BUG BYTES

NEWSLETTER

of

TIBUG

TI - 99/4A - BRISBANE USER GROUP INC
P.O. BOX 3051
CLONTARF MDC, QLD AUST 4019

JUNE 1992

COMING MEETINGS
26 JUNE & 31 JULY

7.30 p.m.
EAST BRISBANE STATE SCHOOL
CNR WELLINGTON RD. AND
STANLEY STREETS,
EAST BRISBANE.

COMMITTEE

PRESIDENT
Ralph Nielsen - 378 1237

VICE-PRESIDENT
Col Christensen - 284 7783

SECRETARY
Garry Christensen - 888 4857

TREASURER
Val Jones - 396 1662

DISK LIBRARIAN
John Reynolds - 357 9758

TAPE LIBRARIAN
John Peacock - 074 673 376

BOOK LIBRARIAN
Trevor Campbell - 351 3107

MODULE LIBRARIAN
John Peacock - 074 673 376

EDITOR
Garry Christensen - 888 4857

Contributions to TIBUG are invited from both members and non-members. Articles for inclusion in the succeeding monthly newsletter are required at least two weeks before the monthly meeting and may be included in that newsletter at the discretion of the Editor. If you have a disk system, please supply script on disk with diagrams separately on paper and as clear and black as possible to facilitate photocopying.

Most original articles by members of TIBUG in this newsletter are on disk and are available to other User Groups on request.

Submissions of articles, reviews, comments and letters from mebers is encouraged, however the editor would ask that members keep the following in mind.

Submissions should be about computers, the TI community in particular, or have general interest value.

The preferred media is floppy disk (any format) however cassette tape is most acceptable for those members who do not have expanded systems. Please remember that handwritten submissions have to be retyped into the computer so that they can be reproduced. Typed submissions can also be used directly if the quality of the type is suitable for photocopying.

The newsletter is produced on the weekend preceding the monthly meeting. Any submissions made after the Friday, one week before the meeting will be held over until the following month.

Submissions are best sent directly to the Editor rather than through the PO Box. The address is Garry Christensen, 18 Zammitt St, Deception Bay QLD 4508.

Contact the editor if you have any difficulties with preparing a submission or have any comments about the newsletter.

ϕϕϕϕϕϕϕϕϕϕϕϕϕϕϕϕϕϕϕϕϕϕϕϕϕϕϕϕϕϕϕϕϕϕ

CONTENTS

EDITORIAL

Half way through the year already. That means that it is tax time and it also means that I am about to start some intensive study for the next 12 months. I am in the process of shedding many of my responsibilities in the various committees that I am on and one of those is this user group. As many of you know, I will be retaining the position of secretary but I must pass the position of editor on to some one else.

With this in mind, I thought it may be a good time to describe what I have been trying to do as editor. I believe that the newsletter is the life-blood of a user group. It provides the link that keeps all members informed and allows all members to participate. My aim has been to make the newsletter relevant to all members.

Over the past couple of years I have included articles that cover everything from Basic through to assembly, TI-artist to Multiplan, games to applications. It's important to note that I do not have a particular interest in all of these subjects. When selecting articles for the newsletter I look for information that others may find useful. The editor should aim to present a varied newsletter so that every member can find at least 1 item of interest or useful fact in every newsletter.

Do you need to write anything for the newsletter? Not necessarily. It would be nice if you could put down a few words each month as a editorial. Nothing fancy, perhaps even an overview of what is in the newsletter. I tend to babble on a bit so don't feel compelled to write pages. Many other newsletters have about a paragraph from the editor.

Do you think that you can handle that job. If so, give me a call.

While I will no longer be editor, I will still be writing some things for the newsletter. I hope to carry on with the 'What's News' and 'In The PO Box' but I fear that those 2 will just about exhaust my spare time. If you have been sitting back and not writing anything because you believe that there is always some one else to do it, now's your chance. Give us a few lines. There have been a couple of terrific letters in recent editions, some more would be welcome.

Well I'll keep it short this month. I will still be venting some thoughts to you so it is not good-bye and I will be attending most meetings (some clash with lectures). I look forward to welcoming a new editor next month.

ΦΦΦΦΦΦΦΦΦΦΦΦΦΦΦΦΦΦΦΦΦΦΦΦΦΦΦΦΦΦΦΦΦΦΦ

WORD PROCESSING #3

by Col Christensen

The next stage to explain in this series is the file handling ability of our WP. That involves saving files to disk, sending them to be printed, loading from disk, and deleting files on disk or in memory. All these can be performed from the command line after escaping from the editor with FCTN/9.

FILE HANDLING

PRINTING FILES

PF is the command to send the text in the memory buffer to a printer. You are prompted for a devicename for your printer. Valid devicenames normally used are PIO (Parallel Input/Output) if you have a parallel printer or RS232 if using the less common type of printer with a serial input. In this discussion I shall use the devicename, PIO, as hardly anyone today is using a serial printer.

So for printfile devicename the entry, PIO, followed by <ENTER> will print out the whole of the text in memory just as it would appear on the screen except for printer codes. The topic of printer codes will be covered in a later issue.

But the WP is more versatile than just being able to print out the complete text buffer. You can direct the printout to cover a specific part of the whole file by preceding the devicename with numbers or letters. The following are valid input for the devicename, PIO.

1 50 PIO prints only lines from 1 to 50.
48 E PIO prints lines 48 through to the end of the text. Remember the "E" represents the last line number.

L PIO prints the line numbers as well as the text but only the first 74 characters on each line. A line would normally be up to 80 characters long but the other 6 are taken up by the 4 digits of the line number and the two spaces following it.

L 33 46 PIO prints (can you guess?), yes, the lines from 33 to 46 including the line numbers. Amazing!

F PIO prints the text in fixed 80 format. One use for this function is to edit an assembly object code file and to save it back to disk by using a disk filename instead of a printer devicename. WP text files are normally saved as VARIABLE length records.

C PIO strips out any control characters like carriage returns, line feeds or new page before printing. I.ve never used this so I must try it out. Maybe you can think of a good use for it.

DELETING FILES

P purges all text in the memory buffers if you answer "Yes" to the prompt. You then have a clean slate again to start on.

DF for delete file appeared in the TI-Writer Editor so that particular disk files could be deleted. Its function has been greatly superceded and enhanced in Funnelweb.s word processor to become a SD (show directory) command. You need to consult the Funnelweb docs to discover all its finer points of disk and file management.

SAVING AND LOADING FILES FROM DISK

Yes, SF is the abbreviated command name for this utility. A prompt then requires the devicename to save the file to. There I go again. I.m sure a preposition is not the correct thing to end a sentence with. There are additional refinements in this command like those for PrintFiles. You can save parts of your text by preceding the devicename with line numbers. Here are some valid entries:

DSK1.FILENAME
1 108 DSK1.FILENAME
26 E DSK1.FILENAME

To load a file, type LF on the command line and then at the prompt the pathname to the file on disk such as:

DSK1.FILENAME
16 120 DSK1.FILENAME loads in just those line numbers from the disk and stores them in the memory buffer numbered from line 1, of course.
0 40 DSK1.FILENAME
100 E DSK1.FILENAME

Things now get deeper. You can merge a disk file with what is already in memory with:

10 DSK1.FILENAME merges all of DSK1.FILENAME and places it after the existing line 10 and before the existing line 11. Obviously the old lines 11 onwards will now have much higher line numbers.

E DSK1.FILENAME merges all the disk file at the end of the existing file in memory.

0 DSK1.FILENAME merges all the disk file at the beginning of the existing file in memory.

And deeper yet! Merge PART of a disk file into the existing text with:

14 50 64 DSK1.FILENAME. It should be no trouble to work that one out if you remember that the first number is the line in the current memory buffer after which the insertion is to be made.

SEARCH

This group allows finding a particular "word" or replacing it with another.

FS (FindString) is followed by a prompt to input the string or word for which a search is to be made. The string must be preceded by a slash and followed by a slash. For example, to find the string "word processor" your entry would be /word processor/ . The search IS case sensitive and will not find a string in upper case (WORD PROCESSOR) when the input is in lower case (/word processor/). After pressing <ENTER>, the text will be searched from the position of the cursor onwards. If you want the whole lot to be searched, do a Show Line 1 first to get the cursor to the beginning. The search will end if successful with the cursor over the first character in the first matching string found. If not successful, the cursor will appear after the end of the last line of text.

RS (ReplaceString) requires an input of the string to be searched for as well as the

string to replace it with. An input such as /RSI/Repetitive Strain Injury/ will replace the abbreviation with the full name for the affliction. BUT when the cursor appears over the first occurrence of the string "RSI" this list of prompts can be seen on the command line - Yes, No, All or Stop?

So type
Y to replace this one and find the next.
N to ignore this one and go on to the next.
A to replace all occurrences of the string.
S to escape to the edit mode.

NOTE WELL. If in the word-wrap mode with the solid cursor, all string replacements will be accompanied by automatic reformatting of that paragraph using the current tab settings. This can be disconcerting if a particular replacement is in a section of your work that has been set out in tabular form. Reformatting will close it all up just one space between each item. To overcome this, turn word wrap off before making the change in that section.

TEXT FORMATTER

The text formatter is a program loaded separately which allows the printing of a text file according to the formatting commands that are imbedded in the text file. These commands in the text file are the ones that set the current left and right margins, paragraph indentation, page length and line spacing etc. for printing

When the formatter program is loaded, the first prompt asks for the INPUT FILENAME. This is the name under which the text file had been saved and will be the file you want to print.

The second prompt of PRINT DEVICENAME will usually be answered as PIO.LF unless you have a serial printer. Then RS232.LF will be the devicename but consult your printer manual in case some special baud rate is needed in the serial devicename. In either case the formatter itself will issue line feed commands to the printer at the required places. To all the other prompts on the formatter screen, just press <ENTER> for now to accept the defaults shown and printing should begin.

THE FORMATTER COMMANDS

The formatter commands, always in upper case and preceded by a dot (hence the name dot commands) are placed at the appropriate positions in the text on a line by themselves. More than one command can be placed on a line. There needs to be only one dot used only at the beginning of a line and each command must be separated by a semicolon. e.g. The very first line in a text file might have the following formatter commands:

.LM5;RM75;IN+5;FI;AD;PL56;CE2

The following explanations will throw some light on how these commands will influence the final printed output.

FORMATTER TEXT DIMENSION COMMANDS

LEFT MARGIN

.LM 10 sets the left margin at column 6 on the page.
.LM +5 adjusts the left margin inwards 5 columns more than the previous setting.
.LM -5 adjusts the left margin outwards 5 columns less than the previous setting.

RIGHT MARGIN

.RM 70 sets the right margin at column 70 on the printed page.
.RM +5 adjusts the right margin outwards 5 columns more than the previous setting.
.RM -5 adjusts the right margin inwards 5 columns less than the previous setting.

INDENT

.IN 8 indents the first line after a CR (carriage return) to column 8.
.IN +10 sets the indentation to 10 columns inwards from the current LM setting.
.IN -5 sets indenting to 5 columns less than the LM setting.

LINE MANIPULATION

.FI (Fill) puts as many whole words as possible on each line to fill within the left and right margin limits.
.NF (No fill) cancels the FI command and prints the part of the document following the .NF exactly as it would appear on the screen.
.AD (Adjust) in conjunction with a Fill command spreads the spacing between words so that the printed text reaches the right margin exactly. Lines ending with a CR will not be adjusted nor will they need to be. The formatter cannot differentiate between printer control codes placed within the text and ordinary text, so adjusted lines containing control codes may not completely reach the right margin.

.NA (No adjust) cancels the adjust command and the printout following it will have raggedy ends.

LINE SPACING

.LS 2 causes printing on every second line only.

PAGE LENGTH

.PL 60 prints 60 lines then starts a new page.

.PL +5 adjusts the page length relative to the previous setting

.PL -4 as above

BEGIN PAGE

.BP forces a new page break. The current .PL value is then restored and countdown starts again.

INTERNAL FORMAT COMMANDS

CENTRE TEXT

.CE centres the next line between the current left and right margins.

.CE 3 centres the next 3 lines.

SPACE

.SP leaves one blank line on the printed page.

.SP 5 leaves 5 blank lines.

The above will cover the basic intricacies in using formatting commands to manipulate the way in which some text is printed. Remember two things:

1. The Tab settings govern the way the text will look on the screen and the formatter commands in the text will control the way the text is printed.

2. On a line of formatter dot commands, an INdent with + or - relative positioning, relates to the last mentioned LM setting. On a line like the example earlier where many dot commands are placed, it is good practice to place an LM first if making changes to this margin.

I guess by now you have discovered what the given one-line formatting example above will do.

In part 4, I shall continue with formatter commands such as underscoring and overstriking, transliterates, page identification commands and file management.

WHAT'S NEWS

First the news from OPA. I have just got off the phone (today is Sunday 21st June) and they have informed me that 14 boards are ready to go and they hope to complete the others in the next couple of days. They are anxious to have the order completed before the deadline that they set last time I phoned (23rd June). I am expecting a phone call in a couple of days so I will be able to give you more information at the next meeting.

Budd Mills in the US has released a mouse for the TI, priced at $40 (US). The software is by Mike Maksimik, of MIDI fame, and it is claimed that this is the best mouse available for the TI. No other details to back up this claim.

Also from Bud Mills is Digiport by Don O'Neil. It will convert digital sound files for the MacIntosh, Amiga, PC, Atari, Sun or Next computers and play them on the TI. Playback speeds of up to 44kHz are supported. The cost is $40 and includes cables and 10 disks of sound files.

Budd Mills Services, 166 Dartmouth Dr, Toledo OH 43614-2911, USA.

On the home front, the speech interface cards that Chas has organized are now available. These cards allow you to put the speech synthasizer in the PE box. Contact Col if you would like more information.

Also from within the club comes the news of the CADET. This is the name that Col has given to his mini-expansion system. Briefly, the Cadet connects to the side of your console and gives some of the powers of an expanded system to console only users. The Cadet provides facilities to load assembly programs from tape (CS1), output to a printer through a PIO port, and has a text editor and formatter built-in, you can even store files in several banks of battery backed up memory. I'm sure Col will be having more to say about this device. There are a number in the construction stage and I believe that orders are being taken.

Still no word from Asgard about the outstanding software. I have written a letter to remind them and to order a small

amount of additional software (including the mouse, Chas).

The letter to Texaments has also been sent. I hope to have a reply and some software from them by the July meeting.

Preparations for the TI-Fair in Sydney are well under way. Looking good guys. Don't forget to put aside both November 28 and 29 if you can.

ÅÅÅÅÅÅÅÅÅÅÅÅÅÅÅÅÅÅÅÅÅÅÅÅÅÅÅÅÅÅÅÅÅÅÅÅÅÅ

SHOP

All the software listed below is in stock (yes, they have arrived). Prices are in Australian dollars and include postage.

Rock Runner	$14.00
Waterworks	$14.00
Beyond Video Chess	$11.50
Rattlesnake Bend	$ 7.75
Castle Darkholm	$ 9.75
Doom Games II	$ 7.75
Doom Games III	$ 7.75
Page Pro Pics #1	$ 8.50
Page Pro Pics #2	$ 8.50
Page Pro Pics #3	$ 8.50
Page Pro Pics #4	$ 8.50
Page Pro Pics #5	$ 8.50
Page Pro Pics #6	$ 8.50
Page Pro Pics #10	$ 8.50
Page Pro Pics #14	$ 8.50
Page Pro Pics #15	$ 8.50
Page Pro Borders #1	$ 8.50
Page Pro Borders #2	$ 8.50
Page Pro Fonts #1	$ 8.50
Page Pro Fonts #2	$ 8.50
Page Pro FX	$16.50
Page Pro Headline Maker	$11.50
Page Pro Headline Fonts #1	$ 8.50
Page Pro Headline Fonts #2	$ 8.50
Page Pro Templates #5	$ 7.70
Page Pro Templates #6	$ 7.50
Page Pro Templates #7	$ 7.50
Pix Pro	$16.00
Quick Run	$11.00
Microsoft Multiplan	$20.00
LOGO II	$20.00
TI Artist Plus	$28.00

THE "NEVER REPEAT A JOKE" PROGRAM

TIPS TO REMEMBER

(The following is by Gene Bohot from UGOC ROM - ed)

1. Do NOT try to copy a disk on a Xerox machine,
2. Do NOT buy a bigger monitor so that 80 columns will fit in TI-Writer.
3. Do NOT use a 3.5" floppy disk in your disk drive to copy short programs.
4. Do NOT use a 2-pound hammer when the prompt says "HIT ANY KEY".
5. Do NOT spray "RAID" in the console to de-bug a program.
6. When swearing at the computer, speak directly into the speech synthasizer from no more that 6 inches.
7. 45 RPM records have to be trimmed to fit into the floppy drive. It is best to have this done by a professional.
8. Do NOT use a poetry book to find the corrections for "Syntax Error".
9. Do NOT try to fix a break in a program with Super Glue.
10. Do NOT use a rubber eraser to erase a program on a disk.
11. Do NOT tell a child that pushing the joystick harder will make the sprite move faster.
12. Do NOT plug the 99/4A into 415V to get a more powerful computer.
13. Do NOT use coloured disks with a black-and-white monitor to give a colour display.
14. Do NOT soak your printer ribbons in assorted household products to try to extend their life.
15. Do NOT try to run a disk label through the printer after is has been affixed to the disk.
16. Do NOT try to convince a CI (Computer Illiterate) how useful a computer is. It wastes your time and annoys the CI.
17. When all else fails - read the Docs.
18. Do not cut or fold a floppy disk to send it through the post in a normal sized envelope.
19. FAIRWARE is not an illegal pyramid scheme: Send them the money.
20. The Gods of Software can be appeased by contributions to your local newsletter. Burnt offerings of faulty printouts are not acceptable.

THE MBX SYSTEM

Back when TI was still supporting the 4A, Milton Bradley (the games company), sold a range of games for the 4A. These games were in module form and had among them Bigfoot, Sewermania, Superfly etc.

What made these games different was that they were designed for use with the Milton Bradley eXpansion System. This was a combined speech synthesis and speech recognition unit which plugged into the joystick and cassette ports of the console.

For those who were not at the last meeting, the club bought one of these units and some software from someone going over to the US. The MBX system consists of A rectangular box, about half the size of the console with a 64 key keypad (8 by 8 matrix), two 9 pin joystick sockets (for MB's own joysticks), a 12V power socket, a 3 pole socket for the microphone and two plugs for connecting to the console.

A triple axis joystick, shaped more like a gun then a joystick. The fire button is placed where a trigger would be on a gun and there are another three buttons on top. The stick itself is also on top and in addition to X and Y movement, it features a knob which can be used to rotate on- screen objects. You can connect two of these joysticks to the MBX system.

A microphone headset, NASA style, which for some reason has a 3 pole (stereo) plug. The microphone only is active as the headphones are there only for comfort. Perhaps MB wanted to have active headphones?

When the system is turned on, it tells you that it is 'READY'. As with any TI peripheral, it must be plugged in and turned on before the console. The console comes up with the title screen and asks you to press any key. Normally this will get you to the next screen - 1 TI BASIC 2 MODULE. With the MB modules, they jump straight from the title screen to the game. Also, the MB modules do not like ramdisks that much with the system crashing occasionally.

All of the MB modules use the MBX system if it is present. Most of them also use the speech recognition feature. When a program uses this feature, the user is asked to say each word which will be used.

Once the user has been 'voiced' into the program, the game begins. The user uses voice commands, the joystick or the keypad to manoeuvre a character around the screen. This is quite fun, particularly when you can say 'Look, no hands!'

On a side note, to get this sort of equipment for an (ugh) IBM or an Amoeba, it can cost more than $500!

TIPS FROM THE TIGERCUB

#40

TIGERCUB SOFTWARE
156 Collingwood Ave.
Columbus, OH 43213

When Texas Instruments developed Extended Basic, they took away the ability of Basic to redefine or color the characters in sets 15 and 16, ASCII 144 to 159, in order to make room in memory for sprites (they did let us have color set 0 instead. That is why Basic programs which use sets 15 and 16 will crash if you try to run them in XBasic.

Finally, John Behnke published in the Chicago Times newsletter an amazing routine which gave us back those missing sets. His routine was 13 sectors long. Recently, Richard Heath published in the L.A. newsletter a shortened version. And, without having any idea how it works, I have managed to scrunch it down to only 4 sectors -

```
1 CALL BXB
29999 !BXB by Jim Peterson,
adapted from VDPUTIL2 by Joh
n Behnke/Richard Heath
30000 SUB BXB :: CALL INIT :
: CALL LOAD(8194,37,194,63,2
40)
30001 CALL LOAD(16368,80,79,
67,72,65,82,37,58,80,79,75,6
9,86,32,37,168)
30002 !
30003 FOR J=1 TO 136 :: CALL
 LOAD(9529+J,ASC(SEG$(][\[]$
,J,1))):: NEXT J :: SUBEND
30004 SUB CHAR(A,A$):: CALL
LOAD(9500,A):: CALL LINK("PO
CHAR",A$):: SUBEND
30005 SUB COLOR(A,B,C):: CAL
L LOAD(9492,8,15+A,(B-1)*16+
C-1)
30006 CALL LINK("POKEV"):: S
UBEND
```

Note than line 30002 is missing. That's because there is no way to key it in. Once again we need a program that writes a program -

```
100 FOR J=1 TO 136 :: READ A
 :: M$=M$&CHR$(A):: NEXT J
110 OPEN #1:"DSK1.BXBDATA",V
ARIABLE 163,OUTPUT :: PRINT
#1:CHR$(117)&CHR$(50)&"][\[]
$"&CHR$(190)&CHR$(199)&CHR$(
136)&M$&CHR$(0)
120 PRINT #1:CHR$(255)&CHR$(
255):: CLOSE #1
```

```
130 DATA 2,224,37,20,3,0,0,0
,2,5,48,48,2,6,37,2,205,133,
2,134,37,17
140 DATA 17,252,4,192,2,1,0,
1,2,2,37,1,2,3,18,0,212,131,
4,32,32,20
150 DATA 208,4,9,80,2,32,3,0
,2,1,37,2,2,2,0,8,2,7,11,0,2
,8,7,0,193
160 DATA 1,192,193,193,180,9
7,133,145,135,21,1,113,136,6
,198,145
170 DATA 135,21,1,113,136,21
0,70,10,198,177,137,220,198,
2,131,37,10
180 DATA 17,240,4,32,32,36,1
6,6,2,224,37,20,3,0,0,0,4,32
,32,32,4
190 DATA 192,216,0,131,124,2
,224,131,224,4,96,0,112
```

RUN that to create a file BXBDATA on the disk. Then load the BXB program, and enter MERGE DSK1.BXBDATA. The unprintable line will pop into place. SAVE this completed BXB routine in MERGE format, and merge it into any Basic-only program. If you want, the result can be run through a Compactor program and turned into multi-statement program lines for more speed.

Or, you can write an Extended Basic program using all 16 character sets for graphics and color - actually 17, because set 0 is also available. Even the characters 24 through 31 can be redefined! Craig Miller has warned against fooling around in that area of memory, but there seems to be no problem with redefining the cursor (30) or the edge character (31).

Sprites can only use characters between 32 and 143 and their color cannot be changed with CALL COLOR(#_,_). I have not found any other bugs, but have not had time for much experimenting.
(see improvement in Tips 47, comments in Tips 57. - Ed.)

Here's an easy Tigercub challenge - run this one in Basic, not Extended Basic.

```
>LIST
100 DISPLAY AT(1,1):0
>RUN
 0
 0
```

Why did it print the zero twice?

I wrote this next one primarily for blind users. It converts each PRINT or DISPLAY directly to speech output and also provides a speech prompt for INPUTs.
(see improvement in Tips 44 - Ed.)

```
100 !PRINT SPEAKER by Jim Pe
terson - to add OPEN #1:"SPE
ECH",OUTPUT and convert PRIN
T and DISPLAY statements to
PRINT #1
110 !Also writes a PRINT #1
for INPUT prompts
120 !Program to be converted
 must first be SAVEd in MERG
E format. Recommend it be RE
Sequenced before SAVEing, to
 make room for INPUT lines
130 PS$=CHR$(156)&CHR$(253)&
CHR$(200)&CHR$(1)&"1"&CHR$(1
81)
140 DISPLAY AT(3,1)ERASE ALL
:"INPUT FILENAME?":"DSK" ::
ACCEPT AT(4,4):IF$ :: OPEN #
1:"DSK"&IF$,INPUT ,VARIABLE
163
150 DISPLAY AT(5,1):"OUTPUT
FILENAME?":"DSK" :: ACCEPT A
T(6,4):OF$ :: OPEN #2:"DSK"&
OF$,OUTPUT,VARIABLE 163
160 PRINT #2:CHR$(0)&CHR$(1)
&CHR$(159)&CHR$(253)&CHR$(20
0)&CHR$(1)&"1"&CHR$(181)&CHR
$(199)&CHR$(6)&"SPEECH"&CHR$
(179)&CHR$(247)&CHR$(0)
170 LINPUT #1:M$ :: P=POS(M$
,CHR$(156),3):: A=POS(M$,CHR
$(162),3):: Z=POS(M$,CHR$(18
1),3)
180 I=POS(M$,CHR$(146),1)::
IF I=0 THEN 210 :: IF Z=0 OR
 Z<I THEN PRINT #2:M$ :: GOT
O 240
190 M2$=SEG$(M$,1,1)&SEG$(M$
,2,1)&PS$&SEG$(M$,I+1,Z-I-1)
&CHR$(0):: PRINT #2:M2$
200 PRINT #2:SEG$(M$,1,1)&CH
R$(ASC(SEG$(M$,2,1))+1)&SEG$
(M$,3,255):: GOTO 240
210 IF P+A=0 THEN PRINT #2:M
$ :: GOTO 240
220 M=MAX(P,A)
230 M$=SEG$(M$,1,2)&PS$&SEG$
(M$,M+1,255):: PRINT #2:M$
240 IF EOF(1)<>1 THEN 170 EL
SE CLOSE #1 :: CLOSE #2
250 DISPLAY AT(12,1)ERASE AL
L:"Type NEW and Enter" :: DI
SPLAY AT(15,1):"Type MERGE D
SK";OF$ :: END
```

```
****************************
      MOLLY DARLING
100 CALL CLEAR :: CALL SCREE
N(5):: FOR SE=1 TO 12 :: CAL
L COLOR(SE,16,5):: NEXT SE
110 DISPLAY AT(3,8):"MOLLY D
ARLING": :" Written and perf
ormed by": :TAB(9);"Eddy Arn
old" :: DISPLAY AT(24,1):"Pr
ogrammed by Jim Peterson"
120 FOR D=1 TO 200 :: NEXT D
 :: DISPLAY AT(12,1):"Just a
 moment.....": :".....look
ing for my music..."
130 DIM N(100),N2(100),A(250
),B(250),C(250):: F=110 :: F
OR J=1 TO 80 :: N(J)=INT(F*1
.059463094^(J-1)+.5):: NEXT
J
140 DATA 16,11,8,16,8,11,16,
4,11,18,11,8
150 DATA 20,16,11,23,11,16,2
5,21,16,28,16,21
160 DATA 23,20,16,23,16,20,2
3,11,16,23,16,11
170 DATA 20,11,16,20,16,11,2
0,8,11,20,11,8
180 DATA 20,11,16,25,16,11,2
3,11,16,20,8,4
190 DATA 18,16,10,18,10,16,1
8,16,10,18,11,16
200 DATA 18,15,11,18,9,15,18
,11,9,18,9,3
210 DATA 28,8,1,28,13,8,28,8
,13,28,13,4
220 DATA 27,20,18,27,18,20,2
0,18,12,20,12,18
230 DATA 25,21,16,25,16,21,2
5,13,16,25,16,13
240 DATA 27,23,21,27,21,23,2
7,23,18,27,18,21
250 DATA 28,23,20,28,20,23,2
8,20,16,27,16,20
260 DATA 30,21,13,28,13,21,2
7,21,13,25,13,21
```

```
270 DATA 23,20,16,23,16,20,2
0,11,16,20,16,11
280 DATA 30,23,13,28,13,23,2
3,20,13,20,13,16
290 DATA 25,21,16,25,16,21,2
5,21,16,27,16,21
300 DATA 28,23,20,20,16,11,1
8,15,11,20,11,15
310 DATA 16,11,8,16,8,11,16,
9,1,16,1,9
320 DATA 16,11,8,16,8,11,16,
1,8,16,13,1
330 DATA 25,21,16,25,16,13,2
5,13,9,25,9,4
340 DATA 23,20,16,23,16,11,2
3,11,8,23,8,4
350 DATA 21,18,11,21,11,9,21
,9,6,20,6,3
360 DATA 21,16,11,20,16,11,2
0,11,8,20,8,4
370 DATA 18,13,10,18,10,6,18
,6,1,20,13,10
380 DATA 22,18,13,28,22,18,2
7,18,22,25,22,18
390 DATA 23,18,15,23,15,11,2
3,11,6,23,6,3
400 DATA 23,21,15,23,15,11,2
3,11,9,23,9,6
410 DATA 16,13,8,16,8,13,16,
13,8,18,13,9
420 DATA 20,11,8,21,8,11,20,
11,8,18,11,6
430 RESTORE 140 :: T=16 :: G
OSUB 480 :: RESTORE 140 :: T
=4 :: GOSUB 480 :: RESTORE 1
80 :: T=12 :: GOSUB 480 :: R
ESTORE 140 :: T=16 :: GOSUB
480
440 RESTORE 210 :: T=28 :: G
OSUB 480 :: RESTORE 170 :: T
=4 :: GOSUB 480 :: RESTORE 2
50 :: T=4 :: GOSUB 480 :: RE
STORE 280 :: T=4 :: GOSUB 48
0 :: RESTORE 190 :: T=8
450 GOSUB 480 :: RESTORE 140
 :: T=16 :: GOSUB 480 :: RES
TORE 290 :: T=48 :: GOSUB 48
0 :: RESTORE 140 :: T=16 ::
GOSUB 480 :: RESTORE 410 ::
T=8 :: GOSUB 480
460 RESTORE 310 :: T=8 :: GO
SUB 480 :: GOTO 490
470 GOTO 490
480 FOR J=1 TO T :: X=X+1 ::
 READ A(X),B(X),C(X):: A(X)=
A(X)+12 :: B(X)=B(X)+12 :: C
(X)=C(X)+12 :: NEXT J :: RET
URN
490 DISPLAY AT(10,1):"Contro
l volume of 3 voices":"using
 1, 2 and 3 keys for":"loude
r and Q, W and E for":"softe
r.":""
500 DISPLAY AT(15,1):"Contro
l speed using `F' for":"fast
er and `S' for slower."
510 DISPLAY AT(18,1):"Change
 key using `A' for":"higher
and `D' for lower."
520 DISPLAY AT(21,1):"Press
`Z' for minor key, `X'":"for
 major key." :: V1,V2,V3=10
:: F,P,Y=0 :: X=200
530 FOR J=1 TO 192 :: CALL S
OUND(-999,N(A(J)-Y),V1,N(B(J
)-Y),V2,N(C(J)-Y),V3):: FOR
T=1 TO X/50 :: P=1^X :: NEXT
 T
540 CALL KEY(0,K,S):: IF S<1
 THEN 710 :: ON POS("123QWEF
SADZX",CHR$(K),1)+1 GOTO 710
,550,560,570,580,590,600,610
,620,630,650,670,690
550 V1=V1-1-(V1=0):: GOTO 71
0
560 V2=V2-2-(V2=0)*2 :: GOTO
 710
570 V3=V3-2-(V3=0)*2 :: GOTO
 710
580 V1=V1+2+(V1=30)*2 :: GOT
O 710
590 V2=V2+2+(V2=30)*2 :: GOT
O 710
600 V3=V3+2+(V3=30)*2 :: GOT
O 710
610 X=X-20-(X<2)*20 :: GOTO
710
620 X=X+20 :: GOTO 710
630 IF F=1 THEN GOSUB 700
640 Y=Y-1-(Y=-20):: GOTO 710
650 IF F=1 THEN GOSUB 700
660 Y=Y+1+(Y=6):: GOTO 710
670 IF F=1 THEN 710 :: GOSUB
 680 :: GOTO 710
680 F=1 :: Y=0 :: FOR W=3 TO
 27 STEP 12 :: N2(W)=N(W)::
N(W)=N(W-1):: N2(W+5)=N(W+5)
:: N(W+5)=N(W+4):: N2(W+10)=
N(W+10):: N(W+10)=N(W+9):: N
EXT W :: RETURN
690 IF F=0 THEN 710 :: GOSUB
 700 :: GOTO 710
700 F=0 :: FOR W=3 TO 27 STE
P 12 :: N(W)=N2(W):: N(W+5)=
N2(W+5):: N(W+10)=N2(W+10)::
 NEXT W :: RETURN
710 NEXT J :: J=192 :: FOR V
=10 TO 30 :: CALL SOUND(-999
,N(A(J)-Y),V,N(B(J)-Y),V,N(C
(J)-Y),V):: NEXT V :: FOR D=
1 TO 500 :: NEXT D :: GOTO 5
30
```

MEMORY FULL

Jim Peterson

NEW DISK DRIVE OWNER?

The world is your oyster, with many programs available at very low cost which compare well with PC programs costing LOTSD more.

THE standard graphics program is TI ARTIST at about US$25, well supported with many disks of graphics, compatible with Triton XB, The Missing Link, RLE and so on.

THE MISSING LINK is well worth having especially if you have any interest in Bit Map Graphics. It too is US$25 and a demo disk is available from the Group Disk Library.

TI BASE is THE database for the TI, a fully featured database at US$25 which would cost lots more for any other computer. It is very close to dBase II. You can use it very simply or get stuck into its command language, micros, and so on. NOT easy, if you want something easy stick to the limited format PERSONAL RECORD KEEPING module!

The only significant word processor is TI Writer, which is a line orientated text processor with mail merge capability. It has been much enhanced in the form of FUNLWEB, available from the disk library,

which includes also the TI Assembler plus a disk manager including sector editor. If you have a disk drive you must get Funlweb, and print out all the docs. You may not use all its features! but they are there.

Funlweb provides one "environment" for operating in- another is BOOT (disk library) which gives you an easy to configure menu for your disk, as well as the ability to catalogue and print from the menu screen.

All disk owners should have MCOPY (disk library) which will organise your disks in a more efficient manner, making disks with more than 16 files operate more quickly and with less wear and tear.

After that the choice is yours and depends on what YOUR interests (and storage space or pocket money) will run to. The group has an extensive disk library which is well worth perusing- disks are inexpensive, and the catalogue even cheaper- just send three disks and return postage!

Perhaps a quick list of the programs I have on my main two utility disks may help- remember I have a strong interest in graphics!

In addition to the usual Funlweb package I have:

T-SHELL, an XB utility which gives you disk catalog and file read functions in the background while you are programming in XB. Available from the disk library.

SQUEEZER a graphics utility which makes 4 tiny TI Artist pics from one full screen one, with different density levels for you to pick the best. (Disk library).

ARCHIVER- packs and unpacks backup disks for more storage on less disks. Must unpack to use! (Disk library).

UNBASHER- makes sense of XB programs with multistatement lines taken to excess, helps to unbug them! (Library).

VDP- allows you to use those early TI Basic programs which use Character Sets 15 and 16, in ExBas (library).

EXTRACTOR- allows you to quickly take out a part of an XB program for use elsewhere (Triton XB can also be used instead!)(library).

RLE- graphics display and print program, TI Artist compatible.

ARTDISK and PHOTO from Harry Brashear (commercial) to print TI Artist pics.

TEXTLOADER allows you to take a program in DV80 text format and load it directly into the computer, no editing required. (Library).

BITS AND PIECES

by Col Christensen

SPEECH ADAPTOR CARDS

The speech adaptor cards are now available. Our thanks go to Chas Bagley for the organising and the temporary outlay of capital for the production of these. For those of you who still are in the dark about the purpose for these cards, here's a short explanation. The speech synthesizer normally "hangs" off the end of the console with the PEB interface block and umbillical cord attached to the syntesizer. A somewhat bulky situation and also more prone to physical disturbance that could cause some program to lock up.

What the card achieves is the relocation of speech into the PEB thus overcoming the above situation. The interface card is about the same size overall as any other PEB card. To instal, the syntesizer box has to be opened, the metal shielding of the syntesizer circuit board removed by undoing two screws and the syntesizer circuit board then plugged into the interface card. It will fit only one way round so there is no danger of it being plugged in back to front. The interface card can then be inserted in the PEB and all is done. There is no little LED light on this card that gives a clue to which way round it is to be inserted in the PEB but it too will only fit the right way round.

Two of the cards have been sold and two have been spoken for. That leaves only four left. Phone me on 2847783 if you wish to reserve one at $33.

CADET CONSOLE EXPANSION

Yes, CADET, that's the fancy name for my pet hardware item. They're now operational, well, the few that have been finished. The price is not exactly bargain basement but then again, I do need some recompense. I calculate that by making the half dozen of the first batch and selling them, I should then be only about $800 in debt. That takes into account construction costs and time and resources involved in its development. Had I forseen the amount of time and sometimes frustration in its development, I would not have started. The two biggest hurdles were the design of

printed circuit board artwork and the making of circuit boards (imagine drilling 300 holes in each with a 0.8mm drill), and the debugging of the software in the eprom. Like PEB cards that contain ROMs for their operating systems, the CADET also needed software in an eprom. Now if there was a bug in the software code in an eprom, you can't just get to work with an eraser and a pencil to correct it. The eprom has to be erased under a special UV light for 15 to 20 minutes, the assembly code has to be changed (after finding the cause of the problem, no mean feat sometimes. One took me three days to find) and the eprom again programmed with the amended code. How many changes do you make to a Basic program you are composing when the first try doesn't measure up to your intentions? That's the number of times the poor development eprom was erased.

Anyway, the CADET is now afloat and I am quite proud of their capabilities. You might notice an "ad" elsewhere in this newsletter. If there is a big demand for them, I would consider going to the expense of having a commercial circuit board produced, then putting them together would be less of a hassle.

CASSETTE LEADS

Do you have a spare set of cassette leads to connect the computer to a cassette recorder? My brother, Bob, is looking for a couple so that his grandchildren can make use of the educational programs he is writing. See the Trading Post section for his phone number. Maybe you didn't know it, but there are three Christensen members of TIBUG.

NEXT MEETING

Just in case you are wondering, the next meeting WILL be held on the usual last Friday of the month even though it will be during the mid-year school holidays. The school secretary will be working through the school holidays and advises that she will un(pad)lock the door to Room 4 for us during the day so that we can gain access using our door key.

LETTER TO THE EDITOR

To all TI User Groups and ATICC Members,

After supporting the TI99/4A for over 9 years and later the Myarc Geneve, we few surviving members of ATICC have decided to disband the group, mainly due to the dwindling membership and lack of input and interest by the members.

With PCs now becomming cheaper and more prevalent, one by one we have succumbed to the inevitable. I now have three PCs and huge amounts of software, and find that I rarely turn my TI system on any more. Perhaps if the Geneve had turned out as I thought, then maybe the situation would be different.

I wish to thank all the TI User Groups for their newsletter and software exchanges over the past nine years and hope that they continue to have support for the TI family of computers.

Goodbye and good luck from what once was ATICC.

Fred Cugley
President
Adelaide TI Computer Club.

TIGERCUB REFORMATTER

THE TIGERCUB REFORMATTER+
Reformatter - CR Adder
LF Stripper - Blank Strip
Hyphenater - Dehyphenater
Justifier - Unjustifier
Marginater - Unmarginater
Version 1.2

by Jim Peterson

The TI-Writer or Funlweb Formatter can be used to reformat D/V80 text files to a greater or lesser line length, but it can garble the text while doing so, and I have seen many examples of such in newsletters.

To use the Funlweb Formatter for this purpose, the text must have carriage returns.

If the ampersand, the "at" sign, the caret, the asterisk followed by two numeric digits, or the period at the beginning of a line, are present in the text, printing througth the Funlweb Formatter will delete them and in some cases delete or garble the text.

When text is printed back to disk with the Formatter, it will contain large blocks of lines with nothing but a line feed, which must be manually deleted.

It will also place a line feed after every line, and convert carriage returns to line feeds. These can be stripped out by printing back to disk with the C option but, contrary to the manual, they are not really stripped out - they are converted to ASCII 32 blanks, which can cause problems in some applications. Also, the carriage returns will have been stripped.

Because of all these complications, I have never been satisfied with the Formatter. Therefore I wrote this program.

My program will reformat text which does not have carriage returns - such as the many files which are now being ported over from IBM - and will add the carriage returns, providing that headers are either indented or followed by blank lines and paragraphs are indented. It will strip the trailing blanks left by printing with the C option from the Funlweb Editor, and will strip line feeds caused by printing to disk from the Funlweb Formatter. It also strips the tab line.

It will of course also reformat text which does have carriage returns, to any greater or lesser length.

It will automatically edit and correct hyphenation that is no longer at the end of a line due to reformatting.

It will offer you the option (which Funlweb does not!) of hyphenating words. If you select this option, it will display the text and the farthest position at which a word could be hyphenated, whenever a word would be broken after two or more characters. If you elect to hyphenate, the maximum number of characters followed by a hyphen will be presented as a default.

This section features two innovations - my CALLKEY with blinking cursor (just press Y or y or N or n, no need to press Enter) and Bruce Harrison's STRACC routine. You can press Enter to accept the default, or type your own shorter string and the default string will be erased so you do not need to delete the extra characters.

An input here which is not followed by a hyphen will be rejected as a presumed error, but sometimes you will want the input to be without a hyphen, if the break is at a predetermined hyphen. In this case, just enter your input again and it will be accepted the second time.

If you select the option to justify, extra blanks will be inserted between words to align the right margin, just as TI-Writer does.

The program will optionally strip blanks inserted by previous justification, and will automatically strip them before justifying.

Text with a preset left margin cannot be properly reformatted, but the program will strip such margins. It will also optionally add a left margin to reformatted text.

The program checks the first 20 lines to find the left margin and the present line length. It is presumed that at least one of the first 20 lines will be a full line.

You can also hyphenate and/or justify and/or add carriage returns, and/or strip blanks and line feeds, and/or add or strip margins, without otherwise reformatting, by selecting a new line length the same as the old.

The program is intended primarily for reformatting back to disk, for use with multiple-column printing programs, but it will offer the option of output to the printer, and will then let you enter printer control codes.

You can even reformat to line lengths greater than 80. In this case, the printer or output file will be opened in the necessary record length.

Program listings, in any language, should never be

reformatted. They will be garbled and impossible to key in correctly.

This program also cannot properly reformat columnized text, text containing graphics or ornamentation, etc. I don't think any program could.

This program is released to the public domain with no restrictions except that no one except myself (Tigercub Software) and non-profit user groups may charge a copying fee for it.

However, if you do find this program useful, I would be grateful if you would spend a few pennies for a postcard to tell me so. I am getting very tired of contributing programs to the TI world and never hearing a word about them again. I don't want money, just a word of thanks to encourage me to keep on writing.

And, of course, if you find any bugs please let me know.

Jim Peterson (Tigercub Software) 156 Collingwood Ave., Columbus OH 43213

```
100 CALL INIT :: CALL LOAD(8
196,63,248):: CALL LOAD(1637
6,65,32,32,32,32,32,255,48):
: CALL LINK("A")
110 DIM I$(61),O$(250),T$(20
)
120 GOTO 160
130 LL,R,H$,J$,J,M,CFLAG,M$,
P$,L,Z,C$,P,X,A$,Q$,CF,IF$,O
F$,CR$,SET,K,S,U$,LM,LM$,SL$
,LMS,SLM$,WO,T$(),Y,PC,CC,RC
140 CALL CLEAR :: CALL SCREE
N :: CALL COLOR :: CALL HCHA
R
150 !@P-
160 DISPLAY AT(12,1)ERASE AL
L:"Read instructions? Y/N" :
: CALL CALLKEY(12,24,"YyNn",
Q$):: IF Q$="N" THEN 240
170 CALL CLEAR :: CALL SCREE
N(5):: FOR J=1 TO 12 :: CALL
 COLOR(J,16,1):: NEXT J
180 ON ERROR 190 :: OPEN #1:
"DSK1.FORM+DOCS" :: GOTO 200
190 RETURN 160
200 FOR J=1 TO 22 :: IF EOF(
1)THEN 220 :: LINPUT #1:M$ :
: DISPLAY AT(J,1):M$
210 NEXT J
220 DISPLAY AT(24,6):"PRESS
ANY KEY" :: DISPLAY AT(24,6)
:"press any key" :: CALL KEY
(0,K,S):: IF S=0 THEN 220
230 IF EOF(1)<>1 THEN CALL C
LEAR :: GOTO 200 ELSE CLOSE
#1
240 CALL CLEAR :: CALL SCREE
N(5):: FOR SET=0 TO 12 :: CA
LL COLOR(SET,2,16):: NEXT SE
T :: CR$=CHR$(13):: ON WARNI
NG NEXT
250 GOSUB 890
260 DISPLAY AT(8,1):"Input f
ilename?":"DSK" :: ACCEPT AT
(9,4)BEEP:IF$ :: ON ERROR 27
0 :: OPEN #1:"DSK"&IF$,INPUT
 :: GOTO 280
270 RETURN 260
280 DISPLAY AT(17,1):"Output
 to 1":"(1) Disk":"(2) Print
er" :: ACCEPT AT(17,11)SIZE(
-1)VALIDATE("12")BEEP:WO
290 IF WO=2 THEN DISPLAY AT(
17,1):"Printer? PIO":"":"" :
: ACCEPT AT(17,10)SIZE(-18):
OF$ :: GOTO 310
300 DISPLAY AT(17,1):"Output
 filename?":"DSK":"" :: ACCE
PT AT(18,4)BEEP:OF$ :: OF$="
DSK"&OF$
310 DISPLAY AT(20,1):"Do you
 want to set left     margin
? Y/N" :: ACCEPT AT(21,13)SI
ZE(1)VALIDATE("YN"):SL$ :: I
F SL$="N" THEN 330
320 DISPLAY AT(23,1):"Left m
argin how many spaces?" :: A
CCEPT AT(24,1)SIZE(2)VALIDAT
E(DIGIT):LMS :: SLM$=RPT$("
",LMS)
330 CALL CLEAR :: GOSUB 890
340 DISPLAY AT(10,1):"Reform
at to length?" :: ACCEPT AT(
10,21)SIZE(3)VALIDATE(DIGIT)
BEEP:R :: Y=MAX(80,R+LMS)
350 ON ERROR 360 :: OPEN #2:
OF$,VARIABLE Y,OUTPUT :: IF
WO=1 THEN 390 ELSE 370
360 RETURN 300
370 DISPLAY AT(10,1):"How ma
ny printer codes? 0" :: ACCE
PT AT(10,25)VALIDATE(DIGIT)S
IZE(-2)BEEP:PC :: DISPLAY AT
(10,1):"" :: RC=1
380 FOR J=1 TO PC :: ACCEPT
AT(10,RC)VALIDATE(DIGIT)BEEP
:CC :: PRINT #2:CHR$(CC);::
RC=RC+LEN(STR$(CC))+1 :: NEX
T J
390 DISPLAY AT(12,1):"Hyphen
ate? Y/N" :: ACCEPT AT(12,16
)SIZE(1)VALIDATE("YN")BEEP:H
$
400 DISPLAY AT(14,1):"Right
justify? Y/N" :: ACCEPT AT(1
4,20)SIZE(1)VALIDATE("YN")BE
EP:J$ :: IF J$="Y" THEN U$="
Y" :: GOTO 420
410 DISPLAY AT(16,1):"Strip
extra blanks? Y/N" :: ACCEPT
 AT(16,25)SIZE(1)VALIDATE("Y
N")BEEP:U$
420 DISPLAY AT(18,5)ERASE AL
L:"READING RECORD" :: DISPLA
Y AT(20,2):"REFORMATTING REC
ORD" :: DISPLAY AT(22,6):"SA
VING RECORD"
430 FOR J=1 TO 20 :: LINPUT
#1:T$(J):: LL=MAX(LL,LEN(T$(
J))):: IF EOF(1)=1 THEN 450
440 NEXT J
450 RESTORE #1 :: FOR J=1 TO
 LL :: FOR K=1 TO 20 :: IF S
EG$(T$(K),J,1)<>" " THEN LM=
J-1 :: J=LL :: K=20
460 NEXT K :: NEXT J
470 LINPUT #1:I$(1):: IF LM>
0 THEN I$(1)=SEG$(I$(1),LM+1
,255)
480 FOR J=2 TO 61 :: IF EOF(
1)THEN 530 :: LINPUT #1:I$(J
):: DISPLAY AT(18,20):J :: I
F ASC(I$(J))=128 THEN 530 EL
SE IF LM>0 THEN I$(J)=SEG$(I
$(J),LM+1,255)
490 IF POS(I$(J-1),CR$,1)<>0
 THEN 500 :: IF ASC(I$(J))=1
3 OR ASC(I$(J))=32 THEN I$(J
-1)=I$(J-1)&CR$
500 NEXT J :: M=J-2
510 IF (ASC(I$(61))=13 OR AS
C(I$(61))=32)AND POS(I$(60),
CR$,1)=0 THEN I$(60)=I$(60)&
CR$
520 IF R>LL THEN 710 ELSE 54
0
530 CLOSE #1 :: M=J-1 :: CFL
AG=1 :: IF POS(I$(M),CR$,1)=
0 THEN I$(M)=I$(M)&CR$ :: GO
TO 520 ELSE GOTO 520
```

```
540 FOR J=1 TO M :: DISPLAY
AT(20,22):J :: GOSUB 900 ::
IF U$="Y" THEN CALL UNFILL(I
$(J))
550 M$=P$&I$(J):: P$=""
560 CALL HSTRIP(M$)
570 L=LEN(M$)+(POS(M$,CR$,1)
<>0):: IF L<=R AND POS(M$,CR
$,1)<>0 THEN Z=Z+1 :: O$(Z)=
M$ :: GOTO 680 ELSE IF L<R T
HEN P$=M$&" " :: GOTO 680
580 IF L=R THEN Z=Z+1 :: O$(
Z)=M$ :: GOTO 680
590 C$=SEG$(M$,1,R):: CALL L
ASTPOS(C$," ",P)
600 IF P<>0 THEN 610 ELSE Z=
Z+1 :: O$(Z)=C$ :: M$=SEG$(M
$,R+1,255):: GOTO 570
610 IF R-P<3 THEN C$=SEG$(M$
,1,P-1):: CALL JUSTIFY(R,C$,
J$):: M$=SEG$(M$,P+1,255)::
Z=Z+1 :: O$(Z)=C$ :: GOTO 57
0
620 X=POS(M$," ",P+1):: IF X
=0 THEN X=LEN(M$)ELSE IF X=R
+1 THEN Z=Z+1 :: O$(Z)=C$ ::
 M$=SEG$(M$,R+2,255):: GOTO
570
630 IF H$="N" THEN 670
640 GOSUB 850
650 DISPLAY AT(12,1):"Hyphen
ate?" :: CALL CALLKEY(15,12,
"YNyn",Q$):: IF Q$="N" OR Q$
="n" THEN CALL HCHAR(2,1,32,
352):: GOTO 670
660 GOSUB 860 :: GOTO 570
670 GOSUB 880 :: GOTO 570
680 NEXT J
690 FOR J=1 TO Z :: DISPLAY
AT(22,20):J :: PRINT #2:SLM$
&O$(J):: NEXT J :: Z=0
700 IF CFLAG=0 THEN I$(1)=I$
(61):: GOTO 480 ELSE CLOSE #
2 :: STOP
710 FOR J=1 TO M :: DISPLAY
AT(20,22):J :: GOSUB 900 ::
IF U$="Y" THEN CALL UNFILL(I
$(J))
720 M$=P$&I$(J):: P$=""
730 CALL HSTRIP(M$)
740 IF POS(M$,CR$,1)<>0 AND
LEN(M$)<=R+1 THEN Z=Z+1 :: O
$(Z)=M$ :: GOTO 840
750 IF LEN(M$)<R THEN P$=M$&
" " :: GOTO 840
760 C$=SEG$(M$,1,R):: CALL L
ASTPOS(C$," ",P):: IF P=0 TH
EN Z=Z+1 :: O$(Z)=C$ :: M$=S
EG$(M$,R+1,255):: GOTO 740
770 IF P=R THEN C$=SEG$(M$,1
,P-1):: CALL JUSTIFY(R,C$,J$
):: Z=Z+1 :: O$(Z)=C$ :: M$=
SEG$(M$,R+1,255):: GOTO 740
780 IF R-P<3 THEN C$=SEG$(M$
,1,P-1):: CALL JUSTIFY(R,C$,
J$):: Z=Z+1 :: O$(Z)=C$ :: M
$=SEG$(M$,P+1,255):: GOTO 74
0
790 X=POS(M$," ",P+1):: IF X
=0 THEN X=LEN(M$)ELSE IF X=R
+1 THEN Z=Z+1 :: O$(Z)=C$ ::
 M$=SEG$(M$,R+2,255):: GOTO
740
800 IF H$="N" THEN 830 :: GO
SUB 850
810 DISPLAY AT(12,1):"Hyphen
ate?" :: CALL CALLKEY(12,12,
"YNyn",Q$):: IF Q$="N" OR Q$
="n" THEN CALL HCHAR(2,1,32,
352):: GOTO 830
820 GOSUB 860 :: GOTO 740
830 GOSUB 880 :: GOTO 740
840 NEXT J :: GOTO 690
850 DISPLAY AT(2,1):M$ :: DI
SPLAY AT(6,1):SEG$(M$,1,R)::
 A$=SEG$(M$,P+1,R-P-1)&"-" :
: DISPLAY AT(10,1):A$&SEG$(M
$,R,X-R+1):: RETURN
860 CALL LINK("ACCSTR",14,1,
LEN(A$),A$):: CF=CF+1 :: IF
POS(A$,"-",1)=0 AND CF=1 THE
N 860
870 CF=0 :: C$=SEG$(C$,1,P)&
A$ :: M$=SEG$(M$,P+1+LEN(A$)
-1,255):: CALL JUSTIFY(R,C$,
J$):: Z=Z+1 :: O$(Z)=C$ :: C
ALL HCHAR(2,1,32,416):: RETU
RN
880 C$=SEG$(C$,1,P-1):: CALL
 JUSTIFY(R,C$,J$):: Z=Z+1 ::
 O$(Z)=C$ :: M$=SEG$(M$,P+1,
255):: RETURN
890 DISPLAY AT(2,2):"TIGERCU
B REFORMATTER+ V1.2":"":" Re
formatter * Hyphenater Ri
ght Justifier * CR Adder Un
filler * Marginater" :: RETU
RN
900 IF SEG$(I$(J),LEN(I$(J))
,1)=" " OR SEG$(I$(J),LEN(I$
(J)),1)=CHR$(10)THEN I$(J)=S
EG$(I$(J),1,LEN(I$(J))-1)
910 IF I$(J)="" OR I$(J)=" "
 THEN I$(J)=CR$ :: RETURN EL
SE RETURN
920 !@P+
930 SUB HSTRIP(M$):: X=1
940 P=POS(M$,"- ",X):: IF P=
0 THEN SUBEXIT ELSE IF P=1 T
HEN 960
950 IF SEG$(M$,P-1,3)<>" - "
 THEN M$=SEG$(M$,1,P-1)&SEG$
(M$,P+2,255)
960 X=P+2 :: GOTO 940
970 SUBEND
980 SUB LASTPOS(A$,B$,Y):: X
,Y=0
990 X=POS(A$,B$,X+1):: IF X>
0 THEN Y=X :: GOTO 990
1000 SUBEND
1010 SUB JUSTIFY(R,C$,J$)
1020 IF J$="N" OR LEN(C$)=R
OR C$="" THEN SUBEXIT
1030 P=1
1040 X=POS(C$," ",P):: IF X=
P THEN P=P+1 :: GOTO 1040 EL
SE Y,P=X :: IF POS(C$," ",P)
=0 THEN SUBEXIT
1050 C$=SEG$(C$,1,X)&" "&SEG
$(C$,X+1,255):: IF LEN(C$)=R
 THEN SUBEXIT ELSE P=X+2
1060 X=POS(C$," ",P):: IF X=
0 THEN P=Y :: GOTO 1060 ELSE
 GOTO 1050
1070 SUBEND
1080 SUB CALLKEY(R,C,V$,K$)
1090 CALL HCHAR(R,C+2,30)::
FOR T=1 TO 3 :: CALL KEY(0,K
,S):: IF S<>0 THEN 1120
1100 NEXT T :: CALL HCHAR(R,
C+2,32):: FOR T=1 TO 3 :: CA
LL KEY(0,K,S):: IF S<>0 THEN
 1120
1110 NEXT T :: GOTO 1090
1120 IF POS(V$,CHR$(K),1)=0
THEN 1090 ELSE K$=CHR$(K)
1130 CALL HCHAR(R,C+2,32)::
SUBEND
1140 SUB UNFILL(M$):: P=1
1150 X=POS(M$," ",P):: IF X=
P THEN P=P+1 :: GOTO 1150
1160 X=POS(M$,"  ",P):: IF X
=0 THEN SUBEXIT
1170 M$=SEG$(M$,1,X)&SEG$(M$
,X+2,255):: GOTO 1160
1180 SUBEND
```

MODULE LIBRARY

GAMES

ADVENTURE ::Adventure game comes with many scenarios, including The Count, Pirate Adventure, Pyramid of Doom, Mystery Fun House, Strange Odyssey, Savage Island Series, The Golden Voyage. This cartridge requires disk drive or tape.

ALPINER ::Climb six of the world's tallest mountains and evade dangerous obstacles with ALPINER. Be careful, the Abominable Snowman is waiting for you atop Mt. Everest! Has speech.

A-MAZE-ING ::A challenging combination of maze games to test your strategy skills. Race against time through increasingly difficult mazes filled with tempting cheese, towering obstacles, and devious cats!

BLACKJACK AND POKER ::These computer-simulated card games allow betting with a bankroll you wish you had. Up to 4 players can play these card favorites.

BLASTO ::Puts you in command of an armored tank traveling through a dangerous mine field. You score points by destroying mines or blasting your opponent's tank!

CAR WARS ::It's your car against the computer's in this exciting race! Score points by out-maneuvering the computer's car as it tries to run you off the track!

MIND CHALLENGERS ::Two exciting and colorful games to challenge your powers of memory and logic. Test the limits of your musical memory with Memory Match, or try to solve the mystery of the baffling Mind Grid.

MUNCHMAN ::Four cunning Hoonos are in hot pursuit of your Munchman while he races to an energiser to change the attack. Can he make it to saftey, or does his fate lie in the mouth of the Hoonos?

PARSEC ::You are commander of the starship PARSEC under attack by the most hostile, deadly aliens in the galaxy. Your mission: destroy as many alien fighters and crusers as possible.

RETURN TO PIRATES ISLE ::More adventure as you search for the teasure. This time you can see what is happening. Good graphics.

STAR TREK ::Search out and destroy the invaders.

THE ATTACK ::You and your ship have been given a mission: Destroy the aliens before they destroy you!

TI INVADERS ::Numerous downright nasty space creatures challenge your survival instincts when they attack your world. Try to destroy these swarming invaders before they demolish your missiles.

TUNNELS OF DOOM ::Enter a world of fantasy where your instincts and imagination determine your chances of survival. Your journey is about to begin-prepare your self. Adventure game with two versions Pennies and Prizes and Quest of the King.

VIDEO CHESS ::Choose your own opponent or play against the computer on any of 3 levels. With this module, your computer can serve as a willing teacher or a challenging opponent.

VIDEO GAMES ::

VIDEO GAMES 1 ::Contains hundreds of variations of three basic games, Pot Shot, Pinball, Doodle.

YAHTZEE ::This exciting dice game combines strategy and chance. Players build points by rolling certain number combinations.

GENERAL

DEMONSTRATION ::This is a very early Demonstration of what the TI could do and is well worth a look and see how far they have advanced since 1979.

HOUSEHOLD BUDGET MANAGEMENT ::A step-by-step guide to better money management. Helps you set budget guidelines, track income and expenses, spot problem areas, keep easily accessible records. Easy to use!

HOME FINANCIAL DECISION ::A valuable, step-by-step guide to help answer your everyday financial questions. Helps you make informed decisions regarding general loans, home and car buying and personal savings. Also lets you compare difference between leasing versus buying and much more.

MUSIC MAKER ::Use the computer to compose music! Simply enter your composition, and the computer plays it back for you instantly.

PERSONAL REAL ESTATE ::Many alternative personal real estate investments can be easily evaluated with this module. It can be a valuable educational tool. Closely follows technicques used by the Realtors National Marketing Institute.(AMERICA)

PERSONAL RECORD KEEPING ::A step-by-step guide to creating, maintaining and utilizing your own computer-based filing system. Useful and convenient for a variety of applications.

VIDEO-GRAPHS ::Interact with preprogrammed graphics - or create your own designs to explore the unique colour and graphics capabillities of your computer.

EDUCATION

ADDITION ::A self-paced addition "tutor" to help your child develop strong math skills. Suitable for children from kindergarten up to grade 8.

ADDITION AND SUBTRACTION 2 ::Guides your child through addition and subtraction skills for numbers up to 18 with colourful tutorial routines and reinforcing drills.

ALLIGATOR MIX ::An arcade game format provides fun and challenge while increasing math skills in addition and subtraction of numbers from 0 to 9.

BEGINNING GRAMMAR ::Engaging and colourful activities that introduce the basic parts of speech and how they're used.

DIVISION ::A self-paced division tutor to help your child develop strong math skills

MINUS MISSION ::An arcade game format provides fun and challenge while increasing skills in subtraction.

MULTIPLICATION ::Teaches the basics of multiplication

MULTIPLICATION 1 ::Provides practice in the fundermentals of multiplication skills.

NUMBER MAGIC ::Drills basic maths using colours and graphics.

READING ON ::Teaches an understanding of maps, graphs and schedules.

SCHOLASTIC SPELLING no's. 3,4,5 ::Spelling lessons for various levels.

SUBTRACTION ::Introduces the concepts of subtraction.

TOUCH TYPING TUTOR ::Teaches touch typing and measures and improves typing speed.

****

TRADING POST

FOR SALE - As a complete unit, TIBUG's fully expanded system including console, modulator, transformer, Expansion box, RS232 card, Memory expansion card, Disk controller card, 2 slimline DSSD disk drives, and Extended Basic module and manual. $450 or $470 with free club membership. Phone Col 284 7783.

FOR SALE - Geneve 9640. Complete with manuals and lots of software. $500. Phone Larry 07 202 1884

FOR SALE - Expanded system with two consoles and Extended Basic. Has one double sided disk drive but no RS232 cord. Two full height single sided drives included but not connected. $275 the lot or make an offer. Phone Ross 285 2173.

FOR SALE - Compumate CP-80 printer, 80 column, 9pin dot matrix, 16k buffer, includes manuals, cable for TI, 2 ribbons. Quick sale $150. Phone Chas on (07) 373 6254.

WANTED - Cassette leads to interconnect console and tape recorder. Phone Bob Christensen on 284 0967.

WANTED - FOR $ALE - FOR EXCHANGE
What do you have or need that can be listed in this column? Contact Garry or Col with details.

WORD SHOOTER

by Don Steffen
(503) 873-4217

WORD SHOOTER is a program I wrote for my grandchildren and one great granddaughter to help them learn the keyboard and spelling of words. It adds color and action and sound to make it interesting for young minds. You should have the speech module connected for best use.

The program starts with a menu display that gives you options of using the wordlist in memory from data statements or typing in your own word-list. Pressing ENTER only will end the wordlist which can be up to 100 words long. You can have more than one wordlist and they can be saved to disk or cassette. Just type a number at the prompt to save each list.

A colorful screen is displayed and the words are paraded in random order across the screen with double sized sprites. Then by pressing the letter keys of each letter of the word in proper order, the sprite will be shot down. The Alpha Lock key must be pressed down. If the wrong key is pressed then it will say UHOH and start the same word over again.

Twenty words are displayed and the score is displayed, then if you type all twenty right, the screen changes to all colors then asks for Y/N? to play again. No time limit is required for pressing each key, but could be added for typing speed checking.

Runs in Extended Basic.

Courtesy of WORDPLAY

```
100 ! **********************
110 ! *                    *
120 ! *  WORD SHOOTER      *
130 ! *                    *
140 ! **********************
150 !
160 ! BY DON STEFFEN XBASIC
170 !
180 !
190 !
200 DIM W$(100),SCORE$(20)
210 DATA FOR PRESCAN
220 C,A,N,B,V,ST,R,H,K,L,S,T=0 :: Z$,TB$,BB$=""
230 CALL CHAR(140,"2424FF0000FF2424"):: CALL CHAR(141,"2424E72424E72424"):: CALL
 COLOR(14,9,2):: CALL COLOR(13,5,5)
240 CALL CLEAR :: CALL SCREEN(12):: GOTO 255 :: CALL KEY
250 CALL SAY :: CALL DELSPRITE :: CALL SPRITE :: CALL SOUND :: CALL MOTION
255 RESTORE 500 :: FOR R=1 TO 50 :: READ W$(R):: NEXT R :: GOTO 700
260 CALL SCREEN(2):: FOR V=1 TO 23 STEP 2 :: FOR H=1 TO 31 STEP 2 :: CALL HCHAR(
V,H,140,1):: CALL HCHAR(V,H+1,141,1)
262 CALL HCHAR(V+1,H,141,1):: CALL HCHAR(V+1,H+1,140,1):: NEXT H :: NEXT V :: FO
R V=5 TO 20 :: CALL HCHAR(V,5,32,24):: NEXT V
264 FOR V=5 TO 20 STEP 15 :: FOR H=5 TO 28 STEP 23 :: CALL HCHAR(V,5,130,24):: C
ALL VCHAR(5,H,130,16):: NEXT H :: NEXT V
270 CALL MAGNIFY(2):: !@P-
272 ! ************************** SPRITE NAME ************
274 TB$="WORD" :: BB$="PLAY" :: FOR B=1 TO 4 :: CALL SPRITE(#B+20,ASC(SEG$(TB$,B
,1)),16,8,B*20+88,0,0)
276 CALL SPRITE(#B+24,ASC(SEG$(BB$,B,1)),16,172,B*20+88,0,0):: NEXT B
278 RESTORE 490 :: FOR S=1 TO 20 :: READ SCORE$(S):: NEXT S
290 FOR R=1 TO 6 :: GOSUB 320 :: CALL DELSPRITE(ALL):: NEXT R :: CALL SCREEN(16)
295 FOR S=1 TO 4 :: DISPLAY AT(12,6)SIZE(20):"PRESS ALPHALOCK DOWN" :: CALL SOUN
D(300,880,1):: CALL HCHAR(12,8,32,20):: NEXT S :: GOTO 360
300 ! ************************** SPRITE LOOP ************
310 !
320 W$(R)=" &W$(R):: FOR S=1 TO LEN(W$(R)):: CALL SPRITE(#S,ASC(SEG$(W$(R),S,1)
),S+1,S*5+40,240,0,-12)
330 CALL SOUND(90,587,20):: CALL SOUND(90,739,20):: CALL SOUND(120,880,20)
340 NEXT S :: CALL HCHAR(12,8,32,20):: RETURN
350 !************************** MAIN LOOP **************
360 H=0 :: CALL DELSPRITE(ALL):: CALL SOUND(3000,3000,30):: FOR T=1 TO 20 :: RAN
DOMIZE
370 R=INT(RND*40)+13 :: GOSUB 320
380 FOR S=2 TO LEN(W$(R)):: L=ASC(SEG$(W$(R),S,1)):: IF L=32 THEN 410
390 CALL KEY(0,K,ST):: IF ST<>1 THEN 390
400 IF K=L THEN CALL MOTION(#S,100,0):: CALL SOUND(100,-5,1):: CALL DELSPRITE(#S
)ELSE 460
410 NEXT S :: H=H+1 :: DISPLAY AT(18,6)SIZE(18):"TRY ";T;" E";"GOT ";H
420 CALL DELSPRITE(ALL):: NEXT T :: GOSUB 600 :: R=7 :: W$(7)=W$(7)&STR$(H):: GO
SUB 320 :: CALL SAY("YOU+GOT",,SCORE$(H))
430 CALL DELSPRITE(ALL):: R=8 :: GOSUB 320 :: CALL SAY("OUT OF TWENTY"):: IF H<1
9 THEN 440
435 CALL SAY("#GOOD WORK#"):: R=0 :: GOSUB 320 :: GOSUB 650
440 ! ************************** REPLAY CHOICE **********
445 CALL SAY("WANT+TO PLAY+AGAIN")
450 CALL DELSPRITE(ALL):: R=11 :: GOSUB 320 :: CALL DELSPRITE(ALL):: R=12 :: GOS
UB 320
452 CALL KEY(0,K,ST):: IF ST=0 THEN 452
455 CALL HCHAR(18,8,32,20):: IF K<>89 THEN END ELSE 360
460 ! ************************** MISS MESSAGE ***********
470 CALL SAY("UHOH"):: DISPLAY AT(12,6)SIZE(20):"YOU MISSED THE ";CHR$(L):: H=H-
```

Extended BASIC Tips

by Bob Relyea

Courtesy of TIsHUG

Making A Menu
Part 1

One of the most useful ways of 'moving around' a program that has several things to choose from in execution is the menu method. Such programs are said to be 'menu driven'. This means that you are continually going to and from (or in and out of) a menu while doing things while the program is running. This is really neat as it saves you from having to boot up the program each time to want to run a particular routine. It is really easy to do and involves the following basic steps:

1. Make the menu page by using Display At statements (or Print if you want to),
2. Send the execution of each number on the menu off to a different part of the program (called a subroutine),
3. At the end of each subroutine direct the flow of the program back to the menu line for another choice.

And believe it or not, that is the long and the short of it! Of course, to explain it in more detail with a working example will take more than a few lines, but the general idea is real simple. The best way to learn how to do this type of thing is to do what you can do after studying this example and experiment by adding just a little bit at a time to see what happens. That Grade Standardising program that I put in a few months ago was made in just that way. When I finally got one part working, I would say to myself, "it would be nice if I could add this feature to my program", and went about working out how to do it. This would sometimes involve asking questions.

Let us look at each of the above three steps in more detail and then we will put it all together for a short program which is menu driven.

1. Using the Display At statement to make a menu. We are going to have a short program with three choices, so we will need three subroutines. This is just a fancy way of saying that we send the computer off to a different line number to do the routine asked for and it automatically skips all the rest. Here is an example of a menu page:

```
100 DISPLAY AT(2,1):"1. CALCULATE AREA."
110 DISPLAY AT(5,1):"2.  CALCULATE VOLUME."
120 DISPLAY AT(8,1):"3. CALCULATE SURFACE AREA."
```

If you type this in and want to see it on the screen after entering 'run' then you will have to temporarily type in line 130 with something like the following on it so it remains on the screen for a while:

```
130 FOR X = 1 TO 1000 :: NEXT X
```

This is a way of making the computer 'slave away' at counting while you examine the goodies.

Now you have to admit, that is not hard! However, there is a more compact way of going about it that, in this simple case, only requires one programming line to accomplish. Here it is:

```
100 DISPLAY AT(2,1):"1. CALCULATE
AREA,": :"2. CALCULATE VOLUME,": :"3.
CALCULATE SURFACE AREA."
```

By placing the ":" symbols the way you see in line 100 with a space between each pair, the computer automatically places the next statement two lines after the previous one. In other words, choice two is place two lines (double spacing) after choice one on the screen. As far as I know, you cannot place them any closer using this method but you can space them further by using an additional ":" and a space for each additional spacing that you want. So, is you want triple spacing then instead of : :, use : : : , and so on for greater spacings.

Following the ": :" the line will be place at the beginning of the next line, that is on column 1. This could be a bit awkward for those like me who have an eprom fitted to their system that places everything too far to the left on the screen. To get around this write your line like the following:

```
100 DISPLAY AT(2,2):"1.  CALCULATE
AREA,": :" 2.  CALCULATE VOLUME,": :
"3.  CALCULATE SURFACE AREA."
```

You may notice that position (2,2) has replaced (2,1) which shifts the first line one column to the right. To make sure that the remaining lines (that is 2. and 3.) are placed one column to the right as well, you have to put a space before the number. Note the way that I did this in new line 100. This will ensure that everything lines up properly on the screen. This procedure can be repeated for any number of columns that you want.

Step two and three of the menu-making procedure will be covered next month as step two involves more than meets the eye and will make use of the CALL KEY statement. o

IN THE P.O. BOX

TI-Shug News Digest, June 1992: Front cover - A message to Gary Bowser of OPA, Editor's Comment, Co-ordinator's Report, Treasurer's Report, Secretaries Notebook, Decoding Eprom Files by Ben Takach, TIM review, TI-Shug Shop, Jenny's Younger Set, Software, TI-Bits and XB Tips by Jim Swedlow, To See or Not To C by Geoff Trott, Writing in Machine Code by J.E. Banfield, Mathamatics Series reviewed, Programming Music by Jim Peterson, Sorting by Ron Brubaker, How to Use Arrays by Andy Frueh, TI-Base Tutorial, Beginning Forth, Newsletter Update.

KC99er (Kansas), Jan/Feb 1992: Calenders, Programmer's Humor, The Blood Bank, One Liners, Tips to Remember, The Timing is Off, Why Didn't It Indent, Kinder Korner.

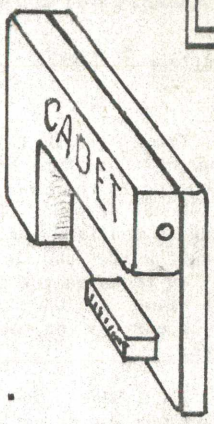
Mar/Apl: TI World News, The Blood Bank.

May/June: New Standards Proposed for the TI-99/4A, Tom's Observations, The Blood Bank, Defragmentation, TI World News, Kinder Korner.

Micropendium, February 1992: Comments, Feedback, Basketball Statistics (Bas), Subindex Two (XB), Art of Assembly, Reader to Reader, A Fast Sort Routine (c99), Tigercub Reformatter, Newsbytes, Micro-Reviews, Newsbytes.

CADET

CONSOLE EXPANSION



Plugs into I/O port on right of the console.
Compatable with all software from tapes and modules.
Features:-
PRINTER PORT. Drive a printer directly from Basics or print text files.
SOFTWARE IN ROM:
Word Processor like the original TI-Writer. Type and print letters, assignments etc.
Text Formatter like original TI version. Add "class" to your printed text.
Tape Games Loader to load and run specially taped assembly games.
32K MEMORY EXPANSION for use by the above software.
ADDITIONAL 32K RAM divided into two devices to store text for later reuse. Supercap backup maintains power to memory for weeks.
I/O EXTENSION. Attach a speech synthesizer or ramdisk. Ramdisk requires optional adaptor card.

Limited number available to order at $100 each.
Ramdisk adaptor card $20.
Contact Col Christensen on (07)2847783.

TI BRISBANE USER GROUP
P.O. BOX 3051
CLONTARF M.D.C.
QLD AUST 4019.

43
GRAEME MORRIS
P.O. BOX 371
CLEVELAND QLD 4163