

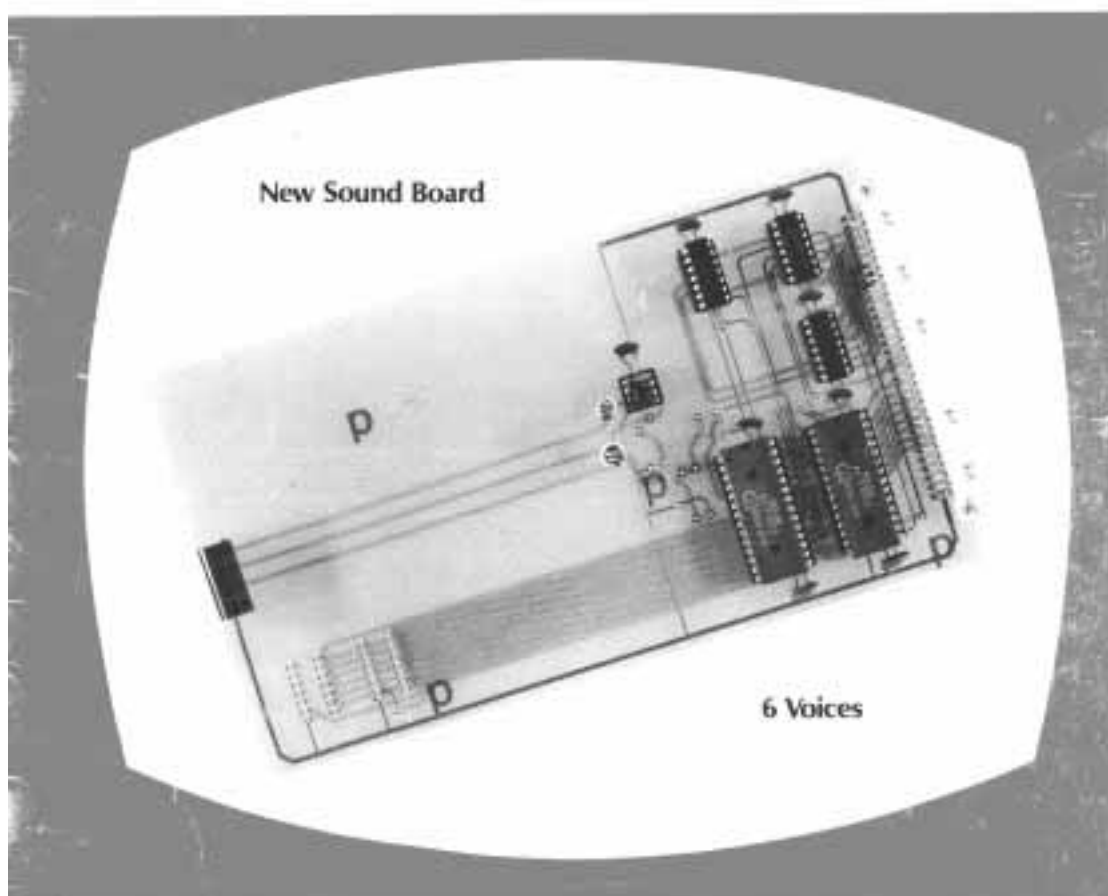
MICROTAN WORLD

Volume No.1

Dec.-Jan. 1983

Issue No.5

Programs · Reviews · Your Letters



SOFTWARE ~ NEW BOARD

PUBLISHERS

MICROTANIC COMPUTER SYSTEMS LIMITED

16 UPLAND ROAD, DULWICH, LONDON SE22. 01 693 1137

SOFTWARE FOR THE MICROTAN SYSTEM

REGISTERED OFFICE
=====

235 FRIERN ROAD
DULWICH
LONDON SE22

MICRO COMPUTERS
CONSULTANTS
SOFTWARE
SERVICES
RENTALS
HARDWARE

SHOWROOMS
=====

16 UPLAND ROAD
DULWICH
LONDON SE22
TEL 01 693 1137

=====

MICROTANIC SOFTWARE PRODUCT LIST

=====

ADVENTURE 1	7K M/C	5.95	TOOLKIT 1	2K EPROM	22.50
ADVENTURE 2	7K BAS	6.95	MICROTANTEL	2K EPROM	16.95
ADVENTURE 3	16K M/C	9.95	HI RES TOOLKIT	2K EPROM	16.95
ADVENTURE PACK 3 x 7K BAS		14.95	FORTH LANGUAGE	IN EPROMS	34.95
THE DEFENDER	7K M/C	6.95	2 PASS ASSEMBLER	EPROM	34.95
TANK RAID	16K M/C	9.95	TOOLKIT 2	2K EPROM	16.95
EARTH ATTACK	7K M/C	6.95	ASIMOV WORD/PROC	EPROMS	49.95
SPACE INVADER	3K M/C	5.95	FILE UTILITIES BAS TAPE		9.95
THE GOBBLER	6K M/C	6.95	M/C RELOCATOR M/C TAPE		5.95
SUB STRIKE	5K BAS	5.95	EPROM PROGRAMMER SELF BUILD		9.95
STAR CHESS	3K M/C	6.95	TEXT PROCESSOR M/C TAPE		14.95
3D MAZE	7K BAS	6.95	BAS LINKED ASSEMBLER TAPE		6.95
3D OXO	7K BAS	6.95	MORSE CODE GENERATOR TAPE		4.95
GRAPHIC PUZZLES	7K M/C	6.95	SINCLAIR PRINTER INTERFACE		29.95
MOLE SWAT	7K BAS	6.95	DISPLAY PLANNER M/C TAPE		12.95
GAMES PACK 1 3 x 7K BAS		8.95	E2 EPROM EXTENSION CARD		22.50
GAMES PACK 2 3 x 7K BAS		8.95	H2 EPROM EXTENSION CARD		24.50
GAMES PACK 3 3 x 7K BAS		8.95	25 x 64 COLOUR VDU BOARD		85.00
GAMES PACK 4 4 x 7K M/C		8.95	LOOSELEAF DATA BASE TAPE		19.95
AUTHOR	7K M/C	6.95	NEW SOUND BOARD		19.95

UNIVERSAL EPROM ERASER FULLY BUILT TO ERASE UP TO 8 EPROMS AT A TIME 34.95

HI RESOLUTION TOOLKIT 16.95

TOOLKIT - ADVENTURE GAME - 3 PASS ASSEMBLER AVAILABLE FOR NEW SCREEN BRD POA

PILOT LANGUAGE NOW AVAILABLE FOR THE MICROTAN - THIS EXTENSION TO BASIC WILL
ENABLE YOU TO WRITE AND RUN COMPUTER ASSISTED LEARNING (CAL) AND COMPUTER
ASSISTED INSTRUCTION(CAI) PROGRAMS ON YOUR MICROTAN - (TAPE OR EPROM) 19.95

LARGE SELECTION OF COMPUTER BOOKS NOW AVAILABLE FROM STOCK - WRITE FOR DETAILS

If You have any programs or Hardware ideas please contact Mr. David Northway

=====

ALL PRICES INCLUDE V.A.T. BUT PLEASE ADD 00.60p FOR POST AND PACKING

MICROTANIC COMPUTER SYSTEMS LTD.

Computer Manufacturer
Software - Hardware - Books

Registration No: 1668843

V.A.T. No. 237794718

16 Upland Road
Dulwich, London SE 22

Telephone: 01-693 1137

Contents



Who's Who and What's What	4
Shop Window	4
The Inside Story	5
ED's Page	6
Veiw Point	7
Warm Start	8
The Sound of Music	9
Price List	10
Order Form	11
Space Trader	12
Advertisements	14
Joysticks Revisited	15
Etch-a-sketch	16
Foolproof Input	18
Data Corruption on Tanram	19
Tanram Circuit Diagram	20
The Three Piece Suite	21



WHO'S WHO AND WHAT'S WHAT



The Company

Microtan Computer Systems Ltd, 16 Upland Road,
Dulwich, London SE22. 01 693 1137

Managing Director	David Northway
Director	Charles Cooper
Director & General Manager	Robin Northway
Design Consultant	Dr. P.N. Gaunt

Microtan World

Published by	Microtan Computer Systems Ltd
Editor	Deryck M. Sutton
Repairs & Technical	Andy Parnell & Clive Reynolds

The Microtan System

1K Computer in Machine Code

Microtan 65 + Keypad
+ u/l case + Graphics

8K Computer Expansion

As above + Tanex +
Mini Motherboard +
Power Supply + Full
ASCII Keyboard + Basic

48K Computer Expansion

As 8K + Tanram + MPS2
Power Supply + System
Motherboard.

Additional Expansion

HI RES Board. VDU
Colour Board. Sound
Board. DOS Board.
MASS Storage Eprom
Board

Languages

Basic. Forth. Pilot
Assembler

Also Available

Microtutor
Controller Board



SHOP WINDOW

Main Microtan Dealers

Waltham Forest Service Centre
889 Leabridge Road,
Nr. Whipps Cross,
Walthamstow. E 17.
Tel: 01 520 7747

Micron Computer Equipment Ltd.
Devonshire House,
Devonshire Square,
Loughborough,
Leicester. LE11 3DW

Overseas Dealers

Gloor Instruments Ltd,
Schwizerstrasse 2,
8610 Uster,
Switzerland.

J Muller,
Korblumeinweg,
Ebersheim,
West Germany.

Micromation Systems Ltd,
21 Elma Park Centre,
Edendale Road,
Edonvale 1610,
South Africa.

Mr Beni
Mayost
Rehov Hashoftim 29
Mercaz Rasco
Kiryat Motzkin
Israel.





The Inside Story

by David Northway

This is the 5th issue of Microtan World and it comes out at the festive season so I would like to take this opportunity to wish you all a very Merry Christmas and a prosperous New Year.

1983 has been a difficult year in many ways but the rewards of success as far as the Microtan computer is concerned has made it all worthwhile.

1984 promises to be just as exciting with the expansion of our system becoming a reality. Research and development continues at a fast rate and interest in the system is growing week by week.

I would apologise to all those who suffered delays during 1983 in the supply of hardware etc. Most items are now in stock and we are trying to keep all stocks at a high level. We now have stocks of the new Sound Board, the Eprom Programmer Board and the new cases.

So I am looking forward to 1984 and working towards our new concept. To this end I think you will find Dr Gaunt's articles very interesting.

Our next issue will be our 1st anniversary until then.





ED'S Page

Issue No. 5 means that we have been going for 10 months already and in 2 short months we will be 1 year old. My thanks to you all for your help.

This issue sees the new Sound Board available and a review is on another page but let me say that this board gives us a quality which turns Microtan into a superb musical instrument and if the output is fed into a high quality amplifier system the results have to be heard to be believed.

New software is under review and if suitable we hope to tell you more of this next issue. You will also see from Dr Gaunt's article that the future of our system is well under way and we hope to expand on this theme shortly.

If like me you have been looking at the range of computers now available you may well agree that the Microtan offers real flexibility and ability to expand neatly with an opportunity for the experimenter to get to grips with the heart of the computer.

I am also beginning to get some real dialogue with you the readers and it is obvious that many of you have a real ability in programming even in machine code. This can only bode well for us all. We already have two contributors starting to talk to each other by Modem which could be the beginning of something very exciting.

It has been suggested that we might offer programs that are printed in the magazine on tapes. So I will make this suggestion. If you would like a tape of any printed programs firstly ring me or write to me to check if a copy is available. Then send me a blank tape with a s.a.e. and 50 pence to cover administrative costs and I will do the rest. In future issues of the magazine I will try and let you know whether a tape can be made available.

Keep those articles coming and also your letters

Merry Christmas

PRINTER'S NOTE: We apologise for the delay in publishing this issue of Microtan World but some of the material was a bit slow in coming through and finally it all caught up in our Christmas rush. We have endeavoured to get it off in the post and hope that you get it in time to read in the holiday period; We also regret the omission of Dr. Gaunt's article on the new system but at the time of going to press it had not arrived, this does leave something to anticipate in the next issue. The new sound board on the cover was just slotted together to show its appearance, without the paper work to hand, our apologies for any misplacements.



VIEWPOINT

Dear Deryck,

Herewith some fairly random jottings re Microtan World and Microtan in general:-

SIMPLE SIMON

Surely, a great majority of Microtan World readers are already in possession of a Basic system and manual. Most of the information published is contained in the manual-granted S.S explains it in a more 'User Friendly' manner - but is this duplication really necessary?

THE PHOTOS

In the last two editions of M.W. we have seen 5 pages devoted to photo's of the Microtan system boards. Would it not be more useful to have written descriptions, specifications etc of the boards (perhaps block diagrams too) so that readers can see what they do rather than what they look like.

TANEX BOARD and others?

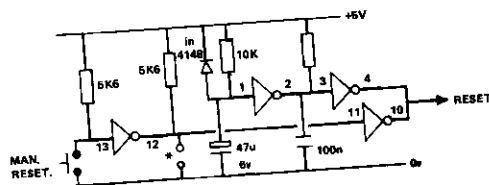
The whole system is primarily for experimentors. Experimentors have a habit of taking cards out, adding to (or subtracting from) them then re-inserting them. With the rack/Motherboard system this is OK BUT what about the tape connector on TANEX? You probably know that small screwdrivers are ultra-sensitive to Gravity (and have been known to defy it at times) so the chances of finding one to dis or re-connect the tape leads are minimal. I have replaced the screw connectors with a plug/socket system using the 0.1" MOLEX pcb mounting plugs used in some of the ET1 boards for Microtan.

AUTO/MANUAL RESET

The following circuit is a slight modification of one published in ELECTOR Nov 83. It provides an automatic RESET on power-up and has been modified so that a more usual push-to-make manual RESET switch can be used. The circuit can be made up on a small piece of Vero-board and fits in the Keyboard case or where ever your RESET switch normally resides. NB The spare gates of the 74LS06 can be used for other external resets or for playing with the IRQ line (more on which later).

Despite the above moans I think that Microtan World is a great improvement on anything that came before. More power to the Editorial Elbows!

Peter Chamberlain
Folkestone
Kent



- In the original article a push to break switch was used here and the extra inverter (13-12) not used. The manual reset switch can of course be connected between the RESET line and 0v, but I had to use up the invertors somehow!

Dear Sir,

I have recently received the first three issues of the magazine and I feel that you are off to a very promising start. There are as yet some obvious fillers included but one assumes that these will be dropped as more system specific matter becomes available. Anyway, having lately been force-fed a great deal of useless information about plastic toy computers, I am in no mood to criticise.

The most intriguing item to date is the rather vague mention of 'the new Microtan System Nine'. Is this a private project of Dr. Gaunt's or is it being developed for sale to us peasants? The really vital question has already been asked by Mr. Marten but he doesn't seem to have got an answer - - - has somebody actually written TBUG 09?

A small gripe to finish with - - - REPAIR BENCH, Issue 1, page 23, paragraph 2. It is unhelpful to tell me that I have been conned into buying incompatible boards. I already know that! Please - WHAT small modification to the Tanex Board.

Incidentally, since you seem to be publishing a set of circuit diagrams, you might slip the Motherboard diagram to the top of the heap and oblige - - -

Niel Mackay
St. Leonards-on-Sea
Sussex

WARM START

M C Matthews B.Sc., FRICS

Seeing Mr. Stanton's note about a warm start, I thought you might be interested in my version.

Originally I wrote this for TANBUG V.1 which of course had no warm start and I was very aggrieved when, on receiving TANBUG V.3B found that the warm start provided in that was nothing like as effective as my own version, added to which my own version wouldn't work! - I have the 64 col x 25 line VDU board.

My principal objection to the one in V.3B is that it will not recover you from the state of affairs when a verify has failed after an attempted save.

The enclosed program has so far recovered me from every position I have got into, unless the program itself has been overwritten, in which case of course nothing will ever help.

In essence it is the first part of the basic set-up routine with three modifications, only the first one of which is essential.

At B697 instead of running through the memory to find out where the top of the memory is and to re-set it all, it finds the position of the end of the program and inserts it in the appropriate stores, it then goes on to complete the setting up and jumps to the start.

The first variation comprises changes in the figures stored in locations 22H and 23H (34, 35 decimal). The amended figures jump to a little routine attached to the end of the warm start, which makes the use of X = USR(X) much easier. The address of the routine to be called is inserted in decimal between the brackets and this routine converts that address to hex, stores it in the appropriate locations and then jumps to it.

The third alteration is a trivial one. Location B669 has been changed from 50H to FFH which has the effect of setting the terminal width to 255 instead of 80.

The routine is completely re-locatable except of course for the sub-routine. This is called three times and the appropriate addresses have to be changed. If B 6D4 is moved then obviously the addresses in B660 and B662 have to be changed to agree.

If it is of interest I have a short Basic program that I call Getloc which can be used to find the positions of up to three different hex numbers in the machine code program.

It is written in Basic simply because it was quicker to write, although I find in practice it is quite fast enough for my purposes. One of these days I shall probably get round to re-writing it in machine code. In the meantime, however, if you wish to check through a program to see where, for example, a particular sub-routine is called, it will do the job accurately and with reasonable speed. If you are interested I will send you a copy.

One final point. I have a mains filter obtained two or three years ago from A.J. Harding, the software firm, which I have found highly satisfactory.

:: Warm start U3B Page 01

```
B628 A90D LDA #000D )
B62A 850C STA #000C
B62C A900 LDA #0000 )
B62E 80C0BF STA #BFC0 @?
B631 8515 STA #0015
B633 8516 STA #0016
```

```
B635 A0FF LDY #00FF #
B637 84A9 STY #00A9 )
B639 98 TYA
B63A A976 LDA #0076 )u
B63C A0E0 LDY #00E0 `
B63E 8518 STA #0018
B640 8419 STY #0019
B642 851B STA #001B
B644 841C STY #001C
B646 A9C1 LDA #00C1 )A
B648 A0CF LDY #00CF 0
B64A 851D STA #001D
B64C 841E STY #001E
B64E A994 LDA #0094 )
B650 A0D1 LDY #00D1 Q
B652 851F STA #001F
B654 8420 STY #0020
B656 A94C LDA #004C )L
B658 8517 STA #0017
B65A 851A STA #001A
B65C 85C3 STA #00C3 C
B65E 8521 STA #0021 !
B660 A9D4 LDA #00D4 )T
B662 A0B6 LDY #00B6 6
B664 8522 STA #0022 "
B666 8423 STY #0023 3
B668 A9FF LDA #00FF )#
B66A 8531 STA #0031 1
B66C A938 LDA #0038 )8
B66E 8532 STA #0032 2
B670 A21C LDX #001C "
B672 B051E0 LDA #E051,X =Q`
B675 95E1 STA #00E1,X a
B677 CA DEX J
B678 D0F8 BNE #B672 Px
B67A A903 LDA #0003 )
B67C 85C2 STA #00C2 B
B67E 8A TXA
B67F 85D7 STA #00D7 W
B681 8537 STA #0037
B683 852F STA #002F /
B685 48 PHA H
B686 852E STA #002E .
B688 206AC9 JSR #C9BA :I
B68B A088 LDX #0088 "
B68D 8685 STX #0085
B68F A5FF LDA #00FF %#
B691 85A6 STA #00A6 &
B693 A997 LDA #0097 )
B695 85A7 STA #00A7 /
B697 A904 LDA #0004 ) Find end of program
B699 8541 STA #0041 A set up in hex
B69B A000 LDY #0000
B69D 8440 STY #0040 @
B69F C8 INY H
B6A0 20D1B2 JSR #B2D1 Q2 Check for Y = 0 and
if it does INC S41
B6A3 8140 LDA (#0040),Y 1@ Get contents of Loc.
B6A5 D0F8 BNE #B69F Px
B6A7 C8 INY H Yes, try
```

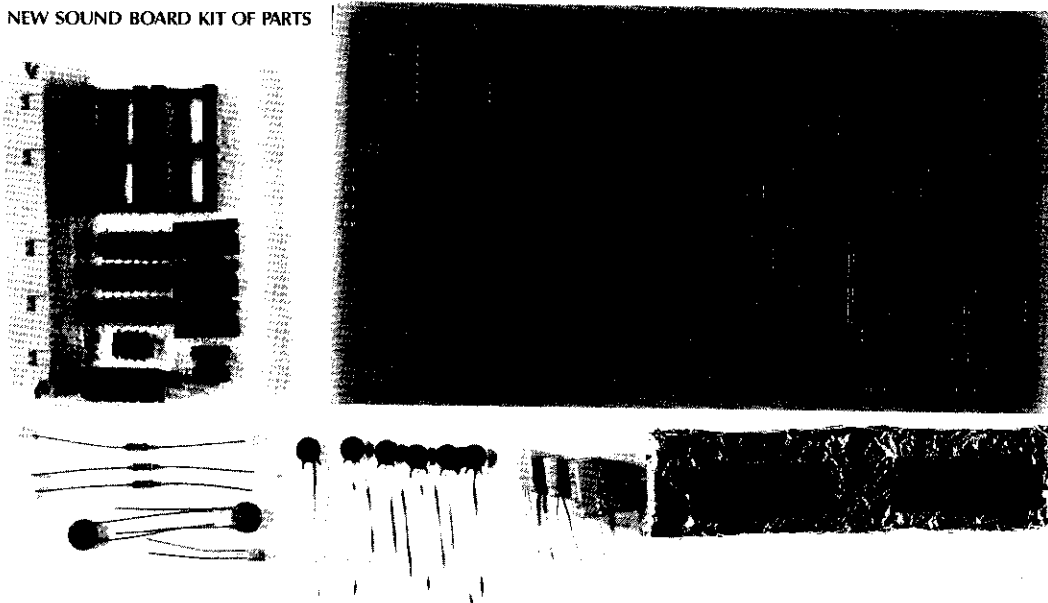
B6A8 20D1B2 JSR #B2D1	Q2	next loc.	B6C1 A910 LDA #0010	)
B6AB B140 LDA (\$0040),Y	10	No, back to start	B6C3 A0CA LDY #00CA	J
B6AD D0F0 BNE #B69F	Pp	Yes try third	B6C5 851B STA \$001B	
B6AF C8 INY	H	loc,	B6C7 841C STY \$001C	
B6B0 20D1B2 JSR #B2D1	Q2	No, back to start	B6C9 A94B LDA #004B	>K
B6B3 B140 LDA (\$0040),Y	10	Yes we have it	B6CB A0C3 LDY #00C3	C
B6B5 D0E8 BNE #B69F	Ph	Save location	B6CD 8518 STA \$0018	
B6B7 C8 INY	H	Put it in	B6CF 8419 STY \$0019	
B6B8 20D1B2 JSR #B2D1	Q2	end of program	B6D1 6C1800 JMP (\$0018)	1
B6BB 849C STY \$009C		address	B6D4 20F5D5 JSR \$D5F5	uU
B6BD A541 LDA \$0041	%A		B6D7 6C3300 JMP (\$0033)	13
B6BF 859D STA \$009D				

BD21	CPY #0	
BD23	BNE #BD27	
BD25	INC \$41	
BD27	RTS	

Now get on with the good work
this takes n from X = USR(n), converts it
to hex and stores it in 33 and 34, then
jumps to it

THE SOUND OF MUSIC!

NEW SOUND BOARD KIT OF PARTS



No not a review of the old musical but of the new Sound Board by Microtan.

Yes the new Sound Board is now available from stock. This new board has 6 independant voices and is fully compatible with all systems. The introduction to the manual says:-

The Microtan Sound Board is a flexible sound generating system capable of producing complex sounds under software control. Its applications include Music Synthesis, Modem Tone Generation - sound effects for games and alarms. The board uses two AY 3 8912 ICS giving six independant channels. Each channel can be controlled in pitch amplitude and envelope shape plus mixing a noise source. Each 8912 has an 8 BIT I/O port especially suited to joystick control. The board uses addresses BCO0 to BCO3 in the I/O are of the memory map.

In short this new board is probably the most versatile and exciting addition for some time.

As good as it is it needs something more and that is a good software program. This is available now in the Music Compiler which allows you to write music direct from printed tunes straight into the computer voice by voice and then save the information for future playing.

So we have a superb Music and Sound Board backed by an excellent software program.

We will be publishing routines in future issues which can be included in your own programs.

SPACE TRADER

By Simon G. Jones

Space Trader

Trader is a game for one player.

You are in charge of a spaceship and with the aid of your computer travel between the planets of the known solar system and make a living. If you manage to make a living you should aim to buy a bigger ship but beware, the value of your cargo may change whilst you are in flight, and it may drop in value

The game is written in Microsoft Basic and although written for the Microtan system it should not prove a very hard task to convert to another system since no graphics are used. The program occupies 6.5 K of memory during runtime and is designed for the Microtan screen size of 32 x 16.

When the program is started, the computer asks for a price stability between 1 and 10. A value of 0 means the prices of goods will never change whereas a value of 10 means that every time you buy or sell an item its price will change. The closer the number to 10 the higher the chance of the price changing.

Having answered the price stability question, the screen will clear and stay clear for several seconds. Don't worry. It takes a fair amount of time to generate prices for the whole solar system.

When finished, the computer displays a menu of the available commands. At the top of the screen is the name of the planet you are on. To return to this menu, press the space bar.

Commands available

T: Travel computer. this displays the names of the nine planets on the left of the screen. At the bottom of the screen is the prompt 'From:'. This is waiting for a number - the number corresponding to the number alongside the planet name. The next prompt is 'To:'. You must reply to this in the same manner as above. The computer will then tell you how much fuel you have and the distance between the two named planets. Remember: to leave your current planet you must enter its number to the 'From:' prompt. Fuel used=distance/one million. If the answer to the first prompt was the current planet number then the computer will ask if you want to travel. The reply must be either 'Y' or 'N' (yes or no). If the reply is 'Y' then you enter

a spacewarp and several seconds later arrive at your named planet. If the answer is 'N' then you return to the initial prompt for a command.

P: Prices computer. This has two functions. It can supply the prices of the products bought and sold on a certain planet, as long as you have visited that planet. The layout is similar to the travel computer in that you must select a planet number. The computer then displays the prices of the products both bought and sold there.

The second function is to provide you with the names of visited planets which trade in a named product. You must again supply the number corresponding to the selected item and the computer will provide the names of all visited planets which trade in that item.

S: Sell. This is straightforward. The computer will tell you the price of the product bought and, if you have any, how much. If you have none then the program returns to the command prompt. If you have any of the named product the computer asks how much you want to sell. You must reply with a number greater than or equal to 1 and less than or equal to the amount you are carrying. (Terminated by 'Return'). The computer then tells you the price it will give you and asks if that is fair. If you reply 'Y' then the transaction takes place. If you reply 'N' then nothing happens and you return to the prompt command.

B: Buy. This is similar to the sell option except you must supply the number corresponding to the item you wish to buy. Having entered this the program proceeds in a similar fashion as above.

C: Cargo computer. This tells you what is in your cargo hold, how much you can carry and how much you are carrying.

D: Dump. This lets you dump any product except fuel, for a fixed price of £100 per ton.

You start on Earth with a ship capable of holding 5000 gallons of fuel and carrying 20 tons. You have £2000. The only place to buy a new ship is on Earth, when you must reply to the prompts as detailed above. You get a 50% refund on the value of your old ship.

Good luck and remember: it's a long walk home from Pluto when you have no fuel and money'!!

```
100 PRINT CHR$(12):PRINT:PRINT TAB(4)"*** Star Traders ***"
110 DATA "Mercury","Venus","Earth","Mars","Jupiter","Saturn"
120 DATA "Uranus","Neptune","Pluto"
130 DATA 36,67,93,142,483,886,1783,2793,3666
140 DATA "Uranium","Plutonium","Vanadium","Tantalum"
150 DATA "Osmium"
160 FOR I=1 TO 9:READ P$(I):NEXT
170 FOR I=1 TO 9:READ S$(I):NEXT
180 FOR I=1 TO 5:READ M$(I):NEXT
190 DEF FNR(X)=INT(X/RND(1)+1)
200 DEF FNE(E)=INT(E*10+.5)/10
210 FOR I=1 TO 9
220 S$(I)=M$(FNR(5)):B$(I)=M$(FNR(5))
230 IF S$(I)=B$(I) THEN 220
240 R(I)=FNR(500)+200:P(I)=FNR(500)+200:R(I)=FNR(20):NEXT
250 F=5000:M=5000:W=20:E=3:C=2000:N=3:V(N)=1
260 REM SET CURSOR TO MIDDLE OF SCREEN
270 POKE 10,224:POKE 11,2
280 PRINT "Price stability 0=stable 10= "
290 PRINT "unstable ":INPUT Q
300 IF Q<0 OR Q>10 THEN GOTO280
310 GOTO430
```

```
320 PRINT CHR$(12)
330 PRINT:PRINT " Command ?"
340 GOSUB2180
350 IF A$="T" THEN 510
360 IF A$="P" THEN 810
370 IF A$="C" THEN 1250
380 IF A$="B" THEN 1580
390 IF A$="S" THEN 1340
400 IF A$="D" THEN 2250
410 IF A$=" " THEN 430
420 GOTO340
430 PRINT CHR$(12):PRINT:PRINT TAB(5) "You are on "P$(N):PRINT:PRINT
440 PRINT "T Travel computer"
450 PRINT "P Prices computer"
460 PRINT "C Cargo computer"
470 PRINT "S Sell"
480 PRINT "B Buy"
490 PRINT "D Dump"
500 GOTO330
510 PRINT CHR$(12) TAB(5) "Travel Computer"
520 PRINT "You have"F"gallons of fuel"
530 PRINT "You are on "P$(N)
```

```

540 FOR I=1 TO 9:PRINT I,F$(I):NEXT
550 PRINT
560 PRINT "From ";
570 GOSUB2180
580 A=VAL(A$):IF A<1 GOTO570
590 PRINT F$(A)
600 PRINT "to ";
610 GOSUB2180
620 E=VAL(A$):IF E<1 OR E=A GOTO610
630 PRINT F$(E):FOR I=1 TO 500:NEXT
640 PRINT CHR$(12):PRINT:PRINT "Distance=fuel used"
650 D=ABS(S(E)-S(A))
660 PRINT "The distance from "F$(A)" to "
670 PRINT F$(E) "is "D"million miles"
680 IF D F THEN PRINT "You don't have enough fuel":GOTO330
690 PRINT "If you do, you will have "F-D
700 PRINT "gallons left."
710 IF A<N THEN 330
720 PRINT "Do you want to travel to "F$(E)" ?"
730 GOSUB2180
740 IF A$="Y" THEN 770
750 IF A$="N" THEN 320
760 GOTO730
770 PRINT CHR$(12):PRINT:PRINT "You are travelling through"
780 PRINT "a hyper space warp to "F$(E)
790 N=I:F=DIV(N)=I:FOR I=1 TO 2000:NEXT
800 PRINT CHR$(12):PRINT:PRINT "You have arrived on "F$(N):GOTO330
810 GOSUB2220
820 PRINT:PRINT "1 Prices for any planet visited"
830 PRINT "2 Prices for any product"
840 GOSUB2180
850 IF A$="1" THEN 880
860 IF A$="2" THEN 1050
870 GOTO840
880 GOSUB2220
890 PRINT:FOR I=1 TO 9:PRINT I,F$(I):NEXT
900 P=VAL(A$):IF P=0 THEN 900
910 IF V(P)=0 THEN PRINT "You have not visited "F$(P):GOTO330
920 PRINT:PRINT "Prices on "F$(P)
930 PRINT S$(P):TAB(15):F$(P):"per ton"
940 PRINT "Rocket fuel:TAB(15):F$(P):"per gallon
950 IF P<3 GOTO1020
960 PRINT "Space ships":TAB(15):F$(P):"1000$-Fuel"
1000 PRINT TAB(16):"Capacity:(1000"
1010 PRINT TAB(16):"Max cargo)"
1020 PRINT "Buyer:
1030 PRINT B$(P):TAB(15):F$(P):"per ton
1040 GOTO330
1050 GOSUB2220
1060 PRINT:FOR I=1 TO 5:PRINT I,F$(I):NEXT:PRINT " 6","Rocket fuel
1070 PRINT:GOSUB2180
1080 A=VAL(A$):IF G=0 OR G>6 THEN 1070
1090 GOSUB2220
1100 IF G<6 THEN 1160
1110 PRINT:PRINT "Rocket fuel prices"
1120 PRINT "Seller:-
1130 FOR I=1 TO 9:IF V(I)=0 GOTO1150
1140 PRINT F$(I):TAB(15):F$(I):"per gallon"
1150 NEXT GOTO330
1160 PRINT "COY prices"
1170 FOR I=1 TO 9:IF V(I)=0 THEN 1200
1180 IF S$(I)=G THEN 1200
1190 PRINT F$(I) " seller:-F$(I)"$ per ton"
1200 NEXT
1210 FOR I=1 TO 9:IF V(I)=0 GOTO1240
1220 IF B$(I)<N$(G) GOTO1240

```

```

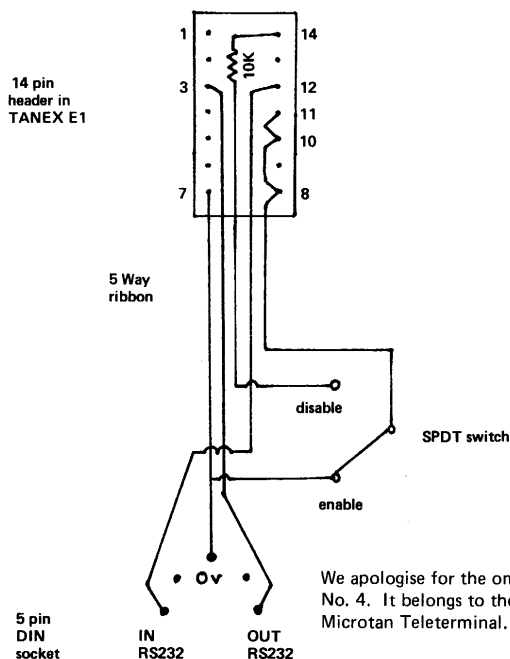
1230 PRINT F$(I) " buyer:-B$(I)"$ per ton"
1240 NEXT:GOTO330
1250 PRINT CHR$(12):TAB(5):"Cargo computer"
1260 PRINT "Cargo":TAB(25):"Tons"
1270 Y=0
1280 FOR I=1 TO 5:IF K(I)=0 GOTO1300
1290 PRINT F$(I):TAB(26):H(I)
1300 Y=Y+H(I):NEXT
1310 PRINT "Total":TAB(26):Y
1320 PRINT "Max cargo":TAB(26):H(I):TAB(26):Y
1330 PRINT "There is room for:TAB(26):Y-TAB(26):H(I):TAB(26):Y
1340 PRINT CHR$(12):PRINT:PRINT F$(N) "F$(N)" Trading Centre"
1350 PRINT "Buyer:-":PRINT B$(N):TAB(15):F$(N):"per ton"
1360 PRINT "Cash $:"C
1370 C=C+S
1380 FOR I=1 TO 5
1390 IF B$(N)=H$(I) THEN A=I:GOTO1410
1400 NEXT
1410 IF H(A)>0 THEN 1440
1420 PRINT "Sorry, you have no "B$(N)
1430 PRINT "on board":GOTO330
1440 PRINT H(A):"tons "B$(N)
1450 INPUT "How much do you want to sell":J
1460 B=FNE(E)
1470 IF E=0 THEN PRINT "Now be sensible":GOTO1450
1480 IF E=H(A) THEN PRINT "You don't have that much":GOTO1450
1490 PRINT "Fine, is that a deal ?"
1500 GOSUB2180
1510 IF A$="Y" THEN 1540
1520 IF A$="N" THEN 320
1530 GOTO1500
1540 PRINT "That means we give you $FNE(B(N)*E)"
1550 PRINT "Good bye, do call again"
1560 G=C+FNE(C(N)*E)
1570 H(A)=H(A)-G:GOSUB2410 :GOTO330
1580 PRINT CHR$(12):PRINT:PRINT F$(N) "S$(N)" Trading Centre"
1590 C$="0"
1600 C$="0"
1610 PRINT "Seller:-":PRINT S$(N):TAB(15):F$(N):"per ton"
1620 PRINT "Rocket fuel:TAB(15):F$(N):"per gallon"
1630 FOR I=1 TO 5:IF V(I)=0 THEN 1640
1640 PRINT "Cash $:"C
1650 PRINT "Cargo room":tons
1660 PRINT "Fuel":gallons
1670 PRINT "Capacity":gallons
1680 PRINT "1":S$(N) "2:Rocket fuel"
1690 PRINT:PRINT "What do you want to buy "
1700 GOSUB2180 :A=VAL(A$):IF A=0 OR A>5 THEN 1700
1710 FOR I=1 TO 5:IF S$(N)=H$(I) THEN D=I:GOTO1740
1720 NEXT
1730 PRINT "Buyer:-":PRINT B$(D):TAB(15):F$(D):"per ton"
1740 PRINT "Rocket fuel:TAB(15):F$(D):"per gallon"
1750 B=N:GOTO1760 :1900 :2030
1760 PRINT "Seller:-":S$(N)
1770 PRINT "Cash $:"C
1780 PRINT "COY prices"
1790 PRINT "Buyer:-":B$(N)
1800 INPUT "How much do you want "F$(E)=FNE(E)
1810 IF E=0 THEN PRINT "Very clever, do it again":GOTO1800
1820 IF E=V THEN PRINT "Your ship can't carry that much":GOTO1800
1830 IF (E=V)>C THEN PRINT "You don't have enough cash":GOTO1800
1840 PRINT "That will cost you $FNE(E(N)*E)"
1850 PRINT "Is that fair ?"
1860 GOSUB2180
1870 IF A$="Y" THEN 320
1880 IF A$="N" THEN 1860
1890 C=C+FNE(F$(N)*E):H(D)=H(D)+FNE(E):GOSUB2410 :GOTO330
1900 PRINT "Rocket fuel"
1910 PRINT "Cost:-$*B(N)

```

```

1920 PRINT "Capacity":H("gallons"
1930 PRINT "Fuel":F("gallons"
1940 INPUT "How much":F:E=FNE(E)
1950 IF E=0 THEN PRINT "Come on, grow up":GOTO1940
1960 IF (R(N)*E)>C THEN PRINT "You can't afford that much":GOTO1940
1970 IF E=(C-E) THEN PRINT "You tanks hold that much":GOTO1940
1980 PRINT "That will cost you $FNE(R(N)*E)"
1990 GOSUB2180
2000 IF A$="Y" THEN C=C-FNE(R(N)*E):F=F+FNE(E):GOTO320
2010 IF A$="N" THEN 320
2020 GOTO1990
2030 PRINT "Space ships":INPUT "What cargo tonnage "F:E=FNE(E)
2040 IF E=0 THEN PRINT "Grow up":GOTO330
2050 INPUT "What fuel capacity "A:A=FNE(A)
2060 IF A=0 THEN PRINT "Grow up":GOTO330
2070 PRINT CHR$(12):PRINT:PRINT "That will cost you":1000+A+1000+E*$
2080 IF 1000+A+1000+E<C THEN PRINT "You can't afford it":GOTO330
2090 PRINT "You get a 50% refund on your":PRINT "old ship"
2100 PRINT "Do you want to buy it ? "
2110 GOSUB2180
2120 IF A$="Y" THEN 320
2130 IF A$="N" THEN 2110
2140 C=C-(1000+A+1000+E):C=C+.5*(1000+H+1000+E):GOTO2150
2150 M=E:M=A:GOTO320
2160 RETURN
2170 REM CORRECT BUG IN GET COMMAND - USE 'GET A$' ON OTHER SYSTEMS
2180 POKE 1,0
2190 IF PEEK(1)=0 THEN 2190
2200 A$=CHR$(PEEK(1))
2210 RETURN
2220 PRINT CHR$(12):TAB(5):"Prices computer"
2230 PRINT "You are on "F$(N)
2240 RETURN
2250 PRINT CHR$(12):PRINT F$(N) "Scrap Co."
2260 PRINT "We pay $ 100 for any element"
2270 FOR I=1 TO 5:PRINT I,H$(I):NEXT
2280 PRINT "Which mineral "
2290 GOSUB2180 :A=VAL(A$):IF A=0 OR A>5 THEN 2290
2300 PRINT H$(A)
2310 IF H(A)>0 THEN 2330
2320 PRINT "You haven't got any":GOTO330
2330 INPUT "How much "F:E=FNE(E)
2340 IF E=0 THEN PRINT "Grow up":GOTO330
2350 IF E=H(A) THEN PRINT "You only have":H(A):"tons":GOTO2330
2360 PRINT "Is that a deal ?"
2370 GOSUB2180
2380 IF A$="Y" THEN H(A)=H(A)-E:C=C+100+E:GOTO330
2390 IF A$="N" THEN 320
2400 GOTO2370
2410 IF (F<H(N)) THEN RETURN
2420 IF C<E THEN GOTO2510
2430 F=F-E:PRINT "Scrap"
2440 IF F=0 THEN 2430
2450 P(N)=1
2460 FOR I=1 TO 9
2470 IF F=N THEN NEXT
2480 IF B$(N)=H$(I) THEN B(I)=B(I)+FNE(H(I))
2490 NEXT
2500 RETURN
2510 C=C+F(N)+FNE(H(I))
2520 FOR I=1 TO 9
2530 IF N=I THEN NEXT
2540 IF S$(I)=S$(N) THEN GOTO2570
2550 NEXT
2560 RETURN
2570 F=F-(I)-FNE(H(I))
2580 IF F=0 THEN 2570
2590 P(I)=1
2600 GOTO2550

```



We apologise for the omission of the above circuit from issue No. 4. It belongs to the article by Doug Deedman on the Microtan Teleterminal.

Advertisements

**FOR JUST
£29.95+p&p
YOU CAN PUT
IT IN PRINT!**

You can connect a Sinclair ZX Printer to your 6502 Computer by simply plugging the right PRINTFACE unit to your system giving you up to 43 characters line or 256 Dixel graphics.

For MICROTAN (ask for Printface 1)

For DRAGON 32 (ask for Printface 2)

For ATOM/ACORN rack (ask for Printface 3)

For ORIC (ask for Printface 4)

For B B C COMPUTER (ask for Printface 5)

For VIC 20 (ask for Printface 6)

HOW TO ORDER
Please enter your requirements in the coupon below, enclosing your cheque or postal order, made payable to: MICROTANIC COMPUTER SYSTEMS LTD. Please allow 14 days for delivery.

MICROTANIC COMPUTER SYSTEMS LTD.
16 UPLAND RD., DULWICH,
LONDON S E 22

TEL: (01) 693 1137

Post to: **MICROTANIC COMPUTER SYSTEMS LTD**
16 UPLAND ROAD · DULWICH · LONDON · S E 22

Please send me the following items: _____

I need 'INTERFACE' UNIT for (state your computer system) _____

I enclose my cheque/P.O. for £ _____

NAME _____

COMPANY _____ ADDRESS _____

TEL (DAY) _____ TEL' (EVE) _____

JOYSTICKS REVISITED

T.K. Jones
Sunderland

The enclosed circuit uses a dual 555 timer in single shot mode to provide pulses proportional to the position of an XY axis joystick. The idea stems from an article by J. Rew in the Tansoft Gazette (June/July 1982) but a 'recap' of the idea will be useful to those who missed it.

A 6522 is used to trigger the one shot via PC5, the control routine then increments the X & Y counters until a time out occurs, these values are then passed to the calling program

and act as a position indicator.

A basic program is attached which moves a 'line' about the high res screen, using Microtan's High Res eeprom. Removal of line 100 allows actual drawing on the screen.

There are a few constructional points to be noted in regards to the circuit. As a NE 556 was used, as opposed to two CMOS ICM755 devices, a separate power supply was used to prevent spikes appearing on the Microtan's lines. Secondly the value of the small timing capacitor can be altered to give the required count, as they stand a range of 1 to 250 is possible.

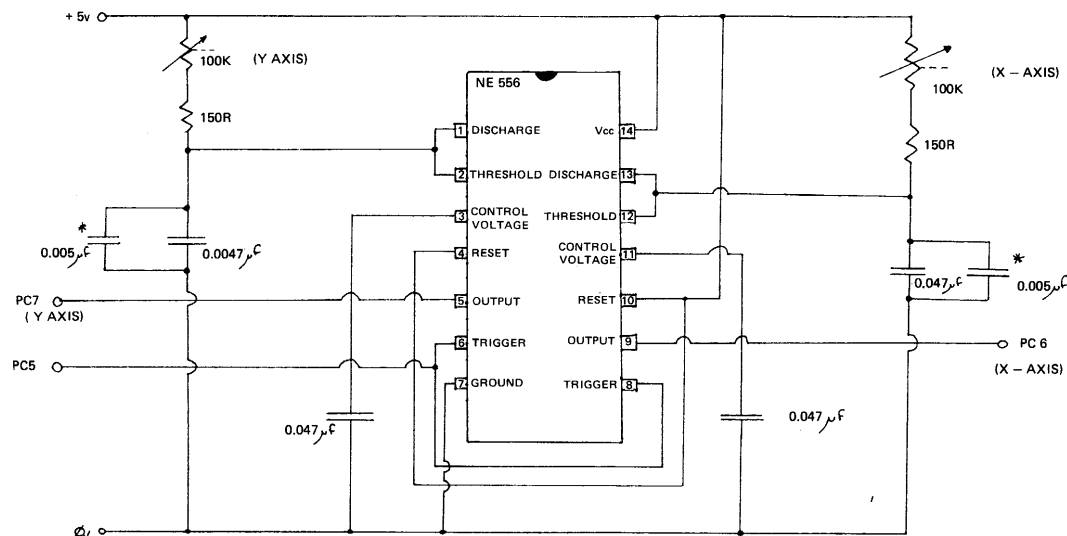


Fig. 1 DUAL AXIS JOYSTICK CIRCUIT

*See text

```

10 REM JOYSTICK TK JONES 16.8.83
15 REM SET UP SUBROUTINE
20 FOR I = 0 TO 44 : READ A : POKE 7936+I,A:NEXT I
30 DATA 169, 32, 141, 227, 191, 162, 0, 160,0,142,225
  191, 44, 225, 191, 16, 251
40 DATA 169, 32, 141, 225, 191, 44, 225, 191, 16, 1, 200, 80,
  1, 232, 173, 225, 191
50 DATA 41, 192, 208, 240, 142, 45, 31, 140, 46, 31, 96
60 REM MAIN PROGRAM
70 POKE 34, 0: POKE 35, 31
75 X = USR(1) : REM CALL JOY-STICK ROUTINE
80 A = PEEK (7981) : REM X
81 B = PEEK (7982) : REM Y
90 POKE 34, 0 : POKE 35, 232 : POKE 276, 119
95 P = USR(P) = REM CALL HIGH RES ROUTINES
100 ! 37,0,0 : ! 39,32,0 : REM CLEAR HIGH & LOW RES SCREENS
110 ! 5,A,B : GOTO 70 : REM PLOT ABSOLUTE X,Y GET NEXT VALUES
  
```

Listing of Basic Program to use Joystick

```

1F00 A9 20          LDA $20
1F02 8D E3 BF      STA $BFE3
1F05 A2 00         LDX $0
1F07 A0 0000       LDY $0
1F09 8E E1 BF      STX $BFE1
1F0C 2C E1 BF      BIT $BFE1
1F0F 10 FB         BPL $1F0C
1F11 A9 20         LDA $20
1F13 8D E1 BF      STA $BFE1
1F16 2C E1 BF      BIT $BFE1
1F19 10 01         BPL $1F1C
1F1B C8            INY
1F1C 50 01         BVC $1F1F
1F1E E8            INX
1F1F AD E1 BF      LDA $BFE1
1F22 29 C0         AND $C0
1F24 D0 F0         BNE $1F16
1F26 8E 20 1F      STX $1F2D
1F29 8C 2E 1F      STY $1F2E
1F2C 60            RTS
1F20               BYT X STORE
1F2E               BYT Y STORE
  
```

MACHINE CODE PRODUCED FROM BASIC PROGRAM

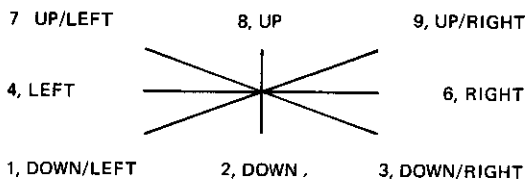
ETCH~A~SKETCH

By P J W Cooper

This program was written to demonstrate the capabilities of chunky graphics. It requires a minimum of 1K of RAM on Tanex plus an ASC11 Keyboard and, of course, the chunky graphics option.

The start of the program is at S400, and the first thing it does is clear the screen and put the two cursors on the screen. Cursor 1 flashes fast and cursor 2 flashes slow. (Don't panic - they are at the same position when the program is started, so you can only see Cursor 1).

Cursor 1 is moved by the numerical keypad, to the right of the keyboard, as follows:-



You can move it faster by using the Repeat key. Various Options now exist:-

Key	Function
D	Draws a (straight?) line between cursor 1 and cursor 2. The line can be moved, being erased and redrawn automatically.
E	Erases a line between cursor 1 and cursor 2. Can be used to clear a section of the display.
P	Plots a single point at position of cursor 1.
U	Unplots a single point at position of cursor 1.
M	Moves cursor 2 to position of cursor 1.
C	Clears the screen and returns cursors to bottom left position.

If you have more than 1K of RAM on Tanex, then a single screen of graphics can be saved (S key) for recall later (R Key).

Program Description

The program occupies 400 - 7A0 and the data (saved screen) occupies S7F0 - S9FF previously published subroutines are included at S600 and S740 to deal with line drawing and plotting respectively.

Zero page locations S3F - 4A are used for scratch pad purposes.

When the program is started, graphics are switched on and it jumps to S4A5 to clear the screen and initialise the cursors. It then goes to S465 which is the main keyboard/cursor flash loop. The delay is at 48E.

If a Key is detected, a check is made to determine if it is a command (C, R, S, D, E, M, P, U).

If not, then it assumes a cursor move, and a line is erased if necessary. The appropriate action is taken to alter the X and Y cursor co-ordinates before returning to redraw the line (if necessary) and jumping back to the keyboard loop.

Here are some main entry points:-

0400	Start of program
0403	Prepares plot routine for Cursor 1
040D	Prepares plot routine for Cursor 2
0417	Cursor 1 erase routine.
041F	Cursor 2 erase routine.
0428	Cursor 1 plot routine.
0433	Check if RAM present at \$800-BFF
043E	Save check
0446	Recall check
0465	Main loop - Keyboard/Cursor flash
048E	Delay loop
049D	Key detected (Process key)
04A5	Clear screen action
0505	Erase line action
052B	Draw line action
054E	Cursor control
05D1	Take Cursor 1 off screen
05DD	Take Cursor 2 off screen
0600	Line draw/erase routine
0679	Screen Save/Recall routine
0740	Graph plot routine

MAIN PROGRAM DIS-ASSEMBLED LISTING

NB: Teleprinter O/P

@ = \$
£ = ##

0400	4C3E05	JMP	0053E
0403	A201	LDX	£00001
0405	B545	LDA	00045,X
0407	9540	STA	00040,X
0409	CA	DEX	
040A	10F9	BPL	00405
040C	60	RTS	
040D	A201	LDX	£00001
040F	B547	LDA	00047,X
0411	9540	STA	00040,X
0413	CA	DEX	
0414	10F9	BPL	0040F
0416	60	RTS	
0417	200D04	JSR	0040D
041A	863F	STX	0003F
041C	204007	JSR	00740
041F	200304	JSR	00403
0422	863F	STX	0003F
0424	204007	JSR	00740
0427	60	RTS	
0428	200304	JSR	00403
042B	A201	LDX	£00001
042D	863F	STX	0003F
042F	204007	JSR	00740
0432	60	RTS	
0433	A9AA	LDA	£000AA
0435	8D000A	STA	00A00
0438	AD000A	LDA	00A00
043B	C9AA	CHP	£000AA
043D	60	RTS	
043E	203304	JSR	00433
0441	D022	BNE	00465
0443	4C7906	JMP	00679
0446	203304	JSR	00433
0449	D01A	BNE	00465
044B	4C9606	JMP	00696
044E	6504	ADC	00004
0450	00	BRK	
0451	00	BRK	
0452	00	BRK	
0453	00	BRK	
0454	00	BRK	
0455	00	BRK	
0456	00	BRK	
0457	00	BRK	
0458	00	BRK	
0459	A54A	LDA	0004A
045B	D002	BNE	0045F
045D	A9FF	LDA	£000FF
045F	853F	STA	0003F
0461	6B	PLA	
0462	854A	STA	0004A
0464	60	RTS	
0465	A900	LDA	£00000
0467	8501	STA	00001
0469	201704	JSR	00417
046C	208E04	JSR	0048E
046F	202804	JSR	00428
0472	208E04	JSR	0048E
0475	200D04	JSR	0040D
0478	A201	LDX	£00001
047A	863F	STX	0003F
047C	204007	JSR	00740
047F	201F04	JSR	0041F
0482	208E04	JSR	0048E
0485	202804	JSR	00428
0488	208E04	JSR	0048E
048B	4C6504	JMP	00465
048E	A020	LDY	£00020
0490	A200	LDX	£00000
0492	A501	LDA	00001
0494	D007	BNE	0049D
0496	CA	DEX	
0497	D0F9	BNE	00492
0499	88	DEY	
049A	D0F4	BNE	00490

```

049C 60 RTS
049D 68 PLA
049E 68 PLA
049F A5 01 LDA #0001
04A1 C943 CMP #00043
04A3 D0DF BNE #04B4
04A5 209407 JSR #0794
04A8 A206 LDX #00006
04AA A900 LDA #00000
04AC 9544 STA #00044,X
04AE CA DEX
04AF 10FB BPL #04AC
04B1 4C6504 JMP #0465
04B4 C952 CMP #00052
04B6 D0D3 BNE #04B8
04B8 4C4604 JMP #0446
04BB C953 CMP #00053
04BD D0D3 BNE #04C2
04BF 4C3E04 JMP #043E
04C2 C944 CMP #00044
04C4 D006 BNE #04C2
04C6 A201 LDX #00001
04C8 8644 STX #00044
04CA D027 BNE #04F3
04CC C945 CMP #00045
04CE D006 BNE #04D6
04D0 A2FF LDX #000FF
04D2 8644 STX #00044
04D4 D010 BNE #04F3
04D6 C940 CMP #00040
04D8 D016 BNE #04F0
04DA 20D005 JSR #05D0
04DD A545 LDA #00045
04DF 8547 STA #00047
04E1 A546 LDA #00046
04E3 8548 STA #00048
04E5 A549 LDA #00049
04E7 854A STA #0004A
04E9 A900 LDA #00000
04EB 854A STA #0004A
04ED 4C6504 JMP #0465
04F0 4C6C05 JMP #05BC
04F3 48 PHA
04F4 8A TAX
04F5 48 PHA
04F6 98 TYA
04F7 48 PHA
04F8 20D405 JSR #05D4
04FA A544 LDA #00044
04FD F013 BEQ #0512
04FF 20D005 JSR #05D0
0502 EA NOP
0503 EA NOP
0504 EA NOP
0505 204405 JSR #0544
0508 A2FF LDX #000FF
050A 863F STX #0003F
050C 200006 JSR #0600
050F 20D005 JSR #05D0
0512 68 PLA
0513 A8 TAX
0514 68 PLA
0515 AA TAX
0516 68 PLA
0517 204E05 JSR #054E
051A 20D405 JSR #05D4
051D A644 LDX #00044
051F CA DEX
0520 3019 BMI #053B
0522 20D405 JSR #05D4
0525 20D005 JSR #05D0
0528 EA NOP
0529 EA NOP
052A EA NOP
052B 204405 JSR #0544
052E A644 LDX #00044
0530 863F STX #0003F
0532 200006 JSR #0600
0535 20D405 JSR #05D4
0538 20D005 JSR #05D0
053B 4C6504 JMP #0465
053E ADF0BF LDA #05FF0
0541 4CA504 JMP #04A5
0544 A204 LDX #00004
0546 8544 LDA #00044,X
0548 953F STA #0003F,X
054A CA DEX
054B 10F9 BPL #0546
054D 60 RTS
054E C939 CMP #00039
0550 D007 BNE #0559
0552 20AC05 JSR #05AC
0555 209C05 JSR #059C
0558 60 RTS
0559 C938 CMP #00038
055B D004 BNE #0561
055D 20AC05 JSR #05AC
0560 60 RTS
0561 C937 CMP #00037
0563 D007 BNE #056C
0565 20AC05 JSR #05AC
0568 20A505 JSR #05A5
056B 60 RTS
056C C936 CMP #00036
056E D006 BNE #0576

```

```

057A 20A505 JSR #05A5
057D 60 RTS
057E C933 CMP #00033
0580 D007 BNE #0589
0582 209C05 JSR #059C
0585 20B505 JSR #05B5
0588 60 RTS
0589 C932 CMP #00032
058B D004 BNE #0591
058D 20B505 JSR #05B5
0590 60 RTS
0591 C931 CMP #00031
0593 D006 BNE #0598
0595 20A505 JSR #05A5
0598 20B505 JSR #05B5
059B 60 RTS
059C A545 LDA #00045
059E C93F CMP #0003F
05A0 F002 BEQ #05A4
05A2 E645 INC #00045
05A4 60 RTS
05A5 A545 LDA #00045
05A7 F002 BEQ #05A8
05A9 C645 DEC #00045
05AB 60 RTS
05AC A546 LDA #00046
05AE C93F CMP #0003F
05B0 F002 BEQ #05B4
05B2 E646 INC #00046
05B4 60 RTS
05B5 A546 LDA #00046
05B7 F002 BEQ #05BB
05B9 C646 DEC #00046
05BB 60 RTS
05BC C950 CMP #00050
05BE D007 BNE #05C7
05C0 A901 LDA #00001
05C2 8549 STA #00049
05C4 4C9904 JMP #04C9
05C7 C955 CMP #00055
05C9 D006 BNE #05D1
05CB A900 LDA #00000
05CD 8549 STA #00049
05CF F0F3 BEQ #05C4
05D1 4CF304 JMP #04F3
05D4 A000 LDY #00000
05D6 843F STY #0003F
05D8 20D304 JSR #04D3
05DB 3008 BMI #05E5
05DD A000 LDY #00000
05DF 843F STY #0003F
05E1 C8 INY
05E2 200004 JSR #040D
05E5 98 TYA
05E6 48 PHA
05E7 204C07 JSR #0740
05EA 68 PLA
05EB A8 TAX
05EC A63F LDX #0003F
05EE B94900 LDA #0049,Y
05F1 D002 BNE #05F5
05F3 A9FF LDA #000FF
05F5 853F STA #0003F
05F7 9649 STX #00049,Y
05F9 204C07 JSR #0740
05FC 60 RTS
05FD 00 BRK
05FE 00 BRK
05FF 00 BRK
0600 A9FF LDA #000FF
0602 48 PHA
0603 48 PHA
0604 A540 LDA #00040
0606 C542 CMP #00042
0608 3010 BMI #061A
060A A8 TAX
060B A542 LDA #00042
060D 8540 STA #00040
060F 98 TYA
0610 8542 STA #00042
0612 A443 LDY #00043
0614 A541 LDA #00041
0616 8441 STY #00041
0618 8543 STA #00043
061A A542 LDA #00042
061C C540 CMP #00040
061E F028 BEQ #0648
0620 48 PHA
0621 18 CLC
0622 6540 ADC #00040
0624 4A LSR A
0625 8542 STA #00042
0627 AA TAX
0628 E8 INX
0629 A543 LDA #00043
062B 48 PHA
062C A541 LDA #00041
062E C543 CMP #00043
0630 F011 BEQ #0643
0632 18 CLC
0633 6543 ADC #00043
0635 4A LSR A
0636 8543 STA #00043
0638 C541 CMP #00041
063A B004 BCS #0640
063C E643 INC #00043

```

```

063E 5003 BVC #0643
0640 18 CLC
0641 6901 ADC #00001
0643 A8 TAX
0644 8A TYA
0645 48 PHA
0646 98 TYA
0647 48 PHA
0648 4C1A06 JMP #061A
064B A541 LDA #00041
064D C543 CMP #00043
064F F00D BEQ #065E
0651 A542 LDA #00042
0653 48 PHA
0654 A543 LDA #00043
0656 48 PHA
0657 A540 LDX #00040
0659 A541 LDA #00041
065B 4C3206 JMP #0632
065E 8A TAX
065F 48 PHA
0660 204007 JSR #0740
0663 68 PLA
0664 AA TAX
0665 68 PLA
0666 8541 STA #00041
0668 68 PLA
0669 8540 STA #00040
066B C9FF CMP #000FF
066D D001 BNE #0670
066F 60 RTS
0670 68 PLA
0671 8543 STA #00043
0673 68 PLA
0674 8542 STA #00042
0676 4C1A06 JMP #061A
0679 A20B LDX #0000B
067B 853F LDA #0003F,X
067D 9DF007 STA #07F0,X
0680 CA DEX
0681 10F8 BPL #067B
0683 E8 INX
0684 B00003 LDA #0300,X
0687 9D0009 STA #0900,X
068A B00002 LDA #0200,X
068D 9D0008 STA #0800,X
0690 CA DEX
0691 D0F1 BNE #0684
0693 4C6504 JMP #0465
0696 A20B LDX #0000B
0698 BDF007 STA #07F0,X
069B 953F STA #0003F,X
069D CA DEX
069E 10F8 BPL #0698
06A0 E8 INX
06A1 B00009 LDA #0900,X
06A4 9D0003 STA #0300,X
06A7 B00008 LDA #0800,X
06AA 9D0002 STA #0200,X
06AD CA DEX
06AE D0F1 BNE #06A1
06B0 4C6504 JMP #0465
06B3 FF ?
06B4 FF ?
06B5 FF ?
06B6 FF ?
06B7 FF ?
06B8 FF ?
06B9 FF ?
06BA FF ?
06BB FF ?
06BC FF ?
06BD FF ?
06BE FF ?
06BF FF ?

```

PLOT ROUTINE DIS-ASSEMBLED LISTING

```

0740 ADF0BF LDA #05FF0
0743 A93F LDA #0003F
0745 C540 CMP #00040
0747 3040 BMI #0789
0749 EA NOP
074A 98 SEC
074B E541 SBC #00041
074D EA NOP
074E 48 PHA
074F 2903 AND #00003
0751 AA TAX
0752 68 PLA
0753 EA NOP
0754 293C AND #0003C
0756 0A ASL A
0757 0A ASL A
0758 0A ASL A
0759 8542 STA #00042
075B A902 LDA #00002
075D A503 STA #00043
075F 9002 BCC #0763
0761 E643 INC #00043
0763 A540 LDA #00040
0765 4A LSR A

```

```

0766 A8 TAX
0767 A901 LDA #00001
0769 9001 BCC #076C
076B 0A ASL A
076C CA DEX
076D 3004 BMI #0773
076F 0A ASL A
0770 0A ASL A
0771 D0F9 BNE #076C
0773 AA TAX
0774 A53F LDA #0003F
0776 3008 BMI #0780
0778 F010 BEQ #078A
077A 8A TAX
077B 1142 ORA #00042,Y
077D 9142 STA #00042,Y
077F 60 RTS
0780 8A TAX
0781 3142 AND #00042,Y
0783 F004 BEQ #0789
0785 5142 EOR #00042,Y
0787 9142 STA #00042,Y
0789 60 RTS
078A 8A TAX
078B 3142 AND #00042,Y
078D F004 BEQ #0793
078F A901 LDA #00001
0791 853F STA #0003F
0793 60 RTS
0794 A900 LDA #00000
0796 AA TAX
0797 9D0002 STA #0200,X
0799 9D0003 STA #0300,X
079D E8 INX
079E D0F7 BNE #0797
07A0 60 RTS

```

HEX DUMP OF MAIN PROGRAM OPTIONAL

```

0400 4C 3E 05 A2 01 B5 45 95
0408 40 CA 10 F9 60 A2 01 B5
0410 47 95 40 CA 10 F9 60 20
0418 0D 04 B6 3F 20 40 07 20
0420 03 04 B6 3F 20 40 07 20
0428 20 03 04 A2 01 B6 3F 20
0430 04 07 60 A9 AA 80 00 0A
0438 AD 00 0A C9 AA 60 20 33
0440 04 D0 22 4C 79 06 20 33
0448 04 D0 1A 4C 96 06 65 04
0450 00 00 00 00 00 00 00 00
0458 00 A5 4A D0 02 A9 FF 85
0460 3F 68 85 4A 60 A9 00 85
0468 01 20 17 04 20 8E 04 20
0470 28 04 20 8E 04 20 00 04
0478 A2 01 B6 3F 20 40 07 20
0480 1F 04 20 8E 04 20 28 04
0488 20 8E 04 4C 65 04 A0 20
0490 A2 00 A5 01 D0 07 CA D0
0498 F9 88 D0 F4 60 68 68 A5
04A0 01 C9 43 D0 0F 20 94 07
04A8 A2 06 A9 00 95 44 CA 10
04B0 FB 4C 65 04 C9 52 D0 03
04B8 AC 46 04 C9 53 D0 03 4C
04C0 3E 04 C9 44 D0 06 A2 01
04C8 86 44 D0 27 C9 45 D0 06
04D0 A2 FF 86 44 D0 10 C9 4D
04D8 D0 16 20 D0 05 A5 45 85
04E0 47 A5 46 85 48 A5 49 85
04E8 4A A9 00 85 44 4C 65 04
04F0 4C BC 05 48 8A 48 98 48
04F8 20 04 05 A5 44 F0 13 20
0500 D0 05 EA EA EA 20 44 05
0508 A2 FF 86 3F 20 00 06 20
0510 D0 05 68 A8 68 AA 68 20
0518 4E 05 20 04 05 A6 44 CA
0520 30 19 20 04 05 20 D0 05
0528 EA EA EA 20 44 05 A6 44
0530 86 3F 0A 00 06 20 04 05
0538 20 D0 05 4C 65 04 AD F0
0540 BF 4C A5 04 A2 04 B5 44
0548 95 3F CA 10 F9 60 C9 39
0550 D0 07 20 AC 05 20 9C 05
0558 60 C9 38 D0 04 20 AC 05
0560 60 C9 37 D0 07 20 AC 05
0568 20 A5 05 60 C9 36 D0 06
0570 20 9C 05 60 EA C9 34
0578 D0 04 20 A5 05 60 C9 33
0580 D0 07 20 9C 05 20 B5 05
0588 60 C9 32 D0 04 20 B5 05
0590 60 C9 31 D0 06 20 A5 05
0598 20 B5 05 60 A5 45 C9 3F
05A0 F0 02 E6 45 60 A5 45 F0
05A8 02 C6 45 60 A5 46 C9 3F
05B0 F0 02 E6 46 60 A5 46 F0
05B8 02 C6 46 60 C9 50 D0 07
05C0 A9 01 B5 49 4C E9 04 C9

```

```

0508 55 D0 06 A9 00 85 49 F0
0500 F3 4C F3 04 A0 00 84 3F
0508 20 03 04 30 08 A0 00 84
05E0 3F C8 20 00 04 98 48 20
05E8 40 07 68 A8 A5 3F B9 49
05F0 00 D0 02 A9 FF 85 3F 96
05F8 49 20 40 07 60 00 00 00
0500 A9 FF 48 48 A5 40 C5 42
0508 30 10 A8 A5 42 85 40 98
0510 85 42 A4 43 A5 41 84 41
0518 85 43 A5 42 C5 40 F0 2B
0520 48 18 65 40 4A 85 42 AA
0528 E8 A5 43 48 A5 41 C5 43
0530 F0 11 18 65 43 4A 85 43
0538 C5 41 B0 04 E6 43 50 03
0540 18 69 01 A8 8A 48 98 48
0548 4C 1A 06 A5 41 C5 43 F0
0550 0D A5 42 48 A5 43 48 A6
0558 40 A5 41 4C 32 06 8A 48
0560 20 40 07 68 AA 68 85 41
0568 68 85 40 C9 FF D0 01 60
0570 68 85 43 68 85 42 4C 1A
0578 06 A2 08 B5 3F 9D F0 07
0580 CA 10 F8 E8 BD 00 03 9D
0588 00 09 BD 00 02 9D 00 08
0590 CA D0 F1 4C 65 04 A2 08
0598 BD F0 07 95 3F CA 10 F8
05A0 E8 BD 00 09 9D 00 03 BD
05A8 00 08 9D 00 02 CA D0 F1
05B0 4C 65 04 FF FF FF FF FF
05B8 FF FF FF FF FF FF FF FF
05C0 FF FF FF FF FF FF FF FF
05C8 FF FF FF FF FF FF FF FF
05D0 FF FF FF FF FF FF FF FF
05D8 FF FF FF FF FF FF FF FF
05E0 FF FF FF FF FF FF FF FF
05E8 FF FF FF FF FF FF FF FF
05F0 FF FF FF FF FF FF FF FF
05F8 FF FF FF FF FF FF FF FF
0700 FF FF FF FF FF FF FF FF
0708 FF FF FF FF FF FF FF FF
0710 FF FF FF FF FF FF FF FF
0718 FF FF FF FF FF FF FF FF
0720 FF FF FF FF FF FF FF FF
0728 FF FF FF FF FF FF FF FF
0730 FF FF FF FF FF FF FF FF

```

```

0738 FF FF FF FF FF FF FF FF
0740 AD F0 BF A9 3F C5 40 30
0748 40 EA 38 E5 41 EA 48 29
0750 03 AA 68 EA 29 3C 0A 0A
0758 0A 85 42 A9 02 85 43 90
0760 02 E6 43 A5 40 4A A8 A9
0768 01 90 01 0A CA 30 04 0A
0770 0A D0 F9 AA A5 3F 30 08
0778 F0 10 8A 11 42 91 42 60
0780 8A 31 42 F0 04 51 42 91
0788 42 60 8A 31 42 F0 04 A9
0790 01 85 3F 60 A9 00 AA 9D
0798 00 02 9D 00 03 E8 00 F7
07A0 60 FF FF FF FF FF FF FF
07A8 FF FF FF FF FF FF FF FF
07B0 FF FF FF FF FF FF FF FF
07B8 FF FF FF FF FF FF FF FF
07C0 FF FF FF FF FF FF FF FF
07C8 FF FF FF FF FF FF FF FF
07D0 FF FF FF FF FF FF FF FF
07D8 FF FF FF FF FF FF FF FF
07E0 FF FF FF FF FF FF FF FF
07E8 FF FF FF FF FF FF FF FF
07F0 01 00 00 E0 03 00 00 00
07F8 00 00 00 00 FF FF FF FF

```

```

0880 00 00 00 00 00 00 00 00
0888 00 00 00 00 00 00 00 00
0890 00 00 20 E0 04 00 08 58
0898 00 00 00 00 00 00 00 00
08A0 00 00 00 00 00 00 00 00
08A8 00 00 00 00 00 00 00 00
08B0 00 00 00 AA 00 00 A0 05
08B8 00 00 00 00 00 00 00 00
08C0 00 00 00 00 00 00 00 00
08C8 00 00 00 00 00 00 00 00
08D0 00 00 00 00 00 90 02 54
08D8 00 00 00 00 00 00 00 00
08E0 00 00 00 00 00 A0 7C 3F
08E8 3F FF FF FC FC FC FC 7C
08F0 0C 24 30 90 C0 C0 99 7E
08F8 FC 41 00 00 00 00 00 00
0900 00 00 00 00 00 55 00 E8
0908 00 00 03 08 2F 3F FF 55
0910 00 00 80 00 00 F0 78 00
0918 C6 BF D0 00 00 00 00 00
0920 00 00 00 00 00 A9 40 AF
0928 D4 00 00 00 00 00 00 5D
0930 00 00 2A F0 02 03 28 5C
0938 32 99 28 6D 00 00 00 00
0940 00 00 00 00 00 00 FF F4
0948 E8 F4 D0 D8 FC FC FC FF
0950 D0 C0 C0 72 90 C0 18 01
0958 00 00 03 00 00 00 00 00
0960 00 00 00 00 80 FF 7F
0968 AF FF 07 00 03 03 08 03
0970 AF FF FF 15 00 00 00 00
0978 00 00 00 00 00 00 00 00
0980 00 00 00 00 00 0A FF 41
0988 0A FF 00 00 00 00 00 00
0990 00 BF FF 00 00 00 00 00
0998 00 00 00 00 00 00 00 00
09A0 00 00 00 00 00 AA 55
09A8 00 2B F4 00 00 00 00 00
09B0 00 AA FF 50 00 00 00 00
09B8 00 00 00 00 00 00 00 00
09C0 00 00 00 00 00 02 F2
09C8 04 00 00 00 00 00 00 00
09D0 00 02 3D 10 00 00 00 00
09D8 00 00 00 00 00 00 00 00
09E0 40 00 00 00 00 00 00 00
09E8 00 00 00 00 00 00 00 00
09F0 00 00 00 00 00 00 00 00
09F8 00 00 00 00 00 00 00 00

```

HEX DUMP DATA (OPTIONAL)

```

0800 00 00 00 00 00 00 00 00
0808 00 00 00 00 00 00 00 00
0810 00 00 00 00 00 00 00 00
0818 00 00 00 00 00 00 00 00
0820 00 00 00 00 00 00 00 00
0828 00 00 00 00 00 00 00 00
0830 00 00 00 00 00 00 00 00
0838 00 00 00 00 00 00 00 00
0840 00 00 00 00 00 00 00 00
0848 00 00 00 00 00 00 00 00
0850 00 00 00 00 00 00 00 00
0858 00 00 00 00 00 00 00 00
0860 00 00 00 00 00 00 00 00
0868 00 00 00 00 00 00 00 00
0870 00 00 00 00 00 00 00 00
0878 00 00 00 00 00 00 00 00

```

FOOLPROOF INPUT

By T K Jones

LISTING : FOOL-PROOF INPUT ROUTINE

```

63992 KEYS = " " : REM INPUT ROUTINE
63993 GETK$ : IF ASC(K$) = 127 AND KEY $ = " " THEN PRINT : GOTO 63998
63994 IF ASC(K$) = 13 THEN 63998
63995 IF ASC(K$) = 127 THEN KEY $ = LEFT $ (KEY$,LEN (KEY$)-1) : GOTO 63993
63996 IF ASC(K$) 48 OR ASC(K$) 57 THEN PRINT
CHR$(127); : GOTO 63993 : REM NUMERICAL I/O
63997 KEYS = KEYS + K$ : GOTO 63993
63998 IF KEYS = " " THEN PRINT "Pardon?"; : GOTO 63993
63999 PRINT : RETURN

```

NOTE,

With 63996 as it stands only numerical, positive input is allowed - useful for menus.

To allow the input of upper and lower case alphabetic character only, alter to:-

```

63996 IF ASC(K$) 60 OR ASC(K$) 122 then
PRINT CHR$(127); : GOTO 63993 : REM ALPHABETIC
TO ALLOW BOTH DELETE 63996
LISTING : EXAMPLE OF CALL ROUTINE

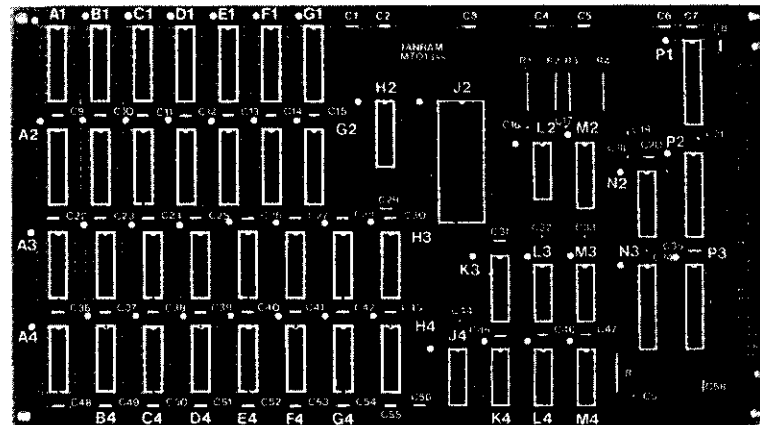
```

```

10 PRINT "GIVE a number between 0 and 9"; : GOSUB
63992 : ANS = VAL (KEY$) : IF ANS 9 THEN 10

```

TANRAM KIT OF PARTS



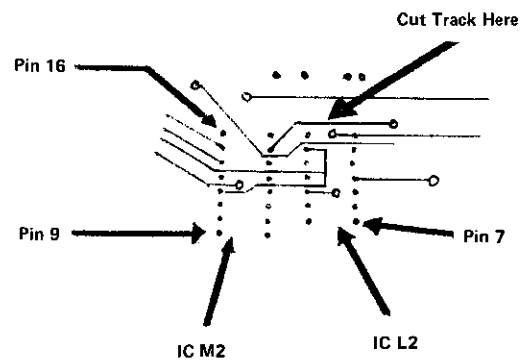
DATA CORRUPTION ON TANRAM

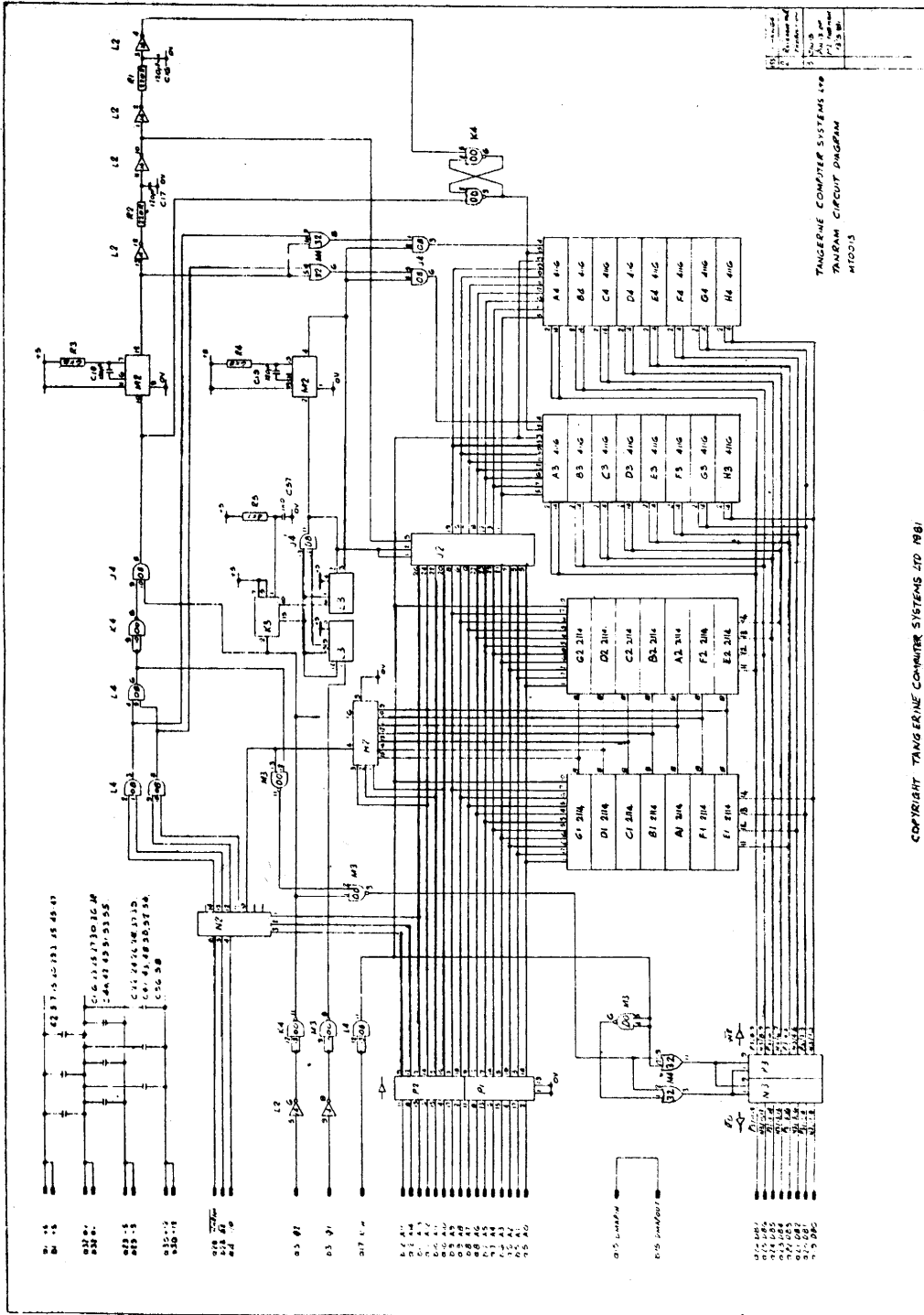
Some Tanram boards are prone to random corruption due to a small timing problem. The problem can be confirmed using the following program:

```
5 FOR I = 1 TO 16: ? : NEXT
10 B$ = 'ANYONE FOR TENNIS?'
20 A = 1: B = 30: C = 0
30 A$ = RIGHTS(B$,A) + LEFTS(B$,B)
40 B$ = A$
50 GOSUB 300
60 C = C+1
70 IF C = 12 THEN D=A: A=B: B=D: C=0
80 GOTO 30
300 POKE 3,0: POKE 48,0
310 ? A$:
320 FOR D1=0 TO 50: NEXT D1
330 RETURN
```

Run the program for some time and if there is any change at all in the statement ANYONE FOR TENNIS your Tanram has a timing error.

To cure the fault, cut the track between IC M2 pin 2 and where it goes to the plated through hole. Bridge the cut with a 470 ohm resistor. Next connect a 330 pf capacitor from pin 2 of IC M2 and the nearest convenient ground. (Pin 1 of IC N2).





COPYRIGHT TANGERINE COMPUTER SYSTEMS LTD 1981

TANRAM

THE THREE PIECE SUITE

Graham Davies

OVARTURE

As the title suggests, the Three Piece Suite (TPS) consists of three programs. OVAR, creates a file on disc; HULTON, retrieves such a file from disc whilst BOLTAN outputs a file, resident in ram, to a printer. Actually, for reasons to be explained, there are two 'BOLTAN' programs, BLTANS for serial printers (I have a TTY) and BLTANP for parallel printers.

The development of these programs derived from LIST, a program by Paul Robinson published in Tansoft Gazette No. 7. So, because my suite contains algorithms similar to those in LIST, I would like to thank Paul Kaufman, Managing Director of Tansoft Ltd., for permission to delve into and use or adapt algorithms from LIST as necessary.

The suite does work and has been tested but it must be remembered however, that the programs are experimental. They were written so that cassette dependent software could be given a disc file-handling capability. I'm now hoping that some bright user out there will show us how to append these programs to an existing package.

DOS CANNOT FILE BY READ ALONE

Having admitted a connection between LIST and the T.P.S. I hasten to add that OVAR and HULTAN are radically different from Paul's program. LIST is essentially limited to the Tandos routines related to 'read-from-disc' and then, only for the direct transfer of a file from disc to printer! On the other hand OVAR and HULTON support the transfer of file text between ram and disc.

WRITE FOR DOSTERITY

Before embarking on a description of the contents and operation of the T.P.S. a discussion of the mechanics of disc file creation, as I understand them, would be useful. Actually, I do hope that one of the Tanelite will augment my efforts.

To get onto the learning curve let us set ourselves the task of creating an imaginary file to be called LES. It consists of D257 (H103) bytes of text resident in ram with a start address of H5000. To comply with the convention given in LIST a D278'th byte must somehow be added by the user and is the 'end-of-text-marker' (Z) which has the hexadecimal value of H1A.

Stepping up the learning curve consists of realising that a file consisting of more than D255 (HFF) bytes of text cannot be loaded in one go. The text must be broken up into 'RECORDS' with each RECORD able to hold up to D255 (HFF) bytes of text.

Before doing the division sum (or as you'll see later - subtraction sum) for LES and getting an answer of TWO records, be advised of another convention created by LIST. The end-of-text-marker is given a record all of its own which I call - the LAST record.

If you have crampons handy for a steep section of the curve, please be made aware of another fact, this time dictated by Tandos! The record transferred to disc actually holds up to D256 (H100) bytes of data; the FIRST byte of the record holds a numerical value giving the number of bytes of text actually present in that record.

Returning to LES we see that there are THREE records. The initial record holds the first D255 bytes of the file with the number D255 resident in the first byte of that record. The second record holds the final FOUR bytes of file, text with the first byte of this record holding the number '4'. I call this the PENULTIMATE record. Finally, the LAST record consists of two bytes; the first byte contains the number one and the second byte contains H1A!

Before transfer to disc each record has to be loaded into a BUFFER and the number of records to be so created, is determined by subtracting HFF from the text still to be transferred. When the answer for the subtraction is negative, end of text is nigh and transfer of the penultimate record is imminent. In OVAR the user-routine TRNSFW fills the write-record buffer and the Tandos routine RECOU places the record on disc.

There are actually two buffers. One is called the sector buffer and is, thankfully, for the 'eyes' of Tandos alone. The other is the RECORD BUFFER (in this case WRITE). Ram space for both must be provided by the user and the buffer addresses are pointed to by the Tandos ram spaces WBUF and WREC. Only the high bytes need be given. The low bytes of the addresses are 00 by default.

A sixteen bit division routine was considered for calculating the number of records but it seemed so complex compared with continuous subtraction. Perhaps the high-flyers out there will agree with the maxims: do not place thy faith in algorithms simply because they work - understanding the algorithm hath great merit.

THE PROGRAM OVAR

The source code was created using the Microtanic Two Pass Assembler. Although it is well documented, an explanation of the program is given.

Starting with a description of the zero page locations, we have

STRTLO/HI define the start locations of the file in ram and are stored for possible later use

CRNTLO/HI are used when counting through the file-text bytes when 'filling' successive record buffers.

ENDATL/H give the end address of text in ram. It is determined when the routine ENDAT1 . . . finds the

end-of-text marker. It is also stored for further use if needed by the user.

LNDATL/H gives the actual number of bytes of text in ram up to the mark H1A.

DTLFTL/H is the number of text bytes still to be transferred after a pass through TRNSFW. On first entering this routine Dataleft = Length of data.

YCOUNT stores the content of the Y register when RECOU is called.

PENFLG identifies the penultimate record. Without it TRNSFW would go into an endless loop.

No doubt one could use fewer locations, but they were all useful when debugging OVAR.

The Tandos locations and routines must now be mentioned, but for a fuller description the manual should be consulted. Incidentally, the routine addresses given in the TPS are correct. Those in the manual are wrong.

GETFIL will be described later but it should be mentioned that it prefers to be called at the start of the program. Experimentalists may care to see how far into the program it may be pushed before it doesn't function. My own frustrating experience indicates; not too far.

IN110 is a routine to set up disc flags and that all I know!

OPENNEW; before the text can be transferred to disc a file must be opened for write. OPENW does this and GETFIL is used to provide the file with a name.

CLOSEW; when all the records have been transferred, CLOSEW does all that is necessary to make that file a 'permanent' feature of the disc. If CLOSEW is not executed the file will not exist!

RECOUT actually transfers a record from the record buffer to disc. It is executed every time the record buffer is filled.

FNAMO is a set of ram locations into which the Filename procured by GETFIL is placed. The routine CHKNAM illustrates how the Filename may be retrieved from FNAMO.

ERREX is Tandos ram space used to point to a user routine related to error messages.

WBUF is a byte in Tandos ram space used to point to the address of the user-assigned ram space for the sector buffer.

WREC is similar to WBUF but points to the record buffer. In both cases only the high byte of the buffer address is stored. The low byte is always 00.

Let us now imagine that the source code has been entered and assembled with the object code being saved to disc using the auto-load facility:

DSAVE OVAR T400

Let us also imagine that LES has been patiently waiting in ram to be filed by OVAR. To do this, type in the line:

OVAR O:LES CR

and OVAR accompanied by whirring goes into action.

GETFIL reads the command line from the location, immediately after OVAR so that on execution of CLOSEW, towards the end of OVAR, LES will be filed in drive O and the filename LES recorded in the directory. The text is interrogated for H1A so that the number of data bytes may be computed and the filename LES is repeated to screen (this is in fact a left-over from debugging that could be useful to the user).

Next, the two buffers are pointed to using an interesting feature of the assembler and the EXIT/ERROR routines are pointed to in the same way.

Following the call for Flags (JSR IN110), file creation commences with the opening of a file for write (JSR OPENW) and after setting up the 'DaTaLeFT' locations TRNSFW swings into action. HFF is subtracted from DTLFTL and the DTLFTH is decremented. A positive answer to this means that text remains for transfer. Consequently the first record buffer is filled using the routine FUL1RECORD; with transfer to disc being implemented by RECOUNT. With only four bytes of text left the next pass through TRNSFW must produce a negative answer for DEC DTLFTH and the record buffer is filled using PENultimateRECORD. At this point PENultimateFLaG is set so that on leaving FILL1/FILL2 the LAST record containing H1A is transferred.

Finally, CLOSEW must be executed and a little tidying up done before returning to monitor.

You may now type in DIR to check if the file created by OVAR has its filename in the directory.

The content of the file may be retrieved from disc using HULTAN and a BOLTAN print routine used to output the file to printer. Alternatively, LIST may be used, but the file will not appear in ram. For test purposes the original file LES could be a length of text created using one of the word processor packages. When this is DLOADED be sure to use the S parameter so that the text starts at H5000 and then insert H1A (Z) to give at least one full record of text. Too much text may hamper checking of the DLOADED LES with the file LES.

HULTAN

With the mechanics of OVAR understood, it is felt that the source code documentation for HULTAN is quite adequate as a description of HULTAN. However, a few points should be explained.

HULTAN was developed before OVAR because HULTAN was the logical progression from LIST and could be debugged using simple files created in BASIC. With HULTAN operational OVAR was created.

Whilst HULTAN can be used to retrieve any old file from disc, meaningful retrieval only occurs with files created by OVAR and HULTAN alternately on LES. Following the example of LIST, a simple file created by BASIC is given as a last program.

BOLTAN

The simple printer control routines given in LIST and the V2.3 manual work with my FX80 but not with my Teletypewriter. A CR is not issued at the end of a line.

A routine was needed to count the required number of characters per line at which point a CR was output. Because of this, two BOLTAN's were compiled. The first for parallel printers and the second for serial printers. Or should I really say; for my TTY. It's unlikely that there'll be room for both BLTANP and BLTANS in this article, so I'll publish the latter. To obtain parallel operation simply place JSR OUTALL in line 83 and JMP NXTXT in line 84. Alternatively, with the exception of line 91, erase lines 83 through 100 and do not forget to use PARON/OFF instead of SERON/OFF! For files created in BASIC do the modification but keep SERON/SEROFF because a CR is present at the end of each record if you use a teletypewriter.

OVERT I HOPE

Throughout this article, I have tried to avert the criticism so often directed at articles and manuals that the given information is implicit rather than explicit. I trust that users will be able to directly load either the source or object code of all three programs and be up and running with a minimum of frustration. If worried then use the 'phone to ring the friendly local user group. Between us we seem to manage to sort out most faults whether hardware or software, so please do make contact especially if you're in the North West.

Graham Davies
15 Newland Drive
Over Hulton, BOLTON, Lancs.
Bolton (0204) 654145

You now know that I've followed John Dawson's concept of using novel titles for programs and from whence they came!

0001 ;	'DVAR'	0400	0090	STA LNDATH	043A 85 57
0002 ;		0400	0091	STA DTLFTH	043C 85 59
0003 ;	copyright G.G.Davies	0400	0092 ;		043E
0004 ;		0400	0093 ;		043E
0005 ;		0400	0094 ;		043E
0006 ;	TO TRANSFER FILE DATA	0400	0095	JSR OUTRET	043E 20 0C F8
0007 ;	FROM RAM TO DISC USING	0400	0096	JSR JCUSF	0441 20 29 F8
0008 ;	TANDOS I/O ROUTINES	0400	0097 ;		0444
0009 ;		0400	0098 ;	OUTPUT FILENAME BEING	0444
0010 ;		0400	0099 ;	CREATED ONTO BOTTOM	0444
0011 STRTLO EPZ #50		0400	0100 ;	LINE OF SCREEN	0444
0012 STRTHI EPZ #51		0400	0101 ;		0444
0013 CRNTLO EPZ #52		0400	0102 CHKNAM LDY #0		0444 A0 00
0014 CRNTHI EPZ #53		0400	0103 FILNAM LDA FNAMO,Y		0446 B9 2E B9
0015 ENDATL EPZ #54		0400	0104 CMP #'		0449 C9 20
0016 ENDATH EPZ #55		0400	0105 BEQ CARET		044B F0 08
0017 LNDATL EPZ #56		0400	0106 JSR OUTALL		044D 20 0E F8
0018 LNDATH EPZ #57		0400	0107 INY		0450 C8
0019 DTLFTL EPZ #58		0400	0108 CPY ##9		0451 C0 09
0020 DTLFTH EPZ #59		0400	0109 BNE FILNAM		0453 D0 F1
0021 PENFLG EPZ #5A		0400	0110 CARET NOP		0455 EA
0022 YCOUNT EPZ #5B		0400	0111 ;		0456
0023 ;		0400	0112 ;	SET UP SECTOR BUFFER	0456
0024 ;	TANDOS ADDRESSES	0400	0113 ;	AND RECORD BUFFER	0456
0025 ;		0400	0114 ;	ADDRESSES FOR WRITE	0456
0026 GETFIL EQU #B7F7		0400	0115 ;		0456
0027 INIID EQU #A806		0400	0116 ;		0456
0028 OPENW EQU #A80C		0400	0117 ;		0456
0029 CLOSEW EQU #A812		0400	0118	LDA #WSCBUF>	0456 A9 0A
0030 RECDUT EQU #A818		0400	0119	STA WBUF	0458 8D FC BB
0031 FNAMO EQU #B92E		0400	0120	LDA #WRCBUF>	045B A9 0B
0032 ERREX EQU #B94F		0400	0121	STA WREC	045D 8D FD BB
0033 WBUF EQU #BBFC		0400	0122 ;		0460
0034 WREC EQU #BBFD		0400	0123 ;		0460
0035 ;		0400	0124 ;	SET UP ERROR RETURN	0460
0036 ;		0400	0125 ;	ADDRESSES	0460
0037 ;	TANBUG ADDRESSES	0400	0126 ;		0460
0038 ;		0400	0127	LDA #EXITW	0460 A9 C9
0039 OUTRET EQU #F80C		0400	0128	STA ERREX	0462 8D 4F B9
0040 OUTALL EQU #F80E		0400	0129	LDA #EXITW>	0465 A9 04
0041 RETMON EQU #F823		0400	0130	STA ERREX+1	0467 8D 50 B9
0042 JCUSF EQU #F829		0400	0131 ;		046A
0043 ;		0400	0132 ;	*****	046A
0044 ;		0400	0133 ;		046A
0045 START JSR GETFIL		0400 20 F7 B7	0134 ;	NOW CREATE THE FILE	046A
0046 ;		0403	0135 ;		046A
0047 ;	INITIALISE START	0403	0136 ;	*****	046A
0048 ;	AND END ADDRESSES	0403	0137 ;		046A
0049 ;	TOGETHER WITH	0403	0138 ;		046A
0050 ;	PENFLG AND YCOUNT	0403	0139 ;	INITIALISE RECORD I/O	046A
0051 ;		0403	0140 ;		046A
0052	LDA #*00	0403 A9 00	0141	JSR INIID	046A 20 06 AB
0053	STA STRTLO	0405 85 50	0142 ;		046D
0054	STA ENDATL	0407 85 54	0143 ;	OPEN THE FILE FOR WRITE	046D
0055	STA CRNTLO	0409 85 52	0144 ;		046D
0056	STA PENFLG	040B 85 5A	0145	JSR OPENW	046D 20 0C AB
0057	STA YCOUNT	040D 85 58	0146 ;		0470
0058	LDA #*50	040F A9 50	0147 ;	SET UP 'LENGTH OF TEXT	0470
0059	STA STRTHI	0411 85 51	0148 ;	REMAINING IN RAM'	0470
0060	STA ENDATH	0413 85 55	0149 ;		0470
0061	STA CRNTHI	0415 85 53	0150	LDA LNDATH	0470 A5 57
0062 ;		0417	0151	STA DTLFTH	0472 85 59
0063 ;	DETERMINE LOCATION OF	0417	0152	LDA LNDATL	0474 A5 56
0064 ;	END-OF-TEXT	0417	0153	STA DTLFTL	0476 85 58
0065 ;		0417	0154 ;		0478
0066 ENDAT1 LDY #0		0417 A0 00	0155 ;		0478
0067 ENDAT2 LDA (ENDATL),Y		0419 B1 54	0156 ;	SINCE 'ONE' RECORD	0478
0068	CMP #*1A	041B C9 1A	0157 ;	CAN HOLD UP TO \$FF	0478
0069	BEQ RECORD	041D F0 0A	0158 ;	BYTES OF TEXT, CONTINUE	0478
0070	CPY #*FF	041F C0 FF	0159 ;	TO SUBTRACT \$FF FROM	0478
0071	BNE CONT1	0421 D0 02	0160 ;	DATA LEFT LOCATIONS	0478
0072	INC ENDATH	0423 E6 55	0161 ;	UNTIL ALL RECORDS,	0478
0073 CONT1 INY		0425 C8	0162 ;	ARE ON DISC	0478
0074	JMP ENDAT2	0426 4C 19 04	0163 ;		0478
0075 RECORD DEY		0429 88	0164	TRNSFW LDY YCOUNT	0478 A4 5B
0076	STY ENDATL	042A 84 54	0165	RCSTRT LDX #0	047A A2 00
0077 ;		042C	0166	SEC	047C 38
0078 ;	NOW COMPUTE THE NUMBER	042C	0167	LDA DTLFTL	047D A5 58
0079 ;	OF DATA BYTES IN TEXT	042C	0168	SBC #*FF	047F E9 FF
0080 ;		042C	0169	STA DTLFTL	0481 85 58
0081 ;		042C	0170	DEC DTLFTH	0483 C6 59
0082 LENDT1 CLD		042C D8	0171	BMI PENREC	0485 30 03
0083	SEC	042D 38	0172	JMP FULREC	0487 4C 99 04
0084	LDA ENDATL	042E A5 54	0173	PENREC LDA #1	048A A9 01
0085	SBC STRTLO	0430 E5 50	0174	STA PENFLG	048C 85 5A
0086	STA LNDATL	0432 85 56	0175	CLC	048E 18
0087	STA DTLFTL	0434 85 58	0176	ADC #*FF	048F 69 FF
0088	LDA ENDATH	0436 A5 55	0177	LDA DTLFTL	0491 A5 58
0089	SBC STRTHI	0438 E5 51	0178	STA WRCBUF	0493 8D 00 0B

0179	JMP FILL1	0496 4C 9E 04	0043 ; TANBUG ADDRESSES	0400
0180 FULREC	LDA #FFF	0499 A9 FF	0044 ;	0400
0181	STA WRBUF	049B 8D 00 0B	0045 OUTRET EQU \$F80C	0400
0182 FILL1	INX	049E E8	0046 OUTALL EQU \$F80E	0400
0183	LDA (CRNTLO),Y	049F B1 52	0047 RETMON EQU \$F823	0400
0184	STA WRBUF,X	04A1 9D 00 0B	0048 JCURSF EQU \$F829	0400
0185	CPY #FFF	04A4 C0 FF	0049 ;	0400
0186	BNE FILL2	04A6 D0 02	0050 ;	0400
0187	INC CRNTHI	04A8 E6 53	0051 HULTAN JSR GETFIL	0400 20 F7 B7
0188 FILL2	INY	04AA C8	0052 ;	0403
0189	CPX WRBUF	04AB EC 00 0B	0053 ; NOW PROVIDE SECTOR AND	0403
0190	BNE FILL1	04AE D0 EE	0054 ; RECORD BUFFERS;	0403
0191	STY YCOUNT	04B0 84 5B	0055 ; LOW BYTE ALWAYS \$00	0403
0192	JSR REOUT	04B2 20 18 AB	0056 ;	0403
0193	LDA PENFLG	04B5 A5 5A	0057 LDA #SECBUF>	0403 A9 07
0194	BNE LASREC	04B7 D0 03	0058 STA RBUF	0405 BD FE BB
0195	JMP TRNSFW	04B9 4C 78 04	0059 LDA #RECBUF>	0408 A9 08
0196 ;		04BC	0060 STA RREC	040A BD FF BB
0197 ;		04BC	0061 ;	040D
0198 ;		04BC	0062 ; INITIALISE RAM START	040D
0199 ; NOW ADD RECOMMENDED		04BC	0063 ; AND CURRENT LOCATION	040D
0200 ; END-OF-FILE INDICATOR		04BC	0064 ; ADDRESSES TOGETHER WITH	040D
0201 ; AT END OF TEXT		04BC	0065 ; YCOUNT	040D
0202 ;		04BC	0066 ;	040D
0203 LASREC LDA #1		04BC A9 01	0067 LDA #\$00	040D A9 00
0204	STA WRBUF	04BE 8D 00 0B	0068 STA RMSTLO	040F 85 50
0205	LDA #1A	04C1 A9 1A	0069 STA CRNTLO	0411 85 52
0206	STA WRBUF+1	04C3 8D 01 0B	0070 STA YCOUNT	0413 85 5B
0207	JSR REOUT	04C6 20 18 AB	0071 LDA #70	0415 A9 70
0208 ;		04C9	0072 STA RMSTHI	0417 85 51
0209 ;		04C9	0073 STA CRNTHI	0419 85 53
0210 EXITW JSR CLOSEW		04C9 20 12 AB	0074 ;	041B
0211	JSR OUTRET	04CC 20 0C F8	0075 ; THE RAM START ADDRESS	041B
0212	JSR JCURSF	04CF 20 29 F8	0076 ; COULD BE USEFUL IF THE	041B
0213	JMP RETMON	04D2 4C 23 F8	0077 ; PRINT ROUTINE 'BLTANX'	041B
0214 ;		04D5	0078 ; IS APPENDED TO 'HULTAN'	041B
0215 ;		04D5	0079 ;	041B
0216	ORG #A00	04D5	0080 ; SET UP ERROR RETURN	041B
0217 WSCBUF BYT 0		0A00 00	0081 ; ADDRESSES	041B
0218	ORG #B00	0A01	0082 ;	041B
0219 WRBUF BYT 0		0B00 00	0083 LDA #EXIT	041B A9 50
0220 ;		0B01	0084 STA ERREX	041D BD 4F B9
0221 ; END OF PROGRAM		0B01	0085 LDA #EXIT>	0420 A9 04
0222 ; ** OVAR **		0B01	0086 STA ERREX+1	0422 BD 50 B9
			0087 ;	0425
0001 ; *** HULTAN ***		0400	0088 ; INITIALISE DISC I/O	0425
0002 ; copyright G.G. David.		0400	0089 ; FLAGS	0425
0003 ;		0400	0090 ;	0425
0004 ; A PROGRAM TO TRANSFER		0400	0091 JSR INIID	0425 20 06 AB
0005 ; A DISC FILE TO RAM		0400	0092 ;	042B
0006 ;		0400	0093 ; OPEN A FILE FOR READ	042B
0007 ; ANY FILE WILL TRANSFER		0400	0094 ;	042B
0008 ; BUT MEANINGFUL TRANSFER		0400	0095 JSR OPENR	042B 20 09 AB
0009 ; REQUIRES A FILE CREATED		0400	0096 ;	042B
0010 ; WITH "OPENW/OPENW"		0400	0097 ; NOW INPUT A RECORD FROM	042B
0011 ; RELATED PROCEDURES		0400	0098 ; DISC TO RECORD BUFFER	042B
0012 ; BASIC CAN CREATE SUCH		0400	0099 ;	042B
0013 ; A FILE!		0400	0100 RECRDA JSR RECIN	042B 20 15 AB
0014 ;		0400	0101 LDA RECBUF+1	042E AD 01 0B
0015 ; SUGGESTED RAM START IS		0400	0102 CMP #FILEND	0431 C9 1A
0016 ; \$7000		0400	0103 BEQ EXIT	0433 F0 1B
0017 ;		0400	0104 ;	0435
0018 ; DATA BYTES AND ZERO		0400	0105 LDY YCOUNT	0435 A4 5B
0019 ; PAGE ADDRESSES		0400	0106 LDX #0	0437 A2 00
0020 ;		0400	0107 ;	0439
0021 FILEND EPZ #1A		0400	0108 ; AND TRANSFER THE RECORD	0439
0022 ;		0400	0109 ; TO RAM	0439
0023 RMSTLO EPZ #50		0400	0110 ;	0439
0024 RMSTHI EPZ #51		0400	0111 TRNSFR INX	0439 E8
0025 CRNTLO EPZ #52		0400	0112 LDA RECBUF,X	043A BD 00 0B
0026 CRNTHI EPZ #53		0400	0113 STA (CRNTLO),Y	043D 91 52
0027 YCOUNT EPZ #5B		0400	0114 CPY #FFF	043F C0 FF
0028 ;		0400	0115 BNE NXTY	0441 D0 02
0029 ; TANDOS ADDRESSES:		0400	0116 INC CRNTHI	0443 E6 53
0030 ; *** THOSE USED HERE		0400	0117 NXTY INY	0445 C8
0031 ; *** ARE CORRECT		0400	0118 CPX RECBUF	0446 EC 00 0B
0032 ;		0400	0119 BNE TRNSFR	0449 D0 EE
0033 GETFIL EQU \$B7F7		0400	0120 STY YCOUNT	044B 84 5B
0034 ERREX EQU \$B94F		0400	0121 JMP RECRDA	044D 4C 2B 04
0035 INIID EQU \$AB06		0400	0122 ;	0450
0036 OPENR EQU \$AB09		0400	0123 ; HERE IF ALL DONE OR AN	0450
0037 CLOSER EQU \$AB0F		0400	0124 ; ERROR	0450
0038 RECIN EQU \$AB15		0400	0125 ;	0450
0039 RBUF EQU \$BBFE		0400	0126 ; SO CLOSE THE FILE, TIDY	0450
0040 RREC EQU \$BBFF		0400	0127 ; UP AND RETURN TO TANBUG	0450
0041 ;		0400	0128 ;	0450
0042 ;		0400	0129 EXIT LDY YCOUNT	0450 A4 5B
			0130 STA (CRNTLO),Y	0452 91 52
			0131 JSR CLOSER	0454 20 0F AB

0132	JSR OUTRET	0457	20	0C	F8	0075	; START FROM \$7000	0944
0133	JSR JCUSF	045A	20	29	F8	0076		0944
0134	JMP RETMON	045D	4C	23	F8	0077	LDY ##0	0944
0135		0460				0078	STRT3 LDA (CRNTLO),Y	0946
0136		0460				0079	CMP ##1A	0948
0137	ORG \$700	0460				0080	BEQ ENDB	094A
0138	SECBUF BYT 0	0700	00			0081	JSR OUTALL	094C
0139	ORG \$800	0701				0082	CPY ##FF	094F
0140	RECBUF BYT 0	0800	00			0083	BNE NXTY1	0951
0141		0801				0084	INC CRNTHI	0953
0142	; END OF THE PROGRAM	0801				0085	NXTY1 INY	0955
0143	*** HULTAN ***	0801				0086	JMP STRT3	0956
						0087		4C 46 09
						0088	; STORE END-OF-TEXT	0959
						0089	; ADDRESS IN CRNTLO/HI	0959
0001	*** BLTANP ***	0900				0090		0959
0002	copyright G.G. Davies	0900				0091	ENDB DEY	88
0003		0900				0092	TYA	98
0004	OUTPUT A FILE STORED	0900				0093	STA CRNTLO	85 52
0005	IN RAM TO A	0900				0094		095D
0006	PARALLEL PRINTER	0900				0095	TIDY UP, SWITCH PRINTER	095D
0007		0900				0096	OFF AND RETURN TO	095D
0008		0900				0097	MONITOR	095D
0009	ZERO PAGE ADDRESSES	0900				0098		095D
0010		0900				0099	JSR OUTRET	20 0C F8
0011	RMSTLO EPZ \$50	0900				0100	JSR PAROFF	20 0E 09
0012	RMSTHI EPZ \$51	0900				0101	JMP RETMON	4C 23 F8
0013	CRNTLO EPZ \$52	0900				0102		0966
0014	CRNTHI EPZ \$53	0900				0103	END ADDRESS OF TEXT	0966
0015		0900				0104	IS SAVED IN	0966
0016		0900				0105	CRNTLO/HI WHILST	0966
0017	TANBUG ADDRESSES	0900				0106	TEXT START ADDRESS IS	0966
0018		0900				0107	SAVED IN RMSTLO/HI	0966
0019	OUTRET EQU \$F80C	0900				0108		0966
0020	OUTALL EQU \$F80E	0900				0109		0966
0021	RETMON EQU \$F823	0900				0110	END OF PROGRAM	0966
0022	JCURSF EQU \$F829	0900				0111	*** BLTANP ***	0966
0023		0900						
0024	ORG \$900	0900						
0025		0900						
0026	STARTO JMP STRT1	0900	4C	2F	09	0001	*** BLTANS ***	0900
0027		0903				0002	copyright G.G. Davies	0900
0028		0903				0003		0900
0029	PRINTER CONTROL	0903				0004	OUTPUT A FILE STORED	0900
0030	ROUTINES:	0903				0005	IN RAM TO A SERIAL	0900
0031	APPROPRIATE PAIR SHOULD	0903				0006	PRINTER	0900
0032	BE SELECTED TO SUIT	0903				0007		0900
0033	PRINTER BEING USED	0903				0008		0900
0034		0903				0009	ZERO PAGE ADDRESSES	0900
0035	PARON LDA ##11	0903	A9	11		0010		0900
0036	JSR OUTALL	0905	20	0E	F8	0011	RMSTLO EPZ \$50	0900
0037	LDA ##02	0908	A9	02		0012	RMSTHI EPZ \$51	0900
0038	JSR OUTALL	090A	20	0E	F8	0013	CRNTLO EPZ \$52	0900
0039	RTS	090D	60			0014	CRNTHI EPZ \$53	0900
0040		090E				0015		0900
0041	PAROFF LDA ##11	090E	A9	11		0016		0900
0042	JSR OUTALL	0910	20	0E	F8	0017	TANBUG ADDRESSES	0900
0043	LDA ##03	0913	A9	03		0018		0900
0044	JSR OUTALL	0915	20	0E	F8	0019	OUTRET EQU \$F80C	0900
0045	RTS	0918	60			0020	OUTALL EQU \$F80E	0900
0046		0919				0021	RETMON EQU \$F823	0900
0047		0919				0022	JCURSF EQU \$F829	0900
0048	SERON LDA ##11	0919	A9	11		0023		0900
0049	JSR OUTALL	091B	20	0E	F8	0024		0900
0050	LDA ##0	091E	A9	00		0025	ORG \$900	0900
0051	JSR OUTALL	0920	20	0E	F8	0026		0900
0052	RTS	0923	60			0027	STARTO JMP STRT1	0900
0053		0924				0028		4C 2F 09
0054	SEROFF LDA ##11	0924	A9	11		0029		0903
0055	JSR OUTALL	0926	20	0E	F8	0030	PRINTER CONTROL	0903
0056	LDA ##1	0929	A9	01		0031	ROUTINES:	0903
0057	JSR OUTALL	092B	20	0E	F8	0032	APPROPRIATE PAIR SHOULD	0903
0058	RTS	092E	60			0033	BE SELECTED TO SUIT	0903
0059		092F				0034	PRINTER BEING USED	0903
0060		092F				0035		0903
0061	STRT1 LDA ##0	092F	A9	00		0036	PARON LDA ##11	0903
0062	STA RMSTLO	0931	85	50		0037	JSR OUTALL	A9 11
0063	STA CRNTLO	0933	85	52		0038	LDA ##02	0905 20 0E F8
0064	LDA ##70	0935	A9	70		0039	JSR OUTALL	0908 A9 02
0065	STA RMSTHI	0937	85	51		0040	RTS	090A 20 0E F8
0066	STA CRNTHI	0939	85	53		0041		090D 60
0067		093B				0042	PAROFF LDA ##11	090E
0068	SWITCH ON PRINTER AND	093B				0043	JSR OUTALL	A9 11
0069	SEND A CARRIAGE RETURN	093B				0044	LDA ##03	0910 20 0E F8
0070		093B				0045	JSR OUTALL	0913 A9 03
0071	JSR PARON	093B	20	03	09	0046	RTS	0915 20 0E F8
0072	JSR OUTRET	093E	20	0C	F8	0047		0918 60
0073	JSR JCUSF	0941	20	29	F8	0048		0919
0074		0944				0049	SERON LDA ##11	0919 A9 11

0050 JSR OUTALL	091B 20 0E F8	0099 JSR JCURSF	0969 20 29 F8
0051 LDA #0	091E A9 00	0100 LDX #0	096C A2 00
0052 JSR OUTALL	0920 20 0E F8	0101 NXTXT CPY #FF	096E C0 FF
0053 RTS	0923 60	0102 BNE NXTY1	0970 D0 02
0054 ;	0924	0103 INC CRNTHI	0972 E6 53
0055 SEROFF LDA #11	0924 A9 11	0104 NXTY1 INY	0974 C8
0056 JSR OUTALL	0926 20 0E F8	0105 JMP STRT3	0975 4C 48 09
0057 LDA #1	0929 A9 01	0106 ;	0978
0058 JSR OUTALL	092B 20 0E F8	0107 ; STORE END-OF-TEXT	0978
0059 RTS	092E 60	0108 ; ADDRESS IN CRNTLO/HI	0978
0060 ;	092F	0109 ;	0978
0061 ;	092F	0110 ENDB DEY	0978 88
0062 STRT1 LDA #0	092F A9 00	0111 TYA	0979 98
0063 STA RMSTLO	0931 85 50	0112 STA CRNTLO	097A 85 52
0064 STA CRNTLO	0933 85 52	0113 ;	097C
0065 LDA #70	0935 A9 70	0114 ; TIDY UP, SWITCH OFF	097C
0066 STA RMSTHI	0937 85 51	0115 ; PRINTER AND RETURN	097C
0067 STA CRNTHI	0939 85 53	0116 ; TO MONITOR	097C
0068 ;	093B	0117 ;	097C
0069 ; SWITCH ON PRINTER AND	093B	0118 JSR OUTRET	097C 20 0C F8
0070 ; SEND A CARRIAGE RETURN	093B	0119 JSR JCURSF	097F 20 29 F8
0071 ;	093B	0120 JSR SEROFF	0982 20 24 09
0072 JSR SERON	093B 20 19 09	0121 JMP RETMON	0985 4C 23 F8
0073 JSR OUTRET	093E 20 0C F8	0122 ;	0988
0074 JSR JCURSF	0941 20 29 F8	0123 ; END ADDRESS OF TEXT	0988
0075 ;	0944	0124 ; IS SAVED IN	0988
0076 ; START FROM \$7000	0944	0125 ; CRNTLO/HI WHILST	0988
0077 ;	0944	0126 ; TEXT START ADDRESS IS	0988
0078 LDY #0	0944 A0 00	0127 ; SAVED IN RMSTLO/HI	0988
0079 LDX #0	0946 A2 00	0128 ;	0988
0080 STRT3 LDA (CRNTLO),Y	0948 B1 52	0129 ;	0988
0081 CMP #1A	094A C9 1A	0130 ; END OF PROGRAM	0988
0082 BEQ ENDB	094C F0 2A	0131 ; *** BLTANS ***	0988
0083 ;	094E		
0084 ; REPLACE NON-PRINTING	094E		
0085 ; CHARACTERS WITH A SPACE	094E		
0086 ;	094E		
0087 CMP #20	094E C9 20		
0088 BCC NXTXTA	0950 90 0A		
0089 CMP #7B	0952 C9 7B		
0090 BCS NXTXTA	0954 B0 06		
0091 JSR OUTALL	0956 20 0E F8		
0092 JMP NXTX	0959 4C 61 09		
0093 NXTXTA LDA #20	095C A9 20		
0094 JSR OUTALL	095E 20 0E F8		
0095 NXTX INX	0961 E8		
0096 CPX #46	0962 E0 46		
0097 BNE NXTXT	0964 D0 0B		
0098 JSR OUTRET	0966 20 0C F8		

FOR TELETYPEWRITER USERS

If the files are created in BASIC, as in "HLTST", a <CR> is issued at the end of each line. In fact the <CR> terminates each record.

Consequently, a <CR> is available at the end of each line to control the teletypewriter. Therefore, either use BLTAMP, if published, with calls to SERON/SEROFF replacing PARON/PAROFF or, carry out the modifications to BLTANS but leave the SERON/SEROFF calls in place.

```

5 REM**
PROGRAM "HLTST"
10 REM** A PROGRAM TO HELP DEBUG THE DISC I/O PROGRAM "HULTAN"
20 REM**
30 REM** REMEMBER THAT IN BASIC A RECORD IS TERMINATED BY A <CR>
40 REM** AND IS ONLY ONE LINE LONG *****
50 REM**
60 REM** FIRSTLY OPEN A FILE FOR WRITE
70 REM**
80 DOPENW "HLTST" :REM** AND DON'T FORGET TO NAME THE FILE
85 REM** AND NOW CREATE THE RECORDS
90 DPRINT "A BELATED SET OF BEST WISHES FROM THE NORTH WEST"
100 DPRINT "USERS ARE CORDIALLY INVITED TO MAKE CONTACT WITH"
110 DPRINT "GRAHAM DAVIES ON BOLTON 654145"
120 DPRINT "THE GROUP IS INFORMAL AND RELIES A LOT ON"
130 DPRINT "USING THE TELEPHONE BUT WE HOPE TO MEET IN BOLTON"
140 DPRINT "DURING SEPTEMBER OR OCTOBER AND THEN REGULARLY"
150 REM** SO WE HAVE SIX RECORDS"
160 REM** NOW ADD THE END-OF-TEXT MARKER HEX1A
170 DPRINT CHR$(26)
180 REM** AND CLOSE THE FILE AFTER WRITE IF YOU WANT IT ON DISC
190 DCLOSEW
200 END
OK

```

An update on this article will be published with the next issue.

MICROTANIC COMPUTER SYSTEMS LTD.

Computer Manufacturer
Software · Hardware · Books

Registration No: 1668843

V.A.T. No. 237794718

16 Upland Road
Dulwich, London SE 22

Telephone: 01-693 1137

TANSOFT 2 PASS ASSEMBLER

A Two Pass Assembler is now available for Tangerine Users. No more messy patches, just amend the Source Code and re-assemble to whatever location you desire. This Assembler is available in a 2732 Eprom and simply plugs into J2 on the Tanex Board. If you are a Basic User, there is also an Extension Eprom Board available which allows Software or Manual Switching between banks of Eprom space located between C000 and DFFF(hex). So 8k of Basic goes in one bank and the Assembler goes into the second bank leaving 8k Free for either our Forth Language or our Asimov Word Processor or 8k of your own Firmware.

What is a 2 Pass Assembler?

For those unfamiliar with 2 Pass Assemblers, they greatly simplify the task of writing Machine Code Programs. The main differences between a 2 Pass Assembler and the Line by Line Assembler (as in XBUG) are:

- a) A copy of the Source Code is always retained, that is, a copy of what YOU actually type in (eg. LDA).
- b) Any Instruction or Data Location can be given a Label (a name).
- c) These Labels may be referenced by Instructions, so, for example, rather than JSR to a specific location you may JSR FRED, where Fred is perhaps the beginning of a Subroutine.

The great advantage of this system is that Instructions can be inserted or deleted from the Source Code, Labels swopped around, Data Areas altered in length etc, without worrying about the effect on addresses.

This system implies that you can assemble object code anywhere in memory.

Facilities:

The facilities offered by this particular Assembler are:

- * Full source code editing including validation of instructions. *
- * Cassette Handling Routines for maintenance of Source Code on Tape. *
- * All standard 6502 Instructions are supported in the same format as used by the XBUG Line Assembler (except for ASL A which is replaced by ASL @ to avoid confusion with Labels). *
- * Extra Instructions are supported - ORG (relative or absolute) EPZ(equate page zero), BYT(for single byte constants) and WOR(for 2 byte addresses). *
- * Hexadecimal, character and decimal constants are supported. *
- * Reference to Labels may include the format Label+constant(eg. FRED+2). *
- * Various Print, List and Display options, including the facility to list the Labels and their Addresses. *
- * Options to Dump or Assemble parts of the Source Code and to Load or Assemble several parts into one program. *
- * Option to Assemble at an Eprom Location but to Store the Object in another RAM Address ready for Eprom Burning. *

*** AND ALL THIS IN 4K !!! ***

SUPPLIED IN EPROM 34.95

SPECIAL OFFER

H2 EXTENSION EPROM BOARD PLUS 2 PASS ASSEMBLER EPROM ... 49.45

SUBSCRIPTION

Microtan Computer Systems Ltd. 16 Upland Road, Dulwich,
London. SE 22. 01 693 1137.

Please send me the next six copies of Microtan World.

Name

Address

.....

I enclose £10:00 in full payment.

Cheques should be made payable to Microtan Computer Systems Ltd.

NEXT ISSUE

A lot of the content of the next issue will depend on you the readers.

We are particularly interested in articles for single board owners,
useful routines, hints and tips you may have found, etc. etc.

Don't forget to let us have your letters telling us what you think
about this issue and of course any comments on the system or any matter
of general interest.

If you know anyone with a Microtan Computer who is not yet subscribing
to Microtan World why not let him have the subscription form and see
if you can persuade him to join us. The more subscribers we can get
the better we can make the magazine.