
ISHUG

NEWS DIGEST

Focusing on the TI99/4A Home Computer

Volume 11, Number 11

December, 1992

Registered by Australia Post - Publication No. NBH5933



Sydney, New South Wales, Australia

\$3

December 1992

Index

All correspondence to:

P.O. Box 1089
Strawberry Hills, NSW 2012
Australia

The Board

Co-ordinator

Dick Warburton (02) 918 8132

Secretary

Terry Phillips (02) 797 6313

Treasurer

Geoff Trott (042) 29 6629

Directors

Rolf Schreiber (042) 85 5519

Russell Welham (043) 92 4000

Sub-committees

News Digest Editor

Bob Relyea (046) 57 1253

BBS Sysop

Ross Mudie (02) 456 2122

BBS telephone number (02) 456 4606

Merchandising

Percy Harrison (02) 808 3181

Publications Library

Russell Welham (043) 92 4000

Software library

Larry Saunders (02) 644 7377

Technical co-ordinator

Geoff Trott (042) 29 6629

Regional Group Contacts

Central Coast

Russell Welham (043) 92 4000

Coffs Harbour

Kevin Cox (066) 53 2649

Glebe

Mike Slattery (02) 692 8162

Hunter Valley

Geoff Phillips (049) 42 8176

Illawarra

Geoff Trott (042) 29 6629

Liverpool

Larry Saunders (02) 644 7377

Sutherland

Peter Young (02) 528 8775

Membership and Subscriptions

Annual Family Dues \$35.00

Associate membership \$10.00

Overseas Airmail Dues A\$65.00

Overseas Surface Mail Dues A\$50.00

TiSHUG Sydney Meeting

The December Meeting will start at 2.00 pm on 5th of December at Ryde Infant School, Tucker Street, Ryde. There will be no Assembler Class before the meeting. As it is the Annual General Meeting, please try to arrive well before 2.00pm so the business can start promptly.

Printed by

The University of Wollongong
Printery Services.

Title	Description	Author	Page No.
Alien attack	Software review	Harrison, Percy	2
Assembly class for Oct, Nov	Software hints	Mudie, Ross	19
Corrections to PRINT USING	Software hints	Trott, Geoff	17
Editor's comment	General interest	Relyea, Bob	1
Getting started #1	Software hints	Mudie, Ross	3
Look at memory	Software review	Green, R.A.	20
Modifying the PE box	Hardware hints	Takach, Ben	11
Modifying the TI99/4A	Hardware hints	Takach, Ben	9
My genealogy program	Software hints	Peterson, Jim	26
Myart	Software review	Shaw, Stephen	25
Newsletter update	General interest	Relyea, Bob	27
Regional group reports	General interest		27
Techo time	80 track disks	Trott, Geoff	17
TI-Base tutorial #20	Data base	Smoley, Martin	23
Tips from the Tigercub #65	Software hints	Peterson, Jim	7
Tips from the Tigercub #67	Software hints	Peterson, Jim	13
TiSHUG shop report	Club news	Harrison, Percy	2
To see or not to c	Software hints	Trott, Geoff	21
Trace subroutine	Software hints	Slattery, Mike	11
Treasurer's report	Club news	Trott, Geoff	8
Use of REF in assembly	Software hints	Mudie, Ross	19
Word processing part 1	Word processing	Christensen, Col	5
Writing in machine code	Software hints	Banfield, J.E.	18
Younger set	Greetings, adventure	Maker, Vincent	16

All articles appearing in this month's issue of the TND are available as text files on disk ready for the formatter. Newsletter editors please note that, if you wish to re-print any articles, contact us, stating which articles you are interested in and giving the date of the TND. These will be dispatched to you promptly at the cost of the media plus postage.

Wanted...
Members contributions to the BBS and the TiSHUG News Digest. Everyone can help, please do!

Editor's Comment

by Bob Relyea

I was glad to get back in the fold after my US holiday and be at the November meeting. I was especially glad to see that there were several members who volunteered to have a 'go' at producing the newsletter. That was really encouraging! What we now need to top it off are more local members to make contributions for the TND. The ones who regularly contribute do a great job but more is needed from different members. Let's see what you can do in 1993. O

TISHUG Shop with Percy Harrison

As I am writing this article well before the TI Faire, I am not able to comment on how successful it was but am sure that there will be a very good coverage in the next issue by some of our committee members.

I am anticipating that we will sell a reasonable amount of stocks at the Faire so this month I will not be including a list of what is available from the shop as this will depend on what is sold in the meantime.

At the November meeting the task of preparing the software disks for release each month was handed over to Larry Saunders, and as the details have to be in to our editor by the end of the week following the monthly meeting, Larry doubts that he can have a description of the disks that he plans to release done in time so these may not appear until the Jan/Feb issue. Larry has assured me, however, that he will release at least three disks at the December meeting. It may take Larry a little time to get into the full swing of things so I hope that you will bear with him for a month or two.

Also at the last meeting, I again raised the subject of altering the content of the magazine so as to cater for some of our younger and newer members by including programs that they can type into their computer and save on disk or cassette then run them and see how they work. Well, for opening my big mouth so often, I was saddled with the job of finding the first article for inclusion in this magazine. Because of the deadline for submitting articles to the Editor, plus the shop workload, I have had to extract an article from the "TEXAS PROGRAM BOOK" titled "ALIEN ATTACK" which I think children and beginners will enjoy. This program is written up elsewhere in this issue.

Lastly, but of extreme importance to the future of our club, I would like to write a little about a proposal that was discussed at our last meeting and which I am sure will be included in the Co-ordinator's report in this issue. That is, the proposal that we open the club up to users and/or owners of IBM compatible computers. This subject was raised some 18 months or more ago and at that time I, and many other members, were against the move as we still had a reasonable membership level and the club was running smoothly.

Today, however, that is not the case as we have a continual dropping-off of membership and currently only have about 129 financial members. I suspect that, as membership renewals fall due, we will see an even greater decline in numbers and if I am correct it will mean that we will no longer be able to sustain our high level of service and expertise that is currently available to our members with the result that more and more of our members will become disenchanted and leave the club, ultimately resulting in the club folding up. To prevent this from happening we need to maintain or increase our revenue and the easiest, simplest and least costly way of achieving this is to increase our membership which can only be done by opening it up to users of computers that will continue to thrive in our community, ie; IBM compatibles.

The main fear held by some of our members is that the IBM compatible element will completely take over our club and the dedicated TI owners will be a forgotten race. I do not believe that this will happen in the foreseeable future as quite a number of our existing members already own or use IBM compatible machines but still belong to the club and quite a few of them are the backbone of the club. If we lose these members the TI Club will die anyway but by opening up our club these members will remain with us and continue to support our TI-only members, financially, technically and with the repair and hardware services that we have and will continue to enjoy but only with the support of the extra revenue that they, the IBM compatible members, will bring into our coffers.

I would ask each of our members to give a little thought to this most important proposal bearing in mind that, should our finances become such that we can no longer afford to offer our members the service and expertise they are currently receiving, then they would have to look elsewhere for help. Already the availability of both software and hardware from other Australian sources has dried up and I doubt that you would be able to purchase or acquire any of the TI dedicated IC's or other components except from our club. Service by outside computer repairers for your TI would not be possible as they would not be able to get parts and even on the most simple of repairs that do not require dedicated components the cost would be more than most of you paid for the computer.

With the opening up of our club we would be able to maintain the standard of our magazine although part of it would be devoted to IBM compatible matters but, more importantly, we would be able to continue to offer you the services and computer exchange deal that you currently enjoy, ie; exchange computers for \$30.00.

In a nutshell, you need your club to keep you and your TI up and running and the club needs to increase its membership to be able to maintain the high standard of service that it currently offers its members.

It is hoped that a majority of our members will see and appreciate the benefits to be gained by supporting the proposal to open our club up to IBM Compatible users and take a few minutes of their time to contact the Co-ordinator, Dick Warburton, either by phone ((02) 918 8132) or letter (PO Box 1089 Strawberry Hills NSW 2012) to confirm their support as I see this move as being the only way in which we can extend the life of the TI community in this country.

Bye for now.

Alien Attack

with Percy Harrison

In order to help our younger members and/or beginners to develop familiarity with their TI 99/4A it was proposed that the magazine include articles which would allow the novice to type in a program save it to disk or cassette and then play and enjoy it. This is the first of such programs which was extracted from the "TEXAS PROGRAM BOOK".

Type the program exactly as it is printed here. If it does not run check that you have not made an error in your entries for each line.

```
100 REM TEXAS PROGRAM BOOK
110 REM
120 REM ALIEN ATTACK
130 REM
140 GOSUB 390
150 GOSUB 500
160 CALL GCHAR(LV,S,X)
170 IF (X=130)+(X=140)THEN 680
180 CALL HCHAR(LV,S,152)
190 FOR J=1 TO 8
200 IF HIT=1 THEN 360
210 CALL GCHAR(LV+J,S,X)
220 IF X<>130 THEN 300
230 SC=SC+10
240 NG=NG-1
250 HIT=1
260 CALL SOUND(300,-5,3)
270 CALL HCHAR(LV+J,S,141)
280 CALL HCHAR(LV+J,S,32)
290 GOTO 330
300 IF X<>140 THEN 330
310 SC=SC+25
320 GOTO 240
330 CALL VCHAR(LV+J,S,139)
340 CALL SOUND(50,-6,4)
350 CALL VCHAR(LV+J,S,32)
360 NEXT J
```

continued on page 14

Getting Started #1

by Ross Mudie

At the TISHUG meeting on 7th November 1992, it was reported that there are a number of new members who are having problems getting started with their computers. It was pointed out that there are no suitable beginners articles in the TISHUG News Digest and the question was asked who can help? Well here I go, I am prepared to have a go, I have not forgotten where I started off with my TI99/4A back in 1983.

The TI99/4A, despite its memory size and speed is a great machine on which to learn the basic principals of computing and a lot more. Its not going to cost a fortune with the TI99/4A, to make a start. Here is an early question, where do we start? Is it:

Programming?
Communicating via the telephone network?
Playing mind bending games?
Word Processing?
Data Base Management?
Process Control?
Helping with teaching of school subjects?

I have done all these things with my TI99/4A and the way that I started was programming in TI BASIC on an unexpanded TI99/4A console, saving and loading on a cassette tape recorder. My first program was a maths program for my kids. It presented a range of maths questions in the selected topics of addition, subtraction, multiplication or division. It gave 10 sums in a set and counted the number correct from the number of sums attempted. Here is a good task for someone wanting to start to learn to program.

I hope to write a series of articles on getting started with the TI99/4A, if there is some feedback that this is the sort of article that TISHUG's newer members want. In this article I will start off by introducing NUMERIC VARIABLES, PRINT, PENDING PRINT, IF-THEN, FOR-TO-NEXT, CALL CLEAR, CALL SCREEN, CALL COLOR and CALL KEY. The RESequence command is used and elementary program editing is introduced.

In either TI BASIC or TI Extended BASIC, the TI99/4A has many programming features which make the task of writing a simple maths program easy. If you have TI Extended BASIC available, it is preferable to use it since it is faster than TI BASIC and with more powerful commands, is easier to use. In either BASIC language, there are two types of variables. Variables are memory areas in the computer which can be used for temporary storage of information in either "STRING" or "NUMERIC" format. The contents of a NUMERIC variable must always be a number which can be directly used in a calculation. The contents of a STRING variable can be either words or numbers, or both. You cannot use numbers in STRING format in a calculation without first converting the numbers in the STRING into NUMERIC format. (More of this in another chapter).

When a variable is used, a way must be found to tell the program which variable you want to work on. In TI BASIC and TI Extended BASIC, variable names all commence with a letter and must be no more than 15 characters in length.

Examples of valid numeric variable names are A, COUNTER, SCORE etc. In fact, in your program the variable name can be used to help you to remember what you as the programmer used the variable for.

Try the following little program which uses PRINT and a NUMERIC VARIABLE called "A".

```
110 PRINT A
120 A=A+1
130 GOTO 110
```

To type the program in, press ENTER after each program line, placing the line number immediately following the

> prompt. Once you have typed in the program and you have the flashing cursor beside the > prompt in a blank line, type the word RUN and press ENTER. The program will commence at line 110 and print the initial value in the variable A (which is zero) and then scroll the whole screen up one line. The program will then step to line 120 where it will add 1 to the existing value stored in the numeric variable A. Line 130 instructs the program to go back to line 110. This program will just keep on going until the computer is turned off or its execution is interrupted. The program can be interrupted by holding the FCTN key on the keyboard and pressing the number 4. This is FCTN CLEAR which will cease program execution whilst retaining the program in the computer memory.

If you want to review the program that you have typed in, type LIST from the prompt whilst your program is not running. To EDIT (or change) the program when the program is not running, type the required line number then hold the FCTN key and press E (the UP arrow key) or X (the DOWN arrow key). When the program is not running it is in the COMMAND or IMMEDIATE mode. When you type the RUN command from the IMMEDIATE mode and the program executes, the computer is in the RUN mode. To make a change to a program line after typing the line number followed by FCTN E or FCTN X, use FCTN D or FCTN S to move the cursor to the required spot in the line to make your changes. FCTN 1 will delete the character under the cursor. To enter INSERT mode, hold the FCTN key and press number 2. As following keys are pressed the remainder of the line will move to the right and each new character will be inserted. When finished in a program line, you can go to the next line down by pressing FCTN X (DOWN arrow) or to the next line up by pressing FCTN E (UP arrow). To EXIT the EDIT MODE from any line just press the ENTER key.

Now that you have a quick overview of the way to edit a program, you may want to save your work on either cassette tape or floppy disk. If cassette tape is available, type SAVE CS1 (in upper case characters) and follow the instructions. (You must do this in the COMMAND mode). If a floppy disk drive is available, place the disk in drive 1 and type SAVE DSK1.TESTPROG1.

This instructs the computer to save the program on disk drive number 1 and to name the program TESTPROG1 on the disk in drive 1.

When working in Extended BASIC, a good shortcut to save typing is to place the SAVE command in a remark as follows:

```
100 ! SAVE DSK1.TESTPROG1
```

(The exclamation point "!" makes the rest of the line inactive in the program, it just becomes a "remark").

When you want to save the program, press 100 FCTN X, then press the ENTER key. Then press FCTN 8 which is REDO and delete the 100 !. This leaves the statement SAVE DSK1.TESTPROG1 in the command buffer and when you press ENTER the program is saved without a lot of typing.

So far the screen of the computer has remained rather cluttered when the program runs. If you want to clear the screen in COMMAND mode, just type CALL CLEAR and press ENTER.

To clear the screen as the program commences to run, add the following to the program:

```
105 CALL CLEAR
```

Note that when the program runs, it clears the screen before commencing to print the changing values of the NUMERIC VARIABLE A.

Each time you want to stop the program, so far, you have had to press FCTN CLEAR (FCTN 4). Edit line 130 to read as follows:

```
130 IF A<20 THEN 110
```


This means that whilst the value in variable A is less than 20, the program will loop back to line 110. If the value in variable A is not less than 20 (that is equal to or greater than 20), the program will run out of program lines to execute and it will end. In this case the use of END or STOP is not required, however if you want your program to cease at a point where it has not run out of program lines to execute, you can include either END or STOP in a program line, e.g.,

```
140 END
```

If you LIST the program, you will notice that the line numbers increment in tens, all except for line 105 which was added in the middle of lines 100 and 110. The reason for using lines every 10 numbers is to allow for ease of modification in the program. To change the line numbers to every ten, starting at 100, just type RES in COMMAND mode and the program line numbers will be "RESEQUENCED". Use LIST to see the change and SAVE your program to tape or disk. This is what your program should be now.

```
100 ! SAVE DSK1.TESTPROG1
110 CALL CLEAR
120 PRINT A
130 A=A+1
140 IF A<20 THEN 120
150 END
```

Note the last number printed on the screen. Can you figure out why it is 19?

Another method of building a control loop is FOR-NEXT. Try the following variation:

```
100 ! SAVE DSK1.TESTPROG2
110 CALL CLEAR
120 FOR A=1 TO 20
130 PRINT A
140 NEXT A
150 END
```

Note the last number printed on the screen. Can you figure out why it is 20 this time? Why does it differ from the results obtained from TESTPROG1?

Edit line 130 to include a semi-colon after A as follows:

```
130 PRINT A;
```

This creates a "pending print" situation, which does not scroll the screen after each print, but only when the screen line is full. Try a comma after A as shown:

```
130 PRINT A,
```

The pending print is now in the next "print zone". The print zones are 14 characters wide.

Add the following lines to your program:

```
102 CALL SCREEN(6)
104 FOR S=1 TO 14
106 CALL COLOR(S,16,1)
108 NEXT S
```

Line 102 will change the screen background colour to a dark blue whilst the program is running. The CALL COLOR will change all the Extended BASIC character colour sets to WHITE (colour 16) with a TRANSPARENT background to each character position. When the program finishes, the screen and character colours will revert to the default of black characters on a cyan (light blue) screen background.

Use RES and your program should now read as follows:

```
100 ! SAVE DSK1.TESTPROG2
110 CALL SCREEN(6)
120 FOR S=1 TO 14
130 CALL COLOR(S,16,1)
140 NEXT S
150 CALL CLEAR
160 FOR A=1 TO 20
```

```
170 PRINT A,
180 NEXT A
190 END
```

Change line 190 and add line 200 to make the program wait until you press the SPACE bar before it ends:

```
190 CALL KEY(5,K,S)
200 IF S<>32 THEN 190
210 END
```

Line 190 checks the keyboard for any pressed keys. If a pressed key is found then the numeric value assigned to that key will be returned from the Key Board Scan program in the variable K. The space bar is assigned the value of 32, so if no key is pressed, or any key other than the space bar is pressed, the IF-THEN statement in line 200 will direct the program back to line 190. (The symbol combination "<>" means "not equal to"). The program will then remain in the loop between 190 and 200 until the SPACE bar is pressed, or FCTN CLEAR is pressed or FCTN = (FCTN QUIT) is pressed or the power is turned off. If you want to save your program do not use FCTN QUIT or turn the power off until you have saved the program on tape or disk, FIRST!

If you want to find out more about the commands and statements used in this article, read the TI BASIC or TI Extended BASIC manuals. To learn about how each function of the programming language works, experiment with it. The best way to learn is to have a go, after all that is how I got started back in 1983.

Some new users with second hand computers may not have the plastic strip which TI provided to indicate the special functions which are available by holding the key designated FCTN and a number key (in the top row). The following descriptions when in programming mode for TI BASIC and TI Extended BASIC may help.

FCTN 1 - DELETE	Deletes character under the cursor.
FCTN 2 - INSERT	After pressing FCTN INSERT, each key press moves the characters from the cursor position to the right and inserts the pressed key.
FCTN 3 - ERASE	Erases the current line.
FCTN 4 - CLEAR	Ceases a LIST command or BREAKs a running program.
FCTN 5 - BEGIN	{Not used when writing or editing a
FCTN 6 - PROCEED	{BASIC or Extended BASIC program.
FCTN 7 - AID	{All 3 may be used in running program.
FCTN 8 - REDO	Good to copy last line. After ENTER, FCTN REDO gives the last line again, allowing line number to be changed.
FCTN 9 - BACK	See note for FCTN 5-6-7.
FCTN 0 -	Not used
FCTN = - QUIT	Exits back to master colour bar screen without saving program or data currently in use. If 32K memory is connected to computer, in TI Extended BASIC, can be disabled by using: CALL INIT :: CALL LOAD(-31806,16)

The Key Board Routines will "auto-repeat" after one second if keys are held depressed.

Reference Material: The following Texas Instruments books and programs are essential reading to learn the TI BASIC and TI Extended BASIC languages.

Beginners BASIC. (Cover is Blue/Black/White etc), LCB4180. Essential reading for beginners.

User's Reference Guide (Cover is Green/Yellow etc), This is the reference manual for TI BASIC.

TI Extended BASIC (Cover is Dark Blue/Purple/Green etc) This is the 224 page reference manual for Extended BASIC. Also 15 page "Product Information" and single sheet addendum.

continued on page 22

Word Processing #1

by Col Christensen, TIBUG Brisbane

This tutorial on word processing on the 99/4A is aimed at the newcomer as well as the not so new user. Word Processing has so many facets that it would be impossible to assimilate all the knowledge the first time through. For that reason I suggest that whether you are a beginner or an expert, you should reread this series or the TI-Writer manual from time to time. There is always something new, maybe a tip or maybe a whole procedure, that you can make use of. I do not claim to be an expert but through research into the manual and by testing out "new" and previously unused features, I feel now qualified to present this series. I have attempted to make it as simple but precise as possible in an effort to smooth the way on the newcomers bumpy road.

WHAT DO I NEED?

TI-Writer, the word processor published by Texas Instruments in 1982 came in the form of a sizeable hard covered manual, a solid state cartridge, one disk housing the Editor and Formatter programs and practice examples and a keyboard overlay. As a newcomer to word processing you should have, apart from the actual software, two important items. One is the keyboard OVERLAY to show the relevant keypresses for the top row of keys, and the other is this tutorial as a guide. The TI-Writer Manual is also useful but a little harder to locate a particular topic in it. Obviously you will already have an expanded system with disk drives and a printer connected to the system.

The word processor (WP) program you will be using will most likely not be the original TI one. Most everybody today uses the editor program supplied with and loaded through Tony McGovern's FUNNELWEB program. Tony has spent days and weeks and months on each upgrade to the original word processor program and there have been quite a number of upgraded versions. This software is FAIRWARE and users need to send off to Tony a reasonable payment if they have not already done so as some recompense for all his efforts. Let's face it, you probably will use your computer for word processing more than anything else. With FUNNELWEB you not only get a top notch word processor but a computer operating system as well. From here on I shall use the abbreviation WP for Tony's version of the word processor as the term TI-Writer is not now appropriate.

TEXT EDITOR

To start with, you need to know the two modes in which you use the WP, the Command mode and the Editor mode.

THE COMMAND MODE

When the WP first loads you see the cursor at the top left of the screen on what is screen line 2 with some command names shown on the line above. This is the COMMAND mode. On this line you type commands that allow you to do something with the document you have written. In the Command mode you have a selection of over a dozen one-letter or two-letter commands to process the text but more about these later. Pressing <E> for edit takes you from the Command mode to the Edit mode.

THE EDIT MODE

When in the EDIT mode, the top line of command prompts disappears and the whole 24 screen lines can be used for typing text. It's in this mode that you will spend most of your WP time. Pressing FCTN/9 takes you from the Edit mode back to the Command mode. Now let's try that again on the keyboard. From the Command mode press <E> to go to the Edit mode and from the Edit mode press FCTN/9 to escape to the Command mode. In the Edit mode there are quite a number of keypress combinations that facilitate different operations but more later on these, too.

SCREEN WINDOWS

One small problem with most computers using TVs as monitors, and the 4A is no exception, is the limitation on the width of the screen when compared with normal paper width. Your printer under most conditions will be required to print 80 characters on each line. But the WP screen is only 40 characters wide so a system of windowing is used. We can have a line 80 characters long but we cannot see all of it at once. Three screen windows are used and they show columns 0 to 39 (left half), 20 to 59 (middle half) and 40 to 79 (right half) respectively. The program starts up with the left window showing on the screen, but further presses of the FCTN/5 key bring up the next window in the sequence left, middle, right, left, middle, right, etc.

WORD WRAP

Word wrap mode can be turned off or on using the CTRL/O (that is a zero) key combination. By CTRL/O I mean that you hold down the CTRL key while you tap the O key once. Word wrap is ON when the WP first begins and an indication of this mode is the appearance of the cursor, a SOLID rectangle. With word wrap ON, typing continues across the screen to the right tab position then automatically restarts on the next line at the left tab position. If at the end of a line only part of a word will fit, the whole word is automatically moved to the beginning of the next line. Another effect also is when typing continues to and past the right edge of the screen the screen flips to display the next window.

With word wrap OFF the cursor appears as a HOLLOW rectangle. In this mode typing continues across the screen to the right tab position only and will not drop to the start of the next line unless <ENTER> is pressed.

SETTING THE TABS

Tab settings control the way your document appears on the screen. As well as the standard tab position presets, you can also preset the left margin tab, the Right margin tab and the paragraph Indentation tab. The Indent tab sets the position or column on the screen that a new paragraph begins at. The Right tab sets the screen column past which typing is not allowed and the Left tab defines the column where each line of text begins at near the left of the screen.

To set the tabs go to the Command mode and press <T> for Tabs. You will see above the cursor a numberline that starts at an invisible 0 and extends to the number 79 (that you would see if you window across to the right window). On this line certain letters are placed to indicate the tab settings required. "L" is for left margin, "R" is for right margin, "I" is for paragraph indenting and "T" is for tab.

Under the Tab line is the cursor on top of the letter "L". That seems a suitable cursor for the left margin so leave the "L" there and move the cursor over to position number 5. Type an "I" over the "T" here to indicate the position for paragraph indentation. Suppose, for example, in typing a document you do not want to have the WP flipping from one window to the next all the time. You can overcome this with a prudent setting of the right margin. Put an "R" for right at position 39. Then all typing will take place on the left screen window.

Tab position presets are fixed by using the letter "T" where required on the tab ruler. You can blank out any unwanted "T"s still showing between the "L" and the "R" settings. If (FCTN/7) is pressed when typing, the cursor will jump to the next "T" setting on that screen line. Use this facility when typing lists or tables in vertical columns. If two or more "L" settings are entered, the rightmost of them only is accepted and, in similar vein, of multiple "R" settings, the leftmost one is accepted.

Whenever you save a document to disk, the tab settings will be saved with it. Conversely, whenever

you load a saved document from disk the tab settings will be retrieved with it and will be in effect when the loaded text appears on the screen.

After setting tabs, you must hit <ENTER> to confirm the settings and you will be returned to the Edit mode with the cursor at the point where it was just before escaping to the Command mode.

SCREEN COLOUR SELECTION

The default screen colours are white characters on a blue background but you can toggle through a selection of 10 colour combinations by successive presses of CTRL/3.

TRYING IT OUT

Now let's see what we can do with what we have seen so far. I hope you are making a list of the special keypresses mentioned to keep for future reference. Not all those mentioned will be found on the overlay.

Starting from first loading the WP, the cursor appears in the Command mode on screen line 2. First set the tabs as above or to your liking. Remember to press <T> to access the tab numberline. When done press <ENTER>. What mode are we now in? Yes, the Edit mode where we can type to our hearts content. But what is this? You have probably set a paragraph indent tab somewhere in from the left margin and the cursor is at the left margin setting on the screen. Well, the computer is not dumb. You have not told it that you are to begin a new paragraph. So press <ENTER>, the signal in the Edit mode to end a paragraph, and there is the cursor at the correct indent position on the second line.

Notice the line numbers down the left side of the screen. Really the only time you need to see these is when copying, moving or deleting parts of your text. Besides, with the line numbers showing you will not see the whole 40 columns of text. Pressing FCTN/0 will toggle the line numbers ON or OFF. Repeat pressing FCTN/0 to see the effect.

Now is the time to type something, anything that comes into your head but first release the alpha lock key for lower case characters and use the shift key for capitals. Just keep typing on and on without worrying about mistakes and do not forget to throw in an occasional comma or full stop from time to time. Notice the "bell" sound as you get near to the right margin setting. Look to see how the word wrap functions. When you have exhausted the phase of the topic you are writing, press <ENTER> twice. The second press of the enter key serves the purpose of leaving one line spacing between paragraphs. Notice the funny character that comes on the screen where the enter key is pressed. This is a signal to a printer to do a carriage return and a line feed at that point. The WP will not allow you to type over a C_r symbol in word wrap mode, the symbol just gets pushed ahead of the cursor. However, the C_r can be deleted using the FCTN/1 delete char combination when necessary. Type another paragraph or so in similar vein expanding on your topic of discussion. Now it is time to correct any errors and make changes.

ERROR CORRECTION AND TEXT MANIPULATION

DELETECHAR (FCTN/1) To remove individual characters, move the cursor to the required point and press FCTN/1 for each character to be deleted or hold the keys down for repeated deletions.

DELETELINE (FCTN/3) The whole line from left margin to right margin will be deleted with FCTN/3.

INSERT CHAR (FCTN/2) Place the cursor at the appropriate position for an insertion and press FCTN/2. Notice that any text to the right of the cursor drops to the line below leaving space after the cursor. Type what has to be added, whether one letter, one word or a number of

lines of text. If necessary you can insert again in a part that has already been inserted. The screen will probably look untidy with bits here and there on different lines. The text then needs to be reformatted as explained shortly.

In the non word wrap mode the insert char puts a character under the cursor and moves the remainder of the line to the right of the cursor one position to the right. Any character already at the right margin will be lost.

INSERT LINE (FCTN/8) Pressing FCTN/8 will move all text on the screen and in memory that is on the cursor line and below it down one line. It then blanks the line that the cursor is on.

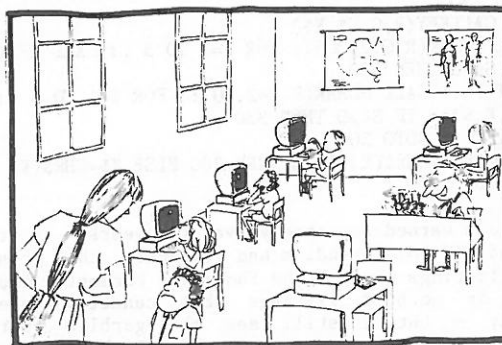
REFORMATTING (CTRL/2) With word wrap on, move the cursor to a point from which you need to tidy up the text and press CTRL/2 and, Bingo! all the text just pops back into place where it should be. Reformatting takes place between the left and right margins as set on the tab numberline and continues down the file in memory until a C_r symbol is reached, i.e. till the end of the paragraph. Try altering the L and R tab settings and reformatting your document paragraph by paragraph. Place the cursor over the first character in the first paragraph and press FCTN/2 (insert char) then CTRL/2 (reformat). Press CTRL/4 to get to the start of the next paragraph and reformat it the same way. Play around with inserting, deleting and reformatting until you are familiar with their intricacies.

OOPS (CTRL/1) This aptly named function can undo deletions and insertions only if no other keys have been pressed since the deletion or insertion. It can only go back to the point before the wrong keypress or series of similar keypresses and take up from there. If, for example, you wanted to insert char (FCTN/2) but pressed the FCTN/3 (erase line) key by mistake. Too late, the line disappears. Not to worry. Press CTRL/1 before touching any other key. Presto! the missing line reappears. The main thing to remember is that if you make a booboo, press OOPS anyway. You never know, your error might be recovered.

CASE CONVERSION (CTRL/> and CTRL/:) If you are like me and watch the keys as you type, you will often look up and find the wrong sized characters on the screen, usually upper case because you forgot to reset the alpha lock key. Tony McGovern has added this little beauty to the word processor. Upper case will be changed to lower case as the cursor moves over them with CTRL/> held down. Is not that terrific? I use it regularly. The opposite effect, lower to upper case conversion, is obtained by using the CTRL/: keys.

CONCLUDING PART 1

So there you are. Try out everything mentioned this month and also work on the cursor movement keys FCTN/S, E, D or X or use the CTRL access key if it is more comfortable that way. Do not forget to keep a list of all those special magic keypresses for ready reference or begin making your own overlay for the top row of keys. Next month commands for Line and File Handling will be presented.



"I'm afraid you'll just have to wait until it's your turn to play with the pencil..."

Tips from the Tigercub #65

by Jim Peterson, Tigercub Software, USA

It is a bit of a nuisance to have to hit Enter after inputting a single character such as Y or N for "yes" or "no". CALL KEY accepts a single character without Enter, but has no blinking cursor to tell you that it is waiting. I should have had this one in my Nuts & Bolts years ago - the CALL KEY WITH CURSOR subprogram! R is the row, C is the TAB position, V\$ is the validation string, such as "YyNn", and the character selected is returned in K\$.

```
30000 SUB CALLKEY(R,C,V$,K$)
30001 CALL HCHAR(R,C+2,30):: FOR T=1 TO 3 :: CALL KEY(O,
K,S):: IF S<>O THEN 30004
30002 NEXT T :: CALL HCHAR(R,C+2,20):: FOR T=1 TO 3 :: C
ALL KEY(O,K,S):: IF S<>O THEN 30004
30003 NEXT T :: GOTO 30001
30004 IF POS(V$,CHR$(K),1)=0 THEN 30001 ELSE K$=CHR$(K)
30005 SUBEND
```

And for a demonstration of the use of that subprogram, here is a little game that no one will ever play to the end -

```
100 DISPLAY AT(3,6)ERASE ALL:"THE ULTIMATE TEST": "" : " An
swer the question with a number according to whether the
number or color shown,"
110 DISPLAY AT(8,1):"or the note sounded, was 1st or 2nd
or 3rd, etc.,"
120 DISPLAY AT(23,6):"PRESS ANY KEY" :: DISPLAY AT(23,6)
:"press any key" :: CALL KEY(O,K,SS):: IF SS=0 THEN 120
ELSE CALL CLEAR
130 DATA 2,BLACK,3,GREEN,5,BLUE,9,RED,12,YELLOW,14,PURPL
E
140 FOR J=1 TO 6 :: READ C(J),C$(J):: CT$=CT$&CHR$(J)::
W$=W$&CHR$(J+48):: NEXT J :: T=2 :: DL=500 :: V$="12"
150 RANDOMIZE :: T$,NN$=CT$:: FOR J=1 TO T :: X=INT(RND
*LEN(T$)+1):: X$=SEG$(T$,X,1):: T$=SEG$(T$,1,X-1)&SEG$(T
$,X+1,255):: Y(J)=ASC(X$)
160 X=INT(RND*LEN(NN$)+1):: X$=SEG$(NN$,X,1):: NN$=SEG$(
NN$,1,X-1)&SEG$(NN$,X+1,255) :: S(J)=ASC(X$):: NEXT J ::
FOR J=1 TO T
170 Z(J)=INT(89*TRND+10):: FOR K=1 TO J-1 :: IF Z(J)=Z(K)
THEN 170
180 NEXT K :: NEXT J :: CALL CLEAR :: CALL COLOR(3,16,1,
4,16,1)
190 FOR J=1 TO T :: CALL SCREEN(C(Y(J))): CALL SOUND(-9
9,110*S(J),0):: DISPLAY AT(12,12):Z(J):: FOR D=1 TO DL
:: NEXT D :: NEXT J
200 CALL CLEAR :: CALL SCREEN(16):: CALL COLOR(3,2,1,4,2
,1): X=INT(3*TRND+1):: W=INT(T*TRND+1):: ON X GOTO 210,23
0,210
210 IF X=1 THEN Q$=C$(Y(W))ELSE IF X=3 THEN Q$=STR$(Z(W)
)
220 DISPLAY AT(12,1):"WHICH WAS ";Q$ :: GOTO 240
230 CALL SOUND(1,30000,30):: DISPLAY AT(12,1):"WHICH WAS
?" :: FOR D=1 TO 200 :: NEXT D :: CALL SOUND(500,110*S(W)
),0)
240 CALL CALLKEY(12,20,V$,K$):: Q=ASC(K$)-48
250 IF Q=W THEN DISPLAY AT(15,12):"RIGHT!" ELSE DISPLAY
AT(15,12):"WRONG!"
260 IF Q=W THEN DL=DL-50 ELSE DL=DL+50
270 IF DL<100 THEN DL=500 :: T=T+1 :: V$=SEG$(W$,1,T)
280 GOTO 150
290 SUB CALLKEY(R,C,V$,K$)
300 CALL HCHAR(R,C+2,30):: FOR T=1 TO 3 :: CALL KEY(O,K,
S):: IF S<>O THEN 330
310 NEXT T :: CALL HCHAR(R,C+2,20):: FOR T=1 TO 3 :: CAL
L KEY(O,K,S):: IF S<>O THEN 330
320 NEXT T :: GOTO 300
330 IF POS(V$,CHR$(K),1)=0 THEN 300 ELSE K$=CHR$(K)
340 SUBEND
```

I have warned repeatedly over the years, in these Tips and in Micropendium and elsewhere, that printing program listings through the Funnelweb Formatter usually results in garbled listings that cannot be keyed in correctly - but I still see the garbled listings published. Here is a fix to the Funnelweb FO file that will partially solve the problem -

Boot DSKU. Select 1. File Utilities. Select 5. Find String. Enter filename FO and the drive number. Enter H for hex. Enter the string 2A23214026. Enter replace string 7C2321605C. When the string is found, enter R for replace, then CTRL W, hit Enter twice to accept the defaults. Thereafter, use FCTN Z instead of & to under line, FCTN C instead of @ to double-strike, and FCTN A instead of * to call a value added file. I do not know why Texas Instruments did not do that in the first place, and I wonder why the McGovern did not make that fix.

Now, can anyone tell me how to replace the ^, which tends to disappear, and the period, which will make the whole line disappear if it happens to be at the beginning of the line?

If you are one of the few who are still interested in recreational computing - the use of the computer to solve puzzles and math problems just for the fun of it - you might be interested in Recreational and Educational Computing, published 8 times a year at 909 Violet Terrace Clarks Summit PA 18411. The annual subscription is \$27. Program listings are in dialects of Basic other than TI but usually not hard to convert.

That is where I found this ridiculously short, simple and fast card shuffling routine.

```
100 DIM C(52)
110 FOR X=1 TO 52 :: C(X)=X :: NEXT X
120 FOR X=52 TO 1 STEP -1 :: I=INT(RND*X+1)
130 T=C(I):: C(I)=C(X):: C(X)=T :: NEXT X
```

In the same place, I read a routine to extract a root to 16-digit accuracy instead of the 8 digits available on a PC from the basic formula $ROOT=NUMBER^{(1/POWER)}$. We do not need it - our obsolete 16k 16-bit computer can give us 14-digit accuracy from the basic formula!

The same publication gave me the idea for this little game -

```
100 DISPLAY AT(3,6)ERASE ALL:"THE GAME OF N": "" : "You and
the computer will take turns adding to a number to
reach a goal."
110 DISPLAY AT(8,1):"If you reach the goal, you win. You
get to go first and you should be able to win almost
every time."
120 RANDOMIZE :: N=INT(RND15)+15 :: R=INT(4*TRND+3):: S=
R+1 :: D=N-INT(N/S)*S :: T=0
130 DISPLAY AT(13,1):"The goal is";N: "" : "Maximum input i
s";R :: DISPLAY AT(19,1):RPT$(" ",28*6)
140 DISPLAY AT(17,1):"Your number?" :: ACCEPT AT(17,14)S
IZE(1)VALIDATE(DIGIT):A :: IF A<1 OR A>R THEN DISPLAY AT
(15,1): "" :: GOTO 130
150 T=T+A :: DISPLAY AT(21,1):"Total is";T :: IF T=N THE
N DISPLAY AT(23,1):"YOU WIN!" :: GOSUB 190 :: GOTO 120
160 IF N-T<S THEN P=N-T :: T=T+P :: DISPLAY AT(19,1):"Co
mputer adds";P :: DISPLAY AT(21,1):"Total is";T :: DISPL
AY AT(23,1):"COMPUTER WINS!" :: GOSUB 190 :: GOTO 120
170 IF T=0 THEN P=D ELSE IF (N-T)/S=INT((N-T)/S) THEN P=I
NT(R*TRND+1) ELSE Y=N-T :: P=Y-INT(Y/S)*S
180 T=T+P :: DISPLAY AT(19,1):"Computer adds";P :: DISPL
AY AT(21,1):"Total is";T :: GOTO 140
190 DISPLAY AT(24,8):"PRESS ANY KEY" :: DISPLAY AT(24,8)
:"press any key" :: CALL KEY(O,K,S):: IF S=0 THEN 190 EL
SE T=0 :: RETURN
```

REC also printed a puzzle which seemed so simple that I could not see why. It goes like this -

A game show host shows you three curtains. Behind one is a new car, behind the other two are goats. You choose one. The host, who can peek behind the curtain, opens one of those you did not pick, and shows a goat. Then he offers to let you change your choice. Should you switch, stand pat, or does it make no difference?

You now have a 50-50 bet, so it makes no difference, right? But some very distinguished mathematicians were saying you should switch, so I wrote

this computer simulation to prove them wrong. Key it in, run it, and be surprised. Do figures lie? Do computers lie? Is there something wrong with my simulation?

```
100 CALL CLEAR
110 DATA CAR BEHIND,A PICKS,HOST SHOWS,A WINS,B WINS,C W
INS
120 FOR J=1 TO 3 :: READ M$ :: DISPLAY AT(J,1):M$ :: NEX
T J :: FOR J=12 TO 14 :: READ M$ :: DISPLAY AT(J,1):M$ :
: NEXT J
130 FOR J=1 TO 1000 :: RANDOMIZE :: X=INT(3*RND+1):: DIS
PLAY AT(1,13):X !RANDOMLY PLACE CAR
140 A=INT(3*RND+1):: DISPLAY AT(2,13):A !PLAYER CHOOSES
150 D=INT(3*RND+1):: IF D=X OR D=A THEN 150 :: DISPLAY A
T(3,13):D :: ! HOST PICKS CURTAIN WITH GOAT
160 IF A=X THEN AA=AA+1 :: DISPLAY AT(12,7):AA ! A DOES
NOT SWITCH
170 B=INT(3*RND+1):: IF B=A OR B=D THEN 170
180 IF B=X THEN BB=BB+1 :: DISPLAY AT(13,7):BB ! B SWITC
HES
190 C=INT(3*RND+1):: IF C=D THEN 190
200 IF C=X THEN CC=CC+1 :: DISPLAY AT(14,6):CC ! C CHOO
ES RANDOMLY
210 NEXT J
```

Here is an improved version of a program that was in a Tips long ago, to strip out the extra blanks from a Filled and Adjusted Funnelweb Formatter file -

```
100 DISPLAY AT(3,6)ERASE ALL:"TIGERCUB UNFILLER": "" : To
remove extra spaces from:" a TI-Writer text which has":
"been Filled and Adjusted by"
110 DISPLAY AT(8,1):"the Formatter, prior to":
reformatting"
120 DISPLAY AT(15,1):"Input file? DSK" :: ACCEPT AT(15,1
6):IF$ :: OPEN #1:"DSK"&IF$,INPUT
130 DISPLAY AT(17,1):"Output file? DSK" :: ACCEPT AT(17,
17):OF$ :: OPEN #2:"DSK"&OF$
140 LINPUT #1:M$ :: P=1
150 X=POS(M$, " ",P):: IF X=P THEN P=P+1 :: GOTO 150
160 X=POS(M$, " ",P):: IF X=0 THEN PRINT #2:M$ :: GOTO 1
80
170 M$=SEG$(M$,1,X)&SEG$(M$,X+2,255):: GOTO 160
180 IF EOF(1)<>1 THEN 140 :: CLOSE #1 :: CLOSE #2
```

While a program is running, the computer periodically pauses for a fraction of a second to do a "garbage collection", getting rid of information it no longer needs, to make room in memory. If this pause occurs at a critical moment in program execution, it can cause problems. Thanks to the Sydney User Group in Australia, here is a CALL LOAD which will force a garbage collection just before that critical point - CALL LOAD(-31885,144,"",-31858,81,169,152,0)

Here is a neat one from Bruce Harrison. Key it in, (you can skip the lines that start with an asterisk) and assemble it, then use ALSAVE to imbed it in any program that opens a disk file. Put CALL LINK("DEVICE",DEV\$) at the beginning of the program and change any line reading OPEN #1:"DSK1.FILENAME" - or whatever - to read - OPEN #1:DEV\$&" .FILENAME" (do not forget the period before the filename!). Now you can load the program from any drive and it will open the file on that same drive!

```
* STRING ASSIGN DEVICE NAME
* PLACES DEVICE NAME IN AN
*
```

```
Extended Basic
STRING
* HARRISON SOFTWARE
* 8 OCTOBER 1990
* FOR USE WITH ALSAVE AND
Extended Basic
```

```
* TAKES ONLY 42 BYTES MEMORY
STRASG EQU >2010
WS EQU >20BA
DEF DEVICE
DEVICE
* USE OUR WORKSPACE
LWPI WS
* GET THE CRU BASE IN R12
MOV @>83D0,R12
```

```
* GET ROM ADDRESS FOR DEVICE
* IN R2
MOV @>83D2,R2
* ENABLE THE ROM
LDCR @ONES,0
* ADDING 4 PUTS US AT THE
* LENGTH BYTE
AI R2,4
* FIRST PARAMETER
LI R1,1
* NOT AN ARRAY VARIABLE
CLR RO
* ASSIGN DEVICE NAME TO A
* STRING
BLWP @STRASG
* CLEAR CRU, DISABLE ROM
LDCR RO,0
* LOAD GPL WORKSPACE
LWPI >83E0
* RETURN TO GPL INTERPRETER
B @>006A
* WORD TO TURN ON ROM IN CRU
ONES DATA >0101
END
```

Getting short on memory, so more next time.

Jim Peterson

Treasurer's Report

by Geoff Trott

This will be my last Treasurer's report as I will not be nominating for the office of Director at the AGM. I have enjoyed my stint in office but I have found it harder, as time goes by, to find the time to do the job properly. I hope that by not being involved in the running of the club, I will be able to spend more time on hardware and software problems. I will not be giving up on the TI99/4A for the foreseeable future and am definitely not interested in spending my leisure time on PCs. I want to enjoy myself! I hope you all have a Merry Christmas and a Happy New Year.

Income for October	\$1470.80
Payments for October	\$1995.13
Excess of expenses over income for October	\$524.33

continued from page 24

the same time so the FIND is quite fast because you are pointing to a smaller segment of the total series with the IF section, and selecting a slot from there. With the FIND/EDIT in VDP memory you save disk access time. This would be much more noticeable with \RES as used in SUBNUM2. \RES contains 15 lines of code where \ED only has 2. It may help you to reread the MAY issue on the new INSTALL feature. I will try to rehash this stuff next month and continue the ideas on handling multiple databases instead of one large database. Good luck for now. Marty.

```
* 05/06/90 ISSCRN/C
SET TALK OFF
CLEAR
SET HEADING OFF
SET RECNUM OFF
WRITE 6,8,"Enter the right hand digits"
SET INVERSE ON
WRITE 2,6," ";
" ";
WRITE 3,6," 74LS Series Integrated ";
"Circuits "
WRITE 4,6," ";
" ";
WRITE 8,16," "
WRITE 9,16," EXAMPLE "
WRITE 10,16," "
SET INVERSE OFF
WRITE 12,6,"Manufacturer You"
WRITE 13,6,"Part Number Enter"
WRITE 15,12,"74LS221 = >221 < ENTER"
WRITE 17,12,"74LS01 = >01 < -1 "
WRITE 19,34,"TO QUIT"
RETURN Copyright Martin A. Smoley 1990
```

Modifying the TI99/4A

by Ben von Takach

The most annoying feature of the 99/4A is the expansion socket at the side of the console. One does not have to be rough or careless to dislodge the plugged in peripheral or the heavy P.E. Box plug with its integral interface card.

This design concept is the cause of the very inflexible keyboard arrangement. Simply, the individual positioning of the keyboard and especially the tilt facility is inhibited. The compact keyboard with its 47 keys is another limitation. Many frequently used key strokes require the simultaneous push of 2 keys. The operation of arrow keys is especially cumbersome.

The very first modification I undertook on the console was the installation of the Rave 99 keyboard interface soon after it became available. Many years later I still regard it to be the most significant add-on, which turned my 994A into an efficient and comfortable machine. I can move the AT style keyboard freely on the desk without any risk of dislodging the P.E. box plug, further, every function is a single key push, separate and dedicated arrow keys and an ever-ready numeric key pad are some of the advantages. The keyboard can be tilted at will and it supports keyboard controlled reset and load interrupt functions.

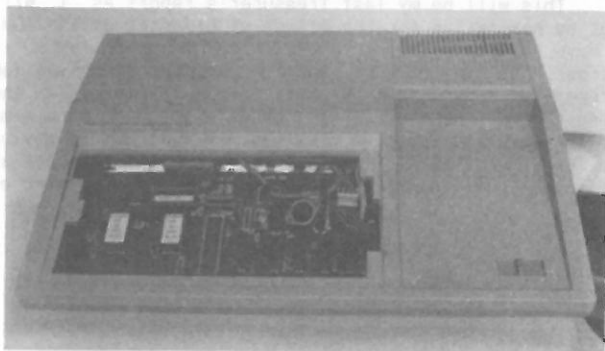


Photo 1: The keyboard interface takes the place of the original keyboard. The IBM XT keyboard plugs into the DIN socket.

The original version was released complete with a 105 key custom keyboard featuring 24 function keys. This enables the single key operation of every TI-Writer command or every multiplan command in conjunction with the multiplan module.

The custom keyboard version is no longer available. The current version uses a 12 function 101 key XT compatible keyboard. The 12 function key limits the number of single key push command options in TI-writer mode respectively in multiplan mode, however, all other above mentioned enhancements are supported. One may purchase the keyboard interface in a kit form without keyboard for US\$92. The total cost will be higher by the time postage is added, and at current exchange rates the all-up price in Australian currency will be about \$150.

The kit does not include some of the common ic-s, nor the resistors and capacitors. The missing items will cost about \$15. The interface takes the place of the TI keyboard in the console, and the keyboard aperture of the moulded cover has to be patched up with a suitable piece of plastic or metal cover. In any case it does not improve the appearance of the console.

Purchasing the interface in a kit form without a keyboard will save dollars, but it leaves one vulnerable to the whim of Murphy. IBM XT compatible keyboards are easy to get, however in this instance the degree of

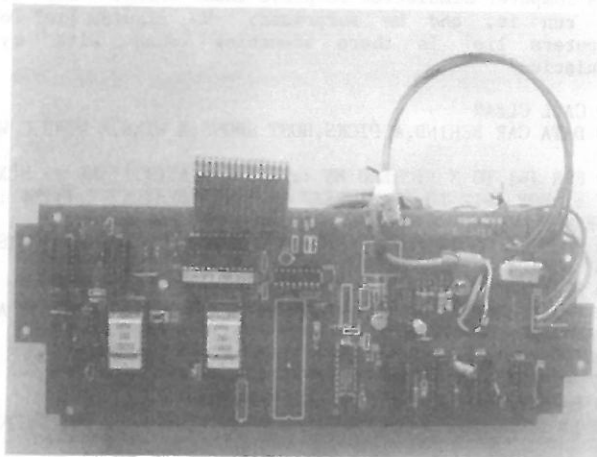


Photo 2: The Rave Keyboard interface kit assembled and ready to be fitted into the console.

compatibility is of major importance. A recently completed project has refused to work with two out of four keyboards and the alpha lock (now called the cap-lock) key would only function with one of the remaining 2. This could be a disaster if you only have access to one keyboard! Nevertheless, it is well worth the effort, it is magic when completed.

The second modification was the upgrade with a Mechatronic 80 column card. The project is very worthwhile for multiplan and word processing, however it is hardly needed for anything else, unless you are a graphics fanatic. Any Basic or Extended Basic programs will have to be virtually re-written to take advantage of the 80 column screen. Further, these programs will not run on a conventional 40 column machine.

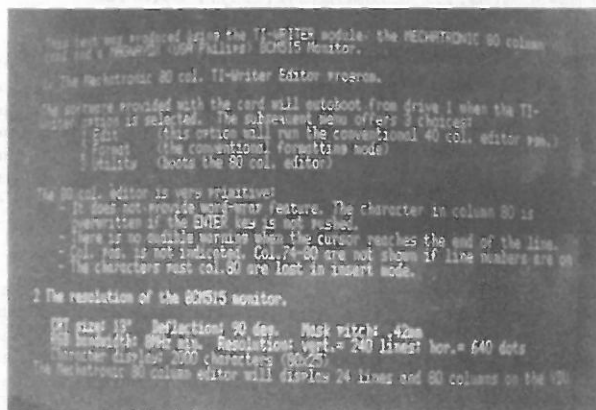


Photo 3: 80 column text on the screen. The characters are very close to each other.

The resolution of the 80 column screen has to be very good, otherwise the text is fuzzy and the characters tend to merge into one solid block. The average analogue RGB monitor will show a little better than CGA quality display. In this day and age we are spoiled by very high resolution resolution displays. The CGA screen is a bit of a let-down. The modification is relatively simple. The video processor chip is removed and a 40 way ribbon cable fitted with a suitable plug is inserted into the vacated socket. The new video interface card is plugged into the side of the console. The other end of the ribbon cable is also plugged in a socket on the interface board.

The card is not drawing power from the console, it has its own plugback power supply.

This brings me to the third modification. The console power supply is just able to provide the essential currents to the console and it requires an external transformer. With the 80 column card added, I needed a second transformer. Together with the monitor and P.E. Box I needed 4 power outlets and wires running everywhere. The answer is one single power supply. Before one can proceed with the design of a power supply, the currents of the +5V, -5V and +12V rails have to be measured. This information is not published in any of the TI service manuals.

The console fitted with the RAVE keyboard interface, the 80 column card and the Extended Basic cartridge plugged in requires:

+5V at 1.25A
 +12V at 75mA
 -5V at 80mA

The mechatronic 80 column card needs +5V at 0.425A.

The current measurements show that the console power supply is hardly able to supply any additional power to peripherals plugged into the expansion socket. On the other hand even the most modest switch mode power supply designed to power a PC would give more than enough power.

Now, faced with the unsightly console sporting a makeshift keyboard cover, the need for a decent case for the new power supply, and a permanent solution for the plug-in 80 column card, obviously a new case was needed accommodate the lot.

Logically, the base dimensions of the new case should be that of the P.E. Box. The height had to be 120mm to fit in everything comfortably. The new case was large enough to accommodate a 3 1/2" 20MB hard disk drive as well. Accommodating the individual components has presented some problems. Each unit had to be accessible for servicing and trouble shooting. The relative position of the expansion socket had to be maintained for obvious reasons. The cartridge port had to face the operator. Eventually all the problems were solved and the new look 99/4A was fired up.

The P.E. Box sits on top of the console case, thus the space requirement remained the same as before. The keyboard is as flexible as any IBM or its clones and the system has worked fine since completed some six years ago. The console case has not been opened since its completion.



Photo 4: The complete, modified system.

The cartridge port socket has gold plated contacts and I had no cartridge related lockups for years (the occasional lockup is invariably due to insufficient seating of one of the cards in the P.E. Box, or to a power glitch).

The P.E. Box interface plug is fitted with a jack screw and it is firmly anchored to the console case (similar to the conventional D plug fixtures).

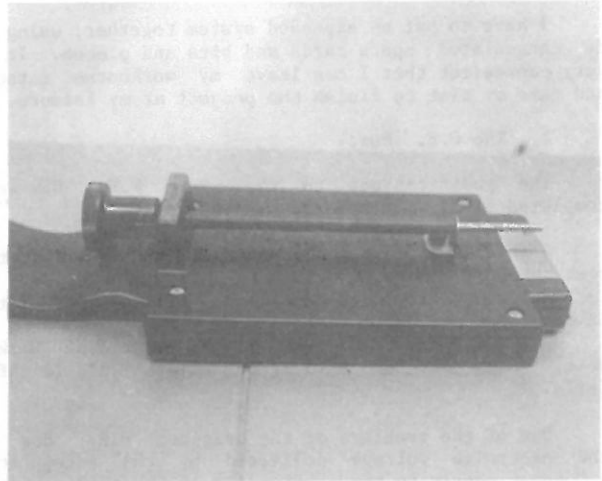


Photo 5: The curse of the dislodged plug fixed for good. A short piece of welding rod, one knob and an angle bracket is all one needs.

One drawback of any one-off design is the absence of hindsight. Any inherent physical shortcoming is permanent! Would I do it differently next time around? Of course I would!

I have not included the master switching feature, thus it lives in a separate enclosure - albeit out of sight - under the desk top. It consists of two interlocked electronic timers. The unit provides instantaneous, on delay and off delay 240V power. The monitor and the printers are supplied from the instantaneous supply. The console is powered from the on delay line and the P.E. Box from the off delay line. Not including this feature in the console case made it necessary to provide a separate supply to the hard drive from the instant source, otherwise the presence of the hard drive is not recognised. It would also be neater to have the 240V sockets for the monitor, P.E. Box and printer incorporated in the console case to avoid the need of multiple outlets.

The console case has no fan. I do switch off the hard drive during lengthy sessions as it generates quite a lot of heat. Although it has given me no trouble so far, I do not feel comfortable letting it run continuously without cooling.

It was a lot of work, you may ask, was it worth it? Well I think it was. I have a system which is as comfortable as any clone, it does everything an XT would and I do know it inside and out. It is certainly more reliable than an el-cheapo Asian import, and if it fails it can be fixed at less cost than any other brand I know. It has survived so far 10 years in the office working at least 2 hours every day without any breakdown.

My TI is used for word processing, bookkeeping, invoicing*, die design*, quotations *, preparation of import documentation*, sending and receiving telexes, sending of faxes, burning eeproms, decoding PLC program codes*, cost calculations*, thermal analysis*, transformer design*, and running the many other leisure programs (tasks marked by an * through own programs). Many TI files which have to be supplied on floppys to third parties are converted to IBM format. It is often more economical to do it on the TI than on a PC.

The pictures illustrate the extent of the modifications and the layout of the workstation. O

Modifying the PE Box

by Ben von Takach

I have to put an expanded system together, using up my accumulated spare cards and bits and pieces. It is very convenient that I can leave my workhorse intact, and take my time to finish the project at my leisure.

1. The P.E. Box.

The modifications to the spare P.E. Box were completed first, for several reasons:

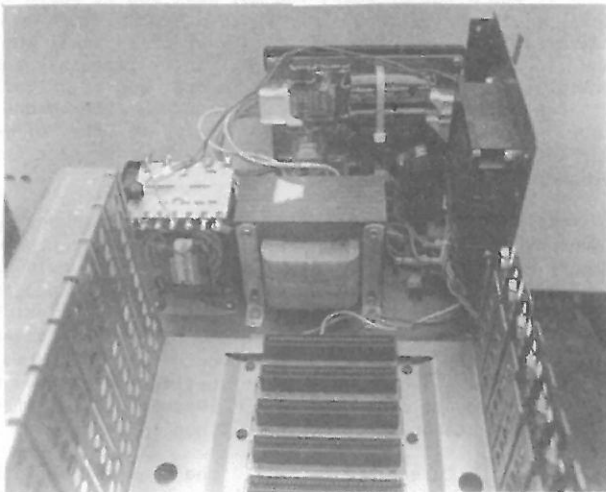
- This is an essential component for testing the rest, thus it has to be operational first.
- The modifications are simple, it can be completed in less than one day.
- The modified P.E. Box is a functional backup unit, which may be used if the original old faithful fails.

One of the problems of the original P.E. Box is the excessive voltage delivered by the unregulated D.C. power supply to the power rails of the card sockets. The nominal +8, +16, -16 volt supply is +11.5, +22, and -24volts respectively (see article by the writer in the March '89 issue of the TND, p.22). The DC supply was reduced by an auto transformer, dropping the 240V mains voltage to 185V. There is ample space in the power supply compartment of the P.E. Box in front of the original mains transformer to accommodate the additional auto-transformer.

The second modification was to get rid of the extremely noisy fan TI has provided. In addition, it is a 110V fan connected to the 100V tap of the transformer primary. This method is not very safe, it could easily become the source of a sudden death syndrome to the entire content of the P.E. Box!

A partial short of the fan winding could in effect raise the secondary voltages by a factor of 2 or more! TI did not even fit a fuse in the fan circuit! The original fan was replaced with a silent 12V brushless DC fan. The 12V DC supply was taken from a spare 12V winding of the Auto-transformer through a bridge rectifier. The P.E. Box is now absolutely noiseless.

The original power supply was fitted with a .5A rated 7805 regulator for the 5V supply of the disk drives. This is not sufficient to drive 3 disk drives. The T0220 case 7805 was replaced by a T03 case 7805KC rated at 1.5A. The two rectifier diodes in the regulated 12V circuit were also replaced with 2 5401 diodes. These are rated at 3A. The P.E. Box is now able to support 3 disk drives, it is silent and the cards are running cool.



It is a TI-99/4A even if his maker may not recognise it!



It features:

- purpose designed ergonomic work-station,
- RAVE99/105 keyboard,
- motherboard, 80 col.card, Rave keyboard control card, 20 MB hard-disk drive and a SM power supply mounted in the enclosure below the P.E. Box,
- fullhouse P.E. Box: RS232, Myarc 512 Ram Card, P-Code, Modem Card, Triple Tech Card, Myarc HPDCC card, Mechatronic Gram Card, and 2 DSDD disk drives,
- 3 printers: TI thermal printer, MP100 Daisy wheel and a GP-100 dot matrix printer,
- a stand alone Full height DSDD disk drive (now replaced by a 720K 3.5" drive),
- Tallgrass streaming tape backup unit (replaced by a second 20MB hard drive),
- * Modem controls are located in the small box behind the keyboard,
- * the black box contains the printer controls, printer select switch and the master on/off switch (the associated timers are below the desk top). ○

TRACE Subroutine

by Mike Slattery

Ever dreamed of being able to send the output from the TRACE command to your printer instead of having it on the screen? Dream no more, for salvation is at hand. The following program will take the TRACE line numbers off the screen and dump them to your printer. To get the program to work, type it in at the end of your program and insert a GOSUB to the line number. Make sure your program reaches the GOSUB, otherwise it will not work. Here is the program.

```
100 OPEN #1:"PIO"
110 PR$=""
120 FOR R=1 TO 24 :: FOR C=3 TO 28 :: CALL GCHAR(R,C,X)
    :: IF C=3 AND X<>60 THEN 150 :: CALL HCHAR(R,C,32)
    :: IF X=60 THEN 140 :: IF X=31 OR X=32 THEN 150 ::
    IF X=62 THEN X=32
130 PR$=PR$&CHR$(X) :: CT=CT+1 ::
    IF CT>75 AND (X=32 OR X=31) THEN PRINT #1:PR$ ::
    PR$="" :: CT=0
140 NEXT C
150 NEXT R
160 PRINT #1:PR$ :: PR$="" :: CT=0
170 CLOSE #1 :: CALL CLEAR
180 RETURN
```

I am sure this program will save those sweating over a baulky program a lot of trouble. Good luck. ○





SMILE!

There's a source
to help you make
better use of the TI99/4A
and Myarc Geneve 9640

MICROpendium, in its 9th great year,
brings you news, programming tips and
product reviews. And we've done this
every month since February 1984.

SIGN ME UP!

- I want Air mail at \$42.00 US
 Surface mail at \$30.00 US
- Charge my  
- I enclose a money order in US
funds

NAME _____

ADDRESS _____

CITY _____

STATE _____

POSTAL CODE _____

COUNTRY _____

Mail to: MICROPENDIUM
P.O. Box 1343
Round Rock, TX 78680
USA

Tips from the Tigercub #67

by Jim Peterson, Tigercub Software, USA

My TI-PD library now has well over 500 disks of fairware (by author's permission only) and public domain, all arranged by category and as full as possible, provided with loaders by full program name rather than filename. Basic programs converted to Extended Basic, etc. The price is just \$1.50 per disk(1), post paid if at least eight are ordered. TI-PD catalog #5 and the latest supplement is available for \$1 which is deductible from the first order.

In a MICROpendium article, Jerry Stern remarked that it would be quite difficult to write a program that would accept input of a formula and then use the formula. He also thought such a program would be very slow. No programmer could resist a challenge like that, so -

```
100 DISPLAY AT(1,3)ERASE ALL:"PROGRAMMABLE CALCULATOR":
":" V1.1 by Jim Peterson" :: CALL INIT
110 DISPLAY AT(5,1):" Input any mathematical formula
in the form of a valid BASIC statement, using A for t
he value to be calcu-"
120 DISPLAY AT(9,1):"lated and B thru F for the values
to be input.":" Examples - ":" A=(B-C) D-7":" A=B-C
+C*.1-C*.0575":" A=INT(ABS(B-C))-PI"
130 DISPLAY AT(17,1):" To change the formula, enter
0 for all values."
135 DISPLAY AT(20,1):"This version can handle FOR/NEXT 1
oops, IF THEN ELSE, MAX, MIN and <>"
140 DISPLAY AT(24,7):"PRESS ANY KEY" :: DISPLAY AT(24,7)
:"press any key" :: CALL KEY(O,K,S):: IF S=0 THEN 140 EL
SE CALL HCHAR(7,1,32,18*32)
150 A$="" :: DISPLAY AT(8,1)ERASE ALL:"FORMULA?" :: LINP
UT F$ :: ON WARNING NEXT
160 DATA ),182,(,183,=,190,+,193,-,194,*,195,/,196, ,197
,ABS,203,ATN,204,COS,205,EXP,206,INT,207,LOG,208
170 DATA SGN,209,SIN,210,SQR,211,TAN,212,PI,221
175 DATA ::,130,FOR,140,TO,177,NEXT,150,STEP,178,IF,132,
THEN,176,ELSE,129,MAX,223,MIN,22,<,>,191,>,192,"",179
180 RESTORE 160 :: FOR J=1 TO 32 :: READ X$,W
190 P=POS(F$,X$,1):: IF P<>0 THEN F$=SEG$(F$,1,P-1)&CHR$(
(W)&SEG$(F$,P+LEN(X$),255):: GOTO 190
200 NEXT J :: J=0
205 P=POS(F$, "",1):: IF P<>0 THEN F$=SEG$(F$,1,P-1)&SEG
$(F$,P+1,255):: GOTO 205
210 IF J=LEN(F$)THEN 240:: J=J+1:: Z$=SEG$(F$,J,1):: I
F POS(".0123456789",Z$,1)=0 THEN A$=A$&Z$ :: GOTO 210
220 N$=N$&Z$ :: Z$="" :: IF J=LEN(F$)THEN 230 :: J=J+1 :
Z$=SEG$(F$,J,1):: IF POS(".0123456789",Z$,1)<>0 THEN 2
20
230 A$=A$&CHR$(200)&CHR$(LEN(N$))&N$&Z$ :: N$="" :: GOTO
210
240 A$=A$&CHR$(130)&CHR$(136)&CHR$(0):: GOSUB 330 :: CAL
L HCHAR(12,1,32,250)
250 W=0 :: IF POS(A$, "B",1)<>0 THEN DISPLAY AT(12,1):"B=
?" :: ACCEPT AT(12,5):B :: W=W+B
260 IF POS(A$, "C",1)<>0 THEN DISPLAY AT(13,1):"C=?" :: A
CCEPT AT(13,5):C :: W=W+C
270 IF POS(A$, "D",1)<>0 THEN DISPLAY AT(14,1):"D=?" :: A
CCEPT AT(14,5):D :: W=W+D
280 IF POS(A$, "E",1)<>0 THEN DISPLAY AT(15,1):"E=?" :: A
CCEPT AT(15,5):E :: W=W+E
290 IF POS(A$, "F",1)<>0 THEN DISPLAY AT(16,1):"F=?" :: A
CCEPT AT(16,5):F :: W=W+F
300 ON ERROR 310 :: GOTO 320
310 CALL SOUND(400,110,0,-4,0):: DISPLAY AT(12,1):RPT$("
",250):: DISPLAY AT(24,5):"INVALID FORMULA" :: RETURN 1
50
320 IF W=0 THEN 150 :: GOSUB 350 :: DISPLAY AT(18,1):"A=
":A :: GOTO 250
330 CALL PEEK(-31952,A,B):: CALL PEEK(256*A+B-65534,A,B)
:: C=256*A+B-65534
340 FOR J=1 TO LEN(A$):: CALL LOAD(C+J-3,ASC(SEG$(A$,J,1
))):NEXT J :: RETURN
350 |*****
*****
```

This method can also be used for the iterative calculator which I published in Tips #65. Just delete lines 100-140, 280-320 and 350 of the above and substitute-

```
100 DISPLAY AT(3,1)ERASE ALL:"ITERATIVE CALCULATOR V1.1"
:"" by Jim Peterson" :: CALL INIT
110 DISPLAY AT(7,1):" Will solve difficult equations s
uch as A=X X-SQR(X) by iteration."
120 DISPLAY AT(11,1):" Input any mathematical formul
a in the form of a valid BASIC statement, using A for
the known value and X"
130 DISPLAY AT(15,1):"for the value to be determined.
":" Examples - ":" A=X X-SQR(X)":" A=SQR(X X)"
140 DISPLAY AT(20,1):" To change the formula, enter
0 for value to calculate."
```

```
280 DISPLAY AT(12,1):"A=?" :: ACCEPT AT(12,5):C :: DISPL
AY AT(16,5):"" :: IF C=0 THEN 150
```

```
350 X=1 :: GOSUB 380
351 IF A<C THEN DISPLAY AT(14,5):X :: Y=X :: X=X*2 :: GO
SUB 380 :: GOTO 351 ELSE 353
352 IF A<C THEN DISPLAY AT(18,5):X :: Y=X :: X=X/2 :: GO
SUB 380 :: GOTO 352
353 IF A=C OR A=B THEN DISPLAY AT(14,5):"" :: DISPLAY AT
(18,5):"" :: DISPLAY AT(16,5):X :: GOTO 280 ELSE B=A ::
Z=(ABS(X-Y))/2 :: Y=X
354 IF A<C THEN X=X+Z :: DISPLAY AT(14,5):X ELSE X=X-Z :
: DISPLAY AT(18,5):X
355 GOSUB 380 :: GOTO 353
```

Here is a little-known peculiarity of TI Extended Basic - 100 ACCEPT AT(1,1):M\$:: IF M\$="" OR ASC(M\$)<32 THEN 100 Now, if you press Enter, which is a null string or "" you would expect execution to go back to 100 - but it tries to find the ASCII of a null string, and crashes! You must write-

```
IF M$="" THEN 100 ELSE IF ASC(M$)<32 THEN 100.
```

And another peculiarity that caused me an hour of total frustration while trying to debug a program - it is well known that CALL KEY in mode 3, CALL KEY(3,K,S), will cause all subsequent INPUT or ACCEPT AT to be in upper case; but what it actually does is internally depress the Alpha Lock, so that ASCII 97 through 122 are read as 65 through 90 - and it disables character sets above 8, ASCII above 95, so that you cannot INPUT or ACCEPT even the printable characters ASCII 96 or 123 through 126, or any FCTN or CTRL input with an ASCII above that.

If you only use the Triton Super Extended Basic module for running programs, not writing them, you may not be aware of some of its most useful features. For example if you are answering an input prompt by typing something shorter over the default on the screen, you do not have to blank out the remaining characters - just use CTRL C. Take a look at page 8 of the manual for other useful features.

In a recent Tips, I gave a method for reading the entire 13 or 14-digit number which the TI has in memory, by printing it to disk in internal format and reading it back. If I had read the Extended Basic manual more carefully, I would have known a simpler method. If you know where the decimal point will be, just use an IMAGE 14 characters long. Try this -

```
PRINT USING "#####":17/19.07
```

If you do not know where the decimal will be, this subprogram will do the job if the number is within the range of -9,999,999,999 to 9,999,999,999; otherwise it will be in exponential notation as usual.

```
100 CALL CLEAR
110 ACCEPT AT(10,1):X :: CALL FULLNUM(12,1,X):: GOTO 110
20000 SUB FULLNUM(R,C,X):: P=POS(STR$(X),".",1):: IF X>9
999999999 OR X<-999999999 OR P=0 THEN DISPLAY AT(R,C):X
:: SUBEXIT
20010 DISPLAY AT(R,C):USING RPT$("#",P-1)&"."&RPT$("#",1
4-P):X :: SUBEND
```

I worked this one up from a routine in the Swedish newsletter "Programbiten". It will convert to/from any base from 2 to 36.

```
100 CALL CLEAR :: CALL SCREEN(2) :: FOR S=0 TO 12 :: CALL
COLOR(S,16,2) :: NEXT S :: X$="0123456789ABCDEFHIJKLMNO
PQRSTUVWXYZ"
110 DISPLAY AT(3,5):"BASE CONVERTING" :: DISPLAY AT(10,1
):"From which base?":"To which base?"
120 ACCEPT AT(10,18)VALIDATE(DIGIT)SIZE(-2):A :: IF A>36
OR A<2 THEN 120
130 ACCEPT AT(12,16)VALIDATE(DIGIT)SIZE(-2):B :: IF B>36
OR B<2 THEN 130

140 DISPLAY AT(14,1):"Number?" :: ACCEPT AT(14,9)VALIDAT
E(SEG$(X$,1,A)):C$
150 FOR I=LEN(C$) TO 1 STEP -1 :: D$=SEG$(C$,I,1):: IF AS
C(D$)>57 THEN E=ASC(D$)-55 ELSE E=VAL(D$)
160 F=F+(E*(ABS(I-LEN(C$)))):: NEXT I
170 FOR J=INT(LOG(F+0.5)/LOG(B)) TO 0 STEP -1 :: G=INT(F/
B J):: F=F-G*B J
180 IF G>9 THEN H$=H$&CHR$(G+55)ELSE H$=H$&STR$(G)
190 NEXT J :: DISPLAY AT(20,1):H$ :: H$="" :: GOTO 120
```

I have finally replaced my faithful Gemini 10X printer with the new NX1020R and it promptly gave me a problem until I tracked down a serious flaw in its logic. The manual fails to warn you emphasized print cannot be used in combination with condensed print. This is also true of other printers. If you try that combination with them, they condense but do not emphasize. The NX1020 gives me emphasized print but it is not condensed!

The Coco column in Computer Monthly had a contest to write the shortest program to figure first class postage. My one-liner is not as short but does a better job.

```
100 INPUT "OUNCES? ":A :: PRINT .23*(INT(A)-(INT(A)<>A))
+.06 :: GOTO 100
```

Here is how that works. The rate is .29 for the first ounce and .23 for each additional ounce, so we can just multiply ounces by .23 and then add .06 more to the total. However, partial ounces count as full ounces. INT(A) strips off any decimal portion so .23*INT(A) multiplies by the ounces not including the decimal part, if any. (INT(A)<>A) compares A to the integer of A. If they are different, INT(A)<>A has a "truth" value of -1 and a double negative is a plus so 1 is added to the number of ounces to be multiplied by. Otherwise it has a "false" value of 0 so nothing is added.

A self-styled financial adviser has been making the headlines lately by claiming that anyone can become a financial wizard by buying a \$19 compound interest calculator. Save yourself \$19 -

```
100 CALL CLEAR :: ON WARNING NEXT
110 DISPLAY AT(12,1):"A sum of $      invested at      %
interest for years compounded times per year will become"
120 DATA 12,11,5,13,1,4,13,20,2,14,12,3
130 FOR J=1 TO 4 :: READ A,B,C :: ACCEPT AT(A,B)VALIDATE
(NUMERIC)SIZE(C):N(J):: NEXT J
140 FOR J=1 TO N(3)*N(4):: N(1)=N(1)+N(1)*N(2)/100/N(4)
:: NEXT J :: DISPLAY AT(16,1) :USING "$#####.##":N(1)::
RESTORE 120 :: GOTO 110
```

But do not believe the answers you get. Such calculations are worthless unless taxes and inflation are considered. Try this one -

```
100 CALL CLEAR :: ON WARNING NEXT
110 DISPLAY AT(12,1):"A sum of $      invested at      %
interest for years compounded times per year w
ith tax rate of      %"
120 DISPLAY AT(16,1):"and average inflation rate of
% will have a buying power of"
130 DATA 12,11,5,13,1,4,13,20,2,14,12,3,15,23,4,17,4,4
140 FOR J=1 TO 6 :: READ A,B,C :: ACCEPT AT(A,B)VALIDATE
(NUMERIC)SIZE(C):N(J):: NEXT J
150 N(2)=N(2)/100 :: N(5)=N(5)/100 :: N(6)=N(6)/100
```

```
160 FOR J=1 TO N(3)*N(4):: I=N(1)*N(2)/N(4):: I=I-I*N(5)
/N(4):: N(1)=N(1)+I
170 N(1)=N(1)-N(1)*N(6)/N(4)
180 NEXT J :: DISPLAY AT(19,1):USING "$#####.##":N(1)::
RESTORE 130 :: GOTO 110
```

By the first method, \$1000 invested at 7% for 10 years compounded quarterly would double in value to \$2001.60. By the second method, if the interest was taxed at 15% it would still be worth \$1950. But if you factor in an average inflation rate of 4% that \$1950 would only have a buying power of \$1305 - if the price of bread today was \$1 per loaf and that price remained constant in relation to wages, you could buy 1000 loaves today or invest the money and buy 1305, not 2001, ten years from now.

I know that this formula is oversimplified, but there is no way to calculate accurately anyway, since future rates of taxes and inflation cannot be predicted.

In Tips #65 I described a method of using DSKU to make the Funlweb Formatter recognize FCTN A, C and Z instead of &, @ and * to avoid garbled program listings. Jan Alexandersson in Sweden says that can be very dangerous. I should have mentioned that you should make the changes to a separate copy of Funlweb which you should not use to print text formatted by others, and you should not distribute text formatted with these alternative codes and those who use the version of TI-Writer which TI sold overseas should not use this method at all, because it uses those FCTN keys for special letters of foreign languages.

continued from page 2

```
370 HIT=0
380 RETURN
390 CALL SCREEN(16)
400 RANDOMIZE
410 REM DEF CHARACTERS
420 CALL CHAR(130,"38D64438EE9292")
430 CALL CHAR(139,"0010101010101000")
440 CALL CHAR(140,"C3FFDBFFC3BD1866")
450 CALL CHAR(152,"FFFFFFFF7E7E3C3C")
460 CALL COLOR(13,3,1)
470 CALL COLOR(14,7,1)
480 CALL COLOR(16,6,1)
490 RETURN
500 S=16
510 LV=6
520 CALL CLEAR
530 REM PLACE ALIEN
540 IF RND>.5 THEN 610
550 NG=NG+1
560 IF NG>110 THEN 880
570 IF RND>.90 THEN 580 ELSE 600
580 CALL HCHAR(24,INT(RND*31)+1,140)
590 GOTO 610
600 CALL HCHAR(24,INT(RND*31)+1,130)
610 PRINT ::
620 CALL HCHAR(LV-2,S,32)
630 REM MOVE OR FIRE
640 CALL KEY(0,K,ST)
650 IF ST=0 THEN 750
660 IF K<>32 THEN 680
670 GOSUB 160
680 IF K<>68 THEN 720
690 S=S+1
700 IF S>31 THEN 730
710 GOTO 750
720 IF K<>83 THEN 750
730 S=S-1
740 IF S<3 THEN 690
750 CALL GCHAR(LV,S,X)
760 IF (X=130)+(X=140) THEN 800
770 CALL HCHAR(LV,S,152)
780 GOTO 540
790 REM HIT
800 CALL SCREEN(7)
810 CALL SOUND(500,-5,2)
820 CALL HCHAR(LV,S,141)
830 CALL COLOR(16,7,1)
```

continued on page 26

MyArt

by Stephen Shaw, England

MyArt pictures are created on a Geneve using MyArt, using a higher resolution than the standard TI99/4A is capable of. Owners with the 80 column card can load and use MyArt pictures using library disk:

XHi by Alexander Hulpke. Two disks. Allows owners of 80 column cards full access to the graphics modes 6 and 7, with Extended BASIC CALL LINKS to many graphics commands, including the ability to save and load in MyArt format.

For TI99/4A owners who do not have an 80 column card (like me!) there is a version of one of the XHi utility programs which does work; not to display but to PRINT MyArt pictures:

SmArtCopy allows standard TI99/4A owners to print MyArt pictures, be they 16 or 256 colours. You may input print tones for each colour used, or for the pictures available here, use the default tones. As printing is in monochrome, not all MyArt pictures will be suitable. SmArtCopy also allows clipping (printing part of a picture) and a wide range of magnifications. For 16 colour pictures set V=2*H; for 256 colour pictures set V=H; and for true proportions always set S=1. There are two versions of Hardcopy on this disk: one is for TI-Artist pictures (remember this will represent colours, can be clipped, and allows magnification up to 999, separately defined for the two axes) and for YAPP, an extended MyArt format which can handle all MyArt pictures plus the longer (up to 424 lines) YAPP ones.

MyArt Pictures for TI99/4A with Hardcopy, or TI99/4A with 80 column card and XHi, or Geneve with Myart.

Any lower case files are colour/texture data to be loaded with Hardcopy option 5.

MA1...KRIS1...topless lady

MA2...SABER2 (AIRCRAFT); SWAN2; TIGER (not the RLE-MacFlix one).

MA3...JANE2, KRISTIN1, RED1 (topless ladies) and UFO2 (a solo bubble).

MA4...GRNDRAGON; REDDRAGON; HAWK2; HUNICORN (with wings).

MA5...CHARLIEB2 (Bloom County cartoon); ODIE1; MA-L&TRAMP; CHICKEN, FOX; GARFIELD21; SESAM (characters from Sesame Street).

MA6...GOBLIN2; WDRKUGEL (ball on patterned design).

MA7...BALLET2 (dancer); KARTE (world map); TRIANGLE

MA8...HOUSDREAM2 (nice house); HWOLF (howling); RABBIT (cartoon).

MA9...CITY2/MY; SNOOPY

MA10...DRACO superb dragon (H=3, V=6, S=1 for an A4 size picture) plus a reflecting ball on a chequer board. (MA10 also has an earlier version of Hardcopy which will not handle YAPP or TI-Artist pictures).

MA11...a YAPP picture of the Mona Lisa, an unclipped transfer of a MacFlix picture. For a good well proportioned picture I suggest you set the variables as follows: Option 2: H=5, V=3, S=0 (this gives similar proportions to those given on the original MacFlix file using PixPro Bit Image mode); Option 3: H=105, V=20, dH=285, dV=400.

continued from page 19

* S=D O=DO This remark is a reminder of the file names

```

DEF D          Program module entry point name

STRREF EQU >2014  Utility to transfer a string to assy

SAVRTN BSS 2      Memory assignment for return address
BUFF BSS 4        4 byte buffer for the incoming string
BUFFER BSS 3      3 byte buffer for the output code
* The byte sized numbers in NOS create the 7 segment
* codes to drive the displays. The digit created is
* above each corresponding byte value.
*      0 1 2 3 4 5 6 6 8 9
NOS BYTE 3,159,37,13,153,73,65,31,1,25

H30 BYTE >30     Hex 30 used to convert a byte from
EVEN                               string to byte sized integer.

BLANK DATA >FFFF  All outputs logic 1 turn display off
THREE DATA 3      A word sized value of three

* CALL LINK("D",VALUE$) Example of link from x/basic

D MOV R11,@SAVRTN  Save the return address

CLR RO           Element in the argument in x/b link
LI R1,1         Argument number in the x/b link
LI R2,BUFF      Where to put string in assembly
MOVB @NOS,@BUFF Max string length allowed
BLWP @STRREF    Get the string from x/b

MOV @BLANK,@BUFFER Place the blanking chars
MOVB @BLANK,@BUFFER+2 in the output buffer.

MOVB @BUFF+0,RO  Get the actual string length
SRL RO,8        for loop counter as a word.
LI R1,BUFF+1    Source of number to process
LI R2,BUFFER+0  Destination of processing

* Right justification is achieved by setting the start
* point in the output buffer, dependant on the length
* of the input string. By taking the string length from
* 3 then adding the result to the buffer start point,
* the modified start point is found in the buffer.

MOV RO,R4       Put the string length in R4
S @THREE,R4     Str len minus 3=negative value
S R4,R2        Neg minus Neg = positive, R2 was
* start point in output buffer for a max length number.

* Conversion process, first a string format digit (in
* hex value range between >30 & >39), has hex 30 taken
* away. Then the byte sized digit between 0 and 9 is
* placed in R3. The byte in R3 is converted to a word
* in R3. The word value in R3 is used as an offset from
* the start of the byte values in NOS & the values are
* placed in the output buffer pointed to by the address
* in R2. R0 controls the loop, since it starts with
* the length of the input string.

LOOP1 SB @H30,*R1 Take hex 30 off a hex input digit
MOVB *R1+,R3     Put the byte (range 0 to 9) in R3
SRL R3,8        Make the byte in R3 into a word
MOVB @NOS(R3),*R2+ Put pattern value in buffer
DEC RO         Decrement control loop counter
JNE LOOP1      If not finished, loop to do next

MOV @BUFFER+0,@>9300 Write three bytes from
MOVB @BUFFER+2,@>9302 BUFFER to the Wire I/O

END MOV @SAVRTN,R11 Put saved return address in R11
RT

END

```

continued from page 20

So there you are! We have looked at the kinds of memory, the owner of the memories, where the memory is, what is in the memories, and the terminology used to talk about all of this. There are a few tricky and devious things we have not looked into (like bank switching). Maybe in future articles we will have a look at them.

My Genealogy Program

by Jim Peterson, Tigercub Software, USA.

Some 20 years ago, my late brother researched our family ancestry and gave me a copy of his work. I was not too much interested. It consisted of charts branching backwards in time, showing parents, grandparents, etc., much like a Biblical recitation of "and Jonah begat Abraham and Abraham begat Noah", etc., etc., except that in modern genealogy the mother who actually bears the child is at least given second billing.

But last year a gift of some old family photos and a visit to some graveyards kindled my interest. However, I wanted to do more than just trace that forking family tree backwards. I wanted to know who my grandfather's cousins were, and who their children and grandchildren were.

I was told that there was no really good genealogy program for the TI-99/A. I obtained a sample of a family group sheet, one of the standard tools used by genealogists, and began recording data on it. I soon filled a disk with DV/80 files of those, which printed out to a very thick file of pages with a lot of wasted space.

I thought of trying to write a genealogy program, but was not sure what I wanted. About that time, I had an amazing piece of good luck - I was put in contact with a distant relative in Sweden who had researched the family history back into the 1700s and beyond!

He sent me a 3.5 disk containing his genealogy program for the PC, and his files on 1400 family members. Since I do not own a PC and never intend to, I ran to Chuck Grimes for help. He accessed the program's options and printed out for me a list of all 1400 names, a cross-reference list of all children, and two cross-reference lists of marriages, plus several of those family tree charts.

About 1000 of those 1400 names were of the Swedish researcher's father's relatives and his wife's relatives, which were of no interest to me, so I went to work to extract the 400 who were actually my blood relatives. After about a week of checking one list against another, back and forth, I was not too impressed with the program.

So, again I thought about writing a genealogy program. I was not interested in being able to sort data seventeen ways from Sunday, and I did not care about printing out those bare-bones family trees, but I wanted to be able to easily find a person by name, and find a complete record of parents, spouse, children, biographical data, and sources of data.

Such a program would be difficult to write and unnecessary. I realized that the best program for my purpose would be no program at all. The magic of Funnelweb and the efficiency of the TI disk controller was all that I needed. I booted up Funnelweb, went into the Editor, set the tab at 39, and typed -

```
[1] JAMES WARREN PETERSON is the son of
[2]> NORTH EDWIN PETERSON and [3]> LINNIE LEONA
STEVENS. He was born 20/8 1923 in Pelican Rapids,
Otter Tail County, Minnesota. He was married 7/7
1956 in Tokyo, Japan to [4]> MIDORI IMAI. Their
children are [5]> MARIANN MIEKO and [6]> ALAN
EDWIN.
```

And that was followed with some biographical data. I saved it to disk, with SF to preserve the tab setting, as filename 001. The > after an index number means that a file exists under that number, with information about the person. So, I typed up a similar file about my father and saved it as 002; and so on. Padding the number with 0's causes the disk controller to catalog

filenames from 001 to 999 in numerical sequence. Now, if I need to add to a file, I just load it into Funnelweb and go to work. Since it is in 40-column format, it is easy to edit on-screen. The TI disk controller can only handle 127 files on a disk, but many of my 400 names are those of children listed in their parents's file without enough data to require a file of their own. When I do run over the 127 limit, it is easy to use an additional disk. If I get more information about such a child, I will just add a > after his number, and set up a new record for him.

What about a printout? I could easily create an .IF file listing all those filenames in numeric sequence, and print them all through the Formatter, using dot commands to change them to 80-column width. I enclosed the index numbers in brackets so I could easily .TL to double-strike, emphasize or underline them. However, I like 40-character 2-column text, so I wrote a little program to catalogue drives 1 and 2 and print all the files in sequence in two columns.

Now, how about finding records? I booted up Funnelweb again, set the tabs at 5, 35, 50, 55, 60 and 65 and began entering names in index number sequence by index number, first name, last name, file number, father's index number, mother's index number and spouse's index number. The resulting file was too big for a simple sorting routine to handle, so I tried using Peter Hoddie's fairware program SORT EXPERIMENT, sorting on the last name field with a secondary sort on the first name. I thought that it did a perfect job, until I found that many names were missing. The documentation for SORT EXPERIMENT says it will handle up to 1000 records or 24k, whichever comes first. It fails to mention that after reading in 24k of data it will begin to sort, without warning you that it did not read the complete file!

So I went to Dennis Faherty's TI-SORT, sold by Insebot. The documentation for that program is very neatly printed but difficult for me to understand. I finally figured it out, and produced an index in alphabetic sequence. I plan to update it with Funnelweb, inserting lines in the proper place, so I will not have to sort it again.

I now have a text-formatted genealogy which I can easily and quickly update. I can print copies of the index and text to send to relatives who do not have a computer, and the printouts will be very easy for them to understand. If any of them do have a computer and a genealogy program, it will be very easy for them to copy the data.

So once again, the best program is the simplest program that will do the job, and the simplest of all programs is no program at all. ○

continued from page 14

```
840 FOR DEL=1 TO 31
850 NEXT DEL
860 CALL HCHAR(LV,S,152)
870 CALL SCREEN(16)
880 FOR J=660 TO 120 STEP -20
890 CALL SOUND(100,J,2)
900 NEXT J
910 IF SC<HSC THEN 930
920 HSC=SC
930 PRINT "SCORE=";SC::
940 PRINT "HIGH SCORE=";HSC:::
950 INPUT "PLAY AGAIN? ":Q$
960 IF SEG$(Q$,1,1)="N" THEN 1010
970 SC=0
980 NG=0
990 CALL COLOR(16,6,1)
1000 GOTO 150
1010 END
```

When typing in the above program make sure that you copy it exactly as it appears and pay particular attention to colons, semi-colons, commas and blank spaces. Have fun, Percy Harrison. ○

Regional Group Reports

Meeting Summary For DECEMBER

Banana Coast	13/12/92	Sawtell
Central Coast	12/12/92	Saratoga
Glebe	10/12/92	Glebe
Hunter Valley	12/12/92	
Illawarra	14/12/92	Keiraville
Liverpool	11/12/92	
Northern Suburbs	24/12/92	
Sutherland	18/12/92	Jannali

BANANA COAST Regional Group (Coffs Harbour Environs)

We never miss meeting at Kerry Harrison's residence 15 Scarba St. Coffs Harbour, 2 pm second Sunday of the month. Visitors are most welcome. Contact Kerry 52 3736, Kevin 53 2649, Rex 51 2485 or John 54 1451.

CENTRAL COAST Regional Group

Regular meetings are normally held on the second Saturday of each month, 6.30pm at the home of John Goulton, 34 Mimosa Ave., Saratoga, (043) 69 3990. Contact Russell Welham (043)92 4000.

GLEBE Regional Group

Regular meetings are normally on the Thursday evening following the first Saturday of the month, at 8pm at 43 Boyce Street, Glebe. Contact Mike Slattery, (02) 692 8162.

HUNTER VALLEY Regional Group

The meetings are usually held on the second Saturday of each month at members homes starting at 3:15 pm. Check the location with Geoff Phillips on (049) 428 176. Note that after 9:00 pm this number is used for the ZZAP BBS which includes TI-99 information. Geoff.

ILLAWARRA Regional Group

Regular meetings are normally held on the second Monday of each month after the TISHUG Sydney meeting, except January, at 7.30pm, at the home of Geoff & Heather Trott, 20 Robsons Road, Keiraville. A variety of activities accompany our meetings, including Word Processing, Spreadsheets and hardware repairs. Contact Lou Amadio on (042) 28 4906 for more information.

LIVERPOOL Regional Group

Regular meeting date is the Friday following the TISHUG Sydney meeting at 7.30 pm. Contact Larry Saunders (02) 6447377 (home) or (02) 7598441 (work) for more information.

NORTHERN SUBURBS Regional Group

Regular meetings are held on the fourth Thursday of the month. If you want any information please ring Dennis Norman on (02)452 3920, or Dick Warburton on (02) 918 8132. Come and join in our fun, Dick Warburton.

SUTHERLAND Regional Group

Regular meetings are held on the third Friday of each month at the home of Peter Young, 51 Jannali Avenue, Jannali at 7.30pm. Peter Young

TISHUG in Sydney

Monthly meetings start promptly at 2pm (except for full day tutorials) on the first Saturday of the month that is not part of a long weekend. They are held at the RYDE INFANTS SCHOOL, Tucker Street (Post Office end), Ryde. Regular items include news from the directors, the publications library, the shop, and demonstrations of monthly software.

DECEMBER MEETING - 5th DECEMBER

The big item this month will be the annual AGM involving the election of officers. Please bring along the copies of the minutes, etc found in the November

issue. The AGM will be followed by a BBQ and an informal discussion about the Faire and anything else that interests us.

The cut-off dates for submitting articles to the Editor for the TND via the BBS or otherwise are:

January/February 5th January

These dates are all Sundays and there is no guarantee that they will make the magazine unless they are uploaded by 6:00pm, at the latest. Longer articles should be to hand well before the above dates to ensure there is time to edit them.

Newsletter Update

by Bob Relyea

TIBUG (Brisbane), July, 1992: Editorial; 3.5 in. Disk Drives; G- An Introduction; G, The High Resolution Language; In The P.O. Box; Meeting Minutes; New Funnelweb Editor; OOPS; Shop; Sixteen - XB Program; Sorting; Tips #41; TI Font Maker; What's News; Word Processing #4; Advertisement for Sound Fix; August, 1992: Editorial; Meeting Minutes; Word Processing #5; Shop; What's News; Murphy's Law; Trading Post; Call Load List; BBS; Tips #42; In the P.O. Box; The TI Learning Machine; Games.

SPIRIT OF 99, September, 1992: Assembly Lesson 5; Automusic; About the Dom; Conni Calendar; Conni Minutes; Dues Announcement; DV/80 to Program Conversion; Graphing Data; Letters To The Editor; Picnic; Plus v.2.0; Review of CSGD - Label Maker; Scratchpads; The Blood Bank (Hymn, Here I am); The Star NX-1020; TI World News.

OTTAWA, September, 1992: Coming Events; Editor's Notes; Two Cents Worth by Bill Gard; Fast Extended Basic; Programs That Write Programs, Part 1; Hotline Numbers.

THE PUG PERIPHERAL, April, 1992: President's Page; From the Librarian; Gen-Tri (Review); I Like Brain Games; Mailroom (Review); Meeting Minutes; Tips #63.

June, 1992: President's Page; Designer Labels (Review); Tips #62; Brain Games; Print Using Your Printer; Help Wanted; Archiving a Headache?; TIBase Tip; Rules of Membership; Reward Offered.

UGOC ROM, August, 1992: Editor Popping Off; President's Message; July Board Minutes; The Member Ship; Test File; Printers & Other Garbage;; Epson (Son of EP); Multi-Column printing; XB Miscellany #16; Convert Program Listing; Lima TI UG; Classic Computer Class; Geoff Warner Bio; Beggars I Have Known.

September, 1992: Editor Popping Off; President's Message; August Board Minutes; The Member Ship; My Life Story; XB Miscellany #17; Words Of Wisdom; Degree Mark; Printing the Caret; Border Paper with TL; BBS Tutorial #2; Write Stuff; Labor Day Holiday; My Genealogy Program; TI Acronyms.

LA 99ers, Aug/Sept, 1992: Hole Currents #2 & 3, A discussion of data transmission methods, Chaos and Fractals by W. Richardson; Computer Robot; New Ribbons from Old; Matchit (a game program); New Hardware; LA 99ers Library.

TIDBITS, August, 1992: Tom's Observations; My Favourite TI Tools; Link Terminal Emulator; Why are Firetrucks Red?; Line Upon Line; All About Character Sets; The ROMOX ECPC; Parable of the TI; Auto-Loading Cassette Files; Back Up Copy?; Program Bit;

THE BOSTON COMPUTER SOCIETY, July, 1992: Basic Miscellany; Hole Currents #1, 2 & 3. ○