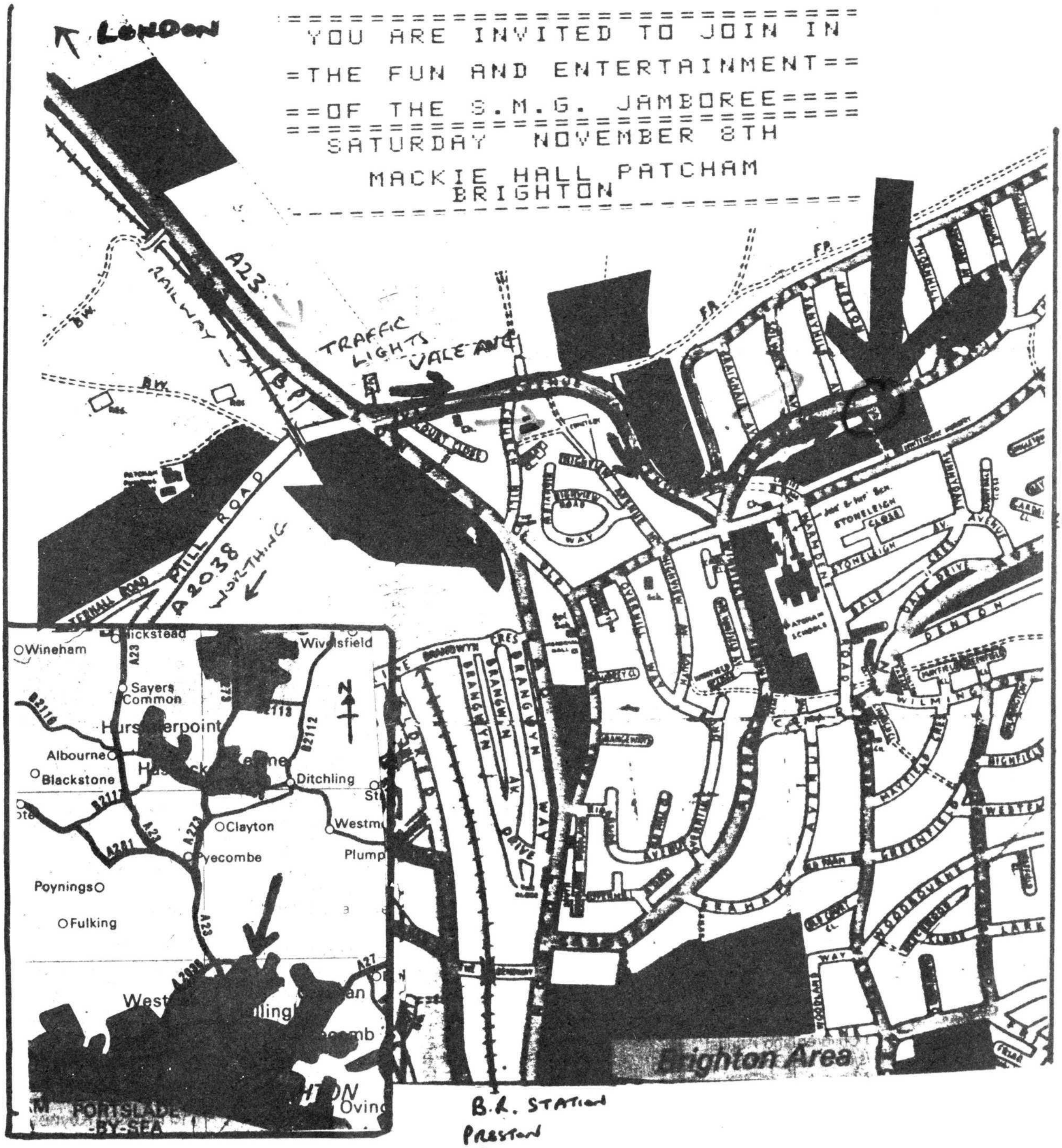
TI*MES

AUTUMN ISSUE NUMBER 14

YOU ARE INVITED TO JOIN IN



YOU ARE INVITED TO JOIN IN

THE FUN AND ENTERTAINMENT

OF THE S.M.G. JAMBOREE

SATURDAY NOVEMBER 8TH

MACKIE HALL PATCHAM
BRIGHTON

ADMISSION CHARGE 50P

SEE THE DEMOS OF
TI99 SYSTEM ALONG WITH
OTHER MICROCOMPUTERS

GO BY COACH OR RAIL TO
PRESTON PARK BRIGHTON
& CATCH A NUMBER 5 BUS
TO MACKIE AVENUE, PATCHAM

THE S.M.G. JAMBOREE

Welcome _____________ to TI99/4a EXCHANGE

TI*MES TI*MES TI*MES TI*MES TI*MES TI*MES TI*MES TI*MES TI*MES TI*MES
AUTUMN ISSUE NUMBER FOURTEEN

40,Barrhill,Patcham,BRIGHTON,East Sussex,BN18UF.Tel:0273 503968(evenings)

EXPANSION IS STILL POSSIBLE

There must be a number of you who wonder about the expense of expanding compared to all the new micros on the market today. Because there is so much choice today I can understand why over 70% of TI99/4a users have gone over to another system. It is very sad of course when we lose members from the group, but one has to face up to reality these days. However all is not lost because in this issue we feature another Do It Yourself expansion system as promised by Dave Hewitt following our last publication which featured Mike Goddards DIY system. Incidently if you HAVE COMMENCED building this please take note of the ERRORS. Best to ring Mike to check your progress. Thanks to both Mike Goddard and Dave Hewit one can be given a challenge. This excellent article on page 35 will provide interesting reading. It confirms my belief which is shared by many TI99 users around the country and indeed all over the world that you cannot put down a super machine like our TI99/4a.

Personal Computer World magazine October edition published details of this group (thanks to our own John Bingham) the response has been staggering. People have written stating that they have brought out the TI99 from store and only just started to use it after many years of nonuse. A fine example of how the TI99/4a is standing the test of TIME. I have got to admit that with a 16bit machine one has power at a very low cost. Proof that our TI99/4a still has very much to offer us whether you have expansion or not.

In addition to our regular contributors I would like to pay tribute to the many WORLDWIDE TI99/4a Users groups, they continue to enrich the use of our machine. In this issue we publish extracts from many such excellent TI users. Hope you will enjoy.

One final note is our next meeting on SATURDAY 8th November, this time not exclusive TI99/4a but joining in with the professionals. The venue is our homebase here in Patcham where a Computer Jamboree has been arranged. Hope you will be able to attend. See some ideas in action and again pick up some bargains for Christmas.

Happy 99ing,

Clive S.

Clive Scally.

TI994a Exchange TI*MES newsletter is supported only by its subscribers. This TI users Group is INDEPENDANT of Texas Instruments and is completely non profit making. - TI*MES is published quarterly, JANUARY, APRIL, JULY, and OCTOBER months.The annual subscription is £10 and includes 4 newsletters. Editorial etc is provided by group members, other user-groups and other related sources. _ Views expressed are those of the writer and not necessarily those of TI994a Exchange. Whilst efforts are made to ensure accuracy no responsibility can be accepted by TI 994a Exchange as a result of the applying of such information found within the pages of TI*MES . You are invited to contribute copy for publication in TI*MES. If you would like to make a contribution please submit COPY ON A4 ONLY this MUST be TYPED with a disk or tape if a program is included. Last DATE OF ACCEPTANCE IS 20TH PRIOR TO PUBLICATION MONTH FOR COPY. Unaccepted material will be returned ONLY if accompanied by a S.A.E. No material may be reproduced without credit to the author and TI994a Exchange U.K. MEMBERSHIP YEAR NOW ENDS MAY

RAMBLES RAMBLES RAMBLES

RAMBLES is (c) Copyright 1986 by Stephen Shaw.
10, Alstone Road. STOCKPORT. Cheshire.
ENGLAND. SK4 5AH.
The contents may be reproduced in part or whole provided this consent is included and a copy of the publication is sent to the author at the above address.

I had so much material left over from the July issue of Rambles. that THIS issue is being started in JUNE! First, some genuine Rambles...

I have seen an advert for a box to fit to your video recorder. which allows you to record music onto the video tape. The advantage of this is that you cannot buy a C960 cassette tape.... the quality of the sound from your recorder will of course depend on the recorder.
Why buy a black box instead of just feeding the sound in? Because the Video Recorder will not record sound without a picture... all the black box does is to inject a blank black screen into the video recorder!
Of course if you have a TI home computer (or possibly any other computer) you can insert the black (or any colour!) screen with your computer. If you feed the sound into the cassette socket, it is transferred to the tv output onto the video.
Setting up may not be easy: the 99/4A modulator is set to about Channel 36, and many video recorders will be set very close to this channel : if they are close, you will obtain interference on the video picture, and may be unable to lock onto it at all. The solution is to retune the video recorder output frequency. All (almost all?) video recorders can do this as in some parts of the world, Channel 36 is used by broadcast stations! The retuning will probably be by means of a very small (I mean SMALL) screw, possibly accessed at the back of the video recorder. Tune it as far from Channel 36 as you can - of course you will also need to retune the video channel on your tv set!

It is becoming difficult to find TI books at my local bookshops, but in the sale I found a copy of Scott Vincents book "Dynamic Games for your TI99/4A" for a fraction of its original price - marked down from 4.95 to 1.50, and then down again to 75p! Irresistable.
But why would I want a book of simple TI Basic games after all these years? Easy- TI Basic games can still be fun. Simple TI Basic programs can be made more sophisticated. And simple TI Basic programs can be material to transfer to faster and more difficult languages.
Want to learn Forth or C ? Take a simple TI Basic program and translate it! You'll learn! Actually, in case you haven't seen it, Scott's book has some very nice programs in it anyway!

```
CCCCC
C
C
C  -99
C
CCCCC
!!!!!!!
```

C.... in answer to the difficulties some of you have had in actually using C I have prepared a short intro for absolute novices. If space permits it will appear in this issue.

Remember, RAMBLES is based on your requests and questions. So keep them coming in!

Good books for C learners are published by QUE and your local bookshop should be prepared to order them for you.
The first two books are for learners, and either one should prove suitable:
C SELF STUDY GUIDE. Jack Purdum.
C PROGRAMMING GUIDE: 2nd Edition. Jack Purdum.
Also of interest, especially for boffins, are:
COMMON C FUNCTIONS by Kim J Brand and
C PROGRAMMERS LIBRARY by Purdum, Leslie, and Stegemoller.

I have already reviewed TI ARTIST, ARTISTS COMPANION and ARTIST EXTRAS all from TEXAMENTS, 53 Center Street, Patchogue, New York, USA, 11772, and also spoken highly of their quick and efficient service. So you wont be too surprised if I tell you I have purchased some more products from them, and also consider these very highly...
ARTISTS COMPANION #2 at US$10 for two disks is a little more expensive than ARTISTS COMPANION #1, at US$18 for 5 disks! What Companion 2 contains is: 13 new Fonts, 60 5x5 Artist Instances (nice owl, TI Logo, etc etc) and also a program which reads an instance and creates a BASIC program which displays it on screen. This can of course be merged into your own programs. Facilities include locating the instance on screen, having several copies of it on screen, and selection of the first character to be redefined. The size of instance you can work with is limited by the number of redefinable characters, but the program does check for repeated characters in case the instance is on the large size.
DISPLAY MASTER at US$15 is a neat machine code program which in essence allows you to create a slide show of TI Artists PICTURES. You can however still obtain value from it with no Artist pictures as it uses windowed captions which are really very neat indeed. You can place a window on screen with text on it over a picture, then remove the window to reveal the picture. OR you can overlap windows, and strip them off... the only restriction is that when removing one window at a time, you must remove them in the reverse order they were put on! Display Master uses DV80 files to obtain its instructions, which are very easy to use.

XB DETECTIVE at US$20 is perhaps a trifle overpriced. as its functions can be had (more untidilv and not so conveniently) in Freeware/Public Domain programs. However if you have a fully expanded system and are still using ExBas you may find the convenience worth having. You load XB DETECTIVE, which is a machine code program, and then access it by means of an interrupt driven key scan. In other words, with an XB program loaded you can quickly obtain a complete list of variable names, locate the positions of variables, and a set selection of reserved words. You can delete lines, including blocks of lines, very quickly indeed - much better than any other way of deleting lines! And you can search for strings.

The prices above do not include postage. I would suggest you add US$4 for one disk, US$2 for each extra disk.

I also have (June 1986) a quotation from TEXAMENTS for a MYARC 512k card PLUS Myarc Level 4 Extended Basic for US$344 inclusive of air mail postage and insurance.

Remember the postman will insist on your giving him ABOUT an extra 22% before he makes delivery! (And you may be able to sell your TO PAY stamps on the package... they are only used on postal imports now, and Stanley Gibbons charge OVER face value for some of the higher valued ones, so you should get SOMETHING for them!!!).

Order sent by airmail with IMO for US$344 on 9th July 1986.

Order received on 22nd July 1986.

THIRTEEN DAYS round trip - incredible!

GOLD MEDAL for service to TEXAMENTS! Review later on!

Genial TRAVelLER diskazine from Barry Traver, mentioned in TI*MES number 11:

Supposed to be bimonthly. My first copy arrived December 10th 1985, as at 10th July 1986, three copies to hand. An overseas price has now been quoted: Six Issues (double sided flippy) for US$50. Address in issue 11. Note that publication dates are somewhat flexible!

I must admit that the second and third issues did not seem to come up to the standard of the first. and I only found one program I liked on Issue 2, and nothing of personal use on issue 3. Issue three contained a 'definitive list' (sic) of modules released by TI. I personally have several not listed - including one which most TI owners will have heard of: Extended Basic!!!

SORRY!
I can use up only SO many pages of TI*MES! And there is so much material to cover.
I regret we again have to give Forth a miss, and cannot continue the TI Writer article.
SORRY!

The members disk library of freeware and public domain disks remains available - if you have a disk system and would like details please send a blank disk and return postage. Well over 100 DISKS and probably over a thousand programs (who counts?). Some remarkable offerings.

I strongly recommend Renee LeBlancs disk disassembler to you - the only disassemler I have which works perfectly, although Renee suggests there may be faults...

New revisions in stock are for:

FUNLWRITER now into Vn 3.3, and if you have an earlier version you'll want this latest one, now on TWO disks.

SBUG2 is now into Version 2.0, but the expanded PRINTED documentation makes it unsuitable for supply by me - just US$13 direct from Ed (more details in library list) PRBASE is now into 2.0, with a much improved CREATE program.- two disks!

C99 is of course into version 2.0, and if you have Version 1, you need version 2.0 !!! (yes, two disks!).

New library items include 11 disks of high resolution graphics, some of them prepared on other machines using a digitiser, and a special loading program for them, which is also compatible with Graphx and TI Artist 2, with conversion capability. (RLE/1 has the program and four pics, the other pics are on RLE/2 to RLE/11

And a disk for TI ARTIST users with some fonts, slides and instances. Why not send for a current library list? Due to reduced demand I am now into deficit on this and unless more interest is shown, supply of new items could be a problem ("free" is a nice word but rarely true!!).

++++++++++++++++++++++

I understand that we have somewhat fewer members these days, which makes it even MORE important for you to drop me a line and tell me what you would like to see in Rambles - or perhaps just as important, what you'd rather NOT see in Rambles.

For instance is it helpful for me to review products? Magazine issues?

Tom Freeman of the LA group has now amended his super two-column print utility, used to prepare this article, to produce a four column print - I found it was just a little bit crowded so I'm not going to use 4 columns here. Using the three column option was useful for the TI*MES index though.

Our friends in LA have printed an article on C99 by Warren Agee in three of their newsletters. Very helpful. It runs to 12 pages, which is expensive to copy especially in one-off runs - even in reduced form Xerox would charge me £1.38 or so (their prices have a habit of going up just after I quote them...!!!), but if anyone wants a copy, I'm happy to oblige at cost.

Dick Altman supplied the LA group with an extensive listing of Freeware programs (I only have 3 not listed) and has kindly sent me a text file of instructions on using TI Writer, which may be of help to anyone without a manual (or anyone struggling with the manual!) although of course you need to be able to read the file! Also on the disk is a TI Writer article by Tom Kennedy. One disk, usual library terms.

Which reminds me of another addition to the library, though on past demand I'm not going to be sending any copies out - apparently nobody is interested in games anymore, except me! The addition is an XB program to enable two players to play CHINESE CHESS, quite a rare game outside Chinese circles. Details of legal moves are supplied, and the program is an excellent example of how well you can program in XB if you really try!

I don't carry every freeware item by the way, although I often spend rare (extinct) funds in obtaining them. Some just aren't worth offering. Be assured that SOME thinning out goes on!

I am sending Clive an Index for the previous issues of TI*MES, which may or may not be in this issue. Is there any demand for an index for other publications such as TI USER, TIDINGS, TI LINES, 99er Magazine, SUPER 99, SMART PROGRAMMER etc etc. I have complete runs of these. The only magazine I'm short of is the Parco mag and the last half dozen "Home Computer Mags" - I gave up getting it when they carried other computers. . Drop me a line if interested in indexes. With a huge pile of material it can sometimes be hard to find a particular item...
Note that every issue of Micropendium carries a full index to product reviews.

The LA group have warned that BRAIN from DATAX at US$50 no less, is a lot less than the product advertised. Is that a worthwhile price for 53 conversion routines -eg to calculate area, volume, change grams to tons, and so on? There are many more routines in the program, but all of this nature, and the LA group have questioned the accuracy of some of them. Treat with care. It may prove to be NOT "the most useful software product ever developed for TI99/4A".

Micropendium has warned that a new module, 4A Flyer, has some odd deficiencies, such as the impossibility of heading nose down into the ground - it is possibly the worst review in Micropendium! I detect though a degree of disappointment after a lot of publicity hype. I suspect it may be a better module than many of the modules TI released!

OK...onto hotter stuff... a review of Myarcs ram card and XB II Level 4 Version 2.0....
Ram card first...
The Myarc ram card fits into the TI PEB in the normal manner and if you wish you can then operate it just as though it was a TI 32k card... but if you have a 512k card you can do a little more...
There are four new commands, available to you in TI BASIC:
CALL RDTEST goes through the card testing the chips - a useful command. If one chip blows, this command will reveal it and it can be replaced.
TI Ex Bas can only use the normal 32k, which leaves the remaining memory for a solid state disk drive, and a printer buffer.
The TI Disk operating system has been so designed that the maximum disk size you can use is 1600 sectors, including disk header, and that is the maximum size you can use with the ram card. If you wish, you can use a disk initialiser to provide an exact duplicate of a standard disk, but you do not need to : just partition the memory to emulate a disk of the size you want, and it is then fully compatible with all disk functions.
The remaining memory is available as a printer buffer : the cursor reappears VERY quickly even when you PRINT a thousand line file! 'course the printer keeps printing for a while....
CALL PART is the command to set the partition parameters.
At this stage, the ram disk is a device called "RD", but you can also set parameters so that your programs recognise it as DSK1 to DSK5 or for instance DSK.TIMP
The commands for this are CALL EMDK and CALL VOL, and these titles can be changed (in TI Basic) at any time without affecting ram contents.
If you set it to DSK1, it disables your normal drive no.1 - and similarly for other drive numbers, DSK2 will disable your second drive and so on.
CALL RDDIR will list the ram disk contents to screen.
The print spooler has its own device names: RS232 becomes SP and PIO becomes SPPIO - and if you wish to abort a spooling operation, you can CALL ABPS.

If you program in machine code, you can also select banks of the ram card by setting CRU bits : the instructions I received were for the 128k ram.

VERY happy with this purchase, and astonished by the service Texaments gave. Using a ram-disk, access to programs - especially in machine code - is incredibly fast, making it very suitable for development work - shifting rapidly from TI Writer to Assembler to C Compiler to debugger and so on.

Just one tiny inconvenience - if you load Funlwriter into the ram disk, you must call the ram disk DSK1 in order to properly use it - a minor quibble. *See later!

And now a bad review- MYARC EXTENDED BASIC II LEVEL 4 VERSION 2.0... don't buy it!!!

Very few of the advertised claims are true for this version : it is by no means compatible with TI ExBas! It does not support SPEECH, it does not support DEF, it does not support SUB, it does not support cassette operation, it does not support INT as advertised, it runs notably slower than TI XB, and the display routines are bugged. CALL LINK code is incomplete.

Myarc promise a free upgrade to Version 2.1, but as the missing code is probably the most difficult to write, and the promise has been dragging for some time...!

The package consists of:

An EPROM to fit into a 128k or 512k Myarc ram card (Texaments kindly kitted mine for me - no charge!), a module. and a disk. The disk can be copied into the ram card and operate from there - after every QUIT you need to reload the language so you SHOULD copy the disk to ram-card!

What extras are there? You have ready access to each display mode from Extended Basic commands: 32 or 40 columns, with the capability of altering the screen width within these parameters (eg using 26 columns of the 32 column screen). The bit map mode allows you to place text (32 column size) fairly easily and there is a good range of bit map commands.

CALL PART can be used from 128k XB. In development work, you can be lazy and instead of typing OLD DSK1.WORK and SAVE DSK1.WORK just type in OLD WORK and SAVE WORK. You can opt to use RUN "DSK1.FILE" and keep variable values intact.

Here is a COMPLETE list of new commands in Version 2.0: CALL GRAPHICS, CALL PEEKV, CALL POKEV, FREESPACE (replaces SIZE), TERMCHAR, VALHEX (preceding listing in Manual Appendix - following appear in Manual index..) CIRCLE, DCOLOR, DRAW, DRAW TO, FILL, MARGINS, POINT, RECTANGLE, WRITE.

The manual cover mentions CALL POINTSTAT (sic) but that is the only reference! It isnt there anyway. WINDOWing is by means of CALL MARGINS which sets the left, right, top and bottom margins of the screen.

CALL VERSION returns 200.

An ambitious project - presently very incomplete. Despite a total disclaimer of any product responsibility whatsoever, Myarc must be sailing VERY close to the wind on "mail fraud" charges!

However... if you are interested in a NEW language for your console rather than one which is a fully compatible ExBas, this package may be of interest, despite one or two bugs and the total absence of memory mapping details. The bit map graphics ARE really very easy to use, and circles are drawn QUICKLY (a good test of a bit map drawing utility!). You also have 255 redefinable characters and 32 sprites in normal graphics mode - yes, 255 definable characters. The characters sets differ from TI's and come close to true lower case. The text mode (40 column) is used almost exactly like the 32 column mode, all the same commands. (NB: Using the ram card's CALL RDDIR you must be in 32 column mode for readable results!).

C FOR TOTAL BEGINNERS

c-99 is a newish language for the TI99/4A which is proving remarkably popular with some of the "boffin" owners.

The reason is this: It is a compiled language: that is, you write a program in c, and when you run it, you are really running machine code- there is no delay for interpretation, as with Basic.

The c package is on two disks, and requires a console, at least 32k ram, at least one disk drive and disk controller. A printer is useful. c-99 can be operated with the Editor Assembler module, or with Extended Basic if you also have Funlwriter v.3.3

The c-99 package has documentation relating to the use of c with the TI99/4A, but minimal information on the language itself, and rather skimpy instructions (for the novice!) on how to actually USE c-99.

This article is intended to help you to take a short piece of c source code and produce a running machine code program.

There are many books on the c language - unfortunately most of them seem to be around twnety pounds or so - and I can strongly recommend the series published by QUE, which includes:

C SELF STUDY GUIDE by Jack Purdham
C Programming Guide-2nd Edition by Jack Purdham
Common C Functions by Kim J Brand
C Programmers Library by Purdum,Leslie and Stegemoller.

(Books-continued...)

Either of the first two alone should be a good start - your bookshop will have to order the books for you in most cases, but there will be no difficulty -if you have a good bookshop!

Note that the above books are on C - not on c-99, which is not a full implementaion. Thus there are a few things you wont be able to do: for instance, c-99 does not (yet) support goto or multi-dimensioned arrays.

Now lets take a BASIC program:

```
100 CALL CLEAR
110 FOR ROW=1 TO 24
120 FOR COLUMN=3 TO 30
130 CALL HCHAR(ROW,COLUMN,42)
140 NEXT COLUMN
150 NEXT ROW
160 CALL KEY(4,K,S)
170 IF S<1 THEN 160
180 IF K<>26 THEN 100
190 END
```

NOTE: The following pages have some none-C material in the blank spaces, so DON'T just skip all these pages without looking at them first!!!!!!

Not too hard to follow that is it? This program-ette clears the screen, and then rather slowly fills it with asterisks.

----continued---->

I have used Key Unit 4 to keep this program comparable with the C listing which follows, in which CTRL Z is used to drop out of the program.
Pressing any other key re runs the program.

Now here is the same program in C Source code. This is written with a text editor such as TI Writer or Editor Assembler. If you use TI Writer, you must save the text WITHOUT the final tabs, either by using PF instead of SF, or by using the special c editor in the FUNLWRITER package.
Remember, c-99 uses the 40 column screen by default, so the column numbers are a little different to our basic program:

```
/* ========================= */
/* Malcolm's test program 1 */
#include dsk1.conio
int row,col;
main() { while(1)
  { row=0:
    putchar(FF);
    while(++row<25)
    { col=6;
      while(++col<35)
      { locate(row,col);
        putchar(42);
      }
    }
    if(getchar()<1) break;
  }
}
/* ====================== */
```

The /* ... */ lines are REM lines and have no effect.
#include dsk1.conio:
The lower case dsk1. is CORRECT!
This line will make the compiler insert here the text which is in the file CONIO, which contains certain common definitions - eg that FF is to be read as 12 (which is a FormFeed for screen OR printer).
CONIO is included in the c-99 package. It will be used with most c programs that you write.
int row,col;
In C, variables MUST be defined before you use them. We here define the two variables row and col to be integers, and by so doing we reserve two bytes of memory for each. Almost all lines of code in c are terminated with a semi colon. From reading the C text books you will appreciate that as used here, row and col are GLOBAL VARIABLES. Dont worry about that now.
main() is the main part of the program. As with ExBas, you can have named sub program such as color() or sound(), and you have the option of having local variables in these - just as you do with ExBas.
{ while(1)
sets up the basic program loop- remember we dont have GOTO in c-99!

The opening { marks the beginning of the command. Provided the stack has a 1 in it when the program loops back to while (1=TRUE) then the loop is reentered, else you drop out of it.
The rest of it you should be able to follow by reference to the BASIC program - perhaps the ++row may throw you a little: this adds one to row and then makes the comparison. row is permanently incremented by this space saving instruction.
eg= row=row+1
if row<25 then...

We now have on disk the above file, lets call it TEST, and it is in DISPLAY VARIABLE 80 format.

It now has to be compiled.

We therefore load the memory image machine code program C99C (using Editor Assembler Option 5 or Funlwriter Users List.
The C99C program will auto start and ask you to input various options.

You may include the c source text as comments in the 9900 source code if you wish. You may use an "in line push" (referring to stack usage) if you wish.
Let's keep it simple: answer n (or N) to the first two questions.

The compiler will now ask for input and output filenames.
Our C source was saved as TEST - remember! So that is our input filename.
Output filename can be anything you want...lets call it TEST/SO.

You will recall that TEST is going to make the compiler load CONIO from Disk 1, so make sure that the file CONIO is on a disk in drive one! and that the input file TEST is on the disk in the drive specified, and that the disk in the output drive specified has room on it for our output file!

Off you go. As the compiler is in action it will tell you what program it is compiling - so (main) will appear on screen.

When it has finished - quite quickly - you will have a 9900 Machine Code SOURCE file called TEST/SO, which must now be assembled using the TI Editor-Assembler.

Select ASSEMBLE and indicate on the screen which appears:
Source file: DSKx.TEST/SO
Object file: DSKx.TEST/OB
List device: [press enter]
Options: [none- press enter with a blank here!]
and off the assembler goes, producing a machine code object file which you can then run... or can you?

continued---->)

No. c-99 produces code which is incomplete. You need to "add the end".
To run our test program, you must:
Select Editor Assembler Option 3 (or FUNLWRITER Utility Option 4).
Enter the first filename: TEST/OB
then a second filename... from the c-99 package. CSUP.
NOW you are ready to roll... press ENTER to move to the next item on the menu [PROGRAM NAME] and ask the computer to start running: START and off you should go.

If you wish to produce a memory image file to load with Editor Assembler Option 5, it is not that simple.

Select Option 3 as before. Now you must load ALL the following files, in this order:
C99PFI TEST/OB CSUP C99PFF SAVE (or FWSAVE)
Now press ENTER to go on to the PROGRAM NAME section and ask the computer to RUN the program called SAVE.
For the output file name indicate something like TEST/OP5 and off it goes...

(Files C99PFI, CSUP, C99PFF are part of the c-99 package.
SAVE is part of the TI Ed/As package.
FWSAVE is part of FUNLWRITER package)

To load and run this new file (TEST/OP5) you must select either Option 5 of Editor Assembler or Funlwriter Utility option 3.

If you have carefully followed these instructions, and made certain that all the right disks were in the right drives! you should now have a working machine code program, compiled by c-99.

Hint: It helps to keep a few blank initialised disks handy, and copy onto them whatever C libraries and functions your program will need, such as CONIO and CSUP!

You MUST always load CSUP after loading the object files produced by C.
You SHOULD always #include either CONIO or STDIO in your C source code. STDIO is REQUIRED for file output.

If you are using a C source file from someone else, make sure you know what "include files" you need - you may need more than STDIO. Include files are used to define labels and also to insert the REFs which the Assembler requires where the C source has included some machine code (denoted with #asm ... #endasm).

Check to see if you need to load any other function files before CSUP.

The strength of c is the ability to write programs in small neat packages and then put them together as and when required.

NB: Some c function files exist in DF80 form ready to load, which have been compiled by C99REL1. These will not work with C99Rel2. It is preferable for C functions to be supplied (and kept) in C Source form - then you can compile and assemble for your version. Memory image files compiled by C99Rel1 will not load with Funlwriter.

Just for reference lets take a look at what the compiler produces from our little test program.

First with text AND with "in line push"

```
*
*c99 v2.0 (c) 1986 Clint Pulley
 REF C$CIND,C$DIV,C$REM,C$ASR,C$ASL,C$EQ,C$NE,C$LT,C$LE
 REF C$GT,C$GE,C$ULT,C$ULE,C$UGT,C$UGE,C$LNEG,C$SWCH
 REF GETCHA,GETS,PUTCHA,PUTS,LOCATE,POLL,TSCRN,EXIT
*/* Malcolm's test program 1 */
*#include dsk1.conio
*/*
*** CONIO : C99 console I/O definitions
**/
*#define stdin  (-1)
*#define stdout  -2)
*#define stderr (-3)
*#define EOF    (-1)
*#define YES      1
*#define NO       0
*#define NULL     0
*#define EOL     10
*#define FF      12
*#define BS       8
*int row,col;
ROW BSS 2
COL BSS 2
*main()
 DEF MAIN
MAIN
*{ while(1)
C$2
 LI 8,1
 ABS 8
 JNE $+6
 B @C$3
*  { row=0;
 CLR 8
 MOV 8,@ROW
*    putchar(FF);
 LI 8,12
 DECT 14
 MOV 8,*14
 BL *12
 DATA PUTCHA
 INCT 14
*    while(++row<25)
C$4
 INC @ROW
 MOV @ROW,8
 DECT 14
 MOV 8,*14         --more-->
```

NEWS: MYARC 128K CARD -> 512k
Have you purchased a Myarc 128k ram card and added your OWN ram chips to bring it to 512k? Any problems? Lou advises that some Eproms supplied with early 128k cards will not be happy with 512k. Drop Myarc a line and tell them the Eprom Number in your card and say you've upgraded yourself (send 2 international reply coupons)

MYARC ^^

```
 LI 8,25
 BL @C$LT
 ABS 8
 JNE $+6
 B @C$5
*     { col=6;
 LI 8,6
 MOV 8,@COL
*       while(++col<35)
C$6
 INC @COL
 MOV @COL,8
 DECT 14
 MOV 8,*14
 LI 8,35
 BL @C$LT
 ABS 8
 JNE $+6
 B @C$7
*        { locate(row,col);
 MOV @ROW,8
 DECT 14
 MOV 8,*14
 MOV @COL,8
 DECT 14
 MOV 8,*14
 BL *12
 DATA LOCATE
 AI 14,4
*          putchar(42);
 LI 8,42
 DECT 14
 MOV 8,*14
 BL *12
 DATA PUTCHA
 INCT 14
*        }
 B @C$6
C$7
*     }
 B @C$4
C$5
*    if(getchar()<1) break;
 BL *12
 DATA GETCHA
 DECT 14
 MOV 8,*14
 LI 8,1
 BL @C$LT
 ABS 8
 JNE $+6
 B @C$8
 B @C$3
*  }
C$8
 B @C$2
C$3
*}
 B *13
 END
```

Does anyone have a FOUNDATION ram card? If so, you can also run Myarcs new Extended Basic Level 4 by asking for an appropriate EPROM (US$10 extra) when you buy the Myarc XB.

Myarc

and now WITH TEXT but WITHOUT "in line push":

```
*c99 v2.0 (c) 1986 Clint Pulley
 REF C$CIND,C$DIV,C$REM,C$ASR,C$ASL,C$EQ,C$NE,C$LT,C$LE
 REF C$GT,C$GE,C$ULT,C$ULE,C$UGT,C$UGE,C$LNEG,C$SWCH
 REF GETCHA,GETS,PUTCHA,PUTS,LOCATE,POLL,TSCRN,EXIT
*/* Malcolm's test program 1 */
*#include dsk1.conio
*/*
*** CONIO : C99 console I/O definitions
**/
*#define stdin  (-1)
*#define stdout (-2)
*#define stderr (-3)
*#define EOF    (-1)
*#define YES      1
*#define NO       0
*#define NULL     0
*#define EOL     10
*#define FF      12
*#define BS       8
*int row,col;
ROW BSS 2
COL BSS 2
*main()
 DEF MAIN
MAIN
*{ while(1)
C$2
 LI 8,1
 ABS 8
 JNE $+6
 B @C$3
*  { row=0;
 CLR 8
 MOV 8,@ROW
*    putchar(FF);
 LI 8,12
 BL 15
 BL *12
 DATA PUTCHA
 INCT 14
*    while(++row<25)
C$4
 INC @ROW
 MOV @ROW,8
 BL 15
 LI 8,25
 BL @C$LT
 ABS 8
 JNE $+6
 B @C$5
*     { col=6;
 LI 8,6
 MOV 8,@COL
*       while(++col<35)
C$6
 INC @COL     ---more--->
```

TI WRITER FORMATTER:

If you LIST a program to disk, and then use TI Writer Formatter to print it - be prepared for problems as concatenation commands (&) are translated by the formatter to be overstrike commands- messed up listing!

You can change formatter to use characters OTHER than & or @ as follows:

Use a sector editor to look for (hex) 23 21 40 26 - yes, the 40 stands for the @ symbol and the 26 stands for the & symbol. Hex 40 is the same as decimal 64! Hex 26 is the same as decimal 38. Change these two words to any character you want.

For instance if you change the 40 26 to 60 5C then Formatter will print & and @ as normal characters but will treat ` as overstrike and \ as underscore commands! Courtesy Rick Cosmano, V.P.S.C.C.6, P.O.Box 21181, El Cajon, Ca, USA, 92021

TI WRITER ^^^^

```
 MOV @COL.8
 BL 15
 LI 8.35
 BL @C$LT
 ABS 8
 JNE $+6
 B @C$7
*       { locate(row,col):
 MOV @ROW.8
 BL 15
 MOV @COL.8
 BL 15
 BL *12
 DATA LOCATE
 AI 14.4
*         putchar(42):
   8,42
 BL 15
 BL *12
 DATA PUTCHA
 INCT 14
*       }
 B @C$6
```

```
C$7
*    }
 B @C$4
C$5
*    if(getchar()<1) break;
 BL *12
 DATA GETCHA
 BL 15
 LI 8.1
 BL @C$LT
 ABS B
 JNE $+6
 B @C$6
 B @C$3
* }
C$8
 B @C$2
C$3
*}
 B *13
 END
```

I recently had the good fortune to hear from a brand new TI owner who along with a TI console, had received my book. He was close enough to visit, and found news of current activity (even if mainly American) quite cheering. Hopefully now a new member, so welcome! Do keep your eyes peeled for new TI owners!

Now... after you use the TI Editor Assembler on this 9900 Source Code, you have object code!

So, to keep our record complete, lets take a look at what the Editor Assembler turns that 9900 source code into.

In this text we are listing also the source code concerned, so you can follow each item of object code.

The columns are:

```
LINE NUMBER| RELATIVE | VALUE    | 9900 SOURCE
OF SOURCE  | MEMORY   | IN THAT  | CODE
CODE       | LOCATION | LOCATION |
```

Note that in many cases one line of source code can produce more than one line of object code!

CONTINUED ON NEXT PAGE →

I now have a detailed article on the operation of the TI disk system, with sample source code. No space to reprint here, but the text is available on disk from the user library. There seems only a remote possibility that room can be found for it in TI*MES!

My priority for the missing items is:

1. TI Writer formatter
2. Disk control details.
3. Forth- program development
4. More on c99

Alwavs happy to change the order- why not vote for the next issues contents! Send in a letter with the above 4 topics in YOUR order of priority. Top vote winning article is guaranteed a place!

CORRECTION to Issue No13 re TI WRITER LF

CORRECTION: ISSUE 13: TI WRITER:
RAMBLES-Page 23-top of column two: Totally wrong, SORRY! When you use LF (LoadFile). to add text to existing text, anything you have written after the line you specify to "insert after" is moved down, it is NOT destroyed! No matter how much you use these programs you keep forgetting SOMETHING! Aaaaagh!

```
 99/4 ASSEMBLER
VERSION 2.S                                                    PAGE 0001
 0001                *c99opt v1.2
 0002                *c99 v2.0 (c) 1986 Clint Pulley
 0003                 REF C$CIND,C$DIV,C$REM,C$ASR,C$ASL,C$EQ,C$NE,C$LT,C$LE
 0004                 REF C$GT,C$GE,C$ULT,C$ULE,C$UGT,C$UGE,C$LNEG,C$SWCH
 0005                 REF GETCHA,GETS,PUTCHA,PUTS,LOCATE,POLL,TSCRN,EXIT
 0006 0000           ROW BSS 2
 0007 0002           COL BSS 2
 0008                 DEF MAIN
 0009                MAIN
 0010                C$2
 0011 0004 0208       LI 8,1
      0006 0001
 0012 0008 0748       ABS 8
 0013 000A 1602       JNE $+6
 0014 000C 0460       B @C$3
      000E 00A4'
 0015 0010 04C8       CLR 8
 0016 0012 C808       MOV 8,@ROW
      0014 0000'
 0017 0016 0208       LI 8,12
      0018 000C
 0018 001A 068F       BL 15
 0019 001C 069C       BL *12
 0020 001E 0000       DATA PUTCHA
 0021 0020 05CE       INCT 14
 0022                C$4
 0023 0022 05A0       INC @ROW
      0024 0000'
 0024 0026 C220       MOV @ROW,8
      0028 0000'
 0025 002A 068F       BL 15
 0026 002C 0208       LI 8,25
      002E 0019
 0027 0030 06A0       BL @C$LT
      0032 0000
 0028 0034 0748       ABS 8
 0029 0036 1602       JNE $+6
 0030 0038 0460       B @C$5
      003A 0086'
 0031 003C 0208       LI 8,6
      003E 0006
 0032 0040 C808       MOV 8,@COL
      0042 0002'
 0033                C$6
 0034 0044 05A0       INC @COL
      0046 0002'
 0035 0048 C220       MOV @COL,8
      004A 0002'
 0036 004C 068F       BL 15
 0037 004E 0208       LI 8,35
      0050 0023
 0038 0052 06A0       BL @C$LT
      0054 0032'
 0039 0056 0748       ABS 8
 0040 0058 1602       JNE $+6
 0041 005A 0460       B @C$7
      005C 0082'
 0042 005E C220       MOV @ROW,8
      0060 0000'
```

John Stocks has sent me a super Mini Memory program which draws hi res graphics FAST. Unfortunately it would take up a lot of room to list and a long time to key in, as John has used look up tables to replace the slooow trig functions, so he offers a tape to any member interested - please send at least return post and packing!

If you have a program which can plot in bit map mode or simulate it, here is a listing (for Myarc Ex Bas) which you can use:

```
100 CALL GRAPHICS(3)
110 FOR A=1 TO 9
120 FOR C=5 TO 8
130 FOR B=1 TO 6
140 FOR ANG=.05 TO 6.285 STEP .005
150 R=SIN(C*ANG)
160 Y=R*SIN(B*ANG)
170 ROW=91+85*Y
180 X=R*COS(A*ANG)
190 COL=130+85*X
200 CALL POINT(1,ROW,COL)
210 NEXT ANG
220 FOR DELAY=1 TO 300
230 NEXT DELAY
240 CALL GRAPHICS(3)
250 NEXT B :: NEXT C :: NEXT A
260 GOTO 100
```

JOHN STOCKS
11 STONEHILL ROAD
Roxwell
CHELMSFORD
CM1 4PF

Using the above, Myarc Ex Bas took 7 mins 23 secs, but by reducing the step in line 140 to .02 and using CALL DRAWTO to connect the points, for minimal loss of definition, the time was brought down to 1 min 57 secs.(for ONE pattern)
John's mini memory program takes around 20 minutes to draw all 216 patterns!!!

*---->more--->

CONTINUED......

```
 99/4 ASSEMBLER
VERSION 2.S
  0043 0062 068F   BL 15
  0044 0064 C220   MOV @COL,8
       0066 0002'
  0045 0068 068F   BL 15
  0046 006A 069C   BL *12
  0047 006C 0000   DATA LOCATE
  0048 006E 022E   AI 14,4
       0070 0004
  0049 0072 0208   LI 8,42
       0074 002A
  0050 0076 068F   BL 15
  0051 0078 069C   BL *12
  0052 007A 001E'  DATA PUTCHA
  0053 007C 05CE   INCT 14
  0054 007E 0460   B @C$6
       0080 0044'
  0055            C$7
  0056 0082 0460   B @C$4
       0084 0022'
  0057            C$5
  0058 0086 069C   BL *12
  0059 0088 0000   DATA GETCHA
  0060 008A 068F   BL 15
  0061 008C 0208   LI 8,1
       008E 0001
  0062 0090 06A0   BL @C$LT
       0092 0054'
  0063 0094 0748   ABS 8
  0064 0096 1602   JNE $+6
  0065 0098 0460   B @C$8
       009A 00A0'
  0066 009C 0460   B @C$3
       009E 00A4'
  0067            C$8
  0068 00A0 0460   B @C$2
       00A2 0004'
  0069            C$3
  0070 00A4 045D   B *13
  0071             END
  0000 ERRORS
```

Just a little more follows....

STOP PRESS STOP PRESS

THE MYARC EXTENDED BASIC VERSION 2.1 NOW TO HAND IT RUNS TI BASIC & EXT BASIC DONT MISS THE NEXT ISSUE FOR MORE DETAILS

Funnelwriter and Myarc RAM card:
I mentioned earlier that there were problems in running Funnelwriter from the Myarc RAM card - in that when you RUN "RD.LOAD" no matter what disk number the ram card is emulating, Funnelwriter always tried to run from Disk Number One, instead of remembering which drive the LOAD file came from.
If you loaded the UTIL file from DM1000 then there was no problem, Funnelwriter thereafter remembered its disk - but not if you entered from Extended Basic.
I am delighted to say that Tony McGovern has now provided the solution in a minor rewrite of Version 3.3, with an additional parameter passed in the original LINK. Funnelwriter is now fully compatible with the Myarc card - and a real delight to use, with instant switching between editor, formatter, compiler, disk manager, assembler... pure heaven!

NOT WANTED...
It is apparent that nobody exept me, is interested in using the superb graphics programs we have. I have asked for samples of work in the last two issues, and NOTHING has come in.
Apparently very few UK owners are writing programs these days - a few isolated non-members have sent in programs for Stainless Software to publish, and John Stocks has submitted his superb mini memory graphics program, but apart from that- silence.
And absolutely nobody is interested in a UK history of the 99/4, either reading or advising.
Come to that - I've had very little TI related post of ANY sort recently. Is anyone else using the machine? Hello????

RAMBLES UPDATE- Issue 13 page 8, THE MINE:
After much pulling out of hair I have now established what this RADIX program is all about, and can now recommend it without reservation (EXCEPT that you need some instructions!). There are in total 25 different screens to battle your way through, but you have a limited capability of choosing the order you go through them. It is worth fifteen pounds

HERE IS SOME NEWS FROM A RARE AND REMOTE SOURCE:

LOGO IN CHINA

RELATIVELY new in China are LOGO programs (usually used to introduce beginners to programming in a lively and graphic way — one can draw designs by giving program commands to a turtle, which moves around the screen leaving a trail in its wake). Two Shanghai teenagers made very creative use of LOGO in a software piece called "The Evolution of Chinese Characters." The turtle is programmed to draw the different shapes of characters such as 山 (mountain) or 水 (water) as they originally appeared in ancient times, and then all their evolutions of form right down to the present.

How to use Chinese characters on computers is a problem that concerns all computer educators, particularly on such machines as the Apple II, which is famous for the amount of educational software written for it, but has very limited memory.

It's even more difficult to use Chinese characters on the LASER 310, with only 18 kilobites of memory. Two programs presented at the fair got around this difficulty, but require the user to generate his own character bank, which would of course be considerably smaller than the 6,000-character bank available on the Chinese Character System.

There's no doubt that computers and CAI software have a bright future in Chinese education, though there is still a long way to go. According to Liu Tianji, head of the child and youth affairs department of China's National Science and Technology Association, early last year his organization sent 10 LASER computers to Tibet for student use. That brings the number of provinces, autonomous regions and major cities having at least some computers for young people up to 30.

CHINA RECONSTRUCTS

And finallv, to round this article off, here is how it looks on disk

```
000A6         A0000A0002A0004B0208B0001B0748B1602B0460C00A4B04C87F345F         0001
A0012BCB08C0000B0208B000CB068FB069CB0000B05CEB05A0C0000BC2207F2CDF             0002
A0028C0000B068FB0208B0019B06A0B0000B0748B1602B0460C0086B02087F319F             0003
A003EB0006BC808C0002B05A0C0002BC220C0002B068FB0208B0023B06A07F2F1F             0004
A0054C0032B0748B1602B0460C0082BC220C0000B068FBC220C0002B068F7F2F8F             0005
A006AB069CB0000B022EB0004B0208B002AB068FB069CC001EB05CEB04607F2ABF             0006
A0080C0044B0460C0022B069CB0000B068FB0208B0001B06A0C0054B07487F30DF             0007
A0096B1602B0460C00A0B0460C00A4B0460C0004B045D7F638F                            0008
40000C$CIND40000C$DIV 40000C$REM 40000C$ASR 40000C$ASL 7F3D7F                  0009
40000C$EQ  40000C$NE  30092C$LT  40000C$LE  40000C$GT  7F4C3F                  0010
40000C$GE  40000C$ULT 40000C$ULE 40000C$UGT 40000C$UGE 7F40AF                  0011
40000C$LNEG40000C$SWCH30088GETCHA40000GETS  3007APUTCHA7F2D2F                  0012
40000PUTS  3006CLOCATE40000POLL  40000TSCRN 40000EXIT  7F30EF                  0013
50004MAIN  7FD6BF                                                              0014
:       99/4 AS                                                                0015
```

It doesn't take up much room does it! Remember though that you MUST load the support file CSUP after loading this, before you can run it! and the start name is appropriately, START.

On the disk file, each "word" of code is separated by a letter A, B or C - these tell the computer how to use the code.

It is possible to turn this disk file back into 9900 source code using a disassembler such as the Universal Dissassembler by Rene LeBlanc or the DisKassebler from Millers Graphics. Note however that although a dissassembler may insert labels, they will not be as helpful as the ones originally used, as these are lost in the assembly process.

Thats all for now. Stephen

ENJOY.

TIGER

AN RLE PICTURE

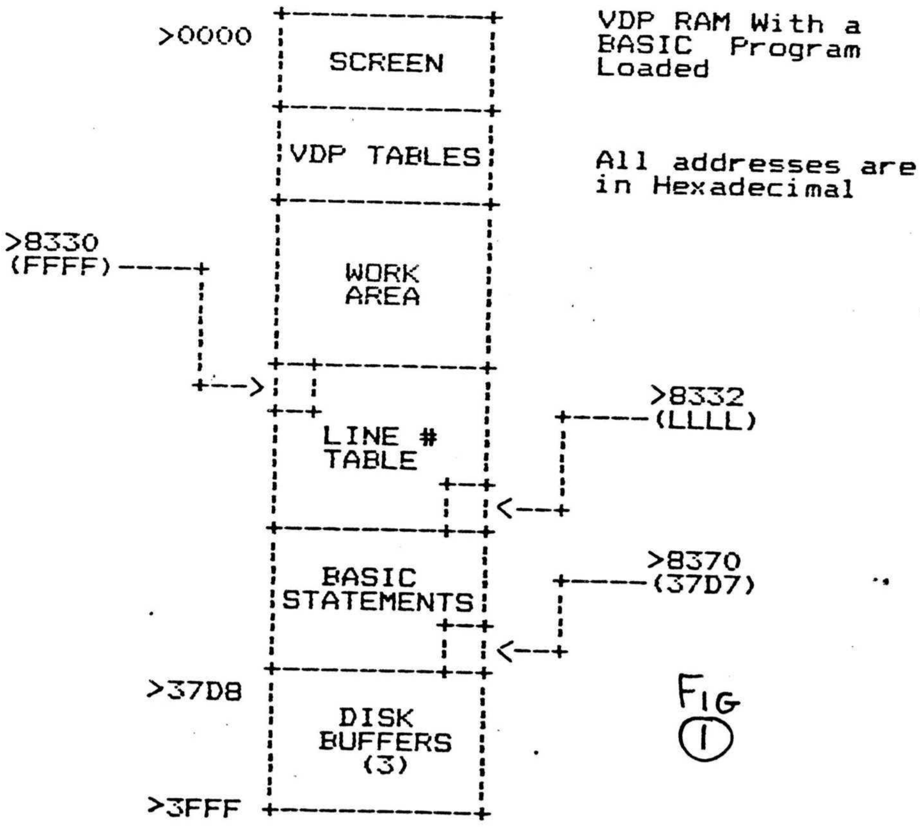
A LOOK AT BASIC PROGRAMS
by R. A. Green

In this article we will have a look at the structure of BASIC programs. To begin with, we will look at a TI BASIC program in VDP RAM just after it has been loaded via the OLD command. We assume that a disk is connected to the system and that the normal 3 disk buffers are allocated.

There are several well known pointers in CPU RAM to the BASIC program. The important ones for our purposes are:

>8370 contains the highest available address in VDP RAM (i.e. >37D7 with the normal 3 disk buffers)
>8330 contains the VDP RAM address of the start of the BASIC line number table
>8332 contains the VDP RAM address of the last byte of the BASIC line number table.

A picture of VDP RAM with our BASIC program loaded is shown below.



VDP RAM With a BASIC Program Loaded

All addresses are in Hexadecimal

FIG ①

Each entry in the line number table is 4 bytes. The first word (2 bytes) is the BASIC line number. The second word is the VDP RAM address of the start of that statement. The statement numbers in the line number table are in reverse order. For example, if we had a 3 statement BASIC program loaded, the line number table and statement area might look as shown in Fig ② →

The BASIC statements are not in the form that you typed, but are "tokenized". In the tokenized form, all BASIC keywords, operators and puncuation are replaced by special one byte tokens. Each BASIC statement is preceeded by a one byte length and the last byte in each statement is always a byte of zeros. The line or statement numbers for the BASIC statements are not part of the statement, they exist only in the line number table.

BASIC always uses the line number table to find a statement. The statements in the statement area are more or less in random order. Actually, they are in the order in which they were typed (including any changes made), the first statement entered being at the highest VDP RAM address.

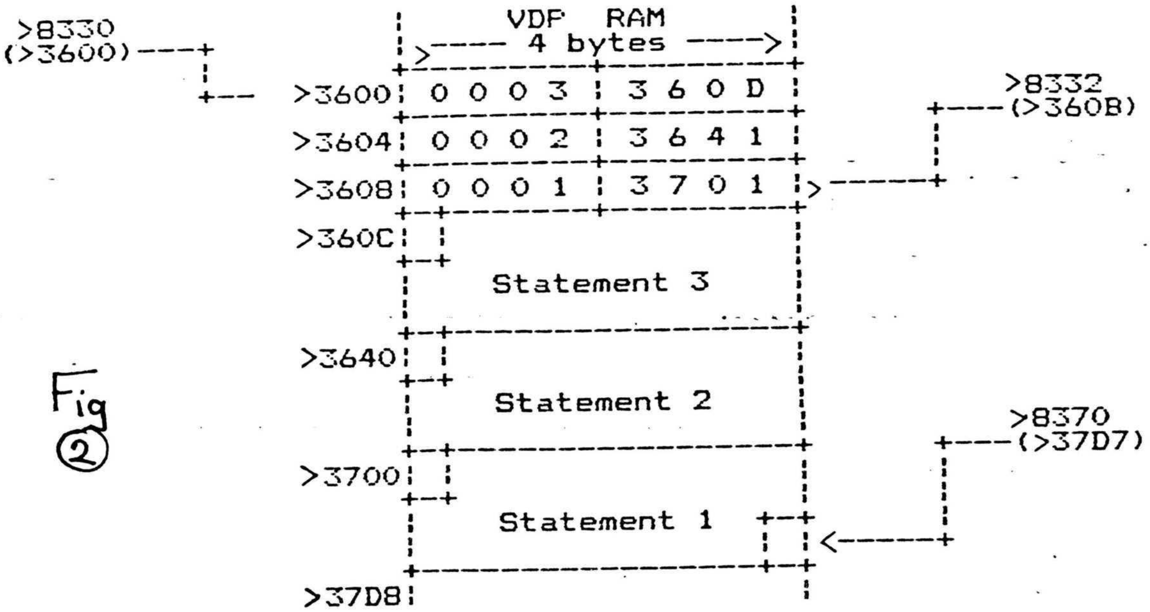


Fig ②

Now, if you were to save this program to disk (or cassette), BASIC adds 4 words in front of the line number table and writes it out just the way it is in VDP RAM (that's why you hear PROGRAM type files sometimes called memory image files). The four words added are:

1. A check word which is the "exclusive or" of words 2 and 3.
2. The value from >8332, the VDP RAM address of the last byte in the line number table.
3. The value from >8330, the VDP RAM address of the first byte of the line number table.
4. The value from >8370, the VDP RAM address of the last byte of the statement area.

So that, our sample program shown before would look on disk or cassette as shown below.

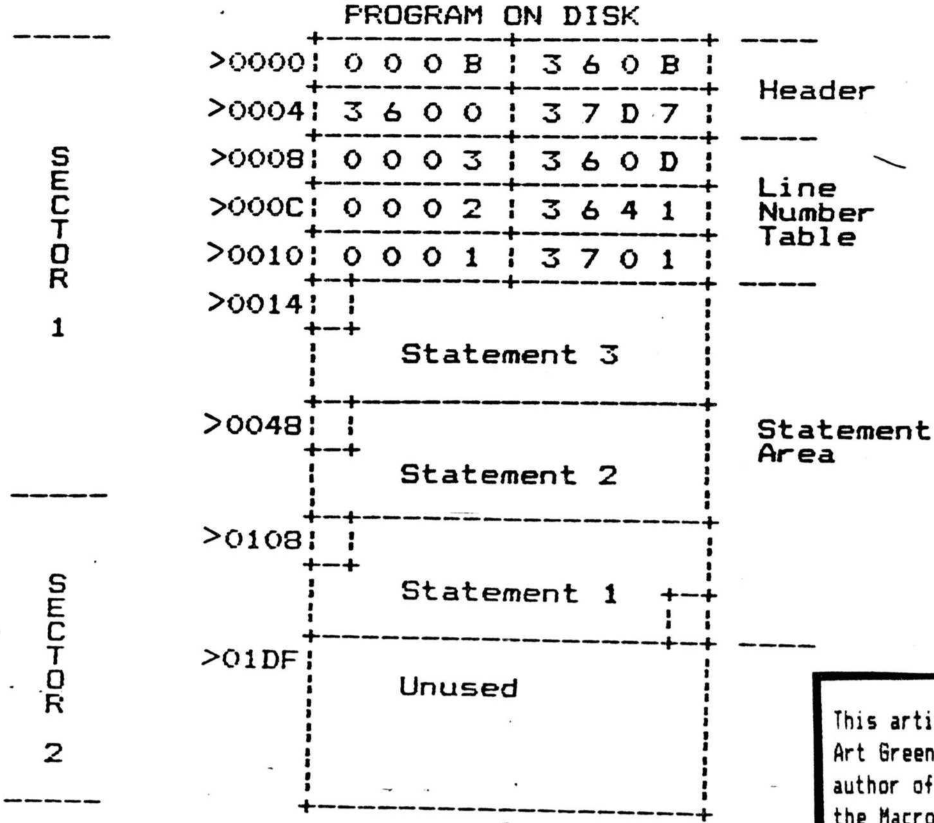


Fig ③

This article is by Art Green - the author of RAG-MAC, the Macro-Assembler NOW available!

CONTINUED→

To calculate where a statement is in the program on disk you have to "relocate" the values in the line number table. The values in the header give us the relocation factor.

Remember that the third word in the header is the VDP RAM address of the start of the line number table. In the file the line number table begins at byte 8. So, if we use the formula:

VDP RAM Address - Word 3 + 8

we can relocate any value. Let's try this with our example program. Let's calculate where in the file statement 2 begins. From the line number table, the VDP RAM address of statement 2 was 3641. So using the above formula, we have

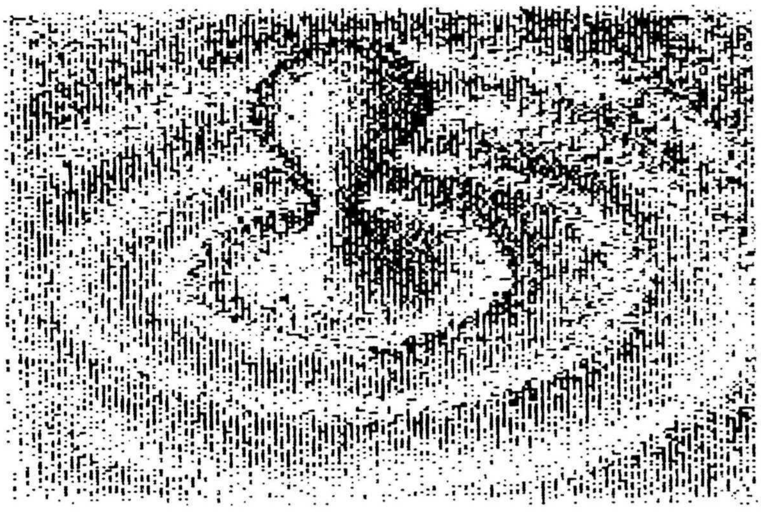
>3641 - >3600 + 8 = >0049

and statement 2 begins at byte >0049 in the file. The picture above shows the length byte which preceeds the statement at byte >0048.

Extended BASIC uses exactly the same structure as TI BASIC except in the case where the Extended BASIC program is to large for the "memory image" format. Even if you have the memory expansion, where Extended BASIC puts your program, it is always loaded and saved through VDP RAM just as though TI BASIC were doing it. In the move to the expansion memory from VDP RAM Extended BASIC relocates the values just as we did above. BASIC programs are always loaded into the high end of the memory expansion so that the VDP RAM address >37D7 corresponds to the memory expansion address >FFE7. From a disk dump of an Extended BASIC program you should now be able to calculate the memory expansion address of any BASIC statement.

As a consequence of this structure of BASIC programs, and of how the statements are accessed, neither BASIC is really concerned with what is in the statement area of a program. They only want entries in the line number table to point to BASIC statements. If some way were devised to insert a memory image asembler program between the line number table and the statement area (and relocating the VDP RAM addresses appropriately), then Extended BASIC would load the assembler program along with the BASIC program into the memory expansion ready to run. A great idea for a fast loader.

Are you still using a single sided disk drive? If you are you MAY be interested to know that commercial FLIPPY DISKS are now available retail in reasonable quantities...
DISKING (who also supply many other good disk related products) can now supply disks of DSDD quality in a flippy jacket, so that you can use both sides as single sided disks.
Now... there has been a lot of comment on flippies, but remember that all the INFOCOM adventure disks are flippies (or reversibles as they are also known), so they can't be all that bad!
These disks are called BANANA disks (lovingly grown by Disking...) and they have the Disking reference 2REV. DISKING supply them at the most reasonable price of: £8.65 for ten disks, plus postage, plus VAT on the total cost including postage. Postage is 95p for one pack of ten disks, £1.90 for two packs, £2.65 for three packs - then refer to their adverts! They take Access and Visa, telephone 0428 722563, or write:
DISKING, Freepost, Liphook, Hants, GU30 7BR
They respond by return, but as FREEPOST introduces up to a week's delay, you may prefer to pay the postage (outside the UK you have no choice) so write to them at:
DISKING, LIPHOOK, Hants, GU30 7EJ



2022- The decades have shown a much higher failure rate in flippy disks compared to none-flippies. The original caveats regarding longevity proved correct. And we have lost many TI programs due to the false economy of flippies. ss

CARTRIDGE REVIEWS ATARI

ATARISOFT REVIEWS By Edwin Armstrong

None of the Atari modules have speech and they all plug into the normal cartridge slot.

The reviewing system

If you do not want to read the whole review then you can see at a glance what the game is like by looking at the ratings for the 6 categories. Each category is given a number between 0 and 10 which is explained below.

The categories

Addictiveness--10=you are glued to your television for hours on end.
0=you switch off after 1 go.

Playability - Does your first go last hours or is it game over within 5 seconds whatever you do.
10=in between the two with lots of skill levels ,1 or 2 players, joystick or keyboard,pause and restart options.

Graphics - what the screen looks like. Example-Buck Rogers would score a 10.

Sound - 10=you play the game just because of the music and sound effects.
0=you have to turn the volume right down.
Example-Slymoids would score a 10.

Overall - An average of the last 4 categories.

Black and White - What the game looks like on a Black and White television
10= Everything perfectly visible.
0= The game is totally unplayable.

1.DIG DUG

You are Dig Dug a small kind of round,white monster who is armed with a jagged blue laser. The aim of the game is to kill all the other monsters before they kill you. The game begins with Dig Dug in the middle of the screen. Dig Dug can burrow around by moving the joystick and fire his laser by pressing the fire button.

The screen is divided horizontally into 5 parts,1 part above ground,4 parts below. The 4 parts below ground contain between 4 to 7 monsters of 2 different types called pooka and Fygar who move backwards and forwards in their small burrows trying to escape. Fygar is a green dragon that breathesfire and if the fire or a monster touches you you will loose a life. Pooka is a roung grey monster. Also under ground are 3 or 4 rocks and if Dig Dug burrows under these they will fall and kill monsters if they are in the way.

To explode and kill a monster you need to pump it up 3 times with your laser. Both types of monster will chase you whenever they can.If you do not destroy all the monsters within a certain time limit they will change into ghosts and travel through the un dug soil towards you.

After any two rocks have been dropped in any one screen a fruit will appear in the middle of the screen for 10 seconds. If you can get to it in time you will receive bonus points. Different ammounts of points are awarded for which part of the screen a monster was killed and in Fygars case whether he was killed from above or from the side. The last monster on any screen will flee to the top left hand corner of the screen turning into a ghost if it has to.

There is a nice little tune that plays throughout the game and the graphics are very colourful. You have the option of starting on any of the first seven screens and there is also a pause option.

A keyboard version would be nice and i think the game is too easy but apart from that quite a good cartridge.

Addictiveness	8
Playability	8
Graphics	6
Sound	8
Overall	7.5 (75%)
B & W	9

2.DEFENDER

You control the space ship Defender which is posted in outer space above a planets surface. Your job is to protect the humanoids.

The screen shows the planets surface to the bottom which is just a jagged white line and to the top is your number of lives and smart bombs,score and long range scanner.

You control your space ship with the joystick and the planets surface will scroll horizontally which everway you are flying. The fire button fires your laser. There are 10 humanoids along the planets surface to be protected by you from alien ships called Landers. Once you have destroyed all the aliens you receive bonus points for humanoids kept alive and go onto a new wave.

Alien ships -

Landers-Green ships that move quite slowly and fire white charges at you. They will try and kidnap humanoids and take them to the top of the screen where they will turn into mutants which are red,a lot faster and will also fire white charges at you.

Bombers-Dark green ships which swoop around the screen quite quickly laying mines that you cannot shoot and that will kill you.

Baiters-Thin white ships which only appear if you take a long time to finish a wave. Very fast and difficult to shoot.

Pods & Swarmers-A pod is like a blue asterisk that floats constantly downwards doing nothing much unless you fire on it in which case the pod disappears and 4 swarmers are released. Swarmers are small and red and swarm round you firing most of the time.

When there are no humanoids left the planet blows up and all the existing landers turn into mutants. On every fifth wave the planet and all 10 humanoids are replaced. Other things to help you are smart bombs which destroy everything on screen,long range scanners which shows the whole planet with everything going on in small and hyper-space which will take you to a random position somewhere else.Other options are 2 skill levels,1 or 2 players,Pause game and re-start game.

The sound effects and explosion noises are very good and it puts you in the mood for blasting aliens but there is not anything in the way of music or a small tune. The graphics are good but there is not much of them,only the planets surface and the aliens.

Compared to Parsec the explosions and scrolling are't really very good and as with Dig Dug the game is too easy and has no keyboard version.

But it should take you quite a while to get bored of Defender and if you like games like Ti Invaders and Parsec then Defender is definately a good buy.

Addictiveness	9
Playability	8
Graphics	5
Sound	7
Overall	7.25 (72.5%)
B & W	10

3.MOON PATROL

The aim of this game is to get to the end of the course in a good time. You control your moon buggy using the joystick or keyboard. You can go faster and slower,shoot infront and above and jump. Your moon buggy has 3 wheels which bobble up and down to the rough ground. The course is divided into 26 parts points labelled a to z and at points E,J,O,T and Z you are awarded bonus points for your time.

The screen shows 2 sets of mountains in the background both moving at different speeds which is quite effective. To the bottom of the screen is your moon buggy moving along the luna surface. Also at the bottom is your score,time,mumber of lives,a counter to show your position on the course,player no.,high score and 3 flashing lights which are supposed to warn you of things to come.

(Moon Patrol review continued)

Things you will come across are small and large rocks,rolling boulders, space plants,tanks and enemy cars which can either be jumped or shot. Small and large craters and mines which can be jumped only. Regular and crater U.F.O's which can be shot only.

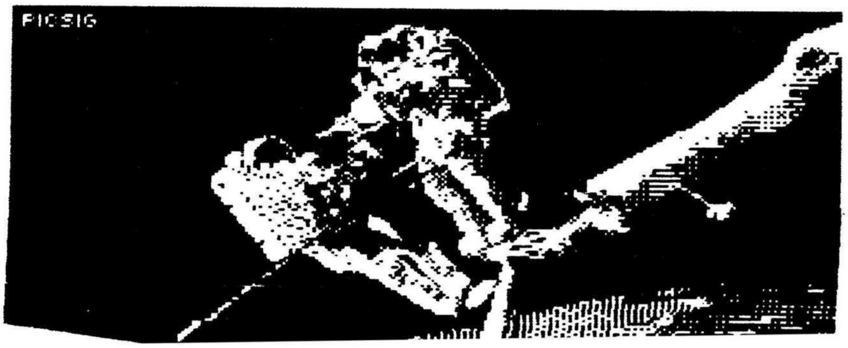
There are 3 types of U.F.O which appear in threes or fours and sometimes 2 or 3 types appear at the same time. The regular U.F.O's hover around and drop bombs to try and kill you. You can shoot these with your missiles but they are not worth points. Crater U.F.O's drop white bombs diagonally in front of you, often forming small craters. Tanks stay stationery on the ground often in awkard positions and shoot at you. Space plants stay in large holes and pop up and down especially when you are trying to jump them. Enemy cars come from behind and must be jumped and then shot.

The faster you are travelling the further you can jump. At points F-J and O-T the second background turns to temples and your car changes colour. The only trouble with this is that for half of the game the body of your car is invisible on a black and white Tv.

There is a simple but effective tune throughout the game. The graphics are very good and colourful and give the impression you are moving. The sound effects are good too. More and more things attack you as you get further along the course and you don't really know what is going to happen next. A very good game that you will want to keep on playing to find out what happens next. It has the usual features such as 1 or 2 players,pause,restart,2 skill levels, joystick or keyboard.

A very good copy of the arcade game and definately worth buying.

Addictiveness	9
Playability	9
Graphics	8
Sound	8
Overall	8.5 (85%)
B & W	5



JUNGLE HUNT

Your girl friend has been kidnapped by cannibals and taken deep into the jungle. You must find and rescue her.

You control your explorer by joystick or keyboard. The first stage involves jumping on and off swinging vines. If you misstime a jump then you fall into the undergrowth and loose a life. The second stage involves swimming across a crocodile infested river. There are 3 crocidiles on the screen at any one time. They swim along quite happily opening and closing their jaws. If you touch their jaws then you loose a life. There is also an air counter and if you don't come up for air every so often you also loose a life. You can kill the crocodiles for bonus points by thumping them in the mouth whem their jaws are closed.

The third stage involves running up a slope and jumping large and small bouncing boulders. If you are hit by a boulder then you land on your head and loose a life which is quite funny.

The last stage takes place in the cannibals camp and involves you jumping over 2 cannibals who are moving from side to side and waving spears and then rescueing your girl friend who is suspended over a big cooking pot by a rope that moves up and down.

While all this is going on you are being timed and the remaining time is then added on to your score. You then go on to the next level where things are faster and stages last longer. There is a little tune at the beginning and a few sound effects throughout the game.

The graphics are quite good but they are the same every time. The game is really all about timing your actions and its not that exciting. There are 3 skill levels,joystick or keyboard and pause options. This game is O.K but nothing brilliant. Not too difficult.

(Jungle Hunt review continued)

Addictiveness	7
Playability	8
Graphics	7
Sound	5
Overall	6.75 (67.5%)

BUGS!

Jungle Hunt-Try pressing pause just as your explorer jumps into the river from the last vine and see what happens to the screen. Also when you are on stages 3 & 4 if you press S and J together your man will move left twice as fast and if you press D and K together your man will move right twice as fast.

RATINGS ROUND UP

1st	Moon Patrol	85%
2nd	Dig Dug	75%
3rd	Defender	72.5%
4th	Jungle Hunt	67.5%

Most Addictive	- Defender & Moon Patrol
Most Playable	- Moon Patrol
Best Graphics	- Moon Patrol
Best Sound	- Dig Dug & Moon Patrol
Best On B & W Tv	- Defender & Jungle Hunt

HIGH SCORES

Here are my highest scores on these cartridges -

Moon Patrol	100,000+
Dig Dug	555,000
Defender	2,615,000
Jungle Hunt	20,000+

Can you do better ?

Availability

I bought these 4 Atarisoft cartridges from my local dealer. At the moment,out of these 4 cartridges as far as i know New Day have in Defender and Parco have in Defender and Jungle Hunt. The others don't seem to be available in the U.K.

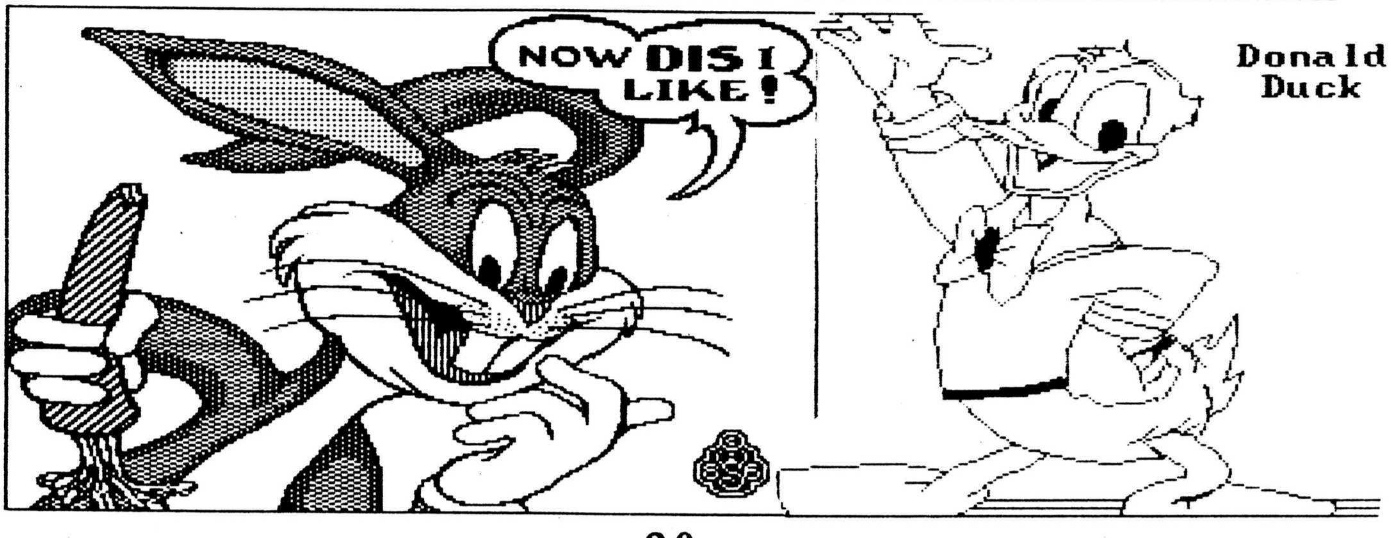
I have around 30 cartridges and lots of tapes and I would be happy to review any of them. Please write to me if you have any suggestions,If I don't have the game then I may be able to get hold of it.

Do you disagree with any of my reviews or ratings?,Have you found any bugs in any of these cartridges?,Have you beaten any of my high scores? If so then please write to me at the address below-

Next time (If there is a next time),among other things there will be a review of Shamus.

Hope you enjoyed the reviews.

Edwin Armstrong,
16 Home Close,Salph End,Renhold,Bedford,MK41 OLB.



IMPROVED UNRUNNABLE BASIC by Richard Heatn

[Ed. Note by Tom Freeman]

The "Unrunnable Basic" program by Stephen Shaw published in our July, 1986 issue (page 19) was relatively long. Richard took the time to analyze it and shorten it considerably, then was kind enough to send it to us. I have tried it out and it works well. You must type it in, save in MERGE format, then merge it into your Basic program that uses the unallowed character sets. I had not realized that already defined subprograms in XBasic could be redefined, but then can! To see what I mean, try typing in: 1ØØ FOR X=1 TO 16::CALL SCREEN(X)::NEXT X and run it. Now add the line: 2ØØ SUB SCREEN(A) :: PRINT A :: SUBEND , run again, and watch the difference!

To see how this program works, try the following in XBasic:

```
1ØØ CALL CHAR(144,"FF")
11Ø CALL CHAR(152,"8Ø8Ø8Ø8Ø8Ø8Ø8Ø8Ø")
12Ø CALL COLOR(15,16,5)
13Ø CALL COLOR(16,5,16)
14Ø CALL HCHAR(1,1,144,384)
15Ø CALL HCHAR(13,1,152,384)
16Ø GOTO 16Ø
```

Lines 11Ø through 13Ø will all give "BAD VALUE" error statements in XBasic. Run it also in Basic to see what happens. Now merge in Richard's program and see how it works exactly like Basic.

CAUTION: No error checking is done for the parameter values, so be careful you don't go outside the range. Also, in contrast to XB, you can define only one color set or character at a time.

```
1 ! "IMPROVED" UnRUNABLE BAS
IC PROGRAMS IN XBASIC by Ric
hard Heath, Tor. Ca.
 (Improved from Stephan Shaw
-John Behnke program.)
2 ! Enables you to run TI BA
SIC in 32K XB using all 16 c
haracter sets.
3 ! MERGE THIS PROGRAM INTO
YOUR BASIC PROGRAM. (Alterna
te method: Run this
prgm thru 3268Ø, then run yo
ur prgm with 3269Ø-3271Ø inc
luded in it.)
4 ! THIS NEXT LINE IS VITAL:
5 CALL LOADUTIL
6 ! TI BASIC PROGRAM FITS IN
 HERE. (Alternate method: 6
RUN "DISK1.YOURPGM+3")
7 ! ..............
8 ! ..............
326ØØ SUB LOADUTIL :: CALL I
NIT :: CALL LOAD(8194,37,194
,63,24Ø)
3261Ø CALL LOAD(16368,8Ø,79,
67,72,65,82,37,58,8Ø,79,75,6
9,86,32,37,168)
3262Ø CALL LOAD(953Ø,2,224,3
7,2Ø,3,Ø,Ø,Ø,2,5,48,48,2,6,3
7,2,2Ø5,133,2,134,37,17)
3263Ø CALL LOAD(9552,17,252,
4,192,2,1,Ø,1,2,2,37,1,2,3,1
8,Ø,212,131,4,32,32,2Ø)
3264Ø CALL LOAD(9574,2Ø8,4,9
,8Ø,2,32,3,Ø,2,1,37,2,2,2,Ø,
8,2,7,11,Ø,2,8,7,Ø,193)
3265Ø CALL LOAD(9599,1,192,1
93,193,18Ø,97,133,145,135,21
,1,113,136,6,198,145)
3266Ø CALL LOAD(9615,135,21,
1,113,136,21Ø,7Ø,1Ø,198,177,
137,22Ø,198,2,131,37,1Ø)
3267Ø CALL LOAD(9632,17,24Ø,
4,32,32,36,16,6,2,224,37,2Ø,
3,Ø,Ø,Ø,4,32,32,32,4)
3268Ø CALL LOAD(9653,192,216
,Ø,131,124,2,224,131,224,4,9
6,Ø,112):: SUBEND
3269Ø SUB CHAR(A,A$):: CALL
LOAD(95ØØ,A):: CALL LINK("PO
CHAR",A$):: SUBEND
327ØØ SUB COLOR(A,B,C):: CAL
L LOAD(9492,8,15+A,(B-1)*16+
C-1)
3271Ø CALL LINK("POKEV"):: S
UBEND
```

<cont. next page>

For those that are interested in how the program works, I have provided "source code," produced with DISkASSEMBER. The only changes to the output were to substitue the names of the XBasic utilities for their addresses in the BLWP's, and to place in the correct locations the names (for CALL LINK) poked into the REF/DEF table in line 3261Ø. It should be a good exercise for you to puzzle out what Richard is actually doing. Of course you need to look at the XB program as well, to see what it is poking into memory. [HINT for the confused: the MSB of R4 is at >251C, decimal equivalent 95ØØ]

```
       DEF   POCHAR,POKEV
       REF   STRREF,VMBW,VSBW
       AORG  >253A
                                                                        253A
POCHAR LWPI  >2514               >Ø2EØ,>2514          '..%.'            253A
       LIMI  >ØØØØ               >Ø3ØØ,>ØØØØ          '....'            253E
       LI    R5,>3Ø3Ø            >Ø2Ø5,>3Ø3Ø          '..ØØ'            2542
       LI    R6,>25Ø2            >Ø2Ø6,>25Ø2          '..%.'            2546
AA     MOV   R5,*R6+             >CD85                '..'              254A
       CI    R6,>2511            >Ø286,>2511          '..%.'            254C
       JLT   AA                  >11FC                '..'              255Ø
       CLR   RØ                  >Ø4CØ                '..'              2552
       LI    R1,>ØØØ1            >Ø2Ø1,>ØØØ1          '....'            2554
       LI    R2,>25Ø1            >Ø2Ø2,>25Ø1          '..%.'            2558
       LI    R3,>12ØØ            >Ø2Ø3,>12ØØ          '....'            255C
       MOVB  R3,*R2              >D483                '..'              256Ø
       BLWP  @STRREF             >Ø42Ø,>2Ø14          '.  .'            2562
       MOVB  R4,RØ               >DØØ4                '..'              2566
       SRL   RØ,5                >Ø95Ø                '.P'              2568
       AI    RØ,>Ø3ØØ            >Ø22Ø,>Ø3ØØ          '. ..'            256A
       LI    R1,>25Ø2            >Ø2Ø1,>25Ø2          '..%.'            256E
       LI    R2,>ØØØ8            >Ø2Ø2,>ØØØ8          '....'            2572
       LI    R7,>ØBØØ            >Ø2Ø7,>ØBØØ          '....'            2576
       LI    R8,>Ø7ØØ            >Ø2Ø8,>Ø7ØØ          '....'            257A
       MOV   R1,R4               >C1Ø1                '..'              257E
       MOV   R1,R3               >CØC1                '..'              258Ø
AD     MOV   *R4+,R6             >C1B4                '..'              2582
       S     R5,R6               >6185                'a.'              2584
       CB    R7,R6               >9187                '..'              2586
       JGT   AB                  >15Ø1                '..'              2588
       SB    R8,R6               >7188                'q.'              258A
AB     SWPB  R6                  >Ø6C6                '..'              258C
       CB    R7,R6               >9187                '..'              258E
       JGT   AC                  >15Ø1                '..'              259Ø
       SB    R8,R6               >7188                'q.'              2592
AC     MOVB  R6,R9               >D246                '.F'              2594
       SLA   R6,12               >ØAC6                '..'              2596
       AB    R9,R6               >B189                '..'              2598
       MOVB  R6,*R3+             >DCC6                '..'              259A
       CI    R3,>25ØA            >Ø283,>25ØA          '..%.'            259C
       JLT   AD                  >11FØ                '..'              25AØ
       BLWP  @VMBW               >Ø42Ø,>2Ø24          '.   $'           25A2
       JMP   AE                  >1ØØ6                '..'              25A6
POKEV  LWPI  >2514               >Ø2EØ,>2514          '..%.'            25A8
       LIMI  >ØØØØ               >Ø3ØØ,>ØØØØ          '....'            25AC
       BLWP  @VSBW               >Ø42Ø,>2Ø2Ø          '.   '            25BØ
AE     CLR   RØ                  >Ø4CØ                '..'              25B4
       MOVB  RØ,@>837C           >D8ØØ,>837C          '...¦'            25B6
       LWPI  >83EØ               >Ø2EØ,>83EØ          '....'            25BA
       B     @>ØØ7Ø              >Ø46Ø,>ØØ7Ø          '.`.p'            25BE
SLAST  END
```

TI99 SYSTEM ON THE MOVE

Traveling with TI
or
Can It Be Done Without Risk To Life Limb or Floppy Disk

by Robert L. Wesslet

Submitted by Terry Atkinson. Reprinted with permission from TINEWS (TEXNET) on the Source.

Articles by Robert L. Wessler, author of TRIVIA99ER.

There are several good reasons for wanting to take your computer with you on a trip. Keeping track of Travalers Checks is probably not one of them. The 99/4A is a sturdy machine, and with a few precautions, it may be taken with little trouble or risk.

Before packing up the machine and sticking it in the trunk, you must decide exactly why you want to take your computer with you. If it is a part of your business and you wish to do some work while you are away, then your computer can be of invaluable help on the trip. If you want to show grandma your new toy, however, it may not be worth the trouble. OK, let's decide if we're going to take our computer on vacation. We don't want to do any business work on vacation this year, so we will be using it strictly for pleasure. Are we going to be staying in motels the entire trip? If so, it may be more trouble than it's worth. Each time you leave the motel, you will probably want to pack the computer and put it away. Just like you don't leave money lying around in your room, neither do you leave expensive electronics. Most motels are safe, but one shouldn't take chances. Besides, if you are traveling and staying in a different motel every night, there are probably going to be plenty of things to do besides computing.

If, however, you are going to have a stationary base with a secure room, a computer can be fun and usefull to have around. Now, we must decide what we will use the computer for.

Well, it looks like we're going to Mom and Dads for a family reunion. We'll have our old room back, and we can lock the younger kids out while we're not around. (This part is important. Many younger children have been around computers all their lives and are very familiar with them. Others, on the other hand, have an Atari Game machine and can quickly ruin a computer while trying to make it work like the game machine they're used to.) Let's take our computer along and entertain the family. What type of games or programs you plan to use should determine how much computer you should take. If you are only going to play a few cartridge based games, then the console and joysticks will be a good system for the trip. If you plan on playing TRIVIA 99er, Infocom adventure games, or using one of the disk based graphics programs to entertain the family, then a full system will be required. If there is a BBS or a Videotex service in the town you will be visiting, then a modem will also be required if you want to access it. If there's only one television there, and people are going to be watching the big game, I would suggest bringing along the monitor also. (It's hard to believe, but my grandmother would rather watch Lawrence Welk reruns than compute!)

We've now decided to take the computer, and how much of it to take. What do we do now? Well, let's pick out the software we want to take. If you have a good carrying case and a place to put it, then your whole library may be taken. Remember, however, that disks do not do well when left in hot cars. The disks should be in a case that will hold them firmly, and should be kept in the air conditioned cab of the car, not in the trunk. Make sure they are kept out of direct sunlight. If it is necessary to leave them in a car unattended for any length of time, it is advisable to take some common sense precautions.

Here are some things which I do. These are not necessarily the best or the only things to do. Use your own judgement when leaving disks unattended in hot autos.

If the car is only going to sit for 1 or 2 hours, setting them under the seat is usually pretty safe. If they are going to sit in the car all afternoon, I will usually put them in a small ice chest on the floor out of the sun. If it is unusually hot, I will even put ice in the ice chest to keep the disks cool. Make sure, however, that the disk holder is not resting in the water, and that the disks will remain dry.

Packing the Computer

We are now set to start packing. I hope you were thinking far enough ahead to keep all of your computer boxes. You were? Good! If not, we will get to you in a little bit.

Unplug the electricity from all of the peripherals and wait 2 minutes.

Now, unplug all of the peripherals from each other. (LEAVE ALL OF THE CARDS IN THE PE BOX. THEY WILL RIDE BETTER THERE, AND TAKE UP LESS ROOM.

Now, place the PE box in its' carton. It may be necessary to cut out part of the cardboard guide in the carton so that the cable and plugs coming out of the back of the PE box will fit.) Now, Roll up the cable and put it in the compartment in the back of the box. Put the electric cable there also. I always put all of my electric cables and monitor cables here also. This keeps them in one place and I haven't forgotten one since my first trip.

Now, you can look at the box and say "WHERE THE HELL AM I GOING TO PUT THIS?". Which brings us to space. I didn't bring up the subject of space before, because until now, we didn't have any idea how much room our little systems really occupy. Check the limitations of your trunk or back seat, and how much luggage you are going to have. If you really love your computer, you can leave the spare tire at home! (JUST KIDDING)! If

2022 "trunk" = American for "car boot" .... bb there isn't enough space, unpack your computer, and look forward to listening to Uncle Charles tell you about his surgery again. If you have enough room, you can continue your packing. If you didn't save the box that your PE box came in, go to the the nearest grocery store or container corporation, and find a box a little bigger than your PE box. If you have some pieces of styrofoam, a piece can be put on each side of the PE box to keep it from sliding around in the packing box. Newspapers will do, but you may want to put plastic around the PE box to keep the ink from getting all over it. The console will fit nicely in a small suitcase if you have lost the box it came in. The speach synthesizer will fit in with the console and may not be necessary for the configuration you have decided upon taking.

If you want to do some word processing such as writing letters to the folks back home, then you may want to bring your printer along also. Remember to pack all conecting cables and to bring computer paper. The paper may be purchased at the destination depending upon the room left in the car. If you plan on printing up mailing lables at the family reunion, be sure to bring pleanty of extras, because everyone is going to want a few personalized computer mailing labels to take home!

NOW, THE IMPORTANT PART. Pick out the cartridges you need to take with you. A small cassette tape carrying case does a nice job of transporting cartridges. There is very little in this world more useless than a 99/4A without the Editor Assembler or the Extended Basic cartridges. Believe me, I speak from experience. Having your entire disk library will not help if you left your cartridges at home.

If you are going to play arcade type games, make sure you have the joysticks. These may or may not fit in the box with the PE system. Once everything is packed, put it all together and see if you've got everything.

1 PE box and cables
2 Console
3 Cartridges
4 Disks (disk based software)
5 Modem (if necessary)
6 Printer (if necessary)
7 Monitor (if necessary)

From here on, you're on your own. Use common sense in handling your computer. If you're not sure of the electrical systems you will be using, use surge protectors for your equipment. Don't set your disks in the sun on a speaker on the rear deck of the car. If they don't melt from the sun, the magnetic speaker will erase it.

ONE LAST NOTE OF CAUTION! Before plugging in the PE box and turning it on, make sure all expansion cards are seated firmly inside. A loose card may cause an electrical arc which can permanently damage the card and/or the PE box. After this check, the system may be set up and run normally. (A periodic check of the cards at home is also a good idea! Remember to unplug the box and wait 2 minutes before making such a check.)

PROGRAMS FROM BRUM

I left things a bit late for this issue so I've put down a few things I have found out about the T.I. It's probably all been done before but some members may not have come across them. One requires XB/joystick, one requires XB/disk drive.

1.AUTOLOAD PROGRAM FROM DISK. (Disk drive and XB). Ever wanted to switch on (master title displayed),press any key to display main menu then press 2. (for extended basic) and your program loads and runs automatically.

You could then take this further, what if your 'autoload' program was a menu that listed all the other programs on the disk. You would have a "catalogue" of programs on that (each) disk. Further still ?. Build into your menu a key to select the program of your choice (press 1.for prog 1.) (press 2. for prog 2.) and so on. Then you could select your favourite game and press one key to "autoload" that program also. One step further ?. You could modify the programs on your disk so that, when you exit from one program you don't go into the command mode to rerun or load another program instead you go automatically back to the menu ready to select another program with one key press. This is how it's all done.

To make a program autoload simply name it "LOAD". (SAVE DSK1.LOAD). Then each time you select XB the program will auto load and run.

Obviously you can only name one program "LOAD" on each disk.

To make another program (selected from menu) auto run from one key press, follow this example. ((This is the program to save under the name LOAD.))

10 DISPLAY AT(2,9)ERASE ALL:"DISK MENU"::DISPLAY AT(4,2) "PRESS....NAME OF PROGRAM::DISPLAY AT(6,6):"1......PROG 1"::DISPLAY AT(8,6):"2...... PROG 2" ::DISPLAY AT(10,6):"3......PROG 3" .And so-on. After you have completed displaying your menu, program follows with 100 CALL KEY (0,K,S) :: IF S=0 THEN 100

110 IF K<49 OR K>57 THEN 100 ..Line 110 allows only keys 1 to 9 access through the call key. You should select your own parameters.

120 ON K-48 GOTO 140,150,160,170,etc. one line number for each program listed on the menu.

140 RUN "DSK1.PROG 1"
150 RUN "DSK1.PROG 2"
160 RUN "DSK1.PROG 3" And so-on.

((prog 1,prog 2 and prog 3 represent the names under which the programs were saved))

That will autoload those programs from the menu. But how do you get back to the menu when you want another program autoloaded ??.

You have to modify your ending options on each program.

E.G.Instead of "PLAY AGAIN (Y/N)".With a call key to respond to ASCII codes 89 and 78 etc.

Do something like ..PLAY AGAIN press P or RETURN TO MENU press M. Then the following call key would be ...IF K=80 THEN (start line)::IF K=77 THEN RUN "DSK1.LOAD" .(don't forget the quotes). You may need to include a CALL CHARSET before the RUN "DSK1.LOAD"if the program you are finishing with utilised CALL CHAR on some ASCII numbers that the new program will not re-utilise.This is because the characters will not reset as they do when a program is ended and you enter the command mode. Don't forget, all your programs must run in extended basic.

2.ICON SIMULATION. (XB and joysticks). I got the idea for this demo when I had the chance to "play" with an APPLE MACINTOSH computer a short while back. I have been on one or two APPLES and was not very impressed. The MACINTOSH impressed me greatly, it should, it costs around £2000. Apparently the mac uses ICON based software in that it displays small pictures (icons) of the utilities/ programs available and the user selects by moving an arrow displayed on the screen by using a "mouse". You "click the mouse" whilst the arrow is touching your icon choice to load your selection. You "click" by pressing a button on the mouse rather like the fire button on the joystick. In my version, the joystick represents the mouse complete with fire button.

```
10 REM ICON SIMULATION
20 DISPLAY AT(12,1)ERASE ALL:"SPEECH SYNTH' CONNECTED(Y/N)"
30 CALL KEY(0,K,S):: IF S=0 THEN 30
40 IF K=78 THEN 60 ELSE IF K=89 THEN SPEAK=1 :: GOTO 60
50 GOTO 20
60 DISPLAY AT(12,1)ERASE ALL:"ALPHA LOCK UP.USE JOYSTICK 1" :: DISPLAY AT(14,3):
"POSITION ARROW IN A BOX"
65 DISPLAY AT(16,4):"THEN PRESS FIRE BUTTON" :: DISPLAY AT(24,16):"press any key
"

66 CALL KEY(0,K,S):: IF S=0 THEN 66
70 CALL CLEAR :: CALL COLOR(9,2,3):: CALL COLOR(10,2,9):: CALL COLOR(11,2,11)::
CALL COLOR(12,2,5):: CALL COLOR(2,2,16)
80 FOR A=1 TO 5 :: READ B,B$,C$ :: CALL CHAR(B,B$)
90 DISPLAY AT(4*A,4):C$ :: NEXT A
100 READ B,B$,C$ :: CALL CHAR(B,B$):: CALL SPRITE(#1,37,2,180,180,0,0)
110 CALL JOYST(1,X,Y):: IF C>65 THEN X=X*5 :: Y=Y*5
120 CALL MOTION(#1,-Y,X):: DISPLAY AT(1,20)SIZE(-7):"ROW ";INT(R/8+1):: DISPLAY
AT(2,20)SIZE(-7):"COL ";INT(C/8+1)
130 CALL POSITION(#1,R,C):: IF R<65 THEN X=X*2 :: Y=Y*2
140 IF C<40 THEN 110 ELSE IF C>48 THEN 110
150 IF R<20 THEN 160 ELSE IF R>32 THEN 160 ELSE COL=4 :: NOTE=784 :: WORD$="GREE
N" :: GOTO 200
160 IF R<51 THEN 170 ELSE IF R>63 THEN 170 ELSE COL=7 :: NOTE=494 :: WORD$="RED"
 :: GOTO 200
170 IF R<84 THEN 180 ELSE IF R>96 THEN 180 ELSE COL=12 :: NOTE=392 :: WORD$="YEL
LOW" :: GOTO 200
180 IF R<116 THEN 190 ELSE IF R>128 THEN 190 ELSE COL=5 :: NOTE=247 :: WORD$="BL
UE" :: GOTO 200
190 IF R<148 THEN 110 ELSE IF R>160 THEN 110 ELSE COL=16 :: NOTE=196 :: WORD$="W
HITE" :: GOTO 200
200 CALL KEY(1,K,S):: IF K=18 THEN CALL SCREEN(COL):: CALL SOUND(100,NOTE,10)::
IF SPEAK=1 THEN CALL SAY(WORD$)ELSE 210
210 GOTO 110
220 DATA 97,"FF818181818181FF",a,104,"FF818181818181FF",h,112,"FF818181818181FF"
,p,120,"FF818181818181FF",x,43,"FF818181818181FF",+
230 DATA 37,"183973FE73391800",%
```

Derek Ford
37 Stotfold Road
Maypole
Birmingham
B14 5JD

////////Hairy/Harry's/Hearing////////

I think I am right in saying that there is now nobody left making a living in the UK out of the TI99/4a alone. It was inevitable that sooner or later people involved in the retail of hardware and software for the TI would have to diversify or quit. For my part, I arrived late in the day - to the delight of some and the annoyance of another - and have always had other strings to the bow. For the record, let me say here that I have the utmost regard for Howard Greenberg of Arcade Hardware, a gentleman, and that we have no hesitation in recommending each other to customers. He has been around since the start, and TI owners owe a lot to him for providing the most consistent, dedicated, and trustworthy service. Let him know. Now then, what of the future?

1) **Software.** Let me tell you a story. There was once a trader in pearls. Through various means (often fair) he set himself the goal of becoming the biggest pearl dealer in the land. Eventually he bought up nearly all the pearls in the world from the sea-god, so that he could name his price. People paid the price, because they wanted pearls. He became rich. Sadly for him, he suddenly had to sell a lot of pearls cheaply, because a few small traders were selling pearls at a more realistic price, and wouldn't go away. Now, because the price of pearls had dropped so much, when newer, rarer pearls were found, none of the dealers were keen to pay the sea-god for them, because they knew they would have difficulty selling them, for the people had got used to lower prices and would wait for them to drop. Then the people were very sad, because they could not get any pearls at all.........
So much for modules. Thankfully, one area that is likely to remain stable price-wise is on tape and disk. Although Basic and even Extended Basic have their limitations, at least new titles appear, and some programmers have been quietly busy developing their skills. I welcome more machine-code efforts for tape with Minimem, and on disk, although many people with disk-systems now spend their time gathering vast collections of juicy items from dubious origins. Never!?! Alternatively, the freeware concept is a really noble one. So noble, in fact, that it is entirely incompatible with human nature. Piracy and freeware are inadequate answers to the problem of software shortage. The only method that ever works is to pay talented people a worthy price for marketable programs. You don't need me to tell you that it is now a bit late in the day for the TI to enjoy that luxury. We do what we can, however. Realize though, that if we don't pay for software, there soon won't be any. That's just the way it is.

2) **Hardware.** It IS possible to buy brand new add-ons from America and Germany (I have been approached by Atronic, who are looking for a reliable distributor in the UK) but how many of you can afford or justify the cost? As far as secondhand is concerned, the concept is more popular than practical. Most folk expect to get something for nothing, but nobody wants nothing for something. In other words, the price sellers want for gear is about the same as buyers expect to pay. You don't need to be an accountant to work out that this is a rather less than lucrative prospect for the dealer. Cut him out! I hear you cry, wringing your hands. Fair enough, but the trouble is you need him more than you think. He's the mug who takes the responsibility of standing in the middle - paying for advertising, spending effort and money creating a market-place, taking risks, suffering the phone ringing AT ANY TIME of day or evening (ask Lindy) and for what? To be accused of ripping people off as often as not. Profit has become a dirty word, but without profit there is no living. No living, no dealer. Simple. Profit is not a sin. Exploitation is. Support the dealers you do still have, and respect the initiative and sacrifice they have made. Cut them out and you hack years off the TI market. Keep them on their toes maybe, and resist exploitation, but don't underestimate the role they play. They aren't in business as a hobby - they have mouths to feed, just like most of you. There - I've said it.

3) **4FRONT.** I am delighted with the way this venture has been received. Thanks must go to the various who have contributed, and also to those who have taken the trouble to mention it in the pages of TI*MES. Perfect it isn't, but blinking good value it is. Issue 2 is out now (deliberately late to be in vogue with most computer products[!]) and on disk uses 32k as well as ExBas to good effect. Looks like the USA market will be penetrated in a humble way, too.

So where the heck is it all leading? You tell me. I don't know, but one thing is certain: there are no losers. If you stick with the TI you've got a great machine. If you can afford to expand it you're improving a great machine whatever you have, and if it suits you, let nobody try to convince you otherwise (including me). Happy Computing.

Hairy Harry from Honiton (New Day)

T. I. TIPS

Jack Topham of the "Chicago TImes" newsletter and Bob Mack of the "TI Talk"(Jackson,MS) both "mine the gold" in other usergroups news letters.

TURBO TI:

1)From the MUSKEGON area UG comes a tip for speeding up your XBASIC programs. They suggest entering a CALL LOAD (-31878, N) where N is the number of sprites used from 0 to 28. Don't forget CALL INIT before the CALL LOAD. Try it.

2022: Repeat: This ONLY applies to Extended Basic Version 100 !!!!

2)From JACKSONVILE UG in ARIZONA comes this tip from Ralph Devine. Many of us use BASIC and XBASIC in a truly unstructured way. We build the program as we go. using RES to open up lines, etc. Our program LISTs in line No. order and "looks" structured. So, finaly when it is done and it RUNs OK, we are pleased. It seems that inside our 99/4A marvel our jumbled line order is preserved and time is lost looking for the next line to execute. Solution! Save the file using the XB MERGE option. Sure, it is slow, but it is saving the program in sequential line order in MEMORY IMAGE format. Type NEW then reload using MERGE then SAVE as a PROGRAM. This can make an appreciable difference in execution time. Give it a try!

3)A memory saver (less memory used, faster execution!) from TI TALK: If you add the following statement to your BASIC or EXTENDED BASIC program, you will save 2 bytes every time you use the numeral "1": line #1@=1 then inplace of 1 type the symbol "@". A little confusing but if memory is a problem, every little bit helps.

4)At the end of your EXTENDED BASIC program, you can "CALL INIT::CALL LOAD(-31962,255) and your console will search for a program called "LOAD" on drive #1. If such a program is not there, you will get a cursor on the XBASIC screen.

TI-WRITER TIP:

The NORTHWEST OHIO UG offered this TI TIP. To avoid a BUFFER FULL notice in the TI-WRITER, you should save the files and then use the SD command to see the file size. Since the BUFFER is full at 92 sectors, you can see right where you are. Thanx NW O UG!

MULTIPLAN TIP:

From the UPPER PINELLAS 99'er GROUP comes a MULTIPLAN TIP: Once you start scrolling, you can release the FCTN or CTRL key and just keep the ARROW key depressed. Try it!

PRINTER TIP:

The NE IOWA UG reported on using "PRINT USING" with the printer, an OBSCURE feature of the 99/4A. Brush up on PRINT USING and IMAGE in the XBASIC manual (Chapter 4,page 150 and 97) and go for it in printing to the printer. Just OPEN # and PRINT # and the rest is the same.

ERROR CHECK:

From TI TALK comes an data error print solution: When you get an error message like "DATA ERROR IN LINE XX", and the line referred to turns out to be a READ statement; you can isolate the error by temporarily changing the READ to PRINT. RUN the program and you will see the last data read correctly just before the error message!

CLASSIFIED ADVERTISMENTS

BARGAIN MODULES

TELEPHONE FOR AVAILABILITY BEFORE ORDERING 0273 503968..... MUSICMAKER £19.95 HOME BUDGET MANAGER £9.95 TE2 MODULE £19.95 ALPINER £9.95 CHISHOLM TRAIL £4.99 EXT BASIC £29.95, MINIMEM £34.95 PARSEC £10.95. ADVENTURE/PIRATE £10.95 . FROGGER £9.95. TI.INVADERS £5.95. CHESS £10.95. Q*BERT £10.95. MUNCHMAN £3.95.SOCCER £7.95 . EARLY LEARNING FUN £3.95. ADDITION/SUBTRACTION 1. £5.95. VIDEO GAMES 1 £4.95. BLASTO £4.95 >>>>>>>>>>> TELEPHONE 0273 503968 to CHECK STOCK AVAILABLE, at these prices stock will soon be exhausted.

JOYSTICKS. HOTSHOT — switched joysticks.Single £7.50 SPECIAL OFFER! TV AERIAL SPLITTER (SWITCHED) will prevent wear on TV socket and cable. Simply flick a switch to £2.25 +25 P+P.

Want a spare TI99/4a console? We have a number at £45.00 each

T.I.BOOKS.

KIDS AND THE TI99/4a - Datamost.

Learning Basic and the magic tricks of the TI can be great fun so why learn any other way. The drawings help you under- stand and remember the material so you'll be writing your own programs by Lesson 5 and forever after. Learn to write computer music, create computer graphics and drawings make your computer talk + ask questions create games and quizzes. Edit and debug programs fast organize your mind and think creatively and best of all it's fun. ITs NOT JUST FOR KIDS! 236 pages and spiral bound. £5.00 (original price £16.95)

GETTING STARTED WITH THE TI99/4A by Stephen Shaw....£4.50

These prices include postage and packing.

Send Order to CHS, 40 BARRHILL PATCHAM BRIGHTON BN1 8UF.

PLEASE ALLOW 28 DAYS TO PROCESS

****

TI*MES BACK ISSUES. The following issues ONLY are available *** 4,8,9,10, 11, 12 and 13. ONLY £1 each to members. TI99/4A EXCHANGE, BRIGHTON.

****

MINIMEM Conversion to Rechargeable Battery. Send Minimem and Crossed cheque £7.50, To- N.J.Petry, Tensal Technology, 15 PENRICE CLOSE, WORLE, W.S.M., Avon, BS22 9AH.

FOR SALE TI99/4a Computer, speech sythesiser, joysticks, cassette lead, ext basic, parsec, video games, maths learning modules, programming books, £100.00 ono - will split - Contact John Stoddart telephone HORSHAM (0403) 69177 after 6pm or weekends.

****

TI Users HEALTHLINE

TOO BUSY EARNING A LIVING not thinking how you can earn a substantial bonus to your present salary? If you are either RUN DOWN, OVERWEIGHT, SUFFERING FROM STRESS CANNOT SLEEP TOO WELL, NOT EATING THE RIGHT FOOD etc etc ..AND YOU WANT TO LIVE A HAPPY HEALTHY LIFE?... YOU CAN AND WILL AS SOON AS YOU ASK ME HOW. For a small outlay you will learn and experience a healthy new way of living, without doubt you will be living proof that you DO FEEL AND LOOK BETTER. Tell your friends they will be convinced. We run local courses in most areas so you have nothing to worry about. You will be a very happy and richer person. Moneyback guarantee. Telephone 0273 503968 and ask for TI USERS HEALTHLINE.. If you want to order your starter kit (worth over £40.00) send a cheque for £29.95 (made payable to TI Users U.K.) Dept H.L., 40 Barrhill, Patcham, BRIGHTON, Sussex, BN1 8UF. (overseas please get in touch with us. This operates in Canada, Australia, and USA. Early 1987 for Europe.) <<<<<<

****

ALPINE QUEST.

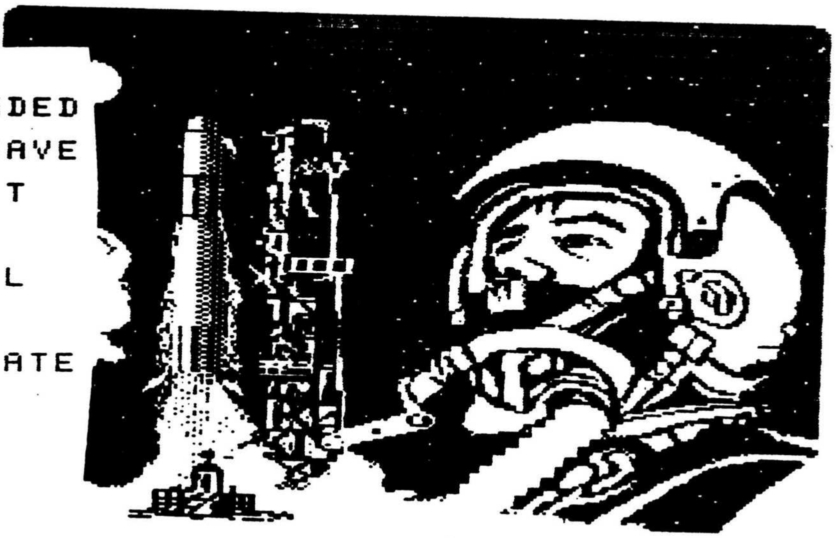
A new exciting graphic and text Adventure in TI BASIC by the author of Intrigue's "ADVENTUREMANIA II". Graphic version uses full memory capacity of the TI computer and text only version on side B runs 200% faster. Control two independent characters who must work together to complete the game. Watch the clock and your energy level and beware the avalanche as you explore the locality in search of all your stolen possessions. AVAILABLE FROM Mark McGurn, 26 Sandiford Road, Holmes Chapel, Cheshire CW4 7BU for only £3.95.

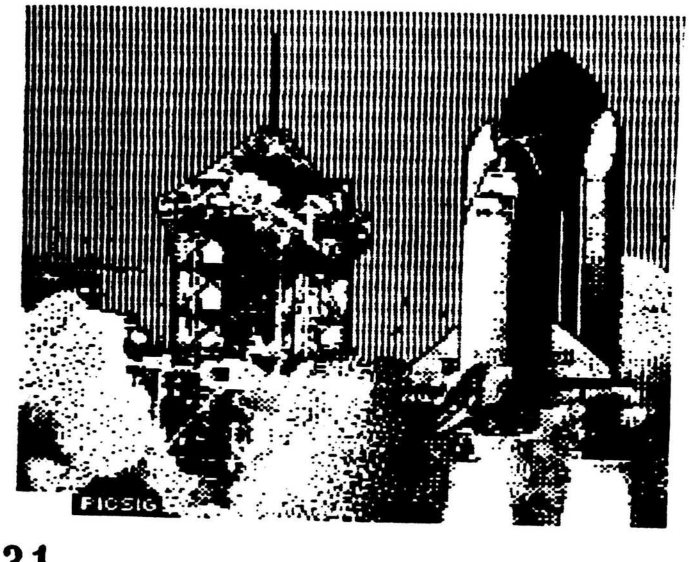
Software swap ring 0494 776180

ADVERTISE ON THIS PAGE FOR 5P A WORD. (MEMBERS ONLY)

Texas Instruments TI-99/4A Home Computer

HAVE YOU RENEWED OR EXTENDED YOUR MEMBERSHIP?? IF YOU HAVE FORGOTTEN THEN PLEASE DO IT RIGHT AWAY. REMEMBER ALL SUBSCRIPTIONS NOW RUN UNTIL MAY 1987 BUT ONLY IF YOU HAVE PAID THE APPROPRIATE MEMBERSHIP SUBSCRIPTIONS THIS IS YOUR ONLY REMINDER





THE GRAPHIC PICTURES HAVE BEEN PRODUCED USING FUNNELWRT PROGRAM DISK AND PIO PRINTER THE PROGRAM IS AVAILABLE AS U55 FROM TI*MES LIBRARY SEE PAGE 4 FOR DETAILS WE SHALL DEMONSTRATE THIS AT >>> COMPUTER JAMBOREE <<< COUNTS AS 15 PROGRAMS ALSO AVAILABLE FROM STEPHEN

```
==============================
 NATIONWIDE TI99/4A USERS
=WILL BE GETTING TOGETHER ====
==AT THE S.M.G. JAMBOREE  ====
==SATURDAY  NOVEMBER 8TH  ====
==MACKIE HALL MACKIE AVE  ====
     PATCHAM BRIGHTON
==============================
==DOORS OPEN 10AM TO 5PM===
==============================
 YOU WILL SEE THE TI99/4A
===PERFORMING AGAINST =========
 THE  PROFESSIONAL MICROS
==VIDEO-RTTY-BUSINESS AND ====
== DESK TOP PUBLISHING   =====
=THIS IS A SHOW WHERE WE  ====
INVITE THE SERIOUS TI99 USERS
TO SEE PRACTICAL APPLICATIONS.
  JAMBOREE IS ORGANISED BY
THE SUSSEX MICROCOMPUTER GROUP
     IN ASSOCIATION WITH
THE BRITISH COMPUTER SOCIETY
AND BRIGHTON,HOVE &DISTRICT
      COMPUTER CLUB
=COME AND SUPPORT YOUR TI=====
  THE PROFESSIONAL MICRO
==============================
```

TIWRITER TIPS FROM TIGERCUB

TI-WRITER SPECIAL CHARACTER MODE

© *by Jim Peterson*

A few newsletters recently have discussed the uses of the TI-Writer special character mode, which the manual did such a poor job of telling us about. Here are some of the things that can be done (on a

(CTRL U, FCTN R, CTRL U, SHIFT M, CTRL U, SHIFT H, CTRL U) Set the left hand margin at 8. Using the TI-Writer TAB for left margin creates problems.

(CTRL U, FCTN R, CTRL U, 4) Select the italic character set.

(CTRL U, FCTN R, CTRL U, 5) Cancel italics, return to standard character set.

(CTRL U, FCTN R, CTRL U, SHIFT G) Print in double-strike mode.

(CTRL U, FCTN R, CTRL U, 7, CTRL U, SHIFT B, CTRL U) Select the international character set for Germany ÄÖÜäöüß . Instead of SHIFT B, use A for England £, C for Denmark ÆØÅæøå, D for France £à°ç§éùè¨, E for Sweden ¤ÄÖÅÜäöåü, F for Italy §°çéàòèì and G for Spain ¡Ñ¿¨Ñ .

(CTRL U, FCTN R, CTRL U, SHIFT B,, CTRL U, SHIFT B, CTRL U) Set the print pitch for elite characters (12 per inch)

(CTRL U, SHIFT R, CTRL U) Restore pitch print to pica (10 cpi).

(CTRL U, SHIFT O, CTRL U) Set print pitch to condensed print.

(CTRL U, SHIFT R, CTRL U) Cancel condensed print.

(CTRL U, SHIFT N, CTRL U) Print in double-width mode (on one line only); to cancel it before end of line, use - DOUBLE (CTRL U, SHIFT T, CTRL U) cancelled.

(CTRL U, FCTN R, CTRL U, SHIFT H) Cancel double-strike.

(CTRL U, FCTN R, CTRL U, SHIFT E) Emphasized mode.

(CTRL U, FCTN R, CTRL U, SHIFT F) Cancel emphasized mode.

(CTRL U, FCTN R, CTRL U, SHIFT -, CTRL U, SHIFT A, CTRL U) Print characters with underline.

(CTRL U, FCTN R, CTRL U, SHIFT -, CTRL U, SHIFT A, CTRL U) Cancel the underlining.

(CTRL U, FCTN R, CTRL U, SHIFT S, CTRL U, SHIFT @, CTRL U) Print in superscript mode, which is always double-struck and unidirectional.

(CTRL U, FCTN R, CTRL U, SHIFT T) Cancels the superscript and unidirectional. Contrary to the manual, it obviously also cancels the double-strike.

(CTRL U, FCTN R, CTRL U, SHIFT S, CTRL U, SHIFT A, CTRL U) Print in subscript mode, unidirectional and double-struck.

(CTRL U, FCTN R, CTRL U, SHIFT T) Cancels subscript, unidirectional and double-struck.

Print italics ***double-struck and underlined,*** return to pica without underlining **but emphasized, then condensed, then double-width in pica; cancel double-width and go to elite type.**

Any combination of codes can be used, including those which move the print head or move the paper (line feeds). (CTRL U, SHIFT H, CTRL U) moves the print head back one space and can be used to overprint characters. (CTRL U, SHIFT M, CTRL U) placed before the end of the line will send the print head back to the beginning of the line and reprint it with the rest of the line.

Macro-commands (see page 120 of the manual) can even be used, although I have not found any worthwhile use for them. I have even managed to code in customized downloadable characters, such as my Tigercub emblem, but I cannot find any way to input CHR$(127).

Remember that these control characters are deleted by the printer and the line of print is shifted left by the number of spaces that they occupied. If you are inserting codes into text that has been prepared in columnar format, or right-justified through the formatter, the best way is to use CTRL O to get the open-square fixed-mode cursor; position it over the character to the right of the point of insertion; FCTN 2 to insert characters; tap the space bar as many times as there are characters to be inserted (be careful not to shove the end of the line into oblivion!); FCTN S to backspace and then fill the blanks with control characters.

The text file of this article was printed through the TI-Writer editor and was also printed by a separate print program; both printouts were identical. The margin setting confused the TI-Writer formatter completely; when this was deleted, it printed out properly except that some of the foreign character sets were repeatedly overstruck and defaced!

==== BASIC PROGRAM ====

I guess I've been neglecting those who don't have the Extended Basic module, so -

```
100 CALL SCREEN(16)
110 CALL CLEAR
120 PRINT TAB(8);"GREENSLEEV
ES": : : : : : : : : : : : :
 :"programmed by Jim Peterso
n"
130 DIM S(15)
140 FOR N=1 TO 12
150 READ S(N)
160 NEXT N
170 M$="42180099SABDC324E7DB
A51866991824004250A00DBC35A66
A5243C7E81994200A57E66BD3CA5
423C187E423CBD5A810099FFC3"
180 RANDOMIZE
190 FOR R=1 TO 12
200 CALL COLOR(R+1,1,1)
210 CALL CHAR(32+R*8,CH$&CH$
)
220 FOR T=R TO 25-R
230 CALL HCHAR(T,R,32+R*8,34
-2*R)
240 NEXT T
250 NEXT R
260 CALL SCREEN(2)
270 FOR R=1 TO 12
280 CALL COLOR(R+1,R+2,1)
290 CH$=SEG$(M$,INT(47*RND+1
)*2-1,8)
300 CALL CHAR(32+R*8,CH$&CH$
)
310 NEXT R
320 DATA 247,277,294,311,330
,370,392,440,494,523,554,587
330 DATA 2,5,5,4,7,5,2,8,5,3
,9,5,1,10,1,2,9,3,4,8,3,2,6,
3,3,3,1,1,5,3
340 DATA 2,6,1,4,7,5,3,5,2,1
,4,2,2,5,2,4,6,1,2,4,4,4,1,1
350 DATA 2,5,1,4,7,5,2,8,5,3
,9,5,1,10,5,2,9,5
360 DATA 4,8,3,2,6,3,3,3,3,1
,5,3,2,6,3,3,7,5,1,6,2,2,5,1
370 DATA 3,4,1,1,2,2,2,4,1,4
,5,1,2,1,5,6,5,1
380 DATA 2,12,9,2,12,7,2,12,
3,3,12,12,1,11,9,2,9,7
390 DATA 4,8,6,2,6,3,3,3,3,1
,5,5,2,6,3,4,7,5,2,5,3
400 DATA 3,5,5,1,4,4,2,5,5,4
,6,1,2,4,1,6,1,1
410 DATA 6,12,9,3,9,12,1,11,
8,2,9,7,4,8,6,2,6,3,3,3,3
420 DATA 1,5,3,2,6,2,3,7,5,1
,6,6,2,5,5,3,4,1,1,2,2,2,4,4
,6,5,1,1,1,5,7,5,1
430 FOR J=1 TO 223 STEP 3
440 READ T,A,B
450 GOSUB 530
460 FOR TT=1 TO T
470 CALL SOUND(-999,S(A),0,S
(B),7)
480 NEXT TT
490 NEXT J
491 FOR V=0 TO 20
492 CALL SOUND(-999,S(A),V,S
(B),V+7)
493 NEXT V
```

```
500 CALL SCREEN(INT(14*RND+2
))
510 RESTORE 330
520 GOTO 270
530 CALL COLOR(A+1,INT(14*RN
D+2),1)
540 CALL COLOR(B+1,INT(14*RN
D+2),1)
550 RETURN
```

```
1 !from 9 T 9 UG newsl. Aug
85
100 PRINT """Hello"" said TI
"
110 PRINT "Press ""ENTER"" t
o continue"
```

If you bite the hand that feeds you, you'll go hungry tomorrow. Don't be a pirate!

MEMORY FULL TO BUSTIN'

Jim Peterson
(c)

ø = Ø

ALTERNATIVE DIY EXPANSION

tel (0865) 863565

5a Lower Whitley Rd.
Farmoor,
Oxford,
OX2 9NU.

D.I.Y. EXPANSION SYSTEM. A different approach

The purpose of this article is to describe a different aporoach to expanding the TI99/4A without the expense of the PEB but instead building the hardware yourself. This article is written to promote expansion and dispel the often heard theory that expansion for the TI99 is too expensive or too difficult to build.

Initialy my system consisted of the basic console, Extended Basic, the Matchbox 32K RAM inside the console and my home built printer interface as a stand alone unit. My desire was to add a pair of disk drives to the system without ending up with a tangle of wires or lot's of units all plugged into each other.

Firstly, I had decided that to build a disk controller card was beyond me so my expansion system would be based around a commercial card. I started looking for a controller, either stand alone or a card for the PEB as either would do. Eventualy, I was offered a TI stand alone controller for £55. I already had a pair of Shugart SA200 single side 40 track 2/3 height drives.

I digress here slightly to mention one important point, namely what drives will work with the TI99. It seems to me that many people will only buy the same make and type of drives that TI used to use. However, disk drives are very standard things and as long as you get drives to suit the format you wish to use any make should work. The TI stand alone controller will only support (at the moment) single sided drives up to 40 track . The disk controller for the PEB will cope with double sided drives up to 40 track. In short any 40 track drive should, therefore, work with the TI controllers bearing in mind a stand alone controller will need modifying to cope with double sided drives.

From this point on, as I have a stand alone controller I designed my expansion system around this. All references will, therefore, apply to the stand alone controller but a similar system could be put together using a disk controller intended for the PEB. A separate circuit diagram is included depending upon which controller you have. Note that I have not tested the circuit shown for the PEB card so I can't guarantee that it will work.

WARNING. The following text describes how I built my expansion which works with no problems (I am writing this using Funnelwriter on my newly expanded system) but in doing so I have used the stand alone controller in a way it was never intended. I can't guarantee that the results I got will be repeatable and I accept NO responsibility for ANY damage which might occur as a result of anybody implementing a similar expansion system.

Firstly, I decided on the enclosure. I have not given concise details of the case as this is probably a matter of personal preference how you house the various components. My case is 380mm wide (the same width as the console), 225mm deep (just deep enough to house the drives) and 130mm tall (The minimum height of my drives). As you can see I have used the smallest box that everything will fit into. You may wish to make yours larger in case you want to further develop things.

The first stage was to mount the drives into the case. I chose to mount them to the far left of the case one above the other mounted conventionaly i.e. horizontally not like the drives in the PEB. However if you prefer or are used to it then mount them on their sides as in the PEB.

Next problem, a power supply? You can, if you wish, build your own. However, I came by a power supply from an old Winchester drive that gave +12 and +5 volts with more than enough power to supply the two drives plus cards so I used this. Because I did not build my own I cannot give details but the power supply circuit in the TI PEB is a good starting piont if you are going to build your own.

Next came the difficult bit, the disk controller. The first job is to remove the guts of the stand alone controller. Removing the case is simply a question of removing the 6 self tap screws in the bottom then the top lifts off. The part that concerns us is the rectangular tin box that occupies the rear half of the case. This is removed by undoing the 3 self tap screws that are visible. Next lift off the top cover of the tin box and inside you will find 3 boards sandwiched together. There are 4 more self tap screws passing through the lower two boards which hold the whole assembly into the case. Remove these and withdraw the tin box from the main case. You now have a collection of boards with the bottom one no longer fixed to the rest apart from by it's wires. To tidy things up we must bolt this board back to the other two using some small nuts and bolts. Once this is done put the 3 cards back into the tin box and put the lid on. I then mounted this unit onto the inside of the right hand end plate of my box, mounting it with 3 bolts and spacers to the 3 mounting flanges. It should be mounted so that the 'input' end that normally plugs into the console is facing towards the front of your expansion box. Then connect the power supply and disk drive ribbon cable. A short note here - the disc controller requires -5 volts as well as +12 and +5. Since my power supply does not have -5V and the current drawn on this supply is negligible I devised the circuit shown to generate -5V from the +5V rail.

That's taken care of the physical aspect, now the problem of wiring it up. It is not necessary to connect all 44 wires on the I/O bus to the expansion box, I traced out which pins were used by the disk controller and have tabulated this along with pins used by the 32K and printer interface later. I also determined that there is a direct connection on most pins on the controller from input to output i.e. the socket for the next peripheral.

My first approach was to use a relatively long cable and connect into the 'output' connector (since they are connected together this should work). However I had no end of problems with disk errors when I tried to load programs. I traced the main problem to a very slow rising edge to the pulse on the READY line (pin 12) Fitting a pull up resistor improved things but it was not a complete cure. The conclusion I drew was that timing is critical with the disk controller and the slow rising edge upset this. There seemed two options to cure this problem. Either fit buffers to the cable (as in the TI PEB) or to drastically shorten the cable. I opted for the second alternative and instead of connecting into the 'output' as mentioned I connected to the 'input'. As this connector is positioned at the front right of my expansion box then a cable emerging through the right hand end of the box to connect to the TI99 I/O bus can be very short indeed. In my case no more than 150 mm. This completely cured the problems of unreliable loading.

Now we have the disks working the next step was to remove the 32K matchbox RAM from my console and modify it to fit into my expansion box. The original circuit used the decoding logic within the console and hence required the 5 conections to the main board in the console. You could, if you wish, connect these from the console via a separate cable but a far neater arrangement is to modify the circuit to have it's own address decoding. This is achieved with a 74LS138 IC readily available for a few tens of pence. Also an invertor is needed to invert the sense of the DBIN signal. Here I used a 74LS04 again readily available. I have included a circuit diagram of the modified memory expansion. The main pin numbers refer to the I/O bus connections and are of relevance to this article or to anybody making a completely stand alone memory expansion and the pin numbers in brackets refer to the connections that would be used if you were building a version of this for the TI PEB. I connected the memory expansion to the 'output' of the disk controller connecting across only the used pins.

Now all that is left is to add a printer interface. Here I used my own device for connection of a parallel printer. This was originaly developed as a stand alone unit and suprisingly few people have shown interest in it. I suspect this may be due to the fact that before I built my expansion box I did not know if my interface was compatible with the disk controller. I now know it is and have included it. The circuit diagram is included for those wishing to build it themselves. You will need the DSR ROM which is available from me in the form of a pre-programmed EPROM for £10. The rest of the components are readily available. I can also supply ready made printed circuit boards. I have two layouts, one for a stand alone unit and the other layout more suited for inclusion in this expansion system. The printer interface is mounted alongside the 32K RAM board and again wired to the 'output' of the disk controller. Both the printer interface and the memory expansion require only +5V supply which are directly connected to the power supply.

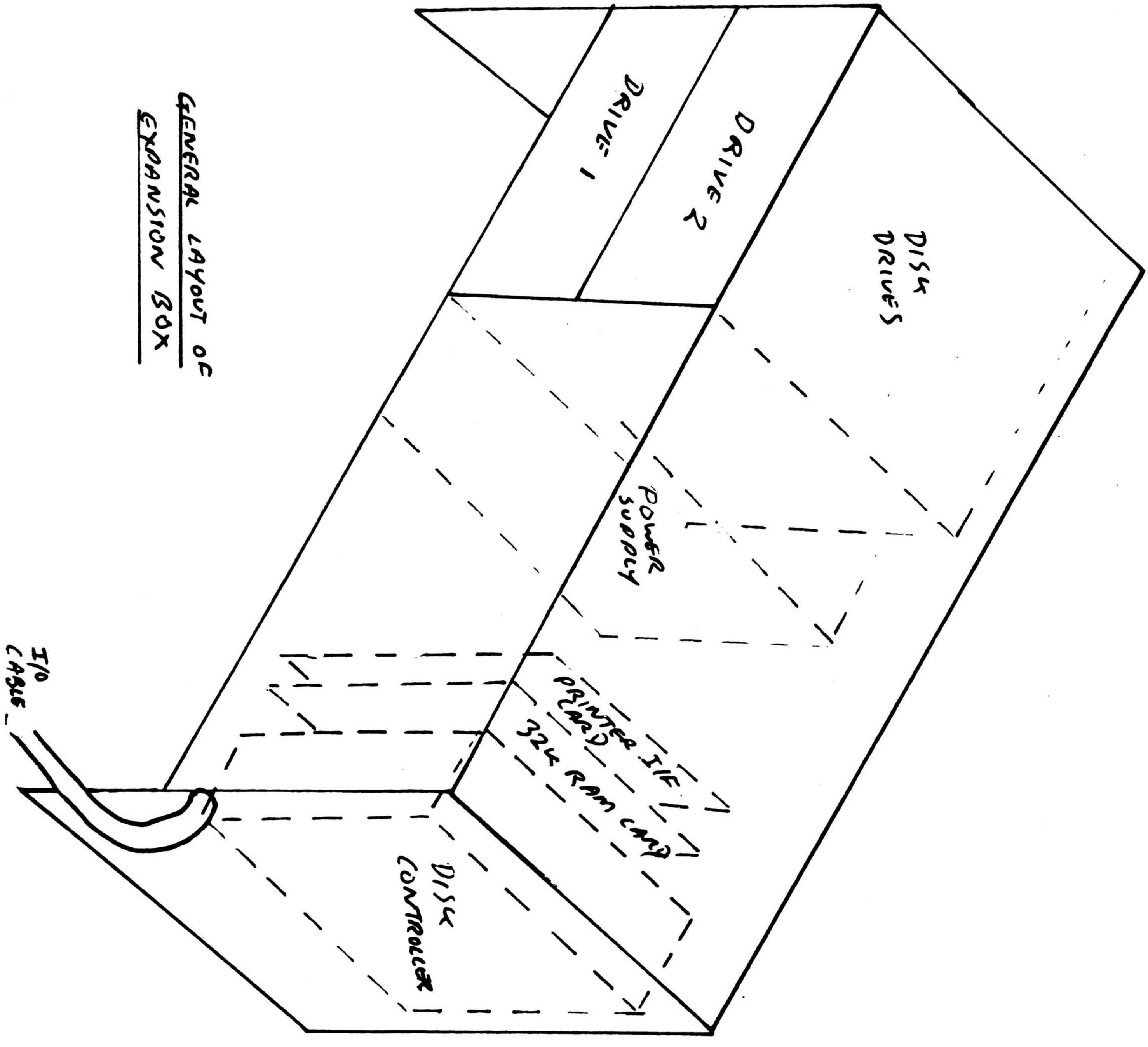
TI99/4A EXPANSION BOX WIRING TABLE

I/O BUS PIN No	TI PEB PIN No	USE	COLOUR	DISK	32K RAM	PRINTER
3	6	RESET	BROWN	YES	NO	YES
4	17	EXTINT	RED	NO	NO	YES
5	40	A 5	ORANGE	YES	YES	YES
6	33	A 10	YELLOW	YES	YES	YES
7	39	A 4	GREEN	YES	YES	YES
8	34	A 11	BLUE	YES	YES	YES
9	52	DBIN	PURPLE	YES	YES	NO
10	42	A 3	GREY	YES	YES	YES
11	31	A 12	WHITE	YES	YES	YES
12	4	READY	BLACK	YES	NO	NO
14	35	A 8	BROWN	YES	YES	YES
15	32	A 13	RED	YES	YES	YES
16	29	A 14	ORANGE	YES	YES	YES
17	38	A 7	YELLOW	YES	YES	YES
18	36	A 9	GREEN	YES	YES	YES
19	30	A 15	BLUE	YES	YES	YES
20	41	A 2	PURPLE	NO	YES	NO
22	51	CRUCLK	GREY	YES	NO	YES
24	50	PH3CLK	WHITE	YES	NO	YES
26	54	W.E	BLACK	YES	YES	NO
28	NOT USED	MBE	BROWN	YES	NO	YES
29	37	A 6	RED	YES	YES	YES
30	44	A 1	ORANGE	NO	YES	NO
31	43	A 0	YELLOW	NO	YES	NO
32	56	MEMEN	GREEN	NO	YES	NO
33	55	CRUIN	BLUE	NO	NO	YES
34	19	D 7	PURPLE	YES	YES	YES
35	24	D 4	GREY	YES	YES	YES
36	22	D 6	WHITE	YES	YES	YES
37	28	D 0	BLACK	YES	YES	YES
38	21	D 5	BROWN	YES	YES	YES
39	26	D 2	RED	YES	YES	YES
40	25	D 1	ORANGE	YES	YES	YES
42	23	D 3	YELLOW	YES	YES	YES
21,22, 23,25	3,5,7,20, 27,47,49, 53	GND	BLACK	YES	YES	YES

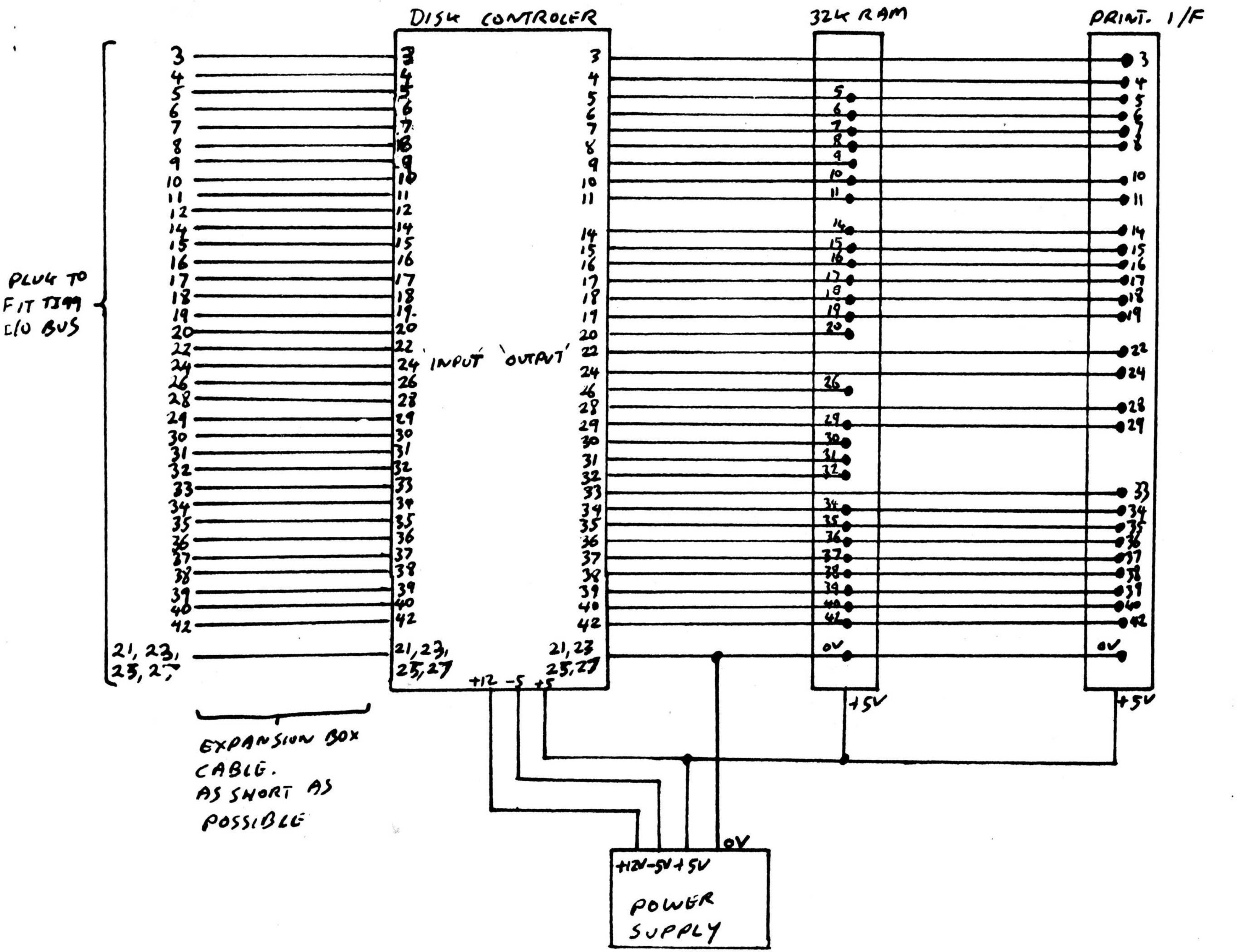
NOTE. I/O BUS PIN's 1,2,13,41,43,44 NOT USED.

USE SCREENED MULTICORE CABLE. AS SHORT AS POSSIBLE.

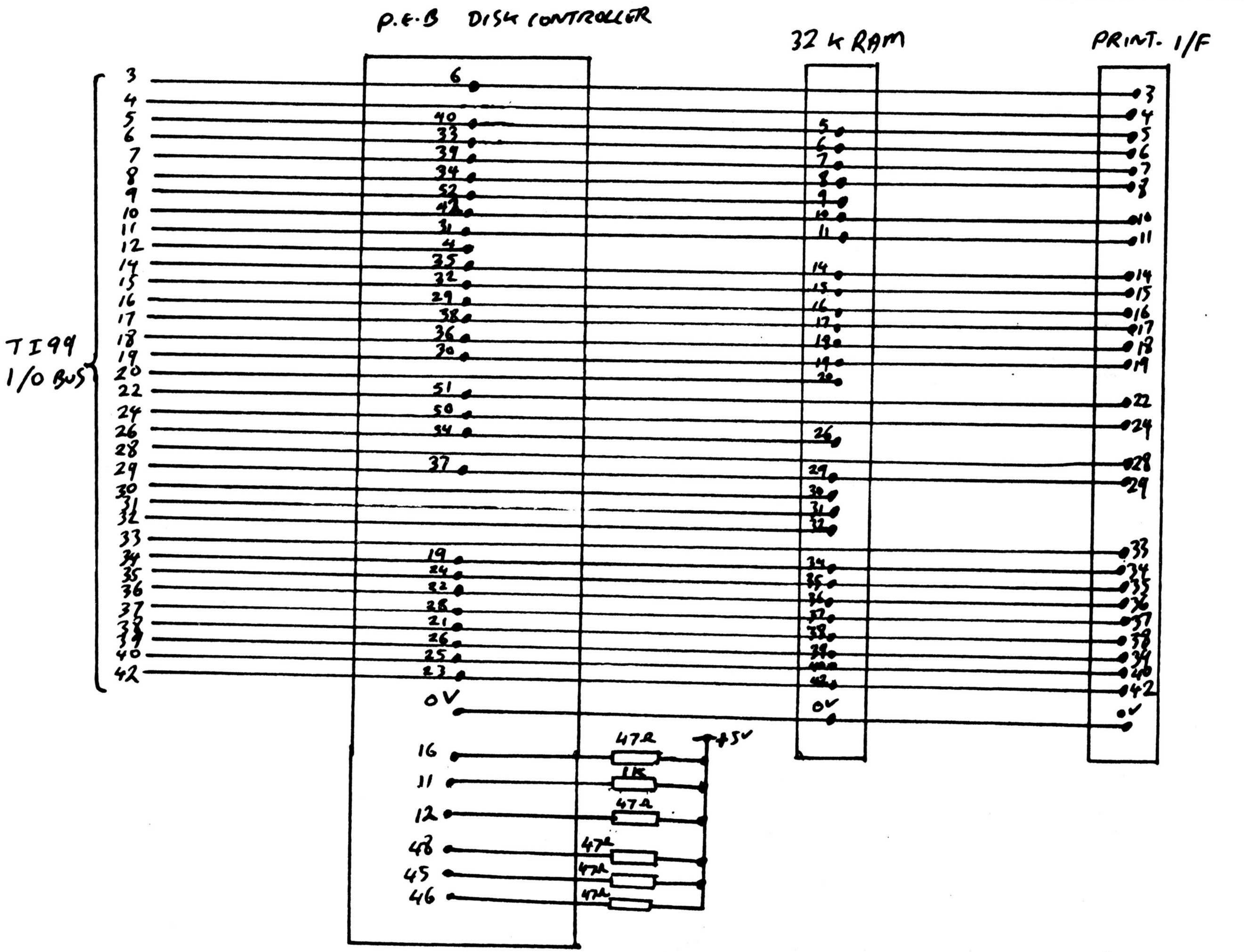
GENERAL LAYOUT OF EXPANSION BOX



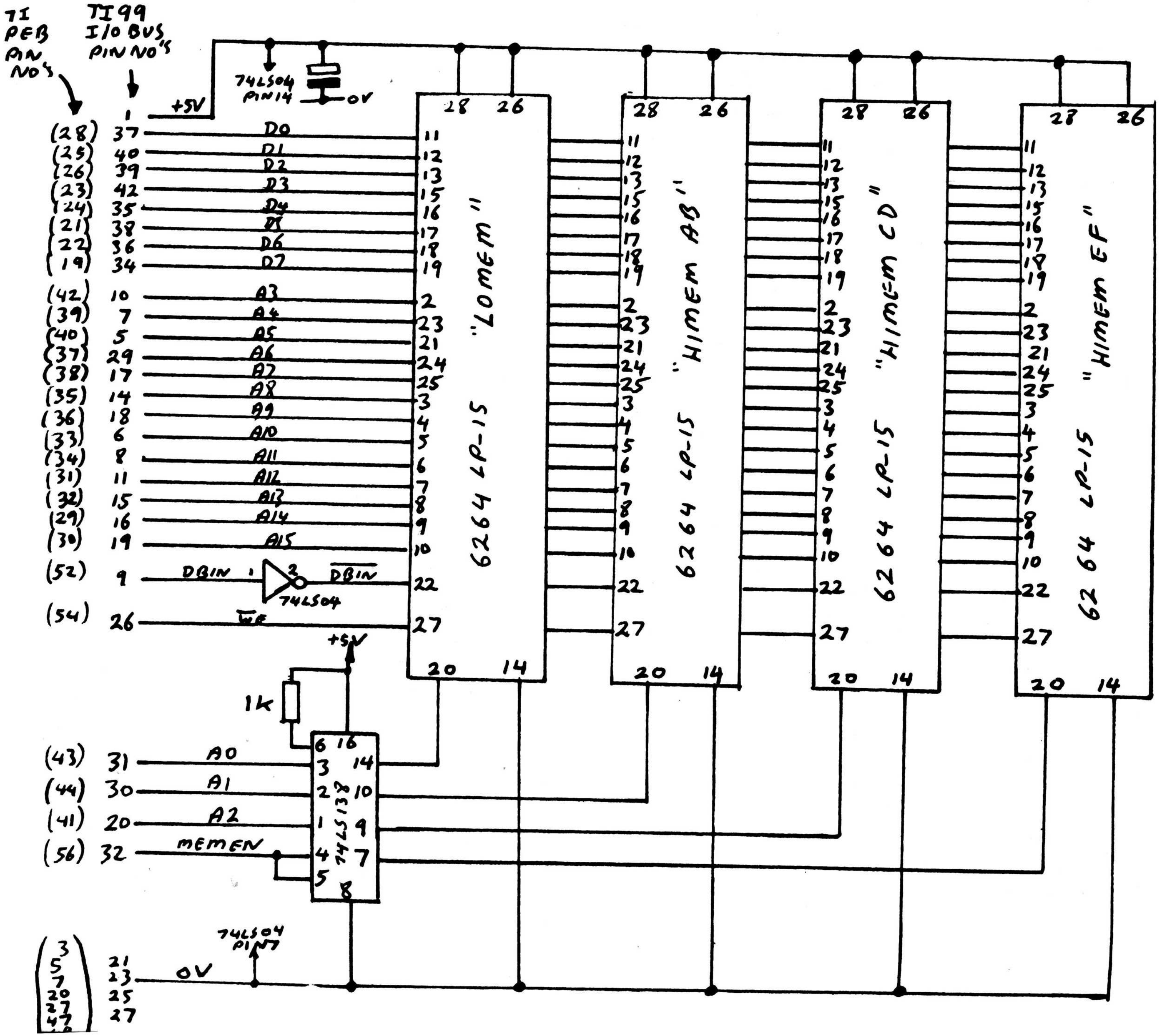
EXPANSION BOX CIRCUIT USING STAND ALONE CONTROLER



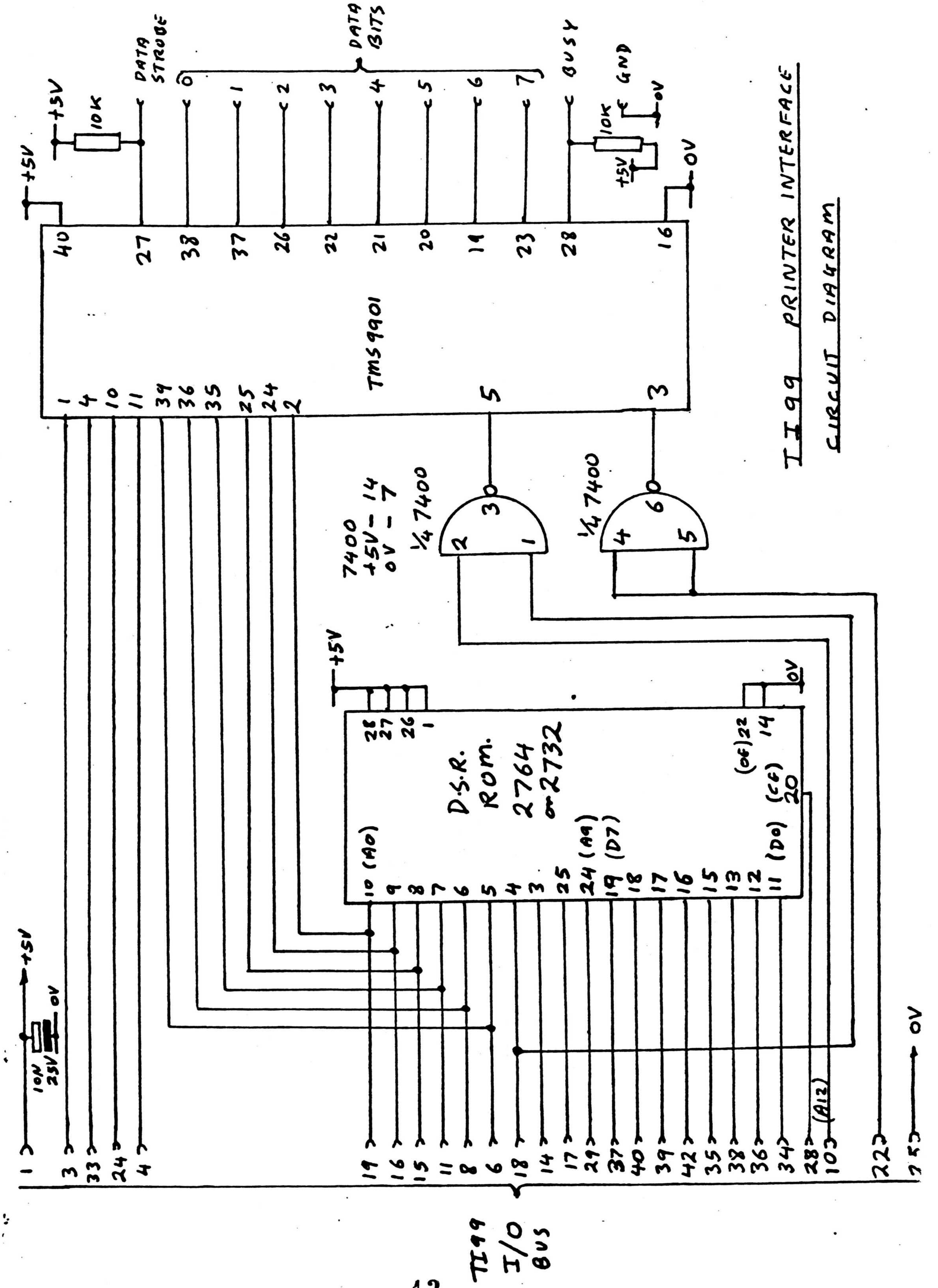
EXPANSION BOX USING P.E.B. DISK CONTROLLER



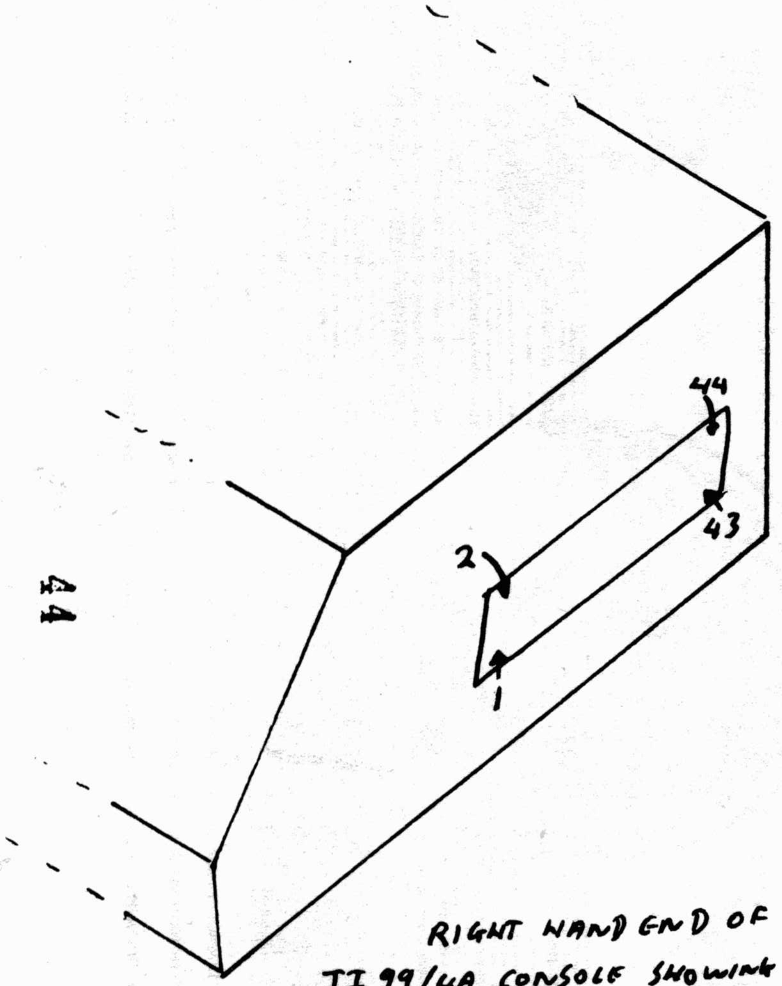
MODIFIED MATCHBOX 32K RAM EXPANSION



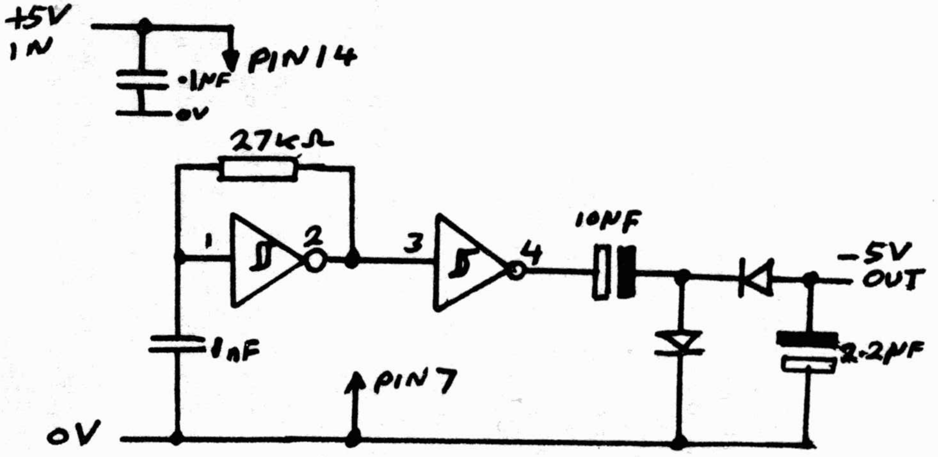
(2022 The final number here was not printed in the magazine)



TI 99 PRINTER INTERFACE CIRCUIT DIAGRAM



RIGHT HAND END OF TI 99/4A CONSOLE SHOWING I/O BUS PIN NUMBERING.



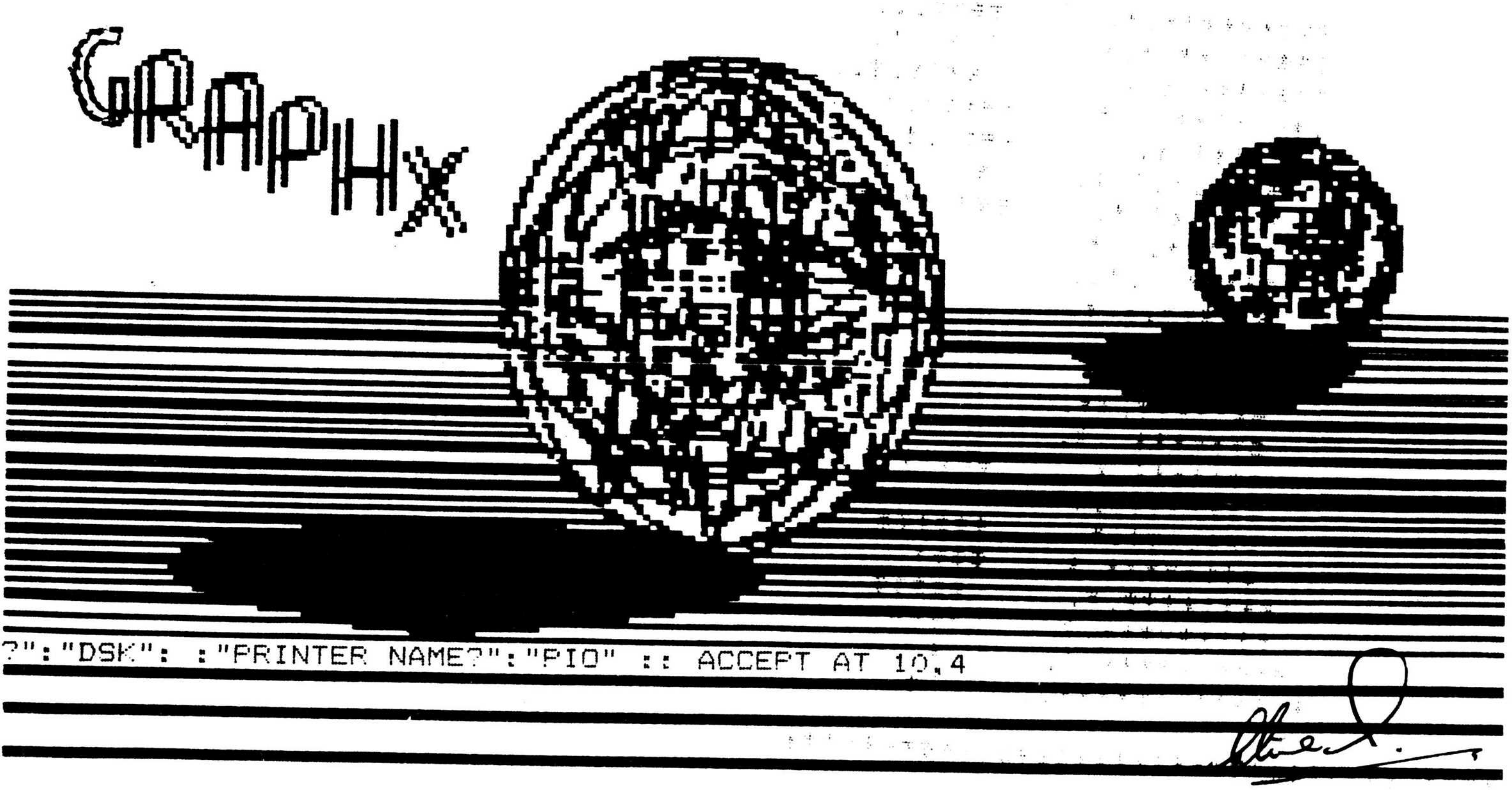
+5V to -5V CONVERTER CIRCUIT.

That about concludes the work I have done. I have ended up with an expanded system that would suit most requirements and is all housed in one relatively small box that sits neatly behind the console. One more piont to mention is cooling. I found that it was not necessary to fit a fan as merely punching some large holes in the bottom and a row of holes along the top of the back panel gave enough convection cooling and things only get slightly warm to the touch. If you find that you have heat problems try larger ventilation holes before resorting to fitting a fan.

I hope this article will stir some people into expanding their systems. The benefits of adding disk drives are enormous, not only is it much faster but you also open up new possibilities with the large amount of excellent public domain software in the user group libraries including Funnelwriter and other languages such as Forth and the small C compiler. The cost need not be too high, if you can find a disk controller for the sort of price I paid and only fit a single drive you will be able to expand your system for well under £200. You can always add a second (or third) drive at a later date.

If anybody has any queries on this article then you may telephone between 7.00 pm and 10.00 pm or address your queries in a letter but please enclose a ssae for a reply.

Happy expanding...... Dave Hewitt.



DIY EXPANSION UPDATE

MIKE GODDARD

After seeing my missive on DIY expansion in print, I now realise that some things are not immediately apparent. The first is that my system will work with whatever you have in the console (32K memory etc.) and speech synthesizer although the synthesizer should be the first thing on the connection as usual.

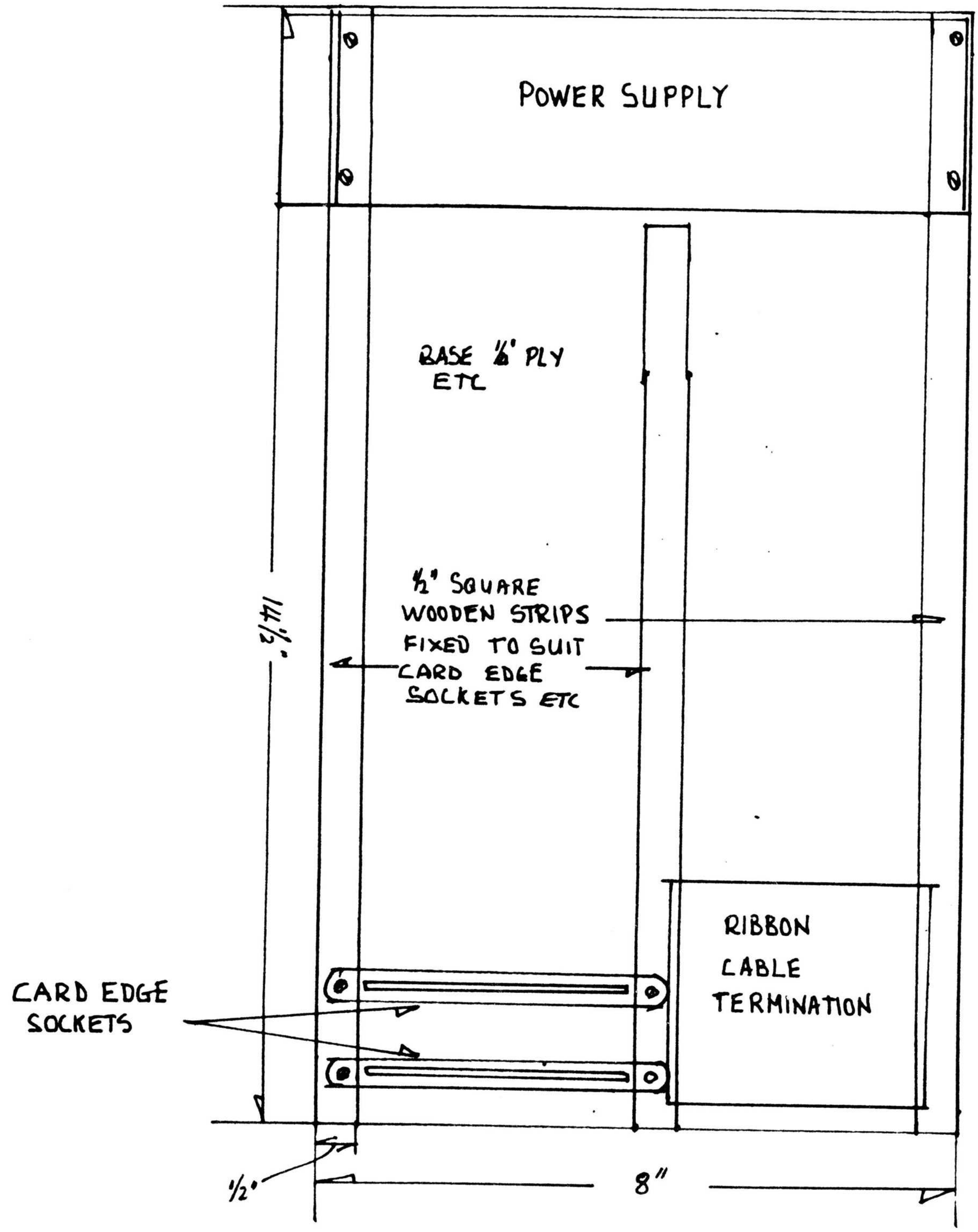
It has also been brought to my attention that not everybody is a "tin basher" although tin bashing for my box is minimal any box of suitable dimensions will do and several ready made boxes are available quite cheaply from various suppliers. Of course a case is not absolutely necessary and a mother board type of rig will work just as well but will not afford protection to either yourself from shock hazard or static "ZAPS" to the circuit boards. The other biggest problem to date is that "TANDYS" have finally informed me that the power supply I used in the original is no longer available, however all is not lost, Henry's Audio Electronics stock a suitable replacement type OF 406 which also contains a mains transformer so at £9.95 + 85p post looks to be a better buy. This unit is slightly larger but will fit into my original case with a bit of juggling, this unit doesn't have a -5 volt connection but one of the unused voltages can be converted when this proves necessary.

Several suggestions for further boards have been put forward such as RS 232 and disc interface and I hope to do something about them in the near future, although at the moment pressure of work and domestic commitments aren't allowing as much time as I have had in the past.

Mike

Errata:-

Data sheet 5 P10 connections are shown looking from the front of the board, not rear as stated



ALL SOCKET CONNECTIONS HARD WIRED BACK TO RIBBON CABLE TERMINATION AS ORIGINAL BOX WIRING : KEEP MAINS AWAY FROM COMPUTER WIRING!

ALL MEASUREMENTS APPROXIMATE AND CAN BE ALTERED TO SUIT MATERIALS TO HAND

TI-99/4A D.I.Y EXPANSION SYSTEM			DATA SHEET 8
DESIGNED M.R.G	DRAWN M.R.G 8:86	"MOTHER BOARD"	NOT TO SCALE

DIS VAR 80 CONVERTER PROGRAM

by Al Kinney

DISPLAY/VARIABLE 80 to PROGRAM CONVERTER

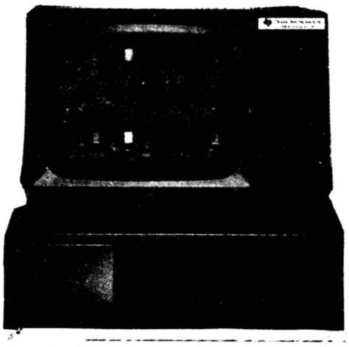
Article written by Al Kinney

Here is a nifty program to convert text that has been stored as DIS/VAR 80 back into a runnable program.(The origins of the program are uncertain to me, but, here it is!)

Now why, you might ask, would you want to do that? Glad you asked!! And now, I'm gonna' tell ya!! If you use COMPUSERVE, there is a Special Interest Group (SIG) for TI home computers, with over 1000 members!! These wonderful folks are constantly trying to out-do each other by putting excellent Public Domain software into the Down Load area of the SIG. Think of it a Mail Box, and the files are NOT bills.

Now, the only problem you have, is that for lots of very technical reasons, the files are stored as text in the DIS/VAR 80 format. In the "OLDEN" days, whenever you downloaded a file, you had to sit and type the darn thing in, and the way I type, I created more errors than I could fix!! Now, all that drudgery is gone, and by simply running the program listed below, you can recreate the program in the MERGE (DIS/VAR 163) format! Then, by removing the exclamation (REM) marks from each line, you will have a MERGEable file.

This also give you another way to edit existing files you may already have. Let's say you have a program, and you want to make a LOT of changes to it! First, you would simply load the program into memory, as usual, by entering "OLD DSK1.NAME". When loaded, you would then enter "LIST "DSK1.NAME/1"". That would cause the program to be written to the disk in DIS/VAR 80 format. The reason for changing the name should be obvious, to prevent over-writing the original file! Now, instead of flailing through the entire program, line by line, looking for variable XYZ, you can use TI-WRITER or EDITOR/ASSEMBLER, and do "Global" search and replace's! Those features are described in the respective manuals. After you have modified the program as needed, you simply run the conversion program, which rewrites the file with a NEW name. When it is finished, type "NEW", and "MERGE "DSK1.PROGRAM/2"", and proceed as before!



```
100 !**********************
110 ! TRANSLATES FROM
120 ! DIS/VAR 80 TO MERGE
130 ! FORMAT
140 !**********************
150 !
160 !USE A FULL SCREEN
170 !EDITOR TO CREATE
180 !EXTENDED BASIC PROGRAMS
190 !
200 !CREATE A FILE USING
210 !TI-WRITER - MAKE
220 !SURE YOU DISABLE THE
230 !WORD WRAP MODE AND
240 !LIMIT THE LENGTH
250 !TO 80 CHARACTERS
260 !
270 CALL CLEAR
280 DISPLAY AT(3,7)BEEP ERAS
E ALL:"***TRANSLATE***"
290 DISPLAY AT(7,5):"DIS/VAR
 80 FILENAME:"
300 ACCEPT AT(9,5)SIZE(15):I
N$
310 DISPLAY AT(12,5)BEEP:"ME
RGED OUTPUT FILENAME:"
320 ACCEPT AT(14,5)SIZE(15):
OUT$
330 OPEN #1:IN$
340 OPEN #2:OUT$,VARIABLE 16
3
350 LINPUT #1:L$
360 S=POS(L$," ",1)
370 ON ERROR 490
380 N=VAL(SEG$(L$,1,S))
390 ON ERROR 440
400 A=INT(N/256)
410 A$=CHR$(N-A6):: PRINT L$
420 PRINT #2:CHR$(A);A$;CHR$
(131);SEG$(L$,S+1,80);CHR$(0
)
430 GOTO 350
440 PRINT #2:CHR$(255);CHR$(
255)
450 CLOSE #2
460 PRINT : :"ENTER ""NEW""
AND THEN ""MERGE""THE TRANSL
ATED FILENAME:":"          ";OU
T$: : :
470 PRINT "REMEMBER TO REMOV
E THE         LEADING ""!"" IN
EVERY LINE.": : : :
480 END
490 ON ERROR 440
500 RETURN 350
```

BACK TO BASICS

Using CALL KEY

by Glenn Davis

Although everybody uses a key-scan now and then, many people do not realize there are some pretty handy tricks that can be used to make the rest of their programming easier. CALL KEY, as implemented on the 99/4A, has six modes to operate in: 0-5. As described in the User's Reference Guide the modes are as follows:

0 - Scans in the same mode scanned immediately before it.
1 - Scans left side of keyboard.
2 - Scans right side of keyboard.
3 - TI 99/4 mode. This has important applications! TI-Writer uses this mode in the text formatter.
4 - Pascal mode. This is true 7-bit ASCII and any code 0-127 can be generated off the keyboard (and more than 30 others). The 99/4 did not have this mode.
5 - TI BASIC standard mode. This is the mode used by BASIC, and if CALL KEY(0,K,S) is performed initially, it behaves like a CALL KEY(5,K,S).

Modes 1 and 2 are used the same way you would use the joysticks via CALL JOYST. The key-codes for modes 1 and 2 generate codes that are in the User's Reference Guide on page III-4. Two people can then operate games on opposite sides of the keyboard without interfering with each other (unlike the Apple //e), for example. But, since people use modes 1 and 2 for games already, I won't bother with an in-depth discussion.

Mode 3 acted differently on the 99/4. It behaved like a CALL KEY(5,K,S) on the 99/4A. Several character codes do not exist in this mode, however: ASCII 0, 16-31, 96, and greater are not available in mode 3. ASCII 96 is "`". (the accent grave) The codes 16-31 are normally ASCII control characters.

Mode 3 has several applications, one of which is accepting a single character from the keyboard, regardless of the case. In this manner, a "Y" (versus a "y") will always be detected when the "Y" key is pressed, whether or not the ALPHA-LOCK is depressed. Many programmers assume (by coding the program in a certain way) that the ALPHA-LOCK is not depressed or prompt the user that it must not be depressed (or vice-versa). Mode 3 allows you to get around that. By specifying CALL KEY(3,K,S) the 99/4A will accept only upper-case, so when the keyboard is scanned, it won't matter if the ALPHA-LOCK is up or down. Amazingly, this also works on INPUT, LINPUT and ACCEPT statements too! A short program can illustrate this.

```
100 CALL KEY(3,K,S)
110 CALL CLEAR
120 INPUT "SEE! ONLY UPPER-CASE
    NO lower-case":A$
130 GOTO 120
```

Try the ALPHA-LOCK in both positions and you'll see that it is impossible to get lower-case on the screen by typing off the keyboard! In either case, the tilde ("~") and other special characters cannot be entered. Try inserting LINPUT and ACCEPT in place of INPUT if you have TI Extended BASIC.

Another application for mode 3 appears when dealing with files on the disk system and printer. Programs are written to prompt for "device.filename". "Device" must be in upper-case for the 99/4A to recognize it as a valid device name. "Filename" can be any combination of upper- and lower-case characters. If you have either a Disk Memory System or an RS232 interface, you should know this. The Disk Memory System manual recommends that filenames be only in upper-case because the Disk Manager (1) Cartridge won't display lower-case correctly. In a program where the user enters the filename, just add a CALL KEY(3,K,S) before the INPUT, LINPUT or ACCEPT (in TI Extended BASIC) statements. This allows entry of a correct filename without fumbling with the shift-key or depressing the ALPHA-LOCK. Beware, though, that if the program is retrieving a disk file that is named with lower-case, the disk controller will not find it if the name is entered this way.

Mode 4 has fewer applications on an elementary level. All ASCII codes in 7-bit format are enabled (0-127), unlike modes 3 and 5, where 16-31 are disabled. As when switching to mode 3, switching to mode 4 affects the INPUT and ACCEPT statements. Typing in mode 4 will allow data in files to be generated that couldn't be generated in modes 3 or 5 (i.e. with control characters in them that were entered via the keyboard. If the values are entered through the program with the CHR$ function, no difference will appear). One note of caution: ENTER and CTRL-M both return the same character code in mode 4 (which makes code 141, normally returned by CTRL-M, unavailable). If this is undesirable, you'll have to program around it. I don't know of any TI software that uses this mode, since it is incompatible with the 99/4. Other codes are also generated.

Mode 4 has some other side-effects that may have some application, although I haven't found many yet. One is that in both TI BASIC and TI Extended BASIC the CLEAR key (FCTN 4) will not stop a program on an INPUT or ACCEPT statement. Try this:

```
100 CALL KEY(4,K,S)
110 INPUT "TRY IT":A$
120 GOTO 110
```

Can't break the program? Hmmm ... Gee, how <u>do</u> you break such a program? Don't turn your 99/4A off. Just let the program sit. The answer comes indirectly from the User's Reference Guide. It tells us the code for CLEAR in mode 5 (normal BASIC mode) is 2. In mode 4 (Pascal) a 2 code must be produced by some key, so which one? Well, looking to the appendix in the User's Reference Guide, we see that a CTRL-B will give a 2 code. Now RUN the above program again. Have you noticed some other unique things ... ERASE (FCTN-3) doesn't work either ... and neither do DELETE and INSERT (FCTN-1 and -2). Look up the codes for those keys in the User's Reference Guide. Try them on the INPUT lines. Neat huh?

Try FCTN-S. So, you can't backspace either! Now if any of you are familiar with any other computer systems, mainframes or telecommunications, then you might be able to guess what key-stroke will result in a backspace like FCTN-S will: CTRL-H. Try this one too. Try some others and tell the rest of us if you find out something interesting!

If you intend on using mode 4 in conjuction with files on the Disk Memory System or the RS232, remember to push the ALPHA-LOCK down. Otherwise the lower-case might give you a headache.

Mode 5 is the normal BASIC mode, which most people should be familiar with. If not, it was discussed in depth in COMPUTE! of November 1983. There are some differences between TI BASIC and TI Extended BASIC as far as applications of mode 5. TI BASIC allows graphic definitions up to ASCII 159, while TI Extended BASIC only allows definitions up to ASCII 143 (16 fewer; the memory is used to keep track of sprites). When PRINTing graphics to the screen instead of the slower CALL HCHAR or CALL VCHAR, no codes past CTRL-O should be used for TI Extended BASIC. Codes up through CTRL-9 may be used in TI BASIC.

Usually, when CALL KEY is used for routine-jumping, two methods are employed. The first method uses a series of IF-THEN statements to check each condition individually. This procedure is, well, slow. IF-THEN allows the 99/4A to make as few as one scan per second. Depending on how it was coded, of course. The second method normally used is a CALL KEY ON exp GOTO where the command keys are consecutive, as in A-G or 1-9. This method is quite fast, but a programmer must use consecutive keys, which makes the mnemonic (memory-aiding) value of such keys poor.

But really, just two statements can handle all the branching from the key-scan to subroutines. How can just two lines do that? The secret is in the two statements, often used separately, but rarely together: POS and ON exp GOTO (or GOSUB). The POS function searches a string for the value returned by CALL KEY, and the ON exp GOTO performs a calculated jump, i.e. a jump determined by the value of exp. The string should be defined early in the program (once only to avoid wasting time) using string literals and/or the CHR$ function. The actual characters contained in the string are one greater (ASCII code) than the value required. For example, the "TE" in the string below represent "S" and "D".

```
100 STRING$=CHR$(0)&"TE"
110 CALL KEY(3,K,S)
120 ON POS(STRING$,CHR$(K+1),1)
    +1 GOTO [illegal or unused
    key routine], [no key
    routine], ["S"-routine],
    ["D"-routine]
```

The full code is presented here:

```
100 CALL KEY(3,K,S)
110 STRING$=CHR$(0)&"TE"
120 CALL KEY(0,K,S)
130 ON POS(STRING$,CHR$(K+1),1)
    +1 GOTO 140,160,180,200
140 PRINT "ILLEGAL KEY"
150 GOTO 120
160 PRINT "NO KEY"
170 GOTO 120
180 PRINT "S KEY"
190 GOTO 120
200 PRINT "D KEY"
210 GOTO 120
```

When you run this program, leave the ALPHA-LOCK off. You'll see why shortly. The character STRING$ will represent the key-codes retrieved by the KEY subprogram. One (1) must be added to this expression because the KEY statement returns negative one (-1) when no key is pressed, which cannot be used as an argument for the CHR$ function. The argument for CHR$ [K+1] then becomes zero when no key is pressed, and zero is in the first position of STRING$ (check the definition in line 110). This makes the routine loop back if no key is pressed to scan the keyboard again. If an illegal (unused) key is pressed the POS function will return a zero, meaning that a match was not found of the strings. Therefore, one (1) must be added to the expression because the ON exp GOTO cannot use zero as an argument. The line, therefore, becomes as is shown in line 120 above. To add more routines to a program, all that is necessary (beyond the routine's code) is to add another line number to the list in the ON exp GOTO and an appropriate character to the string. Programs written in this fashion can be easily maintained and modified.

The keyboard can be scanned up to ten scans per second this way, and it catches virtually every key when waiting for a key to be pressed (the routine above does not; BASIC is slow at scrolling).

Now, finally, we'll cover mode 0, an unusual mode. In the program above, mode 3 was specified once, followed by calling mode 0. This "locks" the computer into mode 3 so all successive mode 0 scans are in mode 3 (refer to the definition of the modes above) until a non-zero mode is called. This is useful in applications programs, for example, an Extended BASIC data-base program that only uses upper-case characters as command letters. A single CALL KEY(3,K,S) at the beginning of the program will "lock" the entire program into upper-case. Likewise, modes 1 and 2 are affected by mode 0 in the same way. Unfortunately, CALL JOYST doesn't recognize mode 0 (it returns a * BAD VALUE error).

The information presented here should give programmers of all levels a little more insight into the powerful TI-99/4A. Take some time to incorporate these ideas into programs of your own and see how much better they RUN! (c) MSP 99

IS THE COMPUTER A FEMALE?

by ROGER GREEN News on Sunday (c)
Caxton House, 13 Borough Road, SE1.

WHAT sex is a computer?

You may never have given the question much thought – imagining, perhaps, that a machine which makes thousands of decisions every minute hasn't got so much as a microsecond to spare for thoughts of the flesh. But you could be wrong.

For a start, why should a computer tell you every time it feels a sexy tingle in its peripherals? It's probably much too busy enjoying itself.

And – what is worse – it may not really fancy you. Can you honestly swear that you asked if it wanted to come and live with you?

Of course, anyone who has got beyond Section One of a computer manual will know how to make it say "I love you. Ready." Computers are always saying they are reading for something or other.

That's why Americans call them "user-friendly" and talk about "human/machine interfacing".

Go along to your local Dixonis and you'll see them: rows of men (and boys) all busily interfacing with the computers.

You won't see many women there – and there's a reason. For computers are undoubtedly female.

Not that men admire computers for their looks. The petite, blonde Apple Mackintosh is the only one that comes near being an electronic Joanna Lumley.

So what do men see in them? For a start they have one of the virtues – if you can call it that – that men used to find in women. They are built for a life of drudgery.

Of course they won't iron shirts or darn socks, but that's only because those bright young inventors in California's Silicon Valley spend all their time in T-shirts and flip-flops. Clearly, though, a computer's place is in the home.

Once men got the idea that computers are a substitute for the uncomplaining housewife of old, they had to find ways of giving them commands.

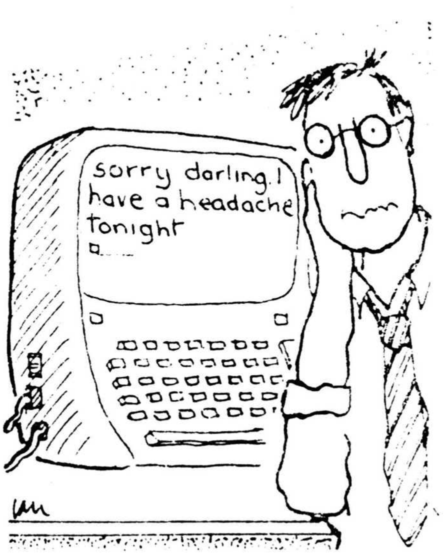
So they came up with the computer language, BASIC, which stands for Brutish Arrogant Sexist Instruction Code.

BASIC has turned the computer into a man's plaything. Male chauvinist programmers can now have a quick peek or poke (not to mention a byte or a nibble) any time they feel the urge.

But what are real live women to make of this? Should they view computers as rivals for men's affections and go round snipping off their cables?

Do they bring out the worst in men, or are they just a harmless way of working off sexist tendencies?

That, as Sir Clive S would probably say, is the 64K question.



FROM L.A.99ERS GROUP PAGES

*** Tom Freeman ***

DOUBLE COLUMN and SIDEWAYS, A REPRISE

Since last summer, when I wrote articles on extended basic programs to print double column text, and to print text sideways (for spreadsheets) using the graphics capabilities of the printer, I have had several requests for a reprint. There were in addition some errors which crept into the printing process, and many people never picked these up in later issues of TOPICS, so I got quite a few "HAALP!" calls. This month's column is a response to these requests - however so as not to completely bore you I have added several enhancements to each program (hence they will be a little longer to type in).

DOUBLE COLUMN

This program makes use of the TI-WRITER formatter to give your text the proper margins and right justify it. The latter is not necessary, but I think it gives the final double column format a neater look. The special formatter commands for underscoring, overstriking and required spacing (&, @, and ^) are preserved, AND ALL SPECIAL CODES YOU SEND TO YOUR PRINTER will also be formatted correctly.

You should begin your text with a line something like this: .LM0;RM56;FI;AD;IN+5;PL300<CR> [Note: <CR> represents the carriage return in TI-WRITER, from now on referred to as TIW, but I can't print the little character for you] The left margin should be 0, but the right margin can be whatever you wish. It will correspond to the column width (minus 1) entered in line 140 of the program. FI(ll) is also necessary, AD(just) is optional, as is IN(dent). The PL (page length) should be greater than the final number of lines in your file, since we do NOT want form feeds to creep in! In determining your right margin remember that TWO columns will be printed side by side, and that you need to compensate for the margins left and right you wish on the final printout, as well as space between.

When you have finished your text, save it using SF, and go to the Formatter. BE SURE that all true "&"s and "@"s have been doubled, as the formatter considers single ones to be formatting commands. "^" needs to be transliterated. Also be careful of an "*" followed by a number (the formatter appears to consider this a command for a mail list). Now for your output file DO NOT use the printer, but instead another disk file, with a different name, e.g. if the input file is DSK1.FILE then the ouput could be DSK1.FILE1. When this process is finished, return to the Editor and load this new file. The first thing you see will be three or more little <LF>'s, and possibly a <PA>. Delete them with FCTN 3. Now scan down the file to the end - you will then see a LOT of <LF>'s depending on how far you were from 300 lines. Delete from here to the end (easiest way is: FCTN 9, D <ENTER>, number of first line to be deleted, comma, E <ENTER>. Note that all the right margins are lined up if you used AD. If you used any printer commands (see end of article) they won't, and you will have to scan the file to see if any wound up at the beginning or end of a line. Once you have finished modifying the file, save it again, using the PF function this time. You may use the same or another name.

Now run the following program in XBasic:

```
100 CALL CLEAR :: DIM A$(200),C(200):: E$=CHR$(27)
:: CR$=CHR$(13):: LF$=CHR$(10):: FF$=CHR$(12):: T$
=CHR$(9):: LT$=CR$&T$ :: PG=1
110 DISPLAY AT(6,1):"DOUBLEPRINT": : :"INPUT FILE?
":"DSK": :"PRINTER NAME?":"PIO" :: ACCEPT AT(10,4)
SIZE(12)BEEP:F$ :: OPEN #1:"DSK"&F$,INPUT :: ACCEP
T AT(13,1)SIZE(-28)BEEP:P$
120 OPEN #2:P$&".CR"
130 DISPLAY AT(1,1)ERASE ALL:"IN THE NEXT 3 INPUTS
,BE SURETHAT TWO TIMES WIDTH LEFT   MARGIN + SPACE
 BETWEEN DOES NOT EXCEED YOUR PRINTER'S   CAPACITY
"
140 DISPLAY AT(7,1):"HOW MANY SPACES LEFT MARGIN?6
": :"HOW MANY BETWEEN COLUMNS?   6": :"WIDTH OF CO
LUMN? 57"
150 ACCEPT AT(8,1)SIZE(-2)BEEP:LEFT :: ACCEPT AT(1
1,1)SIZE(-2)BEEP:BETW :: ACCEPT AT(13,18)SIZE(-2)B
EEP:WIDTH
160 LEFT=LEFT+1 :: RIGHT=LEFT+BETW+WIDTH
170 PRINT #2:CHR$(15);E$;"D";CHR$(LEFT);CHR$(RIGHT
);CHR$(0);!SET CONDENSEDPRINT, TABS
180 DISPLAY AT(15,1):"DO YOU WISH TO RESET LINE
SPACING, COLUMN LENGTH, AND PAGE LENGTH AT EACH PA
GE?   (Y/N) N"
190 ACCEPT AT(18,7)SIZE(-1)VALIDATE("YN")BEEP:AN$
:: IF AN$="Y" THEN CLFLG=1
200 GOSUB 390
210 PRINT #2:E$;"A";CHR$(LS);E$;"C";CHR$(PL)
220 IF EOF(1)THEN CLOSE #1 :: CLOSE #2 :: STOP ELS
E X,Y,X1=0
230 X=X+1 :: LINPUT #1:A$(X):: B=POS(A$(X),LF$,1):
: IF B THEN A$(X)=SEG$(A$(X),1,B-1):: Y=Y+1 :: C(X
)=0 ELSE C(X)=1
240 PRINT X;Y
250 IF X1 THEN 270
260 IF Y=CL THEN X1=X
270 IF Y<2*CL AND EOF(1)=0 THEN 230
280 IF Y<2*CL THEN CLOSE #1 :: GOTO 310
290 GOSUB 350 :: IF CLFLG THEN 300 ELSE 220
300 CALL CLEAR :: PG=PG+1 :: DISPLAY AT(20,11):"PA
GE";PG :: GOTO 200
310 A$(X+1)="" :: EX=0 :: FOR Z=1 TO X :: EX=EX+C(
Z):: IF Z-EX=INT((Y+1)/2)THEN X1=Z :: GOTO 330
320 NEXT Z
330 GOSUB 350
340 CLOSE #2 :: STOP
350 X=0 :: Y=X1
```

```
360 X=X+1 :: PRINT #2:T$;A$(X):: IF C(X)THEN PRINT
 #2:CR$ :: GOTO 360
370 Y=Y+1 :: PRINT #2:T$;A$(Y):: IF C(Y)THEN PRINT
 #2:LT$ :: GOTO 370
380 PRINT #2:LF$ :: IF X<X1 THEN 360 ELSE PRINT #2
:FF$ :: RETURN
390 DISPLAY AT(22,1):"LINES PER COLUMN? 55":"LINE
SPACING 12/72 IN.":"PAGE LENGTH (LINES)? 66"
400 ACCEPT AT(22,19)SIZE(-2)BEEP:CL :: ACCEPT AT(2
3,14)SIZE(-2)BEEP:LS :: ACCEPT AT(24,22)SIZE(-2)BE
EP:PL :: RETURN
```

A quick explanation: Lines 110 to 160 set up the parameters for formatting your printout. You should of course modify the defaults to whatever you most often use, since you then need press only enter at each input. Line 170 then sends the codes to the printer for condensed print and the tabs for each column. Consult your printer manual to confirm that 15, and Escape "D" are the proper codes. The subroutine at line 390 will allow you to input line spacing other than the standard 1/6 inch and set the number of lines per page as well as the printed page length. Lines 180-190 first allow you to signal that you may wish to change these at each page. This is convenient if your text is long, but you wish to put different amounts on each page, or squeeze more in, etc. Line 210 sends the codes for line spacing and page length to your printer (Escape "A" and Escape "C" - check your manual). Line 220 resets parameters for each page, if more is left to do. Now the meat of the program. Line 230 picks up the lines one by one and puts them into the array A$(). An <LF> is checked for (if one is not there the formatter has something special to do, such as underscore or overstrike). Variable Y keeps track of the number of PRINTED lines, X the number of INPUT lines, and the array C() signals which input lines are not to be line fed. Line 240 puts X and Y on the screen, for your interest and can be deleted. X1 will represent the number of input lines to be printed up to the midpoint (i.e. first column) and is determined by line 260. Line 250 skips over this if X1 has already been found. Line 270 returns us for more input, if the end of the file hasn't been reached, and there is still more to do on the page (2*CL lines). If the end of the file has been reached before there were 2*CL lines, then a special routine is needed to caluculate the midpoint (line 310). Otherwise the subroutine at line 350 does ALL the printing. Here X and Y keep track of the left and right columns, and the array C() signals whether the printer should go to the next tab, or return to the same one for an extra line. After all the lines have been printed, there is a form feed and we go back for more (after resetting line spacing and page length if necessary). The last segment in lines 310-340 is for the last, incomplete page and ends in a stop. You could also have it return to line 110.

If your printer doesn't have tab settings, you are almost out of luck, but not quite. It is possible that any special codes you send to your printer won't work - I'm not sure. In any case we take advantage of the computer's tab function. I believe the open statement has to include a VARIABLE 136, or some such. Then make the following substitutions:

```
   360 X=X+1 :: PRINT #2:TAB(LEFT);A$(X);CR$:: IF C(X)
   THEN 360
   370 Y=Y+1 :: PRINT #2:TAB(LEFT+RIGHT);A$(Y);CR$:: IF
   C(Y) THEN 370
```

and delete everything in line 170 after CHR$(15).

If you are embedding control codes to your printer in the text, and also wish to right-justify, there is a problem, i.e. the formatter counts them as characters, but the print-head doesn't move, so the right margins won't line up! The following method will compensate: use a single unused character and transliterate it to the sequence of control codes you wish, and add a 32 which is a space and will move the print-head the one space that the formatter thought was there. E.g. I used the left brace for underlining .TL 123:32,27,45,1 The 32 should precede the control codes for an opening command and follow them for a closing one. The only problem is when these codes come at the beginning or end of a line, then the spaces aren't buried correctly! The only way to solve this is to scan the formatted file on disk and see whether any do in fact appear at the start or end of a line. In these cases, in fixed mode, delete a space at the beginning and insert it somewhere else, and if the extra space is at the end, insert an extra anywhere on the line (this moves the last character to the end).

SIDEWAYS

This program is deceptively simple. All it requires is a printer capable of dot graphics, and most seem to be these days. The usual code is Escape "K" and is the only one the program can use. The program has been revised to allow for variable width lines, and the data creating program has been extensively revised to allow you to have a CHARA1 file on disk and use that for the character definition. No need to type in a lot of CALL CHAR statements!

The actual sideways printing program merely sets up an array of 60 lines (each block letter is 8 dots wide and 480 dot columns are allowed on a page) and then picks one letter at a time off each one. All it needs is a definition of what the letter looks like on its side, and that is the purpose of the data creating program. It uses the hex codes that TI has already built into the computer for each character and sets up a data statement in a MERGE format. This is then merged, once only, into the final program and that's all there is. I can't just give you the data statements to type in, because they are mostly non printable ASCII codes.

For the data creating program, I originally used the characters built into the console, and suggested typing in a LOT of CALL CHAR statements if you wanted others. The version that follows uses data statements to poke an assembly language program directly into memory. This subprogram looks for a file called CHARA1 on DSK1 and if it is there loads it into the pattern descriptor table in VDP ram (it also overwrites the color table and you will see crazy things on the screen until the program is finished!) The AL program requires a DSRLNK for disk access. In XBasic this would have required a LOT of extra typing since the routine doesn't exist in the module. Editor/Assembler does have it. Hence the following program is in Basic and MUST be run with the E/A module in.

```
100 DATA 5,0,2,250,0,0,8,0,0,11,68,83,75,49,46,67,
72,65
110 DATA 82,65,49,0,2,0,16,0,2,1,39,16
120 DATA 2,2,0,21,4,32,33,16,2,0,16,9,200,0,131,86
,4,32
130 DATA 33,32,0,8,4,224,131,124,4,91,255
140 DATA 65,32,32,32,32,32,39,38,255
150 CALL INIT
160 X=10000
170 READ A
180 IF A=255 THEN 220
190 CALL LOAD(X,A)
200 X=X+1
210 GOTO 170
220 IF X>10100 THEN 250
230 X=16176
240 GOTO 170
250 CALL LOAD(8234,63,48)
260 CALL LINK("A")
270 OPEN #1:"DSK1.DATAMERGE",VARIABLE 163
280 FOR X=1 TO 19
290 PRINT #1:CHR$(0);CHR$(X*5);CHR$(147);
300 FOR Y=1 TO 5
310 CALL CHARPAT(X*5+Y+26,C$)
320 GOSUB 470
330 D$=""
340 FOR Z=8 TO 1 STEP -1
350 D$=D$&CHR$(D(Z))
360 NEXT Z
370 PRINT #1:CHR$(199);CHR$(8);D$;
380 IF Y=5 THEN 390 ELSE 410
390 PRINT #1:CHR$(0)
400 GOTO 420
410 PRINT #1:CHR$(179);
420 NEXT Y
430 NEXT X
440 PRINT #1:CHR$(255)&CHR$(255)
450 CLOSE #1
460 STOP
470 FOR Z=1 TO 8
480 E1$=SEG$(C$,2*Z-1,1)
490 E2$=SEG$(C$,2*Z,1)
500 F1=ASC(E1$)-48+7*(ASC(E1$)>60)
510 F2=ASC(E2$)-48+7*(ASC(E2$)>60)
520 D(Z)=F1*16+F2
530 NEXT Z
540 RETURN
```

You now have a DISPLAY, VARIABLE 163 file on the disk in DSK1 named DATAMERGE. Now type in the following program which will become the main program. Note that in line 150 you can set the maximum width of a line if it is less than 80 characters, and that line 160 resets the length of page to one more line than this, so that contiguous spreadsheets can be printed together. Check your printer manual for lines 130 and 160. When you have finished typing the program as printed in this article, then type in MERGE DSK1.DATAMERGE. This will put in 19 lines from 5 to 95. <u>Do not list</u> the program in this form to a printer, because lines 5-95 can't be handled by a printer.

SIDEWAYS is now ready to use. The only limitation is that the files on disk to be used must contain <u>only</u> ASCII characters 32-127, and must have been saved using the PF function of the TIW, not SF.

```
100 CALL CLEAR :: PRINT "SIDEWAYS PRINT ":" by Tom
 Freeman": :
110 DIM D$(126),A$(60):: FOR X=32 TO 126 :: READ D
$(X):: NEXT X
120 P$="PIO.CR"
130 ESC$=CHR$(27):: OPEN #2:P$ :: PRINT #2:ESC$&"A
"&CHR$(7)!RESET LINE FEED TO 7 DOTS(7/72 IN)
140 FLAG=0 :: PRINT "TEXT FILE:":" DSK";:: INPUT "
":F$ :: ON ERROR 140 :: OPEN #1:"DSK"&F$,INPUT ::
ON ERROR STOP
150 INPUT "MAX LINE WIDTH <81 ":Z :: IF Z>80 THEN
150
160 PRINT #2:ESC$;"C";CHR$(Z+1)!RESET FORM FEED TO
 Z+1 LINES
170 FOR X=1 TO 60 :: LINPUT #1:A$(X):: PRINT A$(X)
:: IF EOF(1)THEN 190
180 NEXT X :: GOTO 210
190 FLAG=1 :: CLOSE #1 :: IF X=61 THEN 210
200 FOR X=X+1 TO 60 :: A$(X)=RPT$(" ",Z):: NEXT X
210 FOR X=1 TO 60 :: A$(X)=A$(X)&RPT$(" ",Z-LEN(A$
(X))):: NEXT X :: FOR X=1 TO Z :: PRINT #2:ESC$&"K
"&CHR$(224)&CHR$(1):: PRINT "PRINTER LINE";X
220 FOR Y=60 TO 1 STEP -1 :: B$=SEG$(A$(Y),X,1)
230 PRINT #2:D$(ASC(B$));
240 NEXT Y :: PRINT #2:CHR$(13)&CHR$(10):: NEXT X
:: PRINT #2:CHR$(12):: IF FLAG=0 THEN 170
250 INPUT "DO ANOTHER?(Y/N)":AN$ :: IF AN$="Y" THE
N 140
```

VARIABLE COLUMN PROGRAM LISTER

This program will take any basic or XBasic program which has been listed to disk (e.g. LIST"DSK1.PROGRAM") and reformat it to ANY column width you want. The meat of the program is in lines 180-280 plus subroutine 320. The rest is bells and whistles, enabling you to set a left margin, double space between PROGRAM lines (not printer lines), and set up any printer codes you want. Defaults are retained if you go back for another listing. In case you have reset the printer control codes, a RESET is printed via Escape "@" in line 160. If your printer does not use this method, you'll have to do it some other way.

Because the program checks for a <u>number</u> at the 81st and 161st characters in order to see if the next record in the file is actually a new line number, a possible error arises if a number happens to be there which is both an allowable line number (1-32767) and higher than the previous line number. Then the program assumes it is a new lihe and splits it off. Most of these errors can be avoided if your program to be listed has been RESequenced with a regular interval, and you indicate such at the prompt.

Now a brief explanation of the program. Line 180 picks up a record which may be all or part of a program line, and sets a flag indicating something has been started. Line 190 is the key to the program. The VAL function in XBasic will give an actual number if the characters up to the first space in the string are numbers only. This is what we want to see at the beginning of a program line, i.e. a line number. The ON ERROR 420 statement keeps the program from crashing if this is not a valid number. Instead it makes the program go on and will treat the string as the continuation of a line. Line 200 says that if the length of the string is less than 80 than surely the end of a (listed) program line has been reached, and the subroutine at 320 goes to print it, splitting it if necesary and putting in the margins selected if any. If the length WAS exactly 80 then we go for another record and again check for a valid number at the beginning, and if there is one whether it could be a line number (using either the allowable range or the increment). If the number doesn't fit into one of these categories the string is joined to the previous one and we go on.

Note the use of the following subroutines. 380-410 takes a string of ASCII numbers separated by spaces and concatenates them into a single string suitable for sending to the printer. 430-470 is a simulated cursor which allows the use of the enter key but does not require it, allowing the simple press of the Y or N key as well.

That's all there is to it. The program seems to process the records as fast as my printer can handle them. Note that if you use a disk file as your output device, as I did to produce the program listings in these articles, then the file as listed on the disk will be DISPLAY, VARIABLE X where X is the sum of your listing width and the margin. You cannot load this into TI-WRITER directly. You must use a sector editor such as Advanced Diagnostics to change byte 17 (hex >11) of the file descriptor record (catalog sector) from whatever is there to >50. This also means that you can't use widths that exceed 80 columns directly in TIW.

```
100 O$="PIO" :: W=136 :: P$="15" :: INC=10
110 DISPLAY AT(2,4)ERASE ALL:"VARIABLE COLUMN LIST
ER": :"     by Tom Freeman": :"INPUT LISTFILE?":"D
SK";F$: :"OUTPUT DEVICE?":O$: :"MAXIMUM LINE WIDTH
?";W: :"LEFT MARGIN?";M
120 ACCEPT AT(7,4)SIZE(-28)BEEP:F$ :: ACCEPT AT(10
,1)SIZE(-28)BEEP:O$ :: ACCEPT AT(12,21)SIZE(-3)BEE
P:W :: ACCEPT AT(14,14)SIZE(-3)BEEP:M
130 DISPLAY AT(16,1):"PRNTER CTRLS?(SEP.BY SPACES)
";P$ :: DISPLAY AT(20,1):"LINE NUMBER INCREMENT(0
IF  VARIABLE)";INC: : :"DOUBLE SP?  (Y/N) Y"
140 ACCEPT AT(17,1)SIZE(-28)VALIDATE(DIGIT," ")BEE
P:P$ :: ACCEPT AT(21,11)SIZE(-3)BEEP:INC :: CT$=""
 :: PR$=P$ :: GOSUB 430 :: IF K=89 THEN SP=1 ELSE
SP=0
150 DISPLAY AT(24,1)BEEP:"ALL CORRECT(Y/N)? Y" ::
GOSUB 430 :: IF K=78 THEN 120
160 GOSUB 380 :: OPEN #1:"DSK"&F$ :: OPEN #2:O$,VA
RIABLE W+M :: M$=RPT$(" ",M):: STOPFLAG=0 :: PRINT
 #2:CHR$(27);"@";CT$;
170 GOSUB 290 :: IF STOPFLAG=1 THEN 370
180 ON ERROR STOP :: LINPUT #1:A$ :: FLAG=1
190 ON ERROR 420 :: L1=VAL(SEG$(A$,1,POS(A$," ",1)
-1))
200 IF LEN(A$)=80 THEN 220
210 GOSUB 320 :: FLAG=0 :: GOTO 170
220 GOSUB 290 :: IF STOPFLAG=1 THEN 370
230 LINPUT #1:B$ :: A=POS(B$," ",1):: IF A=0 THEN
280
240 ON ERROR 280 :: L2=VAL(SEG$(B$,1,A-1)):: IF IN
C THEN 260
250 IF L2>32767 OR L2<=L1 THEN 280 ELSE 270
260 IF L2<>L1+INC THEN 280 ELSE 270
270 GOSUB 320 :: A$=B$ :: GOTO 190
280 A$=A$&B$ :: IF LEN(B$)<80 THEN 210 ELSE 220
290 IF EOF(1)THEN CLOSE #1 ELSE RETURN
300 IF FLAG THEN GOSUB 320
310 CLOSE #2 :: STOPFLAG=1 :: RETURN
320 IF M=0 THEN PRINT #2:A$ :: GOTO 350
330 L=LEN(A$):: IF L THEN R$=SEG$(A$,1,W)ELSE 350
340 PRINT #2:M$&R$ :: A$=SEG$(A$,W+1,255):: GOTO 3
30
350 IF SP=1 THEN PRINT #2
360 RETURN
370 DISPLAY AT(24,1)BEEP:"DO ANOTHER (Y/N)? Y" ::
GOSUB 430 :: IF K=89 THEN 110 ELSE STOP
380 A=POS(PR$," ",1):: B=LEN(PR$):: IF B=0 THEN RE
TURN
390 IF ASC(PR$)=32 THEN PR$=SEG$(PR$,2,28):: GOTO
380
400 IF A=0 THEN CT$=CT$&CHR$(VAL(PR$)):: RETURN
410 CT$=CT$&CHR$(VAL(SEG$(PR$,1,A))):: PR$=SEG$(PR
$,A+1,28):: GOTO 380
420 RETURN 200
430 CALL KEY(0,K,S):: IF S=-1 THEN 430 ELSE IF K=1
3 OR K=78 OR K=89 THEN 460
440 IF Y=89 THEN Y=30 ELSE Y=89
450 CALL HCHAR(24,21,Y):: GOTO 430
460 IF K=13 THEN K=89
470 CALL HCHAR(24,21,K):: RETURN
```

A random number speaking sub programme by Brian Rutherford

****

This programme will say the numbers from zero to nine hundred and ninety nine

```
10 REM *********************
20 REM *  A RANDOM NUMBER   *
30 REM *      SPEAKING      *
40 REM *    SUB PROGRAMME   *
50 REM *         BY         *
60 REM * BRIAN RUTHERFORD   *
70 REM * 9 BOMBALA STREET   *
80 REM *    DUDLEY  2290    *
90 REM *********************
100 OPTION BASE 0 :: DIM A$(
21),W$(2):: FOR I=0 TO 21 ::
 READ A$(I):: NEXT I :: CALL
 SPGET(A$(20),W$(1)):: CALL
SPGET(A$(21),W$(2)):: CALL C
LEAR
110 DATA "O",TEEN,TWENTY,THI
RTY,FORTY,FIFTY,SIXTY,SEVENT
Y,EIGHTY,NINETY
120 DATA TEN,ELEVEN,TWELVE,T
HIRTEEN,FOURTEEN,FIFTEEN,SIX
,SEVEN,EIGHT,NINE,HUNDRED,AN
D
130 I=1 :: INPUT "NUMBER ":N
M(I)
140 CALL NUMSP(A$(),NM(),W$(
),I)
160 GOTO 130

8000 SUB NUMSP(A$(),NM(),W$(
),I):: DIM NM$(2)
8010 M$=STR$(NM(I)):: L=LEN(
M$):: IF L<>3 THEN 8020 :: N
M$(0)=SEG$(M$,1,1):: NM$(1)=
SEG$(M$,2,1):: NM$(2)=SEG$(M
$,3,1)
8015 CALL SPGET(NM$(0),H$)::
 IF NM$(1)="0" AND NM$(2)="0
" THEN CALL SAY(,H$,,W$(1)):
: GOTO 8090 :: ELSE 8030
8020 NM$(1)=SEG$(M$,1,1):: N
M$(2)=SEG$(M$,2,1)
8030 IF L=1 OR NM$(1)="0" TH
EN 8070 ELSE IF NM$(1)="1" T
HEN 8050
8040 NM(I)=VAL(NM$(1)):: CAL
L SPGET(A$(NM(I)),X$):: IF N
M$(2)="0" THEN 8080 ELSE CAL
L SPGET(NM$(2),Y$):: GOTO 80
60
8050 NM(I)=VAL(NM$(1)&NM$(2)
):: CALL SPGET(A$(NM(I)),X$)
:: IF NM(I)<=15 THEN 8080 EL
SE CALL SPGET(A$(1),Y$)
8060 IF L=3 THEN CALL SAY(,H
$,,W$(1),,W$(2),,X$,,Y$):: G
OTO 8090 :: ELSE CALL SAY(,X
$,,Y$):: GOTO 8090
8070 IF L=3 THEN CALL SPGET(
NM$(2),X$)ELSE CALL SPGET(NM
$(1),X$)
8080 IF L=3 THEN CALL SAY(,H
$,,W$(1),,W$(2),,X$):: GOTO
8090 :: ELSE CALL SAY(,X$)
8090 SUBEND
```

You can use this programme: i.e. lines 100 to 120 line 140, and the main subprogramme starting at line 8000 in any programme you need numbers spoken.

The programme will say any of the numbers between 0 and 999 in any random order

A programme like this is necessary as with the command call say("56"), the computer will say five six, not fifty six. I have not explained how the programme works on this paper as there is not enough room. but if any one wishes an explanation they can come and see me and I will be happy to explain

So just type in a number between zero and nine hundred and ninty nine, when the screen prompts you for a number.

THE LETTER THAT SAYS ALL

Letters,
Personal Computer World,
32-34 Broadwick Street,
London, W1A 2HG,
England.

Furstlia 35,
1335 Snarøya,
Norway.

OCTOBER 1986 PCW 125

Dear Sirs,

I was very surprised, so much so that I had to blink twice to check that I was not imagining it, when I saw "Texas D-I-Y" in Computer Answers in the June issue. I had long expected never to see mention of the TI-99/4a again in your august magazine. All credit to you for helping small as well as large computer users.

Naturally being an active 99'er, I was not too happy with your term obsolete, but quite understand your view. Maybe "obsolete", but certainly far from extinct. TI still provide servicing, software and hardware are not only still being produced but also still being developed, both in the US and in Europe. A new computer, which is upwardly compatible and which will also run PC software, is in pre-production.

One result of TI's withdrawal is that the remnant army has become a close knit community. Your list of suggested contacts was a commendable effort, but is sadly a little out of date - one is active, one is extinct, and one I have never heard of. The main User group in the UK is now TI99/4a Exchange, 40 Barrhill, Patcham, Brighton, Sussex BN1 8UF. They took over where TIHOME left off, after the latter's demise, and publish a very good quarterly magazine, as well as maintaining a substantial software library and organising twice yearly nationwide user shows. Oxford TI User's, which you quote, is run by a long standing TI stalwart, publishing a monthly more technical magazine, and now holding the old TIHOME software library.

Retailers are: Arcade Hardware, 211 Horton Road, Fallowfield, Manchester M14 7QE; Parco Electrics, 2 Devonshire Court, Heathpark, Honiton, Devon EX14 8SB, who also publish a bimonthly magazine; and newcomer New Day, Jerrard Close, Honiton,Devon EX14 8EF, who also publish a tape/disc magazine. Finally Stephen Shaw, 10 Alstone Road, Stockport, Cheshire SK4 5AH, who holds a very extensive disk library of public domain and Freeware software.

The above mentioned sources also detail the many US sources still active. Another result of TI's withdrawal is that most TI software is now available for a fraction of its original cost - arcade game modules previously costing £25 are still available and now cost only £5 (new).

So for those who still wish to 99'er the scene is far from dead. In fact there is so much material available that I cannot keep up with it.

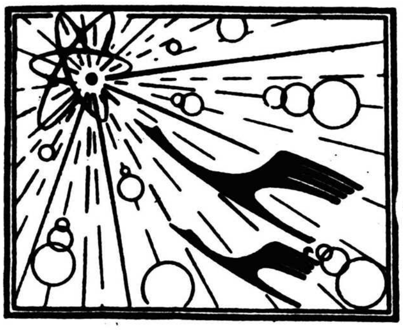
Yours faithfully,

John R.R. Bingham

TI*MES MAGAZINE INDEX 1-13

INDEX TO TI*MES ISSUES 1 TO 13
By Stephen Shaw
A fairly "personal" index but might be of some use to other members!

##	pp	
10	40	!@P+ AND !@P- IN EX BAS
12	44	!@P- : AVOIDING SYNTAX ERS
12	20	28 COLUMN DATA PROGRAM
10	27	32K RAM : DIY PROJECT
12	13	32K RAM PROJECT : CORRECTION
11	60	4ABC : UK TI BULLETIN BOARD
13	5	4FRONT "MAGAZINE" REVIEW
13	35	4FRONT "MAGAZINE" REVIEW #2
10	43	ACCEPT AT (EX BAS)
5	21	ACCURACY : TI DOES WELL
10	54	ADVANCED DIAGNOSTICS :REVIEW
7	5	ADVENTURES : WRITING THEM
8	34	ADVENTURES : WRITING THEM
8	57	ALPHABET SONG : TE2 PROG
6	40	AMBULANCE MODULE REVIEW
3	23	AND IN TI BASIC
8	5	ANTI GLARE TV FILTERS(ROMAG)
8	24	ATLANTIS -INTRIGUE- REVIEW
11	27	AUTOMATIC MOUSE MAZE : PROG
6	53	AUTOSOUND DEMO FOR MINIMEM
10	53	BEST OF TI CARTRIDGES/BOOK
5	31	BIG FOOT MODULE REVIEW
5	12	BINGO NUMBER CALLER PROG
6	20	BINGO PROG ERRATA
4	11	BOOK : ASSEMBLY LANG
9	11	BOOK : ASSEMBLY LANG -MORLEY
9	16	BOOK : BASIC REF MANUAL
10	53	BOOK : BEST OF TI CARTRIDGES
8	7	BOOK : COLLINS PACKS
5	53	BOOK : (FIVE)
8	9	BOOK : FORTH :SCANLON
9	11	BOOK : FORTH TECHNIQUES(PAN)
3	23	BOOK : HARTNELL GAMES
8	9	BOOK : LOGO :A MOLLER
6	43	BOOK : LOGO (ROSS)
9	8	BOOK : LOGO in PAN
9	16	BOOK : LOGO (THORNBURG)
9	25	BOOK : (LOTS)
5	18	BOOK : MASTERING TI99:BROOKS
8	9	BOOK : MULTIPLAN
4	12	BOOK : S DAVIES ASSEMBLY
8	8	BOOK : SAMS
7	59	BOOK : SAMS BOOKS (2)
4	11	BOOK : SMART PROGRAMMER
10	53	BOOK : THE TI PLAYGROUND
10	53	BOOK : TI IN WONDERLAND
11	60	BULLETIN BOARD IN UK: 4ABC
12	50	C : TEST PROGRAM
13	36	C99 : CONDITIONAL EXPRESSION
13	36	C99 : DV80 WORD COUNTER:LIST
13	36	C99 : FOR and SWITCH
12	56	C99 : REVIEW
10	3	CALL A & D WITH PRK
9	42	CALL INIT : CONDITIONAL
4	27	CALL KEY(1&2) SMALL BUG
6	46	CALL LOAD FOR CALL SOUND
11	13	CALL LOADS : LONG LIST
12	43	CALL LOADS : MORE
5	55	CALL SOUND : BELL
4	16	CALL SOUND BELOW 110Hz
6	21	CALL SOUND : EFFECTS
7	61	CALL SOUND EXPERIMENTS
6	46	CALL SOUND WITH CALL LOADS
10	38	CASSETTE CABLE : DIY
9	59	CASSETTE DATA FILES
6	57	CASSETTE PORT PIN OUTS
12	42	CASSETTES : MACHINE CODE ON.
3	4	CAVE MAZE TIB PROGRAM
4	23	CAVE MAZE " CORRECTION
5	31	CHAMPIONSHIP BASEBALL REVIEW
9	10	CHAR SET : GREEK
7	9	CHAR SET : LOWER CASE SET
7	10	CLEANING KEY CONTACTS
12	43	CLEANING PRINTERS, DRIVES &C
8	7	COLLINS STARTER PACKS : REVU
7	49	COLUMBIA THE GEM OF THE OCN.
11	20	COMPACT PROGRAMMING
10	7	COMPARATIVE BASIC TIMINGS
4	28	COMPOSITE VIDEO MODIFICATION
9	25	COMPUTE! BOOK REVIEWS
12	5	CONSOLE HANG UPS
4	24	CONTROL & FCTN KEYS PT 1
5	38	CONTROL & FCTN KEYS PT 2
5	48	CONTROL & FCTN KEYS PT 3
10	5	COPYRIGHT AND FREEWARE
10	18	CORCOMP CONTROLLER PROBLEMS
10	18	CORCOMP RS232<>MYARC EXP MEM
3	29	CORRECTIONS : #2 PAGES 4 &14
4	23	CORRECTION : CAVE MAZE #3
4	7	CORRECTION TO #3 PAGE 31
6	20	CORRECTION : BINGO CALLER #5
7	60	CORRECTION : ISSUE 6 PAGE 55
8	48	CORRECTION: DISKFORMAT #7
8	34	CORRECTION : ISSUE 7 ADV PRG
9	9	CORRECTION : VIDEO : ISSUE 8
10	8	CORRECTION : ISSUE 8 FORTH
10	10	CORRECTION : ISSUE 9 DISKS
13	6	CORRECTION #12 PASCAL LIST
10	56	CORRECTION : MINIMEM LBLA
8	24	CRAZY CLIFF -STAINLESS- REVU
10	57	CRYPTOGRAPHY : PROGRAM
8	57	CURSOR CHANGER PROG :XB+32K
12	24	CURSOR KEYBOARD
9	40	DANCING STICKMAN TIB PROG
11	4	DATA BAR READER : OSCAR
11	15	DATA FILES : PART 1
9	59	DATA FILES ON CASSETTE
4	8	DATA READ AND RESTORE
11	44	DATABASE 1 : REVIEW
12	4	DEBUGGING A BASIC PROGRAM
6	56	DECIMAL TO BINARY CONVERSION
6	21	DEEP IN THE HEART OF TEXAS
6	39	DEMON ATTACK REVIEW
6	27	DIABLO REVIEW
12	21	DIARY PROGRAM : SIMPLEST
11	35	DISK CONTROLLER: MYARC
11	26	DISK MENU LOADER : PROGRAM
7	47	DISK MENU LOADER (XB+32K)
10	7	DISK NOT INITIALISED : HELP!
9	12	DISK SECTOR ACCESS / BASIC
10	10	DISK SECTOR CORRECTION ISSU9
13	14	DISK SECTOR INFO :>0 TO >21
10	11	DISKS : EFFICIENT USE OF
10	12	DISKS : FRACTURED FILES
10	48	DISPLAY ERASE ALL (XB)
10	27	DIY 32K RAM PROJECT
13	49	DIY EXPANSION BOX PROJECT
12	34	DRAUGHTS PROGRAM IN TI BASIC
10	9	DRAW A BIT 1 & 2 : REVIEW
13	24	DUNGEON GOLD : TIB PROGRAM
9	4	ECONOMIC PROGRAMMING /BASIC
12	4	EDITING A BASIC PROGRAM
12	45	EDITOR ASSEMBLER : PIO.
12	45	EDITOR ASSEMBLER : DSR ERROR
10	47	EFFICIENT EX BAS PROGRAMMING
10	11	EFFICIENT USE OF DISKS
10	3	ENHANCED BASIC : CALL A & D
12	16	ENHANCED BASIC : FILES
12	17	EOF
6	40	ESPIAL REVIEW
10	43	EX BAS : ACCEPT AT
11	54	EX BAS BY MECHATRONIK : REVU
11	13	EX BAS "LOAD" TO TITLE SCRN
10	47	EX BAS : SHORTER PROGRAMS
8	55	EX BAS SPEECH : PHRASES
7	51	EX BAS SUB PROGRAMS
8	53	EX BAS SUBPROGRAM PARAMETERS
10	39	EX BAS SUBPROGRAMS
10	39	EX BAS TUTORIAL (McGOVERN)
11	20	EX BAS TUTORIAL (McGOVERN)
10	48	EX BAS : VARIABLE STORAGE
13	49	EXPANSION BOX : DIY PROJECT
13	9	EXPANSION IN 1986 :WORTH IT?
10	55	EXPLORER (MILLER) : REVIEW
13	7	EXPMEM1 WITH MINIMEMORY
2	20	EXTENDED BASIC : REVIEW
7	51	EXTENDED BASIC TUTOR
8	53	EXTENDED BASIC TUTOR
9	51	EXTENDED BASIC TUTOR
10	16	FASTER SIN(X) IN BASIC
12	14	FILES : ARTICLE 2
12	16	FILES : ENHANCED BASIC
12	19	FILES : ERROR TRAPPING
12	17	FILES : FINDING FILE END
12	14	FILES : HIGH SCORES
12	16	FILES : LONG TAPE FILES
12	15	FILES : VERIFY ON CASSETTE
11	4	FOOTBAL MODULE PHM3009
12	46	FORTH : ADD VARIABLES
10	19	FORTH : AMEND SCREENS 58 &59
12	44	FORTH : ARRAYS
8	44	FORTH : BLOAD & BSAVE
8	9	FORTH : BOOK REVIEW :SCANLON
10	14	FORTH : CALL SOUND UTILITY



New Day
COMPUTING

Jerrard Close
Honiton
Devon
EX14 8EF
Tel.(0404) 41856

4FRONT MAGAZINE for the TI99/4a

Dear Friend,

With regard to NEW DAY COMPUTING and 4FRONT, let me first give you some background information.

I have been involved for some years in the TI scene over here, and last year decided to start up a computer business of my own, with the TI taking up about half of my resources and sanity. The idea of a mag on cassette was born, and the fact that many scoffed at the thought ('leave the idiot to it - he'll soon realize he's wasting his time') only made me more determined to have a go. After much time, money, and hair dropping out, the first issue hit the streets. To my shock, the reaction was embarrassingly favourable - 4FRONT was a success with all who saw it.

In essence, the **TAPE** version is made up of articles and programs that will run in Basic OR Extended Basic. Some of the programs, however do need Exbas, PRK etc. The **DISK** version assumes Extended Basic/32k as standard. There are programs on tape in Basic that do not appear on the disk version, and items on disk that could not be included on cassette - eg Ed/Ass material. That way everyone gets good value according to his configuration. There are hints, tips, utilities, games, in fact all that you'd expect to find in a magazine - but ready to run!

Prices outside Europe:

Single issue	**- POUNDS 6.00 (airmail)**
Annual sub (quarterly)	**- POUNDS 22.00 (airmail)**
	- POUNDS 19.00 (seamail)
Bulk sub for User Groups:	
(min. 10 copies to one address)	**- POUNDS 20.00 ea. (air)**
	- POUNDS 17.00 ea. (sea)

Payment by bank draft drawn on London in pounds sterling please.

European prices:

Single issue	**- POUNDS 5.00**
Annual sub (quarterly)	**- POUNDS 20.00**
Bulk sub for User Groups:	
(min. 10 copies to one address)	**- POUNDS 15.00 ea.**

Payment by Eurocheque/bank draft drawn on London in pounds sterling.

Clubs and users requested to respect copyright!

Please state single or double sided when ordering disk version.

Well, that's about it. Go on, spoil yourself and try it. Then if you like it, don't pirate it, get your friends to purchase their own copy, thus keeping one balding English idiot half-sane for a little while longer....

Harry Pridmore

4FRONT

QUALITY PRODUCT

CONTENTS

Inside this issue..............