

Volume 2, Issue 8

January 1st., 1986

TI-LINES is the monthly newsletter of OXON TI USERS, and the form of presentation of any material remains the copyright of each individual author. OXON TI USERS reserves the right to submit any material to other newsletters for publication, with due credit to the author(s) concerned. Submissions are accepted on this understanding.

It is the responsibility of each author to ensure that no copyright infringement will occur by the publication of their articles in TI-LINES, and OXON TI USERS cannot be held responsible for any such infringements. Every effort is made to ensure that any information given in TI-LINES is correct.

TI-LINES is produced and published by:

```
*****
*
*      Peter G. Q. Brooks
*
*      96 Banbury Road
*      OXFORD
*      OX2 6JT
*
*      Tel: Oxford 50822
*
*****
```



Oxon residents are offered a subsidised subscription at £1.56 p.a., payable either as 12 second class postage stamps or by cheque.

THIS OFFER APPLIES UNTIL MAY 1986 ONLY

TI-LINES is available on Associate subscription to Users resident outside Oxfordshire, for £10 p.a. Back issues are £2 including post and packing. New subscriptions begin with Issue 1 of the current volume, up to and including the current issue, regardless of the number of issues elapsed.

Contributions should be submitted either on diskette in TI-Writer compatible files (DIS/VAR 80 or DIS/FIX 80), or in a form which is as legible as possible. Artwork should fit within an A4 area and should not contain colour. Very high contrast line drawings are preferred and these may be produced by arrangement with the publisher.

I n d e x t o a r t i c l e s

	<u>P a g e</u>
EDITORIAL.....	3
Rambling Prose...	
JAMES STRINGFELLOW.....	15
Programs For You To Key In	
PLOTTING A SINE WAVE.....	27
If Guy Fawkes Had Stuck To This Kind Of Plot We'd Have No Fun In November...	
BULLETIN BOARD.....	34
Goodies To Go	
CHANGING XB EDITING COLOURS.....	35
When An Interruption Is Exactly What You Need	
LETTERS.....	38
Stephen Shaw On Contributions For Freeware	
MAXIMEM.....	39
A Competitor For GRAM-Kracker ?	
CLOSE FILE.....	40
A Last Few Words	

-----  
E D I T O R I A L  
-----

BETTER LATE THAN NEVER ??  
-----

Tardiness seems to be my hallmark these days. If it's not TI-LINES it's something else.

In this case, although TI-LINES is late as usual (and a Happy New Year to you all, by the way!) this is not the reason for the heading of this piece.

You may, or you may not, be pleased to hear that AT LAST! the new version of the TIHOME SOFTWARE COLLECTION Catalogue has finally been produced.

Yes, by dint of sleeping only eighteen hours in nine days I have managed to bring together not only the 159 programs which made the grade (sort of) but also the golden words provided as a review of every program.

I cannot begin to express my gratitude to the reviewers, who spent a massive amount of time on this mammoth task. OTIUsers STEPHEN SHAW, GRAHAM HILTON, PHILLIP MARSDEN, RICHARD SIERAKOWSKI, RICHARD BLANDEN, JEREMY BYGOTT, STAN DIXON, JOHN RICE, NEIL LAWSON, and ALLEN BURT laboured through over 850 programs in a matter of a few months, leaving me with the task of compiling their comments and ratings together with the detailed equipment information, which in itself was no minor chore.

The resulting catalogue is a whopping 56 pages long and should be ready for issue by the time that you read this. All the programs have been recoded and partially checked by yours truly (hence the lack of sleep) and are ready to be dragged off their disks at a moment's notice (well, almost).

At the time of writing (January 16th.) I have produced the master A4 copy, and it only requires me to find time to get the reduction masters (A5) done.

There are already over 100 trusting former TIHOME members who paid their 50p for the catalogue some eighteen months to two years ago (!) and who will be eventually hearing it thud through their letterboxes.

However, in order for them to access the Collection, they are going to have to take out a subscription to OXON TI USERS, as the Collection is associated with us and cannot be provided for non-members.

I am now ready also to accept submissions for consideration for inclusion in the New Collection, on roughly the same basis as when PAUL DICKS ran it.

He offered four chosen programs in exchange for each one original accepted, and I will continue that policy.

However, naughty folks took advantage of Paul and according to both the reviewers and my own double-checking they managed to pass off commercial programs, copies of items which appeared in books and magazines, and also translations from other dialects of BASIC, all of which constitute offences under the Copyright Act. The offending items have all been removed from the New Collection (unless any have managed to slip through which I doubt), and the small change that I have made to Paul's original policy is simple: if any items which infringe the Copyright Act are submitted, I will hand all relevant material over to the proper authorities for them to consider the action appropriate.

This may seem a little harsh, if not over the top, but in the States there are a mounting number of cases of individuals and User groups being taken to court and prosecuted for infringements of copyright, simply through having naughty material in their Software Collections. I have no wish to add to their number!

Apart from that, when you think about it, it is a rather nasty individual who will offer you someone else's work and attempt to pass it off as his or her own. With any luck, we might even get some interesting and original programs with such a policy - although from past experience, we may not get anywhere near as many submissions as Paul received!

Programs from the Collection are priced at £1 each, and unless specified otherwise, will be recorded on cassette, one copy each side. Programs may on request be recorded on disk (either single or double sided) although the price per program remains the same. A charge of 65p is made on the final amount to cover post and packing (which it probably won't, but it can be such a hassle trying to charge the exact postage on every package!).

Cheques/Postal Orders can be made payable to either me or TIHOME SOFTWARE COLLECTION, and a full name and address (legible) must be included, for obvious reasons.

The catalogue will be available free to holders of current OTIU membership, while non-members will have to pay £1, deductible from their first request once they have obtained membership. I will be writing to some 2000 former TIHOMErs in due course (as finances permit!).

~~~~~  
BURNED FINGERS  
-----

MicroPress, the subsidiary of Castle House Publishers, who handled the publication of my (one and only so far) book, MASTERING THE TI-99, managed to burn their fingers in the micro market, and have now ceased to publish anything at all on computers. They intend sticking to safer subjects.

I gather that my own book managed to recover its costs and make a profit (how much was not revealed, but from my own figures I would guess about £18000 or so) and has distinguished itself (if that is the expression) by incurring far fewer letters of the "I don't understand what you've done on page so-and-so..." type (Ol' Baldie's right hand creeps round behind him and thumps him on the back...).

I "gather" all of this because I have been negotiating with my publisher the purchase of the outstanding, unsold, copies of my book, and on Monday 13th., hopefully NOT unlucky for me, I received no less than 565 copies.

The price I arranged was low enough that I am able to offer copies to anyone who isn't sick to death of MTT, at the price of £3 per copy inclusive of post and packing. This offer applies largely to existing OTIUsers (go on, buy a dozen copies, you can chuck 'em in stews, use 'em on the fire, and even make handy little snacks out of 'em!) but if anyone knows of a worthy non-OTIUser who wants a copy or three for his mates, well, you know what my home number is!

The copyright in the book should also revert to me shortly, which opens up another interesting avenue for me to explore.

And now that I have the TSC Catalogue finished, I should be able to turn my hand to the dreaded Quinsoft publishing venture and actually get some of the first booklets out in '86!

~~~~~  
ERROR!  
-----

I noticed an error in V2.6 the other day (shows how fast I am these days!). On page 14 there was a full stop "." instead of a greater than ">" in the list of DATA values at the bottom of the page. Incidentally, did anyone find the green cover really devastatingly fantastic? I don't think anyone even noticed...

~~~~~  
WIMPS!  
-----

Wimp is the latest jargon word which you may find cropping up on your cereal packets of a morning (I don't think). It describes the range of latest goodies which are currently top of the list of any decently computer-literate company executive:

WINDOWS  
ICONS  
MICE  
PULL-DOWN MENUS

Hands up all those who don't know what all of these are...if you dare!

~~~~~  
MICROMART IS BACK...ISH  
-----

You may remember a while back that as soon as I recommended the weekly sales mag called MicroMart it promptly disappeared from the shelves. Well, it came back again recently, under new management, on a fortnightly basis.

My reason for mentioning it is that it carried a piece on OTIU, apparently on its ACC page (ACC - Association of Computer Clubs has a page in PCW and has put out info on us in the past, but with an old address). The only reason that I know about it is that I have had an enquiry from a reader from the Isle of Man, who mentioned it in his letter. As it was correctly addressed to me here on Banbury Road, it means that the letter I sent some months ago correcting the original information has indeed been noticed.

However, last September I verified that OTIU was not on the ACC Database, as we don't qualify under the description of a computer club, which makes me wonder how we managed to get into MicroMart.

Still, if it means greater exposure and more members, who cares?

## TK-WRITER TWEAKED

-----

If you have the TK-Writer Extended BASIC loader for the Public Domain TI-Writer enhancement files, and bemoaned that effect of using SD, the Show Directory command - it returns you to the title screen and loses all your text! - then this information from the MSP99 newsletter may help you.

The original source is Super 99 Monthly's June issue, in which a fix suggested by JIM McGARVIE on the San Diego TBBS was presented.

Using a sector editor (like Disk Fixer or maybe Forth), locate the third sector of the EDITAL file. Find byte >14 which should be >53, and change it to >20. Write the changed sector back to disk.

Once that is done, every time you accidentally use SD with the TK-Writer it will merely "honk" at you and clear the command line. Not as satisfactory as actually showing you the directory, but better than a kick in the teeth.

Another suggestion, from WOODWARD WILSON, was to call the TK-Writer from Extended BASIC with:

```
10 RUN "DSK1.TK-WRITER"  
20 END
```

and then add line 265 to the Extended BASIC bootstrap program:

```
265 RUN "DSK1.CAT"
```

or whatever you call your own disk cataloguing file. I did try this, and when I did it with an UN-MODIFIED TK-writer, I just got the title screen as usual and lost all my text (which was only called up for testing purposes).

There may be something odd about my console, so don't take my experience as definitive. Alternatively, why not get a copy of FUNLWRITER instead ?

~~~~~  
HELLO, HELLO  
-----

A warm welcome to new DTIUser FRANCESCO DE LAMA, and I forgot to ask you if your cod and chips was OK!

~~~~~  
VE HAFF VAYS OFF MAKINK YOU TALK GPL!  
-----

Just before Christmas I ordered a new book from ANDREW HOPKINSON and then toddled off for a fortnight's mayhem..er..holiday. When I returned, a copy of TI-99/4A INTERN was on the mat, written by HEINER MARTIN and translated (not very well) by PETER COATES from the German original.

It is a brief presentation (and minimal discussion) of ROM, GROM, and GPL, and I lack the necessary time and knowledge at present to do justice to this book. Certainly you will need an understanding (at the very least) of the vocabulary and processes of 9900 Assembly Language in order to begin to make sense of what you see, and the minimal comments and lack of detailed explanation of almost all of the sections make it a daunting task for the inexperienced (i.e. me).

There is a listing (commented briefly) of the ROM and of three GROMs, but you definitely need your wits about you. This is by no means a pick-it-up-and-thumb-through book. If I remember rightly, the price is £12 and when you consider the potential value in terms of raw information, well worth it.

Contact Andrew on 0703 732801 between 7 and 11 in the evening.

~~~~~  
JAMES STRINGFELLOW  
-----

More than six months ago, James, currently based in France, sent me a disk with a number of programs making use of the poke-it-in-byte-by-byte approach to machine code on the TI. Each time I tried to consider publishing one of the routines, something (usually lack of time for preparation!) reared its ugly head and prevented me.

I have made a special effort and present James' offerings in this issue for your perusal and delectation. If it triggers a flood of similar routines I will be happy to publish them.

James has also noted a handy use for the CTRL U operation with his printer direct from the TI-Writer Editor (and not through the more lengthy approach of the Formatter). He can send high resolution bit-graphics commands to his printer while using the Editor, and I hope to be able to publish a sample and article from him in the near future.

He is also anticipating receiving the dreaded GRAM-Kracker any day now, and I have asked him if he would consider giving TI-LINES the benefit of his experience and opinion once he has had time to evaluate it properly.

1986 just might be shaping up well. If only I could get the newsletter out on time...

~~~~~  
WHEN THE CHIPS ARE DOWN...  
-----

I have been advised that CRAIG MILLER of Smart Programmer fame (and other things I know, but Smart Programmer is the subject at the moment) has recently published details of a fault which can occur with the interface card in the Peripheral Expansion Box. Apparently there is a certain type of chip on the card which is weak and prone to failure, leading to peculiar performance when transferring data to and from the box.

This may lie behind some disturbing disk drive problems which have cropped up recently, where perfectly healthy drives have been refusing to work correctly when incorporated in the box (and with a non-TI controller).

If I can get further details I will pass them on.

~~~~~  
CENTRONICS FIX  
-----

I understand that the replacement ROM, made available in the States to sort out the lack of true Centronics compatibility on the RS232 card, is now available in the UK from TI. The price is probably about £10, but I am not

sure of availability. As soon as I have firmer details, I will of course pass them on.

~~~~~  
OOPS!  
-----

Sorry Neville! Avid readers may have noticed an odd sentence in the letter from NEVILLE BOSWORTH in V2.7, which began: "ALAN DAVEYthis reason...".

Ahem.

This was not due to pigeons perching on the aerial of my TV, but was due to failing eyesight on the part of the Editor, who tried to overtyping the Alan Davey on the following line and missed.

The sentence should have started: "It is for this reason...". Oh dear.

~~~~~  
CHEAPO MANUAL  
-----

I can't recall if I published the fact, but apparently TI have been selling their TI-WRITER manuals without folders for £3.50. Someone approached PHIL WINGROVE I understand, who knew nothing about it at all, and I had started to wonder about my sanity when hooray! a letter arrived from an OTIUser who had bought a copy. I still don't have an exact address to write to, but it is I understand the TECHNICAL LITERATURE people who are chucking the manual out. They are only doing it to available stock, which to me smacks of a prelude to placing the whole of TI-Writer in the Public Domain.

Watch this space...

~~~~~  
SUNRISE ?  
-----

There is a glimmer of something just beyond the horizon, which sources close to me suggest may indicate a possible meeting for OTIUsers in the not too distant future. When I have more information I will pass the word around.

~~~~~  
TOPICS TO COME  
-----

In the coming months I hope to begin covering more and more (and possibly even more with the assistance of fellow OTIUsers!), including Forth, more on Speech, Assembly Language, odd bits like Tile Graphics and Core War, Wa-Tor, and Turing Machines, and newcomer FRANCESCO DE LAMA has agreed to cast his eye over a possible article or two on some of the more complex bits of Maths (especially the stuff which is outside my current experience). It is also possible that VINCE COHEN may succumb to persuasion to write up his program to convert PRK program format files to TI-Writer DIS/VAR 80.

And talking of DIS/VAR WARS (who dat ?), the Galactic Programmer's Lexicon has surfaced in a box of junk aboard the Tongue Of Fire. Will Nilbog The Ugly and Frawd The Deficient be persuaded by Yriaf The Slightly Suspect to



go in search of the Missing Pages ? Has anyone been caught by the PRIVATES yet and did it bother them ?

Find the answers to some of this, and more, in this issue's fractured file!

~~~~~  
WHEN ROWS ARE WRONG  
-----

MSP99 republished some useful summaries of TI-Writer commands in its Oct/Nov newsletter, originally taken from 99er News of TI UG Will County, Romeoville in Illinois.

However, apart from a simple but catastrophic error in the discussion of the PF command, there was also an error in the explanation of one of the many features of FS. It is an error that I have seen a number of times, so it is perhaps worth pointing it out.

FS or FindString lets you do several things. You can search for a letter, word, or phrase beginning after the current cursor position to the end of the document which you have in memory.

Another feature allows you to search within a defined vertical block of text - that is, you specify a start column and an end, and the search will only take place within those columns (this also applies to ReplaceString).

However, for some reason a number of authors have stated that the columnar search is in fact referring to ROWS rather than columns, so that a command like:

1 15 /CAR/

should be looking for an occurrence of CAR in lines 2 to 15 (i.e., from one line AFTER the specified start). This is definitely not the case. The two numbers refer to COLUMNS and not rows or lines.

You may remember that HOWARD GREENBERG provided additional information in an earlier TI-LINES that FS also possessed the same feature that it does in the Editor/Assembler, namely that it can be made to find the "nth" occurrence of a given string.

Thus:

2 /CAR/

will search for the SECONDD occurrence of CAR after the current position of the cursor up to the end of the document.

RS or ReplaceString does not appear to operate in exactly the same manner, but it too can be used to replace a given string within defined column positions.

0 10 /CAR/Car/

will only operate between columns 0 and 10, and ignore any occurrence of CAR which lie beyond column 10.

One further point which might have escaped the reader's notice, and that is that the column numbers used by FS and RS are not the same as those used by T or TAB function. Column 39 in the TAB line is column 40 as far as the two

string search/replace commands are concerned, so you should calculate your use of the numbers with some care.

~~~~~  
WHEN THE MANAGER IS SLOW  
-----

For what it is worth, this is a small tip on speeding up the initialisation of a large number of disks at one session, using the Disk Manager (it does not apply to DM1000, as that has a much faster "box" approach to the process of initialisation).

I am often annoyed by the daft Manager's response to a request to initialise a disk which has never been initialised before. Despite the fact that I have selected the necessary option to initialise a disk, which suggests that I wish to do just that, the dim Manager checks and rechecks the disk until it has finally convinced itself of something that I knew all along: namely that the disk is uninitialised!

The several seconds that must elapse while the Manager undertakes this totally unnecessary checking can prove to be most annoying, so I have evolved a method of bypassing this irritating hiatus. All I do, when I have finished initialising a disk, is to leave the disk in the drive and press PROCEED to begin a fresh initialisation. Once I press ENTER to indicate that I still want to use drive 1 ('cos I don't usually make use of the Single Drive Processing option with a two-drive system), the Manager checks the disk in the drive and straight away comes up with the disk name and then presents the first of the initialisation options. At this point I whip out the previously initialised disk and substitute a fresh, uninitialised disk, and the Manager is none the wiser.

It saves perhaps ten seconds per disk in processing time, which seems like nothing when you spell it out, but sitting in front of the computer and waiting for that period while the damned thing discovers the obvious, well, it can be a bit annoying.

It's amazing what you can find yourself doing to shave a second or two off the time you sit around doing nothing...

~~~~~  
LATE NEWS  
-----

I have just picked up my copy of PERSONAL COMPUTER WORLD, the monthly micro mag, and found a short piece on DTIU on the ACC page. This is presumably the same article that has been seen in MicroMart, as I believe that both mags now have the same publisher (and therefore will save money by making only one payment to the author(s) for publication in both mags). This now explains how a recent enquirer had my correct address, gleaned from the ACC page in MicroMart!

~~~~~  
A CHINK IN THE ARMOUR  
-----

Recently my copy of the excellent TI\*MES (well, six of them, actually) thudded through the letterbox while I was preparing to go to work, and so I

paused for a quick leaf through to see if there was anything juicy. Quite by chance my eyes fell upon part of STEPHEN SHAW's RAMBLES, and you could have knocked me over with a feather (well, a heavy feather anyway).

I have often discussed the possibility of writing a segment of code to be loaded from cassette into the basic console, with the intention of providing indirect access to machine code without any additional hardware, albeit not in the pages of TI-LINES but in the behind-the-scenes chit-chat which goes on in some considerable quantity, but a mixture of lack of time and lack of basic knowledge of both the system and the Cassette Operating System have prevented me from pursuing the idea at all.

I have also had a short series of articles waiting to be developed for inclusion in TI-LINES, based on information provided by RICHARD BLANDEN, which would indicate ways of accessing the VDP registers, but again lack of time and personal knowledge have slowed me up.

Stephen published a small piece of information which concerned both of these subjects: (machine level access with just the basic console, and access to the VDP registers) in his article which had me sitting bolt upright, and as a result of the time I spent reading and re-reading the section I was twenty minutes late for work!

If you have a copy of TI\*MES issue 11, the section begins on page 52 and is a presentation on how you can provide access to 32 sprites in TI BASIC but WITHOUT any additional modules or other hardware.

Stephen refers back to an earlier Ramble in issue 9 which discussed the VDP registers, well worth reading if you can't wait for me to get round to writing it all up!

The source of the information is MARCELLO ZANNINI of the BOLOGNA TI USERS group in Italy, who submitted it to the SYDNEY NEWS DIGEST, published by T.I.S.H.U.G. in Australia where it appeared back in March 1985.

The sprites do not have automatic motion, nor is there a facility to CALL COINC, but their creation and positioning is a masterpiece of lateral thinking (and in the process I noticed a quite clever trick with CHR\$( ) as well).

What fascinates me most about the whole thing is the realisation that here is an opening for the intrepid Experimenting TI Owner, a rare beast which has hardly been seen for the last three years. Marcello, or a colleague, has found a way of writing a data file (i.e., created through the use of OPEN) which the computer can actually OLD. You may be aware that the only thing that OLD appears capable of loading is PROGRAM format files, and that OPEN cannot create such files. Intriguing, isn't it ?

The sprites can be placed with single pixel precision, the ASCII code and the colour can be specified, and all with the use, would you believe, of the resident CALL CHAR( ) subprogram.

The example routine which Marcello produced is designed to work with ONLY the basic system, so you need only the console, a set of cassette leads, a tape recorder, and of course a tape. The rest of your equipment (if you have any) should be turned off.

Under normal TI BASIC use, access to sprites is denied (probably out of spite!). However, under those circumstances when sprites are available (for example, when the MiniMemory module is present and the additional CALLs are available which permit sprite creation), the computer stores all details on

each sprite in a thing called a SPRITE ATTRIBUTE LIST or TABLE. This is set down in RAM, and the start location is stored in another area, known as VDP REGISTER 5.

The clever piece of trickery developed by the Bologna group involves telling the computer where this table begins, by altering the contents of VDP register 5. The technique behind this alteration is very, very interesting.

It involves the creation of a special tape file, in SEQUENTIAL, DISPLAY, FIXED 64 format. This file is opened as an OUTPUT file to CS1, and a small string of characters is sent to it.

Only eight characters are sent: the first six have an ASCII code of zero, and the last two define both the VDP register to be altered, and the start address of the sprite attribute table.

Note that ANY of the VDP registers can thus be altered using this technique.

The calculation is performed like this:

Accept the VDP register number, R, in the range 0 to 7

Accept the location pointer, D, as a code between 0 and 255

Evaluate the expression  $18429 - 256 * R - D$  and assign it to A

Send six CHR\$(0) and CHR\$(INT(A/256)) and CHR\$(A) to the file

The neat trick, by the way, is the use of CHR\$( ) to produce the effect of "clock" arithmetic: CHR\$(A) brings any value up to 32767 to a value in the range 0 to 255. If you are not sure about this, try printing ASC(CHR\$(257)) in the Immediate mode. CHR\$( ) only works with values between 0 and 255, so any value over that technically has 256 subtracted from it until the remainder lies in the range 0-255 (how this is done is very simple, but not really worth discussing here and now!).

The effect of using the first part (CHR\$(INT(A/256))) is to produce a character whose ASCII code represents the "high" byte of a two byte number, while the CHR\$(A) produces a character whose ASCII code represents the "low" byte of the same number.

All of this is by way of detailed explanation of the processes initially involved.

Once the file has been closed, you QUIT or BYE and then reselect TI BASIC.

This is the fascinating bit. When you OLD a program from tape, the first part of the incoming data contains instructions to the computer about the type of data, and where it is to be stored in RAM. The small file that has been created is treated by OLD as if it is the first part of some incoming data (called the "header").

The computer will respond with the usual DATA OK message and instructions to press STOP and ENTER.

Once you ENTER, there is a lengthy pause before anything happens - just as if you had OLDed a whacking great program. Then the screen will go bananas.

Don't panic, Captain Mainwaring!...as they say. Wait until the screen has stopped flashing - it should be black with possibly coloured vertical bars.

Now press one of the QWERTY keys - Stephen's article says to use one of the alphabet keys - and press ENTER. The computer responds with an error honk, and the message MEMORY FULL.

Curiouser and curiouser, as Alice would no doubt have said.

Now type NEW and ENTER, whereupon faithful old TI BASIC READY appears.

You are now ready to start specifying your sprites.

Let us take a step backwards at this point. Earlier I mentioned that the value for D, the location pointer, should lie between 0 and 255. A value of 15, it turns out, will make the sprite attribute table lie in the same memory locations as is normally occupied by the character definitions for ASCII 144 to 159.

If a value for D of 14 is used, the sprite table lies in the same locations as the definitions for ASCII 128 to 143. A little maths should enable you to calculate values of D for other areas.

This puts control of the sprite attribute table within our grasp in TI BASIC as we already possess a command to alter the contents of those very memory locations: CALL CHAR().

If you use a value of 15 for D, making the table begin at the same location as the definition for CHR\$(144), then CALL CHAR(144,.....) will place values in that part of the table.

What values do we use ?

Stephen's article provides the information:

The CALL CHAR() subprogram can alter 8 bytes at a time (16 hex digits). All you need to specify a sprite's screen location, ASCII code, and colour, are 4 bytes. A single CALL CHAR() can thus specify the details for two sprites.

The format is:

```
CALL CHAR(144, "Y1X1A1C1Y2X2A2C2")
```

where Y1 is the screen row co-ordinate in the range 0 to 191 for sprite one

Y2 is the same for the second sprite

X1 is the screen column co-ordinate in the range 0 to 255

X2 is the same for the second sprite

A1 is the ASCII code of the first sprite, plus an offset of 96

A2 is the same for the second sprite

C1 is the foreground colour in the range 0 to 15

C2 is the same for the second sprite

It looks a bit confusing, the more so when you realise that the decimal values have to be converted to hex, but this should not prove too much of a problem for the hard-bitten TI enthusiast with only a console!

Once the initial program (given below) has been run, and the necessary procedures executed, you can experiment in the Immediate mode to see what comes up.

```

100 CALL CLEAR
110 INPUT "REGISTER (0-7) ? ":R
120 INPUT "VALUE (0-255) ? ":D
130 A=18429-256*R-D
140 X*=CHR$(0)&CHR$(0)&CHR$(0)
150 C* =X*&X*&CHR$(INT(A/256))&CHR$(A)
160 PRINT "OPENING FILE"
170 OPEN #1:"CS1",OUTPUT, FIXED
180 PRINT "ABOUT TO STORE FILE"
190 PRINT #1:C$
200 CLOSE #1
210 PRINT "FILE CLOSED"

```

Once you have entered this routine I recommend that you store it on tape before running it.

When you run it, enter a value of 5 for the Register, and 15 for the Value.

I tend to avoid having the remote lead connected to my tape recorder, and the messages on screen tell me exactly when the daft Operating System is really ready to send data to the tape.

When the file has been stored, rewind the tape to just before the header tone, and QUIT. Reselect TI BASIC, and enter OLD CS1 (if you have removed your remote lead, ignore all the cobblers about rewinding the tape and just press ENTER until it says READING, when you should press PLAY!).

When DATA OK appears, press ENTER and wait, as detailed above. When you see the steady black screen, type a letter (say A) and press ENTER. If you get anything other than MEMORY FULL you may have done something wrong and the following may not work.

Type NEW, and press ENTER, and then type this:

```
CALL CHAR(144,"1A1AA10F")
```

This defines two sprites, one with shape and colour, the other without!

The shapely sprite is at location >1A1A, which in English is row 26, column 26 (>1A is 26, bearing in mind that the numbering starts at zero!) and it has an ASCII code of >A1, which is 161 and which includes an "offset" of 96, making it a true code of 65 - or the letter "A" (65 + 96 = 161), and it will have a colour of >0F, or 15, which is equivalent to the value 16 in our CALL SCREEN() or CALL COLOR() - namely, white.

It is possible that the colour definition might extend to both foreground and background - I have yet to investigate.

Either way, towards the top left of your screen should appear a ghostly letter "A" as a sprite. I leave you to begin the experiments!

One final point: Stephen's article says that you must follow the definition of the highest numbered sprite with a CALL CHAR() using "DO" as its string, as the sprite table must end with a value of decimal 208 or >D0.

```

~~~~~
~~~~~

```

Programs For You To Type In

```

*****
*
* 10 !*****
* 11 !*
* 12 !*           CHARSET           *
* 13 !*
* 14 !*   EXTENDED BASIC           *
* 15 !* WITH MEMORY EXPANSION*
* 16 !*
* 17 !* Run program Type New *
* 18 !* Start your program *
* 19 !* CALL LINK("CHAR") *
* 20 !* Then you will have *
* 21 !* a new character set *
* 22 !* without using any *
* 23 !*           memory *
* 24 !*
* 25 !*****
* 26 !
* 27 !
* 28 !
* 29 !CHARLOAD From 30 to 143
* 30 CALL INIT
* 31 CALL LOAD(16368,67,72,65,82,32,32,40,132)
* 32 CALL LOAD(8194,40,150,63,240)
* 33 DATA 255,129,129,129,129,129,129,129,255
* 34 DATA 0,0,0,0,0,0,0,0
* 35 DATA 0,0,0,0,0,0,0,0
* 36 DATA 0,16,16,16,16,0,16,0
* 37 DATA 0,40,40,0,0,0,0,0
* 38 DATA 0,40,124,40,40,124,40,0
* 39 DATA 0,56,84,48,24,84,56,0
* 40 DATA 0,68,76,24,48,100,68,0
* 41 DATA 0,32,80,32,84,72,52,0
* 42 DATA 0,8,16,32,0,0,0,0
* 43 DATA 0,8,16,16,16,16,8,0
* 44 DATA 0,32,16,16,16,16,32,0
* 45 DATA 0,68,40,124,40,68,0,0
* 46 DATA 0,16,16,124,16,16,0,0
* 47 DATA 0,0,0,0,48,16,32,0
* 48 DATA 0,0,0,124,0,0,0,0
* 49 DATA 0,0,0,0,0,48,48,0
* 50 DATA 0,4,8,16,32,64,128,0
* 51 DATA 0,56,68,68,68,68,56,0
* 52 DATA 0,16,48,16,16,16,56,0
* 53 DATA 0,56,68,8,16,32,124,0
* 54 DATA 0,56,68,24,4,68,56,0
* 55 DATA 0,8,24,40,72,124,8,0
* 56 DATA 0,120,64,120,4,68,56,0
* 57 DATA 0,56,64,120,68,68,56,0
* 58 DATA 0,124,4,8,16,32,32,0
* 59 DATA 0,56,68,56,68,68,56,0
* 60 DATA 0,56,68,68,60,4,120,0

```

```

* 61 DATA 0, 48, 48, 0, 48, 48, 0, 0
* 62 DATA 0, 48, 48, 0, 48, 16, 32, 0
* 63 DATA 0, 0, 4, 8, 16, 8, 4, 0
* 64 DATA 0, 0, 0, 124, 0, 124, 0, 0
* 65 DATA 0, 0, 64, 32, 16, 32, 64, 0
* 66 DATA 0, 56, 68, 8, 16, 0, 16, 0
* 67 DATA 32, 16, 0, 56, 72, 72, 52, 0
* 68 DATA 0, 56, 68, 68, 124, 68, 68, 0
* 69 DATA 0, 120, 68, 120, 68, 68, 120, 0
* 70 DATA 0, 56, 68, 64, 64, 68, 56, 0
* 71 DATA 0, 120, 68, 68, 68, 68, 120, 0
* 72 DATA 0, 124, 64, 120, 64, 64, 124, 0
* 73 DATA 0, 124, 64, 120, 64, 64, 64, 0
* 74 DATA 0, 56, 68, 64, 76, 68, 56, 0
* 75 DATA 0, 68, 68, 124, 68, 68, 68, 0
* 76 DATA 0, 56, 16, 16, 16, 16, 56, 0
* 77 DATA 0, 4, 4, 4, 4, 68, 56, 0
* 78 DATA 0, 72, 80, 96, 80, 72, 68, 0
* 79 DATA 0, 64, 64, 64, 64, 64, 124, 0
* 80 DATA 0, 68, 108, 84, 68, 68, 68, 0
* 81 DATA 0, 68, 100, 84, 84, 76, 68, 0
* 82 DATA 0, 124, 68, 68, 68, 68, 124, 0
* 83 DATA 0, 120, 68, 68, 120, 64, 64, 0
* 84 DATA 0, 56, 68, 68, 84, 76, 60, 0
* 85 DATA 0, 120, 68, 68, 120, 72, 68, 0
* 86 DATA 0, 56, 68, 48, 8, 68, 56, 0
* 87 DATA 0, 124, 16, 16, 16, 16, 16, 0
* 88 DATA 0, 68, 68, 68, 68, 68, 56, 0
* 89 DATA 0, 68, 68, 68, 68, 40, 16, 0
* 90 DATA 0, 68, 68, 68, 84, 84, 40, 0
* 91 DATA 0, 68, 40, 16, 16, 40, 68, 0
* 92 DATA 0, 68, 68, 40, 16, 16, 16, 0
* 93 DATA 0, 124, 8, 16, 32, 64, 124, 0
* 94 DATA 0, 16, 40, 16, 0, 0, 0, 0
* 95 DATA 0, 0, 0, 56, 64, 64, 56, 16
* 96 DATA 0, 48, 64, 48, 72, 48, 8, 48
* 97 DATA 0, 16, 40, 68, 0, 0, 0, 0
* 98 DATA 16, 40, 0, 56, 72, 72, 52, 0
* 99 DATA 0, 32, 16, 8, 0, 0, 0, 0
* 100 DATA 0, 0, 56, 72, 72, 72, 52, 0
* 101 DATA 96, 32, 56, 36, 36, 36, 120, 0
* 102 DATA 0, 0, 56, 68, 64, 68, 56, 0
* 103 DATA 12, 8, 56, 72, 72, 72, 60, 0
* 104 DATA 0, 0, 56, 68, 124, 64, 56, 0
* 105 DATA 24, 36, 32, 112, 32, 32, 32, 0
* 106 DATA 0, 0, 60, 68, 60, 4, 4, 56
* 107 DATA 96, 32, 40, 52, 36, 36, 36, 0
* 108 DATA 16, 0, 112, 16, 16, 16, 124, 0
* 109 DATA 8, 0, 24, 8, 8, 72, 72, 48
* 110 DATA 32, 32, 36, 40, 48, 40, 36, 0
* 111 DATA 48, 16, 16, 16, 16, 16, 124, 0
* 112 DATA 0, 0, 168, 84, 84, 84, 84, 0
* 113 DATA 0, 0, 88, 36, 36, 36, 36, 0
* 114 DATA 0, 0, 56, 68, 68, 68, 56, 0
* 115 DATA 0, 0, 120, 36, 36, 56, 32, 32
* 116 DATA 0, 0, 48, 72, 56, 8, 8, 12
* 117 DATA 0, 0, 88, 36, 32, 32, 32, 0
* 118 DATA 0, 0, 60, 64, 56, 4, 120, 0
* 119 DATA 32, 32, 120, 32, 32, 36, 24, 0
* 120 DATA 0, 0, 72, 72, 72, 72, 52, 0
* 121 DATA 0, 0, 68, 68, 40, 40, 16, 0

```



```

* 122 DATA 0,0,212,84,84,84,40,0
* 123 DATA 0,0,68,40,16,40,68,0
* 124 DATA 0,0,68,68,60,4,4,24
* 125 DATA 0,0,124,72,16,36,124,0
* 126 DATA 16,32,48,72,120,64,48,0
* 127 DATA 32,16,0,72,72,72,52,0
* 128 DATA 32,16,48,72,120,64,48,0
* 129 DATA 0,0,32,84,8,0,0,0
* 130 DATA 0,0,0,255,0,0,0,0
* 131 DATA 0,0,0,0,0,0,0,0
* 132 DATA 0,0,0,0,0,0,0,0
* 133 DATA 0,0,0,0,0,0,0,0
* 134 DATA 0,0,0,0,0,0,0,0
* 135 DATA 0,0,0,0,0,0,0,0
* 136 DATA 0,0,0,0,0,0,0,0
* 137 DATA 0,0,0,0,0,0,0,0
* 138 DATA 0,0,0,0,0,0,0,0
* 139 DATA 0,0,0,0,0,0,0,0
* 140 DATA 0,0,0,0,0,0,0,0
* 141 DATA 0,0,0,0,0,0,0,0
* 142 DATA 0,0,0,0,0,0,0,0
* 143 DATA 0,0,0,0,0,0,0,0
* 144 DATA 0,0,0,0,0,0,0,0
* 145 DATA 0,0,0,0,0,0,0,0
* 146 DATA 0,0,0,0,0,0,0,0
* 147 DATA 2,0,3,240,2,1,36,244
* 148 DATA 2,2,3,136,4,32,32,36
* 149 DATA 4,91,19,2
* 150 FOR I=9460 TO 10391 :: READ N :: CALL LOAD(I,N):: NEXT I

```

\*\*\*\*\*

```

*
* 10 !*****
* 11 !* CHARSET TEST PROGRAM *
* 12 !*****
* 13 !
* 14 !
* 15 CALL CLEAR
* 16 CALL SCREEN(13)
* 17 CALL LINK("CHAR")
* 18 I=1
* 19 ACCEPT AT(24,1)SIZE(-28):A$
* 20 DISPLAY AT(I,1):A$
* 21 I=I+2
* 22 GOTO 19
*
*****

```

-----  
 J A M E S S T R I N G F E L L O W  
 -----

Programs For You To Type In

```

*****
*
* 10 !*****
* 11 !*
* 12 !*      KEYTEST
* 13 !*
* 14 !*      EXTENDED BASIC
* 15 !* WITH MEMORY EXPANSION*
* 16 !*
* 17 !*      CALL LINK("KEY")
* 18 !*
* 19 !*****
* 20 !
* 21 !
* 22 CALL INIT :: CALL CLEAR
* 23 CALL LOAD(16376,75,69,89,32,32,32,37,124,"",8194,38,140,63,248)
* 24 FOR I=9460 TO 9869 :: READ A :: CALL LOAD(I,A):: NEXT I
* 25 DATA 2,155,128,0,37,120,0,0,0,0,48,0,203,20,203,53,203,78,203,231,
*      204,71
* 26 DATA 38,42,204,228,205,29,205,75,205,96,0,10,255,255,128,128,128,128,
*      128,128,128,128
* 27 DATA 128,128,128,128,128,128,128,128,128,128,128,128,128,128,128,
*      128,128,128,255,255
* 28 DATA 32,32,32,32,32,80,82,69,83,83,32,65,32,75,69,89,32,32,32,32,
*      32
* 29 DATA 32,32,32,32,32,75,69,89,32,32,32,32,32,32,86,65,76,85,69,32,32,
*      32
* 30 DATA 32,32,80,82,69,83,83,32,82,69,68,79,47,69,83,67,65,80,69,32,32,
*      32
* 31 DATA 96,6,15,0,2,224,36,244,2,0,7,17,4,32,32,48,2,0,8,31,2,1
* 32 DATA 17,0,4,32,32,32,2,0,8,0,2,1,20,0,4,32,32,32,2,128,8,30
* 33 DATA 19,2,5,128,16,249,2,0,0,0,2,1,37,22,2,2,0,32,4,32,32,36
* 34 DATA 2,128,2,224,19,3,2,32,0,32,16,248,2,0,0,229,2,2,37,54,2,3
* 35 DATA 0,22,6,160,38,122,4,224,131,124,4,32,32,28,216,32,131,124,131,
*      124,19,248
* 36 DATA 2,0,1,69,2,2,37,76,2,3,0,22,6,160,38,122,2,0,1,139,208,96
* 37 DATA 131,117,176,96,37,120,4,32,32,32,4,196,209,32,131,117,9,132,2,3,
*      1,148
* 38 DATA 2,0,1,150,6,160,38,78,2,0,2,133,2,2,37,98,2,3,0,22,6,160
* 39 DATA 38,122,4,224,131,124,4,32,32,28,216,32,131,124,131,124,19,248,
*      152,32,131,117
* 40 DATA 37,122,19,22,152,32,131,117,37,121,22,240,4,96,37,170,193,68,4,
*      196,61,32
* 41 DATA 37,20,2,37,0,48,10,133,192,69,176,96,37,120,4,32,32,32,6,0,128,
*      192
* 42 DATA 20,241,4,91,2,224,131,224,4,224,131,124,4,96,0,112,208,114,176,
*      96,37,120
* 43 DATA 4,32,32,32,5,128,6,3,22,248,4,91,255,0
* 44 CALL LINK("KEY")
*
*****

```

Programs For You To Type In

```

*****
*
* 1 REM *****
* 2 REM *
* 3 REM *      SCREEN DUMP      *
* 4 REM *BASIC & MINI MEMORY *
* 5 REM *   FOR EPSON MX 80   *
* 6 REM *   CALL LINK("SCREEN")*
* 7 REM *
* 8 REM *****
* 9 REM
* 10 CALL INIT
* 20 CALL LOAD(28700,127,2,127,232,"",32744,83,67,82,69,69,78,125,20)
* 30 FOR I=32020 TO 32514
* 40 READ A
* 50 CALL LOAD(I,A)
* 60 NEXT I
* 70 DATA 216,32,152,2,126,252,6,224,126,252,216,32,152,2,126,252,6,224,
* 126,252,6,32
* 80 DATA 126,252,2,0,29,0,2,1,126,210,2,2,0,36,4,32,96,40,2,6,29,9
* 90 DATA 200,6,131,86,4,32,96,56,0,8,2,10,4,0,200,10,125,234,2,0,29,0
* 100 DATA 2,1,3,0,4,32,96,36,2,0,29,5,2,1,4,0,4,32,96,36,2,0
* 110 DATA 30,0,2,1,126,254,2,2,0,4,4,32,96,40,200,6,131,86,4,32,96,56
* 120 DATA 0,8,2,10,0,50,6,10,22,254,4,201,192,9,4,32,96,44,9,129,2,33
* 130 DATA 255,128,10,49,2,33,4,0,192,1,2,1,126,192,2,2,0,8,4,32,96,48
* 140 DATA 2,5,0,128,4,200,2,6,0,128,4,195,4,196,4,199,209,227,126,192,6,
* 199
* 150 DATA 129,71,17,5,161,6,97,197,6,199,216,199,126,192,5,131,8,22,21,
* 242,6,196
* 160 DATA 218,4,126,200,5,136,8,21,21,232,2,0,29,5,2,1,0,0,4,32,96,36
* 170 DATA 2,0,30,0,2,1,126,248,2,2,0,4,4,32,96,40,2,6,29,9,200,6
* 180 DATA 131,86,4,32,96,56,0,8,2,10,0,0,200,10,125,234,2,0,29,5,2,1
* 190 DATA 8,0,4,32,96,36,2,0,30,0,2,1,126,200,2,2,0,8,4,32,96,40
* 200 DATA 200,6,131,86,4,32,96,56,0,8,2,10,0,50,6,10,22,254,5,137,2,137
* 210 DATA 2,255,21,27,38,96,126,208,22,158,2,0,29,5,2,1,2,0,4,32,96,36
* 220 DATA 2,0,30,0,2,1,126,246,2,2,0,2,4,32,96,40,200,6,131,86,4,32
* 230 DATA 96,56,0,8,2,10,4,0,200,10,125,234,16,134,2,0,29,0,2,1,1,0
* 240 DATA 4,32,96,36,200,6,131,86,4,32,96,56,0,8,2,10,0,50,6,10,22,254
* 250 DATA 216,32,126,252,156,2,6,224,126,252,216,32,126,252,156,2,120,32,
* 131,124,131,124
* 260 DATA 2,10,0,50,6,10,22,254,4,91,0,0,0,0,0,0,0,0,0,0,0,0,0
* 270 DATA 0,0,0,0,0,31,0,18,30,0,255,0,0,0,0,26
* 280 REM *****
* 290 REM NAME OF PRINTER      RS232. PA=0. DA=8. BA=4800. CR
* 300 DATA 82,83,50,51,50,46,80,65,61,79,46,68,65,61,56,46,66,65,61,52,56,
* 48,48,46,67,82
* 310 REM *****
* 320 DATA 13,10,27,75,255,0,22,105,13,27,65,8,0
*
*****

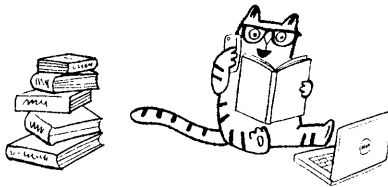
```

-----  
JAMES STRINGFELLOW  
-----

Programs For You To Type In

```
*****  
*  
* 10 REM *****  
* 20 REM *      TESTDUMP      *  
* 30 REM *To test screen dump*  
* 40 REM *DUMP MUST BE LOADED*  
* 50 REM *****  
* 60 CALL CLEAR  
* 70 CALL CHAR(32,"8142241818244281")  
* 80 CALL KEY(0,K,S)  
* 90 IF K(0)80 THEN 80  
* 100 CALL LINK("SCREEN")  
* 110 END  
*  
*****
```

Copyright JAMES STRINGFELLOW 1985



-----  
 J A M E S S T R I N G F E L L O W  
 -----

Programs For You To Type In

```

*****
*
* 10 !*****
* 11 !*
* 12 !*          CLOCK
* 13 !*
* 14 !*          EXTENDED BASIC
* 15 !* WITH MEMORY EXPANSION
* 16 !*
* 17 !*          After setting the
* 18 !*          time, type NEW
* 19 !*
* 20 !*          TO reset time
* 21 !*CALL LINK("S", "000000")*
* 22 !*
* 23 !*****
* 24 !
* 25 !
* 26 CALL CLEAR :: CALL PEEK(8198,A):: IF A <> 170 THEN CALL INIT
* 27 CALL LOAD(16368, 83, 32, 32, 32, 32, 32, 37, 152, 67, 76, 79, 67, 75, 32, 37, 24, "",
*      8194, 37, 226, 63, 240)
* 28 FOR I=9460 TO 9699 :: READ A :: CALL LOAD(I,A):: NEXT I
* 29 DATA 0, 50, 0, 0, 0, 0, 0, 128, 144, 144, 154, 144, 144, 154, 144, 146, 144,
*      150, 154, 148
* 30 DATA 1, 0, 96, 96, 0, 6, 205, 75, 205, 96, 33, 131, 37, 30, 200, 32, 37, 22, 131, 196, 2,
*      224
* 31 DATA 36, 246, 6, 32, 36, 244, 22, 45, 2, 0, 0, 50, 200, 0, 36, 244, 161, 202, 145, 201,
*      21, 38
* 32 DATA 209, 200, 5, 134, 152, 32, 37, 7, 37, 3, 21, 32, 216, 8, 37, 3, 5, 133, 152, 9, 37,
*      1
* 33 DATA 21, 26, 216, 8, 37, 1, 161, 74, 145, 96, 37, 7, 21, 20, 209, 72, 161, 10, 152, 32,
*      37, 5
* 34 DATA 36, 253, 21, 4, 145, 32, 37, 9, 21, 1, 209, 9, 145, 9, 21, 8, 209, 8, 5, 131, 152, 32
* 35 DATA 37, 5, 36, 253, 20, 2, 216, 8, 36, 253, 2, 0, 0, 22, 2, 1, 36, 252, 2, 2, 0, 9
* 36 DATA 4, 32, 32, 36, 2, 224, 131, 224, 4, 91, 2, 224, 36, 246, 2, 12, 0, 6, 2, 0, 0, 50
* 37 DATA 200, 0, 36, 244, 4, 192, 2, 1, 0, 1, 2, 2, 37, 15, 4, 32, 32, 20, 216, 13, 36, 253
* 38 DATA 209, 32, 37, 17, 193, 78, 216, 15, 37, 3, 209, 224, 37, 21, 184, 11, 36, 253, 177,
*      11, 161, 75
* 39 DATA 184, 11, 37, 3, 177, 203, 216, 0, 131, 124, 2, 224, 131, 224, 4, 96, 0, 112, 88, 79
* 40 CALL LINK("CLOCK")
* 41 DISPLAY AT(24,1):"TIME?000000" :: ACCEPT AT(24,6)SIZE(-6)VALIDATE
*      (DIGIT):T$ :: CALL LINK("S", T$)
*
*****

```

Copyright JAMES STRINGFELLOW 1985

-----  
 J A M E S S T R I N G F E L L O W  
 -----

Programs For You To Type In

```

*****
*
* 10 REM *****
* 11 REM *
* 12 REM *BASIC & MINI MEMORY*
* 13 REM * PAGECHANGE
* 14 REM *
* 15 REM *****
* 16 REM
* 17 REM
* 18 REM
* 19 CALL CLEAR
* 20 PRINT " DOUBLE PAGE SCREEN": " _____": " : : : : :
* "PLEASE WAIT " : : : :
* 21 FOR I=2048 TO 2815 STEP 8
* 22 CALL POKEV(I, 128, 128, 128, 128, 128, 128, 128)
* 23 NEXT I
* 24 A$=" ***FIRST PAGE***"
* 25 A=2278
* 26 GOSUB 47
* 27 A$=" TO SEE THE SECOND"
* 28 A=2402
* 29 GOSUB 47
* 30 A$=" PAGE PRESS ANY KEY"
* 31 A=2468
* 32 GOSUB 47
* 33 CALL PEEKV(-32254, X)
* 34 CALL CLEAR
* 35 PRINT " *** SECOND PAGE ***": : : : : : :
* 36 PRINT "YOU CAN GO TO THE FIRST PAGE": " QUICKLY": :
* " BY PRESSING ANY KEY": : : :
* 37 REM
* 38 CALL PEEKV(-32254, X)
* 39 CALL SCREEN(13)
* 40 CALL KEY(5, R, E)
* 41 IF E=0 THEN 40
* 42 CALL PEEKV(-32256, X)
* 43 CALL KEY(5, R, E)
* 44 CALL SCREEN(16)
* 45 IF E=0 THEN 43
* 46 GOTO 38
* 47 FOR I=1 TO LEN(A$)
* 48 CALL POKEV(A+I, ASC(SEG$(A$, I, 1))+96)
* 49 NEXT I
* 50 RETURN
*
*****

```

-----  
 J A M E S S T R I N G F E L L O W .  
 -----

Programs For You To Type In

```

*****
*
* 10 REM *****
* 11 REM *
* 12 REM *          LINES *
* 13 REM *          *
* 14 REM *BASIC & MINI MEMORY*
* 15 REM *          *
* 16 REM *LINES PROGRAM USING*
* 17 REM *    CALL LOADS *
* 18 REM * CALL LINK("LINES")*
* 19 REM *          *
* 20 REM *****
* 21 CALL INIT
* 22 CALL LOAD(28700,127,178,127,232)
* 23 CALL LOAD(32744,76,73,78,69,83,32,125,158,79,76,68)
* 24 CALL LOAD(32676,0,2,255,250,255,252,0,8,0,0,84,124)
* 25 DATA 192,50,17,3,4,32,96,52,16,251,4,91,4,32,96,36,16,2,216,1,140,0
* 26 DATA 6,2,22,252,4,91,192,96,127,174,2,33,29,107,19,253,200,1,127,174,
* 4,91
* 27 DATA 193,1,10,84,225,1,2,68,255,7,161,0,2,64,0,7,97,0,2,3,128,0
* 28 DATA 11,3,192,4,4,32,96,44,224,67,4,32,96,36,2,32,32,0,4,32,96,44
* 29 DATA 208,65,22,4,11,66,224,66,4,32,96,36,4,91,131,0,125,64,4,204,2,5
* 30 DATA 0,1,2,6,0,1,193,237,0,16,194,109,0,12,97,201,17,1,16,2,5,5
* 31 DATA 5,7,193,199,22,1,7,12,194,45,0,18,194,173,0,14,98,10,17,1,16,2
* 32 DATA 5,6,5,8,192,9,192,74,192,173,0,10,6,160,125,2,139,73,0,16,22,4
* 33 DATA 139,74,0,18,22,1,3,128,195,12,17,3,162,69,99,8,16,238,162,134,
* 163,7
* 34 DATA 16,235,2,224,131,32,2,2,127,116,6,160,124,214,2,0,27,0,2,1,208,0
* 35 DATA 4,32,96,36,2,0,24,0,4,194,4,193,6,193,4,32,96,36,6,193,5,129
* 36 DATA 5,128,2,129,1,0,17,247,5,130,2,130,0,3,17,242,4,192,4,193,2,2
* 37 DATA 24,0,6,160,124,226,2,0,32,0,2,1,0,0,2,2,24,0,6,160,124,226
* 38 DATA 2,0,7,17,4,32,96,52,4,224,127,176,4,195,2,6,0,128,2,7,0,96
* 39 DATA 2,8,0,211,2,9,0,163,4,192,5,192,200,0,127,164,5,192,200,0,127,
* 168
* 40 DATA 5,192,200,0,127,166,5,192,200,0,127,170,192,32,127,176,22,10,6,
* 160,124,242
* 41 DATA 2,65,0,15,193,65,2,133,0,2,20,2,2,101,0,2,161,160,127,164,161,
* 224
* 42 DATA 127,168,162,32,127,166,162,96,127,170,193,134,17,3,2,134,1,0,
* 17,4,5,32
* 43 DATA 127,164,161,160,127,164,194,8,17,3,2,136,1,0,17,4,5,32,127,166,
* 162,32
* 44 DATA 127,166,193,199,17,3,2,135,0,192,17,4,5,32,127,168,161,224,127,
* 168,194,73
* 45 DATA 17,3,2,137,0,192,17,4,5,32,127,170,162,96,127,170,4,32,125,60,
* 4,192
* 46 DATA 216,0,131,116,4,32,96,32,208,32,131,117,208,96,131,124,19,10,2,
* 128,5,0

```

```

* 47 DATA 22, 2, 4, 96, 127, 100, 2, 128, 67, 0, 22, 238, 5, 96, 127, 176, 2, 128, 255, 0, *
* 22, 233 *
* 48 DATA 5, 131, 2, 131, 0, 80, 22, 170, 4, 195, 2, 2, 0, 10, 2, 4, 255, 255, 6, 4, 22, 254 *
* 49 DATA 6, 2, 22, 252, 6, 160, 124, 242, 192, 65, 17, 3, 2, 65, 0, 7, 16, 2, 2, 97, 255, 248 *
* 50 DATA 200, 1, 127, 164, 6, 160, 124, 242, 192, 65, 17, 3, 2, 65, 0, 7, 16, 2, 2, 97, 255, *
* 248 *
* 51 DATA 200, 1, 127, 166, 6, 160, 124, 242, 192, 65, 17, 3, 2, 65, 0, 7, 16, 2, 2, 97, 255, *
* 248 *
* 52 DATA 200, 1, 127, 168, 6, 160, 124, 242, 192, 65, 17, 3, 2, 65, 0, 7, 16, 2, 2, 97, 255, *
* 248 *
* 53 DATA 200, 1, 127, 170, 4, 192, 4, 193, 4, 195, 2, 2, 24, 0, 6, 160, 124, 226, 2, 0, 32, 0 *
* 54 DATA 2, 1, 0, 0, 2, 2, 24, 0, 6, 160, 124, 226, 4, 96, 126, 44, 2, 2, 127, 132, 6, 160 *
* 55 DATA 124, 214, 3, 0, 0, 2, 4, 32, 0, 0, 0, 2, 1, 224, 2, 6, 3, 255, 4, 3, 5, 54 *
* 56 DATA 7, 23, 255, 255, 0, 0, 1, 240, 2, 0, 3, 255, 4, 1, 5, 96, 255, 255 *
* 57 FOR I=31958 TO 32656 *
* 58 READ A *
* 59 CALL LOAD(I, A) *
* 60 NEXT I *
* 61 CALL LINK("LINES") *
* *
*****

```

Copyright JAMES STRINGFELLOW 1985





-----  
 J A M E S S T R I N G F E L L O W  
 -----

Programs For You To Type In

```

*****
*
* 1 REM *****
* 2 REM *
* 3 REM *          CRAYON          *
* 4 REM *
* 5 REM * BASIC & MINI MEMORY *
* 6 REM *
* 7 REM * CALL LINK("CRAYON") *
* 8 REM *
* 9 REM *****
* 10 REM
* 11 REM
* 12 CALL CLEAR
* 13 CALL SCREEN(2)
* 14 PRINT "          CRAYON"
* 15 PRINT : : " USE ARROW KEYS TO DRAW"
* 16 PRINT : " 1 TO CHANGE SCREEN COLOUR"
* 17 PRINT : " 2 TO CHANGE PIXEL COLOUR"
* 18 PRINT : " 3 SLOW DRAWING SPEED"
* 19 PRINT : " 4 FAST DRAWING SPEED"
* 20 PRINT : " F TO FRAME SCREEN"
* 21 PRINT : " K TO CLEAR SCREEN"
* 22 PRINT : : "          PLEASE WAIT LOADING"
* 23 CALL SCREEN(13)
* 24 CALL LOAD(28702,127,224,"",32736,67,82,65,89,79,78,125,0,0)
* 25 DATA 2,224,112,184,4,224,131,116,2,0,7,17,4,32,96,52,2,0,1,240,4,32
* 26 DATA 96,52,2,0,2,6,4,32,96,52,2,0,3,255,4,32,96,52,2,0,4,3
* 27 DATA 4,32,96,52,2,0,5,54,4,32,96,52,2,0,27,0,2,1,208,0,4,32
* 28 DATA 96,36,2,0,24,0,4,194,4,193,4,32,96,36,5,128,2,33,1,0,2,129
* 29 DATA 0,0,22,248,5,130,2,130,0,3,22,243,4,192,4,193,4,32,96,36,5,128
* 30 DATA 2,128,24,0,22,250,2,0,32,0,4,32,96,36,5,128,2,128,56,0,22,250
* 31 DATA 2,0,0,2,4,32,96,52,2,0,1,224,4,32,96,52,2,3,0,128,2,4
* 32 DATA 0,96,2,5,0,1,2,6,160,0,2,15,19,136,6,160,127,22,3,0,0,2
* 33 DATA 3,0,0,0,192,143,6,2,22,254,4,32,96,32,4,193,192,96,131,117,2,129
* 34 DATA 0,69,22,2,6,4,16,116,2,129,0,88,22,2,5,132,16,111,2,129,0,68
* 35 DATA 22,2,5,131,16,106,2,129,0,83,22,2,6,3,16,101,2,129,0,82,22,3
* 36 DATA 6,4,5,131,16,95,2,129,0,90,22,3,6,4,6,3,16,89,2,129,0,87
* 37 DATA 22,3,5,132,6,3,16,83,2,129,0,67,22,3,5,132,5,131,16,77,2,129
* 38 DATA 0,75,19,161,2,129,0,49,19,13,2,129,0,50,19,25,2,129,0,51,19,29
* 39 DATA 2,129,0,52,19,29,2,129,0,70,19,29,16,181,2,133,0,15,22,1,4,197
* 40 DATA 5,133,192,5,2,32,7,0,4,32,96,52,2,2,85,240,6,2,22,254,16,166
* 41 DATA 2,38,16,0,2,2,78,32,6,2,22,254,16,60,2,15,19,136,16,156,2,15
* 42 DATA 2,88,16,153,194,67,194,132,4,195,4,196,6,160,127,22,5,131,2,131,
* 1,0
* 43 DATA 17,250,6,3,6,160,127,22,5,132,2,132,0,192,17,250,6,4,6,160,127,
* 22
* 44 DATA 6,3,21,252,6,160,127,22,6,4,21,252,192,201,193,10,4,96,125,220,
* 2,131
* 45 DATA 1,0,17,2,2,3,0,255,2,131,0,0,21,2,2,3,0,1,2,132,0,192
* 46 DATA 17,2,2,4,0,191,2,132,0,0,21,2,2,4,0,1,6,160,127,22,4,96

```

```
* 47 DATA 125,220,195,4,10,92,227,4,2,76,255,7,195,67,2,77,0,7,163,3,99,13 *
* 48 DATA 192,12,4,32,96,44,2,7,127,72,161,205,240,87,4,32,96,36,2,32,32,0 *
* 49 DATA 192,70,4,32,96,36,4,91,128,64,32,16,8,4,2,1,0,0 *
* 50 FOR I=32048 TO 32592 *
* 51 READ A *
* 52 CALL LOAD(I,A) *
* 53 NEXT I *
* 54 CALL HCHAR(23,1,32,32) *
* 55 PRINT " PRESS ANY KEY TO CONTINUE" *
* 56 CALL KEY(O,K,S) *
* 57 IF S=0 THEN 56 *
* 58 CALL LINK("CRAYON") *
* *
*****
```

Copyright JAMES STRINGFELLOW 1985

~~~~~



**"Until the virus has been identified and removed,  
IT has issued an immediate ban on any use of  
e-mail attachments. For more details, please  
refer to the attached document."**

-----  
P L O T T I N G   A   S I N E   W A V E  
-----

p e t e r   B r o o k s

J a n u a r y

I began a writing a simple explanation of how the calculations involved in plotting a sine wave are derived, and promptly fell over an obstacle.

While we tend to work in DEGREES, the computer (most of them, anyway) uses RADIANS. (There are also things called GRADIANS, but you only want one headache at a time!).

Degrees, you may remember, number 360 in a complete circle. However, there are  $2\pi$  radians in a circle. You may remember  $\pi$  (I can't reproduce its Greek symbol here) as a constant, being the ratio between a circle's radius and its circumference (or strictly, between its diameter and circumference).

If there are 360 degrees in a circle, and  $2\pi$  radians in a circle, then one radian is equivalent to  $360/(2\pi)$ , or  $180/\pi$  degrees.

$\pi$  is 3.14159265358... or thereabouts, so  $180/\pi$  is about 57.296 degrees.

All of this is just to explain that if you ask the computer to PRINT the sine of 90, it won't give you the answer for 90 degrees, but for 90 radians.

The sine of 90 degrees is 1, while the sine of 90 radians is about 0.899 (at least, on the TI it is. On my calculator it is outside the normal range, so the 99s must use some form of modulus arithmetic to bring it within range).

In order to convert from degrees to radians, we must divide by 57.296 before using functions like SIN() and so on.

90 degrees therefore becomes  $90/57.296$  or about 1.571. This is the value you would use with SIN() and the answer would be close to 1.

That's the first obstacle (did I mention there was more than one?). The second is not so great. In the example program later, I have made use of a shortcut to generate the 57.296 accurately.

Using radians, the "arc-tangent", or anti-tangent if you will, can be used to generate  $\pi$  to a fair degree of accuracy. It turns out that  $\text{ATN}(1)$  is exactly one quarter of  $\pi$ , so  $4\text{ATN}(1)$  is the same as  $\pi$ .

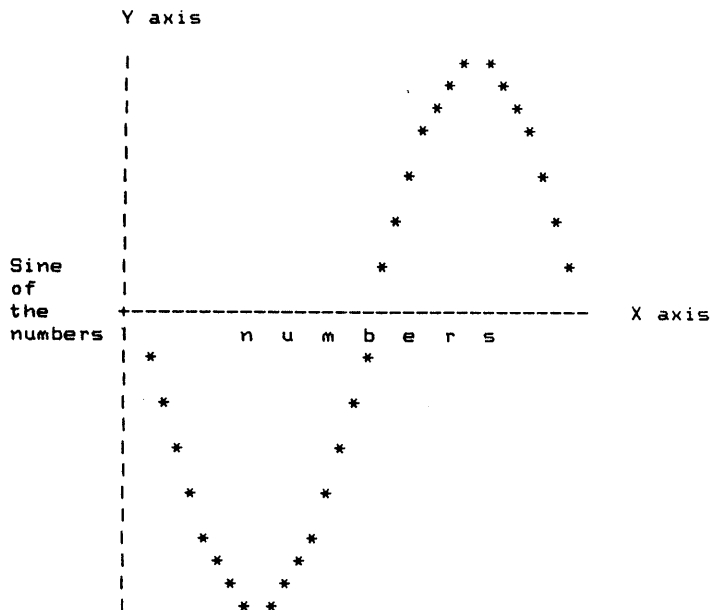
Therefore  $360/(2\pi)$  or  $180/\pi$  is the same as  $45/\text{ATN}(1)$ , which is why the program uses a constant K derived from that small equation.

Why should I go to these lengths? Because it means that I can use degrees (much simpler in this case) when discussing the Maths involved in plotting a sine wave.

Down To Business  
-----

If you have read previous articles in other newsletters, or read my book (cough, cough), you might have seen the sine wave plotted using a special subroutine to give a form of high resolution, and wondered how on Earth the equations were derived. (Go on, then, wonder!).

If you plot, on graph paper, the results of taking the sine of a series of numbers on one axis, and the numbers themselves on the other axis, you will get a graph which looks something like this:



The above graph illustrates a plot of  $Y = \text{SIN}(X)$ ; that is exactly as you would see it in an elementary BASIC program, with  $X$  varying - perhaps in a loop, and the values for  $X$  and  $\text{SIN}(X)$  (i.e.  $Y$ ) would be used with a PLOT command or whatever.

As you will see, things are far from being that simple, largely due to the complexities of the display.

Let's begin by looking at some actual values. Here is a small table which lists a short series of numbers - from zero to four - and the sine of those numbers (working in degrees - we haven't got to the yucky part yet!).

X	SIN(X)
0	0.0000
1	0.0174
2	0.0349
3	0.0523
4	0.0698

These are very small values. Let's look at the full 360 degrees in steps of 30 degrees:

X	SIN(X)
0	0.00
30	0.50
60	0.87
90	1.00
120	0.87
150	0.50
180	0.00
210	-0.50
240	-0.87
270	-1.00
300	-0.87
330	-0.50
360	0.00

As X increases, so SIN(X) goes through a cycle. As X approaches 90 degrees, SIN(X) approaches 1. As X passes 90 and approaches 180, SIN(X) reverses its direction. Once X passes 180 and heads towards 360, SIN(X) produces a similar set of results, only as negatives this time.

(It might interest you to know that SIN(45) is equivalent to the square root of 0.5, or 1 over root 2).

Now to confuse you. You may already be aware of the use of RND. This function produces a number which is less than 1 but greater than 0 - in other words, a fraction. When we want to make use of the RND fractions, we multiply or divide them by a chosen number, and then perhaps add or subtract another number, until we have something lying within a required range.

Thus to get whole numbers, or INTEGERS, which lie between 1 and 6 (for a set of dice, perhaps) we would use INT(RND\*6+1) or INT(RND\*6)+1 - they both have the same effect.

Similarly, we can multiply or divide the numbers obtained with SIN(), and add or subtract other values until we get something which we can use.

So what do we need from the fractions produced by SIN() ?

On the graph, simply plotting SIN(X) will mean that our Y axis will have to be marked in tiny steps - perhaps 0.1 - and also include negative numbers.

If however we were looking to plot directly on the screen we would need much larger numbers, especially when using HCHAR(), which is what we will use to try and make life a bit simpler.

The resulting "points" plotted using HCHAR() will be enormous, but you should get the rough idea.

The values for X will range from 1 to 32 - the full width of the screen.

Y is not so easy. If you look at the range of SIN(X), its lowest value is -1 and its highest is 1. We want something which fits in the range 1 to 24, so as to cover the full range of rows on the screen.

-1 and 0 would cause problems with HCHAR(), so let's begin by shifting the range -1,0,1 up a bit by adding 2. This makes it 1,2,3. The first step is therefore SIN(X) + 2.

Values of 1, 2, and 3 and fractions in between are not going to give very exciting results on screen, so let's "expand" the range by multiplying by something. Whatever number we use, it must not produce a value greater than 24, so if we used 8 that would be fine. The range after multiplying by 8 is 8 to 24 with fractions in between, which is not fantastic but it will do for now. The equation has now built up to:

$$8 * (\text{SIN}(X) + 2)$$

and because of the order in which the computer does its arithmetic, the SIN(X) will be added to 2 before being multiplied by the 8.

The use of INT() will provide us with integers for use with HCHAR(), which although it would accept fractions without quibble, ought ideally to be passed only integers to work with.

This gives us:  $\text{INT}(8 * (\text{SIN}(X) + 2))$

or:  $\text{INT}(8 * \text{SIN}(X) + 16)$

An article later in TI-LINES will explain how you can reduce equations in this way.

The resulting values can be used in this program (note how K is used to convert the X value from degrees to radians):

100	CALL CLEAR	Clear screen
110	K=45/ATN(1)	Set up constant for radian conversion
120	FOR X=1 TO 32	Set up loop
130	Y=INT(8*SIN(X/K)+16)	Calculate row from SIN(column)
140	CALL HCHAR(Y,X,42)	Plot using asterisk (ASCII 42)
150	NEXT X	End of loop
160	GOTO 160	Hold on screen

Hmm. What went wrong? Where are all the pretty curves?

The answer lies with X. Going back to the last table, you can see that the maximum value for X of 32 means that the SIN(X) is not going to be much over 0.5. Putting that value in the equation:

$$\text{INT}(8 * (0.5) + 16) = 20$$

The largest row number is going to be 20.

The smallest value of 1 for X means that SIN(X) will yield something very close to 0:

$$\text{INT}(8*(0)+16) = 16$$

The smallest row number is going to be 16, and between 1 and 32 for X means that SIN(X) will only be a third of the way between 0 and 1, hence the odd slowly stepping-downward line.

One way to improve matters is to increase the range of X to 360 instead of 32. This will make sure that SIN(X) gets to evaluate the full cycle.

But... CALL HCHAR() will have a nervous breakdown once X gets above 32.

The answer? Use X as a loop of 1 to 360, but DIVIDE it by a constant when using HCHAR().

We therefore need to find out what might be a suitable constant.

Let's look at what we have, and what we need:

We will have X varying from 1 to 360, but HCHAR() needs a variation in whole numbers, or integers, ideally of 1 to 32. We will need an equation which allows our constant - call it Z for now - to divide any number between 1 and 360 and produce a result between 1 and 32.

The maximum values, 360 and 32, define the limit:

$$\frac{360}{Z} \text{ must equal } 32, \text{ so:}$$

$$\frac{360}{32} \text{ will equal } Z$$

which makes  $Z = 11.25$ .

This is fine for  $X = 360$ . At the end of the loop, when X is 360,  $X/Z$  will be 32. But at the beginning, when  $X=1$ ,  $X/Z$  will be 0.0888..., and as we are dealing with whole numbers, that is equal to 0. Try using 0 with HCHAR(), and the result is an error.

However, we can manipulate the numbers to our heart's content:

To recap:

	<u>WE HAVE</u>	<u>WE WANT</u>
	X = :	X/Z = :
Minimum	1	1
Maximum	360	32

Try this on paper:

$\text{INT}((X-1)/11.25+1)$                       gulp!

It begins by taking 1 from whatever value X has. Then it divides by 11.25 - our constant - and then adds 1. Finally, the result is converted to an integer (no fractions).

If X = 1,

$\text{INT}((1 - 1) / 11.25 + 1) = 1$

If X = 360,

$\text{INT}((360 - 1) / 11.25 + 1) = 32$

In fact, we can alter our X loop slightly, and make life a little easier.

Technically, our loop of 1 to 360 is not quite correct. The lowest value for X should really be zero, which also means that the highest will not be 360 but 359.

This makes the calculation involving Z easier too:

$\text{INT}(X / 11.25 + 1)$  or  $\text{INT}(X / Z + 1)$

We end up with:

```
100 CALL CLEAR
110 K=45/ATN(1)
120 Z=11.25
130 FOR X=0 TO 359
140 Y=INT(8*SIN(X/K)+16)
150 CALL HCHAR(Y,INT(X/Z+1),42)
160 NEXT X
170 GOTO 170
```

If you run this you will find that the result is a little too awkward. This is caused by too many points all trying to be fitted into too small an area. One way around it is to space out the X values by making the computer evaluate fewer of them, but still keep to a 0 to 359 loop. You can achieve this by adding to the end of line 130 the phrase STEP 12.

However, this is not the end of the story. The use of  $8*\text{SIN}(X/K)+16$  may not produce sine waves in exactly the proportion that you require, and you may have to experiment with the 8 and the  $\text{SIN}(X/K)$  to arrive at a satisfactory solution. As our use of  $8*\text{SIN}(X/K)+16$  forces the values produced to lie in the range 8 to 24, this is highly likely.



One solution is to allow the basic equation to generate positive and negative values, and then add these to a number which marks the vertical midpoint of the screen.

The midpoint of the 24 rows is NOT 12, as you might have thought. If you take row 12 as your midpoint, there are 11 rows above it, but 12 rows below it.

The "true" midpoint is a position at the junction of rows 12 and 13. We'll call this position 12.5 (halfway between the two).

We therefore have:

$$Y = 12.5 + \text{SIN}(\text{something})$$

The X value (converted eventually from degrees to radians by division by K) ranges from 0 to 359, and thus will allow SIN() to generate the full range of values associated with it, namely from -1 to +1.

We will therefore need to multiply that range of -1 to +1 by a value which, when the result is added to 12.5, does not exceed 24 nor drop below 1.

That value is 11.5.

$$12.5 - 11.5 = 1$$

$$12.5 + 11.5 = 24$$

We now have:

$$Y = 12.5 + 11.5 * \text{SIN}(X / K)$$

Note the difference in numbers - multiplying by 11.5, adding 12.5, compared with our previous multiplication by 8 and addition of 16

One final thing, to tidy up the values.

The use of INT() means that an integer will be used with CALL HCHAR(), and hopefully avoid any odd effects which might lead to errors. However, INT() has some undesirable effects under certain circumstances. If the expression produces a value of say 23.9, this should clearly be called 24. INT() will make it 23, and so a process known as "rounding up" should be used. This is straightforward, and simply involves the addition of 0.5 before the use of INT() - you may have come across this before in previous articles on other topics.

The "rounding up" rule is that fractions of 0.5 and above should cause the number under consideration to be rounded up to the next whole number. Thus 23.5 will rounded up to 24, as will 23.9. Fractions less than 0.5 are rounded down, so that 23.3 will be treated as 23. The addition of 0.5 to any number, followed by the use of INT(), performs the rounding up and rounding down exactly according to the rules.

The new line 140 is therefore:

```
140 Y=INT(13+11.5*SIN(X/K))
```

and line 150 likewise:

```
150 CALL HCHAR(Y,INT(X/Z+1.5),42)
```

The next article will look at the result of using functions other than SIN() and at the effect of adding sine, and other, waves.

~~~~~

-----  
BULLETIN BOARD  
-----

SALE FOR SALE FOR SALE FOR SALE FOR SALE FOR SALE FOR SALE FOR SALE FOR SALE

oooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooo

A complete system is offered for sale by former OTIUser GERARD MURRAY. It consists of two consoles, Peripheral Expansion Box, 32K RAM card, drive controller with SSSD 40T full height drive, data cassette recorder, Editor/Assembler, MiniMemory, Extended BASIC, Disk Manager 1, Star Trek, Return To Pirates Isle, Indoor Soccer, Speech synthesizer, TI Joysticks, and a mass of other programs too great in number to list but including Forth with its manual, most of the more popular games like Munchman, Invaders, and so on, as well as utilities like Mailing List. I counted around 50 programs. There are also a number of books, including Starting Forth, Getting Started With The TI-99/4A, NEC 30 hour BASIC, and one on Assembly Language. There are a number of copies of 99er magazine (and copies of TI\*MES and of course TI-LINES). 28 disks, a number of cassettes, a disk box, and a few odds and ends including a dust cover make up the rest of the items.

The asking price is £480. Contact Gerard at 20 Queensborough Gardens, Hyndland, GLASGOW G12 9PP, or on 041 339 2798 after 6 pm any evening (Gerard says he is a night owl and doesn't hit the sack until gone midnight, so late calls may be in order).

oooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooo

OTIUser SYD MICHEL is selling his ALPHACOM 42 printer, with two rolls of paper. The printer has a built-in TI interface which allows it to be plugged into the computer via the "expansion port". It has a standard 40 characters per line, and prints at 80 characters per second, and unlike the TI Thermal printer it uses the full 8 x 8 dot matrix, which permits the use of "sideways" printing to allow a wide range of User-defined tpestyles and probably also screen dumps (if you work at it).

The asking price is £50. Contact Syd on HELENSBURGH 71967.

~~~~~

-----  
CHANGE XBASIC EDIT SCREEN COLOURS  
-----

Taken from the newsletter of the SASKATOON TI COMPUTER CLUB

There are lots of possible uses for the thing including games, but the real feature of this program is that it changes the colour of the EDIT Mode screen as well! Yes, no more black on cyan if you don't want it!!

The colour change is inserted into the USER-DEFINED INTERRUPT (UDI) and is constantly "re-performed" every 1/60 of a second. This makes it seem like the EDIT Mode screen has been changed. In order to return control of the colour commands CALL SCREEN and CALL COLOR, you must load the UDI with zeroes.

Along with the program that loads in the original routine, below are two demo routines to show off your new screen colours.

Of course, you don't need a program to change screen colours once the original file is loaded. All you have to do is poke a single byte value into CPU address 9460. This value is found by doing the following:-

Foreground colour (0 to 15)\*16  
plus Background colour (0 to 15)

PROGRAM NO.1: SCRNCOLR/X  
-----

```
100 !  
110 !  
120 ! SCREEN COLOUR  
130 !  
140 !  
150 !  
160 ! 11/84  
170 !  
180 ! SUBFILE 99  
190 !  
200 CALL CLEAR :: CALL INIT  
210 MEM=9459  
220 !  
230 ! LOAD IN PROGRAM  
240 !  
250 FOR I=1 TO 50  
260 READ X  
270 CALL LOAD(MEM+I,X)  
280 NEXT I  
290 !
```

```

300 ! START UP PROGRAM
310 !
320 CALL LOAD(8194,37,38,"",-31804,36,246)
330 END
340 !
350 ! PROGRAM DATA
360 !
370 DATA 244,0,2,1,0,135,208,96,36,244,216
380 DATA 1,140,2,6,193,216,1,140,2,2,1,0,72
390 DATA 216,1,140,2,6,193,216,1,140,2,2
400 DATA 0,0,32,216,32,36,244,140,0,6,0
410 DATA 22,251,4,91

```

PROGRAM NO. 2: CLRDEMO1/X

---

```

100 !
110 !
120 ! COLOUR CHANGE
130 !
140 ! DEMONSTRATION
150 !
160 !
170 !
180 ! 11/84
190 !
200 ! SUBFILE99
210 !
220 !
230 ! NOTE:
240 ! YOU MUST HAVE
250 ! ALREADY LOADED &
260 ! RUN "SCRNCLR/X"!
270 !
280 !
290 CALL CLEAR :: RANDOMIZE :: DIM C$(15)
300 M$="Screen Colour Change Demo"
310 !
320 FOR C=0 TO 15 :: READ C$(C):: NEXT C
330 !
340 !
350 DISPLAY AT (1,1):M$ :: DISPLAY AT(2,1):RPT$("_",LEN(M$))
355 DISPLAY AT(10,1):"FRGRND COLOUR:"
356 DISPLAY AT(16,1):"BKGRND COLOUR:"
360 !
370 FC=INT(RND*15)
380 BC=INT(RND*15)
390 DISPLAY AT(10,14):C$(FC)
400 DISPLAY AT(16,14):C$(BC)
410 !
420 CVAL=FC*16+BC
430 CALL LOAD(9460,CVAL)
440 !
450 GOTO 370
460 !
470 DATA Transparent,Black,Medium Green,Light Green
480 DATA Dark Blue,Light Blue,Dark Red,Cyan
490 DATA Medium Red,Light Red,Dark Yellow,Light Yellow
500 DATA Dark Green,Magenta,Grey,White

```

PROGRAM NO. 3: CLRDEMO2/X

---

```
100 !
110 !
120 ! COLOUR CHANGER
130 !
140 !
150 !
160 ! 11/84
170 !
180 ! SUBFILE99
190 !
200 !
210 ! NOTE;
220 ! YOU MUST HAVE
230 ! ALREADY LOADED &
240 ! RUN "SCRNCOLR/X"!
250 !
260 !
270 CALL CLEAR
280 M$="SCREEN COLOUR CHANGER"
290 I$=" "
295 BG=5 :: FC=1
300 !
310 DISPLAY AT(1,1):M$ :: DISPLAY AT(2,1):RPT$("_",LEN(M$))
320 !
330 FOR I=1 TO 8
340 READ C1,M1$,C2,M2$
350 DISPLAY AT(3(2*I),1):USING I$:C1,M1$,C2,M2$
360 NEXT I
370 !
380 DISPLAY AT(22,1):"BKGRND:"BC :: ACCEPT AT(22,9)VALIDATE(DIGIT)
```

---

-----  
L E T T E R S  
-----

STEPHEN SHAW has written to me about a number of things which might be of interest to other OTIUsers.

You may have seen the label FREEWARE on some of the items being offered for sale lately. FREEWARE is not exactly the same as PUBLIC DOMAIN, although the principle is similar. In the case of FREEWARE, the author provides you (indirectly in most cases) with a copy of his work, and requests that if you find the program to be useful or entertaining would you consider sending him a small contribution towards the cost of production (not the cost of copying but the actual cost of creation). If you don't find the program to be good at all, then you send nothing but you still keep the program.

Quite often, the sums requested are so small that there are difficulties in obtaining the necessary foreign currency (as most FREEWARE comes from overseas) at an acceptable exchange rate, due largely to the massive charges levied by the Exchange.

Stephen is therefore offering to act as a collection point for money to be sent abroad - you send him the UK currency (as a cheque or whatever) and the name of the intended recipient, and he will collate all the sums and from time to time convert them into the necessary currency and send them off. He has always sent off donations himself so this is not a fly-by-night operation!

Stephen asks that if any OTIUsers are using FREEWARE and haven't yet sent anything would they consider doing so? If \$10 is too much, how about £1? Or even a letter telling the author how much you enjoy his program but you are too broke to send anything?

If you have any enquiries concerning Stephen's Public Domain and Freeware Collection, contact him at 10 Alstone Road, STOCKPORT, SK4 5AH and please include a stamped, self-addressed envelope for any reply.

-----  
U N I V E R S A L   M O D U L E  
-----

A powerful static RAM module: 48K RAM    8K ROM  
-----

Reproduced from TINS newsletter, October 1985

-----  
M A X I M E M  
-----

This module works on both the TI-99/4 and TI-99/4A and expands the memory to 96K RAM.

MAXIMEM gives you:

- ability to dump any module to disk
  - ability to run any module dumped to disk
  - works from computers power supply....no extra power supply required
  - 2 sections of 48K RAM: 32K GRAM 16K RAM (added to the 16K VDP RAM standard 32K RAM expansion card). The result is a full 96K RAM available to the User
  - starts automatically thanks to an 8K GROM that displays choices:
    - 1 - TI BASIC: unchanged
    - 2 - MAXIMEM: offers catalogue of modules contained on disk
    - 3 - EDITOR-ASSEMBLER: an improved TI Editor-Assembler
  - stays plugged into the console. No further module needed
  - holds data even after "Quit" is typed
  - loads many modules giving following choices:

(-or-)

1. EDITOR ASSEMBLER	1. TI-WRITER
2. EXTENDED BASIC	2. PRINCESS & THE FROG
3. ZERO ZAP	3. EDITOR-ASSEMBLER
	4. CONNECT FOUR or
- and so on -

Dimensions: like Widgit

Material required: computer, disk drive and controller, 32K

Price: Just \$199 (CDN). Add postage/insurance. Allow one month for delivery.

Contact: GUY GOURNAY, 146178 CAN INC, 933 Delorimier, Longueuil, PQ J4K 3M8 or phone 514-651-7280 for further information.

Footnote:

This hardware is very similar to Craig Miller's GRAM CRACKER, and is much cheaper .... and CANADIAN! Incidentally, it was also produced BEFORE Craig's unit, and is not a copy of it!

~~~~~

Growing old gracefully is probably a good thing to which to aspire, but the production schedule of TI-LINES militates against such a process. At the time of writing (1st February) I am rushing to get the January issue into its final state (at 3 am!) ready to begin the production of the February issue immediately after. The intention is to get both issues to the photocopier at the same time, so that they can be compiled and posted at the same time. I am cheating a little: JENNY KEARNE has typed up a wealth of material for me over Christmas, and is continuing to do so - there is a stack of articles just waiting for me to find the time to reformat, and generally tidy up - so it is not as if I am having to type everything from scratch in nine hours flat. However, the processes involved in sorting articles into some semblance of order, checking spelling, confirming what any author might claim (where feasible), and printing out so that everything looks neat and pleasing to the eye, all take their toll. During these final hours I reckon I must live at twice the normal rate - I certainly feel twice as old after each issue is finally completed...

Because of the rush to try and get this issue finished, some articles which would have been of interest to owners of minimal systems (jargon for the console and a set of cassette leads) have had to be held over, as have the series' on Sorting & Searching and the Home Computing Weekly articles.

These may have to be put back to the March issue, but rest assured that they will appear. Probably in June...

I mentioned in the Editorial that 1986 was shaping up well, OTIU-wise, and it has continued to do so. We welcome J. K. BOUDIER, AHMED MOHAMMED, D. P. DAFFIN, MIKE GODDARD, and BILL MORAN. Once their eyes have refocused after the marathon reading sessions that they have been forced through (by being sent half a dozen TI-LINES all at once, when one alone is enough to send most folks reaching for the Ibuprofen), I look forward to receiving some interesting articles.

I don't know how many of you managed to catch the TV program MICROLIVE the other day, but there were some fascinating items concerning flat screen technology. I was particularly impressed with the latest flat screen system being developed by a group specialising in fibre optics. Their screen does not require constant refreshing - indeed, they had a demonstration screen which had received an image some months earlier (I thought I heard old Mac say three years, but I must have been hearing things) and it was still there even though the screen was no longer connected to the mains. They have only monochrome screens at present, but colour versions are planned for the very near future. Instead of the usual high tension voltages used in the usual TV sets, the new system only requires 200 volts to update a screen image, which puts it outside the LCD power bracket, but this is not a drawback.

What I want to know is: how do you switch the bleeding thing off...?

Good programming,

Peter Brooks