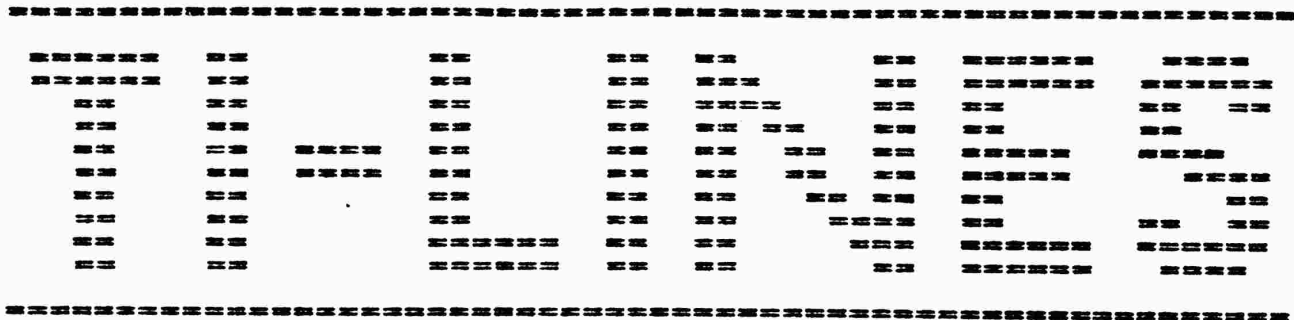


E N D   O F   V O L U M E   1  
T I M E   T O   R E N E W   Y O U R   S U B S C R I P T I O N



Volume 1, Issue 14

May 1st., 1985

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  
61 The Avenue  
Kennington  
OXFORD  
OX1 5PP

Tel: OXFORD 730044. Please do not phone after 10 pm

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

TI-LINES is available on Associate subscription to Users resident outside Oxfordshire, for £10 p.a. Overseas subscriptions are by arrangement. 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, or in a form which is as legible as possible. Art work 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

P a g e

EDITORIAL.....4

The Last Of The First. Services. Its On The Cards.  
 A Tale Of Intrigue. Oops. Hello Folks. Oops II.  
 Divide And Conquer. Memory Man ? Moronic Med-Recs.  
 Unprotection Racket. 4ABC. TIHOME SOFTWARE COLLECTION.  
 A Matter Of Creation. Faulty Copier.

BRIGHTON REPORT.....11

A Slightly Tongue-In-Motorcycle-Welly Saga About A Bunch Of  
 Ordinary Folk Let Loose In Brighton...

STICC.....14

From The Newsletter Of The SASKATOON TI COMPUTER CLUB  
 Changing Colours In The Editor/Assembler By Terry Atkinson

BULLETIN BOARD.....16

Lots Of Useful Things For Sale

CONTACTS.....17

A Few More OTIUers Putting Their Necks On The Chopping Block

INTERFACING THE TI-99/4A.....18

Part 1: INPUT By DAVE HEWITT

DAVID BROWN.....22

More For The Younger OTIUer

CIN-DAY USER GROUP.....24

From Their Newsletter Comes A Brief But Very Useful Overview  
 Of The Use Of The Disk Directory Sectors

GENERAL DESCRIPTION OF THE LOAD SUBPROGRAM.....	25
GENERAL DESCRIPTION OF THE SAVE SUBPROGRAM.....	26
9T9 USER GROUP TORONTO.....	27
The Most Useful Addition To The Repertoire Of The Experimenter The LOAD INTERRUPT SWITCH By John Bannister	
ENHANCED BASIC.....	29
The Closing Stages: CALLs L() and S()	
MSP 99.....	32
By T. D. Bell, Part 1 Of A Discussion Of Disks And Drives	
LETTERS.....	35
From Brian Abrams, Stephen Shaw, And John Roe	
THE PHILADELPHIA AREA TI-99/4A USERS GROUP.....	39
From Their Newsletter: Cleaning Your Keyboard By Ken Chandler	
UNDOCUMENTED FACILITIES.....	41
From Information Supplied By Howard Greenberg, Concerning The TI-Writer	
CLOSE FILE.....	42
One Step Forward, Two Leaps Back	

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

THE END OF THE BEGINNING  
-----

This issue marks the end of the first volume of TI-LINES and also the fourteenth month of the existence of OXON TI USERS. As of May 12th., (i.e. today) we boast some 90-odd members and after I have run a circular out to a mailing list of former TIHOMERs we should increase beyond, ooh, say the hundred mark. I hope welcome as many as possible to the second volume, and to try and combat some of the more insidious gremlins delaying my efforts to publish on time.

As a result of the two-tier subscription system a few current OTIUsers have bending my arm a little and persuaded me to include them as Oxon residents - either by fibbing about their address, or by suggesting that living close to Oxfordshire counts as being an Oxonian. If the cost of production of TI-LINES was not so high I would be happy to make everyone a "local" member, but the subsidy I have forked out to date amounts to over £1000 over the last fourteen months and as you will appreciate I cannot manage this ad infinitum.

From JUNE 1986 therefore all subscriptions will be set at the same level - £10 per year - and I hope to keep it at that level for as long as possible. It may still involve some small subsidy on my part, but at a much more acceptable level. This will enable me to honour my acceptance of 85/86 subscriptions from local and Associates alike, although I have yet to weather the coming year as far as subsidy goes. This is one reason for expanding the group's size: if I time it right, I should be able to produce a 20 page newsletter every month for a reasonable sum and thus avoid total bankruptcy!

Lately some pressure was put on me by a number of individuals to increase the subscription to OTIU but this has not been the cause of my decision. A recent chat with my financial guru left me with the warning that the Inland Revenue would not accept such a deliberate loss-making venture as being tax-deductable and so I must make some obeisance to The System.

I won't bore you with the details, but I will be looking at ways of cutting down on wasted space in the pages of this newsletter (one victim will probably be the title page). The result may well look cramped, but the aim is to provide as good a source of information as is possible. I sincerely hope you like it.

~~~~~  
SERVICES TO OTIUsers  
-----

Some of our august gathering are beginning to flex their muscles and the outcome has been an increase in the amount of UK-originated hardware available. If you look at the BULLETIN BOARD (not yet, not yet) you will find DAVE HEWITT (who was NOT responsible for the modulator modification article!) and GORDON PITT have come up with some useful items. Indeed, my system is now the richer for having a second disk drive which has been specially kitted out with a case and power supply through Gordon, who has also shown me a twin psu which looks very professionally produced. Also on the Bulletin Board you will find OTIUser RICHARD SIERAKOWSKI offering some very cheap double-sided, double density disk for sale. I have some (at half the price it cost me to buy a similar item from a local computer shop) and they are working away happily (if disks can ever be said to be happy...).

I hope to have further news of other goodies shortly and of course OTIUsers will read all about it here (provided my memory works well enough to prompt me to read all the little bits of paper with notes on).

## A READING OF THE CARDS

-----

RICHARD BLANDEN of CALL (3) and Disk Controller Subprogram fame has done it yet again. It seems that the P CODE CARD contains up to 48K of GROM wherein may lie the Operating System (BIOS) of the dreaded UCSD PASCAL system. It seems that with the right knowledge and an Extended BASIC module you could feasibly unlock some of the power of that card and make use of it in non-Pascal programs.

Following on from that, Richard has made the discovery of a fault in the Pascal Utility which formats disks for use with that language. It seems that in the past when owners have attempted to initialise a disk as double-sided the system has ignored their instructions, gone its own sweet way, and initialised the disk as single sided - but made entries in the appropriate sector indicating the existence of a second, initialised, side.

What happens is that the Formatting utility cleverly accepts the command to format double-sided, and then promptly overwrites that command internally and replaces it with a single-sided default.

Richard has worked out a solution, and if you contact him he will be happy to pass it on to you. His address is:

RICHARD BLANDEN  
2 MELODY CLOSE  
ARBOUR LANE  
WINNERSH  
WOKINGHAM  
BERKS

All things being equal I will publish some of Richard's work on both Sector Access using the disk controller subprograms and accessing the VDP registers using PEEKV (not POKEV notice).

In fact, RICHARD SIERAKOWSKI believes that the MYARC disk controller card may well possess the same additional subprograms as the TI card. I will report his findings in the next issue.

~~~~~

## A MATTER OF INTRIGUE

-----

At the beginning of April I was the recipient of a rather nasty letter from one of the software houses serving the TI-99/4A - not, I hasten to add, a member of OTIU. The writer seemed to have a considerable misunderstanding of the Copyright Act, and despite my half-hour conversation with him, persists in the belief that I am conspiring with another software house in an attempt to damage his sales.

After talking to a few individuals, and taking some legal advice just to make sure that the idiot plaintiff really had got hold of the wrong end of the stick, I am now of the firm belief that there is more than one marble loose within the somewhat thick cranium of the said individual.

My solicitor's advice was that a peaceful life could be had by allowing the plaintiff plenty of rope with which to hang himself, and that publicly displaying his stupidity could only aggravate him still further, so I will not mention the name of the scurvy knave.

However, since he was so opinionated when I spoke to him, I will share some of his highly unusual thoughts with you, dear reader, for you to judge his sanity - or lack of it.

For example, The Menace - for so I have christened him - believes that giving someone one of his firm's programs is a criminal act, since it deprives him of the opportunity to sell the program to that someone. If that someone then gives the program to another someone, that is a second sale lost to The Menace. Note that the process is one of GIVING - however, it could also be SOLD - but never copied. If you can find a mug with enough folding stuff you could sell him such a program for £1000, as long as no copying is involved.

The Menace however is threatening to claim lost royalties from me for the software which I gave freely to a second person, who in turn gave it to a third.

This all supposes that each individual was about to purchase the said program before some clot presented him with it as a gift. On that basis British Leyland should be suing the pants off everyone who ever sold a car secondhand, or worse, who had the temerity to actually give someone else a car. Interesting thought, no ?

The Menace has other interesting thoughts. For example, here is how a popular computing magazine ought to conduct reviews of his software:

First, six reviewers should be employed per program. The Menace did not make clear whether he was willing to tie up six copies of his work for the review, or whether the reviewers should take it in turns to receive the one program. This latter course gives something like a four month delay in the provision of a review.

Next, the reviews from the six reviewers should be Mix n' Matched to give a collection of the most praise-filled bits from each, and this should then be published. In the meantime all copies of the program sent out for review will have been either destroyed by the reviewers, returned to the magazine editor, or sent back to The Menace - maybe so he can sell them.

Most important of all, if all the reviewers spit on the program and can find nothing sycophantic to say about it, the magazine must NOT publish anything at all. Instead the editor should contact The Menace for advice on what to do.

What a wonderful way to do things, isn't it boys and girls ? Just think how things would be if everything was reviewed in this way. You would never read a nasty negative thing about any product ever again. Not even if it was, say, Thalidomide. Or DDT. Or Dioxin. Or little teddy bears with naughty little metal spikes behind the eyes.

Now you may have discerned just a hint of sarcasm there. If the programs put out by The Menace were so all-fired Fantastic then I might have been a little less put out. However, of the programs which I reviewed (and incidentally gave three and four star ratings) two at least had no debouncing of the keyboard scans so that you could slip through menus without warning. This point was highlighted in another review given in another User group's newsletter, so I am not, as The Menace insisted over the phone, living in Cloud Cuckoo Land. On top of this, when you consider that the latest quote for a TI Service is £65, it is downright irresponsible - nay, criminal - for a program to require children to hammer the keyboard as fast as they can in order to keep their score up. Ever seen a twelve year old's fingers blurring as he applies several tons of force to one key at a rapid rate of knots ? It's how they get their muscles, you know.

Now I am not particularly bothered if my steaming here eventually finds its way to The Menace, but I have a feeling that if 90 OTIUsers start ringing up every software house who advertise in a certain micro mag and ask: "Are you The Menace ?" it might just stir up a hornet's nest, so until such time as The Menace either wises up or undergoes treatment I would treat this item as a sort of bedtime horror story - a black comedy, if you will.

The only really sour note is that if The Menace continues to tell unsuspecting folks silly things about conspiracies then I will cease to be intrigued by his actions and become a little less jovial about the whole thing. At that point I will hand the matter over to the legal profession, a body not noted for its appreciation of sour notes and joviality.

~~~~~

ERROR MESSAGE

-----

I have had a letter from OTIUser JEREMY BYGOTT about V1.13's article from STICC (the Saskatoon TI Users Club) concerning CALL LOADS. It seems that references to the program listing published by me on page 55 are inaccurate. The changes listed on page 54 refer to lines 220, 260, etc., which are REM lines. Jeremy says that by adding 30 to all the line number references you end up with the right set of instructions. Sorry about that folks, I should have picked it up when I proof-read the material (I didn't type it or I might have spotted it then).

~~~~~

HELLO, HELLO, HELLO, HELLO, HELLO,...

-----

Saying Hello to the recently joined OTIUsers this issue is something of a mammoth task, as T. DAVIES, A. HARPER, PAUL DOYLE (contact for an Australian Group), RENE LAUWERS (Belgian contact), KIRSTY CLIFT (TI employee), F. PARRISH (owner of PARCO ELECTRICS), DAVE FORD, VIVIEN TOPPING, JOHN BAIRD, Dr J. D. BAINES, M. KINGHAM, ERNEST GREGORY, R. ELLIOTT, MIKE BRICK, SIDNEY NORTH, DAVID JOSEPHS, CHARLES SKRZYNSKI (French contact), NORMAN HANCOCK, ROGER MACEY, and by the time that this issue hits the streets, COLIN RANSON have all taken up at least the first year's worth of TI-LINES, and in some cases have not learned their lesson and have signed up for another twelve months. To all, welcome, and to your undoubted contributions over the next year hint hint, welcome also. I draw the line at begging on bended knees, though; at that height, the sun glinting off my bald patch could cause something akin to snow blindness, and then you wouldn't be able to see to write all your lovely articles, would you ?

~~~~~

OOPS!

-----

There were a number of silly mistakes in the last issue, due in part to the haste with which it was put together.

If you had trouble with the Forth screens on page 57, well, part of the problem was a ")" symbol missing from line 12 of screen 31. It goes between the PL @ 50 and the IF M3 ELSE. However, when I correctly entered both screens and tried to use them as suggested, they did not function correctly. I got as far as the instruction to INSERT MASTER - PRESS ANY KEY whereupon the drive loaded - what I don't know - and then locked up. Any Forth mechanics out there ?

There were some daft spelling errors where the speed of my fingers fuddled the TI-Writer (and my brain when I subsequently proof-read), but by and large they were not too horrendous. An FDR (page 27) is actually an FDB - not my doing, I hasten to add (pass the buck!).

Then there were the missing 2s on pages 24 and 25. Both CALL G() statements should have read: (2,1, ....) although the observant among you might have already worked that out.

~~~~~

## TWO INTO ONE WILL GO...

-----

GRAHAM HILTON has found a way of having two programs in memory at once. You will need the 32K RAM and Extended BASIC.

- 1) Select Extended BASIC (XB)
- 2) OLD the first program. It will go into the 32K
- 3) CALL PEEK(-31952,A,B,C,D).
- 4) PRINT A;B;C;D . Note down these Line Number Table pointers
- 5) Make 32K "invisible" with CALL LOAD from p45, issue V1.13
- 6) OLD the second program. It will go into VDP RAM
- 7) When finished running, NEW, and restore the 32K RAM as in V1.13
- 8) Reload Line Number table pointers - CALL LOAD(-31952,?,?,?,?)
- 9) Can now run the program in the 32K

You can FCTN= or BYE or even switch the console off; the program in the 32K will remain there.

~~~~~

## OF COMPUTERS AND CABBAGES

-----

While sweating over the photocopier which I use at my hospital I overheard one of those conversations that makes me want to jump up and down and tear somebody else's hair out. Three of our medical secretaries were discussing the influx of computers into local schools. One secretary ventured the assertion that it could only make cabbages out of the pupils: "pretty soon they'll be born without brains - they won't need 'em", quoth she. The others followed her line of thought and agreed that computers could only make cabbages out of everyone.



Now, I have just spent the last seven years trying to get to grips with the concepts of programming, and the last five years trying to understand tiny parts of the operation of just one computer. From where I stand, computers are set to make people out of cabbages, not the other way around. The moronic chattering of those ladies (who were all middle-aged) reflects the opinion of the vast majority of citizens in the UK - and as we have the most micros per capita in the world, think how that majority must be even greater in other countries. The computer is like no other tool ever invented. A spade is a spade is a spade, while a television is a television is a television. A computer however can be whatever you have the wit to make it - I shouldn't need to give you a list of the possibilities!

~~~~~

## UNPROTECTION RACKET

-----

You may have noticed last issue that two addresses were given for unprotecting Extended BASIC programs. There are actually several addresses, due to the fact that the protection address has not been fully decoded (information courtesy of RICHARD BLANDEN). Thus -31931, -32817<sup>¶</sup>, -32443, and -32699 are all valid addresses for protecting/unprotecting programs.

¶ Peter meant -32187 here

Incidentally, it has been suggested to me that to reveal this information is to act irresponsibly. However, the protection facility serves only one purpose - no matter what others might say - and that is to prevent Users with the basic system plus Extended BASIC from seeing the program listing. It can be for no other purpose - any claims as to its prevention of copying are pure naivety; you give anyone with two tape recorders (and half a brain) a tape and they will duplicate that tape without need for a computer.

My reason for revealing the information is very simple: I have spent the last five years hammering TI for keeping information from us. I try my utmost not to be a hypocrite, and if I was to hold back anything at all about the 99s I would be behaving exactly as TI have done since they launched the machine. I hate, loathe, and utterly detest any suggestion that there are aspects of the 99s which should not be known by the Users; one of the reasons that I produce this newsletter is so that as much information can be spread among as many people as possible. The more we all know, the better use we can make of our systems, and the greater the chance that good quality software will result. I know that some owners believe that the fewer the people that are "in the know", the better for all concerned. That has been the driving force behind every repressive political and religious regime the world has ever known and I for one am committed to destroying that idea, wherever it may rear its ugly head.

From a memory location to a declaration of war - now you can't say that TI-LINES isn't diverse!

~~~~~

## THE REAL BULLETIN BOARD

-----

ALAN DAVEY rang me to advise me of a problem which one or two Users have been experiencing when attempting to access his Bulletin Board - called 4ABC. When attempting to log on to his system you will need to press ENTER in order to complete the connection. In time I will publish a protocol for accessing Alan's board.

TIHOME SOFTWARE COLLECTION  
-----

All the programs in this Collection are now out with the reviewers and some have already been returned. I will set some time aside later this month to begin sifting through the material received, and the Catalogue looks set for a late June, early July, publication.

~~~~~

FILLING UP THE POOL  
-----

I am still extracting information from incoming newsletters and will publish more in the next few issues. Thereafter, provided things settle down, I will add some more to the original information in the pool as well as providing some more Beginner-oriented articles.

~~~~~

CRUD COPIER  
-----

Some of you will have received some rather cruddy copies of TI-LINES due to the condition of the hospital photocopier this time. The state of this machine depends very much on the technical skills of the people using it, and as the majority are non-technical personnel it is hardly surprising that it needs maintaining regularly. Equally hardly surprising, it is never maintained regularly!

~~~~~

CURVING YOUR TEXT CAN BE AN UPHILL TASK.....

IT CAN GIVE YOU THAT SINKING FEELING.....

-----  
R E P O R T   F R O M   B R I G H T O N  
-----

Peter Brooks 1985

Sunday 28th April  
-----

3.30 am

Yawn. Must try and yawn get to yawn bed. Got to be up at yawn rub eyes fall over chair 'cos carried on walking while rubbing filthy habit ought to be banned zzzzzzz...

5.30 am

Wha?! Alarm going off causes sympathetic resonance of skull. Vision is blurred, tongue crawling around on floor on other side of room. Brain still in neutral. Tongue now in motorcycle welly. Fast becoming a sleep junkie. Two hours a night enough for anyone. Trouble is, need thirty five day week. Tongue refuses to return to normal abode until oral cavity undergoes sandblasting. Swill cupful of Domestos around teeth. Eyes snap back into focus. Tongue comes home, all is forgiven. Leap out of bed in Action Replay Slow Motion, narrowly missing edge of table. Miss edge, hit table. Large bump appears on head. Paint it blue, pretend am a policeman. Yawn.

Malcolm and Carole Hedley due to pick self up around 6.30. Time to scrape stubble off face. Glue it onto bald patch. Usual routine - smile into mirror, feel sick, put teeth in, try again, no good, teeth in back to front, grinning at own tonsils.

Gulp down bucketful of All Bran. All part of Anti-Pot-Belly Plan B. High fibre - astronomically high fibre - diet: All Bran and Baked Beans.

Car draws up outside. Malcolm and Carole with Dog.

Off we go.

First stop, Richard Sierakowski in Marlborough. Richard is real policeman. Forget about painting bump blue, get had up for impersonating. Off again. Wonder if I should have worn cycle clips. Trouble is, trousers probably inflate, cause havoc with self floating upside down in car, put Malcolm off driving, worse, someone light cigarette, could be first 99ers in orbit.

10.15 am

Finally find Dome and parking space. Didn't manage to drive past Nuddie Bathing Beach. Boo. With help from Malcolm, Richard, Carole, get boxes into hall. Set up bits on trestle table. Forgot to bring stickers with OXON TI USERS. Visitors probably think we are secret society.

99ers Anonymous ?

10.30 am

And They're Off! Famous Brooko Lower Jaw starts racheting. Try to be multi-tasking, multi-user response unit. Dead loss, only one mouth and half a brain. Meet OTIUsers, promptly forget what they look like five minutes later. Manage to insult Graham Hilton by treating him like stranger ten minutes after meeting him. Oops.

Pace hots up, brain overloading. Trying to sell copies of August HCM/99er and Tidings; few takers. More questions than answers. Brain down to last neuron. Stomach indicates suspicion that throat cut. Carole points out that cup of tea been waiting half hour for me to wet whistle. Takes me an hour to drink it. Oral cavity so hot from gabbling, liquid vapourises on contact with gums. Gabble, gabble; gabble, gabble. Barely have chance to realise who other exhibitors are. Wonder what time Uncle Clive and Auntie Audrey want self to answer questions.

1.00 pm

Encountered Kirsty Clift, remember her name from days of Robin Frowd. Arrange audience with her to ask one billion questions. Only get chance to ask one, promptly thrown into confusion by discussion of crushers, rubbish tips, and lack of TI consoles, all facts at odds with everything else self has ever been told. Brain now running in reverse, eyes turning in every s often to see if each other still there. Aaagh! Sidetracked into viewing new Parco delivery van - 140 mph Warp Factor 7 Mr Spock ?

5.00 pm

Met so many people, faces beginning to be blurred. Getting names confused starting seven sentences with five visitors and finishing none of them. some folks exhibiting masochistic tendencies, joining OTIU without first having full medical. Uncle Clive tells self show over, where was self at Question Time, Steam Radio been and gone.

6.00 pm

Back on the road, multiple debriefing in car.

7.30 pm

Hit the Hogsback (self spends five minutes looking for hairy bits on road), descend on Happy Eater (Easter ?) or was it Little Chef, never mind, FOOD, fellow travellers manoeuvre self into corner by baby's high chair, waitress gives self funny look while taking order, other diners also giving self funny look, should have rolled up trouserlegs, claim Balding Juvenile, get 10% knocked off bill. Stomach convinced Throat been Absent Without Leave

Back in car. Brain gradually coming back to life, trying to remember who said what, who promised what.

10.00 pm

Get to Richard's place, drink best cup of coffee had all year, discuss pros and cons of guard dogs. Malcolm points out terrier can kill dobermann, few eyebrows raised, yes, says Malcolm, terrier get stuck in dobermann's throat kill it stone dead.

Monday 29th April  
-----

12.15 am

Malcolm and Carole drop self off at home, Dog heaves audible sigh of relief, self very grateful to Malcolm for weightlifting and driving, he and Carole still got 40 miles to go, so say cheerio and see them off.

Get indoors, remember that self still has to do some work yet, manage to get sidetracked by snooker final, finally get work done and into bed by 4.30 am.

Dozing off after fantastic day, must do this again soon, Clive says maybe Birmingham in October, great, realise that still didn't get chance to visit Nuddie Bathing Beach

Wonder if Birmingham got Nuddie Beach zzzzzzz...

~~~~~

-----  
S T I C C  
-----

From the newsletter of the SASKATOON TI COMPUTER CLUB, March 1985

CHANGING COLOURS IN THE EDITOR/ASSEMBLER

By TERRY ATKINSON with thanks to TOM FREEMAN who inspired the article

Some time ago, Tom Freeman provided me with a method whereby the colour default on the Editor/Assembler (EA) EDIT1 program could be changed by using the TI Forth Disk Fixer. Not being 'handy' with this (to me) elusive language, I decided to attempt to accomplish the same end by using a standard disk fixer. It worked!

Why change the colours? Well, some of us are using green or orange monitors, or monochrome TVs, and white letters on a blue background do not show up that well. Others may just want a change.

The theory provided by Tom is that every bit of memory space is required by the Editor program and data generated by the User. Hence he used an area of the program utilised by TI to display their copyright. In this space, he wrote a little assembly language utility to produce the colour change, and is in essence a VWTR to change VDP register 7.

Here's how to do it.

First, transfer the EDIT1 file over to a freshly initialised disk. Then boot up your favourite disk fixer. Once the fixer is booted up, call in Sector 22. This sector is headed by the words: >0000 1700 2000 55AA. Ensure you have the right sector. Now locate the word at >0024. It should read >2843. Using the (alter) command, change this, and the next consecutive 15 words to read:

```
>2020 C800 C01E 0200 8717 06C0 D800 8C02
>06C0 D800 8C02 C020 201E C020 FFDB 045B
```

Write sector 22 back to the disk and call in sector 24. This sector is headed by the words: >0014 0420 2DB2 1000. Again, check to ensure that you have the proper sector. Now locate the word at >00B8. It should read >C020. Change this and the next word to read:

```
>06A0 2020
```

Write sector 24 back to disk and that's it. You're done.

Check out your work by loading the EA using the EDIT utility. You should now have a cyan screen with black lettering. If you don't like that colour combination, there is nothing stopping you from selecting your own colours. Merely load sector 22 back in and change the word at >002C which now contains >8717, to read >87xy where 'xy' is equal to your own preferred combination. See the EA manual, page 330. Write the sector back out.

Now, one for you. How about different colours in TI-Writer ? Can it be done ? If so, let us all know. I don't particularly like the combinations in that utility either!

For the curious, I have the source code for the above routine:

```
AORG >22B2          TWO WORDS TO BE REPLACED, TO BRANCH OUT OF PROGRAM
BL @ >2020          THE WORDS REPLACED ARE >C020, >FFD8
AORG >201E
MOV 0,@>201E       TO SAVE REGISTER 0 WHICH NEEDS TO BE USED
LI 0,>8717         FIRST NYBBLE MUST BE 8, SECOND IS THE REGISTER TO BE
                  CHANGED, THIRD IS FOREGROUND COLOUR, AND FOURTH IS THE
                  BACKGROUND COLOUR.
SWPB 0            NEED TO WRITE LSB FIRST
MOVB 0,@>8C02     VDP WRITE ADDRESS (AUTO INCREMENTS)
SWPB 0
MOVB 0,@>8C02     THIS COMPLETES CHANGING VDP REGISTER 7
MOV @>201E, 0     PRESERVES THE PREVIOUS WS REGISTER 0
MOV @>FFD8,0     THESE ARE THE BYTES REPLACED BY THE BL INSTRUCTION
RT              RETURN TO ORIGINAL PROGRAM
```

~~~~~

Editor's Note: A future article will give details of SECTOR ACCESS, using information and routines supplied by RICHARD BLANDEN.

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

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

RICHARD SIERAKOWSKI has a stack of DOUBLE-SIDED, DOUBLE DENSITY disks for sale at £16 for a box of 10. I've bought 60 so far and they have served me OK. He also has a book for sale: GET MORE FROM THE TI-99/4A by GARRY MARSHALL. Offers around £5 please to him at:

Rusholme  
Elcot Lane  
MARLBOROUGH  
Wilts  
SN8 2BA

Tel: 0672 54975

DAVE HEWITT has designed a PARALLEL CENTRONICS PRINTER INTERFACE which plugs into the right hand side of the console. Contact him for more details:

311 London Road  
Headington  
OXFORD  
OX3 9EJ

Tel: Oxford 67165

I understand that the cost will be about £45, which can't be bad for a parallel interface.

KEN LARKIN has a few items for sale:

ALPHACOM 42 WITH TI INTERFACE (plugs in right hand port)	£85
EDITOR/ASSEMBLER MANUAL	£ 7.50
BLACKJACK & POKER MODULE	£ 5.00

Contact him at:

268 Hathersage Road  
Lambwath Road  
HULL  
HU8 OEX

Tel: 0482 706084

GORDON PITT of the WEST MIDLANDS TI USERS has a couple of services for TI owners (one of which I have experienced and can recommend it). Gordon can supply POWER SUPPLIES to suit most requirements - for example, "clean" psu's for disk drives with cases. He will also supply MONITOR CABLES and other interfaces. You can make the initial contact through me.



Power Supplies:

Single £25  
Double £45

DISK DRIVE CASES - APPLY FOR PRICE

(I have a double-sided, double density MPI 40 track drive fitted with a power supply and case through Gordon, and it is a very professional job.)

MIKE ADKINS has some items for sale:

Super Sketch tablet £40 (still in box)

Alpiner module £8.50

Household Budget Management £7.00

All items are "or nearest offer", and you can contact Mike at:

Stout Mill Cottage  
Yarcombe  
HONITON  
Devon  
EX14 9LX

WANTED WANTED WANTED WANTED WANTED WANTED WANTED WANTED WANTED WANTED WANTED

Dr J. D. BAINES is looking for a DIAGNOSTIC module. If anyone has one they'd like to sell, or knows where one might be bought, please let me know so that I can pass the information on.

-----  
C O N T A C T S  
-----

MIKE BRICK 13, Gosselin Road, Bengoe, Hertford, Herts SG14 3LG  
NORMAN B. HANCOCK 7 The Lea, Trentham, Stoke-on-Trent, Staffs ST4 8DY  
Tel: 0782 641375 7pm - 10pm  
RAY ELLIOT 6 Falling Lane, Yiewsley, Middx., UB7 8AB  
Tel: 0895 440606

# INTERFACING THE TI99/4A

## PART 1 INPUT

D. G. HEWITT

One of the main limitations of the TI99/4A computer is the lack of any simple input/output connections for interfacing to the outside world. This article aims to demonstrate how the joystick port can be used from a Basic program to read the status of up to 8 (or even 10) input lines yet needing no modification to the computer merely a simple interface board.

Before we can start to use the joystick port as an input we must first understand how the joysticks are connected. Figure 1 shows the joystick wiring (this may be of help to people trying to interface other joysticks). Each joystick has 5 switch contacts which are common to both joysticks and each has a separate switch return line. When a particular joystick is selected its switch return line goes low (0V) for 150µs. If any switch is closed then the corresponding line is pulled low. The computer determines which joystick is read by the value specified in the CALL JOYST statement. As only one joystick can be read at a time there is no possibility of the other joystick influencing the readings due to the diodes in each joystick.

The next problem is to design an interface to enable TTL logic levels to be read. The first stage is 8 opto isolators to replace the switches in each joystick. I used 2 quad opto isolator packages. This works okay but as usual with the TI99 there are problems. Firstly the current required to drive the opto isolators is in the order of 20mA which is more than most TTL ic's can supply. This problem can be cured by adding buffers to the input lines. The circuit in Figure 2 is the interface so far and for most purposes this will prove satisfactory. This circuit has been built and tested.

The second problem is that the TI99 can only read one joystick at a time. This means there is a time delay of approximately 50ms between reading each joystick. For most applications this will not matter but if you are trying to read dynamic data then it might. One possible cure (which I have not yet tried) is to add a data latch which would latch the data when the first joystick is read. A monostable would also be needed to hold the latched condition until after the second joystick had been read. A circuit that may do this is given in Figure 3.

The third problem is the way that the TI99 software reads the joysticks. In normal use the joystick can be say up or down returning a value of 4 and -4 respectively. In normal use it is impossible for the joystick to be both up and down at the same time therefore the internal software does not cater for this condition. This means that although we have 8 input lines it is more correct to say we have 4 input pairs of which only one can be active at any one time. If both are active together then the computer behaves as if both were inactive. This could be used to our advantage as this could be configured as 4 2input Exclusive OR functions that can detect either input but not both.

### Using the interface

The following two listings show example subroutines to read the interface. The first reads in the pure binary value from each input and returns the values (0 or 1) in the array DT(N) while the second listing reads the inputs as Exclusive Or pairs and returns the result in the array DT(N). Note that a 1 represents 0V and a 0 represents 5v input levels i.e the inputs are active low, negative logic. Note also the Alpha Lock must be up. Both listings are in TI Basic but those with Extended Basic can utilise some of the enhancements such as better IF THEN statements etc.

## LISTING\_1

```
10 GOSUB 1000
20 FOR N=1 TO 10
30 PRINT DT(N);
40 NEXT N
50 END
1000 CALL JOYST(2,X1,Y1)
1005 CALL JOYST(1,X2,Y2)
1010 IF Y1<>4 THEN 1030
1020 DT(1)=1
1030 IF Y1<>-4 THEN 1050
1040 DT(2)=1
1050 IF X1<>4 THEN 1070
1060 DT(3)=1
1070 IF X1<>-4 THEN 1090
1080 DT(4)=1
1090 IF Y2<>4 THEN 1110
1100 DT(5)=1
1110 IF Y2<>-4 THEN 1130
1120 DT(6)=1
1130 IF X2<>4 THEN 1150
1140 DT(7)=1
1150 IF X2<>-4 THEN 1170
1160 DT(8)=1
1170 RETURN
```

## LISTING\_2

```
10 GOSUB 1000
20 FOR N=1 TO 4
30 PRINT DT(N);
40 NEXT N
50 END
1000 CALL JOYST(2,X1,Y1)
1010 CALL JOYST(1,X2,Y2)
1020 IF X1=0 THEN 1040
1030 DT(1)=1
1040 IF Y1=0 THEN 1060
1050 DT(2)=1
1060 IF X2=0 THEN 1080
1070 DT(3)=1
1080 IF Y2=0 THEN 1100
1090 DT(4)=1
1100 RETURN
```

## Further\_development

It is possible to extend this idea of interfacing to include the fire buttons on the joysticks but this has the problem that they also respond to the keys Q and Y so this was not used. It could however yield 10 input lines if this were exploited. I hope this may be of use to some people and I will be pleased to answer any queries anybody may have. Part two which I intend to write in the future will detail how to use a centronics type printer port as an 8 bit output port.

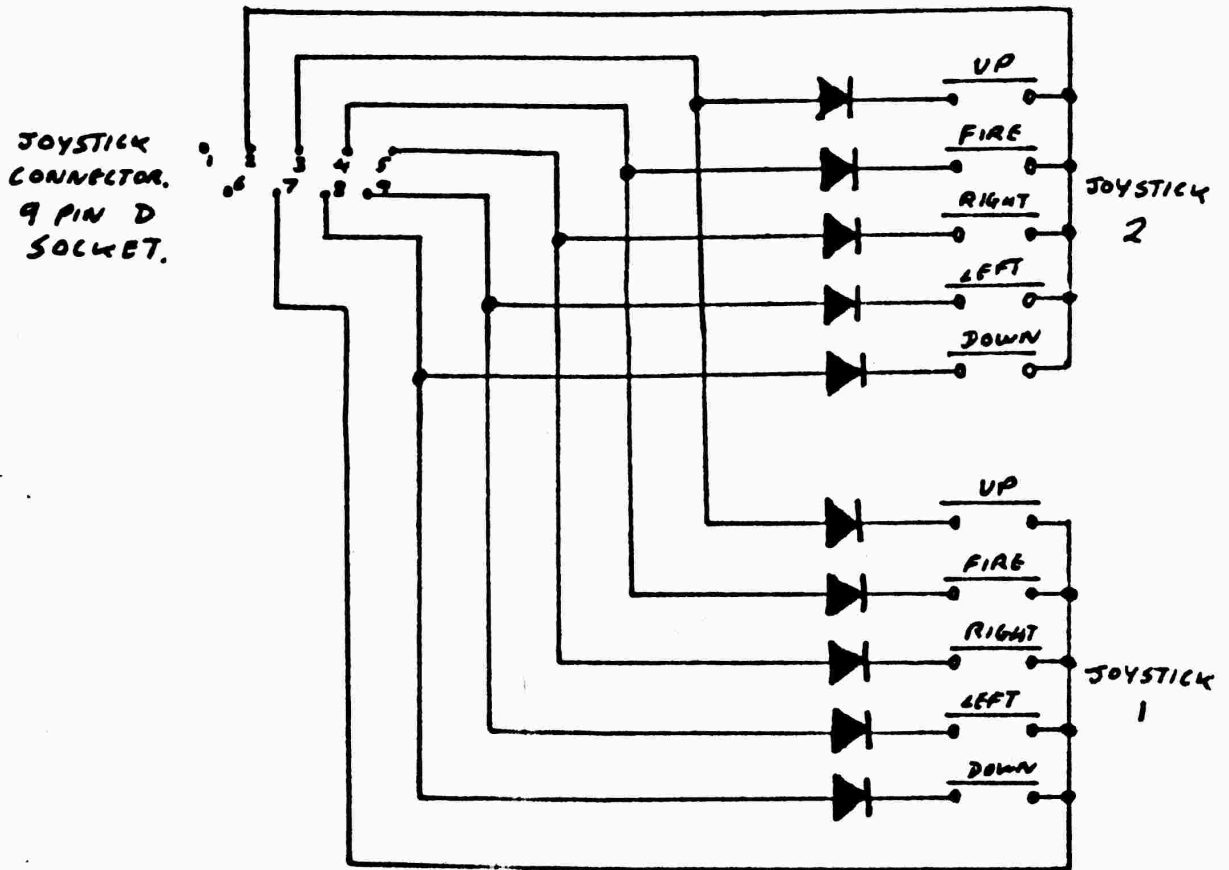


FIG 1 TI99 JOYSTICK WIRING.

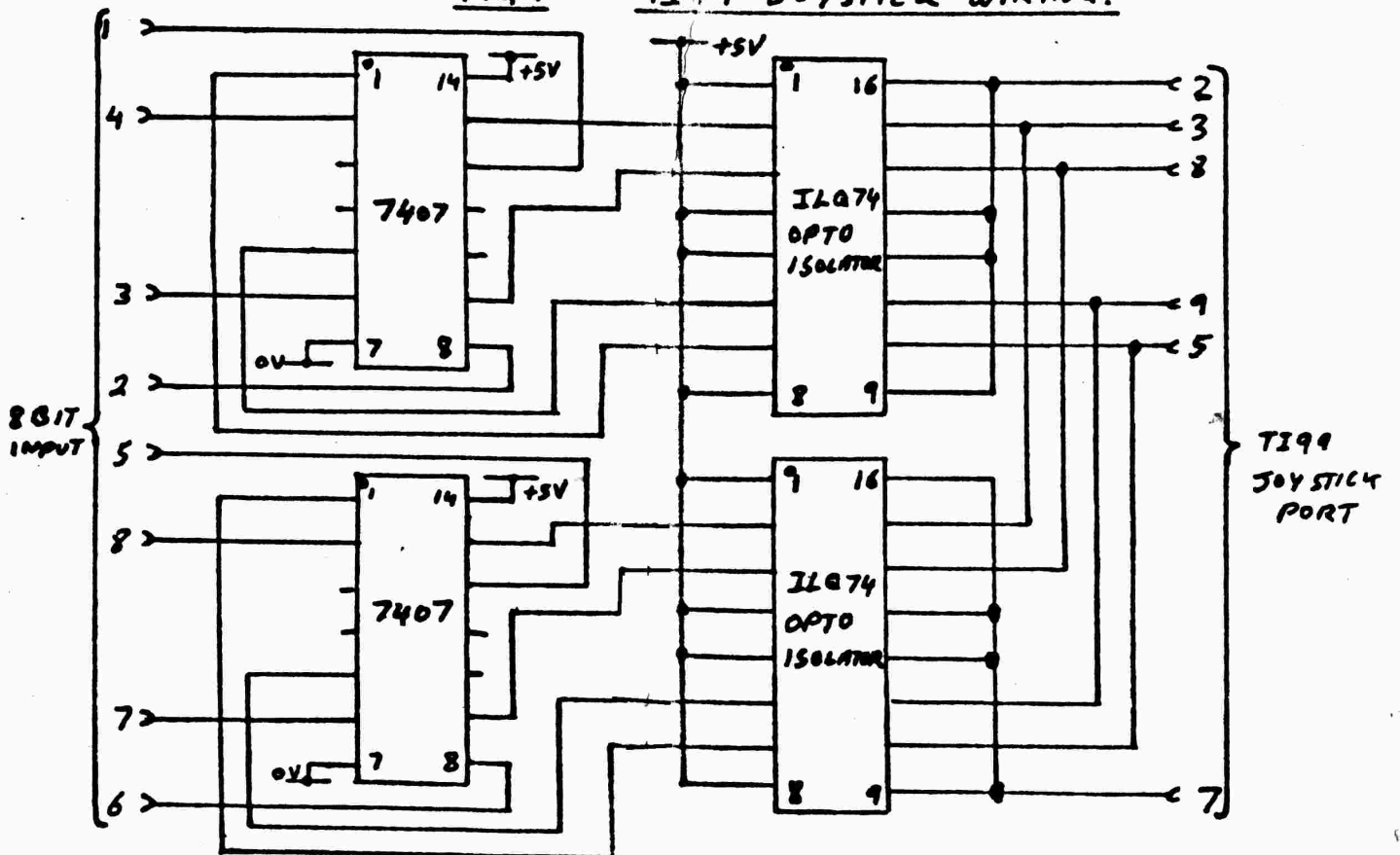
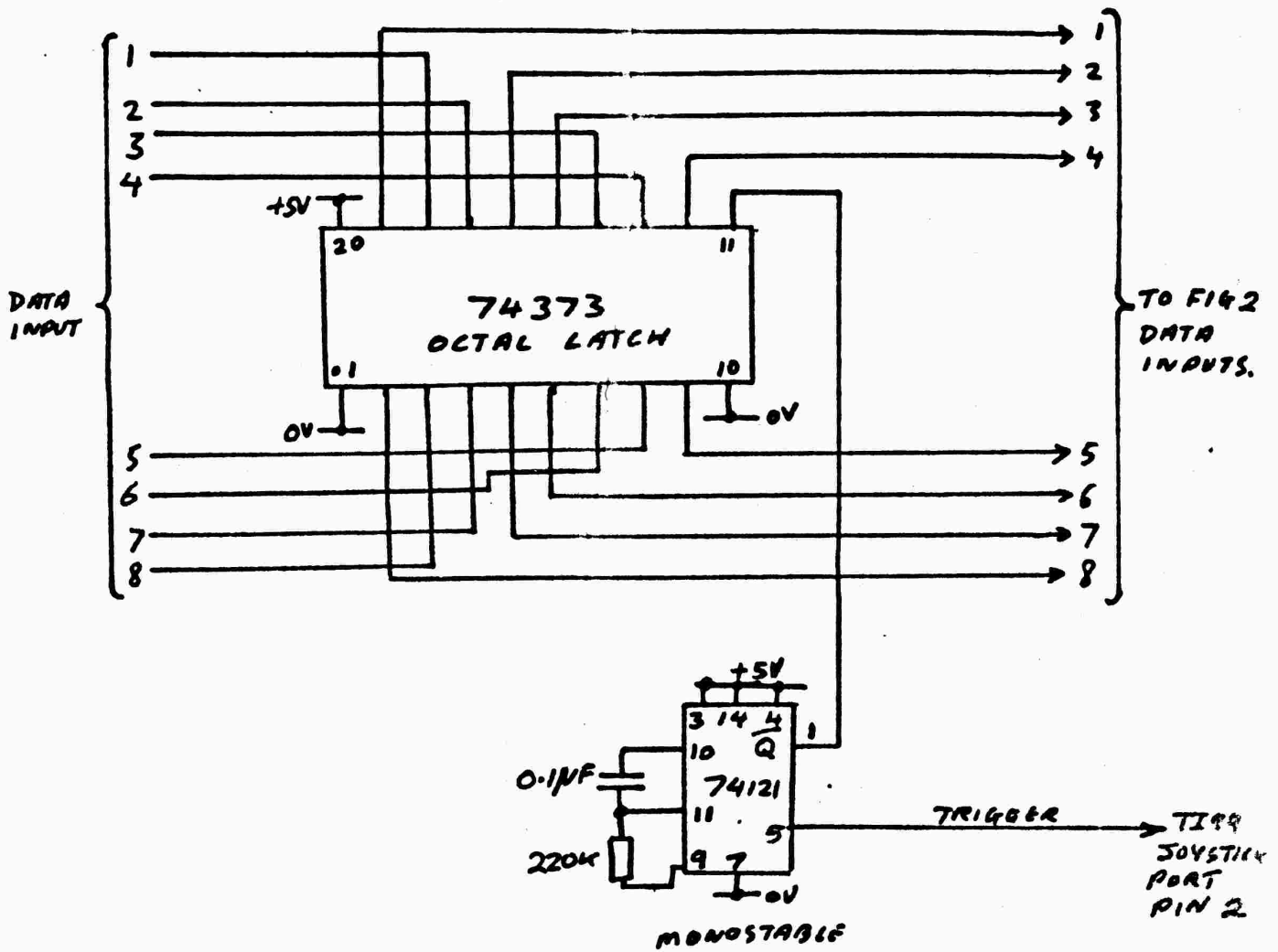


FIG 2. INTERFACE CIRCUIT



F143 SUGGESTED MODIFICATION TO ADD DATA LATCH

# DAVID BROWN

Hello again, this month has contributions from both Richard Owen and from myself, first of all is a small quiz from Richard and then a small program written by me and then part one of a series by Richard about assembly language.

Here goes with the small quiz; it is a list of 22 words which have been scrambled what you have to do is unscramble them. The first one is done for you.

- |                   |                     |
|-------------------|---------------------|
| 1) CARPSE-PARSEC. | 12) YCKSTOIJ        |
| 2) MAINLEP        | 13) TLLOEH          |
| 3) VERDIKDIS      | 14) CCOERS          |
| 4) PETRNIR        | 15) SSCEH           |
| 5) MUTCORPE       | 16) ZIMAANG         |
| 6) SSSTTAIC       | 17) MNNUCHA         |
| 7) IITNESL        | 18) MMEOD           |
| 8) CBSIA          | 19) NIIREDAVT       |
| 9) TRRRAPENECOD   | 20) YCPHHSNTSSZEEER |
| 10) SSEEMALRB     | 21) AEEDRUTVN       |
| 11) SSTTECA       | 22) AEEIMNRRSSSTTUX |

All the words above are TI-related. The answers will be in next month's issue of TI-LINES.

.....  
Now here is a short program that I have made up. It runs in Extended Basic and it is very useful if you are making up alot of characters. You make up your character on a piece of paper as usual and then you enter it into this program and you can immediatly see what your character looks like and decide if it needs altering. When you run the program, you will first be asked MAGNIFICATION? Type in the number 1,2 ,3 or 4 depending on which magnification you want. Then you will be asked COLOUR?-type in the colour number that you want. Then when the bleep sounds, enter the hexadecimal code for the character you want and it will be displayed instantly. Press any key to repeat the whole thing.

```
100 CALL CLEAR:: DISPLAY AT(24,1):"MAGNIFICATION?":ACCEPT AT(24,16):Z
110 DISPLAY AT(24,1):"COLOUR?":ACCEPT AT(24,9):C
120 INPUT A$:CALL CHAR(100,A$)::CALL MAGNIFY(Z)::CALL SPRITE(#1,100,C,96,128)
130 CALL KEY(0,K,S)::IF S=0 THEN 130 ELSE 100
```

This is only a very short routine but I find it very useful.

.....  
Here is another bit from Richard now. Gary Harding has kindly given Richard permission to condense his booklets on assembly language especially for TI-LINES. Originally there were four booklets with around fifteen pages in each one but Richard has squashed all this up.

## Part 1. The Basics.

### How?

On the TI-99/4A, the step to assembly language programming may be made in two ways:

- 1) The mini-memory module (£75) or
- 2) The Editor/Assembler (£45), this is useless without at least £600 of extra

hardware (disks, 32K ram etc).Of the two, the ED/ASS offers far superior features.

Naturally, the sheer cost of putting together such a system will mean that most of us will want to program in 9900 will utilise the MINIMEM's LINE-BY-LINE assembler. This is a pretty sophisticated tool in itself: It certainly beats machine code programming(working out actual HEXADECIMAL numbers which represent assembly language instructions, and poking them into memory.) The one thing that is missing from the MINIMEM's package is a good 9900 assembly language manual... which is where this series-hopefully-comes in.

Why?

I am not suggesting that assembly language is in any way (or rather Gary Harding is not saying) the logical next step for 99 owners to take. On the 99/4A, TI have provided a very powerful BASIC interpreter, with excellent file-handling capabilities...and easy access to colour graphics and sound, and for most tasks BASIC is good enough and is the best language to use. However, there will be some occasions (you may have already encountered some) when the speed of the TI-BASIC is just not up to the job (In games and screen dumps etc) partly because of the way that programs and data are stored in VDP RAM and partly because it is an interpreted language.

Imagine for a moment how long it would take you to go through a Russian newspaper if all you had was a Russian-English dictionary. Well, that's the presice position in which your TMS 9900 microprocessor finds itself when you run a basic program. It reads it line at a time...looks it up in it's "internal dictionary"...and then executes the assembly language "translation" of your BASIC statement. The only answer to this is to buy an English newspaper... In other words, write the program in assembly language directly.

In general, using assembly language to speed up operations is a good idea and is easy too (so I'm told).

.....  
I suppose that I'd better call it a day now or I'll be up all night writing this, more next month-sorry, lot's more next month!

*David Brown*

## CIN-DAY USER GROUP

TI Disks - summary of the information on the first 33 sectors:

Hex	Dec	Meaning (of the 256 bytes in the sector)	SECTOR 0
-----	-----	-----	
00-09	0- 9	The disk name you assigned	
0A-0B	10- 11	Number of sectors initialized (ex h0168 = 360)	
0C	12	Number of sectors per track (ex h09 = 9)	
0D-0F	13- 15	TI identifier - "DSK" or h44534B	
10	16	Copy protection (ex h20 = none, h50 = protected)	
11	17	Number of tracks (ex h28 = 40)	
12	18	Number of sides (ex h01 = single, h02 = double)	
13	19	Disk density (ex h01 = single, h02 = double)	
14-37	20- 55	not used	
38-64	56-100	) This is a bit map of all the sectors on the disk Use depends on if the disk is SS, DS, SD, or DD 1. Take each byte (45 bytes for 360 sectors) 2. Convert to bits (8 bits per byte) 3. Reverse the order of the 8 bits 4. If the bit is "0" then the corresponding sector (0 to 359) is free. If the bit is "1" then the sector is used.	
66-92	102-146		
94-C0	148-192		
C2-EE	194-238		
65,93	101,147	) not used	
C1,EF	193,239		
FO-FF	240-255		

Hex	Dec	Meaning	SECTORS 2-33
-----	-----	-----	
00-09	0- 9	The file name that you used	
0A-0B	10- 11	not used	
0C	12	File Type    bit 0 - 0=fixed            1=variable length bit 4 - 0=none            1=write protected bit 6 - 0=display        1=internal format bit 7 - 0=data            1=program file	
0D	13	Number of records per sector (n/a for program)	
0E-0F	14- 15	Number of sectors per file	
10	16	End of file offset in last sector (n/a for fixed)	
11	17	Record size (n/a for program)	
12-13	18- 19	Number of records per file (n/a for program) note - the bytes are reversed (ex h0102 = h0201)	
14-1B	20- 25	not used	
1C	26	Sector where file is located	) repeats as needed ) to use any sector
1D-1E	27- 28	Number of sectors following note - the bytes are flipped (ex h12 = h2001)	



ENHANCED BASIC : TI Basic when the PRK or Stats modules are plugged in. CALL L:

#### GENERAL DESCRIPTION OF THE LOAD SUBPROGRAM

The LOAD subprogram is a subroutine resident in the Personal Record Keeping and Statistics command modules. When the command module is plugged in, this routine can be called from a BASIC program. It is used to load a data file from an external device into the data area reserved by the PREP subprogram. The CALL statement is used to execute the routine and takes the form:

CALL L(V<sub>f</sub>,V)

where

V<sub>f</sub> = file-name (string expression)

V = return variable (numeric variable)

V<sub>f</sub> is a string constant, variable, or expression which specifies the file to be loaded. V is a numeric variable in which the LOAD routine returns a code to indicate whether the load was or was not successful. V = 0 indicates that an error occurred. Any other value indicates that the load was successful. The following conditions will cause a LOAD failure.

1. The specified file and/or device does not exist.
2. No data area has been allocated.
3. The defined data area is too small for the data to be loaded.
4. General I/O errors.

#### CAUTION

Personal Record Keeping, Statistics, and the LOAD and SAVE routines use "program" type files. This means that 256 byte records are used. Thus the "effective" size of a file is the true size of the file rounded up to the nearest multiple of 256. For this reason it is possible to save a file which cannot be re-loaded into the same size data area.

- 1 -

After Texas Instruments

## CALL S

### GENERAL DESCRIPTION OF THE SAVE SUBPROGRAM

The SAVE subprogram is a subroutine resident in the Personal Record Keeping and Statistics command modules. When the command module is plugged in, this routine can be called from a BASIC program. It is used to SAVE a data file from the data area reserved by the PREP subprogram to an external device. The CALL statement is used to execute the routine and takes the form:

CALL S(V<sub>f</sub>,V)

where

V<sub>f</sub> = file-name (string expression)  
V = return variable (numeric variable)

V<sub>f</sub> is a string constant, variable, or expression which specifies the file in which to save the data. V is a numeric variable in which the SAVE routine returns a code to indicate whether the save was or was not successful. V = 0 indicates that an error occurred. Any other value indicates that the save was successful. The following conditions will cause a SAVE failure.

1. The specified device does not exist.
2. No data area has been allocated.
3. General I/O errors.

#### CAUTION

Personal Record Keeping, Statistics, and the LOAD and SAVE routines use "program" type files. This means that 256 byte records are used. Thus the "effective" size of a file is the true size of the file rounded up to the nearest multiple of 256. For this reason it is possible to save a file which cannot be re-loaded into the same size data area.

- 1 -

After Texas Instruments

## LOAD INTERRUPT SWITCH

October 1984

by Jon Bannister

Have you ever wished you could restart a program after it "crashed", return to a menu after running a machine language game or peek into Command modules and copy them? For less than \$6.00 in parts (including tax) and less than an hours work, you can add that capability to your TI-99/4A. For those not yet familiar with the LOAD interrupt feature of the TMS9900 microprocessor chip, allow me to attempt an explanation. When the LOAD pin (pin 4 on the 9900) is brought low (set to ground) the processor performs an equivalent to a BLWP @hFFFC. The processor reads memory location hFFFC, which becomes the new WP (work space pointer) and hFFFE which becomes the new PC (program counter). The program pointed to by these addresses is then executed. The only time this does not happen is if the RESET pin (pin 6) is held low (which, on the TI-99 series can only happen if you turn ON your computer or INSERT a Command Module). LOAD is a non-maskable interrupt, which means that there is no way of disabling it (theoretically). As it stands, the TI-99/4A does not support this feature. This month we shall show how you can add it.

The parts you'll need for this conversion are as follows: two 6-7 inch pieces of insulated wire, a .1uF bypass capacitor (Radio Shack part number 272-135 which costs \$0.99 for two), a 2.2kOhm resistor (Radio Shack part number 271-8027 or 271-1325 which costs \$0.59 for five) and a momentary contact, normally open push-button switch (Radio Shack part number 275-1547 which costs \$3.79 for five). You'll also need a small Phillips screwdriver, a soldering iron and some rosin-core solder.

The procedure is as follows:

1. Read ALL of these instructions and understand them before continuing.
2. Take the last  $\frac{1}{4}$ " of insulation off of each end of each piece of wire.
3. Tin the bared ends of the wire and the leads on the resistor with solder.
4. Solder the capacitor across the switch contacts using as little of the capacitor's leads as possible.
5. Solder one wire onto one of the switch contacts.
6. Solder one end of the resistor to the other switch contact.
7. Disconnect your Speech Synthesizer (from now on referred to as S.S.) from your computer system.
8. Turn the S.S. over exposing the bottom. Remove the two screws holding the case together.
9. Lift off the bottom of the S.S. (which is now on top). Carefully lift out the printed circuit board (PCB) with its metal enclosure.
10. Remove and store the two screws, washers and nuts.

11. Lift off the metal cover that has a hole on its top. Hold the remaining metal cover in one hand and hold the ends of the black connector in the other. Lift the connector up and away from the copper fitting.
12. When handling the PCB hold it only by the edges and avoid contact with any of the components as this might damage them.
13. Turn the PCB over to the side where no components can be seen. Orient the PCB with the black connector to your right.
14. Solder the wire connected to the resistor to the seventh lead from the bottom of the black connector, and the other wire to one of the 11th, 12th, 13th or 14th leads from the bottom.
15. Route the wires to the top of the PCB. Take the metal cover with the copper fitting and fit it over the left side of the PCB. Make sure the metal tab at the bottom of the cover is over the end of the PCB and that the wires are under the slot at the top of the cover.
16. Flip the board over exposing the component side. Put the remaining cover on the board. Replace the screws, washers and nuts (the washers go under the nuts, not the screw). Tighten the screws. (Remember you're not changing a tyre - take it easy).
17. (Before you drill any holes, place the PCB back into the black box and check for clearance between the switch and PCB). Measure  $\frac{3}{8}$  of an inch over from the side of the S.S. black box at the end with the silver top. Measure down from  $\frac{3}{8}$  to 1 inch. At the cross of these two markers drill a  $\frac{5}{16}$  inch hole.
18. Fit the switch through the hole. Install the washer and nut and tighten.
19. Place PCB into black box (it only goes one way). Put the bottom back on, replace and tighten the screws. Connect the S.S. back to your computer and you're done!

Some hints on how to use this switch: When using the Editor/Assembler Debug program, the Debugger automatically sets hFFFC and hFFFE. If it ever crashes just push the button. From Extended Basic, Minimemory or E/A Basic you can do a reset with a CALL LOAD (-4,131,224,0,36). To peek into cartridges, load Debug and run it. Remove the cartridge and replace it with the one you wish to play with and push the button - voila!

In my next article I'll show how you can explore into cartridges by removing the RESET on the cartridge slot.

Disclaimer: This modification will void the manufacturer's warranty. Neither I, nor the 9T9 Toronto Users Group can be held responsible for errors, omissions or any damage occurring during or after this modification.

-----  
E N H A N C E D    B A S I C  
-----

An enhancement of TI BASIC available through STATISTICS and PRK modules

Peter Brooks    May 1985

References:    TI Document ARCHIV.PRK.DOC.SUBRLS1 courtesy of TI  
              Articles by, and personal communication with, PAUL W. KARIS

CALLs L() and S()

These two subprograms are potentially the most useful (and certainly the most powerful) of any in Enhanced BASIC. We have already used them in previous issues (before seeing their description, which was rather cart-before-the-horse) and you may have noticed that they appear to be the simplest to use.

The general form is:

```
CALL L(V$,V)
CALL S(V$,V)
```

where:

V\$ = string, string expression, or string variable  
V = numeric variable (return variable)

For example:

```
CALL L("CS1",A)
CALL S("DSK1."&D$,C)
```

The string can be any valid peripheral name - eg., cassette or disk and even the RS232 interface.

CALL S

We'll deal with CALL S() first, mainly because you cannot load a file which hasn't first been saved. CALL S(), like CALL L(), uses PROGRAM FORMAT files, which are so-called MEMORY IMAGE dumps. They're called PROGRAM FORMAT for the reason that they are composed of 256 byte records (the same format as BASIC programs), which are transferred in one fell swoop - as OLD and SAVE do with programs. This method uses just one header tone followed by a continuous memory dump.

MEMORY IMAGE simply means that the computer transfers the contents of its memory working with a continuous block from a START address to a FINISH address. This may sound an obvious thing to do, but remember that there are other forms of memory dump - MERGE format files in EXTENDED BASIC, for example, in which the contents of memory are transferred according to a table of line number pointer. If anyone asks for it, I will try to explain (thanks to STEPHEN SHAW and RICHARD BLANDEN) the general principles behind MERGE.

So what dictates where the START and FINISH addresses lie? The HEADER set up by CALL H(). It contains the necessary information (see V1.10) which CALL S() can use to decide exactly how much memory to dump.

If you simply did:

```
CALL P(10000)
NEW
CALL S("CS1",A)
```

the computer would respond with a NUMBER TOO BIG error. We will examine the various error messages at another time, but in this case the message means that the computer could not transfer the contents of memory because it had no header from which to work. There are rather weird circumstances when you CAN get CALL S() to dump memory when a header has not been created, but they involve using a BASIC program which is loaded using CALL L() and then dumped with CALL S(). However, in that case you could find yourself waiting for hours as the computer dumped up to 64K to tape/disk/RS232 due to erroneous values obtained from memory by CALL S().

In a later issue I will present a discussion of the PRK/Stats file structure obtained from information supplied by RICHARD BLANDEN (his Disk Controller Call article is still on the cards - pun intended!).

Anyway, the return variable in both CALLs can be used to indicate whether the save or load was successful. The normal error messages may not apply. Two values are used:

```
0    means unsuccessful
1    means successful
```

Note that, as with BASIC's SAVE and OLD, it is possible to achieve a successful transfer of a program file only to find that the data has been corrupted.

With disk transfer, TIDOS (TI Disk Operating System) will automatically verify CALL S() or L() and assign a 1 or 0 to the return variable accordingly.

For cassette transfer, you must go through the usual CHECK TAPE (Y/N) procedure to verify, although if you decide not to, the return variable will already have had a 1 assigned to it!

Note also that a CALL L() cannot normally be performed without a CALL P() first having been executed.

## CALL L()

-----

The LOAD subprogram operates as for CALL S() and is used to load a program format file into the area of memory previously reserved with CALL P().

This ability has a useful side-effect. Because it deals with program format files, CALL L() can be used to load any program format file - including a console or Extended BASIC program - which could enable those with PRK/Stats but no machine code access (i.e., PEEKV, POKEV) to play detective in a limited way and examine other program files. The emphasis is most certainly on playing detective, as Stephen kindly did for me a few years ago when I began to experiment with Enhanced BASIC. You can certainly try to examine such files using PRK/Stats directly, but nine times out of ten the system will 'hang-up' on you.

The better approach is to make use of CALLs G() and H() to see what has been stored. You will have to do an enormous amount of work, because of the need to examine strings in terms of the ASCII codes of their constituent characters, and numbers in the same way (wait until you try and decode Scientific Notation!).

In fact you may find it advantageous to redefine all entries in the header as CHAR with length 15 to begin with.

Finally, a comment on a point raised by JOHN ROE in his letter elsewhere. When I have given CALL P() as part of an example, I have usually used the value 10000 within the brackets to instruct the machine to set aside a chunk of memory.

The General Description provided by TI does warn you that it is best to make all CALL P() values multiples of 256, since all PRK/Stats files are transferred in that form. Technically therefore the value should be 10240 ( $40 * 256$ ) or 9984 ( $39 * 256$ ). Similarly, when using Enhanced BASIC and requiring space for a BASIC program as well as a reserved area, it is better to try and juggle things so that you have enough space allocated for both items. As John says in his letter, you don't have to reserve 10000 bytes EVERY time!

The next few issues should see the articles from PAUL KARIS and FRANC GROOTJEN of Holland, and I hope to end with a discussion of the error messages and their meanings (as well as the circumstances which generate them) and also a dump from disk using Navarone's Disk Fixer to show you how a PRK/Stats file looks on the surface of a disk.

~~~~~

DISKS AND DRIVES

PART I

By T. D. Bell

(The following article is reprinted from the Central Ohio Ninety-Niners, Inc. Newsletter, SPIRIT OF 99 - FEBRUARY, 1984.)

"Computers are the pencils of the information age. Floppy disks are the paper. You are the brains". Popular mythology.

Fact: computers are so feeble minded they cannot remember anything, they must write all information down or its gone. While they may have no minds and absolutely no memories; they can crunch, shuffle and collate fantastic amounts of data and write thousands of words of paper in a split second. The paper? A precision, glass smooth disk of magnetic crystals thinner than a human hair (three mils,  $3/1000$ ). Under ideal conditions, these floppy disks can remember forever.

Let's go back a little. Before magnetic recording tape, there was paper tape or cards with little patterns of punched holes to save, if the computer programs were saved at all. While this was very inexpensive and great for throwing out of windows during parades, it did little to promote personal computing. It was also very fallible and slow. Then came cassette tape and very expensive 8" disks. While cassettes were faster, they were still relatively slow and there was no way to randomly access individual files. Unless you knew where the information was on the tape, you had to start at the beginning and go through it until you or the computer found the file needed.

The other option? 8" disks developed by IBM which were too costly. Without the mini-floppy technology developed by Shugart Associates in 1976, which offered the first affordable  $5\frac{1}{4}$ " disk drive, personal computing might never have got into the home.

DISK MAKEUP

---

Not unlike audio cassette tape, floppy disks store information on magnetically charged iron oxide particles that coat the surface. Iron oxide is not ground up metal but clinically grown crystals, each having the ability to hold a magnetic charge. This is where the likeness ends.

The crystals that coat recording tape are aligned in the direction of the tape's travel, allowing them to receive the highest degree of magnetization from the recording head. The crystals on the surface of a disk, although very similar to Type 1 recording tape, are in a random order due mostly to the way they are manufactured and various factors involved in the technology of disk recording. While the base film of a disk is about six times thicker than



standard tape. (Base film = 3/1000 of an inch; magnetic oxide = 90/1000000 of an inch).

Quite simply, the reason for the thinness is less chance of a spillover and overwrite of digital information and a better chance that old data has been completely erased when written over.

Digital information is recorded on a series of tracks (concentric magnetic rings), not grooves, put there by the micro system's disk drive read/write head. That's right, when you buy them, they are really blank. Most disks are rated to contain 48 tracks per inch (tpi), while the actual area used is only 40 tracks (single density), or 96 tpi formatted to contain 80 tracks (double density). The TI system uses single sided, single density and double-sided.

When a disk is formatted, track and sector information is written to the disk. Each system formats a floppy according to its own initialization or formatting program. Each track is divided into a number of sectors, enabling it to store and locate information more easily. TI uses 0-358 or 718 sectors. Another section of the disk, called the directory, stores the name of the file along with the track and sector location, where they can be found, and the amount of free space available on the disk. The directory uses the track and sector information to access, almost instantaneously, any file at random. The read/write head searches out the correct data by using the timing hole as a guidepost and a "format" something like a ZIP code) as a destination. This is a vast improvement on serial storage used by cassettes.

#### DISK MANUFACTURE

-----

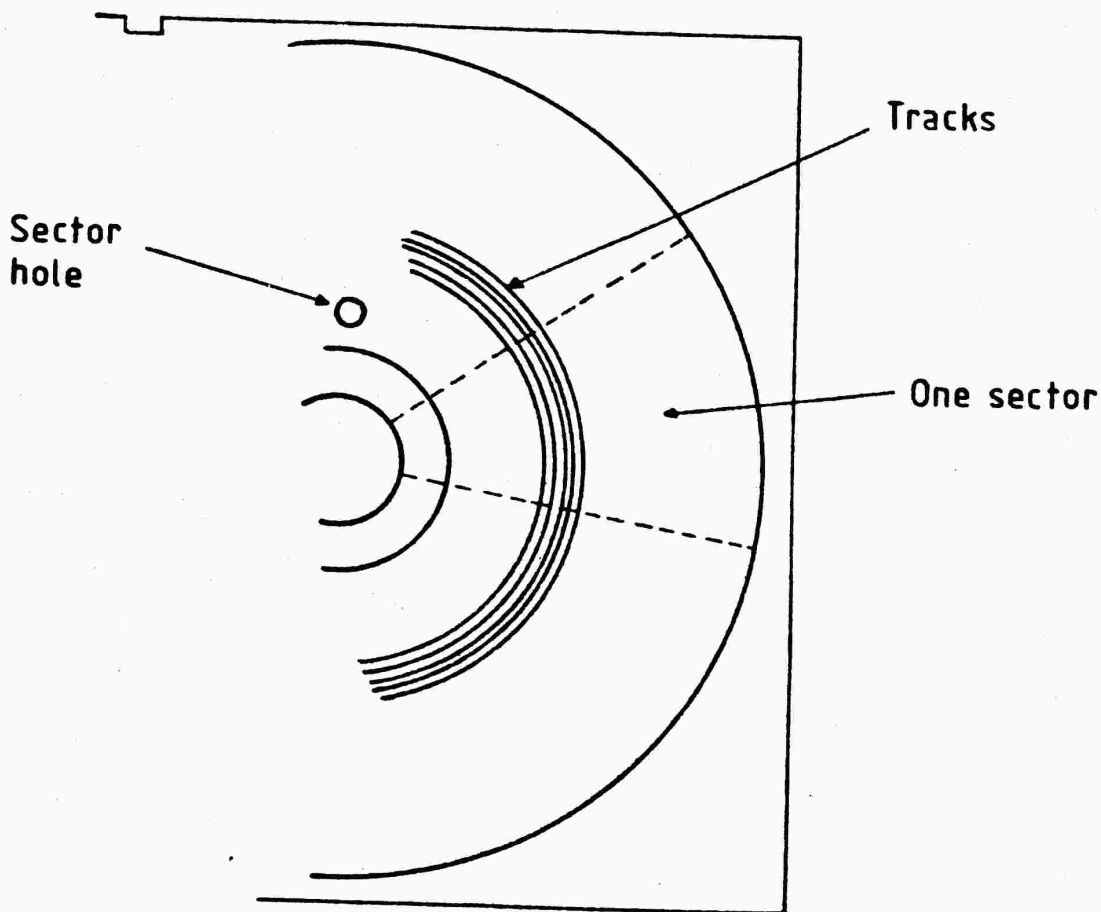
The manufacturing of a floppy disk begins with a continuous sheet of plastic film called a WEB. This is run through a coating machine containing a flurry of 40% magnetic oxide and 60% binders, resins and lubricants. The coated sheets are dried in large high temperature ovens and calendered to compress and polish the surface.

The material is then rolled into 'BUTT ROLLS' and the procedure is repeated for the other side. To stabilize the polyester base, the disk goes to the curing ovens for 24 hours. Next, a stamping machine cuts out the disk shape and the centre hole, along with any required index or sector holes. Finally, a chemical bath gives the disk more durability.

At the same time, in another part of the factory, little jackets are being made from PVC (cured 24 hours in heat chambers), to house and protect the disks. A non-woven synthetic liner is then welded into place as a bearing surface and cleaner for the disk. It is cut to size, folded into shape, with the flaps sonically welded. Centering, indexing and write-protect holes are punched out along with the head-access slot. The disk is then slipped inside the jacket and tested by a certifying machine. The machine fills the entire surface of the disk with data and checks for errors.

Errors found on one side of the disk make it 'single-sided'. If errors are

found on the second side, the disk is useless and is trashed. If no errors are found on either side, it is 'double-sided'. The tests also determine single or double density. A single density disk is usable at low packing levels but cannot meet the rigid requirements of high capacity. The last flap of the jacket is then sonically sealed and a hub-reinforcing ring added. The disks are then labelled, packaged and shipped.



LETTERS

Tel: (0855) 880229

12 THE TENNIS,  
CASSINGTON,  
OXFORD,  
OX8 1EL.

5th. May 1985.

Dear Peter,

Thought you might be interested in the following note relating to the use of Microline 82a printers when the £ sign and # are both required, also when a 0 and a Ø are both required.

You will be aware that in the standard "British" character set a "£" is printed when "Shift 3" is exercised and that in the standard "U.S. ASCII" character set, "#" is printed when "Shift 3" is used.

Also, in my case at least, the "zero" does not have a "/" through the centre of it.

Since the introduction of the G0(+ 3 letters) sequence of Amateur Radio callsigns the radio licensing body has dictated the use of "zero" or "Ø" in both the spoken and written word. To facilitate this use when writing letters using Dave Hewitt's (the Headington, Oxford one!) superb Wordwriter program, I consulted the wizard on the possibility of defining the "£" and the "Ø" in text mode.

Following his suggestion to start with I printed out all the character sets available on my printer from 1 through to 16. This produced a most interesting discovery. Namely, that character set 14 (shown in the Microline 82a manual as unused) contained the "#", the "£" and a "Ø". The character code for the "#", predictably, turned out to be 35 but the character code for the "£" was found to be 173 and the character code for the "Ø" was 169.

On feeding this information back to the Oracle of Headington he, true to form, came up with the appropriate modifications to the Wordwriter program to exploit fully this facility.

It may be that other readers of TI Lines with Microline printers would be glad to know of this potential. My printer is now switched permanently to reproduce character set option 14 and this has no effect on any other use to which I put it.

As a comment, may I say that I have seen many textwriter programs for the TI-99/4A (except TI Writer) and none of them has come up to the performance and efficiency of Dave's Wordwriter.

To prove the points that I have made I shall use:-

|            |                                |
|------------|--------------------------------|
| Shift 3    | to produce: #####              |
| Function 3 | to produce: ££££££££££££££££££ |

© on its own to produce: 000000000000000000

I hope the above may be of interest and you are welcome to publish what you wish. Thank you very much for a very interesting magazine especially the most recent bumper issue.

Yours sincerely,



Brian Abrams.  
(Amateur Callsign G0BII)

---

FORTH Up-date for OTIU members. May 1985

Corrections to SPRITE primitives

LIST screens 58 & 59: What date is on Line 0?  
If it is 1982, the following corrections need to be made (remember to amend the dates too!!!)

SCREEN 58. Amendment to SSDT.  
Screen line 10, ends VFILL ENDIF  
Change to read: VFILL 300 ! SATR ! ENDIF

SCREEN 59. Amendments to SPRITE and MOTION:  
Screen line 7... insert a DROP between VMBW and ENDIF  
eg line should read... ..I 2 VMBW DROP ENDIF 4 +LOOP ;

Screen line 9... presently... hR 100 U\* DROP + SP@ ..  
should be:... hR SLA SWAP OOFF AND OR SP@ ...

These corrections were issued 20 October, 1983, if you want to insert the right date in line 0!



Stephen Shaw

7 Harbury Close,  
Matchborough West,  
Redditch,  
Worcs. B98 OEF

1st April, 1985.

Dear Peter,

Herewith some comments on the material in issue 13 just received.

Firstly, Jeremy Bygott's letter reminds me that I have a bug in my PRK module which has not so far been mentioned in any articles in the magazines I take. When I use the routine to alter a record I find that the next record has been affected. The first character has been erased leaving a blank space where it was. OK you think, so put the character back in. This I did but this amounts to an alteration of the record which then erases the first character of the next record. It gets a bit weary if after having amended the first record of about 120, you have to work your way down all the rest replacing the first character.

After studying your articles on Enhanced Basic and the TI pamphlet, I have managed to produce a working file handling program, which I hoped to use to do some complex data processing by routines not available in the module. Of course, the idea immediately sprang to mind. Why not use this program to do the alteration of records. I wrote a routine and included it in my program and it worked alright but dammit it did the same thing as the module routine, so I am back where I started. It seems likely to me that there is a bug in the Getput routine which causes it. Has anyone any ideas on how to get over this short of buying another PRK module which may have the same bug or even a different one.

I notice that in all your articles on Enhanced Basic you use the Call P command to reserve 10000 bytes, as indeed does the TI pamphlet and other writers on the subject. In the early days of my experimenting I was not clear as to what exactly I was reserving this memory for, so I slavishly followed the same practice. My first program was quite lengthy and it kept crashing with not enough memory. Increasing the 10000 to 11000 or 12000 was no help at all. For a time I abandoned these experiments having got hooked on Peeking with my Minimem. Your articles revived my interest and things were beginning to get a bit clearer. The particular application I was experimenting with involved 120 records of 3 fields, but the average number of bytes in each record was only about 20. So 120 x 20 bytes is only 2400, so why was I reserving 10000 bytes? I tried Call P(4000) to allow plenty of room for the file and the program worked perfectly. The moral is count your bytes before calling P.

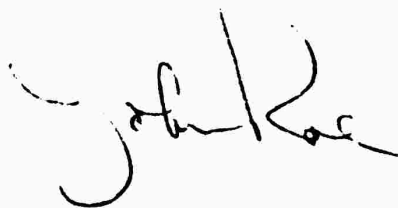
Tony Ralph's article on power supply circuitry interested me. I think he's got it wrong when he refers to a smoothing choke as a resistor. Like any other electrical component it has resistance but that is not its function. The essential feature of a choke is its inductance which is a measure of how much the magnetic field induced by the current affects the current itself. The current produced by a full wave rectifier is not fully DC. It can be regarded as a combination of a DC current and an AC

current with the amplitude of the AC component being less than the DC current. The purpose of the choke is, as its name suggests, to choke off the AC component and it does this by producing an impedance (not resistance) which affects only the AC component and leaves the DC component untouched. Inductance itself is measured in Henrys, but the impedance to AC current produced by that inductance is measured in ohms, which may explain the confusion with resistance. The circuit also includes a large capacitor which works the opposite way to a choke. While being 100% resistant to DC it offers an easy path to AC, so after the choke has done its work any remaining ripple is shunted off by the capacitor and doesn't reach the operative components.

I have at last after a very lengthy wait received my Myarc RS232. I'm now looking for a printer. There has been some correspondence in TI'MES a few months ago that the TI RS232 doesn't have a true parallel interface so that it is difficult to get many printers to work with the TI. However, Myarcs manual suggests that although they have produced an identical interface to TI, they have also included a further option which converts it into a fully standard parallel interface. This is the .HS option (short for hand shake). Have any readers any experience with this? If so, will it work with all the well known printers?

Talking about printers, I see that Tandy are now offering their 4 colour printer CGP115 for £99 for this month. This is quite a bargain if that is what you want. I could use its 4-colour facility in one particular application and could tolerate the  $4\frac{1}{2}$  inch paper for most others but it is quite slow - 12 cps, so if you want it for any considerable volume of text it would be a question of switching on and popping out for a couple of pints.

Yours sincerely



-----  
THE PHILADELPHIA AREA TI-99/4A USERS GROUP  
-----

Taken from the newsletter of that group, dated December 15th 1984

### CLEANING YOUR KEYBOARD

The following tip comes from the 'SPRITE', a monthly newsletter of the 9900 USERS' GROUP INC, by KEN CHANDLER.

Those of you who have had your TIs for a while may develop a small problem with repeat-printing. This occurs when you press a given key once and it returns multiple characters. The problem does not usually affect all the keys, just certain ones which will vary from console to console. It can make word processing a real pain in the neck, since you keep on having to go back to correct the errors.

You can always return your console to TI for servicing, but if it is outside the warranty it will cost you up to (£65)!

The sketch at the end of this article details the construction of each key assembly. The keycaps press-fit down inside the yellow key sleeve. When you press the key down, the keycap/key sleeve assembly slides down inside the keypost, lowering the contact opener and allowing the contacts to close. The spring pushes the assembly back up and opens the contacts again. The contacts actually close with a wiping action intended to keep them clean. However, when they do become dirty, this wiping action can cause multiple closures, which results in several 'keypress' signals being sent to the cpu.

#### WARNING: FIRST UNPLUG THE CONSOLE

The first thing to do is to remove the keycap. You will need something with a tiny hook on the end to slide under the keycap and pull up. A bent paperclip will probably work or maybe a knitting needle. You will need two of them, one on each side of the keycap so that you can pull straight up without bending. If you only have one hook, use your finger on the opposite side of the key to apply pressure and prevent the keycaps from cocking sideways. Use even, steady pressure and the keycap will slide right up. Occasionally the keycap is stubborn and the whole keycap/key sleeve assembly pops out. DON'T PANIC! Simply remove the keycap from the assembly and press the yellow sleeve down over the spring, inside the keypost, until it clicks home. (See Figure 2).

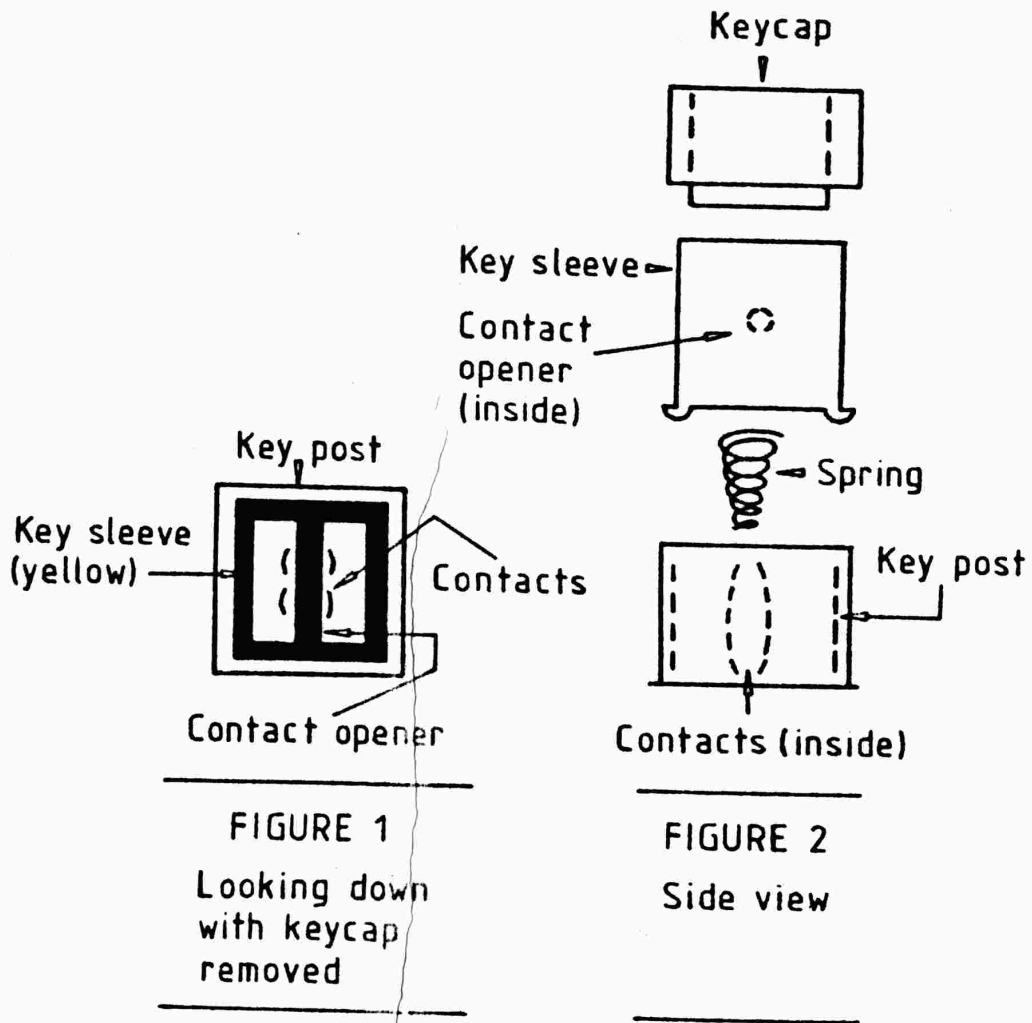
You should now see something that looks like Figure 1. To clean the contacts, you will need a small cotton swab and some ISOPROPYL ALCOHOL (don't use regular rubbing alcohol). Dip the swab in the alcohol and carefully clean in between the contacts, being careful not to bend them out of position. After cleaning, push the yellow key sleeve down with your fingernail and make sure the contacts close properly. Also, if the key sleeve sticks or binds in its travel, put a small amount of silicon lubricant between it and the keypost.

Now you are ready to re-install the keycap. Just press it down inside the key sleeve and you're done!

! 2021 Note:

TI used many types of keyboard in their consoles and the construction method of your keyboard may differ.

2021 Note: This diagram refers to one type of keyboard used. TI used many types of keyboard in their consoles and the construction method of your keyboard may differ.





# TI WRITER

## ----- UNDOCUMENTED FACILITIES -----

Relating to the TI-Writer Editor, from information supplied by HOWARD GREENBERG, and from further research.

It is possible that there are several versions of the TI-Writer manual knocking about, because no mention is made of the facilities (discussed later) in my manual while others whom I have contacted apparently do have them documented.

I am not sure of the usefulness of either facility, but they both exist, and it is always profitable to know all that there is to know about our particular product.

Incidentally, an Extended BASIC loader exists in the Public Domain which enables the TI-Writer program files (EDITA1, EDITA2, etc.) to be loaded and run, and should enable a large number of XB owners to perform word processing without the need for the TI-Writer module. However, I have yet to confirm that the manual has been placed in the Public Domain also; I should have the information shortly.

HOWARD GREENBERG has noted that the FS (Find String) facility in the Editor has a mode of operation similar to that found in the Editor/Assembler's Editor. You can specify:

```
n /xxx/
```

which will find the nth occurrence of xxx beginning immediately after the current cursor location. The fifth occurrence of SPONDULIX can thus be found with:

```
5 /SPONDULIX/
```

The same provision is made for RS (Replace String). The seventh occurrence of RAMPANT SQUITS can be replaced with TROTS thus:

```
7 /RAMPANT SQUITS/TROTS/
```

Further research has shown that the other Find and Replace parameters - namely, Start and Finish columns within which corridor the Find or Replace may operate - may also be specified:

```
n s f /COOEE MISTER SHIFTER/GOOD MORNING VICAR/
```

will find the nth occurrence of the first item between columns s and f and replace it with the second item. Should you wish to, that is.

During the research, I noticed options available with the PF command in the Editor which hadn't previously made themselves known to me. I am now aware that you can strip out Control characters before printing, which can be useful under certain circumstances. I have a listing of a BASIC program which contains characters with ASCII codes less than 32 embedded within it. I can see them clearly on screen when using TI-Writer, but if I attempt to PF normally they act as commands to my OKI Microline printer, causing no end of problems.

Using C RS232.BA=1200 all such characters are removed before printing. The only drawback is Escape sequences which involve mixtures of Control and normal alphanumeric characters: the non-Control characters are left behind, which tends to make the exercise a little pointless.

I am also now aware that the line numbers can be printed as well with the L command, although you do lose the last 4 characters on every line - something of a problem with an 80 column piece!

Finally, I now know that I can PF using the FIXED 80 format - normally it is VARIABLE 80 - which might be useful to someone printing to a disk file (perhaps from TI-Writer to Editor/Assembler ?).

In each case the other facilities - like specifying a start and finish line - can also be utilised, and indeed, you can mix them up as you wish:

1 E C F L RS232.BA=1200

and so on.

I'm beginning to wonder what else lurks inside those EDITA files...

-----  
C L O S E F I L E  
-----

THE LATE, LATE SHOW  
-----

My attempts to get TI-LINES out on time seem doomed to dismal failure. Every time I try to catch up, I fall further behind. Up until now I had managed to put the newsletter out at least within the month that its cover specified, but this month has been the worst to date. Other matters (like work!) have managed to interfere so often and so successfully, and as I near the final stages in this month's output another obstacle rears its detestable head. The photocopier which I normally use to produce the reduced version of the master copy has gone on the blink, and true to NHS form it is not being repaired until there is a 'q' in the month. There are no other reduction copying facilities open to me, and I'm damned if I am going to pay the exorbitant commercial rate (which recently would have cost me over £140 to do £25-worth of work!), so I will be placed in the daft position of beginning volume 2 before the last issue of volume 1 has been published, let alone distributed!

I may even end up distributing both issues together in an effort to keep some semblance of order - at the time of writing (June 3rd) I should have had issue 1 of volume 2 ready to go out anyway. I shall carry on regardless, in true British tradition, except that the only reason my upper lip is stiff is that there is a huge chunk of plastic behind it...

Hope you will bear with me,

Good programming, and better organisation,

Pete Brooks

This page is blank

This page is blank