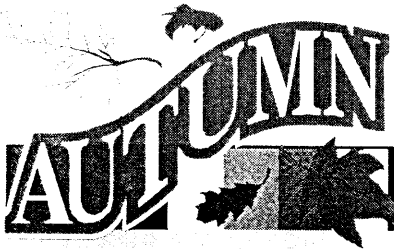


TI*MES

Issue 58
Autumn 1997

The Magazine of the
Texas Instruments TI-99/4A User Group UK



TEXAS INSTRUMENTS
HOME COMPUTER

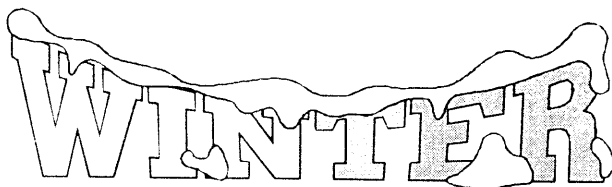
READY-PRESS ANY KEY TO BEGIN

©1981 TEXAS INSTRUMENTS



NEXT ISSUE

TI*MES 59



1997

Report on TI-TREF Fair in Holland

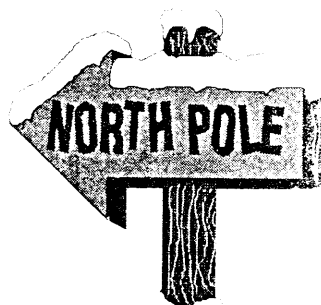
Exciting news about the next European TI Fair

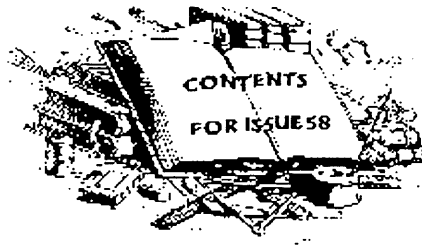
Listings

More updates from the States

BBC to TEXAS music conversion table

And much much More...





Editorial . . . Ian Pare	4
Inside Parsec	4
From the Chairmans Chair . . . Trevor Stevens	7
Retro Reviews from Home Computing Weekly .	18
Linear Regression	19
Letters to the Editor(s)	34
Programming the TI Graphics Listing	36
Rambles by Stephen Shaw	39
PC99 Update & USA News	42
by Richard Speed	

Disclaimer

The views expressed in this magazine are those of the individual authors and not necessarily those of the Editor or the Group.

Editorial E-Mail : TI99@infoserv.demon.co.uk

Editorial

Well, here's the autumn issue (58) of TI*MES and were well and truly in the clutches of Winter, so my apologies to all for getting this issue out late (once again). We are working on the Christmas Issue and promise to have it to you for Christmas (1997!). We are hoping to give Ross the masters before the end of November.

I'd like to say a big thank-you on behalf of Richard & myself, to all of the people who sent in articles for this issue. We're sorry that we haven't been able to publish them all, but rest assured that the one's we've missed will be in the Winter Issue of TI*MES.

In this Issue we have a great graphics listing, old software reviews, an interesting article from Francesco on LINEAR REGRESSION with listing, more PC99 news and updates from Richard Speed, and of course "Rambles" from Stephen Shaw, "From the Chairmans Chair".... and lots more !

Ian Pare (Co-Editor)

Inside Parsec



(by Ian Pare)

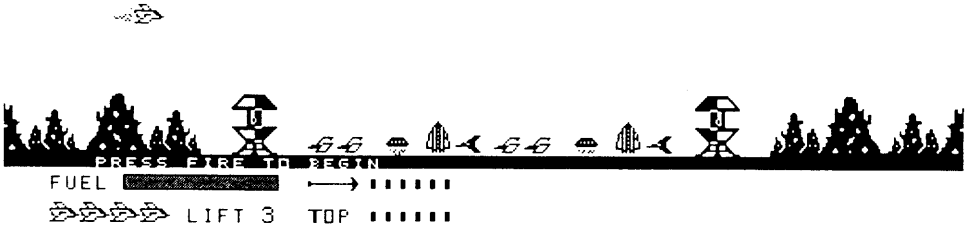
This is the response to an e-mail I sent to Paul Urbanus, co-developer of Parsec. All of the italics are my questions to him. I was delighted to get a reply !

Ian,

Sorry about the long time taken for this reply.

*>It's nice to finally track down someone that wrote such a legend of a
>game for the TI.*

Thanks. Jim Dramis and I had a blast writing this game. We were given complete freedom to come up with whatever we wanted. I was a cooperative education student at the time and was playing around with the Graphics II mode of the



new VDP chip. At that time, no one in the software development group was doing any work to utilize the extended features of this chip, so I decided to play with it myself. That was the most fun job I've ever had. I still play the game every once in a while, and it's still a blast

- >I would like to know, if there has ever been any of the source code
- >released from the original TI Game developments, and if you may have any
- >documents on how such developments may have been undertaken.

I know that TI never did release any of the source code into the public domain, although I understand that some code has 'leaked' out. I don't have any source code for PARSEC, nor do I know of anyone who has the source. I have been out of the TI 99/A community since 1989.

- >The TI-99/4A still has quite a strong membership base in the UK, and we
- >are constantly looking for new projects. I program PC based software for
- >a living and am interested in developing new TI software just for
- >interest.

- >
- >Any pointers, information, code or contacts you may have would be
- >greatly appreciated, that is, if you still keep in touch with the TI99
- >world.

- >
- >The problem in the UK is that we have been starved from almost
- >everything, and its difficult to know where to try next.

>



Inside Parsec

I wish I could give you some contacts, Ian. Like you, I've moved on to the PC, and am mainly doing hardware desing using FPGAs and VHDL (yeah, software again).

Good luck with your TI 99/4A endeavors. If you have any specific questions about programming the TI machine, give me a call and I'll try to answer them for you.

Best Wishes,

Paul Urbanus
urb@urbonix.com

* Paul Urbanus urb@urbonix.com *
* *
* Never wrestle with a hog - you get dirty and the hog likes it. *



Wanted TI-99/4A Surplus

If you have any equipment you no longer need or is faulty, Please let our hardware Guru Know. Fair Prices paid. Also Repairs at no charge for group members.

Please do not throw anything away !!

Ross Bennett, 20 Oak Avenue, Romiley, Stockport, SK6 4DN.
TEL :0161-430-7298 Evenings/Weekends,
FAX :0161-483-4516 (24-Hour)



FROM THE CHAIRMAN'S CHAIR

By T.STEVENS

Well things are still going on here, in fact I had a real BIG problem. The Hard drive on the BBS packed up leaving me in a none recovery position. However I have now managed to put back most of the files which were on the BBS thanks to a 250 meg Syquest drive which is in the Group Machine which is on loan from my good self. This saved my bacon as I had done a back up about 4 weeks prior to the drive packing up its bags. However things are now back to normal and I have set my self a polickey of doing a complete back up of the BBS ever two weeks or so. Due to lack of time I have been unable to put back the Pc99 area. However I hope to do this in the near future. New areas will put on very soon so watch this space. Before the hard drive failed I had decided to replace the devil, (prehaps it knew!). I had already sorted out a 2gig hard drive and fortunately had it in the "bottom drawer ready to fit. So I was able to sort the job out within three days. Also at the time the SCSI card for some reason blew a fuse. So I am wondering if I got a power surge or something, who knows.

For all you out there who love to go onto BBS lines and the internet there will soon be a thing (Code name LIGHTENING) yes it is spelt right! Which will send the poor old telephone companies in a bit of a spin with regard to landline services. This is a service which is being marketed by NORWEB at present, but will be with all the large electricity companies. The service is like a telephone line that comes off your mains supply. Yep that is what I said! It is said to be very cheap as all the wires are in place. How? Well the system uses the national grid and other networks of power supply to deliver your call. The signal will be sent first to substations which will have a route code exchange. This will open up your box which

is fitted where your mains supply enters your house. The system strips all the power line noise off and leaves you with a high quality high speed communications network. This is expected to be faster, said about one and a half times faster than ISDN is at present. However this still remains to be seen. The good side to all this is that telephone calls by normal means will drop.

I shall be going to the TI-FEST in October and will be going with Richard TWYNING. So I will report on what happened next mag.

The one thing that does strike me is that it is the old stagers still writing for the mag. How about *you* picking up a pen a paper or using your machine, and do an article for us, it is nice to hear from anyone out there including none TI members. To give you some ideas. What does your machine have on it that is special. Do you program? Send us a listing. Have you any interesting news? Have you any really old and nearly unknown articles from the past? Have you any problems you want to be answered. Remember even though we are a TI99/4a group we have enough experts out here to solve problems on most computer platforms. I myself know about the following. TI99/4a obviously. Amiga and fully system setups and WB2+ problems. Windows 95 and Windows 98 (Yes I did say that!) I have been working on it with a friend who is (SPIT!!!!) a MICROSOFT programmer and Beta tester. So we can sort out even the smallest or largest problem. If you are really I mean really stuck give me a ring I should be able to talk you out of your problem with you sat at your machine.

Enough of the other machine bit, I want to get back to the old TI. Funny really but I still love my TI and I still think, even with all these lovely graphics etc, that the TI is one of the most user friendly machines I have ever used. It does what I want it too and does it properly. I think TI if they had kept with it would have wiped the floor with the PC. If you look at the new graphic chips etc they could have built an awesome machine. If but maybe

you say, well thats life.

Now to programming. I am still wondering if there is anyone out there who likes this article. Not one person has said or asked a question. LET ME KNOW else I will retire to my corner and have a cry!!!! JOKE.

Well you are going to get some now whether you like it or not!!!!!!!!!!!!

Did you sort out last quarters programs? If not then read them over very carefully again before you start to read this article. The last program we did was to send a @ sign across the screen and delaying it so that it was usable. There are many ways to write a program so try and write one that is say smaller. However the best way to write for me is in the clearest way possible, then I understand it.

General Addressing Mode

When an instruction works on two operands in the operand field, we call this the **source operand** the one we are going to operate on or play around with. The operand where the end result of the operation is placed is called the **destination operand**. Simple **to** and **from** is how I remember it.

le
7D00 A R4,R3

R4 being the source(from) and R3 being the destination (to).

Ther five ways to work with values in a register or memory location. Called General Addressing Modes, they are:

WORKSPACE REGISTER ADDRESSING.

This is what you have been doing in the example programs in the back articles. That is working with the values of 0 to 15 in the registers. Ie A R8,R9

WORKSPACE REGISTER INDIRECT ADDRESSING.

This is where the register contains the memory location where the value to be used is found. Indirect addressing is indicated by specifying a asterisk before the register, like thus: A *R8,*R9 (This adds the value found at R8 to the value found in R9 and then puts the answer in R9.)

If an astrisk is in front of the said register then it represents the phrase *the contents found the memory location specified by the value in this register*. In the previous example if R8 contained >7D00 and R4 contained >7E00 the instruction adds the **word** stored in the memory of >7D00 on to the **word** stored in 7E00 then placing the answer in 7E00. Simple is`nt it!

You do not have to put the astrisk on both operands if you only want one of the registers to hold the address of a memory location.

ie: S *R2,R5

In this example the value found in R5 was used, **not** the contents of the memory location loaded into that Memory location.

A good example of this is when we go back to easy bug with the command

B *R11

This means Branch to the memory location held in R11. Do you see the difference? If not Read this over again NOW!

WORKSPACE REGISTER INDIRECT AUTO-INCREMENT ADDRESSING

This is Symbolised by following the register with a plus (+) sign. This mode increases the memory address stored in the register by one byte or one word, according to the instruction used. Ie, assume R3 is loaded with memory location 7D00, the following instruction

AI *R3+,100

adds 100 to the value found in the memory addressed in R3 by two, leaving >7D02 stored in R3. Try it and see, on your machine. The + does this to the value. The increment was of one word, two bytes, because AI is an instruction operating with words. If however you can imagine that R5 is loaded with >7EF8 and R7 with 7F50 the instruction AB *R5,*R7+ adds the left byte of the word in memory location >7EF8, stored in R5 to the left byte of the word in memory location >7F50, stored in R7. The value in R7 is then incremented by one byte as AB is a byte instruction. If this instruction was executed for a second time then the left byte of the word stored in the memory location found in R5 (>7EF8) would be added to the byte found in 7F51 (stored in R7), because the value in R7 was already incremented by one byte when the instruction was executed the first time round.

This addressing mode can be very useful to us when working with data tables, as you will see later in the series.

SYMBOLIC MEMORY ADDRESSING.

This is when we work directly with a memory location or a label at a memory address. The symbolic memory address is preceded by the @ symbol.

IE CB @ NM,@>7D00
 A @ >7D00,@7DFF

Again the operator (CB) and (A) determine what the @ does.

In the CB example the left byte of the word found at the memory location labeled NM is compared to the left byte of the word found at >7D00.

In the example A(dd) instruction then the word in >7D00 is added to >7DFF putting the answer in >7D00.

INDEX MEMORY ADDRESSING

an indexed memory address is preceded by the @ sign as well but followed by the register like this S @5(R5),R9. Any register except R0 may be used. What happens in the above example is this, S subtracts the word stored in the memory address found by adding 5 to the value of R5 from the value of R9. The difference is placed in R9.

Another example: A R7,@NM-3(R3)

Wow you say what does that do, well A adds the word found in R7 to the Word stored in the location found by subtracting 3 from the value NM , and then adding this to the value in R3. The answer is then placed in the same computed address.

We do not use index memory addressing frequently (Thank GOD some say). However it is used. We will cover more of this subject in more detail later.

We have now covered most of the basic stuff with a few odd exception which will be made apparent when we get there. It is time now to look at Planning our programs. Unlike Basic programs where you can sit down and sort thing out on the fly, you MUST plan you assembly programs else you will not know where you bug is if you come across one. REMEMBER there is no TRACE command!!!!!!

Assembly programs are not difficult to write as long as you do plan them. The line by line assembler due to its lack of editing features make you problems and you can not sit down and just tap away. If however you have E/A then you use the proper editor. Even then planning is very important. So sit down at your word processor or use a paper and pencil.

To give you an idea of how I work out my programs I sit down and jot down what I want my program to do.

So if say I wanted to print out a start message, I would put down first of all the following.

Clear the screen	Routine
Change screen Color	Routine
Print Title	Routine
and so on.....	

I then take each routine and type it in to see if it works, as I want it to.

If it works then I jot it down and put it into a WHEEL file. (Do Not reinvent the Wheel, use what you have already tested and know works.)

Having sorted out that all the routines work on their own, put them together to form the complete program. DO NOT sit down and write the whole program onto paper and then type the lot in as you will have hell on trying to sort out any problem. You have to do a lot of memory research when you write your program from looking at memory tables and sorting out strange errors and effects from the computer by your routines. Never get discouraged . Put it down leave it for a bit and then when you get back you might be able to resolve the problem. I have even gone to bed and then to work and while driving along had the "Light Bulb" solution. Gone home at the days end and BINGO!

In the Minimemory, as it says you do not have much memory to play with. Those with E/A of course have more. Even so if you can save bytes DO IT. Ther are tricks we can use like addressing direct to a memory location instead of using a label for a routine. The classics of this is when you use VMBW and the like. All these have memory locations so why waste memory when you can get there direct. Repeat Routine are also good ones to look at. Like Clear the screen. Why not set this aside in your program and use this one routine as a SUBROUTINE by using our dear old friend BL (Branch and Link) It is used like B (Branch) to transfer control to any memory location desired. However the instruction should not be confused. While B sends control to another memory address(as Basic GOTO), BL lets you return from the routine to the instruction immediately after the BL, just like a GOSUB.

Use them like this :-

```
7D00      BL @NM          BL,s to the label NM
7D04      BL @>7BC0      BL,S to the memory location
```

When using BL the memory location with the instruction immediately following that location of the BL instruction is put into the Register R11. To return like a sub exit in your routine all you do is B *R11. This way the program continues and flies to your next routine in the sequence. This also dummies for the RT instruction in the E/A (ReTurn), so B*R11 = RT

So lets do a subroutine (which you can now put in **your** WHEEL file) to clear your screen.

```
BL @CR
CR CLR R0
LI R1,>2000
LP BLWP@>6024
INC R0
CI R0,768
JLT LP
B *R11
```

USING NOP

This convenient instruction helps you prepare your programs for later editing. NOP stands for (No Operation). By doing this you can NOP some memory locations if you wish to add something into those areas later say to correct errors in your program.

Consider this example, where the memory location >8374 should have been cleared for the program to work correctly. Luckily some memory locations were NOP

```
7D00    LI R5,4
7D04    NOP
7D06    NOP
7D08    NOP
7D0A    BLWP@>6020
```

Now thanks to the free memory locations, you can correct the error so that the program lines read....

```
7D00    AROG>7D04
7D04    CLR@>8374
7D06    END
```

The memory location 7D08 is still free to add something else if you wish. The NOP is ignored by the assembler, so use NOP when you feel you may have to add or correct anything. It is also wise to use the NOP when you are jumping to a label not yet defined with one of your Jump instructions, because it may end up that when the label is finally defined, that it is beyond the 256 byte limit of the instruction. If there are no NOP instructions after or before the jump, you will not be able to change it to a B (Branch) instruction, because it uses two bytes more than the Jump and you will not fit it in, and a re-type is on the cards.

The next example jumps to the label NT, which is beyond the range defined. Causing that dreaded *R-ERROR*

```
7D00      LI R2,5
7D04      CI R2,5
7D08      JEQ NT
7D0A      NOP
7D0C      NOP
7D0E RN   (PROGRAM CONTINUES...)

7F60 NT   CLR R2
7D08 *R-ERROR*
```

The error was caused by the JEQ instruction trying to transfer control to NT which was too far away. The free locations now come into play to correct as so.

```
7F62      AORG >7D08
7D08      JNE RN
7D0A      B@NT
7D0E      END
```

The error is now corrected and you do not have to type you whole program in from the start. E/A owners are lucky. They just redo the source code in the editor and away they go.

COPYING REGISTERS : MOV AND MOVB

Many times you will need to copy the value of one register to another, or memory location. To do this you use MOVE to move a word and MOVE Byte to move a byte. MOV instruction makes a copy of the word value in the source operand, placing it in the destination operand.

le

7D00 MOV R3,R4 Moves word found in R3 into R4
 leaving R3 unchanged.

7D02 MOV R4,@>7E00 Moves word into >7E00

7D06 MOV @>7D24,R5 Moves word found at location 7D24
 into R5

The MOVB work in much the same way instead it copies the left (most significant byte) of the words.

le

7D00 MOVB R7,R2 Copies left Byte of R7 to left Byte
 of R2 the right bytes remain the
 same.

And so on.....

Well next time we will go on a bit further and look at how to save more memory by using fewer labels as we briefly discussed. And we will write a small program to show you how its done. The we will execute the program and also add a program name to it. So see you next time. Hope you understood everthing here discussed if you are unsure let me know and we will bash it out together.

Memory VERY full Fctn Quit BEEP!!!!

Turn off the telly and system.....

GROUP OFFICERS:

Chairman / Trevor Stevens 249 Southwell Road East Rainworth NG21 0BN

Group BBS

Vice Chairman Mark Wills 41 Broxtons Wood Shrewsbury SY5 9QR

General Secretary/ Richard Twynning 24 Peel Road Mansfield NG19 6HB
co-editor

Co-Editor Ian Pare 10 Sotheby Ave Sutton in Ashfield NG17 5JX

Treasurer Alan Rutherford 13 The Circuit Wilmslow SK9 6DA

Disk Librarian Stephen Shaw 10 Alstone Rd Stockport SK4 5AH

Module/Cassette Francesco Lama 14 Granville Court Oxford OX3 0HD
Librarian

Hardware Ross Bennett 20 Oak Avenue Stockport SK6 4DN

TI-99/4A SOFTWARE REVIEWS

Spring Heeled Jack EX BASIC £6

Stainless, 10 Alstone Rd, Stockport, Cheshire SK4 5AH

A simple, uninvolved game, yet great fun to play and most addictive.

At the start the screen displays eight lines on each of which is a hole that glides along, at different speeds, to give the illusion of a series of moving pavements.

Your task is to guide a man from his starting position at the bottom up to the eight level by walking him along and jumping up through the holes. Should you allow him to fall or hit his head on the next level, by mistiming his jump, a life is lost.

Control is from the keyboard, using keys of your own choice, and he may be moved left, right or up. Points are scored for reaching each level and both current and highest scores are displayed. After you have reached the top you move onto a new, more difficult screen — more holes or obstacles such as cars to avoid.

Once a life is lost you re-start at screen one, until finally all lives have been used.

J.W.

instructions	40%
playability	85%
graphics	70%
value for money	80%



Five more for your Texas

Read before you buy — a selection of games for the TI-99/4A rates by our panel

Galactic Zoo-keeper £5.25

Microscopic, 85 Malmesbury Road, Cheddar, Halmie, Cheshire

Numbering a few zoo-keepers among my friends, I was intrigued by this game, expecting it

to perhaps deal with animal conservation on a comic scale, or at least to elevate the general awareness of society about animals.

Not so. The game is rather disappointing, and considering the size of the listing and time to load, achieves very little, relying mainly on setting the player boring, difficult, and unrewarding tasks.

Documentation exists, as the first of two programs, informing you of your task to rescue an animal supposedly at risk and apparently similar to a dragon.

Watchdogs, snail mines, an "invisible" level maze, and cages are the obstacles in your path. You have five lives, but they are not really enough considering the poor response of the keyboard scans. Although the ideal of giving the user instructions via a separate program is, in this field,

novel, it isn't particularly helpful as you have to make notes to use while playing. Games program written in TI BASIC are of necessity limited, and this program is no exception. P.B.

instructions	90%
playability	40%
graphics	65%
value for money	30%



Hunchback Havoc £5.95

Lanteris, 4 Haffenden Rd, Lutterden, Kent

A very attractive presentation in which you collect jewels hidden in a nuclear powered castle. Guide Egor through the castle unlocking doors and riddling fires or other hazards in order to turn on the reactor which makes the jewels visible.

There is a very short time limit before the reactor explodes ending the game. The main difficulty is Egor cannot carry both keys and the fire extinguisher so he must scuttle back and forth.

Four screens show different parts of the castle. Hazards come in six forms, including electrical faults and gas leaks, dealt with in a similar fashion to fires. This

does not add more difficulty, but introduces some graphic variety. Since time is very limited I found it impossible to progress beyond screen four, so I cheated by increasing the time limit so as to view all 24 screens — the same four repeated for six types of hazards. Play might become repetitive if you get that far — not a problem for me though!

I would like to see a time counting marker on screen during play so as to monitor progress. Otherwise, a well designed game with excellent graphics.

C.E.

instructions	70%
playability	90%
graphics	100%
value for money	90%



Caveman EX BASIC, Joystick £4.95

Warwick, 40 Kingsway, Newby, Scarborough, N. Yorks YO12 6SC

Your objective is to recover the throne of the Inkaplonka king through five levels. You must descend through five tunnels and brave the different hazards on each level. These include moving mines, flying shells, dangerous ground, hovering discs and fireboulders. Using the joystick in various ways according to which level you are on the throne must be reached in the best possible time.

The screen display shows all five levels, the time, best time and number of lives remaining from a maximum of five. Access

to each level is obtained by descending a ladder.

Although instructions are provided on screen, because they are loaded separately you need to make notes of how to use the joystick on each level before actually starting. Written instructions on the cassette insert would be more helpful.

Any enjoyment of the game itself is spoilt by various bugs inherent in the program. I failed to get any further than level three, being prevented not only by bad luck but by two BAD VALUES and a SYNTAX ERROR.

J.W.

instructions	90%
playability	50%
graphics	65%
value for money	50%



Bouncing Bugs £6

Stainless, 10 Alstone Rd, Stockport, Cheshire SK4 5AH

Three games in one, each set on a different screen but the style of play remains the same for all.

The stars are five differently coloured bugs. Each in turn

stands on a springboard positioned to the right of the screen. Using keyboard control the springboard must be raised or lowered to the correct height from which to launch the bug, so he land on various targets to the left.

Once released the bug bounces off any walls that he comes to. Keys 1-9 are also used to determine the strength of his bounce. Each game may be played by up

to 10 people who may compete in one to 10 rounds. Points are awarded for every successful landing, the total being displayed after each completed round.

Since there is no random element, before long the exact requirements to land the bugs can be remembered. In the multi-player situation the last player undoubtedly has an unfair advantage as he is able to benefit

from all previous player's experiments and successes.

J.W.

instructions	90%
playability	50%
graphics	60%
value for money	60%



LINEAR REGRESSION

By

Francesco Lama

One of the most common things one is asked to do during a school science project is to plot data on a graph to show the behaviour of one variable against another. In my previous article I have supplied a TML (The Missing Link) based program for plotting any such data after creating appropriate X-Y files. In my program I chose to join up each pair of subsequent points with a straight line, a process which is called "Straight Line Interpolation". I could have equivalently chosen to use parabolas through three subsequent points, third degree polynomials through four subsequent points etc.. None of these would necessarily represent the real behaviour of the dependent variable (Y) against the independent one (X), but, if the points are chosen to be closely spaced, the difference between the plot and the real behaviour is negligible.

In this article I am going to talk about "Linear Behaviour", that is the case in which Y can be said to depend on X in the following way:

$$Y = mX + c$$

Where 'M' is a constant known as the "Gradient" (or the Angular Coefficient) of the straight line plot relating Y to X, and "c" is another constant. If (X1, Y1) and (X2, Y2) are two points which belong to the above straight line, then:

$$m = \frac{Y2 - Y1}{X2 - X1}$$

and 'c' is the value of Y when X=0, that is the point at which the straight line crosses the Y axis (also known as the Y intercept).

Imagine we have determined a number of experimental points (through measurement) (X1,Y1), (X2,Y2), (Xi,Yi), (Xn,Yn). and that we want to find a way of determining the best straight line through them. Clearly what needs to be determined are the constants 'm' and 'c' in the straight line expression above, which define the line

is closest to all the points. In order to accomplish this we must first define the residual for each experimental point. For example the residual for the i th point (X_i, Y_i) is given by:

$$R_i = Y_i - (mX_i + c)$$

where ' i ' here simply represents a general label for an experimental point and is therefore just an index (i.e. Y_i does not mean Y multiplied by i , but simply the i th experimental value of the dependent variable).

If we now add the R_i 's squared we obtain an expression which, one can intuitively perceive, will attain its lowest value when the straight line lies closest to the experimental points. The values of ' m ' and ' c ' which produce such a straight line are obtained by setting the partial derivatives of the above sum with respect to ' m ' and ' c ' equal to zero and solving the simultaneous equations so obtained. This procedure of minimization of the residuals is known as "Linear Regression".

Enough of the mathematics; the program which follows is not only able to provide the values of " m " and " c " for a given set of experimental points, but can also plot the points (complete with their error boxes) and the best fit line through them.

Before I launch into a brief explanation of how the program works, I would like to explain how data points for use with this program must be specified. Since each experimental determination of any quantity (temperature, resistance, length, pressure etc..) involves an error which represents the limit of the accuracy of our measuring method, we must only quote our experimental points as our two readings of X and Y , but also their corresponding experimental uncertainties $DELTX$ and $DELTY$. Therefore the experimental point, far from being infinitesimally small (as points defined in mathematics), is actually a rectangular box with vertices at $(X-DELTX, Y-DELTY)$, $(X+DELTX, Y-DELTY)$, $(X+DELTX, Y+DELTY)$, and $(X-DELTX, Y+DELTY)$, corresponding to an uncertainty about X and Y of $\pm DELTX$ and $\pm DELTY$ respectively (i.e. X and Y can have any value in the range $[X-DELTX, X+DELTX]$ and $[Y-DELTY, Y+DELTY]$ respectively).

I will now concentrate on the main differences between the present program and the one I presented in the Spring issue (N.56).

The DIM statement is clearly different in this case to take account of the different types of arrays needed in a linear regression.

The initialization of the variables at line 100 is also correspondingly different, because it needs to introduce the variables DELTXMA and DELTYMA, the maximum values of DELTX and DELTY respectively (in general they are not the same for all the points).

Line 110 introduces a FLAG which is set equal to "0" at this point. A change of its value to "1" after the execution of lines 1030-1061 prevents them from being executed more than once, unless the program goes through another initialization procedure.

Lines 480-640 have had to be adapted to the reading of the new type of file used, and lines 690-800 have been changed (normalization is meaningless in this case) in order to present the numerical results Of the Linear Regression Procedure and wait for an input (C) to continue execution.

The plotting procedure selection part of the program (lines 1000-1261) only required minor modifications. Because of the presence of errors, lines 1025-1060 were introduced so that the full error boxes would appear on the screen (these operations must, of course, only be carried out once, hence the FLAG=1 at line 1061).

Finally lines 1253 and 1254 are introduced to enable one to choose whether to plot the error boxes on screen or just the points (these two alternative types of plot, which were absent in the original X-YPLOTML program, are accomplished by two new subroutines at 4500 and 5000 respectively).

The Axes Plotting part of the program (lines 1270-1430) is practically identical to the one in the other program but, since we are now plotting only one curve, the Curve Plotting Routine is reduced to four lines which also take account of whether Boxes or Crosses should mark the experimental points (lines 1500-1530).

The only other major difference between the present program and the original plotting program lies in three additional subroutines at 4000, 4500, and 5000. The first is called by line 700 and produces "m", "c", and the statistical correlation coefficient, and the remaining two have already been discussed above.

Two more Extended BASIC programs are supplied for use with LEASTSQtml (the program presented here). The first, called MK_LSQFILE, enables one to produce a suitable experimental data

Linear Regression

file, whether experimental errors are going to be entered explicitly (subroutine 500) or not (subroutine 1000).

The second (RD_LSQFILE) is simply a means of reading a file produced by MK_LSQFILE and printing it on the screen to check for possible mistakes.

I have used these programs to plot some of my students experimental results, and I have found them quite useful. I hope you do too.

```

REQUIRES THE MISSING
LINK SOFTWARE UTILITY
1 REM PROGRAM LEASTSQtml
5 REM Version 2.1
10 REM This program reads in a data file for a
linear least squares fit, produces the values
of the parameters
20 REM of the best fit straight line, and plots
the results. The X/Y arrays contain the coor-
dinates of the points to which the fit is
25 REM applied, and the DELTX/DELTY arrays con-
tain the errors on the coordinates of the
points.
30 REM The arrays start from element 0.
50 DIM X(500),Y(500),DELTX(500),DELTY(500)
60 CALL LINK("CLEAR")
65 CALL LINK("COLOR", 16,5)
70 REM
80 REM initialization of the variables
90 REM
100 XMI=1E33 :: XMA=-1E33 :: YMA=-1E33
::DELTXMA=-1E33 :: DELTYMA=-1E33
105 REM
106 REM This flag is set equal to zero here.
When its value is other than 0 it prevents the
repetition of line 1030-1061, at the end of
107 REM it is set equal to 1
108 REM
110 FLAG=0
```

```
163 CALL LINK("WINDOW")
450 REM
460 REM File Loading
470 REM
480 CALL LINK("CLEAR")
481 CALL LINK("WINDOW")
500 CALL LINK("PRINT", 1+16*L, 1, "FILE
NAME?")
510 CALL LINK("INPUT", 1+16*L, 100, NAME$)
540 OPEN #1:NAME$, SEQUENTIAL, INTERNAL,
INPUT, FIXED
550 INPUT #1:N
580 FOR I=0 TO N
590 INPUT #1:X(I),Y(I),DELTX(I),DELT(Y(I)
593 IF X(I)>XMA THEN XMA=X(I)
594 IF DELTX(I)>DELTXMA THEN DELTXMA =
DELTX(I)
597 IF X(I)<XMI THEN XMI=X(I)
600 IF Y(I)>YMA THEN YMA=Y(I)
601 IF DELTY(I)>DELT(YMA THEN DELTYMA =
DELT(Y(I)
610 IF Y(I)<YMI THEN YMI=Y(I)
620 NEXT I
640 CLOSE #1
660 REM
670 REM Calculation and presentation of least
squares procedure results.
680 REM
690 CALL LINK("CLEAR")
693 CALL LINK("WINDOW")
700 GOSUB 4000
710 CALL LINK("PRINT",1,7,"RESULTS OF LEAST
SQUARE PROCEDURE")
```

Linear Regression

```
720 CALL LINK("PRINT",10,1,"ANGULAR COEFFICIENT
=")
730 CALL LINK("PRINT", 10,160,M)
740 CALL LINK("PRINT", 20,1,"INTERCEPT =")
750 CALL LINK("PRINT", 20,160,B)
760 CALL LINK("PRINT", 30,1,"CORRELATION COEFFI-
CIENT =")
770 CALL LINK("PRINT", 30,160,R)
775 CALL LINK("PRINT", 50,10,"PRESS C TO CON-
TINUE")
780 CALL KEY(0,KP,SP)
790 IF SP=0 THEN 780
800 IF KP<>67 THEN 780
998 REM
999 REM plotting procedure selected here
1000 REM
1008 CALL LINK("CLEAR")
1009 CALL LINK("WINDOW")
1010 CALL LINK("PRINT",1,1,"PLOT DATA (Y/N)?")
1011 CALL LINK("INPUT", 1,200,A$)
1020 IF A$<>"Y" AND A$<>"y" THEN 1610
1025 IF FLAG<>0 THEN 1090
1030 XMI=XMI-DELTXMA
1040 XMA=XMA+DELTXMA
1050 YMI=YMI-DELTYMA
1060 YMA=YMA+DELTYMA
1061 FLAG=1
1090 CALL LINK("PRINT", 9,1,"MINIMUM X=")
1091 CALL LINK("PRINT", 9,100,XMI)
1100 CALL LINK("PRINT",17,1,"MAXIMUM X=")
1101 CALL LINK("PRINT",17,100,XMA)
1110 CALL LINK("PRINT",25,1,"MINIMUM Y=")
```



```
1111 CALL LINK("PRINT", 25, 100, "MAXIMUM Y=")
1120 CALL LINK("PRINT", 33, 1, "MAXIMUM Y=")
1121 CALL LINK("PRINT", 33, 100, YMA)
1130 CALL LINK("PRINT", 41, 1, "X-RANGE?")
1131 CALL LINK("INPUT", 41, 100, XMI)
1132 CALL LINK("INPUT", 49, 100, XMA)
1140 CALL LINK("PRINT", 57, 1, "LENGTH OF X-DIV?")
1141 CALL LINK("INPUT", 57, 150, XDIV)
1150 NXDIV=INT((XMA-XMI)/XDIV)
1160 CALL LINK("PRINT", 65, 1, "LONGER X-DIV (1, 2
etc.)?")
1161 CALL LINK("INPUT", 65, 180, LMARX)
1170 CALL LINK("PRINT", 73, 1, "X-AXIS LABEL (MAX 30
CHARS)")
1171 CALL LINK("INPUT", 81, 1, XLAB$)
1180 CALL LINK("PRINT", 100, 1, "Y-RANGE?")
1181 CALL LINK("INPUT", 100, 100, YMI)
1182 CALL LINK("INPUT", 108, 100, YMA)
1190 CALL LINK("PRINT", 116, 1, "LENGTH OF Y DIV?")
1191 CALL LINK("INPUT", 116, 160, YDIV)
1200 NYDIV=INT((YMA-YMI)/YDIV)
1210 CALL LINK("PRINT", 124, 1, "LONGER Y DIV (1, 2
etc.)")
1211 CALL LINK("INPUT", 124, 180, LMARY)
1220 CALL LINK("PRINT", 132, 1, "Y-AXIS LABEL (MAX 20
CHARS)?")
1221 CALL LINK("INPUT", 140, 1, YLAB$)
1230 CALL LINK("PRINT", 156, 1, "PLOT TITLE (MAX 30
CHARS)?")
1231 CALL LINK("INPUT", 164, 1, TIT$)
1240 CALL LINK("PRINT", 172, 1, "COORDINATES OF ORI-
GIN?")
1241 CALL LINK("INPUT", 180, 1, XX0)
```

Linear Regression

```
1242 CALL LINK("INPUT",180,100,YY0)
1250 CALL LINK("CLEAR")
1251 CALL LINK("WINDOW")
1252 IF DELTXMA=0 AND DELTYMA=0 THEN 1255
1253 CALL LINK("PRINT",1,1,"ERROR BOXES (Y/
N)?")
1254 CALL LINK("INPUT",1,160,BOX$)
1255 CALL LINK("PRINT",10,1,"FULL GRID (Y/N)?")
1256 CALL LINK("INPUT",10,160,A$)
1260 CALL LINK("PRINT",20,1,"HARD COPY (Y/N)?")
1261 CALL LINK("INPUT",20,160,HC$)
1270 CALL LINK("CLEAR")
1271 CALL LINK("WINDOW",12,10,180,241)
1280 REM
1290 REM drawing X-axis
1300 REM
1310 YY=YY0 :: GOSUB 1730 :: IYP0=IYP
1320 XX=XMI :: GOSUB 1680 :: IXPO=IXP :: XX=XMA
:: GOSUB 1680 :: CALL
LINK("LINE",IYP0,IXPO,IYP0,IXP)
1330 IF A$="Y" OR A$="y" THEN GOSUB 2500 ELSE
GOSUB 2000
1380 REM
1390 REM drawing Y-axis
1400 REM
1410 XX=XX0 :: GOSUB 1680 :: IXPO=IXP
1420 YY=YMI :: GOSUB 1730 :: IYP0=IXP :: YY=YMA
:: GOSUB 1730 :: CALL LINK("LINE",IYP,IXPO,
IYP0,IXPO)
1430 IF A$="Y" OR A$="y" THEN GOSUB 3500 ELSE
GOSUB 3000
1497 REM
1498 REM curve plotting routine
```

```

1499 REM
1500 IF BOX$="Y" OR BOX$="y" THEN
GOSUB 4500 ELSE GOSUB 5000
1510 XX=XMI :: GOSUB 1680 :: IXP0=IXP ::
YY=M*XX+B :: GOSUB 1730 :: IYP0=IYP
1520 XX=XMA :: GOSUB 1680 :: YY=M*XX+B :: GOSUB
1730
1530 CALL LINK("LINE", IYP0, IXP0, IYP, IXP)
1571 REM printing X-axis label
1572 CALL LINK("WINDOW", 182, 11, 193, 241)
1573 CALL LINK("PRINT", 1, 29, XLAB$)
1574 REM printing Y-axis label
1575 CALL LINK("WINDOW", 0, 0, 193, 10)
1576 LY=LEN(YLAB$)
1577 FOR I=1 TO LY
1578 CALL LINK("PRINT", I*8, 1, SEG$(YLAB$,
I, 1))
1579 NEXT I
1580 REM printing plot title
1581 CALL LINK("WINDOW", 0, 30, 10, 241)
1582 CALL LINK("PRINT", 1, 30, TIT$)
1589 CALL KEY(0, KP, SP)
1590 IF SP=0 THEN 1589
1600 IF KP<>67 THEN 1589
1605 IF HC$="Y" OR HC$="y" THEN CALL
LINK("DUMP")
1610 CALL LINK("CLEAR")
1615 CALL LINK("WINDOW")
1620 CALL LINK("PRINT", 1, 1, "LOAD NEW FILE?")
1625 CALL LINK("INPUT", 1, 160, A$)
1627 IF A$="Y" OR A$="y" THEN 60
1628 CALL LINK("PRINT", 20, 1, "USE SAME FILE FOR
NEW PLOT?")

```

Linear Regression

```
1629 CALL LINK("INPUT",20,192,A$)
1630 IF A$="Y" OR A$="y" THEN 690
1640 END
1650 REM
1660 REM calculates the screen value of the X co-
ordinate of a point
1670 REM
1680 IXP=INT(15_225*(XX-XMI)/(XMA-XMI))
1690 RETURN
1700 REM
1710 REM calculates the screen value of the Y co-
ordinate of a point
1720 REM
1730 IYP=INT(174-156*(YY-XMI)/(YMA-YMI))
1740 RETURN
1997 REM
1998 REM Draws X-axis divisions
1999 REM
2000 FOR I=0 TO NXDIV
2010 XX=XMI+I*XDIV :: GOSUB 1680
2020 IF (LMARX-I-1)/10=INT((LMARX-I-1)/10) THEN
CALL LINK("LINE",IYP0-5,IXP,IYP0+4, IXP) :: GOTO
2040
2040 NEXT I
2050 RETURN
2497 REM
2498 REM Draws X-axis grid
2499 REM
2500 YY=YMI :: GOSUB 1730 :: IYP0=IYP :: YY=YMA
:: GOSUB 1730
2505 FOR I=0 TO NXDIV
2507 XX=XMI+I*XDIV :: GOSUB 1680
2510 IF (LMARX-I-1)/10=INT((LMARX-I-1)/10) THEN
```

Linear Regression

```
CALL LINK("LINE", IYP, IXP, IYP0+4, IXP) ::
GOTO 2530
2520 IF (LMARX-I-1)/5=INT((LMARX-I-1)/5) THEN
CALL LINK("LINE", IYP, IXP, IYP0+2, IXP) ELSE
CALL LINK("LINE", IYP, IXP, IYP0, IXP)
2530 NEXT I
2540 RETURN
2997 REM
2998 REM Draws Y-axis divisions
2999 REM
3000 FOR I=0 TO NYDIV
3010 YY=YMI+I*YDIV :: GOSUB 1730
3020 IF (LMARY-I-1)/10=INT((LMARY-I-1)/10) THEN
CALL LINK("LINE", IYP, IXP0+5, IYP, IXP0-4) ::
GOTO 3040
3030 IF (LMARY-I-1)/5=INT((LMARY-I-1)/5) THEN
CALL LINK("LINE", IYP, IXP0+5, IYP, IXP0-2) ELSE
CALL LINK("LINE", IYP, IXP0+5, IYP, IXP0)
3040 NEXT I
3050 RETURN
3497 REM
3498 REM Draws Y-axis grid + cross over origin
3499 REM
3500 XX=XMI :: GOSUB 1680 :: IXP0=IXP :: XX=XMA
:: GOSUB 1680
3505 FOR I=0 TO NYDIV
3507 YY=YMI+I*YDIV :: GOSUB 1730
3510 IF (LMARY-I-1)/10=INT((LMARY-I-1)/10) THEN
CALL LINK("LINE", IYP, IXP, IYP, IXP0-4) :: GOTO
3530
3520 IF (LMARY-I-1)/5=INT((LMARY-I-1)/5) THEN
CALL LINK("LINE", IYP, IXP, IYP, IXP0-2) ELSE
CALL LINK("LINE", IYP, IXP, IYP, IXP0)
3530 NEXT I
3532 XX=XX0 :: GOSUB 1680 :: IXP0=IXP :: YY=YY0
```

Linear Regression

```
:: GOSUB 1730 :: IYP0=IYP
3534 CALL LINK("LINE", IYP0+4, IXP0+4, IYP0-4,
IXP0-4)
3536 CALL LINK("LINE", IYP0+4, IXP0-4, IYP0-4,
IXP0+4)
3540 RETURN
3997 REM
3998 REM least squares subroutine
3999 REM
4000 XBAR=0 :: YBAR=0 :: WT=N+1 :: SIGX2=0 ::
SIGXY=0
4010 FOR I=0 TO N
4020 XBAR=XBAR+X(I) :: YBAR=YBAR+Y(I)
4025 SIGX2=SIGX2+X(I)^2 :: SIGY2=SIGY2+Y(I)^2 ::
SIGXY=SIGXY+X(I)*Y(I)
4030 NEXT I
4040 XBAR=XBAR/WT :: YBAR=YBAR/WT ::
SIGX2=SIGX2/WT-XBAR^2 :: SIGY2=SIGY2/WT-YBAR^2
:: SIGXY=SIGXY/WT-BAR*YBAR
4050 M=SIGXY/SIGX2 :: B=YBAR-M*XBAR ::
R=M*SQR(SIGX2/SIGY2)
4060 RETURN
4497 REM
4498 REM plots the error boxes
4499 REM
4500 FOR I=0 TO N
4510 XX=X(I)-DELTX(I) :: GOSUB 1680 :: IXP0=IXP
:: YY=Y(I)+DELTY(I) :: GOSUB 1730 :: IYP0=IYP
4520 XX=X(I)+DELTX(I) :: GOSUB 1680 :: YY=Y(I)-
DELTY(I) :: GOSUB 1730
4530 CALL LINK("BOX", IYP0, IXP0, IYP, IXP)
4535 XX=X(I) :: GOSUB 1680 :: YY=Y(I) :: GOSUB
1730
4536 CALL LINK("CIRCLE", IYP, IXP, 1)
4540 NEXT I
```

```
4550 RETURN
4997 REM
4998 REM plots the points as crosses
4999 REM
5000 FOR I=0 TO N
5010 XX=X(I):: GOSUB 1680 :: YY=Y(I) :: GOSUB
1730
5020 CALL LINK("LINE", IYP-3, IXP, IYP+3, IXP)
5030 CALL LINK("LINE",IYP, IXP-3, IYP, IXP+3)
5040 NEXT I
5050 RETURN
```

```
5 REM Program MK_LSQFILE
10 REM This program creates X-Y files which can
then be used and plotted with the least squares
analysis program LEASTSQtml.
20 REM
22 CALL CLEAR
25 DIM X(500),Y(500),DELTX(500),DELT Y(500)
30 INPUT "Nr of X-Y pairs (MAX 501)":N
31 INPUT "ERRORS (Y/N)?:":A$
32 REM the reason for the N-1 instead of N is
that I am working in option base 0.
35 N=N-1
40 IF A$="Y" OR A$="y" THEN GOSUB 500 ELSE GOSUB
1000
80 INPUT "ENTER FILE NAME ":NAME$
90 OPEN #1:NAME$,SEQUENTIAL,INTERNAL,OUTPUT,FIXED
100 PRINT #1:N
110 FOR I=0 TO N
120 PRINT #1:X(I),Y(I),DELTX(I),DELT Y(I)
130 NEXT I
```

Linear Regression

```
135 CLOSE #1
140 END
500 FOR I=0 TO N
505 PRINT
510 PRINT "X";I;"",DX";I;"",Y";I;"",DY";I
520 INPUT X(I),DELTX(I),Y(I),DELTY(I)
530 NEXT I
540 RETURN
1000 FOR I=0 TO N
1010 DELTX(I)=0:: DELTY(I)=0
1015 PRINT
1020 PRINT "X";I;"",Y";I
1030 INPUT X(I),Y(I)
1040 NEXT I
1050 RETURN
```

```
5 REM Program RD_LSQFILE
10 REM This program reads in and prints on
screen least squares files created with
MK_LSQFILE
22 CALL CLEAR
25 DIM X(500),Y(500)
80 INPUT "ENTER FILE NAME ":NAME$
90 OPEN #1:NAME$,SEQUENTIAL, INTERNAL, INPUT,
FIXED
100 INPUT #1:N
110 FOR I=0 TO N
120 INPUT #1:X(I),Y(I),DELTX(I),DELTY(I)
130 NEXT I
135 CLOSE #1
140 FOR I=0 TO N
145 IF I/7=INT(I/7) THEN PRINT TAB(6);"X";
```



```

TAB(22);"Y"
150 PRINT TAB(0);X(I);TAB(15);Y(I)
155 IF I/7=INT(I/7) THEN PRINT TAB(3); "DELTX";
TAB(19); "DELTY"
156 PRINT TAB(0);DELTX(I);TAB(15);DELTY(I)
157 PRINT
160 NEXT I
170 INPUT "AGAIN (Y/N)?":A$
180 IF A$="Y" OR A$="y" THEN 140
190 INPUT "LOAD ANOTHER FILE (Y/N)?":A$
200 IF A$="Y" OR A$="y" THEN 22
210 END

```

In order to give you some idea of how to go about producing suitable data files, I have here two examples of data taken by my students which can be plotted with LEASTSQtml. First of all load and run MK_LSQFILE in Extended BASIC and type in the following data to create a data file with a name of your choice:

X	DELTX	Y	DELTY
27	0.5	3.0	0.1
98	0.9	1.90	0.08
4.5	0.5	1.39	0.06
3.1	0.3	1.21	0.05
1.6	0.2	1.08	0.04

This table shows the relationship between IMPEDANCE SQUARED (Y in units of OHMS SQUARED TIMES 10000) against ONE OVER CAPACITANCE SQUARED (X in units of ONE OVER FARADS SQUARED TIMES 100000000) for a resistor-capacitor (RC) circuit in series with the mains. The resistor was 100Ω.

You may want, for the sake of variety, to create another file which

Linear Regression

relates to the extension of a spring to the increasingly larger masses which are suspended from it (another example of linearity).

X	DELTX	Y	DELTY
0.100	0.001	0.021	0.004
0.200	0.002	0.066	0.004
0.300	0.003	0.110	0.004
0.400	0.004	0.154	0.004
0.500	0.005	0.193	0.004
0.600	0.006	0.236	0.004
0.700	0.007	0.284	0.004
0.800	0.008	0.327	0.004

In this case X is the MASS expressed in kilograms, and Y is the extension in metres. Now run LEASTSQtml in the TML Environment and use the above files to generate your plots.

The End

Letters to the Editor(s)

Dear Ian,

Following the Summer issue of TI*MES which I received recently, I discovered several old computer magazines gathering dust at my parents house, the contents of which may (or may not!) be of interest to you.

The period 1982-85 was one of great excitement for me, leaving school and studying computers a college, spending endless hours with my trusty TI-99/4A. Unfortunately my work as a professional musician nowadays severely restricts the time I have available to devote to computers, however I still own all my original TI gear and am proud to be a member of the TIUG.

Please find enclosed a program listing from Computer & Video Games October 1984. This issue also contains a review for Buck Rogers in cartridge format. Please also find enclosed a copy of a TI-99/4A hardware review as taken from Electronics & Computing Monthly in May 1982. There are some interesting ads contained within this review - a TI-99/4A is available from Rumbelows for an incredible £329.99 with credit terms available at 42.1% APR !!! Would you by a computer off a man wearing a tank-top like that !!??

I hope these articles are of some interest to you.

Kind Regards

M.Deighton

You can't get a Texas Instruments Home Computer from Rumbelows under 16 K RAM



£329.99

RUMBELOW'S
We save you money and serve you right.

42 - ELECTRONICS & COMPUTING MAY 1982

TEXAS INSTRUMENTS

With the Home Computer from Texas Instruments, you can converse in the five major languages: BASIC, PASCAL, TI-LOGO, ASSEMBLER and it speaks English!



42 - ELECTRONICS & COMPUTING MAY 1982

Thanks for the letter & info Mitch, we'll print the C&VG listing in the Winter Issue of TI*MES. And yes I'd probably have to think twice, before parting with my cash to this man!

Ed.

Thanks also to Ken Hughes, for all of the info he's sent to me ! I'll make sure some of it goes in the next issue... and I promise I will send them back to you, thanks once again !

Ed.

C Regena

GRAPHICS

"Graphics" or drawing pictures on the TI can be a lot of fun, and using graphics in your programs can really enhance them. The TI has 16 colors, and all 16 colors may be used at the same time on the screen, even with high-resolution graphics. Later in this column, I will discuss user-defined graphics characters.

Video-Graphs

First, let's briefly review the TI Video-Graphs command module, since using the command module is an easy way to see graphics on the TI without actually programming. You may see different random color patterns, or you may draw pictures on the screen using the arrow keys and a few function keys. You may save or load a picture on cassette tape.

Because Video-Graphs was one of the first command modules produced by TI, the manual you get with your module may be written for the TI-99/4 console. There are some changes that are necessary for the module to work with the TI-99/4A console. (By the way, if you have the TI-99/4 console, be sure to use the overlay that comes with the module or ask Texas Instruments to send you an overlay. The overlay has all the colors and commands so you don't need to keep referring to the manual.)

Make these changes for the TI-99/4A console. Instead of pressing ENTER, press the period key to return to the activity selection list. You will also need to press the period instead of zero to return to the main index lists. The comma key represents the color GRAY. To save a picture or to get to the TAPE options, press the semicolon key. To change colors, use the virgule/slash key.

The "Patterns" option presents three different random graphics demonstrations. STOP a picture by pressing N. You can't change colors while a picture is stopped. To restart the picture, press 6. While a pattern is going, you may change colors. Let's say you are looking at pulsing lights and want to change all the white squares to magenta. Press / then M then 0.

The "Pictures" option presents four different ways you can draw on the computer. Mosaic and Sketchpad are like using a pen directed by the arrow keys. Color Life is designed to be like the venerable computer game "Life," which replicates cells according to strict rules. Building Blocks has several shapes at the bottom of the screen. You may move the cursor to the shape you want for your picture, then press Y for the pen and move the shape up to your picture. Again, you may change colors by pressing / followed by the present color and then the color desired.

Programming Your Own Graphics

Think of the screen on your monitor or television set as a rectangle divided up into 24 rows and 32 columns. To graphically place a character on the screen, you specify the row number, the column number, and the character number - the ASCII code number of the character you desire. You may also specify a number of repetitions. CALL HCHAR(8,5,65,7) will start in row 8 and column 5 and draw character number 65, which is the letter A, seven times horizontally. CALL VCHAR(12, 14,66,9) will draw the letter B nine times vertically, starting in row 12 and column 14.

If you don't want to draw a picture using A's and B's or the other letters and symbols available, you can define your own high-resolution characters. Each square in the 24 x 32 rectangle can be divided up into an 8 x 8 square, and each dot in that 8 x 8 square can be turned on or off - colored in or not. By specifying with code numbers which dots you want on and which you want off, you can define your own graphics character and then place it on the screen.

Here is an example. I want to draw a small triangle. The dots in the 8 x 8 square are colored in. The next step is to divide the square in half so that there are columns of four squares on each side. Now, working left to right and downward, figure out the hex code for each pattern of four squares. In the first row, 0000 is 0 and 0001 is 1. In the second row 0000 is 0 and 0011 is 3. Continue

78 COMPUTE May 1983

2023 scanning note: The next few pages were copied from COMPUTE! magazine and have been reduced too much to be very readable - especially the listing at the end.

Programming the TI - Graphics

```

530 IF K=13 THEN 760
540 IF Y=70 THEN 740
550 IF K=32 THEN 720
560 IF K<68 THEN 600
570 IF Y=13 THEN 470
580 Y=Y+1
590 GOTO 480
600 IF K<>88 THEN 640
610 IF X=23 THEN 470
620 X=X+1
630 GOTO 480
640 IF K<>83 THEN 680
650 IF Y=6 THEN 470
660 Y=Y-1
670 GOTO 480
680 IF K<>69 THEN 490
690 IF X=15 THEN 470
700 X=X-1
710 GOTO 480
720 CALL HCHAR(X,Y,97)
730 GOTO 470
740 CALL HCHAR(X,Y,98)
750 GOTO 470
760 CALL SOUND(150,440,2)
770 D$=""
780 FOR I=1 TO 8
790 C$=""
800 FOR J=6 TO 9
810 CALL GCHAR(I+15,J,C)
820 C$=C$CHR$(C)
830 NEXT J
840 GOSUB 1050
850 CALL HCHAR(I+15,16,ASC(D1$))
860 D$=D$&D1$
870 C$=""
880 FOR J=10 TO 13
890 CALL GCHAR(I+15,J,C)
900 C$=C$CHR$(C)
910 NEXT J
920 GOSUB 1050
930 CALL HCHAR(I+15,17,ASC(D1$))
940 D$=D$&D1$
950 NEXT I
960 CALL CHAR(136,D$)
970 CALL HCHAR(20,20,136)
980 PRINT "DEFINITION = ";D$
990 PRINT "PRESS 1 TO MODIFY"
1000 PRINT "(6 SPACES) TO START OVE
R"
1010 PRINT "(6 SPACES)3 TO END PROGR
AM":
1020 CALL KEY(0,K,S)
1030 IF (K=49)+(K=50) THEN 220
1040 IF K=51 THEN 1110 ELSE 1020
1050 FOR L=0 TO 15
1060 IF C$=H$(L,1) THEN 1090
1070 NEXT L
1080 L=L-1
1090 D1$=H$(L,2)
1100 RETURN
1110 PRINT ":
1120 END

```

Program 2: Graphics Demonstration

```

120 CALL CLEAR
130 FOR C=33 TO 140
140 READ C$
150 CALL CHAR(C,C$)
160 NEXT C
170 DATA FFFFFFFFFFFFFFFF,0001070F1
F3F7F7F,40C080000000800,00000000

```

```

003C4582,00000304001020E,7FC
180 DATA C0303F080402,00008768101008
.0000006C12473804,00040607030303
07,0000000080E0EF,E0FFFFFFF0FF
FFF
190 DATA 0102FFFFFFF0AFC,05489020C0
8,0000030301110E,00808000CF3,07
0F3F2F271D0602,F0CF0FFFFFFF1F0D
200 DATA 0000FFFFFFF0FFF,0F1FFFFFFF
FFFFFFF,FCFCFCFCFCFCFCFC,7F7F7F3
F1F1F2F2,FFF0FFFFFFCF0C,FCF9FA0D
210 DATA 7080384488102021,0300010204
040402,438C304040818282,0E166EBF
7E,FFFFFFFF0F0301,F8F0F0E0C08
220 DATA 0000000106040E0F,2020418307
1F7FFF,0080000808CF0FFF,0000806A
7FFFFFFF,2224455EFFFFFFF
230 DATA 01FD03798503010D,04B4242424
1C0101,0C083040809020C,0080707C3
E3E1F1F,0000101C3E3FFFFFFF
240 DATA 38300E81406,00000800C020100
C,1F1F3F3F7F7F797,FFFFFFFFFCFA
FD,FEFFFCFCF858810B,749C2008A8FB
FCFC
250 DATA 804040402020401008,1F0F0F0F
070707E7,070301,FFFFFFF7F,F4E9CB8
30F070707,17FFFFFF9DFCFEFE
260 DATA 0F000818FCFCFCFC,F8C807060
906,38D890187C94E407,FFFFFFFF09
0909,FFFFFFFF7F3F1F272,FFFFFFFFCF8
CF8
270 DATA 0303030301010101,7F7F7FDF8E0
FFFFFFF,070730101010303,FFFFFFCF
CFCEFCF1,9090A0A06040C09,202020
202020202
280 DATA FEFCF8F0E0C0C001,1010202040
439418,204040808,7F7F3F3F1F1F0
F,FCF8F0E0E0E0E06FFFF,00001F205F84C
7E
290 DATA 0404848480C0F3FF,00000000C0
20101,1202020204040808,0F0F0707
70737C7,FFFFFFFFB38080F0FF,E0E0C08
000003FFF
300 DATA 7F7F7F3E1C0000F,808000181C1
E3F7E,1010202040808038,FFFF3F3F3
F3F1F1F,FEFEFEFCF0F2F1F,0F0F,FF7
F,FFF0
310 DATA F0E,0780402018050381,00E040
808,4222120A06020101,86463A01,80
0000003050505,000003FC,408
320 DATA 00808101010202,00808404040
40404,00000884020101,000007080A
0A04,033342800810204,0E708001020
C106
330 DATA 800040201010102,80010200808
080C,A0100F,00010638C,0080407807
.40408008080808,808080808E513E2
340 DATA 000007182020404,00C02010000
00001
350 PRINT TAB(6);"0% %&'()* +,."
360 PRINT TAB(6);"!'-/ 0123456"
370 PRINT TAB(6);"789: ;<=>?@"
380 PRINT TAB(5);"0ABCDE FGH IJK"
390 PRINT TAB(5);"L MNOP(3 SPACES)Q
R,S,"
400 PRINT TAB(6);"RSTUVWXYZ[\"
410 PRINT TAB(8);"^\] ^ _`a"
420 PRINT TAB(9);"!@6 \bcd"
430 PRINT TAB(9);"e'fghi j"
440 PRINT TAB(9);"k!lmnop"

```

```
450 PRINT TAB(10):"q' 'r"  
460 PRINT TAB(10):"s tuv":;;;:  
470 FOR I=1 TO 25  
480 READ X,Y,C  
490 CALL HCHAR(X,Y,C)  
500 NEXT I  
510 DATA 18,17,119,18,18,120,19,17,1  
21,20,18,122,19,18,123,20,19,124  
.20,20,125,1  
9,20,126  
520 DATA 18,20,127,17,20,128,17,19,1  
29,18,11,130,18,10,131,19,11,132  
.20,11,125,2  
0,10,134  
530 DATA 19,10,133,20,9,135,20,0,136  
.19,8,137,18,8,138,17,8,139,17,9  
.39,17,10,14  
0,1,1,32  
540 GOTO 540  
550 END
```

©

RAMBLES.

by *Stephen Shaw.*

In Memorium- Ray Kazmer.

For many of you the name of Ray Kazmer may mean little. Perhaps you saw briefly his game The Maze of Grog, or his high quality Extended Basic demo program WOOD-STOCKS CHRISTMAS PRESENT.

Since before this group was established Ray Kazmer supported UK TI users by supplying a significant volume of programs for the disk library, gathered from many US user groups. Ray also provided support to Ken Gilliland, of Notung Software. Ray passed away on Tuesday morning (Los Angeles time) September 9th. "Ye Olde Guru" as Ray called himself will be missed.

I am sorry to have had no material in the last newsletter- a mixture of not knowing who to send text to for what date, plus Summer Holidays, and some time settling my father into a nursing home. Oh yes- and yet another job change coming up shortly...

Rambles by Stephen Shaw

With no contact from other TI users, I have little to comment on- however it IS notable just how many web pages now mention or are dedicated to the old humble TI.

I heard from Austin Platt, author of the program ARIES, sold some years ago by Stainless Software...

"I don't expect you remember me but I wrote an Extended Basic game called 'Aries'. It was a scrolling shoot 'em up game.

I'm writting ATC software in The Netherlands, Germany and Austria using the programming skills I honed to perfection tapping away in my bedroom all those years ago. I wonder how many other programming careers werestarted by the good old TI-99/4A.

Cheers

Austin Platt
Siemens Air Traffic Management

Malcolm Adams was possibly the widest published UK TI programmer,with programs distributed by several companies. Remember Secret Agentfrom Stainless? or Break-point from Parco? Malcolm now has a significant TI web site (with v9t9 downloads).

A couple of years ago, when our local Science Museum had an exhibition on the Internet, there were less than half a dozen pages on the TI-99/4a

Now, there are scores of web pages dealing with this ancient computer, including even the official TI site. You can even read about the TI99/2 and TI99/8.

And the old computers continue to function, although software written for them is no longer taking such advantage of their features as once was... there are numerous hardware projects, supported with a handful of software... there are areas of programming still to be covered, but alas no programmers ready to take on the task - many hours work for no reward...

For those members with access to e-mail, I can now be contacted (heck, no-one writes

to me anymore!) at shawweb@btinternet.com. In due course (perhaps) I might even host a web page- and the first thing on it would be my now rather old TI book, the text of which has always be e-text, produced with TI Writer. Watch out for www.btinternet.com/~shawweb about Christmas time!

In case there are printing problems the symbol before shawweb on the site name is a tilde, a squiggly minus sign! On the PC it is near the right hand shift key. When you see a tilde in a webpage address it means the webmaster has an email address obtained (usually) by quoting the first part after the tilde, before any following oblique strokes (/) then putting an at sign (@) and finally the first part of the web page address leaving out (usually) the domain name (eg WWW).

So: Web page = www.btinternet.com/~shawweb
e-mail to = shawweb@btinternet.com

2023- The old webpages are secured on web.archive.org for your enjoyment.

2023: New web page is: shawweb.myzen.co.uk/stephen/TI.htm (since 2011)

And I'd be grateful for any thoughts on what I can write about here!

Best wishes to you all,
Stephen Shaw

Wanted

***Old Issues of "Computer & Video Games" Magazine
& Other Computer Magazines***

also

Any TI-99/4A Cartridges & Also other 8-bit computers
Oric, Enterprise, Sinclair ZX80, Newbrain, Lynx, Aquarias, Colour
Genie, Dragon, Atari 400/800, Adam, Jupiter Ace, Tandy, Acorn Atom
etc !

for private collection, send details to :

*Ian Pare, 10 Sotheby Avenue, Sutton-In-Ashfield,
Nottinghamshire, England. NG17 5JX.
Tel : +44(0)1623 552549 or E-Mail ian@infoserv.demon.co.uk*

PC99 Update & USA News

from Richard Speed

It seems that barely a week goes by these days without some new bit of TI interest appearing in my mailbox. In the last issue you may recall that I had a look at Stage 3A of PC99 and promised to keep you updated on what it can and can't do. Well, it seems everything is conspiring against me... but in a good way! Since then, I've joined a TI mailing list, received some exciting news about a US TI show next year and had a load of software updates from CaDD.

So where shall I start?

Since I was looking at PC99 last time around, I'll do the same this time. A month or so ago CaDD wrote to me informing me that Stage 4 of PC99 had been released. You may remember I touched on this briefly in the last issue during my on-line interview with Mike Wright. The asking price for the upgrade was \$9.94 (a play on TI99/4) including airmail postage. Naturally I sent off for this immediately (I sent a \$20 dollar note, not really advisable, but I couldn't be bothered to get a money order for so small an amount. I suggested that the guys at CaDD had a beer on me.)

Again, the service amazed me as by the next week I had a copy of Stage 4 of PC99 in my sweaty mits. I opted to do a completely new installation (regrettably the install 'program' is still the same and a little clunky to use) and left my existing Stage 3 installation safely alone.

So what's new? The major enhancement appears to be the inclusion of the Myarc disk controller which, as well as providing more disk drives of greater capacity, also adds some useful commands to Basic (I'd be lost without CALL DIR.) I'll write more about life with Stage 4 in the next issue; I'm still only just getting to grips with it (having spent a good three months working on-site.)

Email is a truly wondrous thing. Computer may talk unto computer. Reading our esteemed General Secretary's comments about email in the last issue reminded me about the TI mailing list, although I must first strongly advise against sending an email to every user in the world! For every 99er you hit, you'd seriously irritate a million other people! Anyhow, getting a list of every email account in the world is nigh on impossible.

But back to email and the TI list. As most of you know, email (in this case, Internet email) allows you to send a message from one computer to another (much like a letter.) For most of us this involves connecting to an Internet service provider and having them send the message for you, unless you have a direct link (like the

company I work for) in which case you can send the message direct. I shan't go into all the technical aspects of how this all works (unless somebody asks me) – but suffice it say there's no reason why a TI user cannot send email via the Internet provided you choose the right Internet service provider.

Of course, this is all leading somewhere and that is the TI list. The TI list is a kind of mail-shot system. What happens is this – you register on it and it begins sending you email. This email is from people on the register. You can reply to these emails or send messages to the group as a whole and so have email discussions with other Tiers around the world (including such luminaries as Charles Good and others.)

So, how do you do it? Well, first you need an Internet email account. Obviously it's easier with a PC, but there's absolutely no reason why you couldn't do this with a TI. A modem (or some other method of talking to your provider – I use ISDN) is also needed.

If there's enough demand, I'll do some research into which providers offer a service suitable for TIs. In the meantime, those of you who have access to email need to do this if you want to join the TI list:

(This text was lifted off the web)

There is a list server for the TI community. A list server is an e-mailaddress that you use to send messages to everyone who has subscribed to that server. The listserver will be administered by Tom Wills and will be an extension of the SouthWest Ninety Niners User Group of Tucson, AZ.

To subscribe to the TI List Server, send an email to the following address:

"majordomo@theriver.com" (without the quotes)

Leave the subject line BLANK and include the following as the message BODY:

"subscribe ti99" (without the quotes of course)

Please remember not to include your signature file in the message body.

After you are approved, just send your message to ti99@theriver.com. Everyone who has subscribed to the list server will receive you email. Also, to find out who is on the server, send a message to Majordomo@theriver.com (with a BLANK subject line) and "who ti99" (without the quotes) in the main body. Again, please do not include your

signature file in the message.

So there you have it. I get about 20 messages a day, all of which are interesting reading. And of course you can post your own questions. As I write I've asked if anyone has any circuit diagrams for the Geneve and Myarc HFDC, and sure enough there's been a reply and an offer to send them to me. That should cheer up Ross no end!

Some of you may remember in the dim and distant past a version of Extended Basic from Apesoft that added a whole bunch of new features to the standard Extended Basic command set. The most notable feature (in my opinion) was the bitmap facilities, which allowed plotting in a specified window on the screen. This has now been placed in the public domain and will soon be available for PC99 users, complete with full manual in Acrobat format. I can confirm that this exists since I've been playing with a beta (pre-release) version for a few months and am very impressed! I'll give you a full review next issue.

More stuff for PC99 users; the entire Tigercub software collection has been converted by Charles Good into PC99 disk files. That's almost 600 disks! I believe that we already have the Tigercub collection in TI format in the group library, if someone can suggest a way of doing it; I'd be happy to add the PC99 version. The only problem is that even in compressed form the library takes up almost 40 megabytes, which would take several hours to transfer via telephone. In the meantime, I'll upload the catalogues to the BBS and if anyone has a request for one or more of the PC99 Tigercub disk files, just give me a call and I'll put them into the PC99 area of the BBS.

More interesting news for TI owners the also have PC99. Apparently CaDD software are planning to write some add-on software for PC99 that will take an Extended Basic program and compile it into proper 9900 assembly code. This assembly program could then be linked and compiled and either run in PC99 or transferred back to the TI. Obviously it's still in the planning stages (and CaDD are, as usual, at great pains to point out that there is no guarantee that anything will ever be produced, and certainly aren't accepting any money until the product is released) – but here's what's been specced so far:

“The Basic Compiler for PC99 is a project that has been specced and is now under way. The basic tools (lex and yacc) for the project have also been acquired. The design goals are very simple:

Take any Basic or Extended Basic program and convert it to 9900 assembly code that can be linked with supplied libraries to produce an assembly version of the Basic

program. [The "any" implies that there are no restrictions on the way the Basic program is written. For example, you will not have to declare variables at the front of the program. In addition, all Basic statements, even things like DEFs are allowed.]

Generate a "library" of functions that can be called from the compiled code. Some of these will be TI ROM routines, such as FADD (add two floating point numbers), etc. For the first pass, some may also be GPL routines.

A scheme has been worked out for indexing and storing simple and array numeric variables using the TI 8-byte floating point system. Similarly for strings, but a convenient garbage collect for strings has not been written yet.

The Basic Compiler (early name BC99) is a PC program that is designed to:

Take a Basic or XB program from a PC99 disk file using dskout.

Compile it to produce 9900-assembly code.

Import the file into a PC99 disk using dskin.

Run the TI assembler to produce an E/A3 file, which can be converted to an E/A5.

The initial spec will allow the generated program to run on any 4A. However, it seems highly desirable for extra memory. This would first be in the Super Space area, and later using AMS memory. The current development version of PC99 now correctly emulates up to 1Mb of AMS memory. Finally, since it is trivial to exchange disks (up to DSDD) between PC99 and a 4A, the compiled programs can easily be transferred and distributed to standard 4A users.

The cost is projected at \$47. It is a standalone product that is not part of PC99, and does not require PC99. However, it is designed to be used with PC99. One of the prime reasons for developing PC99 was to have a Basic compiler. However, like all the products we develop, there is no time commitment or even tentative delivery date. We will only announce the product if and when it is ready to ship."

Interesting stuff, no? Obviously, we'll have to wait and see...

And now a PC99 bug that probably affects nobody but me. If you're running PC99 under Windows NT 4 (Microsoft's proper operating system) and your PC has only one RS232 port, be sure to turn off the second TI RS232 port in the PC99 configuration before running the main emulator, otherwise you'll be rewarded by what is known in NT circles

PC99 - Update & USA News

as the 'blue screen of death.'

Next up, some interesting news about the upcoming Fest West 98. Here's the lowdown:

The Texas Instruments manufacturing facility in the [HYPERLINK "http://interoz.com/lubbock/lubsubd.htm"](http://interoz.com/lubbock/lubsubd.htm) City of Lubbock, Texas will be the one of the sites for the 1998 Fest West - Lubbock. It was at this very same facility that the TI-99/4A computer was built.

Fest West '98 will be hosted for the fourth time by the SouthWest Ninety Niners User Group of Tucson, Arizona. The date for the Lubbock TI Fest West '98 will be February 14, 1998.

The schedule of events is as follows:

Friday, 2/13/98 - Pre-registration and vendor setup.

Saturday, 2/14/98 - Fest West '98 - Lubbock

A TI Experience at the TI Facility from 8:00 till noon.

Vendors activities from 1:00 till 6:00.

Hospitality room from 1:00 till 10:00.

Sunday, 2/15/98 - Vendor tear down time in the morning. Hospitality room open from about 9:00 to ??

On Saturday, September 27, the SW99UG received a letter from Bill Cosby and Kaufman-Astoria Studios. The letter was short and simple. It went like so: "On the behalf of Mr. Cosby, I would like to thank you for your invitation. Unfortunately, due to prior commitments, he will be unable to fulfill your request. Mr. Cosby asked that I pass on his sincere apologies and thank you for your consideration."

The letter was signed by Asata Malloy, Assistant to Mr. Cosby. During the meeting with Texas Instruments officials, the request that Bill Cosby, former spokesman for the TI99/4A be invited to Fest West '98 - Lubbock, was passed on. Apparently Texas Instruments, true to their word, did try to invite Bill Cosby to FW98. This shows that Texas Instruments are committed to a really successful Fest West '98 - Lubbock. Hopefully we'll do better with the other requests.

20th September 1997 - The raffle for Fest West '98 - Lubbock is coming together. Currently the prizes include a Super AMS card from the SouthWest Ninety Niners User Group, a GramCracker, and a Full copy of PC99 from CaDD. Ticket will be \$1.00 each. As tickets are sold, more prizes will be added. There will be one free admission with the purchase of 10 raffle tickets.

Currently, three vendors have indicated they want tables. As soon as the registration fee is received, their names will be included in the advertising for Fest West '98 - Lubbock. Prices for the tables are \$25.00 each. If the fee is sent in before January 1, 1998, the fee for the tables will be \$15.00 each. When sending in their registration fee, vendors should also note any special needs, such as electricity. The fee for the vendors includes two free admissions.

On Friday, August 29, 1997, Tom Wills, Mike Doane, and Ed McCullough met with officials at Texas Instruments and the hotels to finalize plans for Fest West '98 to be held in Lubbock on February 14, 1998. Texas Instruments is going all out for FW98. However, due to the fact the site is an operating production facility, there are certain restrictions we will have to live with. Some of those restrictions are:

No sales of any sort will be permitted on TI grounds. In other words, no vendor activities. Also, admission will be free to this phase of FW98.

Due to security policies, normal for this type of facility, people will not be able to come and go at will. Everyone will have to register in the TI lobby between 8:00 AM and 8:30 AM. An information card and a liability waiver will have to be filled out for each and every person. The activities will end at noon.

TI personnel will be providing tours of the TI facility to those who are interested. No cameras will be allowed in the facility during any of the activities. Arrangements are being made to take "official" pictures, which can be made available to those in attendance.

Except for the tours, those attending will have to remain in the area designated for the activities.

The schedule is projected to include the presentation at TI in the morning, which will include tours, speakers, and a mini museum. Speakers will include people who were instrumental in the development of the TI-99/4A. TI officials are contacting members of the development team to have them present at the activities. As speakers are confirmed, the names will be announced.

In the afternoon, from 1:00 PM until about 6:00 PM, the vendor portion of FW98 will take place at the Sheraton Four Points Hotel. As part of FW98, there will also be a hospitality room set up elsewhere in the hotel which will be open until around 10:00 PM. While the KoKo Inn is an adequate facility, it was felt that the extra cost for staying at the Four Points was well worth the cost. The KoKo Inn is along the lines of an older Super 8 or Motel 6. The amenities at the Four Points (such as an iron and ironing board, hair dryer, in room coffee maker, etc.), location to the TI facility, and many other things too numerous to mention, make the Four Points the place to be.

Those staying at the KoKo Inn will need to make their own arrangements to get to the Four Points Hotel.

On the 17th of August 1997 three members of the Fest West '98 Committee will be meeting with officials of Texas Instruments, Hotel management, and the Lubbock Convention and Visitors Bureau on August 29, 1997. Plans are to finalize all plans for the 1998 Lubbock TI Fest West. The three members of the FW98 committee going to Lubbock include Tom Wills, Mike Doane, and Ed McCullough. One point that is to be worked on is the insistence from Texas Instruments that no sales take place at their facility. If TI maintains that stance, then the actual Fest West '98 will probably take place at the Sheraton Four Points Hotel with a TI Timeline/Museum being set up at the TI facility. A report on the meeting with TI Officials will be posted as soon after the meeting as possible.

All the above information was pulled off the Southwest 99ers Fest West 98 web page which, for those of you with internet access, can be found at [HYPERLINK http://personal.riverusers.com/~twills/fw98.html](http://personal.riverusers.com/~twills/fw98.html) <http://personal.riverusers.com/~twills/fw98.html>

If any members are interested in going, then give me a shout and I'll get some more details for you or, alternatively, email Tom Wills at [HYPERLINK mailto:Twills@TheRiver.Com](mailto:Twills@TheRiver.Com) Twills@TheRiver.Com.

Gosh, I can see I've waffled on for far too long again... next time there'll be some Extended Basic programs for you to play with!

Until next time - keeping TI99ing...

THE MOBB - BBS

+44 (0) 1623- 491282

**The TI-99/4A User Group UK Official BBS
On-line 24 hours a day !**