

OK Quiz Time!

DO YOU HAVE A MODEM ?

Can you get FREE round the clock advice and help?

Can you get instant news, gossip and "hands on" reviews?
24 HOURS A DAY?

Can you get FREE national electronic mail?

Can you "chat" with 70,000 other micro users?

Can you publish your own database for all to see?

Can you take part in national multi-user strategy games?

Can you book a seat on the 3.15 from Euston - at midnight?

NO? That's because you don't have

Micronet 800

*Make the most
of your micro*

SEE CENTRE PAGES FOR DETAILS OF AN EXCLUSIVE SPECIAL OFFER
FOR TI USERS - HALF PRICE SUBSCRIPTION!



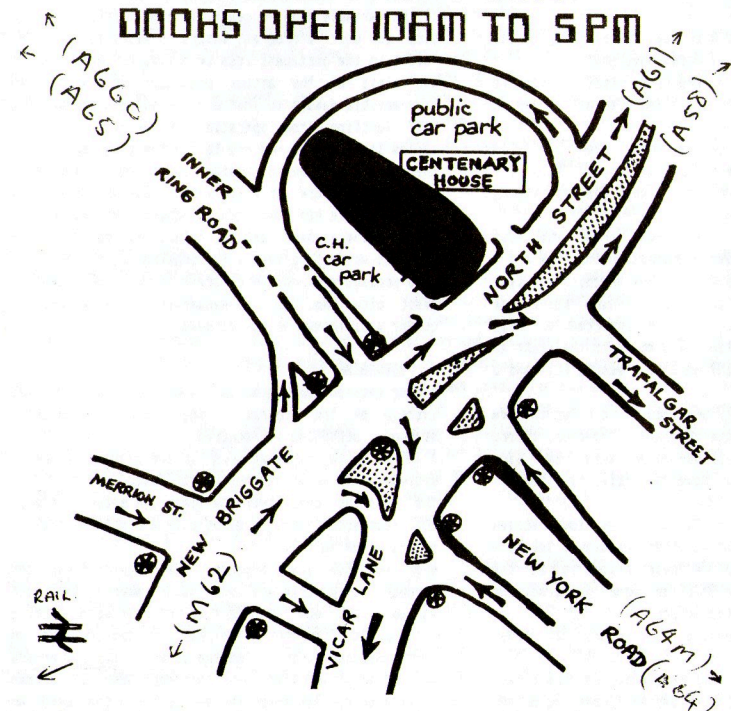
INDEX

- * 4 Successful Programming on TI99 S.Shaw
- * 6 Ext Basic M.Kabala MSP*
- * 8 TI*MES Library reviews Games G.Hilton.
- * 9 Leeds Show P.Marsden.*
- * 10 USA Visit by H.Greenburg (Who writes again to inform you) *
- * 13 D.I.Y. 32k + Cartridge Expander Notes by G. Wolstenholme. *
- * 14 Data Files without tears by P.Walker. *
- * 20 Tips From Tigercub J.Peterson.*
- * 24 DIY Project I.H.Moore. *
- * 27 Classified Adverts (Lots of bargains for TI Users). *
- * 28 SPECIAL FEATURE ON MODEMS. By N.Bosworth, A.Davey. *
- * 32 Using PRESTEL-MICRONET 800 P.Walker.
- * 33 Your Letters.*
- * 34 A Game of Draughts TI BASIC S.Michel. *
- * 38 Review of GramKracker. D.Michaels.*
- * 42 RAMBLES S.Shaw. *

TI99/4A USERS SHOW IN LEEDS

HOW TO GET THERE

CENTENARY HOUSE, NORTH ST
SATURDAY 3RD MAY
DOORS OPEN 10AM TO 5PM



For those people coming from the west via the M62, approach Leeds via the M62, and follow the signs for York on the A64(M) inner ring road. After passing the exits for Otley (A660) and Harrogate (A61), pull off RIGHT into the city centre. The building is then on the left by the traffic lights.

For those people coming via the M1, follow the Harrogate signs, and this road will take you through the city centre, and pass the building, just over the inner ring road.

For motorists coming from the A1, travel down the A64 or A63 into Leeds, and take the second exit after the lights at the start of the inner ring road. The building is on the right, over the bridge.

Rail travellers should go to the traffic lights outside the station, turn along Boar Lane, then turn left at the next ROAD crossroad with traffic lights. This is Briggate, and will take you to the Institute in about half a mile.

Coach travellers will normally arrive at the Wellington Road bus station, and will have to turn left to City Square, and then follow the same route as the rail travellers.

Welcome _____ to TI99/4a EXCHANGE

TI*MES TI*MES TI*MES TI*MES TI*MES TI*MES TI*MES TI*MES TI*MES TI*MES
SPRING ISSUE NUMBER TWELVE

40, Barrhill, Patcham, BRIGHTON, East Sussex, BN18UF. Tel: 0273 503968 (evenings)

4A TI WHO WILL NEVER DIE..

Another three months have passed and still lots of things happening. In this issue we bring you lots of new developments. Our TI99/4a will not lie down. There is a new language known as C99, new hardware which will explore the potential even more.

The TI USERS SHOW to be held in LEEDS (thanks to Phillip Marsden for finding a venue), will be a good example of the new developments taking place. We hope you will make an effort to come and see what is new on the TI99/4a front. Unlike the previous Shows the emphasis will be on individuals who have something to show and talk to you about our TI99/4a. Of course the auctions, swaps, demos and dealers will be there to tempt us with the exciting new add ons and software.

Our cover introduces you to another exciting development for TI99/4a users here in Britain. A colourful graphical and live network which you can intercommunicate directly with thousands of home computer users using your TI99/4a, fancy talking to a beeb or an apple perhaps an apricot, no I would rather keep to making contact with TI Users, but theres nothing to stop the TI99 letting all those others know we are here alive and well.

Thanks to MICRONET 800 for the generous offer they are making to us TI Users. I hope you take up the offer and join us on the network.

TI*MES is certainly now well and truly part of the electronic mail boom. By the way, if you have a Modem and have taken up the offer of joining Micronet, you can contact me on our mailbox number 273503968. I also wish to pay tribute to Neville Bosworth, Alan Davey and Peter Walker who have put in alot of work to explore the potential of Modems and networks, many thanks indeed.

As the Show in Leeds is about to take place we are already looking ahead to the next year. PCW in September, in the same month we shall attend a small gathering in Bournemouth. Next Year, (if there is enough interest) in March 1987 a charter flight to the USA. To visit the show of a lifetime 99'FEST-WEST-87. We will be joining Teresa Masters of the LA 99ers Group in LOS ANGELES as guests. This Show is a 99er World event not to be missed. Advance information will be available at the Leeds show Group stand. If you can't wait until the Summer issue of TI*MES please send an S.A.E. for a leaflet and booking form.

Finally just to prove that our TI99/4a lives on and on, New Day company are producing an exclusive 99er magazine on TAPE or Disk called 4 Front. With all this happening the TI will never Die!!

Happy 99ing,

Blime Dally

TI99/4a Exchange TI*MES newsletter is supported only by its subscribers. This TI users Group is INDEPENDANT of Texas Instruments and is completely non profit making. TI*MES is published quarterly, JANUARY, APRIL, JULY, and OCTOBER months. The annual subscription is £10 and includes 4 newsletters. Editorial etc is provided by group members, other user-groups and other related sources. Views expressed are those of the writer and not necessarily those of TI99/4a Exchange. Whilst efforts are made to ensure accuracy no responsibility can be accepted by TI 99/4a Exchange as a result of the applying of such information found within the pages of TI*MES. You are invited to contribute copy for publication in TI*MES. If you would like to make a contribution please submit COPY ON A4 ONLY this MUST be TYPED with a disk or tape if a program is included (better still use the word processing program in the library). FINAL PRESS DATE 10 20TH PRIOR TO PUBLICATION MONTH. Unaccepted material will be returned ONLY if accompanied by a S.A.E. No material may be reproduced without giving credit to the writer and TI99/4a EXCHANGE.

SUCCESSFUL PROGRAMMING

EDIT" and "DEBUG"

A word about editing and debugging: no matter how good a typist you are, even entering a short program will need the use of these facilities.

EDITING is what you do when you change a program line - perhaps only one character in the line.

A number of computers use 'screen editing', where you move the cursor around the screen until it is placed where you want to make your alteration.

The TI99/4A uses a 'line editor' for programs (some modules also use a screen editor, eg TI-WRITER). To use a line editor, you first select the line number you wish to amend, place it on the bottom of the screen, and move the cursor along the LINE until you hit the place to be amended.

To bring the line you want on screen, key in the line number, then hold FCTN down and press key E or X. Your line will appear with the cursor at the beginning of the line.

Use FCTN and keys S and D to move the cursor over the line without deleting or altering anything.

If you wish merely to alter the line, typing over it may be sufficient, but the 99/4A also allows you to delete and insert characters.

To DELETE a character, place the cursor over it (FCTN plus S or D) and then press FCTN and key 1. This will delete the character the cursor is on, and everything following will move one space to the left.

To INSERT text, place the cursor after the last character you wish to leave untouched, using FCTN and S D, then press FCTN and key 2. Now anything you type will force everything after the insert point one character to the right, and your inserted text will appear. The cursor position also moves to the right as you type. You leave INSERT mode by pressing FCTN and S or D, or pressing ENTER.

When your line is correct, press ENTER to enter the new line into memory, and leave EDIT mode. However, if you wish to amend also the line before or after the line just finished, you may move directly to that by pressing FCTN and E or X.

The maximum line length in TI Basic is four screen lines, but the computer is often capable of taking a longer line, as the absolute restriction is on the length of the line in BYTES of memory used, not the number of characters on the screen.

You may use the edit function to insert 'overlong' lines as follows:

Type in the last part of the required line first. Press ENTER. With the line on screen, press FCTN and key 2, then key in the first part of your line, from the beginning. The part you first entered is pushed to the right.

Now return the line to the screen using the edit mode: type in the line number and then press FCTN and key X. It is possible to overfill a line this way, and you will receive an error message if you do. However you will usually be able to go to an extra half screen line, and in some cases you may be able to squeeze two extra lines in!

The advantage of putting as much as possible in a line is that by using less program lines you save a little memory. In general, lines with a lot of numbers in will be difficult to expand in this fashion, but lines with a lot of text or commands can usually be considerably extended.

When the line is fully entered, press ENTER. Then LIST the line just to make sure it is all right.

You do not have to INSERT at the beginning of the line, but you may find it easier to do so. Give the computer time to move all the characters to the right when using INSERT. Keep your eyes on the screen.

A brief word of warning: in TI BASIC, your program uses the same area of memory as the values of variables, separated by a 'marker' in memory. As a program runs, the free memory is continually filling. Unwanted values are only purged when the memory is full, resulting in short pauses in program operation.

If the variable area of memory is almost full when you stop to edit a program, inserting extra material MAY result in the permanent loss of the marker which MARKS the limit of the actual program. This will cause irreversible damage to your program and may prevent the LIST function from operating correctly, or cause a system lockout when you try to run the program.

Therefore try to avoid running the program and then editing it! Before you edit, take a copy of the program to tape - this has the added benefit of apparently clearing the garbage, and not only gives you a security copy but actually prevents the problem occurring!

This problem will not be apparent if you use extended basic plus the 32k ram expansion, as variables then occupy a different sector of memory.

DEBUGGING

Having entered your program, you type in RUN, and instead of the program running, you receive an error message. DEBUGGING is required.

A BUG is quite simply an error in the program, either a mistype or an error in your use of Basic.

The computer does check the lines you enter, but will only spot such things as using only a single bracket (or quotation mark ".

When you RUN your program, the computer first goes through your program and sets aside memory for each variable and sub program that you have used. During this 'prescan' further errors may be spotted and an error message printed on the screen. Typical errors spotted at this time are incorrect use of arrays (trying to use DIM after the variable has been used) or a mismatched number of FORs and NEXTs. Most errors will only produce an error message as the computer finds them when your program is actually running - for instance trying to GOTO a non-existent line number, or trying to RETURN when there is no outstanding GOSUB. The error messages generated by the TI99/4A are well described in the manual, and will usually indicate a line number in which the computer has met something it cannot cope with.

Unfortunately, the actual error may not be in the line stated in the error message.

For instance, BAD VALUE IN 100 may refer to:
100 CALL HCHAR(ROW,COL,42)

The program line is correct. The error message has appeared because one or both of the variables ROW and COL have a value which is out of range.

(Using CALL HCHAR the ROW and COLUMN values passed to the sub program must be in the ranges 1 to 24 and 1 to 32 respectively. If you go outside this range the computer will halt with an error message.)

To see what the values are, when the error message appears, just type in PRINT ROW;COL then press ENTER. The two values on screen will be the current values of ROW and COL, which have caused the problem.

The job is then to review the program to see how the variables obtained that value, and see what changes need to be made.

This ability to check variable values after an error message is very valuable. Note that once you amend a program line all variables will be reset to zero.

Frequent causes of problem bugs are DATA statements, with a comma too many or too few. The ability to check variable values is useful here:

```
AN array is being filled from a DATA statement, eg
FOR T=1 TO 5
READ N
VAR(T)=N
NEXT T
DATA 23,54,8,A,5
```

An error message will be generated by the above, as 'A' is not a number. When the error message appears it is possible to enter:

```
PRINT T;VAR(T)
```

This will provide the clues needed to lead to the erroneous DATA line - which may be several hundred program lines away!

In such cases you need to have a good idea of what values should be found in connection with each variable, and you may need to spend some time working through the program.

If you cannot fathom why a program is not working as it should merely by reading the LIST, the TI99/4A also has a TRACE option.

Key in TRACE and then RUN. The line numbers will be listed on screen as the program progresses, and you can watch for an unexpected line transfer as the computer moves from one line to the next. Using TRACE will disrupt any screen display. Switch TRACE off by keying in UNTRACE.

To find out what the variable values are at a particular point in the program, you can insert PRINT statements in the program, or instruct the program to BREAK by adding a line in the appropriate place with the instruction: BREAK.

When the program stops you may enter PRINT VAR etc.

To continue the program enter CON (and press ENTER)

NB: Remove the BREAK line when you no longer need it! As your programs become longer, so it becomes more difficult to spot the errors, but finding and removing errors is a very good (if time consuming) way of learning how to use your TI99/4A. With time and experience you will learn to quickly spot the easy bugs and to tackle the harder ones in a logical fashion. Read your manual as often as necessary, especially the section on ERROR MESSAGES. Often the answer to a difficult bug is there just waiting for you to read it.

SYSTEM LOCK OUTS

A lock out has occurred if your console no longer responds to the keyboard (especially QUIT) and ceases to function normally. Unusual sound and graphic effects may occur.

It is quite normal for all computers (and word processors) to lock out from time to time. The cause is an error in the instructions passed to the processor, which it cannot deal with. There are a number of causes:

A STATIC discharge is a frequent cause of problems.

Although computers no longer need the carefully controlled environment of the mainframe, they remain sensitive to static. The problem is most acute in warm dry weather, or if you wear clothing made of artificial fibers (acrylics are particularly bad). Nylon carpeting can also be a problem. The TI99/4A can handle static quite well, but you may meet the problem.

Use of cotton clothing, a humidifier, and an anti static spray on the carpet may be called for in especially hostile environments. A conductive carpet is also sold by some computer suppliers.

Poor communication with modules or peripherals may also be a problem: the contacts are essentially self cleaning, but it may be necessary to disconnect/connect a few times to make good contact. Contacts are silver plated and are subject to tarnishing, and may require this treatment if a module or peripheral is not used for some time. Tar can be deposited on the contacts if there are smokers in the room. In extremely severe cases of pollution, an isopropyl alcohol solvent may be used but great care is required to prevent damage.

Some modules contain insufficient error traps and permit you to pass confusing instructions to the processor.

Loss of the stack/program marker can cause problems in TI BASIC. This occurs when you run a program (filling the stack) and then add to the program. In some cases the computer will add the stack to the program with sometimes colourful results, but permanently destroying the program (if a lock out does not occur). After running a program, it is wise to save it before editing: this appears to clear the stack.

When the computer ceases to function, and possibly makes a piercing sound, DO NOT panic! The only way out of a lock out is to switch off and (after a few seconds) restart. Even the MOST expensive systems sometimes

Extracted from GETTING STARTED WITH THE TI99/4A
By Stephen Shaw (c) Stephen Shaw 1983.

MORE ON EXTENDED BASIC

PARAMETER PASSING BETWEEN EXTENDED BASIC PROGRAMS

by Mike Kabala

One of the more advanced features available to the Extended BASIC programmer is the ability to chain programs by using the RUN statement within a program. This allows the creation of software that would not fit into the memory all at once. When you're done with one program segment, just "RUN" the next one.

Unfortunately, there is one bug. The RUN statement clears out all variables even if it is executed from within a program. Type in the following 2 programs and you will see for yourself. Be sure to save the first as "DSK1.DEMO1" and the second as "DSK1.DEMO1A" and then load and run "DSK1.DEMO1".

```
100 CALL CLEAR
110 DISPLAY AT(1,1):
    "TYPE SOMETHING."
120 ! ACCEPT AT statement is
130 ! needed to put text in
140 ! a predictable screen
150 ! location
160 ACCEPT AT(2,1):A$
170 ! All variables will be
180 ! cleared when the next
190 ! statement is executed
200 RUN "DSK1.DEMO1A"
```

```
100 CALL CLEAR
110 ! Variable has been
120 ! cleared by RUN
130 ! statement
140 PRINT A$
```

See what I mean? Somewhere between DEMO1 and DEMO1A, the value of A\$ got lost. Now you could, if you wanted to, create a file, store A\$ in it, and then read A\$ from that file after entering DEMO1A, but that seems rather wasteful if you only need to pass one or two variables. In the remainder of this article, I will show you two other methods I have found to get around the problem.

The first method will work even if all you have is a console and cassette recorder (although you'll have to change all of the program names to "CS1"). That method is to use the screen as auxiliary memory. Just display the information you want before executing the RUN statement and your chained program will be able to take the data right off the screen as demonstrated below.

Load DEMO1 back into your computer and modify its RUN statement to chain in DEMO2A as shown below. Then save the first program as "DSK1.DEMO2" and the second one as "DSK1.DEMO2A". Finally, load and run "DSK1.DEMO2" and what you type in the first program should be correctly printed by the second.

```
100 CALL CLEAR
110 DISPLAY AT(1,1):
    "TYPE SOMETHING."
120 ! ACCEPT AT statement is
130 ! needed to put text in
140 ! a predictable screen
150 ! location
160 ACCEPT AT(2,1):A$
170 ! All variables will be
180 ! cleared when the next
190 ! statement is executed
200 RUN "DSK1.DEMO2A"
```

```
100 ! Read what last program
110 ! left on the screen.
120 ! Offset of 2 is needed
130 ! for difference between
140 ! ACCEPT and GCHAR
150 ! statements
160 A$=""
170 FOR I=3 TO 30
180 CALL GCHAR(2,I,A)
190 A$=A$&CHR$(A)
200 NEXT I
210 ! Parameter has been
220 ! passed using the
230 ! screen as auxiliary
240 ! memory
250 CALL CLEAR
260 DISPLAY AT(5,1):"YOU TYPED:"
270 DISPLAY AT(6,1):A$
```

For this second method you will need to have the 32K memory

(c)
MSP 99 NEWSLETTER

expansion installed. The trick here is to use the space reserved for an assembly language program for temporary storage. This has the advantage that you don't need to display something to pass it to the next program. Be sure to use the memory between 9984 and 16184 after executing CALL INIT because the rest of memory is already in use by your program and other things.

Save the first program as "DSK1.DEMO3" and the second as "DSK1.DEMO3A". Then run "DSK1.DEMO3".

```
100 CALL CLEAR
110 PRINT "TYPE SOMETHING:"
120 ACCEPT A$
130 !
140 ! Reserve space for
150 ! assembly language
160 ! program
170 !
180 CALL INIT
190 !
200 ! Store length of string
210 !
220 A=LEN(A$)
230 CALL LOAD(9984,A)
240 !
250 ! Store string
260 !
270 FOR I=1 TO A
280 CALL LOAD(I+9984,
    ASC(SEG$(A$,I,1)))
290 NEXT I
300 CALL CLEAR
310 RUN "DSK1.DEMO3A"
```

```
100 CALL CLEAR
110 PRINT "YOU TYPED:"
120 !
130 ! Get string length
140 !
150 CALL PEEK(9984,L)
160 !
170 ! Get string
180 !
190 X$=""
200 FOR I=1 TO L
210 CALL PEEK(I+9984,X)
220 X$=X$&CHR$(X)
230 NEXT I
240 !
250 ! Print string
```

```
260 !
270 PRINT X$
```

You may have noticed that I have only used string type variables in these examples. That's because the data manipulations required are a bit simpler for string variables. That doesn't mean that you can't pass numeric variables, however. Just use STR\$() to convert them to string variables and VAL() to convert them back. I'm sure it wouldn't take too much effort to figure out a way to pass them directly, either. Just be careful if you use the second method that you don't try to pass a number greater than 127 or less than -128 unless you split it up into more than one byte.

Finally, if you don't have a disk drive and want to try the first method, type in the program below and save it to CS1. Then type in the DEMO2A program and save it to the same cassette WITHOUT rewinding it. Then rewind the cassette, load the first program, and run it.

After the program begins running, it will ask you to rewind the cassette. Ignore this message and just proceed to load in the second program or you will end up chaining the first program back into memory.

The reason lines 110 and 160 of the first program had to be modified is that the computer prints 14 lines to the screen when chaining the second program. If this offset had not been accounted for, the second program would have read the wrong portion of the screen.

```
100 CALL CLEAR
110 DISPLAY AT(15,1):
    "TYPE SOMETHING."
120 ! ACCEPT AT statement is
130 ! needed to put text in
140 ! a predictable screen
150 ! location
160 ACCEPT AT(16,1):A$
170 ! All variables will be
180 ! cleared when the next
190 ! statement is executed
200 RUN "CS1"
```


YOUR SOFTWARE LIBRARY

BY Graham Hilton.

648/BOWLS. by W.A. McATH. TI BASIC.

Another game in console basic which simulates a game of Bowls and works rather well. The graphics are fairly simple and represent a bowling green. You have to input which bowl you want to use, the strength of the shot and angle etc. The computer then calculates where it will go to, and then sends it on its plotted course. The computer then plays against you. 1 or 2 players can play this game and there are instructions on screen.

650/GOLF. TI BASIC by Mike Curtis. Members may have seen the EX.BASIC PARCO GOLF game, well this version is written in console basic and works surprisingly well. There are on screen instructions which seem very comprehensive (I know nothing about golf). Good use is made of CALL KEY, so you do not have to ENTER numbers etc - just press the required key. When you get the ball onto the green near the hole the view changes to allow you to get the ball in the hole more easily. Quite a nice game for all you golf enthusiasts.

651/QBONO EXTENDED BASIC.

This is an EX.BASIC version of that classic game QBONO which has been seen on many home computers and some arcade machines. This version is keyboard only and is quite a gripping game. The object is to move your little 'character' up and down a pyramid shape of steps to change their colour while avoiding various obstacles which appear, and also try to avoid falling off the edge. A well written game and a bargain at price.

652/KAMIKAZE EXTENDED BASIC.

This is another bargain from our library. It is a Space Invaders type game in which you have to shoot down wave after wave of aliens which drop diagonally across the screen. The game has 10 - yes TEN! levels and needs joysticks. This is another fine game for members who have EXTENDED BASIC.

653/HUNCH TI BASIC. by Andrew Tibbes

This is a game in which you are QUASIMODO of 'the bells' fame. This is written in console basic and features some very good background graphics of a castle wall with a night sky in the background. Your little man is on top of the castle wall and you have to move him using either keyboard or joysticks along the top from left to right, jumping over several parapets? using the space bar or fire button to jump. At the same time you have to avoid spears crashing down on you, also a guard who is climbing up the wall. When he gets to the top he fires a spear at you - very unhealthy. I found the keyboard to be the best way to play. The graphics were very well done. They reminded me of HUNCHBACK HAVOCK in EX.BASIC. Apart from one or two minor flaws such as not checking for upper or lower case inputs I found this a very nice TI BASIC game and not easy.

654/DEACTIVATE TI BASIC. by Edwin Armstrong.

A simple game in console basic where you have to move a tank around the screen using the arrow keys to deactivate randomly placed mines. You have the option of setting the time limit yourself or having a random timer. A novel feature at the end is the name and address of the author in case you wish to discuss anything about the game or 99/4A's in general.

Please remember to use the special order forms available from the Group library. Programs are only available to members these are not to be distributed commercially. (Backup copies allowed for your own use).

NATIONWIDE TI99/4A USERS SHOW

This time I haven't much spare time due to working seven days per week. Had to be reminded again by Clive of the imminent publication date. I will briefly give details of the show which has now definitely been fixed for Leeds. The site is in North Street, central Leeds. This building is known in Leeds as the Old Dispensary, for those who get lost.

I have been contacted by several people with offers of help at the show, and I am very grateful for the offers. The rough plan for the show is for the dealers to do their own thing, as usual, but for more systems to be available for people to use.

One idea that has been floated is to have the User Group library available for copying onto disk, or perhaps tape where appropriate. This could be extended to the public domain software held by Steven Shaw. The people who hold these libraries will no doubt make their usual charge (you don't get anything for nothing in this world), but please bring your own disks, or buy a pack at the show.

The next idea is that there will be several small demonstrations of control of external motors, lights or other devices. This is simply to show that the 4a can be used for things other than graphics displays. (The Texas range of microchips is used extensively in industrial control, as I have tried in the past to indicate). Do not expect walking robots or the like. These can be controlled by the 4a, it is not a monopoly of the BBC set, as some would have you believe, but I do not own a robot, and have no intention of making one.

I have been contacted by one person who is making his own expansion box, and he has offered to bring it to the show if it is finished in time. I will exhibit my prototype 32k ram expansion at the show, this time without the cover on, so that it can be inspected. I will ask Graham Wolstenholme of Sheffield if he will show his version of the "matchbox" 32k ram.

The Oxford group will no doubt be giving a display of the latest in technological miracles. I hear that they are now into modems in a big way, but I doubt that we will be able to have a telephone line with which we can demonstrate the usefulness of such devices.

I also believe that I have persuaded several friends in Leeds to demonstrate such things as Pascal, Pilot and Swedish Forth. We do not expect to give lectures, but will attempt to give advice where possible.

If anyone else has ideas, get in touch. If you have any hardware or software to show off, please bring it to the show, see me and we will try to find a place for it. I am the one with the black and white beard (I'm getting old now).

For my part, I am doing my best to put together some demonstration programs in the latest (I think) language on the 4a, small-C. If you have disk drives, you should seriously consider getting this language, as it is possibly the best thing since sliced bread for the 4a, including Forth, which I still like. The reason that it is so good is that it is compiled and FAST. The idea behind small-C is that it is compiled from the source code that you write with a word processor (TI-writer or E/A editor) into assembly language source code such as you would write directly with the E/A editor. You then assemble the source code with the normal assembler, and end up with the normal machine code. As a result, the programs run very fast, but not as fast as hand written assembly language programs. This is because the compiler has to be something of a compromise in the useage of registers, and does not take full advantage of the 9900 register set and instruction set.

The compiler has been written for a theoretical cpu with two 16-bit registers instead of the sixteen 16-bit registers as in the 9900. The compiler was I believe first implemented on an 8080 cpu under the CP/M operating system (at least that is where I first saw it), and was written in such a way that the section of the program which generates the assembly language source code can be altered to produce source code for any processor that you wish to use. In this way it is possible to "transport" the compiler to other processors in other computers. ("Transporting" is the latest jargon for the process of modifying programs to run on other machines). This is exactly what Clint Pulley has done, and I take my hat off to him for the work that he has put in on this first version that I have seen. It is crude compared to those that you can buy for more popular machines, but is a basis on which we can build better things. I had considered this last year, and voiced my ideas to members of the user group who met at the London PCW show. The problem I saw was that the compiler normally requires 40k in which to run, but Clint Pulley has overcome this limitation, and has made a working version.

The language itself is very powerful, and is the language used on many commercial machines, both microcomputers and minicomputers, especially Unix-based machines. There is scope therefore for people to learn the rudiments of this language on our humble 4a, and use this as a stepping stone to jobs on computers in the commercial sphere. No longer is the 4a just a toy to be used at home, but can be a stepping stone to greater things, as well as a useful machine at home. This is more than can be said for Basic, no matter how good the version of Basic. Can anyone tell me of any applications where Basic is used for LARGE commercial applications? To make it quite clear of the possibilities of C, there are machines such as the DEC PDP-11 where the operating system is written in C, as well as the applications programs. There is now a chance to make the your computer live up to the advertising hype that used to surround the 4a and all other home computers, in that it can be of great value to children who are looking towards a career in computing. (After all, isn't that the reason that many people gave for buying a computer in the first place?).

Other news on the computer front is that I have received from our friends in Sweden the source code for the Swedish version of Forth. Should anyone be interested in having a copy, please see me at the show. If anyone is interested in the Swedish Forth, or in Pascal, and would like to correspond with enthusiasts in Sweden (very nice people) please contact me at the show and I will arrange for addresses to be passed on.

Must go now and attend to other business, but DON'T FORGET to support your User Group meeting, and keep things alive.

HOWARD WRITES AGAIN . . . yawn!

Frankie may well have gone to Hollywood, but yours truly went to New Jersey. It's certainly not as picturesque, the view as the aircraft landed was like something from a propaganda film about how American industry can work. Miles and miles of industrial complexes. Dirty, but a tribute to the way the United States has managed to keep heavy industry going, whilst ours is in tatters. Mind you, the view wasn't really uppermost in my mind. What was really bugging me was that I was on board an aircraft built by McDonnell-Douglas! Not only that, but in order to get back to Washington that evening, I'd have to use one again. I sweated pounds off!

I was met at the airport by Pat, Lou Phillips charming and competent assistant and driven to Basking Ridge, about 3/4 of an hours journey from Newark airport. Amongst others hanging around in Basking Ridge are A.T. & T. who are to the U.S.A. what British Telecom ought to be here. Very big and very efficient. I mention this 'cos there's a fair amount of brainpower at A.T. & T. which can't be used officially by Myarc, but since over lunch in a local restaurant, just about everyone else in the building was a company man, ideas can be swapped and expounded on.

After lunch, I was given the tour by Lou. He works from home, but that's not the same as saying I work from home. (You might as well claim Her Majesty the Queen works from home!). Nor is it to say that Myarc are a one man band. A great deal of work is undertaken by freelancers. This makes for an efficient way of getting work done. A freelance doesn't get anything if the job isn't done. Properly. Two upstairs bedrooms are devoted to Lou's and Pats offices. It was in Lou's that I noticed the first item of interest. A TI99/8. It had its cover off, but was working. The amusing thing about it is that the title screen is identical to the TI99/4A. Only when you jump to the start menu do you realise that this is a very different beast. In view of the delays in getting the new computer ready for sale, I can't help but think that it might have been an idea to produce the 99/8. Since the new computer will itself be based the 99/8, it would have given good grounding. Tell us about the new computer!

Onto the R&D centre. Lou's basement. This is very spacious, about 2000 sq. feet and is better equipped than some professional workshops I've been in here. There, I was treated to a demonstration of the Level II Extended Basic. Now this can only be used by 128k/512k card owners, but its relevance shouldn't be overlooked. This is because the ExBas that Myarc have developed is going to be the language used on the new computer. Very impressive it was too. All the newer trendy commands that have become par for the course in Basic are there. CIRCLE, PLOT, DRAW, FILL etc. Also included is a MARGIN command. What this does is allow one program to be running in the main screen while another can be dragged into a corner for temporary work. It gives results similar to windows. Something else was explained to me, although I didn't appreciate the full significance. It is possible to retain variables after a program has been quit. This means that a second program can continue working on the data produced by the first. There are lots of other features in the language, the two most impressive being related. These are compactness. Because of the new commands, formerly very long programs can be compacted considerably. The other is the result of the first. Without so much work to chew through, there is a marked increase in speed. Remember this is only on the TI99/4A. When these commands are implemented on the new computer, it will be very fast even in Basic.

What else do Myarc have under development? Why don't you tell us about the new computer? Well one product is a hard disc control card. Hard Drives are what put Myarc's name on the map. They even were subcontractors to TI who sold them in South America, under the TI badge. Apart from experience, this also meant that Myarc had to work to high quality standards. Nothing with the TI badge on is made badly. Quality inspectors from Texas would periodically inspect Myarc's work to ensure it came up to their own standards. The hard disc card under development is Myarc's third attempt at this topic. First came the original hard drive. A bulky box containing drive and all the controllers. Then came the second phase. A hard disc personality card, which in turn connected to a WDS controller card, with that in turn going to an off the shelf hard drive.

Howard Greenberg yawn!

Now comes stage 3. Build the personality card and the WDS card into one unit to go in the Peripheral box. This now means that a standard IBM hard drive can now be attached. The idea isn't only a saving in space, it brings the cost down to £350.00 for the card instead of the £500.00 required for the previous two. Still, it is expensive, and there's a hard drive to be bought yet.

Tell us about the new computer. Of more interest to U.K. users is an 80 column card being developed. Now Foundation (who are now bankrupt) did make a 80 column card, but theirs didn't have any software. This one will have a TI-Writer built in, working in 80 columns. Although of less interest to U.K. users, there will also be a patch in the cards software for Multiplan.

Incidentally, Myarc also have under development a new computer. The case that as featured on the cover of the last TI*MES was just that. A case. Very well made, but quite empty. It will be what holds the new machine though. The keyboard is a pleasure to use. All the keys are just where they should be, and the feel is very positive. The technical spec is much as in the last TI*MES so I won't dwell on that. The computer has the following ports though. A bus to connect to the outside (and presumably use TI and Myarc peripheral systems). A joystick socket, sockets for RGB and composite video (no modulator is planned), power in, cassette port and cartridge port. At last we'll have a machine with its own printer ports. Both RS232 and Parallel are on board. I was though mildly annoyed to discover that disc control interface will be by an external cartridge. This will plug in the back and have a bus to take it to a standard disc drive. Of a great deal more interest is the other big socket at the back of the machine. This will take an MSDOS cartridge and will enable the machine to use most IBM software. That should take care of any problems with lack of software! It looks to be a very capable machine, and provided it gets to the market place, will be a success.

Which brings me onto my next point. It's very difficult to be too overbearing on your host when you're miles from the airport and no idea as to how to get there, but as politely as I could, I pointed out to Lou that he wasn't exactly the most punctual person in the world. There are times when I've wondered if we use different calendars. The new computer has been promised at three monthly intervals for the past year. It may be ready this summer, but don't count on it. That's the worst example, but I've been kept waiting for the 128k cards and the disc controllers too. I will admit that once they're on line, they come with regularity, but this business of announcing a product so far in advance that it seems they're never going to come wins no friends. Lou had the grace to admit that at times his enthusiasm does make for this kind of situation, but it's not deliberate. When you're in the middle of that kind of problem, the easy way out is to take on extra people to get the job done, but strangers, no matter how technically competent can't just pick up in a creative design and expect to achieve the same goals as the original designer.

More Myarc news and this time a problem. Would all those owners with Myarc RS232 cards please pay attention. At the back of the card, (where the sockets are) is a strip of PCB which can have the tracks grounded giving problems with printout. What's happening is that the spring clips in the box are scraping away the solder resist mask and touching the tracks. If you've experienced this problem, you'll know how frustrating it can be. To correct it (or prevent it better still), wrap the exposed strip of card (where the spring clips touch, with insulating tape. That's all that's required.

Despite looking promising last time I wrote, I still haven't closed the deal with Thorn-EMI. I hope to have disc copies of the three games available at Leeds for sale.

After two years plus, I've parted company with my Shinwa CP80. The printer served me faithfully for all that time receiving a great deal of useage than a normal domestic printer was expected to suffer. Towards the end though, I did notice a degradation in print quality. This was corrected shortly before I sold it, by replacing the print head. This cost about £38.00 and I didn't begrudge the cost one bit. If that's all you've got to spend on a printer to make it as good as new after that much work, it's cheap. I now have a CPA 80+ and am delighted with it. Changing the head on a printer is a simple task, apart from removing and inserting the ribbon that goes to the mechanism. Mine

as very firmly wedged and it took a great deal of effort to remove the old one and insert the new one. I was quite worried that I'd tear the ribbon, making the new head useless, but fortunately, all turned out well the second time. The first time I had a shock. Although the new head worked, it was printing blocks. What had happened was that the new heads ribbon was one track slimmer than the older one. This meant that I'd inserted it one track out. Re-doing it achieved the desired results. Since there are some owners out there who's printers must by now be displaying similar symptoms, watch out for that particular problem.

My personal thanks to Peter Walker who wrote the article on file handling in the last edition. Keep writing! File handling is a tricky subject to me and my eyes tend to gloss over when reading the TI manual. This was the first I'd seen on the subject in English. Keep it up!

Craig Miller is being a busy boy. By the time this is published, I should have a second batch of Gramcrackers, hopefully in time for the next show. I should also have a set of his Newsletters. I mention this because his is the finest around. I asked if I could pay a special price for them with authority to copy. This was agreed, and so I'm now able to pass on a full set of the newsletters at £5.00 per set. Some of this is news, which by now is not news. But in there also is a whole heap of useful information. Something else being worked on is a disassembler. Now I know that disassemblers have been around for some time, but anything that comes from Mr. Millers stable tends to be a bit different. (i.e. better).

Footnote to the above piece. I have now received and sold the first gramcrackers and will try to obtain more in time for the Leeds show. I'm told a review of it appears elsewhere in this edition so I won't go into it too thoroughly, but let the independant reviewer have his/her say. What I will say though is that it appears to be the most impressive item seen since Myarc's Ram-Disc (possibly more so), but even more has QUALITY stamped all over it. As long as the exchange rate holds up, I'm selling it for £175.00 and at that price, I would consider it excellent value for money.

CLOSE ARTICLE

This column has over the last year or so, changed from being the Howard Greenberg column to the Arcade Hardware column. The reasons for this are twofold. Firstly, I just don't discover that much interesting news in terms of learning to do something new or re-discovering the wheel, so in order to pad out what I write to fill the page, I have little choice but to include descriptions of new products and re-evaluate older ones. Secondly, it's what those of you who take the trouble to ask me seem to want!

CLOSE ARTICLE part 2.

How are you supposed to have two close articles. It's a contradiction in terms! The reason for this is that I noticed that as the paper finished coming out of the printer, I had some space left and as the article was being printed, I had a glitch.

If you've ever experienced your printer either totally refusing to work despite having been connected correctly and having worked previously, or even worse had it do strange things such as printing in this manner: Theeeeeeee quicccccck ron fffffffox, you'll quite rightly be puzzled. The solution is quite simple. What has happened is that the connector to the printer is loose, so it's either not making at all, or worse, is rattling. Tighten it up and all will be well!!!!!!

BYE
HOWARD

Howard

DIY 32K & CARTRIDGE EXPANDER

1. 32K RAM Project

There was unfortunately an error in part of the veroboard circuit for the 32K RAM. Shown below is the correct wiring sequence for the GND wire. Please note the additional break in the copper strip between GND in and PIN 15.

Hopefully anybody who attempted the project will have noticed the error, when comparing the circuit with the Australian Article. I hope this did not cause problems for anybody.



2. Cartridge Expander and Reset S/W

In the last issue of TIMES there was an article describing how to make a cartridge expander.

I have to date made two (we have two consoles - one for the kids) expanders. One based on the above article but is a direct plug in unit, similar to the WIDGET.

The other unit utilises a Munchman Module, ie all components removed from PCB and connections made between each line and a suitable socket, mounted on top of the module case. In addition a reset switch, push to make, was placed in the line from pin 1 and mounted on the front sloping edge.

Although this unit is only a single part it is ideal for the kids and saves wear and tear on the console, ie they no longer have to switch off and on due to lockouts, etc. This Unit only resets with module inserted.

All parts, except for the module, for either unit, can be purchased from any good electronics supplier. However, you may have to purchase a multi socket with 40 contacts and cut it down to size. (Note:- Inspect the socket before purchase - some have contacts which are not positioned directly opposing. These require more precise alignment of the modules and can cause contact problems).

Whichever unit you make don't forget to clean the console socket before inserting the expander otherwise you may still have lockout problems!

Finally you may have noticed that after inserting a module in the console the socket and its PCB can still be moved - look through the top grill and see! This of course can also lead to bad contacts - solution dismantle console and using PVC tape, tape the socket to its supporting bracket. It may be crude but it works!

3. Logo

Did you know that to obtain LOGO for the BBC with a similar specification to the TI, ie Sprites, it will cost you £250. So come on you LOGO owners let's have some articles.

Byte for now
Graham Wolstenholme
200 Hollinsend Road
Sheffield
S12 2EJ

TEL:- 655582

So much for our game program. Suppose we want to store large quantities of data on cassettes? Well given the very slow speed of data storage my advice is:

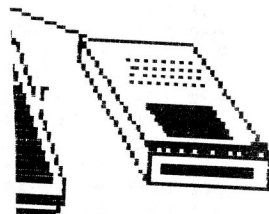
1 Forget the normal rules of Records with subdivided items. Try to fill the maximum 192 byte records as much as possible. For example a 192 byte record can store 21 numeric items.

2 Avoid long strings if you can.

3 Alternatively if you want to create a database program, use the Personal Record Keeping module. This stores its data much faster than BASIC by using PROGRAM format.

4 Use Enhanced Basic. This is the Basic resident in the PRK and Stats modules which features some extra calls (CALL P,L,S,G,H,A,D). These allow you to create and store PRK compatible data files in program format. There have been good explanations of Enhanced Basic in a number of journals and I believe Stephen Shaw can supply the definitive description of how it all works. Therefore I will not try to cover this subject here except to show you how you could store 50 Hiscores and Names using this method.

```
50 DIM HISCORE(50),NAME$(50)
100 CALL L("CS1",C)
110 IF C<>1 THEN 500
120 FOR I=1 TO 50
130 CALL G(1,I,1,M,HISCORE(I))
140 CALL G(1,I,2,M,NAME$(I))
150 NEXT I
160 REM GAME PROGRAM FOLLOWS
170 FOR I=1 TO 50
180 PRINT HISCORE(I);NAME$(I)
190 NEXT I
200 GOTO 1000
500 CALL CLEAR
510 CALL D(6,1,22,"WAS CALL P(1500) USED?")
520 CALL D(8,1,11,"ENTER Y/N Y")
530 CALL A(8,11,1,F,AN$)
540 IF AN$<>"Y" THEN 600
550 CALL D(10,1,13,"RETRY LOADING")
560 GOTO 100
600 CALL D(12,1,21,"USE CALL P(1500) THEN")
610 CALL D(13,1,14,"RELOAD PROGRAM")
620 STOP
1000 CALL H(0,1,1,"HISCORES")
1010 CALL H(0,5,2,2)
1020 CALL H(0,10,1,2)
1030 CALL H(0,10,2,1)
1040 CALL H(0,11,1,5)
1045 CALL H(0,11,2,15)
1050 FOR I=1 TO 50
1060 CALL G(0,I,1,HISCORE(I))
1070 CALL G(0,I,2,NAME$(I))
1080 NEXT I
1090 CALL S("CS1",C)
1100 REM STORING COMPLETE
1110 STOP
```



Before loading or running this program you must enter the command CALL P(1500) to partition the memory for the PRK data. If you don't the PRK data will not load and an error will immediately be issued. Perhaps one day I'll write the definitive guide to Enhanced Basic. In the meantime...

INTERLUDE

It was at this point in writing this article on the TI-Writer that editing came to a halt with the dreaded message "MEMORY FULL Save or Purge". So I immediately saved the file to disk only for the system to crash. Not only did I lose some text completely but also the file header was erased with the sole exception of the Filename. Yes I know one should try to keep back up copies but we're all human. So I had to sit down with a Sector Editor and search all 358 sectors for what looked like bits of the article and build up the header from scratch. Rebuilding the Block-link list (Nybble by Nybble) is about the worst task the 99/4a ever forces on you, especially when your file is fractured into 8 pieces. There is a moral here somewhere.

Apologies to most of you. I know that few of you will completely appreciate the above. Be warned! There is a lot to the disk system that doesn't immediately show itself. However this is all quite out of place in what is supposed to be a gentle introduction to data files. To continue....

LENGTH OF DATA FILES

A common problem when programming with files is knowing how many records exist on a file. This is useful to know since a simple loop or FOR-TO-NEXT loop is usually used to read in the records and it is clearly important to know when to jump out of the loop! I recommend one of the following three techniques to solve the problem:

1 If you have a disk system then the EOF function, used before each INPUT statement, allows you to find when you have reached the end of the file. EOF cannot be used with cassettes.

2 You can use a dummy record as an end of file marker. For example in a string record you could print the item "ZZZZ" to indicate the end or in a numeric record the item 99999. Since both are unlikely to occur normally you can test for their presence as each record is input and be confident that they indicate the end of file.

3 Since the program generating the file usually knows how many records have been printed onto the file, you can print the number of records into record 0 and only use records 1 onwards for data. In this way the first record read indicates how many more follow which can therefore be used to set a FOR-TO-NEXT loop to read in the exact number of records.

Lets look at the three methods:

METHOD 1

```
100 OPEN #1:"DSK1.FILE",INTERNAL, FIXED 40
110 IF EOF(1) THEN 1000
120 INPUT #1:A$(I)
130 I=I+1
140 GOTO 110
```

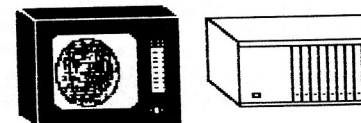
1000 CLOSE #1

1010 REM Rest of program...

METHOD 2

```
100 OPEN #1:"DSK1.FILE",INTERNAL, FIXED 40
110 INPUT #1:A$(I)
120 IF A$(I)="ZZZZ" THEN 1000
130 I=I+1
140 GOTO 110
```

then as method 1



METHOD 3

```

100 OPEN #1:"DSK1.FILE",INTERNAL,FIXED 40   130 INPUT #1:A$(I)
110 INPUT #1:N                               140 NEXT I
120 FOR I=1 TO N                             150 CLOSE #1

```

MAKING BEST USE OF FILE STORAGE SPACE

We have already seen above how cassette files can be used to maximum efficiency by cramming each maximum length 192 byte record with as much information as possible. Good file design also applies to disk files. Each disk sector stores 256 bytes but no record may overlap a sector. Thus one sector can only store one record if the record length is 200; the other 56 bytes are wasted. On the other hand a sector can hold two records of length 128 with no waste of space.

We have also seen that VARIABLE length files can save space. Another way to save space which is especially useful for storing integer numbers is to use DISPLAY rather than INTERNAL format. For example, the number 55 uses 9 bytes when stored in internal format but only 3 when using display (two character bytes plus the comma separator required by display format.) This saving is gained but with the added complication mentioned above of explicitly providing the comma separators between items not to mention quotemarks around any string items.

EXTERNALLY DEFINED DATA STRUCTURE

We have seen how we can use RELATIVE file organization with FIXED length files to provide random access to any record in a file. However the nature of the INPUT statement requires us to read each item in a record sequentially. This has some drawbacks, but there is a method of file design where the item structure is externally defined by your program. If we use DISPLAY/FIXED files we can not only access each record individually but also we can then define fixed length elements of each record for each item. We can read each record as a single item record (as far as the computer is concerned) and use the SEG\$ function to access each or any item within it. To create such records all items are handled as strings (thus gaining the space economies for integer storage referred to above) and concatenating (function) items together before printing to file. Such records contain no length bytes or item separators and this can be of use as well.

I think an example may show what I am driving at. Suppose we wish to store 100 Names (max length 10 characters) and ages (up to 99 years), and three other vital statistics (eg 36,24,36). The traditional way would be as follows:

```

100 OPEN #1:"DSK1.FILE", INTERNAL, FIXED 47 ! 47=11+4*9
110 FOR N=1 TO 100
120 PRINT #1:NAME$(N);AGE(N);B(N);W(N);H(N)
130 NEXT N
140 CLOSE #1

```

Each sector will store 5 records, so the file will need 21 sectors in total. (The directory is the 21st sector of course.)

By comparison look at the following:

```

100 DEF F(A$)=SEG$(A$&RPT$(" ",10),1,L) ! PADS OUT A$ TO CHAR LENGTH OF
L. 110 OPEN #1:"DSK1.FILE", DISPLAY, FIXED 18 ! 18=10+4*2
120 FOR N=1 TO 100
130 L=10
140 X$=F(NAME$(N))
150 L=2
160 X$=X$&F(STR$(AGE(N)))&F(STR$(B(N)))&F(STR$(W(N)))&F(STR$(H(N)))
170 PRINT #1:X$ 180 NEXT N
190 CLOSE #1

```

With the 18 byte records we get 14 records per sector and need 9 sectors overall, a saving of 11 sectors. Lets look at how this "compacted" file is read:

```

50 DIM STAT(100,4)
100 FOR N=1 TO 5
110 READ WIDTH(N)
120 DATA 10,2,2,2,2
130 NEXT N
140 OPEN #1:"DSK1.FILE", DISPLAY, FIXED 18
150 FOR N=1 TO 100
160 P=1
170 INPUT #1:X$
180 ITEM=1 190 NAME$(N)=SEG$(X$,P,WIDTH(ITEM))
200 P=P+WIDTH(ITEM)
210 FOR ITEM=2 TO 5 ! Numerics put in array for program efficiency
220 STAT(N,ITEM-2)=VAL(SEG$(X$,P,WIDTH(ITEM)))
230 P=P+WIDTH(ITEM)
240 NEXT ITEM
250 NEXT N
260 CLOSE #1

```

You may wish to know that the excellent (though hard to understand at first) Navarone Data Base Management System uses this form of file structure. This structure has one other advantage. If for example you had a name and address file, you would get a concatenated Surname and Forename field which can be used as a single element when sorting the file into alphabetic order. You can thus avoid the complication of nested sorts. This is made possible because the file will have no separators or length bytes between the two items.

IDIOT PROOFING

If your program allows the filename to be input during the program, it is wise to allow for a number of events that might crash the program or lose valuable data:

- 1 The device may be empty, write protected or not connected.
- 2 The file may already exist with different attributes.
- 3 The file may already exist with the same attributes.

To guard against these difficulties, you can use ON ERROR trapping for (1) and (2) above. By trapping I/O ERRORS and FILE ERRORS you can stop the program from aborting. The 3rd problem is not trapped by the operating system however. Of the standard TI modules only Multiplan asks you whether you explicitly want to overwrite an existing file. There are two ways to solve the problem in Basic. Firstly you could search the disk directory to look for the existence of the filename. This however is slow. A second and faster way is to OPEN the file and immediately test the value of the EOF() function. A newly created file will give an EOF value of 1, while an existing file, if it contains data, will give a value of 0. Thus you can trap the overwrite before using any PRINT statements.

FINALE

I hope this has been a useful and informative introduction to data files. Should any of the foregoing appear less than crystal clear then I would be happy to explain further by correspondence. You can contact me at:

24 Bacons Drive
Cuffley, Herts, EN6 4DU.

Happy Filing

Peter Walker

TIPS FROM TIGERCUB



Copyright 1985

TIGERCUB SOFTWARE
154 Collingwood Ave.
Columbus, OH 43213

Distributed by Tigercub Software to TI-99/4A Users Groups for promotional purposes and in exchange for their newsletters. May be reprinted by non-profit users groups, with credit to Tigercub Software.

TIJ
UDS:TDIFA
BJIF
JDIF
SURS
STSA
SFBF
TRA

The above is a long division problem in the proper format, with each numeral replaced by a letter. Can you solve it? My program TC-41 Long Division Cryptograms, will generate an infinite number of such puzzles for you, and help you to solve them - and it only costs \$3.00. It took me a week to program, and I've sold 12 copies in 2 years! Doesn't anyone like to exercise their brains anymore?

TIGERCUB CHALLENGE

```
100 FOR J=1 TO 7 : READ M$
: PRINT M$ : NEXT J
30000 DATA AAAAAAAAAAAAAAAA
AAAAAAAAAA,BBBBBBBBBBBBBB,BB
BBBBBBBBBBBB,CCCCCCCCCCCC,CC
DDDDDDDDDDDDDD
30010 DATA "TESTING",,,,,,
,,,,,,,,,,,,,,,,,"TEST
ING"
>RUN
AAAAAAAAAAAAAAAAAAAAAAAAAAAA
BBBBBBBBBBBBBB,BBBBBBBBBBBBBB
```

```
CCCCCCCCCCCCDDDDDDDDDDDDDD
"TESTING"
*****
"TESTING"
* READY *
Can you run this program and get these results? You won't even be able to key in that last DATA item! So, how was this programmed? No, there are no redefined characters!
Do you need something educational? Here is a little routine to give the plural endings for most words. I will leave it to you to develop further - and see if you can teach the computer the plurals of PANTS, TOOTH, MAN, FUNGUS, DATA and the other inconsistencies of the English language.
100 REM PLURAL ENDINGS
: BY Jim Peterson
110 INPUT M$
120 Z$=SEG$(M$,LEN(M$),1)
130 Y$=SEG$(M$,LEN(M$)-1,2)
140 ON POS("EFHSXYZ",Z$,1)+1
: GOTO 270,150,190,180,250,25
0,220,250
150 IF SEG$(M$,LEN(M$)-2,2)
: >"IF" THEN 270
160 PL$=SEG$(M$,1,LEN(M$)-2)
: &"YES"
170 GOTO 280
180 IF (Y$="CH")+(Y$="SH")TH
: EN 250 ELSE 270
190 IF (Y$<>"L")*(Y$<>"R")
: *(Y$<>"AF")*(M$<>"HOOF")THEN
: 270
200 PL$=SEG$(M$,1,LEN(M$)-1)
: &"YES"
210 GOTO 280
220 IF (Y$="AY")+(Y$="EY")+
: Y$="OY")+(Y$="UY")THEN 270
230 PL$=SEG$(M$,1,LEN(M$)-1)
: &"IES"
240 GOTO 280
250 PL$=M$&"ES"
260 GOTO 280
270 PL$=M$&"S"
```

```
280 PRINT PL$
290 GOTO 110
If you want to turn that into a quiz, change line 110 to READ W$, change line 280 to PRINT W$." PLURAL?" : Add lines -
281 INPUT Q$
282 IF Q$<>"PL" THEN 295
283 PRINT : "RIGHT!" :
284 GOTO 110
285 PRINT : "WRONG! PLURAL
OF "W$;" IS "PL$:"
300 DATA BOY,WATCH,WIFE,BOY
(And as such more as you want)
Just one more optional refinement to my Menu Loader. If you want to use a filename ending in an asterisk for those Basic programs which will not run in XBasic, this change will keep you from loading and crashing them.
420 CLOSE #1 : IF SEG$(PG$(K),LEN(PG$(K)),1)="" THEN D
: ISPLAY AT(12,1)ERASE ALL:"RE
TURN TO BASIC AND LOAD BY:"
TYPING OLD DSK1."*PG$(K):" S
: SX$(15)
TOP
The idea of a program that writes a program has stirred up a little interest, so here's another. This routine will aid you in formatting your screen text into neat 28-column lines, and will save the text in program lines of DATA statements. When you are ready to save, type @@@ and enter as the last line, then NEW and MERGE DSK1.LINEFILE
100 !LINEWRITER
: - by Jim Peterson
130 CALL CLEAR : OPEN #1:"D
SK1.LINEFILE",VARIABLE 163
: LN=30000
140 FOR R=1 TO 24 : DISPLAY
: AT(R,1)SIZE(11) : ACCEP
: T AT(R,0)SIZE(1-28):A$ : IF
: A$="@@@" THEN 180 : B$=B&C
: HR$(200)&CHR$(LEN(A$))&A$
```

```
150 X=X+1 : IF X/4=INT(X/4)
: THEN 160 ELSE B$=B&CHR$(179)
: GOTO 170
160 GOSUB 210 : LN=LN+10
170 NEXT R : X=0 : CALL CL
: EAR : GOTO 140
180 IF B$="" THEN 200 : IF
: SEG$(B$,LEN(B$),1)=CHR$(179)
: THEN B$=SEG$(B$,1,LEN(B$)-1)
190 GOSUB 210
200 PRINT #1:CHR$(255)&CHR$(
: 255) : CLOSE #1 : END
210 PRINT #1:CHR$(INT(LN/256)
: )&CHR$(LN-256*INT(LN/256))&
: CHR$(147)&B&CHR$(0) : B$=NU
: L$ : RETURN
Here's something for "JET" and Danny and Gene and all the rest of my friends in Alabama - and in all the rest of Dixie. You've never seen fireworks quite like these before!
100 CALL CLEAR : PRINT TAB(
: 5) "ALABAMA 4th of JULY" :
: : : : "programmed by
Jim Peterson" : FOR D=1 TO
: 200
110 NEXT D : RANDOMIZE
120 DIM S$(12),A$(16),S(16),
: S$(15)
130 DATA 196,220,247,262,294
: ,330,349,392,440,494,523,587
: ,659
140 FOR J=4 TO 16 : READ S(
: J) : NEXT J : FOR SET=2 TO
: 14 : CALL COLOR(SET,1,1) :
: NEXT SET : CALL SCREEN(2)
150 DATA 00,18,24,3C,42,5A,6
: 7,7E,81,99,AA,BD,C3,DB,E7,FF
160 FOR J=1 TO 16 : READ A$
: (J) : NEXT J
170 FOR CH=0 TO 136 STEP 8
: : FOR L=1 TO 4 : X=INT(16*
: RND+1) : B$=B&A$(X) : C$=A$
: (X)&C$ : NEXT L
180 SX$(CH/8-4)=B&C$ : CAL
: L CHAR(CH,B&C$) : GOSUB 350
: : B$=NULL$ : C$=NULL$ : NE
: XT CH
190 FOR J=1 TO 12 : FOR L=1
: TO 6 : X$=CHR$(INT(13*RND+
: 5)*8) : B$=B&X$ : C$=X$
: &X$&C$ : NEXT L
200 S$(J)=B&C$ : B$=NULL$ :
: C$=NULL$ : NEXT J : CALL
```

```
MAGNIFY(2) : 350 CALL MAGNIFY(1) : CALL S
: N 110 and add a line 125 X3=
: 210 FOR J=1 TO 12 : DISPLAY
: PRITE(#CH/8-4,CH,13*RND+3,20
: AT(J,3):S$(J)) : NEXT J : X
: =0,128-30,RND*20-RND*20) : R
: ETURN
: For a longer series
: without repeating, it might
: be better to use this
: method.
: The Home Computer Magazine,
: Vol. 4 No. 3, had a
: program called Elementary
: Addition and Subtraction,
: which generates random
: numbers between 1 and 5 for
: elementary math practice.
: The first time I tried
: it, it asked me for the
: answer to 1 + 1. When I
: answered correctly, it
: produced another random
: problem - 1 + 1 again!
: This is known as the
: idiotic computer syndrome,
: and it helps us to remember
: that our computers are still
: no smarter than their
: programmers!
: Fortunately, this bit
: of idiocy is easy to cure.
: Try this -
: 110 X=INT(5*RND+1)
: 120 IF X=2 THEN 110
: 130 X2=X
: 140 PRINT X;
: 150 GOTO 110
Do you see how it
: works? The first time you
: get a number, X2 will equal
: 0 because it has never been
: given a value. X will be
: selected as a number between
: 1 and 5. Let's suppose it
: is 2. Line 120 compares it
: with X2; 2 is not equal to
: 0, so the program continues
: to line 130, where X2 now
: picks up the value of 2,
: then on to print the value,
: and back to 110. Now,
: suppose that the random
: factor in line 110 picks 2
: again. Line 120 finds that
: 2=2, X=2, and sends the
: program back to 110 to pick
: a different number.
: If you want to avoid a
: give you a tickler file, or
: repeat until after two
: times, change line 120 to
: read 120 IF (X=X2)+(X=X3)THE
: rather complex disk filing
: programs written for this
: purpose. Just use the month
: number (1-12) and date
: (always in two digits,
: 01-31) for the line number -
: 1000 !buy birthday pres
: sent for wife!
: 1009 !wife's birthday!
: 1010 !apologize to wife for
: forgetting birthday
: You can schedule
: several things in one
: program line -
: 1011 !get haircut/change oil
: in car/pinch secretary.....
: - but it might be better
: to add an extra digit (0-9)
: to the line number and
: schedule separately -
: 10110 !get haircut
: 10111 !change oil
: Then, if something
: doesn't get done, just use
: the REDD key to change the
: line number and reschedule
: it for another date. You
: can print out a list of the
: day's chores by simply LIST
: "PI0":7010-7019 (did you
: know you could do that?)
: Better clear up a few
: misunderstandings. The
: "freeware" offers I have
: mentioned in past Tips are
: NOT available from me - send
: your disk and returnabl
: mailer AND RETURN POSTAGE t
: the author of the program.
: And, my copyrighte
: Tigercub Software program
: are NOT freeware. They ca
: only be legally obtained b
: mail order from me - if yo
: copy them from anyone else
: you are stealing!
: As for the program
: which I write and publish o
: distribute without
: copyright, they are also n
: Freeware, they are FREE.
: don't want to be paid fo
: them, and I don't thi
: anyone else should be pa
: for them.
```

```
350 CALL MAGNIFY(1) : CALL S
: N 110 and add a line 125 X3=
: 210 FOR J=1 TO 12 : DISPLAY
: PRITE(#CH/8-4,CH,13*RND+3,20
: AT(J,3):S$(J)) : NEXT J : X
: =0,128-30,RND*20-RND*20) : R
: ETURN
: For a longer series
: without repeating, it might
: be better to use this
: method.
: The Home Computer Magazine,
: Vol. 4 No. 3, had a
: program called Elementary
: Addition and Subtraction,
: which generates random
: numbers between 1 and 5 for
: elementary math practice.
: The first time I tried
: it, it asked me for the
: answer to 1 + 1. When I
: answered correctly, it
: produced another random
: problem - 1 + 1 again!
: This is known as the
: idiotic computer syndrome,
: and it helps us to remember
: that our computers are still
: no smarter than their
: programmers!
: Fortunately, this bit
: of idiocy is easy to cure.
: Try this -
: 110 X=INT(5*RND+1)
: 120 IF X=2 THEN 110
: 130 X2=X
: 140 PRINT X;
: 150 GOTO 110
Do you see how it
: works? The first time you
: get a number, X2 will equal
: 0 because it has never been
: given a value. X will be
: selected as a number between
: 1 and 5. Let's suppose it
: is 2. Line 120 compares it
: with X2; 2 is not equal to
: 0, so the program continues
: to line 130, where X2 now
: picks up the value of 2,
: then on to print the value,
: and back to 110. Now,
: suppose that the random
: factor in line 110 picks 2
: again. Line 120 finds that
: 2=2, X=2, and sends the
: program back to 110 to pick
: a different number.
: If you want to avoid a
: give you a tickler file, or
: repeat until after two
: times, change line 120 to
: read 120 IF (X=X2)+(X=X3)THE
: rather complex disk filing
: programs written for this
: purpose. Just use the month
: number (1-12) and date
: (always in two digits,
: 01-31) for the line number -
: 1000 !buy birthday pres
: sent for wife!
: 1009 !wife's birthday!
: 1010 !apologize to wife for
: forgetting birthday
: You can schedule
: several things in one
: program line -
: 1011 !get haircut/change oil
: in car/pinch secretary.....
: - but it might be better
: to add an extra digit (0-9)
: to the line number and
: schedule separately -
: 10110 !get haircut
: 10111 !change oil
: Then, if something
: doesn't get done, just use
: the REDD key to change the
: line number and reschedule
: it for another date. You
: can print out a list of the
: day's chores by simply LIST
: "PI0":7010-7019 (did you
: know you could do that?)
: Better clear up a few
: misunderstandings. The
: "freeware" offers I have
: mentioned in past Tips are
: NOT available from me - send
: your disk and returnabl
: mailer AND RETURN POSTAGE t
: the author of the program.
: And, my copyrighte
: Tigercub Software program
: are NOT freeware. They ca
: only be legally obtained b
: mail order from me - if yo
: copy them from anyone else
: you are stealing!
: As for the program
: which I write and publish o
: distribute without
: copyright, they are also n
: Freeware, they are FREE.
: don't want to be paid fo
: them, and I don't thi
: anyone else should be pa
: for them.
```


Over 130 original programs in Basic and Extended Basic, available on cassette or disk, only \$3.99 each plus \$1.50 per order for PPH. Entertainment, education, programmer's utilities. Descriptive catalog \$1.99, deductible from your first order.

Tips from The Tigercub, a full disk containing the complete contents of this newsletter Nos. 1 through 14, 50 original programs and files, just \$15 postpaid.

Tips from The Tigercub Vol. 2, another diskfull, complete contents of Nos. 15 through 24, over 60 files and programs, also just \$15 postpaid. Or, both for \$27 postpaid.

Some users groups are putting my copyrighted programs, and those of other programmers, in their software library, "for use but not copying" or "for review and evaluation only". Who do you think you're kidding? I know I won't sell any software to members of pirate clubs, so why should I support them?

If you didn't solve the Long Division Puzzle why not then try dividing 230709 by 835. As for the solution to the Tigercub Challenge, it was right on the same page! Try creating those DATA statements with the LINewriter routine. I don't know why it works, but it does.

I've been asked to print more information on the "program that writes a program". I don't have room for a detailed account, but here are the basics. If you tried my TOKENLIST routine TIMES Noll you already have a list of the token codes

you will need. I won't go into the way that the computer squishes a program line number into only two characters, but you can accomplish it with DEF L\$=CHR\$(INT(LN/256))&CHR\$(LN-256*INT(LN/256)), where LN has been predefined as the value of the line number. If you need to refer to a program line in a statement, as in GOTO 500, use DEF R\$=CHR\$(201)&CHR\$(INT(RN/256))&CHR\$(RN-256*INT(RN/256)), RN being the line number. To print a statement or command, simply print its token character. For instance, the token for DATA is 147, so you would print CHR\$(147). Note that all the punctuation marks used in programming, such as (and +, are also represented by token codes which are NOT the same as their keyboard ASCII value.

To print a variable name, either numeric or string, just enclose it in quotes, "A" or "A\$".

To print a value, or an unquoted string (as in a DATA statement), or the word which follows a CALL, you must print CHR\$(200) followed by a token giving the number of characters to follow, such as CHR\$(5) for a 5-character word such as CLEAR, then the value in quotes. For instance, the token for CALL is 157, so CALL CLEAR is CHR\$(157)&CHR\$(200)&CHR\$(5)&"CLEAR". You can simplify that by predefining DEF U\$(V\$)=CHR\$(200)&CHR\$(LEN(V\$))&V\$, and then simply print CHR\$(157)&U\$("CLEAR").

A quoted string is handled in the same way except that it is preceded by token 199 instead of 200, so you can predefine it as DEF Q\$(V\$)=CHR\$(199)&CHR\$(LE

N(V\$))&V\$ - the computer will take care of the quote marks.

Each program line must end with CHR\$(0), and the last record you print must be CHR\$(255)&CHR\$(255). A MERGE format file is D/V 163, so open the file with OPEN #1:"DSK1.MERGEFILE", VARIABLE 163.

Don't print more than 163 characters in a record or the computer will blow its mind! You can print multiple-statement YBasic lines, but be sure to use the double-colon token CHR\$(130) as the separator, not two of the CHR\$(181) colon tokens.

Any errors you make will usually not show up until you try to MERGE or use the program you have created. I/O ERROR 25 means that you forgot the final 255 & 255; DATA ERROR or SYNTAX ERROR probably means that you left off a CHR\$(0) or gave the wrong count of characters after CHR\$(200).

Here's a bit of psychedelic blues - -

100 REM - FRANKIE & JOHNNIE by Jim Peterson

110 DIM S(12)

120 CALL SCREEN(2)

130 FOR R=1 TO 12

140 CALL COLOR(R+1,1,1)

150 FOR T=R TO 25-R

160 CALL HCHAR(T,R,32+R*8,34-2*R)

170 NEXT T

180 NEXT R

190 DATA 262,294,311,330,349,392,440,494,523,587,40000

200 FOR N=1 TO 11

210 READ S(N)

220 NEXT N

230 FOR J=1 TO 110 STEP 2

240 CALL COLOR(A+1,1,1)

250 READ T,A

260 CALL COLOR(A+1,A+2,A+2)

270 FOR TT=1 TO T

280 CALL SOUND(-999,S(A),0)

290 NEXT TT

300 NEXT J

310 RESTORE 330

320 GOTO 230

330 DATA 2,1,2,2,2,4,2,7,1,1,1,1,7,2,6,4,4,4,2,1,1,11,13,1

340 DATA 2,1,2,2,2,4,2,7,1,1,1,1,7,2,6,4,4,12,1

350 DATA 1,11,3,1,2,5,2,6,2,7,2,9,1,11,1,9,2,10,4,7,1,9,1,11,7,9

360 DATA 4,7,2,8,2,9,1,11,3,9,1,11,1,9,4,8,2,7,6,6

370 DATA 4,4,1,11,3,4,4,3,16,2,1,11,4,7,2,6,4,7,4,6,20,1,8,11

You can too have a blank space in your disk filenames! Just use FCIN V for the blank, instead of the space bar. You can even have a diskfull of 10 programs with invisible filenames consisting of 1 to 10 of those FCIN V's.

However, those invisible characters can do strange things when you list your disk catalog to a printer.

If you want to INPUT a string with leading and/or trailing blanks, just enclose the whole works in quotation marks. Try this -

100 INPUT A\$!type TEST

110 PRINT A\$:LEN(A\$)

120 INPUT A\$!type " TEST "

130 PRINT A\$:LEN(A\$)

140 GOTO 100 !you can even input a blank string of 136 characters

I really shouldn't tell you this, but if you want to make it difficult for someone to LIST your program, just insert a garbage line, every 5th line or so until you run out of memory, consisting of REM followed by 4 or 5 lines of random characters typed with the CTRL key held down.

Here's a program that

can actually read your mind! r a sucker....":...right?":

100 CALL CLEAR

110 PRINT "TIGERCUB MIND READER PROGRAM":

120 PRINT "I'll bet you do

130 GOSUB 440

140 PRINT "And I'll bet another dollar I can tell if whether you are thinking is correct.":

150 GOSUB 440

160 PRINT "And I'll bet another dollar I'm right BOTH times.":

170 GOSUB 440

180 PRINT "And I'll bet one more dollar I can guess what you'll be thinking a minute from now.":

190 GOSUB 440

200 PRINT "OK...":

210 GOSUB 400

220 PRINT "You're thinking that a computer can't possibly know what you are thinking.....right?":

230 GOSUB 400

240 PRINT "So I told you what you were":

250 GOSUB 400

260 PRINT "You owe me a buck.":

270 GOSUB 400

280 PRINT "And you're absolutely right..I can't read your mind.":

290 GOSUB 400

300 PRINT "So I told you correctly that":

310 GOSUB 400

320 PRINT "You owe me another buck.":

330 GOSUB 400

340 PRINT "So I was right BOTH times...right?":

350 GOSUB 400

360 PRINT "That makes three bucks you owe me.":

370 GOSUB 400

380 PRINT "And now it's a minute later":

390 GOSUB 400

400 PRINT "...so you owe me four bucks.":

410 GOSUB 400

420 PRINT "NEVER NEVER bet against a computer!!"

430 END

440 PRINT "Want to bet? Type Y(YES)":

450 CALL KEY(3,K,ST)

460 IF (ST=0)+(K>89)THEN 450

470 RETURN

480 FOR D=1 TO 800

490 NEXT D

500 RETURN

Since the manual doesn't mention it, some folks don't know that you can use IMAGE and PRINT USING for output to the printer. Try this -

100 OPEN #1:"PIQ"

110 INPUT "NAME?":N\$

120 INPUT "AMOUNT?":A

130 PRINT #1,USING "#####.###":N\$,A

140 GOTO 110

Of course, you could also add a line -

105 IMAGE "#####.#####.###"

And change line 130 to

130 PRINT #1,USING 105:N\$,A

John Taylor has written the most complete and versatile SPRITE BUILDER utility program that I have ever seen. It has 22 different options available with a single key press, including rotation and animation. And along with it comes a diskfull of preprogrammed sprites designed by a professional artist. This is being distributed as Freeware. Send two single-sided or one double-sided disks to John Taylor, 2170 Estaline Drive, Florence AL 35630, in a

returnable mailer WITH RETURN POSTAGE, at least - and I hope you'll also include something more!

Attention, programmers! Fred Hawkins of the Lehigh UG is trying to coordinate a project of documenting the operating system by breaking the console ROM down to pages of 256 bytes so that each individual or group can work on just one page. Only those who participate will share in the results! All this is far beyond me, but if you want in, send an SASE and a SSSD disk with return postage and mailer to Fred Hawkins, 1020 N 6th St, Allentown PA 18102 - soon!

Using for output to the printer. Try this -

100 OPEN #1:"PIQ"

110 INPUT "NAME?":N\$

120 INPUT "AMOUNT?":A

130 PRINT #1,USING "#####.###":N\$,A

140 GOTO 110

Of course, you could also add a line -

105 IMAGE "#####.#####.###"

And change line 130 to

130 PRINT #1,USING 105:N\$,A

John Taylor has written the most complete and versatile SPRITE BUILDER utility program that I have ever seen. It has 22 different options available with a single key press, including rotation and animation. And along with it comes a diskfull of preprogrammed sprites designed by a professional artist. This is being distributed as Freeware. Send two single-sided or one double-sided disks to John Taylor, 2170 Estaline Drive, Florence AL 35630, in a

innermost of several nested loops.

MEMORY FULL

Jim Peterson

Muts & Bolts (No. 1), a full disk of 100 Extended Basic utility subprograms in merge format, ready to merge into your own programs. Plus the Tigercub Menuloder, a tutorial on using subprograms, and 5 pages of documentation with an example of the use of each subprogram. All for just \$19.95 postpaid.

Muts & Bolts No. 2, another full disk of 100 utility subprograms in merge format, all new and fully compatible with the last, and with 10 pages of documentation and examples. Also \$19.95 postpaid, or both Muts Bolts disks for \$37 postpaid.

Tigercub Full Disk Collections, just \$12 postpaid! Each of these contains either 3 or 6 of my regular \$3 catalog programs, and the remaining disk space has been filled with some of the best public domain programs of the same category. I am NOT selling public domain programs - my own programs on these disks are greatly discounted from their usual price, and the public domain is a FREE bonus!

TIGERCUB'S BEST PROGRAMMING TUTOR PROGRAMMER'S UTILITIES BRAIN GAMES BRAIN TEASERS BRAIN BUSTERS! MANEUVERING GAMES ACTION GAMES REFLEX AND CONCENTRATION TWO-PLAYER GAMES KID'S GAMES MORE GAMES WORD GAMES ELEMENTARY MATH MIDDLE/HIGH SCHOOL MATH VOCABULARY AND READING MUSICAL EDUCATION KALEIDOSCOPES AND DISPLAYS

For descriptions of these send a PowerC for my catalog.

DIY HARDWARE ARROW KEYBOARD

BY I.E. MCCRE,

34 Bucknalls Lane, Garston, Watford, Herts, WD2 7NQ.

When using MULTIPLAN you need to move up,down,left and right, either by character or by windows. If you use the normal console method you have to continually be pressing two keys at a time, which can slow down the whole process. This can be speeded up slightly by using the updated version with auto repeat, but this is still slower than using the arrows available on most of the latest computers that we use today.

I therefore decided to produce a separate keyboard. I thought about building it into the existing TI keyboard, but felt that there was not enough room to fit it in, even if I stopped using my navarone widget.

On a separate sheet there is a list of the components used with reference made to the "MAPLINS" catalogue numbers as many people can only order by mail and they are not too expensive.

In addition I have produced some diagrams to show the connections to be made. These are indicative only and you will need to work out the details to suit your own requirements.

The most sensible box available was the "desk console", but this had to be reduced in length to avoid the speech synthesizer. If you do not have a S.S or intend to get one the console will just fit in front of the P.E.B. port without the need for modification.

The lid of the console has to be cut to form the holes for the keys, switch and bolts. I replaced the metal lid supplied with a piece of plastic cut from a floppy disc box as this was easier to cut than the metal one supplied.

The next problem is to close the contact for the control or function key just before the letter key. I considered doing this using either relays or electronically but decided to do it mechanically as this was simpler.

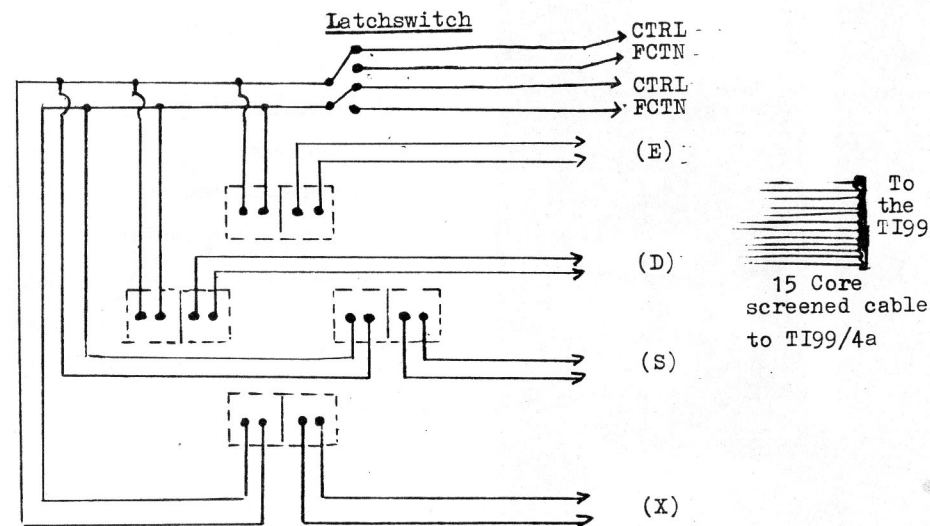
The method I used therefore was to fix two keyboard switches side by side with one keytop connecting them together. To ensure that one switch is activated before the other I have fixed one side higher than the other. This seems to work although once the correct height has been found for the keytop it should be fixed with a spacer or glue as it will come out of adjustment with use.

The keyboard switches are fixed to a veroboard and the various links made to this. The latch switch is connected between the second keyboard switch and the connection to the control or function on the TI keyboard, so that you can change from one to the other.

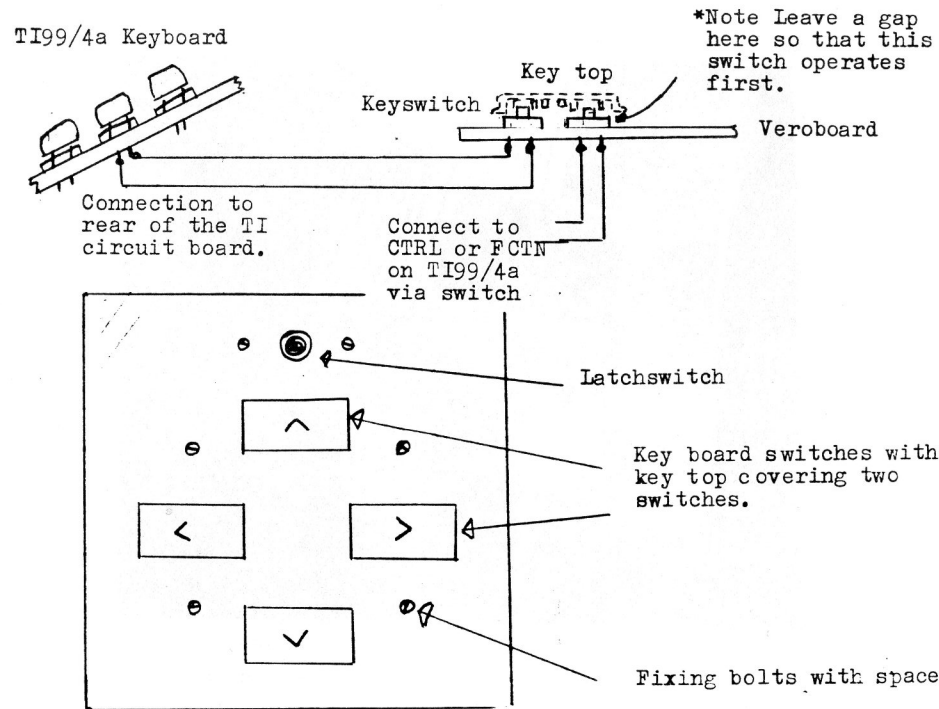
I have not shown the connections to the underside of the TI keyboard as this is obvious when you look at it. I would recommend disconnecting the keyboard before soldering on these connections to ensure that there is no risk of static reaching the main processor board.

I have used a shielded multi cable between the two units as a further precaution. Once inside the case of the TI console I removed the outer casing to the cable so the smaller wires could pass through the space available. The cable was fixed at the back of the TI console with liquid gasket as this holds it rigidly.

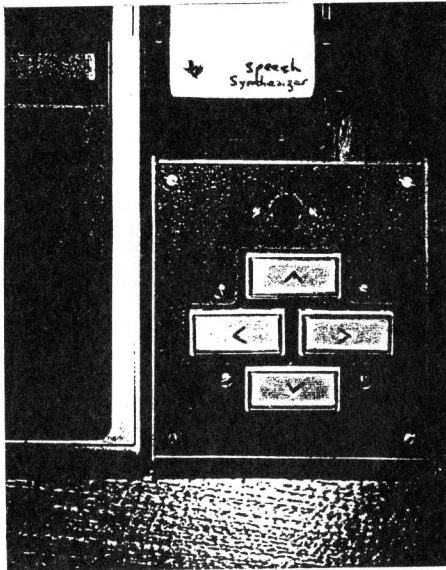
LINE DIAGRAM OF CONNECTIONS TO THE UNDERSIDE OF VEROBOARD



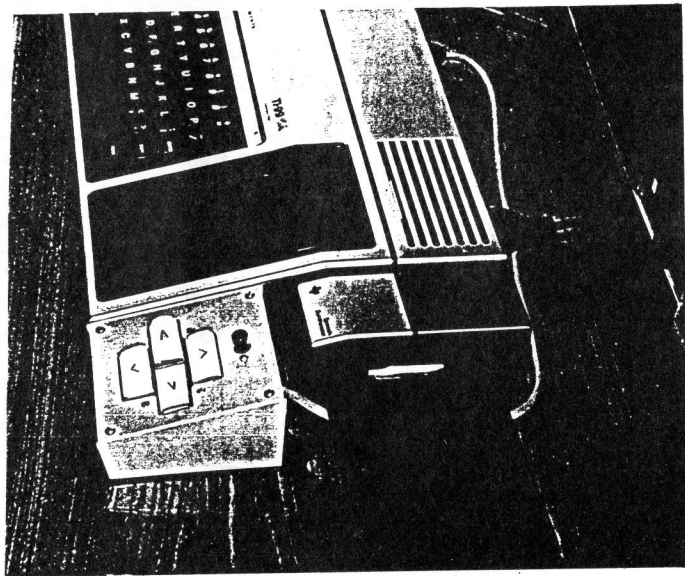
EXAMPLE OF A CONNECTION FROM TI99/4a TO VEROBOARD



TOP VIEW OF VEROBOARD KEY BOARD



TOP VIEW (Note space around key tops)



SIDE VIEW (Note the route for connection of cable)

COMPONENTS LIST	Maplins cat No
1 X Desk console	M6005
8 X Keyboard switches	ff61R
4 X Keyboard switch key tops	ff63T
1 X Latchswitch	FH69A
1 X Veroboard	----
1 metre 15 core screened cable	XR28F
1 X 3/8" Grommets	

COMPUTER HOME SERVICE

SALE OF MODULES.

PARSEC £8.95. ADVENTURE/PIRATE £10.95

TI INVADERS £5.95. CHESS £10.95

MUNCHMAN £3.95. SOCCER £7.95

ADDITION/SUBTRACTION 1. £5.95.

JOYSTICKS. HOTSHOT switched joysticks. Single £7 SPECIAL OFFER!

BEST QUALITY DISKS BY NASHUA.

Nashua 99/DD. Packs of 10. £12.00 +40p P+P.

Library Cases for 10 discs (RRP 3.45) £2. TV AERIAL SPLITTER (SWITCHED) will prevent wear on TV socket and cable. Simply flick a switch to £2.25 +25 P+P

T.I. BOOKS.

KIDS AND THE T199/4a - Dataset.

Learning Basic and the magic tricks of the TI can be great fun so why learn any other way. The drawings help you understand and remember the material so you'll be writing your own programs by Lesson 5 and forever after. Learn to write computer music, create computer graphics and drawings make your computer talk + ask questions create games and quizzes. Edit and debug programs fast organize your mind and think creatively and best of all it's fun.

NOT JUST FOR KIDS!

276 pages and spiral bound. £5.00

GETTING STARTED WITH THE T199/4A by Stephen Shaw...£4.50

MASTERING THE T199/4A by Peter Brooks. £5 or both these books for only £7.00 These prices include postage and packing.

Send Order to COMPUTER HOME SERVICE, 40 BARRHILL PATCHAM BRIGHTON BN1 8UF.

PLEASE ALLOW 28 DAYS TO PROCESS

ALPINE QUEST

A new exciting graphic and text Adventure in TI BASIC by the author of Intrigue's "ADVENTUREMANIA II".

Graphic version uses full memory, capacity of the TI computer and text only version on side B runs 200% faster.

Control two independent characters who must work together to complete the game. Watch the clock and your energy level and beware the avalanche as you explore the locality in search of all your stolen possessions.

AVAILABLE FROM Mark McGurn, 26 Sandiford Road, Holmes Chapel, Cheshire CW4 7BU for only £7.95.

TI TIMES back issues NOW REPRINTED Issues 1,2,3,6 and 7. Also available are issues 4,5,8,9,10 They cost only £1 each to members. T199/4A EXCHANGE, BRIGHTON.

MINIMEM Conversion to Rechargeable Battery. Send Minimem and Crossed cheque £7.50. To: N.I. Petry, Tensel Technology, 3, Lester Drive, WORLE, N.S.W., Avon. BS22 0NG.

Boxed as new £70. 99/4a console only £70. Parsec new boxed

£8. 021 471 2391 evenings.

FOR SALE. Alphacom 42 thermal PRINTER, interface can be supplied for PS232 card or direct connect. £90 DNO. Phone E. Shaw on 0538 760782.

FOR SALE. Munchman, TI Invaders, The Attack £5.50 each. Chess module £8.50. 7 books £2 each. Dual joysticks £5 and other literature-Differs. Tel. 0406 782274.

WANTED WANTED WANTED WANTED WANTED

T199/4A EXCHANGE CAN HELP YOU SELL YOUR UNWANTED TI EQUIPMENT FAST. NO MIDDLEMEN MAKING THE PROFIT. INSTEAD YOU DASH IN. CONTACT THE GROUP FOR AN OFFER OR ADVERTISE IN THESE PAGES

FOR SALE. Alphacom 42 PRINTER and TI Interface with paper. £65 or nearest offer. Editor Assembler manual £5 or nearest offer. LARKIN - 0482 704084.

FOR SALE- TI 99/4a computer £25. Speech synthesizer £15. Joystick £5. Cassette lead £3. MODULES - Extended Basic £25. Terminal Emulator II £9. Parsec. Yahtzee. TI Invaders.

Alpiner. Hunt the Mumpus. Blasto. The Attack. Munchman. Early Reading. Blackjack/Poker - £1 each. Chess £6. Or £100 THE LOT. B.White, Maldon, Essex. Tel 0421 58378.

FOR SALE - T199/4A. TI joysticks, Quickshot adapter, Alphacom printer with interface, Minime, Extended Basic, T2k ram, games, books. £140. Will split. Tel 0507 601648.

WANTED. TI PEB at a reasonable price. George Michel, 34 Pippin Hill, Derby Village, Derby. DES 8PD.

WANTED - 99ers and TI mags. Tel. Faversham 533223.

SWAP - Connect 4, Munchman, Congo-Bongo, Adventure (with Strange Odeev), Microsurgeon and other modules and tapes together or separately for most good games. Ned - 01340 7407.

FOR SALE. STILL SPRING CLEANING. (Only one of each so call for availability.)

DISC DRIVES (second hand excellent condition) Genuine T.I. Internal D.D. £45. T.I. stand-alone - £85.

SPEECH SYNTHESISER - New in box £20.

BOOKS.

Initiation into T199/4a assembly language with the Mini memory. £10.

Getting started with the T199/4a- SHAW. £4.

Smart programming for Sprites- Miller £4. SOFTWARE.

MODULES:-

Extended basic £30. Munchman £3. Parsec £8.

MINER 2049er (rare) £15.

Addition/Subtract 1 £5. Invaders £5.

CASSETTES:- BASIC.

At £7 each - Oldies but Goodies- Games II. (TI) Mania (I) Santa + the Goblins (I) Sianns Saucer (99er) D.I.Y. Adventure (Timeless).

At £2 each - Nodrubs Quest (Program Factory). Munchback Havoc (Lantern) TI Cricket (Andromeda). Funpac Supermouse/Lets go to the races. (Virgin) Bobopods (V). UFO and Fruit machine (Christine computing). Beginners Basic (TI).

CASSETTES:- EXTENDED BASIC.

Shuttle Attack (I) £3.

Intrigue Pentathlon £3. TI 99/4A EXCHANGE. 40 BARRHILL, PATCHAM, BRIGHTON. BN1 8UF 0273 50798.

ADVERTISE ON THIS PAGE FOR 5P A WORD. (MEMBERS ONLY) SWAP SHOP AND FREE Trade Advertisements Rates on application.

!! CRAZY SALE !! COMPUTER + SPEECH +32k RAM/DIY! COMPLETE WITH BOOKS £99.00 DNO EXT BASIC £35.00 LOGO II £35.00 CHESS £8.00 SOCCER £5.00 (EXTENDED BASIC TAPES) PENTATHLON SNOOT OF SPOUT. NOTEWORTHY. LIONEL AND LADDERS, ARENA III BATTLESTAR ATTACK (BASIC TAPES) ADVENTUREMANIA, SANTA AND THE GOBLINS, MANIA. ALL £7.00 EACH OR LOT FOR £18.00. BOOKS Smart Guide to sprites £4.00 Entertainment Games in Basic/ev £4.00 Compute Collection Vol 1 £3.00 99er Aug/Sept £2.00 Tel SHEFFIELD 455582.

MODEMS AND TI99/4A

BY N. BOSWORTH A. DAVEY.

(1) WHAT IS A MODEM?

The word Modem is derived from its main function in MODulating and DEModulating. This means that it is a device that converts universally understood ASCII code from your computer into audible high and low tones that can be sent down a telephone line to another computer. The modem at the other end re-converts these tones back into ASCII text which it then passes on to the computer to which it is attached. Neither the computer nor modem needs to be of the same make as yours in order to communicate.

(2) PROTOCOLS.

Protocols are parameters that must be matched between both computers in order for them to communicate correctly. The various Protocols are considered below: (a) BAUD RATE- This setting ensures that both computers communicate at the same speed. The Baud rate approximates to the number of bits transmitted each second. The most common communication rates are: (aa)300/300 where you and the remote computer communicate at the same speed of 300 baud, which is the speed most commonly used by the free bulletin boards. (ab)1200/75 where the remote computer sends information to you at 1200 bits per second, with you responding to prompts at 75 bits per second, which is as fast as a competent typist can manage. This is used by systems such as Prestel and Microlink. (ac)1200/1200 which is useful for rapid file transfers between two home computer users. (b) PARITY- An eighth Parity bit can be added to the seven bit ASCII code being transmitted to enable transmission errors to be detected. An EVEN parity means that an ASCII code has an even number of ones in it, and an ODD parity means an odd number of ones, while the third option of NONE is obvious. Although the number of data bits to be sent is 7 or 8, the actual number of bits transmitted is larger as a single start bit is automatically added to the front of the code and one or two stop bits can be added to the end of the code to enable the remote device to verify the accuracy of each piece of code being received, thus allowing retransmission to be requested if required. This means that a total of TEN bits are usually transmitted, and or expected by the remote device. The Parity setting is decided by the requirements of the host computer, but if unknown will usually be found to be EVEN. (c) DUPLEX- This setting can be either FULL or HALF Duplex. FULL duplex means that data can flow in both directions at the same time. (usually used for accessing bulletin boards and exchanging programs with other home computers). HALF duplex means that data can flow in both directions, but not at the same time. (usually used for high speed file transfer between two home computers). (d) ECHO ON/OFF-When the Echo is ON, all the characters that you send will be echoed back to your screen by the remote computer. When the Echo is OFF your screen will show the characters as they are typed in. NOTE. This is also often referred to as duplex. Probably because the two setting usually go hand in hand. EG.FULL Duplex / Echo ON, and HALF Duplex / Echo OFF.EXCEPT when two home computers are communicating when ECHO OFF must be selected wether in FULL, or HALF Duplex.

(3) INTERFACING.

All that you need to connect a modem to your TI is an RS232 interface. The cable to connect your RS232 to the modem can be purchased along with the modem in most cases, or you can easily obtain the parts to make your own from your local Tandy store. The connections to be made and the details of the parts required can be found in your RS232 manual. Do not be put off by the number of possible connections, as most modems will operate well if only the signal in, the signal out, and the signal ground are connected.

(4) MODEM CHOICE

The biggest problem with most people is deciding which one would best suit their needs. So here are some pointers as to what features to look for when choosing your modem. (a) BT APPROVAL- Only modems with the green circle sticker can be legally used on the British telephone system. (b) OPERATING SPEEDS- You would be strongly advised to obtain a modem with the ability to operate at the following speeds:300/300, 1200/75, and 1200/1200. If you wish to access boards in the USA it is not necessary (nor legal we suspect) to obtain a modem that will support the american BELL as well as the european CCIT standards.You can simply join BTs PACKET SWITCHSTREAM (I think that joining some databases can give automatic membership to Packet Switchstream). (c) AUTO ANSWER- This option is available, but is not essential unless you wish to operate your own bulletin board. (d) MODEM STATUS LIGHTS- The importance of these, no matter how big or small the modem is obvious as they are going to keep you informed about what the modem is doing or waiting to do.They should be readable at a glance are as follows: 1.PWR(power on) 2.DTR(data terminal ready) 3.DCD(data carrier detected) 4.RXD(incoming data) 5.TXD(outgoing data) 6.ON-LINE(to another modem).



(e) COST- The amount that you pay is up to you. A simple modem operating at 300 baud only, can be bought for around £50, while one that operates at all the popular speeds but does not have auto answer/dial can be bought for £100-150 which is only half the cost of a decent printer! From £150 upwards gives you full auto answer/dial and 1200 to 75 modem controlled speed down which would allow you to output to the modem at twelve hundred with the modem buffering it down to 75 before transmission.

(5) SOFTWARE

All that is basically needed a TE2 module which allows you easy access to all of the free 300 baud boards (including 4/ABC)around the country. It also allows you to transfer files with your friends, albeit at a slow rate. However there are now for disk drive owners several non commercial programs available at negligible cost, that considerably outperform the TE2 and therefore make it a non essential purchase.(although its a good item to have especially if you want to communicate with someone who does not happen to own a disk drive and ram expansion). We will give you a run down

disk drive and ram expansion). We will give you a run down of some of these now.

(a)BI(DSK+32K+MM/EA)- Available from TIMES See issue eight library update for details. This program will do everything that the TE2 does, and more, and is highly recommended.

(b)TE3C(DSK+32K+MM/EA)- Available from S.Shaw. This is the terminal emulation program that TI never finished, but which someone has got working. This is a fairly easy program to use that allows several different communication rates but only 300 baud works reliably. The program incorporates a 24K RAM buffer which will allow you to log on to a bulletin board and download a vast quantity of material very quickly, which can be studied at your leisure after logging off, thus saving on phone bills.

(c)FASTER(DSK+32K+EA/MM/EB) Also available from TIMES See issue eleven library update for details. This is the best of all. The program allows you to print all the information appearing on your screen, and the displayed data can be as much as four thousand characters in front of the printer at any one time, which means that you do not have to pause (as with the TE2 module) while the printer catches up. The program also has a screen dump facility so that only selected screens can be saved to your printer. File exchange is also well supported. The colour of the text as well as the screen can be chosen to suit, and there is even a time elapsed on call display at the top of the screen. The text buffer in this program will allow you to fast or slow scroll through all of your stored information after logging off from the bulletin board(the fast scroll is HYPER fast!).

(d)PRESTEL(DSK+32K+EB/MM/EA) Only available from TIMES TI99/4a Exchange library This is a superb program which is extremely easy to load and operate, which will open up the multicolour world of Prestel and Micronet to you. Free test is available on PRESTEL (see Special offer to join MICRONET)

(6) FUTURE DEVELOPMENTS.

Try your hand at logging onto your nearest free bulletin

board, the number of which you will find in the computer mags. Unfortunately you may be disillusioned by the number of boards and messages dedicated to other computers, to the exclusion of the TI. Well, the answer lies with you. There is no reason why TI modem users should be left out in the cold. All you have to do, even if you are only thinking of buying a modem, is to let us know that you exist. Then if enough of us get together we can pool our know how and resources to get our own special interest group on PRESTEL/MICRONET, and then the sky is the limit. The attraction of these databases is that although the membership fee is quite high, they operate at 1200/75 baud which makes information transfer much quicker and therefore cheaper, and of course something like 98% of the country can phone in to them for only the local call rate. Another great asset of these systems is that special interest group members can chat to one another live! (ED: See special introduction offer exclusive to TI99/4a Exchange members)

(7) HELPFUL TIPS.

For full details on modems call us now

(a)Read the MODEM and RS232 manual/instructions first. (b)-Read them again. Problems can often arise from that small item that got overlooked during the first reading. (c)-Phone Alan Davey on 04606-4511 for advice on modems,wiring connections and Prestel numbers, also how to access free demo on Prestel. (d) Phone NEVILLE BOSWORTH on 0703-441714 for advice on terminal software (loading software if you do not have E/A or M/M) bulletin boards and closed user group developments. Telephone either of us for any other problems (evenings).

EDITORS NOTE This article was transmitted down the telephone line direct to us for TIMES. It is with thanks to Nevill and Alan for the many hours work in finding how best to use a MODEM. Incidentally TIMES use an Interlekt modem, TE2 module, disk drive. We can place an order for you V.A.T. FREE from Interlekt. We strongly recommend TI99ers that you take advantage of MICRONETS great offer. The TI99/4a will certainly be seen to be alive and well if only you will support us.

MULTI-STANDARD MODEM

Let's face it, you're not going to rave about a little box that happily supplies all your communication requirements without fuss or bother.

That's because you can be sure your British designed and built Interlekt modem will perform superbly.

For PCs, Prestel, electronic mail and private viewdata, Interlekt has the V21/V23 modem for you:

Prospect - modem.

originate or answer mode, front panel speed/mode selection.low power consumption.

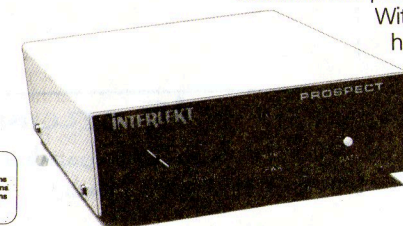
With our one-year warranty and telephone helpline direct to the manufacturer, you won't rant about being neglected after you've bought one of our modems either.

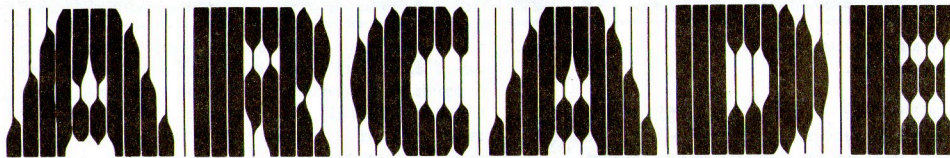
That's why we say, show us an Interlekt modem user, and we'll show you a very contented person.

DEC Interlekt Limited



APPROVED for use with telecommunication systems run by British Telecommunications in accordance with the conditions in the V21/V23 specifications for use.





HARDWARE

211 Horton Road, Fallowfield, Manchester M14 7QE. Tel: 061-225 2248

SOFTWARE

MICRO-SURGEON	£14.95
DEMON ATTACK	£14.95
FATHOM	£14.95
BIGFOOT	£14.95
SUPERFLY	£14.95
SPACE BANDITS	£14.95
HONEY HUNT	£14.95
SEVERMANIA	£14.95
SOUNDTRACK TROLLEY	£14.95
METEOR BELT	£14.95
ADVENTURE/PIRATE	£14.95
RETURN TO PIRATE ISLE	£14.95
FROGGER (see special)	£24.95
Q#BERT (see special)	£24.95
MIDNITE MASON	£19.95
T.I. INVADERS	£ 7.95
MUNCHMAN	£ 7.95
CHISHOLM TRAIL	£ 9.95
CAR WARS	£ 9.95
KILLER CATERPILLAR	£24.95

ADVENTURE TAPES

ADVENTURELAND	£ 9.95
MISSION IMPOSSIBLE	£ 9.95
VOODOO CASTLE	£ 9.95
THE COUNT	£ 9.95
STRANGE ODYSSEY	£ 9.95
MYSTERY FUN HOUSE	£ 9.95
GHOST TOWN	£ 9.95
GOLDEN VOYAGE	£ 9.95

BOOKS

GUIDE TO EXBAS APPLICATIONS	£11.95
T.I. COLLECTION VOL 1	£10.95
33 PROGRAMS FOR THE TI99/4A	£11.95
GUIDE TO SOUND & GRAPHICS	£11.95
BEGINNERS GUIDE TO ASSEMBLY	£11.95
PROGRAMS FOR THE TI99/4A	£14.95
Miller newsletters. Complete set	£5.00 <i>NEW.</i>
SMART PROGRAMMERS GUIDE TO SPRITES	£6.95

PROGRAMS

EXPLORER	£24.95
DIAGNOSTICS	£19.95
DISC REPAIR KIT	£19.95
DFX SCREEN DUMP	£19.95
MULTIPLAN	£49.95
T.I. LOGO II	£49.95
MINI-MEMORY	£49.95
EXTENDED BASIC	£69.95

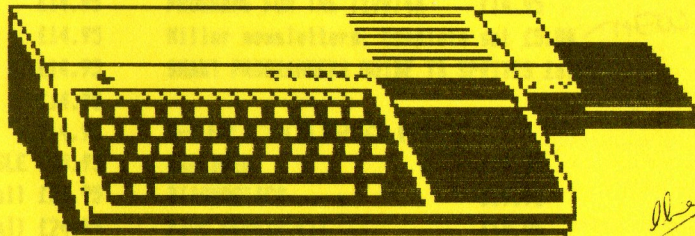
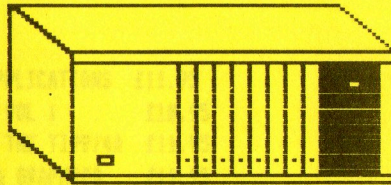
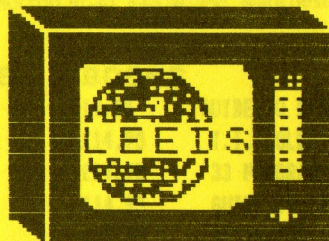
THORN-EMI GAMES POSSIBLY
AVAILABLE ON DISC IN TIME FOR
LEEDS. IF SO COMPUTER WAR
RIVER RESCUE AND SUBMARINE
COMMANDER TO COST £35.00.

TO CLEAR STOCKS - I WILL BE SELLING INFOCOM
ADVENTURES BELOW COST! Limited numbers. THEY WILL
BE AT THOSE PRICES FOR THAT DAY ONLY!

TI USERS SHOW

MEET FELLOW TI99/4A USERS

FOR
TALKS
DEMOS
SWAPS
TI99 GROUPS-DEALERS-AUCTIONS
BRING
ALL OF
YOUR TI
FRIENDS



SATURDAY 3RD MAY 1986 10AM
CENTENARY HOUSE NORTH ST
LEEDS
PRESENT THIS VOUCHER FOR
YOUR FREE ADMISSION

LEEDS TI99/4 U.K. SHOW
SATURDAY 3RD MAY 1986
FREE ADMISSION VOUCHER

NAME _____

ADDRESS _____

Make the most of your micro

MICRONET 800 is a fully interactive Database for Home and Business Micro Users.

There are over 30,000 pages of information on MICRONET, with News, and Views on the whole computer industry.

There is interaction between YOU, US and over 70,000 other users around the U.K. and the world.

There are magazines on computers and business, offering specialist information on everything from how to wire up a serial printer to the best investments on the stock exchange.

A Firm of Solicitors is on-line to give FREE legal advice.

AND MUCH MUCH MORE...

WE ARE OFFERING A SPECIAL DEAL IN THIS MAGAZINE ONLY, OF EIGHT POUNDS OFF THE FIRST QUARTER'S SUBSCRIPTION OF SIXTEEN POUNDS FIFTY.

THAT'S HALF PRICE!

Contact your Micronet Agent, MARTYN AMESS, on 04246 (Battle) 4813 (evening) or 0892 29783 (daytime) for more information on what Micronet can do for you and a special brochure and application form.

If you wish to join immediately, please fill in the coupon below, and send it off to us.

Micronet 800



Cut out the coupon
and send it to:

Micronet 800
8 Herbal Hill, London EC1R 5EJ
Telephone: 01-278 3143

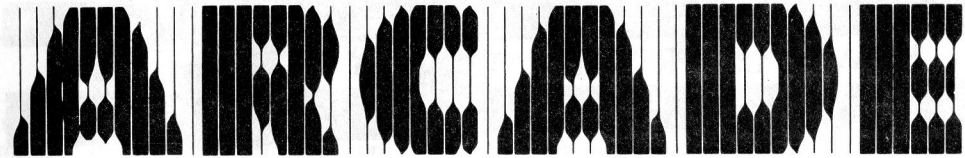
TI USER SPECIAL OFFER

Name _____

Address _____

_____ Telephone No _____

Age _____ Micro TI99/4a _____



HARDWARE

211 Horton Road, Fallowfield, Manchester M14 7QE. Tel: 061-225 2248

PERIPHERALS

MYARC MINI BOX	£450.00
Now to order only	
MYARC RS232 CARD	£125.00
MYARC DISC CONTROL CARD	£185.00
MYARC 128k CARD	£230.00
MYARC 512k CARD (t.o.o.)	£345.00
MYARC HARD DISC CARD	£350.00
(Hard disc card requires drive	

NIGHTINGALE MODEM £145.00

JOYSTICKS

COMP PRO PAIR	£ 35.00
COMP PRO SINGLE	£ 19.25

OTHER ITEMS

RS232 CABLE	£18.00
CENTRONICS CABLE	£18.00
THAMES BINDERS	£ 3.95
CASSETTE'M'GAME FILE	£22.50

SPECIALS

MBX Voice recognition unit + 9 games for only £110.00 including transformer. The nine games alone would come to £134.55. This offer is very limited and unrepeatable.

PARKER SPECIAL

FROGGER + Q*BERT £40.00 save £10.00
ALL THREE PARKER BROS FOR £60.00
(Dependant on availability of Popeye)

NEW PRODUCTS D

MYARC 80 column card. Sorry, no price or availability date.

MILLERS GRAMCRACKER.....£175.00
Price dependant on rate of exchange not going below 1.42. First copies

DISC DRIVES	
Single sided internal	£115.00
Double sided internal	£135.00
Single sided External	£135.00
Double sided External	£155.00

PRINTERS

SHINWA CPA 80	£230.00
Other printers available to order. Delay is only 3 days.	

EDUCATIONAL MODULES

FROG JUMP	£ 9.95
PYRAMID PUZZLER	£ 9.95
STAR MAZE	£ 9.95
SPACE JOURNEY	£ 9.95
DRAGON MIX	£ 9.95
MINUS MISSION	£ 9.95
NUMBER READINESS	£ 9.95

SEE YOU IN LEEDS!

MEET FELLOW TI99/4A USERS

FOR TALKS DEMOS SWAPS

TI99 GROUPS - DEALERS

BRING ALL OF YOUR TI FRIENDS

SATURDAY 3RD MAY 1986 10AM

CENTENARY HOUSE NORTH LEEDS

YOUR FREE ADMISSION

LEADS TI99/4A USER GROUP

FREE ADMISSION

NAME _____

ADDRESS _____

ACCESSING PRESTEL/MICRONET FROM YOUR TI99/4A Transmitted down line via modems by PETER WALKER.

First you will need program B3 from TI99/4a EXCHANGE group library. You must have Disk drive +32k and Extended basic. You also need a modem that supports split speed of 1200/75 baud.

The program does not support all UK Prestel features; features so far noted as unsupported include flashing characters and conceal/reveal. As with many computers there is no capability of saving screen text to disk or printer. (ED: I have seen a viewdata terminal dump out a screen but only with expensive equipment.)

HINTS ON USING PRESTEL/MICRONET 800.

Keep your Identity No (10 digits) and Password (4 characters) secret. Only quote your account number which is needed for addressing mailbox messages. ID and Password are not echoed on the screen for security. Ensure Shift Lock is in appropriate position if using an alpha password. Note there are several Prestel computers and your password can be different on each one, so if altering password, you will need to do so on each machine. (To change your password go to page 0#) Account numbers are sometimes similar to your telephone number.

is used to terminate input fields, not the Enter key. If you suffer from line noise, you can key #00# to resend current screen.

TO SEND A MAILBOX MESSAGE

Select page #. Fill in MBX address field with the A/C number of addressee followed by #. Directory of registered MBX users is on page 6#. Don't forget to register yourself (page

644#). After MBX number, you can type in the frame provided, using upper and lower case letters and other ASCII characters. The cursor position is invisible which makes editing very difficult. However Arrow keys (FCTN ESDX) do work but, in the case of the Up Arrow and Left Arrow, you have to wait for a few seconds while the screen is retransmitted up to the new cursor position. New text overstrikes old.

To erase message and revert to beginning of text field type

CTRL X. Enter key returns the (invisible) cursor to the beginning without erasing text. To revert to MBX address field type CTRL B. To complete message type # or ESCape J. ESCape K aborts MBX and returns to main index. FCTN V (ASCII Delete) displays double vertical line character. CTRL M is the same as Enter. CTRL J is the same as Down Arrow. CTRL M is the same as Left Arrow. CTRL I seems to be the same as space.

SPECIAL MAILBOX EDITING FEATURES

You can use special ESCape sequences to change the colour of the text and background, select Double Height characters and use block graphics, but note that these attributes use screen space to set up and are reset after each line. For example to change the text from White to Red type ESC A. The ESCape character is sent on the TI99 by CTRL ,(fullstop). Full details of these codes plus graphics characters can be found on page 7768# (0600160# for graphics) or for Micronet users pages 37# and 371# or I can give you more info. If you are a member of Micronet 800 (a closed user group for computer buffs) you can see examples of user-created graphics in "The Gallery" starting on page 3#. Pages beginning 82201# are also worth a look.

PRESTEL/MICRONET ACCESS NUMBERS

There is usually a Prestel access number within local call range of you. (BT give this number when you join). Page 31# gives an index to other codes round the country, but note that these are often unavailable to callers from elsewhere. There is a London based 300 baud service on 01-680 8245 which can be dialled nationwide using TE2 or similar emulator. This speed does not support graphics or colour but sends #s in their place.

LEAVING PRESTEL/MICRONET 800

Key # to leave Prestel, then switch off your modem. When quitting from program B3 the normal character set is not reset so the TI has to be switched off and on again before running another program.

FURTHER HELP

I am willing to help any Prestel TI users via MBX 707873778 or by phone 0707 873778.

PETER WALKER ED: We are very grateful to Peter for
24 Bacons Drive, this article.
Cuffley
Herts
EN6 4DU

YOUR LETTERS

>VICTOR REINAR Italy, writes:- I would like to get in touch with some TI tappers of Italy, France or Germany, to exchange ideas and possibly to translate some of their programs. ED Victor can be contacted through TI*MES.

>FRANK WRIGHT Somerset writes:- I have been given a complete "Teletype". (US teleprinter. Since the 99 has control key designations can anyone tell me if I can drive a teletype from the TI. Also has anyone found a solution to the infuriating inability to MERGE from tape rather than disc.

>ERIC HARVEY 16 Gilfil Rd, Hill Top, NUNEATON CV10 7BU requests:- Has anyone the circuit diagram on the Mini Memory module.

>GORDON JONES, Ottershaw, Surrey writes:- Because of a noisy fan I decided to change it for a Radio Spares type axial fan (part 509-030) costing around #15. This was straight forward enough except the new fan was 250v and the old one 115v. Whilst deciding a suitable place to pick up the 250v I noticed on the power input socket (which is of the filter type) the leg which was taken to the expansion box fuse was in fact the neutral supply leg. This is a potentially dangerous and costly fault. A simple reversing of the spade terminals corrects this.

>JOHN DOWNIE 36 Lancaster Ave, Beith, Ayre KA15 1AR writes:- Please could you tell me if an interface could be made to use a disc drive that has a lead for a BBC micro, such as a CUMANA 40/80 DSDD drive, on the TI and if so how to build one.

>LAURA BURNS Editor of MICROPENDIUM writes:- It is a pleasure to get the latest issue of the ever-excellent, ever-informative TI*MES. (ED: Likewise.)

>PHIL MUNRO of Prestwich, Manchester (Tel 061 773 4097) writes:- For the humble "bum" with just the console and ExBasic there appears to be very little in TI*MES except for access to your User group library. Is it allowed to make personal copies of these tapes or are they protected. Also I want to be an AREA CONTACT (ED:- I have to defend ourselves by flicking through issue 11 out of 61 numbered pages, 45 are of interest to persons without expansion, you cannot say that this is not a bad percentage. We realise our shortcomings towards basic only Users and did appeal for writers. We hope to include one next issue. With regard to TI*MES SOFTWARE LIBRARY some confusion arrives from time to time with this. The price you pay includes a cassette or disc. The programs we send you are unprotected. However we do ask that you do not copy and distribute them as this would undermine the library conditions. The programs and cassette are then yours to keep. Please remember that the copyright of the those programs belong to the person or persons who wrote them.

>PETER ODELRYD Jarnaldersringen 340, S-136 65 Handen, SWEDEN :- I am interested in making contact with U.K. TI Users, to exchange information. I also am looking at the use of Modems as dicussed with Alan Davey. (ED: 99ers in Sweden have a strong user group, get in touch with Peter for details.

Address to write is Letters TI99/4a Exchange UK TI Users Group
40 Barrhill Patcham, BRIGHTON BN1 8UF
YOU CAN SEND A FILE DOWN THROUGH THE MODEM OR LEAVE A NOTE ON PRESTEL
OR MICRONET 800. Electronic mailbox number is 273503968.

PLEASE REMEMBER IF YOU ARE WRITING TO ANY OTHER MEMBER OR A CONTRIBUTOR TO THE MAGAZINE TO INCLUDE A STAMPED ADDRESSED ENVELOPE FOR YOUR REPLY. WITHOUT THIS A REPLY WILL NOT BE SENT.

TI BASIC PROGRAM OF DRAUGHTS



```

TI BASIC but you can use the 3E7C78") : 550 FOR I=2 TO 8 : 1070 Q=1
Extended basic module.If you : 300 CALL CHAR(90,"787C3E3F1F : 560 CALL COLOR(I,2,1) : 1080 GOTO 870
enjoy a good board game then : 0F0300") : 570 NEXT I : 1090 NEXT X
this program is well worth : 310 CALL CHAR(91,"1E3E7CFCFB : 580 CALL HCHAR(1,1,104,32) : 1100 IF Q=1 THEN 1470
the time spent in typing in. : F0C000") : 590 CALL HCHAR(24,1,104,32) : 1110 NEXT B
: 320 CALL CHAR(96,"00030F1F3F : 600 CALL VCHAR(1,1,104,48) : 1120 FOR A=1 TO 2
DRAUGHTS can be obtained : 3E7C78") : 610 CALL VCHAR(1,31,104,48) : 1130 FOR B=1 TO 12
from the TIMES Library : 330 CALL CHAR(97,"00C0F0FBFC : 620 CALL COLOR(10,13,13) : 1140 C=B(1)
: 7C3E1E") : 630 CALL KEY(3,K,S) : 1150 IF (AS(C)<>96)*(AS(C)<>
We thank Syd Michel for his : 340 CALL CHAR(98,"787C3E3F1F : 640 IF K<>13 THEN 630 : 1160 THEN 1370
skill and effort in bringing : 0F0300") : 650 CALL CLEAR : 1160 FOR X=1 TO 4
this excellent program for : 350 CALL CHAR(99,"1E3E7CFCFB : 660 CALL CHAR(89,"00C0F0FBFC : 1170 N=BS(X)
all TI99/4a users everywhere : F0C000") : 7C3E1E") : 1180 IF ((X<3)*(AS(C)=96))+
: 360 CALL CHAR(92,"00030F1F3F : 670 GOSUB 3460 : AS(C)=100)THEN 1190 ELSE 137
: 3F7F7F") : 680 GOTO 2520 : 0
PROGRAM LISTING AS FOLLOWS:- : 370 CALL CHAR(93,"00C0F0FBFC : 690 RESTORE 2320 : 1190 IF AS(C+N)=112 THEN 120
: FCFEFE") : 700 PRINT " : 0 ELSE 1360
: 380 CALL CHAR(94,"7E7F3F3F1F : PLEASE" : 1200 IF (A=1)*(((AS(C+2*N)=8
10 REM DRAUGHTS 657 : 0F0300") : 710 FOR A=4 TO 23 : B)+(AS(C+2*N)=92))+((AS(C+N
20 REM WRITTEN BY SYD MICHEL : 390 CALL CHAR(95,"FEFEFCFCFB : 720 FOR AA=11 TO 30 : +2*-ABS(N))=88)+(AS(C+N+20-A
30 REM MARCH 1986 : F0C000") : 730 READ BIT : BS(N))=92))*((AS(C+N-20+ABS(N
40 REM ***** : 400 CALL CHAR(100,"00030F1F3 : 740 CALL HCHAR(A,AA,BIT) : )=112))THEN 1360
100 GOSUB 3460 : 3F7F7F") : 750 NEXT AA : 1210 AS(C+N)=AS(C)
110 CALL CHAR(126,"003B444447 : 410 CALL CHAR(101,"00C0F0FBFC : 760 NEXT A : 1220 NB=C+N
C444444") : BFCFEFE") : 770 GOSUB 3510 : 1230 NA=AS(C)
120 CALL CHAR(129,"007B242423 : 420 CALL CHAR(102,"7E7F3F3F1F : 780 RETURN : 1240 GOSUB 3040
8242478") : 0F0300") : 790 CALL HCHAR(24,1,32,32) : 1250 AS(C)=112
130 CALL CHAR(136,"003B444404 : 430 CALL CHAR(103,"FEFEFCFCFB : 800 MSG$="THINKING!!" : 1260 NB=C
0404438") : BFC0000") : 810 COL=13 : 1270 NA=AS(0)
140 CALL CHAR(131,"007B242424 : 440 DIM B(12),AS(100),BS(4) : 820 GOSUB 2920 : 1280 GOSUB 3040
4242478") : 450 CALL CLEAR : 830 Q=0 : 1290 B(1)=C+N
150 CALL CHAR(132,"007C404040 : 460 FOR I=2 TO 8 : 840 FOR B=1 TO 12 : 1300 IF C+N<B1 THEN 1350
840407C") : 470 CALL COLOR(I,1,1) : 850 C=B(1) : 1310 AS(C+N)=100
160 CALL CHAR(133,"007C404040 : 480 NEXT I : 860 IF (AS(C)<>96)*(AS(C)<>1 : 1320 NB=C+N
8404040") : 490 PRINT TAB(7);"D R A U G : 870 THEN 1110 : 1330 NA=100
170 CALL CHAR(134,"003C404005 : H T S":"IN THIS SIMPLE GAME : 870 FOR X=1 TO 4 : 1340 GOSUB 3040
C444438") : OF":"DRAUGHTS THE PLAYER IS : 880 N=BS(X) : 1350 GOTO 1470
180 CALL CHAR(135,"0044444447 : BLACKAND THE COMPUTER IS WHI : 890 IF ((X<3)*(AS(C)=96))+A : 1360 NEXT X
C444444") : TE,." : S(C)=100)THEN 900 ELSE 1090 : 1370 NEXT B
190 CALL CHAR(120,"001030101 : 500 PRINT "AFTER DECIDING WH : 900 IF (AS(C+N)=88)+(AS(C+N) : 1380 NEXT A
0101038") : O IS TO GO FIRST THE SCREEN : =92)THEN 910 ELSE 1090 : 1390 MSG$=" I HAVE LOST "&
200 CALL CHAR(121,"003B440404 : IS BUILT":"AND PLAY BEGINS." : 910 IF AS(C+2*N)=112 THEN 92 : HR$(Z)
810207C") : 510 PRINT "WHEN PROMPTED ENT : 0 ELSE 1090 : 1400 ROW=24
210 CALL CHAR(122,"003B440404 : ER THE":"POSITION OF THE PIE : 920 AS(C+2*N)=AS(C) : 1410 COL=6
8044438") : CE TO":"BE MOVED(LETTER THEN : 930 NB=C+2*N : 1420 GOSUB 2920
220 CALL CHAR(123,"0009102094 : NUMBER)" : 940 NA=AS(C) : 1430 AS(C)=100
8700000") : 520 PRINT "THEN PRESS ENTER : 950 GOSUB 3040 : 1440 NB=C
230 CALL CHAR(124,"007C40700 : TO MAKE":"THE MOVE,OR ANY OT : 960 AS(C)=112 : 1450 NA=100
4044478") : HER KEY TOCANCEL THE MOVE." : 970 NB=C : 1460 GOSUB 3040
240 CALL CHAR(125,"001020407 : 530 PRINT "TO SAVE A GAME T : 980 NA=AS(0) : 1470 REM
8444438") : O CSI":"WHEN PROMPTED FOR Y : 990 GOSUB 3040 : 1480 IF CASS=0 THEN 2780
250 CALL CHAR(126,"007C400081 : OUR MOVE":"TYPE,.,S":"TO LOA : 1000 AS(C+N)=112 : 1490 CALL HCHAR(24,1,32,32)
0202020") : D A GAME FROM CSI" : 1010 NB=C+N : 1500 MSG$="FROM ? "
260 CALL CHAR(127,"003B444443 : 540 PRINT "TYPE,.,L,." THE G : 1020 NA=AS(0) : 1510 ROW=24
8444438") : AME IS WELL ERROR":"TRAPPED : 1030 GOSUB 3040 : 1520 COL=3
270 CALL CHAR(104,"0") : AND WILL NOT LET YOUAKE AN : 1040 B(1)=C+2*N : 1530 GOSUB 2920
280 CALL CHAR(112,"0") : ILLEGAL MOVE,." NOW PRES : 1050 C=C+2*N : 1540 CALL SOUND(100,440,0)
290 CALL CHAR(88,"00030F1F3F : S ENTER" : 1060 IF C=0 THEN 1430 : 1550 FLAG=0

```

```

1560 GOSUB 2960 : 2070 NA=AS(6) : 4,104,112,112,104,104,112,11 : 2710 ROW=2
1570 IF CASS=0 THEN 2780 : 2080 GOSUB 3040 : 2,104,104,112,112,104,104,13 : 2720 COL=1
1580 R1=K : 2090 IF H>19 THEN 2140 : 6,136 : 2730 GOSUB 2920
1590 CALL SOUND(10,660,5) : 2100 AS(H)=92 : 2420 DATA 132,136,104,104,11 : 2740 ROW=24
1600 MSG$=CHR$(R1) : 2110 NB=H : 2,112,104,104,112,112,104,10 : 2750 CALL KEY(3,K,S)
1610 COL=8 : 2120 NA=92 : 4,112,112,104,104,112,112,13 : 2760 IF S=0 THEN 2
1620 GOSUB 2920 : 2130 GOSUB 3040 : 6,132 : 2770 PRINT :
1630 FLAG=1 : 2140 AS(6)=112 : 2430 DATA 136,136,104,104,11 : 2780 GOSUB 690
1640 GOSUB 2960 : 2150 NB=6 : 2,112,104,104,112,112,104,10 : 2790 IF CASS=0 TH
1650 C1=K : 2160 NA=AS(0) : 4,112,112,104,104,112,112,13 : 2800 FOR CA=12 TO
1660 CALL SOUND(10,770,5) : 2170 GOSUB 3040 : 6,136 : 2810 IF (CA=20)+(
1670 MSG$=CHR$(C1)&" ? " : 2180 IF (ABS(H-6)<>22)*(ABS( : 2440 DATA 133,136,88,89,104, : 40)+(CA=50)+(CA=6
1680 COL=9 : H-5)<>18)THEN 790 : 104,88,89,104,104,88,89,104, : (CA=80)THEN 2860
1690 GOSUB 2920 : 2190 AS((H-6)/2)=112 : 104,88,89,104,104,136,133 : 2820 IF (CA=21)+(
1700 FLAG=0 : 2200 NB=(H+6)/2 : 2450 DATA 136,136,90,91,104, : 41)+(CA=51)+(CA=6
1710 GOSUB 2960 : 2210 NA=AS(0) : 104,90,91,104,104,90,91,104, : (CA=91)THEN 2860
1720 R2=K : 2220 GOSUB 3040 : 104,90,91,104,104,136,136 : 2830 NA=AS(CA)
1730 CALL SOUND(10,660,5) : 2230 REM : 2460 DATA 134,136,104,104,88 : 2840 NB=CA
1740 FLAG=1 : 2240 CALL HCHAR(24,1,32,32) : 104,90,91,104,88,89,104,104,88 : 2850 GOSUB 3040
1750 MSG$=CHR$(R2) : 2250 MSG$="IS ANOTHER JUMP P : 89,104,104,88,89,136,134 : 2860 NEXT CA
1760 COL=14 : 2260 POSSIBLE ?" : 2470 DATA 136,136,104,104,90 : 2870 IF (K=89)+(C
1770 GOSUB 2920 : 2260 COL=4 : 1,104,104,90,91,104,104,90 : 2880
1780 GOSUB 2960 : 2270 GOSUB 2920 : 1,104,104,90,91,136,136 : 2880 GOTO 790
1790 C2=K : 2280 CALL KEY(0,K,S) : 2480 DATA 135,136,88,89,104, : 2890 CASS=0
1800 CALL SOUND(10,770,5) : 2290 IF S=0 THEN 2280 : 104,88,89,104,104,88,89,104, : 2900 GOTO 1470
1810 MSG$=CHR$(C2) : 2300 IF K=89 THEN 1480 : 104,88,89,104,104,136,135 : 2910 END
1820 COL=15 : 2310 GOTO 790 : 2490 DATA 136,136,90,91,104, : 2920 FOR I=1 TO LI
1830 GOSUB 2920 : 2320 DATA 136,136,120,136,12 : 104,90,91,104,104,90,91,104, : 2930 CALL HCHAR(R
1840 MSG$="ENTER)IF OK" : 1,136,122,136,123,136,124,13 : 104,90,91,104,104,136,136 : C(SEG$(MSG$,1,1))
1850 COL=17 : 6,125,136,126,136,127,136,13 : 2500 DATA 136,136,136,136,13 : 2940 NEXT I
1860 GOSUB 2920 : 6,136 : 6,136,136,136,136,136,13 : 2950 RETURN
1870 CALL KEY(0,K,S) : 2330 DATA 136,136,136,136,13 : 6,136,136,136,136,136,13 : 2960 CALL KEY(3,K,S)
1880 IF S=0 THEN 1870 : 6,136,136,136,136,136,13 : 6,136 : 2970 IF S<>1 THEN
1890 IF K<>13 THEN 1480 : 6,136,136,136,136,136,13 : 2510 DATA 136,136,120,136,12 : 2980 IF (FLAG=1)
1900 G=10*(R1-64)+(C1-48)+1 : 6,136 : 1,136,122,136,123,136,124,13 : 2560)+( (FLAG=0)*
1910 H=10*(R2-64)+(C2-48)+1 : 2340 DATA 128,136,104,104,96 : 6,125,136,126,136,127,136,13 : 72))THEN 3000
1920 IF (AS(6)<>92)*(R1<R2)T : 97,104,104,96,97,104,104,96 : 6,136 : 2990 RETURN
HEN 1960 : 97,104,104,96,97,136,128 : 2520 RESTORE 2320 : 3000 IF K=83 THEN
1930 IF (AS(6)=112)+(AS(H)<> : 2350 DATA 136,136,104,104,98 : 2530 FOR I=0 TO 9 : 3010 IF K=76 THEN
112)THEN 1960 : 99,104,104,98,99,104,104,98 : 2540 FOR L=1 TO 10 : 3020 CALL SOUND(5
1940 IF (ABS(H-6)<>9)*(ABS(H- : 99,104,104,98,99,136,136 : 2550 READ AS((I+10)+L),WASTE : 3030 GOTO 2960
6)<>18)*(ABS(H-6)<>11)*(ABS : 2360 DATA 129,136,96,97,104, : 2560 NEXT L : 3040 CALL SOUND(-
(H-6)<>22)THEN 1960 : 104,96,97,104,104,96,97,104, : 2570 FOR W=1 TO 20 : 00,3)
1950 IF (ABS(H-6))11)*(AS(6+ : 104,96,97,104,104,136,129 : 2580 READ WASTE : 3050 ROW1=INT(NB/
((H-6)/2)<>96)*(AS(6+(H-6) : 2370 DATA 136,136,98,99,104, : 2590 NEXT W : 3060 COL1=INT((IM
/2)<>100)THEN 1960 ELSE 205 : 104,98,99,104,104,98,99,104, : 2600 NEXT I : 1-1)*2
0 : 104,98,99,104,104,136,136 : 2610 AS(0)=112 : 3070 CALL HCHAR(4
1960 CALL HCHAR(24,1,32,32) : 2380 DATA 130,136,104,104,96 : 2420 DATA 13,19,15,17,26,22, : L1,NA)
1970 MSG$="ILLEGAL MOVE!!" : 97,104,104,96,97,104,104,96 : 28,24,39,35,33,37 : 3080 CALL HCHAR(4
1980 COL=9 : 97,104,104,96,97,136,130 : 2670 FOR B=1 TO 12 : L1,NA+1)
1990 CALL SOUND(50,330,0) : 2390 DATA 136,136,104,104,98 : 2640 READ B(1) : 3090 CALL HCHAR(5
2000 CALL SOUND(100,110,0) : 99,104,104,98,99,104,104,98 : 2650 NEXT B : L1,NA+2)
2010 GOSUB 2920 : 99,104,104,98,99,136,136 : 2660 DATA 11,9,-9,-11 : 3100 CALL HCHAR(5
2020 FOR DEL=1 TO 200 : 2400 DATA 131,136,112,112,10 : 2670 FOR I=1 TO 4 : L1,NA+3)
2030 NEXT DEL : 4,104,112,112,104,104,112,11 : 2680 READ BS(I) : 3110 RETURN
2040 GOTO 1480 : 2,104,104,112,112,104,104,13 : 2690 NEXT I : 3120 CASS=1
2050 AS(H)=AS(6) : 6,131 : 2700 MSG$=" WANT TO GO : 3130 CALL CLEAR
2060 NB=H : 2410 DATA 136,136,112,112,10 : 2710 FIRST ?" : 3140 PRINT "X SAV

```


PROGRAMMING TIPS

```

CS1":
3150 GOSUB 3460
3160 CA#=""
3170 FOR CA=1 TO 100
3180 CA#=#CA#CHR$(ASC(CA))
3190 NEXT CA
3200 CB#=""
3210 FOR CA=1 TO 12
3220 CB#=#CB#CHR$(B1(CA))
3230 NEXT CA
3240 OPEN #1:"CS1",SEQUENTIA
L,INTERNAL,OUTPUT,FIXED 100
3250 PRINT #1:CA#
3260 PRINT #1:CB#
3270 CLOSE #1
3280 CALL CLEAR
3290 RETURN
3300 CASE=2
3310 CALL CLEAR
3320 PRINT " * LOAD GAME FROM
CS1":
3330 GOSUB 3460
3340 OPEN #1:"CS1",SEQUENTIA
L,INTERNAL,INPUT,FIXED 100
3350 INPUT #1:CA#
3360 INPUT #1:CB#
3370 CLOSE #1
3380 FOR CA=1 TO 100
3390 ASC(CA)=ASC(SEG$(CA#,CA,
1))
3400 NEXT CA
3410 FOR CA=1 TO 12
3420 B1(CA)=ASC(SEG$(CB#,CA,
1))
3430 NEXT CA
3440 CALL CLEAR
3450 RETURN
3460 CALL SCREEN(11)
3470 FOR CUL=8 TO 14
3480 CALL COLOR(CUL,1,1)
3490 NEXT CUL
3500 RETURN
3510 CALL SCREEN(13)
3520 CALL COLOR(11,11,11)
3530 CALL COLOR(10,14,14)
3540 CALL COLOR(9,2,11)
3550 CALL COLOR(9,16,11)
3560 CALL COLOR(12,14,12)
3570 CALL COLOR(13,14,12)
3580 CALL COLOR(14,12,12)
3590 RETURN

```

CURSOR CONTROL IN TI BASIC - by Jim Peterson

Many programs require the movement of a cursor or a figure around the screen by the use of the arrow keys, and it is usually also desirable to be able to move diagonally using the W, R, Z and C keys, and to avoid crashing the program by preventing any attempted movement beyond the 24x32 area of the screen, or to permit 'wrap-around'.

The programming routines often used for this purpose are quite lengthy, requiring 35 lines or more in BASIC for B-directional movement. However, they do move the cursor quite rapidly, which may be essential in game programs. Much more compact routines are available, but they may be slower. The following very compact little routine is attributed to Kurt Garcia of the Houston User's Group.

```

100 R=1
110 C=3
120 CALL KEY(3,K,ST)
130 IF (K<>68)*(K<>69)*(K<>8
3)*(K<>88)+(ST=0) THEN 120
140 C=C+(K=68)*(C<30)-(K=
83)*(C>3)
150 R=R+((K=88)*(R<24))-((K=
69)*(R>1))
160 CALL HCHAR(R,C,42)
170 GOTO 120

```

That routine is a bit slow, taking about 20 seconds to move the cursor around the perimeter of the screen, and it does not permit diagonal moves. This next routine allows diagonal moves but is even slower, requiring 26 seconds to traverse the perimeter.

```

90 CALL CLEAR
100 R=1
110 C=3
120 CALL KEY(3,K,ST)
130 IF ST=0 THEN 120
140 C=C+(ABS((K=82)+(K=68)+(
K=67))*ABS(C<32))+((K=87)+(K
=83)+(K=90))*ABS(C>2)
150 R=R+(ABS((K=90)+(K=88)+(
K=67))*ABS(R<24))+((K=87)+(K
=69)+(K=82))*ABS(R>1)
160 CALL HCHAR(R,C,42)
170 GOTO 120

```

The following is perhaps the best compromise between compactness and speed. It permits diagonal movement, goes around the perimeter in about 20 seconds, and is extremely adaptable.

```

100 R=1
110 C=3
120 CALL KEY(3,K,ST)
130 IF ST=0 THEN 120
140 ON POS("WERDCXZS",CHR$(K
),1)+1 GOTO 120,210,190,180,
160,150,250,240,220
150 R=R-(R<24)
160 C=C-(C<31)
170 GOTO 260
180 C=C-(C<31)
190 R=R+(R>1)
200 GOTO 260
210 R=R+(R>1)
220 C=C+(C>2)
230 GOTO 260
240 C=C+(C>2)
250 R=R-(R<24)
260 CALL HCHAR(R,C,42)
270 GOTO 120

```

You will usually want the cursor to erase itself as it moves along. The movement will be smoother if you 'save' the values of the old position while the new position is being computed, then use them to erase the cursor just before it is reprinted.

```

102 R2=1
112 C2=3
260 CALL HCHAR(R2,C2,32)
265 CALL HCHAR(R,C,42)
266 R2=R 267 C2=C

```

In this version, the cursor will stop at the screen border, or will run along it if struck diagonally. If you prefer it to 'wrap around', change lines 150 and 250 to R=R+1+(R=24). Change lines 160 and 180 to C=C+1+(C=32). Change 190 and 210 to R=R-1-(R=1) and change 220 and 240 to C=C-1-(C=1).

If you want the program to also respond to joystick input, change line 130 to read IF ST=0 THEN 142, and add the Lewis routine:

```

142 CALL JOYST(1,X,Y)
143 Z=((X+3*Y)/4)+5
146 ON Z GOTO 240,250,150,220,120,160,210,190,180
That will activate joystick #1. If you want response
from either joystick, you can add:
143 IF (X<>0)+(Y<>0) THEN 145
144 CALL JOYST(2,X,Y)

```

However, all of these options will slow up the response time. Usually it will be better to require a choice of keyboard or joystick at the beginning of the program. This routine utilizes Doug German's neat little routine which activates whichever joystick is being held when the fire button is pressed.

```

50 PRINT "Will you use:"(1) arrow keys, or:"(2)
joystick?"
52 CALL KEY(3,Q,ST)
54 IF (ST=0)+(Q<49)+(Q>50) THEN 52
56 IF Q=49 THEN 100
58 PRINT "Press fire button"
60 CALL KEY(1,K1,S)
62 CALL KEY(2,K2,S)
64 IF K1+K2<>17 THEN 60
66 J=INT(K1/18+K2/9+1)
114 IF Q=50 THEN 142
130 IF ST=0 THEN 120
142 CALL JOYST(J,X,Y)
delete lines 143 and 144
146 ON Z GOTO 240,250,150,220,142,160,210,190,180
270 IF Q=49 THEN 120 ELSE 142

```

Perhaps you will want to move the cursor around the screen without erasing any text or graphics already on the screen. This is easily done:

```

260 IF (R=R2)*(C=C2) THEN 270
262 CALL GCHAR(R,C,G)
264 CALL HCHAR(R2,C2,G2)
268 G2=G

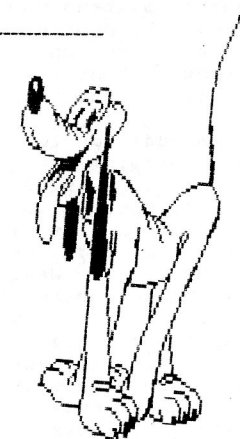
```

And if you want to also use other keys for program input, just add more letters to the string in line 140, and more line numbers in the corresponding positions. For instance, if you want to use the F key to fire at the alien invaders,

```

140 ON POS("WERDCXZSF",CHR$(K),1)+1 GOTO
120,210,190,180,160,150,250,240,220,600
600 CALL SCREEN(16)
610 CALL SCREEN(14)
620 GOTO 120

```



-----> CONTINUED

GRAM KRACKER

SPECIAL

REVIEW

BY

DANNY MICHAELS. SHOALS 99-ERS

The GRAM KRACKER was designed by D.C. Warren and marketed by Millers Graphics. The little box (approx the same size as a widget) contains 56K (expandable to 80K) of battery backed programmable memory and plugs into the module port of the computer. On the surface this may not sound too exciting, but believe me, it is. What makes it so great is the type of memory that's in the box. Before we get into the box let's look at the different types of memory contained in the 99/4A.

The most familiar type of memory to most computer users is Random Access Memory, or RAM. This type of memory stores temporary data like BASIC and ASSEMBLY programs and the value of variables. The data in RAM can easily be changed, and is lost when you turn your computer off. The other common type of computer memory is ROM, or Read Only Memory. This memory retains its data even when power is removed, but as the name implies, you can only read from it. No variable type data can be stored in ROM. In most computer systems ROM memory is used to store the operating system of the machine. The operating system is the program that oversees the entire operation of the computer.

In addition to RAM and ROM, the 99/4A contains another type of memory known as GROM, or Graphics Read Only Memory. GROM is unique to TI computers, as it is a type of memory produced only by Texas Instruments. What makes GROM different from other types of memory is the way it is addressed. If you think of memory as a set of storage boxes such as post office boxes, then memory addressing becomes much easier to understand. Each time the computer retrieves or stores a value

in ROM or RAM it must tell the memory the address of the requested data. Addressing GROM memory is different in that it only has to be told the starting address for a requested block of memory. When a value is read from GROM, it automatically increments itself to the next address in sequence. Our computer is configured so that all data from GROM is read through one normal CPU address. It's sort of like having a post office box that contains another set of post office boxes. This allows the computer to actually have more memory than the CPU can directly address. The only drawback is that since the CPU does not directly address the memory in GROM, it cannot directly execute programs stored there. I know this is supposed to be a review, but lets spend a little more time in tutorial mode and take a quick look at computer languages.

The only language the CPU in a computer really understands is binary code, most commonly known as machine language. It is comprised only of binary values representing on and off states. Each instruction that the CPU understands is made up of a combination of these on and offs. To program in machine language requires that the programmer himself have a great deal of memory, or the patience to look up the code for each instruction he wants the computer to execute. Fortunately, someone developed a language called assembly language to make programming easier. Assembly language uses mnemonics, or easy to remember names, for all the CPU instructions. Once a program source code is written by the programmer another program, called assembler, is run which converts the mnemonics into machine language. The machine language code produced by the assembler can then be loaded into the computers memory and executed. Although assembly language is much easier to use than machine language, it still requires an in depth knowledge of the inner workings of the computer. Somewhere along the trail of development of modern day personal computers many other languages have been developed, some for specific areas of interest, others for more general use of computers. However, all these languages were developed for the same purpose, to make it simpler to

program the computer to do the desired task. The most familiar of these languages is Beginners All-purpose Symbolic Instruction Code or BASIC. Basic is one of the easiest of all programming languages to learn, and requires very little, if any, knowledge of the computers innards. The trade-off for ease of programming in Basic is speed. Since the instructions you use in Basic cannot be directly executed by the computer, there is a middle man, known as the interpreter, who deciphers your instructions and in turn instructs the computer to execute a set of machine language subroutines that accomplish the desired task. In a lot of cases the additional time required for the interpretation of these commands is of no concern. Another trade-off is the inability to access some of the internal features of the machine, such as bit image graphics and 40 column display.

Since the computer cannot execute a program in GROM, what good is it? Well, the folks at TI came up with yet another language called GPL, or Graphics Programming Language. This is an interpreted language. The GPL language closely resembles assembly language, and offers access to all the features of the machine. I'm not sure where the Graphics in the name came from, but I suspect that it is because GPL has commands that make it easier than assembly language when it comes to printing characters on the screen. The GROM memory in our computer contains programs written in GPL language.

Where most computers have the basic language interpreter in ROM, our ROM contains a GPL interpreter. So where is the Basic interpreter? In GROM! Our Basic (and Extended Basic) interpreter is written in GPL. That explains why our Basic is slower than most other computers. The program that is interpreting your Basic programme is itself being interpreted by the program in ROM. This double interpretation takes time. So much time in fact, that some instructions for the Basic interpreter are also stored in ROM. Written in machine language, these subroutines keep our basic from being any slower than it already is. The 99/4A has the capability of accessing 8 GROM's each containing

up to 6K of stored data. Although GROM's contain only 6K of data, they always start at an even 8K boundary in the GROM memory map. This means that there can be 48K (6K x 8) of GROM data, but since there is 2K per GROM of unused space, the computer (or more correctly the GPL interpreter) can actually access 64K GROM. The unused space is not a limitation of the GPL interpreter, but a hardware design limitation of the GROM memory chip itself.

Of the possible 8 addressable GROM's, the TI console contains only three, referred to as GROMs 0, 1 and 2. GROM 0 contains what is known as the monitor, which takes care of the power up routine, and some of the mathematical functions available in the computer. GROMs 1 and 2 contain the TI BASIC interpreter. The other 5 GROM slots are reserved for use in plug in modules. Also reserved for modules is a 8K slot in the computer memory map for "normal" memory such as ROM or RAM. The TI EXTENDED BASIC module contains 24K of GROM (GROMs 3, 4, 5 and 6) and 12K of ROM. Then 12K of ROM is squeezed into the 8K available slot in the memory map by a method known as bank switching. Although there is 12K ROM, only 8K is active at any one time.

There is one other type of memory that our computer is capable of accessing. It's known as GRAM, or Graphics Random Access Memory. As RAM is to ROM, GRAM is to GROM. Unfortunately, TI didn't see fit to include any GRAM in the 99/4A. Boy, wouldn't it be nice to be able to make a few changes to some of those programs, such as Extended Basic, that are written in GPL? Well.....

GRAM KRACKER to the rescue! This fantastic little box contains 40K of simulated GRAM that resides in GROM slots 3, 4, 5, 6 and 7. The memory is actually 8K RAM chips with some fancy electronics that make it appear to the computer as GRAM. And unlike GROMs, this GRAM uses up the entire 8K sections allotted for GROM/GRAM use. The unit also contains 16K of normal RAM memory, which is bank switchable in 8K segments into the available module space. In addition, there are three empty sockets on board for optional GRAMS to be used in GROM slots 0, 1 and 2. There is another 8K of ROM

memory that contains the GRAM KRACKER operating system. This program allows saving or loading the GRAM contents to disk or cassette, and also allows you to save the contents of GROM memory in a module as well. There's also a nifty little memory editor that can be used to inspect or change any portion of RAM or GRAM, including the RAM memory in the console or P-Box. The saving of module contents is facilitated by a cartridge connector located on the GRAM KRACKER. Got a favorite module that you'd like even better if you could make a few small changes? Just plug in the GRAM KRACKER, save its contents to disk, remove the module, and load the program into GRAM. You can then use the memory editor to make the changes, reset the computer, select the program from the menu and enjoy your "new" module! and since the GRAM KRACKER memory is battery backed, you module remains intact even when you turn the computer off. Now for the details....

The GRAM KRACKER is a sturdy piece of hardware. It's housed in a metal enclosure that when inserted in the module slot extends out about as far as the Navarone Widget, and is about 2.5cm higher than the bottom row of keys on the keyboard. This makes it slightly harder to type, especially if you are used to resting your right hand on the blank section below the module slot. The contacts on the edge card that plugs into the computer are gold plated, which reduces the probability of computer difficulties due to oxidized contacts. There are 5 toggle switches on the front of the unit that are used for various reasons. Let's take a look.

The first switch is more or less an on-off switch. In the middle of its three positions is the on, or "normal" mode. The upward position is "off". This position is necessary only when using a module that does not have a rest line through the GRAM KRACKERS module connector. The Atarisoft modules are of this type. With "normal" modules the GRAM KRACKER automatically turns itself off when you plug the module in. The third position of this switch performs a reset of the computer, and is spring loaded to return to the normal position when you remove your

finger.

The next two switches are used only if you have installed the optional memory chips. The OpSys/GRAM0 switch selects between the normal GRAM 0 chip in the console or the GRAM 0 chip in the GRAM KRACKER. Since GRAM 0 contains the power up routine for the computer, any program stored in GRAM 0 must contain a routine that takes over the operation when the computer is turned on. The greatest (for me anyway) use for GRAM 0 is to modify the character set. All patterns for characters used in BASIC, EXTENDED BASIC, and most other modules are contained in GROM 0. Things like a slashed zero and a true lower case characters are possible by transferring the data in GROM 0 to GRAM 0 and then changing the pattern codes. The TI BASIC/GRAM 1-2 switch selects between the TI BASIC interpreter contained in GROMS 1 and 2 and a GPL program stored in GRAMS 1 and 2. Using these GRAMS would allow modifications to console basic, or you could replace Basic with another GPL program.

Next on the panel of the GRAM KRACKER is another three position switch. This switch is used to select between the two banks of 8K RAM in the unit, and also serve as a write protect for all memory in the KRACKER except for GRAMS 1 and 2. The 2 banks of RAM can be independently selected by this switch. Whenever a RAM bank is selected in this manner, the write protect for the unit is switched off. When the switch is in the write protect position, the RAM banks can only be switched by software. This switch has to be out of the write protect position in order to load data into the GRAMS. The loader in the GRAM KRACKER informs you when and where to position the switch for loading. The write protect feature was left off GRAMS 1 and 2 to allow this space to be used as a buffer area.

The last switch is labeled LOADER ON/LOADER OFF. This allows you to select whether or not the GRAM KRACKER loader selection appears on the main menu screen when the computer is first turned on or reset. If the loader is on, it replaces the TI Basic selection on

the menu. By selecting the GRAM KRACKER from the main menu you can load or save memory in the KRACKER, or save a module that is plugged in. The memory editor is also available from the GRAM KRACKER selection.

Now that you know a little more about the GRAM KRACKER, you're probably wondering why you need one. Most folks think it's main use is to copy modules. While it is true that you could obtain and run module software without buying the module (provided of course that you can borrow the module in order to copy it), there are a couple of reasons why this doesn't justify buying a GRAM KRACKER. First of all, module software is copyrighted, and it's against the law to copy them. Secondly, you probably already have most all the modules you'd really use, and with the falling price of cartridges you could most likely buy the ones you want for less money than you'd pay for a GRAM KRACKER. So what makes this contraption so great? I think one reason is because it allows you to enhance the modules that you already have. But not everyone is a programmer, right? Right. To modify module software, most of which is written in GPL, requires a pretty good knowledge of GPL and the internal workings of the computer. This leaves everyone out except for assembly language folks, right? WRONG! The manual that comes with GRAM KRACKER includes detailed instructions on making some of the most commonly wanted modifications to the popular modules. Things like changing the default screen colours in Xbasic, Editor/Assembler and TI Writer, modifying the TE II cartridge to operate at 1200 baud, and changing the Tax Investment and Record Keeping module to print to a PIO connected printer. Also included are instructions for some modifications to the operating system, such as skipping the title screen and menu on power-up and automatically executing item 2 on the menu. The manual I received was a temporary one, but it sufficiently detailed the operation of the unit. Also included was a utility diskette that contains the files for patching in other modifications to the Xbasic and Editor/Assembler modules. The E/A mods add 5 new load and EDIT and ASSM files into GRAM for quick loading times, and the XB mods add 5 new CALLS including an on screen

clock. This disk was also a temporary one. The final version is supposed to contain more modifications including a programme that allows you to load Extended Basic, TI Writer and Editor/Assembler all into the GRAM KRACKER at one time. Now that would be a super module! And Millers Graphics is not the only one offering instructions for modifying your favourite module.

The GRAM KRACKER has created so much traffic in the TI Forum on CompuServe that the SysOps there have dedicated a section of the database exclusively for information concerning its use. This section, labeled "KRACKER HACKERS" contains the details for such things as how to disable the Quit Key on power-up, by-passing the auto disk load upon entering Xbasic, and many others. Millers Graphics has provided much of this information, but since the Forum is a hangout for many of the brightest programming minds in the TI circle, there seems to be an endless supply of tidbits of information on neat things the KRACKER can be made to do. One really neat item provided by Millers Graphics was the code for adding a CALL CAT to Xbasic. This CALL catalogs a disk to the screen without disturbing the program, and is available without having to load in a program every time you turn your computer on!

As excited as I am about being able to modify my much used modules, I think that the greatest thing that the GRAM KRACKER has to offer is the ability to have large programs that do not occupy any of the computers RAM memory. Imagine a full featured terminal or database program with 32K of data buffer! Millers Graphics has promised a GPL assembler and programming manual soon, so it shouldn't be long before programs start to appear that take advantage of this additional programming memory. I think that eventually the GRAM KRACKER will be thought of as an almost necessary peripheral for the serious TI user. The GRAM KRACKER is available from Millers Graphics, 1475 W. Cypress Ave., San Dimas, California 91773. The price is \$US 178.95 plus shipping costs, or \$US 192.45 for the unit including the optional GRAM chips. I think it's worth the price.

FROM THE GROUPS.....

Simple Forth /John Roberts/ LA Group /Thanks to West Jax:
To set two variables A and B, add them, then print the result at (5,2):
53 VARIABLE A 15 VARIABLE B
: RUN CLS A @ B @ + 5 2 GOTOXY . ;

XB "permanent" screen colour changes:
LA Group/ Chicago group:
Set B and F to background and foreground colours of your choice (line 110) and save this program as LOAD on your disk!
100 CALL CLEAR
110 B=2 : F=16 ! your choice
120 C=16*(F-1)+(B-1)
130 CALL INIT : CALL LOAD(9
984,C,C,C,C,C,C,C,C,C,C,C,C,2,0,7,15
+8,4,32,32)
140 CALL LOAD(9999,48,2,0,8,
0,2,1,39,0,2,2,0,8,4,32,32,3
6,2,0,8,8,4)
150 CALL LOAD(10021,32,32,36
2,0,8,16,4,32,32,36,2,0,8,2
4,4,32,32,36,4,91)
160 CALL LOAD(-31804,39,8)
170 CALL LOAD(-31952,255,231
255,231)
180 END

ADDING A LOAD INTERRUPT SWITCH TO THE SPEECH SYNTHESISER:
R J Bailey/ NH99ers Group/LA Group
A load interrupt switch, for use with programs such as Danny Michaels DUMP program, is simply a normally open ("push to make") momentary contact pushbutton switch, which is wired between pins 13 and 23 of the PEB Bus. The speech synthesiser provides easy access to the bus, but you could connect the switch inside your console if you wish.
CARE: You risk damaging your console or speech synthesiser: know what you are doing!
For better results, adding a resistor and a capacitor help:

13---100 Ohm-----|SWITCH---|-----23
|---0.1uF---|

The 44 pin bus is counted as follows:

CONSOLE TOP
2 44
CONSOLE ===== CONSOLE
FRONT 1 43 BACK

CONSOLE BASE

Above is view of console looking at the side.

Even numbered pins on TOP and odd numbered pins BELOW.
Low value pins at the FRONT on the console.

Using the switch without a suitable switch causes console lock up. Pressing it twice in succession causes console lock up- hence a "noisy" switch may cause problems.
Contact debounce can be added by using flip-flop circuitry and/or in the software.

No responsibility accepted for any damage caused by user console modification!

ADDRESS: LA 99ers COMPUTER GROUP
P O Box 3547, Gardena, CA, USA, 90247-7247

COPIING FORTH:
TI's DISK MANAGER 1 module could very happily copy Forth disks, but unfortunately Disk Manager 2 does not: after "copying" with TI Disk Manager 2, you must then use Forth itself to copy Screens 1 to 9. To do this use:
Insert master disk in drive 1:
1 BLOCK UPDATE 2 BLOCK UPDATE 3 BLOCK UPDATE 4 BLOCK UPDATE 5 BLOCK UPDATE
Now insert "copied" disk:
FLUSH
Now insert "master" disk:
6 BLOCK UPDATE 7 BLOCK UPDATE 8 BLOCK UPDATE 9 BLOCK UPDATE
Insert 'copied' disk:
FLUSH.

OR use Forth entirely:
Master disk in Drive 2, blank initialised disk in Drive 1:
0 DISK_LD ! FORTH-COPY
[assumes -COPY has been loaded]

DOUBLE SIDED DOUBLE DENSITY FORTH:
Initialise a disk DSDD and copy your Forth system disk as above (NB: the initial system disk, not one with a BLOAD system on it!).
Now change Screen 3:
360 DISK_SIZE !
n DISK_HI ! [n=number of drives you have]

Forth will now recognise your disk as having 360 screens (180 on each side).

FORTH-COPY however will continue to insist your drive only has 90 screens... to amend FORTH-COPY you need to amend Screens 39,40,0, and 4.

----> continued ----->

1. Edit Screen 39: Change 90 to 360, where it appears in DTEST and twice in FORTH-COPY.
11. Edit Screen 40:
Line 3: Change 168 to 5A0
Line 4: Change 944 to 1244
Line 5: Replace line with:
DUP 10 + 2028 SWAP ! DUP 12 + 0202 SWAP ! DUP 14 + 24 0 FILL
Line 10: Change 165 to 59D
Line 13: Change 4016 to C059

Now edit Screen 33, and modify FORMAT-DISK to read:
: FORMAT-DISK 1+ 514 33616 ! 18 SYSTEM ;

All that is left is to modify the header sectors. This is done with a little Forth program. Make sure the "copy" disk is in drive one.
In immediate mode type in:
HEX 0 DISK_LD !
: DD 0 BLOCK UPDATE
DUP A + 5A0 SWAP ! DUP C + 1244 SWAP !
DUP 10 + 2028 SWAP ! DUP 12 + 202 SWAP !
38 + CB FF FILL 1 BLOCK UPDATE
DUP E + 570 SWAP ! DUP 1C + 4D20 SWAP !
DUP 1E + 5205 SWAP ! 20 + F059 SWAP !
FLUSH ; DECIMAL DD

The new header is easily copied either by using FORTH-COPY or by copying Screens 0 and 1 onto future Forth disks.

Jim Vincent/ 99ers Users Group Association/September 1984

PASCAL NOTE: Guy Stefan-Romano, 99ers Users Group, Sept 1984:
"FILE DOESN'T EXIST" after adding a UNIT to SYSTEM.LIBRARY:
In most USCD Pascals, the system is not provided with a SYSTEM.LIBRARY. TI placed parts of Pascal that would normally reside on disk/ram into chips on the card, to allow our tiny machine to run Pascal. TI also provided a SYSTEM.LIBRARY to allow use of sound, graphics, speech etc.
Because installation of a new UNIT into a SYSTEM.LIBRARY requires that a file be open while data is manipulated, the existing SYSTEM.LIBRARY data is quite vulnerable. TI added protection. You need to call a temporary file -say USER.LIB, transfer all UNITS in SYSTEM.LIBRARY to USER.LIB, then add your new UNITS to USER.LIB and if all goes well, rename the USER.LIB file SYSTEM.LIBRARY
TI were really being quite thoughtful....

FASTER FASTER TI BASIC....
Please refer to page 16 of TI*MES issue 10:
A faster still BASIC SIN routine is possible:
100 INPUT A

110 PRINT A*(1+A*A*(0.0076
*A*A-0.166))
120 BOTO 100

Now compare the time this routine takes compared to PRINT SIN(A).

As before A should fall between 0 and 1.7

If you transfer this routine to machine code, and use it instead of the inbuilt SIN function, you should see some much faster graphics (where SIN is used).

Thanks to John Stocks for this. John points out that mathematically, the two constants in the equation SHOULD be 0.16666 and 0.0084 - but once past A=1.118, the values given above actually give more accurate results.

MEMBERS DISK LIBRARY.
Stephen Shaw, 10 Aistone Road,
STOCKPORT, Cheshire, SK4 5AH

ALL DISKS LISTED ARE SOLD AS ENTIRE DISKS ONLY
CONTENTS CANNOT BE SPLIT OR MIXED
ALL DISKS ARE S.S.S.D.

PRICES: U.K. ONLY:
£3.00 per disk plus £1 p&p per order. OR send your own BLANK disk(s) and the price is:
£1.00 per disk plus £1.00 p&p per order.

OVERSEAS: By quotation only. Two IRC for airmail response please.

For a FULL library list, please send a blank disk with return post and packing.

Most recent additions:
FUNLWRITER Vn 3.1: A loader to enable you to use TI Writer with ExBas module, now extended to load most machine code programs (there are a few exceptions). From Tony McGovern and the Hunter Valley US, a magnificent program, very well written.
C99REL1 (on two disks) -at last a compiler! Input is a DV80 text file in C and output is 9900 Source Code. Does NOT load with Funlwriter at present, Editor Assembler required. Full details on library disk. Written by Clint Pulley.
UNIVERSAL DISASSEMBLER by Rene LeBlanc, this program will disassemble machine code ON DISK in any of the three disk formats- you no longer need to work out what addresses to disassemble! Can also disassemble MiniMemory and console ROM or dump ROM and VRAM. Some useful disk utilities too. Output is in a useful format, and although one or two errors have been seen, output is 95% meaningful. Useful program.

Continued..... ---->

Disk Library Continued...

KNECHT-3: More excellent music including a fast Maple Leaf Rag, 12th Street Rag, Mr Bojangles, and others. UTIL-5: A simulated processor using 3 digit decimal words, interrupt driven clock and disk catalogue for ExBas, and a RAM resident disk manager (DM-99) utilised through CALL LINKS in your own XB programs.

The following items may be paid for as above, but at the request of their authors are also available free of all costs: you need to send sufficient disks and return post and packing. NB: No additional packing is used if you select this option!

DM-99: Available on its own. See UTIL-5 for description.

PILOT-99: TWO disks. A full implementation of TI Pilot, but not in P-Code: this is written in TI Forth for Editor Assembler. TI Forth not required if you have EdAs, the program can load on its own. If you do not have EdAs, you should have TI Forth to load from MM or XB and then load Pilot by using COLD. Possibly slower than TI Basic, but an interesting language, using all the facilities of the console, including bit map mode graphics. Pilot programs are written in DV80 text files, and then interpreted by this program.

CHURCH BELLS: EXTENDED BASIC.

By Robert Davy (14) from TISHUG July 1985. An excellent program. Key this in...

```

100 REM CHURCH BELLS
110 REM ROBERT DAVY
120 REM TISHUG
130 B,C=262 :: RESTORE
140 READ A :: IF A=0 THEN RE
STORE 170 :: GOTO 140
150 FOR J=2 TO 7 STEP 1.3 ::
CALL SOUND(-999,A,J,B,11,C,1
5) :: NEXT J :: C=B :: B=A ::
GOTO 140
160 DATA 262
170 DATA 523,494,440,392,349
180 DATA 330,294,262,523,494
190 DATA 440,392,349,330,294
200 DATA 262,523,440,349,294
210 DATA 494,392,330,262,523
220 DATA 440,349,294,494,392
230 DATA 330,262,0
240 END

```

Article specially submitted by Jim Peterson:

THE POWER OF RELATIONAL EXPRESSIONS

by Jim Peterson

What the h... are those, you say? You may well ask The "blue book" that came with your computer says nothing about them, and most of the programming tutorial books on the subject are equally silent.

If you waded through the computerese and mathematiese text of the User's Reference Guide, you found them discussed on page 11-14 under Relational Expressions and on page 11-51 under IF-THEN-ELSE, but you probably didn't realize their potential. Then, you graduated to Extended Basic and found those easy-to-use, in-the-clear logical expressions AND, OR, NOT and XOR, and you looked no farther.

So, what can a relational expression do? Nothing that can't be done without it. But it can often do the job so much more compactly, so much more efficiently, and therefore so much faster!

So, let's learn to use them. And let's learn in plain English, not computerese. The following may not be technically correct, but it's the way it all works out. First, every expression has a true/false value, which is entirely different and separate from the value of the variables or numbers or strings it contains. On the TI-99/4A, a false statement has a value of 0, which is easy to remember - A FALSEHOOD IS WORTH NOTHING. Unfortunately, a true statement has a value of -1, which doesn't fit in too well!

On some other computer you may have learned that a true expression has a value of +1, but on the TI it's -1. So, in ...F=7 :: IF F=8 THEN...., F=7 has a value of -1 because obviously F does equal 7, and F=8 has a value of 0 because it is not true.

Second, when an IF refers to a variable without an "=" sign, it means "<>0". For instance, IF X THEN 1000 means "if X is more or less than 0, if it is not 0, if it is anything other than 0, then go to 1000".

Third, the computer will try to use the expression mathematically before it tries to interpret its true/false value.

Remember that everything within parentheses is worked first. For instance...X=1 :: Y=2 :: IF (X-1)+(Y-2) THEN 1000...

Since both are true, this works out to IF (-1)+(-1)<0 THEN 1000, and since -1 plus -1 is not 0, we go to 1000. On the other hand, X=1 :: Y=2 :: IF X=1+Y=2 THEN 1000 will first be calculated as X=1+Y, which comes out as X=3, and then as X=3=2, which has a true/false value of 0 (false) because X=3 has a true/false value of 0 (false), not 2!

Finally, always remember that a variable keeps its previous value until the calculation of an entire equation is completed.

X=3 :: X=X+(X+3)*X-X/X X+(X=0) is worked as X=3+(3+3)*3-3/3 3+(3=0).

More logic from Jim....

Now that you have assimilated this vast knowledge, how can it be used? The most common way is in the expression IF (X=1)+(Y=2) THEN 200. In this case, if it is true that X=1 but Y does not equal 2, then -1+0 is <0 so you go to 200. If X is not 1 but Y=2, then 0+1 is still <0, and if X=1 and Y=2 then -1 plus -1 is still <0, so you still go to 200, but if X is not 1 and Y is not 2 then 0+0 is not <0 so you do not. Of course, in Extended Basic, you could simply write IF X=1 OR Y=2 THEN 200.

If you want to go to 200 only if X=1 or if Y=2 but not if both are true, then you can write IF (X=1)+(Y=2)=-1 because either -1 plus 0 or 0 plus -1 will equal -1. In Extended Basic, this is the "exclusive OR", IF X=1 XOR Y=2.

And if you want to go to 200 only if both are true, you can write IF (X=1)+(Y=2)=-2, or more commonly IF (X=1)*(Y=2) because if either or both are not true the multiplication by 0 will give 0. In Extended Basic, this is IF X=1 AND Y=2.

And you can write more complicated versions, carefully watching your parentheses, such as IF (X=1)+(Y=2)*(Z=3) which translates to IF X=1 OR Y=2 AND Z=3.

So, if you're programming in Extended Basic, why bother with all those parentheses? Why not just use OR and AND? In the above cases, that is true. But you have not yet begun to see the power of relational expressions!

Since the true/false value is a numeric value, it can be used in calculations, and it does not have to be used with an IF statement.

For instance, this is a statement that I have used within a loop to alternate control of the two joysticks between two players....

X=X+1+(X=2)*2 :: CALL JOYSTICK(X,Y,Z) . In this, the first time around, X has not been given a value, so the equation is read X=0+1+(0=2)*2 and, and joystick #1 is activated.

Next time around, X=1 and X=1+1+(1=2)*2 gives X a value of 2, since 1=2 has a true/false value of 0. The 3rd time around, X now has a value of 2, and X=2+1+(X=2)*2 which is worked as X=2+1+(-1)*2 and then X=2+1+(-2) which is X=2+1-2 and X=1 again!

If you think that's neat, look at this one from the Airport Area US newsletter, credited to Robert Cooley : X=X=0 :: CALL JOYST(X+2,Y,Z).

Here, the first time around, X does equal 0 so the statement X=0 has a true/false value of -1 so X=-1 and X+2 activates joystick #1.

Then X=-1 so X=0 has a true/false value of 0 so X=0 so X+2 activates joystick #2...and so on! Of course, you could also write IF X=1 THEN X=2 ELSE X=1 if you prefer.

Another example:

A=INT(10*4*4*4*4):: B=INT(10*4*4*4*4):: FOR J=A TO B ...Now, if the random B happens to be smaller than the random A, the loop falls through with nothing happening. You could add a line IF A>B THEN T=1 ELSE T=-1 and FOR J=A TO B STEP T.

But why not just FOR A TO B STEP (B<=A)+ABS(A<=B) . If B<A then -1+ABS(0) gives a STEP -1 to count backwards, but if A<B then 0+ABS(-1) gives STEP 1, and if A=B then 0+ABS(0) equals STEP 0!

Here's another example : 100 INPUT "SCREEN COLOR" :S :: FOR SET=1 TO 14 :: X=SET+1-(GET)=S) : CALL COLOR(SET,X,X):: NEXT SET . That changes the character sets to colors 2 to 16 in sequence, skipping over whatever color has been selected for the screen.

Strings can also be manipulated. 100 P\$(1)="S" 110 INPUT "HOW MANY? " :N :: PRINT "THE PRICE IS "&STR\$(N)&" DOLLAR"&P\$(ABS(N)+1):: GOTO 110

Or, more efficiently 100 INPUT "HOW MANY? " :N :: PRINT "THE PRICE IS "&STR\$(N)&SEGS\$(" DOLLARS",1,7-(N)+1):: GOTO 100

However, it is also possible to overdo it. The following routine will read key input to move the cursor around the screen in all 8 directions, stopping at the borders or travelling along them if struck diagonally. However, it requires so many calculations for each key input that it is not the fastest method for accomplishing this.

```

100 CALL CLEAR :: R=1 :: C=3
110 CALL KEY(J,K,ST):: IF ST
=0 THEN 110
120 C=C+((K=B2)+(K=68)+(K=67))*C(32)-((K=87)+(K=83)
+(K=90))*C(2)
130 R=R+((K=90)+(K=88)+(K
=67))*R(24)-((K=87)+(K=69)
+(K=82))*R(1)
140 CALL HCHAR(R,C,42):: GOTO 110

```

So - for compact, efficient programming, learn to use the relational expressions! But also learn when not to use them!

LANGUAGE CORNER

Or... what does it look like?

This issue of RAMBLES has a little Forth, Basic, Logo and Source Code, but there are other languages too. Taking a brief look at sample listings may help you decide you would like a closer look, so here goes.

First a look at "c". This is a simple listing, which just fills the screen with asterisks, one at a time:

..... keep going.....-----

/* Malcolm's test program 1 */

```
#include dsk1.comio
int row,col;
main()
{ while(1)
{ row=0;
  putchar(FF);
  while(++row<25)
  { col=6;
    while(++col<35)
    { locate(row,col);
      putchar(42);
    }
  }
  if(getchar()<1) break;
}
```

This code is COMPILED by the C99 program into 9900 Source code which you can then assemble into your desired format with Editor/Assembler. Naturally, machine code is quite fast.

Notice that the variables row and col are DECLARED as INTEGERS before we use them.

PUTCHAR places a single character on the screen - FF is predefined in the COMIO file as "clear screen and home cursor to top left".

The double plus sign acts to increment the variable while locate merely locates the cursor.

C99 is based on the 40 column screen.

PILOT 99 takes the following format:

```
IT:
R: Clear 32 col screen. Cursor to top left.
LP: 24
R: LOOP 24 times
TH: :42:
R: Type and Hang
EL:
R: end loop
E:
```

(the END statement E: is essential!)

All we have done is fill a screen with asterisks!

PILOT 99 commands have one or two letters followed by a colon, which is sometimes followed by statement data. The Pilot word may also have a modifier.

Full instructions are supplied with Pilot99.

PASCAL is a rare fish, but here is what it looks like:

```
This program will backup any disk
using sector I/O; both 1 and
multiple drives and single/double
sided. Written in UCSD Pascal for
the TI 99/4A.
($L PRINTER; )
program fastback;
var
buffer:array[1..5120] of integer;
inunit,outunit:integer;
blkbase,bklimit:integer;
ch:string[9];
procedure getfrom;
begin
inunit:=0;
while (inunit <>4) and (inunit <> 5)
and (inunit <> 9) do
begin
write('Enter Source Drive # (4,5,9) ?');
readln(inunit)
end;
blkbase:=0;
while (blkbase <> 180) and
(blkbase <> 360) do
begin
write('# blocks to copy (180/360) ?');
readln(blkbase)
end;
bklimit:=(blkbase div 10) - 1;
end;
procedure getto;
begin
outunit:=0;
while (outunit <>4) and
(outunit <>5) and (outunit <>9) do
begin
write('Enter Copy Drive # (4,5,9) ?');
readln(outunit)
end;
begin
writeln('FASTBACK [V1.0]');
getfrom;
getto;
writeln('Insert disks--<CR> to start');
readln(ch);
for blkbase:=0 to bklimit do
begin
if inunit=outunit then
begin
write('Insert MASTER disk--press
<CR> when ready');
readln(ch)
end;
unitread(inunit,buffer,5120,blkbase*);
if inunit=outunit then
begin
[bit more follows..]---->
```

```
write('Insert COPY disk--press
<CR> when ready');
readln(ch)
end;
unitwrite(outunit,buffer,5120,blkbase*);
end;
writeln('Copy Complete. ')
end.
```

Taken from a US Bulletin Board

LEARNING C

C99REL1 does not have a C tuition guide with it - you need to buy a book on C!

Although C99 is only a subset of a subset of C, books on sale cover the whole C language - not to worry, all the bits we have are covered, and the rest may be added later!

I found two types of tuition guide to C - one type assumed you had knowledge of a UNIX operating system - not a lot of good to us! The other type worked by comparison to BASIC programs.

I found a book in the January sales at one third cover price:
PROGRAMMING IN C FOR THE MICROCOMPUTER USER by Robert J Traister. Published by Prentice Hall, ISBN 0-13-729641-X. 190 pages.

As this may not now be readily available, a similar book can be found in:
C PROGRAMMING GUIDE by Jack Purdum, published by Que, ISBN 0-88022-022-8. 250 pages.

Both books unfortunately start with PRINT and FOR...NEXT, not currently available with C99REL1. C99 does support PUTS and PUTCHAR, as well as WHILE. The main difference between PUTS and PRINTF is that PRINTF permits formatting commands whereas PUTS does not.

Samples:
puts("\nEntry was a digit")
printf("\nEntry was a digit")
have the same effect!
The Que book advises that printf generates more code than puts, and puts is to be preferred if the special provisions of printf are not required. A complex use of printf would be:
printf("x=%d and y=%d", 5, 8)
which could not easily be done using puts.

If you can get hold of at least the book by Jack Purdum, then together with the documentation supplied with C99, you should be able to write possibly your first "machine code" program.

From the Bulletin Boards...

From Subfile 99 on The Source...

In a Basic program you wish the user to press Y or N (or y or n). The variable XR is set to 1 by Y and 0 by N:

1. TI Basic subroutine:

```
100 PRINT "PRESS Y OR N"
110 GOSUB 10700
120 ON XR+1 GOTO 300,400
.....
10700 CALL SOUND(150,800,0)
10701 CALL KEY(3,XK,XS)
10702 IF XS=0 THEN 10701
10703 IF XK<>89 THEN 10706
10704 XR=1
10705 GOTO 10708
10706 IF XK<>78 THEN 10701
10707 XR=0
10708 RETURN
```

2. EXTENDED BASIC SUB PROGRAM:

```
100 DISPLAY AT(5,6):"PRESS Y OR N"
110 CALL YNKEY(XR)
120 ON XR+1 GOTO 300,400
.....
10200 SUB YNKEY(XR)
10201 CALL SOUND(150,1100,0)
10202 CALL KEY(3,XK,XS)
10203 IF XK=78 THEN XR=0 ELS
E IF XK=89 THEN XR=1 ELSE 10
202
10204 SUBEND
```

ROUNDING:

Set R to the number of digits required, when R is positive, numbers to right of decimal point are rounded, if R is negative, numbers to left are rounded.

DEF ROUND(X)=INT(((10^R)*X)+0.5)/(10^R)

To use this:

```
100 R=-2
110 PRINT ROUND(1101)
120 R=3
130 PRINT ROUND(10.4567)
```

Note... DEF processes quite slowly, so use it with care!

RAMBLES is subject to YOUR control. Write to tell me what you want to see and what you don't want to see. The more detailed your requests or questions the better I can meet your requirements. And if nobody writes to me, your favorite subject may not be included at all!!



A FREEWARE REVIEW OF C99

C99- A REVIEW by Stan Dixon

Clint Pulley of Ontario, Canada has implemented Small-C for the TI99. It has been released on the Freeware principle and is the best set of programs I have come across on the list! I am certainly going to write to Clint and even going to enclose a money order.

As I obtained it the programs and files are contained on two single sided disks. They all fit with a bit of spare space onto a double sided disk. The original release fitted on a single sided disk but Clint decided to issue some source files and one or two other programs and utilities also.

In order to write C99 programs you need an Editor, either the one supplied in the E/A package or TIW. If you use TIW you must save the file by using the Print option and naming a disk file as destination. (This avoids the problems caused by the format information included when you save TIW files in the usual way.) It is also necessary to be able to assemble 9900 assembly language source. Obviously the most convenient system is the Editor/Assembler module and its software. However it is possible to run that software using either MM or XB. You need special programs which are also available on the Freeware system.

{ ss: At the present time, C99 will only run from the Editor/Assembler module. It is coded very tightly - including routines in scratchpad ram for speed - and makes direct access to routines in the Editor Assembler GROM. A version for XB may appear later... keep tuned...ss}

The first step in writing a C99 program is to write the C99 source using an Editor and save it. As well as C99 source you can instruct the compiler to include other files and the library functions provided. Then using option 5 of the E/A module you run the C99 compiler. This writes 9900 assembly language source statements to a file of your choice. (That was not a misprint.. think about that idea!) It can be convenient to let the output go to the screen as you check the program for errors. However the usual destination is a disk file. One option is to include the C99 source in this assembly language file as comments.

Having done this and hopefully avoided errors, you now run the assembler against the file you have just produced with the compiler. The usual assembler options are available although you never need to specify the R option. You can now run your program. First it is necessary to load CSUP, a C99 support file, any other library files that may be needed (perhaps for file handling) and then load your object file. This is with the E/A option 3. The program name is always START in these programs. At this stage we can hope that our program works as expected.

There is a full explanation of how to run the system on the release disk and a brief explanation of how the system was implemented. Two text books are recommended and I would add "The Small-C Handbook" by James E. Hendryx to the list. The documentation does not try to teach the C language but there are several source files which are well commented and are useful as examples. There are a number of utilities on the first disk. One for locating compiler errors is of obvious use in developing programs. There is one undocumented utility. It is called OPT and appears, from comments in the source, to be a code optimiser. That is it reads the output from the compiler and chops out duplicated code. This shortens the files for assembly and could be vital if a large application is being attempted. I have used it and it worked and the optimised program still ran! Clint has included a disk directory program which works extremely quickly and can be useful within the E/A environment.

Also provided are two files that permit us to save our working programs in program format. This is explained a little obscurely in the manual but I have managed to make that work. One thing that struck me as peculiar was the fact that for one small program I ended up with three large program format files. However Stephen assures me that there is nothing to worry about. Apparently if the program reserves large areas of memory when running this affects the way the Save utility works. This useful program means that it should be possible to write C99 programs and save the runnable versions to tape.

{ ss. yes you can and it works! AND we have a public domain program which enables ExBas to load these cassette files too.... ss}.
With the availability of loaders for XB and MM we can see greater prospects of machine programs for the unexpanded machine.

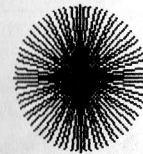
The second disk contains a source file for the directory program on the first disk and the source and object for a TIW loader for the MM module. Also included are XB and MM, E/A versions of a BREAKTHROUGH game that is well documented and very entertaining. I certainly enjoyed playing with it though I usually find Adventure games are more my style! To move on to a C99 application which is supplied on the second disk. This is the source and object code for a text formatter. The program is called RUNOFF and allows the printing of TIW files and other DV 80 text files without the need for embedded commands. It works nicely and lets you offset tables within the text in nofill mode.

{ ss... RUNOFF has a significant advantage over TIW: the text file is not stored in sonsole ram, so a text file of 300 sectors may be printed: useful if you have created a mammoth text file by downloading from a bulletin board

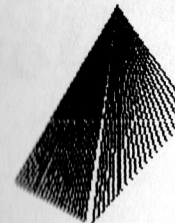
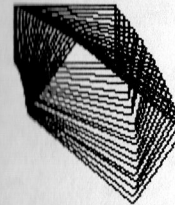
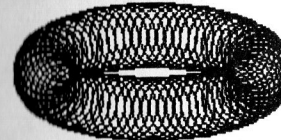
This version of Small-C does not have some of the usual features but it is likely that later revisions will have. I am sure that a positive response to Clint Pulley from some of the people who obtain this program will ensure further development. Stan Dixon.

RTA
HARDWARE
16 LINDEN WALK
NORTH BADDESLEY
SOUTHAMPTON
HANTS SO5 9DH
TEL: 0703 732801

NEW EXTENDED BASIC 2 PLUS



TEXAS TI 99/4a



Yes now in stock, Direct from West Germany, probably the most exciting module produced since TI pulled out of the home computer market. This 44K module contains the complete TI Extended Basic program PLUS 60 EXTRA STATEMENTS!! 20 of which are available with just the console only, and if you have 32K expansion then all of the new statements are available to use in programs.

STATEMENTS NEEDING ONLY THE CONSOLE ARE :-

BHCOPIY - a bit image screen dump.
VPEEK - allows reading VDP RAM.
VPOKE - allows writing to VDP RAM.
GPEEK - allows reading of console GROM'S.
ALLSET - resets ASCII characters 32-126.
WAIT - allows you to set a delay.
MOVE - used to move contents within RAM.
MSAVE - saves segments of RAM to DISK etc.
MLOAD - loads program files into RAM.
QUITOFF - disables QUIT key.

And there are more but the above illustrates the extra power available with just the Module and no memory expansion.

STATEMENTS AVAILABLE WITH 32K EXPANSION :-

CIRCLE, ARCUS, ELLIPS, RECT, TURN, TURNT0, AXIS, DISPLAY, ACCEPT, MOVE, REMOVE, SHIFT, WINDOW, SETCOL, CENTRE, RESET, SETTO, IFSET, INVERT, MOVETO, REMVTO, And many more allowing easy drawing of graphic screens as shown. You also have statements that enable you to SAVE and RECALL screens to DISK. YES Extended Basic 2 Plus now enables you to draw circles, squares, block graphs, line graphs easily, No defining character squares and timewasting with graph paper. And remember it has it's own :-

BUILT IN SCREEN DUMP COMMAND.
AND IS COMPLETELY COMPATIBLE WITH STANDARD EXTENDED BASIC PROGRAMS.

PRICE £ 95.00 inc postage and packing.

EW DAY Price List NEW DAY Price List NEW

Item	usual price	our price
4FRONT Magazine(quarterly)	£	£
Single Issue	cassette	3.50
	disk	4.50
Year Sub - 4 FOR PRICE OF 3!	cassette	10.50
	disk	13.50

BOOKS: HALF PRICE (if you're quick!)

TI99/4a Users Reference Guide	7.20	3.60
Beginners Basic	6.20	3.10
Programming Basic	16.95	8.48
Learning to use the TI99/4a	4.95	2.48
Initiation into Assembly Language	16.95	8.48
TI99/4a Technical Data Manual	8.97	4.48
Editor Assembler Manual	9.95	4.97
Tantalizing Games	3.95	1.97
Time Lost	5.95	2.97
Feel at home with a Home Computer	12.95	6.47
Favourite Programs Explained	9.75	4.87
Getting Started with the TI99/4a	5.95	2.97
Mastering the TI	5.95	2.97
TI99/4a Games Programs	8.85	4.43
Using and Programming the TI99/4a	7.85	3.92
Get more from the TI99/4a	5.95	2.97
Kids and the TI99/4a	17.95	8.97
Sprites, a Turtle, and TI Logo	13.45	6.72
Computer Playground	9.95	4.97

SOLID STATE SOFTWARE

Extended Basic 2 Plus!	95.00
As original EB, but with 60 exciting EXTRA statements. Ring or write for info.	9.95
Video Chess, Parsec, each at	8.95
Munchman, 5-a-side Soccer, each at	7.95
Wumpus, Invaders, Blasto, Attack, Hustle	3.95!
Household Budget Management	3.95!
Beginning Grammar	3.95!

CASSETTE SOFTWARE

NEW! The Puzzler - superb.	4.95
Oldies but Goodies 1	3.95!
Beginners Basic Tutor	1.25!

DISK SOFTWARE

Details on application	
------------------------	--

JUST WHAT IS PUZZLER
 (we're not telling)
 you didn't think TI Basic could do this...

DON'T call it a day on TI
 without calling on Newday
 for discounts
 on Amstrad, Atari, etc...

New Day COMPUTING
 TEL 0404 41856
TEXAS SPECIAL!

NEW FRONT

NEW magazine

HALF PRICE TEXAS BOOK GOLD RUSH.

NEW EB2+ at last!

Household Budget Management
 £3.95

more new titles soon!

DAY Price List NEW DAY Price List NEW DA

Item	usual price	our price
DOT MATRIX PRINTERS		
MANNESMAN TALLY MT80+	219.65	199.95
100cps - Friction/Tractor		
EPSON LX80 100cps draft	293.25	259.95
16cps NLQ - 1k buffer		
CITIZEN MSP-10 - 1k buffer	454.25	409.95
50cps NLQ - 200cps draft		
many others available - please ask		

DISKETTES

New Day Disks SSSD 10 in library case	12.95
Parrot SSSD 10 in library case	19.95
Memron DSDD 5 coloured Red, Yellow, Green, Blue or White	10.95
many others available - please ask	

DISK

Mini Disc 40	22.65	19.95
Mini Disc 80	27.60	24.95
Data Caddy - SPECIAL PRICE!	50.00	24.95
(holds printer/paper)		
Binders (10 pack) SPECIAL PRICE!	30.00	19.95
(for various sizes paper)		
many more consumables available		

STOP PRESS..announcing 4MART..2ndhand TI99/4a
 hard and software, peripherals, etc. bought and sold. Ring for details, or to register 'wants'.
 eg Computers, joysticks, cass. cables (new AND used)
 Epson P40 ser. (mint), speech synths, minimemory.

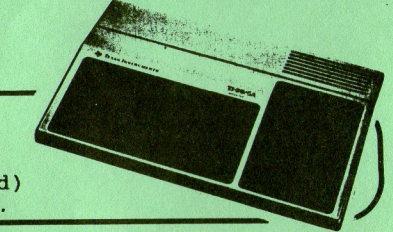
LATE NEWS...COMPLETE Expansion System - £450

NEW DAY, Jerrard Close, Honiton, Devon. EX14 8EF - Tel (0404) 41856

New Day COMPUTING
 TEL 0404 41856
TEXAS SPECIAL!

MANNESMAN TALLY
 Authorised Stockist

REXEL
 Data Caddy



4 FRONT

A FIRST - AT LAST! An exciting NEW development for neglected TI addicts. 4FRONT - not just another magazine, but a magazine on CASSETTE (or disk). No more keyboard crunching, folks, all your programs will be ready to run. (listings will be available if you want them!). 4FRONT will be strictly objective and unbiased in all its content, which will include reviews hints, tips, articles. We don't claim 4FRONT to be the best thing since sliced bread, but it will be just as fresh! Taste and see....

*** Please note that all prices (except books) include delivery and VAT. ***

add £1 for p&p (if order includes books)

4/a hard and software bought and sold.