B
PROGRAM 002 - COUNTER 010 B	PROGRAM 017 - COUNTER 010 B
PROGRAM 003 - COUNTER 020 B	PROGRAM 018 - COUNTER 020 B
PROGRAM 004 - COUNTER 030 B	PROGRAM 019 - COUNTER 030 B
PROGRAM 005 - COUNTER 040 B	PROGRAM 020 - COUNTER 040 B
PROGRAM 006 - COUNTER 050 B	PROGRAM 021 - COUNTER 050 B
PROGRAM 007 - COUNTER 060 B	PROGRAM 022 - COUNTER 060 B
PROGRAM 008 - COUNTER 070 B	PROGRAM 023 - COUNTER 070 B
PROGRAM 009 - COUNTER 080 B	PROGRAM 024 - COUNTER 080 B
PROGRAM 010 - COUNTER 090 B	PROGRAM 025 - COUNTER 090 B
PROGRAM 011 - COUNTER 100 B	PROGRAM 026 - COUNTER 100 B
PROGRAM 012 - COUNTER 110 B	PROGRAM 027 - COUNTER 110 B
PROGRAM 013 - COUNTER 120 B	PROGRAM 028 - COUNTER 120 B
PROGRAM 014 - COUNTER 130 B	PROGRAM 029 - COUNTER 130 B
PROGRAM 015 - COUNTER 140 B	PROGRAM 030 - COUNTER 140 B

SIDE 1: WEST PENN 99'ERS CASSETTE LIBRARY PROGRAMS 001--015
SIDE 2: WEST PENN 99'ERS CASSETTE LIBRARY PROGRAMS 016--030

THIS TAPE IS FROM THE LIBRARY OF THE WEST PENN 99'ERS LIBRARY

ROCKFILE2

```
100 !************************
110 !                        *
120 !CASSETTE INDEX UTILITY  *
130 !          BY            *
140 !    HARRY BRASHEAR      *
150 !    WESTERN NY 99ERS    *
160 !                        *
170 !************************
180 !                        *
190 !    VERSION 2.0 (XB)    *
200 ! ENHANCED AND MODIFED   *
210 !          BY            *
```

```
220 !    MICKEY SCHMITT     *
230 !    WEST PENN 99'ERS   *
240 !      JUNE 13 1989     *
250 !                       *
260 !***********************
270 ON WARNING NEXT :: CALL
MAGNIFY(3)
280 GOTO 290 :: OPTION BASE
1 :: A$,A1$,A2$,AN$,F$,FN$,L
$,L1$,N$,PF$,T1$,T2$ :: I,J,
K,S,S1,S2,SS :: CALL CHAR ::
 CALL CLEAR :: CALL COLOR ::
 CALL SCREEN :: CALL SPRITE
:: CALL KEY :: DIM SIDE1$(18
),SIDE2$(18):: !@P-
290 !
300 CALL CLEAR :: CALL SCREE
N(15):: CALL CHAR(132,"3F60C
F9FBFBFBFBFBFBFBF9FCF603FF
C06F3F9FDFDFDFDFDFDFDF9F30
6FC")
310 CALL SPRITE(#1,132,2,160
,47):: CALL SPRITE(#2,132,2,
160,119):: CALL SPRITE(#3,13
2,2,160,190)
320 CALL COLOR(13,7,15):: CA
LL CHAR(128,"000000FFFF",129
,"0000FF00FF00FF"):: L$=RPT$
(CHR$(128),28):: L1$=RPT$(CH
R$(129),28)
330 DISPLAY AT(24,3):"RECORD
  PLAYBACK   DUPE"
340 DISPLAY AT(1,1):"CASSETT
E TAPE--INDEX UTILITY":L1$:
: :"WOULD YOU LIKE TO PRINT
FROM": :"FILES OR DIRECT INP
UT? F/D D"
350 ACCEPT AT(7,28)SIZE(-1)B
EEP VALIDATE("FD"):F$ :: IF
F$="F" THEN 730 ELSE CALL SP
RITE(#1,132,10,160,47)
360 DISPLAY AT(4,1):"PLEASE
ENTER YOUR FULL NAME:":L$ ::
 ACCEPT AT(7,1)SIZE(28)BEEP:
N$ :: DISPLAY AT(8,1):L$
370 DISPLAY AT(4,1):"ENTER N
UMBER OF TAPE SIDES:2":L$ ::
 ACCEPT AT(4,28)BEEP VALIDAT
E("12")SIZE(-1):SS
380 DISPLAY AT(4,1):"ENTER C
ASSETTE TITLE: SIDE 1" :: AC
CEPT AT(9,1)SIZE(25)BEEP:T1$
 :: DISPLAY AT(4,1):"SPECIAL
 INFOMATION:  SIDE 1"
390 ACCEPT AT(10,1)SIZE(25)B
EEP:A1$
400 IF SS=2 THEN DISPLAY AT(
11,1):L$ ELSE 420
410 DISPLAY AT(4,1):"ENTER C
ASSETTE TITLE: SIDE 2":L$ ::
 ACCEPT AT(12,1)SIZE(25)BEEP
:T2$ :: DISPLAY AT(4,1):"SPE
CIAL INFOMATION:  SIDE 2" :
: ACCEPT AT(13,1)SIZE(25)BEE
P:A2$
420 DISPLAY AT(16,1)BEEP:"AL
L INFORMATION CORRECT? Y/N"
430 CALL KEY(0,K,S):: IF S=0
 THEN 430 ELSE IF K=89 THEN
440 ELSE IF K=78 THEN 270 EL
SE 430
440 A$="SIDE ONE" :: GO SUB
670
450 FOR I=5 TO 19 :: J=I-4 :
: S1=J :: ACCEPT AT(I,2)SIZE
(27)BEEP:SIDE1$(J):: IF SIDE
1$(J)="Q" THEN 480 ELSE IF S
IDE1$(J)="" THEN I=I-2
460 IF I<4 THEN I=5
470 NEXT I
480 IF SS=1 THEN 520 ELSE A$
="SIDE TWO" :: GO SUB 670
490 FOR I=5 TO 19 :: J=I-4 :
: S2=J :: ACCEPT AT(I,2)SIZE
(27)BEEP:SIDE2$(J):: IF SIDE
2$(J)="Q" THEN 520 ELSE IF S
IDE2$(J)="" THEN I=I-2
500 IF I<4 THEN I=5
510 NEXT I
520 DISPLAY AT(24,3)BEEP:" F
ILE    REDO    PRINT" :: CA
LL SPRITE(#1,132,2,160,47)
530 DISPLAY AT(24,10)SIZE(10
):"   REDO" :: CALL KEY(5,K,
S):: CALL SPRITE(#1,132,2,16
0,47,#2,132,2,160,119,#3,132
,2,160,190)
540 IF S=0 THEN 550 ELSE IF
K=70 THEN 690 ELSE IF K=80 T
HEN 560 ELSE IF K=82 THEN 66
0 ELSE 530
550 FOR I=1 TO 80 :: NEXT I
:: CALL SPRITE(#1,132,10,160
,47,#2,132,10,160,119,#3,132
,10,160,190):: GOTO 530
560 CALL SPRITE(#1,132,2,160
,47,#2,132,2,160,119,#3,132,
10,160,190)
570 OPEN #1:"RS232/2.DA=8.BA
=9600" :: PRINT #1:CHR$(27);
CHR$(64);CHR$(27);CHR$(85);C
HR$(1);
580 PRINT #1:CHR$(27);CHR$(4
8);CHR$(15);RPT$("-",70):"!"
;TAB(5);"SIDE 1";TAB(35);"!"
;TAB(39);"SIDE 2";TAB(70);"!
" :: PRINT #1:RPT$("-",70)
590 PRINT #1:"!";TAB(35);"!"
;TAB(70);"!"
600 FOR I=1 TO 17 :: IF SIDE
1$(I)="Q" THEN SIDE1$(I)=""
610 IF SIDE2$(I)="Q" THEN SI
DE2$(I)=""
620 PRINT #1:"!   ";SIDE1$(I
);TAB(35);"!   ";SIDE2$(I);T
AB(70);"!" :: NEXT I
630 PRINT #1:RPT$("-",70):"!
";"   SIDE 1:    ";T1$;" ";A
1$;TAB(70);"!";"!";TAB(70);"
!";"!";" - SIDE 2:    ";T2$;
" ";A2$;TAB(70);"!":RPT$("-"
,70)
640 PRINT #1:"!";TAB(70);"!"
:"!";"   THIS TAPE IS FROM T
HE LIBRARY OF ";N$;TAB(70);"
!";"!";TAB(70);"!":RPT$("-",
70):: CLOSE #1
650 GOTO 530
660 FOR I=1 TO 15 :: SIDE1$(
I),SIDE2$(I)="" :: NEXT I ::
 T1$,T2$,A1$,A2$="" :: GOTO
270
670 DISPLAY AT(4,1):"" :: DI
SPLAY AT(1,1):"ENTER PROGRAM
 NAMES: Q(QUIT)":L$:TAB(11);
A$
680 FOR I=5 TO 19 :: DISPLAY
 AT(I,1):")" :: NEXT I :: RE
TURN
690 CALL SPRITE(#1,132,10,16
0,47):: DISPLAY AT(1,1):"FIL
E NAME:  DSK1." :: ACCEPT A
T(1,19)BEEP SIZE(10):FN$
700 FN$="DSK1."&FN$ :: OPEN
#2:FN$,SEQUENTIAL,INTERNAL,O
UTPUT,VARIABLE 80
710 PRINT #2:N$ :: PRINT #2:
T1$,A1$ :: PRINT #2:T2$,A2$
720 FOR I=1 TO 15 :: PRINT #
2:SIDE1$(I),SIDE2$(I):: NEXT
 I :: CLOSE #2 :: GOTO 530
730 CALL SPRITE(#2,132,10,16
0,119):: DISPLAY AT(7,1):""
:: DISPLAY AT(4,1):"FILE NAM
E:  DSK1.":L$ :: ACCEPT AT(
4,19)SIZE(10):FN$ :: FN$="DS
K1."&FN$
740 CALL SPRITE(#2,132,10,16
0,119):: OPEN #1:FN$,SEQUENT
IAL,INTERNAL,INPUT ,VARIABLE
 80
750 INPUT #1:N$ :: INPUT #1:
T1$,A1$ :: INPUT #1:T2$,A2$
760 FOR I=1 TO 15 :: IF EOF(
1)THEN 780 ELSE INPUT #1:SID
E1$(I),SIDE2$(I)
770 NEXT I
780 CLOSE #1 :: GOTO 560
790 !@P+
800 END
```


TWO JOYSTICKS IN ONE.....BY Curtis Borders.....C.O.N.N.I.E.

This is how I made two joysticks out of one:
First you will need one of those surplus joystick cables. All the pins will have to be there with the exception of pin 1 and 6. (Pins 1 and 6 are not used on the TI 99/4A) You can get one at "Star Surplus" on N. High St. Columbus, OH. They sell for about $1.99.

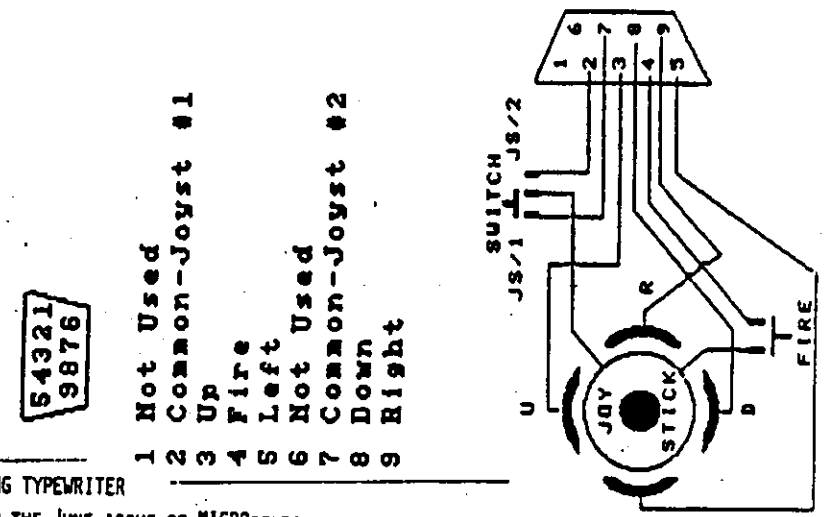
OK, now that we have the cable, take your favorite joystick- it can be Atari, Boss, EPSX500XJ, or Wico, but I wouldn't waste my time on TI joysticks. Take the joystick apart and unsolder the cable from the connectors, all but the ground (or common) wire. That's the wire that connects all the pads together.

Take your new cable and an ohm meter and write down all the pin numbers and what color wire it is, because all color codes may not be the same. Take the _____ color wire from pin 2 to one of the outside terminals of the switch, then take the _____ color wire from pin 7 to the other side of the switch.

Take a short piece of wire from the center of the switch and solder it to the ground (common) wire. If the switch is wired up right, when it is in one position, you will be using joystick number 1, and when it is in the other position, you will be using joystick number 2.

Take the _____ color wire from pin 3 to the joystick up position.
Take the _____ color wire from pin 8 to the joystick down position.
Take the _____ color wire from pin 9 to the joystick right position.
Take the _____ color wire from pin 5 to the joystick left position.
Take the _____ color wire from pin 4 to the joystick fire button. The other side of the fire button will go to the ground (or common) wire.

Use schematic below.



TALKING TYPEWRITER

FROM THE JUNE ISSUE OF MICROPENDIUM COMES THIS VERY USEFUL PROGRAM FOR THE VERY YOUNG IN THE FAMILY OR IN THE SCHOOL TO HELP THEM LEARN THE ALPHABET. CONTRIBUTED BY ELAINE CHAN PH.D OF SEATTLE WA.

EXTENDED BASIC VERSION DISPLAYS A KEY DEPRESSED IN DOUBLE-SIZED IMAGE IN THE CENTER OF THE SCREEN AND SAYS THE LETTER.

```
100 REM TALKING TYPEWRITER, E
XTENDED BASIC REQUIRED
110 CALL CLEAR
120 CALL KEY(0,K,S)
130 IF S=0 THEN 120
140 IF K>90 THEN 120
150 IF K<65 THEN 120
161 CALL CLEAR
162 CALL SPRITE(#1,K,2,85,120)
164 CALL MAGNIFY(2)
170 CALL SAY(CHR$(K))
180 GOTO 120
```

IF YOU WISH TO USE A TE-II CARTRIDGE INSTEAD OF EXTENDED BASIC, THEN REMOVE LINES 162,164, AND 170 AND INSERT :

```
165 CALL HCHAR(12,14,K)
105 OPEN #1:"SPEECH",OUTPUT
170 PRINT #1: CHR$(K)
```

EXTENDED BASIC GROM/ROM PARTS..... THIS INFO IS RIGHT OFF THE INVOICE I RECEIVED WITH THE PARTS FROM TI Dealer Parts:

PART-NUMBER	DESCRIPTION	QUANTITY	UNIT-PRICE
1015960-1113	GROM.EXT.BASIC	1	3.60
1015960-1114	GROM.EXT.BASIC	1	3.60
1015960-1122	GROM.EXT.BASIC	1	3.60
1015960-3115	GROM.EXT.BASIC	1	3.60
1041016-0006	ROM.EXT.BASIC	1	6.80
1501392-1025	ROM.EXT.BASIC	1	4.60
TOTAL FOR PARTS AVAILABLE ONLY FROM T.I.			25.80

ADDITIONAL CHIPS NEEDED 74LS00 AND 74LS74 TO BUILD XBASIC

1015960-1204	EDITOR ASSEMBLER	1	3.60

NO ADDITIONAL CHIPS NEEDED FOR EDITO ASSEMBLER

T.I. WILL NOW TAKE CHARGE CARDS, AND CHARGE LOCAL SALES TAX PLUS 3.00 FOR S/H.

I'M INCLUDING THIS UPDATED INFORMATION FOR THOSE WHO WANT TO BUILD XBASIC AND EDITOR ASSEMBLER ON THE NEW ZENO BOARD WITHOUT THE DESTRUCTION OF A CARTRIDGE. IT TAKES ABOUT 2-WEEKS TO RECEIVE THE PARTS. YOU MUST CALL Dealer Parts AT: 806 741-2265 AND ASK FOR JOE SANCHEZ.

I SOCKETED AN XBASIC CART. AND USED IT TO TEST ALL THE NEW CHIPS I RECEIVED.

INTERNAL BOARD (ZENO BOARD)

THE INTENT HERE IS TO HELP THOSE OF YOU IN THE WEST PENN 99'ERS AS WELL AS ANY IN THE USER GROUP COMMUNITY TO GET STARTED IN THE CONSTRUCTION AND INSTALLATION OF THE BOARD. I WILL TRY TO ADDRESS SOME OF THE PITFALLS AND GIVE SOME HINTS THAT MAY BE HELPFUL. I HOPE THAT YOU REALIZE THAT EVEN THOUGH ERIC ZENO HAS HAD THIS BOARD IN DESIGN FOR ABOUT A YEAR NOW, HE LIKE MOST OF US, IS DOING THIS FOR YOUR BENIFIT, AND WILL NOT QUIT OUR JOBS TO MAKE A LIVING ON TI USERS.

FIRST, THE BOARD DOES HAVE ONE ERROR IN THE TRACE LAYOUT, AND THAT IS THE DATA BUS ON THE CLOCK CHIP U12. MOST OF YOU WILL NEVER USE THIS CIRCUIT, AND THOSE OF YOU THAT DO, SHOULD BE ABLE TO INSTALL THE SIMPLE INVERSION OF THE EIGHT DATA LINES. I.E.: REVERSE THE PINS 15 THROUGH 22 UNDER THAT CHIP. I FEEL THAT THIS SHOULD STOP NO ONE FROM BUYING AND USING THE BOARD.

SECONDLY, THE TRACES ARE SMALL AND TIGHT IN AREAS THAT WILL GIVE A NOVICE FITS. DON'T BUILD IT YOURSELF IF IT LOOKS TO TIGHT FOR YOU AND YOUR EQUIPMENT.

THIRDLY, THE INITIAL INSTRUCTIONS ARE BRIEF, AND INADEQUATE FOR MOST OF YOU, THAT IS WHY I'M INCLUDING THESE TWO PAGES IN THE NEWSLETTER. YOU NEED HELP NOW! I HAVE BUILT TWO AND SINCE I DID THEM WITHOUT INSTRUCTIONS, I FEEL THAT YOU SHOULD DO MUCH BETTER WITH A LITTLE HELP.

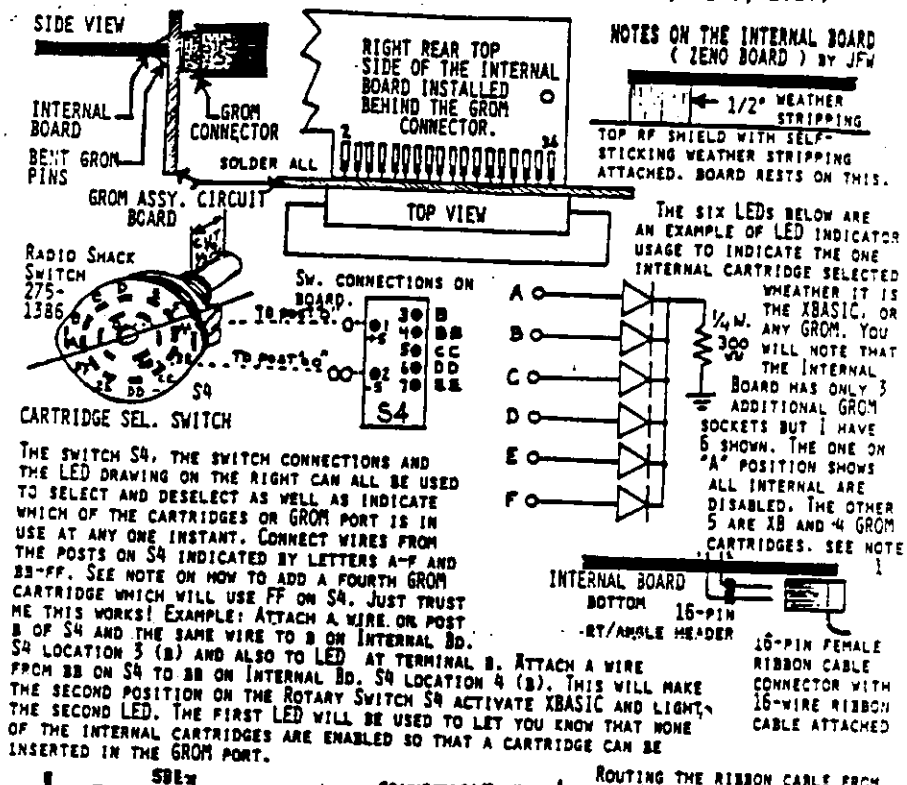
DECIDE WHAT PORTION YOU WISH TO BUILD FIRST. ATTEMPT AND GET ONE CIRCUIT WORKING AT A TIME. DO NOT FULLY SOCKET THE ENTIRE BOARD. SOCKET ONLY THE CIRCUIT YOU WILL NEED FIRST, SUCH AS THE 32K CIRCUIT. MAKE UP THE HARNESS FROM P2 TO THE CPU BOARD, SOLDER THE BOARD TO THE GROM CONNECTOR (SEE PAGE 6), AND BEFORE YOU PLUG IN P2 POWER UP THE CPU. IF ALL SEEMS WELL, THEN PLUG IN P2, THEN PLUG IN THE 32K MEMORY CHIP (FOR EXAMPLE). IF ALL IS STILL WELL, THEN RUN A MEMORY TEST BY LOADING A PROGRAM THAT USES EXPANSION MEMORY OR BY RUNNING A MEMORY DIAGNOSTIC. IF A STEP FAILS, DON'T GO ON , BUT GO BACK AND CHECK FOR A SHORT OR CHECK TO SEE IF P2 HARNESS IS WIRED CORRECTLY.

HERE IS ANOTHER AREA OF CONFUSION. THE P2 CHART ON P. 7 OF ERIC'S INSTRUCTIONS DOESN'T ADHERE TO THE PIN PROTOCOL FOR THE CONNECTOR, THEREFORE USE THE CHART AT THE RIGHT, AND ON THE ASSEMBLY DRAWING ON P. 9 OF INSTRUCTIONS CHANGE THE PIN NUMBERING SCHEME TO THAT SHOWN AT THE RIGHT ALSO. THIS WILL PROVIDE YOU WITH A MEANS TO ASSEMBLE A 16-WIRE HARNESS WITH A 16-PIN RIBBON CABLE CONNECTOR TO MATE WITH A 16-PIN DOUBLE ROW HEADER AND HAVE EVERYTHING COME OUT SO THAT THE WIRES ARE IN ORDER AND EASY TO COUNT AND MATCH CONVENTION.

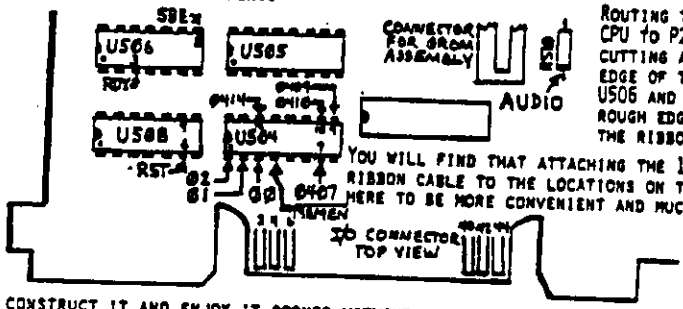
THE SPEECH SYNTHESIZER SCHEMATIC HAS SEVERAL ERRORS INCLUDING CR1 IS SHOWN BACKWARDS, THE ANODE SHOULD GO TO GROUND. AUD P2-8 SHOULD READ P2-10 (UNTIL YOU MAKE CORRECTIONS TO P2 CONN. CHART) AND RDY P2-9 SHOULD READ P2-4 AGAIN UNTIL YOU CHANGE THE P2 CONN. CHART. CR1 AND C2 MUST BE INSTALLED WITH CORRECT POLARITY OBSERVED SO TO HELP, PUT A PLUS SIGN TO LEFT OF C2 ON ASSEMBLY DRWG. AND ALL ⏚ REPRESENT THE CATHODE SIDE OF DIODES HERE.

P2 CONNECTOR (CORRECTED VERSION)

P2	TERM	USE	CPU Conn.
1	0407	32K	U504 P. 7
2	00	CK/CA	U504 P. 3
3	RDY	SPCH	U506 P. 3
4	01	CK/CA	U504 P. 2
5	SBE	SPCH	U506 P. 8
6	02	CK/CA	U504 P. 1
7	RST	SPCH	U508 P. 6
8	SPARE		
9	AUD	SPCH	R510 RIGHT
10	MEMEN-	CK/CA	U504 P. 4
11	SPARE		
12	0414	32K	U504 P.14
13	SPARE		
14	0410	32K	U504 P.10
15	SPARE		
16	0409	32K	U504 P. 9



THE SWITCH S4, THE SWITCH CONNECTIONS AND THE LED DRAWING ON THE RIGHT CAN ALL BE USED TO SELECT AND DESELECT AS WELL AS INDICATE WHICH OF THE CARTRIDGES OR GROM PORT IS IN USE AT ANY ONE INSTANT. CONNECT WIRES FROM THE POSTS ON S4 INDICATED BY LETTERS A-F AND BB-FF. SEE NOTE ON HOW TO ADD A FOURTH GROM CARTRIDGE WHICH WILL USE FF ON S4. JUST TRUST ME THIS WORKS! EXAMPLE: ATTACH A WIRE OR POST B OF S4 AND THE SAME WIRE TO B ON INTERNAL BD. S4 LOCATION 3 (B) AND ALSO TO LED AT TERMINAL B. ATTACH A WIRE FROM BB ON S4 TO BB ON INTERNAL BD. S4 LOCATION 4 (B). THIS WILL MAKE THE SECOND POSITION ON THE ROTARY SWITCH S4 ACTIVATE XBASIC AND LIGHT THE SECOND LED. THE FIRST LED WILL BE USED TO LET YOU KNOW THAT NONE OF THE INTERNAL CARTRIDGES ARE ENABLED SO THAT A CARTRIDGE CAN BE INSERTED IN THE GROM PORT.



ROUTING THE RIBBON CABLE FROM CPU TO P2 CONN. CAN BE DONE BY CUTTING A 1/4" SLOT FROM FRONT EDGE OF TOP RF SHIELD OVER THE U506 AND U505 CHIPS. TAPE THE ROUGH EDGES TO PREVENT CUTTING THE RIBBON CABLE.

YOU WILL FIND THAT ATTACHING THE 12 WIRES USED FROM P2 RIBBON CABLE TO THE LOCATIONS ON THE CPU BOARD LISTED HERE TO BE MORE CONVENIENT AND MUCH SAFER.

I HOPE THAT THESE DRAWINGS AND NOTES WILL AID ALL THOSE WHO PURCHASED AN INTERNAL BOARD TO CONSTRUCT IT AND ENJOY IT SOONER WITHOUT MUCH TROUBLE. BELOW ARE NOTES:

NOTE 1. TO ADD A 4TH GROM PIGGY-BACK IT (UP TO TWO HIGH) ON ANOTHER GROM PIN FOR PIN EXCEPT PIN 14. ATTACH A WIRE FROM PIN 14 OF THE 4TH GROM TO POST FF ON S4.

NOTE 2. DO NOT INSTALL LEDs WITHOUT A 200 OHM RESISTOR IN SERIES.

NOTE 3. THE SPEECH SYNTHESIZER HAS TWO ERRORS, BELOW U1 AUD P2-8 SHOULD BE P2-10 AND BELOW U2 RDY P2-9 SHOULD BE P2-4. (ERROR ON SCHEMATIC)

NOTE 4. MORE LATER

JOHN F. WILLFORTH 10-27-89

TI BASIC FOR BEGINNERS
by Harold Hilburn

BEGINNING PROGRAMMING ON THE TI 99/4A
IN THE IMMEDIATE MODE

Lets redo the keyboarding of last month to get a bit more experience.

PRINT Statement:
PRINT followed by a quote will cause all within the quote to be printed.

Try typing in the following: PRINT "THIS IS A MESSAGE" and press ENTER.

And PRINT "HI, THERE! I SPEAK BASIC, DO YOU?" and press ENTER

Try PRINT 3*4/2 and press ENTER

This last use of PRINT in the immediate mode causes the computer to solve the math problem (3 times four divided by 2) and print the answer.

We can use both of the these PRINT at the same time :

Try PRINT "3 Times 4 divided by 2 ="; 3*4/2 and press ENTER

Also remember, CALL CLEAR followed by ENTER clears the screen of clutter.

ERROR MESSAGES:

You'll probably get many responses from the computer of INCORRECT STATEMENT. No problem, just check to see what's wrong and use the LEFT ARROW key (FCTN S) to back space over to make corrections; or, retype the complete line correctly. You can delete items from the line by using ARROW Keys and delete with FCTN 1.

LET Statement:

The LET Statement names and assigns values to variables.

Try LET A=5
LET A2 = 8
LET ALPHA = 10

Note that the spaces around the equal sign can be used or not. The computer resolves this difference by removing them as not needed. But you must have the space after LET.

Now type PRINT A,A2, ALPHA,ALPHA/A

Note that the space ahead of the first ALPHA was self corrected by the computer. The use of the comma between multiple PRINT statements places the results in two fields of the screen. Also note that the computer still has your LET Statements in its memory. It will hold the data until you dump it or change it. Lets change ALPHA from 10 to 20. How? Its very easy.

Just type LET ALPHA=20

The last value entered replaces the earlier.

Now try PRINT A;A2;ALPHA;A*A2;ALPHA/A;A*A2*ALPHA/20

The new value for ALPHA is there and the data can be used plus we even divided by a raw number, not in the memory. Also, using the semi-colon takes the values to be spaced with only a double space between the values.

Just about anything can be used as a varable name up to fifteen characters. If we were to write a program to figure a bill we could just call it BILL.

Like, LET BILL = 25
And then PRINT "BILL =";BILL

The first BILL in quotes is printed with the enclosed = and followed by the value for BILL, in this case, 25

But what if tax had to be added?

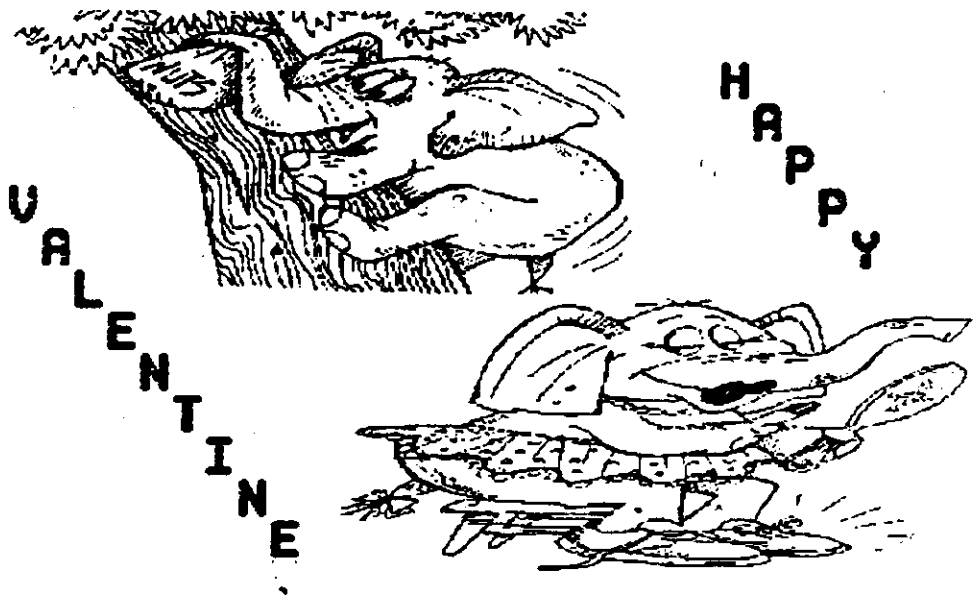
Enter TAX = .065

Then, PRINT "BILL =";BILL+BILL*TAX

Maybe the charges of the BILL would have come from several sources, such as the following. And one further point, the TI doesn't need the LET to be typed. It looks for the equal sign to establish a value file and name. Type in:

```
MATERIAL = 10
LABOR = 10
PROFIT = 5
BILL = MATERIAL + LABOR+PROFIT
PRINT "BILL =";BILL+BILL*TAX
```

continued next month



TI-BASE - From INSCEBOT

TUTORIAL 12.1.1 By Martin Smoley

NorthCoast 99'ers - Sept. 5, 1989

Copyright 1989 By Martin A. Smoley

I am reserving the copyright on this material, but I will allow the copying of this material by anyone under the following conditions. (1) It must be copied in its entirety with no changes. (2) If it is retyped, credit must be given to myself and the NorthCoast 99'ers, as above. (3) The last major condition is that there may not be any profit directly involved in the copying or transfer of this material. In other words, Clubs can use it in their newsletters and you can give a copy to your friend as long as its free.

At this time, work (the way I make money to support my TI) is taking up most of my time. I will try and write a couple example programs (CFs) to keep you going until I can get back to my TI on a regular basis. I'm sorry if the articles are a little skimpy, but it's the best I can do right now.

```
!                        DOTM
SET TALK OFF
 CLEAR
SET RECNUM OFF
SET HEADING OFF
LOCAL TEMP C 19
LOCAL TIME C 11
SELECT 5
CLOSE
USE DSK2.DT'TM
  WRITE 12,10,"TURN YOUR PRINTER ON"
  WRITE 20,10,"ENTER THE TIME"
  WRITE 22,4,"TIME EXAMPLE >12:49 P.M.<"
  WRITE 23,4,"                >          <"
  READSTRING 23,18,TIME
 IF TIME <> "              "
    TOP
   DELETE RECORD
    PACK
   APPEND BLANK
    REPLACE 5.DT WITH .DATE.
    REPLACE 5.TM WITH TIME
 ENDIF
   BOTTOM
   MOVE -1
     SET RECNUM OFF
     SET HEADING OFF
REPLACE TEMP WITH "   SYSTEM LAST RUN "
PRINT (Drft),(E),TEMP,DT,TM
   MOVE
REPLACE TEMP WITH "SYSTEM CURRENT RUN "
PRINT TEMP, DT,TM,(LF)
PRINT (Drft)
     SET TALK ON
     SET HEADING ON
     SET RECNUM ON
CLOSE
SELECT 1
CLEAR
RETURN  Copyright Martin A. Smoley 1989
!
! Save current TIME  DATE to  DT'TM
```

In this issue I have 3 CFs that do almost the same thing, but not quite. Their difference is what makes them interesting. All three of the CFs ask you for the time. They then print out the last time and date the CF was run and the current time and date which you just entered. I created it because I was printing out several copies of the same report in one evening and I couldn't tell the updates from the first printout. Having the time at the top of each printout solved my problem. This CF could be used for the last time you balanced your checkbook or paid your bills, etc. Create a very simple database named DT'TM or DT'TM2 depending on the DB you find in the USE statement in the CF you wish to use. The DB contains 2 fields. The first is named DT, type = D, with a width of 8. The second is named TM, type = C, with a width of 12. The CFs will use this DB to store the time and date for retrieval the next time the CF is run. The first CF (DOTM) is the one I use. It will allow you to set the length of the DT'TM DB by appending as many records as you wish. In other words, you could keep the last two times and dates the DB was run or the last ten if you wish. The CF will eliminate the oldest record in the file and append the newest record to the end of the file. The interesting part of this CF is the use of TOP, BOTTOM, DELETE RECORD and PACK to hold the DB at a pre-determined size. MOVE -1 and MOVE are used to locate the records to be printed.

```
!                        DOTM2
     SET TALK OFF
    CLEAR
LOCAL TEMP C 19
LOCAL TIME C 11
     SET RECNUM OFF
     SET HEADING OFF
  SELECT 5
  CLOSE
  USE DSK2.DT'TM2
     BOTTOM
  WRITE 12,10,"TURN YOUR PRINTER ON"
  WRITE 20,10,"ENTER THE TIME"
  WRITE 22,4,"TIME EXAMPLE >12:49 P.M.<"
  WRITE 23,4,"                >          <"
  READSTRING 23,18,TIME
 IF TIME <> "              "
    APPEND BLANK
     REPLACE 5.DT WITH .DATE.
     REPLACE 5.TM WITH TIME
  ENDIF
   MOVE -1
REPLACE TEMP WITH "   SYSTEM LAST RUN "
PRINT (Drft),(E),TEMP,DT,TM
   MOVE
REPLACE TEMP WITH "SYSTEM CURRENT RUN "
PRINT TEMP, DT,TM,(LF)
PRINT (Drft)
     SET TALK ON
     SET HEADING ON
     SET RECNUM ON
CLOSE
CLEAR
  SELECT 1
RETURN
!
! DOTM2  Copyright Martin A. Smoley 1989
!
! Save current TIME  DATE to  DT'TM2
```

Continued Next Page.

TI-BASE - From INSCEBOT

TUTORIAL 12.1.2 By Martin Smoley

NorthCoast 99'ers - Sept. 7, 1989

Copyright 1989 By Martin A. Smoley

Another CF would be needed to print out any extra times and dates in the DB. The CFs in this article will only print out the last record and the current time and date. The next CF (DOTM2), works almost the same as the first except for the records kept. DOTM2 works with your last two entries, but it keeps all of the previous entries. This CF would be great if you wanted to keep a complete record of the time and date a certain set of CFs was used, but you must remember that you have a self generating DB in the system. This means that each time you use the system DT'TM2 will get larger and therefore you will have less and less disk space for other types of TI-Base use.

```
!                        DOTM3
     SET TALK OFF
    CLEAR
LOCAL TEMP C 19
LOCAL TIME C 11
LOCAL TMTMP C 12
LOCAL DTTMP D 8
     SET RECNUM OFF
     SET HEADING OFF
SELECT 5
CLOSE
USE DSK2.DT'TM
     BOTTOM
  WRITE 12,10,"TURN YOUR PRINTER ON"
  WRITE 20,10,"ENTER THE TIME"
  WRITE 22,4,"TIME EXAMPLE >12:49 P.M.<"
  WRITE 23,4,"                >          <"
  READSTRING 23,18,TIME
 IF TIME <> "              "
    REPLACE DTTMP WITH 5.DT
    REPLACE TMTMP WITH 5.TM
    TOP
    REPLACE 5.DT WITH DTTMP
    REPLACE 5.TM WITH TMTMP
    BOTTOM
    REPLACE 5.DT WITH .DATE.
    REPLACE 5.TM WITH TIME
 ENDIF
  MOVE -1
REPLACE TEMP WITH "   SYSTEM LAST RUN "
PRINT (Drft),(E),TEMP,DT,TM
  MOVE
REPLACE TEMP WITH "SYSTEM CURRENT RUN "
PRINT TEMP, DT,TM,(LF)
PRINT (Drft)
     SET TALK ON
     SET HEADING ON
     SET RECNUM ON
CLOSE
CLEAR
  SELECT 1
RETURN  Copyright Martin A. Smoley 1989
!
! Save current TIME  DATE to  DT'TM
```

DOTM3 is almost identical to DOTM. It only saves two records and reads and prints in the same manner. The real difference is that it holds data in its own variable space while it moves through the DB DT'TM to replace old data with current data. It created this CF to get away from the PACK command. In certain instances this algorithm will be faster. More important it does not place any system messages on your screen. This allows you to hold messages or menu selections in place on the screen without having those annoying system messages that scroll the screen up and throw away the top line on the screen. I'm sure that most of you will find this problem minute, but the idea may help you somewhere else in your programming endeavor. Another idea which you may need from time to time is the selection of a unique number. Neither human selection or the random generation of a computer should be trusted with this task. Unique code numbers are the truest when you have the computer extract pieces of the date, the time and at least two letters of a persons name. Some companies use time,date,zipcode and names. Take a look at some of your junk mail for numbers that might follow this pattern. I mentioned the unique number uses because parts of these time, date CFs could be used to generate a unique number that you could then relate to a person or companies name in a mailing list. This number could then be used to relate two or more DBs together to gather mathematical data, as I have shown in the past. It could also be used as an access code or for other information. In other words, any of these three CFs could be converted to ask you for the time and your access code. It could then save a record of who used the system, with the time and date. There are many many uses for any one idea. You may need to slightly modify a particular CF for a new job, but its easier than writing a new one.

TI Sort

I must put in a couple of plugs for Inscebot. They have created some great software for the TI and I think that TI Sort will be close behind TI Artist and TI Base. I use it more and more as time passes. It is fast, accurate, and very versitile. If you work with any amount of data, I think you should pick up a copy of TI Sort for your collection of utilities.

TI-Base Ver. 2.03

I previously received version 2.02 for testing. I hadn't had much time to play with it when version 2.03 arrived at my door. Version 2.02 corrected a bunch of minor problems and version 2.03 corrects several more. With the latest versions of TI-Base you also have the ability to load from a hard drive and use a PATH function to find the TI-Base main program files. You may never notice some of the problems that are constantly being corrected, but they are being corrected anyway. I bring this up because I feel that the TI community is getting more support from people like Dennis Faherty than you can comprehend, at a very small cost. Please try to support the efforts of our last major software suppliers.

Continued Next Month.

THE TOKEN EXPERIMENT by: Bud Wright
December 20, 1989

Thanks to Jim Peterson for the inspiration to write this utility. It was fun!

Documentation for The Token experiment.

First, a description of the assembly support and then with the rest -

```
CALL LINK("TOK",NL,LN(),TOKS())
```

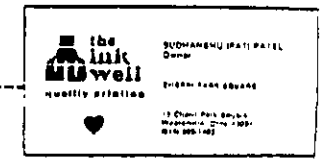
This routine looks at the whole Extended Basic program in memory and returns the total number of xb lines or tokens (NL), an array of the line numbers 1 thru NL that are the actual xb line numbers and the tokens for each of the lines in the array TOKS(). To use this routine as a standalone do the following. OLD your xb program in, then from command mode DIM LN(),TOKS() up to the maximum number of xb lines. CALL INIT::CALL LOAD ("DSK1.TOKO") and then do the CALL LINK("TOK",NL,LN(),TOKS()). Run "JIMP1" to see an example of what can be done with this utility. Running "JIMP1" displays every tokenized line of XB in the small program. You could use this data to create merged XB lines, etc.

The other link routine included allows individual access to each XB line in your program. CALL LINK("TOKEN",LN,A,T$) passes LN "XB line number" to the routine and returns A "address in cpu memory where the XB line is" and T$ "the tokenize code stored in cpu memory for that XB line. "JIMP2" runs from XB. List the program first; 10000 contains a REM with a bunch of X's. Run the program and answer with a number 1 thru 4. Something happened? Now look at line 10000 it has changed, line 1 thru 4 was copied to line 10000! Bet, you can think of plenty of neat tricks to do now! Remember, keep your dummy lines to maximum character length and you will have no problems. I also included a raw XB routine called "JIMP" that prints the cpu token pointer, the xb line, and the address of of the line table. Hope this stuff comes in handy. Enjoy!

```
1 GOTO 100
2 REM TEST JIMP
100 REM TEST
101 A=B+C
200 CALL PEEK(-31952,A,B,C,D
)
210 SLN=A*256+B
220 ELN=C*256+D
230 FOR I=SLN TO ELN STEP 4
240 CALL PEEK(I-65536,A,B,C,
D)
250 PRINT I;A*256+B;C*256+D
260 NEXT I

$$$$$$$$$$$$$$$$$$$$$$$$$$$$

1 GOTO 100
2 REM TEST  JIMP1
100 DIM N(100),T$(100)
101 A=B+C
200 CALL PEEK(-31952,A,B,C,D
)
210 SLN=A*256+B
220 ELN=C*256+D
230 FOR I=SLN TO ELN STEP 4
240 CALL PEEK(I-65536,A,B,C,
D)
250 PRINT I;A*256+B;C*256+D
260 NEXT I
270 CALL LINK("TOK",L,N(),T$
())
280 FOR I=1 TO L
290 PRINT N(I);T$(I)
292 FOR J=1 TO LEN(T$(I))
294 PRINT ASC(SEG$(T$(I),J,1
));
296 NEXT J
298 CALL KEY(0,K,S):: IF S=0
 THEN 298 ELSE PRINT :
300 NEXT I
10000 REM XXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXX

$$$$$$$$$$$$$$$$$$$$$$$$$$$

1 GOTO 100 ! JIMP2
2 PRINT :: PRINT "HELLO, thi
s is line 1" :: GOTO 112
3 PRINT :: PRINT "STOP, this
 is line 2" :: GOTO 112
4 CALL SOUND(1000,110,0,111,
0,112,0):: GOTO 111
100 CALL INIT
110 CALL LOAD("DSK1.TOKO")
111 CALL CLEAR
112 DISPLAY AT(2,1):"Line #(
1-4" :: ACCEPT AT(2,11):LN
120 CALL LINK("TOKEN",LN,A,T
$)
130 PRINT A;LEN(T$);T$;
140 FOR I=1 TO LEN(T$):: PRI
NT ASC(SEG$(T$,I,1));:: NEXT
 I :: PRINT :
142 CALL LINK("TOKEN",10000,
ADD,TOK$)
144 FOR I=1 TO LEN(T$):: CAL
L LOAD(ADD+I-1,ASC(SEG$(T$,I
,1)));: NEXT I
10000 REM ZZZZZZZZZZZZZZZZZZ
ZZZZZZZZZZZZZZZZZZZZZZZZZZ
ZZZZZZZZZZZZZZZZZZZZZZZZZZ
ZZZZZZZZZZZZZZZZZZZZZZZZZZ
ZZZZZZZZZZZZZZZZZZZZZZZZZZ
```



```
* TOKS
* FOR JIM PETERSON
* BY BUD WRIGHT
* 12-19-89

       DEF TOK,TOKEN
*
NUMASG  EQU >200B
NUMREF  EQU >200C
STRASG  EQU >2010
STRREF  EQU >2014
XMLLNK  EQU >2018
KSCAN   EQU >201C
VSBW    EQU >2020
VMBW    EQU >2024
VSBR    EQU >2028
VMBR    EQU >202C
VWTR    EQU >2030
ERR     EQU >2034

MREG    BSS 32

STROUT  BSS 256
* # OF CHARS IN STRING
TEMPT   BYTE 0
TEMPT1  BYTE 0
TEMPT2  BYTE 0
TEMPT3  BYTE 0
*
* CALL LINK("TOK",#LINES,
* LN(),TOKS())
TOK     LWPI MREG
        LI R7,1
        CLR R8
        MOV @>8C00,R5
        MOV @>8C02,R6

TOK1    C R5,R6
        JHE TOK2
        MOVB *R5+,@TEMPT
        MOVB *R5+,@TEMPT1
        MOVB *R5+,@TEMPT2
        MOVB *R5+,@TEMPT3

        MOV @TEMPT2,R4
        DEC R4

        MOVB *R4,R2
        SRL R2,8
        INC R2
        SWPB R2
        MOVB R2,@STROUT
        SWPB R2
        LI R1,STROUT+1
TOK2    MOVB *R4+,*R1+
        DEC R2
        JNE TOK2

        MOV R7,R0
        LI R1,2
        LI R2,STROUT
        BLWP @STRASG

        MOV @TEMPT,@>8C4A
        BLWP @XMLLNK
        DATA >20
        MOV R7,R0
        LI R1,2
        BLWP @NUMASG

        INC R7
        INC R8
        JMP TOK1

TOK2    CLR R0
        LI R1,1
        MOV R8,@>8C4A
        BLWP @XMLLNK
        DATA >20
        BLWP @NUMASG
OVER    LWPI >8300
        B @>6A
*
* LINK("TOKEN",L#,ADD,
* TOKEN$)
TOKEN   LWPI MREG
        CLR R0
        LI R1,1
        BLWP @NUMREF
        BLWP @XMLLNK
        DATA >1288

        MOV @ 8C4A,R9

        MOV @>8C00,R5
        MOV @>8C02,R6
TOKEN1  C R5,R6
        JHE OVER

        MOVB *R5+,@TEMPT
        MOVB *R5+,@TEMPT1
        MOVB *R5+,@TEMPT2
        MOVB *R5+,@TEMPT3

        C @TEMPT,R9
        JEQ TOKEND
        JMP TOKEN1

TOKEND  MOV @TEMPT2,R4
        DEC R0
        MOV R0,@>8C4A
        BLWP @XMLLNK
        DATA 20
        CLR R0
        LI R1,2
        BLWP @NUMASG

        MOV @TEMPT2,R4
        DEC R4
        MOVB *R4,R2
        SRL R2,8
        INC R2
        SWPB R2
        MOVB R2,@STROUT
        SWPB R2
        LI R1,STROUT+1
TOKEND  MOVB *R4+,*R1+
        DEC R2
        JNE TOKEND

        CLR R0
        LI R1,3
        LI R2,STROUT
        BLWP @STRASG
        B @OVER

        END
```



TI-ARTIST FONTSCAN

by L. Doraise, of the Ottawa U.G.

A lot of fonts are available to TI-ARTIST and C.S.G.D. fans, but only if you knew which characters were in the font, and what they looked like, without having to load the graphic program and type the whole range just to see! Peter Hoddie's FONT WRITER and GRAPHIC EXPANDER have the scan option, but perhaps you don't own them; and they have a drawback: all characters being listed together, the symbols and punctuation characters are hard to spot. So I wrote this quickie, to display the characters in four categories: upper case, lower case, digits, and others (symbols and punctuation). The letter "A" if present is drawn on the screen in actual size.

The DIV/VAR 80 font file is read line by line; when a sole character is encountered it means "this is the character defined next"; line 220 will take it's ASCII value A, and the sub HC will show it on the screen, in the appropriate category; it will also be added to a string for the printout. If the character is an "A", the program jumps to a subroutine.

To draw the "A", we Linput the next line in the file; in line 320, we extract the total columns and rows; we don't need the third value in the line, "pixel jump". Each subsequent line contains the decimal equivalents of the hex bytes for one character definition, separated by commas. These values are read and transformed into their hex value by CALL CBYTE; when the string C$ is complete, we CALL CHAR a character above 127 and CALL HCHAR it in the upper corner of the screen (for very big characters, counter K revets to 91 upon reaching character 142, already used for the screen display).

When all the file has been read, you can scan (A)nother file, (P)rint a listing (minus the graphic "A") or (Q)uit. If you wish to get a graphic dump of the screen, with the "A", you can use an assembly screen dump at this point (CALL LOAD the file before line 150; CALL LINK replaces 310-320).

In C.S.G.D. fonts the "A" will not be displayed, as I don't know yet how it's done! These fonts follow the same pattern: UC only, UC/DIGITS/OTHERS (always the same ones), UC/LC/DIG/OTHERS; the value A, the total number of char. in the file tells the story. When you wish to scan a C.S.G.D. font you MUST add the "/CH" suffix.


```
100 REM ** TI-ARTIST FONTSCA          " :: C$=CHR$(14)! print
N /.L.Doraise, Ottawa U.G. /A         320 PRINT #1:BS&B$&"  FONT:
pril 1989                              "&C$&SEG$(F$,6,10):: B$&"UP
110 ON ERROR 430 :: CALL CHA          PER CASE: "&UC$:B$&"LOWER C
R(142,"000000FF",143,"007E42          ASE:  "&LC$(B$&B$&"DIGITS:
5A5A5A427E") :: B$="  "               "&DT$:B$&B$&"OTHERS:  "&SB$
120 B$=RPT$(" ",8) :: LS=RPT$         330 PRINT #1:"------" :: GOT
(CHR$(142),28) :: LD$=B$&SEG$         O 280
(LS,1,12) :: LS=B$&SEG$(LS,1          340 ! ** sub draw "A" **
,20) :: FS="1                         350 DISPLAY AT(1,19)BEEP:"A
130 GOTO 150 :: A$,B$,C$,DT$          " :: LINPUT #1:A$ :: C=VAL(S
,F$,LC$,P$,SB$,UC$,A,C,DT,HB          EG$(A$,1,1)) :: R=VAL(SEG$(A$
,K,LB,LC,P,R,RS,S,SB,UC,V,X           ,3,9)) :: K=127
,Y                                    360 FOR X=1 TO R :: FOR Y=1
140 CALL HCHAR(1,1,65,32*32) :        TO C :: LINPUT #1:A$ :: S=1
:: CALL CHARSET :: CALL ERR(          :: C$=
0,P                                   370 P=POS(A$," ",S) :: IF P=0
150 DISPLAY AT(6,1)ERASE ALL          THEN P=LEN(A$)+1
:"UPPERCASE":LS$ :: LS$="LOWE         380 V=VAL(SEG$(A$,S,P-S))
RCASE":LS:LS$ :: LD$="DIGITS          :: IF V=0 THEN C$=C$&"00" :: GO
":LD$ :: LS$="OTHERS":CHR$(14         TO 400
)&"space":LS$ :: DISPLAY AT(1         390 HB=INT(V/16) :: CALL CBYT
6,1)ERASE ALL:"SCAN WHICH FONT?       E(HB,C$) :: LB=V-16*HB :: CAL
" :: DISPLAY AT(24,1):"CHARSET        L CBYTE(LB,C$)
: UC$,LC$,DT$,SB$                     400 S=P+1 :: IF C$=ER$(A$)TH
                                      EN 370
170 ACCEPT AT(3,4)SIZE(-12)           410 CALL CHAR(K,C$) :: CALL H
BEEP:FS$ :: FS=DSK&FS$                CHAR(X,Y+23,K) :: K=K+1 :: IF
171 IF POS(FS,".",1)<6 THEN           K=142 THEN X=X+1 :: IF X=33 T
FS=FS&".C" :: GOSUB 423 :: GOTO       HEN GOSUB 423 :: GOTO 180 ELS
180                                   E 420
180 IF POS(FS,".",1)<6 THEN           420 NEXT Y :: NEXT X :: RETU
FS=FS&".T                             RN
190 OPEN #1:FS :: INPUT #1:           422 ! ** sub SCAN **
C,A,DT,SB,I                           423 OPEN #1:FS :: INPUT #1:IRN
isplay column                         C,A,C,P,A :: IF A=0 THEN 423 E
200 ! ** scan file **                 LSE IF A=VAL(VARIABLE 254)<>C
210 LINPUT #1:B$ :: IF EOF(1)         424 IF A<28 THEN 428 ELSE IF
THEN 280                              A>44 THEN 425 ELSE 426
220 IF LEN(B$)<>3 THEN 210 EL         425 LC$="abcdefghijklmnopqrs
SE A=ASC(B$)                          tuvwxyz" :: DISPLAY AT(5,1)
LEN=1                                 426 DT$="0123456789" :: SUBSP
230 IF A<65 OR A>90 THEN 240          ELAY AT(17,1):DT$
CALL HC(8,UC,A,UC$):: GOTO            427 SB$="!#$%&'()*+,-./:;<=>?
265 THEN 210 ELSE GOSUB 350           Y AT(21,1):SB$
: GOTO 210 ! upper case               428 UC$="ABCDEFGHIJKLMNOPQRS
draw it                               TUVWXYZ" :: DISPLAY AT(18,2)
240 IF A<97 OR A>122 THEN             UC$
CALL HC(13,LC$,A,LC$):: GOTO          429 CALL HCHAR(21,12,143)
210 ! lower case                      :: SB$="sp "&SB$ :: RETURN
250 IF A>48 AND A<57 THEN             430 CALL ERR(X,Y,HB,LB) :: IF
CALL HC(17,DT$,A,DT$):: GOTO          X=130 THEN ON ERROR 440 ELSE
210 ! digits                          RETURN 470 !if error 39 I/O
260 IF A=32 THEN A=143 ! spa          go back to accept filename
ce shows as a square on scre          440 DISPLAY AT(24,1)BEEP:"ER
en                                    ROR:";X;" in Line";LB :: STO
270 CALL HC(RS,SB,A,SB$):: I          P ! other errors
F SB<30 THEN 210 ELSE SB=12           450 !@P
: RS=22 :: GOTO 210                   460 SUB HC(R,X,A,A$) :: CALL
280 CLOSE #1 :: DISPLAY AT(2          HCHAR(R,X,A) :: IF A<143 THE
4,2)BEEP:"[A]NOTHER [P]RINT           N A$=A$&CHR$(A)ELSE A$="sp "
[Q]UIT" ! end                         &A$
290 CALL KEY(3,K,S):: IF S=0          470 X=X+1 :: SUBEND
OR K<65 AND K<>81 AND K<>8            480 SUB CBYTE(X,C$) :: IF X<1
0 THEN 290                            0 THEN C$=C$&CHR$(X+48)ELSE
300 IF K=65 THEN F$=SEG$(F$,          C$=C$&CHR$(X+55)
4,2) :: GOTO 150 ELSE IF K=81         490 SUBEND
THEN END
310 OPEN #1:PIO :: B$=
```