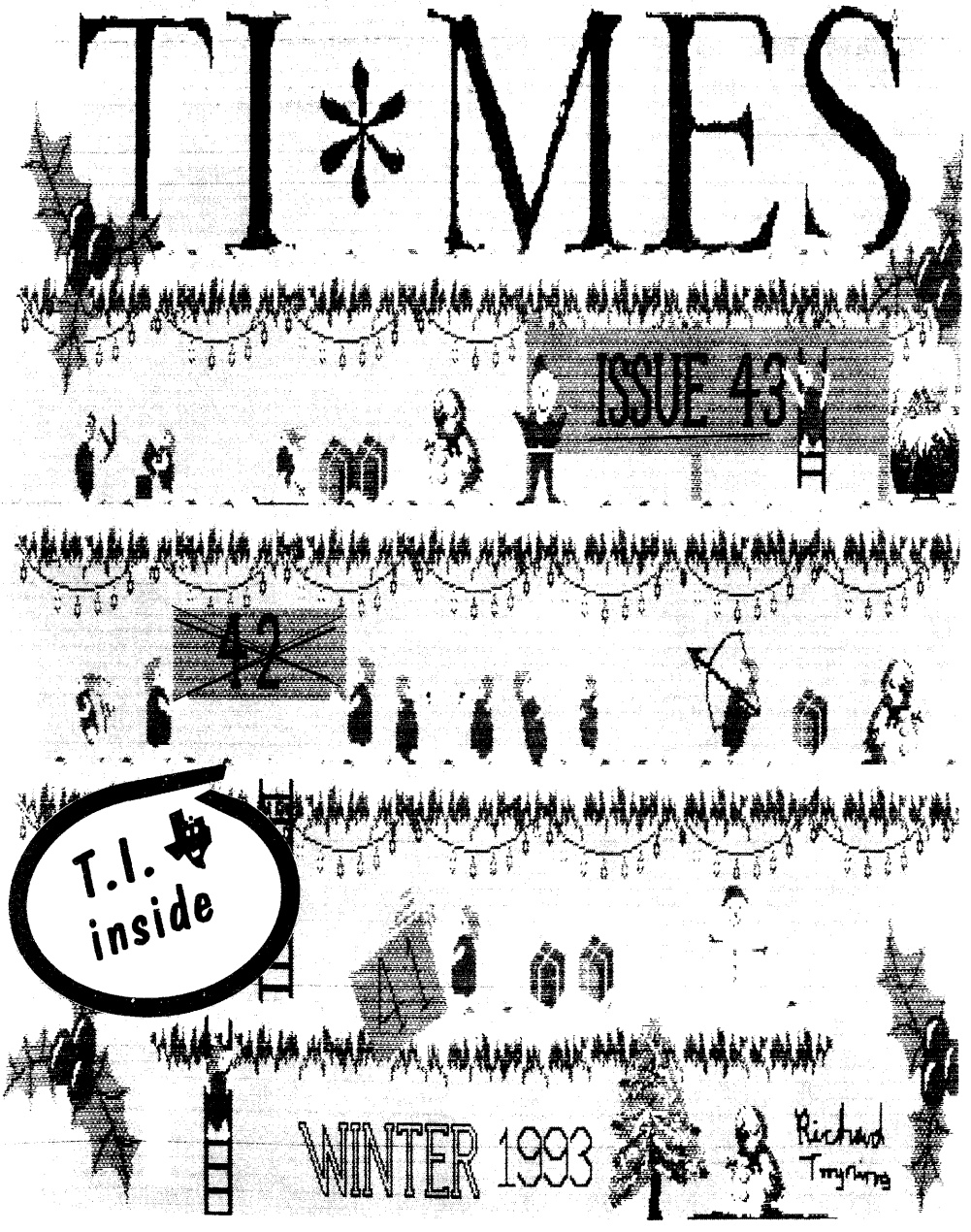
Happy New Decade from T.I. User Group U.K.



TI-99/4A User's Group (U.K.) Head Honcho's!!!

CHAIRMAN:
Trevor Stevens. Tel: 0623 793077
249 Southwell Road East, Rainworth, Notts. NG21 0BN

VICE CHAIRMAN, CASSETTE LIBRARIAN, PROGRAMMING:
Mark Wills. Tel: 0743 350588
12 "Rosehill", Betton Street, Shrewsbury, Shropshire, SY3 7YN

GENERAL SECRETARY, AND ALSO PROGRAMMING!
Richard Twyning. Tel: 0623 27670
24 Peel Road, Mansfield, Notts. NG19 6HB

MEMBERSHIP SECRETARY, BACK ISSUES:
Alasdair Bryce. Tel: 0389 65903
51 Dumbuie Ave., Silverton, Dumbarton, Scotland. G82 2JH

TREASURER:
Alan Rutherford. Tel: 0625 524642
13 The Circuit, Wilmslow, Cheshire. SK9 6DA

TI*MES EDITOR, and the odd bit of hardware!
Gary Smith. Tel: 0636 706767
55 Boundary Road, Newark, Notts. NG24 4AJ

HARDWARE:
Mike Goddard. Tel: 0978 843547
"Sarnia", Cemetary Road, Rhos, Wrexham, Clwyd. LL14 2BY

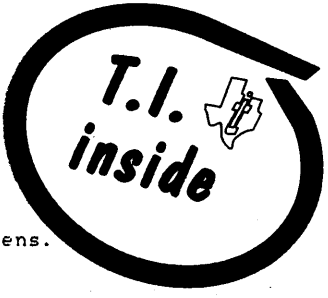
DISK LIBRARIAN, JOURNAL EXCHANGE:
Stephen Shaw.
10 Alstone Road, Stockport, Cheshire. SK4 5AH

PUBLICATIONS:
Mike Curtis. Tel: 0209 219051
21 Treliske Road, Roseland Gardens, Redruth, Cornwall. TR15 1QE

MODULE LIBRARIAN:
Francesco Lama. Tel: 0865 721582
14 Granville Court, Cheney Lane, Oxford. OX3 0HJ

Disclaimer

All views by contributors to this magazine are strictly their own, and do not represent those of the committee. Contrary opinions are very welcome. Errors will be corrected upon request.



CONTENTS

FROM THE CHAIRMANS CHAIR

T. STEVENS c 1993

Well its Christmas, it does'nt seem five minutes since it was Christmas 1992. Thinking on how time flies its its now TEN YEARS since we were left by our creator TI. I can't believe how long I have had my machine. I always look upon this machine as a sort of pandoras box. I always find out something new about it even now. Did you know that the 9900 chip and VDP are able to display 16 colours with 16 shades of that colour. This gives you 256 colours. This was never implimented as it was thought at the time of the machines release that so many colours would not be required. So when you think on how IBM (I'nt it a Bloody Mess) have been going on about 256 color cards for only 6 years or so, we had the concept before any one. The other thing I found out the other night, was that the GPL rom on board our machine has a full Language interpeter built in. It first of all looks at the English text then switches the text to the nominated language. The world of GPL has now opened up to us as one of our members has the full source to GPL. Did you know it was written in Fortoran and 9900 Assembly?

Last quarter I strayed from the path of Sprite Programing. As a result I had a few of our members shouting MORE!!!. I have also had a program written using some of the advanced routines. Mr Murphy sent it to me on disk and he seems to have spent some time on it. Sadly I have no space to print the program, If you wish I can send you a copy so you can view for your self. (Postage return please. I have been working on a program myself from scratch. This is a shooting gallery where you take pot shots at the various moving targets. I had hoped to get it out for this issue but old man time beat me.

In issue 41 (The real 41 Summer 93) we looked at a bit of Multitasking, and started to put togetner Sprite an Joystick. We had a look at a simple pick-up routine. What we are going to do now is to generate a screen with randomly placed alphabet characters on it. The program will then allow you to move you sprite around with your joystick, to pick up one letter. This letter will then be displayed in the top left hand corner of you screen. You then have to place this letter in the CORRECT alphabetical order in the boxes at the top of the screen. Quite a task this one but still very fast.

```
100 CALL CLEAR :: CALL COLOR
(2,7,7):: CALL SCREEN(11)::
CALL CHAR(33,"8080808080808080
FF1C5C487F193C26620747E2FFEF
0E1A33")
110 CALL HCHAR(24,1,40,64)::
 CALL VCHAR(1,31,40,96):: CA
LL HCHAR(2,5,33,26):: CALL S
PEEK(-31808,X,Y):: CALL HCHA
R(INT(X/13)+4,INT(Y/10)+4,C)
:: NEXT C :: NEXT R :: R,C=3
```

```
130 CALL JOYST(1,X,Y):: X=SG
N(X):: Y=-SGN(Y):: CALL GCHA
R(R+Y,X+C,CH):: IF CH=40 THE
N CALL SOUND(-60,110,9):: GO
TO 130 ELSE C=C+X :: R=R+Y
140 CALL LOCATE(#1,R*8-7,C*8
-7):: IF CH=32 THEN 130
150 IF H=0 AND R>2 THEN CALL
 SOUND(-90,440,9):: CALL PAT
TERN(#1,35):: H=CH :: CALL H
CHAR(R,C,32):: CALL HCHAR(2,
3,CH):: GOTO 130
160 IF H AND R=2 AND C=H-60
THEN CALL SOUND(-90,660,9)::
 CALL HCHAR(R,C,H):: CALL PA
TTERN(#1,34):: H=0 :: CALL H
CHAR(2,3,32):: GOTO 130
170 IF H THEN CALL SOUND(-95
,-3,9):: GOTO 130 ELSE 130
```

As with all programming the more complicated you have something the more code you get. The object of all programming is to keep thing as short as possable. This makes things run faster.

Now for the explaination of our program.

Line 100 is really self explainatory. It clears the screen and then sets the colour sets to Dark red foreground on the same background. This makes all characters from 40 to 47 dark red blocks. The screen color is set to dark yellow. The Chars 33, 34 and 35 are defined as 33=BOXES 34=person shape 35=person with object.

Line 110 is the screen set up. This places a wall around the edge of the screen and boxes at the top using the HCHAR and VCHAR statements. Then Sprite #1 is placed on the screen in a stationary position.

line 120 is the Random drop of the letters. RND is replaced by our call peek which places two random numbers into X and Y. We set up two loops one inside the other. C will equate to the letter value. R to the repeat cycle, which in this case is two. At the end of the line we make r and C equal to 3 which is the starting graphic Row and Column position of our sprite person. (Remember Sprite positions are more than graphic numbers.

130 is the work horse of the routine. This starts to move the person around the screen. This routine we have discussed before. but basically we look at the X and Y values. These equate to -4 or 4 and 0 now we come to a fancy bit. we now convert with SGN operator the values returned in X and Y. These will be converted into negative numbers. Since the Joystick returns a -4 when pushed up we change this by placing a minus sign before the function. This will now return 1 with the Y value as negative and -1 with positive,) still remains 0. With the Gchar we now find what character is under our sprite. If the value of 40 pops up we have hit the wall.We then jump back to 130 for a new joystick input else we add the value of X to C and the value of Y to R. This is how our sprite moves in graphic blocks with ease.

140 really is a mover routine. This turns graphic locations

back into sprite location values, so the sprite can be moved. We also check for the space (32) if that is so we jump back to the joystick routine at 130.

150 is also a test routine. It checks the variable CH, to see if it was not equal to 32. Since we already checked for the wall character (40) it must either equal an alphabet character or a box (65-90 or 33). The variable H will tell us which letter of the alphabet our sprite is holding. If H=0 then no letter. To stop our sprite picking up a box, we test it to see if it is not on row 2. So if it not holding a letter and is not on row 2 a beep sound is generated. The next thing is to change the sprite shape, to the shape of the man holding a object. This is done with Call Pattern. In this case char 35. Then we set H equal to the character value of the letter that is on the same screen location as our sprite. Next we place a space char(32) to delete this letter from the screen. Then we display the letter picked up, at the top left corner of our screen. We then jump back to the main routine 130 for joystick input.

160 is a function logic routine with tests. If our sprite was holding a letter or it was on row 2 the the test in line 150 would be false, so the jump would be down this line. So, when our sprite is holding a letter and it is on row 2 then when it is lined up with the box equal to the char held, we will put the letter down. Since 26 boxes on row 2 were placed starting at column 5 we will subtract 60 from the value of the letter our sprite is holding and test to see if we are at the proper column. The A can only be placed at Column 5 since A=char65 and 65-60=5. So our sprite moves to the right position it will drop the letter and a beep sound is generated. The letter is then placed in the box with the Hchar command. The Sprite is then changed to the man not holding anything. Next then letter in the indicator position is deleted with the space (32) again with the Hchar command. Then its back to 130

170. If the test in line 150 and 160 proved false then this line is activated. This means that our sprite is holding a letter but in the wrong box. We generate a noise sound (Type 3) and returns the program back to 130. We also issue an else statement at the end as this looks at the situation when our sprite is not holding a letter but is in row 2 it then puts the program back to 130 for the new joystick input.

As you can see this program has a very good basis for a letter game, or educational "find the letter" type game. You don't have to stick with letters. You could change the letters into shapes, or revalue and turn the letter into numbers. How ever with the last

you decide on.

This program is very simple and could be vamped up quite a bit. Try to add better reward sounds, improve on the graphics. You could also try and keep track of the letters that were placed in the proper box on the first try and display a percent score. (hint: Top value - actual value / 100) You might want to have the program check to see when the boxes are full, so the game can be restarted or....??, the rest is up to you and your imagination.

IS RND RANDOM?

Walter Allum

On page 61 of Issue 41, Jim Peterson said that there had been controversy for years about the randomness of RND. He had done a test but didn't know what to make of the result (of this, more below). In the two years or so of my membership, no ripples from this storm have disturbed my Essex backwater but I must presume that it's still going on since the rapporteur (sjs) chose to print the article a second time without comment.

If there is a problem, it seems to me that the Group could make a contribution particularly members without the add-ons needed for some projects but willing to provide that precious commodity--computer time. For, short of our technical wizards being able to peep into the ROMs, GROMs or wherever to identify the algorithms used and their parameters, observing frequencies of events in lengthy tests is the only way forward - and, even knowing the algorithms, such tests are likely to be needed. We are talking of tens of thousands, if not millions, of RND samples. The requisite statistical assurance just doesn't come with less.

Now, here's where we would need to plan so as to avoid the annoyance of your seeing a column of smoke arising from the console ventilation grille, or of young Sharon's grotty hairdryer bringing out the supply earth trip, just as you were on sample 875391 of your million RND marathon. Perhaps I'm about to teach my grandmothers (figurative, of course) to suck eggs but I haven't seen the matter dealt with elsewhere. Sadly, what I can suggest will not be available to those without XB. This is because of a major difference between the console-only RND and that under XB (detail later). Under XB, each RND is the seed for the next. For example, initial seed 1 produces .42132486540519, then .47367534286618, then .35707275413897 However, if you just work from the screen, you will get .4213248654 and similarly shortened versions of the others. If you later enter just .4213248654 as seed, you will get .7905073588, then .4077304243 and you will think I'm wrong. You must enter the full 14 radix-10 digits. So, the idea is to break long runs into manageable fragments, passing the complete last RND of one to the next for use as seed. That way, we rejoin the same RND sequence as we left and do not hop to another, with unpredictable disturbance of the statistical analysis.

The easy way to make a satisfactory transfer is by writing the RND to, and recovering it from, disk or tape (internal format, of course). I have to confess that I overlooked this method in my initial work and, instead, provided myself with an XB routine to display the full 14 digits on screen so that I could note them down and re-enter them via the keyboard. However, the routine has been very useful in monitoring program action and in my attempts to deduce the algorithm from test results. I am a bit surprised that it took me about 70 simple-Basic statements to provide. Our top-notchers could probably shorten it considerably and, where Assembler was available, speed everything up.

I now make various remarks that help to set the scene but do not form a connected account.

1. We might have expected the same algorithm (possibly with different parameters) to be used both with simple Basic and XB. We might also have expected this to be of the well-known linear congruence type: $X(N+1)\equiv A*X(N) \pmod M$ where X and A are non-zero positives less than M, the latter being a large prime or power of 2. In fact, the truth cannot be as simple as this. Texas say that RND is greater than or equal to 0 but, passing

over whether zero X(N+1) could ever arise from non-zero X(n), it is clear that zero X(n) would result in all subsequent X being zero. Actually, with simple Basic, X=0 leads to .82457439470957 (at least, on my machine). With XB, X=0 produces .21132486540519. Maybe some kind of origin shift and/or rescaling has been applied in the algorithm.

2. Although testing by observing frequencies can proceed without knowing the algorithm, this is still needed for identifying the cycle length, i.e. the number of different pseudo-randoms delivered before the sequence repeats itself. The TI has the potential for very long cycles and one would hope that any tests that we are likely to be able to launch would not be falsified by inadvertantly "going round a second time".

3. Under XB, all seeds differing by a factor $10\wedge$(even no.) give the same result e.g. 13.1 and .131 both produce .76232486540519. Seeds differing by a factor $10\wedge$(odd no.) give different results e.g. 1.31 produces .96642486540519. This is an understandable consequence of the TI storing numbers as seven radix-100 digits. A change of one unit in the 14th radix-10 digit makes a difference if the seed is of order $10\wedge$(odd no.) e.g. seed .12345678912341 produces .24846676792480; changing the last digit to 2 produces .39236497213301. But no change occurs when the terminal 1 of .012345678912341 becomes 2; in both cases, one gets .90064923523633. This also is a consequence of the storage arrangement. Note that the Manual says nothing about the choice of seeds.

4. I have less experience with RND under simple Basic. The Manual says, in effect, that only the first two radix-10 digits count in the seed and that non-integers will be INTed. Actually, fractions are NOT made zero but the two-digit rule is right. For instance, seed .30660284679955 produces .50011497251527 (not the same as from seed 0) but seed .30 would have had the same result. Even sticking to two-digit seeds, the equivalence behaviour under multiplication by $10\wedge 2$ etc. seen for XB does not apply here. Seed .0012 gives .99576977161991 but seed .12 gives .72189862099268. Maybe, within the RND sequences, each RND is the seed for the next but we are blocked off from exploiting this, if true.

5. There are many procedures in the literature for testing the randomness of sequences. Some, no doubt, would demand more than we can manage. But, they should be reviewed for possible application. I mention: frequency of runs of the same number, poker test, coupon collector's test, runs-up-and-down test, autocorrelation and power spectral analysis.

6. It would be desirable to hear opinions about the accuracy targets for any investigation launched. Needs vary with applications. We are not drawing lucky numbers for a national lottery or running an elaborate simulation to establish the safety of some important installation. But, would you be satisfied if you wanted a random sequence of integers 0-9 and learned that your TI was working with respective probabilities e.g. .11 .08 .08 .12 .09 .13 .08 .10 .11 .10 ?

Turning now to practical results under XB. Jim Peterson's background theory was a bit "off" and there is an element of double counting in his program (e.g. triples are also counted as pairs). Putting this right, I make it that his 10000 RND made him 785 pairs, 76 triples and 7 fours. On the hypothesis of uniformly distributed RND, the expectations are 810,81 and 8. I have forgotten how to do multinomial confidence limits but, considering the pairs figure in isolation, I would have given the central 95% confidence band as 732 to 838, which comfortably encloses the ideal expectation. A 5000 RND run of my own, using a modified program, gave 375 pairs, 55 triples, 11 fours and 2 fives. In this case,the confidence band around the pairs result was 339 to 411, enclosing the ideal 405. Thus, although both runs produced

fewer pairs than expected, the evidence does not point firmly to a defect.

I have also done 1000 tests generating the numbers 0-9 and noting the frequency with which e.g.0 is followed by 0,1....9 and so on with each integer. Ideally, there should be 10 of each pairing. The frequencies actually observed ranged from 2 to 20. Such a scatter might well conjure suspicions of a crook algorithm among those not versed in statistics. Actually, the chi-square (81 d/f) was 72.3 which gives no support to fears. The number of appearances of 0,1....9 (expectation 100 each) were 93 125 88 100 110 86 105 99 88 106 , again perhaps worrying to some but achieving only a low degree (approx.0.1 level) of significance for departure from expectation. The total number of pairs, triples...etc. observed was 96 (expectation 90); the 95% confidence 78 to 114. I wonder how much of any controversy may have arisen from people judging by appearances rather analysis! The last thing to say now about this test is that I took the opportunity of checking the doing-by-installments idea. I first did 250 tests; then, using the same initial seed, two 125-RND runs. The results were the same in every detail. I then grafted on a further 750 RND.

To sum up: so far as the above exercises go, there is no clear evidence of non-randomness. But the sample sizes were pitifully small for this class of work. Before anybody gets stuck into more realistic tests, however, I suggest that someone with a broad knowledge of the TI world should outline for us just what charges people have been levelling against RND (and which one!). We could then aim at the right targets.

DISK LIBRARY UPDATE: FUNNELWEB V5 40 COLUMN EDITOR IS NOW AVAILABLE

We have received from Charles Good, the COMPLETE 40 column Funnelweb v5 text editor from Tony McGovern. SOME of the new features include:

--The ability to display on screen and print with almost any printer (using IBM mode, not Epson mode) directly from the editor all the high ASCII graphic shapes, math symbols, and foreign language characters. Just type PF (print file), and the graphics displayed on screen will print on your printer. The result is similar to the commercial software FORM SHOP, but you don't have to use the Funnelweb (TI Writer) formatter to print these graphics. Tony calls this feature ALL CHARS.

--For hard disk users: the ability within ShowDirectory to page back and forth through the entire tree structure of your hard disk directory. Moving the cursor next to the name of a sub directory will allow a display of that sub directory. You can also display the root directory of the currently displayed sub directory. This is also now available on a revised 80 column v5 Funnelweb editor available from us.

--A number of HELP screens can be displayed by pressing H from the command line.

--Foreign language (non English) command line text, commands, and character sets can be optionally selected at powerup or configured to automatically boot when the editor is powered up.

--Text scrolling and windowing up/down from the command line.

--ALL CHARS tab records with saved files are now compatible with other versions of TI Writer. This is true for both the 40 and (new) 80 column Funnelweb v5 editors.

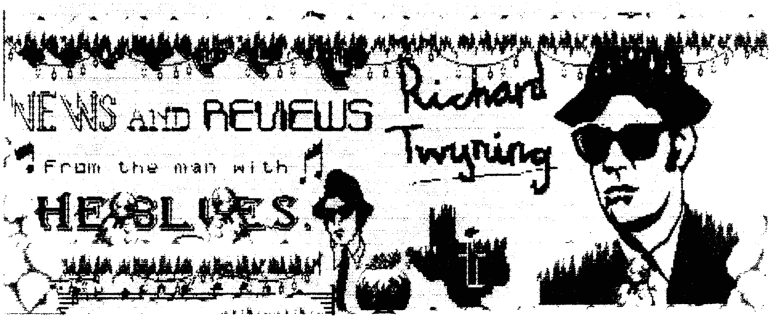
--Ability to freeze text below cursor and scroll text above it!

--Ability to save (but not load) ASCII and UNIX DF128 files.

--Ability to append to an existing file

--Ability to pick up a file name from a Show Directory without amending the SAVE filename.

What we have in hand now has a couple of problems which we hope will be fixed by the time the Sept newsletter is mailed. This text editor is really neat, with SIGNIFICANT new features not available on previous versions of Funnelweb.



~~Dear TI'ers,~~

Merry Christmas
TI'ers,

Welcome to ANOTHER **BIG** article.

I'm sorry about the front cover! I just had to do something witty after the mixup of issue numbers on the front cover of the Autumn edition!

I thought I would try and write a good article for the winter issue, just in case it throws it down with snow and everyone gets stranded.

By the time I've finished the article, if it turns out to be long enough (I'm sure it will), it should keep you occupied for quite some time. If you can't read it all at once, then you can save it and read it during the probable screening of "The Sound Of Music", or the "Snowman!"

If we have power cuts again, then you'll need to keep a candle or two, or gas lamp standing by, in case everything goes dark. Unfortunately, you won't be able to type our game in and play it, unless you've got a generator, or you've developed a portable 4A that runs off of car batteries!

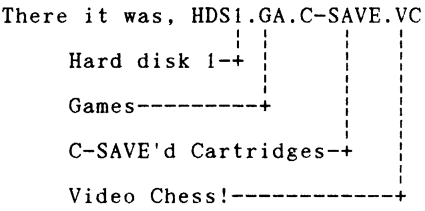
Well where shall I start? I've got alot of things to talk about in this article. Most of which were intended to be in my Autumn article, but it was a bit of a rush to complete it. I had two days over the deadline to finish the article, due to Gary's ear infection. I finished the article on the night that Gary travelled back down to Swindon to pick up the articles, that had been sent down there.

That was on a Wednesday night, and we arranged a get together at Trevor's on the Thursday night to edit TI*MES. While we were sorting out the magazine, we decided to put the Amiga to the test against the worlds' most efficient Chess program. We were slightly confused when loading up Video Chess, because the difficulty levels were Novice, Beginner, and Intermediate. We were sure that there should be an option for Advanced!

The version we used was a disk version that loads via Extended BASIC and then attaches itself to the TI Title Screen.

Then it throws you out to the title screen, and Video Chess appears as an option from the title screen! It's very clever how they've done it. Yet another secret uncovered from the silver thing with a keyboard that we call Pandora's Box.

When I got home, I tried to track down my GRAM kracker version of the cartridge. I couldn't find it on floppy, and my directory has become corrupt on my 40 Meg hard disk, so I had to power up one of my 80 Meg drives.



How's that for a filing system! It's a pity I haven't got my full hard disk system on line properly yet. I still haven't even been able to use floppies from the Hard Disk controller card with the GENEVE, but I think 1.5H version of MDOS might solve that problem (theoretically!).

Well, how did Video Chess do against Battle Chess? We put Battle Chess on Level 9, and played Video Chess on Intermediate. We also played aggressive, which meant Video Chess wasn't being as careful as it could have.

It could be classed as a draw, and then Video Chess started making some moves which appeared to be meaningless to us. This is probably because they were meaningless!

It was getting close to two hours. Video Chess gives you a proper clock, whereas Battle Chess doesn't. Therefore, Video Chess did some totally useless moves that wasted time. It wanted to catch Battle Chess out on the time rule that says you've got to make 40 moves in two hours!!! We didn't let the game play right to the end as it would have took too long.

Battle Chess had a total thinking time of about 2hrs 15mins, against Video Chess's time of about 40mins.

While we were playing though, Battle Chess kept trying to cheat. We set an initial thinking time of three minutes per move, which was what we had put into Video Chess, but it kept having a longer time to think. In the end, the only way round it was to set a new thinking time limit after every move!!

A few days later, I played my Amiga, with Battle Chess, against Video Chess. I played it on Normal, and had Battle Chess on Level 9 again. It was more or less a draw again, but the Amiga had a thinking time of over 3 hours, and Video Chess had just over half an hour! That includes limiting the Amiga's thinking time for every move to stop it cheating!!!

It proves that efficiency is better than speed when it comes to writing decent software.

I had to laugh at a statement made by our Software Support Manager, "Software is only as fast as the machine it is running on!!!!"

PC and Amiga programmers etc. don't know how to be efficient! Their assembly language is too complicated, so they get massive compilers and write it all in C. Then it turns out to be that big, that they need 40 Megs of hard disk and 2 Megs of RAM, to do a job that we could do with 48K RAM and two 180K disks!! And that's no exaggeration.

In a review in December's Personal Computer World about Word Perfect 6 for Windows, it says that Word Perfect 6 will struggle along with only 4 Megs of RAM if it has to, but it prefers 6Megs or more!

Windows is a good example of PC efficiency!!! At work, I've got a Windows BMP (bitmap) background picture of Robert Johnson that I scanned from a CD cover, in 256 colours.

Everyone says it slows down my applications. I can't see how it should, but it appears that Windows has to update the entire screen.
With TI software, if you were doing something in a window, you would only update what was happening in that window, but MessySoft Windows has to re-update the entire screen, which includes the backdrop picture!

The most amusing thing about PC's is the amount of [redacted] you have to put in their startup files because of their [redacted] architecture.

You will note that I said "startup files!", rather than file!

You only need to setup your machine once when you startup, so what do you want two startup files for?!?!?!

On the GENEVE, you have the AUTOEXEC file, which (surprisingly enough) Auto-Execute's when it's finished loading MDOS!
The AUTOEXEC file is just a D/V 80 file that contains the commands you want executing when your machine starts.

A typical AUTOEXEC file on the GENEVE contains the following:

```
TIMODE
ECHO OFF
RAMDISK 500
SPOOL 200
LASTDRIVE=K
ASSIGN A=DSK1:
```

```
ASSIGN B=DSK2:
ASSIGN C=DSK3:
ASSIGN D=DSK4:
ASSIGN E=DSK5:
ASSIGN F=DSK6:
ASSIGN G=HDS1:
ASSIGN H=HDS2:
ASSIGN I=HDS3:
ECHO
ECHO Welcome to the MYARC 9640 Mini-Computer.
ECHO
ECHO ON
```

And that's it! When you look at the file, it's simple to see what's happening. The TIMODE command causes the GENEVE to reserve the memory that you will need for running TI GPL emulation, and it sets up the Gate Array to emulate GROM and GRAM, and also to control the 99/4A memory map. This means you can run GPL at any time, which will take you properly into 4A mode. You can even add it to the end of your AUTOEXEC file if you want, or you could have a special disk that's setup to start the machine and take you right into Extended BASIC or Editor/Assembler.

ECHO OFF tells MDOS to stop sending the commands to the screen as it executes them. It stops the screen from getting cluttered up.

RAMDISK 500 means, give me a 500K RAM Disk which is accessed as DSK5.

SPOOL 200 means, give me a 200K print spooler which allows you to be printing something out while you're doing something else. This is amazing when you're working late trying to finish a report. When you've got a print spooler you don't have to wait for your text to finish printing before you can load TI- Artist and start on some diagrams! I very rarely have a spooler this large unless I'm doing ALOT of printing, such as the odd 300page manual, because I've got a 256K compression buffer connected in between the GENEVE and the printer, and it compresses text to 15% of its original size, so you can fit more in the buffer! It compresses graphics to 85% of its original.

Gary was thinking about building a 64K buffer once. We might start thinking about one after the mother of all 80-column cards is complete, and we get some sort of cheap alternative expansion box produced.

LASTDRIVE=K means that any disk drive cannot be referred to by a letter higher than K. On PC's, all drives are referred to all the time by letters. Normally A and B for two floppy drives, and C for hard disk. MDOS on the GENEVE does the same thing, but because we've got superior hardware control, we have true device names DSK1, DSK2, etc.

The ASSIGN command tells MDOS which letter refers to which disk device name. This includes HDS1 hard disks as well, and will also include SCSI devices when Ron Walters' and Bud Mills' card is available (hopefully more later on this!).

The ECHO command is also usable as a print command. ECHO OFF stops the system displaying commands to the screen, and ECHO ON starts it displaying again, but if you just type any other string after the ECHO command, it will be printed to the screen. The actual ECHO command itself will not be displayed though, if ECHO is turned off.

On the PC, things are crazy! You need TWO startup files! The first one is **autoexec.bat** which is sensible, but then you've got a **config.sys** file too!

Computers are supposed to be getting easier to use aren't they?!?!?!?! Well, here's the **autoexec.bat** and **config.sys** files from one of our office PC's startup disk at work:

autoexec.bat

```
@echo off
prompt $p$g
PATH C:\;C:\DOS;C:\NET;;C:\SBDISK\SPUT
set nwlanguage=ENGLISH
SET BLASTER=A220 I7 D1 T4
SET SOUND=C:\SBPRO
C:\SBPRO\SBP-SET /M:8 /VOC:10 /CD:8 /FM:1
cd \nwclient
lh lsl
lh ne2000
lh ipxodi
vlm /Mx
echo.
CD\
LH /L:1,15904 keyb uk
LH /L:1,6400 doskey
LH /L:1,13984 SHARE
lh /l:1,26720 fastx c:512 s t:5 swb:7
lmouse
VER
f:
LOGIN
```

config.sys

```
DEVICE=C:\DOS\himem.sys
DEVICE=C:\DOS\EMM386.EXE noems
buffers=8,0
files=30
dos=UMB
lastdrive=z
fcbs=1,0
dos=HIGH
DEVICEHIGH /L:1,3872 =C:\DOS\SUPERII.SYS /ERGO480 /35K /45M /BORD 00


DEVICEHIGH /L:1,44304 =C:\DOS\DBLSPACE.SYS /MOVE
SHELL=C:\COMMAND.COM /E:1024 /P
stacks 0,0
```

Your guess what half of that [redacted] does will be as good as anybodies, because to me it means about as much as the lyrics to the current number one dance record!!

It's possible to work out what a few things are meant to be doing, but 99% of it is just mindless, and needless[redacted]!

Each program might also have its own setup file or files, that means it won't run unless it's setup correctly! Here's the setup for Lemmings 2!:

```
[12prefs]
Bios = 666
Boot = 23003
Music = 0
SoundFX = 1
Mono = 0
SoundName = "music\beep.bin"
IntroName = "music\introbp.bin"
Input = 0
Sensitivity = 1
```

The Bios = 666 sounds a bit OMENous!!!

If you're not confused by those, then there are still the **system.ini** and the **win.ini** files if you want to run Windows! They're too long to include here!

It's also sad that when you get different pieces of software, you have to fart about with your **config.sys** file and try and give them the maximum amount of memory to run in!

At work (where I'm currently typing this) when I do a MEM command, it tells me that the largest executable program size is 556K, but the machine's got 4 Megs of RAM!

It also tells me that it's got 155K of upper memory which is all used, and 384K of Adapter RAM/ROM which is all used.

It also looks like Windows has chewed up 2661K of Extended memory out of an original 2917K, and left only 256K!!!

This leads me to re-quote and re-word an intel advert:

> "intel, the standard to which thousands of software packages, have been limited!!!"

The following two statements also spring to mind. The second one is a re-wording of a gospel song by Washington Phillips from the late 20's!

> "RISC your machine. Don't risk your data."

or

> "You may have been to college, or you may have been to school, but if you ain't got a Texas, you're an educated fool."

Here's a couple more funny stories from the office. We've

got a bloke in the office who deals with accounting packages and stock control packages.

These are two nice typical PC programs that between them, eat 100Megs of hard disk, if they are fully installed. Also if you upgrade to 6, you have to have update disks for every version in between. The files are totally incompatible between every version! If you have version 2, you have to use the converter to go from 2 to 3, and then the converter to go from 3 to 4, then 4 to 5, etc.!!!

Different versions of Wordperfect are also incompatible. You can't save text from version 6 and expect it to load 100% on version 5.2!!

"Word, not very perfect" more like!!!

Back to the subject!

Nothing seems to be written in Assembler on PC's. The 4A can still kill PC's for efficiency. All of our software is mostly assembler, and when we do use BASIC, it's much more efficient than PC's because we've got a decent filing system to support it. If we use something like C99, it's also easy to insert assembler into it, so the more processor intensive routines can all be done easily in assembler which would save time and memory.

I think we are still equivalent to PC's in speed. They go about things the long way around all the time, and they do it in some sad little language rather than assembler, and they've got to constantly swap registers in and out of Stack!!!

And the list goes on! And on! And on, including our support manager giving up on Wordperfect 6.0 for DOS because it was running too slow on a 386 PC with 200Megs of hard disk! The same PC that's got a faulty RS232 board, and it was made in 1990, and the purchase date in the back of my CorComp RS232 card manual, says 6th of September 1984, from Parco Electrics, when they were still at 4 Dorset Place!

The final funny story is about the software package that we use, It's for Time & Attendance and Job Costing Systems. The programmer came over to England to see us, and to deal with the 6 pages of bugs that we had collected for him to hopefully sort out.

We were on version 4.11a, which didn't allow the use of extended memory, so if you hadn't got 570K of sad Messy DOS base memory free, !

He said he'd sorted that out in ver . It will use expanded or extended memory, but unless you've got 4Megs

of it, or at least 2Megs, it will run slower, because it swaps the memory in and out of base memory as it wants to use it! It can't use the extended memory directly. Unlike my idea of the extended cartridge port RAM for the 4A. 8K of that is part of the memory map, so we can use it directly after switching it into address >6000.

"I blame it on intel!"

That's enough of showing up the complete inefficiency of the worlds PC's.

Back to our dreamworld of no stack, and no viruses...

It leads me to come up with a good advertising slogan.

"The Texas Instruments TI-99/4A Home Computer. It can't run Windows, and it can't run Word Perfect;
But these are only two of its advantages."

In my last article I didn't have time to report on our visit to Derek Hayward's house to see his system. Gary, Trevor, and I all travelled to Derek's place in Dore, Sheffield, which only took about 30 or 40 minutes.

We started the evening with Derek actually taking his system apart to show us how it all fitted together. It's incredible how it seems that the console PCB was intended to be fitted onto the back of the Expansion Box. It fits perfectly. Derek's built a 2 inch wide frame at the back that holds the TI PCB, and then, in between that and the Expansion Box, is a Zeno Board, and a Rave 99 Keyboard Adapter.

The front of his expansion box is the best thing I have seen in a while though. It's pure evil! Switches for the zeno board, and some mean LED's telling you what you are currently switched on to, such as E/A, Extended BASIC, or the cartridge port.

"If his console is screwed to the back of his box how does he use cartridges?"

He's extended the cartridge port through the expansion box and has got a slot cut into the box on the front through either expansion slot two or three.

This is the only irreversable thing he's done. All of the hardware will go back to its original state. Even the guitar strap!

The clock on the Zeno board is amazing. With the information supplied by Gary, Derek's built a completely separate power supply board that handles the changeover from full power to battery power!

The way that Derek's fixed his 4A's PCB onto the back of the box is probably the best way of doing things when you've got a Rave 99 keyboard adapter. You still need to have the console plugged into the expansion box with the guitar strap. You remove

the 4A keyboard completely, and the Rave adapter plugs into the 4A PCB where the keyboard would normally plug in. You still have the 4A console cluttering up your desk space, and they give you a cover which covers the console's keyboard hole, which now contains the Rave 99 adapter. The IBM style keyboard, then plugs into the Rave adapter in the console.

Because Derek's done away with his console, and removed the PCB, he's saved alot of space, and it means that everything's just a little bit more efficient when it comes to linking things here and there.

When the 80 Column card is sorted out, he'll have no problem with getting the audio signal out to the monitor, because his video output from the box is very close to the video output on the console. Should we also need an interrupt line from the 9958 VDP to the 9900, then this will also only need a short piece of wire, because the 80 Column card will be sat more or less right next to the TMS9900.

Derek was quite impressed with the software that we demonstrated for him. The best thing he saw was The Missing Link with the incredible TML_DEMO program, which shows just about every capability that the Missing Link's got, which includes the text and graphics windowing, multiple sized fonts, turtle graphics, and the excellent sprite control which gives you the full 32 sprites from Extended BASIC. The only drawback to The Missing Link is that it needs 32K RAM, and disk to load it. I think it might be possible to transfer it to tape though, which means that it would work on the console only with our mini-32K expansion, when we get around to designing one!

And, still another possibility is to transfer it to an EPROM and put it on a CRU address and have it on a plug in card for the side of the console!

The number of possibilities with the 4A are endless. It's amazing how you can put assembly language subprograms on a CRU, and if GPL can't find them in your console, or cartridge, then it will do a search of the CRU addresses and look for it on each DSR EPROM!

Imagine an EPROM board that contains assembly language utilities that are contained in EPROM's that are each mapped to a different CRU address. You can add hundreds of commands and utilities to Extended BASIC without using up any XB memory!

Chew on that again IBM!

At the time of writing, unfortunately, there is a problem with the prototype board of the 80-column card that Gary is working on. He's using an 8K RAM chip in place of the DSR EPROM, and he's writing data to it, but it won't read back.

We had the same problem at Trevor's place when we first tried. We used the TI's most advanced, fully integrated, hardware test system (That's Mini-Memory to you!).

The Mini-Memory is amazing for testing hardware. TI (the almighty creators) foresaw this possibility and thought to themselves that they would include the function of being able to inspect and modify CRU. It's an incredible cartridge for testing DSR's.

Gary has completely built all of the CRU select for the board which turns on the status LED, and also simultaneously selects the particular EPROM. I think Gary has decoded the prototype at CRU >1600, so we did a CRU modification with mini-memory, and said we wanted to look at >1600.

Then I typed a 1 and the LED came on, because placing a 1 at >1600 means that the base address of that CRU is active. Then I did a modify on memory address >4000 (I think), which is where every DSR is decoded. I stuck some values in it and tried to read them back, but they came back empty.

Gary had a fiddle with a few wires, and he even put a wire directly onto the chip select and selected the RAM chip and I tried again, but, still nothing.

We thought we had narrowed it down to a buffer chip, but I think Gary has replaced it now, and there's still something hiding away somewhere that's causing the problem.

Mark phoned the other night (22-10-93) and he said he had called Gary the night before, and they talked about the board. He has invited Gary down to Eastcote to use their test equipment (including oscilloscope), and he also said that Gary was going to ask Trevor and I, and try and make a right old party of it! (More beverage and visual material - purely for research purposes!)

Also, we have had another idea for the 80-column card. It's going to be a half size card, so we thought we'd include a Speech adapter on it as well, so you can plug your speech synthesizer directly into the 80-column card, and have it in the expansion box!

Mark and I got chatting about C99, and he said he'd not done it for ages (or C99 programming!!!) since his machine had been down. He wanted to get back into it again rather quickly, and also, he wanted to try and re-write the C99 graphics routines.

To do this he would need to grab a parameter from C99, into a TMS9900 routine, which is not as difficult as it sounds at all!

"Oh Yeah!", the non-assembly language programmers are screaming!!!

The thing about C99 is that it uses stack.

Stack!!!!!!

Yes!!! Stack is very good for passing parameters. It's only crap when you want to use it as your machine's basis for subroutine, and interrupt routine, RETURN control.

On every other machine, when you branch to a subroutine, the current Program Counter value (that points to where you are currently at in your program) must be saved. So what do they do?

They push it on stack, along with all of your register values! (Well, you want to be able to use all of your registers in your subroutine don't you!!!)

When the subroutine's finished, it pulls all of the registers, and the program counter back from stack and then carries on executing.

What we do is keep our registers in memory and point to them with a Workspace Pointer that is part of the CPU. The current location of our running program is pointed to by the Program Counter, which is also part of the CPU.

When the 9900 reaches a Bullwhip (BLWP - Branch and Load Workspace Pointer) it looks at the address that is contained in the operand of the BLWP instruction.

This operand points to a location that contains two more 16-bit words which are the new Workspace location, and the new Program Counter location.

When the 9900 transfers control to the new routine, it puts the current contents of the Workspace pointer into the address that will be Register 13 of the new Workspace, the current Program Counter into Register 14, and the contents of the Status Register into R15. Then the Workspace Pointer, and Program Counter values of the new routine are loaded into the actual Workspace Pointer and Program Counter of the 9900 itself. This means that the new routine is looking at a totally new set of 13 USABLE registers that are totally separate from the registers of the previous routine.

You can still get parameters from the previous routine by fetching them from Workspace Registers of the calling routine, by referencing it from the base address of its Workspace, which is now contained in R13 of your new routine.
You could also pass parameters to the new routine in DATA definitions by following the calling BLWP with DATA directives. You can then use the Program Counter value in R14 as a base reference to the data.

When you want to write machine code routines that link to C99, parameter passing is slightly more complicated because C doesn't handle it quite the way we normally would.

It appears to me after studying some C99 which has been compiled into 9900, and some 9900 graphics routines for C, that memory is set aside for your variables in the exact place where you define them.

For example:

if c99 compiled the following routine:

```
main()
{   int a,b,c;
```

```
        scanf("%d,%d,%d\n",&a,&b,&c);
        printf("a-%d b-%d c-%d\n",a,b,c);
    }


BASIC Version---> INPUT A,B,C
                  PRINT "a-";A;" b-";B;" c-";C
```

When C encounters the **int a,b,c;** which defines the integer variables a,b, and c, it simply puts DATA directives at it's current location in the source code that it's producing.

eg:

```
A     DATA >0000
B     DATA >0000
C     DATA >0000
```

This causes the program to reserve a 16-bit word for each variable, and assign the variable name to it.

This is only a simple example. Integer variables in C99 are 16-bit values, so standard DATA directives are used. Other variable types such as the floating point module are more complicated, so I'll stick to int's as it's alot simpler.

If the compiler found the following line:

```
a=a+b+c;
```

Then it would generate the following TMS9900 instructions to add the variables together. To do the same thing on any other machine is impossible. They can't directly add memory together!

```
A  @B,@A  add B to A and store result in A
A  @C,@A  add C to A and store result in A
```

The outcome of these two instructions cause B to be added to the contents of A and the total is retained in A, and then add the contents of C to that, and retain the total again in A.

The power of our machine means that we can do it directly. On EVERY other processor you can think of, you have to waste time shifting things into registers first! That's one of the reasons we can still write and run programs on the 4A that kill 486's!

So, how does C99 handle parameter passing?
Well, if, after defining the variables in your particular current routine, the compiler encounters a C subroutine (called functions), it will load the parameter variables that are being passed to the function onto its stack, at the next available location (which is ALWAYS pointed to by R14, because C only compiles its programs to run in one Workspace).
It will then jump to the called function, which will

obtain the parameter variables it requires by reading them from the stack in reverse order by using indexed addressing.

For example:

If the function was passed the following variables from the calling routine:

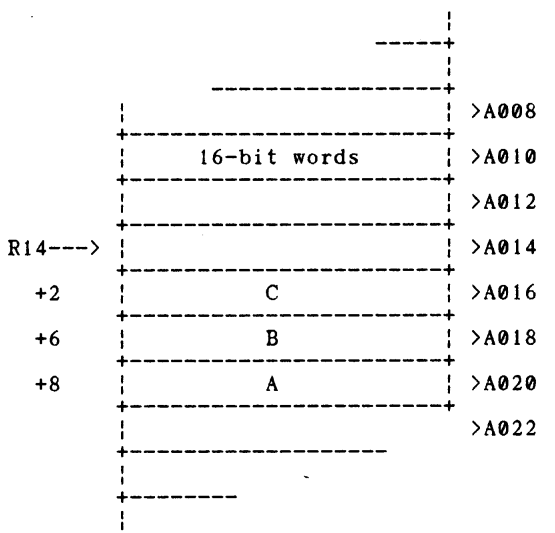
```
        function(a,b,c);
           |       |
           |       |
    Function Name  |
                   |
              Parameters
```

The following code to get the variables from the stack would be generated.

```
MOV  @2(14),R2   variable C
MOV  @4(14),R1   variable B
MOV  @6(14),R0   variable A
```

The source operand (where the variables are being fetched from) is defined using indexed addressing.

This means that the value that is being moved into register R2 is found at a location that is pointed to by R14+2.



They appear on the stack in reverse order, because when the calling program calls the function and puts the parameters onto the stack, it puts them on as it gets to them. A, then B, then C. The stack builds downwards, and after each variable is put on, the stack pointer is decremented. Therefore, the

function (or your TMS9900 subroutine) has to read them back in reverse order.

And that's all there is to it!

I recommend that any Console ONLY owners who don't have the mini-memory module, rush out and write to Texas Instruments and get hold of a mini-memory GROM, and a 4K static RAM chip, or even a complete cartridge if they are still lying around in some amazing wharehouse in Dallas somewhere, as I'm sure they must be.

I don't doubt that there's probably enough spare GROMS, and 9900's left in the world to keep us safely going for a lifetime, or until the Texas Instruments Quantum chip arrives!

Something else I didn't mention in my last article was that Colin Hinson had met Don Bynum.

Who?

You've heard the phrase, "No Robert Johnson, No Rolling Stones or Eric Clapton.",

Well, there's also a phrase, "No Don Bynum, No TI-99/4A."

I'm sure there must have been more people than him working on the design, but apparently, Don Bynum is the man we can consider most responsible for providing a machine that does just about everything, and is 5,000,000,000,000 times more expandable than a BBC Micro, Apple Mac, and PC put together, and that goes on and on and on, no matter what you throw at it.

A machine that has a standard 1978 sound chip that plays digitized sound files that have been recorded on a PC sound blaster, but plays them better than a 1993 Siemens/Nixdorf 486 PC!

A machine that supports three ST-506/412 hard disks, and god knows how many SCSI hard disks!

A machine that can support 8 floppy disk drives (which are the equivalent of 16 drives on a BBC micro if the BBC could ever support more than four!).

A machine that supports 8 Meg or 16 Meg RAM Disks. A machine that will support a Phillips Laser Disk player and give you interactive video (TI Video Controller Card!)

A machine that let's you share data between programs by using background batch processes.

A machine that supports two cassette recorders.

A machine that has totally independant video memory that is addressed by the Video chip, rather than the CPU, which gives you unlimited graphics expandability.

I find it sad that when the BBC wanted to do the Domesday Project with interactive video, Acorn had to totally re-design

the BBC and call it the Master, and it was still nowhere near as powerful as the 4A. I can remember we had one put into our library at school when I was in the 6th Form, and everybody was going mad over it. It was removed, and sent to be repaired after only a week!

The BBC wanted more colours, and they wanted it to support Genlock so you could super-impose computer graphics on top of video pictures from the laser disk.

Acorn actually went to Texas Instruments in Bedford, to have a custom chip manufactured using their E-Beam technology.

"What's E-Beam?"

Well, TI produce custom chips cheaper than anyone else, because they don't use masks. They burn the circuits straight onto silicon, using an electron beam which is accurate to the distance that your finger nails have grown during the last 60 seconds.

Unfortunately, when Don Bynum designed the 4A, he didn't have it all is own way! The reason TI came up with the idea in the first place was to waste chips! They had designed this revolutionary new processor which did for computing what Robert Johnson did for the music industry.

It was so far ahead of its time that no one would dare use it, and thought that it must be too complicated if it was a 16-bit chip. In fact, it's so simple, that it's much simpler to program than 8-bit processors such as the 6800, 6502 etc.

Comparing Intel chips against the TMS9900 is like comparing dance music played on a synthesizer, to the raw talent and power of an early John Lee Hooker track with just an acoustic guitar. There's alot less to go wrong on an acoustic guitar!

When poor old Don had worked hard on an initial design, he took it to the almighty creators, who immediately sent him back!

Too expensive?

He was sent back to the drawing board to take out all the 16-bit wide RAM!

He added the multiplexer, but did have his own way on the scratch PAD. This area of 16-bit RAM is sacred, and is reserved for the chosen few who have the knowledge to position their Workspaces there!!!!

And so, Don returned to the almighty creators, and knelt before the lord of all console owners, who did permit such a gracious device to be finally issued, with somewhat reduced features, and interpreted BASIC interpreter (which wasn't too bad, considering.)

The almighty creators were not aware of the unique Device Independent Operating system which must have escaped their notice on the Day of Judgement. It's not surprising though I suppose.

There are still things in the console that we still probably don't know about!!

Is there another program similar to SoundFX, just waiting to appear, which we can load, and will do something totally amazing?

With all the software that we have now, and all the hardware that's coming out lately, I think we can safely say, that for the TI-99/4A, and GENEVE:

"The Future Starts Here!"

Let's look forward to the next decade.

Well, what's next?

I'm afraid I haven't got much to report again on my CAD program. My time has been taken up with something else! I'm moving up in the world. Seven or eight feet up, to be exact!

"What's this fool on about now?!?!?"

Well, enough's enough!

I haven't got enough room in my room, so I'm building a new room!

Yep! I'm converting the loft (attic) into a 17 foot by 19 foot computer room, and T.V. room. Another advantage is that it won't be having a window, so it's **100%** secure.

From the outside it will be just any old loft that's full of spiders and junk (But, I don't own a PC!). A burglar won't bother messing about with a loft ladder and risk getting caught. We've had a burglar alarm fitted too, so it's even safer.

Everything will be completely secure up there, and I'll have alot of room for more endless expansion (SCSI here we come!)

By the time you are reading this, I should have already moved in (hopefully). It's been difficult trying to find the time to fit it all in. Arriving home at 6pm doesn't help, and time seems to be reduced at the weekends.

I've also had this article to write too, but I've spent time on it every morning at work before I start. I'm using Protext 4 again, running under Windows (Sad isn't it!). At this point in the article, if I check the memory used, it's 44K long!

I won't be printing it on the PC though. It'll be transferred into TI-Writer (or maybe Myarc MY-Word) via an ASCII upload on the Amiga, and a log text to disk on Telco.

By the time you have finished reading this article, you will have read more pages than were included in the first issue of TI*MES, back in Summer 1983!

I don't like to disappoint my readers!

PC's do have their uses! It means I can write my article virtually anywhere, and then finish it off with a professional computer using the TI-Writer Text Formatter when I get home.

All I have to do is a few CTRL-U commands to produce control characters, and I can change font to highlight my text at any point.

Hopefully in my next article, I will be able to include my Batch-It menu for TI-Writer, that will allow different fonts to be selected from a menu!

Having PC's at work has also given me another idea which I hope to have working, and included in the next issue. The idea might initially only appear to be appealing to machine code programmers, but you will realize that it will benefit just Extended BASIC programmers too, as long as they've got a disk drive, and can run Funnelweb.

I'm not sure about Mini-Memory at the moment. I'll have to check it out and see if it will cause a lot of conversion work.

I've got the Mini-Memory cartridge on disk as a Gram Kracker file for the GENEVE, but it's not something I use for machine code programming, since I've got the Editor/Assembler cartridge, and I've also got Funnelweb.

The problem with the Mini-Memory cartridge is that the Line-By-Line assembler will only use two-character label names etc.

This means that my program will have to generate two entirely seperate versions of source code, depending upon the Mini-Memory, or Extended BASIC version. I will also have to think about doing a version for TI BASIC for use with the Editor/Assembler cartridge, but I think that there's some sort of an offset that has to be added to a character when you're displaying it to the screen from a TI-BASIC LINK'ed 9900 subprogram.

For now, I think I'll concentrate on getting the Extended BASIC version finished, and then do the alterations for Mini-Memory or Editor/Assembler.

I suppose I'd better say what the program will actually do!

Imagine you don't know anything about machine code, and you're writing a game. Wouldn't it be amazing if you could have some LINK'ed TMS9900 subroutines that generated screens for you, so you could just call them with a quick CALL LINK.

Before you had time to say "Stackless RISC architecture", the screen would be immediately displayed, complete with colour and character set definitions

So, how can you do it?

With my new idea called ScreenWriter. If you want a certain screen for a game, just use ScreenWriter to draw it. I will have to read up on defining colour sets and doing character definitions in machine code, but it shouldn't bee too difficult.

I won't include sprites (maybe a later version will!). In the first version I'm going to try to use sprites as an overlayed menu, so you've got the full working screen area to chew on. When you've got the screen complete, with colour, and graphics definitions, you will be allowed to save it for re-editing later, or you can tell the program to process it. It will then ask you what the label should be.

The label is the thing that calls it. If you said the label was SCREEN, then, when you wanted to display the screen, you would type:

```
CALL LINK("SCREEN")
```

and XB would jump to the machine code, and the screen would be displayed completely, before you have time to blink!

After ScreenWriter's asked you for the label, it will ask you where you want to send your source code. This will be a **device[.filename]** prompt, so, using our decent filing system you could type PIO to send it to the printer, DSK1.SOURCE to send it to floppy, or HDS3.EA.SOURCE to send it to the Editor/Assembler source code subdirectory of hard disk 3!

It will then send, to your chosen device, the source code for a routine that will generate the screen you have drawn!

This will then need assembling, using just the R option of either the E/A assembler, or Funnelweb. Either one will do.

R means look for R on Workspace register names (e.g. R7). If you wanted to write the following instruction:

```
LI  R7,4
```

which loads the decimal value of 4 into register R7, then you can write:

```
LI  7,4
```

but it's quite sad and confusing, and looks very down market and IBM'ish. It looks alot more sexy if they're written in full with the R in front.

The other options are:

L - Generate a list file
S - Do a symbol table dump
C - Compressed
T - Include text definitions in symbol table.

You don't need any of these because:

L. List files are so you can see an output of the assembly process which includes any errors, but the source code is computer generated, so there aren't any errors.

S. You don't need a symbol table dump, because, there won't be a list file to include it with.

C. TI-BASIC or Extended BASIC won't load compressed object code, unless there's a compressed object loader that I don't know about; But CALL LOAD("DSK1.OBJECT") won't load compressed code.

T. You don't need text definitions in the symbol table, because there won't be a symbol table!

That's it for now about ScreenWriter, the full listing should hopefully be included in the Spring 1994 issue.

On a similar sort of subject, I've got another idea that's related to screen control from machine code. I got the idea from the Amiga! If anybody has seen an Amiga working, you will know that it works on a principle of screens, which can be dragged on and off of the display.

If you load something like Deluxe paint, you can point to the menu bar, hold down the left mouse button, and drag the Deluxe Paint screen completely off of the display.

This gave me the idea of doing something similar for Extended BASIC, or even BASIC screens. The advantage for anyone with an Editor/Assembler supercart, is that 10 extra screens can be stored in the cartridge at >6000.

The routines can totally replace PRINT, DISPLAY, or HCHAR/VCHAR. There will be a special LINK routine that writes the data to whichever screen you specified. There will then, be a routine that updates all of the screens on the display, in order of priority, and in the current screen position where they should reside.

There will also be a separate routine which allows a screen to be slid up or down the display, so you could have a program that uses different menus on different screens, and slides them in and out when required!

Chew on that MicroSoft!

With ScreenWriter to write too, it might not appear until the Summer issue, but at least it gives us something to look forward to.

What's next, apart from you all wondering "How long can this article go on for?!?!?!?!?"

When is a bit-map not a pixel map?!?!?!?!?

What am I on about? Well, have you ever considered how the 4A handles bit-map?

If you put TI-Artist into draw mode, or point mode, where it draws a single pixel when you press the fire button, then

you'd expect it to tell the VDP to draw just that pixel.

Well, if you've got a 9938, or 9958, and YAPP, then it would, but the standard 4A doesn't, and neither does the standard TI Bit-map mode, even if you've got a 9938 or 9958 (Only the high-res screens work as a true bit-map).

Why isn't it a true bit-map then?

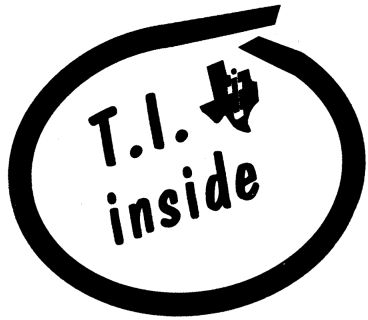
Well, you may, or may not know that the standard TI "Bit-map" has two areas of memory in the VDP's RAM.
These are the pattern table, and colour table. Yep, that's why you've got the _P and _C files when you save a picture with TI- Artist!

Since we're just talking about pixels, we'll forget the colour table and just concentrate on the pattern table.

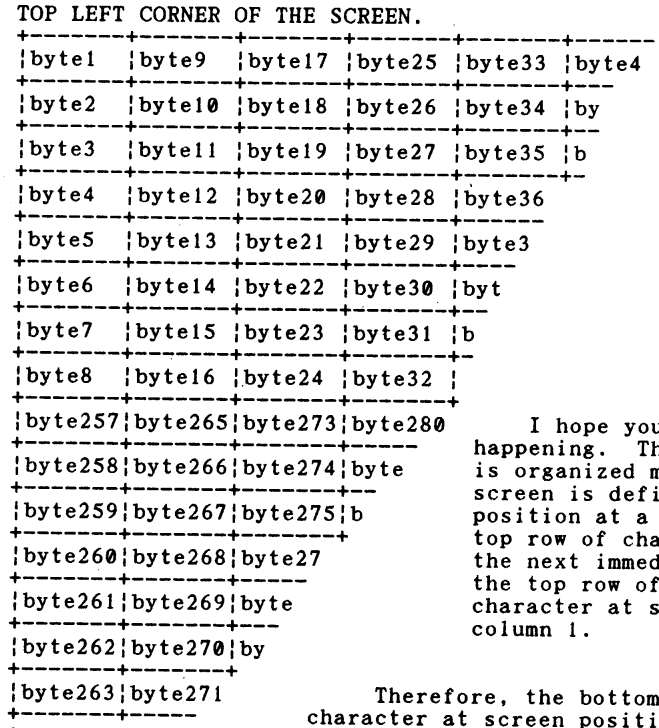
The pattern table works like a monochrome mode would on most other computers. It works totally independantly of colour, because the tables are completely separate.
Each pixel has only a single bit to describe it.
Each byte of memory in the pattern table describes 8 pixels on the screen.

The first byte in the pattern table describes the first 8 pixels on the top row of the screen. The second byte describes the second row of eight pixels, right up to the eighth byte, which describes the 8th row of 8 pixels.

This is where it gets interesting!
The 9th byte describes the next 8 pixels along on the top row, which is more or less like saying the 9th byte in the pattern table describes the top line of the character that would be displayed if you did row 1 column 2, in an HCHAR from XB. Here's a quick diagram showing a simple pattern table layout, with each byte numbered:



TOP LEFT CORNER OF THE SCREEN.



I hope you can follow what's happening. The way the memory is organized means that the screen is defined a character position at a time, along the top row of characters, and then the next immediate byte will be the top row of pixels in the character at screen row 2, column 1.

Therefore, the bottom pixel row of the character at screen position, row 1, col 32, is byte 256. It carries on like this, right down the entire screen.

The reason I have spent time studying this is so that I could write a pixel testing routine for C99, because the main bit-map library didn't include one!

The advantage with C99 is that any C compiler can directly link to assembler, and include assembler source code, directly as part of the C program. Because of this, I had the entire source code for plotting a pixel to the screen, which was written by Jay Holovacs.

Using this source code, I was able to study what was happening, by writing a table that showed the data as it flowed through the workspace registers:

R0	R1	R2	R3	R4	R5	R6	R7	R8	Instruction
*	x 45	y 38	*	*	*	*	*	*	
*	45	38	*	*	*	*	*	45	MOV R1,R8
*	45	38	*	*	*	38	*	45	MOV R2,R6
*	5	38	*	*	*	38	*	45	SRA R1,3
*	5	4	*	*	*	38	*	45	SRA R2,3
*	5	4	*	5	*	38	*	45	MOV R1,R4
*	5	4	*	40	*	38	*	45	SLA R4,3
*	5	4	*	-5	*	38	*	45	S R8,R4
*	5	4	*	5	*	38	*	45	NEG R4
*	5	4	*	6	*	38	*	45	INC R4
*	5	4	*	6	4	38	*	45	MOV R2,R5
*	5	4	*	6	32	38	*	45	SLA R5,3
*	5	4	*	6	-6	38	*	45	S R6,R5
*	5	4	*	6	6	38	*	45	NEG R5
*	5	4	4	6	6	38	*	45	MOV R2,R3
*	5	4	128	6	6	38	*	45	SLA R3,5
*	5	4	133	6	6	38	*	45	A R1,R3
*	5	4	1064	6	6	38	*	45	SLA R3,3
*	5	4	1070	6	6	38	*	45	A R5,R3
6	5	4	1070	6	6	38	*	45	MOV R4,R0
6	5	4	1070	6	6	38	256	45	LI R7,>0100
6	5	4	1070	6	6	38	4	45	SRA R7,0
1070	5	4	1070	6	6	38	4	45	MOV R3,R0
1070	1024	4	1070	6	6	38	4	45	BLWP @VSBR
1070	1024	4	1070	6	6	38	1024	45	SWPB R7

I suppose you want me to explain all that!?!?!?

Here goes! Listen very carefully, I shall say this only once!

2022 note: That line above is taken from a BBC sit com called Allo Allo, and was usually the first dialogue from the leader of the Resistance. Who often had to repeat what she had said...

I'm skipping the parameter passing section of C99, because you should have already read that, earlier in my article.

So, we're assuming that the x and y coordinates of the specified pixel are now contained in registers R1, and R2 of the current workspace (which starts from >8330 in scratch PAD).

First, the routine makes a copy of register R1 into R8, and makes a copy of R2 into R6. This is achieved with the instructions: MOV R1,R8 and MOV R2,R6

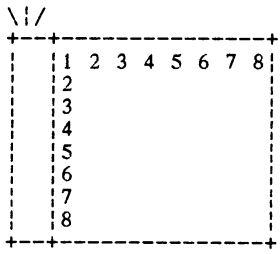
Then, we need to find the character column position, which is done by dividing the original x value by 8. Divides are slow, so it's quicker to do a SRA R1,3 which achieves the same effect!

We also need to find the character row position, which is done by dividing the original y position by 8. Achieved by shifting again: SRA R2,3

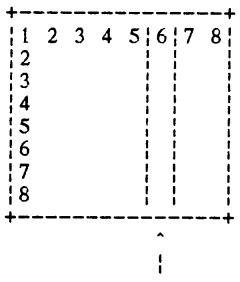
Then, make a copy of register R1 into R4 and multiply R4 by 8:

```
MOV R1,R4
SLA R4,3
```

This gives us the number of the pixel column which is just before the left pixel column of the particular 8 by 8 pixel area that we are interested in:



Then, we need to subtract our desired pixel column from this so we can find the pixel column position in the 8 by 8 pixel area. This is because it's stored as bytes, and we need to find the bit position in the particular byte that specifies it:



This is calculated in a complicated way for a reason which I'm not sure of. This gives a negative value in R4, which is then made positive, and then 1 is added by INCrementing, to give the correct bit position.

Then, we need to find the exact byte in the pattern table that contains our desired pixel. This is initially done in more or less the same way as we found the bit position value:

```
MOV R2,R5
SLA R5,3
S   R6,R5
NEG R5
```

Then, make a copy of the number of character rows into R3:

```
MOV R2,R3
```

and multiply it by 32:

```
SLA R3,5
```

add the number of character columns:

```
A   R1,R3
```

We have now got a value of 133 in R3.

Have you followed what we have just done?

No !

O.K. Let's just chew on it for a minute!

The pixel we are plotting is at x,y coordinates 45,38

If you divide each one by 8, you will find which screen character this pixel is located in.

x -> 45/8 = 5.625

y -> 38/8 = 4.75

What we have done so far, is more or less, find which 8 by 8 character on the screen, our pixel is in. As you can see from the calculations above, this is row 4, column 5.
We need to find the position of this character referenced by counting along each row, rather than specifying the character's row and column values.
This is because the VDP refers to the screen as an entire block of memory. This is why we've got 133 in R3. The character that contains our pixel is 133 characters from the beginning of the screen, if you count along each row.

The next instruction is:

```
SLA R3,3
```

This multiplies the character position of 133 by 8. Why?
Well, we are trying to set a single pixel, so we need to narrow our search down to a single byte. To find the number of bytes that make up 133 character positions, we just multiply the value

by 8. The SLA is the fastest way of doing this, because it works alot quicker than doing a multiply (MPY) instruction.

Then we need to point to the byte in the next character position that contains the pixel. This is done by adding the offset that is contained in R5.

```
A  R5,R3
```

Register R3 now contains the VDP memory location of the byte that contains the pixel that we want to set.
O.K. We've gotten over the worst of it! Now, all we need to do is to set the desired pixel at that byte.

How do we do it? We've got to make a mask which will allow us to set the pixel. This is simple on decent machines. If we'd got something as sad as a Spectrum or BBC, then we'd be buggered, but we haven't, so we're perfectly safe!

We load the value >0100 (hex) into R7. This gives R7 the following bit pattern:

```
     <------- 8 bits -------> |  <------- 8 bits ------->
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
    Most Significant Byte       |    Least Significant Byte
```

The byte that we are interested in, that will contain our mask, is the least significant byte. This is because the VSBW (Video Single Byte Write) routine, looks at the LSB for the byte that it sends to the VDP.

You will notice that the bit, immediately to the left of the least significant byte (which is the least significant bit of the most significant byte!) has been set to one. Now, if our pixel needs to be set at the sixth bit in from the left, we would need to shift the bit that has been set to one, to the right by the correct number of locations, in this case six.
We have the pixel position that we want, already loaded into R0, so how do we shift the contents of R7 by this amount?

This is done by the most amazing chip ever produced, and I didn't know this until I deciphered the bit-map routine!

If you give the 9900 the following instruction:

```
SRA  R1,2
```

The chip will shift the contents of R1 to the right by two bit positions.

However, if you give the following instruction:

```
SRA  R1,0
```

you would think that the chip wouldn't shift the register at all, but, in fact it does. This confused me the first time I saw it and I had to dig out the Volumes of Almighty Commandments for spiritual guidance.

If you specify 0 as the shift value, then the selected register is shifted by the number of bits that are specified in the value of R0.

If R0 contained the value 3, then SRA R7,0 would do a Shift Right Arithmetic on R7 by three bit positions!
Just tell me that there's something that TI didn't think of. I know they didn't think at all about marketing the 4A properly, but they definitely had everything else worked out in fine detail.

So, to get the bit into the correct position, we just perform the instruction:

```
SRA  R7,0
```

O.K., so now, we've got the pixel in the correct position in Register R7, how do we set the pixel on the screen? This is simple! The correct address of the desired byte is contained in R3, so we move this into R0, with:

```
MOV R3,R0
```

Right, we can go ahead and give the VDP a quick Bullwhip!

```
BLWP @VSBR
```

This reads the byte, that will eventually contain our pixel, from VDP RAM, and puts it in the most significant byte of R1. We have set our pixel position in the least significant byte of R7, so we just need to shift this left by 8 positions to get it into the most significant byte:

```
SLA  R7,8
```

Now, we need to set this bit position in the byte that we have read from VDP RAM. This is done with the SOC instruction which sets any bits to 1 in the second operand that are also set to 1 in the first operand.

```
SOC  R7,R1
```

So we've set the pixel, and all we've got to do is write the new value back to VDP RAM so that the pixel will be set.

```
BLWP @VSBW
```

And that's it. The pixel has been plotted to the screen! If our routine had been called from somewhere else by using a Bullwhip, then we use a RTWP (Return Workspace Pointer) instruction to return control to the calling routine.

Here I was going to include something to excite our members!

News about the new SCSI card!

I haven't had a very good response from Bud Mills, etc. And have not had time to review my other sources of information about the card. All I can report at this time is that a team of programmers are working on the DSR to make sure that it will be compatible with the MYARC HFDCC and the GENEVE!

I've got the Entire GPL Assembler, and Linker, from Mark Wills.

Mark came to stay with us last week, and he let me copy the software which comes on SIX disks, including full documents (A manual produced by TI Staff!) and demonstration programs. With GPL, you can do all sorts of things, such as scanning the entire keyboard with a single SCAN instruction! It even has a full Macro facility! Chew on that Microsoft!
Mark says Stephen Shaw did once publish part of the GPL documents in an issue of TI*MES, but I can't remember this. (I will have to check my back issues for this.) Therefore, the GPL Assembler might possibly also be in the disk library.

I will hopefully do a bit more of a report on the GPL Assembler in the next issue of TI*MES, despite it containing a warning that it should not be disclosed to persons other than Texas Instruments employees! If TI want to ignore our attempts to keep their excellent machine alive, then I think we have the right to ignore their threats to keep secret, any information that is rightfully ours. Every one of us is the legal owner of a GPL interpreter, so we ought to have the documents and assembler to allow us to write programs for it!

Don't worry, it includes a full linker that throws out option five files for Editor/Assembler!

Those members who attended the 1993 AGM will know that we were attempting to establish links with East Anglia Region 99'ers, and possibly West Midlands TI Users. Nothing has happened about this for ages, and I've got no more news on it, but we are hoping to establish links in the U.S.A. with Jim Cavanagh's group.

Mark and I first corresponded with Jim on his bulletin board, Am-Can friends, which is used by TI'ers on either side of the North American and Canadian border. We obtained this number from the issue of Micropendium that Gary brought on our camping trip to Iver Heath!

I phoned the board one morning and left a quick message saying who I was, and mentioning the group, and then Mark did the same thing. Jim then informed him that he had only received one other call from England, and that was from me!

Jim has expressed a great interest in the group, and in the 80-column card, and he'll be hopefully reading this article, as we are sending his group a copy of this issue.

I have already done a semi-review of the 80 Column card before, but here's a quick list of its features for Jim's group's benefit, and for the benefit of any of our readers who missed it, and also any new members.

80-Column Card

Features:

YAMAHA V9958 Video Display Processor which gives all existing TI graphics modes, as well as GENEVE graphics modes, but also gives hardware screen scrolling, and YJK Display which will display 19268 colours!

Unique custom design which allows any amount of Static RAM chips to be used.
We are undecided on the base memory because of chip costs, but the board will contain no less than 256K RAM, and will be directly expandable without adding jumpers, to a minimum of 1Meg VRAM.

Built-in Speech Adapter.
Yes! The circuit will only take up half the space that components on other boards take up, so hopefully we'll include a through-port on top of the board, into which you can plug the card out of the Speech Synthesizer, which means you can move it out of the way into the Expansion Box!

We are also hoping that it will be usable on the CONSOLE ONLY, and there will hopefully be a small circuit that plugs into the side of the console and converts the output from the side expansion, into the same pin-out of the expansion box! If we sort out a DSR with a decent set of commands, it will also be programmable from TI-BASIC.
Imagine full, high-res graphics with 19268 colours, all from TI-BASIC.

Also, if we are clever enough to borrow a few ideas from RAM Disk DSR's, then it might be possible to turn part of the card into a RAM Disk!

I'll stop now before I mention the possibilities for word processing packages, with 1 or 2 Meg of VRAM to play with as a text storage area! Chew on that Pord Werfect with a sad limit of 640K!

Returning to the subject of Jim's Bulletin Board:
It's written by Tom Wills in Extended BASIC, but uses machine code for the communications routines, so it's totally customizable! Jim's sent Mark the details about it, and it's freeware, so it looks like our Bulletin Board could be up and running sooner, rather than later, and we can put After Hours back on ice where it belongs, since we have been given zero support, and haven't received anything from Ed Schaum, who had apparently sold his soul to the devil and is now running his board on an IBM!!!
Sadness of the highest order!!!!

If I'd got an IBM I'd never run a bulletin board on it,

and I'll never run a bulletin board on my Amiga. It's too dangerous in a world full of sad, materialistic, nintendo crazed dancers who write nice little viruses for each other!

It makes my day when I hear about somebody finding a virus. For my entire two years at University, doing a Computing Systems degree, I've had a single box of 3.5inch disks, one of which contains PASCAL and C source code.

Everybody else wastes time in college when we have an assignment, printing things out on the network to the VAX laser printer.

All of my source code gets uploaded to the GENEVE via the Amiga, and printed with professional software. Any viruses can't attach themselves to a decent filing system, and they'll simply De-resolve, as the film TRON would put it!!!

All of my assignments get typed on the Funnelweb editor, or MyWord for the longer jobs. I typed my group project for the end of the second year on MyWord, and then had to get it onto an IBM disk.

I did a text upload to the Amiga, but realized that the Amiga wasn't scrolling, because I'd not changed Telco to CR/LF on the Translate Out setting, which makes it send line feeds, as well as carriage returns, at the end of each line.

This worked, and I managed to get it into the main Amiga word processor, which is called Transwrite.

Unfortunately, every line came out with a carriage return symbol on it, and I decided to take them off. Unfortunately, because Amiga's (like PC's) have a CRAP filing system, they rely on the carriage returns to keep control of the text! As soon as I started taking the carriage returns off, I ended up with long lines that went off of the right hand side of the screen!
So I had to leave them in!

I finally added some more carriage returns, and took some others out, to get the text into a decent format (still nowhere near as easily as with TI-Writer!), and saved it on an IBM disk in Word Perfect format!

I gave the file to the one in the group who's job it was to collate all of the text, and put it together into a single report, and he went away with all of our groups' files, and was apparently using MicroSoft Word, for Windows!

The following Monday he told me that because of the carriage returns in the text, he had to spend an entire night removing them, so that MicroSoft Word could format the text properly!
He said he had to setup Macros to enable him to remove the characters!!!

SAD! If we had a similar problem, we'd spend ten minutes (usually less than 10) writing a little Extended BASIC program that reads the original file, strips off the carriage return (CHR$(13)), and prints each line of text out to a new file, without any carriage returns. It's simple.

IBM'ers haven't got such a decent version of BASIC, and

such a decent, well organized filing system though! Let's face it, IBM's haven't got any filing system!

They have to fart about with hundreds of options that are extremely hit and miss in their operation, and then just hope for the best!

I hope Gary has found time to include stories of how he's pi***d off his fellow students who lodge with him in Nottingham.

If they have a lot of calculations to do that need to be included in a report, they get the calculator out and work it out by hand. Then they have to type it into some s**t program like Work Perfect! They manage to complete one problem in half an hour! Gary sits straight down with Extended BASIC and writes a program to generate all of the equations, and print them to a DV/80 file!

He's finished several hours in front of them, and he's already got a TI-Writer file waiting for him! Therefore, the phrase, "It's Better Manually", might still not be correct for the IBM'ers!!!!

Our software is straight to the point, with NO MESSING! We know what we want, and so did TI when they designed it! Even the conversion of a DV/80 file listing of a BASIC or Extended BASIC program back into PROGRAM format. There have been many versions that go via the MERGE format first, but we have discovered that TI had one that goes straight from DV/80 to PROGRAM, and it was written in GPL in 1979!!! (Yes it is included with the GPL Assembler package!)

Where was I? Oh yes. Linking up with the U.S.A.

I phoned up from work last week to get the latest news from Mark, and managed to catch him on the portable phone! During our conversation he decided to take a wander down the high street and visited the local fried nutrition specialists (fish chip shop!)

Then, he stood outside a shop window until we'd finished our conversation and said he was going to go in and have a browse around!!! He said the fellow pedestrians must have thought he was a pimp or something! Well, he is the VICE Chairman!

One more subject, and you'll be glad to know that you will have read my entire article!

This one needs a big introduction!

AND NOW, a new feature which will be part of every one of my articles from now on (oh no! not more for him to write about!).

I got the idea for this, after buying the entire set of TI*MES magazines, issues 1 to 40, from Dr.David Baines.

I got the title of the feature from one of my favourite films, "Somewhere In Time", which is about time travel, and stars Jane Seymour, and Christopher Reeve.

So what's this all about? Well, I'm going to name the

feature *"Somewhere In TI*MES"*, and in each issue I'll travel back in time (back in TI*MES!) and report what was happening at that time of year, exactly 10 years ago!

Here goes!

Somewhere In TI*MES

No.1

We are currently on the Winter issue of TI*MES, and since TI*MES actually started in Summer 1983, I thought that I'd start by catching up, and cover the first three issues!

Summer 1983

The very first issue of TI*MES. On the front cover is a photograph of a complete system, including printer, several P.E.Box cards, Speech Synthesizer, acoustic coupler (MODEM), disks, and several cartridges.

The TI*MES title which is written in the top left of the photo, is not written in the standard Times Roman font!!!

There is also a box at the bottom of the page containing text, and a little picture of a console.
The text reads:

> "TI-99/4A Home Computer. The most advanced, most versatile Home Computer on the market today. All the power and capability you'll need, from the finest colour graphics anywhere to the wonder of synthesized speech. And it is simple to use right from the start."

It's still true ten years later, but they forgot to mention a sound chip that plays digitized sound files!!!

Well, what did this issue contain?

1. Special Review.
(borrowed from Argus Specialist Publications Ltd.)

This was the golden age of computing, with still a few months to go before T.I. were to announce their sad departure from home computing.

In 1983, TI*MES could also have been aimed at people who had not yet bought a 99/4A.

The first issue contained a full review of the 4A, which went from the ports on the console, the layout of the machines internals, and also the expansion box, and peripherals such as the P-Code card.

It even made a comment about Video Chess making strange moves! They were obviously getting near 2 hours, and had not made 40 moves, so Video Chess started to waste their time!

The review closed with the following:

> "The TI-99/4A is certainly a good contender for a home entertainment unit. Whether it stands up as a complete computer system is, however, still an open question."

I think we can very safely give them an answer to that!

2. Your letters.

This letter was a reply to Clive Scally, from Texas Instruments, about an enquiry relating to new peripherals. It turns out that these were Hex-Bus peripherals:

4-Colour printer/plotter £149.95 inc VAT.

RS232 & Centronics interface £119.95 inc VAT.

and Wafertape Microdrive (chew on that Sinclair!) £109.95. This was courtesy of Robin Frowd (Home Computer Program Manager) Now there's a job I wouldn't have minded!

3. Feature, Computer Arcade Games (by Howard Greenberg)

This was a description of arcade games in general, and not a review of 4A games!

It did announce that Atari were going to release Pacman, Centipede, and Defender for the 4A by the end of 1983.

This was followed by several little comments, including a warning to not believe "console only" software if the authors claim fast flowing graphics. Some TI-BASIC cassette software claimed this, but without Mini-Memory, or Extended BASIC, or other hardware, it would be impossible to produce fast graphics and sprites.

There's also a call for users to buy cartridges from Parco Electrics, and not from their local branch of Curry's. Them were the days!!! When you could walk down the high street to buy 32K, and not have change out of £100!!!

I was given Parco's address by Mansfield's branch of Dixons, when I tried to obtain all my main hardware (32K, TI DCC and 1 drive, and Edito2/Assembler). I got all this as a job lot, in a trip down to Parco in 1983!!!

I probably never thought I'd still be using it, and relying on it ten years later. I knew I'd never give up on the TI though.

Everybody else at school said what they had got was better. Even some fool who had a Mattel games console said the speech was better, just because he'd heard nothing but Extended BASIC. The TE-II cartridge wiped the smile from his face!

Four phrases spring to mind:

"Just you wait and see"

"The end is near earthling!"

"Never trust a worm"

"Dioxyrhybose Nucleic Acid"

which are from Peter Pan, Super Demon Attack, Super Fly, and Plant Genetics!

4. Program, Jumping Jack.

This was an example program for movement of a bloke around the screen that was provided as a start for anyone producing a maze game.

5. Arcade

Games and high scores for under 18's, who are most likely over 18 now!

The Attack,	level 1	- 200,000 by Ken Der
TI-Invaders,	level 1	- 13,900 by C. Hunter
Car Wars		- 31,000 by Dave Robichand
Blasto		- 36,000 by David Lewis
Tombstone City		- 352,000 by Jamie Ash

6. Sound.

Some simple sound routines, including a little keyboard routine:

```
1 INPUT DUR
2 CALL KEY(5,A,B)
3 IF B=0 THEN 2
4 NOTE=A+120
5 CALL SOUND(-DUR,NOTE,0,NOTE+50,0,NOTE+100,0)
6 GOTO 2
```

This article opened with the comment, "Texas owners are lucky to have one of the most versatile computers on the market. The sound capabilities of the 4A are unrivaled."

There was no mention of the ability to convert and play, digitized PC .SND and .WAV sound files!!!

There was a little advert for a new software company, called Random Software, with some VERY wacky titles!

RS0001 - Matchstick Man meets the hopper Axeman and The Fog

RS0002 - Desmond Diamond (Great new cult hero)
RS Demonstration

RS0003 - Snailman
Snailman II

Two-tape packs were £5.00 each.

Did anybody out there buy any of these? And were they any good? And, does anybody have a copy lying about somewhere?

7. Cassette Problems

There was a problem page, related to cassette recorders, which gives a list of the tape recorders that DO work with the 4A.

8. Hints and tips

Programming tips mainly, which included finding the amount of free memory in TI-BASIC:

```
>1 MEM=MEM+8
>2 GOSUB 1
>RUN
  * MEMORY FULL IN 1
>PRINT MEM
```

and, also a tip on how to use PRINT statements on multiple-statement lines.

9. Joystick Review.

This sounded amazing, and was a joystick manufactured by Arcade Hardware. It had a steel shaft, which therefore couldn't snap (unless you played SHAMUS from screen 1 to screen 128 without stopping for tea!!!)

The base of the joystick was a grey plastic box, and the stick itself was mounted with a sealed sprung pivot, and was surrounded by grease. 8-way Lever operated microswitches, and a sealed fire button on the side of the shaft.

Just below the joystick review was the cartridge exchange, which was the beginnings of our module library. In 1983 it contained:

Blasto', Soccer, The Attack, Car Wars, Chess,
Hunt the Wumpus, Household Budget Management,
Addition & Subtraction, A-maze-ing,
Video Games 1,
and the TI Hardware & Software developers
cartridge (Mini-Memory to you!).

10. Cursor, moans & groans page.

Asking if you thought 99'er magazine was worth £3 per issue, and if so, where to write to subscribe.

Then, a moan asking TI when they were going to release TI Logo II in the U.K. because it was decided that Logo would be the official language of the classroom (It's a pity that the government didn't recommend a computer for the classroom while they were at it!

The BBC is nowhere near the educational standard of the TI).

Then, a moan asking TI why 4A owners should pay £10 extra for cassette leads? I got twin cassette leads with my console which allows the superior machine to handle two cassette recorders at once, so I don't know what this moan is about!

A moan about some consoles being sold with free Speech Synthesizer, which is of course useless without some software to use it, such as Extended BASIC, or TE-II.

Thank god I was talked into buying an Extended BASIC cartridge instead of Speech Editor by a salesman at ASDA. Yes! ASDA! The Supermarket! I even got my Munchman and TI-Joysticks from there! That was the first hardware I had for the 4A on the Christmas just after buying it!

Next was a thank you note to Argos.
Someone bought a 4A from there which had a fault, which made it switch from upper case to lower case, in the middle of typing a line, and some letters didn't even display. It was taken straight back, and Argos handed over a brand new console. I've never had a single problem with my 4A, and mine was from a mail-order catalog!

Next was a question to TI:

> If a Texas Instruments electronic story book cost £39.95 back in 1983, why was the speech synthesizer - that really only contained the speech chip, and ROM (which probably had fewer words than the electronic story book) - more expensive?

At the bottom of this page was a quite interesting statement:

"COMING IN OUR NEXT EDITION

** NEW COMPUTER FROM TEXAS INSTRUMENTS**
EXCLUSIVE INTERVIEW WITH SOFTWARE AUTHOR"

As you will soon read, the review didn't appear! It was due to review the TI-99/2, which was a 10MHz 9995, and was built to compete with the ZX81 (wipe the floor with the ZX81 more like!) but they decided not to do it, and wait until the next TI launch, which should have been the TI-99/8, which would have completely buggered the competition, but it was never released, and TI pulled out of home computing.

Next was a listing of some of the World-Wide TI User Groups:

Australia

TISHUG, PO Box KX 101, Kings Cross, Syndey, Australia 2011

TIBUG, 23 Musgrave Street, Kippering,

Queensland.

TICHUG, 69 Canopus Cr. Giralang, A.C.T. 2616.

TIMUG, 59 Landsrom Quad, Kilsyth, Victoria.

ATICC, 7 Toorak Ave, Bellvue Heights,
South Australia.

TIUP, PO Box 246, Mt. Lawley,
Western Australia.

U.S.A.

MSP 99 User Groupm PO Box 12351, Saint Paul,
Minnesota, U.S.A. 55112.

Colorado 99/4 User Group, PO Box 3400,
Littleton, CO. 80161.

Cin-day User Group, 11987 Cedacreek Drive,
Cincinnati, Ohio, U.S.A. 45240.

Houston User Group (I bet they didn't have any
trouble getting hardware!)
1719 Briarmeed, Houston, TEXAS 77057.

Canada

Edmonton User Group, PO Box 11983, Edmonton,
Alberta.

Holland

TIJDINGEN, Haagweg 169, 2281 Aj Rijswijk,
Holland.

The first ever issue of TI*MES, then closed with a form to fill in, to place adverts in TI*MES, and the last page was an advert for Arcade Hardware, which in 1983 included:

The Arcade Machine Joystick (described earlier)

The Hexadecimal Keypad.

A numeric keypad that plugged into the joystick port and needed Extended BASIC. (Not a difficult thing to do. I had a similar idea myself once, but only built a few little things!)

Forthcoming attractions were a 32K memory expansion that cost £110.00 for the expansion box, or £125.00 for a stand-alone unit!
When we get around to our 32K project, it will cost about £12!!!

Also, a light pistol, designed to complement a new range of software from Stainless Software.

And that's it, a quick (longer than expected) guided tour of TI*MES, Summer 1983.

Here we go again.

No.2

Autumn 1983

The front cover, an overhead view of several people playing with a 4A, and the standard Times Roman font has arrived!

The issue opens with a short message thanking readers for supporting the magazine, and an apology saying this issue will not contain as many spelling mistakes, and will be more polished!

Then, news of price reductions of the 99/4A:

"October 1983 will be remembered as the month when the TI-99/4A went below £100. Many of you who paid £149-£199 last year will gnash your teeth" (mine was £189.99).

In the U.S.A. in October 1983, the 4A was selling for $89, which some people felt was below the manufacturing cost. TI obviously expected to make back their loss with the sale of expensive cartridges (now we can consider them as cheap when originally paying £70 for Editor/Assembler, and today Nintendo crap costs over £50!!! for the SNES)

So, what was in this issue?

Page 2: News and Views

ADVENTURE module is always sold with the Pirate cassette, and then other adventure tapes can be bought. Suggestion - couldn't one have the choice of adventure when buying the module?

COMING SOON from Pheonix Publishing. `Getting started with the 99/4A by Stephen Shaw'.

TI were becomming more aggressive with their marketing, and released a range of books, on which they offered a 10% discount to members.

The 4A was a bargain, priced under £100, a machine worth £400, and claimed to be equal to the BBC B? The 4A is far beyond anything as crappy as the BBC, and if the original IBM PC sold for £2000, then the 4A was worth £10000 (with expansion box, full memory, RS232, disk, etc.)!!!!

Spectrum owners had to pay £99, plus £20 to enable use of

cartridges. The 4A's got it built in. I suppose the Spectrum had a slight advantage in that you could load assembler from tape, which requires the XB or Mini-Memory cartridges on the 4A.

UK Computer Chart, October 1983.

1. Sinclair Spectrum
2. Commodore Vic 20
3. Texas Instruments TI-99/4A
4. Acorn, BBCrap B
5. Commodore 64

Page 3, Beginners Programming

How to put headers on the start screens of your BASIC games.

Page 5, Your Letters

Mrs. C. Hickman, from London, asking if anyone would supply complete hardware details, and memory map, of the 4A, because she was into hardware, and wanted to build an S100 bus interface for the 4A!

Mr. P.S.Richards, Leicester, "How do I solve `Golden Voyage!'?" He also wanted to exchange Mission Impossible, and Strange Odyssey for other tapes (presumably with the solution!)

Mr.C.Street of Crewkerne, Somerset, wanted to contact other ham radioers in the group. His call sign was GBXET

Mr.Dave Jackson, of Wavertree, Liverpool, who bought the 4A to learn to use computers (which is like buying a dictionary when you are learning to spell).
He complained about the manual supplied with the 4A, and bought a book, called "30 hour BASIC" by Clive Prigmore, which he found most enlightening. "Interesting and rewarding" were his exact words!!!

PARCO Electrics phoned the group saying, "We have won a Major order with BRITISH TELECOM for supply of TI-99/4A Computers."

A reply from Texas Instruments about cassette based software and cartridge based software. Anyone writing to TI with a program they have written, could have had the program marketed through TI, or had the software added to TI's brochure.

Putting software on Command Module can only be done by Texas Instruments (A simple knowledge of assembler, memory addressing, the cartridge port pin-out, and an EPROM programmer, and anyone could do it), due to manufacturing of the semiconductor memories. It is done when the program is written in TMS9900 assembly language.

In fact, this is also wrong! Alot of cartridges are not written in 9900, but written in GPL (Graphics Programming Language), and are interpreted. This is the

problem with TI-BASIC, and Extended BASIC. That's why Myarc Extended BASIC for the 4A, and Advanced BASIC on the GENEVE wipes the floor with TI-BASIC, or TI-XB, because they are written directly in TMS9900.

Page 7, Which comes first, teacher or the chip?

A little article about computing in schools, which was more or less a complaint about BBC Basic programming being taught, instead of decent things, such as studying stackless RISC architectures!

Page 8, Rambles, by Stephen Shaw *in TI*MES- previously in TIdings

The first ever appearance*of Rambles, by Stephen Shaw, including a hint for cassette users:

> When you insert a tape, the tape may be loose on the reel. Always press rewind, so that the tape becomes re-tensioned.

From the U.S.A. A review of `UNOFFICIAL 99/4(A)' magazine which described the use of INPUT in TI-BASIC, examples of using cassette files, press reviews and Maple Leaf Micro Ware, which was a software company that allowed Stephen's Stainless Software to distribute its titles over here.

Stephen also included a letter from the International 99/4A User Group, which reported that the President of Texas Instruments (I wonder if he still is?) Mr. Fred J. Bucy (I've actually got a picture of him!) was taking considerable interest in the 99/4A, and was opening a new office in Lubbock. It also said that Fred Bucy was personally handling Public Relations work, and that the 99/4A would make an appearance in the T.V. show "Matt Houston".

"Matt Houston" has been shown over here, and always includes TI computers, but I've never seen them using a 99/4A.

There was also news from the International 4A User Group about Do It Yourself modules, which were ROMOX's.

They were being sold blank for $10, and $300 for a ROMOX module programmer, $40 for a module eraser, or $38 for a ROMOX which was ready programmed with a game.

Page 9, An excellent publication. A must for TI Users.

An advert for 99'er Home Computer Magazine, August 1993, which describes the standard layout of the magazine, and detailed what was contained in the August 1983 issue.

Such things were a TI-LOGO review, a knitting program!, a TI-CC40 article (including use of wafertape drive), and also a put down of the BBC and its TUBE interface, which was supposed to allow second processors to be added (which the BBC definitely needed with its sad little limited 6502 effort!).

At the time the article was released, there were still no second processors for the BBC, but they thought they would

include an advert for a second processor for the 4A!
Admittedly it was an intel 8085 running at 8MHz, and with 50K of user RAM, which wasn't a patch on the processor in the console, but it did hammer the BBC into the ground, and even allowed the 4A to run CP/M! It cost $600, and even had an option to upgrade to a ten meg hard disk! Another possibility of adding an hard disk drive to the 4A!

Nowadays, our second processor range has been supplemented with the Novix 4016 Forth Processor which easily outperforms the 80386, but then again, what doesn't!

Page 10, PERIPHERALS

A complete glossary of hardware and cartridges, and what they do, including Expansion Box, memory expansion, joysticks, disks, assembly language, personal record keeping, TI-Writer, Term. Emulator II etc... etc... etc...

Page 13, an advert for Stainless Software

Including: a list of all the software companies for which Stainless was the U.K. distributor.

PS Software, FFF Software, Norton Software,
Not Polyoptics, Oak Tree Systems,
Maple Leaf Micro Ware (already mentioned),
Kuhl Software, Roach Software, and a few others!!

Best selling tapes in 1983 were WINGING IT, a flight simulation program in TI BASIC costing £10, and 99 Vaders, also at £10 (Why haven't I ever heard of these?)
Patrick Stassens programs NIGHT FLIGHT (a variation on the arcade Bomber) and MAN AND MONSTERS (like Digger) sold well, despite the speed limitation imposed by TI-BASIC. These both sold for £6 (MAN AND MONSTERS sounds like the game that Trevor and I have attempted writing! I like our title ALOT better though!!!!)

Page 14, BEGINNERS: CASSETTE STORAGE:

What to do if you've spent all night typing a program in, and type, OLD CS1, instead of SAVE CS1. When the instruction, PRESS CASSETTE PLAY, THEN PRESS ENTER appears, just press E, and the cassette routine will stop and put you back at the command line, so you can type the correct command.
Also, a service to cassette users to offer to make backups of commercial software tapes which have been saved using the protected option of Extended BASIC.
There's also a request for people who know how to remove the XB protection, for them not to tell anyone else! And theres a tip for Munch Man, about the SHIFT 8 3 8 cheat mode!

Page 16, Arcade

A review of Munch Man. (including cheat mode again!)

And some high scores:
Barrie Clarke, from Cleve, with 214500 on Parsec, and 59700 on Alpiner.
Simeon Prowse, from Avon, with 13000 on TI-Invaders, level 2.

Page 17, U.S. Scene

The home computer price war.

A nice little report about TI almost forcing Commodore out of business in the caculator market! But, Commodore bought a chip manufacturer, and flooded the market with the severely inferior 6502 which is used in lower quality machines like the C64, BBC Micro, Apple II etc. The result of this lead Commodore to gain a 30% share of the consumer computer market.

To make the 4A more competitive, TI dropped the official U.S. retail price of the 4A to $89. They also re-designed the internals of the console to make it cheaper to manufacture, and also to stop any cartridges that weren't manufactured by TI, from working on the machine. TI's idea was to almost give the machine away, and then make money on the cartridges. They should have done the opposite and flooded the market with VERY cheap cartridges earlier on, to make the machine more attractive.

Extended BASIC (or something better, containing full graphics) should have been included as standard with the machine, allowing assembly language programmers to get going right from the start.

The first few years of TI software were just cassette based, TI-BASIC only, games.

Page 18, PROGRAM LISTING

A program translated to the TI by Rick Mirus, of CIN-DAY Users Group in Cincinatti. It says it gives you pre-defined plotting! Whatever that means!

And, on page 19, a program for a sound and graphics demo, using CALL SOUND, and CALL HCHAR's.

Also, on the right hand side of page 19, something totally un-related to TI. A column about a story of a woman being elected as president of the Computing Service Association! It mentioned that the association was taking a constructive line in educating industry to select the correct equipment.

Clearly, they didn't achieve anyting, or we wouldn't have taken a step backwards a decade with IBM .

Page 20, A Review of Extended BASIC.

I like it when I find a page I don't have to write about! I think we know enough of the power of Extended BASIC.

Page 21

Comparisons of the same program written in TI- BASIC, and with Extended BASIC's multi-statement lines.

Page 22, Software Review

Funware modules.

Modules from a third-party module manufacturer, containing HEN HOUSE, and RABBIT TRAIL. I've seen Hen House, and it wasn't any thing special. I'm unsure about Rabbit Trail. I have got a game with a rabbit in it, and it might be the same one.

Page 23, Classified Ads.

Joysticks for £15!!!

Used cartridges, an advert for the module exchange, teach yourself BASIC cassettes, dust covers etc. etc. etc...

The Autumn issue closed with a back page advert for Christine Computing, and its range of software titles: BLAST IT, CORE!, FRUIT MACHINE, UFO, SPUDZ.

Well, can you manage just one more?!?!?!?!?!?!?

Winter 1983.

Inside the front cover, is an advert for a very weird game, which has turned into a bit of a classic. Unfortunately, besides being weird, it's also got a weird loader, which means that we haven't yet worked out the best way to get it on the GENEVE.

Excalibur didn't give us too many problems. Excalibur loads from cassette into Mini-Memory, and can be loaded at any time, even after going back to the title screen. To transfer Excalibur to disk, we loaded it into Mini-Memory from cassette, and then loaded C-SAVE from Editor/Assembler.

Then we switched back to Mini-Memory, and saved out the cartridge as a Gram Kracker file. This, of course, also took Excalibur with it in the 4K RAM!

Whenever we load this particular copy of the Mini-Memory from disk on the GENEVE, there is Excalibur, already sat in memory!

Suck on that intel!

The weird game I'm talking about, is of course, KIPPY's NIGHTMARE, by Dominic J. Melfi. I don't know what it was, but Mr. Melfi's games, always turned out a bit strange!

On the editorial page of the Winter 83 issue, there was an

happy new year message, which said:

> "inspite of the black cloud that TI cast over 4A owners. Prospects for '84 look bright."

Well, I'll wish every one a happy new year, because prospects for '94 look even brighter!

Despite being brighter, our prospects have a few more colours too! 19268 colours in YJK mode to be exact!

Page 1, Your Letters.

The usual bunch of requests for help with programming problems, hardware problems etc., but the first letter was asking if there was any news from TI about the future of the machine.

This directed the reader to a reply letter from Texas Instruments on page 3:

Dear Mr. Scally

TI-99/4A Home Computer

Thank you for your letter of 7 November. In response to the various enquiries which you have made, I can do no better than to let you have a copy of TI's announcement which was recently made to European Press.

The following points emerge from this release:

1. TI software availability is envisaged to be sufficient for the reasonably foreseeable future.

2. TI will honour warranties and will continue to provide out of warranty service facilities.

3. Although supplies of TI peripherals are limited, various suppliers in Europe offer products which are compatible with the TI-99/4A.

I will ask Mr. Phill Waggott (UK Consumer Products Manager) to provide you with a list of peripheral suppliers in whom you may be interested.

The intellectual property rights in the various TI products are vested in the parent company, Texas Instruments Incorporated of Dallas - and not Texas Instruments Limited. Your question concerning the possibility of a licence to produce and supply command modules and the full range of accessories/peripherals has therefore been referred to the Instruments Incorporated of Dallas - and not Texas Instruments Limited. Your question concerning the possibility of a licence to produce and supply command modules and the full range of accessories/peripherals has therefore been referred to the Corporation's patent department in Dallas. I am told that no decision is expected to be made on this issue until January of 1984 and again, I will ask Mr.Waggott to let

you know the outcome of their deliberations in due course.

Whilst I regret the inconvenience which may have been caused to your user's group as a result of the recent announcement, I am sure you will appreciate that no responsible company could have continued to allow losses of the magnitude announced to continue indefinitely. We believe that the actions taken by TI, particularly with regard to the continued support above specified, strikes a reasonable balance between the various interests which lie in Texas Instruments.

Finally, I do not for one moment dispute that your user's group comprises of serious home computer owners and it is my company's intention to act responsibly.

Yours sincerely

Rod P. Attwooll

Managing Director

This letter was followed by a little comment that the publication date was the 16th of January 1984, and there were no further developments. "Why are TI so silent?" Were Clive Scally's exact words!

Any RESPONSIBLE company would have marketed such an excellent machine correctly in the first place, and any responsible company would not put any third parties through all of the time wasting paperwork and letter writing procedure, just because someone wished to produce software for the machine, which would help TI's profits in the long run. On the bottom of page 3 was a piece of text from Post-Bulletin, Rochester, Minnesota, 12th December 1983.

New York Times News Service

NEW YORK - This is the year in which the home computer will join the sled and the bicycle under the Christmas tree.

In numbers that outstrip even the most optimistic predictions, Commodores, Ataris and Colecos are being snapped up from the shelves. Americans have embraced the home computer as their favourite gadget for a Christmas present, replacing the food processors and video games of Christmas past.

In the middle of 1983, the fierce price war erupted that caused the end of TI-99/4A production. If TI had sold it on features and not price, then it would have beaten the Commodore Vic 20 and Sh**ty 64 put together.

At the end of 1983, people were picking up real bargains though. The 99/4A, which sold for $1100 for years previously, was available for $49.
This made the 99/4A, the biggest selling computer for Christmas 1983!

The 99/4A nearly caused a riot as the low price tag sent a siege of shoppers in Greensboro, N.C., to storm K-mart and grab 99/4A's while stocks lasted.

Toys "R" Us was crowded with people in New Jersey and New York, buying up 4A's, unconcerned with the possibility that hardware, and software, and support, might become scarce!

Warren W. Zorek, consumer electronics buyer at Bloomingdale's said, "I wish we'd bought more", after only buying 10000 machines from TI on the day they threw in the towel!

Ken Williams, president of Sierra On-Line Software, in California, said, "Retailers who were caught with thousands of unsold video game cartridges last Christmas refused this year to order more than a 30-day supply of computer programs."

Williams spent much of 1983 developing software for the 4A, and said he had learned his lesson, after being bewildered by the pace at which computers come into and fall out of favour.

Page 4, FOR THE REST OF US BEGINNERS

Some hints on CALL COLOR and CALL SCREEN by Wayne Page. This included, highlighting some screens by making certain character sets stick out from the rest.

The right hand side of page 4 described a game which was listed on page 5, called CAVE MAZE.

A subterranean labyrinth! Sounds very interesting, but I haven't got enough time (or space) to retype it and include it here!

Pages 6 to 13, From conception, through development, to production. The Arcade Hardware 32K RAM.

The longest TI*MES article ever written up to that point!!! Not just describing the 32K expansion, but also reporting on other products that were available for the 4A, and how he had a faulty modulator that kept blowing up T.V.'s!

Howard Greenberg's 32K project sounded very ambitious, because he wanted the 32K to work on the side of the console, as well as work in the Expansion Box, and I don't think he was even using an adapter card for the side of the console!

Thanks for help with the card went to Stephen Shaw for trying out the 32K. This showed up a loading problem with Mini-Memory. Editor/Assembler and Extended BASIC worked fine, but Mini-Memory wouldn't have it!

It took ten minutes to sort out the problem! Two tracks were so close together, that data was being transferred between the two, and causing corruption! After scraping the tracks away slightly, they solved the problem.

Would this indicate though, that the Mini-Memory loader works slightly faster than Editor/Assembler. We

all know that the Extended BASIC machine code loader is about as efficient as using a 486!

Thanks also went to Gary Harding, Peter Brooks, and Richard Blanden. Three people he had never met, but were still prepared to offer advice and assistance, either by post, or by phone, just because they are 99/4A enthusiasts.

Texas Instruments gave no help at all, thus ensuring that their copyright wasn't infringed if he ever discovered how their 32K RAM worked!

Howard made a comment on TI ceasing production of the 4A, and said that his own view (which I share) was that TI had brought this sorry state of affairs on their own heads!

Their insistence that all solid state software is marketed by themselves has done no good at all. Even though all software was marketed by TI, much of it was not developed by TI.

Milton Bradley, for example, developed many of the modules, and even the MBX expansion system, which I will mention later.

Howard also tried to estimate how much TI were losing on each console. This was the time that 99/4A's had fallen to £100. He said that if the dealer is selling them for £100, he has to be making a profit on them, and so do the parasites at Customs and Excise. This meant that the consoles must have entered the country, costing not much more than £50.

Howard estimated that to produce one himself, using the same components, would have cost £80!!!

Howard went on to mention software suppliers that were still offering titles, such as Stainless Software, and Christine Computing.

In 1983, things were not very good on the hardware scene! Howard reported on what was available, which was mostly cassette leads, joysticks, joystick interfaces, and his 32K memory expansion.

One device that Howard said it would be impossible for him to manufacture was a disk controller card. RS232 cards were available in the states, and he must have been referring to CorComp or Myarc cards, because he said these would be available long after TI cards had ceased to circulate.

There were many more things that Howard wrote about, including the beginning of Tidings, and the very quiet launch of the original 99/4. Howard joined in the third edition after buying his 99/4A. He described TI as a very disorganized company who could do as they pleased because they were more or less a monopoly, and control most of the world's chip manufacture. After having a go at them, however, he commended them. He went through about four T.V.'s before discovering that his modulator was to blame. He took the 4A directly to Bedford himself, and TI fixed it for him while he waited! He even asked for a new keyboard to be fitted, because he didn't like the feel of

the one he had. He offered to pay for the keyboard, but forty minutes later, his baby came out of surgery, and he wasn't charged a penny for the keyboard! I wonder if it was Colin Hinson who carried out the operation!

Howard awarded a prize for dirty tricks to Thorn EMI. He saw River Rescue and Submarine Commander in a catalog, and quickly placed an order, and quoted the catalog numbers! Back came two Atari Cartridges!
He said, "there very helpful at EMI ordering department. If they think you've made a mistake, they correct it for you and send the item they know you really meant to order!" He explained to them that he really wanted, what he'd actually ordered, but they said that they had no record of those items on their computer! And I suppose Lee Harvey Oswald shot Kennedy by himself did he!

After phoning several EMI departments, he was given the excuse that the launch of the titles was being postponed until summer!

He phoned in Summer, but there were no results! An October release sheet came with the news that not only Submarine Commander and River Rescue were being released, but also `Computer War' which was based on the film `War Games'. He placed another order, and waited, and waited!

He then discovered that the launch was postponed until the 11th of November.
On the 11th of November he phoned the administration office to ask why there was no delivery. He found that there second manager had just quit the job, and he was given the statement "There are not, nor will there be any software from Thorn EMI for the 99/4A."

Well, despite EMI's efforts, the software has appeared on disk. I don't know how the software ever escaped, but I'm glad it did. River Rescue is excellent!

Howard closed his article with a section on selecting a printer for the 4A, which put an end to his typewriter! He opted for an unbranded printer in the end, after deciding against the four colour MCP40, which was the printer most used by Tandy and Oric owners.

Despite it being unbranded, he wasn't worried about any problems with it, since it used nearly all EPSON made parts.

Pages 14 and 15 were an advert for Stainless Software

Over 80 programs for the TI-99/4Am including:

Golf, Goblins Revenge, Braintwisters II,
Hang Glider Pilot (Import by Maple Leaf),
Walls and Bridges, Zombie Mambo, Planet Destroyer,
Global Rescue, Shuttle Command, etc. etc. etc....

Pages 16 and 17, Rambles by Stephen Shaw

This opened with the news of TI withdrawing the 99/4A. More than headline news for a few weeks wasn't it!!!

Stephen gave the news that State Troopers had to be called in to handle the riots caused by discount consoles!

Peripherals, and modules also had some nice price cuts. News again that EMI were not going to release any modules, but Funware modules were available from the International User Group, because they didn't have a U.K. source.

IMAGIC went out of business, but they just had time to release Demon Attack and Micro Surgeon.

Stephen agreed to sell cassette software as long as anyone wished to buy it.

He said that now, the only way to buy TI software was via mail order, and gave some very helpful tips on the best way to protect yourself from disappointment.
Always send an SAE to ensure that you get a reply from the address, before you send any money. When sending an SAE overseas, always use International Reply Coupons, which should be available from your Post Office. Quote your address clearly, beware of recorded delivery, because it has to be receipted, and could be undeliverable if no one is home when the first, and only, attempt to deliver the parcel is made.

<u>Pages 18 and 19</u>

A very BIG, two page, advert for PARCO ELECTRICS!!!!

<u>Page 18 appears again! The PARCO advert wasn't numbered!</u>

A continuation of Stephen Shaw's Rambles, with a review of Mr.Melfi's, **KIPPY's NIGHTMARE!!!**
Wow!!! It appears that there was a version for Extended BASIC with 32K expansion! That might be a loadable version!

In this game you control an animated figure who must avoid being caught by any of six monsters, and `poof' them back to their cages. After all of the monsters are dealt with, it's on to the next round.

The animation of the monsters is superb! Points are gained by lighting up dots in the play area (these also act as barriers to the monsters, but do not last for long!), by the length of time you survive, by placing one of your six `poofers' in the play area, and by `poofing' a monster.

The area of the playfield that your character covers, slowly gets smaller as time goes on!

The game is well over 4K and is entirely in machine code. Stephen says it appeared to exceed the 13K the usual tape load can handle.
He explains the strange loader! "First you use Easy Bug to load another tape loader. This loads the program at 1200 Baud (ie. faster than usual). Then this loader loads the actual game... and most of the game seems to reside in the VDP area!!!"

He awarded it 100% for playability and value for money.

Next, some news about cartridge problems which were being caused by a foam strip which is supposedly located in the cartridge port, and is supposed to clean the contacts of your cartridge as you insert it. My console's not got one! Also, more news on reports of some faulty modulators and keyboards, a slight bug in LOGO 2 which meant that MakeShape which worked in a defined procedure in LOGO 1, could only be used in direct mode in LOGO 2.
Stephen said it was also interesting to see that LOGO 2 was copyrighted 1984, even though it was released in 1983. He said that possibly, under U.S. law, this could render it public domain. This is a technicality that means the first series of STAR TREK is also not copyright!!

You can make as many video copies as you like, but the name STAR TREK, is also a trademark!

On the next page, Stephen went through describing the graphics modes of the 9918A/9929 VDP, and explained that writing programs that drew with individual pixels were impossible without machine code, but now we've got the Missing Link and XHI etc.

Pages 21, and 22 were a couple of puzzling things to do with sprites in Extended BASIC, such as deleted sprites being deleted, and then being recreated and remembering their old motion settings!

Pages 23 and 24 were about a book of fifty puzzles, some of which Stephen included. The idea of the book was that it gave the reader the puzzles, which then required a computer program to solve them. Definitely not one for IBM owners!

There was also a GIANT BOOK OF COMPUTER GAMES, with hints on converting them to the 4A.
Stephen also managed to advertise his own book, **"Getting Started with the TEXAS TI-99/4A"**.

Pages 25 to 27 BEGINNING PROGRAMMING

By Ian Godman of Christine Computing.
Presenting a technique for structured programming, that's understandable, and usable.
The two rules he gives are the ones that I use.

1. Think - don't rush.
2. Switch off computer.

When I was writing Workspace, I didn't even have my second GENEVE! I did it all on paper, in GREAT detail, which Mark Wills will tell you, since I went through my programming notes with him when he stayed at my place.

I only had chance, every couple of months, to test it out on Gary's GENEVE.

Pages 28 to 30 The Pete Brooks Pages

I'm surprised Peter felt like writing this. He said he had just finished writing his book, MASTERING THE TI-99!

As I read through the article, I was very surprised! He says he regretted buying his TI. He paid £500+ for his system, and only stuck with it because he couldn't afford to do anything else! I never thought I'd hear anything like that from Peter Brooks. He says he felt compelled to help people who had bought the 4A as a first machine, and I can remember it was a great help when I phoned him in the mid 80's when things seemed to be drying up for me.

PARCO had gone, and most of the software I had got, I had written myself in Extended BASIC.
Without joining TI-LINES, I would have never discovered the Bloxwich Workshops, Richard Sierakowski, Gordon Pitt, or I might not even have a GENEVE today.

If I had stopped using my system, I would not have been so enthusiastic about it when starting college, and Norman would probably never mentioned Gary having a 4A, and I might not have met him or Trevor, and I might not even be writing this article today.

However, if TI had got there marketing act together, today I could be one of there top programmers, writing for whatever chip would be the current world talking point. Something like the TMS999032 with 32, 32-bit registers in a Workspace! It makes me want to visit America and steal the UFO that the U.S. government has posession of which can bend time and space.

I could take my system back in time, and my back issues of TI*MES, and show TI how daft they are going to be in the early 80's!!! Technology would be alot more advanced today than it currently is. If the 9900 had sold in the same numbers as the sad intel 80#86's, then TI would have all that extra development money to play with, and think where we would be nowadays.

OOOPS! Got carried away again! Back to the Peter Brooks pages!

Peter discussed a comparison of the BBC to the 4A, and I didn't like his views. He said that the graphics and sound of the BBC were more sophisticated! No way! I'd like a BBC to come out with the same quality of sound that Sound FX manages, and I'd like to see it throw some shapes around the screen as fast as we can handle sprites!

We beat the BBC completely on every count. The BBC was advertised as having 32K RAM, which is crap, since you have to deduct your screen display from it. The 4A has totally independant VDP RAM, which means we can just plug in a new VDP and upgrade the graphics output totally, to add more colours, and more VDP RAM. As I've alread stated, Acorn had to totally redesign the sad thing, just because the BBC wanted to give the machine more colours, and a Genlock facility! Sadness of the highest order! The 4A had Genlock as standard on the 9918A/9929 chip!

To take time to also shut Apple II owners up, someone produced a card to expand the graphics capabilities of the Apple II, which contained a more advanced VDP. This VDP was the TMS9929.

Peter also reported in his article that a small group

in the U.K. were converting Spectrum peripherals for use with the 4A! This is the first I've heard of this, so it looks like nothing ever happened on the project, and nothing was released, although, it sounds like a quite sexy thing to do, if only you could still get Spectrum peripherals!

It might be interesting, just as a laugh, to design something that allowed Amiga peripherals to be used, or even PCMIA credit-card-slot peripherals. The future still holds alot of possibilities!

He closed with a quick mention of undocumented commands for use with the STATS or PRK cartridges in the GROM port. If you have these cartridges plugged in, then the following calls are available from TI-BASIC:

CALL A, CALL D, CALL G, CALL H, CALL L, CALL P, CALL S.

He says no one has ever written a definitive article on these, so he couldn't say what they do! I knew that PRK had the hidden commands, but not STATS. I've got the STATS cartridge, so I'll have an experiment with them when I find time!

The bottom half of page 30 was a description of a BASIC program that allowed pseudo-single-pixel-plotting!

A listing for this was included on page 31.

Page 32 ARCADE

Games and High Scores for the under 18's.

In 1983 the world record scores were:

ALPINER	94,599
THE ATTACK	1,318,450
CAR WARS	97,380
CHISHOLM TRAIL	603,400
DRIVING DEMON	12,348
HENHOUSE	132,570
TI INVADERS	46,192
MUNCHMAN	408,990
PARSEC	3,492,400
PINBALL (Video Games 1)	10,028,010
RABBIT TRAIL	23,200

Page 33 Video Game Scene

This is by Howard Greenberg, and is continued from the first issue of TI*MES. He missed the second issue, because he was trying to debug the 32K expansion.

This is about the video game industry in general, due to Howard's involvment with the Amusement Arcade industry, which gave him the knowledge to convert joysticks from the arcade machines, to work on his 4A!

He reported on the release of Atari's pole position, and also the American Amusement Machine Exhibition, at which, most of the games seemed to be based on laser disk.

He made the complaint that the industry seemed to be going more towards laser disks, and that it was sad because

the technology was already available for the 4A in the form of the video controller card!

He said that it would not be available in the U.K. but failed to mention that this was because it was only for NTSC, and if it did arrive in the U.K., it would cost about £500!!!

A comment from Howard was that he didn't know where video games were going next, and his own view was that they were all but finished. In 1983, the video games market shrank by 80%.

If only his statement was correct. Children would be getting the benefit of a 4A console for Christmas this year, and not a Commodore 64, or a sad SEGA or NINTENDO effort!

Page 34 SOFTWARE REVIEWS

CASSFILE, A filing system for cassette and TI-BASIC from Christine Computing.

DIYAD, An adventure game writers toolkit from Timeless Software in Scotland.

CHARFILE, From Christine Computing again, this was a character designing program, that stored all of your defined characters on cassette in a sort of a graphics character library.

TOAD GRAPHICS, A very interesting sounding graphics program that allowed TI BASIC to operate as a mini version of logo and draw shapes on screen. It was by Timeless Software, and had a very good review.

I wouldn't mind seeing this. Does anyone out there still have it?

Page 35

Classified Ads, and news of a programming competition. For sale in 1983 was a Speech Synthesizer for £27, and Tombstone City, and The Attack for £11 each.
New products being sold were dust covers for £3.25, tape head demagnitizers for £2.25, and Aerial Splitters for £2.25.

Stephen Shaw's "Getting Started with the 99/4A" was selling for £4.95.

The programming competition was to produce the best Educational program, or the best Business program, or the best Arcade/Fun program. Either TI-BASIC or Extended BASIC programs were allowed.

Page 36

This was an advert for Arcade Hardware, which included the 32K RAM expansion for £125, and the Arcade Machine Joystick for £22. A full picture of the 4A, showing both plugged in was shown.

Inside the back cover was an advert for A.B.M. Control from Timeless Software, which included a screen shot of the game. You had to defend an air base from enemy aircraft.

On the back cover, a classic was born! Timeless Software advertised their new imported title: **DIABLO.** A game in which you've got to move tiles containing a track, and manouvre a ball around the screen. Written entirely in Extended BASIC. It is one of those ideas that works excellently in XB with sprites. I'm sure there is the possibility of a few more games like this in the future.

Well, that's it, I've reviewed the first three issues of TI*MES, and we have officially survived our first decade out of production, with virtually no help from TI. Let's all try and stick together for our even better decade to come. I hope to be covering this article in December 2003, and will no doubt say, if only I knew then what I am going to know in 2003! Hopefully my children will be playing with educational programs on the console by then, while I'm writing my articles on the GENEVE!

Mark Wills and myself with be 33!!! Don't worry Mark, you can call me if you like!

I can't leave this article though, without a quick moan. In issue 41b of TI*MES there was an article on the MBX expansion, saying that it was never officially released. Well, how come PARCO were selling them, and how come I've got one, still with its box, in MINT condition.

There was also something about it not working with the TI Speech Synthesizer! You can leave them plugged in at the same time. If you run a normal TI program with speech, it uses the TI Speech Synthesizer. If you run an MB cartridge, it detects the MBX expansion, and uses the TI Speech Chip which is built into that (Yes the MBX includes the same speech chip!).

If you use an MB cartridge such as Sewer Mania, or Space Bandits, but have the MBX turned off, then they will also use the TI Speech Synthesizer. One thing that won't run without MBX is Championship Baseball, which is an excellent TWO player game. It is more or less unplayable without speech, and definitely needs the keypad. Every field position can be selected from the keypad, and there's an overlay provided.

The only thing that the MBX gets in the way of is the cassette and joystick. It plugs into both of these, but if you do all of your loading from disk, then you can leave it plugged into the cassette port, and just swap the joystick port when the MBX is not required.

That's it! I've dragged on too far now. I'll have complaints that I've written too much! Gary has had a complaint that there are too many pictures, and not enough articles, and this VERY long article will settle that score, but I do like to see alot of pictures in the magazine, as they brighten it up alot.

Oh well, never mind. If you want to see more articles in the magazine, then write more! That's my motto!
A closing phrase:

The Texas Instruments TI-99/4A. Not a home computer, but a computer for the home.

Merry Christmas, and a happy new decade!!!

THE END? Not Quite!

Wot?
No
DRIVEL!!

No!! Unfortunately, Mark's system is down again, and he's currently between jobs. He has also moved back to Shrewsbury.
He has asked me to print this little apology for him, and said he hopes to get something in for the Spring edition.

Don't worry! We'll see that he does!

That's it for now.

Keep
on
TIing!

Just when you thought you weren't going to see anything of our game that we were writing!

Unfortunately, time has beaten us again, and this is all we have:

```
1 ! BUG    DIGGER!
2 CALL SCREEN(4)
20 DIM CP(300,3)
50 CALL COLOR(9,2,4,10,2,4,11,2,4)
90 CALL CHAR(120,"3838107CBABA286C"):: CALL
   COLOR(12,5,4):: CALL CHAR(128,"EFEF0 0BDBD00EEEE"):: 
   CALL COLOR(13,10,16)
91 CALL CHAR(33,"628549381C92A146"):: CALL COLOR(1,16,4)
100 CALL CHAR(100,"00000008000000000000000808000000000
    0000818000000000001818")
110 CALL CHAR(104,"0000081C1C0000000000081C1C0800000000
    183C3C1800000000387C7C38" )
120 CALL CHAR(108,"00387C7C7C38000000387C7C7C7C3800003C7E
    7E7E7E3C003C7EFFFFFFFF7E3C"):: CALL
    CHAR(112,RPT$("F",16))
450 MC=20 :: CR=1 :: CALL CLEAR :: FOR D=1 TO MC ::
    R=INT(RND*19)+4 :: C=INT(RND *29)+2 :: RANDOMIZE ::
    CP(D,1)=R :: CP(D,2)=C :: CALL HCHAR(R,C,33):: NEXT D
451 BLOKE=120
500 CALL VCHAR(3,1,128,21):: CALL VCHAR(3,32,128,21)::
    CALL HCHAR(24,1,128,32):: R=15 :: C=15
700 MCP=1
990 CALL JOYST(1,X,Y):: C2=C+X/4 :: R2=R+Y/4*-1 :: CALL
    GCHAR(R2,C2,X):: IF X=32 THEN CALL HCHAR(R,C,32)::
    CALL HCHAR(R2,C2,BLOKE):: R=R2 :: C=C2
1001 CALL KEY(1,K,S):: IF S=0 THEN 1009 ELSE CALL
     JOYST(1,X,Y):: R2=R+Y/4*-1 :: C2=C+X/4 :: CALL
     GCHAR(R2,C2,X):: IF X>99 AND X<112 THEN CALL
     HCHAR(R2,C2,X+1):: GOTO 1001
1002 IF X=32 THEN CALL HCHAR(R2,C2,100):: GOTO 1001
1009 IF CR>MC THEN CR=1
1010 IF CP(CR,3)=13 THEN CR=CR+1
1011 IF CP(CR,3)<>0 THEN CP(CR,3)=CP(CR,3)-1 :: CR=CR+1
     :: GOTO 1009 1012 IF R>CP(CR,1)THEN NCR=CP(CR,1)+1
     ELSE IF R=CP(CR,1)THEN 1013 ELSE NCR=CP(CR ,1)-1
1013 IF C>CP(CR,2)THEN NCC=CP(CR,2)+1 ELSE IF
     C=CP(CR,2)THEN 1014 ELSE NCC=CP(CR ,2)-1
1014 CALL GCHAR(NCR,NCC,X):: IF X=32 THEN CALL
     HCHAR(CP(CR,1),CP(CR,2),32):: CALL
     HCHAR(NCR,NCC,33):: CP(CR,1)=NCR :: CP(CR,2)=NCC ::
     GOTO 1030
1016 IF X=97 OR X=120 THEN 10000
1030 CALL CHAR(33,"628549381C92A146"):: CALL
     CHAR(33,"C14324181824C283"):: CALL
     CHAR(33,"46A1921C38498562"):: CALL
     CHAR(33,"C14324181824C283"):: CR=CR+1 :: GOTO 990
5000 CALL SAY("GOT ONE"):: CALL
     HCHAR(CP(CR,1),CP(CR,2),32):: GOTO 990
10000 CALL SAY("_"):: GOTO 990
```

And that's really **is** it now!!!!

Richard Twyning, well and truly signing off...

Really THE END!!

MODULE LIBRARY

TITLE	QTY in STOCK	PRICE
ADDITION AND SUBTRACTION 1	3	3.00
ADDITION AND SUBTRACTION 2	1	3.00
ADVENTURE COMPLETE WITH PIRATE TAPE	3	5.00
ADVENTURE MODULE	4	3.50
BEGINNING GRAMMAR	7	3.00
BLACKJACK AND POKER	1	3.00
BUCK ROGERS	1	4.50
CONNECT FOUR	3	3.50
DISK MANAGER*	3	2.00
DISK MANAGER 2*	1	3.50
DIVISION 1	1	3.00
DEFENDER	1	4.50
DONKEY KONG	1	4.00
EARLY LEARNING FUN	1	3.00
EARLY READING	1	3.00
EXTENDED BASIC INC MANUAL	3	22.50
EXTENDED BASIC MODULE	4	15.00
HOUSEHOLD BUDGET MANAGEMENT	1	3.50
HUSTLE	1	3.00
INDOOR SOCCER	2	4.00
MICROSURGEON	1	4.00
MINI MEMORY + LINE ASSEMBLER + MANUAL	1	15.00
MULTIPLICATION 1	1	3.00
MUNCH MAN	2	3.50
NUMBER MAGIC	2	3.50
NUMERATION 1	1	3.00
NUMERATION 2	1	3.00
PARSEC	2	4.00
PROTECTOR	1	4.50
STAR TRECK	1	4.50
SUPER EXTENDED BASIC (TRITON) + MANUAL	1	45.00
TERMINAL EMULATOR II	3	5.00
TI INVADERS	1	4.00
TI LOGO + ORIG. FOLDER WITH MANUAL	1	15.00
TI LOGO II + ORIG. FOLDER WITH MANUAL	1	25.00
TI WRITER + ORIG. FOLDER WITH MANUAL*	1	30.00
YAHTZEE	2	3.00
FOR SALE ALSO 32K SUPERCART	2	25.00

NUMBER PUZZLES - OLD & NEW

Walter Allum

In TI*MES Issue 41, I mentioned a difficult puzzle that haunted us in the sixties and seventies and I promised to give you the solution. Here goes: there isn't one! I apologize to anybody who feels tricked into spending time on a lost cause but, really, none of these problems (not even the famous Last Theorem,lately in the news again) has any value in itself. It's what you achieve while trying to get there that matters.

Whether you favour computer search or high math, I think the way in is by noting that the next number (were it to exist) would be of the form $n\wedge 2-1$ where n is an integer (from the product with 1). Likewise, from its product with 3, we need $3*(n\wedge 2-1)=m\wedge 2-1$ where m is another integer. Recast for convenience as: $m\wedge 2-3*n\wedge 2=-2$ .....(i) From its products with 8 and 120, we require: $k\wedge 2-8*n\wedge 2=-7$ .....(ii) and $h\wedge 2-120*n\wedge 2=-119$ ....(iii) where k and h are integers. These equations are all of Pell type and individually can be solved by methods known for centuries. I prefer "continued fractions", a process involving quite a bit of arithmetic but easily managed on the TI. However, simultaneous Pells are a different ballgame!

It helps to observe that, unless eqns.(i) and (ii) have at least one common n solution larger than the n=11 that gives the fourth number in the sequence, there is no point in involving eqn.(iii). Baker & Davenport (Quart.j.Math. 1969 vol.20 p.129) knew that (i) and (ii) could have only a finite number of common solutions and,by advanced math arguments, showed that the two already known (1,11) were the lot. They had some assistance from a computer in establishing that certain conditions held good -- multi-length arithmetic was required to 600-1040 places.

So, one may ask how far you could have got against such odds. A reasonable target might be to seek new numbers $< 10\wedge 10$ i.e. such that can be checked and fully printed by the TI's normal facilities.

Assessing every number from 121 upwards is clearly a tactic of despair and we must presume that, at the very least, only numbers of form $n\wedge 2-1$ are considered i.e we concentrate on eqns.(i) and (ii). Taking (i) for example, we could evaluate $3*n\wedge 2-2$ for all n from 11 to about $10\wedge 5$ and note which n made this a perfect square Likewise with (ii). Look for n appearing in both streams. To try this out,I wrote a simple program that scanned up to n=1000 in about 3min but I didn't wait the $4\frac{1}{2}$ hours or so to reach n=100000. You could get such a program to identify and print all the common n values (none!) but then you would miss the chance of spotting that the first solutions of (i) are n=1,3,11,41,153,571 and of (ii) are 1,2,11,64,373 interlaced with 1,4,23,134,781 and then realizing, as I joyfully did all those years ago, that the (i) solutions obey the recurrence:N(j+2)=4*N(j+1)-N(j) and the separate sequences the same but with 6 instead of 4. With this knowledge, you can surge up the sequences of n in a few seconds (with a TI, of course). Another idea, more perspiration than inspiration, is to look in the Number Theory books and write a continued fraction program.This would soon give you the few tens of n solutions that you need to place the "fifth number" beyond $10\wedge 10$ but there's a lot of coding to do.

For light relief now, here is another puzzle. Fear not!--it has a solution. You might meet Pell on the way but simple procedures will also do.

> "A crowd of some hundreds turned up for a charity race and were given numbers starting at 1 in the order of their checking in. Bill Brainbox noticed that the sum of the numbers issued before his exactly equalled

the sum of those used after. How many ran and what was Bill's number?"

Wreaking a variation like having M*afters = N*befores (M,N integers) toughens everything up. Except for the triviality M=1, N=3, I haven't found a single solution.

When I can find time, I'll clean up my currently user-unfriendly Continued Fractions/Pell program and put it in our Library. Bet you can hardly wait!

```
100 !LINEWRITER to aid in formatting screen text into 28-column format
and saving it as DATA program lines in MERGE format - by Jim Peterson
110 !strings containing commas, leading blanks, quotes will be ACCEPTed,
 and converted to DATA statements which RUN correctly even though they
120 !are not enclosed  in quotation marks!
130 ! Type, center, hyphenate etc. each line as you want it, and Enter.
 For a blank line, just Enter. When finished, type @@@, Enter
140 !When READY, type NEW, then load program which is to have the text
 displayed. Type MERGE DSK1.LINEFILE, Enter.
150 !Add program line where text display is wanted -    RESTORE 30000
 :: FOR R=1 TO (number of lines) :: READ M$ :: DISPLAY AT(R,1):M$ :: NEXT R
160 CALL CLEAR :: OPEN #1:"DSK1.LINEFILE",VARIABLE 163 :: LN=30000
170 FOR R=1 TO 24 :: DISPLAY AT(R,1)SIZE(1):" " :: ACCEPT AT(R,0)SIZE(-28):A$
:: IF A$="@@@" THEN 210 :: B$=B$&CHR$(200)&CHR$(LEN(A$))&A$
180 X=X+1 :: IF X/4=INT(X/4)THEN 190 ELSE B$=B$&CHR$(179) :: GOTO 200
190 GOSUB 240 :: LN=LN+10
200 NEXT R :: X=0 :: CALL CLEAR :: GOTO 170
210 IF B$="" THEN 230 :: IF SEG$(B$,LEN(B$),1)=CHR$(179)THEN
B$=SEG$(B$,1,LEN(B$)-1)
220 GOSUB 240
230 PRINT #1:CHR$(255)&CHR$(255) :: CLOSE #1 :: END
240 PRINT #1:CHR$(INT(LN/256))&CHR$(LN-256*INT(LN/256))&CHR$(147)&B$&CHR$(0)
 :: B$=NUL$ :: RETURN
```

TI-99/4A USERS GROUP (UK)

MEMBERSHIP NEWS

by Alasdair Bryce

A Happy New Year to one and all.

Since the last issue of TI*MES our numbers have been swelled by the arrival of another three new members. A warm welcome goes to Jeff Kuhlmann, David Milloy and Pat de Roe.

With regard to those of you who are due to renew after this issue of TI*MES but who choose not to do so I would be grateful if you would make use of the pre-paid envelope provided to let me know this so that you won't be bothered with subsequent reminder letters. In addition there is always a market for TI equipment and I often get enquiries from those wishing to expand their systems so it may useful for you to let me know whether you wish to sell your TI hardware if you are moving on to another machine.

For anyone interested the following back issues are still available at £1 each plus P&P: numbers 4,8,9,10.11,13,14,16.18,19,20.22.24.25.27.29 and 32-42. If specifically requested copies of articles from other issues can be supplied for the cost of the photocopying.

Editor's Comments

In the previous issues "Editor's Comments" it was stated that the articles included, appear to be written by the same group of people. This is also true of the current issue, with certain people writing more than others!

We want to see more articles for everyone, and if not, then let's have some input for the Consoletation Zone, for which Trevor hasn't had any replies.

So, if you have any problems, complaints, or whatever, then let's hear it!

We have had a complaint that there are too many pictures and not enough articles, but pictures do help to brighten up the magazine, however, we will try and replace a few blank spaces with little bits of information.

Next Copy Date

The next date for copy for the Spring 1994 issue will be the 1st of March 1994. We have also decided to stick to a set of official copy dates.

1st of March for the Spring issue

1st of June for the Summer issue

1st of September for the Autumn issue

1st of December for the Winter issue

A problem everyone has at one time or another is deciding how a particular program loads. The following are aids from Bruce Carson (user group unknown) and Ken Smith (Great Lakes U.G.). The lists are not quite complete, since so many programs have appeared after they were originally published and many use file types not covered here, but they are a good beginning. The two articles complement each other and have been set up to be folded in half to make a quick reference card.

SIMPLE RULES FOR LOADING PROGRAMS

I. Catalog the disk.

II. Look at titles, if name is:

UTIL1 - Use E/A opt.5 or TI Writer opt.3. Ignore filename request and press enter. Program will autoload (from drive 1 only).

LOAD - Use Extended Basic program will autoload.

xxxDOC, README, or HELP - documentation or instructions. Usually prints through TI Writer.

III. Look at file types:

DIS/FIX - E/A opt.3 DSK1.NAME

DIS/VAR 80 - TI Writer.

DIS/VAR 163 - Extended Basic (also used for merge format).

PROGRAM - Extended Basic, Basic, E/A opt.5 or TI Writer opt.3. try them in that order. Some artist programs also use this format.

INT/FIX - Ususally a file loaded from an Extended Basic program.

DIS/FIX 128 or INT/FIX 128 - may be an Archived file. Un-arc with Barry Boones' Archiver program.

LOAD LIST

PROGRAM	SIZE	D/V	D/F	I/V	I/F
BASIC.....	<51	...	...	...	...
XBASIC.....	<45	163	...	...	...
XB (XMEM).....	>45	...	...	254	...
E/A 5 RUN.....	<34	...	...	...	...
TIWRITER 3 UTIL	<34	...	...	...	...
RLE.....	54	...	...	...	...
GRAPHX.....	54	...	...	...	...
DRAWABIT.....	54	...	...	...	...
DRAWABIT II....	57	...	...	...	...
DRAWNPLOT.....	25	...	...	...	...
TIARTIST.....	25	...	...	...	...
XB + MISNG LNK.	ANY	...	...	254	...

FILE					
XB (MERGE).....	...	163	...	...	...
DATABASE.....	...	163	...	...	...
TI WRITER.....	...	80	...	...	...
E/A 1 EDIT.....	...	80	...	...	...
TI ARTIST.....	...	80	...	...	...
INIT:LOAD:LINK.	...	80	...	...	...
MCFLIX.....	...	...	128	...	...
E/A 3 LOAD+RUN.	...	...	80	...	...
CSGD.....	...	...	...	254	...
JOYPAINT.....	...	...	...	...	128
SQUEEZED.....	...	...	...	...	128
ARCHIVED.....	...	...	128	...	...
INFOCOM ADV....	...	...	255	...	...
S.A. ADVENTURE.	<55	...	...	...	...
TUN. of DOOM...	<53	...	...	...	...

1. Catalog disk
2. Read directory file data
3. Select load environment from table

notes
PROGRAM can be data- eg CHARA1, or MusicPro data files.

notes
RLE also DF128 and DV80.
PAGE PRO...IF128. SFX....DF128
TIPS...IF53 and IF16 MYART..DF128

TIPS FROM
THE
TIGERCUB
No. 14

[This text file was received AFTER it had been transliterated. I don't know what had been replaced with what but have done what I can. Any silly characters left in are my fault for not spotting them!! sjs]

Here's a little something for you who own the Terminal Emulator II and the Speech Synthesizer. Maybe our Congressmen could use it to help them discuss the national debt.

```
100 CALL CLEAR
110 PRINT TAB(7);"NUMBER SPE
AKER": : :"by Jim Peterson":
"       of Tigercub Software"
: : :
120 PRINT " This program wil
l print any":" number of les
s than 67":"digits in number
s and in"
130 PRINT "words, and will s
peak the":"words.": : : :" R
equires Terminal Emulator":"
II and Speech Synthesizer.":
 : :
140 CALL CHAR(39,"0000000000
301020")
150 OPEN #1:"SPEECH",OUTPUT
160 DIM HIGH$(21),NN$(23)
170 DATA ONE,TWO,THREE,FOUR,
FIVE,SIX,SEVEN,EIGHT,NINE
180 DATA TEN,ELEVEN,TWELVE,T
HIRTEEN,FOURTEEN,FIFTEEN,SIX
TEEN,SEVENTEEN,EIGHTEEN,NINE
TEEN
190 DATA TWENTY,THIRTY,FORTY
,FIFTY,SIXTY,SEVENTY,EIGHTY,
NINETY
200 DATA THOUSAND,MILLION,BI
LLION,TRILLION,QUADRILLION,Q
UINTILLION,SEXTILLION,SEPTIL
LION,OCTILLION,NONILLION
210 DATA DECILLION,UNDECILLI
ON,DUODECILLION,TREDECILLION
,QUATTUORDECILLION,QUINDECIL
LION,SEXTEDECILLION
220 DATA SEPTENDECILLION,OCT
ODECILLION,NOVEMDECILLION,VI
GINTILLION
230 FOR J=1 TO 9
240 READ ONE$(J)
250 NEXT J
260 FOR J=1 TO 10
270 READ TEEN$(J)
280 NEXT J
290 FOR J=1 TO 8
300 READ TEN$(J)
310 NEXT J
320 FOR J=1 TO 21
330 READ HIGH$(J)
340 NEXT J
350 PRINT : : :
360 PRINT #1:"NUMBER"
370 INPUT "NUMBER? ":N$
380 L=LEN(N$)
390 FOR J=1 TO L
400 IF POS("0123456789",SEG$
(N$,J,1),1)=0 THEN 360
410 NEXT J
420 IF (VAL(N$)<1)+(VAL(N$)<
>INT(VAL(N$)))THEN 360
430 IF L<67 THEN 470
440 PRINT "HEY! I CAN ONLY C
OUNT TO A":"VIGINTILLION!":
:
450 PRINT #1:"HAY I CAN ONLY
 COUNT TO A VIGINTILLION"
460 GOTO 360
470 IF VAL(N$)>0 THEN 510
480 PRINT : :"ZERO": :
490 PRINT #1:"ZERO"
500 GOTO 360
510 IF L/3=INT(L/3)THEN 540
520 N$="0"&N$
530 GOTO 380
540 X=L/3
550 FOR J=1 TO L STEP 3
560 JJ=JJ+1
570 NN$(JJ)=SEG$(N$,J,3)
580 IF J>1 THEN 610
590 P$=STR$(VAL(NN$(JJ)))
600 GOTO 620
610 P$=P$&"'"&NN$(JJ)
620 NEXT J
630 PRINT : : :P$: : :
640 FOR J=1 TO X
650 GOSUB 670
660 GOTO 1150
670 IF VAL(NN$(J))<>0 THEN 7
10
680 A$=""
690 FLAG=1
700 GOTO 1140
710 FLAG=0
720 H=VAL(SEG$(NN$(J),1,1))
730 T=VAL(SEG$(NN$(J),2,2))
740 TT=VAL(SEG$(NN$(J),2,1))
-1
750 VV=VAL(SEG$(NN$(J),3,1))
760 IF T=0 THEN 1000
770 IF T>9 THEN 810
780 A$=ONE$(T)
790 SP$=A$
```

```
800 GOTO 1000
810 IF T>19 THEN 880
820 A$=TEEN$(T-9)
830 IF T<>19 THEN 860
840 SP$="NINE TEEN"
850 GOTO 1000
860 SP$=A$
870 GOTO 1000
880 IF VV<>0 THEN 950
890 A$=TEN$(TT)
900 IF TT<>8 THEN 930
910 SP$="NINE TEE"
920 GOTO 1000
930 SP$=A$
940 GOTO 1000
950 A$=TEN$(TT)&"-"&ONE$(VV)
960 IF TT<>8 THEN 990
970 SP$="NINE TEE"&ONE$(VV)
980 GOTO 1000
990 SP$=A$
1000 IF H=0 THEN 1080
1010 IF T=0 THEN 1050
1020 A$=ONE$(H)&" HUNDRED &
"&A$
1030 SP$=ONE$(H)&" HUNDRED &
"&SP$
1040 GOTO 1140
1050 A$=ONE$(H)&" HUNDRED"
1060 SP$=A$
1070 GOTO 1140
1080 IF (J<X)+(T=0)+(VAL(N$)
<100)THEN 1140
1090 A$=" & "&A$
1100 IF (TT<>8)*(T<>19)THEN
1130
1110 SP$=" & "&SP$
1120 GOTO 1140
1130 SP$=A$
1140 RETURN
1150 PRINT A$
1160 IF FLAG=1 THEN 1200
1170 PRINT #1:SP$
1180 PRINT HIGH$(X-J)
1190 PRINT #1:HIGH$(X-J)
1200 GOSUB 670
1210 NEXT J
1220 PRINT B$
1230 A$=""
1240 JJ=0
1250 B$=""
1260 P$=""
1270 FOR D=1 TO 500
1280 NEXT D
1290 GOTO 350
```

==========

And, somebody might get mad if I don't include a little music -

```
100 REM - BELL MUSIC program
med by Jim Peterson
110 CALL CLEAR :: CALL SCREE
N(5):: RANDOMIZE
120 FOR CH=96 TO 136 STEP 4
:: FOR L=1 TO 4 :: X$=SEG$("
0018243C425A667E8199A5BDC3DB
E7FF",INT(16*RND+1)*2-1,2)::
 B$=B$&X$ :: C$=X$&C$ :: NEX
T L
130 D$=B$&C$ :: Z$=RPT$(D$,4
)
140 CALL CHAR(CH,Z$):: B$,C$
,Z$=NUL$ :: CALL MAGNIFY(4):
: CALL SPRITE(#CH/4-23,CH,IN
T(15*RND+2),255,255):: NEXT
CH
142 FOR J=1 TO 10 STEP 2 ::
X=9*RND-9*RND :: Y=9*RND-9*R
ND :: CALL MOTION(#J,X,Y,#J+
1,X,Y):: NEXT J
150 FOR J=1 TO 20
155 CALL COLOR(#INT(10*RND+1
),INT(15*RND+2))
160 FOR V=0 TO 16 STEP 4
170 ON J GOSUB 250,270,290,3
10,330,350,370,390,410,430,4
10,390,370,350,330,310,290,2
70,250,270,290,310,330,350
180 NEXT V
190 READ X
200 FOR D=1 TO X*5
210 NEXT D
220 NEXT J
230 RESTORE
240 GOTO 150
250 CALL SOUND(-999,131,V,52
3,V,131*15/2,30,-4,V)
260 RETURN
270 CALL SOUND(-999,165,V,16
7,V)
280 RETURN
290 CALL SOUND(-999,196,V,19
9,V)
300 RETURN
310 CALL SOUND(-999,262,V,26
5,V)
320 RETURN
330 CALL SOUND(-999,330,V,33
3,V)
340 RETURN
350 CALL SOUND(-999,392,V,39
4,V)
360 RETURN
370 CALL SOUND(-999,523,V,39
2,V,330,V)
380 RETURN
390 CALL SOUND(-999,659,V,66
6,V)
400 RETURN
410 CALL SOUND(-999,784,V,79
2,V)
420 RETURN
430 CALL SOUND(-999,1047,V,1
```

```
057,V)
440 RETURN
450 DATA 16,16,2,16,8,16,4,4
,16,2,16,4,16,8,8,16,2,2,16,
4,2,8,16
```

TIPS FROM THE TIGERCUB
#15

Several people have sent me enhancements to my Menu Loader, and I greatly appreciate them. The trouble is, if I incorporated them all the program would take up about 25 disk sectors! So, I have borrowed some ideas, added a few of my own, and here is the result.

It will list and load up to 99 programs, stopping at the end of every screenfull or stopping whenever any key is pressed and then offering you the choice of loading, deleting or quitting. It will ask you to verify a deletion by name before deleting it, and will display the name of the program it is loading. It also contains a feature to warn you if you are getting a bad count of disk sectors used - which I find happening more often than you might realize.

```
100 !by A. Kludge/M. Gordon/
T. Boisseau/J. Peterson/etc.
110 CALL CLEAR :: CALL INIT
:: CALL LOAD(8196,63,248)::
CALL LOAD(16376,67,85,82,83,
79,82,48,8)
120 CALL LOAD(12288,129,195,
126,165,129,153,102,60)
130 CALL LOAD(12296,2,0,3,24
0,2,1,48,0,2,2,0,8,4,32,32,3
6,4,91):: CALL LINK("CURSOR"
)
140 CALL CLEAR :: CALL SCREE
N(5):: FOR S=1 TO 14 :: CALL
 COLOR(S,7,16):: NEXT S :: C
ALL VCHAR(1,31,1,96):: CALL
COLOR(0,2,16)
150 OPTION BASE 1 :: DIM PG$
(99),T$(5)
160 T$(1)="dis/fix" :: T$(2)
="dis/var" :: T$(3)="int/fix
" :: T$(4)="int/var" :: T$(5
)="program"
170 IMAGE ##
180 DISPLAY AT(1,9):"DISKETT
E MENU"
190 ! IF YOU HAVE MORE THAN
ONE DISK DRIVE, DELETE THE !
 IN LINE 200 AND THE FIRST S
TATEMENT IN 210
200 ! DISPLAY AT(12,6):"DISK
? (1-3):" :: ACCEPT AT(12,19
)SIZE(-1)VALIDATE("123"):D$
:: D$="DSK"&D$&"."
210 D$="DSK1." :: OPEN #1:D$
,INPUT ,RELATIVE,INTERNAL ::
 INPUT #1:N$,A,J,K :: DISPLA
Y AT(1,2):SEG$(D$,1,4)&" - D
iskname= "&N$;
220 DISPLAY AT(2,2):"Availab
le=";K;"Used=";J-K:" Prog Fi
lename  Size    Type":"----
---------- ---- -------" ::
I,VT=0 :: TT=J-K
230 FOR X=1 TO 99 :: IF X/20
<>INT(X/20)THEN 260
240 DISPLAY AT(24,1):"Type c
hoice or 99 for more" :: ACC
EPT AT(24,27)VALIDATE(DIGIT)
:K :: IF K=99 THEN 250 :: IF
 K>0 AND K<NN+1 THEN 420 ELS
E 240
250 X=1
260 I=I+1 :: IF I>127 THEN K
=X :: GOTO 360
270 INPUT #1:P$,A,J,B :: NN=
NN+1
280 IF LEN(P$)=0 THEN 320
290 DISPLAY AT(X+4,2):USING
170:NN :: DISPLAY AT(X+4,6):
P$ :: PG$(NN)=P$ :: DISPLAY
AT(X+4,18):USING 170:J :: DI
SPLAY AT(X+4,22):T$(ABS(A)):
: VT=VT+J
300 CALL KEY(0,KK,ST):: IF S
T=0 THEN 310 :: FLAG=1 :: GO
TO 320
310 NEXT X
320 DISPLAY AT(X+4,1):" " ::
 DISPLAY AT(X+4,2):USING 170
:NN :: DISPLAY AT(X+4,6):"Te
rminate" :: DISPLAY AT(X+5,2
):STR$(NN+1)&" Delete"
330 IF VT=TT OR FLAG=1 THEN
350 :: DISPLAY AT(2,25)SIZE(
4):VT
340 FOR @=1 TO 10 :: DISPLAY
 AT(2,25)SIZE(1):CHR$(30)::
DISPLAY AT(2,25)SIZE(1):" "
:: CALL SOUND(-99,110,0,-4,0
):: NEXT @
350 DISPLAY AT(X+6,1):"    C
hoice?" :: ACCEPT AT(X+6,16)
SIZE(2)VALIDATE(DIGIT):K ::
IF K<>NN AND K<>NN+1 THEN 41
```

```
0
360 IF K=NN THEN CALL CLEAR
:: CLOSE #1 :: END
370 DISPLAY AT(X+5,11)SIZE(1
8):" #?" :: ACCEPT AT(X+5,15
)SIZE(2)VALIDATE(DIGIT):KD :
: IF KD<1 OR KD>NN THEN 370
380 DISPLAY AT(X+6,1)SIZE(27
)BEEP:" Verify - Delete ";PG
$(KD):"?" :: DISPLAY AT(X+6,
28)SIZE(1):"Y" :: ACCEPT AT(
X+6,28)SIZE(-1)VALIDATE("YN"
):Q$ :: IF Q$<>"Y" THEN 400
390 DELETE D$&PG$(KD)
400 CLOSE #1 :: CALL VCHAR(1
,3,32,672):: NN=0 :: X=0 ::
GOTO 180
410 IF K<1 OR K>99 OR LEN(PG
$(K))=0 THEN 320
420 CLOSE #1
430 CALL INIT :: CALL PEEK(-
31952,A,B):: CALL PEEK(A*256
+B-65534,A,B):: C=A*256+B-65
534 :: A$=D$&PG$(K):: CALL L
OAD(C,LEN(A$))
440 FOR I=1 TO LEN(A$):: CAL
L LOAD(C+I,ASC(SEG$(A$,I,1))
):: NEXT I :: CALL LOAD(C+I,
0)
450 CALL VCHAR(1,3,32,672)::
 CALL SCREEN(8):: FOR S=0 TO
 14 :: CALL COLOR(S,2,1):: N
EXT S :: DISPLAY AT(12,2):"L
OADING ";A$
460 RUN "DSKX.1234567890"
```

If you don't like my Tigercub cursor, just delete lines 110 (after the CALL CLEAR), 120 and 130. That routine for redefining the cursor has appeared recently in various newsletters without attribution, and I'd like to know who to credit for it.
The secret of it is in line 120, where the numbers after 12288 are the decimal equivelants of the hexadecimal numbers (which are the hex equivelant of the binary numbers represented by the off/on pixels) used to redefine a character.

(cursor redefiner credited to Terry Atkinson; see Tips #16. - Ed.)
(see improvements to the Menuloader in Tips #18, 20, 22, 23, 24, 27, 28, 29, 30 and 32. - Ed.)

You may have noticed that all programs published in the Tigercub's Tips are in 28-column format, just the way they will appear on the screen. And they are printed directly from LISTed actual programs, so that they cannot contain typographical errors - don't you wish the computer magazines did that!?

The problem is that when a program listing is merged into the TI-Writer buffer and printed in the formatter mode, the @, &, * and the exponent sign are treated as control characters, and strange things happen!

The following program will convert a program, which has been listed to disk with LIST "DSK1.FILENAME", into a file in 28-column format which can be loaded into TI-Writer, and will optionally substitute the left and right braces, ASCII 124 and the tilde for the @, &, * and the exponent sign, and transliterate them so that they will print correctly in the formatter mode. However, for that very reason this program will not print correctly! & has been altered in the list below.

When you come to line 280, type DATA shift 2, fctn F, shift 7, fctn G, shift 6, fctn W, shift 8, fctn A.

```
100 DISPLAY AT(2,4)ERASE ALL
:"28-COLUMN CONVERTER" :: DI
SPLAY AT(5,12):"by Jim Peter
son"
110 DISPLAY AT(7,1):" To con
vert a program, saved":"with
 LIST ""DSK1.FILENAME"",":"i
nto 28-column format which":
"can be merged into the text
"
120 DISPLAY AT(11,1):"buffer
 of TI-Writer."
```

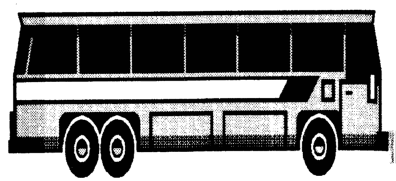
```
130 DISPLAY AT(13,1):" Optio
nally with transliter-":"ate
d @, &, * and ^ for cor-":"r
ect printing from formatter"
:"mode."
140 DISPLAY AT(18,1):" Do yo
u want to print the":"file f
rom the":" (E)ditor?":" (F)o
rmatter?"
150 ACCEPT AT(23,1)VALIDATE(
"EF")BEEP:Q$
160 DIM A$(1000):: CALL CLEA
R :: INPUT "What is the FILE
NAME?        DSK1.":FN$ :: FN
$="DSK1."&FN$ :: PRINT : :
170 INPUT "what is the new F
ILENAME?   DSK1.":PN$ :: PN$
="DSK1."&PN$ :: OPEN #1:FN$,
DISPLAY ,VARIABLE 80,INPUT :
: OPEN #2:PN$,DISPLAY ,VARIA
BLE 80,OUTPUT
180 IF Q$="E" THEN 190 :: PR
INT #2:".TL 126:94;" :: PRIN
T #2:".TL 123:64;" :: PRINT
#2:".TL 125:38;" :: PRINT #2
:".TL 124:42;"
190 FOR L=1 TO 1000 :: LINPU
T #1:A$(L):: IF LEN(A$(L-1))
=80 OR LEN(A$(L-1))=160 THEN
 A$(L-1)=A$(L-1)&A$(L):: L=L
-1
200 IF EOF(1)THEN L=L+1 :: G
OTO 220
210 NEXT L
220 FOR J=1 TO L-1 :: S=1
230 FOR T=1 TO 10 :: B$(T)=S
EG$(A$(J),S,28):: IF Q$="E"
THEN 240 :: GOSUB 280
240 S=S+28 :: NEXT T
250 FOR N=1 TO 10 :: IF B$(N
)<>"" THEN PRINT #2:B$(N)
260 NEXT N
270 NEXT J :: CLOSE #2 :: CL
OSE #1 :: END
280 DATA @,@,&,&,^,^,*,*
281 ! SEE NOTE ABOVE RE 280
290 RESTORE 280
300 FOR W=1 TO 4 :: READ CH$
,R$
310 X=POS(B$(T),CH$,1):: IF
X=0 THEN 330
320 B$(T)=SEG$(B$(T),1,X-1)&
R$&SEG$(B$(T),X+1,LEN(B$(T))
):: GOTO 310
330 NEXT W :: RETURN
```

(see fixes, improvements and revisions of this program in Tips 17, 18, 21, 38 and 50. - Ed.)

Happy hackin'

Jim P.

A Local Bus!!



Local Bus is a technique used on newer PC's, which is used to get around the slow, inefficient data and address busses that were forced upon the world by IBM in1982.

intel have pushed their silicon to melting point to get more speed from their slow architecture, but the maximum rate at which data could travel down a PC's bus was fixed years ago!

The way PC manufacturers have tried to get around this is to connect a seperate bus between the CPU, and local bus cards!

It doesn't appear to have worked though!

Gary safely beat his own intel 486 tranfer peripheral with local bus!

He ran a screen saver which was just bouncing one set of lines around the screen.

His GENEVE was running four sets of mirrored lines in a FORTRAN 99 MDOS program!

And it was keeping up with the PC very nicely!

Shame the GENEVE only had a 16MHz clock, 8-bit Data Bus, slow version of the PAL chip, and slow version of MDOS!

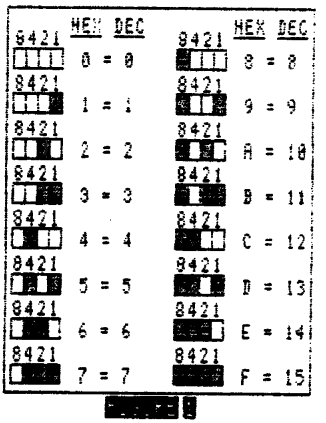
Just imagine a GENEVE based around a TMS9996 (which is a 9995 with a full 16-bit data bus!)

DESIGNING CHARACTERS MADE EASY
(OR AT LEAST UNDERSTOOD)

by: PAUL E. SCHEIDEMANTLE

As the title states I have set out to make designing of characters for both fonts and graphics easier to understand The cryptic way in which T.I. explains every aspect of their computer is best shown in how they explain the designing of characters. They show you a chart similar to FIGURE 1 below and expect you to memorize it or have it in front of you always. Thus making it tedious if not tiresome.

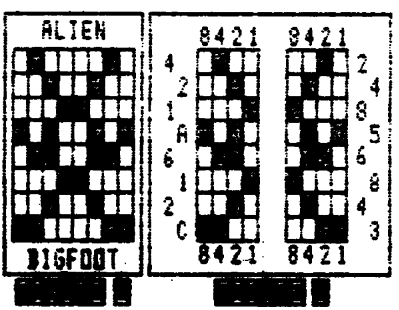
Well lets analyze the chart in FIGURE 1 below. First of all it has been enhansed to include both the numeric values of each dot (pixel) and the decimal equivalent. Now lets look closely and understand why each set not only has a differant CODE (HEX CODE).



HEXADECIMAL is a numbering system that uses base (16) (0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F). In our case it is short hand for those numbers that exceed single digits. But back to the subject. Notice that each colomn has a value above it (8,4,2,1). These and the fact that 10 = A will help you design and code your characters much quicker.

As you can see in the chart when all dots are off there is a value of 0 and that when you turn on the right most dot you have a value of 1 with the value of each dot doubling as it moves to the left. Notice that if you have a 3 that not only are the 2 dots on the right turned on, But more importantly you will now understand why the number is 3; because you simply add them up. After a while this method will become second nature to you and you will find yourself coding your characters in your head, without the aid of the chart. Instead of looking up a set like this "1010" you will automatically think Oh! thats 8 + 2 = A. or "1001" is 9 because 8 + 1 = 9.

FIG2 AND FIG 3 referred

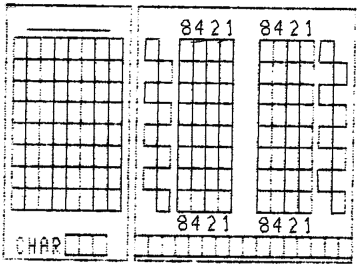


Now lets redesign the lower case "a" to a character we will call our ALIEN BIGFOOT. In FIGURE 2 you will see the shape of the character. While in FIGURE 3 it is broken into it's two halves, which are necessary to code it much more efficiently and to make it easier to see how it is done. Even though the split is in the middle it is still coded from left to right and top to bottom. Let's take each line separately.

	LEFT SIDE			RIGHT SIDE		
		DEC	HEX		DEC	HEX
	8421	= 15	= F	8421	= 15	= F
LINE #1	0100	= 4	= 4	0010	= 2	= 2
LINE #2	0010	= 2	= 2	0100	= 4	= 4
LINE #3	0001	= 1	= 1	1000	= 8	= 8
LINE #4	1010	= 10	= A	0101	= 5	= 5
LINE #5	0110	= 6	= 6	0110	= 6	= 6
LINE #6	0001	= 1	= 1	1000	= 8	= 8
LINE #7	0010	= 2	= 2	0100	= 4	= 4
LINE #8	1100	= 12	= C	0011	= 3	= 3

Now we take the HEX CODE and use it in a program with the CALL CHAR statement. CALL CHAR(97,"422418A5661824C3")

fig 4 here



A bit from Mr.Editor.

Well, it's now official as the front cover will show. The last issue which should have been numbered 42 was numbered issue 41. This was due to me having missed issue 41 the original and when it came to adding the issue number I found which was the last one I had and added one to it, hence the 41. Consider this a cunning test to see who is alert!! So far there has only been three people who have noticed. I must admit, I hadn't noticed until this was pointed out to me. Apologies for this minor mishap and to get things back into order, this issue will be 43. Many thanks to Richard for the Christmas front cover!!
Many thanks goes to all who have entered articles for this issue!

Happy Christmas!

NEWS UPDATE... from Stephen Shaw

Effective NOVEMBER 1ST 1993 Asgard Software is owned by HARRY BRASHEAR, a former software reviewer in MICROpendium.
NEW ADDRESS:
ASGARD SOFTWARE, 2753, Main Str, Newfane, NY, USA, 14108
Telephone 716/778/9104

HARDWARE formerly sold through Asgard (Mouse, Memory card, modules) are now available from Chris Bobbitt trading as ASGARD PERIPHERALS at:
1423 Flagship Dr, Woodbridge, VA, USA, 22192.
Mr Bobbitt is not able to accept phone calls.

Magazine ALGORITHM mentioned in these pages a while ago, has ceased publication. Magazine FRACTAL REPORT was last heard of with a doubled subscription fee of twenty pounds per annum.

24th November saw a working Geneve in the Shaw residence, courtesy of a flying visit from one of our US members. Lack of a colour monitor was a slight problem, but yours truly is using a mono monitor from Mike Goddard which filled the bill, even if it was only mono! The demonstration was a LITTLE late in the day, as our fellow fanatic arrived into Manchester Airport around 8.30pm and THEN had to travel here... finally departing here at erm.... (Happy to post more black beer on to you anytime Jeff!) Jerry was staying with a fellow QL owner, who was interested to see a photo of a TI Waftertape (Sinclair technology!) connected to a TI9/4A in 99er Magazine. He is also interested to know if anyone else has an ICL OPD, as he has two!

TECHNOLOGY...
Laser displays are well known in discos and concerts. In Manchester there is a large public open space called Piccadilly Gardens, with a large multi-story concrete tower from the '60's having a wall of concrete facing the gardens- not the more modern wall of glass! From the opposite side of the gardens adverts are being projected onto this large screen using coloured lasers- green, red and blue, which can mix to form other colours. Ads can be (and usually are) animated, and there is a large laser-drawn analog clock in the sequence.

There is now on commercial sale a battery charger which can SAFELY recharge not only NiCads but also Zinc Chloride AND those nickel cadmium cells which tell you not to recharge them! It will not recharge Lithium batteries. And it is NOT a normal charger but has special safety circuitry - it isn't cheap costing five times the cost of more hazardous chargers! However zinc chloride cells are also cheaper than NiCads, and kinder to the environment. Batteries take about 8 hours to recharge, and none Ni-Cads can be recharged about ten times before performance falls. Due to chemical changes which occur in a "flat" none-NiCad battery, they need to be recharged very soon after depletion! The company is INNOVATIONS INTERNATIONAL, tel 0793 436648

The BBC - having purchased Anime classic AKIRA some time ago- are reputed to be showing AKIRA on tv over Christmas. If you want to know how much footage has been removed, the video version runs for 124 minutes. See how long the BBC version runs for... Akira is one of those nasty Japanese cartoons,

made for adults (hence I suspect a post-9pm showing) which some newspapers have been ranting about.

Sales of digital tape (both versions) have been reported as non existant, and the medium may be said to have failed. What will happen to the new video CD due for launch around Christmas? Not the 12" laser videodisks, but normal audio size CDs capable of an hours video- using Fractal techniques to encode the data! CDs will play on both NTSC and PAL players -no conversion needed- which will make importation easier.

We have used a number of graphics programs here, included several suggested by Clifford Pickover of IBM in America. Manchester recently was home to a video exhibit from UK IBM Artist in Residence William Latham- a half hour tape of various computer produced graphics. The display was in the city's Art Gallery. Computer Art comes of age.

2022 update: Now transferred to dvd/ISO. William is still creating similar images.

Anyone experience TI-Writer dropping characters when it word-wraps? Well, that is not exceptional to TI-Writer, as using a 25MHz PC at work and a rather expensive IBM word processor program, I still get the odd character dropped off!

Use of Computers.

No, it's not a moan at game players. I must admit to being partial to a few games of Sokoban now and again. It's more an article of surprise for me. I am currently on my final year of an electronics course at Nottingham Trent University or Nottingham Polytechnic to those who are a few months out of date. The work load is very heavy as it is on most degrees I suppose, apart from Art and Textiles! One night I went back to my digs with three assignments to do from that day so me and the two people I share a house with set on to get these done.

One of the assignments was to analyse an analogue circuit we had been given and find out all we could about its operation. After some algebra I came out with some equations for the circuit and turned to Extended Basic and The Missing Link for some help. I always do this when it may save some time.

What I did was to write a basic program that would force a frequency into the equations and monitor what the output would be. The input frequency had to vary over the 0 to 1 MHz range and I used the Missing Link to display a graph of the input against a few different output points. Also as the computer was working out the answers to the equations at each frequency I had it saving them to a D/V 80 file so it could be included in TI-Writer.

After about quarter of an hour writing the 'for - next' loops and scaling the graph to the correct size the job was complete and I had all the graphs neatly printed out. In about another hour I had been into the word processor and included the results' table from disk and completed the write up for this assignment. What I couldn't believe is that the other two people in the house had actually sat down with paper/pen and calculator and were typing the results in one by one. It took them over three hours just to create the results' table and graphs. This to me seems a great waste of their two high powered PC's and time.

Considering one of the first computers, Colossus, was designed to crack codes in the world war and save us much time, I find it very unusual that people are taking a step back. The 4A or Geneve are one of the most user friendly systems I've come across and are very simple to adapt to almost any application! Are you guilty of computer misuse? Just think of how much time a small program could save you in the long run!

WHERE ARE YOU OUT THERE??? Do you have any tips for your magazine? Up till now we have had no submission for this artical whatsoever. So if you want to see your name in lights then send in your tip to the Editor. (See front of mag for address).

Here is one that may be of use to the many Mini Memory users. Unlike TI, the Line by line assembler has a bug in it. The program only knows how to NEW its self. If you load into the Mini Memory something you have saved for more work you do not get your original listing from the correct locations. Boy does it take some sorting out. Thanks to a fix you can now do OLD as it was intended with NO sort out problems. What you have to do is load up the Mini Memory, select L for load option. Load in the Line by Line & Lines Program, via easy bug. When loaded Function Quit. Reselect the Easy Bug screen, Select Option (M)odify. You will then be presented with a Hex number on the left of your screen and a location content address, in a hex two digit presentation. IE. M7D00 = FF - > After the > the cursor will flash. What we do next is press enter. This will bring us to the M position type in the address locations below and the values after the >. You will notice that the loction are not sequential. So you will have to come out of the editor. Do this by pressing (.)
Then Type M and the address followed by the new memory value.
IE:

```
M7D00 = 83 - > 92
.
M7FFF = 00 - > 93

.
?
```

NOW HERE ARE THE NEW LOCATIONS:-

Change	Old	to	New
71A9	>3E		>8E
71AF	>3E		>8E
7229	>3E		>8E

What you do now is save your new Line By Line program on ANOTHER tape. When you save, save loctions 7000 to 7FFF.

MULTIPLAN

Once you start scrolling with the Fctn Arrow keys let go of the function key, but keep the arrow depressed. It still scrolls.

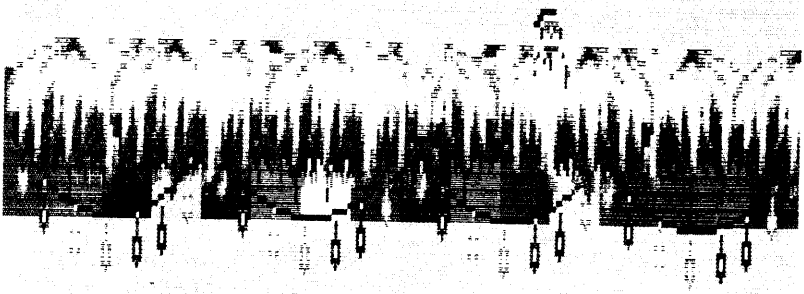
TI WRITER

To avoid a full buffer in TI Writer you should save your files and then use the SD command to see the file size. The magic figure is 92 Sectors.

EXTENDED BASIC

Console search for a "Load" file. What is that you may ask. Well if you include in you extended basic program CALL INIT :: CALL LOAD(-31962,255) your console will search for the LOAD file on DSK1.
So if you wish you can chain two programs together or return to a master selection screen.

MEMORY FULL!!!!!!!!!!!!



Merry

TEXAS INSTRUMENTS Xmas

HOME COMPUTER

Merry Xmas from Richard Twyning.