TI*MES

TI99/4A USERS GROUP (U.K.) CONTACTS.

Chairman: Gordon Pitt Tel.0922 476373
259 Sneyd Lane, Bloxwich, WALSALL, West Midlands. WS3 7LS
General Secretary: Jim Ballinger Tel.0332 772612
5 Offerton Ave., DERBY. DE3 8DU
Membership Secretary & Telecoms:(TI*MES back rumbers) Tel.0707 873778
24, Bacons Drive, CUFFLEY, Herts. EN6 4DU
Treasurer: Alan Rutherford Tel.0625 524642
13 The Circuit, WILMSLOW, Cheshire. SK9 6DA
Publicity Officer: Phil Trotter Tel.0642 817356
80 Martonburn Rd., MIDDLESBROUGH, Cleveland. TS4 2TH
Programming: Mark Wills Tel.0743 64177
37 Abbots Rd., Monkmoor Estate, SHREWSBURY. SY2 5PZ
TI*MES Editor & Distribution: Alan Bailey Tel:081 508 1053
14 Shelley Grove, LOUGHTON, Essex. IG10 1BY
Librarians:Cassette: Nicky Goddard Tel.0978 843547
Sarnia, Cemetary Rd.,Rhos, WREXHAM, Clwd. LL14 2BY
Disk: Stephen Shaw(also Journal Exchange & V.P.)
Tel. 061 432 6097 after 8pm (letters preferred)10 Alstone Rd.STOCKPORT
Cheshire. SK4 5AH
Hardware & Projects: Mike Goddard Tel. & Address as Nicky Goddard above.
Modules: Edward Shaw Tel.0538 360382(5pm to 8pm)
Crow Holt Farm, Basford, LEEK, Staffs. ST13 7DU
Publications: Mike Curtis Tel. 0209 219051
21 Treliske Rd., Roseland Gdns., REDRUTH, Cornwall. TR15 1QE

MAGAZINE CONTENTS

ISSUE NO 29

SUMMER 1990

EDITORIAL

Here is another interesting issue of your magazine for you. There is a report from Stephen on what was obviously a very good annual show in Chester. We must all be grateful not only to the organisers, but also to all those who took the trouble and risk of transporting delicate and heavy equipment for the many demonstrations which were mounted. There were some difficulties in hearing the PA system, and the General Secretary is asking the members of the Committee for help to complete the Minutes. They will appear in the next issue of the magazine.

Peter Walker is making a special offer on some back issues of the magazine. They are normally £2 each, and nos.4,8-14,16,18-20,& 22-27 are available. To reduce the larger stocks however 5+ of them can be had for £1 per copy. This applies to nos.8-11,14,16,18,20,22,25 & 27.

You will see that we have a new contributor, Trevor Stevens, whom we welcome. We are also fortunate in that Ken Wilford of 27 Wyndham Rd.,Blacton, CHESTER, (not Cheshire) CH1 5SE,has kindly agreed to print articles for the magazine, from disk,if this will help you with your contributions. This does seem to represent a hopeful development for the future in desktop publishing!

You will see from the Contacts list that we have to welcome a new Cassette Librarian and a new Publicity Officer. Thank you both for your readiness to help keep our Group alive!

DISCLAIMER

Views expressed in this magazine are those of the individual contributor, not of the Committee.

We try to acknowledge original sources, but please accept our apologies for any we miss.

NEXT COPY DATE

All copy for the next issue should reach the Editor by 1st. September.

Please use a new ribbon, and clean your printer! A4 sheets are a great help, and the print should leave 15mm margins top and sides, and preferably 20mm at the bottom.

CASSETTE LIBRARY REPORT......

NICKY GODDARD

At the recent A.G.M. I was elected cassette librarian and as it will take up more of my time I am handing over the cassette reviews to Mark Wills of Shrewsbury. He will take over as from the next issue of TI*MES.

To order cassettes please contact me at the following address.

'SARNIA',
CEMETERY ROAD,
RHOSLLANERCHRUGOG
WREXHAM,
CLWYD
LL14 2BY
TEL.(0978) 843547

SPECIAL OFFER SPECIAL OFFER SPECIAL OFFER SPECIAL OFFER SPECIAL OFFE

BUY ANY OF THESE SURPLUS STOCK CASSETTES FOR ONLY 50p EACH OR 6 FOR £2.50

UTILITIES -

GAMES WRITERS PACKS 1+2, STARTER PACKS 1+2, TEACH YOURSELF EXTENDED BASIC, PROGRAMMING AIDS 1, BEGINNERS BASIC TUTOR, TEACH YOURSELF BASIC PARTS 1+2, TOAD GRAPHICS.

GAMES -

SOCCER SUPREMO, FUNPACK 1+2, ROBOPODS, BLAST IT, HAUNTED HOUSE + WUMPUS, OLDIES BUT GOODIES GAMES I + II, MANIA, SANTA AND THE GOBLINS,KAT TRAXX, ADVENTURE MANIA, LIONEL AND THE LADDERS, PENTATHALON, SUPER FROGGER, WINGING IT, BATTLESHIPS, HOP IT, GAME PAK III, 3D RACE, FROGIE, TREASURE QUEST + FOUR IN A ROW, COMPUTER MUSIC BOX, RUNNER ON TREITON + OTHELLO, GOBLIN CAVES, DRAGON COMBAT, LASER TANK, ATTACK MAN, PENGI, KONG.

ADVENTURE TAPES -

PYRAMID OF DOOM, VOODOO CASTLE, GHOST TOWN, THE COUNT, STRANGE OYDESSEY, PIRATE ADVENTURE.

CASSETTE REVIEWS....... NICKY GODDARD

All of the games reviewed here are available from the group cassette library at the current library terms.

STAR RATING GUIDE.
One star = terrible,Two stars = "OK",Three stars = quite good, Four stars = very good, Five stars = Brilliant.........

DARTS LIBRARY NO. G49

To begin the game you are asked how many players there are and then it asks you the players name(s).

Then it asks you if you want to throw the dart above or below the center line. You type 'U' to throw above the center line and 'D' to throw below the center line.

Then you are asked what height you want to throw the dart Ø is center and I've worked out by experience that Ø4 is the square below the center and Ø8 is the square below that.

Next you are asked the speed, the higher the number the further the dart will go.

At the end of the game you must score what the computer says you have to score other wise the computer prints out on the screen 'YOU BUSTED' and 2 tones will sound and you have to try again.

Please note that I have only reviewed the one player 3Ø1 game. There are 2 more games which are around the board and cricket.
A very very good Extended Basic keyboard game.
STAR RATING *****

COMPUTER CARD LIBRARY NO. G6Ø

This game is called 'COMPUTER CARD' for a very special reason.

After the title screen the computer sets out 52 cards with numbers by the side of them in numerical order.

Next it asks you your name and then it asks you whether you want to change the computers name, it will not become a 2 player game if you do.

The object of the game is to try to get as many pairs of numbers as possible e.g. 9 of diamonds and 9 of clubs, or 3 of hearts and 3 of spades, by choosing two cards by their numbers each time your turn comes. The one who has the most pairs of cards wins.
A quite good Extended Basic keyboard game.
STAR RATING ***

DEACTIVATION LIBRARY NO. G54

First of all you are asked if you want instructions. Then you are asked if you want a random timer, if you don't you can set the timer yourself.

The object of the game is to run over the mines deactivating them as you go in less time than you have set yourself or as the computer has set for you. If you don't the mine(s) that are left wil blow up and the game is over.

You score 1Ø points each time you run over a mine. At the end of the game whether you have won or not the computer asks you if you want to play again.
A good Extended Basic keyboard game.
STAR RATING ***

PEEKING AND POKING
By T.STEVENS
c1990

When I first bought my machine back in the early 80's I only had the basic machine. I programmed in the TI BASIC with some success. However I looked at some of the other machines and saw that they were able to use the calls PEEK & POKE to go direct to the screen without going through the BASIC interpreter, which on the TEXAS is slow compared to many of its rivals at the time. This made me a little envious but the TI graphics made up for it.

I then found out from the many books I read that there was a way of using POKEs and PEEKs - the Editor Assembler and the MINIMEMORY. Unable to afford the vast expense of the Disk drive, PEB etc, I went for the little MINIMEMORY. When I got it I realised that it was not so small as Texas Instruments made out; that's another story.

Any how after looking at the book that came with the module and being lent by a kind friend the Editor Assembler Manual I sorted a few things out and started to programe. I wanted to get things on the screen first, so on reading up fully I found that the command for this was CALL POKEV. I found that I could put a numerical expression or a variable into the VDP locations 0 to 16383 that's HEX >0000 to >3FFF.

I then discovered that I could then put in a "Poke" list as it's called, all in one statement. You do this by putting in your start address, that's the first number, then you put in value for that location, then the value for the next location and so on.

EG. CALL POKEV(START ADD,A,B,C)

You can also change to another location in that list by stopping your consecutive list with the "" marks. You then put in your new address location and then write again to it. You can see how this is done by this example.

EG. CALL POKEV(700,20,20,"",2,20)

Easy isn't it? If you put your values above 16383 or HEX >3FFF things get lost, the processor gets a headache and the computer locks up. Just turn off and start again.

Now we know how to do the lists, what do we do with them? Well we can put some things on the screen for starters. If you have a look at your minimem book that came with the module you will see on page 76 a location list for the Basic operation mode. Look at the top of the page and you will see the box "SCREEN"; it gives you the location in hex and dec for the screen. The map as it's called, goes from column 1 row 1 which equals 0 through to location 767 which is row 24 column 32. It runs across like this:-

```
00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 ......31
32 33 34 35 36 37 38 39 40 41 42.......................63
64 65 66 67 etc through to 767
```

Try this - CALL POKEV(18,161).....This puts a letter "A" at Row 1 Column 18. Just like Call Hchar(1,18,65).

Go on, have a go at putting "161" in the values between 0 and 767, maybe using a loop if you like.

Ok, now we know how to put a letter on the screen we must now know how to work out the letter codes. This is easy, all you do is off-set the ASCII character number by adding on 96 ie the letter "A" is ASCII code 65 + 96 = 161, hey presto! This means that you can write on the screen like this. Try CALL POKEV(365,72+96,69+96,76+96,76+96,79+96) Did you see how fast it did it? This is because you wrote direct to the processor.

So now we can write to the screen I suppose we could define those characters and have them on the screen that way. That's slower in programe form as the call chars have to scan in the first place. However once in programe, not a lot is gained. If you do want to speed up slow TI BASIC programe scan then you can define your character direct from the CALL POKEV. This is done by first working out your pattern poke list. You do this on this little grid that follows.

```
 128 64 32 16  8  4  2  1      total
  :  :  :  :  :  :  :  :  :
  :  :  :  :  :  :  :  :  :      0
 ---------------------------
  :  :**:  :  :  :  :  :  :
  :  :**:  :  :  :  :  :  :      64
 ---------------------------
  :  :  :**:  :  :  :  :  :
  :  :  :**:  :  :  :  :  :      32
 ---------------------------
  :  :  :**:  :  :  :  :  :
  :  :  :**:  :  :  :  :  :      32
 ---------------------------
  :  :  :**:**:  :**:  :  :
  :  :  :**:**:  :**:  :  :      52
 ---------------------------
  :  :  :  :  :  :  :**:  :
  :  :  :  :  :  :  :**:  :      2
 ---------------------------
  :  :  :  :  :  :  :**:  :
  :  :  :  :  :  :  :**:  :      2
 ---------------------------
  :  :  :  :  :  :  :  :  :
  :  :  :  :  :  :  :  :  :      0
 ---------------------------
```

All you do is fill in the boxes to define the pixels you want turned on in your 8 x 8 box as in the CALL CHAR statements. You then add up from left to right using the top code numbers. This patterns poke list would be 0,64,32,32,52,2,2,0.

Now we have to match it with the ASCII character to change. We do this by knowing the ASCII start location. If we again look at Page 76 we will see that they start at 1024, this equals the SPACE or ASCII code 32 character. The table goes on up to the value of 1535. We can use a formula to work out your location which is :-(ASCII+96)*8 = CODE.

So if we call POKEV(1024,64,32,32,52,2,2,0) all our spaces will equal your pattern in the grid shown.

Try this:-

```
100 CALL CLEAR
110 CALL POKEV(1024,64,32,32,52,2,2,0)
120 GOTO 120
```

As you can see your screen is covered in your little pattern.

Ok! now we want a bit of colour on your character. Yes we can do that as well...

As you most likely know your colours are in sets from 1 to 16 (See your User Manual for full explanation if you are not sure). Each SET defines "color" to those ASCII numbers in that SET via the basic CALL COLOR routine. If we again look at page 76 of the MINIMEM book you will see the box "Color & Sprite Table" This gives the locations of 768 to 799. The colour table we want starts at location 784 and runs through to 799. Therefore it looks like this for the set allocations:-

SET 1 = 784
SET 2 = 785
SET 3 = 786
ETC..... TO
SET 16 = 799

We now have to work out the colour combinations for the Poke list number. This is done by another fomula (Foreground-1)*16+(background-1). So if you look at your user manual and look up say Black for the foreground and Light Blue for the background the value would be via the formula 21. So if we add to our program 115 CALL POKEV(784,21) you will get your desired colours, as 784 represents set 1 which holds your space ASCII code of 32. Easy isn't it?

We mentioned the Sprite Table just a bit back and yes you can run Sprites. The table for them starts at 768 running up to 783. Each Sprite requires 4 bytes to define and a stop value of 208 must be added. You will see by the mathematics that only 3 Sprites can fully be defined. This is because the basic interpreter does not recognise the existence of sprites. However three is better than none. Each Sprite as we have said has four bytes of information; the first two are for the X & Y positions on the screen in pixels for its start location, the next is the ASCII character used off-set by 96 as before in the chars, and the forth byte is the colour of your sprite, with the -1 colour off-set. We then put in the 208. This however only needs putting in once as can be seen with the programe below which displays 3 sprites of different colours.

```
110 CALL CLEAR
120 CALL SCREEN(2)
130 CALL POKEV(768,80,20,161,6)
140 CALL POKEV(772,80,60,162,15)
150 CALL POKEV(776,80,120,163,12,208)
160 GOTO 160
```

Now we have them on the screen, let's move them. All we do is poke a new X or Y value in to renew the sprite position.

Try this....

```
        40 CALL CLEAR
        50 A=80
        60 B=20
        70 CALL POKEV(768,A,B,161,6,208)
        80 A=A+1
        90 B=B-1
        100 CALL POKEV(768,A,B)
        110 GOTO 80
```

So there we are - Sprites in basic. You can however operate the sprites in automatic motion, but that takes some explaining and we will leave it here for now.

Now we get to the PEEKS. There are two types that can be used, They are:- CALL PEEK and CALL PEEKV. The first of these CALL PEEK allows you to look at the CPU RAM from programe. It allows you to look at address from -32768 to 32767 and 0 to 32767 which is 0000 to 7FFF and 8000 to FFFF. To access the higher numbers you subtract 65536 from 32767. The CALL PEEK gives you exactly the same power and format of use as the CALL POKEV we discussed at the beginning of this article.

IE. CALL POKEV(8192,A,M,C(10)""-24679,X)

CALL PEEKV is exactly the same as CALL PEEK except it only looks at the VDP RAM

Both these CALL's read data from the memory and are very useful for picking up data from one location and transporting a duplicate to another location. The call PEEKV can also act as a CALL GCHAR as it will look at your screen locations and return the character in that location.

IE:- CALL PEEKV(18,X) returns the character in Row 1 Col 18.

Well, I hope this article has stirred a few thoughts which can be used in MINIMEMORY to speed up TI BASIC.

Just to whet your appetite here are some PEEKS for immediate use.

1. Check if speech synthesizer is connected.
   CALL PEEK(-28672,X)   If X=96 the YES, If X=0 then NO.

2. Stop Program and return to the master screen.

```
   10 CALL PEEK(2,A<B)
   20 CALL LOAD(-31804,A<B)
```

Have fun............

MODULES MODULES MODULES MODULES

APPEAL TO ALL DISK DRIVE OWNERS........

Do you have any modules that you would consider selling or donating to the module library. Reasonable prices paid. For more information please contact me at the address given below.

The latest list of modules available for purchase follows! please note that as a result of the annual general meeting cheques for modules should be made payable to "E.H.SHAW".

ADDITION AND SUBTRACTION 1	2.ØØ	PARSEC	3.5Ø
ADVENTURE and PIRATE TAPE	5.ØØ	PERSONAL REPORT GENERATOR	3.5Ø
ALLIGATOR MIX	4.5Ø	PROTECTOR	4.5Ø
A-MAZE-ING	4.ØØ	SHAMUS	3.5Ø
BEGINNING GRAMMAR	3.ØØ	DEFENDER	4.ØØ
BIGFOOT	4.ØØ	TUNNELS OF DOOM	4.ØØ
CHISHOLM TRAIL	3.5Ø	OTHELLO	4.ØØ
CONNECT 4	3.5Ø	THE ATTACK	3.5Ø
* DISK MANAGER I	3.ØØ	TI INVADERS	3.5Ø
* DISK MANAGER II	4.ØØ	* TI LOGO II MODULE ONLY	5.ØØ
EARLY READING	2.5Ø	* TI LOGO I + CASSETTE	
EARLY LEARNING FUN	2.5Ø	+ MANUAL 32K REQD.......	12.5Ø
* EDITOR ASSEMBLER (NO MANUAL)	22.5Ø	* TI MULTIPLAN/DISK/MANUAL	25.ØØ
* EXTENDED BASIC AND MANUAL	22.5Ø	* TI WRITER NO MANUAL	5.ØØ
INDOOR SOCCER	4.ØØ	TERMINAL EMULATOR II	5.ØØ
MULTIPLICATION 1	4.5Ø	TOMBSTONE CITY	2.5Ø
MUNCHMAN	4.ØØ	VIDEO GAMES 1	3.5Ø
MUSIC MAKER	5.ØØ	YAHTZEE	4.ØØ
NUMBER MAGIC	3.5Ø	ZERO ZAP	4.5Ø
DEMONSTRATION	3.5Ø	Cassettes for adventure	
		module(please write)	5Øp each.

* MODULES MARKED WITH AN ASTERISK REQUIRE DISKS OR 32K RAM OR BOTH. ALSO PLEASE NOTE THAT EARLY READING NEEDS A SPEECH SYNTH TO RUN.

PURCHASING MODULES FROM THE LIBRARY

You may return any module purchased within four weeks and be refunded the purchase price less postage which will be charged at the rate of 4Ø pence per module.

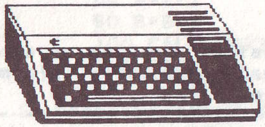
Application to loan/purchase modules.

Name:.................................. Modules required:............
Address:,.............................. ....
....
....
....

I enclose cheque/PO for £......(as indicated on the list) & post to

PLEASE MAKE CHEQUES PAYABLE TO E.H.SHAW.
Foreign orders can only be accepted if a
BANKERS DRAFT is enclosed drawn in STERLING
on a LONDON bank. It also helps if a little
extra is added on for postage overseas.

MR. E.H. SHAW
CROW HOLT FARM
BASFORD
LEEK
STAFFS. ST13 7DU



NOTES FROM A 99ER-HOLIC

BY PETER WALKER

I hope all you "console-only" members will forgive me this quarter for not contributing my usual column in TI*MES. I expect it will be back next issue. So what have I been up to? All of a sudden Alan Bailey phones me up to ask where my contributions are and I realise that with everything else on, I have overlooked the TI*MES press date.

On the membership front "June 1990" renewals are coming in thick and fast. We welcome new members since last issue: T Stevens, R Williams, Louise Goode, A Wyard, O Fouillen (in France), and a welcome return to the fold to Richard Blanden who actively supported the group and the machine in earlier years.

Many of you will know Dave Hewitt: he has asked me to publish his new address - 1 Stadhampton Road, Drayton St Leonard, Wallingford, Oxfordshire, OX10 7AR. (Tel 0865 890066).

Family life has partly kept me away from the machine. My wife is expecting our second child in July so we have been undertaking a lot of decorating and having a new kitchin fitted before the arrival. Also, come April, I am always busy settling up the accounts for the local branch of the RSPB, for which I have been Treasurer for the last 8 years. I have now decided to give this up, but I shall carry on as Membership Secretary of TIUG(UK). Keeping accounts is an excellent job for the TI99 and I have developed my own bookeeping program for this task, using Oaktree's Display Enhancement Package to provide a nice 40 column display and other useful features, such as 'instant' screen switching and 'entry trigger' on a filled field.

I have acquired a good deal of new kit for my system in the last 6 months. I bought a second Horizon Ramdisk from an old member of the group, Tony Bowden. This has allowed me load more commonly used software on the second Ramdisk, such as TI-Base and TI-Artist. Then I was persuaded by Gordon to buy a Mechatronics 80 column card. There's a lot that can be said about this item; perhaps it needs an article of its own. I've hardly had time to exploit all its possibilities, but 80 column TI-Writer (FWeb) is a joy to use since there is no need for windowing; the new Diskreview package is also very effective in 80 columns. Of course moving to 80 columns demands a monitor in place of the TV. This gives very good definition but even this is not 'perfect' and I found that the normal White on Blue default colour that many Text Mode packages use caused slightly fuzzy characters, so wherever screen colours can be changed, I have altered the colours to White on Dark Red, which is very clear if a little dazzling at first! The 80 column card is not without its drawbacks. I have found some packages that won't now work with my machine (but thankfully I have a spare unmodified console). I have found problems with GraphX, Page Pro 99, and even Parsec looks odd though is still playable. Page Pro is a blow since I like this package and have used it for article headers in recent issues of TI*MES. It loads and edits OK but any disk access for pictures or screens causes a crash. Gordon who has a Myarc disk controller, reports no problems, so perhaps its a combination of 80 column card and TI Disk Controller. Can any other 80 column owners help resolve this? The makers of Page Pro do not know of a problem.

I know there are some excellent support packages for the 80 column card around, such as Alexander Hulpke's X80 and XHi. Nevertheless, for 80 column owners who just want some simple ExBas support, I have developed some simple subprograms that make handling the 80 column screen easier and you can find these at the end of this article.

I have also bought a Mechatronics Mouse and have spent a lot of time understanding this and its quirks. I soon decided that the Assembly Language routine provided with the Mouse was flawed, in that the second Mouse key, intended for homing the Mouse to a desired screen position, didn't work. I ended up disassembling the code, and with a little help from a different source code also provided, managed to write a new driver routine without the homing bug. For Assembly freaks the problem was caused by trying to use MOV to move 2 bytes (one word) in one instruction. Unfortunately the two bytes were not on an even word limit causing the MOV to operate on the wrong bytes. The cure was to use two MOVB (move bytes) instead. I also found what looked like a second bug in decoding the Mouse movement, in particular the code which separates positive from negative motion. It is indeed a bug, and you can see this even with the TI-Artist driver, where rapid negative movement (left or up) causes some funny effects. On curing this problem however, I realised that it was a deliberate compromise. As long as the Mouse movement isn't too fast it works OK. My modified code however causes strange movements if the Alpha Lock key is down. The point is that the Mouse Key 2 uses the same arrangement as the Joystick Up position which as many of you will know doesn't function with the Alpha Lock down. The code as supplied got over this problem. In the end, it persuaded me to fit the diode modification to my keyboard to remove once and for all the Alpha Lock problem. I have deposited copies of the amended Mouse software with our disk library. If you came to the AGM you will have seen the Mouse in action and very nice it is too, allowing real artistry with TI-Artist in a way that the joysticks or keyboard never could.

Many of you will know of the East Anglia Region 99ers. They are not restricted to East Anglia either! Our Publications Librarian, Mike Curtis, down in Cornwall, edits their monthly magazine and he asked me to write something on telecommunications since there is a solid core of Prestel users in the EAR99ers. I've written two articles: one on modems, as an introduction to the subject and a second one on using the TI99 for accessing Prestel, BT's videotex service. Given that a number of TIUG members are also telecoms minded, I have asked Alan Bailey if he can reprint in TI*MES too, so you may see these elsewhere in this or later issues. Apologies to members of both groups for whom it will not be new. Thinking more and more about Prestel, I became determined to try to understand Rob Tempelman Plat's "Viditel" package, used to access Prestel, to see if I could remove some irritating bugs it has. Many years ago I overwrote the Dutch screen instructions with English. Now I decided to investigate the character definitions for the screen display. Looking through the program image file, I eventually located defintions for all the characters from the cursor right through to DELete (127), all the graphics characters (contiguous and separated), and a whole load of European accented characters not used by Prestel. I'm not sure how these are used. Anyway, I was able to locate the definitions for the characters that are special to Prestel and modify them to their intended display shape. For example the square brackets [] become left and right arrows. For all Prestel users who want this newly modifed version, I have deposited a copy with both disk libraries (TIUG and EAR) along with some other Prestel based programs I have developed in recent months. These require the 'split baud' RS232 DSR chip developed by Colin Hinson so that one can operate in Basic but with the RS232 port using the special 1200/75 speed. These programs include: Auto logon, File Upload and TI-Artist Instance to Prestel Graphics conversion program.

Finally, I have been preparing for the AGM, which will be long gone by the time you read this. For those who attended, I hope to have demonstrated: Telecoms, Databases, Horizon Ramdisks, 80 column card and Mouse, though whether time will allow justice to all these subjects remains to be seen!

Peter Walker

In the following program, lines 100-200 demonstrate some of the sub programs at lines 30000 onwards. These can be merged with any program you are writing for the 80 column card. Here is a short description of what the sub-programs do.

SUB CLEAR80 - Clears the screen

SUB POSN(R,C) - Positions the cursor at Row R Column C

SUB DSPLAY(R,C,D$) - Displays D$ at Row R Column C

SUB ACCEPT(R,C,L,A$,K) - Moves cursor to R,C and blanks a field length L; accepts A$ at R,C. K is value of entry character, ie 13 for "Enter", 6 for "Redo", 10 for Down Arrow.

SUB COLOR80(F,B,FB,BB,BL) - Sets colours for normal and blinking Foreground and Background. If BL=0 then blinking rate will be set such that there is no blinking, thus allowing a second colour character set.

SUB SBLK - Converts blinking characters to normal.

SUB ACCEPTD(R,C,L,P$,D$,A$,K) - As ACCEPT, see above, but with the addition of a prompt string P$ ahead of the input field, and a default input string D$.

SUB COLINV(F,B) - Sets normal colour to F,B; and non-blinking "blinking" text to the inverse B,F

SUB BOX(R,C,R2,C2) - Draws a Box with top left coordinate R,C and bottom right coordinate R2, C2.

```
100 DIM C$(20)
110 C$(13)="ENTER" :: C$(1)="AID" :: C$(6)="REDO" :: C$(12)="PROCEED
" :: C$(14)="BEGIN" :: C$(15)="BACK" :: C$(10)="DOWN" :: C$(11)="UP"
120 OPEN £1:"SCREEN"
130 CALL COLINV(16,7)
140 CALL BOX(0,0,8,78)
150 CALL DSPLAY(3,4,"TEXT AT 3,4")
160 CALL DSPLAY(5,6,CHR$(18)&"BLINKING AT 5,6")
170 CALL ACCEPTD(10,20,10,"PROMPT:",CHR$(18)&"DEFAULT   ",A$,K)
180 D$=SEG$(C$(K)&RPT$(" ",7),1,7)
190 CALL DSPLAY(12,20,CHR$(18)&D$)
200 GOTO 170
30000 SUB CLEAR80
30010 PRINT £1:CHR$(17);CHR$(19)
30020 SUBEND
30030 SUB POSN(R,C)
30040 PRINT £1:CHR$(27);"b";CHR$(R);CHR$(C)
30050 SUBEND
30060 SUB DSPLAY(R,C,D$)
30070 PRINT £1:CHR$(27);"b";CHR$(R);CHR$(C);D$
30080 SUBEND
30090 SUB ACCEPT(R,C,L,A$,K)
30100 IF L<0 THEN PRINT £1:CHR$(27);"i";
30110 PRINT £1:CHR$(27);"b";CHR$(R);CHR$(C);CHR$(27);"p";CHR$(L)
30120 LINPUT £1:A$ :: CALL KEY(3,K,V):: IF V=0 THEN 30110
30130 IF L<0 THEN PRINT £1:CHR$(27);"j"
30140 SUBEND
30150 SUB COLOR80(F,B,FB,BB,BL)
30155 IF BL=0 THEN PRINT £1:CHR$(27);"f";CHR$(16);
30160 PRINT £1:CHR$(27);"c";CHR$(16*(F-1)+B-1);CHR$(16*(FB-1)+BB-1)
30170 SUBEND
30180 SUB SBLK
30190 PRINT £1:CHR$(27);"e"
30200 SUBEND
30210 SUB ACCEPTD(R,C,L,P$,D$,A$,K)
30220 PRINT £1:CHR$(27);"b";CHR$(R);CHR$(C);P$;
30230 PRINT £1:CHR$(27);"b";CHR$(R);CHR$(C+LEN(P$)+1);D$
30240 PRINT £1:CHR$(27);"b";CHR$(R);CHR$(C+LEN(P$)+1);CHR$(27);"i";C
HR$(27);"p";CHR$(L)
30250 LINPUT £1:A$ :: CALL KEY(3,K,V):: IF V=0 THEN 30220
30260 PRINT £1:CHR$(27);"j"
30270 SUBEND
30280 SUB COLINV(F,B)
30290 PRINT £1:CHR$(27);"c";CHR$(16*(F-1)+B-1);CHR$(16*(B-1)+F-1);CH
R$(27);"f";CHR$(16)
30300 SUBEND
30310 SUB BOX(R,C,R2,C2)
30320 D=C2-C-1
30330 PRINT £1:CHR$(27);"b";CHR$(R);CHR$(C)
30340 PRINT £1:CHR$(201);RPT$(CHR$(205),D);CHR$(187)
30350 FOR RW=1 TO R2-R-1
30360 PRINT £1:CHR$(186);RPT$(" ",D);CHR$(186)
30370 NEXT RW
30380 PRINT £1:CHR$(200);RPT$(CHR$(205),D);CHR$(188)
30390 SUBEND
```

TELECOM TIPS no 1

Modems explained by Peter Walker

Modem is short for Modulator/Demodulator. Modems are used when you want to communicate with remote computers via the telephone network. You can't send digital signals over the telephone network, since it is only designed to transmit analogue audio signals, usually speech, in the bandwidth 300- 3400Hz. Therefore the binary 1s and 0s are converted to audio tones. This principle is the same as computers, including the TI99/4A, use when recording data and programs on a cassette recorder.

The method of transmission is that an audio tone acts as a carrier and is modulated by the data stream. There are various speeds, carriers and modulation methods used in modems as they have developed over the years. They are usually known by the international V recommendations published by the CCITT, the organisation responsible for worldwide telecommunications standards. American modems, at least the lower speed ones, are different and incompatible with the international standard modems, so don't buy a US modem for use in the UK!

As anyone familiar with radio theory will know, the higher the carrier frequency, the higher the data rate possible in the modulation stream. Also, the composite modulated signal possesses a spread spectrum both above and below the carrier frequency. All this needs to be borne in mind when considering modems, especially since the composite signal must fit into the 3100Hz bandwidth of the telephone network.

Lets look at the simplest form of modem, the V21 method, capable of transmission up to 300 baud (bits per second) in full duplex. Full duplex means transmission in both directions at the same time. Half duplex means transmission in only one direction at a time. Let's assume that the rest state of the RS232 (V24) interface to the modem is binary 1. The modem transmits a tone of 980Hz and shifts upwards to 1180Hz when the interface changes to binary 0. This method of modulation, similar to frequency modulation in radio, is known as Frequency Shift Keying (FSK). The modulated tone sounds like a squark, just like on TI99/4A data on a cassette tape. Now V21 is a full duplex system, but we can't use 980 and 1180Hz in the reverse direction, since your telephone line is a 2 wire line, so different frequencies must be used in the reverse direction. For this binary 1 is 1650Hz and binary 0 is 1850Hz.

In order to increase transmission speed, V23 mode was introduced. This is another FSK system using 1300 Hz and 2100Hz. The higher frequency and wider shift allows 1200 baud transmission. However, the modulated signal takes up a much greater proportion of the telephone bandwidth, so a reverse channel of 1200 baud is not possible. V23 therefore can only work in half duplex. A form of full duplex is possible however by squeezing in a very low speed channel at 75 baud in the band below the 1200 baud channel. This uses frequencies of 390 and 450Hz, low enough not to interfere with the bottom end of the 1200 baud spectrum. This is usually referred to as 1200/75 mode and is that used by most videotext systems, such as Prestel.

There are some modems that only feature 1200/75 V23 modulation, but while these may be cheap, they won't allow you to communicate with many of the services employing 300 full duplex. Furthermore, many of the communications packages available for the TI99/4A don't support 1200/75, largely I suspect because most of these packages originate in the USA, where 1200/75 is unknown. It's worth mentioning that the TI99/4A can't be easily made to work at 1200/75, since files are normally opened in a symmetrical mode (extension parameters such as BA=300. for example). To use 1200/75 you have to directly access the RS232 DSR routines and set up the differential speed, unless you have the revised DSR ROM that features split baud working extension parameters eg TX=75.RX=1200. This ROM also features true Centronics protocol on the PIO port.

So I do recommend that you try to get a modem featuring both V21 and V23 modes. Before looking at other desirable features, lts look at some of the higher speeds.

Full duplex 1200/1200 is possible with the V22 mode. This uses a completely different form of modulation, Phase Shift Keying (PSK). The carrier signal is modulated by rapidly shifting the transmitted phase of the signal by one of 4 different phase shifts at a frequency of 600 shifts (or symbols) per second. Each of the phase shifts represents one of 4 possible 'di-bits' namely 00, 01, 10 and 11. The result is that 2 bits are transmitted every 1/600 sec, thus achieving 1200 baud. This is a more complex system than FSK and until recently V22 modems were prohibitively expensive. Some are now available for around £120 and also usually feature V21 and V23 as well. V22 modems also usually feature a 'fallback' 600 baud PSK mode, in case transmission quality on the line is bad.

Going up in speed, "V22 bis" modems allow 2400 baud full duplex transmission with a 1200 baud fallback. These modems employ automatic equalisation to make up for any imbalance in the frequency response of the telephone connection. It employs another, yet more advanced, modulation method, known as Quadrature Amplitude Modulation (QAM). This works similarly to 4 phase PSK, except that for each phase change there are two amplitude levels, thus producing 8 possible symbols, which in turn can represent the 8 possible 'Tri-bits', eg 000, 001, 010, 011, 100, 101, 110 and 111.

Advanced techniques such as QAM can now be used to achieve both 9.6Kbits/s and even 14.4Kbits/s, but I will not cover these here since these are well outside the range of modems likely to be purchased by amateurs. Equally, space prevents me from mentioning the many intermediate systems of half- duplex modems, largely developed for 4 wire private circuits between computers, where the 4 wire circuits allow the same half-duplex mode in both go and return directions. There are even some modems designed to work over wideband telephone circuits, which can support data rates up to 64Kbits/s.

Lets now consider some of the features you should look for when purchasing a modem.

1. Speeds. I consider 300 duplex (V21) and 1200 half duplex (V23) as essential. Check that the latter also provides the 1200/75 mode, needed for Prestel.

2. Modes. Both Originate and Answer modes are essential, since you might be at the 'answer' end of a connection if transferring data from another user. Answer mode on V23 gives you 75/1200 working.

3. Status Lights. It's important to see visually whats going on at the modem interface or line. Carrier Detect is almost essential, Transmit and Receive data very useful. More advanced modems may possess lamps for DTR, DSR and "on-line".

4. Local Test. The ability to loop the transmit and receive paths will allow you to check that your computer and modem are working by echoing back transmitted characters.

These are some optional advanced features you might like:-

5. 1200 full duplex (V22) and 2400 full duplex (V22 bis).

6. Auto-answer, necessary if you want to act as a bulletin board for others to call into.

7. Auto-dialling, allowing your modem to do the difficult chore of dialling the telephone call. Beware! There are two distinct sorts of auto-dialler. The first is driven by an auxiliary port used to interface with specific computers eg the BBC. To make this work from the TI99/4A, you would need to develop a suitable hardware interface and dialling program. Not impossible of course; my article in TI*MES no 26 shows how it could be done. The second type of auto-dialler is completely self-contained in the modem and driven by the Hayes protocol over the RS232 port itself, see below. If your telephone exchange is capable of multifrequency signalling, make sure your modem is too, though loop pulsing will work with all exchanges.

8. Hayes Protocols. These allow you to command the modem by passing specific ASCII codes over the RS232 port when the modem is not 'on- line'. They are all prefixed by the code AT. For example the command ATDT0716181111 would command the modem to dial with multifrequency signalling the number 071 618 1111 (London Prestel) and then go 'on- line'. The code ATF4 would command the modem to set itself to the 1200/1200 speed.

9. Auto-speed sensing. A real luxury, and usually an expensive one. The modem will attempt to communicate with the distant modem at 2400, then if it fails, 1200 then 600, 300 etc. Its worth mentioning here, in passing, that all modems including fixed speed ones, actually answer by first transmitting a 2100Hz tone for a couple of seconds. This can be used as a definite indication of answer. In the international telephone network, the 2100Hz disables any echo-suppressors in the connection. These would normally be used on intercontinental circuits or satellite circuits to remove echoes and work by effectively allowing only "half- duplex speech". Clearly full-duplex data needs a clear channel in both directions.

10. Error Correction. Should you ever need to communicate with a remote system offering this feature, you need a compatible modem. These use a line protocol with error detection and block retransmission, so correcting errors due to line noise.

Finally, a few notes on connecting modems to the TI99/4A. The following diagram is suitable for using RS232 port 1.

```
TI99/4A  RD  2 <------------ 3 RD      MODEM
         TD  3 ------------> 2 TD
      Earth  7 ------------- 7 Earth
Control i/p 20 <------------ 5 CTS
  Fixed o/p  6 ------------> 4 RTS
                             :
                             --> 20 DTR
```

The latter connections (6 to 4 and 20) won't be necessary if your modem inputs default to 'on', but include them if you are not sure. For a complete discussion of the RS232 port on the TI99/4A, see my article in TI*MES no 20.

In the last article, I described modems, how they work, what to look out for when buying one and how to connect them to the TI99/4A.

In this article I will give some information about accessing Prestel from your TI99/4A.

Hardware - you need a full system (disc + 32K + RS232) and modem capable of V23 1200/75 transmission.

Software - The only known program for accessing Prestel is the Dutch program known as Viditel (the Dutch name for Prestel) written by Rob Tempelmans Plat. I do not have a contact for this gentlemen: we are indebted to him for an excellent program. Most versions used in the UK stem from the one I modified some years ago, where I overwrote the Dutch instructions and title with the English equivalents. If your title screen says "VIDITEL" instead of "TI-PRESTEL" then you have a copy of the unaltered program. The software itself comprises an ExBas load file, a DIS/FIX80 assembler loader and a PROGRAM image file which contains the program itself. The latter can be directly loaded as an EA5 file if desired. The program is written in Bit Image mode and can thus use full colour and 40 column display. Text mode offers 40 columns but is restricted to two colours. The program is not without the odd bug. You can often see on Prestel where our program fails to implement capital letter "blast-through" when in graphics mode. Instead, the capital characters 64-94 appear as 32-62. Also, in answer to inumerable requests, the program does not have any features such as Print Screen, Screen Dump, Upload/Download. Its just a basic Prestel emulator.

If you wish to try out Prestel before subscribing, you can see sample pages by using account 4444444444 and password 4444. Prestel costs £8 per quarter rental. There are time charges while you are logged on: 7p per minute peak time, 1p per minute after 6pm and on Sundays. Additionally, some pages are charged separately, eg 50p for a full weather report. All charges are VATable. On top of this you must pay the local call charge telephone costs of access: these appear on your regular phone bill of course. Subscription to Micronet is a good deal extra on top of all this and in my opinion not worth the extra for TI99ers. Micronet, while owned by BT, is actually an independent information provider on Prestel, specialising in information for Micro users. There's nothing of use to 99ers, but Micronet do offer a free 1200/75 only modem to new subscribers. Its up to you of course.

One of the best features of Prestel is the electronic mail, or Mailboxing (MBX) which allows you to send messages to other subscribers. To send me a message, address to MBX 707873778. Some of the recently introduced functions on the updated mailbox system attract a small charge. I recommend a thorough reading of the Mailbox User Guide. The main MBX area in on page 7. The old style MBXs can stil be found on pages 77 and 83.

The Prestel character set is somewhat different from ASCII. Our emulation did not originally display all of the characters that Prestel intended. The main differences are:

PRESTEL TRANSLATION CHART

ASCII	PRESTEL	TI99 SHOWS	(ORIGINAL PACKAGE)
35	£	£	NOT SHIFT 3! SEE BELOW
91	←	[	LEFT ARROW
92	½	\	HALF
93	→	]	RIGHT ARROW
95	#	#	NOT _ ! SEE BELOW
96	—	`	LONG DASH
123	¼	{	QUARTER
124	‖	¦	SEPARATOR
125	¾	}	THREE QUARTERS
126	÷	~	DIVISION
127	█	‖	"DELETE" = PARITY ERROR

I have produced a modified version of Viditel that corrects all these character definitions - a copy can be obtained from the library.

Detailed information on the character set can be found on page 33320907 of Prestel. Numerous Escape codes are provided to change text and background colours and to switch into Graphics mode. This mode uses 6 elements per character space to create block graphics. These characters are known as 'Sixels'. See chart 3 below.

The Escape code is ASCII 27 which is CTRL . on our machine. Prestel also allows the use of as an alternative. All control codes for Prestel are in chart 1 and Escape codes in chart 2. An important point must be made about the # character. In Prestel # is ASCII 95, not 35. Our program sends ASCII 95 when Shift 3 is pressed, just as FCTN U (_) does. If you want to send the UK pound sign (£), which is Prestel character 35, then the following sequence, exploiting the faulty blast through in graphics mode, can be used: @WC@G

One of the biggest problems for us is editing mailboxes and other input fields, since our package does not support a proper cursor. You can sometimes see a partial blank when the cursor rests over an existing character, but its not at all clear.

Using Double Height and different colours can make your MBXs look more attractive. To switch to Double Height, use Escape M and Escape L to revert to single height, ie:

@M DOUBLE HEIGHT @L SINGLE AGAIN

Similarly to change colour use the appropriate escape codes eg:-

WHITE @A RED @B GREEN @D BLUE @G WHITE AGAIN

To change the background colour, first use Escape] to set the background colour to the foreground colour. Then set a new foreground colour. From the normal white on black, this is how you would create blue on yellow:

@C@]@D BLUE ON YELLOW

Escape \ resets the background to black while leaving the current foreground colour.

Full technical information on Prestel and other telecoms subjects is on page 333204. Access numbers for Prestel can be found on page 33311. This gives details of some non-1200/75 Prestel ports eg 081 680 8245 for 300/300 and 081 680 9833 for 1200/1200. This allows one to access Prestel using a more conventional terminal program eg Telco or Fast-Term. The 300 baud access doesn't send any escape codes and so is good for ASCII terminals. It sends asterisks in place of graphics. The 1200 baud access does send all the escape codes, so you must be able to interpret the text from amongst the special codes and graphics. However, these ports do give you the opportunity to upload/download and print. There is an emulator called COMMTY which features 1200/75 speed, but it seems to have a bug in it which causes the left hand two characters not to be displayed.

Finally, the 3 charts below list all the special codes you need to know for fully exploiting Prestel. The first chart shows the special control codes for Prestel, the second the escape codes (ie codes used after ASCII 27) and the third the special Sixel codes for Prestel block graphics. Happy telecommunicating!

Happy telecommunicating!

Peter Walker

STOP PRESS STOP PRESS STOP PRESS STOP PRESS STOP PRESS STOP PRESS STOP PRESS

ASCII	KEY	ALT	FUNCTION	COMMENTS
0	CTRL ,			
1	CTRL A			
2	CTRL B			
3	CTRL C			
4	CTRL D			
5	CTRL E		Who are you?	Some terminals send ID in response
6	CTRL F			
7	CTRL G			
8	CTRL H	FCTN S	Cursor Left	
9	CTRL I	FCTN D	Cursor Right	
10	CTRL J	FCTN X	Cursor Down	
11	CTRL K	FCTN E	Cursor Up	
12	CTRL L		Move to next word space	See Note 1
13	CTRL M	Enter	Home cursor	Screen is NOT cleared
14	CTRL N		Shift out	?
15	CTRL O		Shift in	?
16	CTRL P			
17	CTRL Q		Cursor on	NOT ON TI99!
18	CTRL R			
19	CTRL S			
20	CTRL T		Cursor off	NOT ON TI99!
21	CTRL U			
22	CTRL V			
23	CTRL W			
24	CTRL X		Delete line	Note 2
25	CTRL Y			
26	CTRL Z			
27	CTRL .	@	Escape	See next list for Esc codes
28	CTRL :		SS2	?
29	CTRL =		SS3	?
30	CTRL 8		Cursor Home	
31	CTRL 9			
32	space			
33	!			
34	"			
35	£			Not sent by hash key use @WC@G instead
36-94	as regular ASCII			
95	#	_	End Edit	Sent by Hash key
96-126	as regular ASCII			
127	FCTN V		Del (Parity Error)	

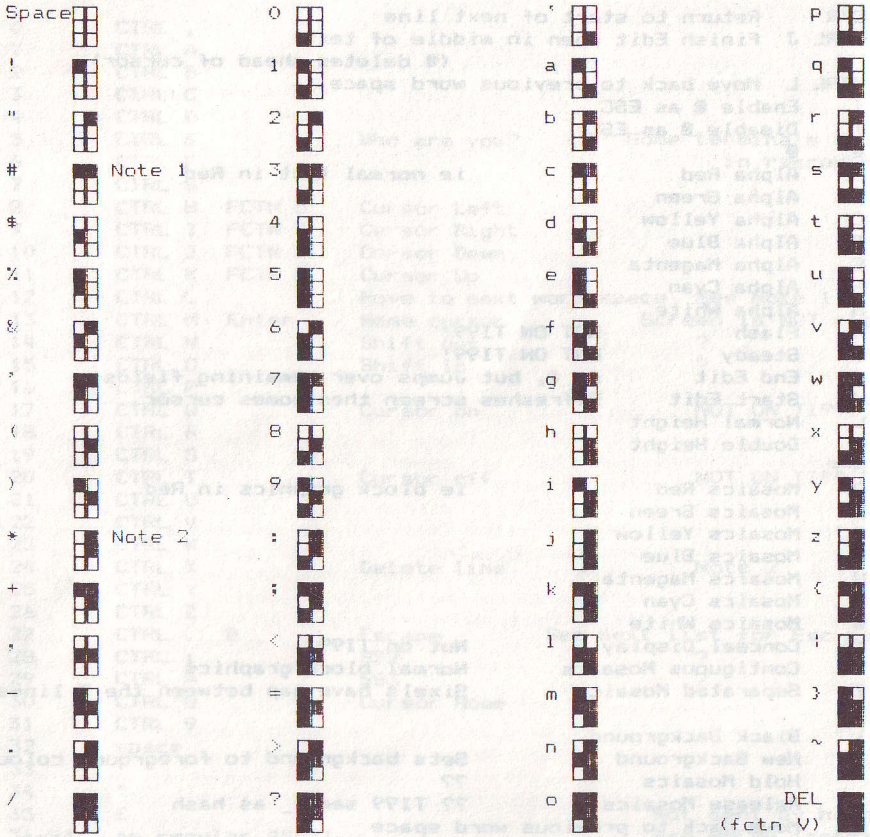
Note 1. CTRL L sent by the host clears the terminal screen.

Note 2. CTRL X is strange. In an MBX, when first used, it homes the cursor. When used again it clears the text ahead of the cursor.

ESCAPE CODES

C/R	Return to start of next line	
CTRL J	Finish Edit when in middle of text	(# deletes ahead of cursor)
CTRL L	Move back to previous word space	
(	Enable @ as ESC	
)	Disable @ as ESC	
@	@	
A	Alpha Red	ie normal text in Red
B	Alpha Green	
C	Alpha Yellow	
D	Alpha Blue	
E	Alpha Magenta	
F	Alpha Cyan	
G	Alpha White	
H	Flash	NOT ON TI99!
I	Steady	NOT ON TI99!
J	End Edit	As #, but Jumps over remaining fields
K	Start Edit	Refreshes screen then homes cursor.
L	Normal Height	
M	Double Height	
N-P		
Q	Mosaics Red	ie block graphics in Red
R	Mosaics Green	
S	Mosaics Yellow	
T	Mosaics Blue	
U	Mosaics Magenta	
V	Mosaics Cyan	
W	Mosaics White	
X	Conceal Display	Not on TI99
Y	Contiguous Mosaics	Normal block graphics
Z	Separated Mosaics	Sixels have gap between the 3 lines
[		
\	Black Background	
]	New Background	Sets background to foreground colour
^	Hold Mosaics	??
_	Release Mosaics	?? TI99 sees _ as hash
b	Move back to previous word space	
d	Delete character	
e	Move to end of line	
f	Move forward to next word space	
h	#	
i	Insert space	
o	Word wrap off	
r	Return to start of next line	
s	*	
v	Verify	
w	Word wrap on	

Graphics Characters - Sixel Patterns



Note 1: Cannot be sent by TI99/4a # key!
Note 2: To send use @s not *
Note 3: In graphics mode, capitals ASCII 64-94 should appear as normal, but on our package appear as 32-62.

The conversion from ASCII to Sixels is as follows, using f as example. f is ASCII 102 or 1100110 in binary. Ignoring bit 6 (counting left from least significant bit), it will always be 1, allocate sixels as on or off according to following bit map:

```
12
34
57
```

Thus f becomes 01
10
01

MODULE WOBBLE ?

MIKE GODDARD :MAY 1990

The main cause of "Module Wobble" is usually a dirty module connector and the simplest remedy is to spray some switch cleaner into the module port and let it dry before switching on again. However this is only a temporary remedy at best and the problem will soon occur again The next step is to open up the console and remove the piece of "dirty rag" which Mr TI fitted in his infinite wisdom and which usually ends up depositing more gunge on the module contacts than would normally accumulate. This is held in a small plastic holder clipped to the front of the module connector.

Cleaning however is only one method of curing the problem because the contacts themselves get worn and lose some of their spring coupled with the fact that not all modules have the same thickness edge connector and we have a set of variables enough to BOGGLE the mind. The only real solution when the connector is really playing up and has just crashed the console right in the middle of "ZAP THE ALIEN MOTHER IN LAW" is to change the connector for a better quality gold plated one these are available from MAPLINS part No JB99H @ 98pence although this version doesn't have the lugs to fit the "Dirty Rag Holder" (a distinct advantage in my opinion) it does work extremely well. I've now fitted this modification to six consoles and haven't had one bad report yet.

The biggest job in fitting the new socket is removing the old one, one method would be to saw through all the connector pins and remove the little bits individually with a soldering iron. The metod I used which works quite well is to heat the pins with a heat gun and pull the connector out in one piece, then thoroughly clean all of the surplus solder from the board with a soldering iron and solder sucker before fitting the new connector.

There is always the possibility that some of the bad connections are caused by the connector that goes into the main circuit board. In my experience this is very rare because this connector doesn't have modules constantly inserted and removed from it and therefore doesn't have the same "usage" as the actual module connector but it is a good idea to give it a clean with some switch cleaner while you have the console apart.

Gold contacts should have a longer life than the cheapo version that Mr TI fitted however ot is still a good idea to give them an initial squirt of switch cleaner and repeat this at regular intervals probably every two or three months would suffice and of course prevention is always better than cure.

MINI MEMORY MADNESS

by Peter Hutchison.

Accessing Graphics from Assembler.

As you may know all screen data, colour data and character data are held in what is known as VDP (Video Display Processor) Memory. To access this memory, as the 9900 cannot directly access it, is via the System Utility Routines (p34 in Mini Memory manual or p248 in Ed/Assem Manual).

The screen contains 24 x 32 characters, a total of 768 character positions, the memory for this starts from address 0 to 767 (see Appendix F). Now, to print a character on the screen depends from where you are calling your program from, that is BASIC or Easy Bug and Mini Memory Run option. If you run from Easy Bug you could enter:-

```
        LI R1, >3000 ; ASCII for zero (O)
        LI RO, 176   ; row 5 column 16 (5*32+16)
        BLWP   @>6024 ; call VSBW routine
```

which prints a O at row 5 and column 16. If in BASIC you'd enter:-

```
        LI R1, >9000 ; ASCII for zero + >60 (96)
        LI RO, 176
        BLWP   @>6024
```

as there is a screen bias of >60 (see p61 for example). To write a string of characters you would use VMBW e.g.

```
        LI RO, 40      ; 2nd line of screen
        LI R1, MG      ; address of text
        LI R2, 16      ; length of text
        BLWP @>6028
        ...
    MG  TEXT 'TI HOME COMPUTER'
```

To create new graphic charcters you can modify the character data stored from address >0400 to >05FF (64 characters from " " to "_") (Appendix F) or >0800 (CHR$ 0) to >0FFF (CHR$ 255) without BASIC (Appendix E). For example try this:-

```
        LI RO, >08F0   ; Address in Character table
        LI R1, CR      ; Address of character data
        LI R2, 8       ; Number of bytes
        BLWP @>6028    ; Define character
        LI RO, 170     ; Screen address (5*32+0)
        LI R1, >1E00   ; ASCII code
        BLWP @>6024    ; Display character
        B *R11
    CR  DATA >3C7E, >DBFF, >E7BD, >C37E
```

The colours of character sets can also be changed, the colours are stored at address >0300 to >031F (or >0380 to >039F without BASIC) for each of the sixteen colour sets. For each byte the most sig. 4 bits contains the foreground (0=transparent to 15=White) and the least sig. bits contain the background. The following program sets all colours to black on cyan:-

```
        LI R3,16        ; Number of sets
```

```
        LI RO,>0380    ; Address of colour table
        LI R1,>1700    ; Black on cyan
    L1  BLWP @>6024    ; Poke colour
        INC RO         ; Next address
        DEC R3         ; Reduce count
        JNE L1
```

To set the background colour as CALL SCREEN does then a special VDP register needs to be set ie register 7. To set the colour to Light Green enter:-

```
        LI RO, >0703   ; Register 7, Light green = 3
        BLWP @>6034    ; Set VDP register using VWTR.
```

There should be enough here to write a simple game in Assembler, although it is alot more long winded than BASIC.

If you want to write to me about anything to do with assmebler write to:-

6 Moorlands View, Free School Lane, Savile Park, Halifax
HX1 2XQ. Tel: 0422 355857.

ParaPrint Review

by Peter Hutchison.

ParaPrint is a printer interface for the TI from DataBioTics. From the package I got was the ParaPrint interface made from an old cartridge with one end that fits to the right hand port of the TI and the other where the cable fits. There was also a printer ribbon cable and a Tandy Universal Adaptor to power it and a thin user guide.

After a bit of trial and error I managed to get a self test by holding down the Space Bar and switching on the computer - you get a print out of all the characters. Then I had a go at printing a message for example :-

```
        10 OPEN #1:"PIO"
        20 PRINT #1:"ITS PARAPRINT!"
        30 CLOSE #1
```

which worked fine on my Brother M1009 printer. Then to really test it out I tried to print out a small BASIC listing using the command :-

```
        LIST "PIO.LM=5.W1"
```

which printed a BASIC program with a left margin of 5 characters and Word Wrap for TI BASIC programs. Trouble is, it uses all the TI processor time so you can't work while you are printing.

According to the manual ParaPrint recognises several device names including PIO, PIO/1, PIO/2, RS232, RS232/1 and RS232/2. There are also some useful functions it can perform :-

.CR	No auto carriage return/line feed.
.DA=n	Set no. of data bits n=7 or 8.
.HX	Print data in Hex.
.LF	No auto line feed.
.SP=n	Set line spacing n=1 to 256.
.LM=n	Set left margin n=1 to 256.
.NU=n	Send n null characters to printer n=1 to 255.
.RM=n	Set right margin n=1 to 256.
.W1	Word Wrap when listing a BASIC program.
.W2	Word Wrap when listing an Extended BASIC program.
.WW	Word Wrap when printing text.

It is also compatible with common modules such as TI-WRITER, MYARC Ext. BASIC, PRK etc. although on the Ed/Assem an extra dot is required after the last option.

Conclusion: For those wanting a fairly cheap interface for access to printers I can recommend ParaPrint for ease of use, its commpatability and its powerful facilities.

ParaPrint (P.I.O) is available from DataBase for #65.

DISK LIBRARY NEW ADDITIONS:

Stephen Shaw. 10 Alstone Road, STOCKPORT, Cheshire, SK4 5AH

This is taken from a file NEW/DISKS which can always be found on the library disk catalogue set (send three disks and return post and packing for a complete set) and also printed copies can be sent to reach you a little faster than TI*MES if you supply some SAEs!

>GAMES 21: TETRIS in XB from MICROPENDIUM, a good XB version; StarTrade an XB trading game; ARCHEDROID; BERLIN; BLADE RUNNER; NAVIGATOR and TIP TOE TOM- by Arto Heino, this last one is worth comparing to TNT. Also from Cheryl Whitelaw (Regena) comes a Geography quiz program, AFRICA, a solitaire card game, PYRAMID, and a forerunner of Yahtzee (you knew all along it was not entirely original!) from Hoyle, called YACHT.

>SmArtCopy now replaces Hardcopy- this is a major upgrade now with the ability to print TI Artist pics-all facilities available, to texturise colours, magnify in both axes by up to 999x, and to clip bits. Also the MyArt section has been upgraded to handle longer pictures -up to 424 lines.

>MA10 is the old Hardcopy disk, retained and renamed to make available the two pics, the glass ball on a chequerboard and the excellent dragon DRACO.

>MA11 is a Yapp format picture for SmArtCopy, called MonaLisa by an Italian artist, converted to MacFlix, and now to an extended MyArt format. This is an unclipped conversion, for best results and true proportions you are recommended to use Opt 2 settings: H=5, V=3, S=0, and option 3 settings of H=105, V=20, dH=285 and dV=400. This is a monochrome pic by the way- no colours were used.

>CONTRACT BRIDGE Vn 3.01 by John H Bull. TWO DISKS. With fifty preplayed hands to compare scores with. The "tutorial" option is far from it- you need to have a bridge rule book or know bridge! Its a card game by the way! The computer plays one partnership and you play the other. John makes the point that the computer does not play at professional level, but you have the ability to save hands and replay them to see if you can improve on your previous performance!

>BRIDGE BIDDING 1. PHD5026 from TI.

>BRIDGE BIDDING 2. PHD5039 from TI.

These two tutorial disks, containing Basic programs, assume a fair knowledge of the card game called Bridge! They originally sold for US$30 each, as far as I can tell TI never brought them to the UK.

>GPL DISASSEMBLER V1 by ART GREEN so it must be good. Requires the ability to run a machine code program irrespective of module- eg a gramkracker, or other device.

continued --->]

>GPL ASSEMBLER V1 by ART GREEN. REQUIRES GRAM DEVICE. TWO DISKS.

>GPL LANGUAGE MANUAL. BY ART GREEN. TWO DISKS.

NB: Art makes the point that his usage of GPL is not based on any official information and therefore his syntax/usage may differ from any "official" documentation that probably now does not exist.

============

***** STAR ITEM THIS TIME ROUND ******:-

>MULTIPLAN VN 4.0 ROM VERSION. FOR USE WITH THE MULTIPLAN MODULE. A comprehensive rewrite of the interpeter portion of the original disk has resulted in a REAL increase in entry and processing speed. At last after many years my dusty module can find a use! Very highly recommended- essential for every Multiplan owner.

============

>MULTIPLAN VN 4.0 GRAM VERSION. If you have a GRAM device this may be of use to you. The files appear to be in the Miller Gram-Kracker format.

>BOOT DISK CHANGER Vn 2.1 by Scott Morrow- true Freeware, its yours free in return for a disk and return post and packing (no extra packing added!) or at your option on usual library terms. Tested with TI Disk controller only. This disk is an intelligent sector editor- it will check named files for any occurrence of DSKn. where n is any number from 1 to 9, and will amend the number to one you select. Thus you can run say TI RUNNER from disk drive 2 if you wish! NB HRD Owners: Use a disk device on CRU 1000 or less when running this program- the sector access routine makes no allowance for HRDs on CRU 1100 but of course the files once they have been amended may be stored on such devices.

>THE MISSING LINK GRAPHICS- REQUIRES THE MISSING LINK- A disk full of graphics programs to run with The Missing Link. Most of the graphics I have printed in Rambles and a few more as well!

Disk copying is one pound per side, and if you wish me to provide the disk thats an extra one pound per disk. Packaging/postage is fixed at one pound no matter how many disks you request. All disks are SSSD and assume ownership of XB.

=====================================

RAMBLES by Stephen Shaw

For TI*MES July 1990

My address is:
10 Alstone Road, STOCKPORT, Cheshire, ENGLAND, SK4 5AH
and I would love to hear from you...
write and say what you want more of/less of in Rambles, ask questions (but not hardware-see Mike; or machine code!)[and if a direct response is required, an SAE is welcome!]. What do you do with YOUR TI? Written any programs? Prepared any graphics?

Back in issue 27, on page 20 I set two TESTS. Not a single response to either of them, however I have now discovered a whole series of questions of the type posed by Test 4, in "The Master Book of Mathematical Recreations" by Fred. Schuh, published by Dover, ISBN 486-22134-2. Refer to puzzles 298 to 301.

---> continued--->]

The same author, in the same book, claims the creation of (and describes) the puzzle Giants and Dwarfs- a program of this can be found on library disk Games-3.

No feedback at all on the module reviews (in fact unexpanded owners remain the great unknown, as they dont seem to want me to write anything for them- no letters anyway! Hello?)-so in this issue a smaller number of module reviews. And remember that ALMOST everything can be obtained from SOMEWHERE, even if you do have to try several US sources.
Martin Blythe at Database is worth checking on- the poor chap had some extremely bad health recently, so it was nice to see him at Chester. Martin has now had a good look at the original F Farrish prices,which were his original guide, and has made some nice reductions, so if you were previously deterred, have another look! There are ample stocks of some quite nice modules, and at Chester Martin had stocks on sale of some modules which were both very good AND pretty rare!

There is a disk program called ELECTRIC HARRY bombing around- please be aware that it is a pirated version of ESPIAL with the pirates name none too well disguised!

Congratulations to our new cassette librarian, all you cassette owners are encouraged to contact Nicky a.s.a.p. and take advantage of our luvly cassette software. Our very sincere apologies to unexpanded owners who have experienced difficulties in obtaining their needs from our previous cassette librarian in recent months. We now have things better arranged, so do please drop Nicky a line- as ever an SAE is very useful.

The puzzle set in the last issue (Page 19) brought in two entries and a query. The first clue is in the first hint- the answer has more than fifteen digits. If you refer to NUMERIC CONSTANTS in your Users Reference Guide, you will see "...numbers will have 13 or 14 digits depending on the value of the number."

So how can a Basic program handle a problem which requires an answer with more than fourteen digits??? Simple, we allow the program to deal with numbers with less than fourteen digits- and create our larger numbers from these smaller numbers. We do this all the time, and learn it in our first or second year of schooling. 2+2=4 is easy. Now try 8+8=.... the answer needs more than one number! We could call it sixteen, but to write every number with a separate character would slow learning down a mite. In writing numbers we use a decimal system, with the rightmost number representing a number of units, the number to the left of it the number of tens of units and so on.

To solve this problem we must be much more specific in our instructions to the computer, which for our purposes is a dunce! (Though fortunately a fast one!).

We need our program to deal with the "carry"s! Then of course we need to work out how to actually do all the multiplications -how to store the little numbers making up ther big number, how to get them back together for testing for the six sevens...

----> continued --->

We can test six digits from every character position, and we can speed things up by only testing every sixth character (or digit) and if - and only if- that is a seven, testing the numbers in front of and behind it. This short cut brings a useful increase in speed.

Here are the only listings sent to me up to the time this issue went to press:

```
99 REM John Seager
100 !ANSWER TO TEST5/A
110 CALL CLEAR :: NUM$="117649" :: DISPLAY AT(1,1):"7 TO THE POWER
OF"
120 FOR POWER=7 TO 300 :: DIS$=NUM$ :: NUM$="" ::
CARRY,COUNT,SEVENS=0
130 FOR J=LEN(DIS$)TO 1 STEP -1 ::
NEWNUM=(VAL(SEG$(DIS$,J,1))*7)+CARRY :: CARRY =INT(NEWNUM/10)
140 IF NEWNUM-CARRY*10<>7 THEN COUNT=0 ELSE COUNT=COUNT+1
150 SEVENS=MAX(COUNT,SEVENS):: NUM$=STR$(NEWNUM-CARRY*10)&NUM$
160 NEXT J
170 IF CARRY>0 THEN NUM$=STR$(CARRY)&NUM$ :: IF CARRY=7 THEN
COUNT=COUNT+1
180 SEVENS=MAX(COUNT,SEVENS):: DISPLAY AT(1,19):STR$(POWER);"=": :
:NUM$
190 IF SEVENS<>6 THEN 210 ELSE DISPLAY AT(24,1):"ANY KEY TO CONTINUE"
200 CALL KEY(0,K,S):: IF S=0 THEN 200 :: DISPLAY AT(24,1)
210 NEXT POWER
220 REM WHY DOES THIS CRASH WHEN THE NUMBER OF DIGITS EXCEEDS 254....
```

```
99 REM FASTER WAY John Seager
100 ! ANSWER TEST5/B
110 CALL CLEAR :: DIM ELEM(26):: ELEM(0)=7 :: POWER,SS=0 :: DISPLAY
AT(1,1):"7 TO THE POWER OF"
120 ELM=SS :: SS,CARRY=0 :: POWER=POWER+1
130 DIS$=STR$(ELEM(ELM)):: FOR I=ELM-1 TO 0 STEP -1 ::
DIS$=DIS$&RPT$("0",10-LEN (STR$(ELEM(I))))&STR$(ELEM(I)):: NEXT I
140 DISPLAY AT(1,19):STR$(POWER);"=": : :DIS$
150 FOR I=6 TO LEN(DIS$)STEP 6 :: IF SEG$(DIS$,I,1)<>"7" THEN 190
160 FOR J=I-5 TO I :: IF SEG$(DIS$,J,6)<>"777777" THEN 180 ELSE
DISPLAY AT(24,1) :"ANY KEY TO CONTINUE"
170 CALL KEY(0,K,S):: IF S=0 THEN 170 :: DISPLAY AT(24,1):: J=I
180 NEXT J
190 NEXT I
200 ELEM(SS)=ELEM(SS)*7+CARRY :: IF ELEM(SS+1)=0 AND ELEM(SS)<1.E+10
THEN 120
210 CARRY=INT(ELEM(SS)/1.E+10):: ELEM(SS)=ELEM(SS)-CARRY*1.E+10
220 SS=SS+1 :: GOTO 200
```

=======================

You may be glad to know (see p 22 last issue) that both HM Customs and HM Post Office have refunded to me the money charged on a pack of disks from Jim Peterson. The Post Office made the interesting comment that under International Regulations they may collect a fee in respect of every item of overseas mail delivered here, as of course everything from abroad has the potential to contain something subject to taxation! They in practice only charge an inspection fee when an item is actually assessed for duty/vat. The refund cheque was drawn on their WAGES account!!!

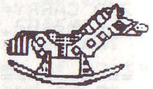
---> more! -->

The Horse Demo on p31 of the last issue caused a few problems- unfortunately as we have already proven, consoles run at different speeds, and it does become a little obvious in programs like this one, in which internal running time is so important to decide how long the horse is going to be in the air, or even how often and how accurate the coincidence check is to be. With only a little tampering to adjust for your own console speed you should make this one work.
If the horse fails to jump, then increase the number 10 in line 580 to say 13 or 14.
If the horse jumps too late, instead of increasing the number in line 580, try instead reducing the loop in line 650, or even remove the loop entirely with a REM.
To bring the horse down faster reduce the loop in line 620 or remove the loop entirely. (And vice versa in all cases!).

PLEASE NOTE that the TWEENING PROGRAMS on pages 71 to 76 of the last issue are entirely suitable to conversion to run with TI XB and THE MISSING LINK, as TML also operates in bit map mode with the ability to use sprites.

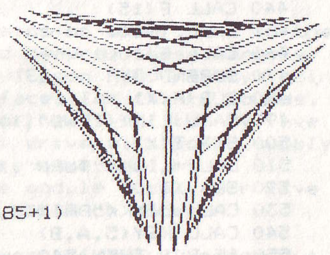
Now for some graphics programs for our latest addition to the TI Graphic family, THE MISSING LINK:

```
98 ! GRAPHICS PROGRAM FOR TI XB + THE MISSING LINK S Shaw from JBM103
100 CALL LINK("CLEAR")
110 CALL LINK("WINDOW",0,10,193,241)
120 CALL LINK("PRINT",2,9,"When ANY KEY is displayed")
130 CALL LINK("PRINT",19,1,"press: to:")
140 CALL LINK("PRINT",30,1,"CTRL FCTN....PRINT TO PIO")
150 CALL LINK("PRINT",66,1,"R run this program again")
160 CALL LINK("PRINT",78,1,"A save to artist file")
170 CALL LINK("PRINT",90,1,"ANY OTHER for different random picture")
180 CALL LINK("PRINT",131,1,"right now...")
190 CALL LINK("PRINT",142,1,"PRESS R-RANDOM PIC")
200 CALL LINK("PRINT",153,1,"press I-INPUT VARIABLE")
210 CALL KEY(5,X,Y)
220 IF Y<1 THEN 210
230 RANDOMIZE
240 CALL LINK("CLEAR")
250 DIM A(30,1):: P=2*PI
260 IF X=73 THEN CALL LINK("INPUT",160,200,N,2,"5"):: IF N>29 OR N<3
THEN 260 ELSE 280
270 N=INT(RND*26)+4
280 CALL LINK("PRINT",50,50,"ONE MOMENT...")
290 FOR X=0 TO N-1 :: A(X,0)=SIN(P/N*X)*96+128 ::
A(X,1)=COS(P/N*X)*96+95 :: NEX T X
300 CALL LINK("CLEAR")
310 CALL LINK("PRINT",9,20,"N="&STR$(N))
320 FOR I=0 TO N-1 :: FOR II=I+1 TO N-1 :: CALL
LINK("LINE",A(I,1),A(I,0),A(II,1 ),A(II,0)):: NEXT II :: NEXT I
330 CALL LINK("PRINT",1,1,"ANY KEY")
340 CALL KEY(5,X,Y)
350 IF Y<1 THEN 340
360 IF X=82 THEN RUN
370 IF X<>65 THEN 230
380 CALL LINK("INPUT",1,1,A$,14,"DSK1.PICTURE")
390 CALL LINK("PRINT",1,1,"' ")
400 CALL LINK("SAVEP",A$)
410 GOTO 330
============================================
```



---> more --->

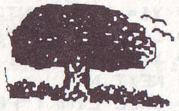
```
100 REM TML XB APRIL 1990 STEPHEN SHAW ENGLAND
110 REM FOR TI XB + THE MISSING LINK
120 REM
130 REM after john corbitt
140 CALL LINK("CLEAR")
150 CALL LINK("WINDOW",0,10,193,241)
160 CALL LINK("PRINT",2,9,"When ANY KEY is displayed")
170 CALL LINK("PRINT",19,1,"press: to:")
180 CALL LINK("PRINT",30,1,"CTRL FCTN....PRINT TO PIO")
190 CALL LINK("PRINT",66,1,"R run this program again")
200 CALL LINK("PRINT",78,1,"A save to artist file")
210 CALL LINK("PRINT",131,1,"right now...")
220 CALL LINK("PRINT",142,1,"PRESS SOMETHING...")
230 CALL KEY(5,A,B)
240 IF B<1 THEN 230
250 REM IKEDA MAP
260 REM from fractal report No 7
270 REM from program by John Corbit 16Aug89 FROM Myarc xb+ti99/4a s
shaw feb 90
280 REM chaotic attractor map
290 REM
300 X,Y=0 :: P=7.70
310 REM
320 CALL LINK("CLEAR")
330 FOR N=1 TO 4399
340 COSTHETA=COS(.4-(P/(1+(X*X+Y*Y))))
350 SINTHETA=SIN(.4-(P/(1+(X*X+Y*Y))))
360 X1=.85+.9*X*COSTHETA-.9*Y*SINTHETA
370 Y1=.9*X*SINTHETA+.9*Y*COSTHETA
380 CALL LINK("PIXEL",(X1+1)*63+1,(Y1+1.7)*85+1)
390 X=X1 :: Y=Y1
400 NEXT N
410 CALL LINK("PRINT",1,1,"ANY KEY")
420 CALL KEY(5,A,B)
430 IF B<1 THEN 420
440 IF A=82 THEN RUN
450 IF A<>65 THEN 420
460 CALL LINK("INPUT",1,1,A$,14,"DSK1.PICTURE")
470 CALL LINK("PRINT",1,1,"' ")
480 CALL LINK("SAVEP",A$)
490 GOTO 410
====================================
```



```
100 ! TI XB plus THE MISSING LINK Turtle type graphi cs
110 REM TML EX BAS MARCH 1990 STEPHEN SHAW ENGLA ND
120 REM
130 CALL LINK("CLEAR")
140 CALL LINK("WINDOW",0,10,193,241)
150 CALL LINK("PRINT",2,9,"When ANY KEY is displayed")
160 CALL LINK("PRINT",19,1,"press: to:")
170 CALL LINK("PRINT",30,1,"CTRL FCTN....PRINT TO PIO")
180 CALL LINK("PRINT",66,1,"R run this program again")
190 CALL LINK("PRINT",78,1,"A save to artist file")
200 CALL LINK("PRINT",131,1,"right now...")
210 CALL LINK("PRINT",142,1,"PRESS SOMETHING...")
220 CALL KEY(5,A,B)
230 IF B<1 THEN 220
240 CALL LINK("PUTPEN",142,80,-30)
```

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

```
250 CALL LINK("CLEAR")
260 FOR I=1 TO 3
270 CALL LLL :: CALL LINK("TURN",60):: NEXT I
280 REM
290 REM
300 CALL XT
310 SUB LLL
320 CALL F(15)
330 CALL T(60)
340 CALL F(100)
350 CALL T(120)
360 CALL F(85)
370 CALL T(120)
380 CALL F(15)
390 CALL T(60)
400 CALL F(55)
410 CALL T(-120)
420 CALL F(85)
430 CALL T(120)
440 CALL F(15)
450 CALL T(60)
460 CALL F(100)
470 SUBEND
480 SUB F(X)
490 CALL LINK("FWD",X):: SUBEND
500 SUB T(X)
510 CALL LINK("TURN",X):: SUBEND
520 SUB XT
530 CALL LINK("PRINT",178,20,"ANY KEY ")
540 CALL KEY(5,A,B)
550 IF B<1 THEN 540
560 IF A=82 THEN RUN
570 IF A<>65 THEN 540
580 CALL LINK("INPUT",178,20,A$,14,"DSK1.PICTURE")
590 CALL LINK("PRINT",178,20,"  ")
600 CALL LINK("SAVEP",A$)
610 GOTO 530
620 SUBEND
```



==

REPORT ON ANNUAL MEET OF TI99/4A USERS GROUP (U.K.)

Held at Chester Northgate Arena on Saturday May 26th 1990

Held in one of the large sports halls of the Chester Northgate Arena, a modern air conditioned building, with comfortable seating and first rate PA equipment supplied by the management, the annual meet again confirmed the strong support of the TI99/4A in the UK by a small handful of users.

With a membership of between 140 and 170, scattered over the UK and with several living abroad, the meet was visited by maybe 50-odd members, who came together to elect group officials for the year, and to see the latest in software and hardware.

Unfortunately we were not able to see the Asgard Mouse, nor the long awaited extensions to Turbo Pasc 99, but on show were the Mechatronic mouse and 80 column peripheral, also a TI99/4A console connected to a Tatung Einstein Monitor. The now discontinued Tatung Einstein like the European version of the TI had a colour-difference signal output, and the monitor can be merely plugged into the TI console. The PAL TI99/4A does NOT have a colour composite video signal output nor an RGB output and cannot readily be connected to "normal" monitors!

The Geneve was represented and also on show was a demo of two TI consoles linked via RS232 to demo the various telecommunications packages, from Telco down to TE2.

The commercial support came from Martin Blythe, trading as DATABASE, from Bronfa, Llanbydder, Dyfed, SA40 9UB. Martin has taken over UK module stocks from Frances Parrish, and had a number of nice special offers for the show. Martin has a very good stock of some modules, such as SHAMUS, and perhaps some of our US friends (including US TI commercial supporters and user groups) may be interested in purchasing some of his titles in bulk. Among the rarities Martin had on offer was a speech synth for just thirty pounds, and a couple of PEBs.

There were some excellent second hand bargains on offer, and in the absence of support from our many unexpanded owners, these largely went unsold- examples include a TI Printer (Epson MX80 really) going for thirty pounds, stand alone RS232 interfaces for thirty pounds, green screen monitors to plug directly into the TI for thirty five pounds, and a double sided stand alone disk drive with power supply and cables for sixty five pounds. There were some nice mint condition books going very cheap too! Ample module supply for five pounds or less and some very rare ones for not much more.

One unexpanded owner went away the proud owner of a mini memory, a stand alone 32k ram, a full Editor Assembler pack (for only five pounds) and the rare Miner 2049er sideways module. Your scribe picked up a Munchman II module to add to the collection. There was even the excellent SuperSketch peripheral on sale! And lots of spare bits and cables and so on.

One display item was a photograph of young George Shaw (present with his favorite teddy bear Martin) measuring two inches by two inches, and two printouts produced on a TI99/4A, based on the photo, one measuring 7/8" by 5/8" and the other 8" by 8".

These represented an interesting beta test of a possible new service for TI owners which may be offered by a famous US based owner (who at present wishes anonymity)- the photograph was scanned on a PC compatible, and cleaned up using Paintbrush (in particular the background was removed).

Then the PC picture was translated to MacPaint format and transferred to the TI via RS232.

From here it could be printed with MacFlix or PixPro, and in fact was translated from MacPaint format to TI Artist format using PixPro.

The TI Artist picture was then printed as the 8x8" picture using SmArtcopy by Alexander Hulpke.

---> continued --->

The smaller picture was made by using SQUEEZER to reduce the TI Artist picture to quarter size (Squeezer provides a choice of four densities and is the only really usable reducer for pictures!). Then the small TI Artist pic was printed used Artist Photographic Vn 2, supplied with Harry Brashears Home Publishing on the 99/4A, Supplement #3.
It is to be noted that while the TI programs could reproduce the scanned photo in the correct aspect ratios, the PC could not- the two computers being used together was a result better than either could do on their own!

We welcome to the Committee a new publicity officer and a new cassette librarian- welcome to Phil and Nicky. And our very own UK programmer has now taken on official programming duties- if you are in need of assistance with Basic or Extended Basic programming please contact Mark, who is presently learning all about machine code with the mini memory. If your needs are more sophisticated, Mark would appreciate a gift of your spare PEB please!

Members are reminded that modules may be bought/sold/borrowed from our module librarian Edward Shaw, and books and publications may be borrowed from our publiations librarian Mike Curtis. If you have hardware needs of any description the people to contact are Mike Goddard, Richard Sierakowski (RSTS), or Database (Martin Blythe). Our chairman Gordon Pitt has continental contacts and may be able to obtain some of the hardware such as the now discontinued Mechatronic line (mouse, 80 column card, etc).

Attendance was possibly reduced by the splendid weather, the bank holiday weekend, and some awful (bank holiday) traffic on the motorways. Members who failed to attend did however miss some good buys and some good demos. We are well aware that the North West is not easy for some of you to get to and the Group is looking into the economics of getting something together in the wilds South of Watford Gap, possibly near to London... watch this space as they say!
Also give Peter Walker a ring if you have a venue which is i)very cheap, ii) preferably near to a railway station and iii) you could organise a local get together- perhaps the most difficult is getting enough rigs together to demo things! Peter has details of all members on a regional basis and can fairly quickly estimate how many live near enough to attend a local do.

Many thanks to Mike for organising our Annual event this year, and to the staff of the Northgate Arena who were of very great assistance.

See you next year...

=====================================

```
 X   X    BBBB       # 15
  X X     B   B
   X      BBBB       By
  X X     B   B      Jim
 X   X    BBBB       Swedlow
```

[This article originally appeared in the User Group of Orange County, California ROM]

RE-MAPPING THE KEYBOARD

You normally see CALL KEY(0,K,S). There are five other values for the first variable, the key unit. They remap your keyboard:

0 Keeps the keyboard in the same mode as the last time a CALL KEY was executed. If this is the key unit on the first CALL KEY in a program, you stay in 4A mode.

1 Splits the keyboard into two 2 smaller boards. Good for games.

3 Remaps the keyboard as a 99/4 (no A). CONTROL and lower case are inactive.

4 Pascal mode.

5 Remaps to the 99/4A mode with lower case and CONTROL active.

When you use 3, regardless of the position of SHIFT and ALPHA LOCK, all alphas return as upper case. The problem is that this condition continues until you do a CALL KEY with a different key unit. Try this:

```
10 CALL KEY(3,A,B) ::
   PRINT : :"Key Unit is 3"
20 INPUT A$ :: PRINT A$
30 CALL KEY(5,A,B) ::
   PRINT : :"Key Unit is 5"
40 INPUT A$ :: PRINT A$ ::
   GOTO 10
```

With ALPHA LOCK up, try inputting lower case letters and see what happens.

A key unit of 3 is very useful to make sure that only upper case alphas are caught by the CALL KEY. If you want lower case later in an INPUT or ACCEPT, however, you must remap the keyboard with another CALL KEY.

USER FRIENDLY/USER PROOF

When programming, you want your program to help the user. You also want to keep the user from crashing your program. Remember, the user will do most any fool thing. One area of vulnerability is inputting information thru INPUT and ACCEPT statements.

Lets say you want an integer between 1 and 9. Here are a number of ways you can input that number.

----> continued ---->

INPUT F can cause two problems. First, any number will be accepted. Second, if your user inputs anything but a number, you get: "WARNING: INPUT ERROR IN nnn TRY AGAIN".
This destroys your screen, scares your user and looks bad too.

ON WARNING can help. If you add ON WARNING NEXT the warning message will be suppressed but any number will be accepted. This coding is probably the best that can be done with INPUT F:

```
10 ON WARNING NEXT
20 INPUT F :: IF F<1 OR F>9 OR
   F<>INT(F) THEN 20
```

Bad values of F (0, 3.1, etc) will still cause the input prompt to be repeated and mess up your screen but you will get a good value in the end.

ACCEPT AT has a number of features that will help. With SIZE and VALIDATE you can avoid some problems:

```
10 ACCEPT AT(5,10)SIZE(1)
   VALIDATE(DIGIT)BEEP:F
```

Your user can still goof you up two ways. Zero is acceptable input and inputting a null will do strange things. SIZE(1) means no bigger than one character but it can be smaller!

ACCEPT F$ will help solve some more problems. Strings will be accepted. This coding is about as user proof as you can get:

```
10 DISPLAY AT(5,1):"1"
20 ACCEPT AT(5,1)SIZE(-1)
   VALIDATE(DIGIT)BEEP:F$ ::
   IF F$="" THEN 20 ELSE
   F=MAX(VAL(F$),1)
```

By making the SIZE value negative, whatever is on the screen at (5.1) will be the default value if ENTER is pressed.

About the only way I have found to mess this up is to have a non-numeric sitting at (5,1). VALIDATE works ONLY on which key is pressed, it assumes that you know what is on the screen!

MEMORY SAVERS

A variable name takes only its length in memory. <A> takes one byte while <MASTER_DEVICE> takes 13 bytes and <A$> takes two bytes.

A number used in a program line , however, takes the number of characters plus two bytes. For example, the number 2.13 would take six bytes of memory and the number 1 would take three bytes.

Strings also take the number of characters in the string plus two. "He won" takes eight bytes.

---> continued --->

Some suggestions:

--Use the shortest possible variable and sub-program names.

--If you use a given number often, assign it to a variable and use that instead.

Figure this way: A=2.13 takes eight bytes (one for A, one for = and six for 2.13). Each time you use A instead of 2.13 you save five bytes. Therefore, after two substitutions you are conserving memory.

-Look for places where you can replace numerics with variables. This line:

```
10 C=0 :: INPUT A(0) :: PRINT A(0)
```

Would take four less bytes if done this way:

```
10 C=0 :: INPUT A(C) :: PRINT A(C)
```

Be sure, however, that you only do this when the variable must be the number you intend it to be!!!!!

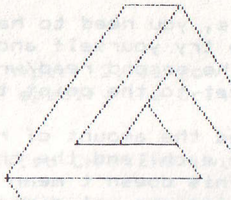
SPEED

Longer variable names slow program execution. I ran this program with progressively longer variable names substituted for <C>:

```
10 C=0 :: FOR I=1 TO 1000 :: A=C :: NEXT I
```

I ran each three times and averaged the results. Here is what I found:

LENGTH OF SUB- STITUTE FOR <C>	AVERAGE RUN TIME (seconds)
1	7.74
2	7.72
4	7.90
6	8.12
7	8.46
10	8.56
15	9.02



As you can see, variables up to three characters in length ran in about the same time. Once the length was longer, however, each additional character in the variable name increased the run time by about one tenth of a second for 1000 executions or .1 millisecond for one.

I also ran this one:

```
10 C=0 :: FOR I=1 TO 1000 ::
   A=0 :: NEXT
```

The average run time was 7.06 seconds. There is a cost when substituting variables for numbers. TANSTAAFL (there ain't no such thing as a free lunch)!!

----> continued ---->

DID YOU KNOW?

You can have more than one item for a VALIDATE. For example,

```
10 ACCEPT AT(5,1)VALIDATE(DIGIT,"Q")
   BEEP:F$
```

Will accept the ten digits and the letter Q. It does.

Enjoy.

==

TI BITS * Number 16 & 17

By Jim Swedlow

[This article originally appeared in the User Group of Orange County, California ROM]

ON DISKS AND DRIVES

A while back the Disk Doctor attended one of our meetings. He had a number of interesting things to say.

o Don't clean your drives until you need to. Your system will tell you when it is time - you will have trouble reading disks.

o When you do clean your drive, use any brand name commercial disk drive cleaner and follow instructions.

o If this fails, you need to have your drive cleaned professionally. If you want to try yourself and you have a double sided drive, be careful with the second read/write head. It is very, very easy to bend the bracket to the point that the head must be realligned.

o He has tested the amount of residue left on heads with brand name disks ($1.00 + each) and the cheepies ($0.25 or so). He found no difference. This doesn't mean that they are of equal quality, only that the cheepies are not dirtier.

o He opposes flippies for single side users. His point is that when you flip the disk and it runs backwards in its cover, dirt is loosened and spun into your drive.

o His overall advise is the first rule of engineering: If it ain't broke, don't fix it.

SOME MORE THOUGHTS ON BACKING UP DISKS

Over the years I have mentioned the importance of backing up your disks. Simply put, disk drives eat disks. On the first weekend of October, I was working on some letters. This was the weekend where the temperature was well over 100 degrees. I blew both my word processing disk and my data disk.

-----> more ----->

I had a backup of the word processor, but it was not configured. That night, after it cooled down a bit, it took me about half an hour to recreate a working disk. The data files were simply lost.

The moral? Keep two back ups of your program disks. One of the disk as you received it (the master) and one of your configured working disk (back up working disk). Don't forget to back up your data disks every now and then. This will save you time and aggravation next time your drive gets hungry.

TI WRITER'S INCLUDE FILE

One of TI Writers nicer features is Include File (.IF). It has a few limitations, but it extends TI Writers capabilities.

TI Writer cannot work on large files. No books in one file here. As you reach the size limit, the time it takes to load and save files increases markedly. Include File to the rescue.

Suppose your have written two chapters of your next book. Your named your files CHAPTER1 and CHAPTER2 (very original). At the end of Chapter 1 (the very last line), add this:

```
.IF DSK1.CHAPTER2
```

Name CHAPTER1 for the Formatter and it will print both chapters. All the formatting commands you set for Chapter 1 will be used when Chapter 2 is printed, so you don't have to restate the margins and such.

Ah, you finish Chapter 3. No problem. At the end of Chapter 1, add another line:

```
.IF DSK1.CHAPTER3
```

You cannot do this at the end of Chapter 2, as you can't chain these commands. Also note that you must specify the drive number (DSK1 in this case).

I prefer to make a master file (called CHAPTER0) will all of the .IF commands:

```
.IF DSK1.CHAPTER1
.IF DSK1.CHAPTER2
.IF DSK1.CHAPTER3
```

Before (not after) your .IF lines, put in your format, header and footer instructions. Now you have all of your format commands in one place that is easy to find and edit.

[later:]

OUCH

Every once in a while I goof and someone catches me. This time it was in TI BITS Number 16 in which I said that you can't chain Include File (.IF) commands in TI Writer.

-----> read on--->

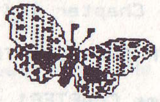
That is what the manual says and I believed the written word. Not so, says UGOC Pres Bob Harper. He is correct. You can end each file with a .IF command for the next file. I still prefer however, a master file that has all of the .IF commands. (Does this apply to all versions of TI Writer I wonder? Advise Stephen please!)

Stephen here... I have tested the above retraction with:
FUNLWEB 4.21
RAG WRITER 4.5
BA WRITER 1.4
and they ALL work in accordance with the manual, refusing to allow you to stack .IF's and insisting that all .IFs be in the one calling file. SO... under what circumstances did Bob Harper and Jim Swedlow beleive otherwise? Please test all your combinations of TI Writer compatibles and let me know!

===============

The following is a reprint of a tutorial which was printed by the Group quite some time back. Reprinted for new ExBas programmers:

EXTENDED BASIC TUTORIALS
from
FUNNELWEB FARM (Australia)



I. INTRODUCTION

In this series of notes on TI Extended Basic for the TI-99/4a we will concentrate on those features which have not received due attention in User-group newsletters or commercial magazines. In fact most of the programs published in these sources make little use of that most powerful feature of XB, the user defined sub-program, or of some other features of XB. Worse still is to find commercially available game programs which are object lessons in how to write tangled and obscure code. The trigger for this set of tutorial notes was a totally erroneous comment in the TI.S.H.U.G Newsdigest in Jun 1983. Some of the books I have seen on TI Basic don't even treat that simpler language correctly, and I don't know of any systematic attempts to explore the workings of XB. The best helper is TI's Extended Basic Tutorial tape or disk. The programs in this collection are unprotected and so open for inspection and it's worth looking at their listings to see an example of how sub-programs can give an easily understood overall structure to a program.

Well, what are we going to talk about then ? Intentions at the moment are to look at

(1) User-defined sub-programs
(2) Prescan switch commands
(3) Coding for faster running
(4) Bugs in Extended Basic
(5) Crunching program length
(6) XB and the Peripheral Box
(7) Linking in Assembler routines

---> continued---->

Initially the discussion will be restricted to things which can be done with the console and XB only. Actually, for most game programming the presence of the memory expansion doesn't speed up XB all that much as speed still seems to be limited by the built-in sub-programs (CALL COINC, etc) which are executed from GROM through the GPL interpreter. The real virtue of the expansion system for game programming, apart from allowing longer programs, is that GPL can be shoved aside for machine code routines in the speed critical parts of the game, which are usually only a very small part of the code for a game. Even so careful attention to XB programming can often provide the necessary speed. As an example, the speed of the puck in TEX-BOUNCE is a factor of 10 faster in the finally released version than it was in the first pass at coding the game.

II. SUB-PROGRAMS in OVERVIEW

Every dialect of Basic, TI Extended Basic being no exception, allows the use of subroutines. Each of these is a section of code with the end marked by a RETURN statement, which is entered by a GOSUB statement elsewhere in the program. When RETURN is reached control passes back to the statement following the GOSUB. Look at the code segments

```
290 ....
300 GOSUB 2000
310 ....
 .
 .
2000 CALL KEY(0,X,Y):: IF Y=1 THEN RETURN ELSE 2000
```

This simple example waits for and returns the ASCII code for a fresh keystroke, and might be called from a number places in the program. Very useful, but there are problems. If the line number of the subroutine is changed, other than by RESequencing of the whole program (and many dialects of Basic for microcomputers aren't even that helpful) then the GOSUBs will go astray. Another trouble, which you usually find when you resume work on a program after a lapse of time, is that the statement GOSUB 2000 doesn't carry the slightest clue as to what is at 2000 unless you go and look there or use REM statements. Even more confusingly the 2000 will usually change on RESequencing, hiding even that aid to memory. There is an even more subtle problem -- you don't really care what the variable "Y" in the subroutine was called as it was only a passing detail in the subroutine. However, if "Y" is used as a variable anywhere else in the program its value will be affected. The internal workings of the subroutine are not separated from the rest of the program, but XB does provide four ways of isolating parts of a program.

(1) Built-in sub-programs
(2) DEF of functions
(3) CALL LINK to machine code routines
(4) User defined BASIC sub-programs

----> continued---->

The first of these, built-in sub-programs, are already well known from console Basic. The important thing is that they have recognizable names in CALL statements, and that information passes to and from the sub-programs through a well defined list of parameters and return variables. No obscure Peeks and Pokes are needed. The price paid for the power and expressiveness of TI Basic and XB is the slowness of the GROM/GPL implementation.

DEF function is a primitive form of user defined sub-program found in almost all BASICs. Often its use is restricted to a special set of variable names, FNA,FNB,... but TI Basic allows complete freedom in naming DEFed functions (as long as they don't clash with variable names). The "dummy" variable "X" is used as in a mathematical function, not as an array index

```
          100 DEF CUBE(X)=X*X*X
```

doesn't clash with or affect a variable of the same name "X" elsewhere in the program. "CUBE" can't then be a variable whose value is assigned any other way, but "X" may be. Though DEF does help program clarity it executes very slowly in TI Basic, and more slowly than user defined sub-program CALLs in XB.

CALL LINK to machine code routines goes under various names in other dialects of Basic if it is provided (eg USR( ) in some). It is only available in XB when the memory expansion is attached, as the TI-99/4a console has only 256 bytes of CPU RAM for the TMS9900 lurking in there. We will take up this topic later.

You should have your TI Extended Basic Manual handy and look through the section on SUB-programs. The discussion given is essentially correct but far too brief, and leaves too many things unsaid. From experiment and experience I have found that things work just the way one would reasonably expect them to do (this is not always so in other parts of XB). The main thing is to get into the right frame of mind for your expectations. This process is helped by figuring out, in general terms at least, just how the computer does what it does. Unfortunately most TI-99/4a manuals avoid explanations in depth presumably in the spirit of "Home Computing". TI's approach can fall short of the mark, so we are now going to try to do what TI chickened out of.

The user defined sub-program feature of XB allows you to write your own sub-programs in Basic which may be CALLed up from the main program by name in the same way that the built-in ones are. Unlike the routines accessed by GOSUBs the internal workings of a sub-program do not affect the main program except as allowed by the parameter list attached to the sub-program CALL.

Unlike the built-in sub-programs which pass information in only one direction, either in or out for each parameter in the list, a user sub-program may use any one variable in the list to pass information in either direction. These sub-programs provide the programming concept known as "procedures" in other computer languages, for instance Pascal, Logo, Fortran.

---> continued--->

The lack of proper "procedures" has always been the major limitations of BASIC as a computer language. TI XB is one of the BASICs that does provide this facility. Not all BASICs, even those of very recent vintage are so civilised. For example the magazine Australian Personal Computer in a recent issue (Mar 84) carried a review of the IBM PCjr computer just released in the US of A. The Cartridge Basic for this machine apparently does not support procedures. Perhaps IBM don't really want or expect anyone to program their own machine seriously in Basic. You will find that with true sub-programs available, that you can't even conceive any more of how one could bear writing substantial programs without them (even within the 14 Kbyte limit of the unexpanded TI-99/4a let alone on a machine with more memory).

The details of how procedures or sub-programs work vary from one language to another. The common feature is that the variables within a procedure are localised within that procedure. How they communicate with the rest of the program, and what happens to them when the sub-program has run its course varies from language to language. XB goes its own well defined way, but is not at all flexible in how it does it.

Now let's look at how Extended Basic handles sub-programs. The RUNning of any XB program goes in two steps. The first is the prescan, that interval of time after you type RUN and press ENTER, and before anything happens.

During this time the XB interpreter scans through the program, checking a few things for correctness that it couldn't possibly check as the lines were entered one by one, such as there being a NEXT for each FOR. The TI BASICs do only the most rudimentary syntax checking as each line is entered, and leave detailed checking until each line is executed. This is not the best way to do things but we are stuck with it and it does have one use.

At the same time XB extracts the names of all variables, sets aside space for them, and sets up the procedure by which it associates variable names with storage locations during the running of a program. Just how XB does this is not immediately clear, but it must involve a search through the variable names every time one is encountered, and appears to trade off speed for economy of storage.

XB also recognizes which built-in sub-programs are actually CALLed. How can it tell the difference between a sub-program name and a variable name? That's easy since built-in sub-program names are always preceded by CALL. This is why sub-program names are not reserved words and can also be used as variable names.

This process means that the slow search through the GROM library tables is only done at pre-scan, and Basic then has its own list for each program of where to go in GROM for the GPL routine without having to conduct the GROM search every time it encounters a sub-program name while executing a program. In Command Mode the computer has no way provided to find user defined sub- program names in an XB program in memory even in BREAK status. XB also establishes the process for looking up the DATA and IMAGE statements in the program.

----> continued--->

Well then, what does XB do with user sub-programs? First of all XB locates the sub-program names that aren't built into the language. It can do this by finding each name after a CALL or SUB statement, and then looking it up in the GROM library index of built-in sub-program names. You can run a quick check on this process by entering the one line program

```
100 CALL NOTHING
```

TI Basic will go out of its tiny 26K brain and halt execution with a BAD NAME IN 100 error message, while XB, being somewhat smarter, will try to execute line 100, but halts with a SUBPROGRAM NOT FOUND IN 100 message.

The XB manual insists that all sub-program code comes at the end of the program, with nothing but sub-programs after the first SUB statement (apart from REMarks which are ignored anyway). XB then scans and establishes new variable storage areas, starting with the variable names in the SUB xxx(parameter list), for each sub-program from SUB to SUBEND, as if it were a separate program. It seems that XB keeps only a single master list for sub-program names no matter where found, and consulted whenever the interpreter encounters a CALL during program execution. Any DATA statements are also thrown into the common data pool. Try the following little program to convince yourself.

```
100 DATA 1
110 READ X :: PRINT X :: READ X :: PRINT X
120 SUB NOTHING
130 DATA 2
140 SUBEND
```

When you RUN this program it makes no difference that the second data item is apparently located in a sub-program. IMAGEs behave likewise. On the other hand DEFed functions, if you care to use them, are strictly confined to the particular part of the program in which they are defined, be it main or sub. During the pre-scan DEFed names are kept within the allocation process separately for each subprogram or the main program. Once again try a little programming experiment to illustrate the point.

```
100 DEF X=1 :: PRINT X;Y :: CALL SP(Y) :: PRINT X;Y
110 SUB SP(Z) :: DEF X=2 :: Z=X :: DEF Y=3
120 SUBEND
```

This point is not explicitly made in the XB manual and has been the subject of misleading or incorrect comment in magazines and newsletters. A little reflection on how XB handles the details will usually clear up difficulties.

TI BASICs assign nominal values to all variables mentioned in the program as part of the prescan, zero for numeric and null for strings, unlike some languages (some Basics even) which will issue an error message if an unassigned variable is presumed upon.

----> coninued---->

This means that XB can't work like TI LOGO which has a rule that if it finds an undefined variable within a procedure it checks the chain of CALLing procedures until it finds a value. However, unlike Pascal which erases all the information left within a procedure when it is finished with it, XB retains from CALL to CALL the values of variables entirely contained in the sub-program. The values of variables transferred into the sub-program through the SUB parameter list will of course take on their newly passed values each time the sub-program is CALLed. A little program will show the difference.

```
100 FOR I=1 TO 9 :: CALL SBPR(0):: NEXT I
110 SUB SBPR(A):: A=A+1 :: B=B+1 :: PRINT A;B
120 SUBEND
```

The first variable printed is reset to 0 each time SBPR is called, while the second, B, is incremented from its previous value each time. Array variables are stored as a whole in one place in a program, within the main program or sub-program in which the DIMension statement for the array occurs. XB doesn't tolerate attempts to re-dimension arrays, so information on arrays can only be passed down the chain of sub-programs in one direction.

Any attempt by a XB sub-program to CALL itself, either directly or indirectly from any sub-program CALLed from the first, no matter how many times removed, will result in an error. Recursive procedures, an essential part of TI LOGO, are NOT possible with XB sub-programs , since CALLing a sub-program does not set up a new private library of values.

Another simple programming experiment will demonstrate what we mean by saying that XB sets up a separate Basic program for each sub-program. RUN the following

```
100 X=1 :: CALL SBPR :: BREAK
110 SUB SBPR :: X=2 :: BREAK :: SUBEND
```

When the program BREAKs examine the value of variable X by entering the command PRINT X, and then CONtinue to the next program BREAK, which this time will be in the main program, where you can once again examine variable values.

We will now summarize the properties of XB sub-programs as procedures in complete XB programs, leaving the details of joining up the various procedures to the next section.

(a) XB treats each sub-program as a separate program, building a distinct table of named (REFed) and DEFed variables for each.

(b) All DATA statements are treated as being in a common pool equally accessible from all sub-programs or the main program as are also IMAGE statements, CHARacters, SPRITEs, COLORs, and File specifications.

------> more ------>

(c) All other information is passed from the CALLing main or sub- program by the parameter lists in CALL and SUB statements. XB does not provide for declaration of common variables available on a global basis to all sub-programs as can be done in some languages.

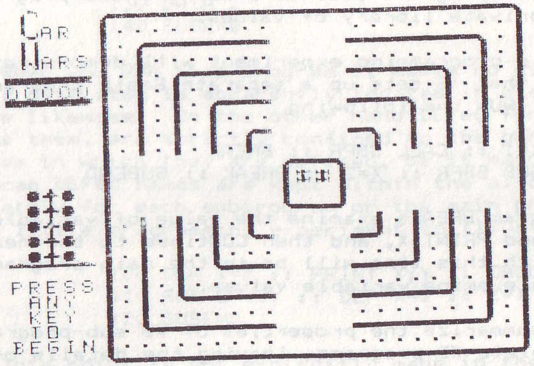
(d) Variable values confined within a sub-program are static, and preserved for the next time the sub-program is CALLed. Some languages such as Pascal delete all traces of a procedure after it has been used.

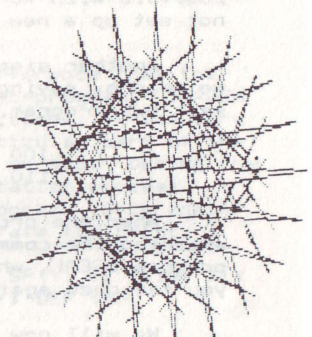
(e) XB sub-programs may not CALL themselves directly or indirectly in a closed chain. Subject to this restriction a sub-program may be CALLed from any other sub-program.

(f) The MERGE command available in XB with a disk system (32K memory expansion optional) allows a library of XB sub-programs to be stored on disk and incorporated as needed in otherprograms.

ARTICLE BY TONY MCGOVERN

[Stephen here again. Want the next part? WRITE TO ME AND TELL ME or you may not see any more!!! I need to know your requirements!]





RAG MACROASSEMBLER IS NOW INTO VERSION 8. AMENDMENTS:
BUGS FIXED.
SBO, SBZ and TB assembled incorrectly when the displacement was negative.
Minor corrections to docs.
=========
SIDE*PRINT by Jim Swedlow now to Version 3.4, changes not documented but docs are larger.
=========
TI ARTIST PLUS! into Version 1.03, changes undocumented.
=========
TI BASE was at the time of writing this into Vn 2.04
This version is compatible with the hard drive systems.. There is a new loader named TIBASEW which will load with a program image loader. The path in this loader is set to WDS1.TIB. The path name may be changed by running @PATH and following instructions.
Remember that the total filename is still limited to 20 characters.
Version 2.02 corrects several minor discrepancies as follows:
o Stopped truncation of X type variables with trailing 0's.
o Fixed edit/append so fields of 25 char didn't allow extra character input.
o Fixed SUM with ;FOR clause for problems - exited command files because cmd line number was destroyed. - Print output was bad due to cr/lf contents destroyed. - editing and appending date fields incorrect due to / constant destroyed.
o Fixed ;FOR clause in print and display so temporary locals get cleaned up.
o Fixed catalog so it works on disks with no name.
o Fixed REPLACE with ;FOR clause when sort field was changed and literals were used.
o Made compatible with hard drive.
Version 2.03 fixed several minor problems in V2.02:
o occasionally a "bad name" message was generated when printing or accessing a file.
o When trying to access a non-existant command file, a garbled message was presented.
o When using literals in a REPLACE directive and there was not enough local space available, an error message was generated, but the directive did not abort, causing random results such as blank screens. Three loaders are provided:
o TIBASE for load and run options o TIBASEP for program image option o TIBASEB for extended BASIC
Version 2.04 corrected a problem in using command files on the MYARC Ram-Disk. It also corrected a problem in nested if's and while's which had been present since the inception of TI-BASE.
======
Updates of all the above are available from the Disk Library at usual rates- to avoid copyright problems please make sure the forward the original purchased master disks for either TI Artist Plus! or TI Base!
======
* * * TI BASE Vn 3 * * *
Announcement of major upgrade to TI Base- TI Base is now into Version 3, with a whole passel of changes, described fully in MICROpendium May 1990, filling page 42. In brief: Command files can

(continued------->)

be in VDP for faster processing; new Report Generator; single key directive entry option; greater use of variables possible; CLOSE more intelligent; DELETE and RECALL now have ;FOR , new midstring comparison (to catch THE BOY with BOY) and a SEG$ equivalent operator and LEN operator. CALL KEY equivalent. SUM and AVERAGE now available for storage. Literals can now by PRINTed and DISPLAYed (hurrah!); WRITE inverse option; new APPEND FROM.... owners of any Vn 2 can upgrade by returning their two (Note: TWO) original purchased disks with US$23 to Texaments, or £16.00 to Stephen Shaw.

@@

MAJOR NEW FRACTAL BOOK FORTHCOMING JUNE 28TH:

Computers, Pattern, Chaos and Beauty by Dr Clifford Pickover is to be published on June 28th by Alan Sutton Publishing, Phoenix Mill, Far Thrupp, STROUD, Glos, GL5 2BU. ISBN 0 862 99 792 5, price £25.00

This book will be 246mm x 172mm, and run to 400 pages of recipes and algorithms to use, with 200 mono illustrations and 8 pages of colour pics. If ordering from the publisher add £1.50 postage. A must for any fractal programmer

======

L L Conner has indicated that the long awaited extensions to Turbo Pasc 99 are now due any day. Watch out next issue for a review of these and the above book!

========

Harrison Software announce upgrade of their Word Processor to Vn 2.0 with a free upgrade to owners of Vn 1 - just return your original purchased disk to Stephen Shaw with a couple of pounds for air mail postage and allow a few weeks for the return cross Atlantic post! The new version allows more flexible document and drive naming, amended cursor movement, more intelligent page saving; allow panel select NLQ on those printers with it; multi copy prints; and faster loading.

Harrison also announce they are ceasing to support the Geneve due to constantly changing operating systems and languages, and the hotch potch of systems users are ending up with due to the methods used to announce and distribute upgrades.

==========

BOOKS ON SALE BY REGENA:

Prices include an estimated US$4 towards postage:
COMPUTE!s Programmers Reference Guide to the TI99/4A..US$19.00
COMPUTE!s First Book of TI Games. US$17.00

Programs on DISK:
TYPE-ETTE- Six programs to learn touch typing. US$10.00
STATES- Five programs to learn States and Capitals. US$10.00
MISCELLANEOUS- 11 progs re state flags, quilt, learn bones, etc.

All programs published in MICROpendium for the TI are available for $4 each plus $5 per order postage.
Payment must be in US$ only please.
Write stating TI99/4A to:
REGENA, 918 Cedar Knolls West, Cedar City, Utah, USA, 84720.

===============

MODULE REVIEW- 4A FLYER- GAME-DATABIOTIC

The author was told to write a flight simulator within a tight time limit, in machine code, to fit entirely within 8k. Under the circumstances it is probably not a bad job, but it is instruments ONLY, and if you point your plane at the ground, there comes a point where you reach a maximum speed even with the throttle wide open - and it is not that fast a speed. There are various anomolies and the instrument only flying gets rather boring quite quickly. For very little more (in dollar terms anyway) you could have the so very much superior Spad XIII, so get that one and enjoy some scenery!

====

MODULE REVIEW- FROGGER- PARKER BROS.

This module is an excellent reproduction of the very popular (well, some years ago...) classic arcade game. It plays very much like the arcade machine. What more praise can there be! If you like Frogger, buy it.

HMMM... dont remember Frogger eh! You must guide a frog over several lanes of busy road without being squashed, THEN across a river, without falling in, being eaten by crocs, or dumped by submerging turtles... and so on.

=====

MODULE REVIEW- D STATION 1- DATABIOTICS.

This started off life as a disk program from the Bethany based International User Group, and is a VERY simple program, although well written. From a fixed base at screen centre you must shoot down flyers and parachutists. Thats about it. Graphics are extremely barren but the game can be difficult. You aim by rotating your gun (180 degree capability) which makes a change from moving left and right at screen bottom.

======

MODULE REVIEW- DEFENDER- ATARISOFT.

This is possibly the best Atarisoft module, and is a first rate emulation of the classic arcade game, with a small wide range radar scan at screen top showing where all aliens are, and a playing area to move your ship around while being attacked from all sides from all sorts. The play can become really frantic. This module should be in your game collection. Supplies are generally good but you may need to pay a slight premium price.

=========

MODULE REVIEW- TI INVADERS- TI MODULE

Yet another classic arcade game, which may start off a little too slowly for todays hardened games player, but a quick check of the museum piece arcade machines shows that they too start very slowly and take a while to speed up- this does NOT make it easy to clear the first screen incidentally!

This TI Module is almost certainly the very best version of Invaders on ANY home computer. The graphics are if anything better than on the arcade machines, and all the arcade play is there.

This is one of only two modules that TI officially released on disk- the disk version had a TEST mode not in the module which allowed you to really slow the game down, and to start at any level UP TO LEVEL 50 (and it went on after level 5C too). Another module which you really should have in your collection if you dont have it. You may not play it too much, but it is a lovely example of what TI could do. Not hard to find.

MODULE REVIEW- JUMPY - TI MODULE

Possibly the ONLY TI Module actually WRITTEN on a TI99/4A system- the others were developed and assembled on larger 9900 computers. You control a cute little Kangaroo who pushes boxes around trying to trap monsters- trapped monsters die and score points. The game becomes surprisingly difficult at level ten. Nice game, fairly difficult, good game action. Reasonable graphics.

============

MODULE REVIEW- MS PACMAN- ATARISOFT.

Remarkably like PacMan... Ms PacMan if you forgot is the one with the ribbons! This version has a faster screen set up (PacMan as a module for the TI had a very slow screen set up!) and more varied screen layouts. Also introducing the bouncing strawberry -so much more difficult to collect than a static bonus fruit. You eat dots and avoid monsters in a maze, clear all the dots for next maze, go over power points to temporarily gain the strength to shoo the monsters back home.

Better than PacMan, comparable to MunchMan, which seems to play faster.

==========

MODULE REVIEW- MUNCHMAN -TI MODULE

Way back when, when PacMan was all the rage and PacMan Fever failed to make the pop charts, TI were thinking of modules for their new Home Computer. A variant of PacMan came out, and Atari jumped on it hard. Gobbling dots was OUT! So a quick rewrite and we instead eat up a trail while the requisite monsters chase us around the maze! This one can get incredibly fast, and you need some pretty good joysticks to play well. If you dont have it, well worth trying.

==========

MODULE REVIEW: MUNCHMAN II. A Triton/Databiotics module. 1987.

Source: Try offering Database a suitable bribe to import some or try TexComp. Supplies scarce.

It has been brought to my attention that some TI owners- especially disk owners- have entirely the wrong impression of what this module is! You will see from the above review that the original Munchman module was actually version 2 - which makes this one Version 3!

This is a complete rewrite by (former?) TI employee John Phillips, responsible for so much module work. Here he has married the two connected screens of Sewermania with the first Munchman variant in which instead of laying a trail you remove one! Yes there is a screen to the right of the first one which you enter through two passages- OR you can use a roving teleport machine! And you must clear BOTH screens to go on! The hoonos are FASTER than you are so you cannot outrun them - except after going over a power dot when you can outrun them and gain good points! You must use skill and tactics in this version. A worthwhile addition to any collection of PacMan variants!

=========

REVIEW-HOME PUBLISHING ON THE 99/4A SUPPLEMENT #3

- 32page booklet plus two disks
-Harry T Brashear, 2753 Main Street, NEWFANE, NY, USA, 14108
-US$15 including post and packing.

Another good title would be "TI Artist Plus! manual, part two" or "What was missed out!".

This booklet had me reaching for my TIA+ manual, then loading TIA+ to check Harry's booklet... wow. I don't have 32 pages to spare, but if you have TI Artist plus! you really do need this booklet, it'll tell you LOTS you need to know, as well as better illustrating and demonstrating some of the more awkward features so briefly touched on in TIA+ manual. Artisles are by Harry himself and by Paul E Scheidemantle (see the Tiger article in TI*MES issue 24, page 45). Articles cover Vectors, with much better coverage for instance of spin, tilt, and ranging. Harry - who is also not 100% in favour of the printout section- gives good examples of its usage. Other aspects of the package are covered also!

Also in this package is a review of Guidelines (see my review later!) and also an article on PagePro Vn 1.5. Harry thinks paperbacks have 55 characters per line. Mine don't! Harry also mentions PixPro (old discontinued version) and PagePro Utilities.

The disks? Yes you get disks too... one disk contains a large font in the form of instances- to be used as you see in magazines and old manuscript, where the first letter of a chapter or even paragraph is so much larger than the rest. And six sets of borders in Font format. And a nice new Shadow font. And four utilities:

Artist Photographic Vn 2 prints out TIA pics in good proportion and good density, in four different sizes, with a total of sixteen different page positions- 7 for the smallest pic and one for the largest! For the two smaller sizes pics can be printed in one pass side by side. Very good output, tightly controlled (ditch that TIA+ print routine!)..!!! Compatible with 24 pin printers as it allows line spacing in 144ths, 216ths, as well as 180ths of an inch.

Page Pro Hi Res Printer- to print Page Pro saved pages including clipped pages, using a higher resolution than Page Pro. by moving the paper up 1/216th of an inch before overstriking. Some fine detail may be lost in the process.

Picture Reference Sheets- This goes through a disk and prints out all the TI Artist or Graphx pics in one go, with either 20 pictures to the page or six- your choice. Filename is printed as well for a handy reference. The prints are also of pretty high quality!

Font Spacer... unhappy with the spacing between TIA fonts - or the lack of it on fonts converted from CSGD format? This program will go through the entire font and insert a standard spacing chosen by you. You may also merge font files, to take advantage of TIA+ ability to handle the larger files!

This package represents a heck of a good value for money. All TI Artist users need it!

============

Review- Guidelines (Disk)
- from Texaments, 53 Center St, Patchogue, NY, USA, 11772.
- US$9.95 plus p&p US$8

A two disk set especially for users of TI Artist plus. A big selling point is the TEMPLATES which is as much a condemnation of TIA+ print routines as anything else! The Banner Template marks out the part of the screen that TIA+ Banner option chooses to print, so you know where to put your banner material. Pagetemp marks off the outer columns which TIA+ chooses not to print. Label and disk jacket templates would be more useful if we were told what print options to use, and if all printers produced the same sizes for these options... The calendar template may find favour.

There are twelve banner borders, all of which are of a fixed size and design, and assume your banner is going to be the FIXED banner size of around two and a half pages (there are better banner utilities around! and you can of course use these frames with them also).
There are 15 fonts, all SMALL one char high apart from one two char high. Very basic simple fonts. There are two sets of slides (anyone ever use these???) and 17 incredibly tiny instances.

What I found the most interesting content is just listed on page one, unillustrated and not further mentioned, a font file called SHAPES. Anyone remember Peter Brooks "Designs for Fun" program? SHAPES is similar- every keyboard key (except ; for some reason) will produce a single character design. These can produce a myriad of patterns for borders, rules, fills, what have you. Using one shape, you can vary the result by positioning subsequent shapes one or two pixels higher or lower, or moving alternate rows left or right, and so on. If you use two shapes together in a pattern, you have even more possibilities. Then add rotate and other vector effects... you have the ability to fairly rapible create unique patterns. This little file made the package worth having for me!!
Postage apart, 9.95 for two disks isnt bad, and TIA+ users may well find this a useful package. I do hope we dont see a flood of pictures with the first and last columns blank though!
=========
REVIEW- THE MISSING LINK. (DISK).
- from TEXAMENTS (see above). -US$25 plus post and packing US$8.

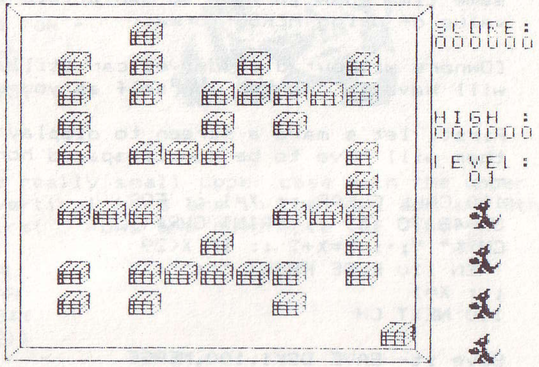
This one disk is a really useful extension to XB for those of us who still program in XB. What it does is place the screen into bit map mode while a program is running, allowing you to do all sorts of clever things with bit map graphics. Naturally the old screen commands are useless in bit map mode, but in providing bit map equivalents, we find very much more powerful versions. And new items like windows, which allow for instance a VERTICAL input field! And a whole range of different sized fonts can be used, on screen together, in different colours... you name it! The smallest font supplied allows for 60 columns per screen line, and is remarkably readable on a black and white tv. The results on a colour tv will be variable.
The disk library has a free demo of this program available which shows off its talents, so no excuses for not having a look, if you have a disk drive.

---->continued...-->

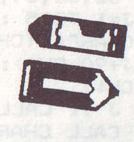
The actual program is much easier to load than the demo disk which has been deliberately crippled.
Drawbacks? The utility is interrupt driven - this allows you to drop out of the bit map screen when the program is not running, and allows a screen to be printed at any time (press CTRL and FCTN). This means it will not function in Myarc XB (which has a bit map screen available anyway). You can use Triton Super XB provided you do not use the clock.
Bit map screens occupy 12k of vdp ram. This only allows 4k of vdpram for handling disk loading and saving, and for storage of variables. The limitation on variables will not affect many programs, and effective program writing will almost always solve any problems you meet - eg using shorter variable and subprogram names, not using new variables when you dont have to and so on.

The VDP of course now cannot load Extended Basic programs other than really tiny ones if they are in PROGRAM disk format, using the form RUN "DSK1.PROG", and such programs must be converted to IV254 form, using a utility provided. Loading and saving in command format are not affected, as you are then in standard graphics mode with only 4k used for the screen display. TML is now my favoured graphics language. Send for the demo disk- just a blank disk and return post and packing to the Disk Library!

And even if graphics is not your scene, the more powerful functions of TML - regarding input fields, sprites, and so on- may stilll make it a very useful package for you. Take a llok at the demo disk!
====



HOPPER



TIPS FROM THE TIGERCUB
#56

Tigercub Software 156 Collingwood Ave. Columbus OH 43213

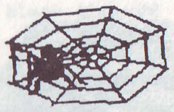
The contents of the first 52 issues of this news letter are available as ready-to-run programs on 5 Tips Disks at $10 each. And my three Nuts & Bolts Disk, $15 each, each contain over 100 subprograms for you to merge into your own programs to do all kinds of wonderful things.
****
TI-PD LIBRARY
I have selected public domain programs, by category, to fill over 200 disks, as full as possible if I had enough programs of the category, with all the Basic-only programs converted to XBasic, with an E/A loader provided for assembly programs if possible, instructions added and any obvious bugs corrected, and with an auto loader by full program name on each disk. These are available as a copying service for just $1.50 post paid in U.S. and Canada. No fairware will be offered without the author's permission. Send SASE for list or $1, refundable, for 9-page catalog listing all titles and authors. Be sure to specify TI-PD catalog.
****

In Tips #55, I showed you some quick and easy ways to create new character sets. Since folks nowadays don't like to key in long programs, let's continue with "tinygram" programming, and at the same time show you how to manipulate strings, and teach you the value of using MERGE format.

[Owners without disk drives can still use these routines but you will have to "merge" yourself as you enter the routines].

First, let's make a screen to display our new characters. Some of them will have to be double-spaced horizontally or vertically, so -

```
100 CALL CLEAR :: X=1 :: FOR
CH=48 TO 159 :: PRINT CHR$(
CH)&" ";:: X=X+2 :: IF X<29
THEN 110 ELSE PRINT "":"":""
;:: X=1
110 NEXT CH
```



Save it- SAVE DSK1.100,MERGE

Now, you might like to move the common punctuation marks into the same character sets as the characters, so that you will not have to reidentify so many sets, also so you can color them easier.

```
120 DATA 32,33,34,44,46
130 FOR J=1 TO 5 :: READ CH
:: CALL CHARPAT(CH,CH$):: CA
LL CHAR(J+90,CH$):: CALL CHA
R(J+122,CH$)
140 NEXT J :: CALL CHARPAT(6
3,CH$):: CALL CHAR(64,CH$)::
:: CALL CHAR(96,CH$)
```

---->more--->

If you want to program in Basic, or use BXB with characters all the way up to ASCII 159, add CALL CHAR(J+1 54,CH$) to the end of line 130 and CALL CHAR(128,CH$) to the end of line 140.

Save by SAVE DSK1.120,MERGE
If you are using that transliteration, you must remember that with upper case characters the ? is @, space is [, ! is \, " is], comma is , period is _. With the lower case they are FCTN keys C, F, A, G, W and V and for the 3rd set (ASCII 129 to 154) they are CTRL comma, period,;,=,* and (.
You can transfer upper case to lower by - CALL CHARPAT(CH,CH$) and then CALL CHAR(CH+32,CH$) or the opposite by CH-32 and if you have BXB merged in you can create a 3rd set by CH+64.

The following are all incompatible with each other, so give them all line number 150 and save them in merge format as 150A, 150B, etc.
The numerals and the upper case letters all have the topmost pixel row blank to provide spacing between lines of text. We can make taller letters by deleting the top row and doubling the 7th row -

```
150 FOR CH=48 TO 126 :: CALL
CHARPAT(CH,CH$):: CALL CHAR
(CH,SEG$(CH$,3,12)&SEG$(CH$,
13,4)):: NEXT CH
151 REM
```

Or, you can double the 3rd row -

```
150 FOR CH=48 TO 95 :: CALL
CHARPAT(CH,CH$):: CALL CHAR(
CH,SEG$(CH$,3,4)&SEG$(CH$,5,
12)):: NEXT CH
151 REM
```

The lower case letters are really small upper case with the upper 3 rows blank. All their vertical bars are in the 4th, 6th and 8th rows, so let's drop the first 3 rows and quadruple the 7th.

```
150 FOR CH=97 TO 127 :: CALL
CHARPAT(CH,CH$):: CALL CHAR
(CH,SEG$(CH$,7,6)&RPT$(SEG$(
CH$,13,2),4)&SEG$(CH$,15,2))
:: NEXT CH
151 REM
```

Or, for topheavy letters, quadruple the 5th row -

```
150 FOR CH=97 TO 127 :: CALL
CHARPAT(CH,CH$):: CALL CHAR
(CH,SEG$(CH$,7,2)&RPT$(SEG$(
CH$,9,2),4)&SEG$(CH$,11,6)):
: NEXT CH
151 REM
```

--->more---->

Or, if you want line spacing -

```
150 FOR CH=97 TO 122 :: CALL
CHARPAT(CH,CH$):: CH$=SEG$(
CH$,5,8)&RPT$(SEG$(CH$,13,2)
,3)&SEG$(CH$,15,2):: CALL CH
AR(CH,CH$):: NEXT CH
151 REM
```

Or, for something silly -

```
150 FOR CH=48 TO 90 :: CALL
CHARPAT(CH,CH$):: CALL CHAR(
CH,SEG$(CH$,3,2)&RPT$(SEG$(C
H$,5,2),4)&SEG$(CH$,9,4)&SEG
$(CH$,15,2)):: NEXT CH
151 REM
```

For some good blocky characters -

```
150 FOR CH=48 TO 90 :: CALL
CHARPAT(CH,CH$):: CALL CHAR(
CH,RPT$(SEG$(CH$,3,2),2)&SEG
$(CH$,5,8)&RPT$(SEG$(CH$,15,
2),2)):: NEXT CH
151 REM
```

Or, if you would prefer them shorter for single-line spacing -

```
150 FOR CH=48 TO 90 :: CALL
CHARPAT(CH,CH$):: CALL CHAR(
CH,"00"&RPT$(SEG$(CH$,3,2),2
)&SEG$(CH$,7,6)&RPT$(SEG$(CH
$,15,2),2)):: NEXT CH
151 REM
```

If you would like numerals the same size as lower case,

```
150 FOR CH=48 TO 57 :: CALL
CHARPAT(CH,CH$):: CALL CHAR(
CH,"0000"&SEG$(CH$,1,6)&SEG$
(CH$,9,4)&SEG$(CH$,15,2))::
NEXT CH
151 REM
```

You can even shrink the lower case to only 4 rows high, although some letters are not very legible -

```
150 FOR CH=97 TO 122 :: CALL
CHARPAT(CH,CH$):: CALL CHAR
(CH,SEG$(CH$,1,6)&SEG$(CH$,5
,4)&SEG$(CH$,11,6)):: NEXT C
H
151 REM
```

Or widen it both left and right -

```
160 FOR CH=48 TO 122 :: CALL
CHARPAT(CH,CH$):: FOR J=1 T
O 15 STEP 2
161 CH2$=CH2$&SEG$("014589CD
",POS("01234567",SEG$(CH$,J,
1),1),1)&SEG$("028A",POS("04
8C",SEG$(CH$,J+1,1),1),1)
162 NEXT J :: CALL CHAR(CH,C
H2$):: CH2$="" :: NEXT CH
163 REM
```

Or even a full 8 columns wide by just changing the "028A" in line 161 to "0129"

For darker characters, we can shade them into the 7th column -

```
160 FOR CH=48 TO 122 :: CALL
CHARPAT(CH,CH$):: FOR J=2 T
O 16 STEP 2 :: IF SEG$(CH$,J
-1,1)="1" THEN CH2$=CH2$&"18
" :: GOTO 163
161 IF CH=67 OR CH=71 OR CH=
99 OR CH=103 THEN 162 :: IF
SEG$(CH$,J-1,1)="4" AND SEG$
(CH$,J,1)="0" THEN CH2$=CH2$
&"60" :: GOTO 163
162 CH2$=CH2$&SEG$(CH$,J-1,1
)&SEG$("0367CBEF",POS("02468
ACE",SEG$(CH$,J,1),1),1)
163 NEXT J :: CALL CHAR(CH,C
H2$):: CH2$="" :: NEXT CH
```

Or shade them both left and right

```
160 FOR CH=48 TO 122 :: CALL
CHARPAT(CH,CH$):: FOR J=1 T
O 15 STEP 2 :: A$=SEG$(CH$,J
,1):: P=POS("0123456789ABCDE
F",A$,1)
161 A$=SEG$("0367CDEF89ABCDE
F",P,1):: B$=SEG$(CH$,J+1,1)
:: P=POS("02468ACE",B$,1)::
B$=SEG$("0367CBEF",P,1):: CH
2$=CH2$&A$&B$
162 NEXT J :: CALL CHAR(CH,C
H2$):: CH2$="" :: NEXT CH
163 CALL CHAR(74,"000C0C0C0C
0C4C38"):: CALL CHAR(106,"00
00000C0C0C4C38")
```

Or shaded into both of the rightmost columns -

```
160 FOR CH=48 TO 122 :: CALL
CHARPAT(CH,CH$):: FOR J=2 T
O 16 STEP 2 :: CH2$=CH2$&SEG
$(CH$,J-1,1)&SEG$("0377EBFF"
,POS("02468ACE",SEG$(CH$,J,1
),1),1):: NEXT J :: CALL CHA
R(CH,CH2$):: CH2$="" :: NEXT
CH
161 REM
162 REM
163 REM
```

---->more---->

Or into all 8 columns -

```
160 FOR CH=48 TO 122 :: CALL
CHARPAT(CH,CH$):: FOR J=1 T
O 15 STEP 2 :: P=POS("012345
6789ABCDEF",SEG$(CH$,J,1),1)
161 A$=SEG$("0367CDEF89ABCDE
F",P,1):: P=POS("02468ACE",S
EG$(CH$,J+1,1),1):: B$=SEG$(
"0367EBFF",P,1):: CH2$=CH2$&
A$&B$
162 NEXT J :: CALL CHAR(CH,C
H2$):: CH2$="" :: NEXT CH
163 REM
```

More neatly, shaded inward at right -

```
160 FOR CH=48 TO 122 :: CALL
CHARPAT(CH,CH$)
161 FOR J=1 TO 15 STEP 2 ::
CH2$=CH2$&SEG$(CH$,J,1)&SEG$
("0C8C",POS("048C",SEG$(CH$,
J+1,1),1),1):: NEXT J
162 CALL CHAR(CH,CH2$):: CH2
$="" :: NEXT CH
163 REM
```

Or inward at right, out- ward at left -

```
160 FOR CH=48 TO 122 :: CALL
CHARPAT(CH,CH$):: FOR J=1 T
O 15 STEP 2
161 CH2$=CH2$&SEG$("0367CBEF
",POS("01234567",SEG$(CH$,J,
1),1),1)&SEG$("0C8C",POS("04
8C",SEG$(CH$,J+1,1),1),1)::
NEXT J
162 CALL CHAR(CH,CH2$):: CH2
$="" :: NEXT CH
163 REM
```

Here's a weirdo -

```
160 FOR CH=48 TO 122 :: CALL
CHARPAT(CH,CH$):: FOR J=9 T
O 15 STEP 2
161 CH2$=CH2$&SEG$("014589CD
",POS("01234567",SEG$(CH$,J,
1),1),1)&SEG$("028A",POS("04
8C",SEG$(CH$,J+1,1),1),1)
162 NEXT J :: CALL CHAR(CH,S
EG$(CH$,1,8)&CH2$):: CH2$=""
:: NEXT CH
163 REM
```

Try changing that to FOR J =1 TO 7 and CALL CHAR(CH,CH2$&SEG$(CH$,9,8))

---------->more!---->

Something modernistic -

```
150 A$="00" :: FOR CH=48 TO
90 :: CALL CHARPAT(CH,CH$)::
CALL CHAR(CH,SEG$(CH$,1,4)&
A$&SEG$(CH$,7,6)&A$&SEG$(CH$
,15,2)):: NEXT CH
151 REM
```

Or perhaps even better -

```
150 A$="00" :: FOR CH=48 TO
90 :: CALL CHARPAT(CH,CH$)::
CH$=SEG$(CH$,3,10)&RPT$(SEG
$(CH$,13,2),2)&SEG$(CH$,15,2
)
151 CALL CHAR(CH,SEG$(CH$,1,
4)&A$&SEG$(CH$,7,2)&A$&SEG$(
CH$,11,2)&A$&SEG$(CH$,15,2))
:: NEXT CH
```

I call this one "Spooky".

```
150 FOR CH=48 TO 122 :: CALL
CHARPAT(CH,CH$):: CH$=SEG$(
CH$,3,14)&SEG$(CH$,1,2):: X$
=SEG$(CH$,1,1)&"0"
151 FOR J=3 TO 15 STEP 2 ::
X$=X$&SEG$(CH$,J,1)&SEG$(CH$
,J-1,1):: NEXT J :: CALL CHA
R(CH,X$):: X$="" :: NEXT CH
```

And "Spooky" backward -

```
150 FOR CH=48 TO 122 :: CALL
CHARPAT(CH,CH$):: FOR J=1 T
O 15 STEP 2 :: CH2$=CH2$&SEG
$(CH$,J,1)&SEG$(CH$,J+3,1)::
NEXT J :: CALL CHAR(CH,CH2$
):: CH2$="" :: NEXT CH
151 REM
```

Now, clear the memory with NEW, then -
MERGE DSK1.100
MERGE DSK1.120
Add a line 500 GOTO 500
And start MERGEing in your series of "150" routines and running them to see what you have created.
Then, save these next routines in MERGE format as 160A, 160B, etc.

All normal characters have the leftmost column of pixels and the two right- most columns blank, for spacing between letters. We can widen the character into the left column -

```
160 FOR CH=48 TO 122 :: CALL
CHARPAT(CH,CH$):: FOR J=1 T
O 15 STEP 2
161 CH2$=CH2$&SEG$("014589CD
",POS("01234567",SEG$(CH$,J,
1),1),1)&SEG$(CH$,J+1,1):: N
EXT J :: CALL CHAR(CH,CH2$):
: CH2$="" :: NEXT CH
162 REM
163 REM
```

---->more--->>

And one more -

```
160 FOR CH=48 TO 122 :: CALL
CHARPAT(CH,CH$):: FOR J=1 T
O 7 STEP 2
161 A$=SEG$("02468ACE",POS("
01234567",SEG$(CH$,J,1),1),1
):: B$=SEG$("0808",POS("048C
",SEG$(CH$,J+1,1),1),1):: CH
2$=CH2$&A$&B$ :: NEXT J
162 CALL CHAR(CH,CH2$&SEG$(C
H$,9,8)):: CH2$="" :: NEXT C
H
163 REM
```

Now, clear the memory, MERGE in 100 and 120, put in a holding line 500 GOTO 500 and start MERGEing in all of the different combinations of the 150 and 160 lines and see how many different character sets you can make!

Memory full,
Jim Peterson

======================================

PRODUCT UPDATE:

The following disks -commercial and disk library - have been updated as follows:

TI WRITER VN 4 BY ART GREEN: VERSION 4.1 amendments: Editor did not perform tabs. Formatter did not respond to up arrow processing user input. Formatter did not process end of file properly for .IF files. Formatter did not process user input of page/letter numbers properly. Note: The installation programs have also been modified for Version 4.1

VERSION 4.2 amendments:
Editor in move lines when moving a single line to itself. For example: M 2,2,1

VERSION 4.3 amendments:
Editor did not handle the autoindent correctly when loading the "tabs" from an existing file.

VERSION 4.4 amendments: Formatter did not handle .DR command correctly. Caused looping when underscoring was attempted. Translation of >00 was in error. Editor now shows Version 4.3 Formatter now shows Version 4.4 NOTE FOR RAM DISK USERS Both the Editor and Formatter exit in the normal way as defined by TI. That is, then return to the caller as if this caller were a GPL program in the cartridge. A well known RAM DISK menu system does not seem to like programs that do this. If you are using such a RAM DISK system you may want to patch the Editor and Formatter to do a software reset on exit by a BLWP to zero. COPY THE DISTRIBUTION DISK FIRST!!! COPY THE DISTRIBUTION DISK FIRST!!! COPY THE DISTRIBUTION DISK FIRST!!! EDITOR PATCH Sector >3F Offset >A8, Is now: >02E0 A032 Patch to: >0420 0000 FORMATTER PATCH Sector >6C Offset >76, Is now: >02E0 BE5C Patch to: >0420 0000

VERSION 4.5 amendments:
Formatter did not handle mailing list insert #1 correctly. It was initialized to garbage. Editor now shows Version 4.3 Formatter now shows Version 4.5 COPY THE DISTRIBUTION DISK FIRST!!! COPY THE DISTRIBUTION DISK FIRST!!! COPY THE DISTRIBUTION DISK FIRST!!! FORMATTER PATCH for V4.5 Using the 4.3 or 4.4 disk. Sector >66 Offset >CF, Is now: >2E34 (Version #) Patch to: >2E35 (".5") Sector >7E Offset >F8, Is now: >018E Patch to: >0190 Sector >7E Offset >FC, Is now: >C48E Patch to: >C48C

==========

(more---->) p45

This program is in TI BASIC but will also run in Extended Basic. The concept is not original- the late Mr Matthews of TX Software had a similar program printed in an early magazine; Patrick Strassen had his similar NIGHT FLIGHT published by Stainless Software, and there have been many others. This one differs in two ways- it is written by a current group member; and it has been written recently- WITHOUT reference to previous attempts at the game. To fire the bomb just press SPACE. The plane will land before all the buildings are destroyed but after MOST of them have- when there is enough room to land! Inspect the coding, practice your typing skills in typing it in, and enjoy a simple old but enduring game.... all in TI Basic....

```
100 REM HELIBOMBER V2.0
      (C)1989 M.WILLS
110 REM TI BASIC
120 REM
130 CALL CLEAR
140 PRINT "INITIALIZING...";
150 HI=1000
160 HI$="Mark"
170 GOSUB 2790
180 GOTO 2510
190 OPTION BASE 1
200 RANDOMIZE
210 DIM GRAPHICS(12),PYX(2)
220 PYX(1)=1
230 PYX(2)=1
240 NB=8
250 SC=0
260 LI=3
270 SH=1
280 V=30
290 RESTORE 340
300 FOR I=1 TO 12
310 READ D
320 GRAPHICS(I)=D
330 NEXT I
340 DATA 34,40,58,95,35,41,5
9,96,33,42,60,123
350 CALL CLEAR
360 SCOL=2
370 CALL SCREEN(SCOL)
380 CALL COLOR(1,4,SCOL)
390 CALL COLOR(2,5,SCOL)
400 CALL COLOR(4,14,SCOL)
410 CALL COLOR(8,12,SCOL)
420 CALL COLOR(9,12,SCOL)
430 CALL COLOR(12,12,SCOL)
440 RESTORE 490
450 READ C,D$
460 IF C=0 THEN 490
470 CALL CHAR(C,D$)
480 GOTO 450
490 DATA 34,FFC3C3C3C7FFFFFF
500 DATA 40,7E66667E6666667E
510 DATA 58,FFFFCDCDCDCDFFCD
520 DATA 95,7F63637F7F7F6363
530 DATA 35,181818183C5A9999
540 DATA 41,181818183C7EFFFF
550 DATA 59,18183C3C7E7EFFFF
560 DATA 96,0808081C3E7F7F7F
570 DATA 38,7E18183C3C3C3C18
580 DATA 39,FFBDDBE7E7DBBDFF
590 DATA 36,0F0081C3C7FF0300
600 DATA 37,FFE01008FCFCF820
610 DATA 33,00000189AFAFEFFF
620 DATA 42,0000082B387C7E7E
630 DATA 60,000000000829BFFF
640 DATA 123,202028687A7B7F7
F
650 DATA 0,NUFFINK
660 CALL HCHAR(24,1,39,32)
670 X=(16-(NB/2))+1
680 FOR I=1 TO NB
690 A=INT(RND*4)+1
700 H=INT(RND*12)+2
710 TS=TS+(H+1)*10
720 CALL VCHAR(24-H,I-1+(X),
GRAPHICS(A),H)
730 CALL VCHAR(23-H,I-1+(X),
GRAPHICS(4+A))
740 NEXT I

750 REM
760 REM
770 X=PYX(2)
780 Y=PYX(1)
790 F=0
800 CALL HCHAR(Y,X,36)
810 CALL HCHAR(Y,X+1,37)
820 CALL SOUND(-1000,-6,15)
830 IF X>=31 THEN 870
840 CALL GCHAR(Y,X+2,BU)
850 IF BU<>32 THEN 1330
860 IF F=1 THEN 1020
870 CALL KEY(0,K,S)
880 IF K=32 THEN 960
890 CALL HCHAR(Y,X,32,2)
900 X=X+1
910 IF X>31 THEN 920 ELSE 80
0
920 X=1
930 Y=Y+1
940 IF Y=23 THEN 1680 ELSE 8
00
950 END
960 BX=X
970 BY=Y+1
980 F=1
990 V=15
1000 K=0
1010 BS=INT(RND*7)+4
1020 CALL VCHAR(BY,BX,38)
1030 CALL GCHAR(BY+1,BX,C)
1040 IF C<>32 THEN 1090
1050 CALL VCHAR(BY,BX,32)
1060 IF BY=23 THEN 1300
1070 BY=BY+1
1080 GOTO 890
1090 IF C=39 THEN 1300
1100 SC=SC+10
1110 BS=BS-1
1120 IF SC=TS THEN 1420
1130 IF BS=0 THEN 1160
1140 IF BY=23 THEN 1300
1150 GOTO 1050
1160 CALL GCHAR(BY+1,BX,CH)
1170 TS=TS+10
1180 IF BY=23 THEN 1260
1190 PO=1
1200 IF GRAPHICS(PO)=CH THEN
 1240
1210 PO=PO+1
1220 IF PO>4 THEN 1240
1230 GOTO 1200
1240 PO=PO+8
1250 CALL HCHAR(BY,BX,GRAPHI
CS(PO))
1260 F=0
1270 CALL SOUND(-100,-5,0)
1280 V=30
1290 GOTO 890
1300 CALL VCHAR(BY,BX,32)
```

```
1310 F=0
1320 GOTO 890
1330 FOR R=1 TO 4
1340 FOR I=2 TO 16
1350 CALL SCREEN(I)
1360 NEXT I
1370 NEXT R
1380 CALL HCHAR(Y,X,32,2)
1390 GOTO 1970
1400 IF LI=0 THEN 1990
1410 GOTO 1990
1420 CALL HCHAR(BY,BX,32)
1430 CALL GCHAR(23,X,CH)
1440 CALL GCHAR(23,X+1,CH1)
1450 IF CH1=32 THEN 1540
1460 IF CH=32 THEN 1540
1470 FOR Z=Y+1 TO 23
1480 CALL HCHAR(Z,X,38,2)
1490 FOR D=1 TO 20
1500 NEXT D
1510 CALL HCHAR(Z,X,32,2)
1520 NEXT Z
1530 CALL HCHAR(23,X,32,2)
1540 FOR Z=Y TO 23
1550 CALL HCHAR(Z,X,36)
1560 CALL HCHAR(Z,X+1,37)
1570 FOR D=1 TO 30
1580 NEXT D
1590 CALL HCHAR(Z,X,32,2)
1600 NEXT Z
1610 CALL HCHAR(23,X,36)
1620 CALL HCHAR(23,X+1,37)
1630 FOR I=X+2 TO 32
1640 CALL HCHAR(23,I,45)
1650 CALL VCHAR(1,I,32,22)
1660 CALL HCHAR(23,I,32)
1670 NEXT I
1680 FOR Z=X TO 31
1690 CALL HCHAR(23,Z,36)
1700 CALL HCHAR(23,Z+1,37)
1710 FOR D=1 TO 30
1720 NEXT D
1730 CALL HCHAR(23,Z,32)
1740 NEXT Z
1750 CALL HCHAR(23,32,36)
1760 FOR D=1 TO 30
1770 NEXT D
1780 CALL HCHAR(23,32,32)
1790 FOR D=1 TO 16
1800 NEXT D
1810 FOR I=3 TO 8
1820 CALL COLOR(I,16,SCOL)
1830 NEXT I
1840 PRINT : : :" WELL DONE
 SHEET";SH;"CRUSHED": :
1850 PRINT "           SCORE
 -";SC
```

```
1860 PRINT :"            LIVE
S -";LI
1870 IF SH+1=4 THEN 3120
1880 PRINT :"       PREPARE F
OR SHEET";SH+1
1890 FOR D=1 TO 400
1900 NEXT D
1910 PRINT : : : : : : : : :
 : : : : : : : : : : : : : : :
 : : : : :
1920 NB=NB+5
1930 SH=SH+1
1940 PYX(1)=(SH*2)+1
1950 PYX(2)=PYX(2)
1960 GOTO 350

1970 REM
1980 CALL SCREEN(SCOL)
1990 YA=Y
2000 YB=Y
2010 XA=X
2020 XB=X+1
2030 A=1
2040 B=1
2050 CALL HCHAR(YA,XA,36)
2060 CALL HCHAR(YB,XB,37)
2070 IF A=0 THEN 2090
2080 CALL HCHAR(YA,XA,32)
2090 IF B=0 THEN 2110
2100 CALL HCHAR(YB,XB,32)
2110 CALL GCHAR(YA+1,XA,C)
2120 CALL GCHAR(YB+1,XB,D)
2130 IF C=32 THEN 2140 ELSE
2160
2140 YA=YA+1
2150 GOTO 2170
2160 A=0
2170 IF D=32 THEN 2200
2180 B=0
2190 GOTO 2210
2200 YB=YB+1
2210 IF A+B=0 THEN 2220 ELSE
 2050
2220 CALL HCHAR(YA,XA,36)
2230 FOR DE=1 TO 50
2240 CALL HCHAR(YB,XB,37)
2250 NEXT DE
2260 CALL HCHAR(YA,XA,32)
2270 CALL HCHAR(YB,XB,32)
2280 LI=LI-1
2290 PYX(1)=Y-2
2300 IF LI=0 THEN 2320
2310 GOTO 760
2320 FOR I=1 TO 8
2330 CALL COLOR(I,16,SCOL)
2340 NEXT I
```

```
2350 PRINT : : : : : : : : :
 : : : : : : : : : : : : : : :
 :
2360 PRINT "      ++ GAME OV
ER ++": : : :
2370 PRINT "      YOU SCORED
 ";SC: : :
2380 IF SC>HI THEN 2390 ELSE
 2420
2390 HI=SC
2400 PRINT "YOU HAVE THE HIG
H SCORE": : :
2410 INPUT "TYPE IN YOUR NAM
E + PRESS  ENTER ":HI$
2420 PRINT "PRESS S TO START
 NEW GAME         I FOR INST
RUCTIONS          E TO EXIT
PROGRAM"
2430 CALL KEY(3,K,S)
2440 IF K=83 THEN 220
2450 IF K=73 THEN 2510
2460 IF K=69 THEN 2480
2470 GOTO 2430
2480 CALL CLEAR
2490 PRINT """NEW"" PROGRAM
TO ERASE."
2500 END
2510 CALL CLEAR
2520 CALL SCREEN(5)
2530 FOR I=1 TO 14
2540 CALL COLOR(I,2,16)
2550 NEXT I
2560 CALL COLOR(13,5,2)
2570 CALL CHAR(128,"FFFFFFFF
FFFFFFFF")
2580 CALL CHAR(35,"3C4299A1A
199423C")
2590 CALL CHAR(39,"000808100
0000000")
2600 CALL CHAR(95,"00FF00000
0000000")
2610 CALL VCHAR(1,1,128,48)
2620 CALL VCHAR(1,31,128,48)
2630 PRINT " HELIBOMBER ";CH
R$(35);" 1989 M.WILLS"
2640 PRINT " ________________
__________": :
2650 PRINT "Bad news. You ar
e very low  on fuel and ther
e is'nt a  runway for miles
 around."
2660 PRINT "Your only option
 is to bomb the deserted cit
y below you and hope you can
 blow enough"
2670 PRINT "of the buildings
 below you  down in order to
 land."
```

```
2680 PRINT :"You have an unl
imited supplyof bombs but be
ware, you   can only drop o
ne at a time."
2690 PRINT :" Press any key
to continue": : : : :
2700 CALL KEY(3,K,S)
2710 IF S=0 THEN 2700
2720 PRINT : : : : : : : : :
 :"You have three lives and
   must complete three citie
s  in order to win the game.
"
2730 PRINT :"todays high sco
re is";HI:"by ";HI$;".";: :
2740 PRINT :"  press any key
 to play": : : : :
2750 CALL KEY(0,K,S)
2760 IF S=0 THEN 2750
2770 CALL CLEAR
2780 GOTO 190
2790 RESTORE 2800
2800 DATA 00000E020E120D0010
101E111111E0000000F1010100F
0001010F111110F00
2810 DATA 00000E111E100E0006
09081C0808080000000E11110F01
0E10101E111111100
2820 DATA 04000C040404040E02
0002020202020A0410101214181412
000C04040404040600
2830 DATA 00001A151515110000
0016090909090000000E1111110E
0000001E11111E1010
2840 DATA 00000F11110F010100
00161910101000000000F100E011E
0008081E0808090600
2850 DATA 0000121212120D0000
0011110A0A040000000111115150A
000000110A040A1100
2860 DATA 00001111110F011E00
001F0204081F0000000000000000
000000000000000000
2870 FOR C=97 TO 121 STEP 4
2880 READ P$
2890 CALL CHAR(C,SEG$(P$,1,1
6))
2900 CALL CHAR(C+1,SEG$(P$,1
7,16))
2910 CALL CHAR(C+2,SEG$(P$,3
3,16))
2920 CALL CHAR(C+3,SEG$(P$,4
9,16))
2930 NEXT C
2940 RETURN

2950 REM ******************
         *                 *
```

```
2960 REM * THIS IS A FREE *
         * WARE PROGRAM & *
         * MAY BE FREELY  *
         * COPIED AMONGST *
2970 REM * OTHER TI-99/4A *
         * USERS, AS LONG *
         * AS THE AUTHORS *
         *  NAME IS NOT   *
2980 REM * CHANGED IN ANY *
         *  WAY, AS THIS  *
         *  CONSITIUTES
 *          *  INFRINGEMENT
 *

2990 REM *  OF COPYRIGHT  *
         *                *
         ******************
3000 REM
3050 REM ******************
         * MY ADDRESS IS: *
         * MARK WILLS,
 *          * 37 ABBOTS ROA
D,*

3060 REM * MONKMOOR,      *
3070 REM * SHREWSBURY,    *
         * SHROPSHIRE,    *
         * SY2 5PZ.
 *          * TEL:0743 6417
7 *

3080 REM ******************
3090 REM
3100 REM * ENJOY THE GAME!*
3110 REM
3120 FOR I=3 TO 8
3130 CALL COLOR(I,16,SCOL)
3140 NEXT I
3150 CALL CLEAR
3160 CALL CHAR(125,"007E7E7E
7E7E7E00")
3170 CALL CHAR(128,"007E7E7E
7E7E7E00")
3180 CALL COLOR(13,9,SCOL)
3190 PRINT "$Z$Z$Z$Z$Z$Z$Z$Z
$Z$Z$Z$Z$Z$Z": : : : : : : :
 : : : : : :
3200 PRINT "$Z$Z$Z$Z$Z$Z$Z$Z
$Z$Z$Z$Z$Z$Z"
3210 PRINT :"         SCORE -
";SC: :"    ANY KEY WHEN RE
ADY": :
3220 PRINT "$Z$Z$Z$Z$Z$Z$Z$Z
$Z$Z$Z$Z$Z$Z"
3230 RESTORE 3470
3240 READ Y,X,R
3250 IF Y=99 THEN 3310
```

```
3260 CALL VCHAR(Y+3,X+4,125,
R)
3270 READ Y,X,R
3280 IF Y=99 THEN 3310
3290 CALL VCHAR(Y+3,X+4,128,
R)
3300 GOTO 3240

3310 REM
3320 CALL COLOR(13,9,SCOL)
3330 FOR D=1 TO 10
3340 NEXT D
3350 CALL COLOR(12,12,SCOL)
3360 CALL COLOR(1,INT(RND*15
)+1,SCOL)
3370 CALL KEY(0,K,S)
3380 IF S<>0 THEN 2380
3390 CALL COLOR(12,9,SCOL)
3400 FOR D=1 TO 10
3410 NEXT D
3420 CALL COLOR(13,12,SCOL)
3430 CALL COLOR(1,INT(RND*15
)+1,SCOL)
3440 FOR D=1 TO 6
3450 NEXT D
3460 GOTO 3320
3470 DATA 2,2,5,5,3,1,4,4,1,
5,5,1,2,6,5
3480 DATA 2,8,5,2,9,1,2,10,1
,2,11,1,2,12,1,4,9,1,4,10,1,
6,9,1,6,10,1,6,11,1,6,12,1
3490 DATA 2,14,5,6,15,1,6,16
,1,6,17,1,6,18,1
3500 DATA 2,20,5,6,21,1,6,22
,1,6,23,1,6,24,1
3510 DATA 8,2,5,8,3,1,8,4,1,
8,5,1,9,6,1,10,6,1,11,6,1,12
,3,1,12,4,1,12,5,1
3520 DATA 9,8,3,8,9,1,8,10,1
,8,11,1,9,12,3,12,11,1,12,10
,1,12,9,1
3530 DATA 8,14,5,9,15,1,10,1
6,1,11,17,1,8,18,5
3540 DATA 8,20,5,8,21,1,8,22
,1,8,23,1,8,24,1,10,21,1,10,
22,1,12,21,1,12,22,1,12,23,1
,12,24,1
3550 DATA 99,0,0
~~~~~~~~~~~~~~~~~~~~~~~~~~~~
~~~~~~~~~~~~~~~~
```